

MasterSCADA 4D™

Руководство пользователя

Версия 1.2.X



Краткое описание руководства пользователя

Данное руководство предназначено для освоения программного обеспечения MasterSCADA 4D. Оно рассчитано как для начального освоения системы, так и для продвинутых пользователей.

В этом учебном пособии описаны основные принципы и методы, применяемые при разработке проектов в MasterSCADA 4D.

Адрес: 125124, г. Москва,

Ул. Ямского Поля 3-я, 2 к3

Телефон: +7 (495) 228-70-77

Email: info@masterscada.ru

Сайт: www.masterscada.ru

1.	Инсталляция.....	47
1.1.	Состав MasterSCADA 4D.....	47
1.2.	Системные требования среды разработки.....	49
1.3.	Системные требования для среды исполнения.....	49
1.4.	Поддерживаемые контроллеры	52
1.5.	Установка MasterSCADA 4D	53
1.5.1.	Установка среды разработки MasterSCADA 4D	54
1.5.2.	Описание ярлыков MasterSCADA 4D	59
1.5.2.1.	Обновление версии на ОС Windows	60
1.5.2.2.	Удаление MasterSCADA 4D.....	60
1.5.2.3.	Запуск среды разработки	60
1.5.2.4.	Установка дополнительных компонентов.....	61
1.5.2.4.1.	Установка OPC Core Components Redistributable	61
1.5.2.4.2.	Установка Microsoft.NET Framework	64
1.5.3.	Установка среды исполнения MasterSCADA 4D.....	66
1.5.3.1.	Установка на ОС Windows	66
1.5.3.1.1.	Установка исполнительной системы MasterSCADA 4D RT.....	66
1.5.3.1.2.	Описание ярлыков MasterSCADA 4D RT.....	71
1.5.3.1.3.	Обновление версии исполнительной системы на ОС Windows.....	71
1.5.3.1.4.	Удаление MasterSCADA 4D RT	72
1.5.3.2.	Установка на ОС Linux или Эльбрус.....	72
1.5.3.2.1.	Установка на ОС Linux или Эльбрус.....	72
1.5.3.2.2.	Обновление среды исполнения на ОС Linux или Эльбрус	78
1.5.3.2.3.	Удаление среды исполнения на ОС Linux или Эльбрус.....	78
1.5.3.3.	Установка на ОС Windows Embedded	78
1.5.3.4.	Установка на ОС QNX	79
1.5.4.	Установка и удаление клиента на ОС Windows.....	79
1.6.	Лицензионная защита	82
1.7.	Совместимость с антивирусами и файрволами	83

2.	Быстрый старт	86
2.1.	БС. Урок 1.....	86
2.1.1.	БС. Урок 1. Создание проекта	86
2.1.2.	БС. Урок 1. Конфигурирование дерева системы	89
2.1.3.	БС. Урок 1. Создание логической части проекта	95
2.1.4.	БС. Урок 1. Создание окна управления	106
2.1.5.	БС. Урок 1. Запуск.....	118
2.2.	БС. Урок 2.....	122
2.2.1.	БС. Урок 2. Конфигурирование всплывающего окна	122
2.2.2.	БС. Урок 2. Запуск.....	130
2.3.	БС. Урок 3.....	130
2.3.1.	БС. Урок 3. Объектный подход при создании проекта	130
2.3.2.	БС. Урок 3. Запуск.....	143
2.3.3.	БС. Урок 3. Операции с библиотеками.....	144
3.	Обзор инструментальной среды MasterSCADA 4D	150
3.1.	Основные характеристики	150
3.2.	Понятия.....	153
3.3.	Описание интерфейса	157
3.4.	Демонстрационные проекты	158
3.5.	Лицензионная политика	159
3.6.	Принятые сокращения и обозначения.....	164
4.	Принципы проектирования	165
4.1.	Анализ объекта автоматизации.....	165
4.2.	Разработка типовых элементов.....	166
4.3.	Создание структуры проекта	169
4.4.	Разработка и отладка программ	180
4.5.	Разработка окон.....	182
4.6.	Конфигурирование связей	183
4.7.	Формирование задач узлов	185

4.8.	Автономная отладка в реальном времени.....	186
4.9.	Компиляция проекта и загрузка конфигураций в узлы	188
5.	Интерфейс редактора проекта	190
5.1.	Стартовое меню	193
5.2.	Панель инструментов.....	194
5.2.1.	Настройки редактора.....	195
5.2.1.1.	Настройки среды	197
5.2.1.1.1.	Вкладка Внешний вид	198
5.2.1.1.2.	Вкладка Компиляция	200
5.2.1.1.3.	Вкладка Клиент HMI	202
5.2.1.1.4.	Вкладка Лог.....	203
5.2.1.1.5.	Вкладка Разное	204
5.2.1.1.6.	Диалог Восстановить умолчания	205
5.2.1.2.	Диалог параметров отчета об ошибках	206
5.2.2.	Панель отмены/возврата действий	208
5.2.3.	Управление проектами	210
5.2.4.	История версий	213
5.2.5.	Вкладка инструментов Проект.....	215
5.2.6.	Вкладка инструментов Исполнение	216
5.2.6.1.	Окно Ошибки проекта	218
5.2.7.	Вкладка инструментов Отладка.....	219
5.2.7.1.	Редактор набора наблюдений.....	220
5.2.8.	Табличное представление проекта	222
5.2.8.1.	Формирование запросов	226
5.2.8.1.1.	Тип условия запроса - Свойство	231
5.2.8.1.2.	Тип условия запроса - Отношение	232
5.2.8.2.	Работа с результатами запросов	232
5.2.8.2.1.	Настройка таблицы результатов запроса	236
5.2.8.2.2.	Фильтрация и сортировка результатов запроса	239

5.2.8.2.3.	Изменение значений свойств в таблице результатов запросов	240
5.2.8.2.4.	Поиск и замена значений в таблице результатов запросов	241
5.2.8.2.5.	Редактирование результатов запросов в CSV	241
5.2.9.	Панель управления интерфейсом	242
5.2.10.	Панель инструментов Справка	242
5.3.	Дерево MasterSCADA	243
5.4.	Панель редакторов	248
5.5.	Контекстное меню	249
5.6.	Контекстная панель	251
5.7.	Панель свойств	253
5.7.1.	Панель свойств. Права	257
5.7.2.	Панель свойств. События	257
5.7.3.	Панель свойств. Отображение свойств, задействованных в связях	258
5.7.4.	Панель свойств. Отображение связей библиотечного элемента	259
5.7.5.	Панель свойств. Отображение свойств, имеющих структуру	261
5.8.	Панель веток дерева	262
5.9.	Легенда	263
5.10.	Типовые инструменты и операции	263
5.10.1.	Связывание переменных	263
5.10.2.	Элементарные инструменты	264
5.10.3.	Типовые инструменты окон	269
5.10.4.	Типовые операции	270
5.10.5.	Типовые инструменты дерева	275
5.10.6.	Горячие клавиши редактора проектов	276
5.10.7.	Работа с вкладками рабочей области	279
5.11.	Операции в редакторе проекта	280
5.11.1.	Операции с проектами	280
5.11.1.1.	Окно выбора трансляции	281
5.11.1.2.	Диалог открытия проекта	283

5.11.1.2.1.	Диалог создания проекта	289
5.11.1.3.	290
5.11.1.4.	Диалог сохранения проекта	291
5.11.1.5.	Автоматическое сохранение действий разработчика	293
5.11.1.6.	Параллельная разработка проекта	293
5.11.2.	Операции с элементами	294
5.11.2.1.	Диалог установки наследования	295
5.11.2.2.	Создание элементов	296
5.11.2.3.	Работа с именами элементов	296
5.11.2.4.	Выделение элементов	297
5.11.2.5.	Подсветка	298
5.11.2.6.	Автоматический переход	298
5.11.2.7.	Автоматическая прокрутка	298
5.11.2.8.	Перетаскивание с удержанием ПК	299
5.11.2.9.	Дублирование и копирование	300
5.11.2.10.	Восстановление умолчаний	301
5.11.2.11.	Восстановление связей	301
5.11.2.12.	Перемещение элементов в дереве	303
5.11.2.13.	Поиск элементов	303
5.11.3.	Локализация проекта	305
5.11.4.	Скрипт С#	306
5.11.4.1.	Пример использования скрипта С#	309
5.11.4.2.	Поиск и замена в тексте С#	313
5.11.5.	Масштабирование в редакторе проекта	315
5.11.5.1.	Комбинированный инструмент масштабирования	315
5.11.5.2.	Инструменты масштабирования	318
5.12.	Служебные диалоговые окна	318
6.	Проект в MasterSCADA 4D	319
6.1.	Методы разработки типов элементов	320

6.2.	Дерево системы	321
6.2.1.	Узел	322
6.2.1.1.	Панель свойств Узла	325
6.2.1.1.1.	Свойство Параметры запуска RT	334
6.2.1.2.	Параметры узла	337
6.2.1.3.	Графический интерфейс	338
6.2.1.3.1.	Шаблон экрана	339
6.2.1.3.1.1.	Панель свойств Шаблона экрана	340
6.2.1.3.2.	Медиа	344
6.2.1.3.2.1.	Элемент Медиа	344
6.2.1.3.3.	Окна	344
6.2.1.4.	Программы	345
6.2.1.5.	Сообщения	345
6.2.1.6.	Службы	346
6.2.1.6.1.	Сохранение состояния	347
6.2.1.7.	Архивы	350
6.2.1.7.1.	Основной архив данных	352
6.2.1.7.2.	Элемент Архив данных	362
6.2.1.7.3.	Архив данных в памяти	364
6.2.1.7.4.	БД справочников	368
6.2.1.7.5.	Базы данных.....	371
6.2.1.7.5.1.	SQLITE	372
6.2.1.7.5.2.	POSTGRESQL	374
6.2.1.7.5.3.	MS SQL	375
	Настройки базы данных MSSQL.....	377
6.2.1.7.6.	Конфигурирование архива данных.....	378
6.2.1.7.6.1.	Слой данных	382
6.2.1.7.6.2.	Шаблоны архивирования	382
6.2.1.7.6.3.	Передача архивных данных	386

6.2.1.8.	Задачи	387
6.2.1.9.	Отчеты.....	389
6.2.1.10.	Внешние каналы	389
6.2.1.11.	Протоколы	390
6.2.1.12.	Встроенные каналы	392
6.2.1.13.	Встроенные модули.....	392
6.2.1.14.	Канал.....	393
6.2.2.	Безопасность	398
6.2.2.1.	Панель свойств Безопасность	403
6.2.2.2.	Пользователи	408
6.2.2.3.	Роли.....	409
6.2.2.4.	Права доступа	414
6.2.2.5.	Архивы	415
6.2.2.5.1.	Сетевая БД настроек	415
6.2.2.5.2.	Архив сообщений ИБ.....	418
6.2.2.6.	Настройки пользователей.....	421
6.2.2.7.	Настройка прав пользователей в режиме разработки.....	424
6.2.2.7.1.	Получение параметров текущего пользователя.....	430
6.2.2.7.2.	Задание стартового окна для пользователя и роли	431
6.2.2.7.3.	Журнал действий пользователей.....	433
6.2.2.7.4.	Настройка пользователя по умолчанию	435
6.2.2.8.	Работа с пользователями в режиме исполнения	436
6.2.2.9.	Active Directory	439
6.2.3.	Получение и отправка данных.....	453
6.2.3.1.	Стандартные протоколы	454
6.2.3.1.1.	Modbus	455
6.2.3.1.1.1.	Modbus TCP	456
	Свойства протокола Modbus TCP	457
	Модуль Modbus TCP	459

6.2.3.1.1.2.	Modbus RTU.....	461
	Свойства протокола Modbus RTU.....	462
	Модуль Modbus RTU.....	465
6.2.3.1.1.3.	Свойства модулей Modbus TCP и Modbus RTU	467
6.2.3.1.1.4.	Свойства каналов Modbus	472
6.2.3.1.1.5.	Modbus RTU over TCP.....	474
	Свойства протокола Modbus RTU over TCP	475
6.2.3.1.2.	DCON.....	477
6.2.3.1.2.1.	Свойства протокола DCON.....	478
6.2.3.1.2.2.	Модуль DCON	480
	Свойства модуля DCON.....	482
6.2.3.1.3.	OPC UA.....	483
6.2.3.1.3.1.	Свойства OPC UA.....	485
6.2.3.1.3.2.	Диалог подключения к серверу OPC UA.....	490
6.2.3.1.4.	OPC DA.....	494
6.2.3.1.4.1.	Свойства OPC DA.....	496
6.2.3.1.4.2.	Диалог подключения к серверу OPC DA.....	499
6.2.3.1.5.	OPC HDA.....	501
6.2.3.1.5.1.	Свойства OPC HDA	503
6.2.3.1.5.2.	Диалог подключения к серверу OPC HDA	506
6.2.3.2.	Дополнительные протоколы	508
6.2.3.2.1.	IEC61850	511
6.2.3.2.1.1.	Свойства IEC61850	516
6.2.3.2.2.	MQTT.....	520
6.2.3.2.2.1.	Свойства MQTT	525
6.2.3.2.2.2.	Свойства канала MQTT	528
6.2.3.2.2.3.	Чтение сложных типов данных в формате JSON из MQTT	529
6.2.3.2.3.	IEC104	529
6.2.3.2.3.1.	Свойства IEC104	531

6.2.3.2.3.2.	Свойства канала IEC104.....	535
6.2.3.2.4.	SNMP Client	537
6.2.3.2.4.1.	Свойства SNMP Client	539
6.2.3.2.4.2.	Свойства канала SNMP	543
6.2.3.2.5.	SiemensPLC	544
6.2.3.2.5.1.	Свойства SiemensPLC	548
6.2.3.2.5.2.	Свойства каналов SiemensPLC	551
6.2.3.2.5.3.	Диалог импорта каналов SiemensPLC	553
6.2.3.2.6.	Mitsubishi.....	557
6.2.3.2.6.1.	Свойства Mitsubishi	560
6.2.3.2.6.2.	Свойства каналов Mitsubishi	563
6.2.3.2.6.3.	Диалог импорта каналов Mitsubishi.....	565
	Экспорт переменных из GX Works	567
6.2.3.2.6.4.	Настройка контроллера Mitsubishi	569
6.2.3.2.7.	Omron	573
6.2.3.2.7.1.	Свойства протоколов Omron	577
	Свойства OmronFinsUDP	577
	Свойства OmronFinsSerialCOM	580
	Свойства OmronFinsSerialTCP.....	584
6.2.3.2.7.2.	Свойства каналов Omron	587
6.2.3.2.7.3.	Диалог импорта каналов Omron	589
	Экспорт переменных из CX-Programmer.....	591
6.2.3.2.8.	Logica_RSBus.....	592
6.2.3.2.8.1.	Свойства модулей LogicaRSBus.....	595
6.2.3.2.8.2.	Свойства каналов Logica RSBus	597
6.2.3.2.8.3.	SPG741	599
6.2.3.2.8.4.	SPT941_10.....	602
6.2.3.2.8.5.	SPT943_1.....	604
6.2.3.2.9.	Logica_SPBus.....	608

6.2.3.2.9.1.	Диалог импорта Logica_SPBus	612
6.2.3.2.9.2.	Свойства устройства Logica_SPBus	613
6.2.3.2.9.3.	Свойства каналов устройства Logica_SPBus.....	615
6.2.3.2.9.4.	Устройство СПГ761.1	615
6.2.3.2.9.5.	Устройство СПГ961.1	619
6.2.3.2.10.	Mercury.....	623
6.2.3.2.10.1.	Свойства модулей Mercury.....	629
6.2.3.2.10.2.	Mercury 230AR	630
6.2.3.2.10.3.	Mercury 230ART	632
6.2.3.2.10.4.	Mercury 230ARTP	636
6.2.3.2.10.5.	Mercury 230ART2P	640
6.2.3.2.10.6.	Mercury 234ARTP	645
6.2.3.2.10.7.	Mercury 234ART2P	650
6.2.3.2.10.8.	Mercury 236ART	655
6.2.3.2.10.9.	Mercury 236ARTP	659
6.2.3.2.11.	SET4.....	664
6.2.3.2.11.1.	Свойства модулей SET4.....	668
6.2.3.2.11.2.	SET-4.....	672
6.2.3.2.11.3.	PSCH-4.....	678
6.2.3.2.12.	Energomera.....	685
6.2.3.2.12.1.	Свойства модулей Energomera.....	689
6.2.3.2.12.2.	CE301	691
6.2.3.2.12.3.	CE303	695
6.2.3.2.13.	Pulsar.....	701
6.2.3.2.13.1.	Свойства модулей Pulsar.....	705
6.2.3.2.14.	ESKO	707
6.2.3.2.14.1.	Модуль ESKO-T.....	711
6.2.3.2.14.2.	Свойства модулей ESKO	713
6.2.3.2.15.	Тепловычислители Теплоком.....	714

6.2.3.2.15.1. ВКТ9	715
Свойства ВКТ9	719
Тепловычислитель ВКТ9.....	722
6.2.3.2.15.2. ВКТ7	726
Свойства ВКТ7	731
Тепловычислитель ВКТ7.....	733
6.2.3.2.16. Vzljot	738
6.2.3.2.16.1. Свойства модулей Vzljot	743
6.2.3.2.16.2. TSR-034(033).....	744
6.2.3.2.16.3. TSR-026M.....	746
6.2.3.2.16.4. TSR-043	749
6.2.3.2.16.5. TSR-032	753
6.2.3.2.16.6. ER	755
6.2.3.2.16.7. TSR-024M.....	756
6.2.3.2.16.8. TSR-042	759
6.2.3.2.17. Mkts	763
6.2.3.2.17.1. Свойства модуля МКТС	769
6.2.3.2.17.2. МКТС.....	770
6.2.3.2.18. Протоколы для работы с базами данных.....	774
6.2.3.2.18.1. Протокол PostgreSQL.....	774
6.2.3.2.18.2. Протокол MSSQL.....	776
6.2.3.2.18.3. Протокол MySQL.....	778
6.2.3.2.18.4. Свойства протоколов для работы с базами данных.....	781
6.2.3.2.18.5. Свойства Запроса к базе данных.....	782
6.2.3.2.18.6. Функциональные блоки для работы с базами данных	785
6.2.3.2.19. ВАСnet	785
6.2.3.2.19.1. Диалог импорта каналов ВАСnet	789
6.2.3.2.19.2. Поиск ВАСnet-устройств.....	792
6.2.3.2.19.3. Окно фильтра ВАСnet	794

6.2.3.2.19.4.	Свойства протокола BACnet.....	795
6.2.3.2.19.5.	Свойства канала BACnet.....	798
6.2.3.2.19.6.	Рекомендации по настройке протокола BACnet	799
6.2.3.2.20.	TEM	802
6.2.3.2.20.1.	Свойства модулей TEM	808
6.2.3.2.21.	Dlms	810
6.2.3.2.21.1.	Поддерживаемые объекты протокола Dlms.....	813
	Объекты DLMS	816
	Объект DATA.....	816
	Объект Register	816
	Объект Extended Register	817
	Объект Demand Register	818
	Объект Profile Generic	818
	Кодирование OBIS.....	820
	Группа А.....	820
	Группа В.....	821
	Группа С.....	822
	Группа D.....	824
	Группа E.....	827
	Группа F.....	828
6.2.3.2.21.2.	Диалог импорта каналов Dlms	828
	Структура Excel-файла.....	830
6.2.3.2.21.3.	Рекомендации по настройке протокола Dlms	831
6.2.3.2.21.4.	Свойства Dlms Device.....	832
6.2.3.2.21.5.	Свойства канала Dlms.....	836
6.2.3.2.22.	SRTP	837
6.2.3.2.22.1.	Свойства протокола GE Open.....	839
6.2.3.2.22.2.	Свойства каналов протокола GE Open	841
6.2.3.3.	MasterSCADA 4D в роли SLave	842

6.2.3.3.1.	Внешние каналы	842
6.2.3.3.1.1.	MasterSCADA 4D как сервер OPC DA	846
6.2.3.3.1.2.	MasterSCADA 4D в роли Modbus Slave	853
6.2.3.3.1.3.	IEC104 Slave	857
6.2.3.3.2.	MasterSCADA 4D как сервер OPC UA	861
6.2.3.3.3.	IEC61850 Slave	868
6.2.3.4.	Типовые каналы обмена	874
6.2.3.5.	Отказ	875
6.2.3.6.	Новые типы протоколов и модулей	875
6.2.3.7.	Рекомендации по настройке DCOM.....	875
6.3.	Дерево объектов.....	875
6.3.1.	Типы и экземпляры MasterSCADA 4D.....	876
6.3.2.	Объект.....	882
6.3.2.1.	Панель свойств Объекта.....	885
6.3.3.	Экземпляр объекта	886
6.3.4.	Параметры объектов	888
6.3.5.	Папка.....	889
6.3.6.	Тег.....	889
6.3.6.1.	Панель свойств Тега.....	890
6.3.6.2.	Стандартные теги.....	891
6.3.6.2.1.	Вещественный тег	892
6.3.6.2.2.	Целый тег	892
6.3.6.2.3.	Логический тег	892
6.3.6.3.	Новый тип тега	892
6.3.7.	Экземпляр тега	892
6.3.8.	Ресурсы	894
6.3.8.1.	Программы.....	894
6.3.8.2.	Окна	895
6.3.8.3.	Сообщения	895

6.3.8.4.	Отчеты.....	895
6.3.9.	Установка пароля на содержимое объектов.....	895
6.4.	Дерево библиотек.....	898
6.4.1.	Библиотека Стандартная.....	900
6.4.1.1.	Типы данных.....	902
6.4.1.1.1.	StatusCode.....	903
6.4.2.	Пользовательская библиотека.....	905
6.4.2.1.	Создание/подключение библиотеки.....	906
6.4.2.2.	Работа с типом Объекта.....	909
6.4.2.3.	Создание типа функционального блока.....	911
6.4.2.4.	Создание типа Справочника.....	914
6.4.2.5.	Работа с Константами.....	917
6.4.3.	Библиотека BaseObjects.....	920
6.4.3.1.	BaseObjects.ФБ.Системные.....	921
6.4.3.1.1.	SetDateAndTime.....	922
6.4.3.1.2.	SEND_EMAIL.....	923
6.4.3.1.3.	READ_ARCHIVE_DATA.....	925
6.4.3.1.4.	PlaySound.....	928
6.4.3.1.5.	SEND_SMS.....	929
6.4.3.1.6.	ReportFB.....	932
6.4.3.1.7.	GET_CURRENT_USERNAME.....	936
6.4.3.1.8.	SysProcessCreate.....	937
6.4.3.1.9.	SysProcessCreateWithResult.....	938
6.4.3.1.10.	READ_SMS.....	940
6.4.3.1.11.	GetObjectName.....	942
6.4.3.1.12.	SysProcessCreateAsync.....	942
6.4.3.1.13.	SysProcessCreateWithResultAsync.....	944
6.4.3.1.14.	COMPort.....	946
6.4.3.1.15.	COMPortSync.....	948

6.4.3.1.16.	COMPortByte	950
6.4.3.1.17.	COMPortByteSync.....	952
6.4.3.1.18.	EventsCounter.....	955
6.4.3.1.19.	GetActualEvents.....	956
6.4.3.1.20.	GetNewEvents.....	958
6.4.3.1.21.	WriteArchiveData.....	960
6.4.3.1.22.	FireBaseEvent.....	961
6.4.3.1.23.	FileToString	964
6.4.3.1.24.	StringToFile	965
6.4.3.1.25.	StructArrayToFile	967
6.4.3.1.26.	XMLString.....	968
6.4.3.1.27.	PING	975
6.4.3.1.28.	GetSystemInfo.....	976
6.4.3.1.29.	RedundancyControl	980
6.4.3.1.30.	ExportEventsArchive	983
6.4.3.1.31.	FileIntegrityControl	985
6.4.3.1.32.	GetOpcUaServerInfo	988
6.4.3.1.33.	GetDataArchiveStatistic	990
6.4.3.1.34.	GetEventsArchiveStatistic	990
6.4.3.1.35.	GetTaskStatistics	991
6.4.3.2.	BaseObjects.ФБ.Users	993
6.4.3.2.1.	UsersGet.....	996
6.4.3.2.2.	UsersAdd	997
6.4.3.2.3.	UsersSetGroups.....	999
6.4.3.2.4.	UsersDelete	1000
6.4.3.2.5.	UsersGetGroups	1001
6.4.3.2.6.	UsersSetPassword	1003
6.4.3.2.7.	UsersRename	1004
6.4.3.2.8.	UsersGetSettings.....	1005

6.4.3.2.9.	UsersChangeOwnPassword	1008
6.4.3.2.10.	UsersGetADUserGroups.....	1009
6.4.3.2.11.	UsersGetBlocked	1011
6.4.3.2.12.	UsersSetSettings	1012
6.4.3.2.13.	UsersSetBlocked.....	1015
6.4.3.2.14.	UsersCloseSession.....	1017
6.4.3.2.15.	UsersGetActiveSessions	1018
6.4.3.2.16.	UsersGetAllowedAddress.....	1019
6.4.3.2.17.	UsersSetAllowedAddress	1020
6.4.3.2.18.	UsersGetAllowedTime.....	1021
6.4.3.2.19.	UsersSetAllowedTime	1022
6.4.3.2.20.	UsersGetNeedResetPassword.....	1024
6.4.3.2.21.	UsersSetNeedResetPassword	1025
6.4.3.2.22.	UsersRoleAdd.....	1026
6.4.3.2.23.	UsersRoleDelete.....	1030
6.4.3.2.24.	UsersGetAllRoles.....	1031
6.4.3.2.25.	UsersGetRoleSettings.....	1033
6.4.3.2.26.	UsersSetRoleSettings	1036
6.4.3.2.27.	UsersRestoreRoleSettings	1040
6.4.3.2.28.	UsersRestoreUser.....	1041
6.4.3.2.29.	UsersSetRoleControlRight.....	1042
6.4.3.2.30.	UsersGetRoleControlRight	1045
6.4.3.2.31.	UsersDeleteRoleControlRight	1047
6.4.3.2.32.	BaseObjects.Функции.Системные	1049
6.4.3.2.33.	Sleep.....	1050
6.4.3.2.34.	RestartPLC	1052
6.4.3.2.35.	GetVariableInfo	1053
6.4.4.	Библиотека Protocols	1055
6.4.4.1.	Protocols.ФБ.....	1055

6.4.4.1.1.	SqlRequest.....	1056
6.4.4.1.2.	SqlRequestWithResult.....	1058
6.4.4.1.3.	HttpClient	1060
6.4.4.1.4.	UdpClient	1063
6.4.5.	Диалог Обновить библиотеки.....	1064
6.5.	Параметры.....	1065
6.5.1.	Панель свойств Параметров	1066
6.5.2.	Окно настройки параметра.....	1069
6.5.3.	Создание нового типа данных.....	1072
6.5.3.1.	Новый тип данных Массив.....	1074
6.5.3.2.	Новый тип данных Перечисление	1079
6.5.3.3.	Новый тип данных Диапазон	1080
6.5.3.4.	Новый тип данных Структура.....	1080
6.6.	Работа со шкалами	1084
6.6.1.	Шкала AI.....	1085
6.6.2.	Шкала DI.....	1090
6.6.3.	Шкала.....	1092
6.6.4.	Динамическая шкала	1092
6.6.5.	Новый тип шкалы	1093
6.7.	Свойства.....	1093
6.7.1.	Категория свойств Общие	1094
6.7.1.1.	Свойство Сохранять	1098
6.7.2.	Категория Задача	1101
6.7.3.	Категория Масштабирование	1106
6.7.4.	Категория свойств Служебные.....	1108
6.7.5.	Категория свойств Разрешения	1111
6.8.	Работа с сообщениями	1114
6.8.1.	Типы стандартных сообщений.....	1116
6.8.1.1.	Тревога.....	1116

6.8.1.2.	Контроль границ	1120
6.8.1.2.1.	Контроль границ. Тексты сообщений	1121
6.8.1.3.	Контроль скорости изменения	1121
6.8.1.4.	Базовое сообщение	1122
6.8.1.5.	Действие пользователей	1123
6.8.1.6.	Системное сообщение	1124
6.8.1.7.	Сообщение ИБ.....	1125
6.8.2.	Новый тип тревоги	1126
6.8.3.	Архивирование сообщений	1127
6.8.4.	Список программных имен параметров тревог	1129
6.9.	Настройка сетевых проектов	1130
6.9.1.	Резервирование узлов.....	1131
6.10.	Справочная информация	1136
6.10.1.	Соответствие элементов JSON и MasterSCADA 4D.....	1137
6.10.2.	Настройка DCOM Windows.....	1141
6.10.3.	Структура таблиц баз данных с архивами	1143
6.11.	Рекомендации по созданию проектов	1149
6.11.1.	Формирование отчета с помощью ФБ READ_ARCHIVE_DATA	1149
7.	Программирование в MasterSCADA 4D.....	1166
7.1.	Общие элементы программ.....	1167
7.1.1.	Параметры программ.....	1167
7.1.2.	Ключевые слова	1171
7.1.3.	Обработка En/EnO.....	1171
7.1.4.	Типы данных.....	1172
7.1.4.1.	Элементарные типы данных.....	1172
7.1.4.2.	Производные типы данных.....	1173
7.1.4.3.	Специальные типы данных	1174
7.2.	Общие средства редактирования программ.....	1174
7.2.1.	Структура программы в дереве	1174

7.2.2.	Клеммники	1176
7.2.2.1.	Клеммник входов	1176
7.2.2.2.	Клеммник выходов.....	1176
7.2.3.	Легенда	1177
7.2.3.1.	Легенда. Вкладка Локальные	1177
7.2.3.2.	Легенда. Вкладка Параметры	1178
7.2.3.3.	Легенда. Вкладка Заголовки ST	1178
7.2.3.4.	Легенда. Вкладка Ошибки	1180
7.2.3.5.	Легенда. Вкладка Скомпилированный код	1180
7.2.3.6.	Легенда. Вкладка Элементы	1181
7.2.3.7.	Легенда. Вкладка Связи	1182
7.2.3.8.	Легенда. Вкладка Палитра	1182
7.2.3.9.	Легенда. Вкладка Действия	1183
7.2.3.10.	Легенда. Вкладка Условия	1183
7.2.3.11.	Легенда. Вкладка Надписи.....	1184
7.2.4.	Редактирование графических языков	1184
7.2.4.1.	Редактирование диаграмм в дереве	1184
7.2.4.2.	Масштабирование диаграмм	1185
7.2.4.3.	Прокрутка диаграмм	1185
7.2.4.4.	Миникарта	1185
7.2.4.5.	Связывание переменных на FBD- и LD-диаграммах	1186
7.2.5.	Компиляция программ	1186
7.3.	Редактор ST.....	1187
7.3.1.	Глобальные переменные	1189
7.3.2.	Вызов функций и функциональных блоков	1190
7.3.3.	Вызов программ объектов	1194
7.3.4.	Функции преобразования типа.....	1196
7.3.5.	ST-подсказчик.....	1197
7.3.6.	Поиск и замена в тексте ST.....	1197

7.3.7.	Экспорт и импорт программ ST	1197
7.3.8.	Синтаксис Master ST	1199
7.3.8.1.	Операторы Master ST	1199
7.3.8.1.1.	Оператор RETURN	1199
7.3.8.1.2.	Оператор IF	1200
7.3.8.1.3.	Оператор CASE	1201
7.3.8.1.4.	Оператор WHILE	1201
7.3.8.1.5.	Оператор REPEAT	1202
7.3.8.1.6.	Оператор FOR	1202
7.3.8.1.7.	Оператор EXIT	1203
7.3.8.1.8.	Операторы определения переменных	1203
7.3.8.1.9.	Оператор индексирования элементов массива	1207
7.3.8.2.	Инфиксные операторы	1207
7.3.8.3.	Особенности вычислений	1209
7.3.8.3.1.	Особенности битовых строк	1209
7.3.8.3.2.	Работа с отдельными битами	1209
7.3.8.4.	Константы Master ST	1210
7.3.8.4.1.	Числовые константы	1210
7.3.8.4.2.	Временные константы	1211
7.3.8.4.3.	Строковые константы	1212
7.3.8.5.	Лексическая структура	1214
7.3.8.5.1.	Идентификаторы	1215
7.3.8.5.2.	Разделители	1215
7.3.8.5.3.	Комментарии	1215
7.4.	Редактор FBD	1215
7.4.1.	Редактирование FBD-диаграммы	1216
7.4.2.	FBD-блок	1218
7.4.2.1.	Панель свойств FBD-блока	1219
7.4.3.	Слой диаграммы	1221

7.4.4.	Операции со связями FBD-диаграммы	1221
7.4.4.1.	Отображение связей	1221
7.4.4.2.	Маршрутные точки	1223
7.4.5.	Панель инструментов FBD-редактора	1224
7.4.5.1.	Панель Слои	1224
7.4.5.2.	Панель Режимы	1225
7.4.5.3.	Панель Сетка	1226
7.4.5.4.	Панель Выравнивание.....	1227
7.4.6.	Порядок исполнения FBD-блоков.....	1227
7.5.	Редактор LD	1230
7.5.1.	Слои LD-диаграммы.....	1230
7.5.2.	Редактирование LD-диаграммы	1230
7.5.2.1.	Размещение элементов алгоритма на LD-диаграмме.....	1231
7.5.2.2.	Перемещение элемента LD-алгоритма.....	1232
7.5.2.3.	Изменение типа контакта/катушки после размещения	1232
7.5.2.4.	Операции со связанными переменными	1233
7.5.3.	Панель инструментов LD-редактора	1234
7.5.4.	Палитра LD	1234
7.5.4.1.	Силовая линия.....	1234
7.5.4.2.	LD-блок	1235
7.5.4.3.	LD-ветвление	1235
7.5.4.4.	LD-терминатор	1236
7.5.4.5.	Другие элементы LD	1236
7.6.	Редактор SFC.....	1239
7.6.1.	Слои SFC-диаграммы	1239
7.6.2.	Редактирование SFC-диаграммы.....	1239
7.6.2.1.	Размещение элементов алгоритма на SFC-диаграмме	1240
7.6.3.	Панель инструментов SFC-редактора.....	1240
7.6.4.	Палитра SFC	1241

7.6.4.1.	Библиотека SFC	1241
7.6.4.1.1.	SFC-ветка	1241
7.6.4.1.2.	SFC-шаг	1242
7.6.4.1.3.	SFC-условие	1242
7.6.4.1.4.	SFC-переход	1242
7.6.4.1.5.	SFC-терминатор	1243
7.6.4.1.6.	SFC-ветвление выбора	1243
7.6.4.1.7.	Параллельное SFC-ветвление.....	1244
7.6.5.	Задание SFC-условий	1244
7.6.5.1.	Диалог редактирования SFC-условия.....	1245
7.6.6.	Задание SFC-действий	1246
7.6.6.1.	Диалог редактирования SFC-действия.....	1247
7.7.	Редактор C#	1250
7.7.1.	Легенда редактора C#	1255
7.7.2.	Программы C#. Системные требования.....	1257
7.7.3.	Программы C#. Примеры	1258
7.7.4.	Программы C#. Работа со сложными типами данных.....	1262
7.7.5.	Программы C#. Использование глобальных переменных	1263
7.7.6.	Программы C#.Подключение сторонних dll	1263
7.7.7.	Компиляция проекта с программой C#.....	1264
7.7.8.	Создание ФБ C#	1264
7.8.	Функции и ФБ библиотеки Стандартная	1265
7.8.1.	Библиотека Стандартные функции	1266
7.8.1.1.	Булевы функции	1266
7.8.1.1.1.	GetBit	1268
7.8.1.1.2.	SetBit	1268
7.8.1.2.	Математические функции	1269
7.8.1.3.	Функции сдвига	1272
7.8.1.4.	Функции сравнения	1273

7.8.1.5.	Тригонометрические функции.....	1275
7.8.1.6.	Функции работы со строками	1277
7.8.1.7.	Функции работы с датой-временем.....	1282
7.8.1.7.1.	CONCAT_DATE/CONCAT_DATE_LOCAL	1290
7.8.1.7.2.	CONCAT_TOD	1291
7.8.1.7.3.	CONCAT_DT/CONCAT_DT_LOCAL.....	1292
7.8.1.7.4.	SPLIT_DATE/SPLIT_DATE_LOCAL	1293
7.8.1.7.5.	SPLIT_TOD	1294
7.8.1.7.6.	SPLIT_DT/SPLIT_DT_LOCAL	1295
7.8.1.7.7.	GET_DT_TIMEZONE.....	1296
7.8.1.8.	Логарифмические функции	1297
7.8.1.9.	Функции выбора	1298
7.8.1.10.	Функции конвертации	1299
7.8.1.10.1.	LREAL_TO	1300
7.8.1.10.2.	REAL_TO	1302
7.8.1.10.3.	BYTE_TO	1304
7.8.1.10.4.	DWORD_TO.....	1305
7.8.1.10.5.	LWORD_TO	1307
7.8.1.10.6.	STRING_TO	1309
7.8.1.10.7.	SYSTEM_PARAM_TO	1312
7.8.1.10.8.	OBJECT_TO	1313
7.8.1.10.9.	DT_TO.....	1313
7.8.1.10.10.	DATE_TO	1315
7.8.1.11.	Функции работы с массивами	1317
7.8.1.12.	Служебные функции	1322
7.8.2.	Библиотека Стандартные ФБ	1324
7.8.2.1.	ФБ 'Детекторы фронтов'.....	1324
7.8.2.2.	ФБ 'Таймеры'	1324
7.8.2.3.	ФБ 'Триггеры'.....	1325

7.8.2.4.	ФБ 'Счетчики'	1326
7.9.	Важно знать	1328
7.9.1.	Ограничения использования функций и ФБ в задаче экрана	1328
7.9.2.	Работа с асинхронными ФБ.....	1330
8.	Создание окон для клиента визуализации.....	1333
8.1.	Окно	1336
8.1.1.	Структура окна	1338
8.1.2.	Свойства Окна	1339
8.2.	Редактор HMI	1341
8.2.1.	Панель инструментов редактора HMI.....	1342
8.2.1.1.	Панель Группа	1343
8.2.1.2.	Панель Просмотр	1344
8.2.1.3.	Панель Z-порядок	1345
8.2.1.4.	Панель Поворот	1346
8.2.1.5.	Панель Уравнять	1346
8.2.2.	HMI. Размещение элементов.....	1347
8.2.2.1.1.	Легенда в редакторе HMI	1348
8.2.2.1.2.	Легенда HMI. Вкладка Палитра	1349
8.2.2.1.3.	Легенда HMI. Вкладка Локальные	1351
8.2.2.1.4.	Легенда HMI. Вкладка Параметры	1351
8.2.2.1.5.	Легенда HMI. Вкладка Ошибки	1352
8.2.2.1.6.	Легенда HMI. Вкладка Элементы	1352
8.2.2.1.7.	Легенда HMI. Вкладка Связи	1353
8.2.2.1.8.	Легенда HMI. Вкладка События.....	1353
8.2.2.1.9.	Легенда HMI. Вкладка Медиа	1353
8.2.2.1.10.	Легенда HMI. Вкладка Триггеры	1354
8.2.2.2.	Размещение Тега, Канала, Объекта в окне.....	1354
8.2.2.3.	Размещение параметра в окне.....	1356
8.2.2.3.1.	Перетаскивание параметра в окно правой кнопкой мыши.....	1356

8.2.2.3.2.	Перетаскивание параметра в окно левой кнопкой мыши.....	1357
8.2.2.4.	Размещение программы в окне	1358
8.2.3.	HMI. Программирование в окнах	1359
8.2.3.1.	Динамизация.....	1359
8.2.3.1.1.	Конвертация значений.....	1360
8.2.3.1.1.1.	Конвертер значений.....	1362
8.2.3.1.1.2.	Неявная конвертация значений	1365
8.2.3.2.	HMI. Действия по условию	1365
8.2.3.3.	HMI. Действия по событию	1368
8.2.3.3.1.	HMI. События	1368
8.2.3.3.1.1.	Событие Клик по ячейке	1371
8.2.3.3.1.2.	Событие Клик по карте	1375
8.2.3.3.1.3.	Событие Клик по маркеру	1376
8.2.3.3.2.	HMI. Действия	1380
8.2.3.3.2.1.	Действие Вызвать Программу.....	1382
8.2.3.3.2.2.	Действие Открыть окно	1383
8.2.3.3.2.3.	Действие Закрывать окно.....	1384
8.2.3.3.2.4.	Действие Закрывать основное окно	1384
8.2.3.3.2.5.	Действие Открыть всплывающее окно.....	1384
8.2.3.3.2.6.	Действие Установить параметр.....	1391
8.2.3.3.2.7.	Действие Выдать импульс	1391
8.2.3.3.2.8.	Действие Показать сообщение	1391
8.2.3.3.2.9.	Действие Завершить сессию.....	1392
8.2.3.3.2.10.	Действие Сменить пользователя	1393
8.2.3.3.2.11.	Действие Вперед	1394
8.2.3.3.2.12.	Действие Назад.....	1394
8.2.3.3.2.13.	Действие Войти в полноэкранный режим	1394
8.2.3.3.2.14.	Действие Выйти из полноэкранного режима	1394
8.2.3.3.2.15.	Действие Открыть окно печати	1395

8.2.3.3.2.16.	Действие Открыть адрес.....	1396
8.2.3.3.2.17.	Действие Показать строку статуса.....	1397
8.2.3.3.2.18.	Действие Вызвать событие.....	1397
8.2.3.3.2.19.	Действие Вызвать метод.....	1398
8.2.4.	Клеммники в редакторе HMI.....	1398
8.2.5.	Менеджер ресурсов HMI.....	1400
8.2.6.	Диалоговое окно Выбор цвета.....	1404
8.2.7.	Создание цвета пользователя.....	1407
8.3.	Библиотека HMI.....	1410
8.3.1.	Категория Диалог.....	1410
8.3.1.1.	Текст.....	1411
8.3.1.2.	Текстовый ввод.....	1413
8.3.1.3.	Пароль.....	1415
8.3.1.4.	Инкремент.....	1417
8.3.1.5.	Кнопка.....	1420
8.3.1.6.	Кнопка с фиксацией.....	1422
8.3.1.7.	Радиокнопка.....	1427
8.3.1.8.	Флаг.....	1429
8.3.1.9.	Выпадающий список.....	1431
8.3.1.10.	Часы.....	1433
8.3.1.11.	ДатаВремя.....	1435
8.3.1.12.	Время.....	1440
8.3.1.13.	Интервал.....	1445
8.3.1.14.	Горизонтальный ползунок.....	1449
8.3.1.15.	Вертикальный ползунок.....	1452
8.3.1.16.	Горизонтальный прогресс.....	1453
8.3.1.17.	Вертикальный прогресс.....	1455
8.3.1.18.	Кнопка выбора цвета.....	1456
8.3.1.19.	Стрелочный прибор.....	1459

8.3.1.20.	Горизонтальный ползунок интервала.....	1462
8.3.1.21.	Вертикальный ползунок интервала	1465
8.3.2.	Категория Контролы	1466
8.3.2.1.	Тренд.....	1467
8.3.2.1.1.	Свойства Тренда	1468
8.3.2.1.2.	Тренд в клиенте визуализации	1478
8.3.2.1.2.1.	Тренд в клиенте визуализации при HMI v1.....	1478
8.3.2.1.2.2.	Тренд в клиенте визуализации при HMI v2.....	1481
	Добавление/удаление перьев тренда в клиенте визуализации	1486
	Табличное представление данных тренда	1487
	Экспорт и печать тренда	1488
	Смещение графиков по оси X.....	1489
	Изменение отображаемого интервала	1490
	Выделение области на тренде	1491
	Масштабирование по оси значений	1492
8.3.2.1.3.	Тренд. Права доступа	1494
8.3.2.1.4.	Методы для работы с трендом	1496
8.3.2.2.	Дискретный тренд	1500
8.3.2.2.1.	Свойства дискретного тренда	1504
8.3.2.3.	График XY.....	1514
8.3.2.3.1.	Свойства Графика XY	1517
8.3.2.3.2.	График XY. Права доступа	1525
8.3.2.4.	Дерево одиночного/множественного выбора	1526
8.3.2.4.1.	Источник данных для дерева одиночного/множественного выбора	1529
8.3.2.4.2.	Методы для работы деревом множественного выбора	1532
8.3.2.5.	Легенда	1534
8.3.2.5.1.	Легенда. Права доступа	1536
8.3.2.5.2.	Легенда в клиенте визуализации.....	1536
8.3.2.6.	Журнал и Архивный журнал	1541

8.3.2.6.1.	Настройка журнала	1544
8.3.2.6.1.1.	Окно настройки журнала. Столбцы таблицы	1546
8.3.2.6.1.2.	Окно настройки журнала. Столбцы легенды	1554
8.3.2.6.1.3.	Окно настройки журнала. Фильтры	1554
8.3.2.6.1.4.	Окно настройки журнала. Стиль строки	1558
8.3.2.6.1.5.	Окно настройки журнала. Стиль заголовка	1560
8.3.2.6.1.6.	Свойства Журнала	1560
8.3.2.6.1.7.	Методы для работы с журналом	1567
8.3.2.6.1.8.	Журнал. Права доступа	1568
8.3.2.6.2.	Работа с журналом в клиенте визуализации	1570
8.3.2.6.2.1.	Журнал в клиенте визуализации при HMI v1	1570
8.3.2.6.2.2.	Журнал в клиенте визуализации при HMI v2	1576
8.3.2.7.	Индикатор событий	1582
8.3.2.8.	Таблица данных	1586
8.3.2.8.1.	Редактор таблицы данных. Столбцы таблицы	1590
8.3.2.8.2.	Редактор таблицы данных. Стиль строки	1594
8.3.2.8.3.	Редактор таблицы данных. Стили ячейки	1596
8.3.2.8.4.	Редактор таблицы данных. Стиль заголовка	1599
8.3.2.8.5.	Таблица данных. Права доступа	1600
8.3.2.8.6.	Таблица данных в клиенте визуализации	1601
8.3.2.9.	Справочник	1603
8.3.2.9.1.	Редактор Справочника. Столбцы таблицы	1608
8.3.2.9.2.	Справочник. Права доступа	1612
8.3.2.9.3.	Пример работы со Справочником	1613
8.3.2.9.4.	Работа со справочником в клиенте визуализации	1631
8.3.2.10.	Карта	1634
8.3.2.10.1.	Редактор карты	1636
8.3.2.10.2.	Свойства Карты	1641
8.3.2.10.3.	Окна для отображения маркеров Карты	1646

8.3.2.10.4.	Источник данных для маркеров Карты	1649
8.3.2.10.5.	Всплывающее окно маркера карты	1654
8.3.2.10.6.	Яндекс.Карты	1661
8.3.2.10.7.	Карта в клиенте визуализации	1662
8.3.3.	Категория Панели	1663
8.3.3.1.	Общие инструменты панелей.....	1664
8.3.3.2.	Панель.....	1664
8.3.3.3.	Стековая панель.....	1666
8.3.3.4.	Панель док	1672
8.3.3.5.	Панель вкладок.....	1675
8.3.3.6.	Контейнер окна.....	1678
8.3.4.	Категория Примитивы	1684
8.3.4.1.	Линия	1684
8.3.4.2.	Граф.....	1686
8.3.4.3.	Ломаная линия.....	1690
8.3.4.4.	Полигон.....	1693
8.3.4.5.	Овал, Прямоугольник, Треугольник	1696
8.3.4.6.	Крест	1700
8.3.4.7.	Стрелка	1701
8.3.4.8.	Картинка	1701
8.3.4.9.	Графический элемент Видео	1703
8.3.5.	Темы библиотеки HMI	1706
8.3.6.	Категория BaseObjects	1706
8.3.6.1.	BaseObjects. Элементы функциональных схем	1708
8.3.7.	Категория Images	1708
8.4.	HMI. Свойства	1709
8.4.1.	Свойство Формат значений.....	1709
8.4.2.	Свойство Маска ввода	1714
8.4.3.	Категория свойств Экран	1716

8.4.4.	Категория свойств Рамка/Линия	1719
8.4.5.	Категория свойств Внешний вид	1719
8.4.6.	Категория свойств Расположение	1723
8.4.7.	Категория свойств Текст	1725
8.4.8.	Категория свойств Трансформация	1727
8.4.9.	Категория свойств Слой	1728
8.5.	Рекомендации по созданию окон	1729
8.5.1.	Создание произвольных фигур.....	1730
8.5.2.	Стартовое окно.....	1731
8.5.3.	Кеширование окон.....	1732
8.5.4.	Особенности создания окон в библиотеке	1737
8.5.5.	Особенности работы с элементами Кнопка и Кнопка с фиксацией	1741
8.5.6.	Массивы структур в окнах	1746
8.5.7.	Создание пользовательской библиотеки примитивов	1755
8.5.8.	Работа с веб-камерой	1759
8.5.9.	Импорт SVG в среду разработки MasterSCADA 4D	1760
8.5.9.1.	Пример импорта SVG.....	1766
8.5.10.	Разработка SVG файла с изображением	1769
8.5.11.	Использование тем.....	1772
9.	Создание отчетов.....	1777
9.1.	Пример создания отчета	1782
9.2.	Редактор отчетов	1791
9.2.1.	Страница (Page).....	1792
9.2.2.	Панель форматирования.....	1800
9.2.3.	Инструментарий.....	1804
9.2.3.1.	Типы секций	1806
9.2.3.2.	Компоненты	1813
9.2.3.3.	Примитивы	1814
9.2.4.	Панели	1815

9.2.4.1.	Словарь данных	1815
9.2.4.1.1.	Меню Новый элемент	1820
9.2.4.2.	Дерево отчетов	1821
9.2.4.3.	Свойства.....	1822
9.2.5.	Контекстное меню	1826
9.2.6.	Статусная панель.....	1828
9.2.6.1.	Инспектор отчета	1829
9.2.7.	"Горячие клавиши" в редакторе отчетов	1830
9.2.8.	Расположение компонентов в отчете	1833
9.2.9.	Предварительный просмотр отчета в дизайнера	1833
9.3.	Источники данных, переменные и функции	1834
9.3.1.	Источники данных	1835
9.3.1.1.	Проект MasterSCADA	1838
9.3.1.1.1.	Создание источника Проект [Текущий проект].....	1839
9.3.1.1.2.	Архив MasterSCADA	1841
9.3.1.1.2.1.	Свойства переменных Архива данных	1846
	Обработка Интеграл	1850
	Обработка Интегральное среднее	1852
	Обработка плохих признаков качества.....	1852
	Обработка плохих признаков при вычислении интеграла	1854
9.3.1.1.2.2.	Способы формирования отчетов	1859
	Формирование отчетов по изменению значений.....	1859
	Формирование отчетов по периодической выборке значений	1861
	Итоговый способ формирования отчета.....	1864
9.3.1.1.3.	Массив структур в отчете	1867
9.3.1.2.	Прочие источники.....	1868
9.3.1.2.1.	Источник данных СУБД	1869
9.3.1.2.1.1.	Пример работы с БД MS SQL	1874
9.3.1.2.1.2.	Запросы	1882

Параметры	1885
Конструктор (Построитель) запросов.....	1891
9.3.1.2.2. Создание и редактирование колонки данных.....	1894
9.3.1.2.2.1. Рассчитываемая колонка данных	1895
9.3.1.2.3. Данные на основе других источников данных	1898
9.3.1.2.4. Данные из кросс-таблиц	1900
9.3.1.2.5. Пользовательские источники данных	1904
9.3.1.2.6. Получить только используемые данные	1906
9.3.2. Переменные	1908
9.3.2.1. Локальные переменные	1909
9.3.2.1.1. Создание переменных.....	1909
9.3.2.1.2. Панель Запросить у пользователя	1913
9.3.2.1.2.1. Окно Элементы.....	1914
9.3.2.1.2.2. Зависимые переменные	1916
9.3.2.1.3. Вид Значение	1923
9.3.2.1.4. Вид Значение с null.....	1924
9.3.2.1.5. Вид Список	1925
9.3.2.1.6. Вид Диапазон.....	1926
9.3.2.1.7. Три режима работы переменной	1928
9.3.3. Системные переменные.....	1932
9.3.4. Функции и расчет итогов	1936
9.3.4.1. Итоги	1957
9.3.4.1.1. Итоги связанные с бэндами.....	1957
9.3.4.1.2. Итоги не связанные с бэндами.....	1967
9.4. Формирование отчета в исполнительной системе	1977
10. Работа в режиме исполнения	1982
10.1. Работа со средой исполнения	1982
10.1.1. Рабочая папка среды исполнения	1982
10.1.2. Подготовка проекта к запуску в среде исполнения	1986

10.1.3.	Автоматическая загрузка проекта	1989
10.1.3.1.	Использование пароля для загрузки проекта	1992
10.1.4.	Ручная загрузка проекта в среду исполнения	1992
10.1.5.	Восстановление работы узла	1995
10.1.6.	MasterSCADA 4D Monitor	1995
10.1.6.1.	Окно настройки среды исполнения	2001
10.1.6.2.	Работа с процессом mpls.....	2006
10.1.7.	Запуск исполнительной системы.....	2012
10.1.8.	Запуск нескольких узлов на одном компьютере	2013
10.1.9.	Обновление проекта в новой версии среды исполнения	2014
10.1.10.	Выбор опций среды исполнения.....	2015
10.1.11.	Коды ошибок среды исполнения	2017
10.1.12.	Панель узла	2018
10.1.12.1.	Вкладка управления узлом	2019
10.1.12.2.	Вкладка Конфигурация.....	2021
10.1.13.	Работа с файлом данных для горячего рестарта.....	2023
10.1.14.	Создание бекапа через командную строку	2024
10.2.	Работа с клиентом визуализации.....	2025
10.2.1.	Выбор версии HMI	2028
10.2.2.	Окно Список ошибок	2029
10.2.3.	MasterSCADA 4D Client Monitor.....	2030
10.2.4.	Особенности первой автоматической загрузки проекта.....	2033
10.2.5.	Окно авторизации пользователей.....	2034
10.2.6.	Задание принтера и пути к сохраняемым файлам по умолчанию	2036
10.2.7.	Горячие клавиши в клиенте визуализации.....	2038
10.2.8.	Многомониторный режим.....	2039
10.2.9.	Очистка кеша MasterSCADA 4D Client	2040
10.3.	Отладка проекта	2040
10.3.1.	Режимы отладки	2040

10.3.1.1.	Непрерывная отладка	2041
10.3.1.2.	Отладка с шагом 1 цикл	2042
10.3.2.	Как получить диагностическую информацию среды исполнения	2042
10.3.3.	Подключение среды разработки к исполняемому узлу.....	2049
10.3.4.	Эмуляция устройства	2051
10.3.5.	Имитация значений параметров	2052
10.3.6.	Сообщения об ошибках.....	2053
10.4.	Порядок исполнения и передачи данных	2057
10.5.	Ошибки ключа защиты	2061
11.	Библиотека OSCAT	2062
11.1.1.	OSCAT.Функции.Математика	2063
11.1.1.1.	D_TRUNC (OSCAT)	2063
11.1.1.2.	DEC1 (OSCAT)	2063
11.1.1.3.	EVEN (OSCAT)	2064
11.1.1.4.	FLOOR (OSCAT).....	2064
11.1.1.5.	FLOOR2 (OSCAT).....	2064
11.1.1.6.	FRACT (OSCAT)	2064
11.1.1.7.	INC (OSCAT)	2065
11.1.1.8.	INC1 (OSCAT)	2065
11.1.1.9.	INC2 (OSCAT)	2065
11.1.1.10.	MID3 (OSCAT).....	2066
11.1.1.11.	MODR (OSCAT)	2066
11.1.1.12.	MODR_old (OSCAT)	2067
11.1.1.13.	RAD (OSCAT).....	2067
11.1.1.14.	RDM (OSCAT).....	2068
11.1.1.15.	SIGN_R (OSCAT).....	2068
11.1.2.	OSCAT.Функции.Арифметические функции	2068
11.1.2.1.	F_LIN (OSCAT).....	2068
11.1.2.2.	F_LIN2 (OSCAT).....	2069

11.1.2.3.	F_POLY (OSCAT).....	2069
11.1.2.4.	F_POWER (OSCAT).....	2069
11.1.2.5.	F_QUAD (OSCAT).....	2070
11.1.2.6.	FRMP_B (OSCAT).....	2070
11.1.2.7.	LINEAR_INT (OSCAT).....	2071
11.1.2.8.	POLYNOM_INT (OSCAT).....	2072
11.1.3.	OSCAT.Функции.Геометрические функции.....	2073
11.1.3.1.	CIRCLE_A (OSCAT).....	2073
11.1.3.2.	CIRCLE_C (OSCAT).....	2073
11.1.3.3.	CIRCLE_SEG (OSCAT).....	2074
11.1.3.4.	CONE_V (OSCAT).....	2074
11.1.3.5.	ELLIPSE_A (OSCAT).....	2075
11.1.3.6.	ELLIPSE_C (OSCAT).....	2075
11.1.3.7.	SPHERE_V (OSCAT).....	2075
11.1.3.8.	TRIANGLE_A (OSCAT).....	2076
11.1.4.	OSCAT.Функции.Дата и время.....	2076
11.1.4.1.	DAY_OF_WEEK (OSCAT).....	2076
11.1.4.2.	DAY_OF_YEAR (OSCAT).....	2077
11.1.4.3.	LEAP_OF_DATE (OSCAT).....	2077
11.1.4.4.	MONTH_OF_DATE (OSCAT).....	2077
11.1.4.5.	YEAR_OF_DATE (OSCAT).....	2077
11.1.5.	OSCAT.Функции.Логические модули.....	2078
11.1.5.1.	BCDC_TO_INT (OSCAT).....	2078
11.1.5.2.	BIT_COUNT (OSCAT).....	2078
11.1.5.3.	BIT_LOAD_B (OSCAT).....	2078
11.1.5.4.	BIT_LOAD_B2 (OSCAT).....	2079
11.1.5.5.	BIT_LOAD_DW (OSCAT).....	2079
11.1.5.6.	BIT_LOAD_DW2 (OSCAT).....	2080
11.1.5.7.	BIT_LOAD_W (OSCAT).....	2080

11.1.5.8.	BIT_LOAD_W2 (OSCAT)	2080
11.1.5.9.	BIT_OF_DWORD (OSCAT)	2080
11.1.5.10.	BIT_TOGGLE_B (OSCAT)	2081
11.1.5.11.	BIT_TOGGLE_DW (OSCAT).....	2081
11.1.5.12.	BIT_TOGGLE_W (OSCAT)	2081
11.1.5.13.	BYTE_OF_BIT (OSCAT)	2082
11.1.5.14.	BYTE_OF_DWORD (OSCAT)	2082
11.1.5.15.	BYTE_TO_GRAY (OSCAT)	2082
11.1.5.16.	CHECK_PARITY (OSCAT).....	2083
11.1.5.17.	DWORD_OF_BYTE (OSCAT)	2083
11.1.5.18.	DWORD_OF_WORD (OSCAT)	2083
11.1.5.19.	GRAY_TO_BYTE (OSCAT)	2084
11.1.5.20.	INT_TO_BCDC (OSCAT)	2084
11.1.5.21.	MUX_2 (OSCAT).....	2084
11.1.5.22.	MUX_4 (OSCAT).....	2085
11.1.5.23.	PARITY (OSCAT)	2085
11.1.5.24.	REFLECT (OSCAT)	2085
11.1.5.25.	REVERSE (OSCAT).....	2086
11.1.5.26.	SHL1 (OSCAT).....	2086
11.1.5.27.	SHR1 (OSCAT)	2086
11.1.5.28.	SWAP_BYTE (OSCAT)	2086
11.1.5.29.	SWAP_BYTE2 (OSCAT)	2087
11.1.5.30.	WORD_OF_BYTE (OSCAT).....	2087
11.1.5.31.	WORD_OF_DWORD (OSCAT)	2087
11.1.6.	OSCAT.Функции.Преобразования.....	2087
11.1.6.1.	BFT_TO_MS (OSCAT)	2088
11.1.6.2.	C_TO_F (OSCAT)	2088
11.1.6.3.	C_TO_K (OSCAT)	2088
11.1.6.4.	F_TO_C (OSCAT)	2088

11.1.6.5.	F_TO_OM (OSCAT)	2089
11.1.6.6.	F_TO_PT (OSCAT)	2089
11.1.6.7.	GEO_TO_DEG (OSCAT)	2089
11.1.6.8.	K_TO_C (OSCAT)	2089
11.1.6.9.	KMH_TO_MS (OSCAT)	2090
11.1.6.10.	MS_TO_BFT (OSCAT)	2090
11.1.6.11.	MS_TO_KMH (OSCAT)	2090
11.1.6.12.	OM_TO_F (OSCAT)	2091
11.1.6.13.	PT_TO_F (OSCAT)	2091
11.1.7.	OSCAT.Функции.Обработка сигналов	2091
11.1.7.1.	AIN (OSCAT)	2091
11.1.7.2.	AOUT (OSCAT)	2092
11.1.7.3.	AOUT1 (OSCAT)	2094
11.1.7.4.	BYTE_TO_RANGE (OSCAT)	2095
11.1.7.5.	MIX (OSCAT)	2095
11.1.7.6.	MUX_R2 (OSCAT)	2096
11.1.7.7.	MUX_R4 (OSCAT)	2096
11.1.7.8.	OFFSET (OSCAT)	2096
11.1.7.9.	OFFSET2 (OSCAT)	2097
11.1.7.10.	OVERRIDE (OSCAT)	2098
11.1.7.11.	RANGE_TO_BYTE (OSCAT)	2098
11.1.7.12.	RANGE_TO_WORD (OSCAT)	2098
11.1.7.13.	SCALE (OSCAT)	2099
11.1.7.14.	SCALE_B (OSCAT)	2099
11.1.7.15.	SCALE_B2 (OSCAT)	2100
11.1.7.16.	SCALE_B4 (OSCAT)	2101
11.1.7.17.	SCALE_B8 (OSCAT)	2103
11.1.7.18.	SCALE_D (OSCAT)	2104
11.1.7.19.	SCALE_R (OSCAT)	2105

11.1.7.20.	SCALE_X2 (OSCAT).....	2105
11.1.7.21.	SCALE_X4 (OSCAT).....	2106
11.1.7.22.	SCALE_X8 (OSCAT).....	2107
11.1.7.23.	STAIR (OSCAT).....	2108
11.1.7.24.	WORD_TO_RANGE (OSCAT).....	2109
11.1.8.	OSCAT.Функции.Датчики	2109
11.1.8.1.	MULTI_IN (OSCAT).....	2110
11.1.8.2.	RES_NI (OSCAT)	2111
11.1.8.3.	RES_NTC (OSCAT)	2111
11.1.8.4.	RES_PT (OSCAT).....	2112
11.1.8.5.	RES_SI (OSCAT)	2113
11.1.8.6.	SENSOR_INT (OSCAT)	2113
11.1.8.7.	TEMP_NI (OSCAT)	2114
11.1.8.8.	TEMP_NTC (OSCAT).....	2115
11.1.8.9.	TEMP_PT (OSCAT)	2115
11.1.8.10.	TEMP_SI (OSCAT).....	2116
11.1.9.	OSCAT.Функции.Модули измерения	2116
11.1.9.1.	T_PLC_MS (OSCAT).....	2116
11.1.9.2.	T_PLC_US (OSCAT).....	2117
11.1.10.	OSCAT.Функции.Модули регулирования	2118
11.1.10.1.	BAND_B (OSCAT)	2118
11.1.10.2.	CTRL_IN (OSCAT)	2119
11.1.10.3.	DEAD_BAND (OSCAT)	2119
11.1.10.4.	DEAD_ZONE (OSCAT).....	2120
11.1.11.	OSCAT.Функции.Управление устройствами	2121
11.1.11.1.	MANUAL (OSCAT)	2121
11.1.12.	OSCAT.ФБ.Арифметические функции	2122
11.1.12.1.	FT_AVG (OSCAT)	2122
11.1.12.2.	FT_MIN_MAX (OSCAT)	2122

11.1.12.3.	FT_RMP (OSCAT)	2123
11.1.13.	OSCAT.ФБ.Модули памяти.....	2124
11.1.13.1.	FIFO_16 (OSCAT)	2124
11.1.13.2.	FIFO_32 (OSCAT)	2125
11.1.13.3.	STACK_16 (OSCAT).....	2125
11.1.13.4.	STACK_32 (OSCAT).....	2126
11.1.14.	OSCAT.ФБ.Генераторы импульсов	2126
11.1.14.1.	A_TRIG (OSCAT)	2126
11.1.14.2.	B_TRIG (OSCAT)	2127
11.1.14.3.	CLICK_CNT (OSCAT)	2127
11.1.14.4.	CLICK_DEC (OSCAT)	2128
11.1.14.5.	CLK_DIV (OSCAT)	2129
11.1.14.6.	CLK_N (OSCAT)	2131
11.1.14.7.	CLK_PRG (OSCAT)	2131
11.1.14.8.	CLK_PULSE (OSCAT).....	2131
11.1.14.9.	CYCLE_4 (OSCAT)	2133
11.1.14.10.	D_TRIG (OSCAT).....	2134
11.1.14.11.	GEN_BIT (OSCAT).....	2134
11.1.14.12.	GEN_SQ (OSCAT)	2136
11.1.14.13.	SCHEDULER (OSCAT).....	2137
11.1.14.14.	SCHEDULER_2 (OSCAT).....	2138
11.1.14.15.	SEQUENCE_4 (OSCAT)	2140
11.1.14.16.	SEQUENCE_64 (OSCAT)	2145
11.1.14.17.	SEQUENCE_8 (OSCAT)	2146
11.1.14.18.	TMAX (OSCAT)	2148
11.1.14.19.	TMIN (OSCAT).....	2149
11.1.14.20.	TOF_1 (OSCAT)	2149
11.1.14.21.	TONOF (OSCAT)	2150
11.1.14.22.	TP_1 (OSCAT).....	2150

11.1.14.23.	TP_1D (OSCAT)	2151
11.1.14.24.	TP_X (OSCAT).....	2152
11.1.15.	OSCAT.ФБ.Логические модули	2153
11.1.15.1.	BYTE_TO_BITS (OSCAT)	2153
11.1.15.2.	DEC_2 (OSCAT)	2153
11.1.15.3.	DEC_4 (OSCAT)	2154
11.1.15.4.	DEC_8 (OSCAT)	2154
11.1.15.5.	MATRIX (OSCAT).....	2155
11.1.16.	OSCAT.ФБ.Триггеры.....	2158
11.1.16.1.	COUNT_BR (OSCAT).....	2158
11.1.16.2.	COUNT_BR_plcd	2159
11.1.16.3.	COUNT_DR (OSCAT)	2159
11.1.16.4.	COUNT_DR_plcd.....	2160
11.1.16.5.	FF_D2E (OSCAT).....	2160
11.1.16.6.	FF_D4E (OSCAT).....	2161
11.1.16.7.	FF_DRE (OSCAT).....	2161
11.1.16.8.	FF_JKE (OSCAT).....	2162
11.1.16.9.	FF_RSE (OSCAT)	2163
11.1.16.10.	LTCH (OSCAT).....	2163
11.1.16.11.	LATCH4 (OSCAT)	2164
11.1.16.12.	SELECT_8 (OSCAT)	2164
11.1.16.13.	SHR_4E (OSCAT)	2165
11.1.16.14.	SHR_4UDE (OSCAT)	2165
11.1.16.15.	SHR_8PLE (OSCAT).....	2166
11.1.16.16.	SHR_8UDE (OSCAT)	2167
11.1.16.17.	STORE_8 (OSCAT)	2168
11.1.16.18.	TOGGLE (OSCAT).....	2168
11.1.17.	OSCAT.ФБ.Генераторы сигналов	2169
11.1.17.1.	_RMP_B (OSCAT).....	2169

11.1.17.2.	_RMP_NEXT (OSCAT).....	2171
11.1.17.3.	_RMP_W (OSCAT).....	2172
11.1.17.4.	GEN_PULSE (OSCAT).....	2172
11.1.17.5.	GEN_PW2 (OSCAT)	2173
11.1.17.6.	GEN_RDM (OSCAT).....	2174
11.1.17.7.	GEN_RDT (OSCAT)	2175
11.1.17.8.	GEN_RMP (OSCAT)	2175
11.1.17.9.	GEN_SIN (OSCAT)	2177
11.1.17.10.	GEN_SQR (OSCAT)	2178
11.1.17.11.	PWM_DC (OSCAT)	2180
11.1.17.12.	PWM_PW (OSCAT)	2181
11.1.17.13.	RMP_B (OSCAT).....	2182
11.1.17.14.	RMP_SOFT (OSCAT).....	2184
11.1.17.15.	RMP_W (OSCAT).....	2186
11.1.18.	OSCAT.ФБ.Обработка сигналов	2186
11.1.18.1.	AIN1 (OSCAT).....	2186
11.1.18.2.	DELAY (OSCAT)	2189
11.1.18.3.	DELAY_4 (OSCAT)	2191
11.1.18.4.	FADE (OSCAT)	2192
11.1.18.5.	FILTER_DW (OSCAT)	2194
11.1.18.6.	FILTER_I (OSCAT)	2195
11.1.18.7.	FILTER_MAV_DW (OSCAT)	2195
11.1.18.8.	FILTER_MAV_W (OSCAT).....	2196
11.1.18.9.	FILTER_W (OSCAT).....	2196
11.1.18.10.	FILTER_WAV (OSCAT)	2197
11.1.18.11.	SEL2_OF_3 (OSCAT).....	2198
11.1.18.12.	SEL2_OF_3B (OSCAT).....	2199
11.1.18.13.	SH (OSCAT)	2200
11.1.18.14.	SH_1 (OSCAT)	2201

11.1.18.15.	SH_2 (OSCAT)	2202
11.1.18.16.	SH_T (OSCAT).....	2203
11.1.18.17.	STAIR2 (OSCAT).....	2204
11.1.18.18.	TREND (OSCAT).....	2206
11.1.18.19.	TREND_DW (OSCAT).....	2206
11.1.19.	OSCAT.ФБ.Модули измерения	2207
11.1.19.1.	ALARM_2 (OSCAT)	2207
11.1.19.2.	BAR_GRAPH (OSCAT).....	2208
11.1.19.3.	CALIBRATE (OSCAT)	2212
11.1.19.4.	CALIBRATE_mplc	2214
11.1.19.5.	CYCLE_TIME (OSCAT).....	2215
11.1.19.6.	DT_SIMU (OSCAT).....	2216
11.1.19.7.	FLOW_METER (OSCAT).....	2216
11.1.19.8.	M_D (OSCAT)	2218
11.1.19.9.	M_T (OSCAT)	2219
11.1.19.10.	M_TX (OSCAT)	2220
11.1.19.11.	ONTIME (OSCAT)	2221
11.1.19.12.	TC_MS (OSCAT).....	2221
11.1.19.13.	TC_S (OSCAT)	2222
11.1.19.14.	TC_US (OSCAT)	2222
11.1.19.15.	METER_STAT (OSCAT)	2222
11.1.20.	OSCAT.ФБ.Преобразования	2224
11.1.20.1.	ASTRO (OSCAT).....	2224
11.1.20.2.	ENERGY (OSCAT).....	2224
11.1.20.3.	LENGTH (OSCAT).....	2225
11.1.20.4.	PRESSURE (OSCAT)	2226
11.1.20.5.	SPEED (OSCAT).....	2228
11.1.20.6.	TEMPERATURE (OSCAT)	2228
11.1.21.	OSCAT.ФБ.Модули регулирования	2229

11.1.21.1.	CONTROL_SET1 (OSCAT)	2229
11.1.21.2.	CONTROL_SET2 (OSCAT)	2232
11.1.21.3.	CTRL_OUT (OSCAT)	2234
11.1.21.4.	CTRL_PI (OSCAT)	2235
11.1.21.5.	CTRL_PID (OSCAT)	2237
11.1.21.6.	CTRL_PWM (OSCAT)	2240
11.1.21.7.	DEAD_BAND_A (OSCAT)	2241
11.1.21.8.	DEAD_ZONE2 (OSCAT)	2242
11.1.21.9.	FT_DERIV (OSCAT)	2243
11.1.21.10.	FT_IMP (OSCAT)	2244
11.1.21.11.	FT_INT (OSCAT)	2245
11.1.21.12.	FT_PD (OSCAT)	2247
11.1.21.13.	FT_PDT1 (OSCAT)	2248
11.1.21.14.	FT_PI (OSCAT)	2249
11.1.21.15.	FT_PID (OSCAT)	2250
11.1.21.16.	FT_PIDW (OSCAT)	2252
11.1.21.17.	FT_PIDWL (OSCAT)	2253
11.1.21.18.	FT_PIW (OSCAT)	2254
11.1.21.19.	FT_PIWL (OSCAT)	2256
11.1.21.20.	FT_PT1 (OSCAT)	2257
11.1.21.21.	FT_PT2 (OSCAT)	2258
11.1.21.22.	FT_TN16 (OSCAT)	2259
11.1.21.23.	FT_TN64 (OSCAT)	2260
11.1.21.24.	FT_TN8 (OSCAT)	2260
11.1.21.25.	HYST (OSCAT)	2261
11.1.21.26.	HYST_1 (OSCAT)	2262
11.1.21.27.	HYST_2 (OSCAT)	2262
11.1.21.28.	HYST_3 (OSCAT)	2263
11.1.21.29.	INTEGRATE (OSCAT)	2264

11.1.22.	OSCAT.ФБ.Управление устройствами	2265
11.1.22.1.	DRIVER_1 (OSCAT)	2265
11.1.22.2.	DRIVER_4 (OSCAT)	2266
11.1.22.3.	DRIVER_4C (OSCAT)	2266
11.1.22.4.	FLOW_CONTROL (OSCAT)	2268
11.1.22.5.	FT_PROFILE (OSCAT)	2269
11.1.22.6.	INC_DEC (OSCAT)	2270
11.1.22.7.	INTERLOCK_4 (OSCAT)	2272
11.1.22.8.	INTERLOCK (OSCAT)	2273
11.1.22.9.	MANUAL_1 (OSCAT)	2275
11.1.22.10.	MANUAL_2 (OSCAT)	2276
11.1.22.11.	MANUAL_4 (OSCAT)	2276
11.1.22.12.	PARSET (OSCAT)	2277
11.1.22.13.	PARSET2 (OSCAT)	2279
11.1.22.14.	SIGNAL (OSCAT)	2280
11.1.22.15.	SIGNAL_4 (OSCAT)	2280
11.1.22.16.	SRAMP (OSCAT)	2282
11.1.22.17.	TUNE (OSCAT)	2283
11.1.22.18.	TUNE2 (OSCAT)	2285

1. ИНСТАЛЛЯЦИЯ

1.1. СОСТАВ MASTERSCADА 4D

SCADA-система **MasterSCADA 4D** включает в себя следующие части:

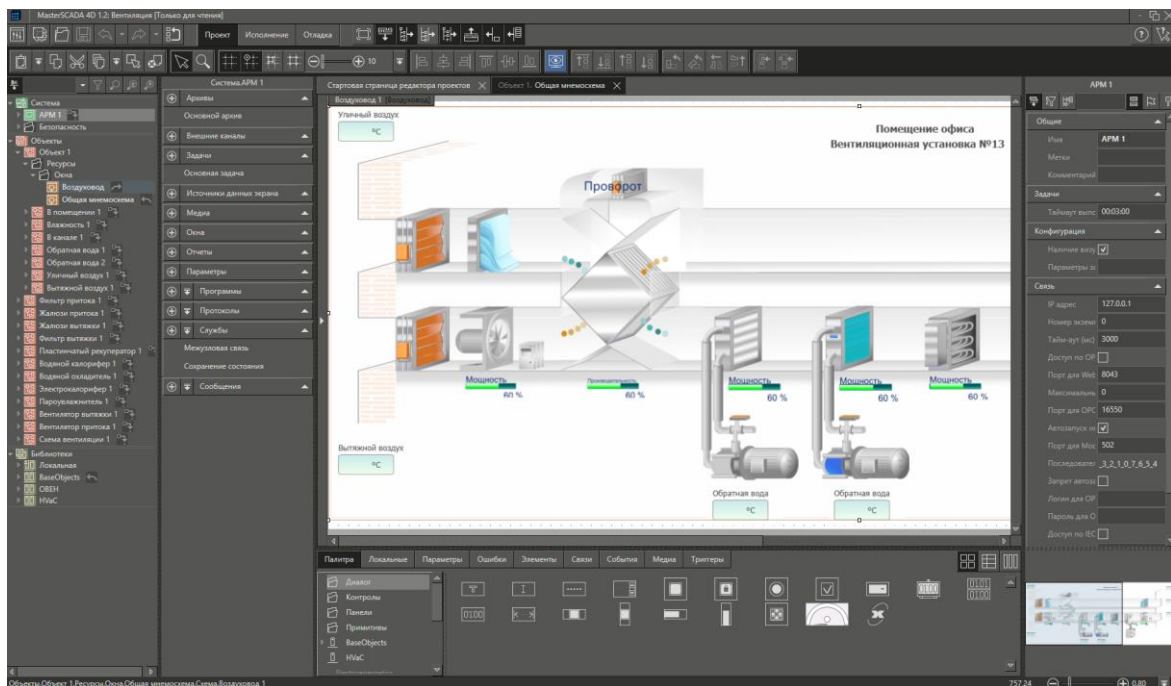
- инструментальная система/среда разработки;
- исполнительная система, состоящая из набора модулей;
- клиент визуализации.

В инструментальной системе создается проект, который впоследствии запускается в реальном времени под управлением исполнительных модулей, доступ к графической части осуществляется при помощи клиента визуализации.

Инструментальная система

Инструментальная среда является приложением для Windows.

Общий вид инструментальной системы:



Проект создается в рамках единой инструментальной системы. Никаких других инструментов или редакторов не требуется. Данные, введенные в системе один раз, становятся доступны для любого элемента проекта.

Работать в инструментальной среде легко и удобно: основным инструментом для создания элементов проекта является контекстное меню, а для настройки связей между элементами — механизм drag-and-drop. Среда имеет большой набор инструментов для тиражирования готовых решений, автоматизации рутинных операций, а также для online- и offline-отладки.

Исполнительная система

В исполнительной системе происходит исполнение созданного проекта.

В рамках одного проекта **MasterSCADA 4D** позволяет программировать:

- программируемые логические контроллеры (ПЛК);
- локальные панели управления (HMI-панели);
- рабочие места операторов (АРМ);
- архивные и технологические серверы;
- облачные сервисы.

Для этих элементов систем управления в **MasterSCADA 4D** имеются специализированные исполнительные системы (runtime). Они устанавливаются на данные устройства, и по команде инструментальной системы загружают и запускают на исполнение предназначенные для них части проекта.

Важно! Запуск проекта под управлением исполнительной системы может быть совершен автоматически и без использования инструментальной системы при старте данного устройства.

Надо отметить, что в **MasterSCADA 4D** имеются исполнительные системы для различных операционных систем, таких как Windows, Linux, QNX, Android, Эльбрус. В зависимости от количества получаемых и передаваемых данных, от функций, заложенных в проект, выбирается коммерческая версия среды разработки. Подробнее смотрите в разделе - **Выбор опций среды исполнения**

Клиент визуализации

Для представления графической информации в **MasterSCADA 4D** используется стандарт HTML5. Для его реализации в состав серверной части большинства исполнительных модулей входит WEB-сервер. Он формирует страницы формата HTML5, которые отображаются в клиенте визуализации **MasterSCADA 4D**.

Следует отметить, что стандарт HTML5 поддерживается всеми современными браузерами, в том числе и нашего производства. Это сильно расширяет доступность отображения графической информации в **MasterSCADA 4D**, т.к. в качестве клиента визуализации можно использовать любое устройство, имеющее в своем составе современный браузер. Это могут быть не только компьютеры, но и операторские панели, смартфоны, планшеты. С любого такого устройства можно подключиться к серверу **MasterSCADA 4D** и получать доступ к той же информации, которая предоставлена оператору на локальном АРМ. Естественно, это возможно только если был открыт доступ к данной информации или управлению.

1.2. СИСТЕМНЫЕ ТРЕБОВАНИЯ СРЕДЫ РАЗРАБОТКИ

Минимальные системные требования:

- операционная система (ОС) – Windows Server 2008 R2 SP1 x64 или Windows 7 SP1 x64 (операционная система должна поддерживать русский язык);
- процессор – Intel® Core™ i3, 2.3 ГГц;
- ОЗУ – 8 Гб;
- дисплей – 1280x1024
- жесткий диск - 10 Гб свободного пространства
- клавиатура, мышь.

Важно! При соблюдении минимальных системных требований среда разработки MasterSCADA 4D может быть установлена, но при разработке проектов может наблюдаться торможение при переключении между вкладками, при работе с деревьями и панелями редактора. Время выполнения длительных операций, например, конвертации проектов, может быть значительным

Рекомендуемые системные требования:

- ОС – Windows 10 x64 или старше;
- процессор – современный многоядерный, не ниже Intel® Core™ i5, 3.4 ГГц
- ОЗУ – не менее 16 Гб;
- дисплей – 1920x1080;
- жесткий диск - SSD
- свободное место на диске – 100 Гб
- клавиатура, мышь.

Важно! Поддерживаются только 64-битные ОС.

1.3. СИСТЕМНЫЕ ТРЕБОВАНИЯ ДЛЯ СРЕДЫ ИСПОЛНЕНИЯ

Среда исполнения MasterSCADA 4D RT имеет модульную структуру.

Состав модулей:

Название модуля	Назначение
Сервер обработки данных	Получение данных по стандартным протоколам: Modbus, DCON и др.

Название модуля	Назначение
OPC UA клиент/сервер	Возможность получения и передачи данных по кроссплатформенному стандарту OPC UA
Сервер архивов	Возможность ведения архивов. В случае ведения архивов на контроллерах необходимо предусмотреть место для хранения архива. Как правило, для этого используется внешняя карта памяти
веб-сервер	Используется в случаях, если необходимо подключение клиентов визуализации , удаленных или локальных
Клиент визуализации	Клиент, который подключается к веб-серверу среды исполнения. Может быть как производства компании "МПС Софт", так и другого производителя, например, браузер Chrome старше v.66 и др.

При комплектации среды исполнения разрешено подключать и отключать те или иные модули в зависимости от потребностей проекта и возможностей оборудования. Каждый модуль требует определенных ресурсов устройства. Минимальные требования для установки среды исполнения **MasterSCADA 4D RT**:

Название модуля	RAM (MB)	Flash (MB)	Процессор
Сервер обработки данных	8	4	200 MHz
OPC UA клиент/сервер	16	8	200 MHz
Сервер архивов	32	32+	200 MHz
веб-сервер	64	64	200 MHz
Клиент визуализации	256	256	800 MHz Cortex A8

Исполнительная система может работать практически на любых типах процессоров, мощность которых удовлетворяет приведенным требованиям. Необходимо, чтобы в наличии имелись API для разработки программного обеспечения для конкретного оборудования.

В зависимости от многих факторов, — например: сложности проекта, количества получаемых данных, особенностей архивирования, требуемого быстродействия, наличия большого количества динамически изменяемых элементов в окне — требования могут быть повышены. Данные характеристики используются для определения возможности портирования **MasterSCADA 4D RT** на новый тип контроллеров, панелей операторов и других форм-факторов.

Поддерживаемые операционные системы:

Название ОС	Минимальная версия	Примечания
Windows	Windows 7	
Windows Embedded Standart	Windows 7	
Linux (в том числе AstraLinux, Debian, AltLinux и др.)	3.18	Поддержано для x86, x64, arm. Как правило, на Linux-подобные системы устанавливается штатная среда исполнения, однако для некоторых сборок операционной системы, возможно, потребуются доработки.
QNX	6.5	Среда исполнения поставляется по запросу.
Android	4.0	Android-устройства чаще всего используются для визуализации. В этом случае достаточно наличия только клиента визуализации, в качестве которого может выступать штатный браузер устройства, и никаких дополнительных приложений устанавливать не требуется. В том случае, если штатный браузер не может быть использован, то по запросу пользователя компания "МПС Софт" может предоставить свою разработку. Данный компонент входит в среду исполнения для данной ОС. Среда исполнения поставляется по запросу.
Эльбрус		Поддержано для Эльбрус x64

Название ОС	Минимальная версия	Примечания
MAC OS		Поддерживается только для клиента визуализации. В качестве клиента визуализации может быть использован браузер, установленный в данной ОС и поддерживающий HTML5

1.4. ПОДДЕРЖИВАЕМЫЕ КОНТРОЛЛЕРЫ

MasterSCADA 4D RT может работать в любых контроллерах с открытой архитектурой, которые удовлетворяют системным требованиям. Если необходимый контроллер отсутствует в списке поддерживаемых, то о возможности его поддержки можно узнать у специалистов технической поддержки. Особенность поддержки контроллера состоит в том, что необходимо обеспечить работу **MasterSCADA 4D RT** не только на процессоре и операционной системе, но и учесть специфические модули ввода/вывода, встроенные каналы и другие функции.

Список уже поддерживаемых контроллеров:

Название контроллера	Операционная система
ADAM3600	Linux
Btune 6000	Linux
Платформы на базе микропроцессоров Эльбрус (для разных архитектур используются разные версии исполнительных систем)	Эльбрус (возможно AstraLinux - по запросу)
Платформы на базе микропроцессоров Baikal T1 и Baikal M	Debian GNU/Linux/QNX (сведения о данных исполнительных системах предоставляется по запросу, по умолчанию не входят в состав MasterSCADA 4D)
Fastwel MK150	Linux
ОВЕН PLC110M02	Linux

Название контроллера	Операционная система
Regul R500	QNX
TREI 915E	Linux, QNX
TREI 903	QNX
WAGO 750	Linux
WinPac 8000, 9000	Windows CE
XPAC	Windows Embedded
BeagleboneBlack	Linux
Raspberry	Linux
WirenBoard 5	Linux
WirenBoard 6	Linux
OSATEC	Linux
СЕРЕБРУМ IRIS	Linux
Любой контроллер без периферийного оборудования и специальных функций	Любая из поддерживаемых ОС

По умолчанию в проект можно добавить predetermined узлы компаний **Овен** и **ICPDAS**. Для подключения узлов других производителей необходимо выполнить **подключение** соответствующей библиотеки

1.5. УСТАНОВКА MASTERSCADА 4D

ОС Windows

Существуют два инсталлятора продуктов **MasterSCADA 4D** (© 2008-2021 MPS Soft):

- *MasterSCADA4D.exe* – мастер установки среды разработки (редактора проекта) **MasterSCADA 4D** (см. **Установка среды разработки MasterSCADA 4D**). Среда разработки также содержит в своем составе исполнительную систему **MasterSCADA 4D RT** для АРМ с ОС Windows;
- *MasterSCADA4DRT.exe* – мастер установки исполнительной системы **MasterSCADA 4D RT** для АРМ с ОС Windows (см. **Установка исполнительной системы MasterSCADA 4D RT**). Данный мастер не устанавливает редактор проекта.

Для установки требуемого продукта **MasterSCADA 4D** необходимо запустить соответствующий инсталлятор на выполнение..

После установки любого из указанных выше продуктов **MasterSCADA 4D** мастер установки инициирует процедуру установки Core Components Redistributable и/или Microsoft.NET Framework, если эти продукты еще не установлены на ПК (см. **Установка OPC Core Components Redistributable** и **Установка Framework**).

Другие ОС

Способ установки исполнительной системы зависит от операционной системы устройства.

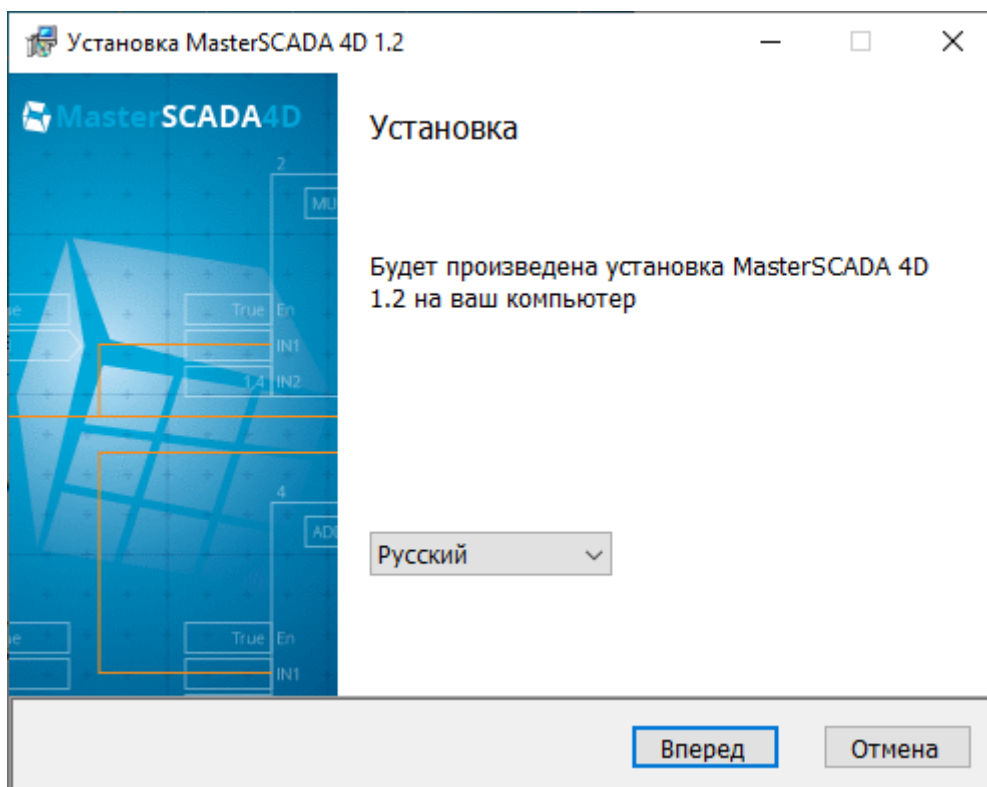
- **Linux**
- **Эльбрус**
- **QNX**
- **Windows Embedded**

1.5.1. УСТАНОВКА СРЕДЫ РАЗРАБОТКИ MASTERSCADА 4D

При запуске *MasterSCADA4D.exe* открывается следующий экран-заставка:

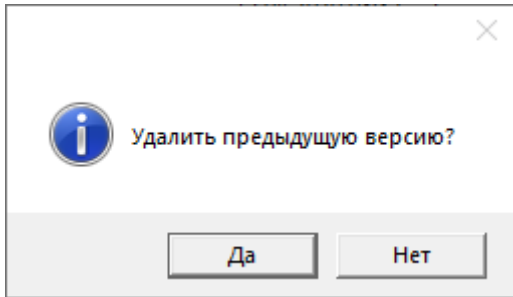


Экран-заставка отображается в течение нескольких секунд, после чего он автоматически исчезает, и открывается начальный диалог мастера установки:



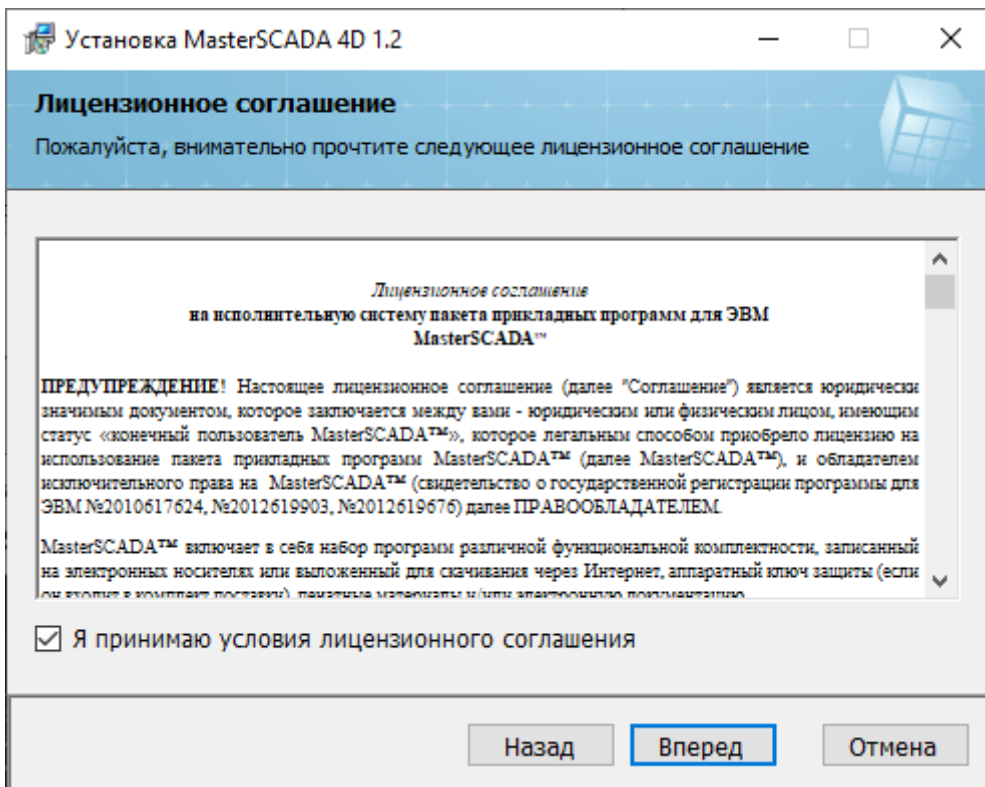
В диалоге выбирается локализация продукта (поддерживаются русский и английский языки).

Если среда разработки **MasterSCADA 4D** уже была установлена ранее, то появится предложение удалить ранее установленную версию:

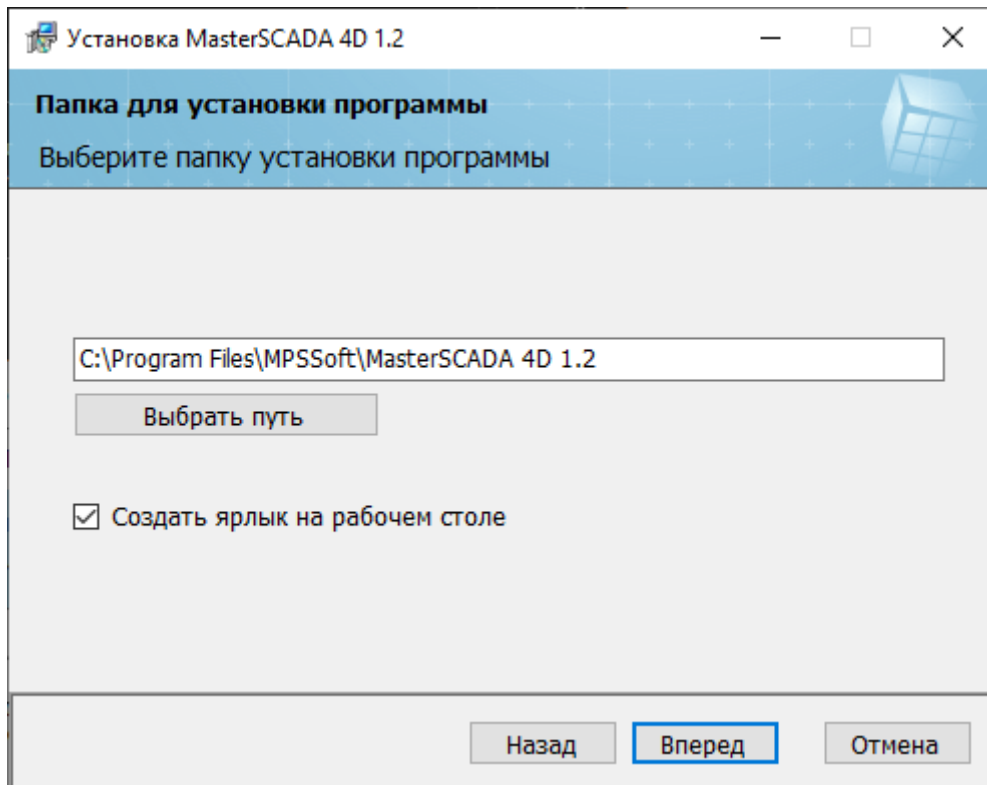


Нужно нажать кнопку Да, для продолжения установки.

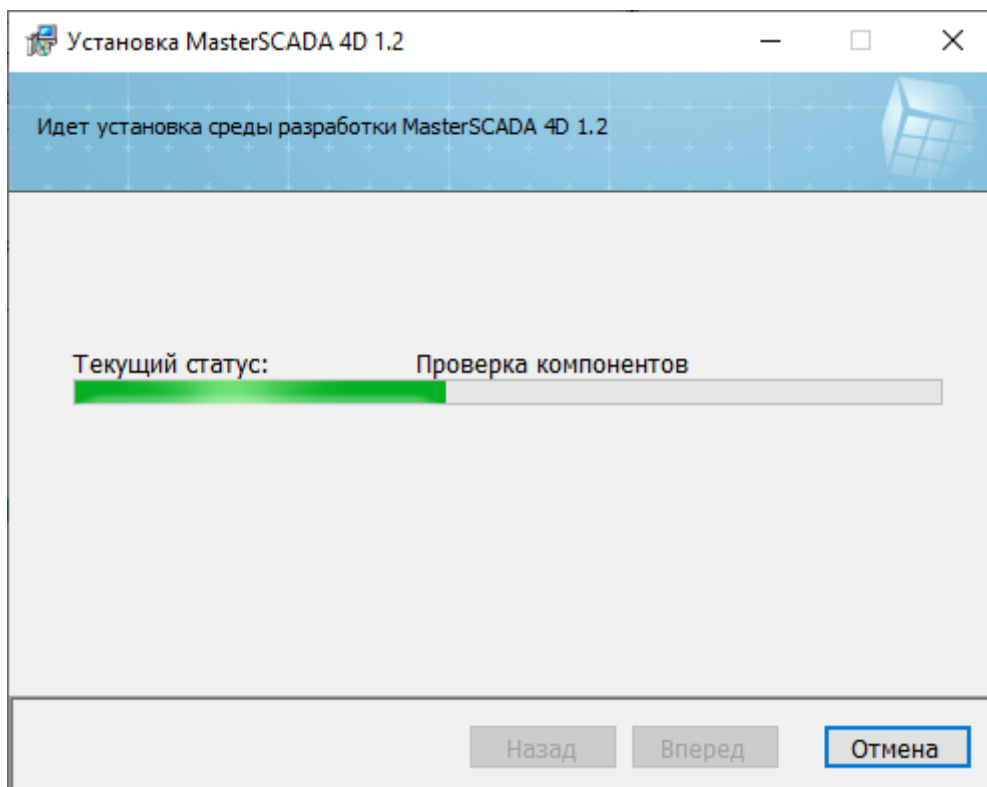
По команде Вперед мастера открывается диалог лицензионного соглашения:



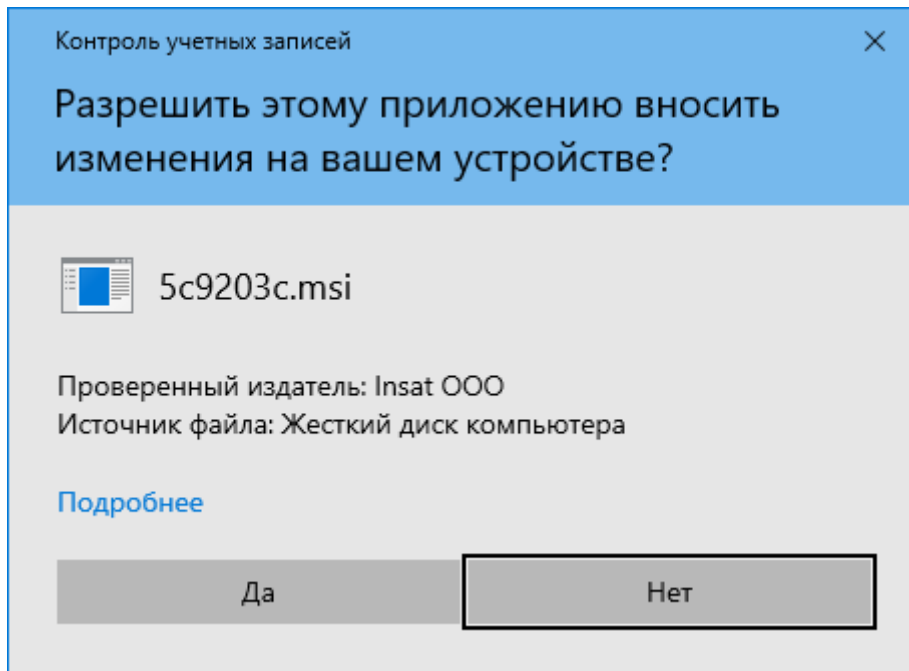
Прочитав текст, отметьте пункт Я принимаю условия лицензионного соглашения и нажмите кнопку Вперед. После этого откроется следующий диалог:



В этом диалоге можно изменить папку установки продукта, воспользовавшись кнопкой **Выбрать путь** установки, а также задать создание ярлыка программы на рабочем столе, отметив пункт **Создать ярлык на рабочем столе**. Нажмите кнопку **Вперед**, и откроется диалог, отображающий процесс установки:

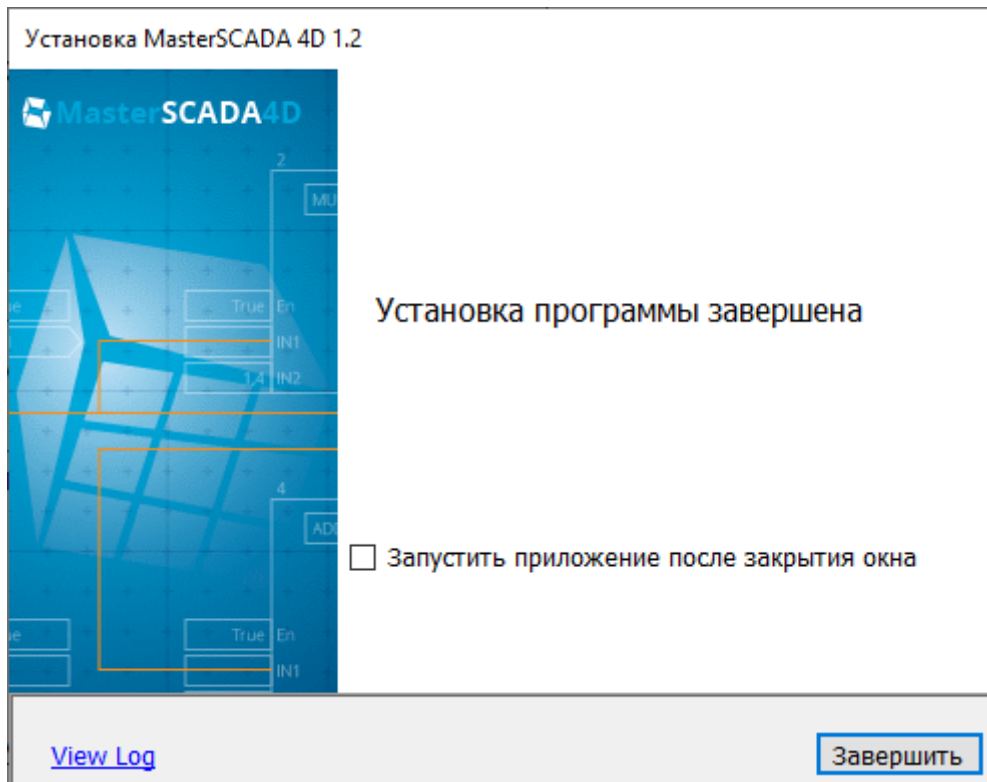


Возможно потребуется дополнительное разрешение на установку программы:



Для продолжения установки нужно нажать на кнопку Да.

Дождитесь завершения установки и появления последнего диалога мастера:



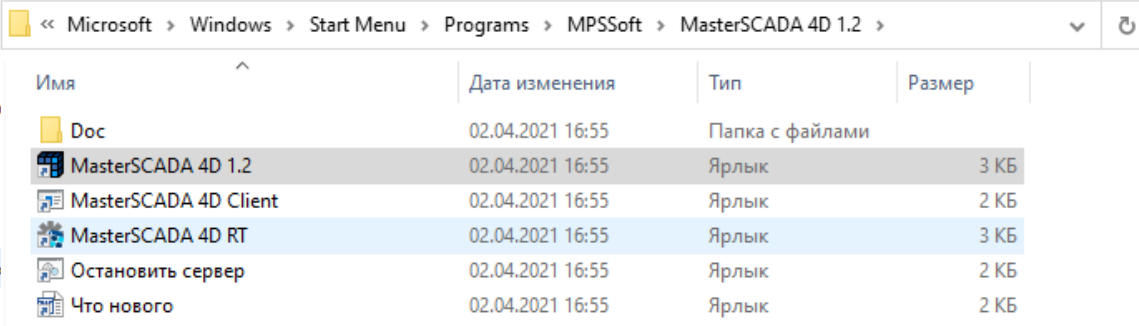
С помощью команды View Log можно посмотреть протокол установки (файл *C:\TEMP\WixSharp\MasterSCADA 4D 1.1.log*).

Если отметить пункт Запустить приложение после закрытия окна, то по команде Завершить запустится среда разработки **MasterSCADA 4D**.

Для завершения работы мастера нажмите кнопку Завершить.

1.5.2. ОПИСАНИЕ ЯРЛЫКОВ MASTERSCADА 4D

По завершении установки мастер создает программную группу Пуск - Все программы - MPSSoft - MasterSCADA 4D, содержащую следующие пункты меню (см. также Использование ярлыков MasterSCADA 4D):

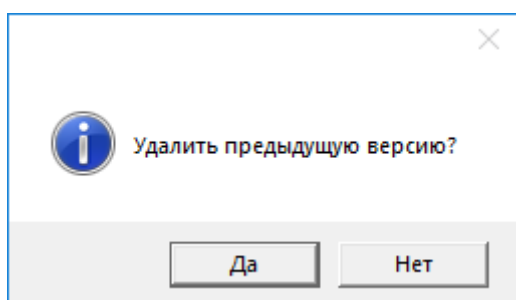


Имя	Дата изменения	Тип	Размер
Дос	02.04.2021 16:55	Папка с файлами	
MasterSCADA 4D 1.2	02.04.2021 16:55	Ярлык	3 КБ
MasterSCADA 4D Client	02.04.2021 16:55	Ярлык	2 КБ
MasterSCADA 4D RT	02.04.2021 16:55	Ярлык	3 КБ
Остановить сервер	02.04.2021 16:55	Ярлык	2 КБ
Что нового	02.04.2021 16:55	Ярлык	2 КБ

- Дос – данная группа содержит следующие ярлыки:
 - Лицензионное соглашение – открывает лицензионное соглашение (файл *<папка установки MasterSCADA 4D>\bin\Config\License.rtf*);
 - Разработка драйверов и ФБ в MasterSCADA – открывает файл *"Разработка драйверов и ФБ в MasterSCADA 4D.docx"* из папки *<папка установки MasterSCADA 4D>\API\Doc*;
 - Справка – открывает настоящую справку (файл *<папка установки MasterSCADA 4D>\bin\Config\Help\plcd.chm*);
- **MasterSCADA 4D** – запускает среду разработки (редактор проекта; команда *<папка установки MasterSCADA 4D>\bin\ProjectEditor.exe*);
- **MasterSCADA 4D Client** – запускает клиент (браузер) MasterSCADA 4D и открывает сайт по умолчанию (команда *<папка установки MasterSCADA 4D>\bin\Config\MasterPLC\WIN32\client\MasterSCADA4DClient.exe http://127.0.0.1:8043*);
- **MasterSCADA 4D RT** – запускает исполнительную систему (серверы *tplc.exe* и *node_ms4d.exe*, веб-сервер *nginx.exe* и др.) из папки установки продукта (команда *<папка установки MasterSCADA 4D>\bin\Config\MasterPLC\WIN32\run_hmi.bat*). Рабочая папка, в которой хранятся конфигурация проекта, сайт и протоколы работы сервера, – *%<профиль пользователя>%\AppData\Roaming\MPSSoft\MasterSCADA4D1.2\Server*
- Остановить сервер – останавливает исполнительную систему (команда *<папка установки MasterSCADA 4D>\bin\Config\MasterPLC\WIN32\stop_hmi.bat*);
- Что нового – открывает файл *<папка установки MasterSCADA 4D>\bin\Config\WhatsNew.rtf*.

1.5.2.1. ОБНОВЛЕНИЕ ВЕРСИИ НА ОС WINDOWS

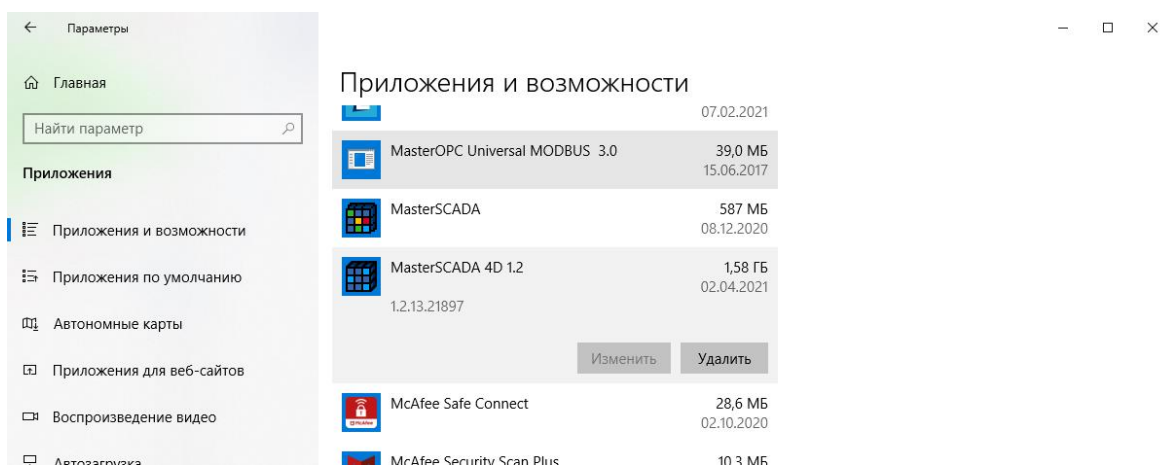
Для обновления версии среды разработки **MasterSCADA 4D** необходимо запустить инсталлятор *MasterSCADA4D.exe*, а для обновления версии исполнительной системы **MasterSCADA 4D RT** необходимо запустить инсталлятор *MasterSCADA4DRT.exe*. Так как на компьютере уже установлена предыдущая версия продукта, то появляется диалоговое окно::



Если нажать кнопку Да, то ранее установленная версия удалится, и запустится установка новой.

1.5.2.2. УДАЛЕНИЕ MASTERSCADА 4D

Для удаления **MasterSCADA 4D** необходимо в панели управления Windows открыть раздел Программы и компоненты (Приложения и возможности), выбрать **MasterSCADA 4D** и нажать кнопку Удалить:



1.5.2.3. ЗАПУСК СРЕДЫ РАЗРАБОТКИ

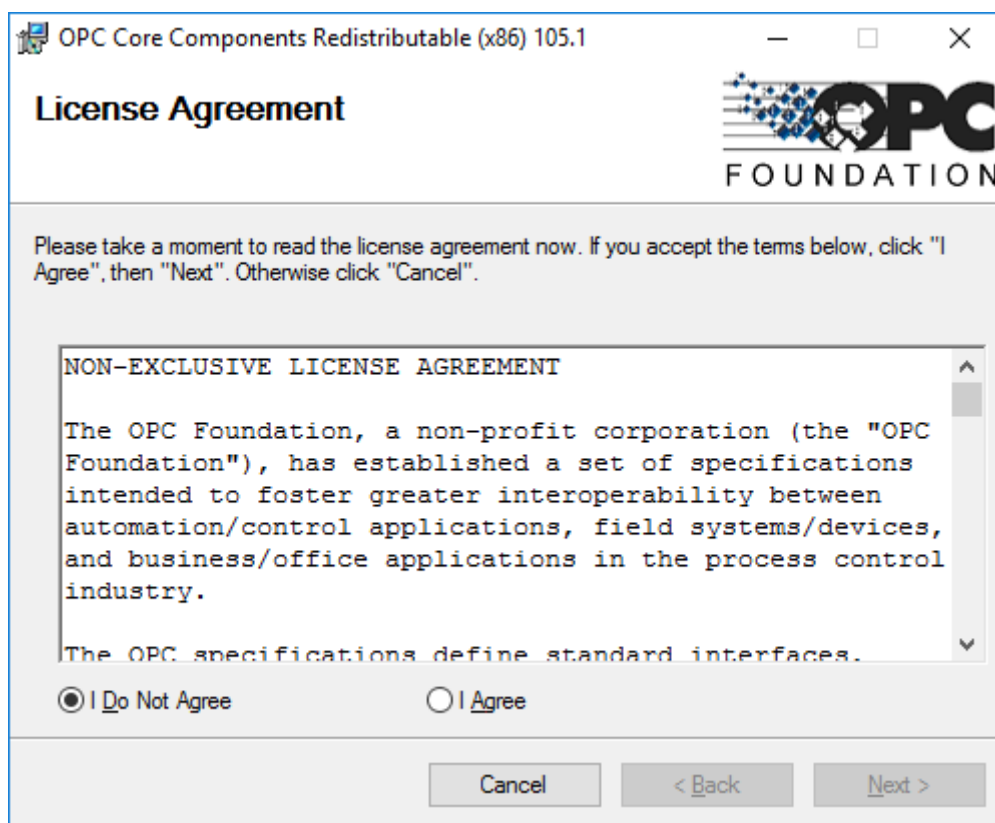
Для запуска среды разработки (редактора проекта) следует выбрать в меню Пуск пункт, соответствующий версии **MasterSCADA 4D**, либо дважды нажать левой кнопкой мыши на ярлык на рабочем столе **MasterSCADA 4D [номер версии]**, либо открыть файл *<папка установки продукта>\bin\ProjectEditor.exe* в проводнике Windows. При запуске открывается стартовое меню редактора проекта (см. **Стартовое меню**).

При старте редактора проекта в директории *%профиль пользователя%\AppData\Roaming\MPSSoft\MasterSCADA4D* создается папка *sys_log*, содержащая протоколы работы (файлы с именами *уу_мм_дд НН_ММ_СС.log*).

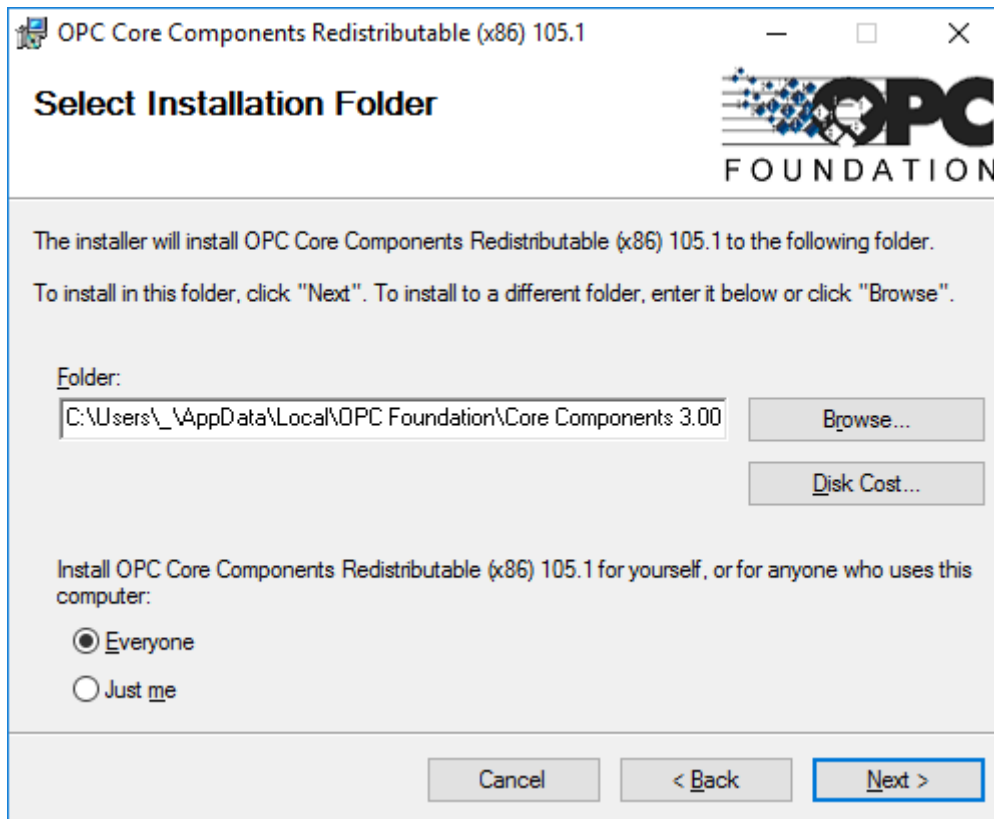
1.5.2.4. УСТАНОВКА ДОПОЛНИТЕЛЬНЫХ КОМПОНЕНТОВ

1.5.2.4.1. УСТАНОВКА OPC CORE COMPONENTS REDISTRIBUTABLE

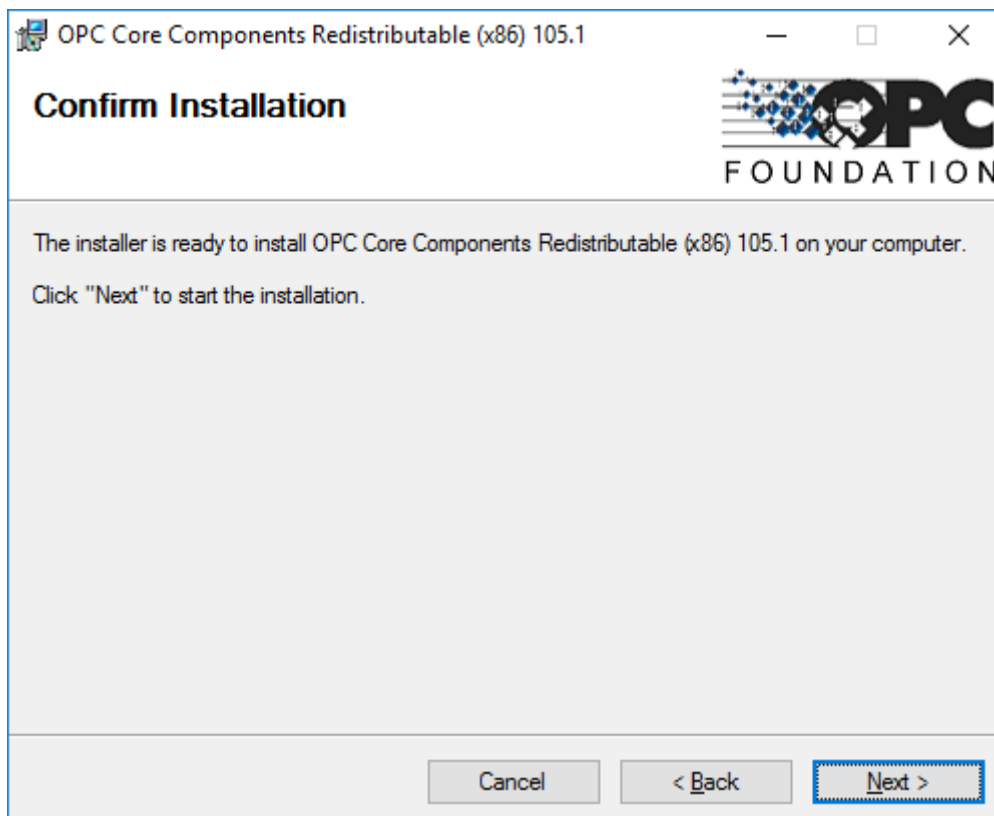
Мастер установки продукта **MasterSCADA 4D** запускает процедуру установки OPC Core Components Redistributable автоматически, если этот продукт еще не установлен на ПК.



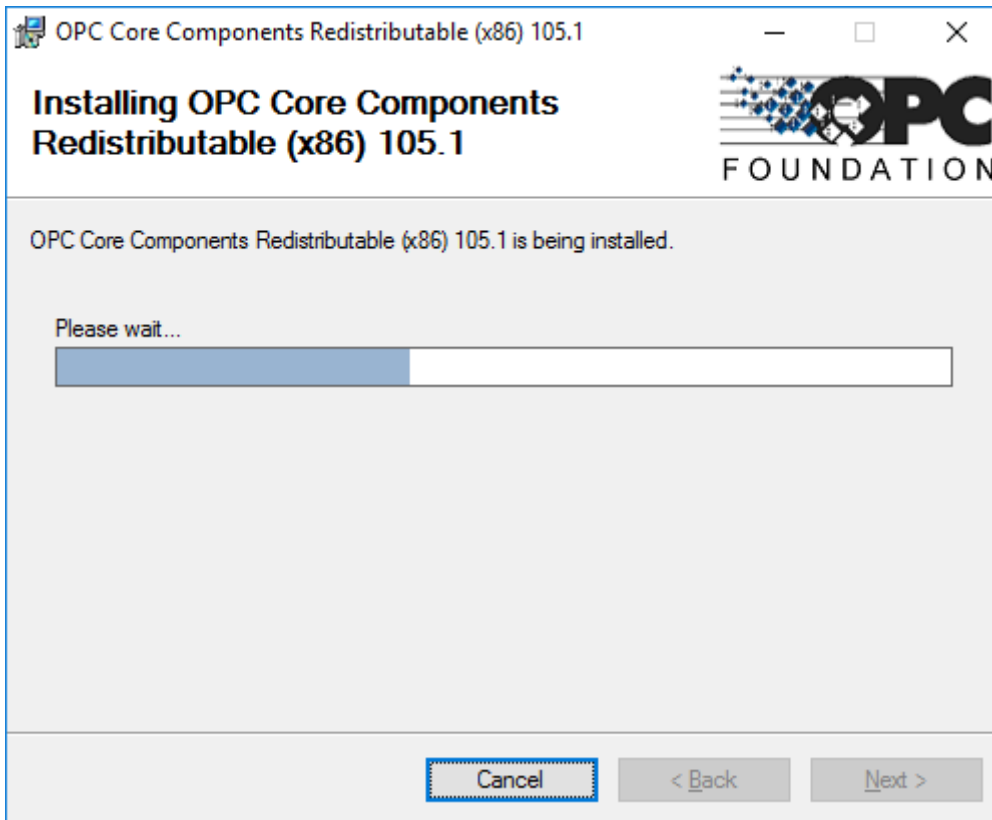
В первом диалоге мастера установки OPC Core Components Redistributable – диалоге лицензионного соглашения – необходимо выбрать пункт I Agree и нажать Next – откроется следующий диалог мастера:



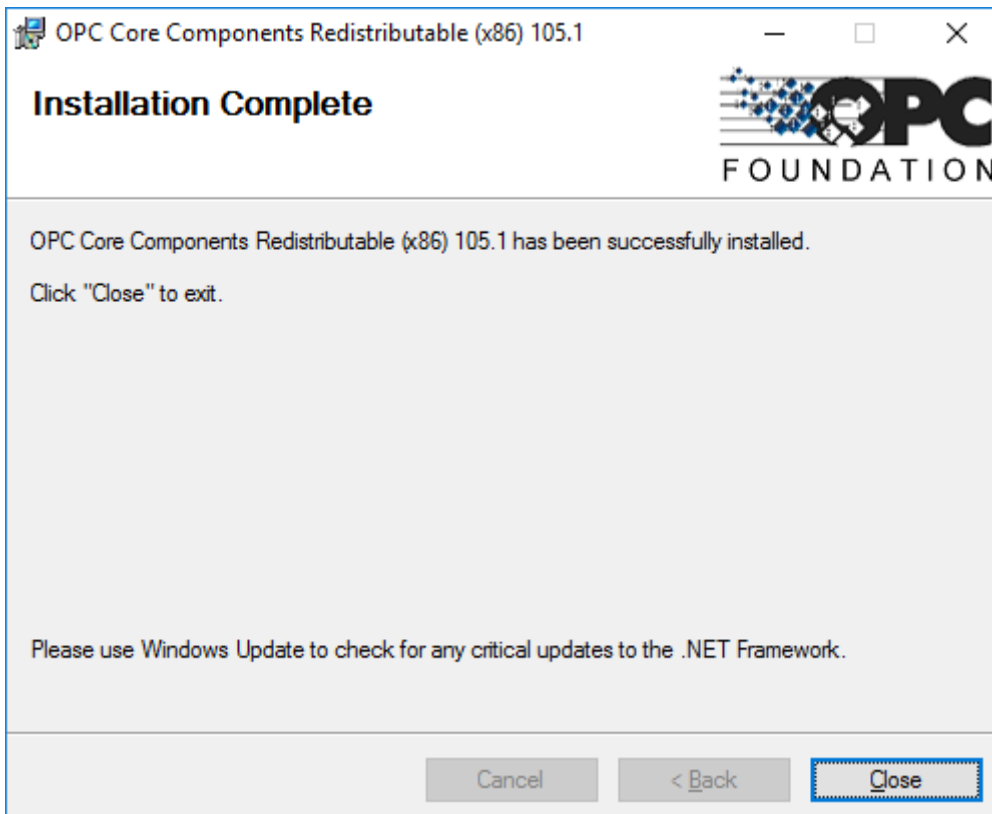
В этом диалоге можно изменить папку установки продукта (кнопка Browse) и задать вид установки: только для себя (Just me) или для всех (Everyone). Нажмите кнопку Next – откроется диалог подтверждения установки:



Для запуска процедуры установки следует нажать Next – в диалоге отобразится ход установки:



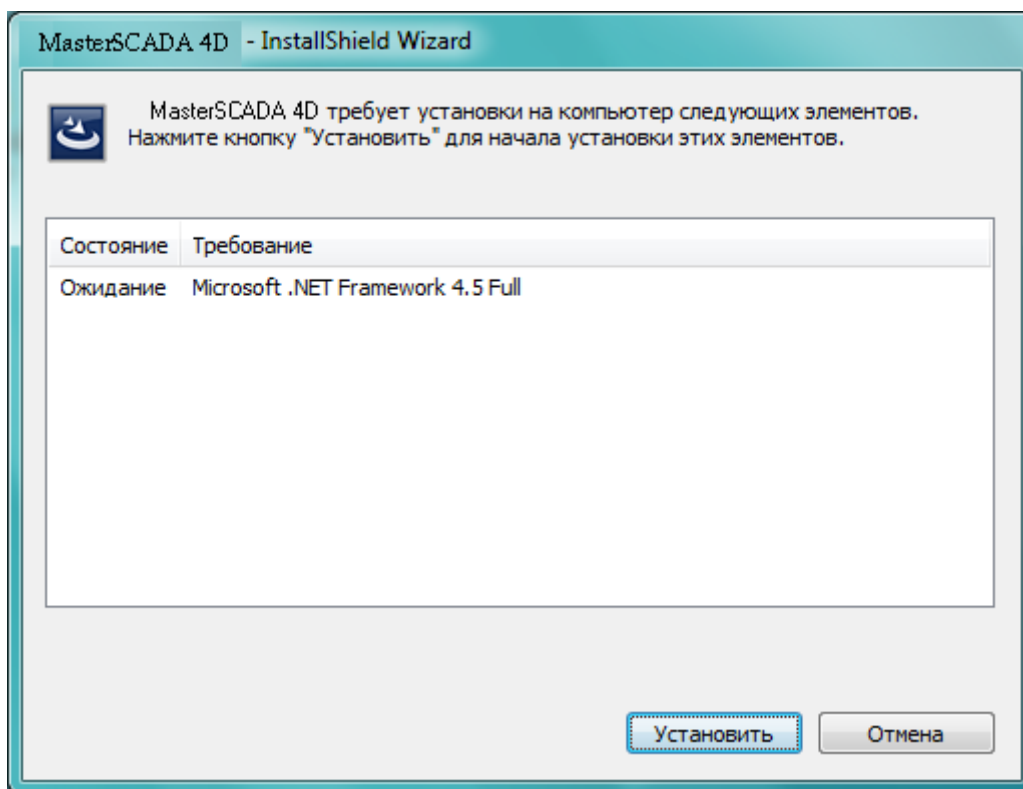
По окончании установки отобразится заключительный диалог мастера:



Для завершения работы мастера требуется нажать Close.

1.5.2.4.2. УСТАНОВКА MICROSOFT.NET FRAMEWORK

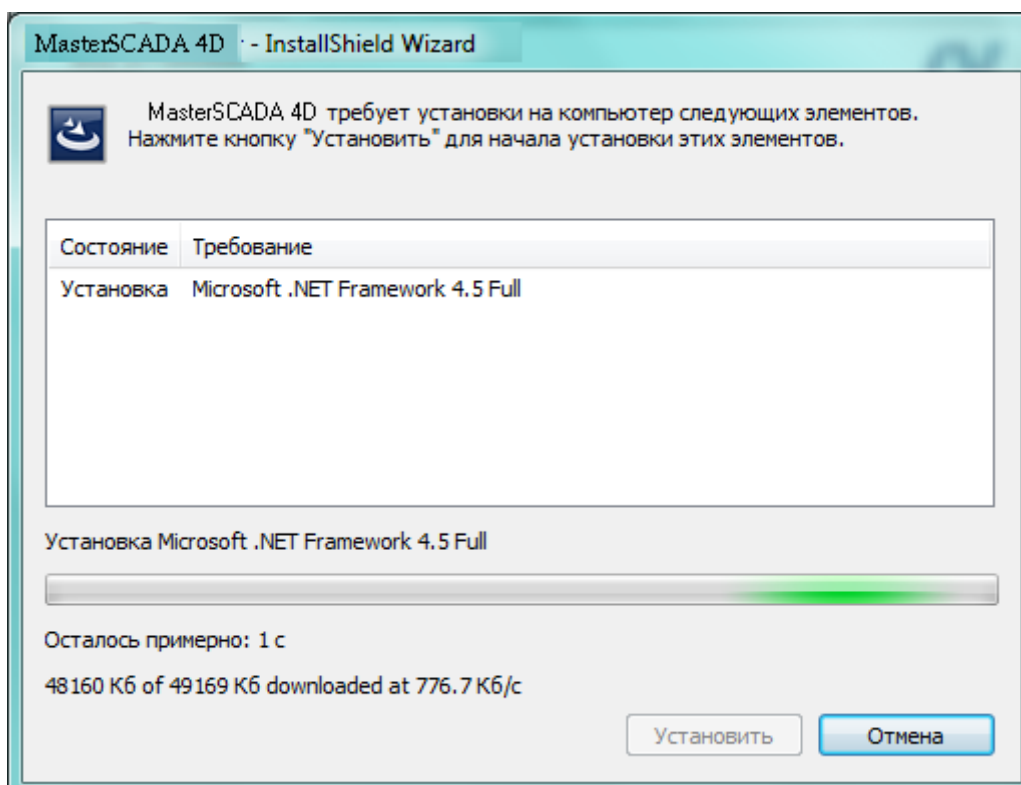
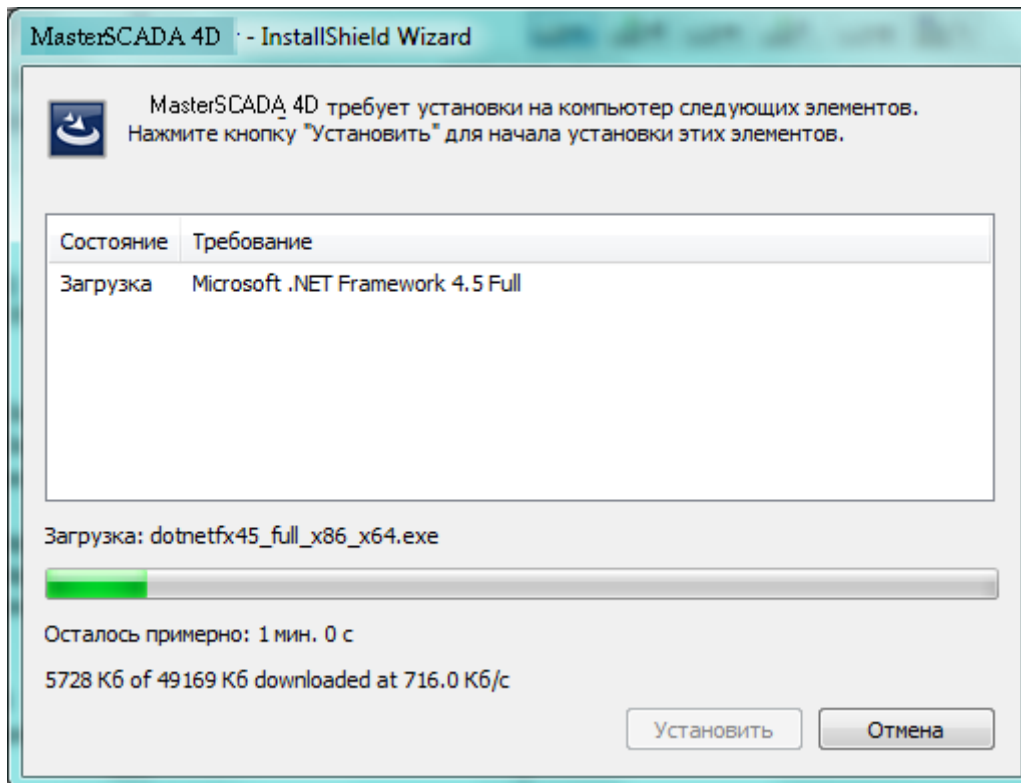
Мастер установки продукта **MasterSCADA 4D** запускает процедуру установки Microsoft.NET Framework автоматически, если этот продукт еще не установлен на ПК.

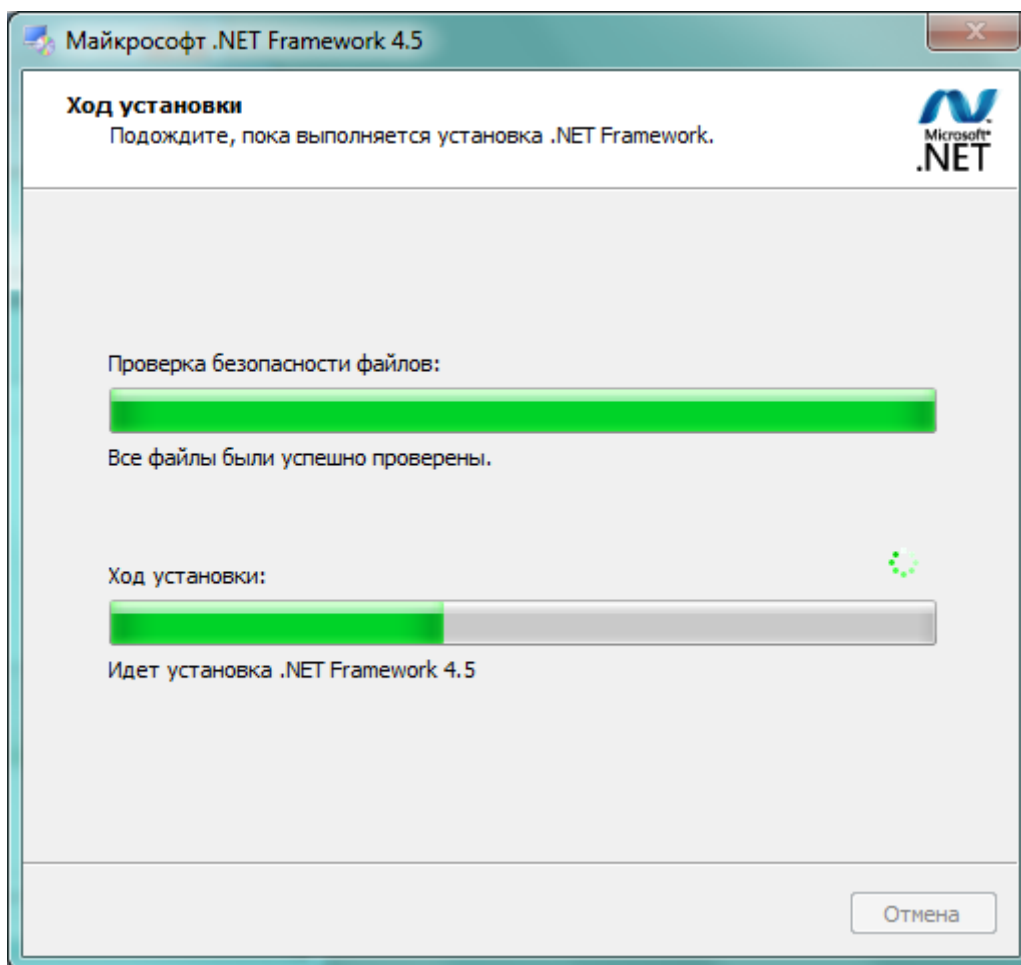


Инструменты диалога:

- Установить – установить Microsoft.NET Framework необходимой версии. Если папка *ISSetupPrerequisites*, содержащая дистрибутивы Microsoft.NET Framework, отсутствует на установочном диске, то дистрибутив будет загружен с сайта Microsoft (при наличии подключения к сети Интернет);
- Отмена – отменить установку.

Нажмите Установить для установки Microsoft.NET Framework – этот процесс производится автоматически (на рисунках ниже показаны загрузка и установка):





После установки Microsoft.NET Framework может потребоваться перезагрузка компьютера.

1.5.3. УСТАНОВКА СРЕДЫ ИСПОЛНЕНИЯ MASTERSCADА 4D

Среда исполнения может быть установлена на устройство без установки среды разработки. Способ установки исполнительной системы зависит от операционной системы устройства.

- **Windows**
- **Linux**
- **Эльбрус**
- **QNX**
- **Windows Embedded**

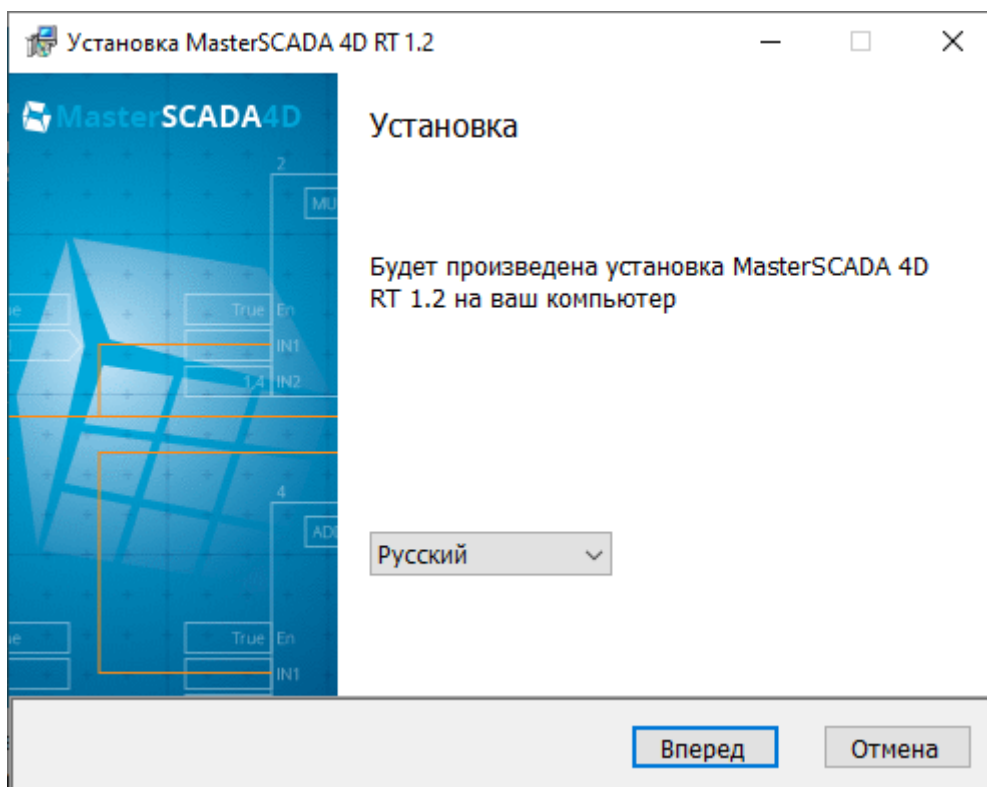
1.5.3.1. УСТАНОВКА НА ОС WINDOWS

1.5.3.1.1. УСТАНОВКА ИСПОЛНИТЕЛЬНОЙ СИСТЕМЫ MASTERSCADА 4D RT

При запуске *MasterSCADA4DRT.exe* открывается следующий экран-заставка:

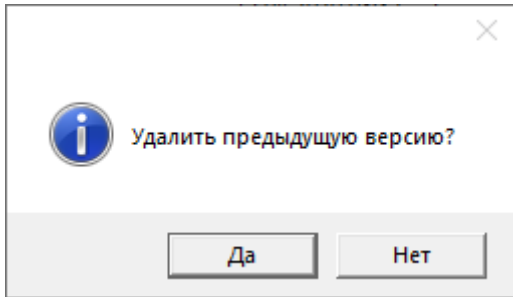


Экран-заставка отображается в течение нескольких секунд, после чего автоматически исчезает, и открывается начальный диалог мастера установки:



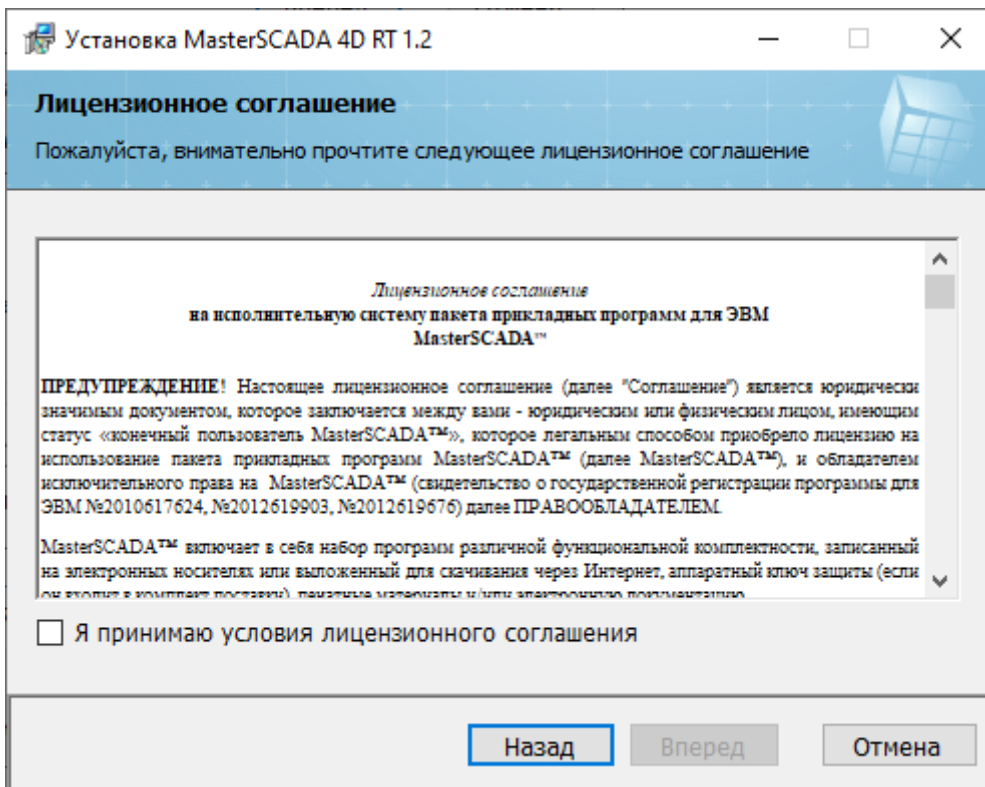
В диалоге выбирается локализация продукта (поддерживаются русский и английский языки).

Если исполнительная система **MasterSCADA 4D RT** уже была установлена ранее, то появится предложение удалить ранее установленную версию:

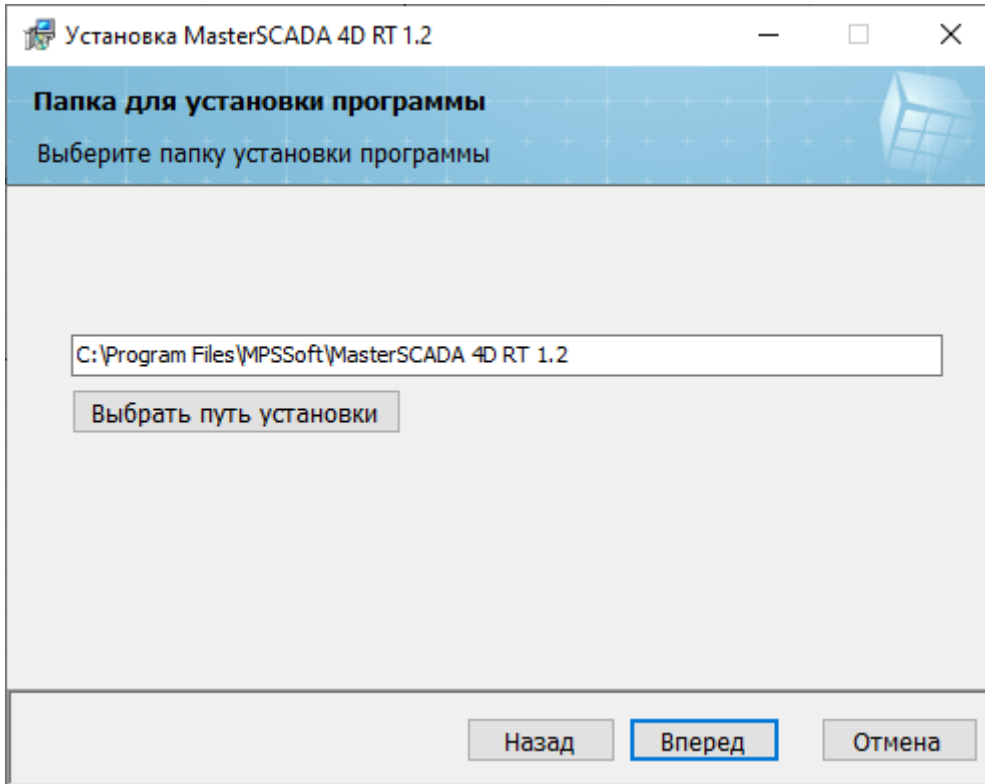


Нужно нажать кнопку Да, для продолжения установки.

По команде Вперед мастера открывается диалог лицензионного соглашения:

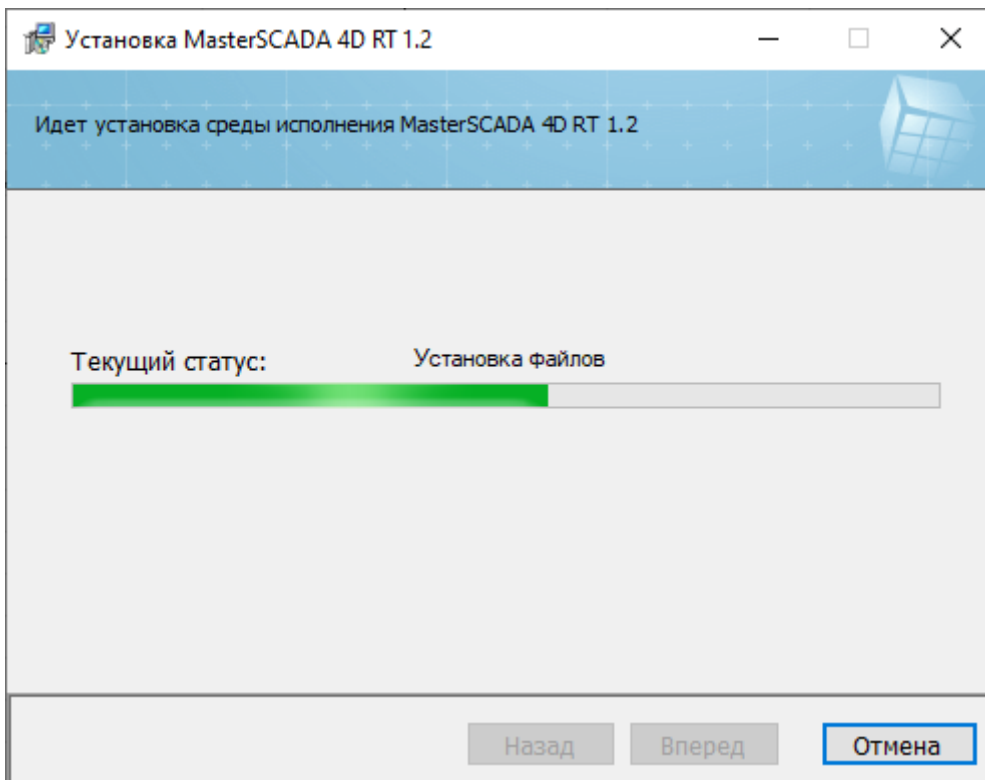


Прочитав текст, отметьте пункт Я принимаю условия лицензионного соглашения и нажмите кнопку Вперед – откроется следующий диалог:

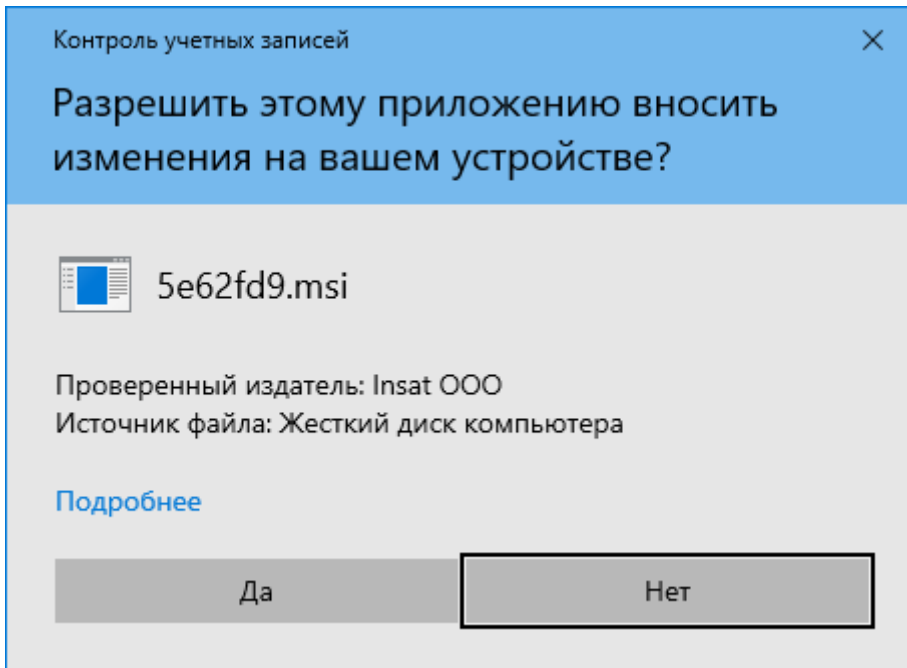


В этом диалоге можно изменить папку установки продукта, воспользовавшись кнопкой **Выбрать путь установки**, а также задать создание ярлыка программы на рабочем столе, отметив пункт **Создать ярлык на рабочем столе**. Нажмите кнопку **Вперед**

– откроется диалог, отображающий процесс установки:



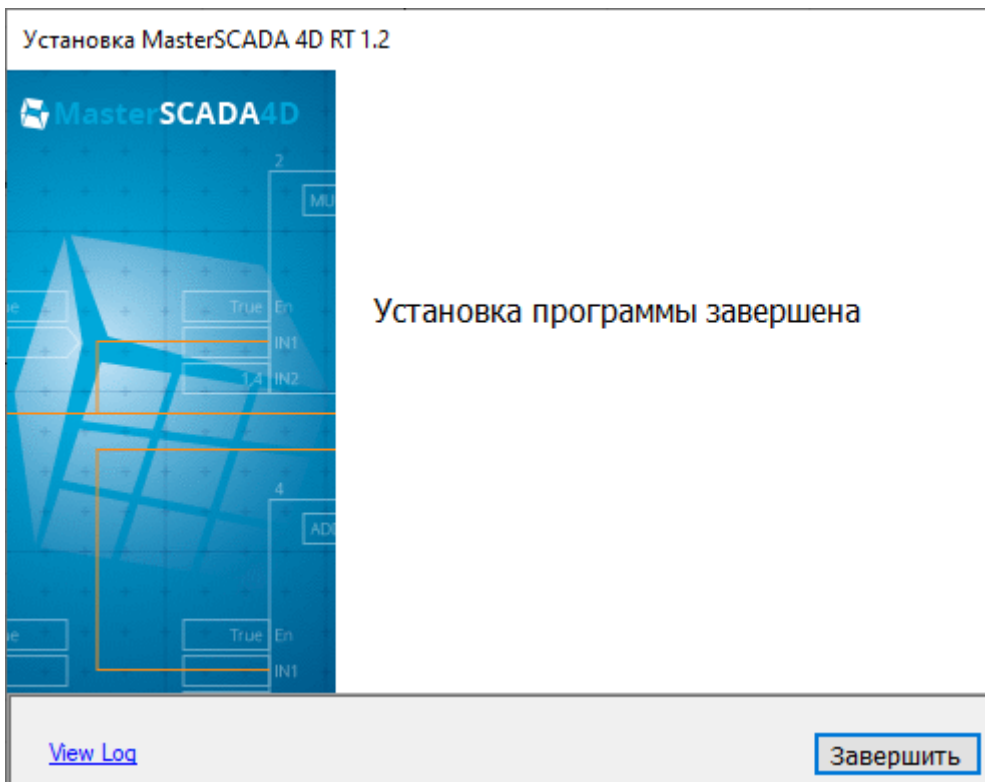
Возможно потребуется дополнительное разрешение на установку программы:



Для продолжения установки нужно нажать на кнопку Да.

Если ранее еще не были установлены **OPC Core Components**, драйвер ключа защиты, то их установка запустится автоматически

Дождитесь завершения установки и появления последнего диалога мастера:



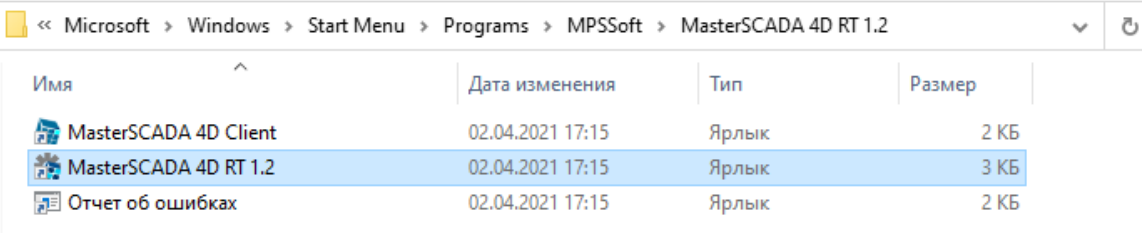
С помощью команды View Log можно посмотреть протокол установки (файл *C:\TEMP\WixSharp\MasterSCADA 4D RT 1.1.log*).

Если отметить пункт Запустить приложение после закрытия окна, то по команде Завершить запустится среда исполнения **MasterSCADA 4D RT**.

Для завершения работы мастера нажмите Завершить.

1.5.3.1.2. ОПИСАНИЕ ЯРЛЫКОВ MASTERSCADА 4D RT

По завершении установки мастер создает программную группу Пуск - Все программы - MPSSoft - MasterSCADA 4D RT, содержащую следующие пункты (см. также **Использование ярлыков MasterSCADA 4D**):

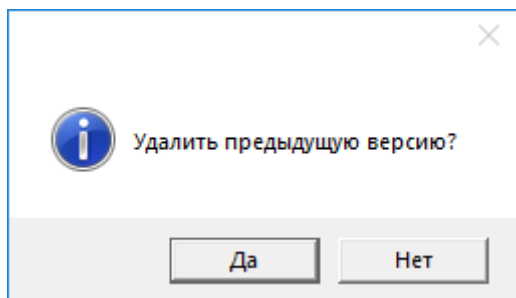


Имя	Дата изменения	Тип	Размер
MasterSCADA 4D Client	02.04.2021 17:15	Ярлык	2 КБ
MasterSCADA 4D RT 1.2	02.04.2021 17:15	Ярлык	3 КБ
Отчет об ошибках	02.04.2021 17:15	Ярлык	2 КБ

- **MasterSCADA 4D Client** – запускает клиент (браузер) **MasterSCADA 4D** из папки установки продукта и открывает сайт по умолчанию (команда *<папка установки MasterSCADA 4D RT>\client\MasterSCADA4DClient.exe http://127.0.0.1:8043*);
- **MasterSCADA 4D RT** – запускает исполнительную систему (серверы *tplc.exe* и *node_ms4d.exe*, веб-сервер *nginx.exe* и др.) из папки установки продукта (команда *<папка установки MasterSCADA 4D RT>\run_hmi.bat*). Рабочая папка, в которой хранятся конфигурация проекта, сайт и протоколы работы сервера, – *%<профиль пользователя>%\AppData\Roaming\MPSSoft\MasterSCADA4DRT1.1\Server*;
- Отчет об ошибках - запускает формирование отчета об ошибках

1.5.3.1.3. ОБНОВЛЕНИЕ ВЕРСИИ ИСПОЛНИТЕЛЬНОЙ СИСТЕМЫ НА ОС WINDOWS

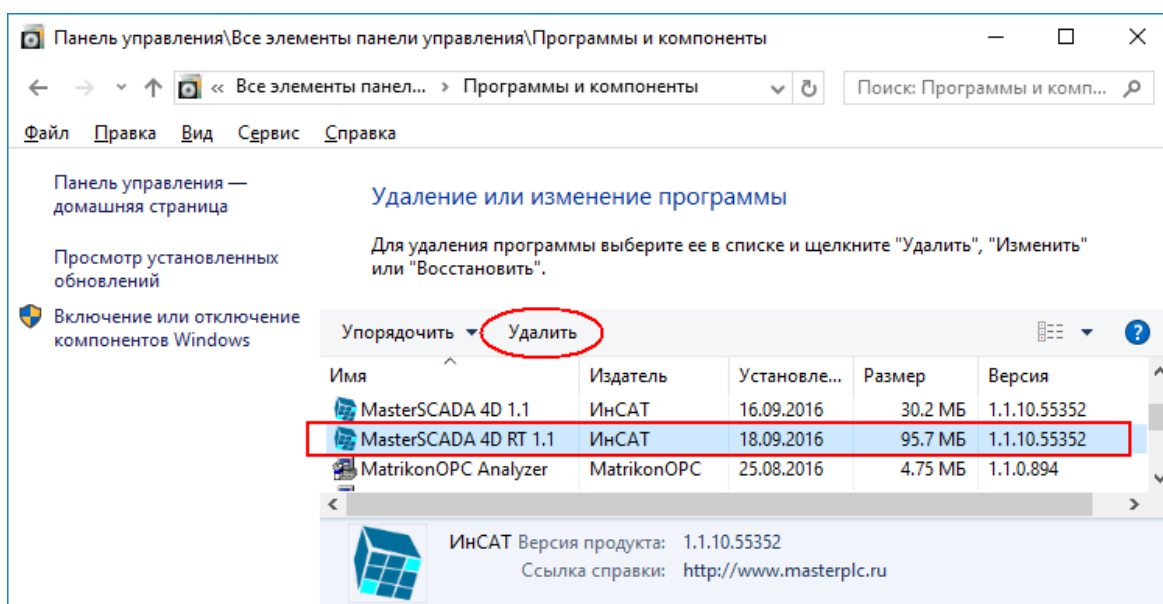
Для обновления версии среды разработки **MasterSCADA 4D** необходимо запустить инсталлятор *MasterSCADA4D.exe*, а для обновления версии исполнительной системы **MasterSCADA 4D RT** необходимо запустить инсталлятор *MasterSCADA4DRT.exe*. Так как на компьютере уже установлена предыдущая версия продукта, то появляется диалоговое окно::



Если нажать кнопку Да, то ранее установленная версия удалится, и запустится установка новой.

1.5.3.1.4. УДАЛЕНИЕ MASTERSCADА 4D RT

Для удаления **MasterSCADA 4D RT** необходимо в панели управления Windows открыть раздел Программы и компоненты, выбрать **MasterSCADA 4D RT** и нажать кнопку Удалить:



1.5.3.2. УСТАНОВКА НА ОС LINUX ИЛИ ЭЛЬБРУС

Установка **MasterSCADA 4D** на операционных системах Linux и Эльбрус происходит одинаково:

- Первичная установка
- Обновление исполнительной системы
- Удаление исполнительной системы

1.5.3.2.1. УСТАНОВКА НА ОС LINUX ИЛИ ЭЛЬБРУС

Исполнительная система **MasterSCADA 4D RT** представляет собой набор файлов, которые необходимо загрузить в устройство и включить необходимые компоненты в автозапуск. В

этом случае, после перезагрузки операционной системы, **MasterSCADA 4D RT** запустится автоматически.

Подобные ОС чаще всего используются в контроллерах, поэтому в тексте упоминается именно этот форм-фактор, хотя данная инструкция относится и к другим устройствам с такими ОС.

Скрипты установки для популярных дистрибутивов Linux можно скачать с FTP-сервера компании "МПС Софт". Доступ предоставляется по запросу. На FTP-сервере техподдержки находятся отдельные комплекты инсталляции для основных поддерживаемых контроллеров, учитывающие их особенности.

Для установки исполнительной системы необходимо скопировать содержимое папки, скачанной с FTP-сервера, во временную папку на устройстве и выполнить команды:

```
chmod 755 ./install.sh — помечает файл исполняемым;
```

```
./install.sh — устанавливает в систему необходимые пакеты: в папку /opt/mplc4 копирует исполнительную систему, обновляет конфигурацию веб-сервера Nginx, также устанавливает пакеты для 32-битных приложений (в случае если linux 64-битный), включает MasterSCADA 4D RT в автозапуск.
```

Запуск самого сервера осуществляется командой:

```
cd/opt/mplc4 - осуществляется переход в папку с установленной исполнительной системой;
```

Важно! На некоторых платформах путь к установленной исполнительной системе может быть другим. Например, у контроллера Wago750 — /home/mplc4, а у контроллера ОВЕН PLC110M2 — /root/mplc4.

```
./mplcstart.sh — запуск исполнительной системы.
```

В результате проделанных операций будут запущены следующие процессы, относящиеся к работе **MasterSCADA 4D**:

- `mplc` — основной процесс исполнительной системы;
- `mplc_servise` — вспомогательный процесс, контролирующий работу основного;
- `nginx` — процесс веб-сервера;
- `node_ms4d` — процесс генератора отчетов.

После этого можно подключаться к контроллеру из среды разработки для загрузки созданного проекта.

Осуществить вышеописанные действия можно при помощи двух приложений:

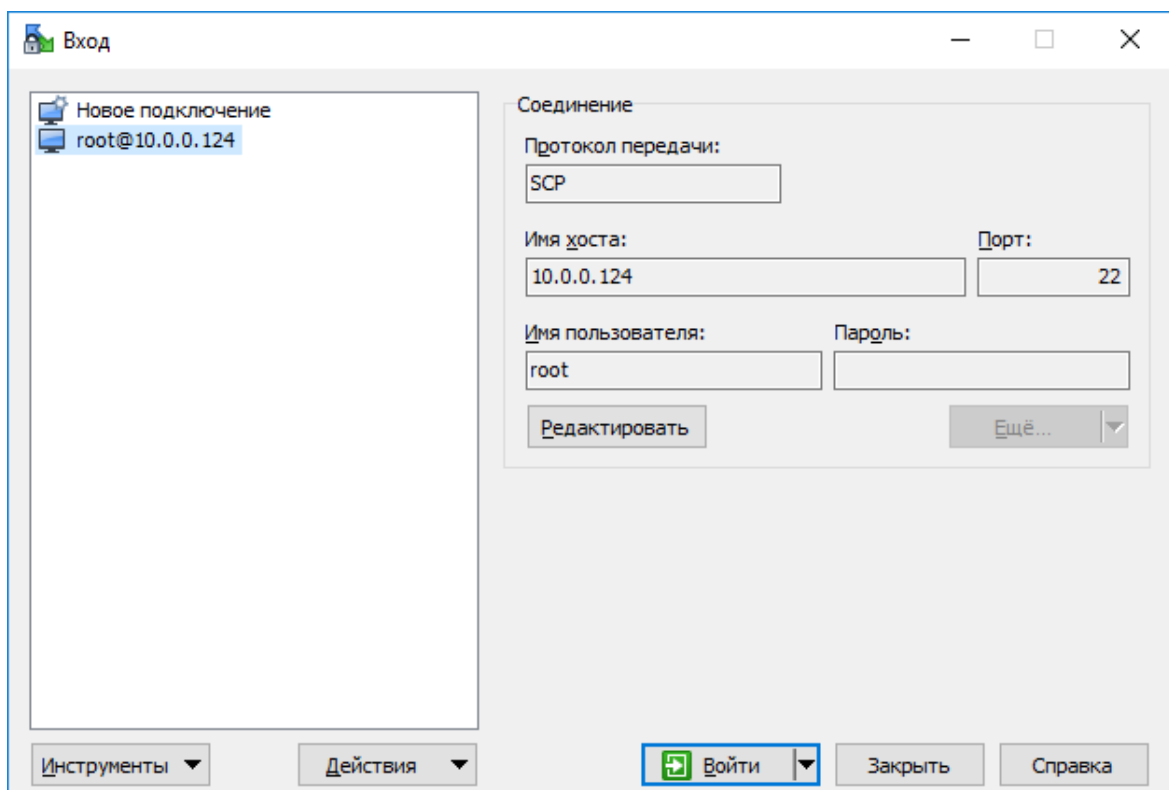
1. PuTTY

2. WinSCP

Для загрузки исполнительной системы **MasterSCADA 4D RT** используется приложение WinSCP.

1. Необходимо создать новое соединение. Настройки WinSCP:

Настройка	Значение
Файловый протокол	SCP
Сервер	IP-адрес контроллера (компьютер и контроллер должны находиться в одной подсети)
Порт	22
Имя пользователя	root
Пароль	отсутствует

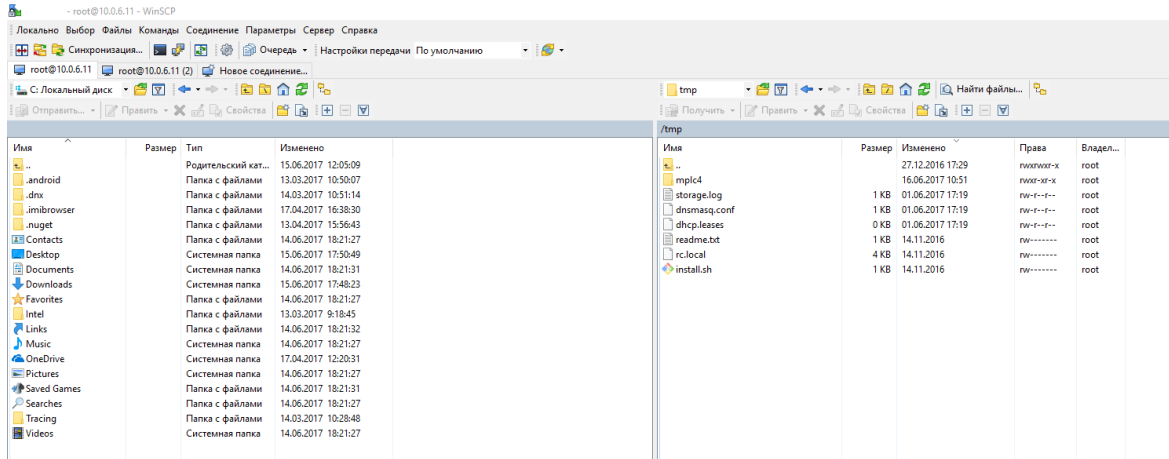


2. Подключить патчкорд к контроллеру и свитчу/компьютеру.

3. Включить питание устройства и нажать кнопку Войти.

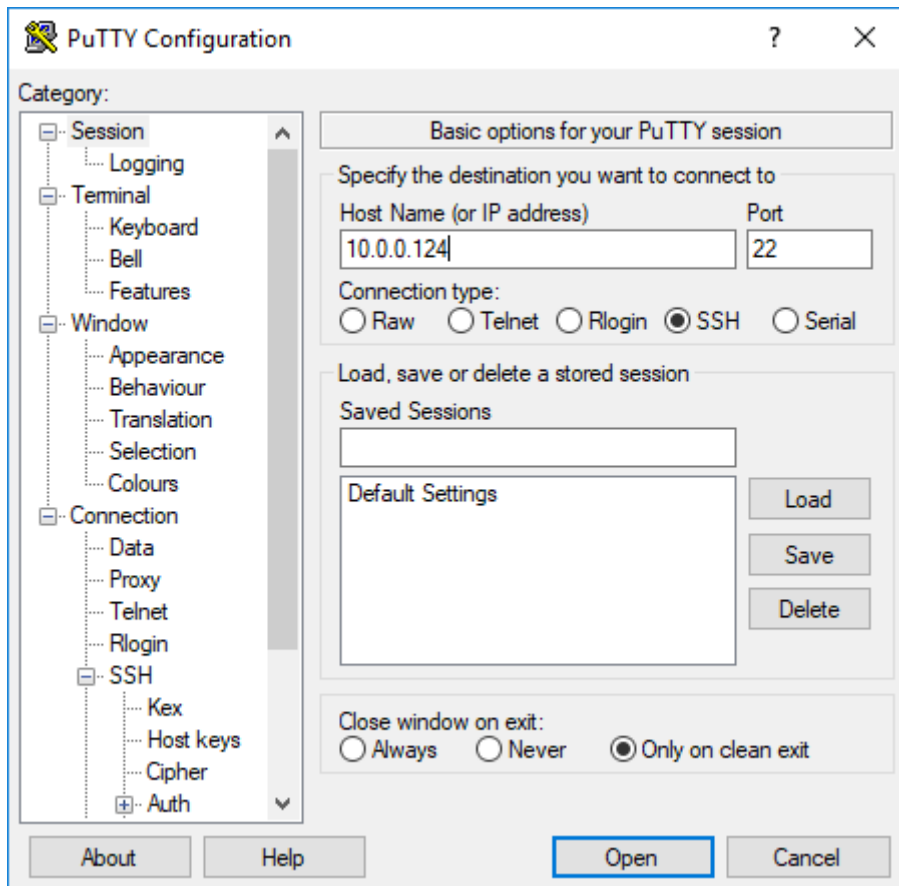
4. При входе необходимо будет указать пароль. Если контроллер подключается впервые, то в появившемся окне нажмите кнопку Да. Если появится окно с ошибкой, то необходимо нажать ОК.

5. В папку `/tmp` контроллера необходимо загрузить все файлы из папки установки.



6. В приложении PuTTY необходимо указать следующие настройки:

Настройка	Значение
connection type	SSH
Host Name(or IP address)	Установить IP-адрес контроллера
port	22



После нажатия кнопки **Open**, необходимо в окне терминала ввести логин — `root` и пароль, ранее заданный в приложении WinCSP.

7. В терминале PuTTY нужно выполнить команды установки:

```
cd/tmp
```

```
chmod 755./install.sh
```

```
/install.sh
```

При необходимости можно запустить установку с дополнительными ключами:

Ключ	Описание
<code>--prefix=<path></code>	Путь, по которому требуется установить RT (по умолчанию: <i>/opt/mplc4</i>)
<code>--platform=<NAME></code>	Задаёт тип платформы (Доступные значения PLC110, REGUL)
<code>--http-port=<port></code>	Изменяет номер порта веб-сервера (по умолчанию: 80)

Ключ	Описание
<code>--kill-codesys</code>	Отключает codesys для запуска RT
<code>--with-https</code>	Включает HTTPS протоков конфигурации Nginx
<code>--without-nginx</code>	Отключает установку Nginx. При указании этой опции установка пройдет успешно без наличия <i>nginx.tar.gz</i> .
<code>--i386-dep</code>	Устанавливает зависимости для I386 архитектуры, необходимо при установке на x64 платформы. требует подключения к сети Интернет.
<code>--nowdt</code>	Устанавливает опцию запуска RT /nowdt
<code>--config-only</code>	Устанавливает опцию обновления только конфигураций без обновления файлов <i>mplc</i> и <i>nginx</i> . С этой опцией можно запустить инсталлятор без <i>mplc.tar.gz</i> и <i>nginx.tar.gz</i>
<code>--start-delay=<seconds></code>	Устанавливает задержку запуска исполнительной системы после загрузки или перезагрузки контроллера. (по умолчанию: 0).

При этом среда исполнения запустится автоматически.

Для управления работой среды исполнения можно использовать следующие команды:

```
/etc/init.d/mplc <start | stop | restart> [local | N ] ["mplc opts"]
```

`start` - полный запуск RT (*mplc_service* и *nginx*). Если что то уже запущено будет выведено предупреждение(перезапуска не происходит)

`stop` - полная остановка RT

`restart`- полный перезапуск RT. Использовать всегда если RT уже запущен и нужно его перезапустить.

`local` - запуск RT в отладочном режиме с выводом в терминал. Если *mplc* уже запущен ни чего не произойдет.

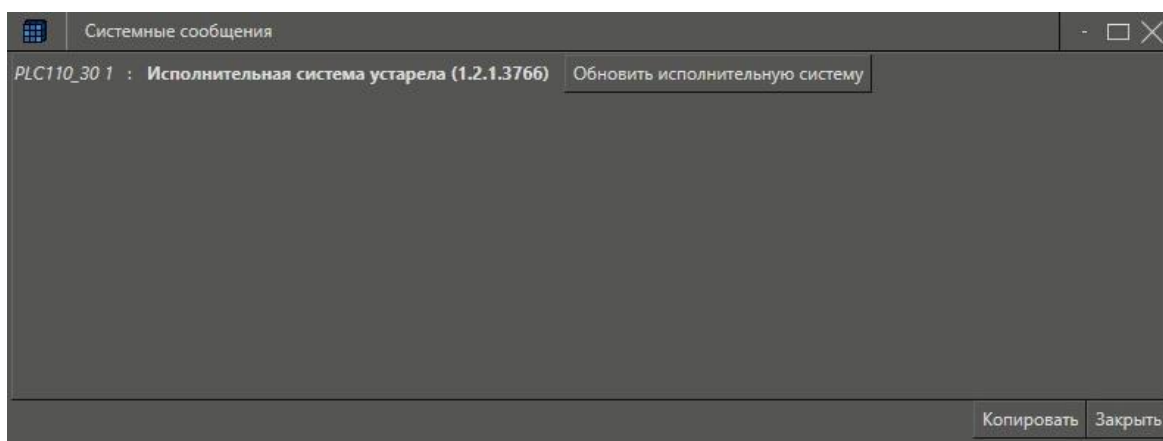
`N` - количество запускаемых сервисов *mplc*

"mplc opts" - дополнительные опции запуска которые будут переданы при запуске mpls_service

Проверить правильность установки можно, подключив приложение **PuTTY** для **диагностики**. Если терминал выдает отладочные сообщения, то система установлена корректно.

1.5.3.2.2. ОБНОВЛЕНИЕ СРЕДЫ ИСПОЛНЕНИЯ НА ОС LINUX ИЛИ ЭЛЬБРУС

Обновление среды исполнения необходимо производить если версия редактора (среды разработки) отличается от версии исполнительной системы. В этом случае система выдаст сообщение: Исполнительная система устарела. Тогда необходимо либо выполнить автоматическое обновление, нажав на кнопку Загрузить исполнительную систему, либо выполнить повторную **установку исполнительной системы**.



1.5.3.2.3. УДАЛЕНИЕ СРЕДЫ ИСПОЛНЕНИЯ НА ОС LINUX ИЛИ ЭЛЬБРУС

Для удаления среды исполнения необходимо запустить скрипт, который поставляется вместе со средой исполнения и копируется в папку контроллера при первичной установке: `/opt/mplc4/uninstall.sh`.

Важно! На некоторых платформах путь к установленной исполнительной системе может быть другим. Например, у контроллера *Wago750* — `/home/mplc4`, а у контроллера *ОВЕН PLC110M2* — `/root/mplc4`.

Для запуска можно использовать приложение PuTTY, параметры подключения описаны в разделе **Первичная установка на ОС Linux QNX Эльбрус**.

1.5.3.3. УСТАНОВКА НА ОС WINDOWS EMBEDDED

Установка среды исполнения на данную ОС будет производиться при помощи специального инсталлятора. В настоящий момент его внешний вид находится в разработке.

По вопросам установки обращайтесь по адресу технической поддержки.

1.5.3.4. УСТАНОВКА НА ОС QNX

Работа с исполнительной системой, установленной на ОС QNX, практически не отличается от работы с исполнительной системой, установленной на ОС **Linux**.

Однако, существует отличие для контроллеров фирмы ТРЕИ с предустановленной операционной системой QNX, у которых для подключения к файловой системе используется протокол FTP, а не SSH. Для загрузки файлов в память устройства можно использовать различные файловые менеджеры, например **Far Manager**.

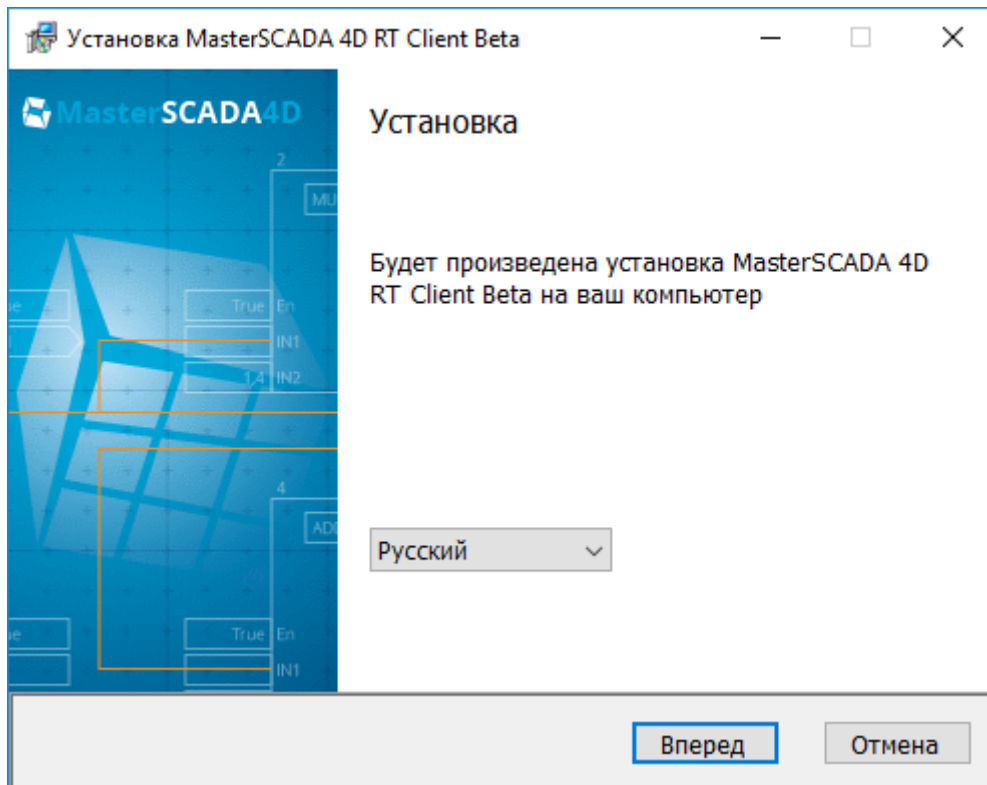
1.5.4. УСТАНОВКА И УДАЛЕНИЕ КЛИЕНТА НА ОС WINDOWS

MasterSCADA 4D Client работает на ОС Windows и устанавливается в случае, если необходимо подключаться к исполнительной системе, работающей на другом компьютере.

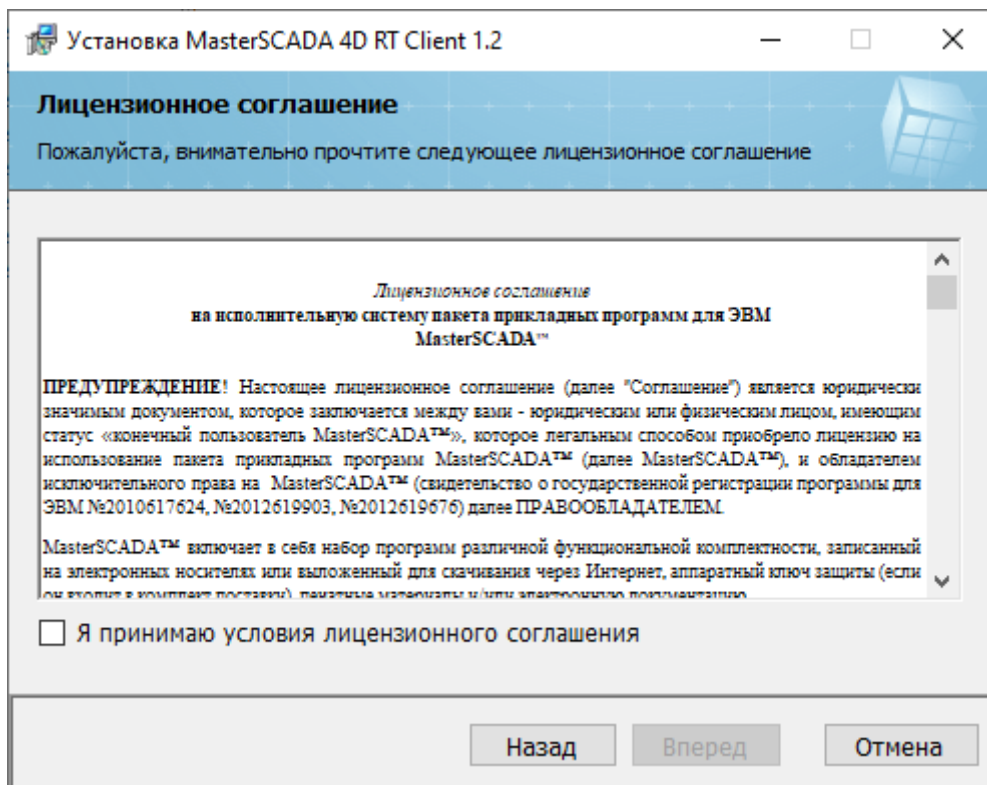
Для того чтобы начать установку, необходимо запустить файл - *MasterSCADA4DRTClient.exe*. Открывается следующий экран-заставка:



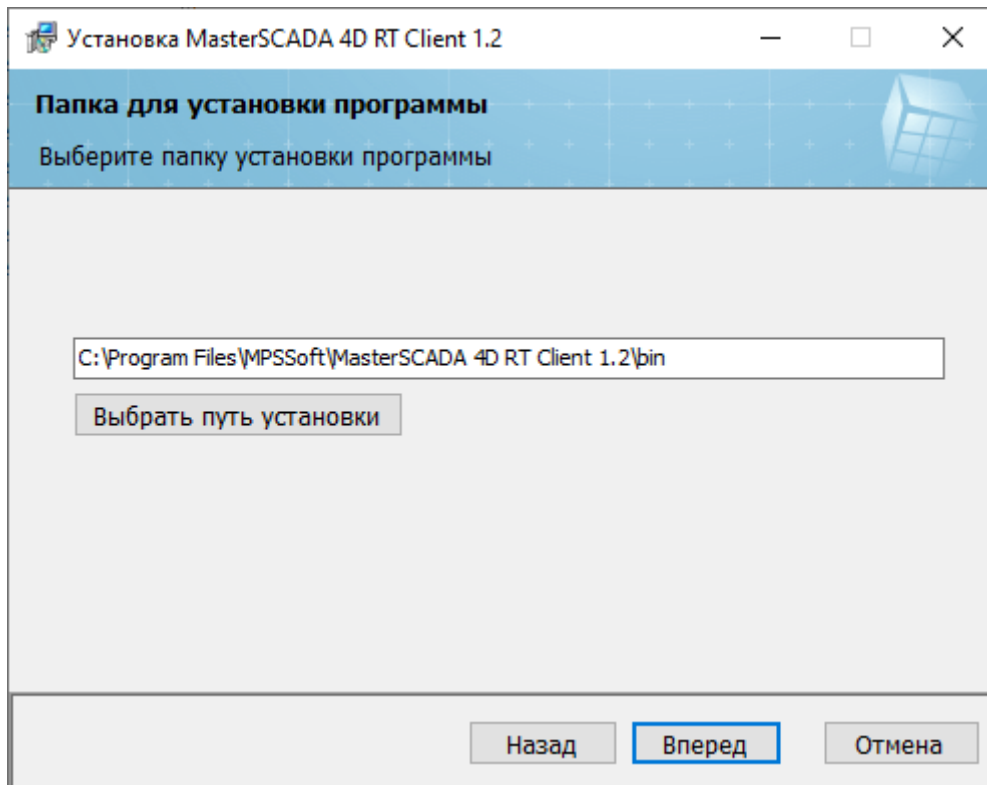
Экран-заставка отображается в течение нескольких секунд, после чего автоматически исчезает, и открывается начальный диалог мастера установки. В появившемся диалоговом окне требуется выбрать язык клиента (по умолчанию Русский) и нажать кнопку Вперед:



Далее, после ознакомления с лицензионным соглашением, необходимо отметить пункт согласия и нажать кнопку Вперед



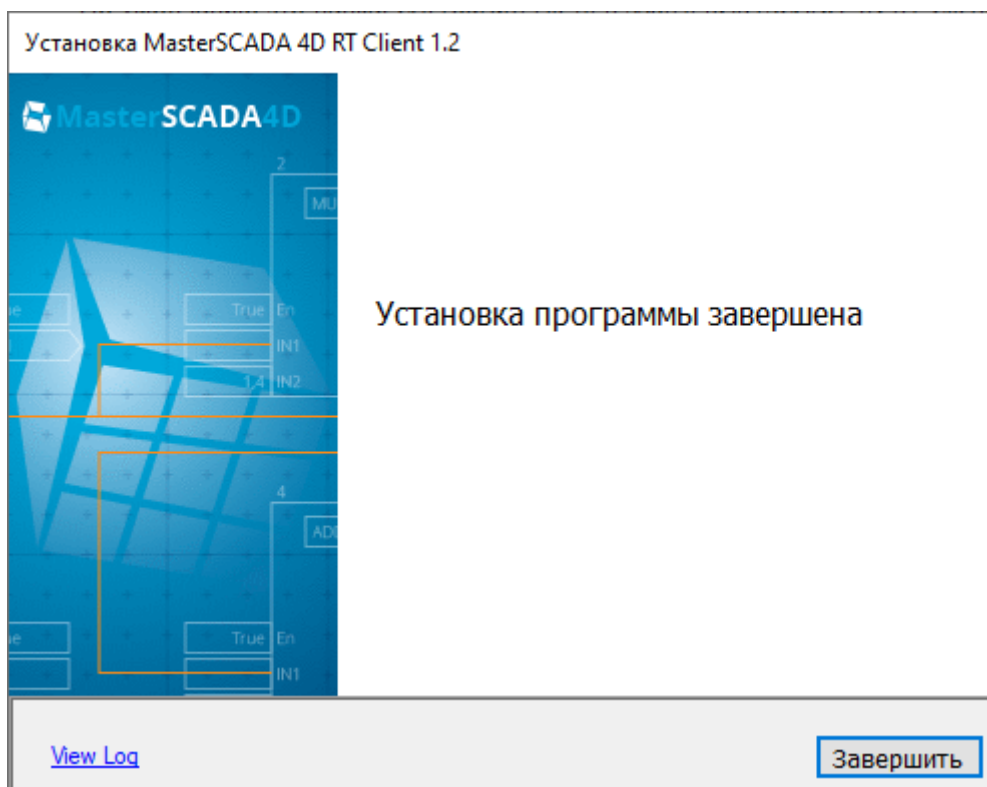
Затем выбирается папка установки приложения:



По умолчанию это папка: *C:\Program Files\MPSSoft\MasterSCADA 4D RT Client [номер версии]*

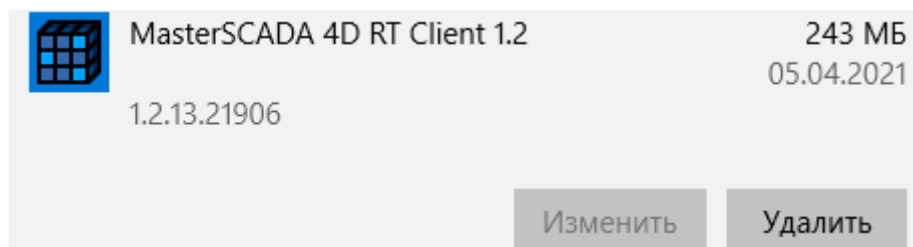
Важно! Установка приложения в папку *C:\Users\User\AppData\Roaming\MPSSoft* позволяет использовать приложение пользователям с ограниченными правами

После окончания установки появится окно, в котором необходимо нажать кнопку **Завершить**:



Для работы с клиентом необходимо использовать **специальный сервис**.

Для удаления **MasterSCADA 4D Client** необходимо в панели управления Windows открыть раздел Программы и компоненты, выбрать **MasterSCADA 4D Client** и нажать кнопку Удалить:



1.6. ЛИЦЕНЗИОННАЯ ЗАЩИТА

Среда разработки, а также демо-версия среды исполнения, в которой введено однократное ограничение времени работы в режиме опроса периферийного оборудования и межузлового обмена, предоставляются бесплатно.

Для коммерческой версии **MasterSCADA 4D RT** предусмотрены следующие варианты защиты: активация, программный ключ и USB-ключ защиты.

USB-ключ защиты - используется на серверах с операционными системами Windows, Linux. Работа не зависит от наличия или отсутствия сети Интернет на компьютере. USB-ключи могут использоваться как локально установленные, так и сетевые. Для корректной работы исполнительной системы с ключом защиты необходима установка драйвера.

Программный ключ используется на серверах с операционными системами Windows, Linux. Для работы ключа нужно будет провести его активацию, это возможно сделать если непосредственно компьютер, с установленной коммерческой версией, будет иметь доступ к сети Интернет. Если невозможно подключить защищаемый компьютер к Интернет, то можно воспользоваться любым другим компьютером, имеющим выход в Интернет.

В других случаях (для ряда контроллеров и, например, для ОС Эльбрус) для защиты программного продукта используется Активация. Без активации исполнительная система будет работать в демо-режиме. При подключении к среде исполнения, в редакторе (среде разработки) будет выдано сообщение, содержащее код активации, который необходимо отправить по электронному адресу **sales@masterscada.ru**. В ответ Вы получите файл для активации. Некоторые контроллеры, поставляются уже с выполненной активацией и могут сразу же приступить к работе.

Для выбора версии среды исполнения обратитесь по электронному адресу **sales@masterscada.ru**.

В комплект поставки программного обеспечения входит документ, в котором подробно описан процесс скачивания, инсталляции, активации и обновления программного обеспечения

Также может быть полезно:

Лицензионная политика

Выбор опций среды исполнения

Ошибки ключа защиты

1.7. СОВМЕСТИМОСТЬ С АНТИВИРУСАМИ И ФАЙРВОЛАМИ

MasterSCADA 4D совместима с антивирусами и файрволами, а также может работать под ограниченной учетной записью пользователя операционной системы, при условии, что при этом разрешена работа с портами, необходимыми для приема и отправки данных, имеется доступ к рабочим папкам **MasterSCADA 4D** с возможностью добавления и удаления файлов. В большинстве случаев достаточно выполнить настройки прав (разрешений), описанные ниже. В особых случаях могут потребоваться дополнительные разрешения.

Для среды разработки

- Назначить права на чтение и редактирование рабочей папки: *c:\Users\Public\Documents\MasterSCADA4D1.2*.
- Назначить права на чтение и редактирование папки, в которой хранятся подготовленные для загрузки в среду исполнения проекты: *c:\Users\[имя пользователя]\AppData\Roaming\MPSSoft*.

- Разрешить запуск процессов `mplc`, `mplc_service`, `nginx`, клиент визуализации, `node_ms4d`.
- Для использования удаленных OPC DA/HDA серверов следует настроить **DCOM Windows**.
- Открыть нужный TCP-порт для добавления тегов OPC UA (определяется настройками).

Для среды исполнения

- Назначить права на чтение и редактирование папки, из которой запускается среда исполнения, по умолчанию это `c:\Users\[имя пользователя]\AppData\Roaming\MPSSoft\MasterSCADA4DRT1.2`.
- Разрешить TCP-порт для работы с веб-server, по умолчанию это 8043 или тот, который указан в настройках **MasterSCADA 4D Monitor**
- Для работы модуля отчетов в режиме исполнения необходимо сделать доступным порт TCP 9615.
- Для использования удаленных OPC DA/ HDA серверов следует настроить **DCOM Windows**.
- Открыть нужный порт TCP для опроса тегов OPC UA.
- Если **MasterSCADA 4D** выступает в роли OPC UA сервера, то необходимо разрешить входящий TCP-порт, по умолчанию это 16550 (может быть изменен в проекте в настройках узла).
- Если используется передача данных от **MasterSCADA 4D** в другие системы через Modbus TCP Slave, то необходимо разрешить входящий TCP-порт, по умолчанию 502 (может быть изменен в настройках проекта у элемента Внешние каналы).
- Для подключения среды разработки к исполнительной системе, для межузловой связи и для синхронизации серверов при резервировании используется UDP-порт 30550. Если на одном компьютере запущено одновременно несколько экземпляров исполнительной системы, то номер порта будет `30550 + <номер экземпляра mplc>`
- Для синхронизации серверов при резервировании кроме UDP-порта используется TCP-порт 30550, если на одном компьютере запущено одновременно несколько экземпляров исполнительной системы, то номер порта будет `30550 + <номер экземпляра mplc>`. Т.е. используется тот же номер порта, что и для UDP
- Для получения лога служебных сообщений исполнительной системы используется TCP-порт 31550. Если на одном компьютере запущено несколько экземпляров исполнительной системы, то номер порта будет `31550 + <номер экземпляра mplc>`
- Для взаимодействия веб-сервера и исполнительной системы используется TCP-порт 30750. Если на одном компьютере запущено несколько экземпляров

исполнительной системы, то номер порта будет $30750 + \langle \text{номер экземпляра mpls} \rangle$

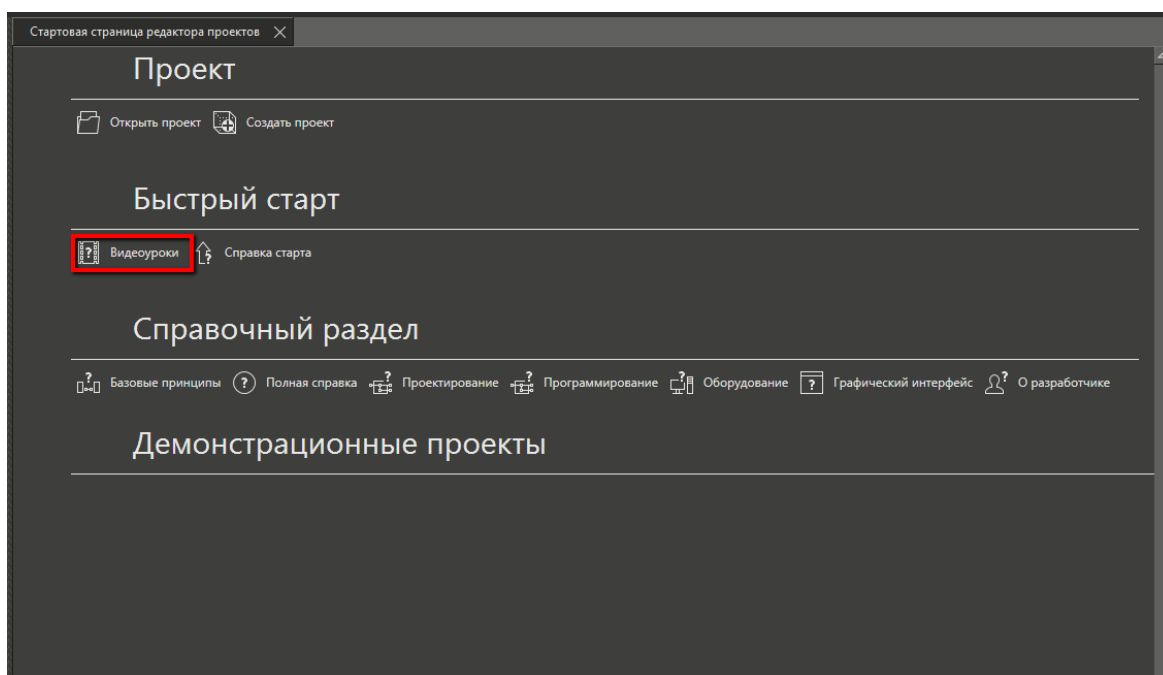
Для клиента визуализации

В большинстве случаев специальных настроек прав (разрешений) не требуются.

2. БЫСТРЫЙ СТАРТ

Цель Быстрого старта (далее – БС) – помочь тем, кто делает первые шаги в освоении **MasterSCADA 4D**, и пока ещё не имеет опыта работы с этой средой.

Данный материал повторяет видеоуроки, размещенные на нашем Youtube-канале. Ссылку на этот канал можно найти на стартовой странице среды разработки.



2.1. БС. УРОК 1

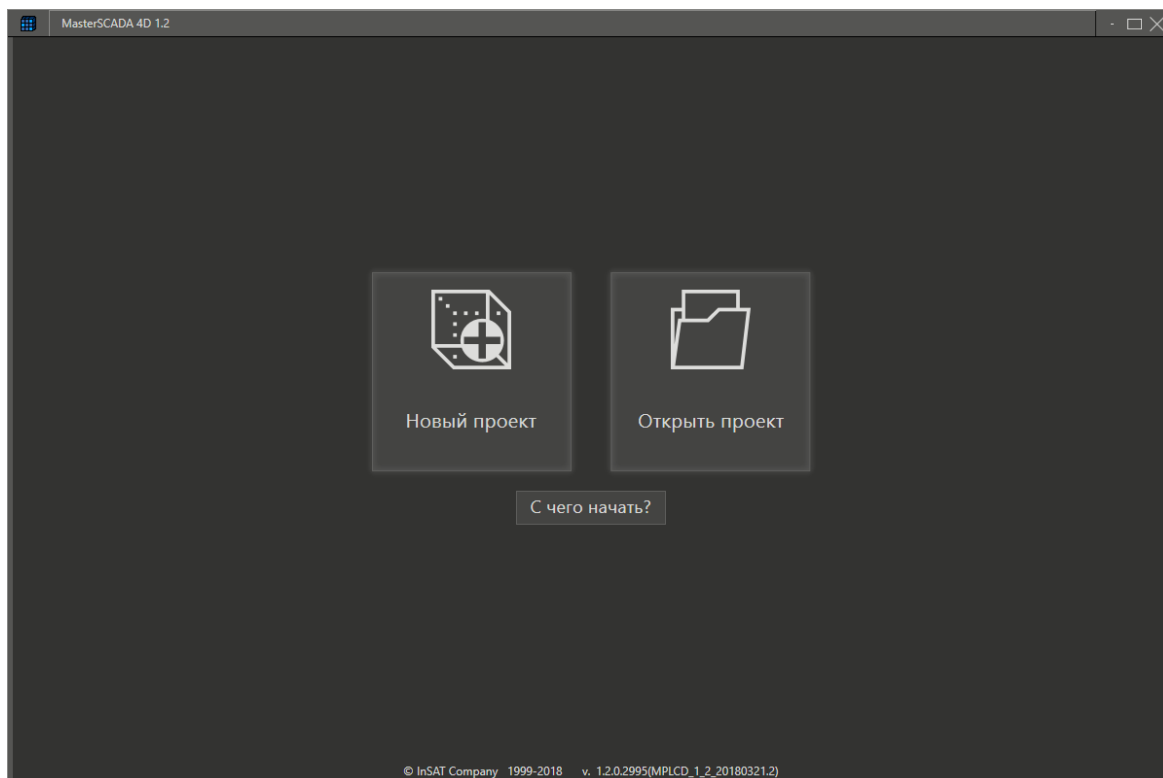
Рассмотрим создание проекта, реализующего управление включением и выключением устройства двумя кнопками – Пуск и Стоп. При нажатии на кнопку Пуск устройство должно включаться. При нажатии на кнопку Стоп устройство должно выключаться.

- **БС. Урок 1. Создание проекта**
- **БС. Урок 1. Конфигурирование дерева системы**
- **БС. Урок 1. Создание логической части проекта**
- **БС. Урок 1. Создание окна управления**
- **БС. Урок 1. Запуск**

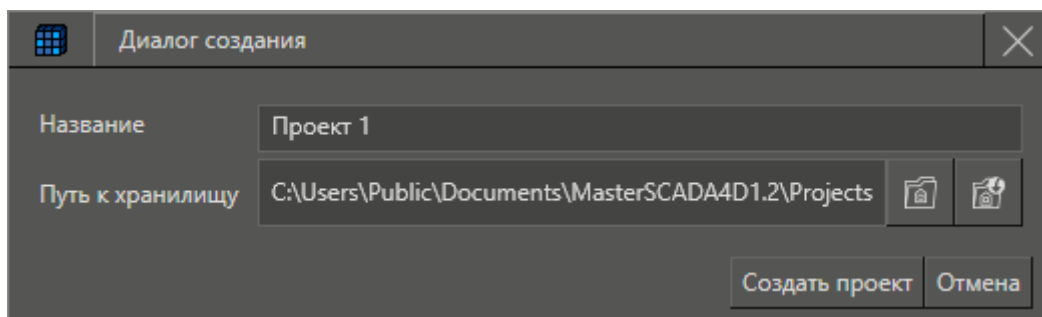
2.1.1. БС. УРОК 1. СОЗДАНИЕ ПРОЕКТА

Для того чтобы начать создание проекта, нам необходимо запустить среду разработки **MasterSCADA 4D**,

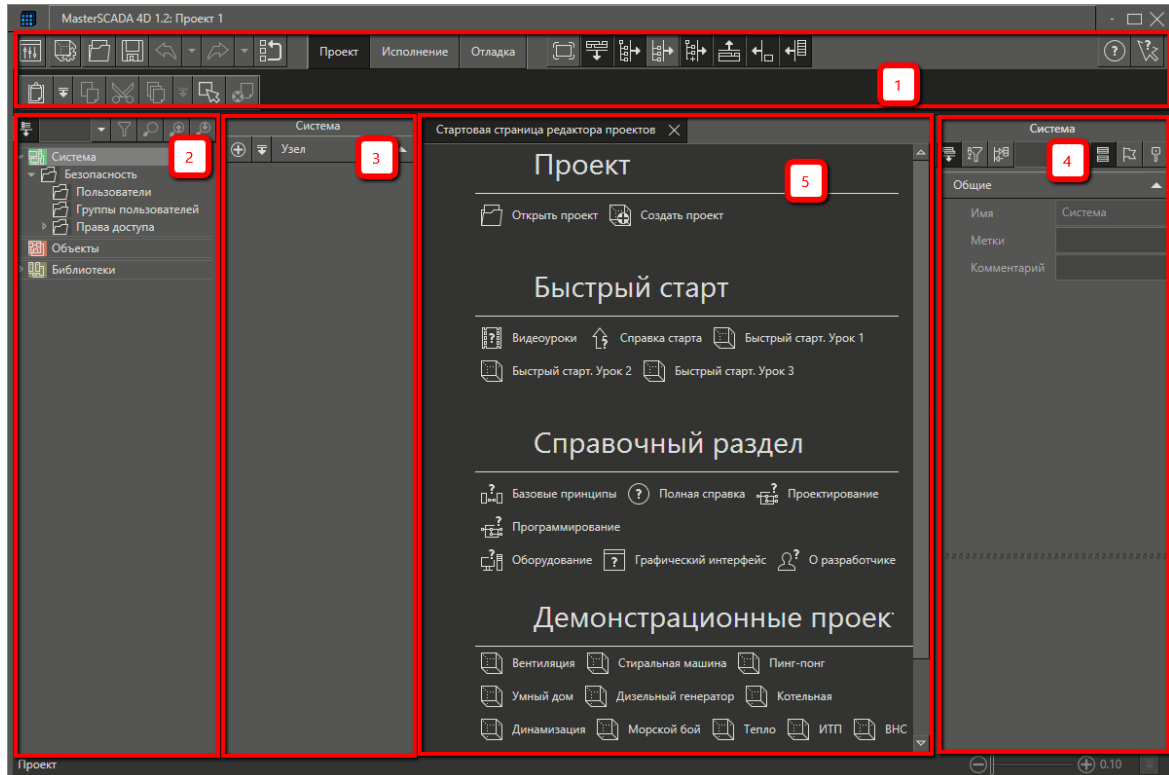
выполнив пункт меню **Windows Пуск\Все программы\MPSSoft\MasterSCADA 4D\MasterSCADA 4D**.



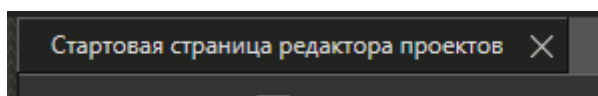
После запуска **MasterSCADA 4D** нам будет предложено либо создать новый проект, либо открыть существующий. Создадим новый проект.



По умолчанию, новому проекту будет присвоено имя Проект 1. Оставим его без изменения - это будет наш самый первый проект в **MasterSCADA 4D**.

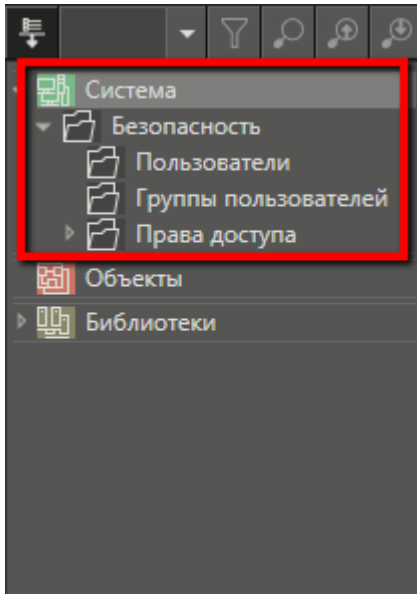


Интерфейс среды разработки состоит из нескольких элементов. В верхней части интерфейса располагается панель инструментов (1). Слева – дерево проекта (2) и контекстная панель (3). В правой части – панель свойств (4). В центральной части интерфейса располагается поле (панель) редакторов (5). В этом поле после запуска отображается стартовая страница с иконками для открытия демонстрационных и учебных проектов. Эту стартовую страницу можно закрыть:

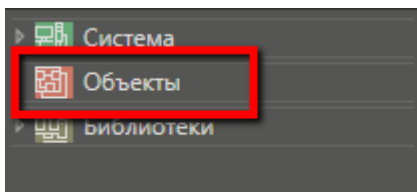


Дерево проекта состоит из трёх частей.

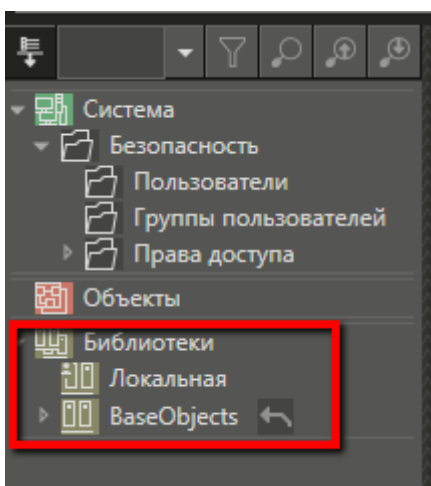
Первая часть – дерево системы. В нём задаётся физическая конфигурация Вашей системы, т.е. указываются устройства, протоколы, каналы связи.



Вторая часть – дерево объектов. В этом дереве создаётся логическая структура проекта. По умолчанию, оно не содержит дочерних элементов.



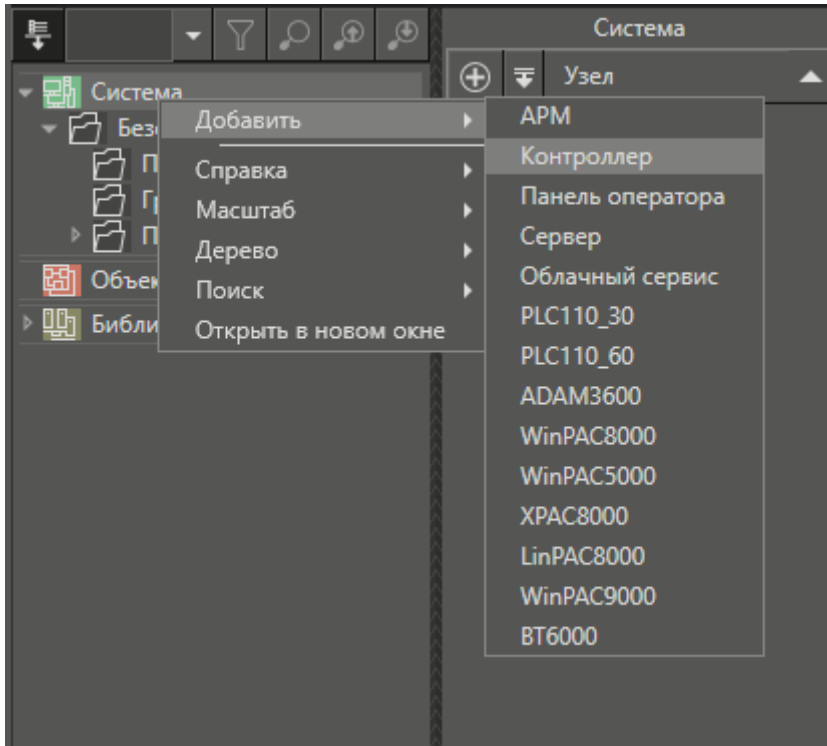
Третья часть дерева содержит библиотеки, как поставляемые со средой разработки, так и созданные пользователем.



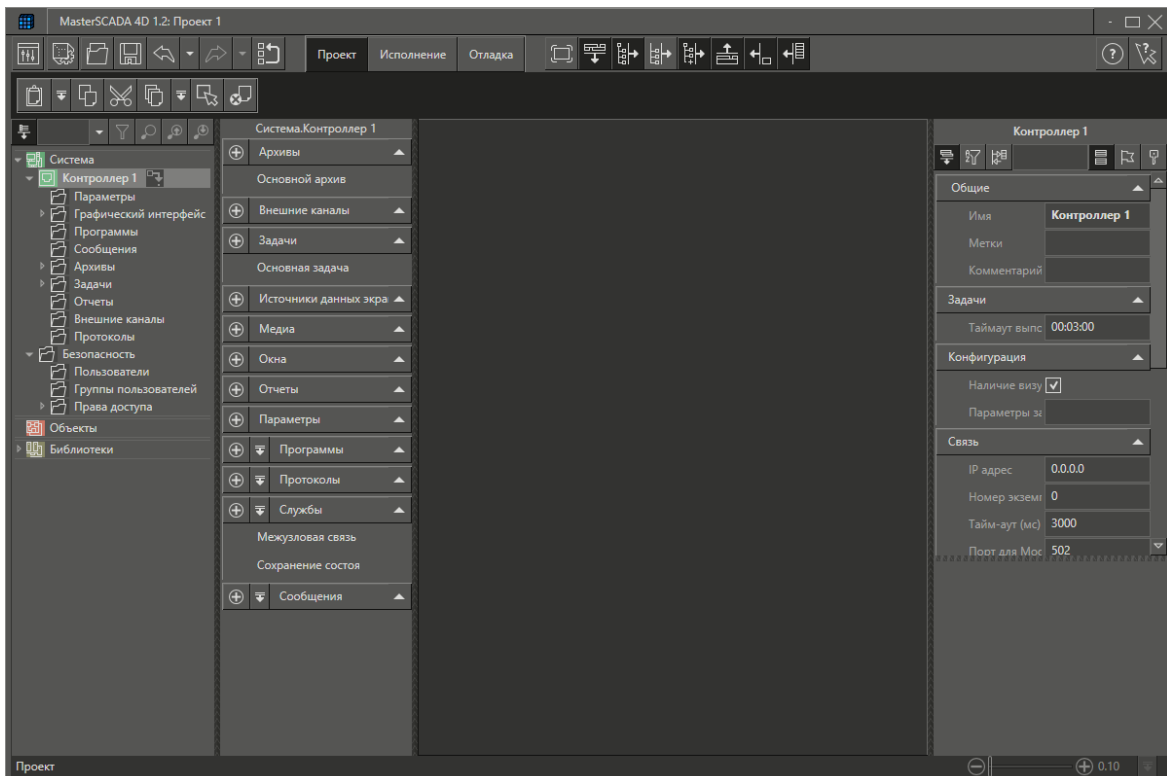
2.1.2. БС. УРОК 1. КОНФИГУРИРОВАНИЕ ДЕРЕВА СИСТЕМЫ

Начнём построение проекта с конфигурирования дерева системы.

Для этого нажмем правой кнопкой мыши на любое место дерева системы, и из контекстного меню выберем устройство, на котором будет исполняться наш проект. Выберем просто Контроллер.

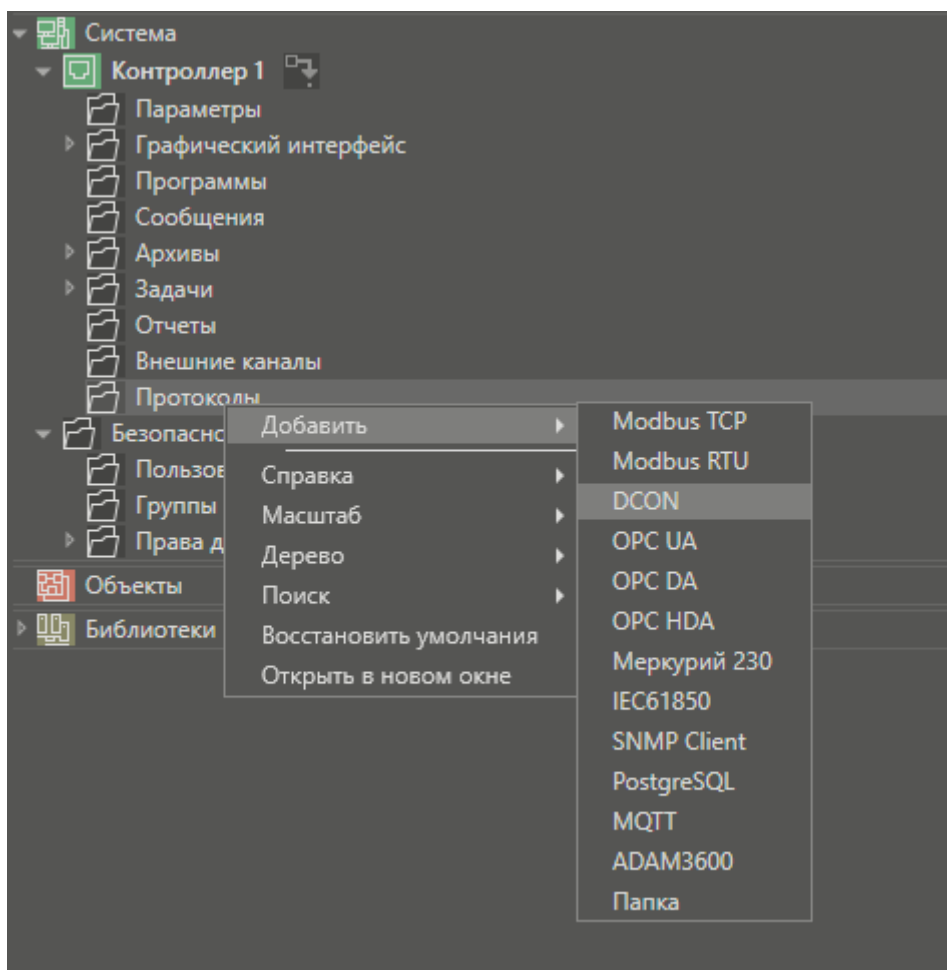


В дерево добавится Контроллер, содержащий свойственные для него элементы:

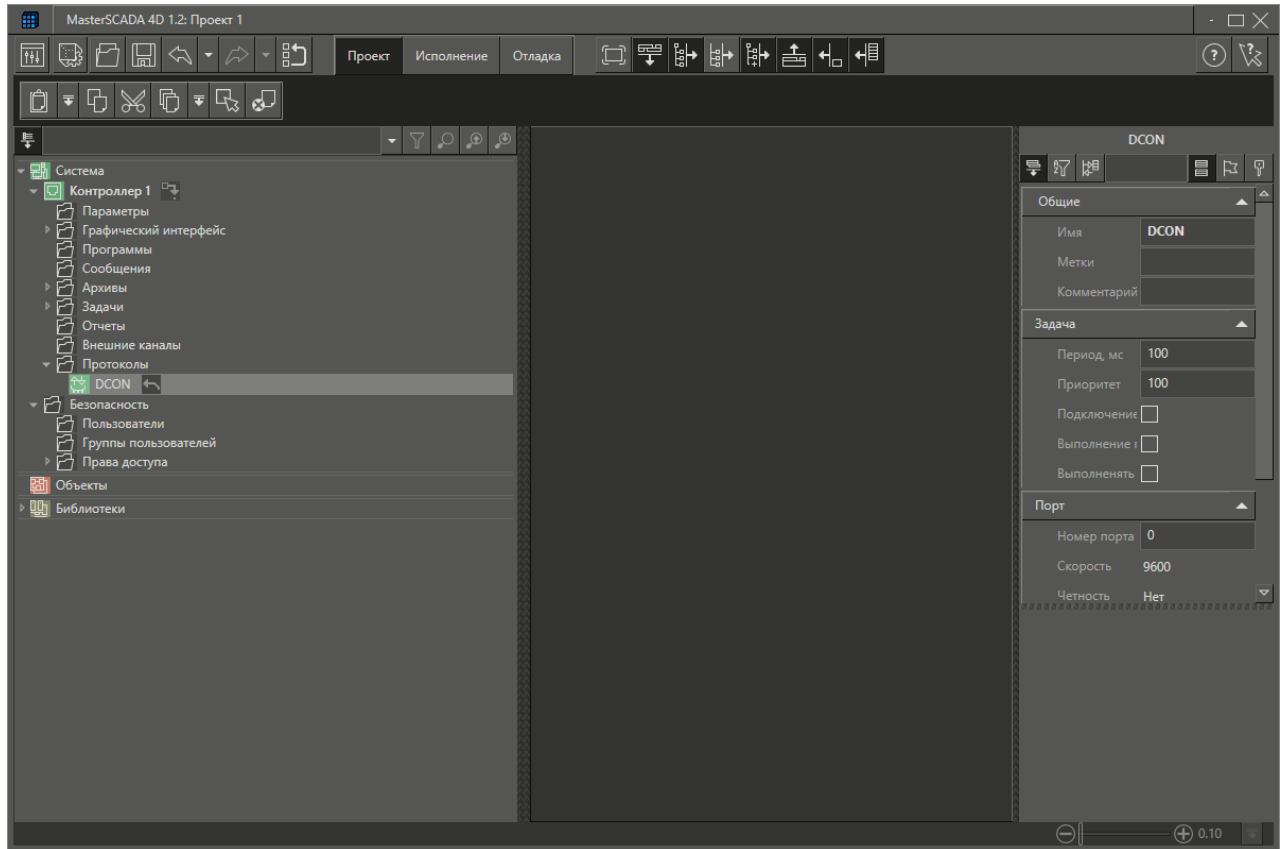


В рамках поставленной задачи нам потребуется один дискретный выход для управления устройством. Некоторые контроллеры имеют встроенные входы/выходы. Однако, в выбранном контроллере таких выходов нет, и мы будем использовать внешний модуль дискретного выхода, подключенный к контроллеру по последовательному порту.

Для этого нажмем правой кнопкой мыши на элемент Протоколы, и из контекстного меню выберем тот протокол, который будем использовать. Пусть, в данном случае, это будет протокол DCON.



В результате получим:

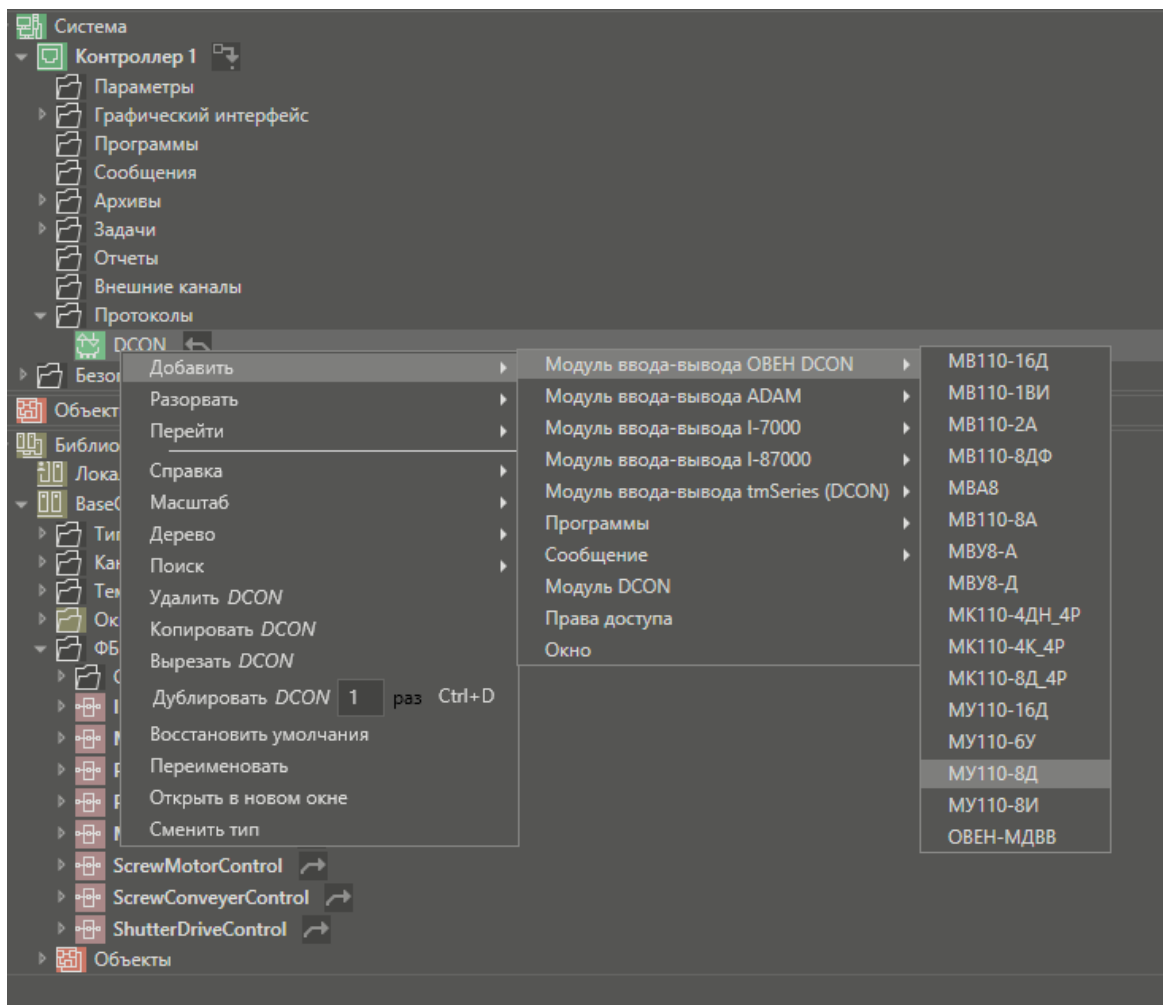


Теперь на панели свойств следует указать номер порта, к которому подключены модули. В нашем случае это порт 4 (настройка зависит от реально подключенного к стенду оборудования). Здесь же необходимо указать скорость обмена, а также то, ведется ли контроль чётности и подсчет контрольной суммы (последние параметры зависят от конфигурации модулей).

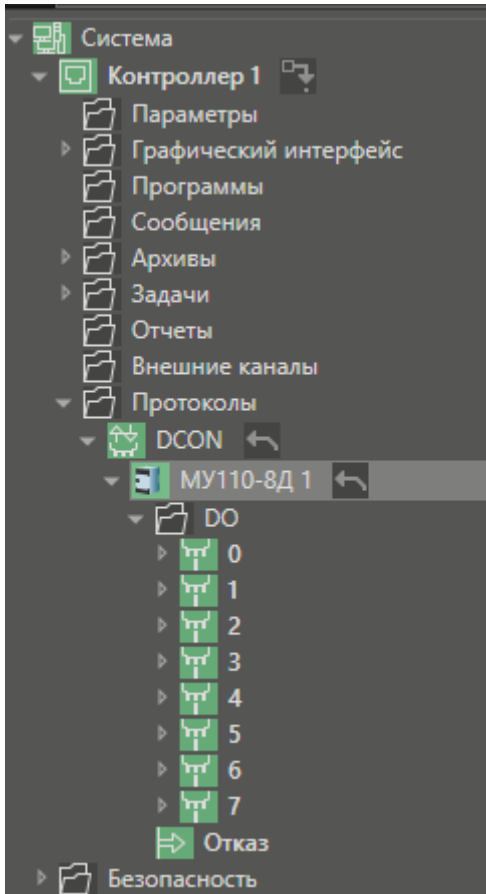
DCON	
Общие	
Имя	DCON
Метки	
Комментарий	
Задача	
Период, мс	100
Приоритет	100
Подключение по условию	<input type="checkbox"/>
Выполнение по условию	<input type="checkbox"/>
Выполнять на резервной	<input type="checkbox"/>
Порт	
Номер порта	4
Скорость	9600
Четность	Нет
Стоп бит	1
Разрядность символа	8
Протокол	
Таймаут	100
Контрольная сумма	<input type="checkbox"/>

Далее нам требуется указать сам модуль.

Нажмем правой кнопкой мыши на протокол DCON, и из контекстного меню выберем Добавить - Модуль Ввода Вывода Овен - МУ110-8Д - это модуль дискретных выходов, одним из которых мы и будем управлять.



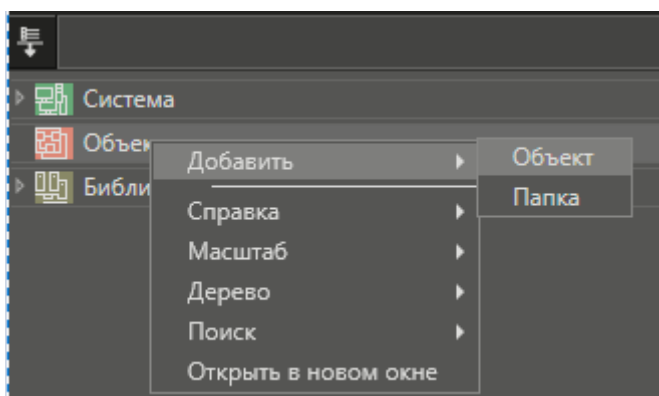
При этом в дерево системы добавится соответствующий элемент, который по умолчанию уже будет иметь правильный набор каналов.

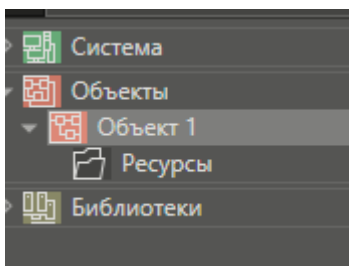


На этом конфигурирование физической структуры системы закончено. Перейдём к созданию логической части проекта.

2.1.3. БС. УРОК 1. СОЗДАНИЕ ЛОГИЧЕСКОЙ ЧАСТИ ПРОЕКТА

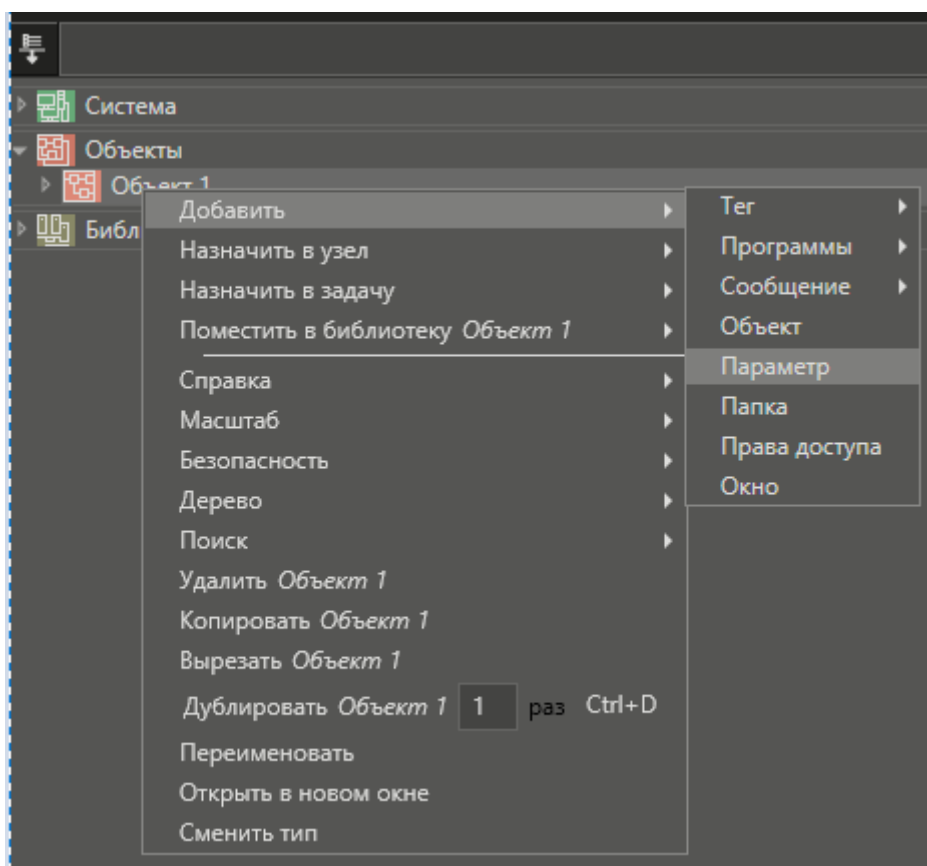
Для начала вызовем контекстное меню дерева объектов, нажав правую кнопку мыши, и выполним пункт Добавить - Объект.



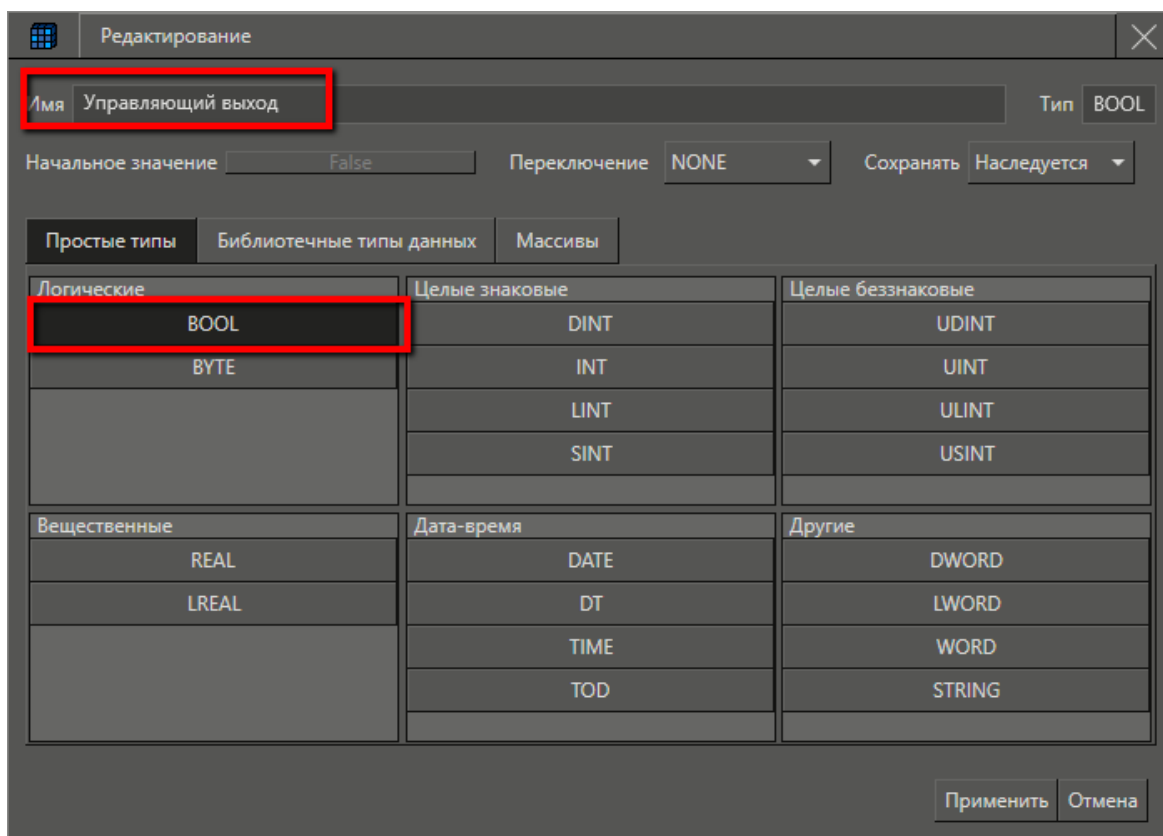


Помимо параметров, в объект можно добавлять программы для обработки данных, графические окна и другие элементы.

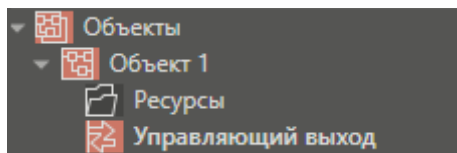
В рамках поставленной задачи у нас будет один управляющий сигнал. Нажмем правой кнопкой мыши на объект, и из контекстного меню выберем Добавить - Параметр.



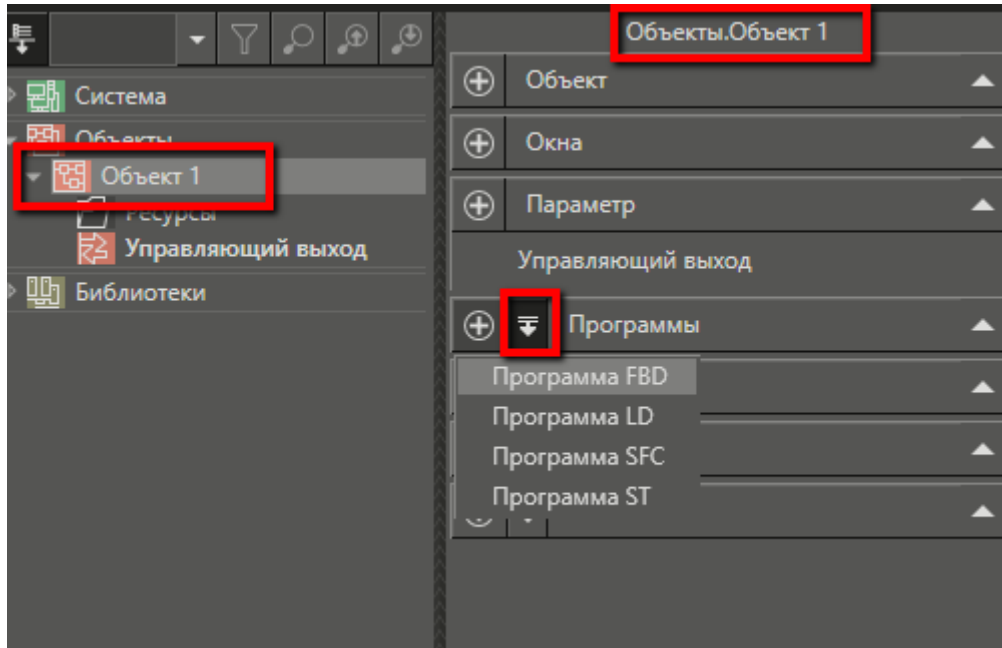
В открывшемся окне свойств параметра зададим ему имя, например, Управляющий Выход и укажем тип параметра – BOOL (логический).



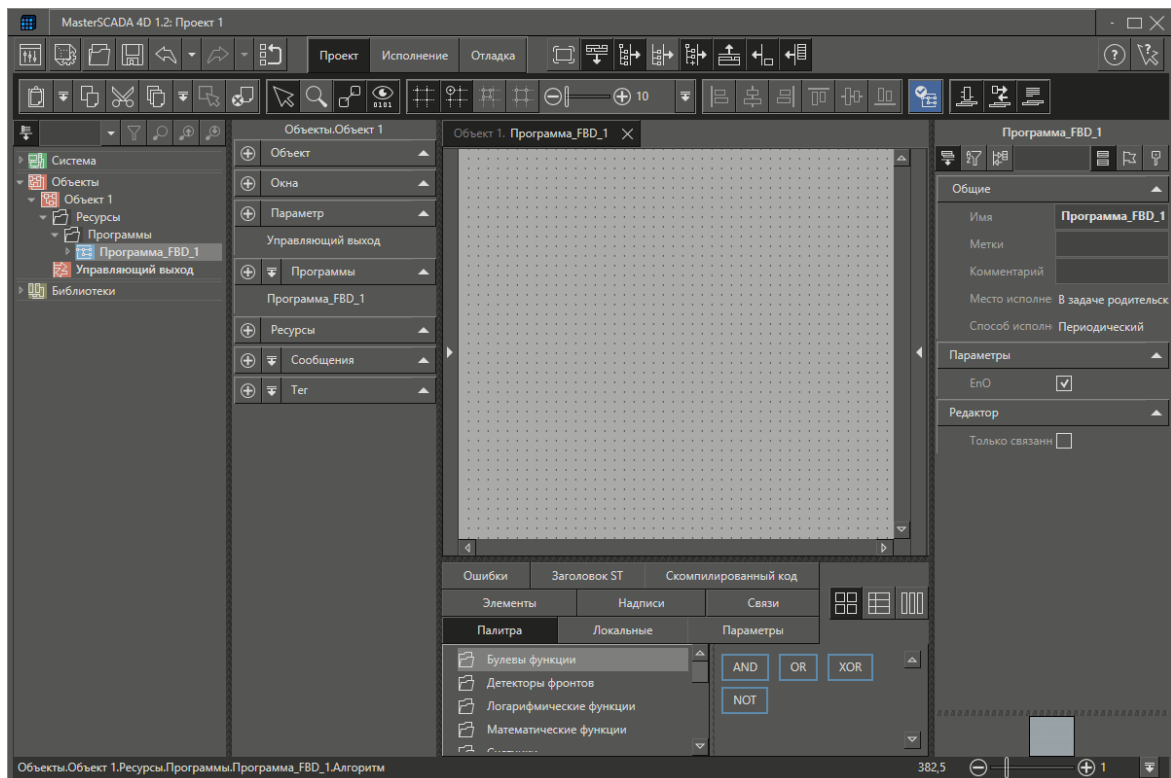
В результате получим:



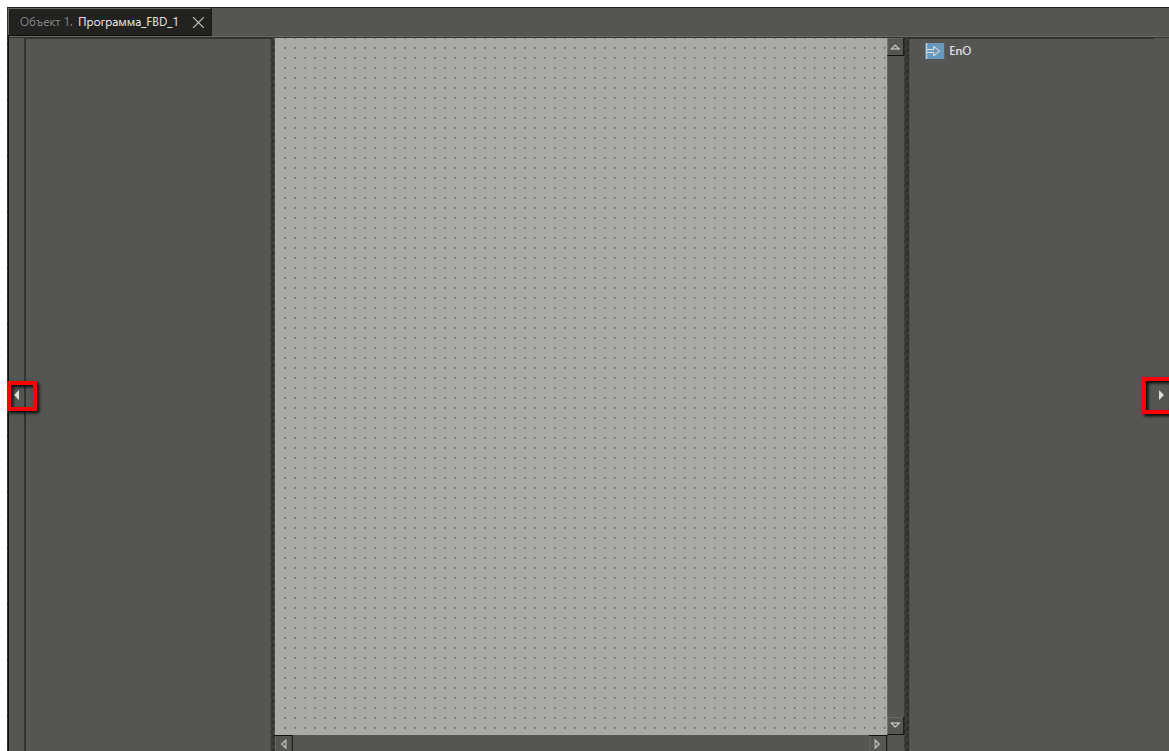
Теперь создадим управляющую программу. Для этого воспользуемся контекстной панелью элемента дерева Объект 1. На рисунке ниже показано, что данная панель дает возможность добавить программу на одном из четырех языков стандарта МЭК 61131-3. Выберем программу на языке функциональных блоков – FBD.



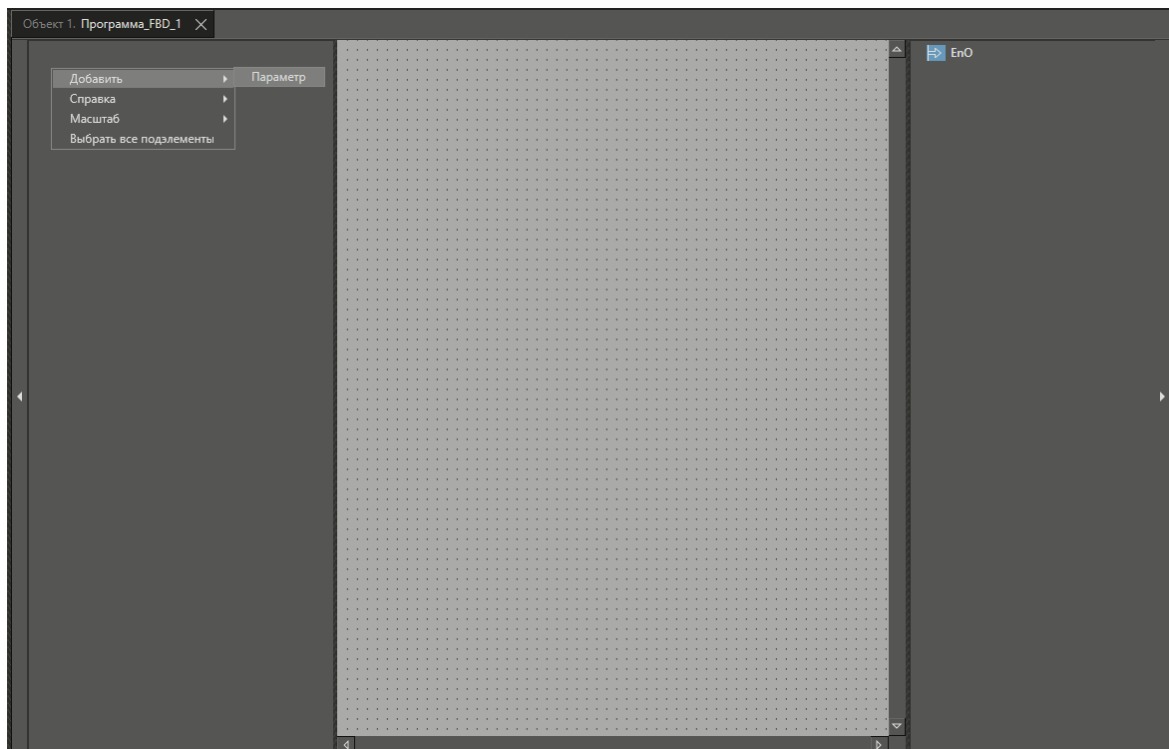
При этом откроется окно редактора этого языка.



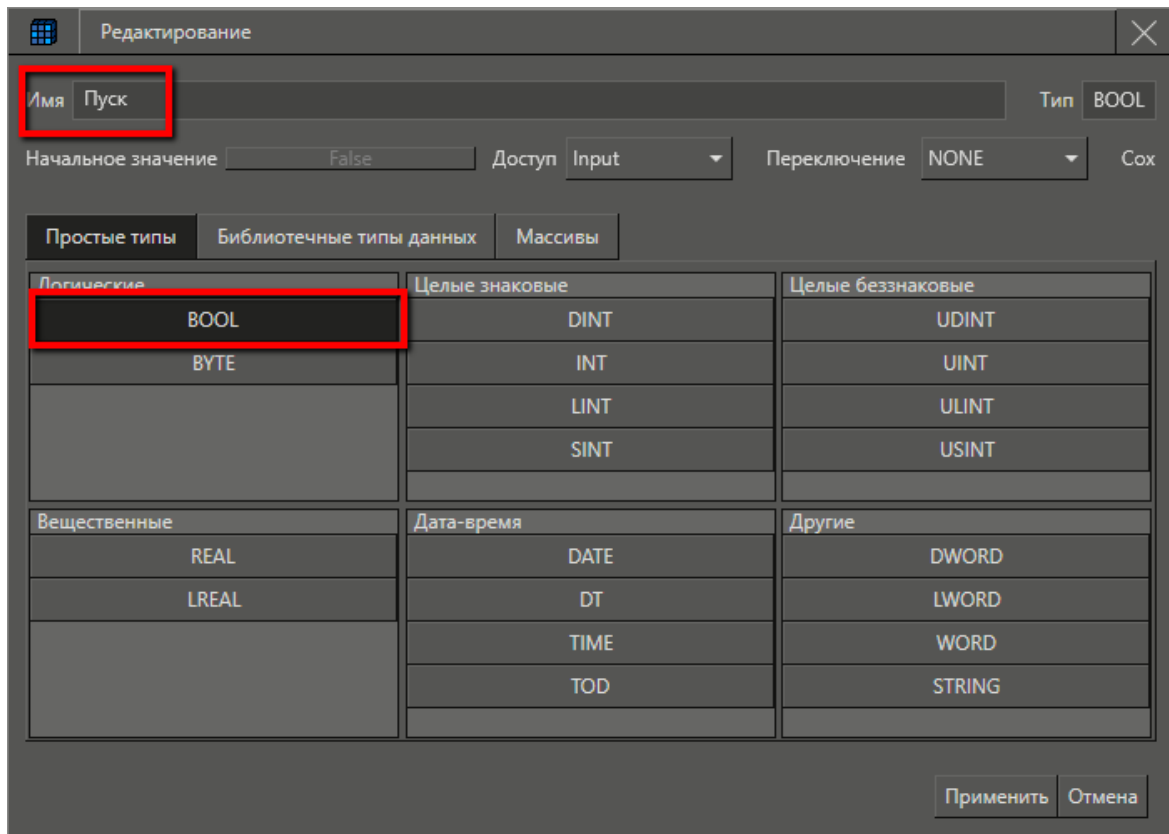
Слева и справа от поля редактирования, по аналогии с электрическими схемами, расположены клеммники. На клеммнике слева находятся входные сигналы, а на клеммнике справа – выходные. Для того чтобы развернуть или свернуть клеммники, необходимо нажать на стрелки, расположенные по краям поля редактирования.



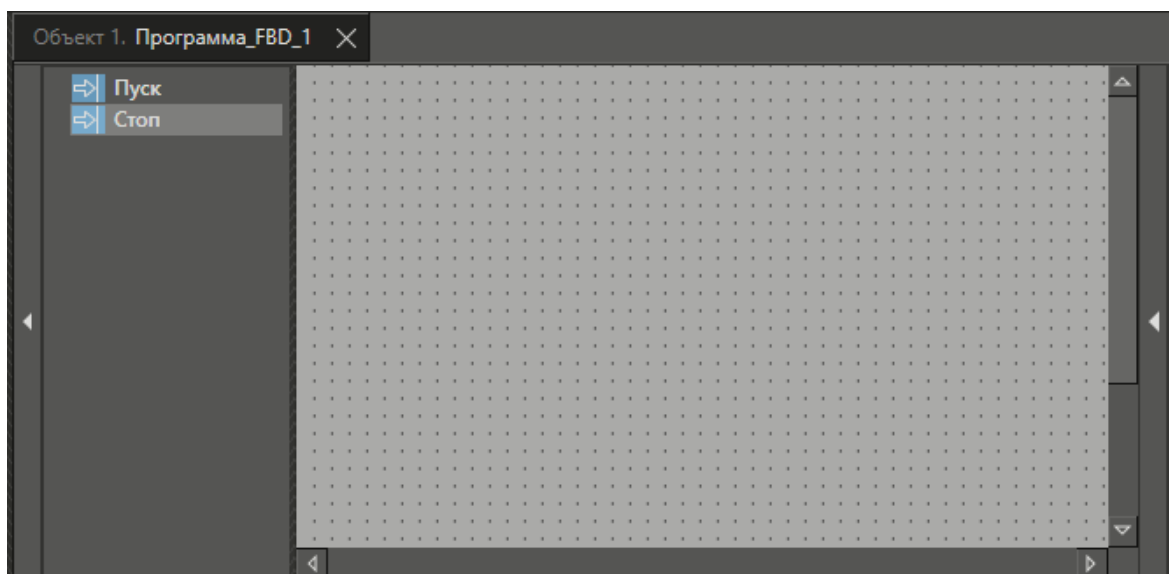
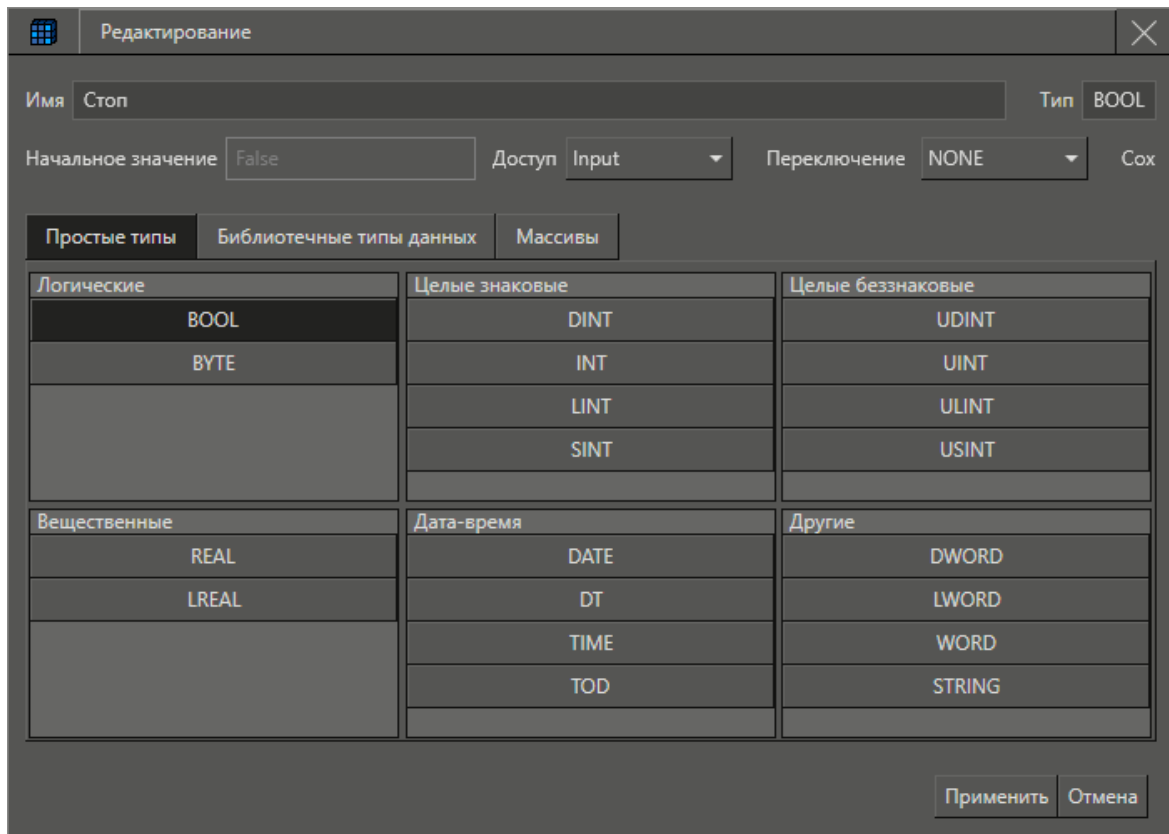
Создадим в программе два входных параметра: Пуск и Стоп . Для этого нажмем правой кнопкой мыши в любое место входного клеммника, и из контекстного меню выберем пункт Добавить - Параметр.



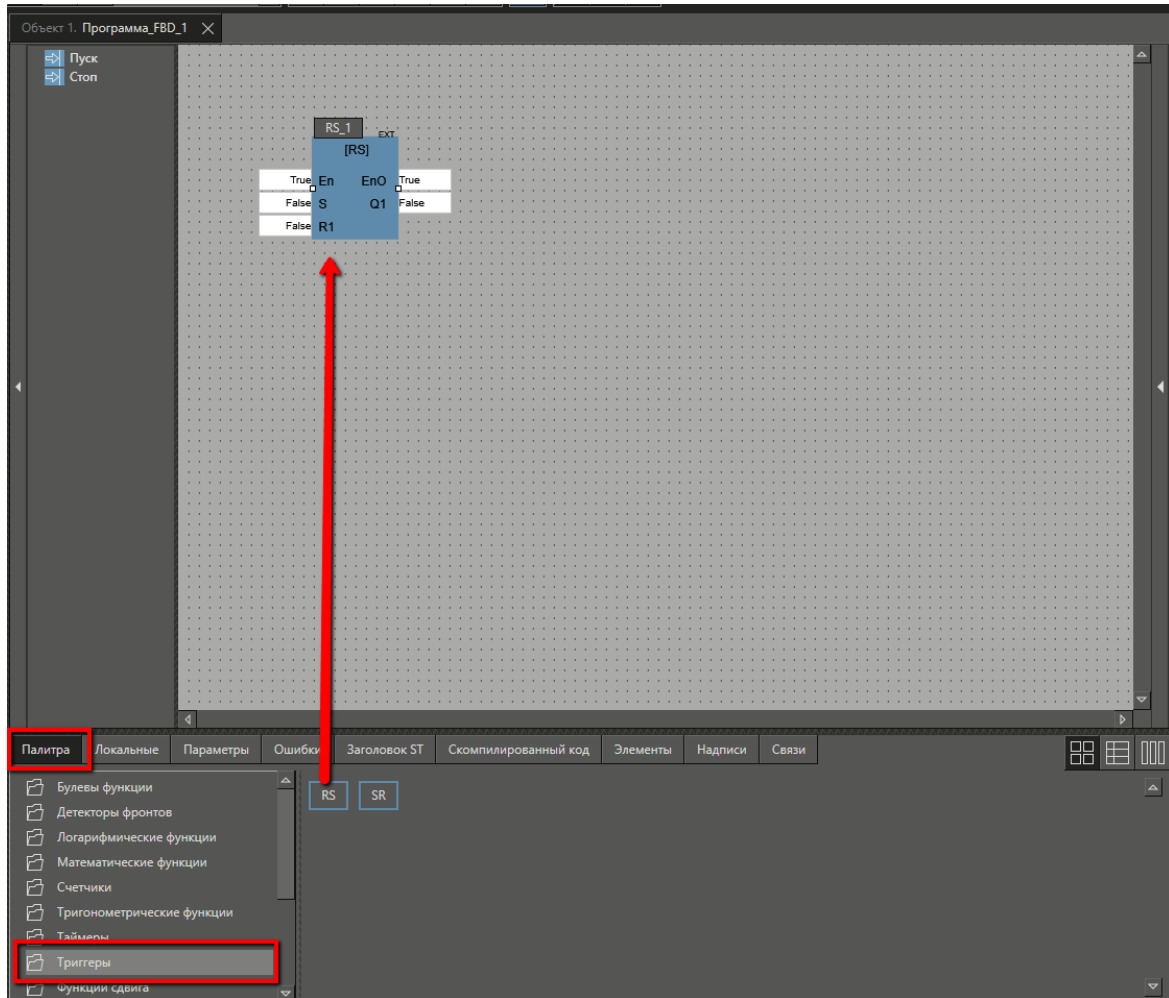
В открывшемся окне свойств параметра изменим его имя на Пуск, а тип – на BOOL.



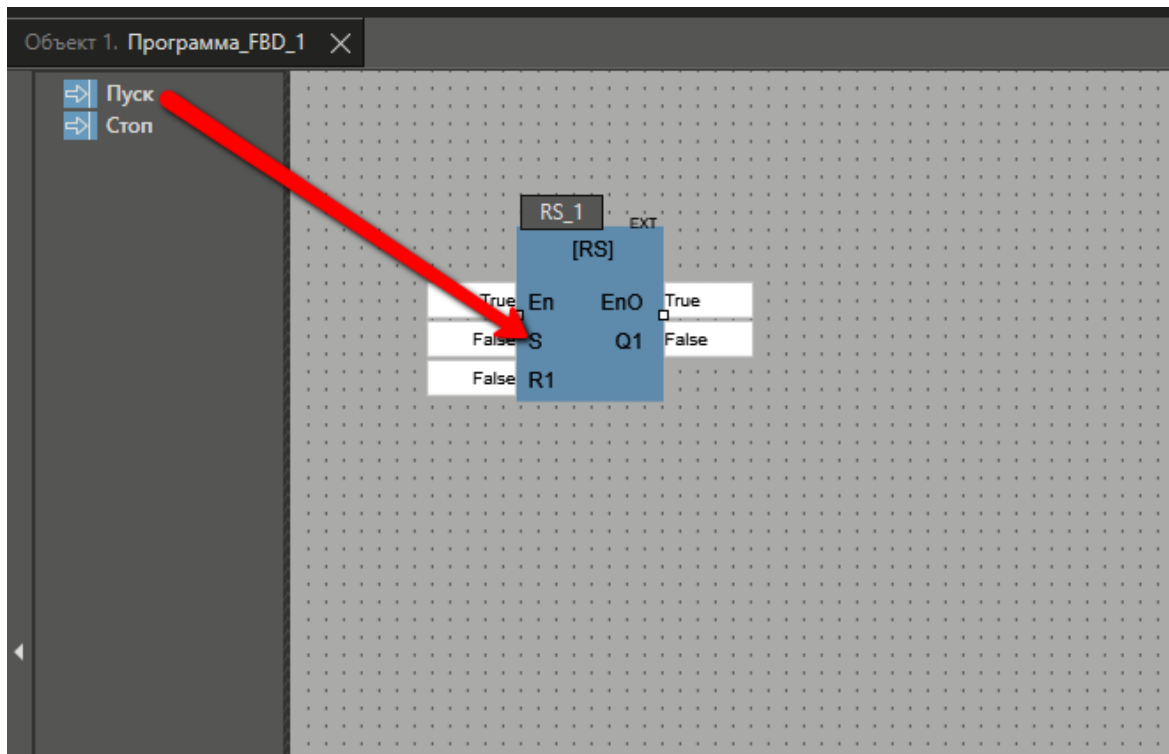
Аналогично создадим параметр Стоп.



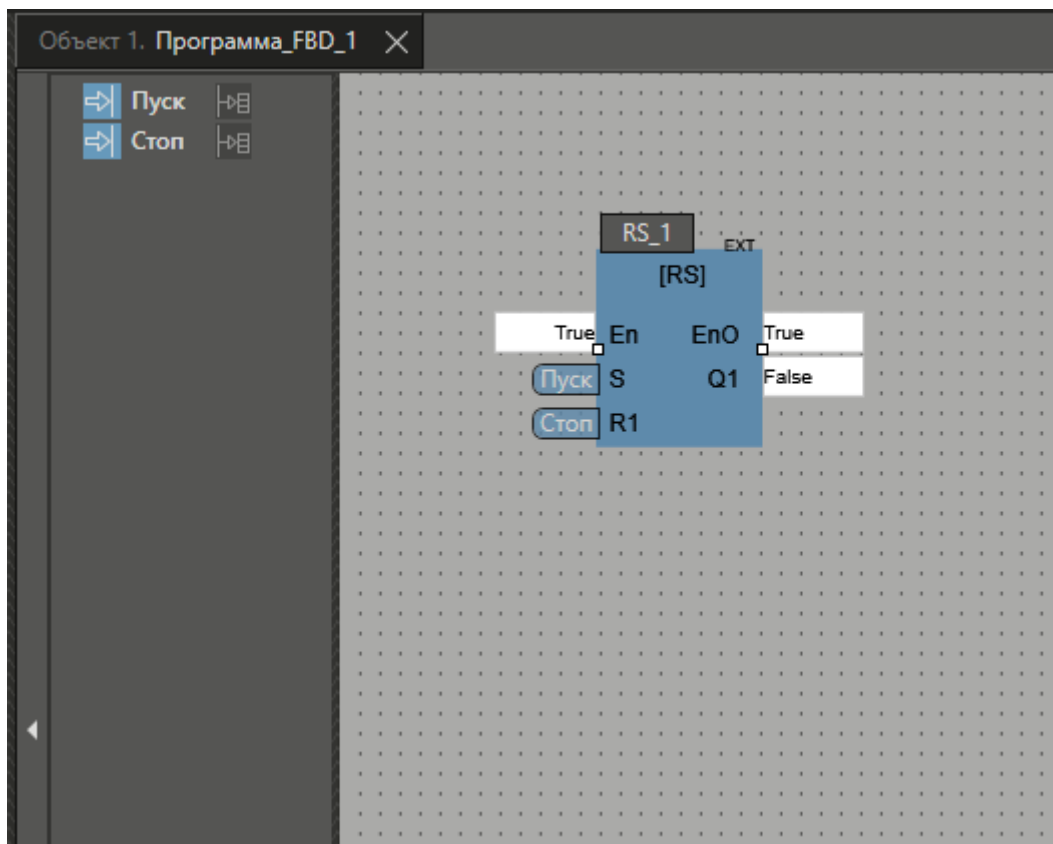
Для формирования логики включения используем обычный RS-триггер. Для того чтобы вставить на схему функциональный блок, достаточно просто перетащить его из Палитры в поле редактора. В Палитре найдем раздел Триггеры и перетащим элемент RS на схему.



Теперь необходимо подключить входы программы к входам RS-триггера (или, как ещё говорят, "связать" или "установить связь"). Для этого нужно нажать левой кнопкой мыши над названием параметра (например, Пуск), и, удерживая кнопку мыши нажатой, перенести указатель на тот параметр, с которым Вы хотите его связать (S), и отпустить над ним кнопку мыши.

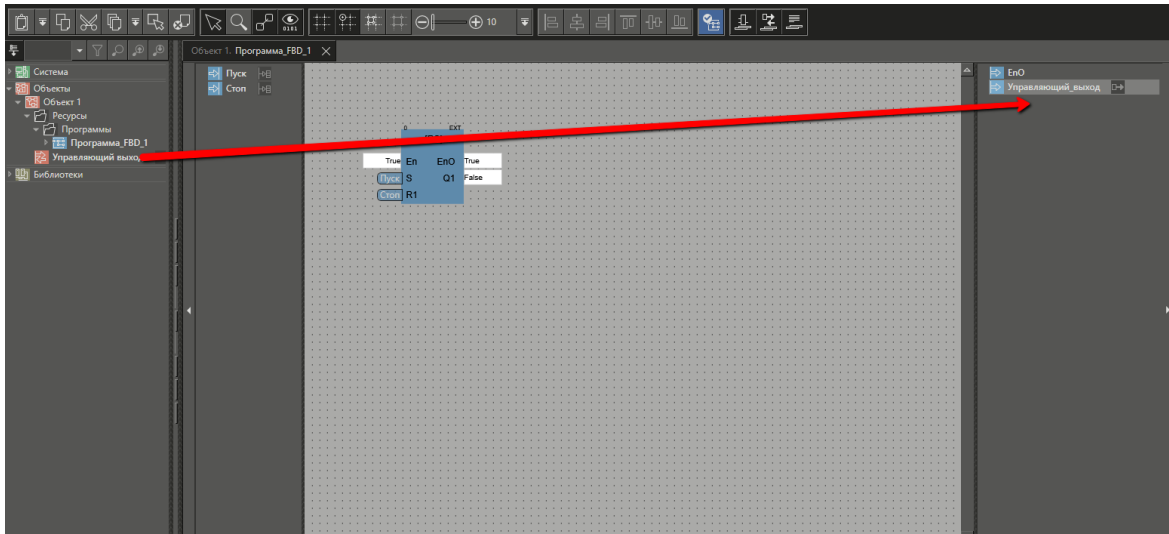


Аналогичным способом свяжем вход программы Стоп с входом триггера R1. И получим результат:

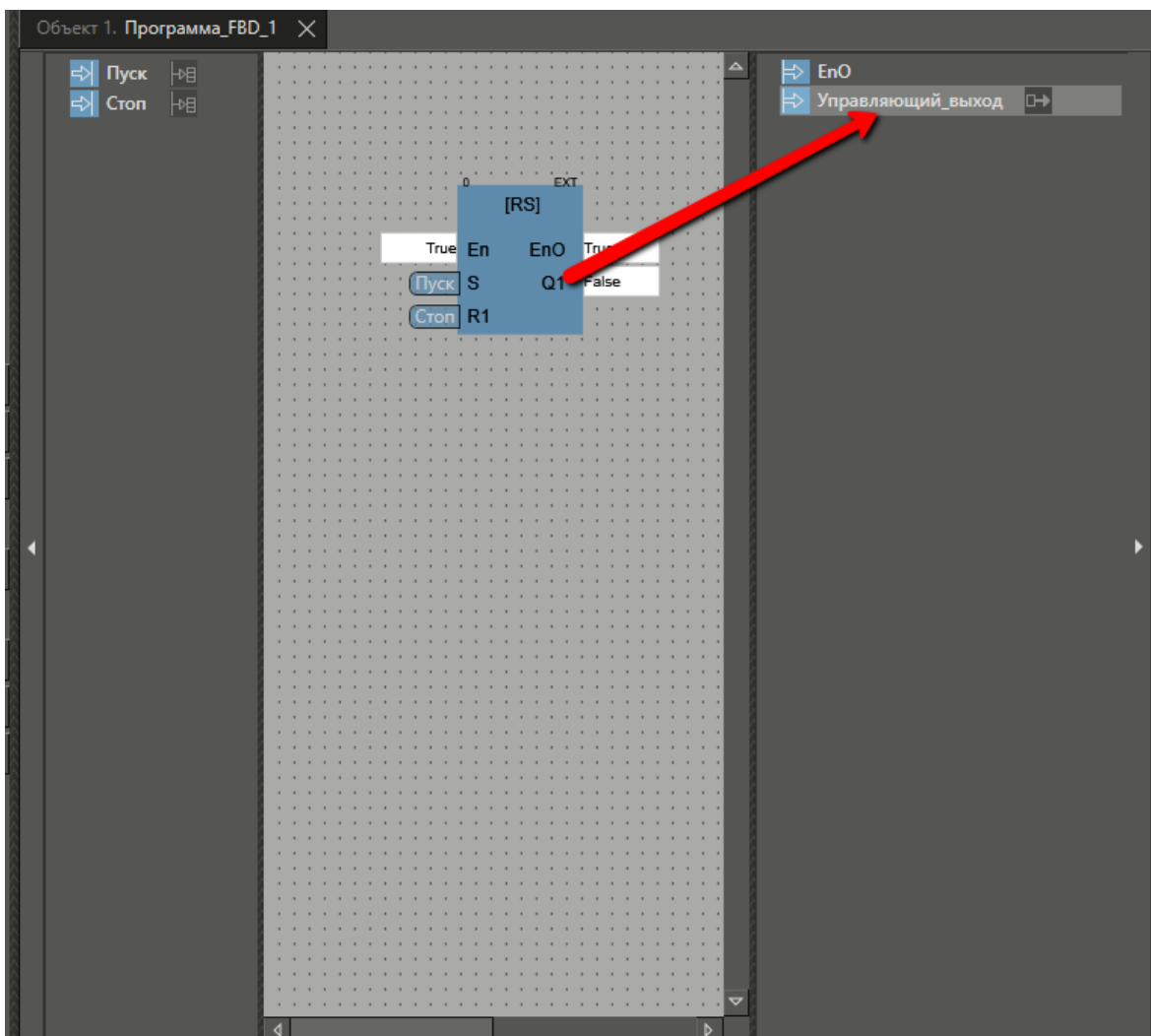


Мы ещё вернёмся к входным параметрам программы, а сейчас создадим выходной параметр.

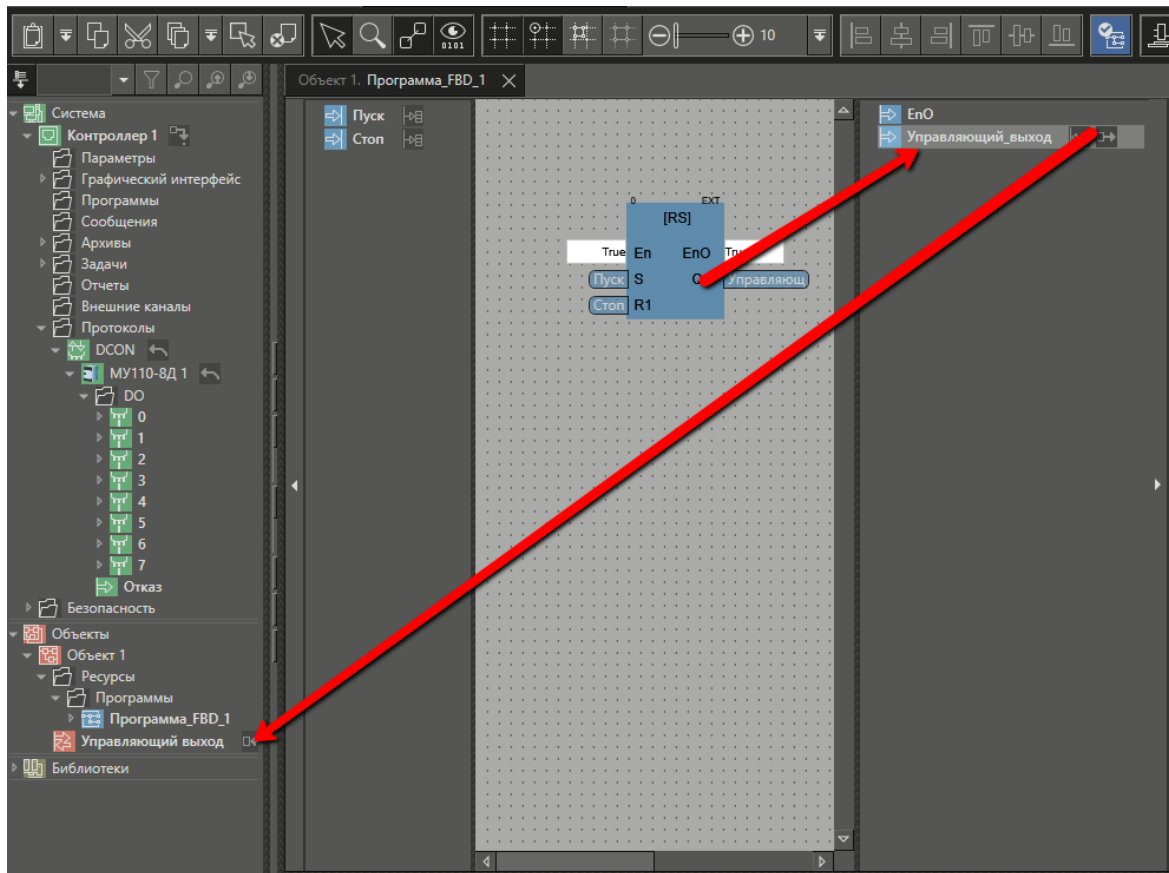
Для этого перетащим параметр объекта Управляющий Выход на выходной клеммник программы, и в результате будет создан выходной параметр программы (по умолчанию – одноимённый), а также связь с ним.



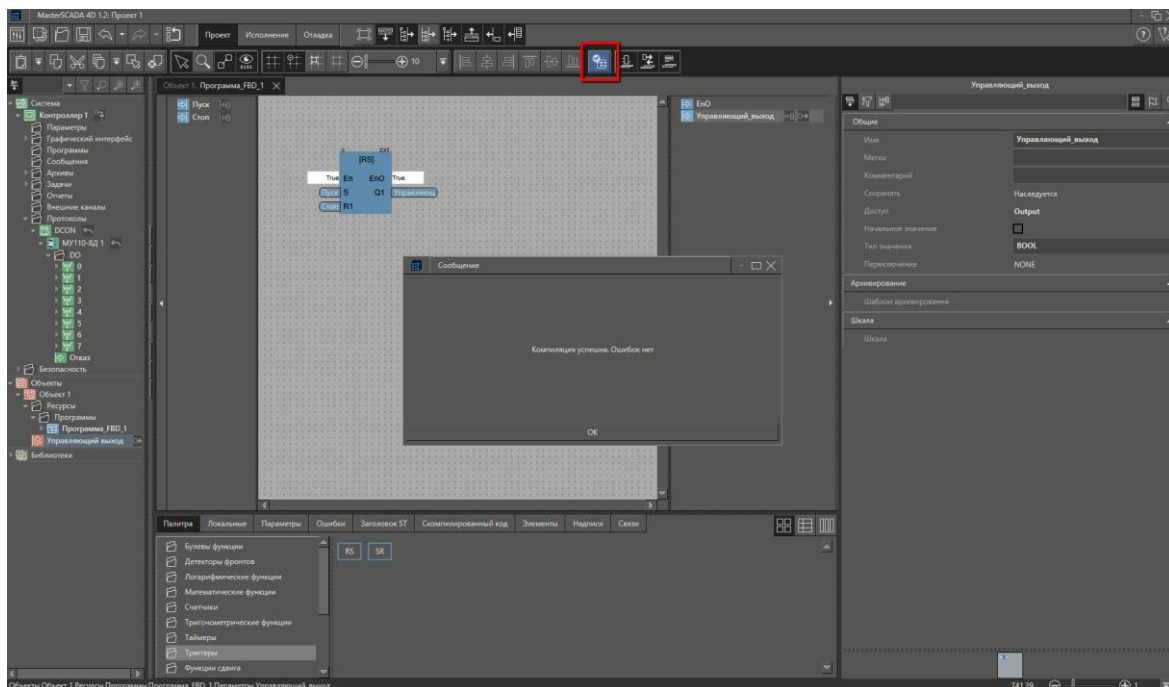
Наконец, свяжем выход триггера Q1 с выходным параметром Управляющий Выход.



Таким образом, мы сформировали выходной тракт от выхода триггера Q1 к параметру Управляющий Выход объекта.



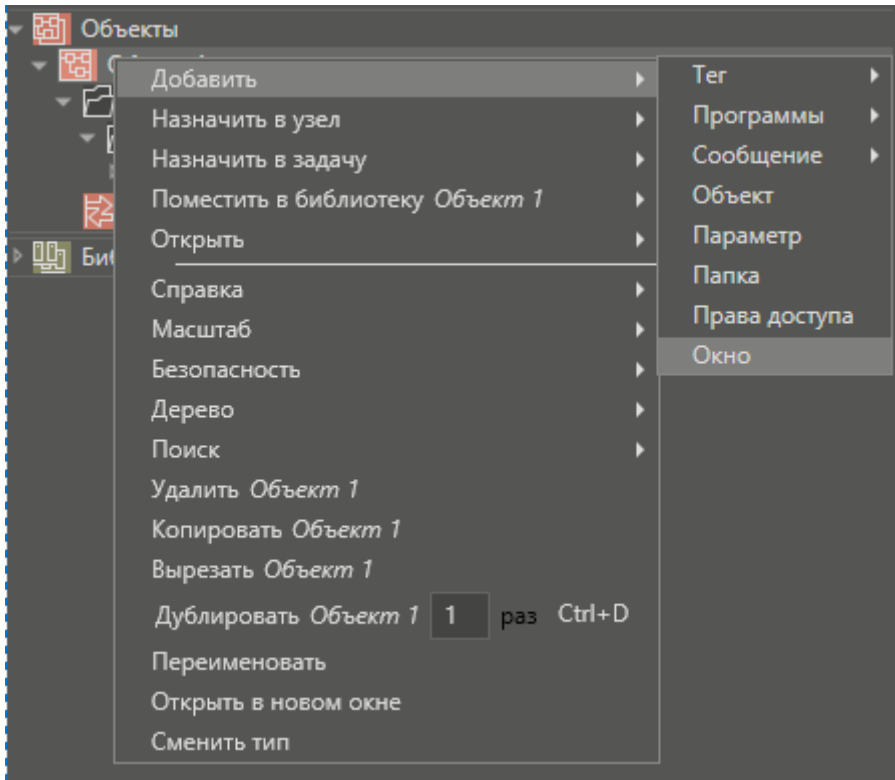
На этом формирование логики управления закончено. На всякий случай, нажмём кнопку Проверить и убедимся, что компиляция пройдет успешно.



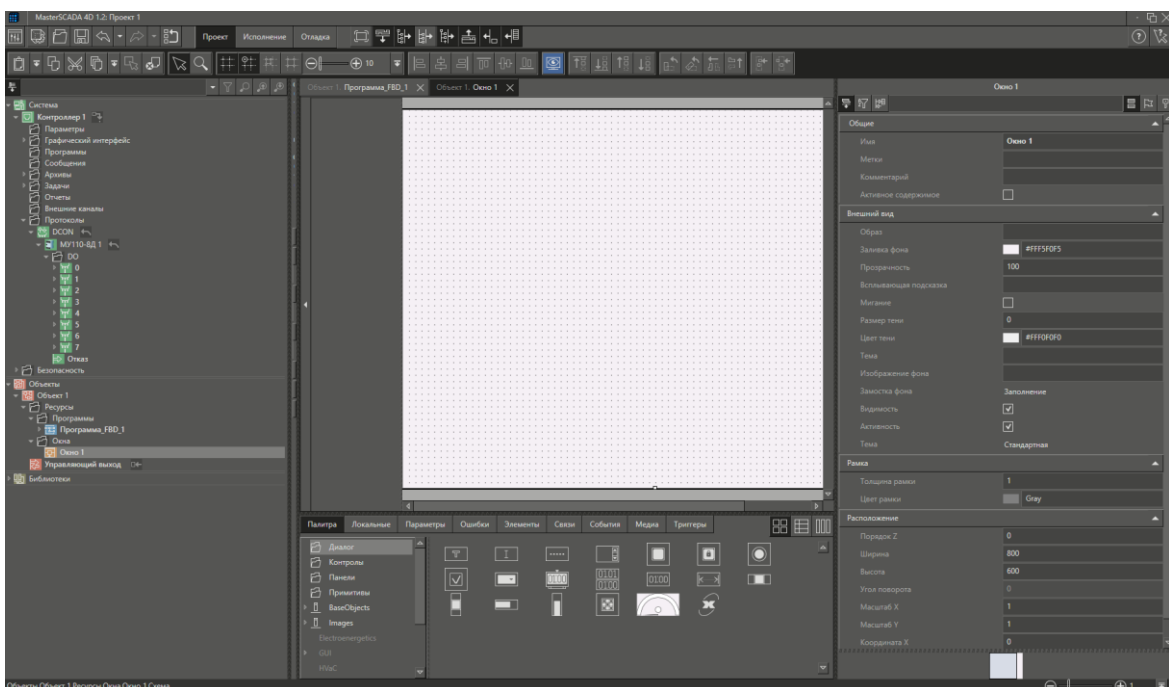
Следующий этап – создание визуализации.

2.1.4. БС. УРОК 1. СОЗДАНИЕ ОКНА УПРАВЛЕНИЯ

Начнем с выбора в контекстном меню объекта пункта Добавить - Окно.

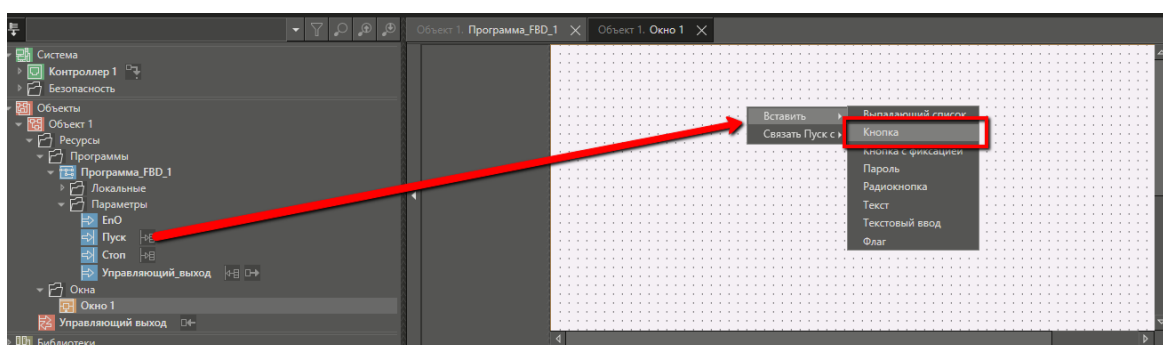


При этом у объекта появится окно, которое сразу откроется в поле редактора графического интерфейса.

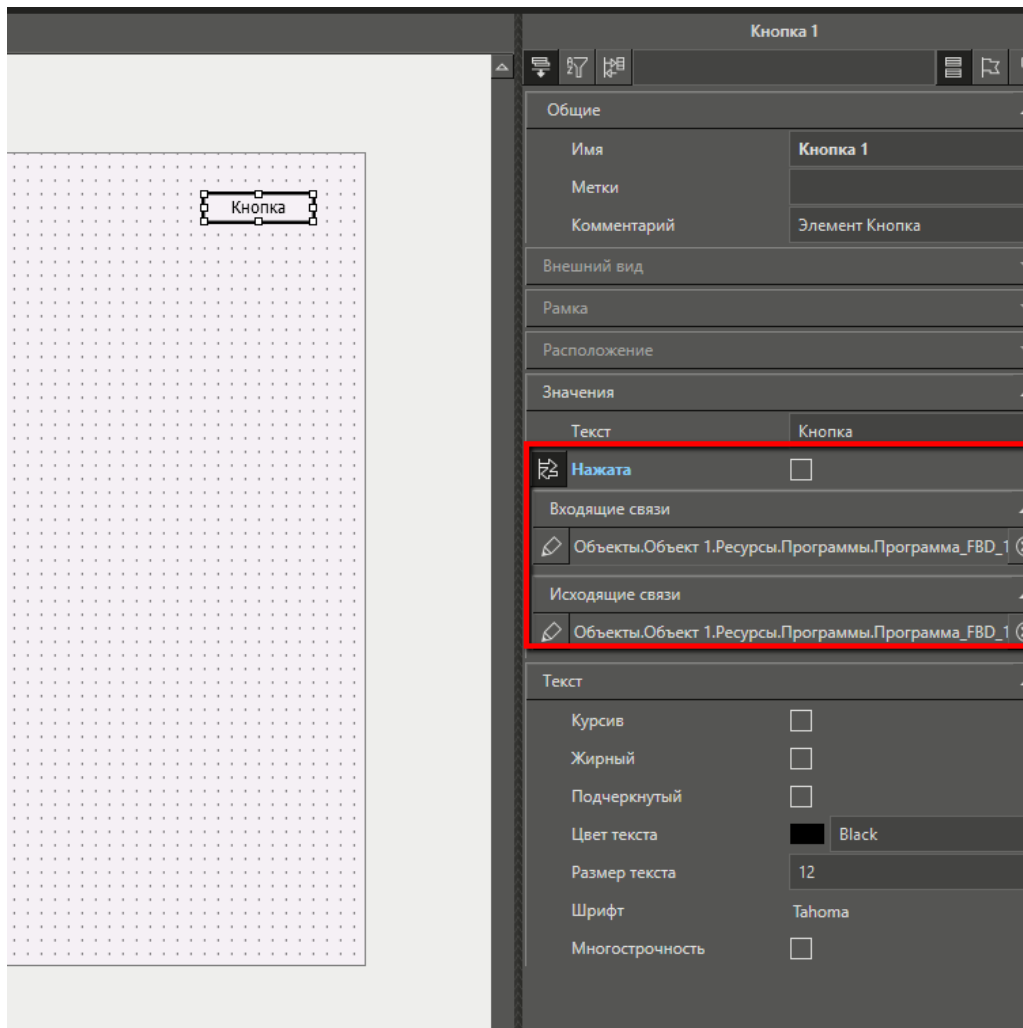


Наша задача заключается в том, чтобы разместить в этом окне две кнопки – Пуск и Стоп, нажатие которых будет передаваться на соответствующие входные параметры программы, а также создать индикацию состояния Управляющего выхода.

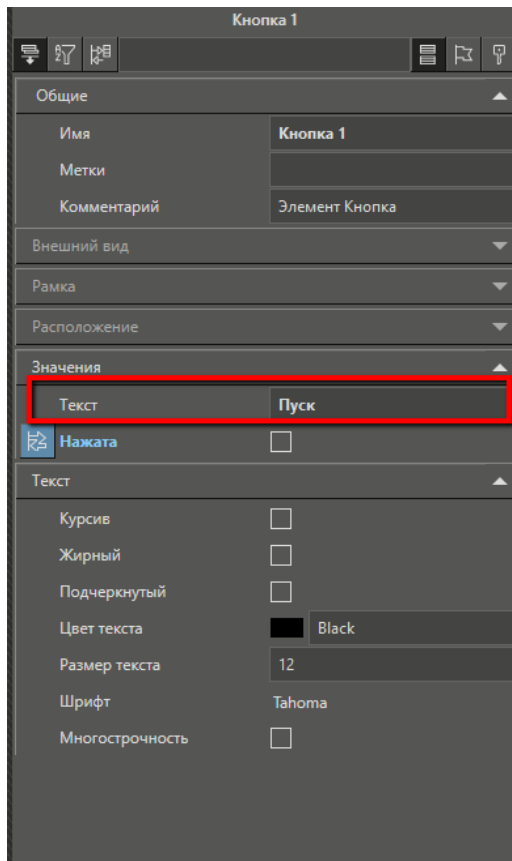
Из группы Параметры созданной нами программы перетащим правой кнопкой мыши параметр Пуск в окно. Из открывшегося контекстного меню выберем пункт Вставить.Кнопка.



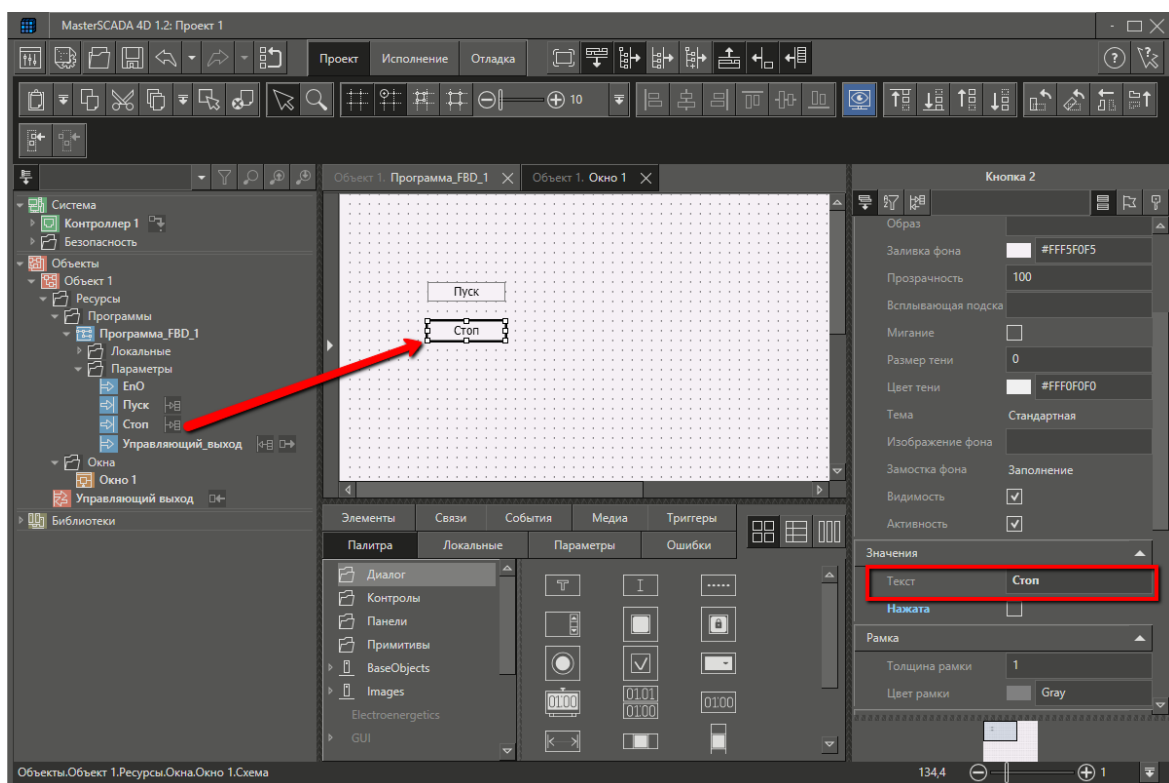
В окно будет вставлена кнопка с уже установленными связями между параметром Пуск и свойством кнопки Нажата. По нажатию кнопки параметр Пуск будет устанавливаться в TRUE, а при отпусканию кнопки этот же параметр будет установлен в False. Таким образом, параметр Пуск будет отражать состояние кнопки.



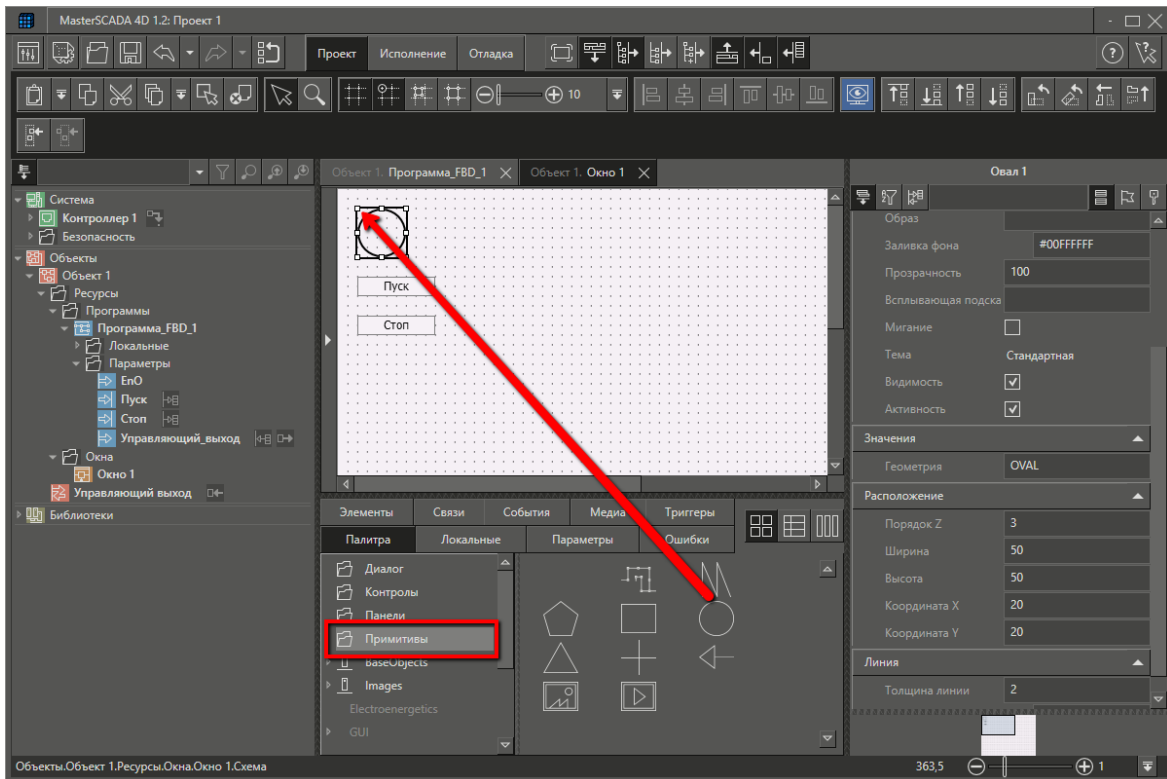
Назовем эту кнопку Пуск, набрав с клавиатуры это название в графе текст панели свойств:



Аналогичные действия произведём с параметром Стоп.

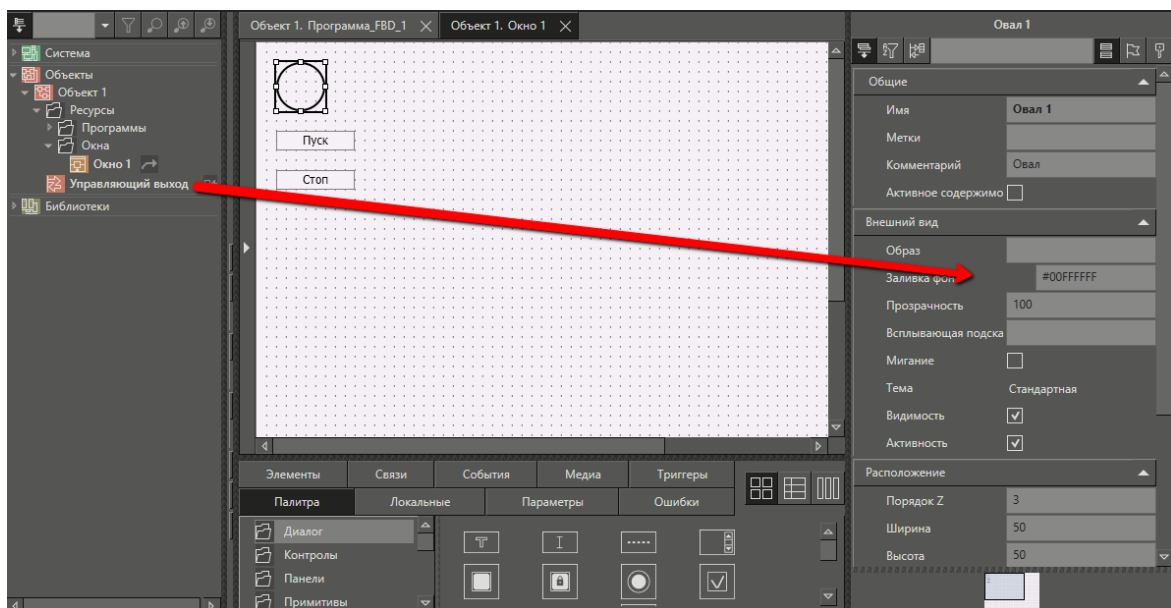



Теперь реализуем отображение состояния Управляющего выхода. Для этого перейдём в Легенду графического редактора, откроем вкладку Палитра, выберем категорию Прimitives, и перетащим левой кнопкой мыши элемент Овал в окно (мнемосхему).

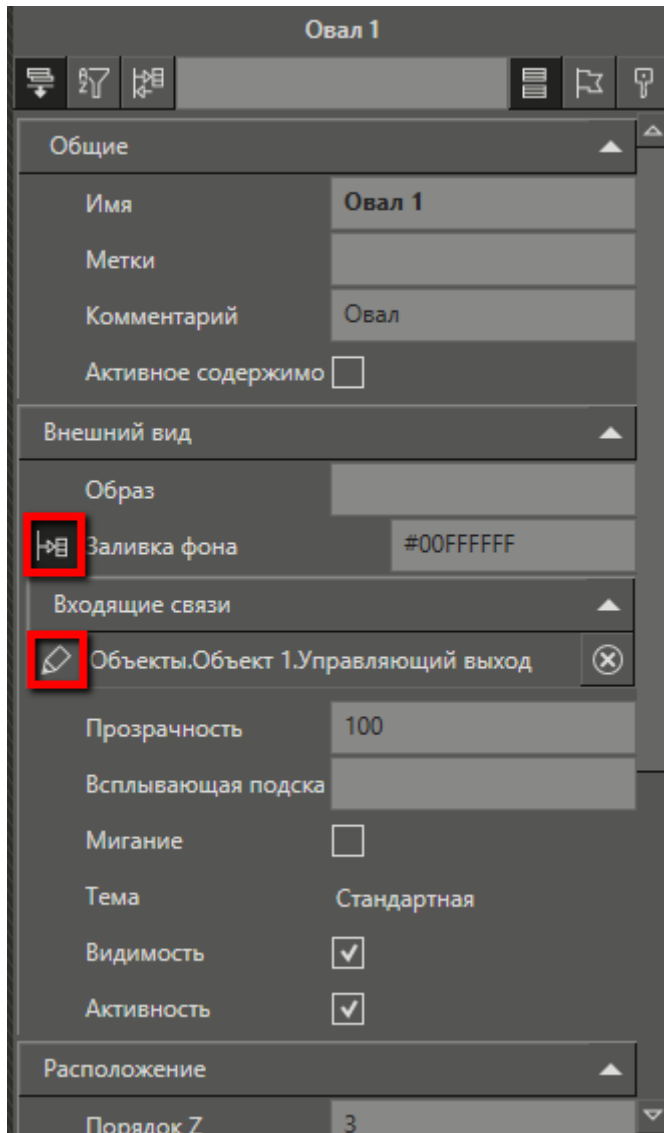


Сделаем так, чтобы цвет овала отображал текущее состояние выходного сигнала: включен или выключен.

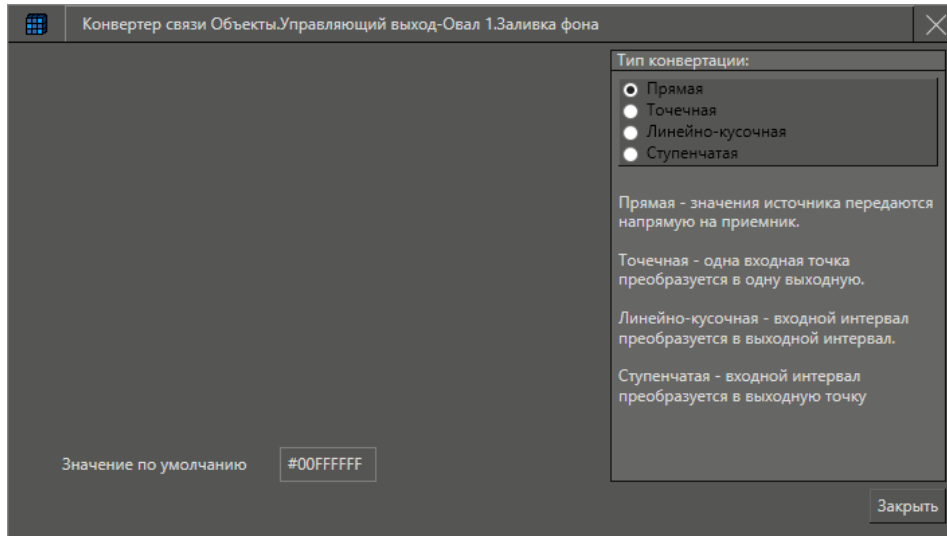
Для этого необходимо параметр объекта Управляющий выход перетащить мышкой на свойство овала Заливка фона.



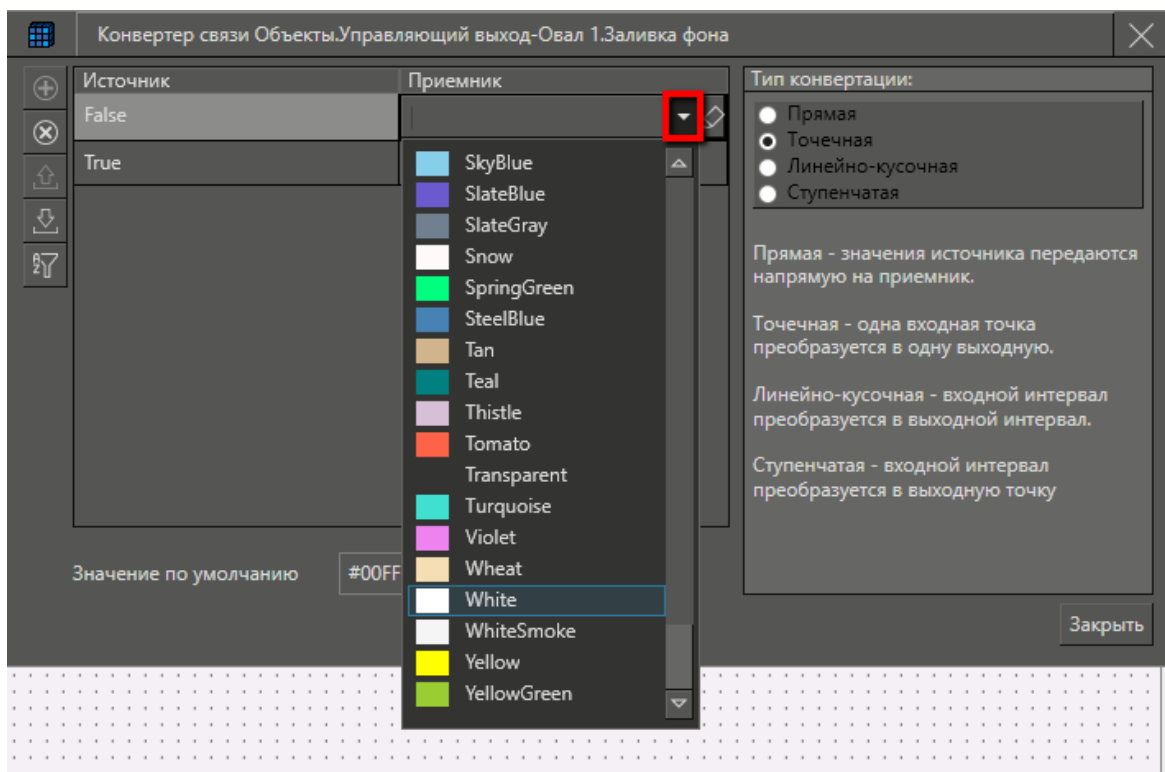
Справа от свойства появится значок связи . При нажатии на него левой кнопкой мыши откроется дополнительная информация о созданной связи. Нажмём на значок .



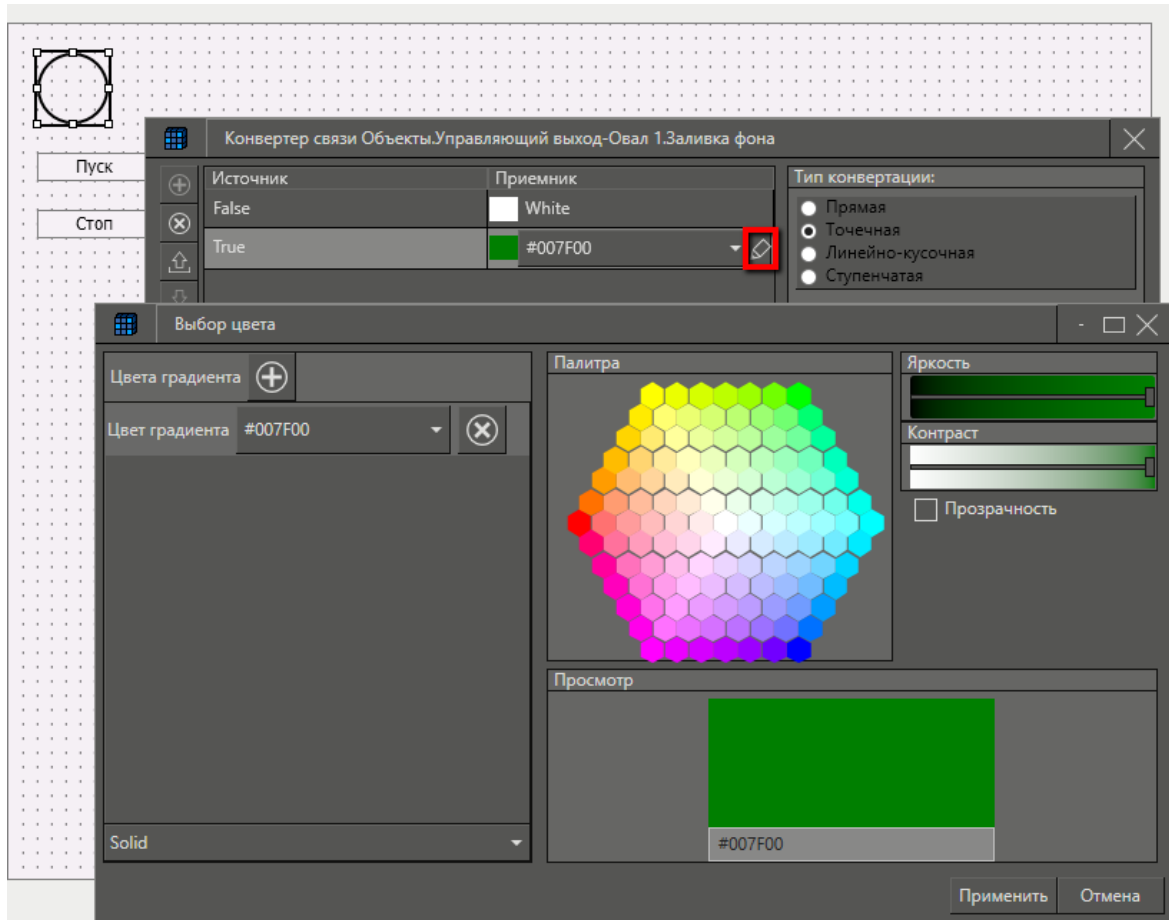
Откроется диалоговое окно:



На любую связь в **MasterSCADA 4D** можно назначить конвертацию. Выберем точечный тип конвертации. Назначим так, чтобы значению False соответствовал белый цвет (выберем из стандартных цветов WHITE). Для того чтобы кнопки выбора стали доступны, необходимо нажать правой кнопкой мыши напротив нужного значения Источника в поле Приемник.

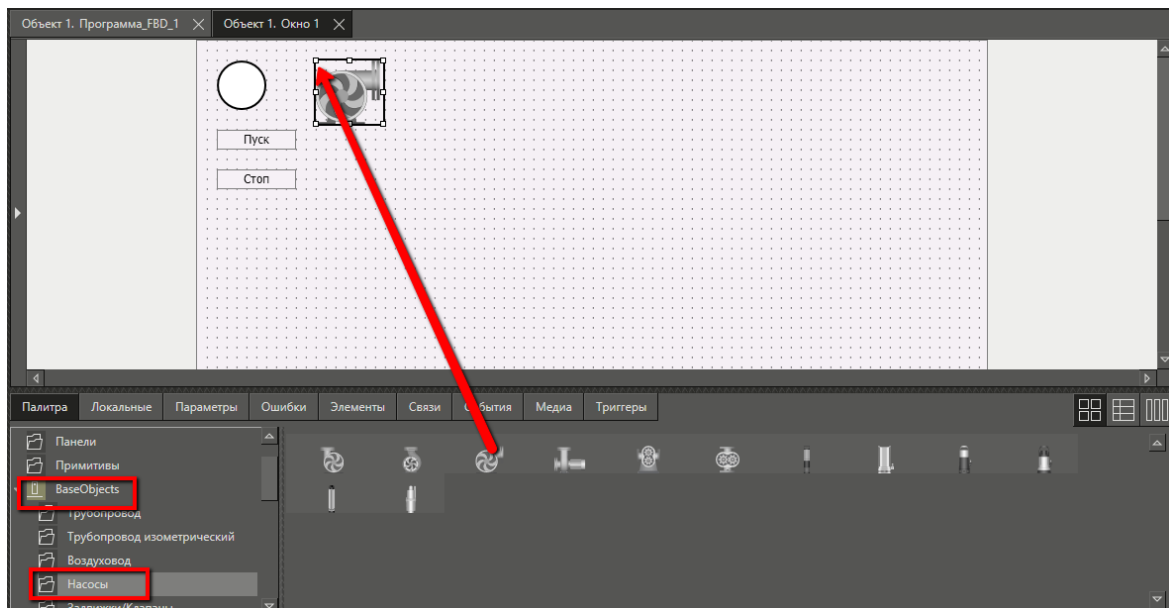


Для значения TRUE выберем зеленый цвет из палитры.

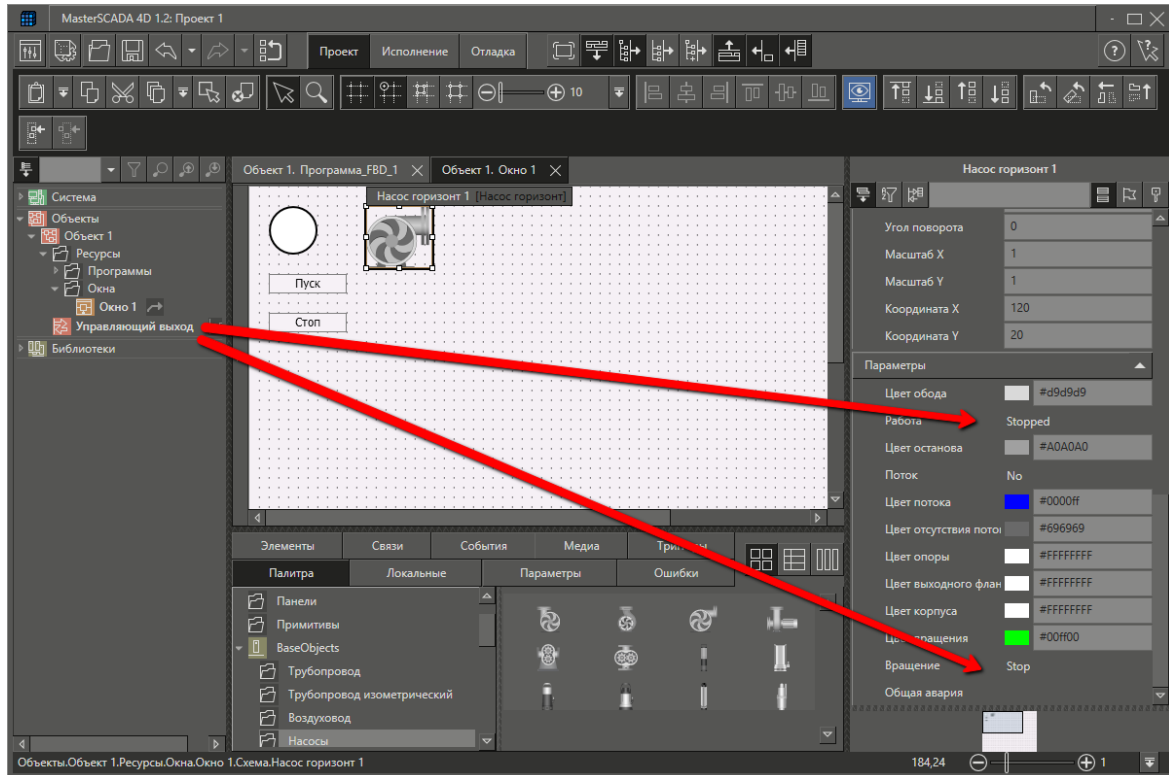


После этого закроем окно конвертации.

Теперь добавим изображение насоса. Для этого выберем любой насос из библиотеки насосов, входящих в библиотеку Общая, и перетащим его в окно.

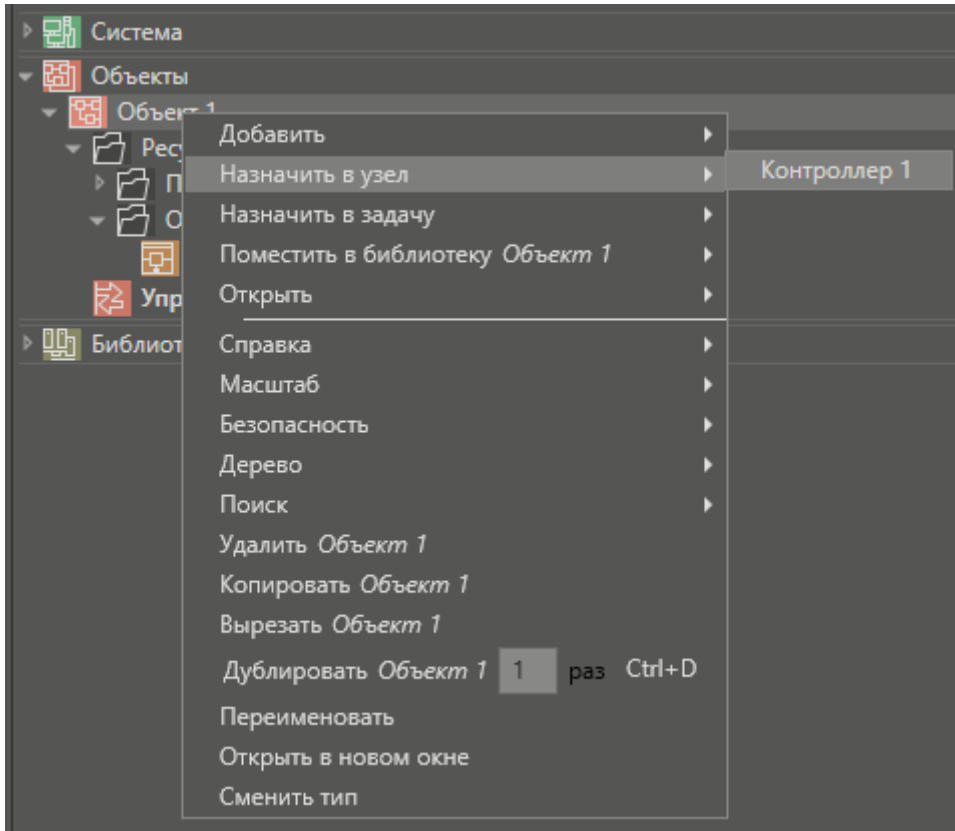


Наконец, свяжем параметр объекта Управляющий выход со свойствами насоса Вращение и Работа.

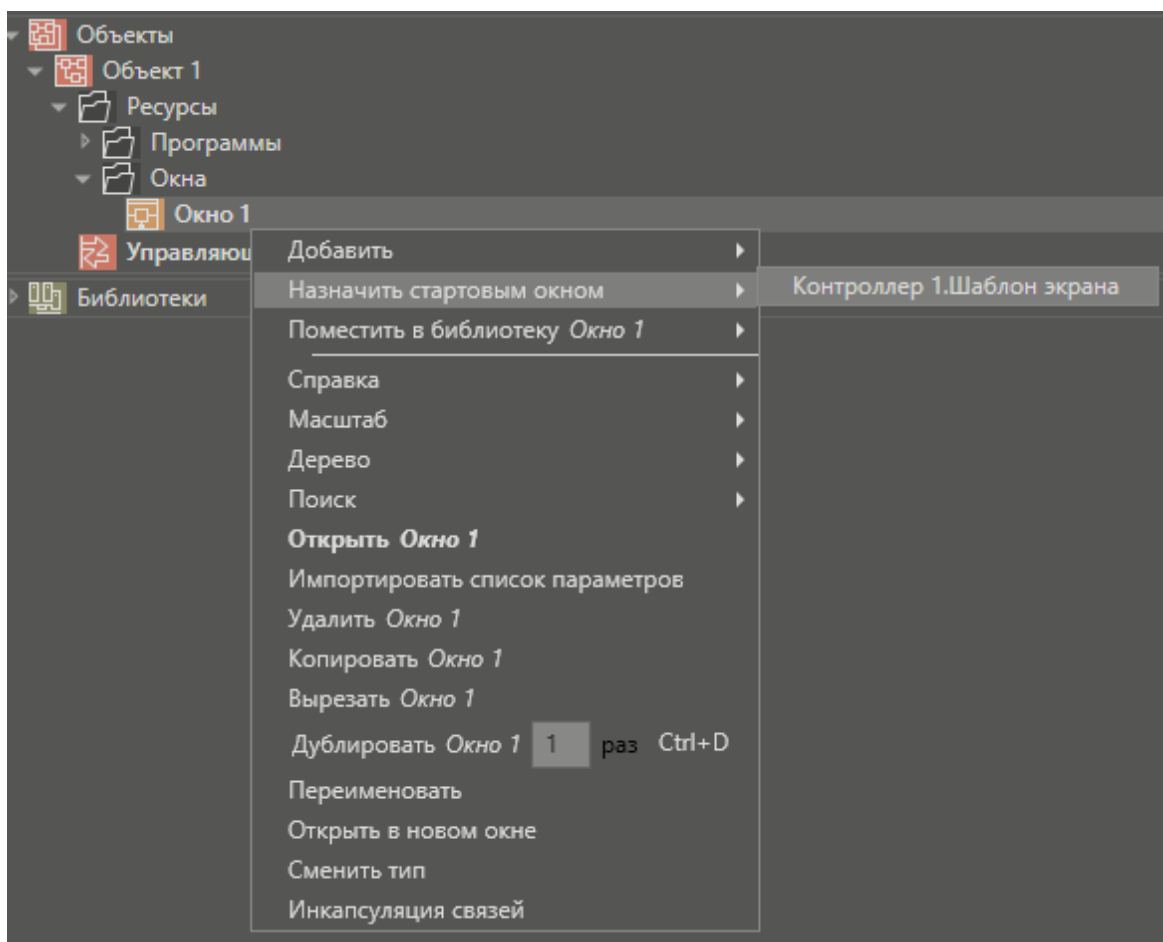


На этом формирование визуализации завершено.

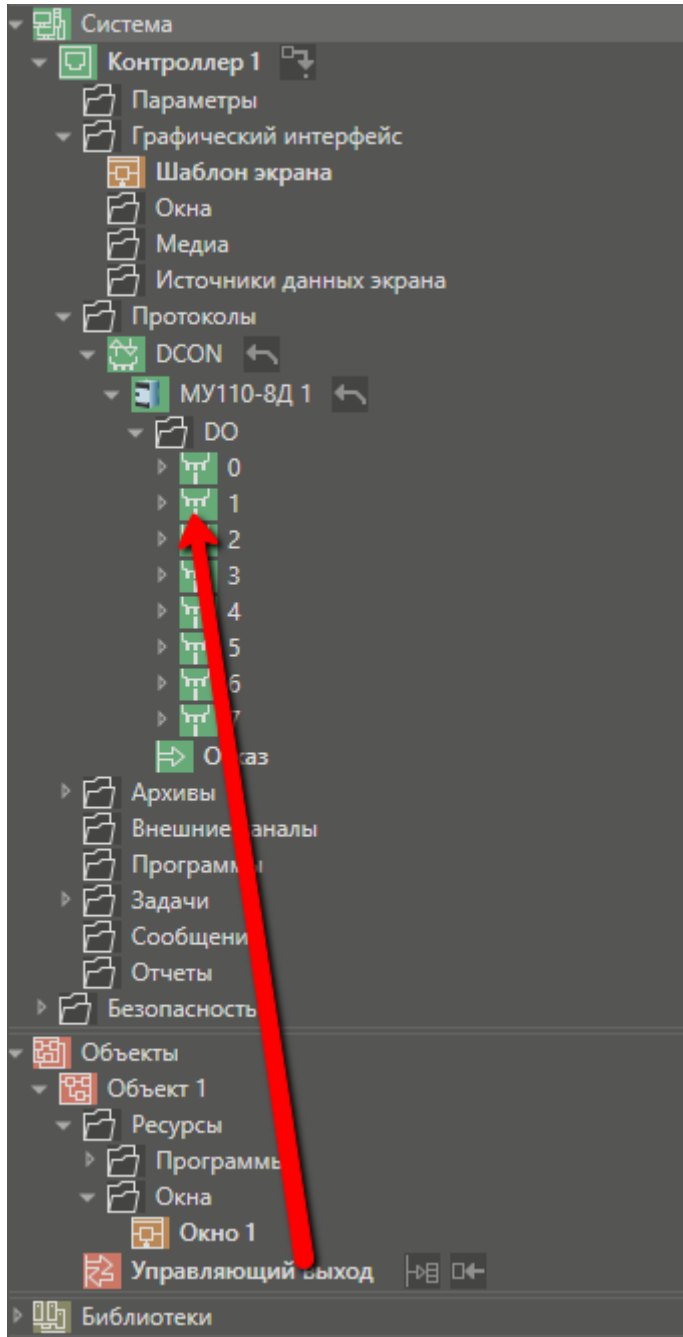
Теперь назначим объект для исполнения в узле контроллера. Это необходимо сделать потому, что узлов в проекте может быть много, и любой объект может быть назначен для исполнения в любом из них. Для этого нажмем правой кнопкой мыши на объект, и из контекстного меню выберем **Назначить в узел**, а затем выберем название узла.



Следующим шагом нужно назначить стартовое окно. В нашем проекте пока есть только одно окно. Но таких окон может быть много, и необходимо указать, какое из них будет стартовым. Для этого нажмем правой кнопкой мыши на это окно, из контекстного меню выберем Назначить стартовым окном, и укажем , где именно стартовое окно будет исполняться. В нашем случае есть только один возможный вариант.



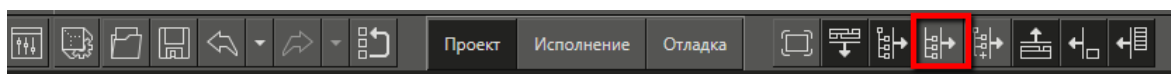
Итак, получился функционально законченный объект, содержащий программу управления, средства визуализации и имеющий один параметр для связи этого объекта с внешним миром. В нашем случае нам требуется при помощи этого параметра управлять дискретным выходом модуля. Для этого перетащим левой кнопкой мыши параметр объекта Управляющий выход на дискретный выход DO.1. При этом создастся связь, передающая значение из параметра Управляющий выход в канал дискретного выхода.



На этом создание проекта закончено.

Полезные советы:

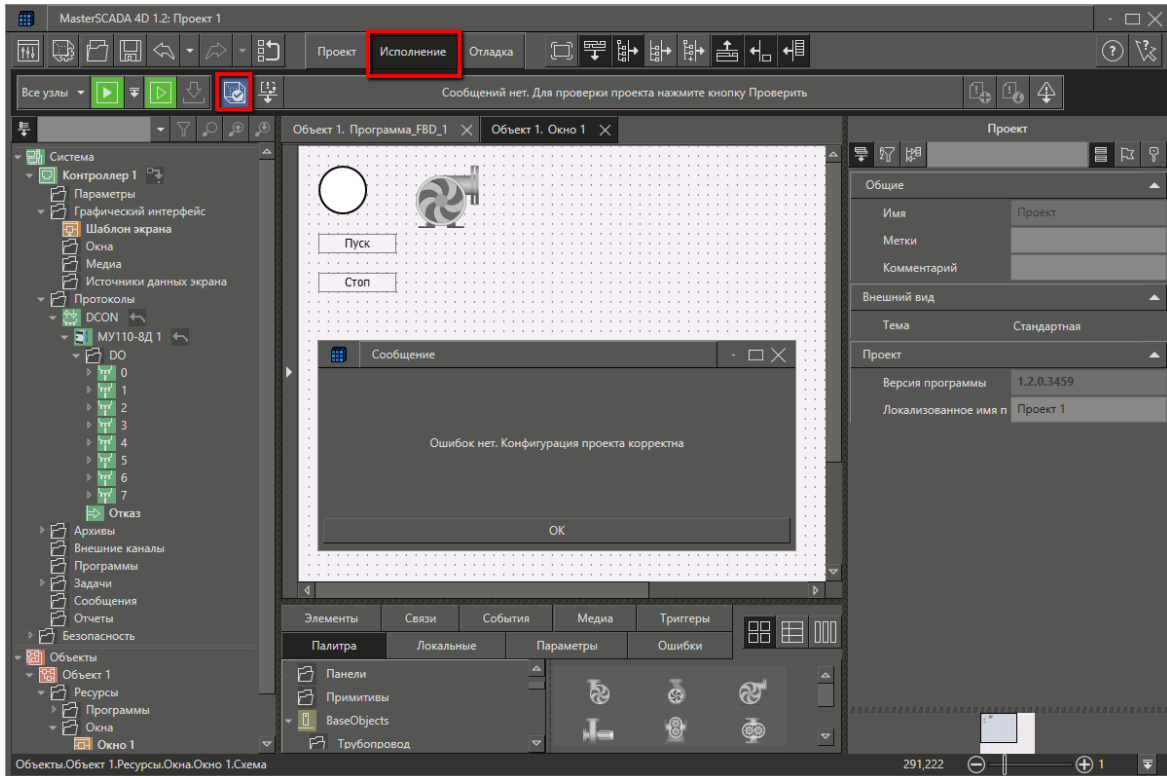
Если в какие-то моменты панель контекста не требуется в работе, то ее можно отключить в панели свойств, отжав соответствующую кнопку:



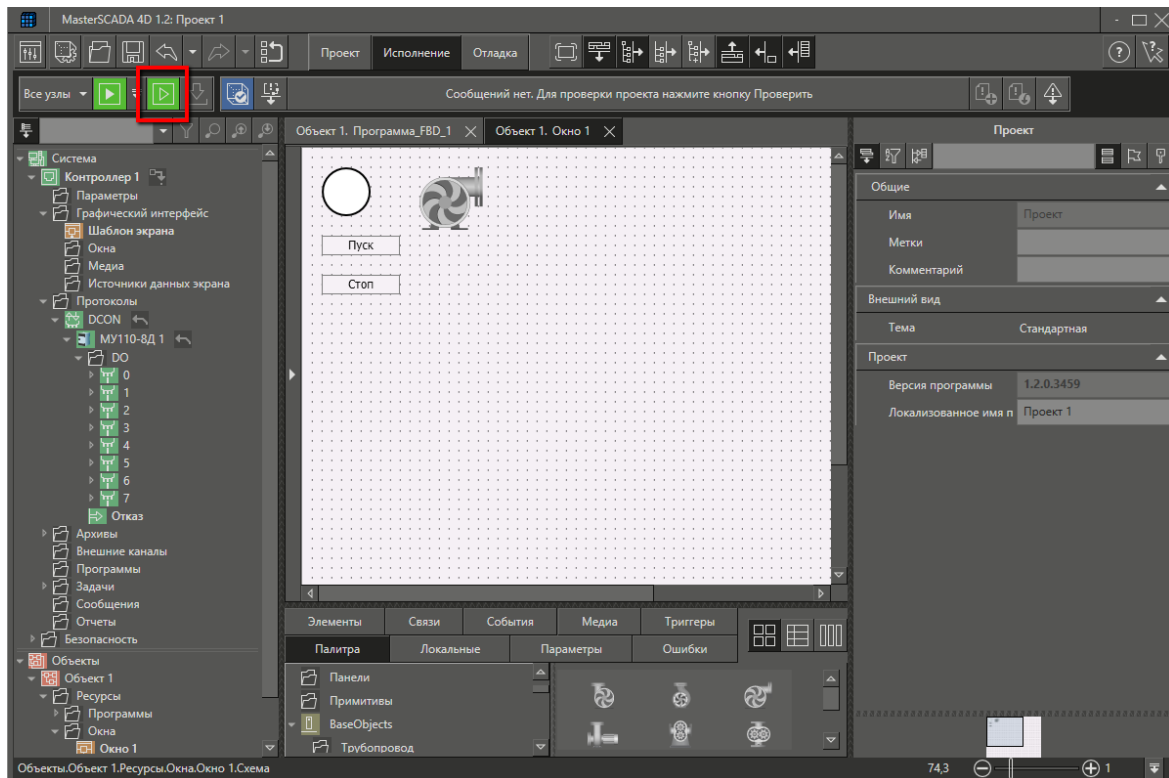
При необходимости, можно также уменьшить размер панели, перетаскив ее правую границу к левой.

2.1.5. БС. УРОК 1. ЗАПУСК

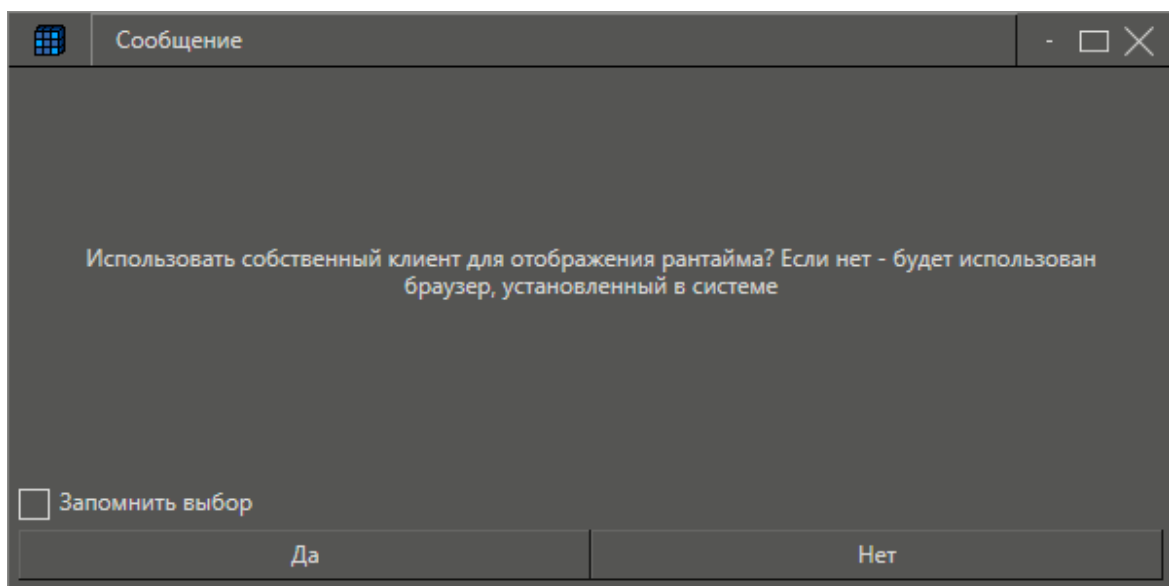
Сначала перейдём на вкладку **Исполнение** панели инструментов и нажмём кнопку **Проверить**:




Затем, убедившись в отсутствии ошибок проекта, запустим проект на исполнение в режиме эмуляции. При этом загрузка в контроллер производиться не будет, а исполнительная система будет эмулироваться на том же компьютере, на котором производилась разработка проекта.

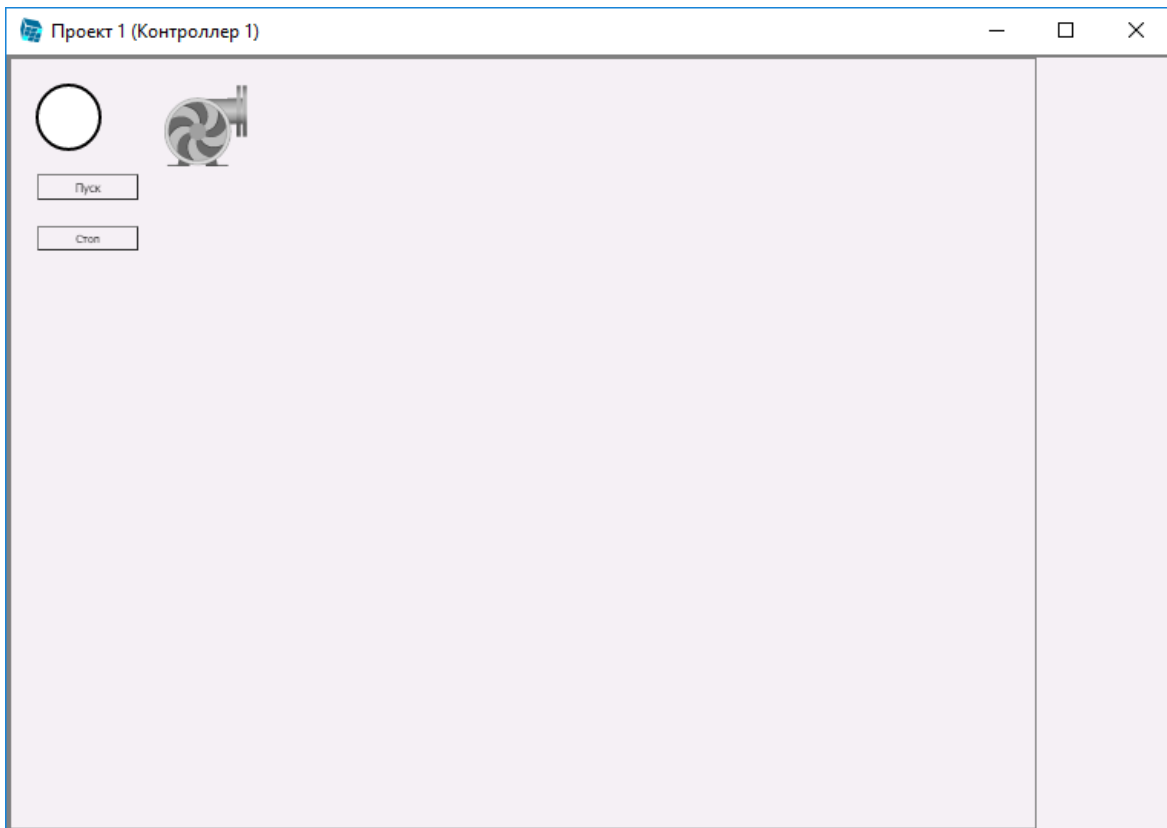


Если режим исполнения запускается впервые, то появится окно, предлагающее для отображения в режиме исполнения на выбор либо клиент визуализации, разработанный компанией "МПС Софт", либо браузер, назначенный на компьютере по умолчанию.

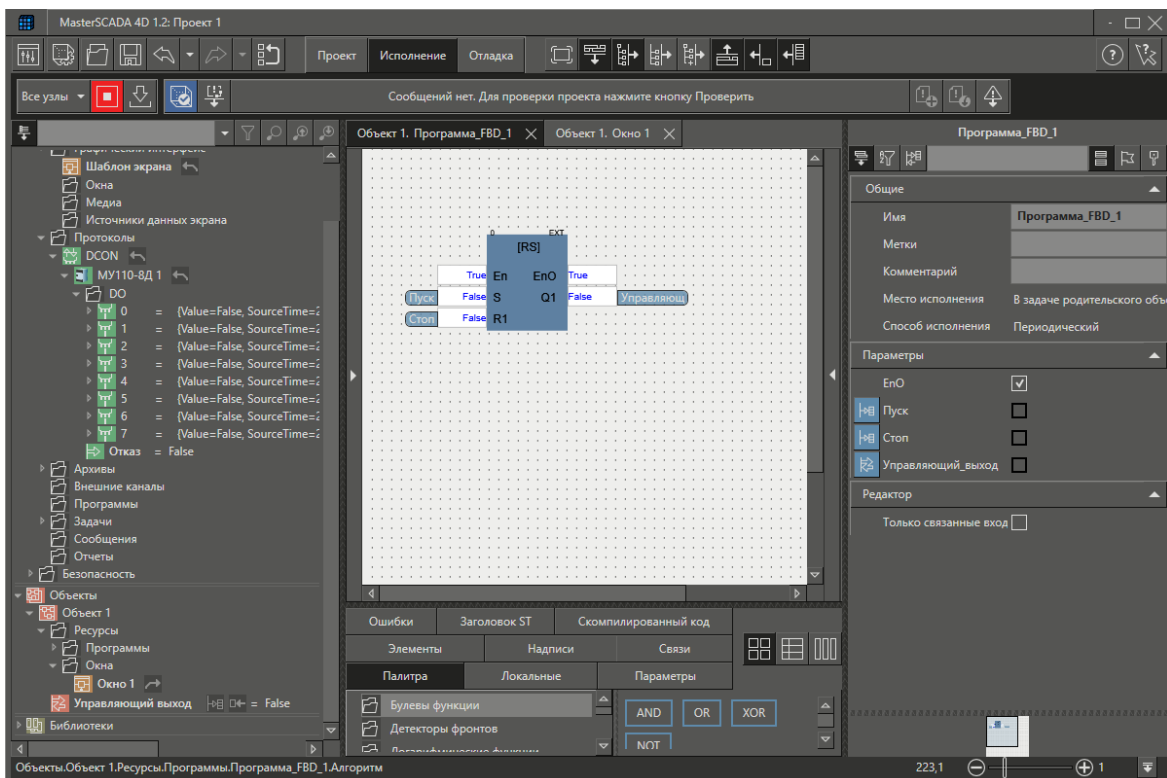


В нашем случае выбор может быть любым.

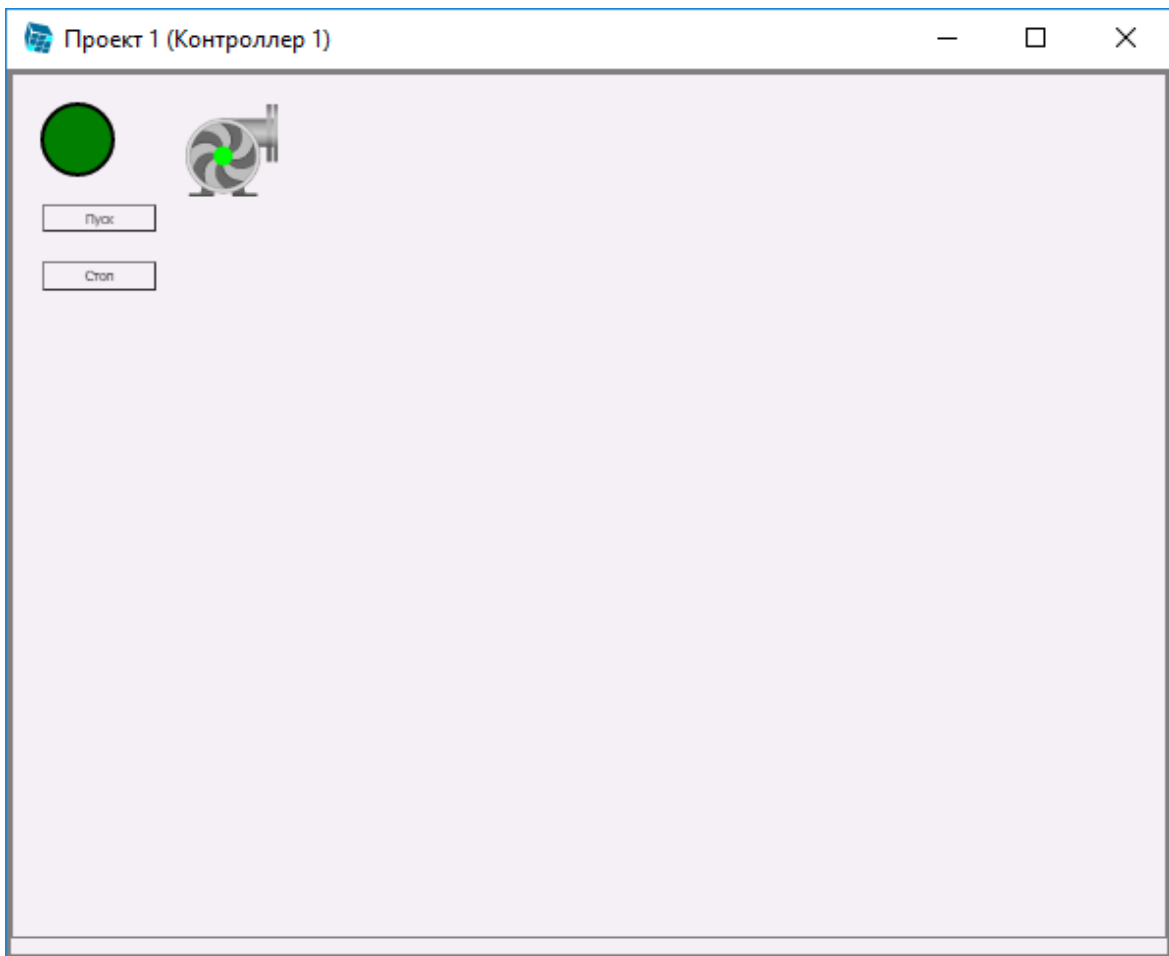
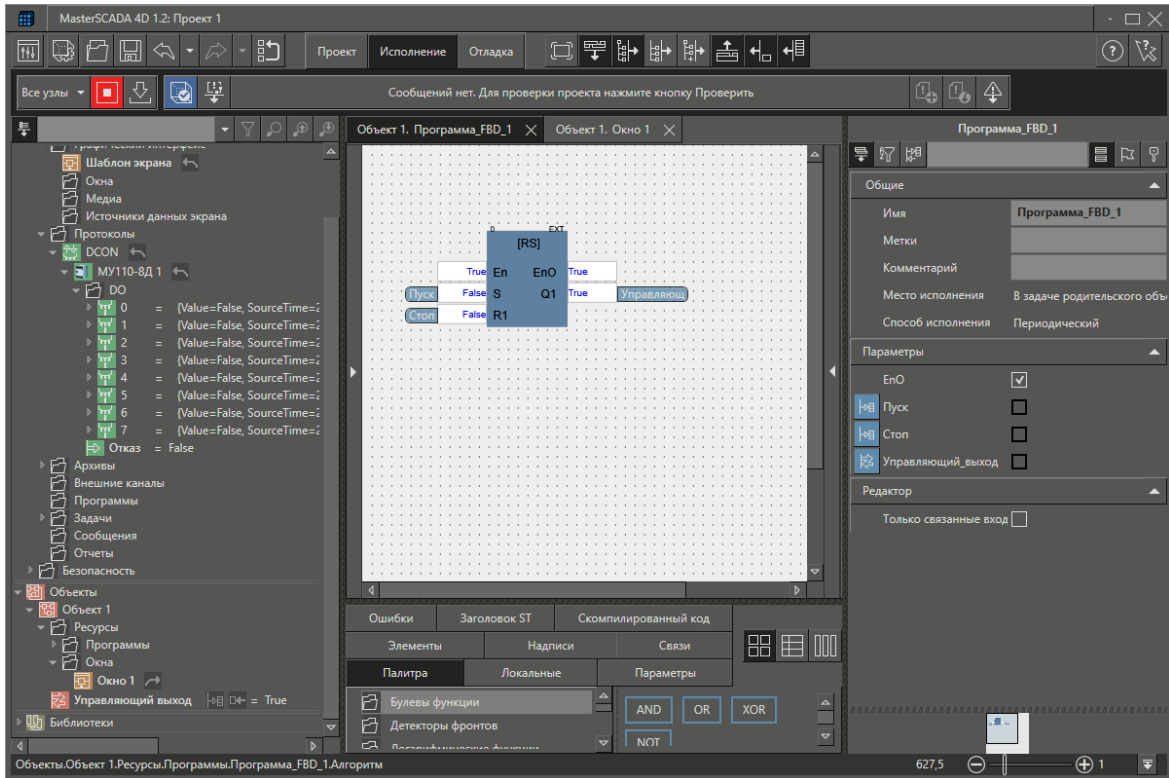
После успешного запуска проекта на исполнение, рядом с названием узла появится знак , и откроется встроенный клиент визуализации (или браузер) с созданной мнемосхемой.



В среде разработки рядом с именами параметров появятся их текущие значения.



Нажмём кнопку Пуск в среде исполнения. Управляющий выход примет значение TRUE, это же значение поступит на дискретный выход модуля, индикатор на мнемосхеме окрасится в зелёный цвет, и начнет вращаться насос.



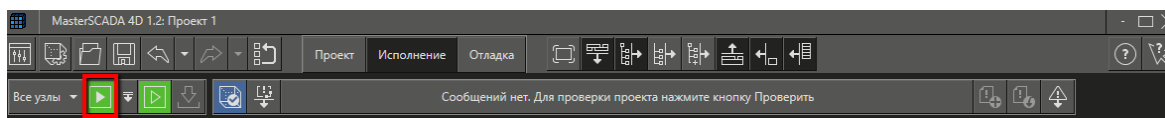
Нажмем кнопку Сtop. Управляющий выход примет значение False, индикатор на мнемосхеме окрасится в белый цвет, а насос остановится.

За счёт чего это происходит? Пока кнопка Пуск нажата, значение параметра Пуск на входе программы равно TRUE. Это значение поступает на вход S триггера, устанавливая выход триггера в состояние TRUE (или, как ещё говорят, "взводит триггер"). После отпускания кнопки триггер остаётся в прежнем состоянии до тех пор, пока не будет нажата кнопка Стоп – это приведёт к появлению значения TRUE на входе R триггера и, соответственно, к установке выхода триггера в состояние False (или, как говорят, "сбросу триггера").

Теперь, если Ваш компьютер доступен по сети с других устройств, откройте браузер на другом устройстве, наберите в адресной строке IP-адрес компьютера, на котором Вы запустили режим исполнения, и убедитесь, что Вы можете управлять насосом совершенно одинаково и в любой последовательности с любого из устройств.

Если у Вас есть контроллер, подключенный к сети, то можно загрузить программу в него.

Для этого отключите режим исполнения, выделите в дереве контроллер, убедитесь, что у него указан IP-адрес, и, вместо кнопки Эмуляция, нажмите кнопку Подключить.



После этого проект будет загружен у контроллер и запущен на исполнение.

2.2. БС. УРОК 2

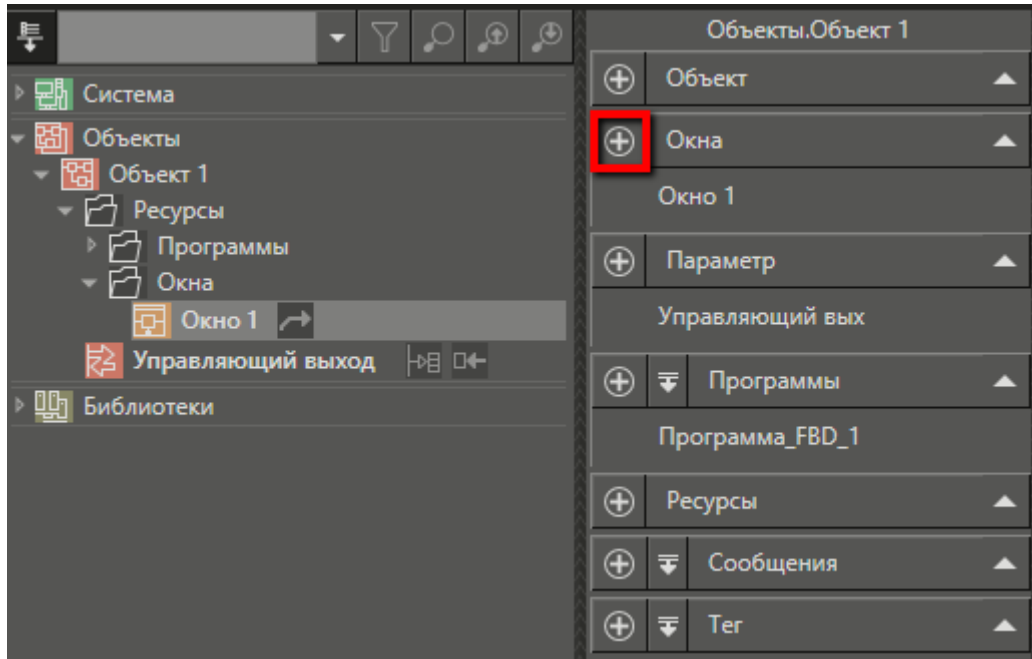
В **Уроке 1** нами был создан проект управления насосом от двух кнопок и визуальное отображение состояния насоса. В **Уроке 2** мы модифицируем созданный ранее проект и реализуем управление этим же насосом в отдельном всплывающем окне.

Мы продолжим с того места, где закончился **Урок 1**. Поэтому, если Вы еще не проходили его, то настоятельно рекомендуем сначала выполнить все рассмотренные в нём действия, это не сложно. Если Вы уже прошли **Урок 1**, то просто откройте этот проект!

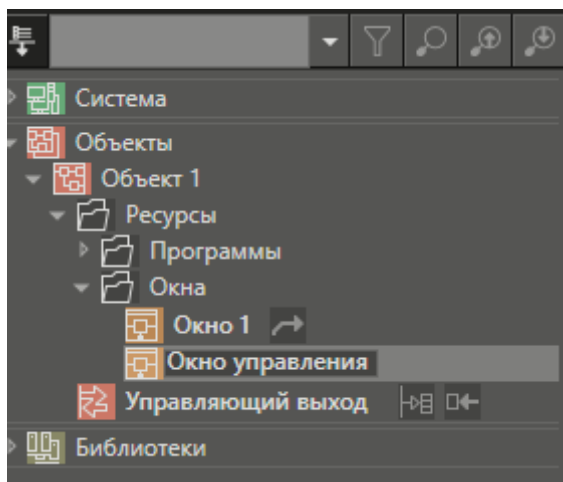
- **БС. Урок 2. Конфигурирование всплывающего окна**
- **БС. Урок 2. Запуск**

2.2.1. БС. УРОК 2. КОНФИГУРИРОВАНИЕ ВСПЛЫВАЮЩЕГО ОКНА

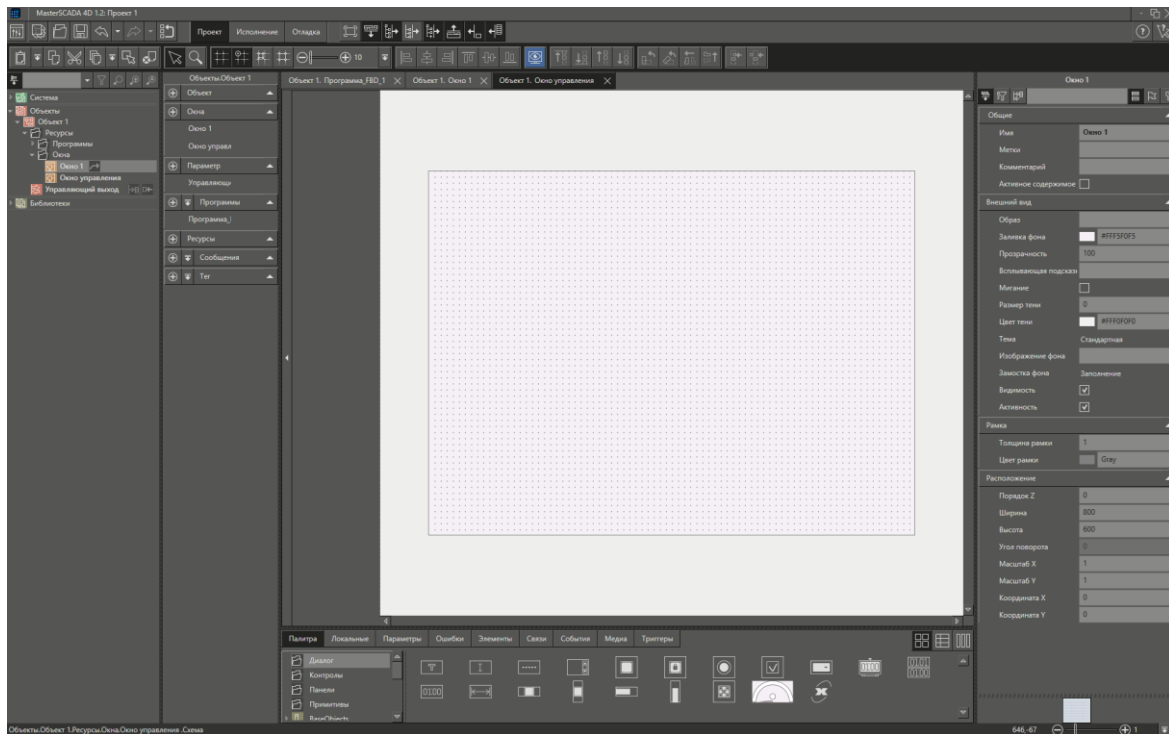
Сначала добавим в **Объект 1** ещё одно окно. Для этого в контекстной панели объекта нажмем соответствующую кнопку:



Пусть новый элемент называется Окно управления. Для изменения имени, присвоенного добавленному элементу по умолчанию, можно воспользоваться панелью свойств, либо, дважды нажав мышью на элемент в дереве, ввести с помощью клавиатуры новое имя в появившемся поле:

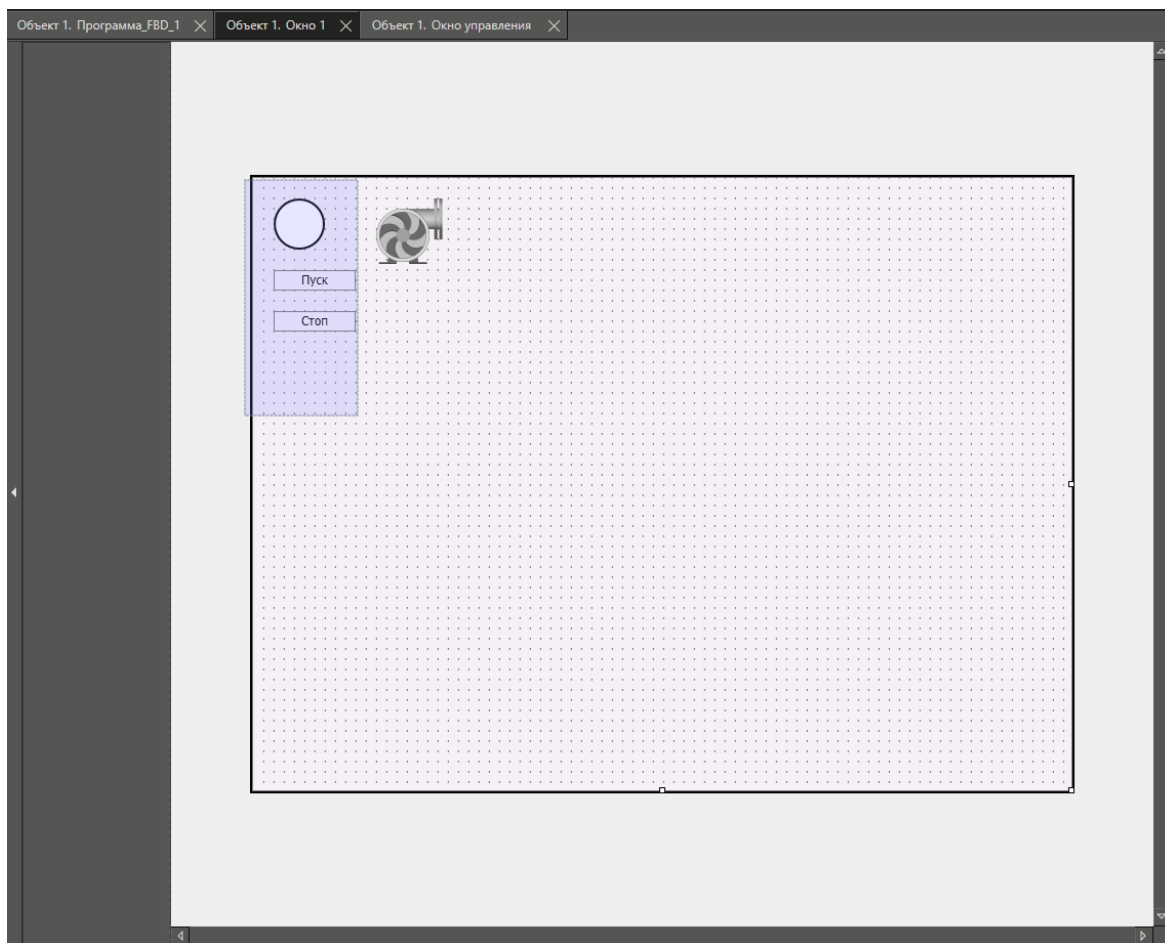


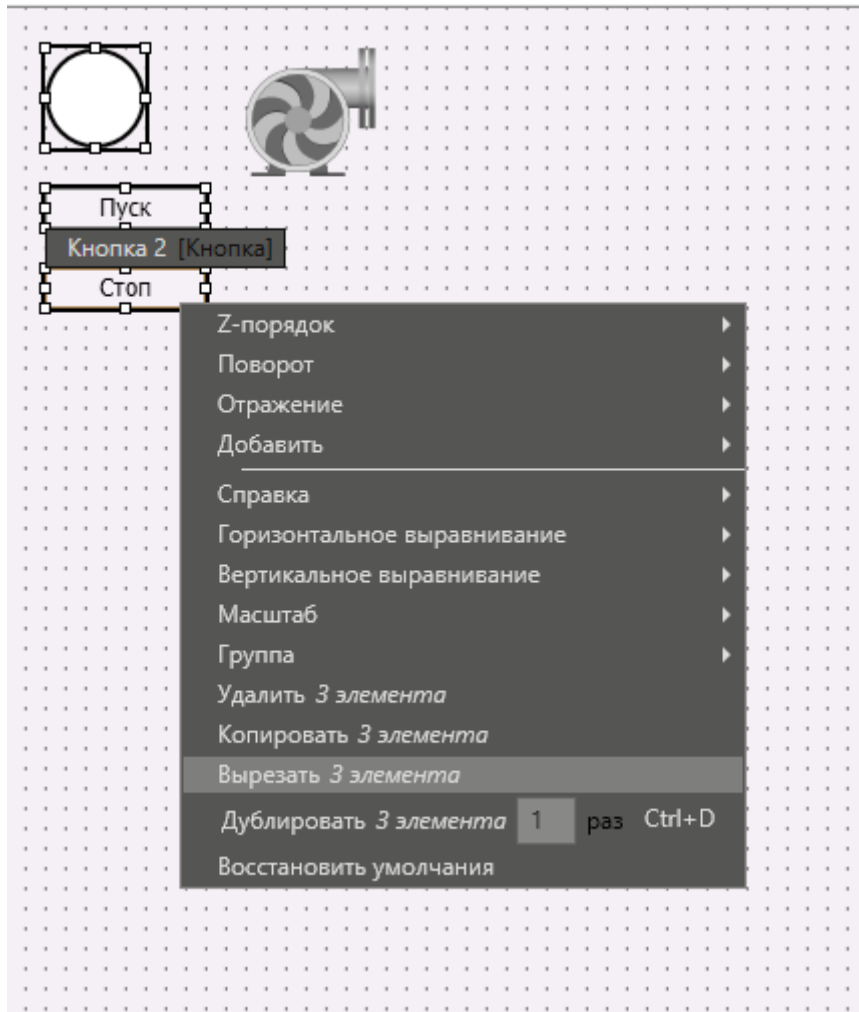
После создания окна автоматически откроется графический редактор:



Перенесём в него элементы управления и индикации из основного окна.

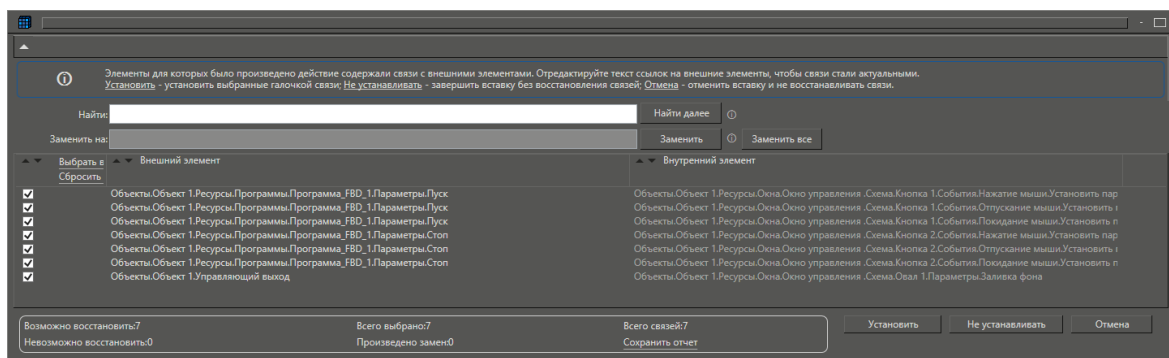
Для этого перейдём на вкладку Окно 1, при помощи мыши выделим обе кнопки и индикатор, и вырежем эти элементы, либо нажав комбинацию клавиш **Ctrl+X**, либо при помощи кнопки **Вырезать** на панели инструментов, либо при помощи контекстного меню выделенных элементов.



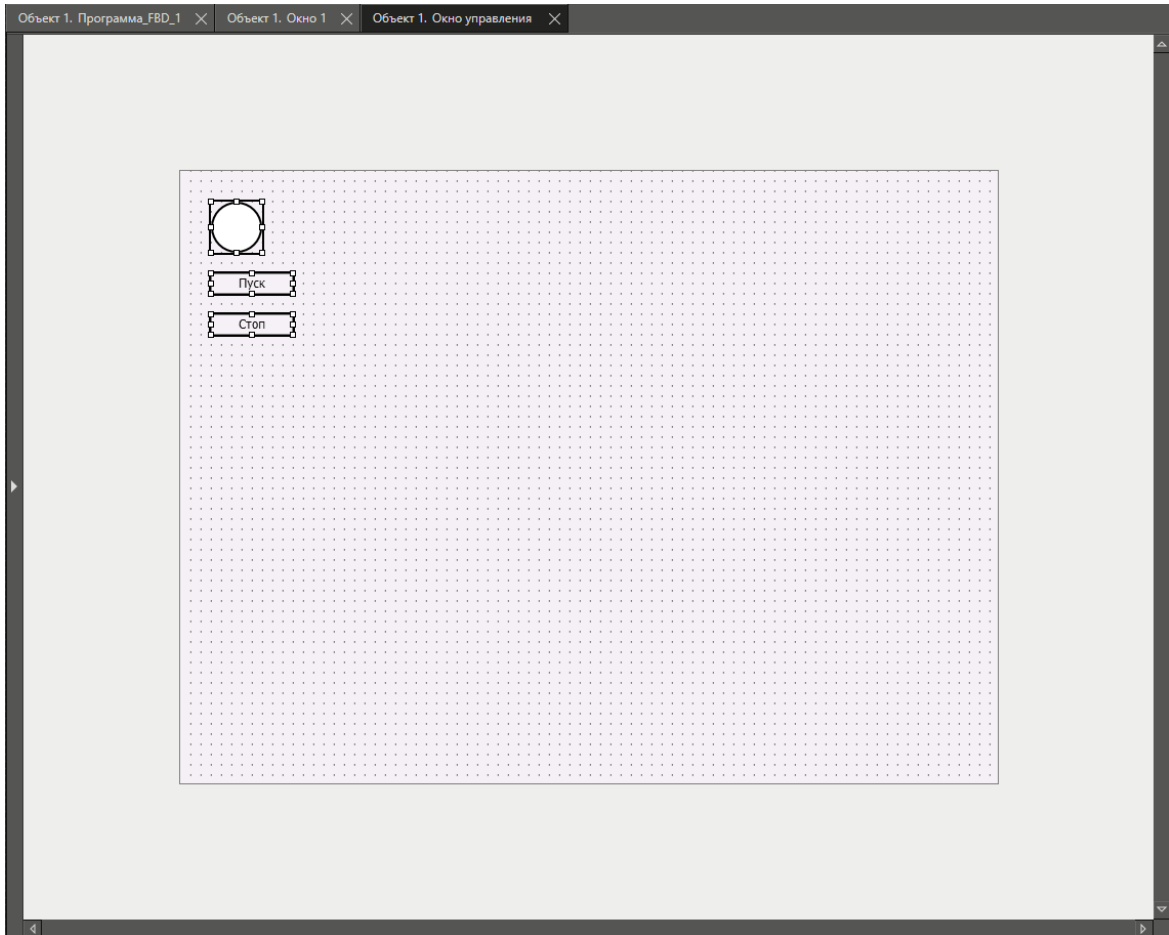


Теперь перейдём на вкладку Окно управления и, либо через контекстное меню Вставить, либо нажав комбинацию клавиш **Ctrl+V**, вставим туда вырезанные элементы.

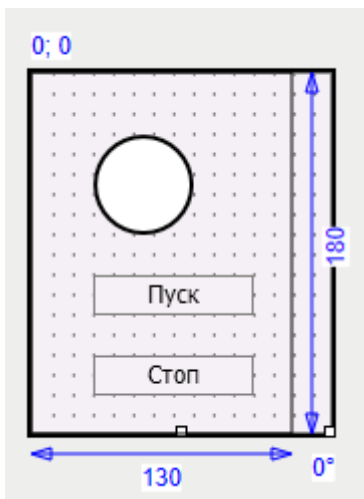
При этом появится диалоговое окно восстановления связей:



Нажмем кнопку **Установить** и получим результат:

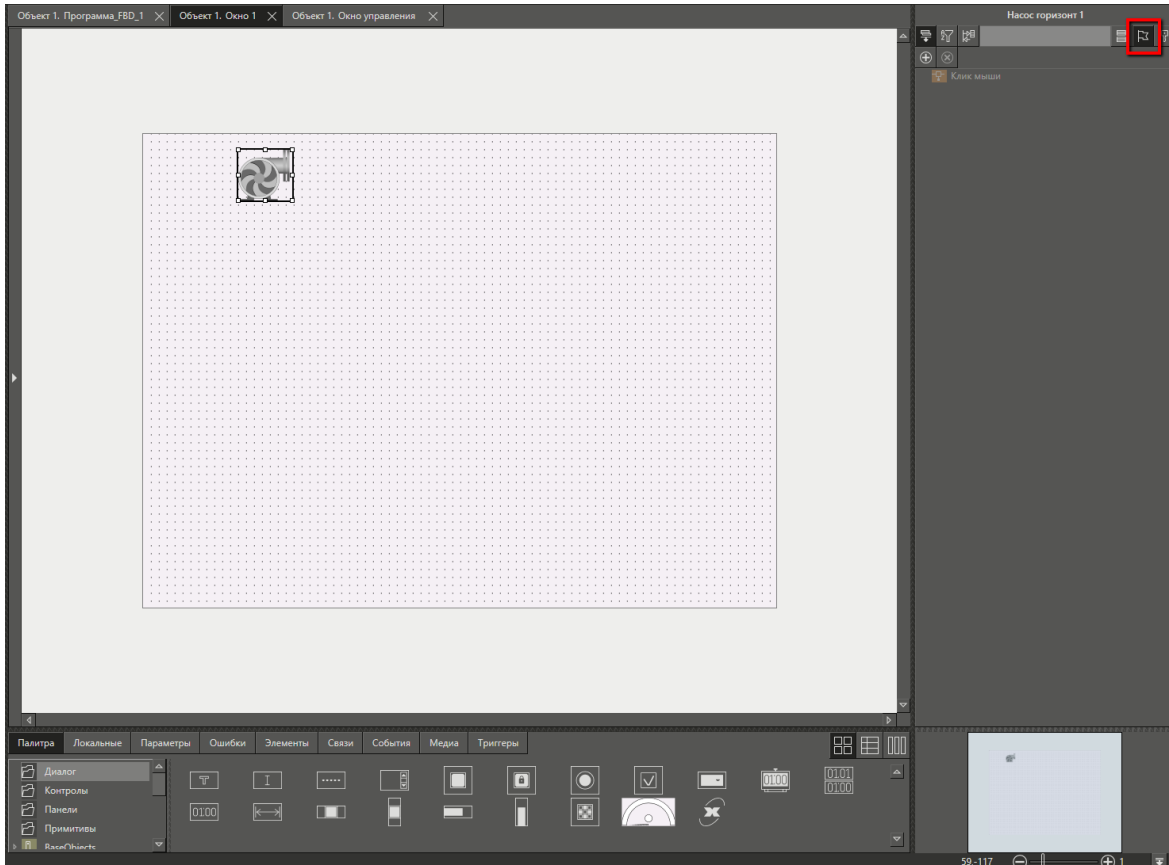


Затем уменьшим размер окна, потянув влево правую границу окна и вверх - нижнюю:

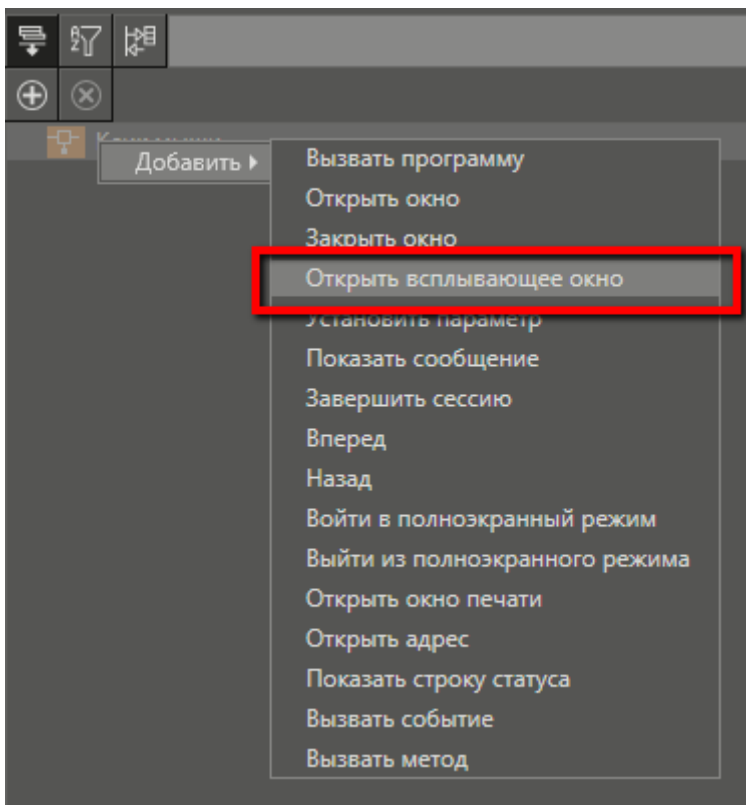


Теперь нам предстоит сделать так, чтобы это окно открывалось по клику мыши на изображении насоса.

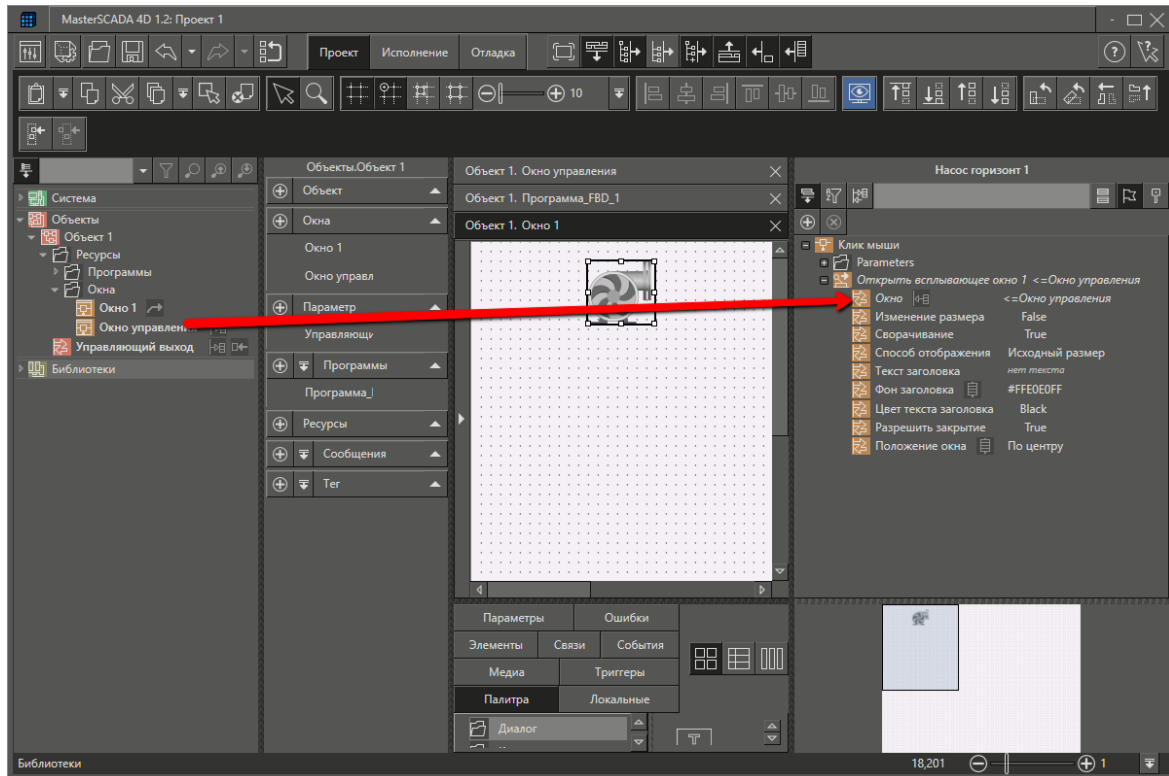
Для этого перейдём на вкладку Окно 1, которое является основным. Выделим в окне изображение насоса, нажав на него левой кнопкой мыши. Панель свойств переведём в режим отображения событий:



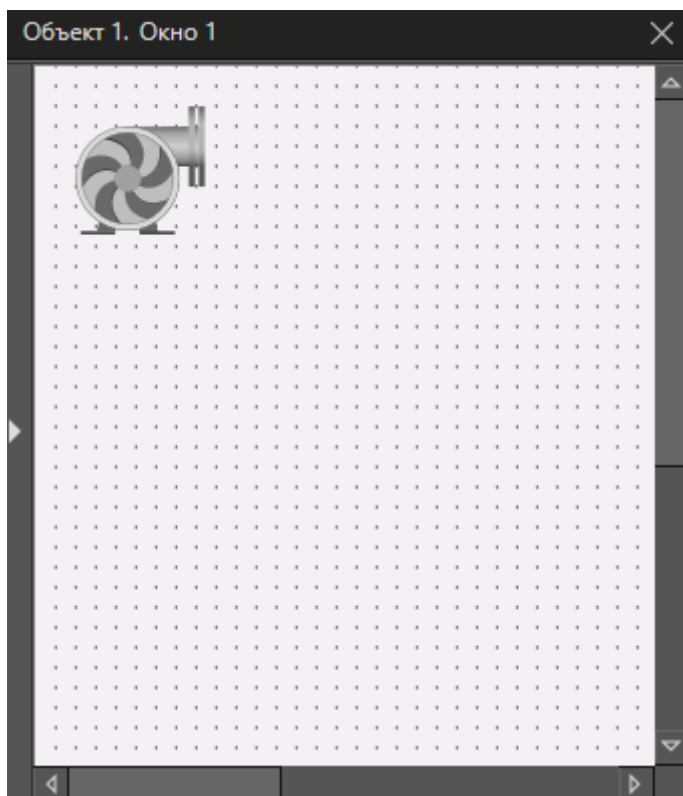
Нажмем правой кнопкой мыши на событие Клик мыши. Из контекстного меню выберем Добавить, а из появившегося списка доступных действий – Открыть всплывающее окно:



Раскроем подробные параметры этого действия. Укажем, какое именно окно будет открываться по клику мыши. Для этого достаточно перетащить требуемое окно (в нашем случае это Окно управления) на параметр Окно действия. В результате этого название окна появится в качестве значения параметра Окно.



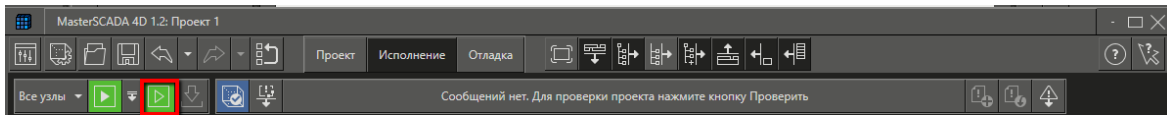
Переместим элемент Насос левее:



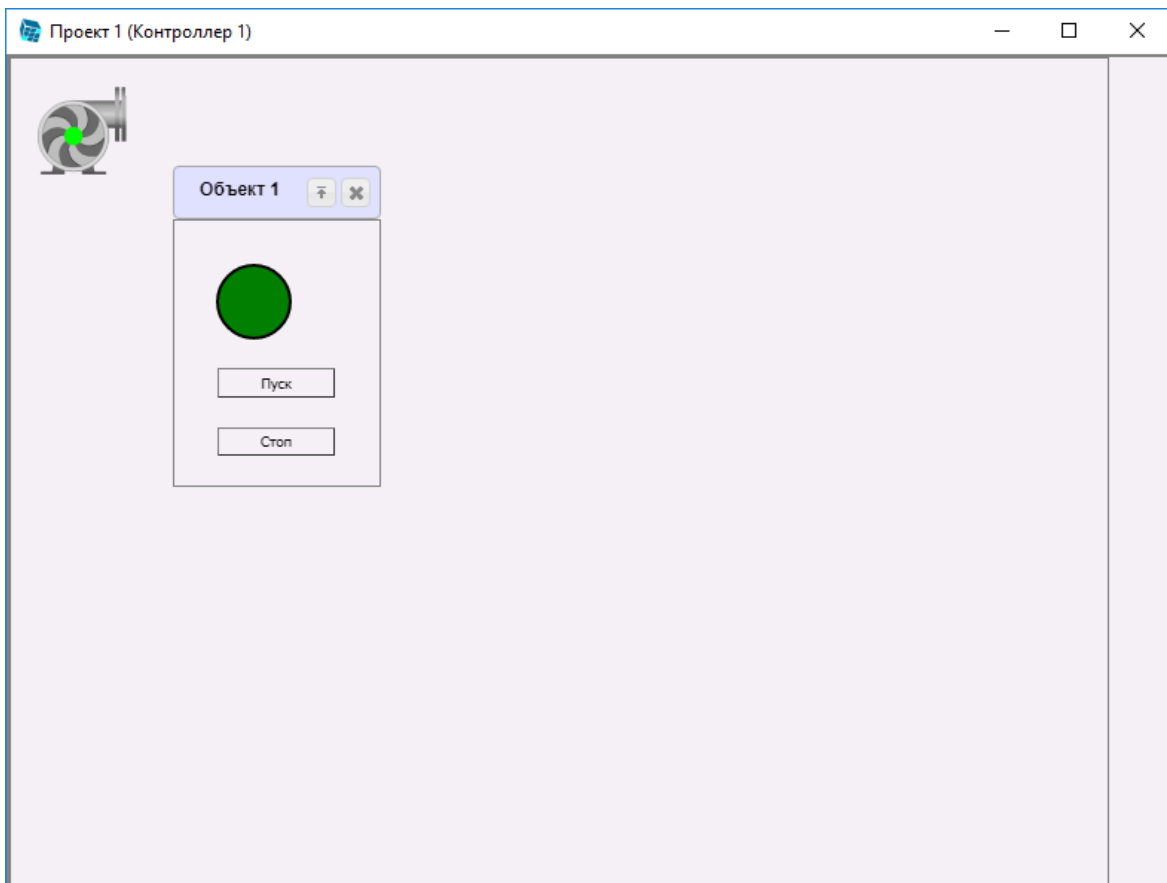
На этом конфигурирование нашего простого задания закончено.

2.2.2. БС. УРОК 2. ЗАПУСК

Сначала перейдем на вкладку Исполнение и, нажав иконку Эмуляция панели инструментов, проверим, что получилось в итоге.



Запустится режим эмуляции и откроется встроенный клиент визуализации (или браузер), на странице которого мы увидим изображение уже знакомого нам насоса. Кликнем по нему левой кнопкой мыши – откроется всплывающее окно с кнопками Пуск, Стоп и индикатором состояния. Нажмем кнопку Пуск – насос включится.

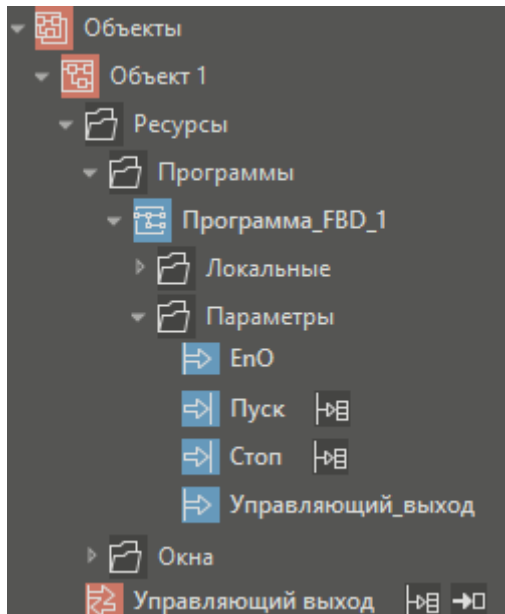


Нажмем кнопку Стоп – насос остановится. Всё, как и в уроке 1, за исключением того, что органы управления вынесены в отдельное окно.

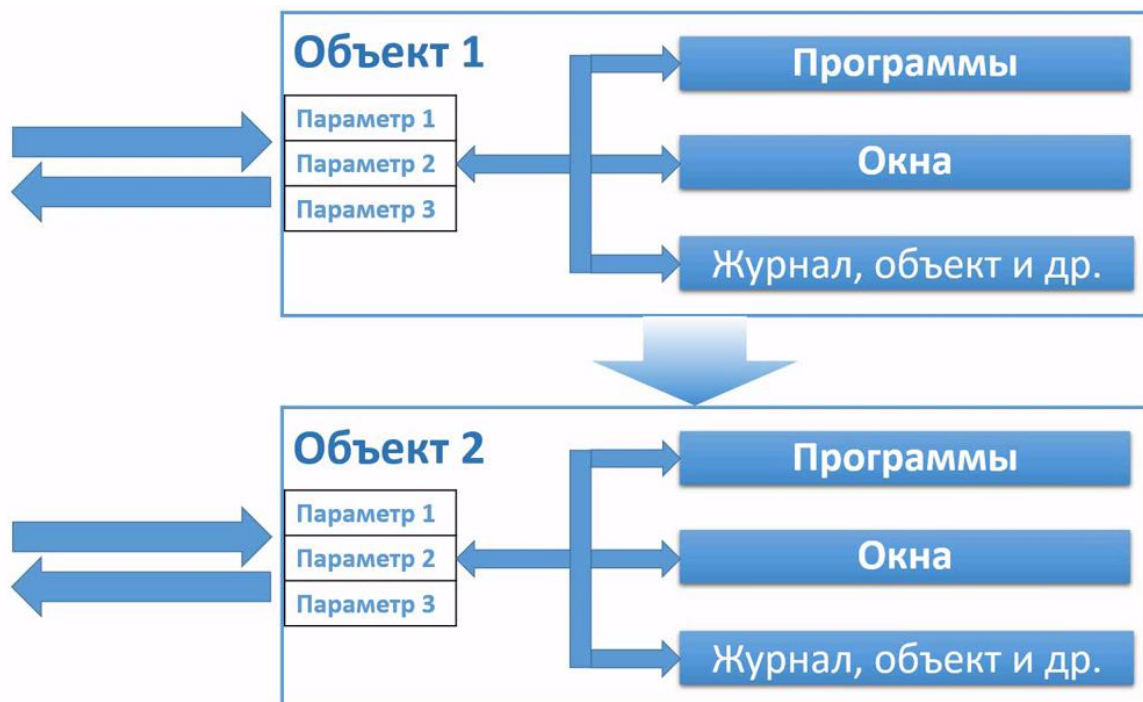
2.3. БС. УРОК 3

2.3.1. БС. УРОК 3. ОБЪЕКТНЫЙ ПОДХОД ПРИ СОЗДАНИИ ПРОЕКТА

Для начала рассмотрим, что представляет собой объект. В **MasterSCADA 4D** объект – это именованная совокупность, объединяющая разнородные элементы проекта и связи между ними.



В объект могут входить программы, окна, отчеты и другие элементы, в том числе и вложенные объекты. Различные параметры этих элементов могут быть связаны между собой. Сам объект тоже имеет параметры – они предназначены для его связи с "внешним миром", что-то наподобие клеммника. Основное назначение этого принципа организации объекта – облегчение процедуры тиражирования. Копия такого объекта создается с сохранением всей внутренней структуры, включая межэлементные связи внутри объекта. Разработчику останется лишь подключить внешние.

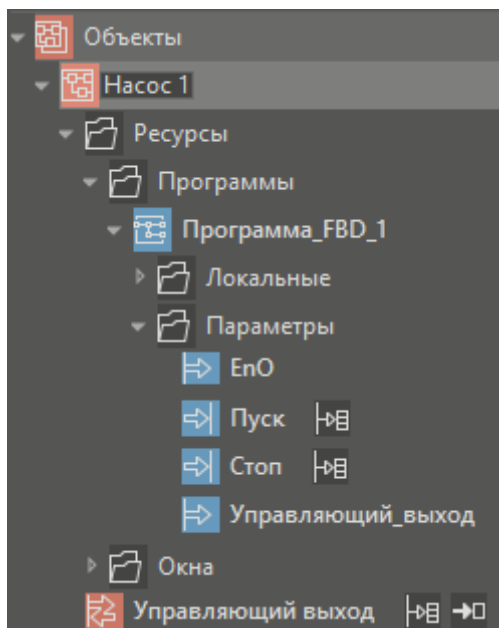


Теперь посмотрим как это можно применить на практике.

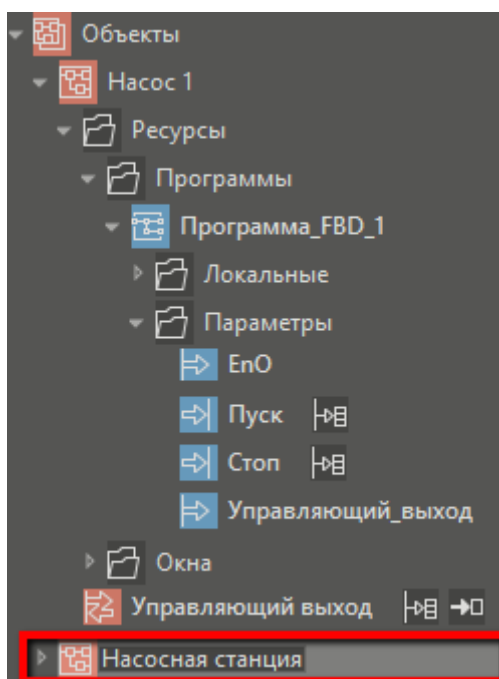
В предыдущем уроке мы создали объект, состоящий из окна с изображением насоса, окна управления этим насосом, вызываемого по нажатию мышью на изображении насоса, и программы, реализующей логику управления насосом. Объект имеет один параметр для связи с внешним миром – это управляющий выход, сигнал с которого поступает на физический дискретный выход модуля контроллера.

А теперь зададимся целью создать насосную станцию, состоящую из нескольких насосов.

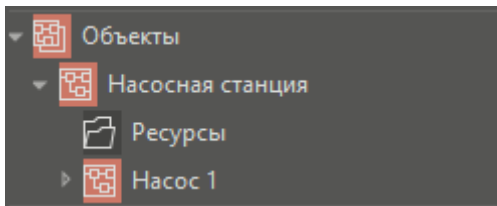
Переименуем созданный ранее объект в Насос 1.



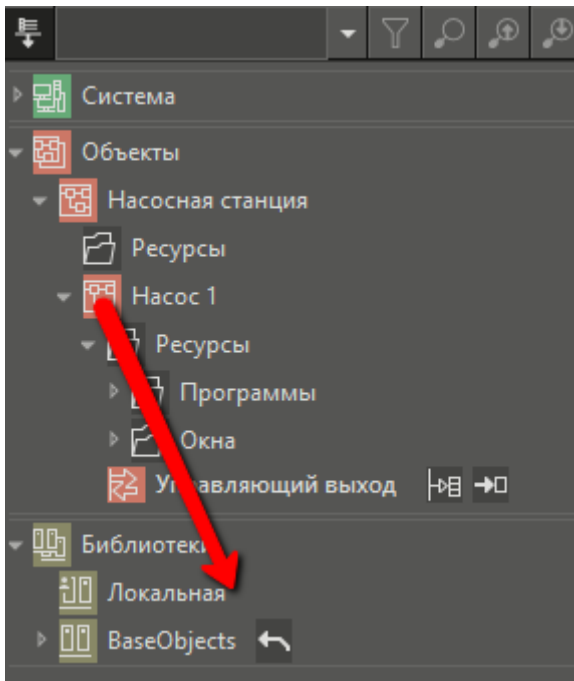
В дереве объектов создадим ещё один объект и назовём его Насосная станция.



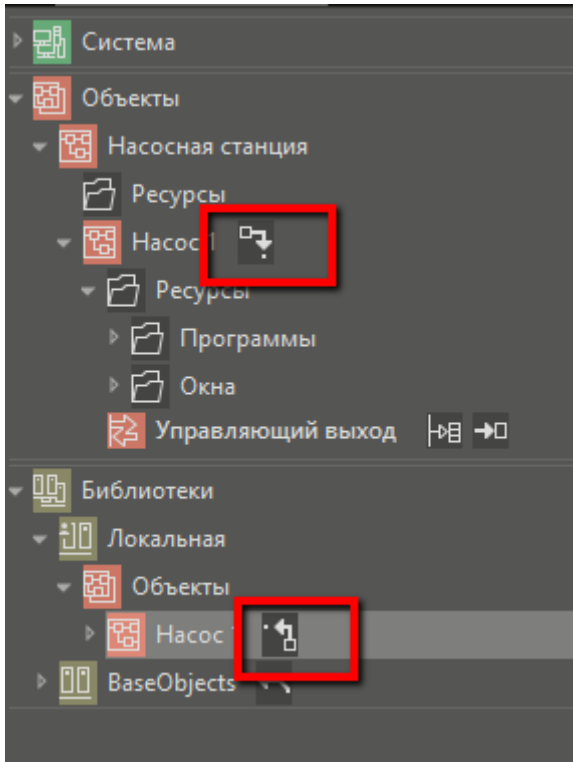
Теперь перетащим левой кнопкой мыши насос в насосную станцию. У нас получился корневой объект Насосная станция, у которого есть вложенный объект Насос 1.



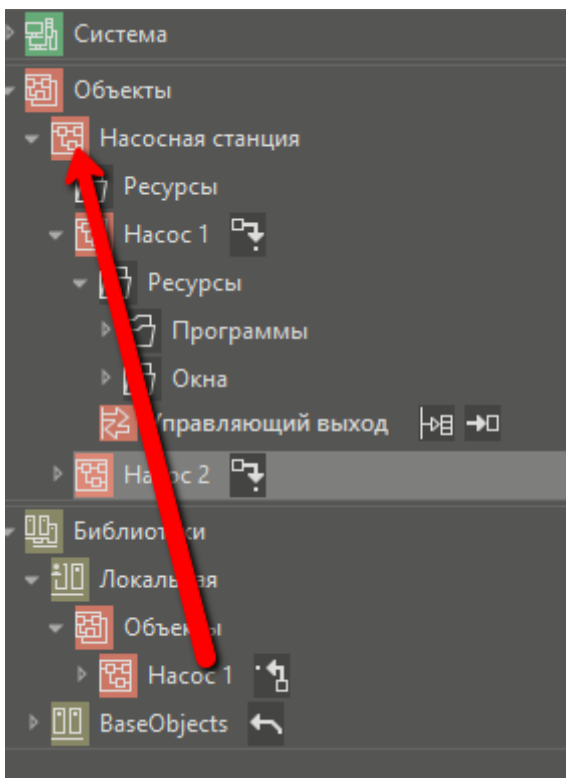
Перенесём его в библиотеку Локальная.



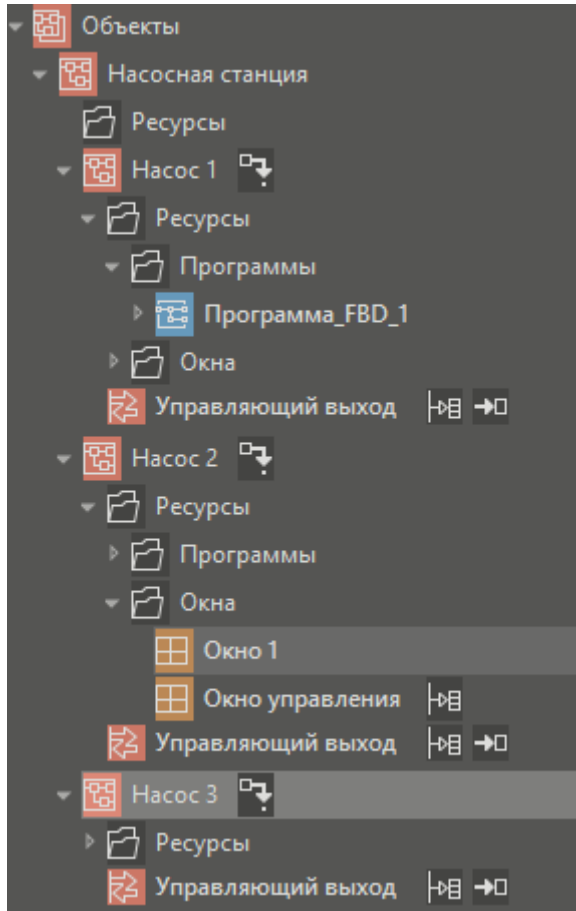
Теперь насос стал библиотечным элементом, а насос в дереве объектов – его наследником, о чём нам говорит специальный значок рядом с именем и всплывающая подсказка.



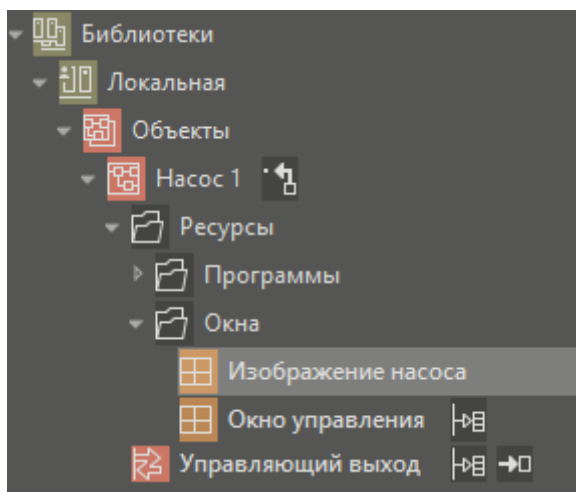
Перетащим библиотечный насос в насосную станцию. Получаем два насоса.



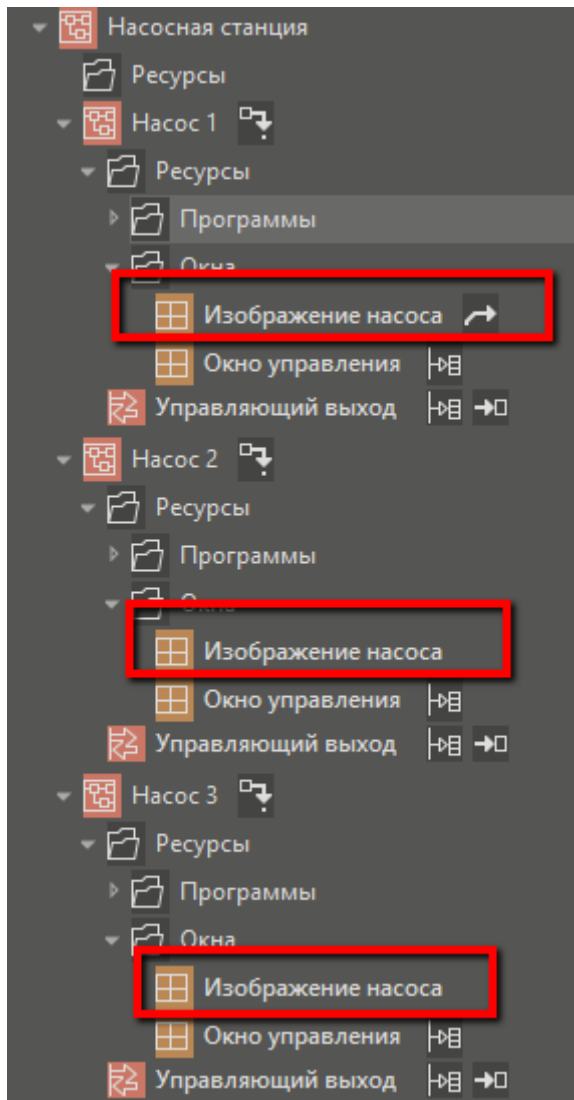
Повторим эту процедуру ещё раз – теперь насосная станция включает уже три насоса, каждый из которых имеет один параметр для связи с внешними элементами или устройствами (Управляющий выход), а внутри содержит окна и программу управления.



Все насосы в дереве объектов являются наследниками библиотечного. Это означает, что изменение чего-либо в библиотечном насосе отразится во всех его наследниках, то есть будет унаследовано. Например, раскроем дерево библиотечного насоса до содержимого папки Окна и переименуем Окно 1 в Изображение Насоса:

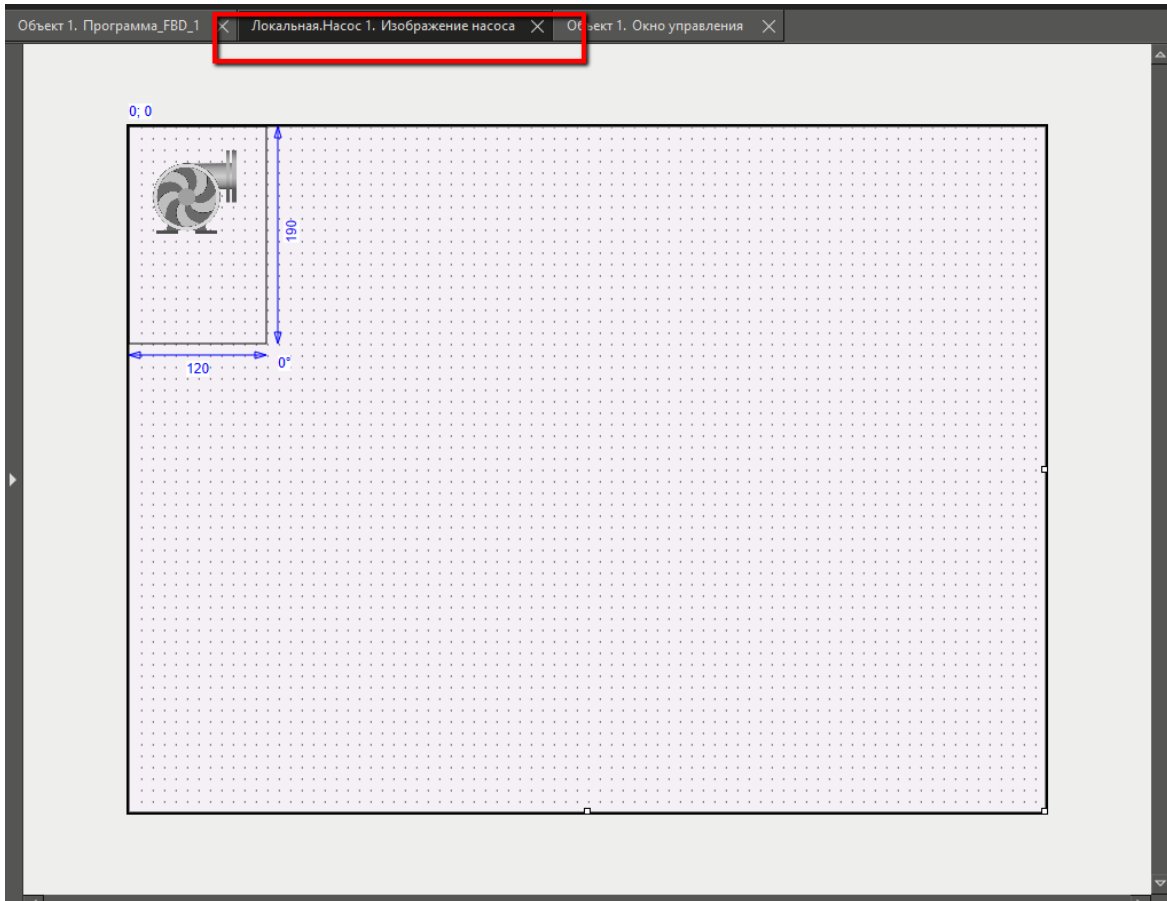


Увидим, что в дереве объектов это окно тоже переименовалось:

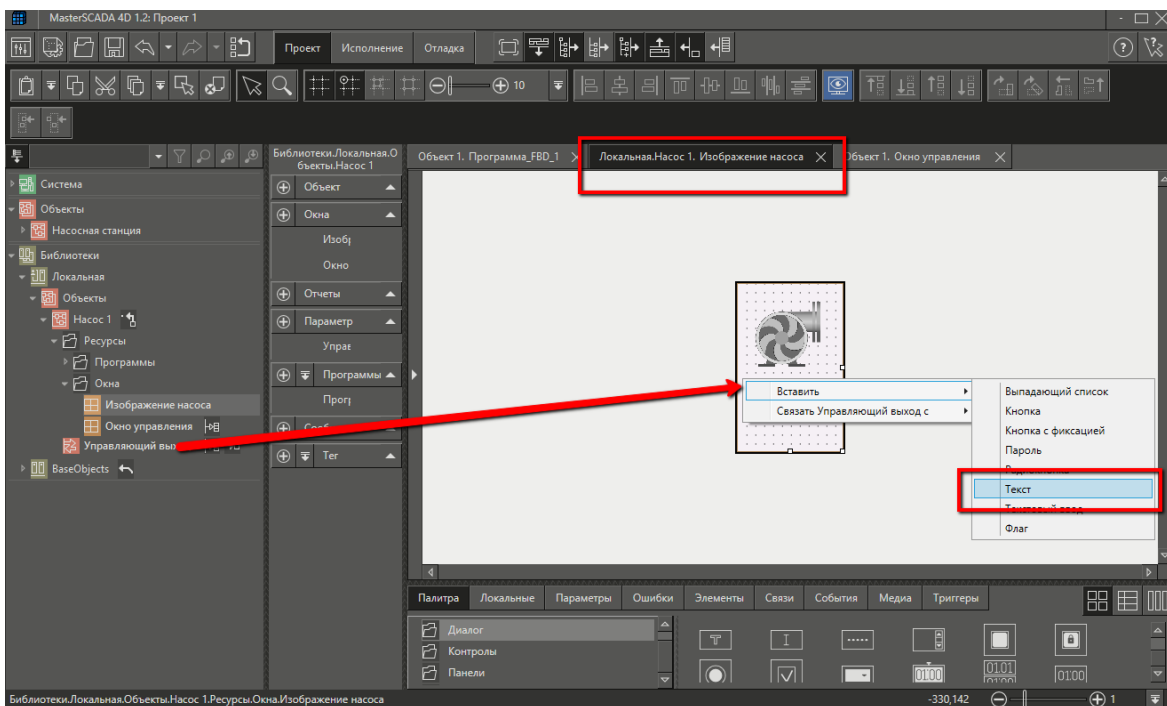


То есть изменение унаследовалось. Строго говоря, изменения будут наследоваться не всегда, а только до тех пор, пока не будет отредактирован сам наследник, и он не станет отличаться от библиотечного элемента. После этого наследование изменений будет сохраняться только для части наследника, совпадающей с библиотечной. Подробнее эти механизмы мы рассмотрим отдельно в других уроках.

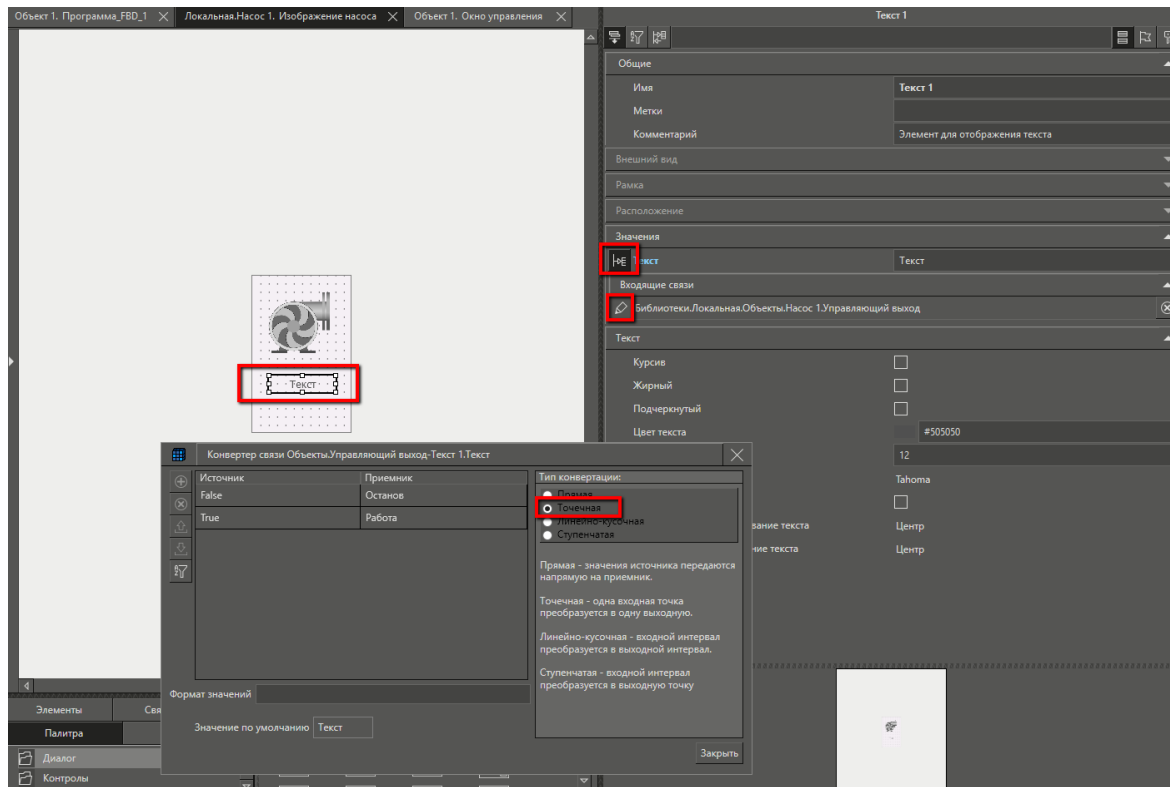
А сейчас внесём ещё некоторые изменения. Откроем окно Изображение Насоса из библиотеки и сделаем его компактнее (окно наследуемого насоса закроем):



Добавим надпись, отражающую состояние насоса. Для этого перетащим правой кнопкой мыши параметр Управляющий выход из библиотечного насоса (мы сейчас работаем с библиотечным элементом), и из открывшегося контекстного меню выберем Вставить.Текст:

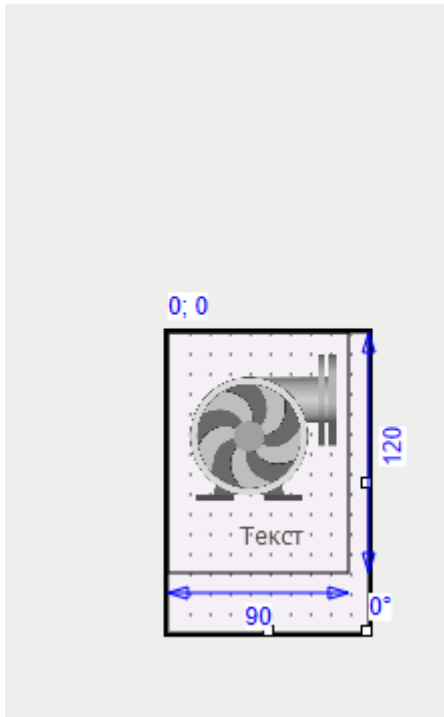


Очень часто рядом с изображением объекта отображают какой-нибудь основной параметр его работы, а иногда и несколько. Для клапана – это его положение; для двигателя с частотным преобразователем – частота, и так далее. Мы выведем просто состояние. Перейдем на панель свойств элемента Текст, откроем конвертор на связи параметра Текст. Выберем точечную конвертацию, и для значения False укажем Останов, а для значения TRUE – Работа.

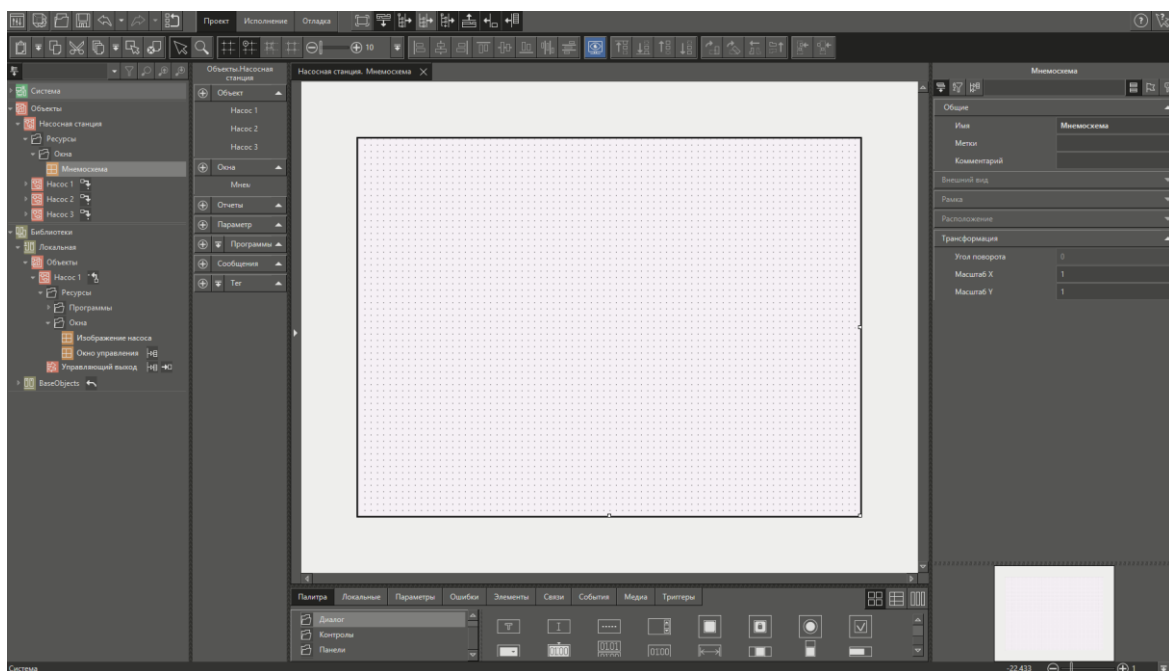


Закроем конвертор.

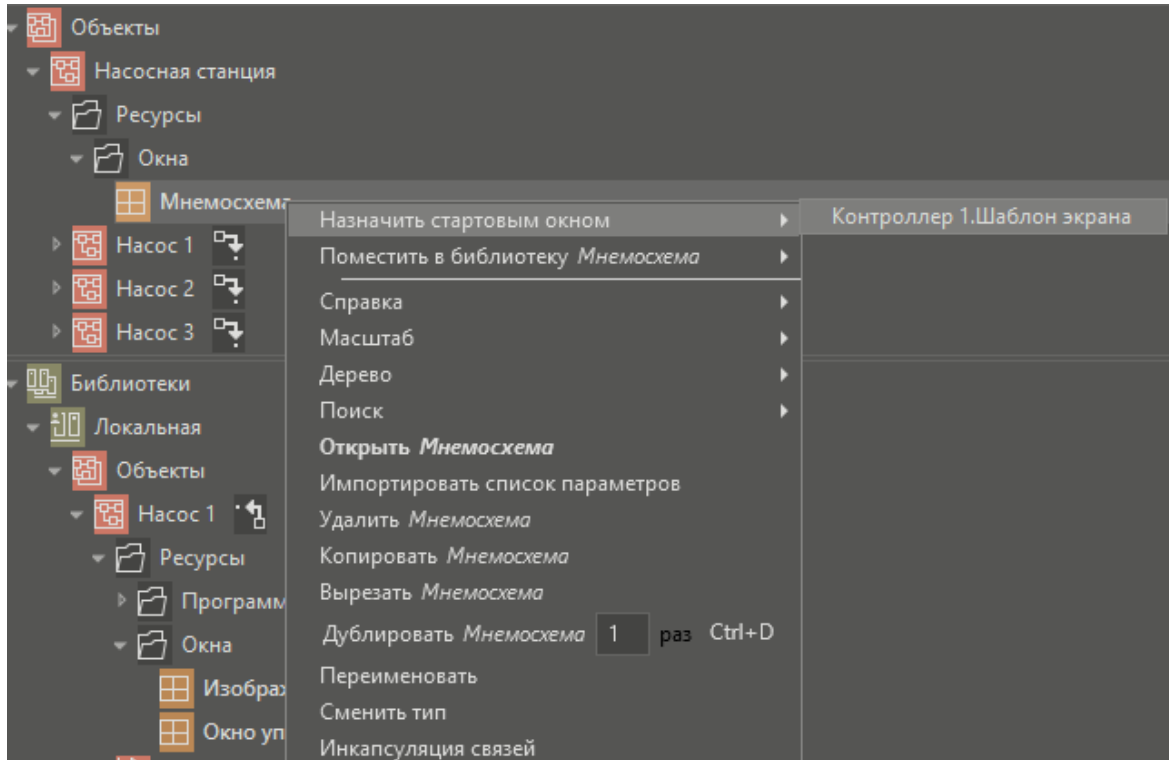
Уменьшим размер окна до размеров картинки:



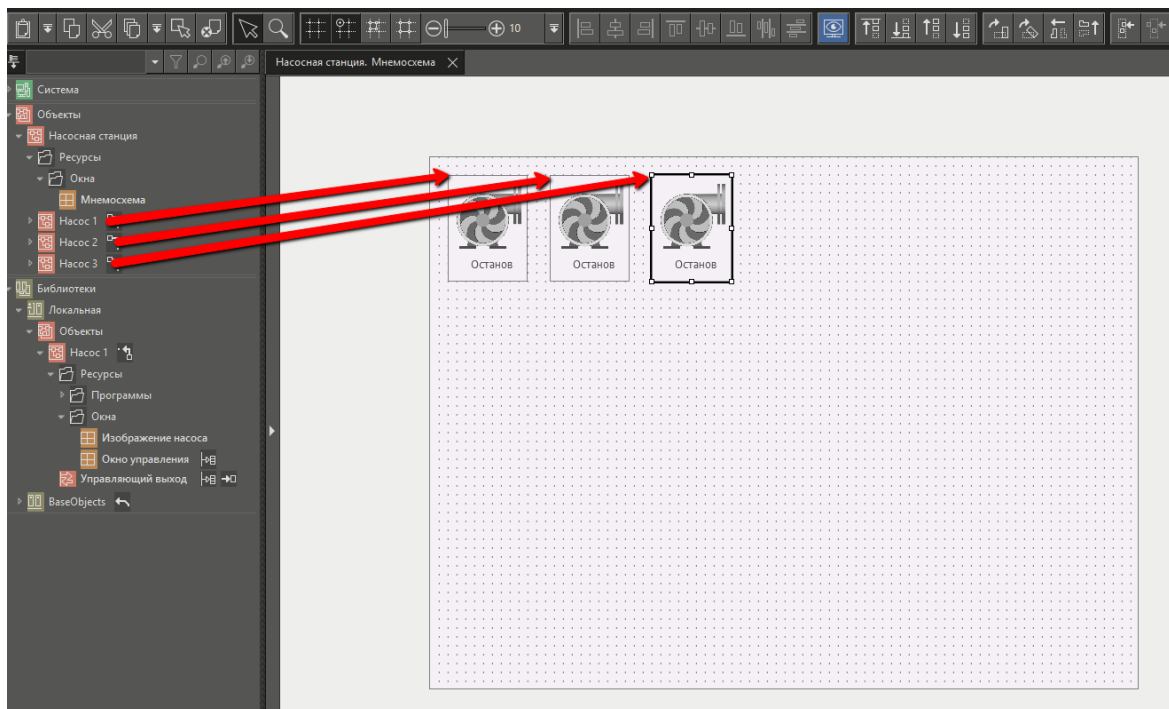
Добавим в насосной станции окно и назовём его Мнемосхема:



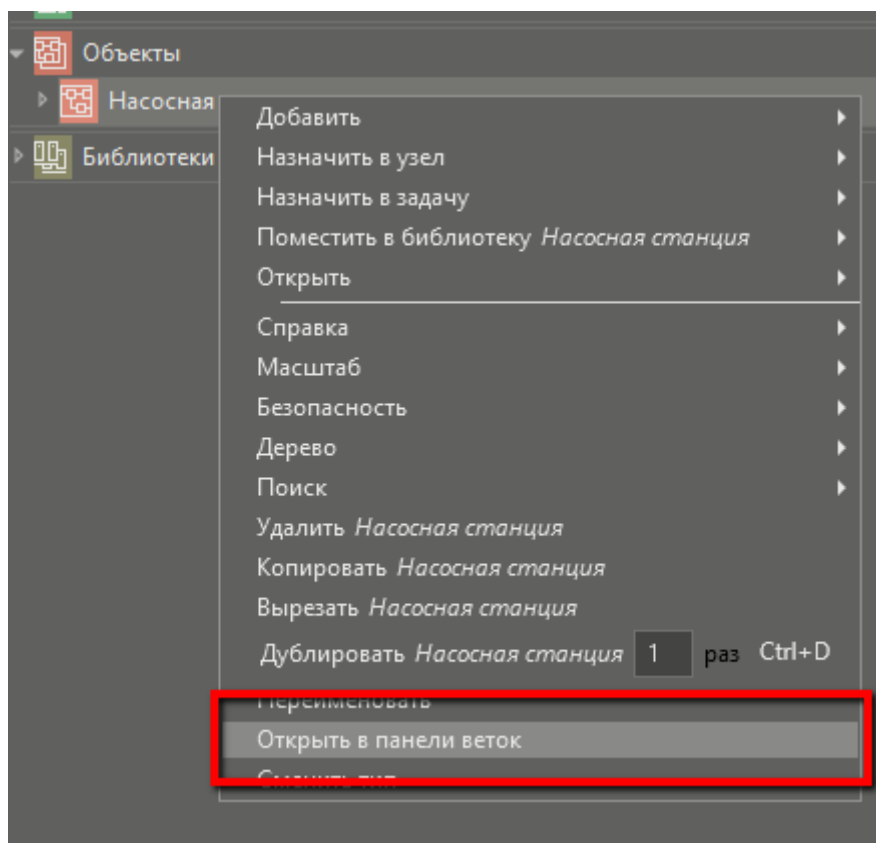
Назначим Мнемосхему стартовым окном узла при помощи контекстного меню:

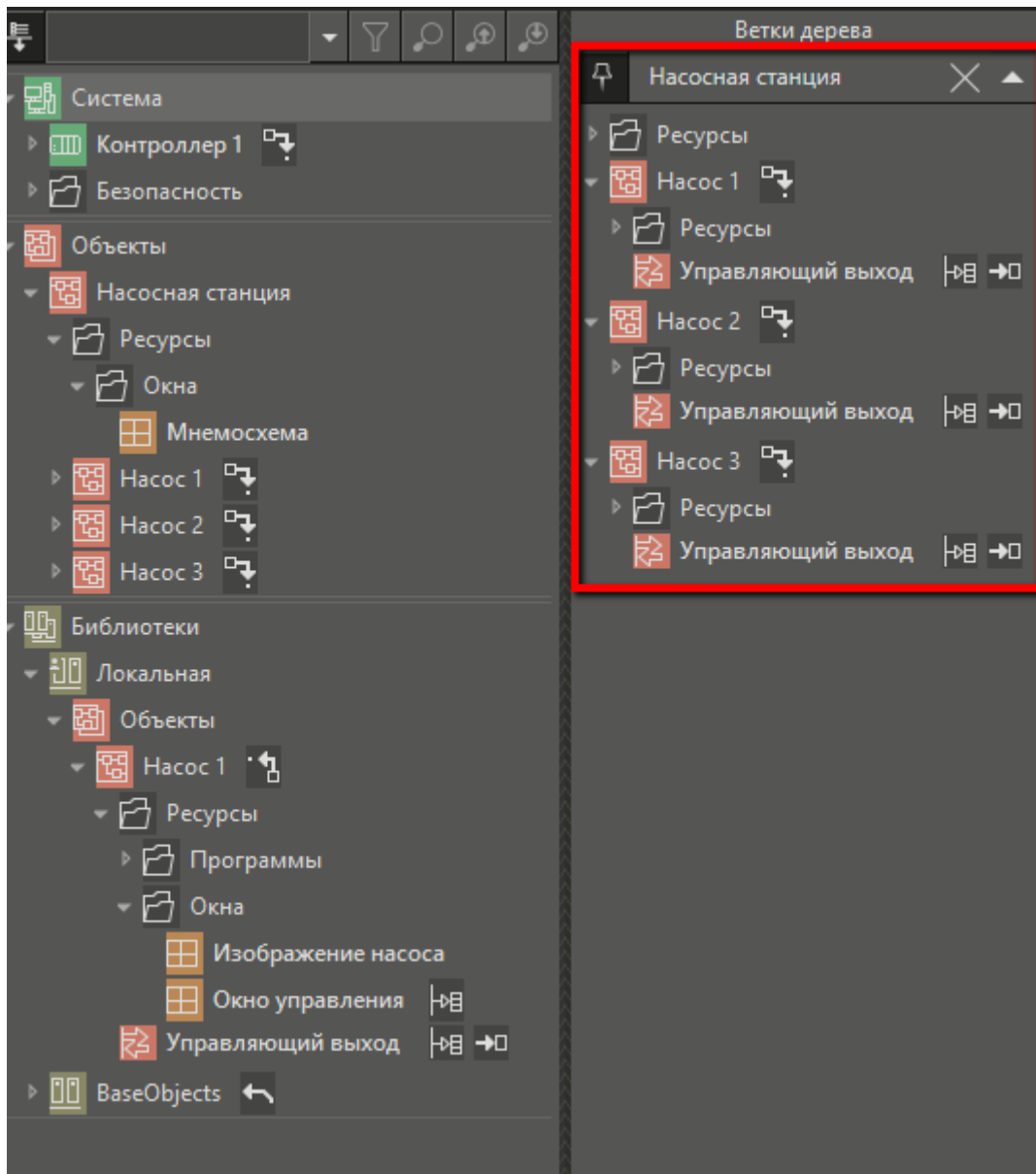


Теперь перетащим поочередно все насосы в окно:

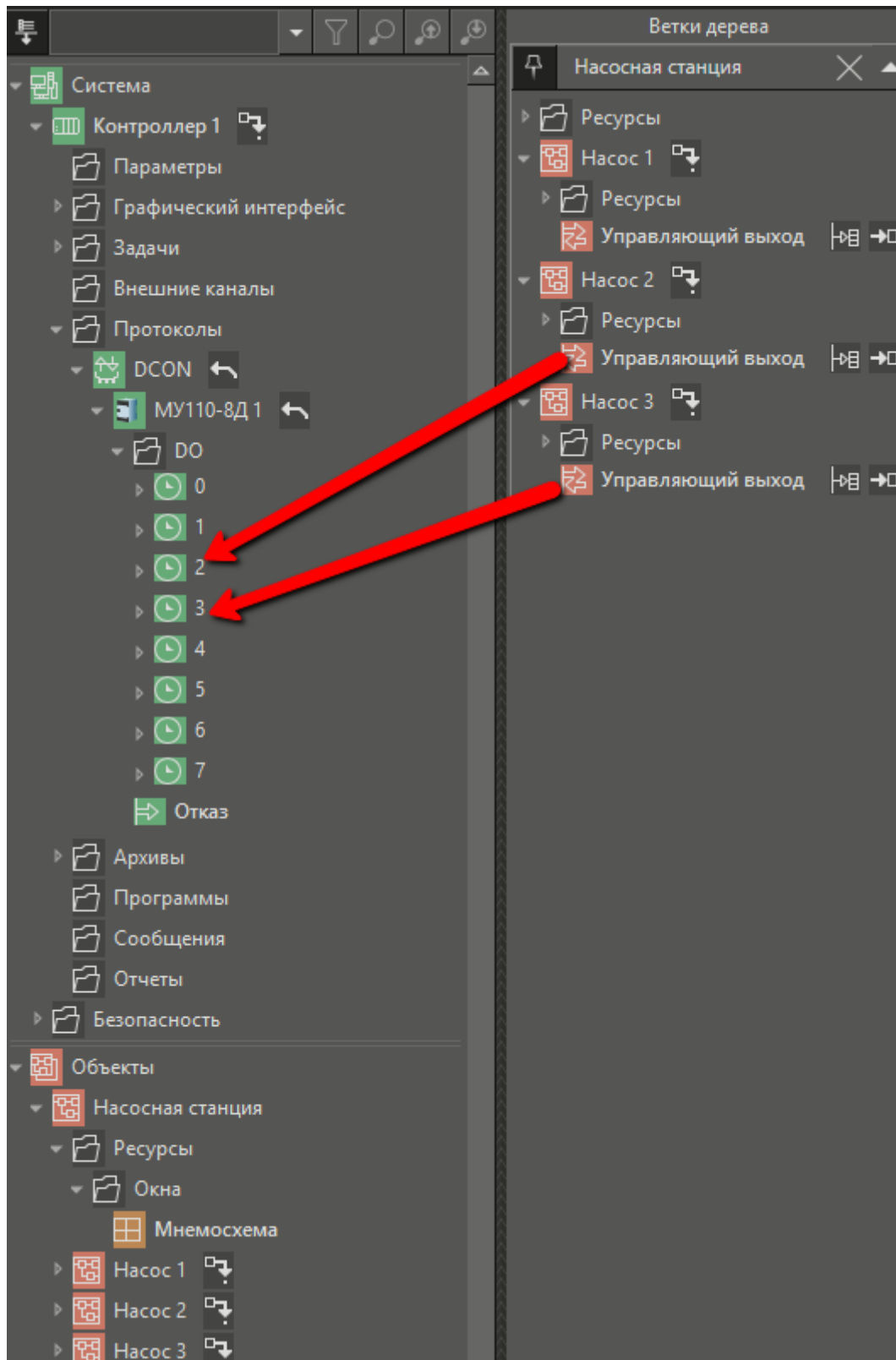


Свяжем выходы новых насосов с физическими выходами модулей ввода/вывода (мы подключили их в Уроке 1). Связывание удобнее производить, открыв деревья связываемых элементов в параллельных окнах. Нажмем правой кнопкой мыши на объект Насосная станция, и из контекстного меню выберем Открыть в панели веток:





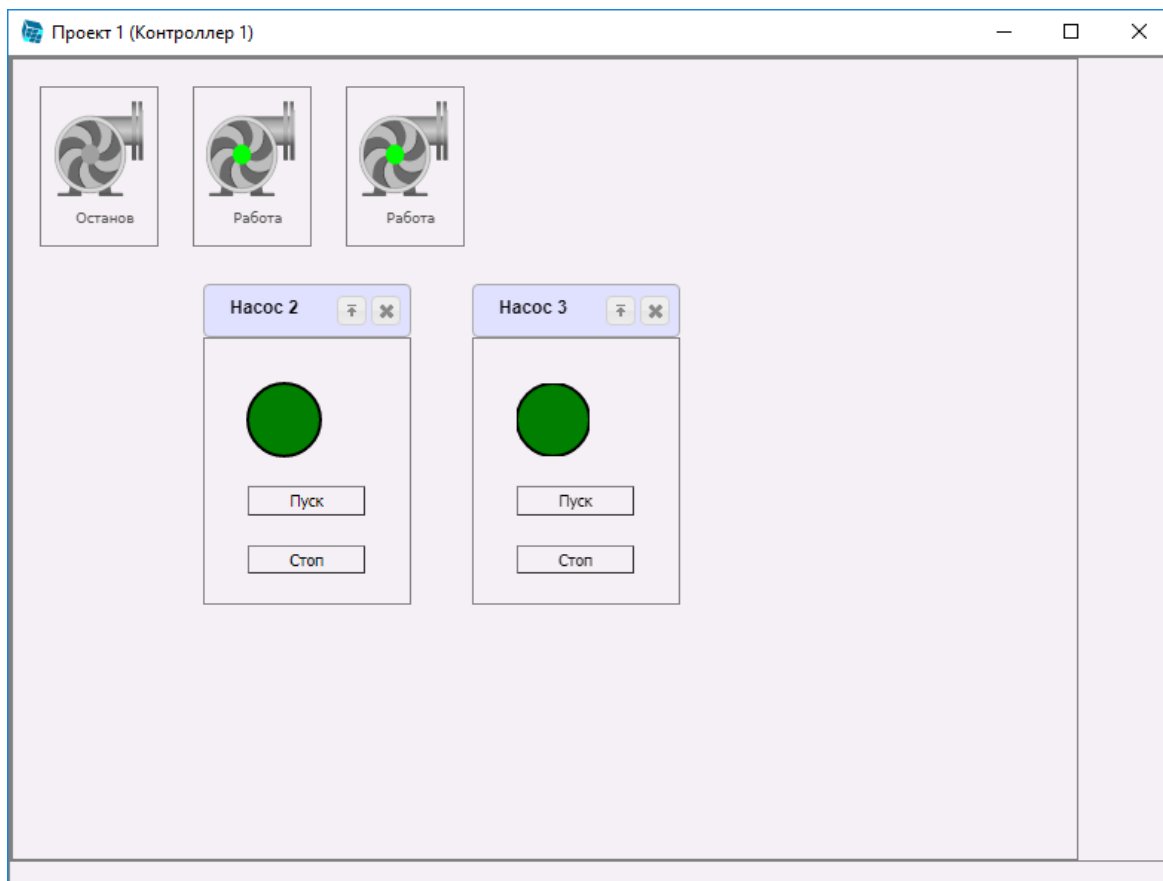
Первый насос у нас уже связан. Теперь свяжем параметр Управляющий выход второго и третьего насоса с последующими дискретными выходами модуля:



На этом наша насосная станция готова!

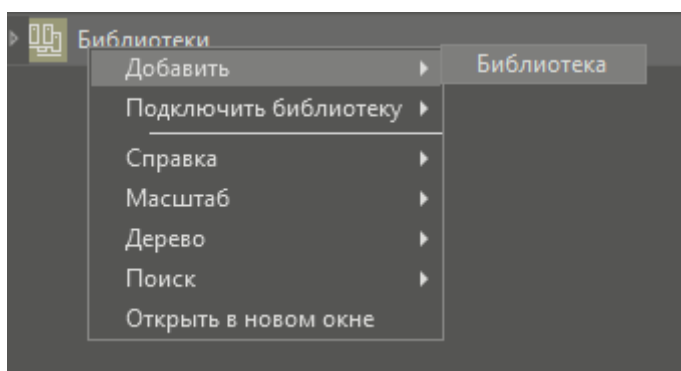
2.3.2. БС. УРОК 3. ЗАПУСК

Запустим режим исполнения. В браузере мы увидим созданную нами насосную станцию с тремя насосами, каждый из которых управляется независимо от других. Насосы можно включать или выключать в любой последовательности. Можно также закрывать ненужные окна, а затем снова их открывать и т.п.

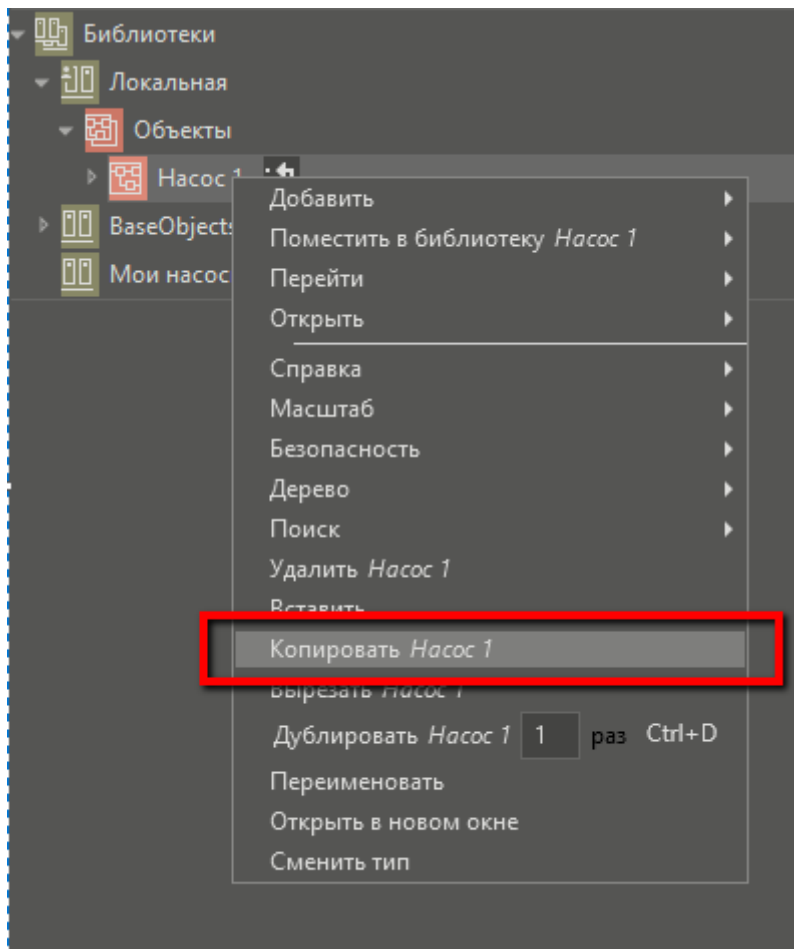


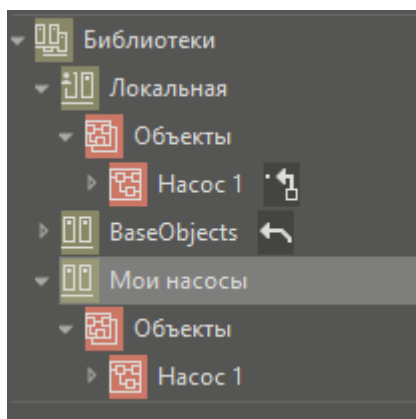
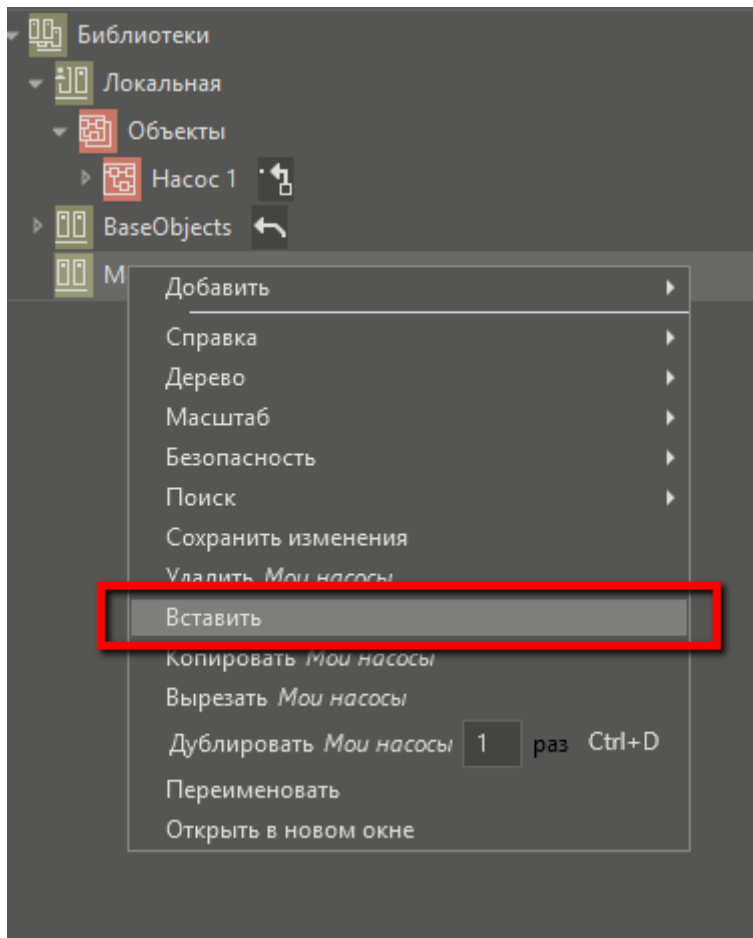
2.3.3. БС. УРОК 3. ОПЕРАЦИИ С БИБЛИОТЕКАМИ

В рамках создания этого проекта мы поместили тиражируемый элемент Насос в библиотеку Локальная. Библиотека Локальная доступна только в том проекте, в котором она была создана. Если же требуется, чтобы библиотечный элемент был доступен для использования и в других проектах, то необходимо создать его в другой библиотеке - в уже имеющейся или в новой. Наждем правой кнопкой мыши на элемент Библиотеки и выберем пункт Добавить - Библиотека:

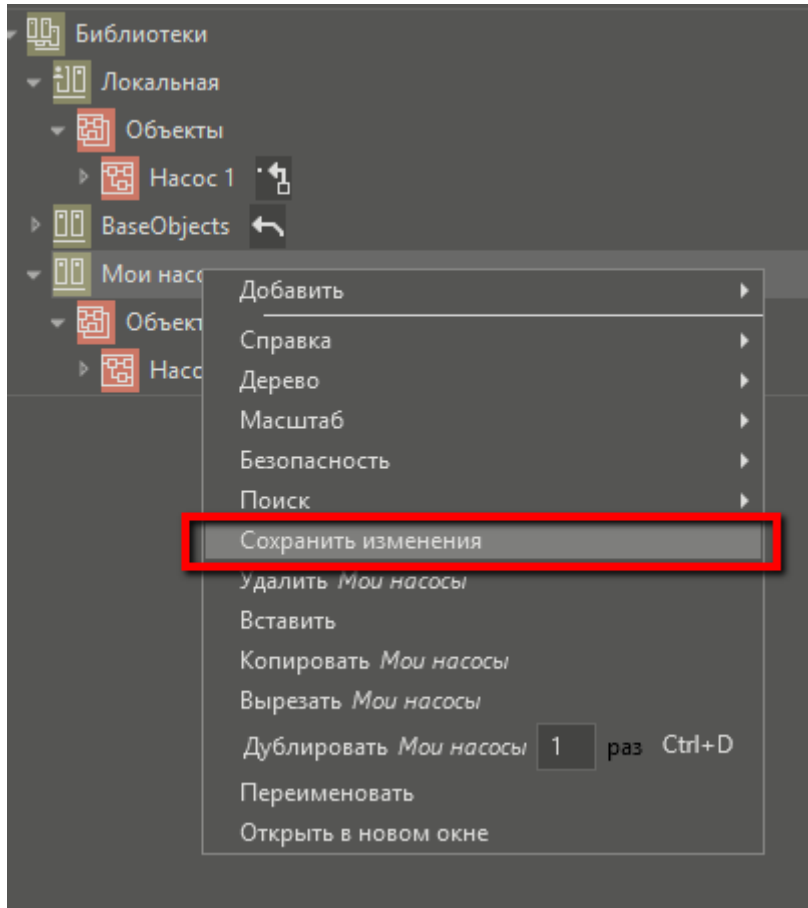


Назовём эту библиотеку Мои насосы. Скопируем созданный нами библиотечный насос из библиотеки Локальная и вставим его в библиотеку Мои насосы:

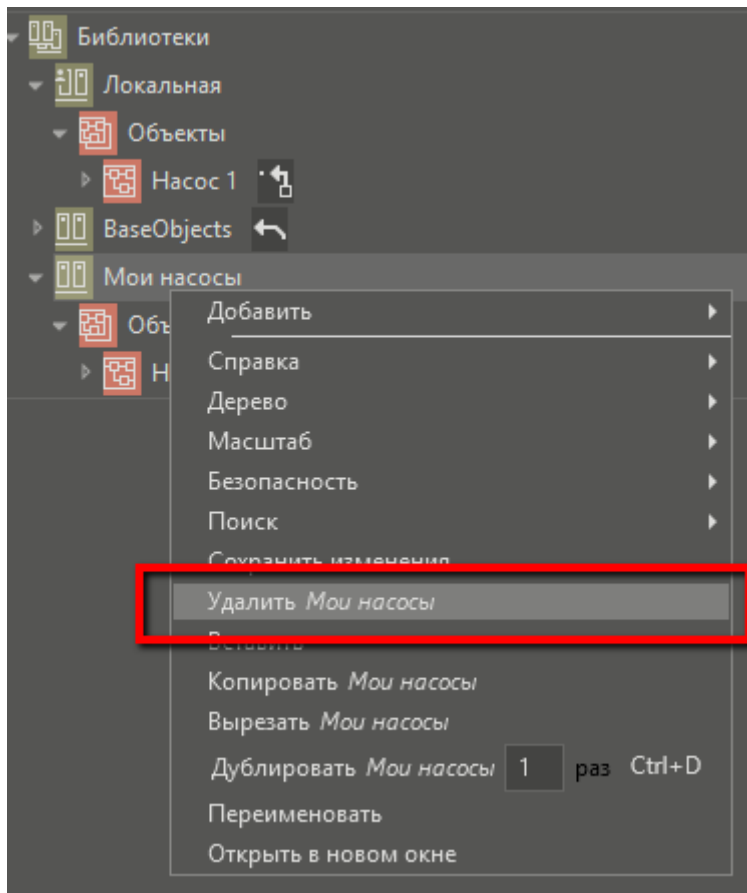




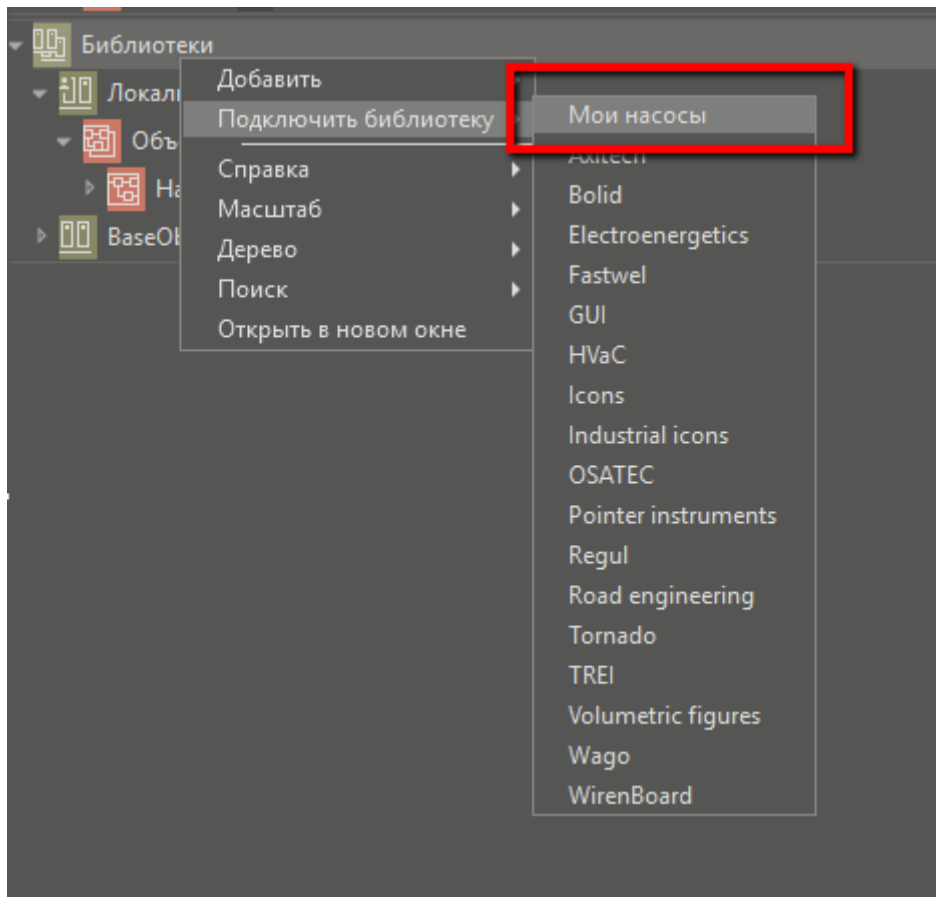
Затем нужно нажать правой кнопкой мыши на название библиотеки и выбрать пункт контекстного меню Сохранить изменения:



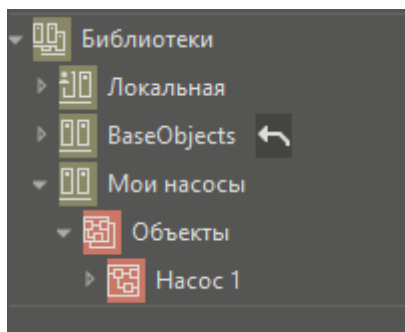
После этого библиотека сохранится на диске и будет доступна для подключения из других проектов через контекстное меню Подключить библиотеку. В данный момент она недоступна для подключения, т.к. уже подключена. Для того чтобы это продемонстрировать, удалим библиотеку Мои насосы из проекта через контекстное меню Удалить:



Теперь подключим её к проекту: Для этого нажмем правую кнопку мыши, выберем пункт Подключить библиотеку, и в открывшемся списке доступных библиотек выберем Мои насосы:



Библиотека подключилась и теперь её можно использовать.



3. ОБЗОР ИНСТРУМЕНТАЛЬНОЙ СРЕДЫ MASTERSCADА 4D

Основные характеристики

Понятия

Описание интерфейса

Демонстрационные проекты

Лицензионная политика

Принятые сокращения и обозначения

3.1. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ

MasterSCADA 4D – это продукт нового поколения SCADA-систем. По сравнению с предыдущей версией, в **MasterSCADA 4D** существенно расширены инструменты по созданию крупных распределенных систем с возможностью использования технологий Интернета, повышены удобство и гибкость, расширены возможности использования различных аппаратных платформ и операционных систем, увеличено число поддерживаемых уровней систем управления и реализована миграция функционала между уровнями.

В **MasterSCADA 4D** легко разрабатываются проекты любого масштаба и сложности.

К основным характеристикам **MasterSCADA 4D** относятся следующие.

Полная вертикальная интеграция – среда разработки **MasterSCADA 4D** обеспечивает включение в проект всех уровней системы управления. Это такие уровни, как:

- программируемые контроллеры;
- локальные HMI-панели;
- АРМ операторов;
- серверы;
- облачные сервисы.

Задачи для всех уровней разрабатываются в рамках единого проекта, единого информационного пространства. Более подробную информацию о том, какие именно устройства, ОС и процессоры поддерживаются, можно получить из раздела справочной системы **Инсталляция-Системные требования среды исполнения**.

Широкая кроссплатформенность – исполнительная система **MasterSCADA 4D** работает в большинстве операционных систем, распространенных в промышленности:

- Windows;

- Linux (различные разновидности, в том числе и операционные системы, одобренные для работы в военной отрасли);
- Android;
- QNX;
- Эльбрус;
- Unix.

Не исключено, что этот список будет расширяться в зависимости от требований той или иной отрасли. Существенным фактором является поддержка этих ОС на всех уровнях систем управления: ПЛК, локальных HMI-панелях, АРМ операторов, серверах, облачных сервисах.

Гетерогенность – реализация возможности использования в рамках одного проекта разнообразных аппаратных платформ, работающих под управлением разных операционных систем, а также различных средств передачи данных между узлами системы: от последовательных портов и GPRS до высокоскоростных оптических линий и Интернета, как универсальной среды передачи данных.

Для проектировщика не имеют значения различия ОС или механизмов передачи данных. Он просто добавляет в проект необходимые ему узлы и связи между ними.

Миграции функционала по вертикали системы управления – эта особенность дает проектировщику системы свободу в выборе уровня, на котором требуется выполнять те или иные функции. Так, например, часть задач визуализации можно перенести в контроллеры, а документирование и архивирование поднять в облачный сервис. Возможность миграции обеспечивается функциональной идентичностью исполнительных модулей для различных уровней систем.

Объектный подход к разработке проектов позволяет многократно сократить трудозатраты на создание и отладку проектов. Данный подход заключается в реализации типизации и наследования объектов – базового понятия в **MasterSCADA 4D**.

Под объектом в **MasterSCADA 4D** понимается именованная совокупность графического представления технологического объекта, его параметров, алгоритмов контроля и управления, окон управления и других доступных элементов проекта, в том числе, других объектов (см. также **Объект**).

Типизация, наследование обеспечивает многократное использование одного и того же объекта в качестве **экземпляра или наследника** как в рамках одного проекта, так и при разработке других систем. При изменении объекта в библиотеке обеспечивается соответствующее изменение всех его экземпляров (наследников) в проекте.

Двухслойная структура проекта – возможность разделения проекта на дерево объектов, в котором описывается графическая и математическая часть проекта, и дерево системы, в котором ведется работа с аппаратными средствами. Такое разделение позволяет вести разработку этих частей проекта независимо друг от друга..

Эта особенность **MasterSCADA 4D** также дает возможность легко переносить проект с одних аппаратных средств на другие, оставив в неизменном виде все программные решения.

Полноценная поддержка языков стандарта МЭК 61131-3 подразумевает использование языков этого стандарта не только для программирования задач в контроллерах, но также для разработки логики управления и вспомогательных скриптов на всех уровнях системы, включая графические клиенты, в том числе интернет-клиенты.

Поддержка облачных решений. В проект **MasterSCADA 4D** может быть добавлен один или несколько узлов типа Облачный сервис. Это полноценный узел проекта, который может обмениваться с другими узлами. Для него можно создавать любые задачи, доступные в проекте **MasterSCADA 4D**.

Отличие узлов типа Облачный сервис заключается в том, что созданные для них задачи загружаются по указанному URL в Интернет на исполнение в облаке. Облачный сервис для запуска этих узлов можно арендовать или приобрести у компании "МПС Софт" или у партнеров компании. Количество одновременно запущенных проектов на одном облачном сервисе ограничивается только его техническими возможностями. Демо-проекты, которые входят в состав среды разработки, и которые можно загрузить локально, работают уже не один год в облачном пространстве компании «МПС Софт».

Унификация системы визуализации. В качестве основной технологии для реализации графического интерфейса в **MasterSCADA 4D** используется стандарт HTML5. Это позволяет использовать в качестве инструмента графического интерфейса не только клиент визуализации **MasterSCADA 4D**, но и любые устройства, имеющие в составе программного обеспечения интернет-браузер, в том числе смартфоны и планшеты. Графическая подсистема – полностью векторная, с возможностью произвольного масштабирования окон (мнемосхем) и с динамизацией любых свойств графических элементов (положение, размер, поворот, цвет и пр.). Для разработки графического интерфейса оператора в состав **MasterSCADA 4D** включен редактор HMI. Для взаимодействия **MasterSCADA 4D** с веб-браузером или окном клиента **MasterSCADA 4D**, в котором в реальном времени отображаются разработанные графические мнемосхемы, используется веб-сервер NGINX.

Расширенный состав библиотек. Для удобства разработчика добавлены различные библиотеки:

- библиотека готовых алгоритмов (помимо собственных алгоритмов поддерживана также библиотека OSCAT (<http://www.oscat.de>), содержащая более 300 алгоритмов);
- библиотека графических элементов, изображений, текстур, иконок;
- библиотека готовых объектов для самых различных отраслей промышленности и других сфер деятельности. Объект в этой библиотеке кроме своего графического изображения содержит алгоритмы, параметры и окна;

- возможность создавать пользовательские библиотеки с готовыми решениями (алгоритмы, графические элементы, объекты).

Стандартные функции SCADA. Наряду с новыми функциями, **MasterSCADA 4D** обладает всеми стандартными функциями, которые должны быть в любом SCADA-приложении:

- система многослойного и многосерверного архивирования;
- система разработки и генерации различных отчетов о работе проекта;
- система тревожных сообщений с возможностью квитирования этих сообщений;
- система разграничения полномочий операторов;
- открытая объектная модель, позволяющая разработать программы, автоматизирующие создание проекта. Например, можно написать скрипт, автоматически создающий элементы проекта путем импорта информации о параметрах технологического объекта из базы данных или таблицы Excel;
- открытость системы для встраивания внешних протоколов, алгоритмов, графических систем, а также для интеграции с другими программами;
- поддержка мультиязычности (см. **Язык среды**);
- удобные инструменты навигации по проекту.

К достоинствам **MasterSCADA 4D** следует отнести также наличие исполнительных систем для большинства **контроллеров**, представленных на отечественном рынке.

В состав **MasterSCADA 4D** включены **демонстрационные проекты**, а также проекты Быстрого старта, снабженные пошаговой инструкцией по их созданию.

3.2. ПОНЯТИЯ

Для того чтобы лучше понимать наши справочные материалы и видеопримеры, сначала необходимо договориться о терминологии.

Общие понятия

Среда разработки (инструментальная среда) – Windows-приложение для разработки проектов, которые затем компилируются и загружаются в среду исполнения.

Среда исполнения (исполнительный или исполняемый модуль) – приложение, которое устанавливается на различные устройства (компьютеры, контроллеры, операторские панели, планшеты), и в которое загружаются проекты, созданные в среде разработки. Для каждой отдельной ОС или процессора предусмотрена своя среда исполнения.

Клиент визуализации (тонкий клиент) – приложение (браузер с поддержкой HTML5), которое используется для отображения графических окон (мнемосхем) в режиме исполнения. Подключается к среде исполнения.

Встроенный клиент визуализации. Как правило, речь идет о Windows-приложении, разработанном компанией «МПС Софт» для отображения графических окон (мнемосхем) в режиме исполнения. Входит в состав среды разработки и загружается автоматически при установке среды исполнения для Windows. Существует также встроенный клиент визуализации для ОС Android.

Проект – это то, что создает пользователь в среде разработки. В одном проекте можно создать сразу несколько различных программ для нескольких узлов.

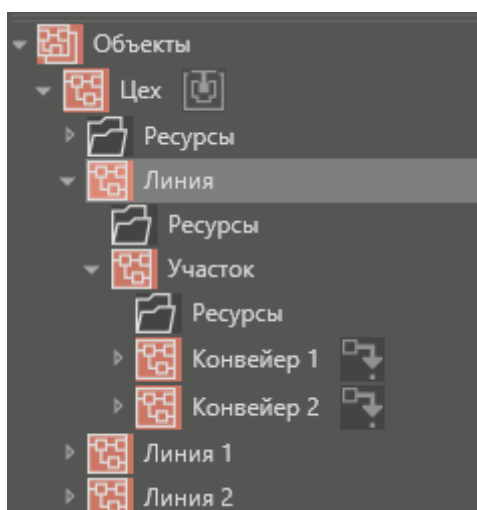
Скомпилированный проект – это один или несколько наборов файлов, полученных в результате преобразования (компиляции) проекта, которые загружаются в среду исполнения. Если, в случаях когда необходимо организовать межузловое взаимодействие, в среде разработки создается единый проект для нескольких устройств (узлов), то для каждого устройства после компиляции подготавливается и загружается в него своя часть, заранее определенная разработчиком для работы именно в этом устройстве (узле).

Типизация – механизм, обеспечивающий многократное использование одного и того же библиотечного объекта как в рамках одного проекта, так и при разработке других систем.

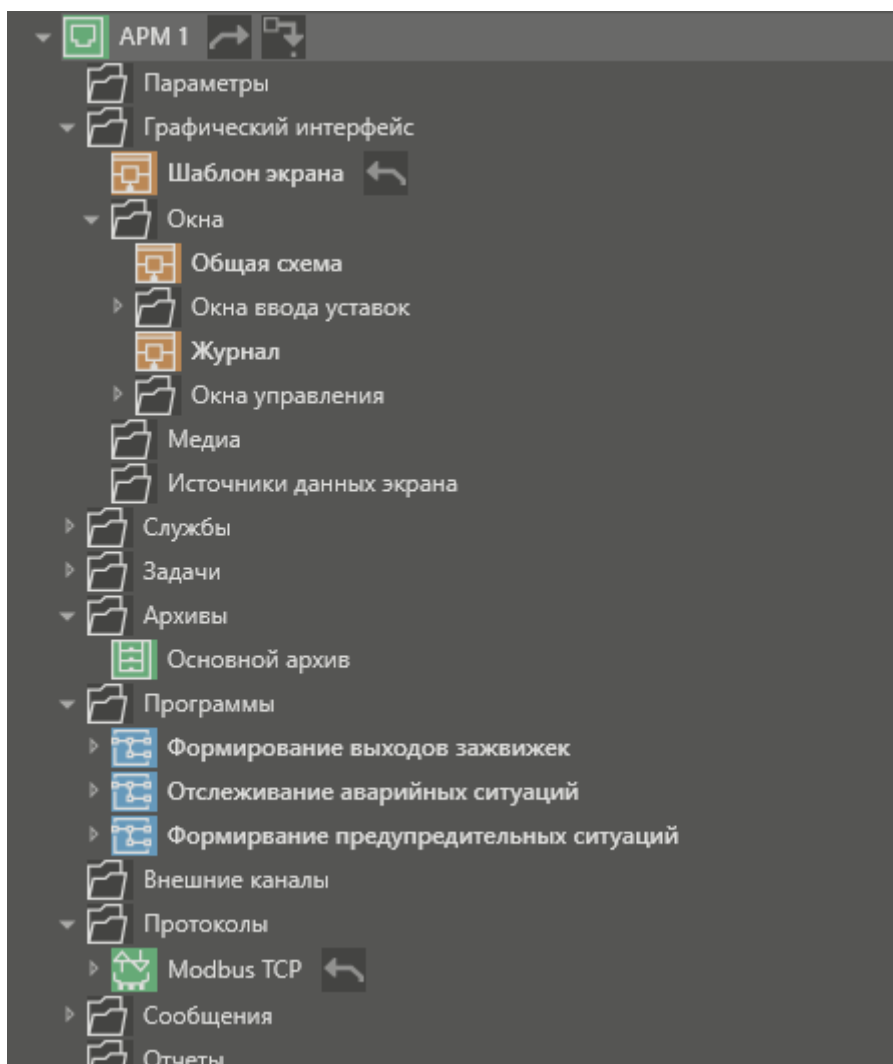
Наследование – механизм, обеспечивающий при изменении типа объекта в библиотеке соответствующее изменение и всех экземпляров этого объекта в проекте.

Объектный метод разработки проектов – метод разработки, при котором реальному физическому объекту (цеху, линии, участку, датчику, задвижке) соответствует виртуальный объект в проекте, к которому применимы графические изображения, программы, сообщения (тревоги). Разработка проекта ведется независимо от выбранного оборудования, т.к. логическая часть (объекты) и физическая часть (протоколы, модули ввода/вывода и т.п.) разделены. При таком подходе время на разработку типовых проектов, а также проектов, в которых встречается много однотипных элементов (например, конвейеров, задвижек, насосов), сокращается за счет использования возможностей наследования и типизации.

Структура проекта при объектном методе разработки:



Линейный метод разработки проектов – метод разработки, при котором программы, окна и другие элементы создаются единым списком либо разделение не имеет связи с реальными объектами. В этом случае проект, как правило, создается в узле.



Элементы проекта

Дерево проекта MasterSCADA – структура, отображающая проект со всеми его элементами (свойствами, связями и т.п.) в виде дерева. Работа над проектом непосредственно в самом дереве упрощает поиск и групповые операции. Для отображения дерева предназначено окно структуры проекта. Другие окна/редакторы могут отображать определенные части дерева проекта в соответствующем виде: например, FBD-диаграмма отображает элементы FBD-программы в графическом виде.

Дерево системы – корневая часть дерева проекта, в которой содержится используемое оборудование. Специфические элементы дерева системы служат для настройки связи проекта с внешними устройствами посредством протоколов, внешних каналов, встроенных каналов. В дереве системы существует возможность создавать как программы, так и окна. Однако, в этом случае разрабатываемый проект полностью привязан к конкретному оборудованию, и при его замене будет затруднена адаптация проекта. Такой способ

разработки называется линейным. При работе в **MasterSCADA 4D** лучше использовать объектный подход, при котором в дереве системы описывается только физическая часть проекта.

Дерево объектов – корневая часть дерева проекта, которая содержит логические и графические элементы, описывающие объект автоматизации.

Дерево библиотек – отдельное дерево, содержащее элементы, которые могут быть использованы в проекте. Любой элемент проекта является наследником какого-либо библиотечного элемента: стандартного, предоставленного разработчиком, или пользовательского.

Узел – верхний (корневой) элемент дерева системы: контроллер, АРМ, сервер, облако и др., которому соответствует реальное или виртуальное оборудование. Количество узлов в дереве соответствует количеству исполнительных систем, которые планируется использовать. Если в проект добавлены, например, два узла, то предполагается использование двух взаимодействующих устройств, запрограммированных средствами **MasterSCADA 4D**. То есть данные от одного устройства планируется передавать в другое устройство, используя при этом внутренний протокол обмена **MasterSCADA 4D**. Все узлы различаются между собой настройками по умолчанию. Например, если в проект добавлен в качестве узла какой-либо контроллер, имеющий встроенный модуль ввода/вывода, то это отразится в дереве системы: в нем появится группа **Встроенные модули**.

Объект – именованная совокупность графического представления технологического объекта, его параметров, алгоритмов контроля и управления, окон управления и других доступных элементов проекта (в том числе других объектов) (см. также подробное описание **Объект**).

Канал – элемент дерева системы, представляющий собой структуру, содержащую один или несколько параметров, а также настройки, специфические для того или иного протокола, необходимые для получения данных из устройств. Данный элемент может находиться в группах **Протоколы**, **Встроенные каналы**, **Встроенные модули**. В проекте канал связывается с параметрами объекта.

Параметр – переменная проекта. Может находиться в программе, в окне, в объекте и др.

Глобальный параметр – параметр, который находится в дереве системы в группе **Параметры** и может использоваться в программах ST без дополнительного объявления внутри программы.

Связь

Внутренний протокол обмена – протокол передачи данных между устройствами, запрограммированными средствами **MasterSCADA 4D**. Используется для связи двух параметров проекта, работающих в связанных между собой узлах.

Программирование

Программа (термин стандарта IEC 61131-3) – это логическая совокупность (сборка) элементов и конструкций языка программирования, выполняющая требуемую обработку сигналов для обеспечения системы программируемых контроллеров функциями управления исполнительными механизмами или процессами. Наследование программ запрещено.

Функциональный блок (ФБ) (термин стандарта IEC 61131-3) – это программный элемент (POU), который вырабатывает произвольное количество выходных элементов данных. В отличие от функции значения всех выходов, а также некоторых внутренних переменных ФБ сохраняются между вызовами и при одних и тех же входных значениях ФБ может вырабатывать различные выходные значения. Поэтому в библиотеке вначале создается тип ФБ, а в программном элементе создаются экземпляры этого ФБ (локальные переменные). В библиотеке допускается наследование типов ФБ.

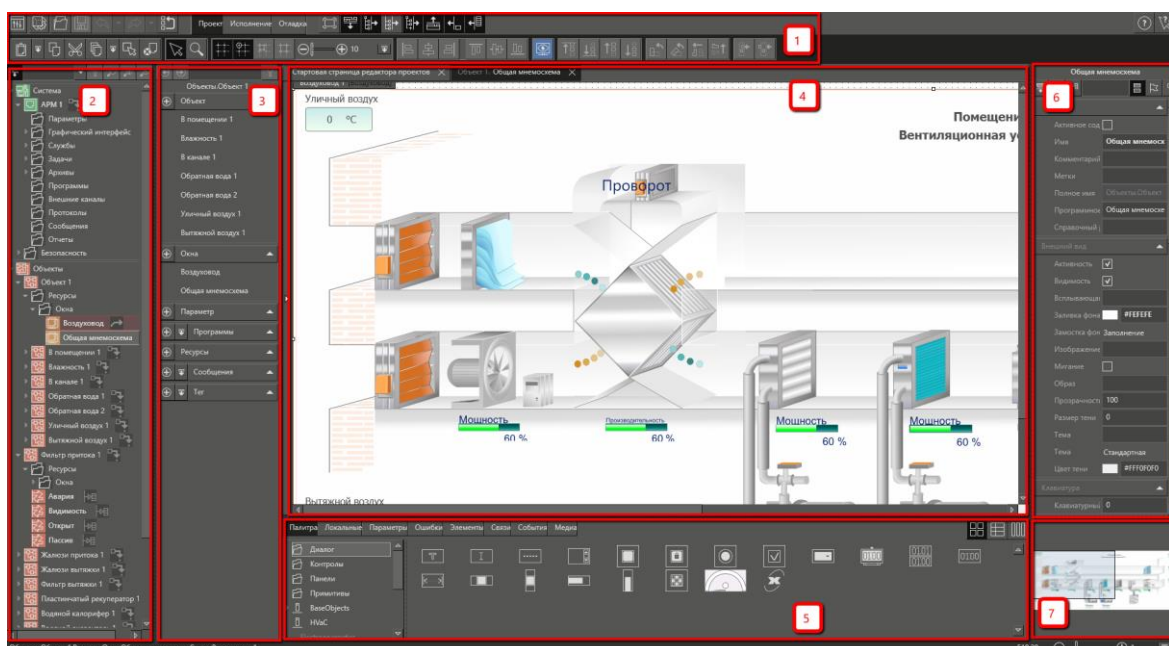
Функция (термин стандарта IEC 61131-3) – элемент языка (POU), который во время выполнения обычно вырабатывает результат в виде одного элемента данных (в т.ч. массив или структуру) и, возможно, дополнительные выходные параметры. Имеет произвольное количество входных параметров. Функция не содержит внутренних переменных, значения которых сохраняются между вызовами, поэтому при одних и тех же значениях входов всегда производит одни и те же выходные значения.

Локальные переменные – параметры, которые находятся только в рамках какого-либо элемента: программы, ФБ, окна.

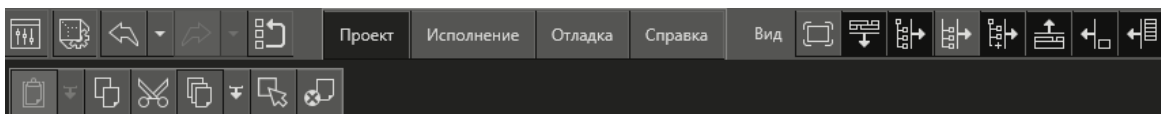
3.3. ОПИСАНИЕ ИНТЕРФЕЙСА

Рассмотрим основные элементы интерфейса среды разработки.

Общий вид:



1. Область меню и панели быстрого доступа.



Панель быстрого доступа позволяет получить быстрый доступ к управлению средой исполнения и проектом, выполнить операции Отменить/Повторить, а также просмотреть историю действий разработчика проекта.

Далее в этой же строке идут пункты Основного меню программы. Каждому пункту меню соответствует своя панель инструментов, располагающаяся под верхней строкой области меню, или панель быстрого запуска. В верхней строке также размещаются кнопки управления видимостью интерфейса.

2. Дерево проекта состоит из трех частей: **дерево системы, дерево объектов, дерево библиотек.**

3. Контекстное меню предназначено для быстрого добавления элементов в текущий проект. Все пункты в данном окне аналогичны пунктам контекстного меню объекта, вызываемого нажатием правой кнопки мыши на этот объект.

4. Рабочая область, в которой происходит редактирование элемента - программы или окна. Элемент открывается для редактирования двойным нажатием на него левой кнопкой мыши.

Подробное описание интерфейса приведено в разделе **Меню редактора проекта.**

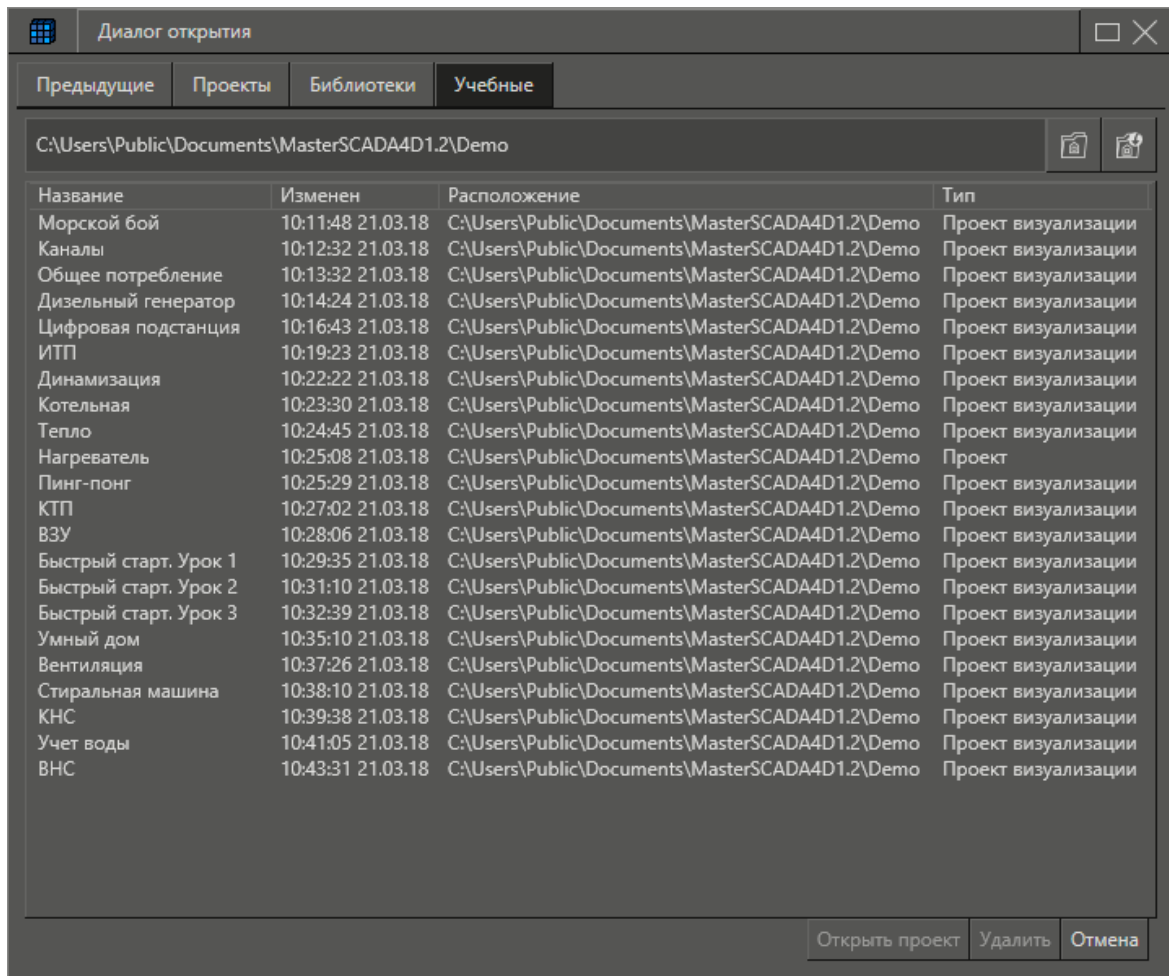
5. Легенда содержит элементы, которые используются в том или ином редакторе на закладке Палитра, и ряд вспомогательных закладок для отладки, анализа и работы. Набор закладок легенды меняется в зависимости от открытого редактора. Более подробно некоторые закладки рассматриваются в разделах, посвященных редакторам среды разработки.

6. Панель свойств отображает настройки текущего элемента, выбранного для работы в каком-либо из деревьев.

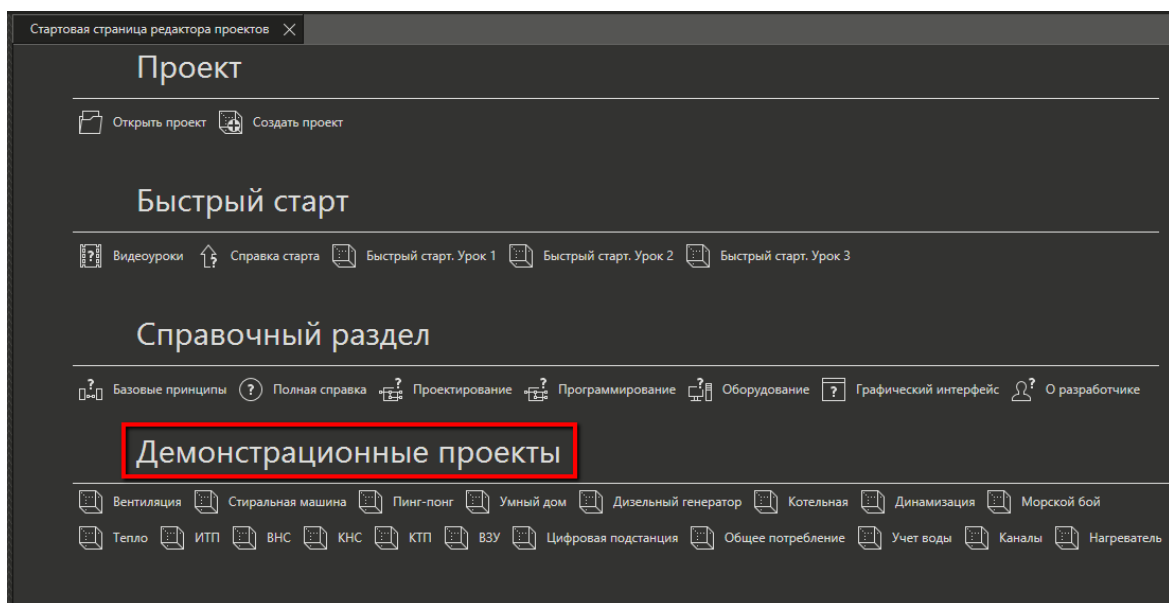
7. Миникарта – средство для навигации в окнах редакторов: FBD-, SFC-, LD-диаграмм и окон.

3.4. ДЕМОНСТРАЦИОННЫЕ ПРОЕКТЫ

В состав MasterSCADA 4D включен ряд демо-проектов; эти проекты отображает **вкладка Учебные** диалога открытия проекта.



А также стартовая страница (см. **Команда Стартовая страница**).



В комплект поставки **MasterSCADA 4D** входит также быстрый старт, описанный в отдельном разделе справочной системы.

3.5. ЛИЦЕНЗИОННАЯ ПОЛИТИКА

Лицензионное соглашение на исполнительную систему пакета прикладных программ для ЭВМ MasterSCADA™

ПРЕДУПРЕЖДЕНИЕ! Настоящее лицензионное соглашение (далее "Соглашение") является юридически значимым документом, которое заключается между вами - юридическим или физическим лицом, имеющим статус «конечный пользователь MasterSCADA™», которое легальным способом приобрело лицензию на использование пакета прикладных программ MasterSCADA™ (далее MasterSCADA™), и обладателем исключительного права на MasterSCADA™ (свидетельство о государственной регистрации программы для ЭВМ №2010617624, №2012619903, №2012619676) далее ПРАВООБЛАДАТЕЛЕМ.

MasterSCADA™ включает в себя набор программ различной функциональной комплектности, записанный на электронных носителях или выложенный для скачивания через Интернет, аппаратный ключ защиты (если он входит в комплект поставки), печатные материалы и/или электронную документацию.

1. ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

- 1.1. MasterSCADA™ - набор программ различной функциональной комплектности.
- 1.2. Правообладатель – лицо, которому принадлежат исключительные права на Программное обеспечение.
- 1.3. Конечный пользователь – физическое или юридическое лицо, которое приобрело лицензию MasterSCADA™ легальным способом.
- 1.4. USB-ключ аппаратной защиты – способ лицензирования MasterSCADA™ подразумевающий использование USB-ключа защиты информации без которого невозможен запуск программы.
- 1.5. Программный сервер лицензирования – программа, без которой невозможен запуск программы MasterSCADA™.
- 1.6. Аппаратная защита – способ лицензирования MasterSCADA™ подразумевающий привязку к аппаратной части оборудования (серверы, программируемые логические контроллеры, панели операторов, устройства сбора и передачи данных).
- 1.7. Декомпиляция – трансляция скомпилированного кода в эквивалентный исходных код на языке программирования высокого уровня.
- 1.8. Дизассемблирование – получение исходного текста программы на ассемблере из скомпилированного машинного кода.
- 1.9. Купон – документ описывающий размер и период оказания услуг технической поддержки MasterSCADA™.

2. ВВОДНЫЕ ПОЛОЖЕНИЯ

- 2.1. Начиная использовать MasterSCADA™, вы, посредством конклюдентного действия, вступаете в правоотношение с ПРАВООБЛАДАТЕЛЕМ, а также подтверждаете вашу волю на возникновение таких правоотношений.
 - 2.2. Правоотношения между вами и ПРАВООБЛАДАТЕЛЕМ регулируются, среди прочего, нормами статей 1235-1238 Гражданского Кодекса РФ.
 - 2.3. Вы, как конечный пользователь MasterSCADA™, несете гражданско-правовую ответственность за нарушение прав и законных интересов ПРАВООБЛАДАТЕЛЯ, совершенное вами при использовании MasterSCADA™. Кроме того, начиная использование MasterSCADA™, вы подтверждаете, что безоговорочно, без каких-либо условий, соглашаетесь с условиями настоящего Соглашения.
 - 2.4. В случае, если вы не согласны с условиями настоящего Соглашения, вы обязаны прекратить использование MasterSCADA™. В случае невыполнения вами данного требования, ПРАВООБЛАДАТЕЛЬ вправе понудить вас к этому с помощью любого и всех средств защиты прав, предусмотренных законодательством Российской Федерации.
 - 2.5. Содержание настоящего Соглашения может быть изменено ПРАВООБЛАДАТЕЛЕМ в любой момент без предварительного уведомления конечных пользователей. Начиная использование MasterSCADA™, вы подтверждаете свою обязанность исполнять положения настоящего Соглашения, в том числе и те, которые могут быть введены в него в будущем. С актуальной версией лицензионного соглашения вы можете ознакомиться на сайте www.masterscada.ru в сети «Интернет».
3. ЛИЦЕНЗИЯ НА MasterSCADA™
- 3.1. MasterSCADA™ является собственностью ПРАВООБЛАДАТЕЛЯ и охраняется законами Российской Федерации и международными нормативно-правовыми актами об авторском праве, а также другими законами и договорами, регулирующими правоотношения, вытекающие из права интеллектуальной собственности. MasterSCADA™ лицензируется, а не продается.
4. ОБЪЕМ ЛИЦЕНЗИИ. Настоящее Соглашение дает вам следующие права, которыми Вы обязуетесь пользоваться добросовестно:
- 4.1. Условия лицензирования. MasterSCADA™ предоставляется вам на условиях «как есть». То есть, используя MasterSCADA™, вы признаете, что MasterSCADA™ может содержать ошибки, как программного кода, так и алгоритмические. Весь риск, связанный с использованием или невозможностью использования MasterSCADA™, вы осознанно принимаете на себя.
 - 4.2. В зависимости от комплекта поставки MasterSCADA может лицензироваться следующими способами:

- 4.2.1. Использование одной лицензии с USB-ключом аппаратной защиты на одном персональном компьютере, в который установлен данный ключ.
- 4.2.2. Использование одной Серверной лицензии с USB-ключом аппаратной защиты с заранее определенным числом лицензий рабочих мест.
- 4.2.3. Использование программного сервера лицензирования ПРАВООБЛАДАТЕЛЯ посредством сети «Интернет».
- 4.2.4. Использование программного сервера лицензирования на стороне пользователя.
- 4.2.5. Использование лицензии с аппаратной защитой.
- 4.2.6. Использование программного сервера лицензирования с аппаратной защитой на стороне пользователя.
- 4.3. Использование MasterSCADA способами, не указанными в п. 4.2 настоящего Соглашения, запрещено!
- 4.4. Хранение и копирование. Вам разрешается хранить копии носителя, на котором поставляется MasterSCADA™, на жестком диске компьютера или сетевого сервера, а также на любом другом носителе исключительно в целях резервного хранения на случай выхода из строя оригинального носителя.
- 4.5. Техническая поддержка. Техническую поддержку на MasterSCADA™ выполняет авторизованная для этих целей ПРАВООБЛАДАТЕЛЕМ компания ООО «МПС Софт» (далее – «МПС Софт»). МПС Софт предоставляет техническую поддержку в соответствии с условиями описанной далее Ограниченной гарантии только тем юридическим или физическим лицам, которые приобрели MasterSCADA™ легальным способом, либо заключили с МПС Софт отдельный договор на предоставление технической поддержки.
- 4.6. Коммерческое использование версий MasterSCADA™, распространяемых ПРАВООБЛАДАТЕЛЕМ на безвозмездной основе, недопустимо.
- 4.7. ПРАВООБЛАДАТЕЛЬ сохраняет за собой все права на MasterSCADA™, в том числе и те, которые передаются вам по настоящему Соглашению.
5. ОГРАНИЧЕНИЯ. Настоящее Соглашение накладывает на вас следующие ограничения. Вы, в свою очередь, начиная использование MasterSCADA™, обязуетесь им следовать.
 - 5.1. Исключительные права. MasterSCADA™, в том числе включенные в него примеры использования, электронная и печатная документация, графические и интерфейсные решения, являются объектами исключительных прав. Копирование печатных материалов, сопровождающих MasterSCADA™, ЗАПРЕЩЕНО.
 - 5.2. Изменение состава. Использование MasterSCADA™ разрешено только в том составе модулей и функций, на которые была приобретена данная лицензия. Несанкционированные изменения этого состава запрещены.

- 5.3. Обратное конструирование, декомпиляция и дизассемблирование. Запрещено предпринимать обратное конструирование, декомпиляцию и дизассемблирование MasterSCADA™.
- 5.4. Разделение MasterSCADA™. MasterSCADA™ лицензируется как единый продукт. Составляющие его части нельзя разделять для использования на нескольких компьютерах. Исключения составляют исходные тексты, о которых в MasterSCADA™ явно указано, что они предназначены для разработки самостоятельных пользовательских программных решений.
- 5.5. Прокат. Не разрешается давать MasterSCADA™ в прокат или временное пользование, а также отчуждение прав и обязанностей конечного пользователя MasterSCADA™ третьим лицам кроме случаев, когда ПРАВООБЛАДАТЕЛЬ дал на то предварительное письменное согласие, а также случаев правопреемства, предусмотренных законодательством Российской Федерации. В случае правопреемства, правопреемник конечного пользователя обязан уведомить ПРАВООБЛАДАТЕЛЯ или уполномоченного представителя о правопреемстве до начала использования MasterSCADA™, а также присоединиться к настоящему Соглашению.
- 5.6. Снятие защиты. Запрещается предпринимать любые действия, приводящие к возможности запуска MasterSCADA™ в несоответствии с условиями приобретенной лицензии.
- 5.7. Прекращение действия Соглашения. Настоящее Соглашение автоматически прекращается, если вы нарушаете хотя бы одно условие данного Соглашения. В таком случае вы должны уничтожить все копии MasterSCADA™ и все составляющие его части, имеющиеся в вашем распоряжении.

Ограниченная гарантия на исполнительную систему пакета прикладных программ для ЭВМ MasterSCADA™

1. Приобретая MasterSCADA™ легальным способом конечный пользователь получает ограниченную гарантию, а также получает право на заключение договора о технической поддержке (Купон), также предоставляющим право на ограниченную гарантию.
2. Ограниченная гарантия. МПС Софт гарантирует, что носители и USB-ключ аппаратной защиты будут свободны от недостатков, связанных с качеством материала или производства в течение 1 (одного) года со дня получения конечным пользователем MasterSCADA™. Вместе с тем, как определено пунктом «Условия лицензирования», MasterSCADA™ предоставляется на условиях «как есть», включая все программные и алгоритмические ошибки, которые могут в нем содержаться.
3. Ответственность МПС Софт. Весь риск, связанный с использованием или невозможностью использования программы, сопроводительной документации и технической поддержки, Конечный пользователь MasterSCADA™ принимает на

себя. Материальная ответственность МПС Софт сводится к ремонту или замене MasterSCADA™ и любых его составляющих, не отвечающих требованиям, установленным настоящей ограниченной гарантией, и возвращенных МПС Софт наряду с копиями всех документов на приобретение MasterSCADA™. Настоящая ограниченная гарантия недействительна, если повреждение MasterSCADA™ или его составляющих является результатом ненадлежащего обращения, неправильного использования или несчастного случая. Любые составляющие MasterSCADA™, полученные в качестве замены в соответствии с настоящей ограниченной гарантией, гарантируются на любой из более продолжительных сроков: период до истечения первоначального гарантийного срока или 30 (тридцать) дней.

4. Нераспространение материальной ответственности на косвенные убытки. В максимальной степени, допускаемой применимым законодательством, Правообладатель и его полномочный представитель МПС Софт не несут материальную ответственность за какие-либо убытки (включающие, не ограничиваясь перечисленным, прямые или косвенные убытки в результате нанесения телесных повреждений, неполучения доходов от хозяйственной деятельности, вынужденных перерывов хозяйственной деятельности, утерь деловой информации или нанесения любых других видов имущественного ущерба), вытекающие из использования или невозможности использования данного изделия, даже в том случае, если Правообладатель и его полномочный представитель МПС Софт был предупрежден о возможности этих убытков.
5. Непредставление других гарантий. В максимальной степени, допускаемой применимым законодательством, МПС Софт отказывается от предоставления каких-либо других прямых или подразумеваемых гарантий, включающих, не ограничиваясь перечисленным, гарантии товарности и пригодности для конкретной цели, в отношении MasterSCADA™, любых его составляющих и приобретенных к нему письменных материалов.
6. Техническая поддержка. Техническая поддержка производится в период действия имеющегося у вас Купона на техническую поддержку MasterSCADA™ и на условиях, обозначенных в этом Купоне.
7. Расширенная техническая поддержка. Расширенная техническая поддержка в период действия ограниченной гарантии, включающая телефонные консультации и/или консультации на территории фирмы МПС Софт и/или сокращенное время ответа на запросы по электронной почте может быть предоставлена по отдельному договору.

3.6. ПРИНЯТЫЕ СОКРАЩЕНИЯ И ОБОЗНАЧЕНИЯ

В данной документации используются следующие сокращения:

SCADA – Supervisory Control And Data Acquisition, диспетчерское управление и сбор данных;

НМИ – человеко-машинный интерфейс;

АСУ – автоматизированная система управления;

АСУ ТП – автоматизированная система управления технологическим процессом;

PLC (ПЛК) – программируемый логический контроллер;

БД – база данных;

ЛК – левая кнопка мыши;

ПК - правая кнопка мыши;

OPC DA – унифицированная технология получения текущих данных из OPC-серверов. Базируется на DCOM (OPC – OLE for Process Control).

OPC HDA – унифицированная технология получения архивных данных из OPC-серверов.

OPC UA – унифицированная кроссплатформенная технология получения текущих и архивных данных.

4. ПРИНЦИПЫ ПРОЕКТИРОВАНИЯ

Данный раздел не претендует на полноту описания приемов и методов проектирования в **MasterSCADA 4D**, в нем изложены только основные принципы разработки проекта. Для того чтобы получить более детальное представление о процессе создания проекта, рекомендуется выполнить проекты **Быстрого старта** и посмотреть **видеопримеры**, а также ознакомиться с назначением элементов проекта **MasterSCADA 4D** и операциями в среде разработки.

Типовой порядок разработки проекта в **MasterSCADA 4D** включает следующие процедуры:

- анализ объекта автоматизации;
- разработка типовых объектов, окон, программ и других элементов в библиотеке;
- создание структуры проекта;
- разработка уникальных программ, окон (мнемосхем) и других необходимых элементов;
- конфигурирование необходимых связей;
- формирование задач узлов;
- компиляция проекта и загрузка конфигураций в узлы;
- отладка проекта.

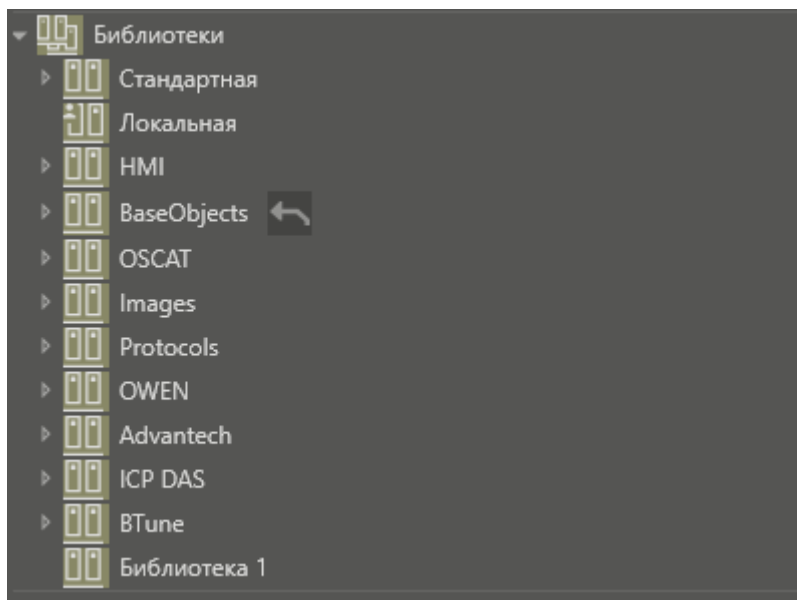
4.1. АНАЛИЗ ОБЪЕКТА АВТОМАТИЗАЦИИ

Прежде чем приступать к созданию проекта в среде разработки **MasterSCADA 4D**, следует провести анализ объекта автоматизации. Самая главная цель анализа состоит в выделении типовых элементов, встречающихся в проекте неоднократно и представляющих собой законченные части программы.

Это необходимо сделать для того, чтобы создать в библиотеке полный набор типовых элементов, которые могут быть использованы в дальнейшем при разработке. Все элементы проекта являются наследниками какого-либо библиотечного элемента. Внесение изменений в библиотечный объект повлечет изменение его экземпляров, что позволит сократить время на модернизацию и отладку проекта.

4.2. РАЗРАБОТКА ТИПОВЫХ ЭЛЕМЕНТОВ

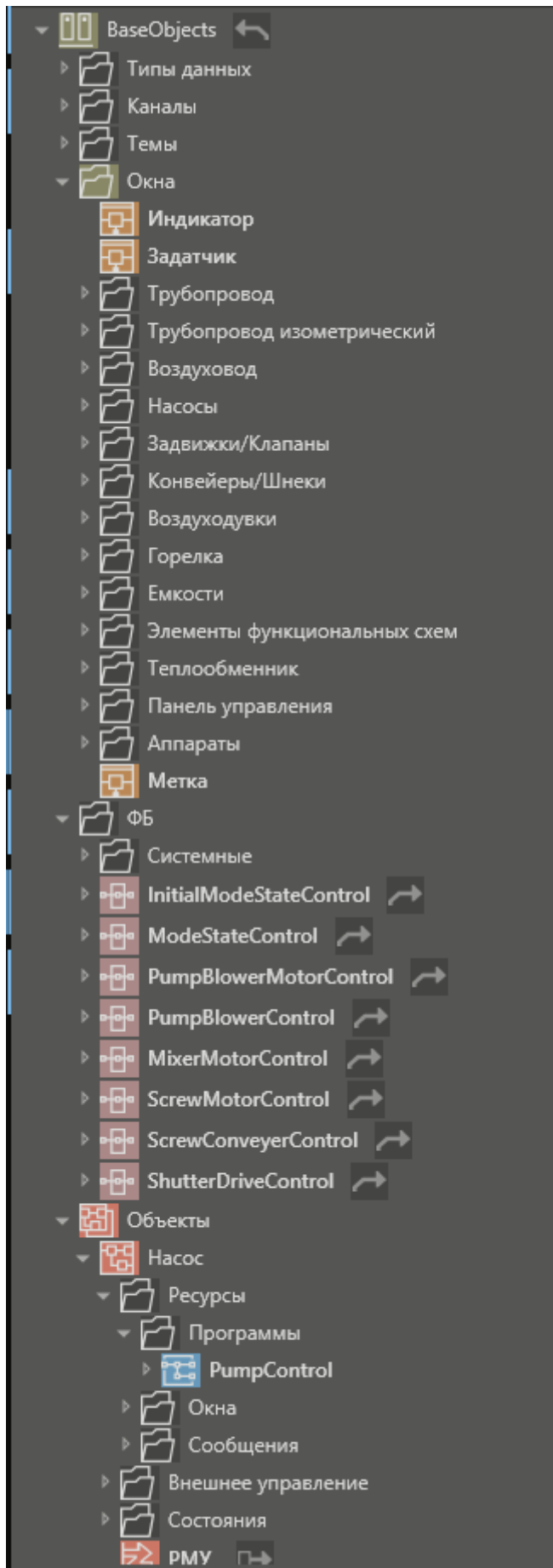
Типовые элементы проекта разрабатываются в дереве библиотек. Если элемент планируется использовать только в рамках текущего проекта, то его можно разрабатывать в библиотеке Локальная. В этом случае созданные элементы будут видны только в рамках текущего проекта. Если предполагается использовать элементы в нескольких проектах, либо когда разработку типовых элементов начинают несколько пользователей одновременно, то следует создать пользовательскую библиотеку при помощи контекстного меню либо контекстной панели.



Если необходимо создать библиотечный объект, который включает в себя окна и программы, то сначала создается библиотечная программа, библиотечное окно, а затем уже библиотечный объект.

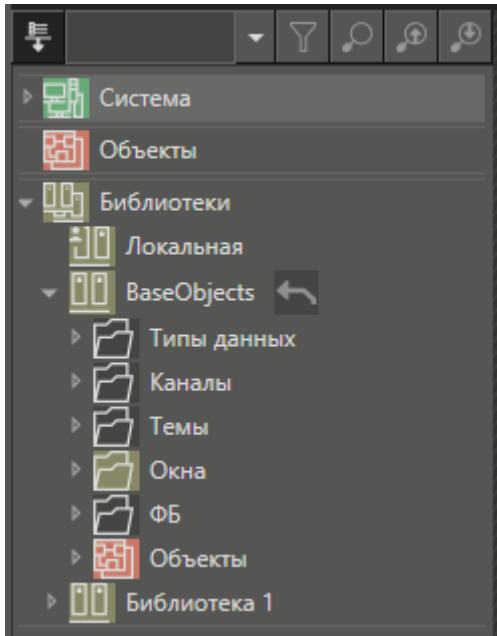
По такому принципу сделана одна из нескольких библиотек, включенных в поставку **MasterSCADA 4D** по умолчанию, – **BaseObjects**.

Так, например, если проанализировать объект Насос, то можно заметить, что сначала были созданы окна и ФБ, а затем из этих элементов собрали библиотечный объект:



4.3. СОЗДАНИЕ СТРУКТУРЫ ПРОЕКТА

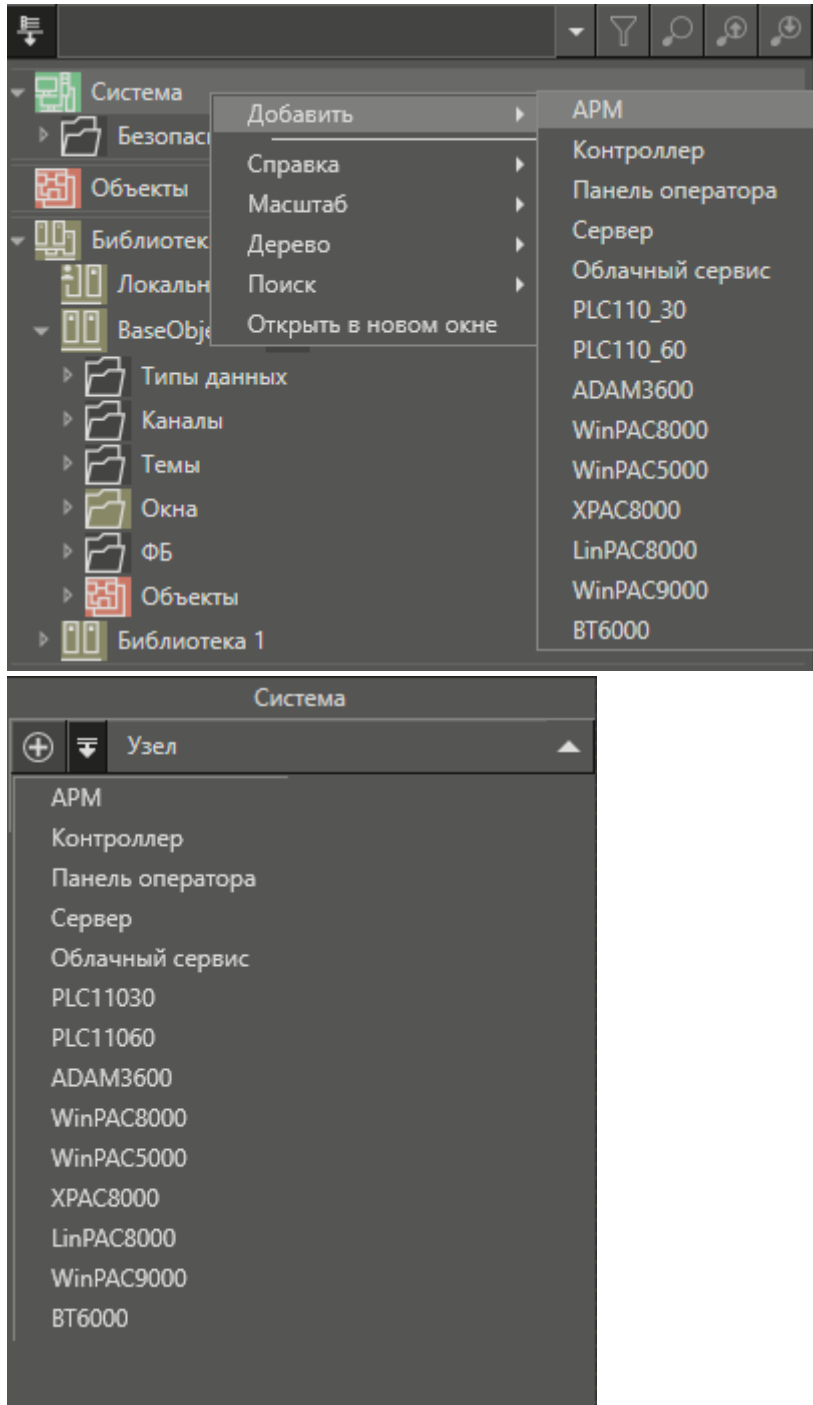
В **MasterSCADA 4D** проект создается из библиотечных элементов в **дереве системы** и **дереве объектов**.



Дерево системы

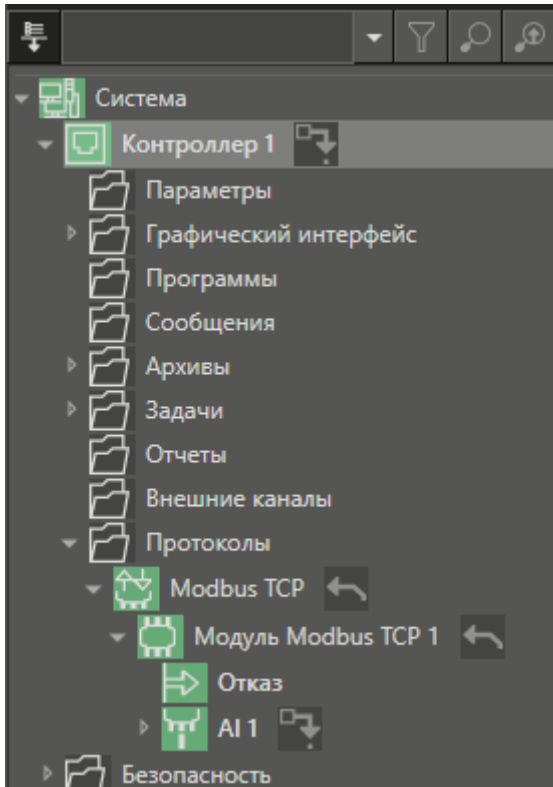
В этой группе создаются **узлы** – конфигурации, загружаемые для исполнения в реальные устройства проекта (АРМы, контроллеры и т. д.).

Для создания узла можно использовать контекстное меню (см. левый рис. ниже), контекстную панель или метод перетаскивания (drag-n-drop) из соответствующей библиотеки, в которой определена начальная структура поддерживаемых устройств (см. правый рис. ниже):



В узле конфигурируются его внешние связи с устройствами и/или OPC-серверами (указываются устройства, протоколы, каналы связи).

Для конфигурирования взаимодействия с оборудованием используется интуитивно понятная логика аппаратно-программного информационного потока: **узел** – протокол – устройство (модуль) – каналы (структуры или параметры):



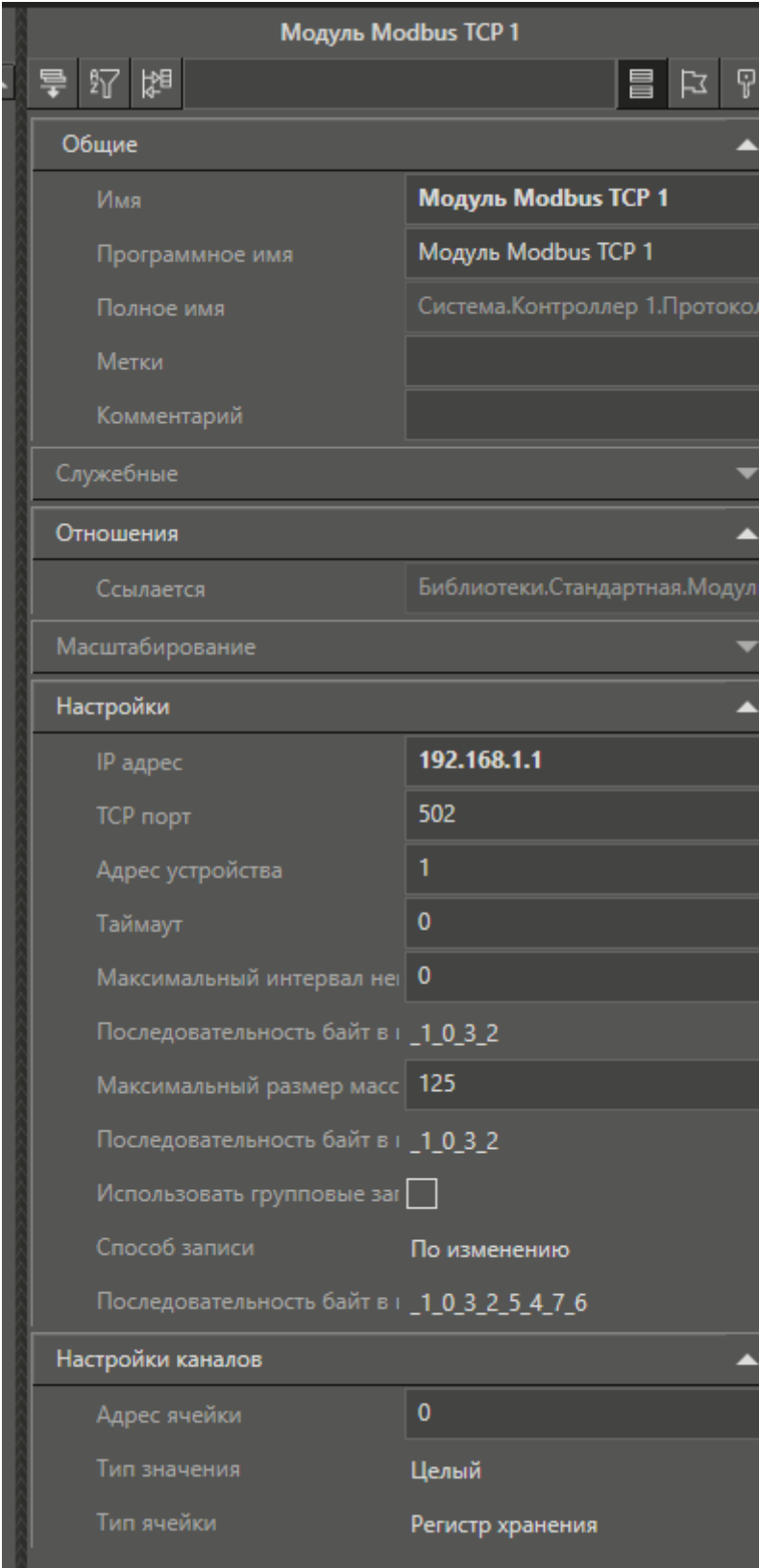
Для всех реальных аппаратных и программных элементов системы управления (АРМов, контроллеров, портов, протоколов и т.п.) в структуре проекта создаются соответствующие элементы проекта. Набор всех необходимых элементов содержится в библиотеках, откуда элементы могут быть вставлены в соответствующую группу дерева методом перетаскивания (см. **Библиотеки**). Для создания элементов могут быть также использованы команды контекстных меню групп дерева.

То есть в группе **Протоколы** вначале создается группа протокола (Modbus TCP в примере), при этом в окне свойств задаются параметры самого протокола:

The screenshot shows the configuration window for a Modbus TCP module. The window is titled "Modbus TCP" and has a toolbar at the top with icons for save, refresh, and other actions. The configuration is organized into several sections:

- Общие (General):**
 - Имя (Name): Modbus TCP
 - Программное имя (Program name): Modbus TCP
 - Полное имя (Full name): Система.Контроллер 1.Протокол
 - Метки (Tags):
 - Комментарий (Comment):
- Служебные (Service):**
- Отношения (Relations):**
 - Ссылается (Refers to): Библиотеки.Стандартная.Протокол
- Задача (Task):**
 - Период, мс (Period, ms): 100
 - Приоритет (Priority): 100
 - Подключение по условию (Connect on condition):
 - Выполнение по условию (Execute on condition):
 - Выполнять на резервном (Execute on backup):
- Протокол (Protocol):**
 - Таймаут (Timeout): 1000
 - Количество повторов при не (Number of retries when not): 2
 - Modbus поверх TCP (Modbus over TCP):

Затем в настройках модуля задаются параметры TCP/IP соединения:



Модуль Modbus TCP 1

Общие

Имя	Модуль Modbus TCP 1
Программное имя	Модуль Modbus TCP 1
Полное имя	Система.Контроллер 1.Протокол
Метки	
Комментарий	

Служебные

Отношения

Ссылается	Библиотеки.Стандартная.Модуль
-----------	-------------------------------

Масштабирование

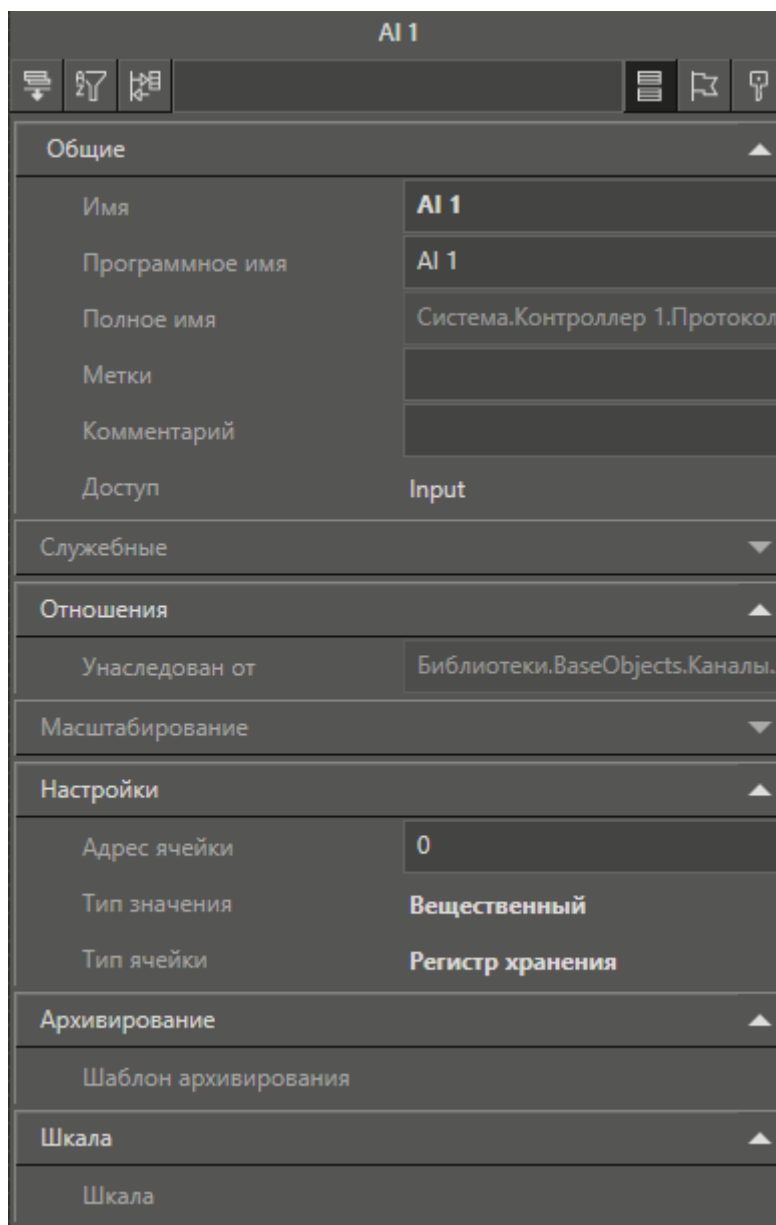
Настройки

IP адрес	192.168.1.1
TCP порт	502
Адрес устройства	1
Таймаут	0
Максимальный интервал не	0
Последовательность байт в	_1_0_3_2
Максимальный размер масс	125
Последовательность байт в	_1_0_3_2
Использовать групповые за	<input type="checkbox"/>
Способ записи	По изменению
Последовательность байт в	_1_0_3_2_5_4_7_6

Настройки каналов

Адрес ячейки	0
Тип значения	Целый
Тип ячейки	Регистр хранения

После чего следует задать адреса каналов и типы ячеек, из которых необходимо читать данные:

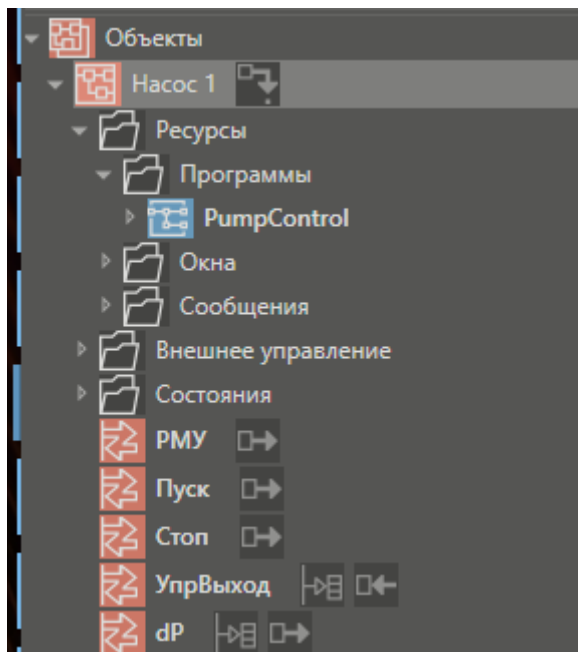


Канал – это структура, содержащая один или несколько параметров для работы с данными и ряд конфигурационных параметров. Например, канал обмена по протоколу Modbus содержит значение, прочитанное драйвером или поставленное в очередь на запись. Специфическими конфигурационными параметрами такого канала являются **Тип ячейки** и **Адрес ячейки** (соответственно адресное пространство Modbus и смещение адреса в этом пространстве), а также **Тип значения** (логический, целый и т.п.). Кроме того, для канала задаются такие общие параметры, как **Доступ** (Input – канал чтения, Output – канал записи), **Имя**, **Комментарий** и др.

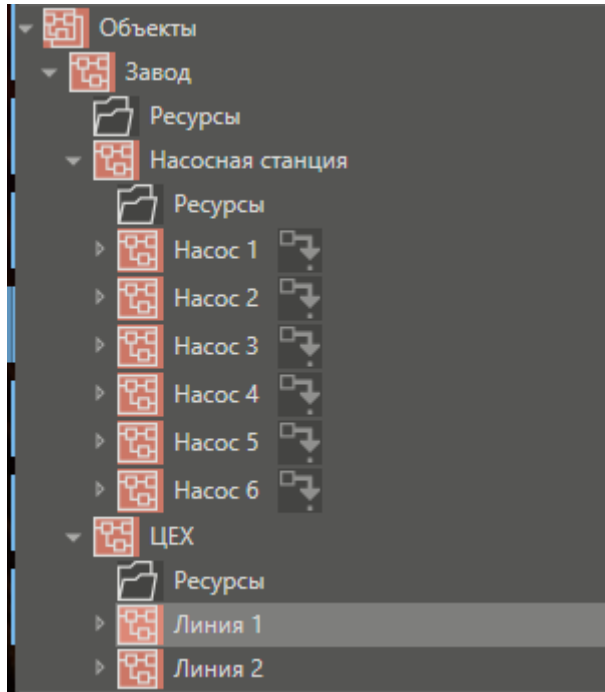
Дерево объектов

В этой группе задается вся логика проекта. При этом в **MasterSCADA 4D** основной идеологией при разработке проекта является объектно-ориентированный подход.

Объект **MasterSCADA 4D** представляет собой именованную совокупность графического представления технологического объекта, его параметров, алгоритмов контроля и управления, окон управления и других доступных элементов проекта (в том числе объект может содержать другие объекты). То есть объект **MasterSCADA 4D** может соответствовать любому реальному технологическому объекту (датчику, исполнительному механизму, аппарату, участку, цеху) и является основной структурной единицей проекта. По сути, объект **MasterSCADA 4D** является мини-проектом: он содержит все параметры технологического объекта, программы и окна контроля и управления и т.п. Для связи с другими элементами проекта в объекте создаются элементы **Параметр**.



Поскольку объект может содержать другие объекты, в дереве объектов может быть создана вся иерархия технологического объекта управления – от датчика до завода в целом:

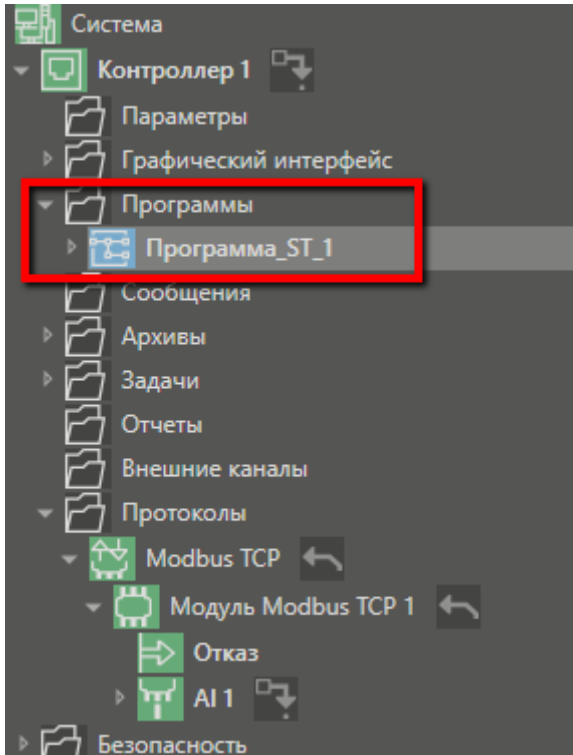


Важно! Типовые объекты разрабатываются заранее и помещаются в библиотеку, а впоследствии из них, как из кирпичиков, формируется дерево проекта.

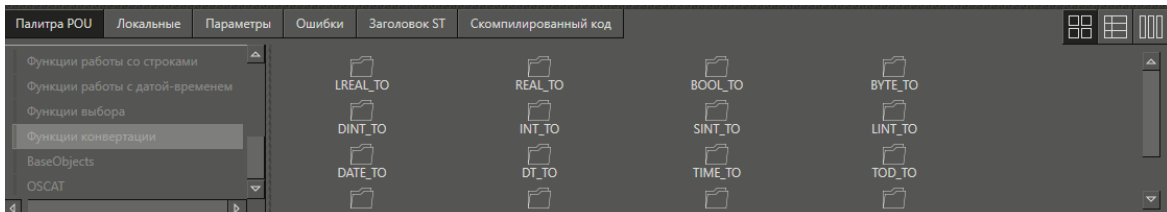
Если узел в проекте один, то никаких действий для добавления того или иного объекта в узел не требуется, все объекты будут назначены в него автоматически (т.е. в режиме исполнения написанная программа будет исполняться в системе единственного узла). Если узлов несколько, то для того чтобы задать исполнение объекта в том или ином узле, необходимо воспользоваться командой **Назначить в узел** (см. также **Удалить из узла**, **свойство Место исполнения** и **свойство Отключить исполнение**). Именно на этом узле и будет физически исполняться созданный объект.

Программы

Как правило, программы по смыслу относятся к какому-либо объекту, однако бывают случаи, когда тот или иной элемент проекта нельзя отнести к какому-то конкретному объекту. Например, если программа относится ко всему узлу сразу и работает со всеми объектами, исполняемыми в узле. В этом случае программы создаются в подгруппе **Программы** группы узла (группы **Ресурсы** в случае объекта или канала). Пользовательские функции и типы функциональных блоков создаются соответственно в подгруппах **Функции** и **ФБ** пользовательской библиотеки (по умолчанию библиотеки Локальная):

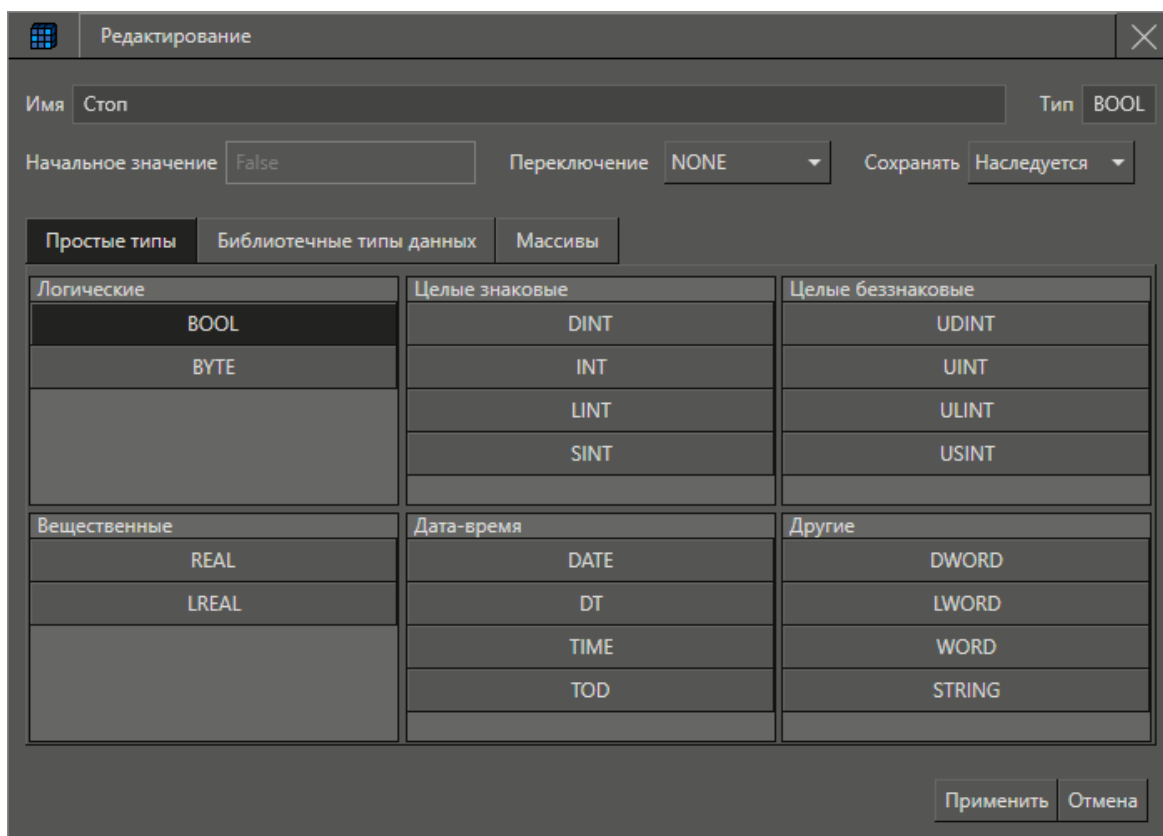


Для упрощения разработки программ в **MasterSCADA 4D** встроена **библиотека Стандартная**, а также поставляются различные подключаемые библиотеки (например, **библиотека OSCAT**), которые содержат большое количество функций и типов функциональных блоков. Предустановленные ФБ и функции доступны в легенде по закладкам редакторов различных языков программирования:



Параметры (переменные проекта)

Типы параметров определяются в соответствующем редакторе (редактор открывается двойным нажатием левой кнопкой мыши на параметр в дереве):

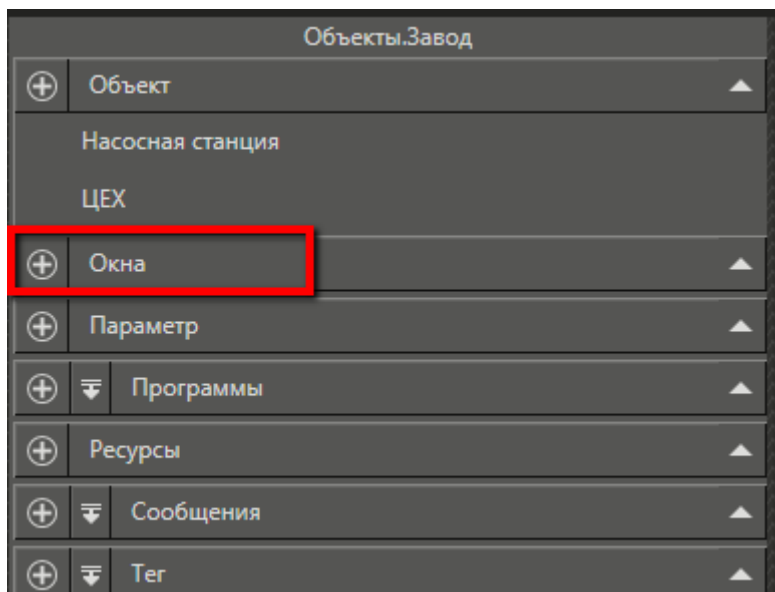


Типы параметров можно также задать в панели свойств:

Стоп	
Общие	
Имя	Стоп
Программное имя	Стоп
Полное имя	Объекты.Завод.Насосная станци
Метки	
Комментарий	
Доступ	Чтение/Запись
Сохранять	Наследуется
Начальное значение	<input type="checkbox"/>
Тип значения	BOOL
Служебные	
Отношения	
Передача значения в	Объекты.Завод.Насосная станци
Архивирование	
Шаблон архивирования	
Архивировать	<input type="checkbox"/>
Шкала	
Шкала	
Разрешения	
Всегда отображать в дереве	<input type="checkbox"/>

Окна

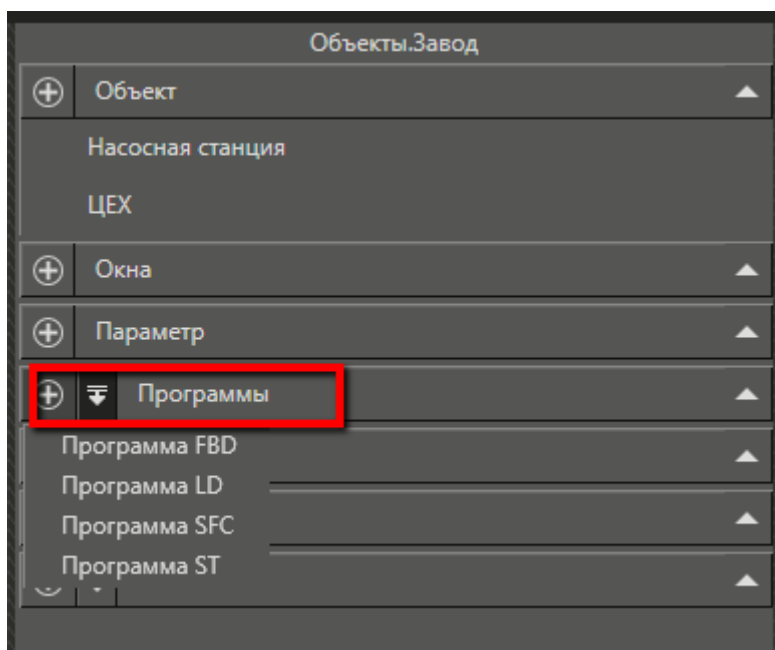
Окна создаются при помощи контекстного меню Объекта или Узла. При этом в центральной части интерфейса **MasterSCADA 4D** откроется редактор **HMI**.



4.4. РАЗРАБОТКА И ОТЛАДКА ПРОГРАММ

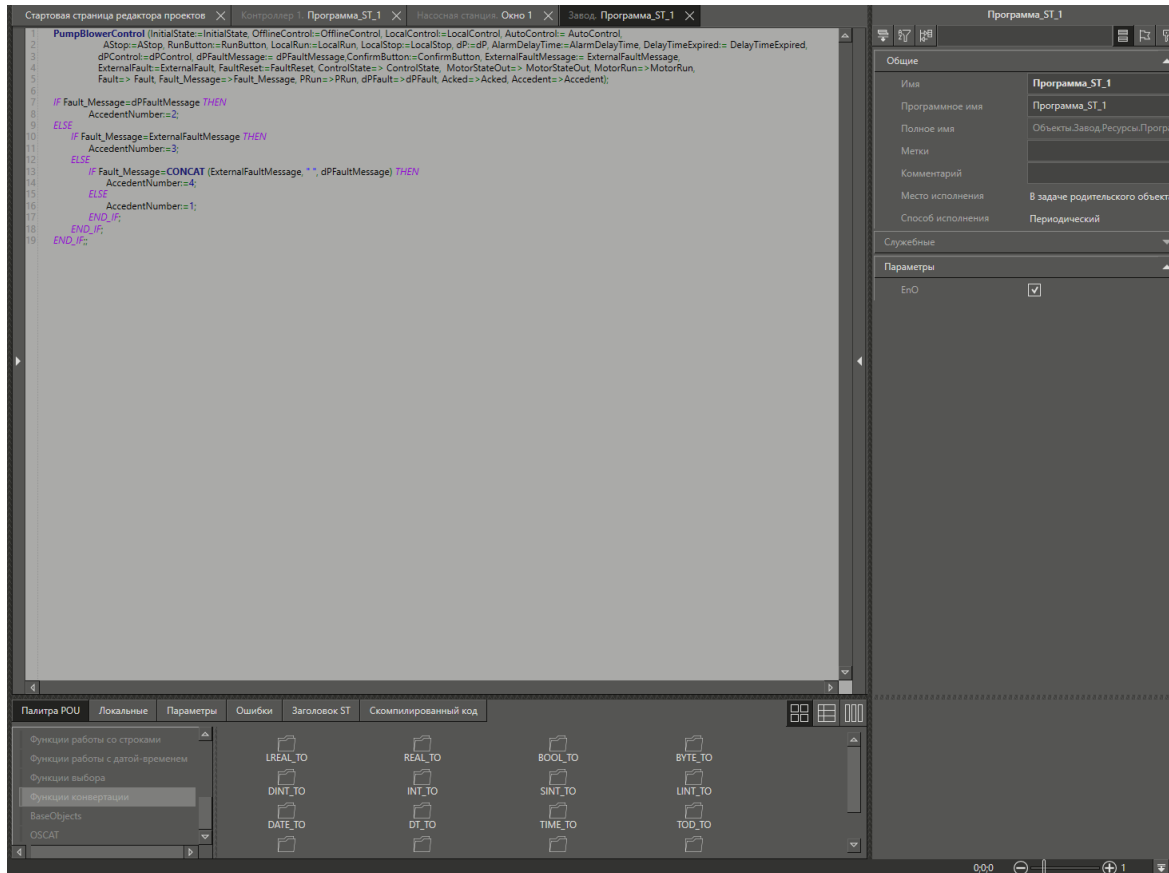
Для разработки программного элемента (программы, функции или функционального блока) в **MasterSCADA 4D** встроены редакторы (для каждого языка программирования имеется свой редактор). Создание программы производится аналогично созданию других элементов: через контекстное меню, контекстное окно или перетаскиванием из библиотеки.

Важно! Типовые программы разрабатываются в библиотеке. Для отладки в режиме исполнения программу необходимо добавить в дерево системы или дерево проекта.



Чтобы открыть программу в соответствующем редакторе, необходимо дважды нажать левой кнопкой мыши на неё в дереве. Редакторы открываются в виде вкладок в


специальном окне (см. **Панель документов и редакторов**). При этом легенда отображает различную информацию программного элемента, а специфические инструменты редактирования (при их наличии в редакторе) добавляются в область меню **MasterSCADA 4D**. На рисунке ниже показан вид **MasterSCADA 4D** при открытии редактора для языка ST:

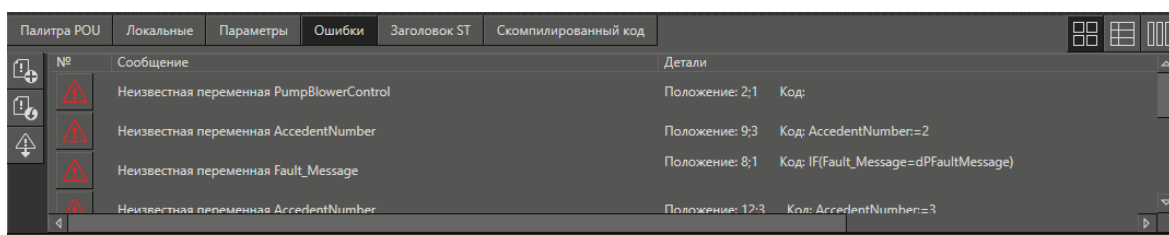


Для проверки программы на наличие ошибок следует нажать кнопку Проверить программу на панели инструментов Проверить вкладки Проект.



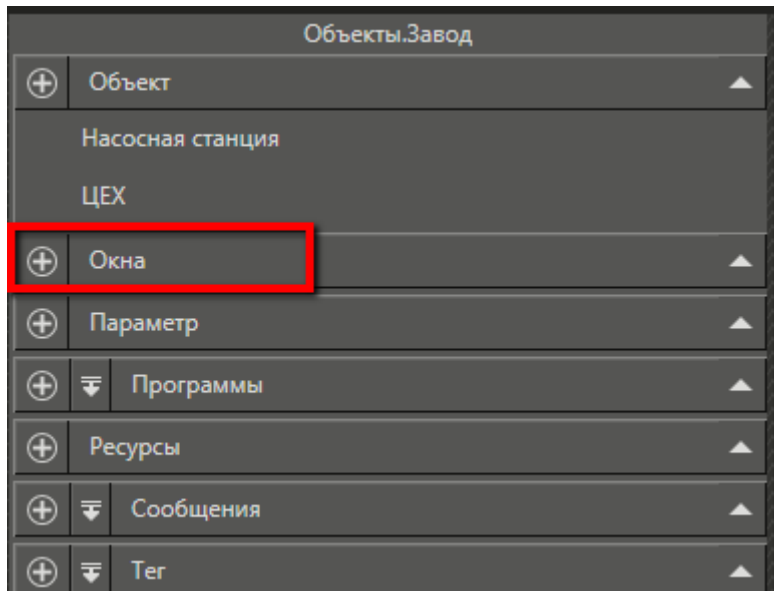
Если ошибок не обнаружено, то кнопка останется синей, а на вкладках Заголовок ST и Скомпилированный код легенды отобразятся соответственно заголовок и полный текст программного элемента на языке Master ST.

При наличии ошибок кнопка изменяет свой цвет на красный – , а информация об ошибках выводится на вкладку Ошибки в нижней части экрана:

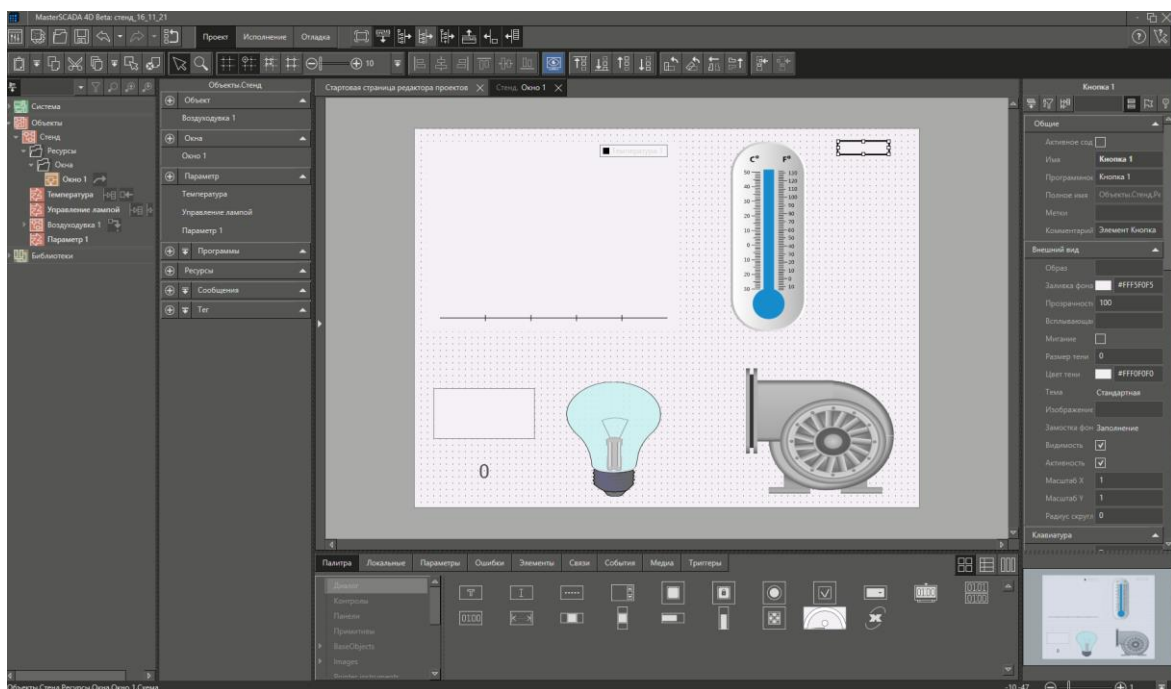


4.5. РАЗРАБОТКА ОКОН

Для разработки окон (мнемосхем) в **MasterSCADA 4D** встроен редактор **HMI**. Создание окна производится аналогично созданию других элементов: через контекстное меню, контекстное окно или перетаскиванием из библиотеки.



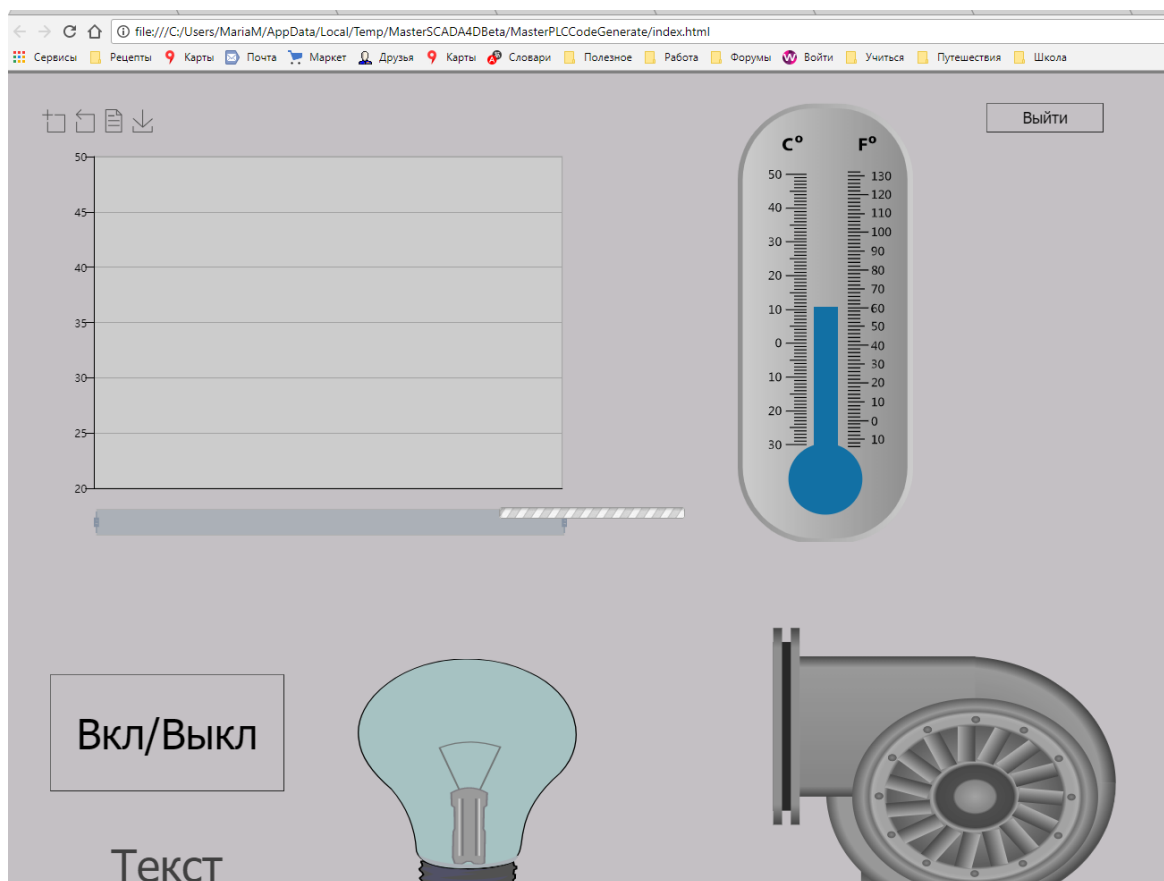
Для того чтобы открыть окно (мнемосхему) в редакторе, необходимо в дереве объектов дважды нажать ЛК на соответствующий элемент Окно. Редактор HMI открывается в виде вкладки в специальной панели (см. **Панель документов и редакторов**), при этом легенда отображает различную информацию мнемосхемы, а специфические инструменты редактирования добавляются в область меню **MasterSCADA 4D**. На рисунке ниже показан вид **MasterSCADA 4D** с открытым редактором HMI:



Для разработки мнемосхем используются графические элементы, которые содержатся в подключаемой **библиотеке НМІ**, а также в других библиотеках. К элементам можно обращаться как из дерева библиотеки, так и из Палитры графического редактора. Для того чтобы разместить элемент в окне, необходимо перетащить его из Палитры в рабочую область редактора.

Для проверки и предварительного просмотра мнемосхемы в браузере (в статическом

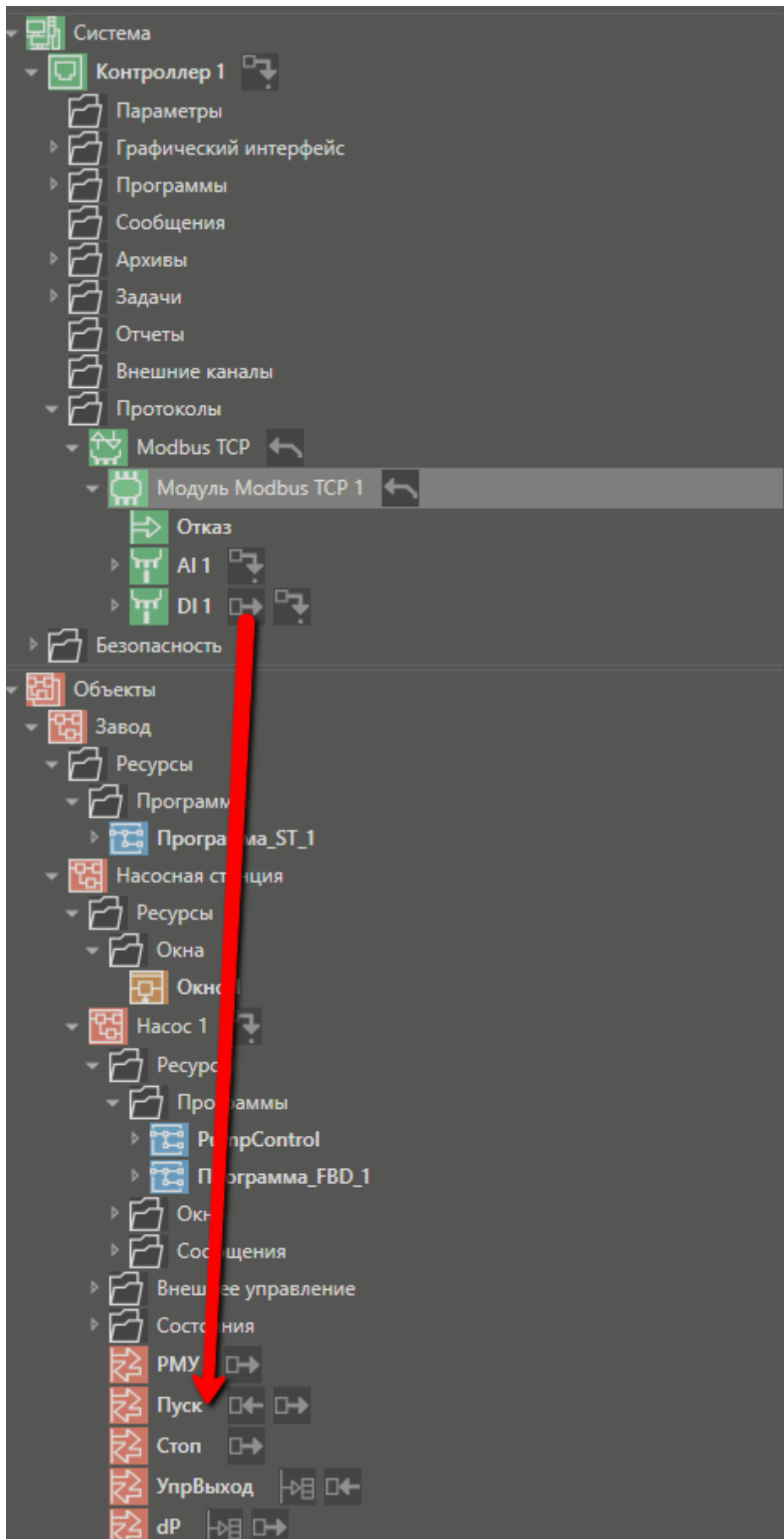
режиме) следует нажать кнопку  панели Просмотр графического редактора:




4.6. КОНФИГУРИРОВАНИЕ СВЯЗЕЙ

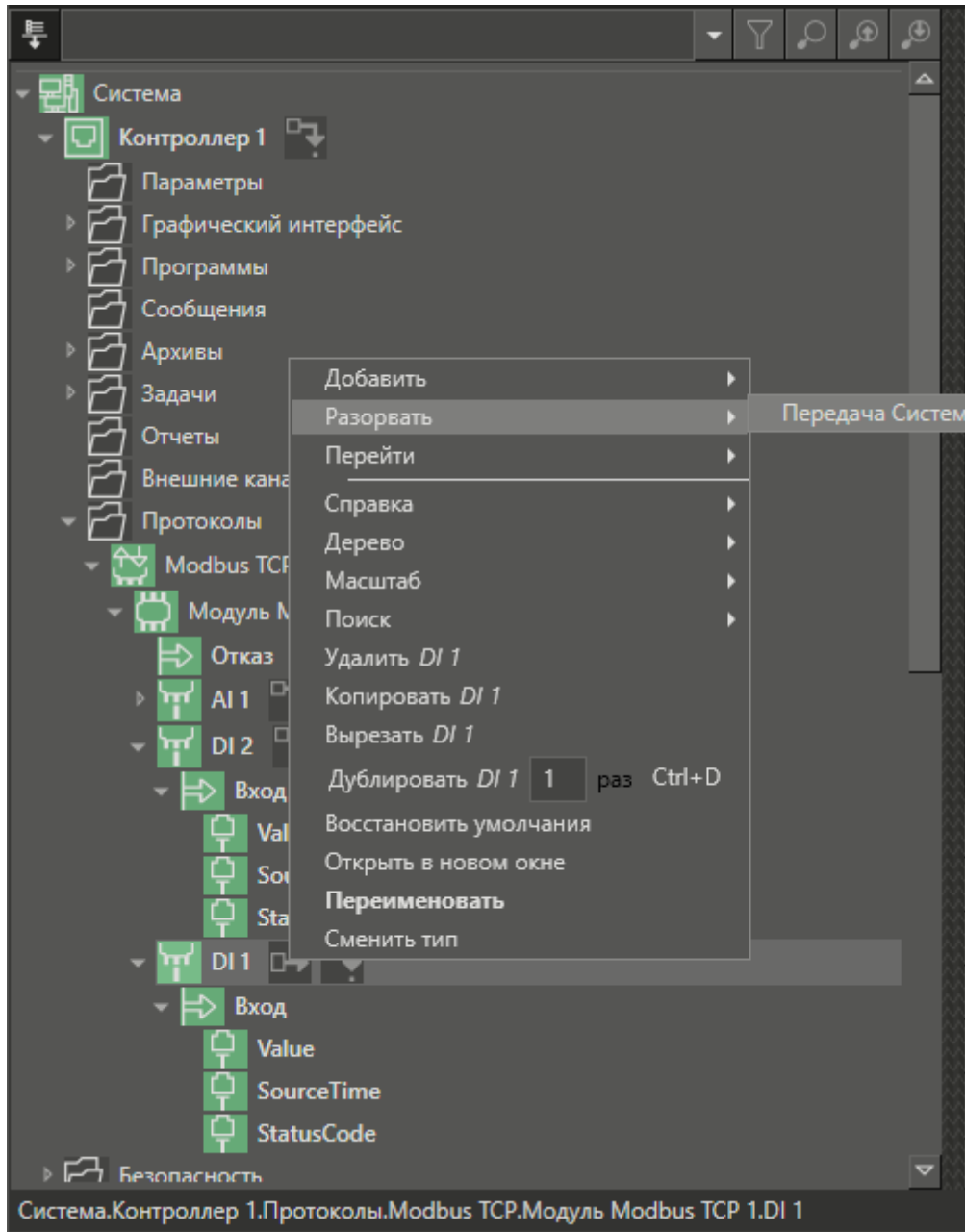
Связи в пределах одного узла (например, Параметр канала – Параметр программы) создаются с помощью перетаскивания элемента на элемент.

Важно! Если связываемые параметры могут как получать, так и отправлять значения, то перетаскивать следует источник данных на приемник данных.



Если установка связи невозможна (например, при перетаскивании выходного сигнала на выходной), то отображается специальный индикатор: 

Для удаления связи используется команда Разорвать контекстного меню элемента.

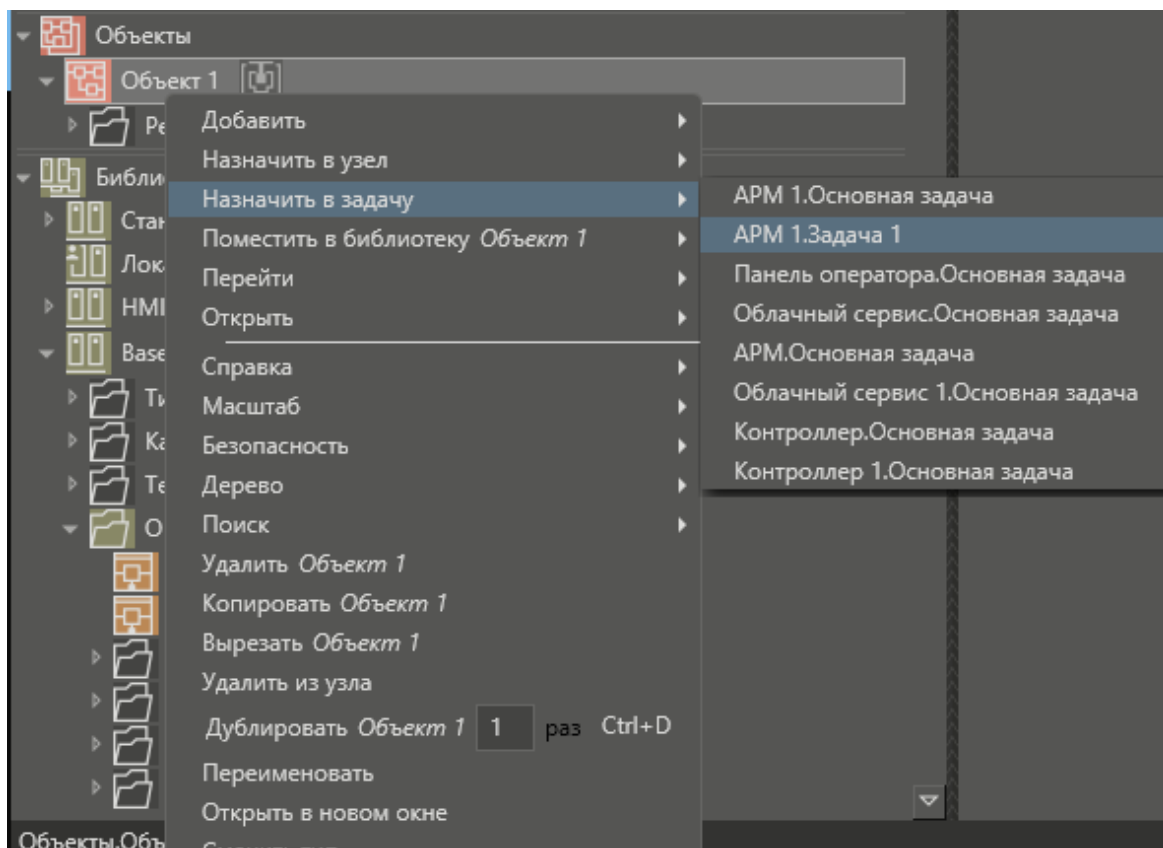


4.7. ФОРМИРОВАНИЕ ЗАДАЧ УЗЛОВ

По умолчанию, все объекты исполняются в **Основной задаче** узла. Период обработки программы зависит от настроек задачи.

В случае если необходимо, чтобы часть объектов или программ исполнялись независимо от основной задачи, то создаются дополнительные задачи в узле в соответствующей группе.

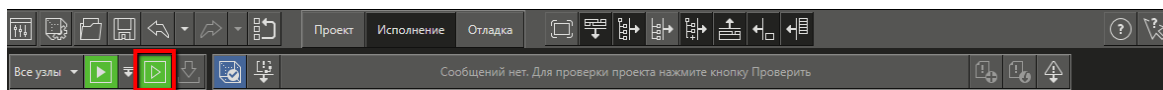
Затем тот или иной элемент проекта при помощи контекстного меню назначается в ту или иную задачу.



4.8. АВТОНОМНАЯ ОТЛАДКА В РЕАЛЬНОМ ВРЕМЕНИ

В состав **MasterSCADA 4D** входит эмулятор, с помощью которого можно проверить работу конфигурации узла в реальном времени автономно (без загрузки непосредственно в устройство).

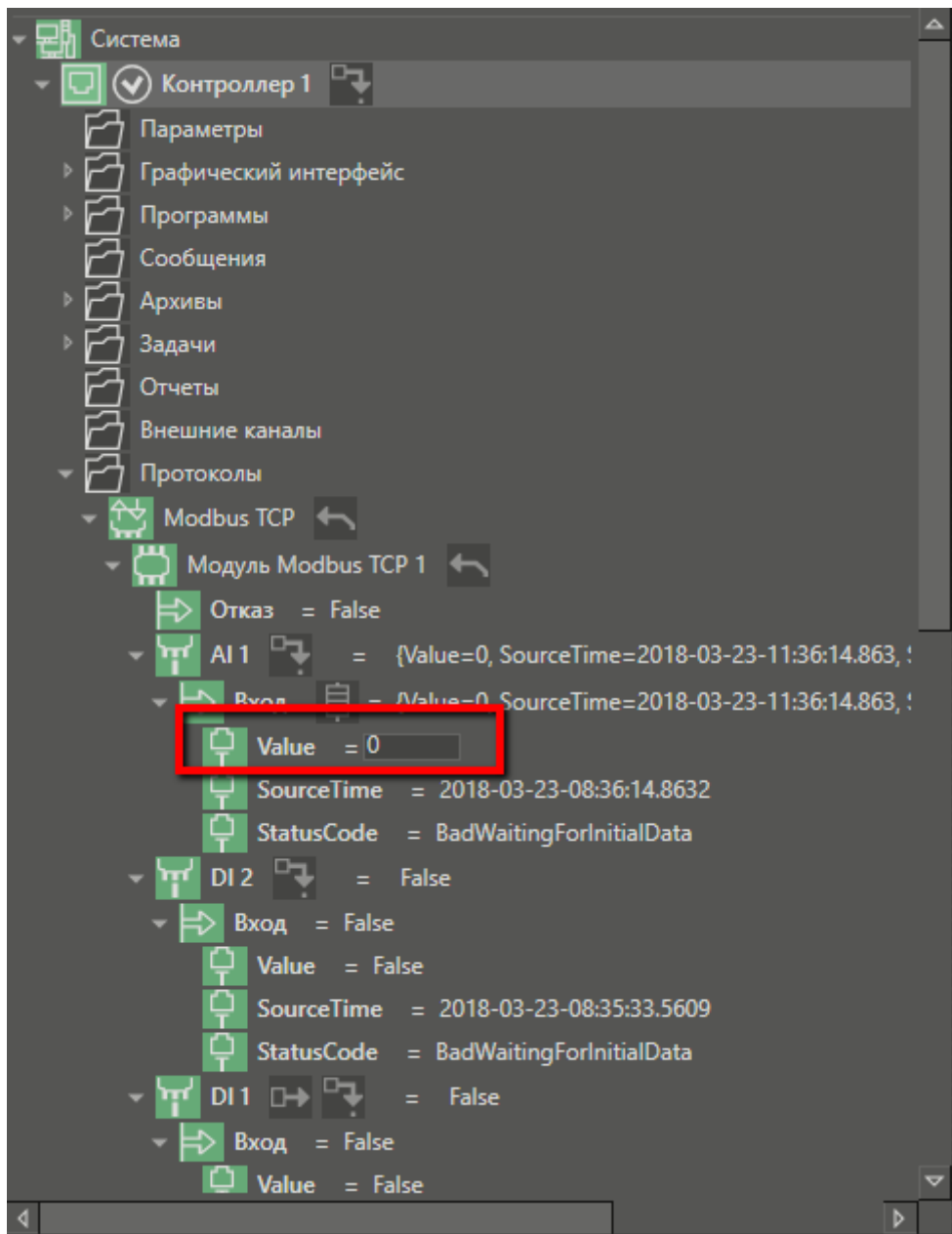
Для запуска эмуляции узла необходимо либо в контекстном меню выполнить пункт Узел.Подключить узел в режиме эмуляции, либо нажать на кнопку Эмуляция во **вкладке инструментов Исполнение**.



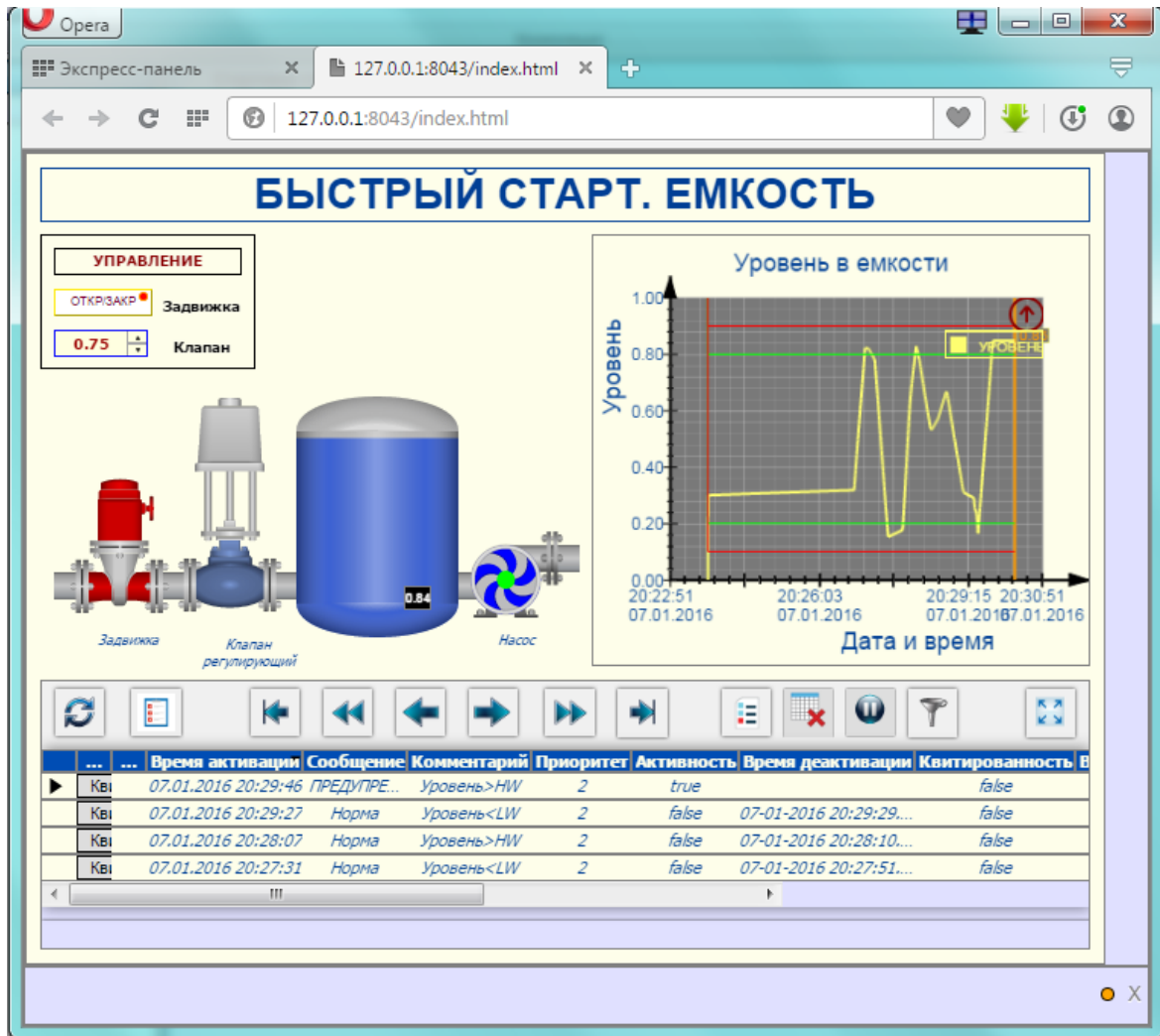
Важно! В режиме эмуляции опрос подключенных к узлу по различным протоколам устройств не происходит.


Если в проект включены несколько узлов, то запускаются несколько исполнительных систем одновременно.

В режиме эмуляции значения параметров узла отображаются и могут быть вручную изменены в различных местах, например, в дереве:



Если конфигурация узла содержит окно, то помимо эмулятора запускается веб-сервер, и окно открывается в клиенте визуализации:

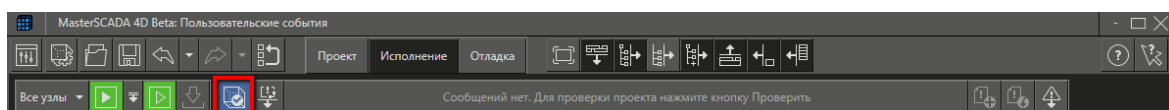


Для выхода из режима эмуляции необходимо нажать кнопку  Отключить панели инструментов вкладки Исполнение.

Если окна содержатся в нескольких узлах, то откроется несколько клиентов визуализации одновременно.

4.9. КОМПИЛЯЦИЯ ПРОЕКТА И ЗАГРУЗКА КОНФИГУРАЦИЙ В УЗЛЫ

Разработанный проект должен быть скомпилирован. Для этого необходимо нажать кнопку Проверить панели инструментов вкладки Исполнение.

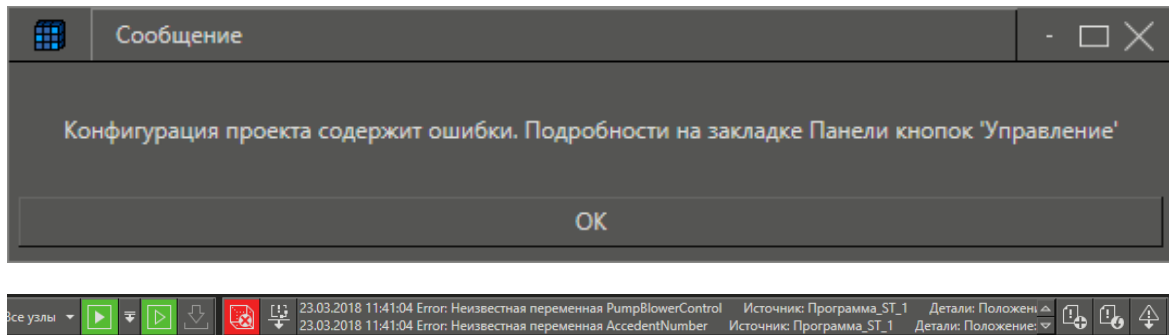


Для проверки конфигурации отдельного узла следует выполнить соответствующую команду контекстного меню в дереве системы узла.

При компиляции проекта производится также проверка и компиляция программ.

В случае успешной компиляции проекта кнопка останется зеленой, и на экране отобразится соответствующее сообщение.

При наличии в проекте ошибок кнопка изменит свой цвет на красный, а на экране отобразится соответствующее предупреждение. Информация об ошибках выведется в **окно панели Компиляция** вкладки Исполнение:



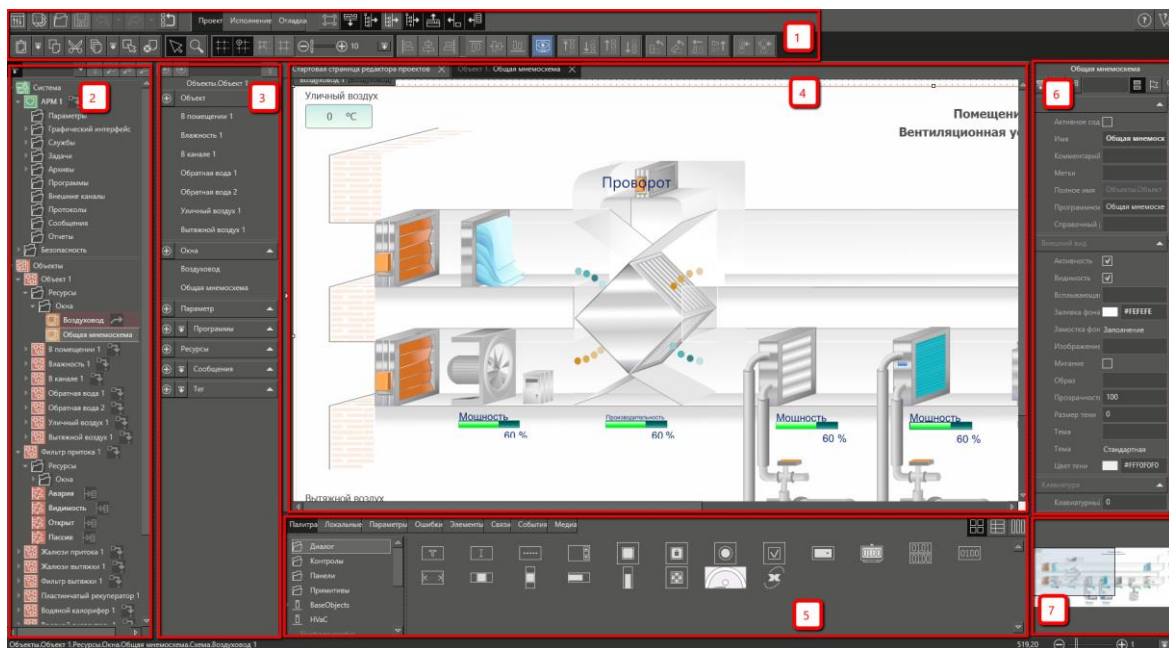
Для загрузки конфигурации в устройство следует выполнить соответствующую команду контекстного меню в дереве системы узла.

5. ИНТЕРФЕЙС РЕДАКТОРА ПРОЕКТА

В заголовке приложения указывается название продукта – MasterSCADA 4D, номер поколения продукта, например, 1.2, а также имя открытого проекта.

В редакторе проекта может быть открыт только один проект. На одном компьютере можно открыть несколько приложений MasterSCADA 4D, в которых могут независимо редактироваться разные проекты, при этом поддерживается копирование и вставка из одного проекта в другой.

Интерфейс редактора проекта состоит из набора панелей:



1. Область меню и панели быстрого доступа.

Панель быстрого доступа позволяет получить быстрый доступ к управлению средой исполнения и проектом, выполнить операции Отменить/Повторить, а также просмотреть историю действий разработчика проекта.

Далее в этой же строке идут пункты Основного меню программы. Каждому пункту меню соответствует своя панель инструментов, располагающаяся под верхней строкой области меню, или панель быстрого запуска. В верхней строке размещены также кнопки управления видимостью интерфейса.

2. Дерево проекта MasterSCADA состоит из трех частей: дерево системы, дерево объектов, дерево библиотек.

3. Окно контекстного меню. Контекстное меню предназначено для быстрого добавления элементов в текущий проект. Все пункты в данном окне аналогичны пунктам контекстного меню объекта, вызываемого правой кнопкой мыши.

4. Рабочая область. В рабочей области открываются вкладки для редактирования элементов (окон, программ, протоколов и др.). Вкладка элемента открывается двойным кликом мыши по нему в дереве проекта. Порядок работы с вкладками описан в соответствующем разделе.

5. Легенда. Набор закладок легенды меняется в зависимости от открытого редактора. Как правило, легенда содержит элементы, которые используются в том или ином редакторе на закладке Палитра, и ряд вспомогательных закладок для отладки, анализа и работы. Более подробно некоторые закладки рассматриваются в разделах, посвященных редакторам среды разработки.

6. Панель свойств отображает настройки текущего элемента, выбранного для работы в каком-либо из деревьев.

7. Миникарта – средство для навигации в окнах редакторов: FBD-, SFC-, LD-диаграмм и мнемосхем.

В нижней части окна редактора содержатся следующие инструменты:

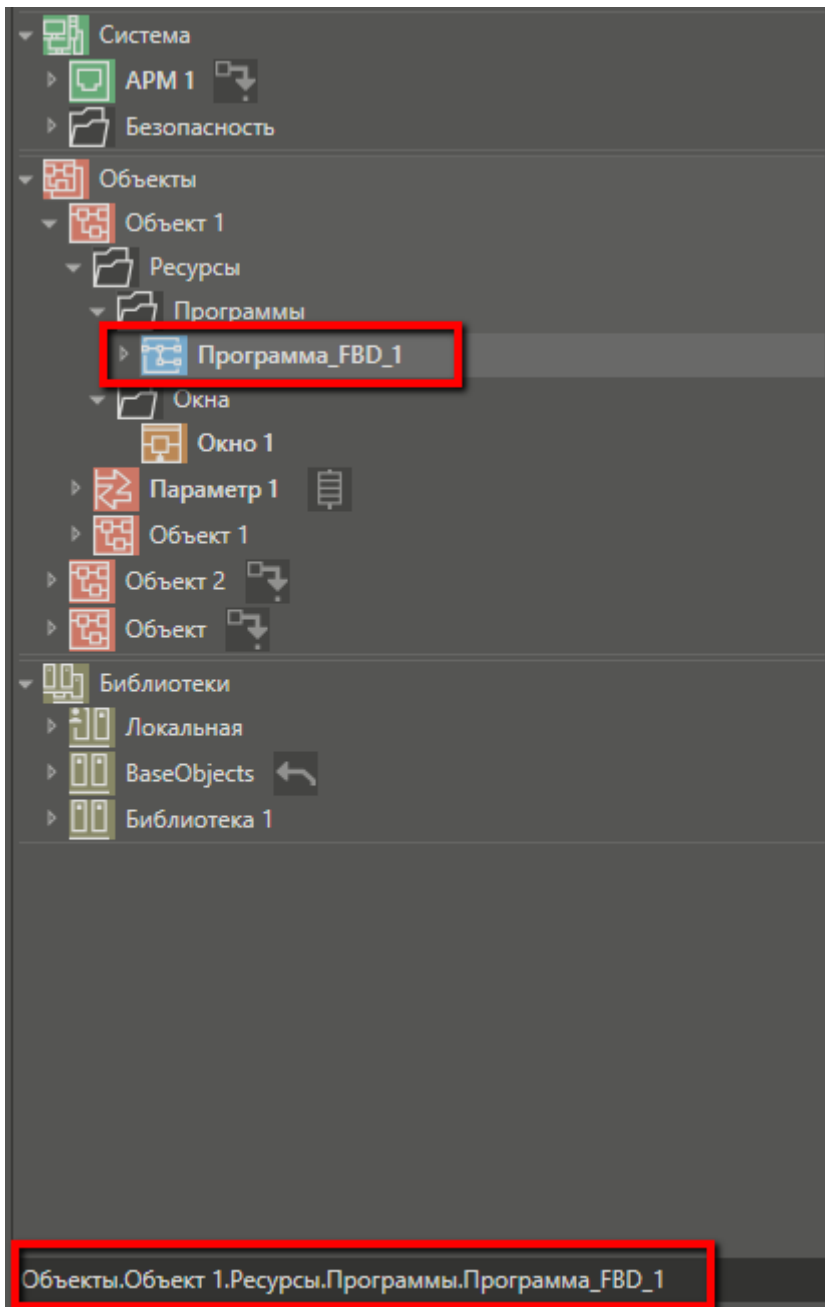
- строка статуса;
- комбинированный инструмент масштабирования



- поле позиции курсора.

Строка статуса

В строке статуса отображается полное имя элемента проекта, на который наведен указатель мыши:



Поле позиции курсора

В этом поле отображается позиция курсора в тексте ST-программы, положение курсора в графическом языке или в графическом редакторе. Поле расположено справа от комбинированного инструмента масштабирования. Позиция курсора отображается как <номер строки/координата X>;<знакоместо в строке/координата Y>:



Выполнение длительных операций

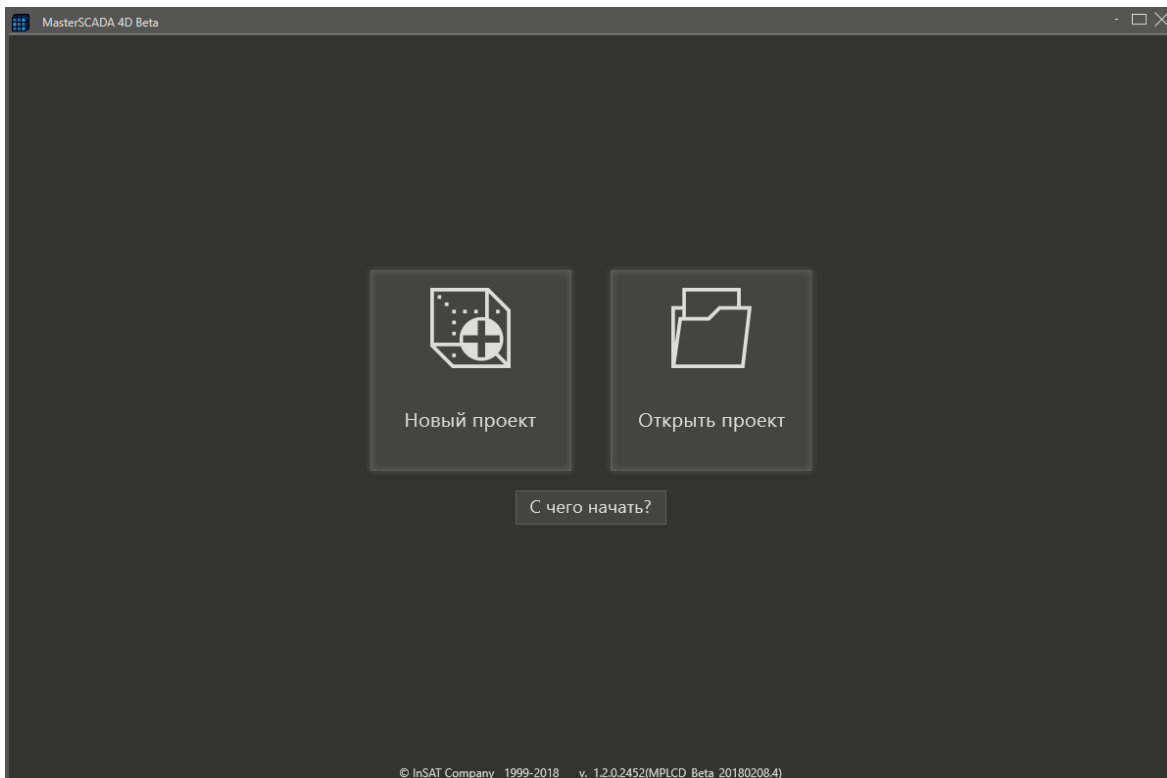
При выполнении отдельных операций, которые могут занимать длительное время, в строке статуса отображается индикатор выполнения (ProgressBar):

: проверка конфигурации проекта; добавление задач фб

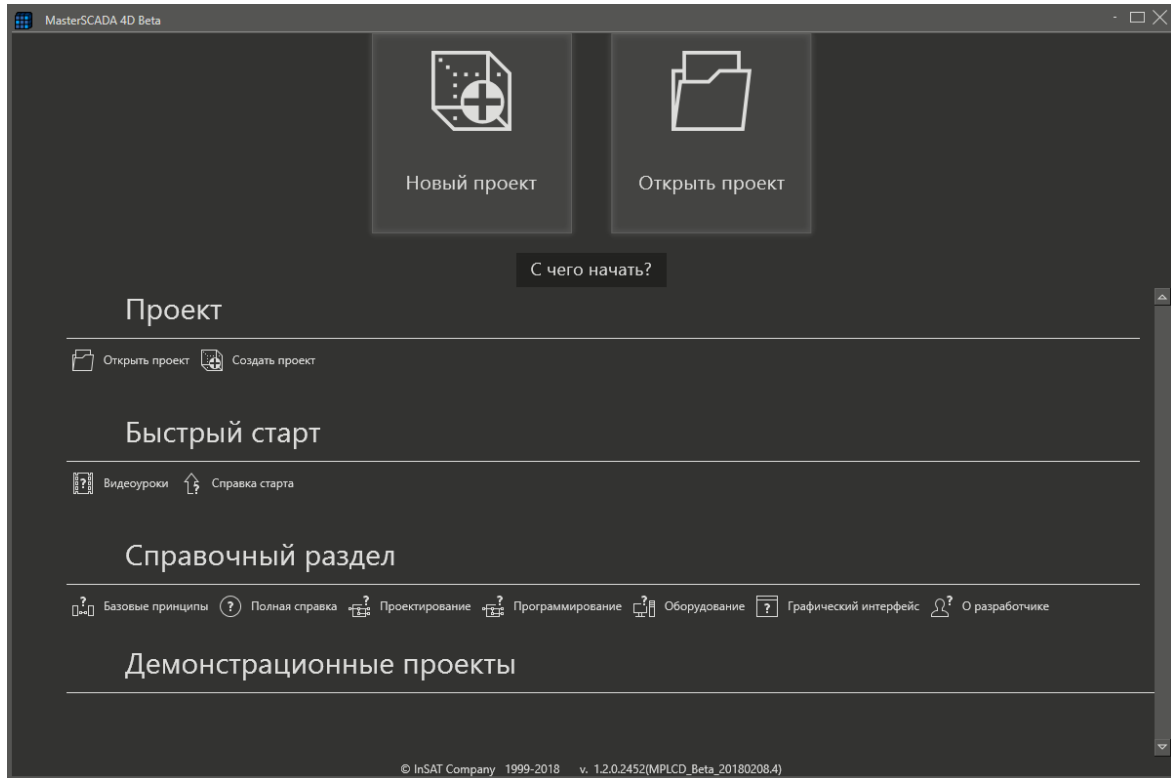
К подобным операциям относятся: подключение библиотек, обновление библиотек, копирование в буфер, вставка из буфера, удаление, проверка конфигурации проекта и др.

5.1. СТАРТОВОЕ МЕНЮ

При запуске редактора проекта открывается Стартовое меню (см. Запуск среды разработки):

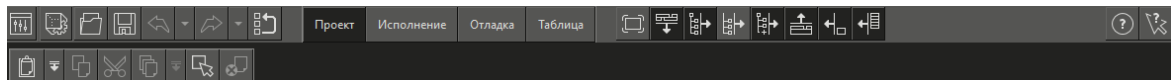


Название кнопки	Описание
Новый проект	Открывает диалог создания проекта.
Открыть проект	Открывает диалог открытия проекта.
С чего начать?	По этой команде в нижней части окна открывается/скрывается стартовая страница (см. Стартовая страница). Вид стартовой страницы показан на рисунке ниже.







5.2. ПАНЕЛЬ ИНСТРУМЕНТОВ

Панель инструментов находится в верхней части среды разработки MasterSCADA 4D и занимает две строки под заголовком проекта.



Верхняя часть панели содержит панели быстрого доступа и вкладки.


Вид	Описание
	Открывает меню настроек среды разработки (редактора проектов)
	Открывает меню управления проектом
	Открывает диалог открытия проектов
	Сохраняет изменения, сделанные в проекте с момента последнего сохранения по текущий момент. После выполнения этой команды теряется возможность отмены и возврата последних действий через панель отмены/возврата действий.

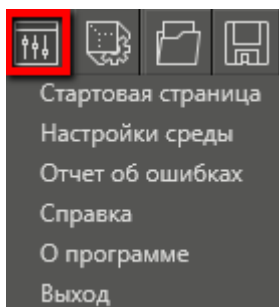
Вид	Описание
	После выполнения этой команды процесс среды разработки может занимать меньший размер оперативной памяти
	Панель отмены и возврата действий позволяет вернуться к любому состоянию проекта до последнего сохранения.
Вкладка Проект	Открывает панель инструментов, необходимую для работы с различными элементами проекта.
Вкладка Исполнение	Открывает панель инструментов, необходимую для работы со средой исполнения.
Вкладка Отладка	Открывает панель инструментов, необходимую для отладки проекта.
Вкладка Таблица	Открывается возможность табличного представления проекта. Составляя различные запросы можно отобразить элементы проекта, объединенные общим свойством, в табличном виде.
	Панель управления интерфейсом позволяет управлять отображением панелей редактора.
	Панель для работы со справочной системой.

Нижняя часть панели инструментов зависит от выбранной вкладки:

- Панель инструментов вкладки Проект
- Панель инструментов вкладки Исполнение
- Панель инструментов вкладки Отладка

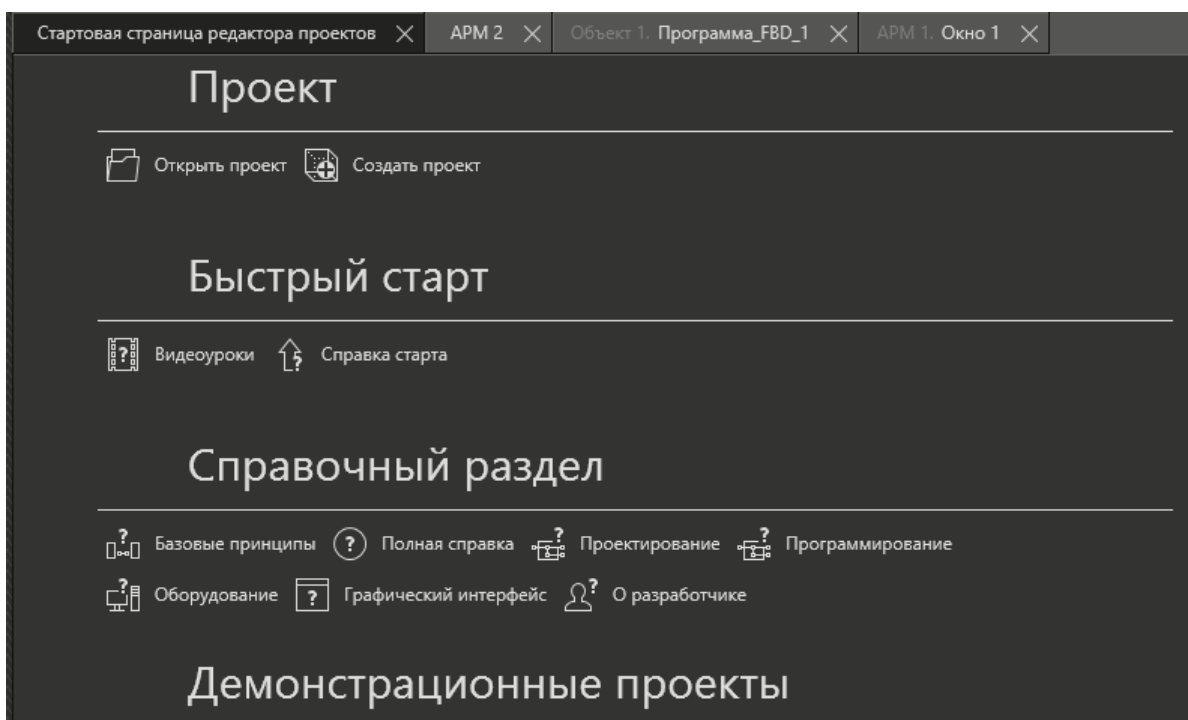
5.2.1. НАСТРОЙКИ РЕДАКТОРА

Для того, чтобы открыть/закрыть меню настроек редактора, используется кнопка , расположенная в левой части области меню редактора проекта:



Пункт меню Стартовая страница

Открывает стартовую страницу в окне документов и редакторов:



Стартовая страница содержит демонстрационные проекты, а также ссылку Видеоуроки на ресурс YouTube, на котором размещены видеопримеры разработки проектов.

Пункт меню Настройки среды

Открывает редактор общих настроек редактора проекта.

Пункт меню Отчет об ошибках

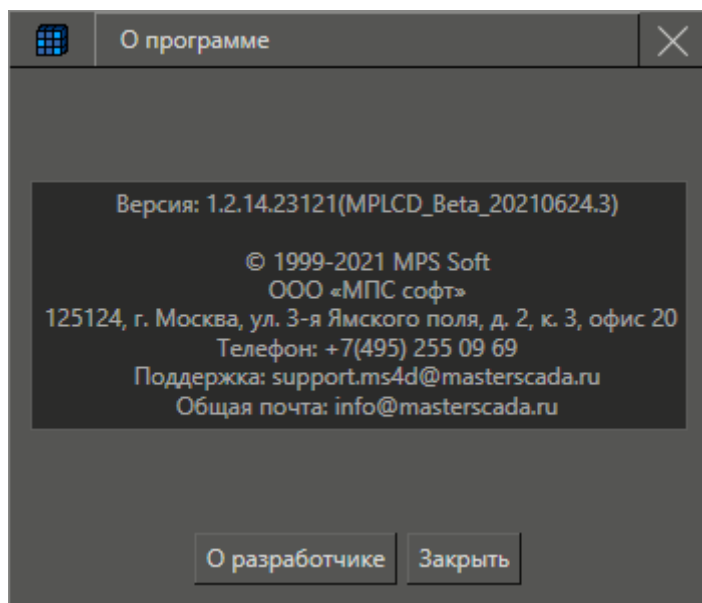
Открывает диалог параметров отчета об ошибках .

Пункт меню Справка

Открывает справочную систему в полном объеме.

Пункт меню О программе


Открывает диалог, содержащий информацию о версии программы и сроке ее обязательного обновления:



Инструменты диалога:

- О разработчике – показать информацию о компании "МПС Софт";
- Закреть – закрыть диалог.

Пункт меню Выход

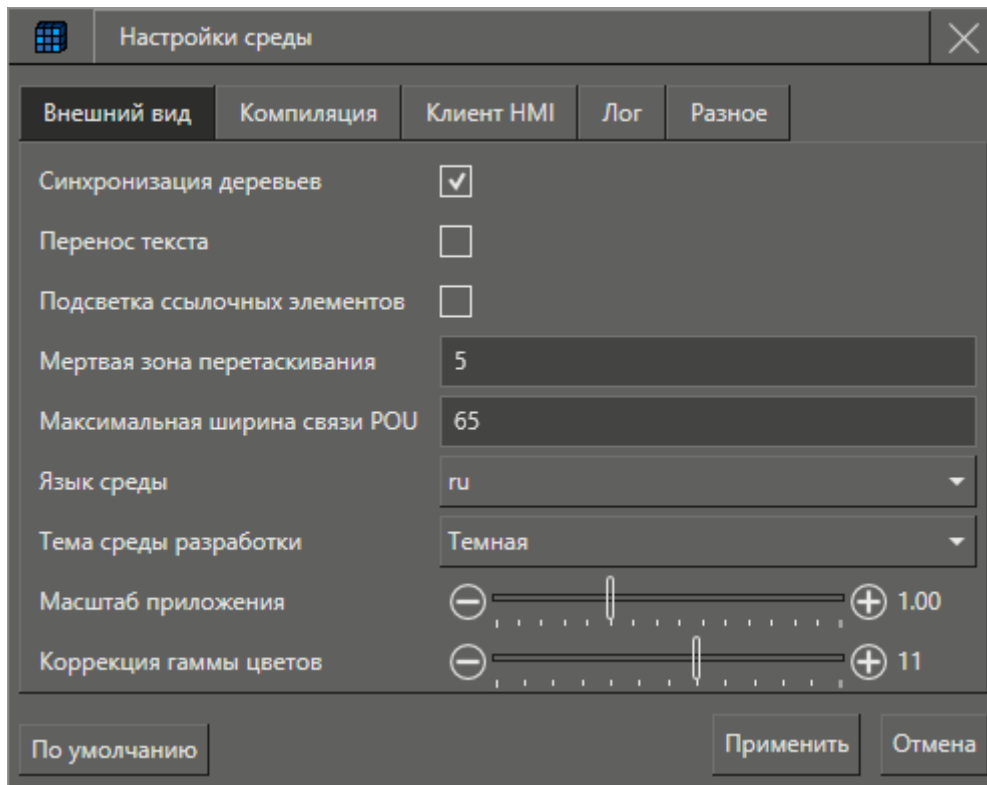
Закрывает текущее окно редактора проекта (аналог инструмента  – см. Типовые инструменты окон).

При закрытии окна закрывается проект, который был открыт в этом окне (см. Закрытие проекта).

5.2.1.1. НАСТРОЙКИ СРЕДЫ

Чтобы открыть диалоговое окно настроек среды необходимо выполнить команду Настройки среды в меню Настройки редактора.

Вид диалогового окна:



Окно содержит несколько вкладок:

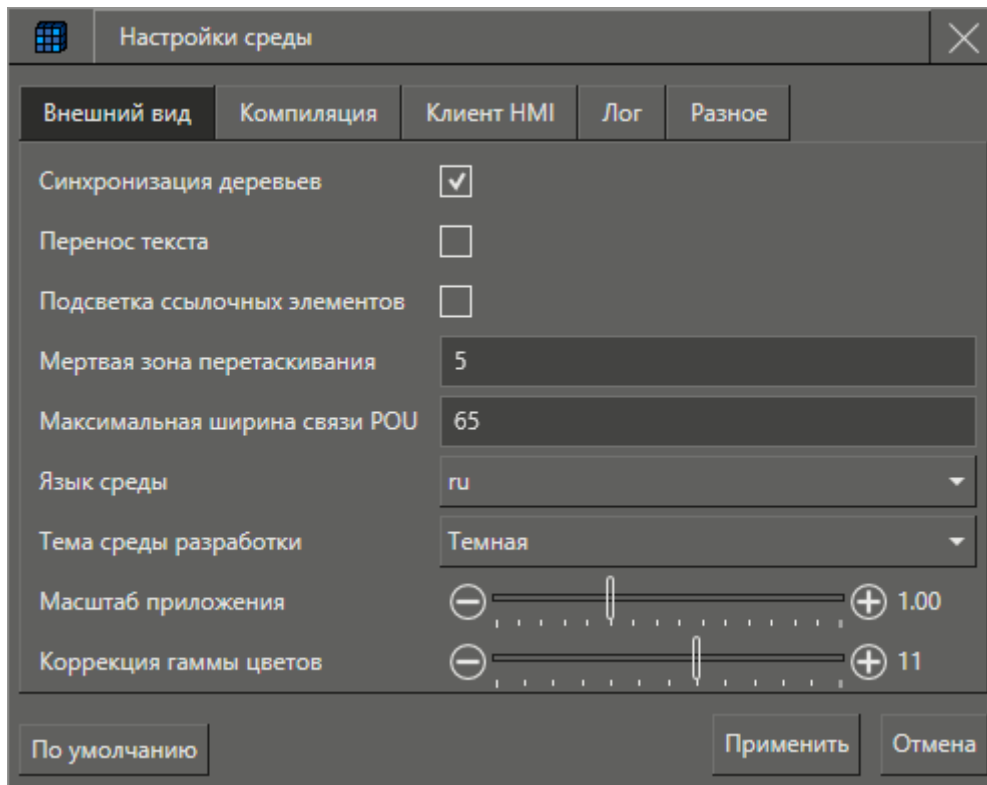
- Внешний вид
- Компиляция
- Клиент HMI
- Лог
- Вкладка Разное

Изменения, сделанные в окне, будут применены после нажатия на кнопку Применить или после закрытия окна (после нажатия на крестик в верхнем правом углу).

Кнопка По умолчанию открывает диалоговое окно, позволяющее сбрасывать все настройки, сделанные пользователем на всех вкладках.

5.2.1.1.1. ВКЛАДКА ВНЕШНИЙ ВИД

Вид вкладки Внешний вид окна Настройки среды:

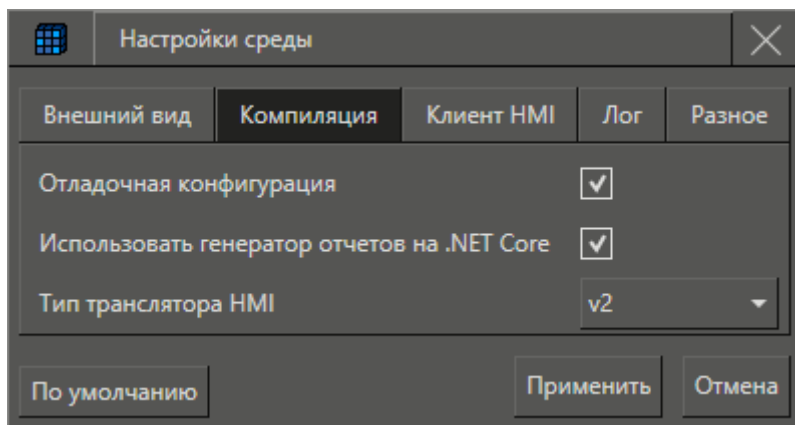


Свойство	Описание
Синхронизация деревьев	<p>Если этот флаг установлен, то во всех разделах дерева проекта выполняется автоматический переход к элементу, выделенному в любом редакторе или панели (в панели веток, в легенде, на схеме и т.п.). Переход выполняется в том случае, если раздел содержит такой элемент.</p> <p>Если флаг синхронизации деревьев не установлен, то автоматический переход к элементу в дереве проекта не производится.</p> <p>Во всех случаях, элемент, выделенный в какой-либо панели или редакторе, выделяется также во всех панелях и редакторах, содержащих этот элемент.</p>
Перенос текста	<p>Если этот флаг установлен, то текст, выходящий за пределы видимой области (например, в листинге ST), автоматически переносится на следующую строку. После изменения состояния флага необходимо закрыть и открыть снова редактор.</p>

Свойство	Описание
Подсветка ссылочных элементов	Если этот флаг установлен, то при наведении курсора на некоторый элемент А разрешена подсветка элементов, имеющих относительно элемента А Ссылку и Внешнюю ссылку .
Мертвая зона перетаскивания	Поле редактирования, в котором задается минимальное значение смещения (в пикселях) для операции перемещения. Если смещение выполняется на меньшую величину, то перемещение не производится.
Максимальная ширина связи ROU	Определяет максимальную ширину ножки функционального блока или функции, которая имеет связь с другим параметром.
Язык среды	Определяет интерфейс среды разработки. Поддерживаются языки: русский, английский. Для смены языка необходимо перезапустить приложение.
Тема среды разработки	Определяет цветовое оформление среды разработки. Поддерживаются варианты тем: темная, светлая. Список тем может быть расширен.
Масштаб приложения	Данный инструмент масштабирования работает так же, как и комбинированный инструмент масштабирования, но влияет на все панели и редакторы.
Коррекция гаммы цветов	Корректирует цвета редактора в рамках текущей темы приложения.

5.2.1.1.2. ВКЛАДКА КОМПИЛЯЦИЯ

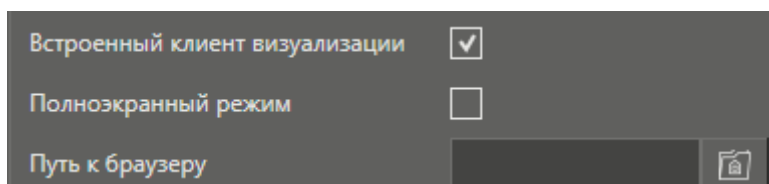
Вид вкладки Компиляция окна Настройки среды:



Свойство	Описание
Отладочная конфигурация	<p>Этот флаг следует установить в случае, если требуется вывод всех отладочных значений в редакторах FBD, LD, SFC.</p> <p>Влияет на отладку локальных переменных программ с типом доступа Temp:</p> <ul style="list-style-type: none"> • флаг установлен - Temp переменные транслируются как Local, но в начале вызова программы им проставляются начальные значения. • флаг снят - Temp переменные объявляются в начале вызова программы и недоступны для отладки.
Использовать генератор отчетов на .NET Core	Позволяет использовать генератор отчетов не на платформе Node.JS, а на .NET Core
Тип транслятор HMI	<p>Определяется, какая версия транслятора HMI будет использоваться при компиляции окон. Возможные варианты:</p> <ul style="list-style-type: none"> • v1 - при компиляции используется только версия транслятора HMI v1. • v1+v2 - используются обе версии, т.е. создается два комплекта файлов для работы в режиме исполнения. По умолчанию в клиенте открывается версия v1, чтобы получить доступ к версии v2 необходимо добавить /test в адресной строке браузера, либо в настройках приложений MasterSCADA 4D Monitor и MasterSCADA 4D Client Monitor. • v2 - при компиляции используется только одна версия транслятора HMI v2. <p>Если MasterSCADA 4D была установлена впервые, то по умолчанию настройка будет v2. Если на компьютере была установлена версия, где использовался транслятор v1, то при установке новой версии появится диалоговое окно, содержащее следующий текст: "У Вас используется устаревший транслятор HMI v1. Перейти на новый транслятор HMI v2? Данную настройку можно изменить в настройках приложения на вкладке Компиляция." Если нажать кнопку Да, то установится значение v2, если Нет, то останется предыдущий вариант.</p>

5.2.1.1.3. ВКЛАДКА КЛИЕНТ HMI

Вид вкладки Клиента HMI окна Настройки среды:

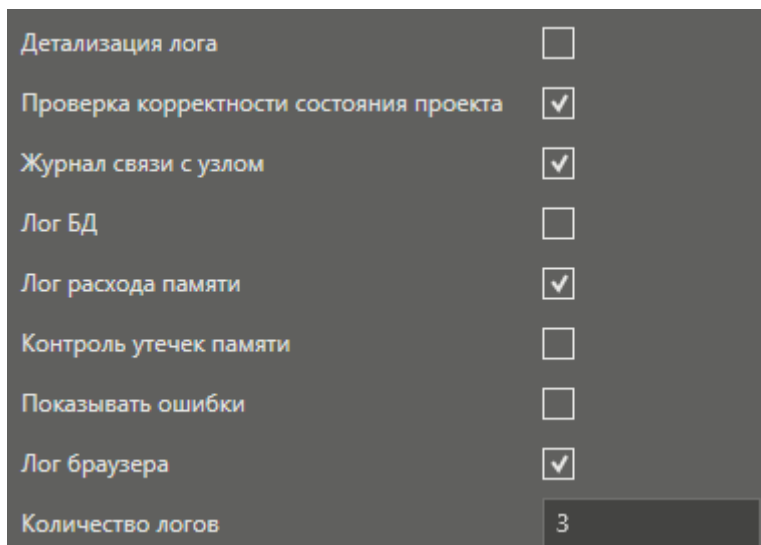


Свойство	Описание
Встроенный клиент визуализации	Если этот флаг установлен, то среда разработки при загрузке проекта в среду исполнения автоматически будет запускать в качестве клиента визуализации приложение, разработанное компанией "МПС Софт". Если флаг снят, то будет автоматически запускаться стороннее приложение. Тип приложения определяется настройкой Путь к браузеру.
Полноэкранный режим	Если этот флаг установлен, то, независимо от настроек Шаблона экрана, во всех проектах всегда будет запускаться клиент визуализации во весь экран.
Путь к браузеру	Если встроенный клиент визуализации не используется, то в данном поле прописывается путь к браузеру, который необходимо запускать при работе с исполнительной системой. Если поле пустое, то будет запускаться тот браузер, который используется в ОС по умолчанию.

Важно! Настройки, сделанные в данной вкладке, используются только для клиента визуализации, который запускается средой разработки в автоматическом режиме.

5.2.1.1.4. ВКЛАДКА ЛОГ

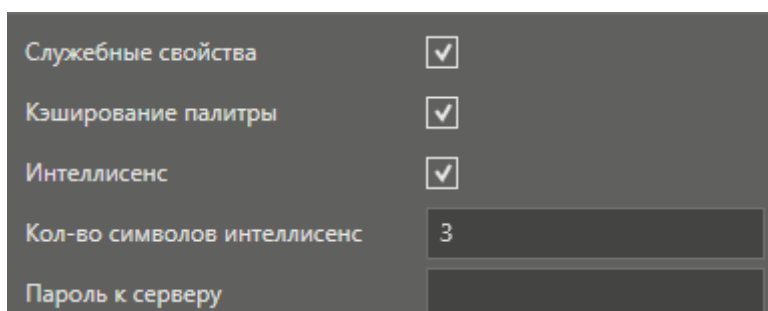
Вид вкладки Лог окна Настройки среды: :



Важно! Настройка данной вкладки должна производиться по рекомендации службы технической поддержки. В случае необходимости, сотрудники техподдержки дадут рекомендации по настройке данной вкладки.

5.2.1.1.5. ВКЛАДКА РАЗНОЕ

Вид вкладки Разное окна Настройки среды:

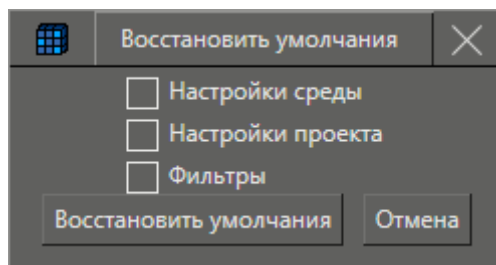


Свойство	Описание
Служебные свойства	Если этот флаг установлен, то в панели свойств у каждого элемента появляется категория свойств Служебные.
Кэширование палитры	Служебная настройка. Необходима для ускорения загрузки палитры элементов.
Интеллисенс	Включает интеллектуальные подсказки в редакторе ST.
Кол-во символов интеллисенс	После введения указанного количества символов редактор ST предлагает варианты развития программы.

Пароль к серверу	<p>Если в среде исполнения задан пароль на загрузку проекта, то загружать проект можно только из той среды разработки, в которой задан такой же пароль.</p> <p>Пароль в среде исполнения можно указать в настройках MasterSCADA 4D Monitor или в свойстве Параметры запуска RT узла</p>
------------------	---

5.2.1.1.6. ДИАЛОГ ВОССТАНОВИТЬ УМОЛЧАНИЯ

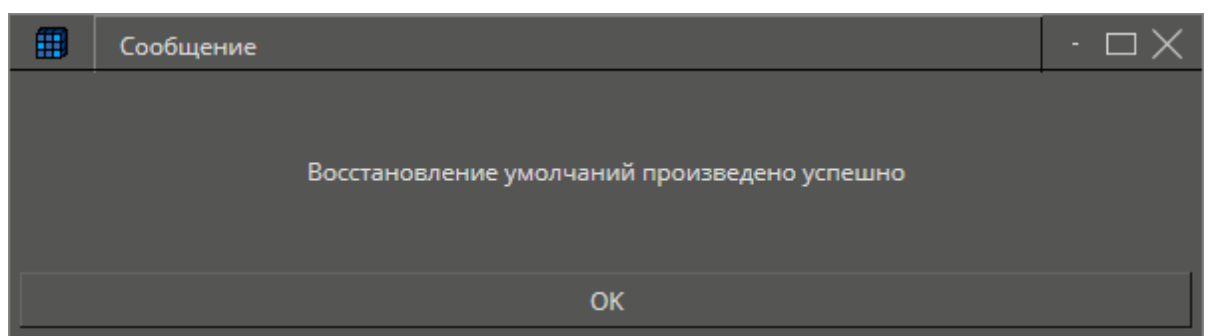
Для того, чтобы открыть диалоговое окно Восстановить умолчания, необходимо нажать на кнопку По умолчанию в диалоговом окне настройки среды.



В диалоге поддерживаются типовые инструменты окон.

При нажатии на кнопку Восстановить умолчания параметрам, которые определяются установленными флагами диалога, присваиваются значения по умолчанию.

В случае успешного восстановления на экране появляется соответствующее сообщение:



Флаг Настройки среды

Если этот флаг установлен, то при нажатии на кнопку Восстановить умолчания параметрам, для задания которых используется редактор общих настроек, присваиваются значения по умолчанию.

При восстановлении по этому флагу изменяются соответствующие поля в файле *%профиль*

пользователя%\AppData\Roaming\MPSSoft\MasterSCADA4D<версия>\MasterPLCUserSettings.xml.

Флаг Настройки проекта

Если этот флаг установлен, то при нажатии на кнопку Восстановить умолчания проект принимает вид по умолчанию (закрываются открытые документы, закрывается дополнительное окно структуры, сворачивается дерево, сбрасываются фильтры и т.п.).

При восстановлении по этому флагу изменяются соответствующие поля в файле *%профиль*

пользователя%\AppData\Roaming\MPSSoft\MasterSCADA4D<версия>\ProjectsConfig\<<имя проекта>\ProjectUserSettings.xml.

Флаг Фильтры

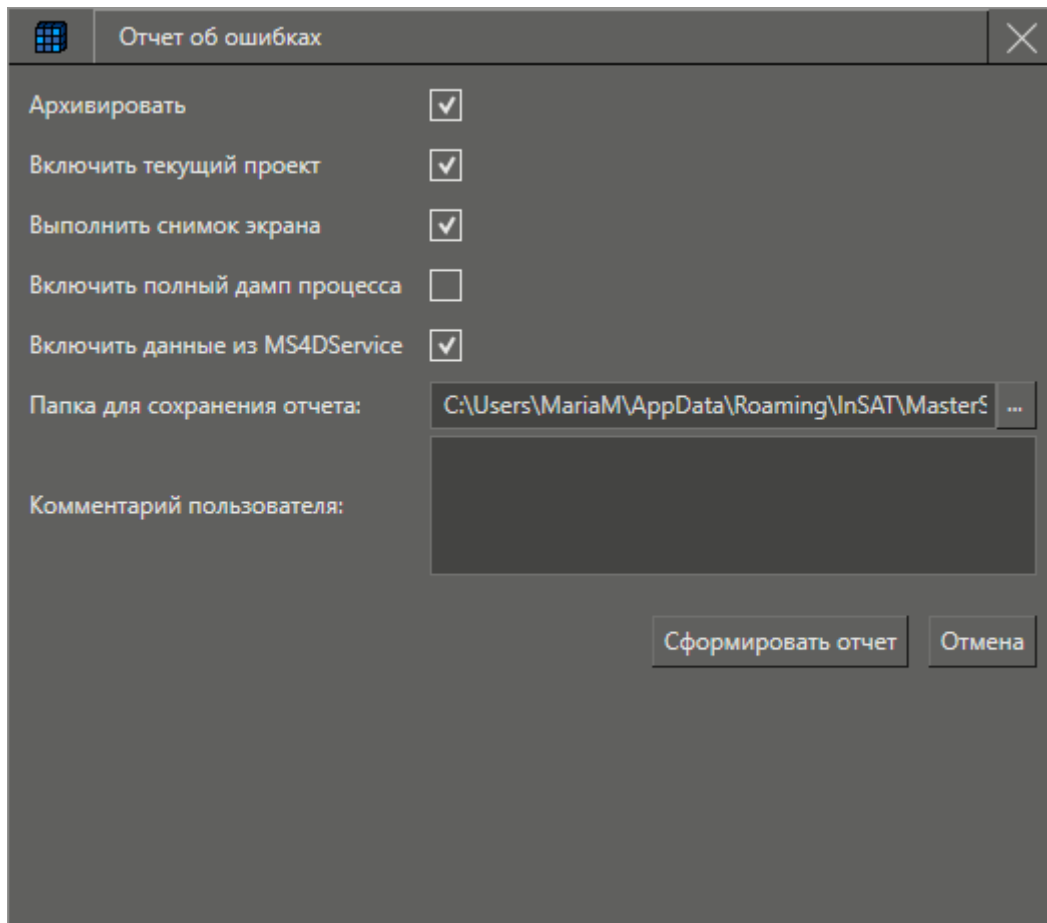
Если этот флаг установлен, то при нажатии на кнопку Восстановить умолчания удаляются пользовательские фильтры.

При восстановлении по этому флагу уничтожаются файлы *<напка проекта>\Settings\Filters\MasterSCADA.Tree.Editor_<имя фильтра>.xml.*

5.2.1.2. ДИАЛОГ ПАРАМЕТРОВ ОТЧЕТА ОБ ОШИБКАХ

Данное диалоговое окно открывается в различных ситуациях, когда существует необходимость создать отчет об ошибках. В любой момент времени отчет об ошибках можно сформировать, выполнив команду Отчет об ошибках в меню Настройки редактора.

Окно имеет вид:



Название	Описание
Архивировать	Отчет об ошибках – это набор файлов и папок. Если установлен данный флаг, то содержимое отчета об ошибках будет заархивировано (рекомендуется, т.к. размер отчета может быть большим и занимать много места на диске).
Включить текущий проект	Если этот флаг установлен, то в отчет об ошибках будет включен текущий проект. Рекомендуется устанавливать данный флаг, т.к. это значительно упростит работу службы техподдержки.
Выполнить снимок экрана	Если этот флаг установлен, то в отчет об ошибках попадет скриншот приложения, на момент формирования отчета.
Включить полный дамп процесса	В отчет об ошибках попадет дамп процесса среды разработки. Рекомендуется выполнять в случае, если есть подозрение на ошибку, возникающую при взаимодействии различных приложений.



Название	Описание
Включить данные из MS4DService	Если флаг установлен, то в лог будет включена отладочная информация среды исполнения, к которой была подключена среда разработки.
Папка для сохранения отчета	Задается папка, где сформируется отчет. По умолчанию это: <i>C:\Users\[имя пользователя]\AppData\Roaming\MPSSoft\MasterSCADA4DBeta\ErrorReports.</i>
Комментарий пользователя	Рекомендуется дополнять отчет об ошибках комментарием. Комментарий должен содержать ответы на следующие вопросы: что разработчик проекта хотел сделать?; что разработчик проекта сделал для того, чтобы решить поставленную задачу?; какой результат разработчик проекта получил в итоге?
Сформировать отчет	После нажатия на эту кнопку отчет сформируется в указанной папке. Имя отчета будет совпадать с датой и временем, когда он был сформирован. После окончания формирования появится диалоговое окно, в котором можно перейти к папке с отчетом, либо скопировать путь к папке в буфер обмена.
Отмена	Отменяет формирование отчета и закрывает окно.

5.2.2. ПАНЕЛЬ ОТМЕНЫ/ВОЗВРАТА ДЕЙСТВИЙ


Вид панели:

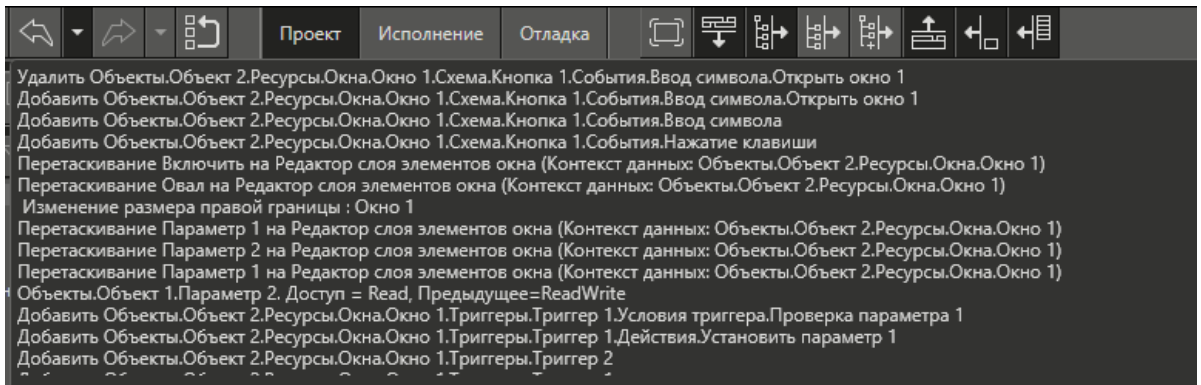


Инструменты отмены действий

-  – инструменты отмены действий;
-  (Ctrl+Z) – отменить последнее действие.

В поле редактирования данная команда равнозначна команде Esc. В то же время, в листинге ST по этой команде выполняется посимвольная отмена ввода.




-  – открыть список действий (список *UnDo*). Последнее действие отображается первым в списке, при наведении курсора действие выделяется:

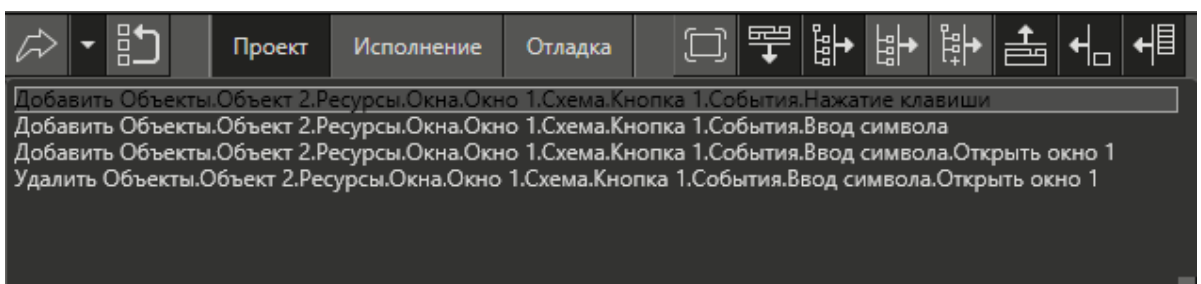


Отменяемое действие может быть макродействием, т.е. содержать несколько элементарных действий.

Если нажать в этом списке на некоторое действие А, то отменяются все последние действия вплоть до А и само действие А.


Инструменты возврата отмененных действий

-  – инструменты возврата (повторного выполнения) действий;
-  (Ctrl+Y) – вернуть последнее отмененное действие;
-  – открыть список отмененных действий (список *ReDo*). Последнее отмененное действие отображается первым в списке, при наведении курсора действие выделяется:



Если нажать в списке на некоторое отмененное действие А, то возвращаются все последние отмененные действия вплоть до А и само действие А.

История редактирования проекта

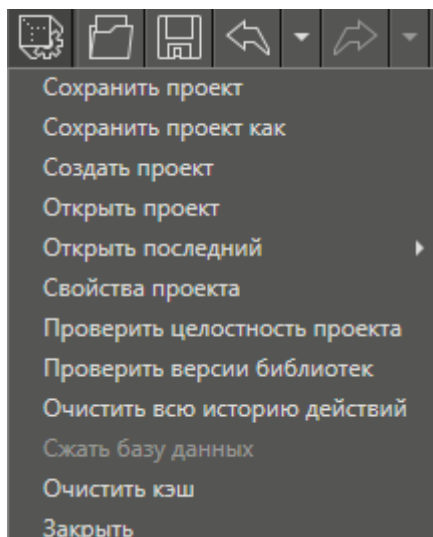
-  – открыть диалог История версий.

5.2.3. УПРАВЛЕНИЕ ПРОЕКТАМИ



Группа меню управления проектами открывается при нажатии на кнопку:

Вид меню:

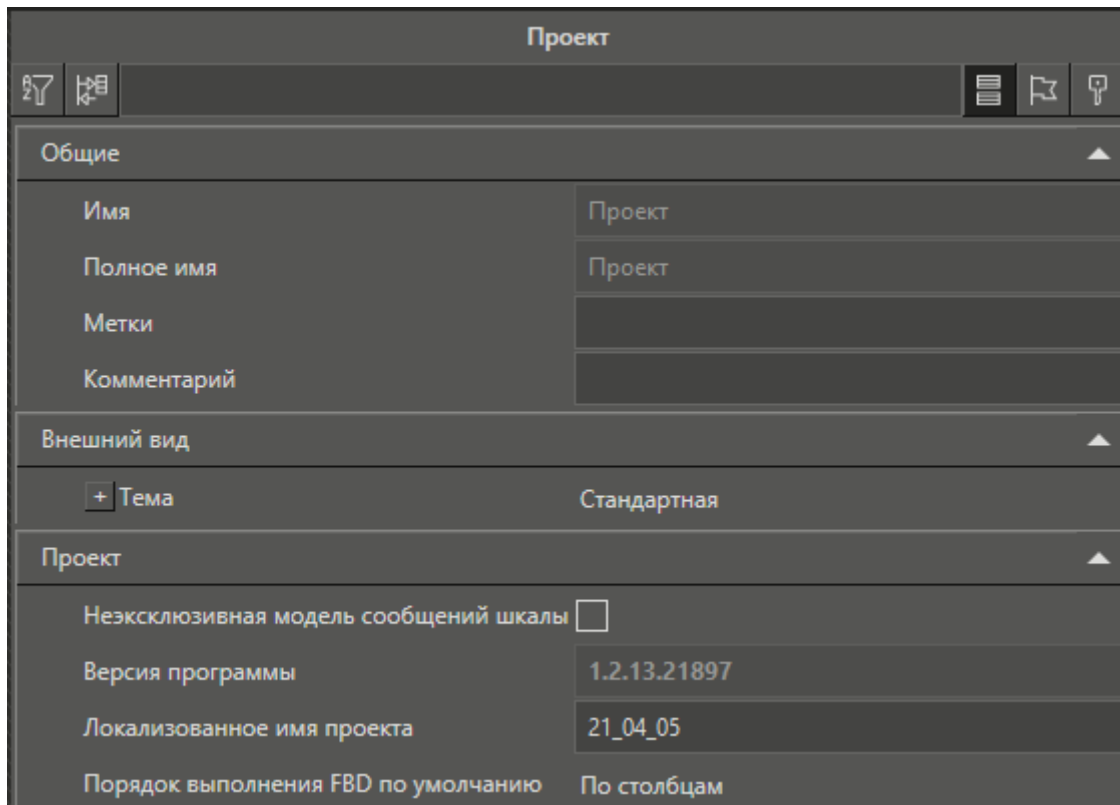


Название	Описание
Сохранить проект	Сохраняет изменения, сделанные в проекте с момента последнего сохранения по текущий момент. После выполнения этой команды теряется возможность отмены и возврата последних действий через панель отмены/возврата действий. После выполнения этой команды процесс среды разработки может занимать меньший размер оперативной памяти .
Сохранить проект как	Открывает диалог, который позволяет сохранить проект под другим именем. Рекомендуется в процессе разработки периодически делать резервные копии проекта, используя данный пункт меню.
Создать проект	Открывает диалог создания нового проекта в еще одном экземпляре среды разработки.
Открыть проект	Открывает диалог открытия существующего проекта в еще одном экземпляре среды разработки.

Название	Описание
Открыть последний	Открывает список последних открытых проектов в среде разработки. Если выбрать один из проектов, то откроется новый экземпляр среды разработки и запустится выбранный проект.
Свойства проекта	В панели свойств открывает свойства текущего проекта. Описание свойств приводится ниже.
Проверить целостность проекта	Проверяет проект на скрытые ошибки. В появляющемся диалоговом окне эти ошибки можно исправить автоматически.
Проверить версии библиотек	Проверяет соответствие используемых в данном проекте библиотек используемой версии среды разработки. Такая процедура запускается автоматически при открытии проекта.
Очистить всю историю действий	В процессе работы все действия разработчика фиксируются в базе данных проекта, благодаря чему можно вернуться ("откатиться") на любой момент разработки проекта. Данная команда удаляет всю историю действий разработчика проекта.
Сжать базу данных	Уменьшает размер проекта после очистки истории действий. Восстанавливает единую структуру базы данных.
Очистить кеш	Удаляет иконки палитры из памяти.
Заккрыть	Закрывает текущий проект.

Панель свойств. Свойства проекта

Свойства проекта открываются в панели Свойств после выполнения соответствующей команды в меню:




Специальные свойства:

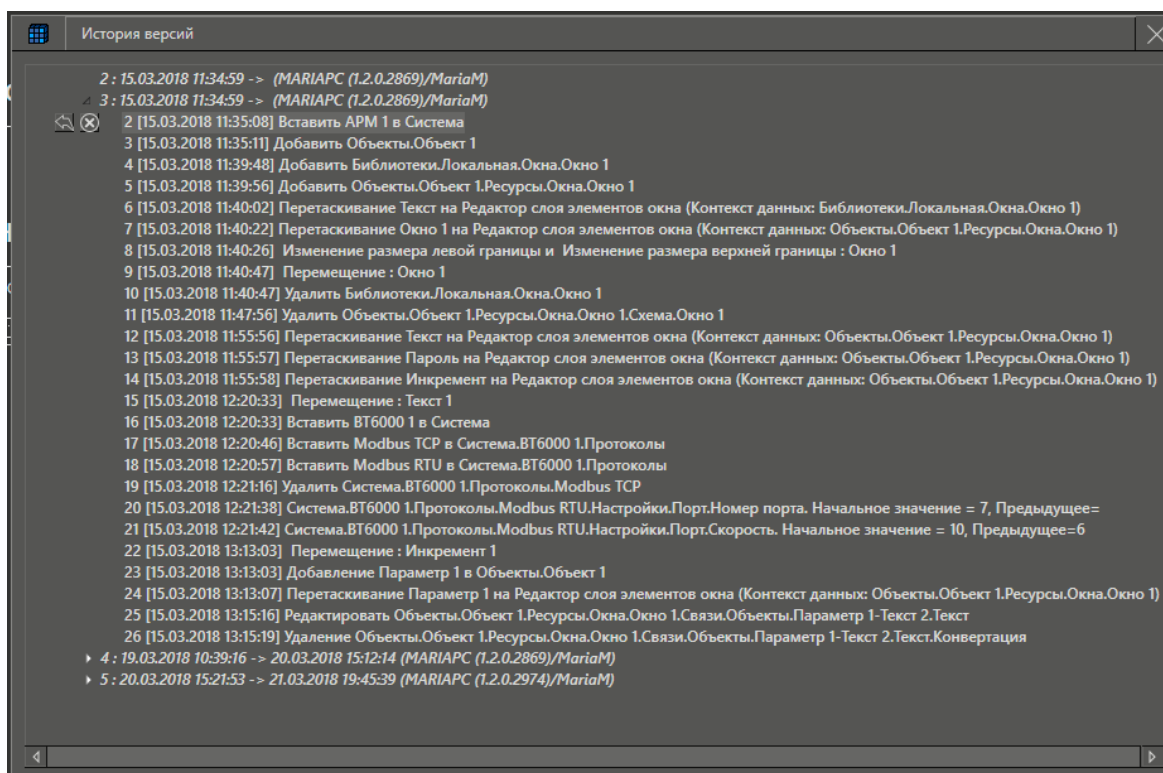
Название	Назначение
<p>Категория Внешний вид</p>	
<p>Тема</p>	<p>Определяется тема графических окон. Тема влияет на фон, тени и цвета элементов.</p> <p>Например, если сменить тему для учебного проекта Умный дом на Изумруд, то автоматически получится изображение</p> <div data-bbox="603 1554 1417 1944" data-label="Image"> </div> <p>Список возможных тем определяется в библиотеке.</p>

Название	Назначение
Категория Проект	
Неэсклюзивная модель сообщений шкал AI	<p>Флаг определяет модель срабатывания нарушения границ шкал.</p> <p>Если флаг снят, то используется эксклюзивная модель - нарушение аварийной границы отменяет активность предупредительной.</p> <p>Если флаг установлен, то используется неэсклюзивная модель - нарушение аварийной границы не отменяет активность предупредительной.</p>
Версия программы	Указывается версия программы, в которой ведется разработка проекта
Локализованное имя проекта	Если в этом свойстве задать русскоязычное имя, то оно будет отображаться в заголовке окна среды, а также в диалоге открытия проектов, но только в том случае, если в редакторе выбран русский язык (Язык среды = ru). В английской локализации это свойство не используется, и отображается имя файла проекта.
Порядок выполнения FBD по умолчанию.	Определяет значение настройки Порядок выполнения у программ FBD по умолчанию.

5.2.4. ИСТОРИЯ ВЕРСИЙ

Диалог История версий открывается при нажатии на иконку панели свойств .

В нем отображается история редактирования проекта в виде дерева:



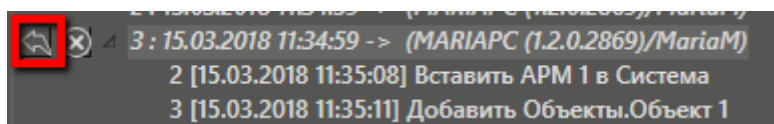
Корневые группы дерева – сессии редактирования проекта – имеют имена <ID сессии>: <время начала сессии> -> <время окончания сессии> (<имя компьютера> (<номер сборки MasterSCADA 4D>)/<имя пользователя>).

Подгруппами сессий являются действия пользователя. Эти подгруппы имеют имена <ID действия> <время действия> <описание действия>.

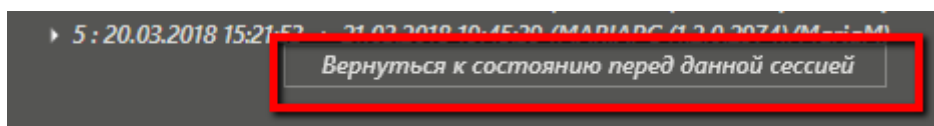
При раскрытии группы действий отображается список элементарных операций, которые выполняются программой в обеспечение этого действия.

Группы действий могут быть раскрыты только в том случае, если в редакторе общих настроек (см. Вкладка Служебное) установлен флаг Детализация истории версий.

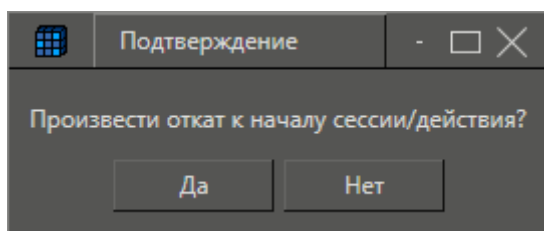
Для возврата к состоянию проекта перед каким-либо действием или перед всей сессией необходимо воспользоваться кнопкой, которая появляется при выделении строки данного диалога.



Сессия и действие имеют также контекстное меню, содержащее соответственно команду Вернуться к состоянию перед данной сессией и Вернуться к состоянию перед данным действием:



После нажатия на данные элементы управления появляется диалог:



После нажатия на кнопку Да происходит возврат к состоянию перед началом данной сессии или к состоянию перед данным действием. Иногда эта процедура занимает длительный период времени (несколько минут).

При помощи кнопки  можно удалить историю с начала и до текущего действия/сессии.



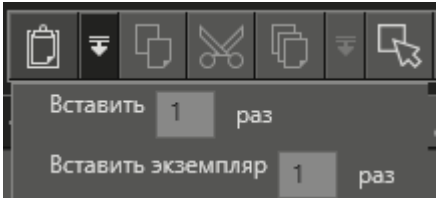

После удаления истории действий рекомендуется выполнить действие Сжать базу данных в меню Управление проектом.

5.2.5. ВКЛАДКА ИНСТРУМЕНТОВ ПРОЕКТ

Вкладка инструментов Проект имеет вид:



При открытии проекта эта вкладка инструментов автоматически активируется.

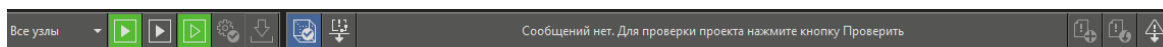
Вид	Название	Описание
	Вставить (CTRL+V)	Добавляет в дерево сохраненный в буфер фрагмент. Если нажать на кнопку  , то откроется диалог, в котором можно задать количество копий, которые необходимо вставить: <div data-bbox="780 1512 1217 1709" style="border: 1px solid black; padding: 5px; margin: 10px 0;">  </div>
	Копировать (CTRL+C)	Копирует выделенное в буфер обмена.

Вид	Название	Описание
	Вырезать (CTRL+X)	Вырезает выделенное в буфер обмена.
	Дублировать (CTRL+D)	Дублирует выделенное. Если нажать на кнопку  , то откроется диалог, в котором можно задать количество копий, которые необходимо сделать: 
	Выбрать все (CTRL+A)	Выбирает все дочерние элементы.
	Удалить (Delete)	Удаляет выделенное.

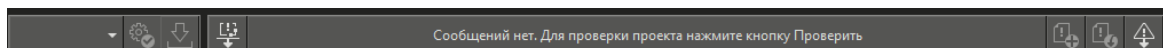
Если открыт какой-либо редактор, то вкладка инструментов Проект дополняется панелью инструментов, применимой для конкретного редактора.

5.2.6. ВКЛАДКА ИНСТРУМЕНТОВ ИСПОЛНЕНИЕ

Вкладка инструментов Исполнение имеет вид если в проект добавлены узлы:









Вид вкладки инструментов, если в проекте нет узлов:





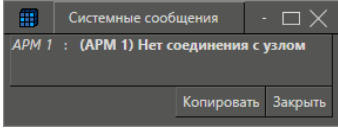


Вкладка содержит панель Устройства и панель Компиляция.



Панель Устройства

Назначение элементов:

Вид	Описание
	Отображает, в какой узел будет происходить загрузка проекта или производиться имитация проекта, а также позволяет выбрать необходимый узел.
	Кнопка подключения к среде исполнения. Если в панели свойств узла установлен флаг Автозапуск исполнительной системы, то в случае когда исполнительная система не запущена, среда разработки попытается запустить ее. При нажатии на кнопку  можно выбрать способ подключения с загрузкой конфигурации и без загрузки конфигурации.
	Запускает эмуляцию исполнительной системы для выбранного узла. При этом проект запускается в той среде исполнения, которая установлена на тот же компьютер, что и среда разработки. Опрос устройств при этом не производится.
	Показывает отличается ли конфигурация, загруженная в исполнительную систему, от той которая была скомпилирована в среде разработки.
	Осуществляет загрузку конфигурации в уже подключенный узел.

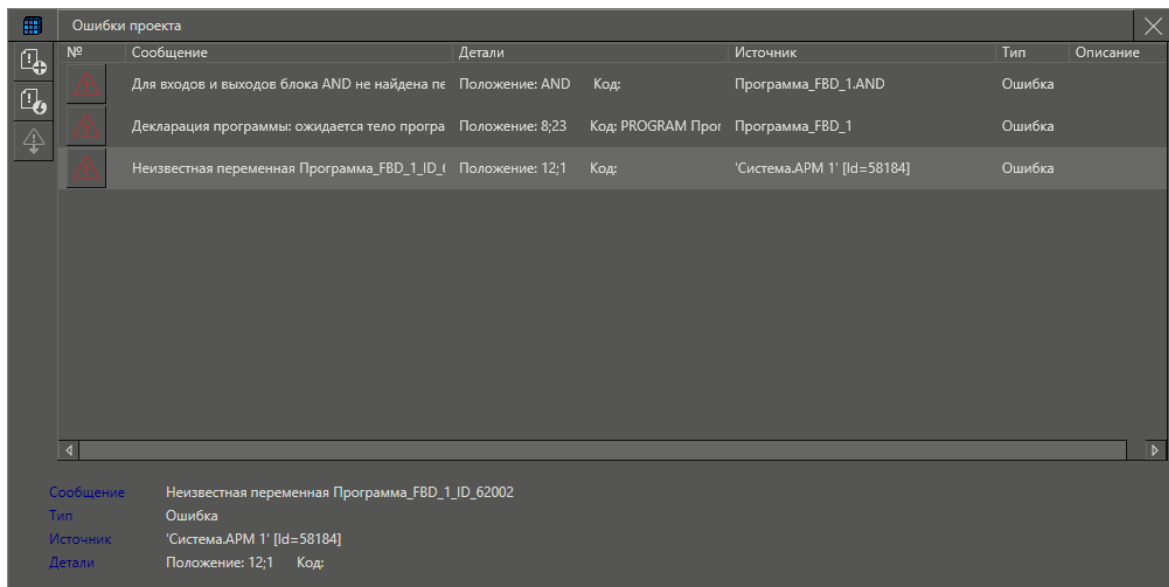
Панель Компиляция

Вид	Описание
	Кнопка проверки конфигурации проекта на ошибки. При проверке происходит компиляция проекта.
	Открывает системное окно сообщений: 
	Поле отображает ошибки, найденные при компиляции проекта.
	В случае затруднений с самостоятельным выявлением причин возникших ошибок, имеется возможность при помощи данной кнопки сформировать отчет об ошибках

Вид	Описание
	для последующей отправки его в службу технической поддержки.
	Копирует текст ошибок, возникших при компиляции, в буфер. В дальнейшем его можно вставить в какой-либо текстовый файл для анализа.
	Открывает окно, в котором отображаются все ошибки, возникшие при компиляции.

5.2.6.1. ОКНО ОШИБКИ ПРОЕКТА

Данное окно содержит таблицу с детальной информацией об ошибках проекта, возникших во время компиляции. Информация об отдельной ошибке образует строку таблицы. Если выделить какую-либо строку в нижней части окна, также выдается вся информация об ошибке:



Столбцы сообщений:

Название	Описание
Сообщение	Выдается текст сообщения
Детали	В случае ошибки программного элемента столбец содержит ее детальное описание. Место ошибки указывает параметр

Название	Описание
	<p>Положение n;m, где n – номер строки с ошибкой, m – знакоместо ошибки в строке.</p> <p>Для поиска указанной ошибки в тексте ST-компонента можно перемещать курсор в указанной строке, наблюдая за номером знакоместа в поле позиции курсора</p>
Источник	Указывается элемент, вызвавший сообщение об ошибке.
Тип	Указывается тип неисправности, например, ошибка, предупреждение и т.п.
Описание	Дополнительное описание проблемы.

5.2.7. ВКЛАДКА ИНСТРУМЕНТОВ ОТЛАДКА

Вкладка инструментов Отладка используется в режиме исполнения, т.е. когда среда разработки подключена к среде исполнения.

Если среда разработки не подключена к среде исполнения, то все кнопки панели не активны.



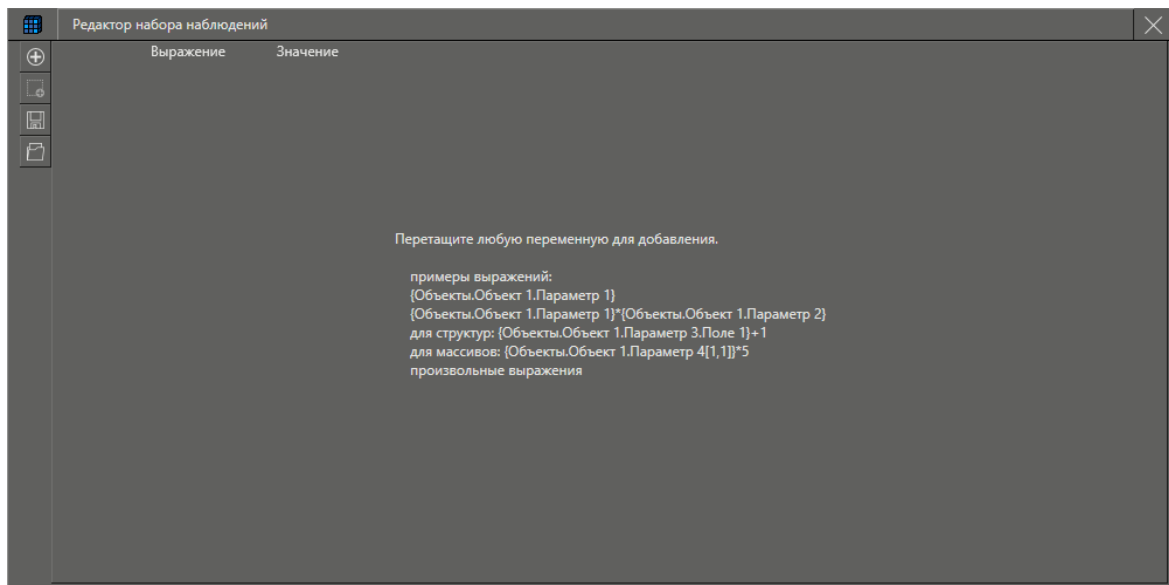
Вид	Назначение
	Если кнопка нажата, то программа перестает выполняться в среде исполнения.
	Запускает исполнение программы в исполнительной системе. Каждая задача узла отработывает 1 цикл и останавливается.
	Кнопка блокирует значения выбранных параметров. В этом режиме устанавливается текущее значение, а значения, пришедшие по связи, игнорируются.

Вид	Назначение
■	Снимает блокировку с выбранных параметров.
■	Открывает диалоговое окно Редактор набора наблюдений.

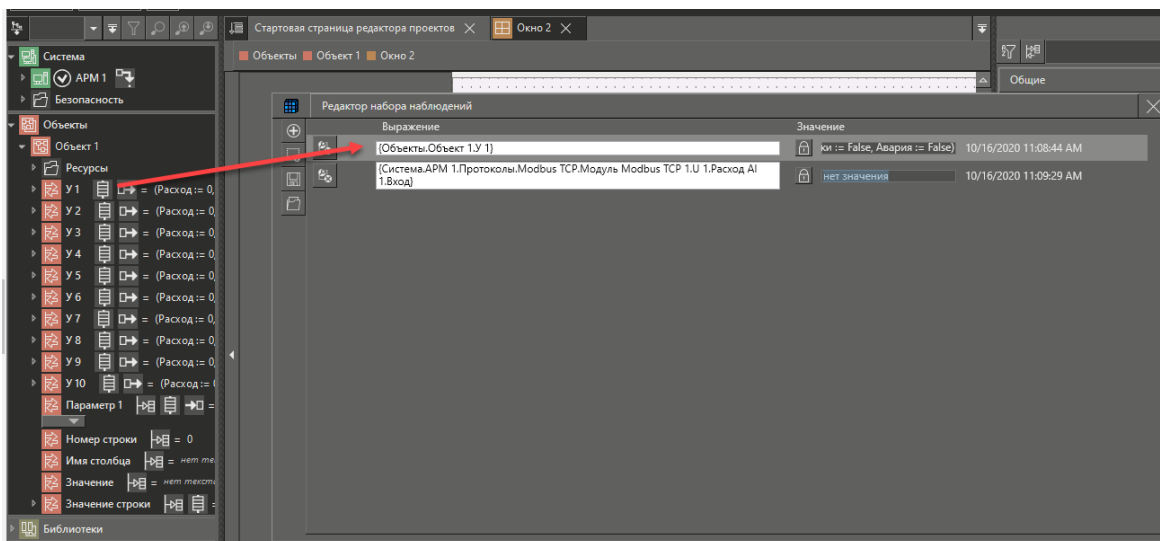
5.2.7.1. РЕДАКТОР НАБОРА НАБЛЮДЕНИЙ

Редактор набора наблюдений открывается по нажатию на соответствующую кнопку панели инструментов вкладки Отладка. Диалоговое окно редактора служит для удобного наблюдения за параметрами в режиме исполнения. Если среда разработки подключена к среде исполнения, то значения всех параметров отображаются в редакторах программ FBD, LD и SFC в дереве проекта, в панели клеммников. Если параметры находятся в разных частях проекта, то наблюдение за ними стандартными способами может быть затруднительно. Все интересующие параметры можно перетащить в диалоговое окно Редактор набора наблюдений.



Вид окна при первом открытии:









Для того чтобы добавить какой-либо параметр нужно перетащить его из дерева проекта в поле редактора наблюдений:



Элементы окна:

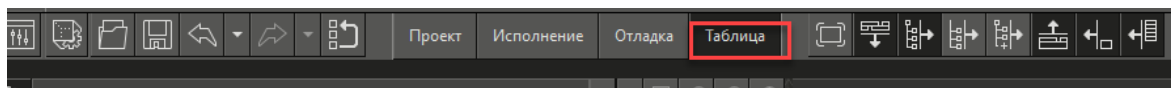
Элемент	Назначение
<p>Столбец Выражение</p>	<p>В данном столбце отображается полное имя контролируемого параметра в проекте. Может быть отредактировано вручную. В набор наблюдений могут быть добавлены параметры разных типов, в том числе отдельные элементы массивов и структур.</p> <p>Правила формирования выражения:</p> <ul style="list-style-type: none"> • Полное имя параметра указывается в фигурных скобках {} • Индекс массива указывается в квадратных скобках []. • Для массивов и структур можно задавать сложные выражения. Например, если для массива указать выражение вида: {Параметр1[1,1]}+{Параметр1[1,2]}+1, то получим в столбце значение сумму трех слагаемых. <p>аналогично сложные выражения для структур</p>
<p>Столбец Значение</p>	<p>Отображает текущее значение параметра. Для того, чтобы изменить установленное значение, необходимо нажать на текущее значение, после чего можно ввести новое:  54. После принятия изменений новое значение будет присвоено и заблокировано. Рядом со значением будет нажата кнопка: . При необходимости блокировку можно снять, отжав кнопку. Справа от значения</p>

Элемент	Назначение
	отображается время последнего изменения контролируемого параметра
	Удаляет параметр из набора наблюдения.
	Очищает список.
	Добавляет новую пустую строку в набор наблюдений. Ее можно будет заполнить вручную, прописав в дереве имя и путь к параметру. В случае ввода некорректных данных появится знак: 
	Позволяет сохранить текущий набор наблюдений. После нажатия на кнопку открывается стандартное диалоговое окно Windows. Сохраненный файл будет иметь расширение xml.
	Позволяет открыть ранее сохраненный набор наблюдений. После нажатия на кнопку открывается стандартное диалоговое окно Windows

Набор наблюдений сохраняется автоматически до тех пор, пока разработчик проекта не очистит список.

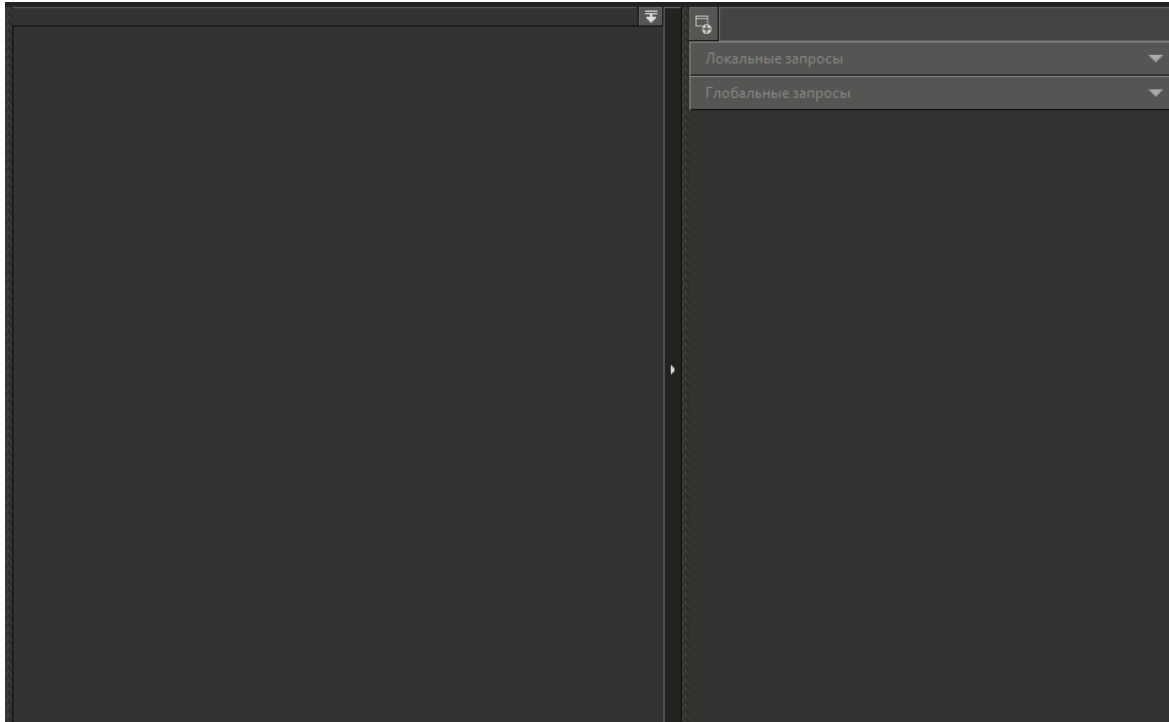
5.2.8. ТАБЛИЧНОЕ ПРЕДСТАВЛЕНИЕ ПРОЕКТА

Табличное представление проекта открывается при переходе на вкладку Таблица в панели инструментов.



Составляя различные запросы, можно отобразить элементы проекта, объединенные общим свойством, в табличном виде, изменять значения их свойств.

Вид вкладки Таблица по умолчанию:



Работа со вкладкой начинается с создания запроса.

Панель с запросами находится в левой части вкладки. Ее можно свернуть/развернуть нажав на левую границу. Изменить ширину можно потянув левой кнопкой мыши за левую границу.

Различают Локальные запросы и Глобальные запросы:

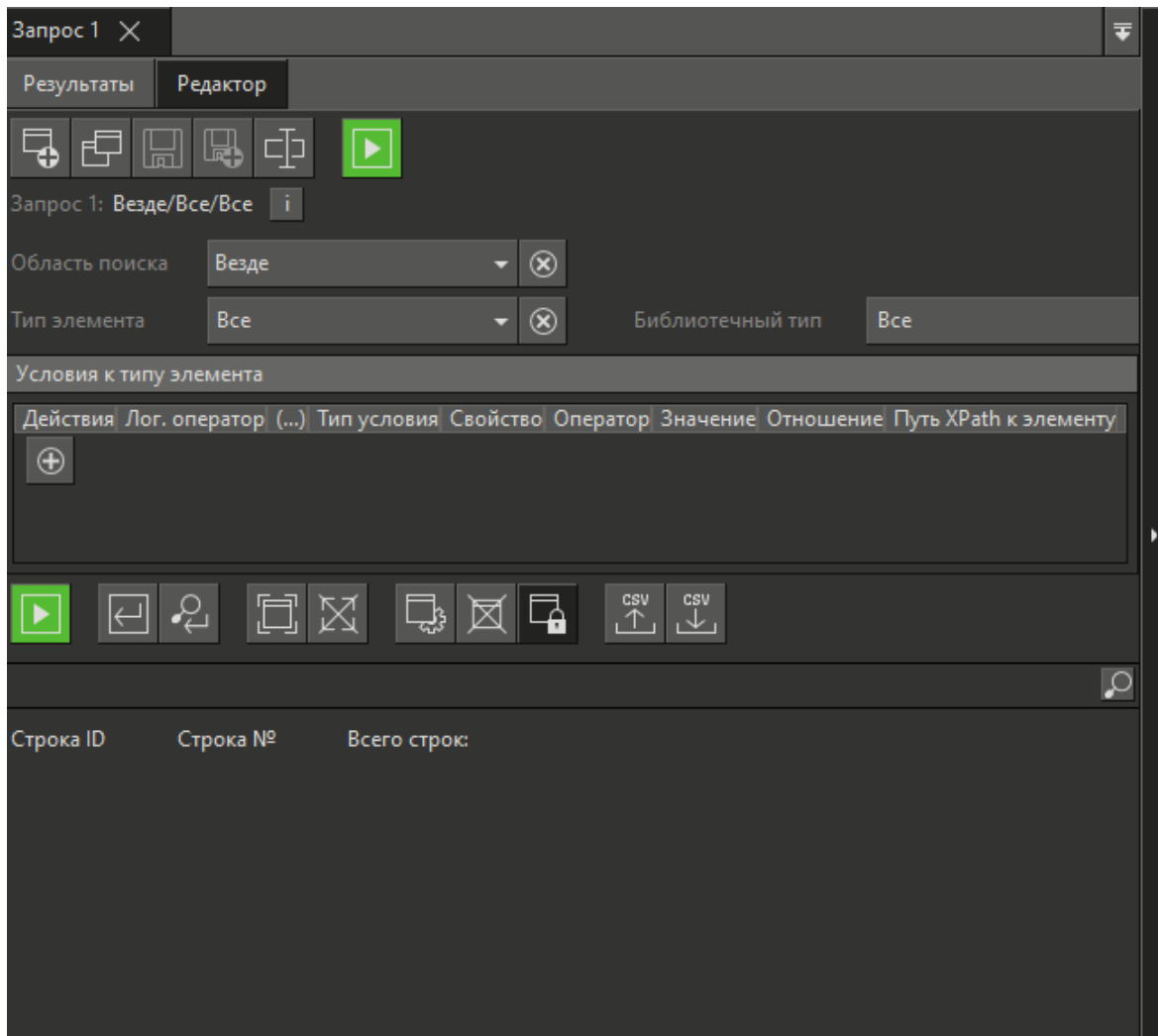
- Локальные запросы доступны только в текущем проекте
- Глобальные запросы после сохранения будут доступны и в других проектах.

Запросы сохраняются в папке `Queries`. Глобальные запросы в папке `c:\Users\[имя пользователя]\AppData\Roaming\MPSSoft\MasterSCADA4D[номер версии]\ProjectsConfig\Queries`

Чтобы добавить запрос нужно нажать на кнопку



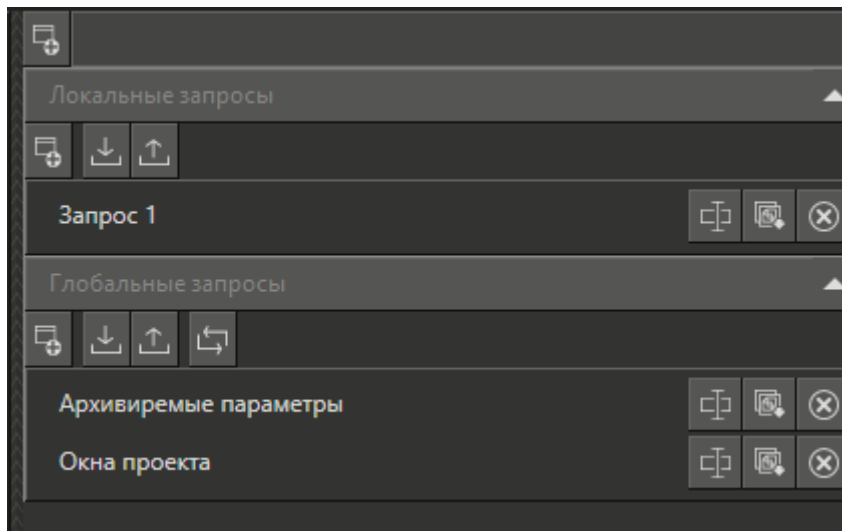
Появится вкладка запроса:







Вкладка в свою очередь имеет две вкладки:

- Редактор - в верхней части формируется запрос, а в нижней части отображаются результаты. См. раздел Формирование запросов
- Результаты - отображаются результаты запроса. См. раздел Работа с результатами запросов.


После того как запросы будут созданы они отобразятся в панели запросов:





Каждая группа запросов имеет панель инструментов:


Элемент	Название	Описание
	Создать новый запрос	Создает новый запрос в соответствующей группе.
	Импортировать запросы	Открывает диалоговое окно Windows, которое позволяет выбрать запросы, которые находятся в произвольном месте и добавить их в проект.
	Экспортировать запросы	Открывает диалоговое окно Windows, которое позволяет выбрать папку, куда сохранятся запросы из текущей группы запросов проекта.
	Обновить список запросов	Проверяет наличие глобальных запросов и обновляет список в группе.

Каждая строка запроса имеет следующие элементы:

Элемент	Название	Описание
	Переименовать запрос	Открывает диалоговое окно, в котором можно переименовать запрос.

	Сделать глобальным/локальным запрос	Перемещает запрос между группами. Если запрос был локальным, то он может быть перенесен в глобальные и наоборот.
	Удалить запрос	Открывает диалоговое окно подтверждения удаления запроса. Запрос будет удален, если нажать в диалоговом окне кнопку Да.

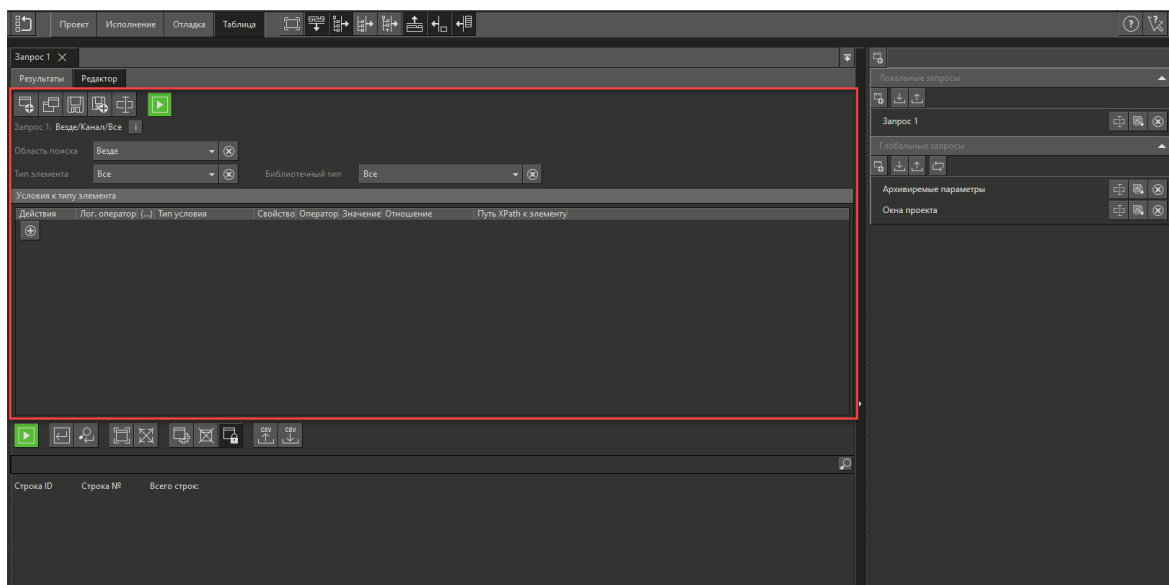
Порядок работы с запросами

1. Создать новый запрос
2. Определить основные и дополнительные параметры запроса
3. Запустить исполнение запроса нажав на кнопку 
4. Обработать результаты запроса: сохранить, изменить значения свойств элементов и др.

5.2.8.1. ФОРМИРОВАНИЕ ЗАПРОСОВ

Сформировать запрос для табличного представления проекта можно во вкладке Редактор. В верхней части находится редактор запросов. В нижней части можно отобразить результаты запроса, чтобы проверить правильно ли он составлен. Работать с результатами удобнее на вкладке Результаты.

Редактор запросов выделен красной рамкой:






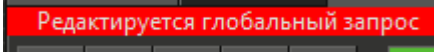

Порядок работы:

1. Определить параметры запросы: основные и дополнительные.

2. Нажать кнопку  Запустить запрос

Редактор открывается при добавлении нового запроса, либо при двойном нажатии левой кнопки мыши по уже существующему в панели запросов.

Панель инструментов редактора запросов:




Элемент	Название	Описание
	Создать новый запрос	Открывает вкладку для создания нового запроса.
	Дублировать запрос	Открывает вкладку нового запроса, с текущими настройками
	Сохранить	Сохраняет редактируемый
	Сохранить как глобальный запрос	Добавляет редактируемый запрос в группу Глобальные. При редактировании глобального запроса появляется в верхней части вкладки надпись: 
	Переименовать запрос	Открывает диалоговое окно, в котором можно переименовать запрос.
	Запустить запрос	Запускает выполнение запроса. Результаты запроса отобразятся в нижней части текущей вкладки или на вкладке Результаты.

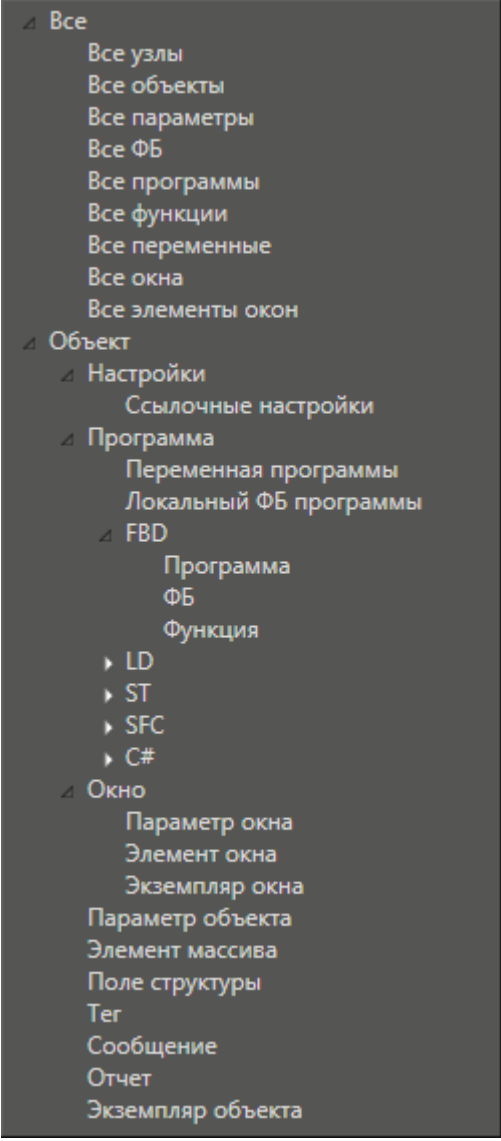
Формирование условий запроса происходит в два этапа:


1. Определяются основные параметры
2. Определяются дополнительные условия выборки.

Основные параметры запроса

Основные параметры запроса определяются следующими полями:

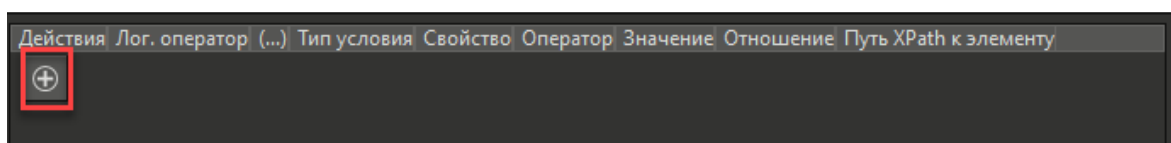
Название	Описание
Область поиска	<p>Определяет место выполнения поиска элементов для отображения. Определение условий запроса нужно начинать с данного поля. Возможные варианты:</p> <ul style="list-style-type: none"> • Везде • В системе • В объекте • В библиотеке • В выделенном элементе дерева. <p>Если выбран вариант В выделенном элементе дерева, то появится дополнительное поле Текущий элемент дерева.</p>
Текущий элемент дерева	<p>Поле доступно, если в поле Область поиска выбрано В выделенном элементе дерева. Вид поля:</p> <div data-bbox="609 1041 1441 1102" style="border: 1px solid black; padding: 2px; margin: 5px 0;"> Текущий элемент дерева Объекты.Объект 1 ✓ ✕ </div> <p>При первом обращении к полю в нем отобразится выделенный элемент в дереве, чтобы выполнить поиск в этом элементе нужно нажать на кнопку  - область поиска будет зафиксирована.</p> <p>Чтобы выбрать другой элемент для поиска нужно нажать на кнопку  - сбросить текущую область, затем выбрать новый элемент в дереве и зафиксировать область поиска кнопкой .</p>
Тип элемента	<p>Определяет, среди каких элементов проекта нужно выполнять поиск. Вид выпадающего списка зависит от поля Область поиска. Вид выпадающего списка для области поиска Везде</p>

Название	Описание
	
Библиотечный тип	Позволяет выбрать нужный библиотечный тип выбранного элемента в поле Тип элемента

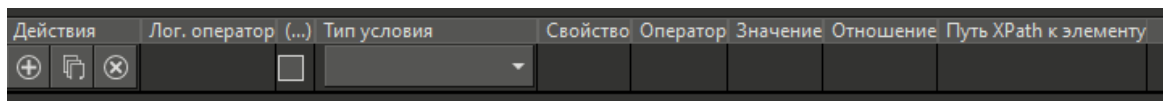
Установить значение поля по умолчанию можно нажав на кнопку  .

Дополнительные параметры запроса

Выбор дополнительных параметров заключается в том, что определяются дополнительные условия к типу элемента. Для этого нужно нажать на кнопку в таблице:






Появится строка, в которой можно задать необходимые значения. Вид строки по умолчанию:



В строке заполнять можно те столбцы, которые имеют более светлый фон. Порядок столбцов в таблице может быть изменен. Для этого необходимо перетащить левой кнопкой мыши заголовок столбца в нужное место.

Столбец Действия

В столбце Действия находятся кнопки для управления запросом:

Вид	Название	Назначение
	Вставить условие ниже	Добавляет ниже новую пустую строку для формирования еще одного условия
	Дублировать условие	Добавляет ниже новую строку для формирования еще одного условия с настройками, как у текущего условия.
	Удалить условие	Удаляет текущее условие. Если условие содержит вложенные условия, то они тоже удалятся.

Столбец Тип условия

В зависимости от выбранного значения поля Тип условия будут доступны разные столбцы.

Порядок настройки для каждого Типа условия смотрите в соответствующих разделах:

- Тип условия запроса - Свойство
- Тип условия запроса - Отношение
- Тип условия запроса - Xpath

Столбец Лог.оператор

Столбец Лог.оператор используется, если добавлено больше одного запроса. В этом случае в выпадающем списке можно выбрать:

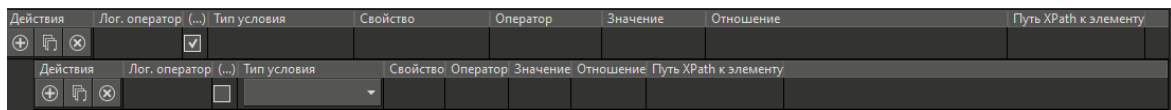
- ИЛИ - в выборку попадут те элементы, которые удовлетворят любому из запросов.
- И - в выборку попадут только те элементы, которые удовлетворят сразу всем запросам.

- И НЕ - в выборку попадут только те элементы, которые удовлетворят другим запросам и не удовлетворит текущему.
- ИЛИ НЕ - в выборку попадут элементы, которые удовлетворят другим запросам или не удовлетворят текущему.

Пример:

Столбец (...)

Столбец (...) используется, если необходимо обобщить несколько условий запроса, взять их в скобки. Если установить флаг в данном столбце, то таблица будет иметь вид:

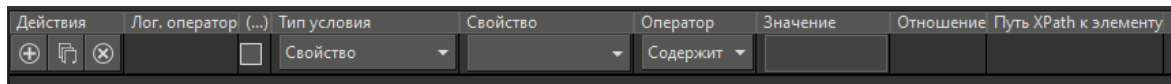


Т.е. появится возможность добавить несколько условий на вложенном уровне.

Пример:

5.2.8.1.1. ТИП УСЛОВИЯ ЗАПРОСА - СВОЙСТВО

Если выбрано значение Свойство в столбце Тип условия запроса, то нужно заполнить столбцы:

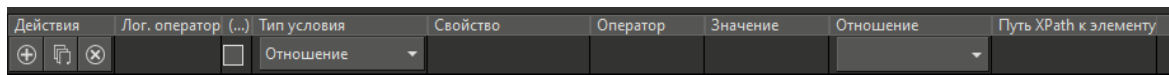


Название	Описание
Свойство	В выпадающем списке будут доступны свойства, которые характерны для выбранного Типа элемента. Так например для элементов Окно и Параметр список свойств будет отличаться.
Оператор	определяет оператор для сравнения значения выбранного свойство со значением, которое будет введено в следующее поле. Возможные варианты: <ul style="list-style-type: none"> • Содержит - значение свойства содержит указанную последовательность символов • Не содержит - значение свойства не содержит указанную последовательность символов • == - значение свойства равно указанной последовательности СИМВОЛОВ

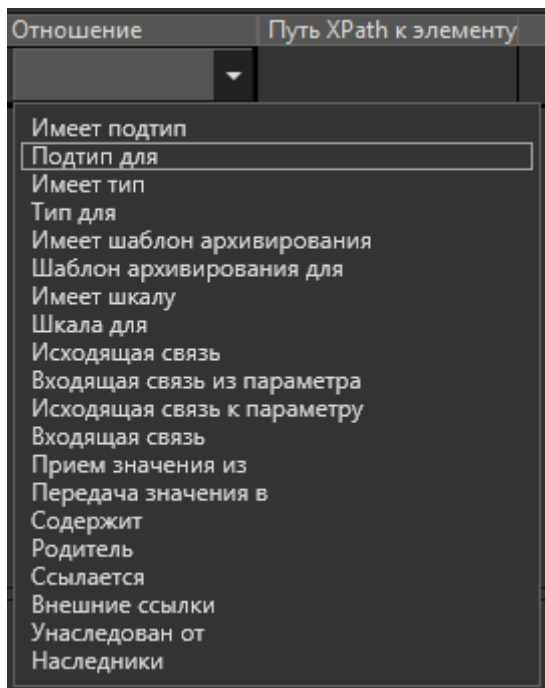
Название	Описание
	<ul style="list-style-type: none"> • != - значение свойства не равно указанной последовательности символов • >, >=, <, <= - выполняется соответствующее сравнение значения свойства с указанной последовательностью символов
Значение	Задается значение, с которым будет сравниваться значение указанного свойства

5.2.8.1.2. ТИП УСЛОВИЯ ЗАПРОСА - ОТНОШЕНИЕ

Если выбрано значение Отношение столбце Тип условия запроса, то нужно заполнить только один одноименный столбец:




В выпадающем списке доступны следующие варианты:



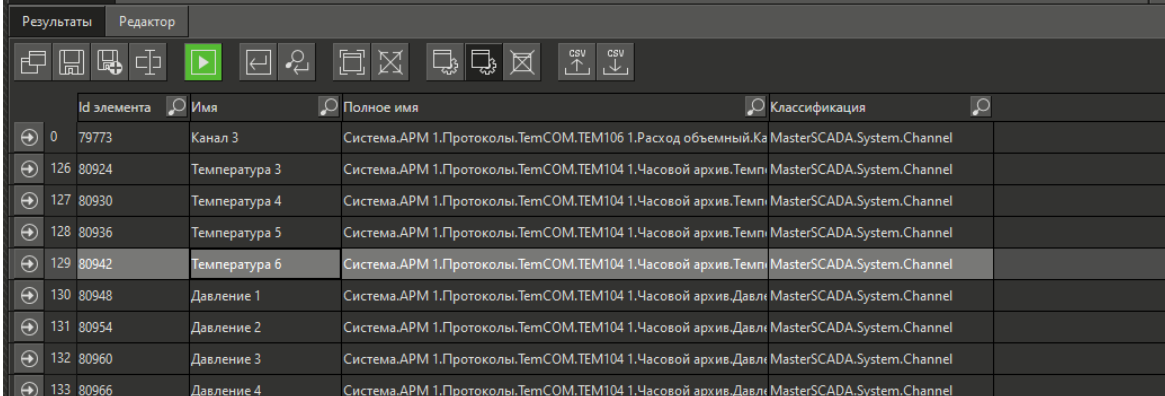
В списке отображаются те значения, которые могут указываться в категории свойств элементов Отношения.

5.2.8.2. РАБОТА С РЕЗУЛЬТАТАМИ ЗАПРОСОВ

Результаты запросов таблицы представления проекта отображаются после запуска запроса при помощи кнопки .

Отображаются результаты в нижней части вкладки Запросы и во вкладке Результаты.

Вид вкладки по умолчанию:




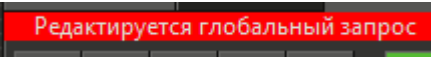








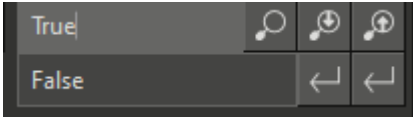





	Id элемента	Имя	Полное имя	Классификация
0	79773	Канал 3	Система.APM 1.Протоколы.ТемCOM.ТЕМ106 1.Расход объемный.Ка	MasterSCADA.System.Channel
126	80924	Температура 3	Система.APM 1.Протоколы.ТемCOM.ТЕМ104 1.Часовой архив.Темп	MasterSCADA.System.Channel
127	80930	Температура 4	Система.APM 1.Протоколы.ТемCOM.ТЕМ104 1.Часовой архив.Темп	MasterSCADA.System.Channel
128	80936	Температура 5	Система.APM 1.Протоколы.ТемCOM.ТЕМ104 1.Часовой архив.Темп	MasterSCADA.System.Channel
129	80942	Температура 6	Система.APM 1.Протоколы.ТемCOM.ТЕМ104 1.Часовой архив.Темп	MasterSCADA.System.Channel
130	80948	Давление 1	Система.APM 1.Протоколы.ТемCOM.ТЕМ104 1.Часовой архив.Давл	MasterSCADA.System.Channel
131	80954	Давление 2	Система.APM 1.Протоколы.ТемCOM.ТЕМ104 1.Часовой архив.Давл	MasterSCADA.System.Channel
132	80960	Давление 3	Система.APM 1.Протоколы.ТемCOM.ТЕМ104 1.Часовой архив.Давл	MasterSCADA.System.Channel
133	80966	Давление 4	Система.APM 1.Протоколы.ТемCOM.ТЕМ104 1.Часовой архив.Давл	MasterSCADA.System.Channel

Вкладка состоит из панели инструментов, таблицы с результатами и строки статуса.

Панель инструментов

Описание элементов:

Вид	Название	Назначение
	Дублировать запрос	Открывает вкладку нового запроса, с текущими настройками
	Сохранить	Сохраняет редактируемый запрос
	Сохранить как глобальный запрос	Добавляет редактируемый запрос в группу Глобальные. При редактировании глобального запроса появляется в верхней части вкладки Редактор надпись: 
	Переименовать запрос	Открывает диалоговое окно, в котором можно переименовать запрос.
	Запустить запрос	Запускает выполнение запроса.

Вид	Название	Назначение
	Режим множественного ввода	<p>Позволяет заменить значения свойства сразу всех выделенных элементов в таблице. Если кнопка нажата, то над столбцами свойств, которые можно изменить появится поле ввода нового значения - </p> <p>После нажатия на кнопку  введенное значение применится ко всем выделенным элементам. См. также Изменение значений свойств</p>
	Режим поиска и замены	<p>Позволяет найти в столбце одинаковые значения и изменить их. Если кнопка нажата, то над столбцами свойств появляются два поля:</p>  <p>В верхнее поле вводятся значения, которые нужно найти, а в нижнее новое значение. См. также Поиск и замена значений в таблице результатов запросов</p>
	Выбрать все	Выделяет все элементы в таблице результатов
	Отменить выбор	Снимает выделение
	Добавить пользовательский столбец	Позволяет добавить в таблицу дополнительные столбцы. См. также Настройка таблицы результатов запроса
	Удалить таблицу	Удаляет все столбцы таблицы, добавленные пользователем. См. также Настройка таблицы результатов запроса
	Режим задания закрепленных столбцов таблицы	<p>Закрепляет указанное число столбцов слева. Если кнопка отжата, то рядом появляется поле, где можно указать то количество столбцов, которое нужно закрепить. Если кнопка нажата, то поле ввода пропадает и столбцы закрепляются. Если вся таблица не помещается в рабочей области, то появится линейка прокрутки, можно будет изменить видимую область, но первые закрепленные столбцы</p>



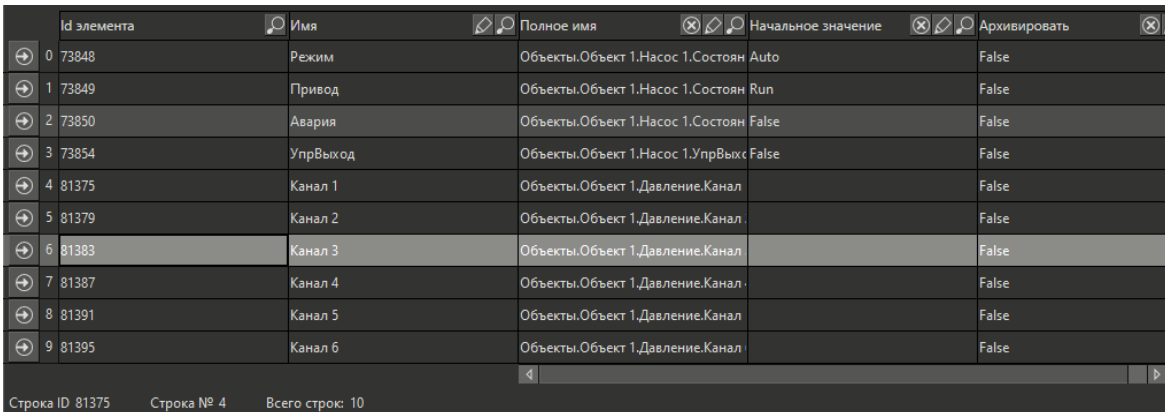
Вид	Название	Назначение
		смещаться не будут. См. также Настройка таблицы результатов запроса
	Экспорт в CSV	Открывает диалоговое окно, в котором можно указать имя файла, в который сохранится содержимое таблицы. См. также Редактирование результатов запросов в CSV
	Импорт в CSV	Открывает диалоговое окно, в котором можно указать имя файла, из которого загрузится в таблицу его содержимое. См. также Редактирование результатов запросов в CSV

Таблица с результатами


Вид таблицы:



Id элемента	Имя	Полное имя	Начальное значение	Архивировать
0 73848	Режим	Объекты.Объект 1.Насос 1.Состоян	Auto	False
1 73849	Привод	Объекты.Объект 1.Насос 1.Состоян	Run	False
2 73850	Авария	Объекты.Объект 1.Насос 1.Состоян	False	False
3 73854	УпрВыход	Объекты.Объект 1.Насос 1.УпрВых	False	False
4 81375	Канал 1	Объекты.Объект 1.Давление.Канал		False
5 81379	Канал 2	Объекты.Объект 1.Давление.Канал		False
6 81383	Канал 3	Объекты.Объект 1.Давление.Канал		False
7 81387	Канал 4	Объекты.Объект 1.Давление.Канал		False
8 81391	Канал 5	Объекты.Объект 1.Давление.Канал		False
9 81395	Канал 6	Объекты.Объект 1.Давление.Канал		False

Строка ID: 81375 Строка №: 4 Всего строк: 10

Таблица с результатами имеет следующие инструменты:

Вид	Название	Назначение
	Элементы управления столбцом - находятся в заголовке столбца. Набор инструментов столбца зависит от его расположения и назначения.	
	Развернуть дополнительную панель фильтрации	Позволяет отфильтровать элементы таблицы. Если кнопка нажата, то появляется поле ввода. В таблице будут отображаться только те элементы, в свойстве которого есть указанная последовательность символов. См. также Фильтрация и сортировка результатов запроса

■	Редактировать столбец	Открывает диалоговое окно, в котором можно указать имя столбца, тип свойства. См. также Настройка таблицы результатов запроса
■	Удалить столбец	Удаляет текущий столбец. См. также Настройка таблицы результатов запроса
■/■	Сортировка по возрастанию/Сортировка по убыванию	Индикатор сортировки столбца. Доступна сортировка только по одному столбцу. По умолчанию сортировка не включена - индикатор не отображается. Если нажать, на заголовок столбца, то появится сортировка по этому столбцу, а индикатор покажет ее направление. См. также Фильтрация и сортировка результатов запроса
Элементы управления строкой		
■	Перейти к элементу	Выделяет в дереве проекта текущий элемент.

Выделенные строки в таблице имеют более светлый фон:

	id элемента	Имя	Полное имя	Начальное значение	Архивировать
+	3	73854	УпрВыход	Объекты.Объект 1.Насос 1.УпрВыход	False
+	0	73848	Режим	Объекты.Объект 1.Насос 1.Состоян	Auto
+	1	73849	Привод	Объекты.Объект 1.Насос 1.Состоян	Run

Выделить несколько строк можно удерживая кнопками SHIFT+стрелки управления курсором вверх/вниз, а также клавишей SHIFT+ клики левой кнопкой мыши по началу и концу выделения.

Если выполнить двойной клик по ячейке, значение которой можно изменить, то появится поле ввода:

	id элемента	Имя	Полное имя	Классификация	Архивировать	
+	0	79947	Объем 1	Система.АРМ 1.Протоколы.ТемCOI	MasterSCADA.System.Channel	True
+	126	80930	Температура 34	Система.АРМ 1.Протоколы.ТемCOI	MasterSCADA.System.Channel	True

Строка статуса

В строке статуса, которая находится под таблицей, отображается ID выделенных элементов, номера их строк, а также сколько всего строк в запросе.


5.2.8.2.1. НАСТРОЙКА ТАБЛИЦЫ РЕЗУЛЬТАТОВ ЗАПРОСА

Таблица с результатами запросов по умолчанию имеет 4 столбца, в которых отображаются свойства, общие для всех элементов проекта:

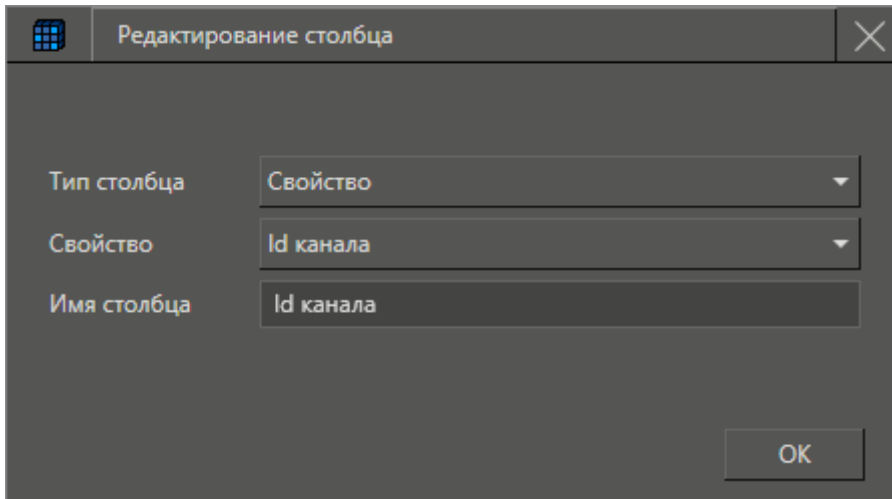
- ID-элемента
- Имя
- Полное имя
- Классификация

Эти столбцы нельзя удалить.

Добавление столбцов

Чтобы добавить столбец нужно нажать на кнопку 


Появится диалоговое окно:




Название	Описание
Тип столбца	<p>Позволяет выбрать, какого типа информация будет отображаться в новом столбце. В зависимости от выбранного типа будет зависеть имя следующего поля. Возможные варианты:</p> <ul style="list-style-type: none"> • Свойство • Отношение • XPath
Свойство/Отношение/XPath	<p>В зависимости от типа столбца будет зависеть имя данного поля.</p> <p>Свойство - в выпадающем списке будут отображаться свойства, которые характерны для выбранного Типа элемента в основных</p>

	<p>параметрах запроса. Так например для элементов Окно и Параметр список свойств будет отличаться.</p> <p>Отношение - в выпадающем списке отображаются те значения, которые могут указываться в категории свойств элементов Отношения.</p> <p>XPath - нужно задать XPath к элементу. В этом случае будет в столбце отображаться начальное значение найденного элемента. Например, если в таблице отображаются объекты, у которых имеется одинаковый параметр и нужно в столбце вывести его начальное значение, то в поле нужно ввести имя параметра.</p>
Имя столбца	Задается имя столбца, которое будет отображаться в таблице.

После нажатия на кнопку ОК столбец будет сформирован правее от уже существующих столбцов. Перезапускать формирование запроса не требуется.

Отредактировать добавленный столбец можно нажав на кнопку  в его заголовке. Откроется диалоговое окно как и при его создании.

Закрепление столбцов

Можно закрепить первые левые столбцы таблицы при помощи кнопки  панели инструментов. Если кнопка отжата, то столбцы не закреплены, рядом появляется поле, где можно указать, то количество столбцов, которое нужно закрепить:




Если кнопка нажата, то поле ввода пропадает и столбцы закрепляются. Если вся таблица не помещается в рабочей области, то появится линейка прокрутки, можно будет изменить видимую область, но первые закрепленные столбцы смещаться не будут.


Порядок отображения столбцов

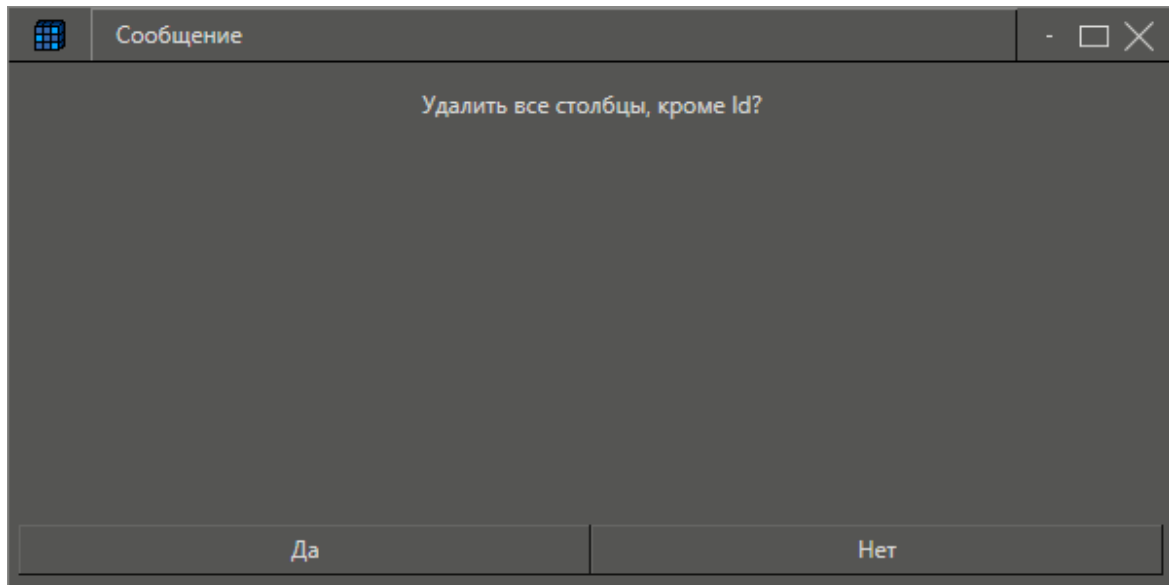
Порядок отображения столбцов таблицы запросов можно изменить. Для этого нужно захватить заголовок столбца левой кнопкой мыши и переместить в нужное место таблицы.

Закрепленные столбцы могут изменить свое место, но только среди закрепленных столбцов. Незакрепленные столбцы могут изменить свое место, но только среди незакрепленных столбцов. Чтобы переместить столбец в закрепленную область нужно сначала отменить закрепление (см. выше) переместить столбец, а затем закрепить столбцы заново.

Удаление столбцов

Удалить столбец, который был добавлен пользователем можно при помощи кнопки , которая находится в заголовке столбца. После нажатия появится диалоговое окно. Если нажать в нем кнопку Да, то столбец будет удален.

Удалить все добавленные пользователем столбцы таблицы можно при помощи кнопки  панели инструментов. Появится диалоговое окно:




Все столбцы пользователя будут удалены после нажатия на кнопку Да.


5.2.8.2.2. ФИЛЬТРАЦИЯ И СОРТИРОВКА РЕЗУЛЬТАТОВ ЗАПРОСА







Таблица с результатами запросов позволяет фильтровать и сортировать элементы.

Сортировка результатов

Доступна сортировка только по одному столбцу. По умолчанию сортировка не включена - индикатор не отображается. Если нажать, на заголовок столбца, то появится сортировка по этому столбцу, а индикатор  покажет ее направление.

Фильтрация результатов

Кнопка  в заголовке столбца позволяет отфильтровать элементы таблицы. Если кнопка нажата, то появляется поле ввода. В таблице будут отображаться только те элементы, в свойстве которого есть указанная последовательность символов:

	Полное имя 	Id элемента 	Имя 	Классификация 
			На	
	1 Объекты.Объект 1.Насос 1	73841	Насос 1	MasterSCADA.Object
	2 Объекты.Объект 1.Насос 2	86833	Насос 2	MasterSCADA.Object

5.2.8.2.3. ИЗМЕНЕНИЕ ЗНАЧЕНИЙ СВОЙСТВ В ТАБЛИЦЕ РЕЗУЛЬТАТОВ ЗАПРОСОВ

Рассмотрим изменение свойств элементов при помощи табличного вида проекта.

После того как таблица с результатами сформирована можно изменить значение как отдельного элемента, так и группы.

Изменение значения свойства одного элемента

Если дважды нажать ЛК мыши в ячейке свойства значение которого можно изменить, то появится поле ввода:


Id элемента	Имя	Полное имя	Классификация	Архивировать
0	79947	Объем 1	Система.АРМ 1.Протоколы.ТемCOI MasterSCADA.System.Channel	True
126	80930	Температура 34	Система.АРМ 1.Протоколы.ТемCOI MasterSCADA.System.Channel	True

Важно! Свойства Id элемента, Полное имя, Классификация изменить нельзя. При двойном нажатии поле ввода появляться не будет.


Если столбцы были добавлены пользователем, но изменить значения этих свойств нельзя, то поле будет иметь вид:

Id элемента	Имя	Полное имя	Классификация	Архивировать	Количество подэлементов
0	79947	Объем 1	Система.АРМ 1.Прот MasterSCADA.Syst True	True	0
1	79953	Объем 2	Система.АРМ 1.Прот MasterSCADA.Syst True	True	0

Изменение значений свойств нескольких элементов


Выделить несколько строк можно удерживая кнопками SHIFT+стрелки управления курсором вверх/вниз, а также клавишей SHIFT+ клики левой кнопкой мыши по началу и концу выделения. Выделенные строки в таблице имеют более светлый фон. Чтобы изменить у нескольких элементов одно и тоже свойство нужно нажать на кнопку  панели инструментов таблицы результатов запроса - включится режим множественного ввода. Над столбцами, которые можно изменить появится поле ввода:

Id элемента	Имя	Полное имя	Классификация	Архивировать
0	79674	Время	Система.АРМ 1.Протоколы.ТемCOI MasterSCADA.System.Channel	False
1	79680	Серийный номер	Система.АРМ 1.Протоколы.ТемCOI MasterSCADA.System.Channel	False
2	79687	Канал 1	Система.АРМ 1.Протоколы.ТемCOI MasterSCADA.System.Channel	False
3	79693	Канал 2	Система.АРМ 1.Протоколы.ТемCOI MasterSCADA.System.Channel	False

После нажатия на кнопку  введенное значение применится ко всем выделенным элементам.

5.2.8.2.4. ПОИСК И ЗАМЕНА ЗНАЧЕНИЙ В ТАБЛИЦЕ РЕЗУЛЬТАТОВ ЗАПРОСОВ


После того как таблица с результатами сформирована в ней можно найти и изменить значение как отдельного элемента, так и нескольких.


Режим поиска и замены включается кнопкой . Если кнопка нажата, то над столбцами свойств, которые можно изменить появляются два поля:




В верхнее поле вводятся значения (последовательность символов), которые нужно найти, а в нижнее на что нужно заменить найденное.

Если значение свойства, которое находится в столбце изменить нельзя, то появится только верхнее поле.


Поиск элементов происходит после нажатия на кнопку .

Кнопки  переводят курсор на следующий найденный результат или на предыдущий соответственно.

Кнопки  производят замену найденных символов на новые. Если нажать левую кнопку, то изменится только одно значение, на котором установлен курсор. Если нажать на правую, то все найденные значения.

5.2.8.2.5. РЕДАКТИРОВАНИЕ РЕЗУЛЬТАТОВ ЗАПРОСОВ В CSV

Значения свойств элементов, которые попали в таблицу с результатами запросов можно изменить во внешнем CSV -файле.

Для этого нужно сначала сохранить результаты запроса, нажав на кнопку  Экспорт в CSV в панели инструментов таблицы.

Откроется диалоговое окно Windows, в котором можно указать место сохранения файла.

Затем файл можно открыть и заменить нужные значения свойств. Например, адреса каналов протоколов, задать имена параметров и т.п. Сохранить файл в формате CSV (разделители запятые). Затем нужно выполнить импорт измененного файла, при помощи

кнопки . Измененные значения автоматически будут присвоены нужным свойствам элементов.









Важно! В файле нельзя редактировать столбец Property~Id элемента~Id

5.2.9. ПАНЕЛЬ УПРАВЛЕНИЯ ИНТЕРФЕЙСОМ

В зависимости от состояния кнопок (нажаты или отжаты) меняется состав отображаемых панелей среды разработки.

Панель управления интерфейсом имеет следующий вид:





Вид	Назначение
	Если кнопка нажата, то рабочая область занимает все рабочее пространство среды разработки MasterSCADA 4D (при этом верхняя часть панели инструментов и заголовок окна остаются доступны).
	Управляет видимостью верхней части Панели инструментов.
	Управляет видимостью Древа.
	Управляет видимостью Контекстной панели.
	Управляет видимостью Панели веток.
	Управляет видимостью Легенды.
	Управляет видимостью Миникарты.
	Управляет видимостью Панели свойств.

5.2.10. ПАНЕЛЬ ИНСТРУМЕНТОВ СПРАВКА


Панель инструментов Справка имеет вид:



Назначение кнопок:

Вид	Описание
	Предназначена для вызова полной справки
	Включает/выключает режим контекстной справки "Что это?" (What's This?).

Работа с контекстной справкой

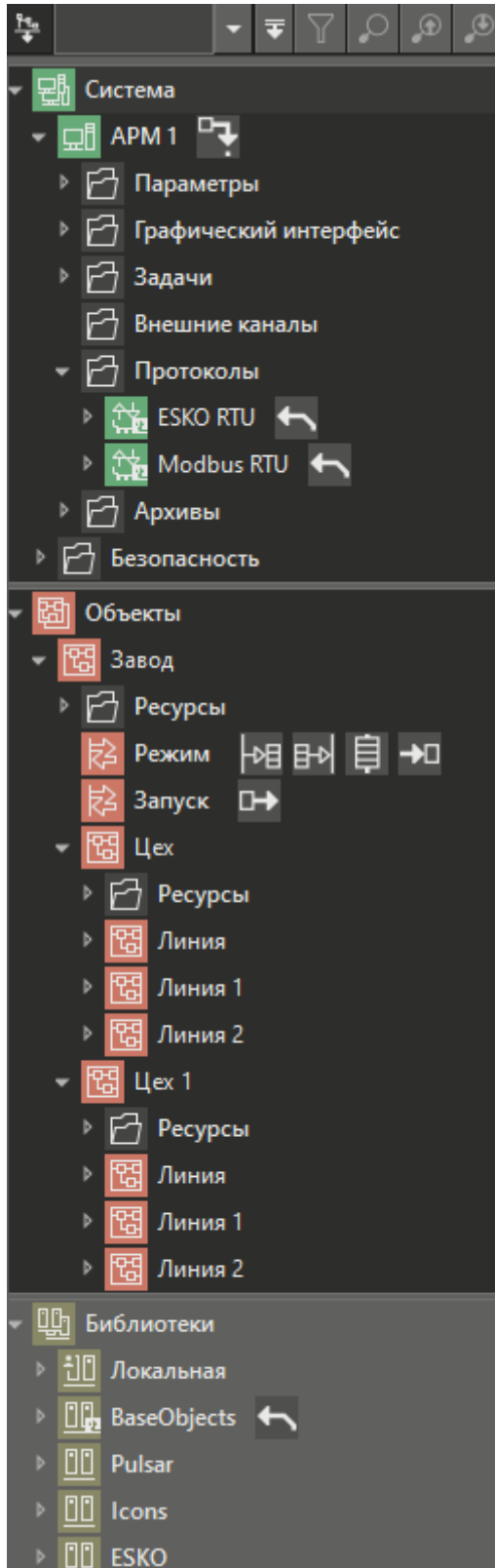
Если режим контекстной справки включен, то курсор имеет вид , и нажатие ЛК на элементе редактора приводит к открытию меню выбора контекстной справки.

Если для элемента возможно выделение, то для вызова контекстной справки можно также выделить элемент и нажать клавишу F1. Для вызова контекстной справки по инструменту меню редактора проекта необходимо навести курсор на инструмент и нажать клавишу F1.

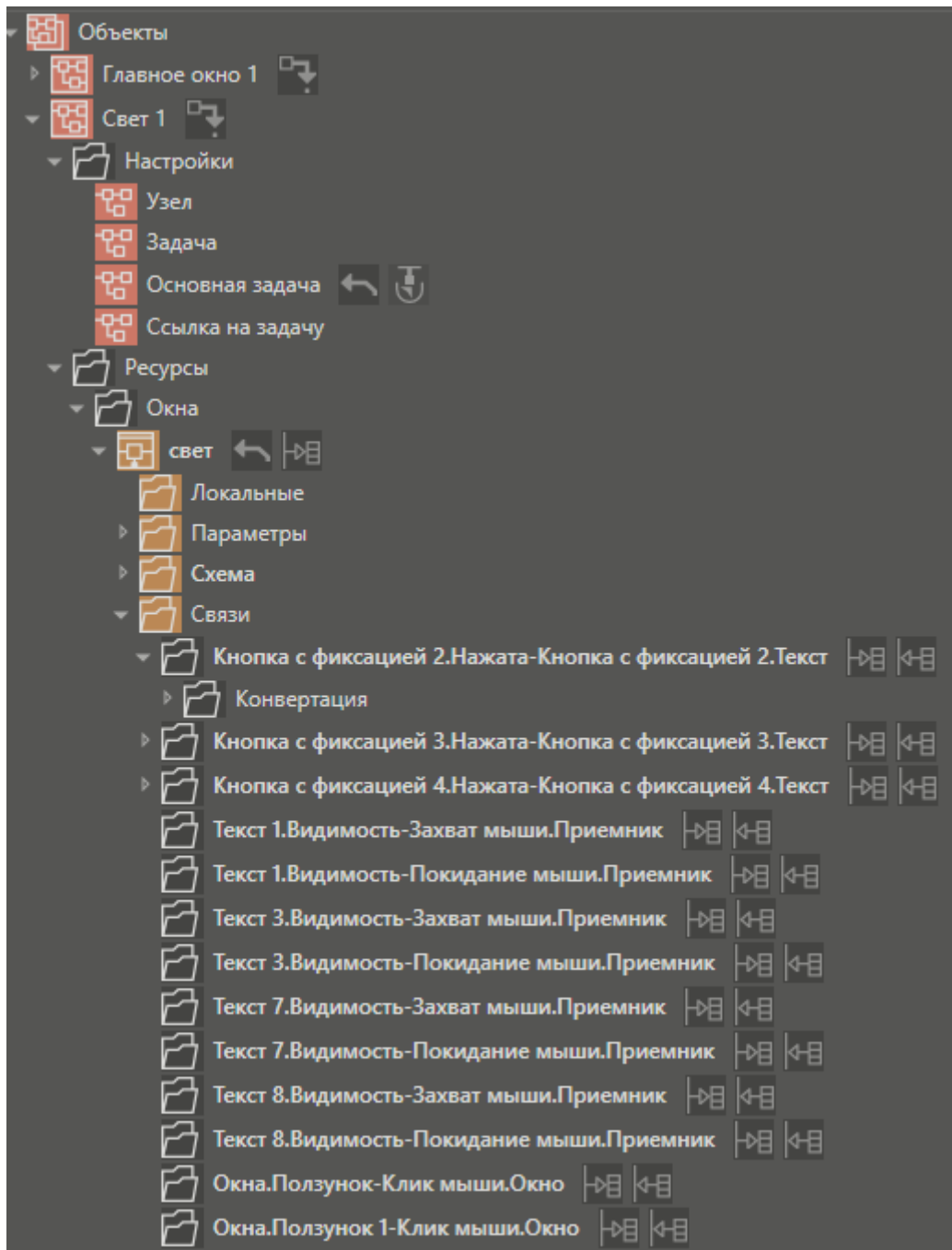
Кроме того, в контекстном меню любого элемента есть пункт Справка, в котором можно выбрать раздел справочной системы, относящийся к выделенному элементу.

5.3. ДЕРЕВО MASTERSCADA

Дерево MasterSCADA занимает левую часть интерфейса редактора. Это основной инструмент работы над проектом. Оно разделено на три основные части – Дерево системы, Дерево объектов и Дерево библиотек:



Все элементы (узлы, объекты, программы, параметры, связи, окна, элементы окон и программ и др.), которые создаются в проекте, имеют свое место в Дереве.





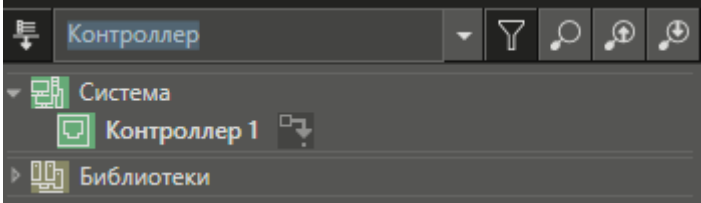





Панель инструментов дерева

В верхней части Дерева находится Панель инструментов дерева:



Элементы Панели инструментов:

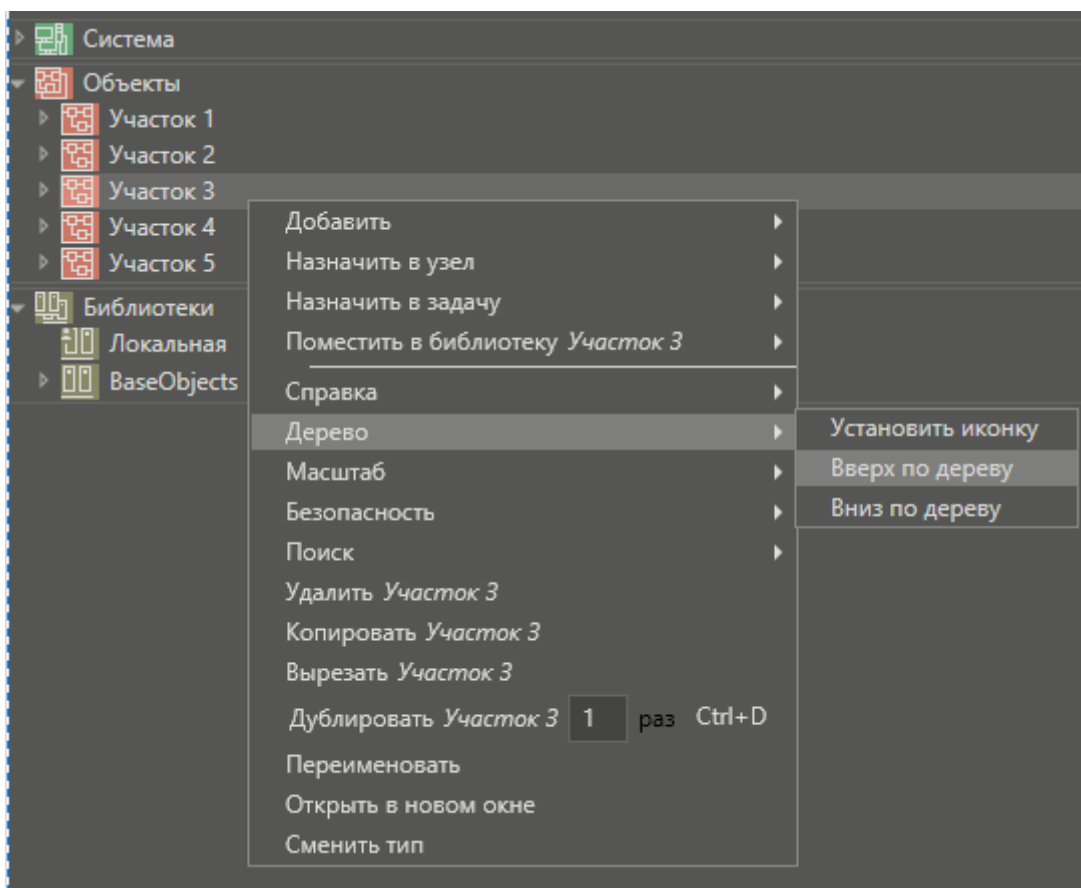
Вид	Описание
	<p>Кнопка управления видимостью элементов в Дереве. Если кнопка нажата, то Деревья отображаются в упрощенном виде. Это означает, что в них присутствуют только те элементы, которые разработчик проекта создает явно, либо те элементы программ и окон, которые в дальнейшем будет логично связать с другими элементами проекта. В библиотеках проекта отображаются только Библиотеки пользователя и BaseObjects, т.к. все другие элементы добавляются в проект либо при помощи контекстного меню (контекстной панели), либо при помощи палитры редакторов. Если кнопка отжата, то Деревья отображаются в полном виде, т.е. в них видны все элементы, связи и настройки.</p>
	<p>В данное поле можно ввести символы (например, часть имени параметра, или объекта, либо ID элемента), либо выбрать ранее введенные символы. Введенные символы могут быть использованы для фильтрации дерева и для поиска элементов в нем.</p>
	<p>Кнопка открывает выпадающий список, в котором флагами нужно указать места поиска элементов.</p>
	<p>Включает фильтр Деревя. Оставляет видимыми только те элементы, которые в своих настройках имеют символы, находящиеся в текстовом поле:</p> 
	<p>Выполняет поиск по всему Дереву. Выдает все элементы, значения свойств которых содержат символы, введенные в текстовом поле. См. Поиск элементов</p>
	<p>Выполняет поиск ближайшего элемента сверху от выделенного в Дереве, свойства которого содержат символы, введенные в текстовом поле. Курсор перемещается на найденный элемент.</p>
	<p>Выполняет поиск ближайшего элемента снизу от выделенного в Дереве, свойства которого содержат символы, введенные в текстовом поле. Курсор перемещается на найденный элемент.</p>

Работа в Дереве

Изменение положения элемента

Важно! Если элементы исполняются в одной и той же задаче узла, то первым исполнится тот элемент, который находится выше в Дереве MasterSCADA.

Для того, чтобы изменить положение элемента в Дереве, его нужно либо перетащить левой кнопкой мыши на новое место, либо нажать клавиши клавиатуры CTRL+ ↓, CTRL+ ↑, либо воспользоваться пунктами контекстного меню элемента: Дерево-Вверх по Дереву, Дерево-Вниз по Дереву.



Работа с иконками

Для того, чтобы разработчику проекта было проще ориентироваться в масштабном проекте, для различных объектов можно задать разные иконки отображения в Дереве.

Для этого в контекстном меню нужно выполнить пункт Дерево-Замена иконки. По этой команде для каждого элемента из выделенной группы последовательно открывается стандартный диалог выбора файла.

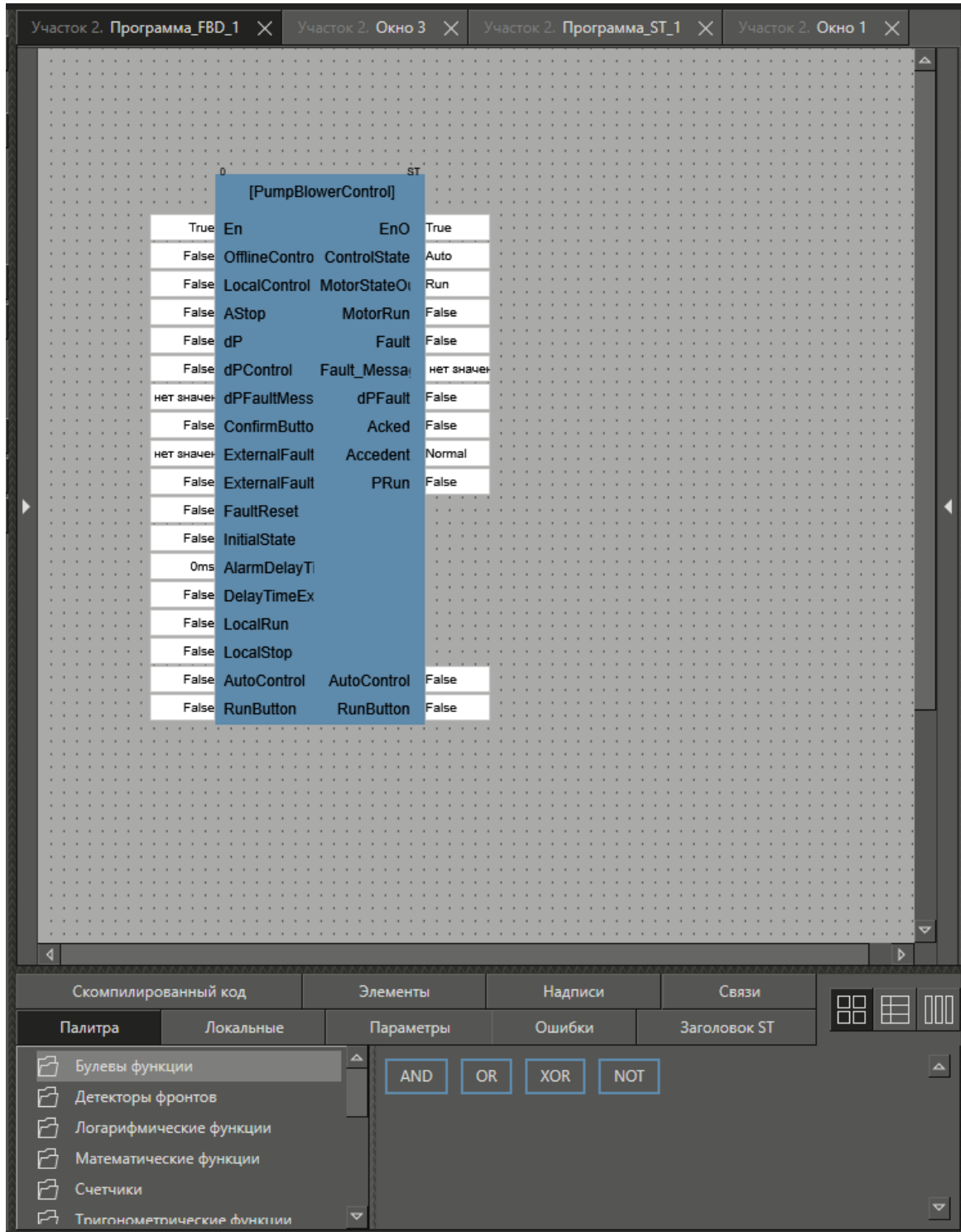
При помощи команды контекстного меню Дерево-Удалить Иконку восстанавливаются иконки по умолчанию для одного элемента или выделенной группы элементов.

Ветки дерева


В случае необходимости, можно открыть ту или иную часть Дерева (ветку) в Панели веток. Для этого в контекстном меню элемента нужно выполнить пункт Открыть в панели веток.

5.4. ПАНЕЛЬ РЕДАКТОРОВ

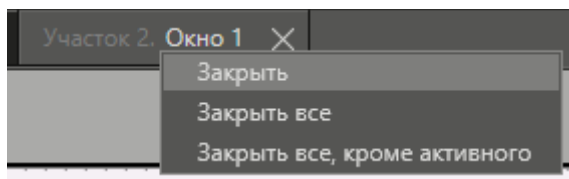
Редакторы программ и окон открываются в центральной части интерфейса как вкладки:



В зависимости от выбранного редактора, в нижней части MasterSCADA 4D открывается соответствующая ему Легенда.

При работе с проектом обращайтесь внимание на заголовки вкладок! В них отображается имя программы/окна/документа, с которым идет работа, а также объект, которому они принадлежат. Если открывается редактор элемента, который нельзя редактировать, то заголовок будет начинаться с символа .

Заголовок вкладки снабжен контекстным меню:

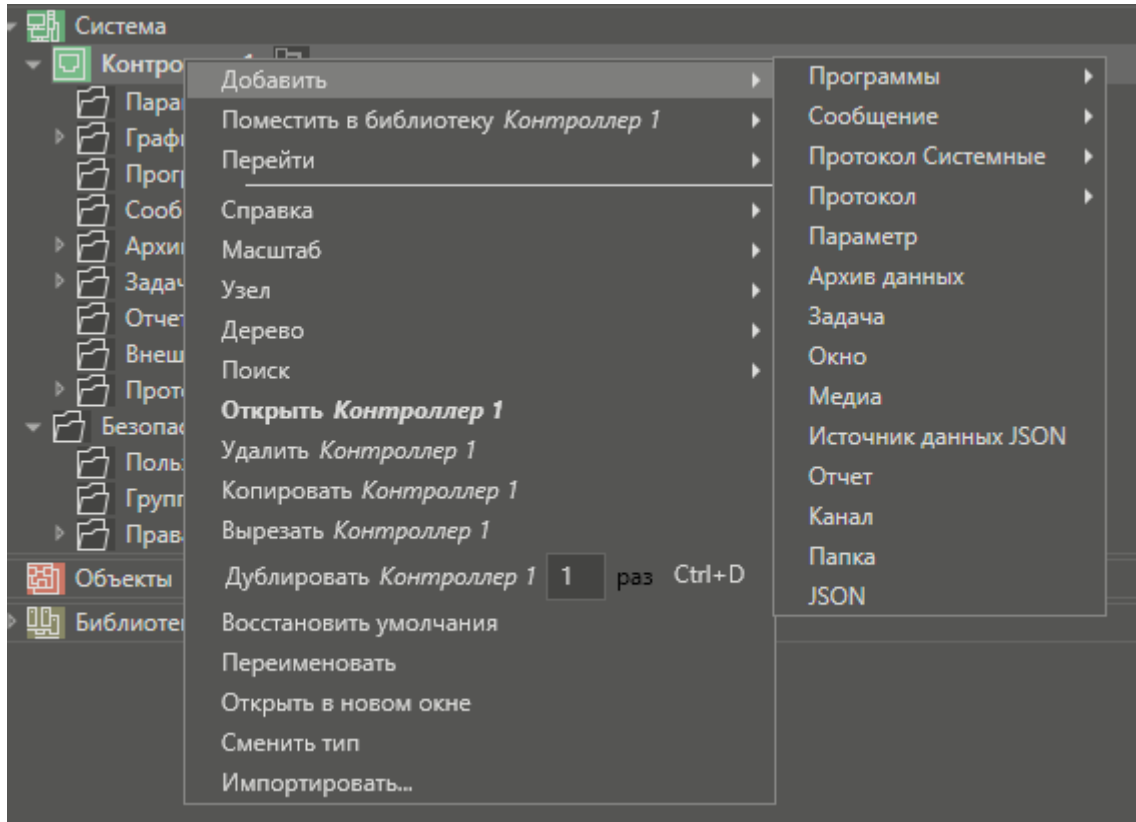


5.5. КОНТЕКСТНОЕ МЕНЮ

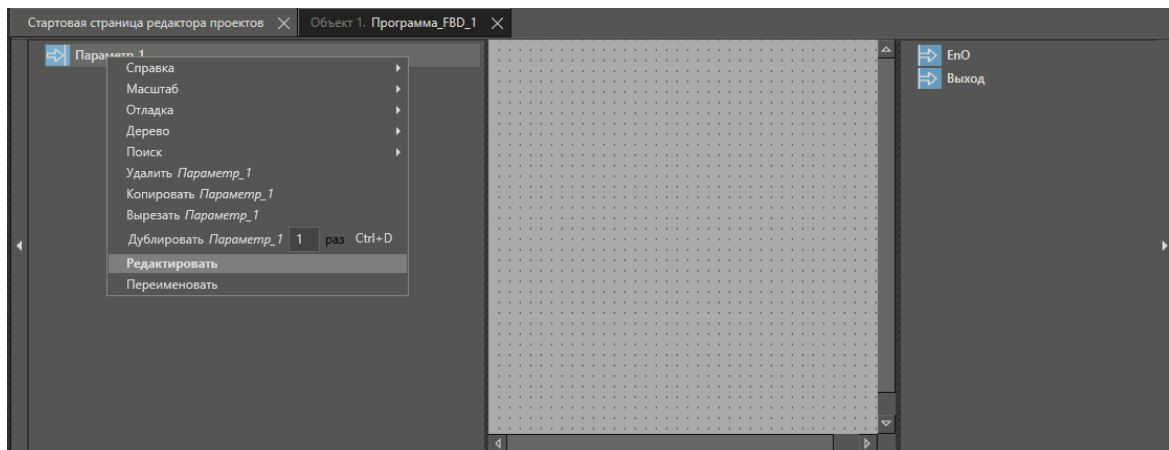
Контекстное меню – это инструмент программы, представляющий собой список возможных команд для выбранного элемента.

Контекстное меню вызывается нажатием правой кнопки мыши на элемент. В зависимости от типа элемента контекстное меню содержит разные команды. Каждый элемент проекта, независимо от места где с ним работают (Дерево системы, Дерево объектов, Легенда или Рабочая область редакторов), имеет контекстное меню.

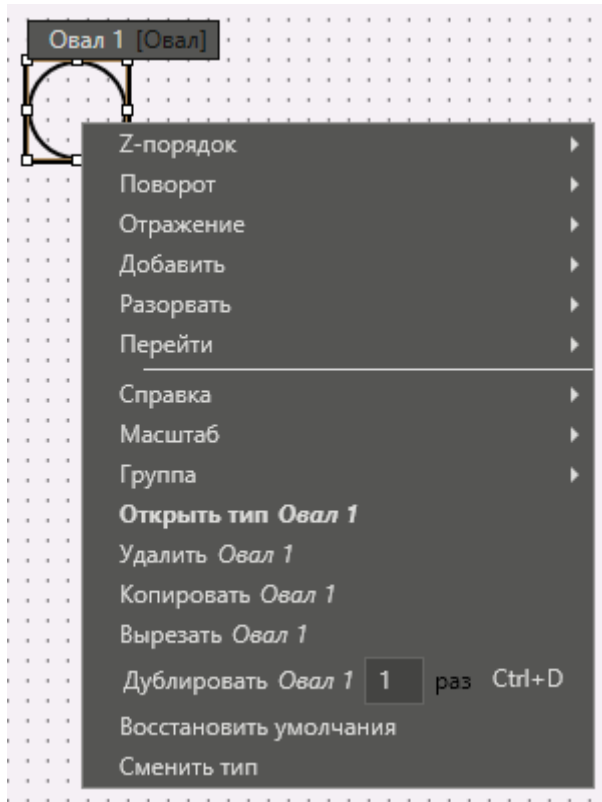
Контекстное меню узла имеет вид:



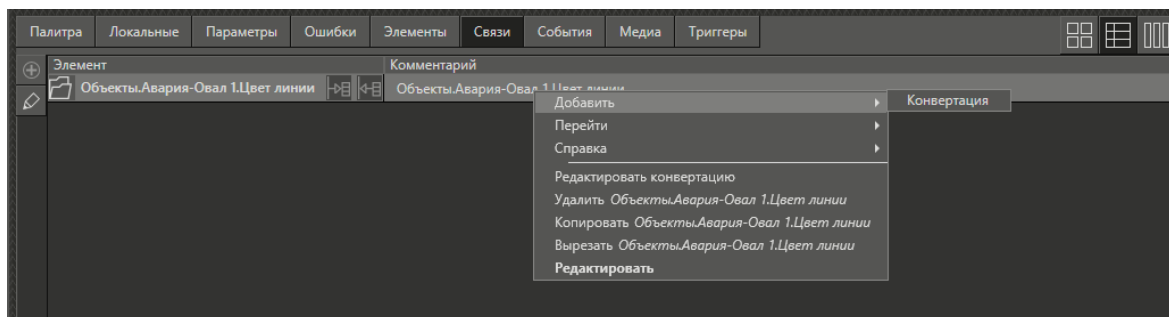
Контекстное меню клеммника программы FBD имеет вид:



Контекстное меню графического элемента имеет вид:



Контекстное меню связи в Легенде графического редактора имеет вид:

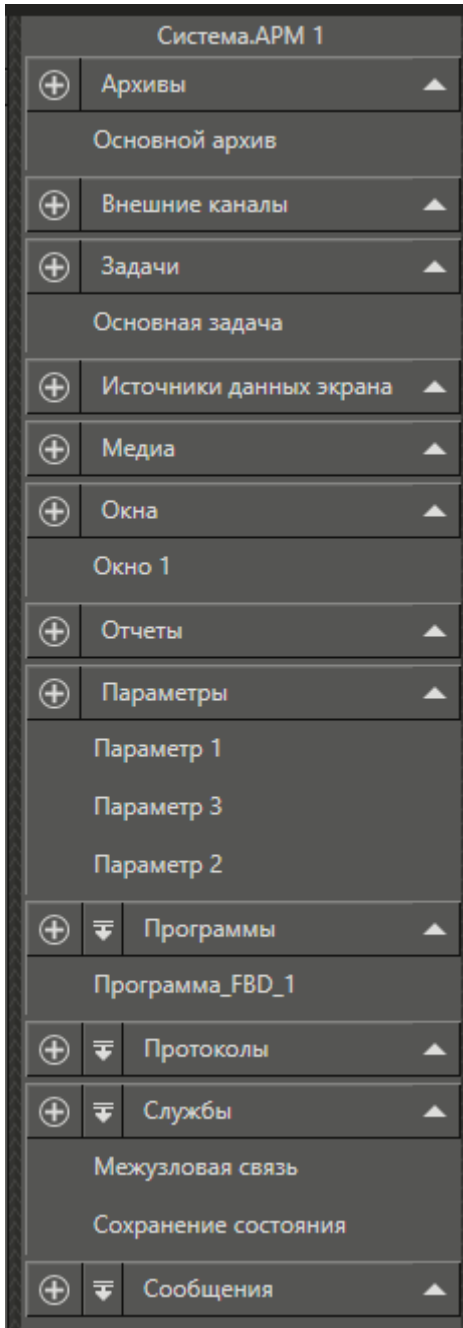


5.6. КОНТЕКСТНАЯ ПАНЕЛЬ

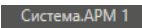

Контекстная панель находится справа от дерева MasterSCADA и обеспечивает альтернативный способ добавления элементов в проект. Контекстная панель открывается при выделении элемента в дереве. Если элемент не имеет собственной контекстной панели, то откроется контекстная панель родительского элемента.


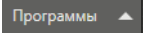



Собственную контекстную панель имеют следующие элементы: дерево системы, дерево объектов, дерево библиотек, Узел, Объект, Библиотека.

Вид контекстной панели для Узла:



Элементы контекстной панели:

Вид	Назначение
	В заголовке отображается элемент, к которому относится панель.
	Если нажать на эту кнопку, то добавится тот элемент, рядом с которым находится эта кнопка .

Вид	Назначение
	Если имеется возможность добавить несколько разных элементов, то при нажатии на эту кнопку откроется список доступных элементов.
	При нажатии на эту кнопку скрывается/раскрывается список элементов, уже добавленных в проект данного типа.
	Переход к элементу в дереве. При этом открывается панель свойств этого элемента.
	Служит для копирования элемента.
	Служит для открытия редактора, необходимого для разработки выделенного в контекстной панели элемента.

5.7. ПАНЕЛЬ СВОЙСТВ


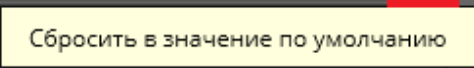
По умолчанию в этой панели отображаются/задаются свойства выделенного элемента проекта (в том числе свойства, которые не отображаются в полном дереве):

АРМ 1	
Общие	
Имя	АРМ 1
Программное имя	АРМ 1
Полное имя	Система.АРМ 1
Метки	
Комментарий	
Служебные	
Задачи	
Конфигурация	
Наличие визуализации	<input checked="" type="checkbox"/>
Параметры запуска RT	
Отношения	
Унаследован от	Библиотеки.Стандартная.Узлы.АРМ
Внешние ссылки	Объекты.Объект 1.Настройки.Узел
Связь	
IP адрес	127.0.0.1
Номер экземпляра MPLC	0
Тайм-аут (мс)	3000
Порт для Modbus TCP	502
Максимальный размер UDP пакета	0
Порт для Web сервера	8043
Доступ по OPC UA	<input type="checkbox"/>
Автозапуск исполнительной системы	<input checked="" type="checkbox"/>
Последовательность байт в Modbus	_3_2_1_0_7_6_5_4
Запрет автозагрузки проекта	<input type="checkbox"/>
Логин для OPC UA	
Пароль для OPC UA	
Порт для OPC UA	16550

У свойств, доступных только для просмотра, шрифт погашен:

Полное имя	Система.АРМ 1
------------	---------------

Переопределенные значения свойств (относительно заданных в типе или относительно заданных по умолчанию) выделяются жирным шрифтом. Чтобы вернуть умолчания, необходимо нажать на кнопку справа в строке свойства:


Имя	Параметр 1
Программное имя	Параметр 1
Полное имя	Объекты.Объект 1.Параметр 1
Метки	
Комментарий	
Начальное значение	0 
Тип значения	INT 

Свойство по умолчанию графических элементов отображается жирным шрифтом.

Установленный (отмеченный) флаг для свойства с типом данных BOOL соответствует значению TRUE, сброшенный (снятый) – значению False.

Значение свойства будет изменено, если после ввода нового значения была нажата клавиша Enter, либо курсор был переведен в любое другое место среды разработки, даже если при этом открылась панель свойств другого элемента.

Для открытия диалогового окна ввода значения необходимо нажать на кнопку:

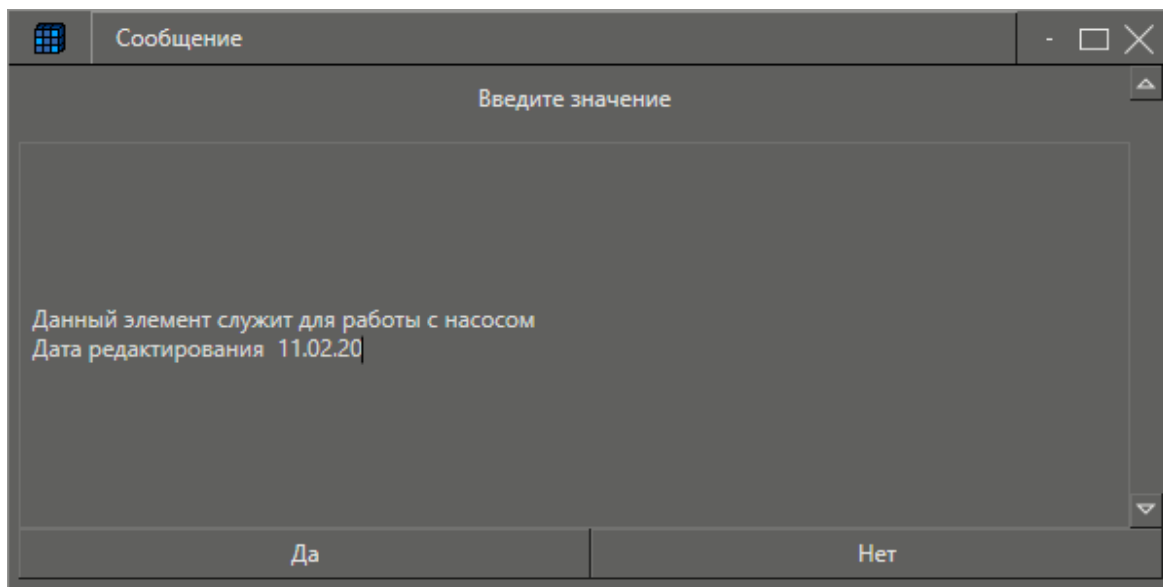
Общие	
Имя	Параметр 1
Метки	
Комментарий	
Начальное значение	0
Тип значения	LREAL
Сохранять	Наследуется
Доступ	Чтение/Запись
Отображать элементы в дереве	<input type="checkbox"/>

В зависимости свойства могут открыться следующие окна:

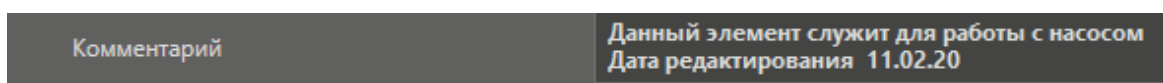
- Окно настройки параметра
- Диалоговое окно Выбор цвета

- Диалоговое окно Выбор Медиа



Для ввода обычных текстовых свойств откроется окно, в котором удобно вводить многострочный текст:




После нажатия на кнопку Да текст в панели свойств отобразится следующим образом:







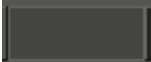


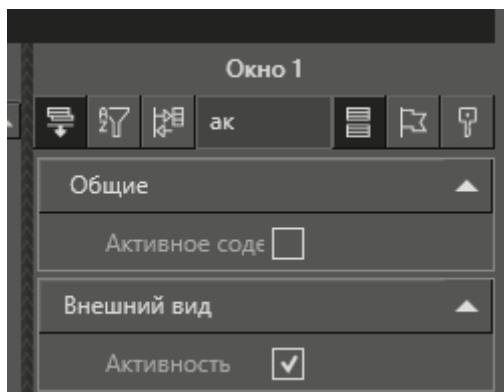
Инструменты панели:

-  – если эта кнопка отжата, то окно имеет полный набор свойств (показан на рисунке выше). В противном случае, окно имеет упрощенный набор свойств, когда показываются только те свойства, которые изменяются чаще остальных;
-  – если эта кнопка отжата, то свойства отображаются разбитыми на категории. Набор категорий для разных элементов разный. Общими для всех элементов являются категории Общие и Служебные (служебные свойства отображаются только в том случае, если в редакторе общих настроек установлен флаг Служебные свойства). Для пользовательских свойств некоторых элементов (например, для пользовательских свойств графического элемента) может быть задана категория. Если кнопка нажата, то свойства отображаются единым списком по алфавиту (без разбиения на категории).


Важно! Изображения Панели свойств в справочной системе соответствуют отжатому состоянию кнопок.

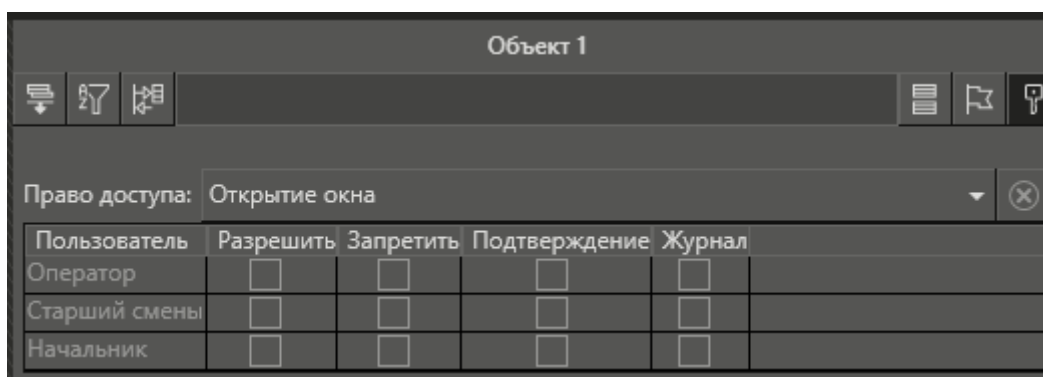
-  – фильтр свойств графического окна или элемента по наличию связи. Если эта кнопка нажата, то отображаются только те свойства, которые задействованы в связях.

-  ,  и  – набор кнопок, находящихся в верхней правой части панели свойств. Если нажата кнопка  , то отображаются свойства (см. рис. выше); если нажата кнопка  , то отображаются события (графического окна или элемента – см. Панель свойств. События); если нажата кнопка  , то отображаются права пользователей (см. Панель свойств. Права);
-  – поиск свойства по подстроке:




5.7.1. ПАНЕЛЬ СВОЙСТВ. ПРАВА

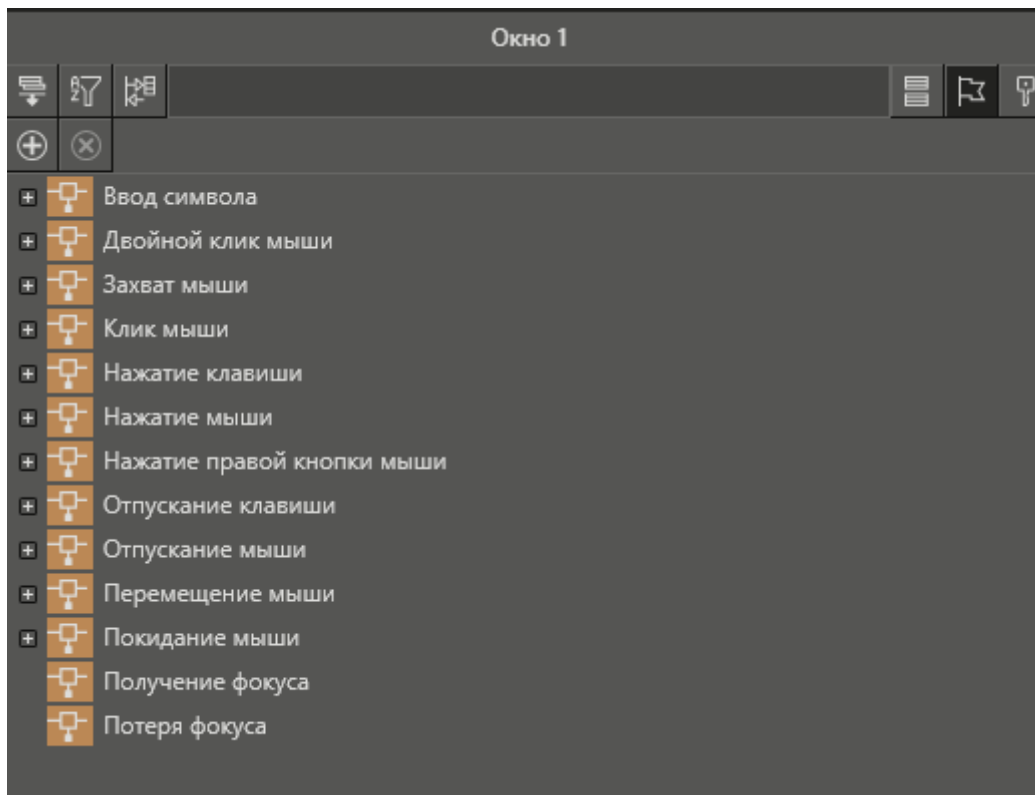
Если нажата кнопка  , то Панель свойств позволяет задать права пользователей проекта для выделенного элемента проекта (см. Задание прав пользователей).




В данном случае кнопка  и инструменты поиска и фильтра недоступны.

5.7.2. ПАНЕЛЬ СВОЙСТВ. СОБЫТИЯ

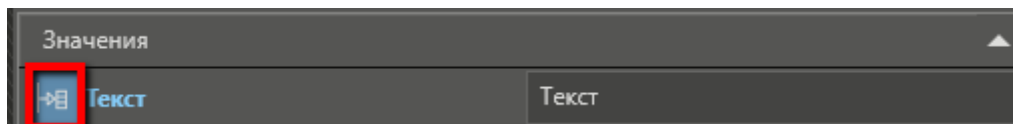
Если нажата кнопка , то Панель свойств представляет собой редактор событий выделенного графического окна или графического элемента – см. HMI. События :






В данном случае кнопка  используется для отображения событий по алфавиту по убыванию/возрастанию, а инструменты поиска и фильтра недоступны.

5.7.3. ПАНЕЛЬ СВОЙСТВ. ОТОБРАЖЕНИЕ СВОЙСТВ, ЗАДЕЙСТВОВАННЫХ В СВЯЗЯХ

Слева от свойства, задействованного в связях, отображается кнопка (на рисунке ниже выделена красной рамкой):



Внешний вид кнопки зависит от направления связи:




-  – у свойства есть только входящие связи;
-  – у свойства есть только исходящие связи;
-  – у свойства есть входящие и исходящие связи.

Связи отображаются в подсказке, всплывающей к переключателю.

Если нажать на кнопку справа от свойства, то раскрывается дополнительная информация:

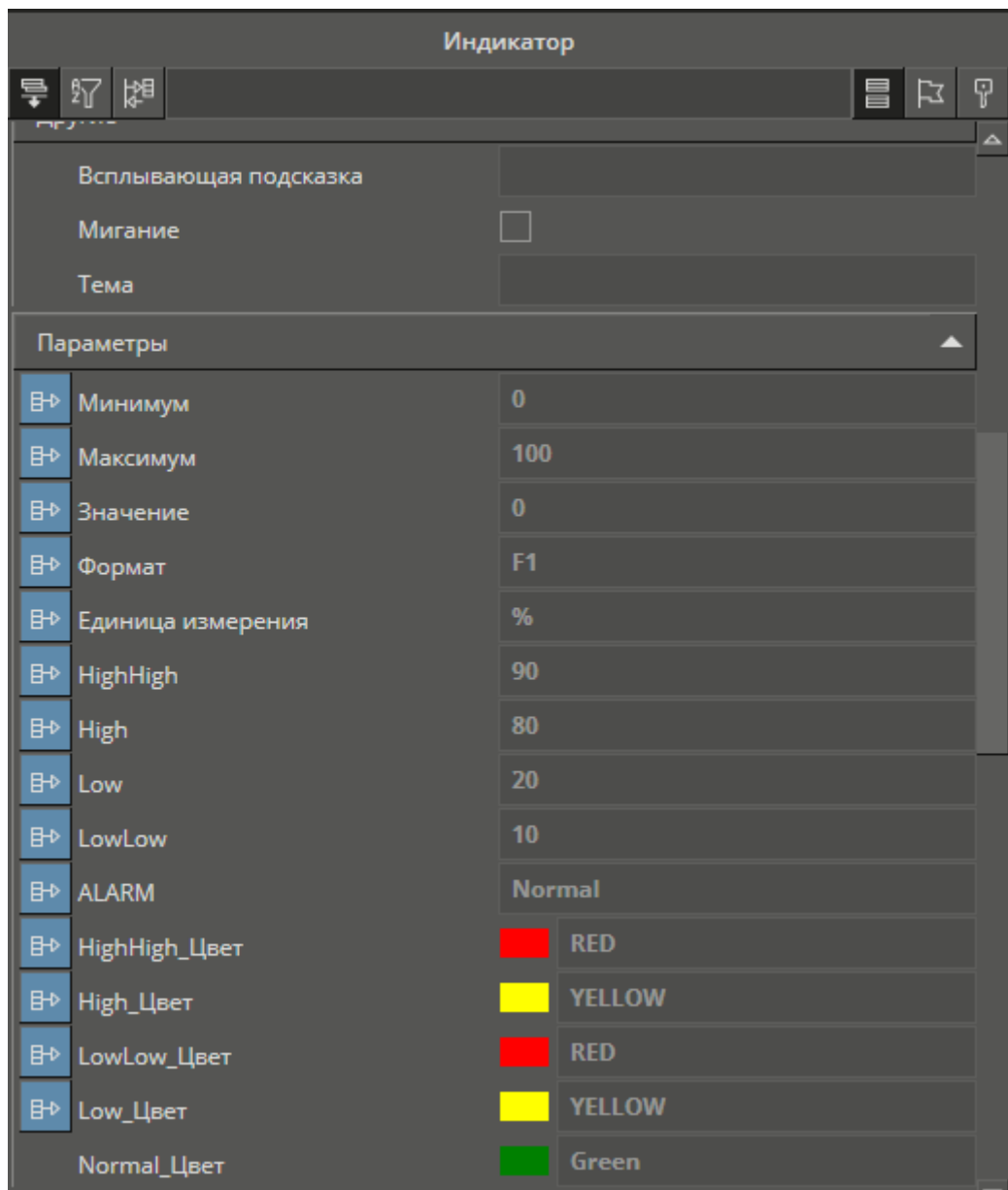


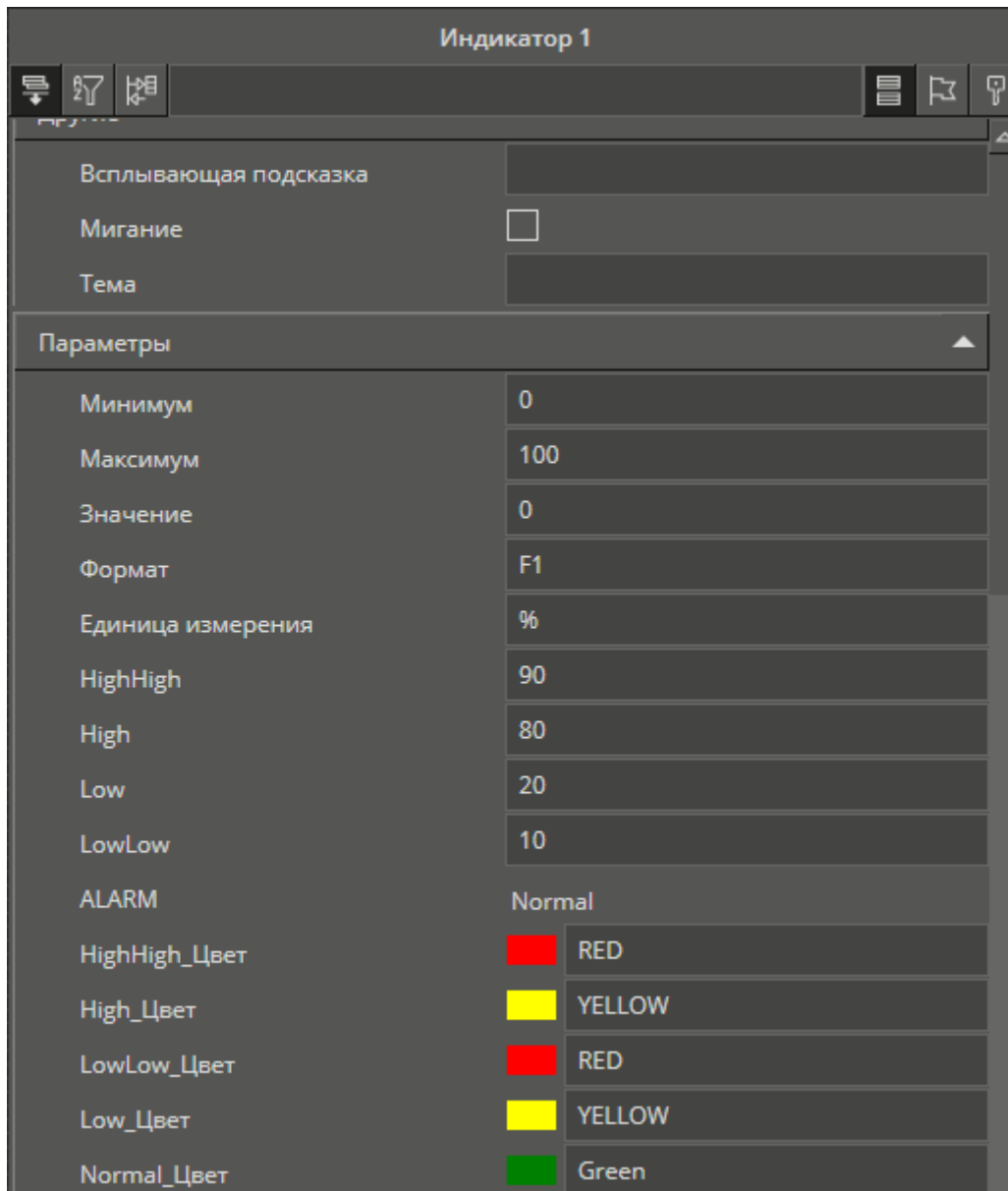
Инструменты строки описания связи:

-  – разорвать связь;
-  - перейти к связанному элементу в дереве;
-  – кнопка, по которой открывается конвертер значений .

5.7.4. ПАНЕЛЬ СВОЙСТВ. ОТОБРАЖЕНИЕ СВЯЗЕЙ БИБЛИОТЕЧНОГО ЭЛЕМЕНТА

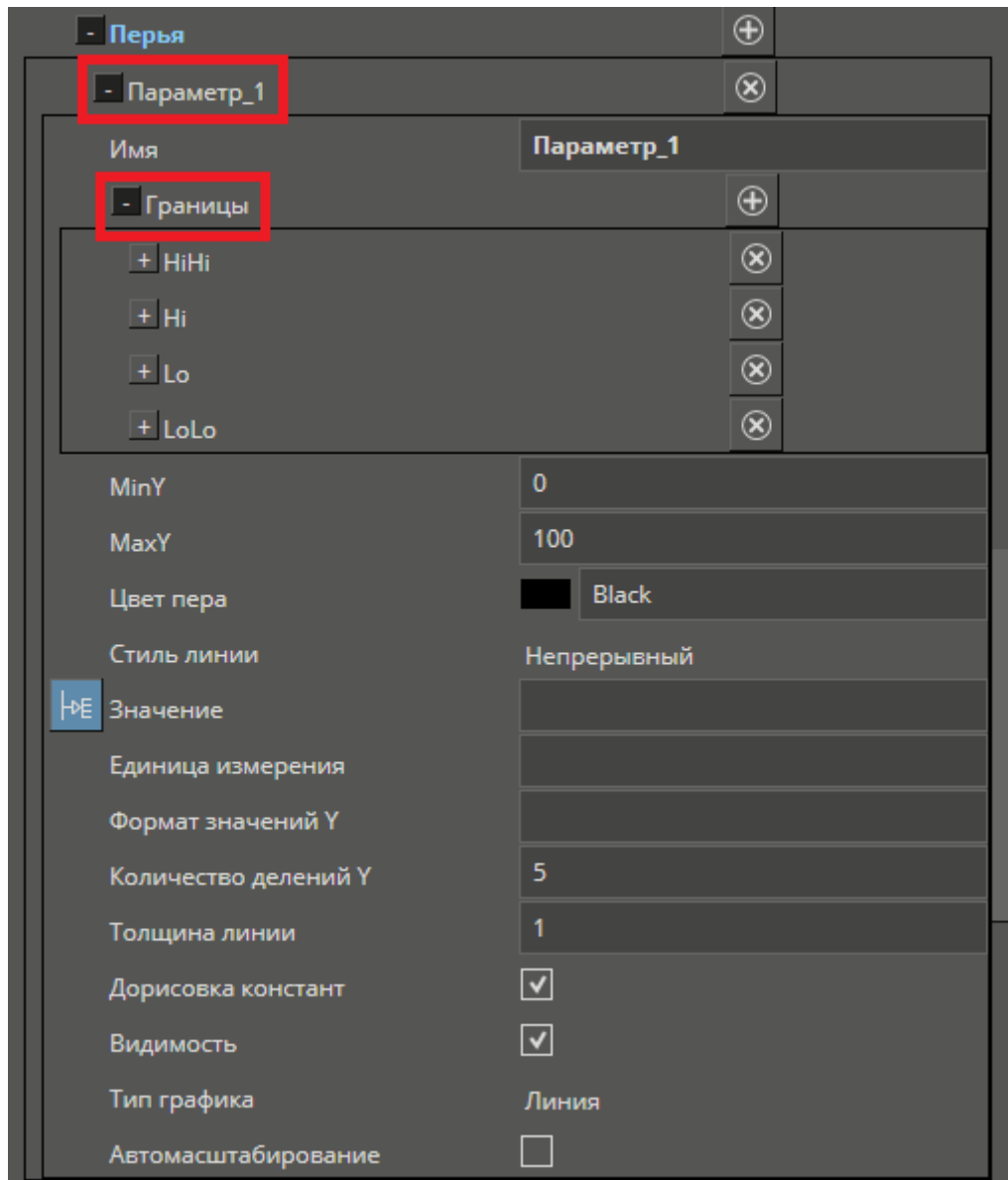
При создании библиотечного элемента внутренние связи типа элемента в панели свойств отображаются, а внутренние связи экземпляра – нет. Рассмотрим на примере элемента Индикатор из библиотеки Библиотеки.BaseObjects.Окна: на верхнем рисунке отображаются свойства библиотечного элемента, а на нижнем рисунке – свойства экземпляра этого элемента, добавленного в другое окно:







5.7.5. ПАНЕЛЬ СВОЙСТВ. ОТОБРАЖЕНИЕ СВОЙСТВ, ИМЕЮЩИХ СТРУКТУРУ

Свойство может иметь структуру (подчиненные параметры/свойства). Подчиненные параметры/свойства также могут иметь структуру:



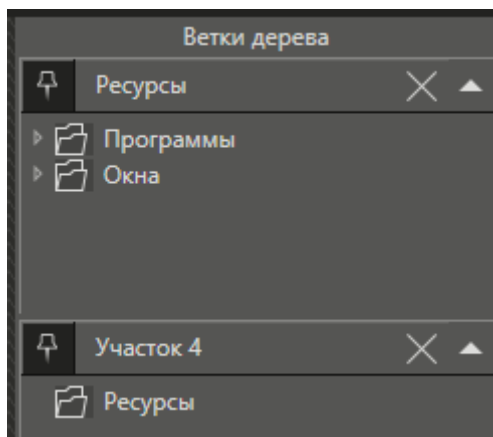
Чтобы раскрыть/свернуть структуру параметра/свойства, необходимо использовать кнопку **+** / **-**, расположенную слева от имени параметра/свойства.

Если параметр/свойство – массив, то предусмотрены следующие инструменты:

-  – добавить элемент массива;
-  – удалить выделенный элемент массива.

5.8. ПАНЕЛЬ ВЕТОК ДЕРЕВА

Панель веток открывается справа от дерева MasterSCADA при помощи специального пункта контекстного меню объекта, группы и других элементов. Дублирует часть дерева. Служит для упрощения работы с деревом. Правила работы с деревом действуют и в панели веток.



Как правило, панель веток открывается в случаях, когда необходимо установить связи между элементами, находящимися в дереве далеко друг от друга.

5.9. ЛЕГЕНДА

Легенда находится в центральной нижней части интерфейса среды разработки MasterSCADA 4D под рабочей областью редакторов. В ней содержатся различные элементы, необходимые для работы в редакторах программ и окон.

Вид легенды зависит от открытого редактора:

Легенда редакторов программ



Легенда редактора НМІ

5.10. ТИПОВЫЕ ИНСТРУМЕНТЫ И ОПЕРАЦИИ


- Элементарные инструменты
- Типовые инструменты окон
- Типовые операции
- Типовые инструменты дерева


5.10.1. СВЯЗЫВАНИЕ ПЕРЕМЕННЫХ

Для связывания переменных/параметров необходимо выполнить перетаскивание одного параметра из любого окна/редактора, в котором параметр отображается, на другой параметр. В общем случае, направление перетаскивания определяет направление передачи значения (см. также Неявная конвертация значений).

Если связь разрешена, то курсор имеет вид , в противном случае курсор выглядит как  (см. также Перетаскивание с удержанием ПК).

Если параметр участвует в связи, то рядом с его именем появляются дополнительные обозначения. По положению стрелочки относительно прямоугольника можно определить направления связи:

 - параметр имеет входящую связь, т.е. принимает значение от какого-либо источника;

 - параметр имеет исходящую связь, т.е. передает свое значение другому параметру.

В тех случаях, если допускается установка только одной связи данного типа, то при попытке установить вторую связь открывается диалог подтверждения замены связи.

Замечания о связях переменных

1. Для переменной допускается задавать несколько входящих связей от параметров окон.
2. InOut-параметр ФБ может передавать свое значение в выходной канал и принимать значение из входного канала. Он также может быть связан с параметрами тегов/объектов. Входы ФБ можно связывать с входными каналами, выходы ФБ – с выходными каналами.
3. Параметр окна может иметь несколько входящих связей.
4. Поддерживается связь из структуры в параметр, а также из параметра в структуру.
5. Отсутствуют ограничения на связь ФБ/тревоги только с параметрами того же объекта или его подобъектов.

См. также Связывание переменных на FBD- и LD-диаграммах .

Примеры возможных типов межузловых связей

1. Параметр объекта Узла 1 -> Параметр объекта Узла 2
2. Параметр окна Узла 1 -> Параметр объекта Узла 2 (значение передается по изменению)
3. Параметр окна Узла 1 <- Параметр объекта Узла 2



5.10.2. ЭЛЕМЕНТАРНЫЕ ИНСТРУМЕНТЫ

- Флаг
- Кнопка
- Набор кнопок типа radio button
- Выпадающий список

- Поле редактирования
- Текстовый редактор
- Инструмент задания пути к папке
- Контекстное меню
- Диалог подтверждения/отмены действия
- Всплывающая подсказка

Флаг

Флаг – это элемент управления, который может иметь следующий вид:

-  – флаг не установлен;
-  – флаг установлен.

Для переключения состояния флага необходимо нажать на левую кнопку мыши внутри флага.

Если флаг установлен, то применяется соответствующая опция (описание опции находится рядом с флагом).

, Кнопка

Кнопка – это элемент управления, имеющий два устойчивых положения: нажатое (утопленное) и отжатое. В разных положениях на кнопке могут также отображаться разные картинки. Ниже показан одна и та же кнопка в двух положениях:



– отжатое положение;

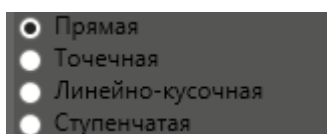


– нажатое (утопленное) положение.

Для изменения состояния кнопки требуется на неё нажать.

Набор кнопок типа radio button


Набор кнопок типа radio button предназначен для выбора только одной функции из ряда взаимоисключающих функций , указанных рядом с кнопками):



Для выбора нужно нажать на соответствующую кнопку. При этом кнопка отмечается точкой внутри.

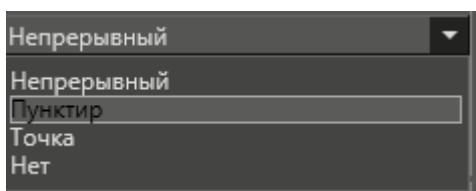
Выпадающий список


Выпадающий список (combo-box) предназначен для выбора значения.

Если список свернут, то отображается только элемент управления списком (содержит окно и кнопку ):



Если список раскрыт, то отображается элемент управления и, собственно, список:



Для того чтобы раскрыть список (или свернуть его без выбора значения), необходимо нажать на кнопку  элемента управления. Для того чтобы свернуть список без выбора значения, можно также просто нажать клавишу Esc.

Для того чтобы выбрать значение из списка, необходимо нажать на это значение. При этом список сворачивается, а выбранное значение отображается в окне элемента управления. Для того чтобы свернуть список с выбором значения, на которое установлен курсор, можно также просто нажать клавишу Enter.

В некоторых случаях окно элемента управления может представлять собой поле редактирования, значение в котором, помимо выбора его в списке, может быть также задано вручную с помощью клавиатуры.

Поле редактирования

Как правило, поле редактирования представляет собой однострочный текстовый редактор, в котором существуют следующие отличия от типовых операций текстового редактора:

- двойное нажатие ЛК – выделение всей строки;
- Esc – отмена всех изменений, выполненных с начала редактирования до нажатия клавиши Enter;
- Enter – применение изменений; для применения изменений (окончания редактирования) можно также выделить другой объект.

Текстовый редактор

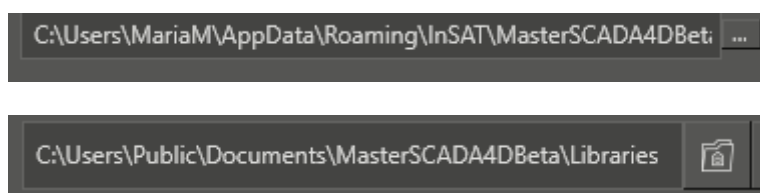
Текстовый редактор обеспечивает ввод и редактирование текста с помощью клавиатуры, мыши и команд контекстного меню.

Типовые операции в текстовом редакторе:




- Ctrl+A – выделение всего текста (аналог команды Выделить все);
- выделение мышью области с текстом – выделение произвольного текста;
- перетаскивание выделенного текста – перемещение этого текста;
- двойное нажатие ЛК по слову – выделение слова;
- Shift+ → / ← – выделение одного символа / снятие выделения одного символа;
- Ctrl+Shift+ → / ← – выделение слова / снятие выделения слова;
- Shift+ ↓ / ↑ – выделение строки / снятие выделения строки;
- Shift+Home/End – выделение текста от позиции курсора до начала/конца строки;
- Shift+PgUp/PgDn – выделение текста от позиции курсора до начала/конца окна;
- Ctrl+Shift+Home/End – выделение всего текста от позиции курсора до начала/конца;
- Backspace или Del – удаление выделенного текста; в отсутствие выделения – удаление одного символа соответственно слева или справа от курсора;
- → / ← – перемещение курсора на одну позицию вправо/влево;
- ↓ / ↑ – перемещение курсора на одну строку вниз/вверх;
- Ctrl+Home/End – перемещение курсора в начало/конец текста.

Инструмент задания пути к папке

Возможные виды инструментов показаны на рисунке:



В данном комбинированном инструменте содержатся следующие элементарные инструменты:

- поле редактирования, в котором путь может быть задан вручную;
- кнопки  или  для открытия стандартного диалога Windows выбора папки;
- кнопка  для задания папки по умолчанию.

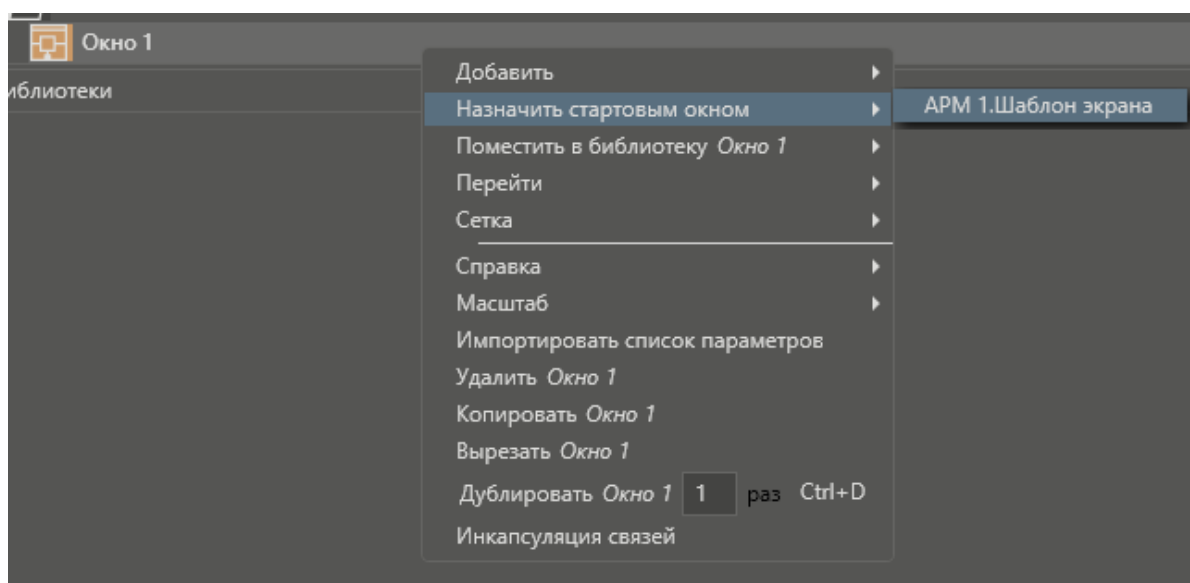
Контекстное меню

Контекстное меню – это список действий/команд, которые могут быть выполнены над объектом (группой объектов), если такие действия/команды определены для этого объекта (группы объектов).

Для вызова контекстного меню объекта необходимо нажать ПК на этом объекте (объект при этом выделяется). Для вызова контекстного меню группы объектов необходимо сначала выполнить групповое выделение этих объектов, после чего, не перемещая мышь, нажать ПК.

Контекстное меню нескольких выделенных объектов содержит общие команды для выделенных объектов.

Контекстное меню может содержать подменю:



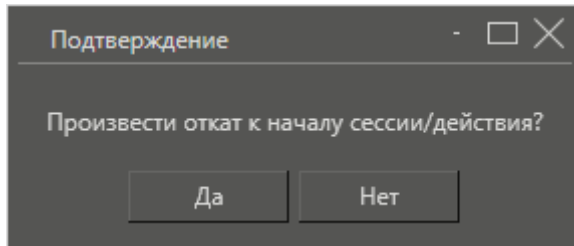
Для того, чтобы закрыть контекстное меню, необходимо нажать на любое свободное место редактора или нажать клавишу Esc. Если были открыты подменю, то однократное нажатие клавиши Esc последовательно закрывает подменю, начиная с самого нижнего уровня, а при всех закрытых подменю закрывает само меню.

Для того, чтобы выполнить команду контекстного меню (подменю), необходимо нажать на соответствующий пункт меню (подменю) или, наведя курсор на команду, нажать клавишу Enter.


Диалог подтверждения/отмены действия

В ходе выполнения команды этот диалог появляется на экране непосредственно перед выполнением действия, как последнее предупреждение пользователя.

В качестве примера, на рисунке показаны диалоги подтверждения возврата к предыдущей версии проекта:

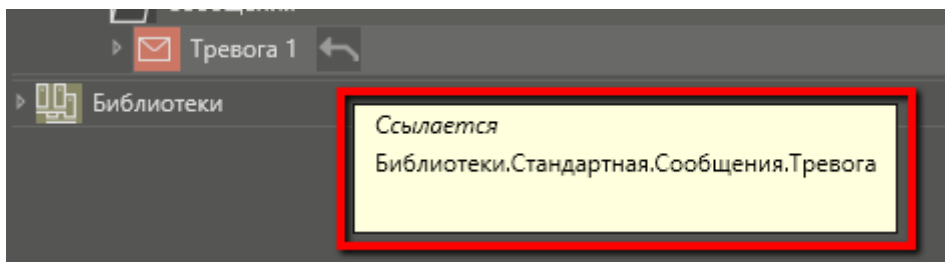


Инструменты диалога:

- Да – выполнить действие и закрыть диалог;
- Нет или  – не выполнять действие и закрыть диалог.

Всплывающая подсказка

Всплывающая подсказка (tooltip) – это маленькое окно с кратким информационным сообщением (подсказкой). Всплывающая подсказка открывается при наведении указателя мыши на тот или иной объект, свойство элемента, либо на символы, расположенные рядом с названием.



Во всплывающей подсказке к элементу дерева структуры проекта отображается имя элемента и комментарий (унаследованный или переопределенный).




Во всплывающей подсказке к графическому элементу отображается имя элемент и его тип . При наведении указателя мышь на свойство элемента, отображается его описание (при наличии)

5.10.3. ТИПОВЫЕ ИНСТРУМЕНТЫ ОКОН

- Инструменты окон
- Инструменты встроенных окон/диалогов
- Кнопка ОК
- Кнопка Отмена

Инструменты окон

Существуют следующие стандартные оконные инструменты и операции:


- двойное нажатие ЛК на заголовке окна – переключение вида отображения (полноэкранный/в окне);
-  – команда минимизации окна;
-  – команда восстановления окна;
-  – команда закрытия окна.

При закрытии окна редактора проекта закрывается проект, который был открыт в этом окне (см. Закрытие проекта). Для закрытия окна редактора может быть использована команда Выход;

- клавиша TAB – перевод фокуса на следующий объект окна.

Инструменты встроенных окон/диалогов

Во встроенных окнах и диалогах редактора проекта поддерживаются, как правило, следующие стандартные оконные инструменты и операции:

-  – команда закрытия окна/диалога;
- изменение размеров окна/диалога;
- прокрутка ;
- двойное нажатие ЛК на заголовке окна/диалога – переключение вида отображения (полноэкранный/в окне);
- клавиша TAB – перевод фокуса на следующий объект окна/диалога;
- кнопка ОК;
- кнопка Отмена.

Кнопка ОК

Применяет выполненные изменения и закрывает диалог/редактор.

Эта кнопка может также иметь название, описывающее её действие.

Кнопка Отмена

Отменяет выполненные изменения и закрывает диалог/редактор.

5.10.4. ТИПОВЫЕ ОПЕРАЦИИ

- Открытие элемента в редакторе
- Переход к редактированию

- Прокрутка
- Перетаскивание
- Выделение
- Выделение рамкой
- Групповое выделение
- Снятие выделения
- Изменение размеров
- Переход по связи

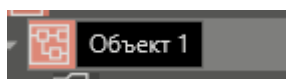
Открытие элемента в редакторе

Для того, чтобы открыть элемент в соответствующем редакторе, необходимо дважды нажать ЛК на этот элемент.

Переход к редактированию

Для перехода к редактированию значения (в дереве или в любом редакторе) необходимо выполнить нажатие на это значение в течение 1 с после его выделения.

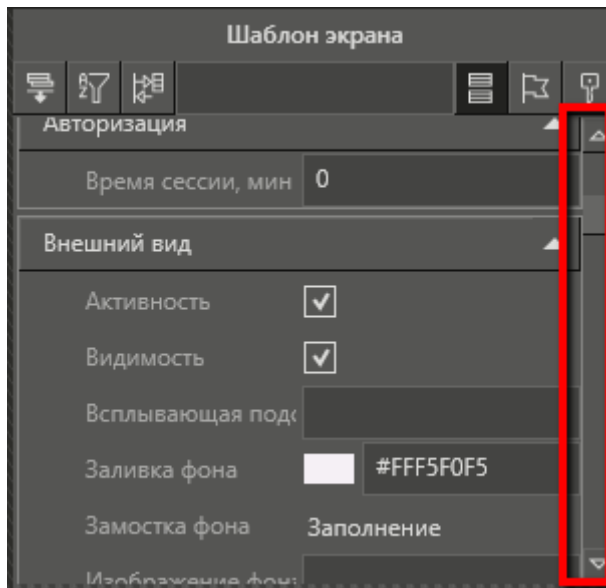
По этой команде на месте значения открывается инструмент его задания (выпадающий список, поле редактирования и т.п.). В поле редактирования текущее значение выделено:



Прокрутка

Если содержимое не помещается по горизонтали и/или по вертикали в видимую область окна, то возможно перемещение содержимого (прокрутка соответственно по горизонтали и/или по вертикали).

Стандартные элементы перемещения содержимого – это полосы горизонтальной и вертикальной прокрутки, которые располагаются соответственно в нижней и правой части окна:



Полосы прокрутки обеспечивают следующие операции:

- однократное нажатие на стрелку – дискретное перемещение содержимого с мелким фиксированным шагом в направлении, противоположном стрелке;
- однократное нажатие на область между стрелкой и ползунком – дискретное перемещение содержимого с шагом "одна видимая область" в направлении, противоположном стрелке;
- перетаскивание ползунка полосы прокрутки – плавное перемещение содержимого в направлении, противоположном перетаскиванию.

Для перемещения содержимого могут использоваться также следующие операции:

- перетаскивание произвольной точки содержимого с удержанием ПК – перемещение в направлении перетаскивания;
- вращение колеса мыши при расположении курсора в окне – прокрутка содержимого окна по вертикали.
- вращение колеса мыши при нажатой клавише SHIFT - прокрутка содержимого окна по горизонтали.

Кроме того, для прокрутки FBD-, LD- и SFC-диаграмм предусмотрены специфические средства (см. Прокрутка диаграмм).

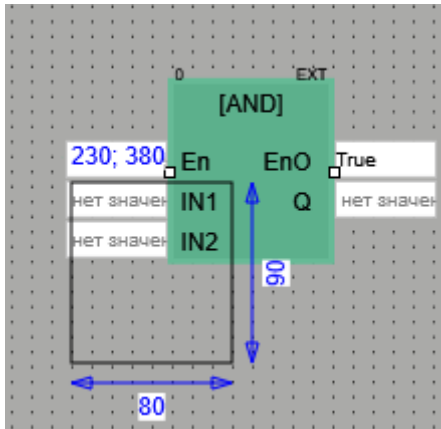
Перетаскивание

Для выполнения операции перетаскивания (метод drag-n-drop, "перетащи и оставь") необходимо выполнить следующие действия:

- установить курсор в нужную позицию (такой позицией может быть как объект, так и свободное место окна/редактора);

- нажать и удерживать ЛК;
- переместить курсор в конечную позицию;
- отпустить кнопку мыши.

При перетаскивании FBD-блока, надписи или графического элемента отображаются:



- координаты левого верхнего угла ограничивающего прямоугольника (слева вверху);
- ширина и высота ограничивающего прямоугольника (рядом с соответствующими стрелками);
- угол поворота (справа внизу, только для графического элемента).

В редакторе проекта поддерживается также перетаскивание с удержанием ПК .

Выделение

Для выделения объекта необходимо нажать на него ЛК (см. также Выделение элементов).

Выделение рамкой

Для выделения (как правило, группы объектов) с помощью мыши необходимо использовать перетаскивание произвольной свободной точки редактора. При этом на экране отображается полупрозрачный прямоугольник. В результате выделяются объекты, которые полностью попадают в этот прямоугольник.

Групповое выделение

Для того, чтобы выделить группу объектов выборочно, необходимо последовательно нажать на эти объекты с одновременным удержанием клавиши CTRL. Для выборочного снятия/выделения необходимо нажать на требуемые выделенные объекты с одновременным удержанием клавиши CTRL.

Для выделения группы объектов в дереве необходимо выполнить нажатие ЛК на объект obj1, после чего, удерживая клавишу SHIFT, выполнить нажатие ЛК на объект obj2: по этой команде выделяются все объекты, расположенные между obj1 и obj2. При этом объекты obj1 и obj2 также выделяются.



Для группового выделения можно также использовать выделение рамкой .



Для выделения дочерних элементов первого уровня используется команда выделения дочерних элементов.

Снятие выделения

Для того, чтобы снять выделение, нужно нажать ЛК на любое свободное место редактора или нажать клавишу ESC.

Изменение размеров

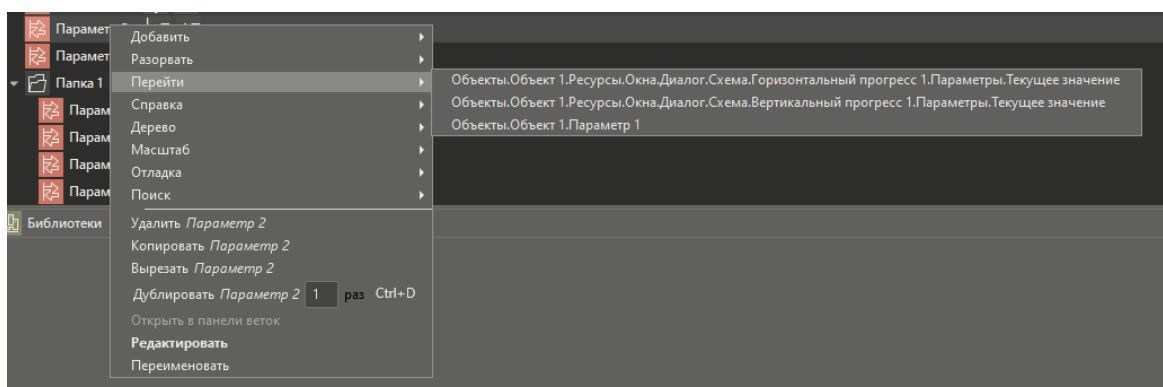
Для изменения размеров выделенного объекта необходимо установить курсор в вершину прямоугольника, ограничивающего объект (курсор при этом принимает вид  или ), нажать ЛК и, удерживая кнопку нажатой, переместить курсор в нужную точку, после чего кнопку мыши отпустить. При одновременном с этим удержанием клавиши CTRL размеры изменяются пропорционально.


Для изменения высоты или ширины выделенного объекта необходимо установить курсор в середину соответствующей стороны ограничивающего объект прямоугольника (курсор при этом принимает вид  или ) и выполнить аналогичные действия.

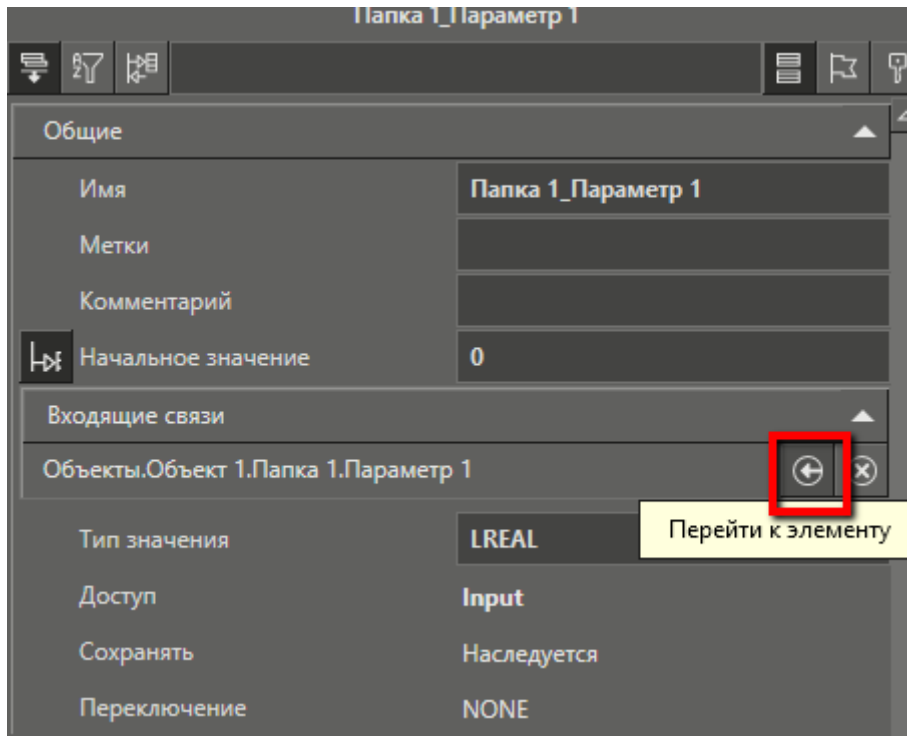
Таким же способом можно изменять размеры окон и диалогов.

Переход по связи

Если параметр проекта имеет связь, то чтобы осуществить быстрый переход в дереве на связанный параметр необходимо в контекстном меню выбрать пункт Перейти:

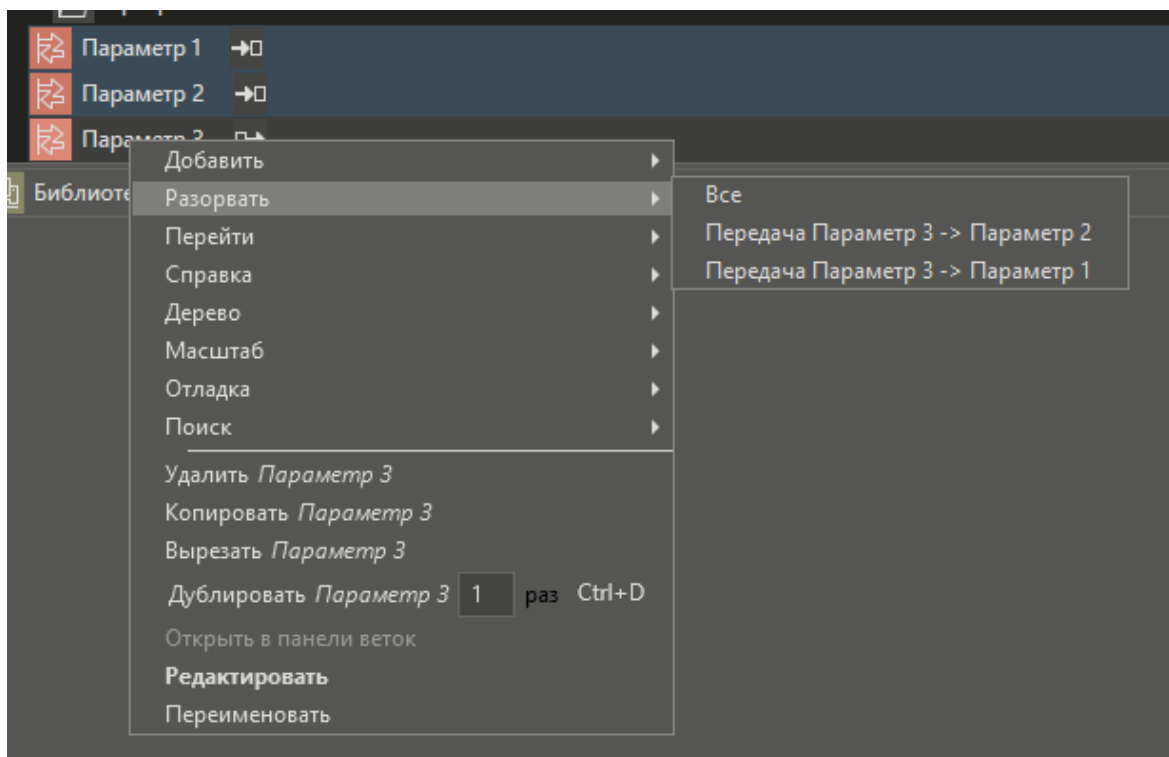


Если свойство имеет связь с параметром, то чтобы осуществить быстрый переход нужно нажать в панели свойств кнопку  в строке свойства:



Разорвать связь









Разорвать связь параметров можно при помощи контекстного меню Разорвать:



Связанные параметры подсвечиваются в дереве проекта. Контекстное меню позволяет выбрать связь, которую нужно разорвать, либо разорвать все связи данного параметра.

5.10.5. ТИПОВЫЕ ИНСТРУМЕНТЫ ДЕРЕВА

Дерево снабжено следующими типовыми инструментами:

-  /  ( / ) – переключатель отображения группы дерева (свернутое/раскрытое, предварительное выделение группы не требуется и при использовании переключателя не производится);
-  – раскрыть выделенную группу дерева;
-  – свернуть выделенную группу дерева;
-  /  – выделить следующий/предыдущий отображаемый элемент.

5.10.6. ГОРЯЧИЕ КЛАВИШИ РЕДАКТОРА ПРОЕКТОВ

Для упрощения работы в среде разработки (в редакторе проектов) можно использовать "горячие" клавиши.

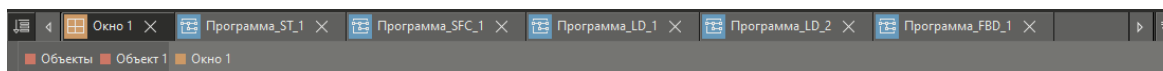
Общие:

Клавиша	Назначение
Клавиша курсора Влево	В диалогах, в текстовых полях свойств перемещение на один символ влево, при навигации по дереву осуществляет переход на следующий элемент вверх того же уровня, либо переходит на уровень родительского элемента. В редакторе окон и в редакторе FBD-программ перемещает выделенный элемент на 1 пиксель влево
Клавиша курсора Вправо	В диалогах, в текстовых полях свойств перемещение на один символ вправо, при навигации по дереву осуществляет переход на следующий элемент вниз, если элемент содержит дочерние элементы, то переход произойдет на дочерний элемент. редакторе окон и в редакторе FBD-программ перемещает выделенный элемент на 1 пиксель вправо
Клавиша курсора Вверх	В диалогах, в текстовых полях свойств перемещение на одну строку вверх, при навигации по дереву осуществляет переход на следующий элемент вверх. редакторе окон и в редакторе FBD-программ перемещает выделенный элемент на 1 пиксель вверх
Клавиша курсора Вниз	В диалогах, в текстовых полях свойств перемещение на одну строку вниз, при навигации по дереву осуществляет переход на следующий элемент вниз, без учета дочерних элементов. редакторе окон и в редакторе FBD-программ перемещает выделенный элемент на 1 пиксель вправо

Клавиша	Назначение
CTRL+ и CTRL-	Увеличивают и уменьшают масштаб редакторов, открывающихся в рабочей области.
Ctrl+1	Устанавливает масштаб открытого редактора по умолчанию
Ctrl+0	Устанавливает масштаб открытого редактора таким образом, чтобы были видны все его элементы
Ctrl+Z	Отменяет совершенное действие
Shift+Ctrl+Z	Возвращает ранее отменное действие
Ctrl+D	Дублирует выделенный элемент/элементы
Ctrl+F	Открывает окно поиска в текстовой программе
F2	Переходит в режим редактирования имени элемента в дереве.
ESC	Отменяет или отказывается от выполнения различных действий, например, перетаскивание элемента по схеме, отмена диалога, сброс выделения и т.п.
F1	Вызывает справочную систему
F11	Открывает текущий редактор во весь экран (деревья, контекстная панель, будут скрыты)
Выделение элементов с нажатой клавишей SHIFT	Позволяет выделить группу видимых элементов в дереве, которые находятся между выделяемыми. Если выделение производится рамкой (мышью с нажатой левой клавишей), то в выделение попадут только те элементы, которые находятся внутри выделяемой рамки полностью
Выделение элементов с нажатой клавишей CTRL	Добавляет новые элементы, к ранее выделенным. Если выделение производится рамкой (мышью с нажатой левой клавишей), то в выделение попадут все элементы, которые находятся в области рамки, даже если они выделены лишь частично

Клавиша	Назначение
Enter	Если нажать клавишу в дереве, на элементе, который может иметь свою вкладку в панели редакторов (например, окна, некоторые протоколы, программы), то эта вкладка откроется
F5	Производит подключение и загрузку текущего проекта к исполнительной системе. Если исполнительная система не запущена, то запустится демоверсия, которая входит в состав среды разработки.
SHIFT+F5	Производит отключение от исполнительной системы, если исполнительная система была запущена средой разработки, она завершит свой процесс.
CTRL+ ↓	Перемещает элемент на одну позицию вниз в дереве проекта
CTRL+ ↑	Перемещает элемент на одну позицию вверх в дереве проекта
CTRL+ колесо мыши	Изменяется масштаб рабочей области редакторов или панели, в которой установлен курсор.
SHIFT+ колесо мыши	Изменение видимой области рабочей области редакторов

Навигация по вкладкам, открытым в рабочей области:

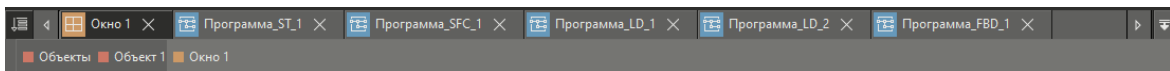





Клавиша	Назначение
CTRL+TAB/ CTRL+PgDn	Производит движение по последовательности открытых вкладок вправо
Ctrl + Shift + Tab / Ctrl + PgUp	Производит движение по последовательности открытых вкладок влево
Ctrl + B	Перемещает в конец последовательности открытых вкладок

Ctrl + E	Перемещает в конец последовательности открытых вкладок
Ctrl + W / Ctrl + F4	Закрывает активную вкладку
Ctrl + Shift + W	Закрывает все вкладки
Alt + P	Скрывает/открывает строку пути к владельцу вкладки

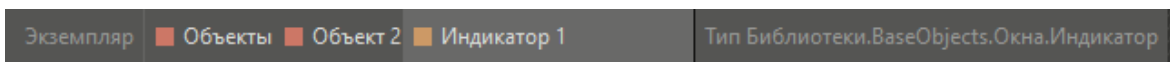
5.10.7. РАБОТА С ВКЛАДКАМИ РАБОЧЕЙ ОБЛАСТИ

Если открыто несколько вкладок элементов в рабочей области, то они располагаются друг за другом:





Если все вкладки не умещаются на экране, то появятся кнопки   нажатие на которые изменит видимые вкладки в окне. При помощи кнопки  можно открыть список всех открытых вкладок.

Под вкладками находится строка, которая показывает, в каком месте проекта находится редактируемый элемент - клик по имени элемента приведет к его выделению в дереве, а следовательно в панели свойств отобразятся и его свойства. Если открыт экземпляр какого-либо библиотечного элемента, то строка будет иметь вид:



С левой стороны будет указано, что данный элемент является экземпляров, а с правой стороны будет указан его тип.

Скрыть информационную панель можно при помощи кнопки .

Клик левой кнопкой мыши по вкладке приведет к переключению на нее, закрыть вкладку можно при помощи кнопки  открытым в рабочей области. Работа с вкладками возможно при помощи горячих клавиш:

Клавиша	Назначение
CTRL+TAB/ CTRL+PgDn	Производит движение по последовательности открытых вкладок вправо

Ctrl + Shift + Tab / Ctrl + PgUp	Производит движение по последовательности открытых вкладок влево
Ctrl + B	Перемещает в конец последовательности открытых вкладок
Ctrl + E	Перемещает в конец последовательности открытых вкладок
Ctrl + W / Ctrl + F4	Закрывает активную вкладку
Ctrl + Shift + W	Закрывает все вкладки
Alt + P	Скрывает/открывает строку пути к владельцу вкладки

5.11. ОПЕРАЦИИ В РЕДАКТОРЕ ПРОЕКТА

5.11.1. ОПЕРАЦИИ С ПРОЕКТАМИ

- Создание проекта
- Открытие проекта
- Сохранение проекта
- Заккрытие проекта
- Удаление проекта

Создание проекта

Для создания проекта используется диалог создания проекта .

Для того, чтобы открыть этот диалог, используются стартовое меню или кнопка панели инструментов Управление проектами

Открытие проекта

Для открытия проекта используется диалог открытия проекта или список последних открытых проектов.

Для того, чтобы открыть диалог открытия проекта, используются стартовое меню или кнопка панели инструментов Управление проектами

Сохранение проекта

Для сохранения проекта используются кнопки панели инструментов Сохранить и Управление проектами

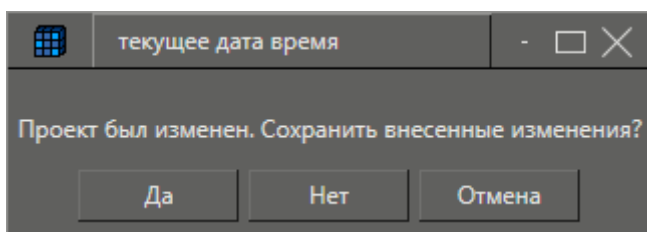
Важно! При сохранении проекта очищаются стеки отмены/возврата действий (стеки UnDo/ReDo). Откатить изменения можно, используя кнопку панели инструментов История версий.

Заккрытие проекта

Для закрытия проекта используются кнопка панели инструментов Управление проектами и кнопка Закрыть

Проект также закрывается при закрытии окна редактора проекта.

Если после редактирования проект не был сохранен, то при закрытии этого проекта открывается соответствующий диалог подтверждения/отмены действия :

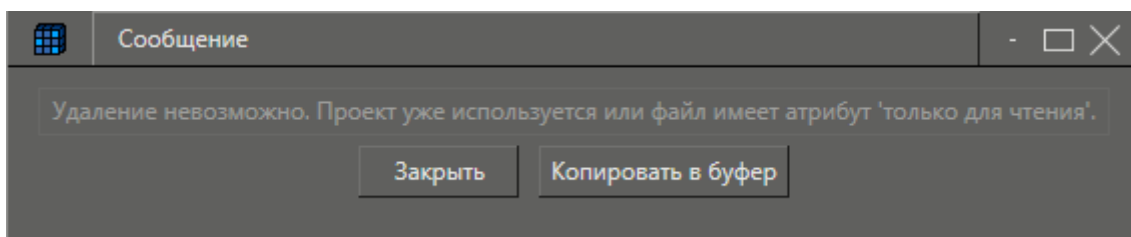


После закрытия проекта открывается стартовое меню .

Удаление проекта

Для удаления проекта используется кнопка Удалить диалога открытия проекта . При выполнении этой команды на экране появляется соответствующий диалог.

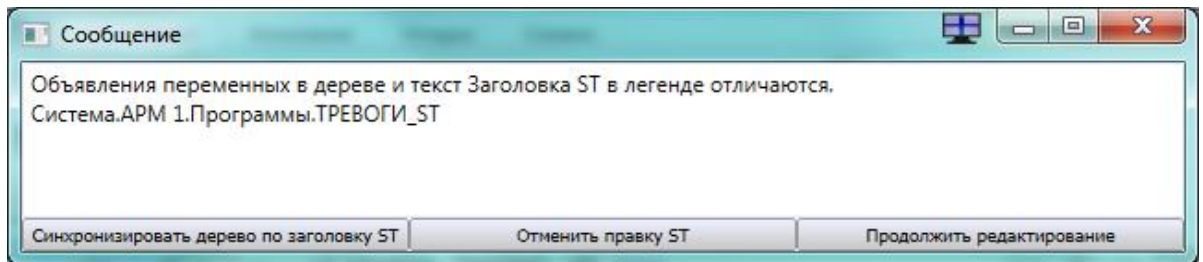
Если удаляемый проект открыт (в любом окне редактора проекта), то на экране появляется сообщение о невозможности удаления:



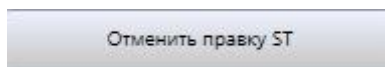
5.11.1.1. ОКНО ВЫБОРА ТРАНСЛЯЦИИ

- Трансляция по дереву
- Трансляция по заголовку
- Возврат к редактированию
- Закрытие диалога выбора трансляции

Окно (см. Инструменты окон) содержит предупреждение о различии наборов переменных программного компонента в дереве и в тексте заголовка на вкладке Легенда. Вкладка 'Заголовок ST', а также инструменты разрешения данной ситуации:

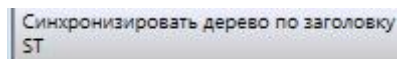


Трансляция по дереву



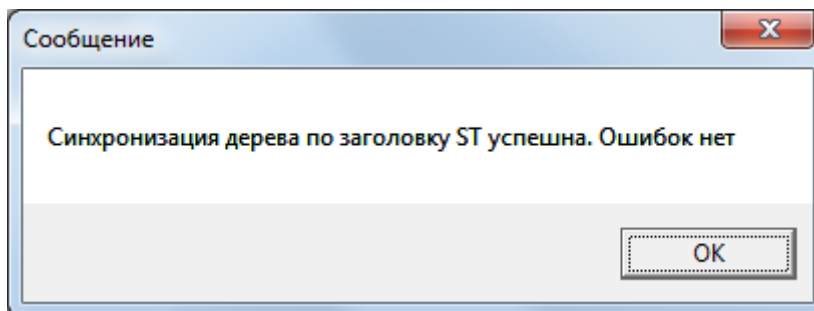
– по этой команде выполняется трансляция программного компонента с переменными, заданными в дереве (текст заголовка в легенде синхронизируется по дереву), и результат трансляции проверяется (см. Команда проверки программного компонента).

Трансляция по заголовку

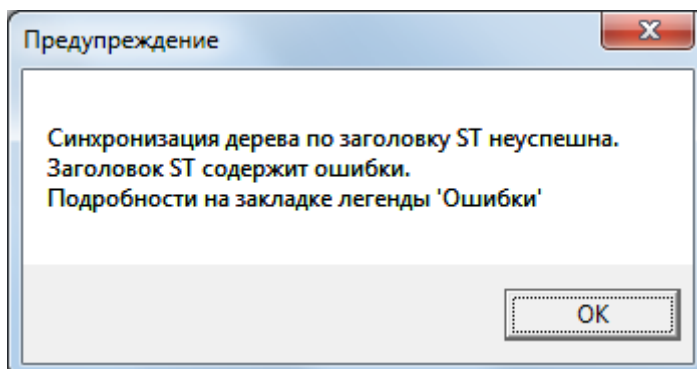


– по этой команде выполняются следующие действия:

- текст заголовка проверяется на наличие ошибок:
- если ошибок в заголовке нет, дерево синхронизируется по тексту заголовка, и на экране появляется соответствующее сообщение:



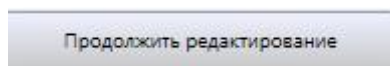
- если ошибки в заголовке есть, на экране появляется соответствующее сообщение:



По команде ОК текст заголовка в легенде синхронизируется по дереву;


- выполняется трансляция.

Возврат к редактированию



– по этой команде диалог закрывается, при этом программный компонент не транслируется (т.е. выполняется возврат к редактированию заголовка).

Заккрытие диалога выбора трансляции

При закрытии диалога с помощью типового инструмента  выполняется трансляция по дереву.

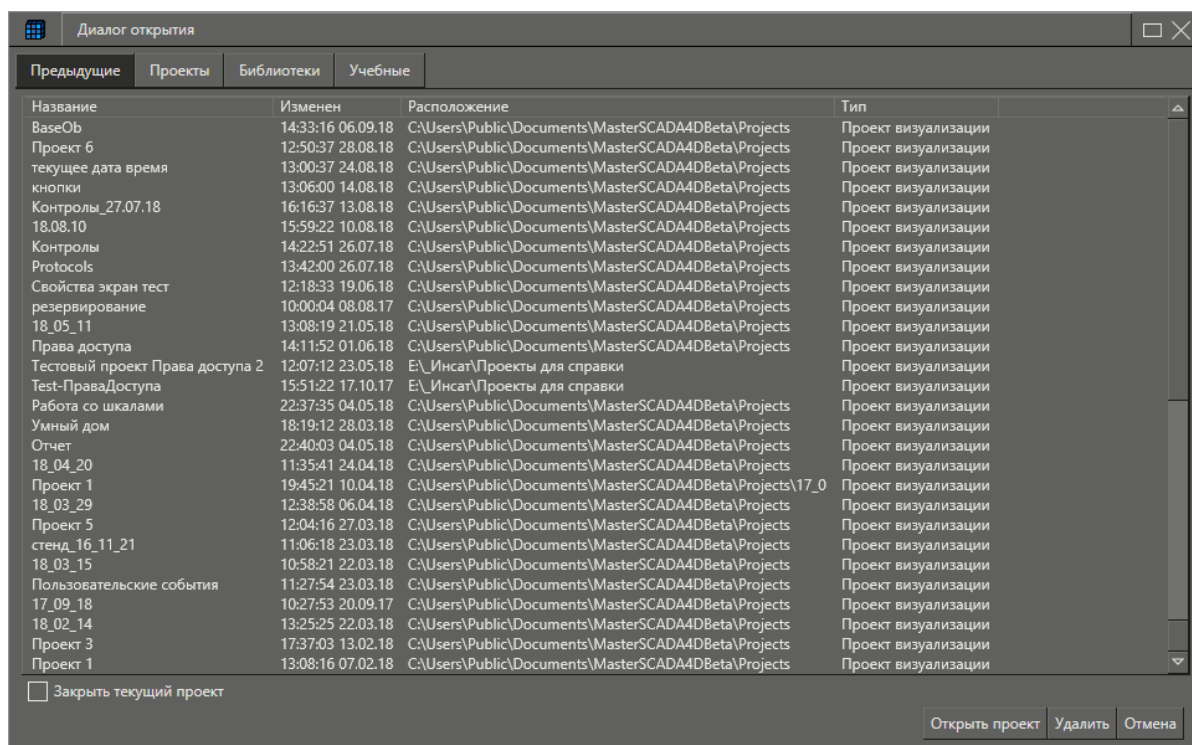
5.11.1.2. ДИАЛОГ ОТКРЫТИЯ ПРОЕКТА

С точки зрения файловой системы, проект представляет собой папку, содержащую файлы (в т.ч. файл БД *.fdb) и вложенные папки. Переименовывать эти элементы средствами ОС не рекомендуется. Для переименования проекта следует использовать инструмент Сохранить как).

Важно! Если проект не отображается в диалоге открытия проекта, то, возможно, что папка проекта и файл БД имеют разные имена.

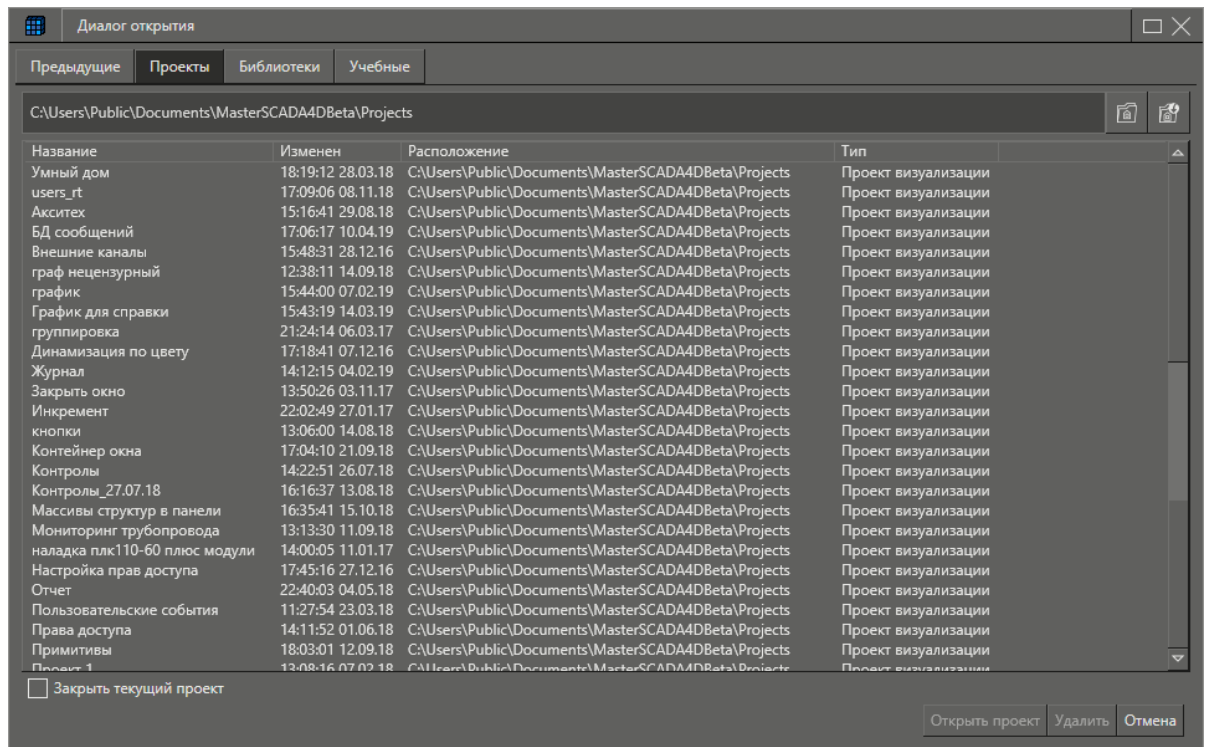
Диалог открытия проекта содержит 4 вкладки.

Вкладка Предыдущие



На этой вкладке отображается таблица параметров последних открытых проектов.

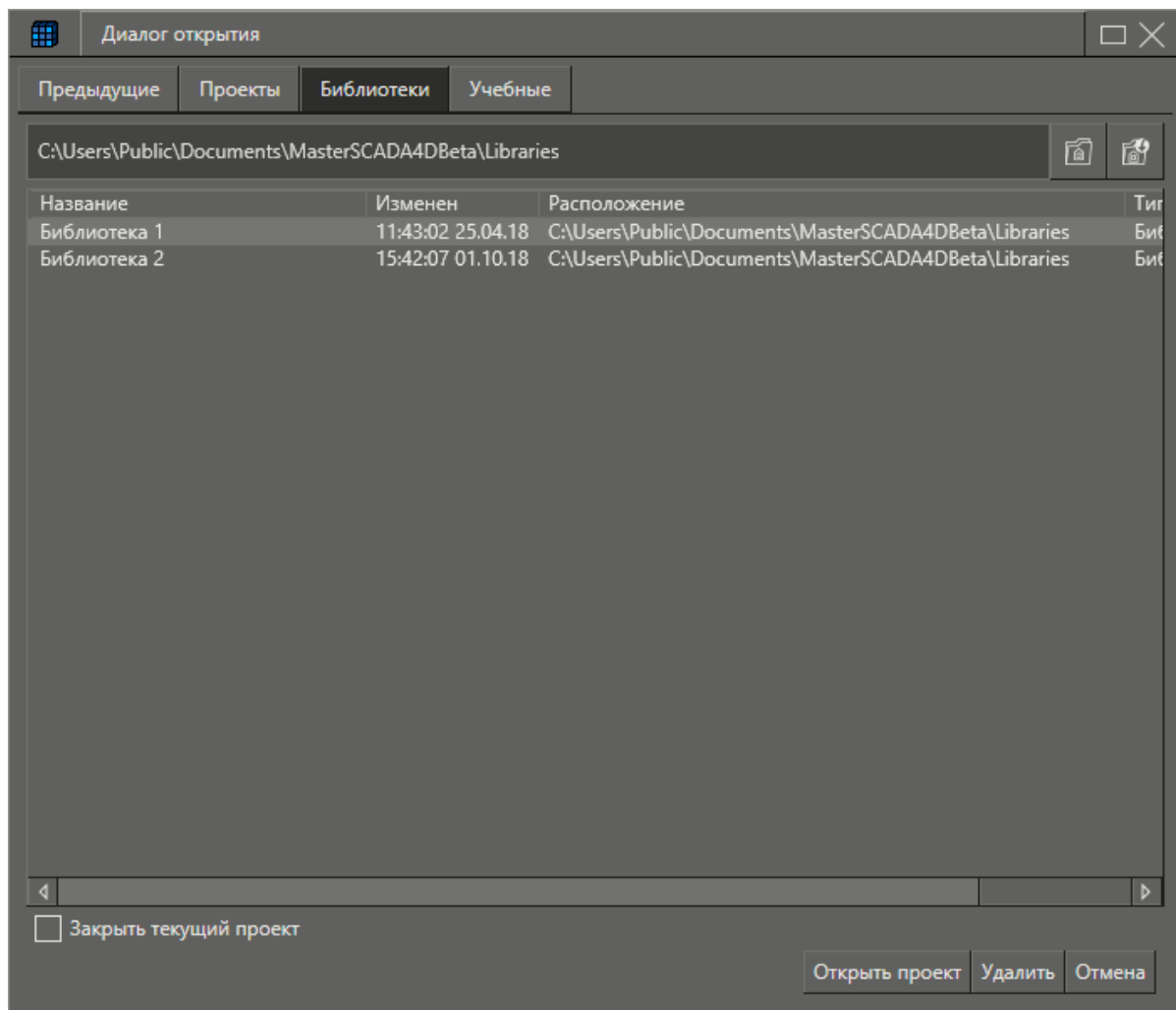
Вкладка Проекты



На этой вкладке отображается таблица параметров проектов, хранящихся в указанной папке.

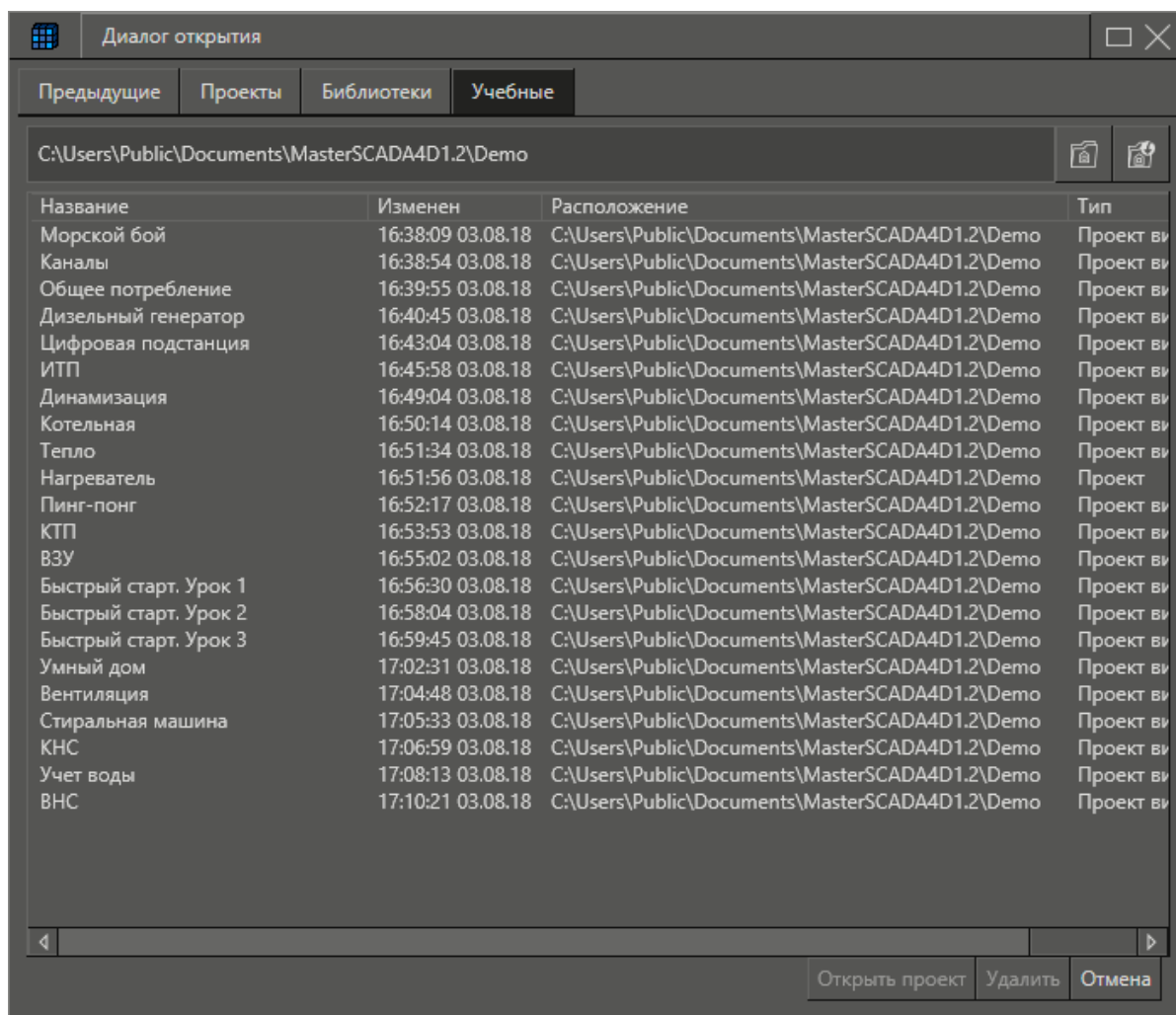
По умолчанию, проекты хранятся в папке
C:\Users\Public\Documents\MasterSCADA4D[номер версии]\Projects

Вкладка Библиотеки





На этой вкладке отображается таблица параметров библиотек, которые были созданы пользователем и сохранены. По умолчанию, библиотеки хранятся в папке *C:\Users\Public\Documents\MasterSCADA4D[номер версии]\Libraries*

Вкладка Учебные



Вкладка отображает демонстрационные проекты, входящие в состав MasterSCADA 4D.

Описание рабочих элементов окна

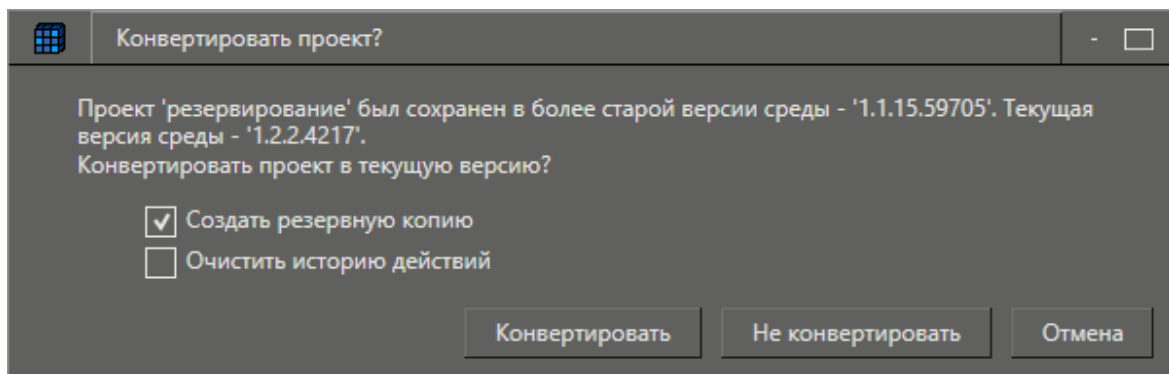
Название	Описание
Строка ввода адреса	Задается папка, из которой нужно открыть проект
	Кнопка позволяет открыть проект, находящийся в любом месте компьютера. После нажатия на неё открывается диалоговое окно Windows для выбора места на диске.
	В строке ввода адреса автоматически задается путь к рабочей папке среды разработки по умолчанию.
Столбцы таблицы	

Название	Описание
Название	Указано имя проекта.
Изменен	Показано время последнего сохранения проекта.
Расположение	Указано место хранения проекта на диске.
Тип	Задан тип проекта. В текущей версии среды разработки все проекты имеют тип Проект визуализации, независимо от настроек, сделанных внутри проекта.
Открыть проект	<p>Кнопка открывает проект, выделенный в таблице. Командой открытия проекта является также двойное нажатие левой кнопки мыши на проекте в списке.</p> <p>При выполнении команды открытия возможны следующие случаи:</p> <ul style="list-style-type: none"> • если выбранный проект открыт в некотором окне редактора проекта, то это окно активируется; • если выбранный проект не открыт ни в одном из окон редактора проекта, то проект открывается в новом окне. <p>Если открывается проект, созданный в другой версии (более ранней или поздней), то появится диалог, предлагающий выполнить конвертации проекта. Описание диалога см. ниже.</p> <p>При открытии проекта сравниваются версии библиотек, используемых в проекте, с версиями библиотек среды разработки, в случае их различия появляется диалог обновления библиотек.</p>
Удалить	Удаляет проект, выделенный в таблице
Отмена	Отменяет открытие проекта, закрывает диалог.
Закреть текущий проект	<p>Позволяет закрывать открытый ранее проект в текущей сессии среды разработки. Если флаг не установлен, то по команде открытия указанный проект открывается в новом окне редактора проекта. Если флаг установлен, то указанный проект открывается в текущем окне.</p> <p>При этом, если в текущем окне был открыт другой проект, то он закрывается. Если проект редактировался, и после этого не был</p>

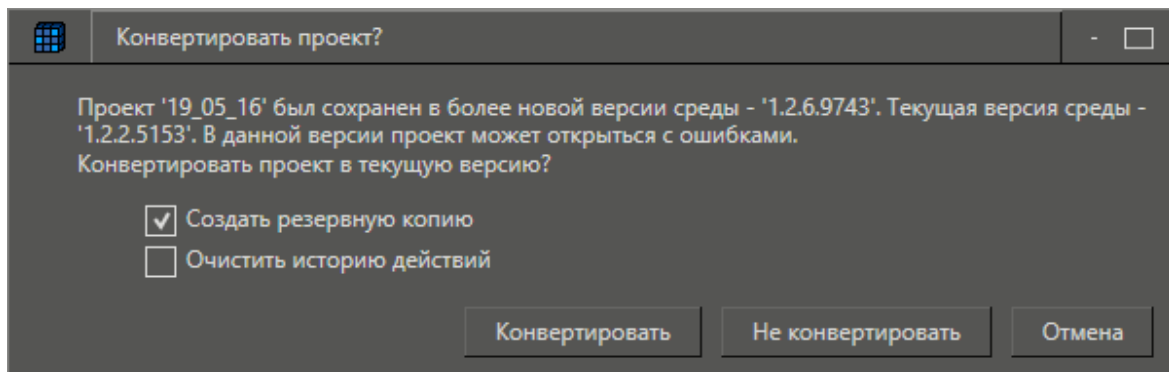
Название	Описание
	сохранен, то на экране появится диалог, с помощью которого можно сохранить или отменить выполненные изменения).

Конвертация проекта

Если проект был последний раз сохранен в одной из предыдущих версий среды разработки, то при открытии проекта появится диалог:



Если версия, в которой последний раз сохранялся проект, была более поздняя (новая), чем версия, в которой происходит открытие, то появится диалог:



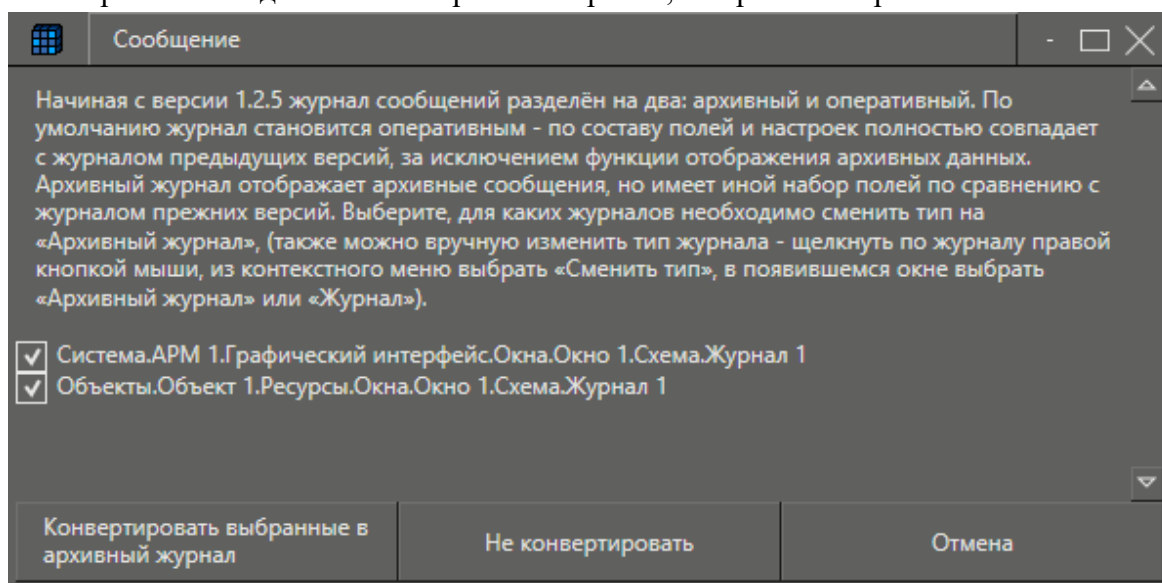
Описание диалога:

Название	Описание
Создать резервную копию	Если флаг установлен, то после конвертации проекта рядом с папкой проекта будет сохранен архив backup_[имя проекта].zip, содержащий проект до конвертации.
Очистить историю действий	Если флаг установлен, то проект, открытый после конвертации, не будет содержать историю действий разработчика.

Название	Описание
Конвертировать	Кнопка запускает процесс конвертации проекта. Время, необходимое для конвертации, зависит от размеров проекта.
Не конвертировать	Кнопка не запускает процесс конвертации. Проект будет открыт, но работать может с ошибками.
Отмена	Кнопка отменяет открытие проекта и закрывает диалог открытия

Конвертация журнала

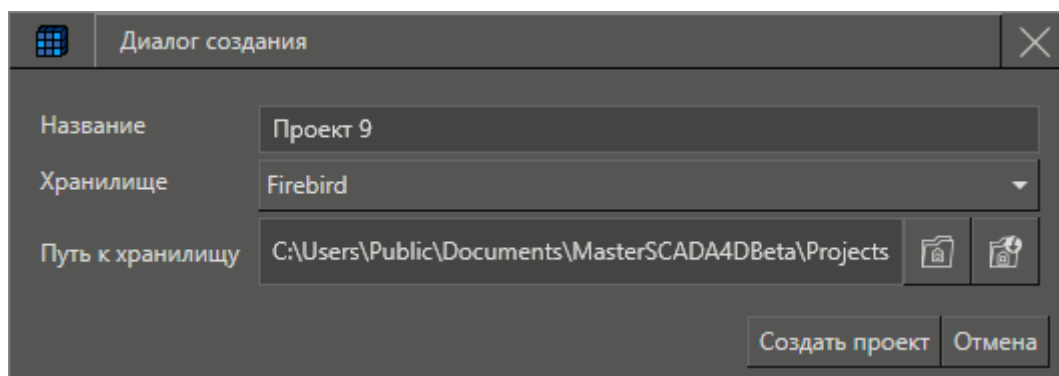
В версии MasterSCADA 4D 1.2.5 появилось два журнала для работы с сообщениями в режиме исполнения: архивный и текущий (оперативный). В более ранних версиях для работы с сообщениями был только один элемент, который работал со всеми сообщениями. Если проект был сделан в более ранних версиях, то при его открытии появится сообщение:



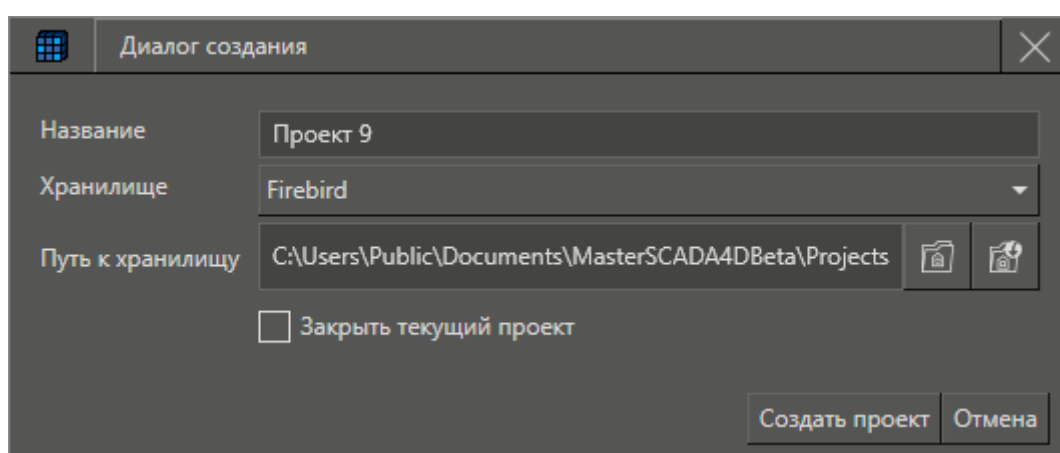
Необходимо выбрать нужное действие, следуя описанию, отображаемому в окне. Подробнее о различиях журналов читайте в разделе Журнал и архивный журнал.

5.11.1.2.1. ДИАЛОГ СОЗДАНИЯ ПРОЕКТА

Вид диалогового при открытии редактора:



Вид диалогового окна, если выполнен пункт Создать проект в меню Управления проектами:



5.11.1.3.

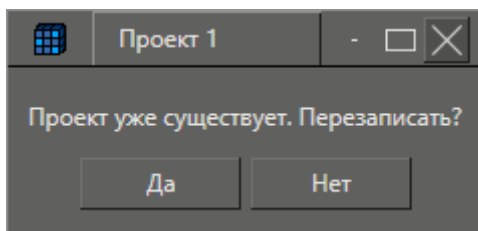
Описание элементов управления:

Название	Назначение
Название	Поле редактирования имени проекта. При создании для проекта выполняется автоматическая генерация имени <Проект> N.
Хранилище	В данной версии не поддерживается.
Путь к хранилищу	Задаёт место хранения проекта. По умолчанию, проект хранится в папке – C:\Users\Public\Documents\MasterSCADA4D<версия>\Projects.
Закреть текущий проект	Закрывает открытый ранее проект.
Отмена	Кнопка служит для отмены создания проекта. Диалог при этом закрывается.

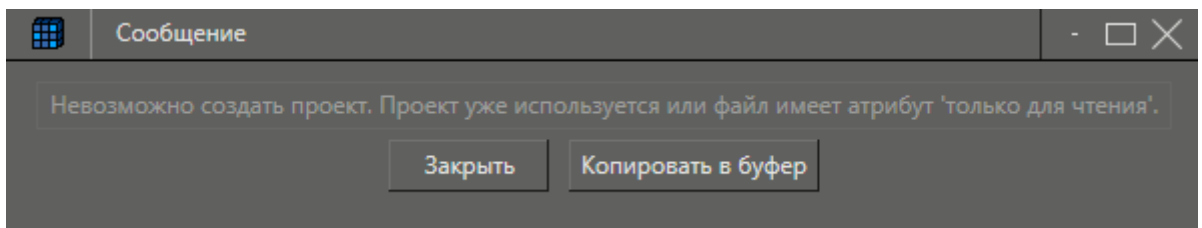
Создать проект	Создает проект в выбранном месте с указанным именем. В случае успешного создания проекта в хранилище создается папка проекта с заданным именем.
----------------	---

При создании проекта могут возникать следующие ситуации:

- если хранилище уже содержит проект с заданным именем, то на экране появляется соответствующий диалог подтверждения/отмены действия:



- если проект с заданными параметрами существует и открыт, то на экране появляется сообщение о невозможности создания проекта:



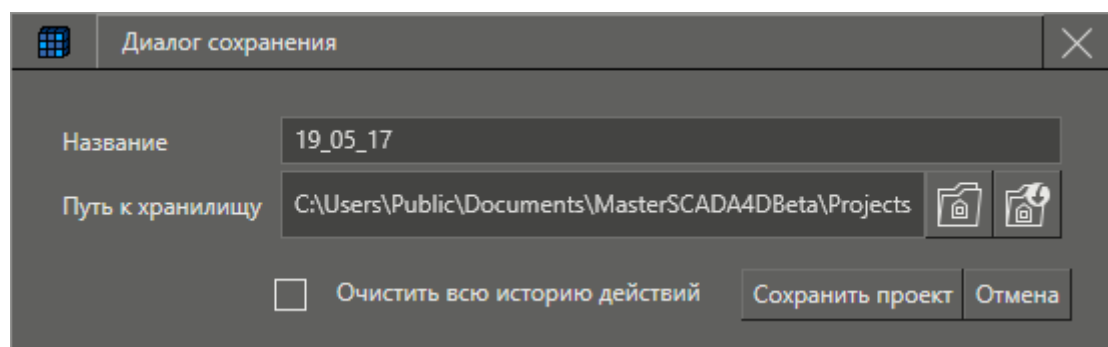
Созданный/перезаписанный проект открывается в новом окне редактора проекта, поэтому создание/перезапись проекта не приводит к закрытию проектов, открытых ранее.



Вновь созданный или перезаписанный проект необходимо сохранить. В противном случае, при закрытии редактора проекта проект удаляется (удаляется папка проекта).

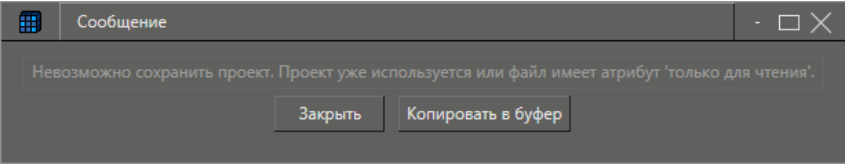
5.11.1.4. ДИАЛОГ СОХРАНЕНИЯ ПРОЕКТА

Если выполняется действие Сохранить как, то появляется диалоговое окно.

Вид окна:



Название	Описание
Название	Задается имя проекта
Путь к хранилищу	Задается путь к папке, в которой необходимо сохранить проект
	Кнопка позволяет сохранить проект в любом месте компьютера. После ее нажатия открывается диалоговое окно Windows для выбора места сохранения на диске.
	При нажатии на эту кнопку, в строке Путь к хранилищу автоматически укажется путь к рабочей папке среды разработки.
Очистить всю историю действий	Флаг позволяет сохранить проект без истории действий разработчика.
Сохранить проект	<p>Сохраняет проект с указанными выше настройками. При выполнении команды возможны следующие случаи:</p> <ul style="list-style-type: none"> • проекта с заданным именем не существует в указанной папке – проект сохраняется; • проект с заданным именем существует в указанной папке и открыт в текущем окне редактора – проект сохраняется; • проект с заданным именем существует в указанной папке, и не открыт ни в одном из окон редактора – открывается диалог, в котором нужно подтвердить (нажать на кнопку Да) или отменить (нажать на кнопку Нет) сохранение проекта (перезапись файла): <div data-bbox="734 1581 1209 1809" data-label="Image"> </div> <ul style="list-style-type: none"> • проект с заданным именем существует в указанной папке и открыт в другом окне редактора – появляется сообщение о невозможности сохранения проекта:

Название	Описание
	
Отмена	Закрывает диалог, отменяя сохранение проекта.

5.11.1.5. АВТОМАТИЧЕСКОЕ СОХРАНЕНИЕ ДЕЙСТВИЙ РАЗРАБОТЧИКА

Каждое действие, совершенное разработчиком, автоматически сохраняется (исключение составляет работа в редакторе отчетов). В случае, если компьютер вдруг непредвиденно выключится, то все изменения в проекте будут сохранены. Если при закрытии проекта в диалоговом окне на вопрос Сохранить изменения ответить Нет, то произойдет откат всех действий разработчика до момента последнего сохранения проекта.

Кроме того, разработчик проекта может вернуться к любому предыдущему состоянию проекта используя диалоговое окно Контроль версий.

5.11.1.6. ПАРАЛЛЕЛЬНАЯ РАЗРАБОТКА ПРОЕКТА

В Masterscada 4D предусмотрены различные возможности совместной работы над одним проектом несколькими разработчиками. Для этого в проекте изначально определяется структура, общая для всех разработчиков, например, дерево системы. Затем каждый из участников работает независимо от коллег, а по завершении работ все части проекта объединяются. Этот же подход можно использовать и при работе в одиночку в том случае, когда проект содержит множество типовых элементов и при этом существует необходимость отладить каждую часть независимо.

Пример 1

Рассмотрим пример, в котором участник Разработчик 1 разрабатывает объект Котельная, а участник Разработчик 2 - объект ГВС. При этом оба объекта должны работать на одном узле.

Шаг 1

Если на начальном этапе сразу можно сформировать дерево системы, то каждый участник берет копию проекта с деревом системы и разрабатывает только свой объект: создает окна, параметры, программы, настраивает связи между деревом системы и деревом объектов. Если объект состоит из множества повторяющихся частей, то разработчик также создает собственную библиотеку, содержащую все типовые пользовательские элементы.

Шаг 2

Затем каждый участник помещает свой, уже готовый, объект в собственную библиотеку и сохраняет ее.

Шаг 3

Библиотека переносится на тот компьютер, на котором находится результирующий проект.

Шаг 4

Каждый участник подключает свою библиотеку к результирующему проекту.

Шаг 5

После этого в дерево объектов необходимо вставить библиотечный экземпляр.

В результате выполнения этих действий появится диалог восстановления связей, позволяющий восстановить связи с деревом системы. Далее, необходимо настроить связи между объектами разных разработчиков, создать стартовое окно и приступить к совместной отладке проекта. В случае необходимости, следует внести изменения в объекты (каждый участник может это сделать на своем компьютере) и повторить Шаги 2-3. В завершение, необходимо выполнить обновление библиотеки в результирующем проекте. При этом экземпляры обновятся автоматически.

Пример 2

Предположим, что участник Разработчик 1 разрабатывает программы для всего проекта, а участник Разработчик 2 – окна.

В этом случае подход принципиально ничем не будет отличаться от метода, описанного в Примере 1. Разница лишь в том, что общей частью для обоих участников будет не только дерево системы, но и параметры дерева объектов. Библиотеки разработчиков будут содержать только окна и только программы.

Важно! При параллельной разработке единого проекта можно также воспользоваться возможностью копировать различные элементы (объекты, окна, протоколы и т.п.) из проектов разработчиков в результирующий проект (для этого все проекты должны быть открыты на одном компьютере). Однако, в этом случае, в дальнейшем одновременное внесение изменений и исправлений может быть затруднено, т.к. каждый раз будет необходимо заново копировать обновленные части проекта и все ранее установленные связи, а изменения, внесенные в объекты, пропадут в результирующем проекте.

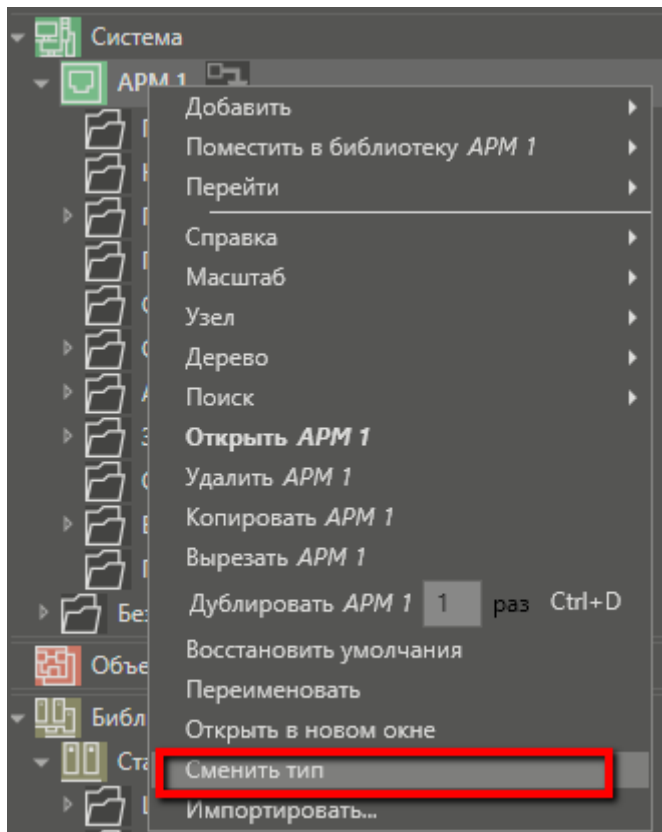
5.11.2. ОПЕРАЦИИ С ЭЛЕМЕНТАМИ

- Работа с именами элементов
- Выделение элементов
- Автоматическая прокрутка
- Перетаскивание с удержанием ПК

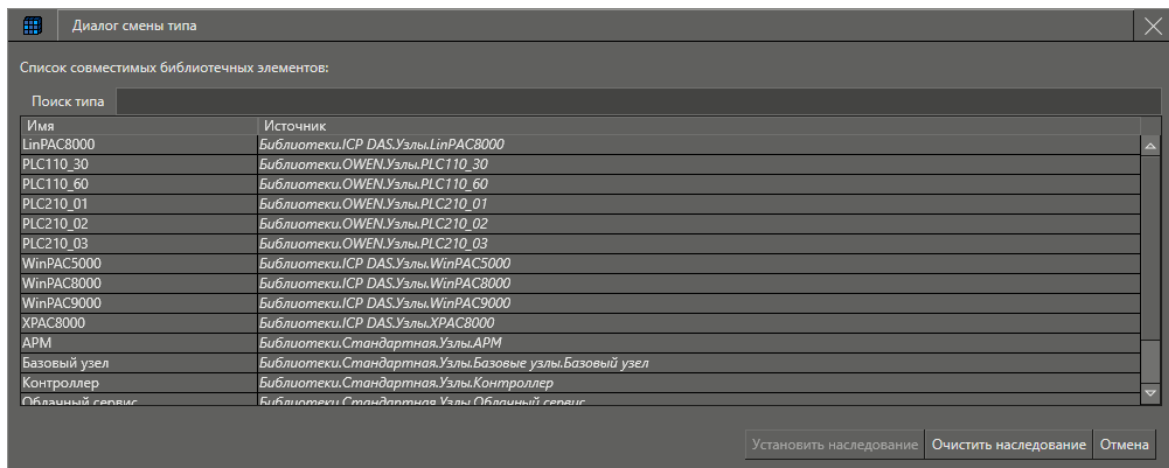
- Дублирование
- Перемещение элементов в дереве
- Поиск элементов

5.11.2.1. ДИАЛОГ УСТАНОВКИ НАСЛЕДОВАНИЯ

Контекстное меню объектов, окон и узлов содержит пункт Сменить тип:



После нажатия на данный пункт меню появится диалоговое окно, в котором можно будет выбрать допустимый тип:



В строке Поиск типа можно ввести символы, в итоге в списке останутся только те элементы, в которых совпадает последовательность введенных символов.

Кнопка Установить наследование установит наследование от выбранного в списке типа.

5.11.2.2. СОЗДАНИЕ ЭЛЕМЕНТОВ

Для создания компонента проекта используются следующие инструменты:

- команда 'Добавить' контекстного меню;
- перетаскивание из библиотеки;
- команды редактирования (см. Панель 'Редактирование');
- другие операции (см., например, Операции с переменными).

При создании компонента, как правило, выполняется автоматическая генерация имени (см. также Имена компонентов).

5.11.2.3. РАБОТА С ИМЕНАМИ ЭЛЕМЕНТОВ

Автоматическое создание имени

При добавлении нового элемента в некоторую группу, этому элементу автоматически присваивается имя $\langle \text{имя} \rangle N$, где

- $\langle \text{имя} \rangle$ – предопределенное имя для данного вида элемента (например, Параметр _ для входа или выхода программы);
- N – индекс (целое число), определяемое по следующим правилам:
- $N=1$, если группа не содержит элементов с подобным именем;
- $N=M+1$, если группа содержит элементы с подобным именами со старшим индексом M и не существует свободного индекса $k < M$;
- $N=k_{\min}$, если группа содержит элементы с подобным именами со старшим индексом M и существуют свободные индексы $k_i < M$ (где k_{\min} – минимальный индекс среди k_i).

Если выполняется операция копирования элемента с предопределенным именем с последующей операцией вставки, то индекс в имени созданного подобным образом элемента всегда больше индекса в имени скопированного элемента.

Произвольное имя элемента при создании

Имена элементов задаются так же, как идентификаторы .

Имя может быть задано с лидирующим или внутренними пробелами (не более одного пробела подряд). При редактировании имени в дереве автоматически обрезаются пробелы в конце имени, а несколько пробелов подряд в других местах имени заменяются одним пробелом.

Важно! Имена не могут начинаться с цифр.

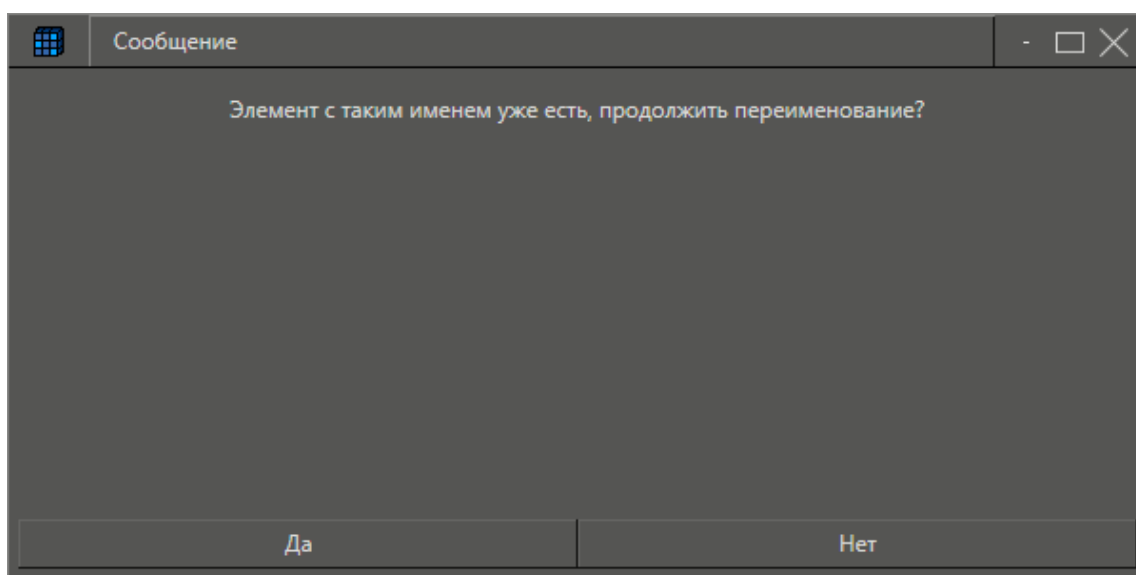
См. также имена переменных программного элемента.

Переименование элементов

Для переименования элемента предназначены также следующие инструменты/

- переход к редактированию;
- команда Переименовать контекстного меню элемента в дереве;
- панель Свойства .

При переименовании элементов появится ошибка, если уже есть элемент с этим именем на одном уровне иерархии дерева проекта:



Если нажать кнопку Да, то элемент будет переименован и в дереве проекта будет два одноименных элемента. Это допустимо, т.к. элементы будут иметь разные идентификаторы. Однако, это может затруднить дальнейшую разработку проекта.

5.11.2.4. ВЫДЕЛЕНИЕ ЭЛЕМЕНТОВ

Для выделения элементов в любом окне/редакторе, где эти элементы отображаются, могут использоваться такие типовые операции, как выделение , выделение рамкой и групповое выделение

Выделенный элемент подсвечивается (выделяется) во всех окнах/редакторах, где он отображается (см. Подсветка).

На FBD-, LD- и SFC-диаграммах могут быть выделены элементы только активных слоев (см. Слои диаграммы). Если в слое выделены элементы, то при деактивации этого слоя выделение снимается.

5.11.2.5. ПОДСВЕТКА

Подсветка – это автоматическое изменение цвета фона элемента для индикации какого-либо свойства или состояния этого элемента (см. также Подсветка на мнемосхеме). Используются следующие цвета, в темной цветовой схеме редактора:

Подсветка в деревьях и панелях

Если на элемент установлен курсор, то фон под элементом светлее, чем в других цветом (на FBD-, LD- и SFC-диаграммах подсвечиваются только элементы активных слоев):



если компонент выделен, он подсвечивается синим цветом (операции выделения описаны в разделе Типовые операции):



если на компонент А установлен курсор, то компонент В подсвечивается серым цветом в том случае, если между А и В существуют отношения , и подсветка вида отношений, которые существуют между А и В, разрешена (см. Вкладка 'Внешний вид'):



5.11.2.6. АВТОМАТИЧЕСКИЙ ПЕРЕХОД

Автоматический переход к компоненту выполняется в открытом разделе окна структуры проекта и включает следующие автоматически выполняемые процедуры:

- необходимое раскрытие дерева и прокрутка таким образом, чтобы компонент был виден;
- выделение компонента.

Автоматический переход выполняется, например, при поиске (см. Панель 'Поиск' и Окно результатов поиска).

5.11.2.7. АВТОМАТИЧЕСКАЯ ПРОКРУТКА

При переносе элемента за пределы видимой части дерева или редактора (например, редактора FBD или редактора HMI) выполняется автоматическая прокрутка содержимого дерева/редактора.

5.11.2.8. ПЕРЕТАСКИВАНИЕ С УДЕРЖАНИЕМ ПК

В редакторе проекта поддерживается как обычное перетаскивание, так и перетаскивание с удержанием ПК.

При перетаскивании элемента на элемент с удержанием ПК открывается меню доступных операций, в случае если такие операции существуют. В противном случае, перетаскивании элемента на элемент с удержанием ПК работает так же, как обычное перетаскивание.

Если при обычном перетаскивании (см. Перемещение элементов в дереве) выполняется перемещение элементов, то при перетаскивании этих же элементов с удержанием ПК открывается меню доступных операций (которые могут быть объединены в подменю – например, Другие операции и Связать).

Ниже приведены возможные команды меню при перетаскивании элементов с удержанием ПК.

Перетаскивание переменных на окно с удержанием ПК описано в разделе Перетаскивание переменной на окно.

Переместить перед

Переместить перетаскиваемый элемент, расположив его в дереве выше элемента, на котором завершено перетаскивание.

Копировать перед

Вставить копию перетаскиваемого элемента выше элемента, на котором завершено перетаскивание.

Копировать экземпляр перед

Вставить экземпляр перетаскиваемого элемента выше элемента, на котором завершено перетаскивание.

Связать с элементом

Установить связь (передачу значения) между элементами.

Создать связанный параметр

Данная команда доступна при перетаскивании с удержанием ПК параметра или папки параметров объекта на объект.

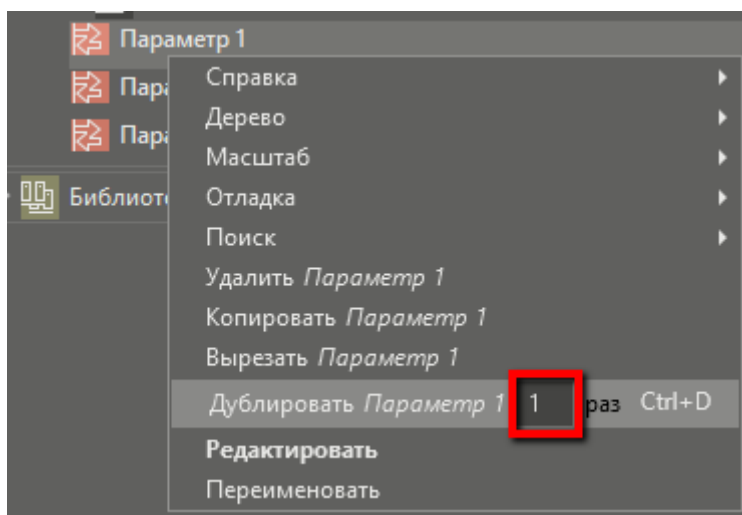
По этой команде в объекте создается соответственно параметр или папка параметров, связанных с исходными параметрами. У создаваемого параметра свойство Доступ имеет противоположное значение относительно исходного параметра.

5.11.2.9. ДУБЛИРОВАНИЕ И КОПИРОВАНИЕ

Дублирование

Для дублирования одного или нескольких выделенных элементов в том же родительском элементе существуют следующие способы:

- комбинация клавиш клавиатуры CTRL+D - создается одна копия элемента
- пункт контекстного меню Дублировать - по умолчанию создается одна копия, однако, можно задать количество копий, которое необходимо сделать:



Копирование

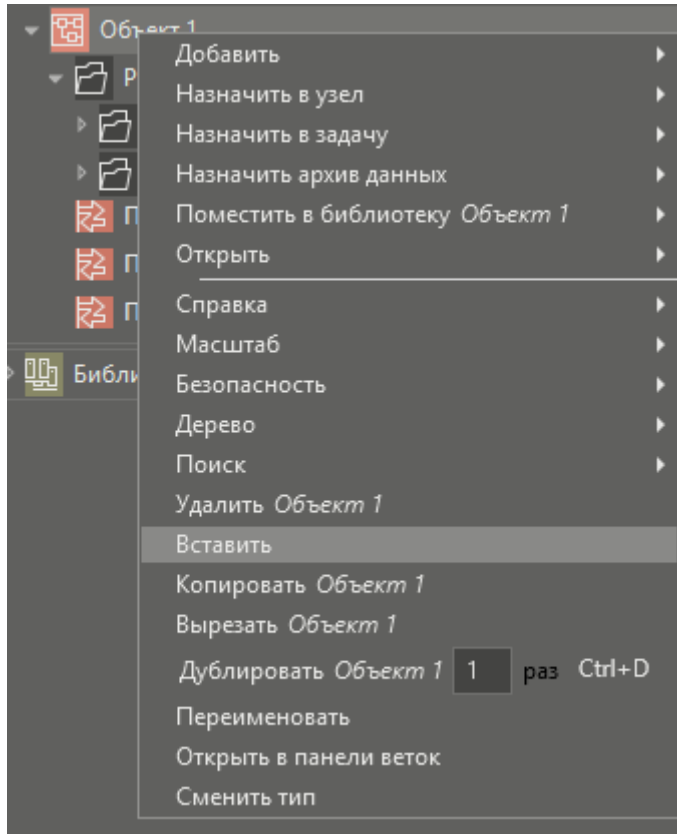
Для копирования одного или нескольких выделенных элементов в буфер обмена существуют следующие способы:

- комбинация клавиш клавиатуры CTRL+C;
- пункт контекстного меню Копировать.

Вставка

Для вставки в проект ранее скопированного или вырезанного в буфер обмена элемента, необходимо выделить родительский элемент и выполнить одно из следующих действий:

- нажать комбинацию клавиш клавиатуры CTRL+V;
- выполнить пункт контекстного меню Вставить;



При вставке и дублировании элементов при наличии внешних связей у скопированного (дублированного) элемента, появится диалоговое окно Восстановление связей

Важно! Поддерживается копирование в одном открытом проекте и вставка в другой проект.

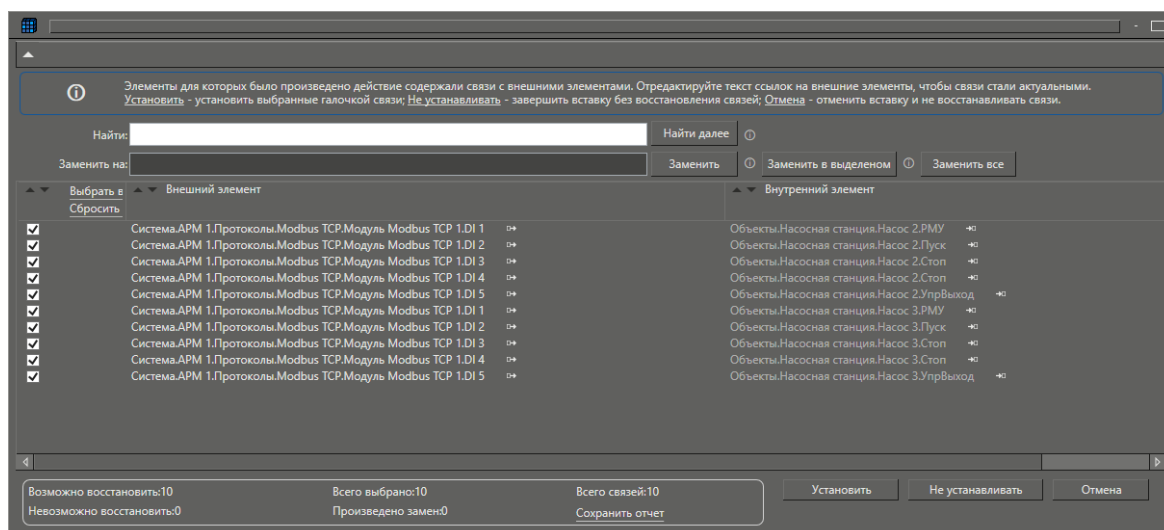
5.11.2.10. ВОССТАНОВЛЕНИЕ УМОЛЧАНИЙ


Enter topic text here.

5.11.2.11. ВОССТАНОВЛЕНИЕ СВЯЗЕЙ

При дублировании/вставки элементов, содержащих внешние связи появится диалоговое окно, в котором можно восстановить, переопределить или вовсе удалить связь.


Вид окна:



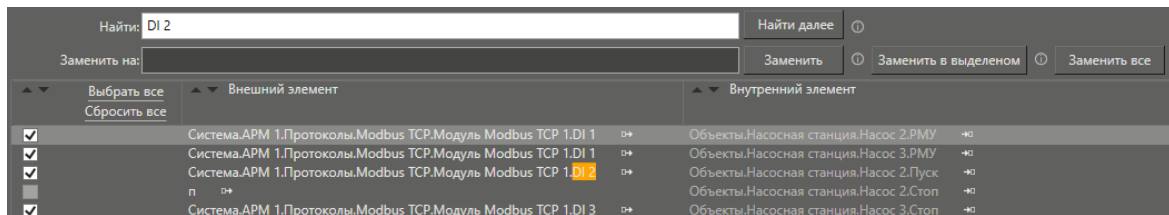
В верхней части окна располагается информационное поле, в котором отражены основные моменты работы с окном. При необходимости это поле можно скрыть, нажав на кнопку 

Список всех связей отображается в таблице, в средней части окна. Таблица состоит из трех столбцов:

- В первом столбце определяется необходимость установки связи между внутренним и внешним элементом. Если флаг установлен, то связь может быть установлена, если флаг снят, то связь не будет установлена. Если флаг дизаблирован, то это значит связь с элементом не может быть установлена. В заголовке столбца можно снять/установить выделение для всех строк таблицы.
- Во втором столбце указывается полное имя внешнего элемента, с кем предполагается установить связь. Элементы этого столбца можно отредактировать. Для начала редактирования нужно кликнуть мышью в ячейку. Если выполнить один клик, то ввод данных будет с начала строки. Если выполнить два клика, то вся строка будет выделена, если кликнуть три раза, то установится курсор в месте кликов. После окончания редактирования производится проверка возможности установить связь. Если связь установить нельзя, то флаг, в первом столбце будет дизаблирован.
- В третьем столбце отображается полное имя параметров элементов, которые получатся в результате копирования/дублирования (внутренние элементы). Данный столбец отредактировать нельзя.

Строки в таблице можно отсортировать, по любому столбцу, для этого нужно нажать в заголовке столбца одну из кнопок 

После ввода какого-либо значения в поле Найти автоматически будут подсвечены оранжевым цветом все элементы в столбце Внешний элемент, которые в своем имени будут иметь введенную последовательность символов:



Кнопка Найти далее или клавиша клавиатуры TAB переведет в режим редактирования строку, следующую по списку после выделенной.

Автоматически заменить найденные символы можно используя поле Заменить на. Существует три варианта замены:



- Заменить только текущую строку, нажав для этого кнопку Заменить или клавишу клавиатуры Enter.
- Выделить несколько строк (можно использовать мышь и клавишу SHIFT для клавиатуре для последовательного выделения, либо мышь и клавишу CTRL, для выборочного выделения) и заменить только в выделенном, нажав кнопку Заменить в выделенном.
- Заменить все встречающиеся символы во всей таблице можно нажав на кнопку Заменить все.

В нижней части отобразится статистическая информация о том, сколько строк было изменено, сколько выбрано и т.п. При необходимости можно сохранить отчет в текстовом файле.

Связи установятся, только после нажатия кнопки Установить. Связи не будут созданы, если нажать кнопку Отмена или Не устанавливать.

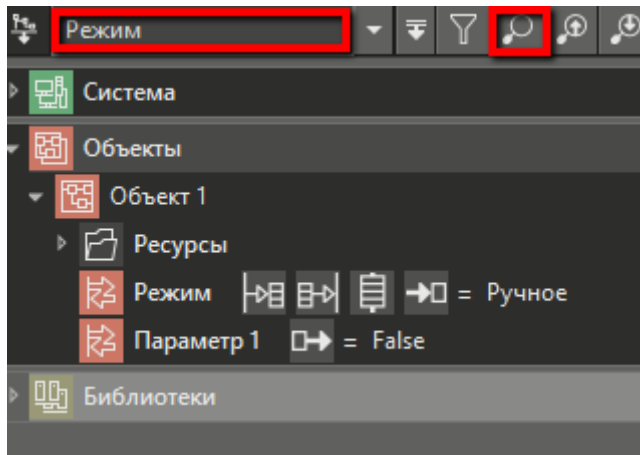
5.11.2.12. ПЕРЕМЕЩЕНИЕ ЭЛЕМЕНТОВ В ДЕРЕВЕ

Для перемещения элемента (выделенной группы элементов) в дереве могут быть использованы следующие инструменты (см. также Перетаскивание с удержанием ПК):

- Команда контекстного меню Дерево.Вверх по дереву (CTRL+ ) и команда Дерево.Вниз по дереву (CTRL+ ) контекстного меню элемента;
- перетаскивание элемента (выделенной группы элементов) на элемент А той же группы – перемещение элемента (выделенной группы элементов) ниже элемента А;
- перетаскивание элемента (выделенной группы элементов) на группу, которая может содержать подобные элементы.

5.11.2.13. ПОИСК ЭЛЕМЕНТОВ

Чтобы найти элемент во всем проекте необходимо в панели инструментов дерева проектов ввести символы, которые могут встречаться в любом месте описания элемента проекта и нажать на кнопку поиска:



Откроется окно, в котором в табличном виде будут отображаться все найденные места, где встречается указанный набор символов:

Имя	Тип	Свойство	Совпадения	Опис	Полное имя
Выпадающий список 1.Выбор-Объекты.Режим	Связь	Имя	ВЫПАДАЮЩИЙ СПИСОК 1.ВЫБОР	Объекты.Объект 1.Ресурсы.Окна.Окно 1.Связи.Выпадающий список	Объекты.Объект 1.Ресурсы.Окна.Окно 1.Связи.Выпадающий список
Выпадающий список 1.Выбор-Объекты.Режим	Связь	Программное имя	ELEMENTS_ВЫПАДАЮЩИЙ СПИСОК	Объекты.Объект 1.Ресурсы.Окна.Окно 1.Связи.Выпадающий список	Объекты.Объект 1.Ресурсы.Окна.Окно 1.Связи.Выпадающий список
Объекты.Режим-Выпадающий список 1.Выбор	Связь	Имя	ОБЪЕКТЫ.РЕЖИМ-ВЫПАДАЮЩИЙ	Объекты.Объект 1.Ресурсы.Окна.Окно 1.Связи.Объекты.Режим-В	Объекты.Объект 1.Ресурсы.Окна.Окно 1.Связи.Объекты.Режим-В
Объекты.Режим-Выпадающий список 1.Выбор	Связь	Программное имя	L..._РЕЖИМ\$ELEMENTS_ВЫПАДА	Объекты.Объект 1.Ресурсы.Окна.Окно 1.Связи.Объекты.Режим-В	Объекты.Объект 1.Ресурсы.Окна.Окно 1.Связи.Объекты.Режим-В
Режим	Параметр	Имя	РЕЖИМ	Объекты.Объект 1.Ресурсы.Программы.Программа_ST_1.Параметр	Объекты.Объект 1.Ресурсы.Программы.Программа_ST_1.Параметр
Режим	Параметр	Программное имя	РЕЖИМ	Объекты.Объект 1.Ресурсы.Программы.Программа_ST_1.Параметр	Объекты.Объект 1.Ресурсы.Программы.Программа_ST_1.Параметр
Режим	Параметр	Имя	РЕЖИМ	Объекты.Объект 1.Ресурсы.Программы.Программа_ST_1.Параметр	Объекты.Объект 1.Ресурсы.Программы.Программа_ST_1.Параметр
Режим	Параметр	Программное имя	РЕЖИМ	Объекты.Объект 1.Ресурсы.Программы.Программа_ST_1.Параметр	Объекты.Объект 1.Ресурсы.Программы.Программа_ST_1.Параметр
Режим безопасности сообщений	Системная настройка	Имя	РЕЖИМ БЕЗОПАСНОСТИ СООБЩЕ	Библиотеки.Стандартная.Протоколы.OPC UA.Настройки.Режим бе	Библиотеки.Стандартная.Протоколы.OPC UA.Настройки.Режим бе
Режим чтения архивов	Системная настройка	Имя	РЕЖИМ ЧТЕНИЯ АРХИВОВ	Библиотеки.Стандартная.Протоколы.OPC UA.Настройки.Режим чт	Библиотеки.Стандартная.Протоколы.OPC UA.Настройки.Режим чт
Полноэкранный режим	Параметр	Имя	ПОЛНОЭКРАННЫЙ РЕЖИМ	Библиотеки.НМЛ.Окна.Контроль.Журнал.Параметры.Полноэкранный	Библиотеки.НМЛ.Окна.Контроль.Журнал.Параметры.Полноэкранный
Полноэкранный режим	Параметр	Имя	ПОЛНОЭКРАННЫЙ РЕЖИМ	Библиотеки.НМЛ.Окна.Контроль.Журнал.Параметры.Полноэкранный	Библиотеки.НМЛ.Окна.Контроль.Журнал.Параметры.Полноэкранный
Режим работы	Параметр	Имя	РЕЖИМ РАБОТЫ	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f
Режим работы	Параметр	Программное имя	РЕЖИМ РАБОТЫ	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f
Режим работы	Параметр	Имя	РЕЖИМ РАБОТЫ	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f
Режим работы	Параметр	Программное имя	РЕЖИМ РАБОТЫ	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f
Режим работы	Параметр	Имя	РЕЖИМ РАБОТЫ	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f
Режим работы	Параметр	Программное имя	РЕЖИМ РАБОТЫ	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f
Режим работы	Параметр	Имя	РЕЖИМ РАБОТЫ	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f
Режим работы	Параметр	Программное имя	РЕЖИМ РАБОТЫ	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f
Режим работы	Параметр	Имя	РЕЖИМ РАБОТЫ	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f
Режим работы	Параметр	Программное имя	РЕЖИМ РАБОТЫ	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f
Режим работы	Параметр	Имя	РЕЖИМ РАБОТЫ	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f
Режим работы	Параметр	Программное имя	РЕЖИМ РАБОТЫ	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f
Режим работы	Параметр	Имя	РЕЖИМ РАБОТЫ	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f
Режим работы	Параметр	Программное имя	РЕЖИМ РАБОТЫ	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f
Режим работы	Параметр	Имя	РЕЖИМ РАБОТЫ	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f
Режим работы	Параметр	Программное имя	РЕЖИМ РАБОТЫ	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f	Библиотеки.BaseObjects.Окна.Горелка.Горелка анфас.Параметры.f
Режим управления	Параметр	Имя	РЕЖИМ УПРАВЛЕНИЯ	Библиотеки.BaseObjects.Окна.Панель управления.Панель управле	Библиотеки.BaseObjects.Окна.Панель управления.Панель управле
Режим управления	Параметр	Программное имя	РЕЖИМ УПРАВЛЕНИЯ	Библиотеки.BaseObjects.Окна.Панель управления.Панель управле	Библиотеки.BaseObjects.Окна.Панель управления.Панель управле
Панель управления воздушнойкой и насосом.Ре	Связь	Имя	ПАНЕЛЬ УПРАВЛЕНИЯ ВОЗДУХОДУ	Библиотеки.BaseObjects.Окна.Панель управления.Панель управле	Библиотеки.BaseObjects.Окна.Панель управления.Панель управле

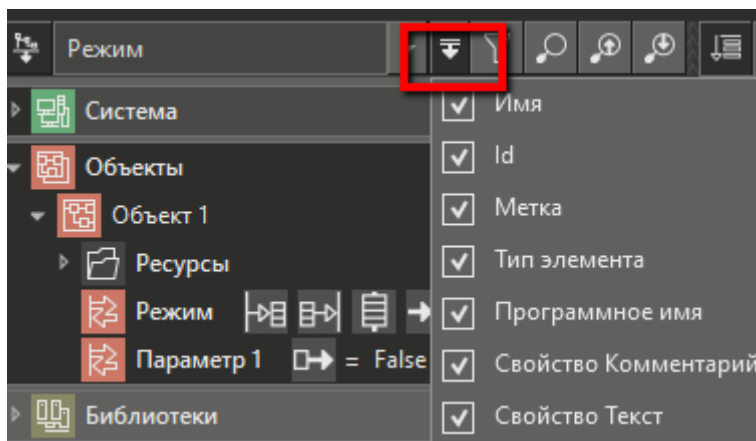
Название	Описание
Имя	Имя элемента проекта, которое в своих настройках содержит искомые символы
Тип	Тип найденного элемента проекта
Свойство	В каком свойстве найденного элемента встречаются искомые символы

Название	Описание
Совпадения	Полное имя свойства, в котором встречается совпадение
Описание	Дополнительное описание, как правило не встречается
Полное имя	Отображается Полное имя найденного элемента. Если в Полном имени указывается Окно или Программа, то при переходе к этому элементу будет открываться соответствующий редактор

Перейти к найденному элементу в дереве и открыть его панель свойств можно двумя способами:

- Выполнив двойной клик по строке;
- Вызвать контекстное меню строки, правой кнопкой мыши и выбрав пункт Перейти к элементу.

Если необходимо изменить область поиска, то необходимо выбрать нужные источники в выпадающем списке:



5.11.3. ЛОКАЛИЗАЦИЯ ПРОЕКТА

Для перевода проекта, разработанного в некоторой локализации, в другую локализацию нужно выполнить следующие действия:

- сменить язык редактора проекта (см. Язык среды);
- открыть проект в редакторе проекта;

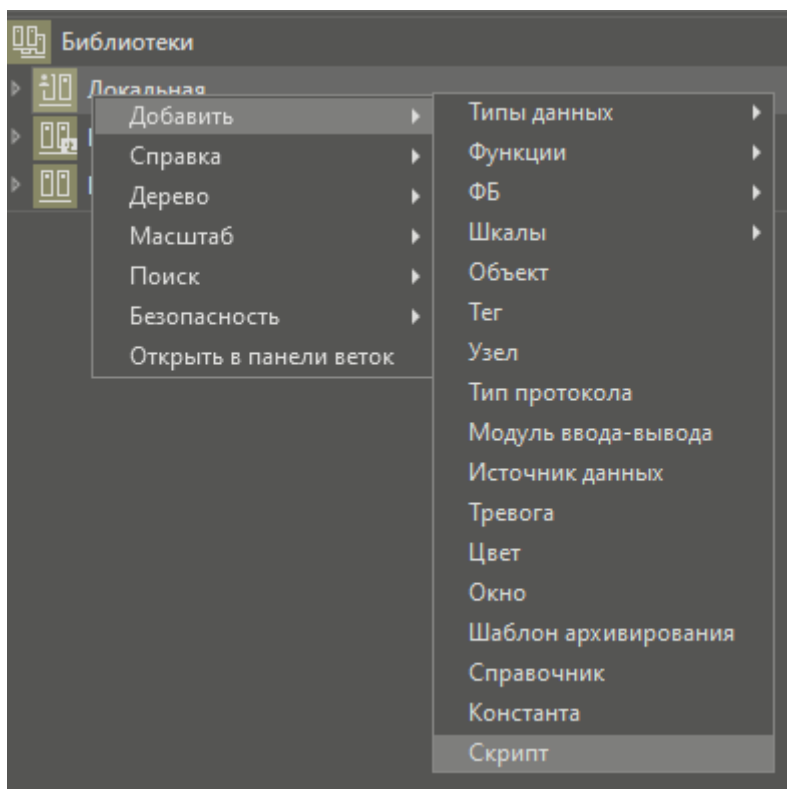
- для каждого локализуемого параметра (свойство 'Локализуемое' (Localizable) = TRUE) каждого пользовательского объекта проекта в окне свойств (см. Окно 'Свойства') задать свойство 'Начальное значение' (Initial value):

External library element	
Full name	System.Controller 1.Resources.Windows.W
Initial value	Quick Start. Example 1. Part 4
Initial value.ru	Быстрый старт. Пример 1. Часть 4
Labels	
Localizable	True
Main property	True
Name	Text
Name.ru	

Начальное значение параметра для русской локализации (Initial value.ru) отображается в том случае, если установлен флаг Служебные свойства .

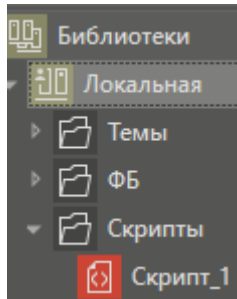
5.11.4. СКРИПТ С#

Для автоматизации разработки проекта можно использовать библиотечный элемент Скрипт:



Назначение пользовательских скриптов – операции с проектами

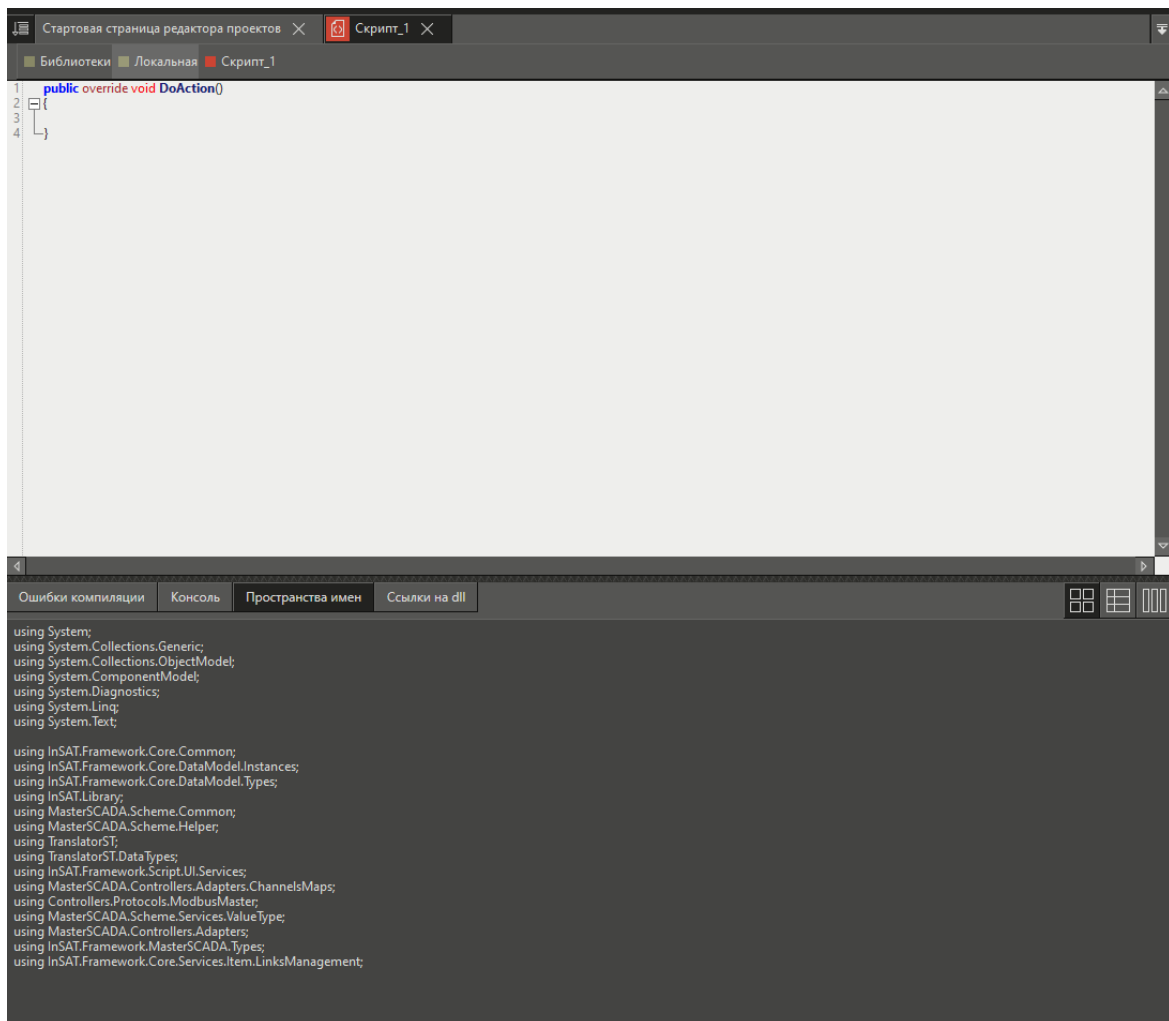
После добавления элемента в библиотеку появится группа Скрипты, которая будет содержать добавленный элемент:



Редактор, предназначенный для редактирования данного элемента откроется автоматически.

Важно! Скрипты C# можно использовать только для создания проектов в среде разработки. В среде исполнения скрипты C# не поддерживаются. Для работы в режиме исполнения нужно использовать элементы Программа C# или ФБ C#

Вид редактора скриптов:



Редактор C# интегрирован из SharpDevelop.



Свойства редактора:



- подсказчик (intellisense);
- подсветка синтаксиса, отступы;
- перетаскивание узлов в редактор;
- палитра базовых методов скрипта;
- автоматическое добавление нужного кода, привязка всех библиотек редактора и автоматическое добавление всех using, чтобы минимизировать усилия пользователя для разработки скрипта, то есть скрипт автоматически видит все области имен редактора;
- до тех пор, пока не начато редактирование скрипта, – вывод текста с подсказкой с чего начать.

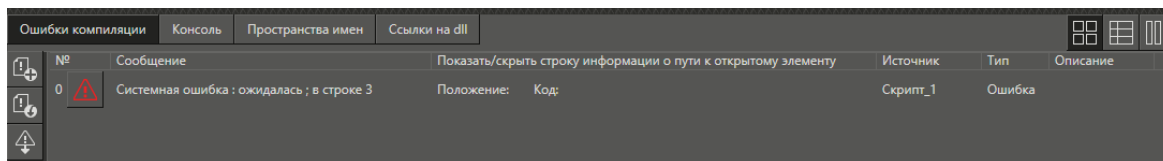
Редактор C# снабжен панелью инструментов Проверить:



Данная панель содержит следующие инструменты:

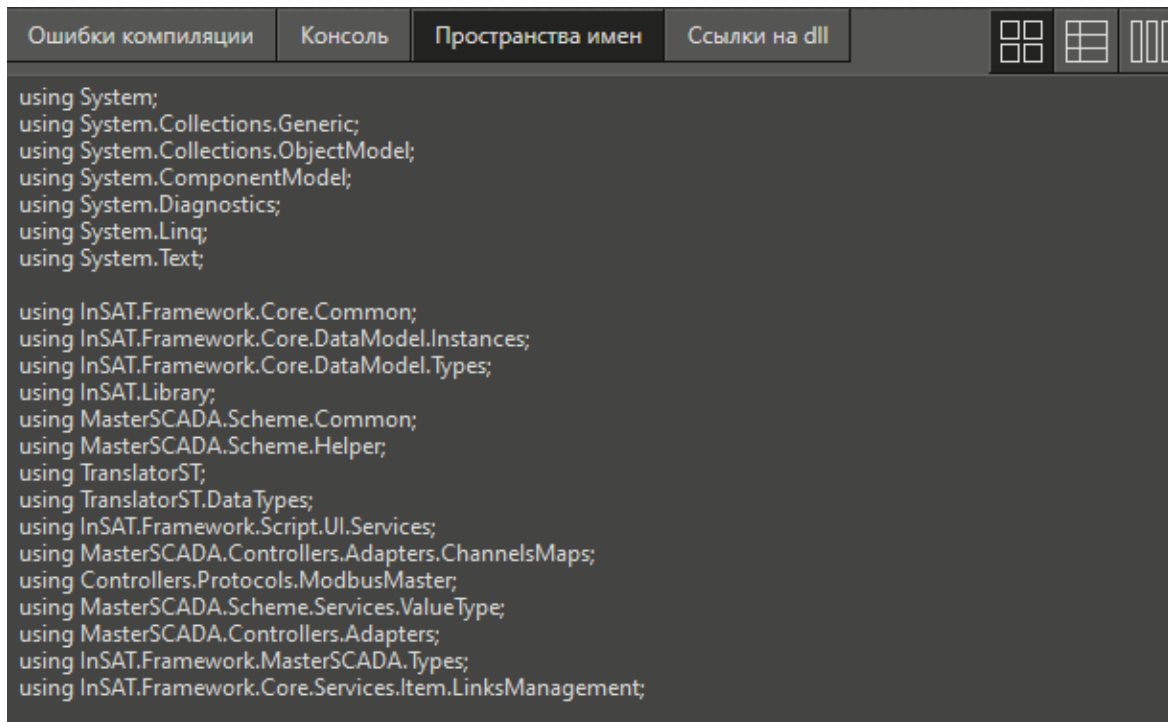
-  – запускает компиляцию и проверку скрипта;
-  – запускает компиляцию и выполнение скрипта.

Если скрипт содержит ошибки, при выполнении любой из команд инструмент  меняет свой цвет на красный (), и информация об ошибках выводится в панель Ошибки компиляции:



Легенда содержит также следующие вкладки:

- Консоль – сообщения об исполнении скрипта;
- Пространства имен – пространства имен;



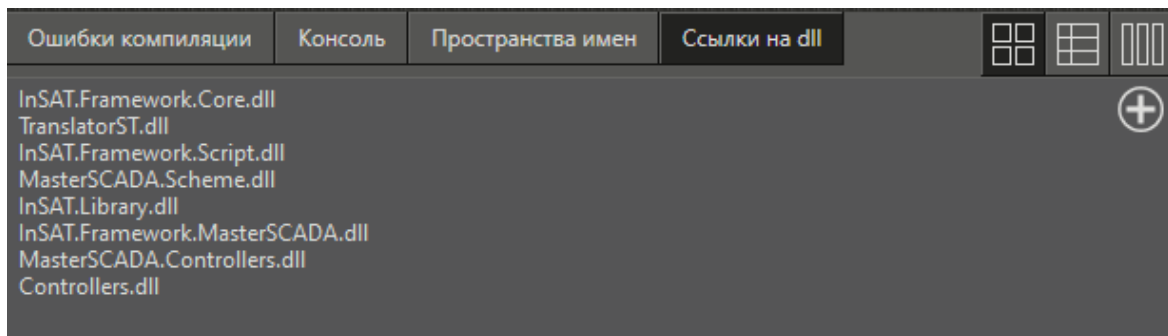
```

using System;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.ComponentModel;
using System.Diagnostics;
using System.Linq;
using System.Text;

using InSAT.Framework.Core.Common;
using InSAT.Framework.Core.DataModel.Instances;
using InSAT.Framework.Core.DataModel.Types;
using InSAT.Library;
using MasterSCADA.Scheme.Common;
using MasterSCADA.Scheme.Helper;
using TranslatorST;
using TranslatorST.DataTypes;
using InSAT.Framework.Script.UI.Services;
using MasterSCADA.Controllers.Adapters.ChannelsMaps;
using Controllers.Protocols.ModbusMaster;
using MasterSCADA.Scheme.Services.ValueType;
using MasterSCADA.Controllers.Adapters;
using InSAT.Framework.MasterSCADA.Types;
using InSAT.Framework.Core.Services.Item.LinksManagement;

```


- Ссылки на dll – используемые библиотеки:



```

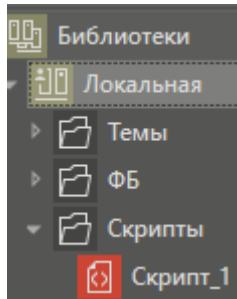
InSAT.Framework.Core.dll
TranslatorST.dll
InSAT.Framework.Script.dll
MasterSCADA.Scheme.dll
InSAT.Library.dll
InSAT.Framework.MasterSCADA.dll
MasterSCADA.Controllers.dll
Controllers.dll

```

Если нажать на кнопку , то откроется диалоговое окно Windows, в котором можно выбрать пользовательскую произвольную библиотеку, при этом она будет скопирована в папку проекта *[имя проекта]\dll_dt*. При компиляции скрипта - автоматически загрузятся в виде assembly, то есть из скрипта можно ссылаться на классы, описанные в них (пользовательских dll).

5.11.4.1. ПРИМЕР ИСПОЛЬЗОВАНИЯ СКРИПТА C#

Создаем новый проект. В библиотеке Локальная создаем компонент Скрипт:



При создании скрипт автоматически открывается в редакторе. Задаем для него следующий код:

```
public override void DoAction()
{
    // Обращение к библиотечному типу АРМ
    var libItem = LibraryRoot.FindChildByFullDisplayName("Стандартная.Узлы.АРМ");
    // Добавление экземпляра АРМа
    var arm = SystemRoot.ManagementService.InsertInheritedCopy(libItem);
    // Добавление Параметра 1
    var param1 = arm.FindChildByDisplayName("Параметры").ManagementService.InsertChildDef(
        SchemeTypes.Attribute);
    // Добавление Параметра 2
    var param2 = arm.FindChildByDisplayName("Параметры").ManagementService.InsertChildDef(
        SchemeTypes.Attribute);
    // Переименование Параметра 1
    param1.DisplayName = "Мой параметр_1";
    // Переименование Параметра 2
    param2.DisplayName = "Мой параметр_2";
    // Обращение к сервису типов для Параметра 1
    var paramService1 = param1.GetService<ValueTypeService>();
    // Обращение к сервису типов для Параметра 2
```

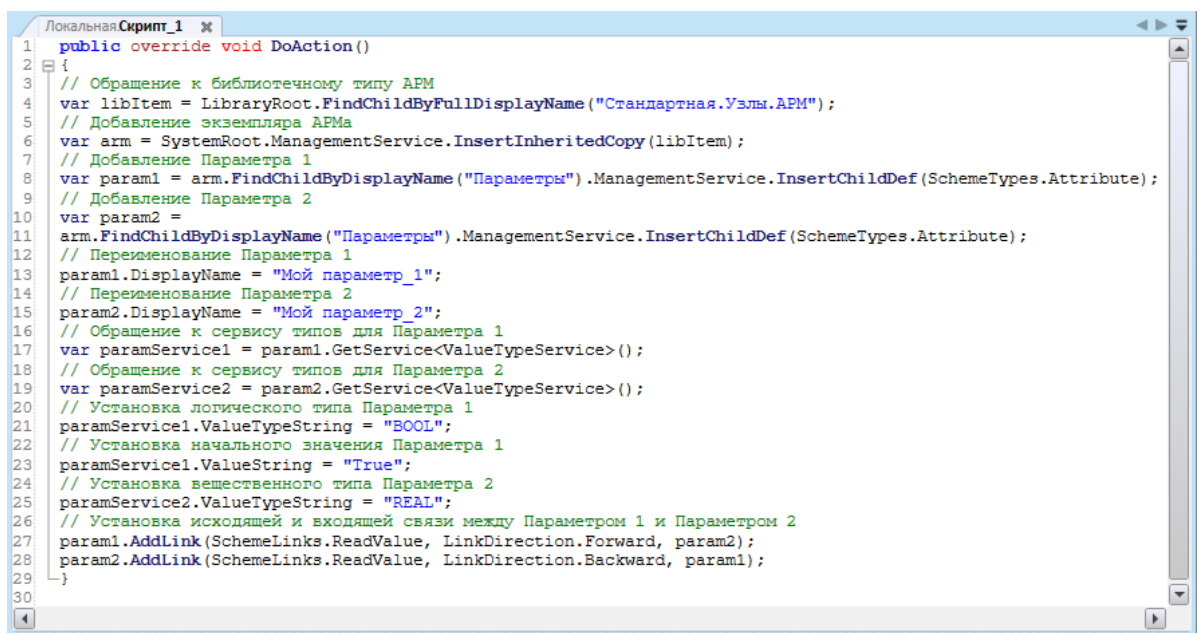
```
var paramService2 = param2.GetService<ValueTypeService>();

// Установка логического типа Параметра 1
paramService1.ValueTypeString = "BOOL";

// Установка начального значения Параметра 1
paramService1.ValueString = "True";

// Установка вещественного типа Параметра 2
paramService2.ValueTypeString = "REAL";

// Установка исходящей и входящей связи между Параметром 1 и Параметром 2
param1.AddLink(SchemeLinks.ReadValue, LinkDirection.Forward, param2);
param2.AddLink(SchemeLinks.ReadValue, LinkDirection.Backward, param1);
}
```



```
Локальный скрипт_1
1 public override void DoAction()
2 {
3     // Обращение к библиотечному типу АРМ
4     var libItem = LibraryRoot.FindChildByFullName("Стандартная.Узлы.АРМ");
5     // Добавление экземпляра АРМа
6     var arm = SystemRoot.ManagementService.InsertInheritedCopy(libItem);
7     // Добавление Параметра 1
8     var param1 = arm.FindChildByDisplayName("Параметры").ManagementService.InsertChildDef(SchemeTypes.Attribute);
9     // Добавление Параметра 2
10    var param2 =
11    arm.FindChildByDisplayName("Параметры").ManagementService.InsertChildDef(SchemeTypes.Attribute);
12    // Переименование Параметра 1
13    param1.DisplayName = "Мой параметр_1";
14    // Переименование Параметра 2
15    param2.DisplayName = "Мой параметр_2";
16    // Обращение к сервису типов для Параметра 1
17    var paramService1 = param1.GetService<ValueTypeService>();
18    // Обращение к сервису типов для Параметра 2
19    var paramService2 = param2.GetService<ValueTypeService>();
20    // Установка логического типа Параметра 1
21    paramService1.ValueTypeString = "BOOL";
22    // Установка начального значения Параметра 1
23    paramService1.ValueString = "True";
24    // Установка вещественного типа Параметра 2
25    paramService2.ValueTypeString = "REAL";
26    // Установка исходящей и входящей связи между Параметром 1 и Параметром 2
27    param1.AddLink(SchemeLinks.ReadValue, LinkDirection.Forward, param2);
28    param2.AddLink(SchemeLinks.ReadValue, LinkDirection.Backward, param1);
29 }
30
```

Добавляем на вкладку Пространства имен следующую директиву:

```
using MasterSCADA.Scheme.Services.ValueType;
```

Вкладка примет следующий вид:

```
using System;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.ComponentModel;
using System.Diagnostics;
using System.Linq;
using System.Text;

using InSAT.Framework.Core.Common;
using InSAT.Framework.Core.DataModel.Instances;
using InSAT.Framework.Core.DataModel.Types;
using InSAT.Library;
using MasterSCADA.Scheme.Common;
using MasterSCADA.Scheme.Helper;
using TranslatorST;
using TranslatorST.DataTypes;
using InSAT.Framework.Script.UI.Services;

using MasterSCADA.Scheme.Services.ValueType;
```

Ошибки Консоль Пространства имен Ссылки на dll

На вкладку Ссылки на dll добавляем следующие библиотеки:

MasterSCADA.Controllers.RT.dll

Controllers.dll


Вкладка примет следующий вид:

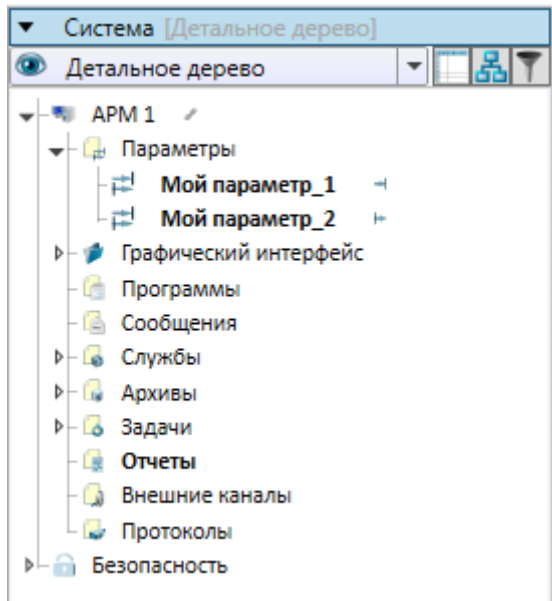
```
InSAT.Framework.Core.dll
TranslatorST.dll
InSAT.Framework.Script.UI.dll
MasterSCADA.Scheme.dll
InSAT.Library.dll

MasterSCADA.Controllers.RT.dll
Controllers.dll
```

Ошибки Консоль Пространства имен Ссылки на dll

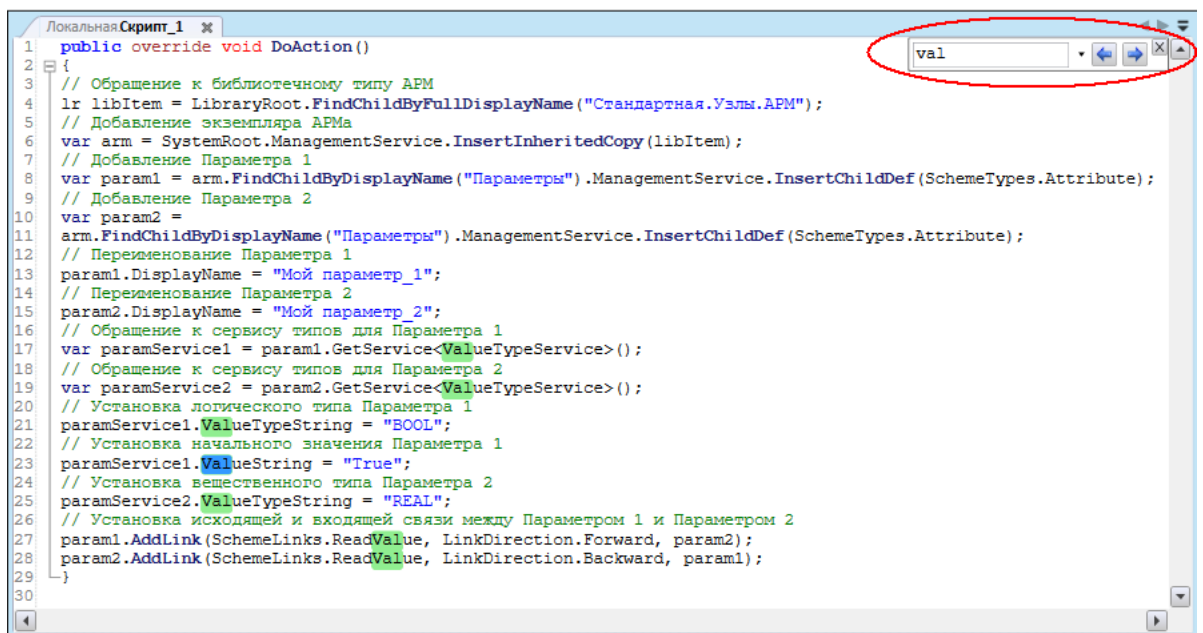


На панели Проверить щелкаем кнопку  для исполнения скрипта. При каждом исполнении скрипта в корневой группе Система создается узел АРМ<n>, а в нем – заданные параметры:



5.11.4.2. ПОИСК И ЗАМЕНА В ТЕКСТЕ C#

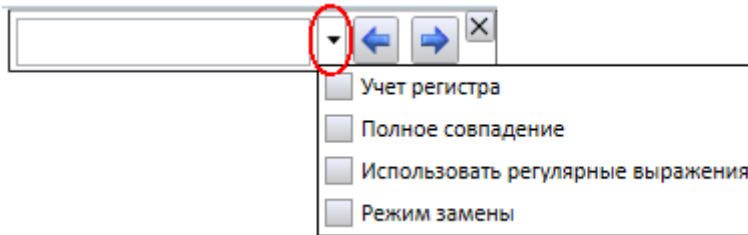
При нажатии сочетания клавиш CTRL+F в редакторе C# открывается панель поиска и замены:



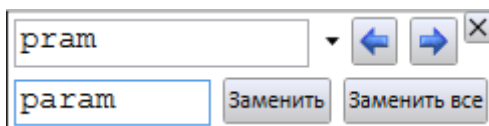
Подстрока для поиска задается в окне панели, при этом все найденные вхождения заданной подстроки подсвечиваются в листинге программы зеленым цветом, а текущее (выделенное) вхождение – синим.



Поиск можно сконфигурировать также следующим образом: выделить подстроку в тексте программы и затем нажать CTRL+F.


При нажатии кнопки  открывается меню, содержащее следующие опции:



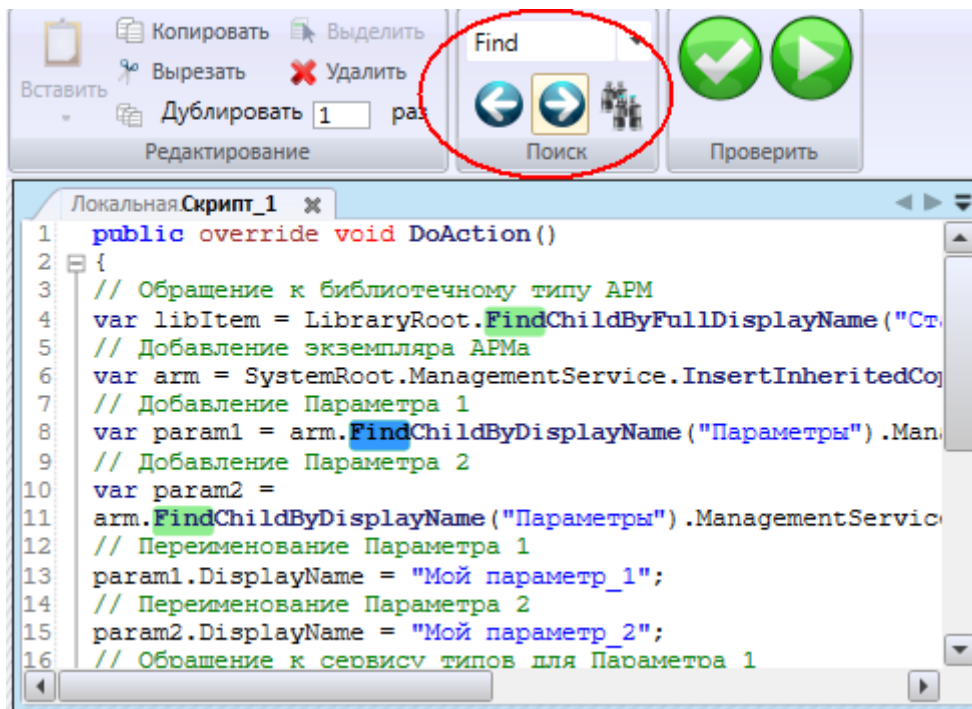
- Учет регистра – если установить этот флаг, при поиске учитывается регистр;
- Полное совпадение – если установить этот флаг, для поиска нужно задавать не подстроку, а слово целиком;
- Использовать регулярные выражения – если установить этот флаг, для поиска можно задавать регулярные выражения (например, \s для поиска пробелов);
- Режим замены – если установить этот флаг, панель поиска становится панелью поиска и замены:



Инструменты  и  предназначены для перехода к предыдущему/последующему вхождению подстроки.

Чтобы скрыть панель, нужно щелкнуть кнопку  в правом верхнем углу панели.

Для поиска можно использовать также панель 'Поиск' среды разработки:



5.11.5. МАСШТАБИРОВАНИЕ В РЕДАКТОРЕ ПРОЕКТА

Для масштабирования в редакторе проекта используются следующие типовые инструменты:

- инструменты масштабирования ;
- комбинированный инструмент масштабирования .

Кроме того, для масштабирования мнемосхем, FBD-, LD- и SFC-диаграмм предусмотрены дополнительные средства (см. Масштабирование диаграмм).

Для масштабирования всего редактора проекта предусмотрен также инструмент Масштаб приложения .


5.11.5.1. КОМБИНИРОВАННЫЙ ИНСТРУМЕНТ МАСШТАБИРОВАНИЯ

- Поле текущего масштаба
- Шкала масштаба
- Настраиваемое меню

Для изменения масштаба отображения содержимого активного (выделенного) окна/редактора предусмотрен следующий комбинированный инструмент, расположенный в нижней правой части редактора проекта:




Поле текущего масштаба



-  – в этом поле масштаб отображается как отношение текущего размера к оригинальному.

Если щелкнуть данное окно, открывается настраиваемое меню комбинированного инструмента масштабирования.

Шкала масштаба


-  - Шкала содержит ползунок и деления, слева и справа шкала ограничена кнопками  и  .

Деления шкалы соответствуют численным значениям масштаба, для отображения/задания которых используется настраиваемое меню .

Перетаскивание ползунка или щелчок шкалы слева/справа от ползунка (включая кнопки  и ) приводит к дискретному (по делениям) уменьшению/увеличению масштаба.

При изменении масштаба кадрирование производится относительно точки установки курсора или выделенного элемента.

Настраиваемое меню

Открывается при нажатии на кнопку: 

Вид меню:

Новый	
Сортировать	
0.10	0.10 x
0.20	0.20 x
0.30	0.30 x
0.40	0.40 x
0.50	0.50 x
0.60	0.60 x
0.70	0.70 x
0.80	0.80 x
0.90	0.90 x
1	1 x
1.10	1.10 x
1.20	1.20 x
1.30	1.30 x
1.40	1.40 x
1.50	1.50 x
2	2 x
3	3 x
4	4 x
5	5 x

Меню содержит несколько предустановленных значений масштаба. Для задания масштаба нужно щелкнуть требуемое значение.

При наведении курсора на значение открывается дополнительное меню, с помощью которого значение можно удалить (для этого нужно щелкнуть кнопку Удалить) или изменить.

Кроме значений, меню содержит следующие команды:

- Новый – добавить значение в меню (последним в списке). По умолчанию добавляется значение 1, которое нужно отредактировать с помощью дополнительного меню;
- Сортировать – сортировать список значений в меню сверху вниз по возрастанию.

5.11.5.2. ИНСТРУМЕНТЫ МАСШТАБИРОВАНИЯ

- Установка масштаба 100%
- Увеличение масштаба
- Уменьшение масштаба

Установка масштаба 100%

Для установки масштаба 100% (т.е. в соответствии с параметрами, заданными на вкладке Внешний вид редактора общих настроек – см. Вкладка 'Внешний вид') используются следующие инструменты:

- команда Масштаб.Масштаб 100% контекстного меню (CTRL+1);
- однократное нажатие на колесо мыши.

Увеличение масштаба

Для увеличения масштаба используются следующие инструменты:

- команда Масштаб.Увеличить масштаб контекстного меню (CTRL++);
- прокрутка колеса мыши от себя с удержанием клавиши CTRL или ПК (этот способ работает не везде).

Уменьшение масштаба

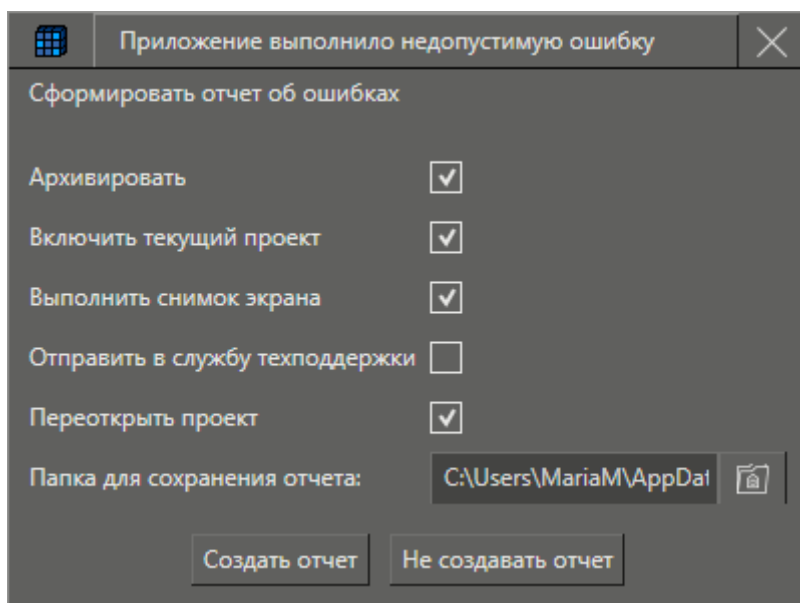
Для уменьшения масштаба используются следующие инструменты:

- команда Масштаб.Уменьшить масштаб контекстного меню (CTRL+-);
- прокрутка колеса мыши к себе с удержанием клавиши CTRL или ПК (этот способ работает не везде).

5.12. СЛУЖЕБНЫЕ ДИАЛОГОВЫЕ ОКНА

Приложение выполнило недопустимую ошибку

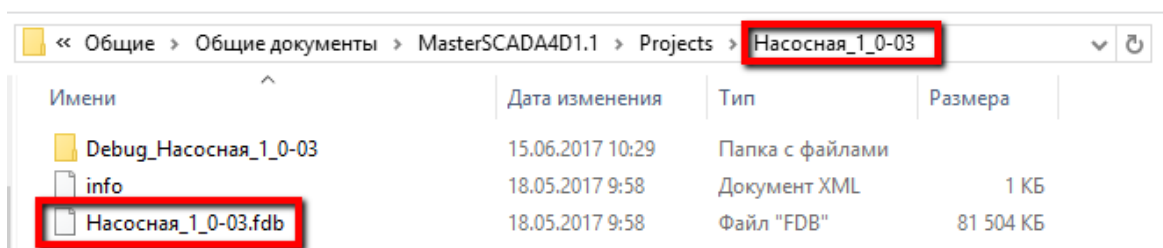
В случае возникновения непредвиденных ошибок, в среде разработки может появиться такое окно:



Рекомендуем нажать кнопку Создать отчет, дополнить появившийся файл отчета комментарием в какой ситуации появилось окно, и отправить файл отчета в техническую поддержку.

6. ПРОЕКТ В MASTERSCADА 4D

С точки зрения файловой системы, разрабатываемый проект представляет собой папку, содержащую файлы (в т.ч. файл БД *.fdb) и другие папки.



Важно! Имя папки и имя файла с расширением .fdb должны быть одинаковыми.

Переименовывать эти элементы средствами ОС не рекомендуется. Для переименования проекта следует использовать команду меню Управление проектом-Сохранить проект как). Если проект не отображается в диалоге открытия проекта, то, возможно, папка проекта и файл БД имеют разные имена.

Работа над любым проектом начинается с добавления необходимых элементов в дерево MasterSCADA

В одном проекте в среде разработки можно создавать одновременно несколько проектов для различных узлов. Это следует делать в случае, если в режиме исполнения необходимо обеспечить взаимодействие между различными устройствами (узлами), в которых

установлена среда исполнения MasterSCADA 4D, через внутренний протокол обмена данными.

Перед загрузкой проекта в среду исполнения его необходимо скомпилировать. В этом случае, для каждого узла создается свой проект, который уже загружается в среду исполнения.

Примерный план разработки проекта показан в разделе: Принципы проектирования

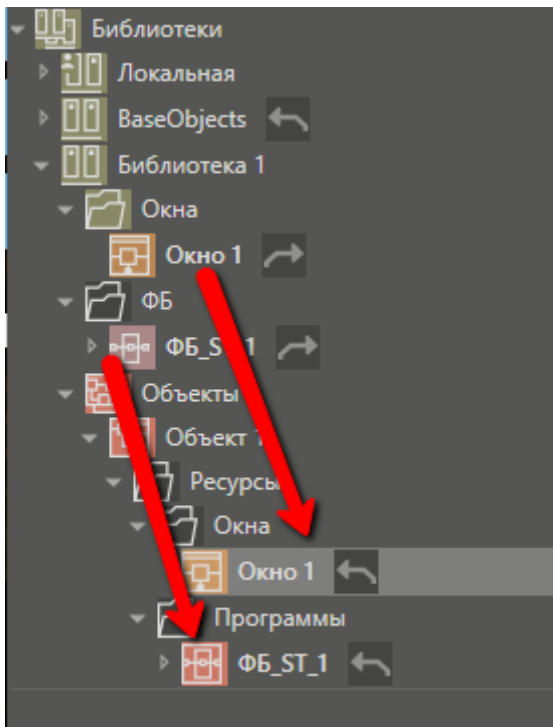
6.1. МЕТОДЫ РАЗРАБОТКИ ТИПОВ ЭЛЕМЕНТОВ

Чаще всего разработчики проектов создают свои типы объектов, каналов или тегов. Существует два подхода к разработке новых типов элементов:

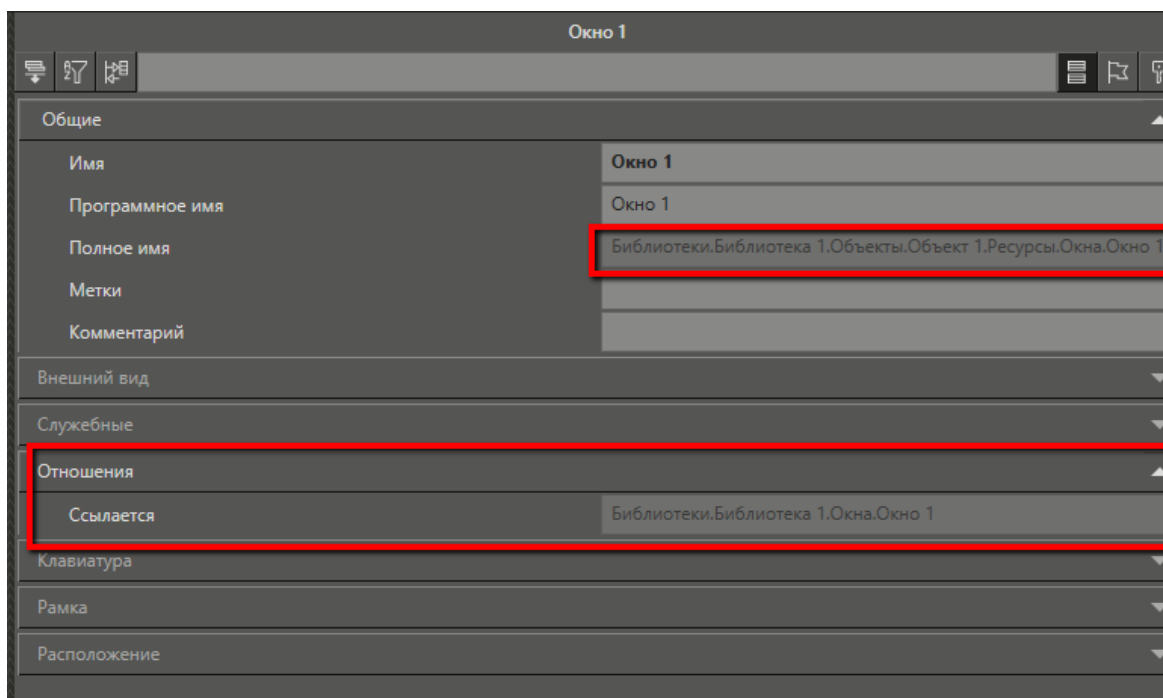
- от типа к экземпляру;
- от экземпляра к типу.

От типа к экземпляру

Такой подход можно назвать классическим. Сначала в библиотеке разрабатываются отдельные части сложного элемента. Например, если объект состоит из окон, программ и параметров, то сначала разрабатываются окна и функциональные блоки в соответствующих группах пользовательской библиотеки. Затем из них составляется библиотечный объект, тег или канал, и настраиваются связи между параметрами объекта и параметрами окна и программы.



В этом случае окна и программы такого объекта будут иметь отношения с типом вида Ссылается, а это значит, что изменить окно или программу в библиотечном объекте будет нельзя ни в библиотеке, ни в дереве объектов.



Для того чтобы проверить работу созданного объекта, его нужно добавить в дерево объектов и произвести отладку обычным способом.

От экземпляра к типу

Этот подход заключается в том, что сначала в дереве системы или в дереве объектов разрабатывается сложный элемент (канал, тег, объект), а уже затем он помещается в библиотеку. При этом в библиотеке создается тип. Такой подход при разработке проектов встречается намного чаще. Во-первых, не всегда получается сразу же оценить, какие элементы будут повторяться в проекте несколько раз, а следовательно их нужно помещать в библиотеку. Во-вторых, при таком подходе проще отладить элемент (при запуске режима отладки, имеется возможность задавать значения параметров и анализировать результат, а библиотечные элементы такой возможности не имеют).

6.2. ДЕРЕВО СИСТЕМЫ

В дереве системы создаются элементы, в которых происходит физическое исполнение разработанного проекта – АРМы, контроллеры, облачные сервисы и т.д. В MasterSCADA 4D эти элементы называются Узлами (см. Узел). В дереве системы определяется список используемых узлов и их настройки из числа поддерживаемых. В этом же дереве настраиваются права пользователей проекта (см. Безопасность).

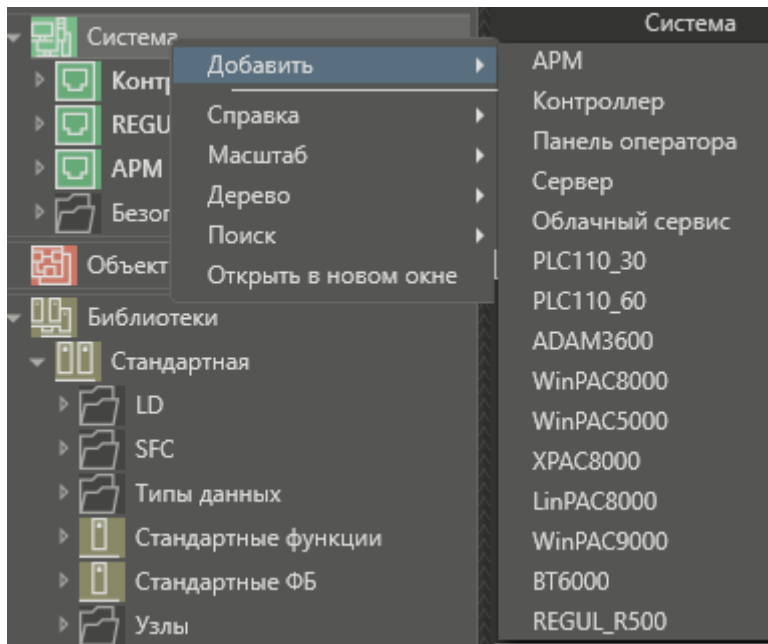
Количество добавленных узлов в дереве Системы должно соответствовать количеству используемых исполнительных систем. Для каждого узла определяется способ взаимодействия с модулями ввода-вывода, другими устройствами (например, счетчиками) и программным обеспечением, т.е. описывается подключенное к тому или иному узлу оборудование и ПО. Специфические элементы дерева системы, такие как протоколы, внешние каналы и встроенные каналы, служат для настройки связи проекта с внешними устройствами.

В дереве системы также существует возможность создавать и программы, и окна, но в этом случае разрабатываемый проект строго привязан к используемому оборудованию, и при смене оборудования будет затруднена адаптация проекта. Такой способ разработки называется линейным. Мы рекомендуем использовать объектный подход, когда в дереве системы описывается только физическая часть проекта, а вся обработка (программы, окна, настроечные параметры) находятся в дереве объектов.

6.2.1. УЗЕЛ

Узел - это верхний элемент дерева системы: Контроллер, АРМ, Сервер, Облако и пр., которому соответствует реальное или виртуальное оборудование. Количество узлов в дереве соответствует количеству исполнительных систем, которые планируется использовать. Если, например, в проект добавлены два узла, то предполагается, что будут использоваться два устройства, запрограммированных средствами MasterSCADA 4D, и которые должны будут взаимодействовать друг с другом. Т.е. данные от одного устройства планируется передавать в другое устройство, используя при этом внутренний протокол обмена MasterSCADA 4D. Все узлы отличаются друг от друга настройками, сделанными по умолчанию. Например, если в проект добавлен в качестве узла какой-либо контроллер, имеющий встроенный модуль ввода-вывода, то это отразится в дереве системы тем, что появится группа Встроенные модули, а для узла АРМ эта группа будет отсутствовать.

Добавить узел в проект можно несколькими способами. Один из способов - это нажать правую кнопку мыши на элементе Система. В контекстном меню пункт Добавить покажет, какие узлы доступны для вставки:

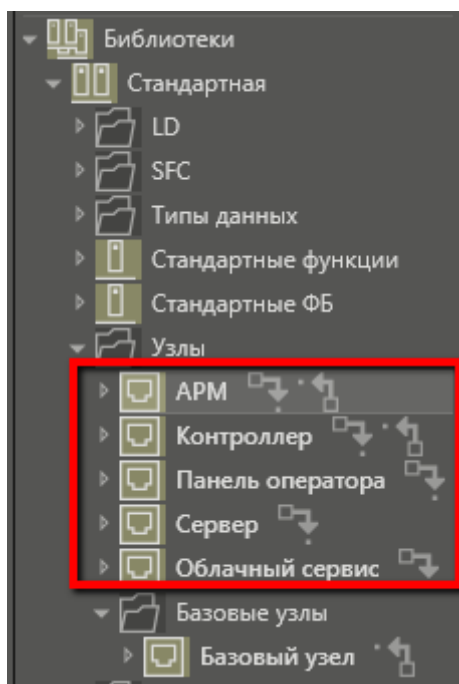


Разделяют два типа узлов:

- универсальные;
- предопределенные.

Универсальные узлы

Универсальные узлы можно использовать на любом устройстве, но они не будут отражать специфические особенности того или иного оборудования. Шаблоны универсальных узлов находятся в Библиотеке Стандартная:



Типы универсальных узлов соответствуют уровням системы управления:



Конфигурации узлов могут содержать различные наборы элементов. Все шаблоны узлов содержат корневые группы, которые содержат элемент стандартной библиотеки Базовый узел.

Важно! Разработчик проекта может менять в библиотеке различные настройки. Однако делать это не рекомендуется, т.к. после переустановки среды разработки настройки пользователя удалятся. Если необходимо создать свой узел с настройками по умолчанию, отличающимися от универсальных узлов, то лучше создать свою собственную библиотеку, в которую добавить свой узел.

Предопределенные узлы

Предопределенные узлы, добавляемые в дерево системы - это элементы, описывающие реальные устройства, как правило, контроллеры, для которых созданы специальные версии исполнительных систем, в которых учитываются их различные особенности. Если

контроллер имеет встроенные средства ввода-вывода, то его конфигурация по умолчанию содержит корневую группу Встроенные каналы.

Если в кейте контроллера могут размещаться устройства ввода-вывода, то его конфигурация по умолчанию содержит корневую группу Встроенные модули

По умолчанию в проект можно добавить предопределенные узлы компаний Овен и ICPDAS. Для подключения узлов других производителей необходимо выполнить подключение соответствующей библиотеки

6.2.1.1. ПАНЕЛЬ СВОЙСТВ УЗЛА

Панель свойств узла имеет вид:

APM 1

Общие

Отношения

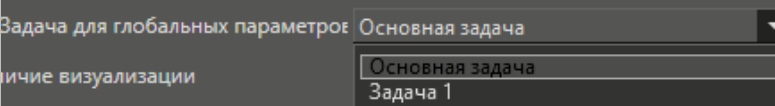
Настройки

Связь

Автозапуск клиента визуализации	<input checked="" type="checkbox"/>
IP адрес	127.0.0.1
Доступ по OPC UA	<input type="checkbox"/>
Номер экземпляра MPLC	0
Тайм-аут (мс)	3000
Порт для OPC UA	16550
Логин для OPC UA	
Пароль для OPC UA	
Адрес клиента OPC UA	
Ограничение длины имени узла в IEC61850	10
Использовать строковые Nodename в OPC UA	<input type="checkbox"/>
Порт для Modbus TCP	502
Порт для Web сервера	8043
Максимальный размер UDP пакета	0
Автозапуск исполнительной системы	<input checked="" type="checkbox"/>
Последовательность байт в Modbus	_3_2_1_0_7_6_5_4
Запрет автозагрузки проекта	<input type="checkbox"/>
Запрет входящих подключений	<input type="checkbox"/>
Максимальное число подключений Modbus TCP	20
Время неактивности подключения Modbus TCP	10s
Доступ по IEC104	<input type="checkbox"/>
Порт для IEC104	2404
ASDU адрес IEC104	1
Период циклической рассылки по IEC104	0ms
Доступ по IEC61850	<input type="checkbox"/>
Порт для IEC61850	102
В IEC61850 параметры SYSTEM_*_VALUE представлять как t, q, stVal	<input checked="" type="checkbox"/>
Использовать TLS для IEC61850	<input type="checkbox"/>

Описание основных свойств Узла:

Свойство	Назначение
Категория Настройки	
Задача для глобальных параметров	Определяет, в какой задачи узла будут архивироваться глобальные параметры. по умолчанию используется Основная задача. Но если параметры меняются

Свойство	Назначение
	<p>чаще/реже, то нужно добавить новую задачу, период работы которой будет удовлетворять потребности работы глобальных параметров.</p> <p>В выпадающем списке будут перечислены задачи, добавленные в узел:</p> 
Наличие визуализации	<p>Если флаг установлен, то проект, работающий в среде исполнения, может иметь окна. В этом случае у узла есть группа Графический интерфейс. В противном случае web-server использоваться не будет. По умолчанию данный флаг отключен для различных видов контроллеров.</p>
Параметры запуска RT	<p>Задаются ключи запуска исполнительной системы.</p>
Таймаут выполнения задач	<p>Устанавливается максимальное время выполнения одного цикла любой из задач системы. Если задача не выполнится за установленное время, то сработает аппаратный Watch Dog устройства. Если аппаратный Watch Dog отсутствует, то произойдет перезапуск исполнительной системы.</p>
Предельный объем памяти процесса (Мб)	<p>Определяет какой объем оперативной памяти может занимать процесс mpls, при превышении указанного значения исполнительная система завершит свой процесс и будет запущена заново. Перед завершением формируется дамп памяти, который попадет в отчет об ошибках при очередном формировании. Величина настройки должна выбираться из особенностей проекта. На размер процесса влияет сложность графических окон, количество передаваемых архивных данных, количество программ и т.п. Величина настройки определяется экспериментально. Если установлено значение 0, то объем памяти процесса исполнительной системы не контролируется.</p>

Свойство	Назначение
Количество хранимых лог файлов	Определяет, какое количество лог-файлов будет храниться. При превышении заданного значения более старые файлы будут удаляться.
Максимальный размер одного лог файла (МБ)	Определяет максимальный размер лог-файлов. При достижении указанного размера будет создаваться новый лог-файл.
Категория Связь	
Автозапуск клиента визуализации	Определяет необходимость запуска клиента визуализации при загрузке конфигурации. Если флаг установлен, то клиент визуализации запустится автоматически.
Запрет загрузки системного ПО	Определяет возможность автоматического обновления исполнительной системы. Если флаг установлен, то на вкладке управления узлом кнопка загрузки системы заблокирована. Также при отличии версий не появится кнопка загрузки в окне сообщений.
IP-адрес	Задается IP-адрес устройства, на котором установлена среда исполнения, и на которое должен загрузиться разрабатываемый проект.
Номер экземпляра MPLC	<p>Если на одном устройстве запущено несколько исполнительных систем, то задается номер экземпляра исполнительной системы, в которую должен загрузиться разрабатываемый проект.</p> <p>Важно! При покупке коммерческой версии среды исполнения необходимо указывать количество экземпляров, которое требуется запускать на одном устройстве. По умолчанию это значение равно одному. При попытке запустить количество экземпляров большее, чем закуплено, то они будут запущены, однако проект загрузится только в закупленное количество экземпляров. В среде разработки появится сообщение об ошибке.</p>

Свойство	Назначение
Тайм-аут	Время ожидания ответа от среды исполнения средой разработки. Если после трех попыток подряд не приходит ответ в заданный период времени, то фиксируется отсутствие связи с узлом. Затем среда разработки периодически делает попытки подключиться к узлу заново.
Порт для Modbus TCP	Если MasterSCADA 4D RT выступает в роли Modbus TCP Slave, то данный порт используется для работы с Master.
Максимальный размер UDP пакета	Для взаимодействия среды исполнения и среды разработки используется собственный протокол, основанный на UDP. Если в сети существуют ограничения на прохождения стандартных UDP-пакетов, то данной настройкой их размер можно уменьшить .
Порт для WEB-сервера	<p>Если клиент визуализации запускается из среды разработки, то данный порт будет использоваться в адресной строке.</p> <p>Данная настройка не влияет на порт, по которому среда исполнения работает с клиентом визуализации. Для изменения номера порта, по которому среда исполнения работает с клиентом визуализации, необходимо внести изменения в среде исполнения</p>
Доступ по OPC UA	<p>Активирует для среды исполнения возможность работать как OPC UA сервер.</p> <p>Возможность получить доступ к конкретному элементу для стороннего клиента будет определяться свойством Доступ через OPCUA</p>
Автозапуск исполнительной системы	Если флаг установлен и задан локальный IP-адрес, то при загрузке проекта в случае когда среда исполнения не была запущена ранее, запустится демо-версия исполнительной системы, которая входит в состав среды разработки. Если среда исполнения закроется,

Свойство	Назначение
	или произойдет отключение от среды исполнения, то среда исполнения также завершит свой процесс.
Последовательность байт в Modbus	Если MasterSCADA 4D RT выступает в роли Modbus Slave, то для обмена с Master будет использоваться данная последовательность байт.
Запрет автозагрузки проекта	Если флаг установлен, то при подключении к среде исполнения среды разработки, загрузки проекта не произойдет. Проект необходимо будет загрузить вручную или при помощи панели управления узлом.
Логин для OPC UA	Задается логин, который будет использоваться в случае, если среда исполнения будет работать как OPC UA сервер.
Пароль для OPC UA	Задается пароль, который будет использоваться в случае, если среда исполнения будет работать как OPC UA сервер.
Порт для OPC UA	Задается номер порта, который будет использоваться в случае, если среда исполнения будет работать как OPC UA сервер.
Запрет входящих подключений	Используется для работы с облачным сервисом. Если, например, контроллер не имеет внешнего IP-адреса, но, тем не менее, является источником данных для облачного сервиса, т.е. установлена межузловая связь, то, в случае если флаг установлен, контроллер будет сам отправлять в облако данные по изменению.
Максимальное число подключений Modbus TCP	Задается максимальное число одновременно подключенных клиентов в случае когда MasterSCADA 4D выступает в роли Modbus Slave
Время неактивности подключения Modbus TCP	Определяет длительность сессии, если клиент не совершает запросов клиенту, в случае когда MasterSCADA 4D выступает в роли Modbus Slave. Если

Свойство	Назначение
	клиент в течении этого времени не совершал запросов, то сессия закрывается.
Доступ по IEC104	Определяет возможность работать MasterSCADA 4D в роли IEC104 Slave.
Порт для IEC104	Задается порт, который будет использоваться для обмена данными с клиентами, если MasterSCADA 4D работает в роли IEC104 Slave.
ASDU адрес IEC104	Задается ASDU адрес, который будет использоваться для обмена данными с клиентами IEC104, если MasterSCADA 4D работает в роли IEC104 Slave.
Период циклической рассылки по IEC104	Определяет порядок передачи данных, в случае если MasterSCADA 4D работает в роли IEC104 Slave. Независимо от установленного значения, данные всегда будут отправлять по изменению: в этом случае отправляются только те значения, которые изменились. Если в данном поле установлено какое-либо значение, отличное от 0, то дополнительно к отправке по изменению, данные будут отправляться по всем внешним каналам с указанным периодом.
Доступ по IEC61850	Определяет возможность работать MasterSCADA 4D в роли IEC61850 Slave. Ограничения: <ul style="list-style-type: none"> • При превышении размера ответа сервера 65099 байт, сервер отвечает сообщением Confirmed-ErrorPDU. • Ограничение на длину полного имени переменной 64 байт. Полная длина имени переменной /LD/LN/DO..... не может превышать 64 байта
Порт для IEC61850	Задается порт, который будет использоваться для обмена данными с клиентами, если MasterSCADA 4D работает в роли IEC61850 Slave.

Свойство	Назначение
Использовать TLS для IEC61850	<p>Задаёт использование защищенного подключения когда MasterSCADA 4D является сервером и передает данные по протоколу IEC61850 Slave.</p> <p>Ключи и сертификаты TLS хранятся в папке IEC61850. Ключи и сертификаты сервера хранятся в папке IEC61850/srv. Ключи и сертификаты клиента хранятся в папке IEC61850/ctl.</p> <p>Имя файла ключа клиента должно быть <i>MasterSCADA4Dctl_key.pem</i>. Имя файла сертификата клиента должно быть <i>MasterSCADA4Dctl.cer</i>.</p> <p>Имя файла ключа сервера должно быть <i>MasterSCADA4Dsrv_key.pem</i>. Имя файла сертификата сервера должно быть <i>MasterSCADA4Dsrv.cer</i>.</p> <p>Сертификаты CA должны храниться в папке IEC61850/ctl/root для клиента и IEC61850/srv/root для сервера. Сертификаты CA могут иметь любое имя и расширение .cer.</p>
В IEC61850 параметры SYSTEM_*_VALUE представлять как t, q, stVal	<p>Определяет возможность передачи признака качества и метки времени. Если флаг отмечен, то для параметров типа SYSTEM_*_VALUE будет создаваться три параметра:</p> <p>stVal - значение</p> <p>t - метка времени</p> <p>q - признак качества</p>
Ограничение длины имени узла в IEC61850	<p>Определяется максимальная длина идентификатора тега протокола IEC61850 Slave. Идентификатор формируется на основании полного имени параметра. Рассмотрим пример. Пусть полное имя параметра будет:</p> <p>Система.ARM_DISPETCHERA.Параметры.Temperatura_max. Т.к. по стандарту IEC61850 имена должны быть заданы только латинскими буквами, то полное имя будет автоматически преобразовано:</p> <p>SystemRoot.ARM_DISPETCHERA.Parameters.Temperatu</p>

Свойство	Назначение
	<p>ga_max. Если в данной настройке установлено значение 8, то из каждой части полного имени будут отброшены лишние символы и в клиенте получится: SystemRo.ARM_DISP.Paramete.Temperat.</p> <p>Важно!Максимальная длина идентификатора может быть 64 байта, включая разделители. Если после применения данной настройки количество символов в идентификаторе будет выше, то данный параметр не будет отображаться в списке.</p>
Адрес клиента OPC UA	<p>Задается IP-адрес и номер порта узла клиента в формате <IP>:<Port>, в случае, если среда исполнения будет работать как OPC UA сервер. Например 192.168.110.1:9001. В примере 192.168.110.1 является IP адресом клиента OPC UA, а 9001 является номером порта клиента, заданный в настройке протокола клиента Порт ReverseHello. Клиент также должен иметь настройку, инициирующую использование данного режима установки соединения. При старте сервер пытается подключиться по адресу клиента, если соединение установлено, то начнется опрос сервера клиентом в обычном режиме.</p> <p>В случае установки значения в данном свойстве сервер не принимает входящие подключения, поэтому для выбора тегов в клиенте вначале нужно отключить настройку Адрес клиента OPC UA, считать теги, затем установить Адрес клиента OPC UA для того, чтобы сервер подключался к клиенту инициативно.</p>
Использовать строковые NodeID в OPC UA	<p>Определяет, что получит клиент при получении дерева параметров в качестве Node ID: ID параметра или его полное имя. Если флаг установлен, то при получении дерева тегов возвращаются строковые иерархические имена тегов. Т.е. OPC UA сервер помимо ID в качестве NodeId принимал также полный путь к элементу в виде строкового NodeId. Этот путь формируется по отображаемому имени.</p>

Свойство	Назначение
	<p>Если ранее дерево было получено клиентом, а потом установили флаг, то исполнительная система будет передавать данные по цифровому Node ID.</p> <p>Примеры, когда данная настройка рекомендована:</p> <ul style="list-style-type: none"> • Один и тот же проект клиента используется для работы с разными, но однотипными узлами/проектами MasterSCADA 4D. В этом случае цифровой NodeID будет отличаться, а строковой - совпадать, чтобы не изменять проект клиента можно воспользоваться настройкой. • Карта клиента формируется вручную, например, в текстовом файле.
Отображение структур в OPC UA сервере	<p>Определяет способ отображения структур в клиенте OPC UA, в случае если MasterSCADA 4D выступает в роли OPC UA сервера. Свойство может принимать следующие значения:</p> <ul style="list-style-type: none"> • Параметр структурного типа - один параметр, передается как ExtensionObject • Отдельные поля - параметр представлен папкой с отдельными полями • Параметр структурного типа и отдельные поля - параметр можно запрашивать как ExtensionObject, также он содержит внутри себя отдельные поля
Таймаут запроса архива	<p>Задаёт таймаут, с которым идет опрос архивных данных по межузловым связям, например, когда есть связь между архивируемыми параметрами двух узлов</p>

6.2.1.1.1. СВОЙСТВО ПАРАМЕТРЫ ЗАПУСКА RT

Данное свойство узла позволяет задать опции командной строки сервера. Например, для включения модема /m3 /b115200 или RTU SLAVE /c3 /b115200.

Можно также передавать опции, задаваемые в *mplc.conf*:

/HDArchiveSavePeriod:30 /HDArchiveSizeLimit:60000 /HDArchiveStore:1

Примеры кодов:

Ключ	Описание
/cN	N - обозначает номер com-порта, по которому система будет передавать данные в режиме Slave через механизм Внешние каналы.
/mN	<p>Обозначает, что модем подключен к com-порту с номером N. Номер определяется на уровне операционной системы, как тот или иной порт определяет в ОС, так и должен быть задан в MasterSCADA 4D. Ключ используется для отправки СМС. По умолчанию при работе с модемом используется строка инициализации АТЕ0 (используется для инициализации модемов ОБЕН ПМ01 и IRZ TU32.)</p> <p>Если требуется другая строка инициализации, то ключ формируется следующим образом: /iT /mN, где Т - текст строки инициализации. Для работы с модемами могут быть использованы и другие ключи, например, для настройки скорости и т.п.</p>
/bZ	Z - это скорость, на которой будет работать com-порт.
/HDArchiveStore:30	Используется в случае, когда MasterSCADA 4D передает архивные данные в M-veSavePeri PLC OPC Server через механизм внешние каналы. Задается период сохранения буфера на флеш. Единица измерения - секунда (при установленной настройке HDArchiveStore:1). Если настройка не задана, то используется значение по умолчанию, равное 10 сек.
/HDArchiveSizeLimit:60000	Используется в случае, когда MasterSCADA 4D передает архивные данные в M-veSizeLimi PLC OPC Server через механизм внешние каналы. Задается объем памяти в kB, используемый для буферизации архивных данных для передачи OPC серверу M-PLC OPC Server. Если настройка не задана, то будет использоваться значение по умолчанию, равное 4000 kB.
/HDArchiveStore:1	Используется в случае, когда MasterSCADA 4D передает архивные данные в M-veStore:1 PLC OPC Server через механизм внешние каналы. При данной настройке будет осуществляться сброс на диск буфера данных (создается файл в папке указанной в настройке Место хранения архива) для передачи HDA-серверу (используется в случае, если сервер не получил все данные в момент перезапуска контроллера). По умолчанию отключено.
/Log:<путь к логу	Указывает по какому пути будет находиться файл с информацией о работе исполнительной системы. По умолчанию лог-файл пишется в папку

внутри контроллера>	<p>исполнительной системы. Особенно актуально введение этого параметра для контроллеров, в которых есть возможность подключить внешний накопитель, например, flash-карту. Такой же лог-файл можно получить используя внешнее приложение.</p> <p>Имя формирующегося файла совпадает с текущей датой создания. Размер лог-файла ограничен 100 Мб. При переполнении текущий лог переименовывается (к имени добавляется "_old") и начинается формироваться новый. В день будет не более 200Мб, т.к. каждый раз файл с окончанием _old будет заменяться более свежим.</p>
/t	<p>Включает расширенную диагностику работы среды исполнения. При использовании данного ключа, в файл с информацией о работе исполнительской системы будет попадать лог обмена среды исполнения с различными устройствами, работающими по заданным в проекте протоколам.</p>
/nowdt	<p>Команда отключения watchdog в среде исполнения. Если режим watchdog работает, то при зависании какой-либо задачи более чем на 20 секунд, произойдет или перезапуск среды исполнения, или перезапуск устройства (зависит от его возможностей).</p>
/iT	<p>Используется для указания строки инициализации модема, где T - текст строки инициализации</p>
/export	<p>Используется для экспорта в файл данных горячего рестарта для контроллеров с энергонезависимой памятью.</p> <p>Важно! Если данный ключ оставить в проекте, то экспорт данных будет происходить при каждом перезапуске среды исполнения. Поэтому после выполнения операции необходимо удалить ключ из Параметров запуска RT и перезапустить среду исполнения.</p>
/delhr	<p>Удаляет файл горячего рестарта.</p> <p>Важно! Если данный ключ оставить в проекте, то удаление файла будет происходить при каждом перезапуске среды исполнения. Поэтому после выполнения операции необходимо удалить ключ из Параметров запуска RT и перезапустить среду исполнения.</p>
/imit	<p>Включает имитацию значений каналов протоколов, в которые добавляются модули ввода-вывода, например, Modbus и DCON. Используется, если необходимо обеспечить изменение значений каналов в случае, когда реальное</p>

	устройство не подключено. Аналоговые значения будут изменяться в диапазоне от 0 до 100, а дискретные менять свое значение с TRUE на False.
/pwd:пароль	<p>Задается пароль, который будет использоваться при загрузке проекта из среды разработки. Если в исполнительной системе задан пароль, то загружать в него проект можно только, если в настройках среды разработки во вкладке Разное задан такой же пароль. Как правило используется для защиты публичного сервера от загрузки в него конфигурации пользователями.</p> <p>Если задан пароль в исполнительной системе, то он проверяется и при автоматическом обновлении версии исполнительной системы.</p>
/udp2	Включает обработку UDP-запросов на получение/запись данных между средой разработки и исполнительной системой в отдельном потоке (по умолчанию отключено, так как в некоторых сетях не доходят ответы при отправке с другого UDP сокета). Данные типы запросов в статистике отображаются как N2

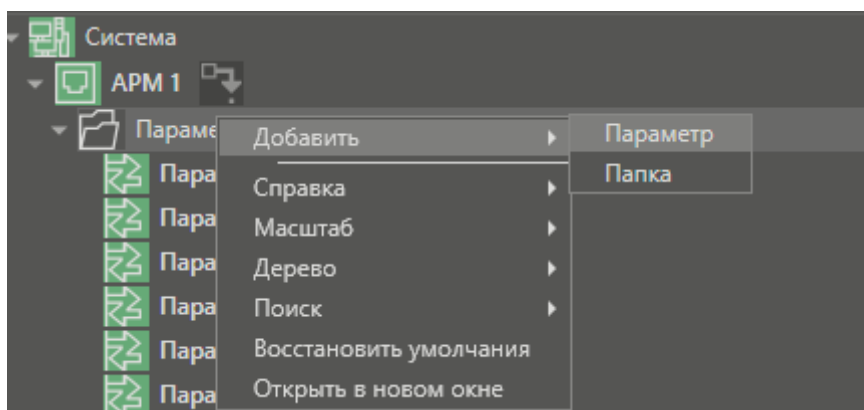
Важно! Отключение режима watchdog должно быть осмысленным. Рекомендуется отключать его только в том случае, когда при его наличии работа невозможна.

Данные ключи можно также использовать при настройке среды исполнения, предназначенной для работы на ОС Windows, при помощи приложения MasterSCADA 4D Monitor

6.2.1.2. ПАРАМЕТРЫ УЗЛА

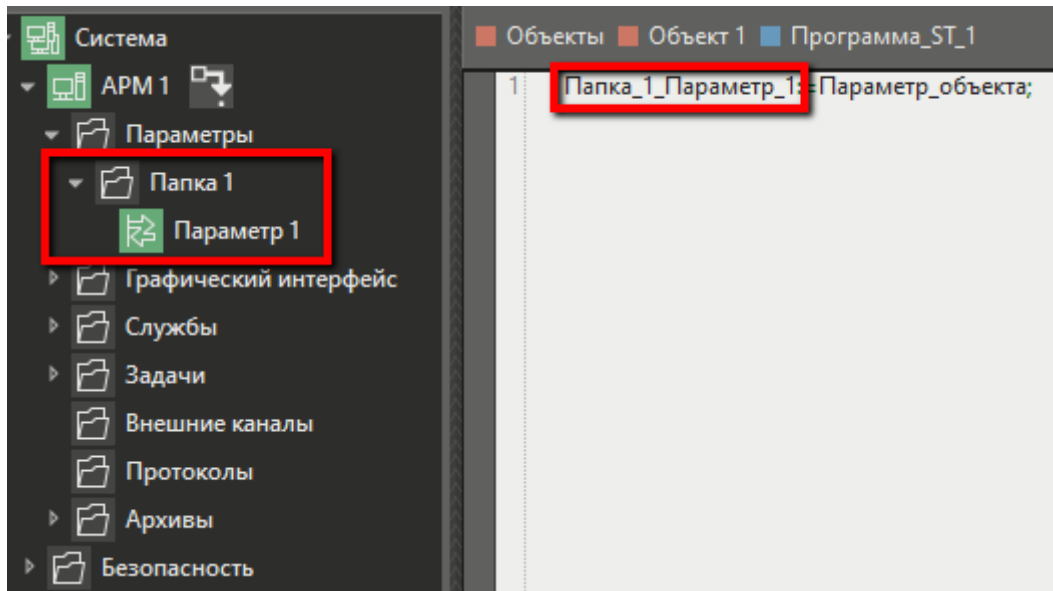
В группе создаются Параметры (переменные). Параметры необходимы для использования их в программах для связи со свойствами графических элементов (для динамизации).

Группа в узле не отображается до тех пор пока не будет добавлен хотя бы один параметр. Добавить параметр в узел в этом случае можно при помощи контекстного меню или контекстной панели.



Связав два параметра, принадлежащих разным узлам, можно осуществить передачу данных между узлами с использованием внутреннего протокола обмена.

Если параметры созданы в дереве системы, то они являются глобальными. Такие параметры можно использовать в программе ST без дополнительного определения, достаточно написать имя переменной в тексте программы. Если параметр находится в папке узла, то тогда его имя будет `ИмяПапки_ИмяПараметра`:



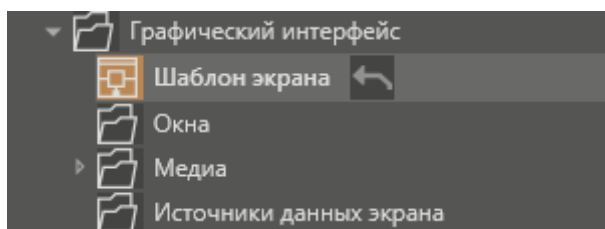
Важно! При использовании глобальных параметров (переменных) в программах дерева объектов или библиотек, проверка программы будет неуспешна. В этом случае ошибки типа: Неизвестная переменная [имя глобальной переменной] можно игнорировать. При проверке всего проекта ошибки не будет.

Параметры узла можно использовать для передачи данных между задачами.

6.2.1.3. ГРАФИЧЕСКИЙ ИНТЕРФЕЙС

Узел содержит данную корневую группу в том случае, когда для него задана поддержка НМІ (т.е. для узла задано свойство Наличие визуализации = TRUE),

Группа Графический интерфейс содержит следующие группы:



- Шаблон экрана
- Окна
- Медиа

- Источники данных экрана

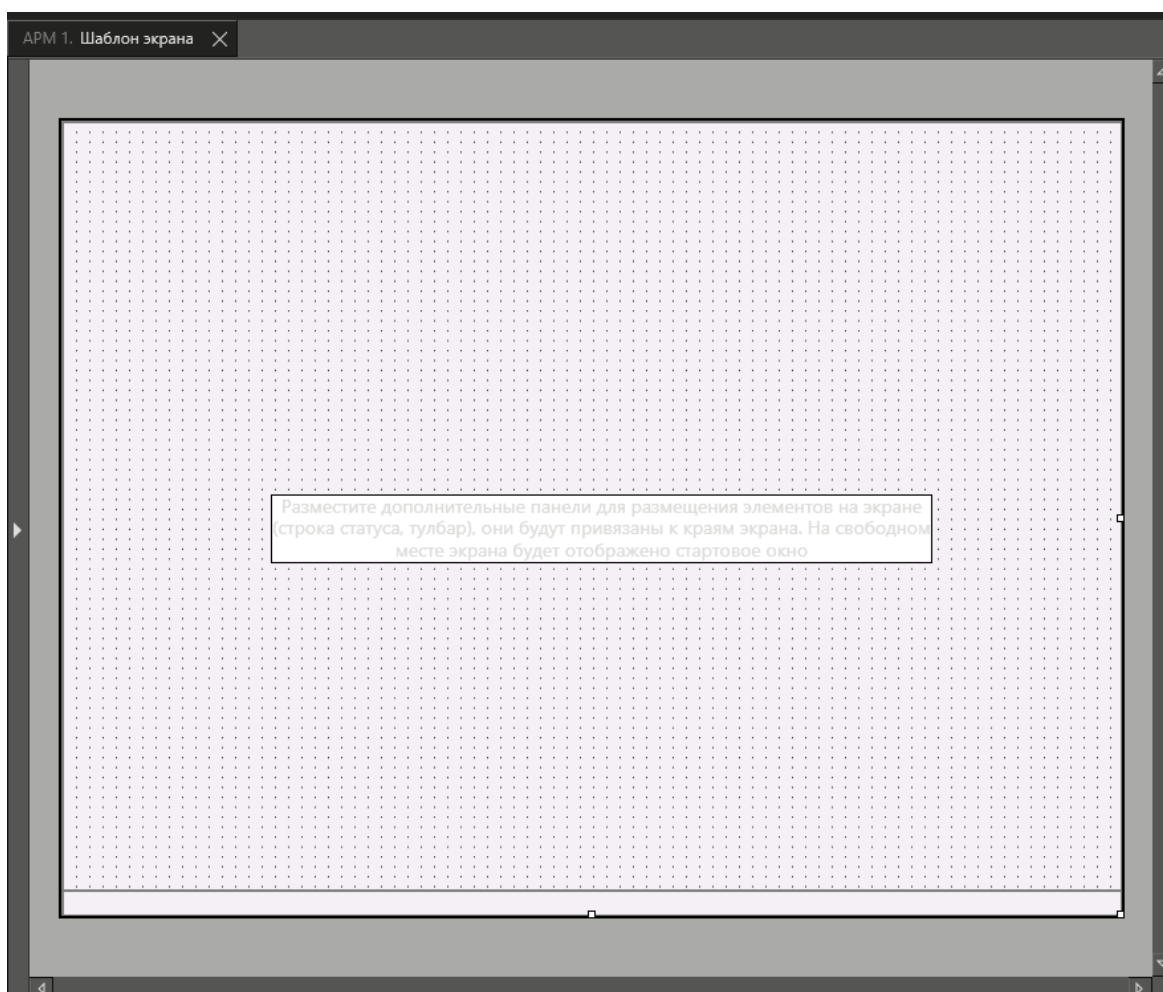
6.2.1.3.1. ШАБЛОН ЭКРАНА

Шаблон экрана – это оболочка для отображения окон узла.

По умолчанию, шаблон экрана состоит из следующих элементов:

- Контейнер стартового окна - по сути этот элемент является обычным Контейнером, в котором в режиме исполнения отображается стартовое окно проекта. Данный элемент занимает по возможности всё свободное место экрана.
- Панель статуса - находится в нижней части шаблона экрана. По умолчанию, данная панель всплывает только если появляются какие-либо системные ошибки в режиме исполнения. Разработчик проекта может разместить на ней какие-либо индикаторы и другие элементы.

Вид шаблона экрана по умолчанию:



В шаблон экрана могут быть добавлены клеммники, которые в свою очередь могут быть использованы в Триггерах, добавленных в шаблон экрана. Например, триггеры в шаблоне экрана могут быть использованы для открытия окон.


6.2.1.3.1.1. ПАНЕЛЬ СВОЙСТВ ШАБЛОНА ЭКРАНА

Вид панели свойств шаблона экрана:

Шаблон экрана	
Общие	
Имя	Шаблон экрана
Полное имя	Система.APM 1.Графический интерфейс.Шаблон экрана
Метки	
Комментарий	
Служебные	
Внешний вид	
Расположение	
Рамка	
Отношения	
Экран	
Тип ширины	Относительные
Тип высоты	Относительные
Сохранять пропорции	По максимальной стороне
Период обновления	100
Период мигания	1000
DPI экрана	72
Полноэкранный режим	<input type="checkbox"/>
Запрет масштабирования	<input checked="" type="checkbox"/>
Высота заголовка всплывающих окон	20
Заголовок окна	
Шрифт заголовка окон	Arial
Размер шрифта заголовка	12
Настройки	
Язык исполнения	ru
Уровень предупреждений	Все
Кодировка сохраняемых csv файлов	utf-8
Место хранения состояния окон	В HMI клиенте
Период задачи экрана, мсек	1000
Слой	

Панель Свойства для элемента Шаблон экрана содержит, наряду с другими, свойства, общие для всех мнемосхем узла. Наиболее часто изменяемые свойства Шаблона экрана:

Название	Назначение
Группа Общие	Описание доступно в разделе справочной системы.
Группа Внешний вид	Настройки, заданные на странице свойств, распространяются только на оболочку экрана. На окна, открывающиеся в контейнерах шаблона экрана эти свойства не распространяются.
Группа Заголовков окна	Определяет размер и шрифт заголовков всплывающих окон.
Группа Рамка	Определяет свойства рамки оболочки окна клиента: цвет и толщину.
Группа Расположение	Определяет размер и ориентацию клиента. По умолчанию установлен размер шаблона экрана 1920x1080.
Группа Служебные	
Группа Экран	
Тип ширины	Описание в разделе Категория свойств Экран у окон
Тип высоты	
Сохранять пропорции	
Период обновления	Период, с которым обновляются данные на экране, и с каким команды с экрана будут отправляться на сервер. Эту настройку необходимо выбирать с учётом реального периода получения данных, периодов задач и особенностей зрения и реакции человека. Так, если данные меняются в каналах с периодом 1 раз в минуту, то устанавливать период обновления данных 50 мс нецелесообразно. Напротив, в случае если период получения данных 1 секунда, а период обновления данных установлен 5 секунд, то оператором могут быть упущены некоторые значения, а команды оператора будут проходить на сервер с задержкой. Как правило, данную настройку разработчики проекта оставляют без изменения.

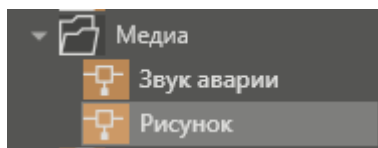
Название	Назначение
Период мигания	Если у элементов свойство Мигание установлено в состояние True, то элемент будет мигать с указанным периодом. Данная настройка одина для всех элементов. Установить разный период мигания для различных элементов нельзя.
DPI экрана	Определяет разрешение экрана. Размерность – число точек на дюйм. Как правило, это поле не редактируют при разработке проекта. Увеличение значения приведет к улучшению четкости картинки, но может увеличить время загрузки окон в браузере.
Полноэкранный режим	Если флаг установлен, то окно клиента, открытое из среды разработки, откроется на весь экран. Данная настройка не влияет на другие способы открытия клиента.
Язык исполнения	Определяет язык интерфейса браузера, на котором будут выдаваться служебные окна и сообщения. Например, окно логина пользователя. Настройка может принимать значения ru (русский язык) и en (английский). Язык среды разработки при этом может не совпадать с языком клиента
Запрет масштабирования	<p>Разрешает или запрещает масштабирование страницы клиента внутри браузера специальными командами, например,</p>  <p>, а также вращением колеса мыши.</p> <p>Если флаг установлен, то при загрузке страницы шаблон экрана займет все место внутри окна браузера, а при изменении размеров окна браузера каждый размер содержимого будет сжиматься или разжиматься. Если изменить масштаб страницы в браузере, например, прокрутив колесо мыши вверх или вниз, то размеры элементов меняться не будут. (Возможны незначительные изменения внешнего вида, но эти изменения не приведут к скрытию элементов за границами видимости). В экранах с поддержкой Touch Screen жесты изменения масштаба восприниматься не будут. Рекомендуется флаг устанавливать</p> <p>Если флаг не установлен, то при открытии клиента шаблон экрана займет все место в окне браузера. Если размер окна браузера изменится, то размер загруженных элементов не изменится до тех</p>

Название	Назначение
	пор, пока оператор вручную не изменит масштаб страницы специальными командами или колесом мыши. Это может привести к скрыванию элементов за границами видимости.
Уровень предупреждений	<p>Настраивает уровень вывода предупреждений в клиенте визуализации в строке статуса. Возможные варианты:</p> <ul style="list-style-type: none"> • Нет (Off) - без предупреждений • Ошибки (Errors) – отображаются только сообщения об ошибках • Предупреждения (Warnings) – отображаются предупреждения и сообщения об ошибках. • Все (All) – отображаются все сообщения (информационные сообщения, предупреждения и сообщения об ошибках).
Период задачи экрана, мсек	Определяется период, с которым будут работать программы, у которых свойство Место исполнения имеет значение В задаче экрана.
Высота заголовка всплывающих окон	Данная настройка определяет высоту заголовков всплывающих окон, окон управления, окон ввода уставок и т.п., созданных пользователем.
Место хранения состояния окон	<p>Определяет возможность использование настройки Сохранить состояние у графических элементов, а также место хранения этих настроек. В текущей версии доступны следующие варианты значения свойства:</p> <ul style="list-style-type: none"> • В НМІ клиенте (client) - настройки будут хранится в локальной базе браузера или встроенного клиента (по умолчанию) • Не сохранять (none) - не сохранять состояние, даже если у каких-то элементов проекта свойство Сохраниться состояние включено
Кодировка сохраняемых csv файлов	Определяет, в какая кодировка будет использоваться при сохранении информации из клиента визуализации в csv-файл, например, при работе с трендом

Название	Назначение
Группа Слой	Описание в разделе Категория Слой у окон

6.2.1.3.2. МЕДИА

Эта группа входит в состав групп Графический интерфейс , Шаблон экрана и Окно . В группе Медиа создаются элементы Медиа (см. Элемент Медиа).



6.2.1.3.2.1. ЭЛЕМЕНТ МЕДИА

Данные элементы создаются в группе Медиа . Элемент Медиа может содержать мультимедийный ресурс; для элемента определена команда контекстного меню Ресурс. Удалить содержимое.

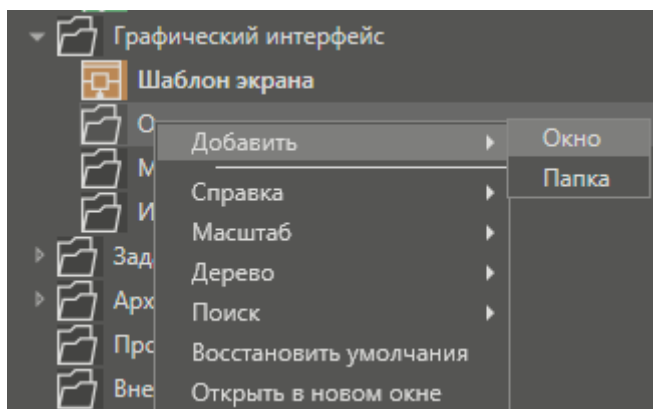
Загруженный ресурс может использоваться графическими элементами, а также ФБ (см. PlaySound).

Для выполнения операций с элементами Медиа используется Менеджер ресурсов НМИ .

Медиа-ресурс, добавленный в узле, будет доступен для всех окон проекта.

6.2.1.3.3. ОКНА

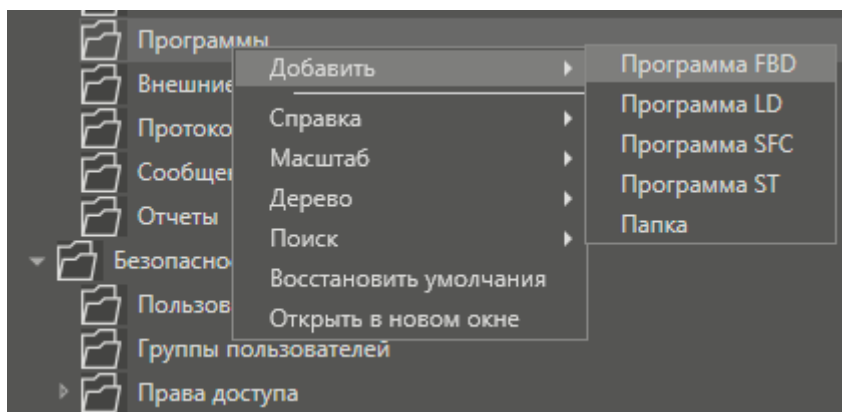
В данной группе создаются Окна.



При добавлении окна автоматически откроется графический редактор.

В данной группе, как правило, создаются окна, которые нельзя по смыслу отнести к какому-либо объекту. Например, окно, которое содержит в себе кнопки перехода на другие окна проекта. Такое окно можно разместить в шаблоне экрана, и тогда оператор в режиме исполнения всегда будет видеть данную панель, независимо от того, какое окно откроется в стартовом контейнере.

6.2.1.4. ПРОГРАММЫ



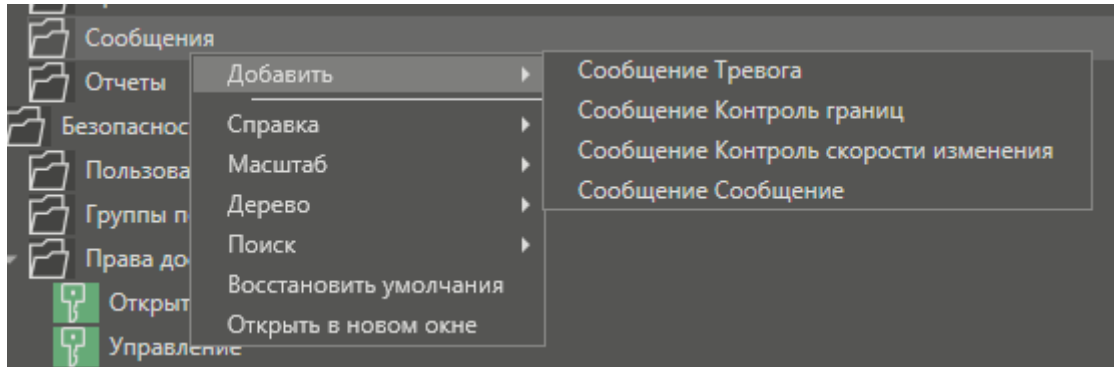
В этой группе создаются программы. Для написания программы можно использовать любой из поддерживаемых языков. Как правило, программы создаются непосредственно в дереве системы в тех случаях, когда нельзя по смыслу отнести ту или иную программу к конкретному объекту. Например, когда в программе производится обработка данных, полученных от каналов. Или если ведется линейная разработка проекта, когда разработчик не разделяет свой объект автоматизации на отдельные части - объекты, а весь проект также делает в дереве системы.

Группа в узле не отображается до тех пор пока не будет добавлен хотя бы одна программа. Добавить программу в узел в этом случае можно при помощи контекстного меню или контекстной панели.

Важно! Программы, добавленные непосредственно в группу Программы узла, будут исполняться раньше программ, находящихся в дереве объектов.

6.2.1.5. СООБЩЕНИЯ

В данной группе создаются пользовательские сообщения, которые нельзя отнести по смыслу к какому-либо объекту.



Группа в узле не отображается до тех пор пока не будет добавлено хотя бы одно сообщение. Добавить сообщение в узел в этом случае можно при помощи контекстного меню или контекстной панели.

6.2.1.6. СЛУЖБЫ

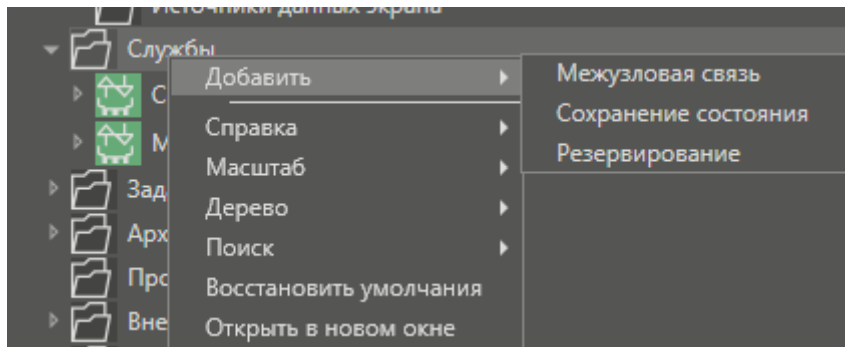
Данная группа узла содержит службы, обеспечивающие связи между различными исполнительными системами, включенными в один проект, а также сохранение данных, которые в последствии используются для горячего рестарта.

Важно! Данная группа не видна в упрощенном дереве.

По умолчанию, при создании проекта данная группа содержит два экземпляра системных протоколов:

- Межузловая связь - используется, если в проекте присутствует несколько узлов, и параметры, принадлежащие разным узлам, связаны друг с другом. Период передачи данных зависит от настроек, сделанных в панели свойств Узла в категории Сеть или в панели свойств данного элемента. Данная служба включена в состав узла по умолчанию. Добавлять ее в проект не требуется.
- Сохранение состояния - используется для восстановления значений после перезагрузки среды исполнения. Как правило, применяется для того, чтобы значения уставок после перезагрузки проекта в среде исполнения приняли значения, введенные оператором либо полученные в результате работы программ в предыдущей сессии. Сохраняются те значения параметров, в настройках которых установлено свойство Сохранять=Да. Период сохранения зависит от настроек, сделанных в панели свойств службы.

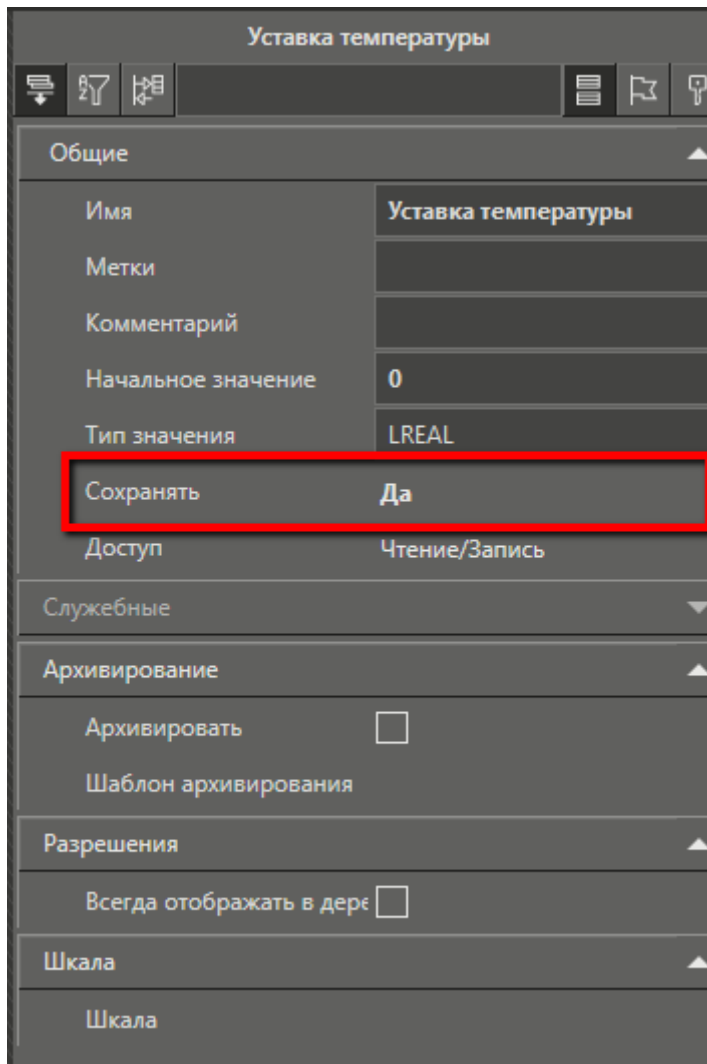
Для организации резервирования узлов необходимо добавить соответствующую службу:



- Резервирование - используется при резервировании узлов.

6.2.1.6.1. СОХРАНЕНИЕ СОСТОЯНИЯ

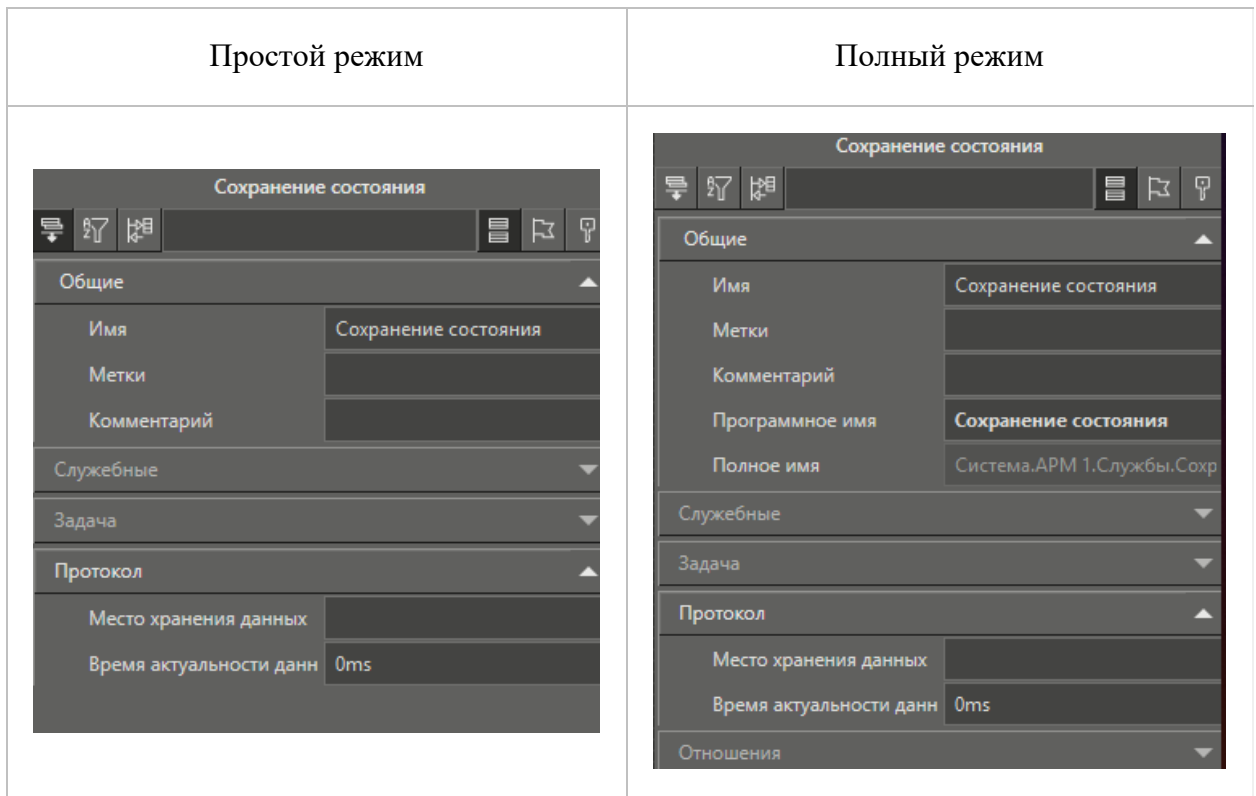
Служба Сохранение состояния используется для восстановления значений после перезагрузки среды разработки, и включена в состав Узла по умолчанию (отдельно добавлять ее в проект не требуется). Как правило, используется для того, чтобы значения уставок после перезагрузки проекта в среде исполнения приняли значения, введенные оператором, либо полученные в результате работы программ в предыдущей сессии. Сохраняются те значения параметров, в настройках которых установлено свойство Сохранять=Да. Данные будут сохраняться в файле *session.bin*. Расположение файла зависит от настроек службы, а также типа ОС, на которой установлена исполнительная система. Поддерживается сохранение значений параметров различных типов данных, в том числе динамических массивов (массивов переменной длины), также длинных строк (более 256 символов), при условии, что параметр имеет тип STRING



Важно! Свойство Сохранять=Да нельзя устанавливать для параметров программ, у которых свойство Доступ=InOut

Период сохранения зависит от настроек, сделанных в панели свойств службы.

Вид панели свойств:



Описание основных свойств службы Сохранение состояния:

Название	Тип	Описание
Категория Протокол		
Место хранения данных	STRING	<p>Определяет место хранения данных. Задается полный путь к папке, в которой следует хранить файл <i>session.bin</i>. Если поле пустое, то файл сохранится в рабочую папку исполнительной системы:</p> <p style="text-align: center;">для ОС Linux - <i>mplc/session.bin</i>.</p> <p>для ОС Windows - в папке <i>Server</i>, расположение которой задается в настройках MasterSCADA 4D Monitor (по умолчанию - <i>c:\ProgramData\MPSSoft\MasterSCADA4DRTBeta\Server\session.bin</i>)</p> <p>В поддерживаемых контроллерах с энергонезависимой памятью (например, PLC110, M903) данные горячего рестарта хранятся в такой памяти.</p>

Название	Тип	Описание
Время актуальности данных	TIME	Определяет время хранения данных. Если прошло времени больше, чем указано в данном поле, то после перезагрузки исполнительной системы параметры примут значения заданные разработчиком проекта по умолчанию.

6.2.1.7. АРХИВЫ

Архивы - это корневая группа узла, в которой находятся элементы для работы с архивами данных и сообщений. Архивирование в MasterSCADA 4D ведется в базах данных (БД).

Для хранения архивов могут быть использованы следующие базы данных:

- SQLITE;
- POSTGRESQL;
- MS SQL.

Существует также техническая возможность подключить следующие БД:

- DB2;
- Firebird;
- ODBC;
- Oracle;
- MySQL.

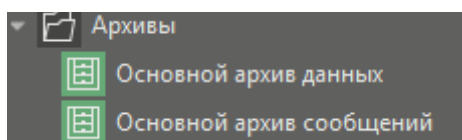
Сроки подключения и особенности работы можно узнать в технической поддержке.

База данных SQLITE 3 входит в состав MasterSCADA 4D. Другие базы данных необходимо установить самостоятельно.

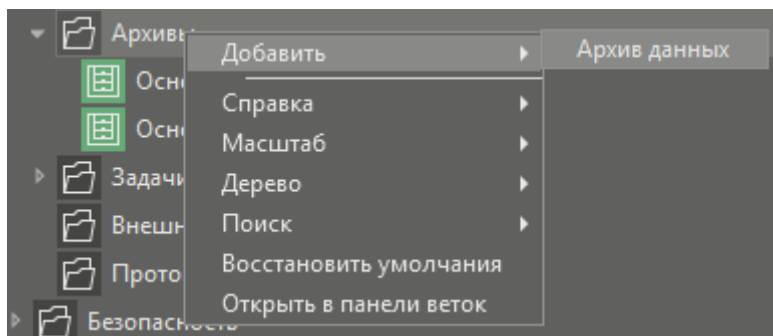
Различают два типа архивов: архив данных и архив сообщений.

Каждый тип архивов хранится в своей базе данных.

В данной группе, по умолчанию, находится элемент Основной архив данных, в котором задаются настройки для архива данных, и Основной архив сообщений:



При помощи контекстного меню группы можно добавить несколько дополнительных архивов.



Важно! Если в начале работы проекта в исполнительной системе была выбрана база данных SQLite, а затем потребовалось перейти на другие базы данных (Postgre или MS SQL), и при этом сохранить накопленные ранее архивы данных и сообщений, то это возможно сделать специальной программой. Программа может быть выслана по запросу в техническую поддержку

Архив данных

Архив данных – это архив значений параметров. При архивировании в архив попадает не только само значение параметра, но также и метка времени и признак качества. В архив попадают значения только тех параметров, в настройках которых установлен флаг Архивировать. Архивироваться могут параметры простых типов, а также параметры сложных типов - массивы, структуры. Сложные типы данных хранятся в БД и читаются в виде JSON.

В MasterSCADA 4D допускается использовать несколько баз данных для одного узла. Можно настроить проект таким образом, что параметры одного объекта будут храниться в одной БД, а параметры другого объекта – в другой БД.

Важно! В текущей версии MasterSCADA 4D не поддерживается резервирование баз данных. Эту задачу следует решать средствами администрирования самих баз данных, либо резервированием серверов, на которых установлена MasterSCADA 4D.

Для длительного хранения данных используются слои. Слой данных – это отдельно хранимый набор архивных значений параметра, отличающийся периодом записи, способом предварительной обработки перед записью и длительностью хранения. Основное назначение слоев – повышение быстродействия при извлечении данных для их отображения или обработки. Предполагается, что разработчик проекта заранее знает, что, например, его будут интересовать средние почасовые значения параметра или просмотр ежеминутных текущих значений. Если такие слои заданы на этапе создания проекта, то появляется возможность при запросе этой информации для отображения на тренде (графике) или подготовке отчета извлечь все необходимые данные без поиска и обработки данных. Тем самым значительно повышается быстродействие этих операций. Можно организовать хранение архивов таким образом, что один слой будет храниться в одной БД, а другой – в другой БД.

Если требуется обеспечить для разных параметров разные настройки архивирования, например, разное время хранения данных, мертвую зону и т.п., то можно использовать шаблоны архивирования.

Для работы с архивами данных в среде MasterSCADA 4D предусмотрены следующие элементы:

График – элемент графического редактора;

READ_ARCHIVE_DATA – функциональный блок, который выполняет выборку из архива параметра. Может использоваться как самостоятельно, так и в программах;

Отчеты формируются в различных формах на основе архивных данных (суточные отчеты, часовые и др.).

Если необходимо обеспечить передачу архивных данных между узлами, включенными в один проект, то в проекте следует установить связь между архивируемыми параметрами объектов разных узлов. В этом случае параметр-получатель будет периодически запрашивать архивные данные у того узла, которому принадлежит параметр-источник. В результате этого образуется два независимых архива: у переменной-источника и у переменной-приемника. В случае если необходимо организовать обмен архивными данными между разными SCADA-системами, либо между разными устройствами, на которых установлена MasterSCADA 4D, но при этом эти узлы разрабатываются независимо друг от друга (в разных проектах), то следует использовать возможности OPC UA.

Архив сообщений

Архив сообщений – это архив текстов сообщений, тревог, а также их дополнительных характеристик (источник, время возникновения, время, когда сообщение потеряло свою актуальность, и др.). В архив попадают все Тревоги. Сообщения, сформированные автоматически (например: о действиях оператора, о нарушении границ), попадают в архив также автоматически.

Для просмотра архивов сообщений в MasterSCADA 4D предусмотрен графический элемент – Журнал.

Важно! Прежде чем приступить к настройкам архивирования необходимо убедиться, что устройство, на котором планируется вести архивы, имеет достаточное место для их хранения. Это особенно актуально при использовании исполнительной системы MasterSCADA 4D в контроллерах.

Различают сообщения условные и безусловные. Активные и неактивные.

6.2.1.7.1. ОСНОВНОЙ АРХИВ ДАННЫХ

Основной архив имеется по умолчанию у всех узлов. В панели свойств данного элемента задаются настройки, необходимые для работы архивирования данных. В архив записываются значения тех параметров узла, для которых свойство Архивировать = TRUE.

Архивироваться могут параметры простых типов, а также параметры сложных типов - массивы, структуры. Сложные типы данных хранятся в БД и читаются в виде JSON.

Архив конфигурируется с помощью следующих параметров (см. также Конфигурирование архива):

Основной архив данных	
Общие	
Имя	Основной архив данных
Полное имя	Система.АРМ 1.Архивы.Основной архив данных
Метки	
Комментарий	
Служебные	
Настройки	
Максимальное время хранения	365.00:00:00
Максимальный объем архива (Mb)	0
Число удаляемых элементов в одной операции	10000
Период чистки архива	01:00:00
БД	
Запись по изменению	<input checked="" type="checkbox"/>
Мертвая зона	0
Минимальный период записи	00:00:00
Максимальный период записи	00:00:00
Период задачи записи архива	00:00:01
Интервал между попытками подключения	00:01:00 ...
Максимальное количество одновременных подклк	0
Максимальный размер очереди записи	10000
Период подсчета статистики	01:00:00
Синхронизация БД при резервировании	<input checked="" type="checkbox"/>
Количество потоков записи	1
Сервер	
Пользователь	
Пароль	
Порт	0
Параметры подключения	
Тип БД	sqlite3

Название	Тип	Назначение
Максимальное время хранения	TIME	Максимальное время хранения данных в архиве. Данные, которые изменились раньше, будут удаляться из базы данных. За один раз удалится данных не более, чем указано в настройке Число удаляемых элементов в одной операции. Период, с которым запускается очистка архива, задается в настройке Период чистки архива
Максимальный объем архива (Mb)	LREAL	Задается максимальный размер архива, при превышении которого более ранние данные будут удаляться из архива. При разработке проекта необходимо учитывать, чтобы эта настройка позволяла обеспечить хранение данных за весь период. Т.к. данные пишутся по изменению, то заранее рассчитать размер, который будет иметь архив в результате работы проекта в режиме исполнения, нельзя. Как правило, эта настройка выставляется на этапе пусконаладочных работ. Разработчик проекта оценивает насколько вырос размер архива за определенный промежуток времени и рассчитывает примерный итоговый размер архива. Если задано значение 0, то это означает, что ограничения на размер архива отсутствуют.
Число удаляемых элементов в одной операции	DINT	Количество элементов (данных), которые удаляются из архива за один раз;
Период чистки архива	TIME	Если в архиве содержатся данные старше, чем указано в поле Максимально время хранения, то старые данные будут удаляться с этим периодом. За один раз удалится количество данных не более, чем указано в поле Число удаляемых элементов в одной операции;
БД	STRING	Место хранения архива. Если это поле остается пустым, то архив пишется в папку того проекта, который загружается в исполнительную систему. Для некоторых predetermined узлов (контроллеров), по умолчанию задается место хранения архивов на выделенной Flash-карте;

Название	Тип	Назначение
Запись по изменению (Ch)	BOOL	<p>Определяет способ записи данных в архив. Если флаг установлен, то в архив будут попадать все данные, изменившиеся на величину большую, чем указано в настройке Мертвая зона, но не чаще, чем указано в настройке Минимальный период записи.</p> <p>Если флаг снят, то будет использоваться периодическая запись данных в архив с периодом, указанным в поле Максимальный период записи (если только значение не равно нулю).</p>
Мертвая зона (Deadband)	LREAL	<p>Значение не записывается в архив, если оно изменяется на величину меньшую, чем задано данным параметром.</p> <p>Данная настройка влияет только на вещественные параметры. Используется только если установлен флаг Запись по изменению. Применяется для параметров, для которых не назначен шаблон архивирования.</p>
Минимальный период записи (Tmin)	TIME	<p>Значения пишутся в архив не чаще, чем 1 раз за данный период. При нулевых значениях, указанных в данном поле, настройка не влияет на работу БД. Применяется для параметров, для которых не назначен шаблон архивирования.</p>
Максимальный период записи (Tmax)	TIME	<p>Значения пишутся в архив с данным периодом, даже если величина значения не менялась, но при этом менялась метка времени. При нулевом значении, указанном в данном поле, настройка не влияет на работу БД. Применяется для параметров, для которых не назначен шаблон архивирования.</p>
Период задачи записи архива	TIME	<p>Период записи в архив при периодической записи.</p>
Интервал между попытками подключения	TIME	<p>Интервал, с которым осуществляется повторное подключение в случае неудачного подключения к БД</p>

Название	Тип	Назначение
Максимальное количество одновременных подключений	DINT	<p>Определяет количество подключений к БД, которые создаются средой исполнения. Если установлено значение 0, то используется стандартное значение в зависимости от типа БД (1 для SQLITE, 3 для остальных). Для SQLITE вся работа между средой исполнения и базой данных идет из одного потока, через одно подключение. Для остальных БД по умолчанию выделяется как минимум 3 потока (каждый со своим подключением):</p> <ul style="list-style-type: none"> • Чтение (если указано > 3 потоков, то чтение может выполняться одновременно из нескольких потоков) • Запись • Удаление старых записей
Максимальный размер очереди записи	DINT	<p>Определяет количество значений, которые хранятся в оперативной памяти в ожидании записи в БД. Используется в случае, если генерация данных идет быстрее, чем их запись, либо в случае, если пропала связь с БД. Если в памяти накопилось значений больше, чем указано в данном поле, то происходит прореживание данных.</p>
Период подсчета статистики	TIME	<p>Определяет период, с которым происходит дефрагментация БД, определение ее объема. Если объем превышает заданный, то определяется количество значений, которые надо удалить (исходя из общего числа в БД). Затем порционно удаляется необходимое количество. (Только для БД Postgre)</p>
Синхронизация БД при резервировании	BOOL	<p>Определяет необходимость синхронизировать БД при использовании двух резервируемых серверов. Если флаг установлен, то синхронизация включена.</p> <p>По умолчанию флаг установлен. Синхронизацию выключают как правило в два случая:</p> <ul style="list-style-type: none"> • Если БД синхронизируется средствами самой БД, либо при помощи других вспомогательных средств.

Название	Тип	Назначение
		<ul style="list-style-type: none"> БД установлена на отдельном сервере, и обе резервируемые исполнительные системы записывают в эту внешнюю базу данных
Количество потоков записи	DINT	<p>Определяет какое количество потоков будет использоваться для работы с базой данных. По умолчанию 1 (в случае если установили значение 0, используется также 1). Если указано больше, то создается заданное число потоков, в каждом отдельное подключение к БД, через которое идет запись данных.</p> <p>Для Sqlite не поддерживается.</p>
Сервер	STRING	При использовании удаленной базы данных, указывается адрес компьютера, на котором находится база данных. (IP-адрес, либо MAC-адрес)
Пользователь	STRING	Параметр подключения к БД, если БД требует аутентификацию.
Пароль	STRING	Пароль доступа к базе данных.
Параметры подключения	STRING	
Тип БД		База данных выбирается из списка . В комплект поставки входит база данных SQLITE. При выборе других баз данных необходимо убедиться, что база данных установлена на устройство.

Пусть DV – разница текущего значения и последнего записанного, DT – разница текущего времени и времени последней записи. Тогда условие записи в архив выглядит примерно так:

$$(Ch \ \&\& \ DV \geq \text{Deadband} \ \&\& \ (Tmin==0 \ || \ DT > Tmin)) \ || \ (Tmax \ != \ 0 \ \&\& \ Tmax < DT)$$

Основной архив сообщений

Основной архив сообщений имеется по умолчанию у всех узлов. В панели свойств данного элемента задаются настройки, необходимые для работы архивирования. В архив записываются информация обо всех сообщениях и тревогах узла.

Архив конфигурируется с помощью следующих параметров

Основной архив сообщений	
Общие	
Имя	Основной архив сообщений
Полное имя	Система.АРМ 1.Архивы.Основной архив сообщений
Метки	
Комментарий	
Служебные	
Настройки	
Максимальное время хранения	365.00:00:00
Максимальное количество одновременных подключений	0
Максимальный объем архива (Mb)	0
Число удаляемых элементов в одной операции	1000
Период чистки архива	01:00:00
БД	
Период задачи записи архива	00:00:00
Интервал между попытками подключения	00:01:00
Максимальный размер очереди записи	10000
Период подсчета статистики	01:00:00
Синхронизация БД при резервировании	<input checked="" type="checkbox"/>
Сервер	
Пользователь	
Пароль	
Порт	0
Параметры подключения	
Тип БД	sqlite3

Описание свойств:

Название	Тип	Назначение
Максимальное время хранения	TIME	Максимальное время хранения сообщений в архиве. Сообщения, которые изменились раньше, будут удаляться из базы данных. За один раз удалится сообщений не более, чем указано в настройке Число удаляемых элементов в одной операции. Период, с которым

Название	Тип	Назначение
		запускается чистка архива, задается в настройке Период чистки архива.
Максимальный объем архива (Mb)	LREAL	Задается максимальный размер архива, при превышении которого более ранние сообщения будут удаляться из архива. При разработке проекта следует учитывать, чтобы эта настройка позволяла обеспечить хранение сообщений за весь период. Т.к. сообщения пишутся по изменению, то заранее рассчитать, какой размер будет иметь архив в результате работы проекта в режиме исполнения, нельзя. Как правило, эта настройка выставляется на этапе пусконаладочных работ, разработчик проекта оценивает насколько вырос размер архива за определенный промежуток времени и рассчитывает примерный итоговый размер архива. Если задано значение 0, то это означает, что ограничения на размер архива отсутствуют. (Только для БД Postgre, sqlite)
Число удаляемых элементов в одной операции	DINT	Количество элементов (сообщений), которые удаляются из архива за один раз;
Период чистки архива	TIME	Если в архиве содержатся сообщения старше, чем указано в поле Максимально время хранения, то старые сообщения будут удаляться с этим периодом . За один раз удалится количество сообщений не более, чем указано в поле Число удаляемых элементов в одной операции;
БД	STRING	Место хранения архива. Если это поле остается пустым, то архив пишется в папку того проекта, который загружается в исполнительную систему. Для некоторых predetermined узлов (контроллеров), по умолчанию задается место хранения архивов на выделенной Flash-карте;

Название	Тип	Назначение
Период задачи записи архива	TIME	Период записи в архив при периодической записи. При периодической записи в архив записываются только изменившиеся сообщения, с заданным периодом.
Интервал между попытками подключения	TIME	Интервал, с которым осуществляется повторное подключение в случае неудачного подключения к БД
Максимальный размер очереди записи	DINT	Определяет количество значений, которые хранятся в оперативной памяти в ожидании записи в БД. Используется в случае, если генерация данных идет быстрее, чем их запись, либо в случае, если пропала связь с БД. Если в памяти накопилось значений больше, чем указано в данном поле, то новые сообщения будут игнорироваться.
Максимальное количество одновременных подключений	DINT	Определяет количество подключений к БД, которые создает среда исполнения. Если установлено значение 0, то используется стандартное значение в зависимости от типа БД (1 для SQLITE, 3 для остальных). Для SQLITE вся работа между средой исполнения и базой данных идет из одного потока, через одно подключение. Для остальных БД по умолчанию выделяется как минимум 3 потока, каждый со своим подключением: <ul style="list-style-type: none"> • Чтение (если указано > 3 потоков, то чтение может выполняться одновременно из нескольких потоков) • Запись • Удаление старых записей
Сервер	STRING	При использовании удаленной базы данных, указывается адрес компьютера, на котором находится база данных. (IP-адрес, либо MAC-адрес)

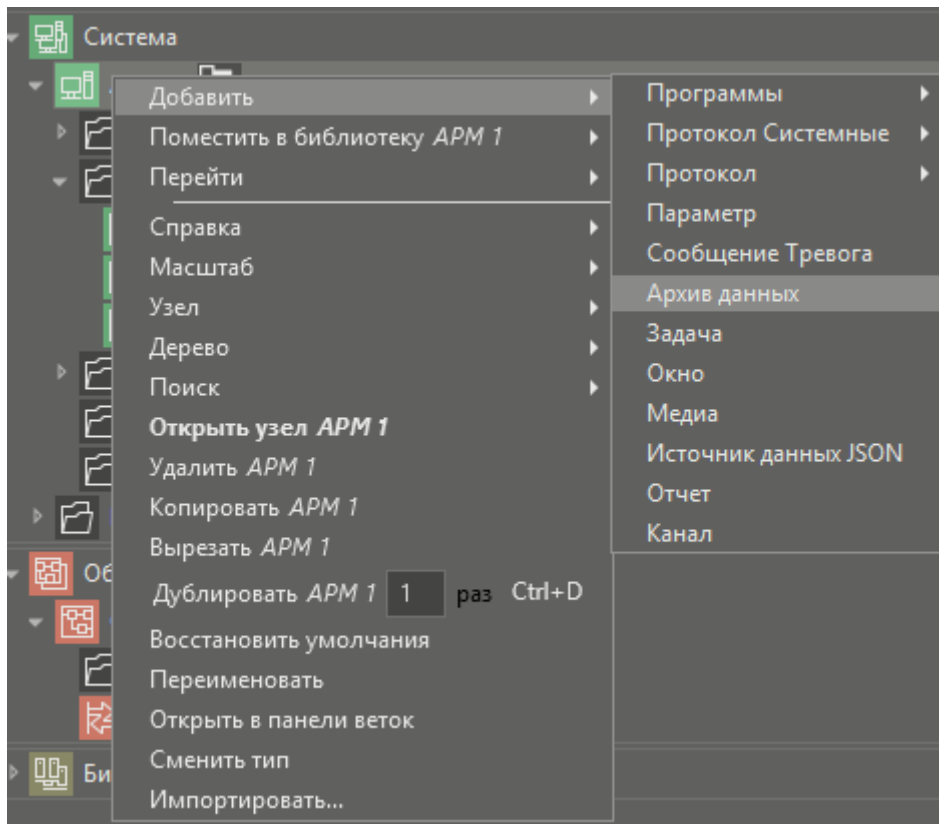
Название	Тип	Назначение
Пользователь	STRING	Параметр подключения к БД, если БД требует аутентификацию.
Пароль	STRING	Параметр подключения к БД, если БД требует аутентификацию.
Порт	DINT	Указывается TCP/IP порт для работы с базой данных. При использовании брандмауэров и антивирусов необходимо разрешить использование данного порта.
Параметры подключения		
Тип БД		Выбирается из списка база данных. В комплект поставки входит база данных SQLITE. При выборе других баз данных нужно убедиться, что база данных установлена на устройство.
Период подсчета статистики	TIME	<p>Определяет период, с которым происходит дефрагментация БД, определение ее объема.</p> <p>Если объем превышает заданный, то определяется количество сообщений, которые надо удалить (исходя из общего числа в БД). Затем порционно удаляется необходимое количество. (Только для БД Postgre)</p>
Синхронизация БД при резервировании	BOOL	<p>Определяет необходимость синхронизировать БД при использовании двух резервируемых серверов. Если флаг установлен, то синхронизация включена.</p> <p>По умолчанию флаг установлен. Синхронизацию выключают как правило в два случая:</p> <ul style="list-style-type: none"> • Если БД синхронизируется средствами самой БД, либо при помощи других вспомогательных средств. • БД установлена на отдельном сервере, и обе резервируемые исполнительные

Название	Тип	Назначение
		системы записывают в эту внешнюю базу данных

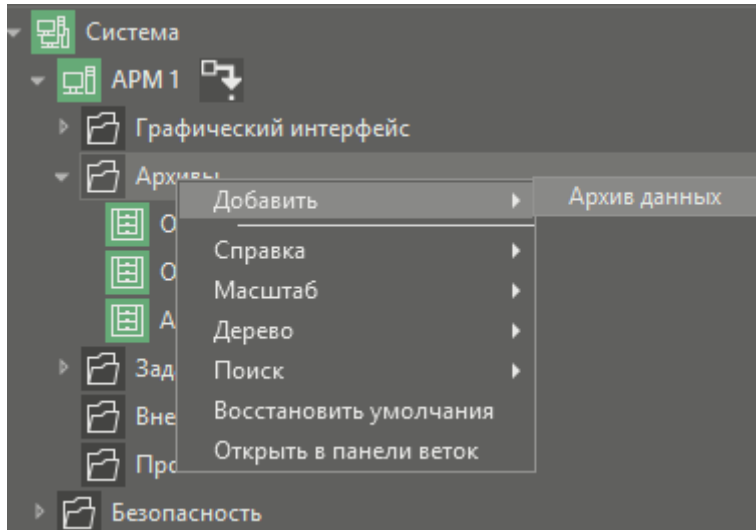
6.2.1.7.2. ЭЛЕМЕНТ АРХИВ ДАННЫХ

Элемент дерева системы Архив данных добавляется в проект в случае, если необходимо обеспечить, чтобы значения параметров разных объектов хранились в разных базах данных.

Добавить элемент в проект можно при помощи контекстного меню узла:

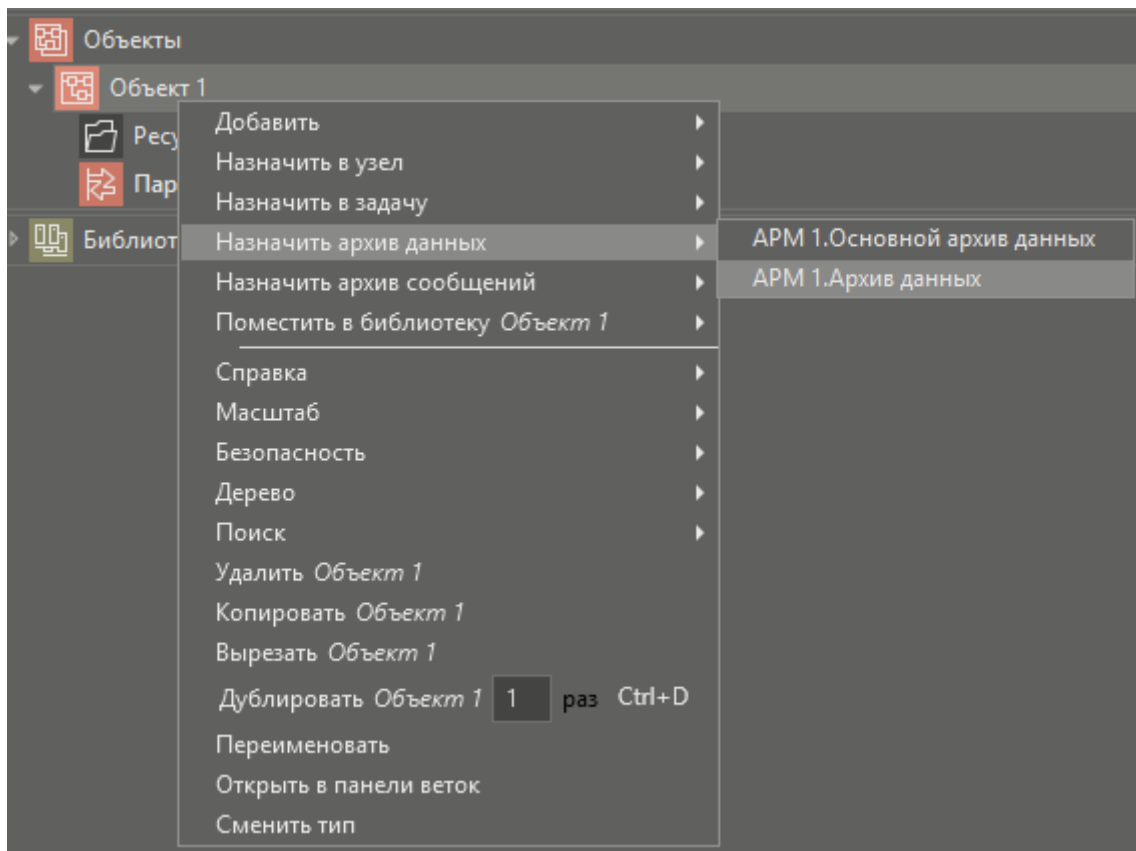


Либо при помощи контекстного меню группы узла Архивы:

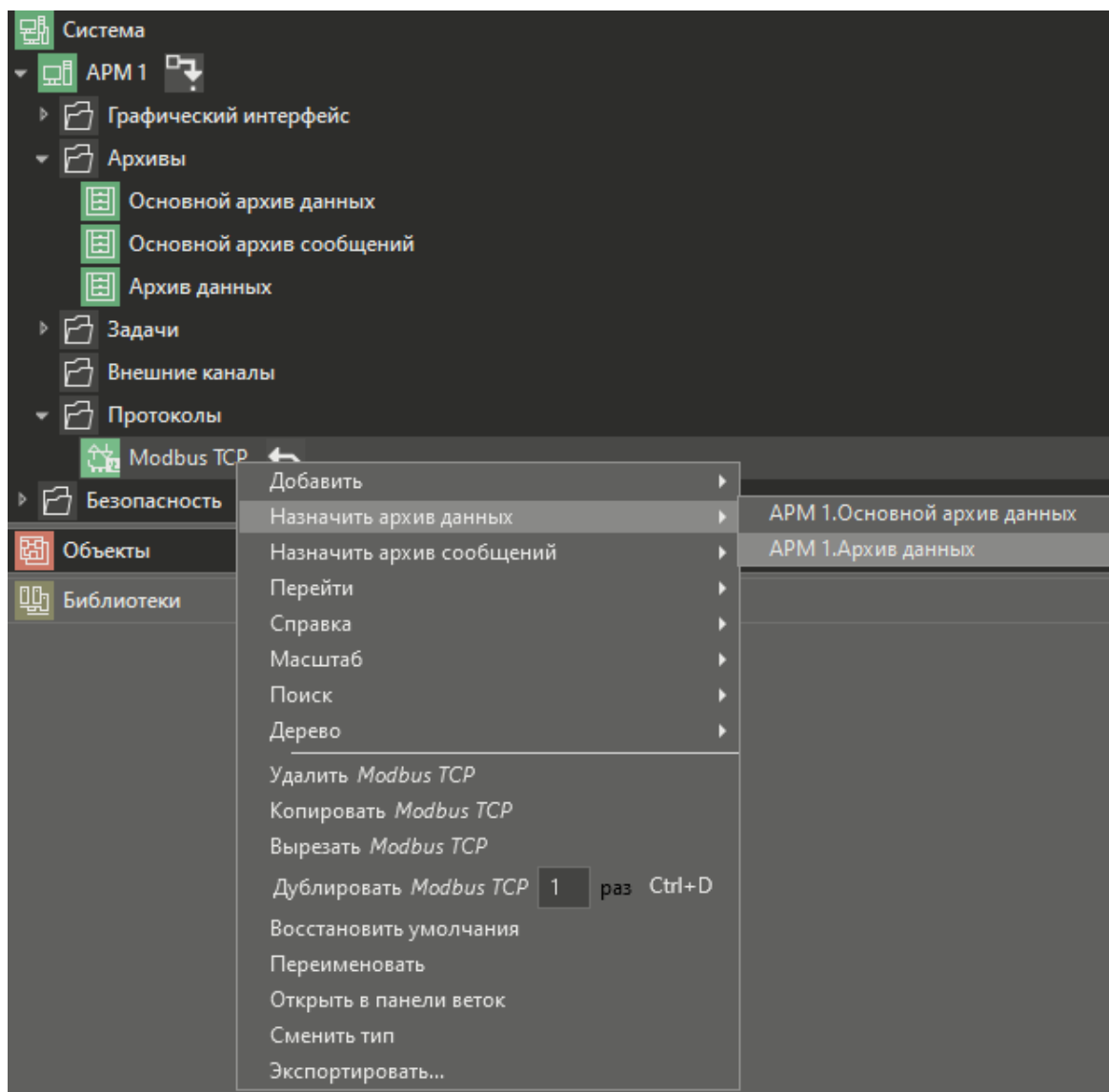


Настройки данного элемента соответствуют настройкам элемента Основной архив данных

Чтобы изменить место архивирования параметров объекта по умолчанию, необходимо в контекстном меню объекта выбрать соответствующий пункт:



Чтобы изменить место архивирования параметров каналов протокола по умолчанию, необходимо в контекстном меню объекта выбрать соответствующий пункт:

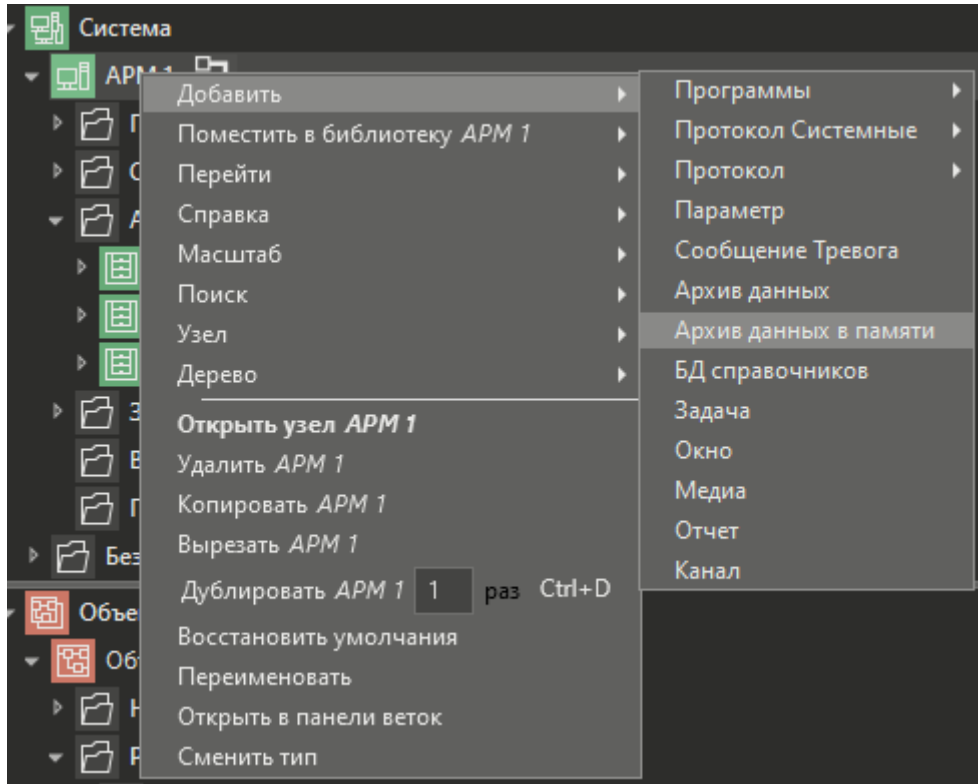


Если архивы протокола или объекта хранятся не в основном архиве данных, то рядом с этим элементом появляется обозначение

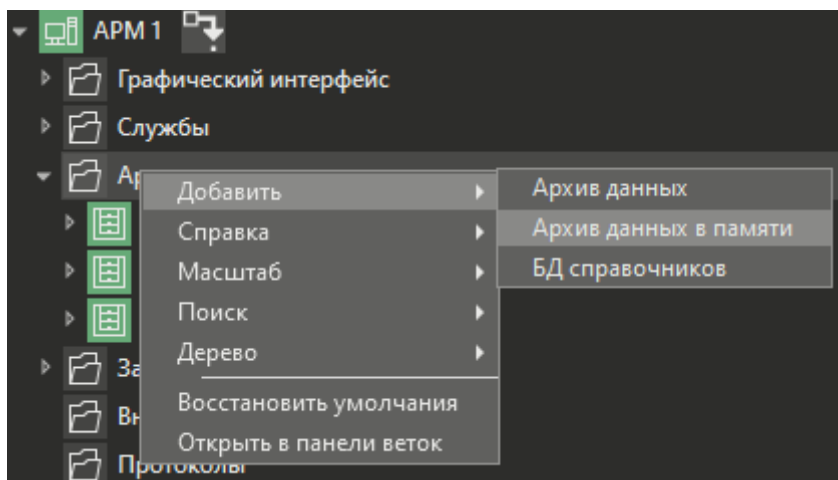
6.2.1.7.3. АРХИВ ДАННЫХ В ПАМЯТИ

Элемент дерева системы Архив данных в памяти обеспечивает хранение в оперативной памяти значений параметров объектов, а также значений каналов протоколов, которые назначены в данный архив в соответствии с заданными ограничениями по времени/количеству значений

Добавить элемент в проект можно при помощи контекстного меню узла:



Либо при помощи контекстного меню группы узла Архивы:



Панель свойств элемента:

Архив данных в памяти

Общие

Имя: Архив данных в памяти

Полное имя: Система.APM 1.Архивы.Архив данных в памяти

Метки:

Комментарий:

Служебные

Настройки

Максимальное время хранения: 01:00:00

Максимальное количество значений: 0

Запись по изменению:

Мертвая зона: 0

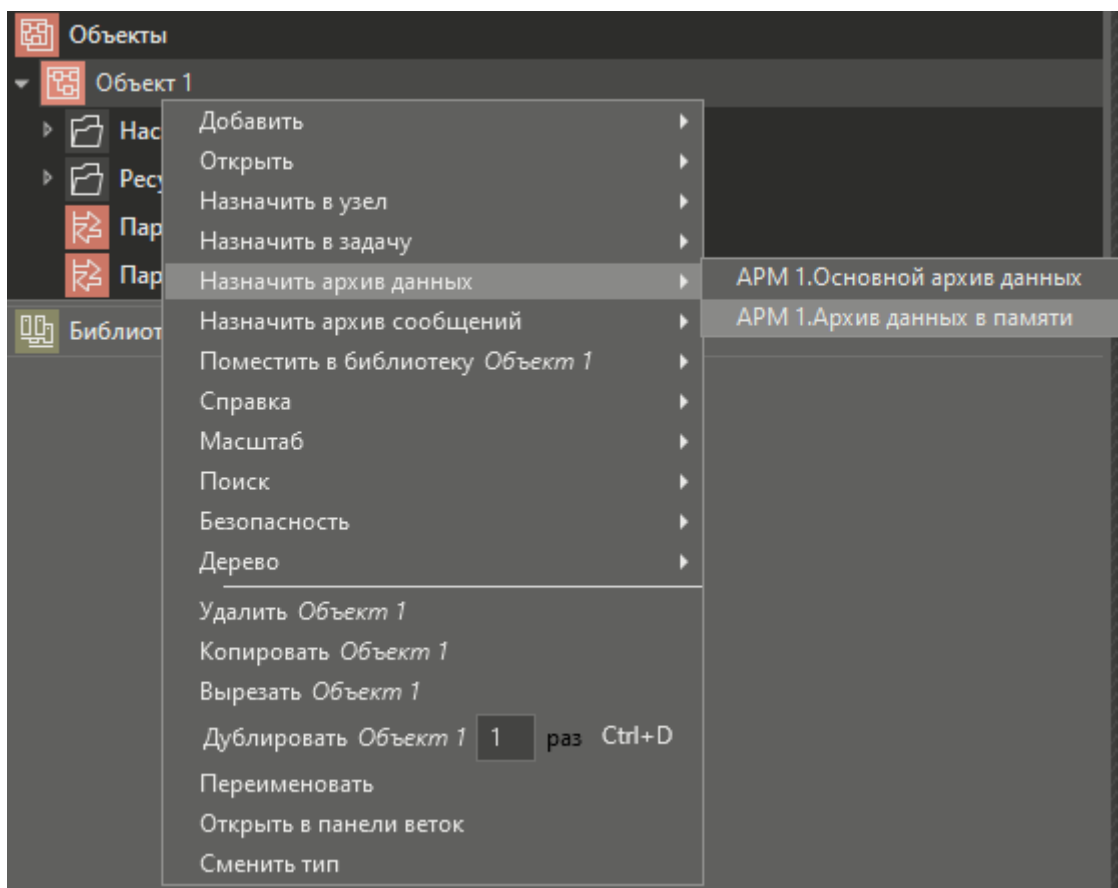
Минимальный период записи: 00:00:00

Максимальный период записи: 00:00:00

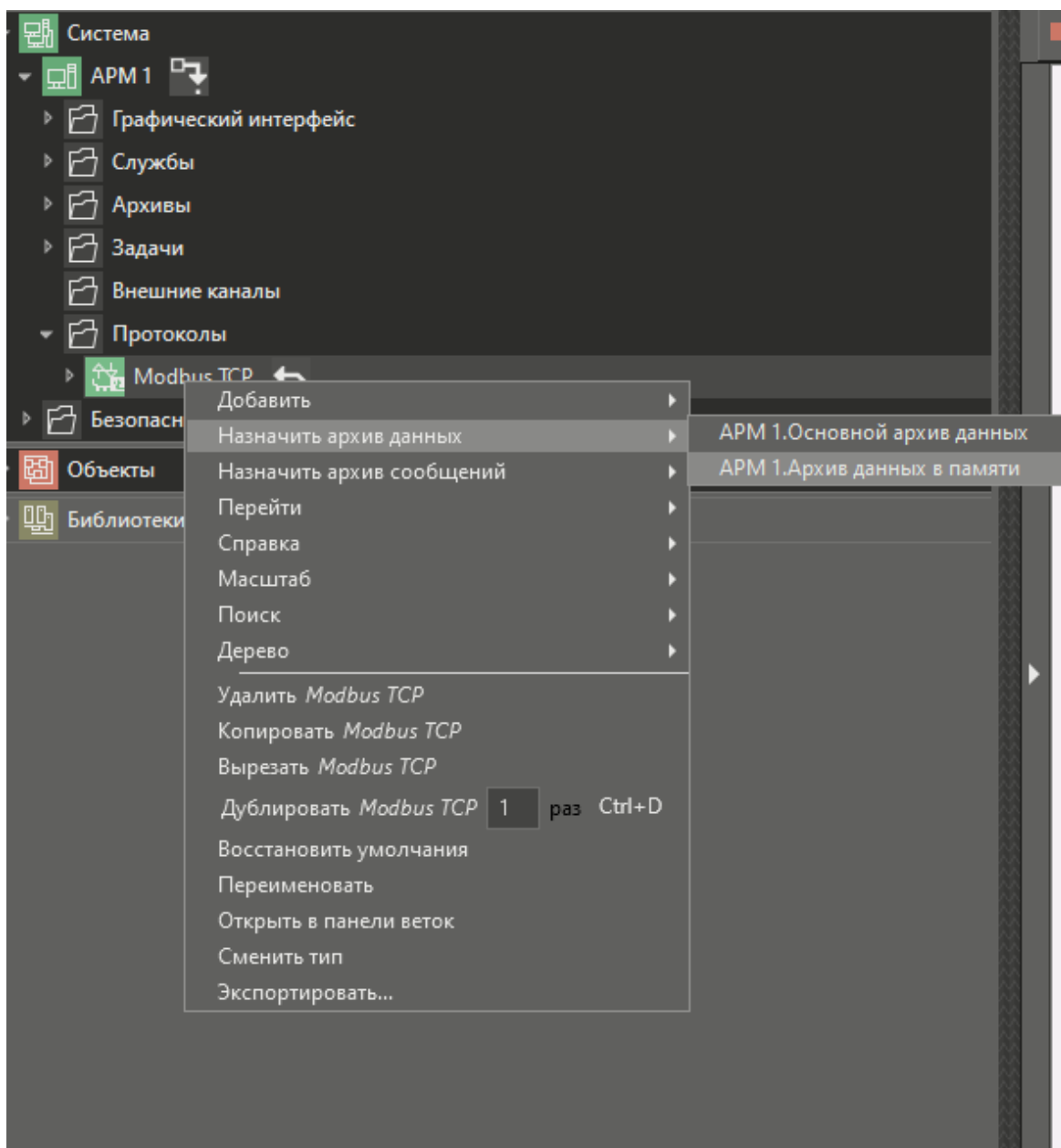
Название	Тип	Назначение
Максимальное время хранения	TIME	Максимальное время хранения данных в архиве. Данные, которые изменились раньше, будут удаляться из памяти
Максимальное количество значений	DINT	Количество хранимых значений по каждому параметру.
Мертвая зона	LREAL	Значение не записывается в архив, если оно изменяется на величину меньшую, чем задано данным параметром. Данная настройка влияет только на вещественные параметры. Применяется для параметров, для которых не назначен шаблон архивирования.
Запись по изменению	BOOL	TRUE – значения записываются в архив при их изменении (рекомендованная настройка); FALSE – значения записываются в архив периодически. Применяется для параметров, для которых не назначен шаблон архивирования.


Название	Тип	Назначение
Минимальный период записи	TIME	Значения пишутся в архив не чаще, чем 1 раз за данный период. При нулевых значениях, указанных в данном поле, настройка не влияет на работу БД. Применяется для параметров, для которых не назначен шаблон архивирования.
Максимальный период записи	TIME	Значения пишутся в архив с данным периодом, даже если они не менялись. При нулевых значениях, указанных в данном поле, настройка не влияет на работу БД. Применяется для параметров, для которых не назначен шаблон архивирования.

Чтобы изменить место архивирования параметров объекта по умолчанию, необходимо в контекстном меню объекта выбрать соответствующий пункт:



Чтобы изменить место архивирования параметров каналов протокола по умолчанию, необходимо в контекстном меню объекта выбрать соответствующий пункт:

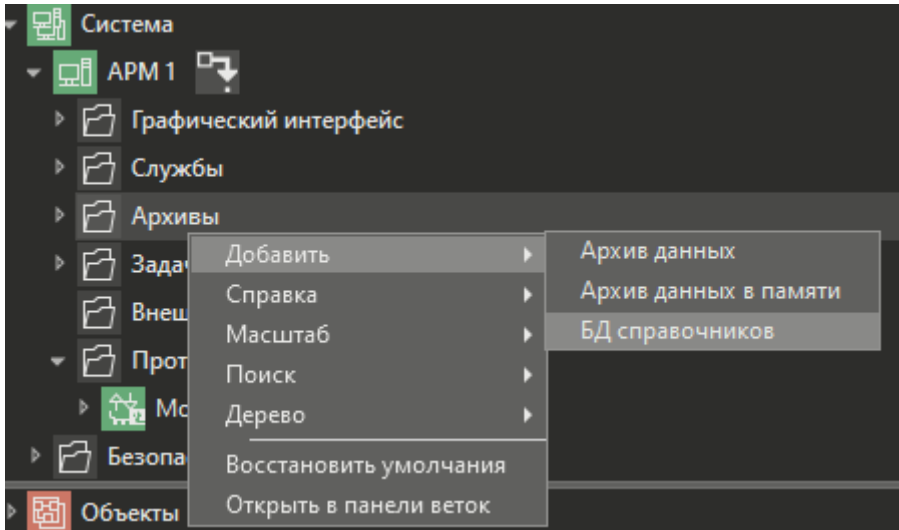


Если архивы протокола или объекта хранятся не в основном архиве данных, то рядом с этим элементом появляется обозначение 

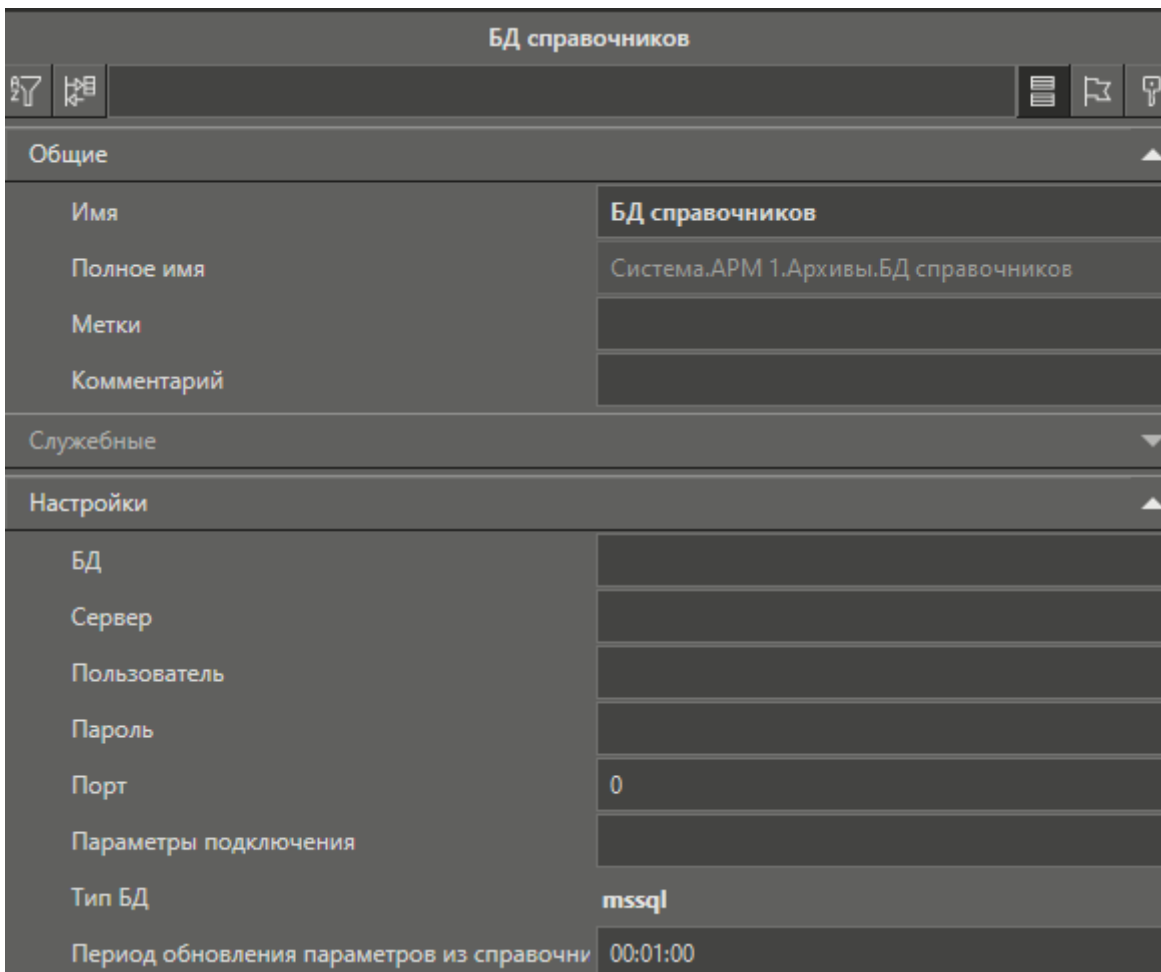
6.2.1.7.4. БД СПРАВОЧНИКОВ

Элемент служит для настройки связи с базой данных, с которой будет взаимодействовать графический элемент Справочник.

Добавить элемент в проект можно при помощи контекстного меню самого узла или его элемента Архивы:



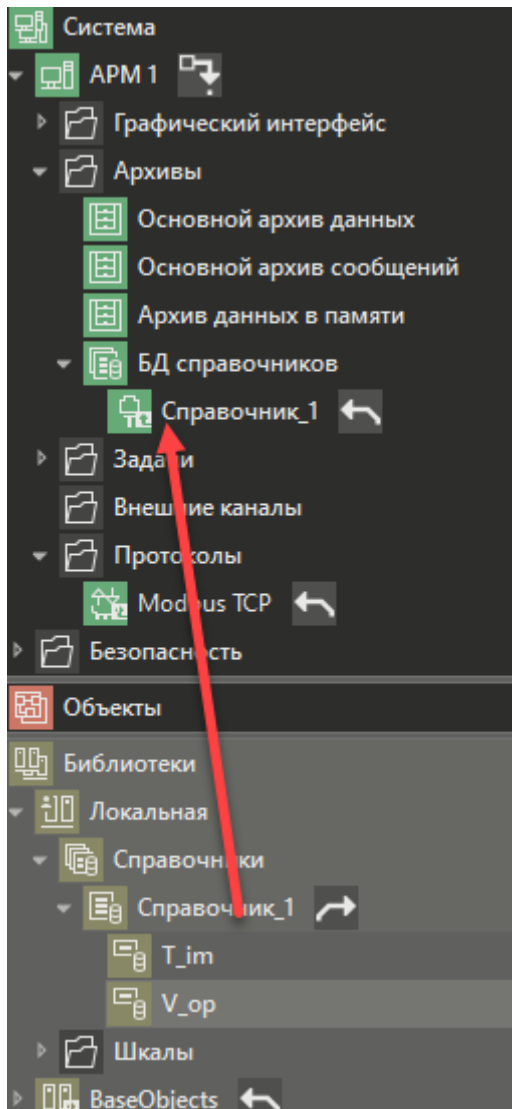
Вид панели свойств:



Описание основных свойств:

Название	Тип	Назначение
БД	STRING	Указывается имя базы данных
Сервер	STRING	Указывается адрес компьютера, на котором находится база данных. (IP-адрес, либо MAC-адрес)
Пользователь	STRING	Параметр подключения к БД, если БД требует аутентификацию.
Пароль	STRING	Пароль доступа к базе данных.
Порт	DINT	Указывается TCP/IP порт для работы с базой данных. При использовании брандмауэров и антивирусов необходимо разрешить использование данного порта.
Параметры подключения	STRING	
Тип БД		База данных выбирается из списка.
Период обновления параметров из справочника	TIME	Период обновления значений полей, получаемых из базы данных

После настройки свойств необходимо установить связь с библиотечным элементом Справочник, в котором должны быть добавлены поля, которые необходимо получать из базы данных. Для этого необходимо перетащить элемент из библиотеки на элемент БД справочников. В результате появится ссылка на библиотечный элемент в дереве



Важно! Набор полей таблицы базы данных должна соответствовать набору полей в Справочнике. Если выбранный справочник ссылается на другие справочники, то в данной БД должны присутствовать таблицы для всех этих справочников

Рекомендации при использовании PostgreSQL

При создании таблиц не используйте графический конструктор, создавать нужно используя скрипты.

6.2.1.7.5. БАЗЫ ДАННЫХ

Для хранения архива данных могут быть использованы следующие базы данных:

- SQLITE;
- POSTGRESQL;
- MS SQL.

Существует также техническая возможность подключить следующие БД:

- DB2;
- Firebird;
- ODBC;
- Oracle;
- MySQL.

Сроки подключения и особенности работы можно узнать в технической поддержке.

Важно! Если в начале работы проекта в исполнительной системе была выбрана база данных SQLite, а затем потребовалось перейти на другие базы данных (Postgre или MS SQL), и при этом сохранить накопленные ранее архивы данных и сообщений, то это возможно сделать специальной программой. Программа может быть выслана по запросу в техническую поддержку

6.2.1.7.5.1. SQLITE

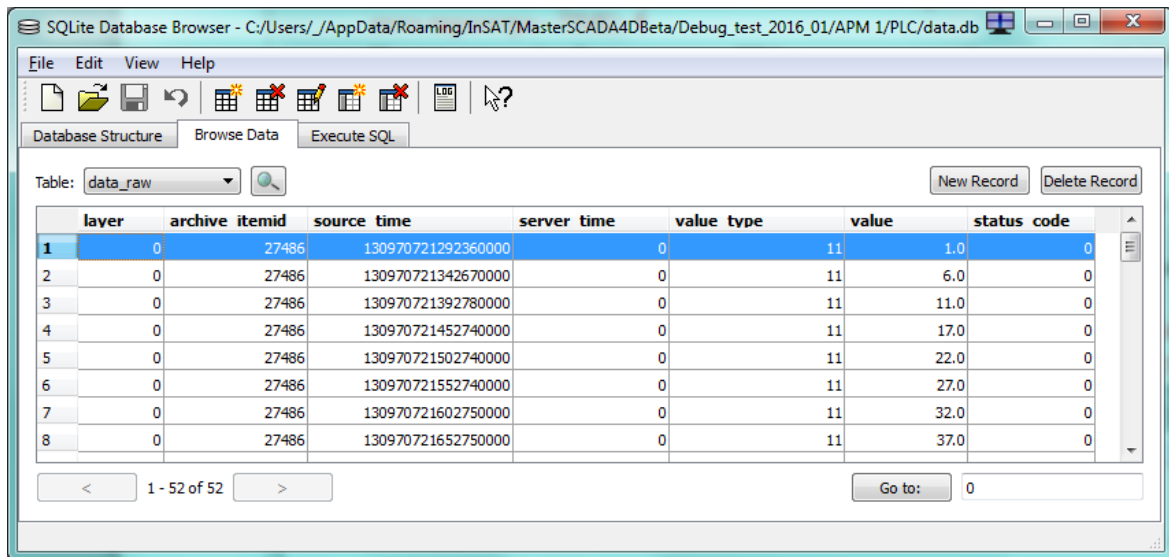
База данных **SQLITE3** входит в состав MasterSCADA 4D. Эта база данных используется для хранения архивов данных и сообщений по умолчанию. При работе с демоверсией среды исполнения, которая входит в состав среды разработки, файл с архивом данных можно найти по адресу: *<профиль пользователя>\AppData\Roaming\MPSSoft\<продукт>\Debug_<имя проекта>\<имя узла>\PLC\data.db*.

Важно! База данных может находиться только на том же устройстве, на котором установлена среда исполнения.

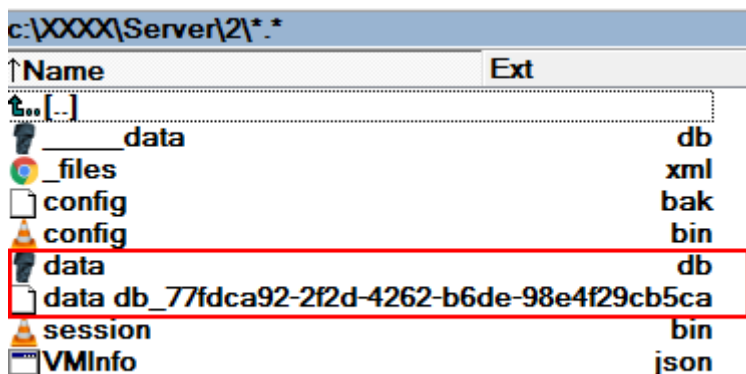
Вид панели свойств элемента Основной архив, если данные хранятся в SQLITE3:

Основной архив	
Общие	
Имя	Основной архив
Метки	
Комментарий	
Служебные	
Настройки	
Максимальное время хранения	60.00:00:00
Максимальный объем архива (Mb)	0
Число удаляемых элементов в одной о	1000
Период чистки архива	01:00:00
БД	
Запись по изменению	<input checked="" type="checkbox"/>
Мертвая зона	0
Минимальный период записи	00:00:00
Максимальный период записи	00:00:00
Период задачи записи архива	00:00:00
Сервер	
Пользователь	
Пароль	
Порт	0
Параметры подключения	
Тип БД	sqlite3

Вместе со значением в архив записываются и связанные с этим значением данные – идентификатор параметра, время источника, время сервера, тип значения и статус:



При загрузке конфигурации, у имеющегося архива проверяется ID-проекта, и, если этот ID отличается от ID текущего проекта, то к имени файла архива добавляется `<id проекта>` и создается новый файл:



6.2.1.7.5.2. POSTGRESQL

Для хранения архивов в базе данных PostgreSQL, ее нужно предварительно установить. Среда исполнения может работать как с локально установленной базой данных, так и с удаленной. Прежде чем приступить к настройке связи с базой данных, необходимо создать пустую базу данных и указать в ее настройках имя пользователя и пароль.

Вид панели свойств элемента Основной архив, если данные хранятся в PostgreSQL:

Настройки	
БД	mplc4_test
Запись по изменению	<input checked="" type="checkbox"/>
Максимальное время хранения	60:00:00:00
Максимальный объем архива (Mb)	0
Максимальный период записи	00:00:00
Мертвая зона	0
Минимальный период записи	00:00:00
Параметры подключения	
Пароль	12345
Период задачи записи архива	00:00:00
Период чистки архива	01:00:00
Пользователь	mplc4_test
Порт	5432
Сервер	10.0.0.139
Тип БД	postgresql
Число удаляемых элементов в одной операции	1000

Для корректной работы необходимо настроить следующие свойства:

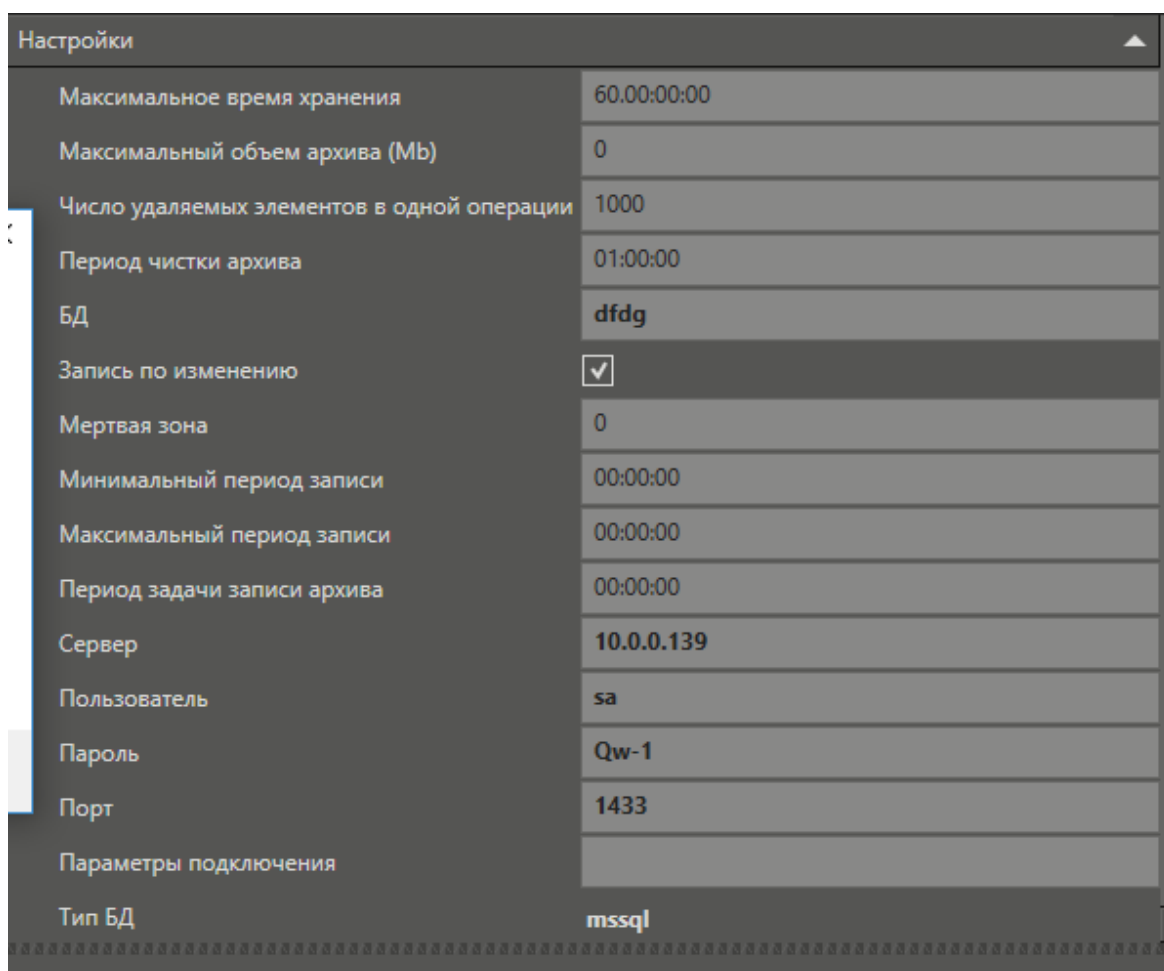
Свойство	Рекомендация
БД	Имя ранее созданной базы данных PosrgreSQL.
Сервер	Указывается IP-адрес компьютера, на котором установлена база данных.
Пользователь	Имя пользователя, который добавлен в настройках самой базы данных. Соответствует LoginRole на сервер Postgre.
Пароль	Пароль пользователя, который добавлен в настройках самой базы данных.
Порт	Порт TCP IP, который используется для передачи данных между средой исполнения и базой данных. Для работы с базой данных PosrgreSQL используется порт 5432.
Тип БД	Выбирается база данных PosrgreSQL.

При первом подключении к базе данных будут созданы все необходимые таблицы и другие необходимые для корректной работы элементы.

6.2.1.7.5.3. MS SQL

Для хранения архивов в базе данных MS SQL, ее необходимо предварительно установить. Среда исполнения может работать как с локально установленной базой данных, так и с удаленной. Прежде чем приступить к настройке связи с базой данной, необходимо создать пустую базу данных, и указать в её настройках имя пользователя и пароль.

Вид панели свойств элемента Основной архив, если данные хранятся в MS SQL:



Для корректной работы необходимо настроить следующие свойства:

Свойство	Рекомендация
БД	Имя ранее созданной базы данных MS SQL.
Сервер	Указывается IP-адрес компьютера, на котором установлена база данных.
Пользователь	Имя пользователя, который добавлен в настройках самой базы данных.
Пароль	Пароль пользователя, который добавлен в настройках самой базы данных.

Свойство	Рекомендация
Порт	Порт TCP IP, который используется для передачи данных между средой исполнения и базой данных. Для работы с базой данных MS SQL используется порт 1433.
Тип БД	Выбирается база данных mssql.

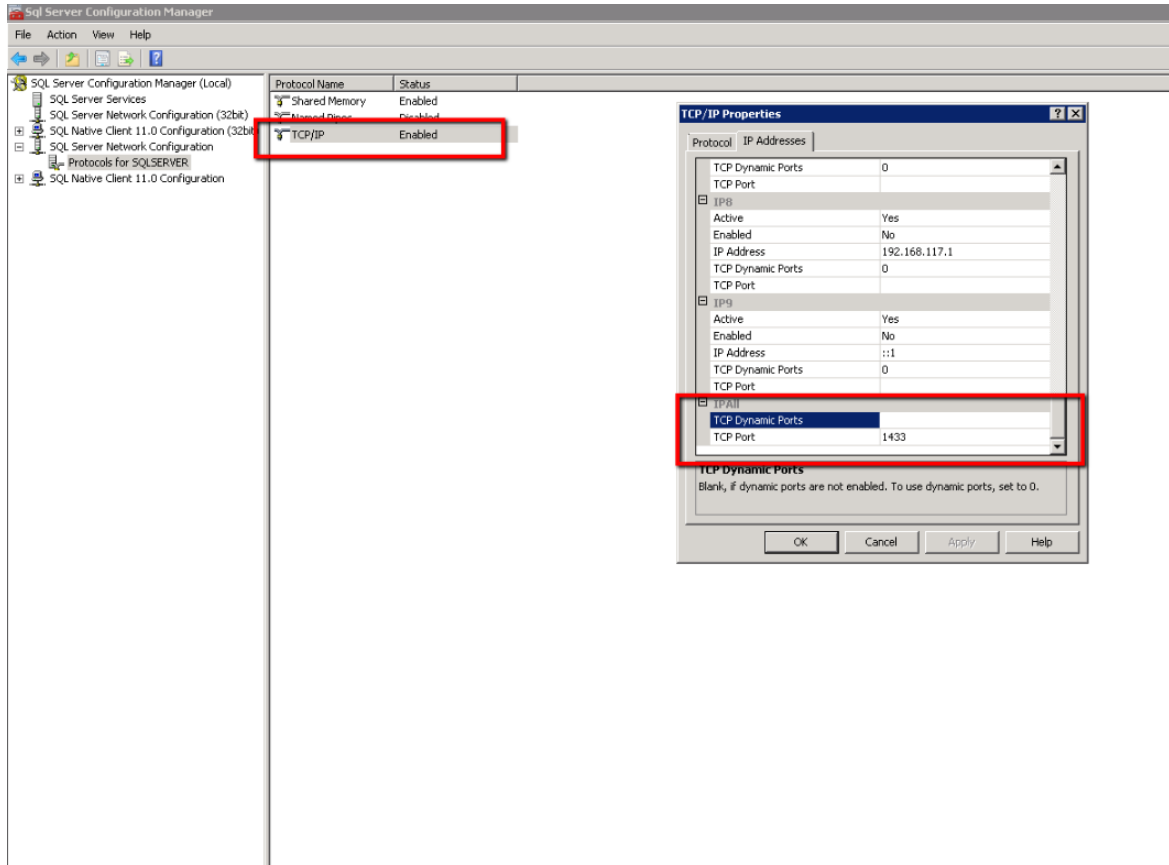
Настройки, которые необходимо сделать непосредственно в базе данных описаны в разделе Настройки базы данных MSSQL

При первом подключении к базе данных будут созданы все необходимые таблицы и другие необходимые для корректной работы элементы.

НАСТРОЙКИ БАЗЫ ДАННЫХ MSSQL

В редакции базы данных SQLEXPRESS, а также в полных редакциях при установке именованного экземпляра, по умолчанию прописан динамический порт равный 0, из-за чего к серверу нельзя подключаться по фиксированному порту 1433, указанному в настройках MasterSCADA 4D. Для корректной работы в настройках базы данных нужно очистить поле, отвечающее за динамический порт, и прописать фиксированный порт 1433. Также нужно проверить что у протокола TCP/IP установлено значение Enabled (в редакции SQLEXPRESS по умолчанию установлено Disabled)

Пример настроек SQL Server Configuration Manager:



6.2.1.7.6. КОНФИГУРИРОВАНИЕ АРХИВА ДАННЫХ

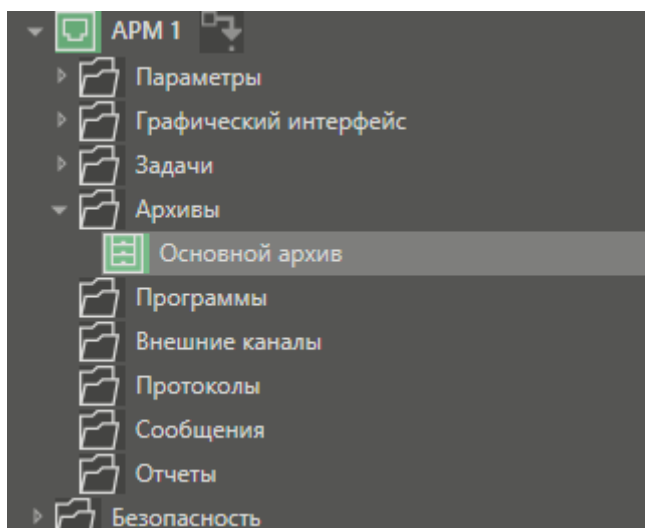
Архивы данных в MasterSCADA 4D хранятся в базах данных.

Прежде чем приступить к разработке проекта, в котором потребуется архивировать данные, необходимо убедиться в том, что устройство, на котором устанавливается среда исполнения MasterSCADA 4D, имеет технические средства для хранения архивов.

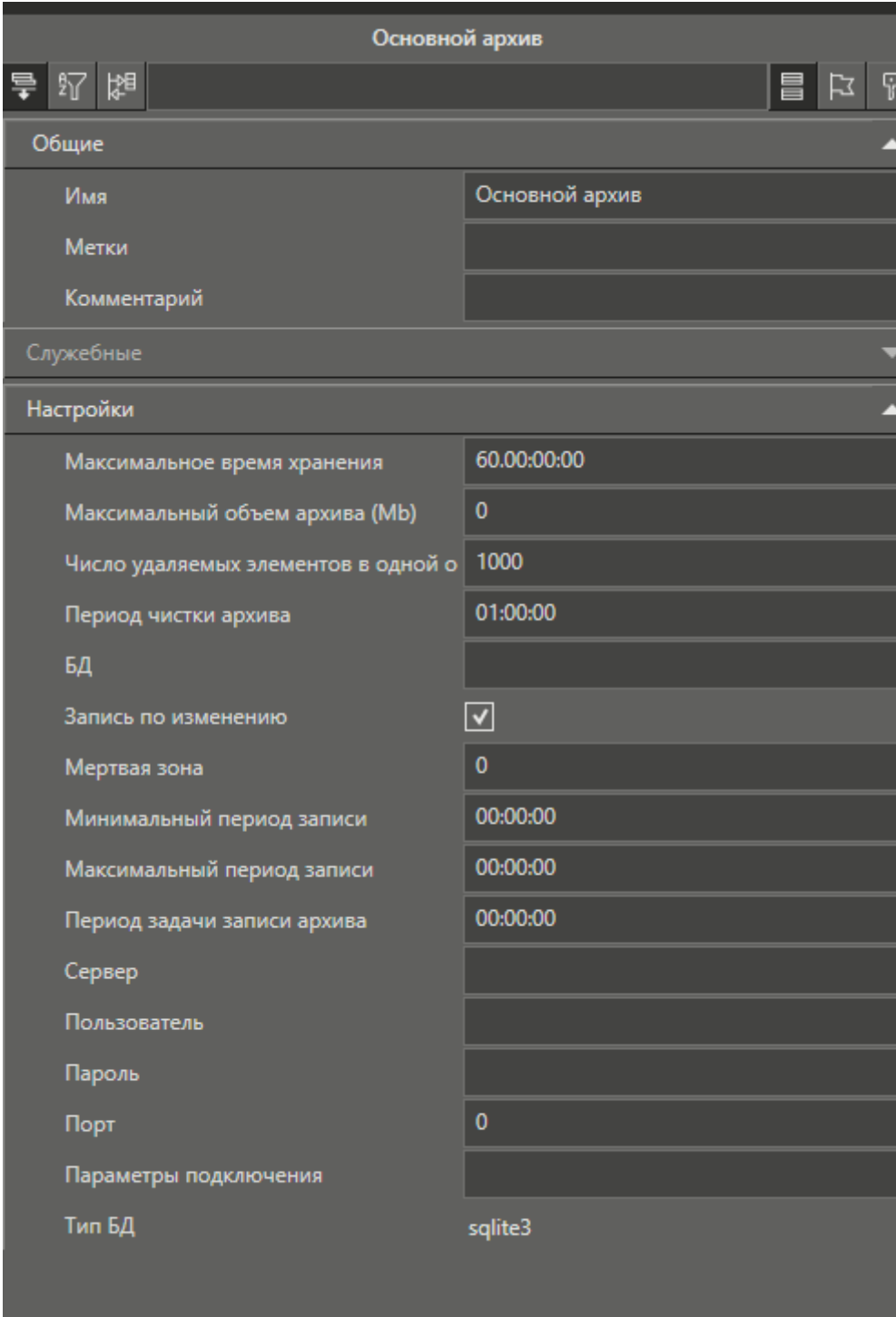
.

В текущей версии могут архивироваться параметры объектов, узлов и каналов, сами теги и каналы, а также настройки. Параметры программ, параметры элементов, исполняемых в задаче экрана, не архивируются.

По умолчанию каждый узел содержит элемент Основной архив в группе Архивы:



В данном элементе определяются основные настройки:



Основной архив	
Общие	
Имя	Основной архив
Метки	
Комментарий	
Служебные	
Настройки	
Максимальное время хранения	60.00:00:00
Максимальный объем архива (Mb)	0
Число удаляемых элементов в одной о	1000
Период чистки архива	01:00:00
БД	
Запись по изменению	<input checked="" type="checkbox"/>
Мертвая зона	0
Минимальный период записи	00:00:00
Максимальный период записи	00:00:00
Период задачи записи архива	00:00:00
Сервер	
Пользователь	
Пароль	
Порт	0
Параметры подключения	
Тип БД	sqlite3

Список параметров, значения которых попадают в архив, определяется настройкой Архивировать этого параметра (тега, канала) :

Параметр 1	
Общие	
Имя	Параметр 1
Программное имя	Параметр 1
Полное имя	Объекты.Объект 1.Параметр 1
Метки	
Комментарий	
Сохранять	Наследуется
Доступ	Чтение/Запись
Начальное значение	0
Тип значения	LREAL
Служебные	
Архивирование	
Архивировать	<input checked="" type="checkbox"/>
Шаблон архивирования	
Разрешения	
Шкала	

Если требуется обеспечить хранение разных параметров с разными настройками, то необходимо создать шаблоны архивирования.

Вместе со значением в архив автоматически записываются связанные с этим значением данные – идентификатор параметра, время источника, время сервера, тип значения и статус.

Для просмотра архивов данных в графическом клиенте можно использовать График.

Если требуется обеспечить хранение архивов объектов в разных БД, например, значения параметров Объекта 1 должны будут храниться в БД MS SQL, а Объекта 2 - в SQLITE, то необходимо элемент Основной архив данных настроить для хранения архивов в MS SQL, а также добавить в узел элемент Архив данных, и настроить его для хранения архивов в SQLITE, а затем в контекстном меню того объекта, который должен храниться в БД SQLITE, необходимо выбрать соответствующий пункт.

Особенности архивирования каналов и тегов

Если установлен флаг Архивировать у канала, у которого свойство Доступ имеет значение Input или InOut, то будет архивироваться Вход канала. Если установлен флаг Архивировать

у канала, у которого свойство Доступ имеет значение Output, то будет архивироваться Выход канала.

Если установлен флаг Архивировать у тега то будет архивироваться его параметр Выход.

Если канал или тег имеет дополнительные параметры в программах, окнах и т.п., то эти параметры архивироваться не будут, если установлен флаг Архивировать у канала. В случае необходимости, у таких параметров флаг Архивировать нужно установить вручную

Особенности настройки архивирования параметров и каналов с типом данных Массив структур

Чтобы архивировать значения полей структур, из которых состоит массив необходимо в настройках параметра установить флаг Отображать элементы в дереве в категории свойств Общие. Затем выделить нужное поле в дереве и установить индивидуально флаг Архивировать

6.2.1.7.6.1. СЛОЙ ДАННЫХ

Слой данных – это отдельно хранимый набор архивных значений параметра, отличающийся периодом записи, способом предварительной обработки перед записью и длительностью хранения. Основное назначение слоев – повышение быстродействия при извлечении данных для их отображения или обработки. Предполагается, что разработчик проекта заранее знает, что, например, его будут интересовать средние почасовые значения параметра или просмотр ежеминутных текущих значений. Если такие слои заданы на этапе создания проекта, то появляется возможность при запросе этой информации для отображения на тренде (графике) или подготовке отчета извлечь все необходимые данные без поиска и обработки данных. Тем самым значительно повышается быстродействие этих операций.

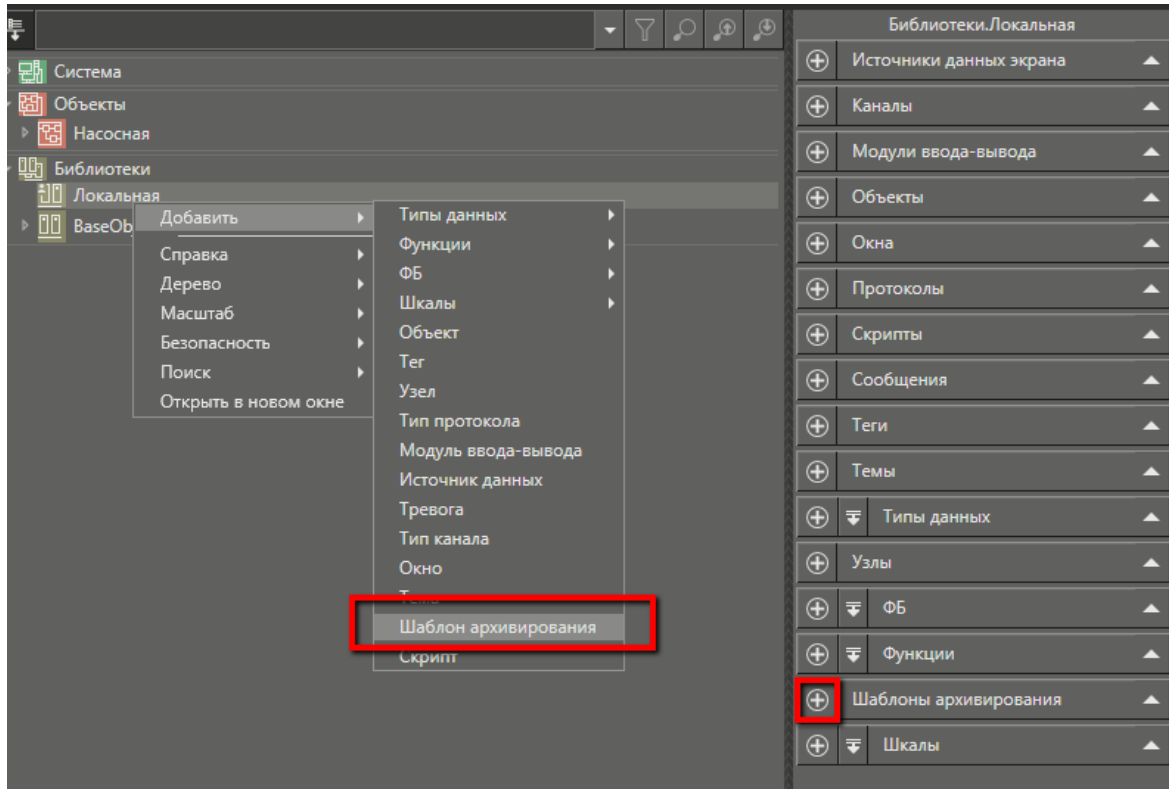
Важно! В текущей версии слои используются при работе с большим объемом данных на трендах автоматически. Настроить слои вручную в данной версии нельзя.

6.2.1.7.6.2. ШАБЛОНЫ АРХИВИРОВАНИЯ

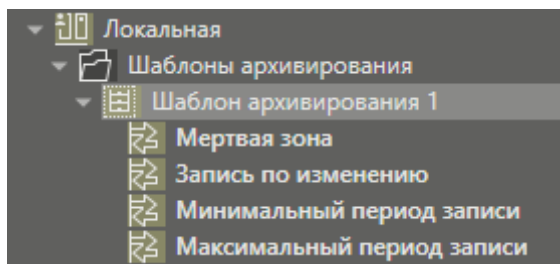
Шаблон архивирования – это библиотечный элемент, содержащий комплекс настроек, которые будут учитываться при архивировании значений параметров. В случае, если шаблон архивирования не назначен параметру, то при архивировании будут использоваться настройки, заданные в настройках элемента Основной архив в группе Архивы. Как правило, шаблоны архивирования используются в том случае, когда разным параметрам требуется обеспечить сохранение в архивах с разными мертвыми зонами.

Создание шаблона архивирования

Создать шаблон архивирования можно в пользовательской библиотеке при помощи контекстного меню, либо в контекстной панели:



В результате получим:



Настроить элемент можно в панели свойств элемента Шаблон архивирования 1:

Шаблон архивирования 1

Общие

Имя: Шаблон архивирования 1

Полное имя: Библиотеки.Локальная.Шаблоны архивирования.Шаблон ар

Метки:

Комментарий:

Служебные

Параметры архивирования

Мертвая зона: 0

Запись по изменению:

Минимальный период записи: 0ms

Максимальный период записи: 0ms

Запись при останове:

Все значения:

Свойство	Тип	Описание
Мертвая зона	LREAL	Значение не записывается в архив, если оно изменяется на величину меньшую, чем задано данным параметром. Данная настройка влияет только на вещественные параметры.
Запись по изменению	BOOL	TRUE – значения записываются в архив при их изменении (рекомендованная настройка); FALSE – значения записываются в архив периодически. Будут записаны все значения параметров типа SYSTEM_*_PARAM, когда изменялось значение (Value).
Минимальный период записи	TIME	Значения пишутся в архив не чаще, чем 1 раз за данный период. При нулевых значениях, указанных в данном поле, настройка не влияет на работу БД.
Максимальный период записи	TIME	Значения пишутся в архив с данным периодом, даже если они не менялись. При нулевых значениях, указанных в данном поле, настройка не влияет на работу БД.

Свойство	Тип	Описание
Запись при останове	BOOL	Если флаг установлен и в данной сессии параметру присваивались значения, то при остановке исполнительной системы последнее значение параметра записывается с текущей меткой времени и признаком качества OprUa_GoodShutdownEvent 0x00A80000. Рекомендуется флаг устанавливать для корректного отображения значений на тренде и для анализа значений параметров в базах данных.
Все значения	BOOL	Если установлен, то идет запись каждого поступающего значения. Будут записаны все значения параметров типа SYSTEM_*_PARAM, когда изменялась метка времени (SourceTime).

Использование шаблона архивирования

Созданный в библиотеке шаблон архивирования может быть выбран в настройках параметров:

Расход 1	
Общие	
Имя	Расход 1
Метки	
Комментарий	
Сохранять	Наследуется
Доступ	Чтение/Запись
Начальное значение	0
Тип значения	LREAL
Служебные	
Архивирование	
Архивировать	<input type="checkbox"/>
Шаблон архивирования	Шаблон архивирования 1
Мертвая зона	12
Запись по изменению	<input checked="" type="checkbox"/>
Минимальный период записи	0ms
Максимальный период записи	0ms
Разрешения	
Всегда отображать в дереве	<input type="checkbox"/>
Шкала	
Шкала	

При необходимости, значения свойств у конкретного параметра можно изменить индивидуально. Если параметру необходимо задать уникальные настройки, которые не будут повторяться у других параметров, то можно выбрать Шаблон архивирования, созданный в Стандартной библиотеке по умолчанию, и в панели свойств параметра переопределить необходимые значения.

6.2.1.7.6.3. ПЕРЕДАЧА АРХИВНЫХ ДАННЫХ

Передача архивных данных между узлами одного проекта

Если два архивируемых параметра принадлежат различным узлам и связаны друг с другом, то архивные значения от источника будут автоматически копироваться в архив приемника. Таким образом, на разных узлах будут создаваться два идентичных архива.

Передача архивных данных другой SCADA

Если требуется передать архивные данные из среды исполнения MasterSCADA 4D в другие SCADA-системы, то используется возможность MasterSCADA 4D выступать в роли OPC UA сервера, либо в роли OPC DA сервера.

Передача архивных данных между параметрами одного узла

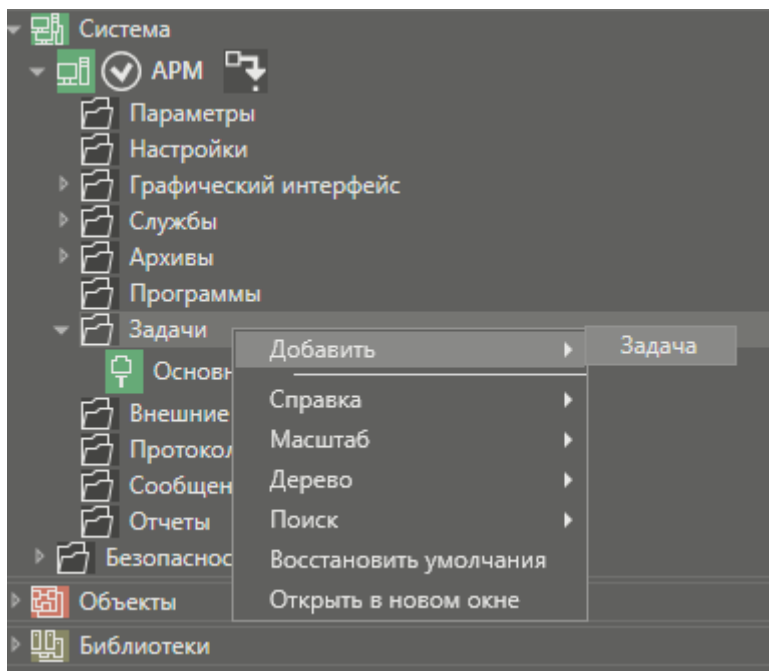
Если неархивируемый параметр связан входящей связью с архивируемым параметром того же узла, то если этот неархивируемый параметр задействован в архивных функциях (на графике или ФБ) то архив будет браться со связанного архивируемого параметра. Допускается только одна связь.

Если два архивируемых параметра связаны между собой в рамках одного узла, то у каждого параметра будет свой, независимый от связанной переменной, архив.

6.2.1.8. ЗАДАЧИ

Данная группа элементов предназначена для настройки задач, связанных с работой объектов, программ, созданных в узле и объектах.

Созданный пользователем в среде разработки проект перед загрузкой в среду исполнения преобразуется (компилируется). В результате компиляции вся логика работы проекта, независимо от того на каком языке она была написана разработчиком проекта, преобразуется в специальные программы, которые будут загружаться в устройство и там циклически исполняться. Количество программ зависит от количества задач, созданных в дереве системы в данной группе. По умолчанию, у каждого узла при его создании создается только одна задача - Основная задача. Разработчик проекта может создать столько задач, сколько необходимо для работы конкретного проекта. Особенности вычисления, например, цикл вычисления, будет зависеть от настроек той или иной задачи. Каждая задача будет работать независимо от других задач в устройстве. Создание нескольких задач будет целесообразно в том случае, если необходимо обеспечить разные циклы вычисления разных программ проекта.

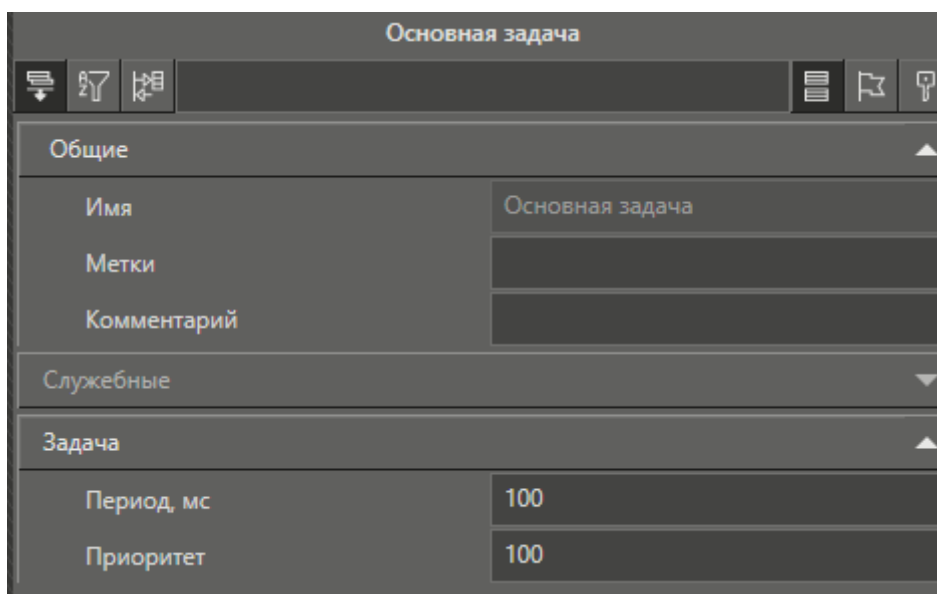


По умолчанию, данная группа содержит один элемент Основная задача. В основную задачу попадают все программы, которые находятся непосредственно в дереве системы в узле, а также объекты, в контекстном меню которых выбран один из пунктов меню:

- Назначить в узел - Название узла;
- Назначить в задачу - Название узла.Основная задача .

Если в проекте имеется только один узел, то назначать объекты в него не требуется. Они, по умолчанию, будут работать в нем. Если в проекте имеется несколько узлов, а объект не назначен ни в одну из задач, то все объекты автоматически будут исполняться в том узле, который является самым верхним в дереве системы.

Основные свойства задачи.



- свойство Период, мс - задает период выполнения задачи в мс. Период выполнения задается независимо для каждой задачи.
- свойство Приоритет - задает приоритет задачи. Если задача с повышенным приоритетом не укладывается в свой цикл вычисления, то в случае одноядерного процессора в устройстве другие задачи исполняться не будут, а в случае наличия многоядерного процессора они будут исполняться реже. В большинстве проектов эту настройку можно оставить без изменения. Использование ее оправдано только когда необходимо гарантировано обеспечить цикл вычисления наиболее важных частей проекта.

Важно! Изменение свойства Приоритет для задач контроллеров допустимо только в том случае, если разработчик проекта убедился, что задача гарантировано успеет уложиться в заданный период и остается время для других задач. В противном случае некорректная настройка приведет к зависанию контроллера. Причем может потребоваться перепрошивка устройства!

Если требуется обеспечить работу какой-либо части проекта независимо от основной задачи, например, с периодом меньшим или большим, то необходимо в группу Задачи добавить еще одну задачу (при помощи контекстного меню либо контекстной панели), после чего выбрать нужный элемент проекта (какой-либо ФБ, Объект и т.п.), и в его контекстном меню выбрать пункт Назначить в задачу-[Название задачи]

6.2.1.9. ОТЧЕТЫ

MasterSCADA 4D поддерживает создание отчетов как на основе текущих данных, так и на основе архивных данных.

В этой группе создаются отчеты, которые в дальнейшем редактируются в дизайнера отчетов.

После создания отчета, дизайнер отчетов открывается автоматически. Для повторного открытия необходимо дважды нажать на созданный элемент, или воспользоваться пунктом контекстного меню Редактировать.

Группа в узле не отображается до тех пор пока не будет добавлен хотя бы один отчет. Добавить отчет в узел в этом случае можно при помощи контекстного меню или контекстной панели.

6.2.1.10. ВНЕШНИЕ КАНАЛЫ

Группа Внешние каналы - это дочерний элемент узла (АРМ, Контроллера, Сервера и др.), который используются для передачи данных в режиме Slave.

Внешние каналы используются для того, чтобы передать значения переменных из устройств, запрограммированных средствами MasterSCADA 4D через протоколы Modbus TCP Slave или Modbus RTU Slave, либо через специализированный M-PLC-OPC-server (возможна передача данных не только через TCP/IP, com-порт, но и через модем).

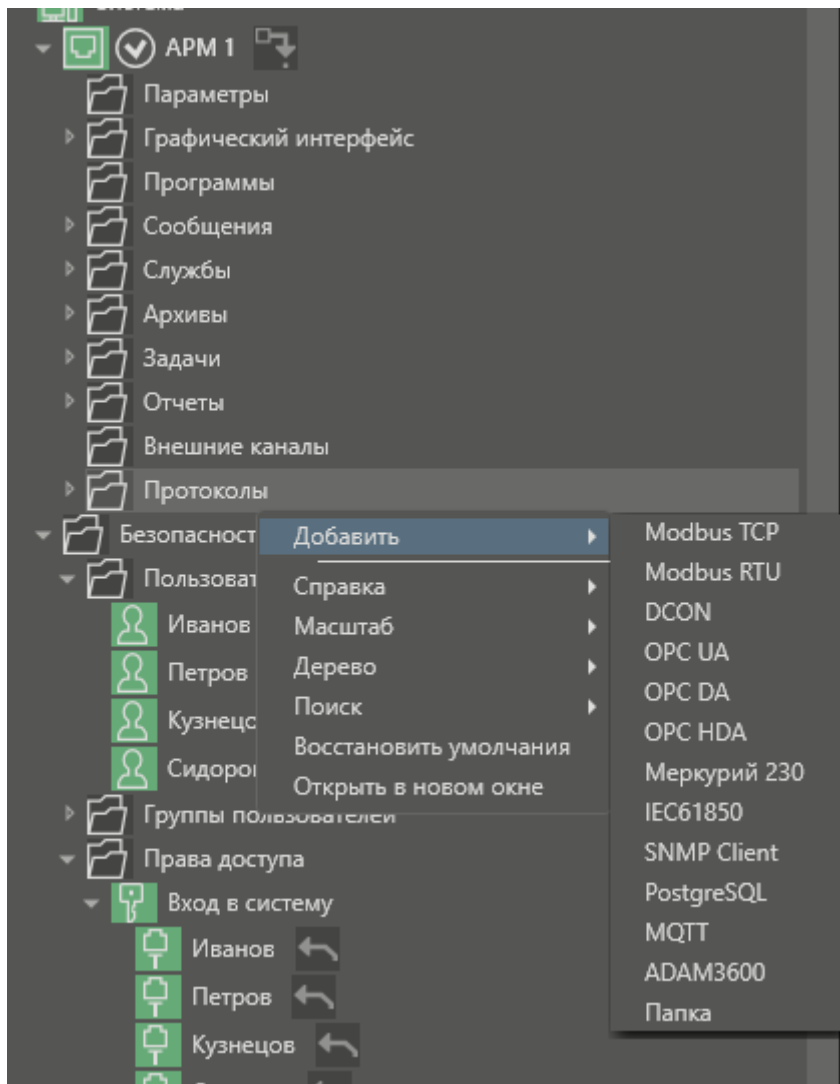
Т.е. в данном случае MasterSCADA 4D выступает в роли Slave.

6.2.1.11. ПРОТОКОЛЫ

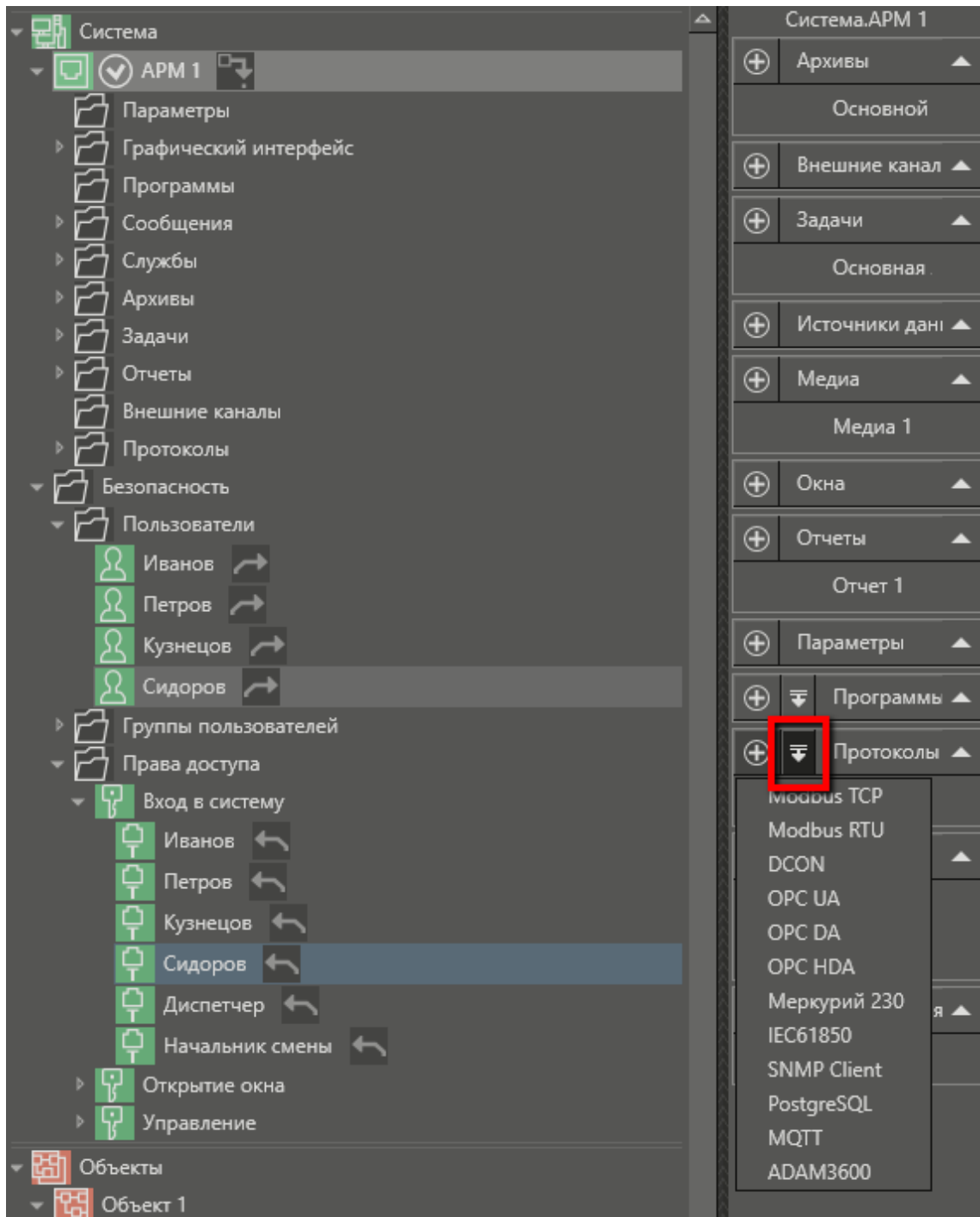
В данную группу добавляют различные протоколы для работы с устройствами (модулями ввода-вывода, счетчиками, контроллерами, запрограммированными не средствами MasterSCADA 4D) и другими приложениями, подключенными к узлу (например, к базе данных, либо к сторонней SCADA-системе). MasterSCADA 4D выступает в роли Master, а подключенное устройство в роли Slave.

Чтобы добавить протокол, можно воспользоваться одним из двух способов:

- Выбрать требуемый протокол в контекстном меню:



- Добавить требуемый протокол, используя контекстную панель:



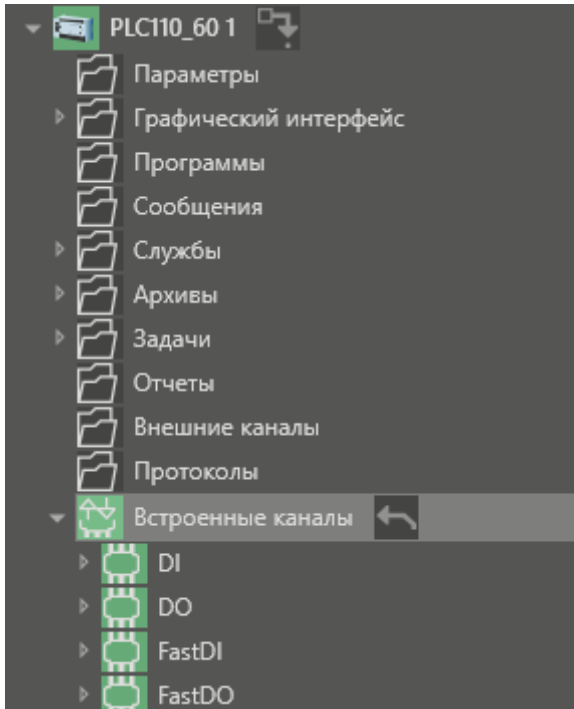
В списке возможных для добавления протоколов будут присутствовать те элементы, которые предназначены для работы с другими устройствами и ПО в режиме Master.

Программы в протоколах

Каждый протокол выполняется в своей задаче. В протокол можно добавить Программу. В этом случае программа будет выполняться в той же задаче, что и протокол. Рекомендуется добавлять задачу в протоколы, необходимо обработать данные сразу же после их получения.

6.2.1.12. ВСТРОЕННЫЕ КАНАЛЫ

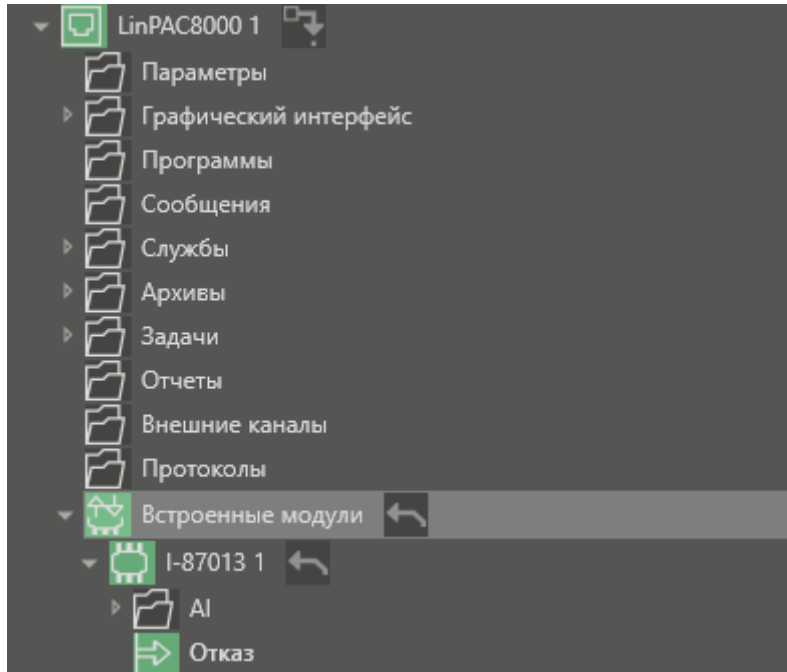
Группа Встроенные каналы присутствует только у определенных узлов.



В этой группе содержатся каналы для получения данных из встроенных физических средств ввода-вывода контроллера, панели оператора.


6.2.1.13. ВСТРОЕННЫЕ МОДУЛИ

Группа Встроенные модули присутствует только у определенных узлов, если в кейте контроллера, для которого создана специальная версия исполнительной системы MasterSCADA 4D, могут размещаться модули ввода-вывода.



В данную группу разработчиком проекта могут добавляться какие-либо модули ввода-вывода, содержащие в себе каналы. Количество каналов и их тип соответствует реальному устройству. Набор модулей ввода вывода может быть также добавлен по умолчанию в момент разработки библиотеки устройства.

6.2.1.14. КАНАЛ

Канал – это сложный элемент дерева Системы, который используется для обмена данными с устройствами. Обозначение в дереве: . Канал является дочерним элементом следующих элементов проекта:

- модуля ввода-вывода, счетчика, другого устройства, добавленного в какой-либо протокол;
- группы Встроенные модули;
- группы Встроенные каналы.

Этот элемент обеспечивает внешние связи узла с модулями ввода-вывода, OPC-серверами, другими контроллерами.

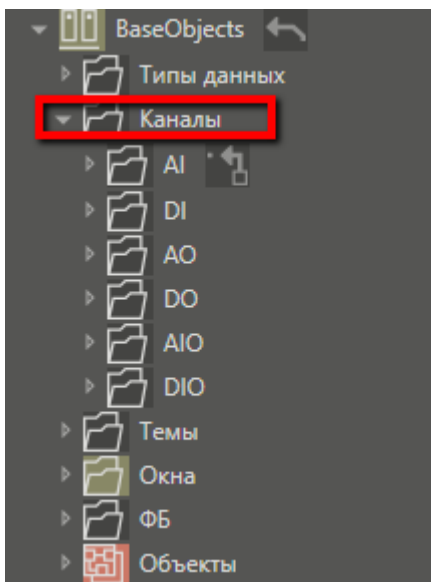
Настройки у каналов, принадлежащих различным протоколам, различны. В зависимости от типа используемого протокола, разработчику проекта будет дана возможность добавить канал с требуемыми настройками.

Канал может содержать в себе основные параметры: Вход, отвечающий за получение (чтение) данных от устройства и Выход, служащий для отправки (записи) данных в устройство.

Настройкой Доступ группы Общие определяется количество параметров канала.

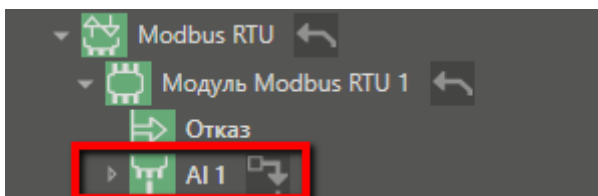
Значение свойства	Параметры канала
Input	Вход
Output	Выход
InOut	Вход, Выход

В библиотеке MasterSCADA 4D представлен ряд каналов predetermined типов, наиболее часто встречающихся в проектах.

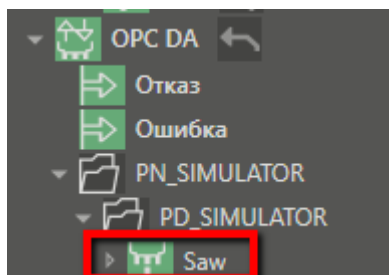


Примеры добавленных каналов в дерево Системы:

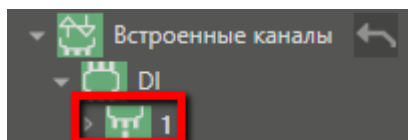
- Канал в модуле Modbus RTU:



- Канал в OPC DA-сервере:



- Канал группы Встроенные каналы:

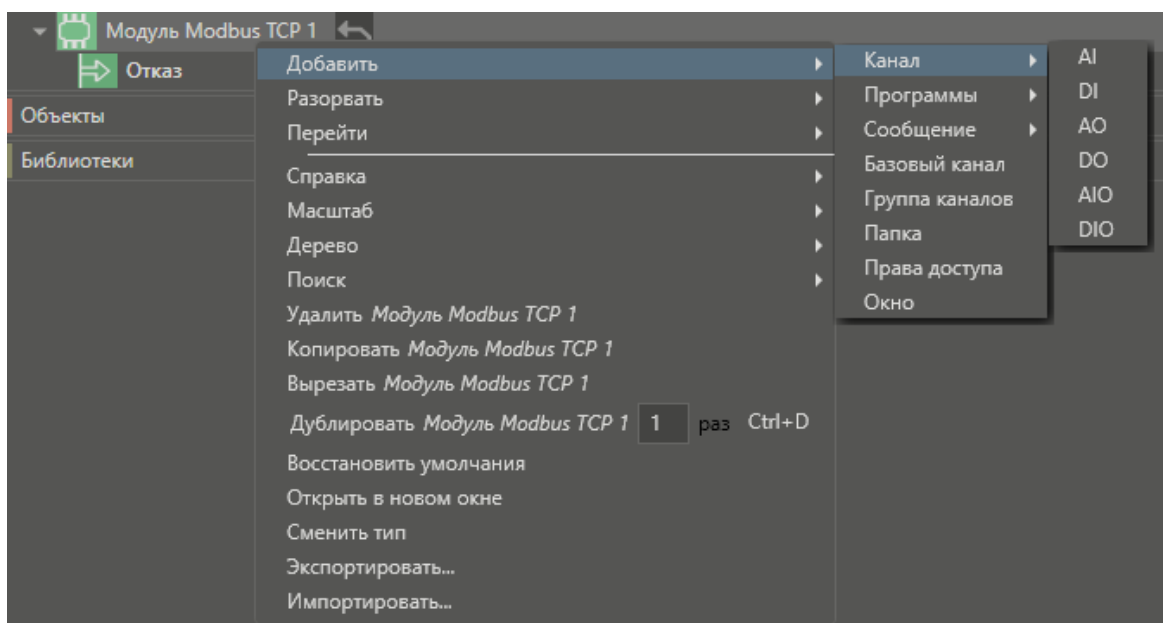


Способы добавления каналов в проект

Ручное добавление

В некоторые модули каналы добавляются вручную. Например, при конфигурировании произвольного Modbus RTU или TCP модуля, каналы добавляются через контекстное меню модуля или контекстную панель. Тип данных определяется также вручную. Количество каналов соответствует количеству переменных, которые необходимо получить из устройства.

При добавлении канала существует возможность выбрать его тип:

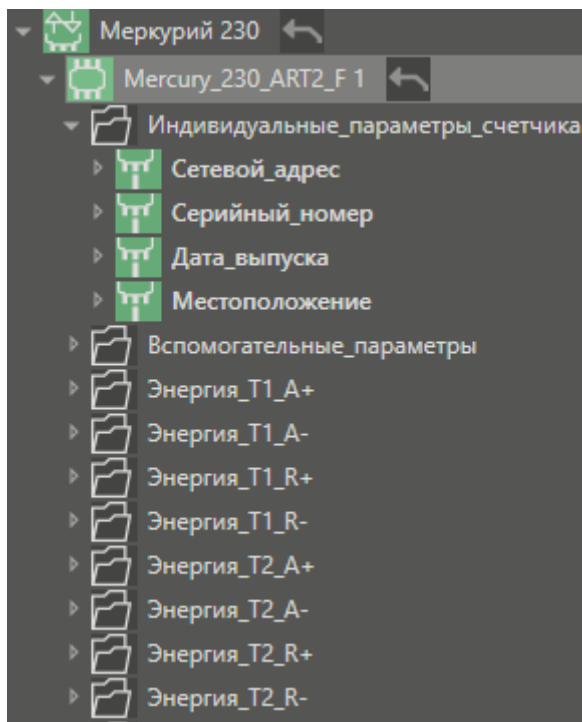


Диалог добавления

В OPC-серверах каналы требуемого типа добавятся после того, как разработчик проекта выберет их во вкладке, в которой осуществляется подключение к OPC-серверам (подробнее описано в разделах, посвященных работе с OPC-серверами).

Автоматическое добавление

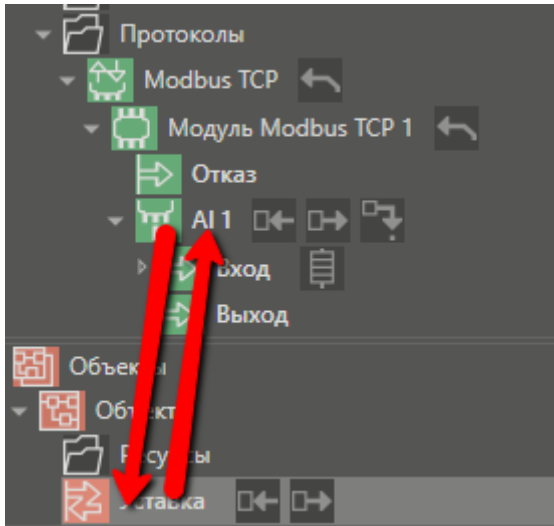
Если в проект добавляется библиотечный модуль ввода-вывода, соответствующий реальному оборудованию, которое поддерживается в MasterSCADA 4D, то каналы добавятся автоматически после добавления модуля в проект :



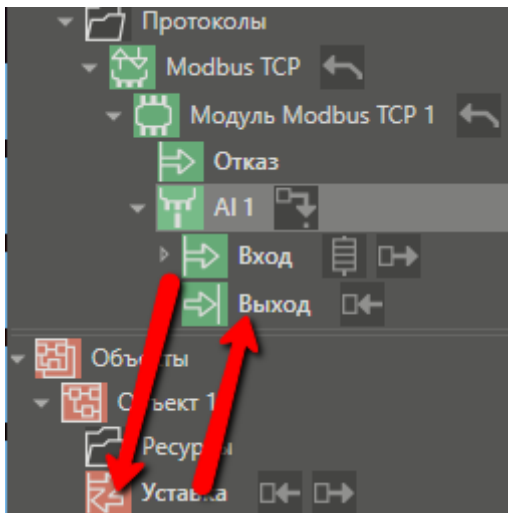
При добавлении predetermined узла, имеющего встроенные физические входы-выходы, каналы также автоматически добавятся в группе Встроенные каналы.

Настройка связи с каналом

Канал, как правило, связывают с параметрами объектов. Параметры объекта можно связывать как с самим каналом, так и с его дочерними переменными. Если свойство Доступ = InOut, то для организации входящей связи, необходимо перетащить параметр на канал, а для организации исходящей связи - перетащить канал на объект (источник данных перетаскиваем на приемник данных).

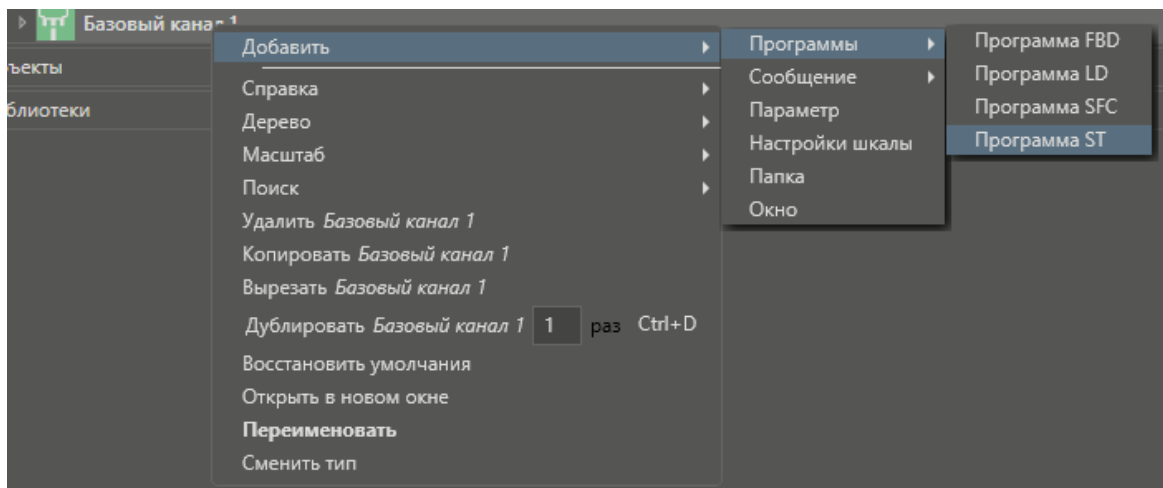


Это будет равнозначно тому, как если бы связь была с параметрами канала:



Дополнительная обработка данных в канале.

Если после получения данных из физического устройства их необходимо обработать, например, выделить биты из байта, то в канале допускается создание программ при помощи контекстного меню:



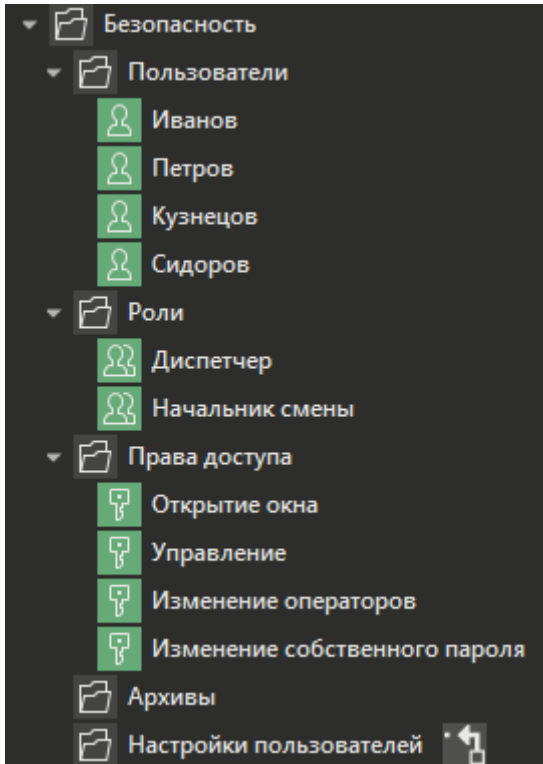
В этом случае у канала появится группа Ресурсы, такая же как и у Объектов и Тегов.

Непосредственно в канал можно также добавить сообщения, окна и другие элементы, характерные скорее для объектов. В этом случае, канал будет отличаться от объекта только возможностью получать данные от устройств и других внешних программ. При использовании таких сложных каналов следует придерживаться тех же принципов, как и других элементов проекта, которые повторяются в проекте неоднократно: если элемент встречается в проекте несколько раз, то необходимо создать в библиотеке его тип, а затем использовать его экземпляры.

6.2.2. БЕЗОПАСНОСТЬ

В настройках данной группы задаются параметры проверки целостности программного обеспечения и проекта, а также в данной группе содержатся элементы для настройки прав доступа операторов (пользователей) в режиме исполнения.

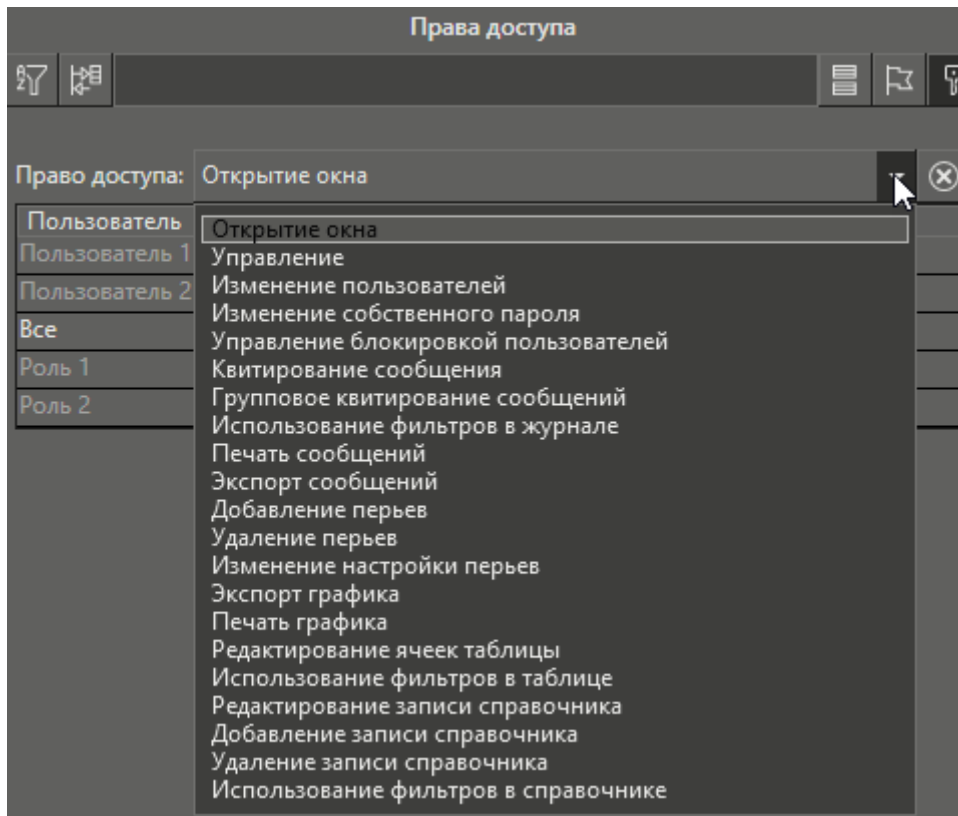
Права доступа могут назначаться как индивидуально, так и для группы лиц. В связи с этим, группа Безопасность содержит в себе следующие основные подгруппы Пользователи, Роли и Права доступа, также вспомогательные Архивы и Настройки пользователей



Если выделить группу Безопасность, перейти в панель Свойств, и переключиться в режим просмотра прав доступа, то можно получить сводную таблицу разрешений по всем пользователям и ролям добавленным в проект

Пользователь	Разрешить	Запретить	Подтверждение	Журнал
Иванов	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Петров	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Кузнецов	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Сидоров	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Диспетчер	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Начальник смены	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Список прав доступа:



Название	Программное имя	Назначение
Общие		
Открытие окна	OpenWindow	Позволяет пользователю открывать окна в режиме исполнения в клиенте визуализации. Если у пользователя нет прав на открытие окна, то появится соответствующее сообщение. Данное право настраивается в дереве системы по умолчанию для всех окон проекта, в дальнейшем умолчания можно изменить в объекте, для всех его дочерних окон, и непосредственно в каждом отдельном окне.
Управление	Control	Позволяет нажимать на кнопки, изменять значения, при помощи различных графических элементов, например, Инкремент, Текстовый ввод и др. Если право на управление имеется, то будут также исполняться назначенные действия. Данное

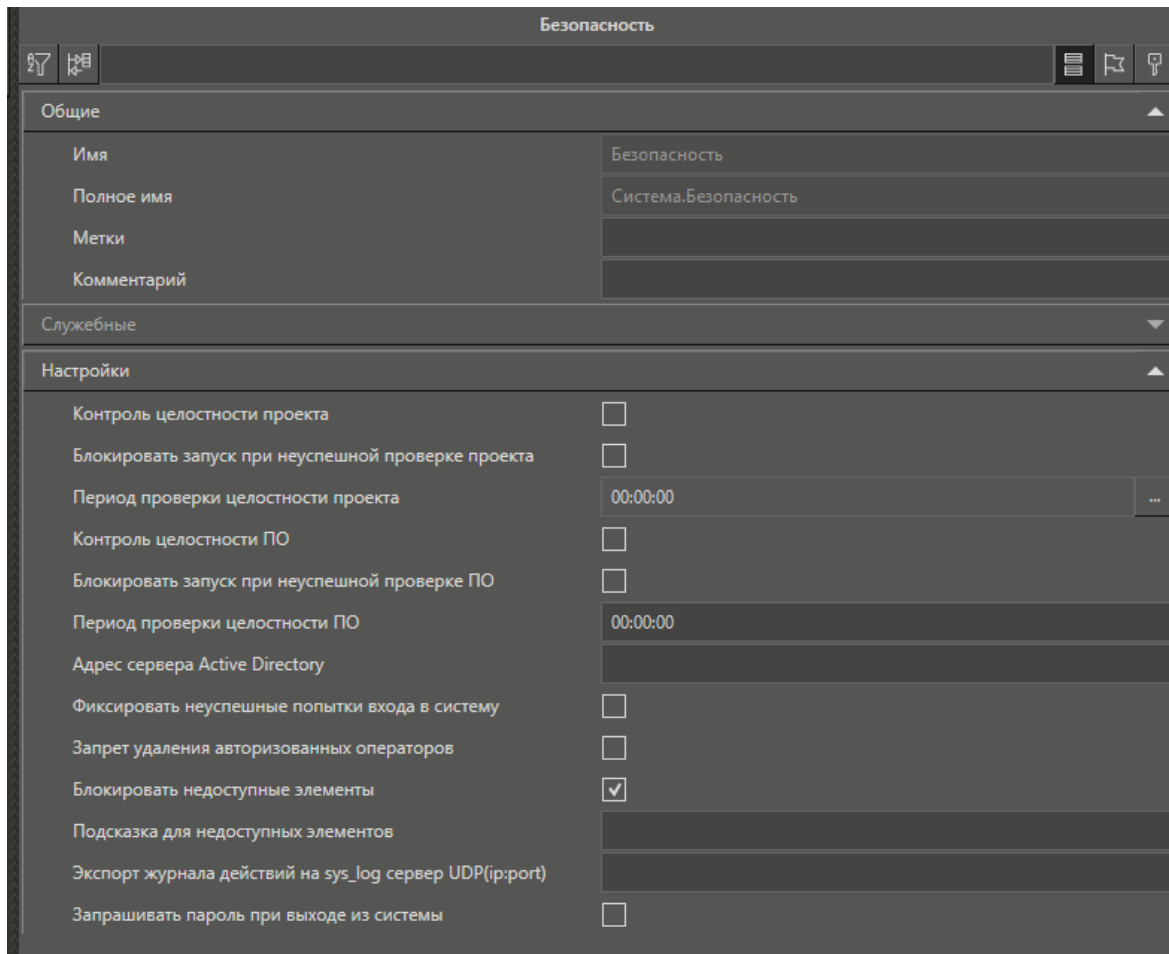
Название	Программное имя	Назначение
		<p>право настраивается в дереве системы по умолчанию для всех элементов проекта, в дальнейшем право можно изменить для конкретного элемента в его панели свойств.</p> <p>Действия захват, покидание и перемещение мыши не учитываются.</p>
Управление пользователями		
Изменение пользователей	ChangeUsers	<p>Позволяет изменять параметры пользователей в режиме исполнения. Если у права стоит флаг Журнал, то в журнал сообщений выдается подробная информация: у какого пользователя что поменяли (если меняется пароль, то текст пароля не пишется, в остальных случаях указывается старое и новое значение настройки/имена измененных ролей). ChangeUsers - программное имя права, для использования его в программах ST. Если установлен флаг Журнал, напротив данного права, то будет сформировано сообщение, содержащее информацию как о новом значении, так и о предыдущем, например, Переименован пользователь с именем Иванов на пользователь Петров. Тип сообщения будет Действие пользователей</p>
Изменение собственного пароля	ChangeSelfPassword	<p>Позволяет изменить собственный пароль пользователя, который авторизован в текущий момент в клиенте визуализации. Если у права стоит флаг Журнал, то в журнал сообщений выдается подробная информация: у какого пользователя что поменяли (если меняется пароль, то текст пароля не пишется, в остальных случаях указывается старое и новое значение настройки/имена измененных ролей). ChangeSelfPassword - программное</p>

Название	Программное имя	Назначение
		имя права, для использования его в программах ST
Управление блокировкой пользователей	BlockUsers	Позволяет заблокировать или снять блокировку пользователя. Блокировка учетной записи пользователя, это состояние при котором пользователь не может запустить новый сеанс работы в клиенте визуализации, но сведения об этом пользователе не удаляются из исполнительной системы (BlockUsers).
Права доступа элементов		
Квитирование сообщений, Групповое квитирование сообщений, Использование фильтров в журнале, Печать сообщений, Экспорт сообщений	JournalAck, JournalGroupAck, JournalUseFilters, JournalPrint, JournalSave	Данные права доступа относятся к элементам Журнал и Архивный журнал. В дереве системы настраиваются умолчания для всех элементов данного типа, которые будут использоваться в проекте. Для каждого отдельного элемента права доступа можно будет изменить, а также изменить умолчания у родительского элемента, например, изменение данного права у объектов, приведет к изменений умолчаний для всех элементов данного типа, которые находятся в окнах этого объекта. Подробное описание прав доступа смотрите в разделе Журнал. Права доступа
Изменение состава перьев, Изменение настройки перьев, Экспорт графика, Печать графика	TrendAddPens, TrendChangePens, TrendSave, TrendPrint	Данные права доступа относятся к элементам Тренд, Легенда, График XY. В дереве системы настраиваются умолчания для всех элементов данного типа, которые будут использоваться в проекте. Для каждого отдельного элемента права доступа можно будет изменить, а также изменить умолчания у родительского элемента, например, изменение данного права у объектов, приведет к изменений умолчаний для всех элементов

Название	Программное имя	Назначение
		данного типа, которые находятся в окнах этого объекта.. Подробное описание прав доступа смотрите в разделах: Тренд. Права доступа, График XY. Права доступа, Легенда. Права доступа
Редактирование ячеек таблицы, Использование фильтров в таблице	DataTableEdit, DataTableFilter,	Данные права доступа относятся к элементу Таблица данных. В дереве системы настраиваются умолчания для всех элементов данного типа, которые будут использоваться в проекте. Для каждого отдельного элемента права доступа можно будет изменить, а также изменить умолчания у родительского элемента, например, изменение данного права у объектов, приведет к изменений умолчаний для всех элементов данного типа, которые находятся в окнах этого объекта. Подробное описание прав доступа смотрите в разделе: Таблица данных. Права доступа
Редактирование записи справочника, Добавление записи справочника, Удаление записи справочника, Использование фильтров в справочнике	DirectoryEdit, DirectoryAdd, DirectoryDelete, DirectoryFilter	Данные права доступа относятся к элементу Справочник. В дереве системы настраиваются умолчания для всех элементов данного типа, которые будут использоваться в проекте. Для каждого отдельного элемента права доступа можно будет изменить, а также изменить умолчания у родительского элемента, например, изменение данного права у объектов, приведет к изменений умолчаний для всех элементов данного типа, которые находятся в окнах этого объекта. Подробное описание прав доступа смотрите в разделе: Справочник. Права доступа

6.2.2.1. ПАНЕЛЬ СВОЙСТВ БЕЗОПАСНОСТЬ

Панель свойств элемента Безопасность имеет вид:



Название	Описание
Контроль целостности проекта	Активирует контроль целостности проекта. Если флаг установлен, то в конфигурацию проекта включается зашифрованный файл <i>cfg_files.dat</i> (шифрование AES 256 фиксированным ключем) со списком MD5 контрольных сумм всех загружаемых из среды разработки файлов конфигурации узла (включая папки <i>cfg</i> и <i>htdocs</i>)
Блокировать запуск при неуспешной проверке проекта	Определяет порядок работы при неуспешной проверке проекта. Если флаг установлен, и при старте режима исполнения найдено отличие контрольных сумм файлов от прописанных в <i>cfg_files.dat</i> , то возникает ошибка запуска узла. Текст ошибки со списком измененных файлов выдается при подключении к узлу клиента визуализации. Если флаг снят, то узел запускается, при этом текст ошибки со списком измененных

Название	Описание
	файлов выдается как системное сообщение в журнале.
Период проверки целостности проекта	<p>Определяет период, с которым будет происходить дополнительный контроль целостности после старта режима исполнения. Если установлено значение отличное от 0, то с этим периодом будет происходить дополнительный контроль целостности. Если при этом стоит настройка Блокировать запуск при неуспешной проверке проекта, то при обнаружении несоответствия происходит перезапуск режима исполнения (перед этим сформируется сообщение в журнале). При несоответствиях сообщение в журнал пишется по факту каждой проверки.</p>
Контроль целостности ПО	<p>Включает контроль целостности программных файлов режима исполнения (включая <i>mplc</i>, <i>*.so</i>, папки <i>nginx</i>, <i>nodejs</i>). Создается два файла со списком контрольных сумм:</p> <p>Для контроля папок <i>nginx</i>, <i>nodejs</i>. Они не обновляются из среды разработки. Этот файл создается при установке исполнительной системы.</p> <p>Для контроля <i>*.so</i> файлов, которые могут обновляться из среды разработки. Этот файл создается при сборке вместе с <i>*.so</i> файлами и загружается вместе с ними.</p>
Блокировать запуск при неуспешной проверке ПО	Определяет порядок работы при неуспешной проверке ПО. Если флаг установлен, то ПО не запустится
Период проверки целостности ПО	Определяет период, с которым будет происходить дополнительный контроль целостности после старта режима исполнения.
Адрес сервера Active Directory	Указывается имя домена (<i>host</i>) или IP-адрес сервера, где настроен Active Directory. Если свойство задано, то разрешается при авторизации в

Название	Описание
	<p>клиенте визуализации задавать не только имена (логины) и пароли пользователей, настроенные непосредственно в MasterSCADA 4D, но и логины и пароли, определенные для пользователей в операционной системе, в AD. В этом случае в окне авторизации отображается текстовое поле ввода имени пользователя без predefined списка.</p> <p>При вводе данных пользователя, если логин совпадает с именем пользователя, добавленным в MaterSCADA 4D, то идет проверка пароля, заданного в MaterSCADA 4D.</p> <p>Если в MaterSCADA 4D такого пользователя нет, то введенные логин/пароль проверяются в указанном сервере AD. Если логин проходит, получается список групп пользователя Active Directory, в исполнительной системе ищутся одноименные роли. Если в исполнительной системе нет ни одной роли, то авторизация запрещается.</p> <p>Если логин и пароль введенные в окне авторизации совпали с данными полученными из AD, и в проекте имеется роль или несколько ролей, соответствующих группам AD, то создается сессия пользователя с правами данных ролей.</p>
<p>Фиксировать неуспешные попытки входа в систему</p>	<p>Определяет необходимость формировать сообщение, если при авторизации пользователя произошли ошибки, например, истек срок действия пароля, был введен неправильный пароль и т.п. Если флаг установлен, то сообщение будет выдаваться</p> <p>Важно! Данная настройка формирует сообщения только при ошибках авторизации пользователей созданных в режиме исполнения.</p>
<p>Запрет удаления авторизованных операторов</p>	<p>Определяет возможность удалять авторизованных пользователей. Если флаг установлен, то пользователя нельзя удалять в режиме исполнения</p>

Название	Описание
	Важно! Данная настройка влияет только на пользователей созданных в режиме исполнения.
Блокировать недоступные элементы	
Подсказка для недоступных элементов	
Экспорт журнала действий на sys_log сервер UDP(ip:port)	
Запрашивать пароль при выходе из системы	<p>Флаг позволяет организовать завершение сессии пользователя по паролю. У действия Завершить сессию нужно использовать параметр Пароль. Завершение сессии произойдет только в том случае, если значение параметра действия будет совпадать с паролем текущего пользователя.</p> <p>Если есть активная сессия пользователя, то залогиниться может только, если совершается логин для роли с правом Управление пользователями. При этом сессия предыдущего пользователя завершится и в журнале появится сообщение Принудительный выход подтвержден <новый_пользователь></p>

Сообщения, возникающие в результате проверки целостности программного обеспечения и проверки целостности проекта, будут иметь тип Системное сообщение

В начале проверки сформируется сообщение Запущена проверка целостности ПО/проекта (приоритет сообщения будет 100), в результате удачного завершения появится сообщение Завершена проверка целостности ПО/проекта. Если проверка будет неудачной, то текст сообщения будет включать в себя код ошибки и список файлов, которые проверку не прошли. Примеры сообщений:

- Ошибка при проверке целостности ПО: отличий в файлах - 3 ...
- Ошибка при проверке целостности проекта: отличий в файлах - 1 4 : cfg/VMInfo.json

Вначале идет список файлов с отличиями, затем идут пары <код ошибки> <имя файла>.

Коды ошибок:

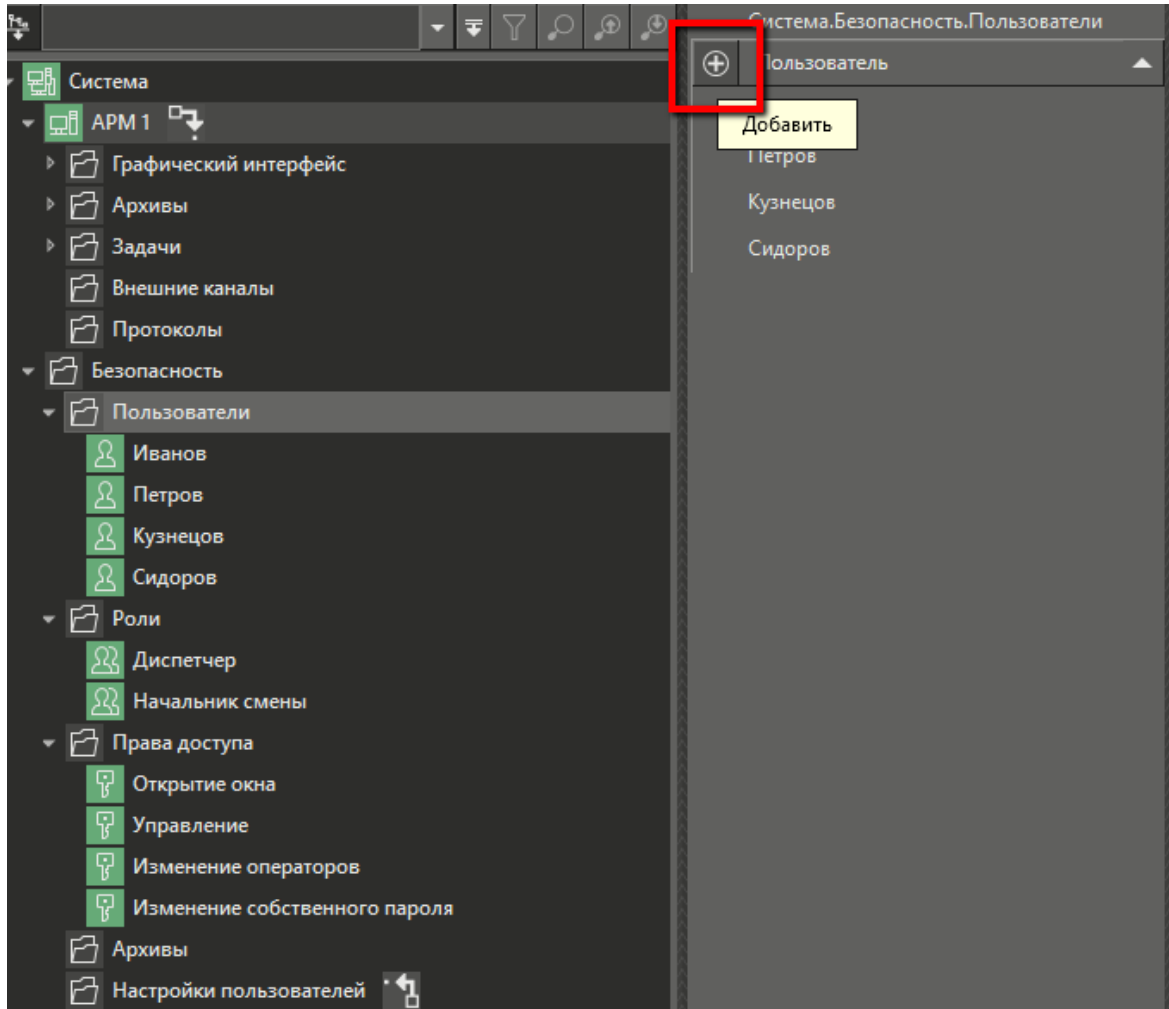
- 1 - Файл отсутствует
- 2 - Ошибка чтения файла
- 3 - Ошибка в пути к файлу
- 4 - Несоответствие MD5 сумм
- 5 - Ошибка дешифрации файла

Контроль целостности при использовании резервирования

Если включено резервирование, то создаются 2 файла - `cfg_files.dat` с checksumмами для основного сервера и `cfg_files_r.dat` - для резервного.

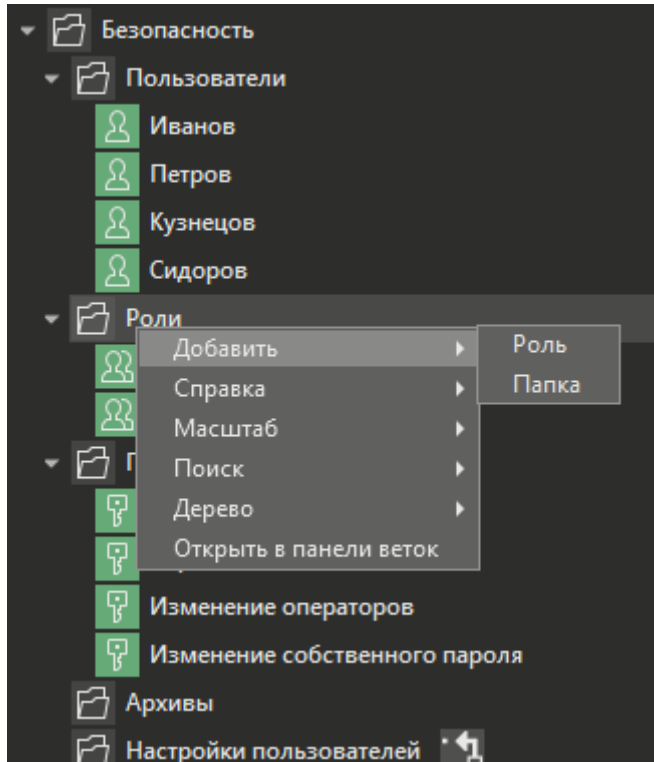
6.2.2.2. ПОЛЬЗОВАТЕЛИ

Данная группа предназначена для создания/удаления пользователей проекта, которые будут допущены к управлению и просмотру в режиме исполнения. Для того, чтобы добавить пользователя, необходимо выполнить пункт контекстного меню группы Добавить.Пользователь, либо воспользоваться контекстной панелью.

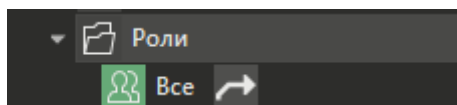


6.2.2.3. РОЛИ

Данная группа предназначена для создания/удаления ролей пользователей проекта. Как правило, роли пользователей используются для того, чтобы разделить пользователей проекта, работающих в режиме исполнения, по категориям, например: диспетчеры, начальники смен, инженеры, ученики и пр. Для того, чтобы добавить роль, необходимо выполнить пункт контекстного меню группы **Добавить.Роль** либо воспользоваться контекстной панелью.



По умолчанию каждый проект содержит роль Все.



В роль Все нельзя вручную добавлять пользователей. Считается, что все пользователи неявно добавлены в эту роль, но ФБ UsersGetGroups не выдает эту роль.

В новом проекте для этой роли разрешены все права, кроме:

- Изменение пользователей
- Управление блокировкой пользователей

Для проектов, сохраненных в версиях 1.2.9 и более ранних, при конвертации в версию 1.2.10 и более позднюю добавляется роль Все, у которой разрешены все права, кроме:

- Открытие окна
- Управление
- Изменение собственного пароля
- Изменение пользователей
- Управление блокировкой пользователей

Панель свойств Роли имеет вид:

Роль 1

Общие

Имя	Роль 1
Полное имя	Система.Безопасность.Роли.Роль 1
Метки	
Комментарий	

Служебные

Авторизация

Длительность сессии	00:00:00
Минимальная длина пароля	0
Количество неповторяемых паролей	0
Использовать сложный пароль	<input type="checkbox"/>
Срок действия пароля	00:00:00
Количество последовательных неуспешных попыток входа	0
Продолжительность блокировки при превышении количества	00:00:00
Минимальный срок действия пароля	00:00:00
Запрет одновременного входа одного пользователя	<input type="checkbox"/>

Безопасность

Блокировать доступ к операционной системе	<input type="checkbox"/>
Фиксировать попытки выполнения действий	<input type="checkbox"/>

Название	Назначение
Категория Общие	Как правило, настраивается свойство Имя. Далее по имени роли можно определить ее полномочия, назначить ей права. Подробное описание данной категории приведено в соответствующем разделе.
Категория Авторизация	
Длительность сессии	Задается максимально возможная длительность сессии пользователя. Если время длительности сессии достигает указанного значения, то сессия пользователя будет завершена.
Минимальная длина пароля	Определяется минимальное количество символов при настройке пароля в режиме исполнения.

Название	Назначение
Количество неповторяемых паролей	<p>Определяет, сколько раз пользователь должен задать неповторяющийся пароль. Если указано значение 0, то ограничения отсутствуют. Пример: Если указано 3, то пользователь должен трижды сменить текущий пароль на новый, отличный от текущего и двух предыдущих, а в дальнейшем (четвертый пароль) может задать пароль, совпадающий с первым.</p>
Использовать сложный пароль	<p>Определяет состав символов при настройке пароля в режиме исполнения. Если флаг установлен, то пароль должен содержать хотя бы одну цифру, одну прописную и одну строчную букву.</p>
Срок действия пароля	<p>Определяет срок действия пароля пользователя. Если время действия текущего пароля достигло указанного, то при начале очередной сессии пользователя появится сообщение и пользователь не будет допущен к работе, до тех пор пока пароль не будет изменен в режиме исполнения</p>
Количество последовательных неуспешных попыток входа	<p>Определяет сколько раз пользователь может попытаться не успешно начать сессию в клиенте визуализации. После успешного запуска сеанса работы счетчик количества неуспешных попыток должен сбрасываться. Если установлено значение 0, то ограничение отсутствует. Данная настройка применяется только к пользователям, созданным в режиме исполнения</p>
Продолжительность блокировки при превышении количества неуспешных попыток входа	<p>Определяет интервал времени в течении которого пользователь, который не смог авторизоваться за количество попыток, разрешенное в предыдущем пункте, не сможет повторить попытки авторизоваться в клиенте визуализации. Данная настройка применяется только к пользователям, созданным в режиме исполнения</p>

Название	Назначение
Минимальный срок действия пароля	<p>Определяет минимальный срок существования пароля. Задается в мс. Если с предыдущей смены пароля прошло времени меньше, чем указано в данном свойстве, то при попытке его смены в режиме исполнения функциональный блок (UsersChangeOwnPassword, UsersSetPassword), выполняющий смену пароля выдаст ошибку.</p>
Запрет одновременного входа одного пользователя	<p>Определяет возможность пользователя авторизоваться с двух разных клиентов визуализации одновременно. Если флаг установлен, то возможны следующие варианты:</p> <ul style="list-style-type: none"> • В случае, если есть активная сессия для этого пользователя с того же самого адреса, то закрывается прошлая сессия, создается новая • Если текущая сессия запущена на другом устройстве, с другим адресом, то в случае, если она неактивна больше 10 секунд, то она закрывается и создается новая. Если от клиента визуализации продолжают приходить запросы к исполнительной системе, то возвращается ошибка авторизации.
Категория Безопасность	
Блокировать доступ к операционной системе	<p>Если флаг установлен, то пользователи, добавленную в эту роль, не смогут воспользоваться стандартными горячими клавишам Windows, например, CTRL+ ALT+DEL.</p> <p>Важно! Чтобы использовать данную функцию клиента визуализации необходимо перед первым его использованием запустить файл c:\Program Files\MPSSoft\MasterSCADA 4D RT 1.2\Redistributable\Interception\install_ЗапускатьОтИмениАдминистратора.bat</p>

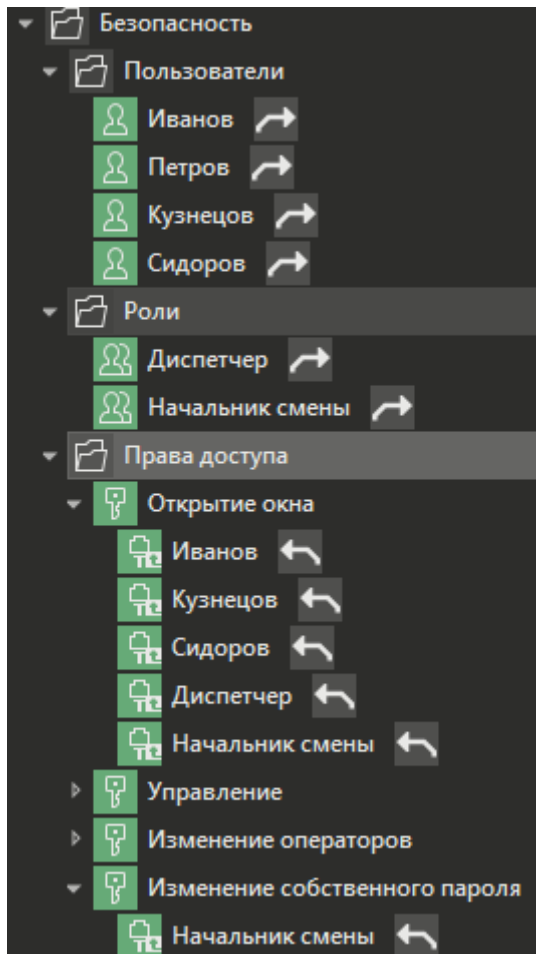
Название	Назначение
Фиксировать попытки выполнения действий	<p>Определяет необходимость фиксирования попыток выполнения действий у пользователей назначенных на эту роль. Если флаг установлен и для действия установлены флаги Подтверждение и Журнал, то запись в журнал будет производится не только, если пользователь выполнит указанное действие, но и попытается его выполнить.</p> <p>Например, пользователь ввел значение, в соответствии с настройками, появилось окно подтверждения, и пользователь нажал Отмена, в этом случае сообщение в журнале также будет сформировано.</p>

Важно! Если пользователь принадлежит нескольким ролям с разными настройками информационной безопасности, то к этому пользователю будут применяться более строгие ограничения из заданных. Например, если установлено разное значение в свойстве Минимальный срок действия пароля, то будет применяться наименьшее значение. Если указано разное ограничение в свойстве Минимальная длина пароля, то будет использоваться максимальное значение среди всех ролей, в которые назначен пользователь.

6.2.2.4. ПРАВА ДОСТУПА

Данная группа содержит подгруппы-глобальных прав в виде дерева.

Эта группа формируется автоматически. При проставлении флагов на панели свойств Безопасность, в данную группу добавляются ссылки на тех или иных пользователей проекта.



6.2.2.5. АРХИВЫ

В группу могут быть добавлены два элемента:

Сетевая БД настроек - предназначен для настройки внешней базы данных, в которой может храниться список пользователей, добавленных в режиме исполнения, а также их настройки (пароли, роли и др.)

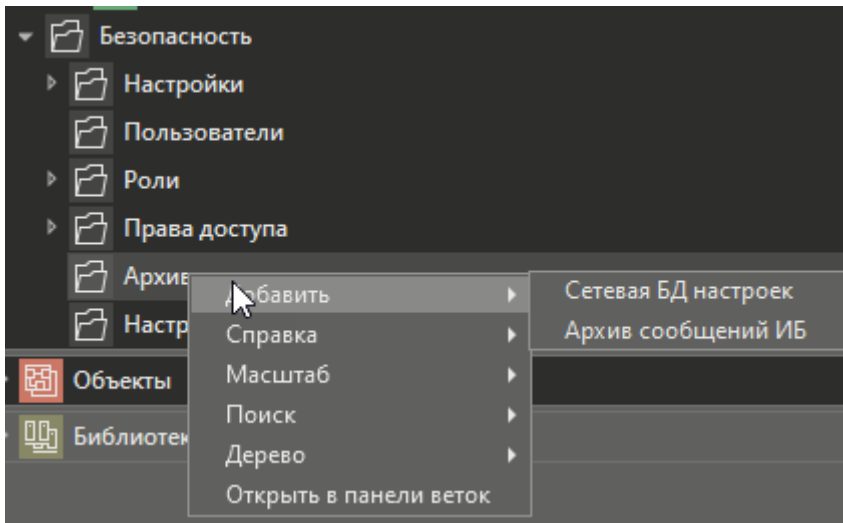
Архив сообщений ИБ - служит для формирования отдельной базы в данных, в которой будут сохраняться автоматически сообщения имеющие тип Сообщения ИБ

6.2.2.5.1. СЕТЕВАЯ БД НАСТРОЕК

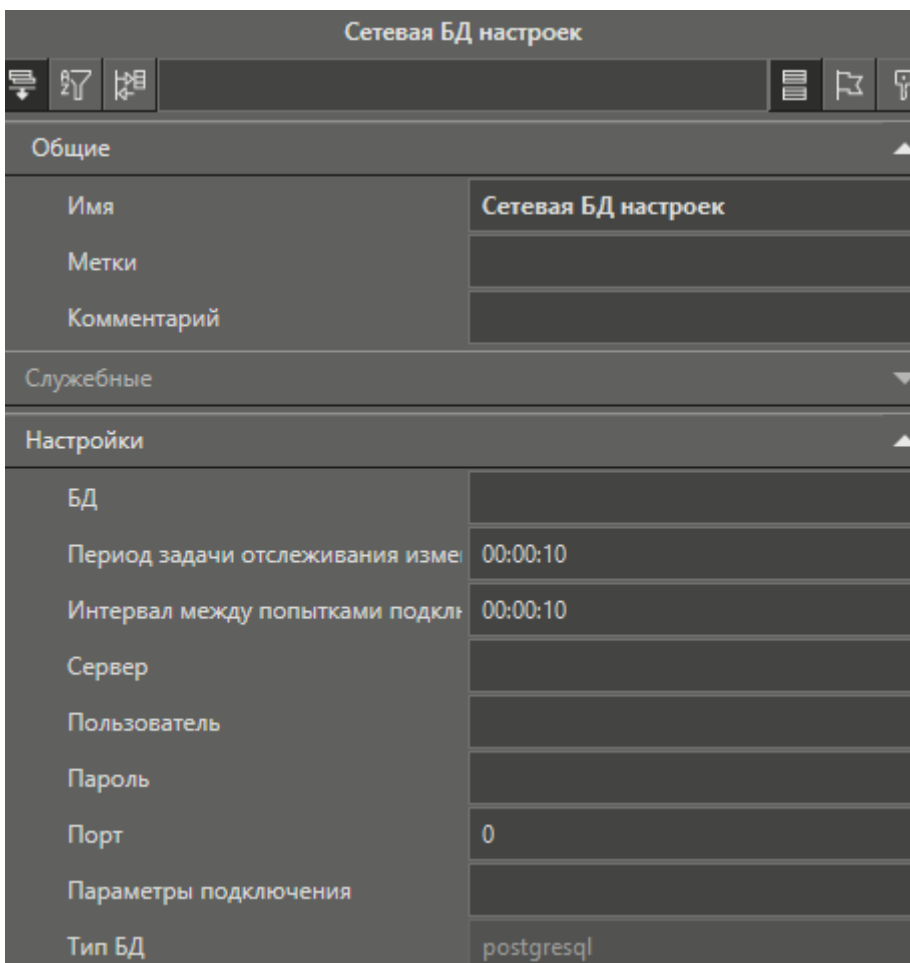
Группа предназначена для настройки внешней базы данных, в которой может храниться список пользователей, добавленных в режиме исполнения, а также их настройки (пароли, роли и др.)

Важно! Для работы этого элемента в режиме исполнения необходима поддержка опции MSRT DB, которая позволяет хранить архивы во внешней базе данных. В базовую версию MasterSCADA 4D RT эта опция не входит.

Для добавления элемента в группу Архивы необходимо в контекстном меню выбрать пункт Добавить.Сетевая БД настроек:



Панель свойств элемента Сетевая БД настроек:



Свойство	Рекомендация
БД	Имя ранее созданной базы данных PostgreSQL.
Период задачи отслеживания изменений	Период с которым внешняя база данных будет синхронизироваться с локальным файлом JSON.
Интервал между попытками подключения	Интервал, с которым осуществляется повторное подключение в случае неудачного подключения к БД
Сервер	Указывается IP-адрес компьютера, на котором установлена база данных.
Пользователь	Имя пользователя, который добавлен в настройках самой базы данных. Соответствует LoginRole на сервер PostgreSQL.
Пароль	Пароль пользователя, который добавлен в настройках самой базы данных.
Порт	Порт TCP IP, который используется для передачи данных между средой исполнения и базой данной, указывается тот порт который задан в самой БД
Параметры подключения	В текущей версии не используется
Тип БД	В текущей версии доступна только БД PostgreSQL.

Если внешняя база данных не добавлена, то список пользователей хранится в локальном файле JSON, у каждого узла отдельно.

Если добавлена, то локальный файл также используется, но при старте и во время работы синхронизируется с внешней базой данных.

Как в локальном файле, так и во внешней базе данных хранится метка времени, которая генерируется при записи новых данных о пользователях.

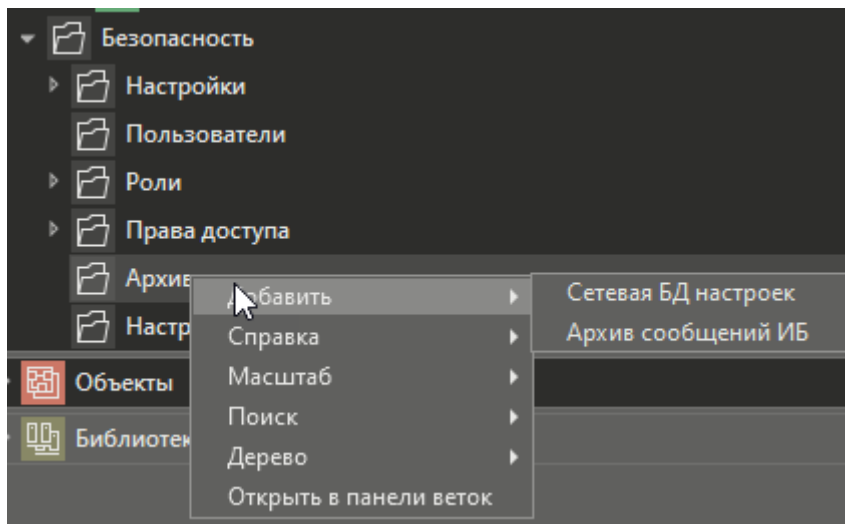
Настройки подключения к внешней базе данных передаются в каждый узел системы. При старте система пытается подключиться к ней, если подключение установлено и метки времени отличаются, то данные из нее передаются в локальный файл. Далее система периодически проверяет метку времени во внешней базе данных - если обнаруживается изменение, данные передаются в локальный файл. При изменении пользователей в системе

(например, используя функциональные блоки) новые данные сохраняются как в локальную, так и внешнюю БД (при этом формируется новая метка времени). Если внешняя БД в этот момент недоступна, то вызываемый функциональный блок сформирует ошибку и не сработает.

6.2.2.5.2. АРХИВ СООБЩЕНИЙ ИБ

Группа предназначена для формирования отдельной базы в данных, в которой будут сохраняться автоматически сообщения имеющие тип Сообщение ИБ со всех узлов проекта.

Для добавления элемента в группу Архивы необходимо в контекстном меню выбрать пункт Добавить.Архив Сообщений ИБ:



Панель свойств элемента Архив сообщений ИБ:

Архив сообщений ИБ	
Общие	
Имя	Архив сообщений ИБ
Полное имя	Система.Безопасность.Архивы.Архив сообщений ИБ
Метки	
Комментарий	
Служебные	
Настройки	
БД	security_events.db
Максимальное время хранения	365.00:00:00
Максимальный объем архива (Mb)	0
Число удаляемых элементов в одной операции	1000
Период чистки архива	01:00:00
Период задачи записи архива	00:00:00
Интервал между попытками подключения	00:01:00
Максимальное количество одновременных подключе	0
Максимальный размер очереди записи	10000
Период подсчета статистики	01:00:00
Синхронизация БД при резервировании	<input checked="" type="checkbox"/>
Сервер	
Пользователь	
Пароль	
Порт	0
Параметры подключения	
Тип БД	sqlite3
Разрешения	
Доступ через OPCUA	По умолчанию

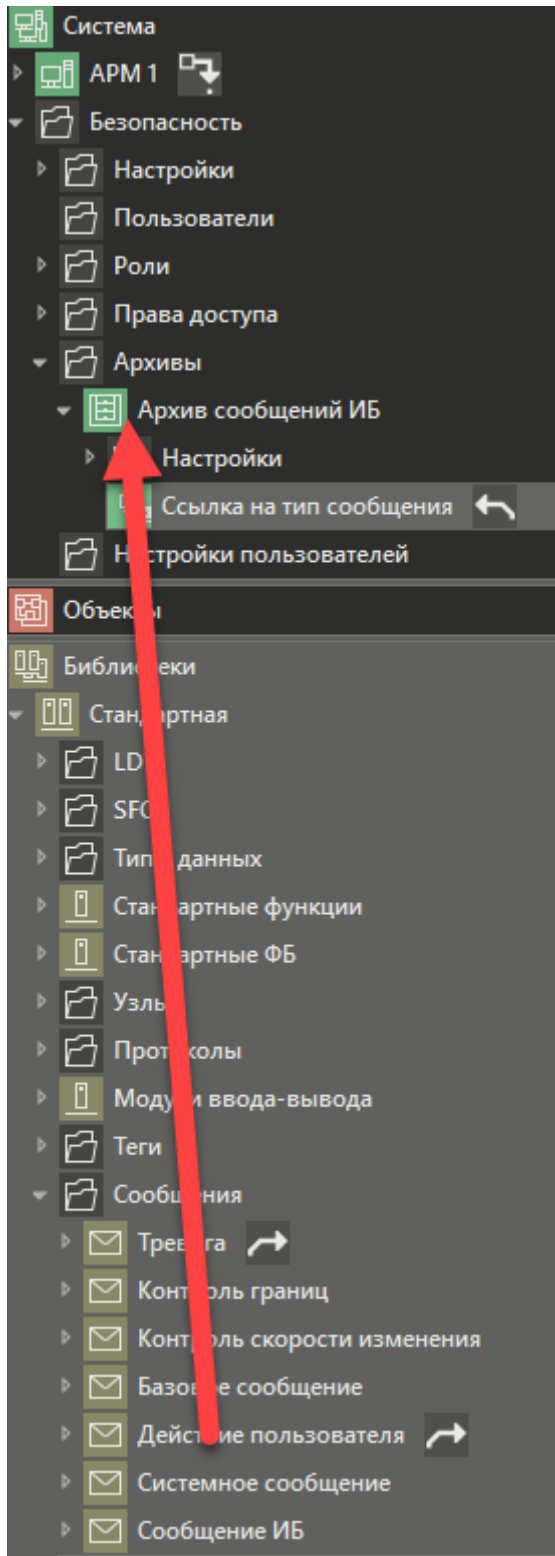
Описание настроек смотрите в разделе Основной архив сообщений

По умолчанию имя БД, в которой предполагается хранить настройки безопасности будет иметь имя security_events.db.

Если внешняя база данных не добавлена, то сообщения типа Сообщение ИБ будут попадать в общий архив сообщений.

Сохранение других типов сообщений

Если необходимо в Архив сообщений ИБ сохранять также и другие типы сообщений, например, Действия пользователей, то необходимо перетащить эти типы сообщений на данный элемент. В результате появится ссылка на этот тип в элементе:



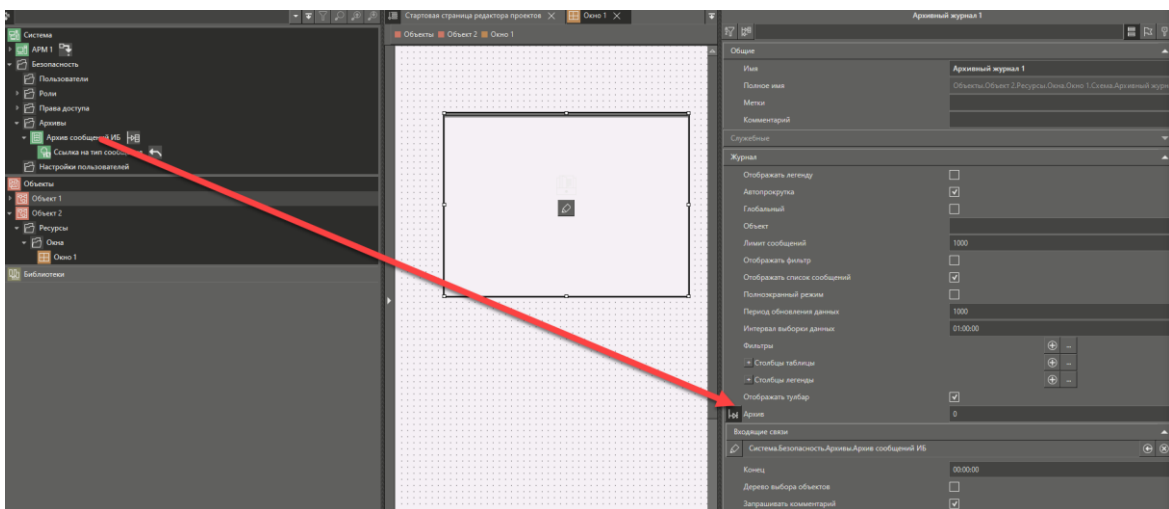
В этом случае сообщения данного типа будут сохраняться и в общем архиве, и в архиве сообщений ИБ

Отображение сообщений информационной безопасности в журнале.

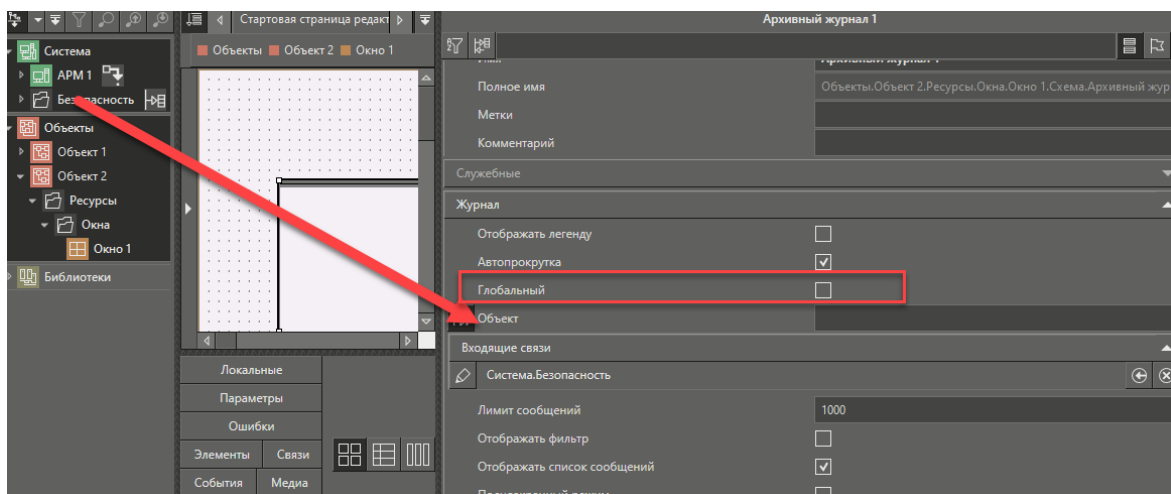
Чтобы создать журнал сообщений информационной безопасности, необходимо добавить в окно элемент Архивный журнал. Далее существует два варианта настройки:

- когда в журнал попадают все типы сообщений, которые хранятся в Архиве сообщений ИБ
- когда в журнал попадают только сообщения, которые имеют тип Сообщение ИБ

Если перетащить элемент Архив сообщений ИБ в свойство журнала Архив, то в журнале будут отображаться сообщения типа Сообщение ИБ, а также все другие типы сообщений, которые находятся в этой базе данных:



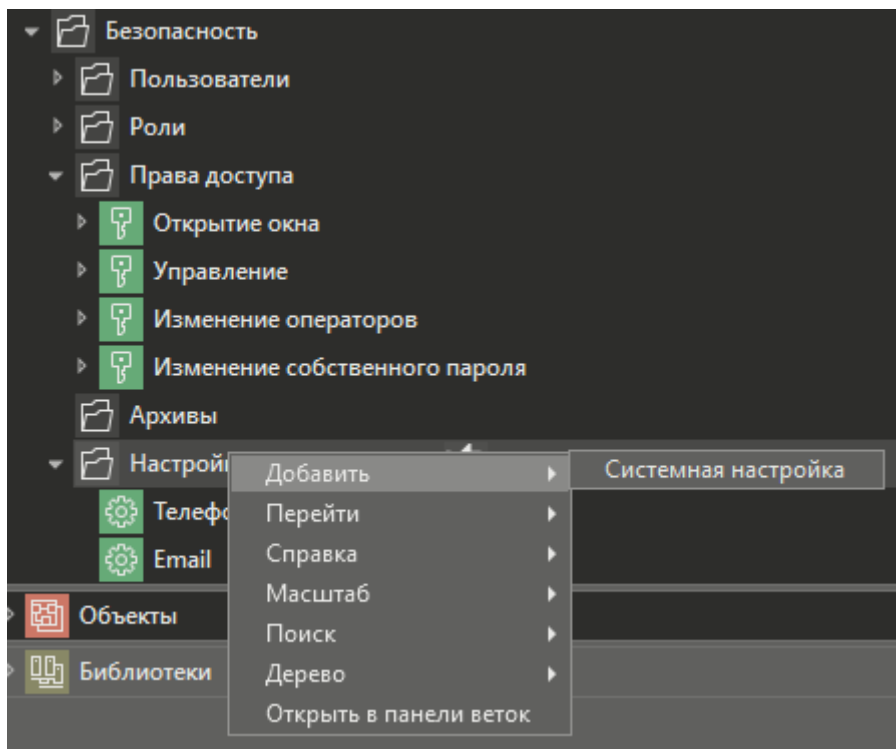
Если перетащить в свойство журнала Объект элемент дерева системы Безопасность и не устанавливать флаг Глобальный, то в этом случае журнале будут отображаться только сообщения, которые будут иметь тип Сообщение ИБ:



6.2.2.6. НАСТРОЙКИ ПОЛЬЗОВАТЕЛЕЙ

Группа содержит список настроек, которые можно дополнительно задавать у пользователей.

Для добавления нового элемента необходимо выполнить пункт контекстного меню Добавить. Системная настройка:



В окне указать тип и имя новой настройки:

Редактирование

Имя Тип **DATE**

Начальное значение Сохранять

Простые типы Библиотечные типы данных Массивы

Логические	Целые знаковые	Целые беззнаковые
BOOL	DINT	UDINT
BYTE	INT	UINT
	LINT	ULINT
	SINT	USINT
Вещественные	Дата-время	Другие
REAL	DATE	DWORD
LREAL	DT	LWORD
	TIME	WORD
	TOD	STRING

После чего в панели свойств пользователя появятся указанные настройки:

Иванов

Общие

Имя **Иванов**

Метки

Комментарий

Авторизация

Пароль

Служебные

Настройки

Телефон 0

Email

Дата рождения 01.01.0001 0:00:00

В режиме исполнения получить доступ к данным настройкам можно при помощи функциональных блоков UsersGetSettings и UsersSetSettings

6.2.2.7. НАСТРОЙКА ПРАВ ПОЛЬЗОВАТЕЛЕЙ В РЕЖИМЕ РАЗРАБОТКИ

В режиме исполнения можно организовать работу операторов с различными правами доступа, когда, например, одному пользователю разрешено управление и просмотр всех окон, а другому пользователю ограничен доступ к части окон и к управлению. Один оператор может при старте видеть одну мнемосхему, а другой - другую.

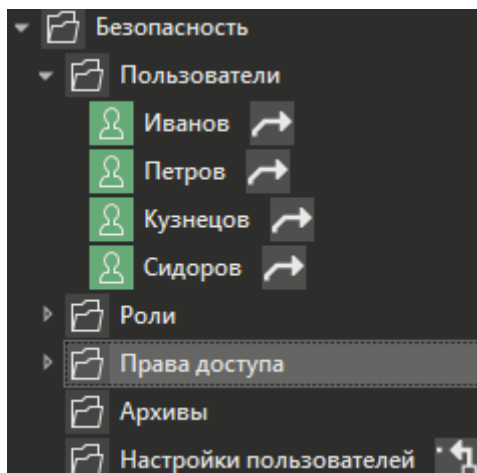
Настраивать права доступа для пользователей можно как индивидуально, так и для группы пользователей.

Для того чтобы в режиме исполнения работали различные операторы с различными правами доступа, необходимо выполнить следующие настройки:

- Создать пользователя,
- Сформировать роли (при необходимости),
- Задать глобальные права доступа,
- Уточнить права доступа к конкретным элементам (при необходимости).

Создание пользователя

Пользователи создаются в дереве системы в группе Безопасность. Пользователи.

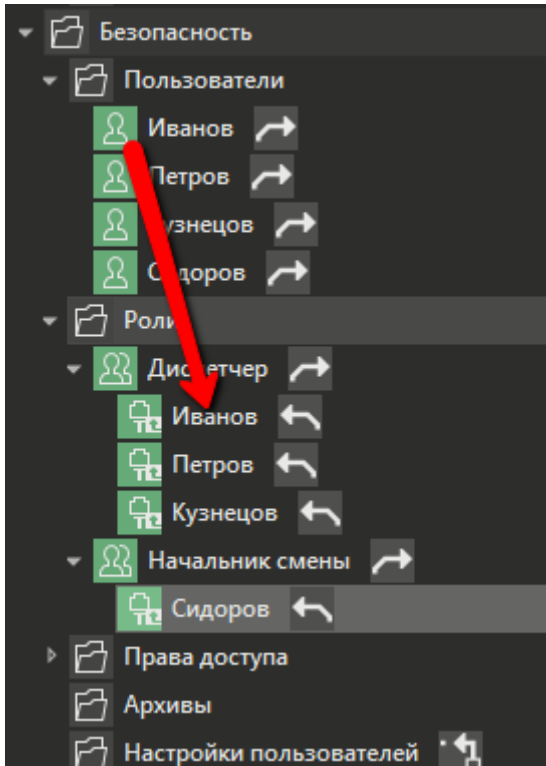


Панель свойств пользователя имеет вид:

Свойство	Описание
Имя	Текст, введенный в данном поле, будет отображаться при старте клиента визуализации в окне авторизации в поле Логин.
Пароль	Текст, введенный в данном поле, необходимо повторить в окне авторизации в поле Пароль.

Создание ролей пользователей.

Если несколько пользователей должны обладать одинаковыми правами доступа, то можно создать роль, а затем в нее перетащить нужных пользователей. В результате этого действия в группе Роли появится ссылка на пользователя.

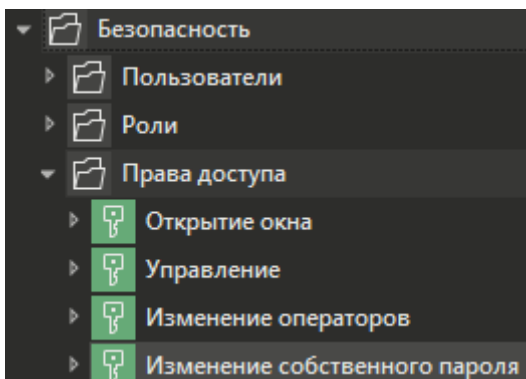


Если пользователь принадлежит нескольким ролям одновременно, то в режиме исполнения будут проверяться права всех ролей, в которые назначен пользователь:


- Если где-то задан запрет, то формируется общий запрет
- Если нет запрета, и где-то задано разрешение, то действие разрешается
- Если где-то установлены флаги Журнал и Подтверждение, то для этого пользователя эти флаги будут всегда срабатывать.

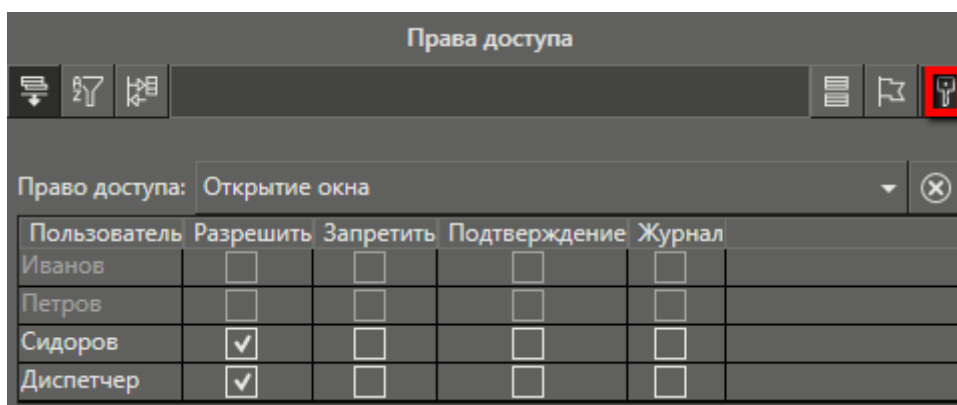
Настройка глобальных прав доступа

Права доступа можно разделить на два типа: Управление и Открытие окон. Именно эти категории прав доступа находятся в группе Безопасность. Права доступа.

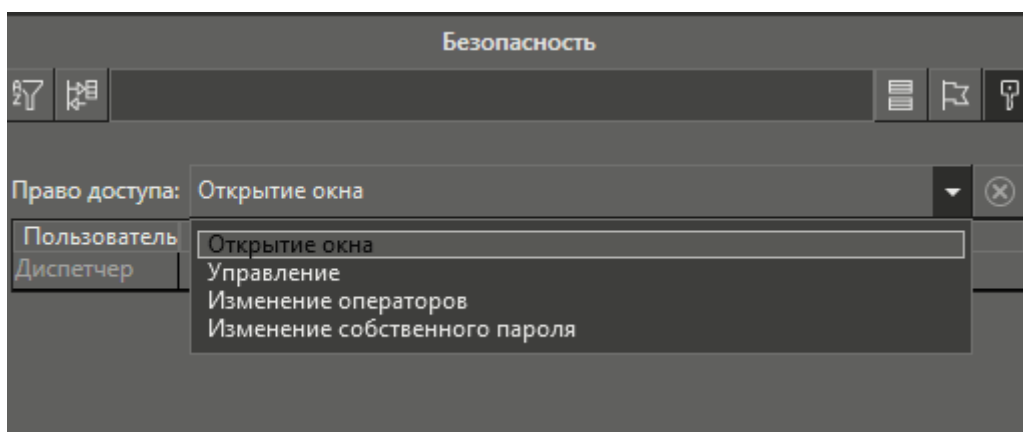


Права, заданные на уровне дерева системы, будут считаться умолчаниями для всех элементов проекта. В дальнейшем их можно будет переопределить.

Для того чтобы сделать настройки, необходимо в панели свойств группы Права доступа нажать на кнопку , и в таблице установить нужные флаги.



В выпадающем списке Право доступа, необходимо выбрать какое именно право планируется настраивать:



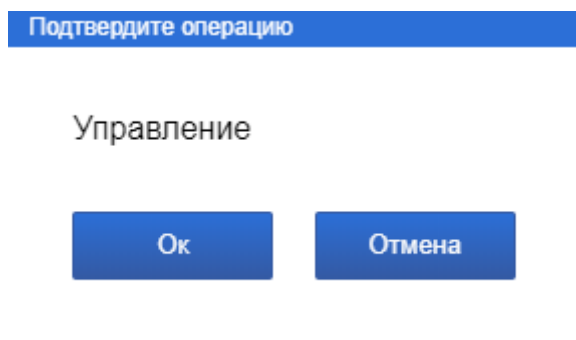
В таблице в столбце Пользователь показан список пользователей и ролей, которые ранее были добавлены в проект.

Каждому пользователю (роли) можно разрешить или запретить действие, установив флаг в соответствующей строке.

Флаг, установленный в столбце Запретить, приведет к запрету всех подобных действий по умолчанию для всего проекта. Т.е. если установлены одновременно флаг Разрешить и флаг Запретить, то это означает, что действие для данного пользователя будет запрещено.

Если флаги Разрешить и Запретить не установлены, то это значит, что в текущий момент, пользователь не сможет выполнять подобные действия, но в дальнейшем это действие можно будет разрешить для конкретного элемента.

Для предотвращения случайных нажатий необходимо установить флаг в столбце Подтверждение. Тогда в режиме исполнения, прежде чем выполнится изменение связанного параметра, появится диалоговое окно:



Если нажать на кнопку Отмена, то действие не выполнится.

Для фиксации выполненных действий пользователей необходимо установить флаг в столбце Журнал. В этом случае будут создаваться сообщения, сохраняемые в базе данных сообщений. Просмотреть сообщения в клиенте визуализации можно будет при помощи элемента Журнал.

Тексты в окне сообщения и записи о произведенном действии формируется автоматически. Изменить текст сообщения можно, если действие было настроено через события графических элементов.

Если необходимо фиксировать не только выполненные действия, но и попытки их выполнения, то необходимо дополнительно в настройках роли установить флаг Фиксировать попытки выполнения действий


Важно! Изменить имя пользователя, его пароль, назначение на роль и производить другие действия над ним в режиме исполнения в текущей версии невозможно. Поэтому если требуется выполнять в режиме исполнения какие-либо действия с пользователями необходимо создавать их в режиме исполнения.

Настройка прав доступа элементов

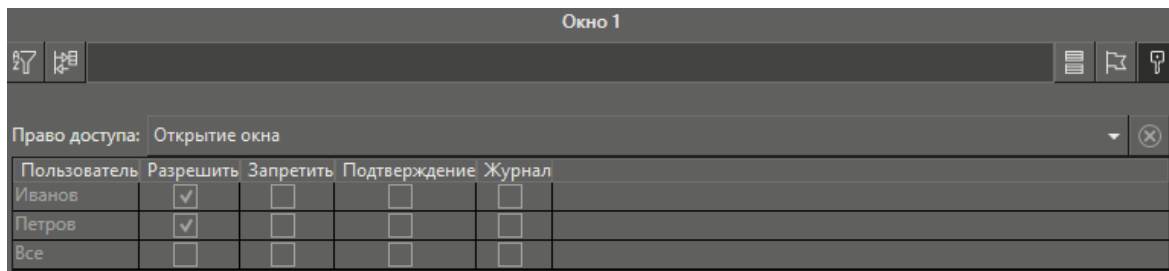
Права доступа конкретного элемента наследуются от его родителей, и могут быть переопределены в непосредственном элементе.

Порядок наследования:

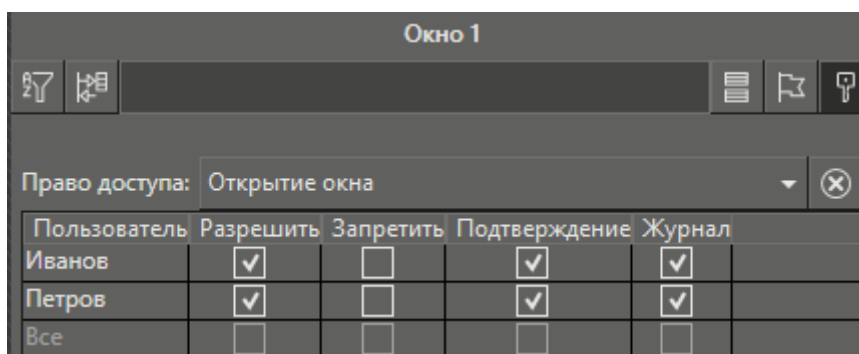
- Глобальные права доступа - настраиваются в дереве системы в группе Безопасность, являются умолчаниями для всех элементов проекта
- Права объектов - настраиваются в панели свойств Объекта, являются умолчаниями для всех его дочерних объектов, окон и элементов этих окон.
- Права окон - настраиваются в панели свойств Окна, являются умолчаниями для всех его элементов.
- Права элемент - настраивается у конкретного элемента.

Для настройки права доступа конкретного элемента необходимо нажать на кнопку  в его панели свойств

Если у элемента используются права доступа от его родителя, то панель свойств будет выглядеть так:



Если какое-либо право было изменено для данного элемента, то панель свойств будет иметь вид:



Т.е. измененные значения будут отображаться ярким шрифтом.

Рассмотрим пример определения права Управление для кнопок. Пусть есть четыре кнопки: первые две кнопки находятся в одном объекте, но в разных окнах, третья и четвертая находятся в другом объекте, в одном и том же окне, и если сессию начнет пользователь, для которого права доступа будут настроены следующим образом:

	Кнопка 1	Кнопка 2	Кнопка 3	Кнопка 4
Глобальные	Разрешено	Разрешено	Разрешено	Разрешено
Право Объекта	Право не переопределялось	Право не переопределялось	Право переопределено - Запрещено	Право переопределено - Запрещено
Право Окна	Право переопределено - Запрещено	Право не переопределялось	Право не переопределялось	Право не переопределялось

Право Элемента	Право не переопределялось	Право не переопределялось	Право не переопределялось	Право переопределено - Разрешено
Итог	Запрещено	Разрешено	Запрещено	Разрешено

Старт проекта с настроенными правами доступа.

При старте клиента визуализации производится проверка прав доступа для текущего пользователя. В процессе работы, для каждого залогинившегося оператора будет идти проверка прав доступа.

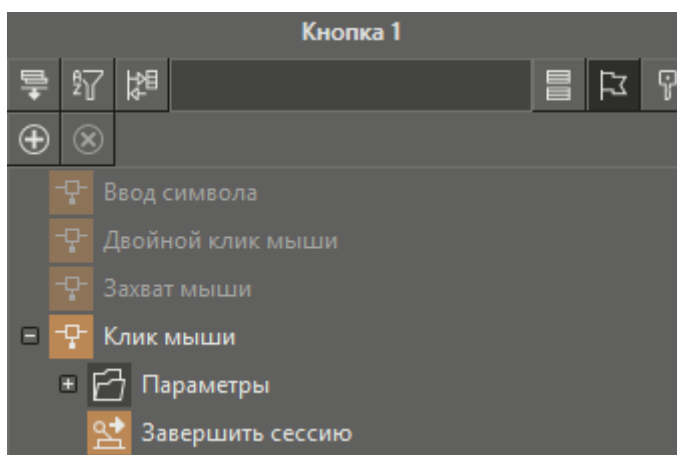
В случае, если пользователь принадлежит разным ролям с разными правами доступа, то тогда все настройки будут суммироваться. Т.е. если в какой-либо группе действие запрещено, то пользователь выполнить его не сможет.

Если планируется добавлять пользователей и совершать другие действия в режиме исполнения, то порядок настройки будет отличаться. Подробнее об этом смотрите в разделе Работа с пользователями в режиме исполнения

Организация завершения сессии оператора

Сессия работы оператора завершится в следующих ситуациях:

1. Если будет закрыт графический клиент.
2. Если истечет время с момента авторизации пользователя, заданное в свойстве Длительность сессии в настройках Роли.
3. Если будет выполнено действие Завершение сессии. Например, разработчик проекта создал кнопку в окне, а затем в ее событие Клик мыши добавил действие:

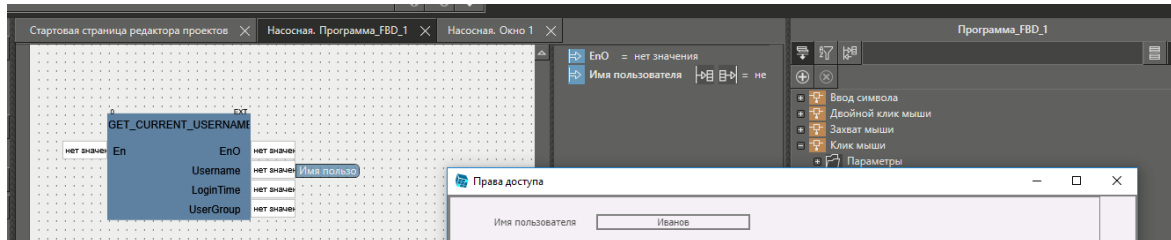


6.2.2.7.1. ПОЛУЧЕНИЕ ПАРАМЕТРОВ ТЕКУЩЕГО ПОЛЬЗОВАТЕЛЯ

Для того, чтобы в режиме исполнения получать параметры текущего пользователя, необходимо добавить в проект ФБ GET_CURRENT_USERNAME .

Важно! Данный функциональный блок должен работать в задаче экрана. Если открыто несколько клиентов, то имена пользователей могут отличаться.

Параметры функционального блока нужно разместить в окне клиента.

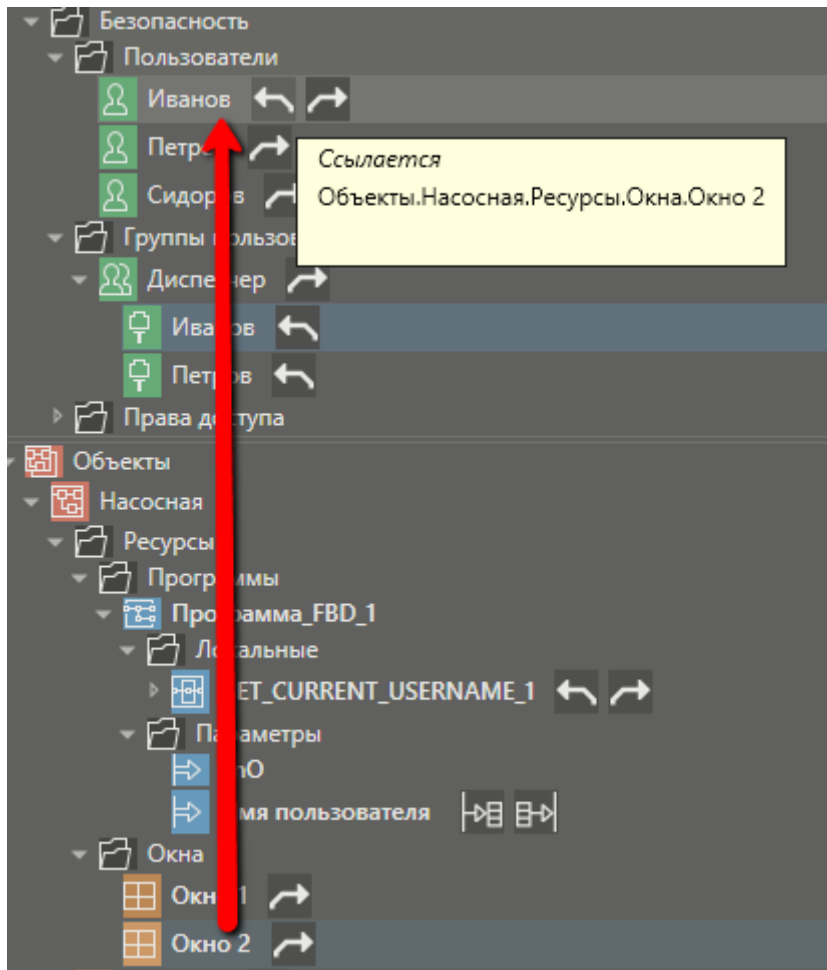


Важно! В интерфейсе среды разработки, подключенной к среде исполнения, значения параметров, исполняемых в задаче экрана, не отображаются, т.к. в один момент времени может быть подключено множество клиентов, и значения параметров везде будут разными

6.2.2.7.2. ЗАДАНИЕ СТАРТОВОГО ОКНА ДЛЯ ПОЛЬЗОВАТЕЛЯ И РОЛИ

Для пользователя

Для того, чтобы задать окно, которое должно открываться при успешной авторизации пользователя, необходимо в дереве перетащить окно на пользователя. В этом случае формируется ссылка на окно:

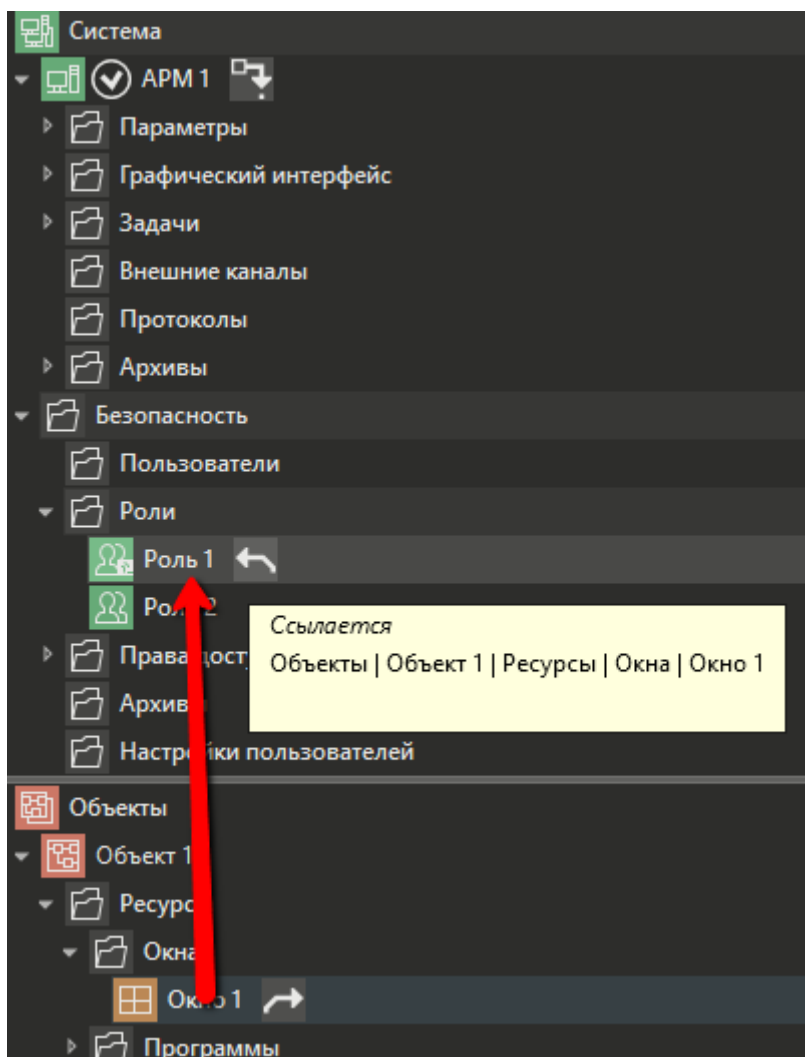


Если начальное окно для пользователя не задано, то при авторизации этого пользователя при запуске проекта откроется окно, установленное как стартовое, стартовая мнемосхема.

Для роли

Для пользователей, которые добавлены в режиме исполнения, стартовое окно будет зависеть от того, в какую роль они будут назначены.

Для того, чтобы задать окно, которое должно открываться при успешной авторизации пользователя, который добавлен в ту или иную роль при запуске клиента визуализации, необходимо в дереве перетащить окно на Роль. В этом случае формируется ссылка на окно:



Если начальное окно для роли не задано, то при авторизации этого пользователя при запуске проекта откроется окно, установленное как стартовое, стартовая мнемосхема.

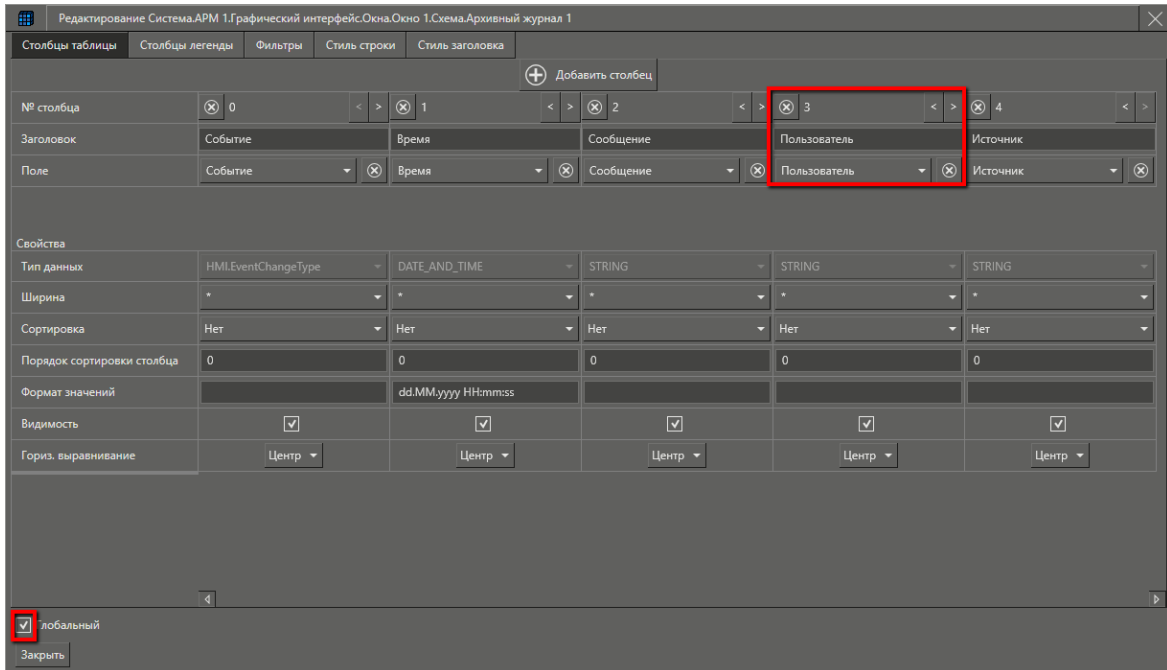
Важно! Если назначены разные стартовые окна для пользователя и для роли, в которую входит этот пользователь, то приоритетнее будет то окно, которое назначено пользователю. Если пользователь назначен в две роли, с разными окнами, то загрузится то окно, которое назначено роли, которая находится выше в дереве.

6.2.2.7.3. ЖУРНАЛ ДЕЙСТВИЙ ПОЛЬЗОВАТЕЛЕЙ

Рассмотрим пример создания Архивного журнала, в котором будут отображаться только сообщения о действиях пользователей.

В группе Безопасность добавлено нужное количество пользователей и ролей. В настройках указано, что действия должны фиксироваться в базе данных.

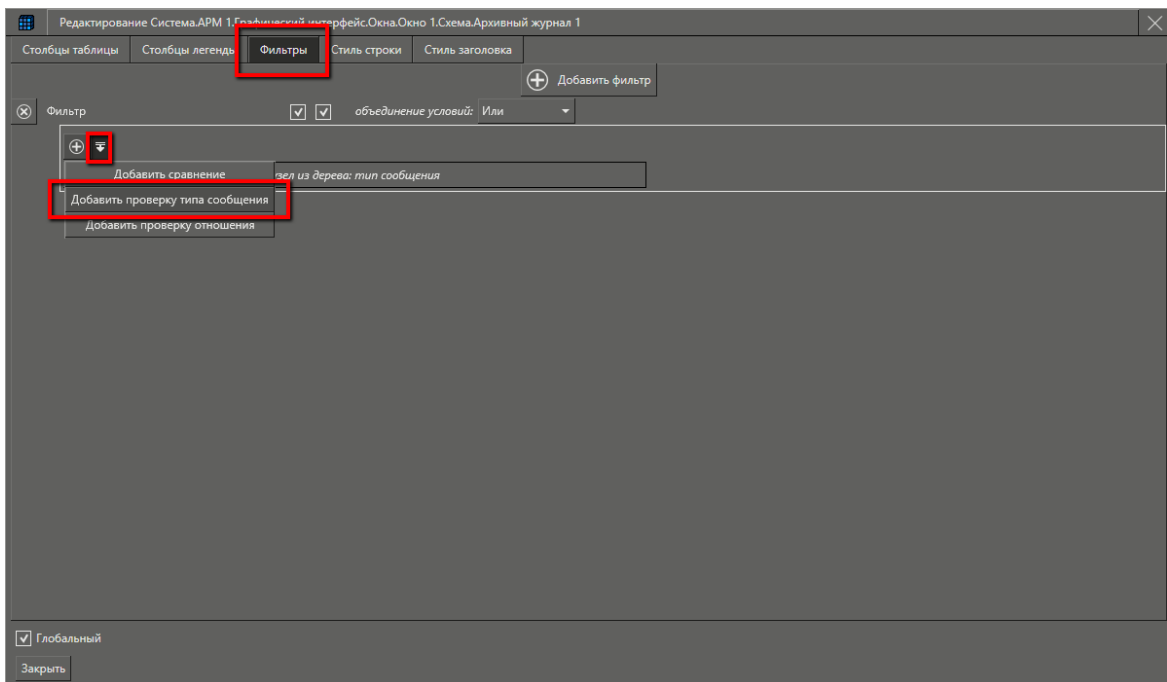
Разместим в окне элемент Архивный журнал. В диалоговом окне настройки журнала добавим столбец, который будет отображать какой Пользователь совершил, то или иное действие:



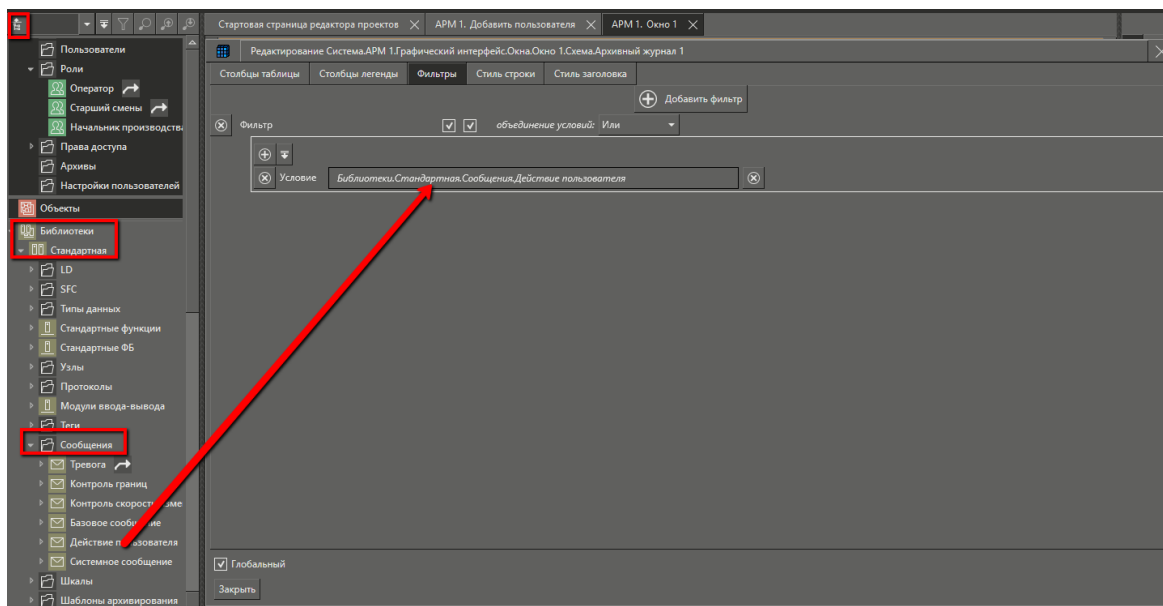
Если установить флаг Глобальный, то в журнал будут попадать все сообщения узла, удовлетворяющие фильтрам.

Далее необходимо настроить фильтры таким образом, чтобы в журнале отображались только сообщения о действиях пользователей.

Вначале добавляем фильтр, который будет проверять тип сообщения:



Появится новый фильтр. В строку фильтра нужно перетащить из Стандартной библиотеки тип сообщения Действия пользователей

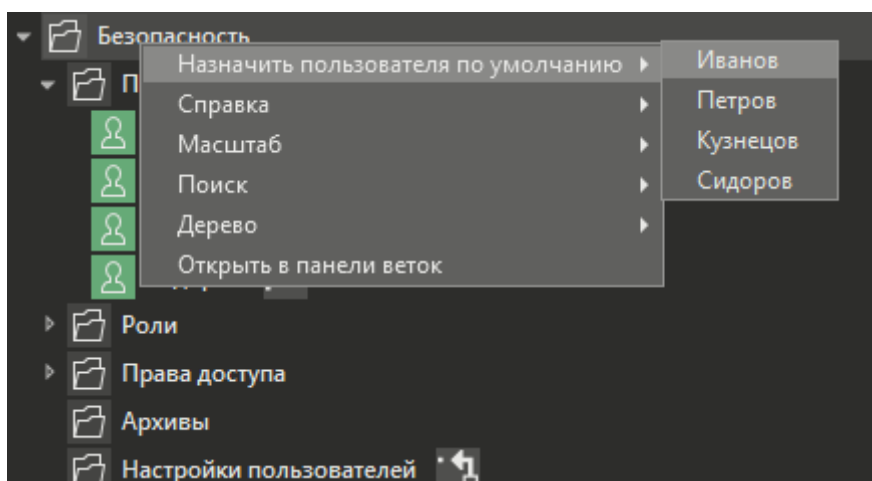


После старта среды исполнения в клиенте визуализации в таком журнале будут отображаться все действия операторов, которые отмечены флагами в настройках прав доступа, а также информация о начале и окончании сессии пользователя.

6.2.2.7.4. НАСТРОЙКА ПОЛЬЗОВАТЕЛЯ ПО УМОЛЧАНИЮ

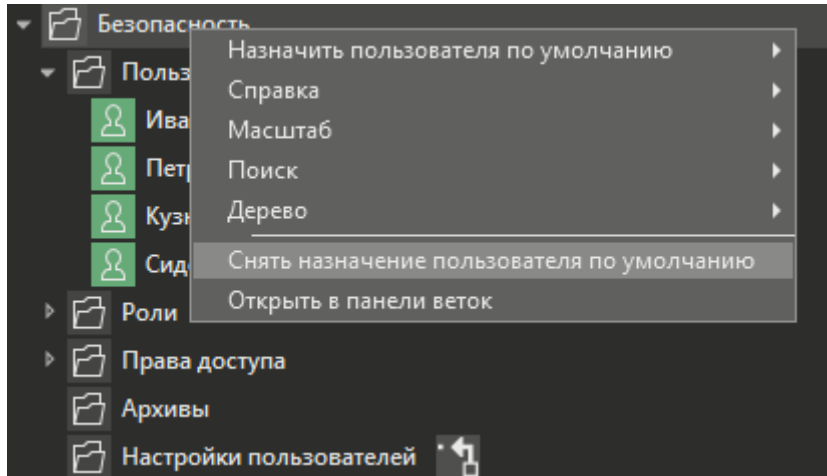
Для автоматического подключения клиента визуализации к исполнительной системе, минуя диалоговое окно авторизации, имеется возможность настроить пользователя по умолчанию.

Для этого в контекстном меню группы Безопасность есть пункт Назначить пользователя по умолчанию:



В результате этих действий, при подключении клиента визуализации к любому из узлов проекта происходит логин под этим пользователем.

Для удаления пользователя по умолчанию можно воспользоваться контекстным меню группы Безопасность пунктом Снять назначение пользователя по умолчанию:

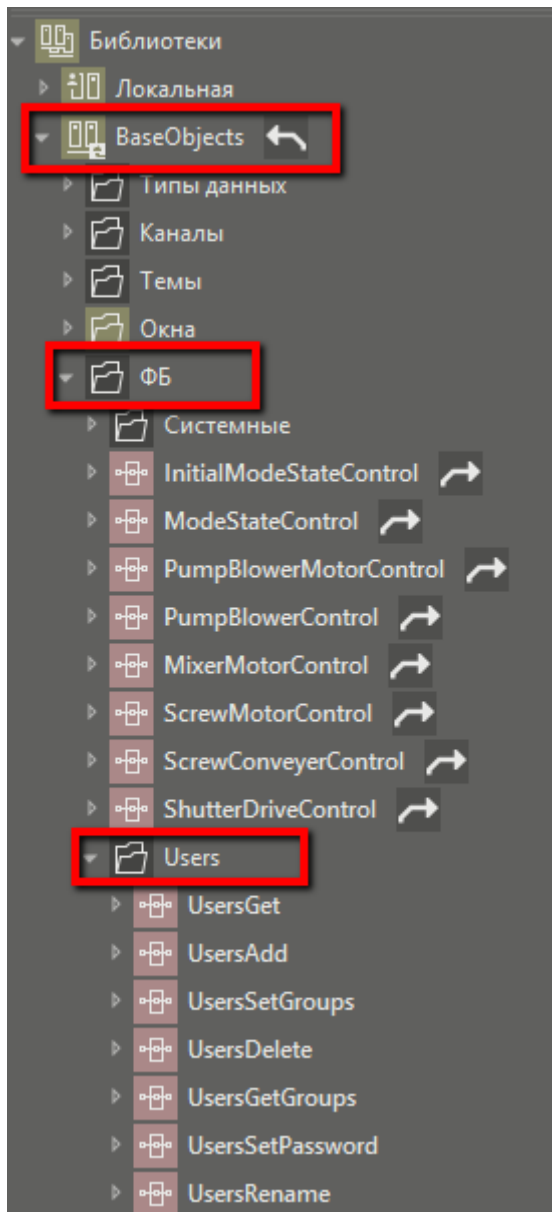


6.2.2.8. РАБОТА С ПОЛЬЗОВАТЕЛЯМИ В РЕЖИМЕ ИСПОЛНЕНИЯ

В режиме исполнения можно выполнять следующие действия с пользователями: добавлять новых пользователей, менять добавленным в режиме исполнения новым пользователям имя, пароль и назначение на роль.

Если одну из вышеупомянутых операций требуется делать в режиме исполнения, то порядок настройки прав доступа в режиме разработки будет таким:

1. Создаются Роли, в группе Безопасность в дереве системы.
2. Для ролей настраиваются права доступа, как глобальные (на уровне дерева системы), так и локальные (непосредственно в настройках каждого из элементов проекта) так, как это описано в разделе Настройка прав пользователей в режиме разработки.
3. Создается окно с элементами управления, которые будут связаны со специальными функциональными блоками из библиотеки BaseObjects.ФБ.Users:



Список функциональных блоков:

Название и ссылка	Назначение
UsersGet	Служит для получения списка пользователей, добавленных в режиме исполнения
UsersAdd	Служит для добавления пользователей в режиме исполнения
UsersSetGroups	Служит для назначения/изменения списка ролей пользователя, ранее созданного в режиме исполнения

Название и ссылка	Назначение
UsersDelete	Служит для удаления пользователя, ранее добавленного в режиме исполнения.
UsersGetGroups	Служит для получения списка ролей, в которые добавлен определенный пользователь
UsersSetPassword	Служит для назначения/изменения пароля пользователя, ранее добавленного в режиме исполнения
UsersRename	Служит изменения имени существующего пользователя, который был добавлен в режиме исполнения
UsersGetSettings	Является типом для экземпляров функциональных блоков, расширенных разработчиками проекта, предназначенных для получения значений дополнительных настроек пользователей в режиме исполнения.
UsersSetSettings	Является типом для экземпляров функциональных блоков, расширенных разработчиками проекта, предназначенных для изменения значений дополнительных настроек пользователей в режиме исполнения.
UsersChangeOwnPassword	Позволяет сменить собственный пароль
UsersGetADUserGroups	Служит для получения данных о пользователе добавленном в ActiveDirectory и получения его групп

Именно при помощи данных ФБ можно организовать работу с пользователями в режиме исполнения.

4. Создается пользователь в режиме разработки, который обладает доступом к новому окну и к элементам управления, созданным разработчиком проекта, а также имеет право доступа изменение пользователей

5. После загрузки проекта в узел, в режиме исполнения формируется список пользователей при помощи созданного окна. Права доступа назначаются пользователям путем добавления их на нужные роли.

Важно! Если пользователь не назначен ни на одну роль, то он не будет обладать правами доступа, позволяющими ему выполнять какие-либо действия в режиме исполнения

В результате работы ФБ UsersAdd создается файл *users.db* БД Sqlite на том узле, где выполняется данный функциональный блок. Все другие функциональные блоки группы BaseObjects.ФБ.Users будут работать с этим файлом.

В случае, если необходимо организовывать работу с операторами в режиме исполнения в различных проектах, то можно разработать библиотечный объект, который будет содержать в себе необходимые функциональные блоки и окно, и использовать этот объект в различных проектах

6.2.2.9. ACTIVE DIRECTORY

Active Directory (AD) — это технология управления компьютерами и устройствами в сети.

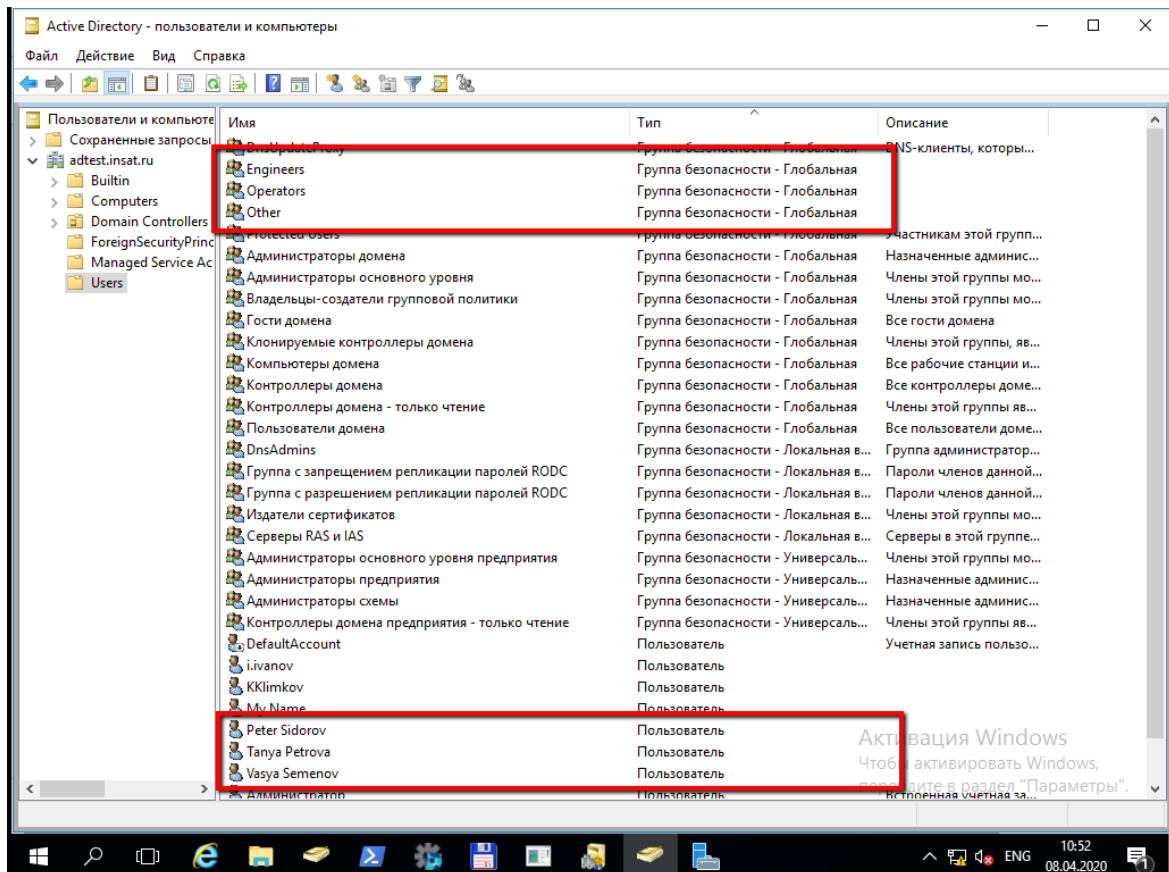
Важно! Функция поддерживается только для исполнительных систем, которые работают под ОС Windows.

Active Directory состоит из нескольких служб для операционной системы Windows Server, позволяющий создавать домены и управлять ими, контролировать все объекты и пользователей в сети. С помощью групповых политик из одной точки устанавливаются параметры доступов и разрешений на остальных компьютерах и устройствах. Предназначение Active Directory - упростить выполнение задач системному администратору в больших сетевых структурах.

Доменные службы (AD DS) управляют связью между пользователями и доменами, включая процесс аутентификации и поиска. С их помощью можно создавать пользователей и группы безопасности с разным доступом к ресурсам операционной системы <https://docs.microsoft.com/ru-ru/windows/security/identity-protection/access-control/active-directory-security-groups>.

Пример настроек Active Directory Windows

На приведенном ниже рисунке изображены созданные администратором три группы безопасности и три пользователя, назначенные в эти группы:



Каждый пользователь имеет свои собственные настройки.


Пользователь Peter Sidorov назначен в группы Engineers и Operators. Имя входа в операционную систему p.sidorov:

Свойства: Peter Sidorov

Профиль служб удаленных рабочих столов COM+

Член групп Входящие звонки Среда Сеансы Удаленное управление

Общие Адрес Учетная запись Профиль Телефоны Организация

 Peter Sidorov

Имя: Инициалы:

Фамилия:

Выводимое имя:

Описание:

Комната:

Номер телефона:

Эл. почта:

Веб-страница:

Свойства: Peter Sidorov ? X

Профиль служб удаленных рабочих столов COM+

Член групп Входящие звонки Среда Сеансы Удаленное управление

Общие Адрес Учетная запись Профиль Телефоны Организация

Имя входа пользователя:
p.sidorov @adtest.insat.ru

Имя входа пользователя (пред-Windows 2000):
ADTEST0\ p.sidorov

Время входа... Вход на...

Разблокировать учетную запись

Параметры учетной записи:

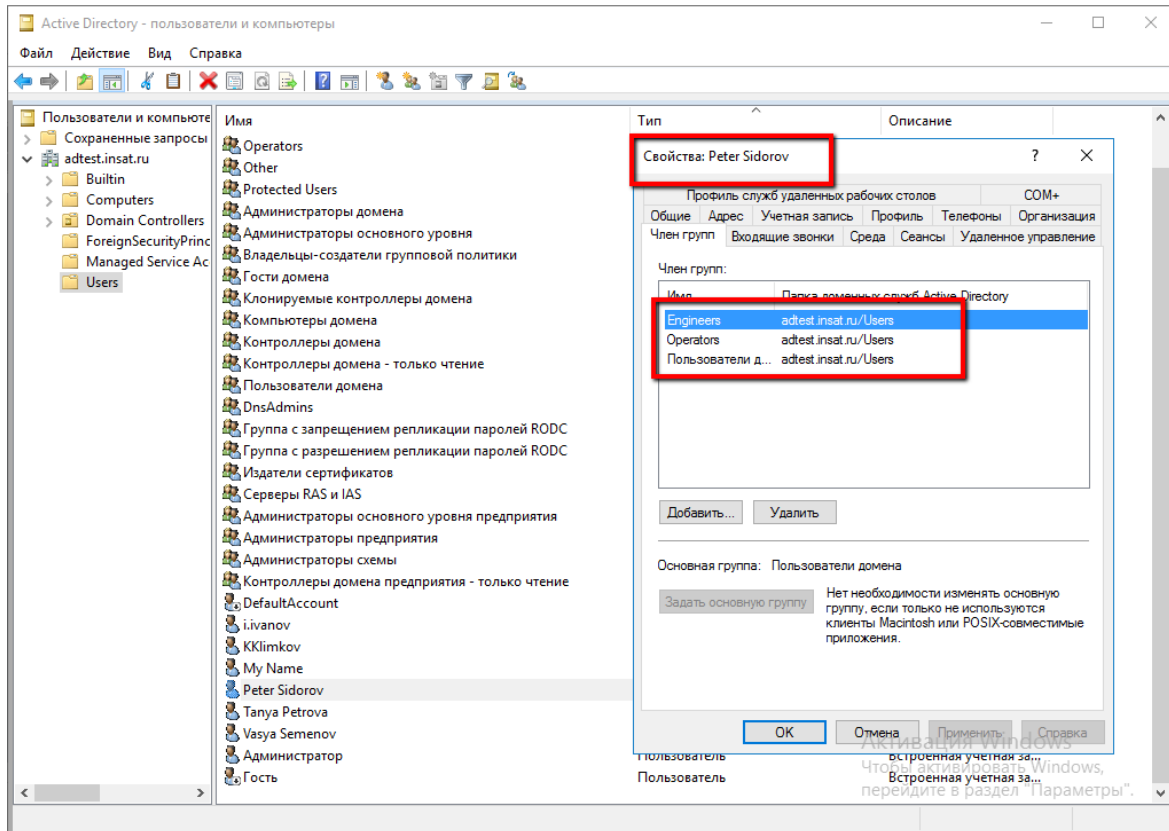
- Требовать смены пароля при следующем входе в систему
- Запретить смену пароля пользователем
- Срок действия пароля не ограничен
- Хранить пароль, используя обратимое шифрование

Срок действия учетной записи

Никогда

Истекает: 8 мая 2020 г.


OK Отмена Применить Справка



Пользователь Tanya Petrova назначен в группу Operators. Имя входа в систему t.petrova

Свойства: Tanya Petrova ? X

Член групп	Входящие звонки	Среда	Сеансы	Удаленное управление	
Профиль служб удаленных рабочих столов				COM+	
Общие	Адрес	Учетная запись	Профиль	Телефоны	Организация

 Tanya Petrova

Имя: Инициалы:

Фамилия:

Выводимое имя:

Описание:

Комната:

Номер телефона:

Эл. почта:

Веб-страница:

Активация Windows

Свойства: Tanya Petrova

Член групп | Входящие звонки | Среда | Сеансы | Удаленное управление

Профиль служб удаленных рабочих столов | COM+

Общие | Адрес | Учетная запись | Профиль | Телефоны | Организация

Имя входа пользователя:
t.petrova @adtest.insat.ru

Имя входа пользователя (пред-Windows 2000):
ADTEST0\ t.petrova

Время входа... | Вход на...

Разблокировать учетную запись

Параметры учетной записи:

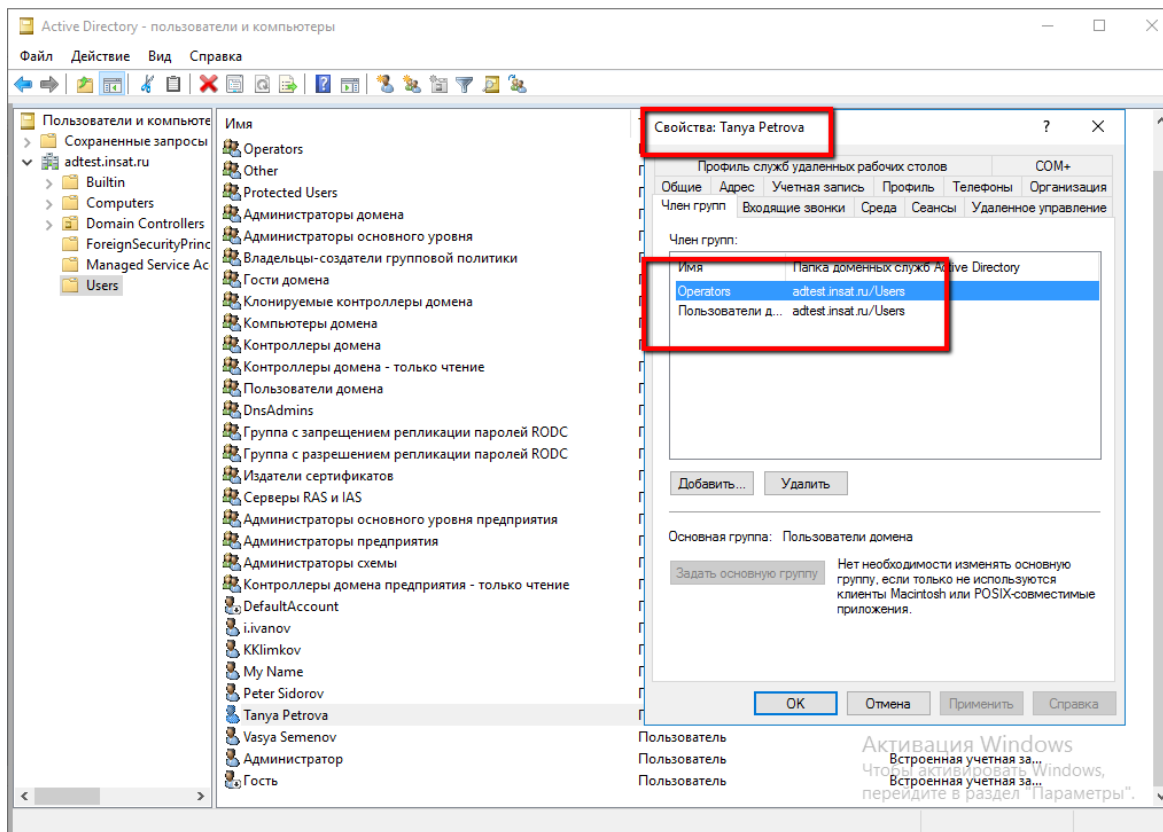
- Требовать смены пароля при следующем входе в систему
- Запретить смену пароля пользователем
- Срок действия пароля не ограничен
- Хранить пароль, используя обратимое шифрование

Срок действия учетной записи:

Никогда

Истекает: 8 мая 2020 г.

OK | Отмена | Применить | Справка




Пользователь Vasya Semenov назначен в группу Other. Имя входа в систему v.semenov:

Свойства: Vasya Semenov ? X

Член групп Входящие звонки Среда Сеансы Удаленное управление
Профиль служб удаленных рабочих столов COM+

Общие Адрес Учетная запись Профиль Телефоны Организация

 Vasya Semenov

Имя: Инициалы:

Фамилия:

Выводимое имя:

Описание:

Комната:

Номер телефона:

Эл. почта:

Веб-страница:

Свойства: Vasya Semenov ? X

Член групп Входящие звонки Среда Сеансы Удаленное управление
Профиль служб удаленных рабочих столов COM+

Общие Адрес Учетная запись Профиль Телефоны Организация

Имя входа пользователя:
 @adtest.insat.ru

Имя входа пользователя (пред-Windows 2000):

Разблокировать учетную запись

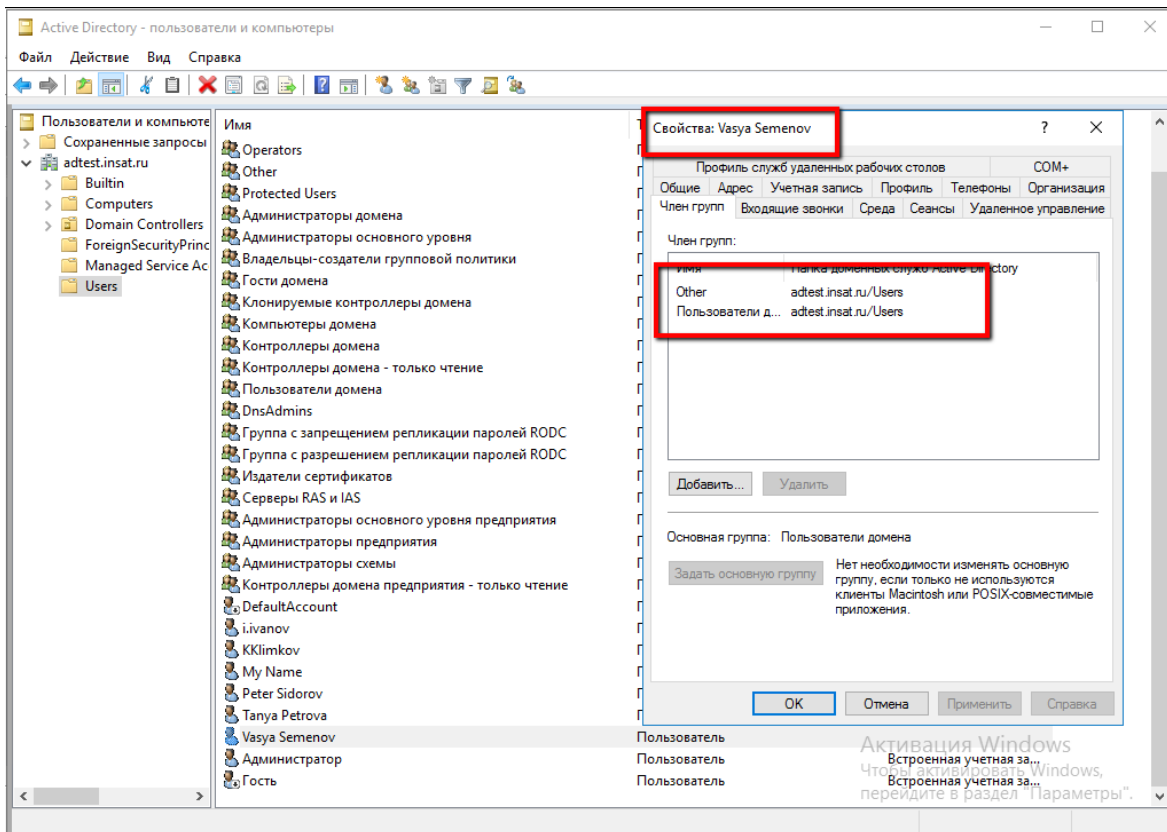
Параметры учетной записи:

- Требовать смены пароля при следующем входе в систему
- Запретить смену пароля пользователем
- Срок действия пароля не ограничен
- Хранить пароль, используя обратимое шифрование

Срок действия учетной записи

Никогда

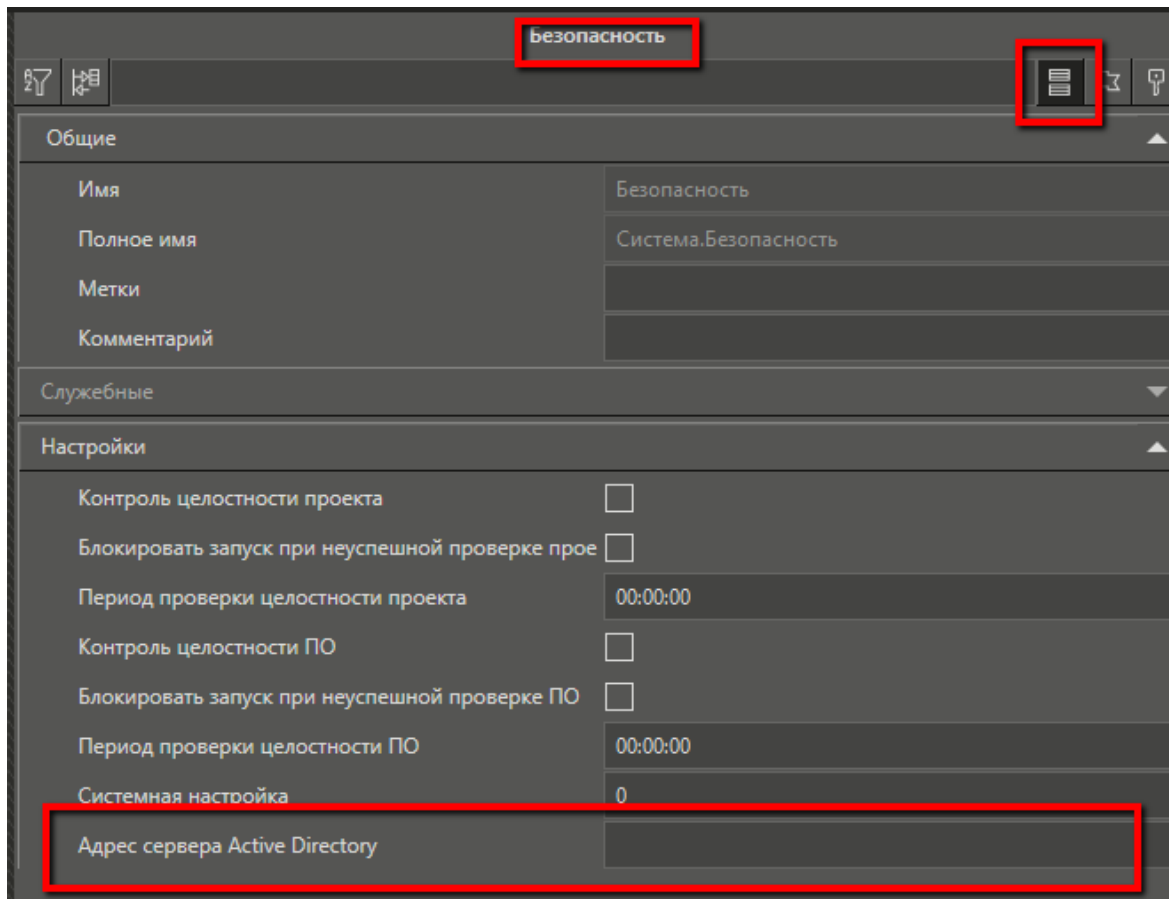
Истекает:



Взаимодействие MasterSCADA 4D и ActiveDirectory

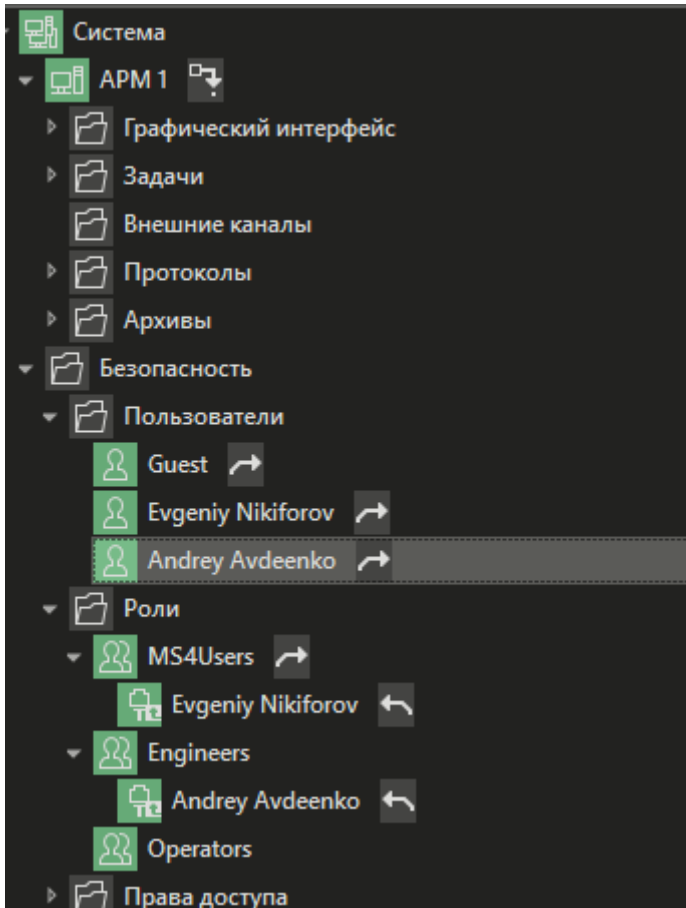
MasterSCADA 4D позволяет объединить встроенную систему Безопасности и группы Active Directory.

Для того, чтобы в клиент визуализации можно было зайти под пользователем AD, нужно настроить свойство элемента Безопасность Адрес сервера Active Directory:



В данном поле можно задать как host так и IP-адрес. Если значение не задано, то при попытке подключения клиента визуализации в окне авторизации список пользователей отображаться не будет.

Также необходимо создать роли, соответствующие группам, добавленным в AD. В операционной системе заведены три группы безопасности Engineers, Operators и Other. Создадим роли Engineers, Operators в элементе Безопасность:



В созданные роли можно также добавить и пользователей, которые созданы в MasterSCADA 4D.

Затем необходимо назначить права доступа созданным ролям. Обязательно необходимо настроить право доступа Открытие окна:

Безопасность					
Право доступа: Открытие окна					
Пользователь	Разрешить	Запретить	Подтверждение	Журнал	
Guest	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Evgeniy Nikiforov	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Andrey Avdeenko	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
MS4Users	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Engineers	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Operators	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

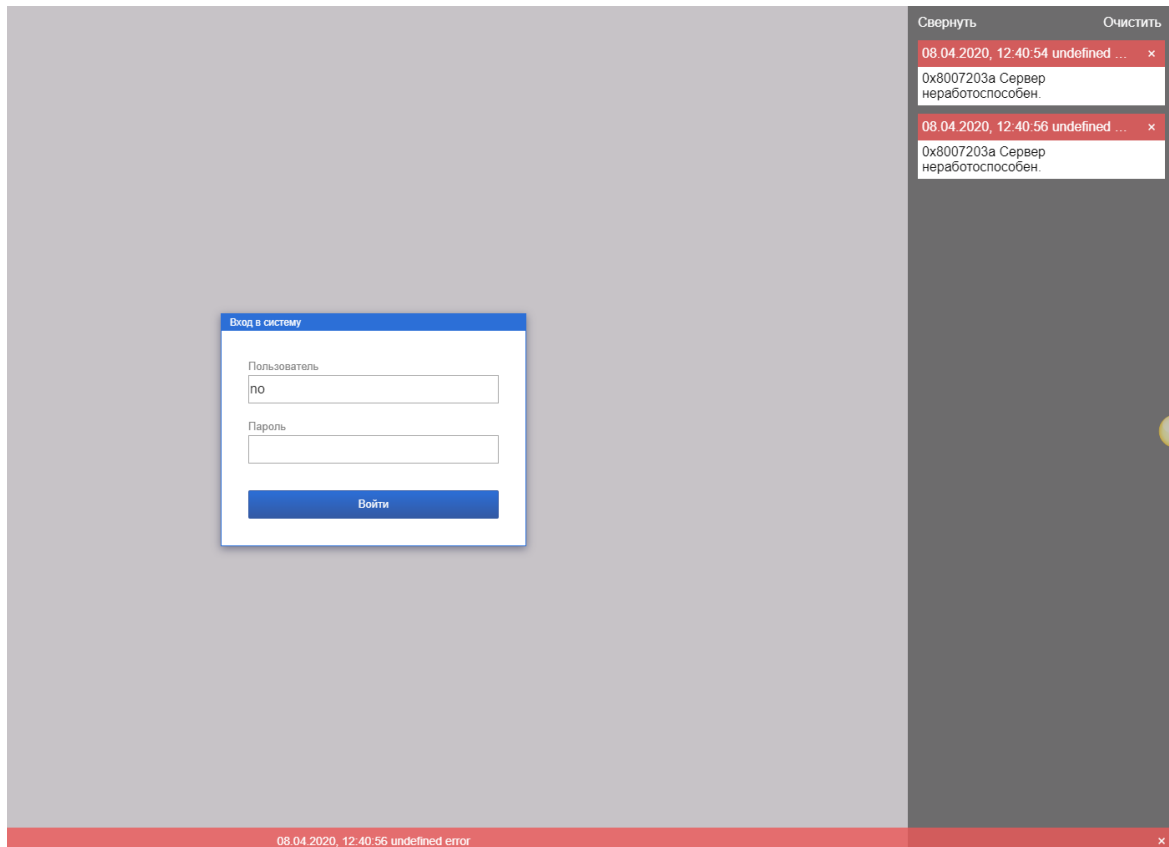
А также права доступа Управление:

Безопасность					
Право доступа: Управление					
Пользователь	Разрешить	Запретить	Подтверждение	Журнал	
Guest	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Evgeniy Nikiforov	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Andrey Avdeenko	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
MS4Users	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Engineers	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Operators	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Другие права доступа настраиваются в зависимости от поставленных задач.

Далее в режиме исполнения при подключении клиента визуализации в окне авторизации пользователей необходимо ввести либо имя пользователя созданного в MasterSCADA 4D, либо пользователя добавленного в ActiveDirectory

Если сервер AD недоступен, то после попытки авторизации появится ошибка в клиенте визуализации:



Получить информацию о добавленных в AD пользователях можно при помощи специального служебного функционального блока UsersGetADUserGroups

6.2.3. ПОЛУЧЕНИЕ И ОТПРАВКА ДАННЫХ

MasterSCADA 4D поддерживает получение/отправку данных как в режиме Master, так и в режиме Slave.

Master

Для того чтобы MasterSCADA 4D при обмене данными с модулями ввода-вывода (другими приложениями) выступала в роли Master, необходимо настроить группу узла Протоколы

В базовую версию среды исполнения входят протоколы:

- Modbus RTU
- Modbus TCP
- DCON
- OPC UA
- OPC DA (только для исполнительной системы для Windows)
- OPC HDA (только для исполнительной системы для Windows)

Помимо этого, можно дополнить список используемых протоколов в среде исполнения, приобретя соответствующую лицензию.

Среда разработки MasterSCADA 4D поставляется с API, которое позволяет разработать свои собственные протоколы. Ознакомиться с документацией можно в папке установки среды разработки: *C:\Program Files (x86)\MPSSoft\MasterSCADA 4D 1.2\API\Doc\Разработка драйверов и ФБ в MasterSCADA 4D.docx*

Важно! Среда разработки позволяет добавить в проект любой из представленных в библиотеке протоколов. Однако, среда исполнения может не поддерживать тот или иной протокол из-за ограничений ОС либо из-за ограничения приобретенной лицензии на среду исполнения.

Slave

Для того чтобы MasterSCADA 4D работала в роли Slave, необходимо настроить группу узла Внешние каналы для передачи данных по протоколам:

- Modbus RTU
- Modbus TCP
- OPC DA

Либо настроить панель свойств узла для использования протоколов:

- OPC UA
- IEC 104

Смотрите так же:

Порядок вычислений и передачи данных

6.2.3.1. СТАНДАРТНЫЕ ПРОТОКОЛЫ

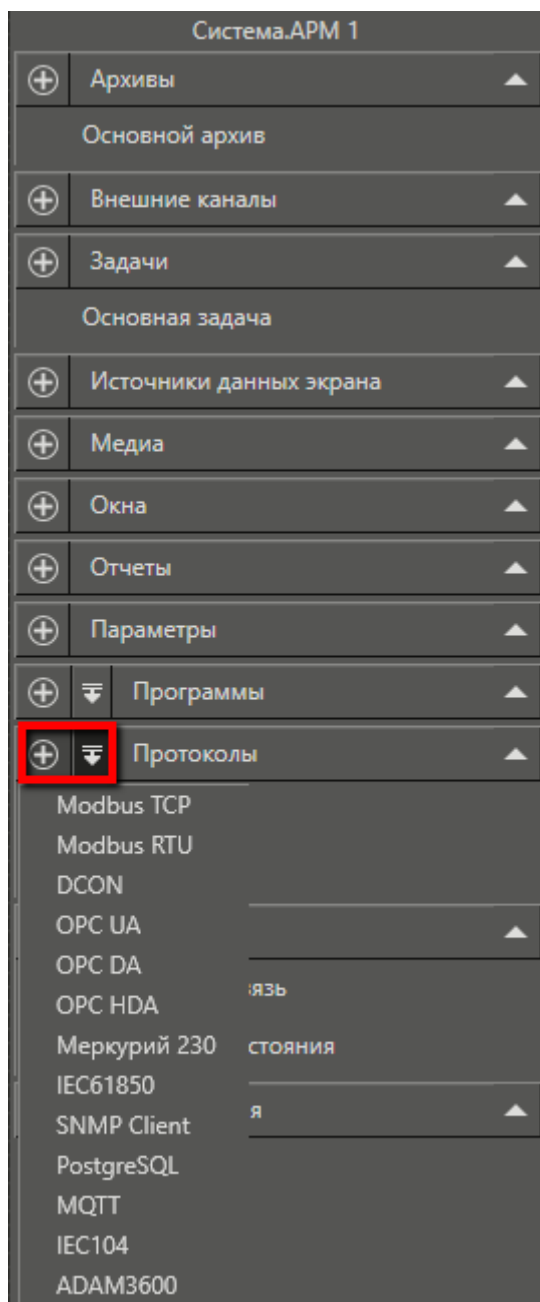
В базовый комплект MasterSCADA 4D входят следующие протоколы:

- Modbus RTU
- Modbus TCP
- DCON
- OPC UA
- OPC DA (только для исполнительной системы для Windows)
- OPC HDA (только для исполнительной системы для Windows)

Это означает, что любое устройство, на котором установлена исполнительная система поддерживает данные протоколы, при наличии соответствующих физических портов.

В дереве библиотек данные протоколы хранятся в категории Стандартная, в группе Протоколы.

В проект добавляются, как правило, через контекстную панель узла, либо контекстное меню узла, либо контекстное меню группы Протоколы.



6.2.3.1.1. MODBUS

Modbus TCP

Свойства протокола Modbus TCP

Модуль Modbus TCP

Modbus RTU

Свойства протокола Modbus RTU

Модуль Modbus RTU

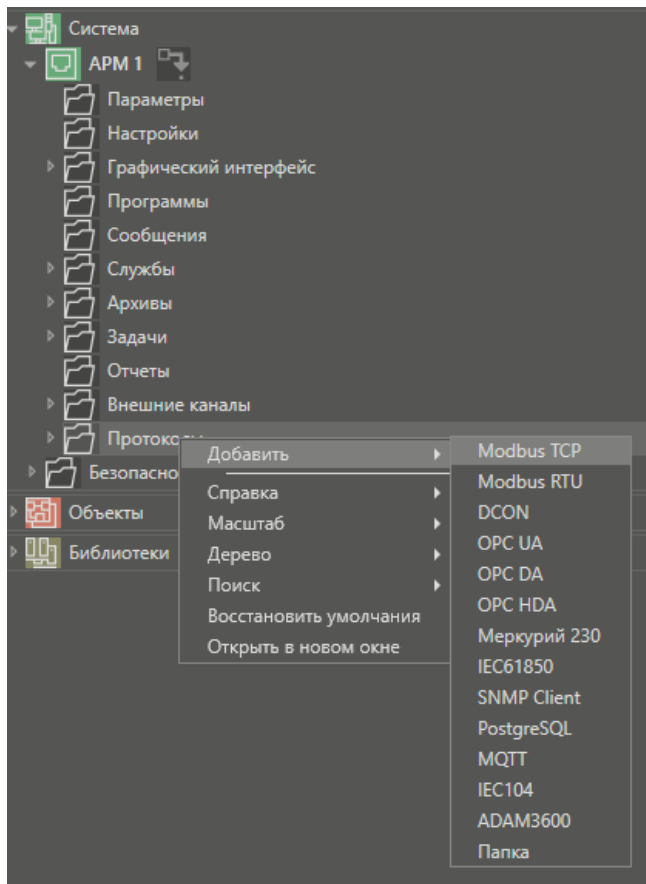
Свойства модулей Modbus TCP и Modbus RTU

Свойства каналов Modbus

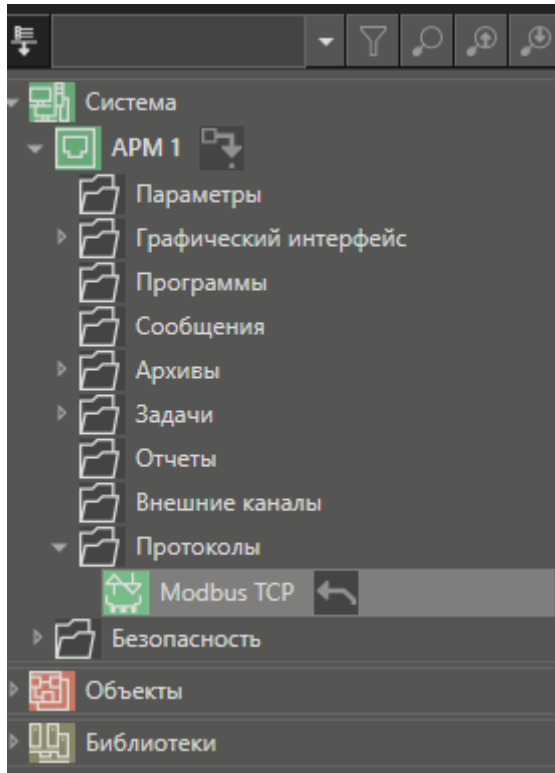
6.2.3.1.1.1. MODBUS TCP

Modbus TCP – это протокол обмена по MODBUS по сети по протоколу TCP/IP. При обмене узел играет роль ведущего (Master).

Чтобы добавить возможность опроса устройств по данному протоколу необходимо выбрать соответствующий элемент в контекстном меню узла, или в контекстном меню группы Протоколы, либо через контекстную панель узла:



Получим результат в упрощенном дереве:



Далее нужно настроить панель свойств. Затем добавить модуль ввода-вывода, который работает по этому протоколу.

При необходимости, в протоколе могут быть созданы программы, окна и др. элементы. В этом случае у протокола появится группа Ресурсы, такая же как и у элементов объект, тег, канал.

СВОЙСТВА ПРОТОКОЛА MODBUS TCP

Настройка элемента производится в панели свойств.

Вид панели свойств:

Modbus TCP

☰
☰

Общие ▲

Имя	Modbus TCP
Метки	
Комментарий	

Служебные ▼

Задача ▲

Период, мс	100
Приоритет	100
Подключение по условию	<input type="checkbox"/>
Выполнение по условию	<input type="checkbox"/>
Выполнять на резервном	<input type="checkbox"/>

Протокол ▲

Таймаут	1000
Количество повторов при неудачном опросе	2
Modbus поверх TCP	<input type="checkbox"/>

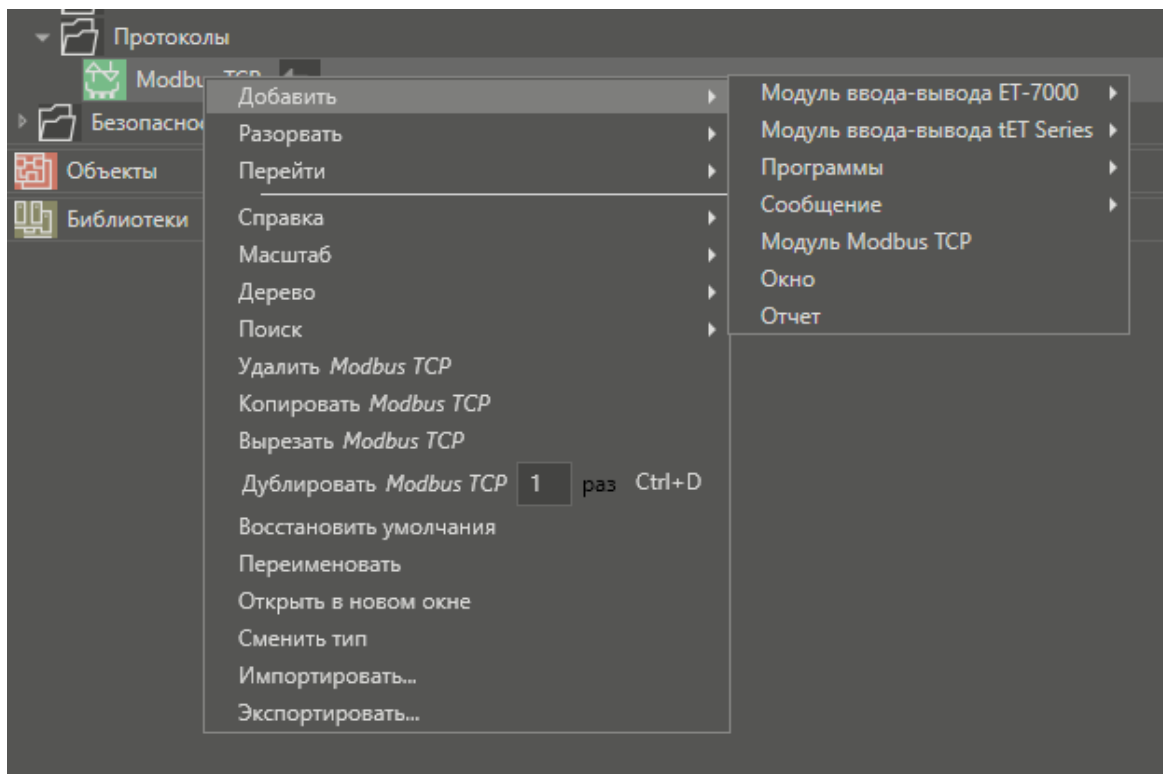
Описание:

Название	Рекомендации
Категория Общие	Как правило, для данного элемента не настраивается. Подробное описание данной категории приведено в соответствующем разделе.
Категория Задача	Задается период, с которым будут опрашиваться модули ввода-вывода. Определяется способ работы протокола при резервировании узла. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи у протоколов.
Таймаут	Задается значение таймаута по умолчанию для подключаемых модулей. Это максимальное время ожидания ответа от модуля ввода-вывода (мс). Если ответ от модуля не будет получен в течение заданного интервала времени, то попытка опроса будет

	считаться неудачной, и зафиксирована ошибка данного запроса для конкретного модуля.
Количество повторов при неудачном опросе	<p>Указывается предельное количество неудачных попыток получить ответ модуля ввода-вывода, произошедших подряд. Если за указанное количество попыток получить ответ от модуля не удалось, то сформируется сигнал TRUE у параметра Отказ у того модуля, с которым не удалось обнаружить связь.</p> <p>Если при обращении к модулю он не ответит ни разу за указанное количество попыток подряд, то следующие запросы к модулю в рамках данного цикла опроса посылаться не будут, сформируется отказ модуля и выставится признак недостоверности по всем входам этого модуля. Если при запросе ответ придет, но с ошибкой, то сформируется отказ модуля и выставится признак недостоверности по входам в данном запросе, но посылка других запросов по этому модулю продолжится.</p>
Modbus поверх TCP	<p>Свойство предназначено для работы с конвертерами Ethernet-COM (например, Moxa NPort5150). Если флаг установлен, то происходит упаковка стандартного запроса Modbus RTU в TCP-пакет, и сформированный пакет передается конвертеру. Конвертер извлекает из пакета запрос Modbus и направляет его в устройство.</p> <p>При ответе устройства преобразование происходит в обратном порядке. Если необходимо подключить несколько устройств через один преобразователь, то можно использовать протокол Modbus RTU over TCP.</p>

МОДУЛЬ MODBUS TCP

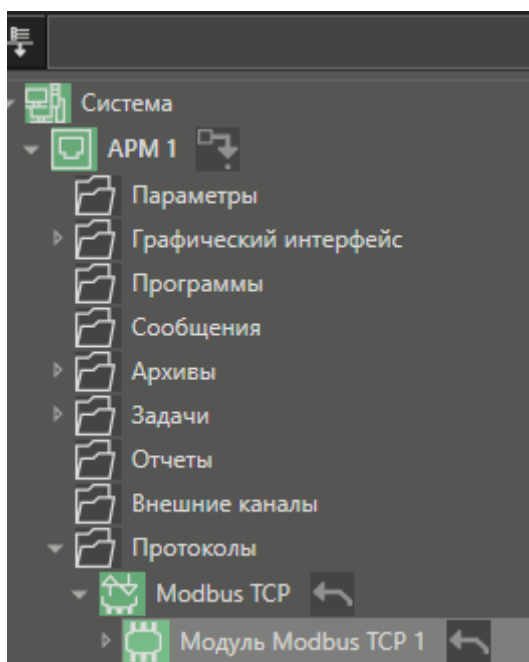
Модуль ввода-вывода добавляется в дерево системы в соответствующий протокол при помощи контекстного меню протокола:



В проект можно добавить как predetermined модули ввода-вывода (т.е. ранее созданные в библиотеке компанией "МПС Софт" или разработчиком проекта), которые работают по данному протоколу, так и произвольный модуль.

При добавлении predetermined модуля в проект, автоматически добавятся и каналы. Количество каналов будет соответствовать возможностям устройства, которое описывает predetermined модуль.

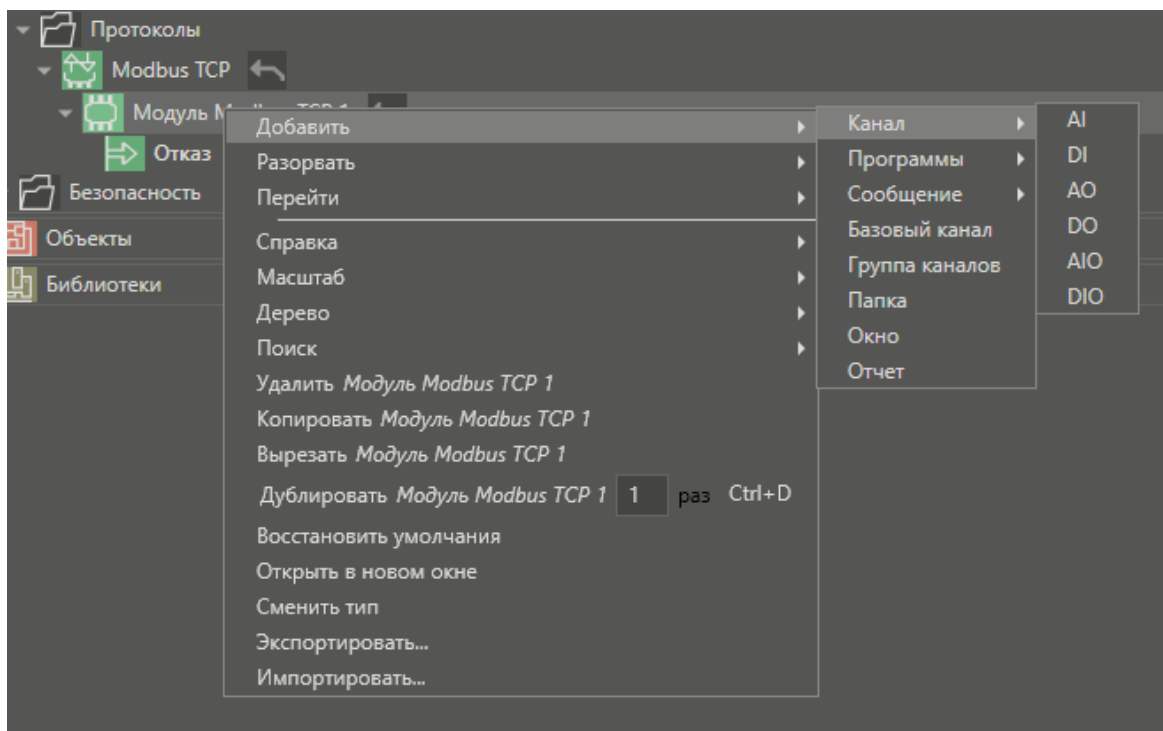
Если добавить элемент Модуль Modbus TCP, каналы необходимо будет добавить вручную.



Все модули ввода-вывода содержат параметр Отказ. Если он принимает состояние TRUE, то это значит, что исполнительная система не может установить связь с данным модулем.

После добавления модуля в проект MasterSCADA 4D необходимо настроить его панель свойств, а затем добавить в модуль требуемое количество каналов.

Можно добавить Базовые каналы, в которых выполнить все настройки самостоятельно, либо каналы predetermined типов, в которых будет достаточно прописать адрес и тип ячейки, в которой хранятся данные в устройстве.

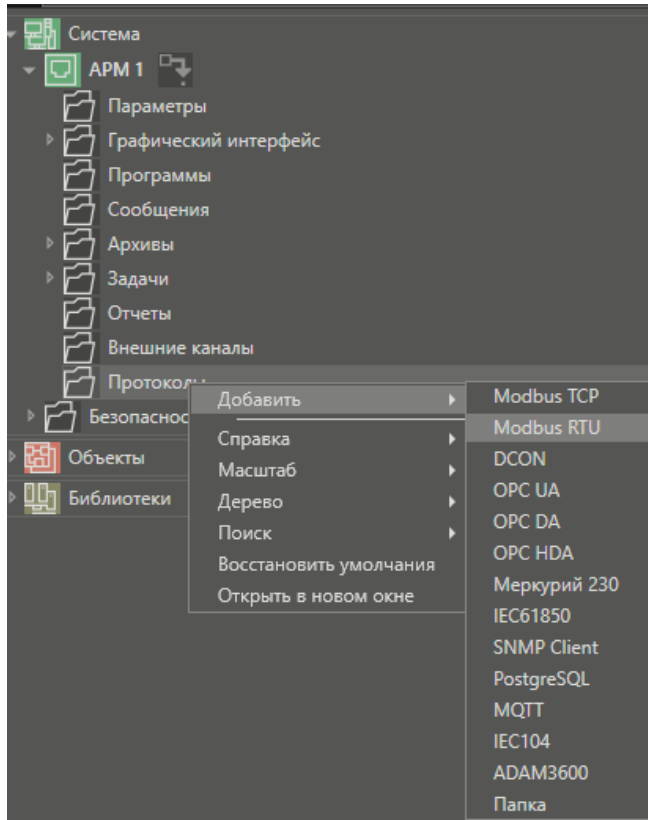


В случае необходимости, в модуле могут быть созданы программы, окна и др. элементы. В этом случае у модуля появится группа Ресурсы, такая же как и у элементов объект, тег, канал.

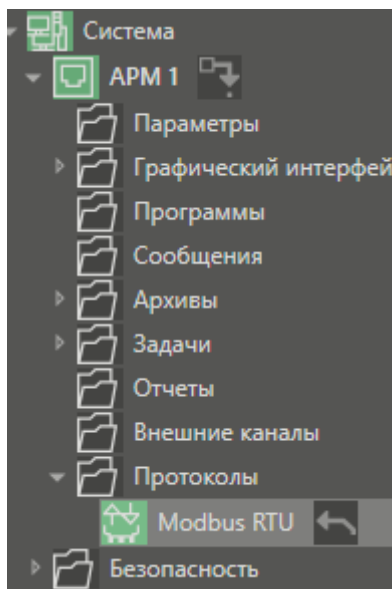
6.2.3.1.1.2. MODBUS RTU

Modbus RTU - это стандартный протокол обмена по MODBUS по последовательному интерфейсу. При обмене узел играет роль ведущего (Master).

Чтобы добавить возможность опроса устройств по данному протоколу, необходимо выбрать соответствующий элемент в контекстном меню узла или в контекстном меню группы Протоколы, либо через контекстную панель узла:



Получим результат в упрощенном дереве:



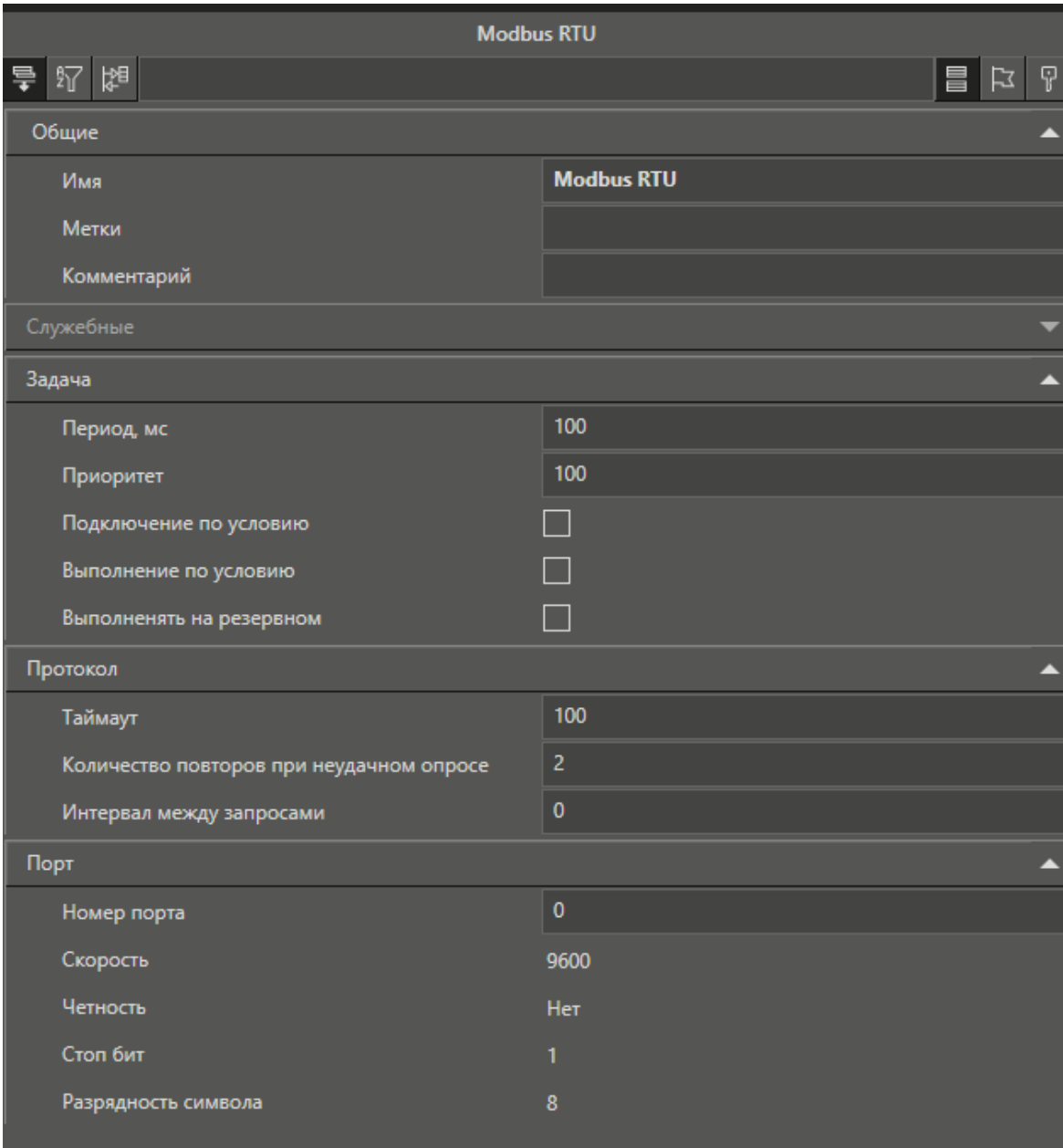
Далее нужно настроить панель свойств. Затем добавить модуль ввода-вывода, который работает по этому протоколу.

При необходимости, в протоколе могут быть созданы программы, окна и др. элементы. В этом случае у протокола появится группа Ресурсы, такая же как и у элементов объект, тег, канал.

СВОЙСТВА ПРОТОКОЛА MODBUS RTU

Настройка элемента производится в панели свойств.

Вид панели свойств:



Modbus RTU	
Общие	
Имя	Modbus RTU
Метки	
Комментарий	
Служебные	
Задача	
Период, мс	100
Приоритет	100
Подключение по условию	<input type="checkbox"/>
Выполнение по условию	<input type="checkbox"/>
Выполнять на резервном	<input type="checkbox"/>
Протокол	
Таймаут	100
Количество повторов при неудачном опросе	2
Интервал между запросами	0
Порт	
Номер порта	0
Скорость	9600
Четность	Нет
Стоп бит	1
Разрядность символа	8

Описание:

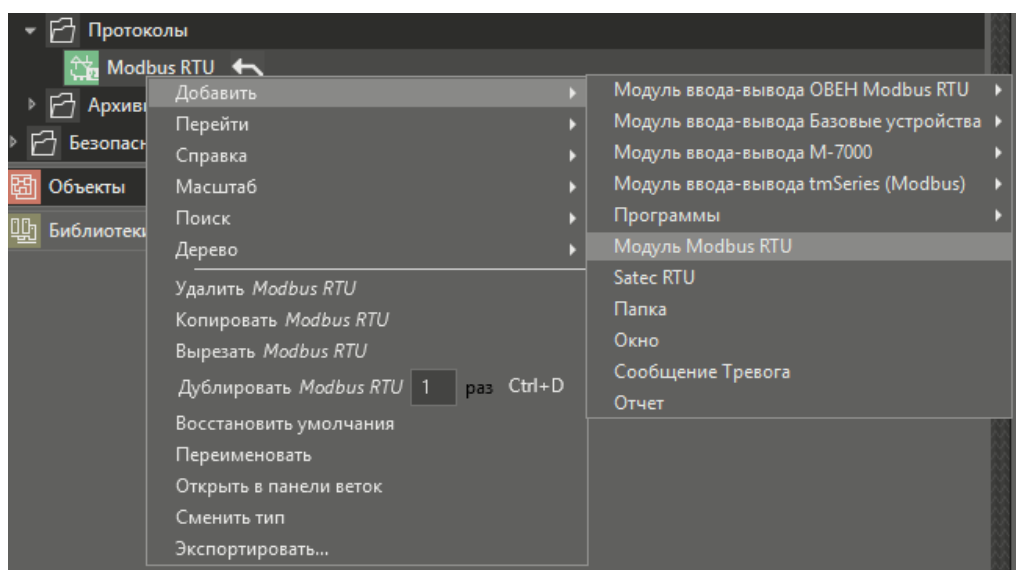
Название	Рекомендации
Категория Общие	Как правило, для данного элемента не настраивается. Подробное описание данной категории приведено в соответствующем разделе.

Название	Рекомендации
Категория Задача	<p>Задается период, с которым будут опрашиваться модули ввода-вывода. Определяется способ работы протокола при резервировании узла. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи у протоколов.</p>
Таймаут	<p>Задается значение таймаута по умолчанию для подключаемых модулей. Это максимальное время ожидания ответа от модуля ввода-вывода (мс). Если ответ от модуля не будет получен в течении заданного интервала времени, то попытка опроса будет считаться неудачной, и зафиксируется ошибка данного запроса для конкретного модуля.</p>
Количество повторов при неудачном опросе	<p>Указывается предельное количество неудачных попыток получить ответ модуля ввода/вывода, произошедших подряд. Если за указанное количество попыток получить ответ от модуля не удалось, то сформируется сигнал TRUE у параметра Отказ у того модуля, с которым не удалось обнаружить связь.</p> <p>Если при обращении к модулю он не ответит ни разу за указанное количество попыток подряд, то следующие запросы к модулю в рамках данного цикла опроса посылаться не будут, сформируется отказ модуля и выставится признак недостоверности по всем входам этого модуля. Если при запросе ответ придет, но с ошибкой, то сформируется отказ модуля и выставится признак недостоверности по входам в данном запросе, но посылка других запросов по этому модулю продолжится.</p>
Интервал между запросами	<p>Интервал (мс) между запросами к модулям ввода-вывода. Если значение не задано (установлено в 0), то используется значение по умолчанию - 5 мс. Данная настройка необходима для того, чтобы устройства отделяли новый запрос от ответа на предыдущий запрос.</p>
Номер порта	<p>Номер COM-порта, к которому подключаются модули ввода-вывода.</p>
Скорость	<p>Скорость, на которой будет производится опрос модулей ввода-вывода (бит/с). Значение должно выбираться в зависимости от</p>

Название	Рекомендации
	возможностей подключаемого устройства. Значение по умолчанию 9600.
Четность	Определяется необходимость контроля четности. Возможные значения: Нет, Чет, Нечет.
Стоп бит	Задается число стоповых битов (1, 1.5, 2), значение по умолчанию – 1;
Разрядность символа	Задается число информационных битов (7 или 8), значение по умолчанию – 8;

МОДУЛЬ MODBUS RTU

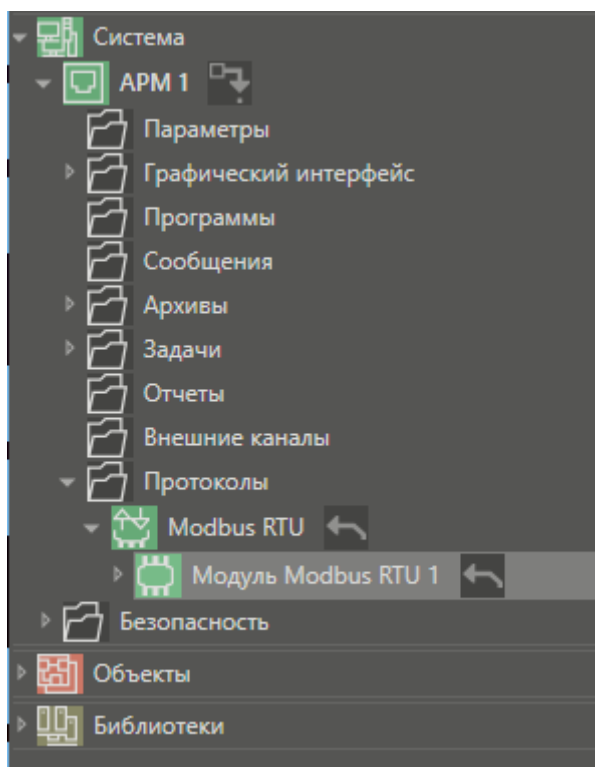
Модуль ввода-вывода добавляется в дерево системы в соответствующий протокол, при помощи контекстного меню протокола:



В проект можно добавить как predetermined модули ввода-вывода (т.е. ранее созданные в библиотеке компанией "МПС Софт" или разработчиком проекта), которые работают по данному протоколу, так и произвольный модуль.

При добавлении predetermined модуля в проект, автоматически добавятся и каналы. Количество каналов будет соответствовать возможностям устройства, которое описывает predetermined модуль.

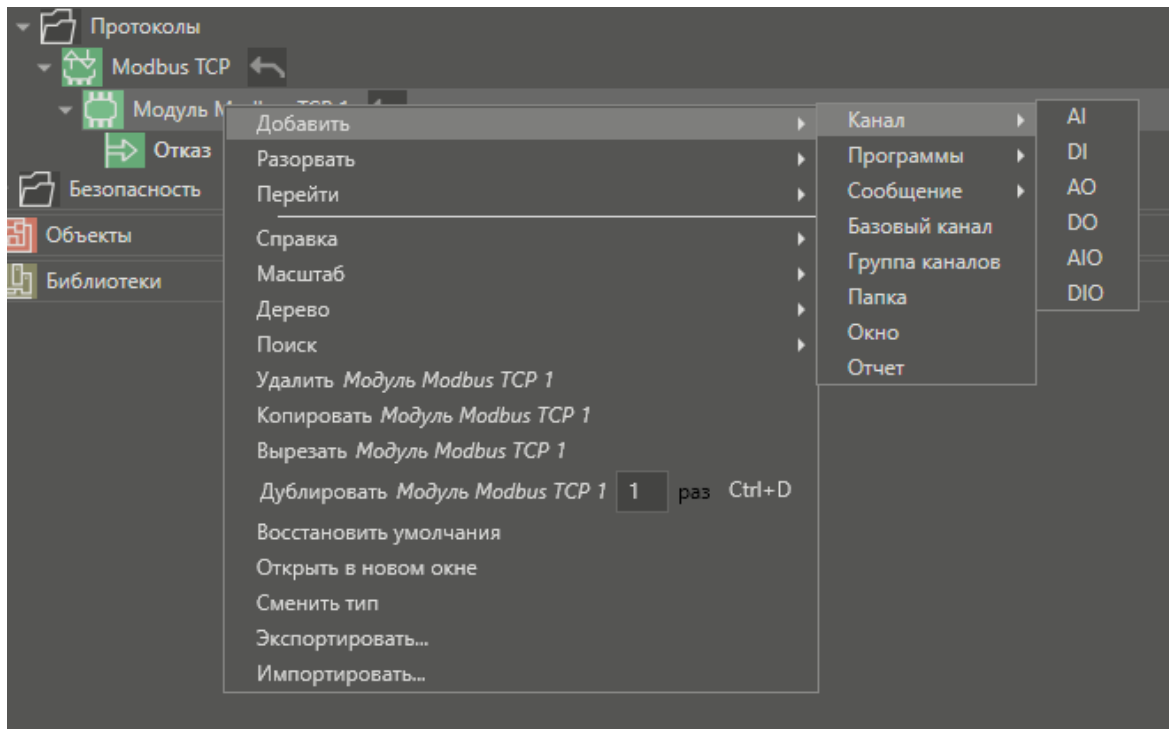
Если добавить элемент Модуль Modbus RTU, то каналы необходимо будет добавить вручную.



Все модули ввода-вывода содержат параметр Отказ. Если он принимает состояние TRUE, то это значит, что исполнительная система не может установить связь с данным модулем.

После добавления модуля в проект MasterSCADA 4D необходимо настроить его панель свойств. Затем необходимо добавить в модуль нужное количество каналов.

Можно добавить Базовые каналы, в которых выполнить все настройки самостоятельно, либо каналы predetermined types, в которых будет достаточно прописать адрес и тип ячейки, в которой хранятся данные в устройстве.



В случае необходимости, в модуле могут быть созданы программы, окна и др. элементы. В этом случае у модуля появится группа Ресурсы, такая же как и у элементов объект, тег, канал.

6.2.3.1.1.3. СВОЙСТВА МОДУЛЕЙ MODBUS TCP И MODBUS RTU

Настройка элементов производится в панели свойств.

Вид панели свойств модуля Modbus TCP:

Модуль Modbus TCP 1

Общие

Имя	Модуль Modbus TCP 1
Метки	
Комментарий	

Служебные

Масштабирование

Включить масштабирование	<input checked="" type="checkbox"/>
SourceMin	
SourceMax	
TargetMin	
TargetMax	

Настройки



IP адрес	10.0.0.1
TCP порт	502
Адрес устройства	1
Таймаут	0
Максимальный интервал неиспользуемых адресов	0
Последовательность байт в вещественных	_1_0_3_2
Максимальный размер массива	125
Последовательность байт в целых	_1_0_3_2
Использовать групповые запросы	<input type="checkbox"/>
Способ записи	По изменению
Последовательность байт в вещественных (8 байт)	_1_0_3_2_5_4_7_6




Настройки каналов

Адрес ячейки	0
Тип значения	Целый
Тип ячейки	Регистр хранения

Вид панели свойств модуля Modbus RTU:

Модуль Modbus RTU 1

Общие ▾

Служебные ▾

Отношения ▾

Задача ▲

Настройки каналов ▾

Масштабирование ▾

Настройки ▲

Подключение по условию

Опрос по условию

Способ записи По умолчанию

Адрес	1
Таймаут	0
Максимальный интервал неиспользуемых адресов	0
Последовательность байт в вещественных	_3_2_1_0
Максимальный размер массива	125
Последовательность байт в целых	_3_2_1_0
Использовать групповые запросы	<input checked="" type="checkbox"/>
Способ записи	По изменению
Последовательность байт в вещественных (8 байт)	_1_0_3_2_5_4_7_6

Описание:

Название	Описание
Категория Общие	Как правило, для данного элемента не настраивается. Подробное описание данной категории приведено в соответствующем разделе.
Категория Задача	Задается способ опроса устройства. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи.

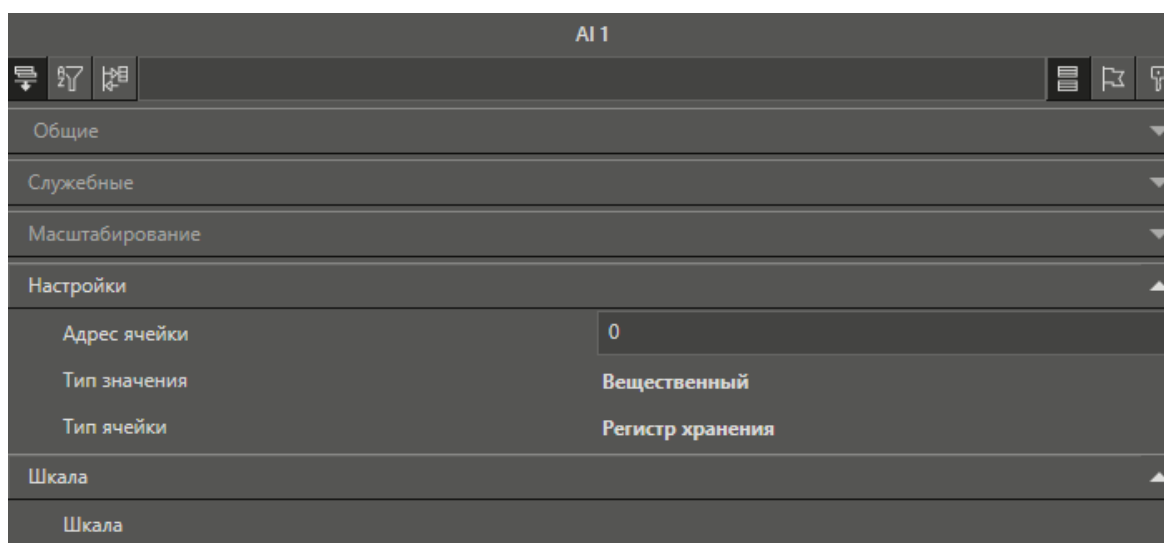
Название	Описание
Категория Масштабирование	<p>Если модуль ввода-вывода не имеет встроенного масштабирования, то при помощи данной категории можно задать соответствие между значениями в модуле и реальными значениями. Подробное описание данной категории приведено в соответствующем разделе.</p>
IP-адрес	<p>Задается IP-адрес модуля (значение IP-адреса определяется документацией на подключаемое устройство). Для резервирования каналов связи можно указать несколько IP-адресов в данном свойстве через запятую.</p> <p>В протоколе Modbus TCP переключение происходит, если два цикла подряд не было ни одного успешного ответа от модуля.</p> <p>В протоколе Modbus RTU over TCP переключение происходит, если два цикла подряд не было ни одного успешного ответа от всех модулей.</p>
TCP-порт	<p>Задается TCP-порт модуля ввода-вывода (номер порта определяется документацией на подключаемое устройство). В начале работы исполнительная система открывает порт. Если нет ответа на запрос, то происходит переоткрытие порта исполнительной системой.</p>
Адрес устройства (Адрес)	<p>Задается адрес устройства (значение адреса определяется документацией на подключаемое устройство).</p>
Таймаут	<p>Максимальное время ожидания ответа от модуля ввода-вывода (мс). Если ответ от модуля не будет получен в течении заданного интервала времени, то попытка опроса будет считаться неудачной, и зафиксирована ошибка данного запроса для конкретного модуля.</p>

Название	Описание
Максимальный интервал неиспользуемых адресов	При превышении этого параметра запрос не попадает в данный групповой запрос.
Последовательность байт в вещественных	Порядок следования байтов в вещественных числах. Данный параметр обеспечивает обмен с устройствами с различным порядком следования байтов в вещественных числах.
Максимальный размер массива	Максимальное количество последовательных адресов, по которым производится чтение данных за один запрос.
Последовательность байт в целых	Порядок следования байтов в целых числах. Данный параметр обеспечивает обмен с устройствами с различным порядком следования байтов в целых числах
Использовать групповые запросы	Данная настройка используется для записи данных в устройство. Если установлено TRUE, то групповые запросы разрешены, если False – запрещены. Чтение данных всегда осуществляется групповыми запросами.
Способ записи	Способ записи значений из MasterSCADA 4D в устройство (по изменению значения или периодически).
Последовательность байт в вещественных (8 байт)	Порядок следования байтов в вещественных числах. Данный параметр обеспечивает обмен с устройствами с различным порядком следования байтов в вещественных числах.
Настройки каналов	Задаются значения по умолчанию для создаваемых каналов. В дальнейшем эти настройки у каналов можно будет изменить.

Важно! Если несколько модулей ввода-вывода Modbus TCP имеют одинаковый IP-адрес и TCP-порт, то устанавливается одно TCP-соединение.

6.2.3.1.1.4. СВОЙСТВА КАНАЛОВ MODBUS

Вид панели свойств:



Описание:

Название	Описание
Категория Общие	Как правило для данного элемента не настраивается. Подробное описание данной категории приведено в соответствующем разделе.
Категория Масштабирование	Если модуль ввода-вывода не имеет встроенного масштабирования, то при помощи данной категории можно задать соответствие между значениями в модуле и реальными значениями. Подробное описание данной категории приведено в соответствующем разделе.
Адрес ячейки	Задается адрес ячейки, из которой требуется прочитать/записать данные.
Тип значения	Задается тип для запроса Modbus. Тип в MasterSCADA 4D определяется типом параметра Вход и может отличаться от типа, указанного в данном поле.
Тип ячейки	Значение в данном поле определяет то, какой функцией будут прочитаны/записаны данные в устройство.

Категория Шкала	В данной категории можно назначить шкалу и определить ее настройки
-----------------	--

Тип ячейки

Тип ячейки задаётся в соответствии со спецификацией Modbus, предоставляется производителем подключаемого устройства, и может быть только одним из четырёх типов, приведённых в таблице ниже. Иногда производители указывают не тип ячейки, а номер функции. Соответствие типа ячеек и номеров функций приведено в таблице ниже.

Тип ячейки	Тип значения ячейки	Номер функции	
		Чтение	Запись
Логическая ячейка (Coils)	Bool	1 (0x01)	5 (0x05); 15 (0x0F)
Дискретный вход (Discrete Inputs)	Bool	2 (0x02)	
Регистр хранения (Holding Registers)	Word	3 (0x03)	6 (0x06); 16 (0x10)
Входной регистр (Input Registers)	Word	4 (0x04)	

Например, если производитель указал, что значение должно читаться функцией 1, то необходимо указать тип ячейки Логическая ячейка, функцией 2 - Дискретный вход и т.д.

Запись может осуществляться функциями одиночного (5 или 6 - в зависимости от типа ячейки) или группового (15 или 16) доступа (в том случае когда нужно изменить значения нескольких ячеек, адреса которых следуют подряд).

Некоторые устройства не поддерживают функции записи одиночных ячеек. Для использования групповой записи необходимо в настройках устройства (модуля) на панели свойств установить флаг Использовать групповые запросы. Тогда запись даже одиночных ячеек будет осуществляться функцией 15 для логической ячейки и функцией 16 для регистра хранения.

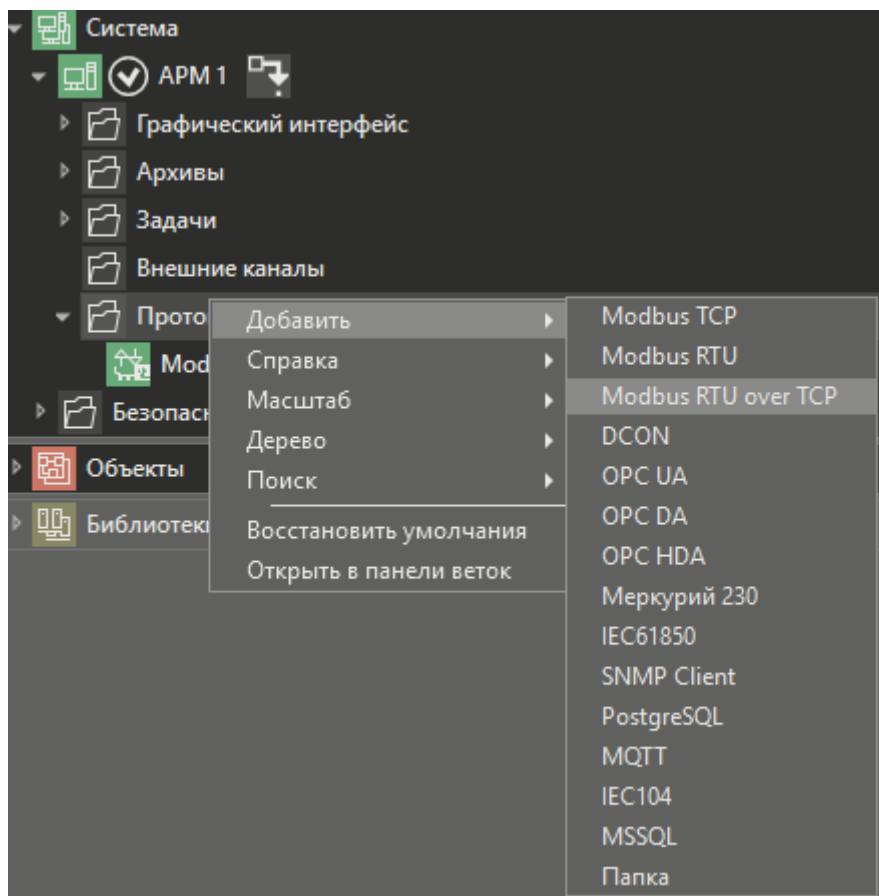
Для чтения всегда используются групповые запросы. Максимальное количество записей в запросе определяется настройкой модуля Максимальный размер массива

Если настройки сделаны верно, то при подключении к устройству и загрузке в него конфигурации должна появиться связь с модулем.

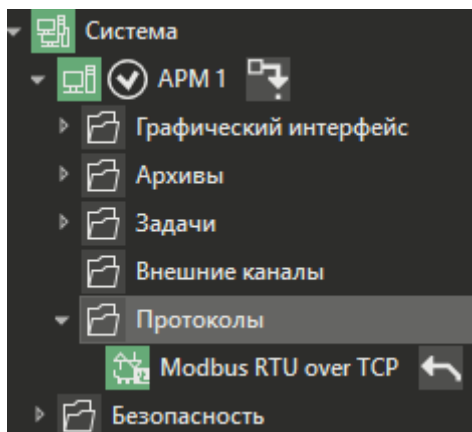
6.2.3.1.1.5. MODBUS RTU OVER TCP

Modbus RTU over TCP предназначен для опроса RTU модулей, подключенных через преобразователи протоколов.

Для того, чтобы добавить возможность опроса устройств по данному протоколу, необходимо выбрать соответствующий элемент в контекстном меню узла или в контекстном меню группы Протоколы, либо через контекстную панель узла :



Получим результат в упрощенном дереве:



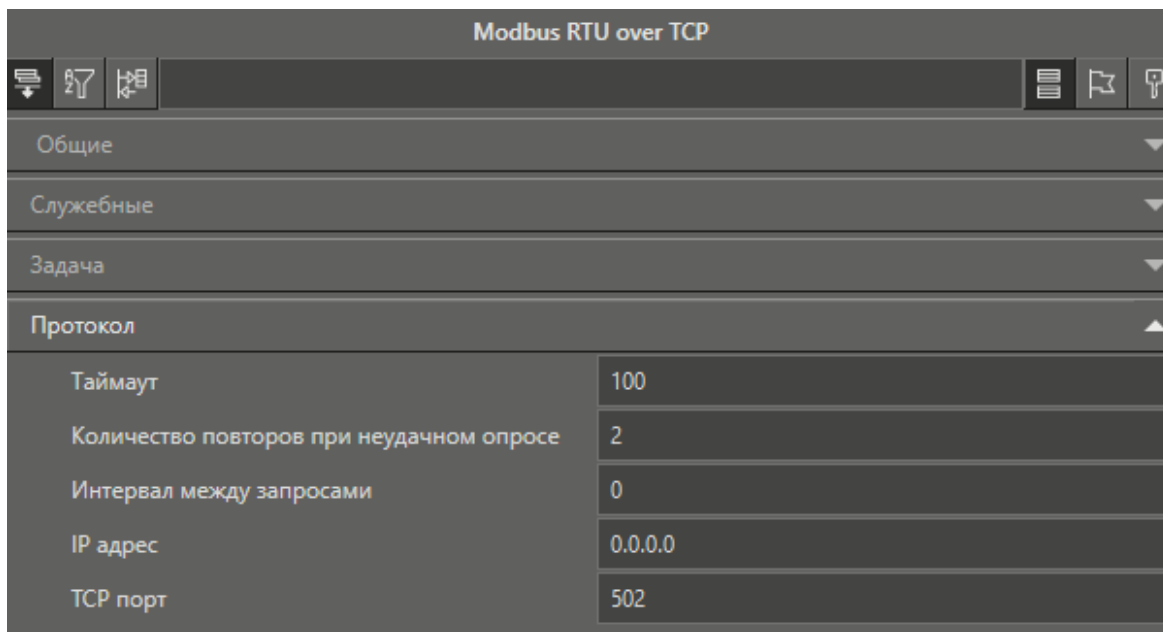
Далее необходимо настроить панель свойств. Затем добавить модуль ввода-вывода, который работает по этому протоколу.

При необходимости в протоколе могут быть созданы программы, окна и др. элементы. В этом случае у протокола появится группа Ресурсы, такая же как и у элементов объект, тег, канал.

СВОЙСТВА ПРОТОКОЛА MODBUS RTU OVER TCP

Настройка элемента производится в панели свойств.

Вид панели свойств:



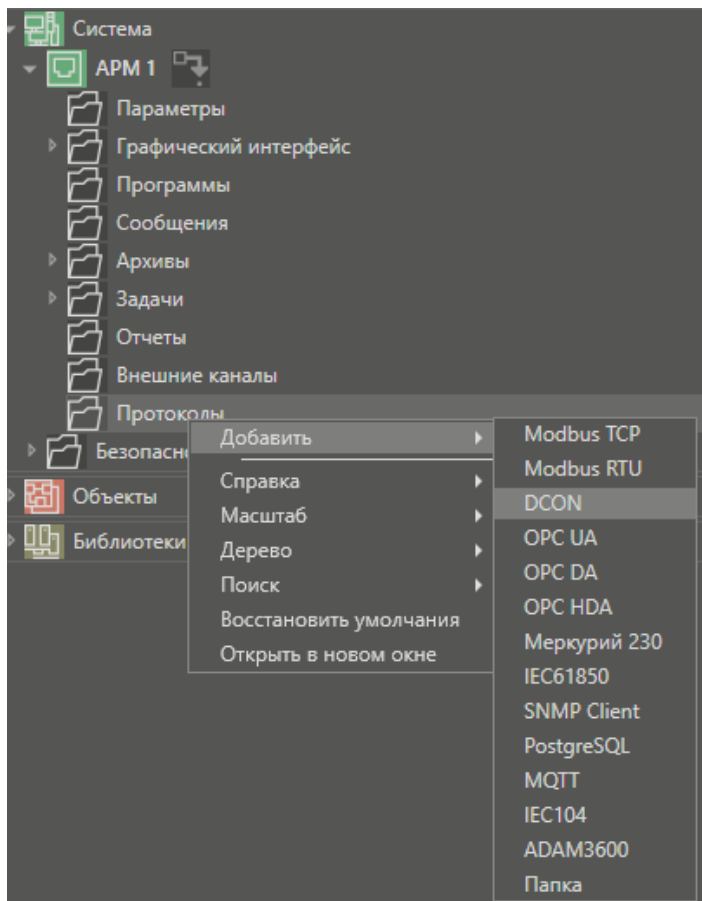
Описание основных свойств:

Название	Рекомендации
Категория Общие	Как правило, для данного элемента не настраивается. Подробное описание данной категории приведено в соответствующем разделе
Категория Задача	Задается период, с которым будут опрашиваться модули ввода-вывода. Определяется способ работы протокола при резервировании узла. Подробное описание категории приведено в разделе, описывающем настройку категории Задача
Таймаут	Задается значение таймаута по умолчанию для подключаемых модулей. Это максимальное время ожидания ответа от модуля ввода-вывода (мс). Если ответ не будет получен от модуля в течение заданного интервала времени, то попытка опроса будет считаться неудачной, и зафиксируется ошибка данного запроса для конкретного модуля.
Количество повторов при неудачном опросе	<p>Указывается предельное количество неудачных попыток получить ответ модуля ввода-вывода, произошедших подряд. Если за указанное количество попыток получить ответ от модуля не удалось, то сформируется сигнал TRUE у параметра Отказ у того модуля, с которым не удалось обнаружить связь.</p> <p>Если при обращении к модулю он не ответит ни разу за указанное количество попыток подряд, то следующие запросы к модулю в рамках данного цикла опроса посылаться не будут, сформируется отказ модуля и выставится признак недостоверности по всем входам этого модуля. Если при запросе ответ придет, но с ошибкой, то сформируется отказ модуля и выставится признак недостоверности по входам в данном запросе, но посылка других запросов по этому модулю продолжится.</p>
Интервал между запросами	Определяет интервал между запросами. Если этот интервал не задан, то MasterSCADA 4D рассчитает его сама в зависимости от скорости.
Ip-адрес	Задается IP-адрес модуля (значения IP-адреса определяется документацией на подключаемое устройство).
TCP порт	Задается TCP-порт модуля ввода-вывода (номер порта определяется документацией на подключаемое устройство)

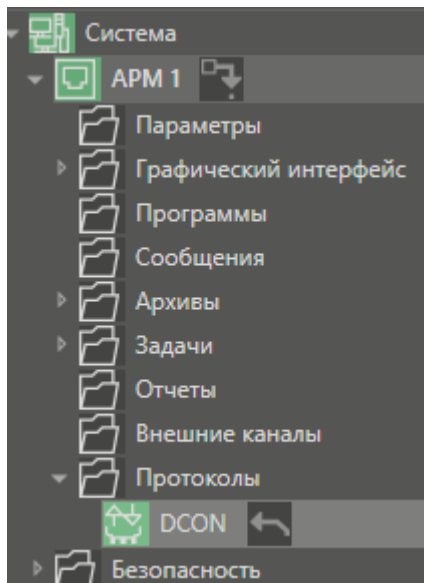
6.2.3.1.2. DCON

DCON - это стандартный протокол обмена по последовательному интерфейсу. При обмене узел играет роль ведущего (Master).

Для того чтобы добавить возможность опроса устройств по данному протоколу, необходимо выбрать соответствующий элемент в контекстном меню узла, или в контекстном меню группы Протоколы, либо через контекстную панель узла :



Получим результат в упрощенном дереве:



Далее нужно настроить панель свойств. Затем добавить модуль ввода-вывода, который работает по этому протоколу.

В случае необходимости, в протоколе могут быть созданы программы, окна и др. элементы. В этом случае у протокола появится группа Ресурсы, такая же как и у элементов объект, тег, канал.

6.2.3.1.2.1. СВОЙСТВА ПРОТОКОЛА DCON

Настройка элемента производится в панели свойств.

Вид панели свойств:

DCON

☰
🔍
🔗

☰
🔗
🔑

Общие ▲

Имя	DCON
Метки	
Комментарий	

Служебные ▼

Задача ▲

Период, мс	100
Приоритет	100
Подключение по условию	<input type="checkbox"/>
Выполнение по условию	<input type="checkbox"/>
Выполнять на резервном	<input type="checkbox"/>

Протокол ▲

Таймаут	100
Контрольная сумма	<input type="checkbox"/>

Порт ▲

Номер порта	0
Скорость	9600
Четность	Нет
Стоп бит	1
Разрядность символа	8

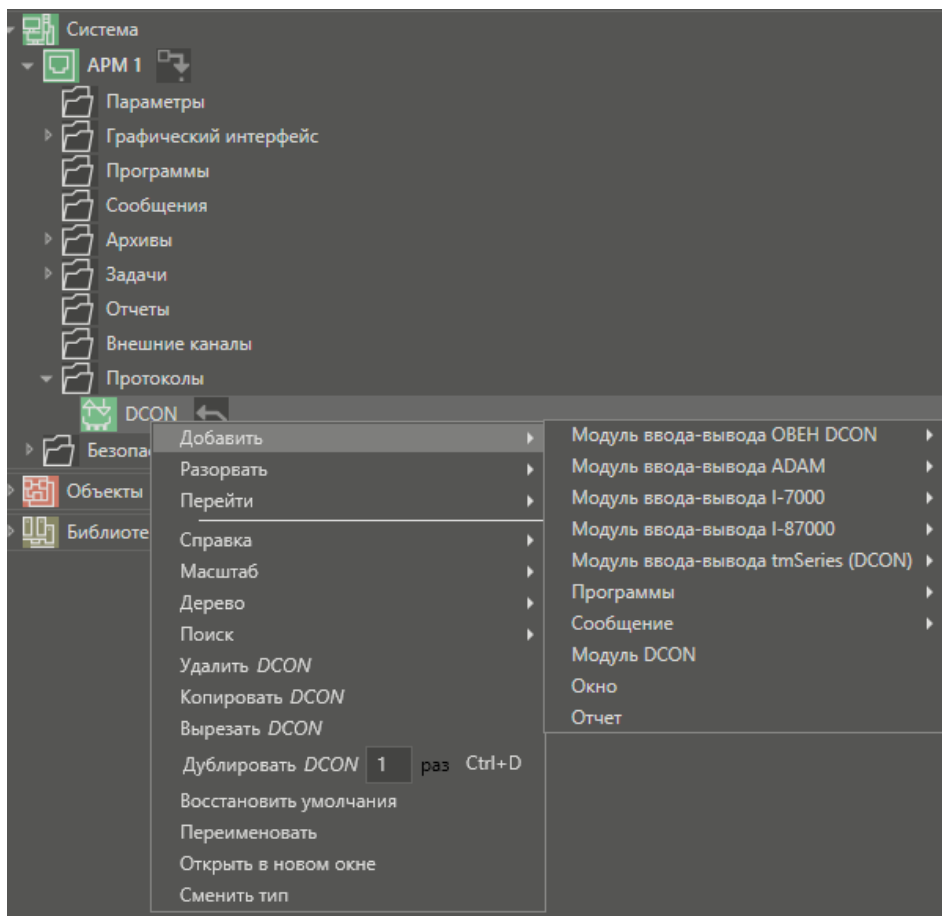
Описание:

Название	Рекомендации
Категория Общие	Как правило, для данного элемента не настраивается. Подробное описание данной категории приведено в соответствующем разделе
Категория Задача	Задается период, с которым будут опрашиваться модули ввода-вывода. Определяется способ работы протокола при резервировании узла. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи у протоколов

Название	Рекомендации
Таймаут	Задается значение таймаута по умолчанию для подключаемых модулей. Это максимальное время ожидания ответа от модуля ввода-вывода (мс). Если ответ не будет получен от модуля в течение заданного интервала времени, то попытка опроса будет считаться неудачной, и зафиксируется ошибка данного запроса для конкретного модуля. После двух таких неудачных попыток сформируется сигнал TRUE у параметра Отказ у того модуля, от которого не удалось получить ответ.
Контрольная сумма	Если флаг установлен, то контрольная сумма вычисляется, если флаг не установлен, то не вычисляется.
Номер порта	Номер COM-порта, к которому подключаются модули ввода-вывода
Скорость	Скорость, на которой будет производиться опрос модулей ввода-вывода (бит/с). Эта скорость должна выбираться в зависимости от возможностей подключаемого устройства. Значение скорости по умолчанию 9600
Четность	Определяется необходимость контроля четности. Возможные значения: Нет, Чет, Нечет
Стоп бит	Задается число стоповых битов (1, 1.5, 2), значение по умолчанию – 1;
Разрядность символа	Задается число информационных битов (7 или 8), значение по умолчанию – 8;

6.2.3.1.2.2. МОДУЛЬ DCON

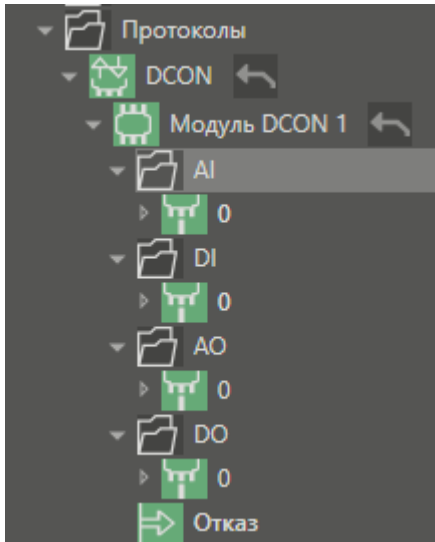
Модуль ввода-вывода добавляется в дерево системы в соответствующий протокол, при помощи контекстного меню протокола:



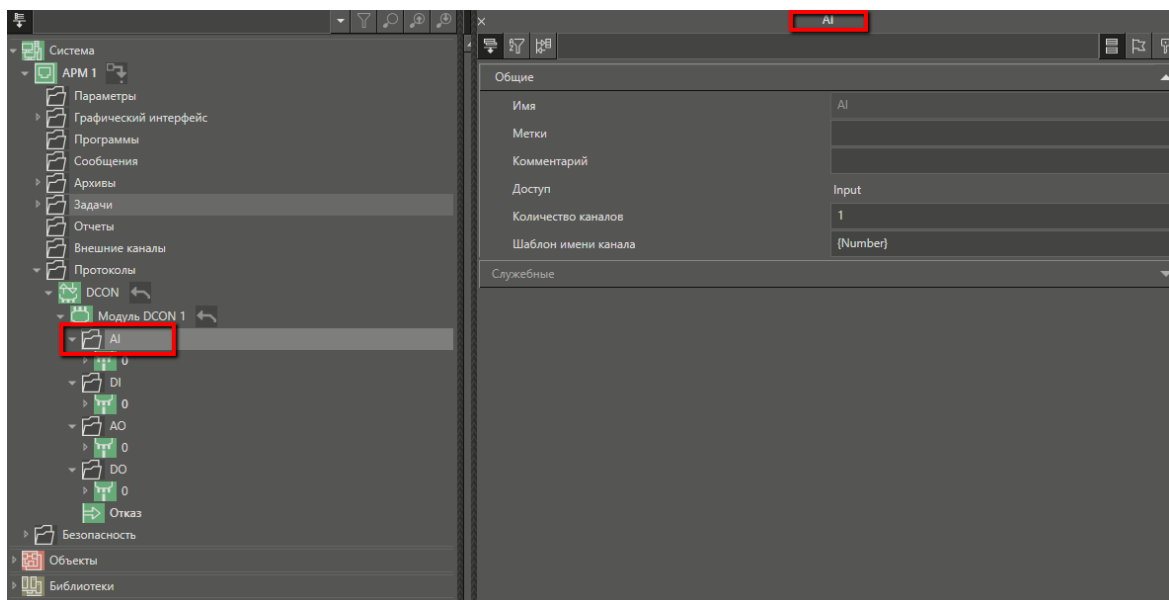
В проект можно добавить как predeterminedные модули ввода-вывода (т.е. ранее созданные в библиотеке компанией "МПС Софт" или разработчиком проекта), работающие по данному протоколу, так и произвольный модуль.

При добавлении predeterminedного модуля в проект, автоматически добавятся и каналы. Количество каналов будет соответствовать возможностям устройства, которое описывает predeterminedный модуль.

Если добавить элемент Модуль DCON, то по умолчанию добавится модуль, у которого будет создано 4 группы для добавления каналов predeterminedных типов: AI, DI, AO, DO.



В панели свойств каждой из групп можно указать какое количество каналов будет использоваться:



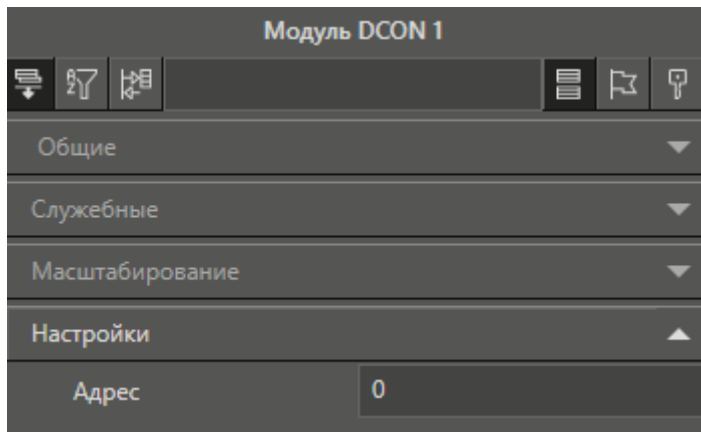
Все модули ввода-вывода содержат параметр Отказ. Если он принимает состояние TRUE, то это означает, что исполнительная система не может установить связь с данным модулем.

После добавления модуля в проект MasterSCADA 4D требуется настроить его панель свойств.

В случае необходимости, в протоколе могут быть созданы программы, окна и др. элементы. В этом случае у протокола появится группа Ресурсы, такая же как и у элементов объекта, тег, канал.

СВОЙСТВА МОДУЛЯ DCON

Вид панели свойств:



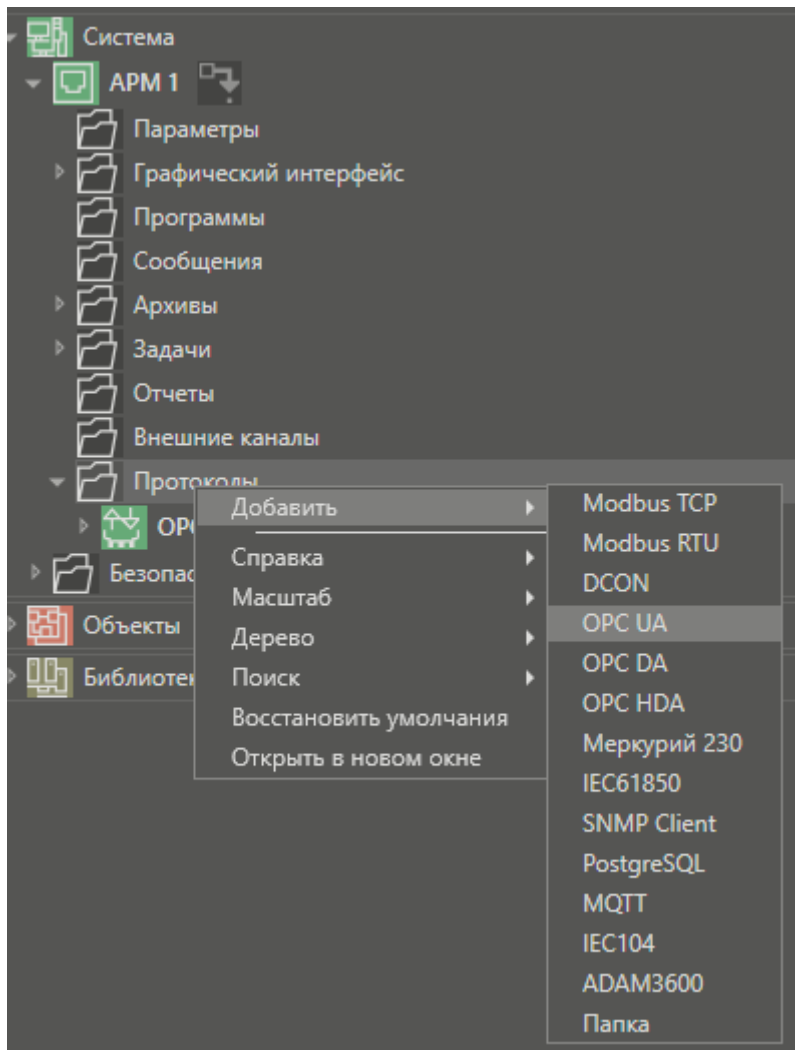
Описание:

Название	Описание
Категория Общие	Как правило, для данного элемента не настраивается. Подробное описание данной категории приведено в соответствующем разделе.
Категория Задача	Задается способ опроса устройства. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи.
Категория Масштабирование	Если модуль ввода-вывода не имеет встроенного масштабирования, то при помощи данной категории можно задать соответствие между значениями в модуле и реальными значениями. Подробное описание данной категории приведено в соответствующем разделе.
Адрес	Задается адрес устройства (значения адреса определяется документацией на подключаемое устройство).

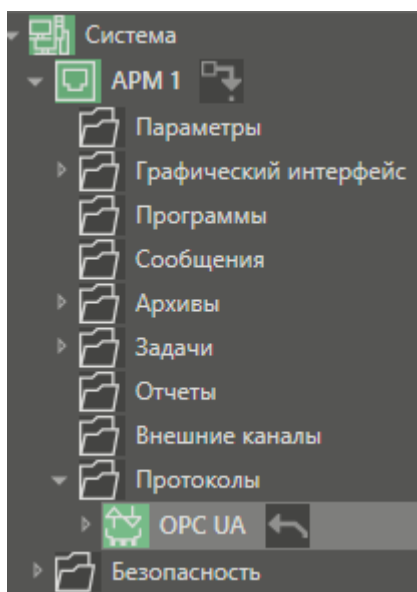
6.2.3.1.3. OPC UA

MasterSCADA 4D поддерживает платформонезависимый стандарт OPC UA.

Для того чтобы MasterSCADA 4D выступала в роли клиента, необходимо в группу узла Протоколы добавить соответствующий протокол:

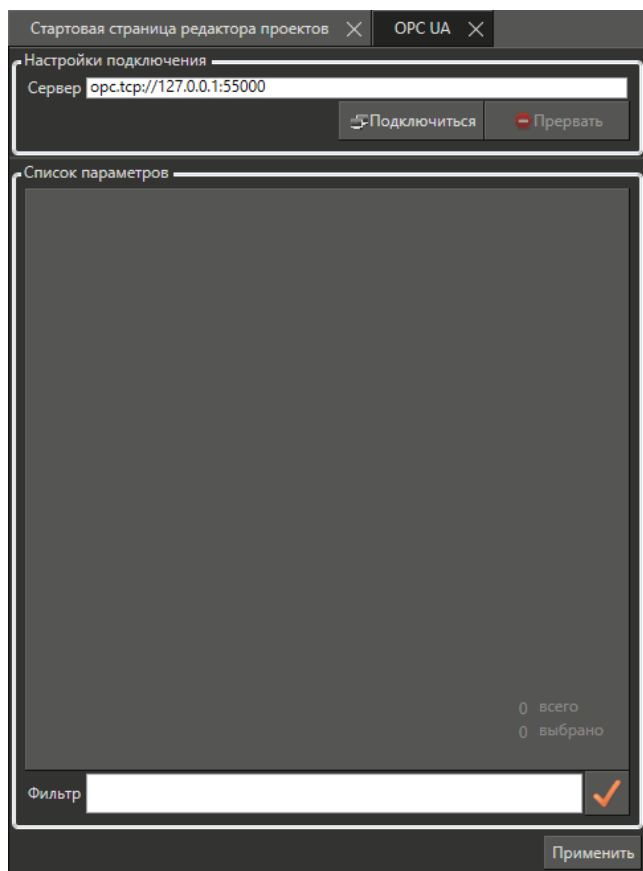


Получим результат в упрощенном дереве:



Затем необходимо выполнить подключение к OPC UA серверу для добавления каналов.

Для этого нужно дважды нажать левую кнопку мыши по элементу в дереве. При этом откроется вкладка, в которой можно выполнить подключение к OPC UA серверу:



В строке Сервер необходимо задать IP-адрес и TCP/IP порт сервера в виде: `opc.tcp://[IP-адрес][номер порта]`. Пример записи показан в значении по умолчанию: `opc.tcp://127.0.0.1:55000`.

Важно! Перед началом подключения к серверу необходимо убедиться, что сервер запущен в режим исполнения.

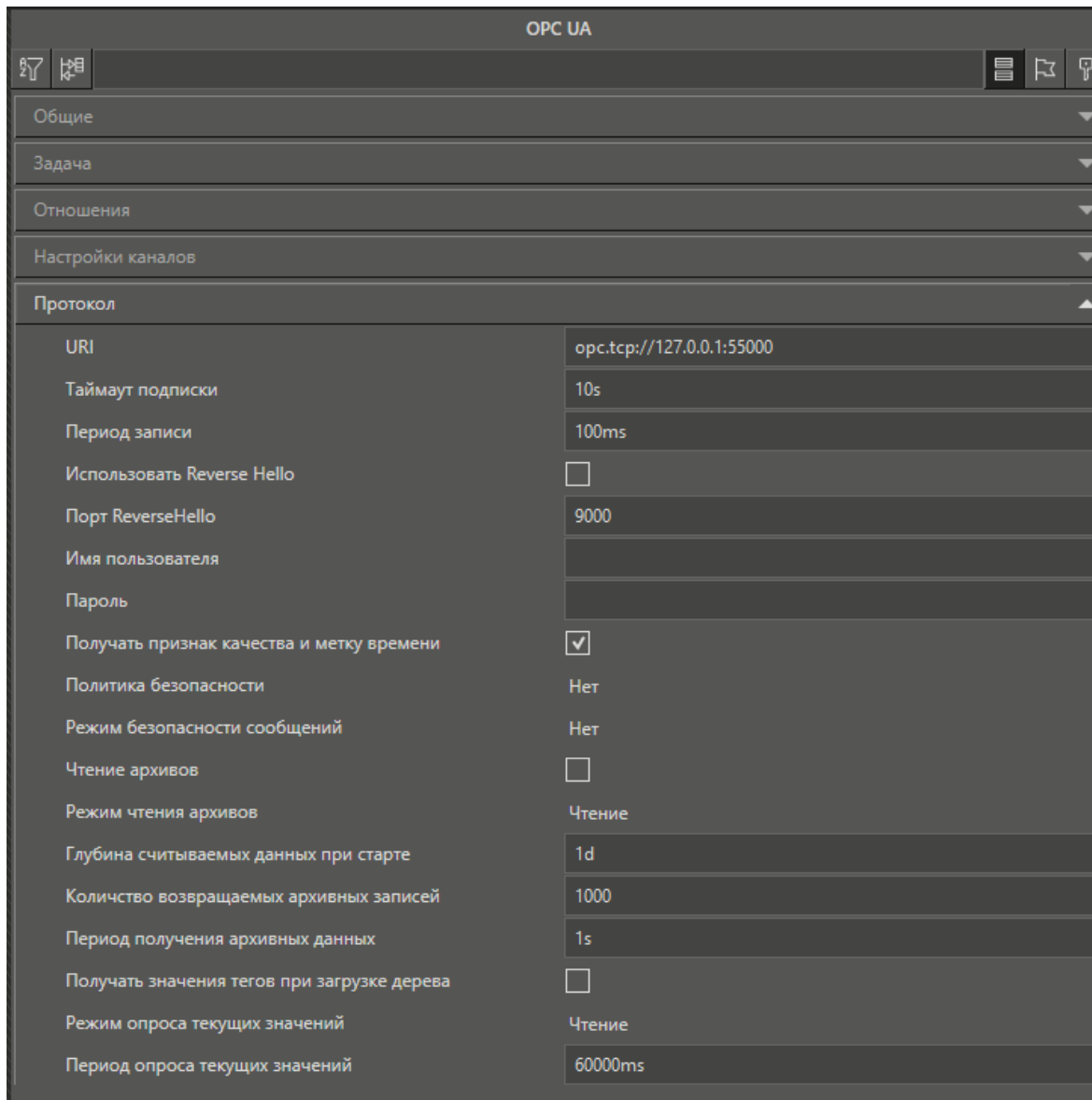
Затем следует нажать кнопку Подключиться. Для отмены подключения необходимо нажать кнопку Прервать. Если попытка подключения окажется удачной, то в группе Список параметров отобразятся параметры подключенного сервера в виде дерева, в котором следует выбрать те элементы, с которыми требуется работать в проекте MasterSCADA 4D. Далее необходимо нажать кнопку Применить. После этого в дерево системы добавятся каналы predetermined типов.

После подключения к серверу необходимо настроить панель свойств OPC UA сервера.

При работе по протоколу OPC UA создается, если его не было ранее, сертификат в рабочей папке исполнительной системы *ClcCertificateStore*

6.2.3.1.3.1. СВОЙСТВА OPC UA

Вид панели свойств:



Описание:

Название	Рекомендации
Категория Общие	Как правило, для данного элемента не настраивается. Подробное описание данной категории приведено в соответствующем разделе.
Категория Задача	Задается период, с которым MasterSCADA 4D подписывается получать измененные значения из сервера. Определяется способ работы протокола при резервировании узла. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи у протоколов.

Название	Рекомендации
Категория Протокол	
URI	Задается IP-адрес и TCP/IP порт сервера, в виде: opc.tcp://[IP-адрес]:[номер порта]. Данное поле заполняется автоматически после добавления сервера. В случае необходимости его можно изменить.
Таймаут подписки	Если в течении этого времени (мс) не приходило сообщений от сервера, то MasterSCADA 4D отключится от сервера и попытается подключиться к нему заново. Тип значения свойства - Time.
Период записи	Запись в OPC UA сервер происходит по изменению. Данное поле определяет период, в течении которого происходит накопление данных на запись, после чего происходит их запись в OPC UA сервер. Тип значения свойства - Time.
Использовать Reverse Hello	<p>Включает режим инициативного подключения к данному протоколу со стороны сервера. Используется в случае если у устройства, где установлена исполнительная система нет внешнего IP-адреса.</p> <p>Если включен этот режим, то у клиента может не получиться загрузить конфигурацию тегов, поэтому рекомендуется получить список тегов с выключенной настройкой, а затем проводить опрос с уже установленным флагом.</p>
Порт Reverse Hello	Указывается номер порта, на котором будет ожидать подключение со стороны сервера. Если в проекте несколько протоколов OPC UA, у каждого протокола должен быть настроен уникальный номер порта, который соответствует номеру порта в настройке Адрес клиента OPC UA того сервера, к которому подключается данный протокол.
Имя пользователя	Задается имя пользователя в случае неанонимного подключения. Если данное поле пустое, то MasterSCADA 4D считает, что соединение анонимное.

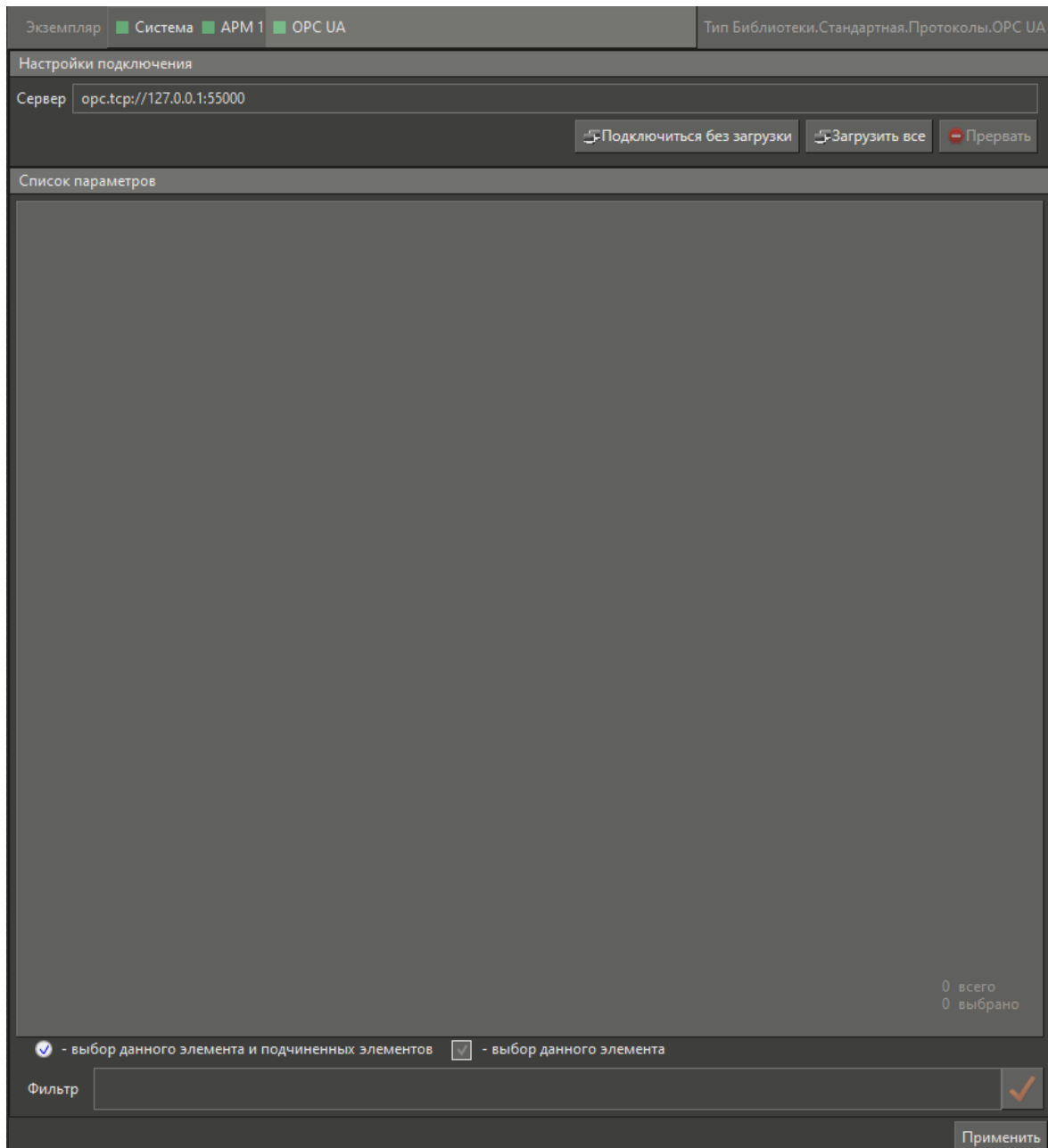
Название	Рекомендации
Пароль	Задается пароль в случае неанонимного подключения.
Получать признак качества и метку времени	<p>Определяет тип данных для каналов: SYSTEM_*_PARAM (включает метку и качество), или атомарные. Если параметр атомарного типа архивируется, то в режиме чтения архивов из OPC UA сервера в архив будут записываться метки времени и признаки качества, получаемые от сервера, но текущие значения метки времени и признака качества будут недоступны.</p> <p>Если данные приходят с плохим признаком качества, то эти данные не будут записываться в сервере.</p>
Политика безопасности	<p>В случае если используется шифрование (Свойство Режим безопасности сообщений=SingAndEncrypt), в данном поле выбирается тип шифрования. Возможны варианты:</p> <ul style="list-style-type: none"> • Нет • Basic128Rsa15 • Basic256 • Basic256Sha256
Режим безопасности сообщений	<p>Выбирается режим работы безопасности.</p> <p>Нет - данные передаются в открытом виде</p> <p>Подпись - данные подписываются</p> <p>Подпись и шифрование - данные подписываются и шифруются.</p> <p>Тип шифрования определяется настройкой Политика безопасности</p>
Чтение архивов	Если флаг установлен, то MasterSCADA 4D будет получать архивные данные от сервера.
Режим чтения архивов	<p>Настройка определяет метод получения архива переменных:</p> <ul style="list-style-type: none"> • Чтение - Однократно в Период получения архивных данных происходит запрос чтения по каждому параметру за интервал времени от метки времени последнего полученного значения по данному параметру по текущее время. В том случае если

Название	Рекомендации
	<p>данных еще не было получено, запрос производится за интервал времени от момента, предшествующего текущему времени на значение параметра Глубина считываемых данных при старте, по текущее время</p> <ul style="list-style-type: none"> • Подписка - при данном методе данные получаются методом, аналогичным получению текущих данных, но при этом указывается количество считываемых за один цикл архивных записей, определяемых параметром Количество возвращаемых архивных записей. Т.е. за один запрос по подписке может прийти несколько значений по одному параметру с различными отметками времени • Чтение и подписка - при этом режиме архивные данные, ранее накопленные в приборе, будут получены методом Чтение, а дальнейшая работа будет производиться методом Подписка
Период получения архивных данных	Период, с которым OPC-клиент обращается к OPC-серверу для получения архивных данных в режиме работы Чтение.
Глубина считываемых данных при старте	Если в архиве MasterSCADA 4D не было записей по тому или иному параметру, то она попытается считать архивные данные за указанное количество дней. Если в архиве уже есть записи по архивируемым параметрам, то данные будут получены с момента последней записи в архиве по текущий момент. Т.е. при повторных подключениях эта настройка не учитывается. Тип значения свойства - Time.
Количество возвращаемых архивных записей	Максимальное число записей по одному параметру, передаваемое серверов в режиме чтения архивных записей Подписка за один цикл опроса.
Получать значения тегов при загрузке дерева	Определяет будут ли прочитаны не только типы тегов, но и их значения при подключении и добавлении тегов в диалог подключения. Чаще всего используется, если сервер по параметрам возвращает неконкретный тип (например KontarOpc возвращает DataTypes.BaseDataType (24) как для дискретных, так и для аналоговых). Если установить флаг, при подключении к серверу среда разработки зачитывает не только типы, но и

Название	Рекомендации
	сами значения, и тип тега может установить по типу полученного значения.
Режим опроса текущих значений	<p>Определяет способ опроса OPC-сервера. Возможные варианты:</p> <ul style="list-style-type: none"> • Чтение - в этом случае с сервера запрашиваются значения всех каналов, с периодом, указанным в поле Период опроса текущих значений. • Подписка - в этом случае при старте OPC-сервер будет подписываться на каналы, решение об отправке данных будет принимать OPC-сервер, будут отправляться только те данные, которые изменились, с периодом указанным в категории Задача OPC-сервера. • Чтение и подписка - в этом случае будут использоваться оба режима опроса: данные будут поступать только те, которые изменились с периодом указанным в категории Задача OPC-сервера и будут запрашиваться все данные с периодом, указанным в поле Период опроса текущих значений.
Период опроса текущих значений	Определяет период опроса данных для режима Чтение.
Категория Настройки каналов	
NodeId	NodeId - это идентификатор канала OPC-сервера. В данном поле можно задать значение по умолчанию, но, как правило, оно не задается. У отдельных каналов данное поле заполняется автоматически

6.2.3.1.3.2. ДИАЛОГ ПОДКЛЮЧЕНИЯ К СЕРВЕРУ OPC UA

Данный диалог открывается во вкладке документов и редакторов по команде Открыть контекстного меню протокола OPC UA.

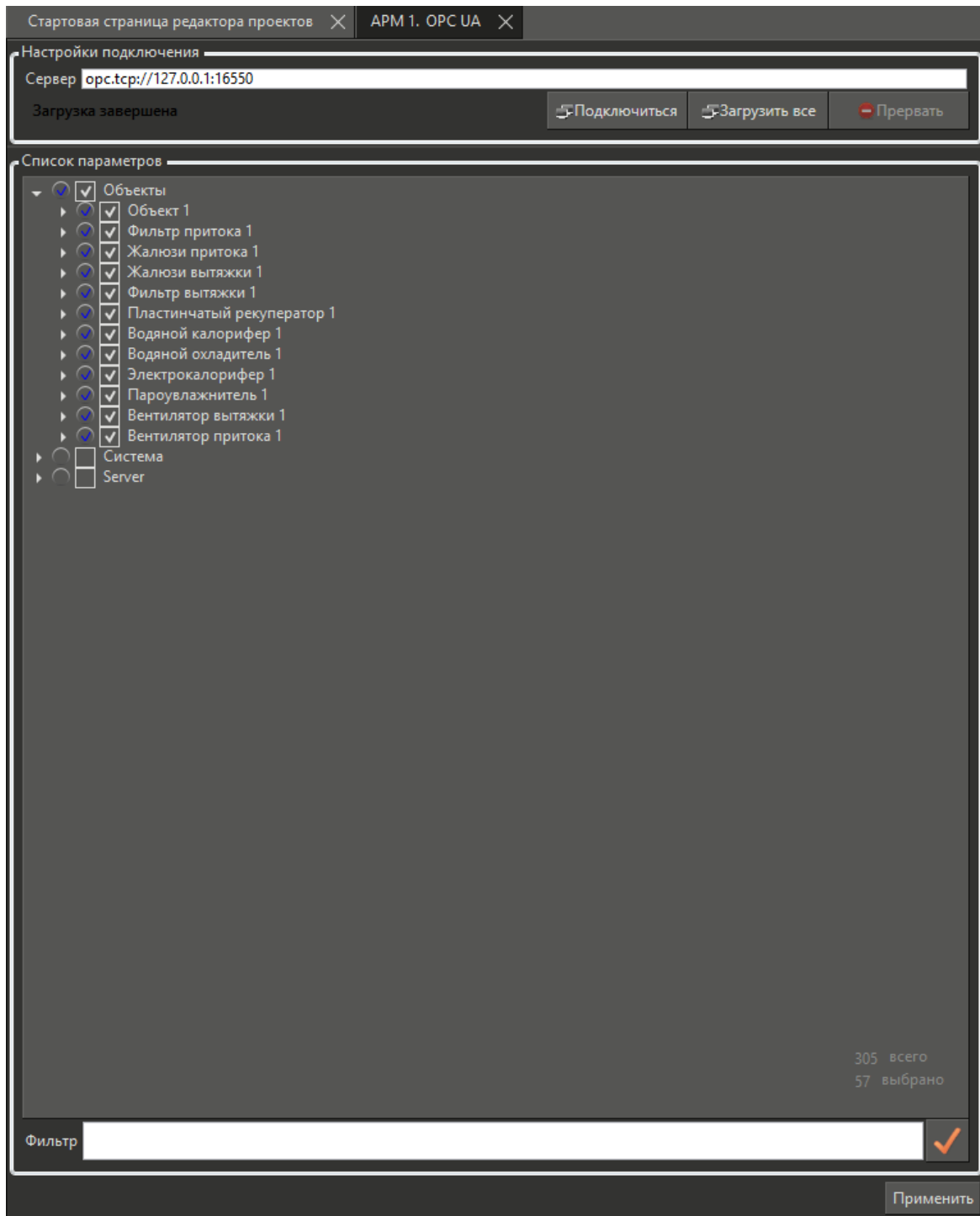


Группа Настройки подключения:


Название	Описание
Сервер	Задается IP-адрес и TCP/IP порт сервера, в виде: opc.tcp://[IP-адрес][номер порта]. Пример записи показан в значении по умолчанию: opc.tcp://127.0.0.1:55000.
Подключиться без загрузки	Выполняет подключение к ранее запущенному в режим исполнения серверу. В группе диалога Список параметров видны только корневые группы. При раскрытии группы грузятся ее дочерние элементы. При этом идет проверка, существуют ли уже такие



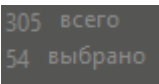

Название	Описание
	группы, каналы в проекте. Если существуют, то у них устанавливается флаг, сигнализирующий о том, что они добавлены в проект. При установке круглой галочки сначала загружаются все дочерние группы этого элемента, а затем все загруженные элементы выбираются для вставки.
Загрузить все	Выполняет подключения к ранее запущенному в режим исполнения серверу. В группе диалога Список параметров видны все возможные для добавления группы и каналы. Важно! Если сервер имеет большую конфигурацию, то этот процесс может занять длительное время.
Прервать	Прерывает подключение

После выполнения команд Подключиться или Загрузить все в группе Список параметров отобразятся доступные и добавленные каналы:



Инструменты диалога:

Название	Описание
	Раскрывает/закрывает группу элементов.

Название	Описание
	Позволяет выделить/снять выделение со всей группы вложенных элементов. Если флаг синий, то это означает, что добавлены все дочерние элементы, если флаг серый, то это означает, что в проект были добавлены не все элементы группы, а только их часть.
	Позволяет выделить/ снять выделение с отдельного элемента.
	Выдает информацию о том, сколько параметров может быть добавлено в проект всего, и сколько уже было выбрано.
Фильтр	Фильтр отображения конфигурации OPC UA-сервера, к которому выполнено подключение. В строке Фильтр задается подстрока поиска. Для применения фильтра требуется нажать кнопку 
Применить	Добавляет в дерево системы каналы, выбранные в группе Список параметров

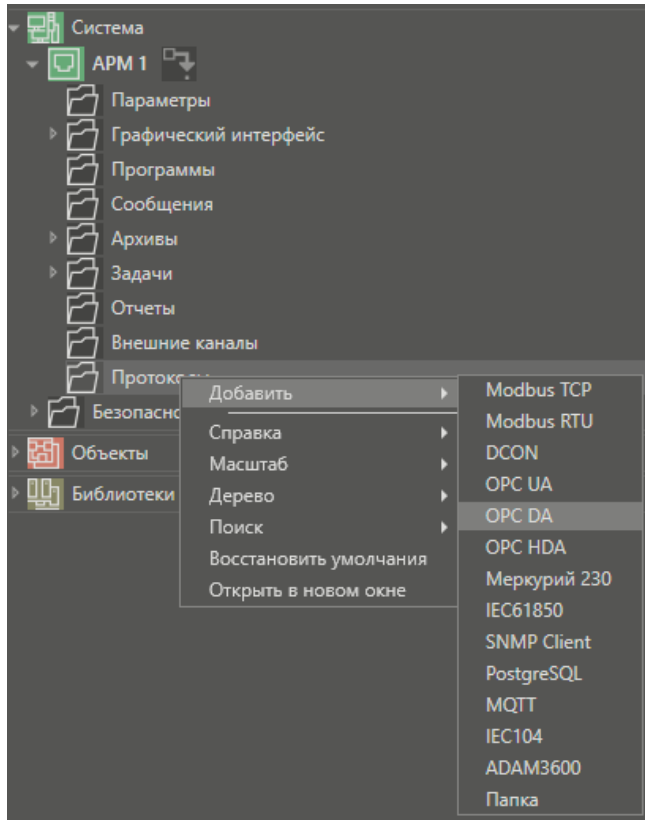
Важно! Изменения вступят в силу только после нажатия на кнопку Применить

6.2.3.1.4. OPC DA

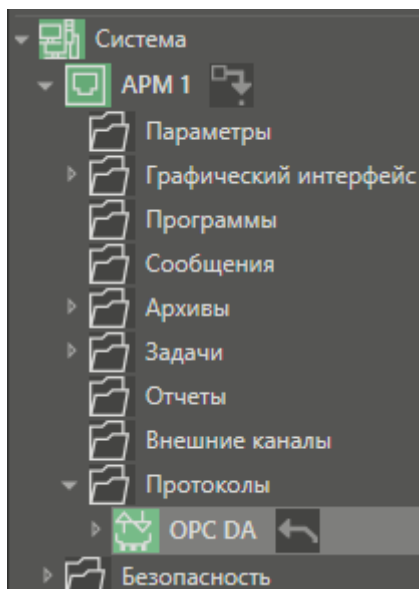
MasterSCADA 4D поддерживает стандарт OPC DA. Данный стандарт описывает получение текущих данных.

Это платформозависимый стандарт, который работает только на ОС Windows. Поэтому, если исполнительная система установлена на какой-либо другой ОС, то с OPC DA серверами она работать не сможет, даже если OPC-сервер установить на другой компьютер.

Для того чтобы MasterSCADA 4D выступала в роли клиента, необходимо в группу узла Протоколы добавить соответствующий протокол:

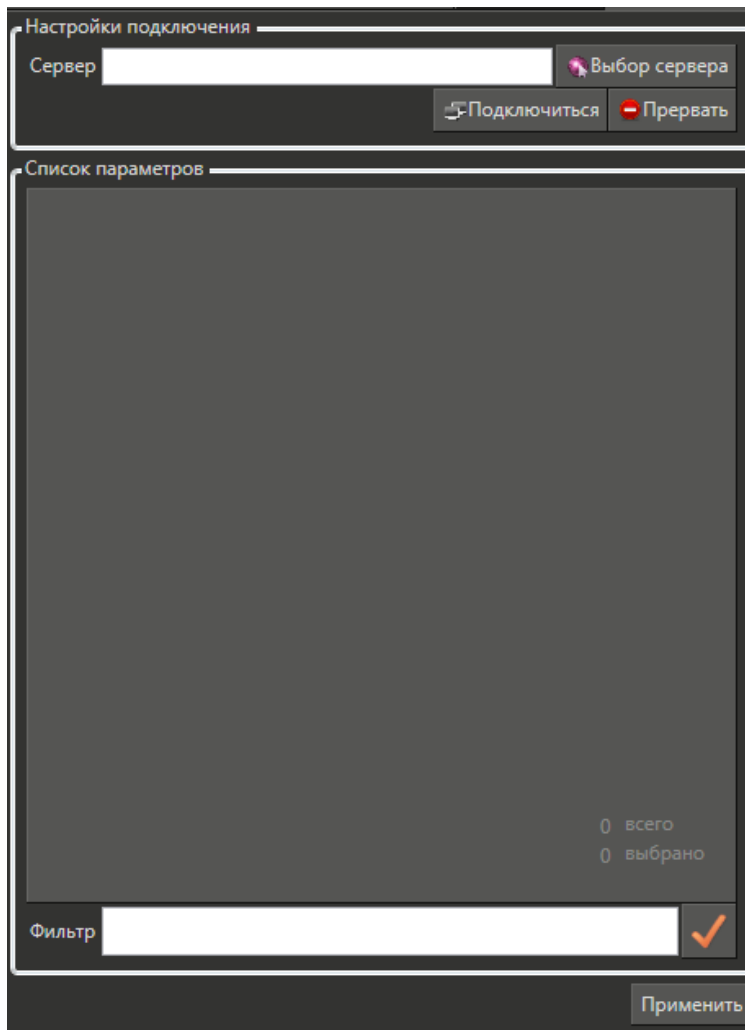


Получим результат в упрощенном дереве:



Затем необходимо выполнить подключение к OPC DA серверу для добавления каналов.

Для этого следует дважды нажать левую кнопку мыши на элементе в дереве. Откроется вкладка, в которой можно выполнить подключение к OPC DA серверу.



Для подключения требуется нажать кнопку **Выбор сервера**, при этом в диалоговом окне отобразятся доступные OPC DA серверы.

Затем следует нажать кнопку **Подключиться**.

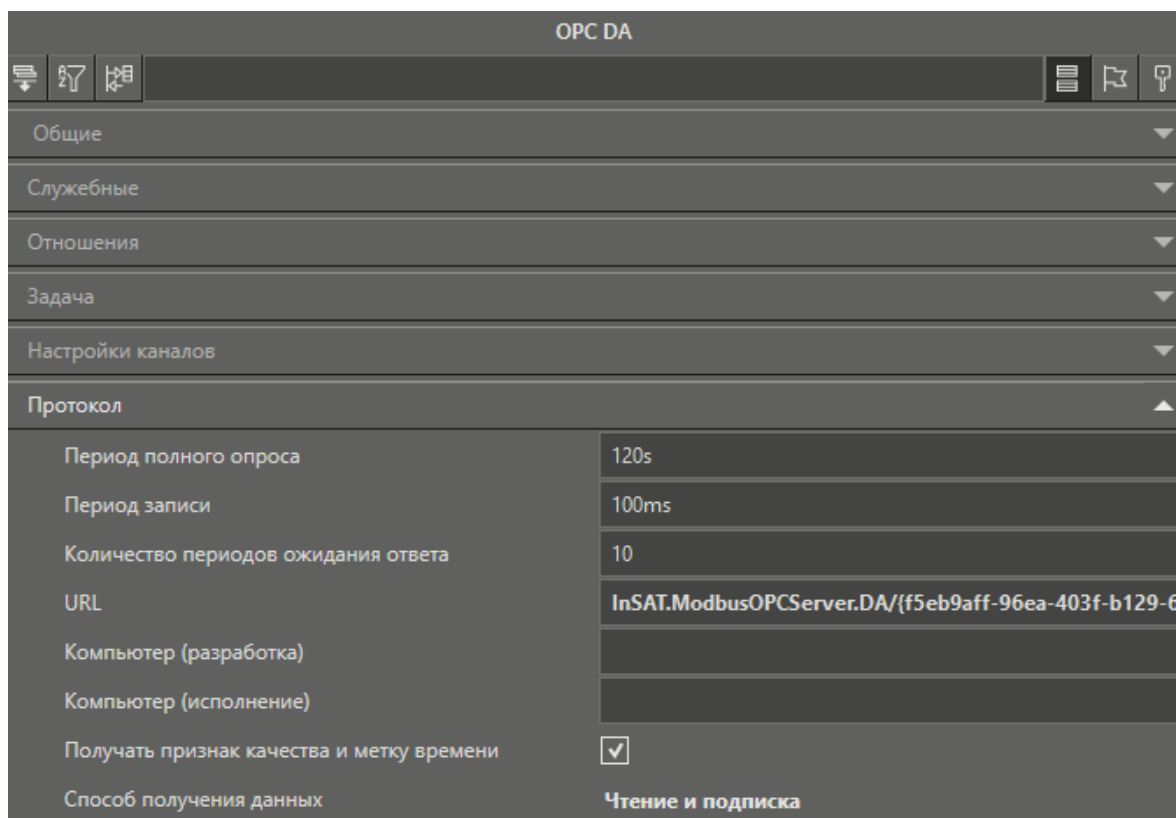
Если поиск и подключение к серверу будет удачным, то в группе список параметров отобразится дерево параметров OPC DA сервера, где нужно выбрать те элементы, с которыми необходимо работать в проекте MasterSCADA 4D. Далее необходимо нажать кнопку **Применить**. После этого в дерево системы добавятся каналы predefined типов

Важно! Поиск удаленных OPC-серверов не поддерживается. Для добавления удаленного OPC-сервера можно установить его локально, либо прописать ProgID вручную, а затем в панели свойств задать свойство **Компьютер** (разработка)

После чего необходимо настроить панель свойств OPC DA сервера.

6.2.3.1.4.1. СВОЙСТВА OPC DA

Вид панели свойств:



Описание:

Название	Рекомендации
Категория Общие	Как правило, для данного элемента не настраивается. Подробное описание данной категории приведено в соответствующем разделе
Категория Задача	Задается период, с которым MasterSCADA 4D подписывается получать измененные значения из сервера. Определяется способ работы протокола при резервировании узла. Подробное описание этой категории приведено в разделе, описывающем настройку категории Задачи у протоколов
Категория Протокол	
Период полного опроса	Если в данном поле установлено значение (в мс), то это означает, что MasterSCADA 4D будет получать не только изменившиеся данные с периодом, заданным в поле Период категории Задача, т.е. значения по Подписке, но и все значения в режиме Чтение, с периодом указанным в данном поле. Тип значения свойства -Time.

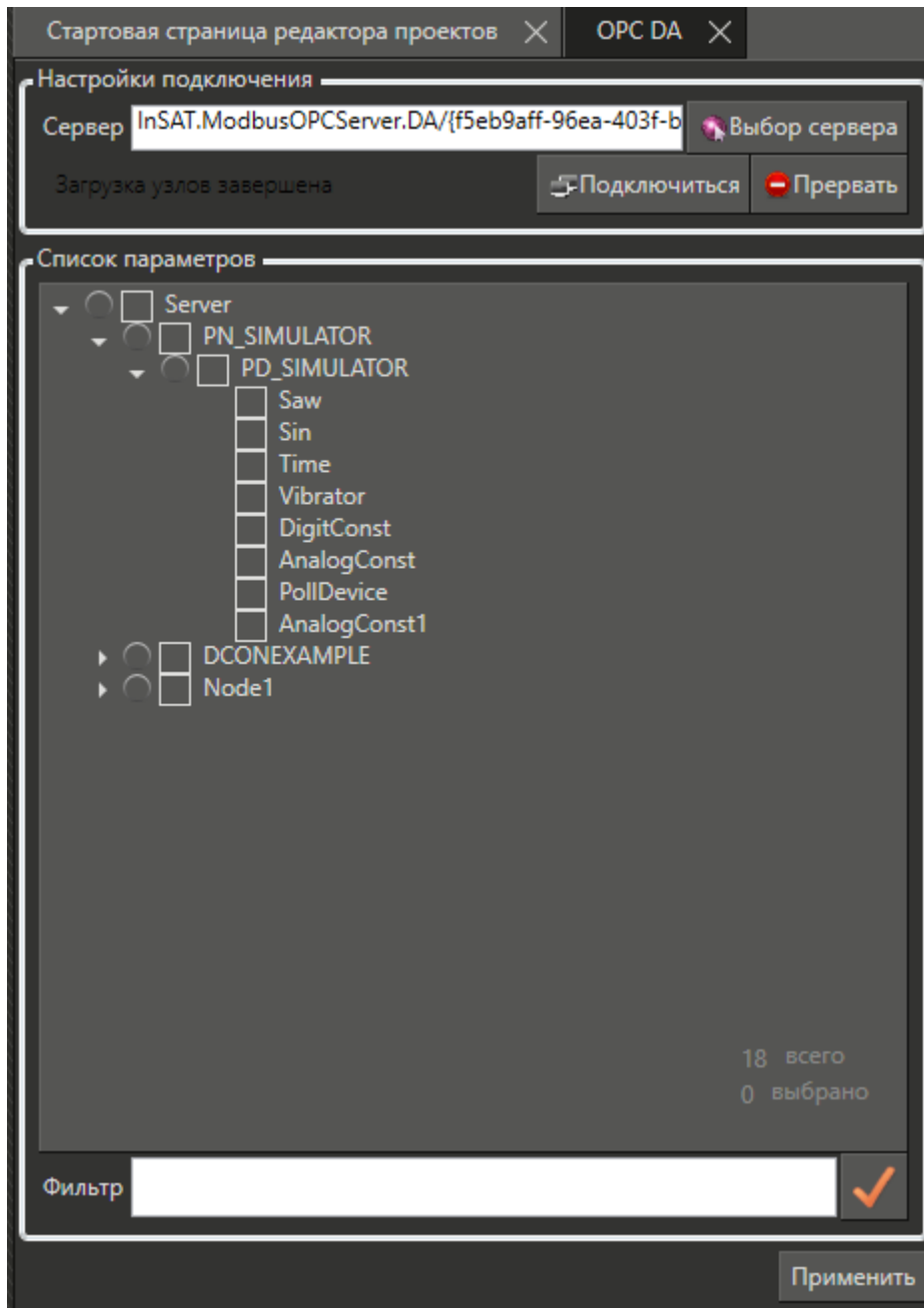
Название	Рекомендации
Период записи	Запись в OPC DA сервер происходит по изменению. Данное поле определяет период, в течении которого происходит накопление данных на запись, после чего происходит их запись в OPC DA сервер. Тип значения свойства - Time.
Получать признак качества и метку времени	Определяет тип данных для каналов: SYSTEM_*_PARAM (включает метку и качество), или атомарные. Если параметр атомарного типа архивируется, то в архив будут записываться метки времени и признаки качества, получаемые от сервера, но текущие значения метки времени и признака качества будут недоступны.
Количество периодов ожидания ответов	Если за указанное количество периодов подряд сервер не пришлет ответ, то MasterSCADA 4D отключится от сервера, а значение параметра Отказ протокола примет значение TRUE. Затем произойдет попытка подключиться к серверу заново.
URL	Задается ProgID и GUID. Поле заполняется автоматически после того, как сервер будет выбран в диалоге Выбор сервера. Если поле заполнять вручную, то достаточно ввести только ProgID. ProgID и GUID указывается в документации на OPC-сервер.
Компьютер (разработка)	Заполняется в случае, если для получения списка каналов необходимо подключаться к серверу, установленному на другом компьютере. Можно задать IP-адрес копьютера или DNS-имя.
Компьютер (исполнение)	Заполняется, если в режиме исполнения необходимо получать данные от OPC-сервера, установленного на другом компьютере. Можно задать IP-арес копьютера или DNS-имя.
Способ получения данных	<p>OPC DA клиент может работать в трех режимах:</p> <p>Подписка - MasterSCADA 4D подписывается к OPC-серверу на получение только тех данных, которые изменились за период, который указан в категории Задача в свойстве Период.</p> <p>Рекомендуется как самый эффективный способ получения данных . Если в категории Задача установлен флаг в поле Выполнять по условию, то в протоколе появится параметр</p>

Название	Рекомендации
	<p>Выполнить. Если выдать импульс на указанный параметр, то выполнится запрос обновления подписки.</p> <p>Чтение - независимо от того, изменились данные или нет, MasterSCADA 4D запрашивает с сервера все данные с периодом, указанным в поле Период полного опроса. Если в поле Период полного опроса установлено значение 0, то полный опрос производиться не будет. Однако, если в категории Задача установлен флаг в поле Выполнять по условию, то в протоколе появится параметр Выполнить. В этом случае, полный опрос будет производиться по переднему фронту изменения указанного параметра.</p> <p>Чтение и подписка - оба метода работают независимо друг от друга.</p>
Категория Настройки каналов	
NodeId	<p>NodeId - это идентификатор канала OPC-сервера. В данном поле можно задать значение по умолчанию, но как правило, не задается. У отдельных каналов данное поле заполняется автоматически</p>

Важно! Для работы с удаленными OPC-серверами необходимо настроить DCOM Windows.

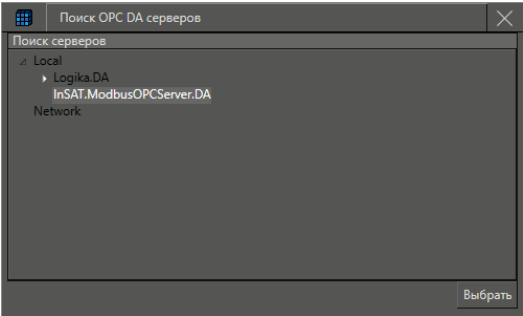

6.2.3.1.4.2. ДИАЛОГ ПОДКЛЮЧЕНИЯ К СЕРВЕРУ OPC DA

Данный диалог открывается в панели документов и редакторов по команде Открыть контекстного меню протокола OPC DA.



Инструменты диалога:

Название	Описание
Сервер	Задается ProgID и GUID, поле заполняется автоматически, после того, как сервер будет выбран в диалоге Выбор сервера. Если поле заполнять вручную, то достаточно ввести только ProgID. ProgID и GUID указываются в документации на OPC-сервер.

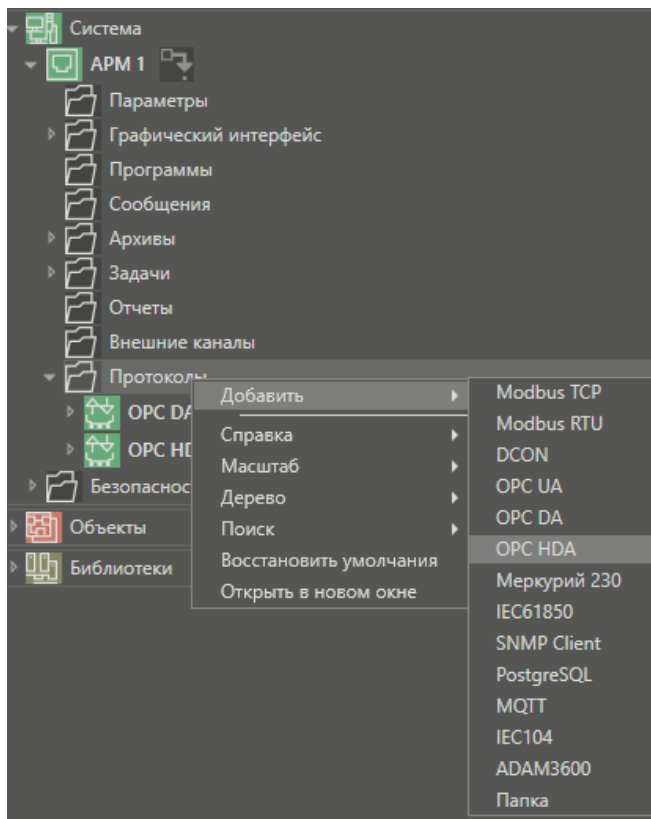
Название	Описание
Выбор сервера	<p>Открывает диалог выбора сервера. В нем отображаются все OPC-серверы, установленные на компьютере. Вид окна:</p>  <p>В нем нужно выбрать нужный OPC-сервер и нажать кнопку Выбрать</p>
Прервать	Прерывает подключение к серверу
Список параметров	<p>Отображается конфигурация сервера OPC DA, к которому выполнено подключение. При установке/снятии флагов с параметров сервера, соответствующие каналы создаются/удаляются в группе протокола OPC DA в дереве узла (иерархия групп OPC-сервера также воспроизводится в дереве узла) после нажатия кнопки Применить. Информация об общем количестве тегов и числе выбранных тегов отображается в нижней правой части раздела.</p>
Фильтр	<p>В строке задается подстрока поиска. Для применения фильтра требуется нажать кнопку </p>
Применить	Добавляет в дерево системы каналы, выбранные в группе Список параметров

6.2.3.1.5. OPC HDA

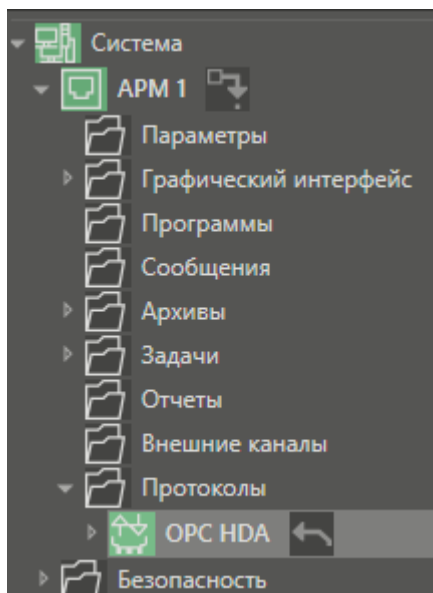
MasterSCADA 4D поддерживает стандарт OPC HDA. Данный протокол описывает получение архивных данных.

Это платформозависимый стандарт, который работает только на ОС Windows. Поэтому, если исполнительная система установлена на какой-либо другой ОС, то с OPC HDA серверами она работать не сможет, даже если OPC-сервер установить на другой компьютер.

Для того чтобы MasterSCADA 4D выступала в роли клиента, необходимо в группу узла Протоколы добавить соответствующий протокол:

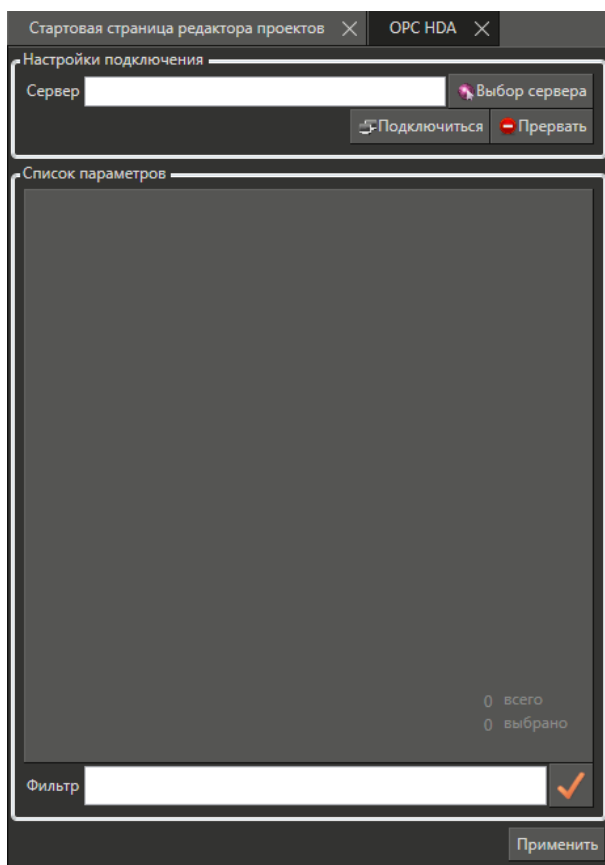


Получим результат в упрощенном дереве:



Затем необходимо выполнить подключение к OPC DA серверу для добавления каналов.

Для этого следует дважды нажать левую кнопку мыши на элементе в дереве. Откроется вкладка, в которой можно выполнить подключение к OPC DA серверу.



Для подключения требуется нажать кнопку **Выбор сервера**, при этом в диалоговом окне отобразятся доступные OPC HDA сервера.

Важно! Поиск удаленных OPC-серверов не поддерживается. Нужно установить OPC-сервер локально, добавить его, а затем, если необходимо работать с ним удаленно, задать нужные настройки в панели свойств OPC-сервера.

Затем следует нажать кнопку **Подключиться**.

Если поиск и подключение к серверу будет удачным, то в группе список параметров отобразится дерево параметров OPC DA сервера, в котором нужно выбрать те элементы, с которыми необходимо работать в проекте MasterSCADA 4D. Далее необходимо нажать кнопку **Применить**. После этого в дерево системы добавятся каналы predefined типов.

После этого необходимо настроить панель свойств OPC HDA сервера.

6.2.3.1.5.1. СВОЙСТВА OPC HDA

Вид панели свойств:

OPC HDA

🏠
🔍
🔧

☰
🚩
🔑

Общие

Служебные

Задача

Протокол

Получать признак качества и метку времени

Количество периодов ожидания ответа: 10

URL: InSAT.ModbusOPCServer.HDA/{f18a6209-45b0-472c-8415-2}

Компьютер (разработка):

Компьютер (исполнение):

Период чтения: 1s

Период опроса состояния: 10s

Период запроса при старте: 1d

Максимальное число значений в запросе: 1000

Способ получения данных: ReadAndSubscribe

Настройки каналов

Nodeld

Описание:

Название	Рекомендации
Категория Общие	Как правило, для данного элемента не настраивается. Подробное описание данной категории приведено в соответствующем разделе
Категория Задача	Задается период, с которым MasterSCADA 4D подписывается получать измененные значения из сервера. Определяется способ работы протокола при резервировании узла. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи у протоколов
Получать признак качества и метку времени	Определяет тип данных для каналов: SYSTEM_*_PARAM (включает метку и качество), или атомарные. Если параметр атомарного типа архивируется, то в режиме чтения архивов из OPC HDA сервера в архив будут записываться метки времени и признаки качества, получаемые от сервера, но текущие значения метки времени и признака качества будут недоступны

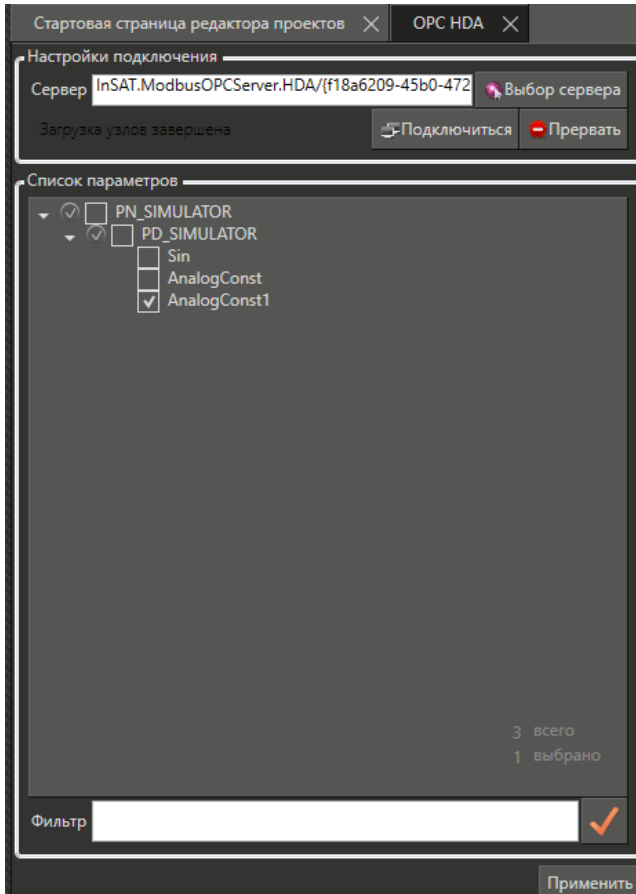
Название	Рекомендации
Количество периодов ожидания ответов	Если за указанное количество периодов подряд сервер не пришлет ответ, то MasterSCADA 4D отключится от сервера, а значение параметра Отказ протокола примет значение TRUE. Затем произойдет попытка подключиться к серверу заново.
URL	Задается ProgID и GUID. Поле заполняется автоматически после того, как сервер будет выбран в диалоге Выбор сервера. Если поле заполнять вручную, то достаточно ввести только ProgID. ProgID и GUID указывается в документации на OPC-сервер.
Компьютер (разработка)	Заполняется в случае, если для получения списка каналов необходимо подключаться к серверу, установленному на другом компьютере. Можно задать IP-адрес компьютера или DNS-имя.
Компьютер (исполнение)	Заполняется, если в режиме исполнения необходимо получать данные от OPC-сервера, установленного на другом компьютере. Можно задать IP-адрес компьютера или DNS-имя.
Период чтения	Период, с которым OPC клиент обращается к OPC серверу для получения архивных данных в режиме работы Чтение. Тип значения свойства - Time.
Период опроса состояния	С заданным периодом MasterSCADA 4D контролирует состояние подключения к серверу. Тип значения свойства - Time.
Период запроса при старте	Если MasterSCADA 4D подключается к OPC-серверу впервые, то она попытается считать архивные данные за указанное количество дней. Если в архиве уже есть записи по архивируемым параметрам, то данные будут получены с момента последней записи в архиве по текущий момент. Т.е. при повторных подключениях эта настройка не учитывается. Тип значения свойства - Time.
Максимальное число значений в запросе	Количество запрашиваемых OPC-клиентом архивных записей за один цикл опроса в режиме работы Подписка.

Название	Рекомендации
Способ получения данных	<p>OPC HDA клиент может работать в трех режимах:</p> <p>Подписка (Subscribe) - клиент при старте подписывается получать новые значения, желаемый период получения данных указывается в поле Период категории Задача.</p> <p>Чтение и подписка (ReadAndSubscribe) - клиент при старте читает, все значения, которых у него нет. Когда чтение закончено он подписывается на новые значения.</p> <p>Чтение (Read)- клиент с периодом заданным в свойстве Период чтения читает значения из сервера.</p>
NodeId	<p>NodeId - это идентификатор канала OPC-сервера. В данном поле можно задать значение по умолчанию, но как правило, оно не задается. У отдельных каналов данное поле заполняется автоматически</p>

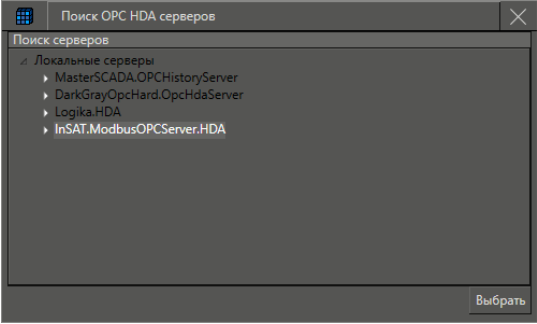
Важно! Для работы с удаленными OPC-серверами необходимо настроить DCOM Windows.


6.2.3.1.5.2. ДИАЛОГ ПОДКЛЮЧЕНИЯ К СЕРВЕРУ OPC HDA

Данный диалог открывается в панели документов и редакторов по команде Открыть контекстного меню протокола OPC HDA.



Инструменты диалога:

Название	Описание
Сервер	<p>Задается ProgID и GUID. Поле заполняется автоматически после того, как сервер будет выбран в диалоге Выбор сервера. Если поле заполнять вручную, то достаточно ввести только ProgID. ProgID и GUID указывается в документации на OPC-сервер.</p>
Выбор сервера	<p>Открывает диалог выбора сервера. В нем отображаются все OPC-серверы, установленные на компьютере. Вид окна:</p>  <p>В нем нужно выбрать нужный OPC-сервер и нажать кнопку Выбрать.</p>

Прервать	Прерывает подключение к серверу.
Список параметров	Отображается конфигурация сервера OPC HDA, к которому выполнено подключение. При установке/снятии флагов с параметров сервера, соответствующие каналы создаются/удаляются в группе протокола OPC HDA в дереве узла (иерархия групп OPC-сервера также воспроизводится в дереве узла) после нажатия кнопки Применить. Информация об общем количестве тегов и числе выбранных тегов отображается в нижней правой части раздела.
Фильтр	В строке задается подстрока поиска, для применения фильтра нужно нажать кнопку  .
Применить	Добавляет в дерево системы каналы, выбранные в группе Список параметров.

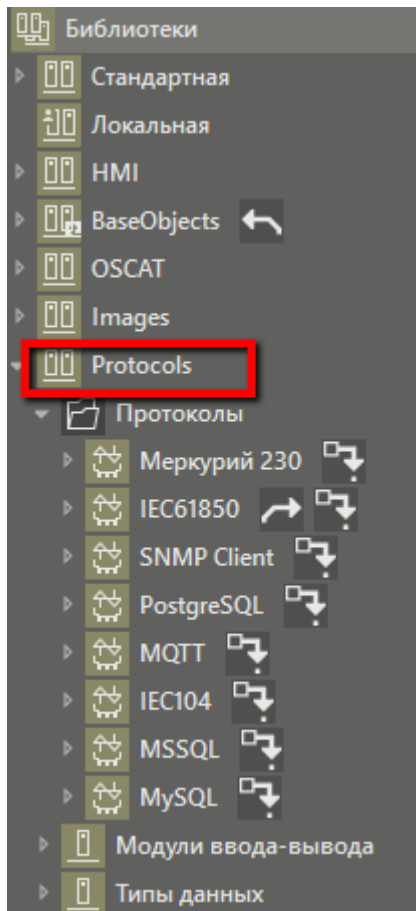
6.2.3.2. ДОПОЛНИТЕЛЬНЫЕ ПРОТОКОЛЫ

Кроме стандартных протоколов, входящих в состав среды исполнения, в нее можно добавить и другие протоколы.

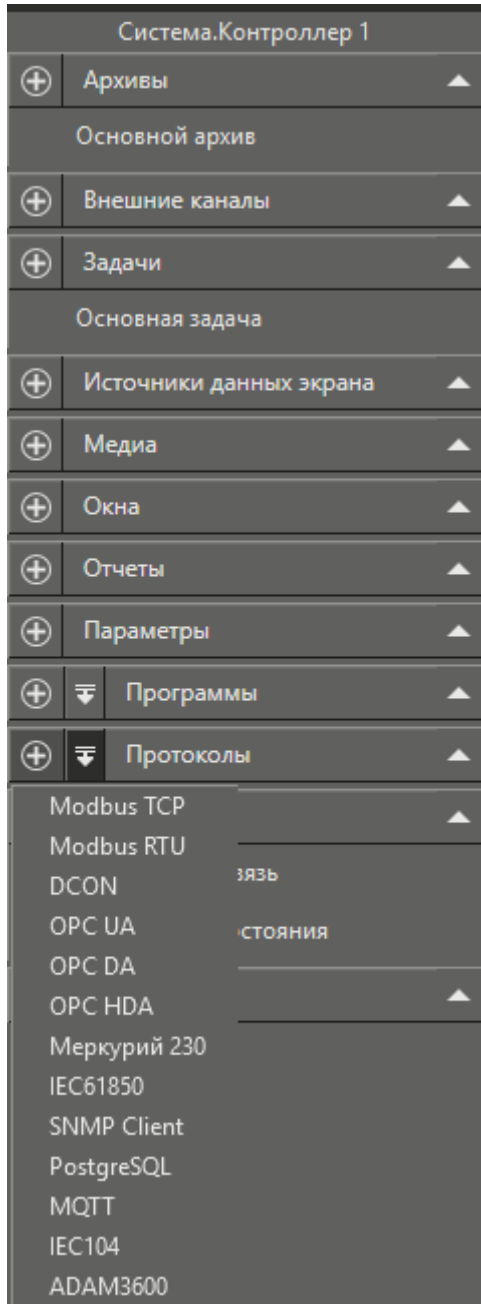
Важно! По умолчанию, дополнительные протоколы не входят в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте sales@masterscada.ru

В среде разработки дополнительные протоколы и соответствующие им модули ввода-вывода, а также дополнительные вспомогательные элементы, необходимые для работы могут храниться в библиотеке Protocols, а также в специализированных библиотеках, которые необходимо подключать дополнительно для использования в проекте.

Вид библиотеки Protocols:



В контекстном меню узла в пункте Добавить.Протокол, а также в контекстной панели узла отображаются все протоколы, подключенные к проекту, как содержащиеся в библиотеке Protocols, так и в других библиотеках:



MasterSCADA 4D при работе с этими протоколами выступает в роли Master.

По умолчанию доступны в среде разработки следующие протоколы:

- IEC61850
- SNMP Client
- MQTT
- IEC104

Можно подключить следующие библиотеки, содержащие протоколы:

- VKT7
- SET4

- VKT9
- Mercury
- SET4
- Energomera
- Pulsar
- ESKO
- Тепловычислители Теплоком
- Vzljot
- Mkts
- BACnet
- TEM

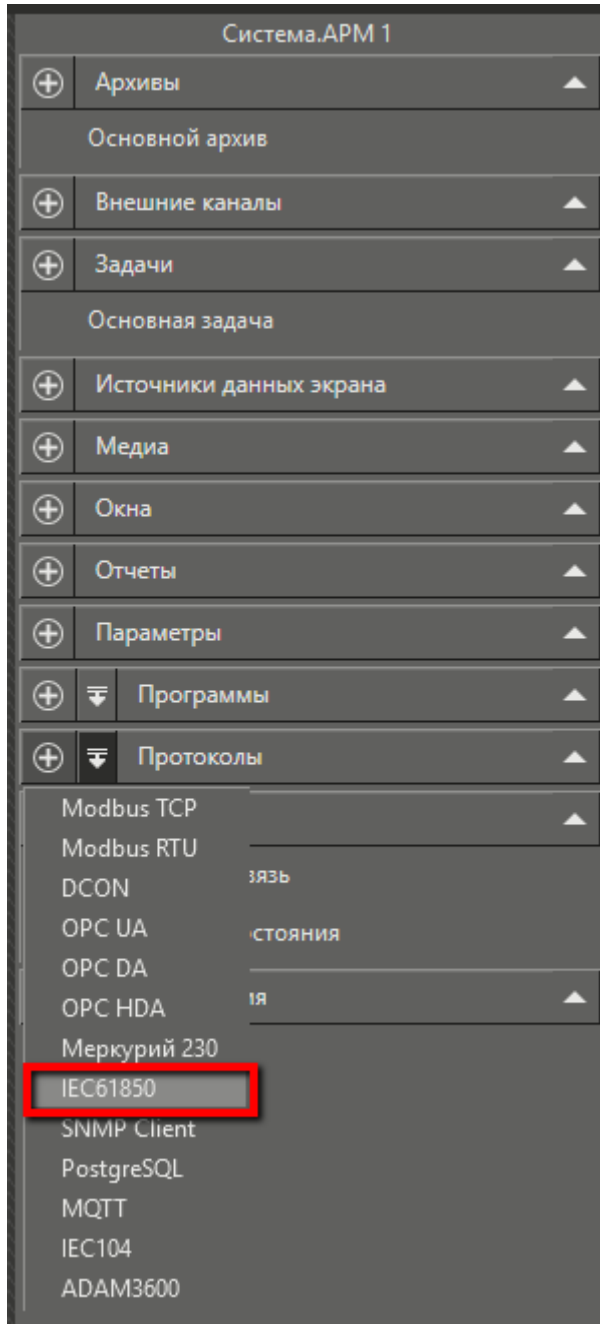
Протоколы для работы с базами данных:

- Протокол PostgreSQL
- Протокол MSSQL
- Протокол MySQL

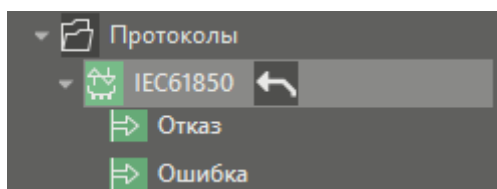
6.2.3.2.1. IEC61850

MasterSCADA 4D поддерживает платформонезависимый протокол МЭК-61850, который широко используется в энергетике.

Для того чтобы MasterSCADA 4D выступала в роли клиента, необходимо в группу узла Протоколы добавить IEC61850:

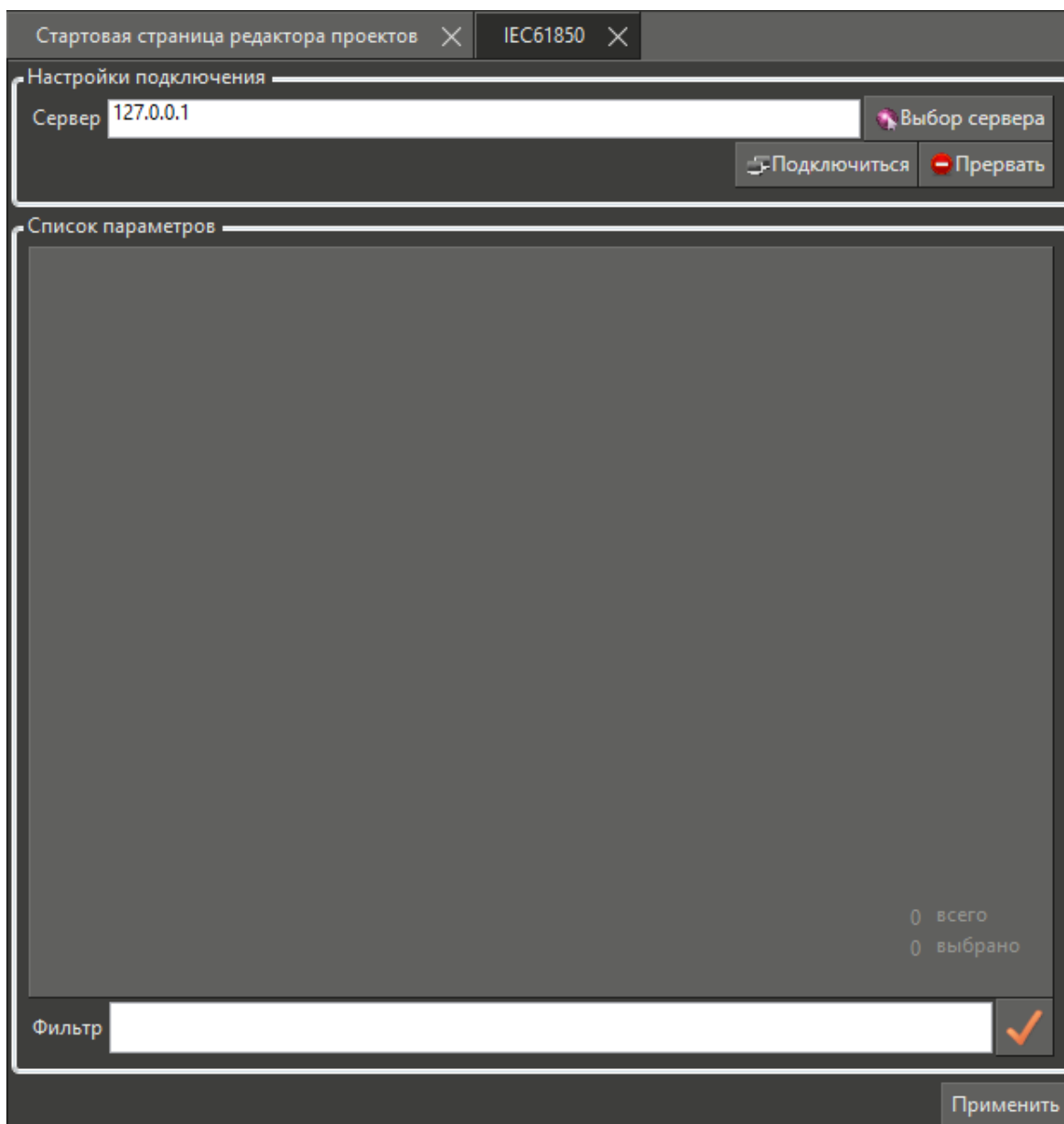


Получим результат в упрощенном дереве:

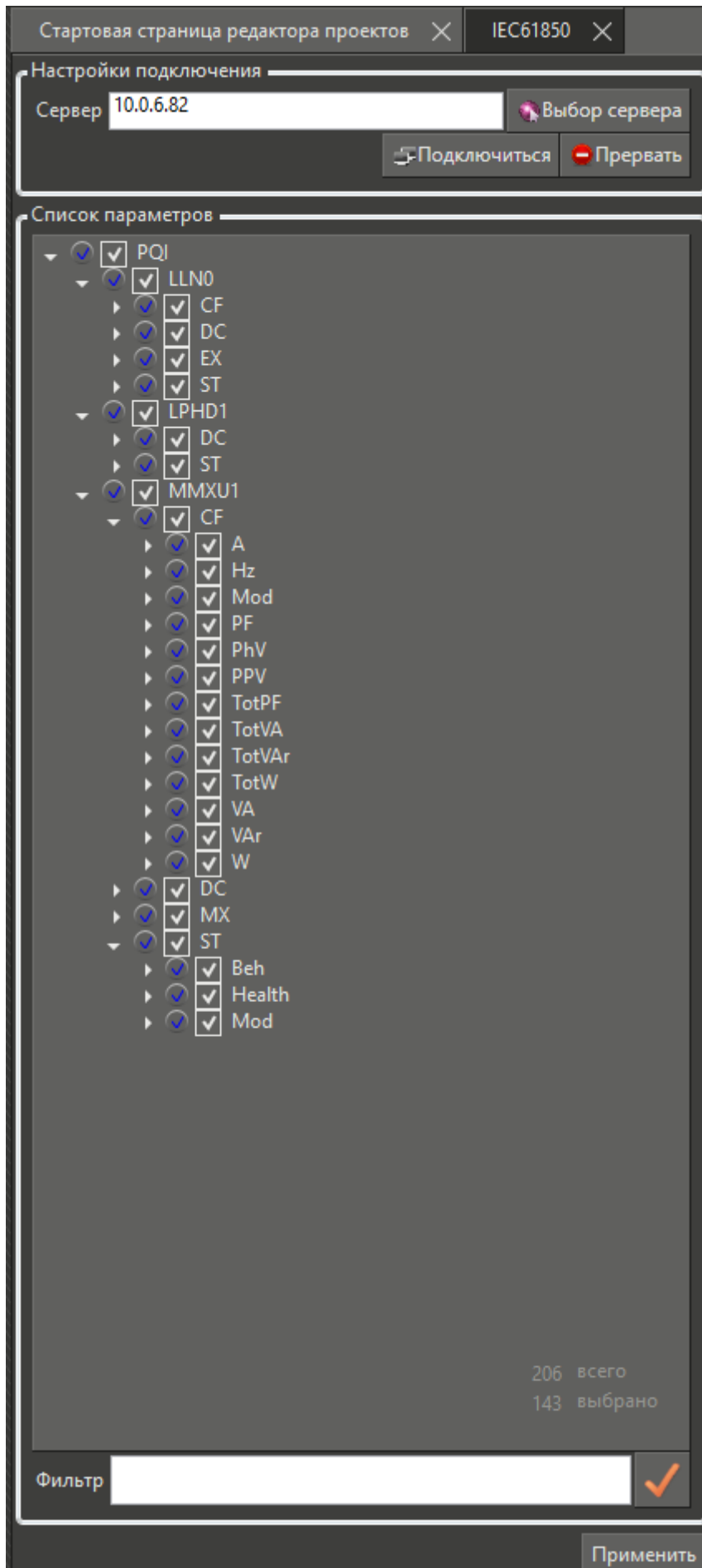


Затем, для того чтобы добавить каналы, необходимо выполнить подключение к серверу.

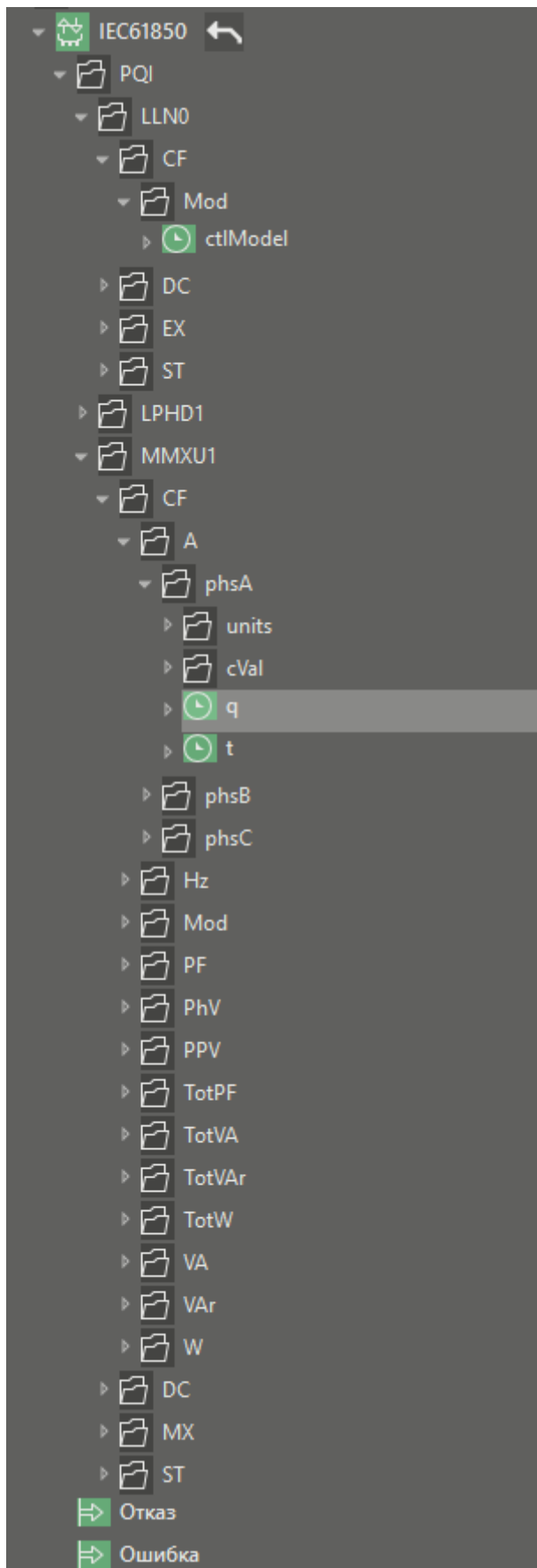
Для этого нужно дважды нажать левую кнопку мыши на элементе в дереве. При этом откроется вкладка:



В строке Сервер необходимо указать IP-адрес устройства, на котором установлен ИЕС61850-сервер, а затем нажать кнопку Подключиться. В результате отобразится список доступных параметров, в котором флагами нужно отметить необходимые:



В результате получим:



Список каналов и их настройки формируются автоматически. Дополнительная настройка каналов не требуется.

При первом добавлении протокола, настройки формируются также автоматически, но, в случае необходимости, их можно изменить в панели свойств.

6.2.3.2.1.1. СВОЙСТВА IEC61850

Вид панели свойств:

IEC61850	
Общие	
Имя	IEC61850
Программное имя	IEC61850
Полное имя	Система_APM 1.Протоколы.IEC61850
Метки	
Комментарий	
Служебные	
Отношения	
Задача	
Настройки каналов	
NodeID	
FC	
Протокол	
IP адрес	127.0.0.1
Параметры	
TCP порт	102
Использовать атрибуты \$q и \$t для \$stVal	<input type="checkbox"/>
Кол-во тегов в запросе чтения	1
Каталог для записи файлов	/
Разрешение чтения файлов	<input type="checkbox"/>
Разделитель папок	/
Удаление прочитанных файлов	<input type="checkbox"/>
Интервал чтения файлов	1
Использовать отчеты	<input type="checkbox"/>
Использовать наборы данных	<input type="checkbox"/>
Исключать параметры, получаемые через отчеты из циклического опроса	<input type="checkbox"/>

Название	Рекомендации
Категория Общие	Как правило, для данного элемента эта категория не настраивается. Подробное описание данной категории приведено в соответствующем разделе
Категория Задача	Задается период, с которым MasterSCADA 4D будет стараться получать значения из сервера. Определяется

Название	Рекомендации
	<p>способ работы протокола при резервировании узла. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи у протоколов.</p>
Настройки каналов	<p>Задаются значения по умолчанию для создаваемых каналов. В настройке не нуждаются, т.к. индивидуальные настройки каналов формируются автоматически.</p>
IP-адрес	<p>IP-адрес сервера, который отправляет и получает данные от MasterSCADA 4D.</p>
TCP-порт	<p>Номер TCP-порта сервера, по которому осуществляется подключение.</p>
Использовать атрибуты \$q и \$t для \$stVal	<p>При установке этого параметра, значения \$q и \$t передаются как атрибуты Quality (Качество) и TimeStamp (Метка времени) для stVal.</p> <p>Важно!!! При установке этого свойства тип параметра stVal меняется на SYSTEM_LINT_PARAM.</p> <p>Если при опросе параметра \$stVal происходит ошибка, то атрибут Quality также устанавливается в BAD.</p> <p>При отсутствии или ошибке получения признака качества q, для параметров типа SYSTEM_*_PARAM атрибут StatusCode выставляется в зависимости от результата опроса. При успешном опросе он принимает значение GOOD, в противном случае – BAD.</p> <p>Правило формирования атрибута Quality приведено в таблице ниже.</p>
Кол-во тегов в запросе чтения	<p>Определяет количество параметров, опрашиваемых в одном запросе.</p>
Каталог для записи файлов	<p>Задается каталог на локальном носителе, в который будет производиться запись файлов, полученных от сервера IEC 61850.</p>

Название	Рекомендации
Разрешение чтения файлов	Флаг определяет необходимость чтения файлов. Если флаг установлен, то из устройства считываются файлы при старте. После этого происходит сравнение считанных файлов с файлами, хранящимися в устройстве, и чтение новых файлов происходит уже с периодом, заданном в свойстве Интервал чтения файлов
Разделитель папок	Задается разделитель папок файловой системы, установленный в сервере IEC 61850.
Удаление прочитанных файлов	Флаг определяет необходимость удаления файлов в сервере IEC 61850 после их получения.
Интервал чтения файлов	Задается периодичность чтения файлов с сервера IEC 61850. Значение свойства задается в секундах.
Использовать отчеты	Включает получение отчетов от устройства.
Использовать наборы данных	Включает групповой опрос набора данных DataSets, определенного в устройстве. В этом случае данные, определённые в DataSets исключаются из индивидуального опроса.
Исключать параметры, получаемые через отчеты из циклического опроса	Если флаг установлен, то параметры от устройства будут поступать только в случае получения отчетов.
Использовать TLS соединение	<p>Задаёт использование защищенного подключения когда MasterSCADA 4D является клиентом.</p> <p>Ключи и сертификаты TLS хранятся в папке IEC61850. Ключи и сертификаты сервера хранятся в папке <i>IEC61850/srv</i>. Ключи и сертификаты клиента хранятся в папке <i>IEC61850/clt</i>.</p> <p>Имя файла ключа клиента должно быть <i>MasterSCADA4Dclt_key.pem</i>. Имя файла сертификата клиента должно быть <i>MasterSCADA4Dclt.cer</i>.</p>

Название	Рекомендации
	<p>Имя файла ключа сервера должно быть <i>MasterSCADA4Dsrv_key.pem</i>. Имя файла сертификата сервера должно быть <i>MasterSCADA4Dsrv.cer</i>.</p> <p>Сертификаты CA должны храниться в папке <i>IEC61850/ctl/root</i> для клиента и <i>IEC61850/srv/root</i> для сервера. Сертификаты CA могут иметь любое имя и расширение <i>.cer</i>.</p>

Атрибут Quality формируется в соответствии с таблицей:

Признак качества IEC61850	Цифровое значение IEC61850	Описание	Quality
QUALITY_VALIDITY_GOOD	0	Хороший признак качества (нет ошибок).	GOOD
QUALITY_DETAIL_OVERFLOW	4	Переполнение. Например, значение измеренной величины может превышать диапазон значений для рассматриваемого типа данных.	UNCERTAIN
QUALITY_DETAIL_OUT_OF_RANGE	8	Значение находится вне диапазона значений.	SENSOR_CAL
QUALITY_DETAIL_BAD_REFERENCE	16	Потеря калибровки.	SENSOR_CAL
QUALITY_DETAIL_OSCILLATORY	32	Дребезг контакта (частое включение-выключение дискретного сигнала).	UNCERTAIN
QUALITY_DETAIL_FAILURE	64	Внутренняя или внешняя неисправность.	UNCERTAIN
QUALITY_DETAIL_OLD_DATA	128	Данные не обновлялись в течении долгого времени.	UNCERTAIN

Признак качества IEC61850	Цифровое значение IEC61850	Описание	Quality
QUALITY_DETAIL_INCONSISTENT	256	Несоответствие данных действительности.	UNCERTAIN
QUALITY_DETAIL_INACCURATE	512	Значение не соответствует точностным характеристикам.	UNCERTAIN
QUALITY_SOURCE_SUBSTITUTED	1024	Значение установлено пользователем.	LOCAL_OVERRIDE
QUALITY_TEST	2048	Устройство находится в режиме тестирования.	LOCAL_OVERRIDE
QUALITY_OPERATOR_BLOCKED	4096	Обновление данных заблокировано оператором.	LOCAL_OVERRIDE

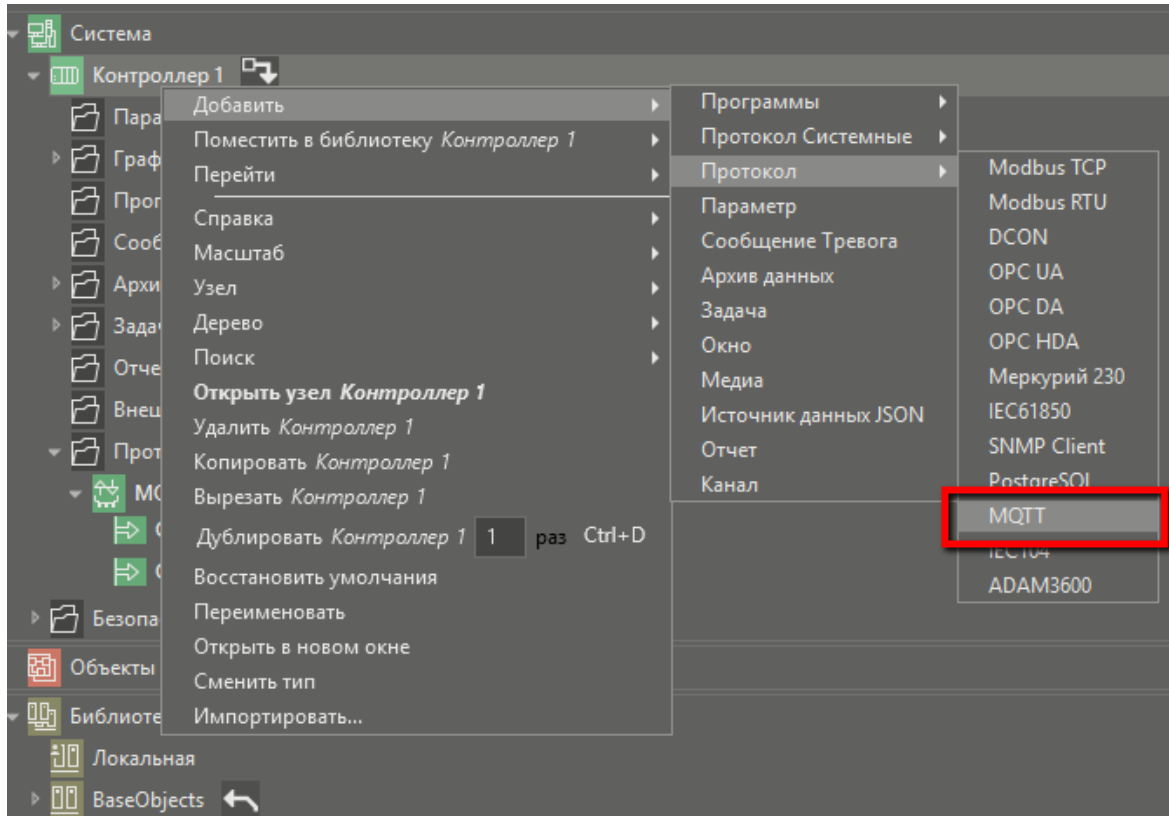
6.2.3.2.2. MQTT

Протокол MQTT – Message Queuing Telemetry Transport – протокол для передачи последовательности сообщений с телеметрическими данными, то есть информации от датчиков температуры, влажности, освещенности и др.

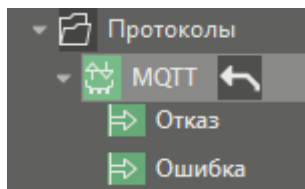
Это платформонезависимый протокол, который может использоваться в исполнительных системах, предназначенных для различных операционных систем.

Важно! По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте sales@masterscada.ru

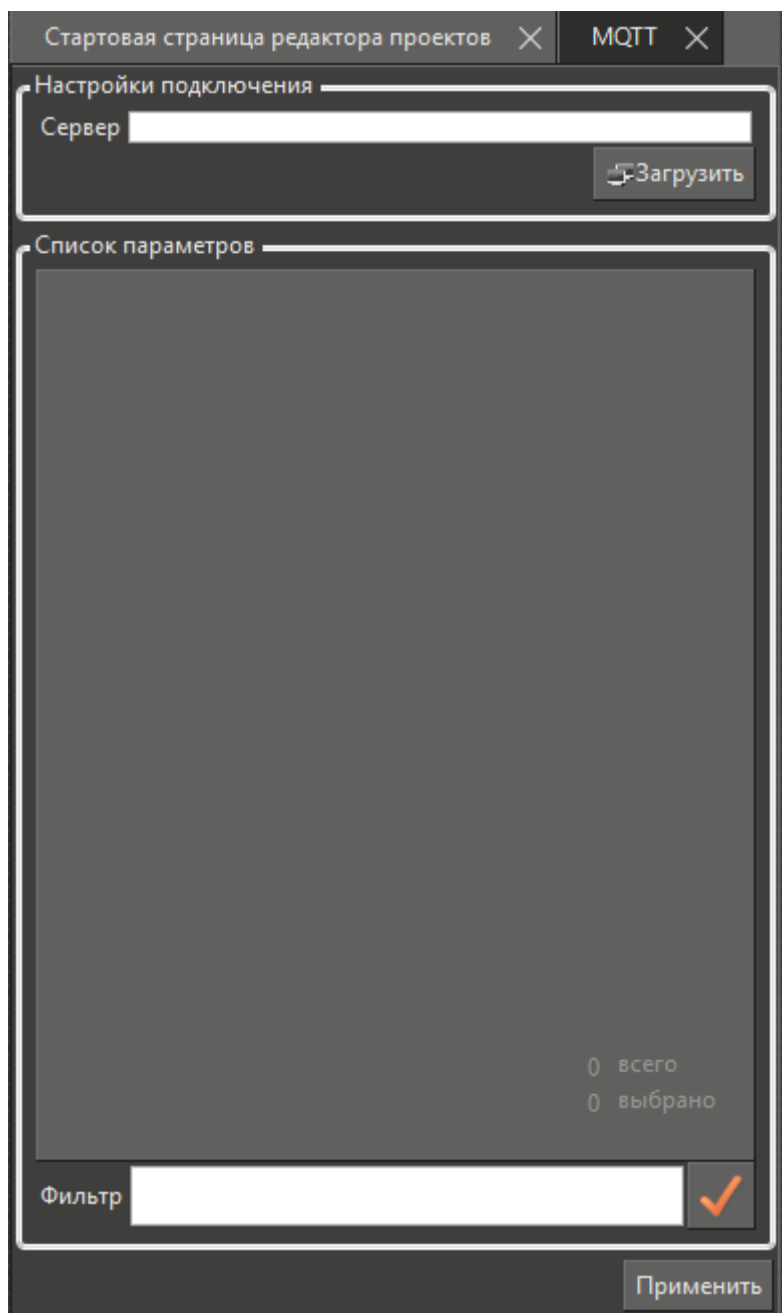
Для добавления протокола в проект необходимо выполнить пункт контекстного меню узла Добавить.Протокол.MQTT:



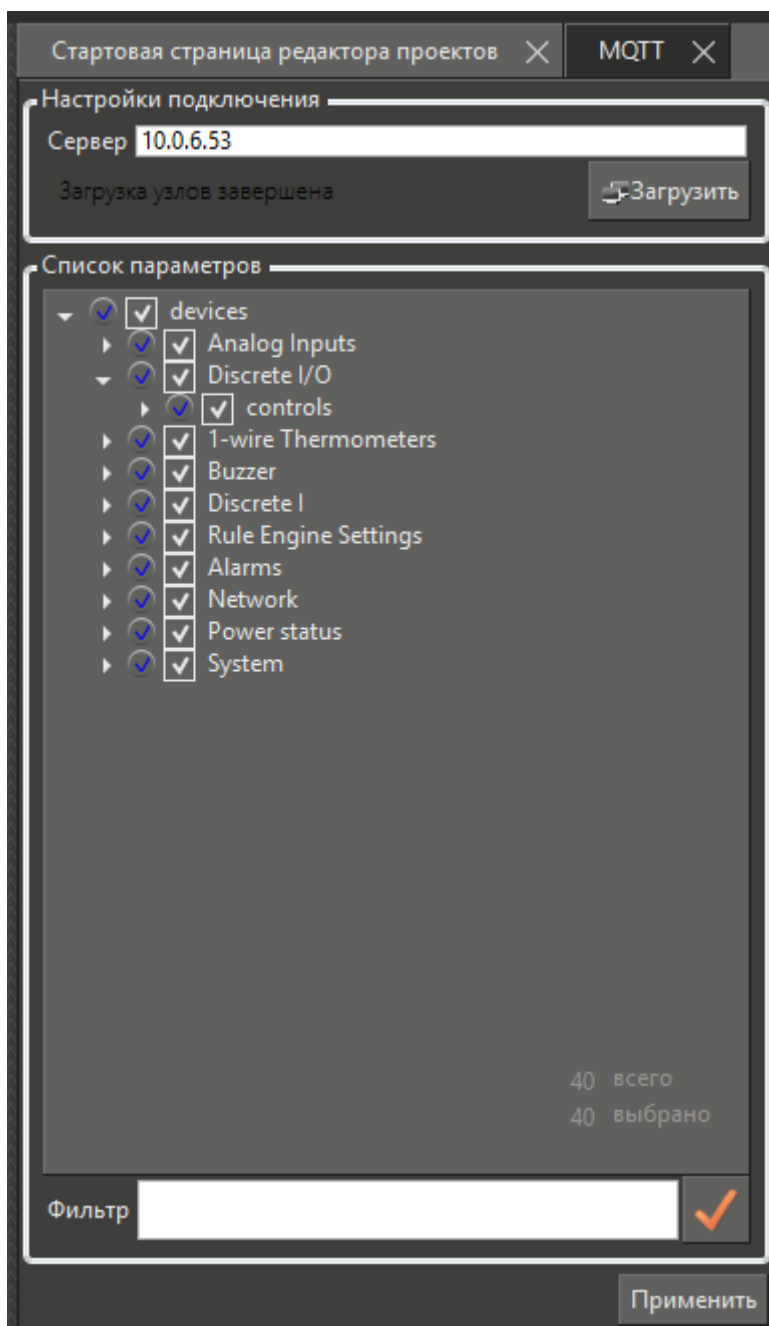
В результате получим:



Панель добавления тегов в протокол открывается, если выполнить двойное нажатие левой кнопкой мыши на новый элемент:

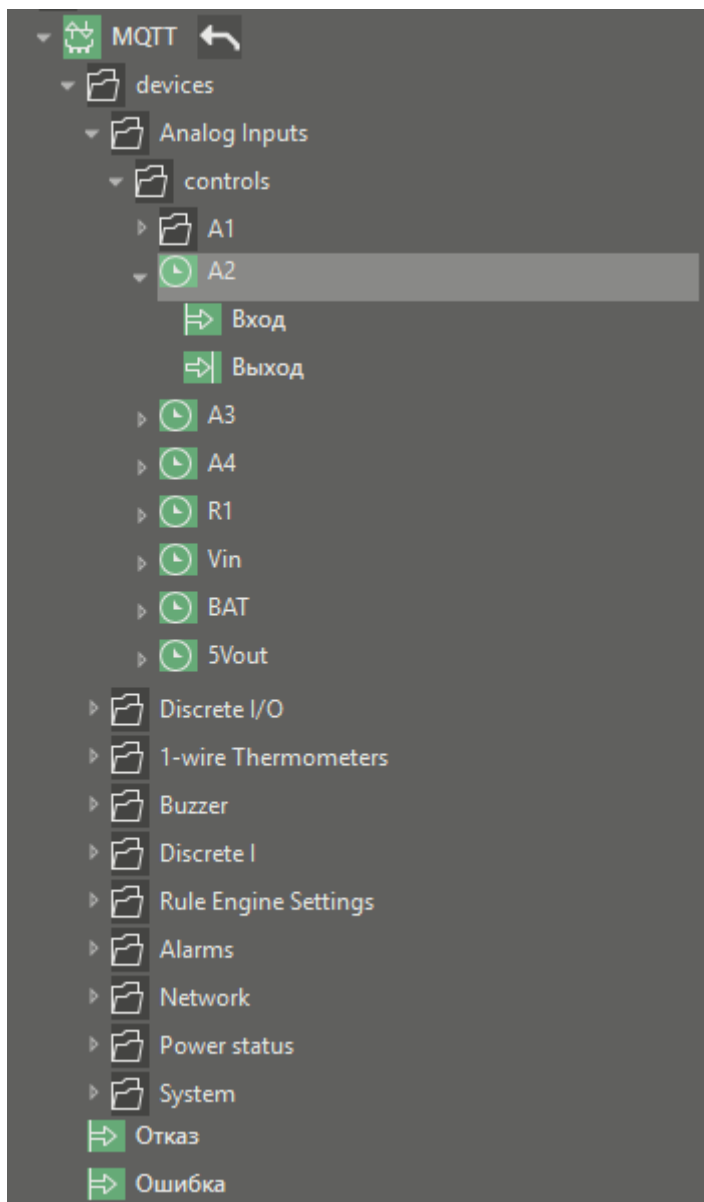


В строке Сервер необходимо указать IP-адрес устройства, на котором установлен MQTT-сервер, а затем нажать кнопку Загрузить. В результате отобразится список доступных параметров, в котором флагами нужно отметить необходимые:



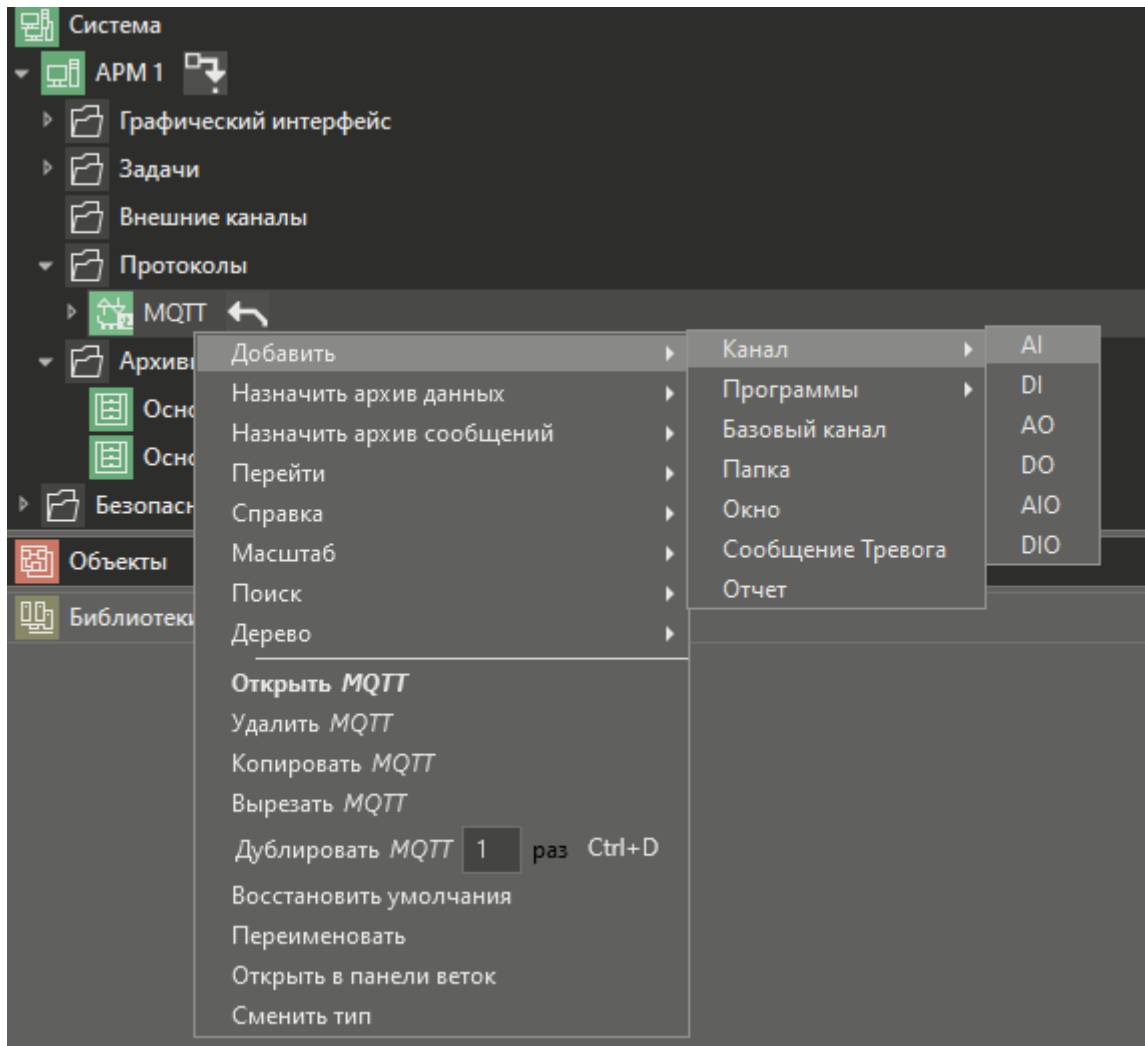
Далее необходимо нажать кнопку Применить.

В результате получим:



Свойства каналов в этом случае, как правило, не настраиваются.

Если изначально топика нет на сервере, то в протоколе нужно вручную создать канал:



В настройках указать путь к топику.

Дополнительные настройки протокола задаются в панели свойств.

6.2.3.2.2.1. СВОЙСТВА MQTT

Вид панели свойств протокола MQTT:

MQTT

☰ 🔍 🔗 ☰ 🔍 🔗

Общие ▲

Имя MQTT

Метки

Комментарий

Служебные ▼

Задача ▲

Период, мс 100

Приоритет 100

Подключение по условию

Выполнение по условию

Выполнять на резервном

Настройки каналов ▲

Топик

Протокол ▲

Сервер(разработка) 10.0.6.53

Сервер(исполнение)

Порт 1883

Имя пользователя

Пароль

Получать метку времени и при

Игнорируемые топики

Суффикс для записи

Название	Рекомендации
Категория Общие	Как правило для данного элемента эта категория не настраивается. Подробное описание данной категории приведено в соответствующем разделе.
Категория Задача	Задается период, с которым MasterSCADA 4D будет стараться получать значения из сервера. Определяется

Название	Рекомендации
	способ работы протокола при резервировании узла. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи у протоколов.
Сервер (разработка)	Задается IP-адрес устройства, на котором установлен MQTT-сервер, для получения списка каналов. Можно задать IP-адрес или DNS-имя.
Сервер (исполнение)	Задается IP-адрес устройства, на котором установлен MQTT-сервер, для получения данных в режиме исполнения. Можно задать IP-адрес или DNS-имя. Если MQTT-сервер находится на том же устройстве, на котором установлена среда исполнения, то данное поле можно оставить пустым.
Порт	Номер порта TCP IP, который используется для работы с MQTT-сервером.
Имя пользователя	Задается имя пользователя, если оно задано в настройках MQTT-сервера.
Пароль	Задается пароль, заданный в настройках MQTT-сервера.
Получать метку времени и признак качества	Определяет тип данных для каналов: SYSTEM_*_PARAM (включает метку и качество) или атомарные. Флаг должен быть установлен, т.к. за один цикл опроса может быть получено несколько значений одновременно.
Игнорируемые топики	Игнорировать топики, которые заканчиваются на значение, заданное в данном поле. Значения могут быть заданы через запятую.
Суффикс для записи	В некоторых серверах топик на чтение отличается от топика на запись окончанием. В этом случае можно задать окончание в данном поле. Например, для того чтобы работало изменение значений входов/выходов контроллера Wirenboard, добавленных через протокол MQTT, необходимо выставлять значение суффикса для записи /op.

6.2.3.2.2. СВОЙСТВА КАНАЛА MQTT

Вид панели свойств:

Описание основных свойств:

Название	Рекомендации
Группа Настройки	
Торіс	Указывается имя раздела (топика)
QoS	<p>Указывается тип QoS. Доступные варианты:</p> <ul style="list-style-type: none"> • 0 - без подтверждения получения, • 1 - с подтверждением получения, • 2 - с подтверждением подтверждения получения.

Описание остальных свойств смотрите в соответствующих разделах

6.2.3.2.2.3. ЧТЕНИЕ СЛОЖНЫХ ТИПОВ ДАННЫХ В ФОРМАТЕ JSON ИЗ MQTT

Если в топик MQTT передаются данные в формате JSON, то нужно в пользовательской библиотеке создать необходимый сложный тип данных, и у параметра Вход канала установить данный тип.

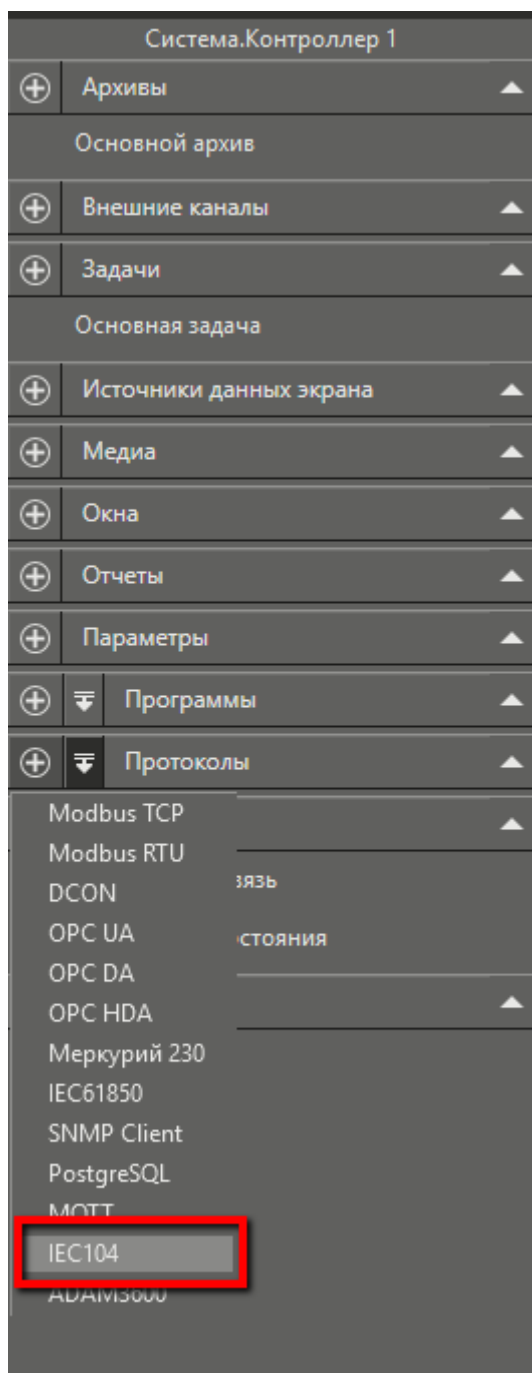
Правила формирования такого типа данных смотрите в разделе: Соответствие элементов JSON и MasterSCADA 4D

6.2.3.2.3. IEC104

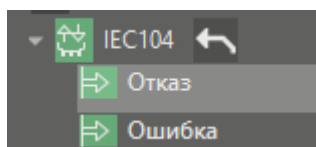
MasterSCADA 4D RT может получать/отправлять данные от устройств и других приложений по протоколу IEC 60870-5-104 (МЭК 60870-5-104). Это платформонезависимый протокол, который может использоваться в исполнительных системах, предназначенных для различных операционных систем.

Важно! По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте sales@masterscada.ru

Если необходимо получать данные по протоколу IEC 60870-5-104, то в узел, при помощи контекстного меню или контекстной панели, добавляется протокол IEC104:

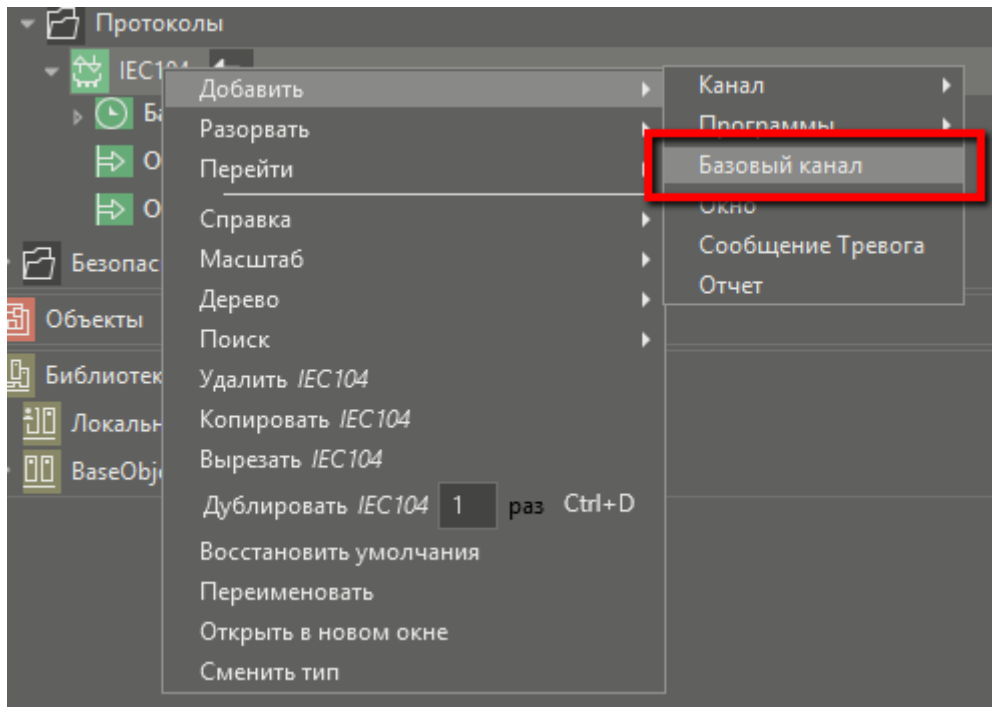


В результате получим:



Затем необходимо настроить панель свойств.

При помощи контекстного меню протокола можно добавить каналы, например, универсальный Базовый канал:



Затем настраивается панель свойств канала.

6.2.3.2.3.1. СВОЙСТВА IEC104

Вид панели свойств протокола IEC104:

IEC104

Общие

Имя	IEC104
Полное имя	Система.АРМ 1.Протоколы.IEC104
Метки	
Комментарий	

Служебные

Задача

Отношения

Настройки каналов

Протокол

IP адрес	127.0.0.1
IP порт	2404
Общий адрес CA	1
t0	30
t1	15
t2	10
t3	20
k	12
w	8
Период общего опроса	0s
Группы общего опроса	0
Период опроса счетчиков	0s
Группы опроса счетчиков	0
Период опроса командой 102	0s
Запрашивать данные групповым запросом при старте	<input checked="" type="checkbox"/>

Название	Рекомендации
Категория Общие	Как правило, для данного элемента эта категория не настраивается. Подробное описание данной категории приведено в соответствующем разделе.
Категория Задача	<p>Задается период, с которым MasterSCADA 4D будет стараться получать значения из сервера. Определяется способ работы протокола при резервировании узла.</p> <p>Подробное описание категории приведено в разделе, описывающем настройку категории Задачи у протоколов.</p>

Название	Рекомендации
Категория Настройки каналов	Задаются значения по умолчанию для создаваемых каналов. В дальнейшем эти настройки у каналов можно будет изменить.
Категория Протокол	
IP-адрес	IP-адрес сервера, который отправляет и получает данные от MasterSCADA 4D.
IP-порт	Номер порта сервера, по которому осуществляется подключение. Как правило, используется порт 2404
Общий адрес CA	Common address (адрес устройства)
t0	Таймаут при установлении соединения. Максимальный диапазон значения составляет от 1 до 255 с точностью до 1 секунды. По умолчанию установлено наиболее часто используемое значение.
t1	Таймаут при посылке или тестировании APDU. Максимальный диапазон значения составляет от 1 до 255 с точностью до 1 секунды. По умолчанию установлено наиболее часто используемое значение.
t2	Таймаут для подтверждения в случае отсутствия сообщения с данными $t2 < t1$. Максимальный диапазон значения составляет от 1 до 255 с точностью до 1 секунды. По умолчанию установлено наиболее часто используемое значение.
t3	Таймаут для посылки блоков тестирования в случае долгого простоя. Максимальный диапазон значения составляет от 1 до 255 с точностью до 1 секунды. По умолчанию установлено наиболее часто используемое значение.
k	Максимальная разность между переменной состояния передачи и номером последнего подтвержденного APDU.

Название	Рекомендации
	Максимальный диапазон значений составляет от 1 до 32767 APDU с точностью до 1 APDU.
w	Последнее подтверждение после приема w APDU. Максимальный диапазон значений составляет от 1 до 32767 APDU с точностью до 1 APDU. Значение w не должно быть более двух третей значения k.
Период общего опроса	Определяется период общего опроса (используется команда <100>) Если установлено значение 0. то общий опрос использоваться не будет
Группы общего опроса	В данном поле перечисляются группы общего опроса. 0 - общий опрос, 1 - первая группа, 2 - вторая группа и т.д. Перечисление производится через запятую без пробелов - 0,12,11,9
Период опроса счетчиков	Определяется период опроса счетчиков параметров станции (если 0, запрос не генерируется), командой опроса <101>: C_CI_NA_1
Группы опроса счетчиков	В данном поле перечисляются группы опроса счетчиков. 0 - общий опрос, 1 - первая группа, 2 - вторая группа и т.д. Перечисление производится через запятую без пробелов - 0,1,2
Период опроса командой 102	Определяется период запроса команды считывания <102>: C_RD_NA_1 (если 0, запрос не генерируется). Данной командой производится считывание тегов, в настройках которых включен соответствующий параметр.
Запрашивать данные групповым запросом при старте	Определяет способ опроса данных при старте исполнительной системы. Если настройка включена, то при старте значения всех тегов устройства опрашиваются с помощью группового запроса (командами 100,101). В противном случае опрос при старте не производится, и сервер ожидает получения данных от станции (по изменению или по циклу отдачи), либо ответа на общий опрос.

APDU - Application Protocol Data Unit (Протокольный Блок Данных Прикладного уровня).

ASDU - Application Service Data Unit (Блок Данных Прикладного уровня).

6.2.3.2.3.2. СВОЙСТВА КАНАЛА IEC104

Вид панели свойств:

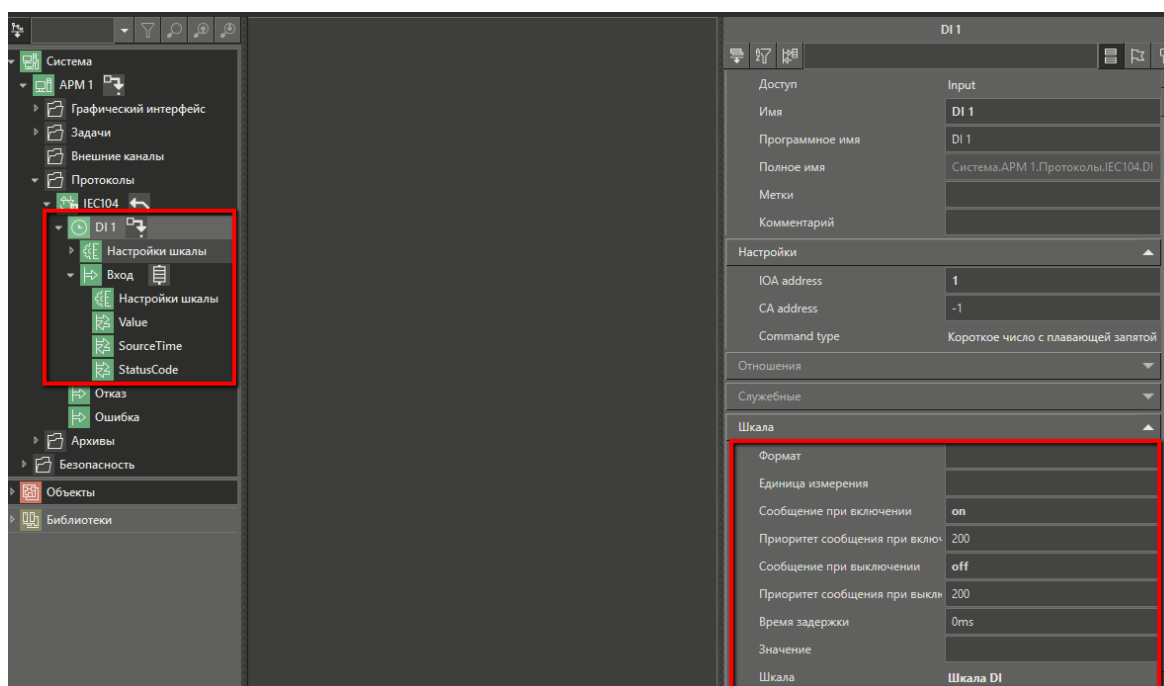
Описание:

Название	Рекомендации
Общие	Для чтобы дерево проекта выглядело информативным и понятным, рекомендуется изменить имя канала и установить тип доступа: Input – для получаемых значений, Output – для команд. Подробное описание данной категории приведено в соответствующем разделе
Шкала	Можно выбрать шкалу из ранее добавленных в библиотеку. Однако, как правило, у каналов шкала не настраивается, т.к. чаще всего каналы связывают с параметрами объектов или узлов, и шкалу настраивают уже у связанных параметров.

Название	Рекомендации
IOA address	Information object address - адрес переменной.
CA address	Адрес устройства или обособленной части устройства. Если CA=-1, то используется CA, указанный в настройках протокола CA. В некоторых источниках CA устройства называют также ASDU устройства.
Command type	<p>Тип команды, настраивается для выходных каналов. Согласно требованиям протокола поддерживаются следующие типы команд:</p> <div data-bbox="619 752 1366 969" style="background-color: #333; color: #eee; padding: 5px; border: 1px solid #555;"> <pre>45:C_SC_NA_1 Single 46:C_DC_NA_1 Double 47:C_RC_NA_1 Step 48:C_SE_NA_1 Set Point Normalized 49:C_SE_NB_1 Set Point Scaled 50:C_SE_NC_1 Set Point Short 51:C_BO_NA_1 Bitstring</pre> </div> <ul style="list-style-type: none"> • CON <45> - одноэлементная команда - C_SC_NA_1 • CON <46> - двухэлементная команда - C_DC_NA_1 • CON <47> - команда пошагового регулирования - C_RC_NA_1 • CON <48> - команда уставки, нормализованное значение - C_SE_NA_1 • CON <49> - команда уставки, масштабированное значение -C_SE_NB_1 • CON <50> - команда уставки, короткое число с плавающей запятой - C_SE_NC_1 • CON <51> - строка из 32 битов - C_BO_NA_1
ReadCmd102	Определяет возможность опроса канала командой 102. Если флаг установлен, то опрос канала будет выполняться командой <102>: C_RD_NA_1, если в настройках устройства включен такой опрос.
Duration Command	<p>Определяет длительность команды. Возможные варианты:</p> <div data-bbox="619 1805 1366 1939" style="background-color: #333; color: #eee; padding: 5px; border: 1px solid #555;"> <pre>0:no additional 1:short pulse 2:long pulse 3:persistent output</pre> </div> <ul style="list-style-type: none"> • 0 – нет дополнительного определения длительности выходного импульса;

Название	Рекомендации
	<ul style="list-style-type: none"> • 1 – короткий импульс (длительность определяется системным параметром на КП); • 2 – длинный импульс (длительность определяется системным параметром на КП); • 3 – постоянный выход;
Select or Execute	Определяет тип запроса.: TRUE - Select, False - Execute . При типе запроса Execute прибор сразу обрабатывает команду. При типе запросе Select прибор посылает запрос подтверждения, и только после подтверждения от сервера обрабатывает команду.

Если параметр канала имеет тип SYSTEM_BOOL_PARAM, и при этом каналу назначить шкалу (с генерацией сообщений при активации и/или деактивации), то сообщения будут формироваться по каждому полученному значению, независимо от периода опроса протокола.



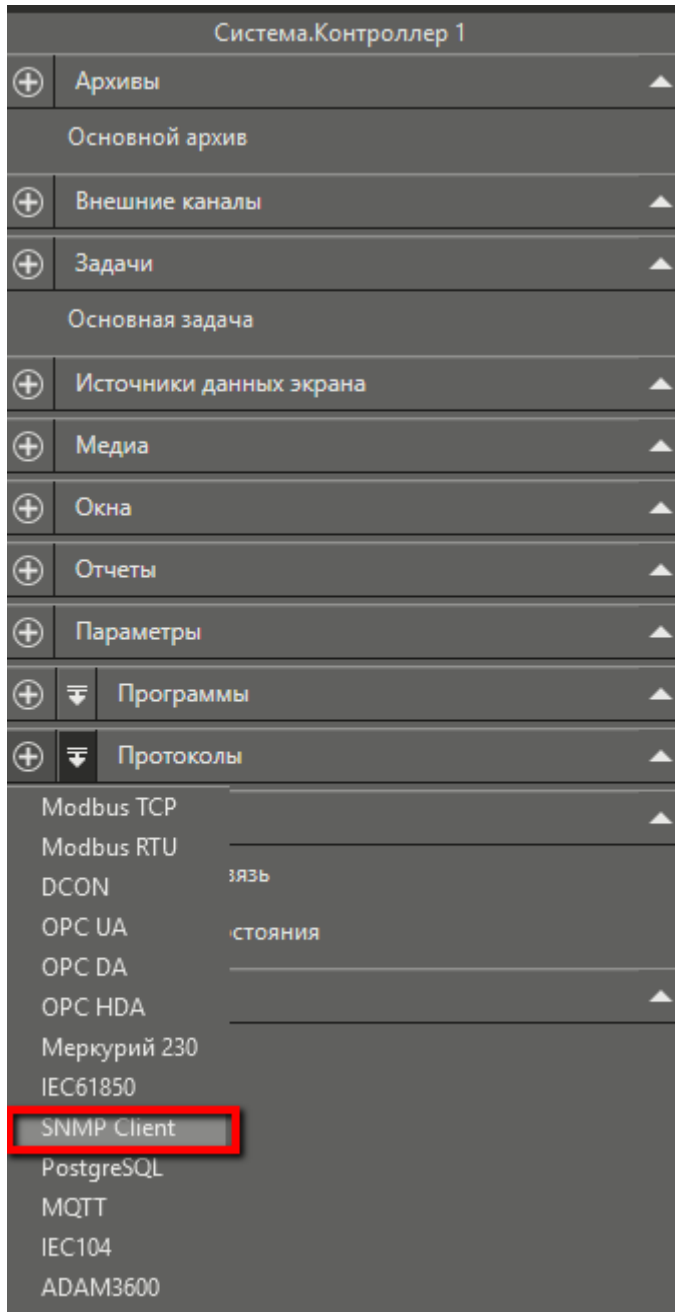
6.2.3.2.4. SNMP CLIENT

Simple Network Management Protocol — простой протокол сетевого управления.

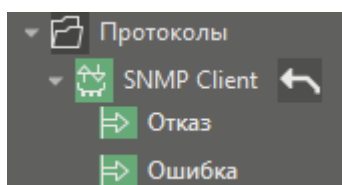
MasterSCADA 4D RT может получать/отправлять данные от устройств и других приложений по протоколу SNMP (версии 1 и 2). Это платформонезависимый протокол, который может использоваться в исполнительных системах, предназначенных для различных операционных систем.

Важно! По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте sales@masterscada.ru

Если необходимо получать данные по протоколу SNMP, то в узел, при помощи контекстного меню или контекстной панели, добавляется протокол SNMP Client:

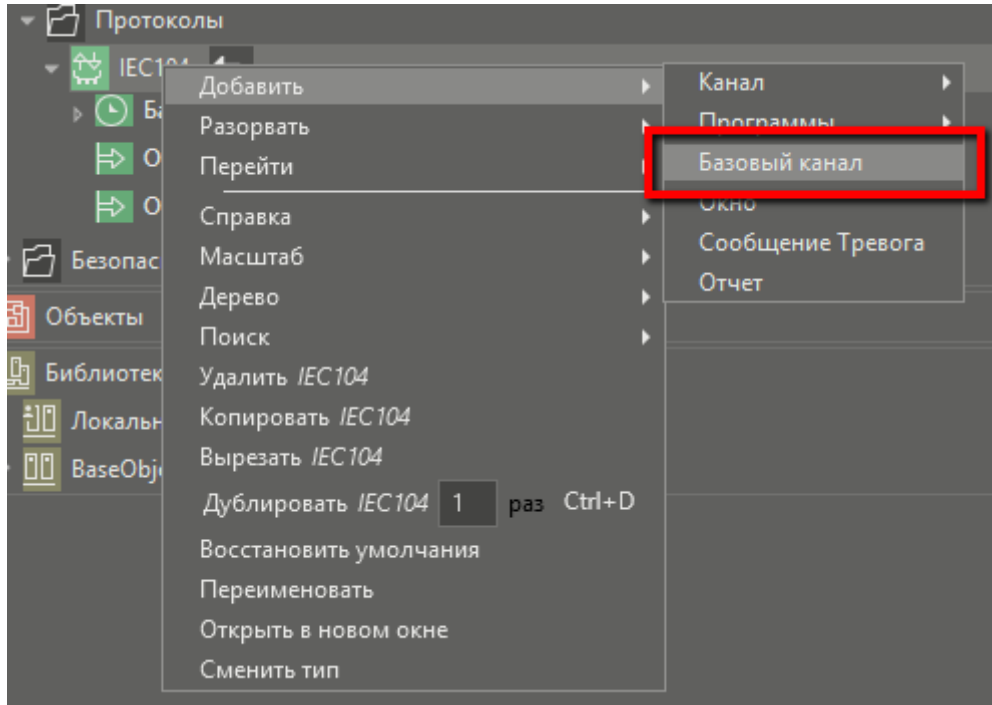


В результате получим:



Затем необходимо настроить панель свойств.

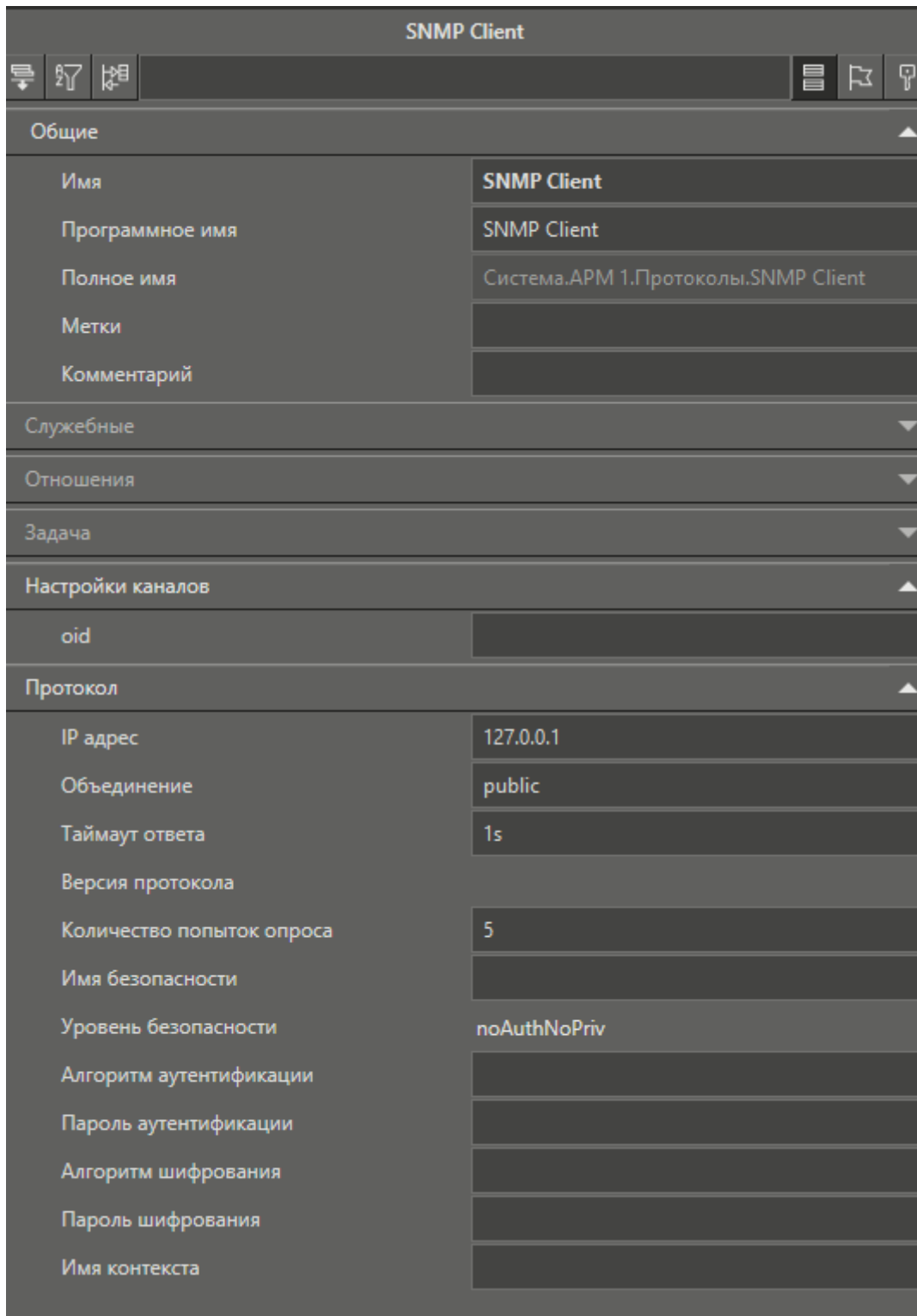
При помощи контекстного меню протокола можно добавить каналы, например, универсальный Базовый канал:



Затем настраивается панель свойств канала.

6.2.3.2.4.1. СВОЙСТВА SNMP CLIENT

Вид панели свойств протокола SNMP Client:



Название	Рекомендации	Версия
Категория Общие	Как правило, для данного элемента эта категория не настраивается. Подробное	Все

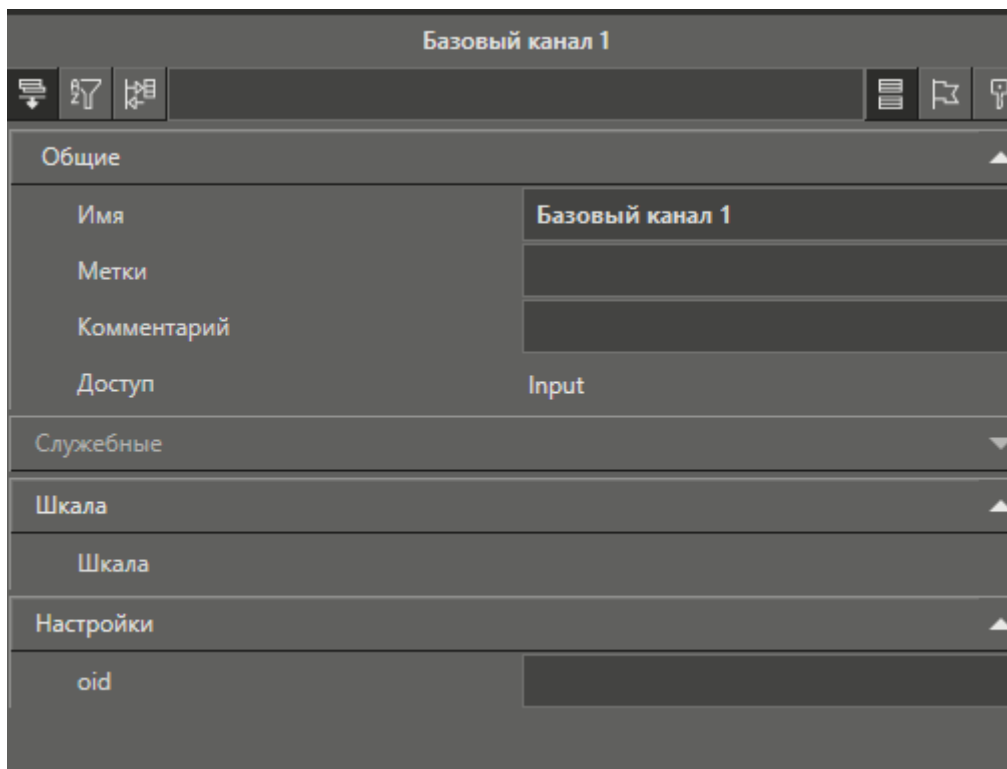
Название	Рекомендации	Версия
	описание данной категории приведено в соответствующем разделе.	
Категория Задача	<p>Задается период, с которым MasterSCADA 4D будет стараться получать значения из сервера.</p> <p>Определяется способ работы протокола при резервировании узла. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи у протоколов.</p>	Все
Настройки каналов	Задаются значения по умолчанию для создаваемых каналов. В дальнейшем эти настройки у каналов можно будет изменить.	Все
IP-адрес	<p>IP-адрес сервера, который отправляет и получает данные от MasterSCADA 4D.</p> <p>При необходимости можно указать порт (формат записи ip:port), а также тип используемого протокола - tcp/udp (формта записи tcp::ip:port, udp::ip:port)</p>	Все
Объединение	Настройка SNMP Community: как правило, для чтения – public, для записи – private.	Все
Таймаут ответа	Время ожидания отклика от устройства.	Все
Версия протокола	Задается версия используемого протокола.	Все
Количество попыток опроса	Задается алгоритм, по которому выход протокола Отказ принимает значение TRUE. Если опрос не успешен после	Все

Название	Рекомендации	Версия
	исчерпания заданного числа попыток, то выдается отказ протокола.	
Имя безопасности	Задается имя пользователя, используемое при подключении к устройству.	v3
Уровень безопасности	<p>Задается уровень безопасности работы с устройством. Существует три варианта:</p> <ul style="list-style-type: none"> • oNoAuthNoPriv - без аутентификации и без шифрования. Используется только имя пользователя. • oauthNoPriv - с аутентификацией, но без шифрования. Для аутентификации нужен пароль, но обмен данными не шифруется. • oauthPriv - с аутентификацией и с шифрованием. 	v3
Алгоритм аутентификации	<p>Задается алгоритм хэширования пароля аутентификации. Варианты:</p> <ul style="list-style-type: none"> • MD5 • SHA 	v3
Пароль аутентификации	Устанавливается пароль, используемый при подключении к устройству.	v3
Алгоритм шифрования	<p>Задается алгоритм шифрования передаваемых данных. Варианты:</p> <ul style="list-style-type: none"> • DES • AES128 	v3

Название	Рекомендации	Версия
Пароль шифрования	Устанавливается пароль шифрования передаваемых данных.	v3
Имя контекста	Задается набор доступной информации управления по объекту SNMP.	v3

6.2.3.2.4.2. СВОЙСТВА КАНАЛА SNMP

Вид панели свойств:



Описание:

Название	Рекомендации
Общие	Для того чтобы дерево проекта выглядело информативным и понятным, рекомендуется изменить имя канала и установить тип доступа: Input – для получаемых значений, Output – для команд. Подробное описание данной категории приведено в соответствующем разделе

Шкала	Можно выбрать шкалу из ранее добавленных в библиотеку. Однако, как правило, у каналов шкала не настраивается, т.к. чаще всего каналы связывают с параметрами объектов или узлов, и шкалу настраивают уже у связанных параметров.
OID	Задается идентификатор объекта, с которого необходимо получать данные

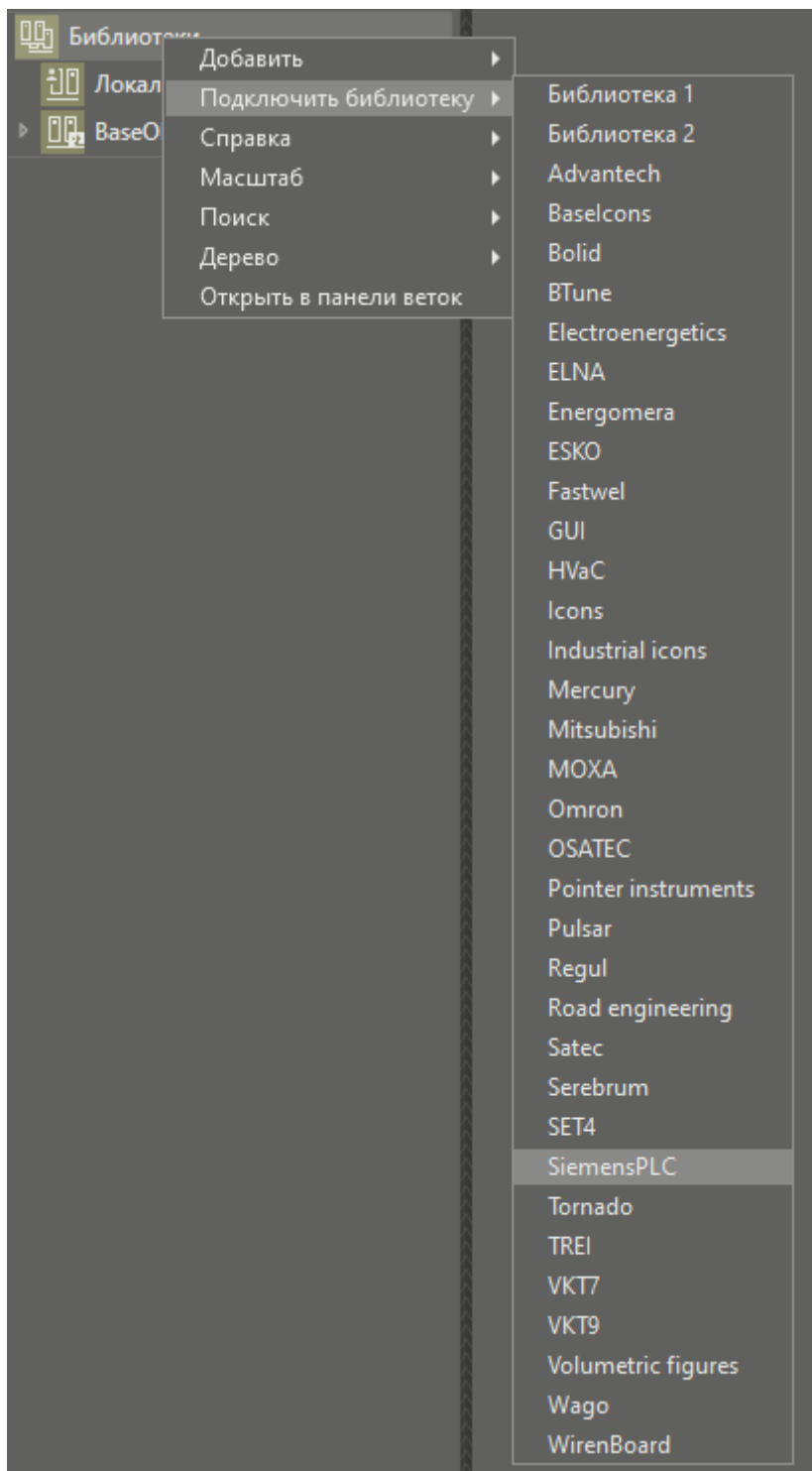
Важно! В текущей версии автоматической загрузки каналов из mib-файлов не предусмотрено. Идентификаторы (адреса) задаются вручную.

6.2.3.2.5. SIEMENSPLC

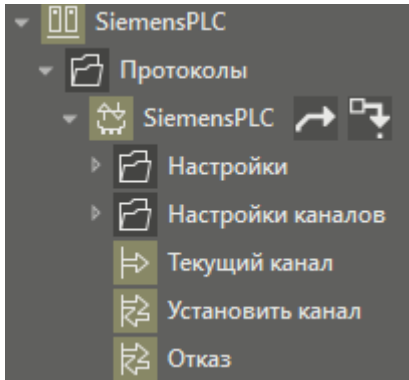
MasterSCADA 4D RT поддерживает работу по протоколу Profinet, который в среде разработки называется SiemensPLC и предназначен для опроса контроллеров фирмы Siemens серии S7 – S7-200, S7-300, S7-400, S7-1200, S7-1500. В качестве интерфейса SiemensPLC использует сети, базирующиеся на TCP/IP - Ethernet, Wi-Fi.

Важно! По умолчанию протокол не входит в состав исполнительной системы. Стоимость и условия лицензирования этого протокола необходимо уточнять по электронной почте sales@masterscada.ru

Для получения возможности использования протокола в проекте необходимо подключить соответствующую библиотеку:



В полном дереве библиотека SiemensPLC имеет вид:

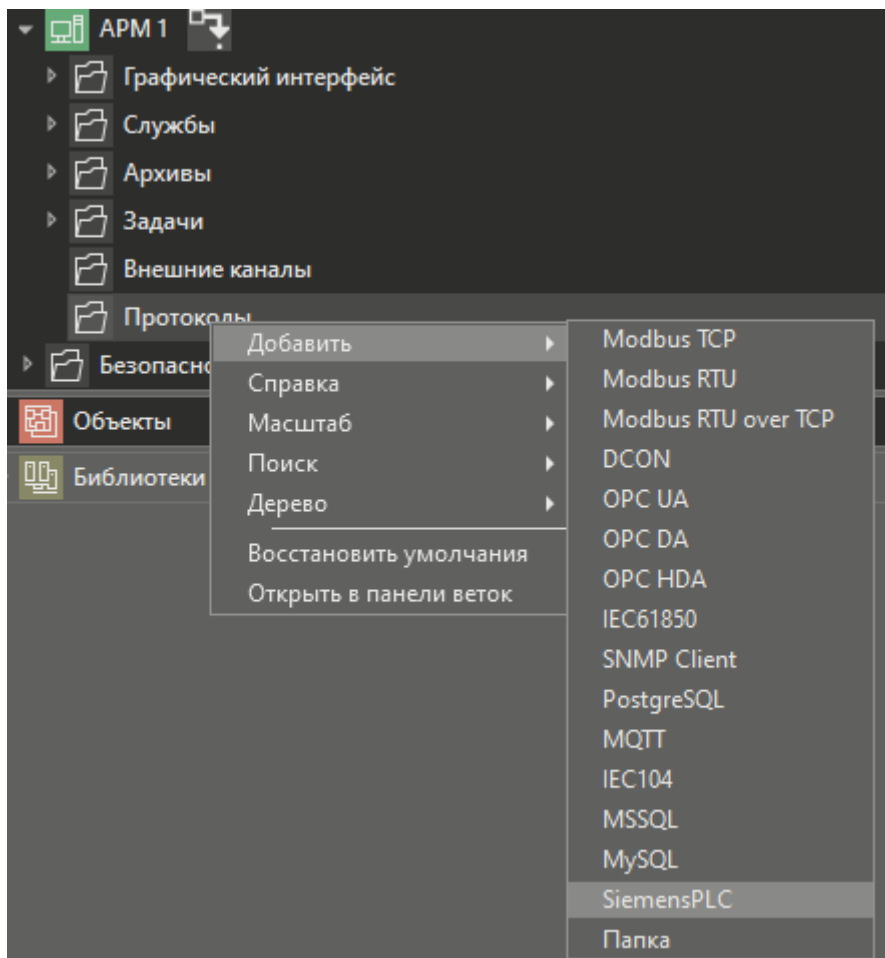


Назначение параметров

Название	Назначение
Текущий канал	Текущий номер резервного канала (0 - основной канал)
Установить канал	Позволяет установить номер резервного канала
Отказ	Формируется отказ протокола, если исполнительная система не имеет связи с опрашиваемым контроллером.

После этого в проект может быть добавлен протокол любым известным способом, например, при помощи контекстного меню узла, или с использованием контекстной панели, или путем перетаскивания элемента из библиотеки в дерево системы.

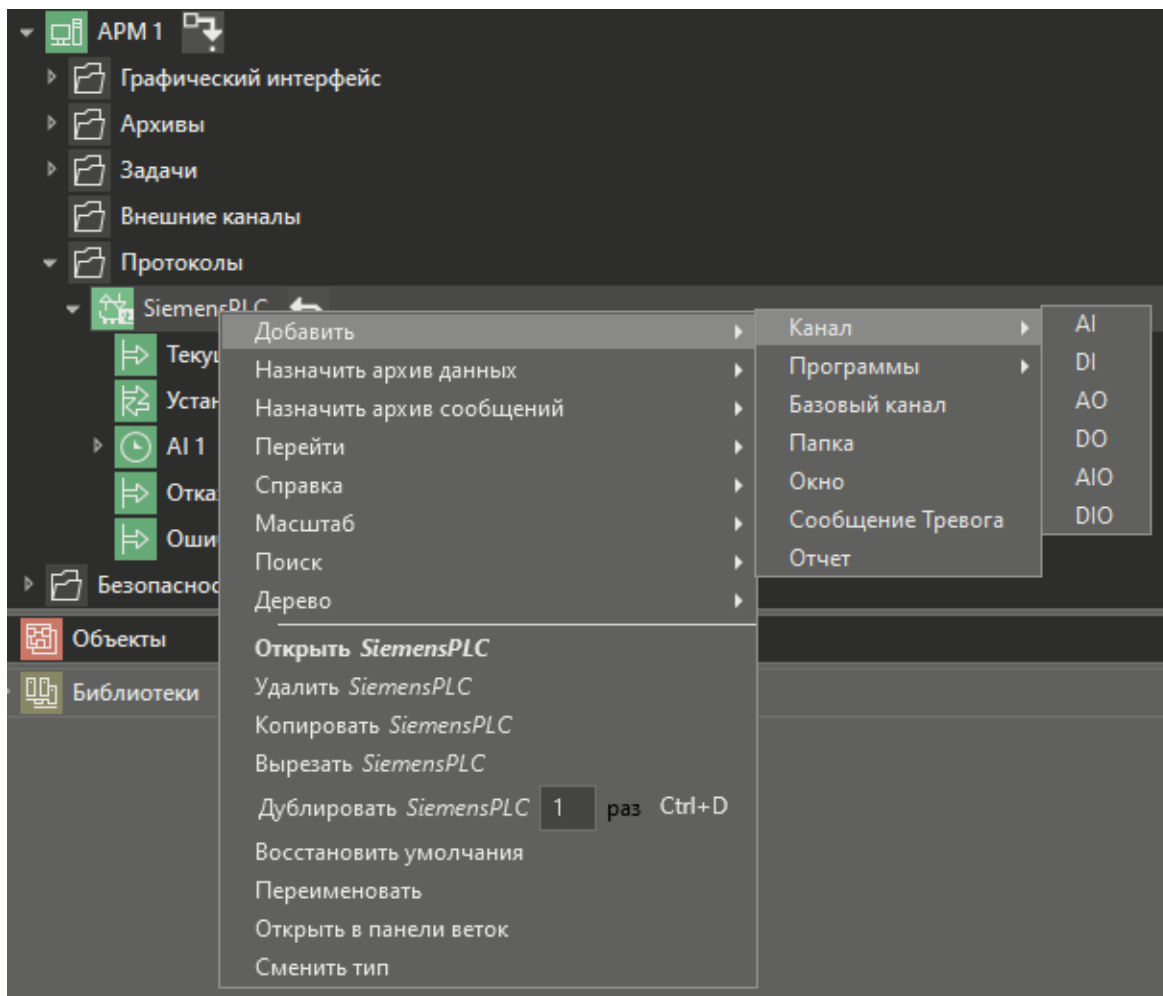
Рекомендуется добавлять элементы в проект при помощи контекстного меню:



Затем необходимо настроить панель свойств протокола.

Далее требуется добавить каналы в протокол. Сделать это можно вручную, либо импортировать из файла.

Добавить канал вручную можно при помощи контекстного меню протокола:



Далее необходимо настроить панель свойств канала, ориентируясь на его свойства, заданные в приложении TIA Portal.

Для автоматического добавления каналов нужно дважды нажать левую кнопку мыши по протоколу в дереве. При этом откроется вкладка, в которой можно выполнить импорт каналов.

6.2.3.2.5.1. СВОЙСТВА SIEMENSPLC

Вид панели свойств протокола SiemensPLC:

SiemensPLC

🔍
🔗
☰
🏠
🔑

Общие ▲

Имя	SiemensPLC
Полное имя	Система.APM 1.Протоколы.SiemensPLC
Метки	
Комментарий	

Служебные ▼

Задача ▼

Отношения ▼

Настройки каналов ▼

Протокол ▲

IP адрес	127.0.0.1
Порт	102
Таймаут	3000
Монтажная стойка	0
Слот процессора	1
Тип соединения ISOTCP243	<input type="checkbox"/>
Использовать резервирование	<input type="checkbox"/>
Выставлять Отказ при переключении	<input type="checkbox"/>
IP резерв 1	
IP резерв 2	
Монтажная стойка резерв 1	-1
Монтажная стойка резерв 2	-1
Слот процессора резерв 1	-1
Слот процессора резерв 2	-1

Название	Рекомендации
Категория Общие	<p>Как правило, для данного элемента эта категория не настраивается. Подробное описание данной категории приведено в соответствующем разделе</p>

Название	Рекомендации
Категория Задача	Задается период, с которым MasterSCADA 4D будет стараться получать значения из сервера. Определяется способ работы протокола при резервировании узла. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи у протоколов.
Категория Настройки каналов	Задаются значения по умолчанию для создаваемых каналов. В настройке не нуждаются, т.к. индивидуальные настройки каналов формируются автоматически.
IP-адрес	Основной IP-адрес устройства, которое отправляет и получает данные от MasterSCADA 4D. Важно! Адрес контроллеров Siemens по умолчанию 192.168.0.1 – рекомендуем изменять данный адрес в контроллере. Этот адрес также используется по умолчанию многими маршрутизаторами, что может приводить к конфликтам сети.
TCP-порт	Номер TCP-порта устройства, по которому осуществляется подключение.
Таймаут	Время, в течение которого исполнительная система ожидает установления соединения с контроллером
Монтажная стойка	Номер рейта (Rack) контроллера.
Слот процессора	Слот рейта (Slot), в котором установлен опрашиваемый процессор.
Тип соединения ISOTCP243	Тип поддерживаемого контроллером протокола
Использовать резервирование	Включает функцию резервирования каналов связи. В протоколе SiemensPLC существует возможность задания трех резервных каналов подключения к контроллеру. Переключение на резервный канал может осуществляться двумя способами: <ul style="list-style-type: none"> • вручную, для этого нужно указать номер канала в параметре протокола Установить канал

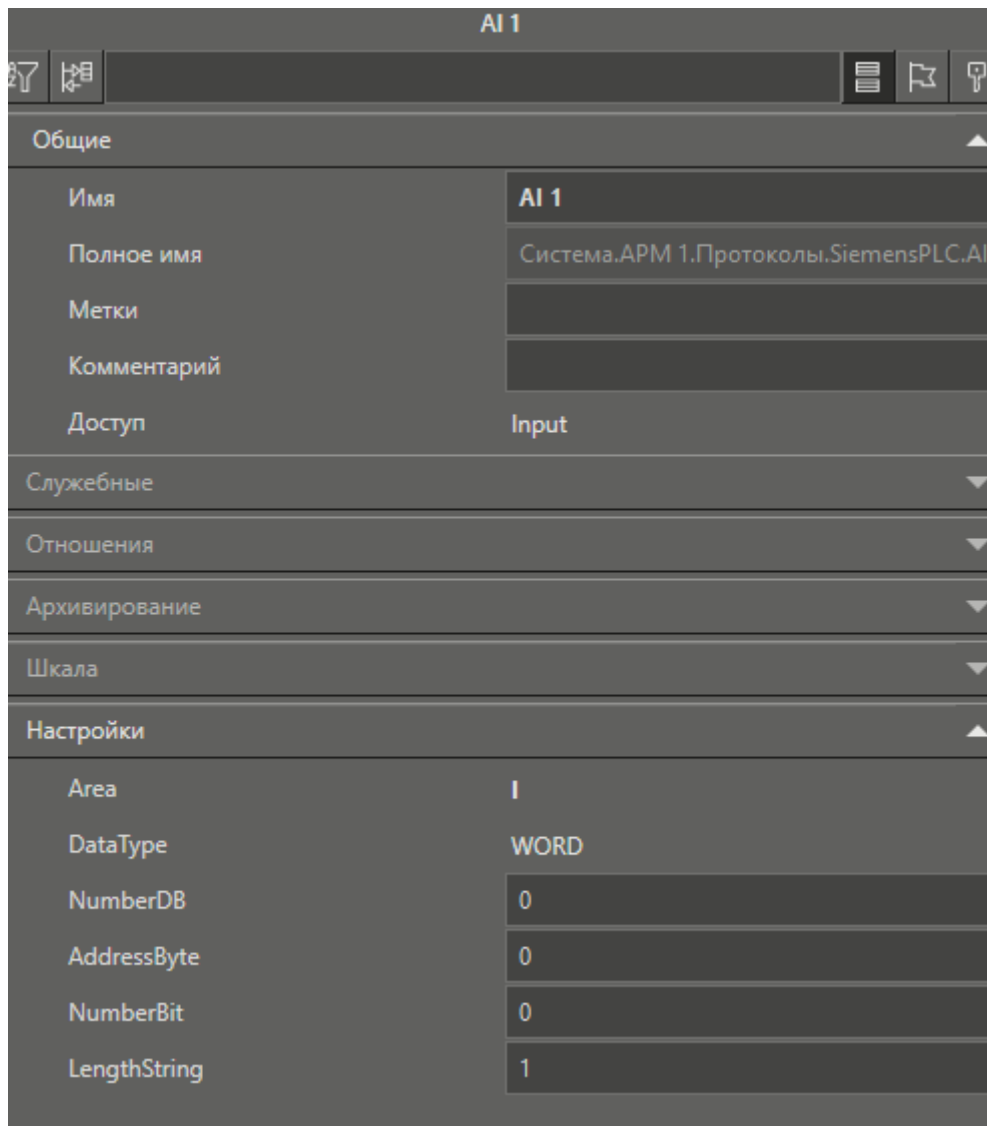
Название	Рекомендации
	<ul style="list-style-type: none"> автоматически, когда исполнительная система отслеживает состояние связи с основным каналом, и если связь пропадает происходит переключение на резервный канал 1, в случае отказа резервного канала 1 - переключение на канал 2 и т.д по кругу. Автоматического возврата на основной канал при его восстановлении не происходит - это необходимо сделать вручную
Выставлять Отказ при переключении	Определяет - нужно ли выставлять "плохой" признак качества каналам, пока происходит переключение на резервный канал.
IP резерв 1	Первый резервный IP-адрес устройства, которое отправляет и получает данные от MasterSCADA 4D.
IP резерв 2	Второй резервный IP-адрес устройства, которое отправляет и получает данные от MasterSCADA 4D.
Монтажная стойка резерв 1	Номер крейта (Rack) резервного канала 1. Значение по умолчанию -1 - в этом режиме значение берется от настройки основного канала
Монтажная стойка резерв 2	Номер крейта (Rack) резервного канала 2. Значение по умолчанию -1 - в этом режиме значение берется от настройки основного канала
Слот процессора резерв 1	Слот крейта (Slot), в котором уставлен опрашиваемый процессор резервного канала 1. Значение по умолчанию -1 - в этом режиме значение берется от настройки основного канала.
Слот процессора резерв 1	Слот крейта (Slot), в котором уставлен опрашиваемый процессор резервного канала 2. Значение по умолчанию -1 - в этом режиме значение берется от настройки основного канала.

Важно! Монтажная стойка и слот процессора у резервного канала задаются если используется резервирование процессорных модулей. Если резервируются только каналы связи с контроллером, то можно оставить значения данных параметров по умолчанию (-1).

6.2.3.2.5.2. СВОЙСТВА КАНАЛОВ SIEMENSPLC

Если канал протокола добавляется вручную, то необходимо настроить панель свойств.

Вид панели свойств:



Описание основных свойств

Название	Описание
Area	<p>Радается регион опрашиваемого канала. Регион – сегмент памяти, в которой расположена переменная. Существуют следующие варианты:</p> <ul style="list-style-type: none"> • I – входы. • Q – выходы. • M – ячейки памяти.

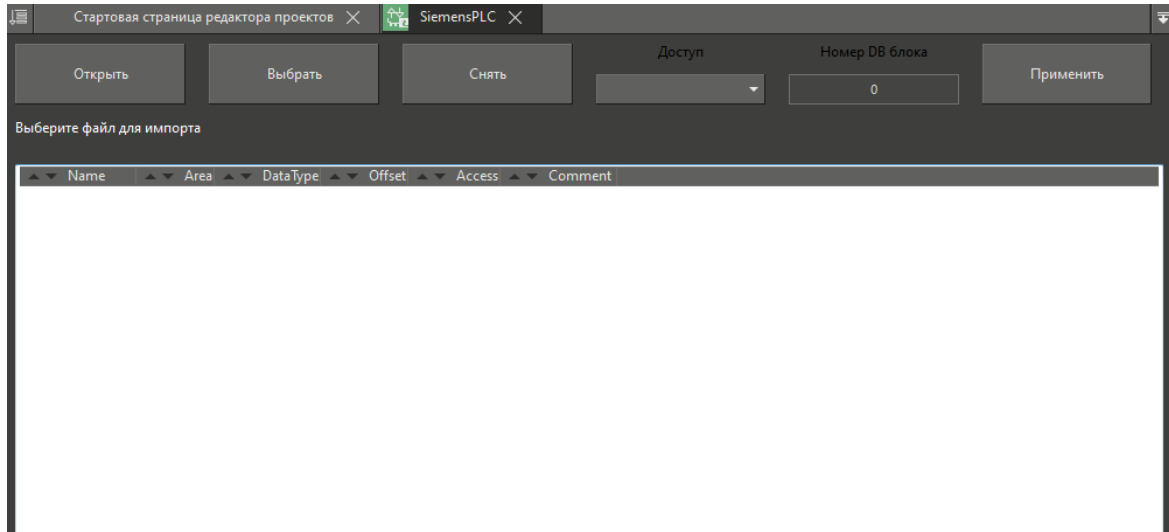
Название	Описание
	<ul style="list-style-type: none"> • DB – блоки данных (DataBlock). При этом у DataBlock должна быть выключена настройка Optimized block access. • T – таймеры (обратный отсчет оставшегося времени, в миллисекундах). • C – счетчики (накопленное значение).
DataType	Задается тип данных опрашиваемой переменной в контроллере. Значение подбирается в зависимости от установленного типа данных в контроллере.
NumberDB	Настройка присутствует только у региона DB. В данном поле необходимо указать, какой номер имеет Data Block, в котором находится опрашиваемая переменная.
AddressByte	Адрес стартового байта, в котором находится опрашиваемая переменная
NumberBit	Определяет извлекаемый номер бита, если тип данных установлен в BOOL
LengthString	Определяет максимальную длину строки

Описание остальных свойств смотрите в соответствующих разделах

6.2.3.2.5.3. ДИАЛОГ ИМПОРТА КАНАЛОВ SIEMENSPLC

Для автоматического добавления каналов в протокол SiemensPLC нужно дважды нажать левую кнопку мыши по протоколу в дереве. При этом откроется вкладка, в которой можно выполнить импорт каналов. Импортировать можно как переменные из таблицы PLC Tags, так и переменные из Data Block.

Вид вкладки по умолчанию:



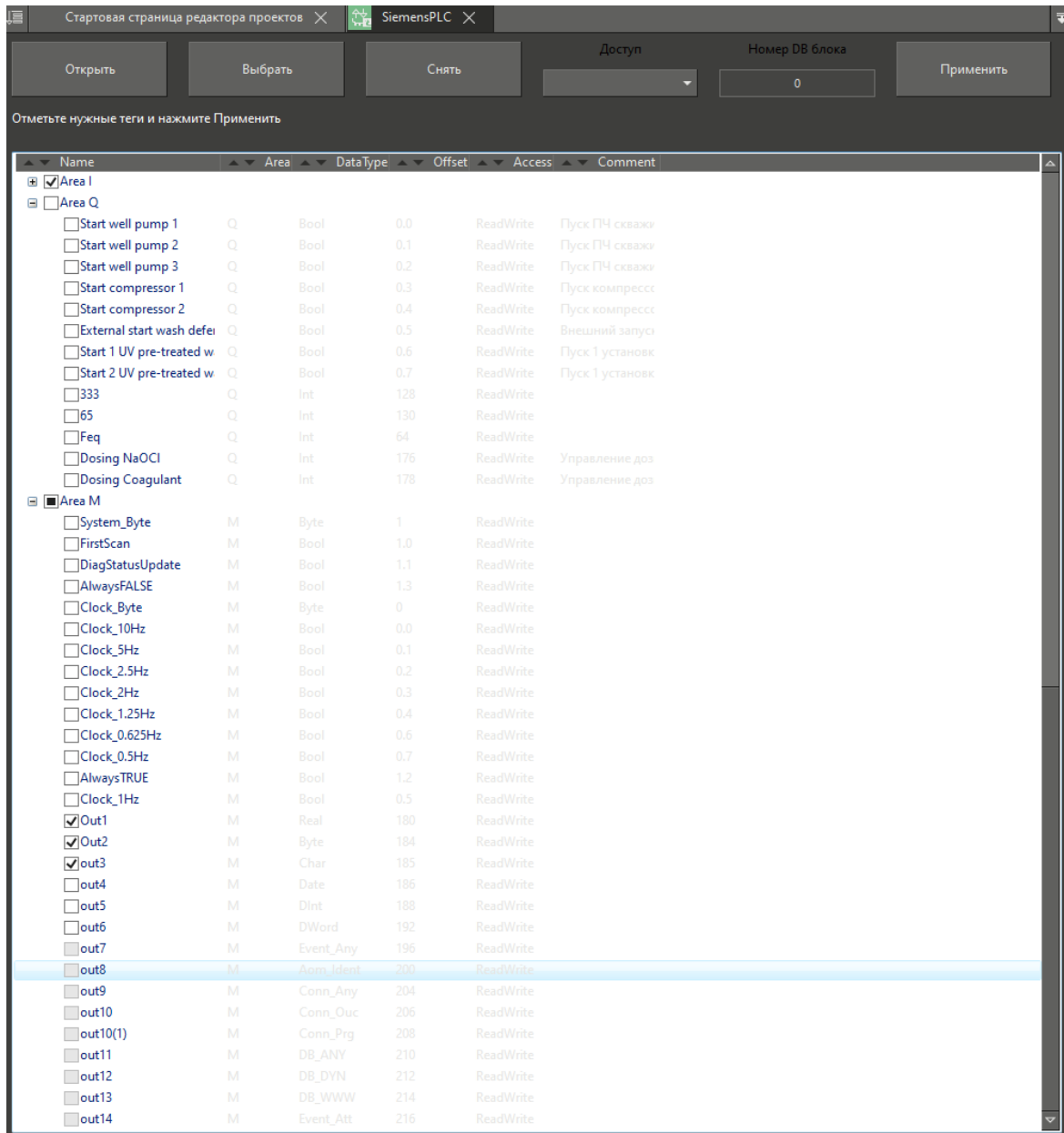
Описание элементов управления:

Элемент	Описание
Открыть	Открывает диалоговое окно Windows, в котором можно выбрать файл, из которого нужно выполнить импорт данных. Позволяет открыть файлы Siemens Export File (*.db, *.xlsx). Для открытия файла типа *.xlsx, на компьютере должен быть установлен MS Office. После того как файл будет открыт в таблице, расположенной на данной вкладке появится список каналов.
Выбрать	Если выделить группу каналов или только один и нажать кнопку Выбрать. То напротив этих каналов установится флаг. Флаг также можно установить/снять нажав по нему в поле таблицы
Снять	Если выделить группу каналов или только один, из тех которые ранее были отмечены флагами в таблице и нажать кнопку Снять. То напротив этих каналов флаг будет снят. Флаг также можно установить/снять нажав по нему в поле таблицы
Доступ	Определяет доступ каналов, полученных из файлов *.db
Номер DB блока	Указывается номер DB блока.

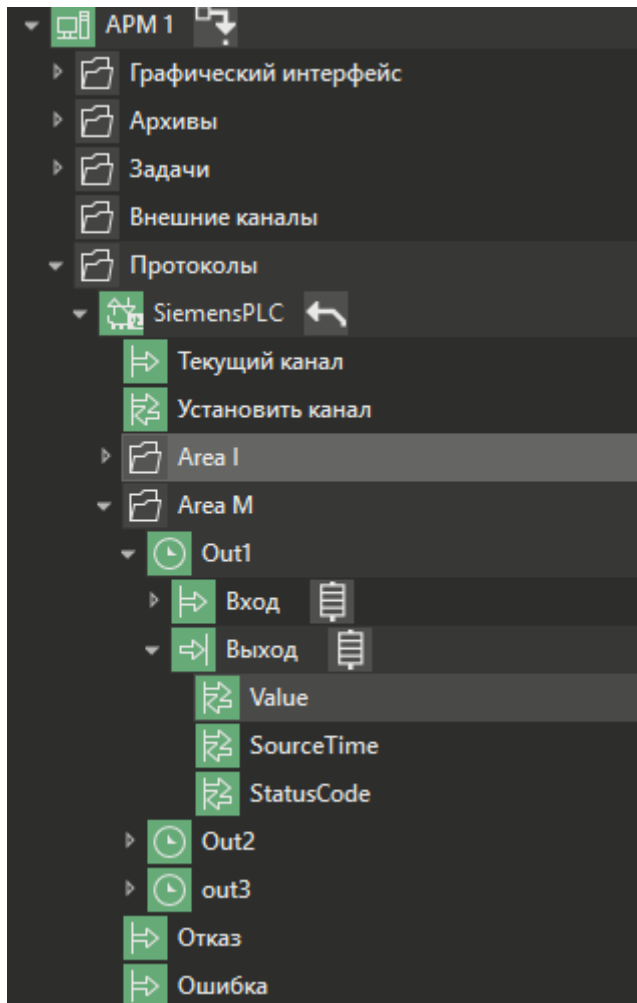
Применить

После нажатия на кнопку Применить отмеченные флагами каналы будут добавлены в дерево системы MasterSCADA 4D

Пример вида вкладки после открытия файла с переменными:



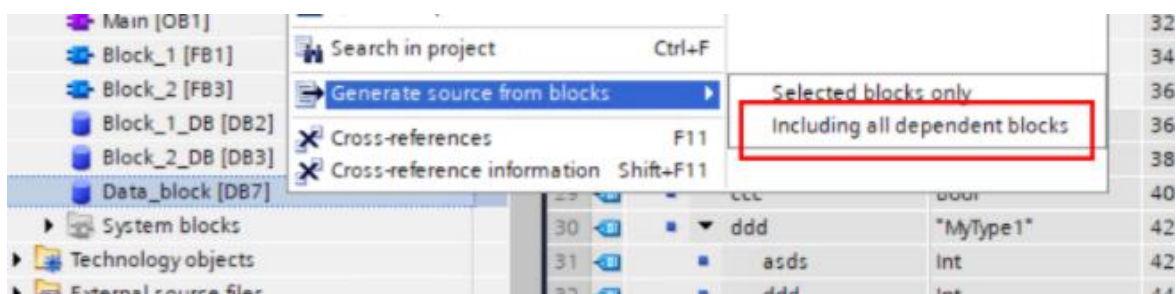
Вид дерева системы после нажатия на кнопку Применить:



Импорт тегов может занять некоторое время.

Работа с пользовательскими типами

В протокол могут быть добавлены пользовательские типы данных - структуры, которые были созданы Tia Portal в PLC Data Types. Для их генерации в Tia Portal нужно нажать:



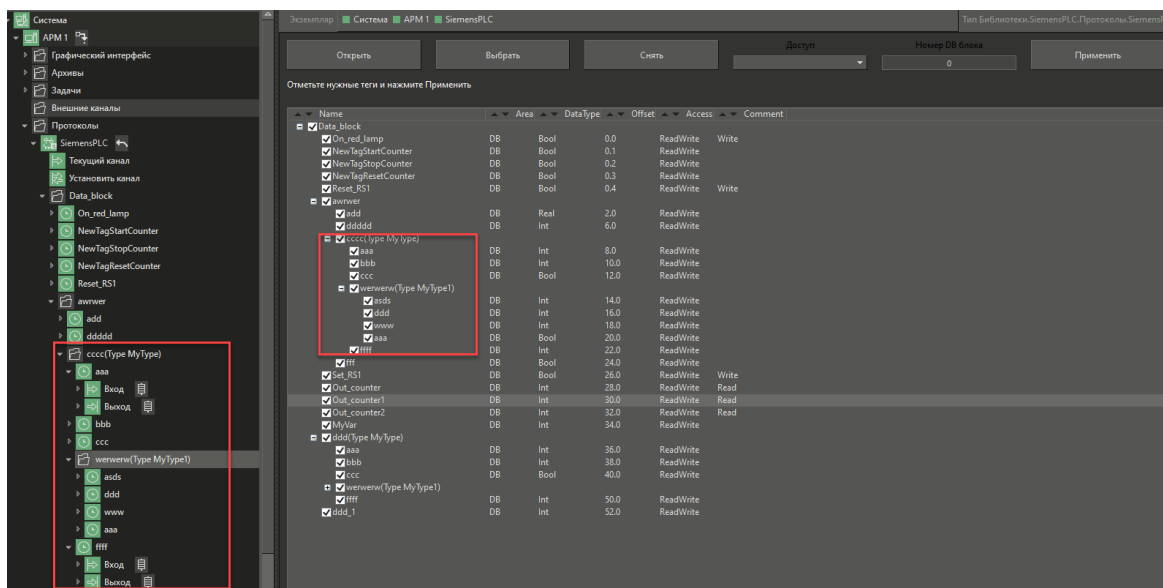
В начале файла будут добавлены все использованные структуры.

Вариант, когда структуры сохраняются отдельными файлами - не поддерживается. Однако разработчик может сам открыть файл *Data_block.db*, и добавить эти структуры в начало файла.

Также не допускается делать генерацию файла по нескольким Data_Block сразу, необходимо делать для каждого отдельно.

При обработке пользовательских структур не анализируется корректность типа данных. Если что-то будет указано некорректно, то MasterSCADA 4D отобразит в ошибке номер строки не с реальной ошибкой, а с номером строки DB блока, где находится структура.

После добавления в дерево проекта каждое поле структуры будет иметь вид отдельного канала, в папке с именем структуры:

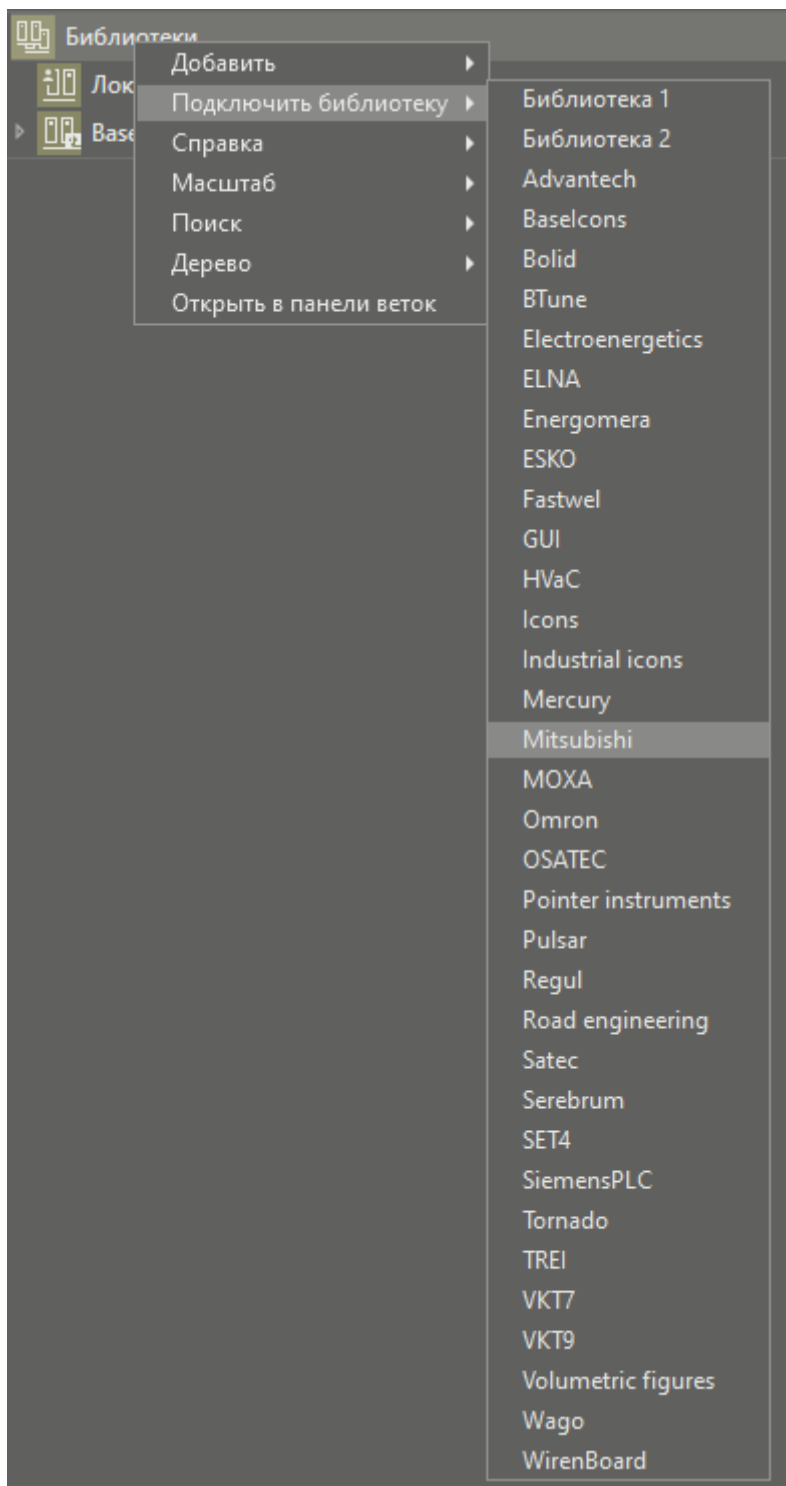


6.2.3.2.6. MITSUBISHI

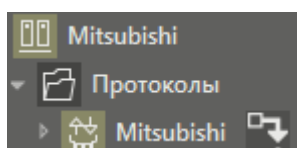
MasterSCADA 4D поддерживает работу по открытому протоколу промышленных систем SLMP, разработанный японской корпорацией Mitsubishi для программируемых контроллеров серии Melsec. В качестве интерфейса SLMP использует сети базирующиеся на TCP/IP и UDP - Ethernet, Wi-Fi, GPRS.

Важно! По умолчанию протокол не входит в состав исполнительной системы. Стоимость и условия лицензирования этого протокола необходимо уточнять по электронной почте sales@masterscada.ru

Для получения возможности использования протокола в проекте необходимо подключить соответствующую библиотеку:

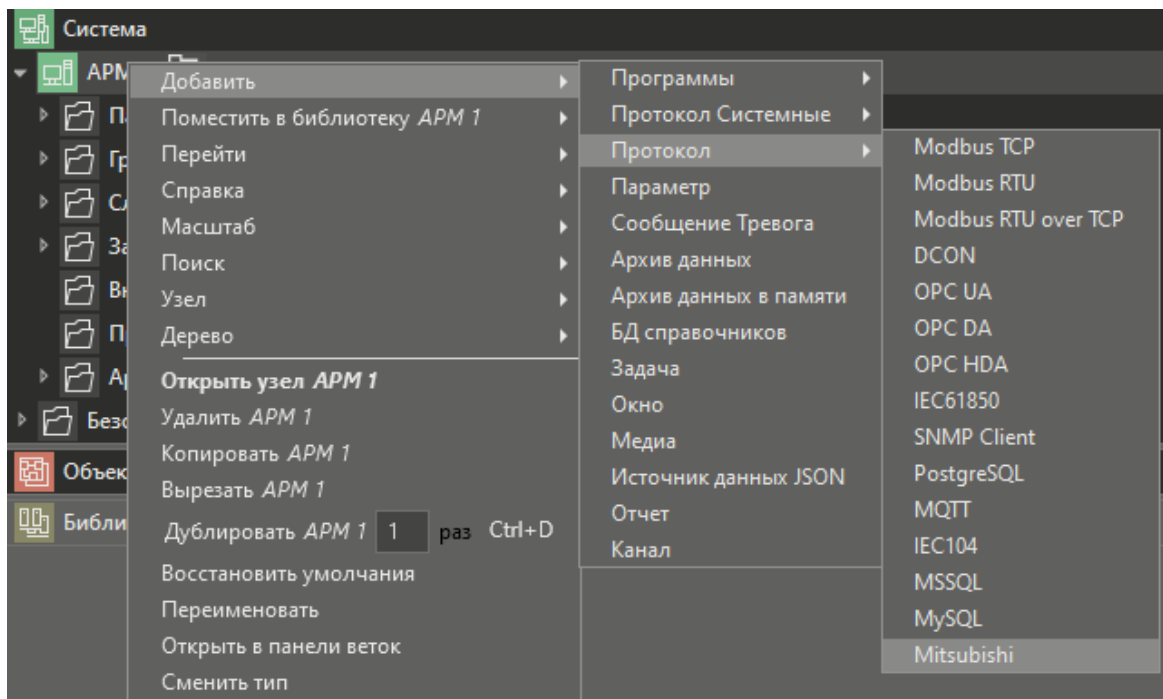


В полном дереве библиотека Mitsubishi имеет вид:



После этого в проект может быть добавлен протокол из библиотеки любым известным способом, например, при помощи контекстного меню узла, или с использованием контекстной панели, или путем перетаскивания элемента из библиотеки в дерево системы.

Рекомендуется добавлять элементы в проект при помощи контекстного меню:



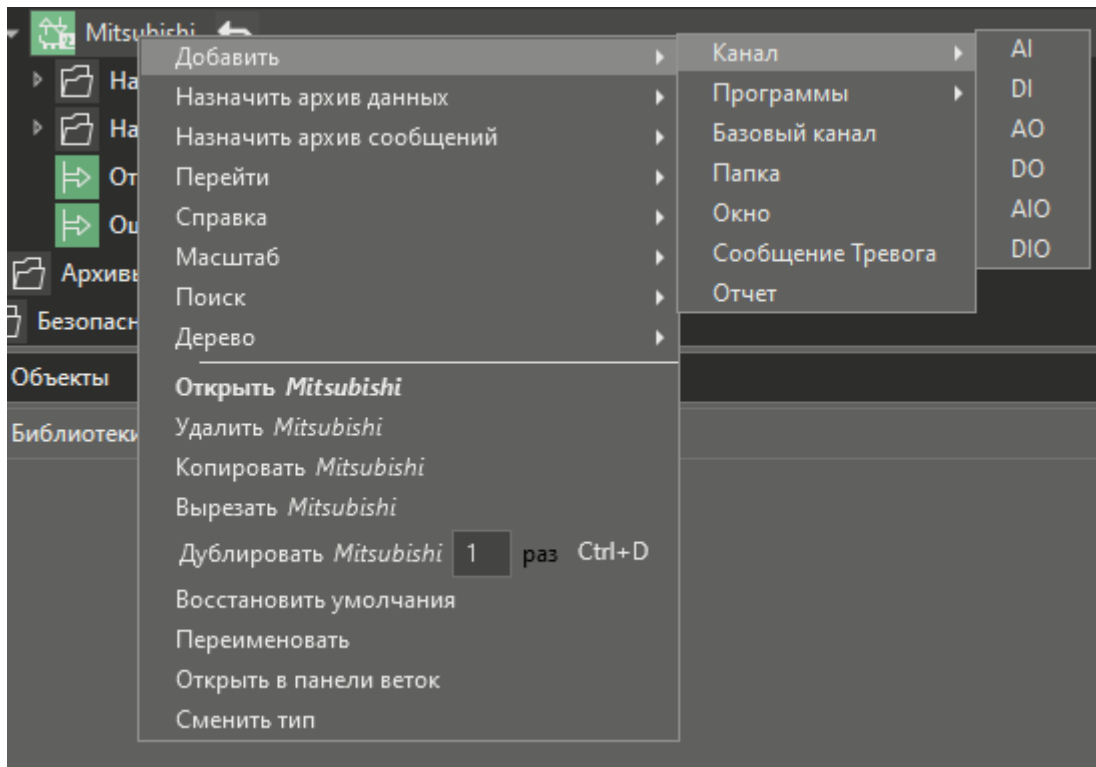
Затем необходимо настроить панель свойств протокола.

Протокол Mitsubishi имеет выход Отказ

Протокол Mitsubishi имеет выход Ошибка, который принимает текстовое значение в зависимости от ошибки, возникшей при работе с устройством.

Далее требуется добавить каналы в протокол. Сделать это можно вручную, либо импортировать из файла.

Добавить канал вручную можно при помощи контекстного меню протокола:



Далее необходимо настроить панель свойств канала, ориентируясь на его свойства, заданные в приложении GX Works 3.

Для автоматического добавления каналов нужно дважды нажать левую кнопку мыши по протоколу в дереве. При этом откроется вкладка, в которой можно выполнить импорт каналов.

6.2.3.2.6.1. СВОЙСТВА MITSUBISHI

Вид панели свойств:

Mitsubishi

Общие

Имя	Mitsubishi
Полное имя	Система.АРМ 1.Протоколы.Mitsubishi
Метки	
Комментарий	

Служебные

Задача

Отношения

Настройки каналов

Протокол

IP адрес	127.0.0.1
Порт	5000
Обмен по UDP	<input type="checkbox"/>
Таймаут	1000
Количество попыток	3
Контроллер FX3	<input type="checkbox"/>
Количество регистров в запросе	50
Destination Network	0
Destination Stantion	255
Destination Module	1023
Максимально допустимый разрыв в запросе чтения	0

Название	Рекомендации
Категория Общие	Как правило, для данного элемента эта категория не настраивается. Подробное описание данной категории приведено в соответствующем разделе
Категория Задача	Задается период, с которым MasterSCADA 4D будет стараться получать значения из сервера. Определяется способ работы протокола при резервировании узла. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи у протоколов.
Категория Настройки каналов	Задаются значения по умолчанию для создаваемых каналов. В настройке не нуждаются, т.к. индивидуальные настройки каналов формируются автоматически.

Название	Рекомендации
Категория Протокол	
IP-адрес	Основной IP-адрес устройства, которое отправляет и получает данные от MasterSCADA 4D. См. раздел Настройка контроллера Mitsubishi
Порт	Номер TCP-порта устройства, по которому осуществляется подключение.
Обмен по UDP	Определяет возможность обмена данными по UDP. См. раздел Настройка контроллера Mitsubishi
Таймаут	Время, в течение которого исполнительная система ожидает установления соединения с контроллером
Количество попыток	<p>Указывается предельное количество неудачных попыток получить ответ от контроллера, произошедших подряд. Если за указанное количество попыток получить ответ не удалось, то сформируется сигнал TRUE у параметра Отказ протокола</p> <p>Если при обращении к контроллеру он не ответит ни разу за указанное количество попыток подряд, то следующие запросы в рамках данного цикла опроса посылаться не будут, сформируется отказ модуля и выставится признак недостоверности по всем входам протокола. Если при запросе ответ придет, но с ошибкой, то сформируется отказ и выставится признак недостоверности по входам в данном запросе, но посылка других запросов по этому контроллеру продолжится.</p>
Контроллер FX3	Определяется тип контроллера. Если установлен флаг, то будет использоваться старая версия протокола (для контроллера FX3), если флаг снят, то новая версия протокола (для FX5)
Количество регистров в запросе	Определяет максимальное количество регистров в одном запросе чтения. Значение по умолчанию - 50. Только для контроллера FX3.

Название	Рекомендации
Destination Network	Указывается адрес удаленной сети. По умолчанию - 0. Только для контроллера FX5.
Destination Station	Указывается адрес удаленного устройства. По умолчанию - 255 (0xFF). Только для контроллера FX5.
Destination Module	Указывается адрес удаленного модуля (станция, CPU, резервированный CPU и т.д.). Адрес по умолчанию - 1023 (0x03FF - Own Station). Только для контроллера FX5.
Максимально допустимый разрыв в запросе чтения	Параметр определяет максимально допустимый разрыв между адресами в запросе чтения, при котором происходит объединение в один запрос. Например, производится опрос регистров с номерами - 0, 1, 2, 4, 5. Если параметр будет установлен в 0, то регистры будут запрошены за два запроса (0-2, 4-5), если же установить его в 1 или выше, то за один запрос (0-5, а не использованный регистр 3 будет отброшен при разборе ответа). Увеличение данного параметра сокращает количество запросов и ускоряет опрос, но удлиняет кадр ответа.

Важно! Настройки Destination Network, Station и Module по умолчанию установлены для прямого опроса моноблочного ПЛК, без резервирования, в остальных случаях может потребоваться настройка данных параметров. Значения параметров можно найти в официальной документации к протоколу SLMP.

6.2.3.2.6.2. СВОЙСТВА КАНАЛОВ MITSUBISHI

Если канал протокола добавляется вручную, то необходимо настроить панель свойств:

AI 1

🔍 📄
☰ 🚩 🔑

Общие ▲

Имя	AI 1
Полное имя	Система.APM 1.Протоколы.Mitsubishi.AI 1
Метки	
Комментарий	
Доступ	Input

Служебные ▼

Отношения ▼

Настройки ▲

MemoryArea	Data register (D)-Word
InterpretAs	WORD
MemoryAddress	0
BitAddress	0

Архивирование ▲

Архивировать	<input type="checkbox"/>
Шаблон архивирования	

Шкала ▲

Шкала	
-------	--

Описание основных свойств

Название	Описание
MemoryArea	Задается область памяти опрашиваемой переменной
InterpretAs	Тип данных переменной.
MemoryAddress	<p style="text-align: center;">Адрес переменной в области памяти.</p> <p>Важно! Примечание. Для регионов Input (X) и Output(Y) в контроллере используется восьмеричная система исчисления. Т.е. нужно указывать номера в соответствии с их номерами на контроллере (X0-X7,X10-X17 и т.д.). Если адрес регистра будет</p>

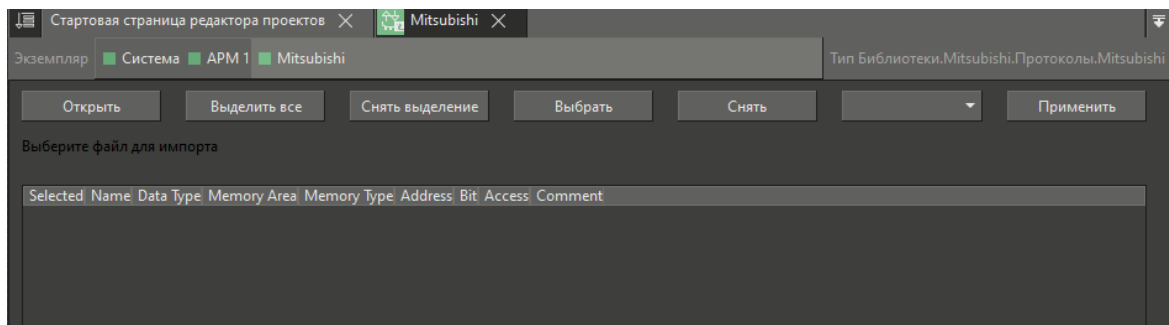
Название	Описание
	содержать число 8 или 9, то опрос данного тега производится не будет. Адрес остальных тегов задаются в десятичной системе
BitAddress	Номер извлекаемого бита (для областей Data Registers, Special Registers).

Описание остальных свойств смотрите в соответствующих разделах

6.2.3.2.6.3. ДИАЛОГ ИМПОРТА КАНАЛОВ MITSUBISHI

Для автоматического добавления каналов в протокол Mitsubishi нужно дважды нажать левую кнопку мыши по протоколу в дереве. При этом откроется вкладка, в которой можно выполнить импорт каналов. Импортировать можно из файла с расширением *csv*. Получить его можно в приложении GX Works.

Вид вкладки по умолчанию:



Описание элементов управления:

Элемент	Описание
Открыть	Открывает диалоговое окно Windows, в котором можно выбрать файл, из которого нужно выполнить импорт данных. После того как файл будет открыт в таблице, расположенной на данной вкладке появится список каналов.
Выделить все	Выделяет все каналы в таблице
Снять выделение	Снимает выделение со всех ранее выбранных каналов в таблице

Элемент	Описание
Выбрать	Если выделить группу каналов или только один и нажать кнопку Выбрать. То напротив этих каналов установится флаг. Флаг также можно установить/снять нажав по нему в поле таблицы Selected
Снять	Если выделить группу каналов или только один, из тех которые ранее были отмечены флагами в таблице и нажать кнопку Снять. То напротив этих каналов флаг будет снят. Флаг также можно установить/снять нажав по нему в поле таблицы Selected
Выпадающий список Доступ	Определяет доступ выделенных каналов. доступ также может быть изменен щелчком мыши по ячейке Тип доступа.
Применить	После нажатия на кнопку Применить отмеченные флагами каналы будут добавлены в дерево системы MasterSCADA 4D

Пример вида вкладки после открытия файла с переменными:

Стартовая страница редактора проектов Mitsubishi

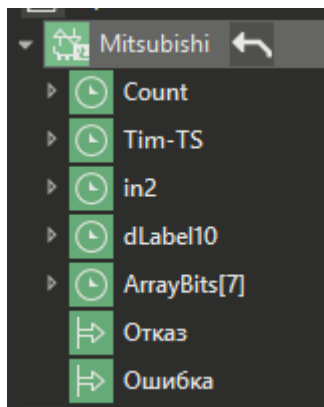
Экземпляр Система APM 1 Mitsubishi Тип Библиотеки.Mitsubishi.Протоколы.Mitsubishi

Открыть Выделить все Снять выделени Выбрать Снять Применить

Отметьте нужные теги и нажмите Применить

Selected	Name	Data Type	Memory Area	Memory Type	Address	Bit	Access	Comment
<input checked="" type="checkbox"/>	Count	INT	Data Register (D)-Word		100		ReadWrite	
<input checked="" type="checkbox"/>	Tim-TS	BOOL	Timer	(TS)-Bit	1		ReadWrite	
<input type="checkbox"/>	Tim-TC	BOOL	Timer	(TC)-Bit	1		ReadWrite	
<input type="checkbox"/>	Tim-TN	INT	Timer	(TN)-Word	1		ReadWrite	
<input type="checkbox"/>	In1	BOOL	Input (X)-Bit		1		ReadWrite	
<input checked="" type="checkbox"/>	in2	BOOL	Data Register (D)-Word		300	0	ReadWrite	
<input type="checkbox"/>	IcdLabel5-CS	BOOL	Counter	(CS)-Bit	100		ReadWrite	
<input type="checkbox"/>	IcdLabel5-CC	BOOL	Counter	(CC)-Bit	100		ReadWrite	
<input type="checkbox"/>	IcdLabel5-CN	INT	Counter	(CN)-Word	100		ReadOnly	
<input type="checkbox"/>	ddd[0]	INT	Data Register (D)-Word		400		ReadOnly	[0]
<input type="checkbox"/>	ddd[1]	INT	Data Register (D)-Word		401		ReadWrite	[1]
<input type="checkbox"/>	Label8-ST5	BOOL	Retentive Timer	(ST5)-Bit	0		ReadWrite	
<input type="checkbox"/>	Label8-STC	BOOL	Retentive Timer	(STC)-Bit	0		ReadWrite	
<input type="checkbox"/>	Label8-STN	INT	Retentive Timer	(STN)-Word	0		ReadWrite	
<input type="checkbox"/>	udLabel9	DWORD	Data Register (D)-Word		500		ReadWrite	
<input checked="" type="checkbox"/>	dLabel10	DINT	Data Register (D)-Word		600		ReadWrite	
<input type="checkbox"/>	wLabel11	INT	Data Register (D)-Word		700		ReadWrite	ddd
<input type="checkbox"/>	eLabel12	REAL	Data Register (D)-Word		800		ReadWrite	
<input type="checkbox"/>	tmLabel13	TIME	Data Register (D)-Word		900		ReadWrite	
<input type="checkbox"/>	sLabel14	STRING[32]	Data Register (D)-Word		1000		ReadWrite	
<input type="checkbox"/>	ArrayBits[0]	BOOL	Data Register (D)-Word		1200	0	ReadOnly	[0]
<input type="checkbox"/>	ArrayBits[1]	BOOL	Data Register (D)-Word		1201	0	ReadWrite	[1]
<input type="checkbox"/>	ArrayBits[2]	BOOL	Data Register (D)-Word		1202	0	ReadOnly	[2]
<input type="checkbox"/>	ArrayBits[3]	BOOL	Data Register (D)-Word		1203	0	ReadOnly	[3]
<input type="checkbox"/>	ArrayBits[4]	BOOL	Data Register (D)-Word		1204	0	ReadOnly	[4]
<input type="checkbox"/>	ArrayBits[5]	BOOL	Data Register (D)-Word		1205	0	ReadOnly	[5]
<input type="checkbox"/>	ArrayBits[6]	BOOL	Data Register (D)-Word		1206	0	ReadOnly	[6]
<input checked="" type="checkbox"/>	ArrayBits[7]	BOOL	Data Register (D)-Word		1207	0	ReadOnly	[7]
<input type="checkbox"/>	ArrayBits[8]	BOOL	Data Register (D)-Word		1208	0	ReadWrite	[8]

Вид дерева системы после нажатия на кнопку Применить:



ЭКСПОРТ ПЕРЕМЕННЫХ ИЗ GX WORKS

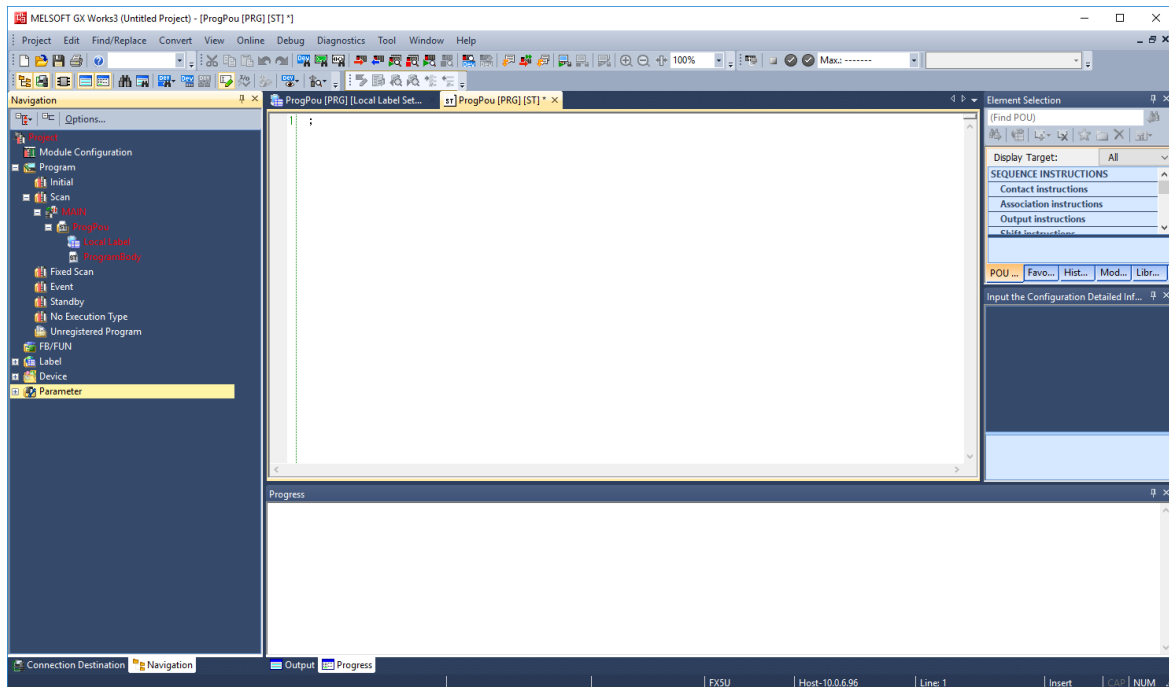
Среда разработки GX Works допускает экспорт глобальных переменных в сторонний файл. Для экспорта, в дереве проекта необходимо выбрать раздел Label - Global Label - Global.

После этого можно приступать к импортированию тегов в MasterSCADA 4D

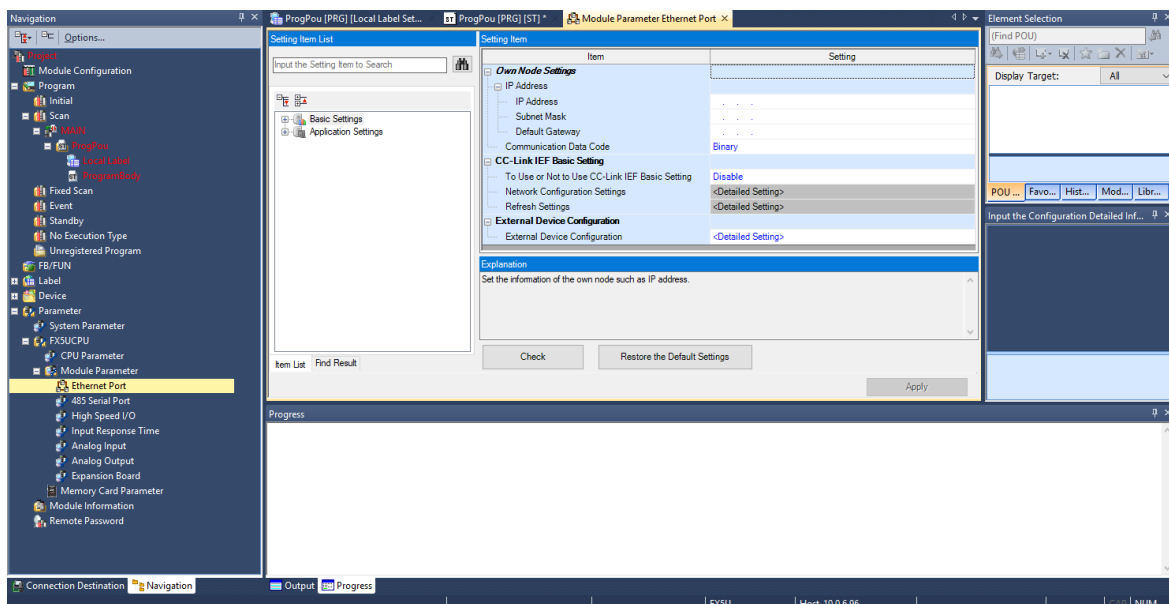
6.2.3.2.6.4. НАСТРОЙКА КОНТРОЛЛЕРА MITSUBISHI

Для корректной работы MasterSCADA 4D необходимо настроить контроллер на работу с протоколом SLMP - данная операция делается средствами среды разработки GX Works 3.

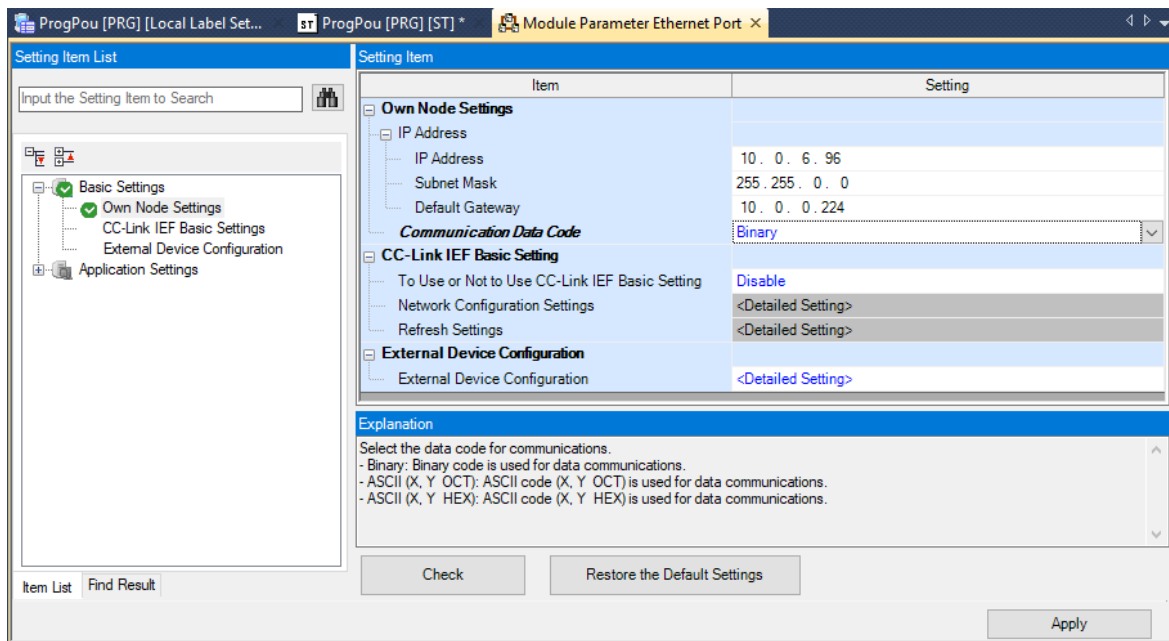
Запустите GX Works и создайте новый проект и укажите тип используемого контроллера или откройте уже существующий проект.



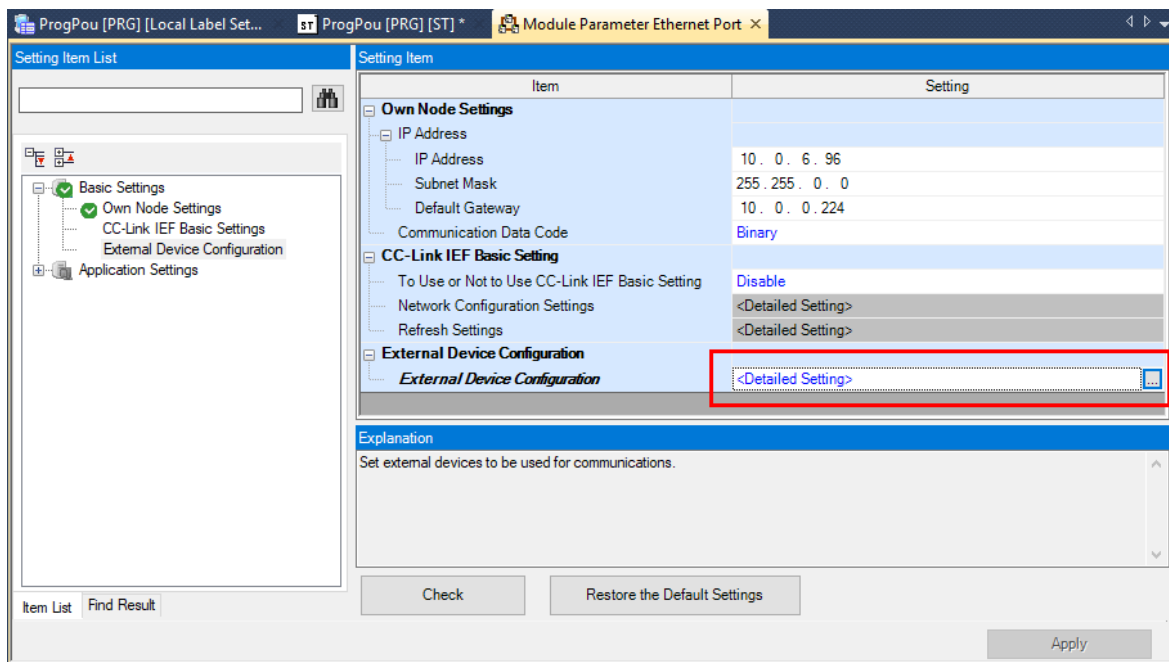
В дереве проекта откройте раздел Parameter - Имя контроллера (например FX5UCPU) - Module Parameter - Ethernet Port.



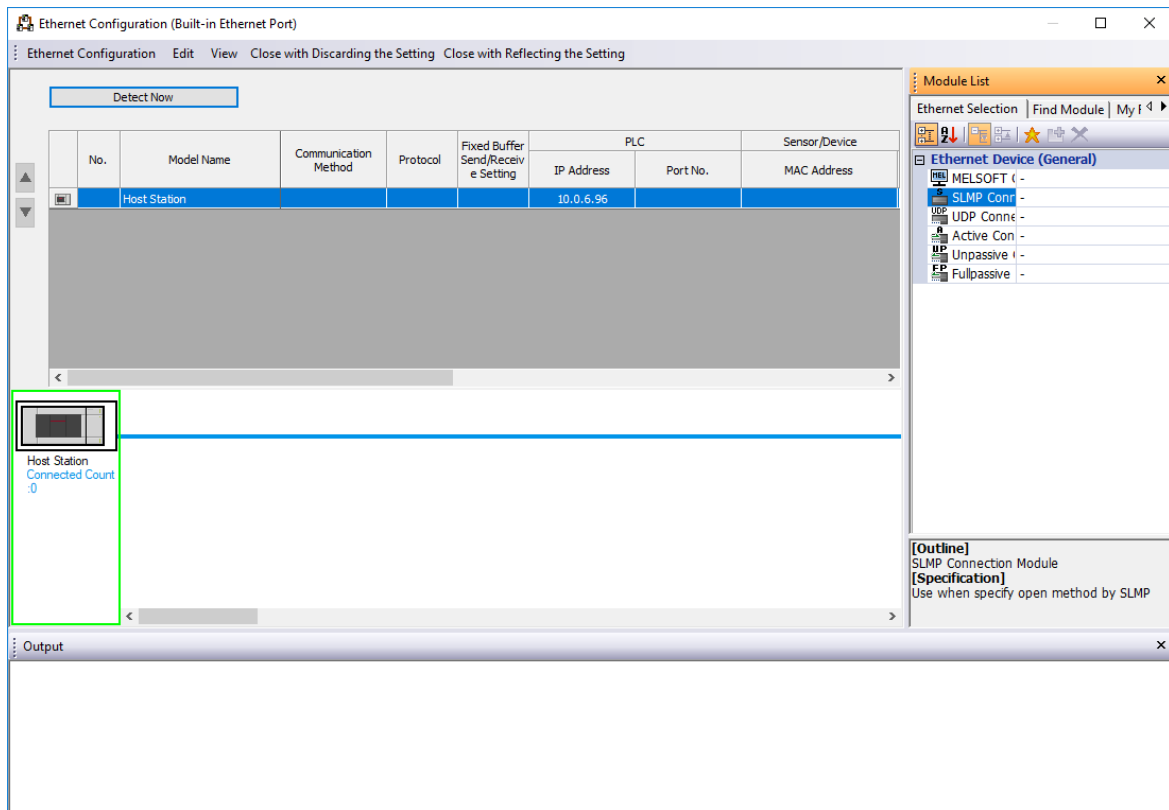
Пропишите параметры подключения - IP адрес контроллера, маску и шлюз, а также тип протокола - Binary.



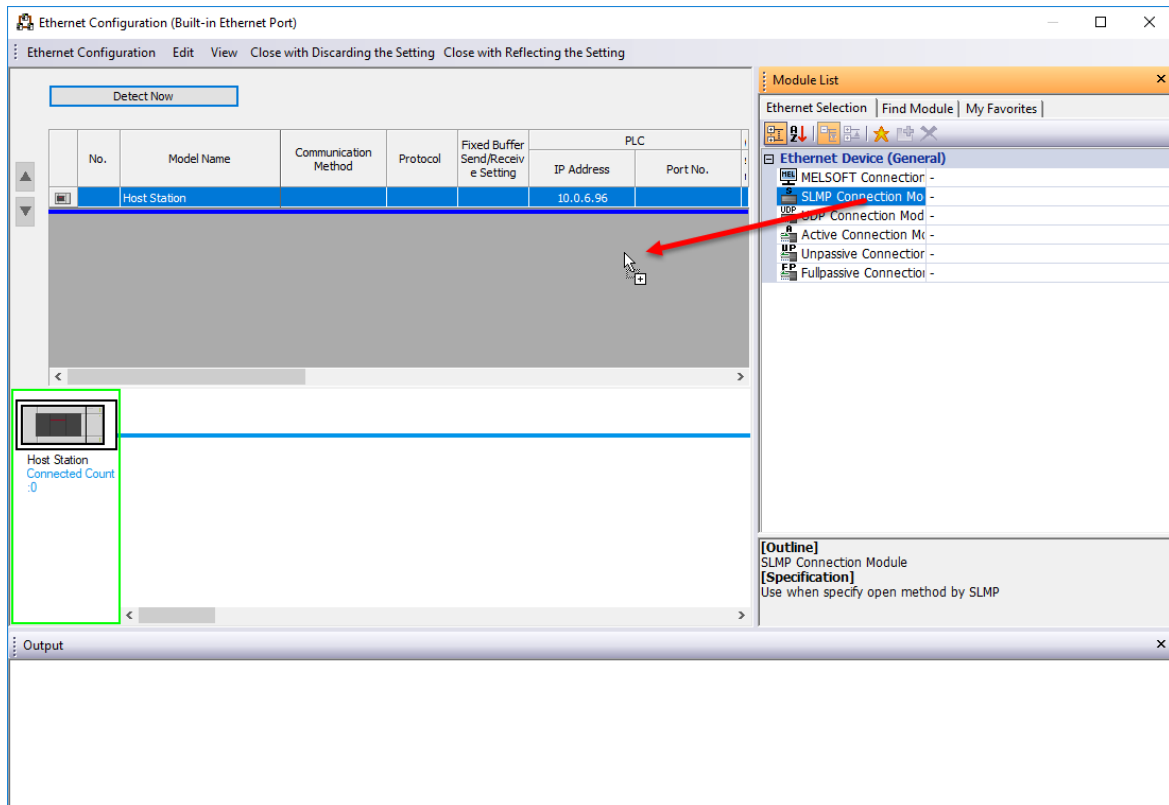
Затем нажмите на раздел External Device Configuration.



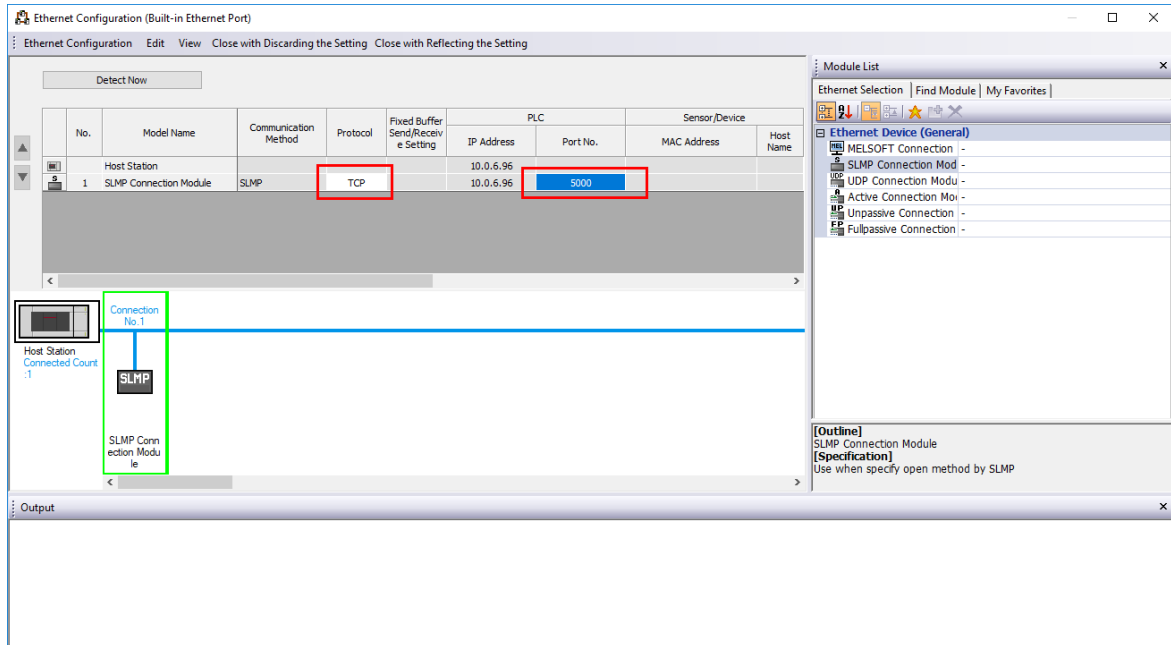
Откроется конфигуратор Ethernet настроек.



С его помощью, необходимо добавить SLMP коннект. Для этого необходимо перетащить его в таблицу:

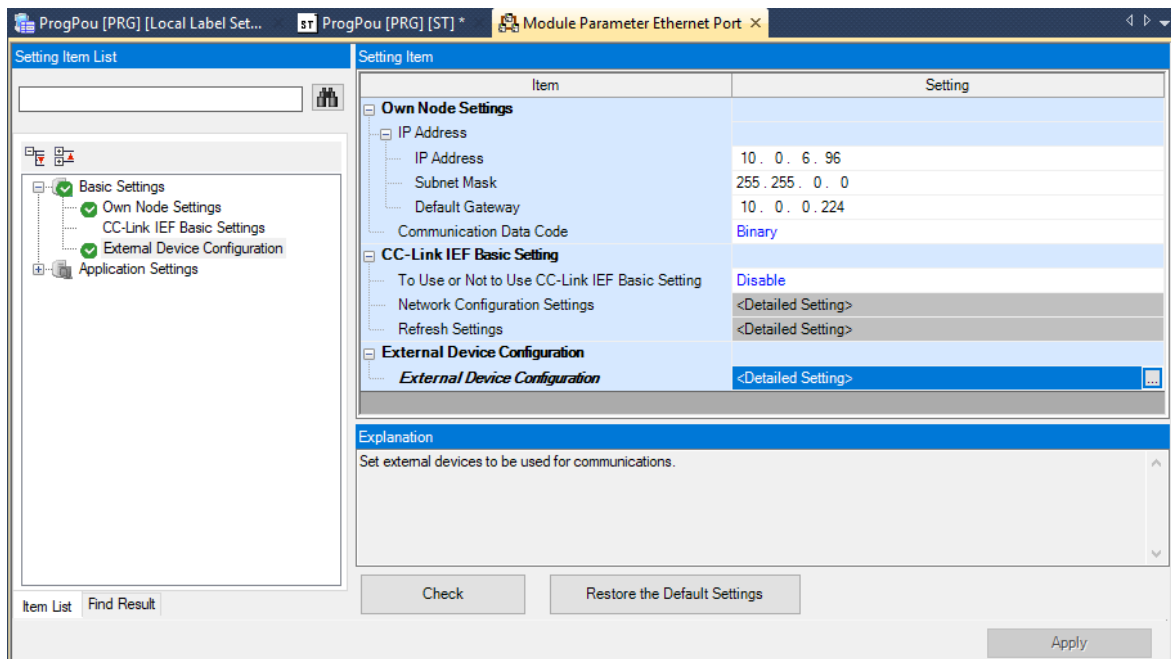


Затем необходимо выбрать протокол TCP и порт - 5000 (порт можно изменить - и этот же порт указывать в настройках MasterSCADA 4D).



На этом настройка завершена - нажмите на кнопку Close with Reflecting Settings.

На панели Module Parameter Ethernet Port нажмите кнопку Check - для проверки корректности настроек, и если ошибок нет - Apply.

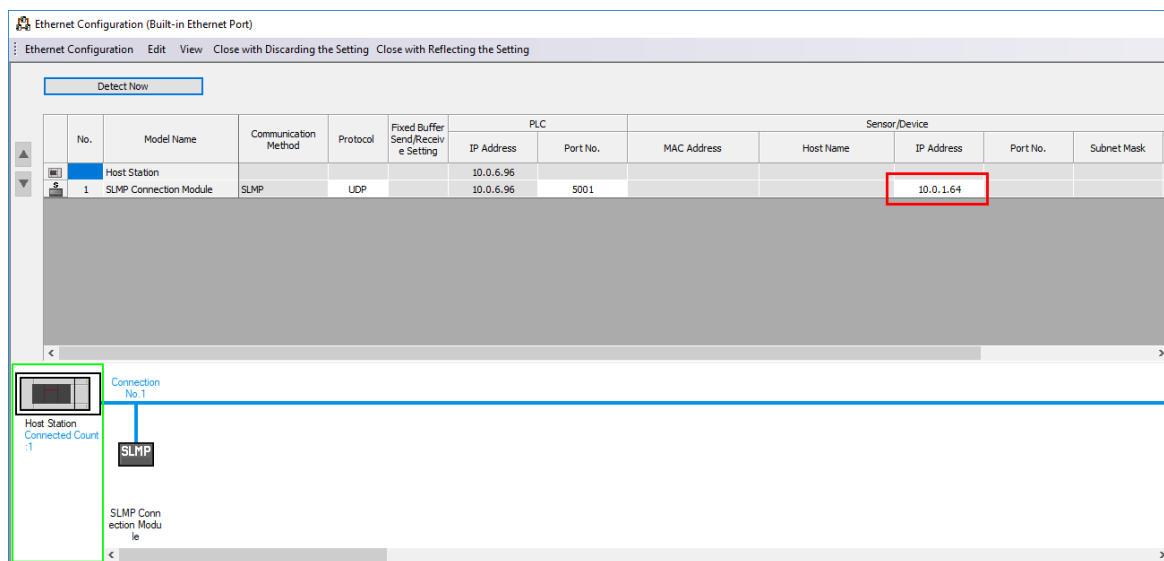


На этом настройка контроллера для Mitsubishi завершена. Можно "заливать" программу в контроллер и проверять функционирование опроса переменных исполнительной системой.

Важно! Для того чтобы контроллер применил настройки необходимо произвести их загрузку в контроллер, а также перезагрузить контроллер

Настройка UDP

Если необходимо вести обмен с контроллером по UDP, то необходимо выбрать данный тип протокола, а также необходимо указать IP адрес компьютера с исполнительной системой:



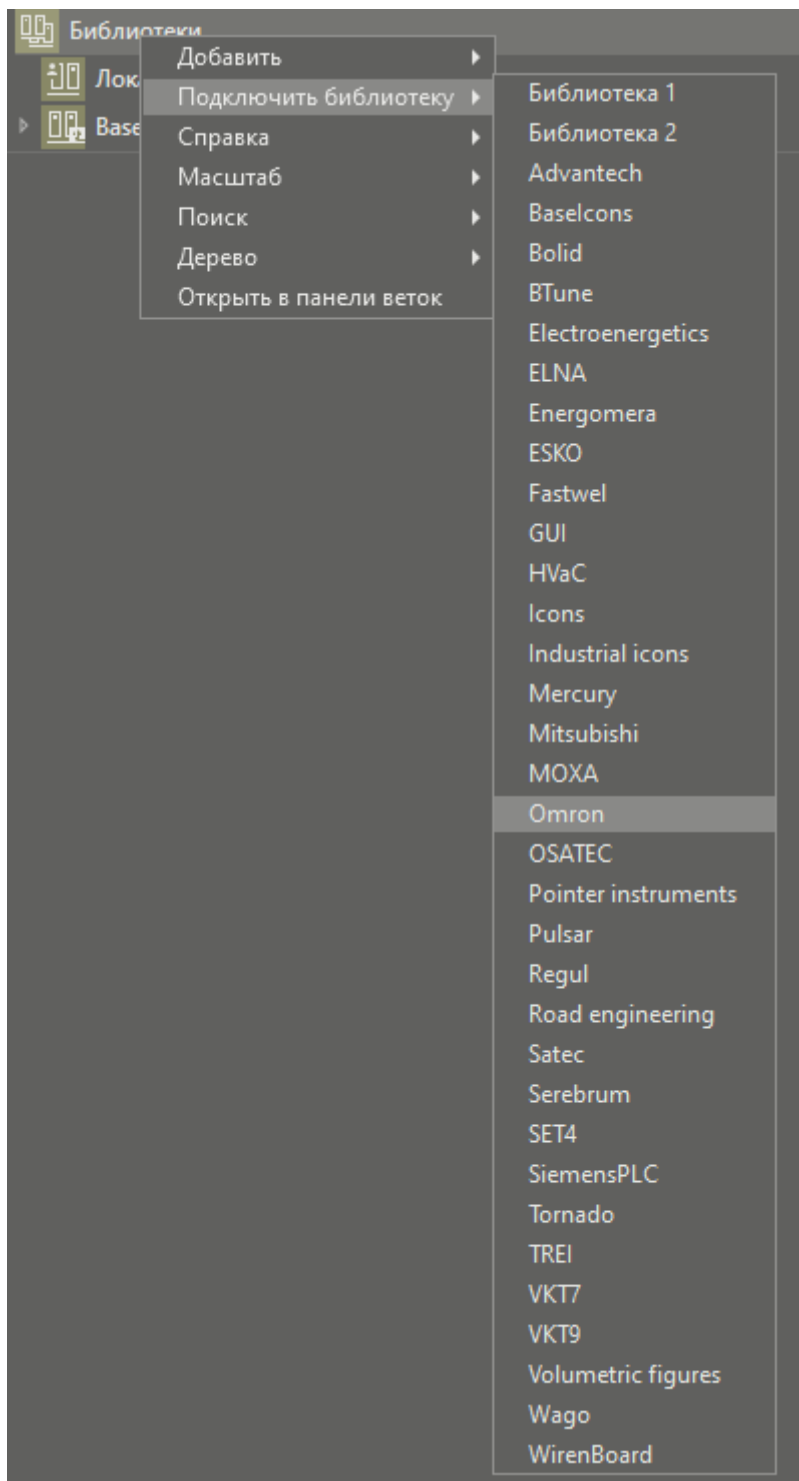
В остальном настройка аналогична.

6.2.3.2.7. OMRON

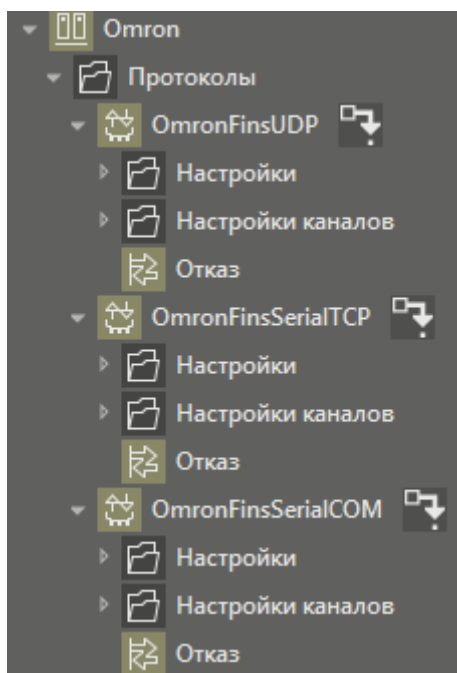
MasterSCADA 4D RT поддерживает работу с контроллерами по протоколу Fins. Протокол FINS – открытый протокол промышленных систем автоматизации разработанный японской корпорацией OMRON. В качестве интерфейса FINS использует как сети базирующиеся на на TCP/IP - Ethernet, Wi-Fi, GPRS, так и на последовательных канал связи - RS-232, RS-485, радио- и GSM-каналы.

Важно! По умолчанию протокол не входят в состав исполнительной системы. Стоимость и условия лицензирования этого протокола необходимо уточнять по электронной почте sales@masterscada.ru

Для получения возможности использования протокола в проекте необходимо подключить соответствующую библиотеку:



В полном дереве библиотека Omron имеет вид:

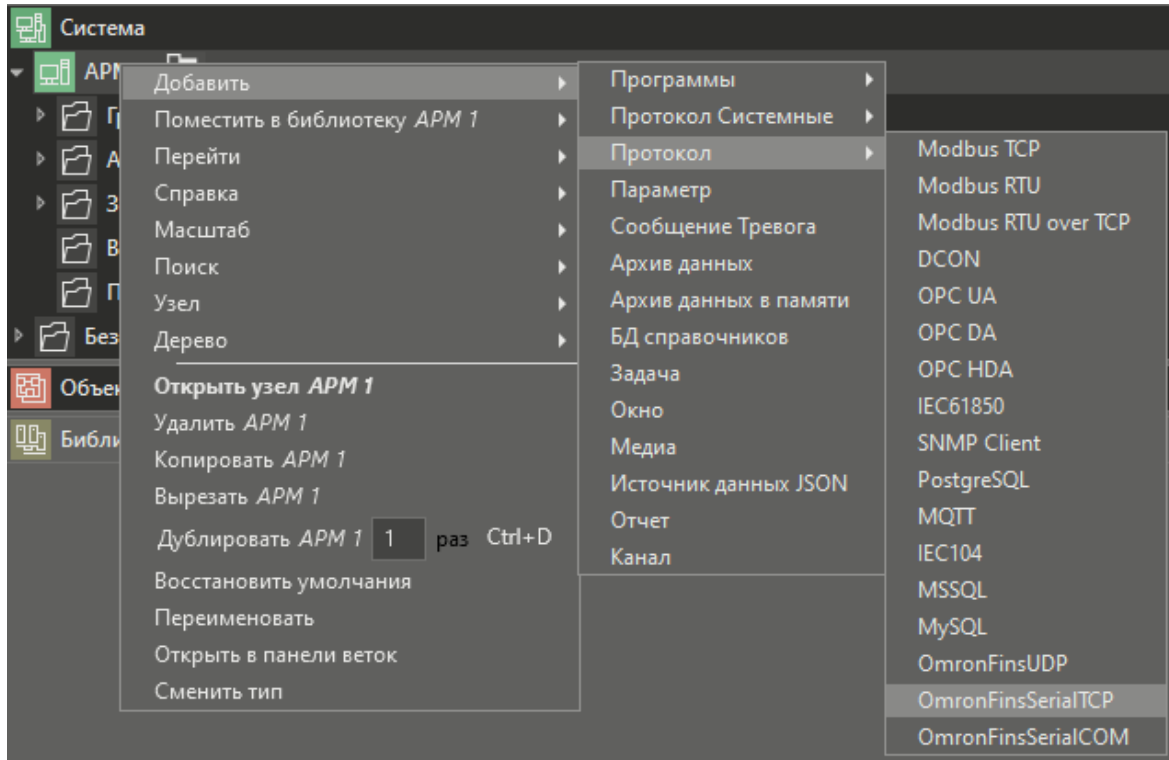


Назначение протоколов:

Название	Назначение
OmronFinsSerialTCP	Предназначен для опроса контроллеров фирмы Omron имеющих последовательный интерфейс, при использовании конвертеров Ethernet-COM (Моха NPort и подобные)
OmronFinsSerialCOM	Предназначен для опроса контроллеров фирмы Omron имеющих последовательный интерфейс, для работы в сетях RS-232 и RS-485
OmronFinsUDP	Предназначен для опроса контроллеров фирмы Omron имеющих интерфейс Ethernet.

После этого в проект может быть добавлен любой из протоколов библиотеки любым известным способом, например, при помощи контекстного меню узла, или с использованием контекстной панели, или путем перетаскивания элемента из библиотеки в дерево системы.

Рекомендуется добавлять элементы в проект при помощи контекстного меню:



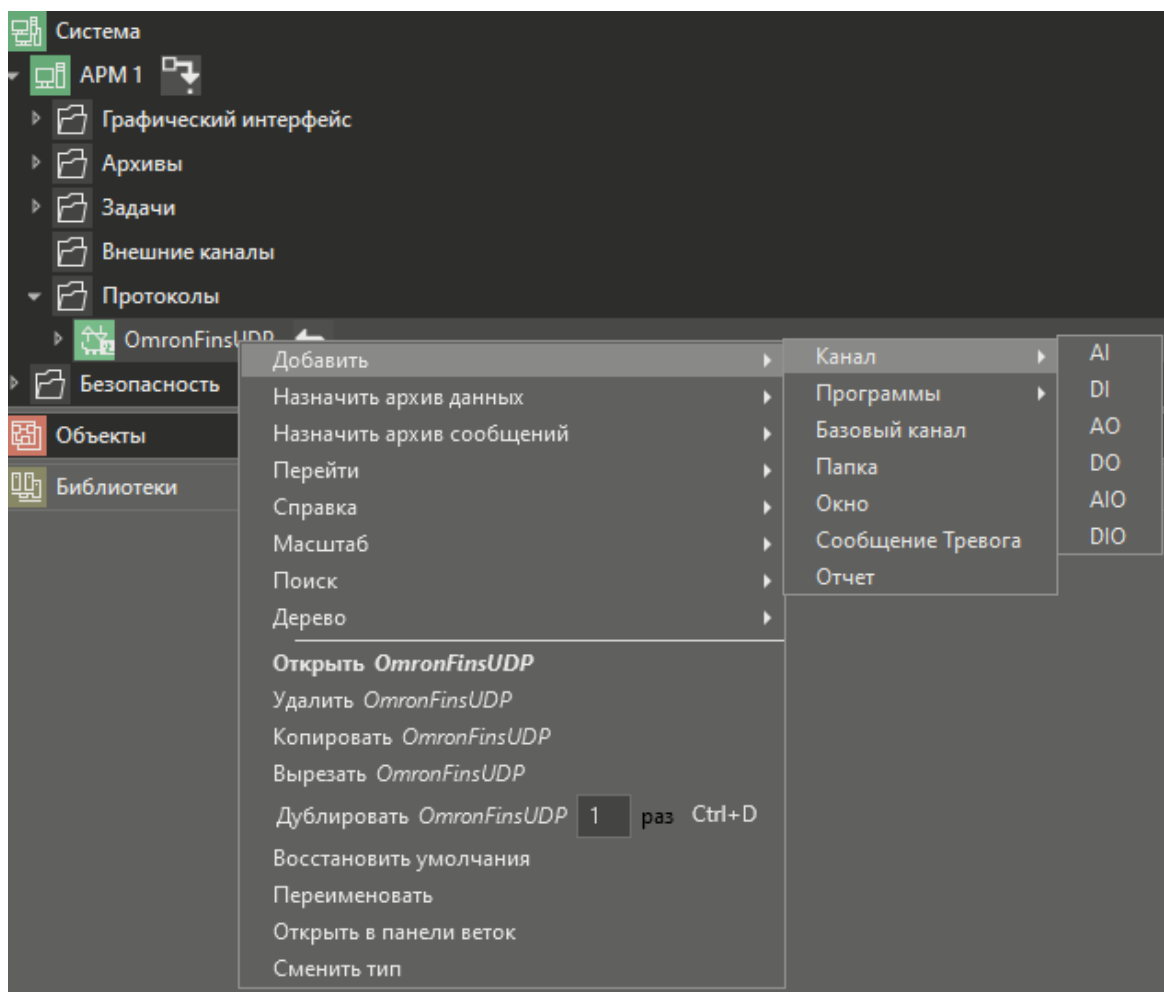
Тип добавляемого протокола зависит от реального способа подключения контроллера.

Затем необходимо настроить панель свойств протокола.

Все протоколы библиотеки Omron имеют выход Отказ

Далее требуется добавить каналы в протокол. Сделать это можно вручную, либо импортировать из файла.

Добавить канал вручную можно при помощи контекстного меню протокола:



Далее необходимо настроить панель свойств канала, ориентируясь на его свойства, заданные в приложении CX-Programmer.

Для автоматического добавления каналов нужно дважды нажать левую кнопку мыши по протоколу в дереве. При этом откроется вкладка, в которой можно выполнить импорт каналов.

6.2.3.2.7.1. СВОЙСТВА ПРОТОКОЛОВ OMRON

Свойства OmronFinsUDP

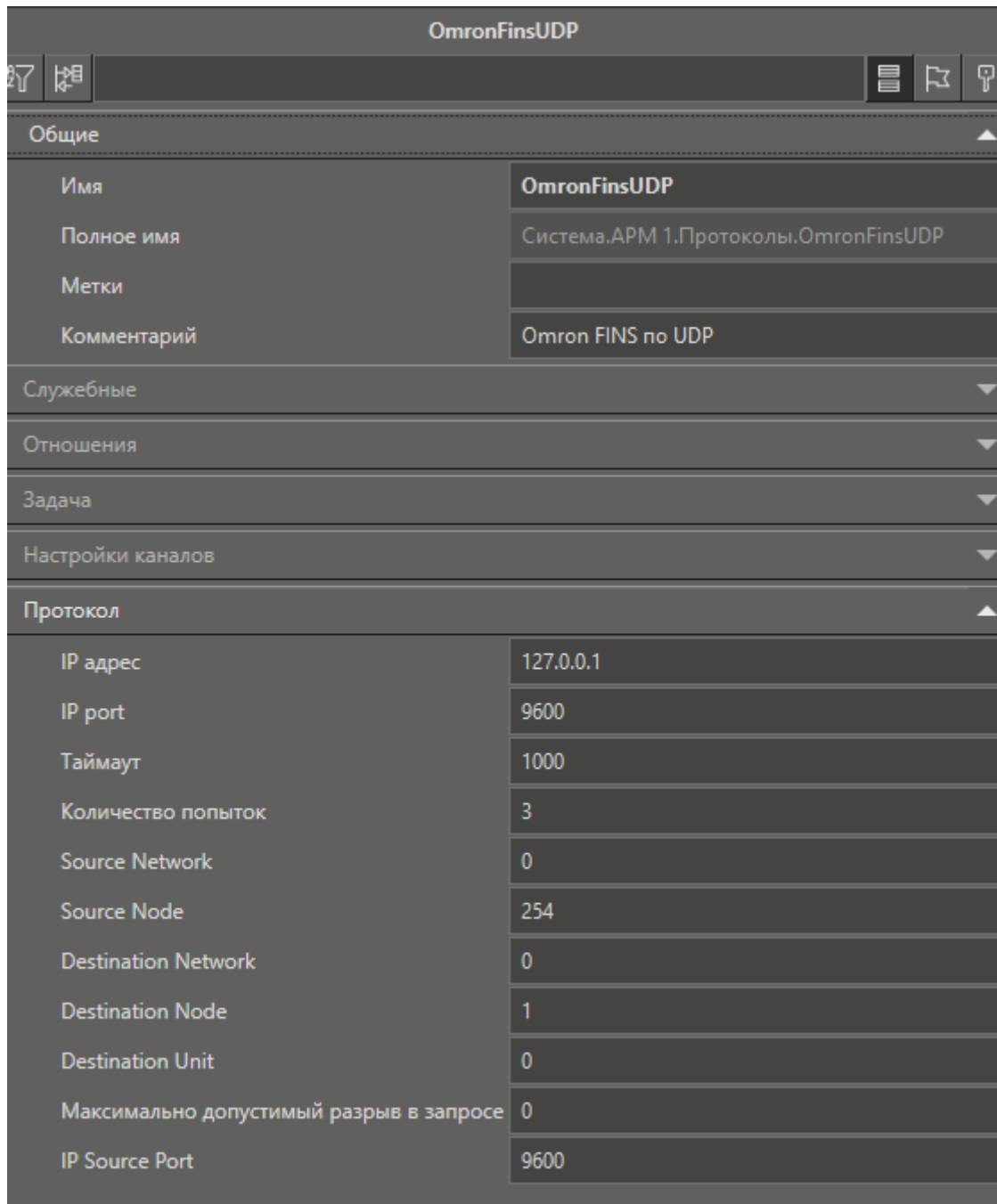
Свойства OmronFinsSerialCOM

Свойства OmronFinsSerialTCP

СВОЙСТВА OMRONFINSUDP

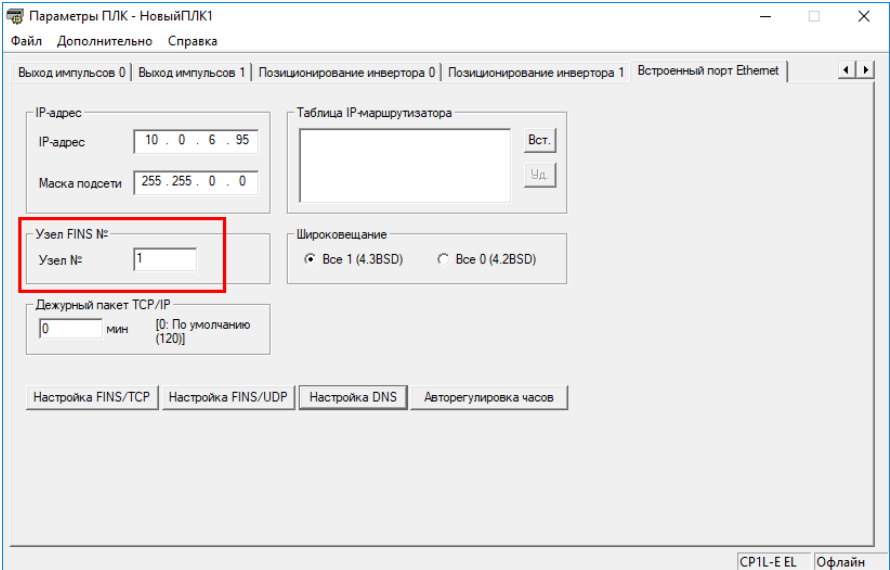
Описание свойств протокола OmronFinsUDP

Вид панели свойств:



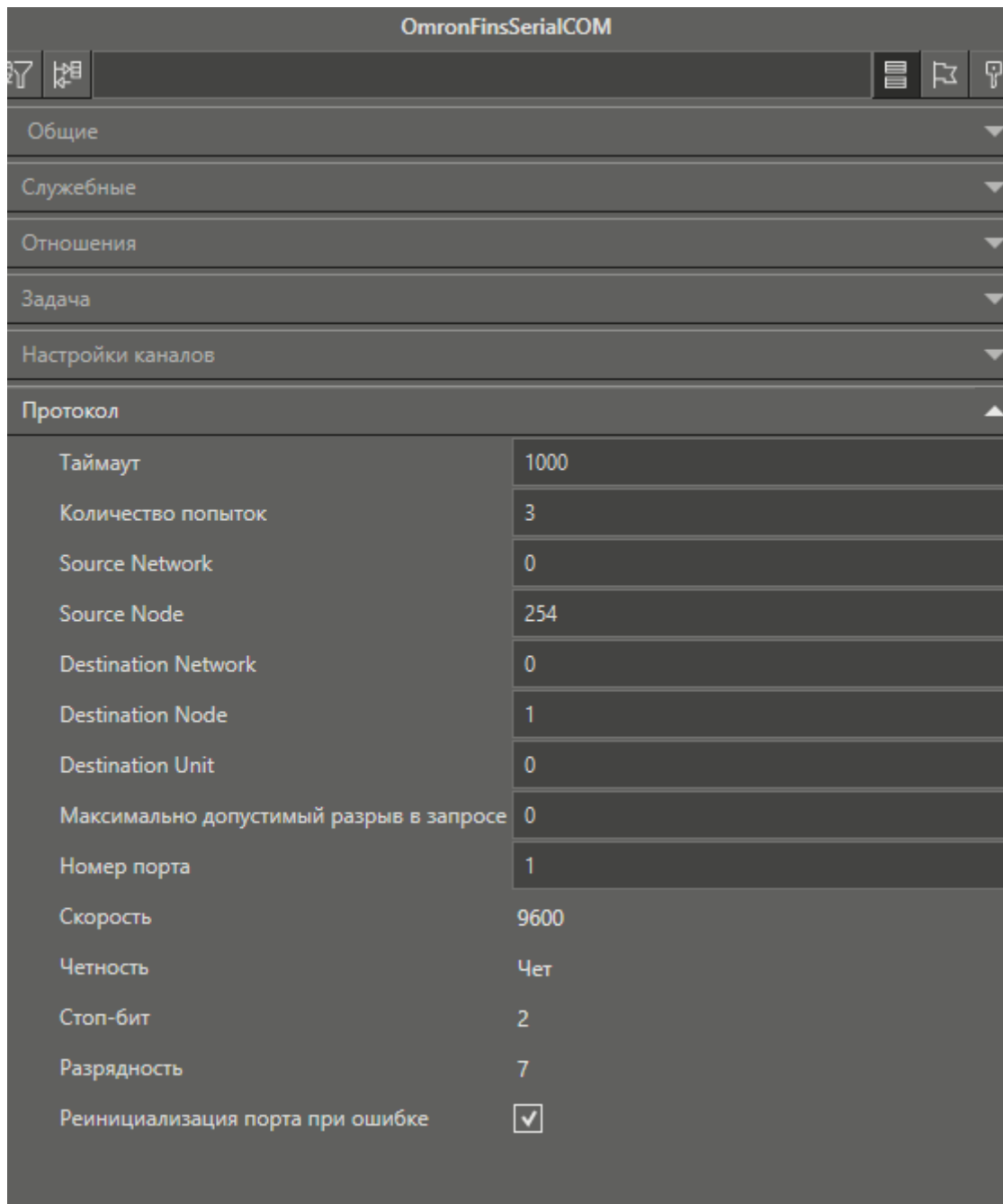
Название	Рекомендации
Категория Общие	Как правило, для данного элемента эта категория не настраивается. Подробное описание данной категории приведено в соответствующем разделе
Категория Задача	Задается период, с которым MasterSCADA 4D будет стараться получать значения из сервера. Определяется способ работы протокола при резервировании узла. Подробное описание

Название	Рекомендации
	категории приведено в разделе, описывающем настройку категории Задачи у протоколов.
Категория Настройки каналов	Задаются значения по умолчанию для создаваемых каналов. В настройке не нуждаются, т.к. индивидуальные настройки каналов формируются автоматически.
Категория Протокол	
IP адрес	Основной IP-адрес устройства, которое отправляет и получает данные от MasterSCADA 4D.
IP порт	UDP порт для запросов к контроллеру. По умолчанию - 9600.
Таймаут	Время, в течение которого исполнительная система ожидает установления соединения с контроллером
Количество попыток	<p>Указывается предельное количество неудачных попыток получить ответ от контроллера, произошедших подряд. Если за указанное количество попыток получить ответ не удалось, то сформируется сигнал TRUE у параметра Отказ протокола</p> <p>Если при обращении к контроллеру он не ответит ни разу за указанное количество попыток подряд, то следующие запросы в рамках данного цикла опроса посылаться не будут, сформируется отказ модуля и выставится признак недоверности по всем входам протокола. Если при запросе ответ придет, но с ошибкой, то сформируется отказ и выставится признак недоверности по входам в данном запросе, но посылка других запросов по этому контроллеру продолжится.</p>
Source Network	Адрес сети источника. Адрес по умолчанию - 0, локальная сеть (сам контроллер). Адреса 1-32 - удаленная сеть Controller Link.
Source Node	Адрес узла источника. 0 - внутренняя сеть PC. 1-255 - удаленная сеть. По умолчанию - 254.

Название	Рекомендации
Destination Network	Адрес сети назначения. Используется если контроллер работает как шлюз. Значение по умолчанию - 0, локальная сеть контроллера, 1-127 удаленная сеть.
Destination Node	<p>Адрес контроллера. Задается специальным аппаратным переключателем или в среде разработки программы для контроллера:</p>  <p>The screenshot shows a software window titled 'Параметры ПЛК - НовыйПЛК1'. It contains several configuration fields: 'IP-адрес' (10.0.6.95), 'Маска подсети' (255.255.0.0), 'Узел FINS №' (1), 'Узел №' (1), and 'Широковещание' (All 1 (4.3BSD)). The 'Узел FINS №' field is highlighted with a red rectangle.</p>
Destination Unit	Адрес модуля. По умолчанию 0 - CPU. Остальные значения - модули подключенные к контроллеру
Максимально допустимый разрыв в запросе чтения	<p>Параметр определяет максимально допустимый разрыв между адресами в запросе чтения, при котором происходит объединение в один запрос. Например, производится опрос регистров с номерами - 0, 1, 2, 4, 5. Если параметр будет установлен в 0, то регистры будут запрошены за два запроса (0-2, 4-5), если же установить его в 1 или выше, то за один запрос (0-5, а не использованный регистр 3 будет отброшен при разборе ответа). Увеличение данного параметра сокращает количество запросов и ускоряет опрос, но удлиняет кадр ответа.</p>
IP Source Port	UDP порт для ответов от контроллера. Значение по умолчанию - 9600, 0 - динамический порт (выбирается системой).

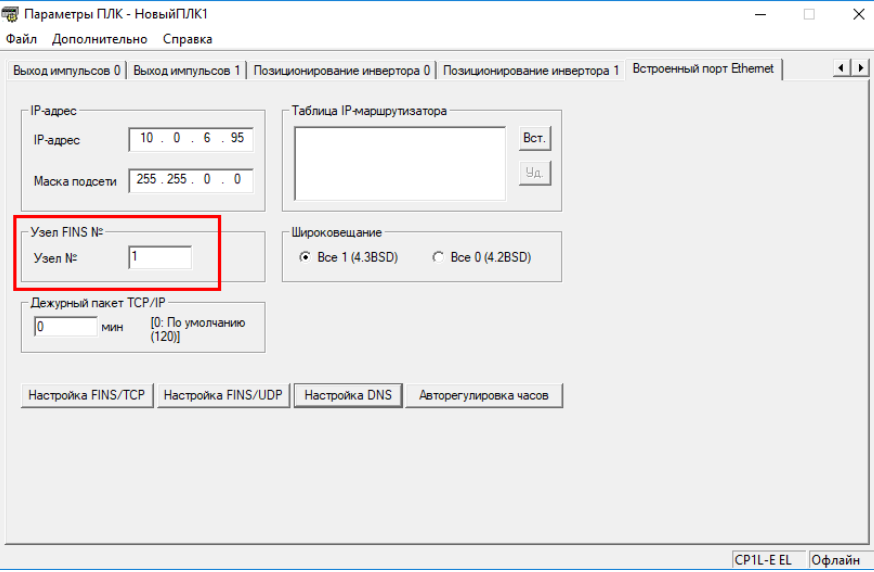
Описание свойств протокола OmronFinsSerialCom

Вид панели свойств:



Название	Рекомендации
Категория Общие	Как правило, для данного элемента эта категория не настраивается. Подробное описание данной категории приведено в соответствующем разделе

Название	Рекомендации
Категория Задача	Задается период, с которым MasterSCADA 4D будет стараться получать значения из сервера. Определяется способ работы протокола при резервировании узла. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи у протоколов.
Категория Настройки каналов	Задаются значения по умолчанию для создаваемых каналов. В настройке не нуждаются, т.к. индивидуальные настройки каналов формируются автоматически.
Категория Протокол	
Таймаут	Время, в течение которого исполнительная система ожидает установления соединения с контроллером
Количество попыток	<p>Указывается предельное количество неудачных попыток получить ответ от контроллера, произошедших подряд. Если за указанное количество попыток получить ответ не удалось, то сформируется сигнал TRUE у параметра Отказ протокола</p> <p>Если при обращении к контроллеру он не ответит ни разу за указанное количество попыток подряд, то следующие запросы в рамках данного цикла опроса посылаться не будут, сформируется отказ модуля и выставится признак недоверности по всем входам протокола. Если при запросе ответ придет, но с ошибкой, то сформируется отказ и выставится признак недоверности по входам в данном запросе, но посылка других запросов по этому контроллеру продолжится.</p>
Source Network	Адрес сети источника. Адрес по умолчанию - 0, локальная сеть (сам контроллер). Адреса 1-32 - удаленная сеть Controller Link.
Source Node	Адрес узла источника. 0 - внутренняя сеть РС. 1-255 - удаленная сеть. По умолчанию - 254.

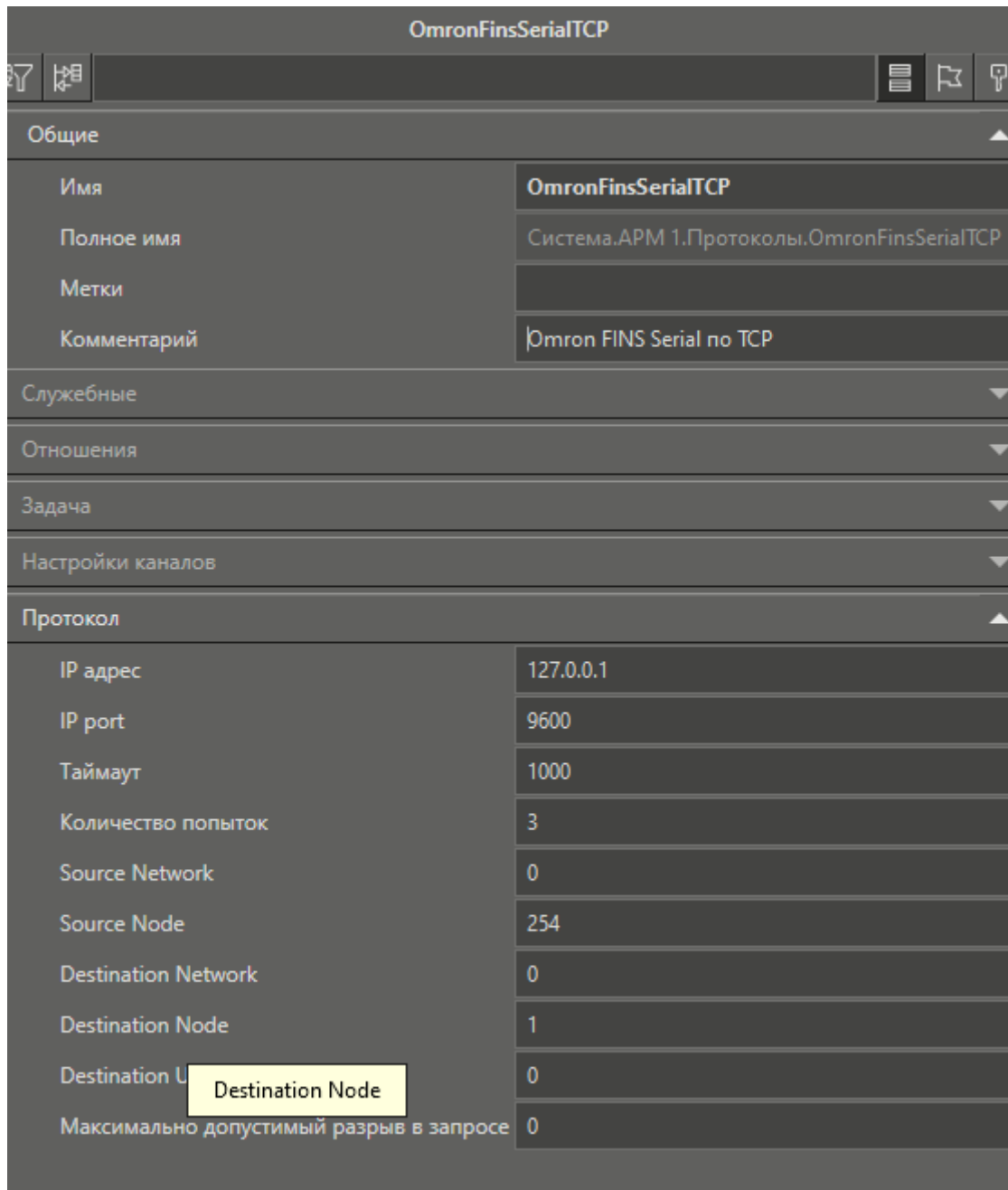
Название	Рекомендации
Destination Network	Адрес сети назначения. Используется если контроллер работает как шлюз. Значение по умолчанию - 0, локальная сеть контроллера, 1-127 удаленная сеть.
Destination Node	<p>Адрес контроллера. Задается специальным аппаратным переключателем или в среде разработки программы для контроллера:</p> 
Destination Unit	Адрес модуля. По умолчанию 0 - CPU. Остальные значения - модули подключенные к контроллеру
Максимально допустимый разрыв в запросе чтения	<p>Параметр определяет максимально допустимый разрыв между адресами в запросе чтения, при котором происходит объединение в один запрос. Например, производится опрос регистров с номерами - 0, 1, 2, 4, 5. Если параметр будет установлен в 0, то регистры будут запрошены за два запроса (0-2, 4-5), если же установить его в 1 или выше, то за один запрос (0-5, а не использованный регистр 3 будет отброшен при разборе ответа). Увеличение данного параметра сокращает количество запросов и ускоряет опрос, но удлиняет кадр ответа.</p>
Номер порта	Номер COM-порта, к которому подключается контроллер Omron

Название	Рекомендации
Скорость	Скорость, на которой будет производиться опрос (бит/с). Значение должно выбираться в зависимости от возможностей подключаемого устройства. Значение по умолчанию 9600.
Четность	Определяется необходимость контроля четности. Возможные значения: Нет, Чет, Нечет.
Стоп-бит	Задается число стоповых битов (1, 1.5, 2), значение по умолчанию – 2;
Разрядность	Задается число информационных битов (7 или 8), значение по умолчанию – 7;
Реинициализация порта при ошибке	При отказе устройства порт будет закрыт и открыт снова

СВОЙСТВА OMRONFINSSERIALTCP

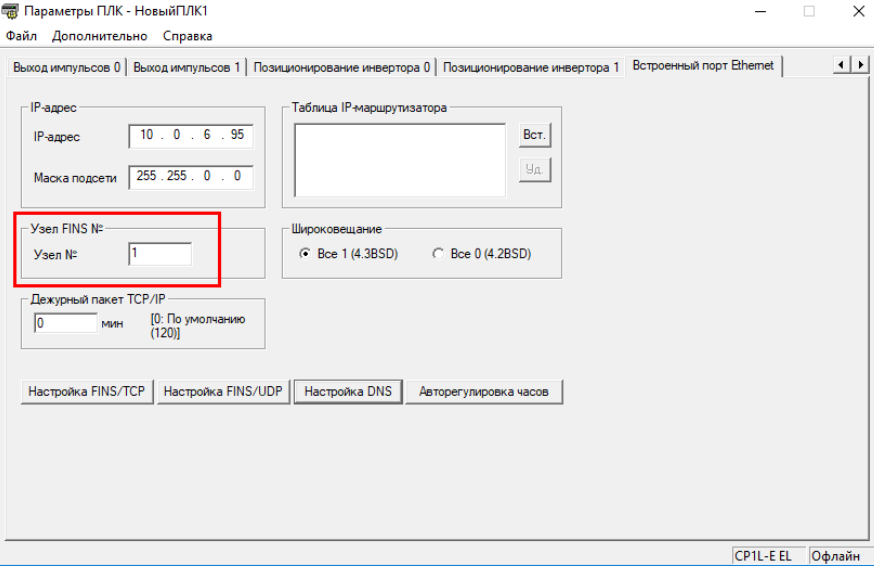
Описание свойств протокола OmronFinsSerialTCP

Вид панели свойств:



Название	Рекомендации
Категория Общие	Как правило, для данного элемента эта категория не настраивается. Подробное описание данной категории приведено в соответствующем разделе
Категория Задача	Задается период, с которым MasterSCADA 4D будет стараться получать значения из сервера. Определяется способ работы протокола при резервировании узла. Подробное описание

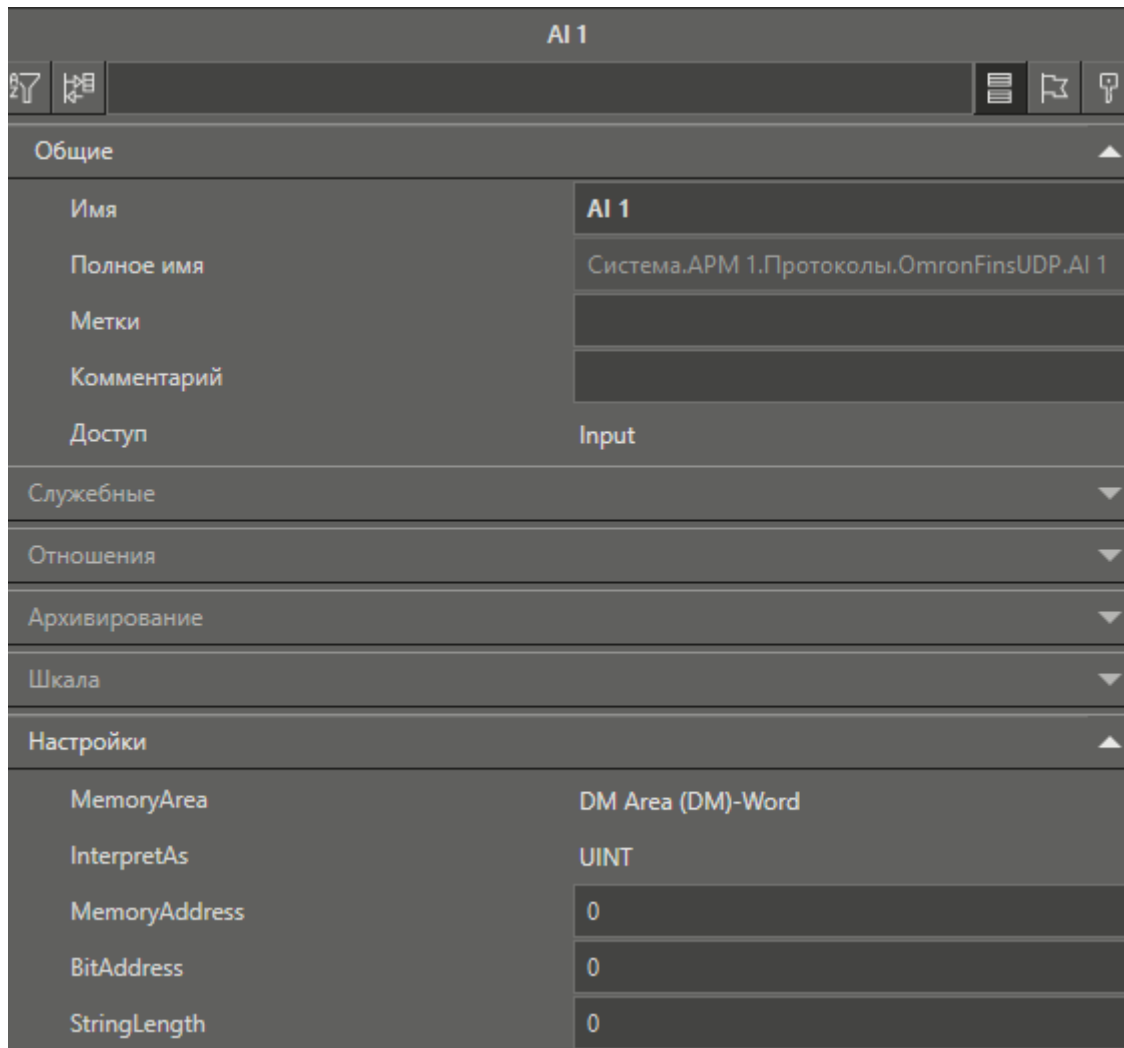
Название	Рекомендации
	категории приведено в разделе, описывающем настройку категории Задачи у протоколов.
Категория Настройки каналов	Задаются значения по умолчанию для создаваемых каналов. В настройке не нуждаются, т.к. индивидуальные настройки каналов формируются автоматически.
Категория Протокол	
IP адрес	Основной IP-адрес устройства, которое отправляет и получает данные от MasterSCADA 4D.
IP порт	UDP порт для запросов/ответов к контроллеру. По умолчанию - 9600.
Таймаут	Время, в течение которого исполнительная система ожидает установления соединения с контроллером
Количество попыток	
Source Network	Адрес сети источника. Адрес по умолчанию - 0, локальная сеть (сам контроллер). Адреса 1-32 - удаленная сеть Controller Link.
Source Node	Адрес узла источника. 0 - внутренняя сеть PC. 1-255 - удаленная сеть. По умолчанию - 254.
Destination Network	Адрес сети назначения. Используется если контроллер работает как шлюз. Значение по умолчанию - 0, локальная сеть контроллера, 1-127 удаленная сеть.
Destination Node	Адрес контроллера. Задается специальным аппаратным переключателем или в среде разработки программы для контроллера:

Название	Рекомендации
	
Destination Unit	<p>Адрес модуля. По умолчанию 0 - CPU. Остальные значения - модули подключенные к контроллеру</p>
Максимально допустимый разрыв в запросе чтения	<p>Параметр определяет максимально допустимый разрыв между адресами в запросе чтения, при котором происходит объединение в один запрос. Например, производится опрос регистров с номерами - 0, 1, 2, 4, 5. Если параметр будет установлен в 0, то регистры будут запрошены за два запроса (0-2, 4-5), если же установить его в 1 или выше, то за один запрос (0-5, а не использованный регистр 3 будет отброшен при разборе ответа). Увеличение данного параметра сокращает количество запросов и ускоряет опрос, но удлиняет кадр ответа.</p>

6.2.3.2.7.2. СВОЙСТВА КАНАЛОВ OMRON

Если канал протокола добавляется вручную, то необходимо настроить панель свойств.

Вид панели свойств:

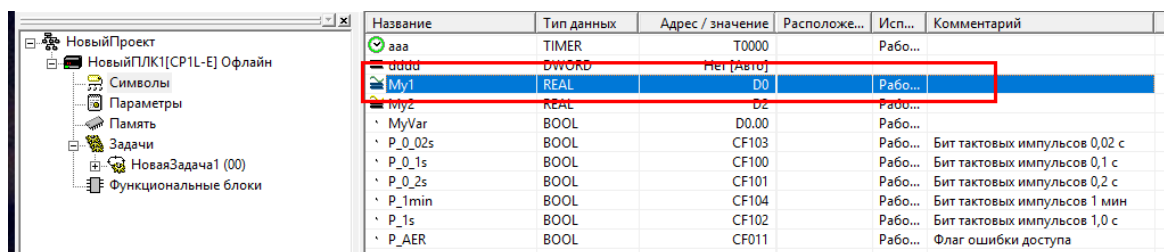


Описание основных свойств

Название	Описание
MemoryArea	Задается область памяти опрашиваемой переменной
InterpretAs	Тип данных переменной
MemoryAddress	Адрес переменной в области памяти.
BitAddress	Номер извлекаемого бита (для областей Bit, Completion Flag, Status).
LengthString	Определяет максимальную длину строки (если выбранный тип данных - строка)

Описание остальных свойств смотрите в соответствующих разделах

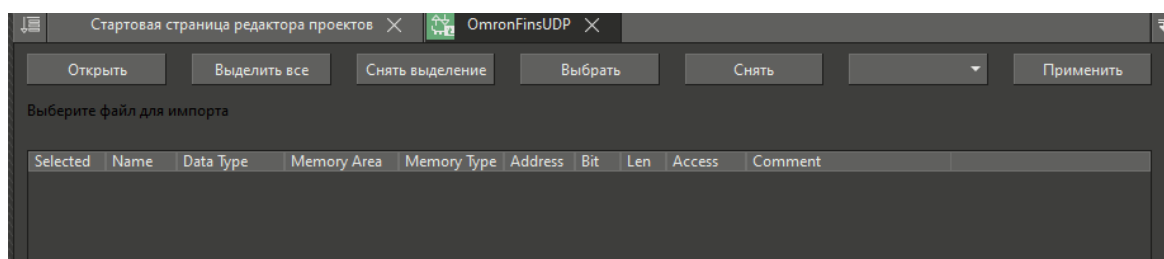
Для получения списка переменных и их адресов нужно открыть проект контроллера в среде разработки – CX-Programmer. Далее найти нужную переменную в разделе Символы:



6.2.3.2.7.3. ДИАЛОГ ИМПОРТА КАНАЛОВ OMRON

Для автоматического добавления каналов в протокол Omron нужно дважды нажать левую кнопку мыши по протоколу в дереве. При этом откроется вкладка, в которой можно выполнить импорт каналов. Импортировать можно из файла с расширением *sxpr*. Получить его можно в приложении CX-Programmer

Вид вкладки по умолчанию:



Описание элементов управления:

Элемент	Описание
Открыть	Открывает диалоговое окно Windows, в котором можно выбрать файл, из которого нужно выполнить импорт данных. После того как файл будет открыт в таблице, расположенной на данной вкладке появится список каналов.
Выделить все	Выделяет все каналы в таблице
Снять выделение	Снимает выделение со всех ранее выбранных каналов в таблице
Выбрать	Если выделить группу каналов или только один и нажать кнопку Выбрать. То напротив этих каналов установится флаг.

Элемент	Описание
	Флаг также можно установить/снять нажав по нему в поле таблицы Selected
Снять	Если выделить группу каналов или только один, из тех которые ранее были отмечены флагами в таблице и нажать кнопку Снять. То напротив этих каналов флаг будет снят. Флаг также можно установить/снять нажав по нему в поле таблицы Selected
Выпадающий список Доступ	Определяет доступ каналов.
Применить	После нажатия на кнопку Применить отмеченные флагами каналы будут добавлены в дерево системы MasterSCADA 4D

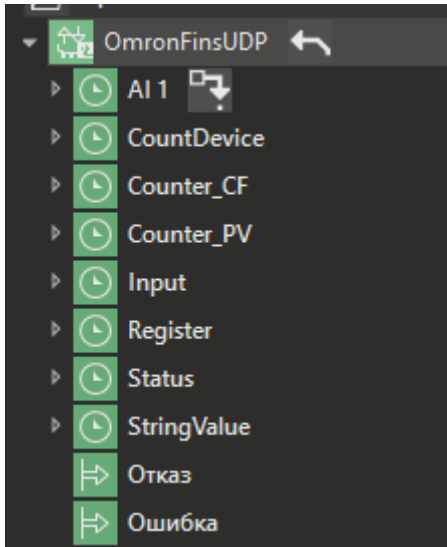
Пример вида вкладки после открытия файла с переменными:

Selected	Name	Data Type	Memory Area	Memory Type	Address	Bit	Len	Access	Comment
<input type="checkbox"/>	AutoVal	DWORD						ReadOnly	local variable
<input checked="" type="checkbox"/>	CountDevice	INT	DM Area (DM)	Word	20			ReadWrite	Count of products
<input checked="" type="checkbox"/>	Counter_CF	BOOL	Counter Area (C	Completion Fl	0001			ReadOnly	Counter
<input checked="" type="checkbox"/>	Counter_PV	BOOL	Counter Area (C	PV	0001			ReadOnly	Counter
<input type="checkbox"/>	Heat	BOOL	CIO Area (CIO)	Bit	0	06		ReadOnly	Output 6
<input checked="" type="checkbox"/>	Input	BOOL	CIO Area (CIO)	Bit	0	00		ReadOnly	Input 1
<input checked="" type="checkbox"/>	Register	WORD	DM Area (DM)	Word	110			ReadWrite	Register status
<input checked="" type="checkbox"/>	Status	DINT	DM Area (DM)	Word	0			ReadWrite	State of machine
<input checked="" type="checkbox"/>	StringValue	STRING	DM Area (DM)	Word	1000		10	ReadWrite	String state of machine
<input type="checkbox"/>	Temperature	REAL	DM Area (DM)	Word	60			ReadWrite	Temperature of oven
<input type="checkbox"/>	Timer1_CF	BOOL	Timer Area (TIM	Completion Fl	0001			ReadOnly	Timer
<input type="checkbox"/>	Timer1_PV	BOOL	Timer Area (TIM	PV	0001			ReadOnly	Timer
<input type="checkbox"/>	TotalCount	LINT	DM Area (DM)	Word	30			ReadWrite	Total count of products
<input type="checkbox"/>	Vakuuum	LREAL	DM Area (DM)	Word	40			ReadWrite	Vakuuum

Не поддерживаемые переменные выделяются красным шрифтом, импортировать данные переменные в конфигурацию нельзя.

Переменные регионов Timer и Counter создают две строки с окончаниями _CF (Complementation Flag) и _PV (Present Value).

Вид дерева системы после нажатия на кнопку Применить:



ЭКСПОРТ ПЕРЕМЕННЫХ ИЗ CX-PROGRAMMER

Среда разработки CX-Programmer допускает экспорт символов (переменных) в сторонний файл. Для экспорта необходимо открыть таблицу символов:

Имя	Тип данных	Адрес / значение	Расположе...	Исп...	Комментарий
aaa	TIMER	T0000		Рабо...	
dddd	DWORD	Нет [Авто]			
in1	BOOL	0.00		Вход	Дискретный вход
My1	REAL	D0		Рабо...	Текущее значение температуры
My2	REAL	D2		Рабо...	Текущее значение давления
My3	WORD	D4		Рабо...	Статус машины
MyVar	BOOL	D0.00		Рабо...	
P_0_02s	BOOL	CF103		Рабо...	Бит тактовых импульсов 0,02 с
P_0_1s	BOOL	CF100		Рабо...	Бит тактовых импульсов 0,1 с
P_0_2s	BOOL	CF101		Рабо...	Бит тактовых импульсов 0,2 с
P_1min	BOOL	CF104		Рабо...	Бит тактовых импульсов 1 мин
P_1s	BOOL	CF102		Рабо...	Бит тактовых импульсов 1,0 с
P_AER	BOOL	CF011		Рабо...	Флаг ошибки доступа
P_CIO	WORD	A450		Рабо...	Параметр области CIO
P_CV	BOOL	CF004		Рабо...	Флаг переноса (CV)
P_Cycle Time Error	BOOL	ΔΔ01 0R		Рабо...	Флаг ошибки времени цикла

Затем выделить необходимые для импорта переменные.

Важно! Переменные должны иметь статичный адрес - переменные с типом Авто импортироваться не будут.

Имя	Тип данных	Адрес / значение	Расположе...	Исп...	Комментарий
aaa	TIMER	T0000		Рабо...	
dddd	DWORD	Нет [Авто]			
in1	BOOL	0.00		Вход	Дискретный вход
My1	REAL	D0		Рабо...	Текущее значение температуры
My2	REAL	D2		Рабо...	Текущее значение давления
My3	WORD	D4		Рабо...	Статус машины
MyVar	BOOL	D0.00		Рабо...	
P_0_02s	BOOL	CF103		Рабо...	Бит тактовых импульсов 0,02 с
P_0_1s	BOOL	CF100		Рабо...	Бит тактовых импульсов 0,1 с
P_0_2s	BOOL	CF101		Рабо...	Бит тактовых импульсов 0,2 с
P_1min	BOOL	CF104		Рабо...	Бит тактовых импульсов 1 мин
P_1s	BOOL	CF102		Рабо...	Бит тактовых импульсов 1,0 с

Затем нужно в контекстном меню выбрать пункт Повторно используемые файл.Сохранить как...

Название	Тип данных	Адрес / значение	Расположе...	Исп...	Комментарий
aaa	TIMER	T0000		Рабо...	
dddd	DWORD	Нет [Авто]			
in1	BOOL	0.00		Вход	Дискретный вход
My1	REAL	D0		Рабо...	Текущее значение температуры
My2	REAL				Давление
My3	WORD				Давление
MyVar	BOOL				
P_0_02s	BOOL				импульсов 0,02 с
P_0_1s	BOOL				импульсов 0,1 с
P_0_2s	BOOL				импульсов 0,2 с
P_1min	BOOL				импульсов 1 мин
P_1s	BOOL				импульсов 1,0 с
P_AER	BOOL				поступа
P_CIO	WORD				сти CIO
P_CY	BOOL				(CY)
P_Cycle_Time_Error	BOOL				времени цикла
P_Cycle_Time_Value	UDINT				сканирования
P_DM	WORD				сти DM
P_EM0	WORD				сти EM0
P_EM1	WORD				сти EM1
P_EM2	WORD				сти EM2
P_EM3	WORD				сти EM3
P_EM4	WORD				сти EM4
P_EM5	WORD				сти EM5
P_EM6	WORD				сти EM6

В стандартном диалоговом окне сохранения Windows сохраните файл с нужным именем. Файл будет иметь расширение *sxp*.

После этого можно приступить к импортированию каналов в MasterSCADA 4D.

6.2.3.2.8. LOGICA_RSBUS

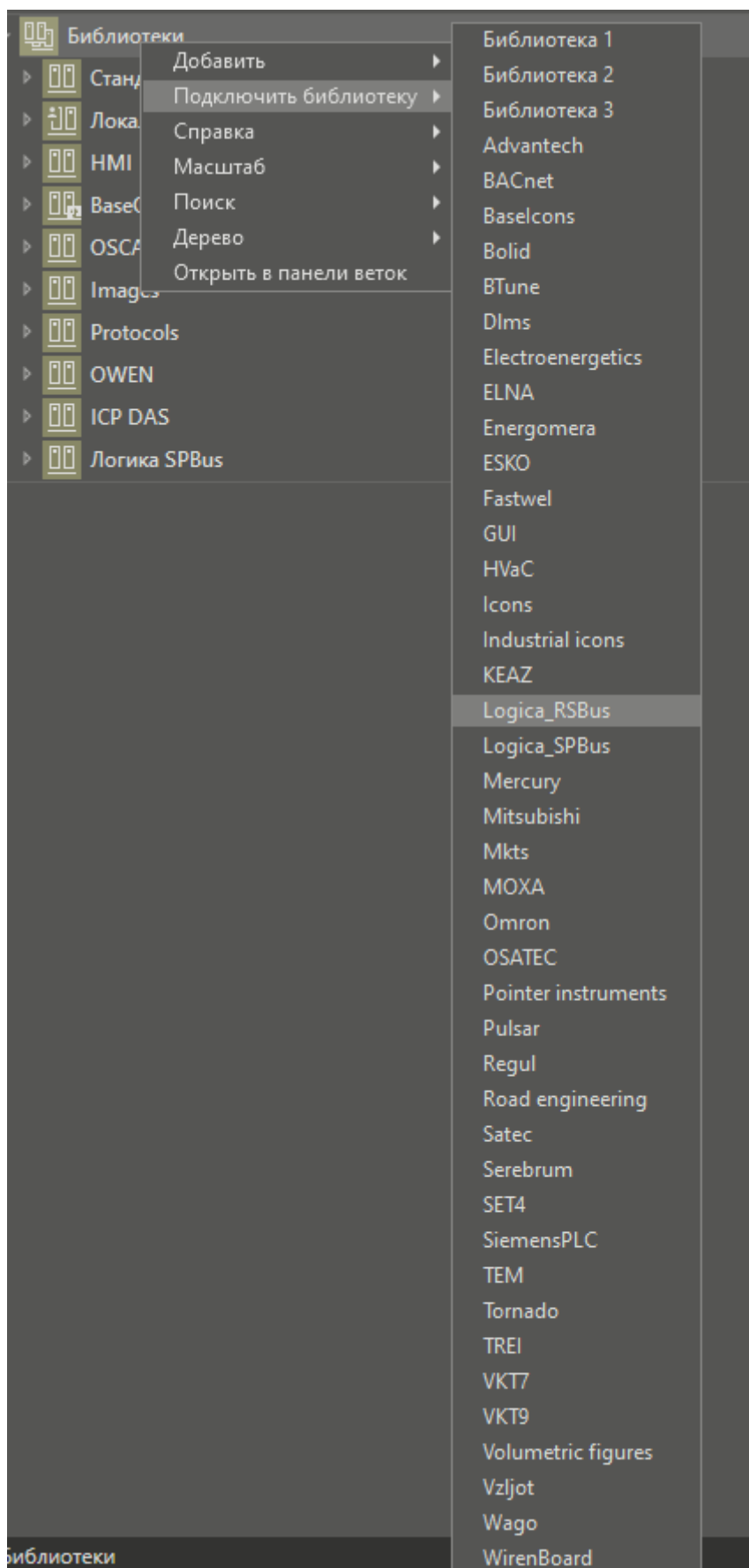
MasterSCADA 4D поддерживает работу с приборами Логика по протоколу RSBUS в исполнительных системах для Windows и для Linux.

Протокол RSBUS может работать через последовательные интерфейсы (RS-232, RS-485, GSM), а при использовании конвертеров Ethernet-COM (Муха NPort 5150 и т.п.) также и по сетям TCP/IP (Ethernet, Wi-Fi).

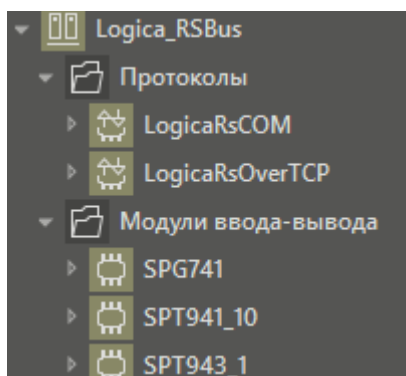
При обмене узел играет роль ведущего.

Важно! По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте sales@masterscada.ru

Для получения возможности использования протокола в проекте необходимо подключить соответствующую библиотеку:

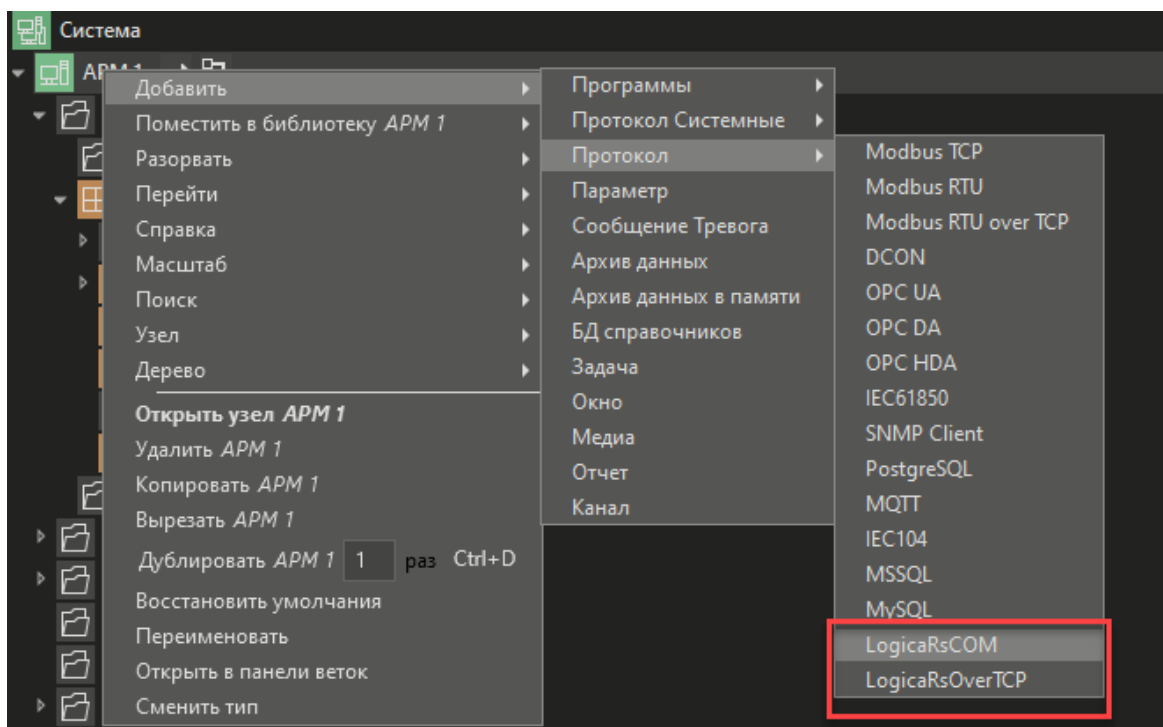


В полном дереве библиотека Logika_RSBus имеет вид:



После этого в проект могут быть добавлены сначала протокол, а затем устройство, любым известным способом, например, при помощи контекстного меню узла, или с использованием контекстной панели, или путем перетаскивания элемента из библиотеки в дерево системы.

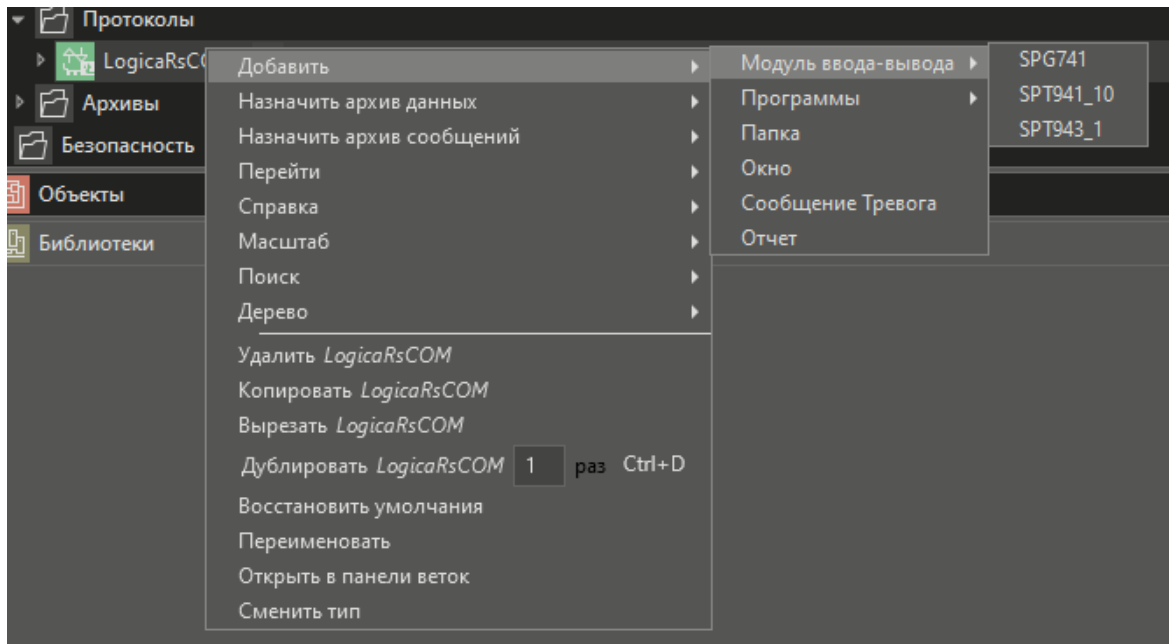
Рекомендуется добавлять элементы в проект при помощи контекстного меню:



Тип добавляемого протокола зависит от реального способа подключения устройств:

- **LogicaRSCom** - если устройство подключено по последовательному интерфейсу. Настройки этого протокола практически полностью соответствуют настройкам Modbus RTU. Добавлено свойство Реинициализация порта при ошибке - при отказе устройства порт будет закрыт и открыт снова.
- **LogicaRsOverTCP** - если счетчик подключен через конвертер Ethernet-COM. Настройки этого протокола соответствуют настройкам Modbus RTU over TCP

Затем при помощи контекстного меню протокола можно добавить нужный модуль ввода вывода:



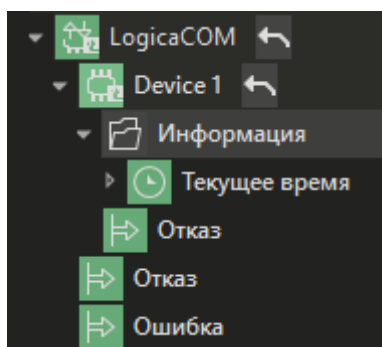
Поддержаны модули:

- СПГ741
- СПТ941.10
- СПТ941.3

Каждый модуль ввода-вывода имеет выход Отказ

Затем необходимо настроить панель свойств модуля. Свойства каналов заданы в соответствии с документацией приборов, но их можно изменить в их панели свойств.

В дереве системы в упрощенном дереве элемент имеет вид:



Устройство имеет специальную группу Информация с каналом Текущее время, в котором отображается время, установленное в приборе.

6.2.3.2.8.1. СВОЙСТВА МОДУЛЕЙ LOGICARSBUS

Вид панели свойств устройства Logica_RSBus:

SPG741 1

🔍 📄
☰ 🏠 🔑

Общие ▲

Имя	SPG741 1
Полное имя	Система.APM 1.Протоколы.LogicaRsCOM.SPG741
Метки	
Комментарий	Корректор расход газа СПГ741

Служебные ▼

Задача ▼

Отношения ▲

Ссылается	Библиотеки.Logica_RSBus.Модули ввода-вывода.!
-----------	---

Масштабирование ▼

Настройки ▲

Адрес	0
Глубина часового архива	24
Глубина суточного архива	30
Глубина месячного архива	12
Количество архивных записей за цикл	5

Настройки каналов ▼

Сервисные ▲

Размер архива	64
Адрес времени	243
Код устройства	18217

Описание характерных свойств:

Название	Описание
Категория Настройки	
Адрес	Адрес устройства

Название	Описание
Количество архивных записей за цикл	Количество считываемых архивных записей за цикл опроса.
Глубина часового архива	Количество считываемых записей часового архива при старте опроса. 0 - все записи
Глубина суточного архива	Количество считываемых записей суточного архива при старте опроса. 0 - все записи.
Глубина месячного архива	Количество считываемых записей месячного архива при старте опроса. 0 - все записи
Категория Настройки каналов	Задаются значения по умолчанию для создаваемых каналов. В настройке не нужны, т.к. индивидуальные настройки каналов формируются автоматически.
Категория Сервисные	
Размер архива	Размер архивной записи в байтах. У СПГ741 СПТ943 - 64, у СПТ941 - 68
Адрес времени	Адрес ОЗУ, где находятся регистры времени.
Код устройства	При установлении связи прибор сообщает его идентификатор, если не совпадает с тем, что указано у устройства, то опрос прекращается. Все параметры каналов имеют признак качества BAD, в лог-файлы исполнительной системы выводится сообщение: Incorrect code device. Set Заданное Get Полученное Можно задать в 0 - тогда код будет проигнорирован.

Важно! Свойства категории Сервисная менять не рекомендуется.

6.2.3.2.8.2. СВОЙСТВА КАНАЛОВ LOGICA RSBUS

Вид панели свойств канала устройства Logica_RSBus:

The screenshot shows a configuration window for a variable named 'P'. The window is divided into several sections:

- Общие (General):**
 - Доступ (Access): Input
 - Имя (Name): P
 - Полное имя (Full name): Система.APM 1.Протоколы.LogicaRsCOM.SPG741
 - Метки (Tags):
 - Комментарий (Comment): Давление
- Служебные (Service):** (Collapsed)
- Шкала (Scale):** (Collapsed)
- Архивирование (Archiving):**
 - Архивировать (Archive):
 - Шаблон архивирования (Archiving template):
- Масштабирование (Scaling):** (Collapsed)
- Настройки (Settings):**
 - Тип переменной (Variable type): Текущее
 - Тип данных (Data type): Float
 - Адрес1 (Address1): 552
 - Адрес2 (Address2): 0

Описание характерных свойств:

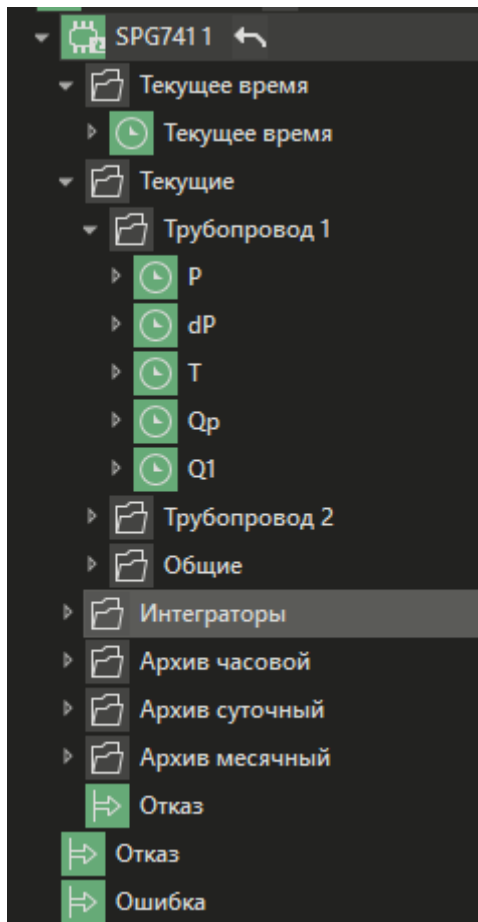
Название	Назначение
Тип переменной	<p>Определяется тип канала:</p> <ul style="list-style-type: none"> • Текущее • Интегратор • Архив часовой • Архив суточный • Архив месячный
Тип данных	Задается тип данных тега, который соответствует данному каналу

Название	Назначение
Адрес 1	Основной адрес тега
Адрес 2	Дополнительный адрес (для интегратора)

Если конфигурация прибора была добавлена при помощи диалога импорта, то настройки каналов заполняются автоматически/

6.2.3.2.8.3. SPG741

Модуль СПГ741 фирмы Логика, работающий по протоколу LogicaRSBus в дереве MasterSCADA 4D имеет вид:



Описание каналов:

Название	Назначение
Группа Текущее время	

Название	Назначение
Текущее время	Отображается текущее время прибора
Группа Текущие.Трубопровод 1(2)	
P	Давление
dP	Перепад давления
T	Температура
Qp	Рабочий расход
Q1	Стандартный расход
Гр	
Группа Текущие.Общие	
dP	Перепад давления
P6	Атмосферное давление
P3	Давление
P4	Давление
T3	Температура
Группа Интеграторы	
Vp1	Тотальный объем в рабочих условиях по трубе 1
Vp2	Тотальный объем в рабочих условиях по трубе 2
V1	Тотальный объем в стандартных условиях по трубе 1

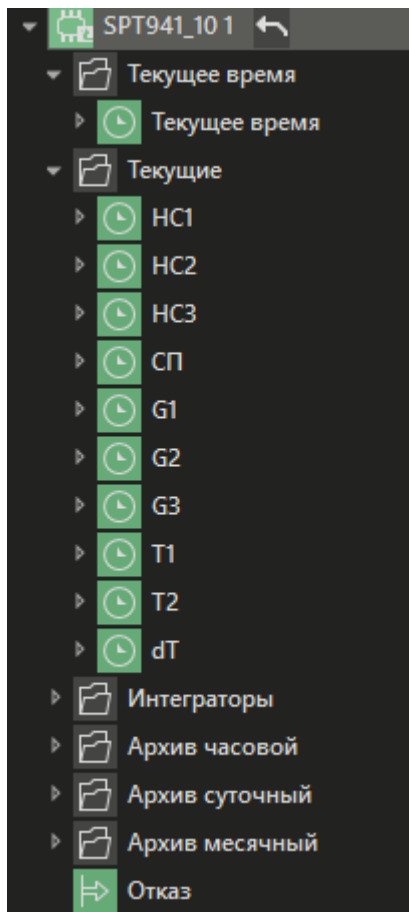
Название	Назначение
V2	Тотальный объем в стандартных условиях по трубе 2
V	Тотальный объем в стандартных условиях по потребителю
ТС	Тотальное время интегрирования
Группы Архив часовой/суточный/месячный	
ТС	Время счета
НС	Сборка нештатных ситуаций за интервал архивирования
P1	Среднее давление газа
T1	Средняя температура газа
Vp1	Интегральный объем газа в рабочих условиях
V1	Интегральный объем газа, приведенный к стандартным условиям
P2	Среднее давление газа
T2	Средняя температура газа
Vp2	Интегральный объем газа в рабочих условиях
V2	Интегральный объем газа, приведенный к стандартным условиям
V	Суммарный объем газа, приведенный к стандартным условиям
Vп	Суммарный объем газа, приведенный к стандартным условиям, израсходованный сверх нормы поставки

Каждый модуль ввода-вывода имеет выход Отказ

У параметров каналов отвечающих за передачу архивных данных флаг Архивировать установлен по умолчанию.

6.2.3.2.8.4. SPT941_10

Модуль СПГ941.10 фирмы Логика, работающий по протоколу LogicaRSBus в дереве MasterSCADA 4D имеет вид:



Описание каналов:

Название	Назначение
Группа Текущее время	
Текущее время	Отображается текущее время прибора
Группа Текущие.	
НС1	Флаги нештатной ситуации

Название	Назначение
НС2	Флаги нештатной ситуации
НС3	Флаги нештатной ситуации
СП	Схема учета
G1	Объемный расход теплоносителя по трубопроводу 1
G2	Объемный расход теплоносителя по трубопроводу 2
G3	Объемный расход теплоносителя по трубопроводу 3
T1	Температура теплоносителя по трубопроводу 1
T2	Температура теплоносителя по трубопроводу 2
dT	Разность температур
Группа Интеграторы	
V1	Объем
V2	Объем
V3	Объем
M1	Масса
M2	Масса
M3	Масса
Q	Масса
ti	Время интегрирования

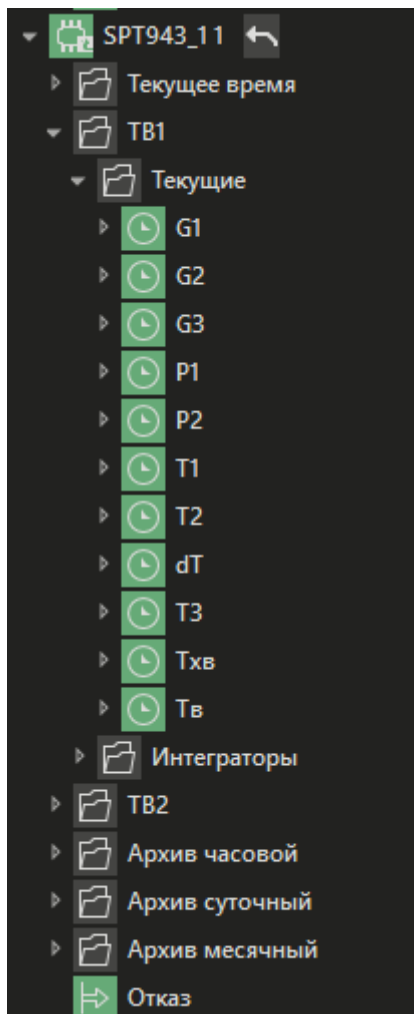
Название	Назначение
Группы Архив часовой/суточный/месячный	
НС1	Флаги нештатной ситуации
НС2	Флаги нештатной ситуации
НС3	Флаги нештатной ситуации
СП	Схема потребления
T1	Средняя температура в трубе 1
T2	Средняя температура в трубе 2
V1	Суммарный объем теплоносителя по трубе 1
V2	Суммарный объем теплоносителя по трубе 2
V3	Суммарный объем теплоносителя по трубе 3
M1	Суммарная масса теплоносителя по трубе 1
M2	Суммарная масса теплоносителя по трубе 2
M3	Суммарная масса теплоносителя по трубе 3
Q	Суммарная тепловая энергия
ti	Суммарное время счета

Каждый модуль ввода-вывода имеет выход Отказ

У параметров каналов отвечающих за передачу архивных данных флаг Архивировать установлен по умолчанию.

6.2.3.2.8.5. SPT943_1

Модуль СПГ943.1 фирмы Логика, работающий по протоколу LogicaRSBus в дереве MasterSCADA 4D имеет вид:



Описание каналов:

Название	Назначение
Группа Текущее время	
Текущее время	Отображается текущее время прибора
Группы ТВ1(2).Текущие	
G1	Объемный расход теплоносителя по трубопроводу 1
G2	Объемный расход теплоносителя по трубопроводу 2

Название	Назначение
G3	Объемный расход теплоносителя по трубопроводу 3
P1	Давление теплоносителя в трубе 1
P2	Давление теплоносителя в трубе 2
T1	Температура теплоносителя по трубопроводу 1
T2	Температура теплоносителя по трубопроводу 2
dT	Разность температур
T3	Температура теплоносителя по трубопроводу 3
Tхв	Температура холодной воды
Tв	Температура воздуха
Группа ТВ1(2).Интеграторы	
V1	Объем
V2	Объем
V3	Объем
M1	Масса
M2	Масса
M3	Масса
Q	Масса
ti	Время интегрирования

Название	Назначение
Qr	Тепловая энергия ГВС
Группы Архив часовой/суточный/месячный	
НС1	Флаги нештатной ситуации
НС2	Флаги нештатной ситуации
НС3	Флаги нештатной ситуации
СП	Схема потребления
P1	Средняя давление в трубе 1
P2	Средняя давление в трубе 2
T1	Средняя температура в трубе 1
T2	Средняя температура в трубе 2
T3	Средняя температура в трубе 3
Тхв	Температура холодной воды
Тв	Температура воздуха
V1	Суммарный объем теплоносителя по трубе 1
V2	Суммарный объем теплоносителя по трубе 2
V3	Суммарный объем теплоносителя по трубе 3
M1	Суммарная масса теплоносителя по трубе 1
M2	Суммарная масса теплоносителя по трубе 2

Название	Назначение
M3	Суммарная масса теплоносителя по трубе 3
Q	Суммарная тепловая энергия
ti	Суммарное время счета
Qг	Тепловая энергия в трубе ГВС

Каждый модуль ввода-вывода имеет выход Отказ

У параметров каналов отвечающих за передачу архивных данных флаг Архивировать установлен по умолчанию.

6.2.3.2.9. LOGICA_SPBUS

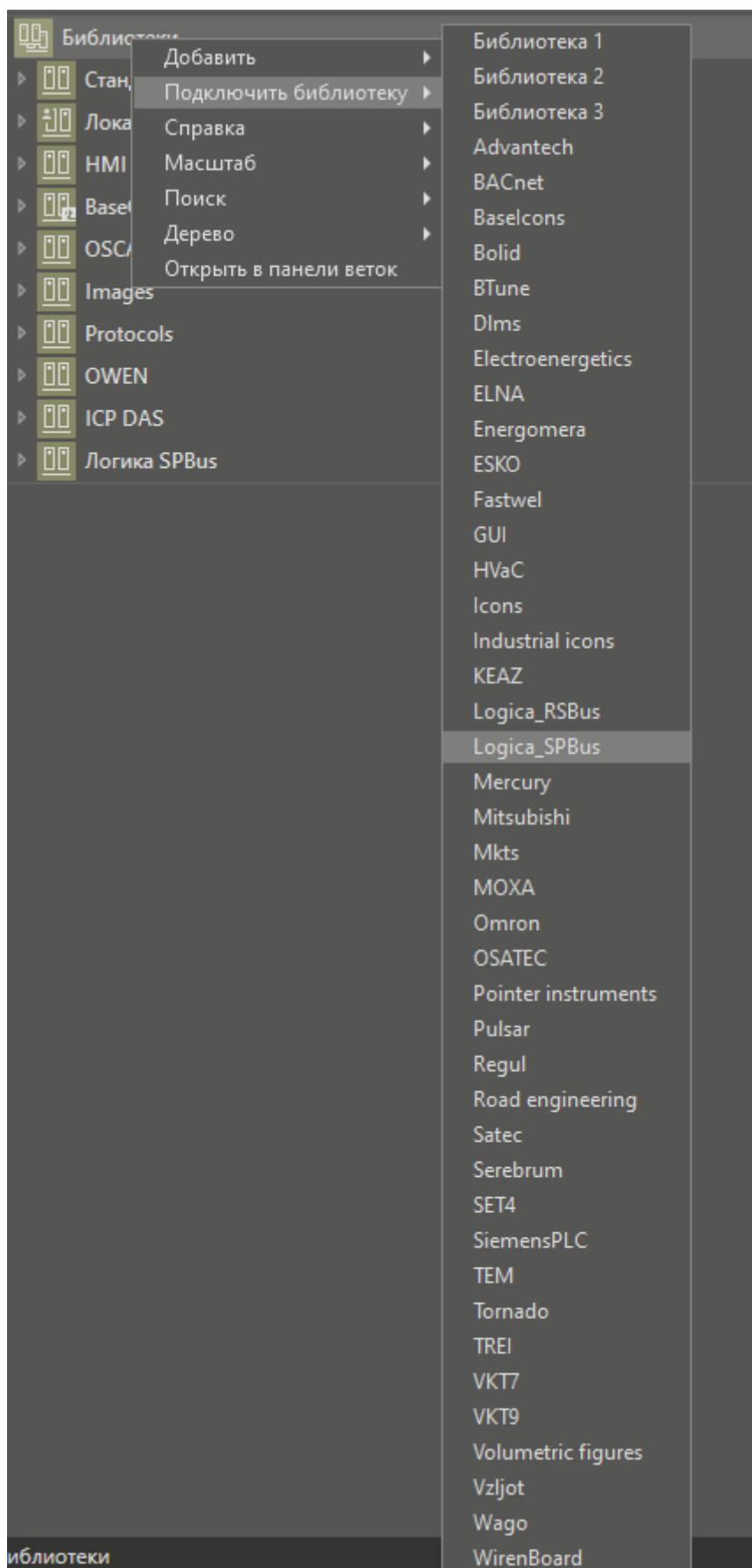
MasterSCADA 4D поддерживает работу с приборами Логика по протоколу SPBus в исполнительных системах для Windows и для Linux.

Протокол SPbus может работать через последовательные интерфейсы (RS-232, RS-485, GSM), а при использовании конвертеров Ethernet-COM (Моха NPort 5150 и т.п.) также и по сетям TCP/IP (Ethernet, Wi-Fi).

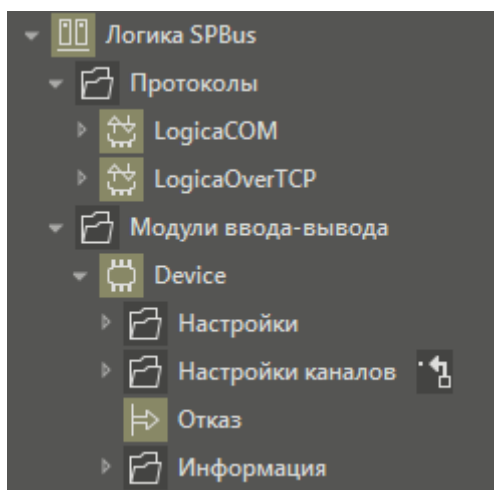
При обмене узел играет роль ведущего.

Важно! По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте sales@masterscada.ru

Для получения возможности использования протокола в проекте необходимо подключить соответствующую библиотеку:

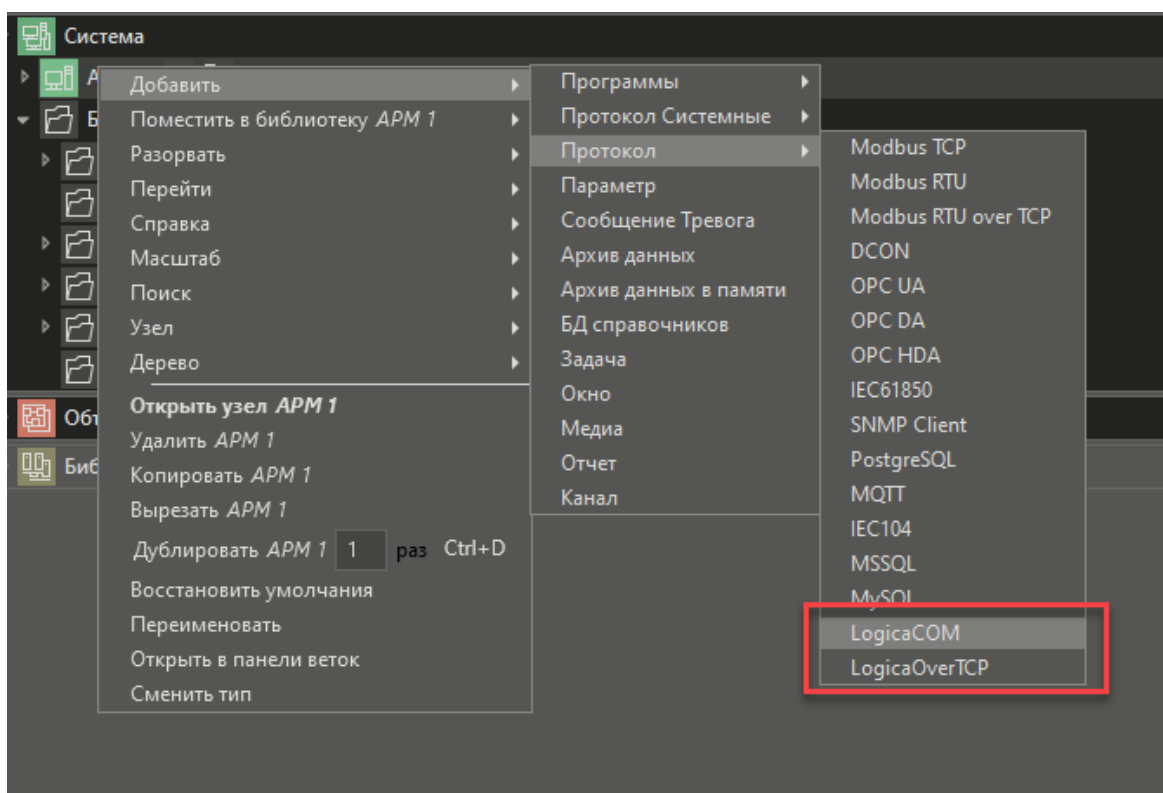


В полном дереве библиотека Логика SPBus имеет вид:



После этого в проект могут быть добавлены сначала протокол, а затем устройство, любым известным способом, например, при помощи контекстного меню узла, или с использованием контекстной панели, или путем перетаскивания элемента из библиотеки в дерево системы.

Рекомендуется добавлять элементы в проект при помощи контекстного меню:



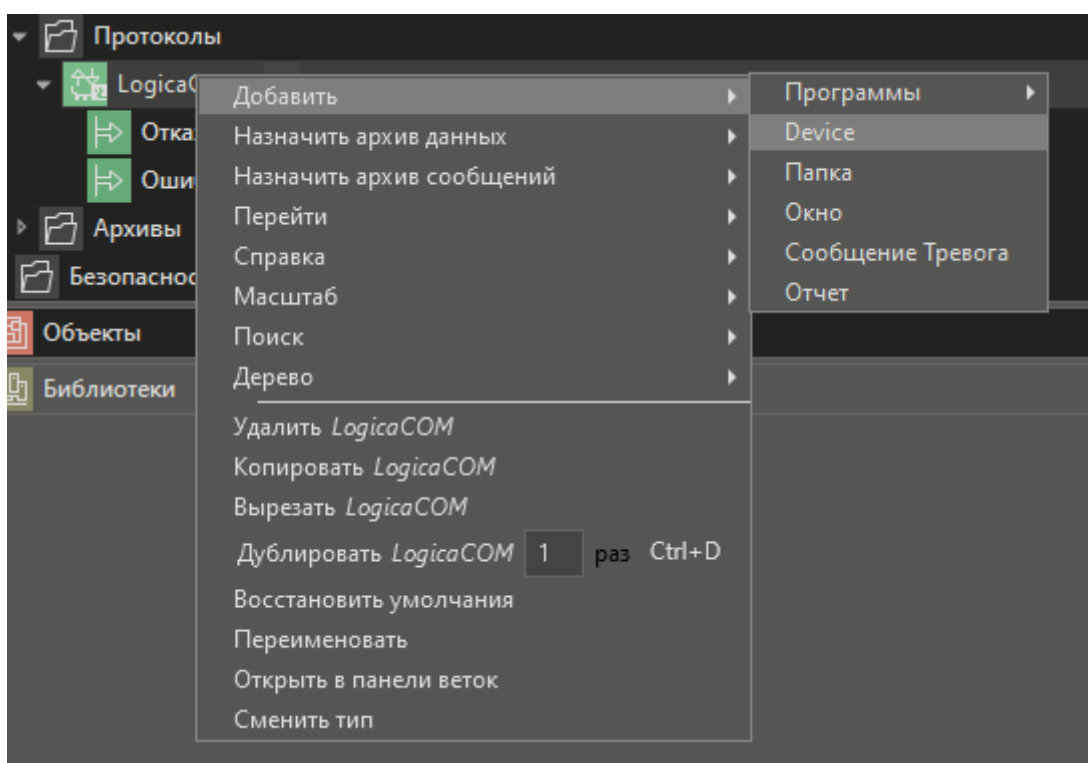
Тип добавляемого протокола зависит от реального способа подключения устройств:

- LogicaCom - если устройство подключено по последовательному интерфейсу. Настройки этого протокола практически полностью соответствуют настройкам Modbus RTU. Добавлено свойство Реинициализация порта при ошибке - при отказе устройства порт будет закрыт и открыт снова.

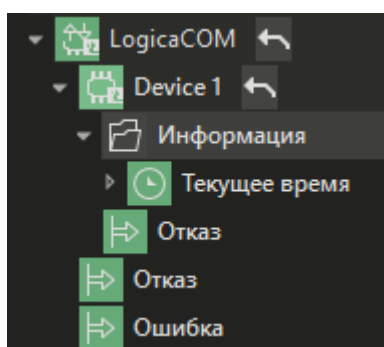
- LogicaOverTCP - если счетчик подключен через конвертер Ethernet-COM. Настройки этого протокола соответствуют настройкам Modbus RTU over TCP

В оба протокола добавлено свойство Собственный адрес - это адрес на шине RS485 от имени которого будут посылаться запросы. Собственный адрес задается только для шины SPbus и только в случае, если предполагается обмен по интерфейсу RS485 без использования маркерного цикла, поддерживаемого адаптером АПС79 и приборами учета. Диапазон собственного адреса - от 0 до 31. Собственный адрес не должен совпадать с другими адресами устройств на шине (т.е. если адреса устройств 1 и 2, то собственный нужно поставить 3 или другой).

Затем при помощи контекстного меню протокола можно добавить устройство:



В дереве системы в упрощенном дереве элемент имеет вид:



Устройство имеет специальную группу Информация с каналом Текущее время, в котором отображается время, установленное в приборе.

Добавить канал в устройство можно при помощи контекстного меню вручную, либо через диалог импорта (рекомендуется).

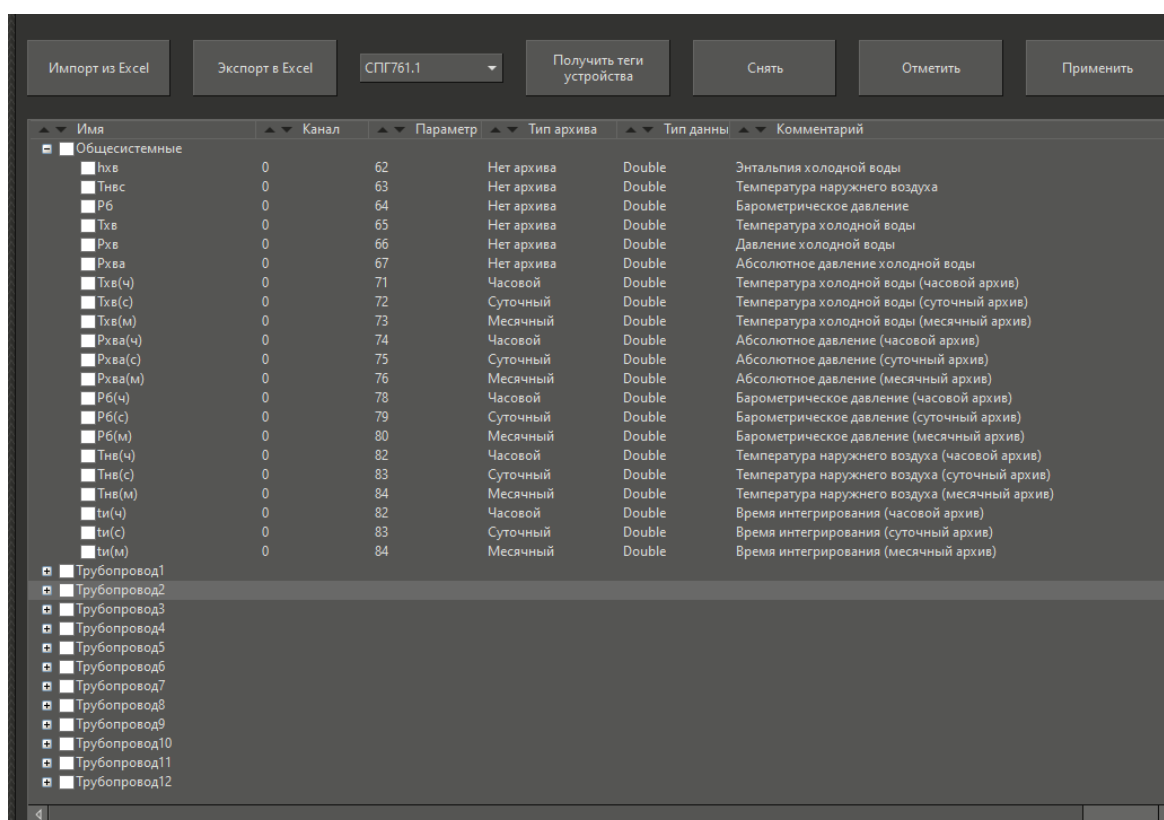
Свойства устройства задаются в его панели свойств.

6.2.3.2.9.1. ДИАЛОГ ИМПОРТА LOGICA_SPBUS

Диалог импорта для устройства протокола Logica_SPBus имеет вид:

Для автоматического добавления каналов и выбора устройств для работы по протоколу Logica_SPBus нужно дважды нажать левую кнопку мыши по устройству в дереве. При этом откроется вкладка, в которой можно выполнить выбор устройства и импорт каналов.

Вид вкладки:



Основную часть вкладки занимает таблица, в которой будут отображаться найденные объекты. Флагами можно отметить необходимые каналы для добавления в дерево проекта

В верхней части вкладки находятся элементы управления.

Описание элементов управления:

Кнопка	Назначение
Импорт из Excel	Открывает диалоговое окно, в котором можно выбрать файл Excel, в котором хранится описание каналов для устройства, которое не указано

Кнопка	Назначение
	в выпадающем списке, описанном ниже или которые были сохранены ранее при помощи кнопки Экспорт в Excel
Экспорт в Excel	Позволяет сохранить в файл выбранную конфигурацию.
Выпадающий список с именами приборов	В выпадающем списке перечислены приборы, для которых добавлена конфигурация каналов. Поддержаны приборы СПТ961.1 и СПТ761.1. Если необходимо добавить в проект другое устройство обратитесь в службу технической поддержки
Получить теги устройства	Добавляет каналы того устройства, которое выбрано в выпадающем списке.
Снять	Снимает выделение с выбранного канала
Отметить	Отмечает выделенный канал
Применить	Добавляет отмеченные элементы в дерево. Настройки каналов получают настройки указанные в конфигурации. У параметров каналов, которые передают архивные значения флаг Архивировать будет установлен автоматически

Важно! После закрытия вкладки значения выбранных тегов не сохраняются в таблице. При следующем открытии таблица будет пустой, при этом каналы в дереве будут сохранены.

6.2.3.2.9.2. СВОЙСТВА УСТРОЙСТВА LOGICA_SPBUS

Вид панели свойств устройства Logica_SPBus:

Device 761

🔍 🏠
☰ 🚩 🔑

Общие ▲

Имя	Device 761
Полное имя	Система.APM 1.Протоколы.LogicaCOM.Device 761
Метки	
Комментарий	Для добавления тегов используйте утилиту импорта

Служебные ▼

Задача ▼

Отношения ▲

Ссылается	Библиотеки.Логика SPBus.Модули ввода-вывода.Device
-----------	--

Масштабирование ▼

Настройки ▲

Адрес	1
Количество считываемых параметров за цикл	10
Глубина часового архива	24
Глубина суточного архива	30
Глубина месячного архива	12
Количество архивных записей за цикл	5

Настройки каналов ▲

Номер канала	0
Номер параметра	0
Тип архива	Нет архива

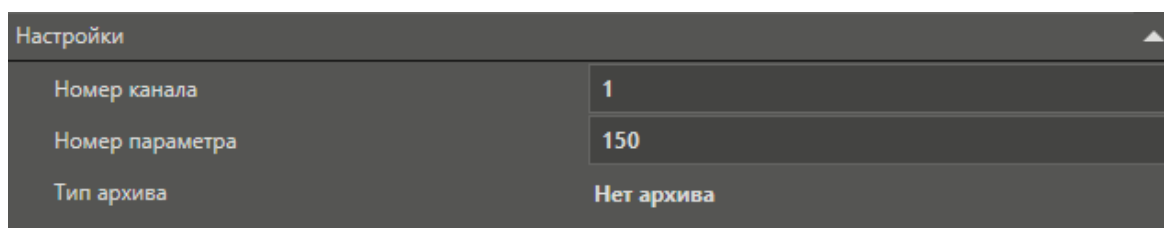
Описание свойств:

Название	Описание
Адрес	Адрес устройства
Количество считываемых параметров за цикл	Количество считываемых архивных записей за цикл опроса. Верхний предел - 25.
Глубина часового архива	Количество считываемых записей часового архива при старте опроса. 0 - все записи

Название	Описание
Глубина суточного архива	Количество считываемых записей суточного архива при старте опроса. 0 - все записи.
Глубина месячного архива	Количество считываемых записей месячного архива при старте опроса. 0 - все записи
Категория Настройки каналов	Задаются значения по умолчанию для создаваемых каналов. В настройке не нуждаются, т.к. индивидуальные настройки каналов формируются автоматически.

6.2.3.2.9.3. СВОЙСТВА КАНАЛОВ УСТРОЙСТВА LOGICA_SPBUS

Вид панели свойств канала устройства Logica_SPBus:



Название	Назначение
Номер канала	Задается в соответствии со структурой прибора и не может назначаться произвольно.
Номер параметра	Задается номер параметра
Тип архива	Определяет из какого архива прибора будет использоваться значение

Если конфигурация прибора была добавлена при помощи диалога импорта, то настройки каналов заполняются автоматически/

6.2.3.2.9.4. УСТРОЙСТВО СПГ761.1

Корректор СПГ 761.2 Логика предназначен для измерения электрических сигналов, соответствующих параметрам природного газа, и последующего вычисления расхода и объема газа, приведенных к стандартным условиям ($T_c=20\text{ }^\circ\text{C}$, $P_c=0,101325\text{ МПа}$).

Если в выпадающем списке диалога импорта каналов был выбран данный тип прибора, то после нажатия кнопки Получить теги устройства в таблице будут доступны следующие каналы для добавления в дерево проекта.

Список возможных каналов:

Название	Назначение
Группа Общесистемные	
Тнвс	Температура наружного воздуха
Рб	Барометрическое давление
Рб(ч)	Архив почасовых значений барометрического давления
Рб(с)	Архив суточных значений барометрического давления
Рб(м)	Архив месячных значений барометрического давления
Тнв(ч)	Архив почасовых значений температуры наружного воздуха
Тнв(с)	Архив суточных значений температуры наружного воздуха
Тнв(м)	Архив месячных значений температуры наружного воздуха
ти(ч)	Время интегрирования (часовой архив)
ти(с)	Время интегрирования (суточный архив)
ти(м)	Время интегрирования (месячный архив)
Группы Трубопровод 1-12	
dP	Результат преобразования измеренных значений перепада давления
dP1	Измеренное значение перепада давления, соответствующее первому (основному) датчику перепада давления

Название	Назначение
dP2	Измеренное значение перепада давления, соответствующее второму (дополнительному) датчику перепада давления
dP3	Измеренное значение перепада давления, соответствующее третьему (дополнительному) датчику перепада давления
P	Измеренное значение перепада давления, соответствующее третьему (дополнительному) датчику перепада давления
Pa	Абсолютное давление (для вычислений)
T	Температура газа
G	Массовый расход
Qp	Объемный расход газа при рабочих условиях
Q	Объемный расход газа при стандартных условиях
M	Масса газа нарастающим итогом
V	Объем газа при стандартных условиях нарастающим итогом
Vp	Объем газа при рабочих условиях нарастающим итогом
Phi	Измеренная относительная влажность
hgi	Измеренная удельная объемная теплота сгорания
Roi	Измеренная плотность
Qo	Измеренный расход влажного газа
Td1	Результат измерений первым дополнительным датчиком по трубопроводу

Название	Назначение
Тд2	Результат измерений вторым дополнительным датчиком по трубопроводу
dP_Qo_g(ч)	Архив часовых средних значений измеряемых перепада давления или расхода
dP_Qo_g(с)	Архив суточных средних значений измеряемых перепада давления или расхода
dP_Qo_g(м)	Архив месячных средних значений измеряемых перепада давления или расхода
T(ч)	Архив часовых значений температуры
T(с)	Архив суточных значений температуры
T(м)	Архив месячных значений температуры
P(ч)	Архив часовых значений абсолютного давления
P(с)	Архив суточных значений абсолютного давления
P(м)	Архив месячных значений абсолютного давления
M(ч)	Архив часовых значений массы газа
M(с)	Архив суточных значений массы газа
M(м)	Архив месячных значений массы газа
V(ч)	Архив часовых объема газа при стандартных условиях
V(с)	Архив суточных объема газа при стандартных условиях
V(м)	Архив месячных объема газа при стандартных условиях

Название	Назначение
Vp(ч)	Архив часовых значений объема газа при рабочих условиях
Vp(с)	Архив суточных значений объема газа при рабочих условиях
Vp(м)	Архив месячных значений объема газа при рабочих условиях
hg(ч)	Архив часовых значений средневзвешенной удельной объемной теплоты
hg(с)	Архив суточных значений средневзвешенной удельной объемной теплоты
hg(м)	Архив месячных значений средневзвешенной удельной объемной теплоты
Д1(ч)	Архив часовых средних значений параметра, измеряемого первым дополнительным датчиком по трубопроводу
Д1(с)	Архив суточных средних значений параметра, измеряемого первым дополнительным датчиком по трубопроводу
Д1(м)	Архив месячных средних значений параметра, измеряемого первым дополнительным датчиком по трубопроводу
Д2(ч)	Архив часовых средних значений параметра, измеряемого вторым дополнительным датчиком по трубопроводу
Д2(с)	Архив суточных средних значений параметра, измеряемого вторым дополнительным датчиком по трубопроводу
Д2(м)	Архив месячных средних значений параметра, измеряемого вторым дополнительным датчиком по трубопроводу

6.2.3.2.9.5. УСТРОЙСТВО СПГ961.1

Тепловычислители СПТ 961 Логика предназначены для измерения электрических сигналов, соответствующих параметрам теплоносителя, с последующим расчетом тепловой энергии и количества теплоносителя. Тепловычислители рассчитаны на применение в составе теплосчетчиков для водяных и паровых систем теплоснабжения и иных

измерительных систем, где в качестве теплоносителя используются вода, конденсат, перегретый пар либо сухой или влажный насыщенный пар.

Если в выпадающем списке диалога импорта каналов был выбран данный тип прибора, то после нажатия кнопки **Добавить** теги устройства в таблице будут доступны следующие каналы для добавления в дерево проекта.

Список возможных каналов:

Название	Назначение
Группа Общесистемные	
hхв	Энтальпия холодной воды
Тнвс	Температура наружного воздуха
Рб	Барометрическое давление
Тхв	Температура холодной воды
Рхв	Давление холодной воды
Рхва	Абсолютное давление холодной воды
Тхв(ч)	Температура холодной воды (часовой архив)
Тхв(с)	Температура холодной воды (суточный архив)
Тхв(м)	Температура холодной воды (месячный архив)
Рхва(ч)	Абсолютное давление (часовой архив)
Рхва(с)	Абсолютное давление (суточный архив)
Рхва(м)	Абсолютное давление (месячный архив)
Рб(ч)	Барометрическое давление (часовой архив)

Название	Назначение
Рб(с)	Барометрическое давление (суточный архив)
Рб(м)	Барометрическое давление (месячный архив)
Тнв(ч)	Температура наружного воздуха (часовой архив)
Тнв(с)	Температура наружного воздуха (суточный архив)
Тнв(м)	Температура наружного воздуха (месячный архив)
ти(ч)	Время интегрирования (часовой архив)
ти(с)	Время интегрирования (суточный архив)
ти(м)	Время интегрирования (месячный архив)
Группы Трубопровод 1-12	
dP	Результат преобразования измеренных значений перепада давления
dP1	Перепад давления на основном датчике
dP2	Перепад давления на втором датчике
dP3	Перепад давления на третьем датчике
P	Измеренное давление теплоносителя
Pa	Абсолютное давление теплоносителя (для вычислений)
T	Температура теплоносителя
G	Массовый расход теплоносителя
w	Тепловая мощность по трубопроводу

Название	Назначение
M	Масса теплоносителя нарастающим итогом
Ws	Тепловая энергия нарастающим итогом
Vo	Объем теплоносителя нарастающим итогом
Qo_g	Измеренный расход
Дт1	Результат измерений первым дополнительным датчиком по трубопроводу
Дт2	Результат измерений вторым дополнительным датчиком по трубопроводу
dP_Qo_g(ч)	Архив часовых средних значений измеряемых перепада давления или расхода теплоносителя
dP_Qo_g(с)	Архив суточных средних значений измеряемых перепада давления или расхода теплоносителя
dP_Qo_g(м)	Архив месячных средних значений измеряемых перепада давления или расхода теплоносителя
T(ч)	Архив часовых значений температуры теплоносителя
T(с)	Архив суточных значений температуры теплоносителя
T(м)	Архив месячных значений температуры теплоносителя
P(ч)	Архив часовых значений давления теплоносителя
P(с)	Архив суточных значений давления теплоносителя
P(м)	Архив месячных значений давления теплоносителя
M(ч)	Архив часовых значений массы транспортированного теплоносителя

Название	Назначение
M(с)	Архив суточных значений массы транспортированного теплоносителя
M(м)	Архив месячных значений массы транспортированного теплоносителя
W(ч)	Архив часовых значений тепловой энергии
W(с)	Архив суточных значений тепловой энергии
W(м)	Архив месячных значений тепловой энергии
Vo(ч)	Архив часовых значений объема транспортированного теплоносителя
Vo(с)	Архив суточных значений объема транспортированного теплоносителя
Vo(м)	Архив месячных значений объема транспортированного теплоносителя
ДТ1(ч)	Архив часовых средних значений параметра, измеряемого первым дополнительным датчиком по трубопроводу
ДТ1(с)	Архив суточных средних значений параметра, измеряемого первым дополнительным датчиком по трубопроводу
ДТ1(м)	Архив месячных средних значений параметра, измеряемого первым дополнительным датчиком по трубопроводу
ДТ2(ч)	Архив посуточный средних значений параметра, измеряемого первым дополнительным датчиком по трубопроводу
ДТ2(с)	Архив суточных средних значений параметра, измеряемого первым дополнительным датчиком по трубопроводу
ДТ2(м)	Архив месячных средних значений параметра, измеряемого первым дополнительным датчиком по трубопроводу

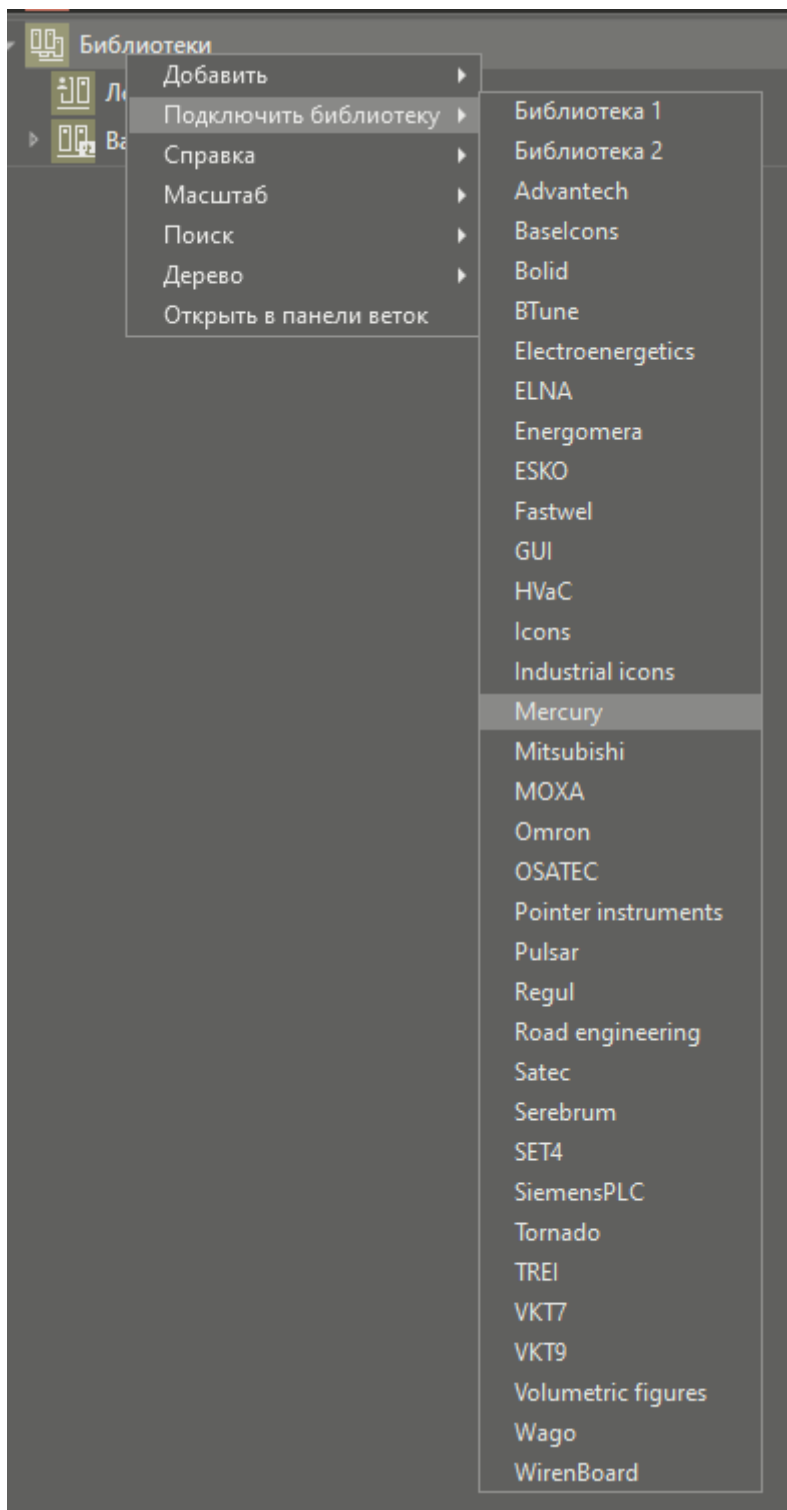
6.2.3.2.10.MERCURY

MasterSCADA 4D поддерживает работу со счетчиками электроэнергии Меркурий 230, 233, 234, 236 компании Инкотекс.

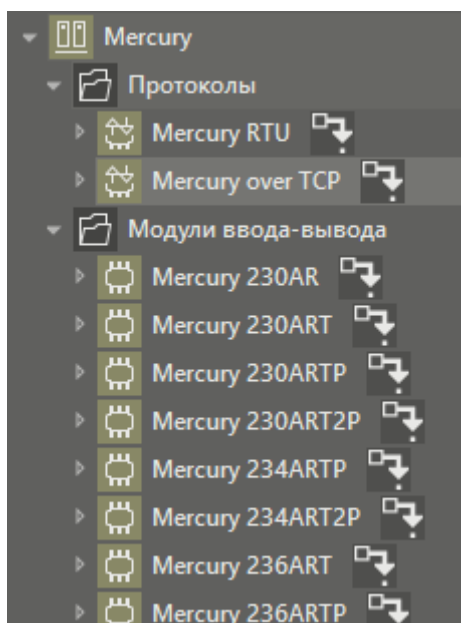
При обмене узел играет роль ведущего. Поддерживается считывание архивов из счетчика. Работа с этими устройствами возможна через последовательные интерфейсы (RS-232, RS-485, GSM, CAN), а при использовании конвертеров Ethernet-COM (Моха NPort 5150 и т.п.) также и по сетям TCP/IP (Ethernet, Wi-Fi).

Важно! По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте sales@masterscada.ru

Для получения возможности использования модулей ввода-вывода Mercury в проекте необходимо подключить соответствующую библиотеку:

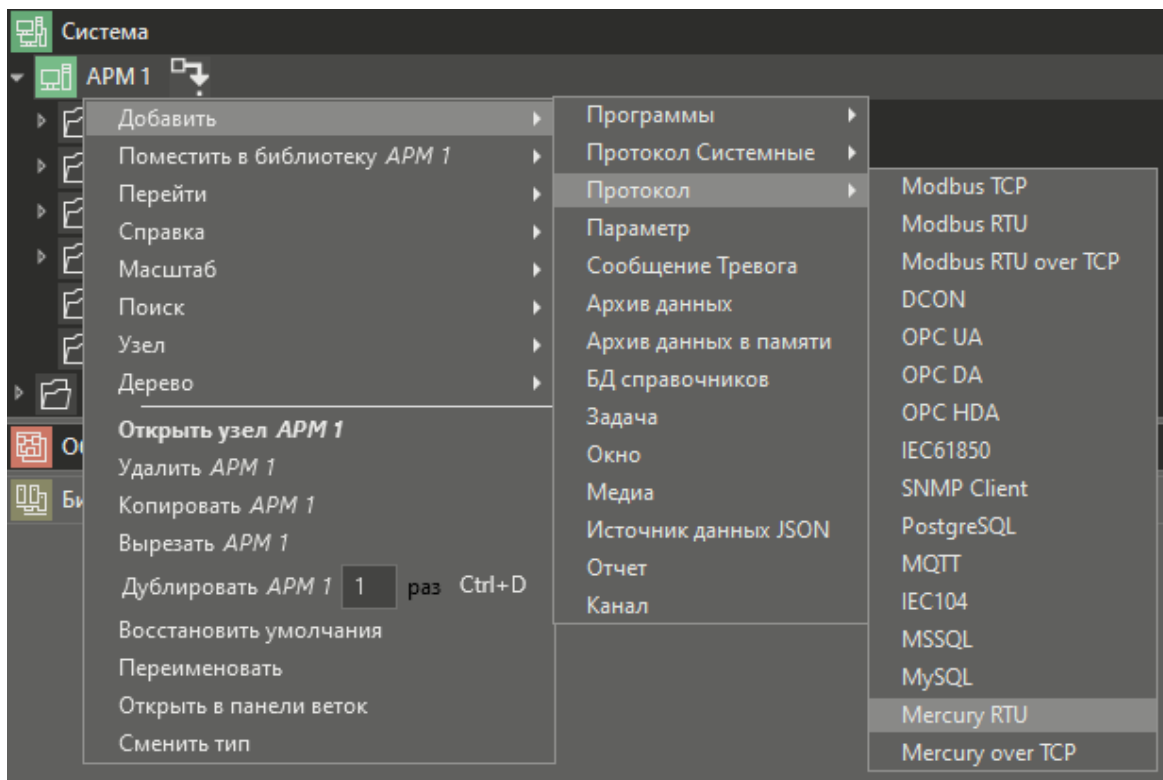


В полном дереве библиотека Mercury имеет вид:



После этого в проект могут быть добавлены сначала протокол, а затем устройство, любым известным способом, например, при помощи контекстного меню узла, или с использованием контекстной панели, или путем перетаскивания элемента из библиотеки в дерево системы.

Рекомендуется добавлять элементы в проект при помощи контекстного меню:



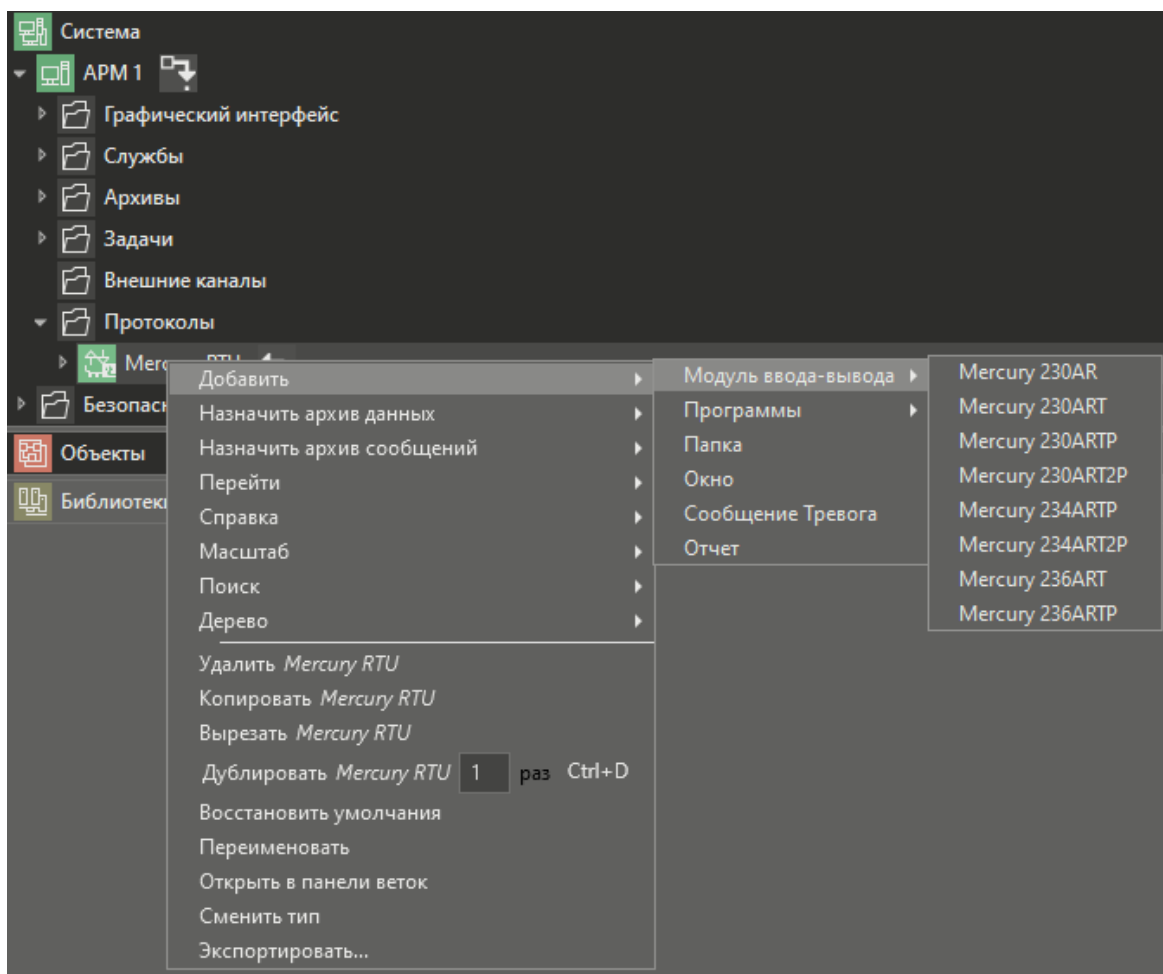
Тип добавляемого протокола зависит от реального способа подключения счетчика:

- Mercury RTU - если счетчик подключен по последовательному интерфейсу. Настройки этого протокола практически полностью соответствуют настройкам

Modbus RTU. В настройки протокола Mercury RTU добавлена настройка Режим "Эхо" - ее необходимо включить если опрос счетчика производится по шине CAN через конвертер Меркурий-221.

- Mercury over TCP - если счетчик подключен через конвертер Ethernet-COM. Настройки этого протокола соответствуют настройкам Modbus RTU over TCP

Затем при помощи контекстного меню протокола можно добавить модули ввода-вывода, которые уже содержат все необходимые каналы:



Счетчики Меркурий отличаются наличием возможности работы с теми или иными параметрами – количеством направлений учета, наличием тарифов, профилями мощности, журналами. Чтобы определить возможности счетчиков, нужно проанализировать буквенные и цифровые обозначения, следующие после номера модели (230, 233, 234, 236):

- А – измерение активной энергии;
- R – измерение реактивной энергии;
- Т – наличие тарификатора;
- 2 – количество направлений учета. Если цифра отсутствует – только прямое направление;
- P – профиль мощности и журнал сообщений.

Таким образом, счетчик Mercury 230ART является счетчиком серии 230 с поддержкой измерения активной и реактивной мощности в одном направлении, тарификатором. Профиль мощности не поддерживается.

Каждый счетчик имеет те группы каналов, которые характерны для его работы.

Группа	Назначение
Counter_Param	Параметры счетчика
Measure	Текущие значения напряжения, тока и частоты
Power	Значения мощности
Energy	Накопленные энергии
PowerProfile	Профили мощности. Каналы предназначены для передачи архивных значений счетчика – профиля мощности.
Synchronization time	Параметры синхронизации времени.

Поддержаны наиболее используемые функции счетчиков.

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если хоть у одного параметра канала, входящего в состав группы PowerProfile или Energy, снять флаг Архивировать, то выключается из опроса вся эта группа целиком.

Поддерживаются следующие модули ввода-вывода:

- Mercury 230ART
- Mercury 230ARTP
- Mercury 230ART2P
- Mercury 234ARTP
- Mercury 234ART2P
- Mercury 236ART
- Mercury 236ARTP

После добавления в дерево проекта модулей ввода-вывода, настраивается панель свойств модулей.

Каждый модуль ввода-вывода имеет выход Отказ, а также группы каналов, согласно его максимальной конфигурации.

Если в опросе каких-либо каналов нет необходимости, то можно либо удалить их из конфигурации элемента уже после добавления его в дерево системы, либо сначала создать экземпляр модуля в пользовательской библиотеке, затем из его конфигурации удалить незадействованные каналы, после чего использовать в проекте уже новый видоизмененный библиотечный элемент.

6.2.3.2.10.1. СВОЙСТВА МОДУЛЕЙ MERCURY

Вид панели свойств:

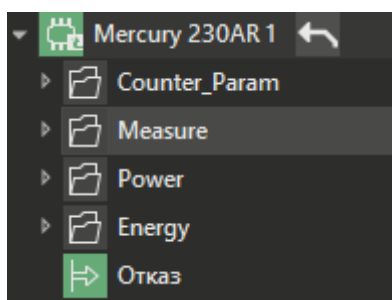
Описание:

Название	Описание
Категория Общие	Как правило, для данного элемента эта категория не настраивается. Подробное описание данной категории приведено в соответствующем разделе.

Название	Описание
Категория Задача	Задается способ опроса устройства. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи.
Категория Масштабирование	Если модуль ввода-вывода не имеет встроенного масштабирования, то при помощи данной категории можно задать соответствие между значениями в модуле и реальными значениями. Подробное описание данной категории приведено в соответствующем разделе.
Адрес	Задается адрес устройства в сети (значение адреса определяется документацией на подключаемое устройство). Заводским значением адреса, у счетчиков Меркурий являются последние 3 цифры серийного номера прибора, а если число превышает 240, то последним двум цифрам
Пароль	Задается пароль доступа к данным счетчика
HEX пароль	Определяется способ
Количество считываемых записей профиля мощности за цикл	Определяет количество считываемых каждый цикл опроса записей профиля мощности (для архивированных каналов группы PowerProfile).

6.2.3.2.10.2. MERCURY 230AR

Вид счетчика добавленного в протокол в дереве системы:



Состав каналов счетчика:

Название	Назначение
Группа Counter_Param	
SerialNumber	Указывается серийный номер счетчика
Группа Measure	
Группа U	Содержит каналы предназначенные для получения текущих значений напряжения для каждой фазы:
A	Фаза А
B	Фаза В
C	Фаза С
Группа I	Содержит каналы предназначенные для получения текущих значений тока для каждой фазы.
A	Фаза А
B	Фаза В
C	Фаза С
F	Служит для получения значения частоты
Группа Power	
Группа P	Содержит каналы предназначенные для получения текущих значений активной мощности для каждой фазы.
SUM	Суммарное значение по всем фазам
A	Фаза А
B	Фаза В
C	Фаза С
Группа Q	Содержит каналы предназначенные для получения текущих значений реактивной мощности для каждой фазы.
SUM	

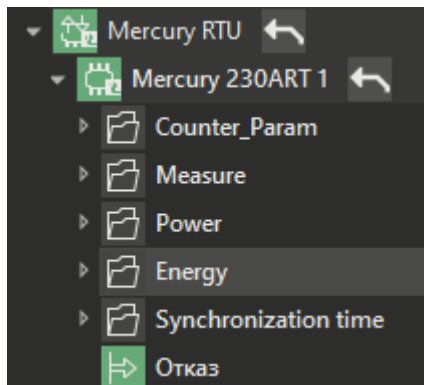
Название	Назначение	
A	Суммарное значение по всем фазам	
B		Фаза А
C		Фаза В Фаза С
Группа S	Содержит каналы предназначенные для получения текущих значений полной мощности для каждой фазы.	
SUM	Суммарное значение по всем фазам	
A	Фаза А	
B	Фаза В	
C	Фаза С	
Группа Energy		
Группа FromReset.TarifSum	Содержит накопленное значение энергии с момента сброса счетчика	
m	Для активной активной энергии прямого направления	
A+	Для реактивной энергии прямого направления	
R+		

Каждый модуль ввода-вывода имеет выход Отказ

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если хоть у одного параметра канала, входящего в состав групп PowerProfile или Energy, снять флаг Архивировать, то выключается из опроса вся эта группа целиком.

6.2.3.2.10.3. MERCURY 230ART

Вид счетчика добавленного в протокол в дереве системы:



Состав каналов счетчика:

Название	Назначение
Группа Counter_Param	
SerialNumber	Указывается серийный номер счетчика
CurrentTime	Показывает текущее время, установленное в счетчике
Группа Measure	
Группа U	Содержит каналы предназначенные для получения текущих значений напряжения для каждой фазы:
A	Фаза A
B	Фаза B
C	Фаза C
Группа I	Содержит каналы предназначенные для получения текущих значений тока для каждой фазы.
A	Фаза A
B	Фаза B
C	Фаза C
F	Служит для получения значения частоты
Группа Power	

Название	Назначение
<p>Группа P</p> <p>SUM</p> <p>A</p> <p>B</p> <p>C</p>	<p>Содержит каналы предназначенные для получения текущих значений активной мощности для каждой фазы.</p> <p>Суммарное значение по всем фазам</p> <p>Фаза A</p> <p>Фаза B</p> <p>Фаза C</p>
<p>Группа Q</p> <p>SUM</p> <p>A</p> <p>B</p> <p>C</p>	<p>Содержит каналы предназначенные для получения текущих значений реактивной мощности для каждой фазы.</p> <p>Суммарное значение по всем фазам</p> <p>Фаза A</p> <p>Фаза B</p> <p>Фаза C</p>
<p>Группа S</p> <p>SUM</p> <p>A</p> <p>B</p> <p>C</p>	<p>Содержит каналы предназначенные для получения текущих значений полной мощности для каждой фазы.</p> <p>Суммарное значение по всем фазам</p> <p>Фаза A</p> <p>Фаза B</p> <p>Фаза C</p>
Группа Energy	
Группа FromReset	<p>Содержит накопленное значение энергии с момента сброса счетчика для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>

Название	Назначение
Группа Year_now	<p>Содержит накопленное значение энергии за текущий год для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
Группа Year_before	<p>Содержит накопленное значение энергии за предыдущий год для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
Группа Day_now	<p>Содержит накопленное значение энергии за текущие сутки для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
Группа Day_before	<p>Содержит накопленное значение энергии за предыдущие сутки для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
Группа Month	<p>Задержит архив накопленных месячных значений для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится три канала.</p> <p>A+ - для активной энергии прямого направления</p>

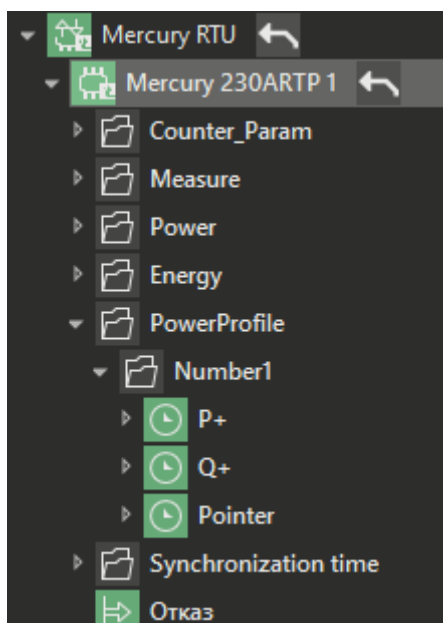
Название	Назначение
	R+- для реактивной энергии прямого направления Pointer - служебный канал По умолчанию все параметры каналов группы архивируются.
Группа Synchronization time	
Run	Позволяет выполнить синхронизацию времени. Изменяет время, заданное ранее в счетчике.
Error	Сообщает об ошибке синхронизации.

Каждый модуль ввода-вывода имеет выход Отказ

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если хоть у одного параметра канала, входящего в состав групп PowerProfile или Energy, снять флаг Архивировать, то выключается из опроса вся эта группа целиком.

6.2.3.2.10.4. MERCURY 230ARTP

Вид счетчика добавленного в протокол в дереве системы:



Состав каналов счетчика:

Название	Назначение
Группа Counter_Param	
SerialNumber	Указывается серийный номер счетчика
CurrentTime	Показывает текущее время, установленное в счетчике
Группа Measure	
Группа U A B C	Содержит каналы предназначенные для получения текущих значений напряжения для каждой фазы: Фаза А Фаза В Фаза С
Группа I A B C	Содержит каналы предназначенные для получения текущих значений тока для каждой фазы. Фаза А Фаза В Фаза С
F	Служит для получения значения частоты
Группа Power	
Группа P SUM A B C	Содержит каналы предназначенные для получения текущих значений активной мощности для каждой фазы. Суммарное значение по всем фазам Фаза А Фаза В Фаза С

Название	Назначение
<p>Группа Q</p> <p>SUM</p> <p>A</p> <p>B</p> <p>C</p>	<p>Содержит каналы предназначенные для получения текущих значений реактивной мощности для каждой фазы.</p> <p>Суммарное значение по всем фазам</p> <p>Фаза A</p> <p>Фаза B</p> <p>Фаза C</p>
<p>Группа S</p> <p>SUM</p> <p>A</p> <p>B</p> <p>C</p>	<p>Содержит каналы предназначенные для получения текущих значений полной мощности для каждой фазы.</p> <p>Суммарное значение по всем фазам</p> <p>Фаза A</p> <p>Фаза B</p> <p>Фаза C</p>
Группа Energy	
Группа FromReset	<p>Содержит накопленное значение энергии с момента сброса счетчика для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
Группа Year_now	<p>Содержит накопленное значение энергии за текущий год для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
Группа Year_before	<p>Содержит накопленное значение энергии за предыдущий год для каждого из четырех возможных тарифов, и для их суммы. Каждый</p>

Название	Назначение
	<p>тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
Группа Day_now	<p>Содержит накопленное значение энергии за текущие сутки для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
Группа Day_before	<p>Содержит накопленное значение энергии за предыдущие сутки для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
Группа Month	<p>Задержит архив накопленных месячных значений для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится три канала.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+- для реактивной энергии прямого направления</p> <p>Pointer - служебный канал</p> <p>По умолчанию все параметры каналов группы архивируются.</p>
Группа PowerProfile.Number1	
R+	<p>Передаются архивные значение активной мощности прямого направления. По умолчанию параметр канала архивируется.</p>

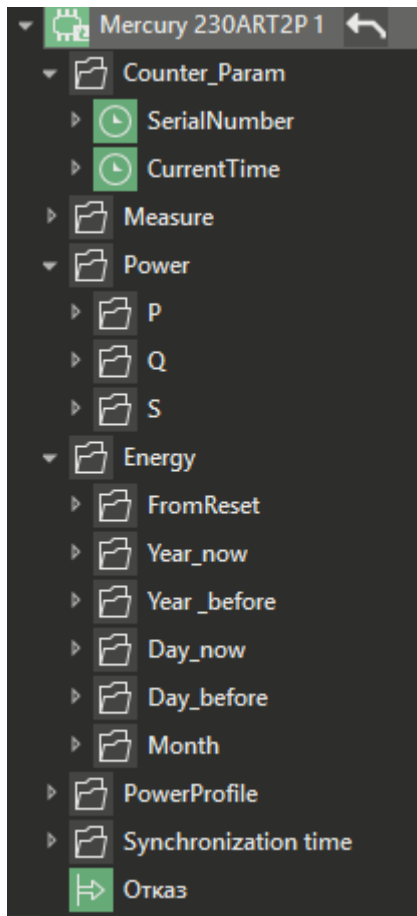
Название	Назначение
Q+	Передаются архивные значение реактивной мощности прямого направления. По умолчанию параметр канала архивируется.
Pointer	Служебный канал.
Группа Synchronization time	
Run	Позволяет выполнить синхронизацию времени. Изменяет время, заданное ранее в счетчике.
Error	Сообщает об ошибке синхронизации.

Каждый модуль ввода-вывода имеет выход Отказ

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если хоть у одного параметра канала, входящего в состав групп PowerProfile или Energy, снять флаг Архивировать, то выключается из опроса вся эта группа целиком.

6.2.3.2.10.5. MERCURY 230ART2P

Вид счетчика добавленного в протокол в дереве системы:



Состав каналов счетчика:

Название	Назначение
Группа Counter_Param	
SerialNumber	Указывается серийный номер счетчика
CurrentTime	Показывает текущее время, установленное в счетчике
Группа Measure	
Группа U	Содержит каналы предназначенные для получения текущих значений напряжения для каждой фазы:
A	Фаза A
B	Фаза B
C	Фаза C

Название	Назначение
Группа I	Содержит каналы предназначенные для получения текущих значений тока для каждой фазы.
A	Фаза А
B	Фаза В
C	Фаза С
F	Служит для получения значения частоты
Группа Power	
Группа P	Содержит каналы предназначенные для получения текущих значений активной мощности для каждой фазы.
SUM	Суммарное значение по всем фазам
A	Фаза А
B	Фаза В
C	Фаза С
Группа Q	Содержит каналы предназначенные для получения текущих значений реактивной мощности для каждой фазы.
SUM	Суммарное значение по всем фазам
A	Фаза А
B	Фаза В
C	Фаза С
Группа S	Содержит каналы предназначенные для получения текущих значений полной мощности для каждой фазы.
SUM	Суммарное значение по всем фазам
A	Фаза А
B	Фаза В
C	Фаза С

Название	Назначение
	Фаза С
Группа Energy	
Группа FromReset	<p>Содержит накопленное значение энергии с момента сброса счетчика для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится четыре канала.</p> <p style="text-align: center;">A+ - для активной энергии прямого направления R+ - для реактивной энергии прямого направления A- - для активной энергии обратного направления R- - для реактивной энергии обратного направления</p>
Группа Year_now	<p>Содержит накопленное значение энергии за текущий год для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится четыре канала.</p> <p style="text-align: center;">A+ - для активной энергии прямого направления R+ - для реактивной энергии прямого направления A- - для активной энергии обратного направления R- - для реактивной энергии обратного направления</p>
Группа Year_before	<p>Содержит накопленное значение энергии за предыдущий год для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится четыре канала.</p> <p style="text-align: center;">A+ - для активной энергии прямого направления R+ - для реактивной энергии прямого направления A- - для активной энергии обратного направления R- - для реактивной энергии обратного направления</p>

Название	Назначение
Группа Day_now	<p>Содержит накопленное значение энергии за текущие сутки для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится четыре канала.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p>
Группа Day_before	<p>Содержит накопленное значение энергии за предыдущие сутки для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится четыре канала.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p>
Группа Month	<p>Задержит архив накопленных месячных значений для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится пять каналов.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p> <p>Pointer - служебный канал</p> <p>По умолчанию все параметры каналов группы архивируются.</p>
Группа PowerProfile.Number1	

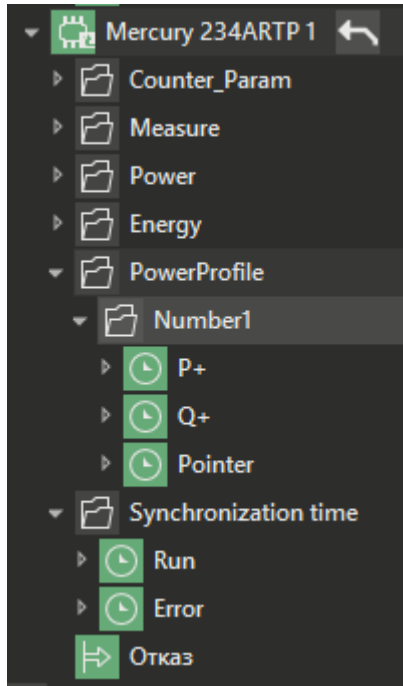
Название	Назначение
P+	Передаются архивные значение активной мощности прямого направления. По умолчанию параметр канала архивируется.
Q+	Передаются архивные значение реактивной мощности прямого направления. По умолчанию параметр канала архивируется.
P-	Передаются архивные значение активной мощности обратного направления. По умолчанию параметр канала архивируется.
Q-	Передаются архивные значение реактивной мощности обратного направления. По умолчанию параметр канала архивируется.
Pointer	Служебный канал.
Группа Synchronization time	
Run	Позволяет выполнить синхронизацию времени. Изменяет время, заданное ранее в счетчике.
Error	Сообщает об ошибке синхронизации.

Каждый модуль ввода-вывода имеет выход Отказ

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если хоть у одного параметра канала, входящего в состав групп PowerProfile или Energy, снять флаг Архивировать, то выключается из опроса вся эта группа целиком.

6.2.3.2.10.6. MERCURY 234ARTP

Вид счетчика добавленного в протокол в дереве системы:



Состав каналов счетчика:

Название	Назначение
Группа Counter_Param	
SerialNumber	Указывается серийный номер счетчика
CurrentTime	Показывает текущее время, установленное в счетчике
Группа Measure	
Группа U	Содержит каналы предназначенные для получения текущих значений напряжения для каждой фазы:
A	Фаза A
B	Фаза B
C	Фаза C
Группа I	Содержит каналы предназначенные для получения текущих значений тока для каждой фазы.
A	Фаза A
B	Фаза A

Название	Назначение
С	Фаза В Фаза С
F	Служит для получения значения частоты
Группа Power	
Группа P SUM A B C	Содержит каналы предназначенные для получения текущих значений активной мощности для каждой фазы. Суммарное значение по всем фазам Фаза А Фаза В Фаза С
Группа Q SUM A B C	Содержит каналы предназначенные для получения текущих значений реактивной мощности для каждой фазы. Суммарное значение по всем фазам Фаза А Фаза В Фаза С
Группа S SUM A B C	Содержит каналы предназначенные для получения текущих значений полной мощности для каждой фазы. Суммарное значение по всем фазам Фаза А Фаза В Фаза С
Группа Energy	

Название	Назначение
Группа FromReset	<p>Содержит накопленное значение энергии с момента сброса счетчика для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
Группа Year_now	<p>Содержит накопленное значение энергии за текущий год для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
Группа Year_before	<p>Содержит накопленное значение энергии за предыдущий год для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
Группа Day_now	<p>Содержит накопленное значение энергии за текущие сутки для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
Группа Day_before	<p>Содержит накопленное значение энергии за предыдущие сутки для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p>

Название	Назначение
	R+ для реактивной энергии прямого направления
Группа Month	<p>Содержит архив накопленных месячных значений для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится три канала.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+- для реактивной энергии прямого направления</p> <p>Pointer - служебный канал</p> <p>По умолчанию все параметры каналов группы архивируются.</p>
Группа Emonth_begin	<p>Содержит архив накопленных значений, сохраненных в начале каждого месяца для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится три канала.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+- для реактивной энергии прямого направления</p> <p>Pointer - служебный канал</p> <p>По умолчанию все параметры каналов группы архивируются.</p>
Группа PowerProfile.Number1	
P+	Передаются архивные значение активной мощности прямого направления. По умолчанию параметр канала архивируется.
Q+	Передаются архивные значение реактивной мощности прямого направления. По умолчанию параметр канала архивируется.
Pointer	Служебный канал.
Группа Synchronization time	

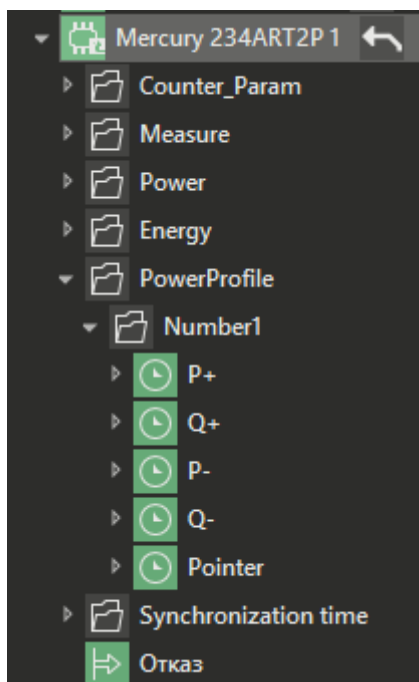
Название	Назначение
Run	Позволяет выполнить синхронизацию времени. Изменяет время, заданное ранее в счетчике.
Error	Сообщает об ошибке синхронизации.

Каждый модуль ввода-вывода имеет выход Отказ

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если хоть у одного параметра канала, входящего в состав групп PowerProfile или Energy, снять флаг Архивировать, то выключается из опроса вся эта группа целиком.

6.2.3.2.10.7. MERCURY 234ART2P

Вид счетчика добавленного в протокол в дереве системы:



Состав каналов счетчика:

Название	Назначение
Группа Counter_Param	
SerialNumber	Указывается серийный номер счетчика

Название	Назначение
CurrentTime	Показывает текущее время, установленное в счетчике
Группа Measure	
Группа U A B C	Содержит каналы предназначенные для получения текущих значений напряжения для каждой фазы: Фаза А Фаза В Фаза С
Группа I A B C	Содержит каналы предназначенные для получения текущих значений тока для каждой фазы. Фаза А Фаза В Фаза С
F	Служит для получения значения частоты
Группа Power	
Группа P SUM A B C	Содержит каналы предназначенные для получения текущих значений активной мощности для каждой фазы. Суммарное значение по всем фазам Фаза А Фаза В Фаза С
Группа Q SUM A	Содержит каналы предназначенные для получения текущих значений реактивной мощности для каждой фазы. Суммарное значение по всем фазам

Название	Назначение
В	Фаза А
С	Фаза В
	Фаза С
Группа S	Содержит каналы предназначенные для получения текущих значений полной мощности для каждой фазы.
SUM	Суммарное значение по всем фазам
А	Фаза А
В	Фаза В
С	Фаза С
Группа Energy	
Группа FromReset	<p>Содержит накопленное значение энергии с момента сброса счетчика для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится четыре канала.</p> <p>А+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p> <p>А- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p>
Группа Year_now	<p>Содержит накопленное значение энергии за текущий год для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится четыре канала.</p> <p>А+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p> <p>А- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p>

Название	Назначение
Группа Year _before	<p>Содержит накопленное значение энергии за предыдущий год для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится четыре канала.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p>
Группа Day_now	<p>Содержит накопленное значение энергии за текущие сутки для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится четыре канала.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p>
Группа Day_before	<p>Содержит накопленное значение энергии за предыдущие сутки для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится четыре канала.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p>
Группа Month	<p>Задержит архив накопленных месячных значений для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится пять каналов.</p>

Название	Назначение
	<p>A+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p> <p>Pointer - служебный канал</p> <p>По умолчанию все параметры каналов группы архивируются.</p>
Группа Emonth_begin	<p>Содержит архив накопленных значений, сохраненных в начале каждого месяца для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится пять каналов.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+- для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p> <p>Pointer - служебный канал</p> <p>По умолчанию все параметры каналов группы архивируются.</p>
Группа PowerProfile.Number1	
P+	Передаются архивные значение активной мощности прямого направления. По умолчанию параметр канала архивируется.
Q+	Передаются архивные значение реактивной мощности прямого направления. По умолчанию параметр канала архивируется.
P-	Передаются архивные значение активной мощности обратного направления. По умолчанию параметр канала архивируется.
Q-	Передаются архивные значение реактивной мощности обратного направления. По умолчанию параметр канала архивируется.

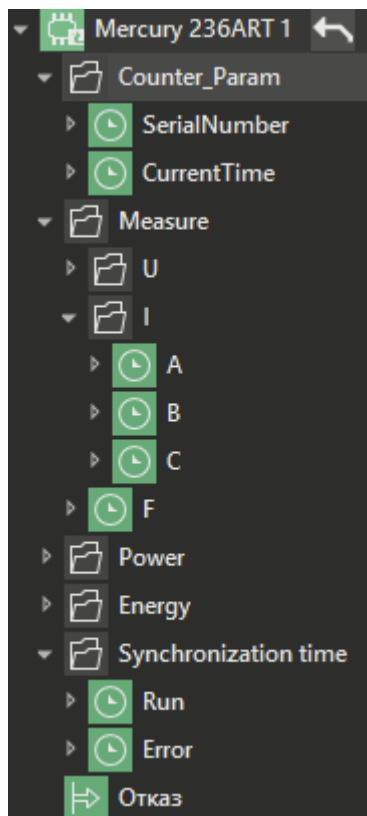
Название	Назначение
Pointer	Служебный канал.
Группа Synchronization time	
Run	Позволяет выполнить синхронизацию времени. Изменяет время, заданное ранее в счетчике.
Error	Сообщает об ошибке синхронизации.

Каждый модуль ввода-вывода имеет выход Отказ

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если хоть у одного параметра канала, входящего в состав групп PowerProfile или Energy, снять флаг Архивировать, то выключается из опроса вся эта группа целиком.

6.2.3.2.10.8. MERCURY 236ART

Вид счетчика добавленного в протокол в дереве системы:



Состав каналов счетчика:

Название	Назначение
Группа Counter_Param	
SerialNumber	Указывается серийный номер счетчика
CurrentTime	Показывает текущее время, установленное в счетчике
Группа Measure	
Группа U	Содержит каналы предназначенные для получения текущих значений напряжения для каждой фазы:
A	Фаза A
B	Фаза B
C	Фаза C
Группа I	Содержит каналы предназначенные для получения текущих значений тока для каждой фазы.
A	Фаза A
B	Фаза B
C	Фаза C
F	Служит для получения значения частоты
Группа Power	
Группа P	Содержит каналы предназначенные для получения текущих значений активной мощности для каждой фазы.
SUM	Суммарное значение по всем фазам
A	Фаза A
B	Фаза B
C	Фаза C

Название	Назначение
<p>Группа Q</p> <p>SUM</p> <p>A</p> <p>B</p> <p>C</p>	<p>Содержит каналы предназначенные для получения текущих значений реактивной мощности для каждой фазы.</p> <p>Суммарное значение по всем фазам</p> <p>Фаза A</p> <p>Фаза B</p> <p>Фаза C</p>
<p>Группа S</p> <p>SUM</p> <p>A</p> <p>B</p> <p>C</p>	<p>Содержит каналы предназначенные для получения текущих значений полной мощности для каждой фазы.</p> <p>Суммарное значение по всем фазам</p> <p>Фаза A</p> <p>Фаза B</p> <p>Фаза C</p>
Группа Energy	
<p>Группа FromReset</p>	<p>Содержит накопленное значение энергии с момента сброса счетчика для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
<p>Группа Year_now</p>	<p>Содержит накопленное значение энергии за текущий год для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>

Название	Назначение
Группа Year _before	<p>Содержит накопленное значение энергии за предыдущий год для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
Группа Day_now	<p>Содержит накопленное значение энергии за текущие сутки для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
Группа Day_before	<p>Содержит накопленное значение энергии за предыдущие сутки для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
Группа Month	<p>Задержит архив накопленных месячных значений для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится три канала.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+- для реактивной энергии прямого направления</p> <p>Pointer - служебный канал</p> <p>По умолчанию все параметры каналов группы архивируются.</p>
Группа Emonth_begin	<p>Содержит архив накопленных значений, сохраненных в начале каждого месяца для каждого из четырех возможных тарифов, и для</p>

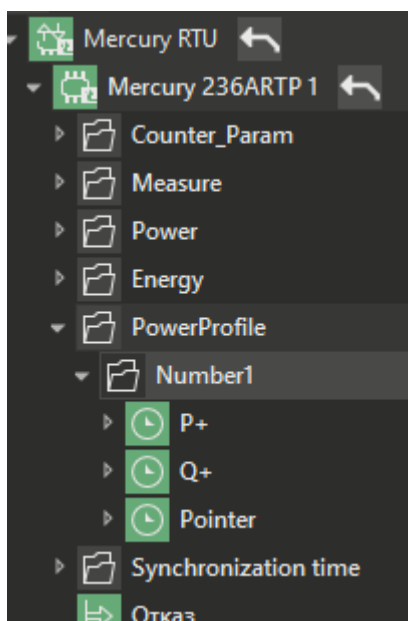
Название	Назначение
	<p>их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится три канала.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+- для реактивной энергии прямого направления</p> <p>Pointer - служебный канал</p> <p>По умолчанию все параметры каналов группы архивируются.</p>
Группа Synchronization time	
Run	Позволяет выполнить синхронизацию времени. Изменяет время, заданное ранее в счетчике.
Error	Сообщает об ошибке синхронизации.

Каждый модуль ввода-вывода имеет выход Отказ

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если хоть у одного параметра канала, входящего в состав групп PowerProfile или Energy, снять флаг Архивировать, то выключается из опроса вся эта группа целиком.

6.2.3.2.10.9. MERCURY 236ARTP

Вид счетчика добавленного в протокол в дереве системы:



Состав каналов счетчика:

Название	Назначение
Группа Counter_Param	
SerialNumber	Указывается серийный номер счетчика
CurrentTime	Показывает текущее время, установленное в счетчике
Группа Measure	
Группа U	Содержит каналы предназначенные для получения текущих значений напряжения для каждой фазы:
A	Фаза А
B	Фаза В
C	Фаза С
Группа I	Содержит каналы предназначенные для получения текущих значений тока для каждой фазы.
A	Фаза А
B	Фаза В
C	Фаза В

Название	Назначение
	Фаза С
F	Служит для получения значения частоты
Группа Power	
Группа P	Содержит каналы предназначенные для получения текущих значений активной мощности для каждой фазы.
SUM	Суммарное значение по всем фазам
A	Фаза А
B	Фаза В
C	Фаза С
Группа Q	Содержит каналы предназначенные для получения текущих значений реактивной мощности для каждой фазы.
SUM	Суммарное значение по всем фазам
A	Фаза А
B	Фаза В
C	Фаза С
Группа S	Содержит каналы предназначенные для получения текущих значений полной мощности для каждой фазы.
SUM	Суммарное значение по всем фазам
A	Фаза А
B	Фаза В
C	Фаза С
Группа Energy	
Группа FromReset	Содержит накопленное значение энергии с момента сброса счетчика для каждого из четырех возможных тарифов, и для их суммы.

Название	Назначение
	<p>Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
Группа Year_now	<p>Содержит накопленное значение энергии за текущий год для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
Группа Year_before	<p>Содержит накопленное значение энергии за предыдущий год для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
Группа Day_now	<p>Содержит накопленное значение энергии за текущие сутки для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>
Группа Day_before	<p>Содержит накопленное значение энергии за предыдущие сутки для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится два канала.</p> <p>A+ для активной энергии прямого направления</p> <p>R+ для реактивной энергии прямого направления</p>

Название	Назначение
Группа Month	<p>Содержит архив накопленных месячных значений для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится три канала.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+- для реактивной энергии прямого направления</p> <p>Pointer - служебный канал</p> <p>По умолчанию все параметры каналов группы архивируются.</p>
Группа Emonth_begin	<p>Содержит архив накопленных значений, сохраненных в начале каждого месяца для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится три канала.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+- для реактивной энергии прямого направления</p> <p>Pointer - служебный канал</p> <p>По умолчанию все параметры каналов группы архивируются.</p>
Группа PowerProfile.Number1	
P+	Передаются архивные значение активной мощности прямого направления. По умолчанию параметр канала архивируется.
Q+	Передаются архивные значение реактивной мощности прямого направления. По умолчанию параметр канала архивируется.
Pointer	Служебный канал.
Группа Synchronization time	
Run	Позволяет выполнить синхронизацию времени. Изменяет время, заданное ранее в счетчике.
Error	Сообщает об ошибке синхронизации.

Каждый модуль ввода-вывода имеет выход Отказ

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если хоть у одного параметра канала, входящего в состав групп PowerProfile или Energy, снять флаг Архивировать, то выключается из опроса вся эта группа целиком.

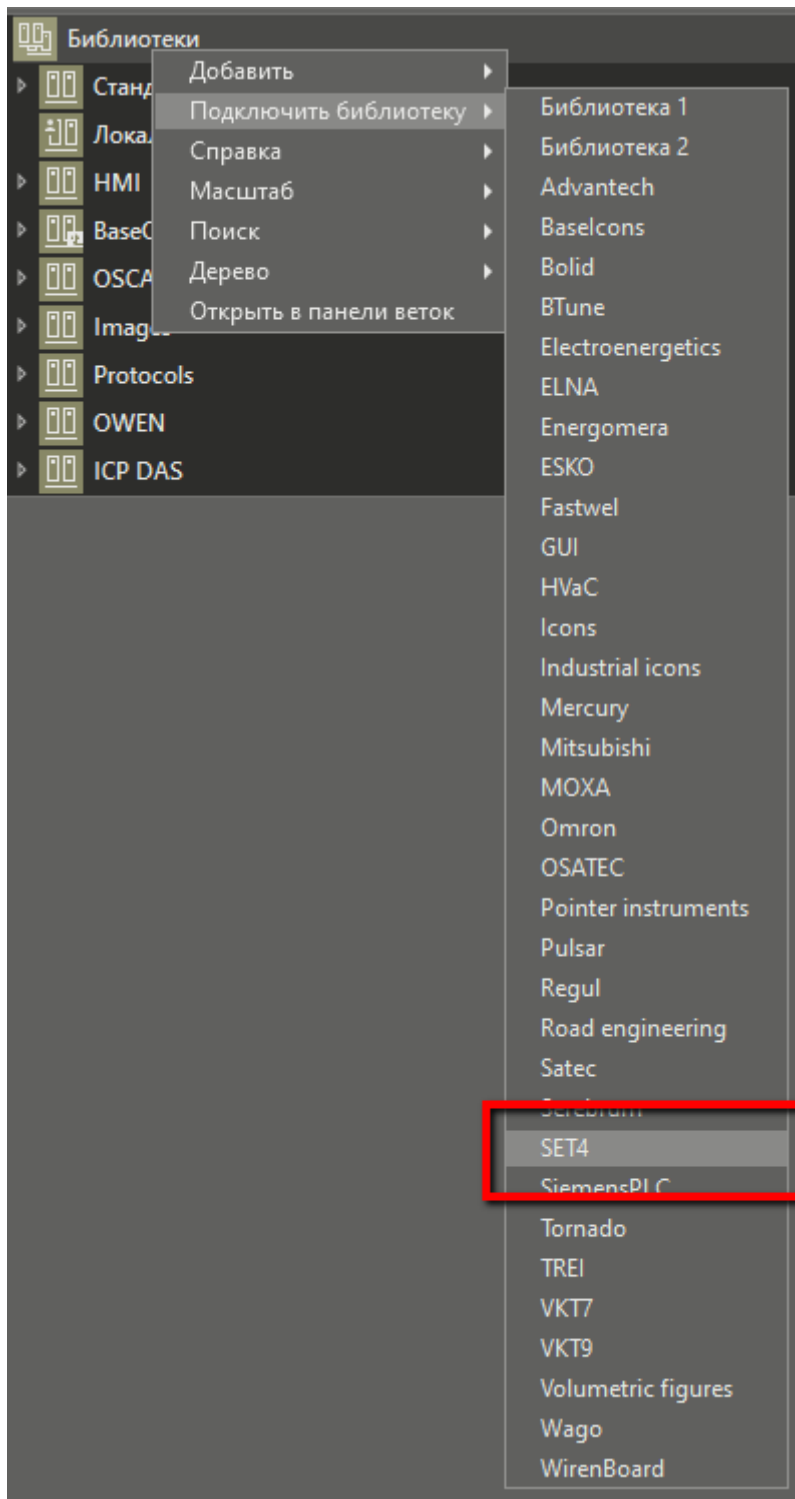
6.2.3.2.11.SET4

MasterSCADA 4D поддерживает работу со счетчиками электроэнергии СЭТ-4ТМ, ПСЧ-4ТМ фирмы Нижегородское научно-производственное объединение им. Фрунзе.

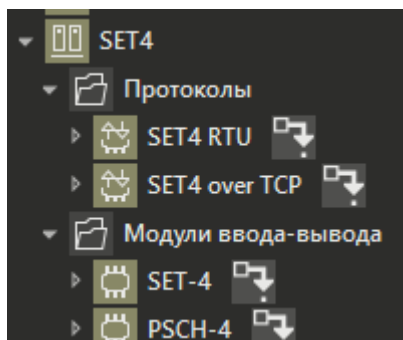
При обмене узел играет роль ведущего. Поддерживается считывание архивов из счетчика. Работа с этими устройствами возможна через последовательные интерфейсы (RS-232, RS-485, GSM), а при использовании конвертеров Ethernet-COM (Моха NPort 5150 и т.п.) также и по сетям TCP/IP (Ethernet, Wi-Fi).

Важно! По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте sales@masterscada.ru

Для получения возможности использования этих счетчиков в проекте необходимо подключить соответствующую библиотеку:

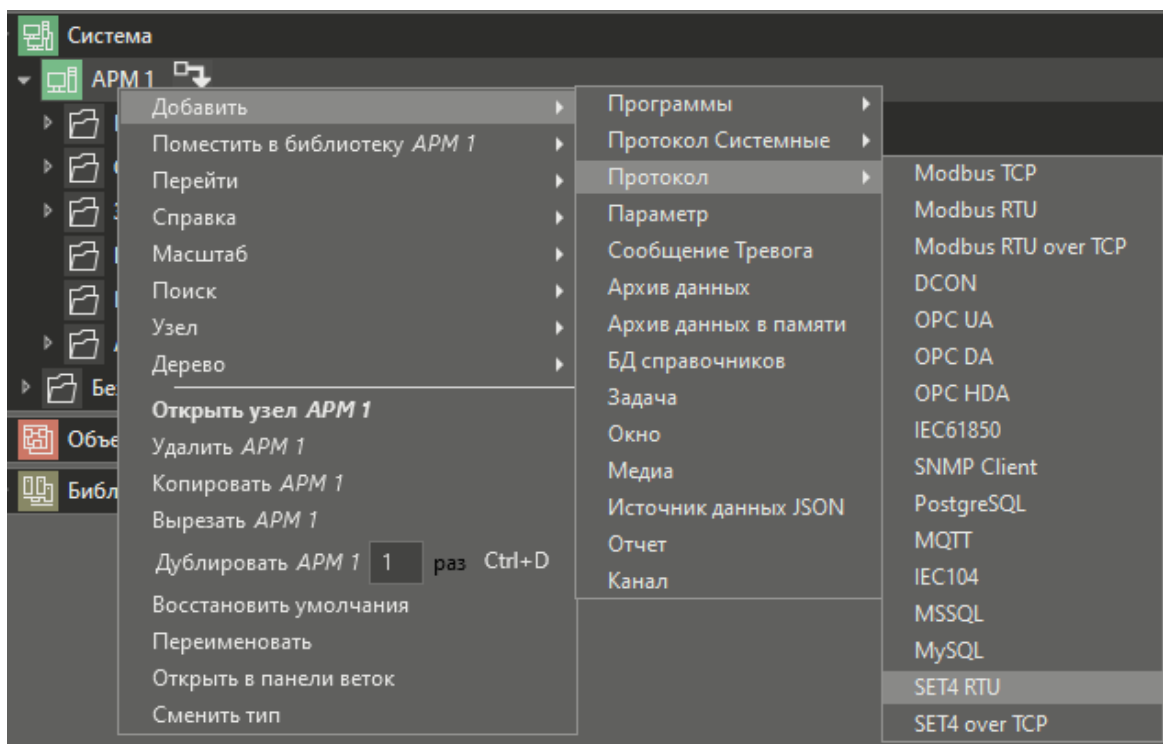


В полном дереве библиотека SET4 имеет вид:



После этого в проект могут быть добавлены сначала протокол, а затем устройство, любым известным способом, например, при помощи контекстного меню узла, или с использованием контекстной панели, или путем перетаскивания элемента из библиотеки в дерево системы.

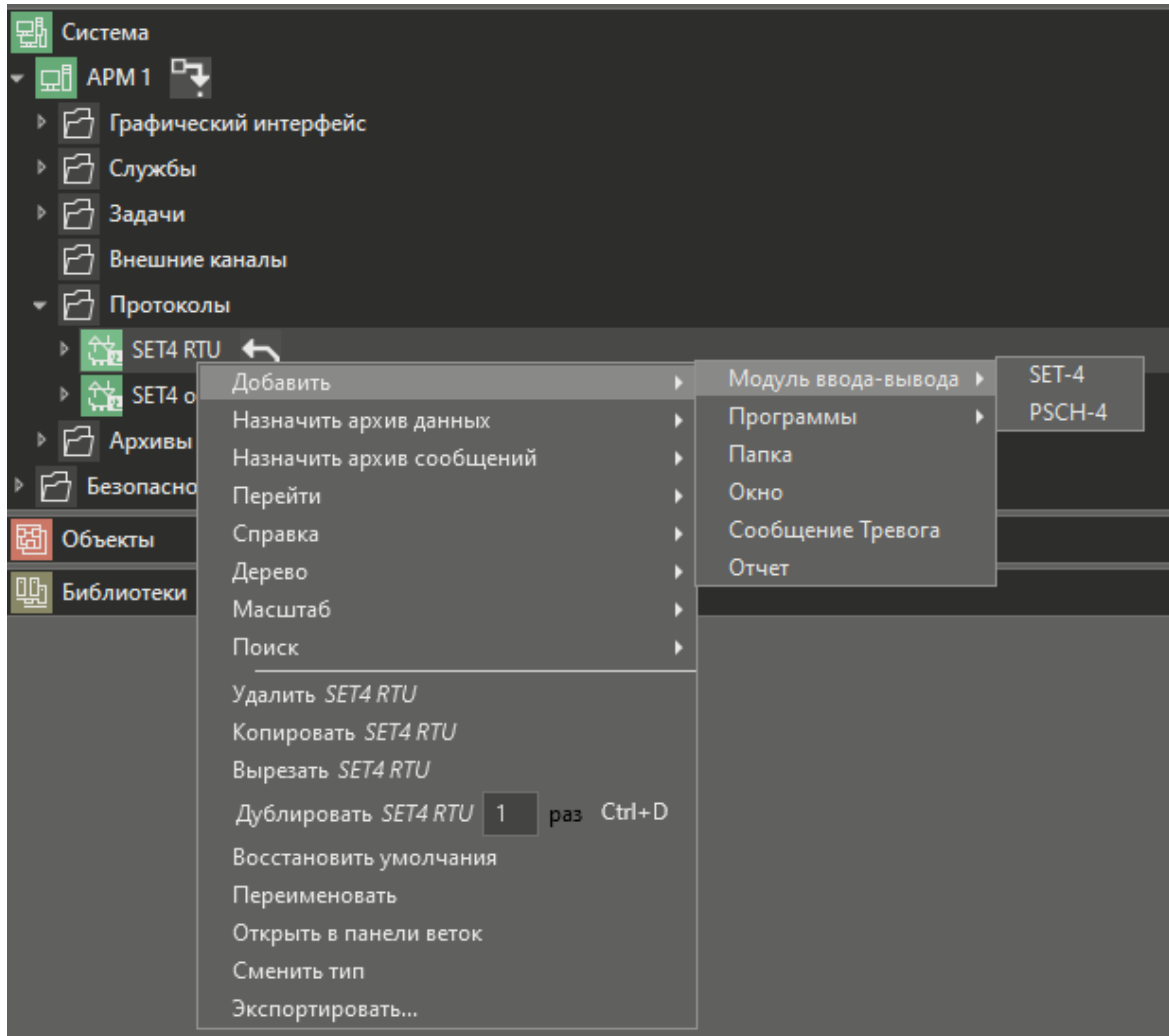
Рекомендуется добавлять элементы в проект при помощи контекстного меню:



Тип добавляемого протокола зависит от реального способа подключения счетчика:

- SET4 RTU - если счетчик подключен по последовательному интерфейсу. Настройки этого протокола практически полностью соответствуют настройкам Modbus RTU.
- SET4 over TCP - если счетчик подключен через конвертер Ethernet-COM. Настройки этого протокола соответствуют настройкам Modbus RTU over TCP

Затем при помощи контекстного меню протокола можно добавить модули ввода-вывода, которые уже содержат все необходимые каналы:



Каждый счетчик имеет те группы каналов, которые характерны для его работы.

Группа	Назначение
Counter_Param	Параметры счетчика
Measure	Текущие значения напряжения, тока и частоты
Power	Значения мощности
Energy	Накопленные энергии
PowerProfile	Профили мощности. Каналы предназначены для передачи архивных значений счетчика – профиля мощности. Профиль мощности – архив срезов

Группа	Назначение
	мощности за заданный пользователем интервал времени (как правило, 30 или 60 минут). Счетчик может содержать один или два профиля мощности, поэтому группа содержит две группы – Number1 и Number2.
Synchronization time	Параметры синхронизации времени.

Поддержаны наиболее используемые функции счетчиков.

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если хоть у одного параметра канала, входящего в состав группы PowerProfile или Energy, снять флаг Архивировать, то выключается из опроса вся эта группа целиком.

Поддерживаются следующие модули ввода-вывода:

- SET-4
- PSCH-4

После добавления в дерево проекта модулей ввода-вывода, настраивается панель свойств модулей.

Каждый модуль ввода-вывода имеет выход Отказ, а также группы каналов, согласно его максимальной конфигурации.

Если в опросе каких-либо каналов нет необходимости, то можно либо удалить их из конфигурации элемента уже после добавления его в дерево системы, либо сначала создать экземпляр модуля в пользовательской библиотеке, затем из его конфигурации удалить незадействованные каналы, после чего использовать в проекте уже новый видоизмененный библиотечный элемент.

6.2.3.2.11.1. СВОЙСТВА МОДУЛЕЙ SET4

Вид панели свойств:

PSCH-4 1

Общие

Имя	PSCH-4 1
Полное имя	Система.APM 1.Протоколы.SET4 RTU.PSCH-4 1
Метки	
Комментарий	Счетчик ПСЧ-4

Служебные

Задача

Подключение по условию	<input type="checkbox"/>
Опрос по условию	<input type="checkbox"/>
Способ записи	По умолчанию

Отношения

Масштабирование

Настройки

Адрес	2
Пароль	000000
Тип контрольной суммы (CRC-true, KC-false)	<input checked="" type="checkbox"/>
Количество считываемых записей профиля мощности за цикл	5
Автоматически определять Кс и Сi	<input checked="" type="checkbox"/>
Кс (ручное)	1
Сi (ручное)	1

Описание:

Название	Описание
Категория Общие	Как правило, для данного элемента эта категория не настраивается. Подробное описание данной категории приведено в соответствующем разделе.
Категория Задача	Задается способ опроса устройства. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи.
Категория Масштабирование	Если модуль ввода-вывода не имеет встроенного масштабирования, то при помощи данной категории можно задать соответствие между значениями в модуле и реальными значениями. Подробное описание данной категории приведено в соответствующем разделе.

Название	Описание
Адрес	Задается адрес устройства в сети (значение адреса определяется документацией на подключаемое устройство). Заводским значением адреса, у счетчиков Меркурий являются последние 3 цифры серийного номера прибора, а если число превышает 240, то последним двум цифрам
Пароль	Задается пароль доступа к данным счетчика
Тип контрольной суммы (CRC-true, KC-false)	Определяет вид контрольной суммы. Контрольная сумма может быть двух видов: KC и CRC. Установите нужный вариант контрольной суммы.
Количество считываемых записей профиля мощности за цикл	Определяет количество считываемых каждый цикл опроса записей профиля мощности (для архивированных каналов группы PowerProfile).
Автоматически определять Kc и Ci	Определяет способ определения параметров Kc и Ci. Параметры Kc и Ci определяют тип модификации счетчика (см. табл. ниже). Если флаг установлен, то параметры считываются из счетчика, в противном случае параметры берутся из заданных вручную в настройках.
Kc (ручное)	Определяет значение коэффициента Kc, используется в формуле для мгновенных мощностей. Зависит от типа счетчика (см. табл. ниже). При автоматическом определении настройка не используется
Ci (ручное)	Определяет значение коэффициента Ci, используется в формуле для тока. Зависит от типа счетчика (см. табл. ниже). При автоматическом определении настройка не используется.

Таблица коэффициентов:

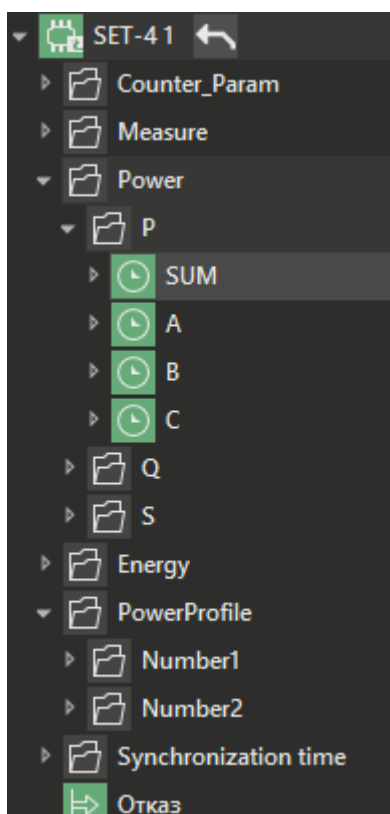
Тип счетчика	Uном, В	Iном (Imax), А	Kс	Ci
СЭТ-4ТМ.02, ПСЧ-4ТМ.05	57,7	5 (7,5)	1	1
	57,7	1 (1,5)	1	1
	120-230	5 (7,5)	2	1
	120-230	1 (1,5)	1	1
СЭТ-1М.01	230	5 (7,5)	1	1
СЭТ-4ТМ.03	57,7	1 (10)	1	1
	120-230	1 (10)	2	1
СЭТ-1М.01М	230	5 (10)	1	1
СЭТ-4ТМ.03	57,7	1(10)	1	1
	120-230	1(10)	2	1
СЭТ-4ТМ.02.03М	57,7-115	5 (10)	1	1
	57,7-115	1(2)	1	1
	120-230	5 (10)	2	1
	120-230	1(2)	1	1
ПСЧ-4ТМ.05М ПСЧ-4ТМ.05Д (без Iном=1 А)	57,7-115	5 (7,5)	1	1
	57,7-115	1 (1,5)	1	1
	120-230	5 (7,5)	2	1
	120-230	1 (1,5)	1	1
ПСЧ-3ТМ.05	230	5 (100)	20	1
ПСЧ-3ТМ.05М ПСЧ-4ТМ.5МК.20-25	120-230	5 (100)	20	1

Тип счетчика	Uном, В	Iном (Imax), А	Kс	Ci
ПСЧ-3ТМ.05Д	120-230	5 (75)	20	1
ПСЧ-4ТМ.5МК.00-19	57,7-115	5 (10)	1	1
	57,7-115	1(2)	1	1
	120-230	5 (10)	2	1
	120-230	1(2)	1	1
ПСЧ-4ТМ.5МК.20-25	120-230	5 (100)	20	1

6.2.3.2.11.2. SET-4

Счетчики СЭТ-4ТМ предназначены для измерения и многотарифного учета активной и реактивной электроэнергии (в том числе и с учетом потерь), ведения массивов профиля мощности нагрузки с программируемым временем интегрирования (в том числе и с учетом потерь), фиксации максимумов мощности, измерения параметров трехфазной сети и параметров качества электроэнергии.

Вид счетчика добавленного в протокол в дереве системы:



Состав каналов счетчика:

Название	Назначение
Группа Counter_Param	
SerialNumber	Указывается серийный номер счетчика
CurrentTime	Показывает текущее время, установленное в счетчике
Группа Measure	
Группа U	Содержит каналы предназначенные для получения текущих значений напряжения для каждой фазы, а также межфазные напряжения:
A	Фаза А
B	Фаза В
C	Фаза С
AB	Межфазное АВ
BC	Межфазное ВС
CA	Межфазное СА
Группа I	Содержит каналы предназначенные для получения текущих значений тока для каждой фазы.
A	Фаза А
B	Фаза В
C	Фаза С
F	Служит для получения значения частоты
Группа Power	
Группа P	Содержит каналы предназначенные для получения текущих значений активной мощности для каждой фазы.
SUM	

Название	Назначение
A	Суммарное значение по всем фазам
B	Фаза А
C	Фаза В Фаза С
Группа Q	Содержит каналы предназначенные для получения текущих значений реактивной мощности для каждой фазы.
SUM	Суммарное значение по всем фазам
A	Фаза А
B	Фаза В
C	Фаза С
Группа S	Содержит каналы предназначенные для получения текущих значений полной мощности для каждой фазы.
SUM	Суммарное значение по всем фазам
A	Фаза А
B	Фаза В
C	Фаза С
Группа Energy	
Группа FromReset	<p>Содержит накопленное значение энергии с момента сброса счетчика для каждого из восьми возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится четыре канала.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p>

Название	Назначение
Группа Year_now	<p>Содержит накопленное значение энергии за текущий год для каждого из восьми возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится четыре канала.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p>
Группа Year _before	<p>Содержит накопленное значение энергии за предыдущий год для каждого из восьми возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится четыре канала.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p>
Группа Day_now	<p>Содержит накопленное значение энергии за текущие сутки для каждого из восьми возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится четыре канала.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p>
Группа Day_before	<p>Содержит накопленное значение энергии за предыдущие сутки для каждого из восьми возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится четыре канала.</p>

Название	Назначение
	<p>A+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p>
Группа Month	<p>Задержит архив накопленных месячных значений для каждого из восьми возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится пять каналов.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p> <p>Pointer - служебный канал</p> <p>По умолчанию все параметры каналов группы архивируются.</p>
Группа Emonth_begin	<p>Содержит архив накопленных значений, сохраненных в начале каждого месяца для каждого из восьми возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится пять каналов.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+- для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p> <p>Pointer - служебный канал</p> <p>По умолчанию все параметры каналов группы архивируются.</p>
Группа Day	<p>Задержит архив накопленных суточных значений для каждого из восьми возможных тарифов, и для их суммы. Каждый тариф</p>

Название	Назначение
	<p>представлен в виде группы. В каждой группе содержится пять каналов.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p> <p>Pointer - служебный канал</p> <p>По умолчанию все параметры каналов группы архивируются.</p>
Группа Day_begin	<p>Содержит архив накопленных значений, сохраненных в начале каждого дня для каждого из восьми возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится пять каналов.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+- для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p> <p>Pointer - служебный канал</p> <p>По умолчанию все параметры каналов группы архивируются.</p>
Группы PowerProfile.Number1 и PowerProfile.Number2	
P+	<p>Передаются архивные значение активной мощности прямого направления. По умолчанию параметр канала архивируется.</p>
Q+	<p>Передаются архивные значение реактивной мощности прямого направления. По умолчанию параметр канала архивируется.</p>
P-	<p>Передаются архивные значение активной мощности обратного направления. По умолчанию параметр канала архивируется.</p>

Название	Назначение
Q-	Передаются архивные значение реактивной мощности обратного направления. По умолчанию параметр канала архивируется.
Pointer	Служебный канал.
Группа Synchronization time	
Run	Позволяет выполнить синхронизацию времени. Изменяет время, заданное ранее в счетчике.
Error	Сообщает об ошибке синхронизации.

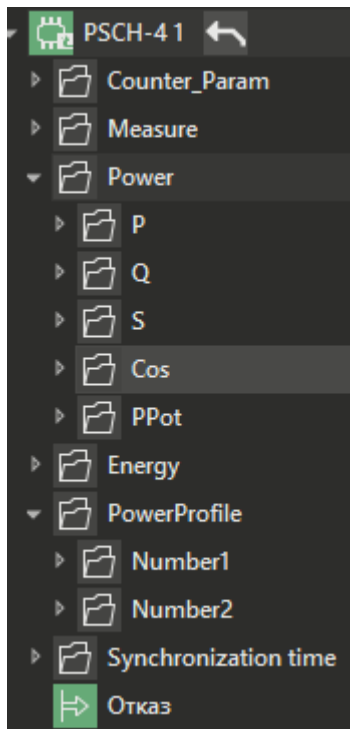
Каждый модуль ввода-вывода имеет выход Отказ

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если хоть у одного параметра канала, входящего в состав групп PowerProfile или Energy, снять флаг Архивировать, то выключается из опроса вся эта группа целиком.

6.2.3.2.11.3. PSCH-4

Счетчики ПСЧ-4ТМ предназначены для измерения и многотарифного учета активной и реактивной электроэнергии (в том числе и с учетом потерь), ведения массивов профиля мощности нагрузки с программируемым временем интегрирования (в том числе и с учетом потерь), фиксации максимумов мощности, измерения параметров трехфазной сети и параметров качества электроэнергии

Вид счетчика добавленного в протокол в дереве системы:



Состав каналов счетчика:

Название	Назначение
Группа Counter_Param	
SerialNumber	Указывается серийный номер счетчика
CurrentTime	Показывает текущее время, установленное в счетчике
Группа Measure	
Группа U	Содержит каналы предназначенные для получения текущих значений напряжения для каждой фазы, а также межфазные напряжения:
A	
B	Фаза A
C	Фаза B
AB	Фаза C
BC	Межфазное AB
CA	Межфазное BC

Название	Назначение
	Межфазное CA
Группа I	Содержит каналы предназначенные для получения текущих значений тока для каждой фазы.
A	Фаза A
B	Фаза B
C	Фаза C
F	Служит для получения значения частоты
Группа Power	
Группа P	Содержит каналы предназначенные для получения текущих значений активной мощности для каждой фазы (кВт).
SUM	Суммарное значение по всем фазам
A	Фаза A
B	Фаза B
C	Фаза C
Группа Q	Содержит каналы предназначенные для получения текущих значений реактивной мощности для каждой фазы (кВАр).
SUM	Суммарное значение по всем фазам
A	Фаза A
B	Фаза B
C	Фаза C
Группа S	Содержит каналы предназначенные для получения текущих значений полной мощности для каждой фазы (кВА).
SUM	Суммарное значение по всем фазам
A	Фаза A
B	Фаза A

Название	Назначение
С	Фаза В Фаза С
Группа Cos SUM А В С	Содержит каналы для получения текущих значений коэффициентов активной мощности. Суммарное значение по всем фазам Фаза А Фаза В Фаза С
Группа PPot SUM А В С	Содержит каналы для получения текущих значений мощности активных потерь в линии передачи (кВт). Суммарное значение по всем фазам Фаза А Фаза В Фаза С
Группа Energy	
Группа FromReset	Содержит накопленное значение энергии с момента сброса счетчика для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится четыре канала. А+ - для активной энергии прямого направления R+ - для реактивной энергии прямого направления А- - для активной энергии обратного направления R- - для реактивной энергии обратного направления
Группа Year_now	Содержит накопленное значение энергии за текущий год для каждого из четырех возможных тарифов, и для их суммы. Каждый

Название	Назначение
	<p>тариф представлен в виде группы. В каждой группе содержится четыре канала.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p>
Группа Year _before	<p>Содержит накопленное значение энергии за предыдущий год для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится четыре канала.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p>
Группа Day_now	<p>Содержит накопленное значение энергии за текущие сутки для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится четыре канала.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p>
Группа Day_before	<p>Содержит накопленное значение энергии за предыдущие сутки для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится четыре канала.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p>

Название	Назначение
	<p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p>
Группа Month	<p>Задержит архив накопленных месячных значений для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится пять каналов.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+ - для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p> <p>Pointer - служебный канал</p> <p>По умолчанию все параметры каналов группы архивируются.</p>
Группа Emonth_begin	<p>Содержит архив накопленных значений, сохраненных в начале каждого месяца для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится пять каналов.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+- для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p> <p>Pointer - служебный канал</p> <p>По умолчанию все параметры каналов группы архивируются.</p>
Группа Day	<p>Задержит архив накопленных суточных значений для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится пять каналов.</p> <p>A+ - для активной энергии прямого направления</p>

Название	Назначение
	<p>R+ - для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p> <p>Pointer - служебный канал</p> <p>По умолчанию все параметры каналов группы архивируются.</p>
Группа Day_begin	<p>Содержит архив накопленных значений, сохраненных в начале каждого дня для каждого из четырех возможных тарифов, и для их суммы. Каждый тариф представлен в виде группы. В каждой группе содержится пять каналов.</p> <p>A+ - для активной энергии прямого направления</p> <p>R+- для реактивной энергии прямого направления</p> <p>A- - для активной энергии обратного направления</p> <p>R- - для реактивной энергии обратного направления</p> <p>Pointer - служебный канал</p> <p>По умолчанию все параметры каналов группы архивируются.</p>
Группы PowerProfile.Number1 и PowerProfile.Number2	
P+	Передаются архивные значение активной мощности прямого направления. По умолчанию параметр канала архивируется.
Q+	Передаются архивные значение реактивной мощности прямого направления. По умолчанию параметр канала архивируется.
P-	Передаются архивные значение активной мощности обратного направления. По умолчанию параметр канала архивируется.
Q-	Передаются архивные значение реактивной мощности обратного направления. По умолчанию параметр канала архивируется.
Pointer	Служебный канал.

Название	Назначение
Группа Synchronization time	
Run	Позволяет выполнить синхронизацию времени. Изменяет время, заданное ранее в счетчике.
Error	Сообщает об ошибке синхронизации.

Каждый модуль ввода-вывода имеет выход Отказ

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если хоть у одного параметра канала, входящего в состав групп PowerProfile или Energy, снять флаг Архивировать, то выключается из опроса вся эта группа целиком.

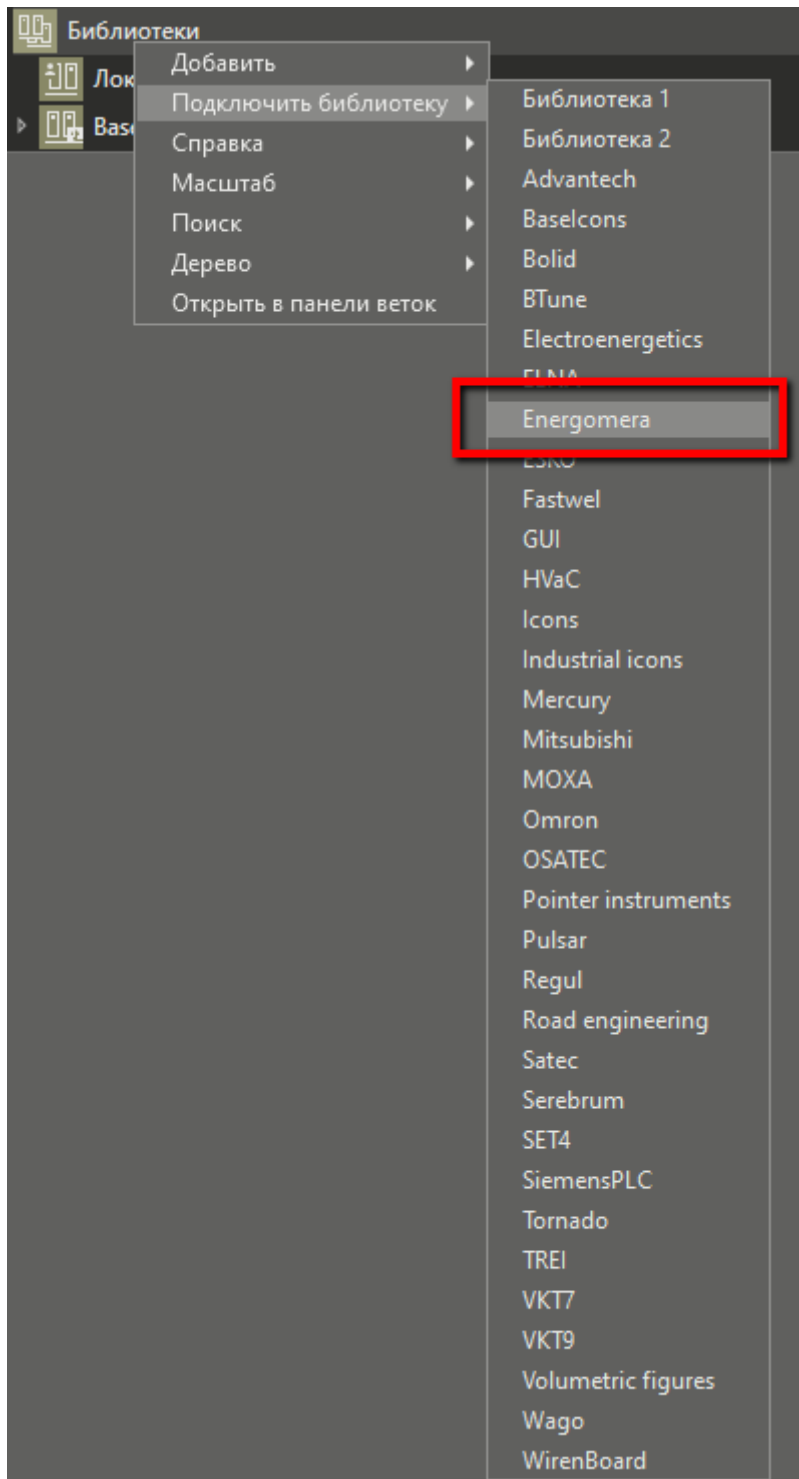
6.2.3.2.12.ENERGOMERA

MasterSCADA 4D поддерживает работу со счетчиками CE301, CE303 фирмы Энергомера по протоколу МЭК61107.

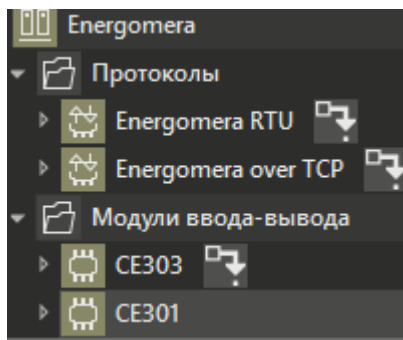
При обмене узел играет роль ведущего. Поддерживается считывание архивов из счетчика. Работа с этими устройствами возможна через последовательные интерфейсы (RS-232, RS-485, GSM), а при использовании конвертеров Ethernet-COM (Моха NPort 5150 и т.п.) также и по сетям TCP/IP (Ethernet, Wi-Fi).

Важно! По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте sales@masterscada.ru

Для получения возможности использования этих счетчиков в проекте необходимо подключить соответствующую библиотеку:

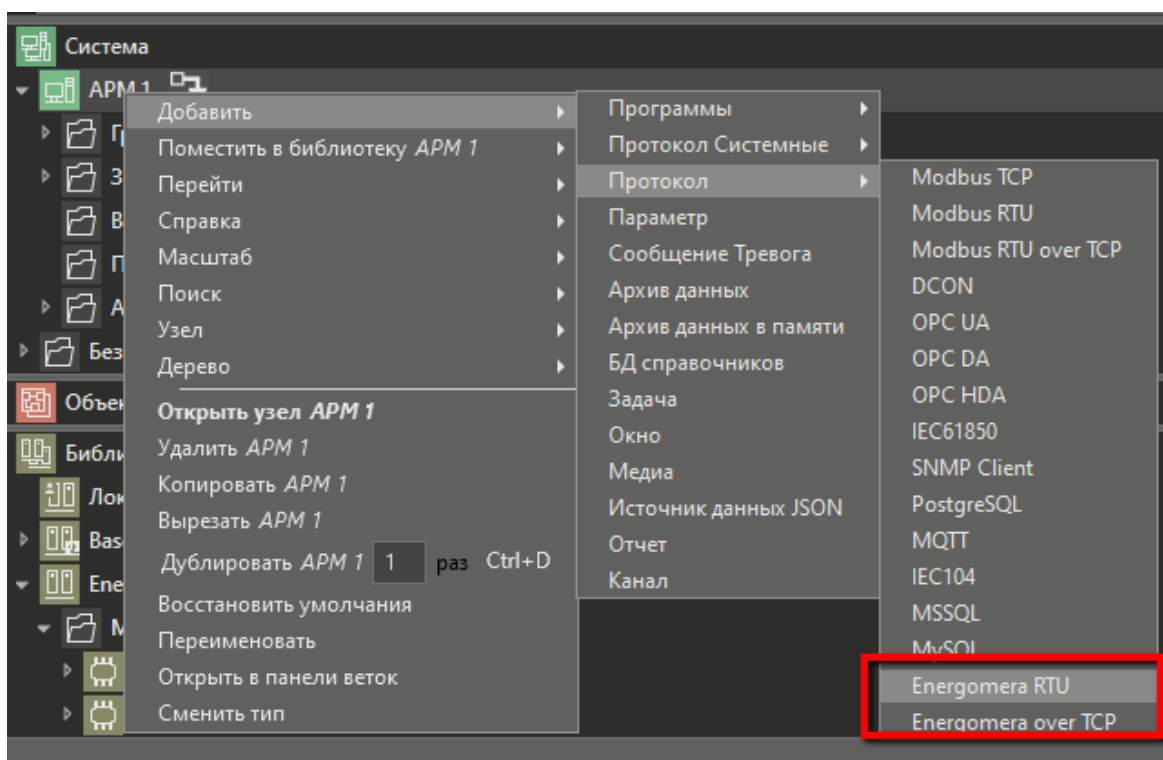


В полном дереве библиотека Energomera имеет вид:



После этого в проект могут быть добавлены сначала протокол, а затем устройство, любым известным способом, например, при помощи контекстного меню узла, или с использованием контекстной панели, или путем перетаскивания элемента из библиотеки в дерево системы.

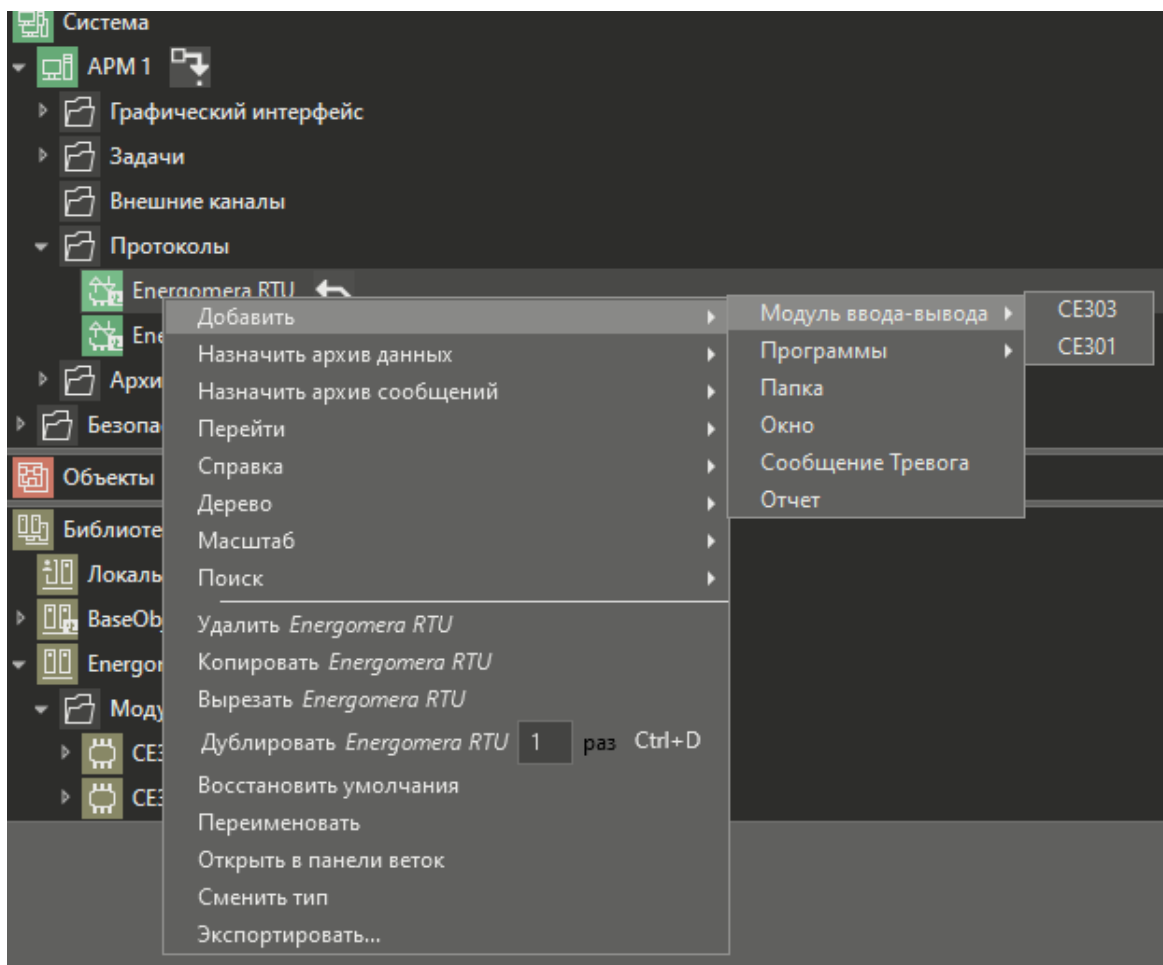
Рекомендуется добавлять элементы в проект при помощи контекстного меню:



Тип добавляемого протокола зависит от реального способа подключения счетчика:

- Energomera RTU - если счетчик подключен по последовательному интерфейсу. Настройки этого протокола практически полностью соответствуют настройкам Modbus RTU.
- Energomera over TCP - если счетчик подключен через конвертер Ethernet-COM. Настройки этого протокола соответствуют настройкам Modbus RTU over TCP

Затем при помощи контекстного меню протокола можно добавить модули ввода-вывода, которые уже содержат все необходимые каналы:



Каждый счетчик имеет те группы каналов, которые характерны для его работы.

Группа	Назначение
Counter_Param	Параметры счетчика
Measure	Текущие значения напряжения, тока, мощности и частоты
PowerProfile	Профили мощности. Каналы предназначены для передачи архивных значений счетчика – профиля мощности.
Energy	Содержит текущие значения энергий
EnergyArchive	Содержит архивные значения энергий
Maximum	Содержит архив месячных максимумом мощности

ActualPower	Содержит архив фактической мощности ежемесячно
-------------	--

Поддержаны наиболее используемые функции счетчиков.

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если у архивируемого параметра группы PowerProfile снять флаг Архивировать, то этот канал выключится из опроса. Если у архивируемого параметра группы EnergyArchive снять флаг Архивировать, то опрос канала продолжится, но значение в канал не запишется.

Поддерживаются следующие модули ввода-вывода:

- SE301
- SE303

После добавления в дерево проекта модулей ввода-вывода, настраивается панель свойств модулей.

Каждый модуль ввода-вывода имеет выход Отказ, а также группы каналов, согласно его максимальной конфигурации.

Если в опросе каких-либо каналов нет необходимости, то можно либо удалить их из конфигурации элемента уже после добавления его в дерево системы, либо сначала создать экземпляр модуля в пользовательской библиотеке, затем из его конфигурации удалить незадействованные каналы, после чего использовать в проекте уже новый видоизмененный библиотечный элемент.

6.2.3.2.12.1. СВОЙСТВА МОДУЛЕЙ ENERGMOMERA

Вид панели свойств:

CE303 1

🔍
🔗
☰
🏠
🔑

Общие ▲

Имя	CE303 1
Полное имя	Система.APM 1.Протоколы.Energomera RTU.CE303 1
Метки	
Комментарий	

Служебные ▼

Задача ▼

Отношения ▼

Масштабирование ▼

Настройки ▲

Адрес	5
Пароль	
Количество считываемых записей архивов за цикл	5

Описание:

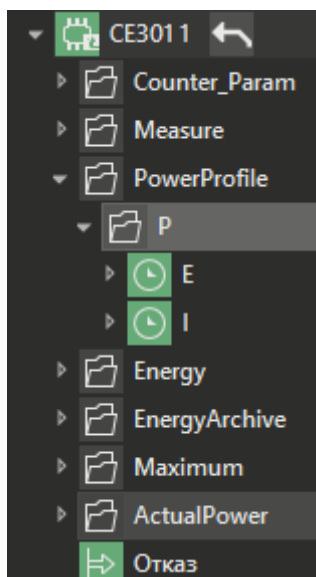
Название	Описание
Категория Общие	Как правило, для данного элемента эта категория не настраивается. Подробное описание данной категории приведено в соответствующем разделе.
Категория Задача	Задается способ опроса устройства. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи.
Категория Масштабирование	Если модуль ввода-вывода не имеет встроенного масштабирования, то при помощи данной категории можно задать соответствие между значениями в модуле и реальными значениями. Подробное описание данной категории приведено в соответствующем разделе.
Адрес	Задается адрес устройства в сети (значение адреса определяется документацией на подключаемое устройство). Заводским значением адреса, у

	счетчиков Меркурий являются последние 3 цифры серийного номера прибора, а если число превышает 240, то последним двум цифрам
Пароль	Задается пароль доступа к данным счетчика
Количество считываемых записей архивов за цикл	Определяет количество считываемых каждый цикл опроса записей из архива счетчика

6.2.3.2.12.2. CE301

Счетчики CE301 предназначены для измерения и многотарифного учета активной электроэнергии в прямом и обратном направлении, ведения массивов профиля мощности нагрузки с программируемым временем интегрирования, фиксации максимумов мощности, измерения параметров трехфазной сети и параметров качества электроэнергии

Вид счетчика добавленного в протокол в дереве системы:



Состав каналов счетчика:

Название	Назначение
Группа Counter_Param	
CurrentTime	Показывает текущее время, установленное в счетчике
Группа Measure	

Название	Назначение
Группа U A B C	Содержит каналы предназначенные для получения текущих значений напряжения для каждой фазы. Фаза А Фаза В Фаза С
Группа I A B C	Содержит каналы предназначенные для получения текущих значений тока для каждой фазы. Фаза А Фаза В Фаза С
Группа P A B C	Содержит каналы предназначенные для получения текущих значений активной мощности для каждой фазы. Фаза А Фаза В Фаза С
Группа AngleU AB BC CA	Содержит каналы предназначенные для получения текущих значений углов между фазами. Угол между фазами А и В Угол между фазами В и С Угол между фазами С и А
Группа F F	Служит для получения значения частоты
Группа Bat Bat	Содержит информацию об уровне заряда батареи

Название	Назначение
Группа Phases	Содержит каналы, в которых отображается наличие или отсутствия напряжения в фазах
UA	Фаза А
UB	Фаза В
UC	Фаза С
Группа PhasesUpDown	Содержит каналы, в которых отображается наличие отклонения напряжения в фазах
UAup	Превышение напряжения по фазе А
UBup	Превышение напряжения по фазе В
UCup	Превышение напряжения по фазе С
UAdown	Понижение напряжения по фазе А
UBdown	Понижение напряжения по фазе В
UCdown	Понижение напряжения по фазе С
Группа PowerProfile	
Группа PEI	Передаются архивные значение активной мощности. Параметры каналов архивируются
E	Потребленная мощность
I	Отпущенная мощность
Группа Energy	
Группа EP	Содержит две группы каналов, в которых содержатся текущие значения активной энергии. В группе E - потребленная энергия, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы.
Группа EPD	Содержит две группы каналов, в которых содержатся текущие значения активной энергии за текущие сутки. В группе E - потребленная энергия,

Название	Назначение
	а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы.
Группа EPM	Содержит две группы каналов, в которых содержатся текущие значения активной энергии за текущий месяц. В группе E - потребленная энергия, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы.
Группа EnergyArchive	
Группа EPDLog	Содержит две группы каналов, в которых содержатся архивные суточные значения активной энергии. В группе E - потребленная энергия, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы. Параметры каналов архивируются.
Группа EPDendLog	Содержит две группы каналов, в которых содержатся архивные значения активной энергии на конец каждых суток. В группе E - потребленная энергия, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы. Параметры каналов архивируются.
Группа EPMLog	Содержит две группы каналов, в которых содержатся архивные месячные значения активной энергии. В группе E - потребленная энергия, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы. Параметры каналов архивируются.
Группа EPMendLog	Содержит две группы каналов, в которых содержатся архивные значения активной энергии на конец каждого месяца. В группе E - потребленная энергия, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы. Параметры каналов архивируются.
Группа Maximum	
Группа PMax	Содержит две группы каналов, в которых содержатся архивные значения активной мощности за каждый месяц. В группе E - потребленная мощность, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы. Параметры каналов архивируются.
Группа ActualPower	

Название	Назначение
Группа PActual	Содержит две группы каналов, в которых содержатся архивные значения фактической активной мощности за каждый месяц. В группе Е - потребленная мощность, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы. Параметры каналов архивируются.

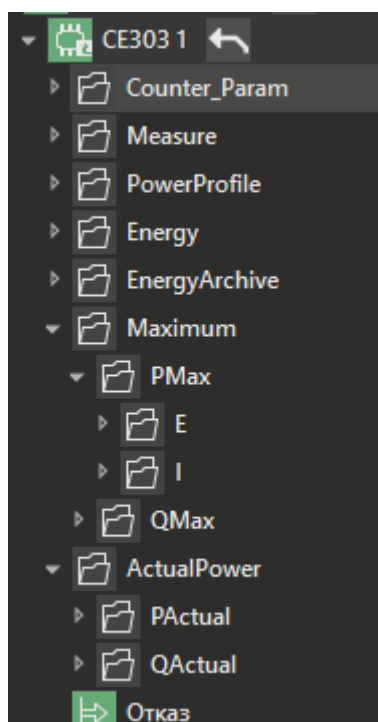
Каждый модуль ввода-вывода имеет выход Отказ

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если у архивируемого параметра группы PowerProfile снять флаг Архивировать, то этот канал выключится из опроса. Если у архивируемого параметра группы EnergyArchive снять флаг Архивировать, то опрос канала продолжится, но значение в канал не запишется.

6.2.3.2.12.3. CE303

Счетчики CE303 предназначены для измерения и многотарифного учета активной и реактивной электроэнергии в прямом и обратном направлении, ведения массивов профиля мощности нагрузки с программируемым временем интегрирования, фиксации максимумов мощности и измерения параметров трехфазной.

Вид счетчика добавленного в протокол в дереве системы:



Состав каналов счетчика:

Название	Назначение
Группа Counter_Param	
CurrentTime	Показывает текущее время, установленное в счетчике
Группа Measure	
Группа U A B C	Содержит каналы предназначенные для получения текущих значений напряжения для каждой фазы. Фаза А Фаза В Фаза С
Группа I A B C	Содержит каналы предназначенные для получения текущих значений тока для каждой фазы. Фаза А Фаза В Фаза С
Группа P A B C	Содержит каналы предназначенные для получения текущих значений активной мощности для каждой фазы. Фаза А Фаза В Фаза С
Группа Q A B C	Содержит каналы предназначенные для получения текущих значений реактивной мощности для каждой фазы. Фаза А Фаза В Фаза С

Название	Назначение
Группа AngleU AB BC CA	Содержит каналы предназначенные для получения текущих значений углов между фазами. Угол между фазами А и В Угол между фазами В и С Угол между фазами С и А
Группа AngleIU AB BC CA	Содержит каналы предназначенные для получения текущих значений углов между фазными векторами тока и напряжения. Угол между фазами А и В Угол между фазами В и С Угол между фазами С и А
Группа CosF SUM A B C	Содержит каналы для получения текущих значений коэффициентов активной мощности. Суммарное значение по всем фазам Фаза А Фаза В Фаза С
Группа TanF SUM A B C	Содержит каналы для получения текущих значений отношений реактивной к активной мощности. Суммарное значение по всем фазам Фаза А Фаза В Фаза С
Группа F F	Служит для получения значения частоты

Название	Назначение
Группа Bat Bat	Содержит информацию об уровне заряда батареи
Группа Phases UA UB UC	Содержит каналы, в которых отображается наличие или отсутствия напряжения в фазах Фаза А Фаза В Фаза С
Группа PhasesUpDown UAup UBup UCup UAdown UBdown UCdown	Содержит каналы, в которых отображается наличие отклонения напряжения в фазах Превышение напряжения по фазе А Превышение напряжения по фазе В Превышение напряжения по фазе С Понижение напряжения по фазе А Понижение напряжения по фазе В Понижение напряжения по фазе С
Группа PowerProfile	
Группа P E I	Передаются архивные значение активной мощности. Параметры каналов архивируются Потребленная мощность Отпущенная мощность
Группа Q E I	Передаются архивные значение реактивной мощности. Параметры каналов архивируются Потребленная мощность Отпущенная мощность

Название	Назначение
Группа Energy	
Группа EP	Содержит две группы каналов, в которых содержатся текущие значения активной энергии. В группе E - потребленная энергия, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы.
Группа EPD	Содержит две группы каналов, в которых содержатся текущие значения активной энергии за текущие сутки. В группе E - потребленная энергия, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы.
Группа EPM	Содержит две группы каналов, в которых содержатся текущие значения активной энергии за текущий месяц. В группе E - потребленная энергия, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы.
Группа EQ	Содержит две группы каналов, в которых содержатся текущие значения реактивной энергии. В группе E - потребленная энергия, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы.
Группа EQD	Содержит две группы каналов, в которых содержатся текущие значения реактивной энергии за текущие сутки. В группе E - потребленная энергия, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы.
Группа EQM	Содержит группы каналов, в которых содержатся текущие значения реактивной энергии за текущий месяц. В группе E - потребленная энергия, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы.
Группа EnergyArchive	
Группа EPDLog	Содержит две группы каналов, в которых содержатся архивные суточные значения активной энергии. В группе E - потребленная энергия, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы. Параметры каналов архивируются.

Название	Назначение
Группа EPDendLog	Содержит две группы каналов, в которых содержатся архивные значения активной энергии на конец каждых суток. В группе Е - потребленная энергия, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы. Параметры каналов архивируются.
Группа EPMLog	Содержит две группы каналов, в которых содержатся архивные месячные значения активной энергии. В группе Е - потребленная энергия, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы. Параметры каналов архивируются.
Группа EPMendLog	Содержит две группы каналов, в которых содержатся архивные значения активной энергии на конец каждого месяца. В группе Е - потребленная энергия, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы. Параметры каналов архивируются.
Группа EQDLog	Содержит две группы каналов, в которых содержатся архивные суточные значения реактивной энергии. В группе Е - потребленная энергия, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы. Параметры каналов архивируются.
Группа EQDendLog	Содержит две группы каналов, в которых содержатся архивные значения реактивной энергии на конец каждых суток. В группе Е - потребленная энергия, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы. Параметры каналов архивируются.
Группа EQMLog	Содержит две группы каналов, в которых содержатся архивные месячные значения реактивной энергии. В группе Е - потребленная энергия, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы. Параметры каналов архивируются.
Группа EQMendLog	Содержит две группы каналов, в которых содержатся архивные значения реактивной энергии на конец каждого месяца. В группе Е - потребленная энергия, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы. Параметры каналов архивируются.
Группа Maximum	

Название	Назначение
Группа PMax	Содержит две группы каналов, в которых содержатся архивные значения активной мощности за каждый месяц. В группе E - потребленная мощность, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы. Параметры каналов архивируются.
Группа QMax	Содержит две группы каналов, в которых содержатся архивные значения реактивной мощности за каждый месяц. В группе E - потребленная мощность, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы. Параметры каналов архивируются.
Группа ActualPower	
Группа PActual	Содержит две группы каналов, в которых содержатся архивные значения фактической активной мощности за каждый месяц. В группе E - потребленная мощность, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы. Параметры каналов архивируются.
Группа QActual	Содержит две группы каналов, в которых содержатся архивные значения фактической реактивной мощности за каждый месяц. В группе E - потребленная мощность, а в группе I - отпущенная. Каждая группа содержит значения каждого тарифа и их суммы. Параметры каналов архивируются.

Каждый модуль ввода-вывода имеет выход Отказ

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если у архивируемого параметра группы PowerProfile снять флаг Архивировать, то этот канал выключится из опроса. Если у архивируемого параметра группы EnergyArchive снять флаг Архивировать, то опрос канала продолжится, но значение в канал не запишется.

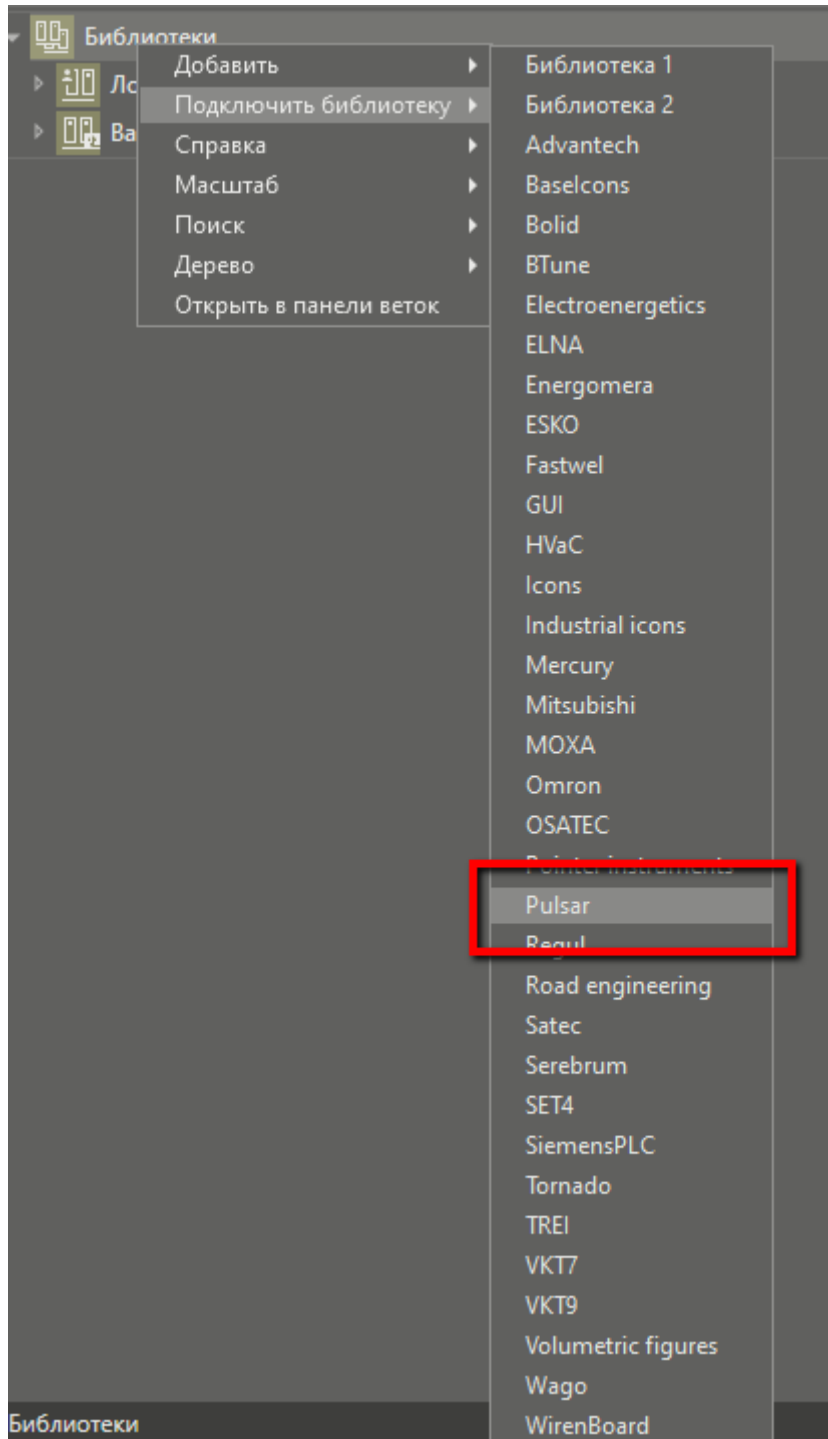
6.2.3.2.13.PULSAR

MasterSCADA 4D поддерживает работу с различными модификациями импульсных счетчиков-регистраторов компании Тепловодохран.

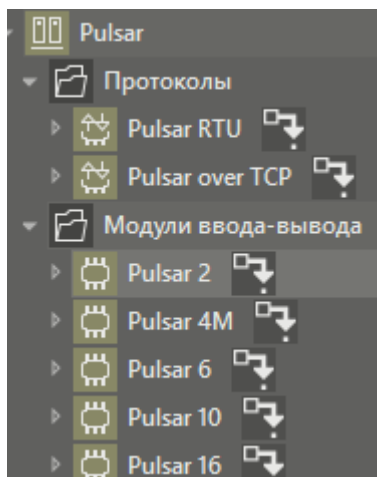
При обмене узел играет роль ведущего. Поддерживается считывание архивов из счетчика. Работа с этими устройствами возможна через последовательные интерфейсы (RS-232, RS-485, GSM), а при использовании конвертеров Ethernet-COM (Моха NPort 5150 и т.п.) также и по сетям TCP/IP (Ethernet, Wi-Fi).

Важно! По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте sales@masterscada.ru

Для получения возможности использования этих счетчиков в проекте необходимо подключить соответствующую библиотеку:

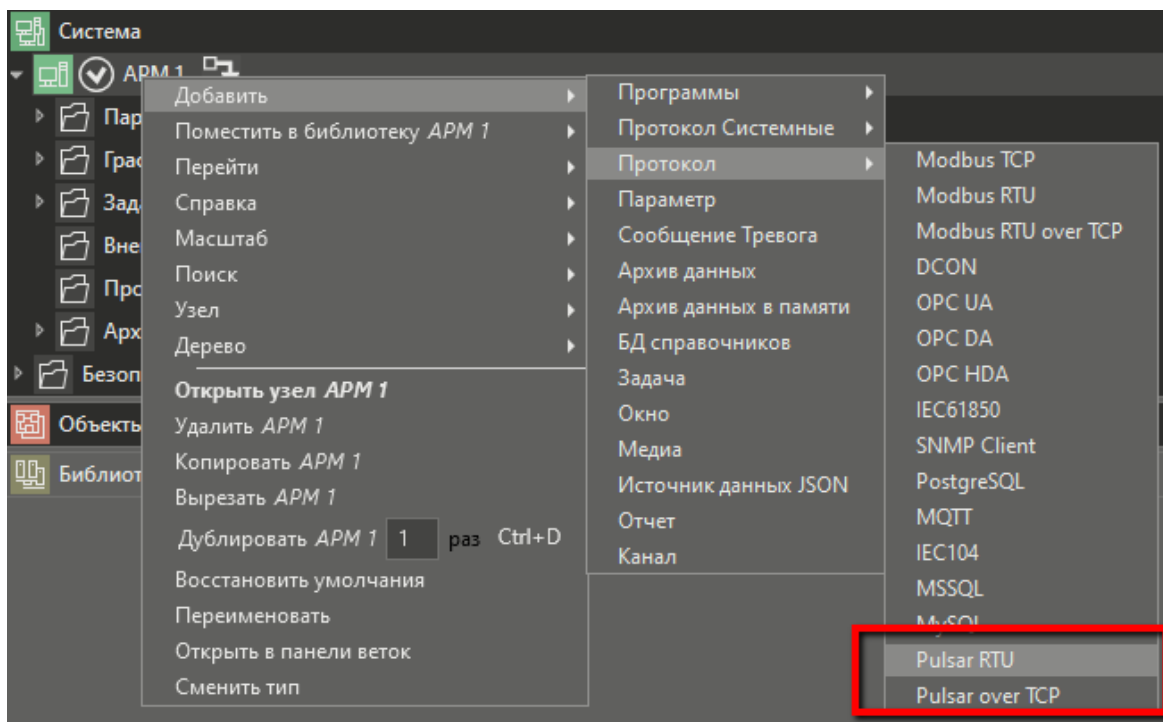


В полном дереве библиотека Pulsar имеет вид:



После этого в проект могут быть добавлены сначала протокол, а затем устройство, любым известным способом, например, при помощи контекстного меню узла, или с использованием контекстной панели, или путем перетаскивания элемента из библиотеки в дерево системы.

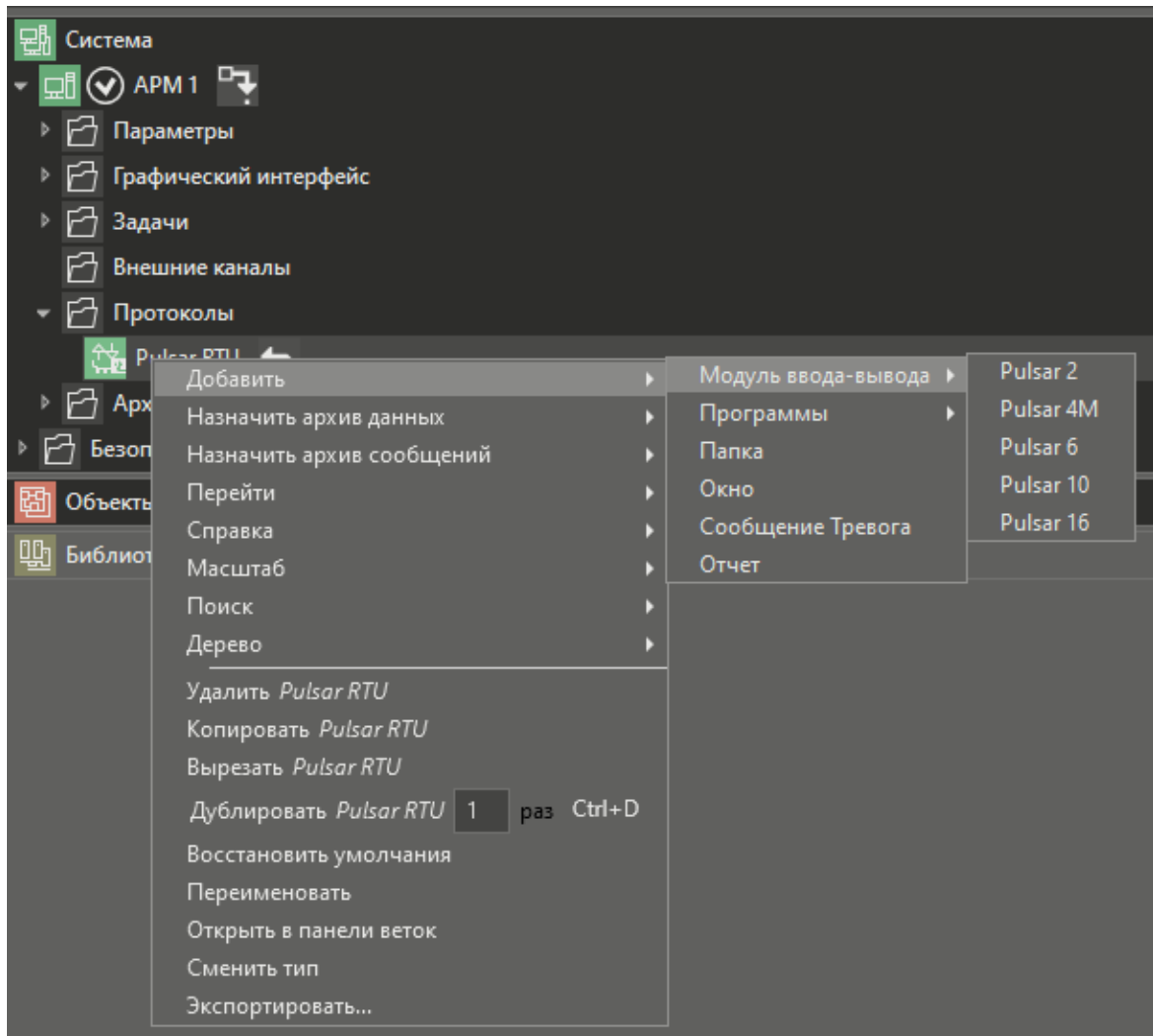
Рекомендуется добавлять элементы в проект при помощи контекстного меню:



Тип добавляемого протокола зависит от реального способа подключения счетчика:

- Pulsar RTU - если счетчик подключен по последовательному интерфейсу. Настройки этого протокола практически полностью соответствуют настройкам Modbus RTU.
- Pulsar over TCP - если счетчик подключен через конвертер Ethernet-COM. Настройки этого протокола соответствуют настройкам Modbus RTU over TCP

Затем при помощи контекстного меню протокола можно добавить модули ввода-вывода, которые уже содержат все необходимые каналы:



Каждый счетчик имеет predetermined набор каналов, который характерен для него. Поддержаны наиболее используемые функции счетчиков.

Таблица каналов:

Название	Назначение
CurrentTime	Канал служит для получения текущего времени устройства
Группа Current	Содержит считанные из прибора значения представленные в виде нарастающего итога. Группа содержит набор каналов, по количеству каналов прибора.

Название	Назначение
Группа ArchiveHour	Группа содержит каналы для получения архивных часовых значений. Может быть получено 1080 записей. Группа содержит набор каналов, по количеству каналов прибора.
Группа ArchiveDay	Группа содержит каналы для получения архивных суточных значений. Может быть получено 180 записей. Группа содержит набор каналов, по количеству каналов прибора.
Группа ArchiveMonth	Группа содержит каналы для получения архивных месячных значений. Может быть получено 24 записи. Группа содержит набор каналов, по количеству каналов прибора.

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если у архивируемого параметра снять флаг Архивировать, то этот канал выключится из опроса.

Поддерживаются следующие модули ввода-вывода:

Pulsar 2 - 2-канальный счетчик импульсов.

Pulsar 4M - 4-канальный счетчик импульсов.

Pulsar 6 - 6-канальный счетчик импульсов.

Pulsar 10 - 10-канальный счетчик импульсов.

Pulsar 16 - 15-канальный счетчик импульсов

После добавления в дерево проекта модулей ввода-вывода, настраивается панель свойств модулей.

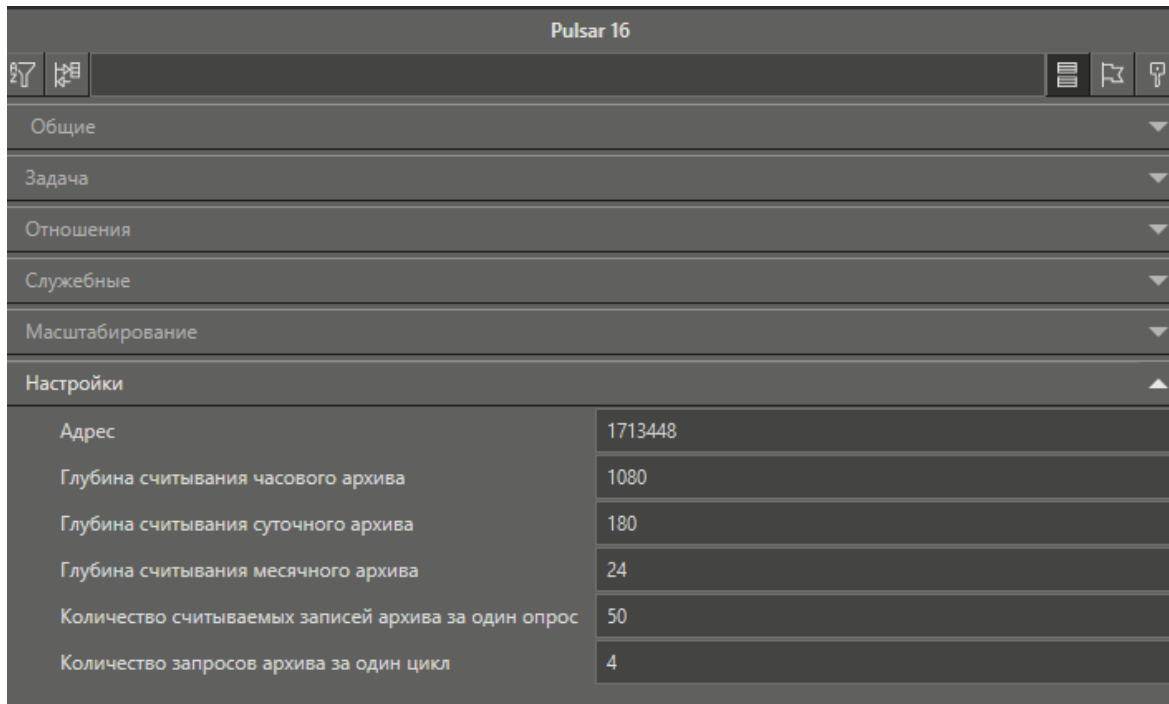
Каждый модуль ввода-вывода имеет выход Отказ, а также группы каналов, согласно его максимальной конфигурации.

Если в опросе каких-либо каналов нет необходимости, то можно либо удалить их из конфигурации элемента уже после добавления его в дерево системы, либо сначала создать экземпляр модуля в пользовательской библиотеке, затем из его конфигурации удалить незадействованные каналы, после чего использовать в проекте уже новый видоизмененный библиотечный элемент.

Важно! Конфигурирование устройства, в том числе задание веса импульса, необходимо выполнять с помощью сервисного ПО предоставляемого производителем устройства.

6.2.3.2.13.1. СВОЙСТВА МОДУЛЕЙ PULSAR

Вид панели свойств:



Описание:

Название	Описание
Категория Общие	Как правило, для данного элемента эта категория не настраивается. Подробное описание данной категории приведено в соответствующем разделе.
Категория Задача	Задается способ опроса устройства. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи.
Категория Масштабирование	Если модуль ввода-вывода не имеет встроенного масштабирования, то при помощи данной категории можно задать соответствие между значениями в модуле и реальными значениями. Подробное описание данной категории приведено в соответствующем разделе.
Категория Настройки	

Название	Описание
Адрес	Задается адрес устройства в сети (значение адреса определяется документацией на подключаемое устройство).
Глубина считывания часового архива	Определяет количество считываемых при старте исполнительной системы записей часового архива. По умолчанию задано максимальное значение
Глубина считывания суточного архива	Определяет количество считываемых при старте исполнительной системы записей суточного архива. По умолчанию задано максимальное значение
Глубина считывания месячного архива	Определяет количество считываемых при старте исполнительной системы записей месячного архива. По умолчанию задано максимальное значение
Количество считываемых записей архивов за один запрос	Определяет количество считываемых каждый цикл опроса записей из архива счетчика
Количество запросов архива за один цикл	Архивные каналы прибора опрашиваются поочередно, каждый такт опроса (при наличии не считанных архивных записей). Настройка определяет сколько максимально может быть выполнено опросов архива каждого канала за один цикл опроса архивной группы. Например, если настройка Количество считываемых записей архива за запрос установлено в 50, а данная настройка в 4, то за один запрос будет считано не более 200 записей архива каждого канала.

При старте исполнительная система считывает архив прибора на заданную глубину. При этом за один цикл опроса архивной группы считывается $\text{Количество архивных записей} = \text{Количество считываемых записей архива за запрос} * \text{Количество запросов архива канала за один цикл}$. После того как все архивные записи были считаны, при каждом опросе устройства происходит опрос времени прибора и сравнивается со временем последней считанной архивной записи - если отчетный период прошел (закончился час, день или месяц) происходит считывание новых архивных записей.

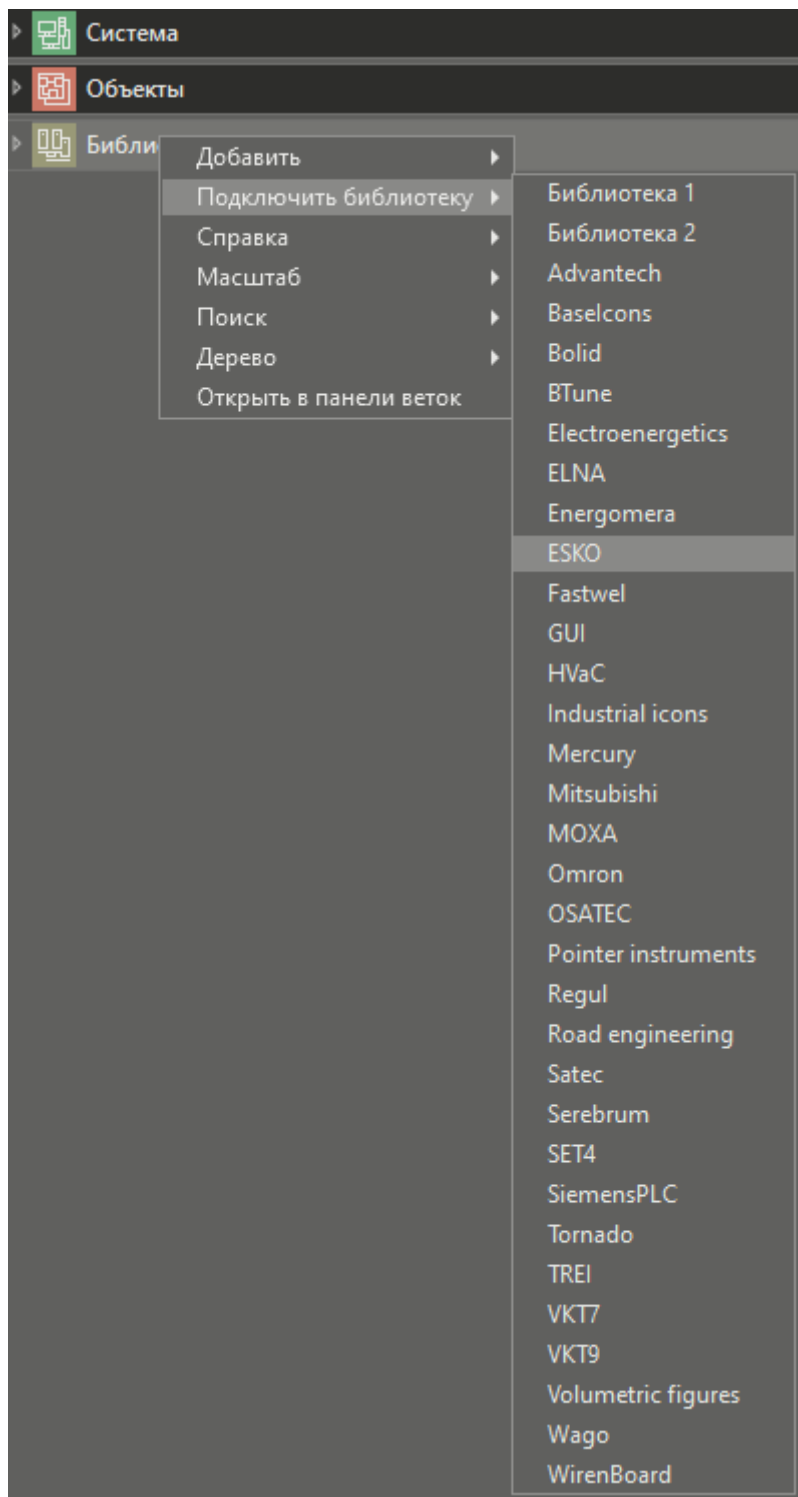
6.2.3.2.14.ESKO

MasterSCADA 4D поддерживает работу с теплосчетчиком ЭСКО-Т компании Энергосервисная компания ЗЭ.

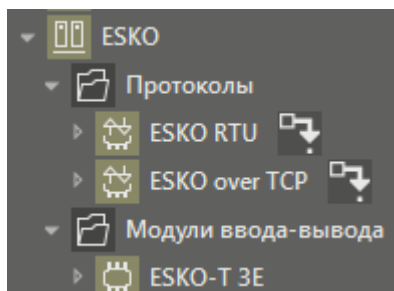
При обмене узел играет роль ведущего. Поддерживается считывание архивов из счетчика. Работа с этими устройствами возможна через последовательные интерфейсы (RS-232, RS-485, GSM), а при использовании конвертеров Ethernet-COM (Моха NPort 5150 и т.п.) также и по сетям TCP/IP (Ethernet, Wi-Fi).

Важно! По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте sales@masterscada.ru

Для получения возможности использования этих счетчиков в проекте необходимо подключить соответствующую библиотеку:

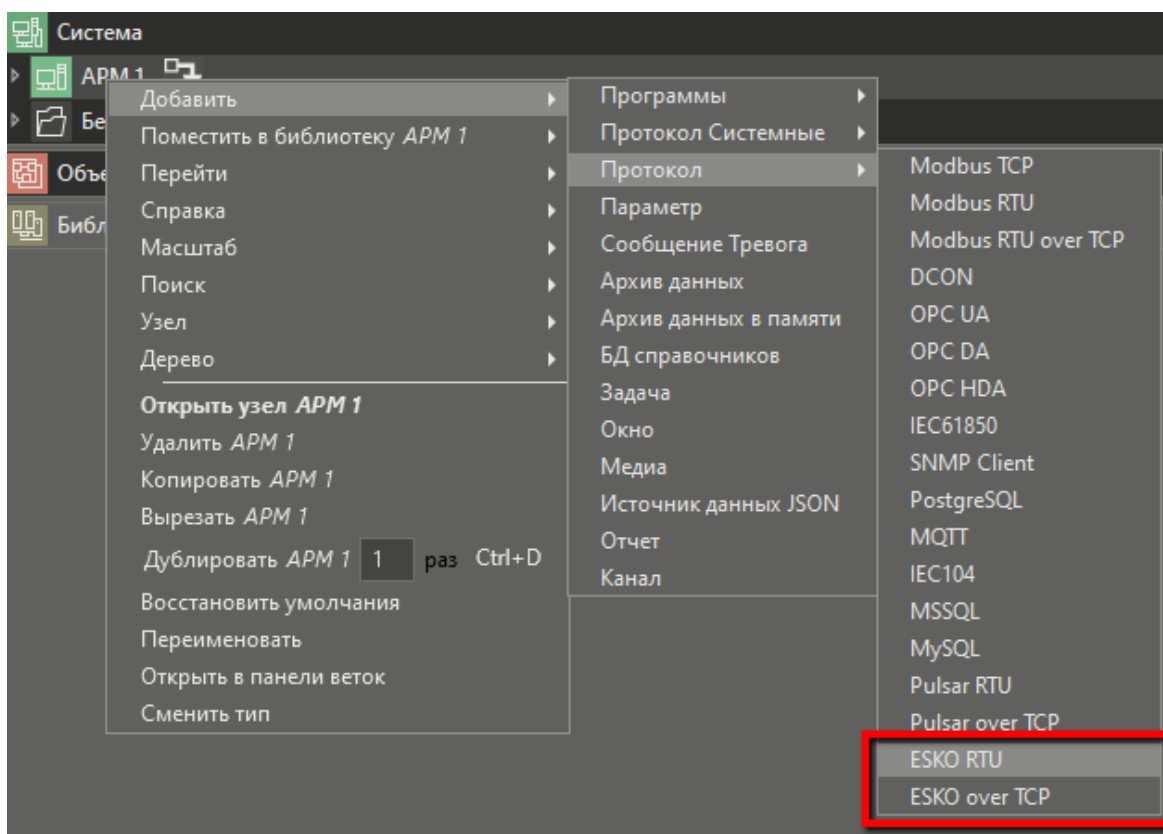


В полном дереве библиотека ESKO имеет вид:



После этого в проект могут быть добавлены сначала протокол, а затем устройство, любым известным способом, например, при помощи контекстного меню узла, или с использованием контекстной панели, или путем перетаскивания элемента из библиотеки в дерево системы.

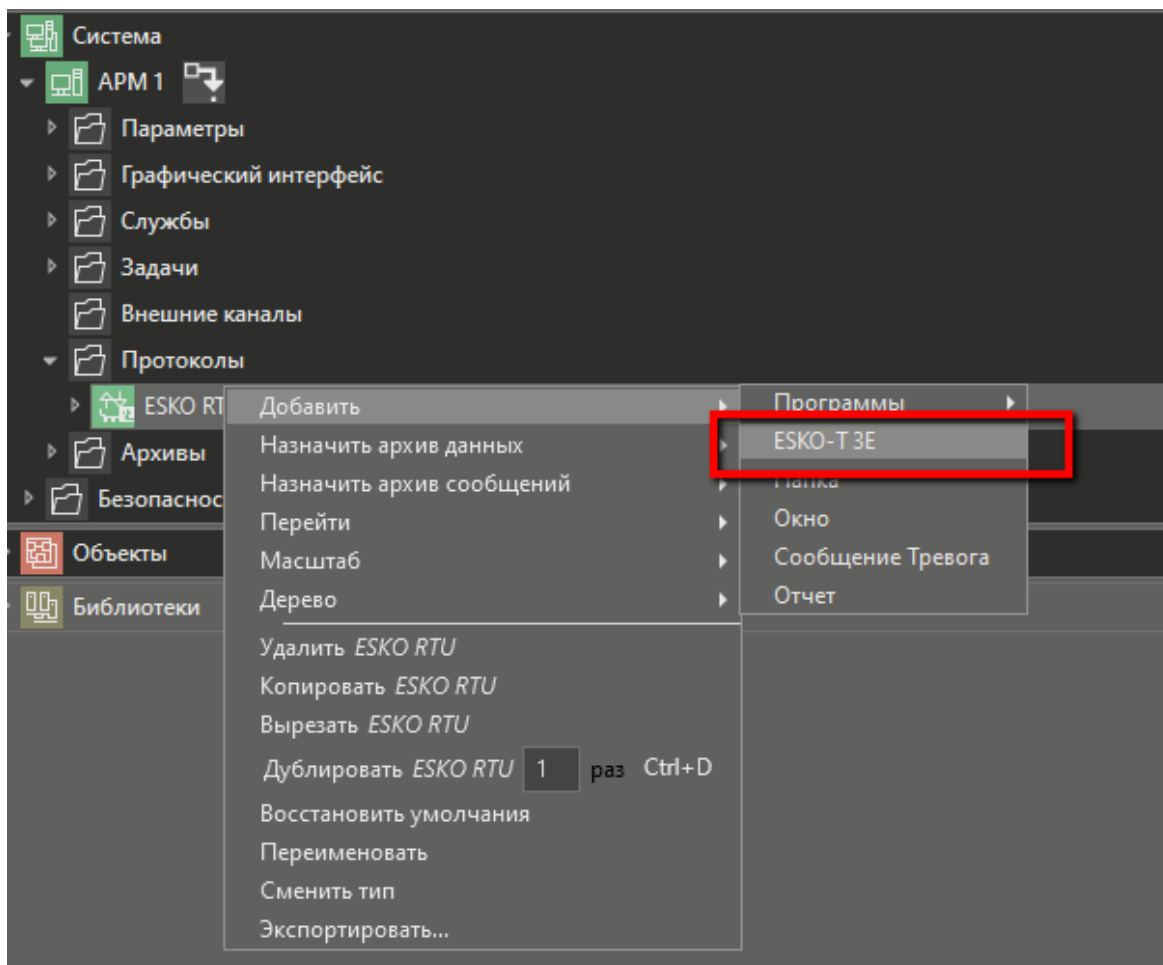
Рекомендуется добавлять элементы в проект при помощи контекстного меню:



Тип добавляемого протокола зависит от реального способа подключения счетчика:

- ESKO RTU - если счетчик подключен по последовательному интерфейсу. Настройки этого протокола практически полностью соответствуют настройкам Modbus RTU.
- ESKO over TCP - если счетчик подключен через конвертер Ethernet-COM. Настройки этого протокола соответствуют настройкам Modbus RTU over TCP

Затем при помощи контекстного меню протокола можно добавить модуль ввода-вывода ESKO-T, который уже содержит все необходимые каналы:



Счетчик имеет predetermined набор каналов, который характерен для него. Поддержаны наиболее используемые функции. Описание каналов устройства смотрите в разделе Модуль ESKO-T

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если снять флаг, то данные от прибора будут запрашиваться, но значение в канал поступать не будет.

После добавления в дерево проекта модулей ввода-вывода, настраивается панель свойств модуля.

Каждый модуль ввода-вывода имеет выход Отказ, а также группы каналов, согласно его максимальной конфигурации.

Если в опросе каких-либо каналов нет необходимости, то можно либо удалить их из конфигурации элемента уже после добавления его в дерево системы, либо сначала создать экземпляр модуля в пользовательской библиотеке, затем из его конфигурации удалить недействующие каналы, после чего использовать в проекте уже новый видоизмененный библиотечный элемент.

6.2.3.2.14.1. МОДУЛЬ ESKO-T

Элемент, соответствующий реальному теплосчетчику ЭСКО-Т, имеет выход Отказ, а также группы каналов, согласно его максимальной конфигурации:

Таблицы каналов:

Название	Назначение
CurrentTime	Канал служит для получения текущего времени устройства
Группа Current	Содержит считанные из прибора текущие значения. Группа содержит следующие каналы: T1-T3 - Температура по каждому каналу измерений, С P1-P3 - Давление по каждому каналу измерений, МПа Gv1-Gv3 -Объемный расход по каждому каналу измерений, м3/ч Gm1-Gm3 - Массовый расход по каждому каналу измерений, т/ч
Группа Total	Группа содержит каналы для получения накопленных значений, которые хранятся в приборе. Группа содержит следующие каналы: Q -Накопленная энергия, ГКал V1-V3 -Накопленный объем по каждому каналу измерений, м3 M1-M3 - Накопленная масса по каждому каналу измерений, т
Группа Archive	Содержит три дочерние группы, описанные ниже. Структура всех трех групп одинаковая: Q -Энергия, Гкал M1-M3 - Массовый расход по каждому каналу измерений, т T1-T2 - Средняя температура, С Tsw - Средняя температура холодной воды, С P1-P2 - Давление среднее, МПа Psw - Среднее давление холодной воды, МПа
Группа ArchiveHour	Группа содержит каналы для получения архивных часовых значений. Может быть получено 1504 записей.

Название	Назначение
Группа ArchiveDay	Группа содержит каналы для получения архивных суточных значений. Может быть получено 512 записей
Группа ArchiveMonth	Группа содержит каналы для получения архивных месячных значений. Может быть получено 32 записи.

6.2.3.2.14.2. СВОЙСТВА МОДУЛЕЙ ESKO

Вид панели свойств:



Описание:

Название	Описание
Категория Общие	Как правило, для данного элемента эта категория не настраивается. Подробное описание данной категории приведено в соответствующем разделе.

Название	Описание
Категория Задача	Задается способ опроса устройства. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи.
Категория Масштабирование	Если модуль ввода-вывода не имеет встроенного масштабирования, то при помощи данной категории можно задать соответствие между значениями в модуле и реальными значениями. Подробное описание данной категории приведено в соответствующем разделе.
Категория Настройки	
Адрес	Задается адрес устройства в сети (значение адреса определяется документацией на подключаемое устройство).
Глубина считывания часового архива	Определяет количество считываемых при старте исполнительной системы записей часового архива. По умолчанию задано максимальное значение
Глубина считывания суточного архива	Определяет количество считываемых при старте исполнительной системы записей суточного архива. По умолчанию задано максимальное значение
Глубина считывания месячного архива	Определяет количество считываемых при старте исполнительной системы записей месячного архива. По умолчанию задано максимальное значение
Количество считываемых записей архивов за цикл	Определяет количество считываемых каждый цикл опроса записей из архива счетчика

6.2.3.2.15. ТЕПЛОВЫЧИСЛИТЕЛИ ТЕПЛОКОМ

MasterSCADA 4D поддерживает работу с тепловычислителями (теплосчетчиками) компании Теплоком - ВКТ-7 (VKT7) и ВКТ-9 (VKT9).

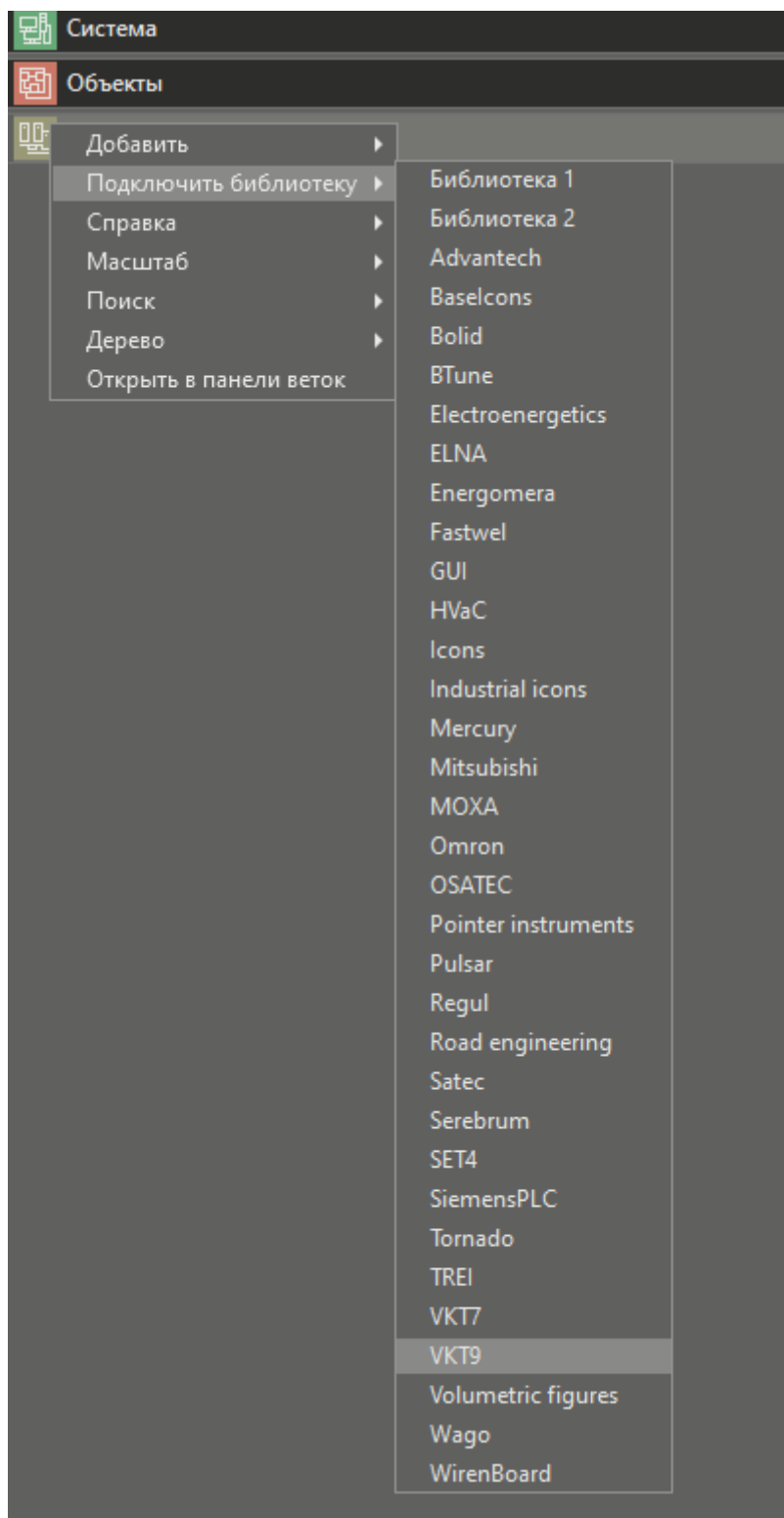
Работа с этими устройствами возможна через последовательные интерфейсы (RS-232, RS-485), а при использовании конвертеров Ethernet-COM (Моха NPort 5150 и т.п.) также и по сетям TCP/IP (Ethernet, Wi-Fi).

6.2.3.2.15.1. VKT9

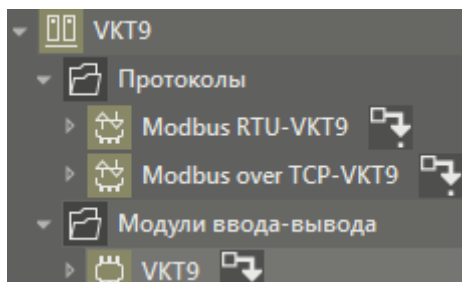
MasterSCADA 4D RT работает с теплосчетчиками VKT9 компании Теплоком по платформонезависимым протоколам, которые поддерживаются исполнительными системами, предназначенными для различных операционных систем.

Важно! По умолчанию протокол не входит в состав исполнительной системы. Стоимость и условия лицензирования этого протокола необходимо уточнять по электронной почте sales@masterscada.ru

Для получения возможности использования теплосчетчиков VKT9 в проекте необходимо подключить соответствующую библиотеку:

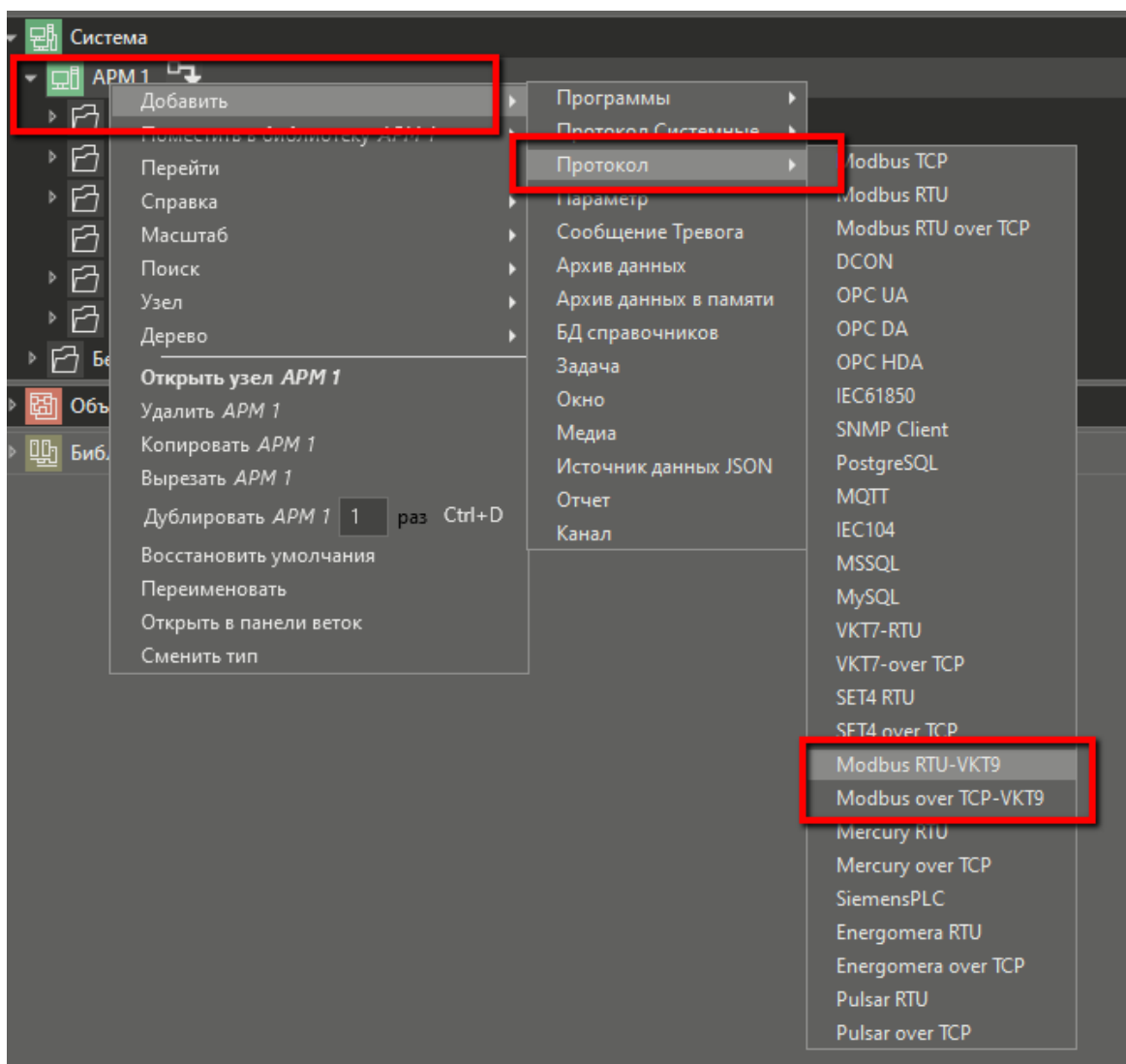


В полном дереве библиотека VKT9 имеет вид:



После этого в проект могут быть добавлены сначала протокол, а затем устройство, любым известным способом, например, при помощи контекстного меню узла, или с использованием контекстной панели, или путем перетаскивания элемента из библиотеки в дерево системы.

Рекомендуется добавлять элементы в проект при помощи контекстного меню:

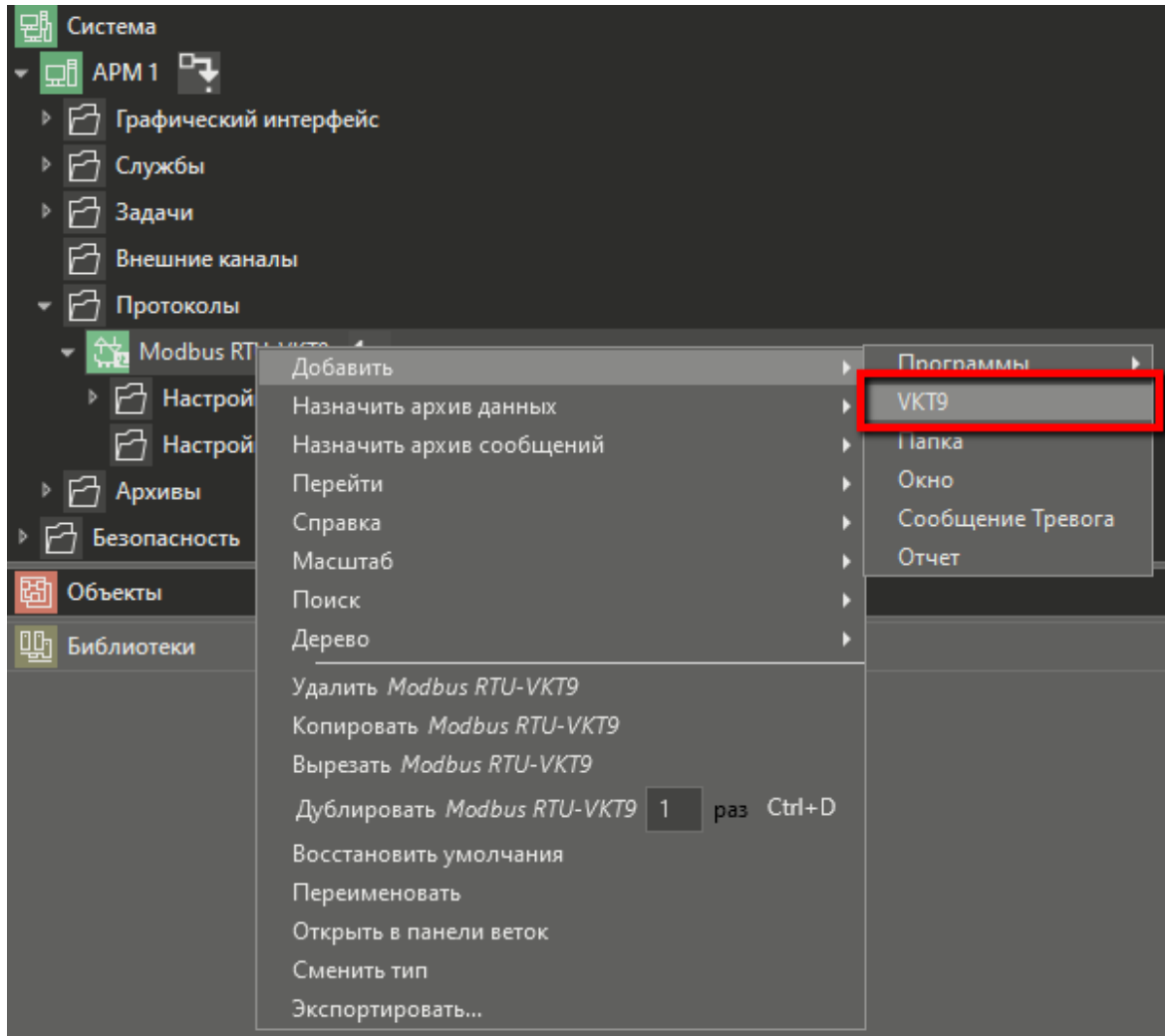


Тип добавляемого протокола зависит от реального способа подключения теплосчетчика:

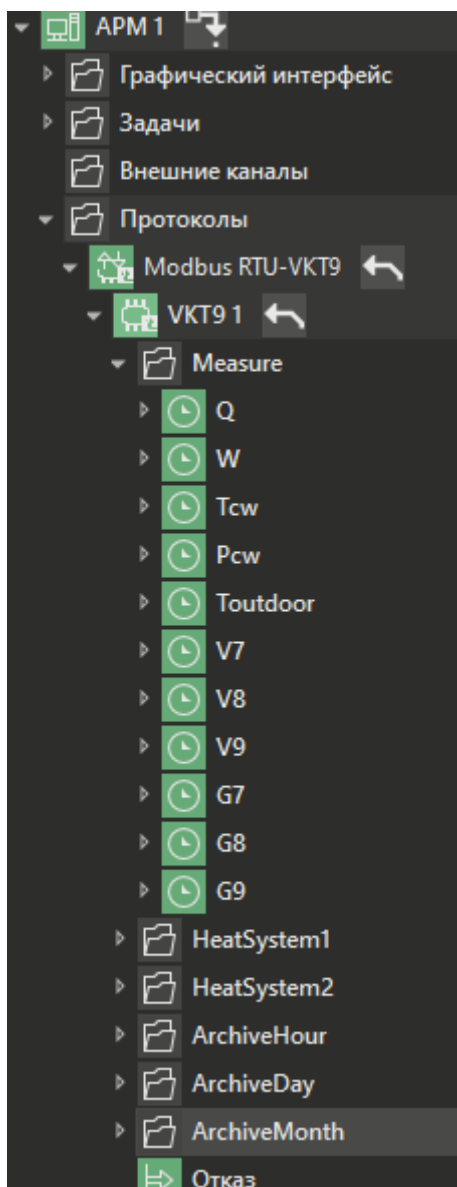
- Modbus RTU-VKT9 - если счетчик подключен по последовательному интерфейсу. Настройки этого протокола соответствуют настройкам Modbus RTU.

- Modbus over TCP-VKT9 - если счетчик подключен через конвертер Ethernet-COM. Настройки этого протокола соответствуют настройкам Modbus RTU over TCP

После этого, в настроенный протокол добавляется элемент, соответствующий теплосчетчику VKT9:



В дереве системы элемент VKT9 имеет вид:



Элемент VKT9 имеет выход Отказ, а также группы каналов, согласно его максимальной конфигурации. Подробное описание смотрите в разделе Тепловычислитель VKT9.

Если в опросе каких-либо каналов нет необходимости, то можно либо удалить их из конфигурации элемента VKT9 уже после добавления его в дерево системы, либо сначала создать экземпляр элемента VKT9 в пользовательской библиотеке, затем из его конфигурации удалить незадействованные каналы, после чего использовать в проекте уже новый видоизмененный библиотечный элемент.

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если снять флаг, то данные от прибора будут запрашиваться, но значение в канал поступать не будет.

Для корректной работы добавленного элемента необходимо настроить его панель свойств.

СВОЙСТВА VKT9

Панель свойств элемента VKT9 имеет вид:

The screenshot shows a software interface for configuring an element named 'VKT9 1'. The interface is organized into sections:

- Общие (General):**
 - Имя (Name): VKT9 1
 - Метки (Tags):
 - Комментарий (Comment):
- Служебные (Service):**
 - Задача (Task):
 - Масштабирование (Scaling):
- Настройки (Settings):**
 - Адрес (Address): 0
 - Максимальный разрыв адресов (Maximum address gap): 5
 - Отслеживать качество измерений (Track measurement quality):
 - Количество считываемых записей архива за цикл (Number of archive records per cycle): 5

Название	Описание
Категория Общие	Как правило, для данного элемента эта категория не требует настройки. Подробное описание данной категории приведено в соответствующем разделе.
Категория Задача	Задаёт способ опроса устройства. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи.
Категория Масштабирование	Если модуль ввода-вывода не имеет встроенного масштабирования, то при помощи данной категории может быть задано соответствие между значениями в модуле и реальными значениями. Подробное описание данной категории приведено в соответствующем разделе.
Категория Настройки	
Адрес	Указывает адрес счетчика в сети.

<p>Максимальный разрыв адресов</p>	<p>Определяет максимально допустимое количество пропущенных адресов между смежными адресами в запросе чтения. Например, когда производится опрос регистров с номерами - 0, 1, 2, 4, 5., то в случае если значение данного свойства задано равным 0, все регистры будут прочитаны за два запроса (0-2, 4-5), а в случае если для данного свойства будет задано значение 1 или выше, все регистры будут прочитаны за один запрос (0-5, а не используемый регистр 3 будет проигнорирован при разборе ответа. Увеличение значения данного свойства сокращает количество запросов и ускоряет опрос, а его уменьшение делает пакеты более устойчивыми к помехам. Следует также иметь ввиду, что увеличение значения данного свойства может приводить к формированию слишком длинных пакетов, которые прибор будет не способен передать, и, по этой причине, выдавать код ошибки. По умолчанию установлена оптимальная настройка, ориентированная на уменьшение количества запросов. Если опрос вашего прибора постоянно происходит с ошибкой, то значение свойства следует уменьшить или задать равным 0.</p>
<p>Отслеживать качество измерений</p>	<p>Определяет будет ли из прибора считываться информация о качестве полученных данных. Измеряемые параметры теплосчетчика имеют специальные регистры определяющие достоверность измерения (обрыв датчика, отключение датчика и т.д.). Если этот флаг отмечен, то исполнительная система считывает данные регистры, и в случае отказа выставляет плохой признак качества (2156658688, OpCUa_BadSensorFailure)</p>
<p>Количество считываемых записей архива за цикл</p>	<p>Задаёт максимальное количество архивных значений, считываемых из прибора за один цикл опроса.</p> <p>При первом подключении исполнительной системы к прибору, когда в базе данных исполнительной системы еще отсутствуют записи от этого прибора, из него будут считаны все накопленные архивные значения. А в том случае, если в базе данных уже содержатся записи от этого прибора, то исполнительной системой будут считаны архивные значения, накопленные в приборе со времени последней записи в базе данных по текущий момент.</p>

	<p>На каждом цикле опроса прибор будет отправлять количество значений, не превышающее указанное в данном свойстве.</p> <p>После того, как из прибора будут считаны и сохранены все ранее накопленные архивные значения, исполнительная система станет получать из него архивные значения только по мере их формирования.</p>
--	--

ТЕПЛОВЫЧИСЛИТЕЛЬ VKT9

Элемент, соответствующий реальному тепловычислителю VKT9, имеет выход Отказ, а также группы каналов, согласно его максимальной конфигурации:

Название группы	Назначение
Measure	Общие измеренные значения
HeatSystem1	Параметры теплосистемы 1
HeatSystem2	Параметры теплосистемы 2
ArchiveHour	Часовые архивы
ArchiveDay	Суточные архивы
ArchiveMonth	Месячные архивы

Состав группы Measure:

Название канала	Описание
Q	Общая тепловая энергия
W	Мощность общая
Tcw	Температура холодной воды
Pcw	Давление холодной воды

Название канала	Описание
Toutdoor	Температура наружного воздуха
V7	Накопленный объем по каналу 7
V8	Накопленный объем по каналу 8
V9	Накопленный объем по каналу 9
G7	Расход по каналу 7
G8	Расход по каналу 8
G9	Расход по каналу 9

Состав групп HeatSystem:

Название канала	Назначение	Название канала	Назначение
Qo	Тепловая энергия отопления	Gm1	Массовый расход канала 1
Qgvs	Тепловая энергия ГВС	Gm2	Массовый расход канала 2
M1	Масса канала 1	Gm3	Массовый расход канала 3
M2	Масса канала 2	dGm	Разница массового расхода
M3	Масса канала 3	Gv1	Объемный расход канала 1
M1R	Масса обратная канала 1	Gv2	Объемный расход канала 2
M2R	Масса обратная канала 2	Gv3	Объемный расход канала 3
M3R	Масса обратная канала 3	T1	Температура канала 1

Название канала	Назначение	Название канала	Назначение
dM	Разность масс	T2	Температура канала 2
V1	Объем канала 1	T3	Температура канала 3
V2	Объем канала 2	dT1	Разница температур канала 1
V3	Объем канала 3	dT2	Разница температур канала 2
V1R	Обратный объем канала 1	dT3	Разница температур канала 3
V2R	Обратный объем канала 2	P1	Давление канала 1
V3R	Обратный объем канала 3	P2	Давление канала 2
Wo	Мощность отопления	P3	Давление канала 3
Wgvs	Мощность ГВС		

Состав групп Archive:

Название канала	Назначение	Название канала	Назначение
Q	Тепловая энергия общая	HS1_P1	Давление канала 1
Tsw	Температура холодной воды	HS1_P2	Давление канала 2
Psw	Давление холодной воды	HS1_P3	Давление канала 3
Toutdoor	Температура наружного воздуха	HS2_Qo	Тепловая энергия отопления
V7	Объем в канале 7	HS2_Qgvs	Тепловая энергия ГВС

Название канала	Назначение	Название канала	Назначение
V8	Объем в канале 8	HS2_M1	Масса канал 1
V9	Объем в канале 9	HS2_M2	Масса канал 2
HS1_Qo	Тепловая энергия отопления	HS2_M3	Масса канал 3
HS1_Qgvs	Тепловая энергия ГВС	HS2_M1R	Обратная масса канал 1
HS1_M1	Масса канал 1	HS2_M2R	Обратная масса канал 2
HS1_M2	Масса канал 2	HS2_M3R	Обратная масса канал 3
HS1_M3	Масса канал 3	HS2_dM	Разница масс
HS1_M1R	Обратная масса канал 1	HS2_V1	Объем канала 1
HS1_M2R	Обратная масса канал 2	HS2_V2	Объем канала 2
HS1_M3R	Обратная масса канал 3	HS2_V3	Объем канала 3
HS1_dM	Разница масс	HS2_V1R	Обратный объем канала 1
HS1_V1	Объем канала 1	HS2_V2R	Обратный объем канала 2
HS1_V2	Объем канала 2	HS2_V3R	Обратный объем канала 3
HS1_V3	Объем канала 3	HS2_T1	Температура канала 1
HS1_V1R	Обратный объем канала 1	HS2_T2	Температура канала 2
HS1_V2R	Обратный объем канала 2	HS2_T3	Температура канала 3

Название канала	Назначение	Название канала	Назначение
HS1_V3R	Обратный объем канала 3	HS2_dT1	Разница температур канала 1
HS1_T1	Температура канала 1	HS2_dT2	Разница температур канала 2
HS1_T2	Температура канала 2	HS2_dT3	Разница температур канала 3
HS1_T3	Температура канала 3	HS2_P1	Давление канала 1
HS1_dT1	Разница температур канала 1	HS2_P2	Давление канала 2
HS1_dT2	Разница температур канала 2	HS2_P3	Давление канала 3
HS1_dT3	Разница температур канала 3		

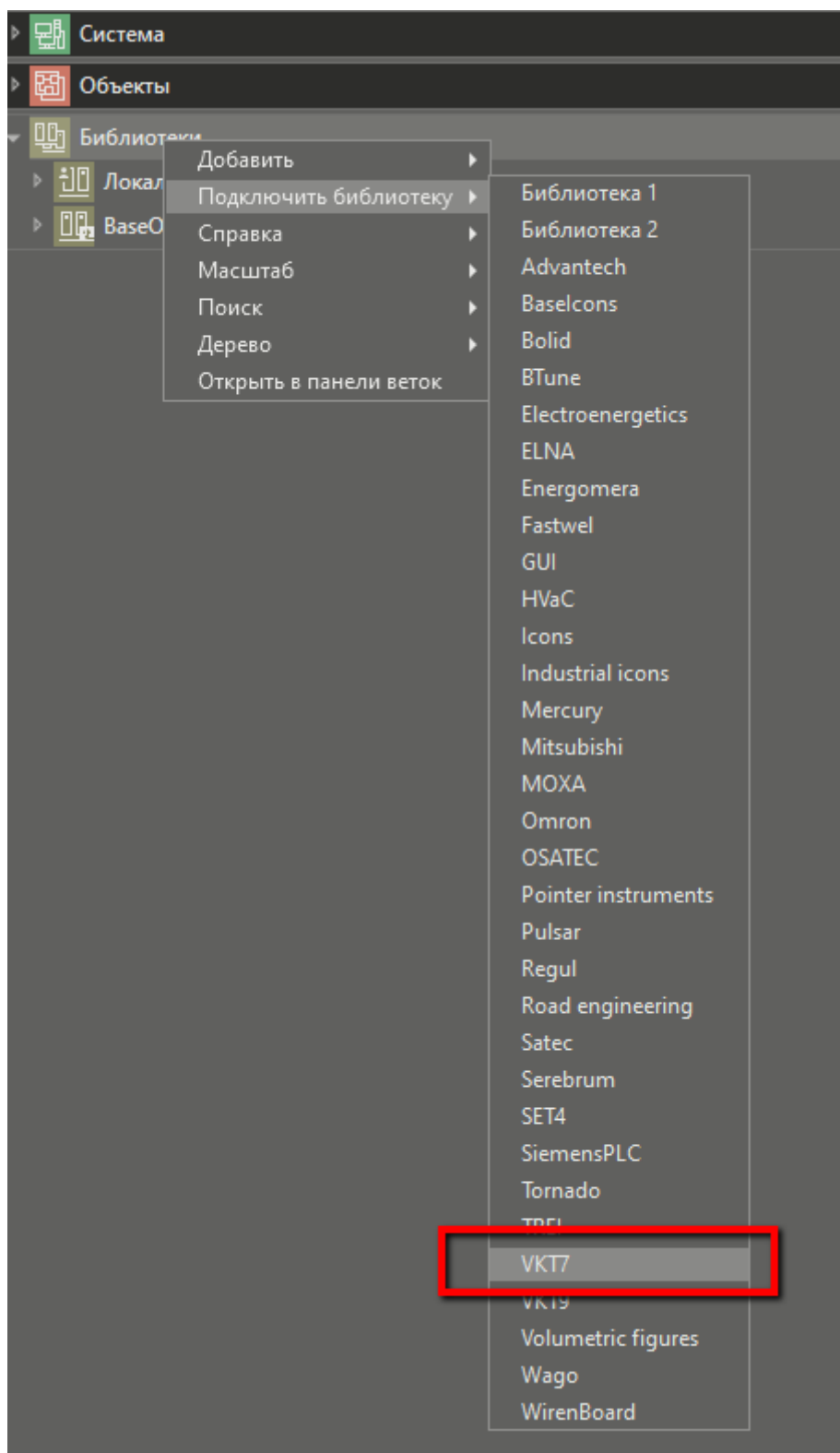
Каналы содержат типовые параметры.

6.2.3.2.15.2. VKT7

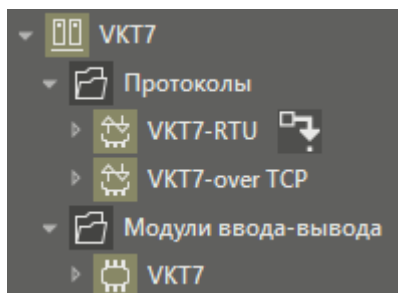
MasterSCADA 4D RT может работать с теплосчетчиками VKT7 компании Теплоком по платформонезависимому протоколу, который может использоваться в исполнительных системах, предназначенных для различных операционных систем.

Важно! По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте sales@masterscada.ru

Чтобы была возможность использовать теплосчетчики VKT7 в проекте необходимо подключить соответствующую библиотеку:

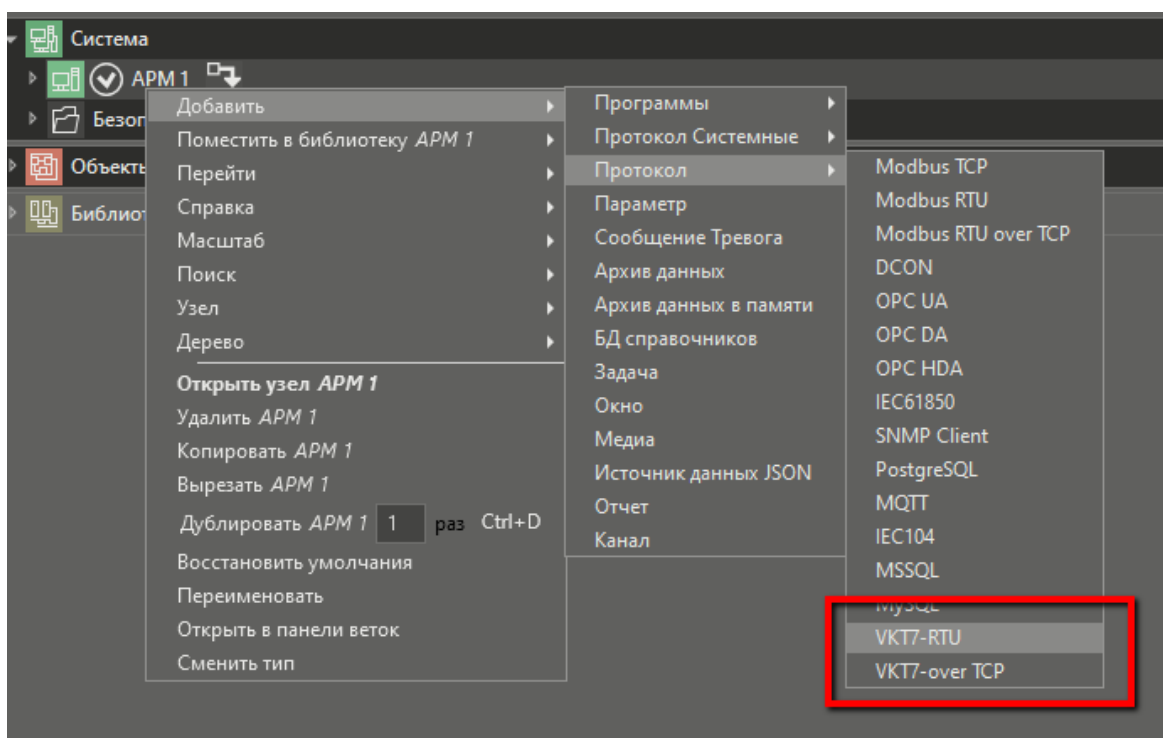


В полном дереве библиотека VKT7 имеет вид:



После этого в проект могут быть добавлены сначала протокол, а затем устройство, любым известным способом, например, при помощи контекстного меню узла, или с использованием контекстной панели, или путем перетаскивания элемента из библиотеки в дерево системы.

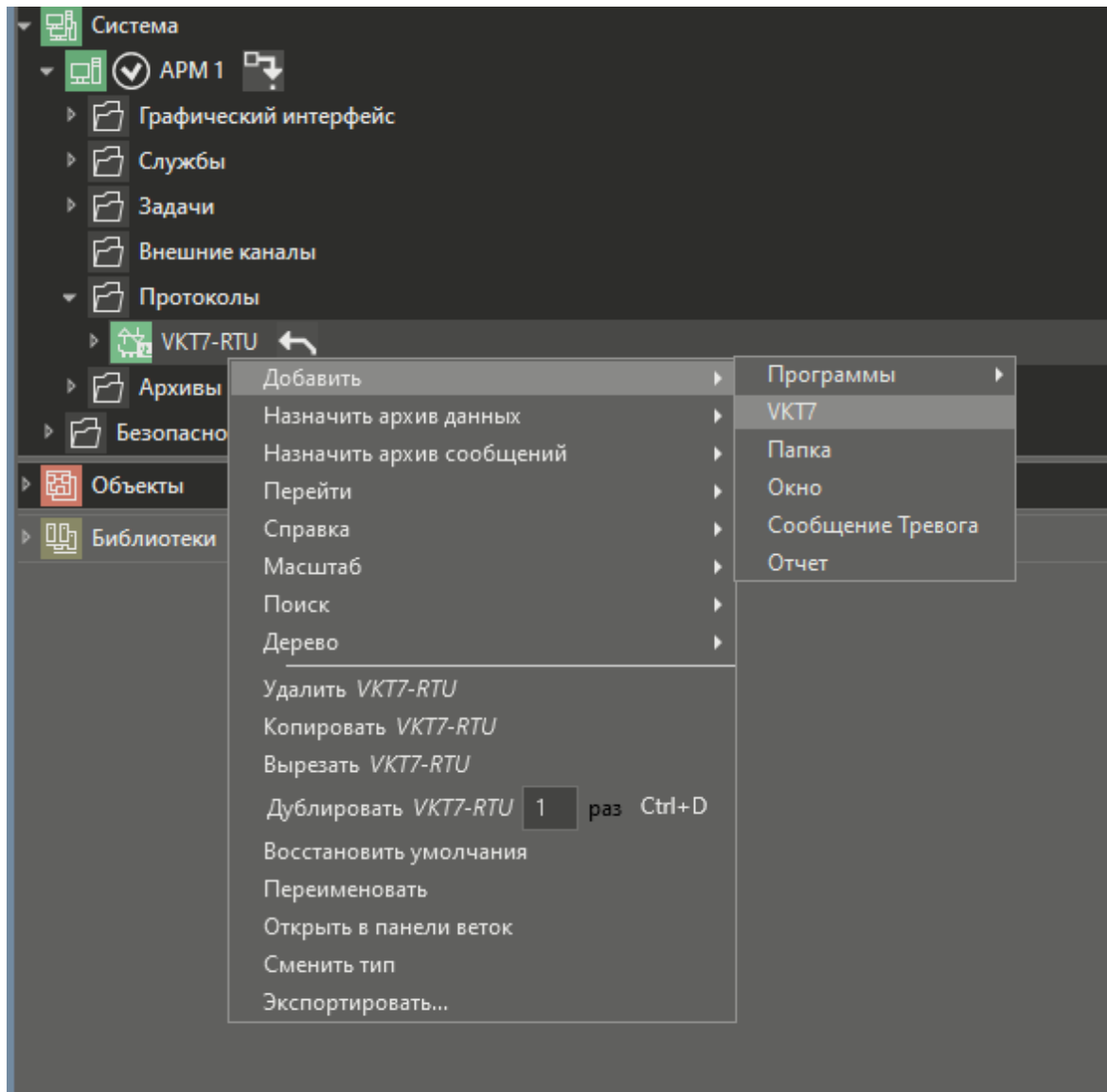
Рекомендуется добавлять элементы в проект при помощи контекстного меню:



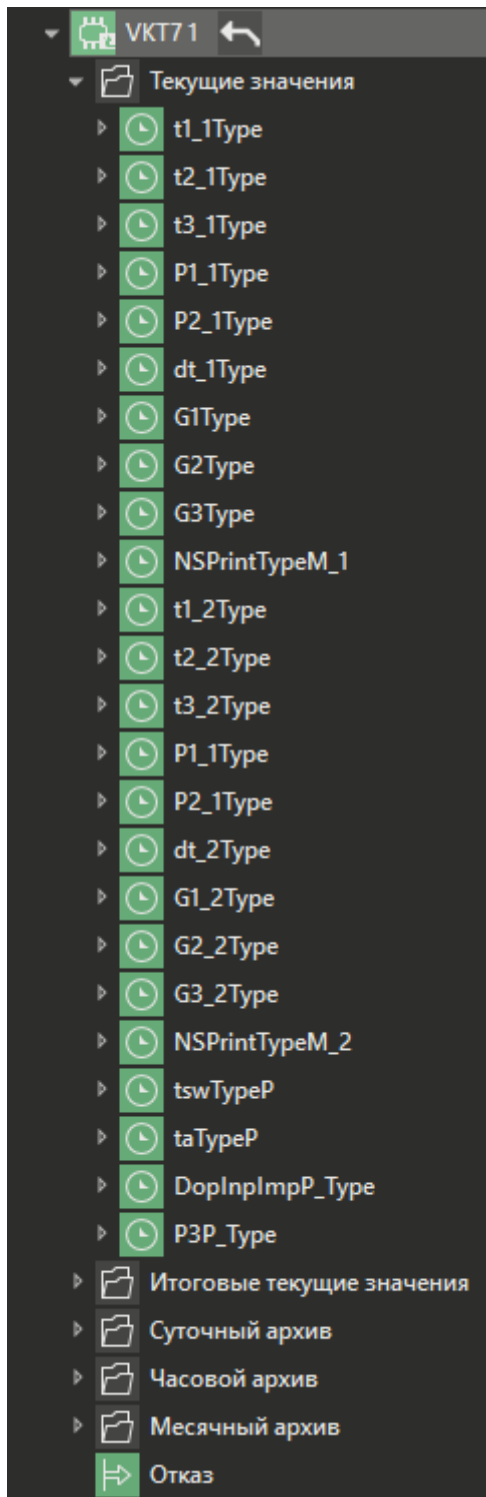
Тип добавляемого протокола зависит от реального способа подключения теплосчетчика:

- Modbus RTU-VKT7 - если счетчик подключен по последовательному интерфейсу. Настройки этого протокола соответствуют настройкам Modbus RTU.
- Modbus over TCP-VKT7 - если счетчик подключен через конвертер Ethernet-COM. Настройки этого протокола соответствуют настройкам Modbus RTU over TCP

После этого, в настроенный протокол добавляется элемент, соответствующий теплосчетчику VKT7:



В дереве системы элемент VKT7 имеет вид:



Элемент VKT7 имеет выход Отказ, а также группы каналов, согласно его максимальной конфигурации. Подробное описание смотрите в разделе Тепловычислитель VKT7.

Если в опросе каких-либо каналов нет необходимости, то можно либо удалить их из конфигурации элемента VKT7 уже после добавления его в дерево системы, либо сначала создать экземпляр элемента VKT7 в пользовательской библиотеке, затем из его конфигурации удалить незадействованные каналы, после чего использовать в проекте уже новый видоизмененный библиотечный элемент. .

Для корректной работы добавленного элемента необходимо настроить его панель свойств.

СВОЙСТВА VKT7

Панель свойств элемента VKT7 имеет вид:

The screenshot shows the 'VKT7 1' properties panel. It is divided into several sections:

- Общие (General):**
 - Имя (Name): VKT7 1
 - Метки (Tags):
 - Комментарий (Comment):
- Служебные (Service):**
 - Задача (Task):
 - Масштабирование (Scaling):
 - Настройки (Settings):
 - Адрес (Address): 0
 - Глубина чтения часового архива (Hourly archive depth): 20
 - Глубина чтения суточного архива (Daily archive depth): 20
 - Глубина чтения месячного архива (Monthly archive depth): 5
 - Число записей архива за раз (Number of records per batch): 5

Название	Описание
Категория Общие	Как правило, для данного элемента эта категория не требует настройки. Подробное описание данной категории приведено в соответствующем разделе.
Категория Задача	Задаёт способ опроса устройства. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи.
Категория Масштабирование	Если модуль ввода-вывода не имеет встроенного масштабирования, то при помощи данной категории может быть задано соответствие между значениями в модуле и реальными значениями. Подробное описание данной категории приведено в соответствующем разделе.

Название	Описание
Категория Настройки	
Адрес	Указывает адрес счетчика в сети.
Глубина чтения часового архива	Задает количество записей часового архива, которое будет получено при первом подключении к прибору, когда в базе данных исполнительной системы еще отсутствуют записи от этого прибора. Всего в приборе может быть сохранено 1152 записей.
Глубина чтения суточного архива	Задает количество записей суточного архива, которое будет получено при первом подключении к прибору, когда в базе данных исполнительной системы еще отсутствуют записи от этого прибора. Всего в приборе может быть сохранено 128 записей.
Глубина чтения месячного архива	Задает количество записей месячного архива, которое будет получено при первом подключении к прибору, когда в базе данных исполнительной системы еще отсутствуют записи от этого прибора. Всего в приборе может быть сохранено 32 записей.
Число записей архива за раз	<p>Задает максимальное количество архивных значений, считываемых из прибора за один цикл опроса.</p> <p>При первом подключении исполнительной системы к прибору, когда в базе данных исполнительной системы еще отсутствуют записи от этого прибора, из него будут считаны накопленные архивные значения за интервал заданный в свойствах типа Глубина чтения, описанные выше. А в том случае, если в базе данных уже содержатся записи от этого прибора, то исполнительной системой будут считаны архивные значения, накопленные в приборе со времени последней записи в базе данных по текущий момент.</p> <p>На каждом цикле опроса прибор будет отправлять количество значений, не превышающее указанное в данном свойстве.</p>

Название	Описание
	После того, как из прибора будут считаны и сохранены все ранее накопленные архивные значения, исполнительная система станет получать из него архивные значения только по мере их формирования.

ТЕПЛОВЫЧИСЛИТЕЛЬ VKT7

Тепловычислитель ВКТ-7 архивирует 1152 часовых, 128 суточных и 32 месячные записи и итоговые показания результатов измерений и диагностики параметров теплоснабжения.

Элемент, соответствующий реальному тепловычислителю VKT7, имеет выход Отказ, а также группы каналов, согласно его максимальной конфигурации.

Состав группы Текущие значения:

Название канала	Назначение	Название канала	Назначение
t1_1Type	Температура воды в подающем трубопроводе Тв 1	t3_2Type	Температура воды 3 Тв2
t2_1Type	Температура воды в обратном трубопроводе Тв 1	P1_2Type	Избыточное давление в подающем трубопроводе Тв2
t3_1Type	Температура воды 3 Тв1	P2_2Type	Избыточное давление в обратном трубопроводе Тв2
P1_1Type	Избыточное давление в подающем трубопроводе Тв1	dt_12Type	Разность температур воды t1-t2 Тв2
P2_1Type	Избыточное давление в обратном трубопроводе Тв1	G1_2Type	Объемный расход в обратном трубопроводе Тв2

Название канала	Назначение	Название канала	Назначение
dt_1Type	Разность температур воды t1-t2 Тв1	G2_2Type	Объемный расход в обратном трубопроводе Тв2
G1Type	Объемный расход в обратном трубопроводе Тв1	G3_2Type	Объемный расход в трубопроводе 3 Тв2
G2Type	Объемный расход в обратном трубопроводе Тв1	NSPrintTypeM_2	Наличие нештатной ситуации по Тв2
G3Type	Объемный расход в трубопроводе 3 Тв1	tswTypeP	Температура холодной воды
NSPrintTypeM_1	Наличие нештатной ситуации по Тв1	taTypeP	Температура воздуха
t1_2Type	Температура воды в подающем трубопроводе Тв 2	DopImpImpP_Type	Счетный вход
t2_2Type	Температура воды в обратном трубопроводе Тв 2	P3P_Type	Избыточное давление 3

Состав группы Итоговые текущие значения:

Название канала	Назначение	Название канала	Назначение
V1_1Type	Объем воды в подающем трубопроводе Тв1	V2_2Type	Объем воды в обратном трубопроводе Тв2
V2_1Type	Объем воды в обратном трубопроводе Тв1	V3_2Type	Объем воды 3 Тв2

Название канала	Назначение	Название канала	Назначение
V3_1Type	Объем воды 3 Тв1	M1_2Type	Масса воды в подающем трубопроводе Тв2
M1_1Type	Масса воды в подающем трубопроводе Тв1	M2_2Type	Масса воды в обратном трубопроводе Тв2
M2_1Type	Масса воды в обратном трубопроводе Тв1	M3_2Type	Масса воды 3 Тв2
M3_1Type	Масса воды 3 Тв1	Mg_2Type	Масса воды, отобранной из системы Тв2
Mg_1Type	Масса воды, отобранной из системы Тв1	Qo_2Type	Тепловая энергия полная Тв2
Qo_1Type	Тепловая энергия полная Тв1	Qg_2Type	Тепловая энергия в трубопроводе 3 Тв2
Qg_1Type	Тепловая энергия в трубопроводе 3 Тв1	Qnt_2TypeНІ Р	Время нормальной работы Тв2
Qnt_1TypeНІР	Время нормальной работы Тв1	Qnt_2TypeР	Время отсутствия счета Тв2
Qnt_1TypeР	Время отсутствия счета Тв1	DopInpImpP_ Type	Счетный вход
V1_2Type	Объем воды в подающем трубопроводе Тв2		

Состав групп Суточный (часовой, месячный) архив:

Название канала	Назначение	Название канала	Назначение
t1_1Type	Температура воды в подающем трубопроводе Тв 1	tswTypeP	Температура холодной воды
t2_1Type	Температура воды в обратном трубопроводе Тв 1	taTypeP	Температура воздуха
t3_1Type	Температура воды 3 Тв1	DopImpImpP_Type	Счетный вход
P1_1Type	Давление в подающем трубопроводе Тв1	P3P_Type	Давление 3
P2_1Type	Давление в обратном трубопроводе Тв1	M1_1Type	Масса воды в подающем трубопроводе Тв1
dt_1Type	Разность температур воды t1-t2 Тв1	M2_1Type	Масса воды в обратном трубопроводе Тв1
V1Type	Объем воды в подающем трубопроводе Тв1	M3_1Type	Масса воды 3 Тв1
V2Type	Объем воды в обратном трубопроводе Тв1	Mg_1Type	Масса воды, отобранной из системы Тв1
V3Type	Объем воды 3 Тв1	Qo_1Type	Тепловая энергия полная Тв1
NSPrintTypeM_1	Наличие нештатной ситуации по Тв1	Qg_1Type	Тепловая энергия в трубопроводе 3 Тв1
t1_2Type	Температура воды в подающем трубопроводе Тв 2	Qnt_1TypeNIP	Время нормальной работы Тв1
t2_2Type	Температура воды в обратном трубопроводе Тв 2	Qnt_1TypeP	Время отсутствия счета Тв1

Название канала	Назначение	Название канала	Назначение
t3_2Type	Температура воды 3 Тв2	QntNS_1	Длительность нештатной ситуации по параметрам Тв1
P1_2Type	Давление в подающем трубопроводе Тв2	M1_2Type	Масса воды в подающем трубопроводе Тв2
P2_2Type	Давление в обратном трубопроводе Тв2	M2_2Type	Масса воды в обратном трубопроводе Тв2
dt_12Type	Разность температур воды t1-t2 Тв2	M3_2Type	Масса воды 3 Тв2
V1_2Type	Объем воды в подающем трубопроводе Тв2	Mg_2Type	Масса воды, отобранной из системы Тв2
V2_2Type	Объем воды в обратном трубопроводе Тв2	Qo_2Type	Тепловая энергия полная Тв2
V3_2Type	Объем воды 3 Тв2	Qg_2Type	Тепловая энергия в трубопроводе 3 Тв2
NSPrintTypeM_2	Наличие нештатной ситуации по Тв2	Qnt_2TypeH IP	Время нормальной работы Тв2
tswTypeP	Температура холодной воды	Qnt_2TypeP	Время отсутствия счета Тв2

Каналы содержат типовые параметры.

В зависимости от статусов различных регистров исполнительная система формирует дополнительные признаки качества параметров каналов

OpсUa_Bad - формируется если от прибора не был получен ответ.

OpсUa_BadSensorFailure - формируется, если невозможно прочитать значение или, если у поступившего сигнала обнаружена нештатная ситуация. Как правило означает, что датчик отключен или не исправлен.

OpсUa_BadNoCommunication - формируется, если исполнительная система запрашивает значение сигнала, а устройство сигнализирует об отсутствии сигнала среди активных тегов для опроса. Как правило это означает, что канал есть в конфигурации, но переменная отсутствуют в приборе.

OpсUa_UncertainSensorNotAccurate - формируется, если прибор вернул, что значение запрошенного элемента находится вне диапазона (в приборе в данном случае на месте числового значения индицируется надпись Вне диапазона).

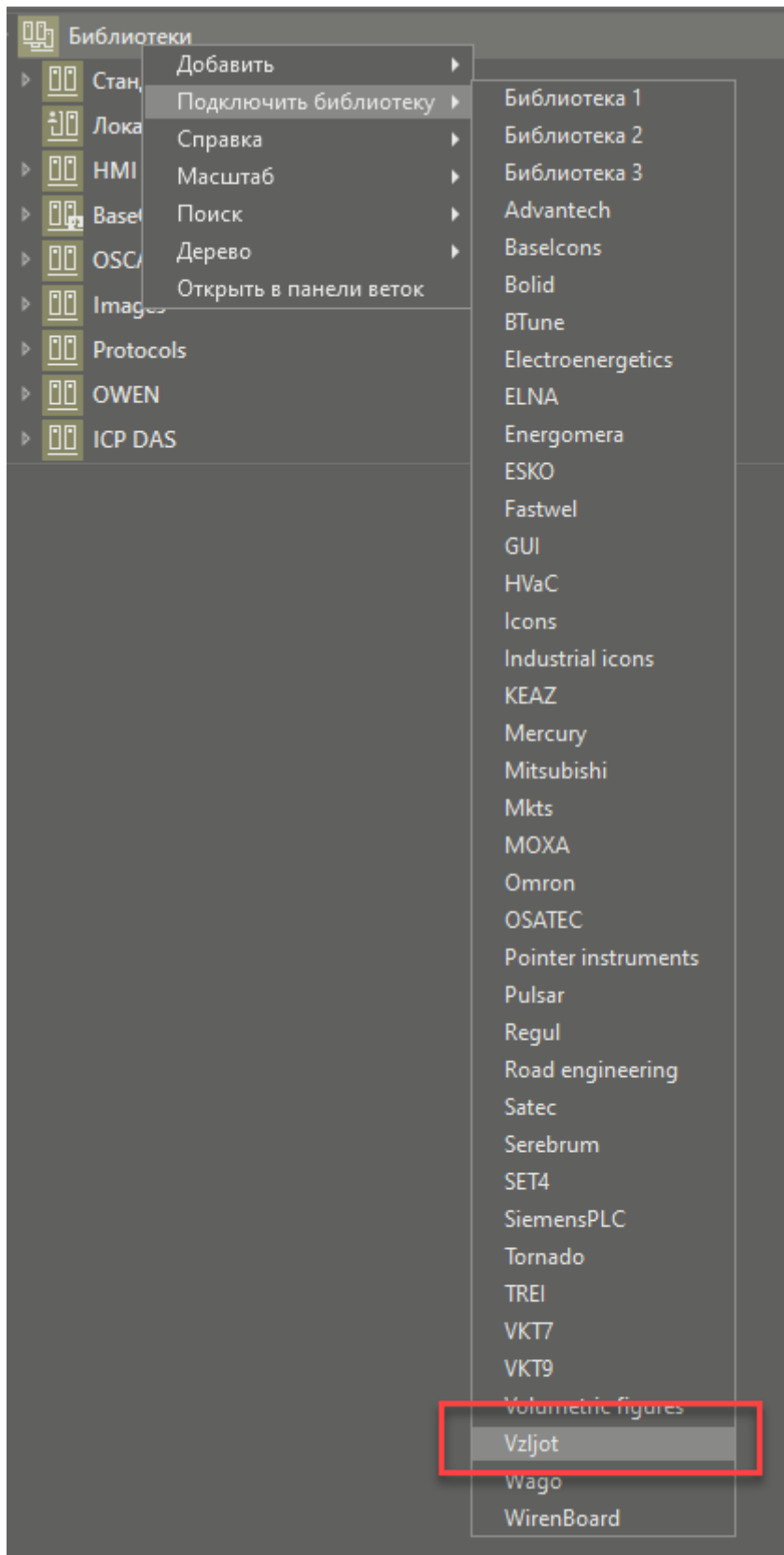
6.2.3.2.16.VZLJOT

MasterSCADA 4D поддерживает работу с приборами ГК Взлет. Поддерживаются приборы: ТСР-034(033), ТСР-026М,ТСР-043, ТСР-032, ЭР, ТСР-024М, ТСР-042

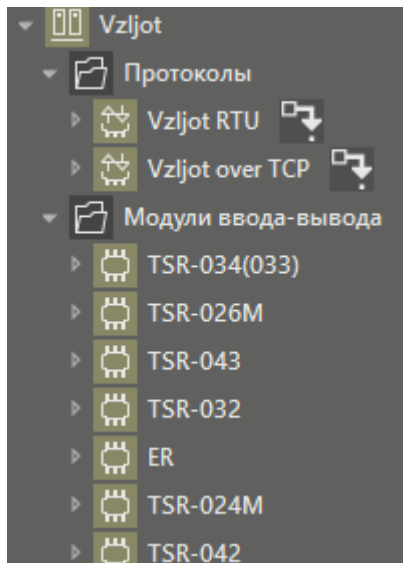
При обмене узел играет роль ведущего. Поддерживается считывание архивов из счетчика. Работа с этими устройствами возможна через последовательные интерфейсы (RS-232, RS-485, GSM), а при использовании конвертеров Ethernet-COM (Моха NPort 5150 и т.п.) также и по сетям TCP/IP (Ethernet, Wi-Fi).

Важно! По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте sales@masterscada.ru

Для получения возможности использования этих счетчиков в проекте необходимо подключить соответствующую библиотеку:

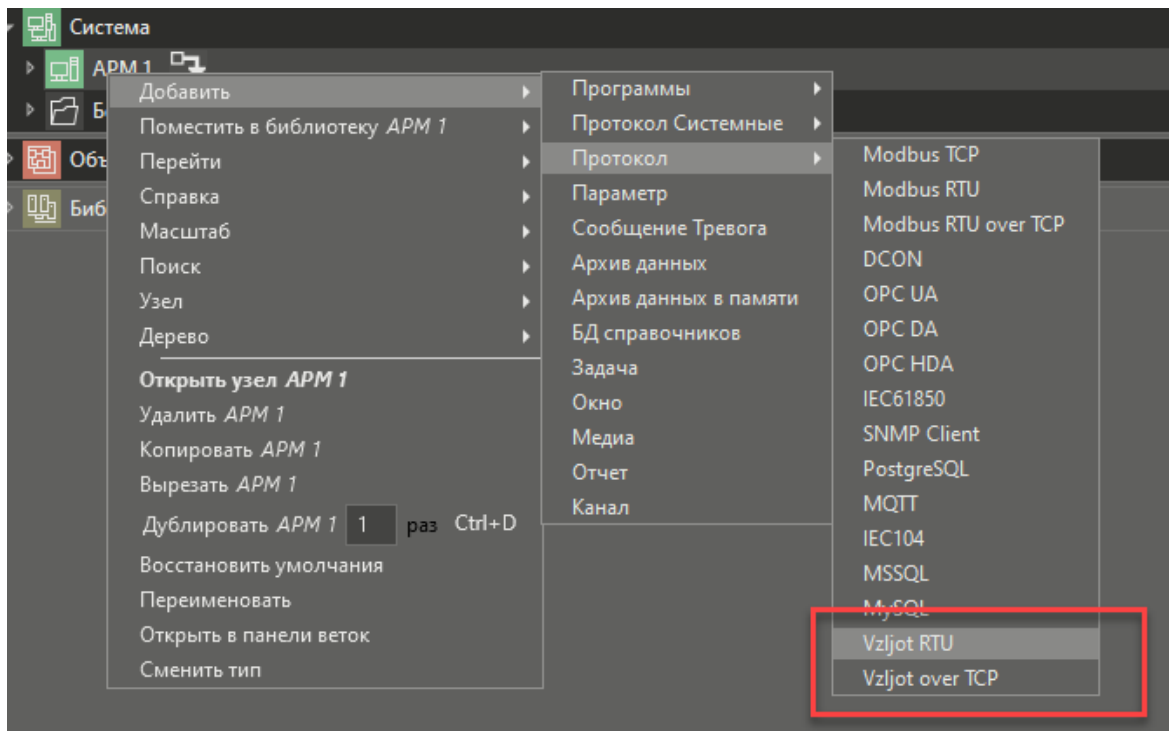


В полном дереве библиотека Vzljot имеет вид:



После этого в проект могут быть добавлены сначала протокол, а затем устройство, любым известным способом, например, при помощи контекстного меню узла, или с использованием контекстной панели, или путем перетаскивания элемента из библиотеки в дерево системы.

Рекомендуется добавлять элементы в проект при помощи контекстного меню:

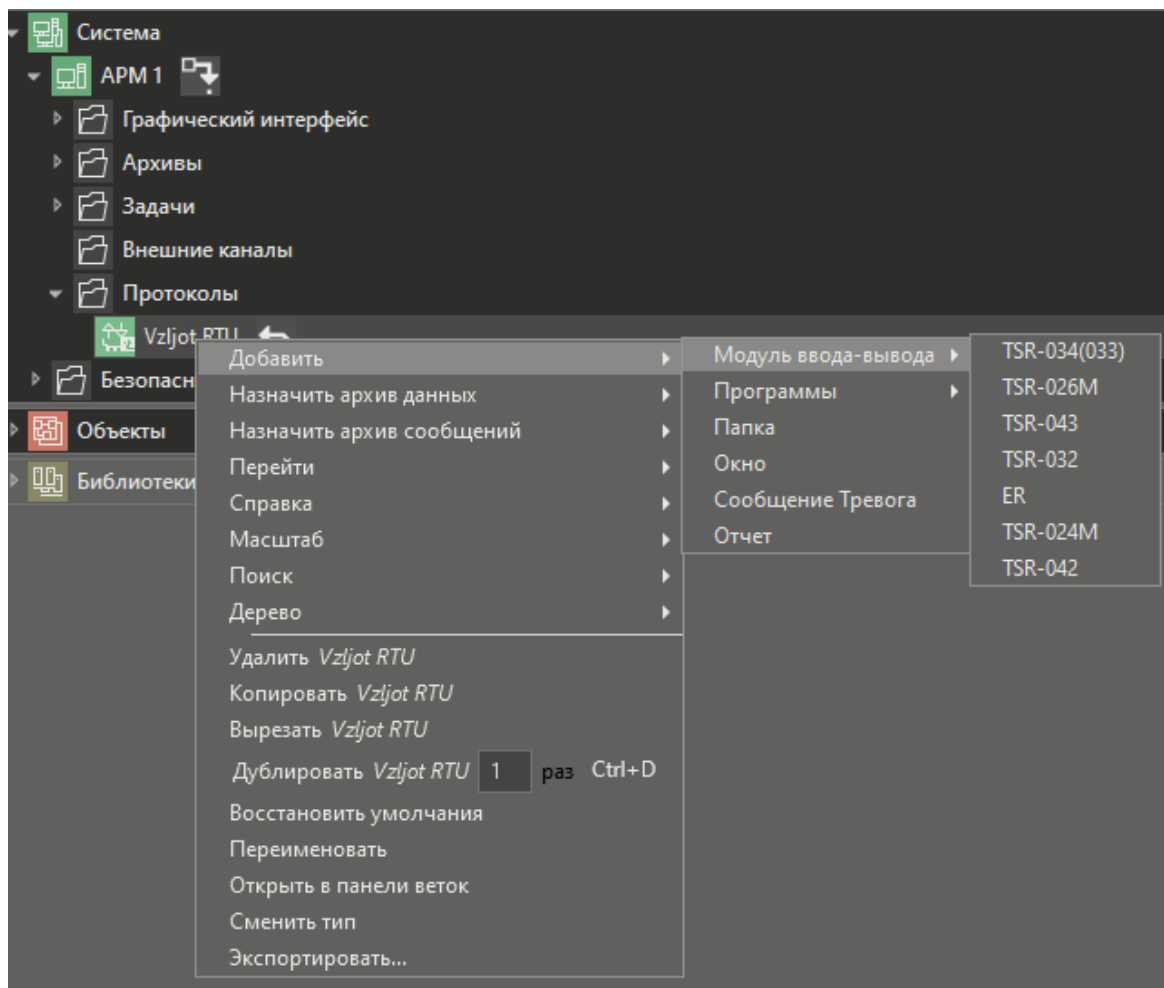


Тип добавляемого протокола зависит от реального способа подключения счетчика:

- Vzljot RTU - если счетчик подключен по последовательному интерфейсу. Настройки этого протокола практически полностью соответствуют настройкам Modbus RTU.

- Vzljot over TCP - если счетчик подключен через конвертер Ethernet-COM. Настройки этого протокола соответствуют настройкам Modbus RTU over TCP

Затем при помощи контекстного меню протокола можно добавить модули ввода-вывода, которые уже содержат все необходимые каналы:



Каждый счетчик имеет predetermined набор каналов, который характерен для него. Поддержаны наиболее используемые функции счетчиков.

Описание групп приборов:

Название	Назначение
Группа Current	Содержит набор групп с текущими измеренными значениями прибора
Группа Total	Группа содержит каналы для получения накопленных значений, которые хранятся в приборе.

Название	Назначение
Группы HeatSystem1 - 4	Содержат набор параметров теплосистем 1-4.
Группа ArchiveHour/ArchiveHourHeatSystem	Группа содержит каналы для получения архивных часовых значений. Группа содержит набор каналов, по количеству каналов прибора.
Группа ArchiveDay/ArchiveDayHeatSystem	Группа содержит каналы для получения архивных суточных значений. Группа содержит набор каналов, по количеству каналов прибора.
Группа ArchiveMonth/ArchiveMonthHeatSystem	Группа содержит каналы для получения архивных месячных значений. Группа содержит набор каналов, по количеству каналов прибора.

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если у архивируемого параметра снять флаг Архивировать, то этот канал выключится из опроса. Если снять флаг Архивировать у параметра Pointer, то вся группа выключится из опроса

Поддерживаются следующие модули ввода-вывода:

- TSR-034(033)
- TSR-026M
- TSR-043
- TSR-032
- ER
- TSR-024M
- TSR-042

После добавления в дерево проекта модулей ввода-вывода, настраивается панель свойств модулей.

Каждый модуль ввода-вывода имеет выход Отказ, а также группы каналов, согласно его максимальной конфигурации.

Если в опросе каких-либо каналов нет необходимости, то можно либо удалить их из конфигурации элемента уже после добавления его в дерево системы, либо сначала создать экземпляр модуля в пользовательской библиотеке, затем из его конфигурации удалить незадействованные каналы, после чего использовать в проекте уже новый видоизмененный библиотечный элемент.

6.2.3.2.16.1. СВОЙСТВА МОДУЛЕЙ VZLJOT

Вид панели свойств модуля ввода-вывода Vzljot

TSR-026M 1	
Общие	
Имя	TSR-026M 1
Полное имя	Система.АРМ 1.Протоколы.Vzljot RTU.TSR-026M 1
Метки	
Комментарий	Взлет TCP-026(M)
Службные	
Задача	
Отношения	
Масштабирование	
Настройки	
Адрес	1
Глубина считывания часового архива	1504
Глубина считывания суточного архива	512
Глубина считывания месячного архива	32
Количество считываемых записей архива за цикл	5

Описание:

Название	Описание
Категория Общие	Как правило, для данного элемента эта категория не настраивается. Подробное описание данной категории приведено в соответствующем разделе.
Категория Задача	Задается способ опроса устройства. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи.
Категория Масштабирование	Если модуль ввода-вывода не имеет встроенного масштабирования, то при помощи данной категории можно задать соответствие между значениями в модуле и реальными значениями. Подробное описание данной категории приведено в соответствующем разделе.
Категория Настройки	

Название	Описание
Адрес	Задается адрес устройства в сети (значение адреса определяется документацией на подключаемое устройство).
Глубина считывания часового архива	Определяет количество считываемых при старте исполнительной системы записей часового архива. По умолчанию задано максимальное значение
Глубина считывания суточного архива	Определяет количество считываемых при старте исполнительной системы записей суточного архива. По умолчанию задано максимальное значение
Глубина считывания месячного архива	Определяет количество считываемых при старте исполнительной системы записей месячного архива. По умолчанию задано максимальное значение
Количество считываемых записей архивов за цикл	Определяет количество считываемых каждый цикл опроса записей из архива счетчика

6.2.3.2.16.2. TSR-034(033)

Счетчики TSR-034(033) (название производителя TCP-034, TCP-033) предназначен для измерения, индикации, регистрации параметров теплоносителя и тепловой энергии, а также других параметров в теплосистемах различного типа, обеспечивает учет в одной теплосистеме при общем количестве трубопроводов до трех, энергонезависимый.

Вид счетчика добавленного в протокол в дереве системы:



Состав каналов счетчика:

Название	Назначение
CurrentTime	Текущее время прибора
Группа Current.	
T1-T3	Параметры температуры в каждом канале, С. Показания датчиков температуры ПТ1-ПТ3.
G1-G3	Параметры массового расхода в каждом канале, т/ч. Показания вычисленного массового расхода датчиками ПР1-ПР3
F1-F3	Параметры объемного расхода в каждом канале, м3/ч. Показания расходомеров ПР1-ПР3

Название	Назначение
E1-E6	Тепловая мощность б, ГДж/ч
P1-P3	Давление договорное в каждом канале, МПа
Pcw	Давление холодной воды, МПа
Tcw	Температура холодной воды, МПа
Группа Total	
W1-W6	Энергия, количество теплоты в отдельных ветвях теплосистемы, Мдж
M1-M3	Значения массы теплоносителя в каждом канале, кг
V1-V3	Значения объема теплоносителя в каждом канале, м3
Подгруппы ArchiveHour, ArchiveDay, ArchiveMonth группы Archive	
W4-W6	Архив тепла в трубопроводе. Значения W1-W3 не архивируются в приборе.
M1-M3	Архив массы
T1-T3	Архив температуры

Каждый модуль ввода-вывода имеет выход Отказ

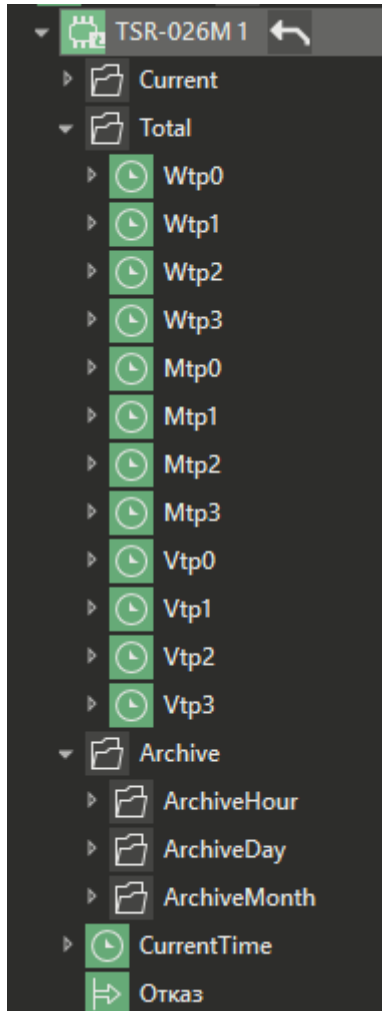
Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если у архивируемого параметра снять флаг Архивировать, то этот канал выключится из опроса.

6.2.3.2.16.3. TSR-026M

Счетчики TSR-026 (название производителя TCP-026M) предназначен для измерения, индикации, регистрации параметров теплоносителя и тепловой энергии, а также других параметров в теплосистемах различного типа, обеспечивает учет в одной теплосистеме при

общем количестве трубопроводов до четырех, электропитание =24В, встроенная аккумуляторная батарея.

Вид счетчика добавленного в протокол в дереве системы:



Состав каналов счетчика:

Название	Назначение
CurrentTime	Текущее время прибора
Группа Current.	
T1-T5	Параметры температуры в каждом канале, С. Показания датчиков температуры ПТ1-ПТ5.
P1-P4	Давление в каждом канале, МПа. Показания датчиков давления ПД1-ПД4.

Название	Назначение
Fv1-Fv4	Параметры объемного расхода в каждом канале, м ³ /ч. Показания расходомеров ПР1-ПР4
FvLog1/FvLog2	Объемный расход в канале ЛОГ1/ЛОГ2 , м ³ /ч
Gtp0-Gtp3	Массовый расход, т/ч
Etp0-Etp3	Тепловая мощность 0, Гкал/ч
Tcw	Температура холодной воды,С
Группа Total	
Wp0-Wp3	Тепло в рабочем режиме, Гкал
Mtp0-Mtp3	Масса в рабочем режиме, т
Vpt0-Vpt3	Объем в рабочем режиме, м ³
Подгруппы ArchiveHour, ArchiveDay, ArchiveMonth группы Archive	
Qs	Общее тепло, Гкал
Ms	Общая масса, т
Qgvs	Тепло с ГВС, Гкал
Mgvs	Масса с ГВС, т
Q1 - Q4	Тепло по каждому трубопроводу, Гкал
M1-M4	Масса по трубопроводу 1, т
V1-V4	Объем по трубопроводу 1, м ³

Название	Назначение
Taverage1-Taverage4	Средневзвешенная температура в трубопроводе, С
T1-T4	Средняя температура в трубопроводе 1, С
P1-P4	Средняя давление в трубопроводе 1, МПа
Tcw	Температура холодной воды, С
Pcw	Давление холодной воды, МПа
Vcw	Объем холодной воды, м3
Toutdoor	Температура наружного воздуха, С

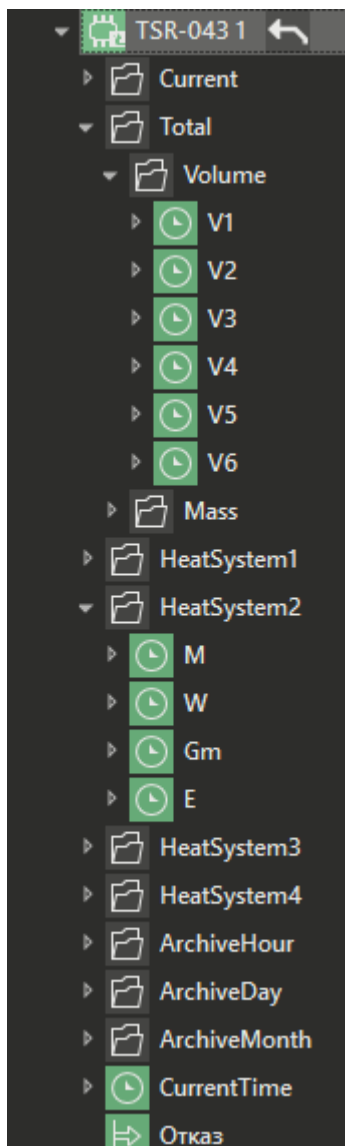
Каждый модуль ввода-вывода имеет выход Отказ

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если у архивируемого параметра снять флаг Архивировать, то этот канал выключится из опроса.

6.2.3.2.16.4. TSR-043

Теплосчетчик – регистратор ТСП-043 используется для объектов для абонентского учета тепла с точностью, регламентируемой для коммерческого учета.

Вид счетчика добавленного в протокол в дереве системы:



Состав каналов счетчика:

Название	Назначение
CurrentTime	Текущее время прибора
Группа Current.Temperature	
T1-T5	Параметры температуры в каждом канале, С. Показания датчиков температуры ПТ1-ПТ5.
Группа Current.Pressure	

Название	Назначение
P1-P4	Параметры давления в каждом канале, Мпа. Показания датчиков давлений ПД1-ПД4.
Pcw	Давление холодной воды, МПа
Группа Current.FlowV	
G1-G6	Параметры объемного расхода в каждом канале, м3/ч. Показания расходомеров ПР1-ПР6
Группа Current.FlowM	
Gm1-Gm6	Параметры объемного расхода в каждом канале, т/ч. Показания вычисленного массового расхода датчиками ПР1-ПР6
Группа Total.Volume	
V1-V6	Значения накопленного объема в каждом канале, л. Накопленный объем в каналах ПР1-ПР6
Группа Total.Mass	
M1-M6	Значения накопленной массы в каждом канале, кг
Группы HeatSystem1-4	
M	Накопленная масса, т
W	Тепловая энергия, Мдж
Gm	Массовый расход, т/ч
E	Тепловая мощность, ГДж/ч
Группы ArchiveHour, ArchiveDay, ArchiveMonth	

Название	Назначение
M plus Pipe 1 - 3	Масса в прямом направлении в трубопроводе 1-3
M minus Pipe 1 - 3	Масса в обратном направлении в трубопроводе 1-3
V Pipe 1 - 3	Объем каждого трубопровода
Leak	Масса утечки
Mix	Масса подмеса
Temperature forward Pipe 1 - 3	Температура прямого трубопровода 1-3
Temperature reverse Pipe 1 - 3	Температура обратного трубопровода 1-3
Temperature Tcw	Температура холодной воды
Temperature Tout	Температура атмосферы
Pressure Pipe 1 - 3	Давление в трубопроводе 1-3
Pressure Pcw	Давление холодной воды
Pressure Pout	Давление атмосферы
Q	Интегратор тепла
Pointer	Указатель записи (сервисный тег). Если снять флаг архивировать у параметра данного канала, то вся группа будет исключена из опроса.

Каждый модуль ввода-вывода имеет выход Отказ

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если у архивируемого параметра снять флаг Архивировать, то этот канал выключится из опроса. Если снять флаг Архивировать у параметра Pointer, то вся группа выключится из опроса

6.2.3.2.16.5. TSR-032

Счетчики TSR-032 (название производителя TCP-032)) предназначен для измерения, индикации, регистрации параметров теплоносителя и тепловой энергии в теплосистемах различного типа, конфигурации и назначения, а также других вспомогательных параметров на узлах учета.

Вид счетчика добавленного в протокол в дереве системы:



Состав каналов счетчика:

Название	Назначение
CurrentTime	Текущее время прибора
Группа Current.	

Название	Назначение
T1-T3	Параметры температуры в каждом канале, С. Показания датчиков температуры ПТ1-ПТ3.
G1-G3	Параметры массового расхода в каждом канале, т/ч. Показания вычисленного массового расхода датчиками ПР1-ПР3
F1-F3	Параметры объемного расхода в каждом канале, м3/ч. Показания расходомеров ПР1-ПР3
E1-E6	Тепловая мощность б, ГДж/ч
P1-P3	Давление договорное в каждом канале, МПа
Pcw	Давление холодной воды, МПа
Tcw	Температура холодной воды, МПа
Группа Total	
W1-W6	Энергия, количество теплоты в отдельных ветвях теплосистемы, Мдж
M1-M3	Значения массы теплоносителя в каждом канале, кг
V1-V3	Значения объема теплоносителя в каждом канале, м3
Подгруппы ArchiveHour, ArchiveDay, ArchiveMonth группы Archive	
W4-W6	Архив тепла в трубопроводе. Значения W1-W3 не архивируются в приборе.
M1-M3	Архив массы
T1-T3	Архив температуры

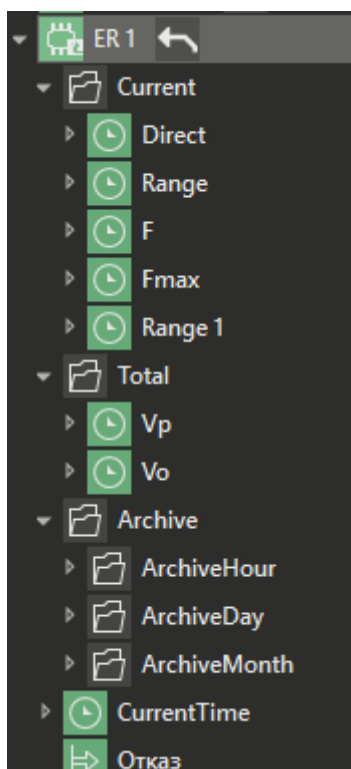
Каждый модуль ввода-вывода имеет выход Отказ

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если у архивируемого параметра снять флаг Архивировать, то этот канал выключится из опроса.

6.2.3.2.16.6. ER

Прибор ER (название производителя Взлет ЭР) предназначен для измерения расхода и объема горячей и холодной воды.

Вид счетчика добавленного в протокол в дереве системы:



Состав каналов счетчика:

Название	Назначение
CurrentTime	Текущее время прибора
Группа Current.	
Direct	Направление потока (0-прямой, 1-обратный)

Название	Назначение
Range	Текущий диапазон (0-№1 малый, 1 - №2 средний, 2-№3 большой)
F	Текущий расход, м3/ч
Fmax	Максимальный расход, м3/ч
Range 1	Регистр ошибок прибора
Группа Total	
Vp	Объем прямого потока, м3
Vo	Объем обратного потока, м3
Подгруппы ArchiveHour, ArchiveDay, ArchiveMonth группы Archive	
Vp	Объем в прямом направлении, м3
Vo	Объем в обратном направлении, м3

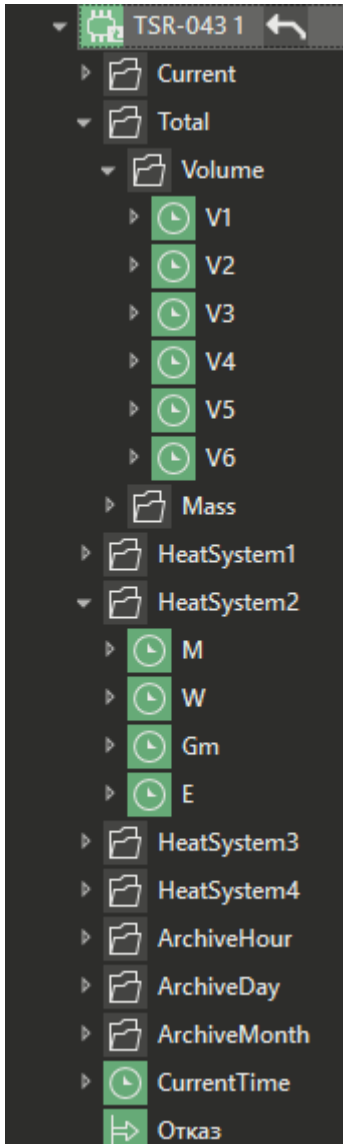
Каждый модуль ввода-вывода имеет выход Отказ

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если у архивируемого параметра снять флаг Архивировать, то этот канал выключится из опроса.

6.2.3.2.16.7. TSR-024M

Теплосчетчик – регистратор TSR-024M (название производителя TCP-024M) предназначен для измерения, индикации, регистрации параметров теплоносителя и тепловой энергии в теплосистемах различного типа, конфигурации и назначения, а также других вспомогательных параметров на узлах учета.

Вид счетчика добавленного в протокол в дереве системы:



Состав каналов счетчика:

Название	Назначение
CurrentTime	Текущее время прибора
Группа Current.Temperature	
T1-T9	Параметры температуры в канале, С. Показания датчиков ПТ1-ПТ6 и показания каналов t6-t8
Группа Current.Pressure	

Название	Назначение
P1-P6	Параметры давления в каждом канале, Мпа. Показания датчиков давлений ПД1-ПД6.
Группа Current.FlowV	
G1-G9	Параметры объемного расхода в каждом канале, м3/ч. Показания расходомеров ПР1-ПР9
Группа Current.FlowM	
Gm1-Gm6	Параметры объемного расхода в каждом канале, т/ч. Показания вычисленного массового расхода датчиками ПР1-ПР6
Группа Total.Volume	
V1-V9	Значения накопленного объема в каждом канале, л. Накопленный объем в каналах ПР1-ПР9
Группы HeatSystem1-3	
Wtc	Накопленная масса, т
Wgv	Тепловая энергия, Мдж
Mtc	Тепловая энергия, Мдж
Gtc	Массовый расход, т/ч
Etc	Тепловая мощность, ГКал/ч
Egv	Энергия ГВС, Гкал/ч
Группа Contract constant	Группа содержит значения договорных констант, где: Т - договорная температура, С

Название	Назначение
	<p>P - договорное давление, МПа</p> <p>Q - договорной расход, м3/ч</p> <p>E - договорная нагрузка по теплу, Гкал/ч</p>
<p>Группы ArchiveHourHeatSystem1 -3, ArchiveDayHeatSystem1 -3, ArchiveMonthHeatSystem1 -3</p>	
Wtc	Общее тепло потребленное абонентом.
Wgv	Тепло, отобранное с водоразбором
Mtc	Общая потребленная масса
Wtc1 - Wtc4	Тепло по трубопроводу 1-4
Mtc1- Mtc4	Масса по трубопроводу 1 - 4
Vtc1 - Vtc4	Объем по трубопроводу 1 - 4
Tavgw1 - Tavgw4	Средневзвешенная температура трубопровода 1-4
Tavg1 - Tavg4	Средняя температура трубопровода 1 - 4
P1-P4	Среднее давление трубопровода 1 - 4
Tcw	Температура холодной воды
Error	Набор флагов нештатных ситуаций.

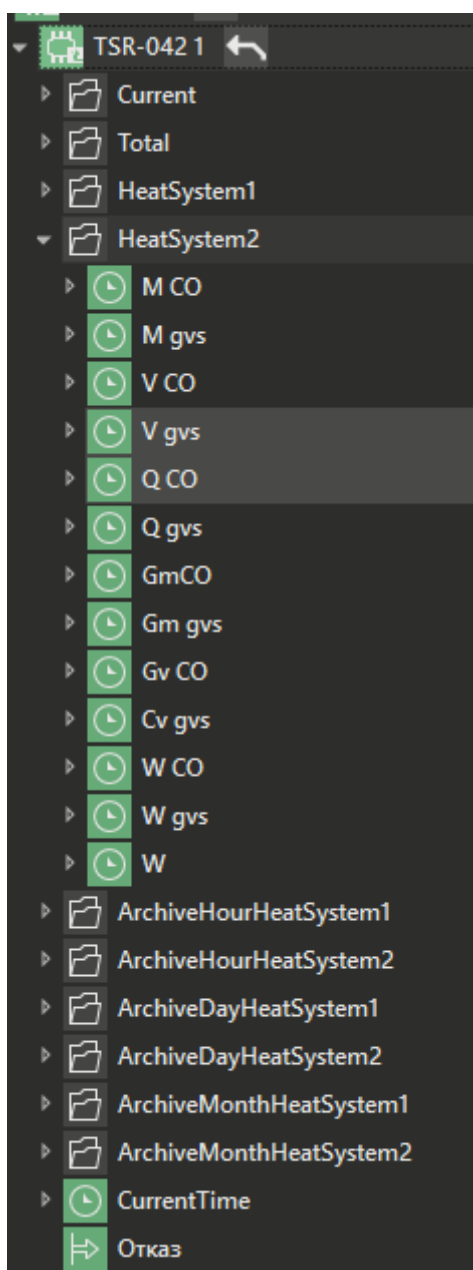
Каждый модуль ввода-вывода имеет выход Отказ

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если у архивируемого параметра снять флаг Архивировать, то этот канал выключится из опроса.

6.2.3.2.16.8. TSR-042

Теплосчетчик – регистратор TSR-042 (название производителя TCP-042) с резервным питанием и USB-интерфейсом. TCP-042 и TCP-042M полностью идентичны по структуре каналов, но имеют отличия в размере архива - версия TCP-042M имеет большее количество архивных записей. Определить тип прибора можно по версии прошивки - прибор с большим количеством записей имеет прошивку 66.00.20.00 и выше.

Вид счетчика добавленного в протокол в дереве системы:



Состав каналов счетчика:

Название	Назначение
CurrentTime	Текущее время прибора

Название	Назначение
Группа Current.Temperature	
T1-T4	Параметры температуры в канале, С. Показания датчиков ПТ1-ПТ4
Tcw	Температура холодной воды, С
Toutdoor	Температура наружного воздуха, С
Группа Current.Pressure	
P1-P4	Параметры давления в каждом канале, Мпа. Показания датчиков давлений ПД1-ПД4.
Группа Current.FlowV	
G1-G7	Параметры объемного расхода в каждом канале, м3/ч. Показания расходомеров ПР1-ПР7
Группа Current.FlowM	
Gm1-Gm4	Параметры объемного расхода в каждом канале, т/ч. Показания вычисленного массового расхода датчиками
Группа Total.Volume	
V1-V9	Значения накопленного объема в каждом канале, л. Накопленный объем в каналах ПР1-ПР9
Группа Total.Mass	
M1-M4	Накопленная масса в каналах 1-4
Группа Total.Energy	
W1-W4	Накопленная энергия в каналах 1-4

Название	Назначение
Группы HeatSystem1-3	
M CO	Накопленная масса CO, т
M gvs	Накопленная масса ГВС, т
V CO	Накопленный объем CO, м3
V gvs	Накопленный объем ГВС, м3
Q CO	Накопленная тепловая энергия CO, Гкал
Q gvs	Накопленная тепловая энергия ГВС, Гкал
GmCO	Массовый расход CO, т/ч
Gm gvs	Массовый расход ГВС, т/ч
Gv CO	Объемный расход CO, м3
Cv gvs	Объемный расход ГВС, м3
W CO	Энергия по CO, Гкал/ч
W gvs	Энергия по ГВС, Гкал/ч
W	Энергия по теплосистеме, Гкал/ч
Группы ArchiveHourHeatSystem1 -2, ArchiveDayHeatSystem1 -2, ArchiveMonthHeatSystem1 -2	
Wot	Энергия с нарастающим итогом по ОТ, Гкал
Wgvs	Энергия с нарастающим итогом по ГВС, Гкал

Название	Назначение
Wts	Энергия с нарастающим итогом по теплосистеме, Гкал
Mot	Масса с нарастающим итогом по ОТ, т
Mgvs	Масса с нарастающим итогом по ГВС, т
Vot	Объем с нарастающим итогом по ОТ, м3
Vgvs	Объем с нарастающим итогом по ГВС, м3
WotP	Энергия по ОТ в штатном режиме, Гкал
WgvsP	Энергия по ГВС в штатном режиме, Гкал
MotP	Масса по ОТ нарастающим итогом в штатном режиме, т
MgvsP	Масса по ГВС нарастающим итогом в штатном режиме, т
VotP	Объем по ОТ нарастающим итогом, м3
VgvsP	Объем по ГВС нарастающим итогом, м3
Vleak	Объем утечки, м3
Mleak	Масса утечки, т
Wleak	Энергия утечки, Гкал

Каждый модуль ввода-вывода имеет выход Отказ

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если у архивируемого параметра снять флаг Архивировать, то этот канал выключится из опроса.

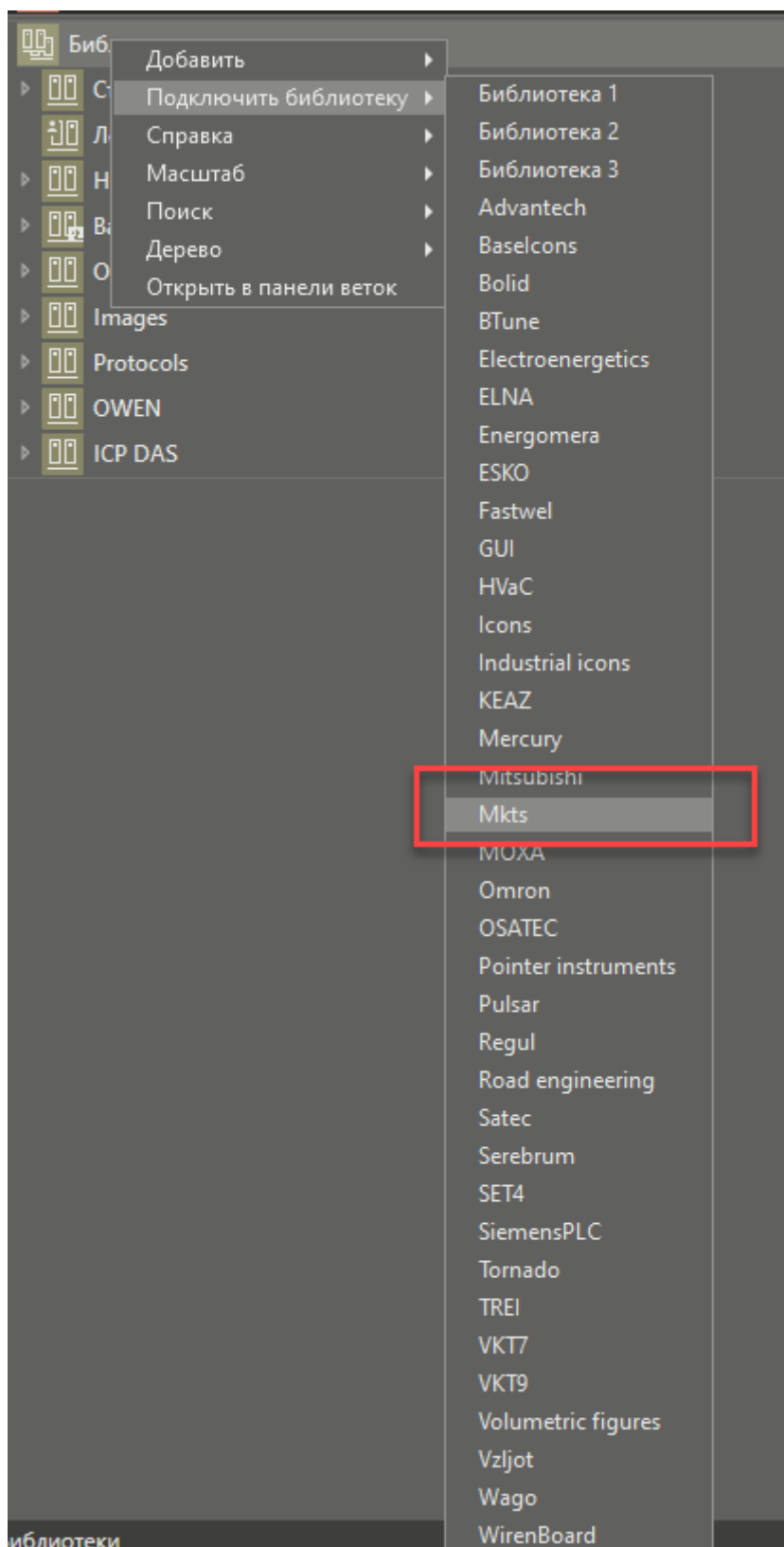
6.2.3.2.17.МКТС

MasterSCADA 4D поддерживает работу с приборами Интелприбор. Поддерживаются теплосчетчики МКТС

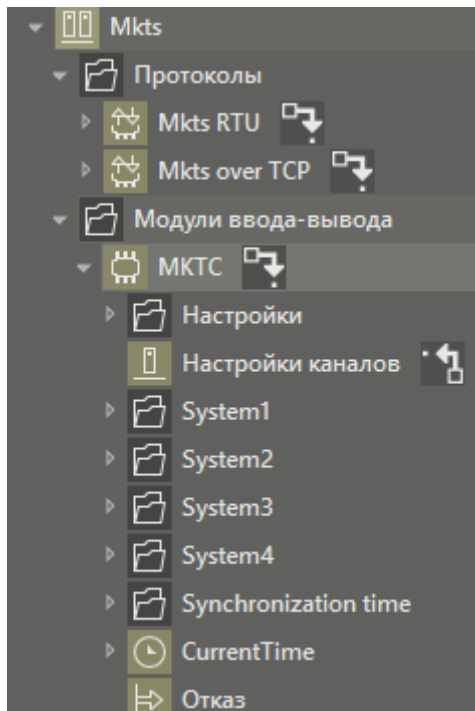
При обмене узел играет роль ведущего. Поддерживается считывание архивов из счетчика. Работа с этими устройствами возможна через последовательные интерфейсы (RS-232, RS-485, GSM), а при использовании конвертеров Ethernet-COM (Моха NPort 5150 и т.п.) также и по сетям TCP/IP (Ethernet, Wi-Fi).

Важно! По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте sales@masterscada.ru

Для получения возможности использования этих счетчиков в проекте необходимо подключить соответствующую библиотеку:

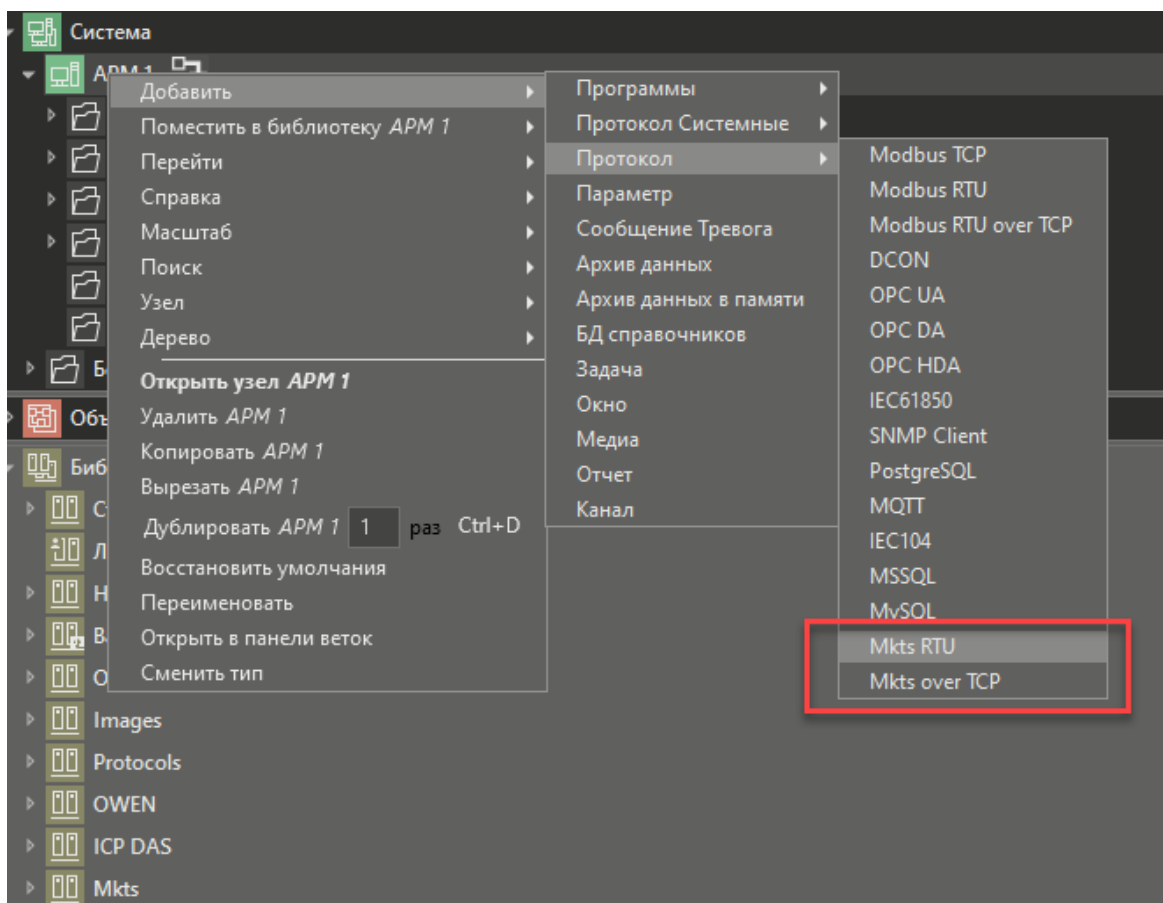


В полном дереве библиотека Vzljot имеет вид:



После этого в проект могут быть добавлены сначала протокол, а затем устройство, любым известным способом, например, при помощи контекстного меню узла, или с использованием контекстной панели, или путем перетаскивания элемента из библиотеки в дерево системы.

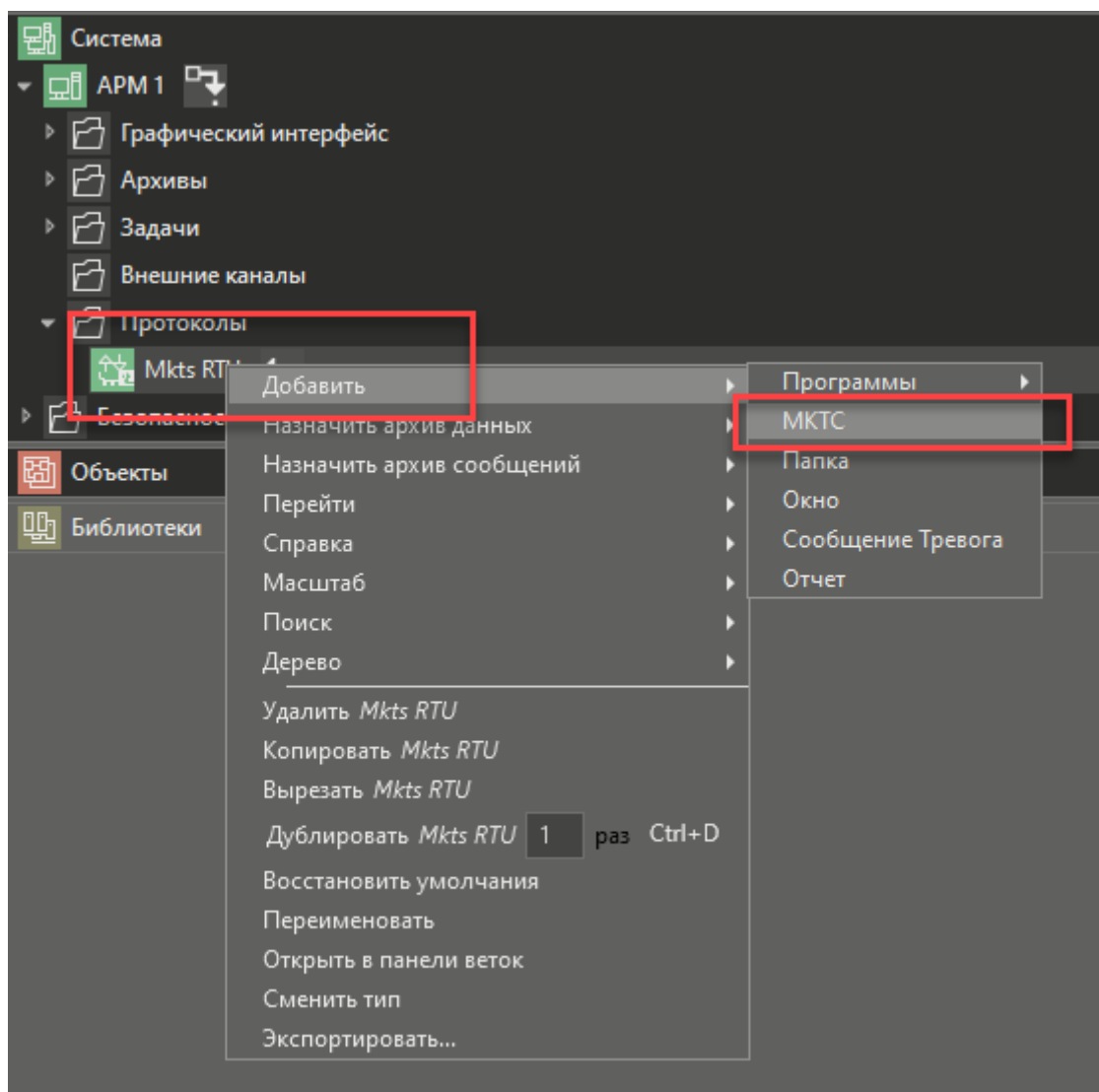
Рекомендуется добавлять элементы в проект при помощи контекстного меню:



Тип добавляемого протокола зависит от реального способа подключения счетчика:

- Mkts RTU - если счетчик подключен по последовательному интерфейсу. Настройки этого протокола практически полностью соответствуют настройкам Modbus RTU.
- Mkts over TCP - если счетчик подключен через конвертер Ethernet-COM. Настройки этого протокола соответствуют настройкам Modbus RTU over TCP

Затем при помощи контекстного меню протокола можно добавить модули ввода-вывода, которые уже содержат все необходимые каналы:



Каждый счетчик имеет predetermined набор каналов, который характерен для него. Поддержаны наиболее используемые функции счетчиков.

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если у архивируемого параметра снять флаг Архивировать, то этот канал выключится из опроса. Если снять флаг Архивировать у параметра Pointer, то вся группа выключится из опроса

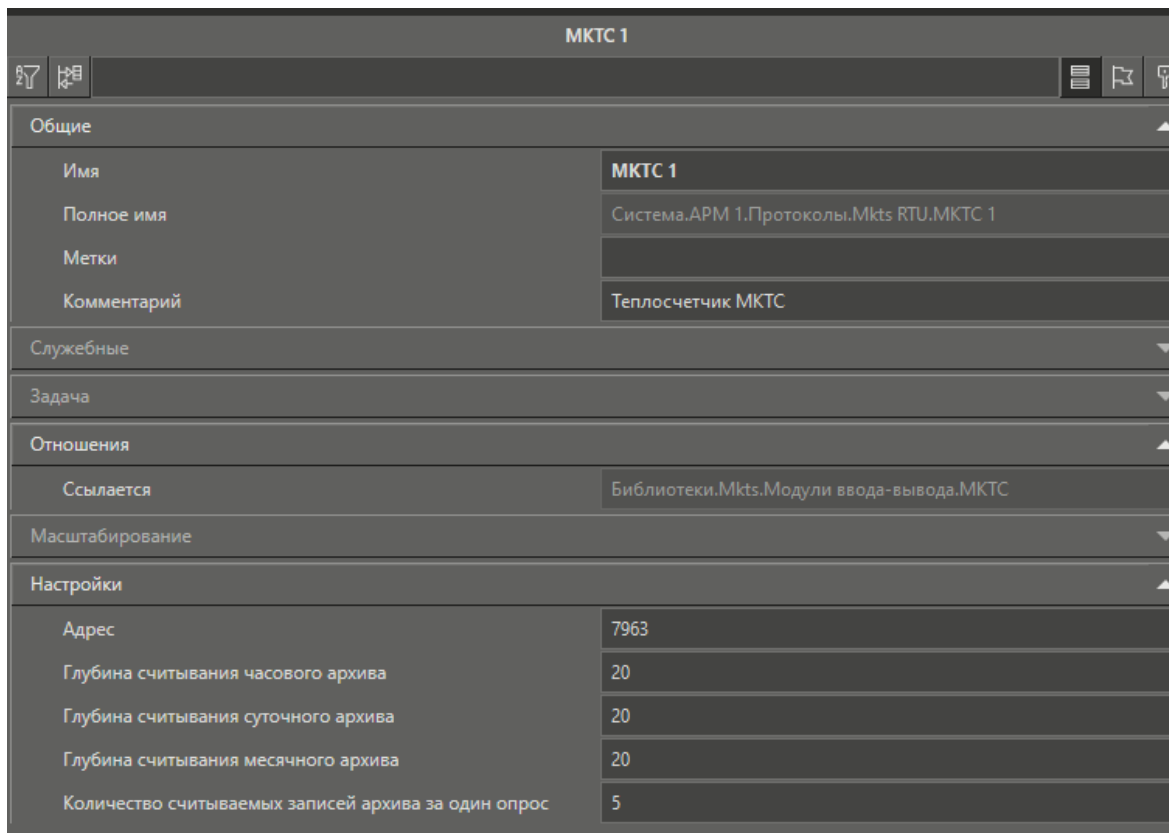
После добавления в дерево проекта модулей ввода-вывода, настраивается панель свойств модулей.

Каждый модуль ввода-вывода имеет выход Отказ, а также группы каналов, согласно его максимальной конфигурации.

Если в опросе каких-либо каналов нет необходимости, то можно либо удалить их из конфигурации элемента уже после добавления его в дерево системы, либо сначала создать экземпляр модуля в пользовательской библиотеке, затем из его конфигурации удалить недействующие каналы, после чего использовать в проекте уже новый видоизмененный библиотечный элемент.

6.2.3.2.17.1. СВОЙСТВА МОДУЛЯ МКТС

Вид панели свойств модуля ввода-вывода протокола Mkts:



MKTS 1	
Общие	
Имя	MKTS 1
Полное имя	Система.АРМ 1.Протоколы.Mkts RTU.MKTS 1
Метки	
Комментарий	Теплосчетчик МКТС
Служебные	
Задача	
Отношения	
Ссылается	Библиотеки.Mkts.Модули ввода-вывода.МКТС
Масштабирование	
Настройки	
Адрес	7963
Глубина считывания часового архива	20
Глубина считывания суточного архива	20
Глубина считывания месячного архива	20
Количество считываемых записей архива за один опрос	5

Описание:

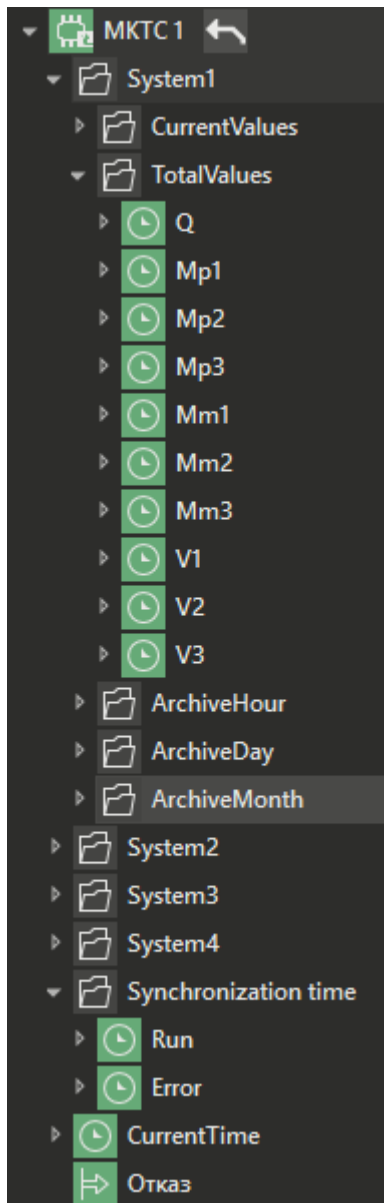
Название	Описание
Категория Общие	Как правило, для данного элемента эта категория не настраивается. Подробное описание данной категории приведено в соответствующем разделе.
Категория Задача	Задается способ опроса устройства. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи.
Категория Масштабирование	Если модуль ввода-вывода не имеет встроенного масштабирования, то при помощи данной категории можно задать соответствие между значениями в модуле и реальными значениями. Подробное описание данной категории приведено в соответствующем разделе.

Название	Описание
Категория Настройки	
Адрес	Задается адрес устройства в сети (значение адреса определяется документацией на подключаемое устройство).
Глубина считывания часового архива	Определяет количество считываемых при старте исполнительной системы записей часового архива. По умолчанию задано максимальное значение
Глубина считывания суточного архива	Определяет количество считываемых при старте исполнительной системы записей суточного архива. По умолчанию задано максимальное значение
Глубина считывания месячного архива	Определяет количество считываемых при старте исполнительной системы записей месячного архива. По умолчанию задано максимальное значение
Количество считываемых записей архивов за цикл	Определяет количество считываемых каждый цикл опроса записей из архива счетчика

6.2.3.2.17.2. МКТС

Теплосчетчик МКТС одновременно может обслуживать до четырех тепловых систем.

Вид счетчика добавленного в протокол в дереве системы:



Состав каналов счетчика:

Название	Назначение
Группы System1 - System4	Каждая группа содержит набор групп каналов для каждой обслуживаемой тепловой системы. Описание группы смотрите ниже.
Группа Synchronization time	Содержит набор каналов для обеспечения синхронизации времени между исполнительной системой и прибором: <ul style="list-style-type: none"> • Run - позволяет выполнить синхронизацию времени. Изменяет время, заданное ранее в счетчике.

Название	Назначение
	<ul style="list-style-type: none"> • Error - сообщает об ошибке синхронизации
CurrentTime	Текущее время прибора

Состав каналов групп System1 - System4:

Название	Назначение
Группа CurrentValues	Содержит текущие измеренные значения системы
Gv1 - Gv3	Объемный расход в трубопроводе 1 - 3
T1 - T3	Температура в трубопроводе 1 - 3
P1 - P3	Давление в трубопроводе 1 - 3
Tcw	Температура холодной воды
Pcw	Давление холодной воды
Gm1 - Gm3	Массовый расход в трубопроводе 1 - 3
W	Тепловая мощность
Tout	Температура атмосферы
Pout	Давление атмосферы
Группа TotalValues	Содержит накопленные значения системы
Q	Тепловая энергия
Mr1 - Mr3	Положительная масса в трубопроводе 1 - 3

Название	Назначение
Mm1 - Mm3	Отрицательная масса в трубопроводе 1 - 3
V1- V3	Объем в трубопроводе 1 - 3
Группы ArchiveHour, ArchiveDay, ArchiveMonth	Содержат архивные значения за час, за день, за месяц соответственно
M plus Pipe 1 - M plus Pipe 3	Масса в прямом направлении трубопровода 1- 3
M minus Pipe 1 - M minus Pipe 3	Масса в обратном направлении трубопровода 1 - 3
V Pipe 1 - V Pipe 3	Объем трубопровода 1 -3
Leak	Масса утечки
Mix	Масса подмеса
Temperature forward Pipe 1 - Temperature forward Pipe 3	Температура прямого трубопровода 1 - 3
Temperature reverse Pipe 1 - Temperature reverse Pipe 3	Температура обратного трубопровода 1 - 3
Temperature Tcw	Температура холодной воды
Temperature Tout	Температура атмосферы
Pressure Pipe 1 - Pressure Pipe 3	Давление трубопровод 1 - 3
Pressure Pcw	Давление холодной воды
Pressure Pout	Давление атмосферы
Q	Интегратор тепла

Название	Назначение
Pointer	Указатель записи (сервисный тег)

Каждый модуль ввода-вывода имеет выход Отказ

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если у архивируемого параметра снять флаг Архивировать, то этот канал выключится из опроса. Если снять флаг Архивировать у параметра Pointer, то вся группа выключится из опроса

6.2.3.2.18.ПРОТОКОЛЫ ДЛЯ РАБОТЫ С БАЗАМИ ДАННЫХ

В данном разделе описаны дополнительные протоколы, которые необходимы для чтения и записи значений параметров в произвольные базы данных.

Данные протоколы не связаны с функцией архивирования.

Поддерживаются следующие типы баз данных:

Протокол PostgreSQL

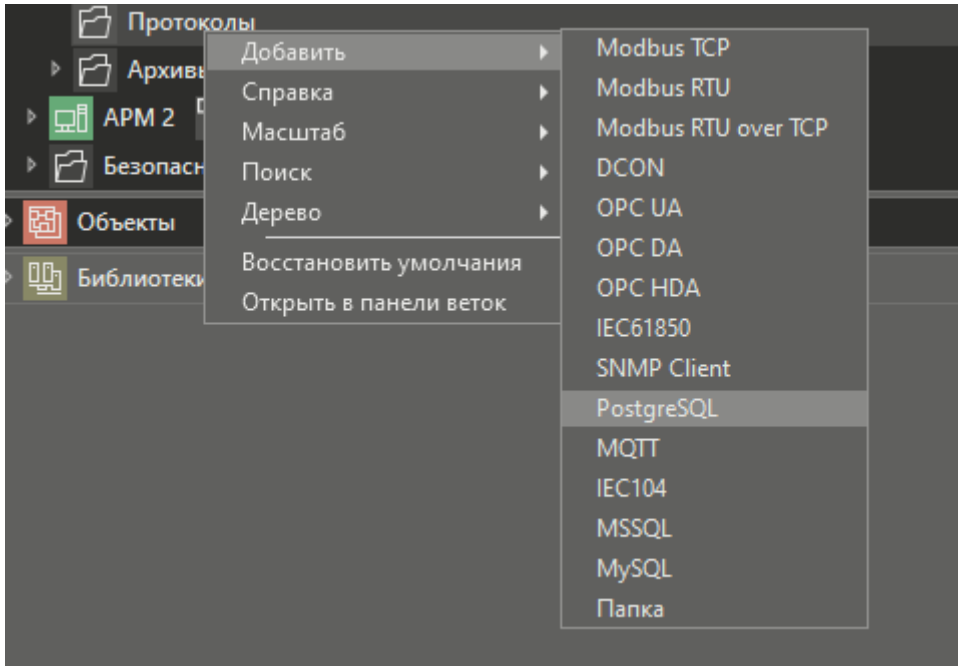
Протокол MSSQL

Протокол MySQL

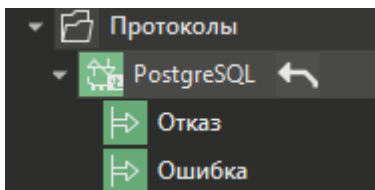
6.2.3.2.18.1. ПРОТОКОЛ POSTGRESQL

Протокол используется для формирования SQL запросов к базе данных PostgreSQL v 9.X.

Чтобы добавить протокол необходимо выбрать соответствующий элемент в контекстном меню узла, или в контекстном меню группы Протоколы, либо через контекстную панель узла:

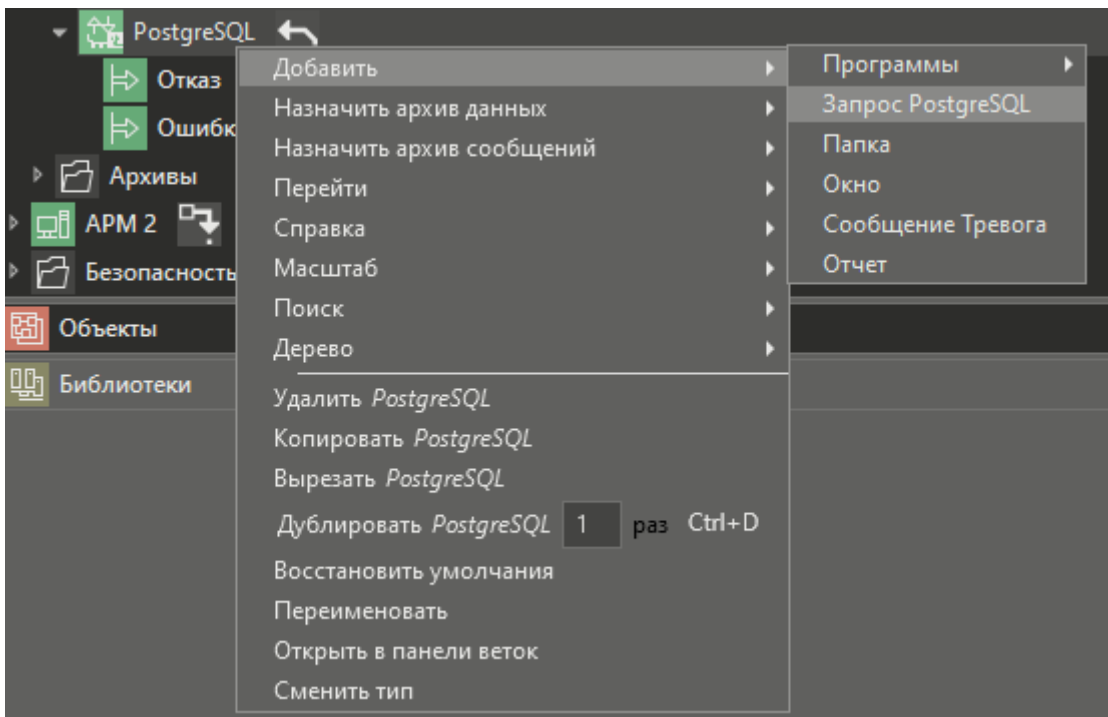


Получим результат:



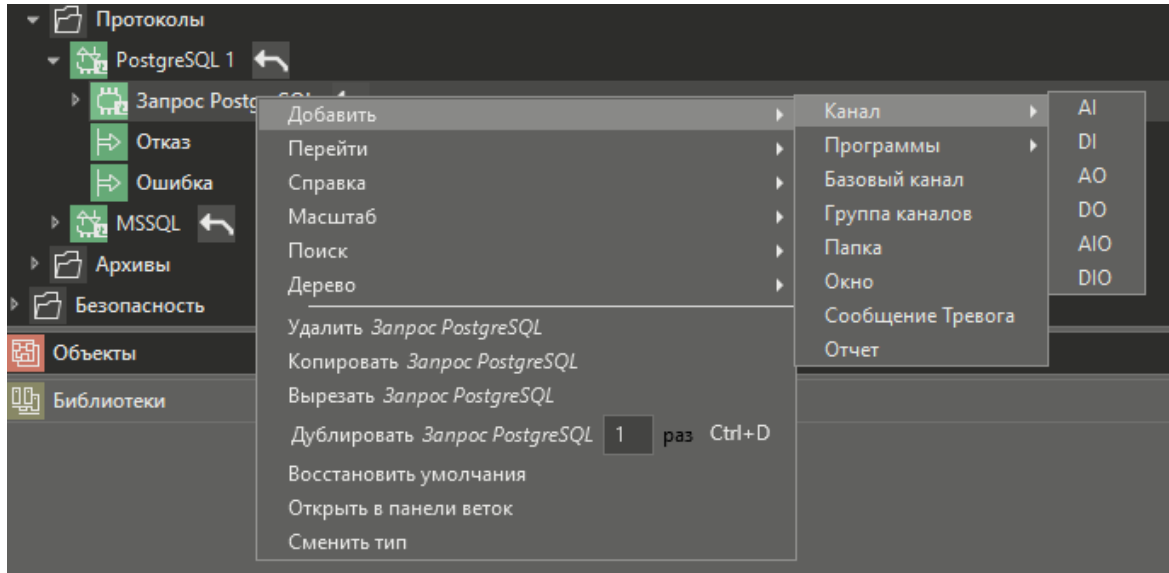
Далее необходимо настроить панель свойств протокола.

Затем при помощи контекстного меню протокола можно добавить Запрос PostgreSQL:



В рамках одного протокола все запросы выполняются последовательно через одно подключение к базе данных.

Затем при помощи контекстного меню Запроса PostgreSQL можно добавить Каналы:



Тип канала и другие настройки зависят от поставленной задачи.

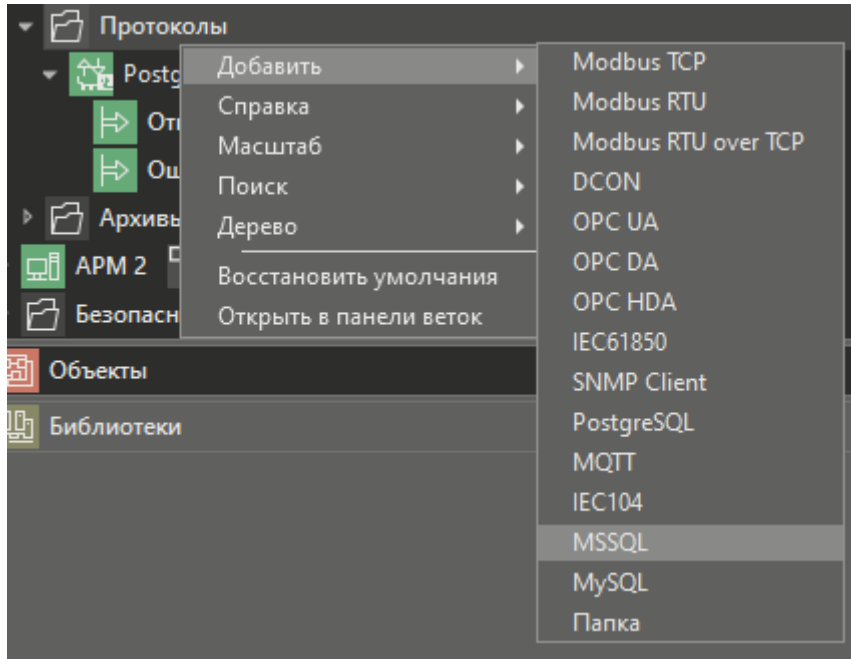
Важно! Имена каналов должны совпадать с именами переменные присутствуют в базе данных, к которой идет обращение

Затем необходимо задать настройки в панели свойств Запрос PostgreSQL.

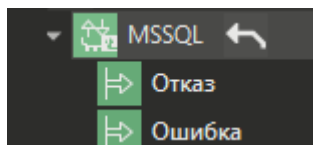
6.2.3.2.18.2. ПРОТОКОЛ MSSQL

Протокол используется для формирования SQL запросов к базе данных MS SQL.

Чтобы добавить протокол необходимо выбрать соответствующий элемент в контекстном меню узла, или в контекстном меню группы Протоколы, либо через контекстную панель узла:

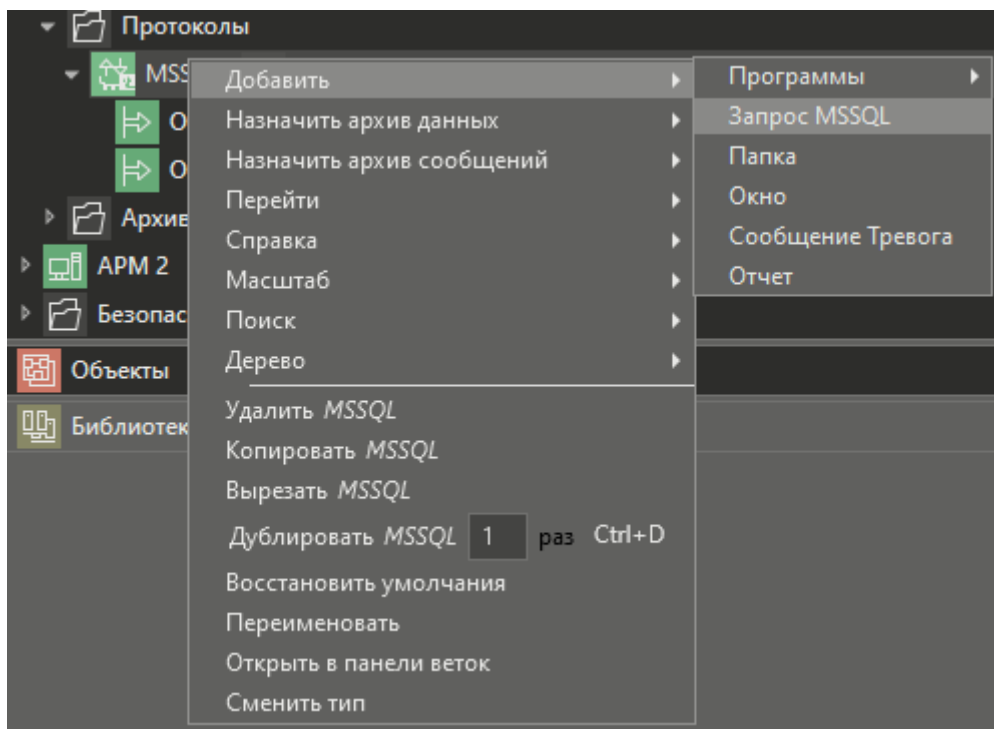


Получим результат:



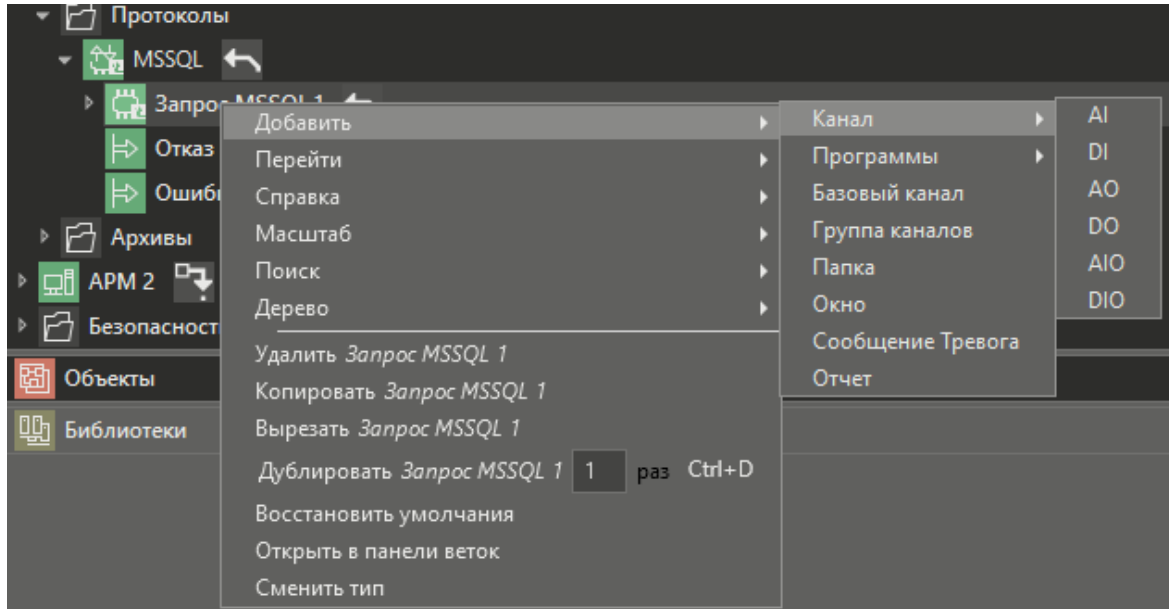
Далее необходимо настроить панель свойств протокола.

Затем при помощи контекстного меню протокола можно добавить Запрос MSSQL.:



В рамках одного протокола все запросы выполняются последовательно через одно подключение к базе данных.

Затем при помощи контекстного меню Запроса MSSQL можно добавить Каналы:



Тип канала и другие настройки зависят от поставленной задачи.

Важно! Имена каналов должны совпадать с именами переменные присутствуют в базе данных, к которой идет обращение

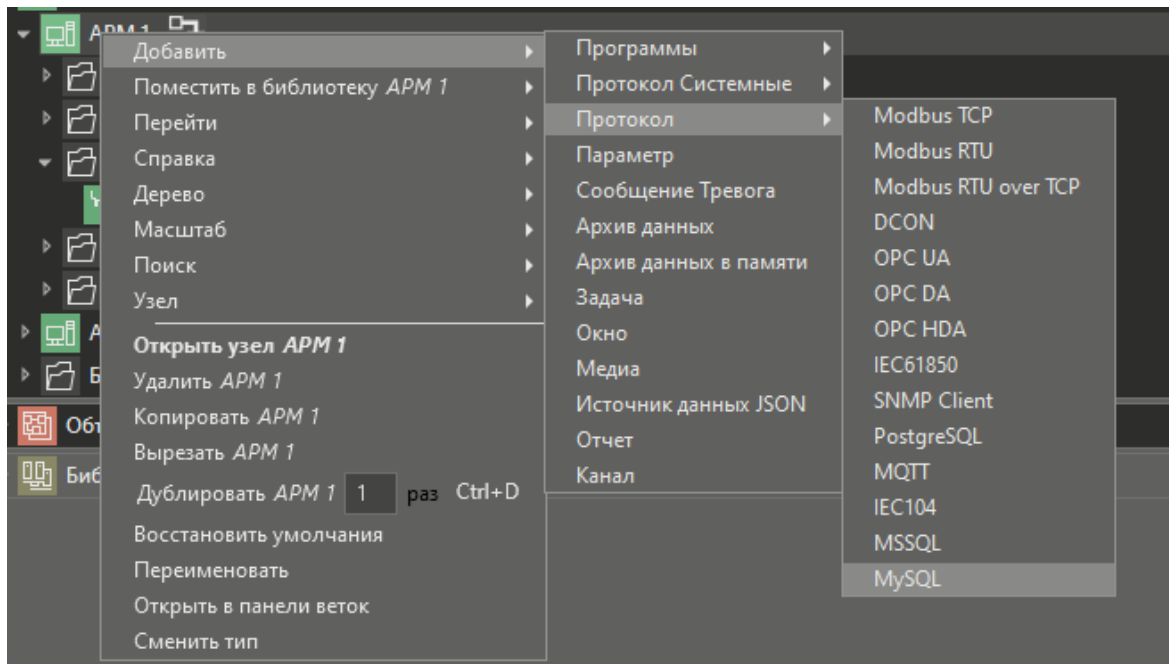
Затем необходимо задать настройки в панели свойств Запрос MSSQL.

Настройки, которые необходимо сделать непосредственно в базе данных описаны в разделе Настройки базы данных MSSQL

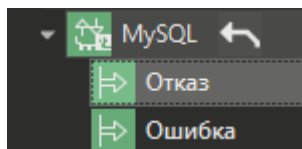
6.2.3.2.18.3. ПРОТОКОЛ MYSQL

Протокол используется для формирования SQL запросов к базе данных MySQL.

Чтобы добавить протокол необходимо выбрать соответствующий элемент в контекстном меню узла, или в контекстном меню группы Протоколы, либо через контекстную панель узла:

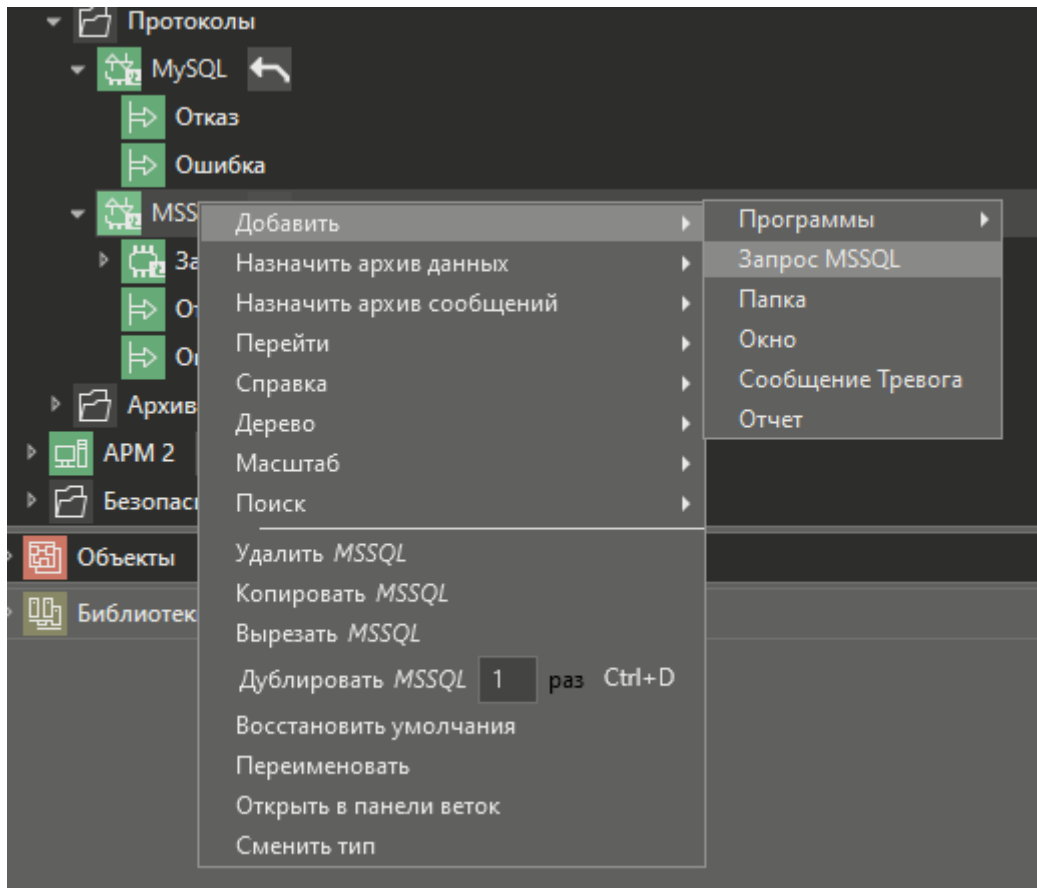


Получим результат:



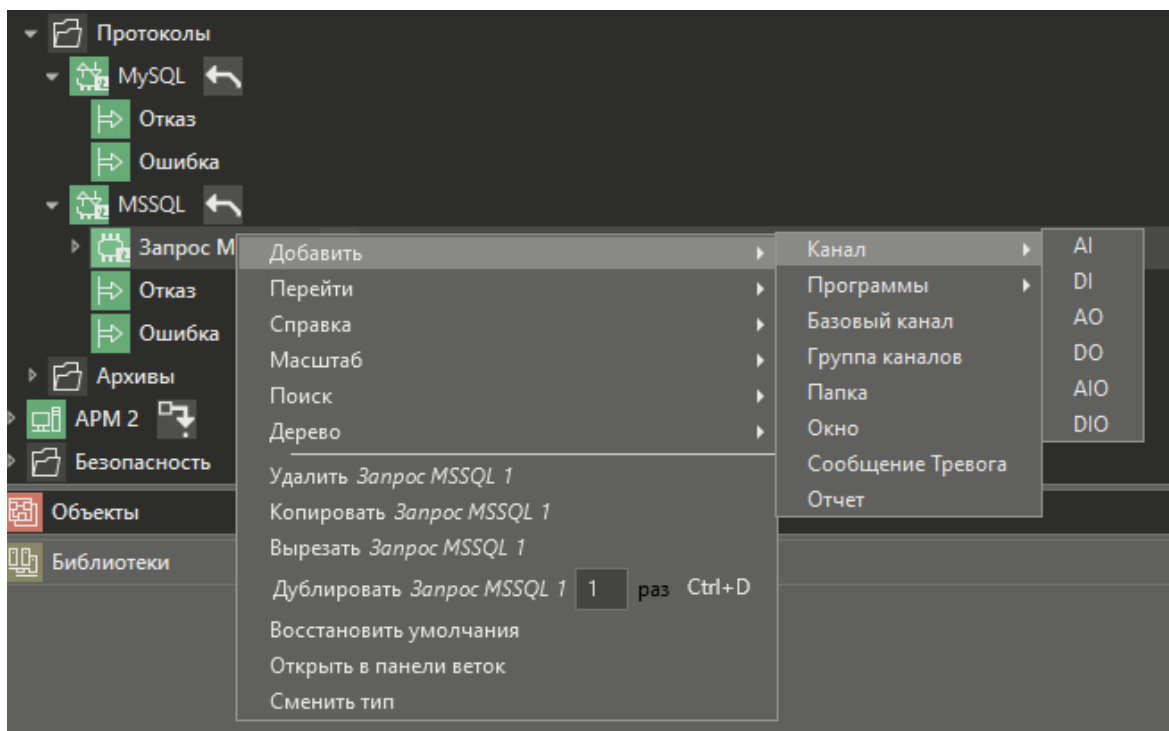
Далее необходимо настроить панель свойств протокола.

Затем при помощи контекстного меню протокола можно добавить Запрос MySQL:



В рамках одного протокола все запросы выполняются последовательно через одно подключение к базе данных.

Затем при помощи контекстного меню Запроса MySQL можно добавить Каналы:



Тип канала и другие настройки зависят от поставленной задачи.

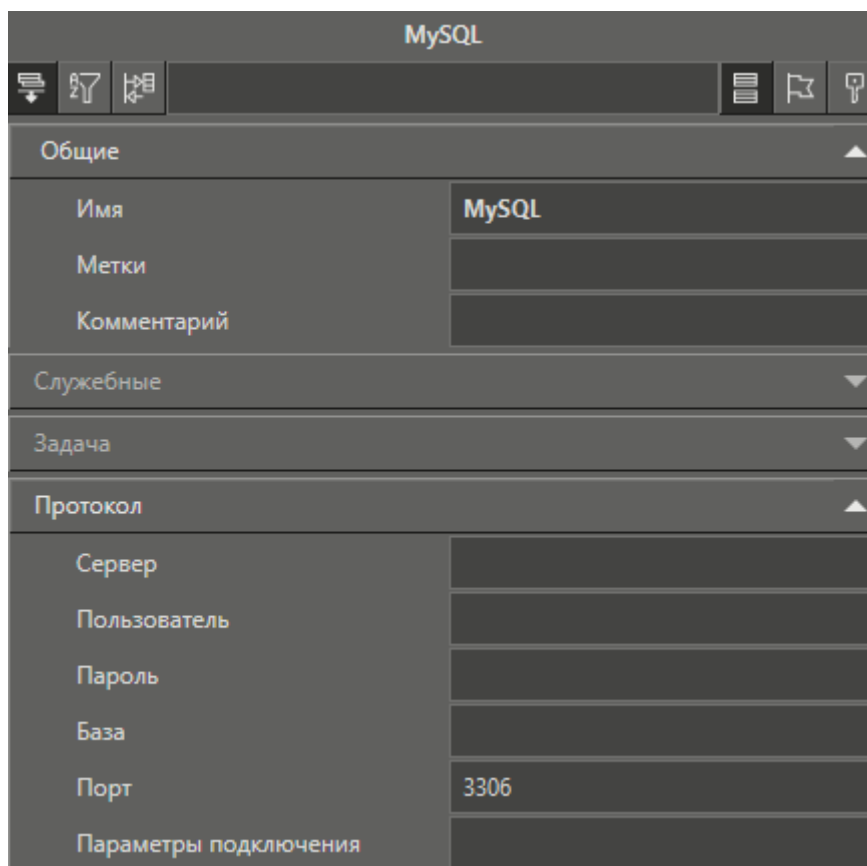
Важно! Имена каналов должны совпадать с именами переменные присутствуют в базе данных, к которой идет обращение

Затем необходимо задать настройки в панели свойств Запрос MySQL.

6.2.3.2.18.4. СВОЙСТВА ПРОТОКОЛОВ ДЛЯ РАБОТЫ С БАЗАМИ ДАННЫХ

В данном разделе описываются настройки протоколов для работы с внешними базами данных.

Вид панели свойств:



Описание:

Название	Рекомендации
Категория Общие	Как правило, для данного элемента не настраивается. Подробное описание данной категории приведено в соответствующем разделе.
Категория Задача	Задается период, с которым будет выполняться подключение к базе данных. Если используется свойство Подключение по условию, то

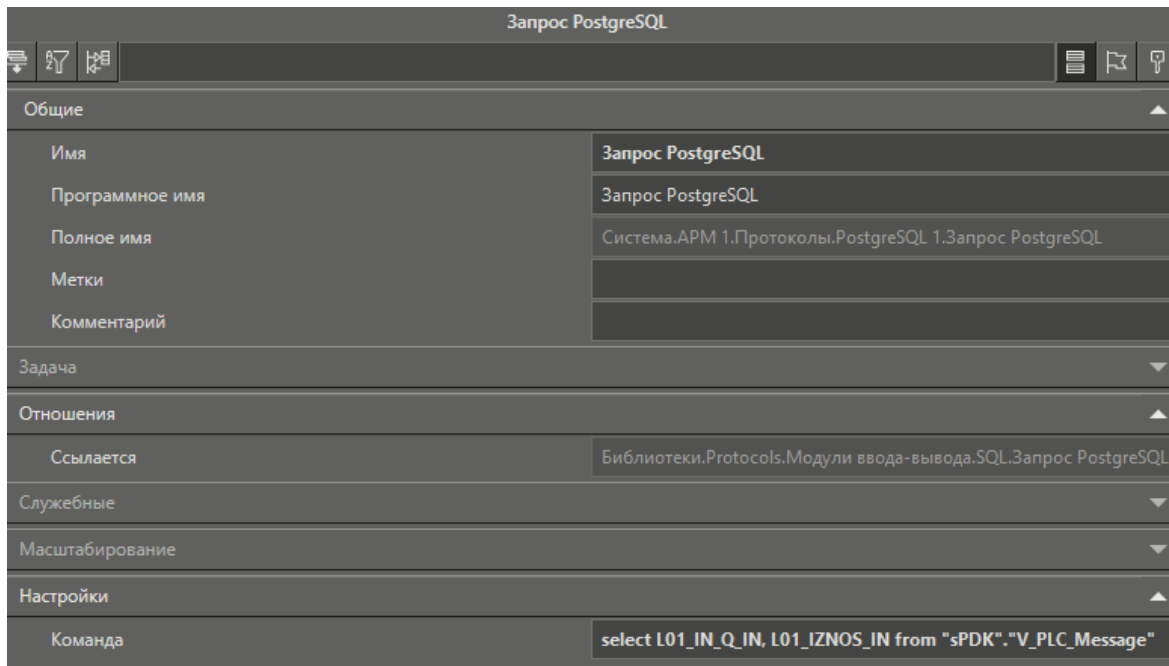
Название	Рекомендации
	<p>если параметр Подключиться становится равным False, то закрывается подключение к базе данных</p> <p>Определяется также способ работы протокола при резервировании узла. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи у протоколов.</p>
Категория Протокол	
Сервер	IP-адрес, DNS, или имя компьютера, на котором находится база данных.
Пользователь	Имя пользователя для подключения к базе данных. Настраивается при конфигурировании базы данных.
Пароль	Пароль для подключения к базе данных. Настраивается при конфигурировании базы данных.
База	Имя базы данных, к которой необходимо обращаться.
Порт	Номер TCP порта, через который идет обмен данных между MasterSCADA 4D и базой. Данный порт должен быть разрешен для использования в применяемых антивирусах и в брандмауэре Windows.
Параметры подключения	Задается в случае, если необходимо передать дополнительные параметры в строку подключения. Например, для установки таймаута 60 сек - connect_timeout=60.

Настройки, которые необходимо сделать непосредственно в базе данных описаны в разделе Настройки базы данных MSSQL

6.2.3.2.18.5. СВОЙСТВА ЗАПРОСА К БАЗЕ ДАННЫХ

Данные настройки производятся при работе с произвольными базами данных.

Вид панели свойств элементов Запрос MySQL, Запрос MSSQL, Запрос PostgreSQL:



Описание:

Название	Рекомендации
Категория Общие	Как правило, для данного элемента не настраивается. Подробное описание данной категории приведено в соответствующем разделе.
Категория Задача	Задается способ опроса базы данных. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи.
Категория Масштабирование	При помощи данной категории можно задать соответствие между значениями в модуле и реальными значениями. Подробное описание данной категории приведено в соответствующем разделе.
Категория Настройки	
Команда	<p>Вводится SQL-запрос. Для чтения значений из канала в тексте запроса вводится имя выходного канала в фигурных скобках.</p> <p>На приведенном рисунке L01_IN_Q_IN, L01_IZNOS_IN - каналы протокола, аналогичные переменные присутствуют в базе данных, к которой идет обращение, а "sPDK"."V_PLC_Message" - название</p>

Название	Рекомендации
	<p>таблицы в указанной базе данных. Параметр канала в режиме исполнения принимает значение, записанное в базе данных.</p> <p>Для записи в базу данных используется команда INSERT. Например, INSERT INTO table (column1, column2) VALUES ({output_channel1}, {output_channel2}) или INSERT INTO "TABLENAME"(DATA) VALUES(@Channelname).</p> <p>Важно! Необходимо ставить пробел после @Channelname, иначе запись не сработает. Channelname - имя выходного канала в дереве протокола.</p>

В режиме исполнения, в случае невозможности выполнения запроса, параметр Ошибка принимает текстовое значение. Например, "Bad channel name: '@CHANNEL_NAME' in command: SQL_COMMAND" - канал не найден.

Синтаксис запросов для работы с базами данных

Есть два способа вставки данных в запрос:

- текст запроса {ИмяКанала} текст запроса..
- текст запроса {\$ИмяКанала} текст запроса..

{CH} Данные и запрос передаются отдельно и "объединение" происходит на сервере БД.

Плюсы:

- Защита от SQL-"инъекций"
- Выполнение запроса происходит немного быстрее

Минусы:

- Не во все части sql запроса можно вставить. Например, сервер БД не позволит выполнить такой запрос: select {CH1} from {CH2}
- Плохо работает с процедурами. По каким-то причинам для mssql не работает подобный вызов вызов процедуры: exec test_proc @param1={CH1}

{\$CH} Данные из каналов напрямую вставляются в тело запроса в момент отправки на сервер.

Плюсы:

- Можно динамизировать запросы, т.е. писать запросы вида: select {\$CH1} from {\$CH2} where {\$CH3} = {CH4}

Минусы:

- Подстановка происходит как есть, без экранирования спецсимволов. Если в CH1 будет текст вида 1; DROP TABLE test; select 1 он отправится на сервер и успешно там выполнится

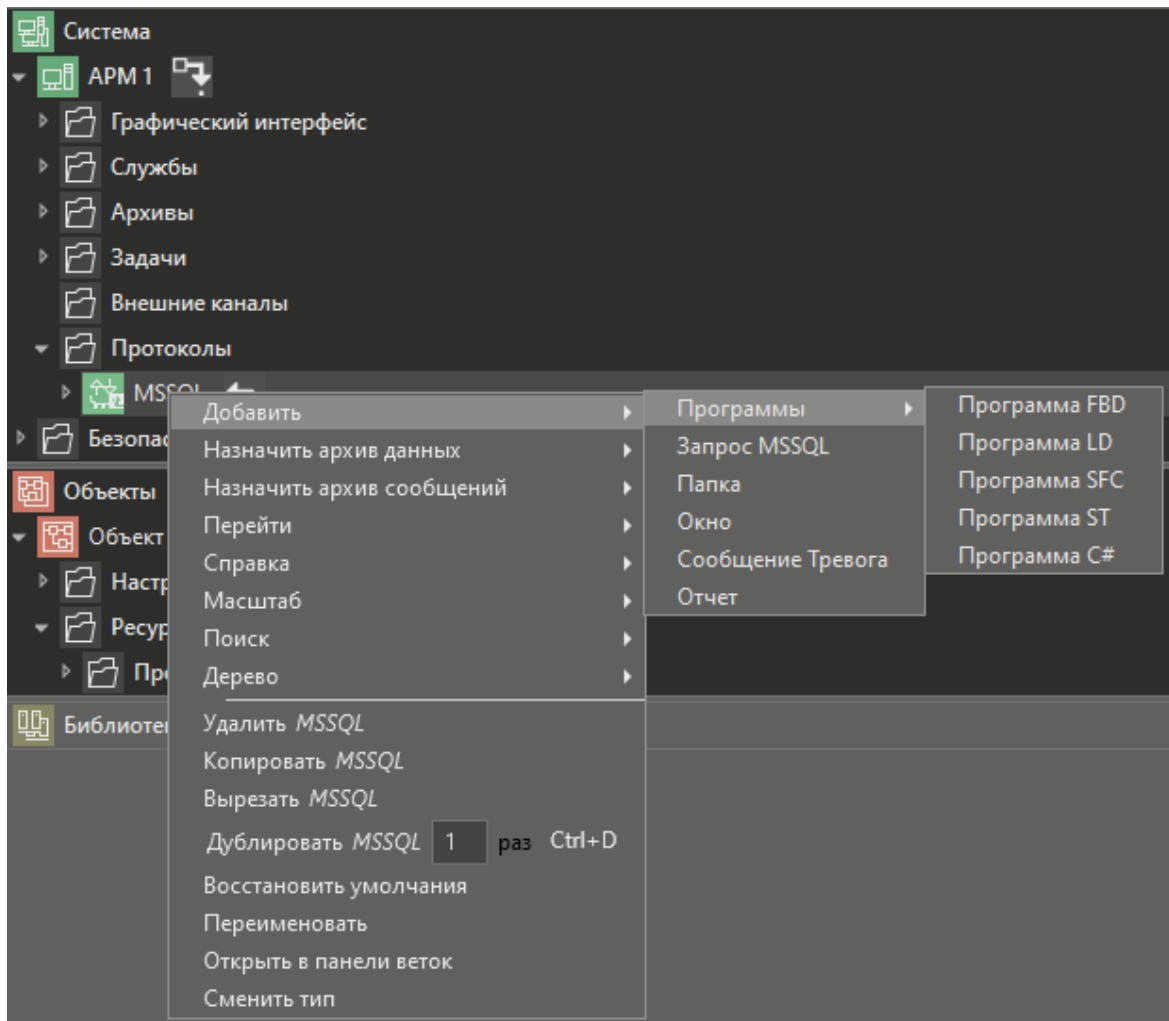
6.2.3.2.18.6. ФУНКЦИОНАЛЬНЫЕ БЛОКИ ДЛЯ РАБОТЫ С БАЗАМИ ДАННЫХ

Расширить возможности протоколов можно при помощи функциональных блоков.

- SqlRequest
- SqlRequestWithResult

Важно! Данные ФБ и их наследников можно использовать только в программах внутри протоколов SQL (MSSQL, PostgreSQL, MySQL).

Чтобы добавить программу в протокол необходимо воспользоваться контекстной панелью или контекстным меню:

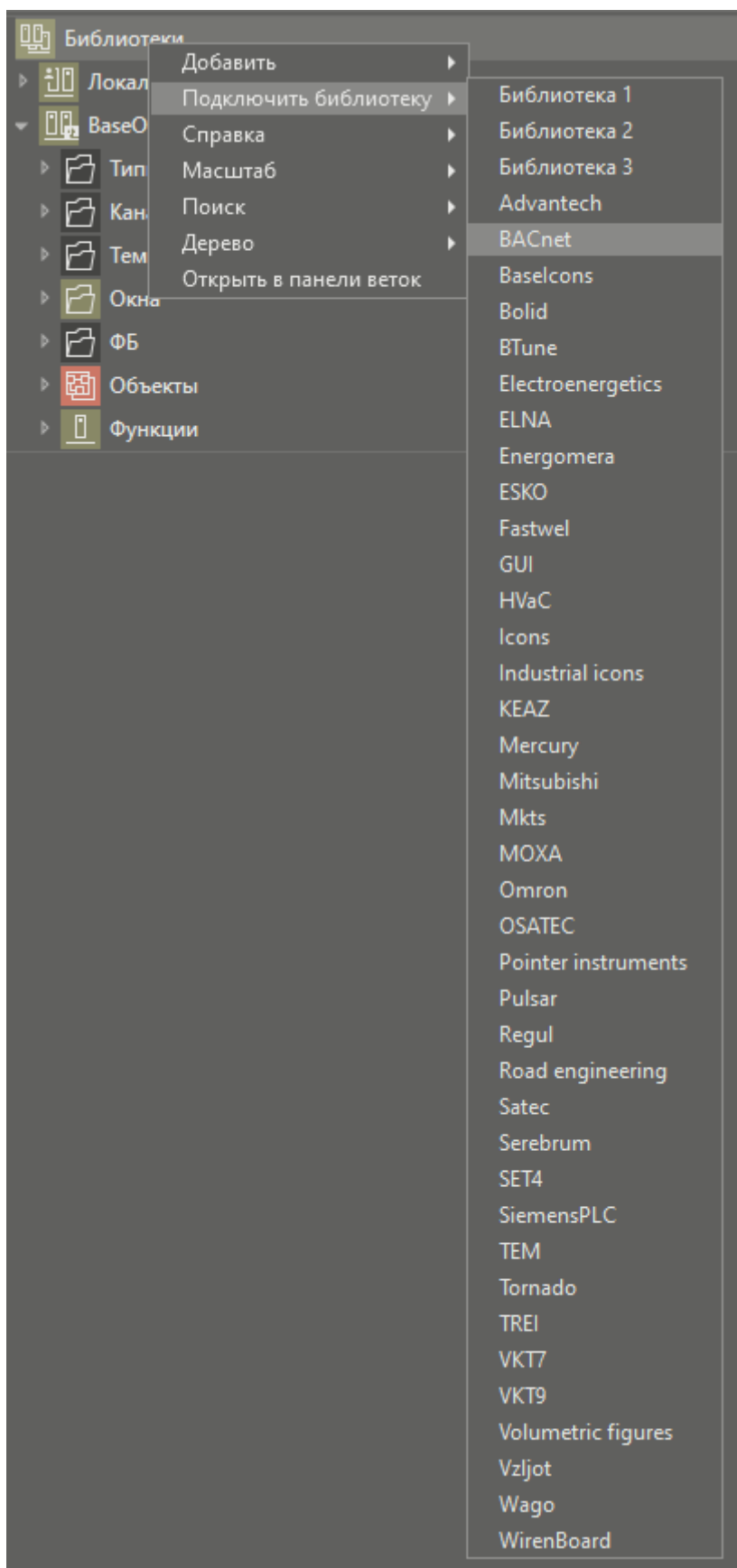


6.2.3.2.19. BACNET

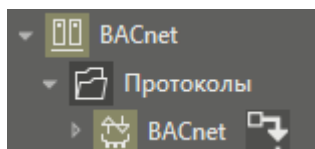
MasterSCADA 4D RT поддерживает работу по протоколу ВАСnet, который предназначен для опроса ВАСnet-устройств по протоколу ВАСnet/IP, обмен ведется по сетям TCP/IP.

Важно! По умолчанию протокол не входит в состав исполнительной системы. Стоимость и условия лицензирования этого протокола необходимо уточнять по электронной почте sales@masterscada.ru

Для получения возможности использования протокола в проекте необходимо подключить соответствующую библиотеку:

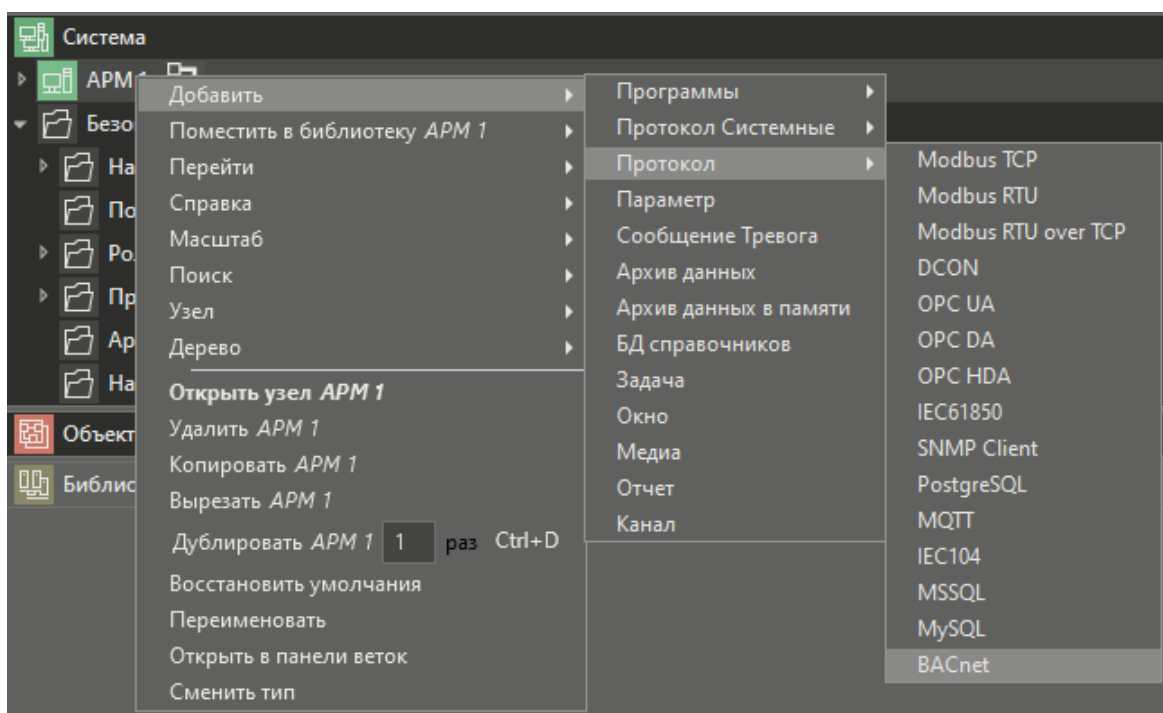


В полном дереве библиотека BACnet имеет вид:



После этого в проект может быть добавлен протокол любым известным способом, например, при помощи контекстного меню узла, или с использованием контекстной панели, или путем перетаскивания элемента из библиотеки в дерево системы.

Рекомендуется добавлять элементы в проект при помощи контекстного меню:



Затем необходимо настроить панель свойств протокола.

Далее требуется добавить каналы в протокол. Обычно это делается на вкладке протокола. Для этого нужно дважды нажать левую кнопку мыши по протоколу в дереве. При этом откроется вкладка, в которой можно выполнить импорт каналов.

Важно! Для использования диалога импорта устройство должно быть подключено непосредственно к компьютеру. Т.е. если планируется подключать устройство к контроллеру, то для формирования списка каналов нужно подключить устройство к компьютеру, считать теги, выбрать нужные или сохранить в Excel. Потом уже подключить устройство к контроллеру и вставить туда устройство с готовым списком каналов или импортировать их из Excel файла.

Далее необходимо настроить панель свойств канала, если не все настройки были заданы при импорте.

Добавить канал вручную можно при помощи контекстного меню протокола.

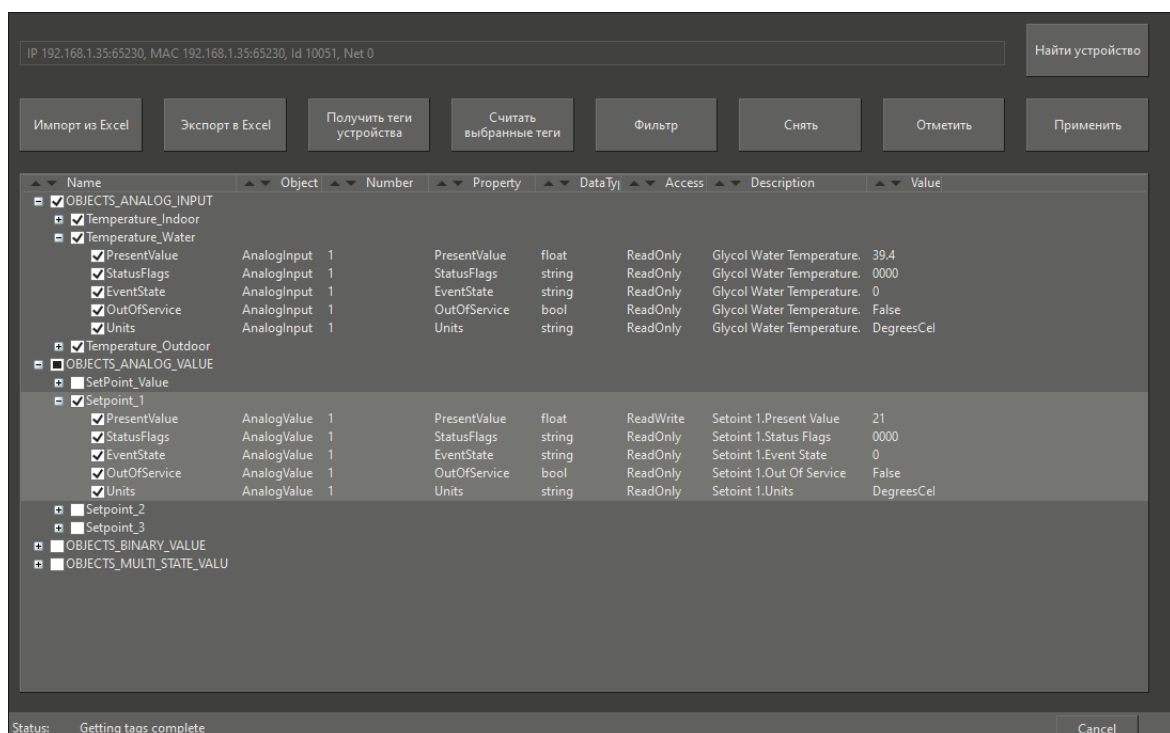
В протокол могут быть добавлены каналы только одного устройства. Для добавления каналов другого устройства, нужно добавить еще один протокол ВАСnet.

6.2.3.2.19.1. ДИАЛОГ ИМПОРТА КАНАЛОВ ВАСNET

Для автоматического добавления каналов и поиска устройств для работы по протоколу ВАСnet нужно дважды нажать левую кнопку мыши по протоколу в дереве. При этом откроется вкладка, в которой можно выполнить поиск устройства и импорт каналов.

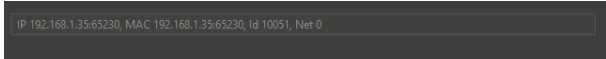
Важно! Для использования диалога импорта устройство должно быть подключено непосредственно к компьютеру. Т.е. если планируется подключать устройство к контроллеру, то для формирования списка каналов нужно подключить устройство к компьютеру, считать теги, выбрать нужные или сохранить в Excel. Потом уже подключить устройство к контроллеру и вставить туда устройство с готовым списком каналов или импортировать их из Excel файла.

Вид вкладки по умолчанию:



Описание элементов вкладки:

Элемент	Описание
Найти устройство	Кнопка открывает диалоговое окно Поиск устройств, которое позволяет найти устройства подключенные к сети. Порядок

Элемент	Описание
	работы с окном описан в разделе Поиск ВАСnet-устройств
	Строка, которая содержит информацию об устройстве, выбранном в окне Поиск устройств.
Импорт из Excel	Позволяет импортировать каналы и их настройки из Excel. Вид файла показан ниже.
Экспорт в Excel	Позволяет экспортировать каналы и их настройки в Excel. Вид файла показан ниже.
Получить теги устройства	Считывает все поддерживаемые объекты и свойства из устройства, которое раньше было найдено и выбрано.
Считать выбранные теги	Позволяет считать текущие значения свойств, отмеченных в таблице объектов
Фильтр	Фильтрует считанные объекты, скрывая не нужные объекты или свойства. Кнопка открывает окно фильтра.
Снять	Снимает флаг выбора у всех выделенных элементов в таблице. Выделенные элементы таблицы отличаются более светлым фоном.
Отметить	Устанавливает флаг выбора у всех выделенных элементов в таблице. Выделенные элементы таблицы отличаются более светлым фоном.

Элемент	Описание
Применить	Добавляет выделенные в таблице каналы в дерево проекта, свойства каналов наследуются из таблицы.
Таблица каналов	Отображаются все каналы устройства и их свойства, которые были считаны автоматически после нажатия на кнопку Получить теги устройства или получены из файла Excel. Каналы отображаются в виде дерева, напротив, каждого элемента имеется поле выбора. Если поле отмечено галочкой, то это значит элемент и все его дочерние элементы выбраны. Если узел дерева отмечен квадратным маркером, то это значит, что не все дочерние элементы были отмечены. Назначение столбцов описано ниже.
Status	Строка статуса отображает текущее состояние
Cancel	Кнопка отмены завершает работу с панелью, каналы добавлены не будут в дерево.

Вид файла Excel, который получается в результате экспорта файла, и который можно использовать в качестве импорта:

1	Name	BacnetObject	Number	Property	DataType	Access	Description	Value
2	OBJECTS_ANALOG_INPUT.Temperature_Indoor.PresentValue	AnalogInput	0	PresentValue	float	ReadOnly	Indoor Temperature.Present Value	21.2
3	OBJECTS_ANALOG_INPUT.Temperature_Indoor.StatusFlags	AnalogInput	0	StatusFlags	string	ReadOnly	Indoor Temperature.Status Flags	0000
4	OBJECTS_ANALOG_INPUT.Temperature_Indoor.EventState	AnalogInput	0	EventState	string	ReadOnly	Indoor Temperature.Event State	0
5	OBJECTS_ANALOG_INPUT.Temperature_Indoor.OutOfService	AnalogInput	0	OutOfService	bool	ReadOnly	Indoor Temperature.Out Of Service	False
6	OBJECTS_ANALOG_INPUT.Temperature_Indoor.Units	AnalogInput	0	Units	string	ReadOnly	Indoor Temperature.Units	62
7	OBJECTS_ANALOG_INPUT.Temperature_Water.PresentValue	AnalogInput	1	PresentValue	float	ReadOnly	Glycol Water Temperature.Present Value	39.7
8	OBJECTS_ANALOG_INPUT.Temperature_Water.StatusFlags	AnalogInput	1	StatusFlags	string	ReadOnly	Glycol Water Temperature.Status Flags	0000
9	OBJECTS_ANALOG_INPUT.Temperature_Water.EventState	AnalogInput	1	EventState	string	ReadOnly	Glycol Water Temperature.Event State	0
10	OBJECTS_ANALOG_INPUT.Temperature_Water.OutOfService	AnalogInput	1	OutOfService	bool	ReadOnly	Glycol Water Temperature.Out Of Service	False
11	OBJECTS_ANALOG_INPUT.Temperature_Water.Units	AnalogInput	1	Units	string	ReadOnly	Glycol Water Temperature.Units	62
12	OBJECTS_ANALOG_INPUT.Temperature_Outdoor.PresentValue	AnalogInput	2	PresentValue	float	ReadOnly	Outdoor Temperature.Present Value	12
13	OBJECTS_ANALOG_INPUT.Temperature_Outdoor.StatusFlags	AnalogInput	2	StatusFlags	string	ReadOnly	Outdoor Temperature.Status Flags	0000
14	OBJECTS_ANALOG_INPUT.Temperature_Outdoor.EventState	AnalogInput	2	EventState	string	ReadOnly	Outdoor Temperature.Event State	0
15	OBJECTS_ANALOG_INPUT.Temperature_Outdoor.OutOfService	AnalogInput	2	OutOfService	bool	ReadOnly	Outdoor Temperature.Out Of Service	False
16	OBJECTS_ANALOG_INPUT.Temperature_Outdoor.Units	AnalogInput	2	Units	string	ReadOnly	Outdoor Temperature.Units	62
17	OBJECTS_ANALOG_VALUE.Setpoint_Value.PresentValue	AnalogValue	0	PresentValue	float	ReadWrite	Effective SetPoint.Present Value	21
18	OBJECTS_ANALOG_VALUE.Setpoint_Value.StatusFlags	AnalogValue	0	StatusFlags	string	ReadOnly	Effective SetPoint.Status Flags	0000
19	OBJECTS_ANALOG_VALUE.Setpoint_Value.EventState	AnalogValue	0	EventState	string	ReadOnly	Effective SetPoint.Event State	0
20	OBJECTS_ANALOG_VALUE.Setpoint_Value.OutOfService	AnalogValue	0	OutOfService	bool	ReadOnly	Effective SetPoint.Out Of Service	False
21	OBJECTS_ANALOG_VALUE.Setpoint_Value.Units	AnalogValue	0	Units	string	ReadOnly	Effective SetPoint.Units	DegreesCelsius
22	OBJECTS_ANALOG_VALUE.Setpoint_1.PresentValue	AnalogValue	1	PresentValue	float	ReadWrite	Setpoint 1.Present Value	21
23	OBJECTS_ANALOG_VALUE.Setpoint_1.StatusFlags	AnalogValue	1	StatusFlags	string	ReadOnly	Setpoint 1.Status Flags	0000
24	OBJECTS_ANALOG_VALUE.Setpoint_1.EventState	AnalogValue	1	EventState	string	ReadOnly	Setpoint 1.Event State	0
25	OBJECTS_ANALOG_VALUE.Setpoint_1.OutOfService	AnalogValue	1	OutOfService	bool	ReadOnly	Setpoint 1.Out Of Service	False
26	OBJECTS_ANALOG_VALUE.Setpoint_1.Units	AnalogValue	1	Units	string	ReadOnly	Setpoint 1.Units	DegreesCelsius
27	OBJECTS_ANALOG_VALUE.Setpoint_2.PresentValue	AnalogValue	2	PresentValue	float	ReadWrite	Setpoint 2.Present Value	19
28	OBJECTS_ANALOG_VALUE.Setpoint_2.StatusFlags	AnalogValue	2	StatusFlags	string	ReadOnly	Setpoint 2.Status Flags	0000
29	OBJECTS_ANALOG_VALUE.Setpoint_2.EventState	AnalogValue	2	EventState	string	ReadOnly	Setpoint 2.Event State	0
30	OBJECTS_ANALOG_VALUE.Setpoint_2.OutOfService	AnalogValue	2	OutOfService	bool	ReadOnly	Setpoint 2.Out Of Service	False
31	OBJECTS_ANALOG_VALUE.Setpoint_2.Units	AnalogValue	2	Units	string	ReadOnly	Setpoint 2.Units	DegreesCelsius
32	OBJECTS_ANALOG_VALUE.Setpoint_3.PresentValue	AnalogValue	3	PresentValue	float	ReadWrite	Setpoint 3.Present Value	17
33	OBJECTS_ANALOG_VALUE.Setpoint_3.StatusFlags	AnalogValue	3	StatusFlags	string	ReadOnly	Setpoint 3.Status Flags	0000
34	OBJECTS_ANALOG_VALUE.Setpoint_3.EventState	AnalogValue	3	EventState	string	ReadOnly	Setpoint 3.Event State	0
35	OBJECTS_ANALOG_VALUE.Setpoint_3.OutOfService	AnalogValue	3	OutOfService	bool	ReadOnly	Setpoint 3.Out Of Service	False
36	OBJECTS_ANALOG_VALUE.Setpoint_3.Units	AnalogValue	3	Units	string	ReadOnly	Setpoint 3.Units	DegreesCelsius

Таблица каналов имеет следующие столбцы:

Таблица тегов

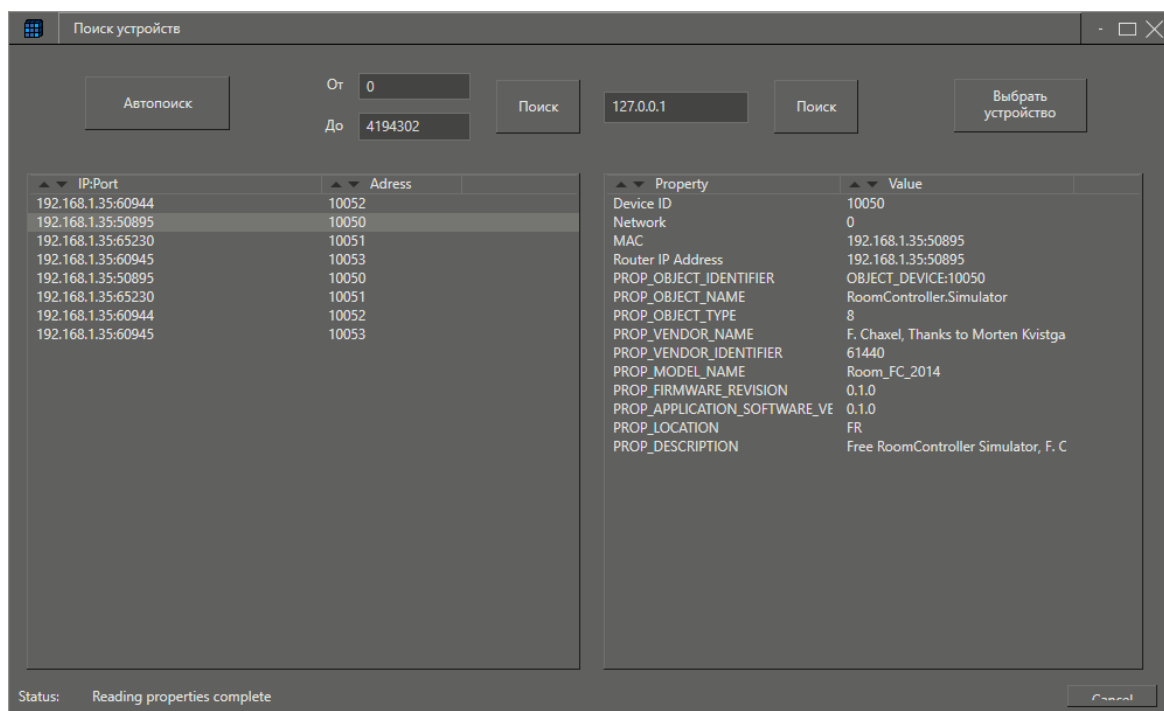
В таблицу выводятся все считанные из устройства, поддерживаемые теги. По умолчанию объекты сгруппированы по типу объекта. Каждый объект содержит его свойства. Таблица имеет следующие столбцы:

- Name - полное имя объекта. Аналогичная иерархия и имена будут при импорте каналов в дерево проекта.
- Object - тип BACnet объекта.
- Number - номер BACnet объекта.
- Property - свойство BACnet объекта, которое будет опрашивать данный канал.
- DataType - тип данных канала.
- Access - тип доступа канала. Определяет типом BACnet объекта и свойства.
- Description - описание канала. Состоит из двух частей, разделенных точкой. В первой части - считанное свойство Description объекта BACnet, во второй части - локализованное название считываемого свойства.
- Value - текущее значение тега. У некоторых тегов, имеющих сложную структуру тега (например объект Sheduler), реальное значение не выводится.

6.2.3.2.19.2. ПОИСК BACNET-УСТРОЙСТВ

Окно поиска BACnet- устройств запускается с вкладки протокола, после нажатия на кнопку Найти устройство.

Вид окна:



Возможны следующие способы поиска устройств:

- Автоматический - нужно нажать на кнопку Автопоиск, тогда будет произведен поиск по всем доступным сетям всех доступных устройств;
- Поиск в диапазоне ВАСnet-адресов - нужно в полях От и До определить диапазон адресов и нажать кнопку Поиск, которая находится рядом с этими полями. Поиск устройств в определенном диапазоне ВАСnet-адресов (например, с 1 по 10 устройство);
- Поиск по IP - в поле, в котором задан шаблон IP-адреса можно ввести любой произвольный IP-адрес и нажать кнопку Поиск, которая находится правее, тогда произведется поиск устройства по заданному IP-адресу

Для поиска запускается специальная утилита.

Найденные устройства отобразятся в панели со столбцами - IP:Port (IP-адрес и порт устройства) и Address (указывается ВАСnet-адрес)

Свойства выделенного устройства отображаются в панели справа. В панель выводятся основные параметры (IP-адрес, MAC, адрес устройства, номер сети), а также вспомогательные свойства, которые упрощают идентификацию устройства - имя устройства, тип, производитель, расположение, описание. Вспомогательные настройки для дальнейшей работы не используются

В нижней части окна находится панель Status. В данную панель выводится состояние механизма поиска устройств. В правой части находится кнопка отмены поиска.

Чтобы продолжить работу с нужным устройством его нужно выбрать в панели слева и нажать на кнопку Выбрать устройство. Тогда окно закроется и отобразится Диалог импорта каналов

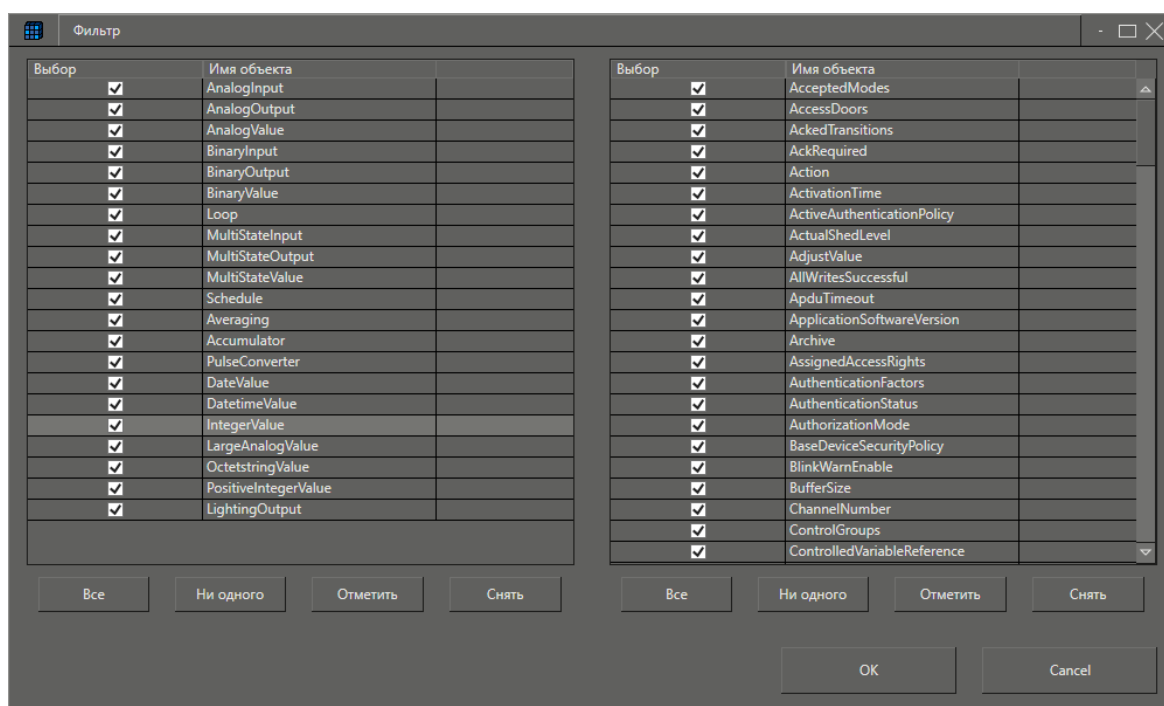
Общий механизм поиска и обнаружения устройств

Для идентификации ВАСnet-устройства, утилита поиска посылает широковещательные (в случае автоматического поиска или поиска в диапазоне) или на заданный IP-адрес (в случае поиска по IP) запросы whois. Устройства, находящиеся в сети, должны отслеживать подобные запросы, и при их обнаружении ответить специальным ответом - i am devicenumber. При получении данного ответа утилита добавляет данное устройство в список найденных, и производит опрос его основных параметров - mac, номер устройства и номер сети. При щелчке по найденному устройству в панели устройств, производится считывание вспомогательных свойств, которые выводятся на панель свойств.

6.2.3.2.19.3. ОКНО ФИЛЬТРА ВАСNET

Окно фильтра открывается со вкладки протокола ВАСnet. после нажатия на кнопку Фильтр.

Вид окна:



Слева находится таблица, в которой можно выбрать все ВАСnet-объекты, которые могут быть опрошены, в таблице справа находится таблица, где представлены все ВАСnet свойства, которые могут быть опрошены. Под каждой таблицей находятся независимые элементы управления каждой таблицей. Используя данные кнопки, можно выполнять групповое включение флагов в панели объектов или свойств:

- Все - включает все флаги.
- Ни одного - сбрасывает все флаги.
- Выбрать - включает флаги во всех выделенных строках.
- Снять выбор - выключает флаги во всех выделенных строках.

После нажатия кнопки ОК, выполняется фильтрация таблицы считанных тегов согласно настройкам данного окна.

Важно! Фильтрация применяется к текущим отображаемым тегам. Если повторно запустить считывание тегов из устройства или импортировать из Excel, настройки фильтра будут сброшены.

6.2.3.2.19.4. СВОЙСТВА ПРОТОКОЛА VASNET

Вид панели свойств:

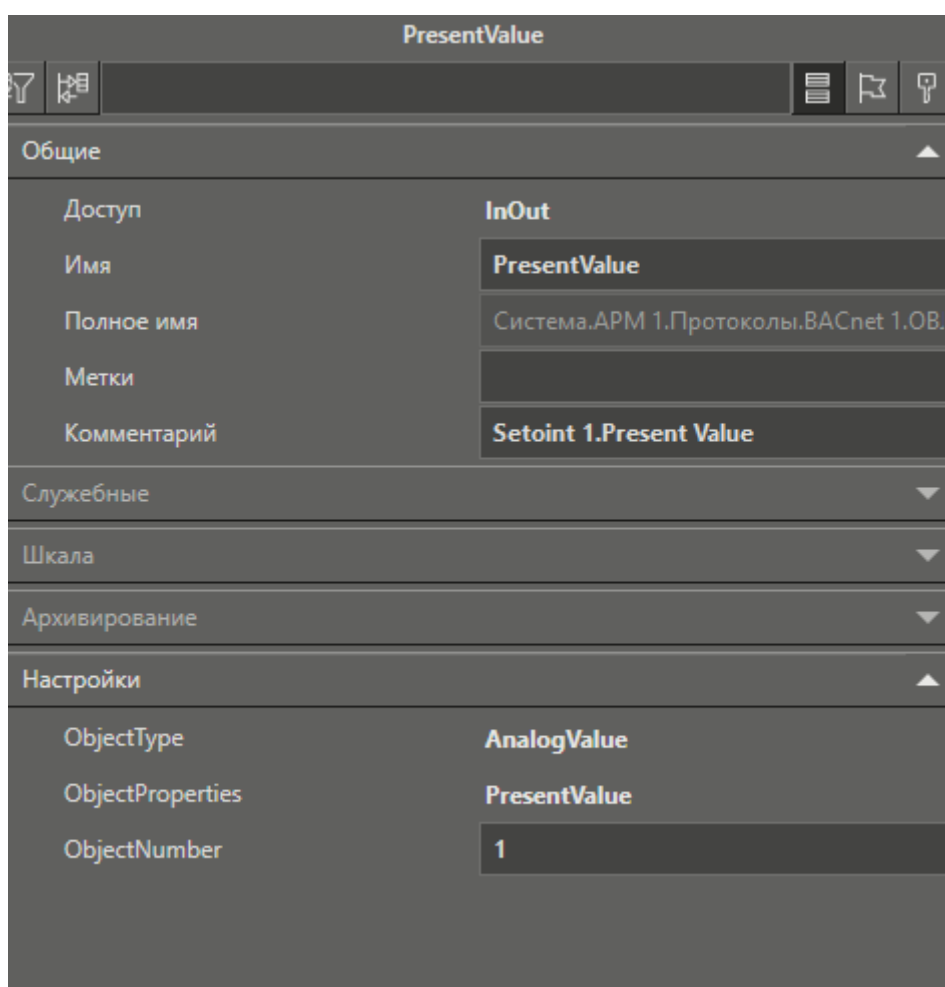
BACnet	
Общие	
Имя	BACnet
Полное имя	Система.APM 1.Протоколы.BACnet
Метки	
Комментарий	
Служебные	
Задача	
Период, мс	100
Приоритет	100
Подключение по условию	<input type="checkbox"/>
Выполнение по условию	<input type="checkbox"/>
Выполнять на резервном	<input type="checkbox"/>
Выполнять запись на резервном	<input type="checkbox"/>
Способ записи	По изменению
Формировать отказ при отказе всех модулей	<input type="checkbox"/>
Формировать отказ узла	<input checked="" type="checkbox"/>
Задержка записи при старте опроса (циклов)	0
Задержка сброса отказа на резервном, мс	10000
Отношения	
Ссылается	Библиотеки.BACnet.Протоколы.BACnet
Настройки каналов	
Протокол	
IP адрес	127.0.0.1:47808
MAC адрес	255.255.255.255:47808
Таймаут	1000
Количество попыток	3
Номер устройства	1
Номер сети	0
Приоритет записи	None

Название	Рекомендации
Категория Общие	Как правило, для данного элемента эта категория не настраивается. Подробное описание данной категории приведено в соответствующем разделе
Категория Задача	Задается период, с которым MasterSCADA 4D будет стараться получать значения из сервера. Определяется способ работы протокола при резервировании узла. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи у протоколов.
Категория Настройки каналов	Задаются значения по умолчанию для создаваемых каналов. В настройке не нуждаются, т.к. индивидуальные настройки каналов формируются автоматически.
Категория Протокол	
IP-адрес	IP-адрес устройства и номер порта. При поиске устройства заполняется автоматически. Формат записи: IP:port
MAC-адрес	MAC-адрес устройства и номер порта. При поиске устройства заполняется автоматически. Формат записи: MAC-адрес:port, например,
Таймаут	Время, в течение которого исполнительная система ожидает установления соединения с контроллером, мс
Количество попыток	Определяется количество попыток выполнения запросов чтения/записи. Если при опросе/записи канала запрос выполнить не удастся указанное количество попыток, происходит переход к следующему каналу.
Номер устройства	BACnet-адрес устройства. Исполнительная система не использует данный параметр, он необходим для корректного функционирования диалога поиска (по данному параметру производится поиск устройства). При импорте каналов считывается автоматически.

Название	Рекомендации
Номер сети	Используется при работе с различными ВАСnet сетями, имеющими разные номера (например, при объединении ВАСNET IP и ВАСNET MS/TP). При импорте тегов считывается автоматически.
Приоритет записи	При выполнении записи в ВАСnet-устройство, есть возможность задать приоритет - от 1 до 16, где 1 - высший приоритет, 16 - низший. None - без выдачи приоритета (устройством воспринимается как низший приоритет).

6.2.3.2.19.5. СВОЙСТВА КАНАЛА ВАСNET

Вид панели свойства канала ВАСnet:



Панель свойств заполняется автоматически после выполнения необходимых действий на вкладке протокола.

Свойство	Назначение
Комментарий	Описание канала. Состоит из двух частей, разделенных точкой. В первой части - считанное свойство Description объекта ВАСnet, во второй части - локализованное название считываемого свойства.
ObjectType	Тип ВАСnet объекта
ObjectProperties	Свойство ВАСnet объекта, которое будет опрашивать данный канал
ObjectNumber	Номер ВАСnet объекта

6.2.3.2.19.6. РЕКОМЕНДАЦИИ ПО НАСТРОЙКЕ ПРОТОКОЛА ВАСNET

Ниже описывается рекомендованная последовательность настройки протокола ВАСnet.

Важно! В связи с тем что механизм поиска работает с сетью на время работы с ней настоятельно рекомендуется отключить фаервол/антивирус на компьютере, либо добавить утилиту в список доверенных и разрешить ей работу по протоколу UDP, включая широковещательные запросы.

Перед началом работ, рекомендуется убедиться в наличии связи с устройствами - сделать Ping к устройствам, а также попробовать подключиться тестовым ВАСnet-клиентом - например, YABE <https://sourceforge.net/projects/yetanotherbacnetexplorer/>

Поиск устройств

В узел добавьте протокол ВАСnet, дважды кликните по нему. В открывшейся вкладке нажмите кнопку Найти устройство. Откроется окно Поиск устройств

Важно! При поиске утилита посылает широковещательные запросы и ждет ответы от устройства, в течении двух таймаутов заданных в настройках протокола (по умолчанию 500 мс). При сильно нагруженной сети или работе через беспроводные сети, рекомендуется увеличить таймаут в настройках устройства, а уже затем запускать утилиту поиска устройств.

Если адреса устройств точно не известны нажать кнопку Автопоиск - запустится автоматический поиск. В этом режиме утилита поочередно перебирает все доступные сети и посылает запрос WhoIs и ожидает ответ. В строку состояния поиска при этом выдается опрашиваемая сеть и обнаруженные устройства.

Если известны точные номера устройств (или номер конкретного устройства) можно воспользоваться поиском в диапазоне. Принцип работы в этом режиме аналогичен автоматическому поиску, но в запросе WhoIs также передается заданный диапазон адресов.

Если автоматический поиск не обнаружил полный список необходимых устройств, можно еще 1-2 раза повторить поиск - иногда ответы устройств могут теряться, и повторные запросы могут решить проблему.

Если не удалось найти все или часть устройств, можно воспользоваться поиском по IP-адресу. В этом режиме утилита также посылает запрос WhoIs, но уже на указанный адрес, что в ряде случаев (например при работе через VPN) может привести к получению ответа от устройства.

После получения ответов и завершения опроса сетей, все доступные устройства добавляются в панель устройств. При этом возможна ситуация, когда одно и то же устройство может быть добавлено несколько раз, в этом случае можно выбрать любое из них. При выборе устройства, происходит однократное считывание его вспомогательных параметров (модель, производитель, описание, расположение), с помощью которых можно идентифицировать нужное устройство. Выберите нужное устройство и нажмите кнопку **Выбрать устройство**. Утилита импорта закроется, настройки выбранного устройства применятся во вкладке протокола.

Если никакой из вариантов поиска не дал результата, то тогда нужно закрыть утилиту импорта и ввести параметры устройства в настройках протокола. Для этого потребуются ввести:

- Адрес роутера - IP адрес и порт устройства (например 192.168.100.12:47808).
- Номер сети - взять из настроек контроллера.
- Номер устройства и адрес устройства - это один и тот же адрес контроллера. Найдите его в настройках контроллера и укажите это значение в обоих полях.

Утилита импорта тегов при этом сможет функционировать - она может обращаться к устройству не только через широковещательные запросы, но и использовать обычный режим опроса (как исполнительная система в режиме исполнения). Поэтому даже при ручном задании настроек подключения возможность импорта тегов сохраняется и можно переходить к следующему шагу.

Импорт тегов

Откройте вкладку протокола.

Нажмите на кнопку **Получить теги устройства**. Сначала утилита произведет поиск и установит связь с устройством. В связи с тем что ВАСnet устройства могут применяться в составе сложных сетей, сделан многоступенчатый алгоритм установления связи:

1. На первом этапе выполняется попытка прямого опроса параметра по заданному в настройках IP адресу и адресу устройства (аналогично как работает исполнительная система в режиме исполнения).

2. Если первый этап был не успешен, посылается адресный запрос WhoIs на IP-адрес заданный в настройках устройства.

3. Если второй этап был не успешен, производится широковещательный запрос WhoIs по всем сетям. После этого, если устройств обнаружено несколько происходит их сравнение по IP-адресу, адресу устройства и сети.

Если на каком-либо этапе удалось обнаружить нужное устройство, параметры устройства выводятся в поле Свойства устройства, и начинается считывание всех поддерживаемых ВАСнет-объектов, которые затем выводятся в таблицу переменных. В таблицу выведены все основные параметры (которые потом переносятся в настройки канала), а также текущее значение тега - оно не используется, и служит для упрощения идентификации объекта пользователем.

После считывания, все переменные сгруппированы по типу объекта. При необходимости, используя Фильтр, можно скрыть не нужные объекты или его свойства. При этом переменные не удаляются из памяти - отключение фильтра вернет все переменные обратно в таблицу, повторное считывание переменных из устройства не требуется.

В связи с тем, что может возникать задача исправления имен объектов, изменения иерархии или структуры тегов, комментариев, реализован механизм экспорта и импорта тегов в Excel. При этом, если в таблице отмечены какие-либо теги, то при экспорте в Excel появляется диалоговое окно с запросом сохранения всех или только отмеченных переменных.

Таким образом итоговая последовательность рекомендуется следующая:

1. Считать теги из устройства.
2. Развернуть дерево, найти нужные объекты и отметить их. Если требуется - дополнительно опросить текущее состояние объекта используя кнопку Считать выбранные теги.
3. Если нужны только определенные свойства (в большинстве случаев достаточно только Present Value), вызвать окно фильтра, в таблице свойств отключить все свойства, и отметить нужные, нажать ОК, дерево будет отфильтровано.
4. Отметить нужные объекты флагами. При этом можно использовать выделение (в том числе групповое - по правилам принятым в Windows) и затем включить флаг используя раздел меню Выбор тегов.
5. Если требуется отредактировать имена объектов или изменить иерархию, сохраните отмеченные теги в Excel файл. Отредактируйте файл средствами Excel, сохраните и вызовите импорт из Excel в утилите импорта.

Важно! Значения импортированных из Excel файла переменных также можно опрашивать.

6. Отметьте нужные теги для импорта и нажмите кнопку Применить. Теги будут перенесены в дерево проекта.

7. После этого попробуйте выполнить опрос, загрузив проект в исполнительную систему.

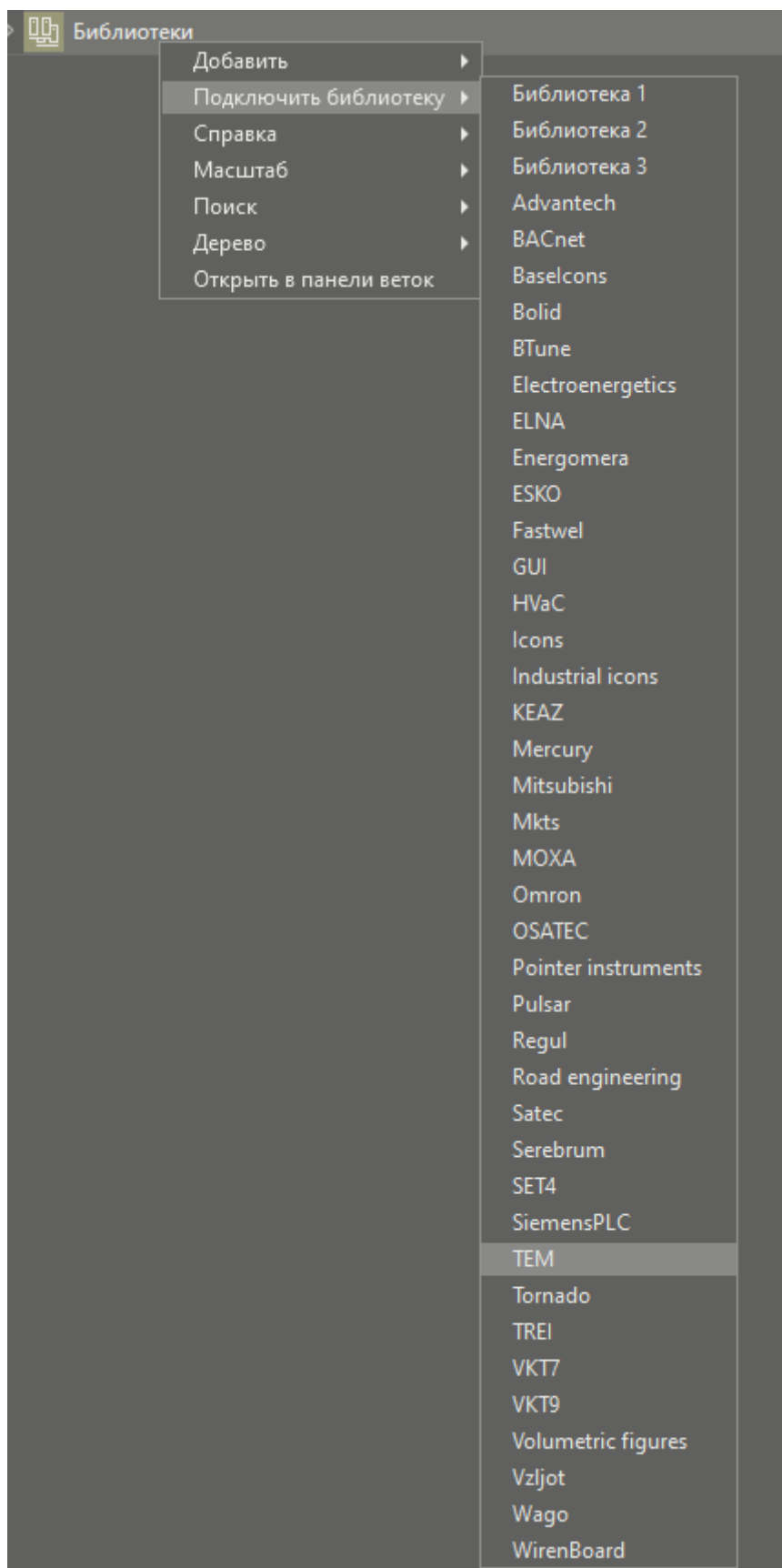
6.2.3.2.20. ТЕМ

MasterSCADA 4D поддерживает работу с тепловычислителями ТЭМ-104 и ТЭМ-106, а также с новыми версиями вычислителей серии ТЭСМА.

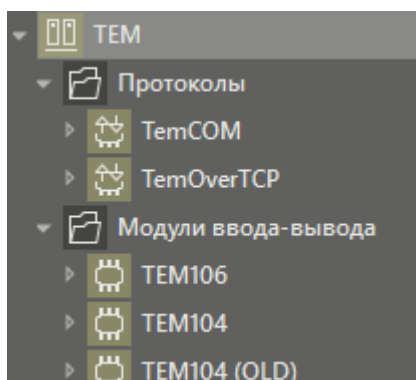
При обмене узел играет роль ведущего. Поддерживается считывание архивов из счетчика. Работа с этими устройствами возможна через последовательные интерфейсы (RS-232, RS-485, GSM), а при использовании конвертеров Ethernet-COM (Моха NPort 5150 и т.п.) также и по сетям TCP/IP (Ethernet, Wi-Fi).

Важно! По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте sales@masterscada.ru

Для получения возможности использования этих счетчиков в проекте необходимо подключить соответствующую библиотеку:

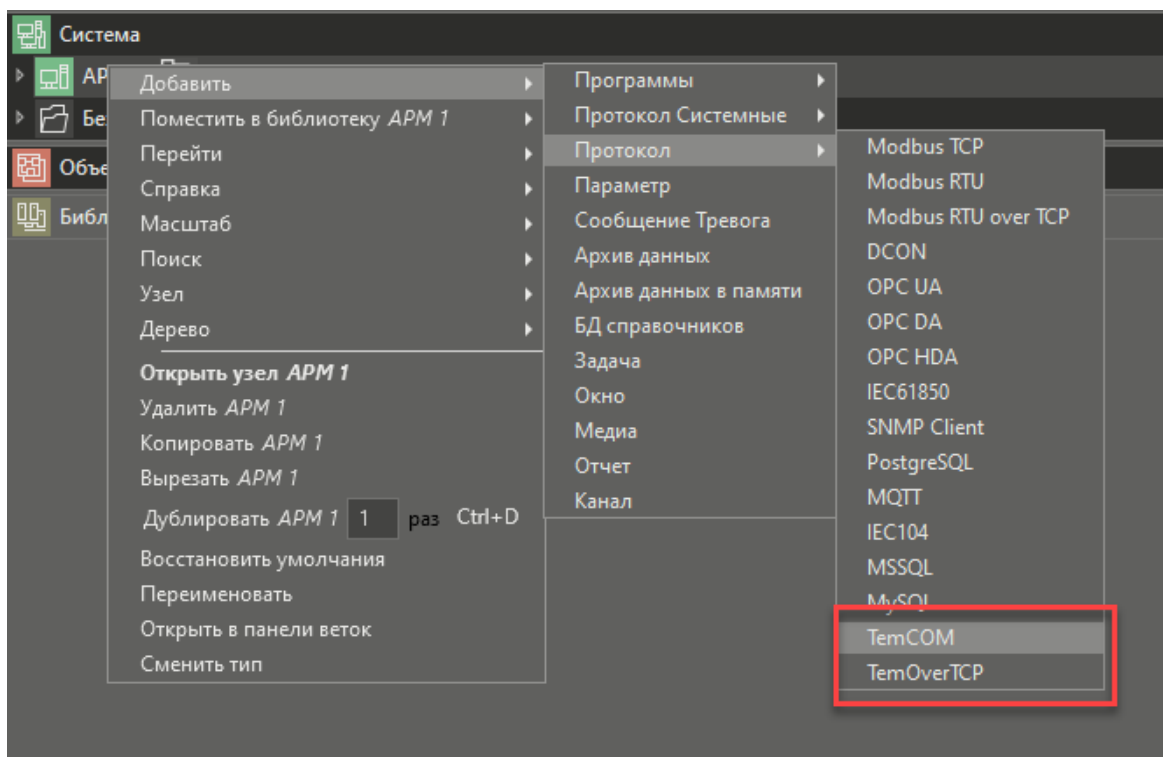


В полном дереве библиотека TEM имеет вид:



После этого в проект могут быть добавлены сначала протокол, а затем устройство, любым известным способом, например, при помощи контекстного меню узла, или с использованием контекстной панели, или путем перетаскивания элемента из библиотеки в дерево системы.

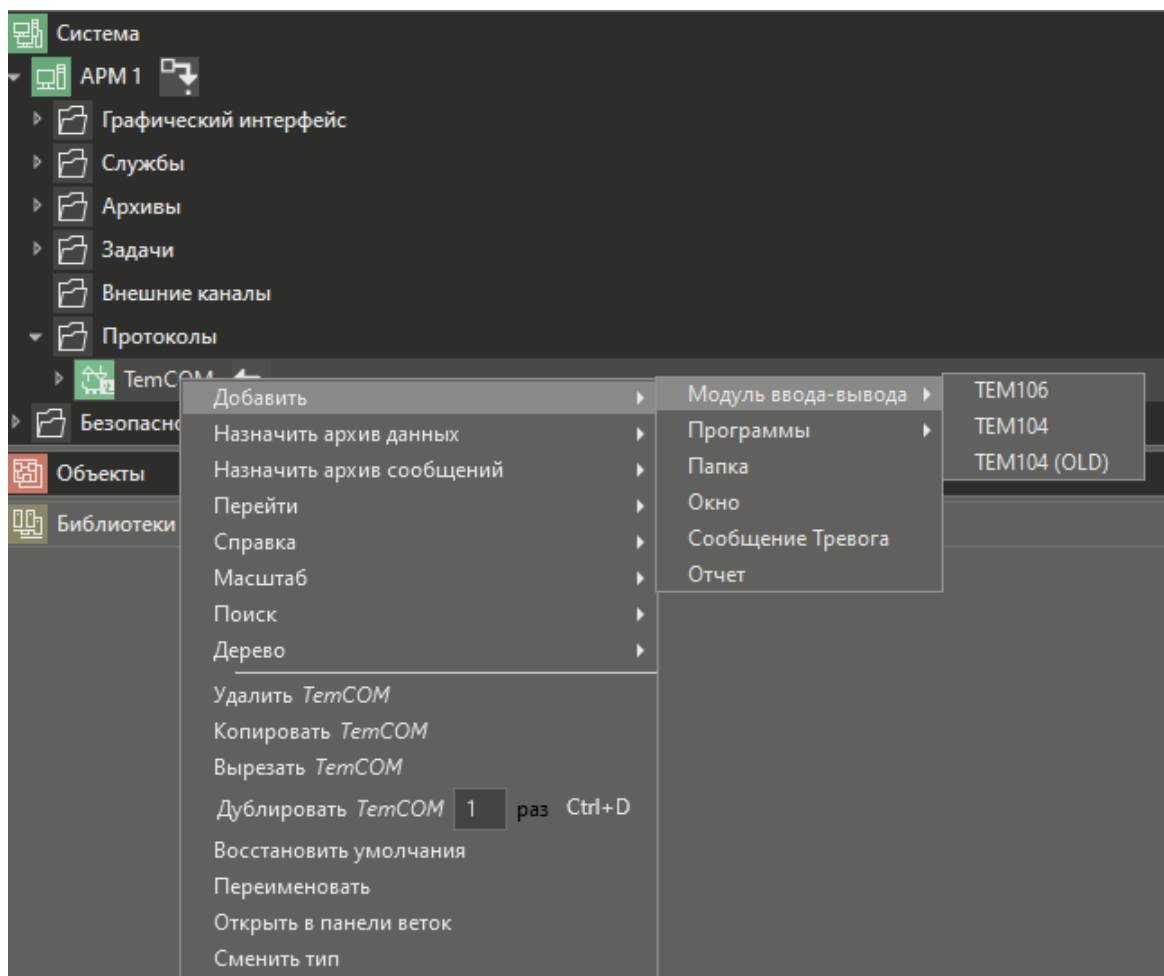
Рекомендуется добавлять элементы в проект при помощи контекстного меню:



Тип добавляемого протокола зависит от реального способа подключения счетчика:

- TEM RTU - если счетчик подключен по последовательному интерфейсу. Настройки этого протокола практически полностью соответствуют настройкам Modbus RTU. Также добавлена настройка Реинициализация узла при ошибке - если флаг установлен, то в случае отказа устройства порт будет закрыт и открыт заново.
- TEM over TCP - если счетчик подключен через конвертер Ethernet-COM. Настройки этого протокола соответствуют настройкам Modbus RTU over TCP

Затем при помощи контекстного меню протокола можно добавить модули ввода-вывода, которые уже содержит все необходимые каналы:



Каждый тепловычислитель имеет predetermined набор каналов, который характерен для него. Поддержаны наиболее используемые функции. Описание каналов устройств смотрите в разделе Модуль ТЕМ

Важно! У параметров каналов предназначенных для передачи архивных данных флаг Архивировать установлен по умолчанию. Если снять флаг, то данные от прибора будут запрашиваться, но значение в канал поступать не будет.

После добавления в дерево проекта модулей ввода-вывода, настраивается панель свойств модуля.

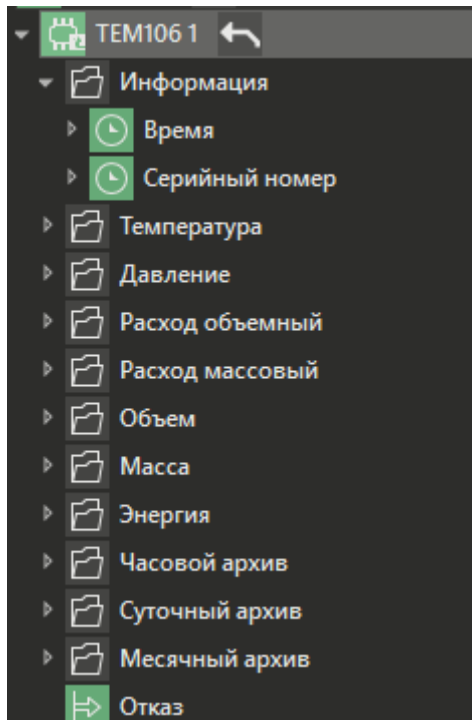
Каждый модуль ввода-вывода имеет выход Отказ, а также группы каналов, согласно его максимальной конфигурации.

Если в опросе каких-либо каналов нет необходимости, то можно либо удалить их из конфигурации элемента уже после добавления его в дерево системы, либо сначала создать экземпляр модуля в пользовательской библиотеке, затем из его конфигурации удалить незадействованные каналы, после чего использовать в проекте уже новый видоизмененный библиотечный элемент. Модуль ТЕМ

Модули добавляются в протокол ТЕМ. Соответствуют приборам ТЭМ-104 и ТЭМ-106 разработки группы компаний "ТЭМ".

Счетчики имеют одинаковый состав групп, но ТЭМ-104 позволяет подключать меньшее количество датчиков расхода - 4 (против 6 у ТЭМ-106), поэтому он имеет меньшее количество каналов в группах связанных с выводом расхода и энергий.

Каналы счетчиков объединены в группы по их назначению.



В таблице описано назначение каналов счетчика:

Название	Назначение
Категория Информация	
Время	Выдается время, установленное в приборе
Серийный номер	Выдается серийный номер счетчика
Категория Температура	
Канал 1...Канал 6	Измеренные значения температуры, градусы Цельсия
Категория Давление	

Название	Назначение
Канал 1...Канал 6	Измеренные значения давления, МПа
Категория Расход объемный	
Канал 1...Канал 4 (Канал 6)	Текущие значения объемного расхода, м ³ /ч
Категория Расход массовый	
Канал 1...Канал 4 (Канал 6)	Текущие значения массового расхода, т/ч
Категория Объем	
Канал 1...Канал 4 (Канал 6)	Накопленные значения объема энергоносителя, м ³
Категория Масса	
Канал 1...Канал 4 (Канал 6)	Накопленные значения массы энергоносителя, тонны
Категория Энергия	
Канал 1...Канал 4 (Канал 6)	Накопленные значения энергии, МВт*ч
Категория Часовой/Суточный/Месячный архив	
Объем1-Объем (Объем 6)	4 Объем теплоносителя протекшего за интервал времени по каждому каналу, м ³ .
Масса 1-Масса (Масса 6)	4 Масса теплоносителя протекшего за интервал времени по каждому каналу, тонны.

Название	Назначение
Энергия 1 Энергия 4 (Энергия 6)	- Отпущенная энергия за интервал времени по каждому каналу, МВт*ч.
Температура 1 Температура 6	- Средняя температура теплоносителя за интервал времени по каждому каналу, градус Цельсия.
Давление 1 Давление 6	- Среднее значение давления теплоносителя за интервал времени по каждому каналу, МПа.
Расход 1- Расход 4 (Расход 6)	Среднее значения расхода теплоносителя за интервал времени по каждому каналу, т/ч.
Pointer	Сервисный канал. Указатель на последнюю считанную запись. Снимать флаг Архивировать у данного канала не рекомендуется, в противном случае архивные значения из счетчика будут считываться постоянно сначала.

Порядок работы с архивами прибора

Для всех каналов группы Часовой/Суточный/Месячный архив установлен флаг Архивировать.

При старте, исполнительная система начинает считывать архивные записи вглубь на, заданное в настройках устройства, количество записей (настройки Глубина считывания часового, суточного, месячного архива). При этом каждый цикл считывается не большее количество записей, чем задано в поле Количество считываемых записей архива за один цикл. После считывания всех записей исполнительная система считывает только новые записи, проверяя их наличие каждый цикл опроса.

6.2.3.2.20.1. СВОЙСТВА МОДУЛЕЙ ТЕМ

Панель свойств модуля ТЕМ имеет вид:

TEM104 1

🔍 🏠
☰ 🏠 🔑

Общие	
Имя	TEM104 1
Полное имя	Система.АРМ 1.Протоколы.TemCOM.TEM104
Метки	
Комментарий	ТЭМ104
Служебные	
Задача	
Отношения	
Масштабирование	
Настройки	
Адрес	1
Глубина часового архива	0
Глубина суточного архива	0
Глубина месячного архива	0
Количество архивных записей за цикл	5

Название	Описание
Категория Общие	Как правило, для данного элемента эта категория не требует настройки. Подробное описание данной категории приведено в соответствующем разделе.
Категория Задача	Задаёт способ опроса устройства. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи.
Категория Масштабирование	Если модуль ввода-вывода не имеет встроенного масштабирования, то при помощи данной категории может быть задано соответствие между значениями в модуле и реальными значениями. Подробное описание данной категории приведено в соответствующем разделе.
Категория Настройки	

Название	Описание
Адрес	Указывает адрес счетчика в сети.
Глубина часового архива	Определяет количество считываемых при старте исполнительной системы записей часового архива.
Глубина суточного архива	Определяет количество считываемых при старте исполнительной системы записей суточного архива.
Глубина месячного архива	Определяет количество считываемых при старте исполнительной системы записей месячного архива.
Количество архивных записей за цикл	Определяет количество считываемых каждый цикл опроса записей из архива счетчика.

6.2.3.2.21.DLMS

MasterSCADA 4D поддерживает работу по протоколу Dlms.

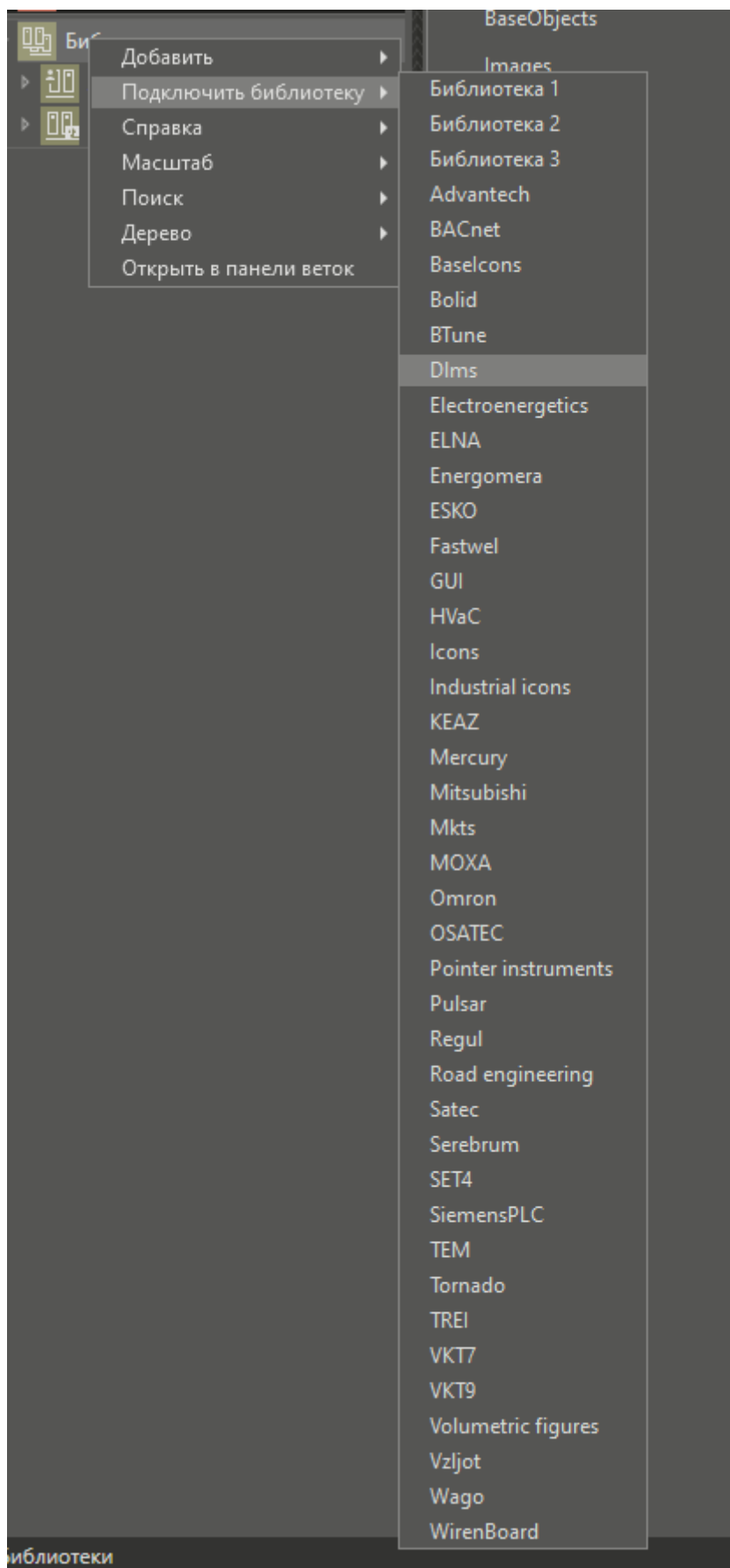
Протокол DLMS – открытый объектный протокол для обеспечения сетевого обмена с различными приборами учета - электричество, теплосчетчики, вода, газ. Протокол может работать через последовательные интерфейсы (RS-232, RS-485, GSM), а при использовании конвертеров Ethernet-COM (Моха NPort 5150 и т.п.) также и по сетям TCP/IP (Ethernet, Wi-Fi). Протокол имеет механизмы многоуровневой аутентификации и шифрования, поддерживает передачу архивов и групповой опрос переменных. Также протокол обеспечивает возможность получения полного списка объектов данного устройства.

Основными переменными протокола являются объекты. Объект DLMS - это различные варианты структур данных, предназначенных для передачи параметров устройства (текущие значения, архивы, состояния счетчика, параметры подключения). Каждый тип объекта имеет различную структуру данных, при этом некоторые атрибуты объекта могут иметь динамический тип данных, устанавливаемый производителем устройства. Подробное описание поддерживаемых объектов смотрите в разделе Поддерживаемые объекты протокола Dlms

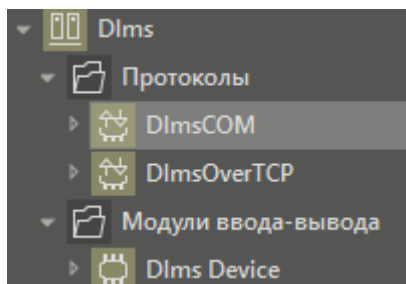
При обмене узел играет роль ведущего.

Важно! По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте sales@masterscada.ru

Для получения возможности использования протокола в проекте необходимо подключить соответствующую библиотеку:

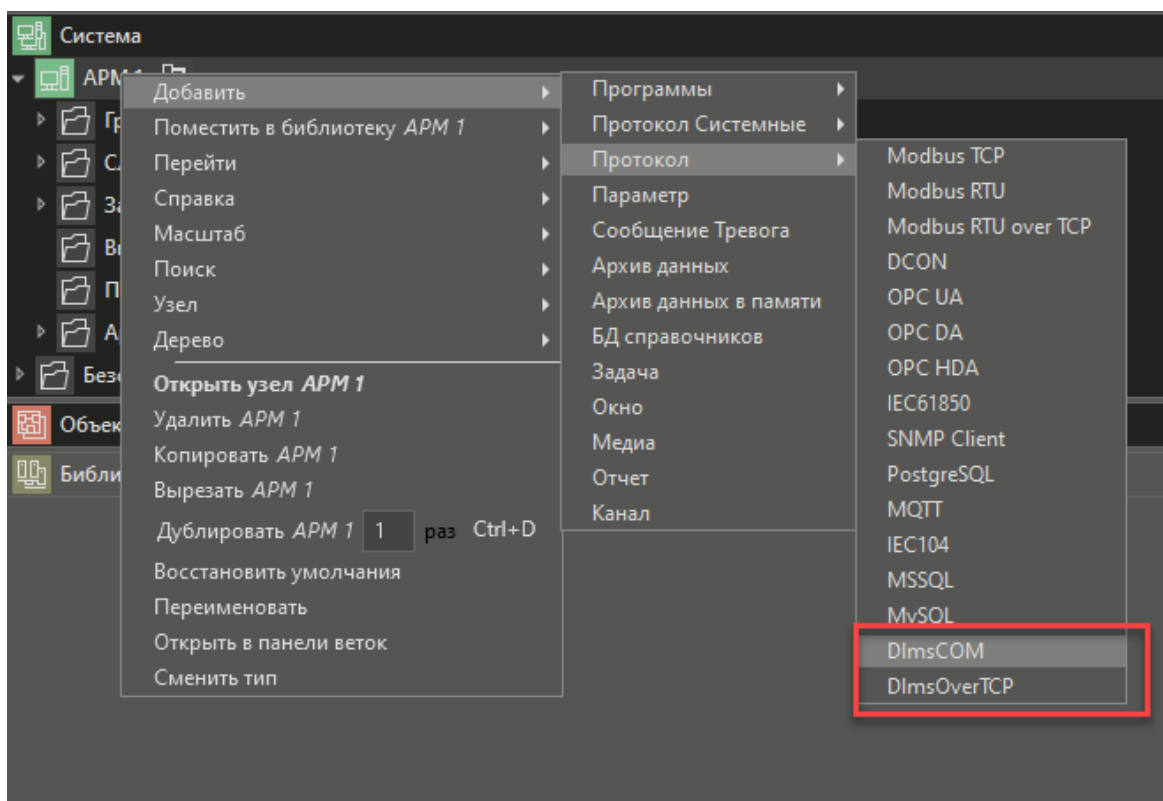


В полном дереве библиотека Dlms имеет вид:



После этого в проект могут быть добавлены сначала протокол, а затем устройство, любым известным способом, например, при помощи контекстного меню узла, или с использованием контекстной панели, или путем перетаскивания элемента из библиотеки в дерево системы.

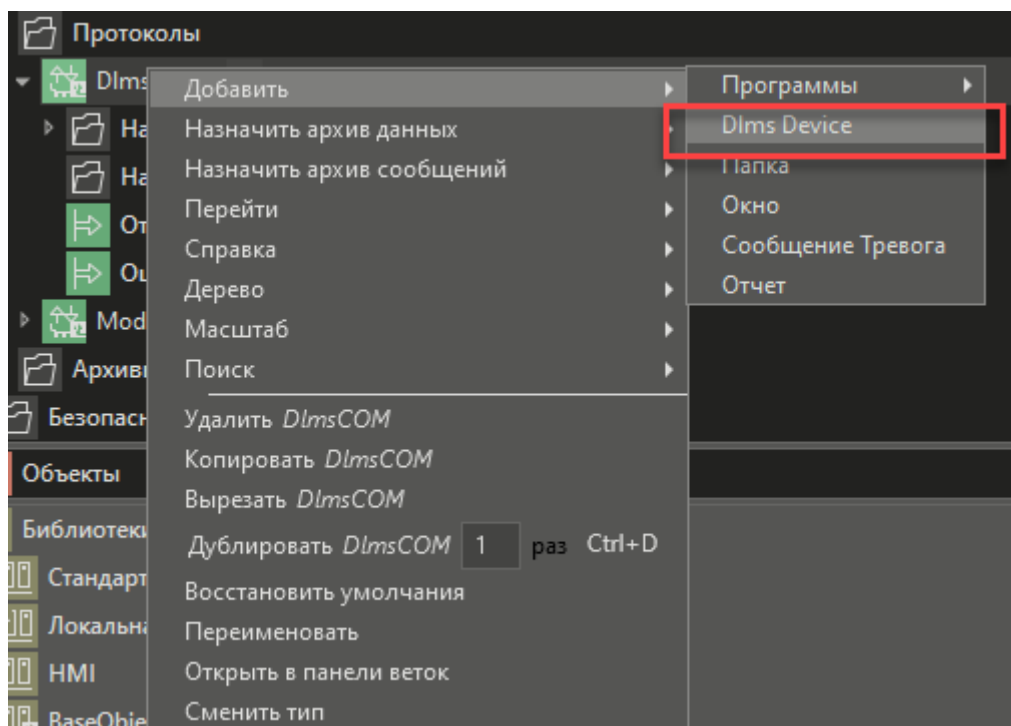
Рекомендуется добавлять элементы в проект при помощи контекстного меню:



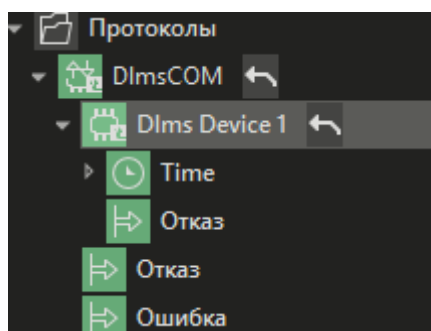
Тип добавляемого протокола зависит от реального способа подключения устройств:

- DImsCOM - если счетчик подключен по последовательному интерфейсу. Настройки этого протокола практически полностью соответствуют настройкам Modbus RTU. Также добавлена настройка
- DImsOverTCP - если счетчик подключен через конвертер Ethernet-COM. Настройки этого протокола соответствуют настройкам Modbus RTU over TCP

Затем при помощи контекстного меню протокола можно добавить устройство:



В дереве системы в упрощенном дереве элемент имеет вид:



Устройство имеет специальный канал TIME, в котором отображается время, установленное в приборе.

Добавить канал в устройство можно при помощи контекстного меню вручную, либо через диалог импорта (рекомендуется).

Важно! Для использования диалога импорта устройство должно быть подключено непосредственно к компьютеру. Т.е. если планируется подключать устройство к контроллеру, то для формирования списка каналов нужно подключить устройство к компьютеру, считать теги, выбрать нужные или сохранить в Excel. Потом уже подключить устройство к контроллеру и вставить туда устройство с готовым списком каналов или импортировать их из Excel файла.

6.2.3.2.21.1. ПОДДЕРЖИВАЕМЫЕ ОБЪЕКТЫ ПРОТОКОЛА DLMS

Основными переменными протокола являются объекты. Объект DLMS - это различные варианты структур данных, предназначенных для передачи параметров устройства

(текущие значения, архивы, состояния счетчика, параметры подключения). Каждый тип объекта имеет различную структуру данных, при этом некоторые атрибуты объекта могут иметь динамический тип данных, устанавливаемый производителем устройства.

В настоящий момент драйвером для опроса поддерживаются следующие объекты DLMS:

Объект	Номер	Описание
Data	1	Объект данных. Предназначен для хранения не именованных величин различных типов.
Register	3	Регистр данных. Имеет логическое имя, значение, масштаб и единицы измерения. Предназначен для хранения именованных величин различных типов. Это основной объект, в котором хранятся большинство текущих параметров устройства (напряжения, токи, мощности, энергии).
Extended Register	4	Расширенный регистр. Аналогичен регистру, но имеет дополнительные атрибуты - статус и время фиксации. Предназначен для хранения именованных величин различных типов, зафиксированных в определенный момент времени (например максимум и минимум мощности).
Demand Register	5	Регистр усреднения. Класс предназначен для фиксации среднего значения величины методом скользящего окна за определенный период времени. Данный класс может быть использован для вычисления и хранения пиковых значений мощности, а также средних значений напряжения (тока) за интервал измерения
Profile Generic	7	Универсальный профиль. Данный интерфейсный класс предназначен для хранения и доступа к группам данных - так называемым «захваченным объектам». Захваченными объектами являются атрибуты или элементы атрибутов объектов. Захваченные объекты собираются периодически (профиль нагрузки), либо при наступлении какого-то условия (журналы событий). Универсальный профиль - основной способ получения архивных значений прибора. В настоящий момент поддерживаются только профили нагрузок - журналы событий не поддерживаются.
Clock	8	Время. Объект предназначенный для хранения времени, а также осуществления автоматического перевода стрелок на зимнее/летнее время. В драйвере объект Clock добавлен как статичная не

Объект	Номер	Описание
		удаляемая сервисная переменная, которая настроена на вывод текущего времени счетчика.

DATA

Для идентификации каждый объект имеет специальный идентификатор - OBIS код. OBIS код - 6-значный идентификатор, в качестве разделителя используется точка - A.B.C.D.E.F. Каждый байт идентификатора может определять свойство опрашиваемого объекта.

OBIS байт	Описание
A	Вид учета (0 - абстрактные объекты, 1 - электричество, 4,5,6 - теплосчетчик, 7 - газ, 8,9 - вода).
B	Номер канала
C	Виде параметра. Зависит от параметра A. Например для A=1, C=21 - активная мощность по фазе 1.
D	Способ обработки данных. Зависит от параметра A. Например при A=1, D=7 - мгновенное значение, D = 8 - интеграл (накопленное значение) с начала эксплуатации.
E	Индекс тарифа или гармоники (для A=1)
F	Индекс архиватора. Для текущих - 255.

Пример OBIS кода - 1.0.31.7.0.255, параметр электроэнергии, ток по фазе L1, текущее значение, нулевая гармоника (сумма гармоник). Используя OBIS кодирование можно определить предназначение основных объектов устройства, для упрощения этой задачи устройство имеет диалог импорта.

Также в OBIS введены несколько идентификаторов группы C и D определяющие специфические объекты (не предусмотренные базовым стандартом):

- Код C=94 - определяет данных как специфический объект для определенной страны. Код страны определяется параметром D и в большинстве случаев

совпадает с телефонным кодом страны (например Россия - код 7). Остальные OBIS байты определяют сам параметр.

- Коды C=128-199, 244 - коды для объектов определенным производителем устройства. Остальные OBIS байты определяют сам параметр.

ОБЪЕКТЫ DLMS

Enter topic text here.

Объект Register

Объект Extended Register

Объект Demand Register

Объект Profile Generic

ОБЪЕКТ DATA

Data - объект данных протокола Dlms. Предназначен для хранения не именованных величин различных типов.

Объект имеет всего 2 атрибута:

Номер	Атрибут	Тип данных
1	Логическое имя объекта (OBIS код)	Строка
2	Значение	Любой

Для опроса в режиме исполнения доступен только атрибут №2.

ОБЪЕКТ REGISTER

Объект Register - регистр данных протокола Dlms, основной способ хранения измеряемых и вычисляемых параметров устройства.

Номер	Атрибут	Тип данных
1	Логическое имя объекта (OBIS код)	Строка

2	Значение	Любой, кроме массивов, структур и времени.
3	Масштаб и единица измерения	16 битное беззнаковое целое

Масштаб и единица измерения - поле состоящее из двух байт. В старшем байте хранится масштаб в виде показателя степени (-128..127), а в младшем код единицы измерения. Утилита импорта может определять масштаб и выводить единицу измерения. В режиме исполнения к данному параметру обращение не производится (масштаб хранится в настройках тега).

Для опроса в режиме исполнения доступен только атрибут №2.

ОБЪЕКТ EXTENDED REGISTER

Объект Extended Register - расширенный регистр протокола DImS. Объект имеет дополнительные атрибуты - статус и время фиксации. Предназначен для хранения именованных величин различных типов, зафиксированных в определенный момент времени (например максимум и минимум мощности).

Номер	Атрибут	Тип данных
1	Логическое имя объекта	Строка
2	Зафиксированное значение	Любой, кроме массивов, структур и времени
3	Масштаб и ед. измерения	Короткий целый
4	Статус	Строка, число, вещественное число - на выбор производителя
5	Время фиксации	Время

Масштаб и единица измерения - поле состоящее из двух байт. В старшем байте хранится масштаб в виде показателя степени (-128..127), а в младшем код единицы измерения. Утилита импорта может определять масштаб и выводить единицу измерения.

Для опроса в режиме исполнения доступны все атрибуты, но имеет смысл опрашивать только зафиксированное значение и время.

ОБЪЕКТ DEMAND REGISTER

Объект Demand Register - объект усреднения протокола DImS. Класс предназначен для фиксации среднего значения величины методом скользящего окна за определенный период времени. Данный класс может быть использован для вычисления и хранения пиковых значений мощности, а также средних значений напряжения (тока) за интервал измерения.

Номер	Атрибут	Тип данных
1	Логическое имя объекта	Строка
2	Текущее среднее значение	Любой, кроме массивов, структур и времени
3	Последнее среднее значение	Любой, кроме массивов, структур и времени
4	Масштаб и ед. измерения	Короткий целый
5	Статус	Строка, число, вещественное число - на выбор производителя
6	Время фиксации	Время
7	Время старта	Время
8	Длительность периода усреднения	32 беззнаковое целое
9	Количество периодов	16 битное беззнаковое целое

Масштаб и единица измерения - поле состоящее из двух байт. В старшем байте хранится масштаб в виде показателя степени (-128..127), а в младшем код единицы измерения. Утилита импорта может определять масштаб и выводить единицу измерения. В режиме исполнения к данному параметру обращение не производится (масштаб хранится в настройках тега).

ОБЪЕКТ PROFILE GENERIC

Объект Profile Generic - универсальный профиль протокола DImS. Данный интерфейсный класс предназначен для хранения и доступа к группам данных - так называемым «захваченным объектам». Захваченными объектами являются атрибуты или элементы атрибутов объектов. Захваченные объекты собираются периодически (профиль нагрузки), либо при наступлении какого-то условия (журналы событий). Универсальный профиль - основной способ получения архивных значений прибора. В настоящий момент поддерживаются только профили нагрузок - журналы событий не поддерживаются.

Номер	Атрибут	Тэг типа данных
1	Логическое имя объекта	Строка
2	Буфер данных	Массив или упакованный массив
3	Список захватываемых объектов	Массив
4	Период захвата	32 битное беззнаковое целое
5	Метод сортировки	Перечисление
6	Занятых записей	32 битное беззнаковое целое
7	Всего записей	32 битное беззнаковое целое

Основное назначение Profile Generic- получения архивных значений (профиль мощности, месячные и посуточные срезы).

При старте драйвер производит инициализацию профиля мощности - определяет тип архива (часовой, получасовой, суточный или месячный), а также получает список захватываемых объектов. После этого проверяет на совпадение списка захватываемых объектов с параметрами тегов Capture Object OBIS - если среди захватываемых объектов не обнаружен заданный у тега профиля, то опрос производится не будет. Также в списке захватываемых объектов обязательно должен присутствовать параметр "Время счетчика" (OBIS - 0.0.1.0.0.255). В противном случае профиль мощности также не будет опрашиваться.

При старте драйвер определяет текущее время счетчика, смещает стартовое время опроса профиля (глубину считывания) на заданное в настройках устройства, и начинает поочередное считывание записей (количество считываемых записей также определяется настройкой). По завершению считывания в каждом цикле опроса происходит сравнение текущего времени, с последней считанной записью, и если прошло времени более чем 1

период захвата (полчаса, час, сутки, месяц - в зависимости от типа профиля) происходит считывание новых записей профиля.

Важно! Поддерживаются только профили стандартных размерностей времени (полчаса, час, день, месяц) - профили с другим временным шагом не поддерживаются.

КОДИРОВАНИЕ OBIS

Группа А

Группа В

Группа С

Группа D

Группа E

Группа F

ГРУППА А

Группа А из OBIS кода протокола DImS определяет тип энергии.

Значение А	Вид учета
0	Абстрактные параметры. Параметры не связанные с видом энергии - время счетчика, настройки, списки объектов
1	Электроэнергия
4	Расчеты за тепловую энергию
5,6	Тепловая энергия
7	Газ
8	Холодная вода
9	Горячая вода
15	Другие виды энергий
Остальные параметры	Зарезервировано.

Как правило прибор учета имеет группу параметров А=0 (абстрактные параметры) и группу измеряемых энергий.

ГРУППА В

Группа определяет номер канала.

Значение В	Вид учета
0	Канал не указан. Информация о канале не важна.
1..64	Канал 1-64
65..127	Специальные вспомогательные коды. В этом случае весь OBIS код является специфическим (определенным страной или производителем).

Значение B	Вид учета
128..199	Коды производителя устройства.

ГРУППА С

Группа С определяет вид параметра. Она напрямую зависит от типа системы учета - параметра А (электроэнергия, тепловая энергия и т.д.). Каждый вид энергии и абстрактные параметры имеют собственные таблицы группы С.

Ниже приведена таблица для электроэнергии (как наиболее распространенной системой учета протокола DLMS).

Значение С				Описание
Сумма	L1	L2	L3	
0				Объекты общего назначения
1	21	41	61	Положительная активная мощность (QI+QIV)
2	22	42	62	Отрицательная активная мощность (QII+QIII)
3	23	43	63	Положительная реактивная мощность (QI+QII)
4	24	44	64	Отрицательная реактивная мощность (QIII+QIV)
5	25	45	65	Реактивная мощность QI
6	26	46	66	Реактивная мощность QII
7	27	47	67	Реактивная мощность QIII
8	28	48	68	Реактивная мощность QIV

Значение С				Описание
Сумма	L1	L2	L3	
9	29	49	69	Положительная полная мощность (QI+QIV)
10	30	50	70	Отрицательная полная мощность (QII+QIII)
11	31	51	71	Ток: С=11 – любой фазы, иначе - соответствующей
12	32	52	72	Напряжение: С=12 – любой фазы
13	33	53	73	Фактор мощности со знаком («+» - импорт, «-» - экспорт.)
14	34	54	74	Частота сети
15	35	55	75	Модуль активной мощности для измерения гармоник ($\text{abs}(QI+QIV)+\text{abs}(QII+QIII)$)
16	36	56	76	Сетевая мощность ($\text{abs}(QI+QIV)-\text{abs}(QII+QIII)$)
17	37	57	77	Активная мощность QI
18	38	58	78	Активная мощность QII
19	39	59	79	Активная мощность QIII
20	40	60	80	Активная мощность QIV
81				Угловые измерения
82				Безразмерные величины (счет импульсов)
83				Потери в трансформаторах и линиях передачи

Значение С				Описание
Сумма	L1	L2	L3	
84	85	86	87	Фактор мощности без знака
88				Удельные потери в линиях (А ² часы)
89				Удельные потери в железе (В ² часы)
90				Суммарный ток всех фаз
91				Ток нейтрали
92				Напряжение нейтрали
93				Коды, устанавливаемые ассоциациями
94				Код, устанавливаемыми странами. Идентификатор страны определяется параметром D, и в большинстве случаев равен телефонному коду (в России - код 7).
96				Электрические константы счетчиков
97				Объекты регистров ошибок
98				Списки объектов (архивы)
99				Профили электрических величин (профили мощности, архивы).
100..127				Резерв
128..199, 240				Коды устанавливаемые производителями

ГРУППА D

Группа D определяет способ обработки данных. Группа D также напрямую зависит от типа системы учета - параметра A (электроэнергия, тепловая энергия и т.д.). Каждый вид энергии и абстрактные параметры имеют собственные таблицы группы D.

Ниже приведена таблица для электроэнергии (как наиболее распространенной системой учета протокола DLMS).

Значение D	Описание
0	Среднее значение за расчетный период (с момента последнего сброса)
1	Общий минимум 1 (с начала эксплуатации)
2	Общий максимум 1 (с начала эксплуатации)
3	Минимум 1 (в течение расчетного периода)
4	Текущее среднее 1 (из регистров усреднения)
5	Последнее среднее 1 (из регистров усреднения)
6	Максимум 1 (в течение расчетного периода)
7	Мгновенное значение
8	Интеграл с начала эксплуатации до текущего момента (накопленное значение с начала эксплуатации)
9	Интеграл с начала текущего расчетного периода (накопленное значение например за месяц)
10	Интеграл превышения величиной установленного порога
11	Общий минимум 2
12	Общий максимум 2
13	Минимум 2

Значение D	Описание
14	Текущее среднее 2
15	Последнее среднее 2
16	Максимум 2
17	Интеграл с начала эксплуатации до конца последнего закончившегося периода записи с периодом 1
18	Интеграл с начала эксплуатации до конца последнего закончившегося периода записи с периодом 2
19	Интеграл с начала текущего расчетного периода до конца последнего периода записи с периодом 1
20	Интеграл с начала текущего расчетного периода до конца последнего периода записи с периодом 2
21	Общий минимум 3
22	Общий максимум 3
23	Минимум 3
24	Текущее среднее 3
25	Последнее среднее 3
26	Максимум 3
27	Текущее среднее 5
28	Текущее среднее 6
29	Интеграл от начала текущего периода записи профиля с периодом 1 до текущего момента

Значение D	Описание
30	Интеграл от начала текущего периода записи профиля с периодом 2 до текущего момента
31	Порог нижнего предела (провала)
32	Счетчик провалов
33	Продолжительность провала
34	Величина провала
52	Минимум для периода записи 2
53	Максимум для периода записи 1
54	Максимум для периода записи 2
55	Среднее за тест
58	Интеграл за время теста
128...254	Коды, определяемые производителями

ГРУППА E

Для объектов типа Энергия, группа E протокола Dlms указывает номер тарифа. Для объектов типа "мгновенное значение", "среднее значение", "ток", "напряжение", "мощность" группа определяет номер гармоники.

- 0 - сумма тарифов или гармоник.
- 1..63 - номер тарифа.
- 1..127 - номер гармоники.
- 128...254 - коды, определяемые производителем.

При параметре C=81, группа E определяет углы между фазами напряжения и тока:

Angle	U(L1)	U(L2)	U(L3)	I(L1)	I(L2)	I(L3)	I(L0)
U(L1)	(0)	1	2	4	5	6	7
U(L2)	10	(11)	12	14	15	16	17
U(L3)	20	21	(22)	24	25	26	27
I(L1)	40	41	42	(44)	45	46	47
I(L2)	50	51	52	54	(55)	56	57
I(L3)	60	61	62	64	65	(66)	67
I(L0)	70	71	72	74	75	76	(77)

ГРУППА F

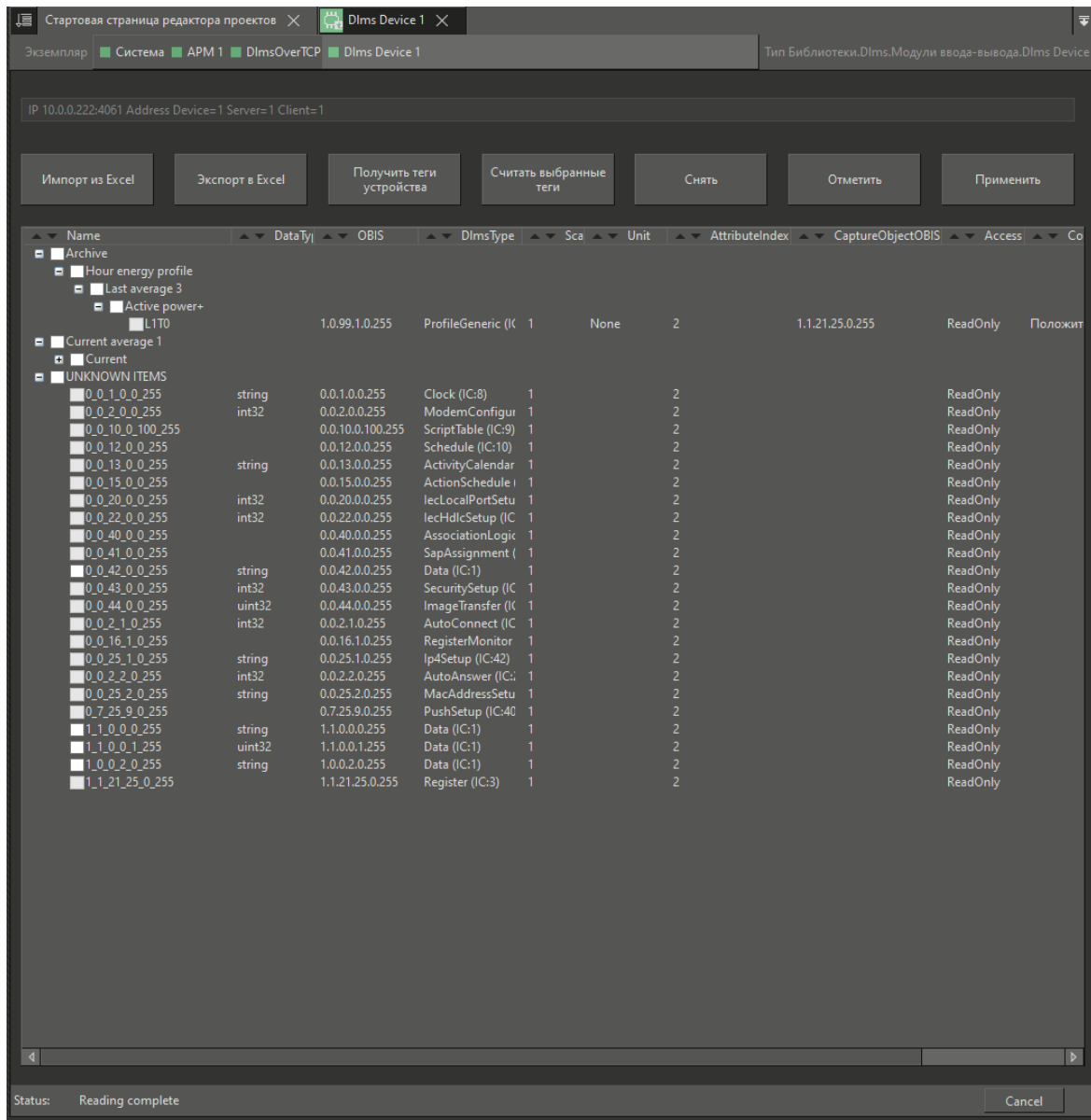
Группа F используется для указания исторических (архивных) данных. Для обозначения текущего используется «255». Для обозначения архивных данных используется один из двух способов:

- Относительно текущего расчетного периода при помощи счетчика расчетных периодов по модулю 100 или по модулю 12, при этом в поле «F» указывается значение счетчика расчетных периодов
- При помощи создания профиля с необходимой глубиной и периодом записи. Счетчик периодов при этом может не использоваться, его роль выполняет метка времени, являющаяся частью каждой записи.

6.2.3.2.21.2. ДИАЛОГ ИМПОРТА КАНАЛОВ DLMS

Для автоматического добавления каналов и поиска устройств для работы по устройству Dlms Device нужно дважды нажать левую кнопку мыши по протоколу в дереве. При этом откроется вкладка, в которой можно выполнить поиск устройства и импорт каналов.

Вид вкладки:



В верхней части отображается строка, в которой указаны настройки устройства заданные в панели свойств (значения свойств обновляется при переоткрытии вкладки).

Основную часть вкладки занимает таблица, в которой будут отображаться найденные объекты. Флагами можно отметить необходимые каналы для добавления в дерево проекта

Описание элементов управления:

Кнопка	Назначение
Импорт из Excel	Открывает диалоговое окно, в котором можно выбрать файл Excel, в котором хранится описание каналов.
Экспорт в Excel	Позволяет сохранить в файл выбранную конфигурацию

Кнопка	Назначение
Получить теги устройства	Подключается к устройству и считывает данные из него
Считать выбранные теги	Считывает значение из устройства
Снять	Снимает выделение с выбранного объекта
Отметить	Отмечает выделенный объект
Применить	Добавляет отмеченные элементы в дерево.

Важно! Для использования диалога импорта устройство должно быть подключено непосредственно к компьютеру. Т.е. если планируется подключать устройство к контроллеру, то для формирования списка каналов нужно подключить устройство к компьютеру, считать теги, выбрать нужные или сохранить в Excel. Потом уже подключить устройство к контроллеру и вставить туда устройство с готовым списком каналов или импортировать их из Excel файла.

СТРУКТУРА EXCEL-ФАЙЛА

Для корректной работы в диалоге импорта устройства DImS Device Excel-файл должен иметь определенную структуру:

Имя столбца	Описание
Name	Имя переменной. Поскольку утилита отображает имена в иерархическом виде, то в качестве разделителя уровня используется точка.
DataType	Тип данных в OPC сервере. Возможные варианты - double, int32, uint32, string, datetime и т.д.
Access	Тип доступа. На данный момент - ReadOnly, т.е. все теги используются только в режиме чтения
Comment	Комментарий к тегу.

Имя столбца	Описание
OBIS	OBIS идентификатор тега.
DlmsType	Тип объекта Dlms. Возможные варианты - Data, Register, Extended Register, Demand Register, Profile Generic.
Scale	Множитель - на это значение умножается полученное из прибора значение. Поле носит информационный характер и не импортируется.
Unit	Тип измеряемого параметра
AttributeIndex	Индекс атрибута объекта. Для Data и Register - всегда 2 (может не указываться)
CaptureObjectOBIS	Для Profile Generic - объект захвата.
Value	Считанное из прибора значение. Поле носит информационный характер и не импортируется.

Важно! Привязка к столбцам в утилите производится по имени столбца - порядок столбцов не важен. Имена не должны содержать пробелов и полностью совпадать с указанными. Также вы можете взять любой из поставляемых Excel файлов, и сделать собственный на его основе.

6.2.3.2.21.3. РЕКОМЕНДАЦИИ ПО НАСТРОЙКЕ ПРОТОКОЛА DLMS

Ниже приведен рекомендуемый алгоритм настройки прибора для опроса его драйвером протокола DLMS.

1. Добавить протокол DLMS нужного типа. Задать его параметры (порт, скорость, четность).
2. Добавить устройство в протокол. Задать его параметры - адрес в сети (как правило последние 5 цифр серийного номера), параметры безопасности, пароль, адрес сервера. Эту информацию следует взять из документации к прибору или уточнить у производителя.
3. Загрузить проект в исполнительную систему. В приборе есть сервисный, не удаляемый тег времени, который стандартен и обязателен для всех приборов. Если опрос пойдет успешно, и отобразится значение времени - значит связь с прибором установлена и можно добавлять каналы. Если связь не устанавливается, то следует перепроверить соединение,

настройки узла и устройства. Также можно попробовать опросить сторонним DLMS-клиентом - например, Gurux DLMS Director.

4. Открыть диалог импорта устройства.

5. Если к прибору есть готовый Excel-файл, то нажать на кнопку Импорт из Excel и выбрать его.

6. Если готового файла нет, то выполнить считывание тегов из устройства - кнопка Получить теги устройства. После считывания из прибора, рекомендуется сразу сохранить файл в Excel - чтобы более не считывать его повторно - Экспорт в Excel.

7. Отметить нужные каналы. Если требуется - прочитать их текущие значения, для более точной идентификации.

8. Отмеченные каналы, рекомендуется сохранить в Excel-файл - для повторного использования именно нужных для данной задачи тегов.

9. Нажать на кнопку Применить. Каналы добавятся в дерево.

10. Загрузить проект в исполнительную систему - убедится что все каналы считываются. Если какие то из тегов имеют признак качества Uncertain, то это может говорить о несоответствии типа данных (например тип тега имеет тип double, а полученное значение - datetime). В этом случае следует обратиться в техподдержку.

11. Включить режим группового опроса. После чего снова загрузить проект в исполнительную систему. Если опроса нет, и все каналы BAD, значит прибор не поддерживает групповой опрос, и его нужно выключить. Если прибор не поддерживает групповой опрос, это также выдается в лог исполнительной системы.

6.2.3.2.21.4. СВОЙСТВА DLMS DEVICE

Вид панели свойств Dlms Device:

DIms Device 1

🔍 📄
☰ 🗑️ 🔑

Общие ▲

Имя	DIms Device 1
Полное имя	Система..АРМ 1.Протоколы.DImsCOM.DIms Dev
Метки	
Комментарий	

Служебные ▼

Задача ▼

Отношения ▼

Масштабирование ▼

Настройки ▲

ServerAddress	1
ClientAddress	1
Режим Wrapper	<input type="checkbox"/>
Адрес устройства	3
Пароль	33333333
Групповой опрос	<input type="checkbox"/>
Количество объектов в запросе	0
Глубина получасового архива	0
Глубина часового архива	0
Глубина суточного архива	0
Глубина месячного архива	0
Количество архивных записей за цикл	5
Security	None
Authentication	Low
SystemTitle	4775727578313233
AuthenticationKey	D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF
BlockCipherKey	000102030405060708090A0B0C0D0E0F
DedicatedKey	00112233445566778899AABBCCDDEEFF

Настройки каналов ▼

Название	Назначение
ServerAddress	Адрес DLMS сервера в устройстве. Зависит от настроек устройства
ClientAddress	Адрес DLMS клиента (драйвера). Зависит от настроек устройства
Режим Wrapper	Определяет тип интерфейса связи. Если флаг установлен, то используется Wrapper, если флаг снят, то HDLC
Адрес устройства	Физический адрес устройства на шине. Как правило адрес - последние 5 цифр серийного номера устройства (без нулей в начале).
Пароль	Пароль подключения. В ряде режимов аутентификации может быть не доступен. Определяется настройками счетчика.
Групповой опрос	Групповой опрос счетчика, опрос нескольких атрибутов счетчика за один запрос. Может не поддерживаться устройством.
Количество объектов в запросе	При включенном групповом опросе определяет количество запрашиваемых объектов (атрибутов) за один запрос. 0 - все атрибуты счетчика.
Глубина получасового архива	Количество считываемых записей получасовок при старте опроса. 0 - все записи
Глубина часового архива	Количество считываемых записей часового архива при старте опроса. 0 - все записи
Глубина суточного архива	Количество считываемых записей суточного архива при старте опроса. 0 - все записи.
Глубина месячного архива	Количество считываемых записей месячного архива при старте опроса. 0 - все записи

Название	Назначение
Количество архивных записей за цикл	Количество считываемых записей за один опрос.
Security	<p>Определяет тип безопасности. Устанавливаются в зависимости от настроек счетчика. Возможные варианты:</p> <ul style="list-style-type: none"> • None • Authentication • Encryption • AuthenticationEncryption
Authentication	<p>Определяет тип аутентификации. Возможные варианты:</p> <ul style="list-style-type: none"> • None, • Low, • High, • HighMd5, • HighSha1, • HighGmac, • HighSHA256
SystemTitle	Системный заголовок.
AuthenticationKey	Ключ аутентификации
BlockCipherKey	Блочный ключ шифрования
DedicatedKey	Специальный ключ

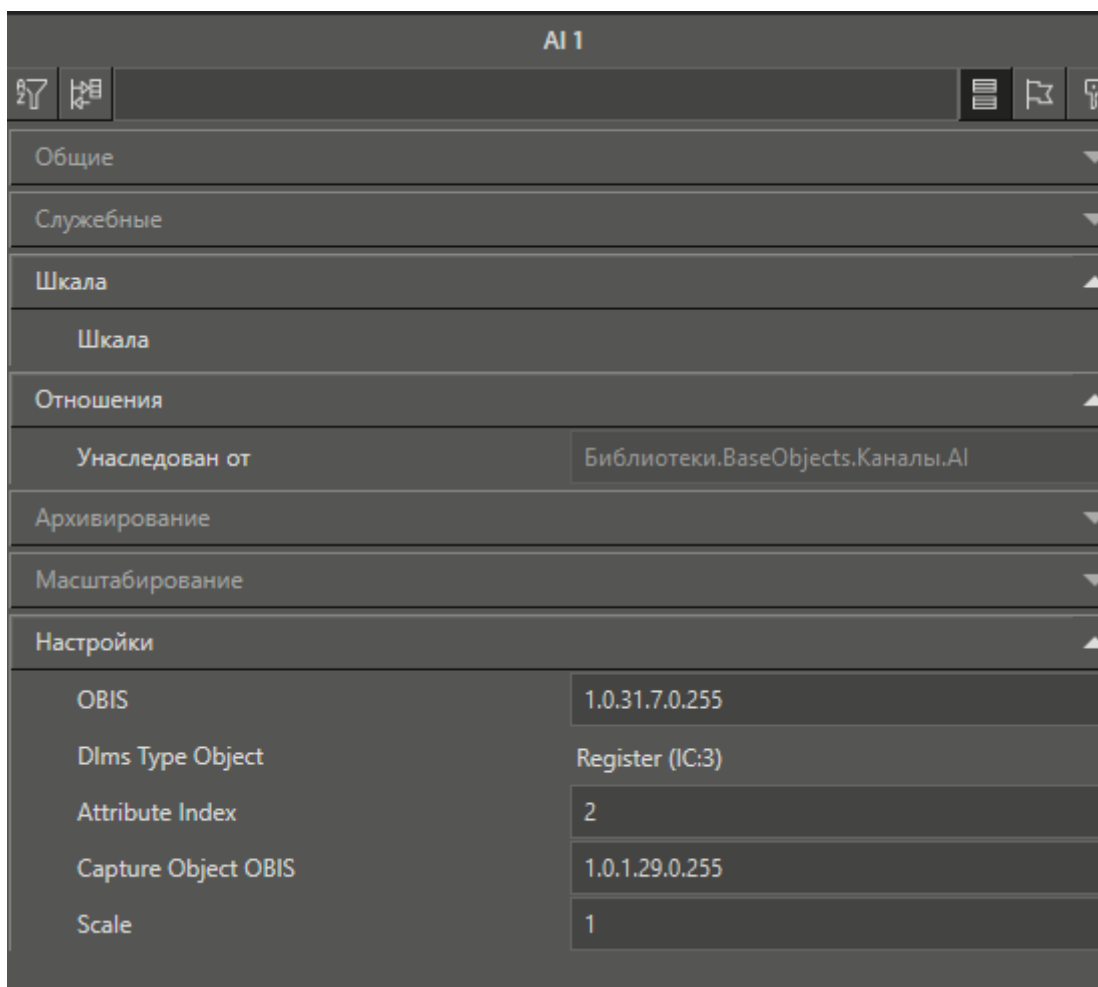
Спецификация СПОДЭС определяет конфигурацию значений ServerAddress-ClientAddress как способ идентификации уровня доступа при подключении:

Сервер	Клиент	Уровень доступа
1	16	Публичный клиент

1	32	Считыватель показаний
1	48	Конфигуратор

6.2.3.2.21.5. СВОЙСТВА КАНАЛА DLMS

Вид панели свойства канала Dlms:



Панель свойств заполняется автоматически после выполнения необходимых действий на вкладке устройства.

Свойство	Назначение
OBIS	Указывается OBIS код переменной
Dlms Type Object	Указывается тип объекта. Для тегов поддерживаются типы - Data, Register, Extended Register, Demand Register, Profile Generic

Свойство	Назначение
Attribute Index	Индекс атрибута объекта. Доступен для Extended Register и Demand Register. У Data и Register - всегда опрашивается индекс 2 (значение)
Capture Object OBIS	Список захватываемых объектов (для Profile Generic). Объект Profile Generic содержит список захватываемых объектов (объектов, чей архив хранится в данном профиле). Данный параметр позволяет указать к какому именно захватываемому объекту будет привязан этот тег.
Scale	Множитель. Считанное из прибора значение умножается на указанный коэффициент

6.2.3.2.22.SRTP

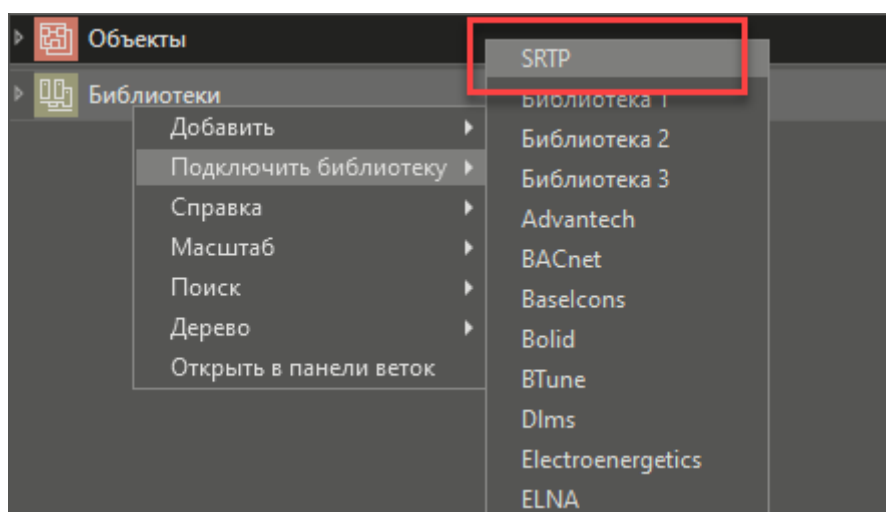
MasterSCADA 4D поддерживает работу с устройствами по протоколу GE Fanuc SRTP

Протокол SRTP работает через Ethernet. При обмене узел играет роль ведущего.

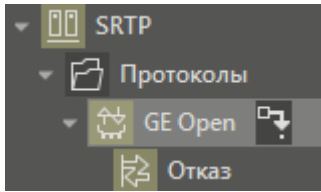
Важно! По умолчанию этот протокол не входит в состав среды разработки. Получить библиотеку можно в службе технической поддержки.

Важно! По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте sales@masterscada.ru

Для получения возможности использования протокола в проекте необходимо подключить соответствующую библиотеку:

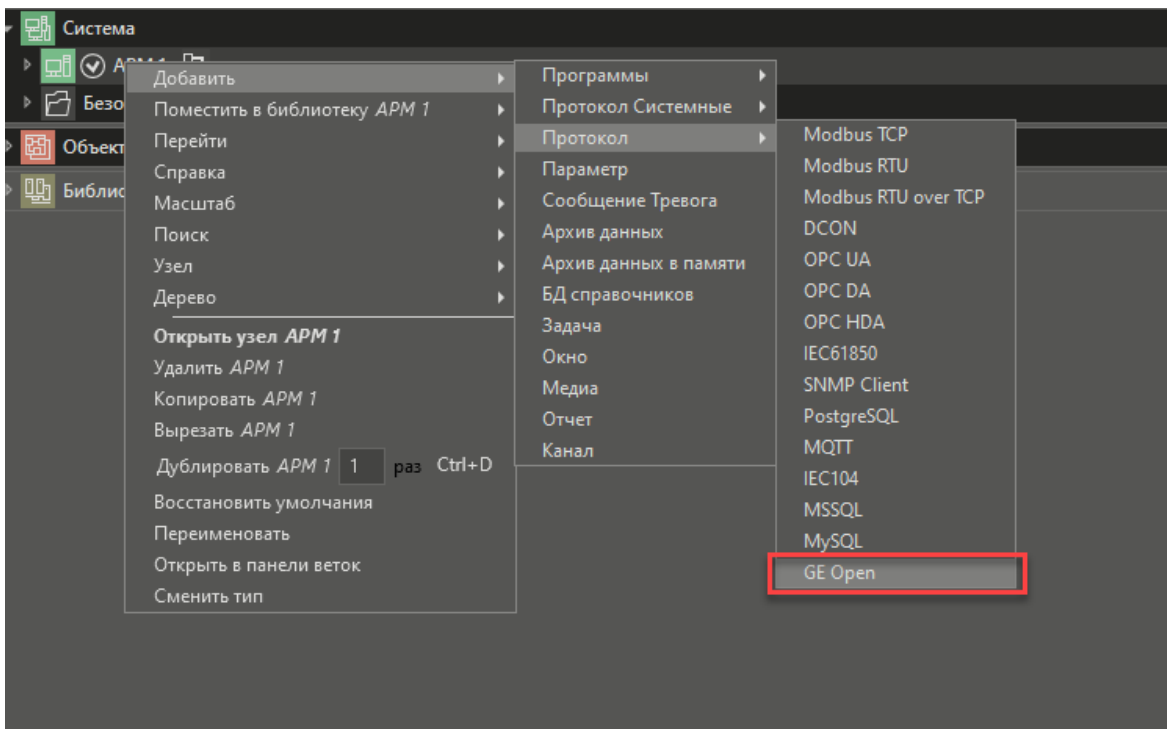


Библиотека SRTP имеет вид:



После этого в проект может быть добавлен протокол GEOpen любым известным способом, например, при помощи контекстного меню узла, или с использованием контекстной панели, или путем перетаскивания элемента из библиотеки в дерево системы.

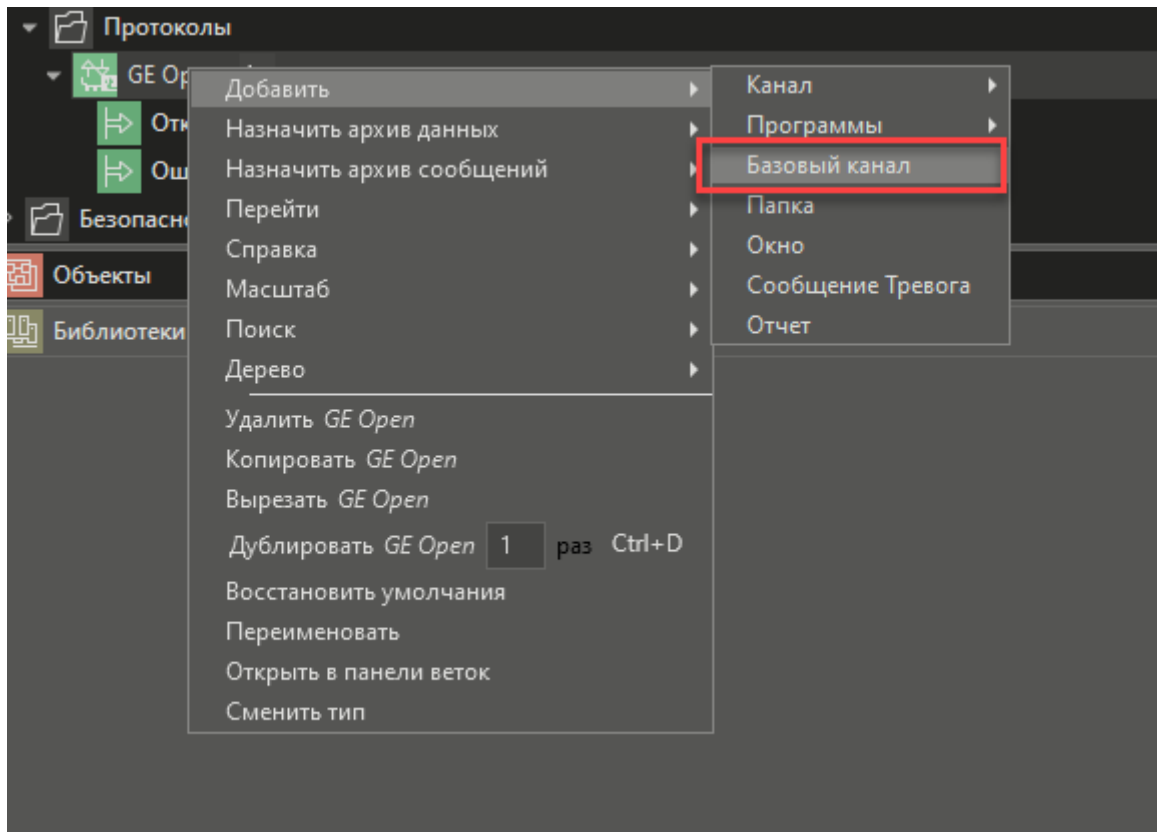
Рекомендуется добавлять элементы в проект при помощи контекстного меню:



Затем необходимо настроить панель свойств протокола.

Далее требуется добавить каналы в протокол.

Добавить канал вручную можно при помощи контекстного меню протокола:



Далее необходимо настроить панель свойств канала.

Для изменения настроек каналов добавленных в проект можно использовать табличное представление проекта.

6.2.3.2.22.1. СВОЙСТВА ПРОТОКОЛА GE OPEN

Вид панели свойств протокола GE Open:

GE Open

🔍 📄
☰ 🚩 📌

Общие

Имя	GE Open
Полное имя	Система.APM 1.Протоколы.GE Open
Метки	
Комментарий	Протокол GE Fanuc Ethernet

Служебные

Задача

Отношения

Настройки каналов

Протокол

IP адрес	127.0.0.1
Порт	18245
Таймаут	1000
Количество попыток	3
Максимальное количество байт в запросе	10
Максимально допустимый разрыв в запросе чтения	10

Разрешения

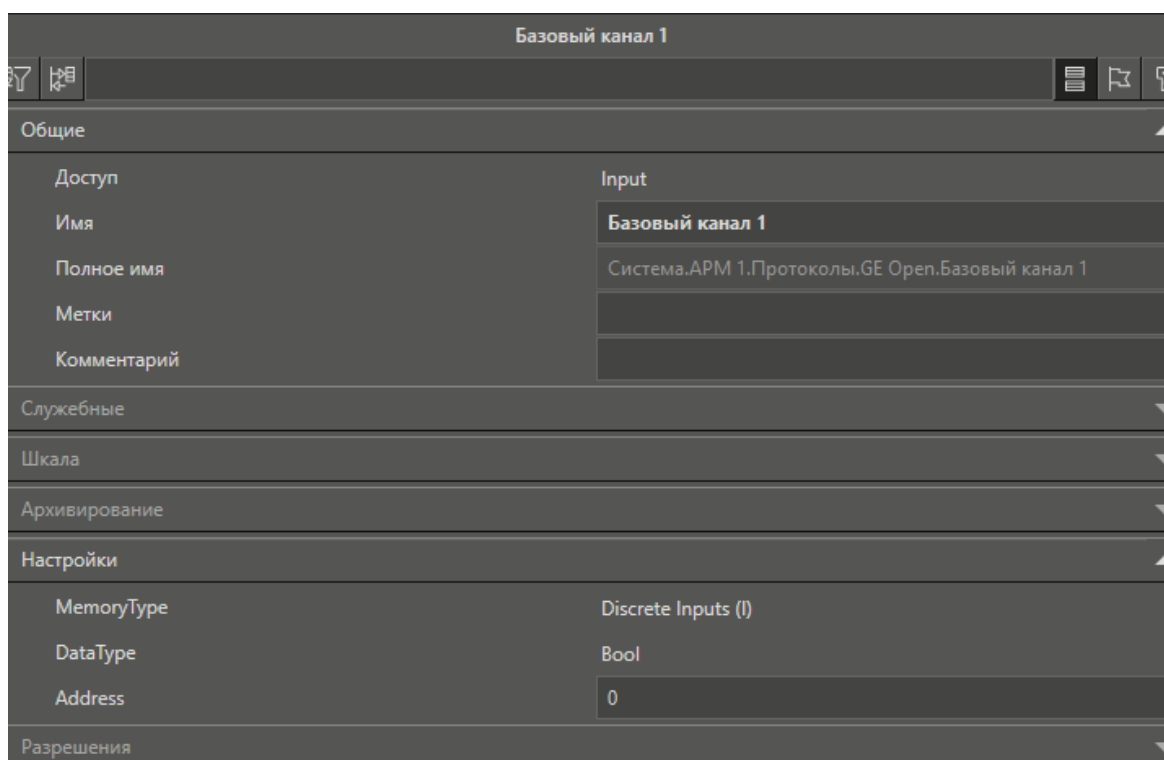
Доступ через OPCUA	По умолчанию
--------------------	--------------

Название	Рекомендации
Категория Общие	Как правило, для данного элемента эта категория не настраивается. Подробное описание данной категории приведено в соответствующем разделе
Категория Задача	Задается период, с которым MasterSCADA 4D будет стараться получать значения из сервера. Определяется способ работы протокола при резервировании узла. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи у протоколов.
Категория Настройки каналов	Задаются значения по умолчанию для создаваемых каналов.
IP-адрес	Основной IP-адрес устройства, которое отправляет и получает данные от MasterSCADA 4D.

Название	Рекомендации
ТСР-порт	Номер ТСР-порта устройства, по которому осуществляется подключение. По умолчанию установлено 18245
Таймаут	Время, в течение которого исполнительная система ожидает установления соединения с контроллером
Максимальное количество байт в запросе	Максимальное количество байт данных в одном пакете. Если пакет получается длиннее, то он будет разделен на части
Максимально допустимый разрыв	<p>Если между элементами меньше этой величины, то они объединяются в один пакет. Измеряется не в байтах, а в элементах (т.е. или биты, или байты - в зависимости от того, что опрашиваем).</p> <p>По умолчанию установлено стоит 10 байт</p>
Категория Разрешения	Описание см. в разделе Категория свойств Разрешения

6.2.3.2.22.2. СВОЙСТВА КАНАЛОВ ПРОТОКОЛА GE OPEN

Вид панели свойств каналов протокола GE Open



Описание основных свойств

Название	Описание
MemoryType	<p>Устанавливается сегмент памяти. В выпадающем списке нужно выбрать нужный.</p> <ul style="list-style-type: none"> • Есть битовые сегменты - I, Q, M, T, SA, SB, SC, S, G. Они могут читаться по байтам. • Есть WORD сегменты по 2 байта - AI, AQ, R.
DataType	<p>Устанавливается тип данных канала.</p> <ul style="list-style-type: none"> • У Word-сегментов можно указать типы - Word, Short, Long, DWord, Long, Float. Указывать BOOL нельзя. • У битовых сегментов - всегда тип только BOOL, другие указывать нельзя
Address	Указывается адрес каналов. Нумерация начинается с 1.

Описание остальных свойств смотрите в соответствующих разделах

6.2.3.3. MASTERSCADА 4D В РОЛИ SLAVE

Для того чтобы MasterSCADA 4D работала в роли Slave, необходимо настроить группу узла Внешние каналы для передачи данных по протоколам:

- Modbus RTU
- Modbus TCP
- OPC DA

Либо настроить панель свойств узла для использования протоколов:

- OPC UA
- IEC 104

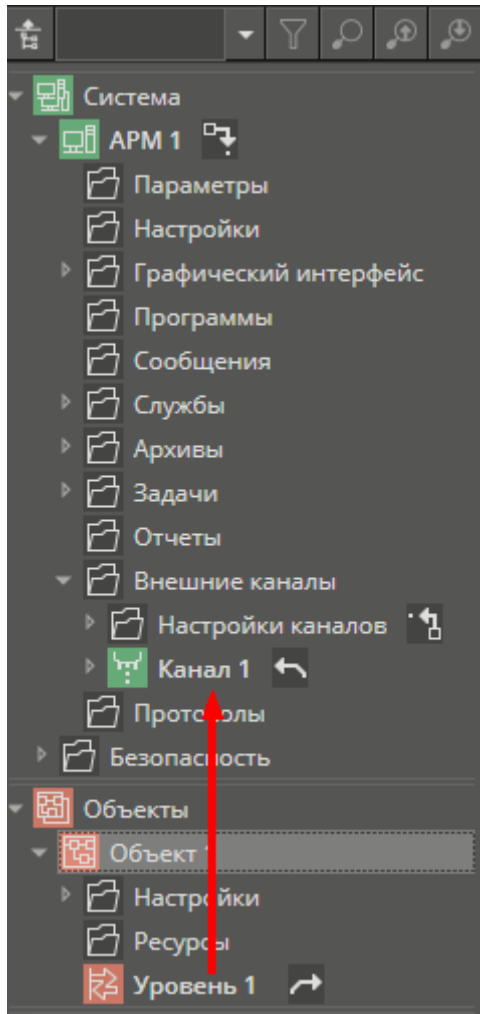
6.2.3.3.1. ВНЕШНИЕ КАНАЛЫ

Группа Внешние каналы представляет собой дочерний элемент узла (АРМ, Контроллера, Сервера и др.).

Внешние каналы используются для того, чтобы передать значения переменных из устройств, запрограммированных MasterSCADA 4D через протоколы Modbus TCP Slave или Modbus RTU Slave, IEC104 Slave либо через специализированный M-PLC-OPC-server (возможна передача данных не только через TCP/IP, com-порт, но и через модем).

Формирование группы Внешние каналы возможно вручную, когда при помощи контекстного меню группы добавляются каналы, задается их имя, тип данных и др. настройки, а затем настраивается связь этих каналов с элементами проекта.

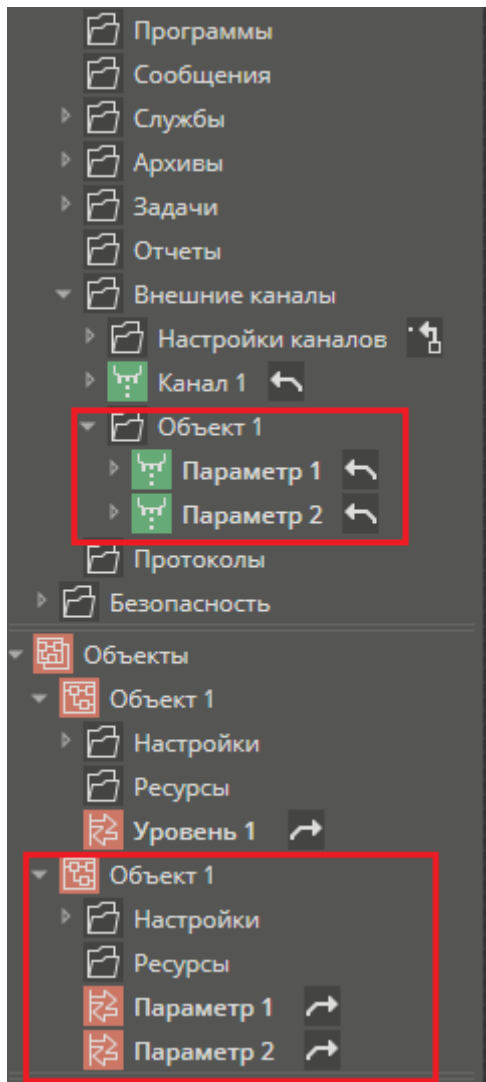
Для этого параметр перетаскивается на имя канала:



Автоматическое создание внешних каналов

Для облегчения построения связей проекта реализованы следующие возможности:

- перетаскивание канала модуля либо параметра проекта на группу Внешние каналы. При этом создается внешний канал со ссылкой на родительский элемент;
- перетаскивание модуля ввода-вывода, либо папки с параметрами, либо объекта на группу Внешние каналы. При этом создается вся структура сложного элемента в виде папок, содержащих внешние каналы со ссылками на соответствующие каналы модуля.



Панель свойств внешнего канала.

Параметр 1

Общие

Имя: Параметр 1

Полное имя: Система.АРМ 1.Внешние каналы.Параметр 1

Метки:

Комментарий:

Доступ: Чтение/Запись

Служебные

Отношения

Настройки

Адрес: -1

Адрес IEC104: -1

Тип значения: По умолчанию

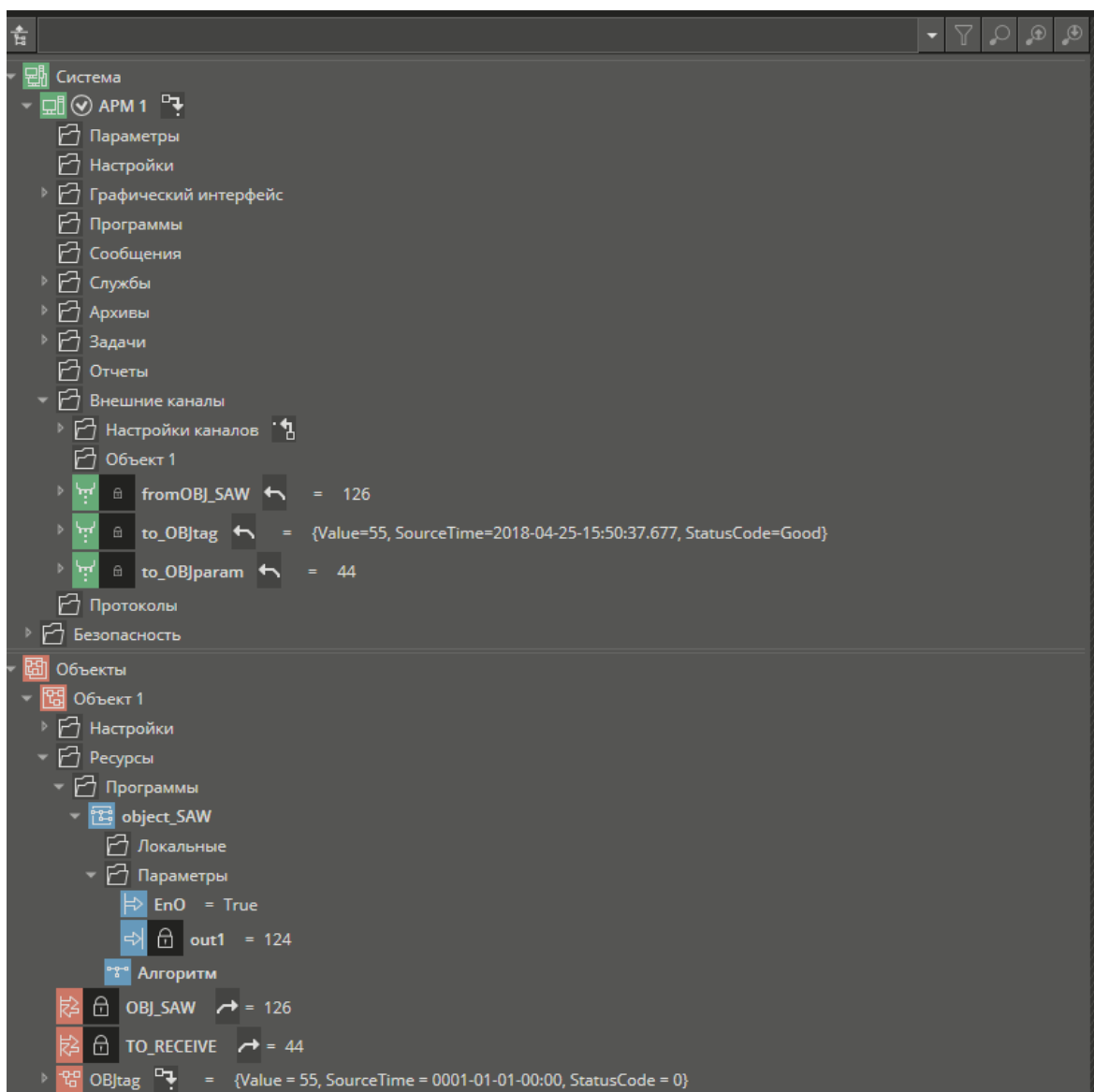
Описание основных свойств:

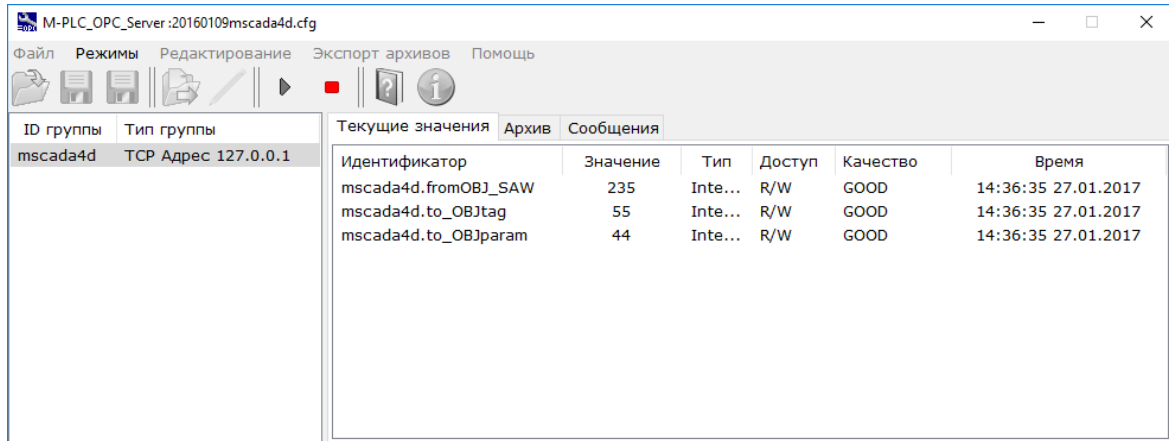
Свойство	Описание
Доступ (Категория Общие)	<p>Определяются действия, которые могут выполняться с данным каналом:</p> <ul style="list-style-type: none"> • Чтение - через OPC/Modbus Slave/ IEC104 Slave можно только читать. • Чтение/Запись - через OPC/Modbus Slave/ IEC104 Slave можно читать и писать. • Запись - через OPC/Modbus Slave/IEC104 Slave можно только писать.
Тип значения	<p>Определяется функция, которая будет использована для работы с данными по Modbus Slave. Если установлено значение по умолчанию, то тип значения выбирается исходя из типа значения параметра, на который ссылается канал.</p>
Адрес	<p>Задается адрес канала для доступа по Modbus Slave. Если установлено значение-1, то адрес сформируется</p>

Свойство	Описание
	автоматически при формировании карты переменных Modbus Slave.
Адрес IEC104	Задается адрес канала для доступа по IEC104 Slave.

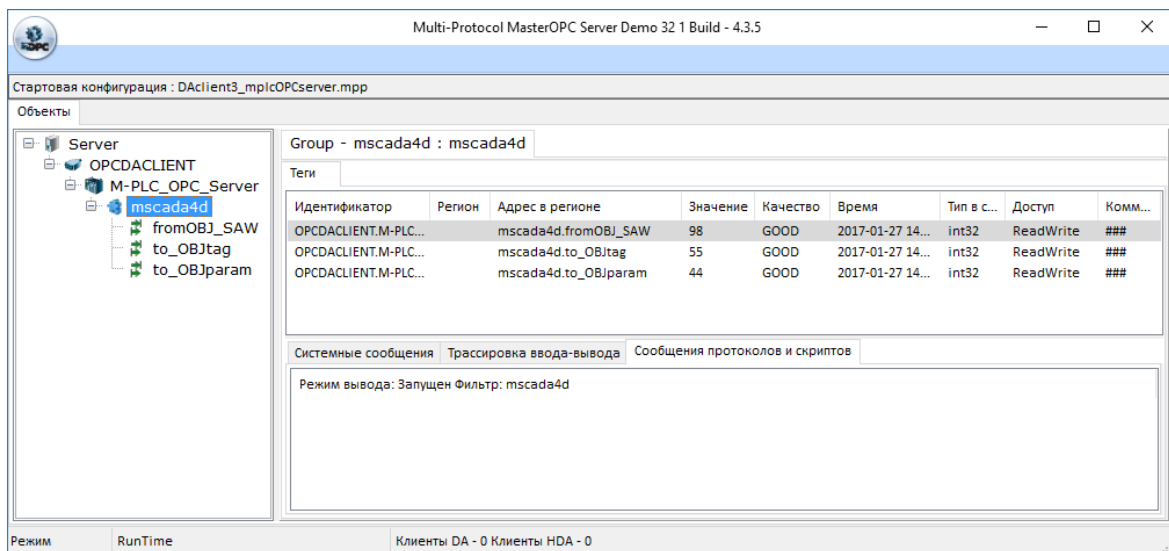
6.2.3.3.1.1. MASTERSCADA 4D КАК СЕРВЕР OPC DA

Каналы, созданные в группе Внешние каналы, доступны для сервера M-PLC OPC Server.



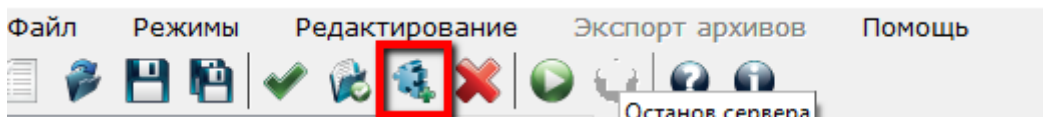


В паре с M-PLC OPC Server, MasterSCADA 4D является сервером OPC DA, и переменные проекта MasterSCADA 4D доступны клиенту OPC DA:

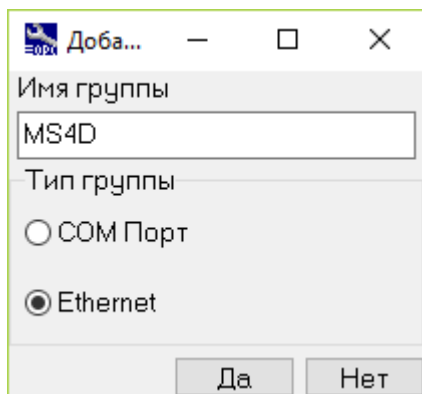


Конфигурирование M-PLC OPC Server

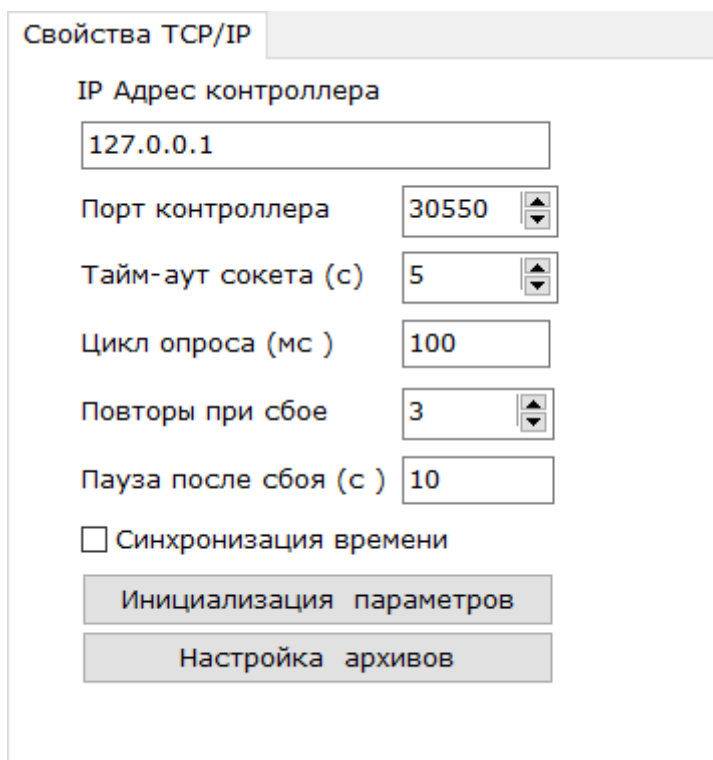
Для конфигурирования сервера необходимо нажать иконку в панели инструментов Добавить группу:



Затем указать имя группы, а также выбрать канал, по которому сервер будет подключаться к среде исполнения. Как правило это Ethernet:

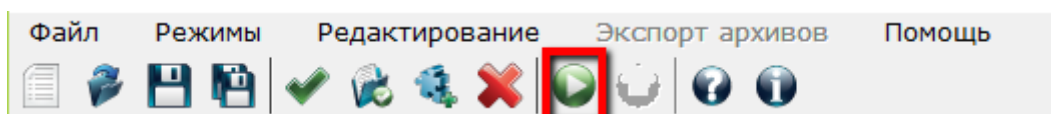


Далее требуется задать настройки подключения:



После этого необходимо нажать на кнопку Инициализация параметров. При этом среда исполнения MasterSCADA 4D должна быть запущена. В открывшемся диалоговом окне следует выбрать параметры, которые необходимо передавать другим OPC-клиентам.

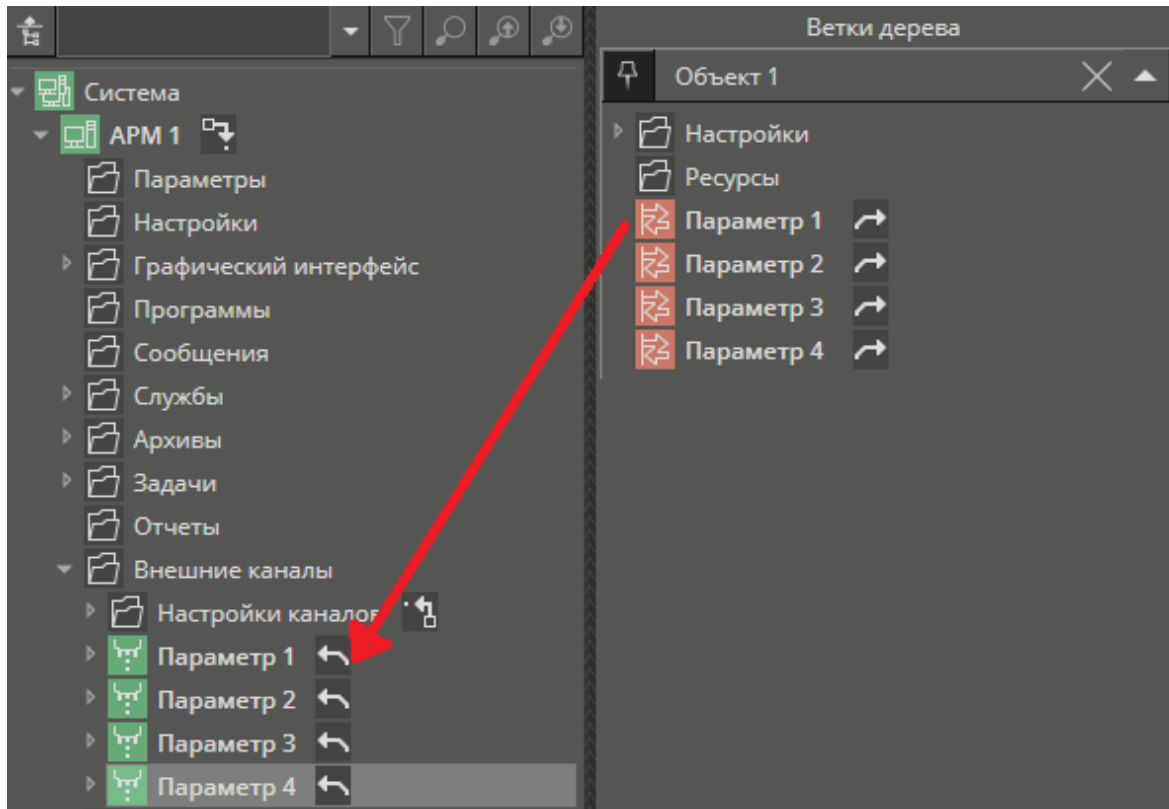
Для запуска сервера в режим исполнения нужно нажать иконку Старт сервера:



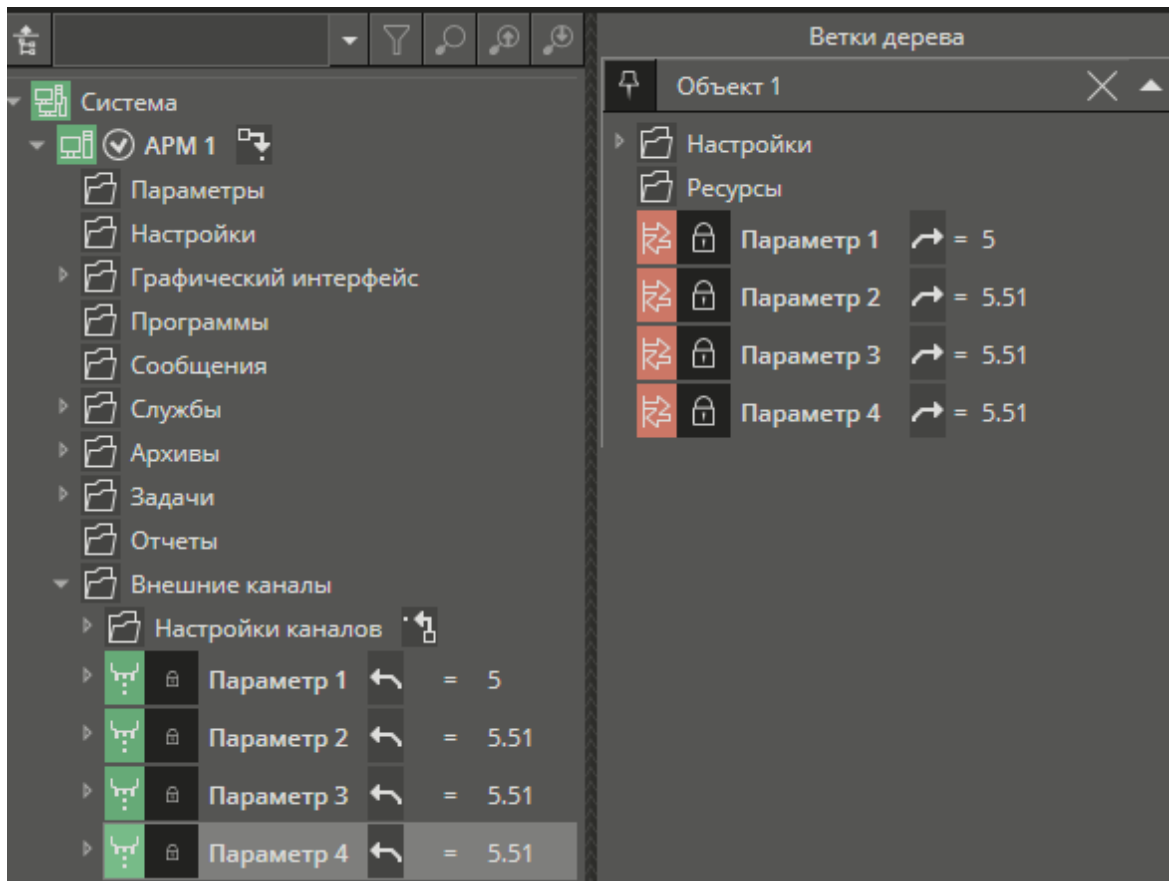
Запись параметра объекта в архив MasterSCADA 3

Архивируемый параметр объекта может дополнительно записываться в архив формата MasterSCADA 3. Для реализации такого режима необходимо выполнить следующие действия:

- перетащить параметр на группу Внешние каналы узла (при этом в группе создается ссылка на параметр):



- запустить узел:

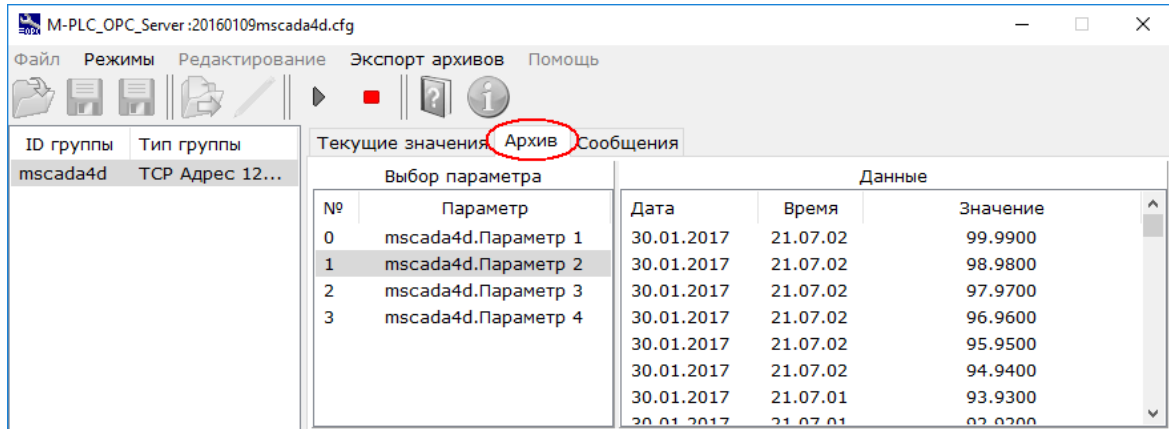


- подключить сервер M-PLC OPC Server, инициализировать параметры, и в настройке архивов для архива 0 задать Такт=1:

Задача	Такт	Всего записей	Записи за сеанс связи	Автоэкспорт	Период запроса
Задача 0	1	4080	240	0	0
Задача 1	0	4080	240	0	0
Задача 2	0	4080	240	0	0
Задача 3	0	4080	240	0	0

- запустить M-PLC OPC Server.

После этого значения параметра записываются не только в архив MasterSCADA 4D (БД <профиль пользователя> \AppData\Roaming\MPSSoft\MasterSCADA4D<..>\Debug_<проект>\<узел>\PLC\data.db), но и в архив формата MasterSCADA 3 (в памяти буфер 4 МБ, с дублированием на диске в папке архива MasterSCADA 4D, файл архива MasterSCADA 3 – data0_0):



Передача внешних каналов через модем.

Если необходимо передать данные через подключенный либо встроенный модем, то в настройке узла Параметры запуска RT следует задать:

```
/m2 /b115200
```

Где m2 - номер порта, к которому подключен GSM-modem, а b115200 - скорость.

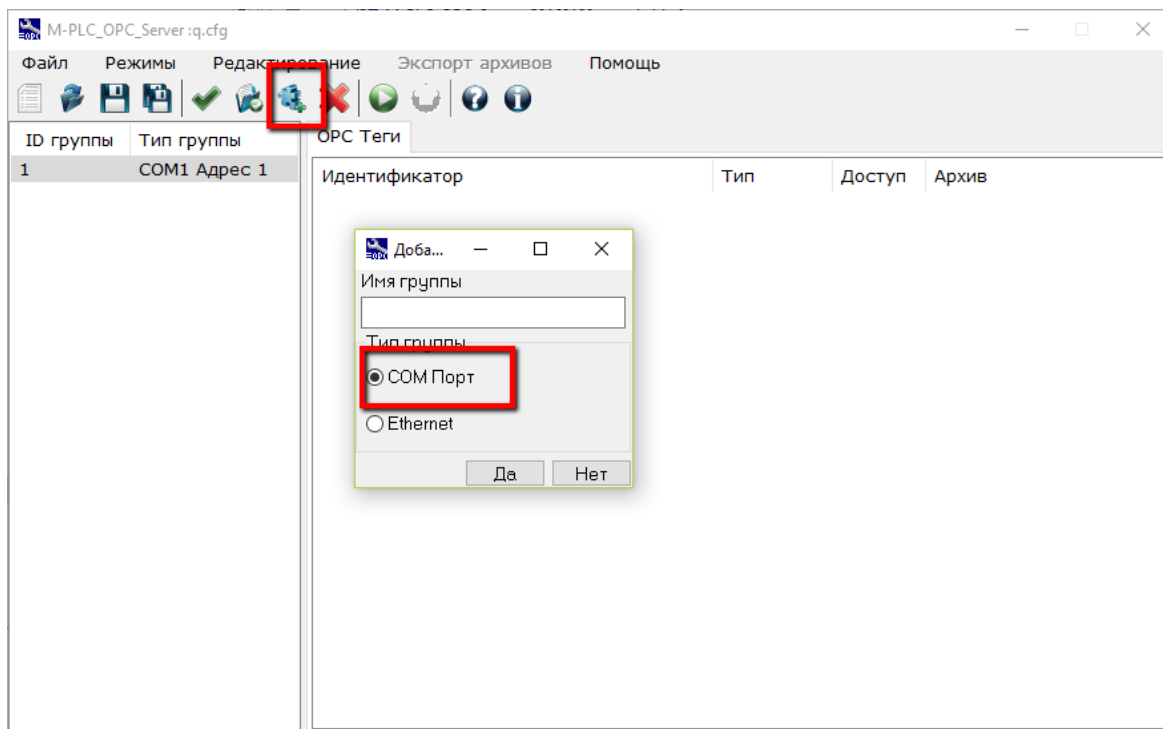
При этом используется следующая строка инициализации модема, подключенного к устройству, на котором работает среда исполнения:
ATE0Q0V1&C0&D2&S1S0=0S7=60+CBST=71,0,1

Если модем имеет другую строку инициализации, то настройка Параметры запуска RT будет иметь вид: /m2/ b115200 /i[строка инициализации]

Устройство, запрограммированное MasterSCADA 4D, будет выступать в роли Slave. Инициативная передача данных не поддерживается.

В настройках M-PLC OPC Server необходимо сделать следующее.

Добавить группу, в настройках которой выбрать COM Порт.



В настройках COM-порта, к которому физически подключен gsm-модем, задать его настройки. Значения выбираются исходя из возможностей подключаемого оборудования. Для настройки специальных параметров модема следует нажать кнопку Модем:

Редактирование группы - 1

Свойства RS-232

Номер порта: 1

Скорость: 1200

Тайм-аут (с): 10

Межсимвольный тайм-аут (мс): 50

Повторы при сбое: 3

Пауза после сбоя (с): 10

Адрес контроллера: 1

Цикл опроса (с): 1

Синхронизация времени

Модем

Инициализация параметров

Настройка архивов

Журнал ошибок Готово Отмена

В открывшемся диалоговом окне необходимо установить флаг Разрешить модем, указать телефон, на который требуется дозваниваться, а также строку инициализации.

Модем

Строка инициализации

ATE0Q0V1X4&C1&D2&S0S0=0+IFC=0,0;+CBST=7,0,1

Строка дозвона

ATD8111111111

Разрешить модем

Рекомендуемые настройки для GSM

Межсимвольный тайм-аут - 5000 мс

Тайм-аут - 10 с

Да Нет

По окончании настройки следует провести инициализацию параметров.

6.2.3.3.1.2. MASTERSCADА 4D В ПОЛИ MODBUS SLAVE

Внешние каналы, наряду с доступом через стандарт OPC, могут быть также доступны другим системам по протоколам Modbus TCP, Modbus RTU. В этом случае исполнительная

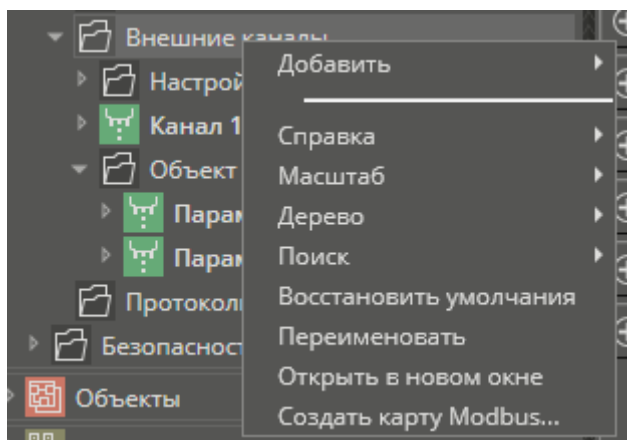
система MasterSCADA 4D выступает в роли Slave, а другое приложение - в роли Master. В качестве Master'a может выступать и другое устройство, на котором установлена исполнительная система MasterSCADA 4D.

Необходимо задать адреса Modbus (IEC104) и определить, какая функция будет использоваться Master'ом для чтения/записи данных. Для этого в панели настроек каналов необходимо задать поля Адрес и Тип значения.

Канал 1	
Общие	
Доступ	Чтение/Запись
Имя	Канал 1
Метки	
Комментарий	
Служебные	
Тип элемента	MasterSCADA.System.ExternalChannel
Id элемента	59058
Позиция	1
Id описания	
Запрет наследования	<input type="checkbox"/>
Id элемента внешней библиотеки	
Количество подэлементов	0
Справочный раздел	
Id канала	
Тип канала	Reference
Настройки	
Тип значения	Вещественный
Адрес	0
Адрес IEC104	0

Автоматическое формирование адресов Modbus

Адреса могут формироваться автоматически (рекомендуется). Для этого необходимо в контекстном меню группы Внешние каналы выбрать пункт меню Создать карту Modbus.



При этом откроется диалоговое окно, в котором нужно указать место хранения и имя csv-файла.

Вид csv-файла:

	A	B	C	D	E	F	G	H	I
1	Группа	Параметр	Адрес	Адрес(hex)	Направление	Тип	Тип значения	Комментарий	
2		Канал 1		0 0x0000	Чтение/Запись	РегистрХранения	System.Single		
3	Объект 1	Параметр 1		2 0x0002	Чтение/Запись	РегистрХранения	System.Single		
4	Объект 1	Параметр 2		4 0x0004	Чтение/Запись	РегистрХранения	System.Single		
5									
6									

Адресное пространство для логических и вещественных переменных различается. Система будет выбирать тот или иной канал для получения/отправки, в зависимости от того какой функцией она будет опрошена.

В дальнейшем данный файл можно использовать при конфигурировании продукта Modbus Universal MasterOPC Server.

Подключение по Modbus TCP Slave

Вышеуказанных настроек достаточно для подключения клиентов по Modbus TCP.

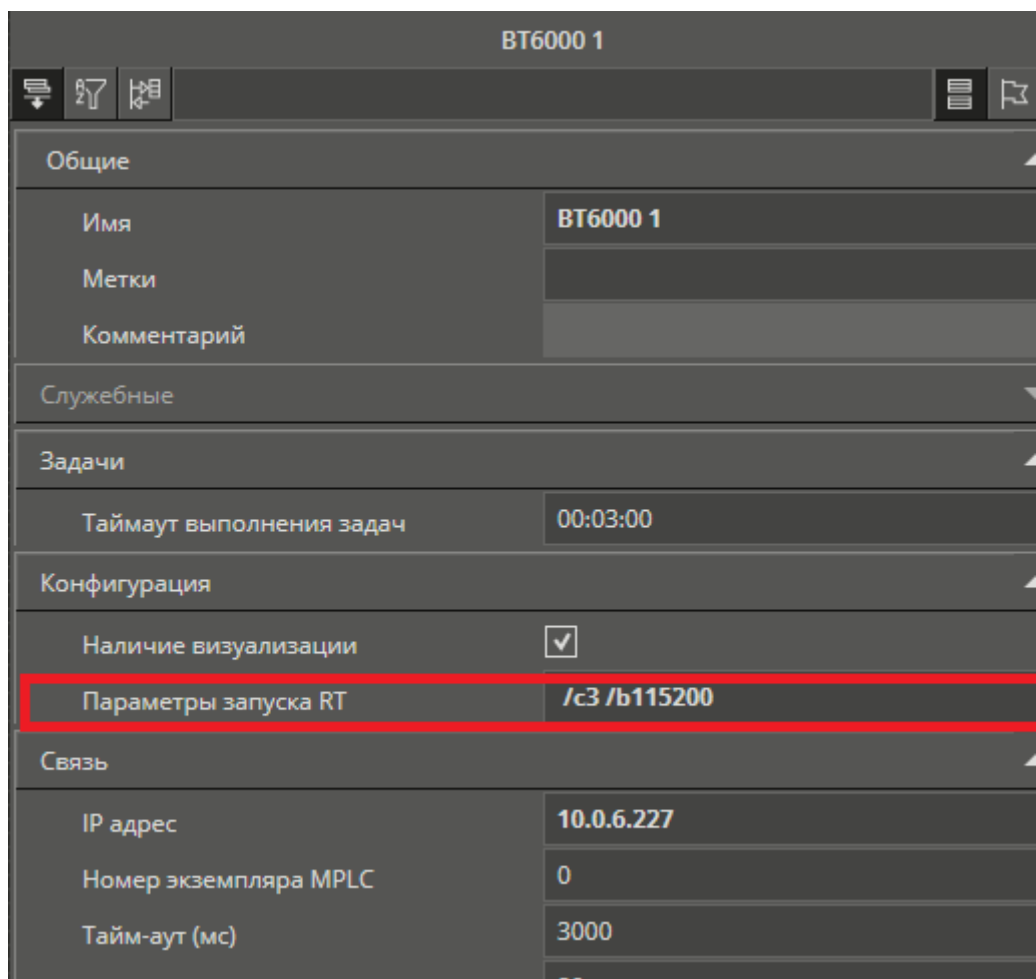
Никаких дополнительных настроек в среде разработки не требуется.

В клиенте достаточно указать IP-адрес узла, в котором будет работать проект MasterSCADA 4D, указать номер порта TCP/IP 502 и создать карту переменных.

Важно! Для того чтобы отключить возможность работать с внешними каналами по Modbus TCP, оставив при этом связь через Modbus RTU, следует в настройках узла в поле Порт для ModbusTCP установить значение 0

Подключение по Modbus RTU Slave

Чтобы устройство, запрограммированное MasterSCADA 4D, отвечало по Modbus RTU Slave, необходимо в настройке узла Параметры запуска RT задать строку параметров, как показано на рисунке:



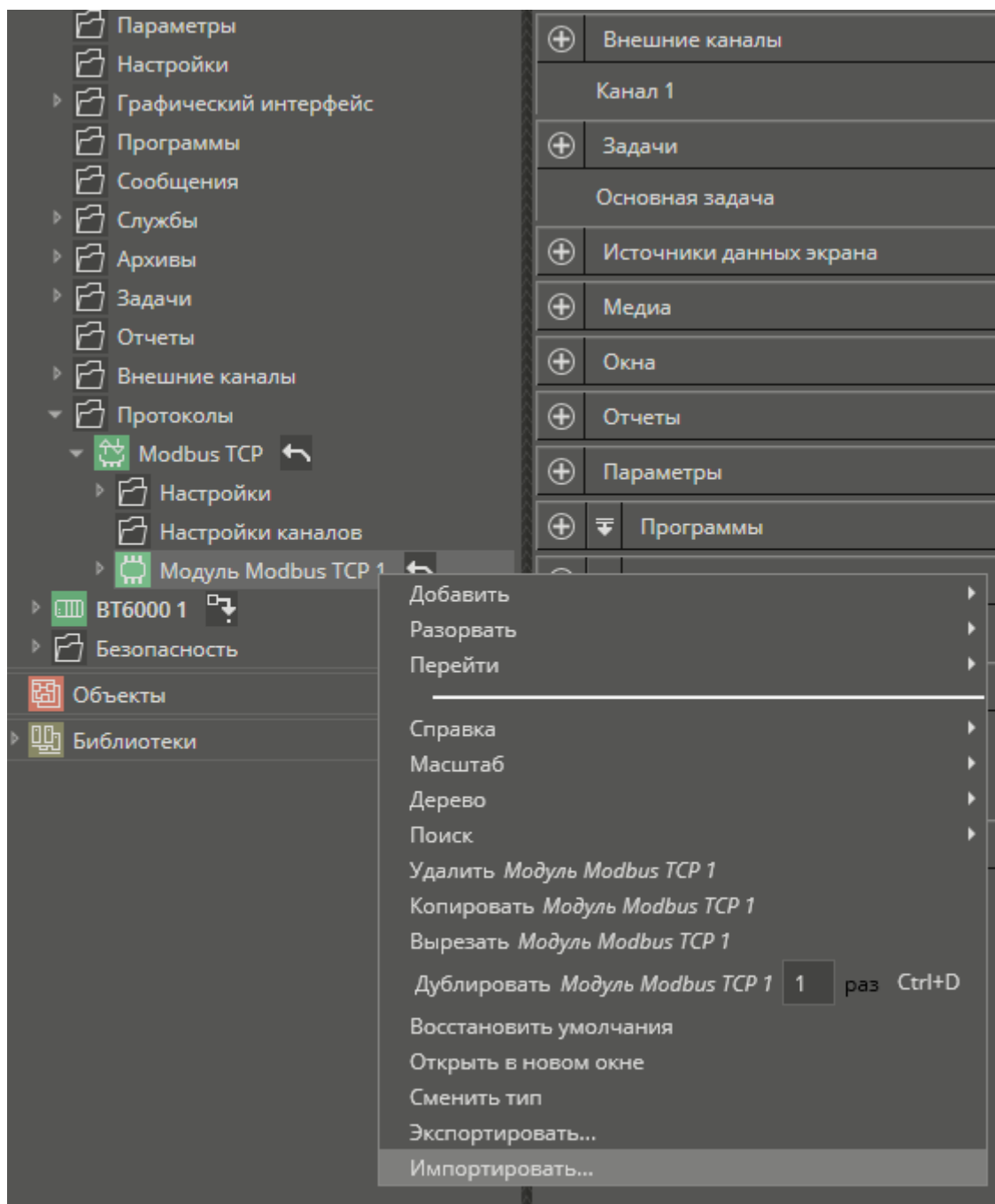
Настройка имеет вид /cN /bS, где N - номер порта, а S - скорость.

При этом адрес устройства по умолчанию будет равен 1. Если к одному com-порту подключено несколько устройств, запрограммированных средствами MasterSCADA 4D, то необходимо для каждого устройства задать свой адрес, тогда настройка Параметры запуска RT будет иметь вид: /c2 /b115200 /a2, где a2 - адрес устройства.

Связь между проектами MasterSCADA 4D через Modbus

В случае, если в качестве клиента используется MasterSCADA 4D, т.е. когда создаются два автономных проекта, связанных между собой через протокол Modbus, то в проект клиента можно добавить карту переменных автоматически.

Для этого в контекстном меню модуля Modbus необходимо выполнить пункт Импортировать.



В диалоговом окне выбрать csv-файл, который получился в результате автоматического создания карты переменных в проекте, слушащем поставщиком данных.

6.2.3.3.1.3. IEC104 SLAVE

Внешние каналы, наряду с доступами через стандарт OPC, Modbus TCP, Modbus RTU могут быть также доступны другим системам по протоколу IEC 104. В этом случае исполнительная система MasterSCADA 4D выступает в роли Slave, а другое приложение - в роли Master. В качестве Master'a может выступать и другое устройство, на котором установлена исполнительная система MasterSCADA 4D.

Для включения этого протокола необходимо в настройках узла разрешить доступ по нему, а также установить другие настройки:

APM 1

Общие

Служебные

Задачи

Конфигурация

Отношения

Связь

Автозапуск клиента визуализации	<input checked="" type="checkbox"/>
IP адрес	127.0.0.1
Доступ по OPC UA	<input type="checkbox"/>
Номер экземпляра MPLC	0
Тайм-аут (мс)	3000
Порт для OPC UA	16550
Логин для OPC UA	
Пароль для OPC UA	
Адрес клиента OPC UA	
Порт для Modbus TCP	502
Порт для Web сервера	8043
Максимальный размер UDP пакета	0
Автозапуск исполнительной системы	<input checked="" type="checkbox"/>
Последовательность байт в Modbus	_3_2_1_0_7_6_5_4
Запрет автозагрузки проекта	<input type="checkbox"/>
Запрет входящих подключений	<input type="checkbox"/>
Максимальное число подключений Mod	20
Время неактивности подключения Mod	10s
Доступ по IEC104	<input checked="" type="checkbox"/>
Порт для IEC104	2404
ASDU адрес IEC104	1
Период циклической рассылки по IEC10	0ms
Доступ по IEC61850	<input type="checkbox"/>
Порт для IEC61850	102
В IEC61850 параметры SYSTEM*_VALUE	<input checked="" type="checkbox"/>

Необходимо задать адреса IEC104 и определить, какая функция будет использоваться Master'ом для чтения/записи данных. Для этого в панели настроек каналов необходимо задать поля Адрес IEC104 и Тип значения.

Канал 1	
Общие	
Доступ	Чтение/Запись
Имя	Канал 1
Метки	
Комментарий	
Служебные	
Тип элемента	MasterSCADA.System.ExternalChannel
Id элемента	59058
Позиция	1
Id описания	
Запрет наследования	<input type="checkbox"/>
Id элемента внешней библиотеки	
Количество подэлементов	0
Справочный раздел	
Id канала	
Тип канала	Reference
Настройки	
Тип значения	Вещественный
Адрес	0
Адрес IEC104	0

Подключение по IEC104 Slave

Вышеуказанных настроек достаточно для подключения клиентов по IEC 104.

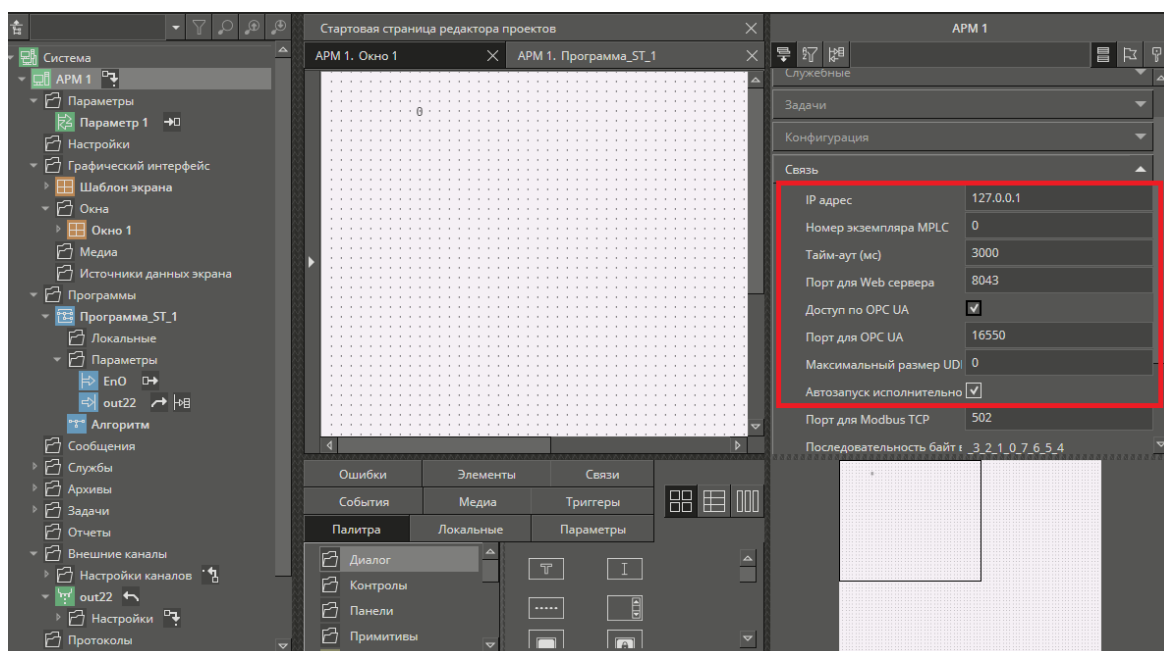
Никаких дополнительных настроек в среде разработки не требуется.

В клиенте достаточно указать IP-адрес узла, в котором будет работать проект MasterSCADA 4D, указать номер порта TCP/IP (2404 по умолчанию), адрес ASDU (по умолчанию) и создать карту переменных, согласно настройкам сделанным в настройках элементов группы Внешние каналы.

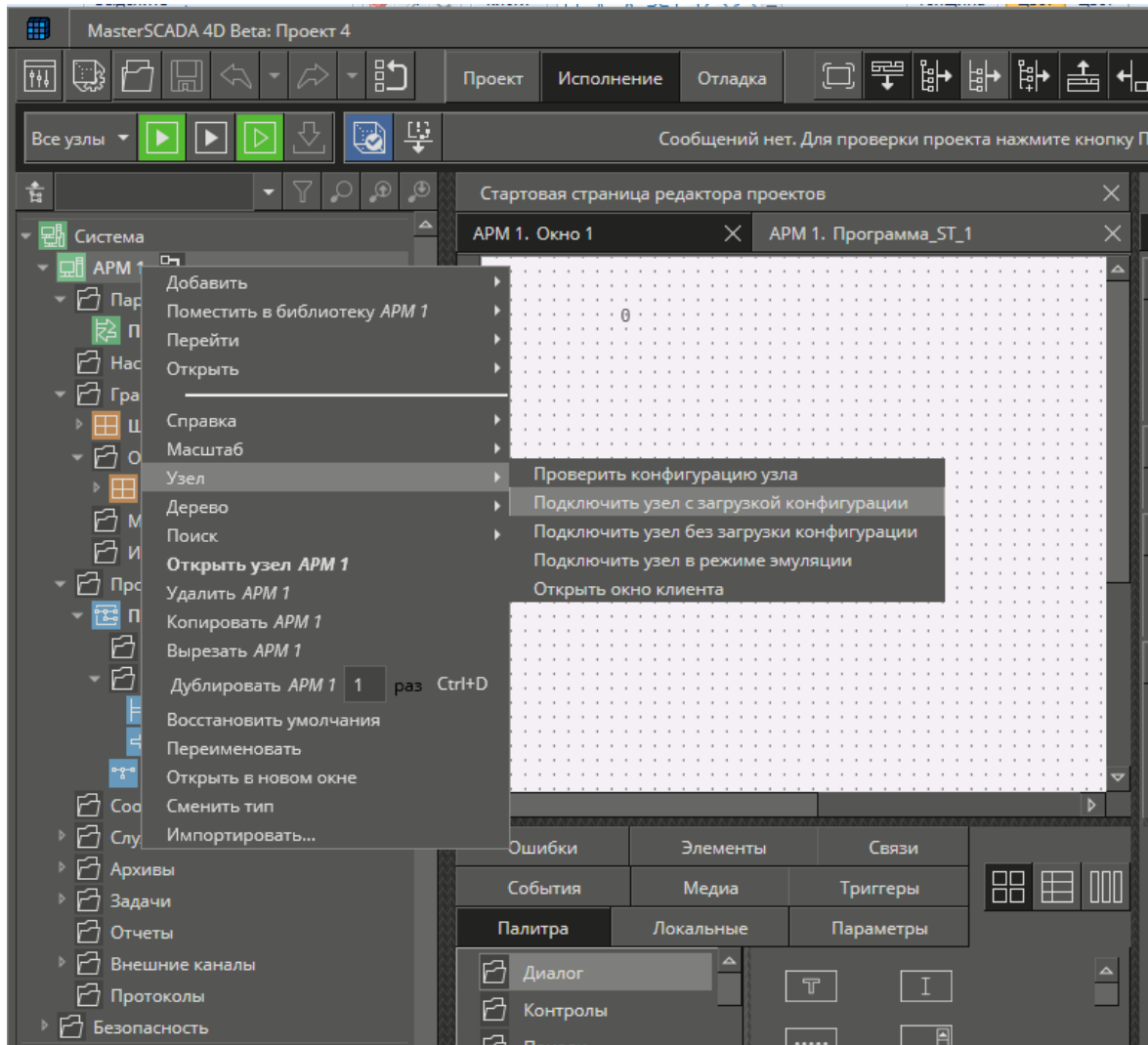
6.2.3.3.2. MASTERSCADА 4D КАК СЕРВЕР OPC UA

При работе в роли сервера MasterSCADA 4D поддерживает режимы чтения и подписки. Поддерживаются простые типы данных, а также возможна работа с одномерными массивами, состоящими как из элементов простых типов данных, так и из предопределенных типов вида SYSTEM_*_PARAM. Также поддерживаются одномерные массивы структур.

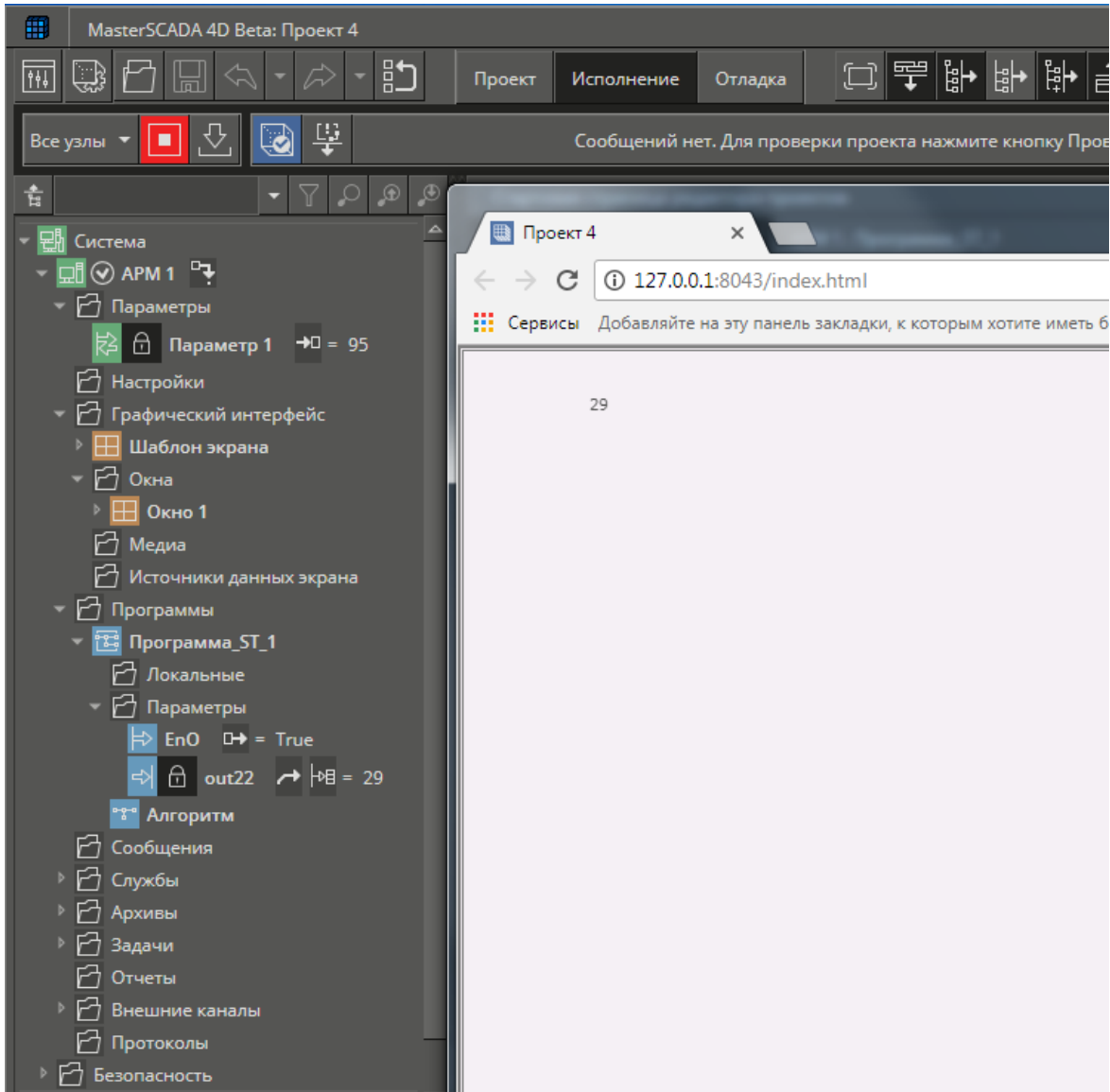
Предположим, что в проекте создан узел с параметрами связи, как показано на рисунке ниже:



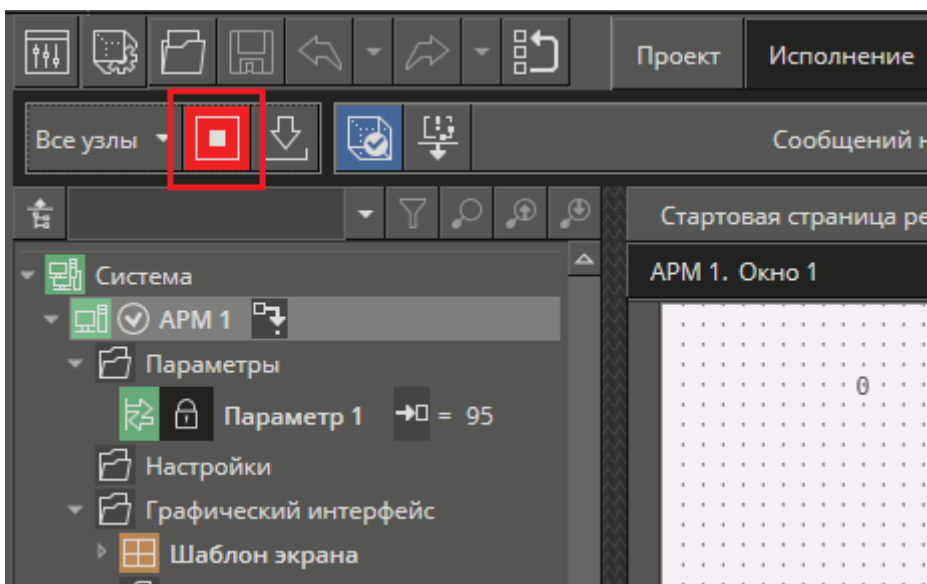
Запустим среду исполнения и загрузим в нее текущий проект. Для этого можно в контекстном меню узла выбрать пункт Узел.Подключить с загрузкой конфигурации



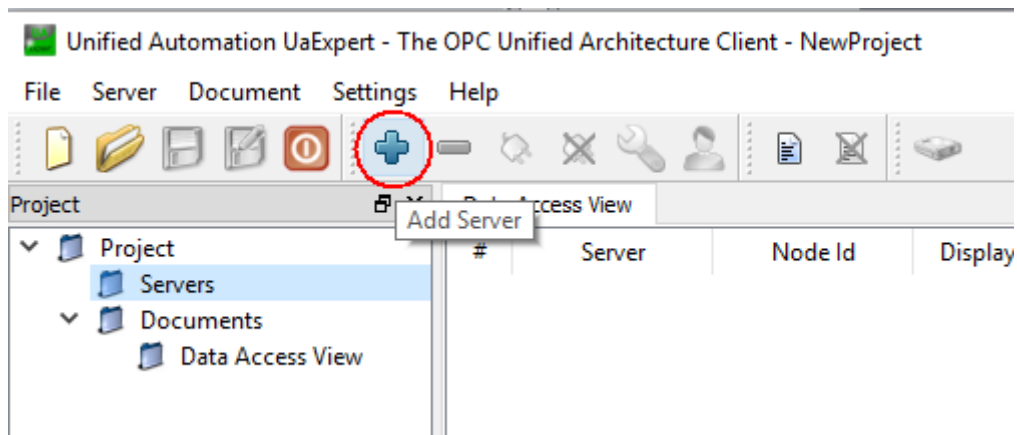
По окончании загрузки конфигурации узла среда подключается к исполнительной системе (при заданных параметрах, кроме того, графический клиент откроется автоматически):



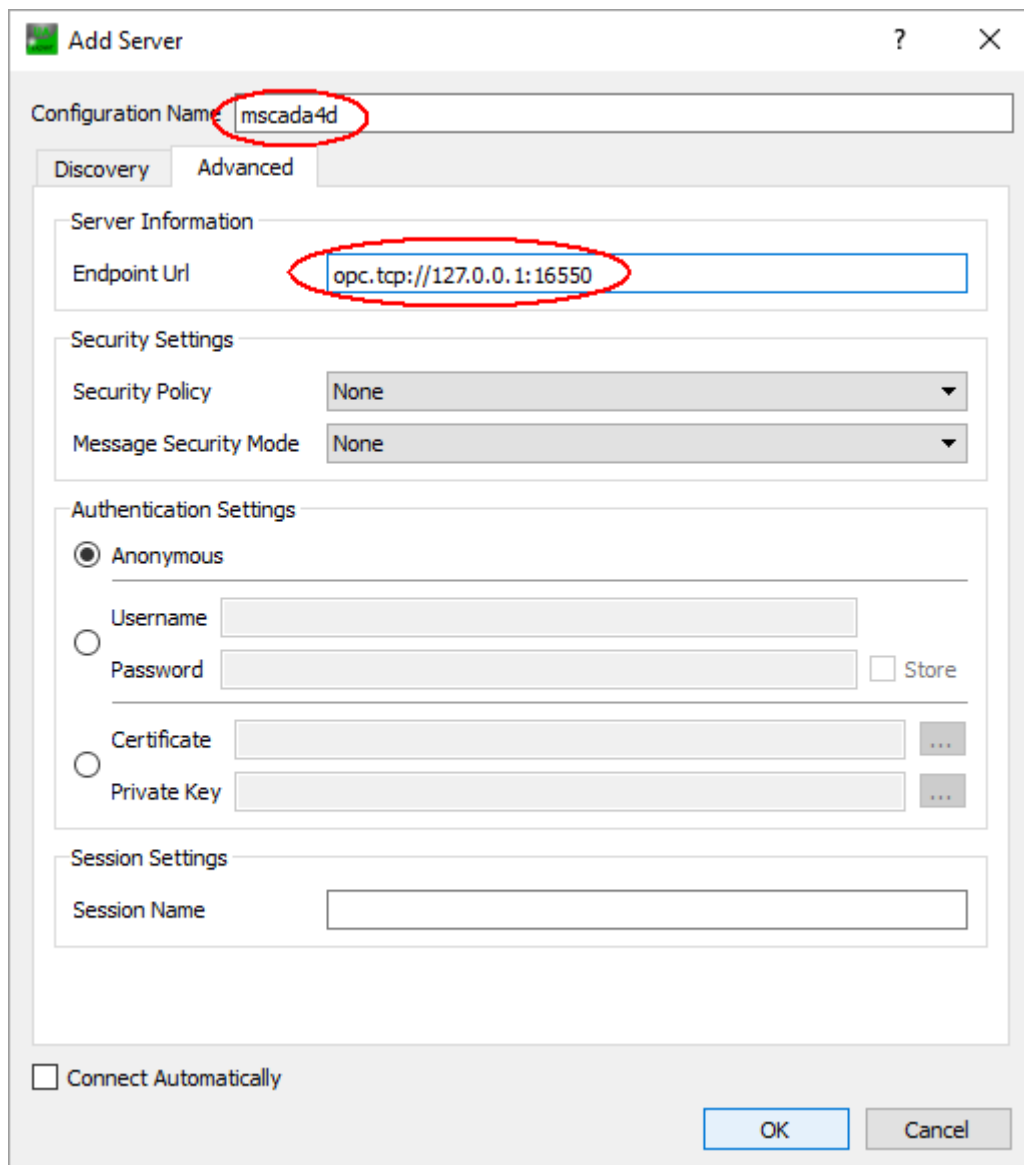
Отключаем среду от исполнительной системы:



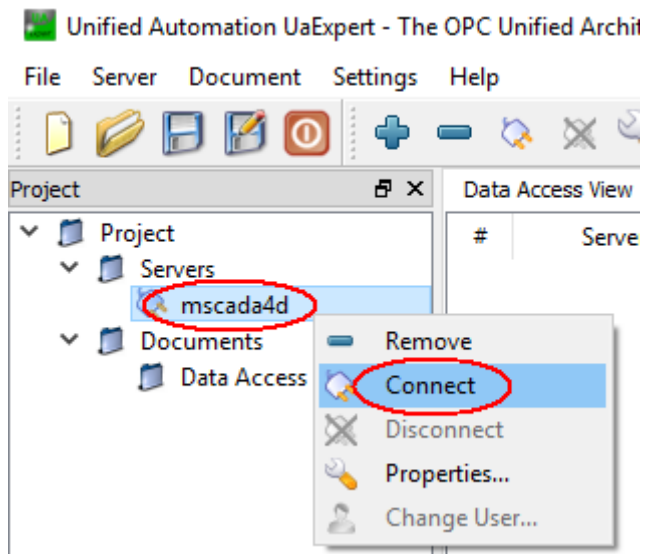
Запускаем клиент OPC UA (в примере используется UaExpert) и нажимаем кнопку добавления сервера:



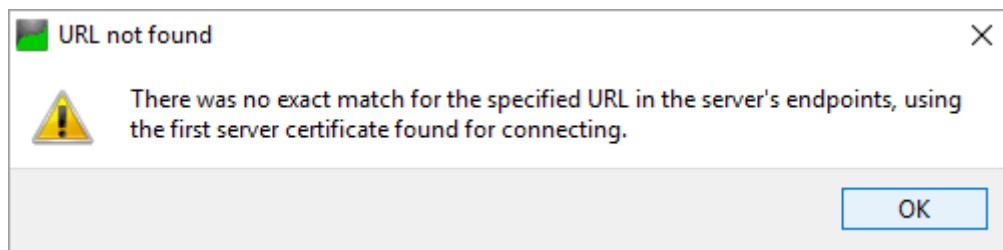
На вкладке Advanced открывшегося диалога конфигурируем подключение к исполнительной системе MasterSCADA 4D как к серверу OPC UA:



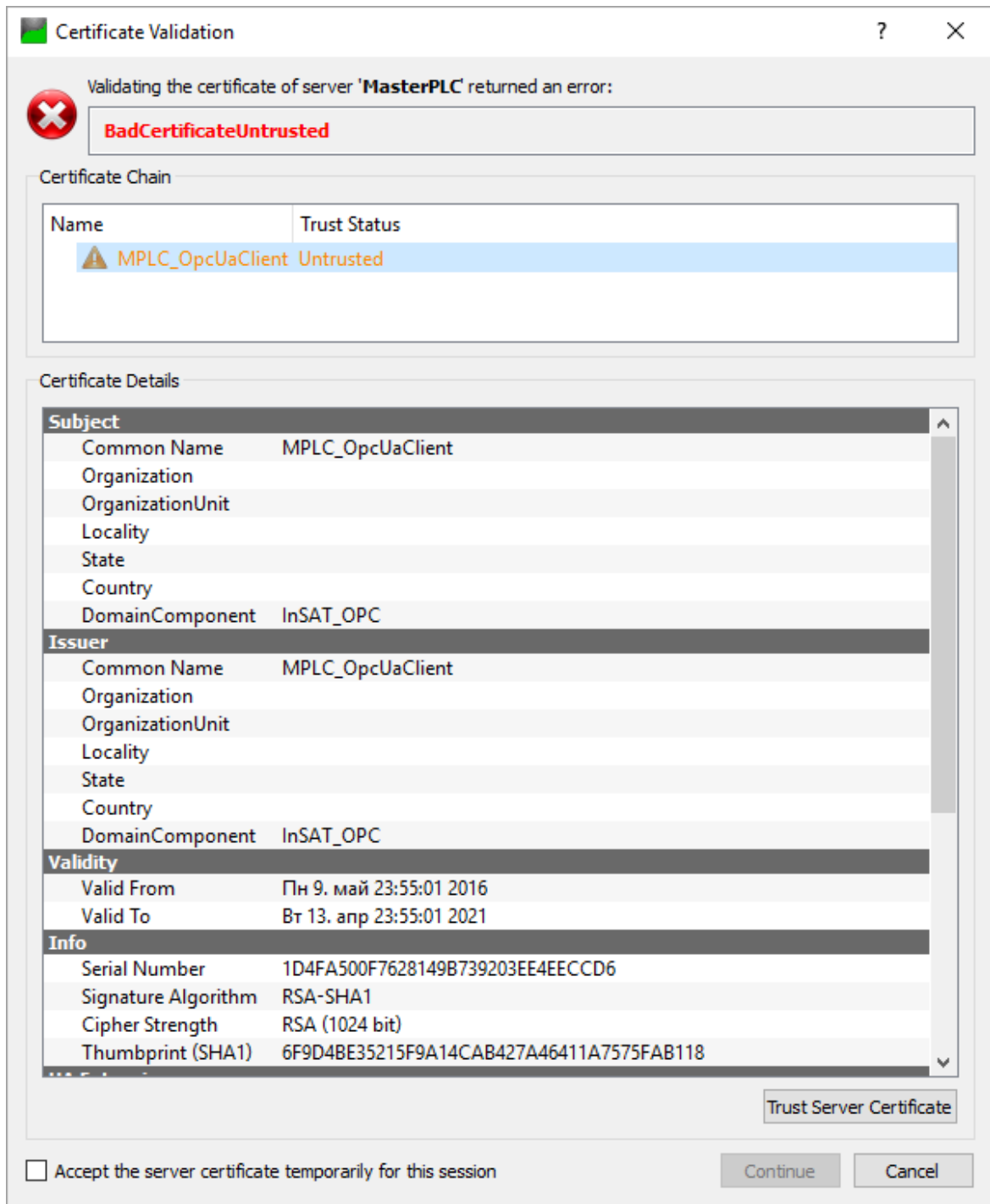
Сервер добавляется в дерево UaExpert. Для подключения к этому серверу выполняем команду Connect из его контекстного меню:



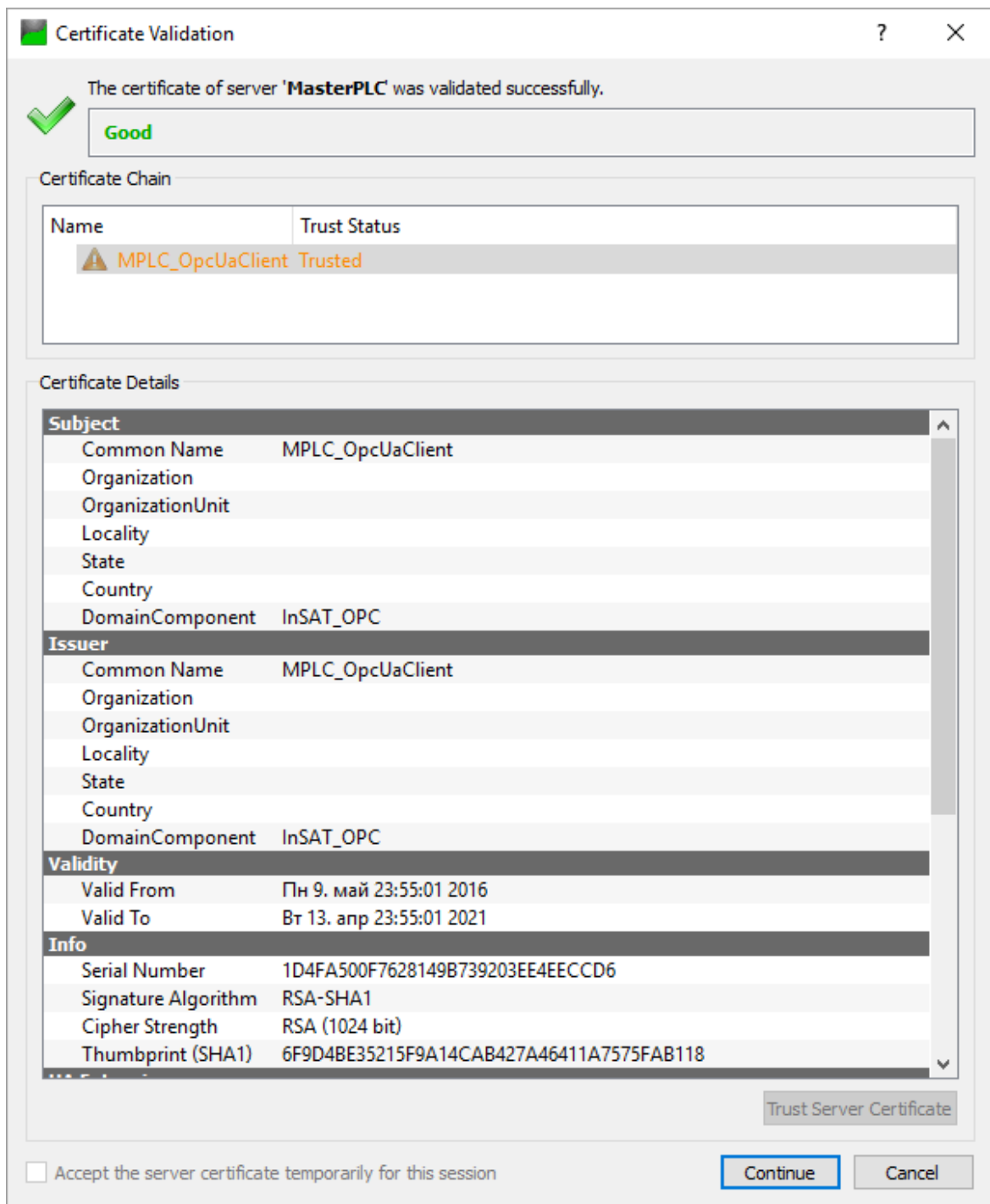
Открывается следующее сообщение:



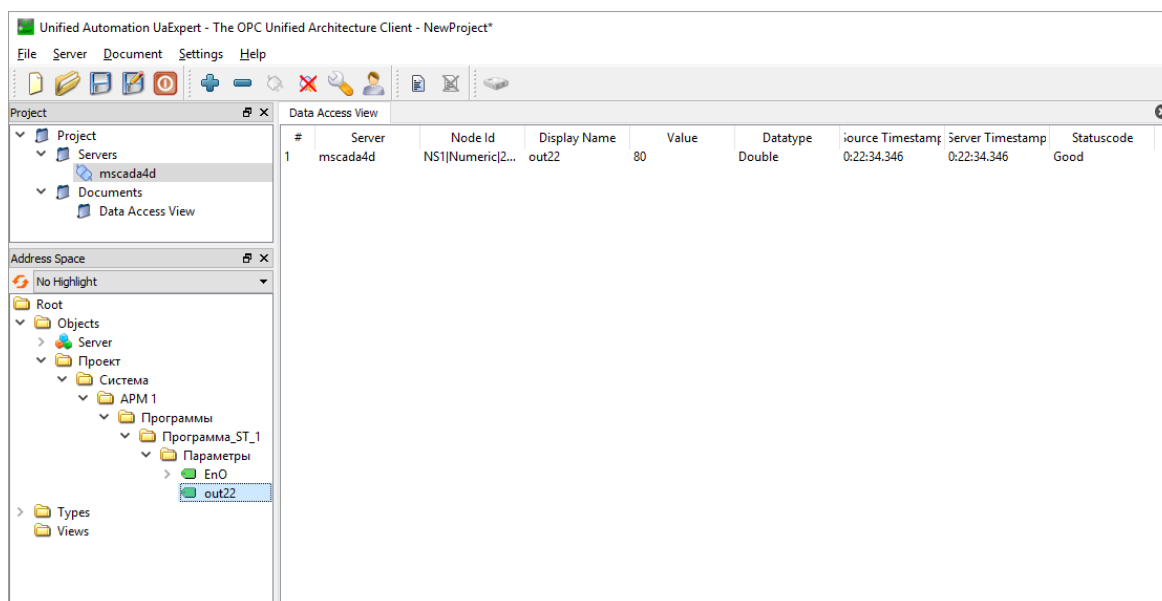
Нажимаем ОК – открывается диалог с сообщением о том, что сертификат OPC UA сервера (т.е. сертификат MasterPLC, исполнительной системы MasterSCADA 4D) не принадлежит к числу доверенных:



Для исправления ситуации нажимаем кнопку Trust Server Certificate:



Нажимаем Continue, после чего выполняется подключение к серверу:



Элементы проекта будут отображаться в дереве клиента в зависимости от настройки элементов Доступ через OPCUA. Способ отображения структур в OPC UA клиенте определяется свойством узла Отображение структур в OPC UA сервере

При работе по протоколу OPC UA SLAVE создается, если его не было ранее, сертификат в рабочей папке исполнительной системы *CertificateStore*

Порядок передачи данных при подписке

В клиенте, получающим данные от исполнительной системы MasterSCADA 4D, можно установить следующие настройки согласно стандарту OPC UA.

Publishing Interval - период, с которым сервер отправляет клиенту пакеты с данными

Queue Size - настройка, задаваемая у тега, указывает максимальное количество значений по этому тегу, передаваемых в одной подписке

SamplingInterval - настройка, задаваемая у тега, определяет как часто можно отправлять значения. Если установлено значение -1 или 0, то клиенту отправляется каждое изменение параметра (не более **QueueSize** в каждом пакете). Если больше 0, то клиенту отправляются значения, если его метка отличается от предыдущего отправленного больше на **SamplingInterval**

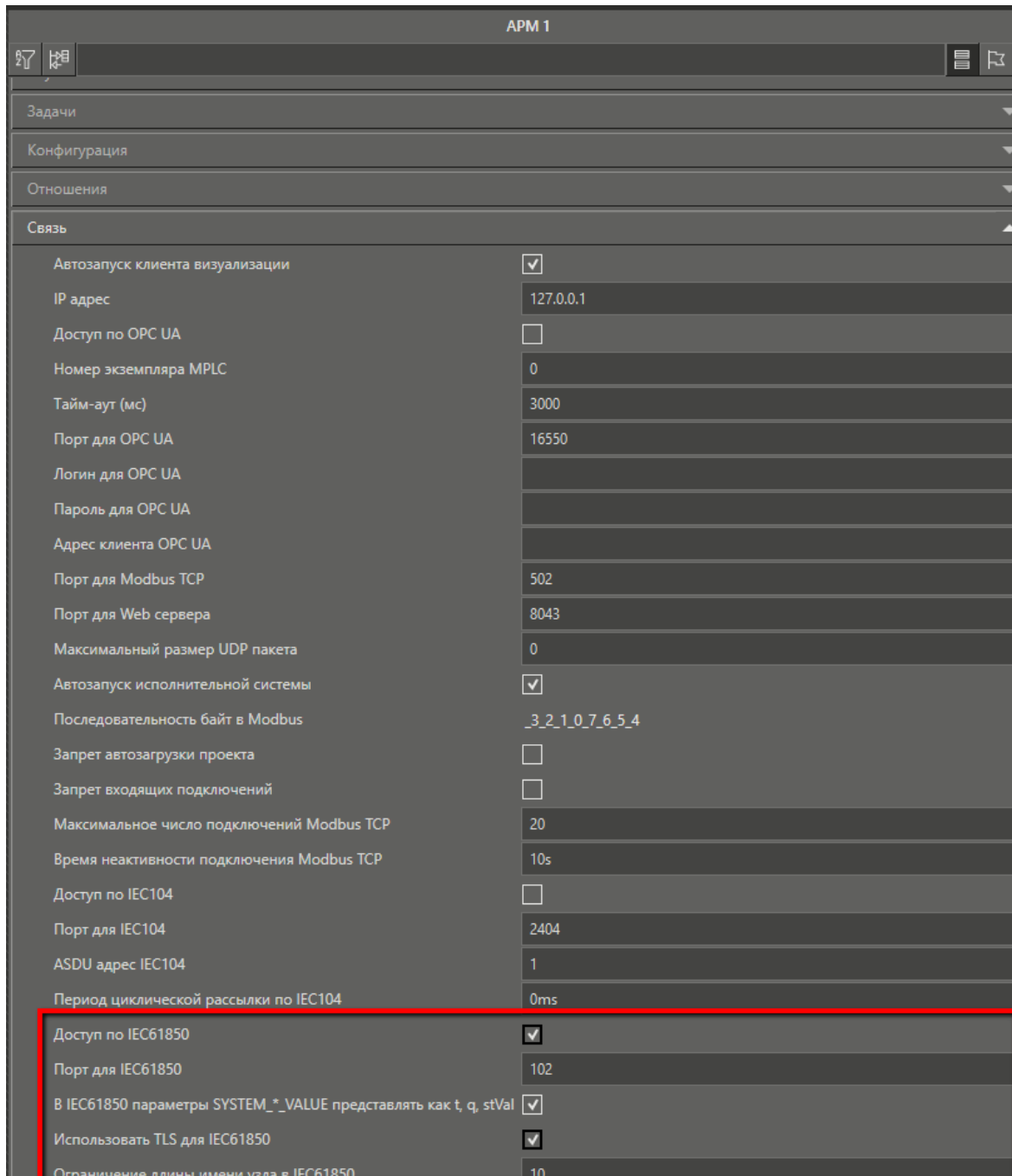
Поддержка параметра у тега подписки **MonitoringMode=Sampling**.

Например, в клиенте установлено: **PublishingInterval=1сек QueueSize =10, SamplingInterval = -1**, а значение меняется раз в 100мс, то сервер должен отправлять клиенту по 10 значений, но не чаще 1 раз в секунду

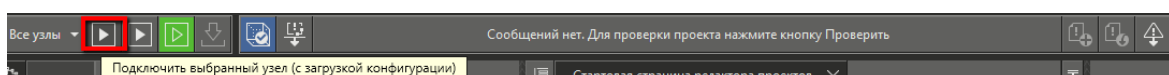
6.2.3.3.3. IEC61850 SLAVE

Рассмотрим пример настроек MasterSCADA 4D, которые позволят получить данные по протоколу IEC61850 во внешнем приложении. В этом случае исполнительная система будет выступать в роли Slave (сервера), а другое приложение в роли Master (клиента).

Предположим, что в проекте создан узел с параметрами связи, как показано на рисунке ниже:



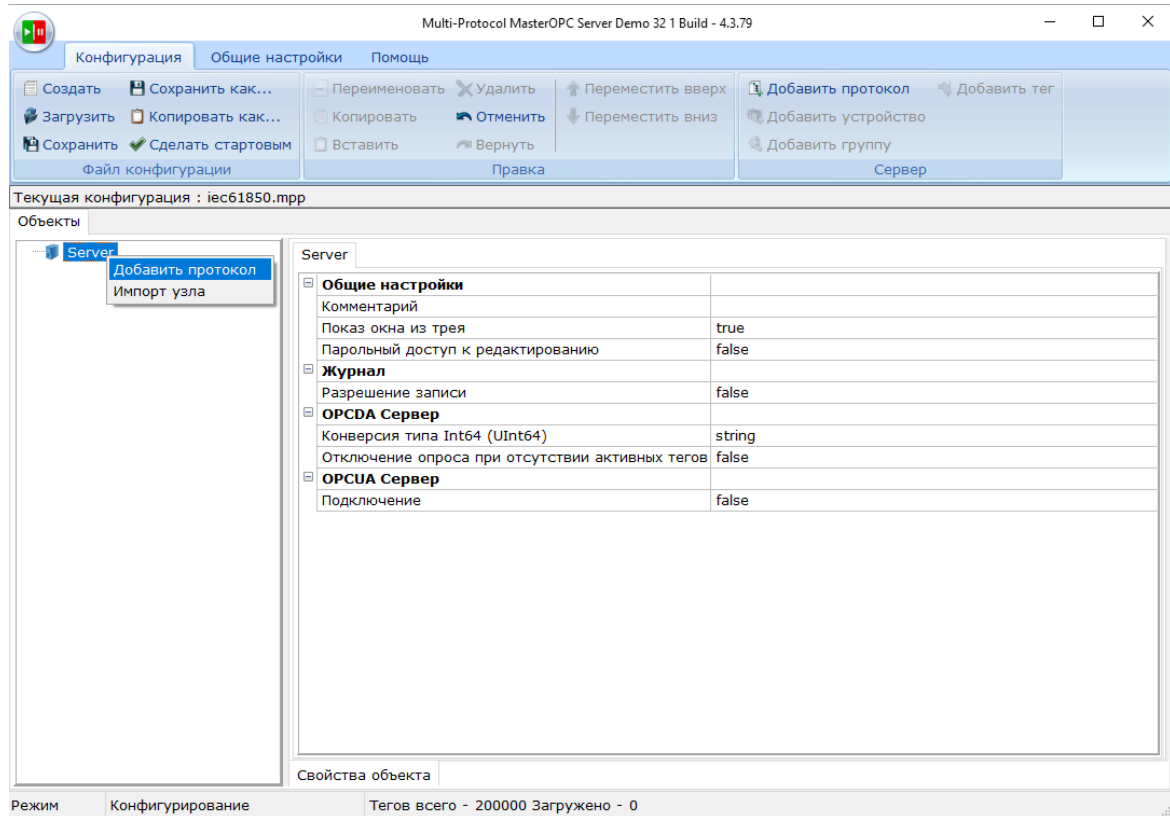
Запустим среду исполнения и загрузим в нее текущий проект:

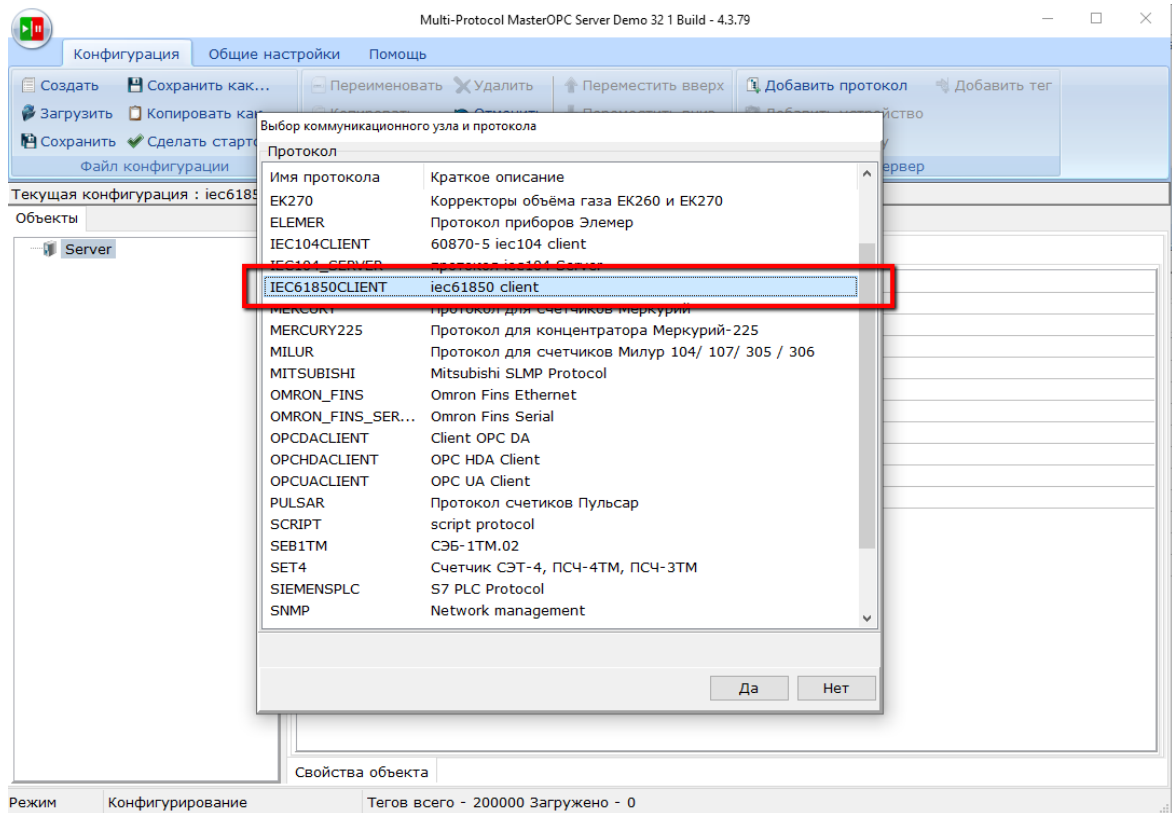


После этого параметры, добавленные в проект, будут доступны клиентам по протоколу IEC61850.

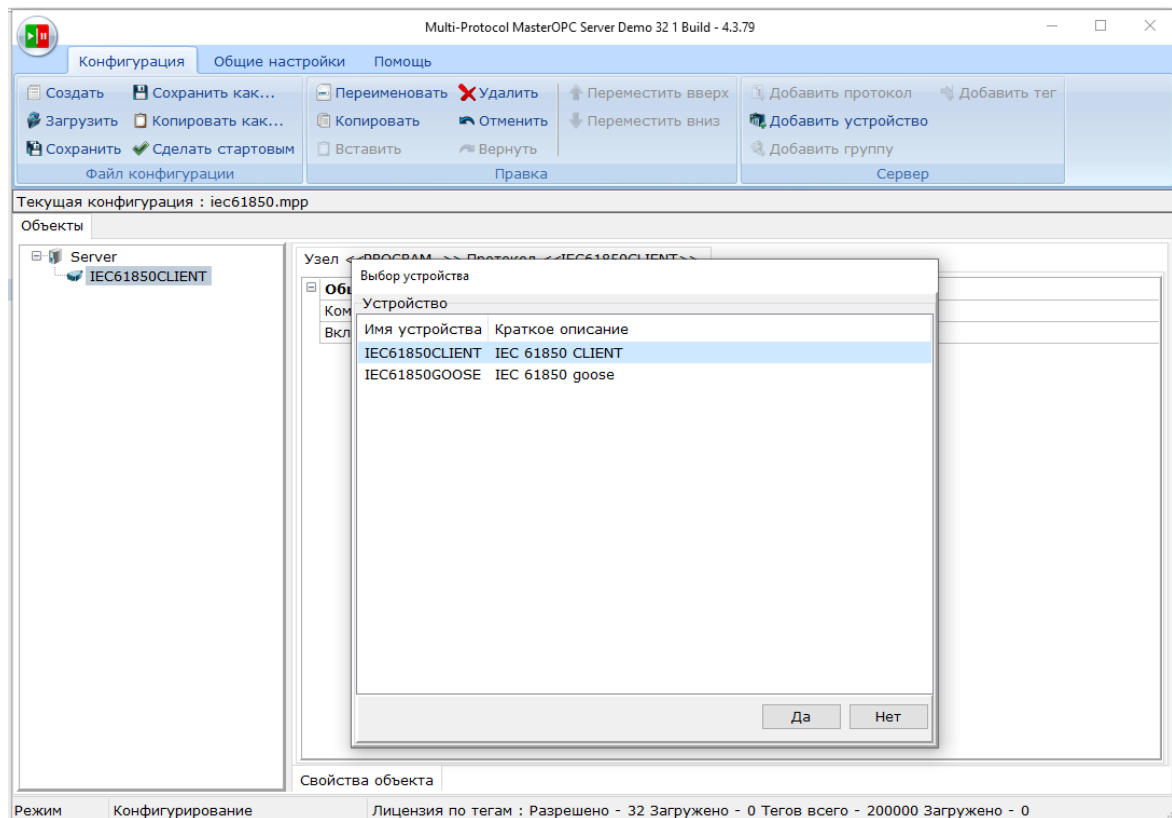
Рассмотрим пример подключения по протоколу IEC61850 приложением Multi-Protocol MasterOPC Server.

Добавим протокол IEC61850:

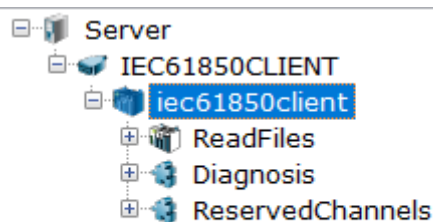




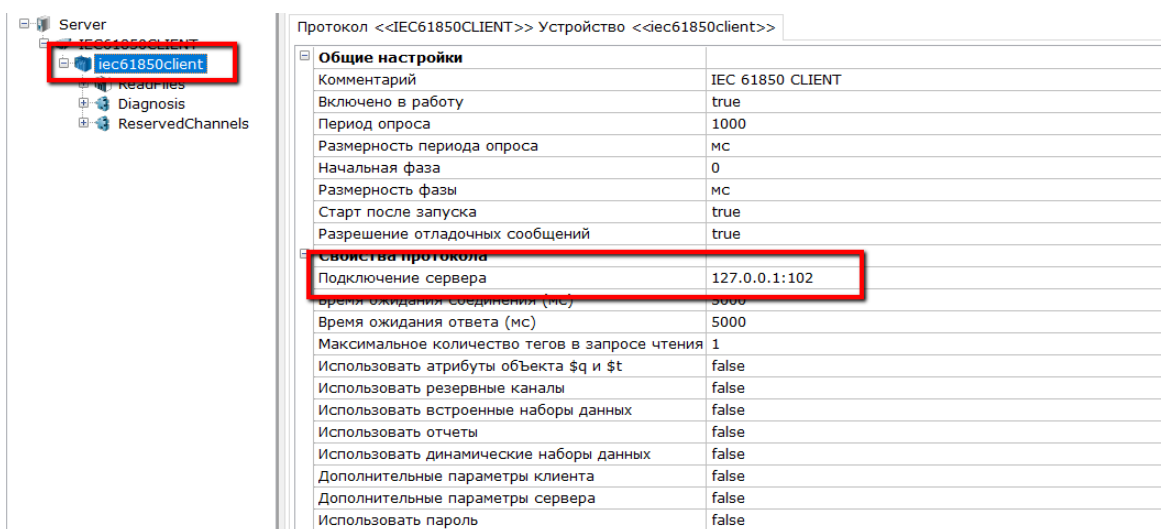
Затем при помощи контекстного меню элемента IEC61850 добавляем устройство:



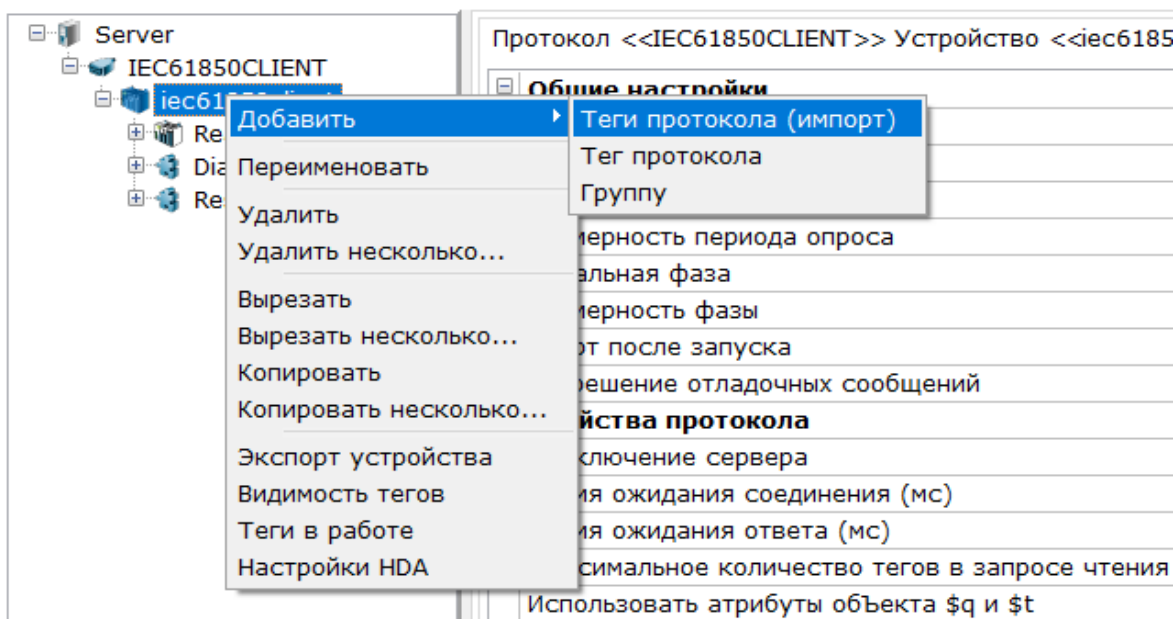
Получим результат:



В настройках устройства необходимо указать IP-адрес сервера и номер порта:

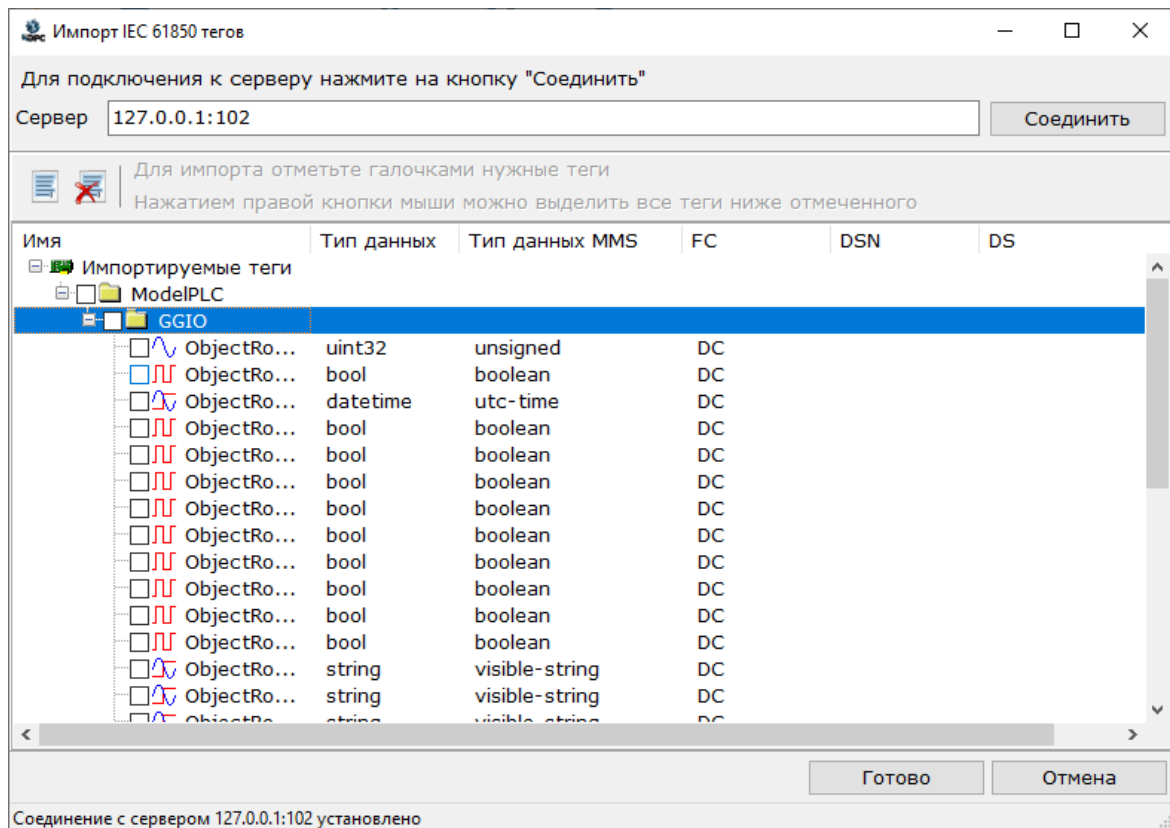


Затем в контекстном меню устройства необходимо выбрать пункт Добавить.Теги протокола (импорт):



Откроется диалоговое окно, в котором нужно нажать кнопку Соединить. После установления связи с исполнительной системой в нижней части окна появится список

параметров, те которые нужно отслеживать необходимо отметить флагами и нажать кнопку Готово:

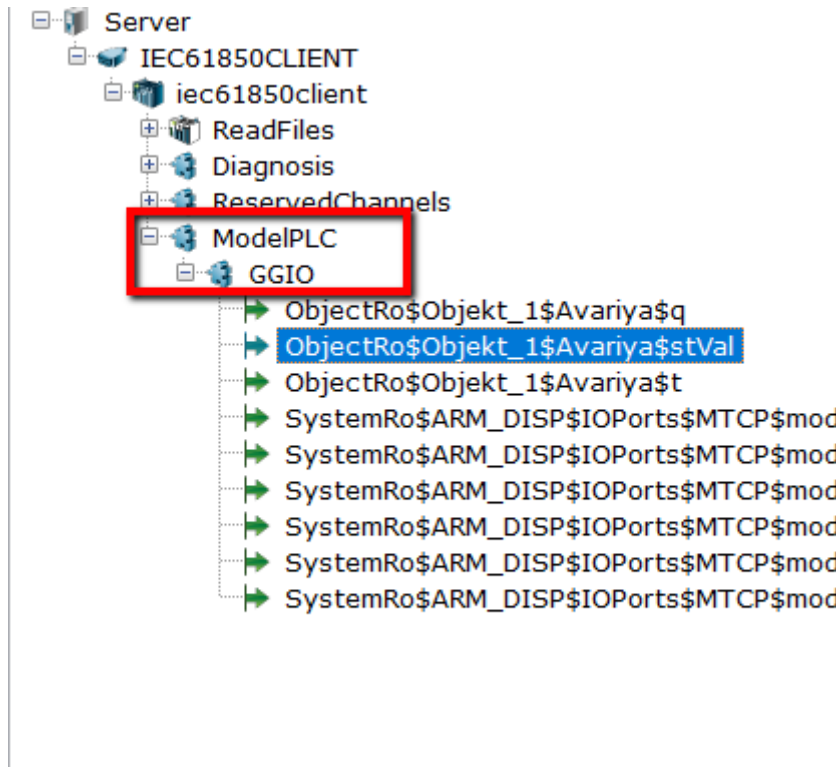


Если установлен флаг В IEC61850 параметры SYSTEM_*_VALUE представлять как t, q, stVal в настройках узла MasterSCADA 4D, то параметры типа SYSTEM_*_VALUE будут представлены в списке тремя параметрами, например:

<input type="checkbox"/>		ObjectRoot\$Objekt_1\$Temperatur\$q	uint32	unsigned
<input type="checkbox"/>		ObjectRoot\$Objekt_1\$Temperatur\$stVal	float	float
<input type="checkbox"/>		ObjectRoot\$Objekt_1\$Temperatur\$t	datetime	utc-time

Если какие-то параметры не видны в клиенте, то возможно причина в том, что установлено недостаточное ограничение в настройках узла MasterSCADA 4D, в поле Ограничение длины имени узла в IEC61850.

После нажатия на кнопку Готово в клиенте появится новый узел с выбранными параметрами:



После этого можно приступать к опросу. Для этого необходимо запустить приложение Multi-Protocol MasterOPC Server

6.2.3.4. ТИПОВЫЕ КАНАЛЫ ОБМЕНА

По умолчанию библиотека MasterSCADA 4D содержит следующие каналы:

Название	Тип данных параметров	Доступ
AI	SYSTEM_LREAL_PARAM	Чтение
DI	SYSTEM_BOOL_PARAM	Чтение
AO	SYSTEM_LREAL_PARAM	Запись
DO	SYSTEM_BOOL_PARAM	Запись
AIO	SYSTEM_LREAL_PARAM	Чтение и запись
DIO	SYSTEM_BOOL_PARAM	Чтение и запись

Эти структуры содержат параметр StatusCode (статус), начальное значение которого установлено в OpсUa_BadWaitingForInitialData = 0x80320000 (2150760448) для того, чтобы

до получения значения по связи не выполнялась запись начального значения в архив, а также выдача начального значения на выходные каналы. Если в дальнейшем опрос произойдет успешно, то значение параметра поменяется на Good (0). Если при первом опросе возникнет ошибка, то параметр установится в OpcUa_BadNoCommunication 0x80310000 (2150694912). Если при опросах сначала будет установлено Good, а потом возникнет ошибка, то параметр примет значение OpcUa_BadOutOfService 0x808D0000 (2156724224) (при этом в Value останется последнее полученное значение).

6.2.3.5. ОТКАЗ

Данный параметр входит в состав всех модулей ввода-вывода. Он принимает значение TRUE в случае отказа соответствующего оборудования или разрыва связи с ним, в остальных случаях – False.

Этот параметр также присутствует у протоколов, которые не содержат модулей ввода-вывода.

Условия, когда отказ протокола снимается:

- В настройках протокола снят флаг Выполнять на резервном
- Узел находится в режиме резервного и с момента переключения в этот режим прошло время, указанное в настройке Задержка сброса отказа на резервном.
- На узле MASTER не установлен отказ (ни от какого протокола и нет программного отказа)

6.2.3.6. НОВЫЕ ТИПЫ ПРОТОКОЛОВ И МОДУЛЕЙ

Раздел находится в разработке

6.2.3.7. РЕКОМЕНДАЦИИ ПО НАСТРОЙКИ DCOM

DCOM Windows настраивается в том случае, если среда разработки или среда исполнения находятся на одном компьютере, а OPC DA или HDA сервер на другом, в одной локальной сети.

Инструкция по настройке DCOM поставляется, как правило, вместе с OPC-сервером.

Если необходимо связать среду исполнения и OPC DA/HDA сервер через Internet, то можно использовать OPC-серверы компании "МПС Софт", которые могут преобразовывать стандарт OPC DA/HDA в UA.

6.3. ДЕРЕВО ОБЪЕКТОВ

В дереве объектов находятся все необходимые инструменты для создания логической и графической части проекта. В объектах можно создавать параметры для связи с деревом

системы, а затем эти параметры использовать в различных программах, окнах и отчетах, также созданных в объектах.

Первым в данном дереве создается Объект.

Объект представляет собой именованную совокупность графического представления технологического объекта, его параметров, алгоритмов контроля и управления, окон управления и других доступных элементов проекта (в том числе, объект может содержать другие объекты). Объект – это эквивалент реального физического объекта (датчика, клапана, аппарата, цеха и т.д.) в проекте. Каждый объект может иметь внутри себя параметры, связанные с каналами, находящимися в дереве системы, окна, описывающие технологический объект, программы, в которых обрабатываются параметры и другие элементы. Для объекта могут создаваться тревоги (аварийные сообщения).

В случае, если проект состоит из нескольких типовых объектов, например, в цеху есть три однотипные линии, которые включают в себя набор одинаковых конвейеров, то целесообразно сначала создать в библиотеке типовой объект Конвейер. Этот объект следует включить в другой типовой библиотечный объект Линия, который уже можно добавить в дерево объектов. За счет тиражирования ускорится разработка, а за счет использования наследования упростится дальнейшее внесение правок.

6.3.1. ТИПЫ И ЭКЗЕМПЛЯРЫ MASTERSCADА 4D

В MasterSCADA 4D широко используются принципы ООП (объектно-ориентированного программирования).

Тип – это библиотечный элемент проекта (окно, параметр, объект, тревога и т.п.), который содержит в себе все необходимые настройки.

Экземпляр – это элемент дерева системы, дерева объектов или дерева библиотек, который произошел (имеет отношения) из библиотечного типа. Это значит, что экземпляр изначально имеет все настройки такие же, как у типа. Поведение экземпляра в проекте зависит от его отношений с типом.

Если какая-то часть проекта (программа, окно, часть окна, объект) повторяется несколько раз в одном проекте или встречается в нескольких проектах, то в дерево системы и в дерево объектов обычно добавляются экземпляры, а их типы находятся в дереве библиотек. Элемент может быть добавлен в проект любым удобным способом:

- при помощи контекстного меню;
- при помощи контекстной панели;
- перетаскиванием из дерева библиотек;
- при помощи легенды какого-либо редактора.

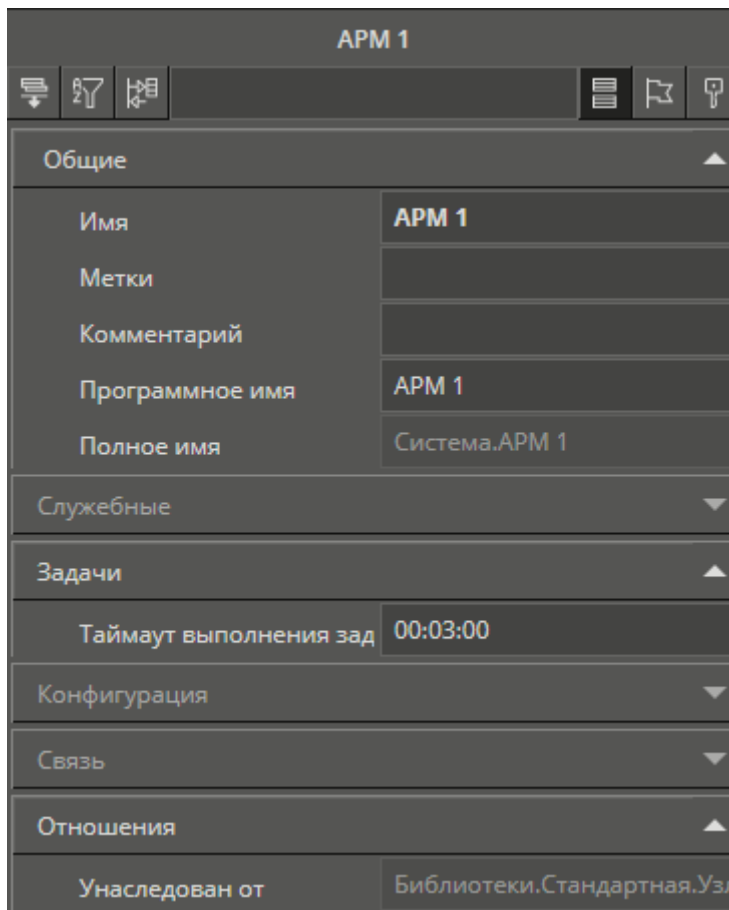
У элементов дерева системы и дерева объектов в панели свойств в группе Отношения показано, с каким именно библиотечным типом имеется связь. Отношения между типом и экземпляром могут быть разные: Ссылается, Имеет тип, либо Унаследован от.

Рассмотрим, для каких элементов проекта какие отношения являются наиболее характерными.

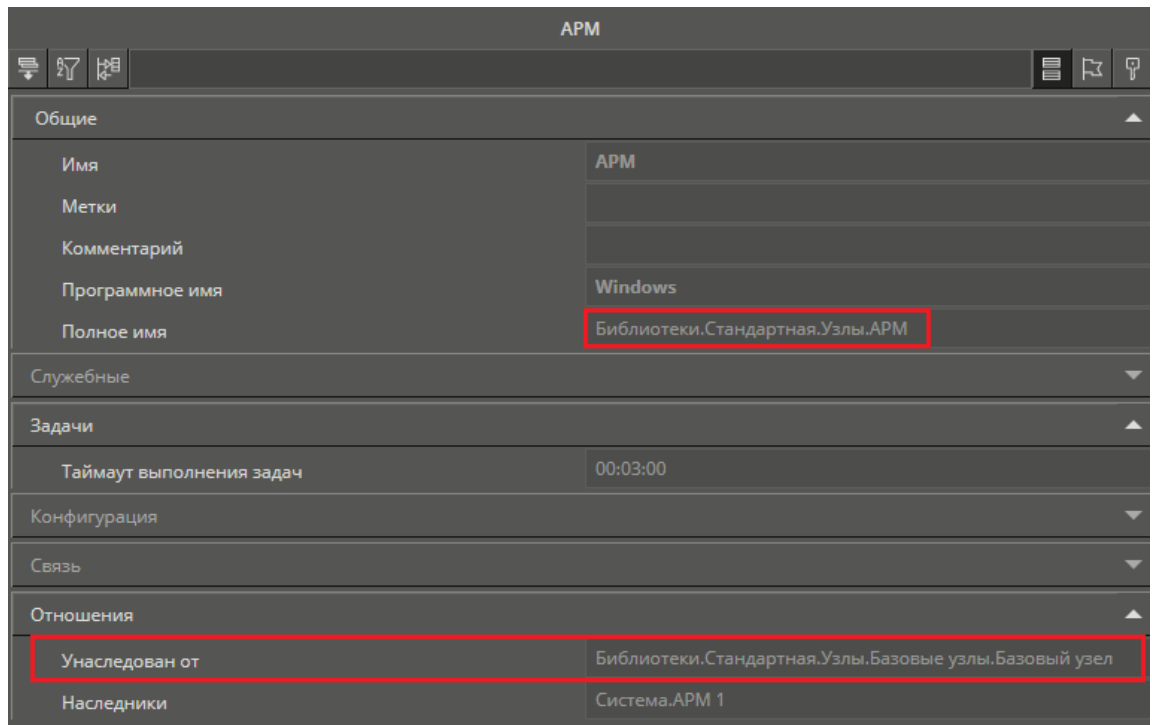
Отношение	Элементы	Описание
Имеет тип	Параметры	При назначении параметру библиотечного типа данных, в дерево автоматически добавляется необходимое количество полей структуры, в зависимости от библиотечного элемента. При изменении библиотечного типа данных, автоматически изменится вид и настройки параметра в дереве системы или в дереве объектов. Разработчик проекта может изменить свойства параметра, но не его внутреннюю структуру.
Ссылается	Программы, Окна, Экземпляры объектов	Если экземпляры объектов, окна или программы изначально были созданы в библиотеке, а затем используются в объектах, в узлах, в каналах или в тегах, то в этом случае экземпляр не содержит в себе текстов, схем программ или окон, т.к. они хранятся только в типе. Экземпляр содержит в себе только элементы, необходимые для связи с другими элементами проекта. Это позволяет значительно уменьшить размер создаваемого проекта и увеличить скорость работы в режиме разработки. Разработчик не может изменить внутреннюю структуру отдельного экземпляра после его добавления в проект.
Унаследован от	Объекты, каналы, теги, узлы	В библиотеке создается тип, который при добавлении в дерево системы или в дерево объектов может быть отредактирован разработчиком. Например, можно добавить окно, программу, изменить список параметров и т.п.
Нет свойства	Параметры, Объекты, Программы, Окна	Характерно для элементов, которые добавлены в проект в базовом виде. Например, параметр, у которого в настройках выбран простой тип данных, или объект, не содержащий по умолчанию окон, программ и параметров, или стандартные функциональные блоки, функции и др. Исключение составляют дочерние элементы экземпляра объекта; у них нет данного свойства, т.к. они являются не

Отношение	Элементы	Описание
		самостоятельными элементами, а ссылками на элементы типа.

Например, если добавить в дерево системы узел АРМ, то у него будет указано:



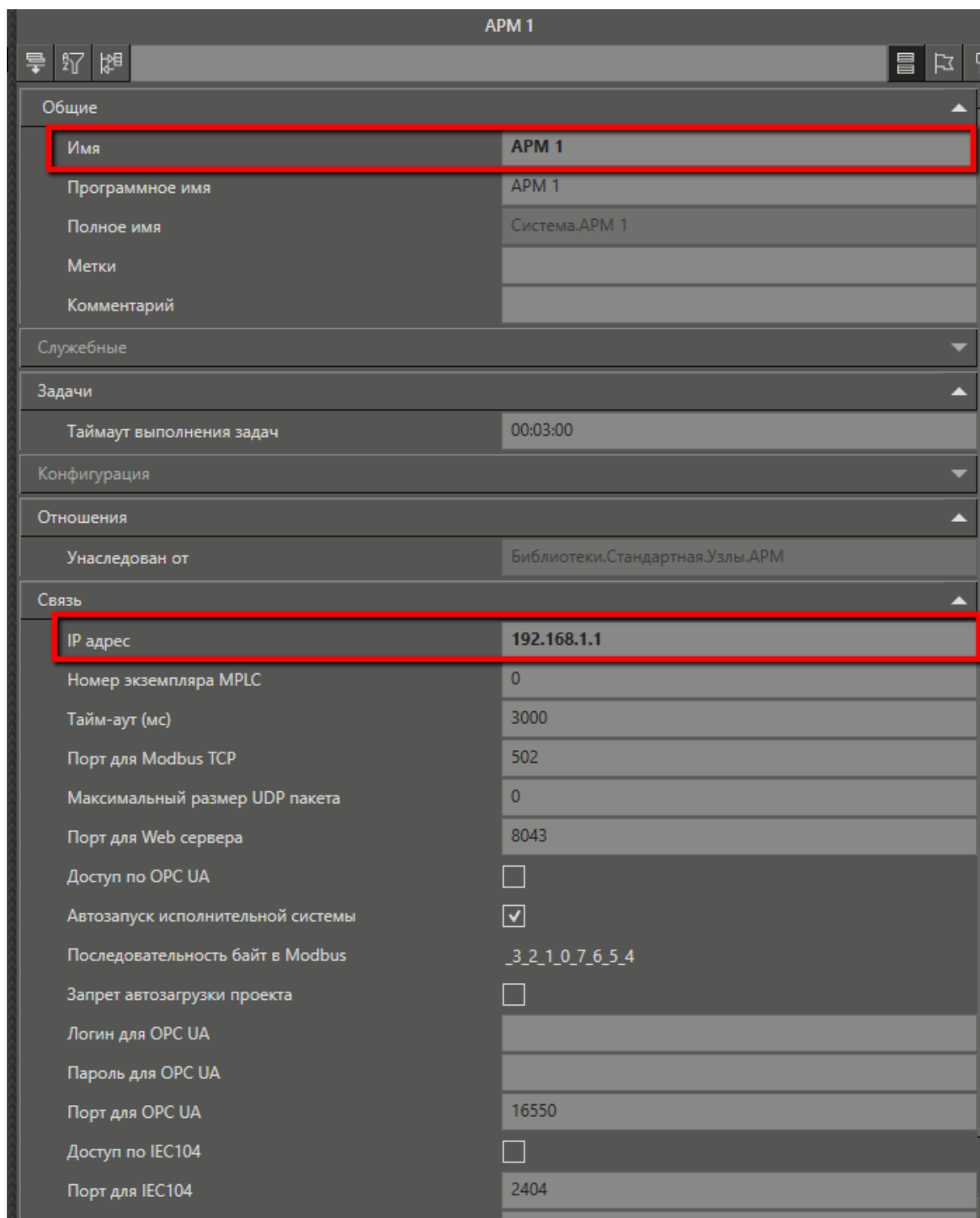
Для расширения базового функционала типы могут наследоваться друг от друга. Рассмотрим узел АРМ в дереве библиотек. Несмотря на то, что он сам является типом (и мы можем добавить его в проект), в его настройках видно, что он унаследован от другого типа. От Базового узла он отличается именем и рядом предопределенных настроек.



Разработчики проектов могут создавать свои собственные типы в пользовательских библиотеках.

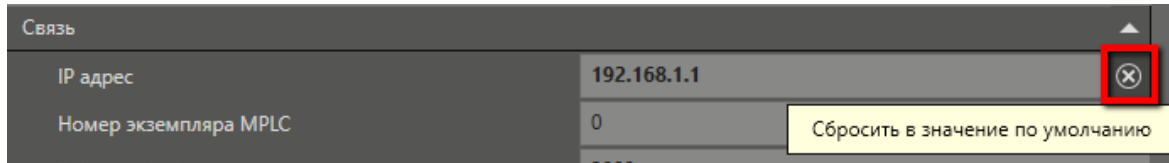
Важно! Если изменить библиотечный тип, то это приведет к изменению всех его экземпляров в проекте, за исключением тех конкретных свойств, окон и программ, которые уже были модифицированы.

При анализе элементов необходимо обращать внимание на панель свойств. В ней свойства, которые не менялись разработчиком (т.е. которые совпадают со свойствами типа), отображаются обычным шрифтом, а измененные – жирным.



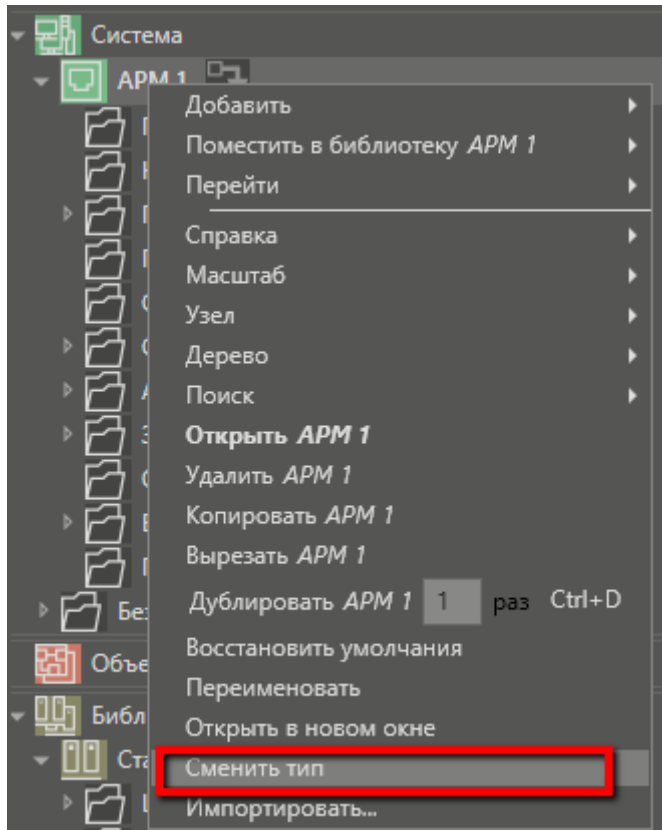
Это означает, что если настройку IP-адреса изменить в типе, то она не поменяется в проекте автоматически.

Настройки и составные части экземпляра можно синхронизировать с типом несколькими способами. Во-первых, если выполнить пункт контекстного меню элемента Восстановить умолчания. В этом случае все дочерние группы и свойства элемента синхронизируются с типом. Данный пункт меню можно применить не ко всему элементу, а только к его отдельным частям, которые были выделены. Во-вторых, если нажать на кнопку, которая появляется при наведении курсора справа от того или иного свойства:

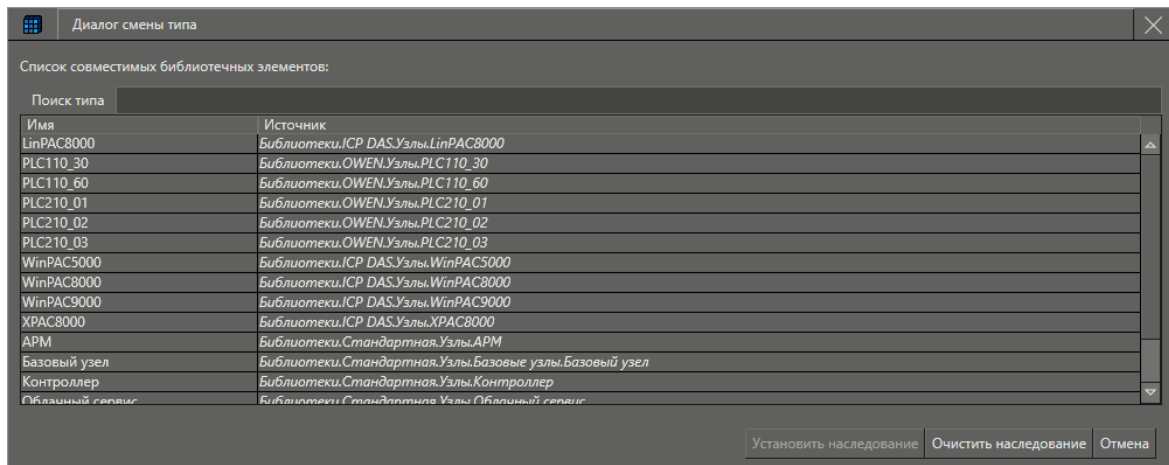


Это приведет только к синхронизации текущего свойства.

В случае необходимости, можно сменить тип. Для этого в контекстном меню элемента нужно выбрать соответствующий пункт Сменить тип.

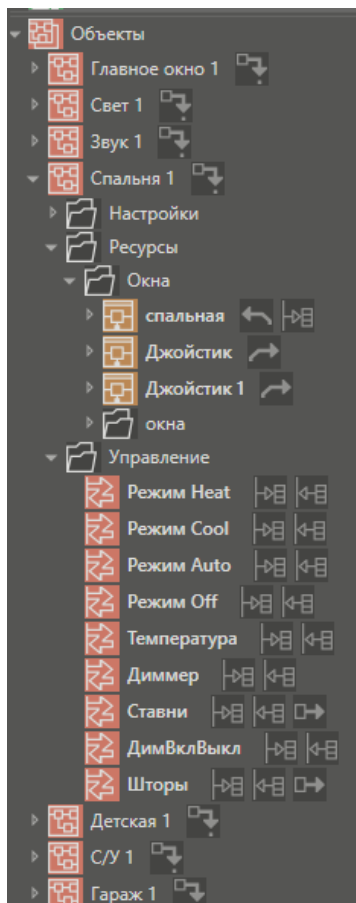


После чего появится диалоговое окно, в котором можно будет выбрать допустимый тип:

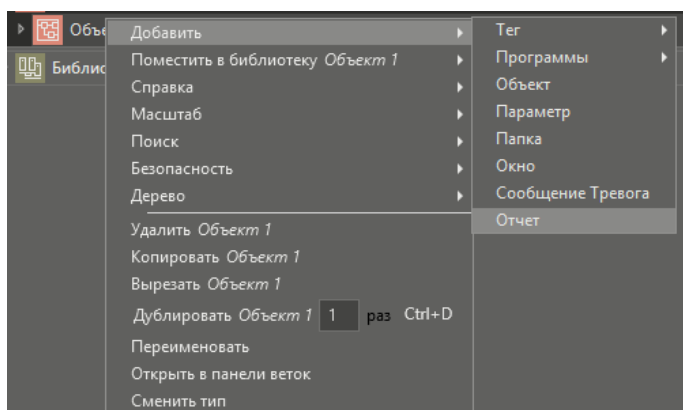


6.3.2. ОБЪЕКТ

Объекты – это структурные единицы, позволяющие разработчику проекта оперировать технологическими понятиями.



Объект представляет собой именованную совокупность графического представления технологического объекта, его параметров, алгоритмов контроля и управления, окон управления и других доступных элементов проекта (в том числе, объект может содержать другие объекты). Элементы могут быть добавлены в объект, в частности, с помощью команды контекстного меню **Добавить** или контекстной панели:



Объект не может содержать Каналы, обеспечивающие связи с модулями ввода-вывода или OPC-серверами. Каналы могут быть созданы только в определенных группах дерева

системы). Т.е. объекты MasterSCADA 4D позволяют разработчику создавать логическую часть проекта как набор реальных технологических объектов, но при этом в узлах могут конфигурироваться только внешние связи узлов.

Для связи с каналами в объекте могут использоваться элементы Тег и Параметр.

Параметр объекта обеспечивает внешние связи этого объекта в рамках проекта.

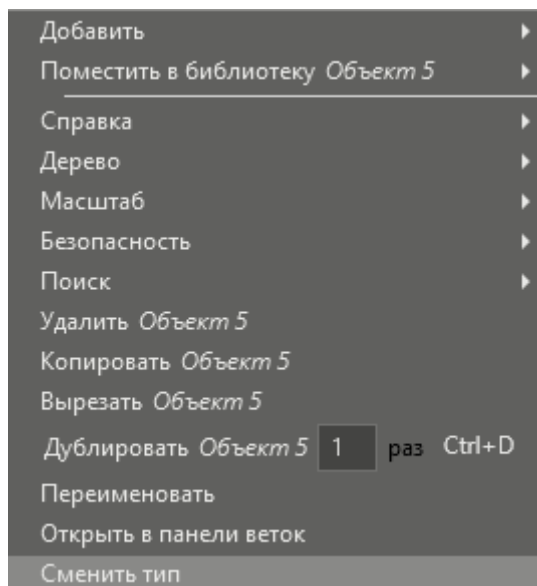
Использование библиотечных объектов в дереве объектов

Если объект используется в проекте неоднократно, то необходимо воспользоваться механизмом типизирования элементов проекта.

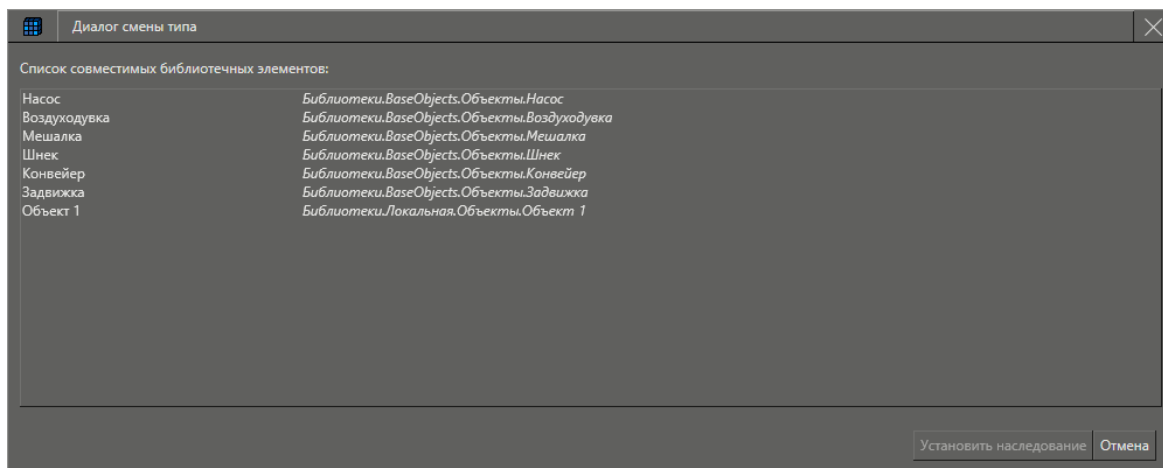
Типизация обеспечивает многократное использование одного и того же объекта как шаблона в рамках одного проекта, так и при разработке других систем. При изменении типа объекта в библиотеке, наследование обеспечивает изменение всех объектов – экземпляров этого типа – в проекте. Это значительно упрощает задачу тиражирования и модифицирования объектов.

Вариант 1

Если объект, созданный в библиотеке, в дальнейшем требуется модифицировать (добавлять параметры, окна, менять программы и т.п.), то необходимо добавить в дерево объектов элемент Объект, а затем в контекстном меню выбрать пункт Сменить тип:



Откроется диалоговое окно, в котором следует выбрать нужный библиотечный объект:



В этом случае тип отношения между библиотечным объектом и экземпляром будет Унаследован от. При таком типе отношений объект может быть изменен в дереве объектов.

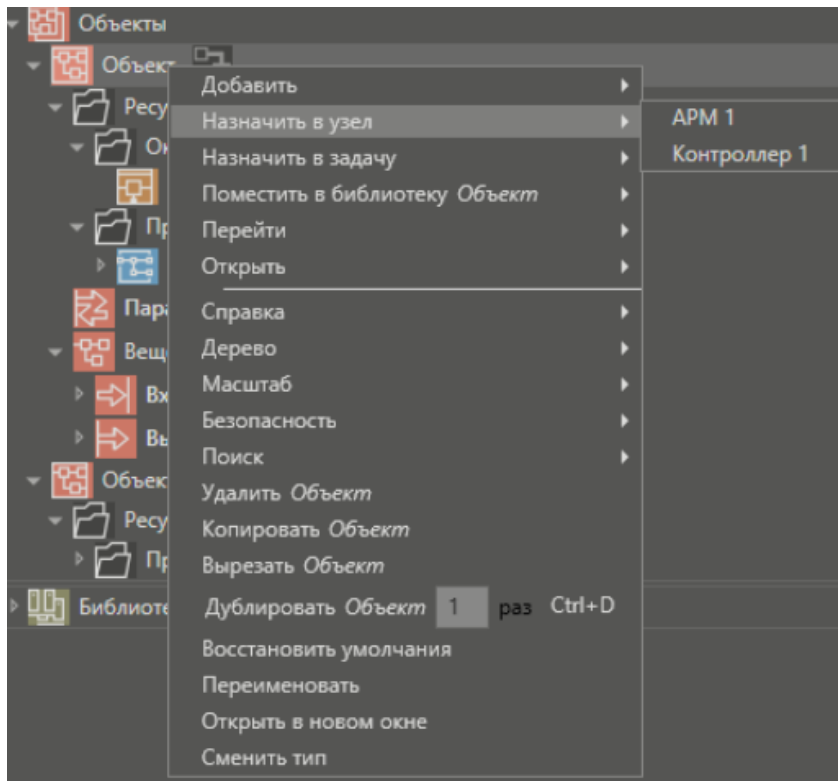
Важно! В версии 1.2.4 при перетаскивании объекта из библиотеки левой кнопкой мыши добавляется в проект объект с типом отношения Унаследован от.

Вариант 2

Если перетащить объект из библиотеки в дерево объектов правой кнопкой мыши и в появившемся списке выбрать Вставить экземпляр , то вставится элемент Экземпляр объекта, который нельзя изменить в дереве объектов. Тип отношения между библиотечным объектом и экземпляром будет Ссылается.

Место исполнения объекта

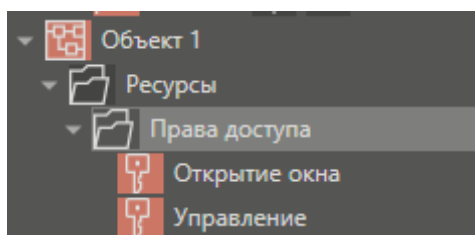
Для того чтобы задать место исполнение объекта, предусмотрены пункты контекстного меню Назначить в узел и Удалить из узла , а для распределения объектов по задачам – команды Назначить в задачу и Удалить из задачи.



Более подробно о назначении объектов в узлы описано в разделе Задачи.

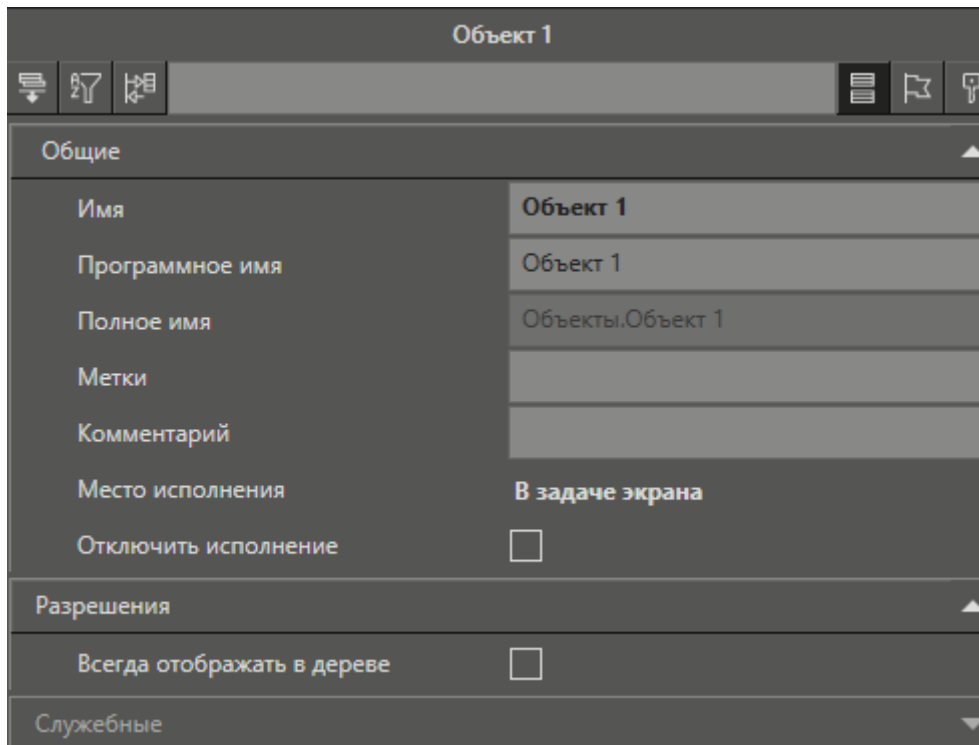
Права доступа операторов

Для объекта могут быть назначены права доступа для пользователей; группа Права доступа размещается в группе Ресурсы (см. также Задание прав пользователей). Если выбрать в дереве группу Права доступа, то в панели свойств объекта открывается вкладка Панель Свойств. Права



6.3.2.1. ПАНЕЛЬ СВОЙСТВ ОБЪЕКТА

Панель свойств Объекта имеет вид:



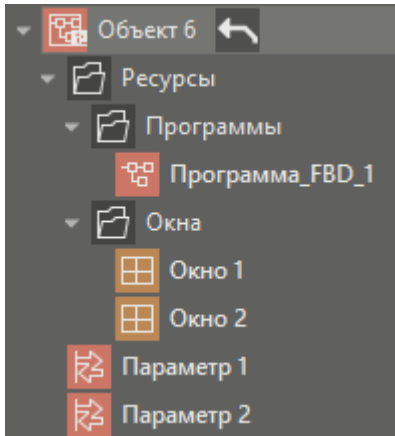
Категория	Рекомендации
Общие	В данной категории у объекта, как правило, редактируются свойства Имя, Место исполнения.
Разрешения	Данную категорию имеет смысл настраивать только у вложенных объектов, которые необходимо отображать в дереве, несмотря на то, что у них содержимое родительского объекта скрыто от других разработчиков проекта.

6.3.3. ЭКЗЕМПЛЯР ОБЪЕКТА

Экземпляр объекта - это элемент дерева объектов, который появляется путем вставки объекта из библиотеки. Имеет отношения с типом Ссылается. Экземпляр содержит в себе только элементы, необходимые для связи с другими элементами проекта. Это позволяет значительно уменьшить размер создаваемого проекта и увеличить скорость работы в режиме разработки. Разработчик не может изменить внутреннюю структуру отдельного экземпляра после его добавления в проект. Экземпляр объекта содержит ссылки на окна и программы.

Изменение типа в библиотеке приведет к изменению всех его экземпляров.

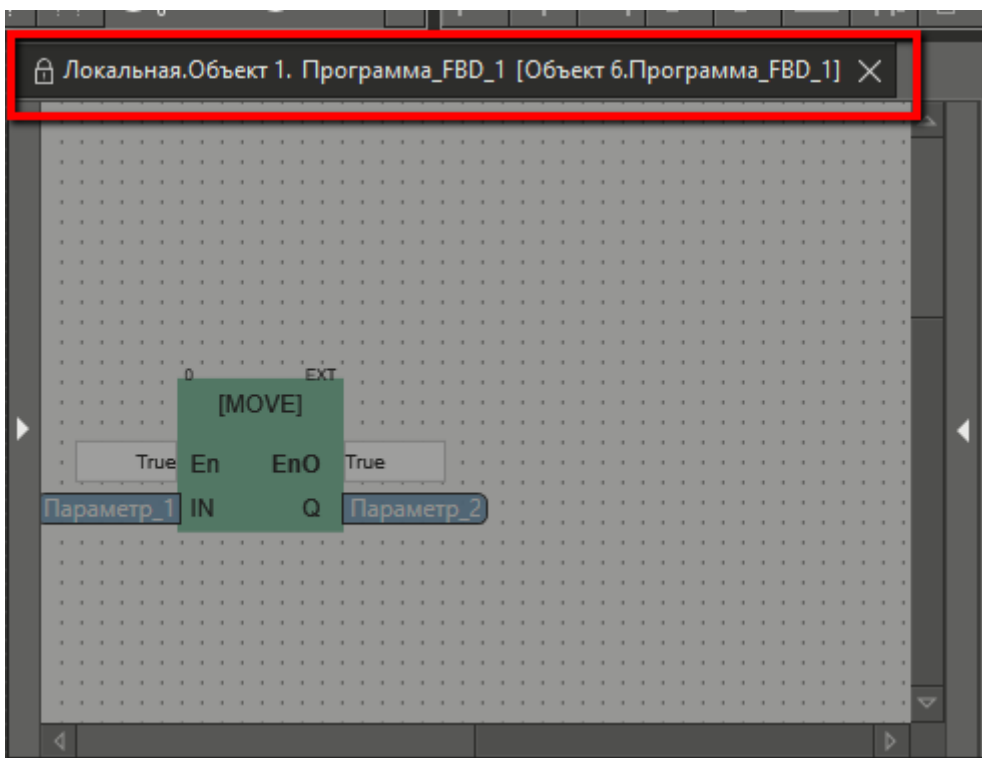
Вид экземпляра объекта в дереве:



Использование ссылок на окна и программы

Если дважды нажать на ссылку на окно или на программу экземпляра объекта, то откроется соответствующий редактор, но редактирование будет недоступно.

Вид:



Заголовок открытого окна будет иметь вид: Путь к типу в библиотеке.[Путь к экземпляру, ссылке на который открыли].

Ссылка на окно

Экземпляр объекта (или его окно, а точнее ссылку на него) можно перетащить в какое-либо окно, так же как и элемент Объект, левой или правой кнопками мыши.

Ссылку на окно экземпляра объекта можно назначить стартовым окном узла.

Ссылка на программу

Если среда разработки подключена к среде исполнения, то при нажатии на ссылку на программу экземпляра объекта откроется редактор, в котором можно отслеживать текущие значения, характерные для конкретного экземпляра объекта, что может помочь в отладке программы.

Свойства экземпляра объекта

Свойства экземпляра объекта такие же, как и у элемента Объект. Аналогично настраиваются и права доступа.

Сохранение данных горячего рестарта

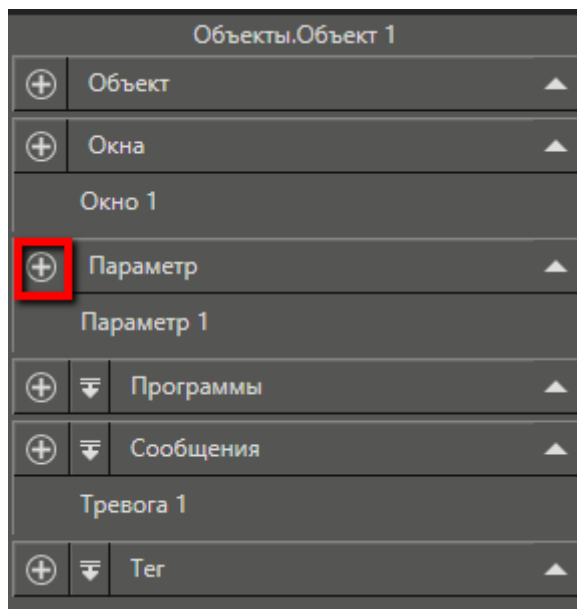
Если у экземпляра объекта в дереве объектов стоит Сохранять=Да, то будут сохраняться все параметры экземпляра (включая локальные и вложенные экземпляры ФБ), кроме тех, у которых стоит Сохранять=Нет.

Значение Наследуется в дереве объектов означает, что будут сохраняться только те у которых установлено значение свойства Да

6.3.4. ПАРАМЕТРЫ ОБЪЕКТОВ

Параметры в объектах создаются для связи объекта с другими объектами, либо узлами.

Добавить параметр в объект можно при помощи контекстного меню либо контекстной панели.



Если два параметра принадлежат двум объектам, которые исполняются в разных узлах, то передача данных между узлами будет осуществляться автоматически, с периодом, установленным в настройках службы Межузловая связь.

6.3.5. ПАПКА

Назначение этой группы – дополнительное структурирование проекта. Папка не имеет специальных настроек.

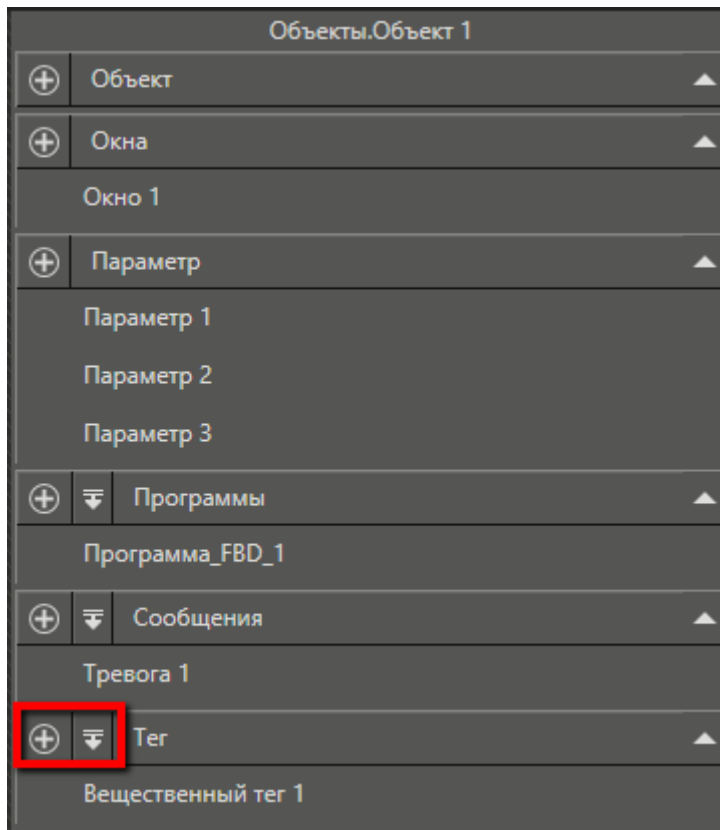
6.3.6. ТЕГ

Тег - это элемент дерева объектов, объединяющий в себе возможности параметра и объекта. Тег, как и параметр, служит для передачи данных между разными элементами проекта, но в отличие от параметра может иметь окна для произвольного отображения данных в других окнах, а также программы и тревоги для обработки полученных данных. Тег содержит по умолчанию два параметра - Вход и Выход: данные со входа передаются на выход. Если перетащить какой-либо параметр на Тег, то данные автоматически будут передаваться на параметр Вход Тега. Если перетащить Тег на какой-либо параметр, то данные будут передаваться с Выхода тега приемнику..

Отличия от элемента Объект:

- Внутренняя структура тега скрыта для разработчика проектов в дереве объектов. Доступны только те параметры тега, которые необходимы для настройки внешних связей.
- В проект можно добавить только тот тег, тип которого задан в библиотеке.

Стандартные теги находятся в дереве библиотек в категории Стандартная. Т.к. теги могут быть связаны с каналами, то в данной библиотеке созданы типы тегов, параметры которых соответствуют параметрам типовых каналов обмена (AI_LREAL, DI_BOOL и т.п.). Эта библиотека не отображается в упрощенном дереве, т.к. стандартные теги можно добавить в проект через контекстное меню, либо контекстную панель Объекта:



В тег, находящийся в дереве объектов, можно добавить окно, тревогу или программу при помощи контекстного меню пункта Добавить. Это имеет смысл в двух случаях:

1. Если необходимо создать уникальный тег, который будет использоваться в проекте один раз.
2. Если ведется разработка библиотечного типа тега на базе какого-либо стандартного тега по методу от экземпляра к типу

6.3.6.1. ПАНЕЛЬ СВОЙСТВ ТЕГА

Вид панели свойств:

Вещественный тег 1

☰
🔍
📄

Общие ▲

Имя	Вещественный тег 1
Программное имя	Вещественный тег 1
Полное имя	Объекты.Насосная.Вещественный тег 1
Метки	
Комментарий	

Служебные ▼

Отношения ▲

Унаследован от	Библиотеки.Стандартная.Теги.Вещественн
----------------	--

Шкала ▲

Шкала	
-------	--

Описание

Название	Назначение
Категория Общие	Задается имя узла и другие стандартные настройки. Подробное описание приведено в разделе справки: Категория свойств Общие.
Категория Отношения	Показывает экземпляром какого типа является данный элемент. Подробное описание приведено в разделе: Типы и экземпляры MasterSCADA 4D
Шкала	Можно выбрать шкалу для тега, для автоматического формирования сообщений при нарушении границ.

6.3.6.2. СТАНДАРТНЫЕ ТЕГИ

Группа Теги библиотеки Стандартная содержит встроенные типы тегов:

- Вещественный тег;
- Целый тег;
- Логический тег.

В группе Теги пользовательской библиотеки создаются типы тегов пользователя.

6.3.6.2.1. ВЕЩЕСТВЕННЫЙ ТЕГ

Тег , в котором тип данных параметров Вход и Выход – SYSTEM_LREAL_PARAM. Данные со входа автоматически передаются на выход

6.3.6.2.2. ЦЕЛЫЙ ТЕГ

Тег , в котором в котором тип данных параметров Вход и Выход – SYSTEM_DINT_PARAM. Данные со входа автоматически передаются на выход.

6.3.6.2.3. ЛОГИЧЕСКИЙ ТЕГ

Тег, в котором в котором тип данных параметров Вход и Выход – SYSTEM_BOOL_PARAM. Данные со входа автоматически передаются на выход

6.3.6.3. НОВЫЙ ТИП ТЕГА

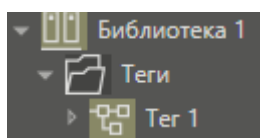
Новый тип тега создается в том случае, если нельзя решить задачу, используя Параметр, например, когда при получении данных на вход тега, их необходимо обработать в программе, отобразить нестандартным образом в окне, передать результат работы программы на выход, и выдать какое-либо сообщение.

Новый тип тега создается в пользовательской библиотеке.

При создании нового типа тега можно использовать оба метода создания типов.

- От типа к экземпляру
- От экземпляра к типу

Созданные типы тегов будут находиться в разделе пользовательской библиотеки Теги:

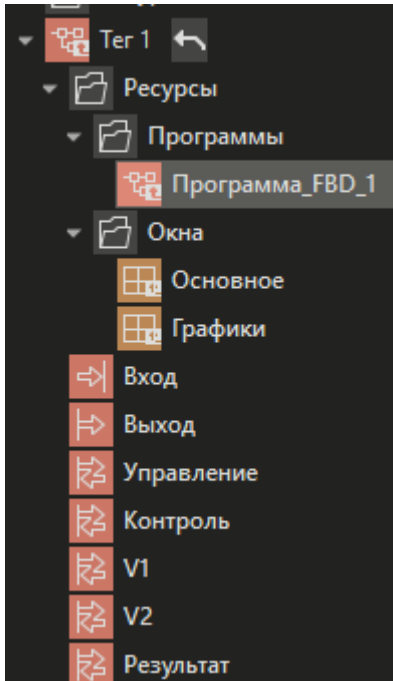


6.3.7. ЭКЗЕМПЛЯР ТЕГА

Экземпляр тега - это элемент дерева объектов, который появляется путем вставки тега из библиотеки. Имеет отношения с типом Ссылается. Экземпляр содержит в себе только элементы, необходимые для связи с другими элементами проекта. Это позволяет значительно уменьшить размер создаваемого проекта и увеличить скорость работы в режиме разработки. Разработчик не может изменить внутреннюю структуру отдельного экземпляра после его добавления в проект. Экземпляр тега содержит ссылки на окна и программы.

Изменение типа в библиотеке приведет к изменению всех его экземпляров.

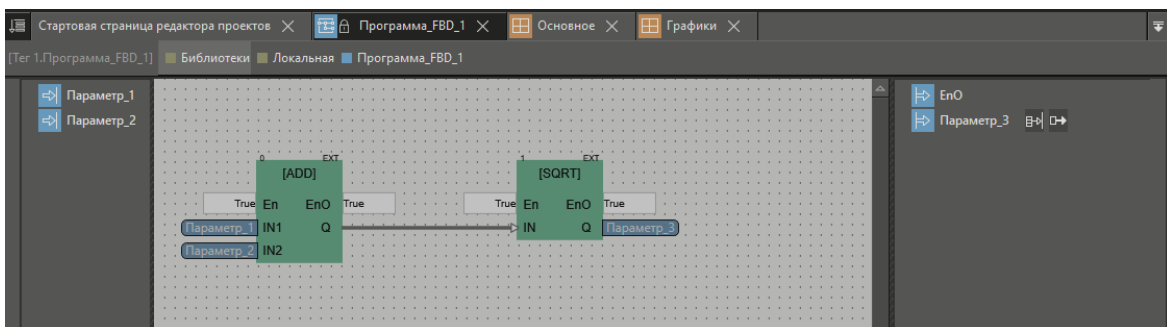
Вид экземпляра объекта в дереве:



Использование ссылок на окна и программы

Если дважды нажать на ссылку на окно или на программу экземпляра тега, то откроется соответствующий редактор, но редактирование будет недоступно.

Вид:



Заголовок открытого окна будет иметь вид: Путь к типу в библиотеке.[Путь к экземпляру, ссылке на который открыли].

Ссылка на окно

Экземпляр тега (или его окно, а точнее ссылку на него) можно перетащить в какое-либо окно, так же как и элемент Тег, левой или правой кнопками мыши.

Ссылка на программу

Если среда разработки подключена к среде исполнения, то при нажатии на ссылку на программу экземпляра тега откроется редактор, в котором можно отслеживать текущие значения, характерные для конкретного экземпляра объекта, что может помочь в отладке программы.

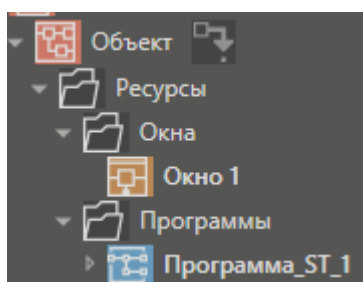
Свойства экземпляра тега

Свойства экземпляра объекта такие же, как и у элемента Тег. Можно настроить индивидуальные права доступа для окон экземпляров тегов.

6.3.8. РЕСУРСЫ

Группа ресурсы может быть у таких элементов проекта как Объект, Тег, Канал.

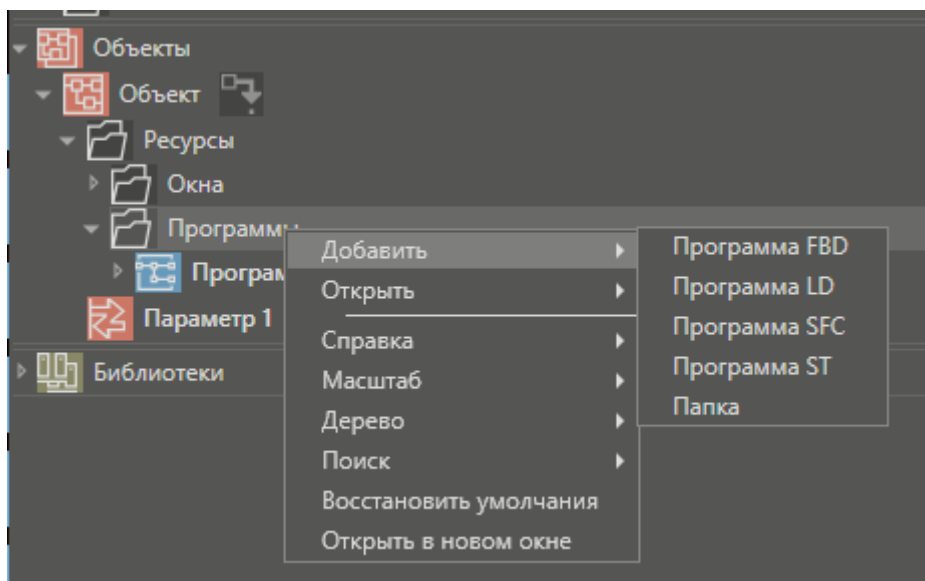
По умолчанию группа Ресурсы не содержит дочерних элементов. После выполнения пункта контекстного меню или контекстной панели Добавить-Программы, Добавить-Окна или Добавить-Сообщения, в ней появятся соответствующие группы: группа Окна, группа Программы, группа Сообщения



Если для объекта (тега) права доступа отличаются от глобальных прав, назначенных в группе Безопасность в дереве системы, то в группу Ресурсы объекта можно добавить индивидуальную группу параметров Права доступа, и настроить их специально для этого объекта (тега).

6.3.8.1. ПРОГРАММЫ

В этой группе отображаются и создаются программы. Для написания программы можно использовать любой из поддерживаемых языков программирования.



В дереве объектов создаются уникальные программы, которые встречаются в проекте однократно, и если эти программы не планируется использовать в других проектах. В противном случае, в дереве объектов следует использовать типизированные программы.

6.3.8.2. ОКНА

В этой группе отображаются и создаются окна.

При добавлении окна автоматически откроется графический редактор.

В дереве объектов создаются уникальные окна, которые в проекте встречаются однократно, и эти окна не планируется использовать в других проектах. В противном случае в дереве объектов следует использовать типизированные окна.

6.3.8.3. СООБЩЕНИЯ

В данной группе находятся сообщения объектов.

В группу можно добавлять экземпляры стандартного сообщения Тревога, а также экземпляры пользовательских сообщений.

6.3.8.4. ОТЧЕТЫ

MasterSCADA 4D поддерживает создание отчетов как на основе текущих данных, так и на основе архивных данных.

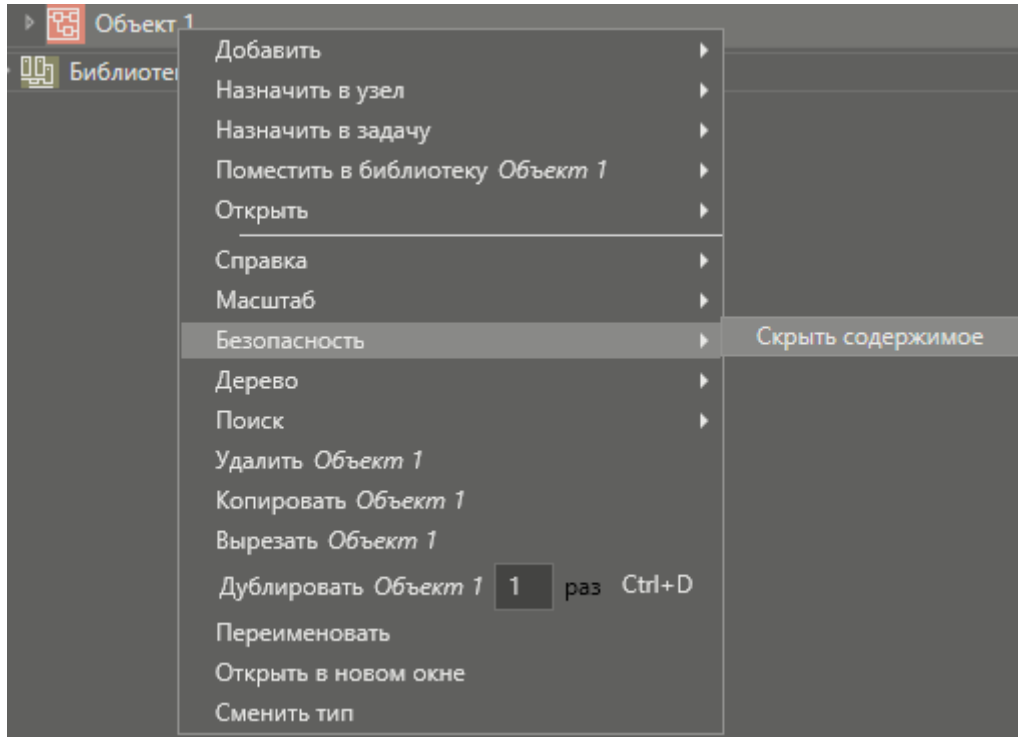
В этой группе создаются отчеты, которые в дальнейшем редактируются в дизайнера отчетов.

После создания отчета, дизайнер отчетов открывается автоматически. Для повторного открытия необходимо дважды нажать на созданный элемент, или воспользоваться пунктом контекстного меню Редактировать.

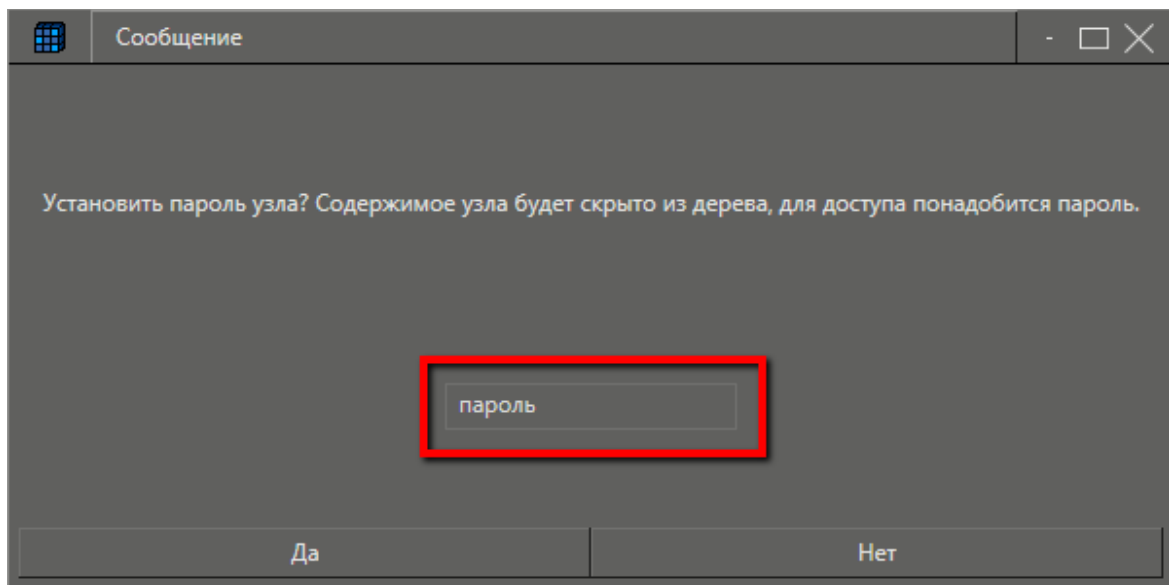
Группа в узле не отображается до тех пор пока не будет добавлен хотя бы один отчет. Добавить отчет в узел в этом случае можно при помощи контекстного меню или контекстной панели.

6.3.9. УСТАНОВКА ПАРОЛЯ НА СОДЕРЖИМОЕ ОБЪЕКТОВ

В контекстном меню объекта предусмотрена команда Безопасность.Скрыть содержимое:

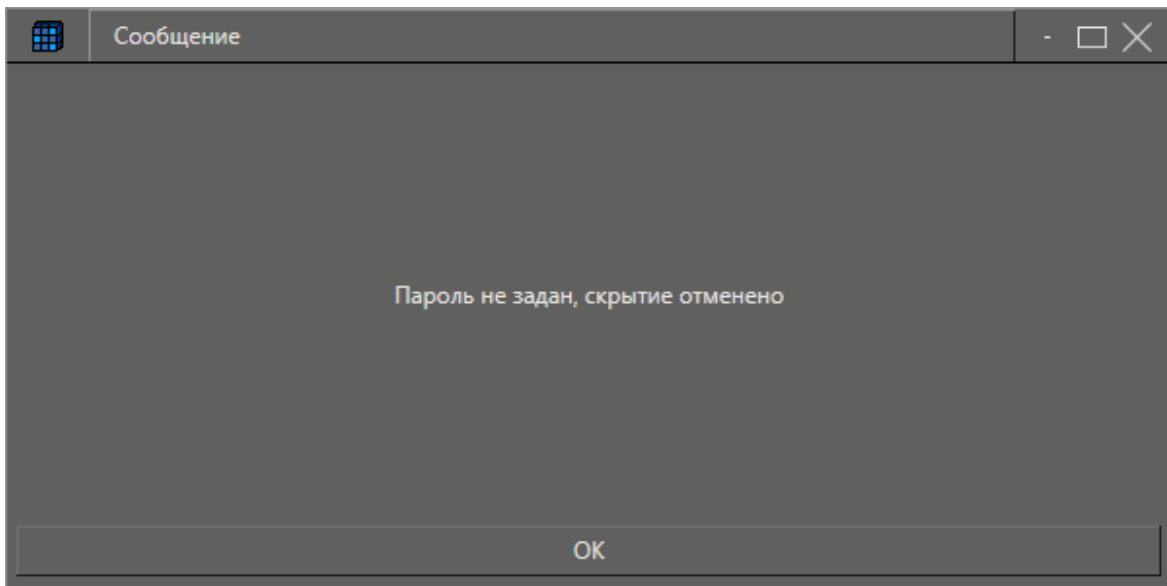


По этой команде открывается диалоговое окно, в котором можно задать пароль:

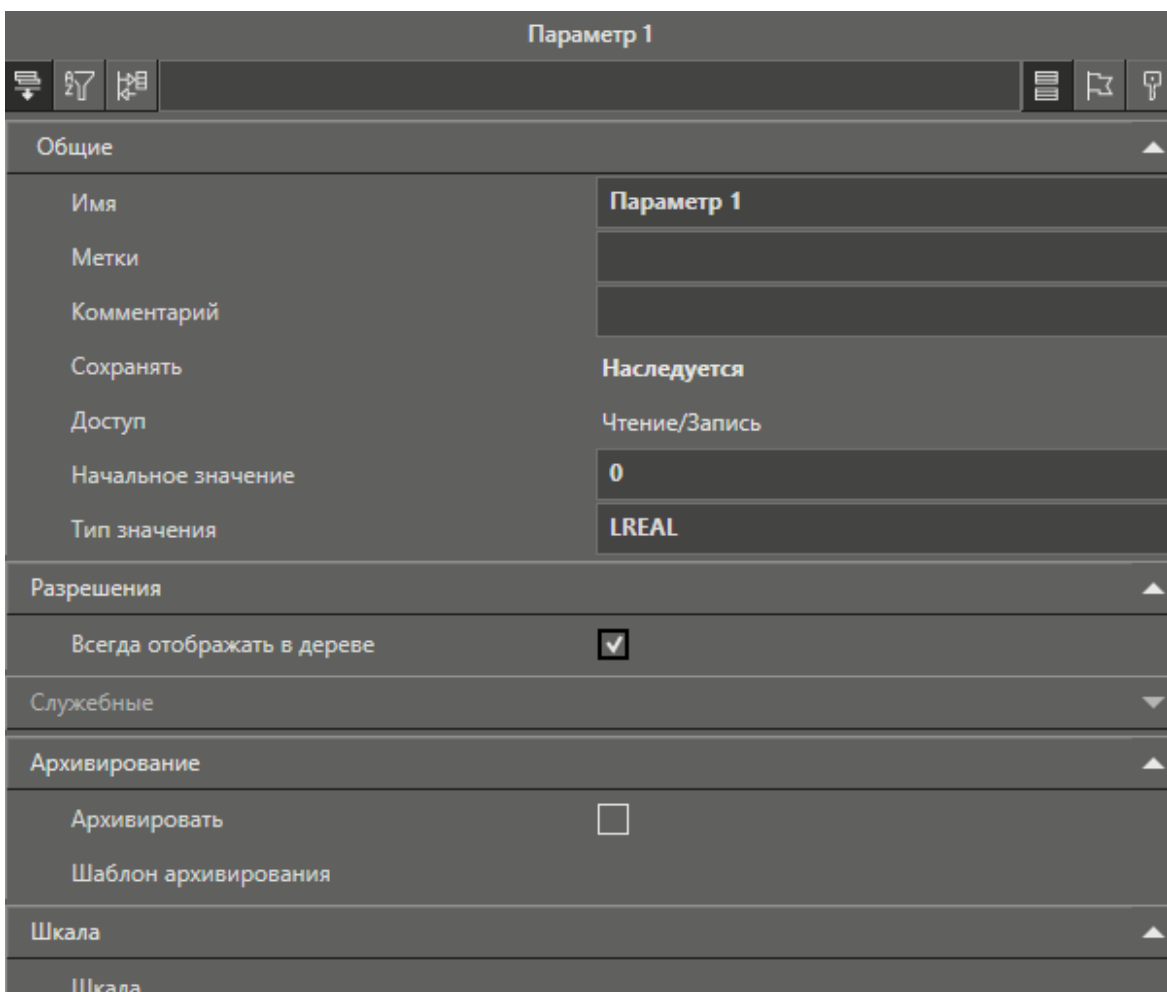


Если нажать кнопку Нет, то диалоговое окно закрывается.

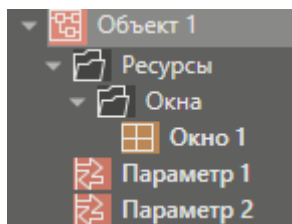
Если нажать кнопку Да, не задав пароль, то окно закрывается, и открывается сообщение об отмене скрытия содержимого объекта:



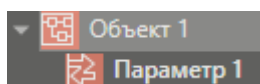
Если задать пароль и нажать кнопку Да, то содержимое объекта скрывается, и в дереве остается видимым только собственно элемент Объект, а также элементы проекта, у которых в панели свойств установлен флаг Всегда отображать в дереве:



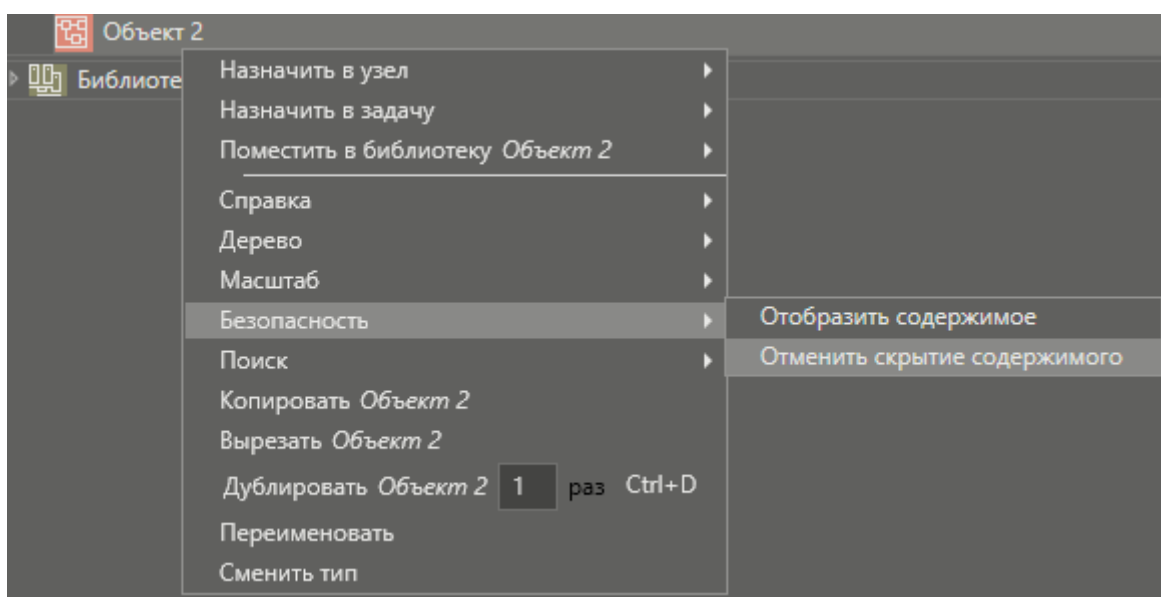
Вид объекта в дереве до скрытия содержимого:



Вид после:



Чтобы отменить данную операцию, следует в контекстном меню выбрать пункт Отменить скрывание содержимого:




Если требуется, чтобы содержимое объекта отобразилось только на время текущей сессии работы с проектом, то необходимо выполнить пункт контекстного меню **Отобразить содержимое**.

При выполнении любого из этих пунктов откроется диалоговое окно, в котором следует ввести ранее установленный пароль.

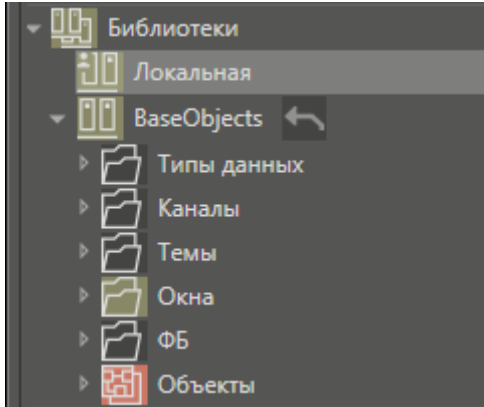
6.4. ДЕРЕВО БИБЛИОТЕК

В дереве библиотек находятся библиотеки проекта и выполняются операции с ними.


Важно! Элементы проекта, которые добавляются в дерево системы или в дерево объектов чаще всего являются экземплярами какого-либо библиотечного элемента: узла, протокола, программы, функции, канала, параметра и др.

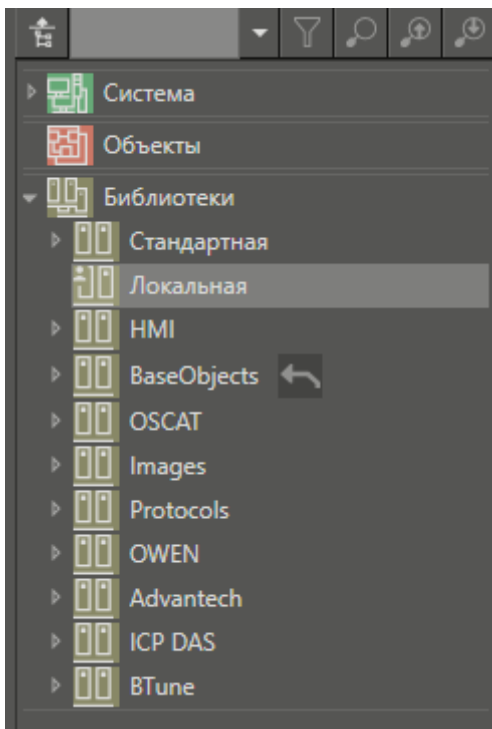
Количество видимых элементов дерева библиотек определяется кнопкой панели инструментов дерева . Если кнопка нажата, то в дереве библиотек видны только те

элементы, которые нельзя добавить в проект при помощи контекстного меню, контекстной панели или легенд редакторов. Этот режим отображения дерева используется по умолчанию. Таким образом, по умолчанию, после старта среды разработки библиотека имеет вид:



Если пользователь ранее создал свою библиотеку или подключил библиотеку, созданную другим пользователем, то помимо библиотеки типовых объектов BaseObjects такие библиотеки также будут видны при нажатом состоянии кнопки.

Если кнопка отжата , то отображаются все элементы подключенных к проекту библиотек.



В редакторе проекта определены два вида библиотек: поставляемые и пользовательские (см. Пользовательская библиотека). Оба вида библиотек имеют одинаковую структуру.

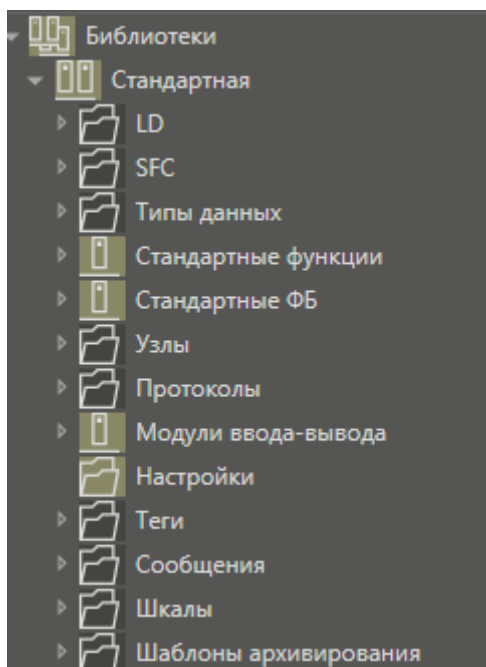
Поставляемые библиотеки содержат ряд элементов, необходимых для разработки проекта (например, для конфигурирования обмена с соответствующим оборудованием) или существенно упрощающих разработку. Поставляемые библиотеки хранятся в папке *<папка установки MasterSCADA 4D>\Configs\MasterPLC\Config\Library*.

Важно! По умолчанию к проекту подключены не все библиотеки. Чтобы подключить библиотеку к проекту, необходимо выполнить команду Подключить библиотеку.

Библиотеки содержат шаблоны элементов. Поставляемая библиотека может содержать встроенные шаблоны, редактирование которых запрещено. По встроенным шаблонам в пользовательской библиотеке могут быть созданы редактируемые шаблоны тех же элементов. Для этого используется перетаскивание шаблона из поставляемой библиотеки в пользовательскую. По шаблонам в узлах или в объектах создаются элементы, обеспечивающие соответствующие функции.

6.4.1. БИБЛИОТЕКА СТАНДАРТНАЯ

Поставляемая библиотека Стандартная содержит элементы, необходимые для разработки проекта. Эта библиотека создается в проекте по умолчанию, и ее нельзя удалить. Она не отображается в упрощенном дереве, т.к. ее содержимое может быть добавлено в проект через контекстное меню, контекстную панель, легенды и панели свойств различных элементов.



Описание категорий библиотеки

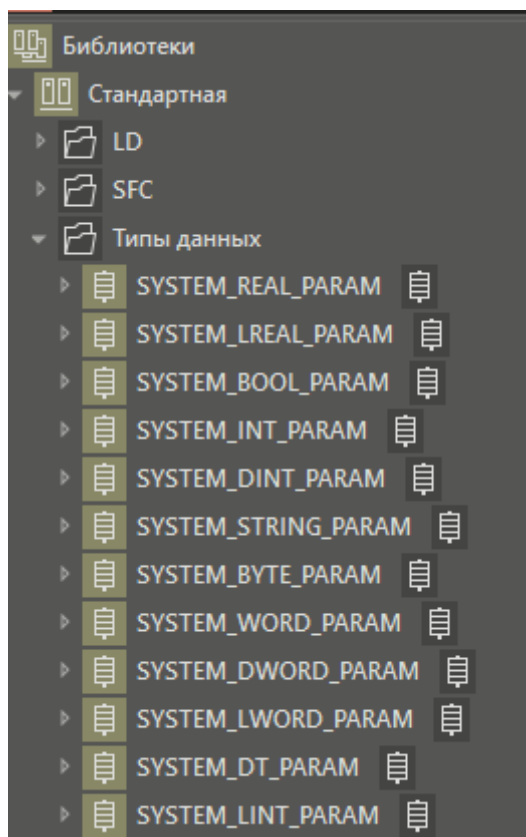
Название	Описание
LD	Содержит стандартные элементы для разработки схем на языке LD, согласно стандарту МЭК 61131-3.
SFC	Содержит стандартные элементы для разработки схем на языке SFC, согласно стандарту МЭК 61131-3.
Типы данных	Содержит стандартные неатомарные типы данных.
Стандартные функции	Содержит стандартные функции для разработки схем на языке FBD и программ ST, согласно стандарту МЭК 61131-3.
Стандартные ФБ	Содержит стандартные функциональные блоки (ФБ) для разработки схем на языке FBD и программ ST, согласно стандарту МЭК 61131-3.
Узлы	Содержит типы наиболее распространенных узлов.
Протоколы	Содержит стандартные протоколы.
Модули ввода-вывода	Содержит типы модулей ввода-вывода для стандартных протоколов, а также элементы, на основании которых можно разработать свой собственный тип модуля ввода-вывода.
Настройки	
Теги	Содержит теги, на основании которых можно сделать свои экземпляры тегов.
Сообщения	Содержит основные типы сообщений, доступные в MasterSCADA 4D. На основании этих типов можно создать свои типы сообщений.
Шкалы	Содержит основные типы шкал. На основании этих типов можно создать свои типы шкал.

Название	Описание
Шаблоны архивирования	Содержит основные типы шаблонов архивирования. На основании этих типов можно создать свои типы шаблонов архивирования.

6.4.1.1. ТИПЫ ДАННЫХ

Группа Типы данных библиотеки Стандартная содержит встроенные производные типы данных (структуры).

Вид группы:



Каждая из этих структур содержит поля:

- Value – значение соответствующего типа (REAL, LREAL, BOOL, INT, DINT, STRING, BYTE, WORD, DWORD, LWORD, DT, LINT);
- SourceTime – метка времени, тип данных DT;
- StatusCode – статус (качество), тип данных UDINT. Параметр StatusCode (статус), начальное значение которого установлено в `OpcUa_BadWaitingForInitialData = 2150760448`, для того чтобы при старте не выполнялась запись в выходные параметры.

В группе Типы данных пользовательской библиотеки создаются производные типы данных пользователя.

Группа Типы данных библиотеки HMI содержит встроенные типы данных для свойств графических элементов (см. Типы данных библиотеки HMI).

6.4.1.1.1. STATUSCODE

Входит в структуру типов данных вида SYSTEM_*_PARAM:

На данном выходе отображается число, соответствующее тому или иному признаку качества стандарта OPC UA. Наиболее часто используются следующие признаки:

Значение	Название	Описание
0	OpcUa_Good	"Хороший" признак качества, значение достоверно.
9830400	OpcUa_GoodLocalOverride	Значение переопределено пользователем.
1084489728	OpcUa_UncertainDataSubNormal	Не определенное значение.
1083506688	OpcUa_UncertainSubNormal	Одно из входных значений недостоверно.
1083375616	OpcUa_UncertainSensorNotAccurate	Значение вне границ датчика.
1083441152	OpcUa_UncertainEngineeringUnitsExceeded	Значение вне диапазона изменения.
1083179008	OpcUa_UncertainLastUsableValue	Обновление значения остановлено.
2147483648	OpcUa_Bad	"Плохой" признак качества, значение не достоверно (как правило, данный признак выводится при отсутствии связи с устройством).
2156462080	OpcUa_BadConfigurationError	Ошибка конфигурации.
2156527616	OpcUa_BadNotConnected	Алгоритм имеет проблемы со связью, например, у входа алгоритма оборвана связь с переменной алгоритма-источника данных.

Значение	Название	Описание
2150694912	OpcUa_BadNoCommunication	Связь с источником данных для какого-либо из входов отсутствует. Последнее опрошенное значение не существует. Или алгоритм работает, но выходное значение пока еще не сформировано (например, идет расчет интеграла). В MasterSCADA 4D OPC переменные имеют данный признак качества до первого опроса OPC сервера. Данный признак также имеют переменные с не заданным значением.
2156593152	OpcUa_BadDeviceFailure	Предписанные алгоритмом операции невозможно выполнить (например, деление на ноль или переполнение порядка)
2156658688	OpcUa_BadSensorFailure	Хотя бы одна входная переменная не удовлетворяет условиям, накладываемым на входные переменные алгоритмом, или хотя бы одна входная переменная поступила с любым из плохих признаков качества.
2156724224	OpcUa_BadOutOfService	Сбой связи либо при существующих настройках алгоритма данный выход не формируется.
2150760448	OpcUa_BadWaitingForInitialData	Значение еще не получено от источника данных. Используется в случае, если для параметра не задано начальное значение.

При получении данных с использованием стандарта OPC DA признаки качества преобразуются в признаки качества стандарта OPC UA согласно таблице ниже.

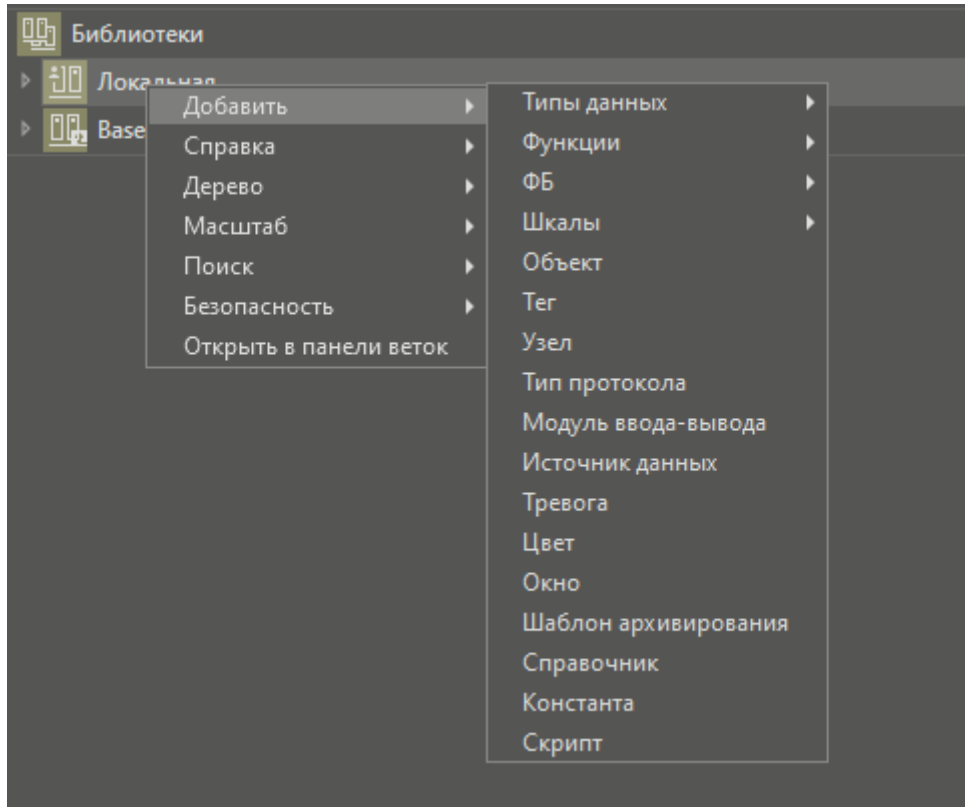
OPC DA	OPC UA
OPC_QUALITY_GOOD	OpcUa_Good
OPC_QUALITY_LOCAL_OVERRIDE	OpcUa_GoodLocalOverride
OPC_QUALITY_UNCERTAIN	OpcUa_Uncertain

OPC DA	OPC UA
OPC_QUALITY_SUB_NORMAL	OpcUa_UncertainSubNormal
OPC_QUALITY_SENSOR_CAL	OpcUa_UncertainSensorNotAccurate
QUALITY_EGU_EXCEEDED	OpcUa_UncertainEngineeringUnitsExceeded
OPC_QUALITY_LAST_USABL	OpcUa_UncertainLastUsableValue
OPC_QUALITY_BAD	OpcUa_Bad
OPC_QUALITY_CONFIG_ERROR	OpcUa_BadConfigurationError
OPC_QUALITY_NOT_CONNECTED	OpcUa_BadNotConnected
OPC_QUALITY_COMM_FAILURE	OpcUa_BadNoCommunication
OPC_QUALITY_DEVICE_FAILURE	OpcUa_BadDeviceFailure
OPC_QUALITY_SENSOR_FAILURE	OpcUa_BadSensorFailure
OPC_QUALITY_LAST_KNOWN	OpcUa_BadOutOfService
OPC_QUALITY_OUT_OF_SERVICE	OpcUa_BadOutOfService
OPC_QUALITY_WAITING_FOR_INITIAL_DATA	OpcUa_BadWaitingForInitialData

В случае если признак качества приходит при опросе с использованием какого-либо протокола, то значение данного выхода может принимать различные значения, близкие по смыслу к установленным признакам качества.

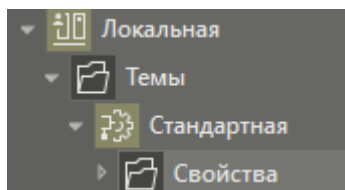
6.4.2. ПОЛЬЗОВАТЕЛЬСКАЯ БИБЛИОТЕКА

Пользовательская библиотека – это библиотека, создаваемая пользователем, и которая может содержать произвольные элементы:



По своей сути пользовательская библиотека - это проект определенной формы, который хранится в папке: *C:\Users\Public\Documents\MasterSCADA4D\Libraries*

При создании проекта в нем автоматически создается пользовательская библиотека Локальная:



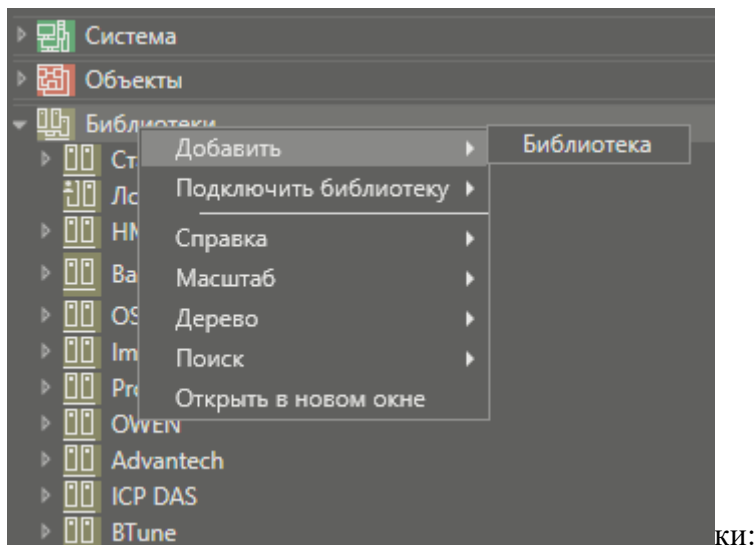
По умолчанию в библиотеке Локальная есть папка Темы, которая содержит тему Стандартная, которую нельзя удалить. В нее можно сразу добавлять параметры темы, которая используется по умолчанию.

Также в пользовательскую библиотеку можно добавлять элемент Скрипт, который служит для автоматической разработки проекта.

6.4.2.1. СОЗДАНИЕ/ПОДКЛЮЧЕНИЕ БИБЛИОТЕКИ

Создание библиотеки

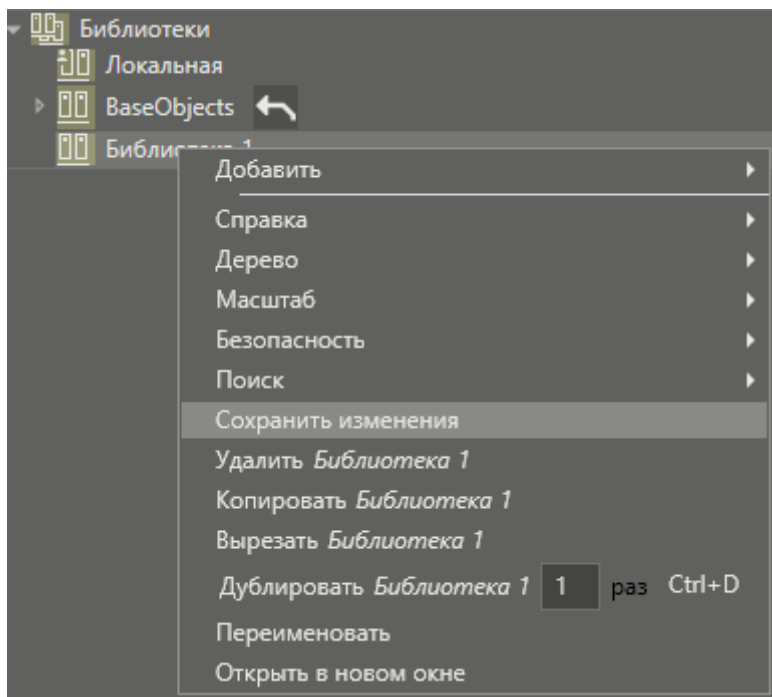
Создать пользовательскую библиотеку можно при помощи пункта Добавить контекстного меню Библиотеки:



Добавить в библиотеку элементы можно любым из доступных способов.

Важно! После выполнения этих действий библиотека будет доступна для использования только в рамках текущего проекта.

Для того чтобы эта библиотека была доступна в других проектах, необходимо выполнить пункт контекстного меню созданной библиотеки Сохранить изменения:



Пользовательская библиотека сохраняется вместе с зависимостями. Например, если в пользовательской библиотеке использованы функциональные блоки других библиотек, то

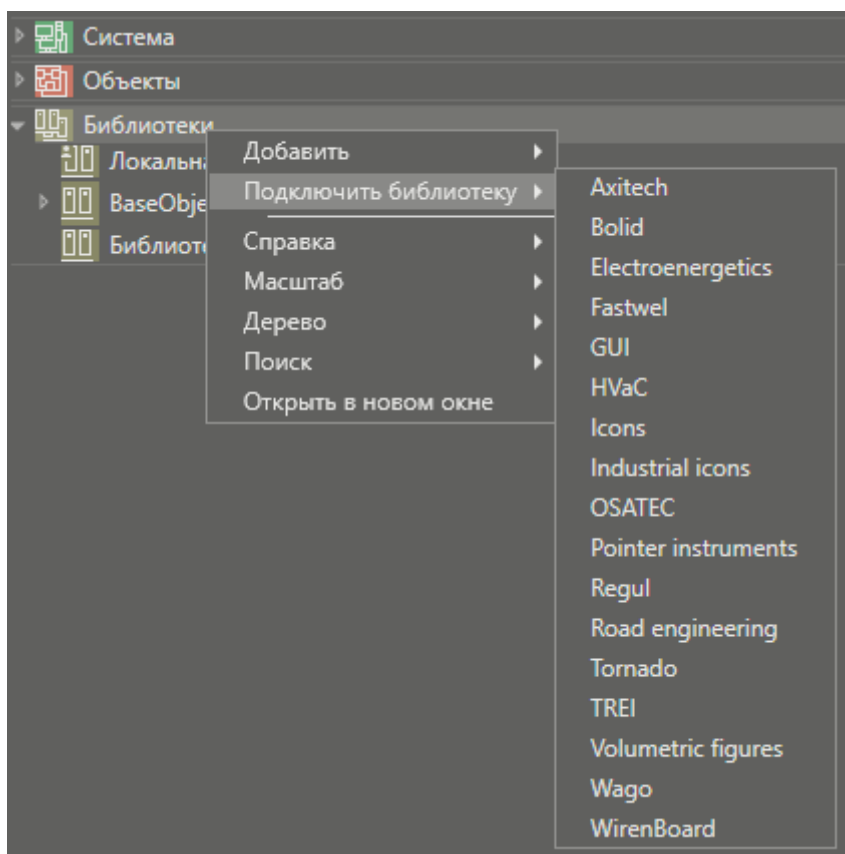
после сохранения такой библиотеки и ее подключения к другому проекту, ранее установленные связи будут работать.

Если в библиотеку помещается некоторый элемент со связями, и для сохранения этих связей в библиотеке не хватает каких-либо элементов, то при выполнении команды Сохранить изменения открывается предупреждение с перечислением элементов, которые необходимо предварительно добавить в библиотеку (тем не менее, кнопкой Сохранить библиотеку появившегося диалогового окна, библиотека может быть сохранена)

Сохраненная библиотека появится в папке— *C:\Users\Public\Documents\MasterSCADA4D<версия>\Libraries*. Для ОС Windows библиотека - это набор файлов и папок.

Подключение библиотеки к среде разработки

Сохраненную на одном компьютере библиотеку можно скопировать на другой компьютер, на котором также установлена среда разработки MasterSCADA 4D, в папку: *C:\Users\Public\Documents\MasterSCADA4D<версия>\Libraries*. Затем в среде разработки выполнить пункт контекстного меню Подключить библиотеку:



В этом меню отображаются все библиотеки, которые могут быть подключены к среде разработки: как библиотеки, разработанные компанией "МПС Софт", но не подключенные по умолчанию, так и скопированные библиотеки пользователей.

Обновление библиотек в проекте

Если в папку хранения библиотек скопирована новая версия уже подключенной библиотеки, то изменения вступят в силу после Обновления библиотек:

Механизм обновления всех библиотек запускается в следующих случаях:

- Автоматически при открытии проекта в среде разработки;
- При выполнении пункта меню Управление проектами.

Индивидуально библиотеку можно обновить, выполнив соответствующий пункт ее контекстного меню.

После обновления библиотек экземпляры обновятся автоматически, в соответствии со своим типом отношения с библиотечным элементом.

6.4.2.2. РАБОТА С ТИПОМ ОБЪЕКТА

Если объект был создан в библиотеке, то в дереве объектов он может быть представлен в виде экземпляра, либо в виде наследника.

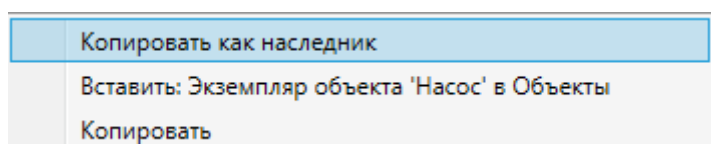
Экземпляр объекта - это элемент, который появляется путем вставки объекта из библиотеки. Имеет отношения с типом Ссылается. Экземпляр содержит в себе только элементы, необходимые для связи с другими элементами проекта. Это позволяет значительно уменьшить размер создаваемого проекта и увеличить скорость работы в режиме разработки. Разработчик не может изменить внутреннюю структуру отдельного экземпляра после его добавления в проект. Экземпляр объекта содержит ссылки на окна и программы.

Наследник объекта - это элемент, который появляется путем вставки объекта из библиотеки. Имеет отношения с типом Унаследован от, является копией библиотечного объекта, содержит всю внутреннюю структуру родительского объекта, любой элемент наследника может быть изменен.

Добавление в проект

Перетаскивание правой кнопкой мыши

При перетаскивании типа объекта правой кнопкой мыши появляется контекстное меню, в котором можно выбрать как именно нужно добавить объект в проект:



Если был выбран пункт Копировать, то добавится несвязанный с типом объект, точная копия типа. Изменение типа в библиотеке не приведет к изменению скопированного объекта.

Перетаскивание левой кнопкой мыши

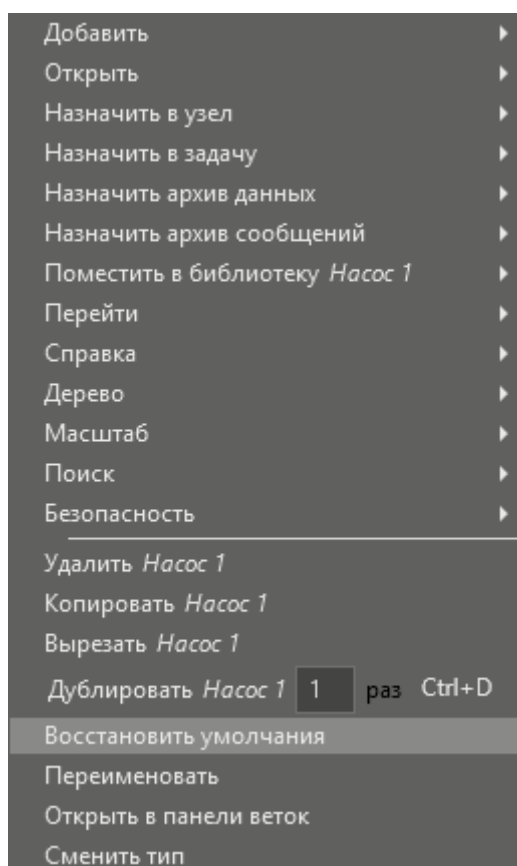
Если перетаскивать объект из библиотеки левой кнопкой мыши, то по умолчанию добавится наследник

Редактирование типа

Если тип был отредактирован в библиотеке, то работа экземпляра в проекте будет полностью соответствовать типу.

Наследники автоматически изменятся в соответствии с изменениям в типе, но за исключением тех элементов и свойств наследника, которые были изменены после добавления в проект.

Если необходимо отменить изменения, сделанные в наследнике, то в контекстном меню наследника необходимо выполнить пункт Восстановить умолчания:



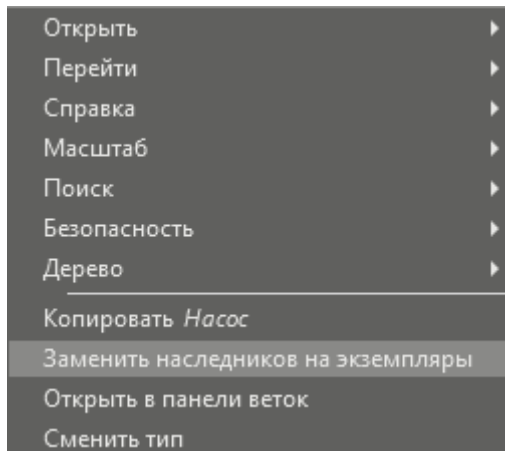
В этом случае наследник опять станет точной копией типа.

Использование экземпляров объектов в новом типе

В типе библиотечного объекта могут быть добавлены экземпляры других объектов. При создании в проекте экземпляра такого объекта в него вставляются также вложенные экземпляры. Они имеют ссылки на программы и окна. Параметры вложенного экземпляра могут иметь внешние связи. Его окна можно вытаскивать в другие окна, а также открывать в результате выполнения действий, назначенных на события

Автоматическая замена наследника на экземпляр

Если в проект был добавлен наследник, то для того чтобы заменить его на экземпляр, который сохранил бы ранее установленные связи, необходимо в библиотеке в контекстном меню типа объекта выполнить пункт **Заменить наследников на экземпляры**:

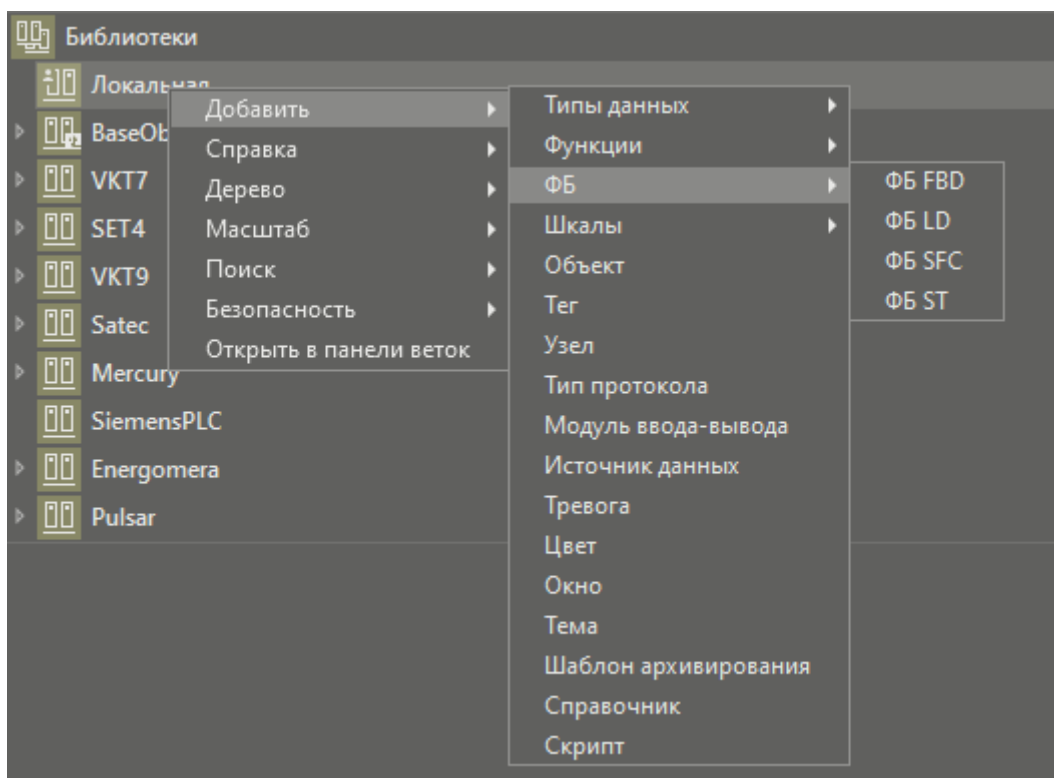


6.4.2.3. СОЗДАНИЕ ТИПА ФУНКЦИОНАЛЬНОГО БЛОКА

Создание ФБ штатными средствами среды разработки

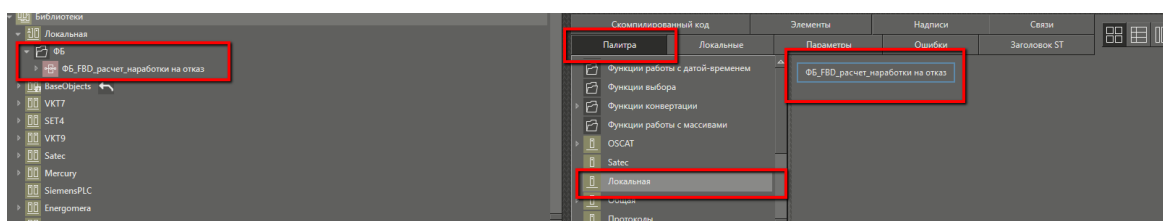
В MasterSCADA 4D можно создать новый функциональный блок, используя стандартные средства среды разработки.

Для этого в пользовательскую библиотеку нужно добавить функциональный блок:



В зависимости от того, какой пункт контекстного меню будет выбран откроется соответствующий программный редактор, где можно создать набор входных и выходных параметров, а также используя стандартные функциональные блоки создать новый алгоритм.

Созданный функциональный блок будет отображаться как в дереве библиотек, так и в легенде того или иного программного редактора:



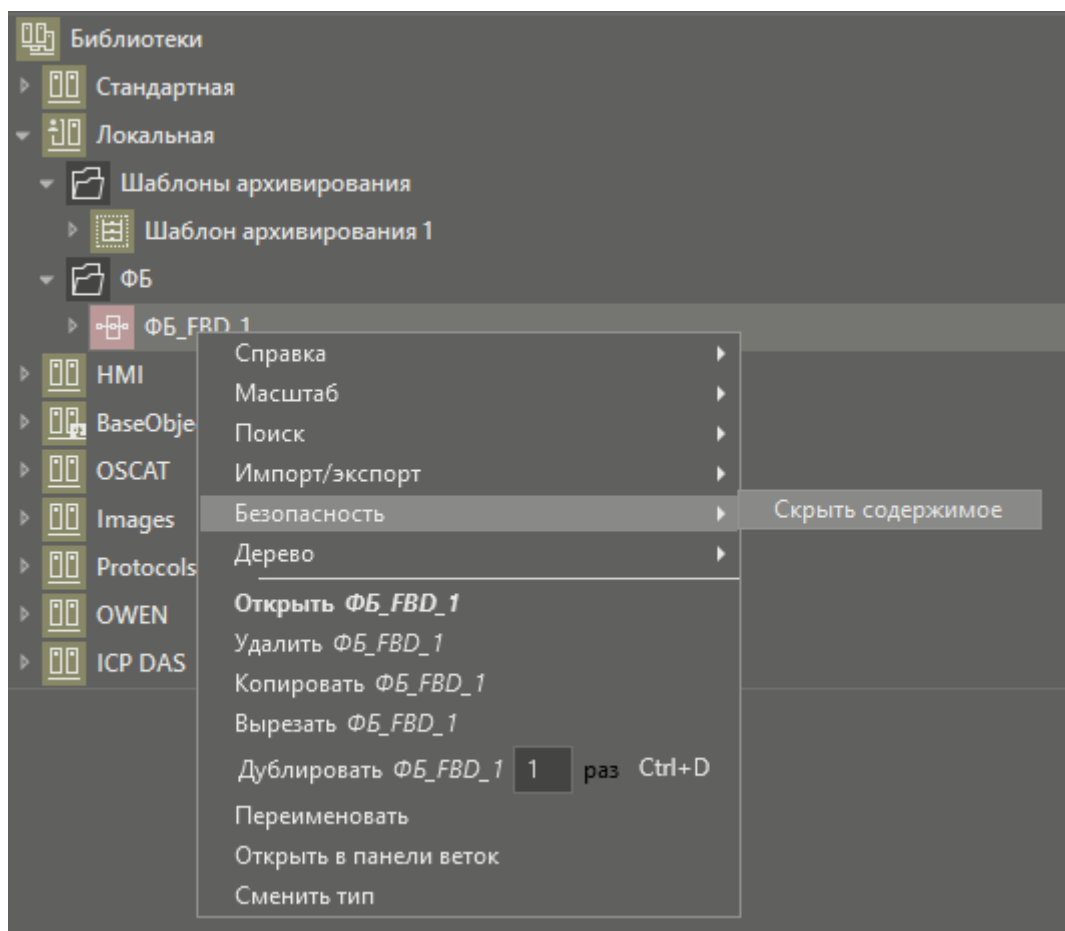
Далее в проекте можно использовать созданный функциональный блок также как и стандартные функциональные блоки:

- независимо в дереве системы или дереве объектов
- внутри какой-либо программы

В обоих случаях в проекте будет создан экземпляр функционального блока.

Скрыть содержимое функционального блока

Библиотечный функциональный блок имеет пункт контекстного меню Безопасность.Скрыть содержимое:



После выполнения этого пункта меню откроется окно задания пароля, такое же как и для аналогичного пункта меню объекта.

После того как пароль будет задан открыть его содержимое, т.е. открыть редактор с программой этого ФБ будет нельзя.

При использовании экземпляра такого ФБ разработчик проекта увидит только входы и выходы ФБ, но не его логику работы.

Сохранение данных горячего рестарта

Если у экземпляра функционального блока расположенного независимо в дереве объектов или в дереве системы, или в какой-либо программе стоит Сохранять=Да, то будут сохраняться все параметры экземпляра (включая локальные), кроме тех, у которых стоит Сохранять=Нет.

Значение Наследуется означает, что будут сохраняться только те его параметры, у которых стоит Сохранить=Да

Создание ФБ с использованием внешних приложений

К MasterSCADA 4D есть возможность подключить функциональные блоки созданные на языке на C++. *Подробное описание смотрите в файле c:\Program Files*

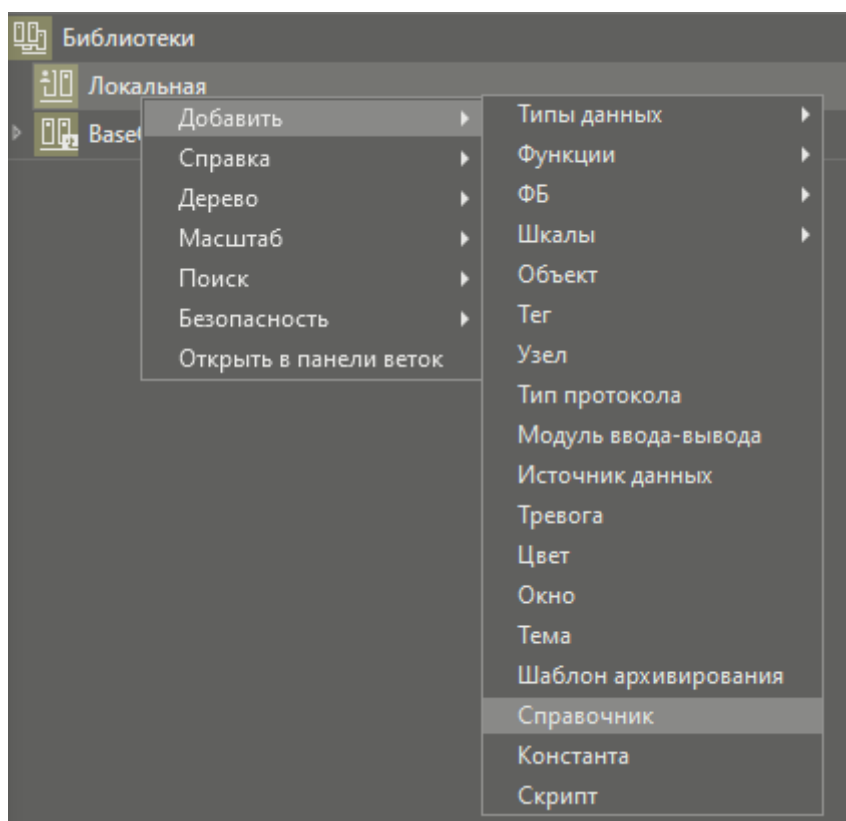
(x86)\MSSoft\MasterSCADA 4D Beta\API\Doc\Разработка драйверов и ФБ в MasterSCADA 4D.docx

Важно! При использовании функциональных блоков, созданных во внешних приложениях, необходимо уделить особое внимание тестированию взаимодействия его с различными элементами проекта! Также необходимо убедиться, что выбранная исполнительная система поддерживает использование таких функциональных блоков.

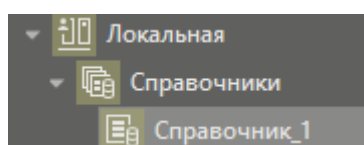
6.4.2.4. СОЗДАНИЕ ТИПА СПРАВОЧНИКА

Элемент служит для формирования списка полей, значения которых необходимо получать из базы данных для работы графического элемента Справочник.

Для добавления элемента в пользовательскую библиотеку нужно выполнить пункт контекстного меню библиотеки Добавить.Справочник:



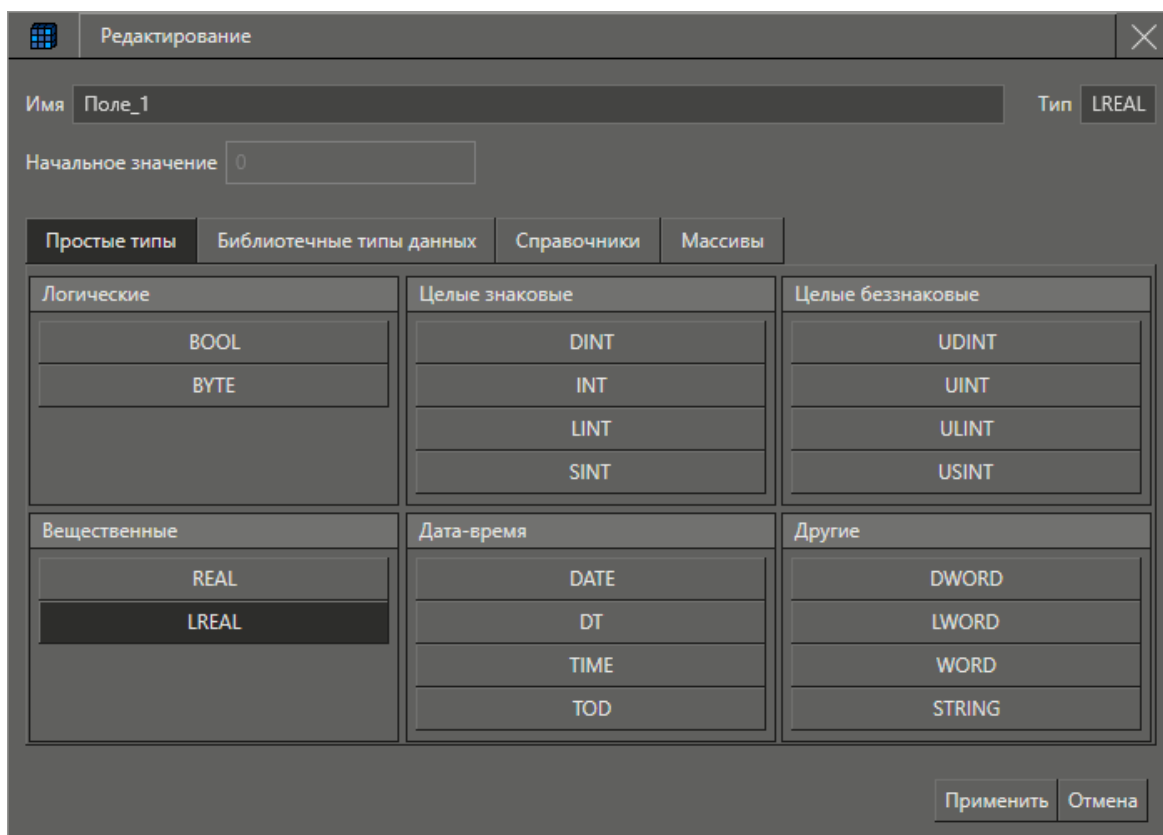
Появится группа Справочники, в которую будет добавлен элемент Справочник_1:



Имя добавленного элемента можно изменить в его панели свойств.

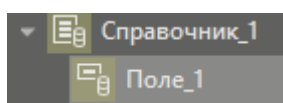
Важно! Имя Справочника должно совпадать с именем таблицы в базе данных.

Далее в Справочник нужно добавить Поля. После выполнения пункта Добавить.Поле контекстного меню Справочника откроется диалоговое окно, в котором нужно выбрать тип данных для этого поля:



В данном окне можно выбрать как простые типы данных, так и любые библиотечные. Также в качестве типа данных поля можно выбрать и другой справочник, который был ранее добавлен в библиотеку.

После выбора типа данных появится новый элемент в справочнике:



Набор полей справочника и их свойства (имя, тип данных и др.) должны совпадать с набором полей в реальной таблице базы данных, с которой необходимо обмениваться данными.

Вид панели свойств Поля справочника:

Кроме стандартных свойств категория Общие содержит дополнительные свойства:

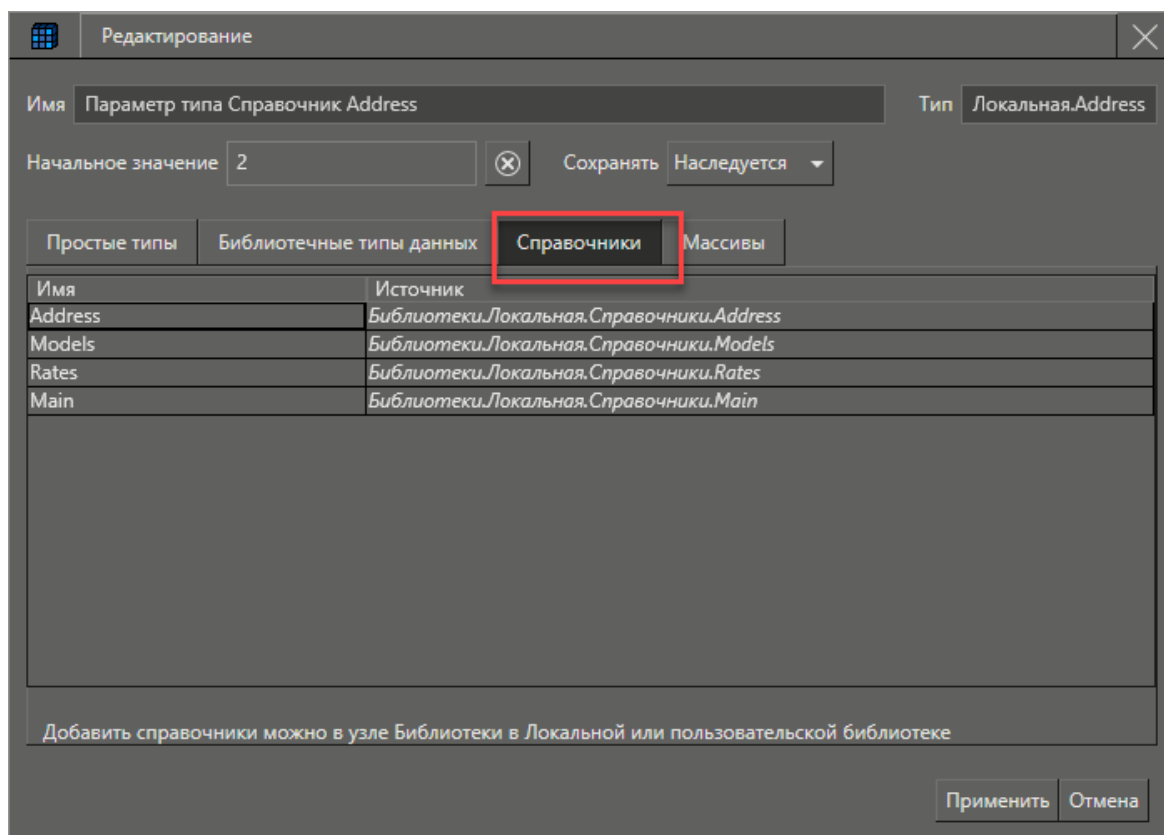
Название	Тип	Назначение
Ключевое поле	ВОО L	Определяет является ли данное поле ключом для справочника. Если флаг установлен, то через это поле идут ссылки на данный справочник из других справочников, а также из обычных параметров. Если ни у одного поля не установлен флаг Ключевое поле, то в таблице необходимо создать дополнительное числовое поле Id, которое будет являться ключом.
Отображаемое имя	ВОО L	Определяет будет ли значение данного поля использовано в отображении значений полей других справочников, ссылающихся на данный.

Описание других свойств находятся в соответствующих разделах.

Далее необходимо перетащить Справочник на элемент дерева системы БД справочников.

Параметр типа Справочник

Параметр объекта или программы может иметь тип Справочник. После добавления типа Справочник в библиотеку в окне настройки параметра появится дополнительная вкладка Справочники:



Во вкладке можно выбрать нужный справочник, при этом узел, где исполняется данный объект, должен иметь связь с базой данных, соответствующей этому справочнику. В этом случае параметр будет ссылаться на некоторую запись из данного справочника. Работа с таким параметром аналогична параметру структурного типа с некоторыми особенностями:

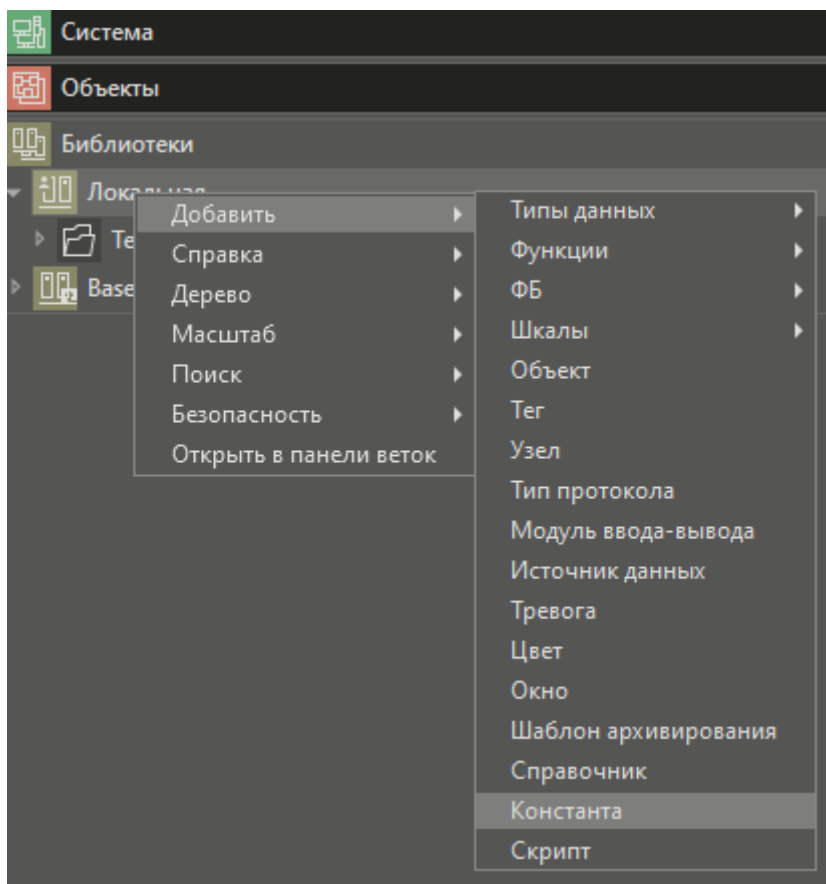
- Нельзя изменять отдельные поля справочника
- Отдельные поля можно перетаскивать в окна, передавать в другие параметры. Будет браться значение поля выбранной записи
- Если параметр экземпляра объекта имеет тип справочник, то в режиме разработки можно из комбобокса выбрать нужную запись (в случае, если доступно подключение к БД, где хранятся справочники этого типа). В этом случае в режиме исполнения в этот параметр записывается ссылка на выбранный экземпляр
- Период обновления значений полей выбранной записи задается в настройке БД справочников Период обновления параметров из справочника.

6.4.2.5. РАБОТА С КОНСТАНТАМИ

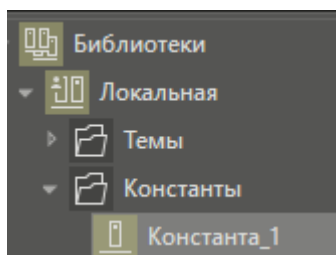
В пользовательскую библиотеку можно добавить Константу. Которую в дальнейшей можно использовать в программах ST и FBD.

Создание Константы

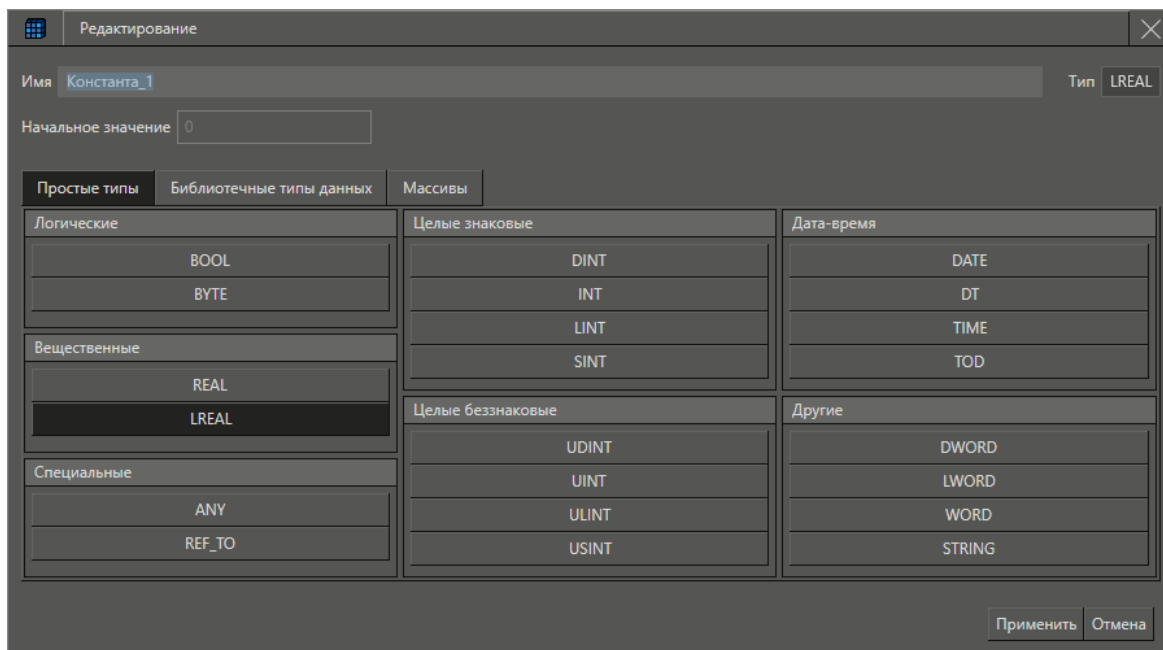
Добавить константу можно при помощи контекстного меню библиотеки:



Появится группа Константы:



Откроется окно настройки Константы:



Порядок работы с окном такой же как и с окном настройки параметра. Нужно указать имя, тип константы и ее значение в поле Начальное значение. Константу можно настраивать и в ее панели свойств. Рекомендуется давать константам уникальные имена, даже если они находятся в разных библиотеках, если имена будут совпадать, то при компиляции проекта, где указаны только имена констант будет использоваться та константа, которая находится в дереве библиотек выше.

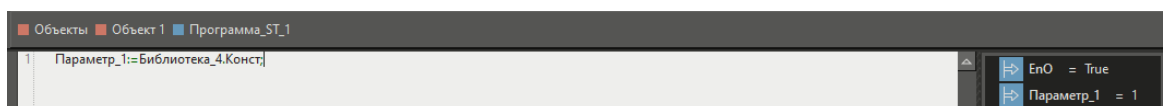
При компиляции проекта вместо констант подставляются их начальные значения.

Использование Констант

В программах ST

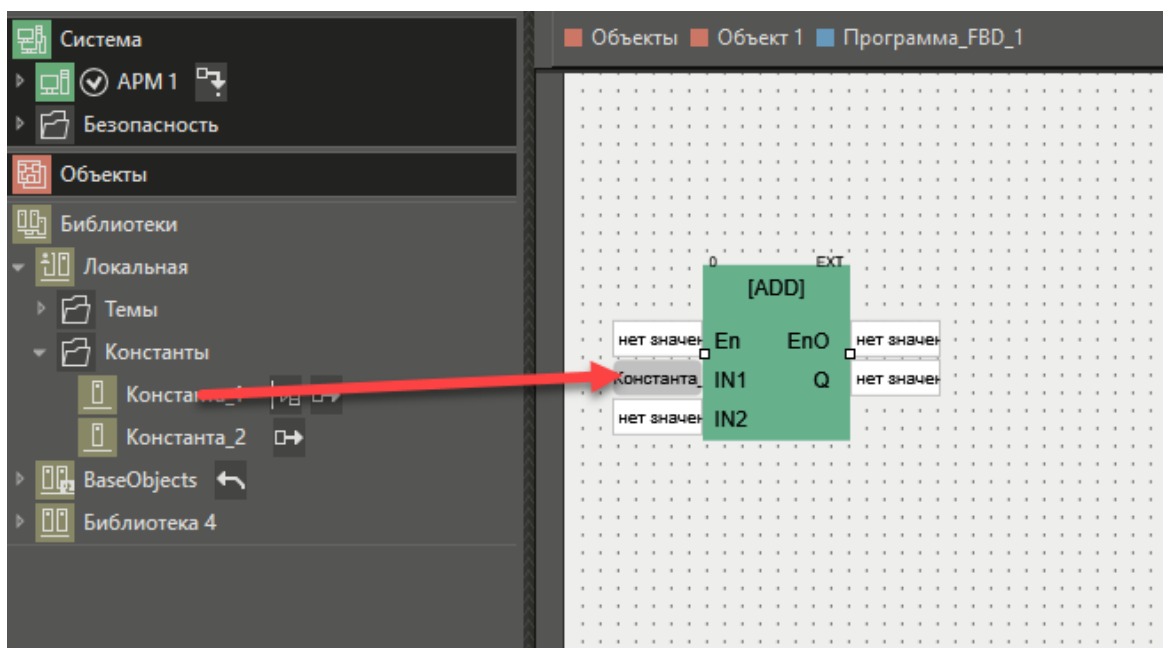
Для использования в программах ST нужно в коде указать имя константы или перетащить ее в код из библиотеки. Можно использовать как простое имя константы, например, Константа_1, так и составное с именем библиотеки Локальная.Константа_1.

Если перетащить константу в текст программы, то она добавится в виде Имя_библиотеки.Имя_константы:



В программах FBD

Параметры программ FBD можно связывать с константами методом drag&drop:

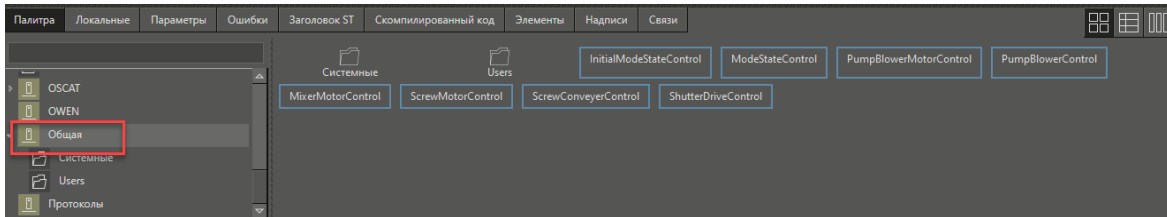


6.4.3. БИБЛИОТЕКА BASEOBJECTS

Библиотека BaseObjects подключена по умолчанию в среде разработки. Содержит следующие типы элементов:

- Типы данных - содержит типы данных, которые используются в объектах данной библиотеки; могут использоваться самостоятельно;
- Каналы - содержит predetermined каналы, которые используются в дереве системы, в различных протоколах, например, Modbus;
- Темы - содержит настройки, которые применяются к окнам данной библиотеки при смене темы в проекте либо в отдельном окне;
- Окна - содержит окна, которые использовались в объектах данной библиотеки; могут использоваться самостоятельно;
- ФБ - содержит дочернюю группу Системные, а также функциональные блоки, которые использовались для объектов данной библиотеки; могут использоваться самостоятельно;
- Объекты - группа содержит комплексные элементы, описывающие работу того или иного технологического объекта, и которые могут быть использованы как составные части проектов. Каждый объект содержит внутри себя экземпляры окон и программ.

В палитре программ библиотека имеет название Общая:



6.4.3.1. BASEOBJECTS.ФБ.СИСТЕМНЫЕ

Данная библиотека содержит функциональные блоки, которые не входят в состав стандарта МЭК 61131-3. Эти функциональные блоки предназначены как для работы с внешними приложениями, так и для решения различных задач внутри проекта.

SetDateAndTime

SEND_EMAIL

READ_ARCHIVE_DATA

PlaySound

SEND_SMS

ReportFB

GET_CURRENT_USERNAME

SysProcessCreate

SysProcessCreateWithResult

SysProcessCreateAsync

SysProcessCreateWithResultAsync

COMPort

COMPortByte

COMPortByteSync

WriteArchiveData

FireBaseEvent

FileToString

StringToFile

XMLString

PING

GetSystemInfo

RedundancyControl

ExportEventsArchive

FileIntegrityControl

GetOpcUaServerInfo

GetDataArchiveStatistic

GetEventsArchiveStatistic

6.4.3.1.1. SETDATEANDTIME

Функциональный блок SetDateAndTime устанавливает системную дату и время устройства, на котором запущена исполнительная система.

Вид в редакторе FBD	Вид в дереве

Описание входов и выходов:

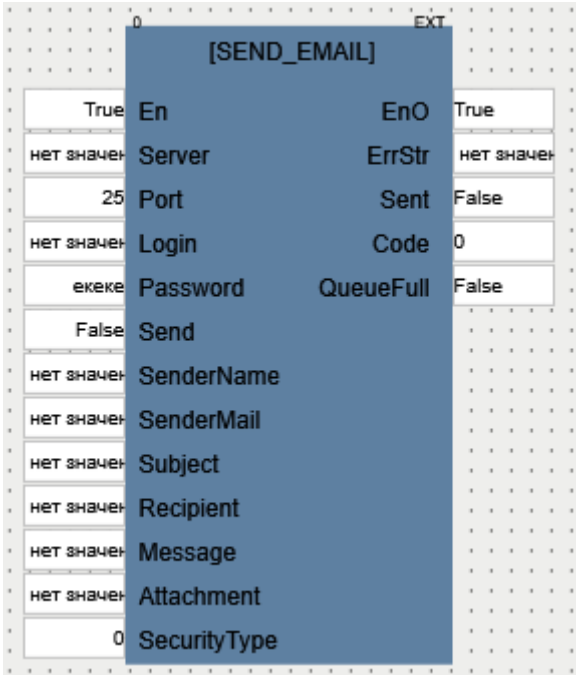
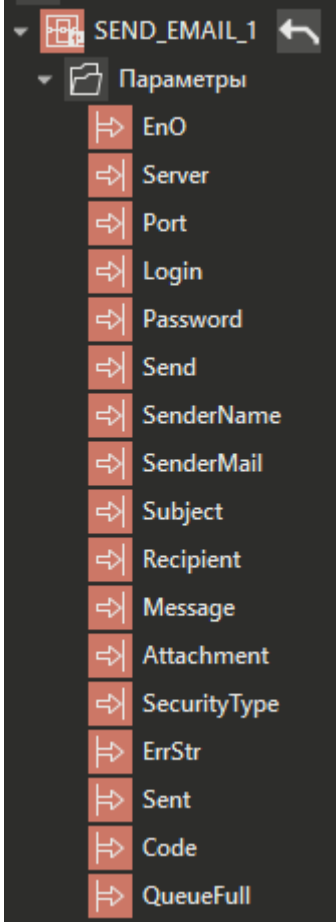
Название	Тип	Назначение
Execute	BOOL	Вход ФБ. По переднему фронту изменения значения на этом входе устанавливаются системная дата и время.
DateAndTime	DT	Вход ФБ. Поступают устанавливаемые системная дата и время.

IsUTC	BOOL	Вход ФБ. Указывается формат времени на входе DateAndTime (TRUE – UTC, False – локальное время).
ErrorCode	DINT	Выход ФБ. Выдается информация об ошибке.

Важно! Если среда разработки запускает демоверсию среды исполнения, входящую в ее состав, автоматически, то для корректной работы данного функционального блока среда разработки должна быть запущена под правами администратора.

6.4.3.1.2. SEND_EMAIL

Функциональный блок SEND_EMAIL служит для отправки электронных писем при наличии доступа в Интернет. Этот функциональный блок может использоваться как внутри программы, так и непосредственно в дереве MasterSCADA.

Вид в редакторе FBD	Вид в дереве
	

Входы и выходы ФБ:

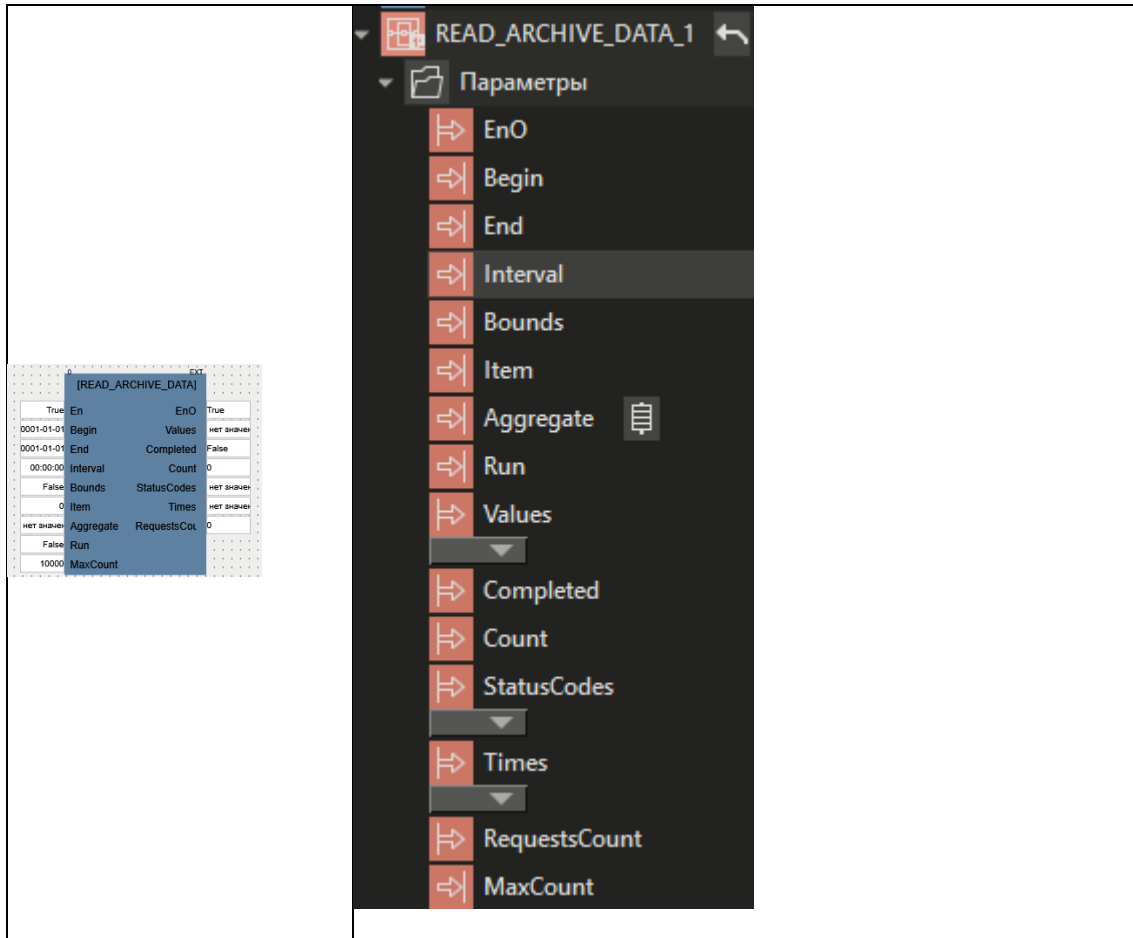
Название	Тип	Назначение
Входы		
Server	STRING	Сервер SMTP
Port	DINT	Порт сервера (в текущей версии ФБ работает только через порт 25 без шифрования).
Login	STRING	Логин для авторизации на сервере.
Password	STRING	Пароль для авторизации на сервере. После ввода значения, по умолчанию в среде разработки оно будет замаскировано звездочками.
Send	BOOL	Отправка сообщения по переднему фронту изменения значения на этом входе. По каждому импульсу, поступившему на вход Send, сообщение добавляется в очередь на отправку. Отправка проходит асинхронно, по каждому отправленному сообщению выдается импульс на выход Sent (одновременно выдается ошибка по данному сообщению, если она возникла).
SenderName	STRING	Имя отправителя.
SenderMail	STRING	Адрес почты отправителя.
Subject	STRING	Тема письма.
Recipient	STRING	Адрес почты получателя. Если необходимо отправить письмо нескольким получателям, то необходимо писать их адреса через запятую и пробел.
Message	STRING	Текст сообщения.
Attachment	STRING	Полный путь к файлу, который требуется добавить к письму. Если к одному письму необходимо добавить несколько вложений, то в этом случае путь к ним

Название	Тип	Назначение
		указывается через точку с запятой (;). Например, <i>C:\Users\Operator\Desktop\1.txt;C:\Users\Operator\Desktop\2.txt</i>
SecurityType	UINT	Задается тип шифрования SMTP: <ul style="list-style-type: none"> • 0 - NO_SECURITY • 1 - USE_TLS • 2 - USE_SSL
Выходы		
ErrStr	STRING	Текст ошибки
Sent	BOOL	Принимает значение TRUE после отправки сообщения на 1 такт.
Code	DINT	Код ошибки. Значение 0 обозначает отсутствие ошибки.
QueueFull	BOOL	Показывает, когда очередь на отправку сообщений переполнена.

6.4.3.1.3. READ_ARCHIVE_DATA

Функциональный блок READ_ARCHIVE_DATA предназначен для выборки данных из архива по заданному параметру.

Вид в редакторе FBD	Вид в дереве
---------------------	--------------



Входы и выходы ФБ:

Название	Тип	Назначение
Входы		
Begin	DT	Время начала выборки.
End	DT	Время конца выборки.
Interval	TIME	Интервал, используемый для агрегатных функций. Если установлен 0, то идет обработка за весь интервал выборки.
Bounds	BOOL	Включение граничных значений при установленном параметре TRUE.

Название	Тип	Назначение
Item	REF_TO	Вход ФБ для установки связи с параметром, архив которого необходимо получить. При установке связи на этот вход будет передаваться не значение параметра, а его ID. Если ФБ находится внутри программы ST, то в программе можно указать полное имя архивируемого элемента. См раздел Специальные типы данных.
Aggregate	UINT	Тип выборки (перечисление из возможных агрегатов: мгновенное значение, интеграл и т.п.).
Run	BOOL	По переднему фронту изменения значения на этом входе новый запрос ставится в очередь.
MaxCount	DINT	Возвращает не более заданного числа архивных записей. Если записей больше в заданном диапазоне, то в выборку попадут все последние значения., а более ранние значения будут проигнорированы. По умолчанию установлено значение 10000.
Выходы		
Values	ARRAY [*] OF ANY	Массив архивных значений параметра, связанного с ФБ.
Completed	BOOL	Формируется результат выполнения запроса. После отработки запроса этот выход на один такт принимает значение TRUE. (При этом одновременно заполняются остальные выходы с данными выполненного запроса).
Count	UINT	Количество значений в выборке.
StatusCodes	ARRAY [*] OF DINT	Массив признаков качества.
Times	ARRAY [*] OF	Массив меток времени.

Название	Тип	Назначение
	DATE_AND_TIME	
RequestsCount	INT	Количество запросов в очереди на выполнение.

Рекомендации по работе с ФБ

Если входу Run присвоить значение TRUE в то время когда выполняется один запрос, то новый запрос ставится в очередь (увеличивается значение RequestsCount). Как только запрос выполнен, то при очередном вызове ФБ на выход Completed выдается TRUE, и значение RequestsCount уменьшается. То есть, сколько запросов было добавлено, столько раз на Completed произойдет переход в состояние TRUE. Таким образом, код обработки результата имеет вид:

```

READ_ARCHIVE_DATA_1();

if (READ_ARCHIVE_DATA_1.Completed) then

MyRes := READ_ARCHIVE_DATA_1.Values

end_if

```

6.4.3.1.4. PLAYSOUND

Функциональный блок PlaySound воспроизводит аудиофайл формата WAV, MP3 или OGG. Используется для привлечения внимания операторов при возникновении аварийных ситуаций. Этот функциональный блок может использоваться в ST и FBD программах.

Вид в редакторе FBD	Вид в редакторе ST
	<pre> PlaySound_1(Play:=Включить_звук, Rep:=TRUE, Sound:="mp3.MP3");//(где: Включить_звук - это логический параметр программы ST, связанный с параметром, который отвечает за запуск воспроизведения). </pre>

Важно! Программа, в которой находится ФБ, должна выполняться в задаче экрана. Часто программы, которые должны работать в задаче экрана, размещают в отдельном объекте. Для этого в свойствах объекта необходимо выбрать место исполнения в задаче экрана. При этом дополнительно переопределять место исполнения дочерних элементов этого объекта не требуется.

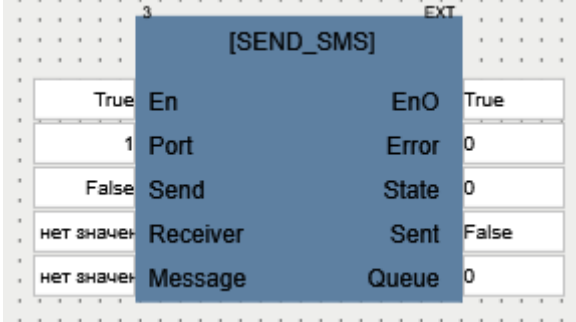
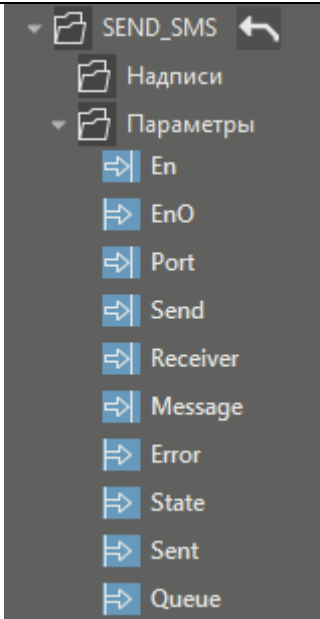
Назначение входов функционального блока:

Название	Тип	Назначение
Play	BOOL	Вход ФБ. По переднему фронту изменения значения на этом входе запускается воспроизведение файла
Rep	BOOL	Вход ФБ. Определяет, требуется ли повторное проигрывание звукового файла. При значении входа False – однократное воспроизведение, а при значении TRUE воспроизведение будет циклически повторяться до тех пор, пока вход Play находится в состоянии TRUE.
Sound	STRING	Вход ФБ. Имя аудиофайла с расширением (например, snd.OGG)/ Важно! Название файла указанное на данном входе должно полностью совпадать с реальным названием файла, учитывая регистр. Например, если файл имеет название Sound.OGG, то и на вход нужно подать значение Sound.OGG, если файл имеет название sound.ogg, то и на вход - sound.ogg
Stop	BOOL	Вход ФБ. Останавливает воспроизведение/

Важно! Перед использованием аудиофайла его необходимо добавить в Элемент Медиа.

6.4.3.1.5. SEND_SMS

Функциональный блок SEND_SMS служит для отправки SMS-сообщений через встроенный GSM-модем или GSM-модем, подключенный к COM-порту устройства. Этот функциональный блок может использоваться как внутри программы, так и непосредственно в дереве MasterSCADA.

Вид в редакторе FBD	Вид в дереве
 <p>The screenshot shows a blue block labeled [SEND_SMS] in a grid-based editor. The block has several inputs and outputs: <ul style="list-style-type: none"> En: Boolean input, currently set to True. EnO: Boolean output, currently set to True. Port: Integer input, currently set to 1. Error: Integer output, currently set to 0. Send: Boolean input, currently set to False. State: Integer output, currently set to 0. Receiver: String input, currently set to "нет значен". Sent: Boolean output, currently set to False. Message: String input, currently set to "нет значен". Queue: Integer output, currently set to 0. </p>	 <p>The screenshot shows a project tree structure for the SEND_SMS block. The root is SEND_SMS, which contains: <ul style="list-style-type: none"> Надписи (Labels) Параметры (Parameters), which includes: <ul style="list-style-type: none"> En EnO Port Send Receiver Message Error State Sent Queue </p>

Отправка сообщения производится целиком – без ограничений, накладываемых размером одного SMS-сообщения. Поддерживается отправка фрагментированных SMS сообщений.

Входы и выходы ФБ:

Название	Тип	Назначение
Входы		
Port	DINT	В текущей версии на этот вход необходимо подать тот же номер, который указан в настройке Параметры запуска RT.
Send	BOOL	Отправка SMS-сообщения производится по переднему фронту изменения значения на этом входе. По каждому импульсу, поступившему на вход Send, сообщение добавляется в очередь на отправку. Отправка проходит асинхронно, по каждому отправленному сообщению выдается импульс на выход Sent (одновременно выдается ошибка по данному сообщению, если она возникла).
Receiver	STRING	Задается телефон получателя SMS-сообщения в формате 7*****. Если необходимо отправить сообщения нескольким абонентам, то необходимо номера телефонов писать через запятую и пробел.

Название	Тип	Назначение
Message	STRING	Определение текста сообщения.
Выходы		
State	DINT	Код состояния. Отображается после срабатывания переднего фронта параметра ФБ Send.
Sent	BOOL	Принимает значение TRUE после отправки сообщения.
Error	DINT	Выдаются коды ошибок.
Queue	DINT	Показывает количество сообщений в очереди на отправку.

Ниже перечислены наиболее часто встречающиеся коды ошибок и коды состояний.

Коды ошибок:

Номер	Описание
5	Ошибка разбора PDU
6	Истекло время ожидания ответа от модема
10	Порт занят выполнением другого запроса
11	Ошибка кодировки сообщения
12	Номер получателя пуст
302	The operation to be done by the AT command is not allowed.
310	There is no SIM card
331	No network service is available.

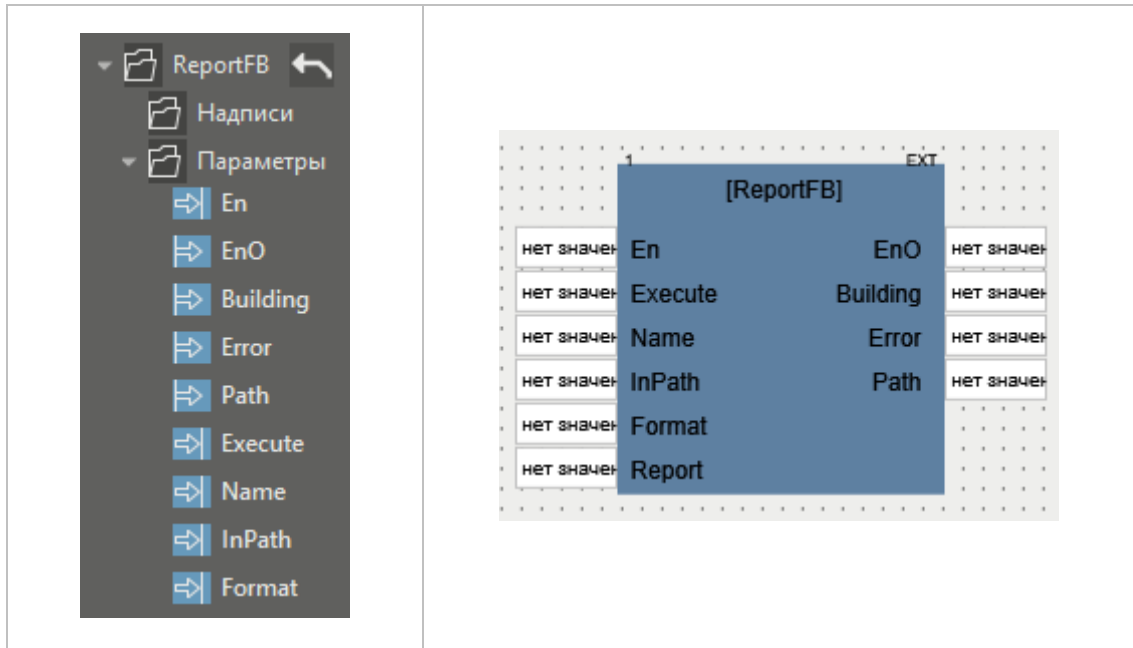
Коды состояний:

Номер	Описание
0	Ожидание инициализации
1	Ожидание команды
2	Начало чтения
16	Инициализация отправки с модема
17	Начало отправки SMS
18	Отправка SMS
34	Удаление прочитанного SMS
33	Чтения 1 SMS

6.4.3.1.6. REPORTFB

Функциональный блок ReportFB служит для запуска процедуры формирования отчета. Может использоваться для решения вспомогательных задач: открыть в окне клиента сформированный отчет, распечатать и др. Этот функциональный блок может использоваться как самостоятельно, вне программ, так и в составе любой программы.

Вид в дереве	Вид в редакторе FBD
--------------	---------------------



Функциональный блок имеет входы и выходы. На входы подаются команды для управления формированием отчета, а на выходы выдается информация о результатах.

Название	Тип	Назначение
Входы		
Execute	BOOL	По переднему фронту изменения значения на этом входе начинает формироваться отчет.
Name	STRING	<p>Указывается имя отчета в проекте. Один функциональный блок может использоваться для формирования разных отчетов, шаблоны которых заданы в дереве системы в группе узла Отчеты, либо в группе Ресурсы у объектов. Для наглядности, для каждого шаблона добавляется свой функциональный блок. В этом случае данный вход является константой, которая задается в поле свойства входа Начальное значение.</p> <p>Имя отчета содержит в себе имя объекта и его родительских объектов, в которых находится отчет и название самого отчета. Можно скопировать свойство отчета Полное имя, вставить его в свойство Начальное значение входа функционального блока и удалить оттуда несущественные элементы. Например, если свойство Полное имя имеет значение: Объекты.Формирование отчета.Ресурсы.Отчеты.Отчет 1, то нужно оставить только:</p>

Название	Тип	Назначение
		<p>Формирование отчета. Отчет 1 либо Объекты. Формирование отчета. Отчет 1</p> <p>Если отчет создан в дереве системы, то в свойстве указывается только название отчета. Имя узла не добавляется.</p>
InPath	STRING	<p>Указывается место на диске, где сформируется отчет. Может быть указана произвольная папка.</p> <p>Важно! Вы можете указать произвольную папку на диске, в случае если собираетесь для просмотра отчетов использовать внешние по отношению к продуктам компании МПС Софт средства.</p> <p>Следует учитывать, что для отображения отчета в окне клиента исполнительной системы (и для выполнения других действий, которые будут выполняться из графического клиента с файлом отчета), файл отчета должен быть доступен web-серверу, а значит должен находиться внутри папки <code>\htdocs</code> проекта, загруженного в среду исполнения.</p> <p>Если этот параметр не задан, то по умолчанию используется папка <code><папка узла>\htdocs\</code> (например, если узел запускается локально из среды разработки, то используется папка <code><имя пользователя>\AppData\Roaming\MPSSot\MasterSCADA4D<версия продукта>\Debug_<имя проекта>\<имя узла>\PLC\htdocs</code>, а если узел запускается из папки запуска проекта по умолчанию среды исполнения для Windows, то используется папка <code>c:\Users\<имя пользователя>\AppData\Roaming\MPSSot\MasterSCADA4D RT<версия продукта>\Server\htdocs</code>);</p>
Format	STRING	<p>Задается формат файла отчета (pdf, html и xlsx). По умолчанию, если вход не задан, то отчет сформируется в формате pdf. В случае если требуется отображать отчет в клиенте, то желательно выбирать формат сохранения html или pdf. Для экспорта в MS Excel требуется установить значение xlsx.</p>

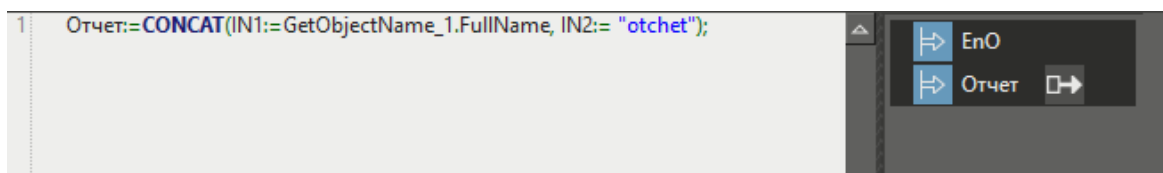
Название	Тип	Назначение
Report	REF_TO	<p>Служит для установления прямой связи с отчетом. Если использовать данный вход, то вход Name можно не использовать.</p> <p>Способы установления связи с отчетом:</p> <ul style="list-style-type: none"> • Если ФБ используется вне программы, добавлен в дерево. Перетащить отчет на данный вход в дереве. • Если ФБ используется внутри программы. В программу добавляется параметр типа REF_TO. На данный параметр перетаскивают отчет, а уже этот параметр связывают с данным входом ФБ
Выходы		
Building	BOOL	Информирует, что идет формирование отчета. Этот выход принимает значение TRUE на время генерации отчета.
Error	STRING	Информирует о возникновении ошибок при генерации отчета. Текст ошибки записывается в этот выход.
Path	STRING	<p>Указывает путь к созданному файлу отчета. Этот путь можно использовать при создании проекта для открытия отчета в контейнере окна, копирования, печати и др.</p> <p>Если генерация отчета завершена успешно, то отчет создается в папке reports, которая, в свою очередь, создается в папке, указанной на входе InPath. Если имя отчета, задано кириллицей то сформированный отчет будет иметь имя: <i><транслитерация имени отчета в проекте>_YYYY_mm_DD_HH_MM_SS.<значение входа Format></i>.</p>

Рекомендации по работе с отчетом

Формирование входа Name

При создании библиотечного объекта (типа объекта), который содержит и сам отчет, и инструменты для его формирования (функциональные блоки, дополнительные параметры), когда точное положение отчета в дереве объектов неизвестно, можно использовать выход FullName функционального блока GetObjectName, и функцию CONCAT

Пример программы:



В приведенном примере создаваемый в библиотеке отчет носит имя otchet

Между входом Name функционального блока ReportFB и выходом получившейся программы устанавливается связь.

6.4.3.1.7. GET_CURRENT_USERNAME

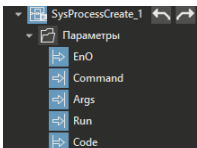
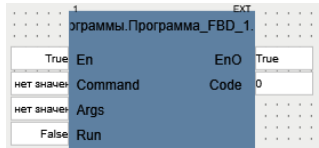
Функциональный блок GET_CURRENT_USERNAME выдает информацию о пользователе, который авторизовался в клиенте, подключенном к среде исполнения. У сервера могут быть одновременно несколько клиентов, и на каждом клиенте авторизован свой пользователь, поэтому функциональный блок должен работать в задаче экрана. Этот функциональный блок может использоваться как самостоятельно, вне программ, так и в составе любой программы (при условии, что программа или объект, в котором находится ФБ, выполняется в задаче экрана).

Вид в дереве		Вид в редакторе FBD
Название	Тип	Назначение
Username	STRING	Выход ФБ. Имя текущего пользователя.
LoginTime	DT	Выход ФБ. Время начала сессии (время входа в систему).

UserGroup	STRING	Выход ФБ. Роль пользователей, в которую назначен данный пользователь.
ClientAddress	STRING	Выход ФБ. Адрес устройства, на котором открыт клиент визуализации
SessionExpireTime	DT	Выход ФБ. Время окончания сессии пользователя. Значение 0 указывает на то, что время сессии не ограничено
FullName	STRING	Выход ФБ. Выдается полное имя пользователя, в случае если авторизован пользователь Active Directory

6.4.3.1.8. SYSPROCESSCREATE

Функциональный блок SysProcessCreate служит для запуска различных процессов и выполнения команд операционной системы из проекта, работающего в среде исполнения. Например: копирования файлов, печати сохраненных отчетов и других задач. Находится в библиотеке BaseObjects. Этот функциональный блок может использоваться как самостоятельно, вне программ, так и в составе любой программы.

Вид в дереве	Вид в редакторе FBD	Вид в редакторе ST
		<pre>SysProcessCreate_1(Command:="chrome.exe", Args:="http://127.0.0.1:8043",Run:=Запустить); //(где: Запустить - это логический параметр, связанный с кнопкой без фиксации в окне).</pre>

Для корректной работы необходимо определить входы функционального блока.

Название	Тип	Назначение
Command	STRING	Вход ФБ. Задается имя исполняемого файла (может включать полный путь) или команды. Например, если вход принимает значение iexplore.exe, то в результате работы функционального блока откроется Internet Explorer. Если процесс запускается через bat-файл, то вход может принимать значение D:\print.bat.

Args	STRING	Вход ФБ. Аргументы, которые могут быть использованы для запуска процесса или команды.
Run	BOOL	Вход ФБ. По переднему фронту изменения значения на этом входе среда исполнения попытается запустить процесс, указанный на входе Command, используя строку аргументов, заданной на входе Args.
Code	DINT	Выход ФБ. Выдается код ошибки (код ошибки зависит от выполняемой команды).

Следует учитывать, что при использовании данного функционального блока могут возникнуть неисправности, не связанные с работой MasterSCADA 4D. Разработчик проекта берет на себя ответственность за возможные сбои в работе сторонних процессов.

Важно! Для непрерывных операций необходимо использовать ФБ SysProcessCreateAsync. В противном случае, пока выполняется команда, указанная в ФБ, другие задачи исполняются не будут.

Особенности работы функционального блока в исполнительной системе для Linux

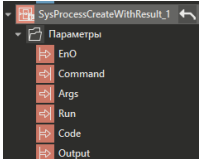
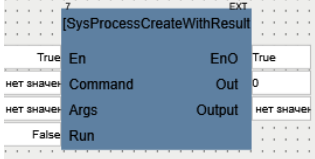
Для запуска GUI-приложений (например, Firefox) требуется указывать имя пользователя ОС, который работает в текущий момент. Для этого необходимо использовать команду su.

В этом случае на вход Command необходимо подать su - [имя пользователя] [имя приложения], либо на вход Command подать su - [имя пользователя], а на вход Args - [имя приложения].

Пример записи в ST: SysProcessCreate_1(Command:="su - admin firefox", Args:="http://127.0.0.1:8043",Run:=Запустить); //(где Запустить это логический параметр связанный с кнопкой без фиксации в окне.)

6.4.3.1.9. SYSPROCESSCREATEWITHRESULT

Функциональный блок SysProcessCreateWhithResult служит для запуска различных процессов и выполнения команд операционной системы из проекта, работающего в среде исполнения. Например: копирования файлов, печати сохраненных отчетов и других задач. При этом, в отличие от ФБ SysProcessCreate, он может возвращать какие-либо значения. Находится в библиотеке BaseObjects. Этот функциональный блок может использоваться как самостоятельно, вне программ, так и в составе любой программы.

Вид в дереве	Вид в редакторе FBD	Вид в редакторе ST
		<pre>SysProcessCreate_1(Command:="chrome.exe", Args:="http://127.0.0.1:8043",Run:=Запустить); //(где: Запустить - это логический параметр, связанный с кнопкой без фиксации в окне).</pre>

Для корректной работы необходимо определить входы функционального блока.

Название	Тип	Назначение
Входы		
Command	STRING	Задается имя исполняемого файла (может включать полный путь) или команды. Например, если вход принимает значение iexplore.exe, то в результате работы функционального блока откроется Internet Explorer. Если процесс запускается через bat-файл, то вход может принимать значение: D:\print.bat.
Args	STRING	Аргументы, которые могут быть использованы для запуска процесса или команды.
Run	BOOL	По переднему фронту изменения значения на этом входе среда исполнения попытается запустить процесс, указанный на входе Command, используя строку аргументов, заданную на входе Args.
Выходы		
Code	INT	Выдается код ошибки (код ошибки зависит от выполняемой команды).
Output	STRING	Результат вызова команды.

Следует учитывать, что при использовании данного функционального блока могут возникнуть неисправности, не связанные с работой MasterSCADA 4D. Разработчик проекта берет на себя ответственность за возможные сбои в работе сторонних процессов.

Важно! Для непрерывных операций необходимо использовать ФБ SysProcessCreateWithResultAsync. В противном случае, пока выполняется команда, указанная в ФБ, другие задачи исполняются не будут.

Особенности работы функционального блока в исполнительной системе для Linux

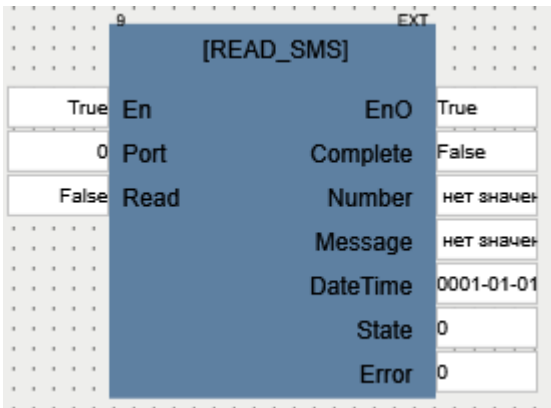
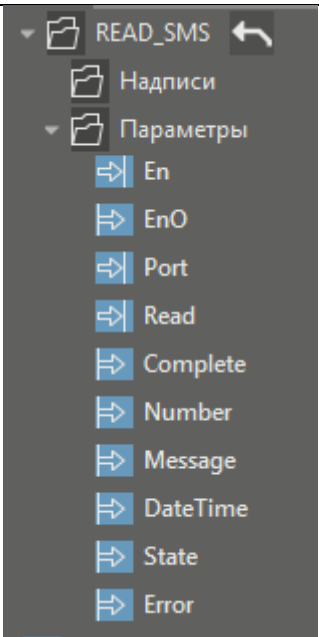
Для запуска GUI-приложений (например, Firefox) требуется указывать имя пользователя ОС, который работает в текущий момент. Для этого необходимо использовать команду su.

В этом случае на вход Command необходимо подать su - [имя пользователя] [имя приложения], либо на вход Command подать su - [имя пользователя], а на вход Args - [имя приложения].

Пример записи в ST: SysProcessCreateWithResult_1(Command:="su - admin firefox", Args:="http://127.0.0.1:8043",Run:=Запустить); //(где: Запустить - это логический параметр, связанный с кнопкой без фиксации в окне).

6.4.3.1.10.READ_SMS

Функциональный блок READ_SMS служит для получения SMS-сообщений через встроенный GSM-модем или GSM-модем, подключенный к COM-порту устройства. Этот функциональный блок может использоваться как внутри программы, так и непосредственно в дереве MasterSCADA.

Вид в редакторе FBD	Вид в дереве
	

Поддерживается получение фрагментированных SMS сообщений.

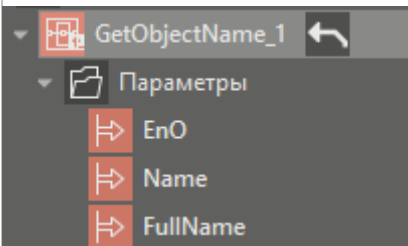
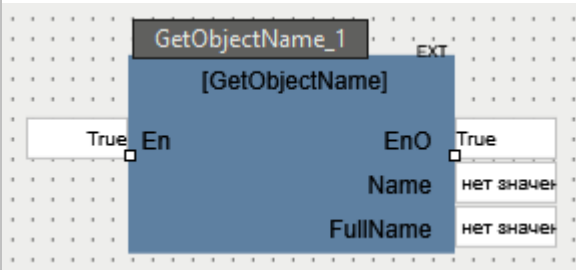
Входы и выходы ФБ:

Название	Тип	Назначение
Входы		
Port	DINT	В текущей версии на этот вход нужно подать тот же номер, который указан в настройке Параметры запуска RT.
Read	BOOL	Получить следующее SMS. Если после получения сообщения Read=TRUE, то производится получение следующего SMS. То есть, если на входе установлена константа TRUE, то будет происходить постоянное получение сообщений в цикле. .
Выходы		
Complete	BOOL	Устанавливается в TRUE при получении нового SMS, и сбрасывается в False при начале нового запроса). На каждое полученное SMS будет формироваться один импульс выхода Complete.
Number	STRING	Выдается номер телефона, с которого было отправлено сообщение.
Message	SRING	Текст сообщения.
DataTime	DT	Время приема последнего сообщения.
State	DINT	<p>Текущее состояние ФБ:</p> <ul style="list-style-type: none"> • 0 - Ожидание инициализации • 1 - Ожидание команды • 2 - Начало чтения • 33 - Чтения 1 SMS • 34 - Удаление прочитанного SMS • 16- Инициализация отправки с модема

Название	Тип	Назначение
		<ul style="list-style-type: none"> • 17 - Начало отправки SMS • 18 - Отправка SMS
Error	DINT	<p>Номер ошибки:</p> <ul style="list-style-type: none"> • 0 - Нет ошибок • 1 - Нет подключения к модему • 5 - Ошибка разбора PDU • 6 - Превышен лимит ожидания от модема

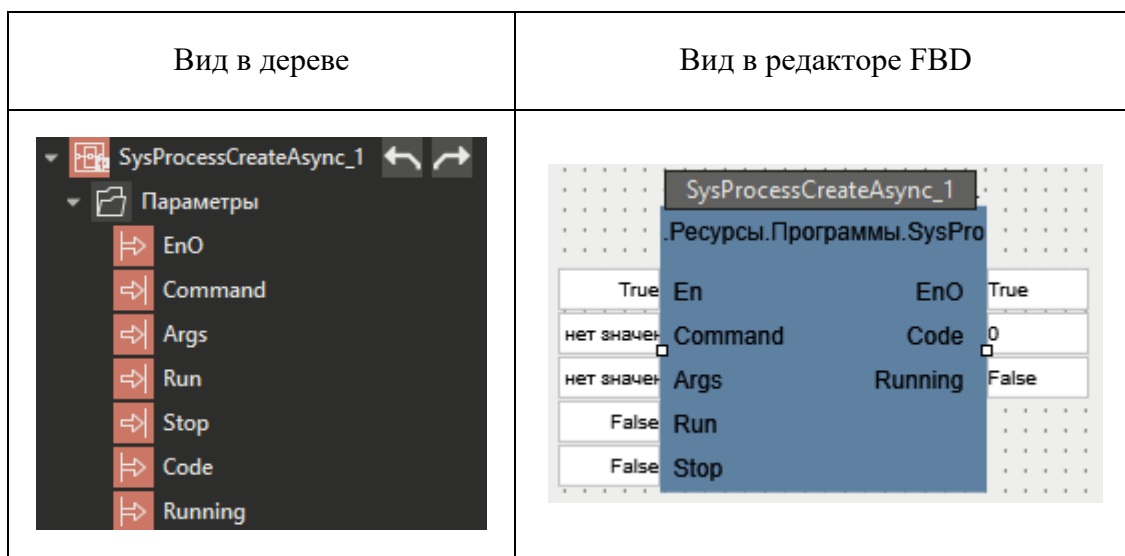
6.4.3.1.11.GETOBJECTNAME

Функциональный блок GetObjectName выдает информацию об объекте (или экземпляре объекта), в котором он сам находится.

Вид в дереве		Вид в редакторе FBD
		
Название	Тип	Назначение
Name	STRING	Выход ФБ. Отображает Имя объекта. Например, Насосная.
FullName	STRING	Выход ФБ. Отображается Полное имя объекта. Например, Объекты.Завод.Инженерные сооружения.Насосная.

6.4.3.1.12.SYSPROCESSCREATEASYNC

Функциональный блок SysProcessCreateAsync служит для запуска различных процессов и выполнения команд операционной системы из проекта, работающего в среде исполнения. При этом команда, вызванная с использованием данного ФБ, будет работать независимо от цикла работы исполнительной системы. Например: ping -t, telnet и др.



Для корректной работы необходимо определить входы функционального блока.

Название	Тип	Назначение
Command	STRING	Вход ФБ. Задается имя исполняемого файла (может включать полный путь) или команды. Например, если вход принимает значение iexplore.exe, то в результате работы функционального блока откроется Internet Explorer. Если процесс запускается через bat-файл, то вход может принимать значение: D:\print.bat.
Args	STRING	Вход ФБ. Аргументы, которые могут быть использованы для запуска процесса или команды.
Run	BOOL	Вход ФБ. По переднему фронту изменения значения на этом входе среда исполнения попытается запустить процесс, указанный на входе Command, используя строку аргументов, заданную на входе Args.
Stop	BOOL	Вход ФБ. Останавливает выполнение процесса.
Code	DINT	Выход ФБ. Выдается код ошибки (код ошибки зависит от выполняемой команды).
Running	BOOL	Выход ФБ. Показывает выполнение процесса. Если значение TRUE, то процесс выполняется.

Стоит учитывать, что при использовании данного функционального блока могут возникнуть неисправности, не связанные с работой MasterSCADA 4D. Разработчик проекта берет на себя ответственность за возможные сбои в работе сторонних процессов.

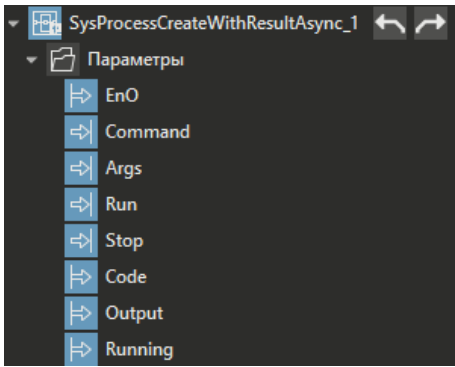
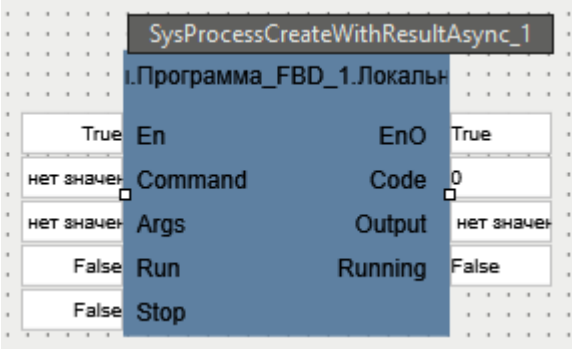
Особенности работы функционального блока в исполнительной системе для Linux

Для запуска GUI-приложений (например, Firefox) требуется указывать имя пользователя ОС, который работает в текущий момент. Для этого необходимо использовать команду `su`.

В этом случае на вход `Command` необходимо подать `su - [имя пользователя] [имя приложения]`, либо на вход `Command` подать `su - [имя пользователя]`, а на вход `Args` - `[имя приложения]`.

6.4.3.1.13.SYSPROCESSCREATEWITHRESULTASYNC

Функциональный блок `SysProcessCreateWhithResultAsync` служит для запуска различных процессов и выполнения команд операционной системы из проекта, работающего в среде исполнения. При этом, в отличие от ФБ `SysProcessCreateAsync`, он может возвращать какие-либо значение. Находится в библиотеке `BaseObjects`. Этот функциональный блок может использоваться как самостоятельно, вне программ, так и в составе любой программы.

Вид в дереве	Вид в редакторе FBD
	

Для корректной работы необходимо определить входы функционального блока.

Название	Тип	Назначение
Входы		
Command	STRING	Задается имя исполняемого файла (может включать полный путь) или команды. Например, если вход принимает значение <code>ieexplore.exe</code> , то в результате работы функционального блока откроется Internet Explorer. Если

Название	Тип	Назначение
		процесс запускается через bat-файл, то вход может принимать значение: D:\print.bat.
Args	STRING	Аргументы, которые могут быть использованы для запуска процесса или команды.
Run	BOOL	По переднему фронту изменения значения на этом входе среда исполнения попытается запустить процесс, указанный на входе Command, используя строку аргументов, заданную на входе Args.
Stop	BOOL	Останавливает выполнение процесса.
Выходы		
Code	INT	Выдается код ошибки (код ошибки зависит от выполняемой команды).
Output	STRING	Результат вызова команды.
Running	BOOL	Показывает выполнение процесса. Если значение TRUE - процесс выполняется.

Важно! Стоит учитывать, что при использовании данного функционального блока могут возникнуть неисправности, не связанные с работой MasterSCADA 4D. Разработчик проекта берет на себя ответственность за возможные сбои в работе сторонних процессов.

Особенности работы функционального блока в исполнительной системе для Linux

Для запуска GUI-приложений (например, Firefox) требуется указывать имя пользователя ОС, который работает в текущий момент. Для этого необходимо использовать команду su.

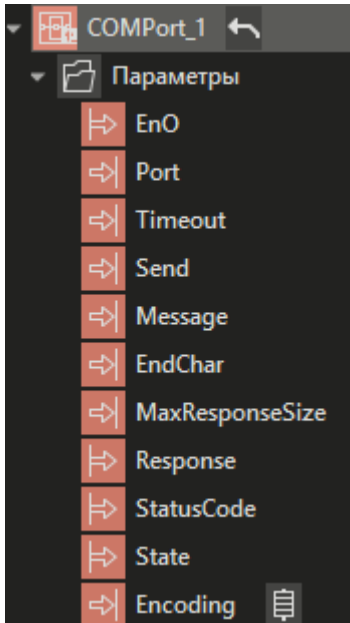
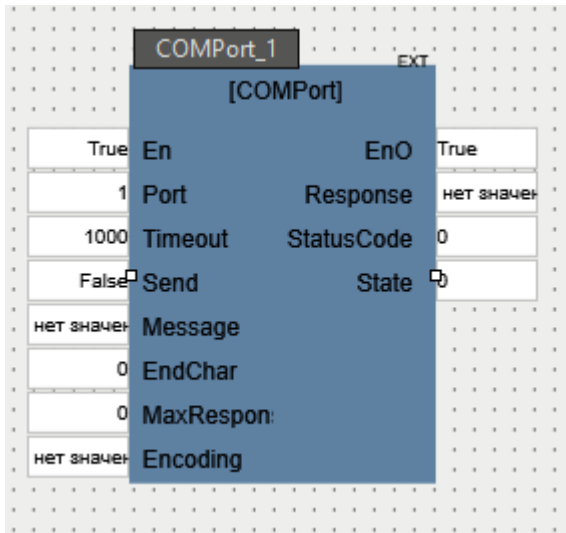
В этом случае на вход Command необходимо подать su - [имя пользователя] [имя приложения], либо на вход Command подать su - [имя пользователя], а на вход Args - [имя приложения].

Пример записи в ST: SysProcessCreate_1(Command:="su - admin firefox", Args:="http://127.0.0.1:8043",Run:=Запустить); //(где: Запустить - это логический параметр, связанный с кнопкой без фиксации в окне).

6.4.3.1.14.COMPORT

Функциональный блок COMPort служит для работы с COM-портом устройства (со строковыми данными), на котором установлена среда исполнения. Находится в библиотеке BaseObjects. Этот функциональный блок может использоваться как самостоятельно, вне программ, так и в составе любой программы.

Важно! Для корректной работы функционального блока необходимо добавить протокол, который работает по COM-порту, например, Modbus RTU, в задаче которого будет работать ФБ. На вход функционального блока нужно подать имя того порта, который настроен в протоколе. При работе функциональный блок будет использовать настройки (скорость, четность и др), указанные в протоколе. Если протокол имеет в своем составе модули ввода-вывода, то сначала будет произведен опрос модулей, а затем будет произведена работа функциональных блоков.

Вид в дереве	Вид в редакторе FBD
	

Входы и выходы ФБ:

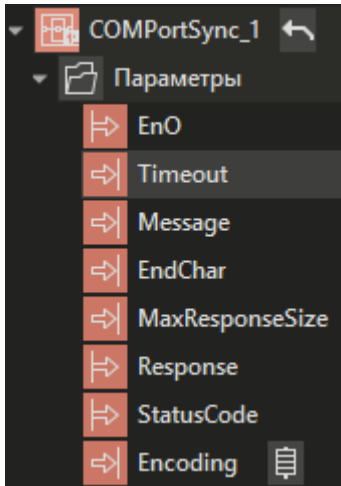
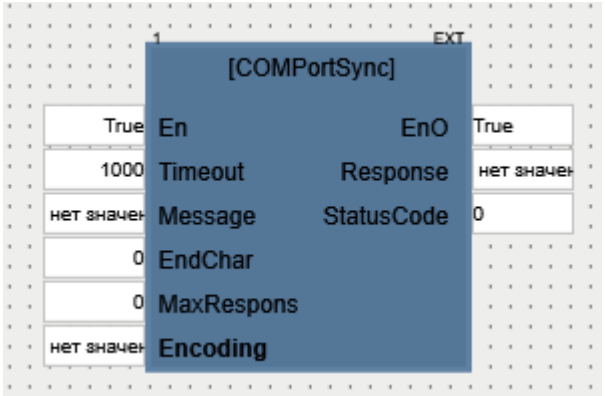
Название	Тип	Назначение
Входы		

Название	Тип	Назначение
Port	DINT	Задается номер COM-порта протокола, в задаче которого должен работать данный блок.
Timeout	DINT	Задается время ожидания. По умолчанию установлено значение 1000. Если данный параметр не требуется, то необходимо установить значение 0.
Send	BOOL	Отправляет запрос по переднему фронту изменения значения на этом входе.
Message	STRING	Задается текст запроса.
EndChar	DINT	Устанавливается завершающий символ строки (если 0, то не используется).
MaxResponse	INT	Задается максимальное количество символов в ответе. По умолчанию установлено значение 0, означающее, что используется стандартный размер буфера.
Encoding	Encoding	<p>Определяется соответствие, с которым идет преобразование строки в массив байт и обратно. Возможные варианты:</p> <ul style="list-style-type: none"> • UTF8 (по умолчанию), • CP1251, • ASCII
Выходы		
Response	STRING	Выдается ответ
StatusCode	DINT	<p>Выдается код ответа:</p> <ul style="list-style-type: none"> • 5 — ответ получен по истечению Timeout; • 0 — ответ получен по достижению завершающего символа (или по достижению MaxResponse символов в ответе)
State	DINT	Показывает состояние ФБ.

Название	Тип	Назначение
		<ul style="list-style-type: none"> • 0 — ожидание регистрации порта; • 1 — ожидание отправки запроса; • 2 — отправка запроса. <p>Важно! Если нет протокола с таким же номером COM-порта, как указан на входе Port, то всегда будет выдаваться значение State - 0.</p>

6.4.3.1.15.COMPORTSYNC

Функциональный блок COMPortSync служит для синхронной отправки сообщений на COM-порт (со строковыми данными). Находится в библиотеке BaseObjects.

Вид в дереве	Вид в редакторе FBD
	

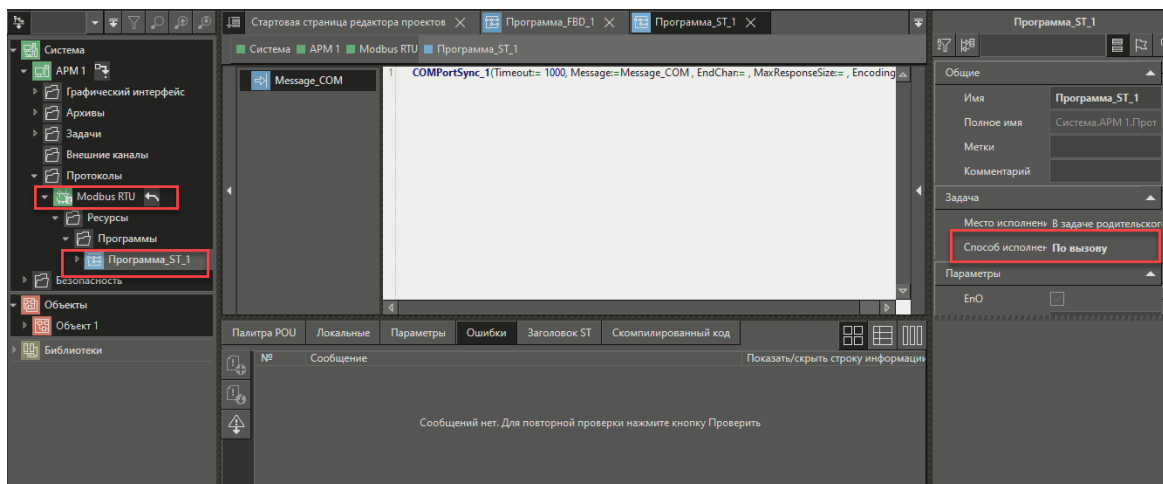
Входы и выходы ФБ:

Название	Тип	Назначение
Входы		
Timeout	DINT	Задается время ожидания. По умолчанию установлено значение 1000. Если данный параметр не требуется, то необходимо установить значение 0.
Message	STRING	Задается текст запроса.

Название	Тип	Назначение
EndChar	DINT	Устанавливается завершающий символ строки (если 0, то не используется).
MaxResponse	INT	Задается максимальное количество символов в ответе. По умолчанию установлено значение 0, означающее, что используется стандартный размер буфера.
Encoding	Encoding	<p>Определяется соответствие, с которым идет преобразование строки в массив байт и обратно. Возможные варианты:</p> <ul style="list-style-type: none"> • UTF8 (по умолчанию), • CP1251, • ASCII
Выходы		
Response	STRING	Выдается ответ
StatusCode	DINT	<p>Выдается код ответа:</p> <ul style="list-style-type: none"> • 5 — ответ получен по истечению Timeout; • 0 — ответ получен по достижению завершающего символа (или по достижению MaxResponse символов в ответе)

Рекомендации по работе

Для работы с COM-портом в проект необходимо добавить протокол, работающий по COM-порту (например, ModbusRTU). В настройках протокола задать настройки COM-порта (порт, скорость, стоп бит и тд). и в него добавить программу, обычно используется программа ST, в которую добавляют уже данный ФБ. Способ исполнения программы необходимо установить По вызову:



Отправка сообщений осуществляется по факту вызова функционального блока.

Если используется программа FBD, то управлять вызовом ФБ можно при помощи входа En.

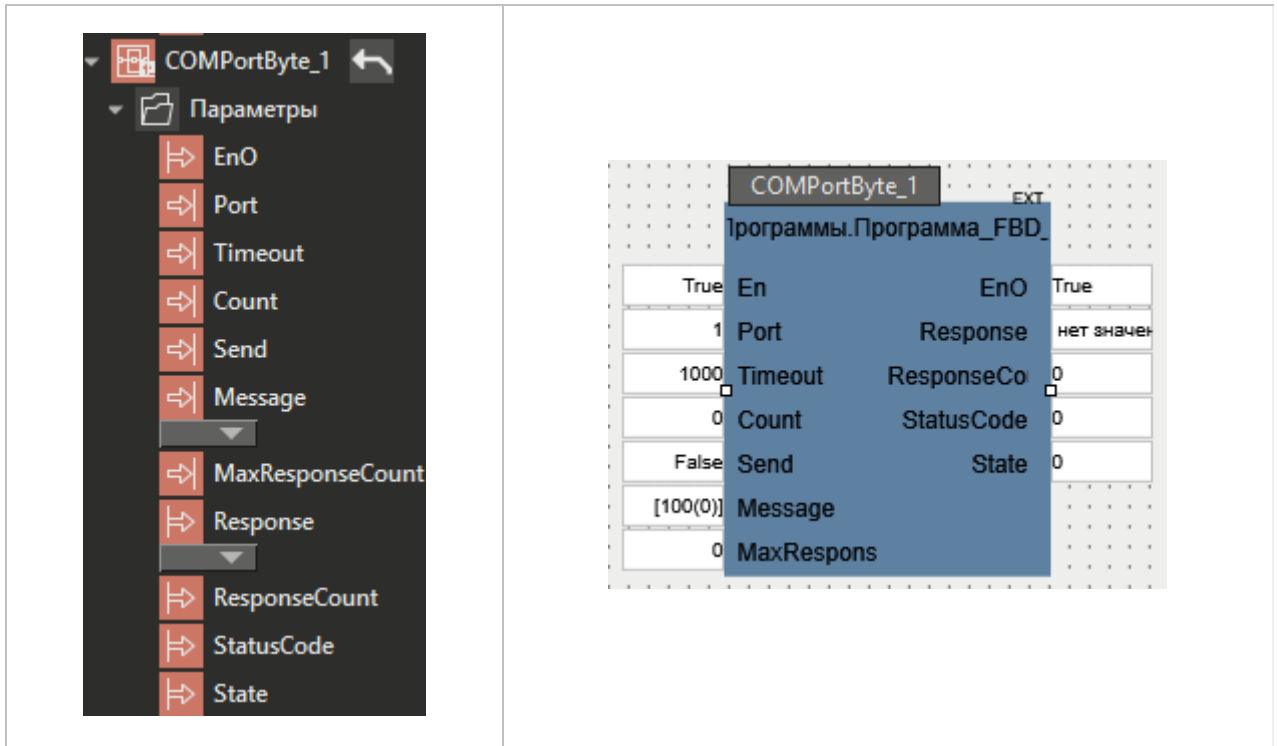
6.4.3.1.16.COMPORTBYTE

Функциональный блок COMPortByte служит для работы с COM-портом устройства (с данными типа ARRAY [1..100] of BYTE), на котором установлена среда исполнения. Находится в библиотеке BaseObjects. Этот функциональный блок может использоваться как самостоятельно, вне программ, так и в составе любой программы.

Важно! Для корректной работы функционального блока необходимо добавить протокол, который работает по COM-порту, например, Modbus RTU, в задаче которого будет работать ФБ. На вход функционального блока нужно подать имя того порта, который настроен в протоколе. При работе функциональный блок будет использовать настройки (скорость, четность и др), указанные в протоколе. Если протокол имеет в своем составе модули ввода-вывода, то сначала будет произведен опрос модулей, а затем будет произведена работа функциональных блоков.

Вид в дереве

Вид в редакторе FBD



Входы и выходы ФБ:

Название	Тип	Назначение
Входы		
Port	DINT	Задается номер COM-порта протокола, в задаче которого должен работать данный блок.
Timeout	DINT	Задается время ожидания. По умолчанию установлено значение 1000. Если данный параметр не требуется, то необходимо установить значение 0.
Count	INT	Задается количество байт, которое необходимо передать.
Send	BOOL	Отправляет запрос по переднему фронту изменения значения на этом входе.
Message	ARRAY [1..100] of BYTE	Задается запрос.

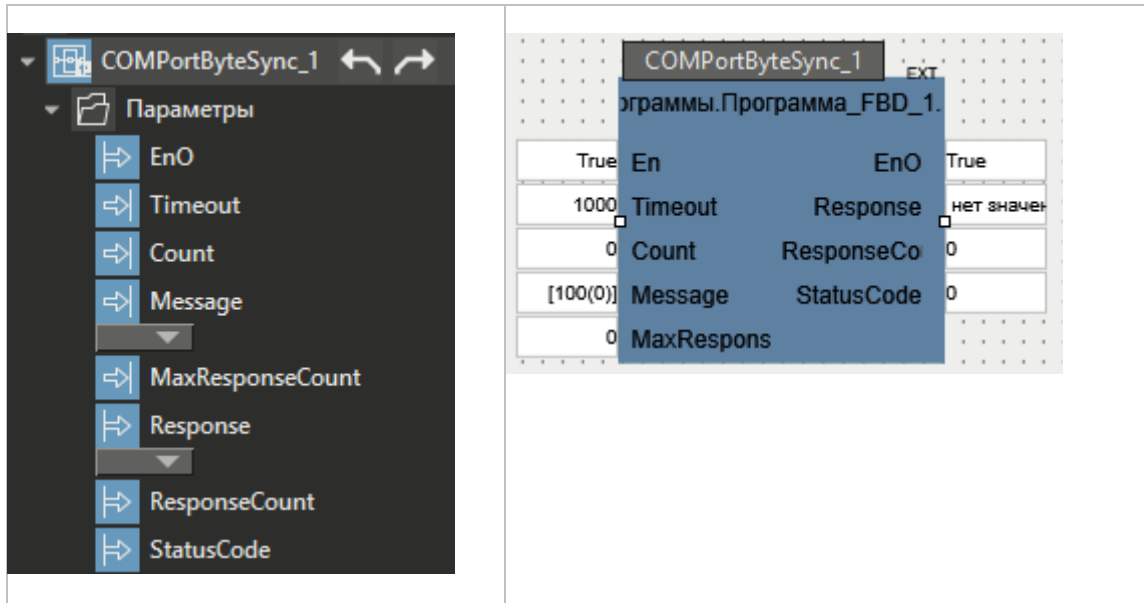
Название	Тип	Назначение
MaxResponseCount	INT	Задается максимальное количество символов в ответе. По умолчанию установлено значение 0, означающее, что используется стандартный размер буфера.
Выходы		
Response	ARRAY [1..100] of BYTE	Выдается ответ.
ResponseCount	INT	Показывается фактическое количество полученных байт.
StatusCode	DINT	Выдается код ответа: <ul style="list-style-type: none"> • 5 — ответ получен по истечению Timeout; • 0 — ответ получен по достижению количества символов в ответе, заданном в MaxResponseCount.
State	DINT	Показывает состояние ФБ. <ul style="list-style-type: none"> • 0 — ожидание регистрации порта; • 1 — ожидание отправки запроса; • 2 — отправка запроса. <p>Важно! Если нет протокола с таким же номером COM-порта, как указан на входе Port, то всегда будет выдаваться значение State - 0.</p>

6.4.3.1.17.COMPORTBYTESYNC

Функциональный блок COMPortByteSync служит для работы с COM-портом устройства (с данными типа ARRAY [1..100] of BYTE), на котором установлена среда исполнения. Данный ФБ обеспечивает синхронную отправку сообщения и ожидание получения ответа.

Находится в библиотеке BaseObjects.

Вид в дереве	Вид в редакторе FBD
--------------	---------------------



Входы и выходы ФБ:

Название	Тип	Назначение
Входы		
Timeout	DINT	Задается время ожидания. По умолчанию установлено значение 1000. Если данный параметр не требуется, то необходимо установить значение 0.
Count	INT	Задается количество байт, которое необходимо передать.
Message	ARRAY [1..100] of BYTE	Задается запрос.
MaxResponse Count	INT	Задается максимальное количество символов в ответе. По умолчанию установлено значение 0, означающее, что используется стандартный размер буфера
Выходы		
Response	ARRAY [1..100] of BYTE	Выдается ответ.

Название	Тип	Назначение
ResponseCount	INT	Показывается фактическое количество полученных байт.
StatusCode	DINT	<p>Выдается код ответа:</p> <ul style="list-style-type: none"> • 5 — ответ получен по истечению Timeout; • 0 — ответ получен по достижению количества символов в ответе, заданном в MaxResponseCount.

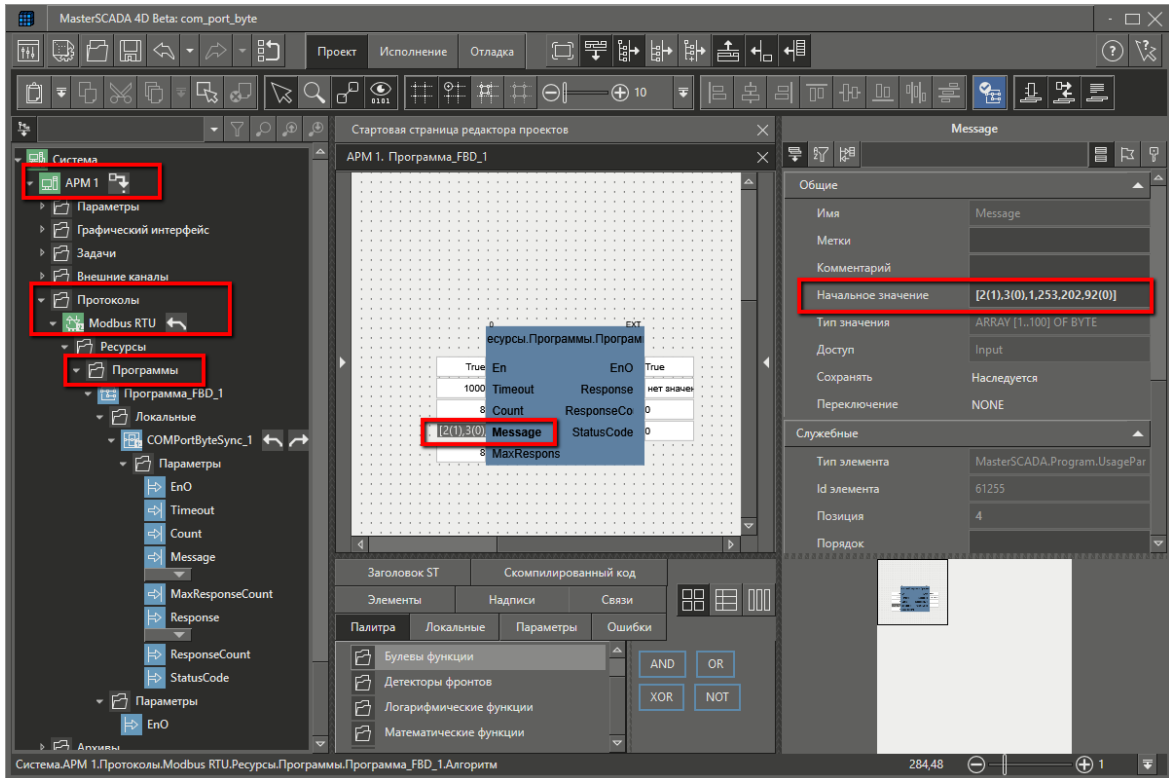
Важно! COMPortByteSync допустимо использовать только в программе, расположенной в протоколе, поэтому номер порта не задается (используется всегда порт протокола, в котором выполняется программа).

Отправка запроса будет производиться с периодом, указанным в свойствах протокола в категории Задача.

У данного ФБ отсутствует вход SEND, имеющийся у ФБ COMPortByte, т.к. ФБ COMPortByteSync выполняется синхронно, т. е. отслеживания выполнения не требуется. В случае если отправку производить не нужно, то стандартному входу En следует присвоить значение False.

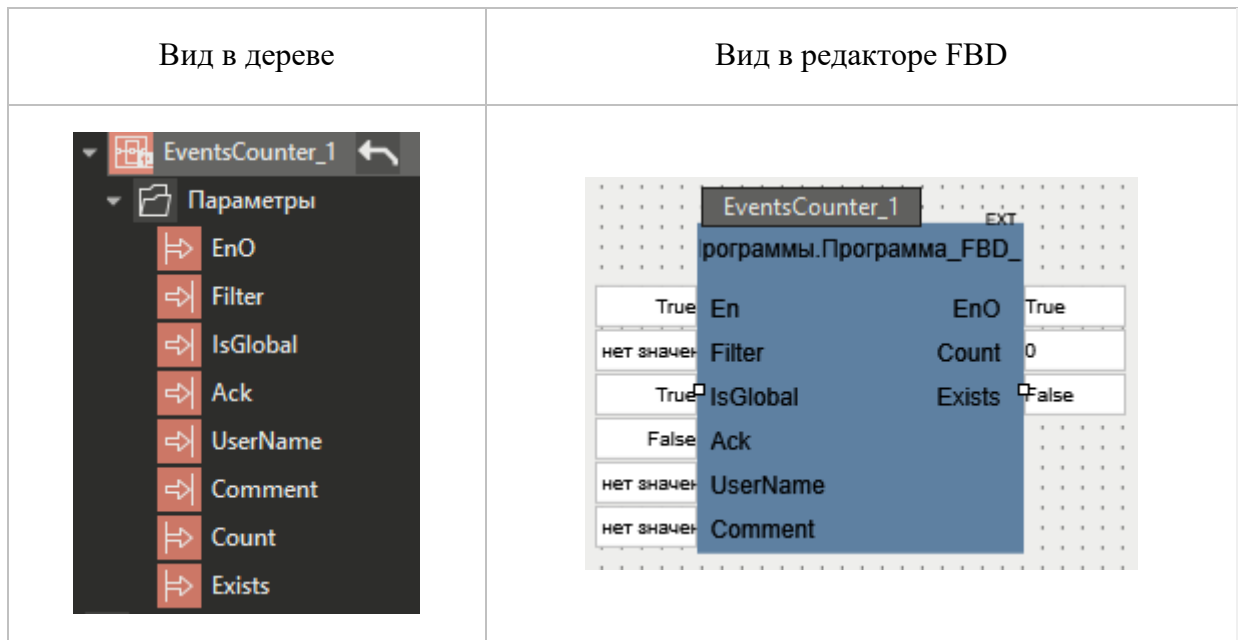
Порядок работы

1. Добавить в узел какой-либо протокол, например, Modbus RTU.
2. В настройках протокола установить номер СОМ-порта.
3. Добавить в протокол в какую-либо Программу.
4. Добавить в программу нужные ФБ.
5. Установить связи с параметрами проекта или установить константы на входах ФБ.



6.4.3.1.18.EVENTSCOUNTER

Функциональный блок EventsCounter служит для подсчета количества сообщений, удовлетворяющих заданному фильтру. Позволяет также квитировать сообщения.

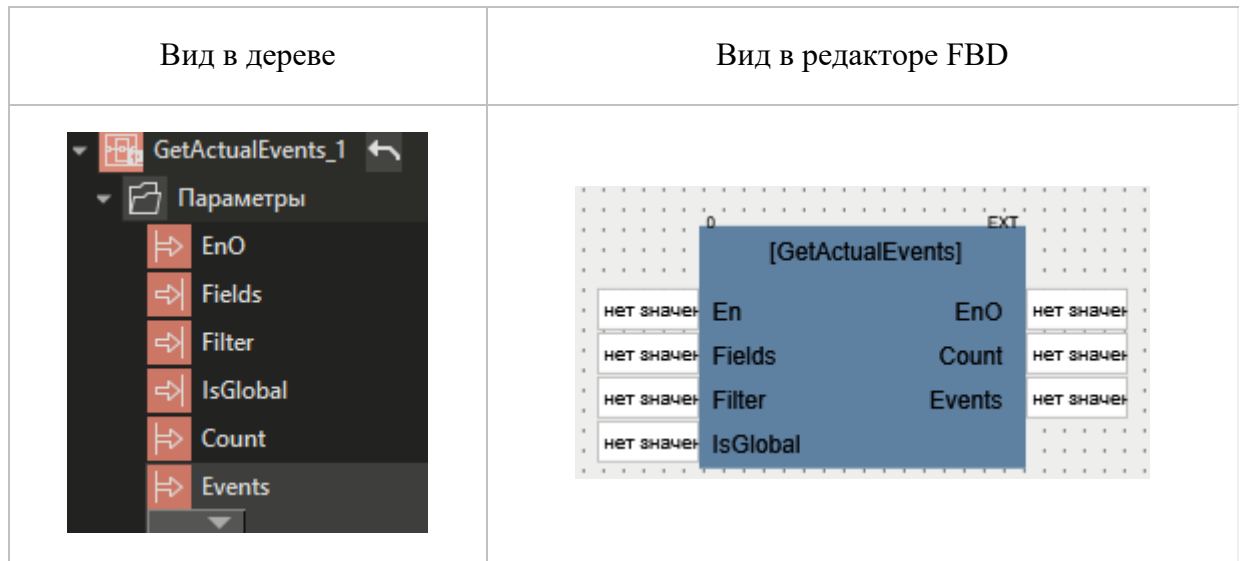


Входы и выходы ФБ:

Название	Тип	Назначение
Входы		
Filter	STRING	Задается фильтр, определяющий сообщения, которые должны быть подсчитаны. Пример, ["Active=TRUE", "Severity <= 100 or Severity >= 900"], где: Active соответствует свойству тревоги Активность, а Severity соответствует свойству Приоритет. Полный список соответствий приведен в разделе Список программных имен параметров тревог
IsGlobal	BOOL	Определяет место поиска сообщений. Если вход принимает значение False, то ФБ подсчитывает сообщения только того объекта (и его подобъектов), в котором находится.
Ack	BOOL	Позволяет квитировать выбранные сообщения.
UserName	STRING	Задается имя оператора, квитирующего сообщение. Т.к. функциональный блок работает на сервере (в среде исполнения), к которому может быть подключено несколько графических клиентов, то у ФБ отсутствует информация о том, какой именно оператор квитирует сообщение.
Comment	STRING	Комментарий, который можно добавить при квитировании сообщения.
Выходы		
Count	DINT	Показывает число текущих сообщений, удовлетворяющих заданному фильтру.
Exists	BOOL	Принимает значение TRUE, если есть хотя бы одно сообщение, удовлетворяющее заданному фильтру.

6.4.3.1.19.GETACTUALEVENTS

Функциональный блок GetActualEvents выдает количество и список сообщений, удовлетворяющих заданному фильтру. Список сообщений выдается в виде массива структур.



Входы и выходы ФБ:

Название	Тип	Назначение
Входы		
Fields	STRING	Перечисляются список полей, которые необходимо отобразить. Значения полей (параметры тревоги) вводятся через запятую. Указываются программные имена полей (параметров). Список полей и их программные имена смотрите в разделах Тревога и Список программных имен параметров тревог
Filter	STRING	Задается фильтр, определяющий сообщения, которые должны быть подсчитаны. Пример, ["Active=TRUE", "Severity <= 100 or Severity >= 900"], где: Active соответствует свойству тревоги Активность, а Severity соответствует свойству Приоритет. Полный список соответствий приведен в разделе Список программных имен параметров тревог.
IsGlobal	BOOL	Определяет место поиска сообщений. Если вход принимает значение False, то ФБ подсчитывает сообщения только того объекта (и его подобъектов), в котором находится.

Название	Тип	Назначение
Выходы		
Count	DINT	Показывает число сообщений, удовлетворяющих заданному фильтру.
Events	ARRAY [*] OF ANY	Выдает список сообщений, которые удовлетворяют условиям фильтра. Каждый элемент соответствует одному сообщению, количество полей структуры зависит от входа Fields.

Рекомендации по работе с ФБ

Для использования результатов полученных на выходе Events необходимо в пользовательской библиотеке создать тип данных массив структур переменной длины. Порядок и количество полей структуры должно соответствовать порядку и количеству полей, указанных на входе ФБ Fields. Затем создать параметр этого типа данных. Параметр связать уже с выходом блока Events и использовать в дальнейшем в проекте.

6.4.3.1.20.GETNEWEVENTS

Функциональный блок GetNewEvents при вызове возвращает только те сообщения, которые были получены в исполнительной системе с момента предыдущего вызова и удовлетворяют условиям фильтра. В фильтре не обязательно указывать "Active=true". В любом случае в выборку попадут только те сообщения, которые стали активными с момента прошлого вызова, а также безусловные сообщения, которые появились с момента прошлого вызова.

Вид в дереве	Вид в редакторе FBD
	

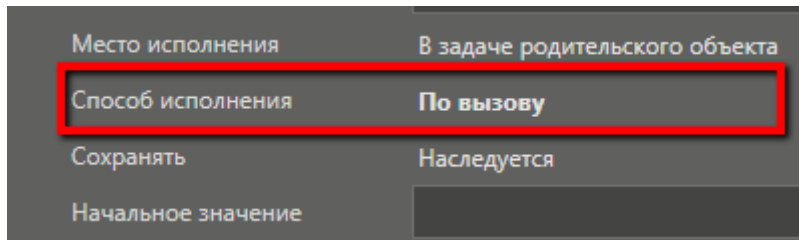
Входы и выходы ФБ:

Название	Тип	Назначение
Входы		
Fields	STRING	Перечисляются список полей, которые необходимо отобразить. Значения полей (параметры тревоги) вводятся через запятую. Указываются программные имена полей (параметров). Список полей и их программные имена смотрите в разделах Тревога и Список программных имен параметров тревог
Filter	STRING	Задается фильтр, определяющий сообщения, которые должны быть подсчитаны. Пример, ["Severity <= 100 or Severity >= 900"], где Severity соответствует свойству Приоритет. Полный список соответствий приведен в разделе Список программных имен параметров тревог.
IsGlobal	BOOL	Определяет место поиска сообщений. Если вход принимает значение False, то ФБ подсчитывает сообщения только того объекта (и его подобъектов), в котором находится.
Выходы		
Count	DINT	Показывает число сообщений, удовлетворяющих заданному фильтру.
Events	ARRAY [*] OF ANY	Выдает список сообщений, которые удовлетворяют условиям фильтра. Каждый элемент соответствует одному сообщению, количество полей структуры зависит от входа Fields.

Рекомендации по работе с ФБ

Для использования результатов полученных на выходе Events необходимо в пользовательской библиотеке создать тип данных массив структур переменной длины. Порядок и количество полей структуры должно соответствовать порядку и количеству полей, указанных на входе ФБ Fields. Затем создать параметр этого типа данных. Параметр связать уже с выходом блока Events и использовать в дальнейшем в проекте.

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



6.4.3.1.21. WRITEARCHIVEDATA

Функциональный блок WriteArchiveData предназначен для записи в архив значений заданного параметра. Этот функциональный блок может использоваться как внутри программы, так и непосредственно в дереве MasterSCADA.

Вид в редакторе FBD	Вид в дереве

Входы и выходы ФБ:

Название	Тип	Назначение
Входы		
Item	REF_TO	<p>Служит для связи с архивируемым параметром, значения которого требуется записать в архив. После установки связи входа ФБ с архивируемым параметром, на входе отобразится ID архивируемого параметра, который используется в таблицах баз данных.</p> <p>Важно! Если ФБ находится внутри программы, например, FBD или ST, то в программу необходимо добавить вход типа REF_TO, связать его с архивируемым параметром, а</p>

Название	Тип	Назначение
		затем уже этот вход программы связать со входом ФБ Item. Если ФБ находится внутри программы ST, то в программе можно указать полное имя архивируемого элемента. См раздел Специальные типы данных.
Value	ANY	Задается значение параметра, которое требуется записать в архив.
Time	DT	Указывается метка времени значения, которое записывается в архив. Если метка времени не задана (0), то ФБ присваивает текущее время.
StatusCode	DINT	Указывается признак качества, с которым требуется записать значение в архив.
IgnoreFilter	BOOL	Позволяет игнорировать настройки фильтра архивирования (то есть писать даже в случае если значение не изменилось).
AllowInsert	BOOL	Позволяет записать в архив значение даже если его метка времени не является последней из всех меток по данному параметру. В этом случае, при записи сбрасывается кеш данных по этому параметру, т.е. данные, накопленные в оперативной памяти, но еще не успевшие записаться в базу данных, будут удалены.

Рекомендации по работе с ФБ

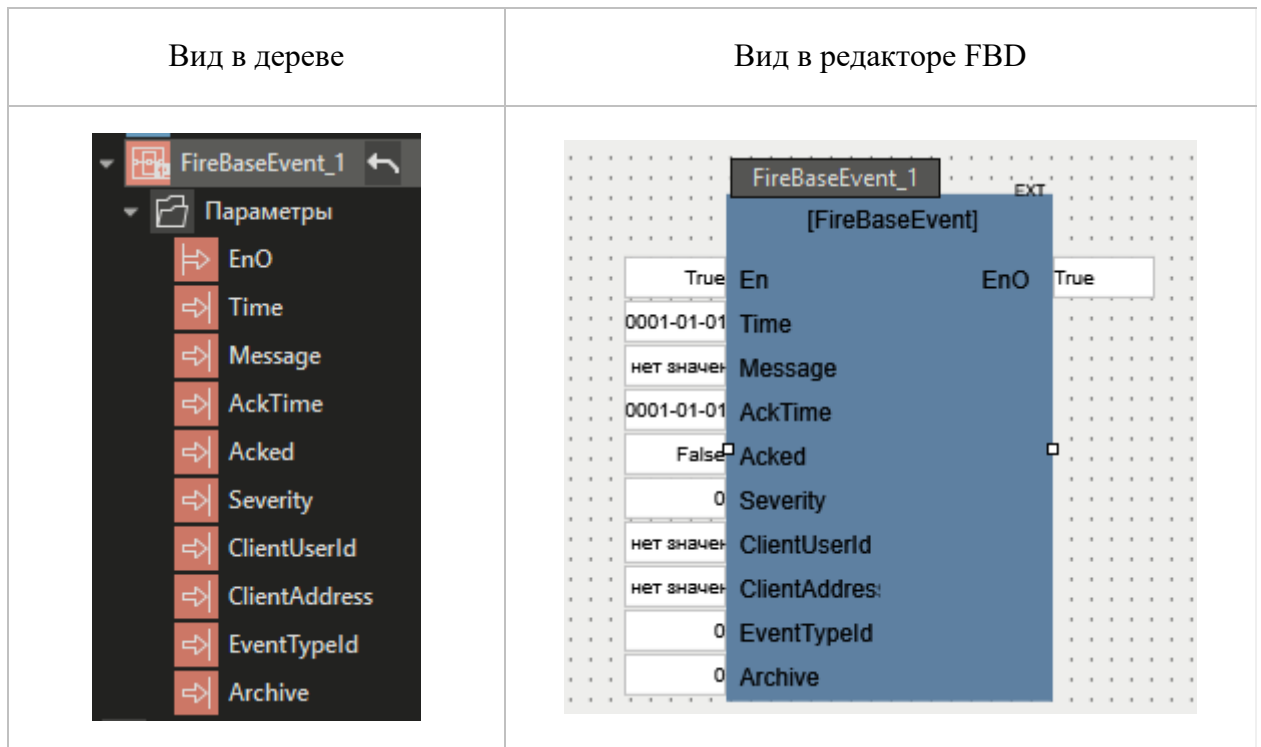
Запись значений в архив осуществляется по вызову ФБ. Для управления вызовом можно использовать стандартный вход En. Для контроля работы ФБ используется выход EnO.

Смотрите так же

Специальные типы данных.

6.4.3.1.22.FIREBASEEVENT

Функциональный блок FireBaseEvent служит для формирования безусловных сообщений. При помощи этого функционального блока сообщения могут быть сформированы на основе данных, полученных из внешней базы данных, а также на основе значений полученных непосредственно в MasterSCADA 4D



Входы и выходы ФБ:

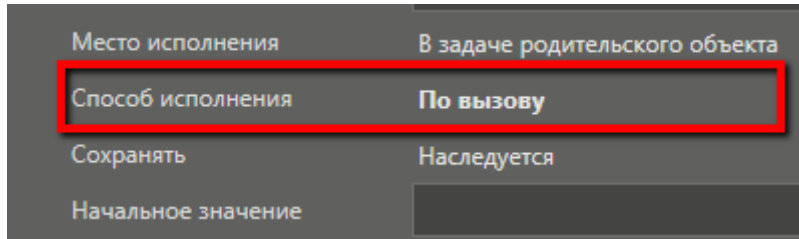
Название	Тип	Назначение
Входы		
Time	DT	Задается время возникновения сообщения.
Message	STRING	Задается текст сообщения.
AckTime	DT	Задается время квитирования.
Aacked	BOOL	Определяет было ли квитировано сообщение. Если вход равен TRUE, то сообщение считается квитированным.
Severity	DINT	Определяет приоритет сообщения.
ClientUserId	STRING	Указывается имя пользователя.
ClientAddress	STRING	Указывается адрес клиента визуализации.

Название	Тип	Назначение
EventTypeId	LINT	<p>Указывается ID источника сообщения. Если необходимо выдавать различные сообщения с одинаковым временем. В этом случае необходимо обеспечить уникальность пар <Time, EventTypeId>.</p> <p>Если выдаются сообщения, полученные от сторонней системы, то необходимо получить ID источников сообщений в этой сторонней системе, если такие есть, то их надо передавать на данный вход.</p> <p>Если такой возможности нет, и времена, выдаваемых сообщений могут совпадать, то нужно в вызывающей данный функциональный блок программе ST инкрементировать EventTypeId, если время нового сообщения совпадает с предыдущим. И сбрасывать в 0, если отличается. Пример текста ST:</p> <pre>if (newTime = prevTime) then typeId := typeId + 1 else typeId := 0 end_if;</pre>
Archive	REF_TO	<p>Определяет место сохранения сообщений, которые были сформированы с использованием данного функционального блока. Для этого необходимо связать нужный архив сообщений и данный параметр. Связь может быть установлена также и с архивом информационной безопасности. Если связь не установлена, то сообщения будут сохраняться в Основной архив сообщений.</p>

В случае если необходимо добавить дополнительные поля сообщения, то нужно добавить в пользовательскую библиотеку экземпляр функционального блока FireBaseEvent. Затем добавить в него нужное количество входов и далее в проекте использовать экземпляры нового функционального блока.

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



О том, была ли работа ФБ успешной, можно судить по выходу ФБ EnO. Если формирование сообщения будет неудачным, то выход примет значение False.

6.4.3.1.23.FILETOSTRING

Функциональный блок FileToString служит считывания данных из файла. Прочитанная информация выдается в виде текстовой строки. Находится в библиотеке BaseObjects. Этот функциональный блок может использоваться как самостоятельно, вне программ, так и в составе любой программы.

Вид в дереве	Вид в редакторе FBD

Входы и выходы ФБ:

Название	Тип	Назначение
Входы		
FileName	STRING	Указывается путь к файлу, его имя и расширение. Может быть задан как путь относительно рабочей папки исполнительной системы, так и абсолютный путь.
Read	BOOL	Читает из файла данные по переднему фронту и инициирует передачу на выход функционального блока.

Название	Тип	Назначение
Encoding	Encoding	<p>Определяет кодировку файла. Возможные варианты:</p> <ul style="list-style-type: none"> • UTF8 (по умолчанию), • CP1251, • ASCII
Выходы		
Output	STRING	Результирующая строка, полученная после изменения значения с False на TRUE входа Read
ErrorCode	DINT	Выдается код ошибки операционной системы. Для Windows и для Linux могут быть получены разные значения в одних и тех же ситуациях. Полный список ошибок смотрите в документации на операционную систему.

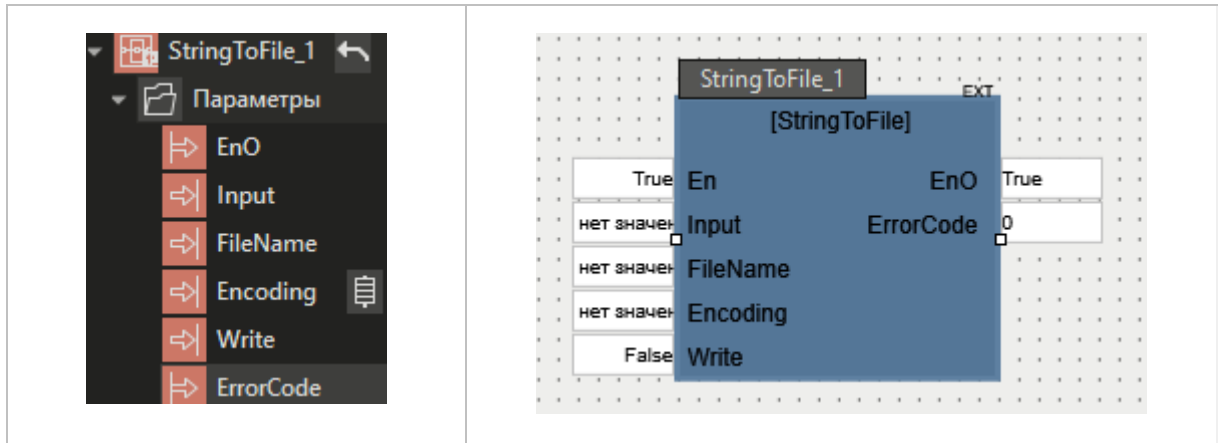
Примеры кодов:

Windows	Linux	Описание
2	2	Файл не найден
3	2	Путь к файлу не найден
5	13	Нет доступа к файлу

6.4.3.1.24.STRINGTOFILE

Функциональный блок StringToFile служит для записи данных в XML-файл. Находится в библиотеке BaseObjects. Этот функциональный блок может использоваться как самостоятельно, вне программ, так и в составе любой программы.

Вид в дереве	Вид в редакторе FBD
--------------	---------------------



Входы и выходы ФБ:

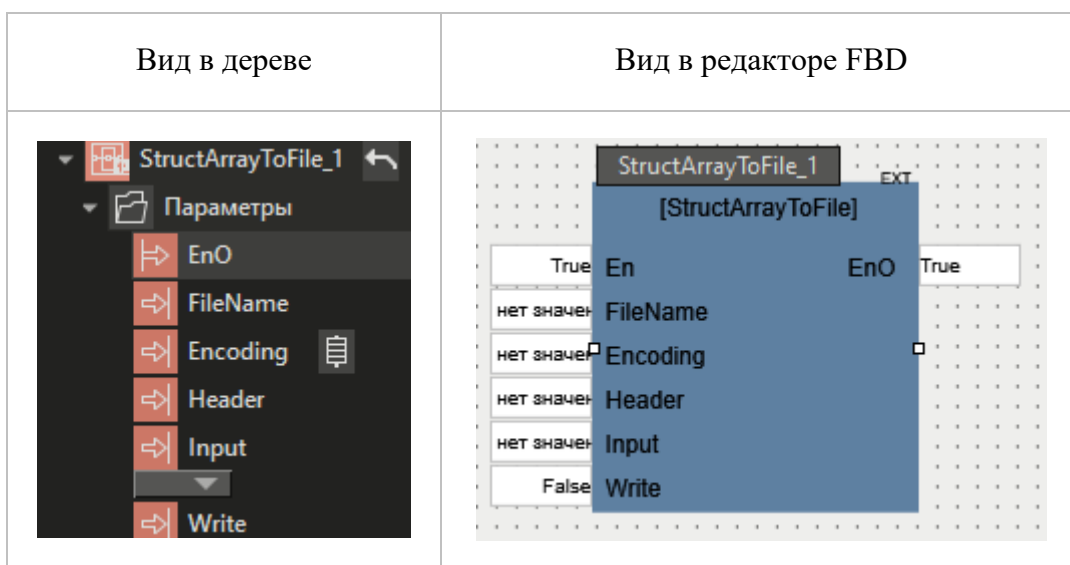
Название	Тип	Назначение
Входы		
Input	STRING	Указывается строка, которая будет записана в XML-файл.
FileName	STRING	Указывается путь к файлу, его имя и расширение. Путь может быть задан как путь относительно рабочей папки исполнительной системы, так и абсолютный путь.
Write	BOOL	Записывает данные в файл по переднему фронту.
Encoding	Encoding	<p>Определяется соответствие, с которым идет преобразование входной строки. Возможные варианты:</p> <ul style="list-style-type: none"> • UTF8 (по умолчанию), • CP1251, • ASCII
Выходы		
ErrorCode	DINT	Выдается код ошибки операционной системы. Для Windows и для Linux могут быть получены разные значения в одних и тех же ситуациях. Полный список ошибок смотрите в документации на операционную систему.

Примеры кодов:

Windows	Linux	Описание
0	0	Нет ошибок
2	2	Файл не найден
3	2	Путь к файлу не найден
5	13	Нет доступа к файлу

6.4.3.1.25.STRUCTARRAYTOFILE

Функциональный блок StructArrayToFile читает на входе получает массив структур и записывает полученные данные в CSV-файл. Находится в библиотеке BaseObjects. Этот функциональный блок может использоваться как самостоятельно, вне программ, так и в составе любой программы.



Входы и выходы ФБ:

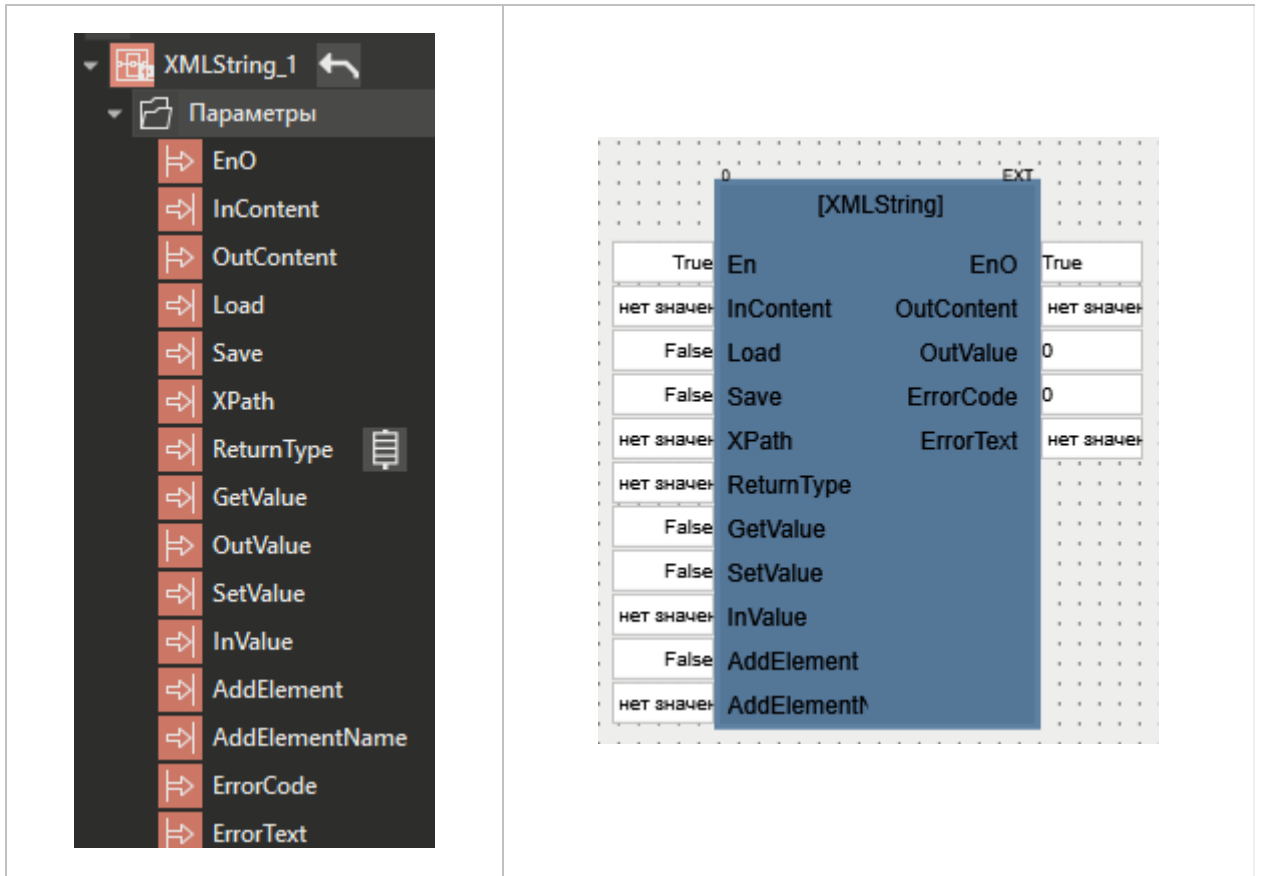
Название	Тип	Назначение
Входы		
FileName	STRING	Указывается путь к файлу, его имя и расширение. Путь может быть задан как путь относительно рабочей

Название	Тип	Назначение
		папки исполнительной системы, так и абсолютный путь.
Encoding	Encoding	<p>Определяется соответствие, с которым идет преобразование входного значения. Возможные варианты:</p> <ul style="list-style-type: none"> • UTF8 (по умолчанию), • CP1251, • ASCII
Header	STRING	Задается первая строка csv файла с заголовками через ; (A;B;C). Заголовок соответствует имени поля структуры. Должен быть на английском (в имени можно использовать только цифры и _)
Input	ARRAY [*] OF ANY	Задается массив структур, который нужно сохранить в csv-файл
Write	BOOL	Записывает данные в файл по переднему фронту.

6.4.3.1.26.XMLSTRING

Функциональный блок XMLString служит для загрузки XML в строковом виде, его модификации и сохранения. Находится в библиотеке BaseObjects. Этот функциональный блок может использоваться как самостоятельно, вне программ, так и в составе любой программы.

Вид в дереве	Вид в редакторе FBD
--------------	---------------------



Входы и выходы ФБ:

Название	Тип	Назначение
Входы		
InContent	STRING	Указывается XML-строка, которую необходимо прочитать / модифицировать / сохранить.
Load	BOOL	Загружает содержимое строки во внутреннее представление по переднему фронту
Save	BOOL	Сохраняет по переднему фронту содержимое внутреннего представления в виде строки
XPath	STRING	Указывается XPath-запрос для определения элемента или атрибута, с которым происходит работа. Также может быть выборка группы атрибутов, в этом случае возвращается массив значений.

Название	Тип	Назначение
ReturnType	BaseObjects.XMLReturnType	<p>Указывается тип возвращаемого на выходе OutValue значения:</p> <ul style="list-style-type: none"> • String • Array • Struct • ArrayOfStruct • Boolean • Integer • Double
GetValue	BOOL	Начинает по переднему фронту чтение значения, заданное на входе XPath
SetValue	BOOL	Устанавливает по переднему фронту значение указанное на входе InValue на элемент/атрибут указанный на входе XPath
InValue	ANY	Задается значение, которое необходимо установить элементу/атрибуту;
AddElement	BOOL	Добавляет по переднему фронту дочерний элемент (к элементу, что задан в XPath) значение элемента задается на входе InValue как дочерний в элемент XPath;
AddElementName	STRING	имя добавляемого элемента.
Выходы		
OutContent	STRING	Выдается результирующая строка, полученная после изменения значения с False на TRUE входа Save
OutValue	ANY	Выдается результат запроса, полученного после изменения значения с False на TRUE входа GetValue;

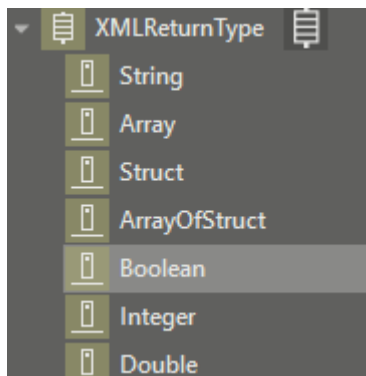
Название	Тип	Назначение
ErrorCode	DINT	Выдается код ошибки;
ErrorText	STRING	Выдается текст ошибки.

Коды и тексты ошибок:

- 0 - "Успешно",
- 1 - "Файл не найден",
- 2 - "Ошибка чтения",
- 3 - "Ошибка выделения памяти",
- 4 - "Внутренняя ошибка",
- 5 - "Синтаксический анализатор не смог определить тип тега",
- 6 - "Ошибка при разборе XML-декларации",
- 7 - "Ошибка при разборе комментария",
- 8 - "Ошибка при разборе CDATA секции",
- 9 - "Ошибка при разборе декларации типа документа",
- 10 - "Ошибка при разборе PCDATA секции",
- 11 - "Ошибка при разборе открывающего тега элемента",
- 12 - "Ошибка при разборе атрибута элемента",
- 13 - "Ошибка при разборе закрывающего тега элемента",
- 14 - "Несовпадение открывающего и закрывающего тега элемента",
- 15 - "Невозможно присоединить узел, так как корень не является элементом или корнем документа",
- 16 - "Синтаксический анализ произведен в документе, не имеющем элементов",
- 17 - "Элемент не найден",
- 18 - "Ошибка XPath-запроса".

XMLReturnTypes

Специальный тип данных, предназначенный для работы данного функционального блока. По сути своей представляет Перечисление со следующими значениями:



В случае если выбор типа производит оператор вручную, то для этого можно использовать элемент Выпадающий список. Параметр, получающий данные от выпадающего списка и передающий их на вход функционального блока (или программы, в которой он находится) должен иметь также тип XMLReturn type.

Рекомендации по использованию

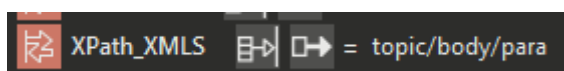
Как правило ФБ XMLString используется в комплекте с функциональными блоками FileToString и StringToFile.

Изменение существующего элемента

1. На вход InContent была подана строка:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="../helpproject.xsl" ?>
<topic template="Default" status="" lasteditedby="MariaM" xmlns:xsi="http://www.w3.org
  <title>New topic</title>
  <body>
    <header><para styleclass="Heading1">New topic</para></header>
    <para styleclass="Normal">Enter topic text here.</para>
  </body>
</topic>
```

2. Содержимое будет передано во внутренний обработчик функционального блока, после изменения значения входа Load.
3. Теперь можно изменить содержимое загруженной строки. Изменим текст Enter topic text here. Для этого сформируем XPath-запрос. Этот текст находится в элементе para, поэтому запрос будет иметь вид: topic/body/para:



4. На вход InValue установим значение, например, Справочная система MasterSCADA 4D.
5. После изменения значения входа SetValue с False на TRUE элемент будет изменен внутри функционального блока.

- Чтобы увидеть его на выходе OutContent результат, необходимо изменить значение входа Save с False на TRUE. Если вывести в окно значение выхода OutContent, то в клиенте визуализации получим:

OutContent	<pre>instance" xsi:noNamespaceSchemaLocation=" ../helpproject. <title>New topic</title> <body> <header> <para styleclass="Heading1">New topic</para> </header> <para styleclass="Normal">Справочная система MasterSCADA 4D.</para> </body> </topic></pre>
------------	---

Изменение существующего атрибута

Продолжаем работать со строкой, представленной в предыдущем разделе. Изменим значение атрибута styleclass элемента para.

- Сформируем новый XPath-запрос topic/body/para/@styleclass. Видно, что к атрибуту элемента необходимо добавлять символ @.
- На вход InValue установим значение, например, Heading2.
- После изменения значения входа SetValue с False на TRUE элемент будет изменен внутри функционального блока.
- Чтобы увидеть его на выходе OutContent результат, необходимо изменить значение входа Save с False на TRUE. Если вывести в окно значение выхода OutContent, то в клиенте визуализации получим:

OutContent	<pre>instance" xsi:noNamespaceSchemaLocation=" ../helpproject. <title>New topic</title> <body> <header> <para styleclass="Heading1">New topic</para> </header> <para styleclass="Heading2">Справочная систем MasterSCADA 4D.</para> </body> </topic></pre>
------------	--

Добавление нового элемента

Продолжаем работать со строкой, представленной в предыдущем разделе. В элемент body добавим новый элемент para1

1. Сформируем новый XPath-запрос topic/body
2. На входе AddElementName укажем имя нового элемента para1
3. На входе InValue установим значение, которое хотим присвоить новому элементу. Например, MasterSCADA 4D - любимая SCADA-система.
4. После изменения значения входа AddElement с False на TRUE элемент будет добавлен внутри функционального блока.
5. После изменения значения входа SetValue с False на TRUE элемент будет изменен внутри функционального блока.
6. Чтобы увидеть его на выходе OutContent результат, необходимо изменить значение входа Save с False на TRUE. Если вывести в окно значение выхода OutContent, то в клиенте визуализации получим:

6.4.3.1.27.PING

Функциональный блок Ping служит для проверки соединения с указанным устройством. Находится в библиотеке BaseObjects. Этот функциональный блок может использоваться как самостоятельно, вне программ, так и в составе любой программы.

Вид в дереве	Вид в редакторе FBD

Входы и выходы ФБ:

Название	Тип	Назначение
Входы		
Address	STRING	Указывается IP-адрес или DNS имя сервера
Check	BOOL	Запускает проверку. По переднему фронту начинается проверка соединения - исполнительная система отправляет серию пакетов на указанный адрес, количество пакетов определяется входом RepeatCount. Важно! Если прошлая проверка, не окончена, то передний фронт входа будет игнорироваться
TimeOut	TIME	Задается время ожидания ответа от сервера
RepeatCount	DINT	Указывается количество попыток отправки команды подряд, если все они окажутся не успешны, то выход Connection примет значение false.
Выходы		
Completed	BOOL	Сигнализирует об окончании проверки. Выставляется значение TRUE на один цикл после завершения проверки.
Connection	BOOL	Выдается результат проверки. Если соединение было установлено, то появится значение TRUE, иначе false.
SuccessCount	DINT	Выдается количество успешных ответов
AverageTime	LREAL	Выдается среднее время ответа, мс
ErrorText	STRING	Выдается текст ошибки. Например, Достигнут таймаут запроса.

6.4.3.1.28.GETSYSTEMINFO

Функциональный блок GetSystemInfo служит для получения информации об исполнительной системе, в которой работает проект. Находится в библиотеке BaseObjects.

Этот функциональный блок может использоваться как самостоятельно, вне программ, так и в составе любой программы.



Выходы ФБ:

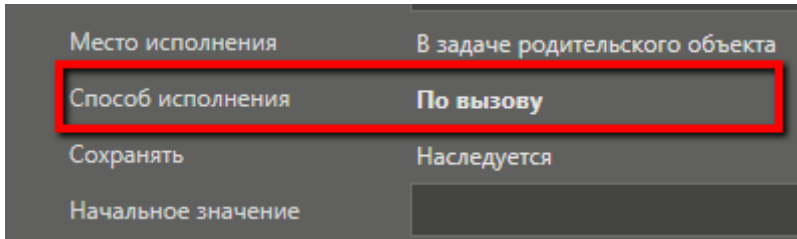
Название	Тип	Назначение
RTVersion	STRING	Версия исполнительной системы
RTVersionDate	STRING	Дата сборки исполнительной системы
SerialId	STRING	Серийный номер исполнительной системы
PlatformName	STRING	Имя платформы исполнительной системы, например, windows, linux, plc110 и др.
PlatformId	DINT	Числовой код используемой платформы.

Название	Тип	Назначение
ProjectName	STRING	Имя проекта
NodeName	STRING	Имя узла, в котором работает данный функциональный блок
ProjectId	STRING	ID проекта. GUID, который формируется при компиляции проекта.
ProjectVersionId	STRING	ID версии проекта. См. История версий
ProjectVersionDate	STRING	Время последнего изменения проекта
IDEVersion	STRING	Версия среды разработки, в которой создавался проект
RTCCommandLine	STRING	Параметры запуска исполнительной системы.
MemorySize	LINT	Размер памяти процесса исполнительной системы
DemoVersion	BOOL	Показывает работает ли демоверсия исполнительной системы для Windows или версия неактивирована. Если значение TRUE, то через час отключится опрос протоколов.
Flags	DWORD	<p>Выдаются битовые флаги состояния:</p> <ul style="list-style-type: none"> • Restarting 0x0001 - начата перезагрузка системы • InReserv 0x0002 - система в режиме SLAVE • Failure 0x0004 - Установлен признак «Отказ» <ul style="list-style-type: none"> • HasError 0x0008 - Установлен признак «Ошибка» • ProgramFailure 0x0010 - Установлен признак "Програмный отказ" • ManualProgramFailure 0x0020 - Установлен признак "Ручное управление программным отказом"

Название	Тип	Назначение
		<ul style="list-style-type: none"> • ExternalModuleFault 0x0200 - отказ протоколов • ConnectionWithAnotherControllerFault 0x0400 - ошибка синхронизации с узлом MASTER • ConnectionWithUpperLevelFault 0x1000 - отказ связи с верхним уровнем (в данный момент не выставляется) • IgnoreFailure 0x00040000 - установлен признак игнорирования отказа • DemoVersion 0x00080000 - демо версия windows или неактивированная версия • NeedActivation 0x00100000 - данную систему необходимо активировать • ImitationMode 0x00400000 - система в режиме имитации (без доступа к портам и работы драйверов) • DemoMode 0x00800000 - система перешла в демо режим через час после старта (без доступа к портам и работы драйверов) <ul style="list-style-type: none"> • 0x00010000 - игнорировать, всегда установлено • 0x00020000 - игнорировать, всегда установлено • PLCUniversalBuild 0x01000000 - Возможно обновление версии через среду разработки • HasNewErrors 0x02000000 //Появились новые ошибки программ драйверов)

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



6.4.3.1.29. REDUNDANCYCONTROL

Функциональный блок RedundancyControl служит для управления резервированными узлами.

Важно! Функциональные блоки в каждом узле работают независимо и показывают/управляют тем узлом, в котором выполняется. Находится в библиотеке BaseObjects. Этот функциональный блок может использоваться как самостоятельно, вне программ, так и в составе любой программы.

Вид в дереве	Вид в редакторе FBD

Входы и выходы ФБ:

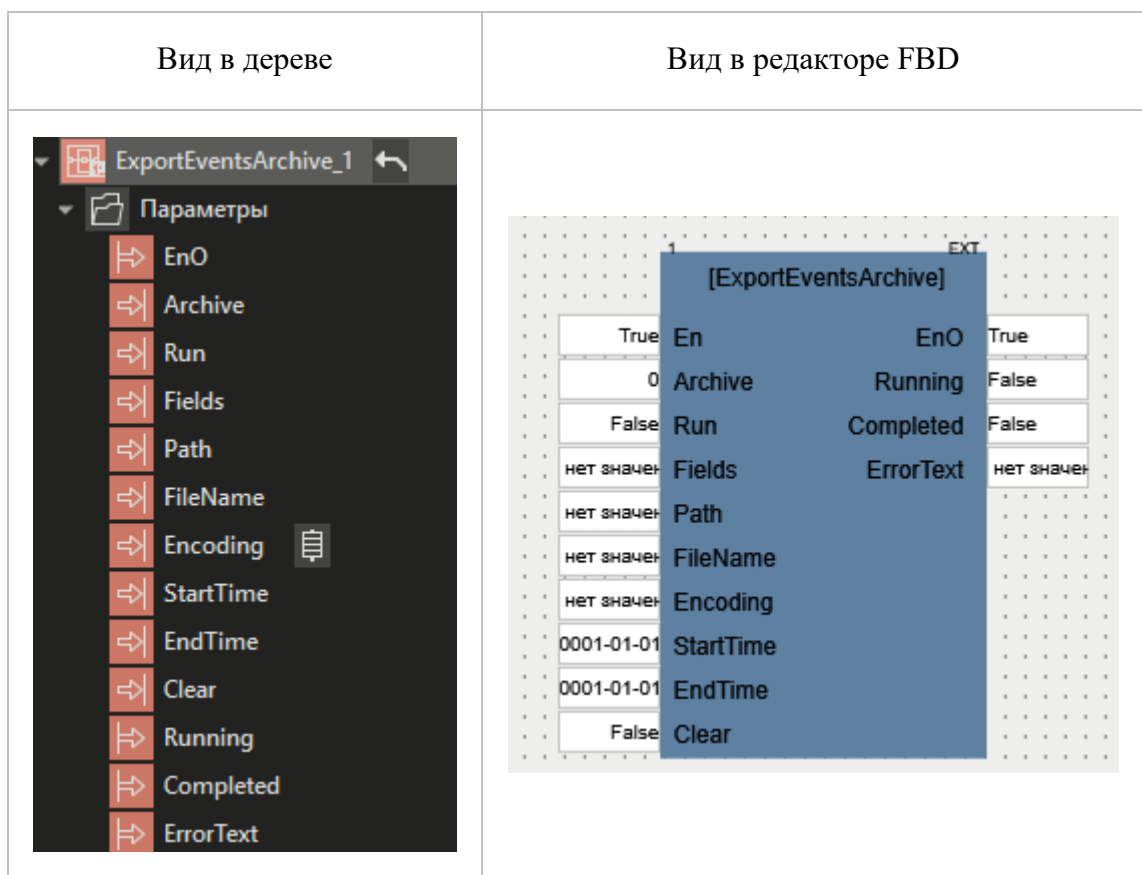
Название	Тип	Назначение
Входы		
SetProgramFault	BOOL	Устанавливает программный статус отказа узла.
ResetProgramFault	BOOL	Сбрасывает программный статус отказа узла.
SetIgnoreFault	BOOL	Устанавливает признак игнорирования отказа узла.
ResetIgnoreFault	BOOL	Сбрасывает признак игнорирования отказа узла.
ChangeToSlave	BOOL	<p>Временно переводит узел в режим SLAVE (другой при этом станет мастером)</p> <p>Важно! Узел может стать основным (Master) только, если он переключился в режим резервного (Slave) как минимум 15 сек назад. Это ограничение позволяет избежать закливания при переключениях.</p>
Выходы		
IsRedundancyEnabled	BOOL	Показывает активировано ли резервирование в работающем проекте. Если значение выхода False, значит в проекте функции резервирования не используются для данного узла.
IsMaster	BOOL	Показывает состояние узла, на котором исполняется. Если значение TRUE, то узел находится в режиме Master, если значение false, то Slave. Этот выход можно использовать в программах, например, которые отвечают за отправкой Email сообщений. Т.е. если узел находится в состоянии Slave, то отправку сообщений не производить
Number	DINT	Показывает, каким назначено данное устройство в среде разработки.

Название	Тип	Назначение
		<ul style="list-style-type: none"> • 1 - в среде разработки устройство было назначено как Узел 1, его IP-адрес задан в поле IP адрес категории свойств Связь в настройках узла • 2 - в среде разработки устройство было назначено как Узел 2, IP-адрес задан в поле IP адрес узла 2 категории свойств Резервирование <p>Если значение данного выхода отобразить в окне, то в окне будет отображаться значение того узла, который находится в состоянии Master, т.к. клиент подключается автоматически к тому узлу, который в текущий момент</p>
IsFault	BOOL	Сообщает о наличии отказа узла. Узел имеет статус отказа, если значение выхода TRUE.
IsError	BOOL	Сообщает о наличии ошибки узла. Узел имеет ошибки, если значение выхода TRUE.
IsProgramFault	BOOL	Сообщает о наличии программного отказа узла. Отказ вызван программно, если значение выхода TRUE.
IsProgramError	BOOL	Сообщает о наличии программной ошибки узла. Узел имеет статус ошибки, вызванный программно, если значение выхода TRUE.
IsIgnoreFault	BOOL	Показывает, что узел имеет статус игнорирования отказа
StateFlags	DWORD	<p>Отображается флаг состояния узла:</p> <ul style="list-style-type: none"> • 0x0200 - отказ одного из протоколов • 0x0400 - ошибка синхронизации Slave с Master • 0x2000 - включен режим работы без опроса протоколов (ставится через час работы демоверсии исполнительной системы)
IsOtherConnected	BOOL	Отображает наличие связи с другим узлом. Если значение выхода TRUE - связь есть.

Название	Тип	Назначение
IsOtherFault	BOOL	Отображает наличие отказа у другого узла. Если значение выхода TRUE, то другой узел имеет статус Отказ.
IsOtherError	BOOL	Отображает наличие ошибки у другого узла. Если значение выхода TRUE, то другой узел имеет статус Ошибка.
StatSyncPeriod	DINT	Отображает период обновления данных резервного узла.

6.4.3.1.30.EXPORTEVENTSARCHIVE

Функциональный блок ExportEventsArchive служит для экспорта архива сообщений в csv-файл. Находится в библиотеке BaseObjects. Этот функциональный блок может использоваться как самостоятельно, вне программ, так и в составе любой программы. Данный ФБ работает только в задачах узла.



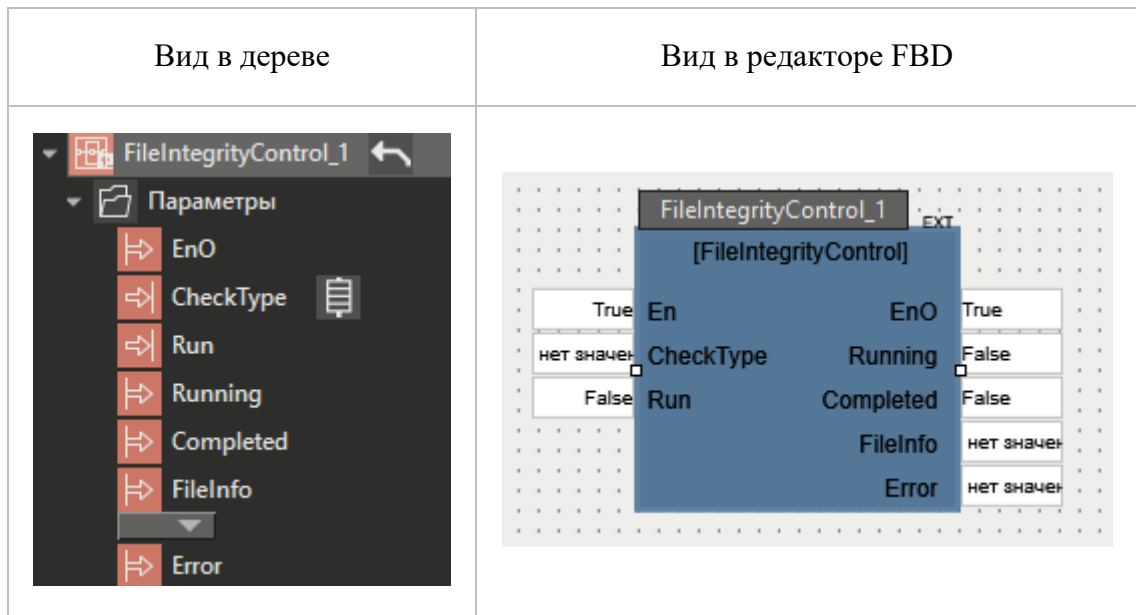
Входы и выходы ФБ:

Название	Тип	Назначение
Входы		
Archive	REF_TO	Указывается связь с архивом. Можно перетащить Основной архив сообщений или Архив сообщений ИБ на вход левой кнопкой мыши, либо указать ID нужного архива в данном свойстве.
Run	BOOL	Запускает процесс сохранения файла по переднему фронту. Если файл с таким же именем в той же папке уже существовал ранее, то файл перезапишется.
Fields	STRING	Перечисляются список полей, которые необходимо сохранить. Значения полей (параметры тревоги) вводятся через запятую. Указываются программные имена полей (параметров). Список полей и их программные имена смотрите в разделах Тревога и Список программных имен параметров тревог
Path	STRING	Указывается путь к папке, в которую сохранится файл. Можно указывать как локальную так и сетевую папку. Если значение входа не указано, то файл создастся рядом с файлом EventsData.db
FileName	STRING	Задается имя сохраняемого файла. Если вход не задан, то имя файла сформируется автоматически имя будет состоять из даты и времени сохранения. Для указания имени можно использовать символы кириллицы и латиницы.
Encoding	Encoding	<p>Определяет кодировку файла. Возможные варианты:</p> <ul style="list-style-type: none"> • UTF8 (по умолчанию), • CP1251, • ASCII
StartTime	DT	Указывается время, с которого необходимо начать экспорт сообщений. Если указано значение 0, то экспорт производится с начала архива.

Название	Тип	Назначение
EndTime	DT	Указывается время, до которого необходимо выполнить экспорт сообщений. Если указано значение 0, то экспорт производится до конца архива.
Clear	BOOL	Определяет необходимость очистки архива после экспорта. Если вход принимает значение TRUE, то после экспорта происходит удаление архива целиком, если вход EndTime не задан, если вход используется, удаляются записи с начала и до указанного времени. Во время экспорта не выполняется запись в архив, новые значения остаются в очереди записи, поэтому после очистки архива неэкспортированные записи поступят в архив и не будут потеряны при удалении архива.
Выходы		
Running	BOOL	Информирует, что идет формирование файла экспорта. Этот выход принимает значение TRUE на время сохранения файла.
Completed	BOOL	Сигнализирует об окончании формирования файла экспорта. Выставляется значение TRUE на один цикл после завершения.
ErrorText	STRING	Указывается текст ошибки, если не удалось выполнить сохранение.

6.4.3.1.31.FILEINTEGRITYCONTROL

Функциональный блок FileIntegrityControl служит для управления контролем целостности ПО или проекта и получения списка отличий. Находится в библиотеке BaseObjects. Этот функциональный блок может использоваться как самостоятельно, вне программ, так и в составе любой программы. Данный ФБ работает только в задачах узла. Для работы функционального блока в настройках безопасности должен быть установлен флаг Контроль целостности проекта. Контроль целостности исполнительной системы необходимо выполнять на версиях, которые устанавливаются независимо от среды разработки (коммерческие версии, демоверсии, которые имеют отдельный инсталлятор, а не входят в состав среды разработки).

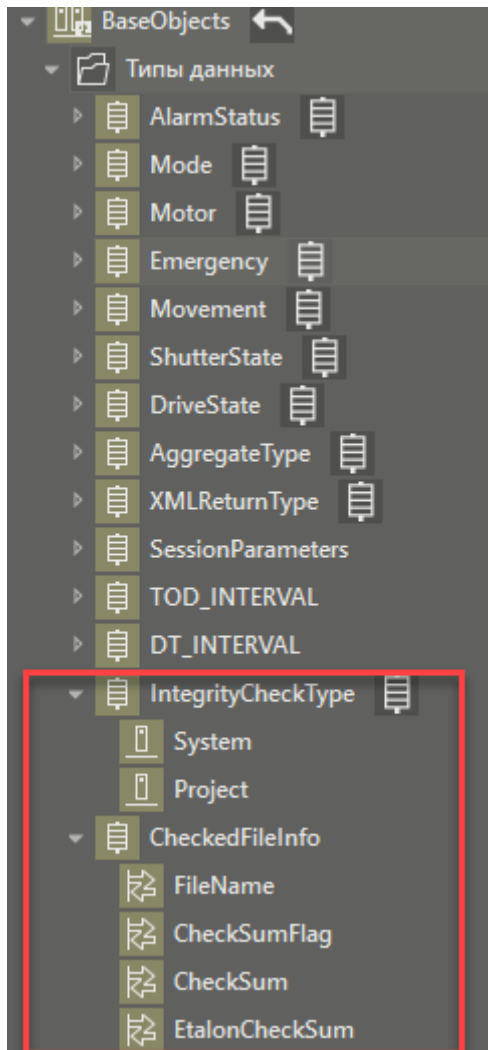


Входы и выходы ФБ:

Название	Тип	Назначение
Входы		
CheckType	BaseObjects.IntegrityCheckType	Указывается какая проверка будет выполнена: контроль целостности ПО или проекта. Вход может принимать два значения: <ul style="list-style-type: none"> • Project - выполняется проверка целостности проекта • System - выполняется проверка целостности исполнительной системы.
Run	BOOL	Запускает процесс проверки по переднему фронту.
Выходы		
Running	BOOL	Информирует, что идет проверка. Этот выход принимает значение TRUE на проверки.
Completed	BOOL	Сигнализирует об окончании проверки. Выставляется значение TRUE на один цикл после завершения проверки.

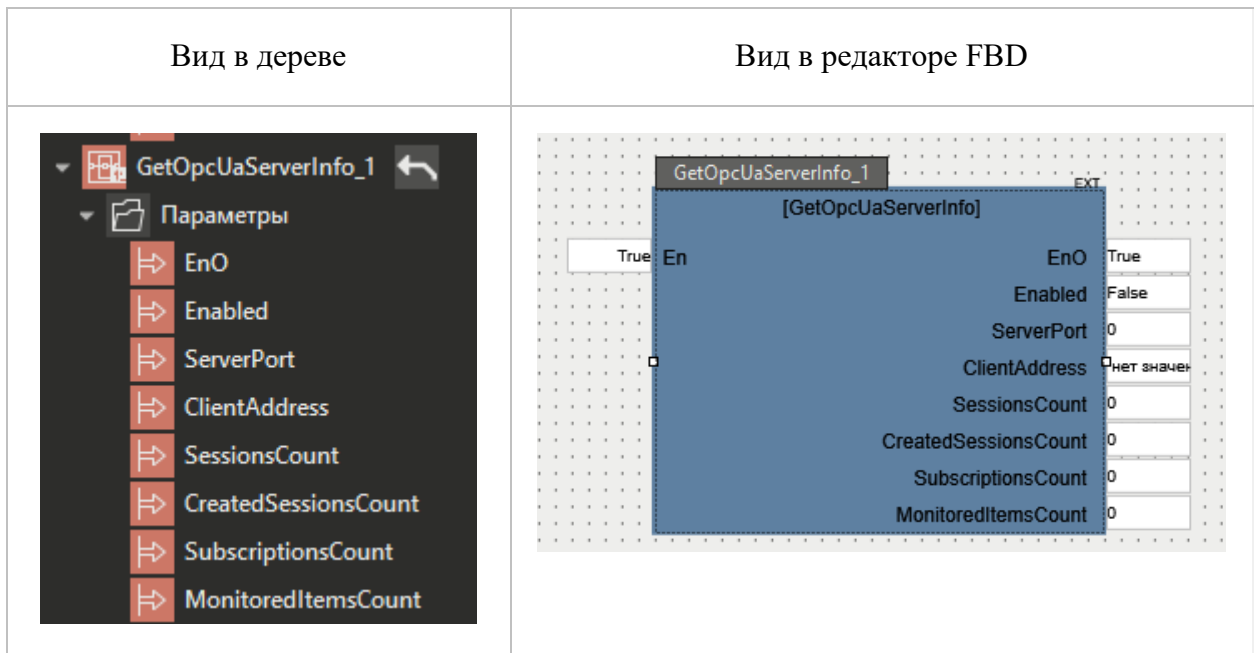
Название	Тип	Назначение
FileInfo	ARRAY [*] OF CheckedFileInfo	<p>Выдается описание по каждому проверяемому файлу. Выход представляет собой массив структур. Каждый элемент массива соответствует проверяемому файлу. Каждая структура содержит следующие поля:</p> <ul style="list-style-type: none">• FileName - имя проверяемого файла, STRING• CheckSumFlag - признак соответствия контрольной сумме (BOOL). Если TRUE, то соответствует• CheckSum - полученная контрольная сумма, STRING• EtalonCheckSum - эталонная контрольная сумма, STRING
Error	STRING	Указывается текст ошибки, если не удалось выполнить проверку.

Вспомогательные типы данных находятся в библиотеке BaseObjects. Типы данных:



6.4.3.1.32.GETOPCUASERVERINFO

Функциональный блок GetOpcUaServerInfo служит для получения информации и статистики по работе OPC UA сервера (когда MasterSCADA 4D выступает в роли сервера). Находится в библиотеке BaseObjects. Этот функциональный блок может использоваться как самостоятельно, вне программ, так и в составе любой программы. Данный ФБ работает только в задачах узла.

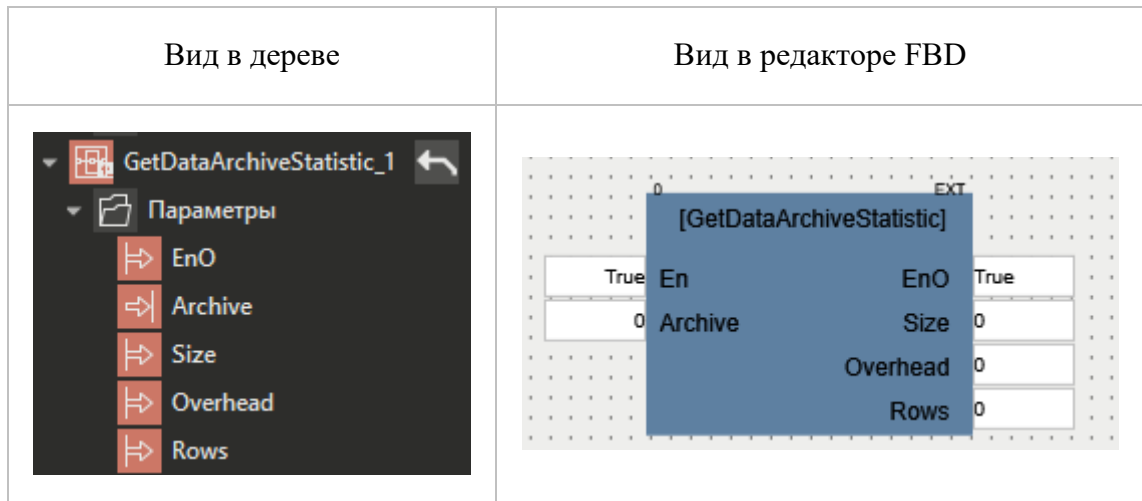


Входы и выходы ФБ:

Название	Тип	Назначение
Выходы		
Enabled	BOOL	Сервер включен
ServerPort	UDINT	Выдается порт сервера
ClientAddress	STRING	Указывается адрес клиента с портом для режима ReverseHello
SessionsCount	UDINT	Указывается количество подключенных сессий клиентов
CreatedSessionsCount	UDINT	Выдается общее количество созданных сессий клиентов
SubscriptionsCount	UDINT	Указывается количество подключенных подписок клиентов
MonitoredItemsCount	UDINT	Указывается количество параметров во всех подписках клиентов

6.4.3.1.33.GETDATAARCHIVESTATISTIC

Функциональный блок GetDataArchiveStatistic служит для получения информации о работе архива данных. Находится в библиотеке BaseObjects. Этот функциональный блок может использоваться как самостоятельно, вне программ, так и в составе любой программы. Данный ФБ работает только в задачах узла.

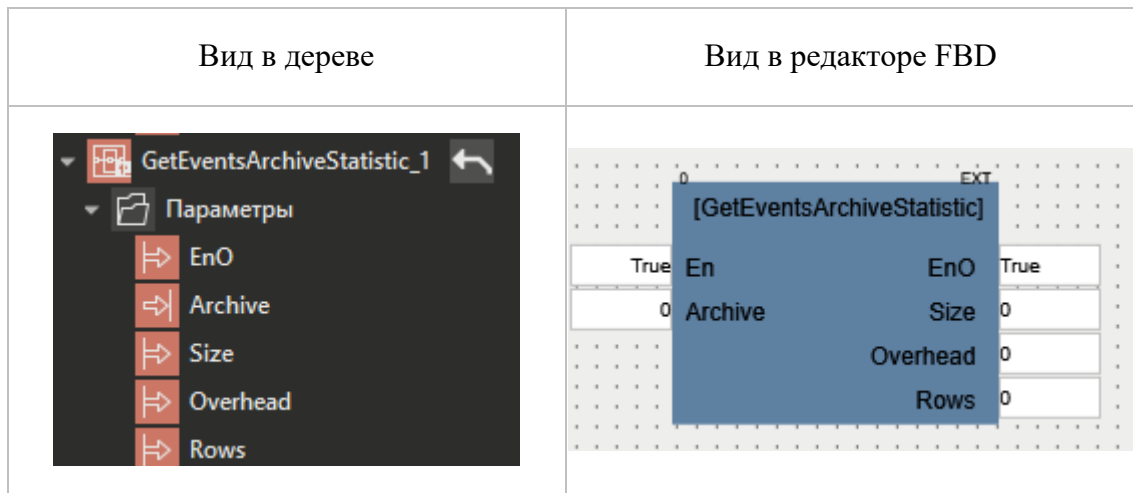


Входы и выходы ФБ:

Название	Тип	Назначение
Входы		
Archive	REF_TO	Указывается связь с архивом.
Выходы		
Size	LINT	Выдается размер таблицы
Overhead	LREAL	Отношение текущего размера к максимальному, заданному в настройках, %
Rows	DINT	Число строк в таблице (data_raw)

6.4.3.1.34.GETEVENTSARCHIVESTATISTIC

Функциональный блок GetEventsArchiveStatistic служит для получения информации о работе архива данных. Находится в библиотеке BaseObjects. Этот функциональный блок может использоваться как самостоятельно, вне программ, так и в составе любой программы. Данный ФБ работает только в задачах узла.

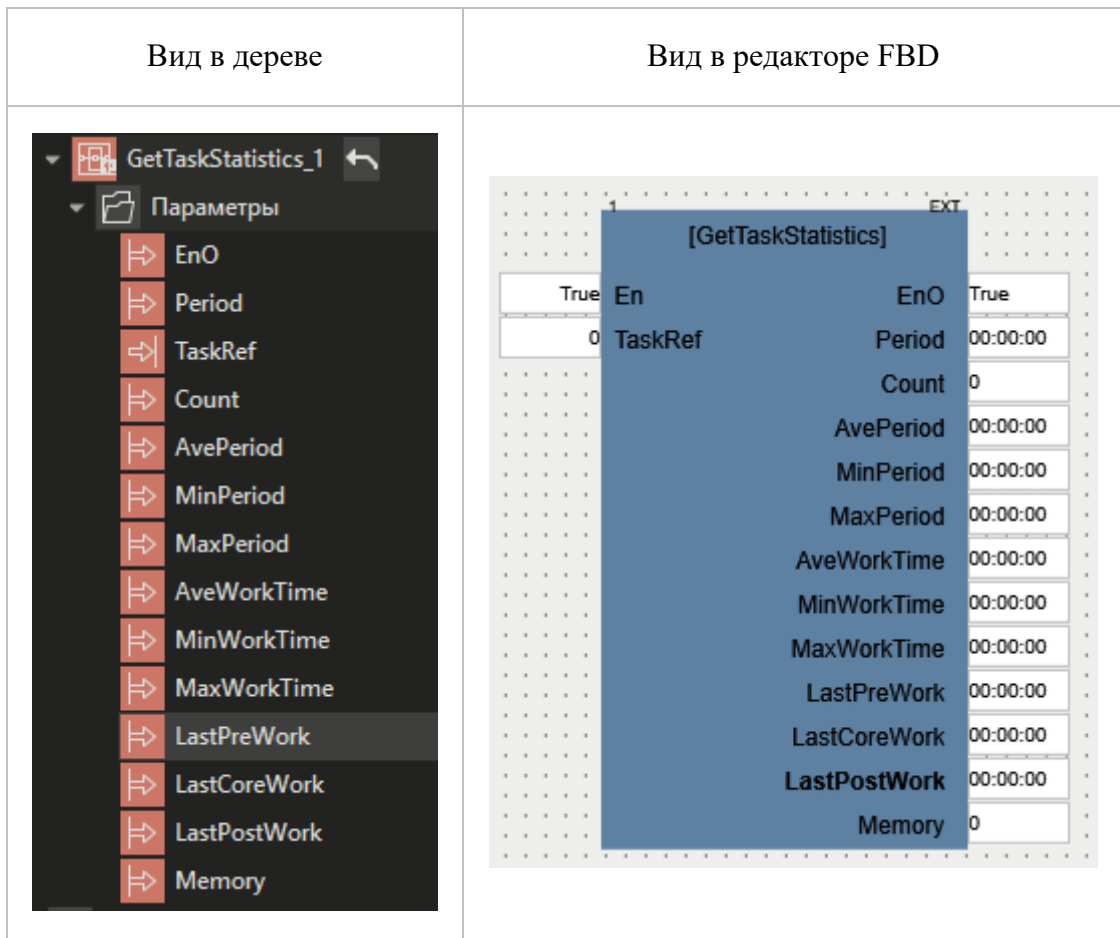


Входы и выходы ФБ:

Название	Тип	Назначение
Входы		
Archive	REF_TO	Указывается связь с архивом.
Выходы		
Size	LINT	Выдается размер таблицы
Overhead	LREAL	Отношение текущего размера к максимальному, заданному в настройках, %
Rows	DINT	Число строк в таблице (events_recs)

6.4.3.1.35.GETTASKSTATISTICS

Функциональный блок GetTaskStatistics служит для получения статистики задач исполнительной системы. Находится в библиотеке BaseObjects. Этот функциональный блок может использоваться как самостоятельно, вне программ, так и в составе любой программы.



Входы и выходы ФБ:

Название	Тип	Назначение
Входы		
TaskRef	REF_TO	Нужно на вход перетащить задачу или протокол, статистику по которой нужно получить или указать ID задачи или протокола.
Выходы		
Period	TIME	Выдается заданный период
Count	LINT	Количество циклов
AvePeriod	TIME	Среднее время цикла

Название	Тип	Назначение
MinPeriod	TIME	Минимальное время цикла
MaxPeriod	TIME	Максимальное время цикла
AveWorkTime	TIME	Среднее время реального выполнения
MinWorkTime	TIME	Минимальное время реального выполнения
MaxWorkTime	TIME	Максимальное время реального выполнения
LastPreWork	TIME	Время чтения внешних данных на последнем цикле
LastCoreWork	TIME	Время выполнения задачи на последнем цикле
LastPostWork	TIME	Время отдачи внешних данных на последнем цикле
Memory	LINT	Объем памяти виртуальной машины

6.4.3.2. BASEOBJECTS.ФБ.USERS

Данная группа содержит функциональные блоки, предназначенные для работы с пользователями проекта в режиме исполнения.

В результате работы функционального блока, если у права доступа Изменение пользователей установлен флаг Журнал, то сформируется сообщение, содержащее информацию как о новом значении, так и о предыдущем. Тип сообщения будет Действие пользователей

Список функциональных блоков:

Название и ссылка	Назначение
UsersGet	Служит для получения списка пользователей, добавленных в режиме исполнения
UsersAdd	Служит для добавления пользователей в режиме исполнения

Название и ссылка	Назначение
UsersSetGroups	Служит для назначения/изменения списка ролей пользователя, ранее созданного в режиме исполнения
UsersDelete	Служит для удаления пользователя, ранее добавленного в режиме исполнения.
UsersGetGroups	Служит для получения списка ролей, в которые добавлен определенный пользователь
UsersSetPassword	Служит для назначения/изменения пароля пользователя, ранее добавленного в режиме исполнения
UsersRename	Служит изменения имени существующего пользователя, который был добавлен в режиме исполнения
UsersGetSettings	Является типом для экземпляров функциональных блоков, расширенных разработчиками проекта, предназначенных для получения значений дополнительных настроек пользователей в режиме исполнения.
UsersSetSettings	Является типом для экземпляров функциональных блоков, расширенных разработчиками проекта, предназначенных для изменения значений дополнительных настроек пользователей в режиме исполнения.
UsersChangeOwnPassword	Позволяет сменить собственный пароль
UsersGetADUserGroups	Служит для получения данных о пользователе добавленном в ActiveDirectory и получения его групп
UsersGetBlocked	Служит для получения информации о состоянии блокировки пользователя.
UsersSetBlocked	Служит для блокировки учетной записи пользователя, при которой пользователь не может запустить новый сеанс работы в

Название и ссылка	Назначение
	клиенте визуализации, но сведения об этом пользователе не удаляются из исполнительной системы.
UsersCloseSession	Служит для удаления действующей сессии пользователя в клиенте визуализации
UsersGetActiveSessions	Служит для получения списка сессий клиентов визуализации, которые активны в текущий момент
UsersGetAllowedAddress	Служит для получения списка адресов, с которых может подключаться пользователь, залогиненный в клиенте визуализации к исполнительной системе
UsersSetAllowedAddress	Служит для определения списка адресов, с которых может подключаться пользователь, залогиненный в клиенте визуализации к исполнительной системе
UsersGetAllowedTime	Служит для получения интервала времени, в течении которого пользователь может иметь залогиниться в клиенте визуализации
UsersSetAllowedTime	Служит для указания интервала времени, в течении которого пользователь может иметь залогиниться в клиенте визуализации
UsersGetNeedResetPassword	Служит для получения информации, потребуется ли пользователю при следующей авторизации сменить свой пароль.
UsersSetNeedResetPassword	Служит для установки условия, чтобы пользователь при следующей авторизации сменил свой пароль
UsersRoleAdd	Служит для добавления ролей пользователей и их настроек в режиме исполнения
UsersRoleDelete	Служит для удаления ролей пользователей и их настроек в режиме исполнения.
UsersGetAllRoles	Служит для получения списка ролей пользователей в режиме исполнения

Название и ссылка	Назначение
UsersGetRoleSettings	Служит для получения настроек роли пользователей в режиме исполнения
UsersSetRoleSettings	Служит для изменения настроек роли пользователей в режиме исполнения
UsersRestoreRoleSettings	Восстанавливает настройки роли, которые были заданы в режиме разработки
UsersRestoreUser	Восстанавливает настройки пользователя, которые были заданы в режиме разработки
UsersSetRoleControlRight	Служит для изменения прав доступа к элементу в режиме исполнения
UsersGetRoleControlRight	Служит для получения назначенных прав доступа для Роли к элементу
UsersDeleteRoleControlRight	Служит для удаления прав доступа какой-либо Роли к элементу в режиме исполнения

6.4.3.2.1. USERSGET

Функциональный блок UsersGet служит для получения списка пользователей/ Если используется для хранения настроек пользователей сетевая БД настроек, то отобразится список доступный для всех узлов проекта. Если сетевая БД настроек не используется, то информация о пользователях хранится в локальном файле, который доступен только для текущего узла.

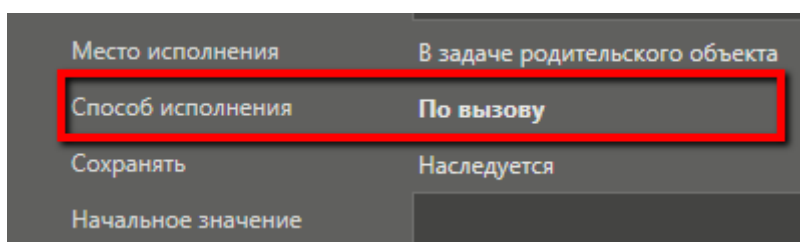
Вид в редакторе FBD	Вид в дереве

Входы и выходы ФБ:

Название	Тип	Назначение
Delimiter	STRING	Вход ФБ. Задается разделитель, который будет использоваться при формировании списка пользователей. В качестве разделителей могут использоваться символы запятая, точка, точка с запятой и другие.
Users	STRING	Выход ФБ, на котором формируется список пользователей, уже добавленных в тот узел, на котором выполняется данный ФБ

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



Как правило, для получения списка пользователей, в окне создают Кнопку. Затем в событие Клик мыши этой кнопки добавляют действие Вызвать программу (для этого функциональный блок следует перетащить на вход действия Программа).

Для удобного отображения списка пользователей можно использовать элемент Выпадающий список. Для этого на вход ФБ Delimiter следует подать символ ~ (тильда), и выход ФБ Users соединить со свойством элемента Список.

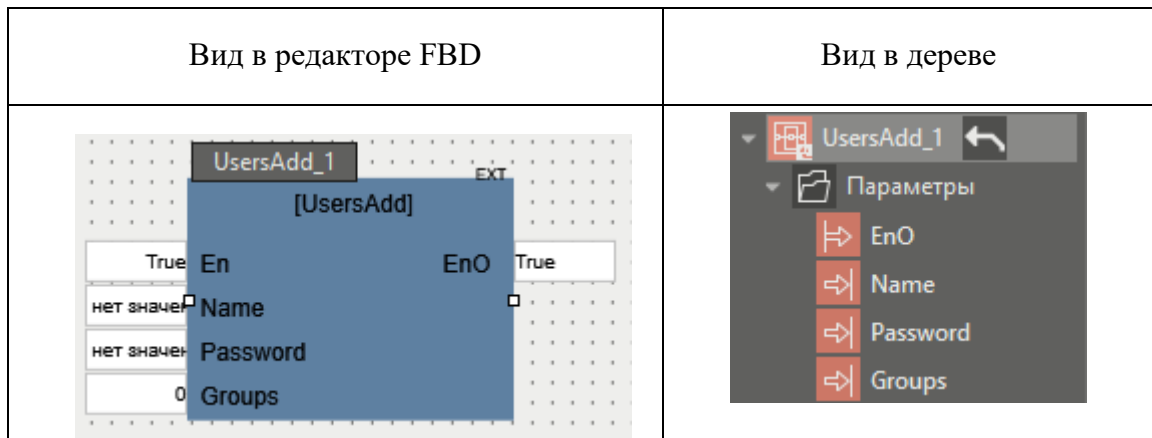
6.4.3.2.2. USERSADD

Функциональный блок UsersAdd служит для добавления пользователей в режиме исполнения. Пользователь добавится в тот узел, на котором выполняется ФБ.

Важно! Для работы данного функционального блока у авторизованного пользователя должно быть настроено право доступа Изменение операторов

В результате работы функционального блока, если у права доступа Изменение пользователей установлен флаг Журнал, то сформируется сообщение, содержащее

информацию как о новом значении, так и о предыдущем. Тип сообщения будет Действие пользователей

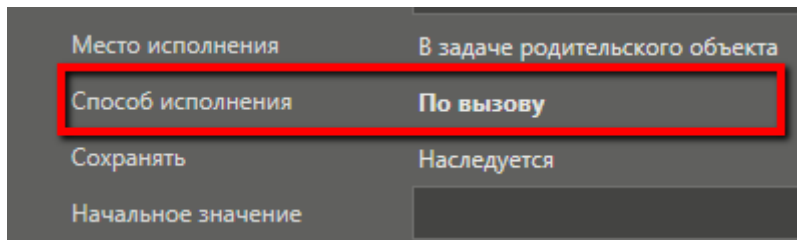


Входы и выходы ФБ:

Название	Тип	Назначение
Name	STRING	Вход ФБ. Задается имя нового пользователя.
Password	STRING	Вход ФБ. Устанавливается пароль пользователя. Для ввода пароля можно использовать графический элемент Пароль, при этом вход ФБ необходимо соединить со свойством элемента Текст.
Groups	STRING	Вход ФБ. Устанавливаются роли, в которые необходимо назначить нового пользователя. Если таких ролей несколько, то следует перечислить их через запятую. Если пользователю не назначить роль, то он добавится, но не будет иметь никаких прав доступа. Таким образом, назначая пользователя на ту или иную роль в режиме исполнения, оператор тем самым определяет ему права. Список доступных ролей, добавленных в проект, можно выяснить, используя ФБ UsersGetGroups.

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



Как правило, новый пользователь создается после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляются действие Вызвать программу путем перетаскивания функционального блока на вход действия Программа.

О том, была ли работа ФБ успешной, можно судить по выходу ФБ EnO. Если добавления пользователя не произойдет, то выход примет значение False.

6.4.3.2.3. USERSSETGROUPS

Функциональный блок UsersSetGroups служит для назначения/изменения списка ролей пользователя.

Важно! Для работы данного функционального блока у авторизованного пользователя должно быть настроено право доступа Изменение пользователей

В результате работы функционального блока, если у права доступа Изменение пользователей установлен флаг Журнал, то сформируется сообщение, содержащее информацию как о новом значении, так и о предыдущем. Тип сообщения будет Действие пользователей

Вид в редакторе FBD	Вид в дереве

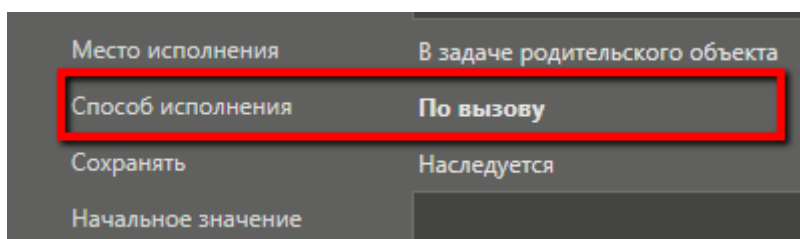
Входы и выходы ФБ:

Название	Тип	Назначение
Name	STRING	Вход ФБ. Задается имя пользователя, роль которого требуется изменить. Список пользователей можно получить, используя ФБ UsersGet.

Название	Тип	Назначение
Groups	STRING	<p>Вход ФБ. Устанавливаются роли, в которые необходимо назначить пользователя. Если таких ролей несколько, то следует перечислить их через запятую. При выполнении этой процедуры пользователь будет удален из всех ролей, на которые был назначен ранее. Т.е. если пользователь, например, был ранее в Роли 1, в Роли 2, а теперь он должен быть еще и в Роли 3, то на вход следует подать значение Роль 1, Роль 2, Роль 3. Назначая пользователя на ту или иную роль в режиме исполнения, оператор, тем самым, определяет ему права. Список доступных ролей, добавленных в проект, можно выяснить, используя ФБ UsersGetGroups.</p>

Рекомендации по работе с ФБ

В панели свойств ФБ, в категории Общие, рекомендуется установить свойство Способ исполнения в состояние По вызову:



Как правило, назначение в группу пользователя происходит после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания функционального блока на вход действия Программа.

О том, была ли работа ФБ успешной, можно судить по выходу ФБ EnO. Если добавления не произойдет, то выход примет значение False

6.4.3.2.4. USERSDELETE

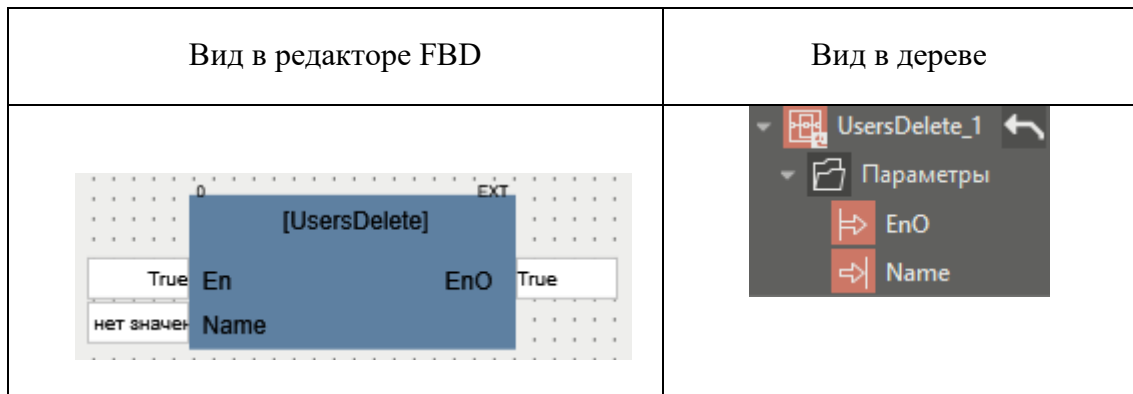
Функциональный блок UsersDelete служит для удаления пользователя, ранее добавленного в режиме исполнения.

Важно! Данный ФБ не работает с пользователями, добавленными в режиме разработки.

Важно! Для работы данного функционального блока у авторизованного пользователя должно быть настроено право доступа Изменение операторов

В результате работы функционального блока, если у права доступа Изменение пользователей установлен флаг Журнал, то сформируется сообщение, содержащее

информацию как о новом значении, так и о предыдущем. Тип сообщения будет Действие пользователей

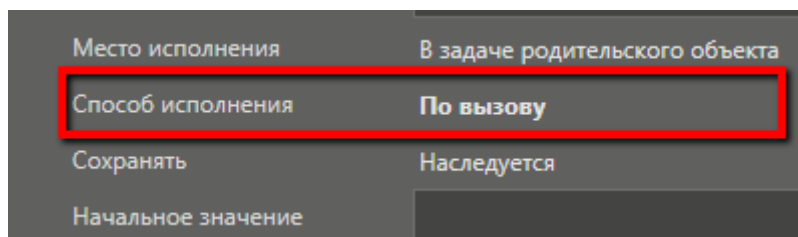


Входы и выходы ФБ:

Название	Тип	Назначение
Name	STRING	Вход ФБ. Задается имя пользователя, которого требуется удалить. Список добавленных пользователей можно получить, используя ФБ UsersGet

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:

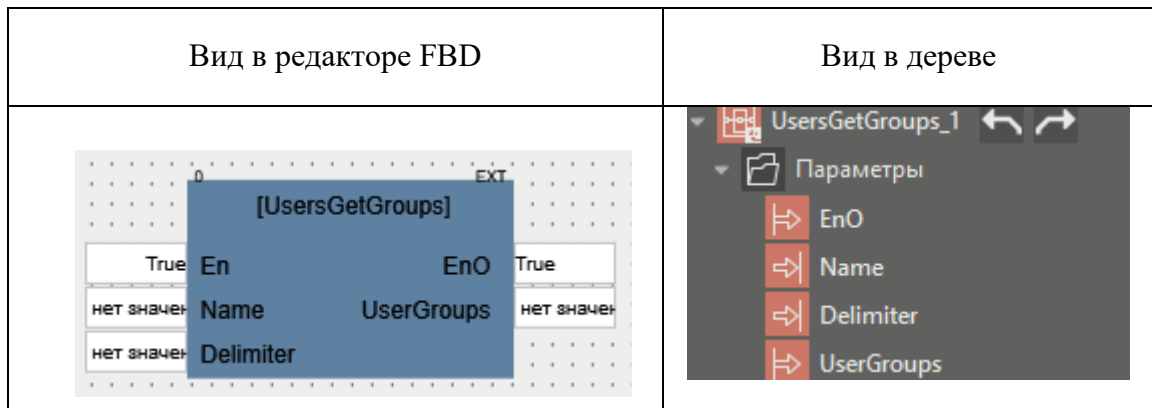


Как правило, удаление пользователя происходит после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания функционального блока на вход действия Программа.

О том, была ли работа ФБ успешной, можно судить по выходу ФБ EnO. Если удаления не произойдет, то этот выход примет значение False.

6.4.3.2.5. USERSGETGROUPS

Функциональный блок UsersGetGroups служит для получения списка ролей, в которые добавлен определенный пользователь.

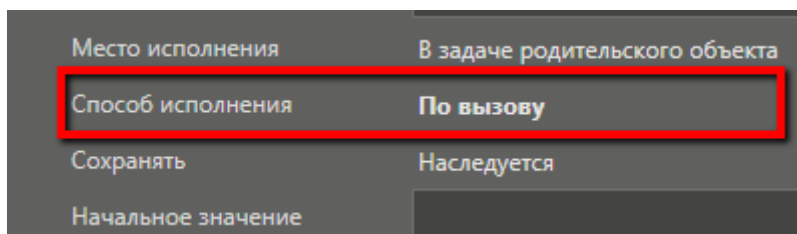


Входы и выходы ФБ:

Название	Тип	Назначение
Name	STRING	Задается имя пользователя, по которому требуется получить информацию о том, на какие роли он был назначен.
Delimiter	STRING	Вход ФБ. Задается разделитель, который будет использоваться при формировании списка ролей. В качестве разделителей могут использоваться символы запятая, точка, точка с запятой, тильда и другие.
UserGroups	STRING	Выход ФБ, на котором формируется список ролей, на которые назначен пользователь, указанный на входе Name.

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



Как правило, для получения списка групп в окне создают Кнопку, затем в ее событие Клик мыши добавляют действие Вызвать программу (для этого функциональный блок следует перетащить на вход действия Программа).

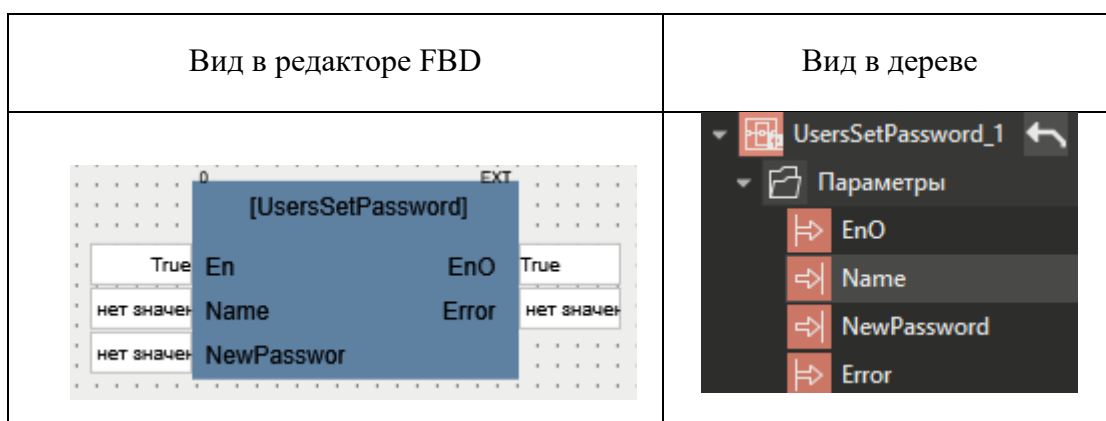
Для удобного отображения списка групп можно использовать элемент Выпадающий список. Для этого на вход ФБ Delimiter следует подать символ ~ (тильда), и выход ФБ UserGetGroups соединить со свойством элемента Список.

6.4.3.2.6. USERSSETPASSWORD

Функциональный блок UsersSetPassword служит для назначения/изменения пароля пользователя, ранее добавленного в режиме исполнения.

Важно! Для работы данного функционального блока у авторизованного пользователя должно быть настроено право доступа Изменение пользователей

В результате работы функционального блока, если у права доступа Изменение пользователей установлен флаг Журнал, то сформируется сообщение вида Пароль пользователя Иванов изменен. Тип сообщения будет Действие пользователей

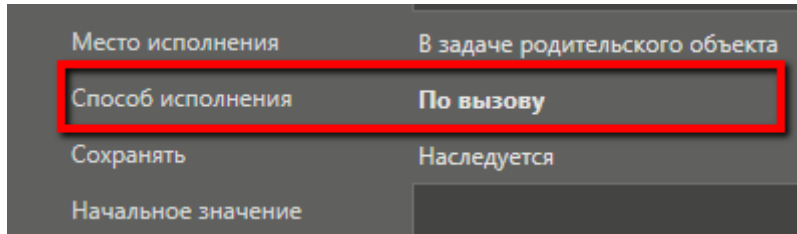


Входы и выходы ФБ:

Название	Тип	Назначение
Name	STRING	Вход ФБ. Задается имя пользователя, пароль которого требуется изменить. Список добавленных пользователей можно получить, используя ФБ UsersGet.
NewPassword	STRING	Вход ФБ. Устанавливается новый пароль пользователя. Для ввода пароля можно использовать графический элемент Пароль, при этом вход ФБ нужно соединить со свойством элемента Текст
Error	STRING	Выход ФБ. Формируется ошибка в случае, если новый пароль не удовлетворяет требованиям политики безопасности паролей, настроенной для роли, которой принадлежит пользователь.

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



Как правило, изменение пароля пользователя происходит после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания этого функционального блока на вход действия Программа.

О том, была ли работа ФБ успешной, можно судить по выходу ФБ EnO. Если добавление не произойдет, то выход примет значение False.

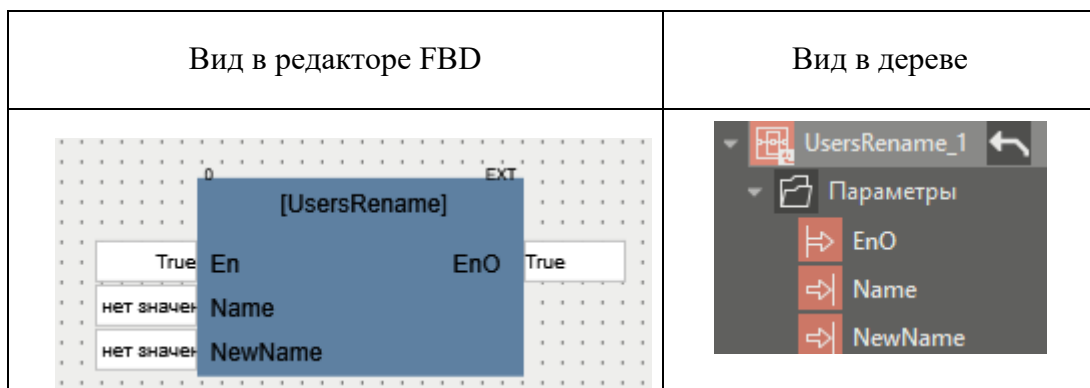
6.4.3.2.7. USERSRENAME

Функциональный блок UsersRename служит изменения имени существующего пользователя, который был добавлен в режиме исполнения. При переименовании пользователя, его назначение на роли и пароль сохраняются.

Важно! Данный ФБ не работает с пользователями, добавленными в режиме разработки.

Важно! Для работы данного функционального блока у авторизованного пользователя должно быть настроено право доступа Изменение пользователей

В результате работы функционального блока, если у права доступа Изменение пользователей установлен флаг Журнал, то сформируется сообщение, содержащее информацию как о новом значении, так и о предыдущем. Тип сообщения будет Действие пользователей

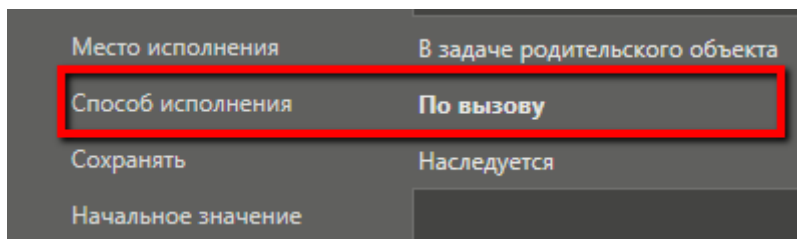


Входы и выходы ФБ:

Название	Тип	Назначение
Name	STRING	Вход ФБ. Задается имя пользователя, которое требуется изменить.
NewName	STRING	Вход ФБ. Задается новое имя пользователя.

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



Как правило, изменение имени происходит в результате нажатия на Кнопку. Для этого в ее событие Клик мыши добавляются действие Вызвать программу путем перетаскивания функционального блока на вход действия Программа.

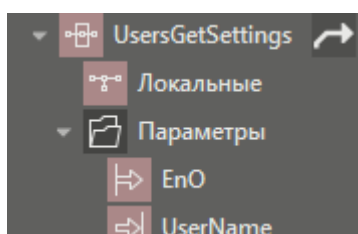
Для того, чтобы не ошибиться с написанием имени пользователя, можно вход Name соединить со свойством Выбор графического элемента Выпадающий список, у которого список строк формируется с использованием ФБ UsersGet.

О том, была ли работа ФБ успешной, можно судить по выходу ФБ EnO. если переименования не произойдет, то выход примет значение False

6.4.3.2.8. USERSGETSETTINGS

Функциональный блок UsersGetSettings является типом для экземпляров функциональных блоков, расширенных разработчиками проекта, предназначенных для получения значений дополнительных настроек пользователей в режиме исполнения.

Вид в дереве библиотек

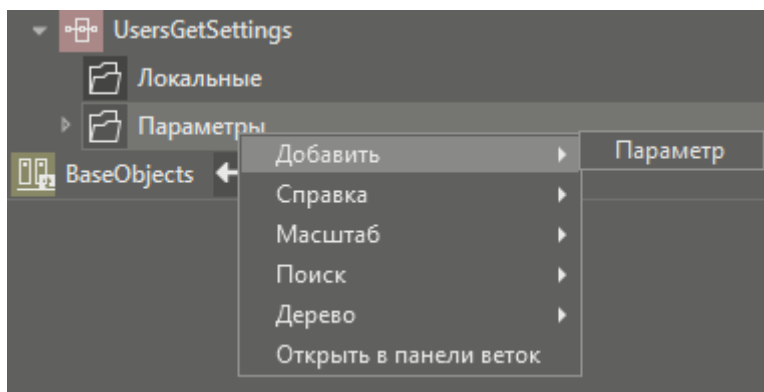


Входы и выходы ФБ:

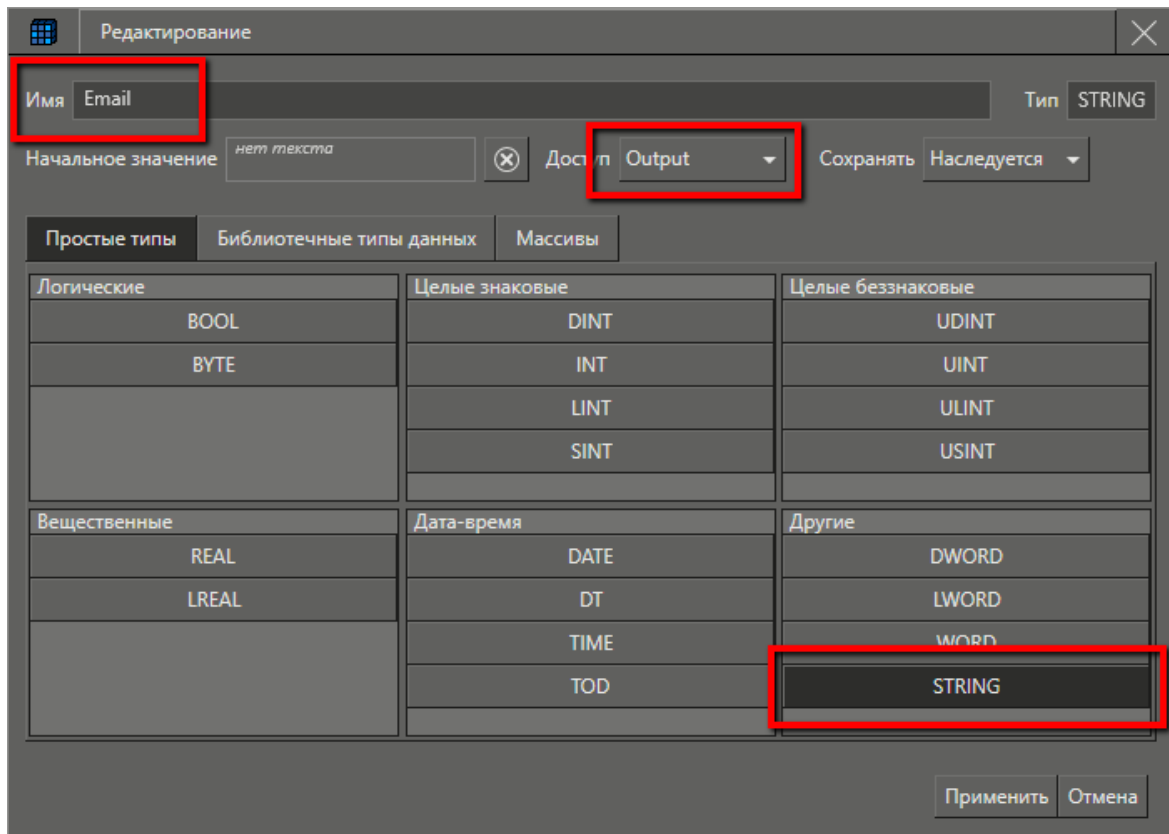
Название	Тип	Назначение
UserName	STRING	Задается имя пользователя, по которому требуется получить значения дополнительных настроек

Рекомендации по работе с ФБ

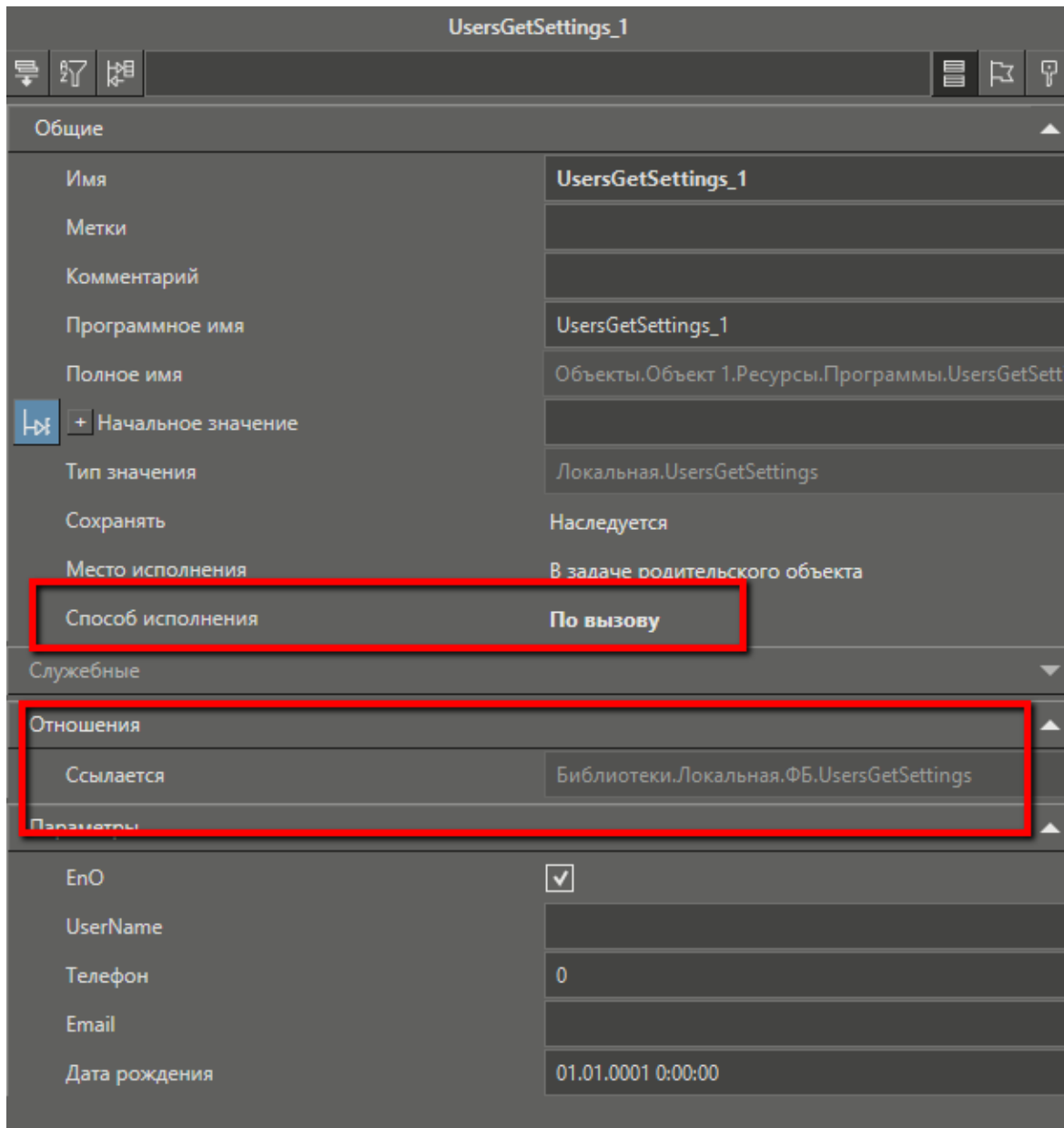
Список дополнительных настроек пользователя зависит от требований проекта, чтобы создать экземпляр функционального блока, который может работать в конкретном проекте необходимо добавить в пользовательскую библиотеку экземпляр функционального блока UsersGetSettings. Затем добавить в него нужное количество выходов.



Имена и типы данных выходов должны совпадать с именами дополнительных настроек пользователя, доступ к которым нужно получить:



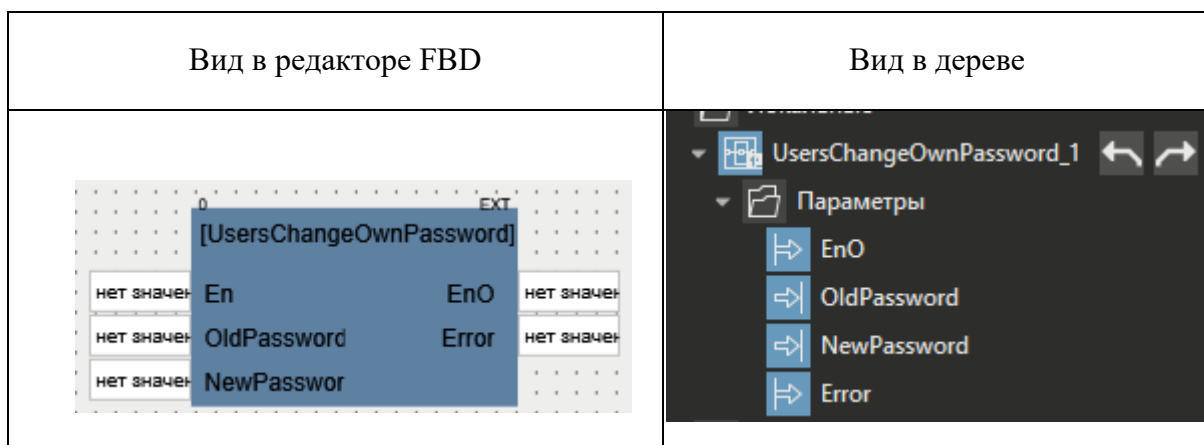
Далее в проекте необходимо использовать созданный функциональный блок. В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



Как правило, для получения значений дополнительных настроек пользователя в окне создают Кнопку, затем в ее событие Клик мыши добавляют действие Вызвать программу (для этого функциональный блок следует перетащить на вход действия Программа).

6.4.3.2.9. USERSCHANGEOWNPASSWORD

Функциональный блок UsersChangeOwnPassword служит для изменения собственного пароля текущего пользователя.



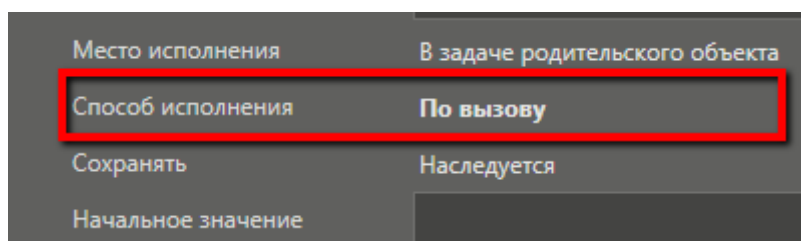
Входы и выходы ФБ:

Название	Тип	Назначение
OldPassword	STRING	Вход ФБ. Задается старый пароль пользователя.
NewPassword	STRING	Вход ФБ. Задается новый пароль пользователя.
Error	STRING	Выход ФБ. Формируется в случае возникновения ошибки, например, если старый пароль указан неверно.

Важно! Для работы ФБ UsersChangeOwnPassword у пользователя должно быть настроено право доступа Смена собственного пароля

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:

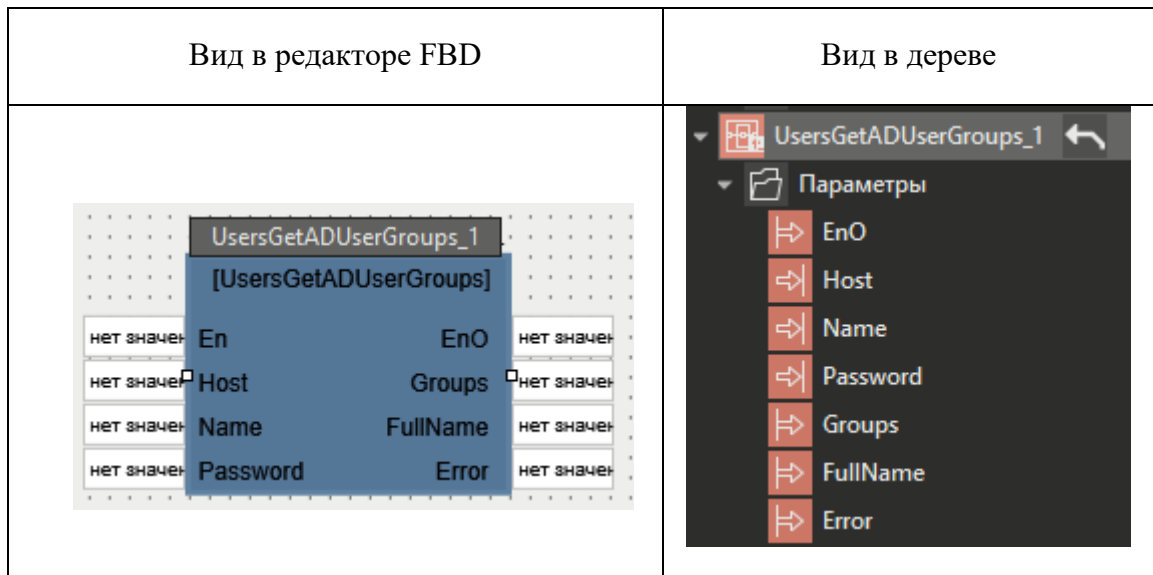


Как правило, изменение пароля происходит в результате нажатия на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания функционального блока на вход действия Программа.

Входы функционального блока можно связать с параметрами Текст элементов Пароль

6.4.3.2.10.USERSGETADUSERGROUPS

Функциональный блок UsersGetADUserGroups служит для получения списка групп, в которые добавлен пользователь операционной системы (Active Directory).



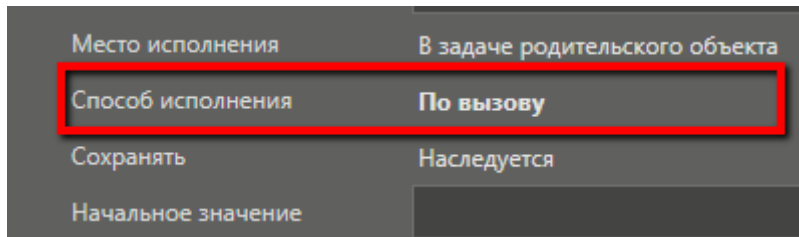
Входы и выходы ФБ:

Название	Тип	Назначение
Host	STRING	Вход ФБ. Указывается имя домена (host) или IP-адрес сервера, где настроен Active Directory, например, adtest.mpssoft.ru или 10.0.6.90
Name	STRING	Вход ФБ. Задается имя (логин) пользователя AD, о котором необходимо получить информацию.
Password	STRING	Вход ФБ. Задается пароль пользователя AD, о котором необходимо получить информацию
Groups	STRING	Выход ФБ. Выдается список всех групп, в которые назначен пользователь AD. Если групп несколько, то они будут перечислены через запятую.
FullName	STRING	Выход ФБ. Выдается полное имя пользователя, с которым он зарегистрирован в AD.
Error	STRING	Выход ФБ. Выдается текст ошибки в случае неуспешной работы функционального блока

Если сервер доступен, пользователь существует в системе и был указан на фходе ФБ правильный пароль, то на выходе получим список групп, к которой он принадлежит в AD, и полное имя.

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:

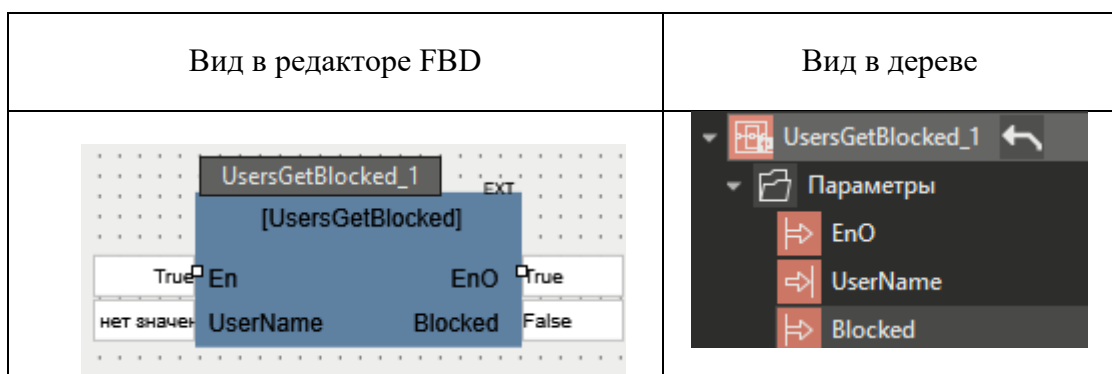


Как правило, для получения списка групп в окне создают Кнопку, затем в ее событие Клик мыши добавляют действие Вызвать программу (для этого функциональный блок следует перетащить на вход действия Программа).

6.4.3.2.11.USERSGETBLOCKED

Функциональный блок UsersGetBlocked служит для получения информации о состоянии блокировки пользователя. Блокировка учетной записи пользователя, это состояние при котором пользователь не может запустить новый сеанс работы в клиенте визуализации, но сведения об этом пользователе не удаляются из исполнительной системы.

Важно! Для работы данного функционального блока у авторизованного пользователя должно быть настроено право доступа Управление блокировкой пользователей.

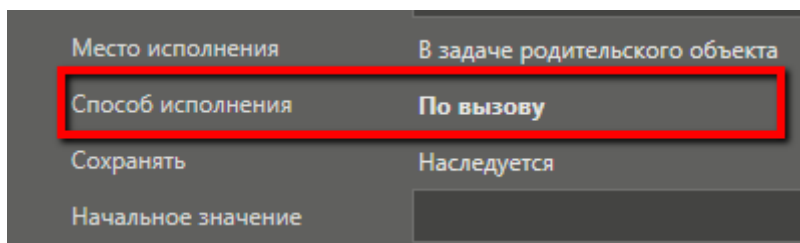


Входы и выходы ФБ:

Название	Тип	Назначение
UserName	STRING	Вход ФБ. Задается имя пользователя, информацию о наличии или отсутствии блокировки которого нужно получить.
Blocked	BOOL	Выход ФБ. Выдается состояние блокировки пользователя. TRUE - пользователь заблокирован, False - разблокирован

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



Как правило, получение информации о состоянии блокировки пользователя происходит после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания этого функционального блока на вход действия Программа.

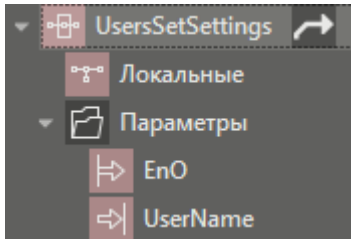
6.4.3.2.12.USERSSETSETTINGS

Функциональный блок UsersSetSettings является типом для экземпляров функциональных блоков, расширенных разработчиками проекта, предназначенных для изменения значений дополнительных настроек пользователей в режиме исполнения.

Важно! Для работы данного функционального блока у авторизованного пользователя должно быть настроено право доступа Изменение пользователей

В результате работы функционального блока, если у права доступа Изменение пользователей установлен флаг Журнал, то сформируется сообщение. Тип сообщения будет Действие пользователей

Вид в дереве библиотек

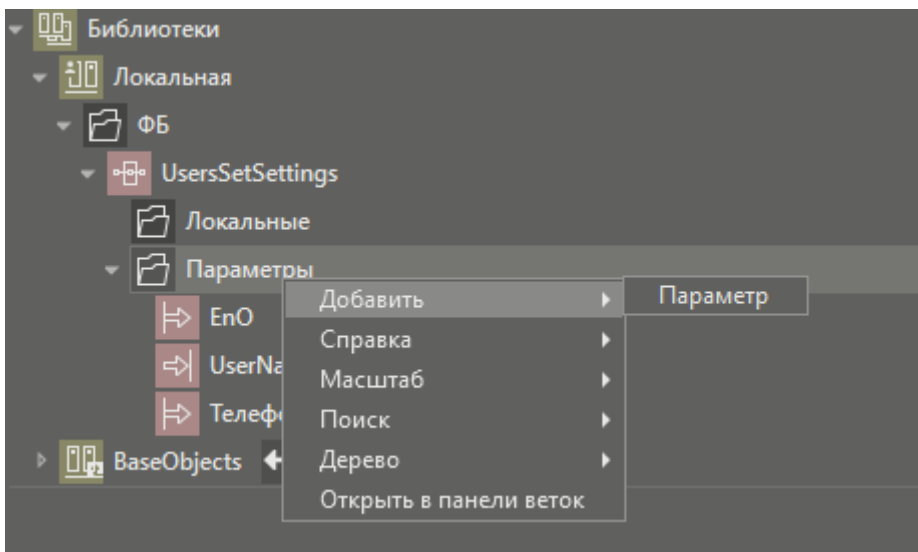


Входы и выходы ФБ:

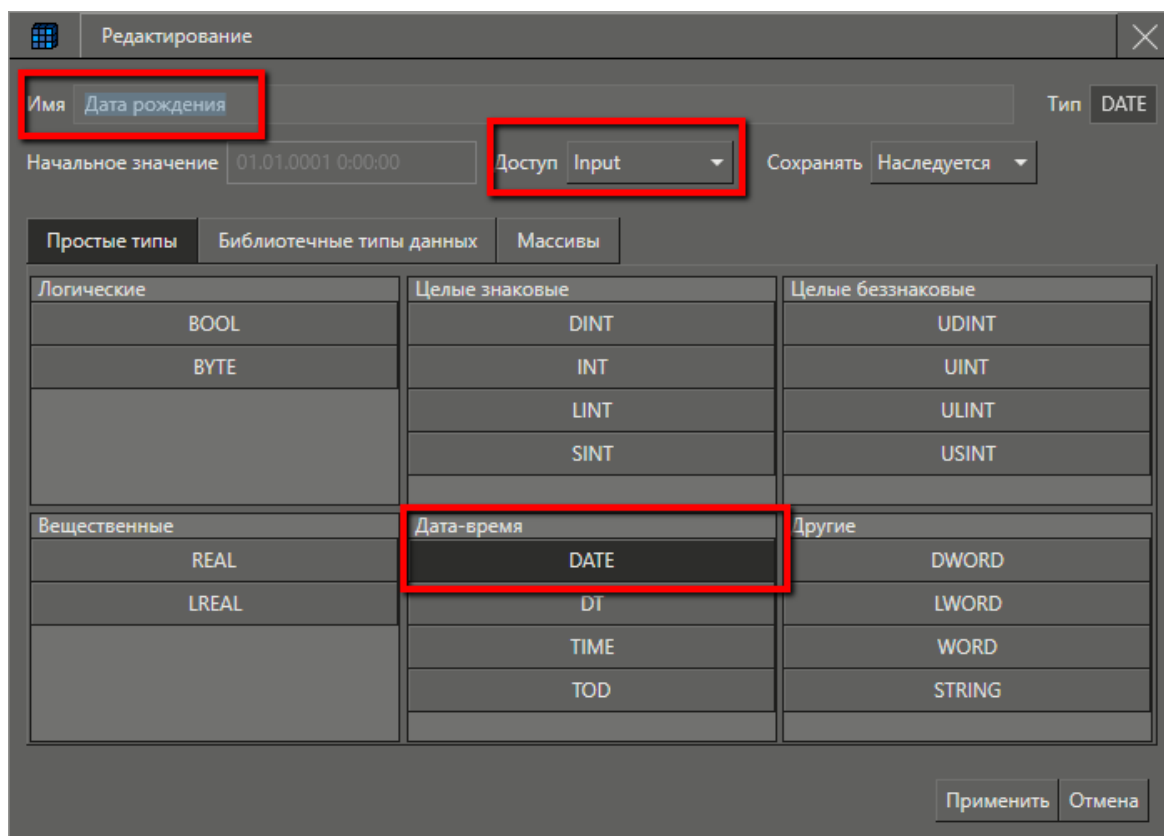
Название	Тип	Назначение
UserName	STRING	Задается имя пользователя, значения дополнительных настроек которого нужно изменить

Рекомендации по работе с ФБ

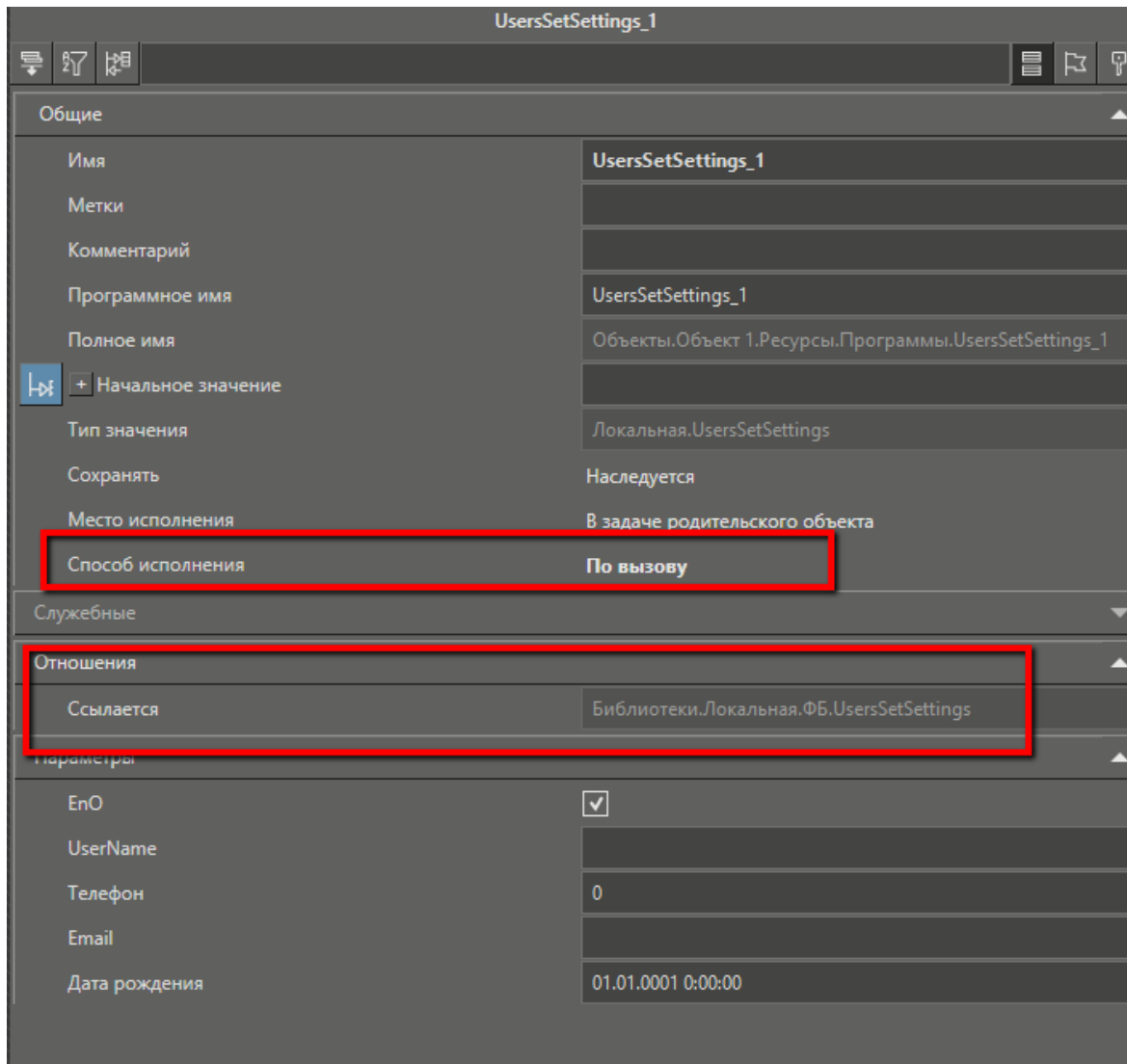
Список дополнительных настроек пользователя зависит от требований проекта, чтобы создать экземпляр функционального блока, который может работать в конкретном проекте необходимо добавить в пользовательскую библиотеку экземпляр функционального блока UsersSetSettings. Затем добавить в него нужное количество входов.



Имена и типы данных входов должны совпадать с именами дополнительных настроек пользователя, значения которых нужно изменить:



Далее в проекте необходимо использовать созданный функциональный блок. В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:

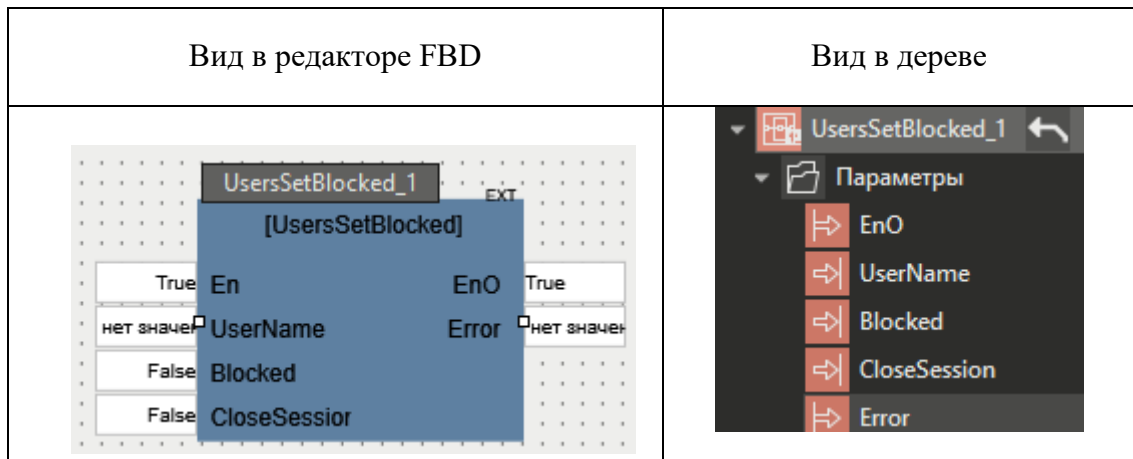


Как правило, для изменения значений дополнительных настроек пользователя в окне создают Кнопку, затем в ее событие Клик мыши добавляют действие Вызвать программу (для этого функциональный блок следует перетащить на вход действия Программа).

6.4.3.2.13. USERSSETBLOCKED

Функциональный блок UsersSetBlocked служит для блокировки учетной записи пользователя, при которой пользователь не может запустить новый сеанс работы в клиенте визуализации, но сведения об этом пользователе не удаляются из исполнительной системы.

Важно! Для работы данного функционального блока у авторизованного пользователя должно быть настроено право доступа Управление блокировкой пользователей.

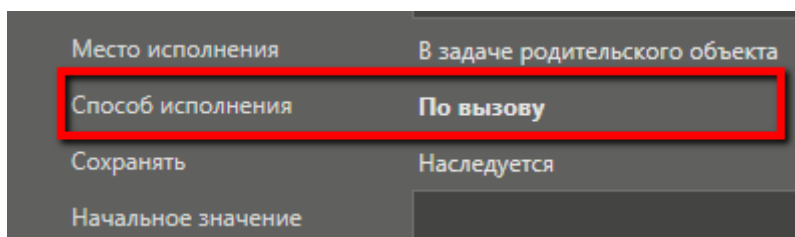


Входы и выходы ФБ:

Название	Тип	Назначение
UserName	STRING	Вход ФБ. Задается имя пользователя, чьей блокировкой необходимо управлять.
Blocked	BOOL	Вход ФБ. Определяет признак блокировки. Если значение TRUE - пользователь будет заблокирован. Если False - разблокирован
CloseSession	BOOL	Вход ФБ. Определяет необходимость закрытия сессии. Если вход Blocked равен TRUE и CloseSessions равен TRUE, то текущие сессии пользователя закрываются
Error	STRING	Выход ФБ. Выдается текст ошибки, если действие блокировка/снятие блокировки не выполнено

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:

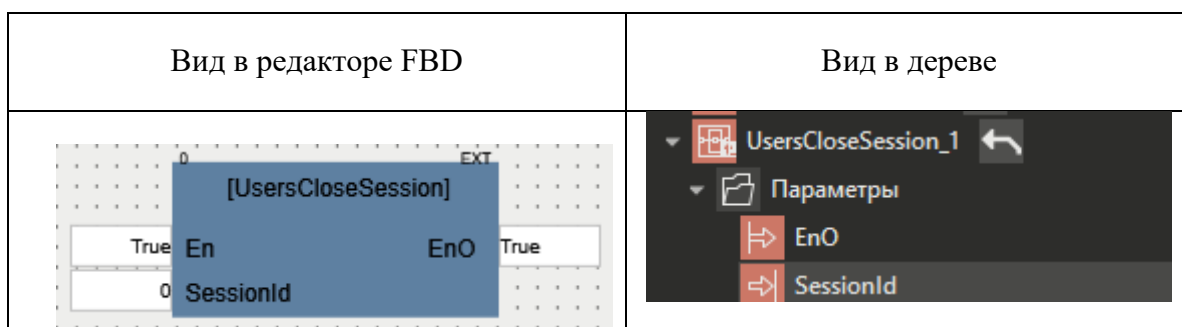


Как правило, изменение состояния блокировки пользователя происходит после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания этого функционального блока на вход действия Программа.

6.4.3.2.14.USERSCLOSESESSION

Функциональный блок UsersCloseSession служит для удаления действующей сессии пользователя в клиенте визуализации.

Важно! Для работы данного функционального блока у авторизованного пользователя должно быть настроено право доступа Управление блокировкой пользователей.

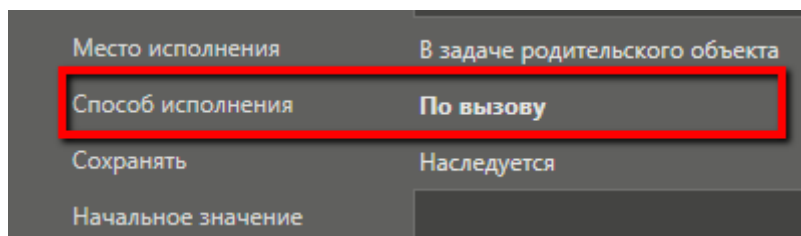


Входы и выходы ФБ:

Название	Тип	Назначение
SessionId	DINT	Вход ФБ. Задается номер сессии, которую нужно удалить. Получить действующие номера сессий можно, используя функциональный блок UsersGetActiveSessions

Рекомендации по работе с ФБ

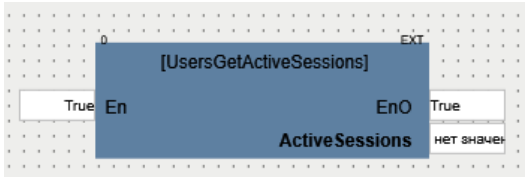
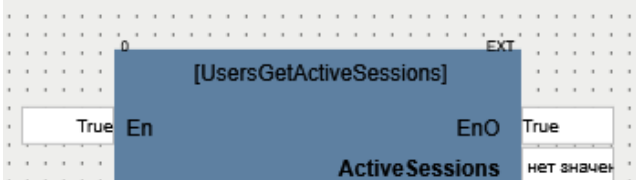
В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



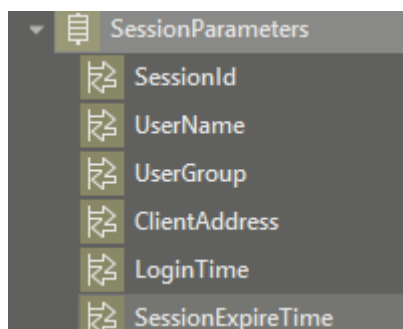
Как правило, удаление сессии пользователя происходит после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания этого функционального блока на вход действия Программа.

6.4.3.2.15. USERSGETACTIVSESSIONS

Функциональный блок UsersGetActiveSessions служит для получения списка сессий клиентов визуализации, которые активны в текущий момент

Вид в дереве		Вид в редакторе FBD
		
Название	Тип	Назначение
ActiveSessions	ARRAY [*] OF SessionParameters	Выход ФБ. Выдается массив структур с информацией об активной сессии. Количество элементов массива зависит, от количества подключенных клиентов визуализации к исполнительной системе: одно подключение - один элемент массива.

Структура SessionParameters находится в библиотеке BaseObjects. Типы данных и имеет следующие поля:



Название	Тип	Назначение
SessionId	DINT	Указывается номер сессии
UserName	STRING	Имя текущего пользователя в данной сессии

Название	Тип	Назначение
UserGroup	STRING	Выдаются роли, в которые назначен данный пользователь.
ClientAddress	STRING	Адрес устройства, на котором открыт клиент визуализации
LoginTime	DT	Время начала сессии (время входа пользователя в систему).
SessionExpireTime	DT	Время окончания сессии пользователя. Значение 0 указывает на то, что время сессии не ограничено

6.4.3.2.16. USERSGETALLOWEDADDRESS

Функциональный блок UsersGetAllowedAddress служит для получения списка адресов, с которых может подключаться пользователь, залогиненный в клиенте визуализации к исполнительной системе

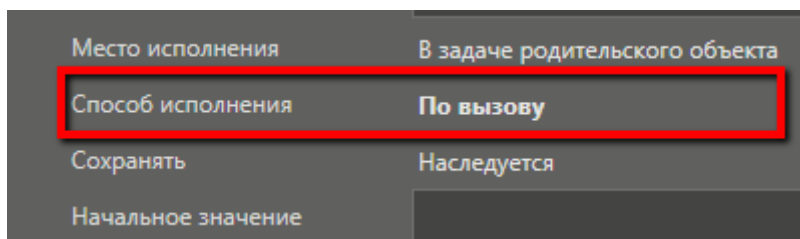
Вид в редакторе FBD	Вид в дереве

Входы и выходы ФБ:

Название	Тип	Назначение
UserName	STRING	Вход ФБ. Задается имя пользователя, информацию об ограничении которого необходимо получить.
AllowedAddress	STRING	Выход ФБ. Выдается список разрешенных адресов. Адреса указываются через запятую. Например, 10.0.0.1,10.0.0.17

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



Как правило, запрос доступных адресов подключения происходит после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания этого функционального блока на вход действия Программа.

6.4.3.2.17. USERSSETALLOWEDADDRESS

Функциональный блок UsersSetAllowedAddress служит для определения списка адресов, с которых может подключаться пользователь, залогиненный в клиенте визуализации к исполнительной системе

Важно! Для работы данного функционального блока у авторизованного пользователя должно быть настроено право доступа Управление блокировкой пользователей.

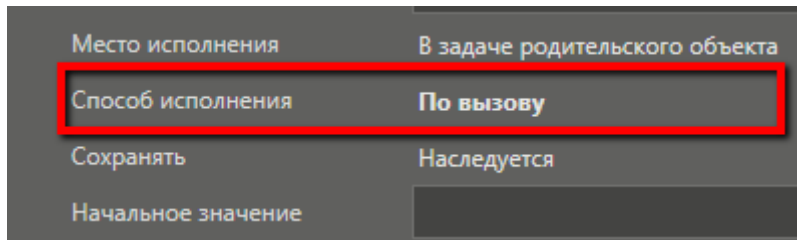
Вид в редакторе FBD	Вид в дереве

Входы ФБ:

Название	Тип	Назначение
UserName	STRING	Задается имя пользователя, для которого необходимо установить ограничение
AllowedAddress	STRING	Указывается список разрешенных адресов. Адреса указываются через запятую. Например, 10.0.0.1,10.0.0.17

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



Как правило, определение доступных адресов подключения происходит после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания этого функционального блока на вход действия Программа.

6.4.3.2.18.USERSGETALLOWEDTIME

Функциональный блок UsersGetAllowedTime служит для получения интервала времени, в течении которого пользователь может иметь залогиниться в клиенте визуализации

Вид в редакторе FBD	Вид в дереве

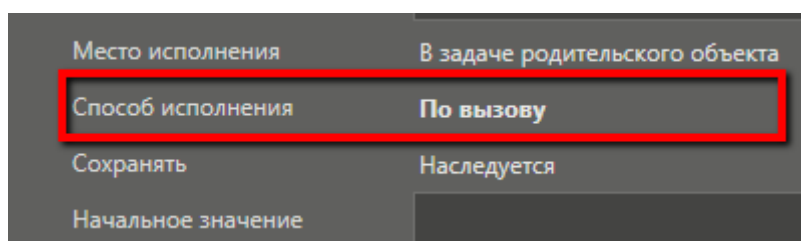
Входы и выходы ФБ:

Название	Тип	Назначение
UserName	STRING	Вход ФБ. Задается имя пользователя, информацию об ограничении которого необходимо получить.
DateIntervals	ARRAY [*] OF DT_INTERVAL	Выход ФБ. Выдает массив структур, где каждый элемент массива указывает на интервал дат, в течении которого разрешен вход пользователя. Структура DT_INTERVAL находится в библиотеке BaseObjects. Типы данных и имеет следующие поля:

Название	Тип	Назначение
		<ul style="list-style-type: none"> • Min - тип поля DT. Определяет нижнюю границу интервала • Max - тип поля DT. Определяет верхнюю границу интервала
TimeIntervals	ARRAY [*] OF TOD_INTERVAL	<p>Выход ФБ. Выдает массив структур, где каждый элемент массива указывает на интервал времени, в течении которого разрешен вход пользователя.</p> <p>Структура TOD_INTERVAL находится в библиотеке BaseObjects. Типы данных и имеет следующие поля:</p> <ul style="list-style-type: none"> • Min - тип поля TOD. Определяет нижнюю границу интервала • Max - тип поля TOD. Определяет верхнюю границу интервала
DisableDaysOfWeek	ARRAY [1..7] OF BOOL	<p>Выход ФБ. Выдает массив, который состоит из семи элементов (тип BOOL). Каждый элемент массива отвечает за день недели (Понедельник - первый элемент, Воскресенье - седьмой). Если значение TRUE для какого-либо элемента массива, то в соответствующий день недели вход пользователя запрещен.</p>

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:

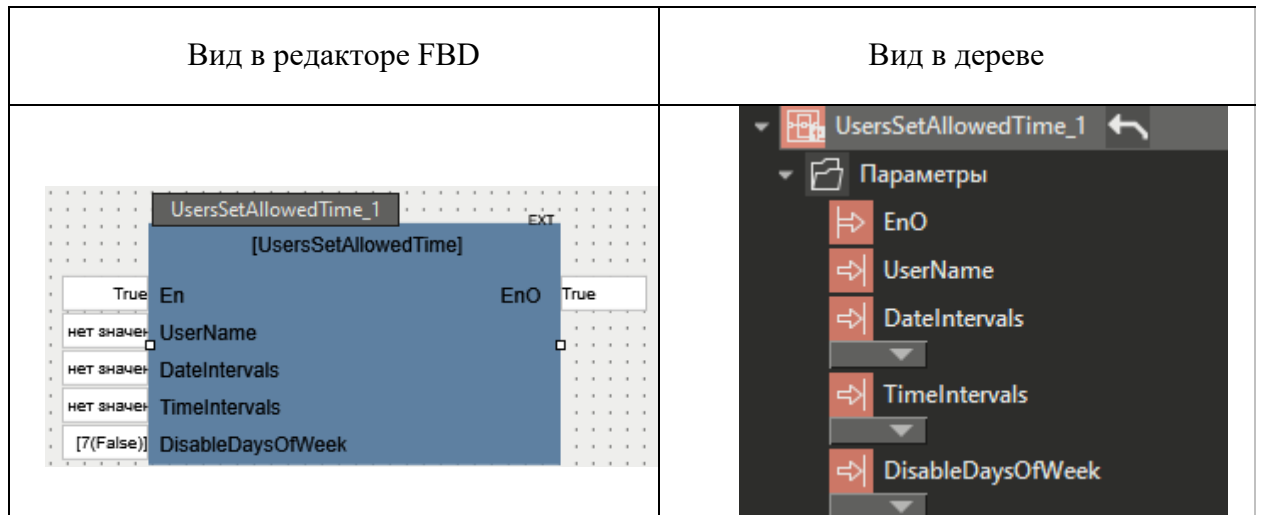


Как правило, запрос времени подключения происходит после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания этого функционального блока на вход действия Программа.

6.4.3.2.19. USERSSETALLOWEDTIME

Функциональный блок UsersSetAllowedTime служит для указания интервала времени, в течении которого пользователь может иметь возможность залогиниться в клиенте визуализации.

Важно! Для работы данного функционального блока у авторизованного пользователя должно быть настроено право доступа Управление блокировкой пользователей.



Входы ФБ:

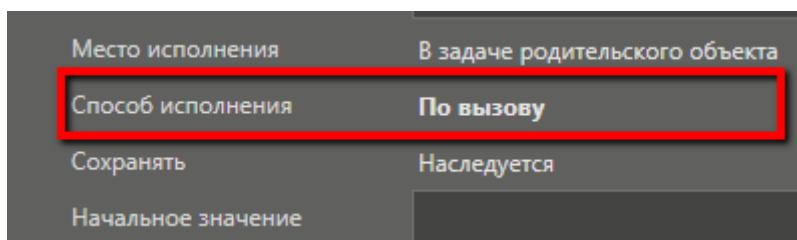
Название	Тип	Назначение
UserName	STRING	Задается имя пользователя, информацию об ограничении которого необходимо получить.
DateIntervals	ARRAY [*] OF DT_INTERVAL	<p>Задается массив структур, где каждый элемент массива указывает на интервал дат, в течении которого разрешен вход пользователя.</p> <p>Структура DT_INTERVAL находится в библиотеке BaseObjects. Типы данных и имеет следующие поля:</p> <ul style="list-style-type: none"> • Min - тип поля DT. Определяет нижнюю границу интервала • Max - тип поля DT. Определяет верхнюю границу интервала
TimeIntervals	ARRAY [*] OF TOD_INTERVAL	Задается массив структур, где каждый элемент массива указывает на интервал времени, в течении которого разрешен вход пользователя.

Название	Тип	Назначение
		<p>Структура TOD_INTERVAL находится в библиотеке BaseObjects. Типы данных и имеет следующие поля:</p> <ul style="list-style-type: none"> • Min - тип поля TOD. Определяет нижнюю границу интервала • Max - тип поля TOD. Определяет верхнюю границу интервала
DisableDaysOfWeek	ARRAY [1..7] OF BOOL	<p>Задается массив, который состоит из семи элементов (тип BOOL). Каждый элемент массива отвечает за день недели (Понедельник - первый элемент, Воскресенье - седьмой). Если значение TRUE для какого-либо элемента массива, то в соответствующий день недели вход пользователя запрещен.</p>

Важно! Если пользователь выполнил вход в разрешенное время, и наступает неразрешенное, то сессия пользователя будет закрыта.

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:

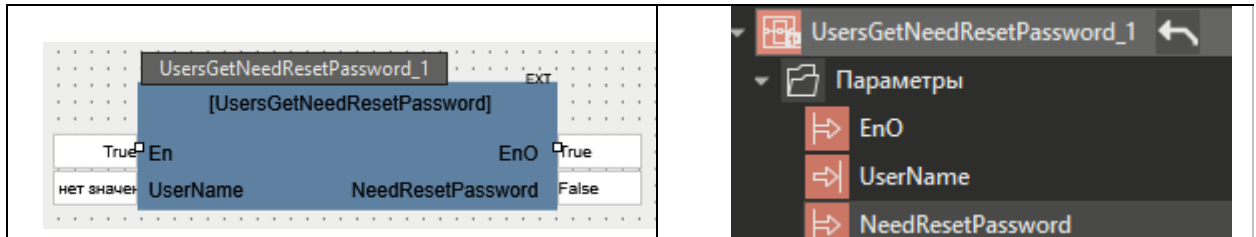


Как правило, ограничение времени подключения происходит после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания этого функционального блока на вход действия Программа.

6.4.3.2.20.USERSGETNEEDRESETPASSWORD

Функциональный блок UsersGetNeedResetPassword служит для получения информации требуется ли пользователю при следующей авторизации сменить свой пароль.

Вид в редакторе FBD	Вид в дереве
---------------------	--------------

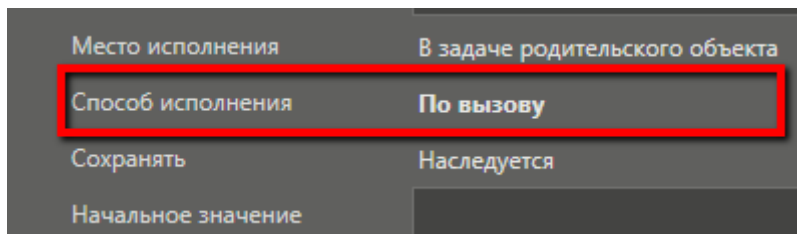


Входы и выходы ФБ:

Название	Тип	Назначение
UserName	STRING	Вход ФБ. Задается имя пользователя, информацию о котором необходимо получить.
NeedResetPassword	BOOL	Выход ФБ. Отображается необходимость сменить пароль. Если выход принимает значение TRUE, то при авторизации пользователя, текущий пароль нужно будет сменить.

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:

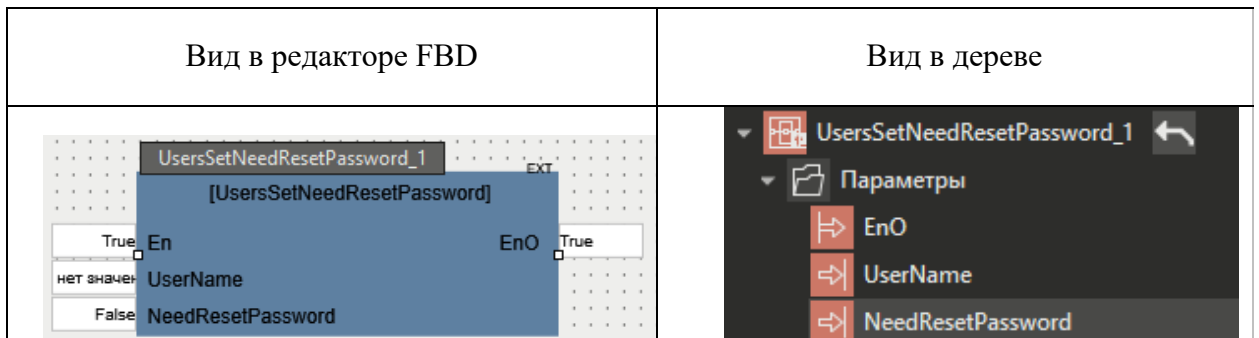


Как правило, запрос информации о необходимости смены пароля происходит после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания этого функционального блока на вход действия Программа.

6.4.3.2.21. USERSSETNEEDRESETPASSWORD

Функциональный блок UsersSetNeedResetPassword служит для установки условия, чтобы пользователь при следующей авторизации сменил свой пароль

Важно! Для работы данного функционального блока у авторизованного пользователя должно быть настроено право доступа Управление блокировкой пользователей.

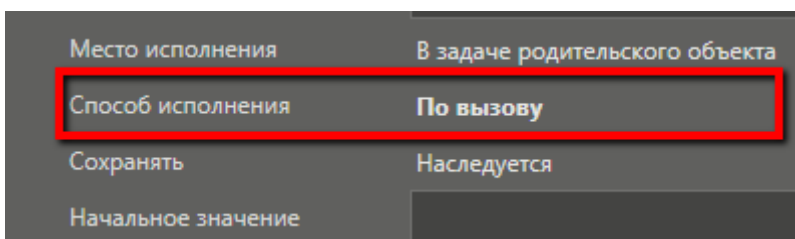


Входы ФБ:

Название	Тип	Назначение
UserName	STRING	Задается имя пользователя, информацию о котором необходимо получить.
NeedResetPassword	BOOL	Задается необходимость сменить пароль. Если выход принимает значение TRUE, то при авторизации пользователя текущий пароль нужно будет сменить. Необходимо, чтобы у пользователя, который будет менять свой пароль было право Изменение собственного пароля

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:

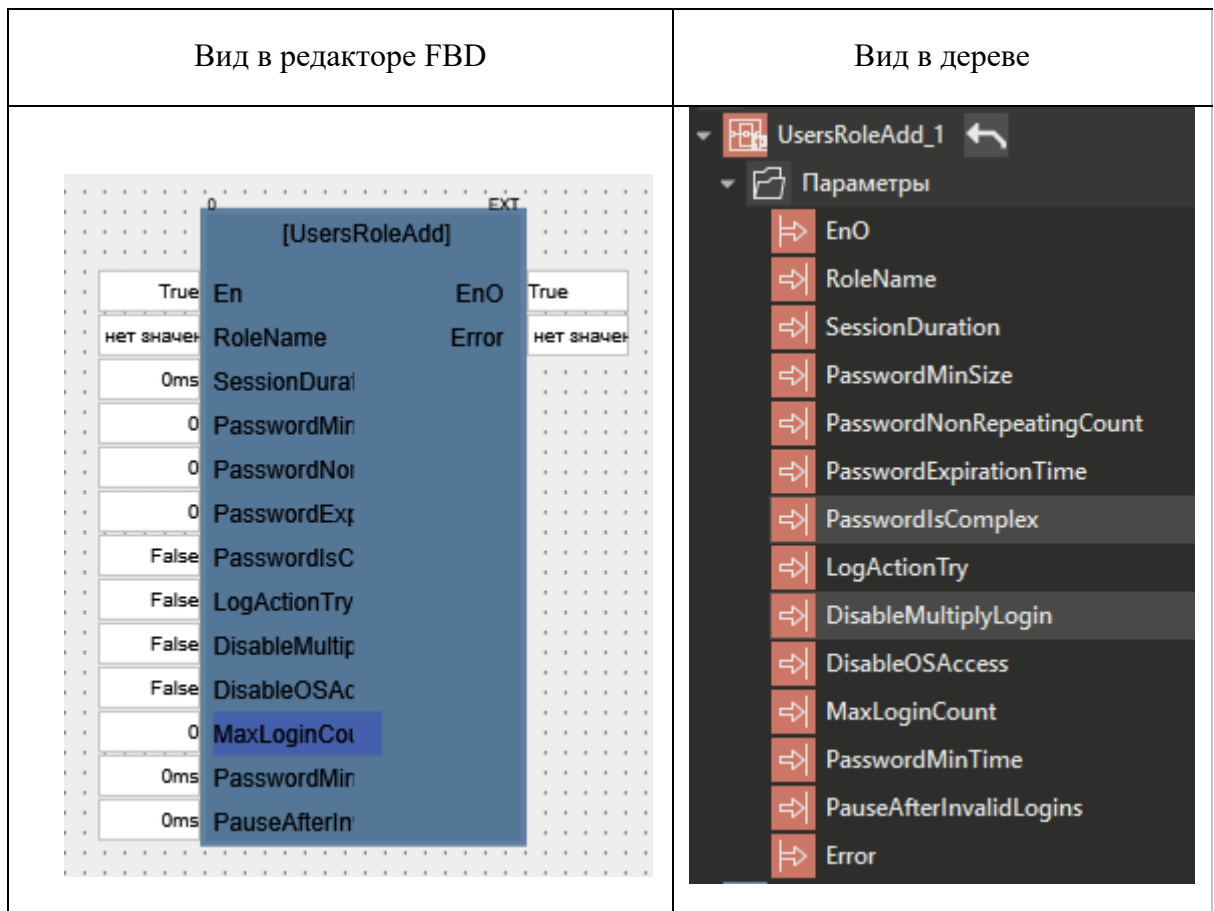


Как правило, запрос информации о необходимости смены пароля происходит после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания этого функционального блока на вход действия Программа.

6.4.3.2.22.USERSROLEADD

Функциональный блок UsersRoleAdd служит для добавления ролей пользователей и их настроек в режиме исполнения. Если используется для хранения настроек пользователей сетевая БД настроек, то изменения выполненные на одном узле будут доступны на всех остальных узлах, которые добавлены в проект. Если сетевая БД настроек не используется, то информация о пользователях хранится в локальном файле, который доступен только для текущего узла. Таким образом, настройки выполненные при помощи данного ФБ будут применяться только для того узла, где они были выполнены.

Важно! Для работы данного функционального блока у авторизованного пользователя должно быть настроено право доступа Изменение операторов



Входы и выходы ФБ:

Название	Тип	Назначение
RoleName	STRING	Задается имя создаваемой роли
SessionDuration	TIME	Указывается длительность сессии. Если время длительности сессии достигает указанного значения, то сессия пользователя будет завершена.

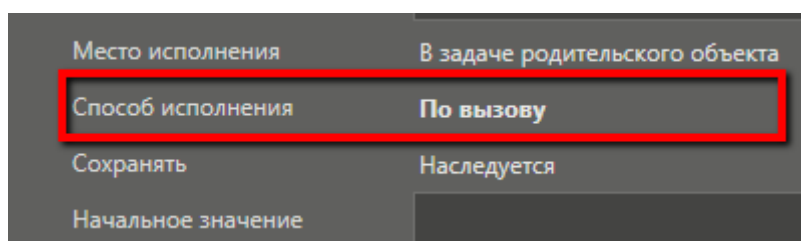
Название	Тип	Назначение
PasswordMinSize	INT	Определяется минимальное количество символов при настройке пароля в режиме исполнения.
PasswordNonRepeatingCount	INT	Определяет, сколько раз пользователь должен задать неповторяющийся пароль. Если указано значение 0, то ограничения отсутствуют. Пример: Если указано 3, то пользователь должен трижды сменить текущий пароль на новый, отличный от текущего и двух предыдущих, а в дальнейшем (четвертый пароль) может задать пароль, совпадающий с первым.
PasswordExpirationTime	TIME	Определяет срок действия пароля пользователя. Если время действия текущего пароля достигло указанного, то при начале очередной сессии пользователя появится сообщение и пользователь не будет допущен к работе, до тех пор пока пароль не будет изменен в режиме исполнения
PasswordIsComplex	BOOL	Определяет состав символов при настройке пароля в режиме исполнения. Если значение TRUE установлено, то пароль должен содержать хотя бы одну цифру, одну прописную и одну строчную букву.
LogActionTry	BOOL	Определяет необходимость фиксирования попыток выполнения действий у пользователей назначенных на эту роль. Если значение TRUE установлено и для действия установлены флаги Подтверждение и Журнал, то запись в журнал будет производиться не только, если пользователь выполнит указанное действие, но и попытается его выполнить. Например, пользователь ввел значение, в соответствии с настройками, появилось окно подтверждения, и пользователь нажал Отмена, в этом случае сообщение в журнале также будет сформировано.
DisableMultiplyLogin	BOOL	Определяет возможность пользователя авторизоваться с двух разных клиентов визуализации одновременно. Если значение TRUE установлено, то возможны следующие варианты:

Название	Тип	Назначение
		<ul style="list-style-type: none"> • В случае, если есть активная сессия для этого пользователя с того же самого адреса, то закрывается прошлая сессия, создается новая • Если текущая сессия запущена на другом устройстве, с другим адресом, то в случае, если она неактивна больше 10 секунд, то она закрывается и создается новая. Если от клиента визуализации продолжают приходить запросы к исполнительной системе, то возвращается ошибка авторизации.
DisableOSAccess	BOOL	<p>Если значение TRUE установлено, то пользователи, добавленную в эту роль, не смогут воспользоваться стандартными горячими клавишам Windows, например, CTRL+ ALT+DEL.</p> <p>Важно! Чтобы использовать данную функцию клиента визуализации необходимо перед первым его использованием запустить файл c:\Program Files\MPSSot\MasterSCADA 4D RT 1.2\Redistributable\Interception\install_ЗапускатьОтИмениАдминистратора.bat</p>
MaxLoginCount	DINT	<p>Определяет сколько раз пользователь может попытаться не успешно начать сессию в клиенте визуализации. После успешного запуска сеанса работы счетчик количества неуспешных попыток должен сбрасываться. Если установлено значение 0, то ограничение отсутствует. Данная настройка применяется только к пользователям, созданным в режиме исполнения</p>
PasswordMinTime	TIME	<p>Определяет минимальный срок существования пароля. Задается в мс. Если с предыдущей смены пароля прошло времени меньше, чем указано в данном свойстве, то при попытке его смены в режиме исполнения функциональный блок (UsersChangeOwnPassword, UsersSetPassword), выполняющий смену пароля выдаст ошибку.</p>
PauseAfterInvalidLogins	TIME	<p>Определяет интервал времени в течении которого пользователь, который не смог авторизоваться за количество попыток, разрешенное в предыдущем пункте, не сможет</p>

Название	Тип	Назначение
		повторить попытки авторизоваться в клиенте визуализации. Данная настройка применяется только к пользователям, созданным в режиме исполнения
Error	STRING	Выдается текст ошибки, если роль не была создана.

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



Как правило, новая роль создается после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляются действие Вызвать программу путем перетаскивания функционального блока на вход действия Программа.

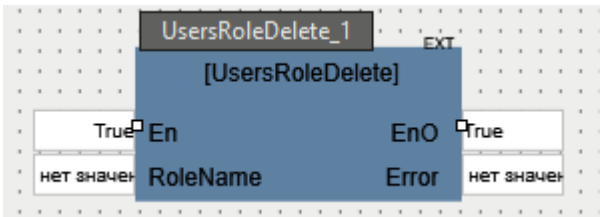
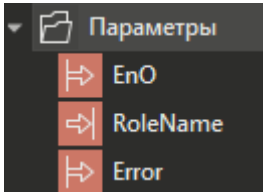
О том, была ли работа ФБ успешной, можно судить по выходу ФБ ЕпО. Если добавления пользователя не произойдет, то выход примет значение False.

6.4.3.2.23.USERSROLEDELETE

Функциональный блок UsersRoleDelete служит для удаления ролей пользователей и их настроек в режиме исполнения. Применяется только для ролей, которые были добавлены в режиме исполнения. Если используется для хранения настроек пользователей сетевая БД настроек, то изменения выполненные на одном узле будут доступны на всех остальных узлах, которые добавлены в проект. Если сетевая БД настроек не используется, то информация о пользователях хранится в локальном файле, который доступен только для текущего узла. Таким образом, настройки выполненные при помощи данного ФБ будут применяться только для того узла, где они были выполнены.

Важно! Для работы данного функционального блока у авторизованного пользователя должно быть настроено право доступа Изменение пользователей

В результате работы функционального блока, если у права доступа Изменение пользователей установлен флаг Журнал, то сформируется сообщение. Тип сообщения будет Действие пользователей

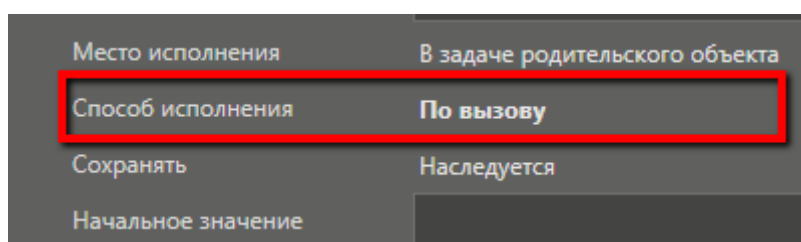
Вид в редакторе FBD	Вид в дереве
	

Входы и выходы ФБ:

Название	Тип	Назначение
RoleName	STRING	Задается имя удаляемой роли
Error	STRING	Выдается текст ошибки, если роль не была удалена.

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:

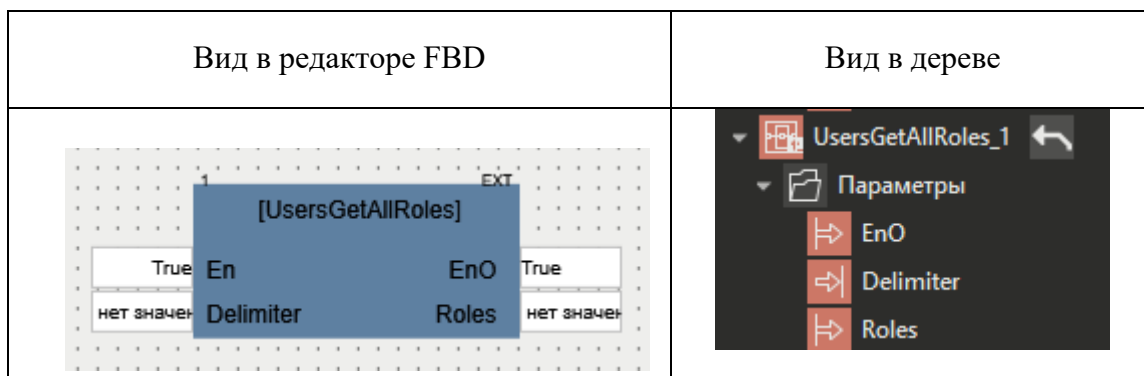


Как правило, роль удаляется после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания функционального блока на вход действия Программа.

О том, была ли работа ФБ успешной, можно судить по выходу ФБ EnO. Если добавления пользователя не произойдет, то выход примет значение False.

6.4.3.2.24.USERSGETALLROLES

Функциональный блок UsersGetAllRoles служит для получения списка ролей пользователей в режиме исполнения. Если используется для хранения настроек пользователей сетевая БД настроек, то появится список ролей всех узлов, иначе только того узла, на котором выполняется ФБ.

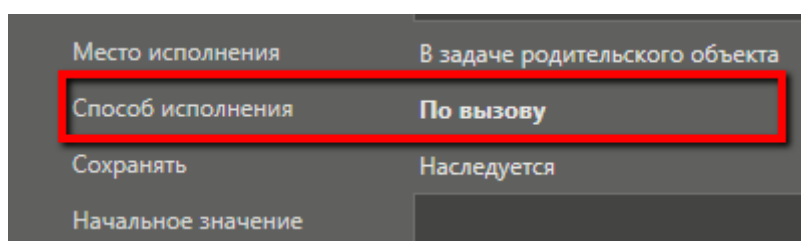


Входы и выходы ФБ:

Название	Тип	Назначение
Delimiter	STRIN G	Указывается символ разделитель, через который будет выдаваться список ролей. Если используется символ, то ~, то выход ФБ можно связать со свойством Список графического элемента Выпадающий список
Roles	STRIN G	Выдается список ролей через указанный разделитель.

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:

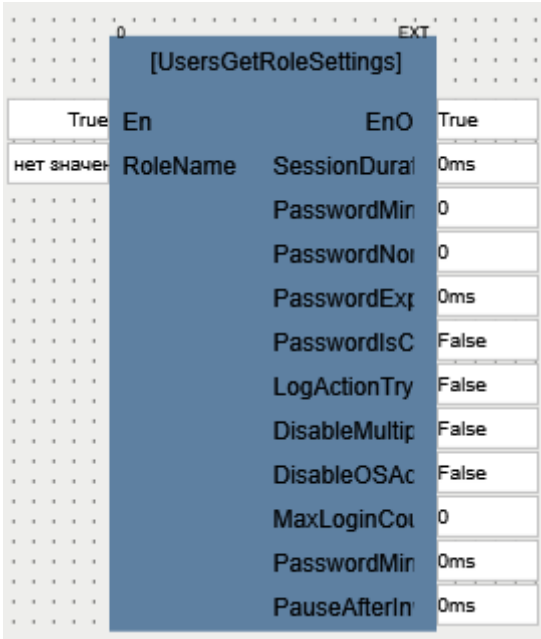
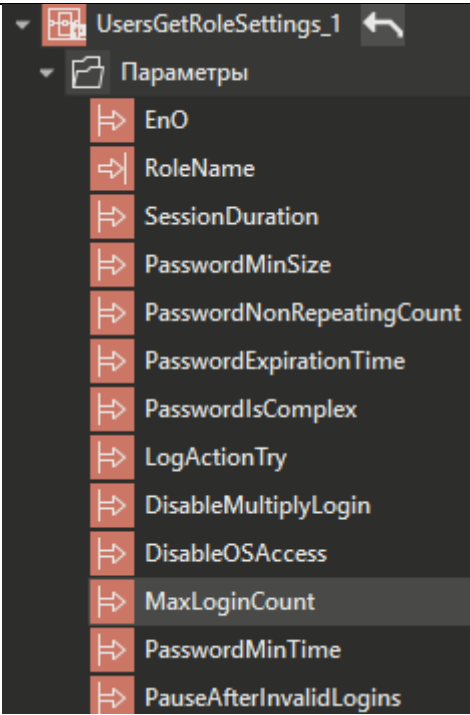


Как правило, список ролей получают после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания функционального блока на вход действия Программа.

О том, была ли работа ФБ успешной, можно судить по выходу ФБ EnO. Если добавления пользователя не произойдет, то выход примет значение False.

6.4.3.2.25.USERSGETROLESETTINGS

Функциональный блок UsersGetRoleSettings служит для получения настроек роли пользователей в режиме исполнения. Если используется для хранения настроек пользователей сетевая БД настроек, то можно получить значения настроек всех ролей проекта. Если сетевая БД настроек не используется, то информация о пользователях хранится в локальном файле, который доступен только для текущего узла.

Вид в редакторе FBD	Вид в дереве
	

Входы и выходы ФБ:

Название	Тип	Назначение
RoleName	STRING	Задается имя создаваемой роли
SessionDuration	TIME	Выдается длительность сессии. Если время длительности сессии достигает указанного значения, то сессия пользователя будет завершена.

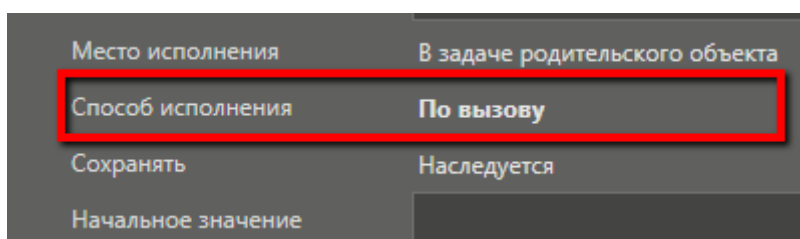
Название	Тип	Назначение
PasswordMinSize	INT	Выдается минимальное количество символов при настройке пароля в режиме исполнения.
PasswordNonRepeatingCount	INT	Выдается, сколько раз пользователь должен задать неповторяющийся пароль. Если указано значение 0, то ограничения отсутствуют. Пример: Если указано 3, то пользователь должен трижды сменить текущий пароль на новый, отличный от текущего и двух предыдущих, а в дальнейшем (четвертый пароль) может задать пароль, совпадающий с первым.
PasswordExpirationTime	TIME	Выдается срок действия пароля пользователя. Если время действия текущего пароля достигло указанного, то при начале очередной сессии пользователя появится сообщение и пользователь не будет допущен к работе, до тех пор пока пароль не будет изменен в режиме исполнения
PasswordIsComplex	BOOL	Выдается состав символов при настройке пароля в режиме исполнения. Если значение TRUE установлено, то пароль должен содержать хотя бы одну цифру, одну прописную и одну строчную букву.
LogActionTry	BOOL	Выдается необходимость фиксирования попыток выполнения действий у пользователей назначенных на эту роль. Если значение TRUE установлено и для действия установлены флаги Подтверждение и Журнал, то запись в журнал будет производиться не только, если пользователь выполнит указанное действие, но и попытается его выполнить. Например, пользователь ввел значение, в соответствии с настройками, появилось окно подтверждения, и пользователь нажал Отмена, в этом случае сообщение в журнале также будет сформировано.
DisableMultiplyLogin	BOOL	Выдается возможность пользователя авторизоваться с двух разных клиентов визуализации одновременно. Если значение TRUE установлено, то возможны следующие варианты:

Название	Тип	Назначение
		<ul style="list-style-type: none"> • В случае, если есть активная сессия для этого пользователя с того же самого адреса, то закрывается прошлая сессия, создается новая • Если текущая сессия запущена на другом устройстве, с другим адресом, то в случае, если она неактивна больше 10 секунд, то она закрывается и создается новая. Если от клиента визуализации продолжают приходить запросы к исполнительной системе, то возвращается ошибка авторизации.
DisableOSAccess	BOOL	<p>Показывает состояние настройки, отвечающей за доступ к операционной системе. Если значение TRUE установлено, то пользователи, добавленную в эту роль, не смогут воспользоваться стандартными горячими клавишам Windows, например, CTRL+ ALT+DEL.</p> <p>Важно! Чтобы использовать данную функцию клиента визуализации необходимо перед первым его использованием запустить файл c:\Program Files\MPSSot\MasterSCADA 4D RT 1.2\Redistributable\Interception\install_ЗапуститьОтИмениАдминистратора.bat</p>
MaxLoginCount	DINT	<p>Выдается сколько раз пользователь может попытаться не успешно начать сессию в клиенте визуализации. После успешного запуска сеанса работы счетчик количества неуспешных попыток должен сбрасываться. Если установлено значение 0, то ограничение отсутствует. Данная настройка применяется только к пользователям, созданным в режиме исполнения</p>
PasswordMinTime	TIME	<p>Выдается минимальный срок существования пароля. Задается в мс. Если с предыдущей смены пароля прошло времени меньше, чем указано в данном свойстве, то при попытке его смены в режиме исполнения функциональный блок (UsersChangeOwnPassword, UsersSetPassword), выполняющий смену пароля выдаст ошибку.</p>
PauseAfterInvalidLogins	TIME	<p>Выдается интервал времени в течении которого пользователь, который не смог авторизоваться за количество</p>

Название	Тип	Назначение
		попыток, разрешенное в предыдущем пункте, не сможет повторить попытки авторизоваться в клиенте визуализации. Данная настройка применяется только к пользователям, созданным в режиме исполнения

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



Как правило, информация о роли получается после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания функционального блока на вход действия Программа.

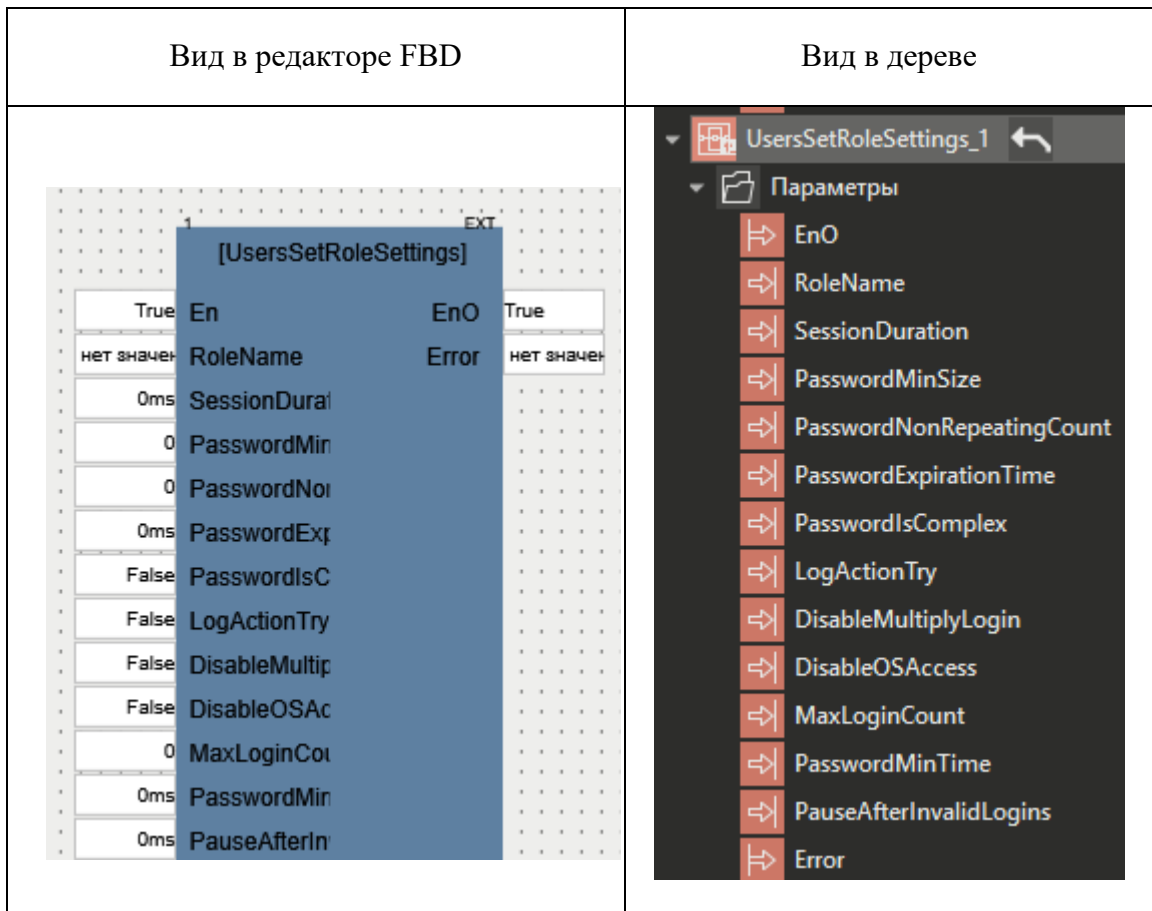
О том, была ли работа ФБ успешной, можно судить по выходу ФБ ЕпО. Если добавления пользователя не произойдет, то выход примет значение False.

6.4.3.2.26. USERSSETROLESETTINGS

Функциональный блок UsersSetRoleSettings служит для изменения настроек роли в режиме исполнения. Если используется для хранения настроек пользователей сетевая БД настроек, то изменения выполненные на одном узле будут доступны на всех остальных узлах, которые добавлены в проект. Если сетевая БД настроек не используется, то информация о пользователях хранится в локальном файле, который доступен только для текущего узла. Таким образом, настройки выполненные при помощи данного ФБ будут применяться только для того узла, где они были выполнены.

Важно! Для работы данного функционального блока у авторизованного пользователя должно быть настроено право доступа Изменение пользователей

В результате работы функционального блока, если у права доступа Изменение пользователей установлен флаг Журнал, то сформируется сообщение, содержащее информацию как о новом значении, так и о предыдущем. Тип сообщения будет Действие пользователей



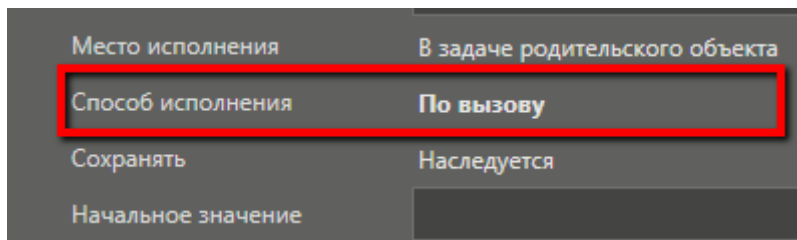
Входы и выходы ФБ:

Название	Тип	Назначение
RoleName	STRING	Задается имя изменяемой роли
SessionDuration	TIME	Указывается длительность сессии. Если время длительности сессии достигает указанного значения, то сессия пользователя будет завершена.
PasswordMinSize	INT	Определяется минимальное количество символов при настройке пароля в режиме исполнения.
PasswordNonRepeatingCount	INT	Определяет, сколько раз пользователь должен задать неповторяющийся пароль. Если указано значение 0, то ограничения отсутствуют. Пример: Если указано 3, то пользователь должен трижды сменить текущий пароль на новый, отличный от текущего и двух предыдущих, а в

Название	Тип	Назначение
		дальнейшем (четвертый пароль) может задать пароль, совпадающий с первым.
PasswordExpirationTime	TIME	Определяет срок действия пароля пользователя. Если время действия текущего пароля достигло указанного, то при начале очередной сессии пользователя появится сообщение и пользователь не будет допущен к работе, до тех пор пока пароль не будет изменен в режиме исполнения
PasswordIsComplex	BOOL	Определяет состав символов при настройке пароля в режиме исполнения. Если значение TRUE установлено, то пароль должен содержать хотя бы одну цифру, одну прописную и одну строчную букву.
LogActionTry	BOOL	Определяет необходимость фиксирования попыток выполнения действий у пользователей назначенных на эту роль. Если значение TRUE установлено и для действия установлены флаги Подтверждение и Журнал, то запись в журнал будет производиться не только, если пользователь выполнит указанное действие, но и попытается его выполнить. Например, пользователь ввел значение, в соответствии с настройками, появилось окно подтверждения, и пользователь нажал Отмена, в этом случае сообщение в журнале также будет сформировано.
DisableMultiplyLogin	BOOL	<p>Определяет возможность пользователя авторизоваться с двух разных клиентов визуализации одновременно. Если значение TRUE установлено, то возможны следующие варианты:</p> <ul style="list-style-type: none"> • В случае, если есть активная сессия для этого пользователя с того же самого адреса, то закрывается прошлая сессия, создается новая • Если текущая сессия запущена на другом устройстве, с другим адресом, то в случае, если она неактивна больше 10 секунд, то она закрывается и создается новая. Если от клиента визуализации продолжают приходить запросы к

Название	Тип	Назначение
		исполнительной системе, то возвращается ошибка авторизации.
DisableOSAccess	BOOL	<p>Если значение TRUE установлено, то пользователи, добавленную в эту роль, не смогут воспользоваться стандартными горячими клавишам Windows, например, CTRL+ ALT+DEL.</p> <p>Важно! Чтобы использовать данную функцию клиента визуализации необходимо перед первым его использованием запустить файл c:\Program Files\MPSSot\MasterSCADA 4D RT 1.2\Redistributable\Interception\install_ЗапускатьОтИмениАдминистратора.bat</p>
MaxLoginCount	DINT	<p>Определяет сколько раз пользователь может попытаться не успешно начать сессию в клиенте визуализации. После успешного запуска сеанса работы счетчик количества неуспешных попыток должен сбрасываться. Если установлено значение 0, то ограничение отсутствует. Данная настройка применяется только к пользователям, созданным в режиме исполнения</p>
PasswordMinTime	TIME	<p>Определяет минимальный срок существования пароля. Задается в мс. Если с предыдущей смены пароля прошло времени меньше, чем указано в данном свойстве, то при попытке его смены в режиме исполнения функциональный блок (UsersChangeOwnPassword, UsersSetPassword), выполняющий смену пароля выдаст ошибку.</p>
PauseAfterInvalidLogins	TIME	<p>Определяет интервал времени в течении которого пользователь, который не смог авторизоваться за количество попыток, разрешенное в предыдущем пункте, не сможет повторить попытки авторизоваться в клиенте визуализации. Данная настройка применяется только к пользователям, созданным в режиме исполнения</p>
Error	STRING	Выдается текст ошибки, если изменения не были внесены.

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



Как правило, изменения вносятся после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания функционального блока на вход действия Программа.

О том, была ли работа ФБ успешной, можно судить по выходу ФБ EnO. Если добавления пользователя не произойдет, то выход примет значение False.

6.4.3.2.27. USERSRESTOREROLESETTINGS

Функциональный блок UsersRestoreRoleSettings восстанавливает настройки роли, которые были заданы в режиме разработки. С ролями, которые были созданы в режиме исполнения данный функциональный блок работать не будет. Если используется для хранения настроек пользователей сетевая БД настроек, то изменения выполненные на одном узле будут доступны на всех остальных узлах, которые добавлены в проект. Если сетевая БД настроек не используется, то информация о пользователях хранится в локальном файле, который доступен только для текущего узла. Таким образом, настройки выполненные при помощи данного ФБ будут применяться только для того узла, где они были выполнены.

Важно! Для работы данного функционального блока у авторизованного пользователя должно быть настроено право доступа Изменение пользователей

В результате работы функционального блока, если у права доступа Изменение пользователей установлен флаг Журнал, то сформируется сообщение, содержащее информацию как о новом значении, так и о предыдущем. Тип сообщения будет Действие пользователей

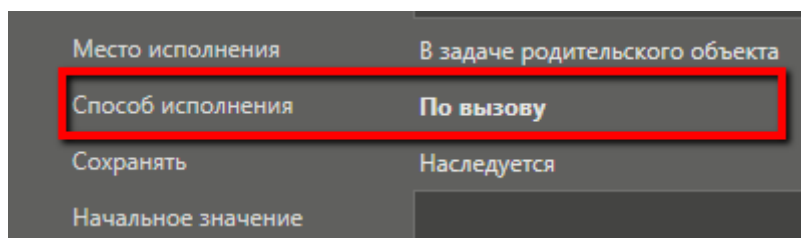
Вид в редакторе FBD	Вид в дереве

Входы и выходы ФБ:

Название	Тип	Назначение
RoleName	STRING	Задается имя роли
Error	STRING	Выдается текст ошибки, если умолчания не были восстановлены

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



Как правило, умолчания восстанавливаются после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания функционального блока на вход действия Программа.

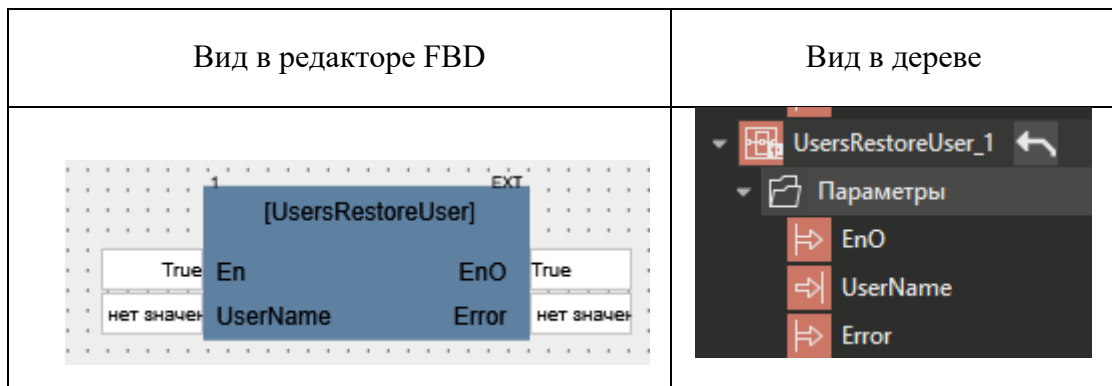
О том, была ли работа ФБ успешной, можно судить по выходу ФБ EnO. Если добавления пользователя не произойдет, то выход примет значение False.

6.4.3.2.28.USERSRESTOREUSER

Функциональный блок UsersRestoreUser восстанавливает настройки пользователя, которые были заданы в режиме разработки. С пользователями, которые были созданы в режиме исполнения данный функциональный блок работать не будет. Если используется для хранения настроек пользователей сетевая БД настроек, то изменения выполненные на одном узле будут доступны на всех остальных узлах, которые добавлены в проект. Если сетевая БД настроек не используется, то информация о пользователях хранится в локальном файле, который доступен только для текущего узла. Таким образом, настройки выполненные при помощи данного ФБ будут применяться только для того узла, где они были выполнены.

Важно! Для работы данного функционального блока у авторизованного пользователя должно быть настроено право доступа Изменение пользователей

В результате работы функционального блока, если у права доступа Изменение пользователей установлен флаг Журнал, то сформируется сообщение. Тип сообщения будет Действие пользователей

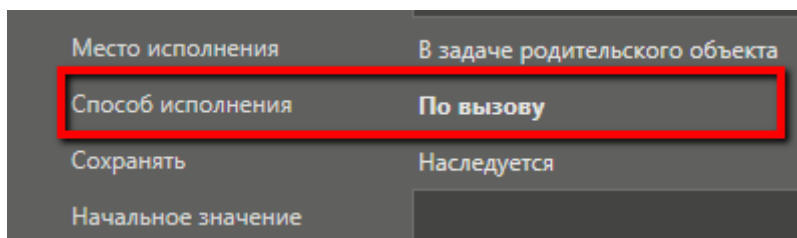


Входы и выходы ФБ:

Название	Тип	Назначение
UserName	STRING	Задается имя пользователя
Error	STRING	Выдается текст ошибки, если умолчания не были восстановлены

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



Как правило, умолчания восстанавливаются после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляются действие Вызвать программу путем перетаскивания функционального блока на вход действия Программа.

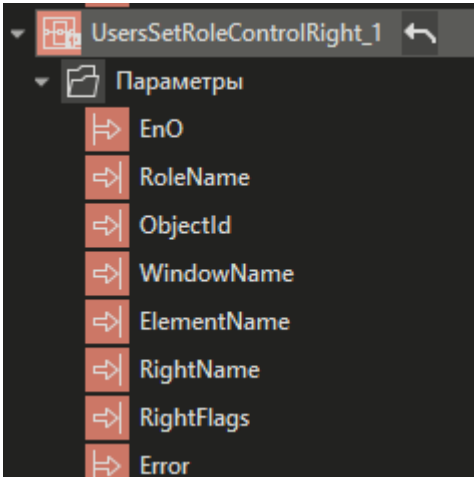
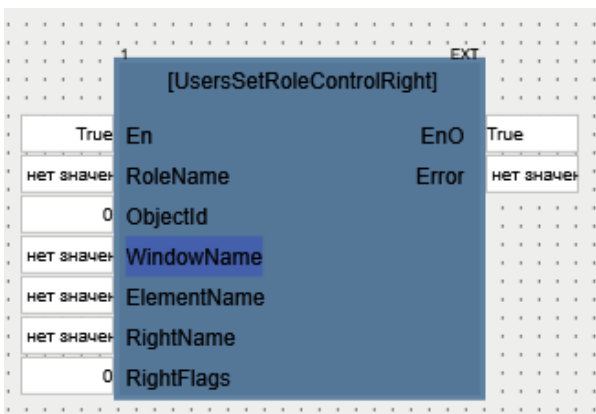
О том, была ли работа ФБ успешной, можно судить по выходу ФБ EnO. Если добавления пользователя не произойдет, то выход примет значение False.

6.4.3.2.29. USERSSETROLECONTROLRIGHT

Функциональный блок UsersSetRoleControlRight служит для изменения прав доступа к элементу в режиме исполнения. Если используется для хранения настроек пользователей сетевая БД настроек, то изменения выполненные на одном узле будут доступны на всех остальных узлах, которые добавлены в проект. Если сетевая БД настроек не используется, то информация о правах хранится в локальном файле, который доступен только для текущего узла. Таким образом, настройки выполненные при помощи данного ФБ будут применяться только для того узла, где они были выполнены.

Важно! Для работы данного функционального блока у авторизованного пользователя должно быть настроено право доступа Изменение пользователей

В результате работы функционального блока, если у права доступа Изменение пользователей установлен флаг Журнал, то сформируется сообщение, содержащее информацию как о новом значении, так и о предыдущем. Тип сообщения будет Действие пользователей

Вид в редакторе FBD	Вид в редакторе FBD
	

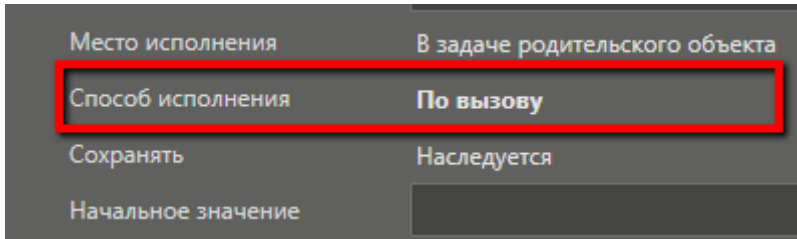
Входы и выходы ФБ:

Название	Тип	Назначение
RoleName	STRING	Задается имя изменяемой роли
ObjectId	REF_TO	Задается ссылка на объект или экземпляр объекта. Можно, например, в программе ST, на данный вход подать строковую переменную, которая будет содержать полное имя объекта/экземпляра, права доступа к которому нужно изменить. Либо можно установить связь между входом ФБ

Название	Тип	Назначение
		и объектом/экземпляром, в этом случае на вход ФБ будет передаваться уникальный идентификатор.
WindowName	STRING	Задается имя окна, права доступа к которому нужно изменить, который содержится в объекте/экземпляре, который задан на входе ObjectId. Вход может быть не задан, если устанавливается право на объект.
ElementName	STRING	Задается имя элемента в окне. Если элемент в окне находится, например в контейнере или в панели, то имя элемента задается в иерархическом виде, например, Панель 1.Инкремент 1 Вход может быть не задан, если устанавливается право на объект или на окно.
RightName	STRING	Указывается программное имя права доступа, которое нужно изменить.
RightFlags	DINT	Задается битовая маска прав, где <ul style="list-style-type: none"> • первый бит отвечает за флаг Разрешить, • второй бит - флаг Запретить, • третий бит - флаг Подтверждение, • четвертый бит - флаг Журнал. Таким образом, чтобы разрешить пользователю изменять например значение инкремента, и это действие зафиксировать в журнале нужно установить значение на входе ФБ 9, чтобы запретить действие 2.
Error	STRING	Выдается текст ошибки, если изменения не были внесены.

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



Как правило, изменения вносятся после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляются действие Вызвать программу путем перетаскивания функционального блока на вход действия Программа.

6.4.3.2.30.USERSGETROLECONTROLRIGHT

Функциональный блок UsersGetRoleControlRight служит для получения назначенных прав доступа для Роли к элементу. Если используется для хранения настроек пользователей сетевая БД настроек, то можно получить значения настроек всех элементов и ролей проекта. Если сетевая БД настроек не используется, то информация о пользователях, ролях и правах хранится в локальном файле, который доступен только для текущего узла.

Вид в редакторе FBD	Вид в дереве

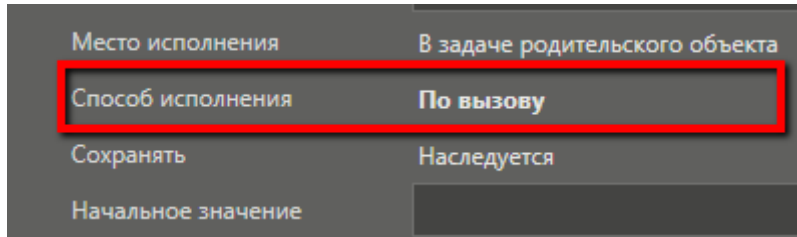
Входы и выходы ФБ:

Название	Тип	Назначение
RoleName	STRING	Задается имя проверяемой роли
ObjectId	REF_TO	Задается ссылка на объект или экземпляр объекта. Можно, например, в программе ST, на данный вход подать строковую переменную, которая будет содержать полное

Название	Тип	Назначение
		имя объекта/экземпляра, права доступа к которому нужно изменить. Либо можно установить связь между входом ФБ и объектом/экземпляром, в этом случае на вход ФБ будет передаваться уникальный идентификатор.
WindowName	STRING	Задается имя окна, который содержится в объекте/экземпляре, который задан на входе ObjectId. Вход может быть не задан, если проверяется право на объект.
ElementName	STRING	Задается имя элемента в окне. Если элемент в окне находится, например в контейнере или в панели, то имя элемента задается в иерархическом виде, например, Панель 1.Инкремент 1 Вход может быть не задан, если проверяется право на объект или на окно.
RightName	STRING	Указывается программное имя права доступа, которое нужно проверить.
RightFlags	DINT	Выход ФБ. Выдается битовая маска прав, где <ul style="list-style-type: none"> • первый бит отвечает за флаг Разрешить, • второй бит - флаг Запретить, • третий бит - флаг Подтверждение, • четвертый бит - флаг Журнал. Таким образом,если разрешено пользователю изменять например значение инкремента, и это действие зафиксировать в журнале появится значение на выходе ФБ 9, если запрещено действие, значение 2.

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



Как правило, проверка выполняется после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляются действие Вызвать программу путем перетаскивания функционального блока на вход действия Программа.

6.4.3.2.31. USERSDELETEROLECONTROLRIGHT

Функциональный блок UsersDeleteRoleControlRight служит для удаления прав доступа какой-либо Роли к элементу в режиме исполнения. Если используется для хранения настроек пользователей сетевая БД настроек, то изменения выполненные на одном узле будут доступны на всех остальных узлах, которые добавлены в проект. Если сетевая БД настроек не используется, то информация о правах хранится в локальном файле, который доступен только для текущего узла. Таким образом, настройки выполненные при помощи данного ФБ будут применяться только для того узла, где они были выполнены.

Важно! Для работы данного функционального блока у авторизованного пользователя должно быть настроено право доступа Изменение пользователей

В результате работы функционального блока, если у права доступа Изменение пользователей установлен флаг Журнал, то сформируется сообщение. Тип сообщения будет Действие пользователей

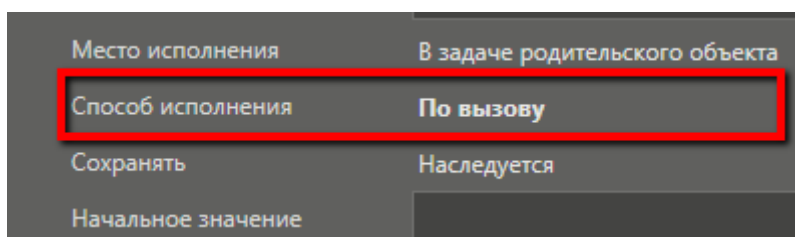
Вид в редакторе FBD	Вид в дереве

Входы и выходы ФБ:

Название	Тип	Назначение
RoleName	STRING	Задается имя изменяемой роли
ObjectId	REF_TO	Задается ссылка на объект или экземпляр объекта. Можно, например, в программе ST, на данный вход подать строковую переменную, которая будет содержать полное имя объекта/экземпляра, права доступа к которому нужно изменить. Либо можно установить связь между входом ФБ и объектом/экземпляром, в этом случае на вход ФБ будет передаваться уникальный идентификатор.
WindowName	STRING	Задается имя окна, который содержится в объекте/экземпляре, который задан на входе ObjectId. Вход может быть не задан, если удаляется право на объект.
ElementName	STRING	Задается имя элемента в окне. Если элемент в окне находится, например в контейнере или в панели, то имя элемента задается в иерархическом виде, например, Панель 1.Инкремент 1 Вход может быть не задан, если удаляется право на объект или на окно.
RightName	STRING	Указывается программное имя права доступа, которое нужно удалить.
Error	STRING	Выдается текст ошибки, если изменения не были внесены.

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:



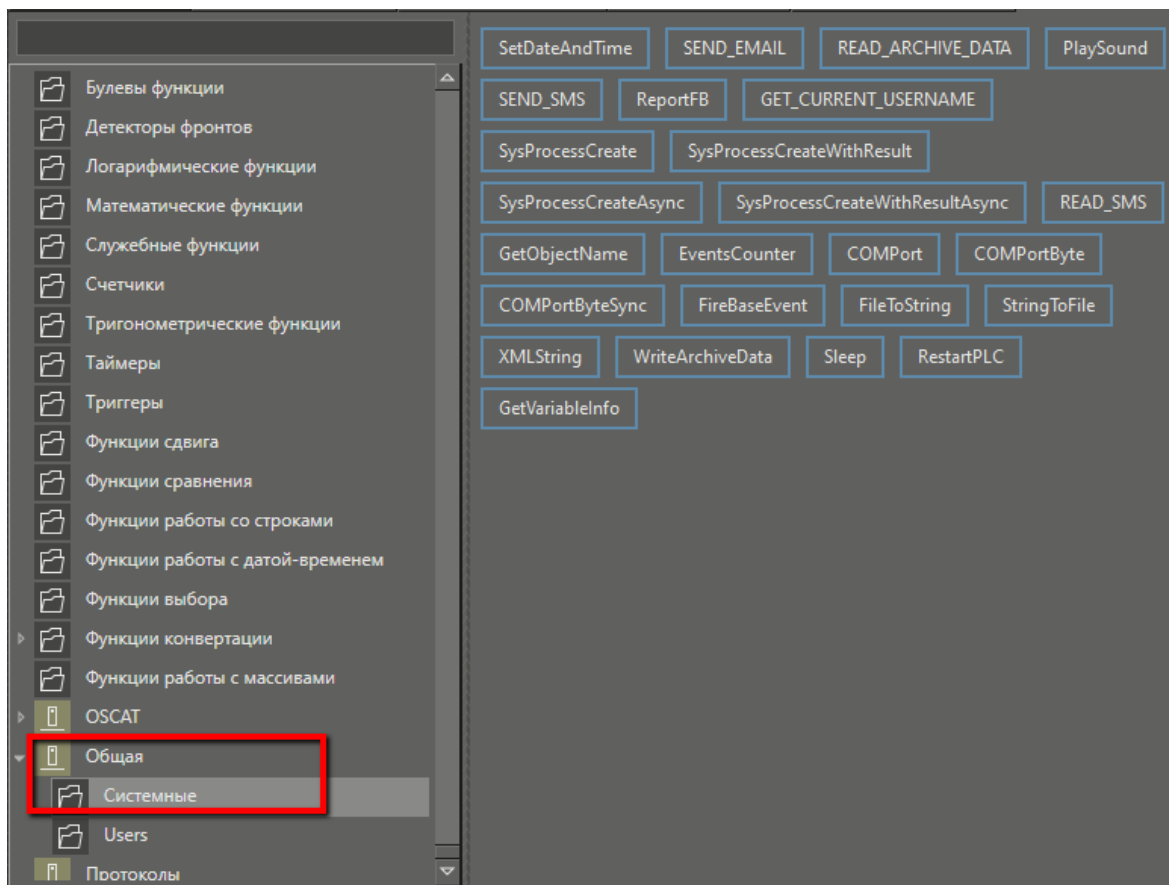
Как правило, изменения вносятся после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляются действие Вызвать программу путем перетаскивания функционального блока на вход действия Программа.

6.4.3.2.32.BASEOBJECTS.ФУНКЦИИ.СИСТЕМНЫЕ

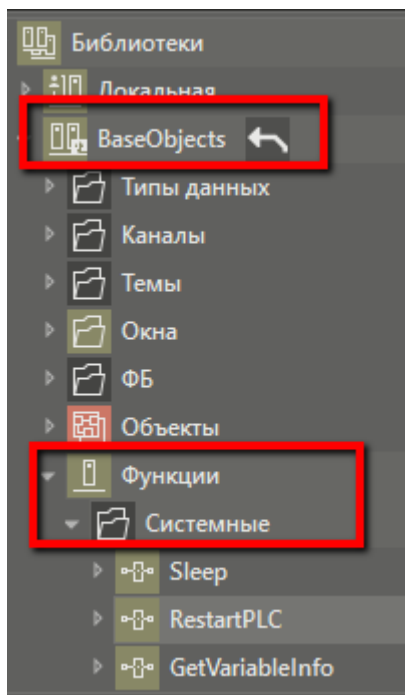
Данная группа содержит системные функции:

- Sleep
- RestartPLC
- GetVariableInfo

В палитре редакторов находится в библиотеке Общие:

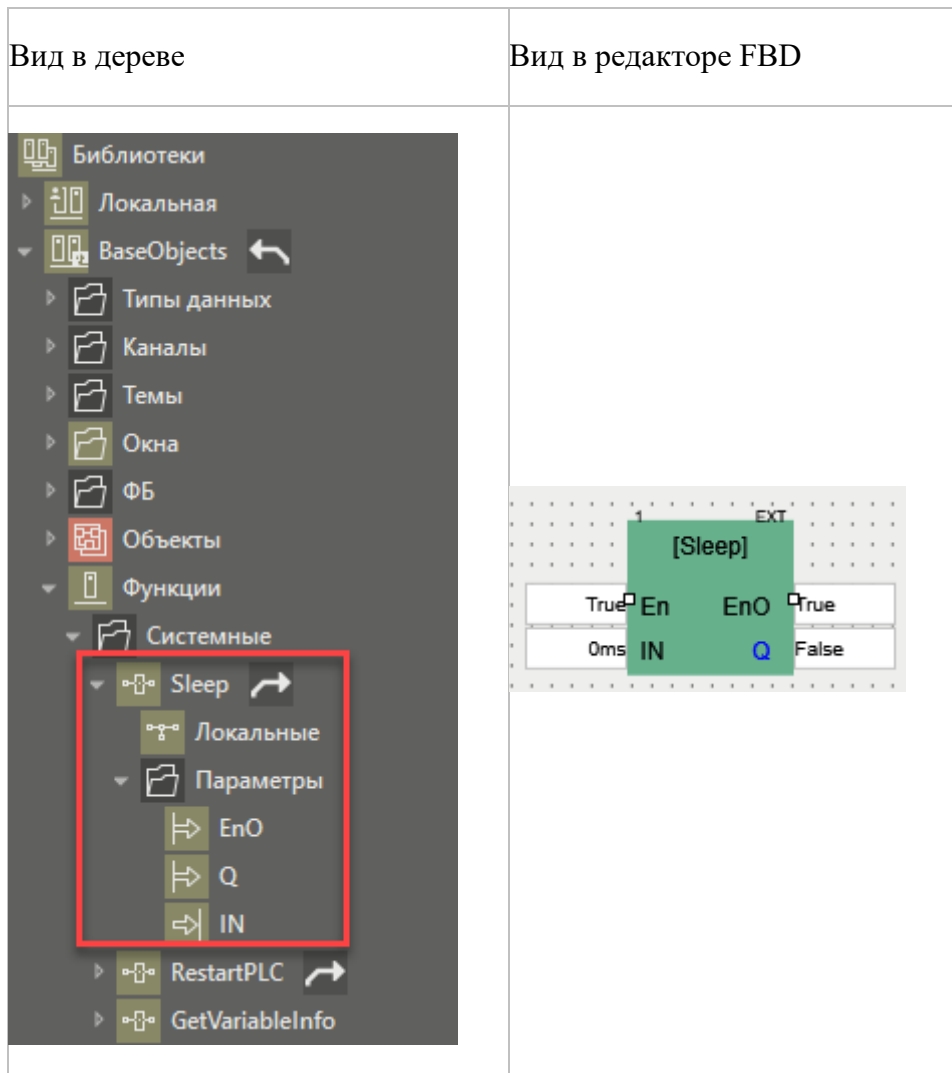


В дереве библиотек находится в категории BaseObjects:



6.4.3.2.33.SLEEP

Функция Sleep служит для приостановки работы потока, в котором работает программа, содержащая данную функцию в режиме исполнения. Находится в библиотеке BaseObjects.Функции.Системные

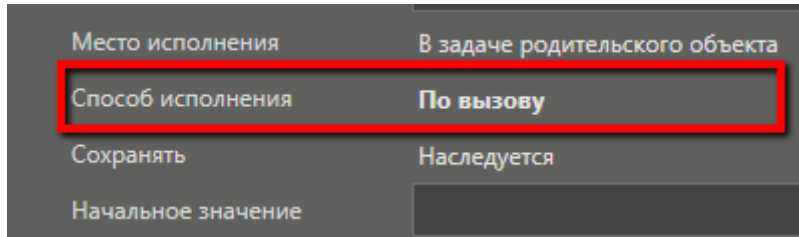


Функция содержит один вход и выход:

Название	Тип	Назначение
IN	TIME	Вход функции. Задается время на которое необходимо остановить поток, в которой работает программа. Если установлено значение 0, то произойдет переключение на другой поток операционной системы
Q	BOOL	Выход функции

В программе ST функцию можно вызывать без записи результата в переменную:
 BaseObjects.Sleep(T#500ms);

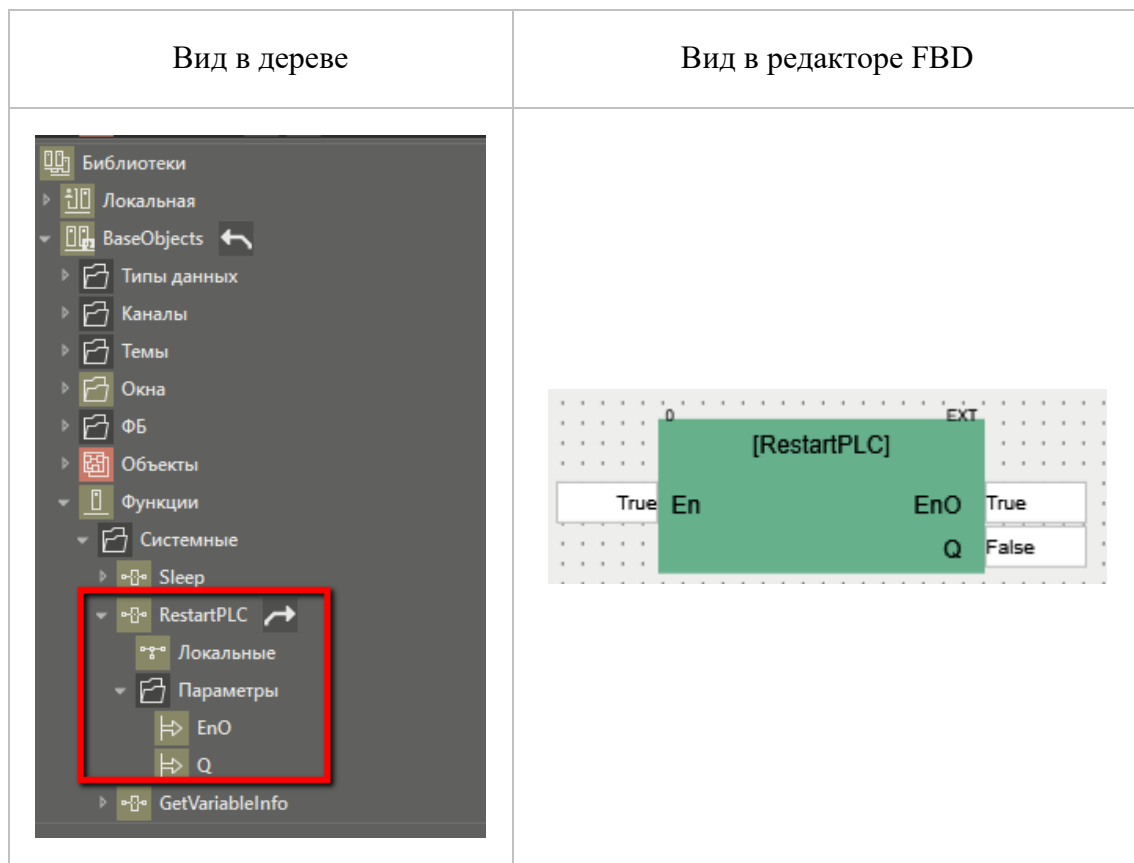
В панели свойств программы, содержащей данную функцию, в категории Общие, нужно установить свойство Способ исполнения в состояние По вызову:



Один из возможных вариантов вызвать программу: нажать пользователю на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания программы на вход действия Программа.

6.4.3.2.34.RESTARTPLC

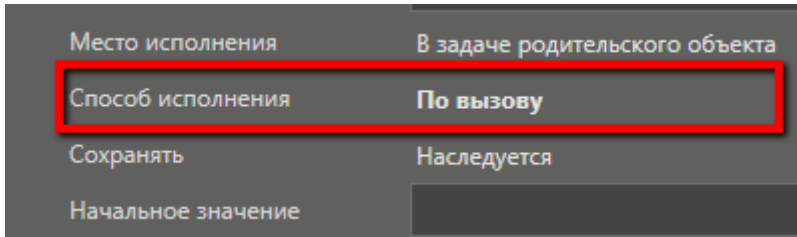
Функция RestartPLC служит для перезагрузки исполнительной системы в режиме исполнения. Находится в библиотеке BaseObjects.Функции.Системные



Функция не содержит специальных входов и выходов. При ее вызове происходит перезагрузка исполнительной системы.

В программе ST функцию можно вызывать без записи результата в переменную: `BaseObjects.RestartPLC();`

В панели свойств программы, содержащей данную функцию, в категории Общие, нужно установить свойство Способ исполнения в состояние По вызову:

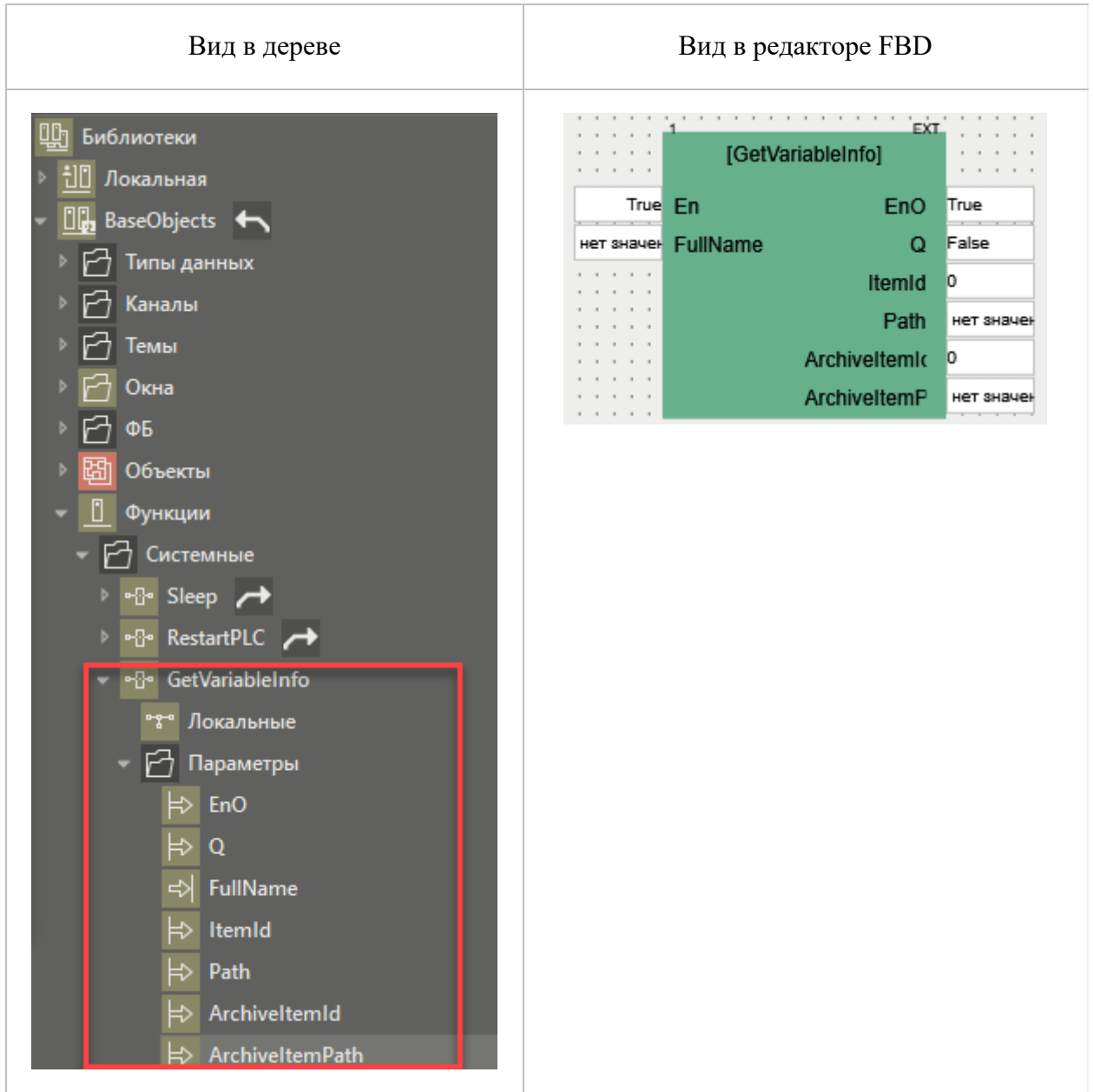


Один из возможных вариантов вызвать программу: нажать пользователю на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания программы на вход действия Программа.

6.4.3.2.35.GETVARIABLEINFO

Функция `GetVariableInfo` служит для поиска параметра по полному имени и выводу информации о его `Id/Path`, который соответствует полям `itemid/path` в таблице `items` в архиве данных

в режиме исполнения. Находится в библиотеке `BaseObjects.Функции.Системные`



Входы и выходы:

Название	Назначение
FullName	Вход функции. Тип STRING. Задается полное имя параметра в проекте
ItemId	Выход функции. Тип LINT. Выдается ID параметра в объекте. Если используются экземпляры объектов, то выдается ID экземпляра.

Название	Назначение
Path	Выход функции. Тип STRING. Выдается путь к параметру внутри экземпляра объекта. Если параметр имеет тип массив структур, то в выдается путь к полю.
ArchiveItemId	Выход функции. Тип LINT. Выдается ID архивируемого параметра, связанного с указанным. Если флаг Архивировать установлен, у самого параметра, то будет выдаваться его собственный ID.
ArchiveItemPath	Выход функции. Тип STRING. Выдается путь к архивируемому параметру, связанному с указанным, внутри экземпляра объекта. Если связанно параметр имеет тип массив структур, то в выдается путь к полю.
Q	Выход функции. Тип BOOL.

6.4.4. БИБЛИОТЕКА PROTOCOLS

Библиотека Protocols подключена по умолчанию к среде разработки.

Данная библиотека содержит протоколы, необходимые для работы с внешними устройствами и программным обеспечением, которые подключены к среде исполнения по умолчанию, но использование которых может потребовать дополнительных опций.

Подробное описание работы с протоколами, содержащимися в данной библиотеке смотрите в разделе [Дополнительные протоколы](#)

Также в этой библиотеке хранятся модули ввода-вывода, специальные типы данных необходимые для работы с этими протоколами.

В группе ФБ содержатся функциональные блоки, которые могут расширить возможности протоколов, содержащихся в этой библиотеке.

Состав группы ФБ:

- SqlRequest
- SqlRequestWithResult
- HttpClient

6.4.4.1. PROTOCOLS.ФБ

В группе ФБ содержатся функциональные блоки, которые могут расширить возможности протоколов, содержащихся в библиотеке Protocols.

Состав группы ФБ:

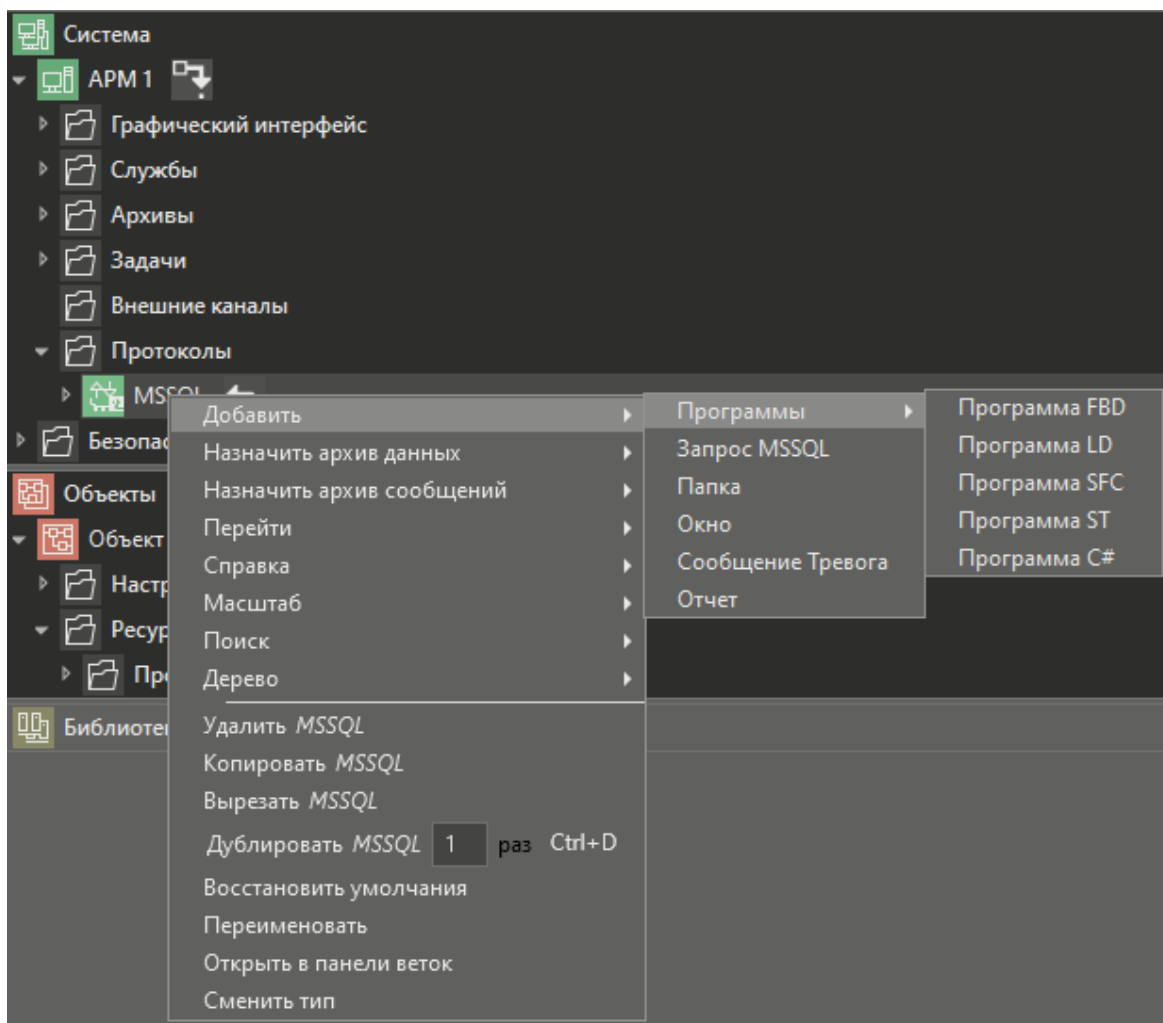
- SqlRequest
- SqlRequestWithResult
- HttpClient

6.4.4.1.1. SQLREQUEST

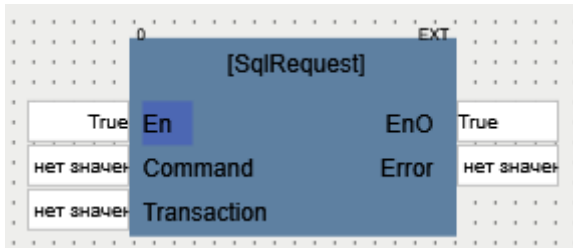
Функциональный блок SqlRequest служит для выполнения запросов к СУБД.

Важно! Данный ФБ можно использовать только в программах внутри протоколов SQL (MSSQL, PostgreSQL, MySQL).

Чтобы добавить программу в протокол необходимо воспользоваться контекстной панелью или контекстным меню:



Затем в программу вставить функциональный блок из палитры. Вид функционального блока в редакторе FBD:



Для корректной работы необходимо определить входы функционального блока.

Название	Тип	Назначение
Command	STRING	Вход ФБ. Задается текст SQL-запроса. Формат записи такой же как и в свойстве Команда элемента Запрос
Transaction	Protocols.SQL Transaction	<p>Вход ФБ. Определяет порядок работы функционального блока. Возможные значения:</p> <p>None - значение по умолчанию один вызов ФБ одна транзакция</p> <p>Start - открывает транзакцию, до тех пор пока значение не сменится на Commit или Rollback все вызовы функционального блока будут идти в одной транзакции. Так же это блокирует подключение к БД до тех пор пока транзакция не завершится.</p> <p>Commit - положительно завершает транзакцию и записывает изменения в БД. Так же освобождает подключение к БД.</p> <p>Rollback - отрицательно завершает транзакцию и откатывает все запросы выполненные за время транзакции. Так же освобождает подключение к БД.</p>
Error	STRING	Выход ФБ. Выдается текст ошибки.

Тип данных SQLTransaction, находится также в библиотеке Protocols

Функциональный блок выполняется синхронно.

Если в Command необходимо использовать входные/выходные параметры, то нужно создать экземпляр данного функционального блока. Чтобы создать экземпляр функционального блока необходимо добавить в пользовательскую библиотеку экземпляр функционального блока SqlRequest. Затем добавить в него нужное количество выходов или входов. И далее в протокол вставлять экземпляры этого наследника.

Важно! Чтение в массив структур не поддерживается

Рекомендации по работе с функциональным блоком

Если ФБ `SqlRequest` вызывается в циклах программ, то в теле цикла необходимо обязательно задавать выход из него, используя условие: `if SqlRequest_1.EnO=false then exit end_if`

Если в момент вызова функционального блока не все его входы будут определены, то в лог-файле исполнительной системы будет сформирована ошибка запроса.

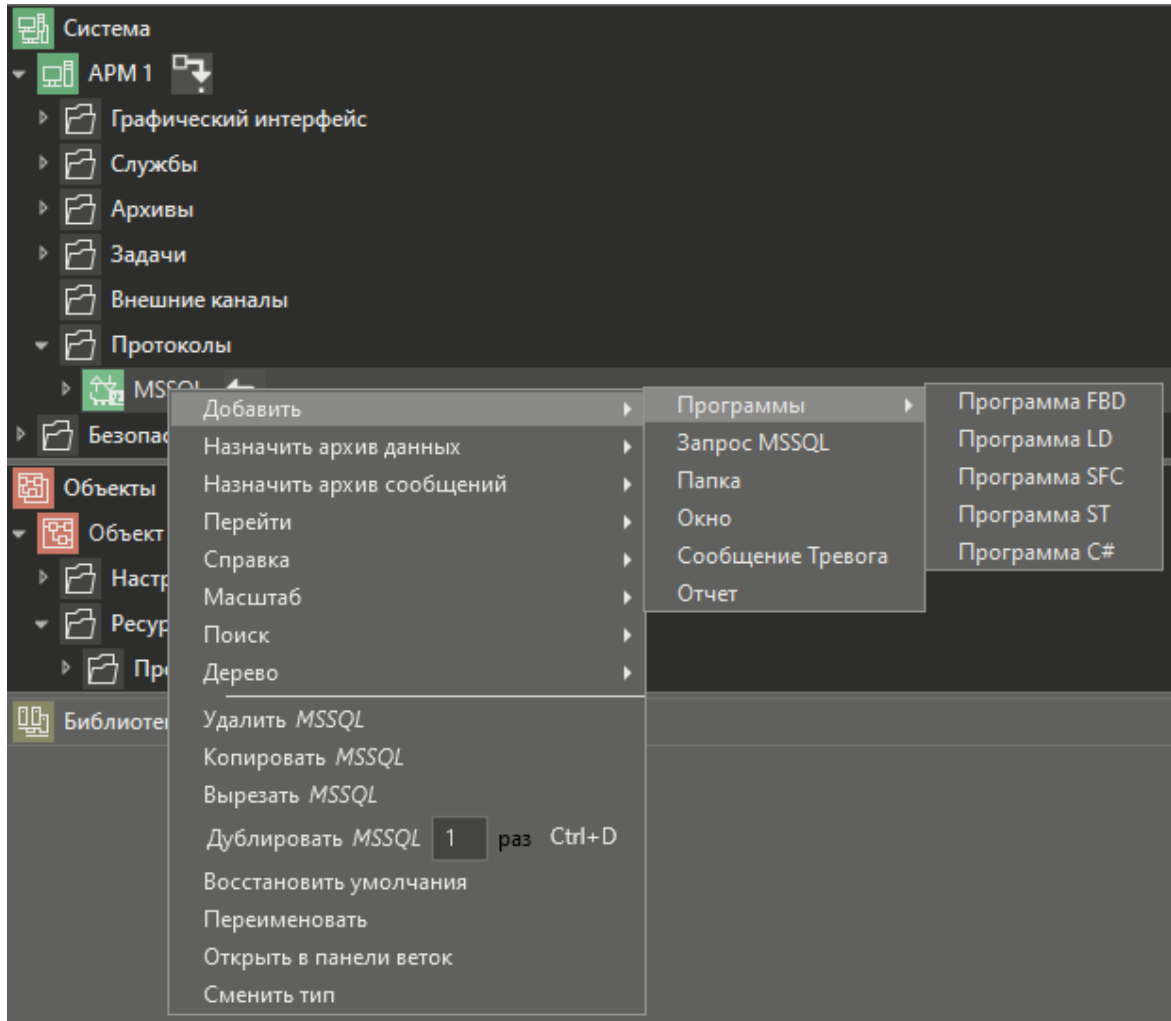
6.4.4.1.2. SQLREQUESTWITHRESULT

Функциональный блок `SqlRequestWithResult` служит типом для экземпляра функционального блока способного выполнять запросы к СУБД и получать ответы.

Чтобы создать экземпляр функционального блока необходимо добавить в пользовательскую библиотеку экземпляр функционального блока `SqlRequestWithResult`. Затем добавить в него нужное количество выходов и входов, а также задать тип результата (должен быть массив структур).

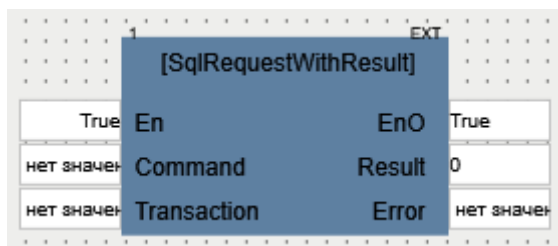
Важно! Экземпляры ФБ, сделанные на основании `SqlRequestWithResult`, можно использовать только в программах внутри протоколов SQL (MSSQL, PostgreSQL, MySQL).

Чтобы добавить программу в протокол необходимо воспользоваться контекстной панелью или контекстным меню:



Затем в программу вставить функциональный блок, сделанный на базе SqlRequestWithResult из палитры.

Вид функционального блока SqlRequestWithResult в редакторе FBD по умолчанию:



Для корректной работы необходимо определить входы функционального блока.

Название	Тип	Назначение
Command	STRING	Вход ФБ. Задается текст SQL-запроса. Формат записи такой же как и в свойстве Команда элемента Запрос

Название	Тип	Назначение
Transaction	Protocols.SQLTransaction	<p>Вход ФБ. Определяет порядок работы функционального блока. Возможные значения:</p> <p>None - значение по умолчанию один вызов ФБ одна транзакция</p> <p>Start - открывает транзакцию, до тех пор пока значение не сменится на Commit или Rollback все вызовы функционального блока будут идти в одной транзакции. Так же это блокирует подключение к БД до тех пор пока транзакция не завершится.</p> <p>Commit - положительно завершает транзакцию и записывает изменения в БД. Так же освобождает подключение к БД.</p> <p>Rollback - отрицательно завершает транзакцию и откатывает все запросы выполненные за время транзакции. Так же освобождает подключение к БД.</p>
Result	ANY	<p>Выдается результат выборки в виде массива структур. Тип выхода нужно переопределить в наследнике, чтобы он соответствовал запрашиваемым данным</p>
Error	STRING	<p>Выход ФБ. Выдается текст ошибки.</p>

Тип данных SQLTransaction, находится также в библиотеке Protocols

Функциональный блок выполняется синхронно.

Рекомендации по работе с функциональным блоком

Если ФБ SqlRequestWithResult вызывается в циклах программ, то в теле цикла необходимо обязательно задавать выход из него, используя условие: `if SqlRequestWithResult_1.EnO=false then exit end_if`

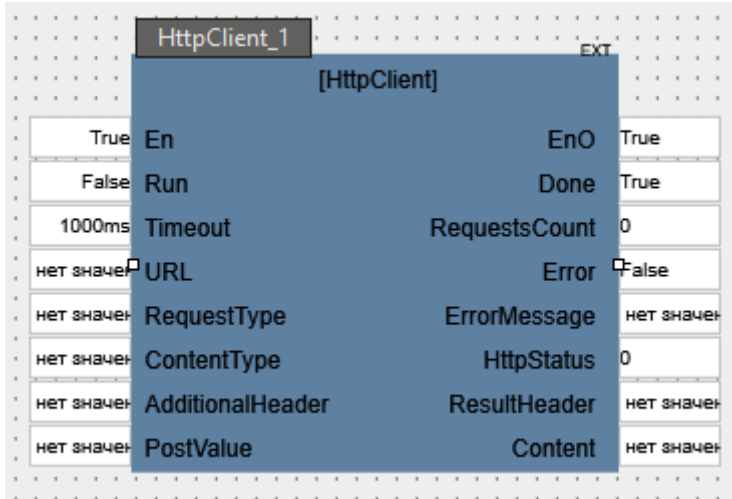
Если в момент вызова функционального блока не все его входы будут определены, то в лог-файле исполнительной системы будет сформирована ошибка запроса.

Функциональный блок выполняется синхронно.

6.4.4.1.3. HTTPCLIENT

Функциональный блок HttpClient предназначен для выполнения HTTP-запросов (версия HTTP 1.1/HTTPS) типа Get и Post. Функциональный блок находится в библиотеке Protocols

Вид функционального блока в программе FBD:

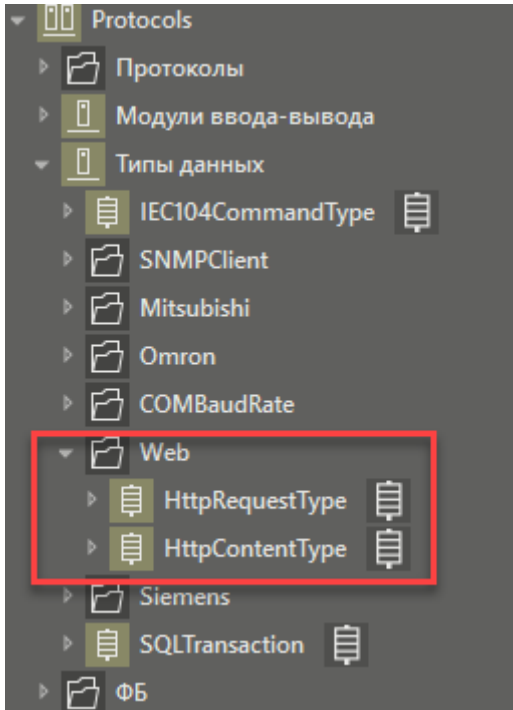


Описание входов и выходов:

Название	Тип	Назначение
Входы		
Run	BOOL	Запускает выполнение запроса по переднему фронту. Если предыдущий запрос еще не выполнен, новый ставится в очередь
Timeout	TIME	Время ожидания запроса
URL	STRING	Указывается адрес сервера
RequestType	Protocols.HttpRequestType	Указывается тип запроса: Get или Post
ContentType	Protocols.HttpContentType	<p>Определяет тип содержимого ресурса. Возможны варианты:</p> <ul style="list-style-type: none"> • Form • JSON • XML • TextPlain

Название	Тип	Назначение
		<ul style="list-style-type: none"> • TextHTML
AdditionalHeader	STRING	Указывается заголовок HTTP-запроса. Нужно прописывать те заголовки, которые могут использоваться в запросах. Например, заголовок Connection определяет, в каком состоянии останется соединение после данного запроса.
PostValue	STRING	Указывается параметр POST-запроса (в JSON-формате).
Выходы		
Done	BOOL	Формируется результат выполнения запроса. После отработки запроса этот выход на один такт принимает значение TRUE.
RequestsCount	INT	Отображается число невыполненных запросов.
Error	BOOL	Сигнализирует о наличии ошибки при выполнении запроса. Если выход принимает значение 0, то это значит, что запрос выполнен без ошибок
ErrorMessage	STRING	Текст ошибки, возникшей при выполнении запроса. Если выход принимает значение по error, то это значит, что запрос выполнен без ошибок
HttpStatus	INT	Выдается HTTP-код выполнения запроса. Список кодов можно найти в открытом доступе в сети Internet.
ResultHeader	STRING	Выдается заголовок HTTP-ответа
Content	STRING	Выдается тело HTTP-ответа. Содержит информацию о ресурсе в формате, заданном на входе ContentType.

Специальные типы данных для данного ФБ находятся также в библиотеке Protocols.

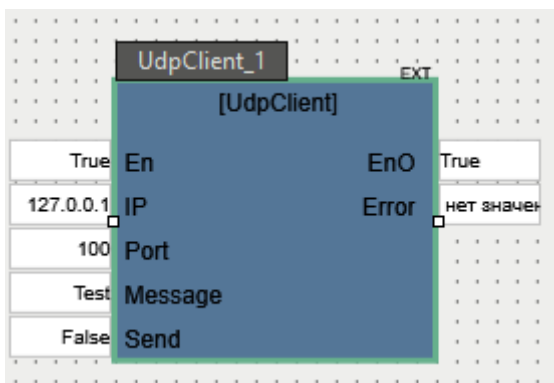


Типы данных отображаются только в полном дереве.

6.4.4.1.4. UDPCLIENT

Функциональный блок UdpClient служит для отправки UDP-сообщений. Функциональный блок находится в библиотеке Protocols

Вид функционального блока в программе FBD:



Параметры ФБ:

Название	Назначение
Входы	
IP	IP-адрес сервера, куда нужно отправить сообщение

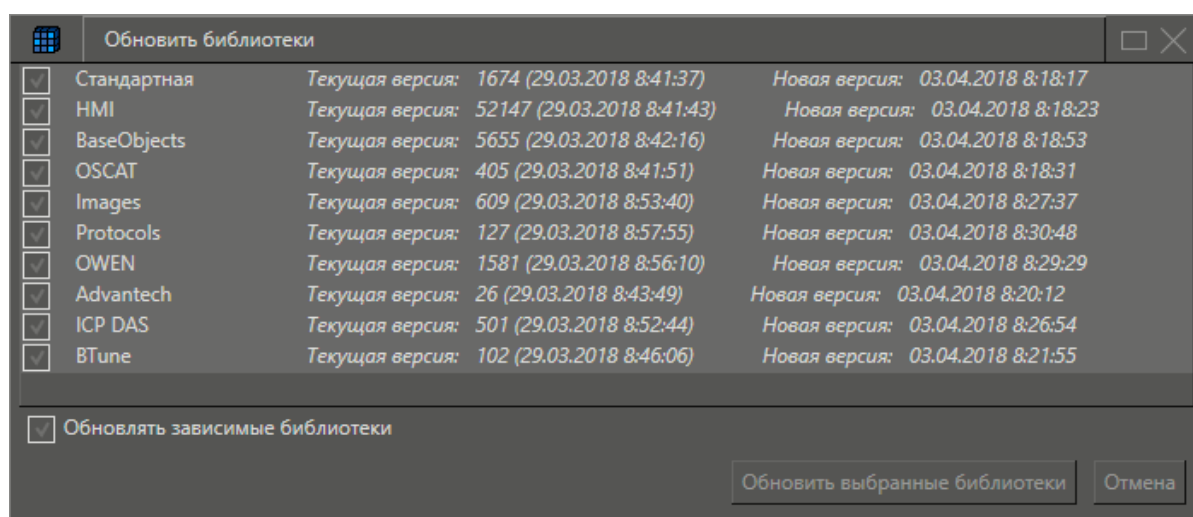
Название	Назначение
Port	Порт сервера, куда нужно отправить сообщение
Message	Текст сообщения.
Send	Управляет отправкой сообщения. Сообщение отправится при изменении входа с False на True. Отправка сообщения идет синхронно, чтобы можно было использовать ФБ в цикле программы ST
Выходы	
Error	Текст сообщения об ошибке, если она произойдет.

6.4.5. ДИАЛОГ ОБНОВИТЬ БИБЛИОТЕКИ

Для открытия данного диалога предусмотрена команда Проверить версии библиотек

После установки более новой версии MasterSCADA 4D или обновления библиотек этот диалог может автоматически отображаться на экране при открытии проекта.

Диалог отображает таблицу параметров библиотек, доступных для обновления:



Столбец выбора библиотеки для обновления

Содержит флаг разрешения обновления библиотеки.

Столбец текущей версии библиотеки

Содержит индекс и время создания библиотеки в проекте.

Столбец новой версии библиотеки

Содержит время создания новой версии библиотеки.

Кнопка Обновить выбранные библиотеки

Команда обновления выбранных библиотек в проекте. По этой команде обновляются те библиотеки, для которых в крайнем левом столбце установлен флаг разрешения этой операции (если установлен флаг Обновлять зависимые библиотеки, то обновляются также библиотеки, зависящие от выбранных).

Флаг Обновлять зависимые библиотеки

Если этот флаг установлен, то библиотеки, подключенные к данному проекту, на которые ссылается одна из выбранных для обновления библиотек, также обновляются.

Кнопка Отмена диалога обновления

Команда отмены обновления библиотек.

6.5. ПАРАМЕТРЫ

Параметры – это переменные проекта. Параметры служат для передачи данных между элементами, например, от одного объекта к другому, от объекта к свойству графического элемента в окне и т.п.

Параметры могут быть добавлены в большинство элементов проекта:

- Узел
- Объект
- Канал
- Тег
- Программу
- Окно
- Тревогу

Важно! В дерево системы и в дерево объектов могут добавляться только экземпляры библиотечных программ (функциональных блоков) и экземпляры тревог, тип отношений которых Ссылается. Поэтому параметры могут добавляться только в типы этих элементов в дереве библиотек.

Механизм добавления параметров везде одинаковый: через пункт контекстного меню элемента или клеммника Добавить, либо через контекстную панель. При создании параметра автоматически откроется диалоговое окно, в котором можно выбрать тип параметра и задать другие настройки. На любом этапе разработки проекта настройки могут быть изменены в панели свойств Параметра.

В качестве типа параметра могут использоваться типы стандарта МЭК 61131-3, а также другие библиотечные типы, которые входят в состав MasterSCADA 4D. Помимо этого, разработчик проекта может создавать свои собственные типы данных в пользовательских библиотеках.

6.5.1. ПАНЕЛЬ СВОЙСТВ ПАРАМЕТРОВ

Панель свойств параметров имеет вид:

Параметр 1

Общие

Имя	Параметр 1
Программное имя	Параметр 1
Полное имя	Объекты.Объект 1.Параметр 1
Метки	
Комментарий	
Сохранять	Наследуется
Доступ	Чтение/Запись
Начальное значение	0
Тип значения	LREAL

Служебные

Отношения

Разрешения

Всегда отображать в дереве

Архивирование

Архивировать	<input type="checkbox"/>
Шаблон архивирования	Шаблон архивирования
Мертвая зона	0
Запись по изменению	<input checked="" type="checkbox"/>
Минимальный период записи	0ms
Максимальный период записи	0ms

Шкала

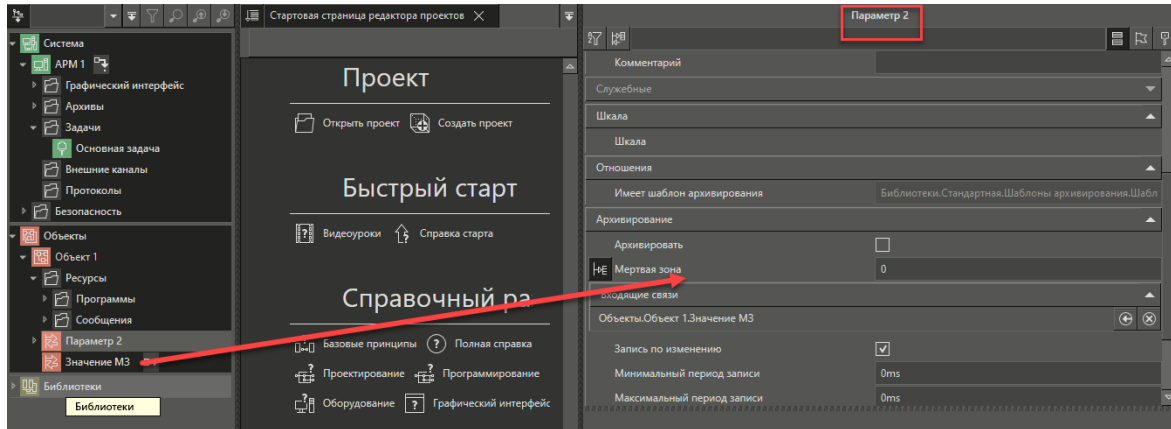
Шкала	Градус
Формат	
Единица измерения	градус
Максимум	100
Минимум	0
HiHi	90
Использовать HiHi	<input checked="" type="checkbox"/>
Hi	80
Использовать Hi	<input checked="" type="checkbox"/>
Lo	20
Использовать Lo	<input checked="" type="checkbox"/>
LoLo	10
Использовать LoLo	<input checked="" type="checkbox"/>
Гистерезис	0
Максимальная скорость изменения	0
Время задержки	0ms

Содержит следующие категории:

Название	Рекомендации
Общие	Наиболее важные свойства в данной категории для параметров: Сохранять, Доступ, Начальное значение, Тип значения. Подробное описание общих свойств приведено в разделе Категория свойств Общие.
Отношения	В данной категории отображаются связи, имеющиеся у параметра в проекте, а также показываются пути к типу параметра в дереве библиотек и путь к шаблону архивирования и шкале, если они назначены для этого параметра.
Разрешения	Описание настроек см. в разделе Категория свойств Разрешения.
Архивирование	Если значение параметра требуется архивировать, то необходимо установить флаг в свойстве Архивировать. В этом случае, значения параметра будут сохраняться в базе данных. При необходимости, можно выбрать шаблон архивирования. При этом появятся настройки, соответствующие данному шаблону, и которые можно переопределить для конкретного параметра.
Шкала	Шкала используется для аналоговых и логических параметров (см. Работа со шкалами).
Параметры	Эта категория свойств присутствует у параметров вида структура и массив. В ней можно задать начальные значения полям структуры и массива.

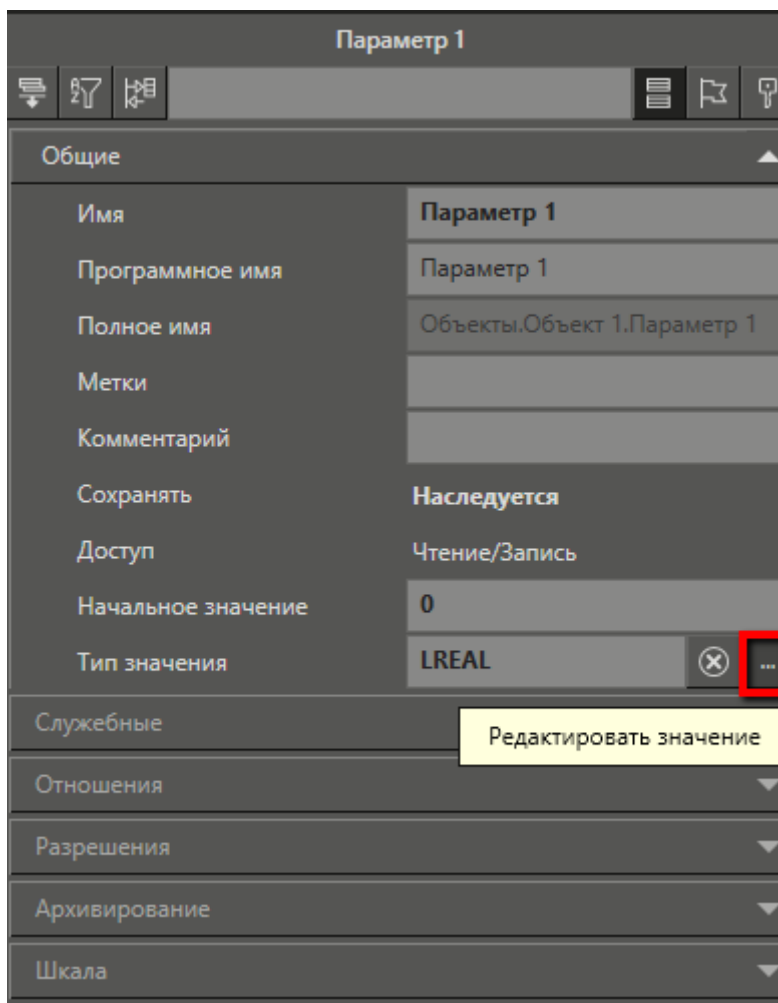
Настройка Мертвой зоны

Свойство Мертвая зона можно изменять в режиме исполнения, для этого нужно перетащить в поле параметр, от которого нужно сделать зависимость:



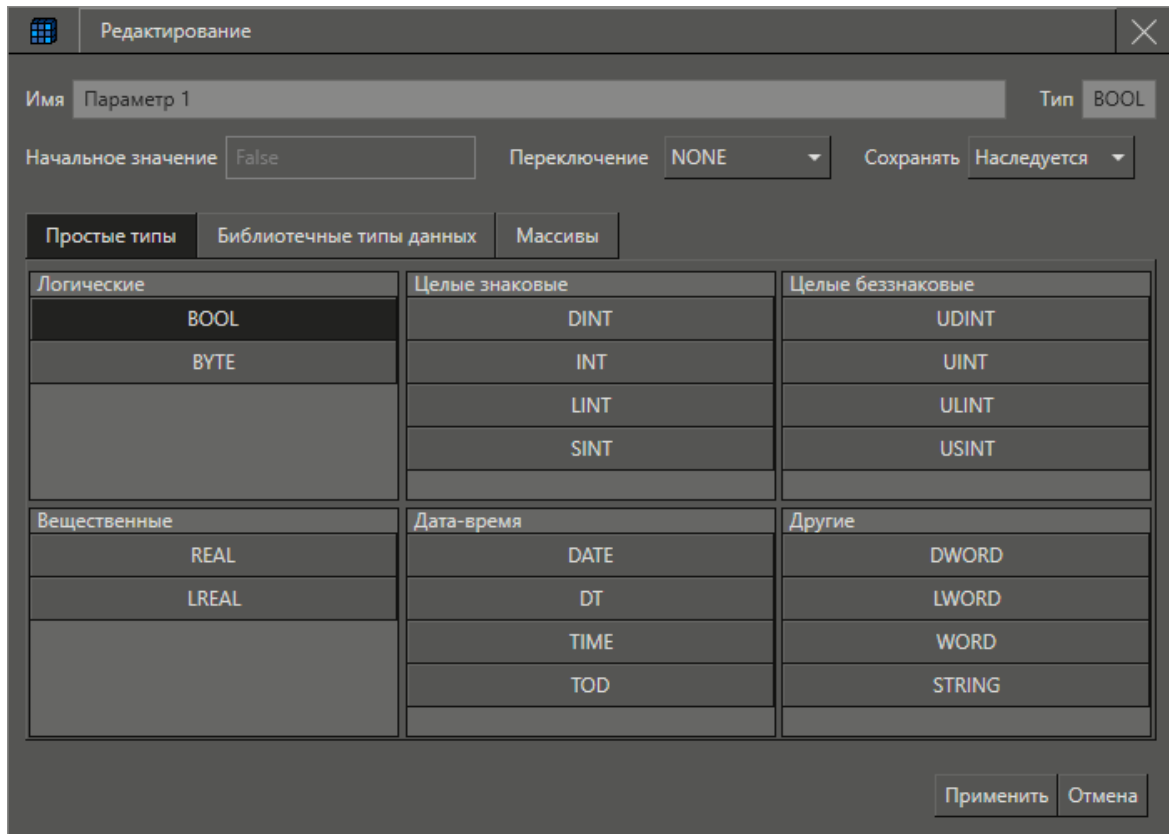
6.5.2. ОКНО НАСТРОЙКИ ПАРАМЕТРА

Окно настройки параметра открывается автоматически при добавлении параметра в проект. В любой момент времени это окно может быть вызвано двойным нажатием левой кнопки мыши на параметре, либо нажатием на кнопку в панели свойств:



В зависимости от вида параметра (параметр объекта, вход программы, параметр канала и т.п.), окно имеет различный набор инструментов.

Вид окна для параметра объекта типа BOOL:



Описание:

Элемент	Описание
Имя	<p>В текстовом поле задается имя параметра. Имя параметра объекта может содержать спецсимволы (" , ; , ? , : и другие), в отличии от параметров узлов и программ</p> <p>Важно! Имена не могут начинаться с цифр.</p>
Тип	Отображается выбранный тип параметра.
Начальное значение	Задается начальное значение параметра, которое будет присвоено ему при старте исполнительной системы.
Доступ	<p>Задается для параметров программ. Определяет доступ к элементу:</p> <ul style="list-style-type: none"> • <code>input</code> - означает, что параметр принимает значение от внешнего источника для использования его при работе программы;

Элемент	Описание
	<ul style="list-style-type: none"> • output - означает, что значение параметра получилось в результате работы программы; • InOut - означает, что значение параметра берется от внешнего источника перед началом работы программы, затем меняется в программе, и может быть отправлено внешнему источнику. Для таких параметров нельзя устанавливать свойство Сохранять в состоянии Да. <p style="text-align: center;">Для локального Параметра программы:</p> <ul style="list-style-type: none"> • Local - значение параметра между вызовами программы будет сохраняться; • Temp - после срабатывания программы значение параметра не сохраняется.
Переключение	Используется только для параметров программ. Определяет поведение параметра типа BOOL.
Сохранять	<p>Определяет, требуется ли сохранять последнее значение параметра, полученное в результате работы среды исполнения, при повторном старте. Может принимать одно из 4 значений:</p> <ul style="list-style-type: none"> • Наследуется - свойство определяется у родительского элемента; • Да - значение параметра сохраняется, настройки задаются в службе Сохранение состояния; <ul style="list-style-type: none"> • Нет - Значение параметра не сохраняется; • Константа - значение параметра не может быть изменено в режиме исполнения; <p>Важно! Свойство Сохранять=Да нельзя устанавливать для параметров программ, у которых свойств Доступ=InOut.</p>
Вкладка Простые типы	В данной вкладке выбирается один из простых типов данных стандарта МЭК 61131-3.
Вкладка Библиотечные типы данных	В данной вкладке можно выбрать один из библиотечных типов, как поставляемых вместе с MasterSCADA 4D, так и созданных пользователем в какой-либо пользовательской библиотеке. Вкладка может содержать все пользовательские типы данных (массивы,

Элемент	Описание
	структуры и др.). Одна строка соответствует одному типу данных, в каждой строке находится имя типа и путь к нему в дереве библиотек.
Вкладка Массивы	В данной вкладке можно назначить параметру тип данных Массив на основе простых и библиотечных типов. При работе с параметрами дерева объектов и дерева системы данная вкладка используется редко. Как правило, сначала создается библиотечный тип данных вида Массив, содержащий в себе все необходимые настройки (количество осей, количество параметров в оси и тип данных,), а затем уже нужный тип выбирается из списка библиотечных типов.

Если в библиотеку были добавлены Справочники, то окно настройки параметра будет содержать еще одну вкладку, описание в разделе Создание типа Справочника

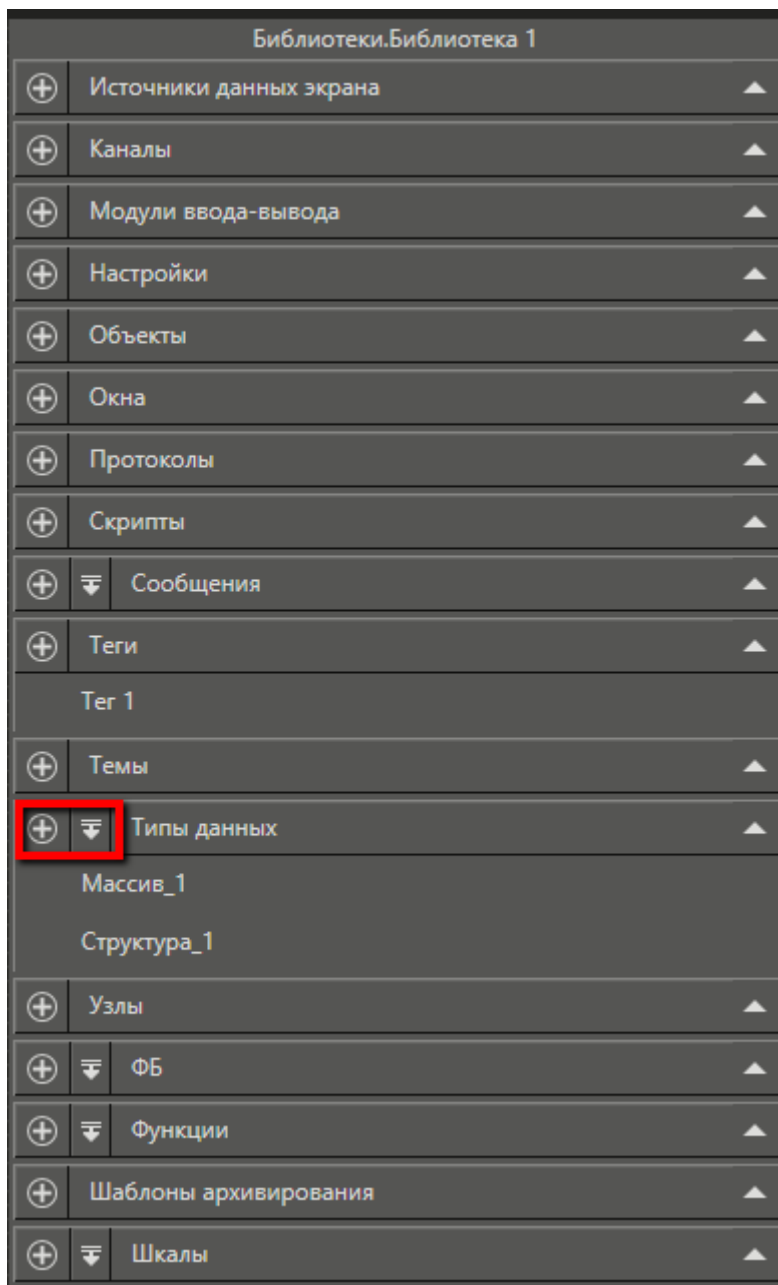
6.5.3. СОЗДАНИЕ НОВОГО ТИПА ДАННЫХ

Разработчик проекта может создавать свои типы данных в пользовательской библиотеке.

Возможные типы данных:

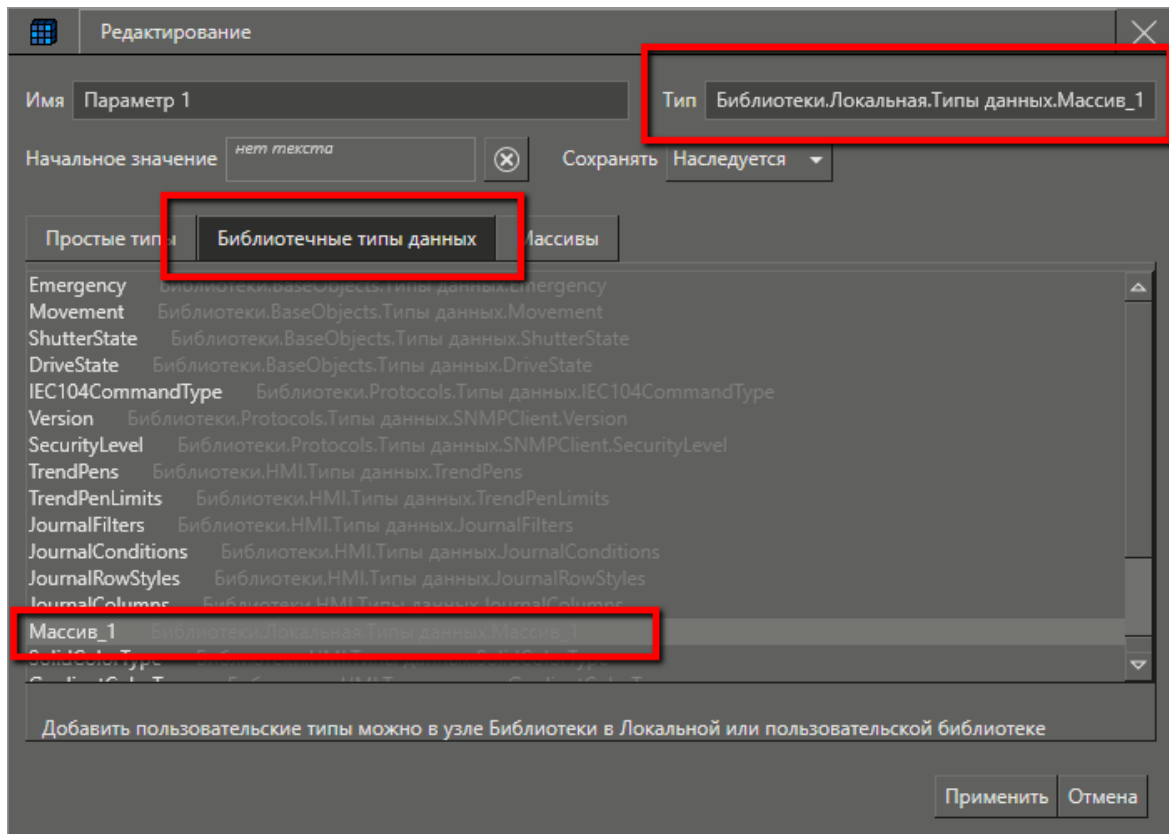
- Массив (набор элементов одного типа с доступом к элементу по индексу);
- Перечисления;
- Диапазон;
- Структура (набор элементов разного типа с доступом к элементу (полю) по имени).

Новый тип данных добавляется при помощи контекстного меню библиотеки, либо ее контекстной панели:



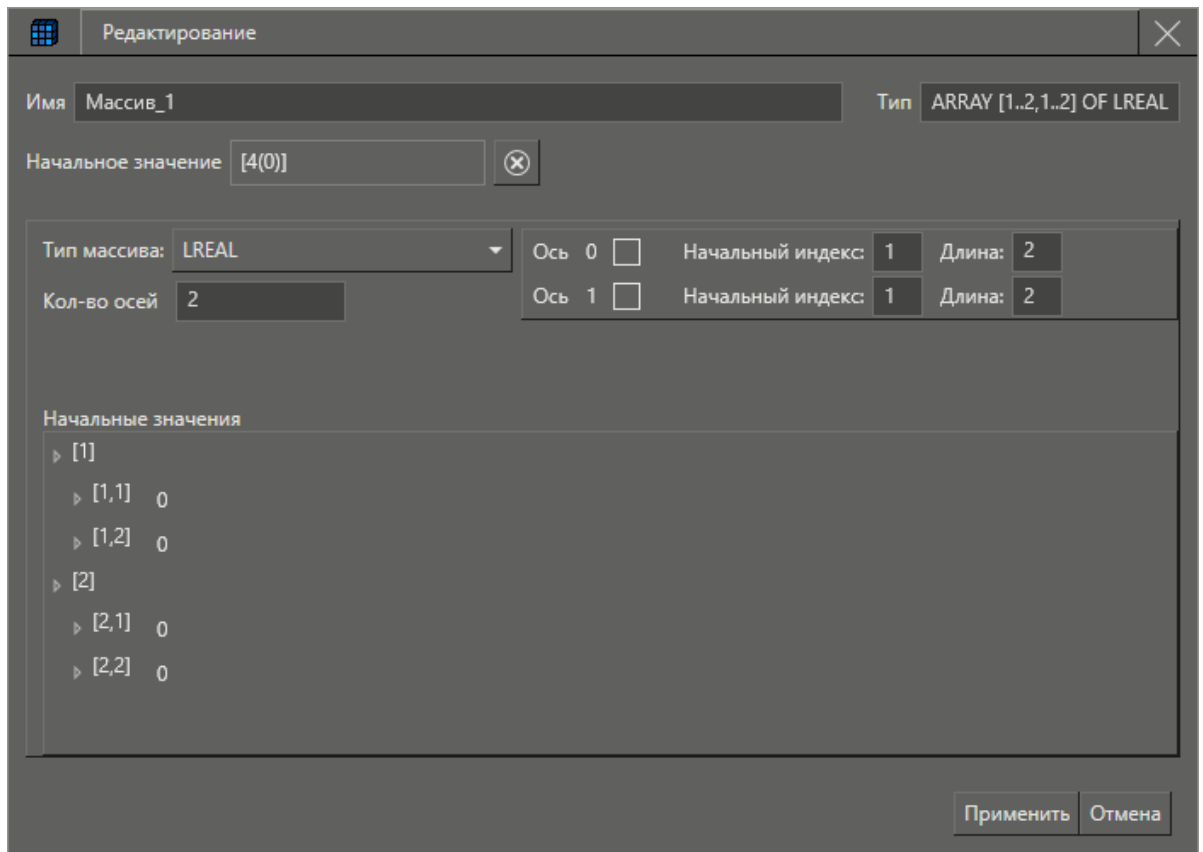
Настройка нового типа данных зависит от его вида.

После добавления нового типа данных, его можно назначить параметру проекта в окне настройки, во вкладке Библиотечные типы данных:



6.5.3.1. НОВЫЙ ТИП ДАННЫХ МАССИВ



После добавления этого типа в библиотеку автоматически откроется окно Настройки типа Массив:



Это окно открывается также при двойном нажатии левой кнопкой мыши на тип Массив, либо при помощи контекстного меню типа Редактировать. Вид окна совпадает с видом вкладки Массивы, а также доступен на вкладке Массивы Окна настройки параметра.

Производный тип данных Массив конфигурируется с помощью инструментов окна:

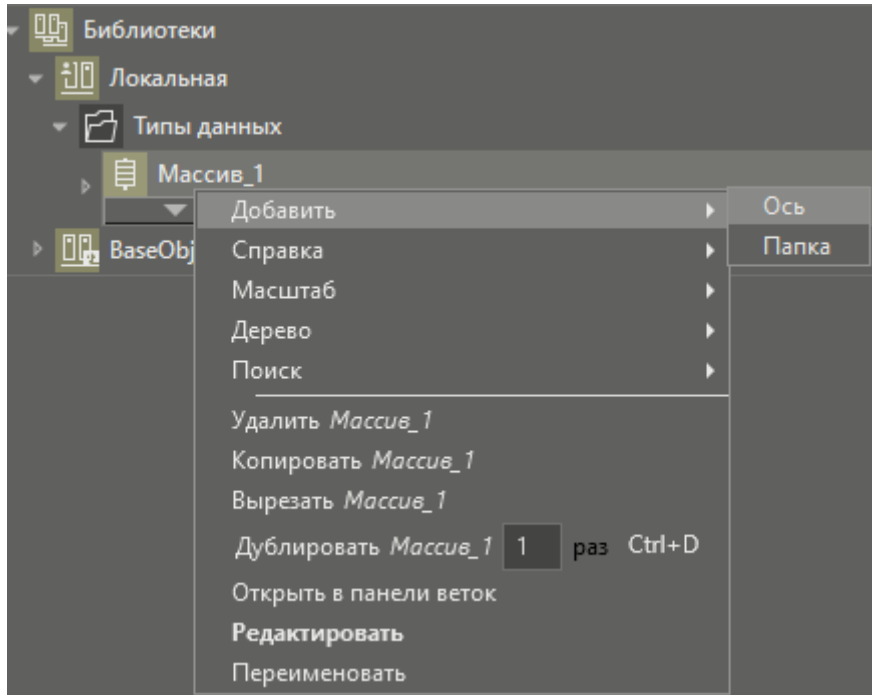
Элемент	Описание
Имя	Задается имя массива
Тип	Формируется тип массива. Данное поле может быть отредактировано вручную (изменения вступят в силу после нажатия на кнопку Применить), либо оно сформируется автоматически, в зависимости от настроек сделанных ниже.
Начальное значение	Позволяет установить значения по умолчанию для всех элементов массива. Формат записи: [A(B)], где: A - обозначает скольким элементам массива, начиная с первого, нужно присвоить значение по умолчанию.

Элемент	Описание
	В - обозначает значение по умолчанию, которое требуется присвоить.
Тип массива	Определяет тип каждого из элементов массива
Кол-во осей	Определяет количество осей (измерений). В зависимости от значения, установленного в данном поле, образуется соответствующее количество строк, определяющих длину каждой оси. Массив может содержать не более трех осей.
<p data-bbox="379 965 587 999">Настройки оси</p> 	<p data-bbox="852 826 1302 860">Поле служит для настройки оси:</p> <ul style="list-style-type: none"> <li data-bbox="799 898 1445 1055">• Флаг - если флаг установлен, то размер оси будет динамическим (т.е. может быть изменен в режиме исполнения). В поле Тип вместо 1..n пишется *, например, ARRAY [*] OF LREAL ; <li data-bbox="799 1077 1422 1155">• Начальный индекс - определяет начальный индекс по оси; <li data-bbox="884 1178 1334 1211">• Длина - определяет длину оси.
Группа Начальные значения	<p data-bbox="687 1245 1469 1368">Позволяет задать индивидуальные начальные значения каждому элементу массива. Для этого нужно нажать левой кнопкой мыши на текущем начальном значении, и появится поле для редактирования:</p> 

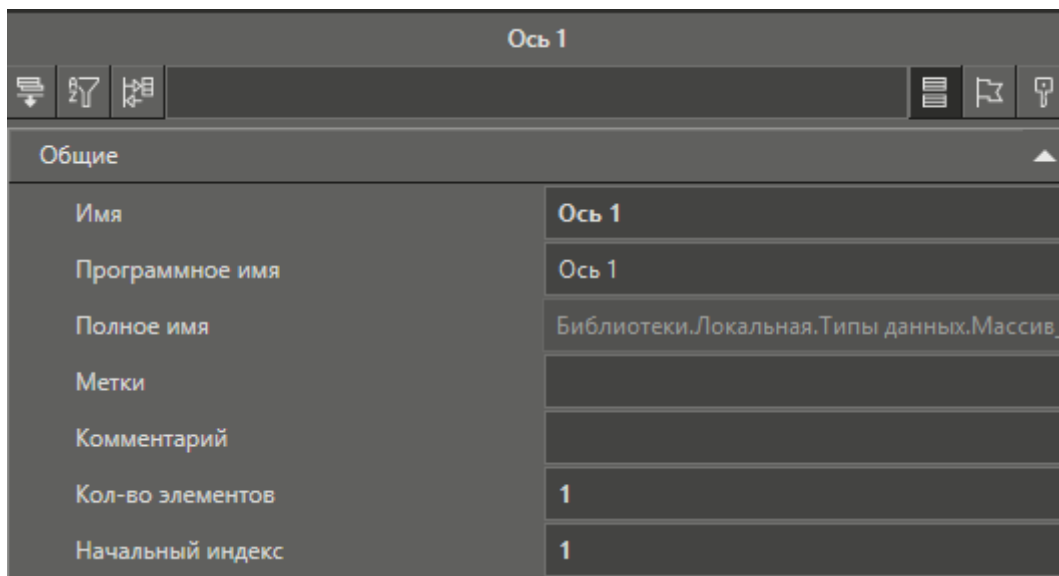
Важно! Для дальнейшей работы с параметрами, которым назначен тип данных Массив с динамическим размером осей, следует использовать специальные функции.

Добавление осей через контекстное меню

Ось можно добавить в массив, используя пункт Добавить-Ось контекстного меню:



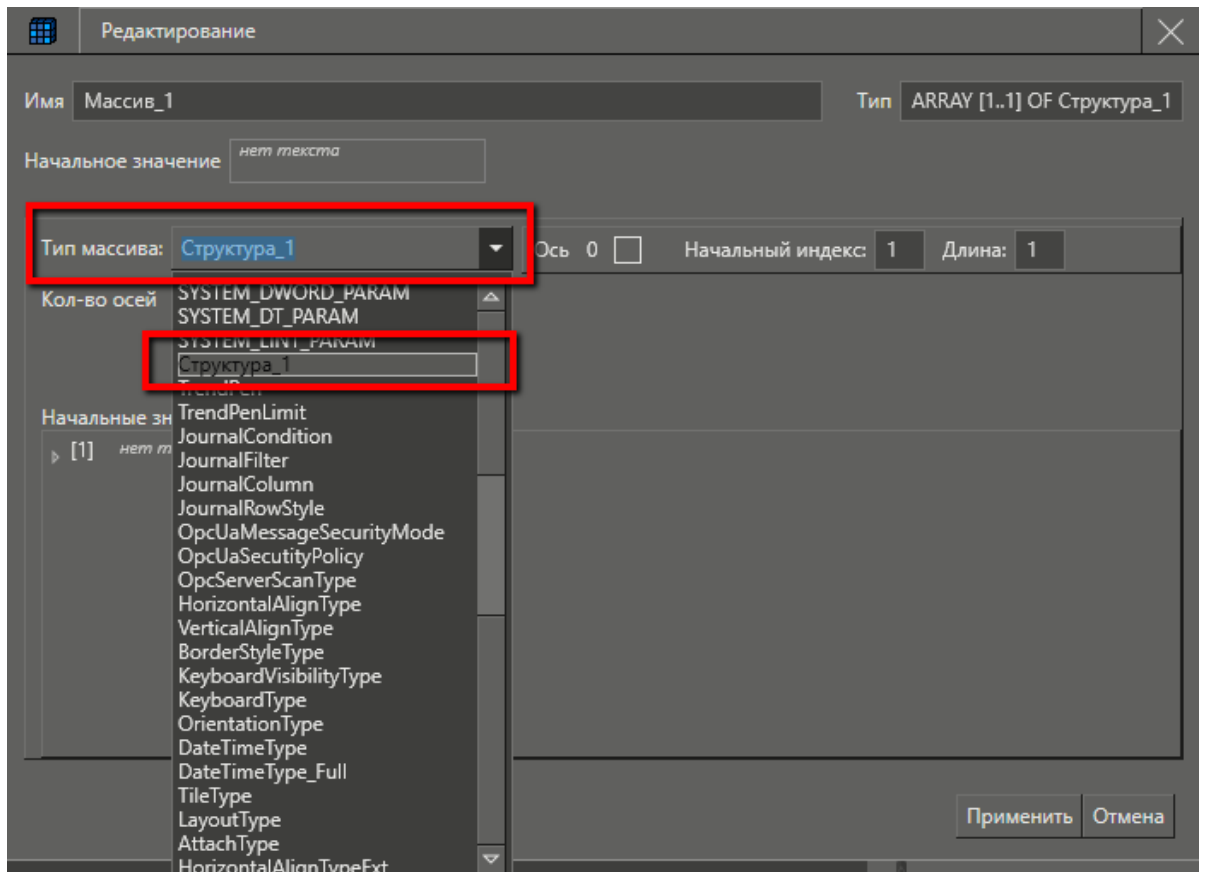
Затем, в панели свойств оси необходимо задать количество элементов и начальный индекс :



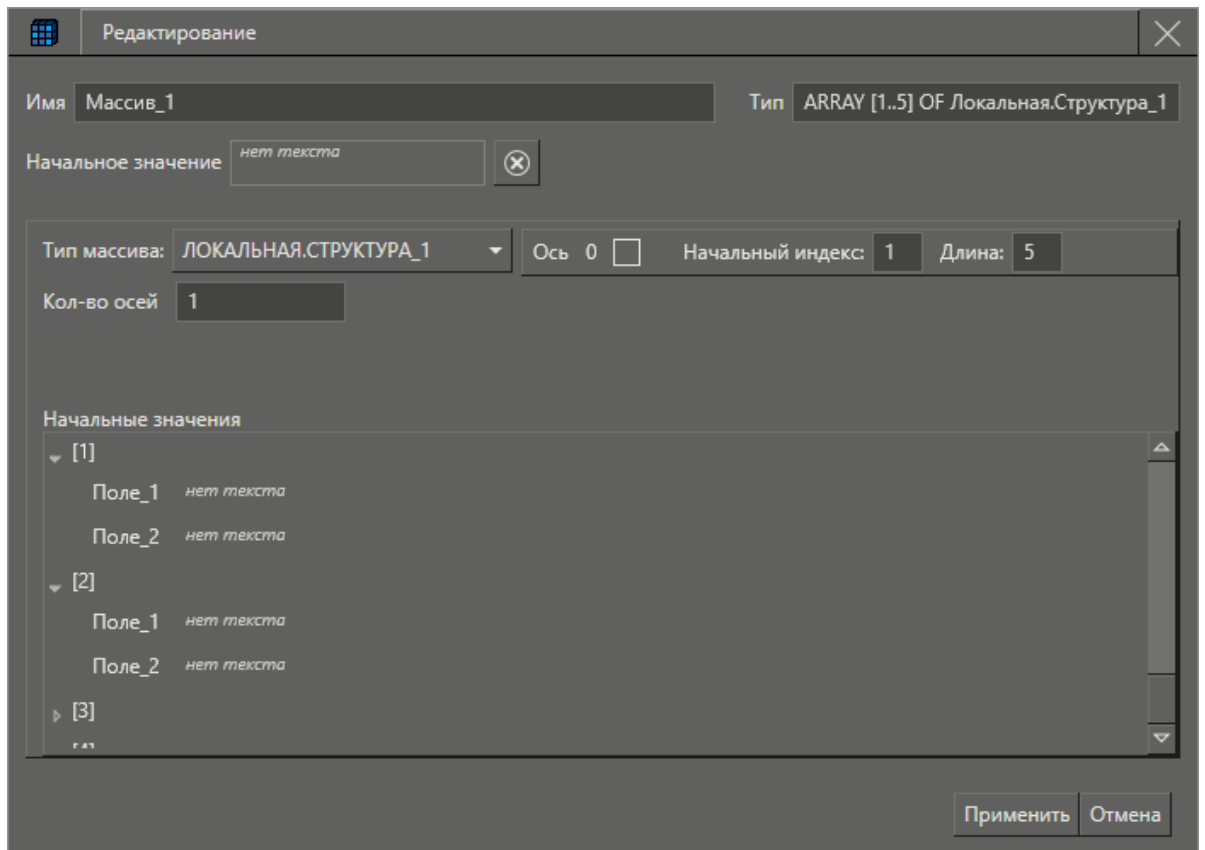
Создание массива Структур

Для того чтобы создать массив структур, сначала необходимо сначала создать новый тип данных Структура в библиотеке.

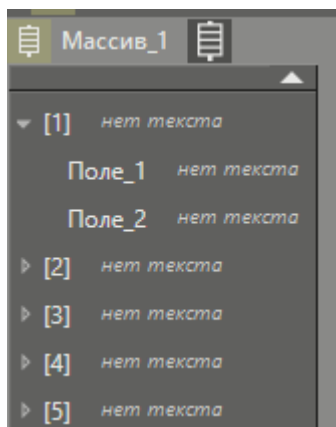
Затем создать новый тип данных Массив, и в окне настройки в поле Тип массива выбрать ранее созданный тип:



После указания других настроек, например длины, получим результат в окне, где можно каждому полю задать начальное значение:



Вид в дереве библиотек:

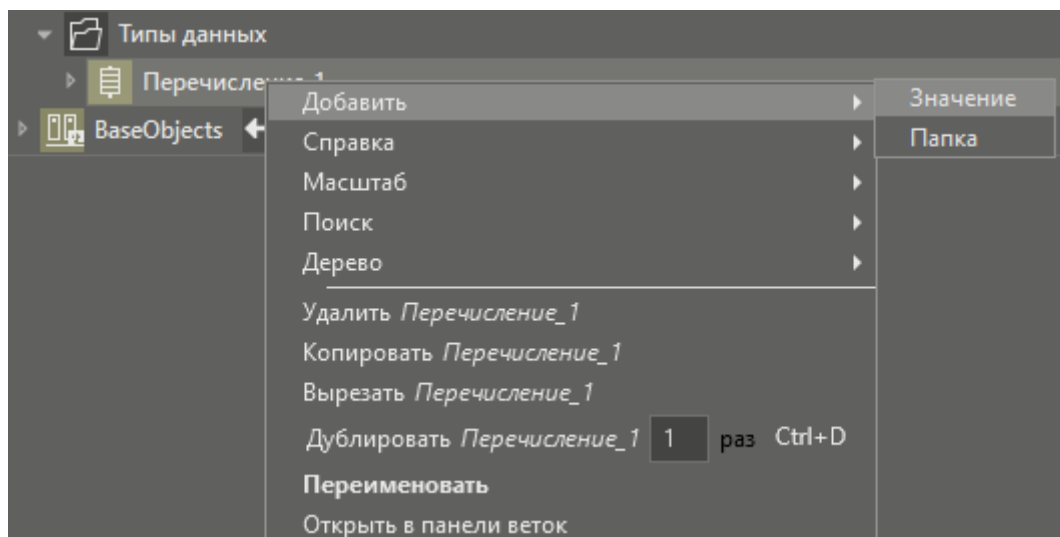


6.5.3.2. НОВЫЙ ТИП ДАННЫХ ПЕРЕЧИСЛЕНИЕ

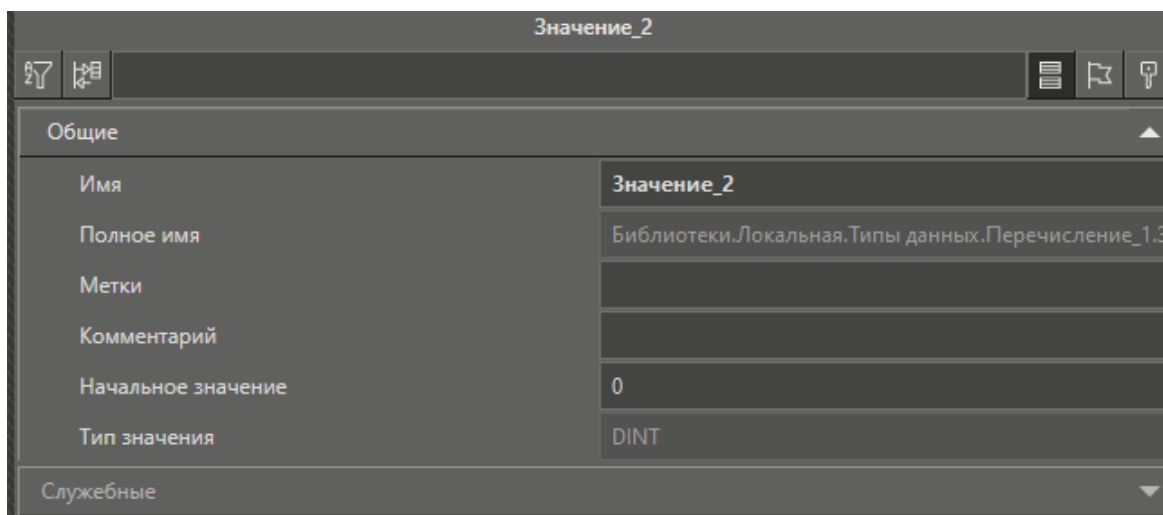
Перечисление используется в случае, когда параметр может принимать только predetermined values.

После добавления нового типа данных в библиотеку, рекомендуется изменить его имя в панели свойств.

Затем, при помощи контекстного меню следует добавить необходимое количество значений:



Затем, в панели свойств задать его имя:



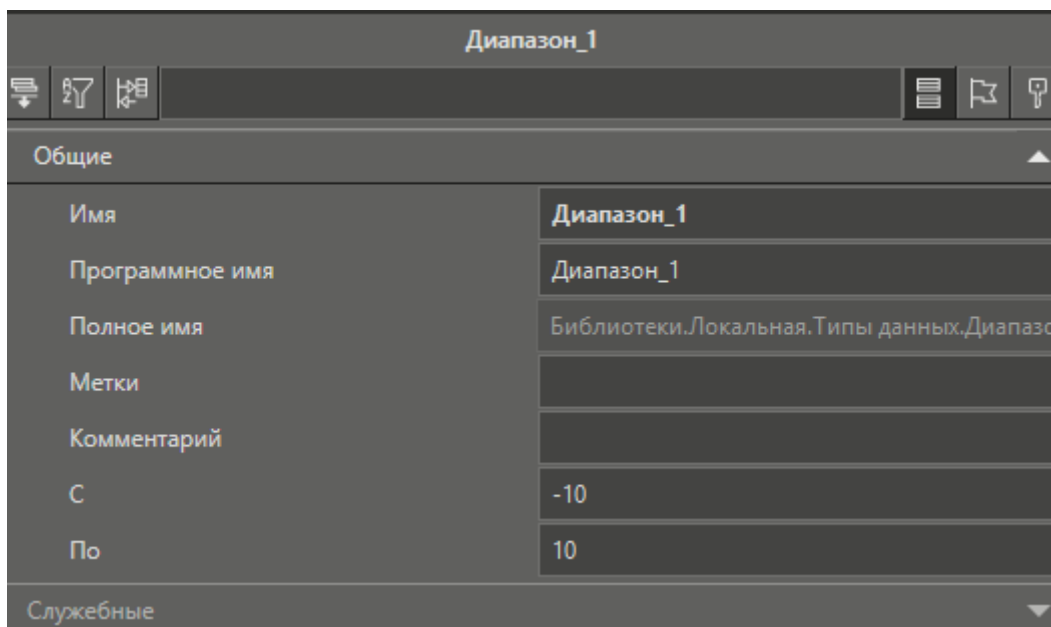
Параметр, которому назначен тип Перечисление сможет принимать значения, которые заданы в поле Имя его дочерних элементов.

Важно! Имя значения Перечисления не может содержать пробелы.

Удалить созданное значение перечисления можно, используя контекстное меню либо кнопку клавиатуры Del.

6.5.3.3. НОВЫЙ ТИП ДАННЫХ ДИАПАЗОН

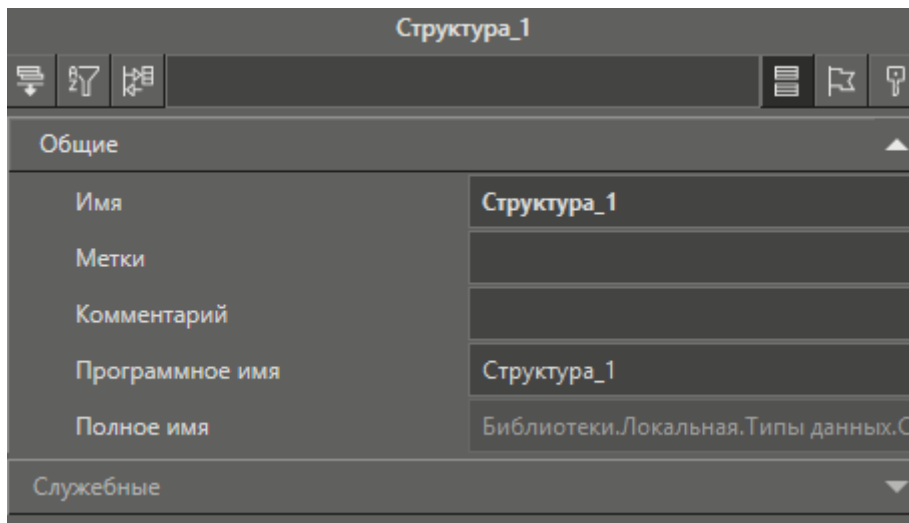
После добавления этого типа данных, необходимо в панели свойств задать его имя и значения диапазона (свойства С и По)



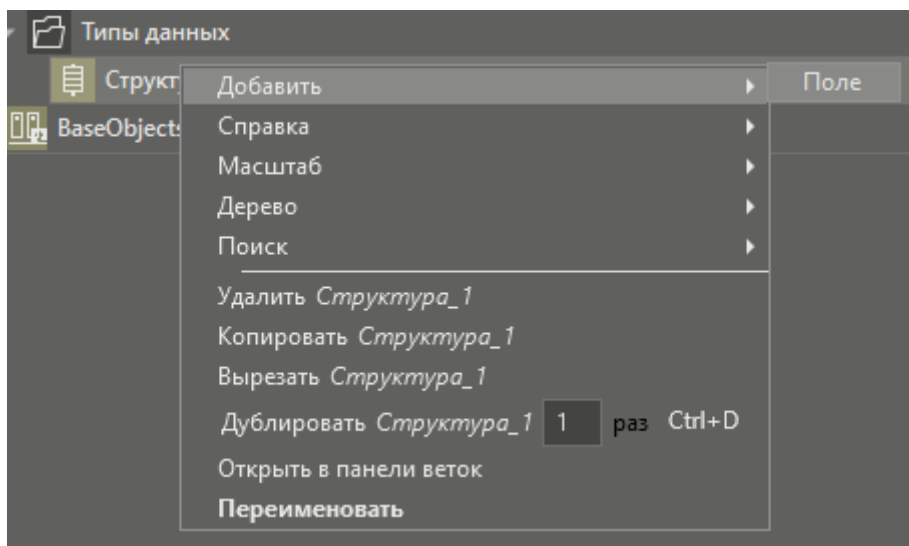
Важно! Значения свойств С и По носят информативный характер. Сообщения о нарушениях границ диапазона выдаваться не будут .

6.5.3.4. НОВЫЙ ТИП ДАННЫХ СТРУКТУРА

После добавления Структуры в библиотеку желательно изменить ее имя по умолчанию, в дереве либо в панели свойств:



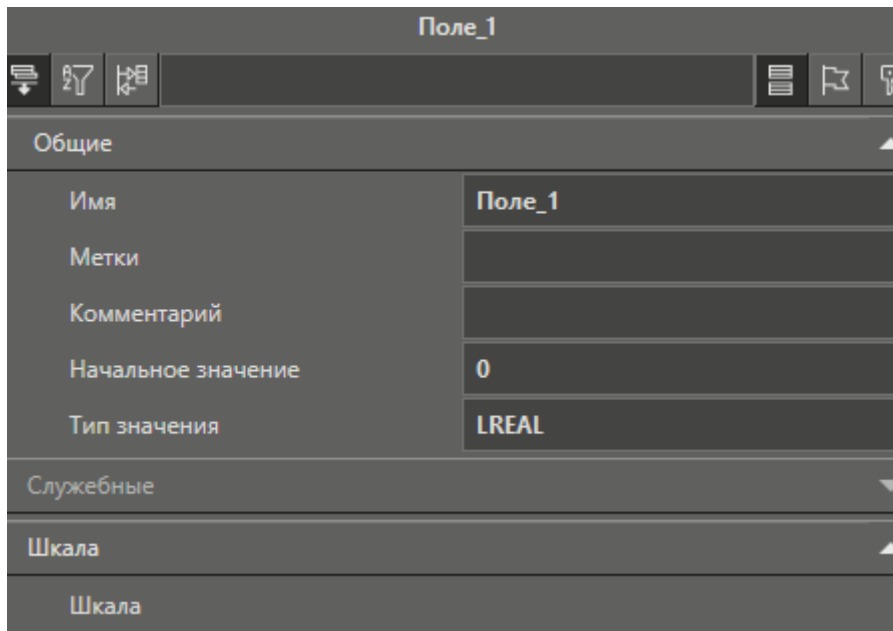
Затем, при помощи контекстного меню добавить требуемое количество Полей :



Автоматически откроется окно Настройки параметров, в котором можно установить все необходимые свойства.

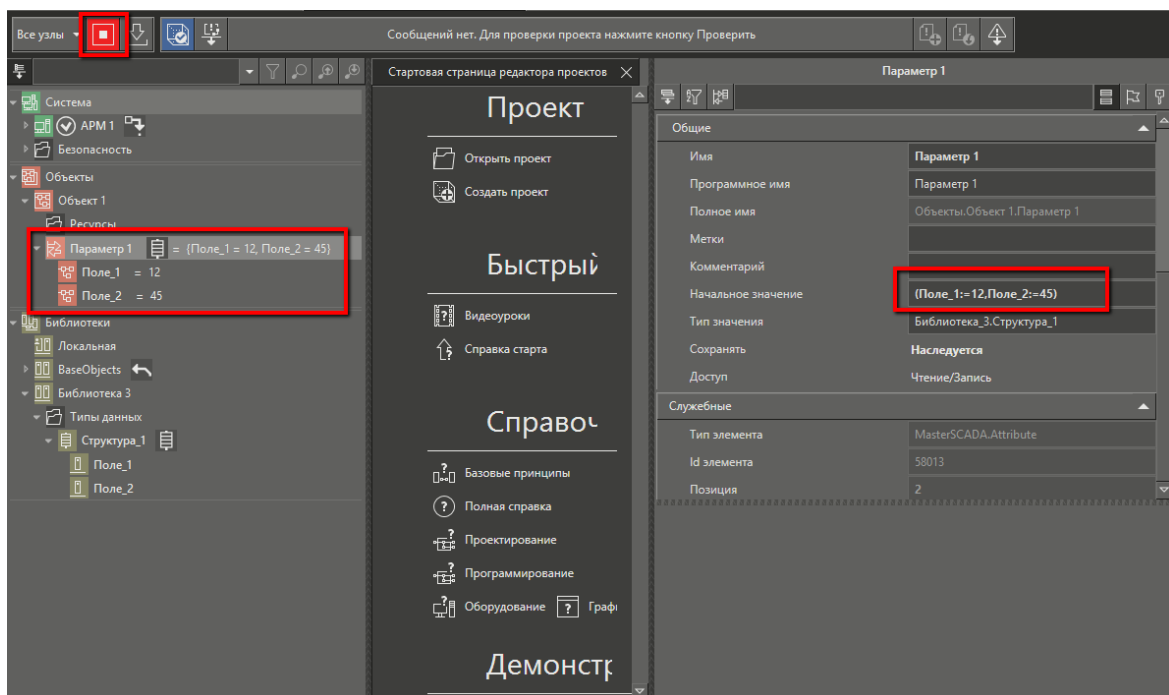
Важно! Имя Поля структуры не может содержать пробелы!

В дальнейшем свойства можно будет изменить либо в панели свойств, либо повторно открыв окно Настройки параметров, и выполнив двойное нажатие на Поле структуры в дереве.



Для полей структуры может быть назначена шкала

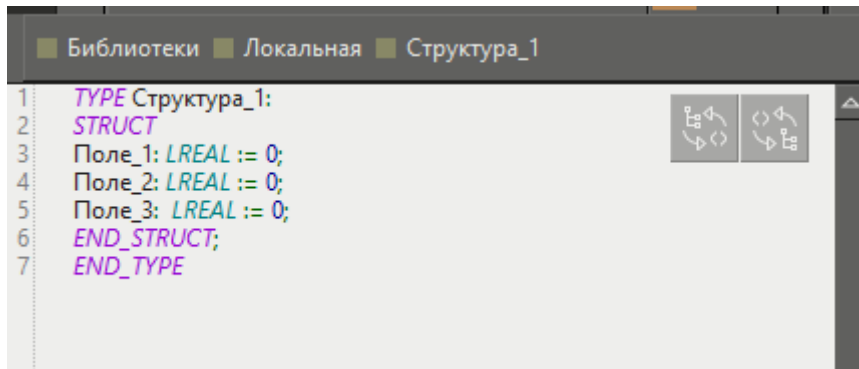
Свойство Начальное значение для полей структуры можно установить в панели свойств параметра, который имеет данный тип в виде $(\{\text{имя поля 1}\}:=\{\text{значение поля 1}\}, (\{\text{имя поля 2}\}:=\{\text{значение поля 2}\}, \dots, (\{\text{имя поля n}\}:=\{\text{значение поля n}\}))$. Пример показан на рисунке:



Важно! При компиляции проекта начальное значение каждого поля будет браться из свойства Начальное значение параметра имеющего тип данных Структура, а если его там нет, то из самого поля структуры.

Текстовый редактор для типов структур

При создании структуры в библиотеке или по двойному клику на типе уже созданной структуры в библиотеке открывается текстовый редактор, в средней части программы, аналогичный вкладке Заголовок в редакторе ST.



```
1  TYPE Структура_1:
2  STRUCT
3  Поле_1: LREAL := 0;
4  Поле_2: LREAL := 0;
5  Поле_3: LREAL := 0;
6  END_STRUCT;
7  END_TYPE
```

Объявить поля структуры можно в данном редакторе.

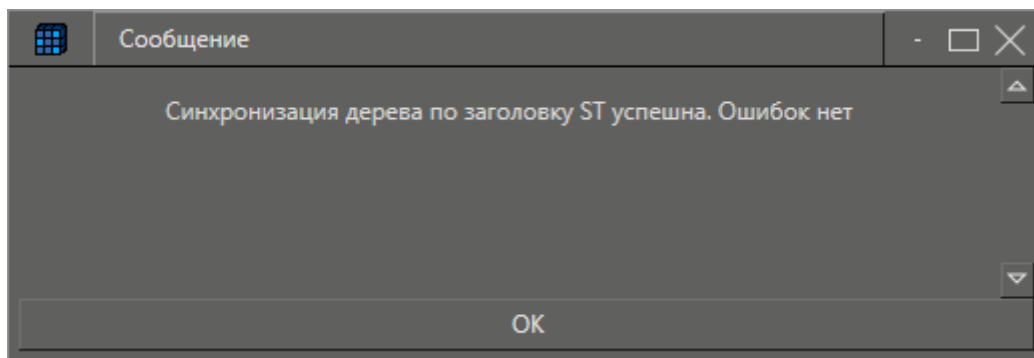
Перед началом редактирования рекомендуется выполнить синхронизацию редактора и дерева, чтобы в редактор добавились поля, созданные в дереве, для этого необходимо нажать кнопку:



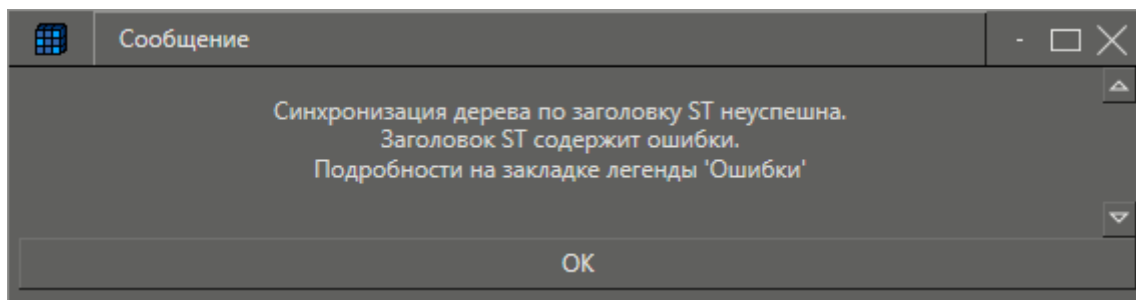
После внесения изменений нужно нажать кнопку синхронизации дерева и редактора, чтобы в дерево папали изменения, сделанные в редакторе:



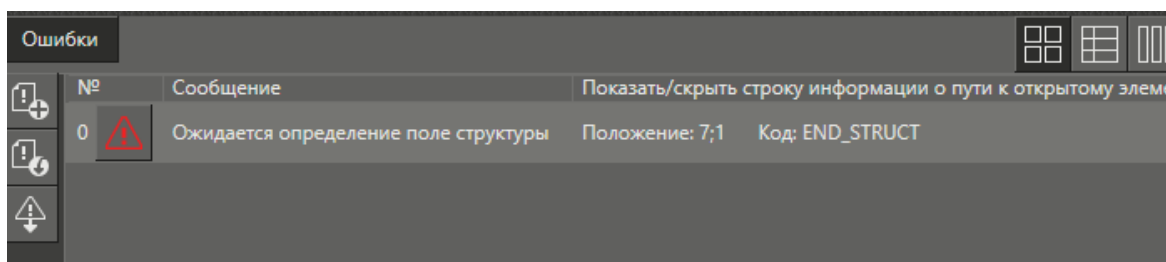
Если синтаксис верный то появится сообщение.



В противном случае появится сообщение об ошибке:



В нижней части редактора отображается вкладка Ошибки:



6.6. РАБОТА СО ШКАЛАМИ

Шкала – это инструмент разработки, позволяющий путем минимальных настроек реализовать контроль параметров, выдачу сообщений, а также настроить отображение в окнах, графиках, отчетах.

Элементы проекта, которым могут быть назначены шкалы:

- Параметр;
- Тег;
- Канал.

При назначении шкалы элементам Тег и Канал настройки наследуются их дочерним параметрам Вход и Выход.

Шкалы могут быть назначены параметрам любого типа данных.

В стандартную библиотеку MasterSCADA 4D входят три типа шкал:

- Шкала AI - используется только для аналоговых параметров;
- Шкала DI - используется только для дискретных параметров;
- Шкала - может быть использована для параметра любого типа данных.

Разработчик проекта может добавить свои типы шкал, которые будут являться экземплярами стандартных.

Для того чтобы назначить элементу шкалу, следует в свойствах в группе Шкала выбрать из выпадающего списка необходимую. Список включает в себя как стандартные шкалы, так и собственные шкалы разработчика.

6.6.1. ШКАЛА AI

Шкала предназначенная для аналоговых параметров дерева объектов, дерева системы, а также для аналоговых полей структур. Позволяет осуществлять: контроль параметров по 4-м границам: верхней и нижней аварийными, верхней и нижней предупредительными, а также контроль скорости изменения.

Если аналоговому параметру (тегу, каналу, полю структуры) назначена шкала, то в его панели свойств появится расширенный список настроек:

Шкала AI

Общие

Служебные

Параметры шкалы

Формат	
Единица измерения	
Максимум	100
Минимум	0
HiHi	0
Использовать HiHi	<input type="checkbox"/>
Hi	0
Использовать Hi	<input type="checkbox"/>
Lo	0
Использовать Lo	<input type="checkbox"/>
LoLo	0
Использовать LoLo	<input type="checkbox"/>
Гистерезис	0
Максимальная скорость изменения	0
Время задержки	0ms
Приоритет HiHi сообщения	750
Приоритет Hi сообщения	500
Приоритет Lo сообщения	500
Приоритет LoLo сообщения	750
Приоритет сообщения о скорости изменения	750
Текст HiHi сообщения	Верхнее аварийное
Текст Hi сообщения	Верхнее предупредительное
Текст Lo сообщения	Нижнее предупредительное
Текст LoLo сообщения	Нижнее аварийное
Текст сообщения о скорости изменения	Максимальная скорость изменения

Описание настроек:

Название	Тип	Описание
Формат	STRING	Влияет на графическое отображение элемента, например, на количество знаков после запятой, которые будут

Название	Тип	Описание
		отображается в элементе Текст. Определяет вид свойства Формат значения.
Единица измерения	STRING	Если назначить единицу измерения, то при размещении параметра на графике, либо в отчете, либо в графическом элементе в окне, обозначение единицы измерения устанавливается автоматически.
Максимум	Соответствует типу параметра	Определяет верхнюю границу изменения значения параметра. Если в окне графического редактора параметр отображается predeterminedными способами, и при этом свойство элемента позволяет настраивать границы, то значение нужного свойства будет установлено автоматически. Например, при добавлении такого параметра на график, ось Y будет по умолчанию иметь указанную границу. Если перетащить параметр в окно в виде Стрелочного прибора, то граница также будет выставлена автоматически.
Минимум	Соответствует типу параметра	Определяет нижнюю границу изменения значения параметра. Если в окне графического редактора параметр отображается predeterminedными способами, и при этом свойство элемента позволяет настраивать границы, то значение этого свойства будет установлено автоматически. Например, при добавлении такого параметра на график, ось Y, будет по умолчанию иметь указанную границу. Если перетащить параметр в окно в виде Стрелочного прибора, то граница также будет выставлена автоматически.
HiHi	Соответствует типу параметра	Если значение параметра превышает заданную величину, то выдается соответствующее сообщение. Сообщение будет считаться активным до тех пор, пока параметр не примет значение меньше, чем величина A, где $A = HiHi - \text{Гистерезис}$. Приоритет сообщения, появляющегося в результате срабатывания границы, равен 750.
Использовать HiHi	BOOL	Если флаг установлен в состояние TRUE, то граница HiHi будет использоваться в работе.

Название	Тип	Описание
Hi	Соответствует типу параметра	<p>Если значение параметра превышает заданную величину, то выдается соответствующее сообщение. Когда сообщение будет считаться активным зависит от настройки проекта Неэсклюзивная модель сообщений шкалы AI.</p> <p>Если используется Эксклюзивная модель сообщений, то сообщение будет активно до тех пор, пока параметр не примет значение меньшее, чем величина A, где $A=Hi-Гистерезис$ или не будет превышена граница HiHi.</p> <p>Если используется Неэсклюзивная модель сообщений, то сообщение будет активно до тех пор, пока параметр не примет значение меньшее, чем величина A, где $A=Hi-Гистерезис$</p> <p>Приоритет сообщения, появляющегося в результате срабатывания границы, равен 500.</p>
Использовать Hi	BOOL	Если флаг установлен в состояние TRUE, то граница Hi будет использоваться в работе.
Lo	Соответствует типу параметра	<p>Если значение параметра опускается ниже заданной величины, то выдается соответствующее сообщение.</p> <p>Если используется Эксклюзивная модель сообщений, то сообщение будет активно до тех пор, пока параметр не примет значение большее, чем величина A, где $A=Lo+Гистерезис$ или не будет нарушена граница LoLo.</p> <p>Если используется Неэсклюзивная модель сообщений, то сообщение будет активно до тех пор, пока параметр не примет значение меньшее, чем величина A, где $A=Lo+Гистерезис$</p> <p>Приоритет сообщения, появляющегося в результате срабатывания границы, равен 500.</p>
Использовать Lo	BOOL	Если флаг установлен в состояние TRUE, то граница Lo будет использоваться в работе.

Название	Тип	Описание
LoLo	Соответствует типу параметра	Если значение параметра опускается ниже заданного значения, то выдается соответствующее сообщение. Сообщение будет считаться активным до тех пор, пока параметр не примет значение, превышающее А, где $A=LoLo+\Gamma$ гистерезис. Приоритет сообщения, появляющегося в результате срабатывания границы равен 750.
Использовать LoLo	BOOL	Если флаг установлен в состояние TRUE, то граница LoLo будет использоваться в работе.
Гистерезис	Соответствует типу параметра	Зона нечувствительности: При возвращении параметра из аварийной или предупредительной зоны будет считаться, что параметр пересек границу, если его значение изменилось на величину А, где: $A= \text{Значение границы}+\Gamma$ гистерезис, для нижних границ, и $A=\text{Значение границы}-\Gamma$ гистерезис, для верхних границ. Влияет на время окончания события, которое вызывает соответствующее сообщение.
Максимальная скорость изменения	Соответствует типу параметра	Если значения параметра изменится за 1 такт на величину большую, чем указана в данном поле, то выйдет соответствующее сообщение. Приоритет сообщения, появляющегося в результате срабатывания границы равен 750
Время задержки	Time	Событие, вызвавшее сообщение о нарушении границ, будет считаться завершенным с задержкой на указанное время.
Приоритет HiHi сообщения	DINT	Определяет приоритет сообщения. По умолчанию значение 750
Приоритет Hi сообщения	DINT	Определяет приоритет сообщения. По умолчанию значение 500
Приоритет Lo сообщения	DINT	Определяет приоритет сообщения. По умолчанию значение 500

Название	Тип	Описание
Приоритет LoLo сообщения	DINT	Определяет приоритет сообщения. По умолчанию значение 750
Приоритет сообщения о скорости изменения	DINT	Определяет приоритет сообщения. По умолчанию значение 750
Текст HiHi сообщения	STRING	Определяет текст сообщения, который будет формироваться в результате нарушения данной границы.
Текст Hi сообщения	STRING	Определяет текст сообщения, который будет формироваться в результате нарушения данной границы.
Текст Lo сообщения	STRING	Определяет текст сообщения, который будет формироваться в результате нарушения данной границы.
Текст LoLo сообщения	STRING	Определяет текст сообщения, который будет формироваться в результате нарушения данной границы.
Текст сообщения о скорости изменения	STRING	Определяет текст сообщения, который будет формироваться в результате превышения скорости изменения.

Важно! Если назначить шкалу параметру окна или параметру программы, то сообщения при нарушении границы и скорости изменения выдаваться не будут. При отображении такого параметра в окне в predetermined виде от шкалы будет использоваться настройка формата и единицы измерения.

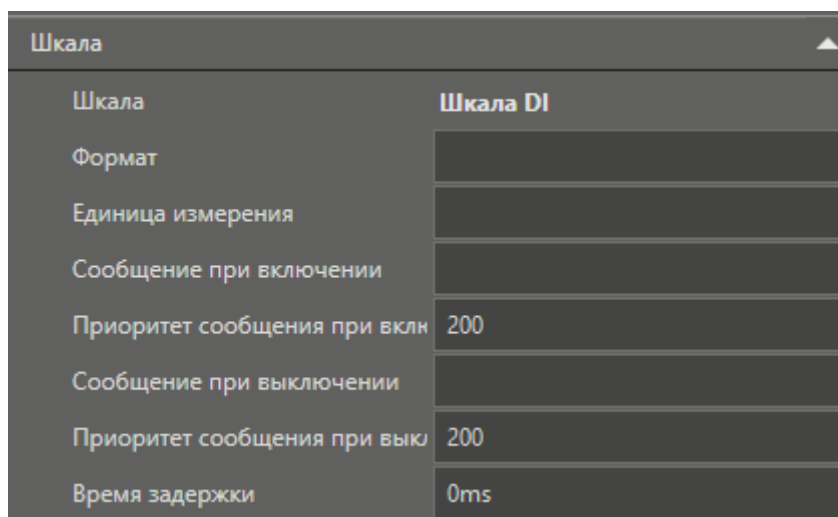
Если после назначения шкалы перетащить параметр в окно, например, в виде инкремента, пера графика и др, то часть настроек элементов окна будут унаследованы от настроек шкалы.

6.6.2. ШКАЛА DI

Шкала предназначена для дискретных параметров дерева объектов, дерева системы, а также для дискретных полей структур. Позволяет формировать сообщения по переднему и заднему фронту изменения значений этих параметров. Шкала DI может быть использована для формирования сообщений на основе архивных данных, полученных от различных

источников, например, в случае если в среду исполнения поступили архивные данные по стандарту OPC UA, IEC 104 и т.п.

Если дискретному параметру (тегу, каналу, полю структуры) назначена шкала, то в его панели свойств появится расширенный список настроек:



Название	Тип	Описание
Формат	STRIN G	Не используется
Единица измерения	STRIN G	Не используется
Сообщение при включении	STRIN G	Определяет текст сообщения при изменении параметра из состояния False в TRUE. Сообщение будет активным до тех пор, пока параметр не изменит свое состояние из TRUE в False.
Приоритет сообщения при включении	DINT	Определяет приоритет сообщения при включении. В дальнейшем это значение можно использовать для фильтрации сообщений в Журнале, Индикаторе сообщений, в ФБ EventsCounter.
Сообщение при выключении	STRIN G	Определяет текст сообщения при изменении параметра из состояния TRUE в False. Сообщение будет активным до тех пор, пока параметр не изменит свое состояние из False в TRUE.

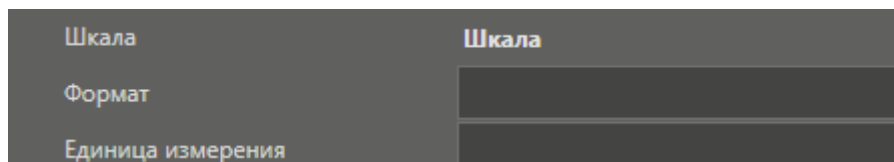
Название	Тип	Описание
Приоритет сообщения при выключении	DINT	Определяет приоритет сообщения при выключении. В дальнейшем это значение можно использовать для фильтрации сообщений в Журнале, Индикаторе сообщений, в ФБ EventsCounter.
Время задержки	TIME	Событие, вызвавшее сообщение, будет считаться завершенным с задержкой на указанное время.

Важно! Если назначить шкалу параметру окна или параметру программы, то сообщения выдаваться не будут.

6.6.3. ШКАЛА

Может использоваться у любых типов параметров. Влияет только на отображение параметров в окнах, отчетах и т.п.

Если параметру (тегу, каналу) назначена шкала, то в его панели свойств появится расширенный список настроек:



Название	Тип	Описание
Формат	STRING	Влияет на графическое отображение элемента, например, на количество знаков после запятой, которые будут отображаться в элементе Текст. Определяет вид свойства Формат значения.
Единица измерения	STRING	Если назначить единицу измерения, то при размещении параметра на графике, либо в отчете, либо в графическом элементе в окне обозначение единицы измерения устанавливается автоматически.

6.6.4. ДИНАМИЧЕСКАЯ ШКАЛА

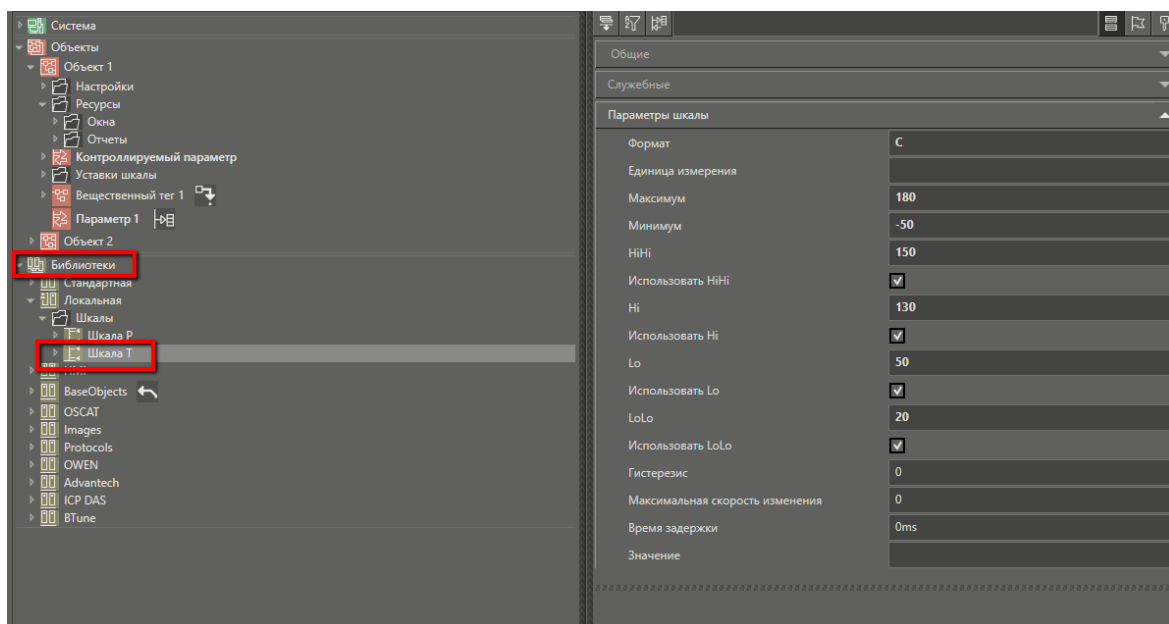
Если необходимо, чтобы значения границ шкал менялись в режиме исполнения, то необходимо установить связь между каким либо параметром и свойством, отвечающим за значения границ.

6.6.5. НОВЫЙ ТИП ШКАЛЫ

Новый тип шкалы создается в случае когда требуется задать разные настройки шкалы по умолчанию.

Например, если в проекте есть параметры, получающие значения и от датчиков температуры, и от датчиков давления, то логично предположить, что числовые значения границ будут различными.

В этом случае, в пользовательской библиотеке необходимо создать два новых типа Шкалы на базе стандартной Шкалы AI:



6.7. СВОЙСТВА

Категория свойств Общие

Категория Задача

Категория Масштабирование

Категория свойств Служебные

Категория свойств Разрешения

6.7.1. КАТЕГОРИЯ СВОЙСТВ ОБЩИЕ

В данном разделе представлены описания настроек, которые могут встречаться у различных элементов в группе Общие панели Свойств.

Название	Описание	Элементы
Доступ	<p>Для Параметра объекта, узла, канала, тега эта настройка определяет доступ к данному параметру для установки внешних связей, т.е. то, какие действия внешние элементы могут производить с данным элементом (в данном случае, внешними считаются элементы, которые находятся в другом объекте).</p> <p>Возможны значения:</p> <ul style="list-style-type: none"> • Чтение - означает, что допускаются только исходящие связи к внешним источникам; • Запись - означает, что допускаются только входящие связи от внешних источников; • Чтение/Запись - означает, что допускаются и входящие, и исходящие связи с внешними источниками. <p>Для элемента Канал возможные значения:</p> <ul style="list-style-type: none"> • input - означает, что в канале будет только один параметр, свойство Доступ которого - Чтение . Используется в случае когда канал получает данные от устройства; • output - означает, что в канале будет только один параметр, свойство Доступ которого - Запись. Используется в случае когда канал записывается данные в устройство; • InOut - означает, что в канале будет два параметра: один используется для чтения из устройства, а второй для записи. <p>Для Параметра программы:</p> <ul style="list-style-type: none"> • input - означает, что параметр принимает значение от внешнего источника для использования его при работе программы; 	Параметр, Канал, Переменная программы

Название	Описание	Элементы
	<ul style="list-style-type: none"> • output - означает, что значение параметра получается в результате работы программы; • InOut - означает, что значение параметра берется от внешнего источника перед началом работы программы, затем меняется в программе, и может быть отправлено внешнему источнику. Для таких параметров нельзя устанавливать свойство Сохранять в состоянии Да. <p>Для локального Параметра программы:</p> <ul style="list-style-type: none"> • Local - значение параметра между вызовами программы будет сохраняться; • Temp - после срабатывания программы значение параметра не сохраняется. 	
Имя	<p>Название элемента, которое дается ему разработчиком проекта. Для каждого языка, поддерживаемого в среде разработки, можно задать свое название элемента.</p> <p>Имя параметра объекта и самого объекта может содержать спецсимволы (" , ; , ? , : и другие), в отличии от параметров узлов и параметров программ</p>	У всех элементов
Комментарий	Краткое описание элемента, добавляемое разработчиком проекта.	У всех элементов
Место исполнения	<p>Определяет место, в котором будет исполняться объект, программа или функциональный блок: в сервере или в визуальном клиенте (в браузере).</p> <p>Элемент будет исполняться в сервере в том случае, если выбран один из двух вариантов:</p> <ul style="list-style-type: none"> • В задаче родительского объекта – элемент будет исполняться в задаче объекта-родителя. Устанавливается по умолчанию; • В основной задаче узла – если родительский объект будет исполняться в задаче пользователя, 	Программы, ФБ, Объекты

Название	Описание	Элементы
	<p>то при установке этого значения текущий элемент будет исполняться в основной задаче того узла, в который он назначен.</p> <p>Элемент будет исполняться в клиенте, в браузере, если в выпадающем списке выбрано :</p> <ul style="list-style-type: none"> • В задаче экрана – элемент будет исполняться в приложении клиента (браузера). Период исполнения задается в элементе Шаблон экрана. Программы и функциональные блоки могут работать в ограничениями. <p>Поддерживаются связи между параметрами задачи экрана и параметрами сервера (подписка на чтение/запись), т.е. если один объект выполняется в задаче сервера, а другой – в задаче экрана, то связи между параметрами этих объектов отрабатываются в обе стороны.</p> <p>Внешние связи из параметров сервера в параметры задачи экрана передаются в метаданных задачи (JSTask.Metadata.Parameters).</p>	
Метки	<p>Разработчик проекта может создать свои обозначения элементов, записать их в данном поле, и в дальнейшем использовать эти обозначения при поиске элементов.</p>	У всех элементов
Начальное значение	<p>Определяет первоначальное значение параметра при старте среды исполнения. Если среда исполнения запускается впервые, то параметр примет значение, установленное в данном поле. При повторных запусках значение параметра будет зависеть от настройки Сохранять. Если параметр сохраняется, то при повторном старте будет установлено значение, которое было получено при исполнении. Иначе будет установлено значение, заданное в данном поле.</p>	Параметры
Отключить исполнение	<p>Если флаг установлен в состояние TRUE, то объект исполняться не будет. Как правило, данная настройка используется при отладке проекта, когда требуется исключить из работы какие-либо части проекта.</p>	Объект

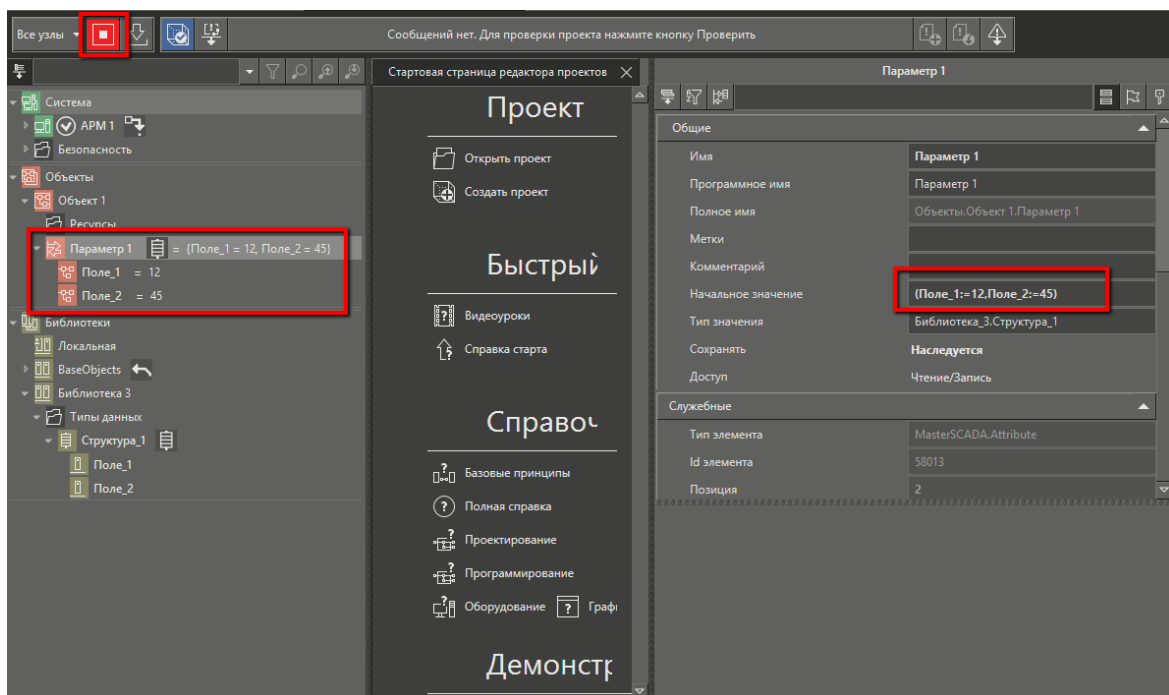
Название	Описание	Элементы
	Например, когда проект уже сделан, но на объекте еще не подключена часть оборудования.	
Полное имя	Показывает путь к элементу в дереве, например, Система.АРМ 1.Протоколы.Modbus TCP.Модуль Modbus TCP 1.AI. Носит информативный характер. Не редактируется.	У всех элементов
Сохранять	<p>Определяет, требуется ли сохранять последнее значение, полученное в результате работы среды исполнения, при повторном старте. Может принимать одно из 4 значений:</p> <ul style="list-style-type: none"> • Наследуется - свойство определяется у родительского элемента; • Да - значение сохраняется, настройки задаются в службе Сохранение состояния; • Нет - значение не сохраняется; • Константа - значение не может быть изменено в режиме исполнения. <p>Подробное описание смотрите в разделе Свойство Сохранять</p> <p>Важно! Свойство Сохранять=Да нельзя устанавливать для параметров программ, у которых свойств Доступ=InOut</p> <p>Важно! Свойство не работает для параметров, тип которых Массив с динамическим размером осей</p>	Параметры, тревоги и параметры тревог, экземпляр объекта, экземпляр ФБ
Способ исполнения	<p>Программы и ФБ могут исполняться постоянно, либо по какому-либо условию, например, по действию оператора, в результате работы другой программы и т.п.:</p> <ul style="list-style-type: none"> • По вызову - исполняется по условию; • Периодически – выполняется периодически (период определяется в задаче узла). 	Программы, ФБ

Название	Описание	Элементы
Тип значения	Определяет тип данных.	Параметры
Отображать элементы в дереве	Определяет способ отображения параметров с типом массив структур. Если флаг установлен, то элементы массива будут представлены в виде дерева. Каждый элемент можно выделить и сделать индивидуальные настройки, например, установить флаг Архивировать	Параметры, Каналы

Примеры настроек

Свойство Начальное значение для полей структуры

Свойство Начальное значение для полей структуры можно установить в панели свойств параметра в виде (`{имя поля 1}:= {значение поля 1}`), (`{имя поля 2}:= {значение поля 2}`),..., (`{имя поля n}:= {значение поля n}`). Пример показан на рисунке:



Свойство Сохранять для экземпляров ФБ

Если экземпляр функционального блока используется в пользовательской библиотечной программе.

6.7.1.1. СВОЙСТВО СОХРАНЯТЬ

Определяет, требуется ли сохранять последнее значение, полученное в результате работы среды исполнения, при повторном старте. Свойство соответствует модификатору RETAIN.

Может принимать одно из 4 значений:

- Наследуется - свойство определяется у родительского элемента;
- Да - значение сохраняется, настройки задаются в службе Сохранение состояния;
- Нет - значение не сохраняется;
- Константа - значение не может быть изменено в режиме исполнения.

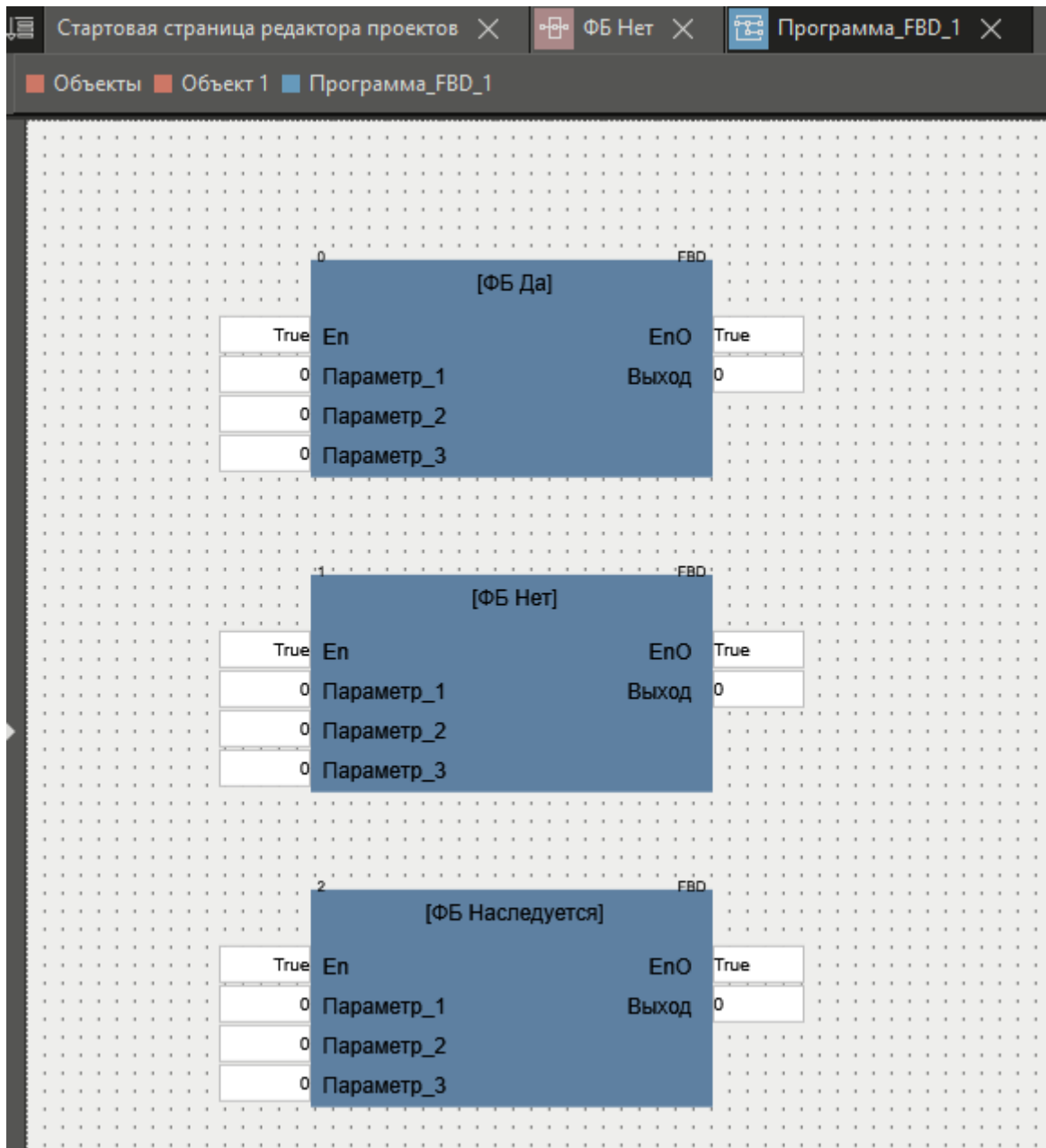
Пусть Программа 1 состоит из трех экземпляров функциональных блоков ФБ Да, ФБ Нет, ФБ Наследуется, каждый функциональный блок содержит три параметра. При создании типов ФБ у всех их параметров установили разные настройки свойства Сохранять: у Параметра 1 = Наследуется, у Параметра 2 =Нет, у Параметра 3= Да

У самих типов были выполнены настройки:

ФБ Да - Сохранять =Да,

ФБ Нет -Сохранять =Нет

ФБ Наследуется - Сохранять = Наследуется



Рассмотрим, какие параметры функциональных блоков будут сохраняться, если в зависимости от значения свойства Сохранять у их родительского элемента Программа_FBD_1

	Да	Нет	Наследуется
ФБ Да	Параметр 1 - Да Параметр 2 -Нет Параметр 3 -Да	Параметр 1 - Нет Параметр 2 -Нет Параметр 3 -Нет	Параметр 1 - Да Параметр 2 -Нет Параметр 3 -Да

	Да	Нет	Наследуется
ФБ Нет	Параметр 1 - Нет Параметр 2 -Нет Параметр 3 -Нет	Параметр 1 - Нет Параметр 2 -Нет Параметр 3 -Нет	Параметр 1 - Нет Параметр 2 -Нет Параметр 3 -Нет
ФБ Наследуется	Параметр 1 - Да Параметр 2 -Нет Параметр 3 -Да	Параметр 1 - Нет Параметр 2 -Нет Параметр 3 -Нет	Параметр 1 - Нет Параметр 2 -Нет Параметр 3 -Да
Вывод	Если у родителя установлено свойство Сохранять = Да, то будут сохраняться все параметры дочерних элементов за исключением тех, у которых выставлено значение Нет	Если у родителя установлено свойство Сохранять=Нет, то независимо от того какие свойства будут установлены для его дочерних элементов сохраняться последнее значение не будет. Для всех элементов будет использоваться модификатор NON_RETAIN	Если у родителя установлено свойство Сохранять = Наследуется, то будут сохраняться параметры только тех дочерних элементов, у которых установлено значение Да,

Такой же принцип сохранения действует и при работе с экземплярами объектов.

Важно! Свойство Сохранять=Да нельзя устанавливать для параметров программ, у которых свойств Доступ=InOut

6.7.2. КАТЕГОРИЯ ЗАДАЧА

Категория задача присутствует у Протоколов и Служб, модулей ввода-вывода, запросов к базам данных

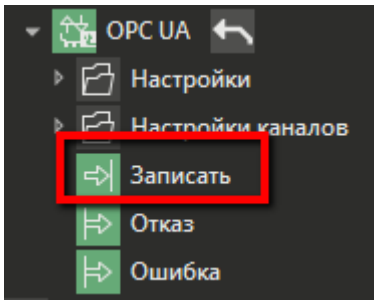
Категория Задача панели свойств у протоколов и служб может содержать все либо часть из приведенных ниже свойств:

Задача	
Период, мс	100
Приоритет	100
Подключение по условию	<input type="checkbox"/>
Выполнение по условию	<input type="checkbox"/>
Выполнять на резервном	<input type="checkbox"/>
Выполнять запись на резервном	<input type="checkbox"/>
Способ записи	По изменению
Формировать отказ при отказе всех модулей	<input type="checkbox"/>
Формировать отказ узла	<input checked="" type="checkbox"/>
Задержка записи при старте опроса (циклов)	0
Задержка сброса отказа на резервном, мс	10000

Имеет следующие настройки:

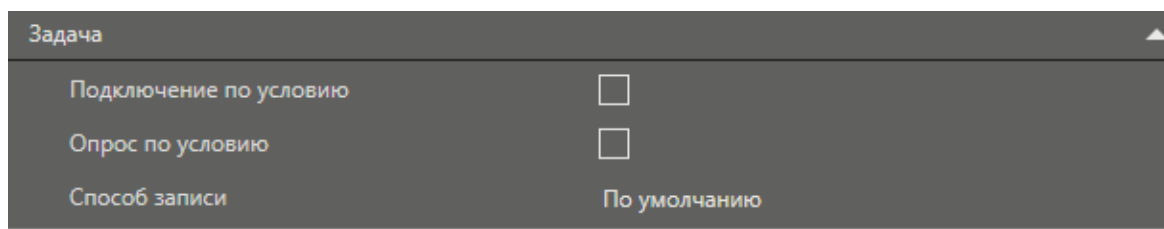
Название	Описание	Служба и Протокол
Период, (мс)	<p>Задается период, с которым будет производиться опрос модулей ввода-вывода. Данный период должен выбираться с учетом возможностей устройств, работающих по данному протоколу.</p>	Все
Приоритет	<p>Задает приоритет протокола. Если работа по протоколу с повышенным приоритетом не укладывается в свой период опроса, то в случае одноядерного процессора в устройстве другие задачи исполняться не будут, а в случае наличия многоядерного процессора они будут исполняться реже. В большинстве проектов эту настройку можно оставить без изменения. Использование ее оправдано только когда необходимо гарантировано обеспечить цикл вычисления</p>	Все

Название	Описание	Служба и Протокол
	наиболее важных частей проекта.	
Подключение по условию	Если флаг установлен, то в дереве протокола появляется параметр Подключить. Если этот параметр связать с каким-либо параметром, тогда если связанный параметр принимает значение TRUE, то начинается опрос с указанным периодом. Если параметр принимает значение False, то опрос прекращается.	Все протоколы, кроме DCON и Modbus. Модули
Выполнение по условию	Если флаг установлен, то в дереве протокола появляется параметр Выполнить. Если данный параметр связать с каким-либо параметром проекта, тогда если связанный параметр принимает значение TRUE, то по переднему фронту изменения значения произойдет единичный опрос.	Все протоколы кроме DCON и Modbus. Модули
Выполнять на резервном	Если данный флаг установлен, то будет выполняться опрос устройств, подключенных по настраиваемому протоколу, узлом, находящимся в состоянии резервного.	Все протоколы
Выполнять запись на резервном	Если флаг установлен, то если узел находится в состоянии резервного, то будут отправляться значения на выходы каналов. Если флаг снят, то будет выполняться только чтение значений.	Все протоколы

Название	Описание	Служба и Протокол
	<p>Значение свойства учитывается, если установлен флаг Выполнять на резервном.</p>	
Способ записи	<p>Определяет способ передачи данных от исполнительной системы протоколу. Возможные значения:</p> <p>По изменению - запись происходит при изменении значения выхода канала</p> <p>По изменению времени - запись происходит при изменении значения или метки времени выхода канала</p> <p>Периодический - запись происходит на каждом цикле</p> <p>По условию - появляется параметр Записать (тип BOOL). Запись всех выходных каналов устройства будет происходить только по переднему фронту изменения значения этого параметра:</p> 	Протоколы, кроме DCON и Modbus
Формировать отказ при отказе всех модулей	<p>Определяет способ формирования выхода Отказ протокола. Если значение свойства равно TRUE, то выход Отказ примет значение TRUE в случае когда отказали</p>	Все протоколы

Название	Описание	Служба и Протокол
	все модули. Это может быть вызвано, например, отказом шины, тогда имеет смысл переключение на резервный сервер.	
Формировать отказ узла	Определяет будет ли осуществлен переход на резервный сервер когда сформировался выход Отказ у данного протокола. Если флаг установлен, то при формировании отказа данного протокола формируется также общий отказ узла, и, в случае резервирования, происходит переключение на резервный сервер.	Все протоколы
Задержка записи при старте опроса (циклов)	Определяет будут ли учитываться первые значения при расчетах при старте системы. При старте сервера или при переключении в режим MASTER заданное этой настройкой число циклов выполняется без записи выходов. Это необходимо для того, чтобы значения выходных параметров начинали выдаваться не сразу, а через некоторое время, когда уже обновятся значения входных параметров и пересчитаются алгоритмы.	Все протоколы
Задержка сброса отказа на резервном	Определяет время (задается в мс), через которое будет снят признак Отказ у протокола для узла, который стал резервным.	Все протоколы

Категория Задача панели свойств у модулей ввода-вывода и запросов к базам данных:



Название	Описание
Подключение по условию	Если данное свойство связать с каким-либо параметром (например, перетащить параметр из дерева на это свойство), тогда если связанный параметр принимает значение TRUE, то начинается опрос с указанным периодом. Если параметр принимает значение False, то опрос прекращается.
Выполнение по условию	Если данное свойство связать с каким-либо параметром (например, перетащить параметр из дерева на это свойство), тогда если связанный параметр принимает значение TRUE, то по переднему фронту изменения значения произойдет единичный опрос.
Способ записи	<p>Определяет способ передачи данных от исполнительной системы протоколу. Возможные значения:</p> <p>По изменению - запись происходит при изменении значения выхода канала</p> <p>По изменению времени - запись происходит при изменении значения или метки времени выхода канала</p> <p>Периодический - запись происходит на каждом цикле</p> <p>По условию - появляется параметр Записать (тип BOOL). Запись всех выходных каналов устройства будет происходить только по переднему фронту изменения значения этого параметра.</p> <p>По умолчанию - используется настройка Способ записи протокола</p>

6.7.3. КАТЕГОРИЯ МАСШТАБИРОВАНИЕ

Масштабирование используется в случае если модули ввода-вывода, работающие по протоколам Modbus, DCON не поддерживают встроенное масштабирование сигналов.

Масштабирование назначается аналоговым каналам. Если настройки масштабирования заданы на уровне протокола или модуля ввода-вывода, то эти настройки будут считаться настройками по умолчанию для их каналов.

Вид категории в панели свойств:

Описание:

Название	Описание
Включить масштабирование	Если будут настроены значения масштабирования, то когда флаг установлен, будет применяться масштабирование.
SourceMin	Минимальное значение в устройстве.
SourceMax	Максимальное значение в устройстве.
TargetMin	Минимальное значение в MasterSCADA 4D.
TargetMax	Максимальное значение в MasterSCADA 4D.

Рассмотрим ситуацию, когда, например, значения в устройстве меняются в диапазоне от 0 до 65535, а нам требуется получить значения в MasterSCADA 4D, меняющиеся в диапазоне от 4 до 20. В этом случае панель свойств будет иметь вид:

Масштабирование	
Включить масштабирование	<input checked="" type="checkbox"/>
SourceMin	0
SourceMax	65535
TargetMin	4
TargetMax	20

6.7.4. КАТЕГОРИЯ СВОЙСТВ СЛУЖЕБНЫЕ

В данном разделе представлены описания настроек, которые могут встречаться у различных элементов в категории Служебные панели свойств. Данные свойства доступны только в том случае, если в редакторе общих настроек установлен флаг Служебные свойства. По умолчанию флаг снят.

Важно! Рекомендуется использовать данную категорию свойств после консультации с технической поддержкой


Состав свойств у разных элементов может отличаться. Служебные свойства Проекта открываются если в дереве не выделен ни один из элементов (для этого нужно нажать левую кнопку мыши на свободном месте в дереве).

Вид категории в панели свойств для элемента АРМ:

Служебные	
Подтип	WIN32
Категория	Controller
Категории родителей	
Программный обработчик	MasterSCADA.Controllers.Adapters.DebugControl
Тип элемента	MasterSCADA.System.Controller
Id элемента	60002
Позиция	0
Порядок	
Id описания	
Запрет наследования	<input type="checkbox"/>
Id элемента внешней библиотеки	
Количество подэлементов	184
Справочный раздел	
Шаблон имени последовательного порта	

Описание часто используемых свойств:

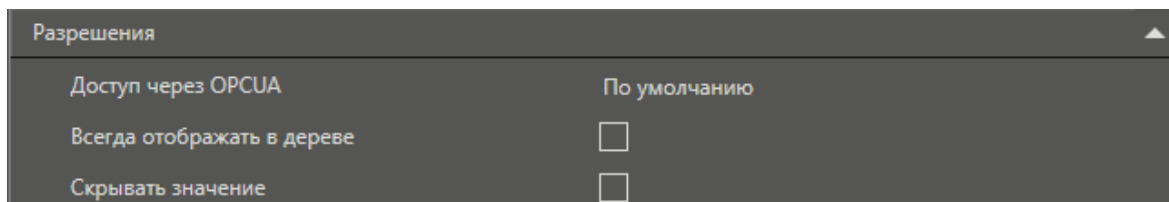
Название	Описание	Элементы
Программное имя	Используется для идентификации элемента независимо от текущего языка. Вручную Программное имя можно менять только если разработчик проекта осознает смысл этого действия! Менять вручную это свойство не рекомендуется.	У всех элементов
Категория	Описано в разделе Особенности создания окон в библиотеке	Все элементы
Порядок категории	Описано в разделе Особенности создания окон в библиотеке	Параметры окон
Id элемента	Числовой уникальный идентификатор элемента в рамках проекта. Данный номер указывается в случае возникновения ошибок при компиляции. Данный номер можно использовать для поиска элемента в дереве проекта	Все элементы
Запрет наследования	<p>Определяет возможность изменения наследников. Если флаг установлен, то элемент не будет отображаться в дереве и в панели свойств родительского элемента, который является наследником библиотечного элемента. Рассмотрим пример:</p> <p>Создали в библиотеке окно. В этом окне создали параметр, который нужен для его логики работы, но значение которого нельзя менять. В этом случае нужно установить у параметра данный флаг. Тогда при использовании наследника этого окна, параметр не будет виден в проекте и будет использоваться значение, заданное в его типе, в библиотеке</p>	Все элементы

Название	Описание	Элементы
Шаблон имени последовательного порта	<p>Задается правило, как формируется имя устройства, к которому будет подключаться драйвер, например, Modbus RTU.</p> <p>Если указано /dev/ttyO%d, то вместо %d подставится номер порта в проекте минус 1.</p>	Узел
Необходимая опция RT	<p>Если у библиотечного объекта установлено это свойство, то считается количество его экземпляров и наследников, включенное в конфигурацию каждого из узлов.</p> <p>При заказе коммерческой версии среды исполнения необходимо указать количество таких элементов в проекте.</p> <p>Если у библиотечного объекта скрыто содержимое, то данное свойство не отображается и изменить его нельзя.</p>	Объект
Образ	<p>Позволяет задать произвольный вид элемента. При нажатии на кнопку , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора медиа-ресурса. Можно в качестве образа использовать SVG элементы.</p>	Элементы графического редактора
Справочный раздел	<p>Можно самостоятельно привязывать разделы справочной системы к тому или иному элементу.</p> <p>Справочная система состоит из двух типов файлов: [название].chm и [название].dic.</p> <p>В файлах типа [название].dic для каждого раздела справочной системы формируется</p>	Все элементы

Название	Описание	Элементы
	<p>строка вида: proj_Kanal;help\proj.chm::Kanal.htm>sss// Канал</p> <p>Файл [название].chm представляет собой файловый архив, состоящий из различных htm-файлов.</p> <p>Если указать в свойстве Справочный раздел значение proj_Kanal, то при вызове контекстной справки для этого элемента откроется файл справки <i>proj.chm</i>, раздел <i>Kanal.htm</i>.</p>	

6.7.5. КАТЕГОРИЯ СВОЙСТВ РАЗРЕШЕНИЯ

В данном разделе представлены описания настроек, которые могут встречаться у различных элементов в группе Разрешения панели Свойств.



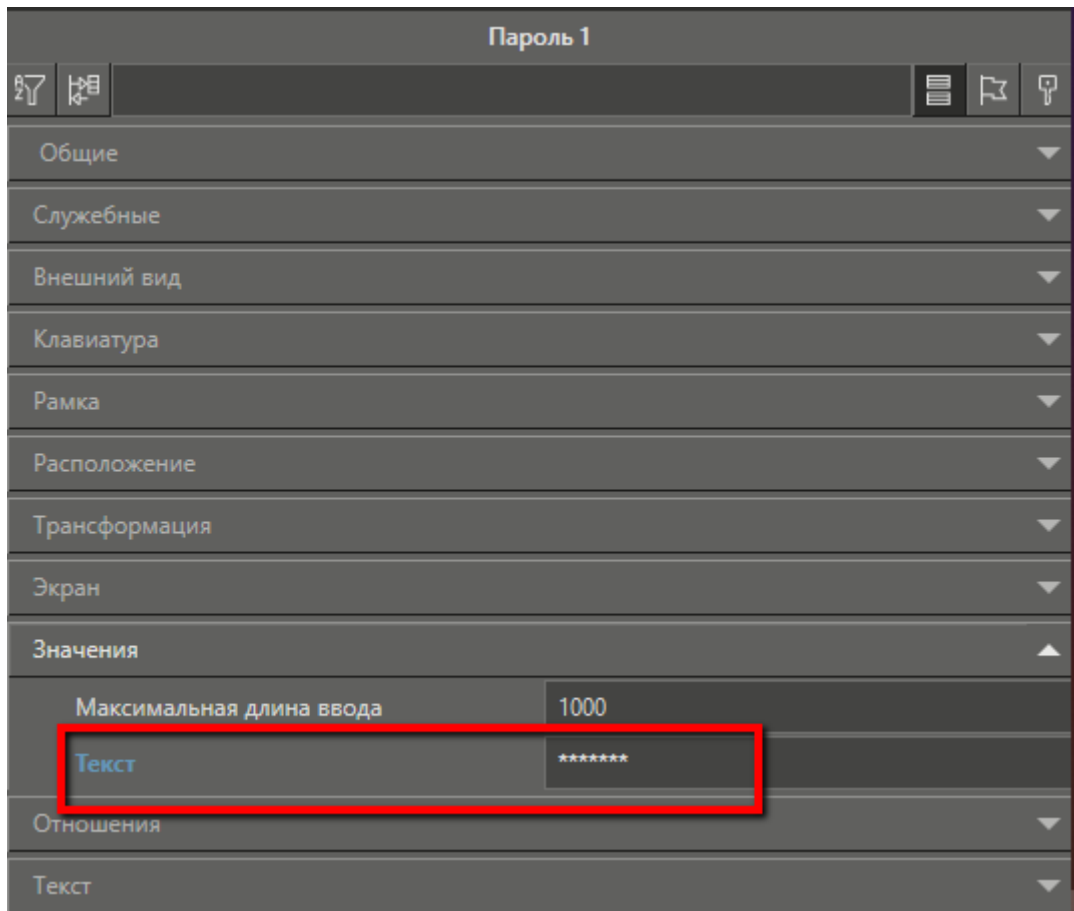
Название	Описание	Элементы
Доступ через OPCUA	<p>Определяет возможность доступа к элементу проекта через OPC UA, в случае, когда MasterSCADA 4D выступает в роли OPC UA сервера.</p> <p>Свойство может принимать следующие значения:</p> <ul style="list-style-type: none"> По умолчанию - используется значение ближайшего родителя, где свойство имеет значение Да или Нет Да - элемент доступен 	<p>Узел</p> <p>Протокол</p> <p>Модуль</p> <p>Канал</p> <p>Параметр канала, объекта, программы</p> <p>Объект, тег</p> <p>Экземпляр объекта, тега</p>

Название	Описание	Элементы
	<ul style="list-style-type: none"> Нет - элемент недоступен (но если у одного из дочерних стоит Да, то данный элемент тоже будет отображаться в иерархии OPC UA) 	
Всегда отображать в дереве	Определяет видимость элемента в дереве, в случае если объект защищен паролем.	Объекты, параметры
Скрывать значение	Определяет видимость введенного значения в среде разработки. Если флаг установлен, то введенные данные в поле Начальное значение будут отображаться в виде точек (звездочек). По умолчанию данный флаг установлен у свойств и параметров ФБ, которые служат для ввода паролей в среде разработки.	Входы и выходы функциональных блоков, свойства, параметры

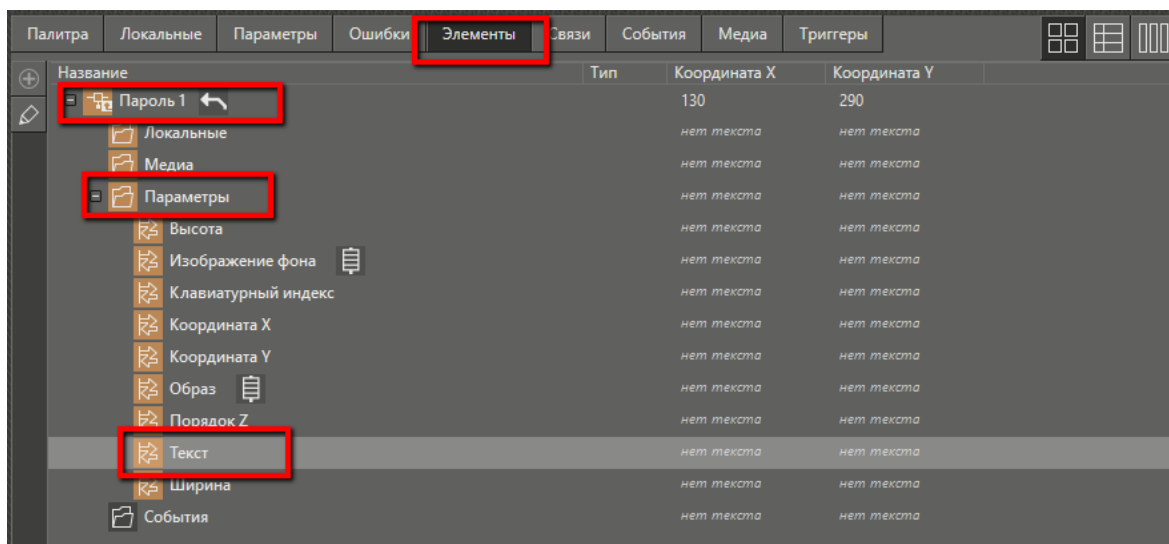
Пример использования свойства Скрывать значение

Рассмотрим пример как установленный флаг Скрывать значение повлиял на отображение Начального значения в панели свойств.

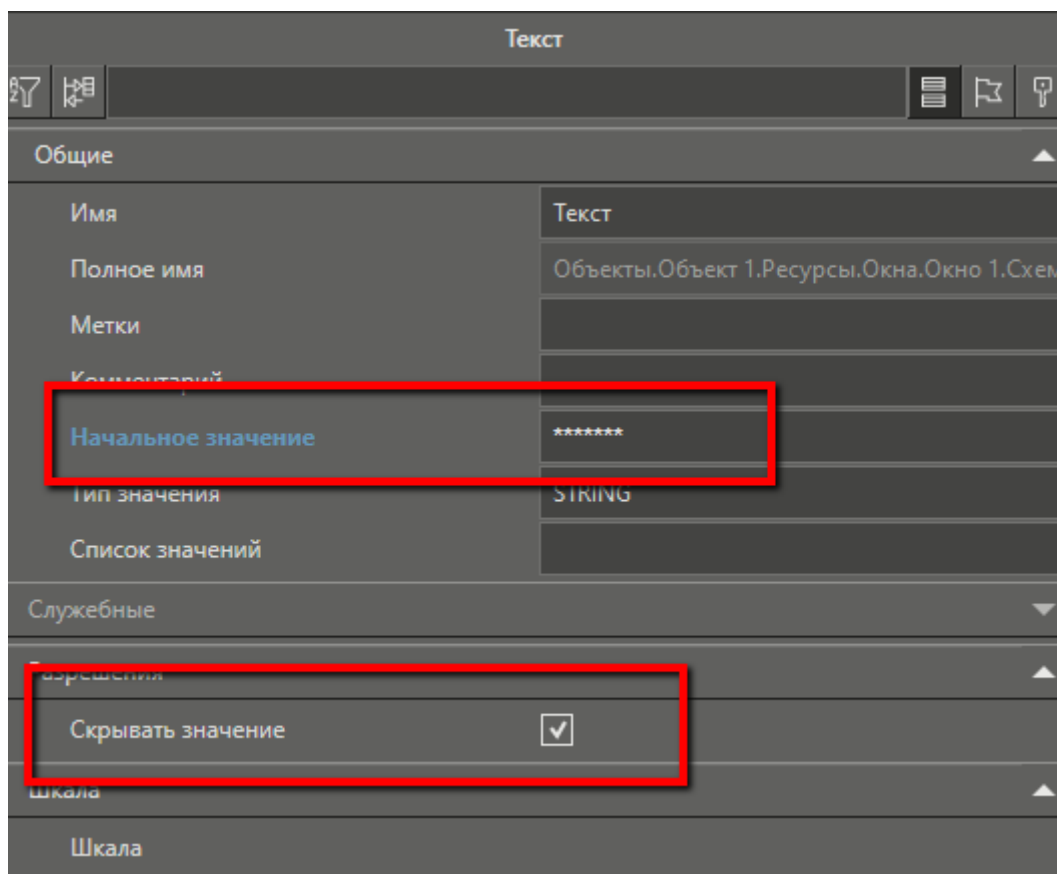
Добавим в окно элемент Пароль. Обратим внимание, что после ввода значение в свойство Текст оно заменяется звездочками:



Перейдем на панель Элементы легенды окна. Раскроем внутреннюю структуру элемента Пароль. В группе Параметры выделим Текст:



Перейдем на его панель свойств и увидим, что там установлен флаг в свойстве Скрывать значения и в свойстве Начальное значение также отображается замаскированный введенный ранее пароль:



6.8. РАБОТА С СООБЩЕНИЯМИ

Сообщения предназначены для регистрации событий, возникающих при работе АСУ ТП.

Сообщение является активным, если зарегистрированное событие еще актуально в данный момент времени. Сообщение становится неактивным, если событие, приведшее к его появлению, закончилось. По каждому источнику может быть только одно активное сообщение – то, которое появилось последним. Например, произошло событие – появилось активное сообщение, затем выключили среду исполнения (факт окончания события не был зафиксирован). Когда среду исполнения перезапустили, событие повторилось заново. Несмотря на то, что перед этим окончание предыдущего события не было зафиксировано, сообщение о нем не будет считаться активным. Активным будет считаться только последнее сообщение, у которого нет окончания.

Все сообщения, которые появляются во время работы исполнительной системы, сначала попадают в оперативную память устройства, а затем постепенно копируются из нее в базу данных. Если связь с базой данных установить не удастся, то сообщения будут сохраняться в оперативной памяти до тех пор пока связь с БД не будет восстановлена. Пока сообщение активно, его копия остается в оперативной памяти. Последнее неактивное сообщение по какому-либо источнику также остается в оперативной памяти до тех пор, пока не появится новое сообщение. Таким образом в штатном режиме по каждому источнику/типу сообщения в оперативной памяти будет храниться не более одного сообщения. Это

необходимо для того, чтобы быстрее получать доступ к активным сообщениям в журнале. Журнал – это элемент окна, который служит для работы с сообщениями в клиенте.

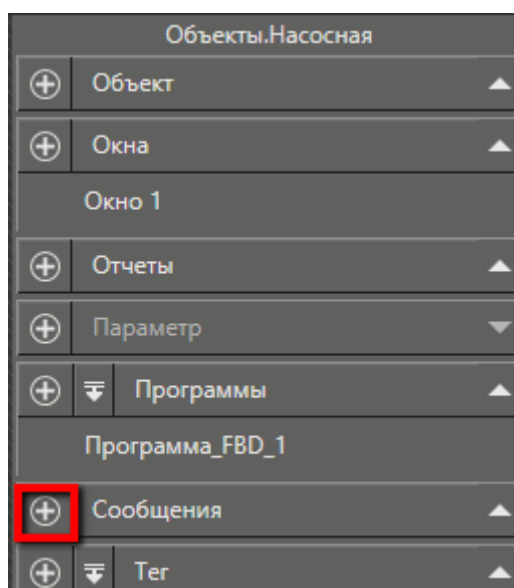
Сообщения можно разделить на условные и безусловные.

Условное сообщение– это сообщение, регистрирующее событие, у которого есть начало и есть конец. Например, если от модуля ввода-вывода пришел дискретный сигнал, обозначающий аварию частотного преобразователя, то до тех пор пока значение сигнала не изменится, сообщение будет считаться активным. В MasterSCADA 4D для работы с такими сообщениями предусмотрен тип сообщения Тревога. Тревога предназначена для работы с произвольными условными сообщениями. В случае если при использовании шкал у параметров заданы аварийные и предупредительные границы, то при нарушении границ автоматически будут выдаваться сообщения о нарушении границ и превышении скорости изменения.

Безусловное сообщение – это сообщение, которое регистрирует событие, имеющее начало, но не имеющее конца. Например, сообщение о смене режима работы насосов: тот, что работал резервным стал основным, а тот, что был основным, переключился в резерв. Для работы с произвольными безусловными сообщениями в MasterSCADA 4D используется элемент Базовое Сообщение. К безусловным сообщениям можно отнести сообщения, фиксирующие действия операторов. Эти сообщения, в зависимости от настроек прав доступа, формируются автоматически.

В MasterSCADA 4D система сообщений поддерживает механизм квитирования. Квитирование – это операция, производимая, как правило, оператором (реже автоматически) для подтверждения факта приема сообщения. Сообщение может быть квитировано оператором в журнале при помощи специальных механизмов, а также программно.

Сообщения могут создаваться в узлах, объектах, тегах и каналах при помощи контекстного меню либо контекстной панели:



6.8.1. ТИПЫ СТАНДАРТНЫХ СООБЩЕНИЙ

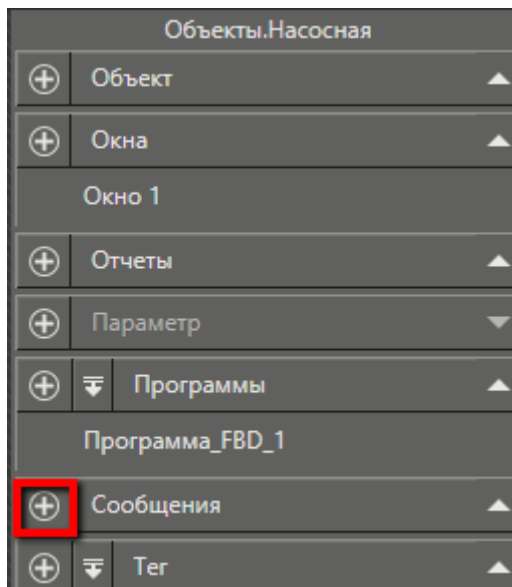
В MasterSCADA 4D по умолчанию предусмотрены следующие типы сообщений:

- Тревога
- Контроль границ
- Контроль скорости изменения
- Базовое сообщение

6.8.1.1. ТРЕВОГА

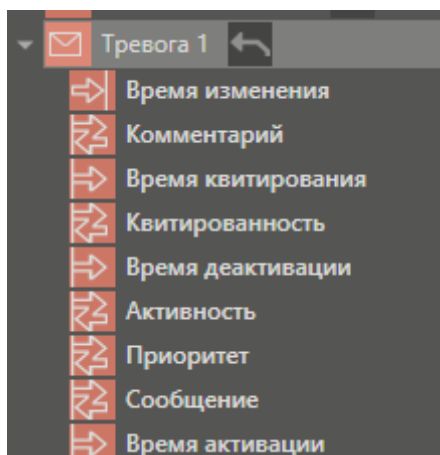
Тип сообщения Тревога служит для формирования условных сообщений.

Тревоги могут создаваться в узлах, объектах, тегах и каналах при помощи контекстного меню либо контекстной панели:

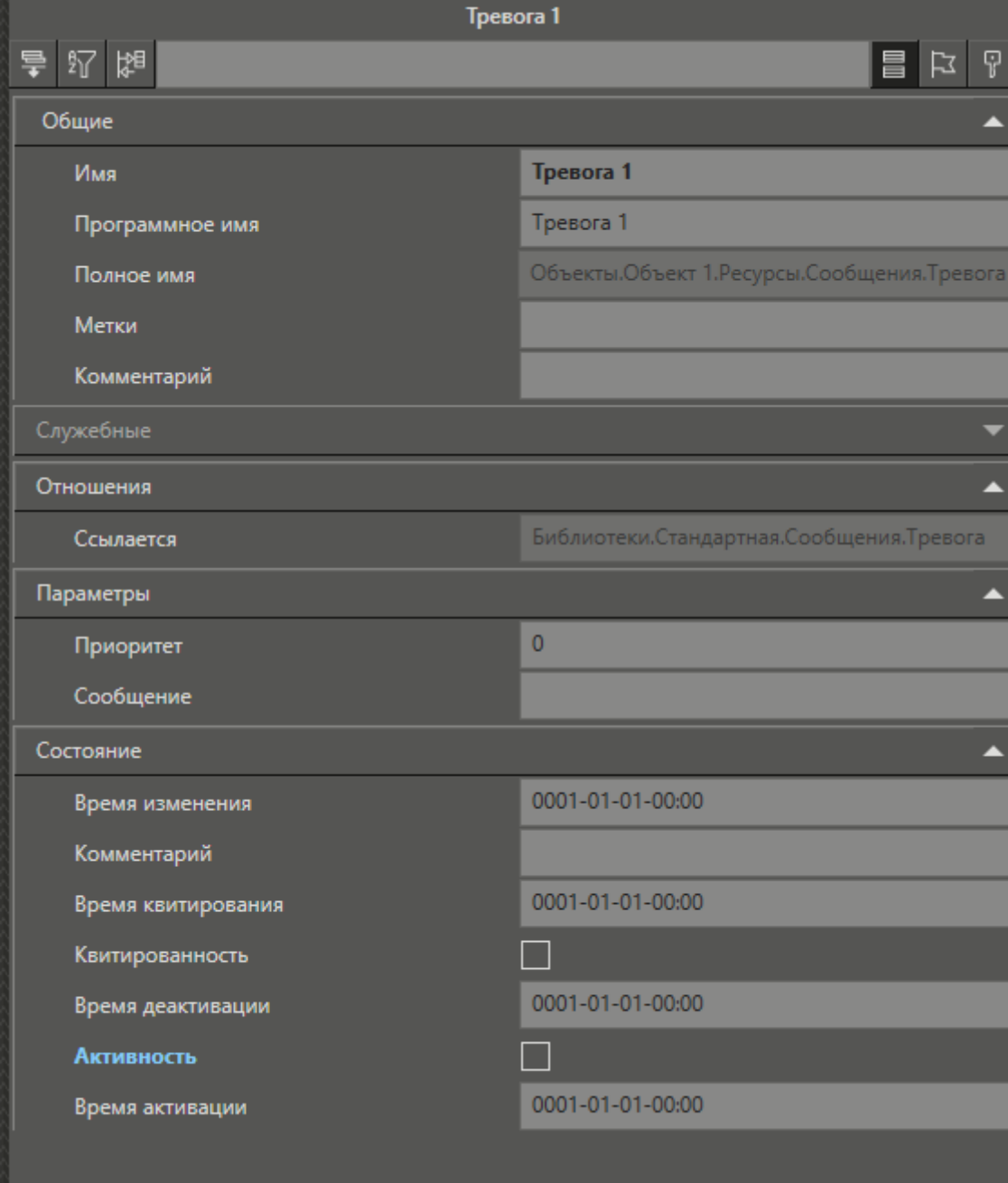


Созданное сообщение появится в группе Сообщения.

В Дереве Тревога имеет следующий вид:



Параметры, представленные в дереве, отображаются также и в панели свойств элемента, в группах Параметры и Состояние:



Тревога 1	
Общие	
Имя	Тревога 1
Программное имя	Тревога 1
Полное имя	Объекты.Объект 1.Ресурсы.Сообщения.Тревога
Метки	
Комментарий	
Служебные	
Отношения	
Ссылается	Библиотеки.Стандартная.Сообщения.Тревога
Параметры	
Приоритет	0
Сообщение	
Состояние	
Время изменения	0001-01-01-00:00
Комментарий	
Время квитирования	0001-01-01-00:00
Квитированность	<input type="checkbox"/>
Время деактивации	0001-01-01-00:00
Активность	<input type="checkbox"/>
Время активации	0001-01-01-00:00

Для связи параметров Тревоги разработчик проекта может использовать и дерево проекта, и панель свойств. Связь, настроенная в дереве, будет также отображаться в панели свойств (и наоборот).

Описание свойств (параметров) Тревоги:

Название (Программное имя)	Тип	Назначение
Приоритет (Severity)	DINT	<p>Служит для присвоения сообщениям уровня важности. Каждый разработчик проекта может ввести свою систему приоритетов, например, 1 – авария, самый высокий, а 5 - информация, самый низкий. Назначение сообщениям разной приоритетности можно использовать в дальнейшем для настройки фильтров журнала, например, выводить в журнал только сообщения с приоритетом равным единице. В приведенном примере это будет обозначать показ только аварийных сообщений.</p>
Сообщения (Message)	STRING	<p>Задается текст сообщения. Как правило, значение задается в виде константы. В базах данных возможно хранение сообщений больше 255 символов. При необходимости можно установить связь параметра со строковой переменной (например, выходом программы), в которой будет формироваться разный текст сообщения.</p> <p>Можно в текст тревоги включать значения различных параметров тревоги на тот момент времени, когда тревога стала активной, для этого нужно создать шаблон тревоги.</p> <p>Возможные варианты создания шаблона тревоги: <code><Текст>{Программное_имя_1}{Программное_имя_2}</code> <code><Текст><Программное_имя_3>;</code> <code><Текст>{/пробел[ы]/Программное_имя_1/пробел[ы]/} {/пробел[ы]/</code> <code>Программное_имя_2} <Текст><Программное_имя_3/пробел[ы]/>.</code></p> <p>Шаблоны не могут содержать пустые скобки ({ }), скобки с пробелом(-ми) внутри ({ }), повторения открывающихся или закрывающихся скобок ({ { , } }), не открытые и не закрытые скобки ({ , }).</p> <p>Строка внутри скобок должна полностью соответствовать программному имени параметра тревоги.</p> <p>В случае несоответствия сообщения шаблону, программа выведет исходное сообщение.</p> <p>Если сообщение соответствует шаблону, но вместо программного имени в скобках будет выведено текущее значение искомого параметра. Например, рассмотрим использование значения</p>

Название (Программное имя)	Тип	Назначение
		<p>параметра тревоги Активность в сообщении. Текст параметра тревоги Сообщение будет иметь вид:</p> <p>Сработала охрана. Значение параметра Активность = {Active}</p> <p>Тогда в режиме исполнения в журнале мы получим:</p> <p>Сработала охрана. Значение параметра Активность = true.</p> <p>Если необходимо в текст сообщения тревоги добавить значение произвольного параметра, то нужно создать новый тип тревоги, в который добавить нужные параметры и при создании шаблона тревоги использовать его программное имя.</p> <p>Важно! Важно! В базах данных MS SQL и Postgre возможно хранение сообщений больше 255 символов. В других БД поддерживается хранения сообщений до 255 символов.</p>
Время изменения (ChangeTime)	DT	Показывается время последнего изменения Тревоги. Совпадает с одним из трех параметров: временем активации, временем квитирования, временем деактивации.
Комментарий (Comment)	STRING	Может содержать произвольный текст. Как правило, заполняется оператором при квитировании сообщения.
Время квитирования (AckedTime)	DT	Выдается момент времени, в который оператор квитировал сообщение, т.е. когда свойство Квитированность изменилось с False на TRUE.
Квитированность (Acked)	BOOL	Отображается состояние квитирования сообщения, т.е. показывается, зафиксировал ли его оператор или нет. Если Квитированность =TRUE, то сообщение помечено как квитированное. Если Квитированность = False, то сообщение считается неквитированным. В зависимости от этого параметра можно задать фильтры журнала сообщений, например, для отображения только неквитированных сообщений или неквитированных и активных. Значение данного параметра может быть изменено при помощи специальных инструментов журнала сообщений.

Название (Программное имя)	Тип	Назначение
		При старте исполнительной системы для всех тревог, которые не срабатывали устанавливается значение TRUE для параметра тревоги Квитированность. После того как тревога сработает - параметра Активность изменит значение с False на TRUE квитированность примет значение False до того момента, пока сообщение не будет квитировано пользователем или программно.
Время деактивации (InactiveTime)	DT	Отображается момент времени, в который сообщение перестало быть актуальным (активным), т.е. когда свойство Активность изменилось с TRUE на False. При этом сообщение становится архивным.
Активность (Active)	BOO L	Если значение этого свойства становится равным TRUE, то сообщение считается актуальным (активным). Изменение значения свойства с TRUE на False, означает, что сообщение перестает быть актуальным. Это свойство необходимо связать с каким-либо параметром (например, выходом программы).
Время активации (ActiveTime)	DT	Отображается время, когда Активность изменила состояние с False на TRUE.

Важно! Если разработчик работает с тревогами в программе, то необходимо использовать программные имена полей тревог

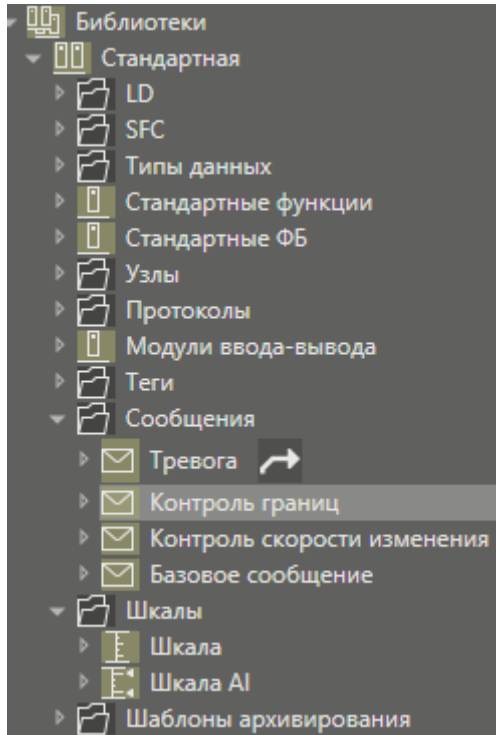
Тип сообщения Тревога хранится в библиотеке Стандартная, в группе Сообщения.

Разработчик проекта на основе данного типа может создать свои типы тревог, например, добавив свои собственные параметры, которые будут архивироваться и выводиться в журнал.

6.8.1.2. КОНТРОЛЬ ГРАНИЦ

Сообщение о нарушении границ – это условное сообщение, поступающее в случае, если значение параметра в режиме исполнения вышло за заданные пределы в назначенной шкале.

Этот тип сообщения находится в библиотеке Стандартная:



Настройка элемента не производится.

Сообщения будут автоматически попадать в журнал сообщений в соответствии с настройками журнала.

Время активации, время деактивации и другие параметры состояния будут зависеть от настроек шкалы параметра и его значения в тот или иной момент времени.

6.8.1.2.1. КОНТРОЛЬ ГРАНИЦ. ТЕКСТЫ СООБЩЕНИЙ

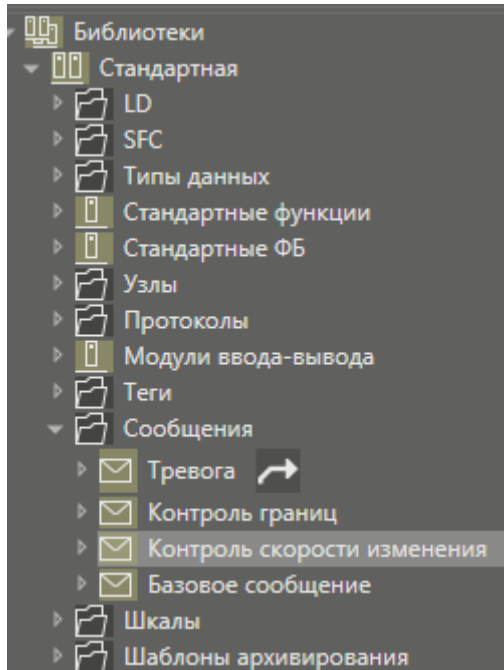
В зависимости от того, какую границу нарушит параметр, для которого назначена шкала, текст сообщения будет разным:

- Верхнее аварийное,
- Верхнее предупредительное,
- Нижнее предупредительное,
- Нижнее аварийное.

6.8.1.3. КОНТРОЛЬ СКОРОСТИ ИЗМЕНЕНИЯ

Сообщение о превышении скорости изменения – это условное сообщение, поступающее в случае, если в режиме исполнения значение параметра изменилось быстрее, чем задано в настройках назначенной шкалы.

Такой тип сообщения находится в библиотеке Стандартная:



Настройка элемента не производится.

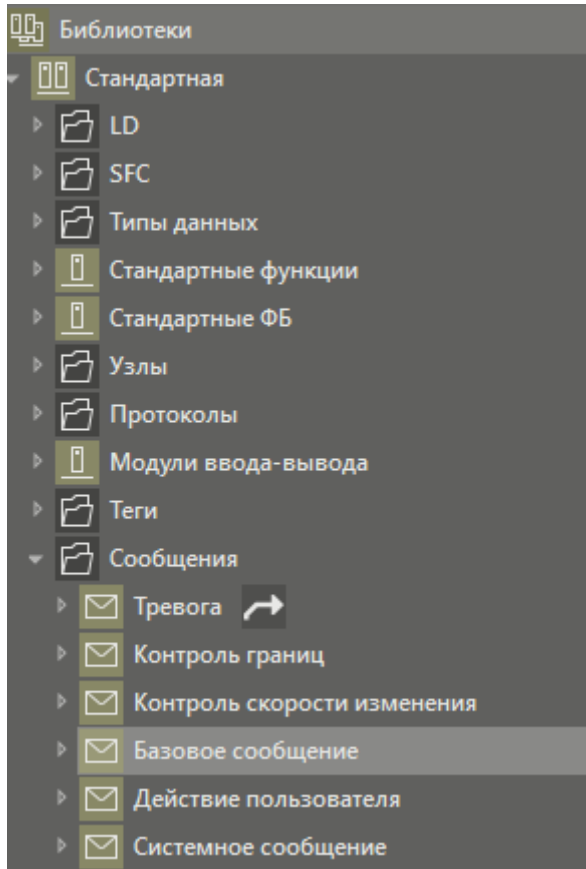
Сообщения будут автоматически попадать в журнал сообщений в соответствии с настройками журнала. Текст сообщения будет: Максимальная скорость изменения.

Время активации, время деактивации и другие параметры состояния будут зависеть от настроек шкалы параметра и его значения в тот или иной момент времени.

6.8.1.4. БАЗОВОЕ СООБЩЕНИЕ

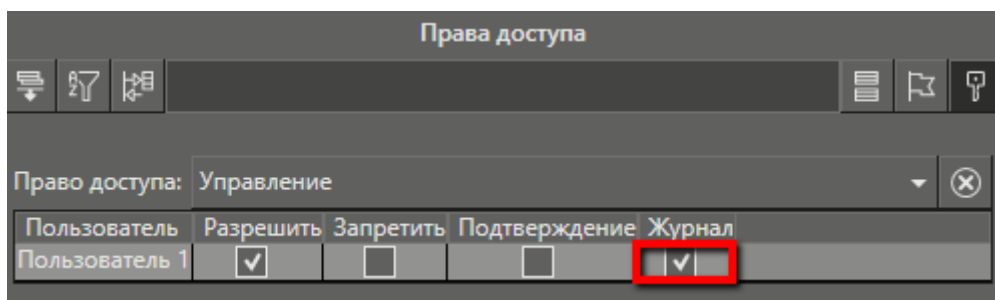
Базовое сообщение используется для формирования безусловных сообщений. Безусловное сообщение создается в MasterSCADA 4D при помощи ФБ FireBaseEvent.

Данный тип сообщений находится в библиотеке Стандартная:



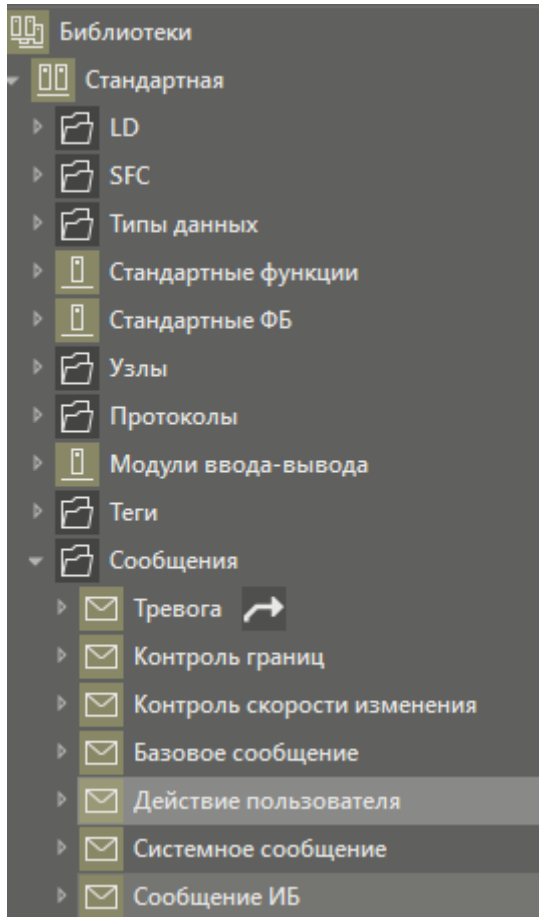
6.8.1.5. ДЕЙСТВИЕ ПОЛЬЗОВАТЕЛЕЙ

Этот тип сообщений используется для формирования сообщений по действиям пользователей. Если в настройках прав доступа для конкретного действия установлен флаг Журнал, то когда пользователь произведет данное действие, сообщение о нем появится в системе автоматически:



Сообщения, сигнализирующие о начале и окончании сессии пользователей формируются автоматически.

Данный тип сообщений находится в библиотеке Стандартная:



Дополнительных настроек не требуется.

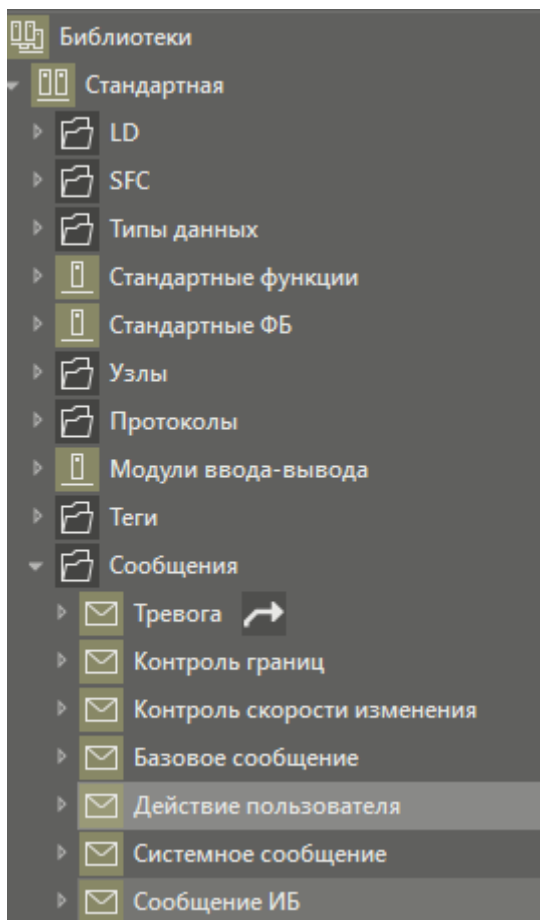
Сообщения будут автоматически попадать в журнал сообщений, в соответствии с настройками журнала.

6.8.1.6. СИСТЕМНОЕ СООБЩЕНИЕ

Этот тип сообщений используется для формирования безусловных сообщений следующих видов:

- Сообщения об ошибках ключа защиты
- Сообщения об ошибках, возникших во время проверки целостности системного и прикладного ПО

Данный тип сообщений находится в библиотеке Стандартная:



Сообщения данного типа формируются автоматически. Просмотреть сообщения можно используя журнал сообщений, в соответствии с настройками журнала.

6.8.1.7. СООБЩЕНИЕ ИБ

Этот тип сообщений используется для формирования безусловных сообщений следующих видов:

Сообщение об ошибке ключа защиты.

- "Ключ не найден. Проект остановлен."
- "Ключ не найден. Проект будет остановлен через %d минут."

Сообщения контроля целостности

- "Ошибка при проверке целостности проекта: "
- "Ошибка при проверке целостности ПО: "

Сообщения об операциях над пользователями в режиме исполнения Сообщения данного типа состоят из постоянной части "Неуспешная попытка логина пользователя" к которой добавляется причина возникновения сообщения:

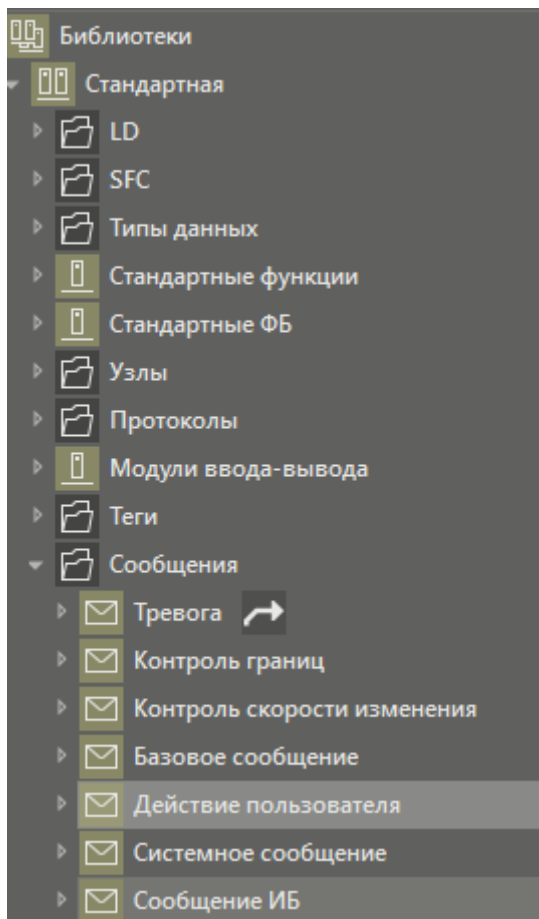
- "Неудачная смена пароля"

- "Необходимо сменить пароль"
- "Пользователь заблокирован"
- "Срок действия пароля истёк"
- "Недопустимое время логина"
- "Недопустимый адрес логина"

Сообщение о блокировке пользователя, которая была вызвана настройкой Количество последовательных неуспешных попыток входа "Пользователь " + login + " заблокирован по превышению попыток логина"

Сообщения об авторизации пользователя: "Вход в систему"

Данный тип сообщений находится в библиотеке Стандартная:



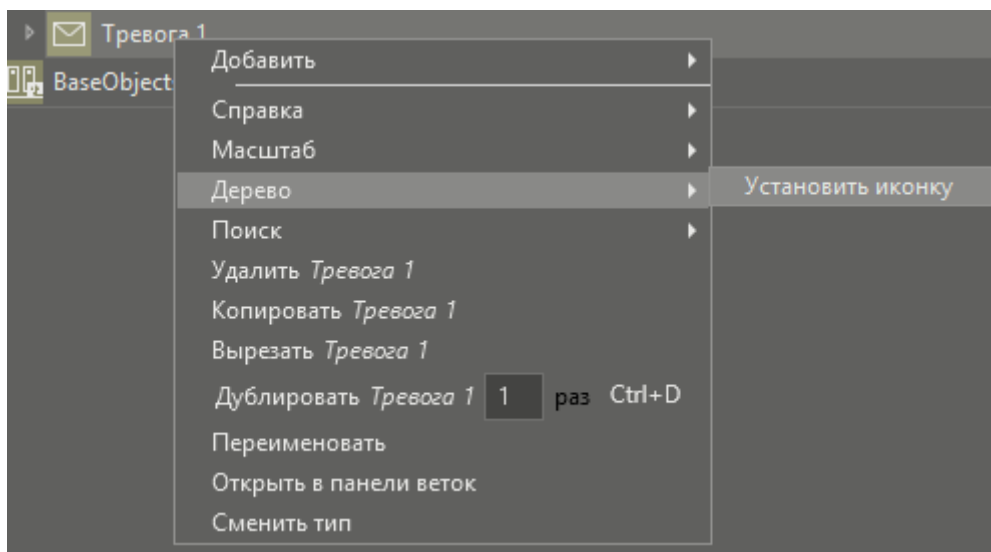
Дополнительных настроек не требуется.

Сообщения будут автоматически попадать в журнал сообщений, в соответствии с настройками журнала.

6.8.2. НОВЫЙ ТИП ТРЕВОГИ

Разработчик проекта может создать свой Тип тревоги в пользовательской библиотеке. Это целесообразно делать в случае когда необходимо кроме стандартных параметров тревоги сохранять в базе данных и выводить в журнале дополнительную информацию, либо в том случае, если набор параметров соответствует требованиям, но необходимо задать предопределенные значения свойств (параметров).

Если также необходимо, чтобы разные типы тревог имели разные обозначения в столбце журнала в поле Иконка, то следует в библиотеке при помощи контекстного меню Тревоги выбрать пункт *Дерево.Установить иконку*.



6.8.3. АРХИВИРОВАНИЕ СООБЩЕНИЙ

Все сообщения, возникающие в среде исполнения, автоматически попадают в архив. Архивирование сообщений поддерживается на всех ОС.

Для хранения архивов могут быть использованы следующие базы данных:

- SQLITE;
- POSTGRESQL;
- MS SQL.

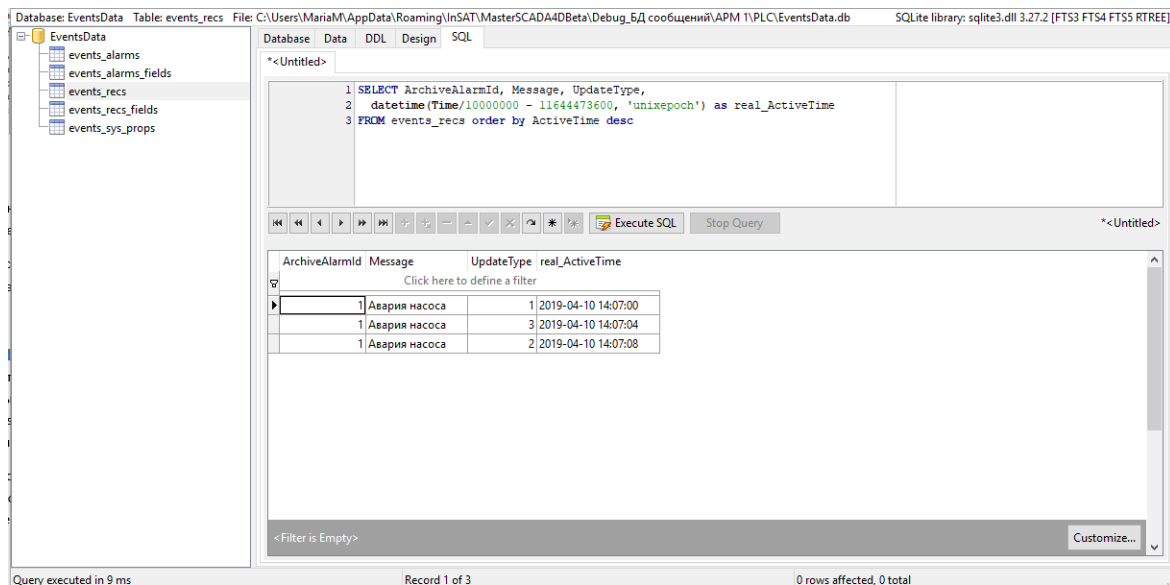
Настройки архивирования сообщений задаются в панели свойств элемента Основной архив сообщений.

При архивировании сообщений в архив записывается новая строка при каждом изменении состояния сообщения:

- Появление;
- Квитирование;

- Исчезновение.

Пример базы данных сообщений:



На рисунке представлено окно программы SQLiteExpert.

Архив сообщений по умолчанию

По умолчанию архив сообщений хранится в базе данных SQLite.

Файл базы данных носит название *EventsData.db*.

Если проект запускается в среде исполнения, входящей в состав среды разработки, то архив будет храниться в папке: *<профиль пользователя>\AppData\Roaming\MPSSot\<продукт>\Debug_<имя проекта>\<имя узла>\PLC\EventsData.db*.

В среде исполнения, установленной независимо от среды разработки, файл архива будет храниться в папке: *c:\Users\[имя пользователя]\AppData\Roaming\MPSSot\MasterSCADA4DRT1.2\Server\EventsData.db*

Архив сообщений на других ОС хранится внутри папки *tplc*.

Работа с активными сообщениями

Для просмотра активных сообщений используется элемент Журнал. В базе сообщений в таблице *events_alarms* есть столбцы *EventType* и *LastRecId*. *LastRecId* указывает на последнюю запись сообщения активации в таблицу *events_recs*. При загрузке исполнительной системы в журнал будут загружены из базы данных активные сообщения

Работа с архивными сообщениями

Для просмотра архивов сообщений в MasterSCADA 4D предусмотрен графический элемент – Архивный журнал.

6.8.4. СПИСОК ПРОГРАММНЫХ ИМЕН ПАРАМЕТРОВ ТРЕВОГ

В программах, на входах функциональных блоков, предназначенных для работы с сообщениями нужно использовать программные имена, а не те которые отображаются в дереве.

Таблица соответствия между названием параметра тревоги и его программным именем:

Название/Описание	Тип	Программное имя
Основные - параметры, которые отображаются в дереве проекта		
Время активации	DATE_AND_TIME	ActiveTime
Время деактивации	DATE_AND_TIME	InactiveTime
Время квитирования	DATE_AND_TIME	AckedTime
Активность	BOOL	Active
Квитированность	BOOL	Acked
Приоритет	DINT	Severity
Сообщение	STRING	Message
Комментарий, выставляемый при квитировании	STRING	Comment
Время изменения	DATE_AND_TIME	Time
Вспомогательные - параметры, которые не отображаются, но которые можно использовать при работе с сообщением в журнале или в функциональных блоках, программах		

Название/Описание	Тип	Программное имя
Идентификатор записи сообщения в архиве	DINT	RecId
Id источника сообщения (если источник внутри экземпляра объекта, то в ItemId передается Id экземпляра объекта)	LINT	ItemId
Идентификатор сообщения	STRING	EventId
IP - адрес клиента, с которого выполнено квитирование	STRING	ClientAddress
Имя пользователя, который выполнил квитирование	STRING	ClientUserId
Путь внутри экземпляра с Id, указанным в параметре ItemId	STRING	Path -
Тип сообщения	STRING	EventType
Битовая маска состояния: 0 - не активно, не квитировано; 1 - не активно, квитировано; 2 - активно, не квитировано; 3 - активно, квитировано.	UINT	EventState
Имя объекта, включающее полный путь в дереве объектов	STRING	EventFullObjectName
Имя объекта	STRING	EventObjectName
Источник сообщений	STRING	EventSourceName

6.9. НАСТРОЙКА СЕТЕВЫХ ПРОЕКТОВ

Поддерживаемые архитектуры:

- Клиент-серверная
- Сервер-серверная
- Резервирование узлов

6.9.1. РЕЗЕРВИРОВАНИЕ УЗЛОВ

Описание алгоритма резервирования.

Общие положения

Два устройства с загруженной в них исполнительной системой MasterSCADA 4D специальной конфигурации работают одновременно: одно имеет статус Основной (MASTER), другое — Резервный (Slave). Клиент должен подключаться к тому узлу, который находится в режиме основного. Если связь с этим узлом будет потеряна, то страница клиента автоматически переподключится к тому узлу, который находится в состоянии основного. Программы на обоих узлах выполняются независимо. Опрос устройств могут вести два узла одновременно. Однако для протоколов, в которых параллельный опрос устройств невозможен, получение текущих данных по протоколам, добавленным в проект, производит только один из них — тот, который работает в режиме основного. В процессе работы, параметры, у которых в настройке Сохранять установлено значение Да, передаются от основного узла резервному. Период передачи данных определяется в настройках службы резервирования. В общем случае, при возникновении отказа основного узла управление переходит к резервному. Переключение клиентов визуализации к тому узлу, который находится в состоянии Master происходит без перезагрузки страницы клиента визуализации. Раз в 10 секунд происходит синхронизация сессий пользователей между Master и Slave.

Причины возникновения отказа

- Ручное формирование отказа через панель управления узлом.
- В результате отказа любого протокола добавленного в проект. Если оба узла выставят отказ, то в режиме основного останется последний узел, получивший этот статус.
- Программный отказ.

Правила переключения

Переключение MASTER—Slave происходит в следующих случаях:

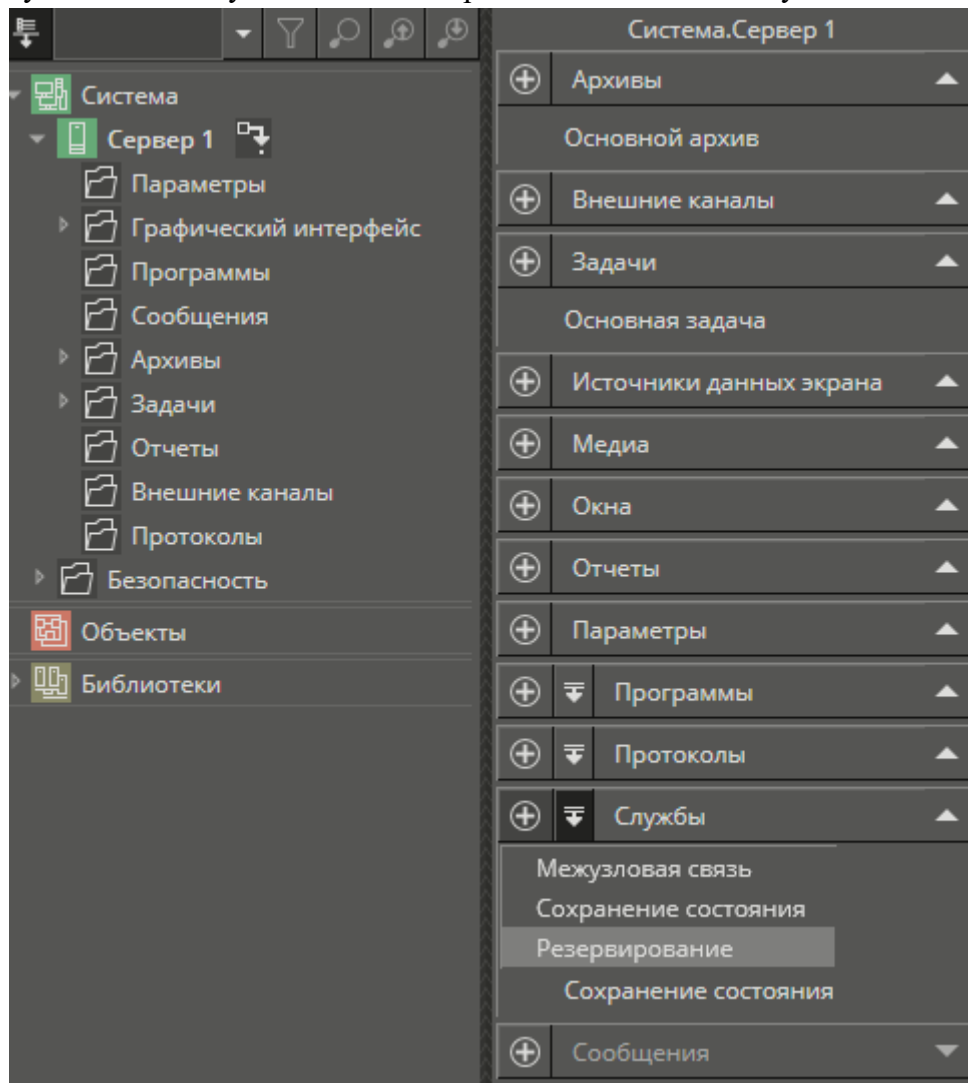
1. если текущий узел Slave и при этом другой узел не отвечает или тоже находится в режиме Slave, то осуществляется переключение текущего узла в MASTER;
2. если текущий узел MASTER, и у него возникает отказ, а на другом узле нет отказа, то осуществляется переключение текущего узла в Slave, а другой узел переключается в MASTER по правилу 1;

3. если текущий узел MASTER и другой узел тоже MASTER, то осуществляется переключение текущего узла в Slave.

Оба узла равноправны, однако на узле 2 имеется пауза при старте 10 сек, для того чтобы при одновременном включении узел 1 загрузился раньше и стал MASTER. Но в дальнейшем переключение идет по общим правилам.

Настройка резервирования

1. Добавить службу Резервирование. Для этого в контекстном меню группы Службы нужно выбрать соответствующий пункт.



2. После выполнения предыдущего пункта, в панели свойств узла появятся дополнительные настройки, позволяющие определить в сети второй сервер из резервируемой пары.

APM 1

Общие

Служебные

Резервирование

Таймаут	1s
IP адрес узла 2	10.0.0.23,192.168.0.23
Количество повторов при неудачном опросе	1
Количество передаваемых архивных значений	100
Выдержка времени при старте	1s
Доступ к OPC UA серверу в режиме SLAVE	<input checked="" type="checkbox"/>

Отношения

Настройки

Связь

Свойство	Назначение
IP-адрес узла 2	Задается IP-адрес устройства (компьютера, контроллера и т. п.), которое выполняет роль второго сервера. Если устройство имеет две сетевые карты, то задаются два IP-адреса через запятую. IP-адрес узла 1 при резервировании задается в группе настроек Связь в поле IP адрес.
Таймаут	Таймаут опроса состояния компьютеров.
Количество повторов при неудачном опросе	Определяется количество неуспешных попыток связи основного и резервного серверов, выполненных подряд, прежде, чем будет сформирована ошибка связи.
Количество передаваемых архивных значений	Указывается максимальное количество значений одного параметра, передаваемых в одном запросе архива при резервировании. По умолчанию установлено значение 100
Выдержка времени на старте	Определяет через какое время (N) после загрузки конфигурации будет возможен переход с основного на резервный для обновления основного Рассмотрим порядок работы:

Свойство	Назначение
	<p>1) Обновляется конфигурация в резервном сервере (Slave).</p> <p>2) Сервер перезагружается</p> <p>3) В течении времени N после перезагрузки резервного принудительно остается в режиме Slave (основной видит этот статус и не пытается передать управление при своем отказе, среда разработки считает, что загрузка все еще идет в него)</p> <p>4) Снимается запрет резервного на переход в Master. Среда получает статус успешной загрузки и начинает загрузку файлов в него</p> <p>5) После загрузки файлов в основной он передает управление на резервный и на N секунд ставится признак запрета на переход в Master.</p> <p>6) По истечении задержки основной перезагружается.</p> <p>7) основной загрузился. Аналогично п.3 в течении N секунд действует запрет на перевод основного в Master</p> <p>8) По истечении этого времени снимается запрет и среда разработки выдает статус о загрузке конфигурации в оба контроллера. Если нет отказов, то остается сервер, что был в начале алгоритма основным - Slave, а тот, что был резервный - Master.</p>
<p>Доступ к OPC UA серверу в режиме SLAVE</p>	

3. Расширенный доступ к настройкам, отвечающим за резервирование, можно получить в панели настройки. Для этого в дереве системы следует выделить службу Резервирование.

Резервирование

Общие

Служебные

Задача

Период, мс	1000
Приоритет	100
Подключение по условию	<input type="checkbox"/>
Выполнение по условию	<input type="checkbox"/>
Выполнять на резервном	<input type="checkbox"/>
Выполнять запись на резервном	<input type="checkbox"/>
Способ записи	По изменению
Формировать отказ при отказе всех модулей	<input type="checkbox"/>
Формировать отказ узла	<input checked="" type="checkbox"/>
Задержка записи при старте опроса (циклов)	0
Задержка сброса отказа на резервном, мс	10000

Отношения

Протокол

Таймаут	1s
IP адрес узла 2	0.0.0.0
Количество повторов при неудачном опросе	1
Количество передаваемых архивных значений	100
Выдержка времени при старте	1s
Доступ к OPC UA серверу в режиме SLAVE	<input checked="" type="checkbox"/>

Свойство	Назначение
Категория Задача	
Период	Период, с которым будут передаваться данные от основного узла резервному. Другие свойства описаны в соответствующем разделе.

Важно! В базовую версию MasterSCADA 4D RT не входят возможности по резервированию узлов. Эта функция является отдельной опцией. С ней можно ознакомиться в демоверсии программы.

Порядок синхронизации данных между основным и резервным

Синхронизация текущих данных

Между основным и резервным передаются только те текущие значения, для которых свойство Сохранить имеет значение Да или Наследуется, но во втором случае у одного из родительских элементов должно быть установлено Сохранять=Да. Поддерживается синхронизация значений параметров различных типов данных, в том числе длинных строк (более 256 символов), при условии, что параметр имеет тип STRING, поддерживается синхронизация динамических массивов (массивов переменной длины)

Кроме значения передается и признак блокировки. Период, с которым основной будет передавать информацию резервному, определяется одноименной настройкой, которая находится в категории Задача службы Резервирование.

Данные передаются по внутреннему протоколу. Каждый сервер имеет свой файл, необходимый для восстановления после рестарта. После получения необходимых данных от основного резервный записывает и в свой файл *session.bin*.

Синхронизация архивных данных

Архивные данные передаются в процессе работы основного сервера к резервному, за один раз по каждому параметру может быть передано значений не больше, чем указано в свойстве Количество передаваемых архивных значений.

Если резервный сервер был отключен, то после подключения он получит значения всех архивируемых в проекте параметров, начиная с того времени, когда было записано в базу данных резервного сервера последнее значение по каждому из них.

Синхронизация сообщений

Все текущие сообщения и их состояния передаются на резервный сервер.

Если резервный сервер был отключен, то после включения он получит все сообщения, которые возникали с момента последней записи в базу данных

Обработка запросов при работе резервирования

UDP запросы следующих типов -

- Запрос резервного сервера на получение архивных данных
- Запрос резервного сервера на получение архивных/новых сообщений

Обрабатываются в отдельном потоке. Данные типы запросов в статистике отображаются как N2

6.10. СПРАВОЧНАЯ ИНФОРМАЦИЯ

Соответствие элементов JSON и MasterSCADA 4D

Настройка DCOM Windows

Структура таблиц баз данных с архивами

6.10.1. СООТВЕТСТВИЕ ЭЛЕМЕНТОВ JSON И MASTERSCADА 4D

Если текст заключен в фигурные скобки, то в MasterSCADA 4D полученная информация будет иметь тип данных Структура.

Если текст заключен в квадратные скобки, то MasterSCADA 4D полученная информация будет иметь тип данных Массив.

Ключи соответствуют именам полей структур. Тип значения ключа, передаваемое в формате JSON должно соответствовать такому же типу в MasterSCADA 4D.

Пример

Итоговый тип данных Структура

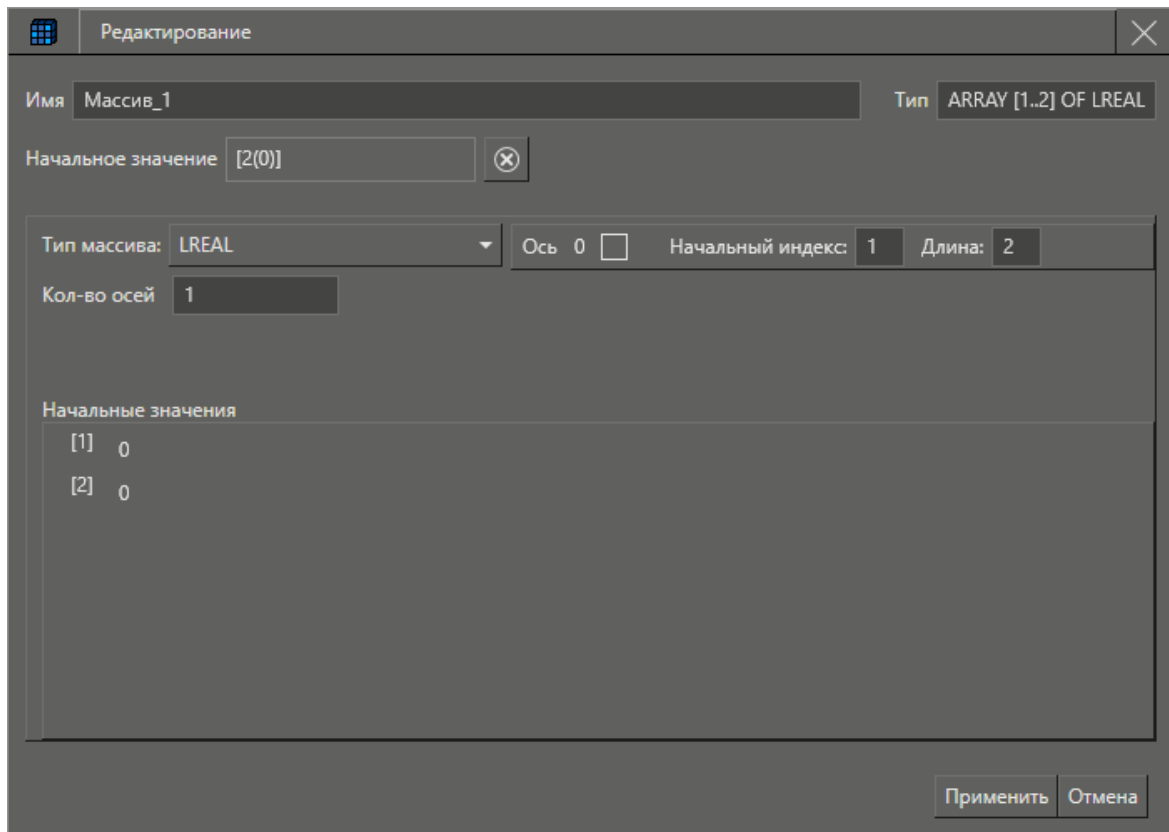
Пусть в запросе JSON приходят значения вида:
{"t":2,"t2":"т","a":[1.1,2],"a2":[{"b":true,"c":5},{"b":false,"c":7}]}

Это значит, чтобы в MasterSCADA 4D разобрать такую строку нужно в пользовательской библиотеке создать структуру состоящую из 4 полей:

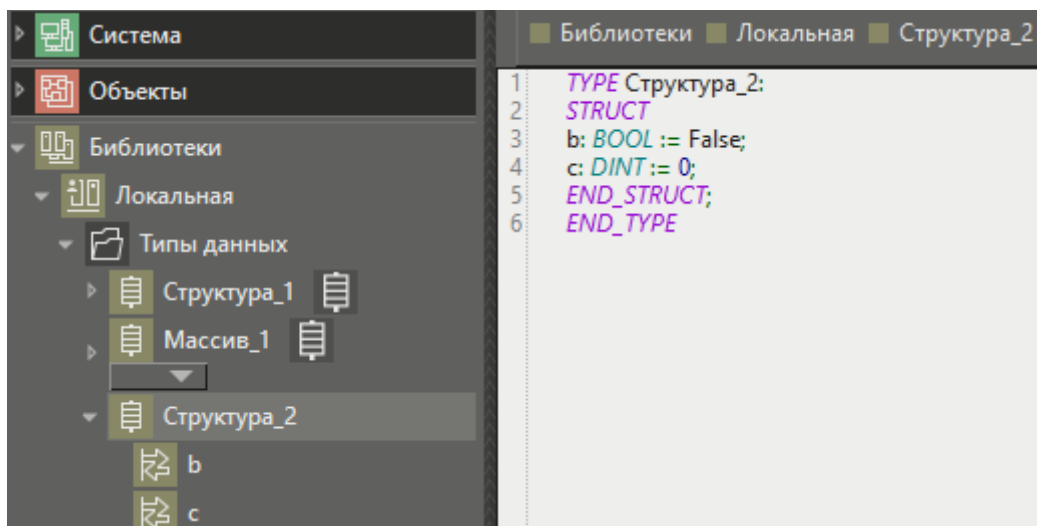
Имя поля	Тип поля
t	DINT
t2	STRING
a	Массив, состоящий из элементов с типом данных LREAL
a2	Массив состоящий из элементов с типом данных Структура.

Прежде чем приступать к созданию итоговой структуры, нужно создать в пользовательской библиотеке вспомогательные типы данных.

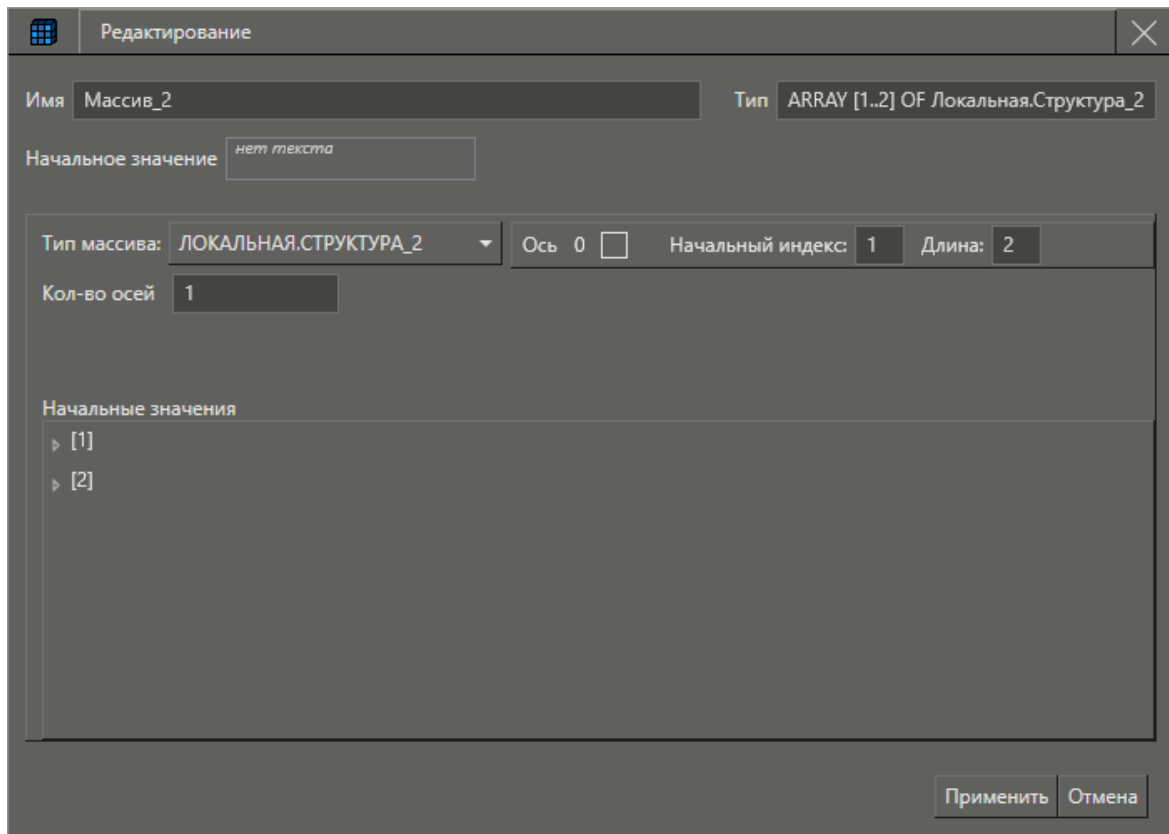
Массив для поля a:



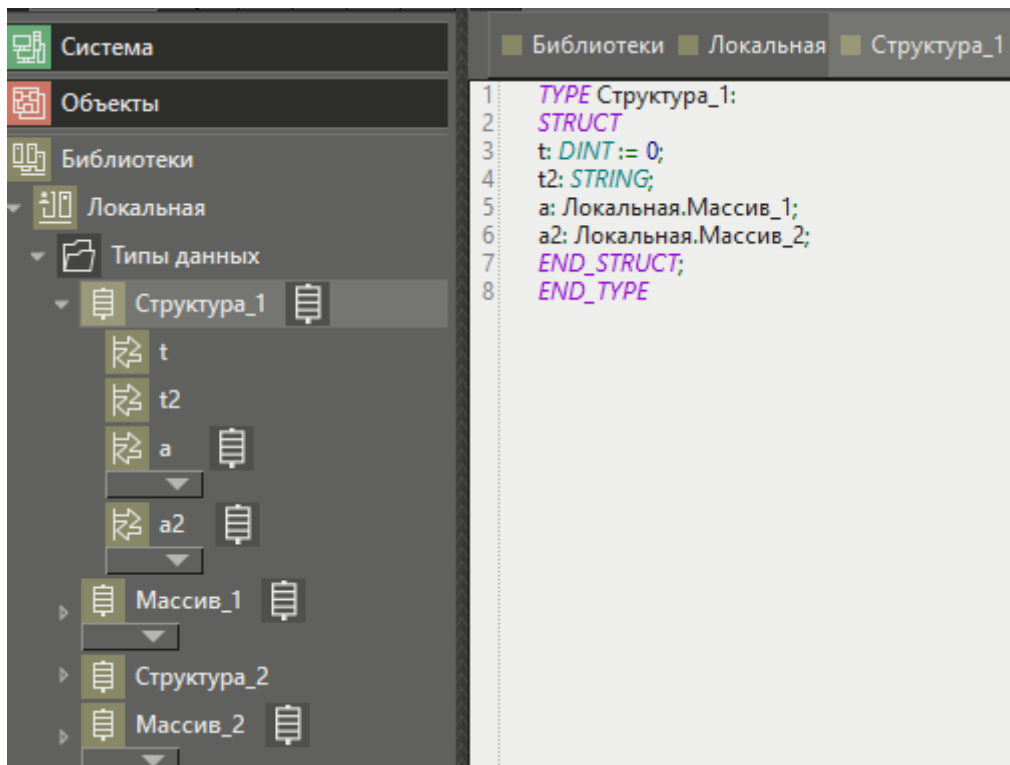
Структуру для создания массива поля a2:



Массив для поля a2 на базе созданной выше структуры:



Затем создаем результирующий тип данных:



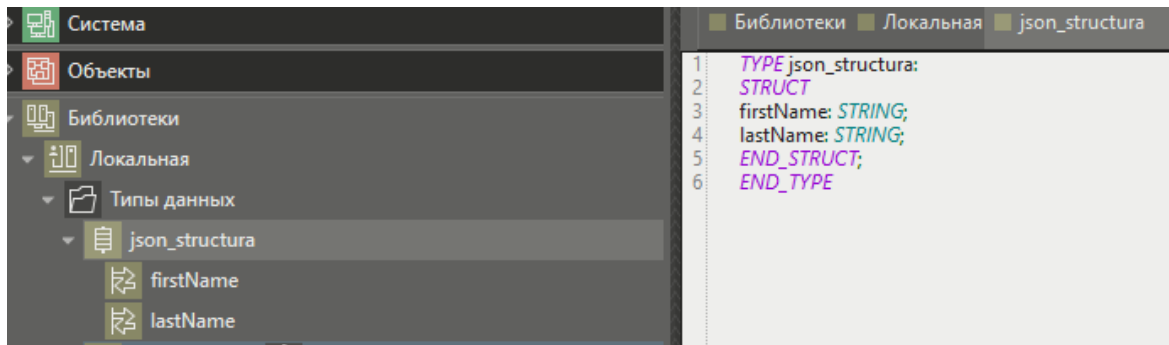
Итоговый тип данных Массив

Пусть в запросе JSON приходят значения вида:

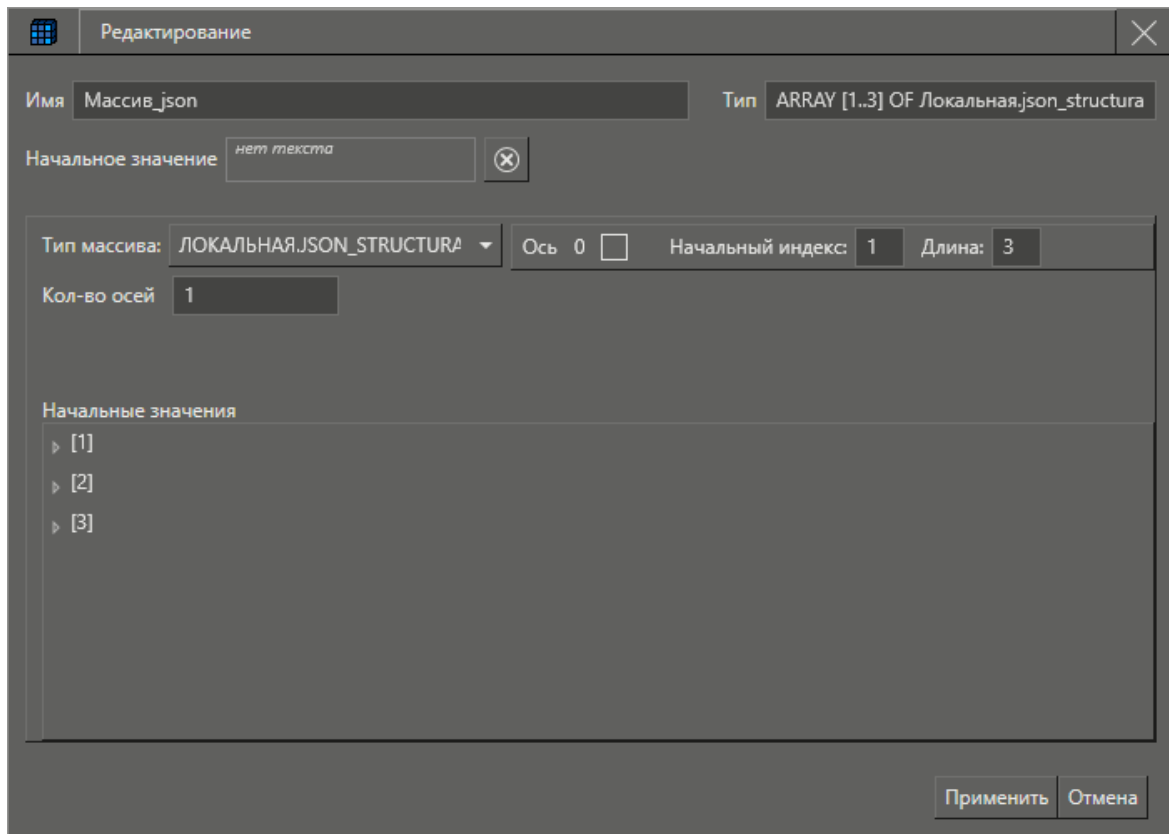
```
[  
  {"firstName":"Илья", "lastName":"Кукушкин"},  
  {"firstName":"Алексей", "lastName":"Шмелев"},  
  {"firstName":"Георгий", "lastName":"Московский"}  
]
```

Видно, что итоговый тип данных будет массив структур.

Сначала нужно создать структуру, из которой будут состоять элементы массива:



Затем создаем итоговый тип данных:



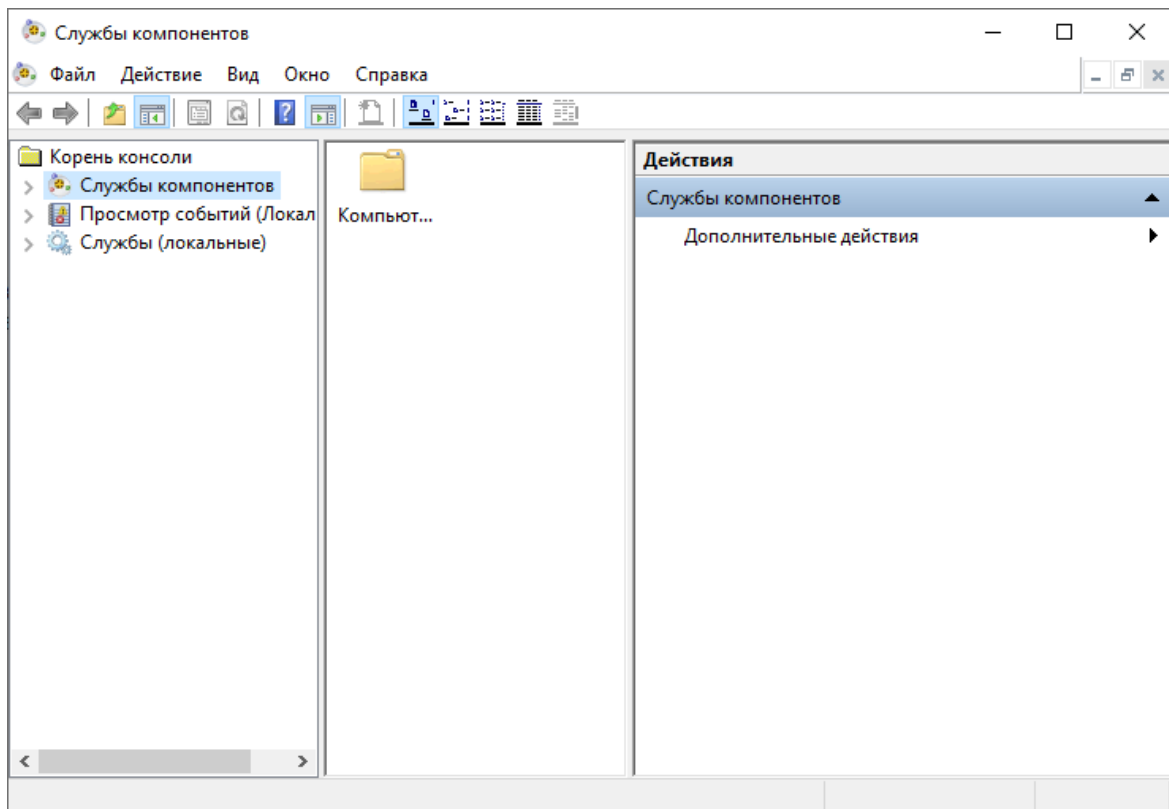
Теперь созданный тип данных нужно назначить параметру, который связан с выходом функции `STRING_TO_OBJECT`, либо параметру канала протокола MQTT

6.10.2. НАСТРОЙКА DCOM WINDOWS

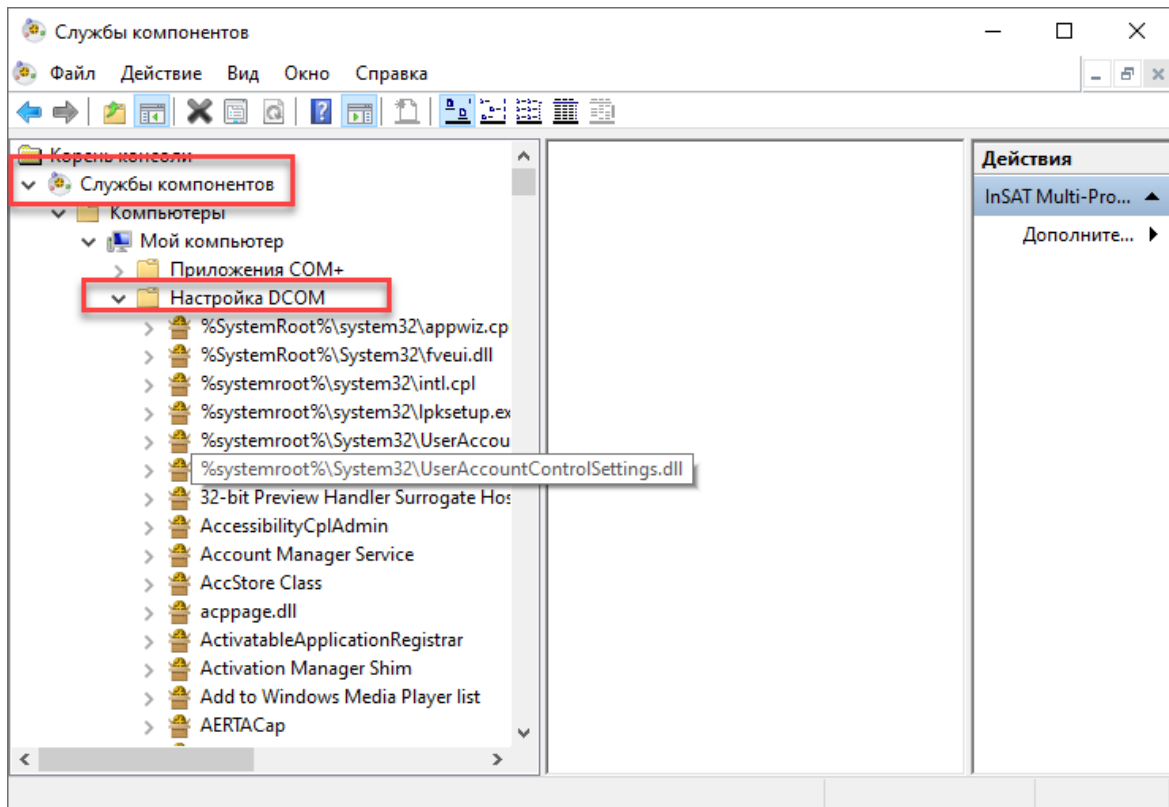
Для работы с удаленными OPC DA и OPC HDA серверами необходимо настроить DCOM Windows. Удаленным считается OPC-сервер, который запускается не на том компьютере, где установлена исполнительная система.

Для этого командной строке Windows или в приложении Выполнить, которое, как правило, открывается комбинацией клавиш Win + R наберите dcomcnfg.

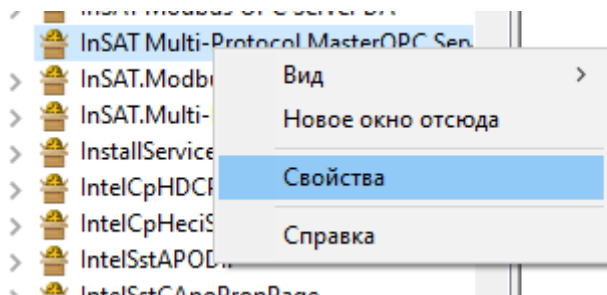
Откроется окно Службы компонентов:



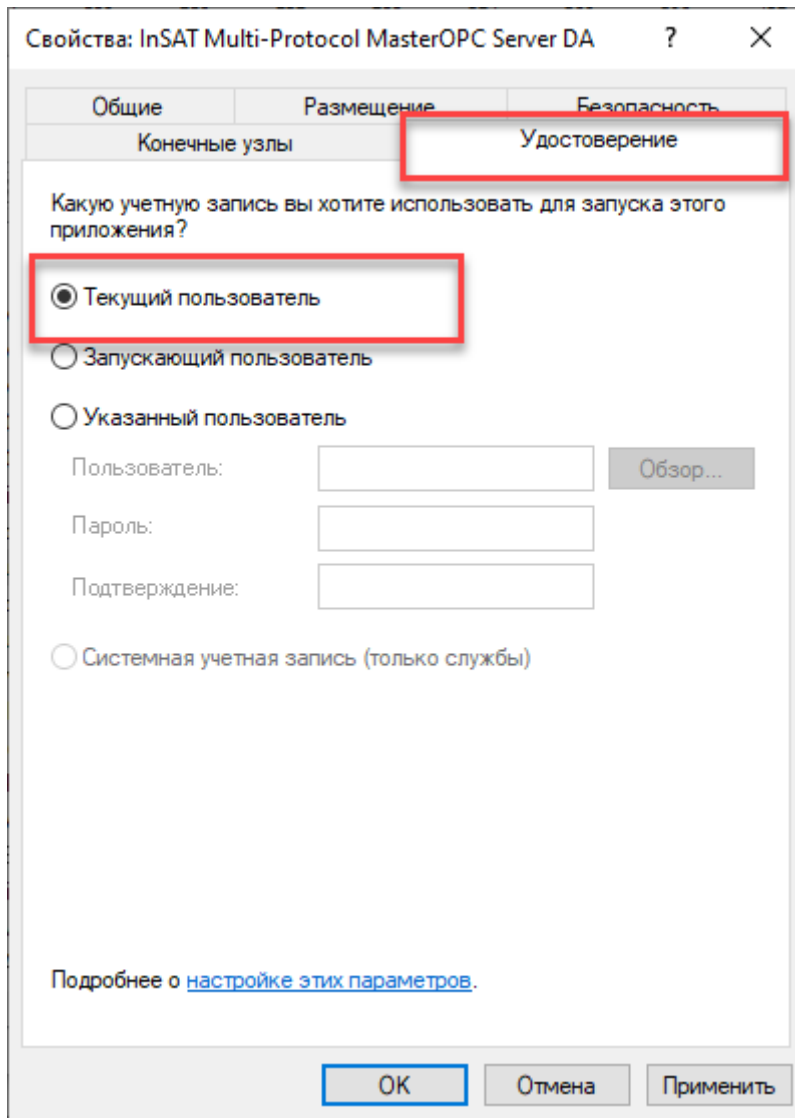
В правой части окна, в дереве необходимо выбрать элемент Службы компонентов - Компьютеры - Мой компьютер - Настройка DCOM:



Далее в этом списке требуется найти имя OPC-сервера, с которым нужно установить связь. В контекстном меню элемента выберите пункт Свойства:



В открывшемся диалоговом окне, на вкладке Удостоверение нужно указать учетную запись Текущий пользователь:



Изменения вступят в силу, после нажатия на кнопку ОК.

Возможно могут потребоваться и другие настройки DCOM, которые могут быть рекомендованы производителями OPC-серверов и OPC Foundation.

6.10.3. СТРУКТУРА ТАБЛИЦ БАЗ ДАННЫХ С АРХИВАМИ

Для хранения архива данных и архива сообщений используются свои базы данных. В обеих базах есть таблица projects, в которой хранится информация о проектах подключенных к БД

Таблица projects

Название	Назначение
id	Первичный ключ
name	Имя проекта
guid	Уникальный GUID проекта, который

Архив данных

База данных, в которой хранятся архивы данных MasterSCADA 4D имеет следующие таблицы: projects, items, raw_data, sys_props. Таблицы создаются автоматически при первом запуске исполнительной системы. При изменении списка архивируемых параметров таблицы будут дописываться. Уже существующие записи удаляться не будут, за исключением случаев, которые указаны в настройках проекта.

Назначение таблиц баз данных:

Название	Назначение
projects	Информация о проектах подключенных к БД.
items	Информация об архивируемых параметрах
raw_data	Непосредственно таблица с архивными данными
sys_props	Служебная таблица, хранит версию структуры БД и другие служебные свойства.

Таблица items

Название	Назначение
id	Первичный ключ
project_id	Внешний ключ для связи с таблицей projects
itemid	ID параметра в проекте. Формируется автоматически и указывается в соответствующем свойстве категории Служебные

Название	Назначение
path	Путь к полю параметра или к параметру внутри экземпляра объекта
name	Полное иерархическое имя параметра в проекте
first_time	Время первого значения (FILETIME).
last_time	Время последнего значения (FILETIME)
count	Показывает число записей в базе
type	ОрсУа код типа значения параметра

Таблица rawdata

Название	Назначение
layer	ID слоя данных: 0 - базовый слой 1- минутный слой 2- часовой слой
archive_itemid	Внешний ключ для связи с таблицей item
source_time	Время формирования значения (FILETIME)
server_time	Время сервера, в настоящее время не используется (FILETIME)
status_code	ОрсУа код статуса данных
value	Значение числовых данных (double). В SQLite3 хранит все возможные типы данных.
s_value	Данные типа string. Отсутствует в SQLite3

Название	Назначение

Таблица sys_props

Название	Назначение
name	Имя параметра
value	Значение параметра

При просмотре баз данных сторонними средствами, для перевода значений типа данных FILETIME в более удобные для восприятия, можно использовать произвольные утилиты из сети Интернет.

Архивы сообщений

Для формирования архива сообщений формируются таблицы: projects, events_alarms, events_recs, events_alarms_fields, events_recs_fields.

Назначение таблиц баз данных:

Название	Назначение
projects	Информация о проектах подключенных к БД.
events_alarms	Информация о всех типах сообщений в проекте
events_recs	Архив сообщений
events_alarms_fields	Дополнительные поля сообщений (добавляемые пользователем в проекте у тревог или ФБ FireBaseEvent)
events_recs_fields	Значения дополнительных полей сообщений (добавляемые пользователем в проекте)

Таблица events_alarms

Название	Назначение
id	Первичный ключ
project_id	Внешний ключ для связи с таблицей projects
itemid	<p>ID параметра источника сообщения в проекте. Формируется автоматически и указывается в соответствующем свойстве категории Служебные</p> <ul style="list-style-type: none"> • для тревог - ID тревоги • для сообщений от шкалы параметра - Id параметра <ul style="list-style-type: none"> • для сообщений, выданных из ФБ - ID ФБ • для сообщений безопасности и контроля целостности - ID элемента Безопасность • для сообщений о действиях пользователя - ID элемента, с которым совершается определенное действие
path	Путь к источнику сообщения внутри экземпляра объекта
name	Полное иерархическое имя источника сообщения в проекте
type_id	<p>Подтип сообщения. Возможные варианты</p> <ul style="list-style-type: none"> • для тревоги = 0. • для сообщений от шкалы параметра определяется типом уставки (HiHi = 1, Hi = 2, Lo = 3, LoLo = 4, RateOfChange = 5) • при выдаче сообщения через FirebaseEvent определяется входом EventTypeId • для системных сообщений и сообщений безопасности определяется типом сообщения. Старшие 16 бит определяют категорию - <ul style="list-style-type: none"> 1 - сообщения о действиях пользователя 2 - сообщения об операциях изменения параметров безопасности, изменения списка пользователей (настроек и паролей), сообщения ошибки защиты или контроля целостности 3 - системные сообщения

Таблица events_recs

Название	Назначение
id	Первичный ключ
alarm_id	Внешний ключ для связи с таблицей events_alarms
time	Время последнего изменения состояния сообщения. Тип FILETIME
update_type	Тип изменения: 1 - Активирована 2 - Деактивирована (для условных сообщений) 3 - Квитировано
active_time	Время активации сообщения. Тип FILETIME
in_active_time	Время деактивации сообщения. Тип FILETIME
acked_time	Время квитирования сообщения. Тип FILETIME
acked	Состояние квитирования сообщения. Тип BOOL.
active	Состояние активности сообщения. Тип BOOL.
severity	Приоритет сообщения
message	Текст сообщения
comment	Комментарий оператора, заданный при квитировании сообщения (также может быть задан параметром при генерации сообщения с использованием тревоги или ФБ)
user_address	Ip адрес узла, на котором был открыт клиент визуализации, в котором квитировали сообщение.
user_name	Имя пользователя, который квитировал сообщение.

Таблица events_alarms_fields

Название	Назначение
id	Первичный ключ
name	Имя дополнительного параметра сообщения

Таблица events_recs_fields

Название	Назначение
event_rec_id	Внешний ключ для связи с таблицей events_recs
field_id	Внешний ключ для связи с таблицей events_alarms_fields
value	Значение параметра

6.11. РЕКОМЕНДАЦИИ ПО СОЗДАНИЮ ПРОЕКТОВ

Состав раздела:

Формирование отчета с помощью ФБ READ_ARCHIVE_DATA

6.11.1. ФОРМИРОВАНИЕ ОТЧЕТА С ПОМОЩЬЮ ФБ READ_ARCHIVE_DATA

Содержание статьи:

1. Общее описание
2. Постановка задачи
3. Генерация архивного значения
4. READ_ARCHIVE_DATA
5. Массив структур типа SYSTEM_*_PARAM
6. Таблица данных. Условное выделение ячеек
7. Выгрузка данных в .CSV файл с помощью ФБ StringToFile

1. Общее описание.

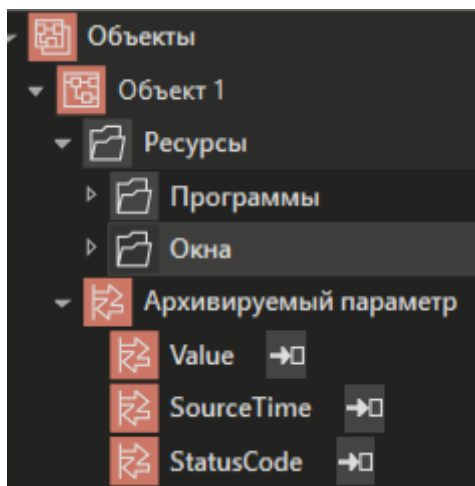
На большинстве контроллеров формирование отчетов, с использованием стандартного редактора отчетов, технически невозможно, поэтому если стоит задача отобразить архивные данные в табличном виде на контроллере, то придется использовать ФБ READ_ARCHIVE_DATA (далее RAD) с выводом результата в графический элемент Таблица данных и записью в сторонний .csv файл. Этот пример может быть также полезен для дополнительного освоения MasterSCADA 4D в других прикладных задачах.

2. Постановка задачи.

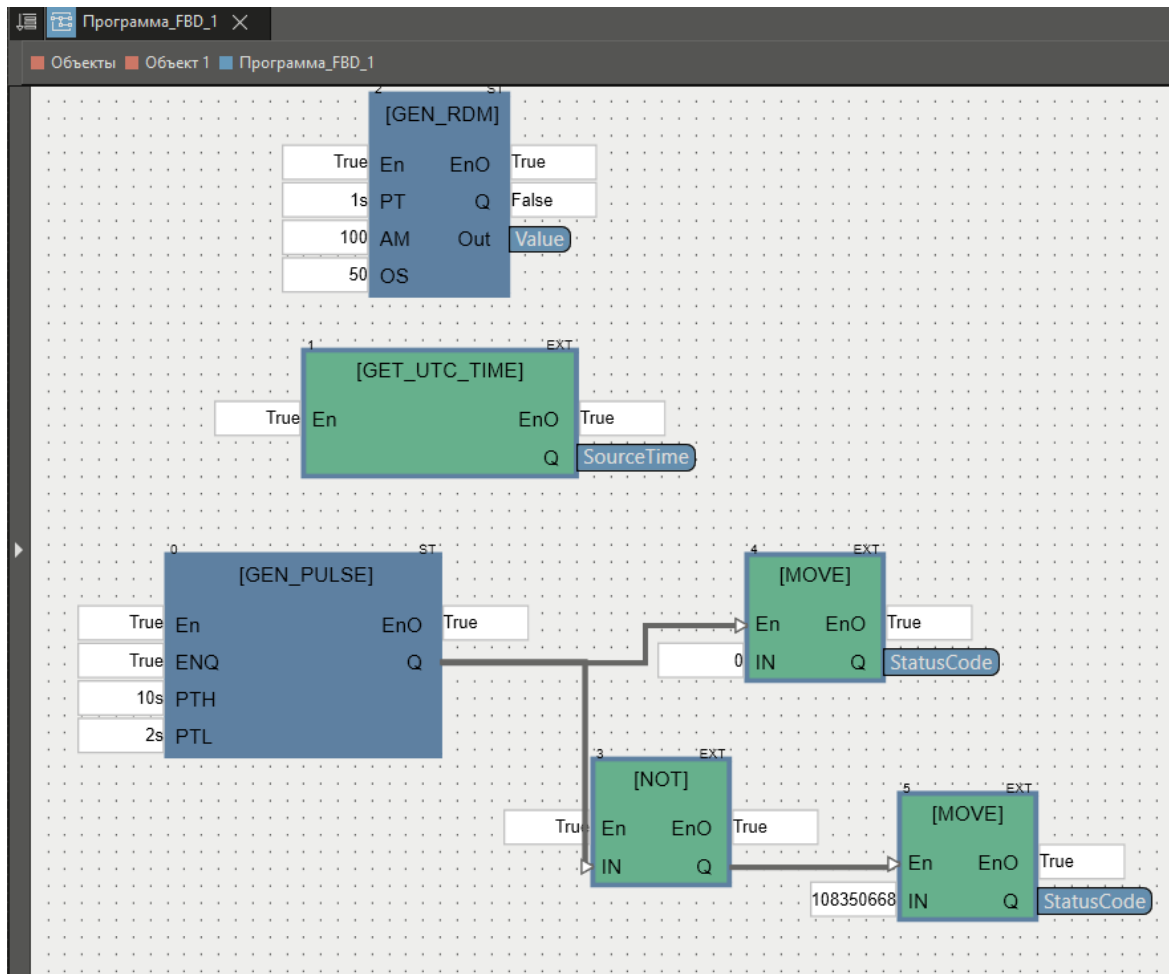
Сгенерировать архивные значения параметра объекта, при этом формируя недостоверный признак качества. Отобразить этот параметр на графике, выделив аварийный диапазон значений. Использовать начало и конец отображаемого интервала тренда для формирования массивов значений, признаков качества и меток времени с помощью RAD. Полученные массивы собрать в единый массив структур и вывести его на таблицу данных. В таблице выделить ячейки с аварийными значениями, а также строку с недостоверным признаком качества. Также этот массив нужно записать в сторонний .csv файл.

3. Генерация архивного значения

В проекте есть архивируемый параметр типа SYSTEM_LREAL_PARAM с настройками архивирования по умолчанию. Нужно задать значения полей Value, SourceTime и StatusCode.



Делаем это с помощью генераторов сигналов из библиотеки OSCAT:



Т.е. значение меняется произвольно от 0 до 100 раз в 1 секунду, метка времени формируется с помощью функции GET_UTC_TIME, а признак качества будет недостоверным в течении 2 секунд.

У параметра установим настройки шкалы AI, будем использовать верхний и нижний аварийные значения:

Архивируемый параметр

SourceTime	0001-01-01-00:00
StatusCode	16#80320000

Шкала

Формат	
Единица измерения	
Максимум	100
Минимум	0
HiHi	90
Использовать HiHi	<input checked="" type="checkbox"/>
Hi	0
Использовать Hi	<input type="checkbox"/>
Lo	0
Использовать Lo	<input type="checkbox"/>
LoLo	10
Использовать LoLo	<input checked="" type="checkbox"/>
Гистерезис	0
Максимальная скорость изменения	0
Время задержки	0ms
Шкала	Шкала AI

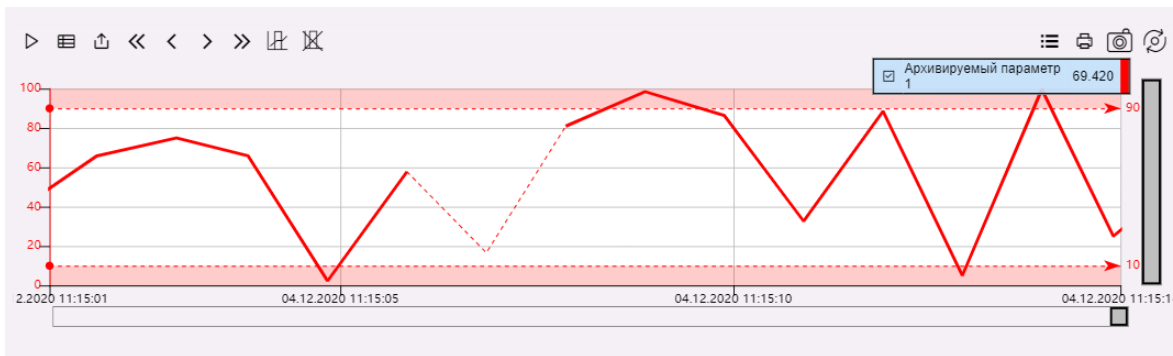
Отношения

Архивирование

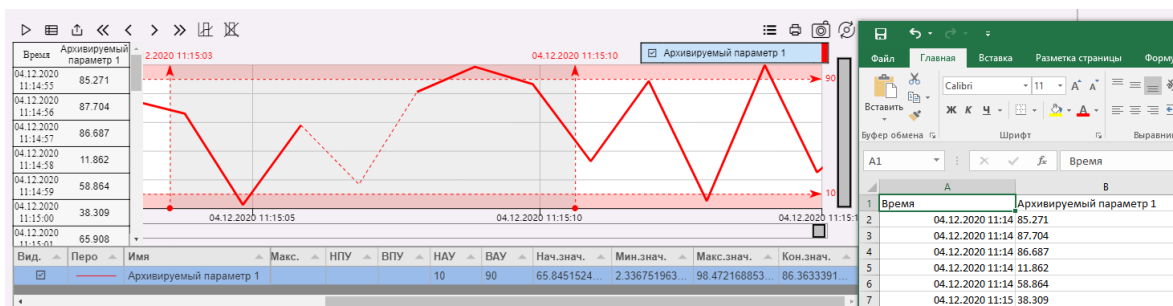
Архивировать	<input checked="" type="checkbox"/>
Шаблон архивирования	

Нужно обработать полученные архивные данные, отобразить эти данные в табличном виде и экспортировать эти данные. Для решения такой задачи обычно используется стандартный редактор отчетов, но на контроллерах сформировать их нельзя. В таком случае можно воспользоваться графическим элементом Тренд. На стартовое окно из палитры нужно вытащить тренд и архивный параметр перетащить на график.

Получим результат:



Тренд позволяет вывести архивные значения за отображаемый интервал, экспортировать эти данные в .csv, а с помощью легенды тренда можно получить максимальное и минимальное значения за отображаемый период.

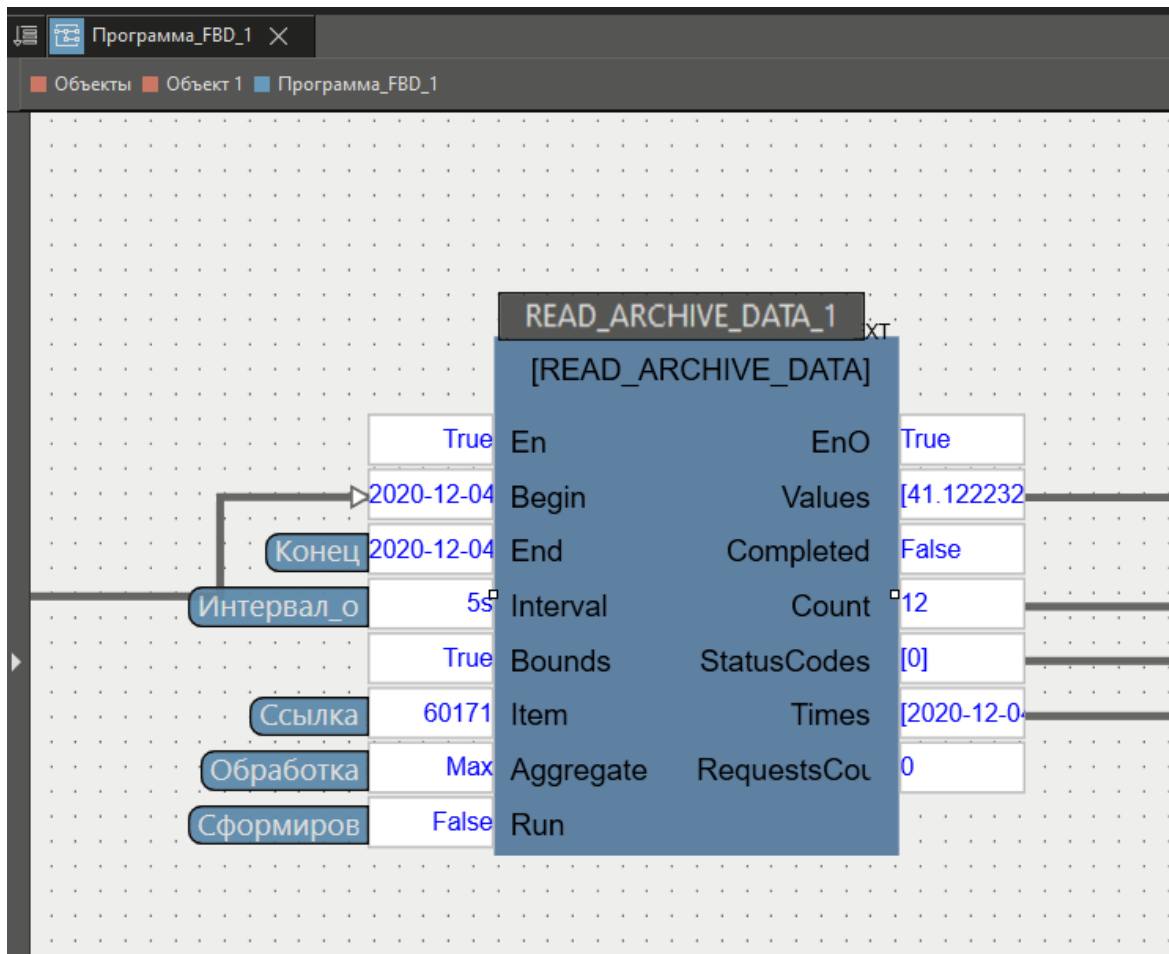


Если на данном этапе для вас и заказчика проблема решена - то можно смело дальше не читать. Но если: тренд вообще не предполагался, выводимая таблица слишком маленькая и без выделения всех ячеек, в таблице нет признака качества, обработок в легенде недостаточно и вообще требуется построить аналог периодического отчета, а не отчета по изменению - приступаем к следующему пункту.

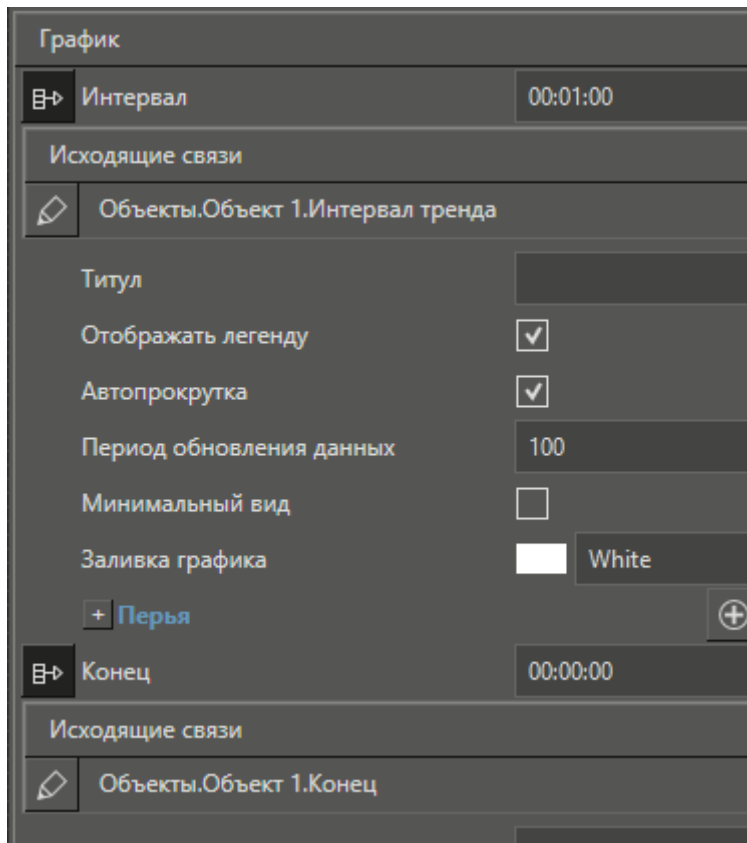
4. READ_ARCHIVE_DATA

ФБ READ_ARCHIVE_DATA позволяет получить массив значений, массив меток времени и массив признаков качества за определенный промежуток времени, а также выполнить обработку архивируемых значений за интервал.

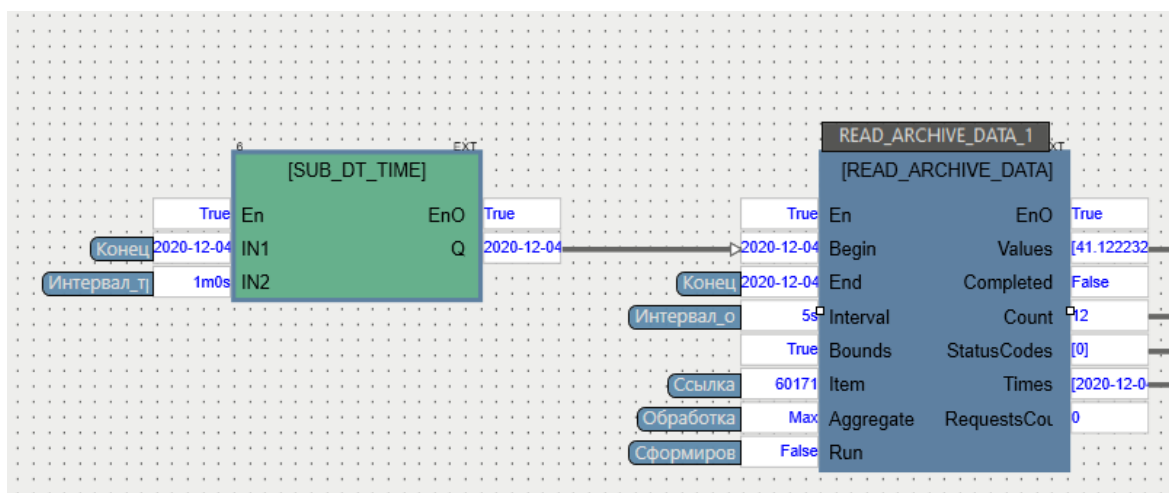
Вид блока в редакторе FBD:



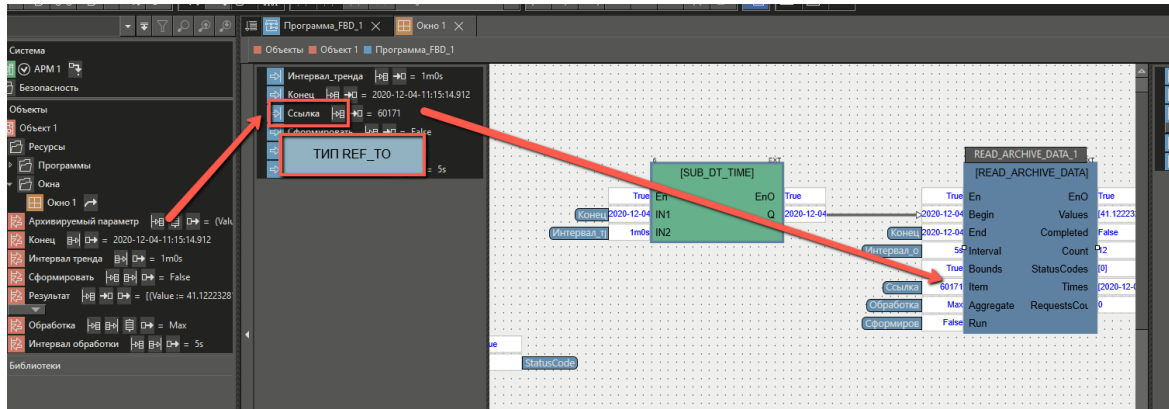
В нашем примере начало и конец будем брать из графика, так как тренд может возвращать по исходящей связи значения интервала и конца:



Здесь мы это делаем для удобства, в вашем проекте будут свои начало и конец для RAD. Тренд не возвращает свойство Начало, поэтому его мы посчитаем отдельно в программе:

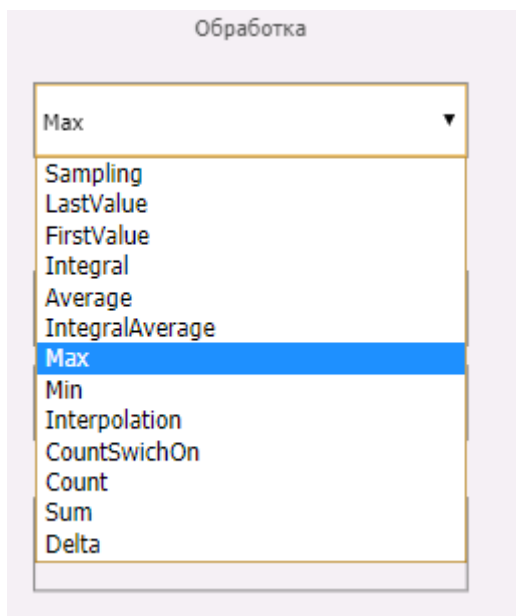


Далее нужно задать параметр, по которому будем производить запрос в базу. Для этого нужно задать входную переменную Item. Это можно сделать вручную, узнав id архивного параметра в служебных свойствах, либо создать входную переменную программы Ссылка типа REF_TO (этот тип возвращает id элемента, с которым связан), связать с архивным параметром и ножкой Item ФБ RAD:

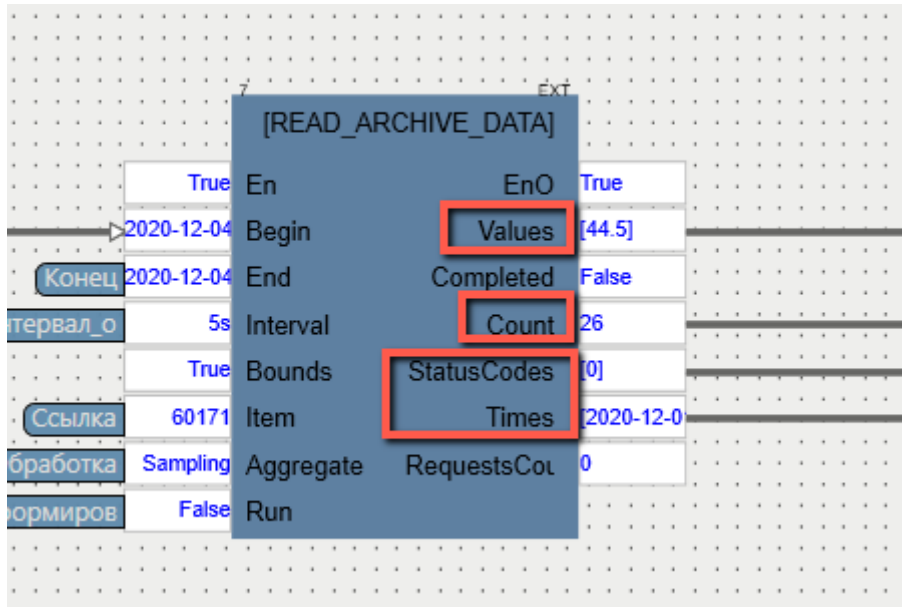


Далее определим, как обрабатывать архивные значения параметров с помощью входа **Aggregate**. По умолчанию вход имеет значение **Sampling**. В таком случае после вызова ФБ вернется массив архивных значений без какой-либо обработки, т.е. результатом будет та же таблица, которую формирует тренд, или отчет по изменению. Но если стоит задача провести обработку архивных значений за небольшие интервалы, т.е. построить периодический отчет, то нужно поменять значения входа **Aggregate** и **Interval**.

У нас есть архивные значения за большой промежуток времени, например, расход электричества за год, и нужно посчитать среднее потребление в месяц. В таком случае выбирается значение входа **Aggregate** - **Average**, а интервал 1 месяц, разница между **Begin** и **End** будет составлять один год. Результатом будет массив из 12 значений, и каждое значение будет среднее за интервал в 1 месяц, т.е. система автоматически просуммирует все значения и поделит на число таких значений. Тут важно еще следить за входом **Bound**, чтобы учитывать значения на границах интервалов. В примере мы с вами посмотрим все возможные обработки, поэтому **Aggregate** будем менять с помощью выпадающего списка на окне.



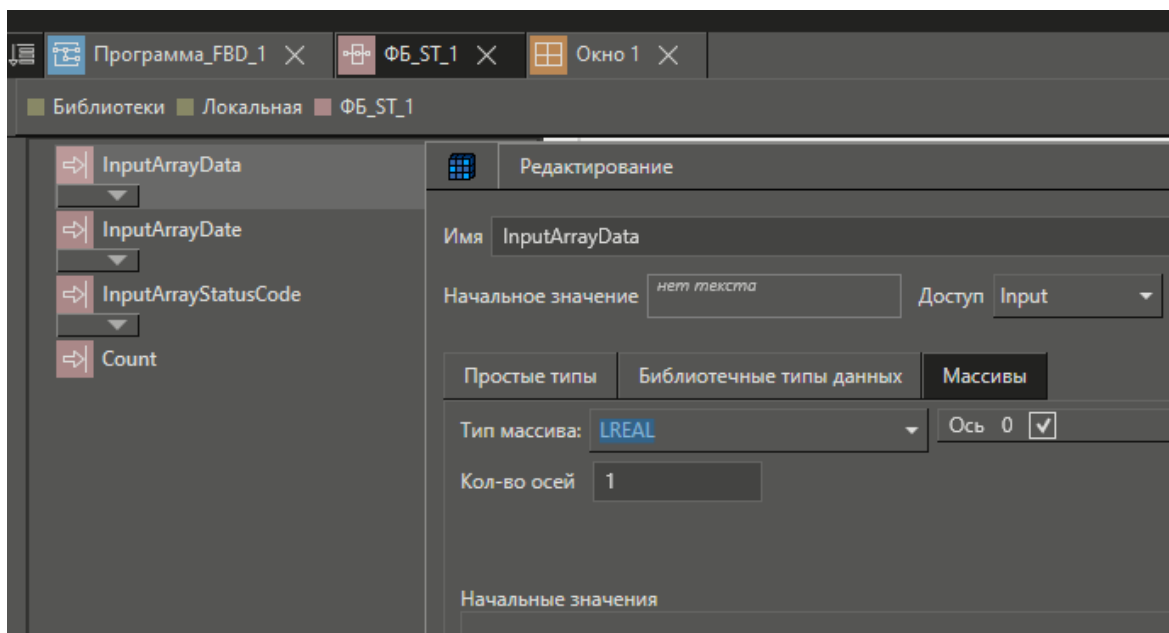
После вызова этого ФБ с помощью задания **True** входа **Run** (на окне обычной кнопкой) получим нужные нам массивы, а также количество элементов в этих массивах:



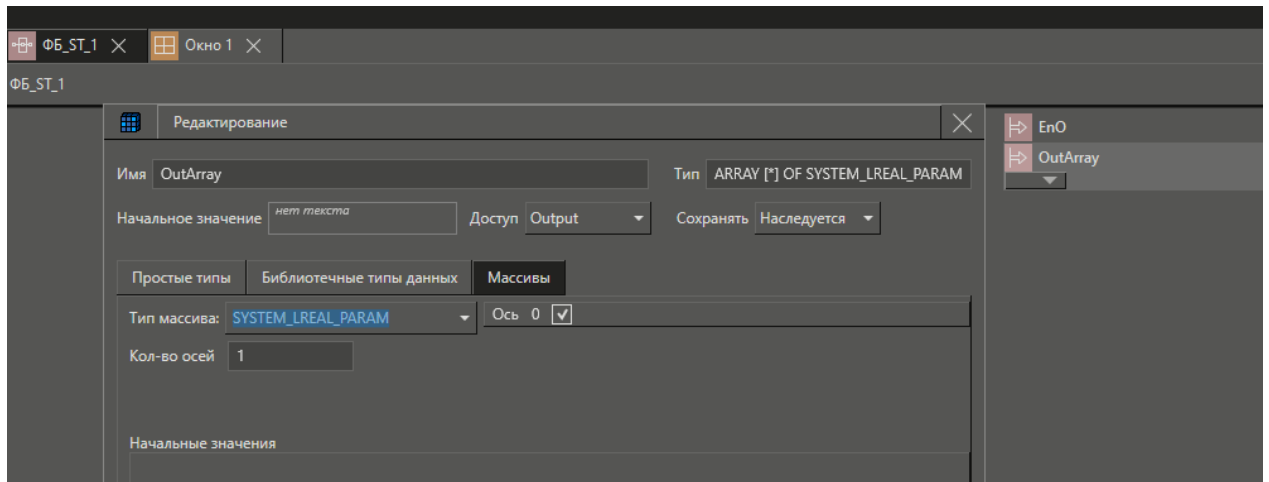
Далее с массивами можно делать любые операции. В частности, если текущих обработок недостаточно и у вас свои уникальные расчеты, то далее нужно писать свой ФБ, который бы производил эти вычисления с полученными данными. Нам также нужно это сделать, нам нужно собрать массив значений, меток времени и признаков качества в единый массив структур, так как источником данных для таблицы является именно массив структур.

5. Массив структур типа SYSTEM_*_PARAM

В локальную библиотеку добавим ФБ на языке ST. Входными переменными будут динамические массивы типа LREAL, DT и UDINT, так как количество значений, возвращаемых RAD, разное после каждого вызова ФБ:



Также входной переменной будет число элементов массива, а результатом ФБ - массив структур типа SYSTEM_LREAL_PARAM:



Тело программы:

```
OutArray := CREATE_ARRAY(INIT:= Переменная_2, SIZE:= Count); //каждый раз создаем
новый массив с новой длиной
```

```
for i := 1 to Count do
```

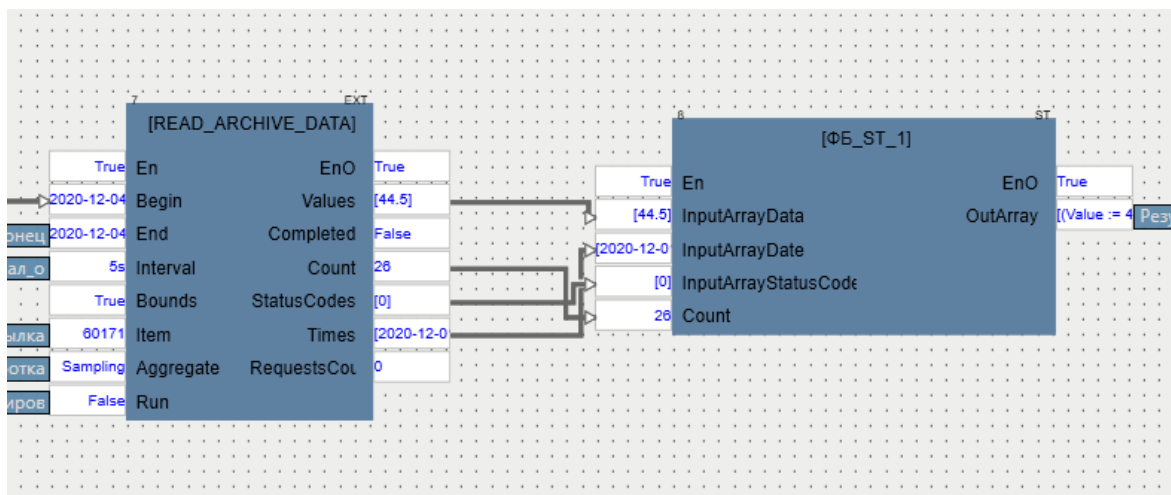
```
OutArray[i].Value := InputArrayData[i]; //простое присвоение
```

```
OutArray[i].SourceTime := InputArrayDate[i];
```

```
OutArray[i].StatusCode := InputArrayStatusCode[i];
```

```
END_FOR;
```

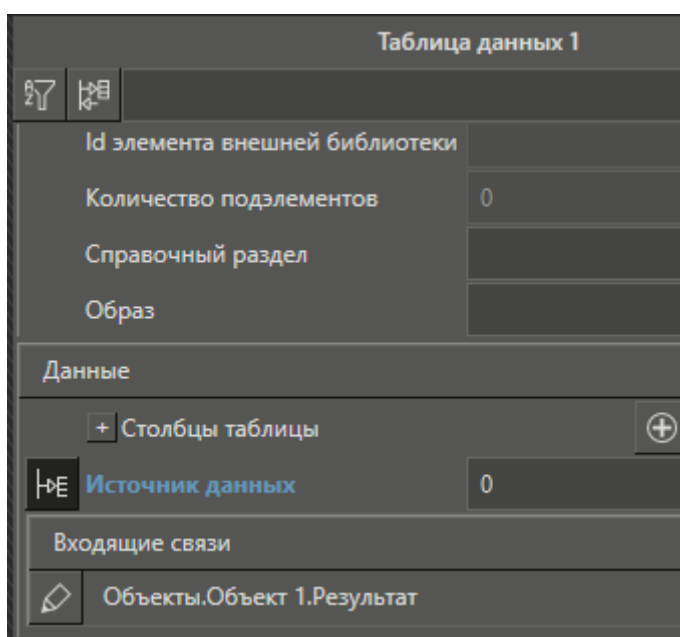
И экземпляр этого ФБ используемой в основной программе:



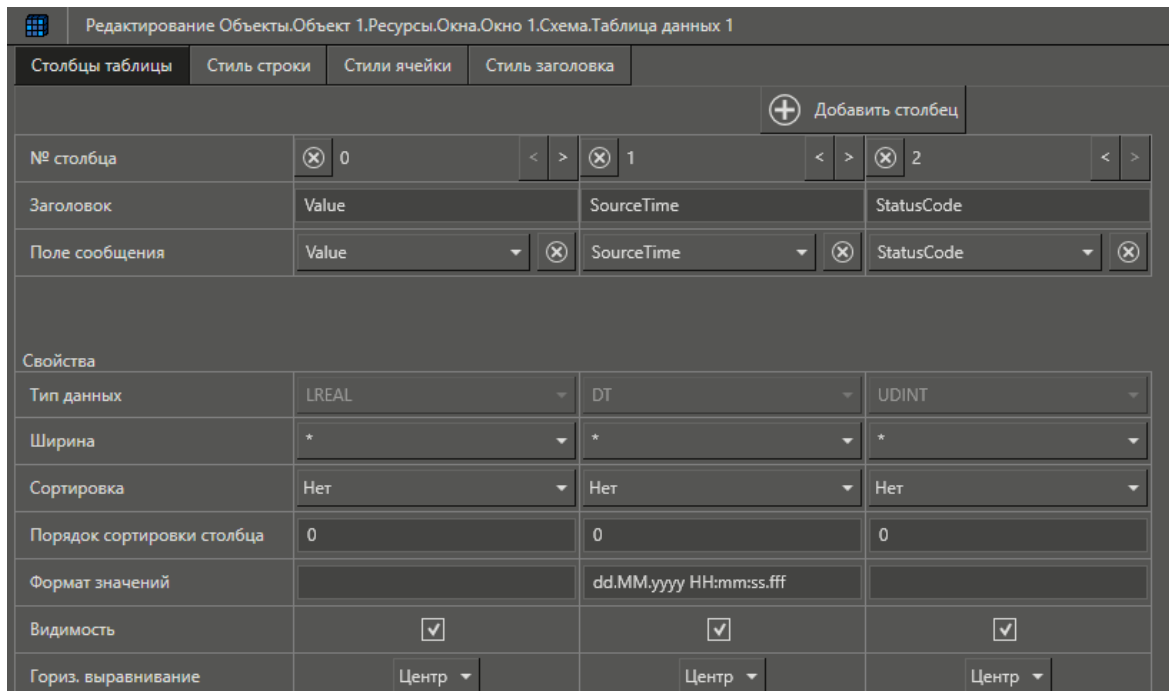
OutArray будем использовать в качестве источника для таблицы данных.

6. Таблица данных. Условное выделение ячеек

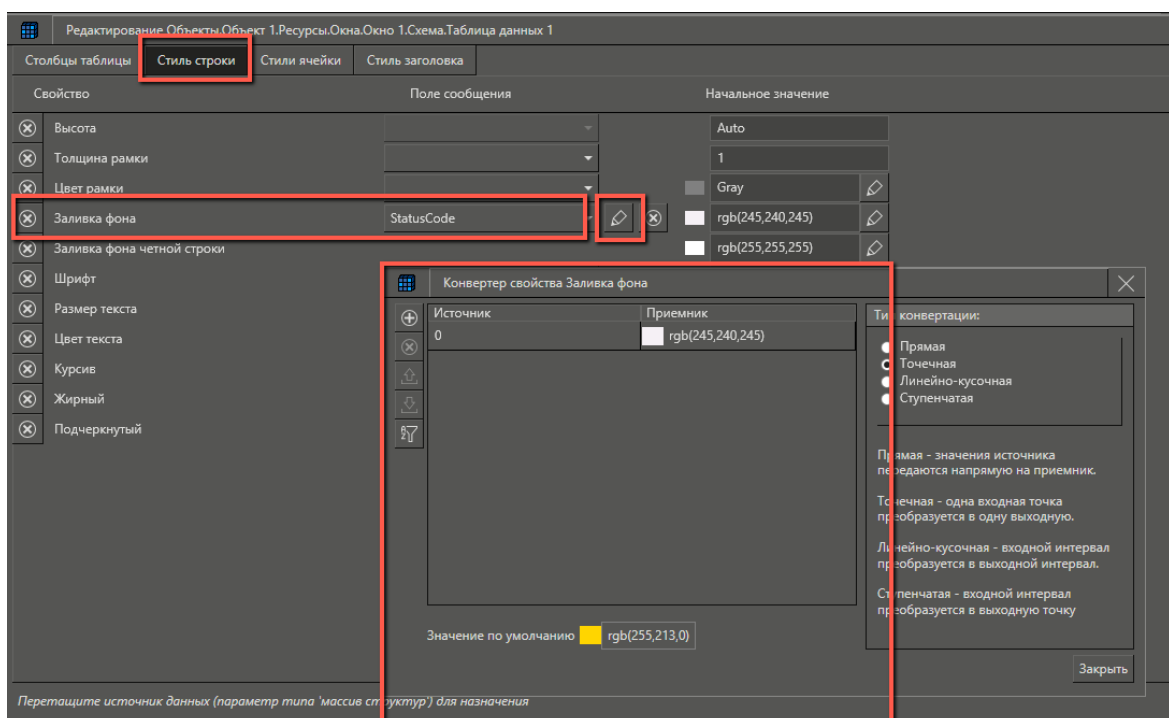
На окно вытащим графический элемент Таблица данных и установим связь свойства Источник данных этой таблицы с параметром типа динамический массив структур SYSTEM_LREAL_PARAM:



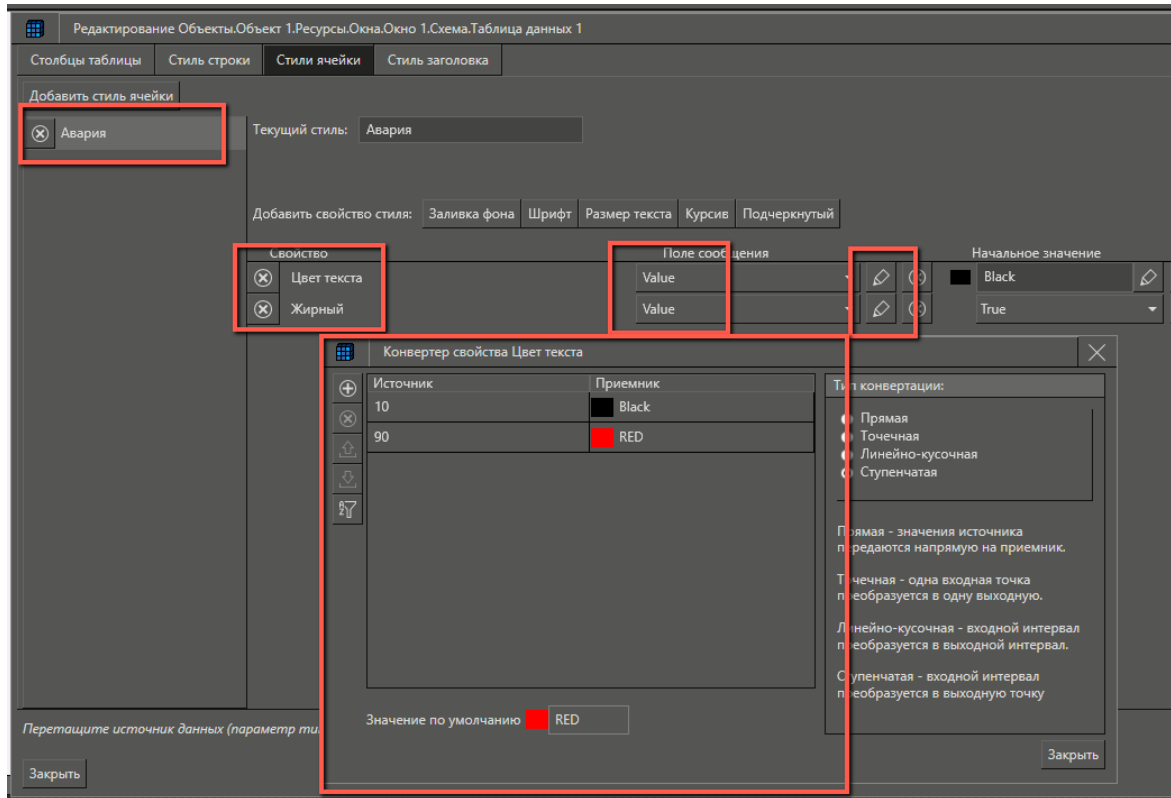
Дважды совершим клик по таблице и в редакторе таблицы зададим столбцы:



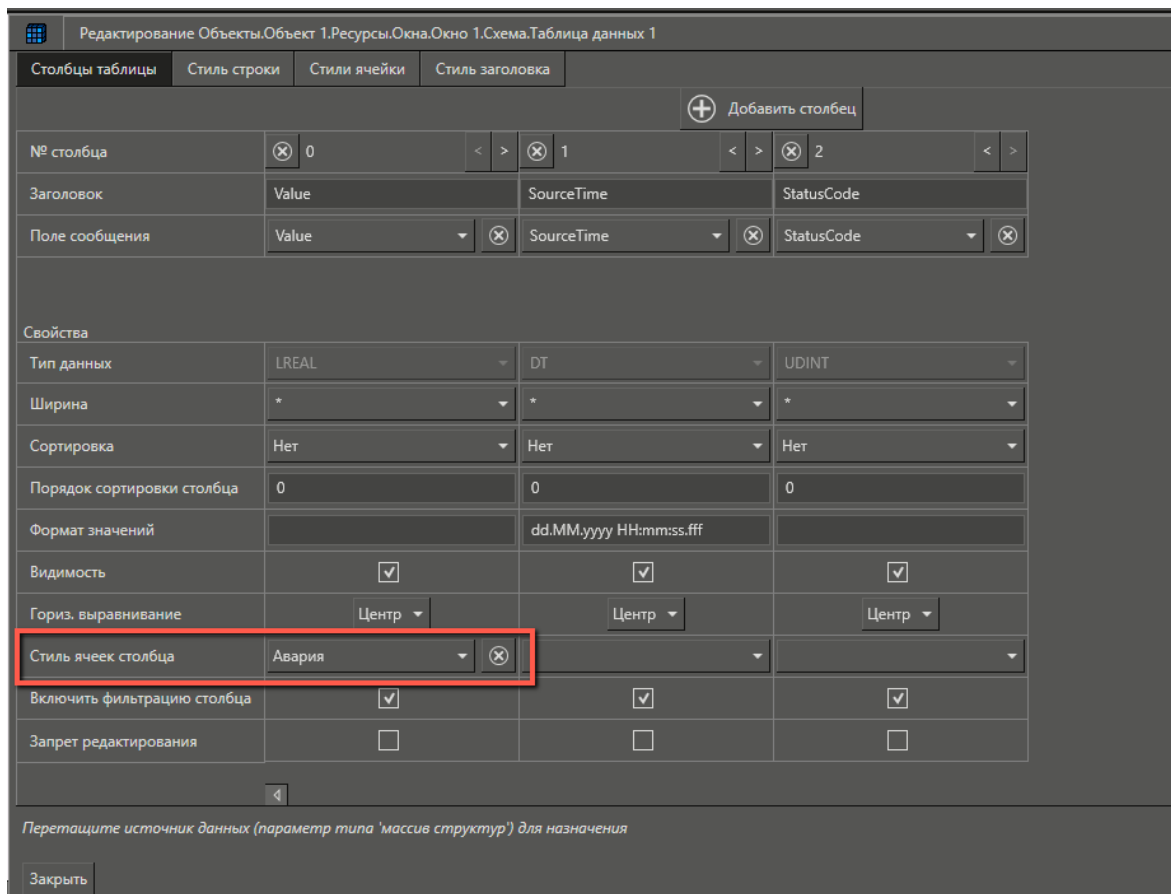
Необходимо выделить в таблице те ячейки, которые не попали в нормальный диапазон 10..90, а также выделить строки, соответствующие плохому признаку качества. Для начала перейдем в стиль строки и зададим точечную конвертацию по StatusCode:



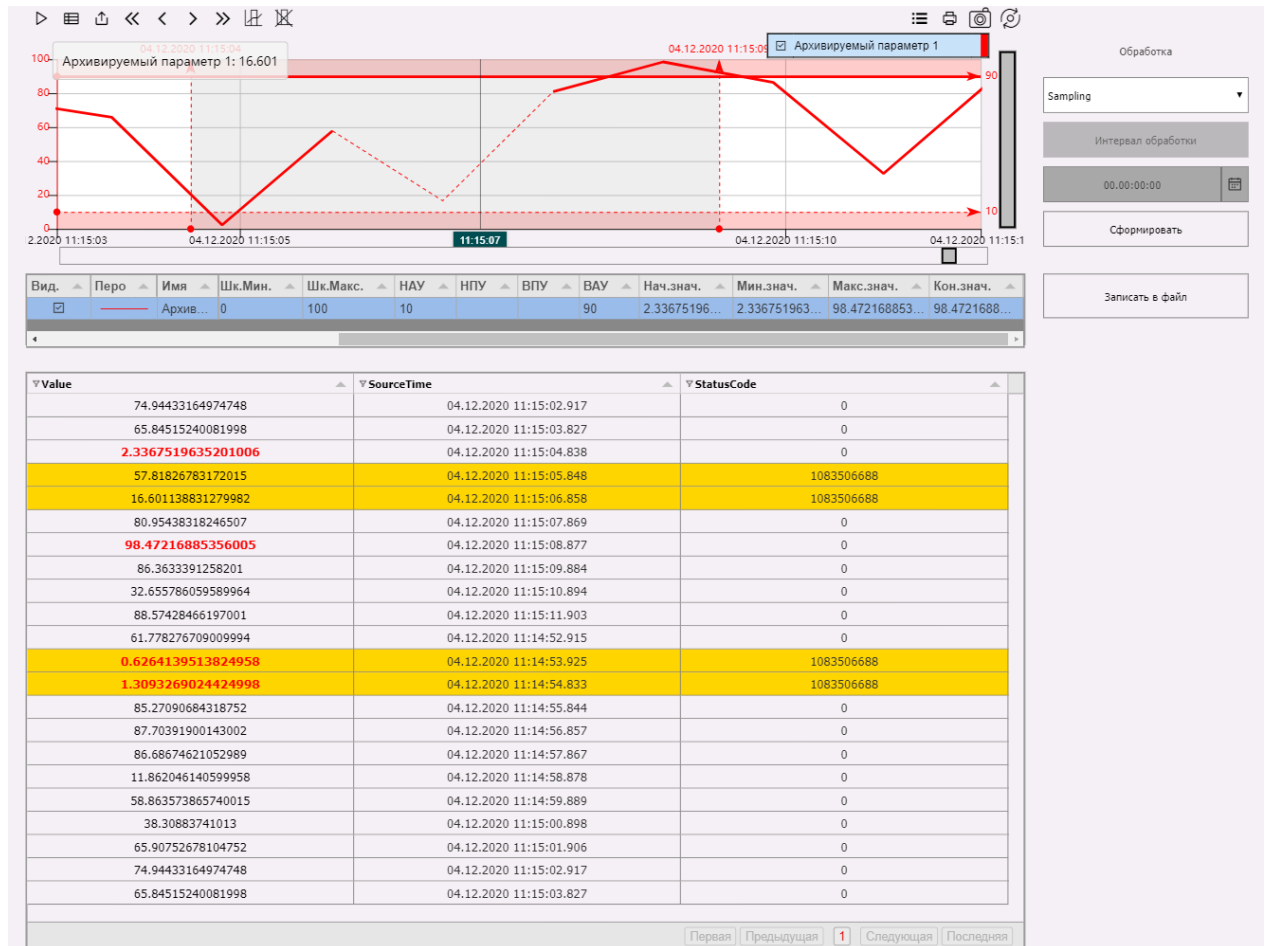
Затем в стиле ячейки создадим стиль Авария, в котором по значениям Value будем менять цвет текста и его жирность:



Вернемся в редактор столбцов таблицы и для столбца Value установим стиль ячеек Авария:



В результате в среде исполнения вы сможете задать обработку аварийных значений на схеме, а также выделить нужные значения в таблице данных:



7. Выгрузка данных в .CSV файл с помощью ФБ StringToFile

В примере выше есть один большой недостаток - эту таблицу не экспортировать и не распечатать, на момент написания этого документа такая задача еще стоит в очереди на разработку. Но есть способ записать массив структур в сторонний .csv с помощью ФБ StringToFile. Создадим программу:

```
count := UPPER_BOUND(ARR:= Результат, DIM:= 1);
```

```
a_1 := Результат[1].Value;
```

```
b_1 := Результат[count].Value;
```

```
if a_1 <> a and b_1 <> b THEN
```



```

    Переменная_7 := Переменная_8;

    a := Результат[1].Value;

    b := Результат[count].Value;

ELSIF a = a_1 and b = b_1 and i <> count THEN

    for i := 1 to count do

        Переменная_1 := CONCAT(IN1:= LREAL_TO_STRING(Результат[i].Value) , IN2:=
Переменная_2);

        Переменная_4 := CONCAT(IN1:= DT_TO_STRING(Результат[i].SourceTime), IN2:=
Переменная_2);

        Переменная_5 := CONCAT(IN1:= UDINT_TO_STRING(Результат[i].StatusCode),
IN2:= "");

        Переменная_6 := CONCAT(IN1:= CONCAT(IN1:= CONCAT(IN1:= Переменная_1,
IN2:= Переменная_4) , IN2:=Переменная_5) , IN2:=Переменная_3);

        Переменная_7 := CONCAT(IN1:= Переменная_7, IN2:= Переменная_6);

    END_FOR;

END_IF;

StringToFile_1(Input:= Переменная_7, FileName:="C:\Users\User\Desktop\43.csv" , Write:=
Запись);

```

Что здесь происходит?

count := UPPER_BOUND(ARR:= Результат, DIM:= 1); - получаем количество элементов массива структур

a_1 := Результат[1].Value; - получаем первый элемент массива структур

b_1 := Результат[count].Value; - получаем последний элемент массива структур

Эти значения нам нужны для того, чтобы затем обнулять результат записи в файл.

```

if a_1 <> a and b_1 <> b THEN

```

```

    Переменная_7 := Переменная_8;

```

```

    a := Результат[1].Value;

```

```
b := Результат[count].Value;
```

Переменная_8 - пустая строка, Переменная_7 - результат, который надо записать в csv файл. Поэтому если мы поменяли интервал, то предыдущий результат надо обнулить.

ELSIF a = a_1 and b = b_1 and i <> count THEN - если же интервал не менялся, то можно задать значение переменной типа STRING

```
for i := 1 to count do
```

```
    Переменная_1 := CONCAT(IN1:= LREAL_TO_STRING(Результат[i].Value) , IN2:=  
    Переменная_2);
```

```
    Переменная_4 := CONCAT(IN1:= DT_TO_STRING(Результат[i].SourceTime), IN2:=  
    Переменная_2);
```

```
    Переменная_5 := CONCAT(IN1:= UDINT_TO_STRING(Результат[i].StatusCode),  
    IN2:= "");
```

```
    Переменная_6 := CONCAT(IN1:= CONCAT(IN1:= CONCAT(IN1:= Переменная_1,  
    IN2:= Переменная_4) , IN2:=Переменная_5) , IN2:=Переменная_3 );
```

```
    Переменная_7 := CONCAT(IN1:= Переменная_7, IN2:= Переменная_6);
```

```
END_FOR;
```

```
END_IF;
```

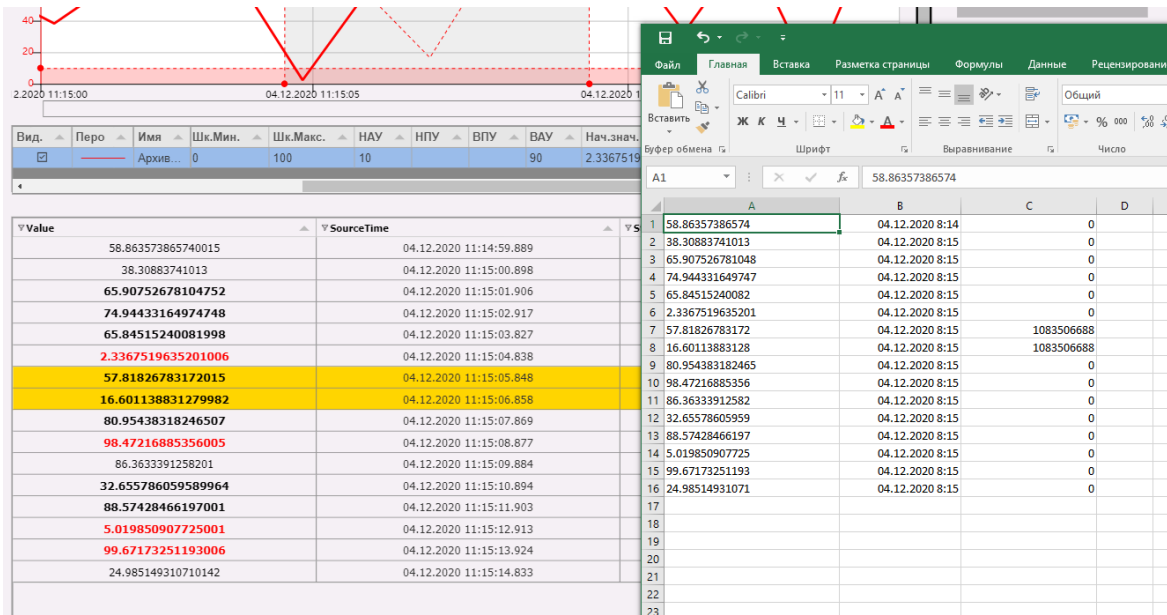
Почему так сложно?

Потому что разделителем между столбцами в csv является точка с запятой “;”, а переходом на следующую строку перевод каретки. Поэтому в цикле мы каждый раз приклеиваем сначала к Value значение SourceTime через точку с запятой, затем к этой новой строке приклеиваем значение признака качества, а уже потом к строке целиком добавляем перевод каретки. И так каждый элемент массива.

```
StringToFile_1(Input:= Переменная_7, FileName:="C:\Users\User\Desktop\43.csv" , Write:=  
    Запись);
```

Это вызов ФБ SrtingToFile - на вход идет строговое представление массива структур, далее нужно обязательно указать путь к файлу, куда пишем значения.

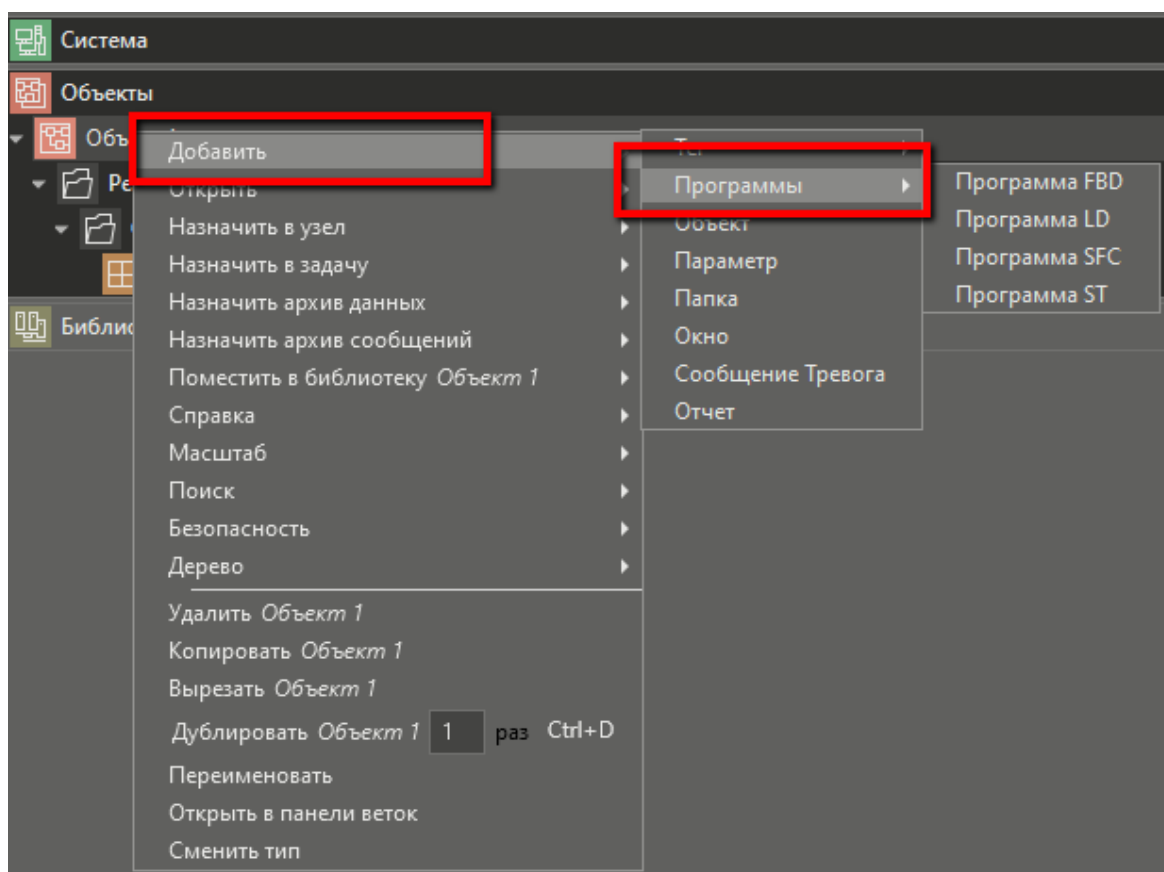
В результате получим:



7. ПРОГРАММИРОВАНИЕ В MASTERSCADA 4D

В MasterSCADA 4D реализована полноценная поддержка языков стандарта МЭК 61131-3, что подразумевает использование языков стандарта не только для программирования задач в контроллерах, но также для разработки логики управления и вспомогательных скриптов на всех уровнях системы, включая графические клиенты, в том числе интернет-клиенты.

Добавить программу в проект можно используя контекстное меню пункт Добавить. Программу какого-либо элемента (объекта, узла, канала, тега, окна и др.):



В зависимости от выбранного пункта контекстного меню откроется соответствующий программный редактор:

- Редактор FBD
- Редактор LD
- Редактор SFC
- Редактор ST

Основные понятия

Программа (термин стандарта IEC 61131-3) – это логическая совокупность (сборка) элементов и конструкций языка программирования, выполняющая требуемую обработку сигналов для обеспечения системы программируемых контроллеров функциями

управления исполнительными механизмами или процессами. Наследование программ, запрещено, допускается отношение с типом только Ссылается. Однако, в случае, если программа входит в состав объекта, который является наследником типа, а не его экземпляром, то программа будет иметь отношение Унаследован и может быть изменена.

Функция (термин стандарта IEC 61131-3) – элемент языка (POU), который во время выполнения обычно вырабатывает результат в виде одного элемента данных (в т.ч. массив или структуру) и, возможно, дополнительные выходные параметры. Имеет произвольное количество входных параметров. Функция не содержит внутренних переменных, значения которых сохраняются между вызовами, поэтому при одних и тех же значениях входов всегда производит одни и те же выходные значения.

Перегруженная функция - может оперировать с элементами входных данных различных типов в пределах родового типа данных. Например, функция ADD позволяет складывать параметры обладающие любыми числовыми типами данных. Выход функции будет определяться в зависимости от входов, например, если на входы функции ADD были поданы параметры с типами данных INT и REAL, то выход функции станет REAL, т.к. для определения выхода используется тот тип, который охватывает больший диапазон значений.

Функциональный блок (ФБ) (термин стандарта IEC 61131-3) – это программный компонент (POU), который вырабатывает произвольное количество выходных элементов данных. В отличие от функции значения всех выходов, а также некоторых внутренних переменных ФБ сохраняются между вызовами и при одних и тех же входных значениях ФБ может вырабатывать различные выходные значения. Поэтому в библиотеке вначале создается тип ФБ, а в программном компоненте создаются экземпляры этого ФБ (локальные переменные). В библиотеке допускается наследование типов ФБ.

Локальные переменные – параметры, которые находятся только в рамках какого-либо элемента: программы, ФБ, окна.

7.1. ОБЩИЕ ЭЛЕМЕНТЫ ПРОГРАММ

7.1.1. ПАРАМЕТРЫ ПРОГРАММ

Параметры для связи с другими элементами проекта

Добавление параметров

Как правило, параметры добавляются в клеммниках программы. Также могут быть добавлены в структуру программы в дереве.

Если перетащить параметры из какого-либо элемента проекта в программу (в клеммник, в дерево или в поле программ), то в программу добавятся параметры, имена которых будут совпадать с перетаскиваемыми и связаны с ними. Если имя перетаскиваемого параметра будет содержать спецсимволы, то они заменятся знаками

подчеркивания. Если перетащить папку с параметрами, то имена созданных параметров в программе будет иметь вид: [имя папки]_[имя параметра]

Имена параметров

Имена параметров программ задаются так же, как идентификаторы .

Имя переменной в дереве может быть задано с лидирующим или внутренними пробелами (не более одного пробела подряд). При трансляции пробелы заменяются знаками подчеркивания.

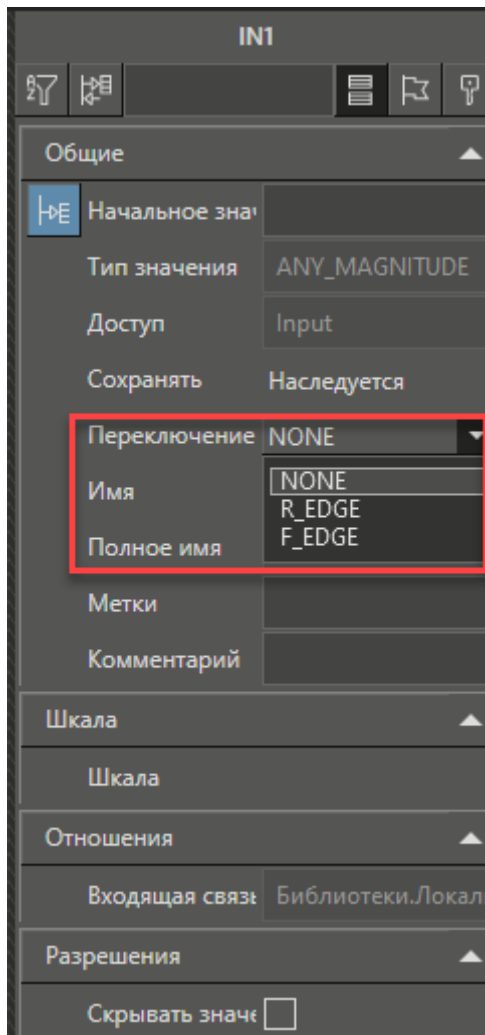
Переменные программного компонента (локальные переменные и аргументы) должны иметь уникальные имена в пределах компонента.

В качестве имен переменных нельзя использовать ключевые слова и спецсимволы

Свойства параметров программ

При добавлении параметров в программу открывается диалоговое окно настройки, такое же как и при создании параметров проекта.

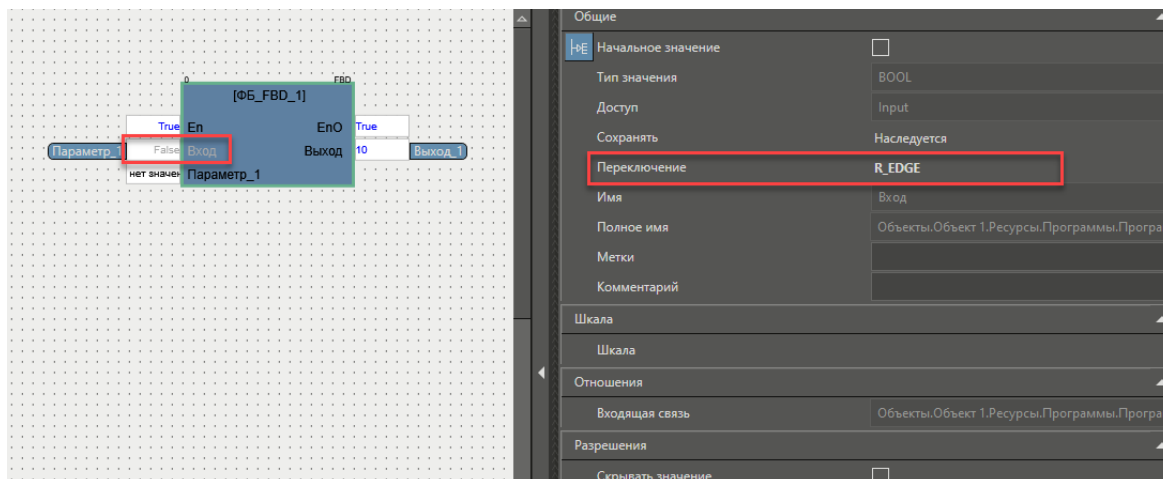
В отличие от параметров типа **BOOL** добавляемых в объект или в узел параметры программ имеют дополнительную настройку Переключение, которая может быть задана как в окне настройки, так и в панели свойств:



Настройка может принимать три значения:

- None- после получения значение не изменяется до следующего изменения связанного параметра
- R_EDGE - после получения TRUE параметр будет равен TRUE в течении одного цикла, затем изменится на FALSE, т.е. параметр будет работать таким образом, как и R_TRIG
- F_EDGE - после получения TRUE параметр изменяться не будет, параметр на один цикл станет равен TRUE только после того, как по связи придется изменение связанного значения с TRUE на False. Т.е. параметр будет работать таким образом, как и F_TRIG

Эта настройка может быть применена и к входам и выходам функциональных блоков, которые размещаются в программе, например, при редактировании схемы в редакторе FBD:



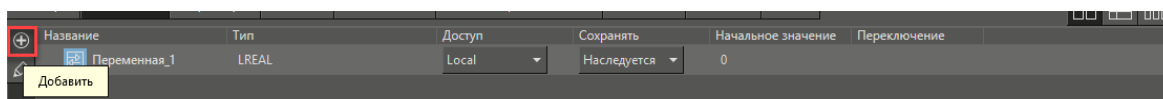
Тогда будет считаться, что между входным параметром и входом ФБ будет установлен R_TRIG

Локальные параметры

Локальные параметры это параметры программы, которые используются только внутри программы. Свойства локальных параметров, такие же как и у тех, что добавляются на клеммники программ, однако их нельзя связывать с внешними элементами проекта, они используются только внутри программ и не могут быть архивированы.

Добавить локальные параметры, а также просмотреть уже существующие можно в Легенде во вкладке Локальные или в структуре программы в дереве в группе Локальные.

Добавление на вкладке вкладки программы Локальные:



Свойства параметра можно настроить непосредственно во вкладке или в его панели свойств.

Настройка свойства Доступ

В программе у локальных переменных можно выбрать два типа доступа:

- Local - это обычные переменные, работают как и другие параметры проекта.
- Temp - это переменные, которые инициализируются в начале каждого вызова программы. Можно изменять значение такой переменной внутри программы, но на следующем вызове она снова будет иметь свое начальное значение.

Работа Temp переменных в зависимости от настройки Отладочная конфигурация:

- Флаг установлен - Temp переменные транслируются как Local, но в начале вызова программы им проставляются начальные значения.

- Флаг снят - Temp переменные объявляются в начале вызова программы и недоступны для отладки.

7.1.2. КЛЮЧЕВЫЕ СЛОВА

Ключевые (служебные) слова – это идентификаторы , зарезервированные для специального использования.

Список ключевых слов:

ACTION, END_ACTION, ARRAY, OF, AT, CASE, ELSE, END_CASE, CONFIGURATION, END_CONFIGURATION, CONSTANT, EN, ENO, EXIT, FALSE, F_EDGE, FOR, TO, BY, DO, END_FOR, FUNCTION, END_FUNCTION, FUNCTION_BLOCK, END_FUNCTION_BLOCK, IF, THEN, ELSIF, END_IF, INITIAL_STEP, END_STEP, NOT, MOD, AND, XOR, OR, PROGRAM, END_PROGRAM, NULL R_EDGE, F_EDGE, READ_ONLY, READ_WRITE, REPEAT, UNTIL, END_REPEAT, RESOURCE, ON, END_RESOURCE, RETAIN, NON_RETAIN, RETURN, STEP, STRUCT, END_STRUCT, TASK, TRANSITION, FROM, END_TRANSITION, TRUE, TYPE, END_TYPE, WHILE, END_WHILE, WITH.

Кроме того, ключевыми словами являются:

- лексемы, обозначающие типы данных (real, uint и т.п. – см. Элементарные типы данных);
- префиксы временных констант (dt, tod и т.п. – см. Временные константы).

Ключевые слова не чувствительны к регистру, т.е. while и WHILE являются идентичными ключевыми словами.

Рекомендации

Null

В редакторе поддержаны проверки = NULL и <> NULL.

NULL может появляться или при чтении запросами из баз данных, или при обращении к несуществующему индексу массива.

7.1.3. ОБРАБОТКА EN/ENO

Все функции, программы и функциональные блоки имеют параметры En и EnO

Если поставляемые или пользовательские функции, функциональные блоки или программы используются в другой программе, то при создании логики работы можно использовать вход En и выход EnO.

Вход En

Элемент будет исполняться, если на вход En поступит сигнал TRUE. Период исполнения блока будет зависеть от периода работы программы, в которой он находится и от способа ее вызова.

Важно! Вход En поддерживается только в графических языках. В языке ST не поддерживается.

Выход EnO

После выполнения элемента выход EnO ставится равным TRUE. Если в ходе выполнения алгоритма элемента возникла ошибка, то выход EnO устанавливается в False. Таким образом выход EnO можно использовать для диагностики работы элемента.

7.1.4. ТИПЫ ДАННЫХ

Разделяют следующие виды типов данных:

- Элементарные типы данных
- Производные типы данных
- Специальные типы данных

7.1.4.1. ЭЛЕМЕНТАРНЫЕ ТИПЫ ДАННЫХ

Для указания элементарного типа данных определены следующие ключевые слова:

- **BOOL** – булево значение, 1 бит (TRUE или FALSE, TRUE соответствует 1, 1 соответствует TRUE, FALSE соответствует 0, 0 соответствует FALSE);
- **SINT** – целое со знаком, 1 байт (-128 ... 127);
- **USINT** – целое без знака, 1 байт (0 ... 255);
- **INT** – целое со знаком, 2 байта (-32768 ... 32767);
- **UINT** – целое без знака, 2 байта (0 ... 65535);
- **DINT** – целое со знаком, 4 байта (-2147483648 ... 2147483647);
- **UDINT** – целое без знака, 4 байта (0 ... 4294967295);
- **LINT** – целое со знаком, 8 байт (-1.797693134862315e+308 ... 1.797693134862315e+308);
- **ULINT** – целое без знака, 8 байт;
- **TIME, DATE, TIME_OF_DAY, DATE_AND_TIME** – значения переменных этих типов задаются аналогично соответствующим временным константам (см. Временные константы);

- REAL – вещественное число, 4 байта (-3.402823466e+38 ... 3.402823466e+38);
- LREAL – вещественное число, 8 байт (-1.797693134862315e+308 ... 1.797693134862315e+308);
- STRING – строка 1-байтовых символов переменной длины;
- BYTE – битовая строка длиной 8 бит;
- WORD – битовая строка длиной 16 бит;
- DWORD – битовая строка длиной 32 бита;
- LWORD – битовая строка длиной 64 бита;

7.1.4.2. ПРОИЗВОДНЫЕ ТИПЫ ДАННЫХ

Допустимыми производными типами данных являются перечисление, диапазон, массив и структура.

Как и элементарные типы данных, производные типы могут быть использованы в декларациях переменных (см. Редактор переменной).

Перечисление

Элемент данных типа ПЕРЕЧИСЛЕНИЕ может принимать только те значения, которые определены в типе. Описание создания нового типа данных ПЕРЕЧИСЛЕНИЕ

Диапазон

Значение элемента данных типа ДИАПАЗОН может принадлежать только тому диапазону, который определен в типе. Описание создания нового типа данных ДИАПАЗОН

Массив

Массив – это совокупность элементов одного типа, имеющая общее имя (идентификатор).

Многомерный массив – это массив массивов.

При объявлении переменной типа МАССИВ резервируется место в памяти, достаточное для размещения всех элементов массива. Для обращения к элементам используется оператор индексирования элементов массива. Описание создания нового типа данных МАССИВ.

Структура

Структура – это совокупность элементов разного типа, имеющая общее имя (идентификатор).

Элемент данных типа СТРУКТУРА имеет структуру (набор подэлементов заданного типа), определенную в типе. Описание создания нового типа данных СТРУКТУРА

7.1.4.3. СПЕЦИАЛЬНЫЕ ТИПЫ ДАННЫХ

Специальные типы данных используются в функциях и функциональных блоках.

REF_TO

Считывает ID связанного параметра. Пусть вход функционального блока или функции имеет тип данных REF_TO, если связать такой вход с каким-либо элементом проекта, например, параметром, то по связи будет передаваться не текущее значение параметра, а его ID. Используется в ФБ работы с архивами WriteArchiveData и READ_ARCHIVE_DATA.

Если функциональный блок, содержащий вход или выход данного типа, находится внутри программы ST, то в программе можно указать полное имя архивируемого элемента.

Пример:

```
READ_ARCHIVE_DATA_11(Begin:=Start , End:= End, Interval:= , Bounds:= , Item:=  
'Система.APM 1.Протоколы.Pulsar over TCP.Pulsar 4M 1.ArchiveHour.Ch1.Вход',  
Aggregate:= , Run:= Run);
```

ANY

Позволяет связывать входы и выходы функций и ФБ с параметрами имеющими произвольные типы данных. Если вход функции имеет тип ANY, то при установке связи входа с параметром типа BOOL значение функции будет рассчитываться исходя из реального значения входа, т.е. BOOL.

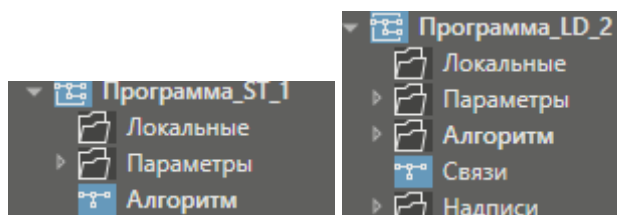
В некоторых случаях используются родовые типы данных, когда необходимо уточнение, какой диапазон типов данных можно использовать. Родовые типы данных идентифицируются использованием префикса ANY.

Так, например, если установлен тип данных ANY_NUM, то его можно связывать как с типами данных ANY_REAL, ANY_INT и с их дочерними типами. ANY_REAL является родовым для простых типов REAL и LREAL, а ANY_INT для LINT, DINT, INT, SINT, ULINT, UDINT, UINT, USINT

7.2. ОБЩИЕ СРЕДСТВА РЕДАКТИРОВАНИЯ ПРОГРАММ

7.2.1. СТРУКТУРА ПРОГРАММЫ В ДЕРЕВЕ

Программный компонент имеет predetermined структуру, зависящую от языка. Вид некоторых программ в полном дереве:



В полном дереве отображается вся внутренняя структура программы.

Набор корневых групп программного компонента зависит от языка.

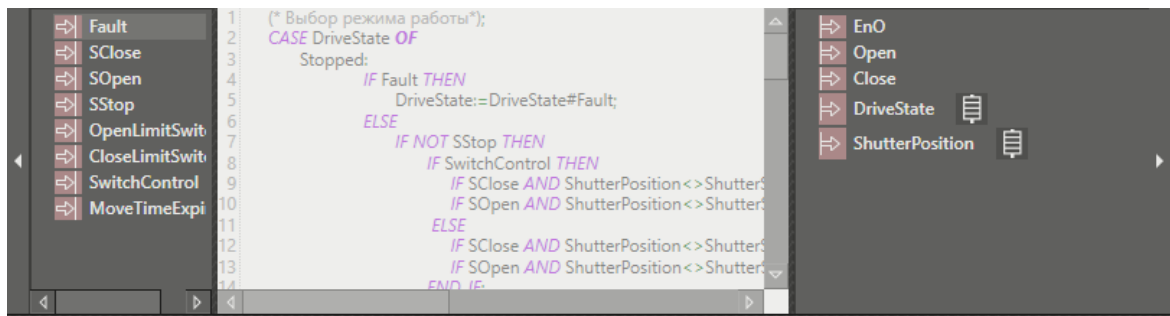
Группы программ:

Название	Языки	Назначение
Локальные	Все	Содержит локальные переменные программного компонента и экземпляры функциональных блоков. Данные переменные используются для внутренней работы программы
Параметры	Все	Содержит переменные программы, которые могут быть использованы для связи с внешними элементами.
Алгоритм	Все	Этот компонент представляет собой текст ST-программы или содержит структуру FBD-, LD- или SFC-алгоритма (без связей между элементами алгоритма)
Надписи	SFC,FBD, LD	Эта группа содержит пользовательские надписи.
Связи	FBD и LD	Содержит связи элементов
Действия	SFC	Содержит действия, как используемые в алгоритме, так и созданные предварительно (т.е. без назначения в какой-либо SFC-шаг). Предварительно созданные действия могут быть затем использованы (см. Инструмент описания SFC-действия). Действие может быть задано на любом языке.
Условия	SFC	Содержит условия, как используемые в алгоритме, так и созданные предварительно. Предварительно созданные условия могут быть затем использованы

Название	Языки	Назначение
		(см. Инструмент описания SFC-условия). Условие может быть задано на любом языке, кроме SFC.

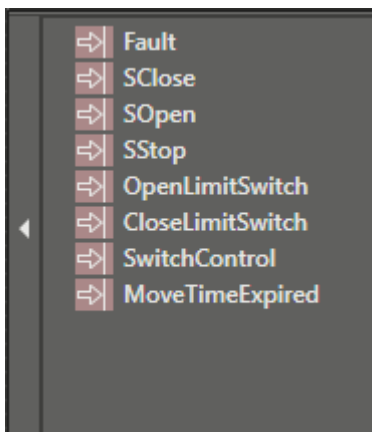
7.2.2. КЛЕММНИКИ

Клеммники – это панели программных редакторов, расположенные соответственно слева (клеммник входов) и справа (клеммник выходов) от алгоритма:



7.2.2.1. КЛЕММНИК ВХОДОВ

Клеммник входов содержит входные параметры программ и окон. Свойство Доступ параметров программ имеет значение Input. Клеммник располагается слева поля редактора (см. Клеммники):



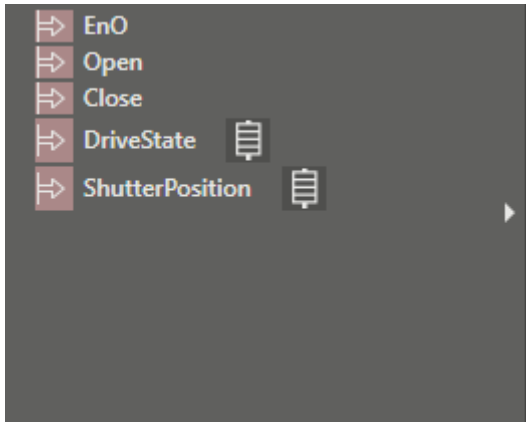
Видимость клеммника определяет кнопкой слева от списка параметров

В положении  отображается только сам переключатель, в положении  – весь клеммник.



Параметры созданные в клеммнике появятся в легенде редактора, во вкладке Параметры, а также в соответствующей группе в дереве.

7.2.2.2. КЛЕММНИК ВЫХОДОВ

Клеммник выходов содержит выходные параметры программ. Свойство Доступ параметров программ имеет значение Output. Клеммник располагается справа от поля редактора (см. Клеммники):



Видимость клеммника определяет кнопкой справа от списка параметров.

В положении  отображается только сам переключатель, в положении  — весь клеммник.

Параметры созданные в клеммнике появятся в легенде редактора, во вкладке Параметры, а также в соответствующей группе в дереве.

7.2.3. ЛЕГЕНДА

Легенда, представляющая собой набор вкладок, является вспомогательным средством редакторов программ.

Видимостью легенды управляет переключатель 'Легенда' (см. Панель управления видимостью окон).

Вне зависимости от языка, легенда содержит следующие вкладки:

- Легенда. Вкладка 'Локальные';
- Легенда. Вкладка 'Параметры';
- Легенда. Вкладка 'Заголовок ST';
- Легенда. Вкладка 'Ошибки';
- Легенда. Вкладка 'Скомпилированный код'.

Кроме того, легенда содержит вкладки, специфичные для того или иного языка (такие вкладки доступны, если в окне документов и редакторов активен соответствующий редактор).

7.2.3.1. ЛЕГЕНДА. ВКЛАДКА ЛОКАЛЬНЫЕ

Эта вкладка содержит локальные параметры программы, позволяет отредактировать их, а также позволяет создавать новые локальные параметры. Свойства локальных параметров, такие же как и у тех, что добавляются на клеммники программ, однако их нельзя связывать с внешними элементами проекта, они используются только внутри программ и не могут быть архивированы.

Во вкладке локальные создаются экземпляры функциональных блоков.

Название	Тип	Доступ	Сохранять	Начальное значение	Переключение
Переменная_1 = нет текста	LREAL	Local	Наследуется	0	
Переменная_2 = нет текста	ARRAY [1..5] OF REAL	Local	Наследуется	нет текста	

В структуре программы в дереве соответствует группе Локальные

7.2.3.2. ЛЕГЕНДА. ВКЛАДКА ПАРАМЕТРЫ

Эта вкладка отображает уже созданные параметры (в том числе и в панели клеммников), которые можно использовать для связи с другими элементами. Можно отредактировать уже существующие параметры, а также создать новые.

Название	Тип	Доступ	Сохранять	Начальное значение
EnO = True	BOOL	Output	Наследуется	True
Параметр_1 = нет значения	LREAL	Input	Наследуется	0

7.2.3.3. ЛЕГЕНДА. ВКЛАДКА ЗАГОЛОВОК ST


На этой вкладке отображается заголовок ST-кода, полученного в результате трансляции программы (заголовок содержит операторы определения переменных):

Палитра	Локальные	Параметры	Ошибки	Заголовок ST	Скомпилированный код	Элементы	Надписи	Связи
<pre> 1 VAR_INPUT 2 Параметр_1: LREAL := 0; 3 Параметр_2: LREAL := 0; 4 END_VAR 5 6 VAR_OUTPUT 7 EnO: BOOL := True; 8 Параметр_2: LREAL := 0; 9 END_VAR 10 11 </pre>								

На вкладке не отображаются переменные, объявленные непосредственно в тексте программы.

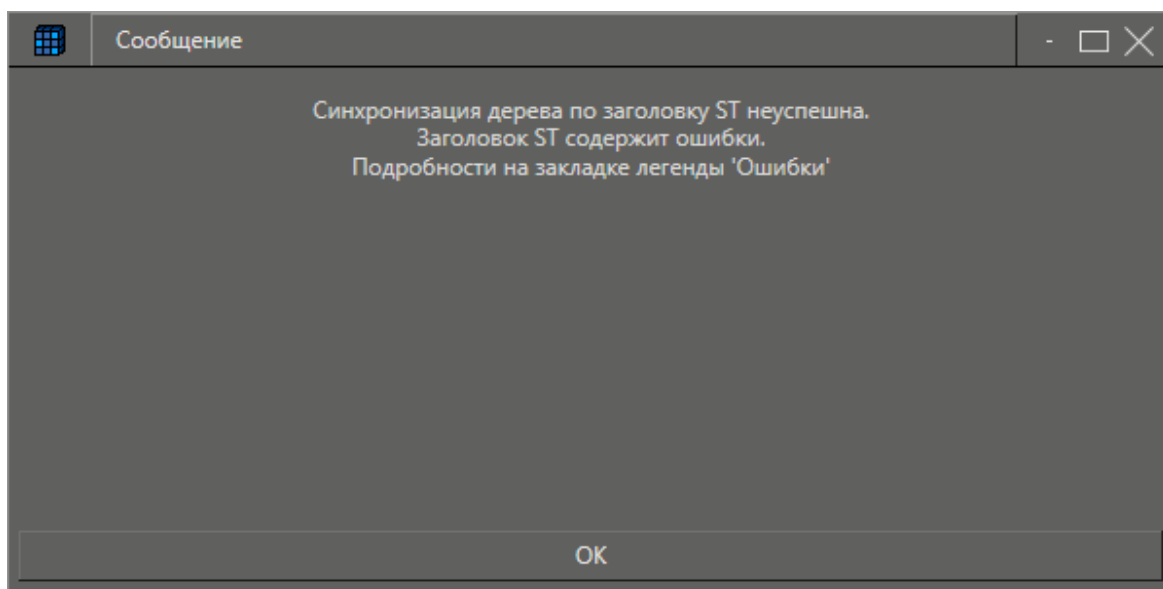
Вкладка представляет собой текстовый редактор, в котором, как и в редакторе ST, используется ST-подсказчик.



Вкладка имеет две кнопки для синхронизации: . Левая кнопка отображается всегда. Правая кнопка появляется, если разработчик проекта изменил текст заголовка вручную.

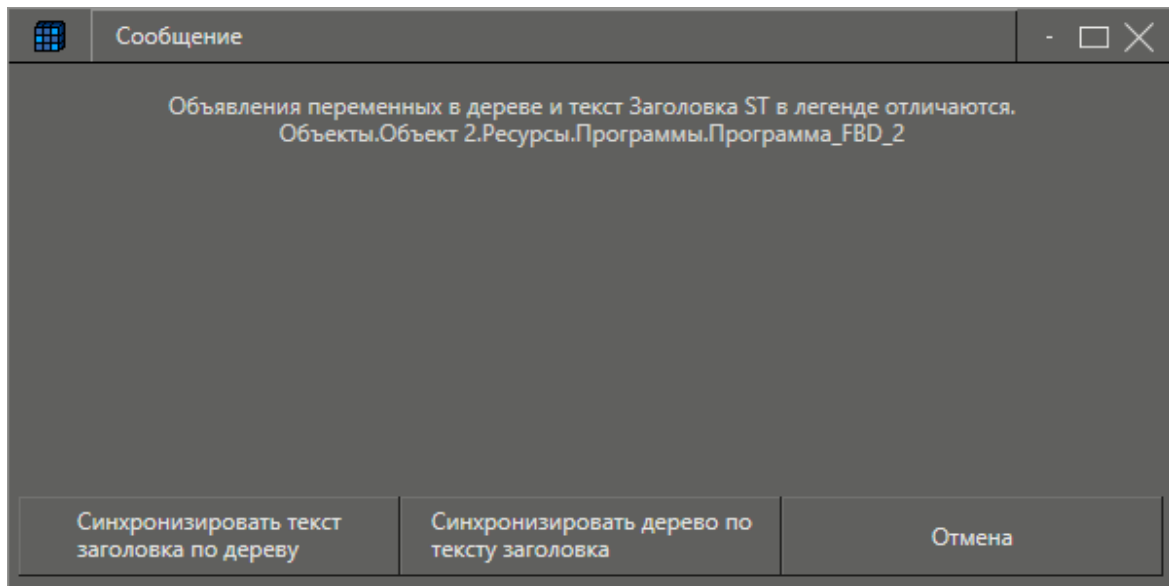
Левая - синхронизирует текст вкладки по дереву. Т.е. если во вкладке были сделаны изменения, то после нажатия этой кнопки они отменятся, и в текст добавятся переменные из дерева.

Правая синхронизирует дерево по тексту вкладки. Т.е. если во вкладке были сделаны изменения, добавлены параметры, то они будут добавлены в дерево, а несуществующие в заголовке параметры будут удалены из дерева. Если при редактировании заголовка были допущены ошибки, то при синхронизации появится сообщение:



Текст заголовка ST и дерево программы изменяться не будет, до тех пор пока ошибка не будет устранена и не будет проведена синхронизация.

Если разработчик проекта изменил текст вкладки Заголовок ST, но не произвел синхронизацию, то при попытке скомпилировать текст программы появится сообщение:



Синхронизация происходит в соответствии с выбором пользователя.

При выборе кнопки Отмена компиляция конфигурации проекта прекращается, выдается сообщение об ошибке "Прервано действие синхронизации заголовка.". Если компиляция проводилась перед запуском проекта в исполнительной системе, то запуск не произойдет.

7.2.3.4. ЛЕГЕНДА. ВКЛАДКА ОШИБКИ

На вкладке отображаются ошибки, которые были обнаружены при компиляции программы.



В отличие от таблицы в окне ошибок, строки таблицы на вкладке не имеют контекстного меню. Если в программах FBD, SFC и LD дважды кликнуть по строке с ошибкой, то откроется вкладка Скомпилированный код, курсор будет установлен в месте, соответствующему ошибке.

7.2.3.5. ЛЕГЕНДА. ВКЛАДКА СКОМПИЛИРОВАННЫЙ КОД

На этой вкладке отображается ST-код, полученный в результате компиляции программы.

```

1 PROGRAM Программа_ST_1_ID_80556
2 VAR_OUTPUT
3 ENO: BOOL := True;
4 Параметр_3: LREAL := 0;
5 END_VAR
6
7 VAR_INPUT
8 Параметр_1: LREAL := 0;
9 Параметр_2: LREAL := 0;
10 END_VAR
11
12
13
14 if Параметр_1=10 and Параметр_2>30 THEN Параметр_3:=Параметр_1 END_IF;
15
16 END_PROGRAM

```

В отличие от вкладки Легенда. Вкладка Заголовок ST, данная вкладка редактором не является.

7.2.3.6. ЛЕГЕНДА. ВКЛАДКА ЭЛЕМЕНТЫ

На этой вкладке отображаются элементы программ, написанных в редакторах LD, SFC, FBD.

Вид вкладки для программы LD:

Название	Тип	Декларация	Порядок выполнения	Координата X	Координата Y
Силовая линия 1				0	0
Катушка 1				0	0
Контакт 1				0	0

Вид вкладки для программ SFC:

Название	Тип	Декларация	Порядок выполнения	Координата X	Координата Y
Ветка 1				0	0
Аварийная ситуация				0	0
Ветвление условий И				0	0
Условие 1				0	0
Действия				0	0
Выключение				0	0
Прыжок 1				0	0
Основная ветка				0	0

Вид вкладки для программы FBD:

Название	Тип	Декларация	Порядок выполнения	Координата X	Координата Y
ADD	ADD	ADD	1	100	420
AND	AND	AND	5	660	380
Параметры				0	0
ENO				0	0
IN1				0	0
IN2				0	0
Q				0	0
GT	GT	GT	3	380	340
LT	LT	LT	2	340	180
RS_1	RS	RS_1	4	620	220
SUB	SUB	SUB	0	80	80

Выбор элемента во вкладке приведет к его выделению на схеме. Наиболее полно столбцы заполняются для программ FBD. Двойной клик по заголовку столбца приведет к сортировке элементов по данному свойству.

Название	Назначение
Название	Название элемента
Тип	Имя типа библиотечного элемента
Декларация	Имя экземпляра функционального блока
Порядок выполнения	Номер выполнения элемента в цикле программы
Координата X	Положение элемента на схеме по оси X
Координата Y	Положение элемента на схеме по оси Y

7.2.3.7. ЛЕГЕНДА. ВКЛАДКА СВЯЗИ

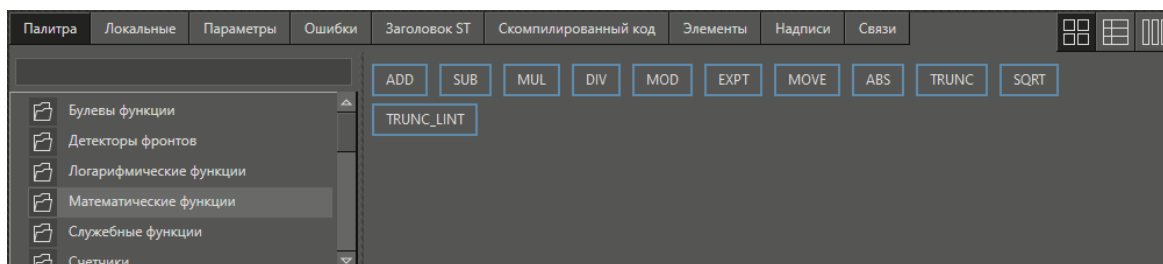
На этой вкладке отображаются связи программ, написанных в редакторах LD, SFC, FBD.

Название	Источник	Приемник	Сохранять	Разъединить
Программа_FBD_1.Параметр_1-RS.RS.S	Параметр_1	RS.S	нет текста	
Программа_FBD_1.Параметр_2-RS.RS.En	Параметр_2	RS.En	нет текста	

7.2.3.8. ЛЕГЕНДА. ВКЛАДКА ПАЛИТРА

Вкладка Палитра легенды содержит элементы, характерные для того или иного редактора. Элементы находятся в категориях - отображаются в левой части палитры. Элементы выбранной категории будут располагаться справа.

Из палитры элементы можно перетащить на схемы редакторов, а также в текст языка ST.



При редактировании программного компонента на языке ST, LD или FBD на вкладке отображаются программные компоненты из подключенных библиотек (в том числе компоненты, созданные пользователем в корневых группах Функции и ФБ пользовательских библиотек).

Способ отображения элементов в палитре

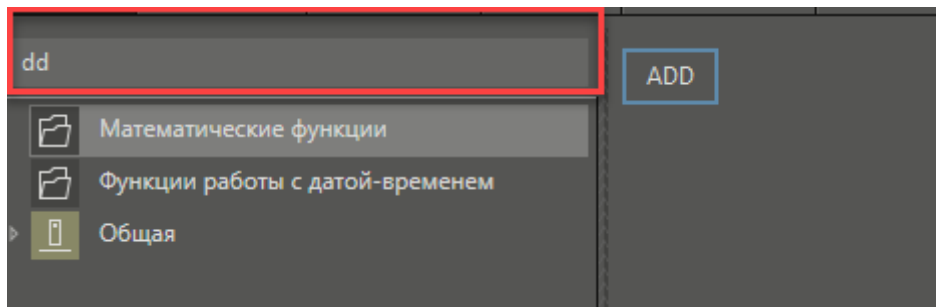
В палитре можно выбрать способ отображения элементов в палитре при помощи кнопок



- Вид списком - элементы располагаются друг за другом слева направо, сверху вниз. При наведении на имя элемента отображается путь к нему в библиотеке и комментарий.
- Табличный вид - элементы располагаются в таблице, где в первом столбце отображается имя, а во втором комментарий. При наведении на имя элемента отображается путь к нему в библиотеке и комментарий
- Вид колонок - элементы располагаются друг за другом сверху вниз слева направо. При наведении на имя элемента отображается путь к нему в библиотеке и комментарий

Фильтрация элементов в палитре

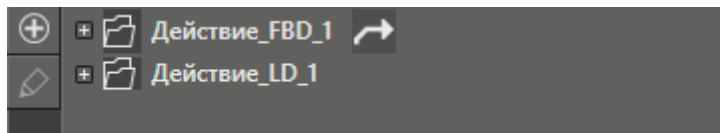
В палитре можно отфильтровать элементы по имени. Если известно имя элемента или часть имени, то можно ввести символы в строке:



В этом случае останутся только те категории и элементы, в которых содержится данная последовательность символов.

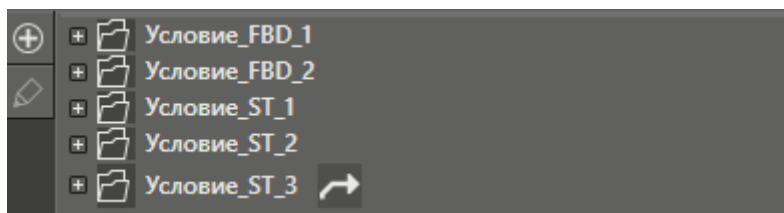
7.2.3.9. ЛЕГЕНДА. ВКЛАДКА ДЕЙСТВИЯ

На этой вкладке отображаются уже созданные действия в редакторе SFC, их можно отредактировать, а также создать новые.



7.2.3.10. ЛЕГЕНДА. ВКЛАДКА УСЛОВИЯ

На этой вкладке отображаются уже созданные условия в редакторе SFC, их можно отредактировать, а также создать новые.



7.2.3.11. ЛЕГЕНДА. ВКЛАДКА НАДПИСИ

Эта вкладка отображает надписи уже созданные разработчиком, можно отредактировать уже созданные и создать новые надписи.

	Название	Надпись	Координата X	Координата Y
	Надпись 1	Запуск	0	0
	Надпись 2	Оставновка	100	800
	Надпись 3	Выбор условий	60	560

7.2.4. РЕДАКТИРОВАНИЕ ГРАФИЧЕСКИХ ЯЗЫКОВ

Редактирование диаграмм в дереве

Масштабирование диаграмм

Прокрутка диаграмм

Окно Миникарта

Связывание переменных на FBD- и LD-диаграммах

Операции с надписями

7.2.4.1. РЕДАКТИРОВАНИЕ ДИАГРАММ В ДЕРЕВЕ

Поскольку в структуре любого программного компонента существует группа Алгоритм (см. Структура программного компонента), содержащая все элементы алгоритма, диаграммы FBD, LD и SFC могут редактироваться с помощью типовых операций в дереве.

Примеры (см. также Операции с переменными):

- перетаскивание функции/программы на группу Алгоритм – размещение программного элемента в виде FBD-блока на диаграмме. При таком размещении левый верхний угол FBD-блока располагается в точке (0,0) диаграммы (в верхнем левом углу);
- перетаскивание элемента LD-алгоритма из библиотеки на группу Алгоритм или на группу Алгоритм. Силовая линия программного компонента LD – размещение элемента на диаграмме;

- перетаскивание SFC-действия из корневой группы Действия программного компонента SFC на группу Алгоритм.Шаг<n>.Действия или на компонент Алгоритм.Шаг<n>.Действия.<действие> – соответственно добавление или замена действия в шаге Шаг<n>;
- выделение элемента алгоритма в группе Алгоритм или ее подгруппе и нажатие Del – удаление элемента.

7.2.4.2. МАСШТАБИРОВАНИЕ ДИАГРАММ

Помимо способов масштабирования, общих для всех редакторов (см. Масштабирование в редакторе проекта), для масштабирования мнемосхем, FBD-, LD- и SFC-диаграмм предусмотрены следующие дополнительные средства:

- масштабирование рамкой (см. 'Масштабирование рамкой');
- Миникарта;
- команда Показать все

'Показать все'

- Показать все (CTRL+0) – устанавливает такой масштаб, при котором видны все элементы диаграммы (в случае мнемосхемы – вся мнемосхема).

7.2.4.3. ПРОКРУТКА ДИАГРАММ

Помимо средств прокрутки, которые используются во всех окнах редактора проекта (см. Прокрутка), при редактировании мнемосхем, FBD-, LD- и SFC-диаграмм могут использоваться дополнительные средства прокрутки:

- перетаскивание в миникарте зоны видимости (см. Окно 'Миникарта') – плавное перемещение диаграммы в направлении, противоположном перемещению зоны видимости;
- однократное нажатие кнопок клавиатуры → / ← / ↓ / ↑ – дискретное перемещение с мелким фиксированным шагом в направлении, противоположном стрелке.

7.2.4.4. МИНИКАРТА

Миникарта находится в нижнем правом углу MasterSCADA 4D, под панелью свойств.



В миникарте FBD-, LD-, SFC-диаграмма или мнемосхема (окно) всегда отображается полностью.

Часть диаграммы, видимая в редакторе, обозначается в миникарте черным прямоугольником (далее этот прямоугольник называется зоной видимости):

Для изменения размеров зоны видимости (и, соответственно, масштаба части диаграммы, видимой в редакторе) нужно установить курсор в окно Миникарта и вращать колесо мыши.

Перетаскивание зоны видимости в миникарте обеспечивает прокрутку диаграммы (см. Прокрутка диаграмм).

Масштаб отображения в самой миникарте можно изменять, изменяя размеры ее окна (см. Изменение размеров).

7.2.4.5. СВЯЗЫВАНИЕ ПЕРЕМЕННЫХ НА FBD- И LD-ДИАГРАММАХ

Для связывания переменных на FBD- и LD-диаграммах используется общая операция связывания переменных (см. Связывание переменных)

Задание связей переменных на FBD- и LD-диаграмме возможно только в том случае, если включен режим связывания (см. 'Связывание').

Если перетащить параметр из дерева проекта на ножку блока FBD, то в клеммнике появится параметр программы. Тип нового входного/выходного параметра программы будет соответствовать типу параметра дерева проекта, за исключением случая, когда связь устанавливается со входом FBD, который имеет тип REF_TO - тогда и параметр программы будет иметь тип REF_TO.

Допустимые связи:

- вход программного компонента – вход FBD-блока;
- выход FBD-блока – выход программного компонента;
- выход FBD-блока – вход того же самого или другого FBD-блока.

У входа FBD-блока и выхода программного компонента может быть только одна связь.

У выхода FBD-блока может быть множество межблочных связей, но только одна связь с выходом программного компонента.

7.2.5. КОМПИЛЯЦИЯ ПРОГРАММ

Независимо от того, на каком языке была написана программа разработчиком проекта при подготовке проекта к запуску программа будет проверена внутренними средствами MasterSCADA 4D и транслирована в язык ST. Просмотреть результат компиляции можно на вкладке Скомпилированный код.

Компиляция программы осуществляется в двух случаях:

1. Если была нажата кнопка в панели инструментов редактора программ



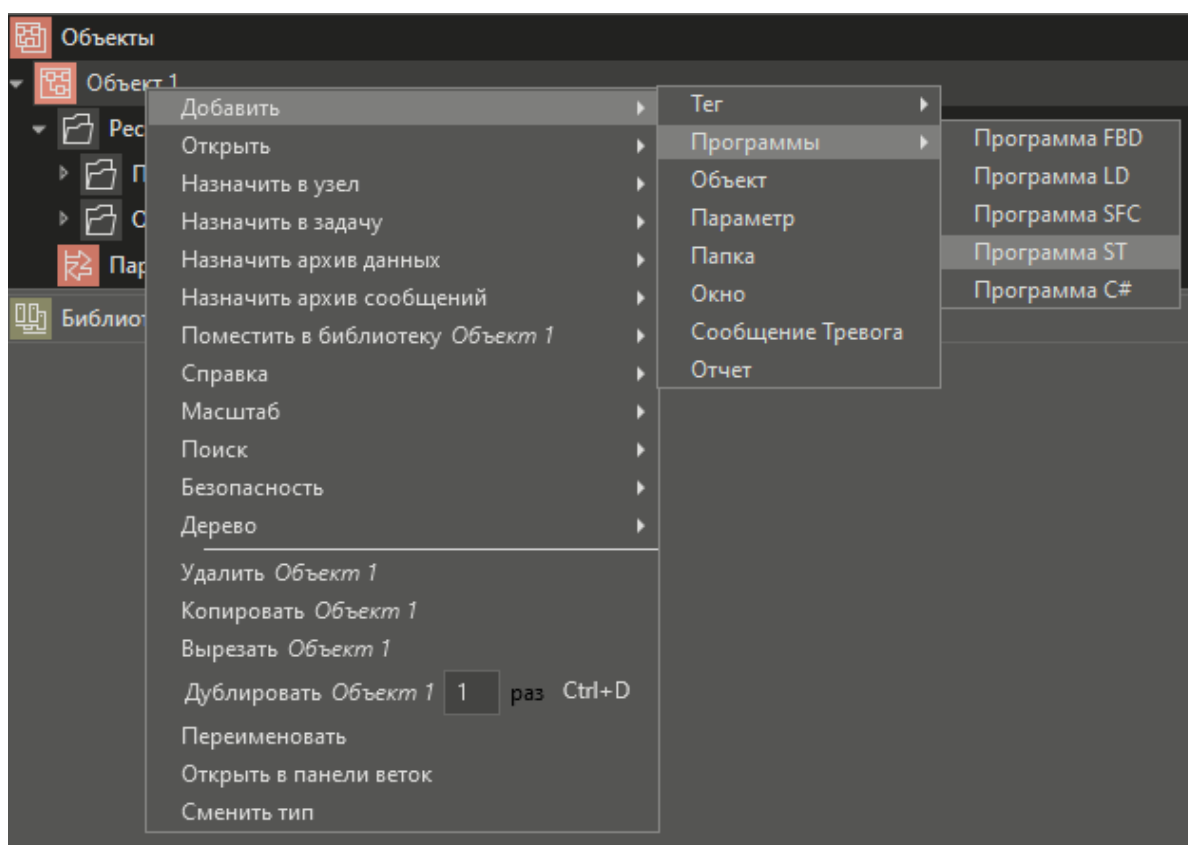
2. При подготовке проекта к запуску в исполнительной системе.

Если в результате компиляции произошли ошибки, то с ними можно ознакомиться на вкладке Легенды Ошибки

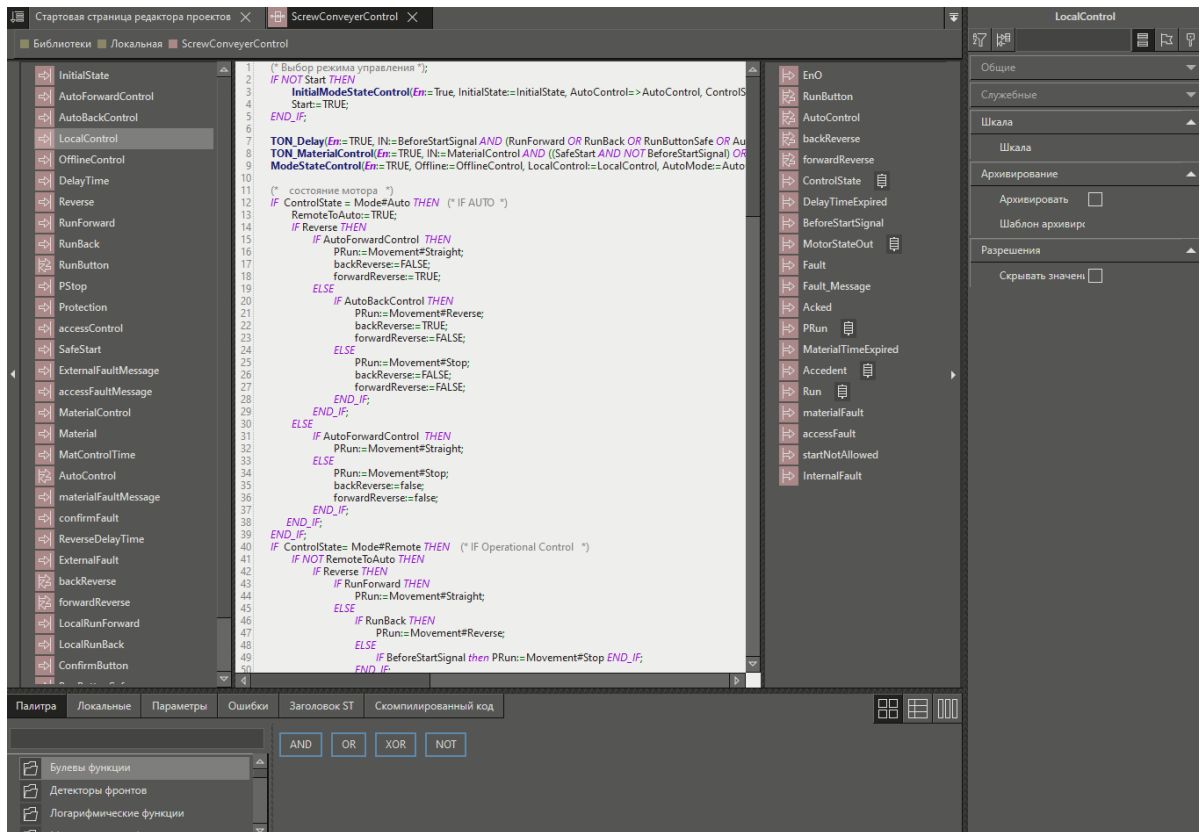
7.3. РЕДАКТОР ST

Master ST – текстовый язык общего назначения (см. Синтаксис Master ST , а также все подразделы разделов Общие элементы программ, Общие средства редактирования программ и Язык Master ST).

Редактор откроется автоматически при добавлении программы ST в дерево объектов или в дерево системы, либо при добавлении ФБ ST в дерево библиотек.

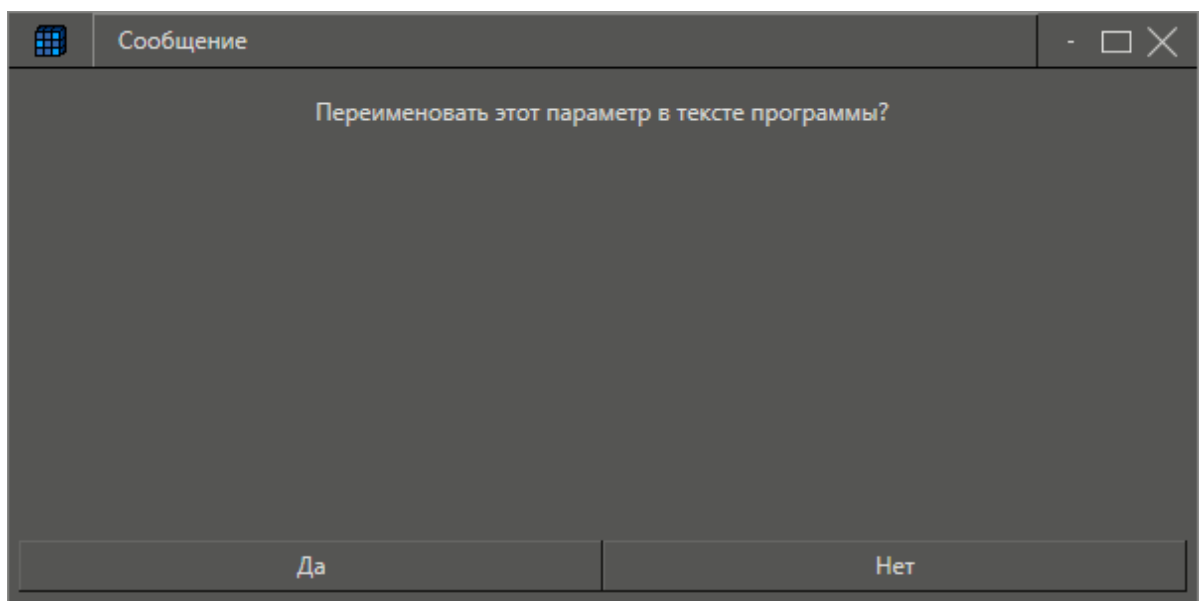


Вид редактора:



Редактор ST представляет собой текстовый редактор, для масштабирования в редакторе используются стандартные операции (см. Масштабирование в редакторе проекта).

В редакторе имеется клеммник входов и клеммник выходов. При переименовании параметров в клеммнике появится запрос если старое имя встречалось в тексте, нужно ли переименовать в тексте программы.



Нажатие на кнопку Да приведет к изменению текста программы: старое имя параметра заменится на новое.

Ключевые слова и имена Ф/ФБ выделяются различным шрифтом и цветом, строки кода нумеруются:

```
1 out1:=0;
2 IF(in0&1)&NOT (out1&16#1)THEN out1:=out1+16#1;END_IF;
3 IF(in1&1)&NOT (out1&16#2)THEN out1:=out1+16#2;END_IF;
4 IF(in2&1)&NOT (out1&16#4)THEN out1:=out1+16#4;END_IF;
5 IF(in3&1)&NOT (out1&16#8)THEN out1:=out1+16#8;END_IF;
6 IF(in4&1)&NOT (out1&16#10)THEN out1:=out1+16#10;END_IF;
7 IF(in5&1)&NOT (out1&16#20)THEN out1:=out1+16#20;END_IF;
8 IF(in6&1)&NOT (out1&16#40)THEN out1:=out1+16#40;END_IF;
9 IF(in7&1)&NOT (out1&16#80)THEN out1:=out1+16#80;END_IF;
10 out2:=ADD(BOOL_TO_USINT(in1),BOOL_TO_USINT(in2));
```

Для вставки в текст ST-программы переменной или программного компонента можно выполнить перетаскивание этой переменной или компонента из любого окна/редактора, в котором эта переменная или компонент отображается (из окна структуры, легенды, клеммника и т.п.), а также использовать ST-подсказчик .

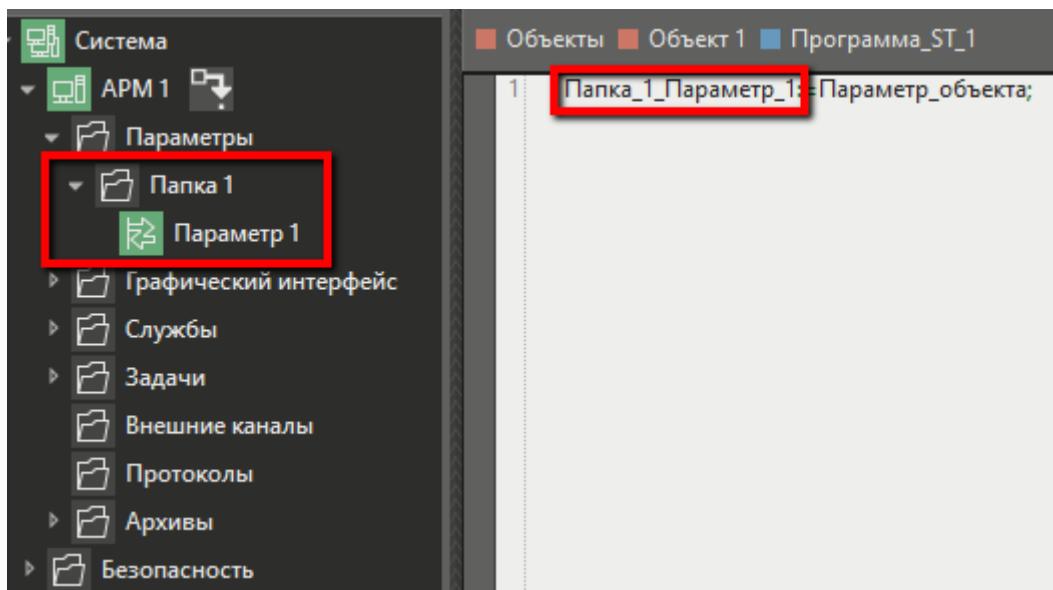
В редакторе ST задается тело программного компонента (код без конструкции PROGRAM...END_PROGRAM и заголовка, содержащего операторы определения переменных).

Операторы определения переменных могут использоваться в редакторе ST, однако для переменных, объявленных таким образом, нельзя задать связь.

На языке ST может быть разработан любой программный компонент – программа , функция и функциональный блок.

7.3.1. ГЛОБАЛЬНЫЕ ПЕРЕМЕННЫЕ

Если параметры созданы в дереве системы, то они являются глобальными. Такие параметры можно использовать в программе ST без дополнительного определения, достаточно написать имя переменной в тексте программы. Если параметр находится в папке узла, то тогда его имя будет ИмяПапки_ИмяПараметра:



Важно! При использовании глобальных параметров (переменных) в программах дерева объектов или библиотек, проверка программы будет неуспешна. В этом случае ошибки типа: Неизвестная переменная [имя глобальной переменной] можно игнорировать. При проверке всего проекта ошибки не будет.

7.3.2. ВЫЗОВ ФУНКЦИЙ И ФУНКЦИОНАЛЬНЫХ БЛОКОВ

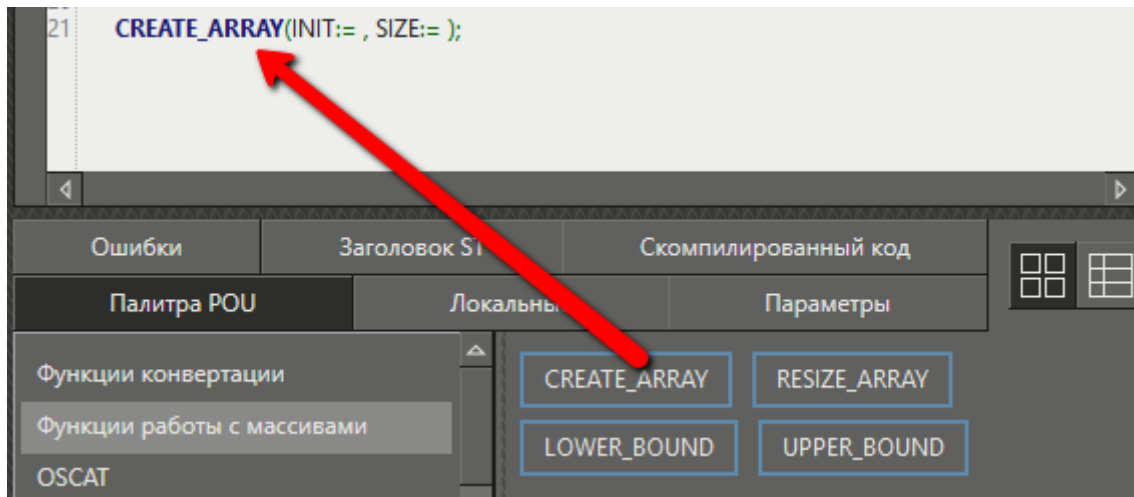
Вызов функций

Функция (термин стандарта IEC 61131-3) – это программный компонент (POU), который возвращает один элемент данных (в т.ч. массив или структуру) как результат своего выполнения (т.н. возвращаемое значение) и имеет произвольное количество входных элементов (VAR_INPUT и VAR_IN_OUT). Функция не содержит внутренних переменных, значения которых сохраняются между вызовами, поэтому при одних и тех же значениях входов всегда производит одни и те же выходные значения.

В программы добавляется ссылка на тип в библиотеке. Как правило, для вызова функции используют следующую форму записи: [имя функции]([вход1] := [выражение1], [вход2] := [выражение2],...)

Вход EN в языке ST не используется. Считается, что он равен TRUE. В вызове функции может быть задано произвольное число входных аргументов, в т.ч. ни одного.

Если перетащить функцию из библиотеки в тело программы, то в форме записи будут присутствовать все входы:



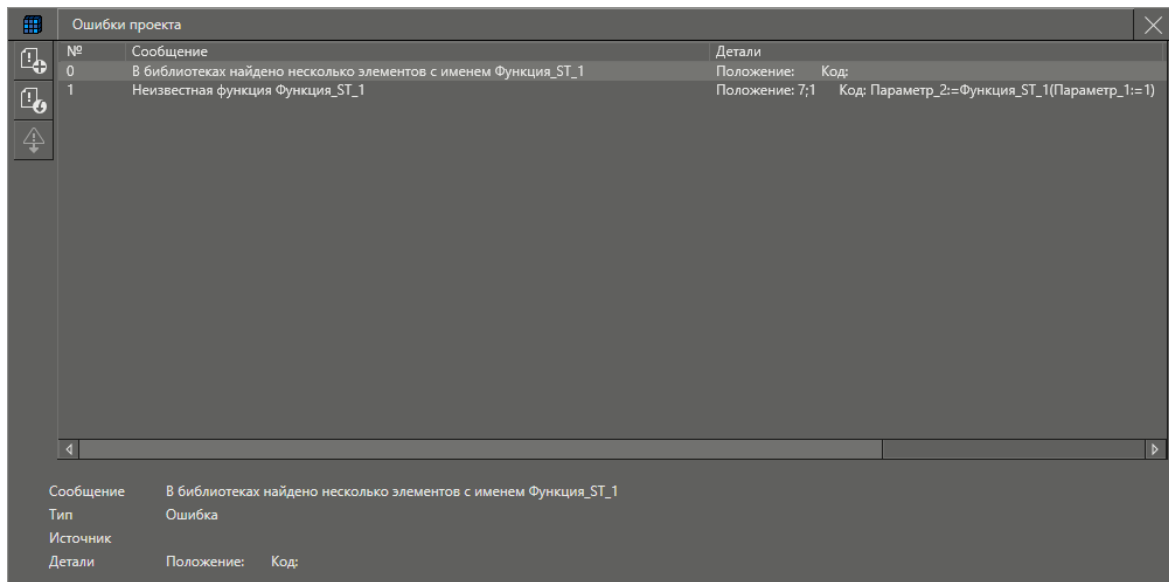
Если вход не определить, то будет считаться, что он равен значению по умолчанию.

Функция возвращает значение того типа данных, которое задано для ее выхода Q.

Если функция кроме выхода Q имеет другие выходы, то для присвоения такого выхода функции какому-либо параметру необходимо использовать следующую форму записи:

[имя функции]([вход1] := [выражение1], [вход2] := [выражение2], ...[выход функции] =>[Выход программы]...)

В случае, если разные библиотеки проекта содержат функции с одинаковыми именами, то при попытке компиляции появится сообщение о том, что найдено несколько элементов с данным именем.



Если нет возможности задать у функций разные имена, то необходимо перед именем функции указать имя библиотеки, в которой она находится:

[имя библиотеки].[имя функции]([вход1] := [выражение1], [вход2] := [выражение2], ...[выход функции] =>[Выход программы]...)

Примеры:

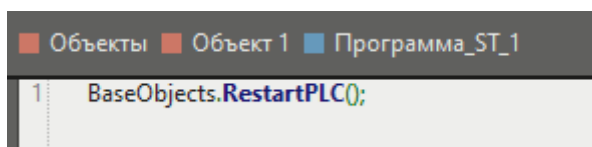
mINT:=TRUNC(IN:=3.14) - выходу программы mINT присваивается значение полученное в результате расчета функции TRUNC, при условии, что ее вход равен IN равен 3.14

mINT1:= OSCAT.D_TRUNC(X:=2*mREAL) - в данном случае вход функции X рассчитывается путем умножения на 2 значения входа программы mREAL, после вычисления значения функции оно присваивается выходу программы mINT1;

mREAL2:=myADD(x:=(3+mREAL), y:=3) - вход x пользовательской функции myADD рассчитывается исходя из значения параметра программы mREAL, значение входа y не меняется, после вычисления значения функции оно присваивается параметру программы mREAL2;

Q:= SPLIT_DATE(IN:=Дата_время, YEAR =>Параметр_1) - происходит вызов функции SPLIT_DATE вход IN равен входу программы Дата_время, после вычисления значения функции выход функции YEAR присваивается выходу программы Параметр_1

Если результат функции не нужно записывать в переменную, то функцию можно вызвать в виде func(). Например, если программа, содержит функцию RestartPLC:



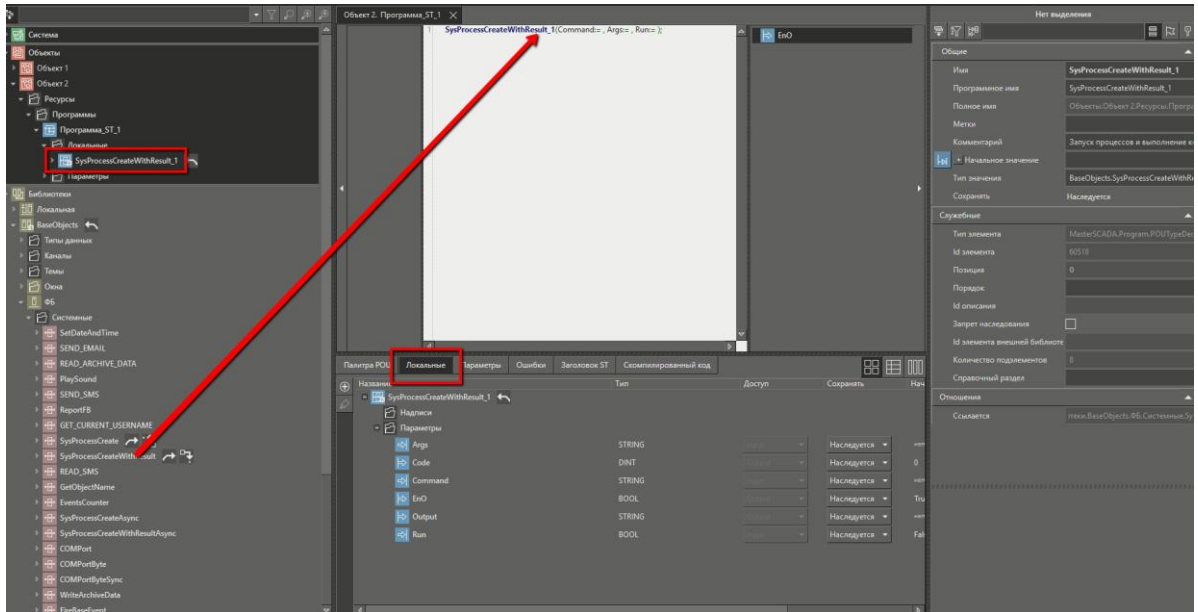
```
1 BaseObjects.RestartPLC();
```

Важно! При использовании функции без записи результата в переменную, проверка программы будет неуспешна. В этом случае ошибки типа: Ожидается тело функции, можно игнорировать. При проверке всего проекта ошибки не будет.

Вызов ФБ


В отличие от функций в программах ST вызываются не типы, а экземпляры.

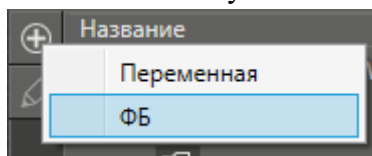
Для того чтобы создать экземпляр ФБ в программе нужно перетащить тип ФБ из Палитры в поле программы. Также можно перетащить нужный функциональный блок из дерева библиотек:



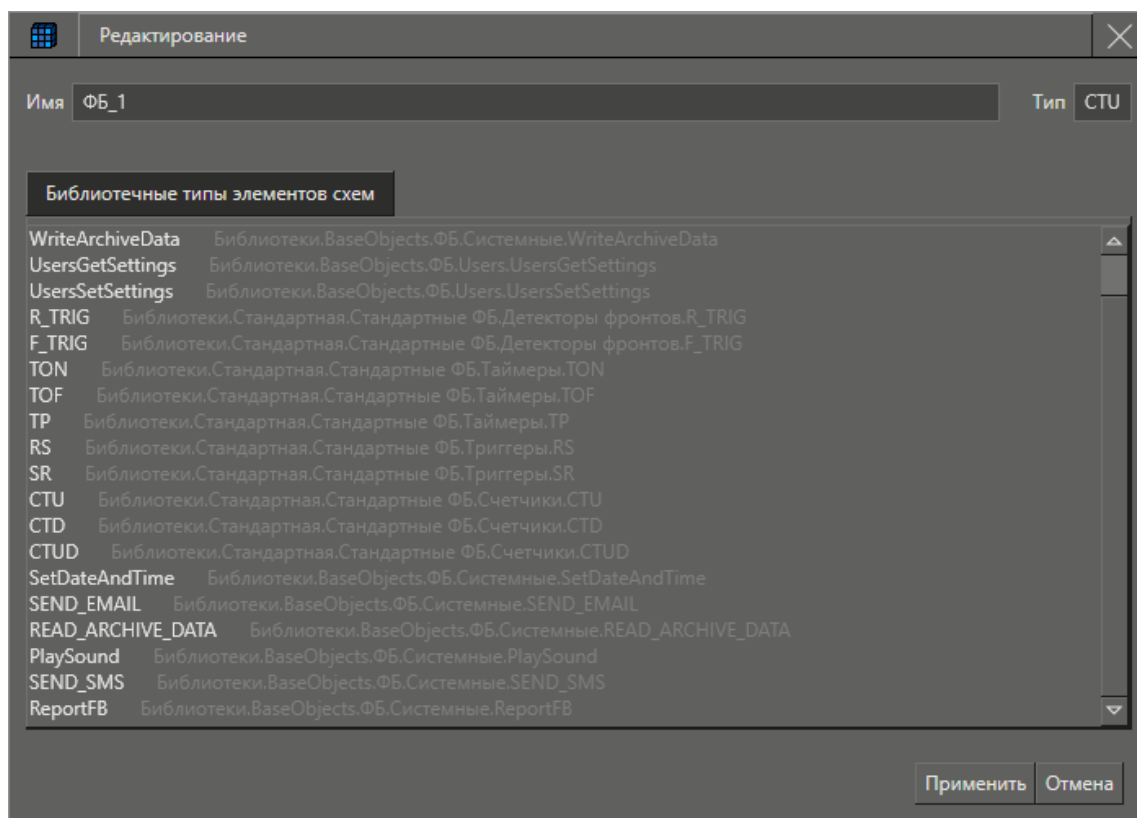
Функциональный блок добавится в текст программы со всеми входами. Затем эти входы можно задать, например, указать константу, либо перетащить в нужное место локальную переменную программы или параметр программы из клеммника входа. При этом видно, что в группе Локальные в дереве и в одноименной панели палитры появился экземпляр данного ФБ.

Можно сначала добавить экземпляр во вкладку легенды Локальные. Это можно сделать двумя способами:

- перетащить его из дерева библиотек в поле вкладки.
- нажать кнопку . В появившемся меню выбрать пункт ФБ:



. Появится элемент ФБ 1, если дважды нажать по нему левой кнопкой мыши, то откроется диалоговое окно, где нужно выбрать имя типа, и указать имя его экземпляра.:



У появившегося экземпляра можно установить значения по умолчанию для его входов, связи параметров проекта с параметрами экземпляра ФБ.

После чего можно обращаться к этому ФБ в теле программы ST.

Форма записи для обращения к параметру ФБ: [имя экземпляра ФБ].[имя параметра]

Примеры использования ФБ

Помним, что экземпляр ФБ уже находится во вкладке Локальная.

`gen_rdm_1.Am:=1` - в данном случае входу Am экземпляра ФБ `gen_rdm` присваивается значение 1.

`VXX:=1.1+1.65*gen_rdm_1.out` - тут переменной VXX присваивается значение выражения, где `gen_rdm_1.out` это выход out экземпляра функционального блока `gen_rdm`

`gen_rdm_1(PT:=t#10s, OS:=0,Out=>Параметр_1)` - производится вызов экземпляра функционального блока, значения входов PT и OS производятся непосредственно в форме вызова, значение выхода Out присваивается выходу программы `Параметр_1`.

7.3.3. ВЫЗОВ ПРОГРАММ ОБЪЕКТОВ

В одной программе ST можно вызывать другие программ объектов и узлов.

Если вызываемая программа находится в том же объекте, то в тексте достаточно обратиться к ней по имени, например, Программа_FBD_1(V_1:=V_1_ST , V_2:=V_2_ST,OUT=>Итог_1);

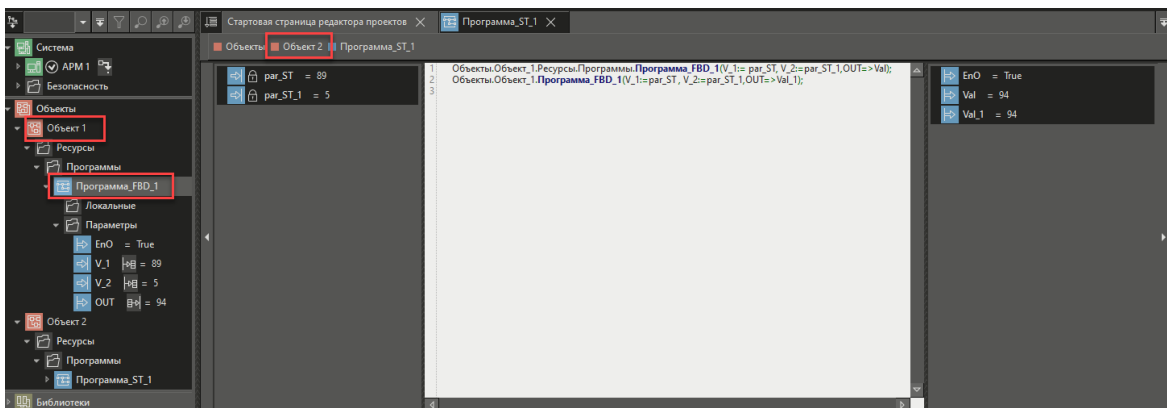
В этом случае на входы программы будут переданы значения параметров V_1_ST, V_2_ST, а результат выполнения будет передан на выход программы ST Итог_1:



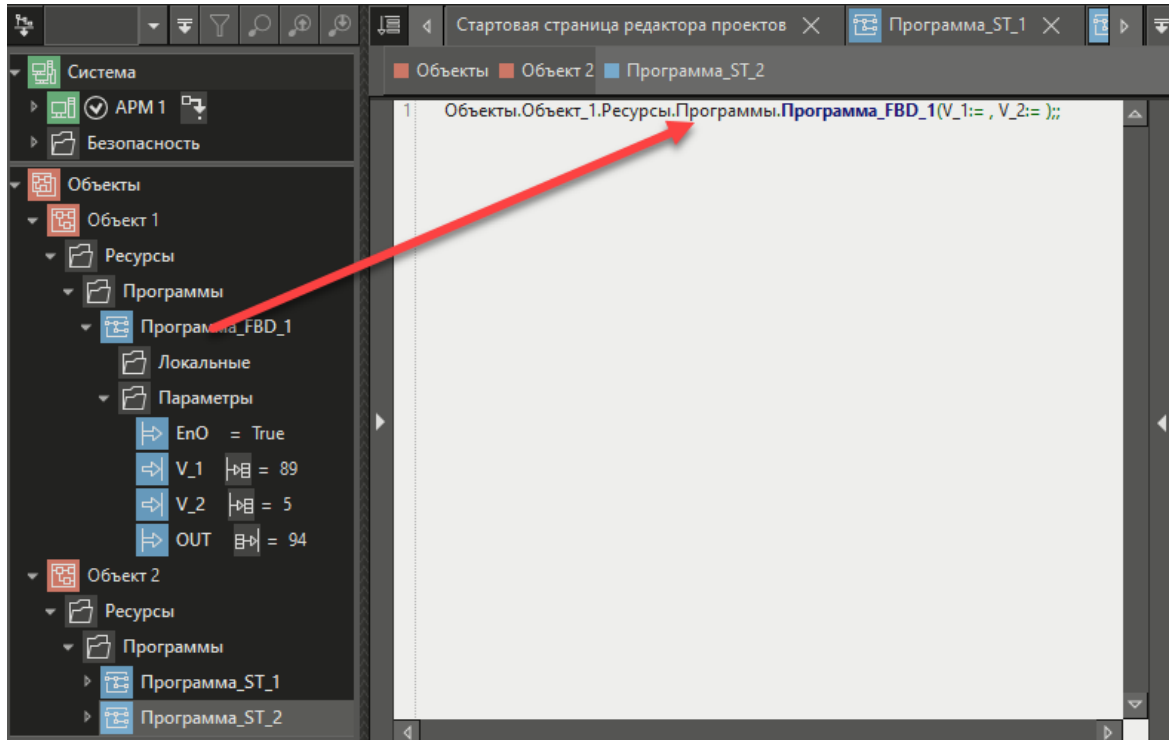
Если программа находится в другом объекте или узле, то обратиться к ней можно по полному имени, или по имени, в котром отсутствуют вспомогательные слова. Например:

Объекты.Объект_1.Ресурсы.Программы.Программа_FBD_1(V_1:= par_ST, V_2:=par_ST_1,OUT=>Val);

Объекты.Объект_1.Программа_FBD_1(V_1:=par_ST , V_2:=par_ST_1,OUT=>Val_1);



Если перетащить программу из дерева в программу, то она добавится в программу со всеми входами:



Вызываемая программа выполнится в рамках текста программы. Если вход не задан, то программа будет использовать последнее переданное в него значение - в другом вызове или по связи в случае периодического выполнения.

7.3.4. ФУНКЦИИ ПРЕОБРАЗОВАНИЯ ТИПА

К таким функциям относятся:

- функции `<TYPE1>_TO_<TYPE2>` преобразования типа TYPE1 в тип TYPE2 – например, `bool_to_real`, `sint_to_lreal`, `int_to_bool` и т.п..

Функции `<TYPE1>_TO_BOOL(myVar)` возвращают TRUE при любом ненулевом, в т.ч. отрицательном, значении myVar типа данных TYPE1.

При преобразовании REAL/LREAL в целочисленный тип данных значение округляется. Чтобы отбросить дробную часть, нужно использовать функцию TRUNC, целая часть к нулю;

- функции `*_BCD_TO_*` и `**_TO_BCD_*` выполняют преобразования между переменными типов BYTE, WORD, DWORD и LWORD и переменными типов USINT, UINT, UDINT и ULINT (обозначены соответственно знаками "*" и "**"), когда соответствующие битовые строки содержат данные в формате BCD.

Примеры:

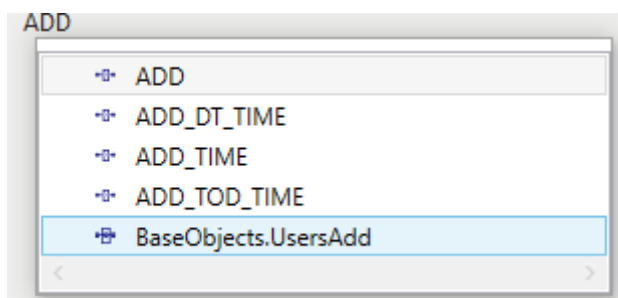
```
USINT_TO_BCD_BYTE(25) = 2#0010_0101
```

```
WORD_BCD_TO_UINT (2#0011_0110_1001) = 369
```

7.3.5. ST-ПОДСКАЗЧИК

Помимо стандартного способа редактирования кода ST (с помощью мыши и клавиатуры), предусмотрен ввод с помощью ST-подсказчика (IntelliSense, Интеллисенс)– при вводе символов (с помощью клавиатуры) открывается окно, содержащее список доступных переменных и функций.

Чтобы открыть подсказчик вручную, надо нажать сочетание клавиш CTRL+SPACE.



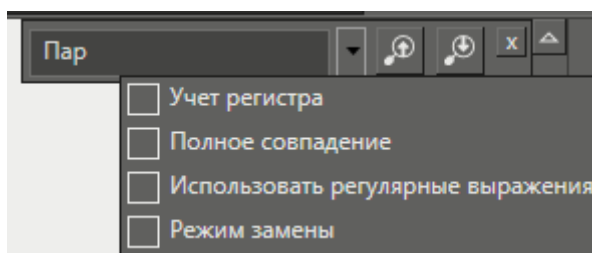
Чтобы из подсказчика вставить в ST-код переменную/функцию, нужно выбрать ее в списке при помощи мыши или клавиш курсора и кнопки Enter.

Для управления подсказчиком используется редактор общих настроек (вкладка Разное , параметры Интеллисенс и Количество символов интеллисенс).

Чтобы скрыть окно подсказчика, нужно щелкнуть кнопку ESCAPE. По этой команде окно подсказчика скрывается до возникновения условий его нового появления.

7.3.6. ПОИСК И ЗАМЕНА В ТЕКСТЕ ST

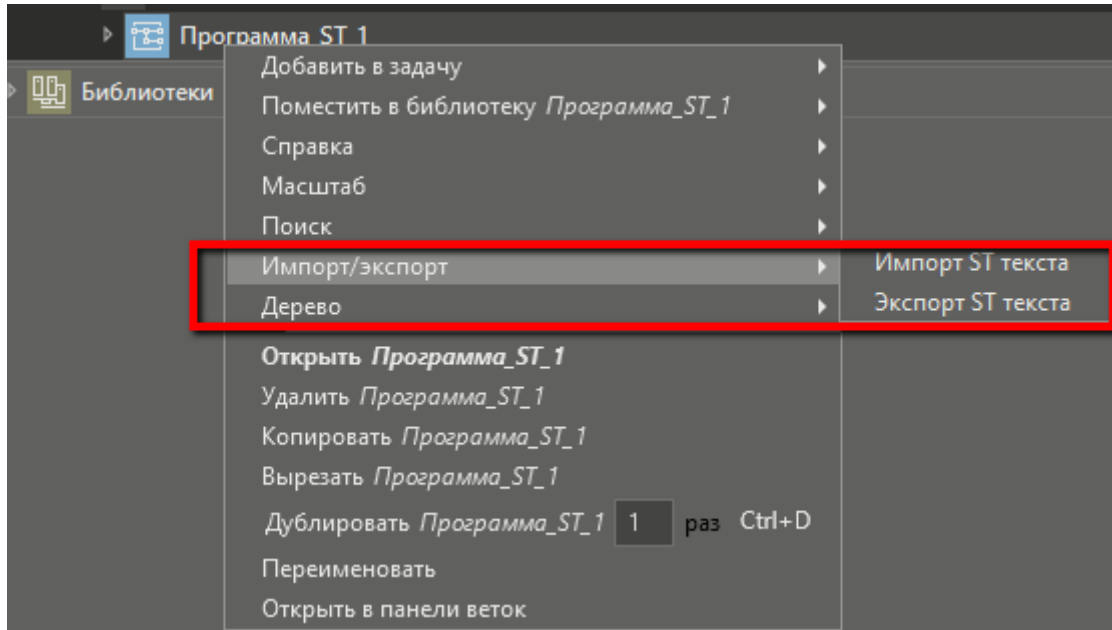
Чтобы открыть диалог поиска необходимо нажать комбинацию клавиш CTRL+F. Диалог появится в верхнем левом углу.



7.3.7. ЭКСПОРТ И ИМПОРТ ПРОГРАММ ST

Допустим импорт и экспорт программ, функциональных блоков и функций, которые созданы на языке Master ST.

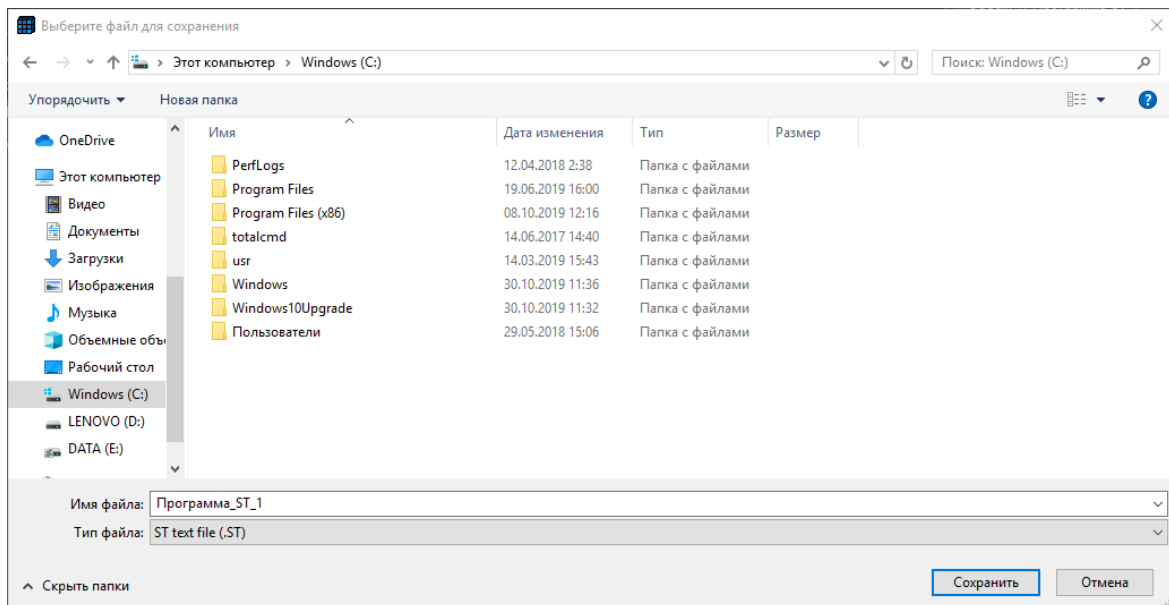
В контекстном меню этих элементов есть соответствующие пункты:



Экспорт ST текста

Важно! Рекомендуется проводить экспорт программы, в работоспособности которой убедились в режиме исполнения, в противном случае, импортировать неработающую программу будет невозможно

В результате выполнения данного пункта открывается диалоговое окно операционной системы, в котором необходимо выбрать место сохранения и имя файла:



После нажатия на кнопку Сохранить в указанном месте создастся текстовый файл с расширением ST.

Импорт ST текста

В результате выполнения данного пункта открывается диалоговое окно операционной системы, в котором необходимо выбрать файл, содержащий текст ST программы. Файл должен иметь расширение ST.

Согласно считанной информации из файла формируется текст программы в редакторе, заполняется вкладка легенды Заголовков, на ее основании формируется дерево программы (добавляются необходимые параметры и экземпляры ФБ).

7.3.8. СИНТАКСИС MASTER ST

Программа на языке Master ST – это совокупность последовательно исполняемых ST-предложений.

ST-предложение – это последовательность лексем, определяющая выполнение логически законченного промежуточного действия. Таким действием может быть присвоение переменной результата вычислений, вызов ФБ и т.п. Операторы Master ST (кроме инфиксных) также образуют предложения.

Каждое ST-предложение должно завершаться точкой с запятой.

Длина строки программы не ограничивается, лексемы разделяются произвольным числом пробелов, знаков табуляции или символов перевода строки.

7.3.8.1. ОПЕРАТОРЫ MASTER ST

Enter topic text here.

7.3.8.1.1. ОПЕРАТОР RETURN

Синтаксис:

```
return
```

Алгоритм: выход из Ф, ФБ или программы (например, как результат выполнения оператора IF).

Пример

```
if x>100 then
```

```
y:=100;
```

```
return;
```

```
else
```

```
y:=x;
```

```
end_if;
```

7.3.8.1.2. ОПЕРАТОР IF

Данный оператор начинается с ключевого слова if и заканчивается ключевым словом end_if. Существует несколько вариантов задания данного оператора.

```
if {выражение} then {последовательность предложений} end_if
```

Алгоритм: если {выражение} истинно, выполняется {последовательность предложений}, иначе никаких действий не производится.

```
if {выражение} then {последовательность предложений1}  
else {последовательность предложений2} end_if
```

Алгоритм: если {выражение} истинно, выполняется {последовательность предложений1}, иначе выполняется {последовательность предложений2}.

```
if {выражение1} then {последовательность предложений1}  
elseif {выражение2} then {последовательность предложений2}  
...  
elseif {выражениеN} then {последовательность предложенийN}  
else {последовательность предложений} end_if
```

Алгоритм: выполняется первая по порядку {последовательность предложений}, для которой соответствующее {выражение} истинно. Если все {выражения} ложны, выполняется {последовательность предложений}, следующая за ключевым словом else.

Блок "else {последовательность предложений}" может отсутствовать. В этом случае, если все {выражения}, следующие за elsif, ложны, никаких действий не производится.

Количество блоков "elsif {выражение} then {последовательность предложений}" не ограничено.

Пример

```
if x>0 and x<=10 then y:=5;  
elseif x>10 and x<20 then y:=15;  
elseif x=20 then y:=20;  
else y:=0;
```

```
end_if;
```

7.3.8.1.3. ОПЕРАТОР CASE

Синтаксис:

```
case {выражение} of  
    {список значений}: {последовательность предложений}  
    ...  
    {список значений}: {последовательность предложений}  
else {последовательность предложений}  
end_case
```

В {выражении} могут использоваться целые числа и целочисленные переменные.

{Список значений} представляет собой один или несколько параметров, разделенных запятой. В данном контексте параметр – это целое число или диапазон. Диапазон задается в следующем виде:

```
{нижняя граница} .. {верхняя граница}
```

Алгоритм: выполняется первая по порядку {последовательность предложений}, для которой результат вычисления {выражения} принадлежит множеству, заданному соответствующим {списком значений}. Если результат вычисления {выражения} не принадлежит ни одному из заданных множеств, выполняется {последовательность предложений}, следующая за ключевым словом else (если блок "else {последовательность предложений}" отсутствует, никаких действий не производится).

Пример

```
case int1+10 of  
    11..20 : out1:=10;  
    21,22,23 : out1:=20;  
    23 : out1:=23; (*не выполняется никогда*)  
    100, 88..98, 99 : out1:=100;  
else out1:=50;  
end_case;
```

7.3.8.1.4. ОПЕРАТОР WHILE

Синтаксис:

```
while {выражение} do {последовательность предложений} end_while
```

Алгоритм: пока {выражение} истинно, выполняется {последовательность предложений}.

Пример

```
locINT:=0;
```

```
while locINT<=10 do
```

```
locINT:=locINT+1;
```

```
end_while;
```

7.3.8.1.5. ОПЕРАТОР REPEAT

Синтаксис:

```
repeat {последовательность предложений} until {выражение} end_repeat
```

Алгоритм: выполнение {последовательности предложений} повторяется до тех пор, пока {выражение} ложно. Если {выражение} истинно, {последовательность предложений} выполняется 1 раз.

Пример

```
locINT:=1;
```

```
repeat
```

```
locINT:=locINT+2;
```

```
until locINT=11
```

```
end_repeat;
```

7.3.8.1.6. ОПЕРАТОР FOR

Синтаксис:

```
for {инициализация переменной цикла} to {выражение1} by {выражение2} do  
{последовательность предложений} end_for
```

Переменная цикла должна иметь целочисленный тип данных.

Инициализация переменной цикла задается в виде

```
{имя переменной}:= {выражение}
```


Алгоритм: пока значение переменной цикла меньше или равно значению {выражения1} выполняется {последовательность предложений}. По завершении каждого цикла к переменной цикла прибавляется значение {выражения2}; если оно не задано, прибавляется 1.

Пример

```
for i:=1 to 5 do
  if ar[i]=3 then
    INDEX1:=i;
  return;
end_if;
end_for;
```

7.3.8.1.7. ОПЕРАТОР EXIT

Синтаксис:

```
exit
```

Алгоритм: выход из цикла. В случае вложенных циклов выход осуществляется только из текущего цикла и не затрагивает внешние.

Пример

```
out2:=0;
for i:=1 to 3 do
  for k:=1 to 3 do
    if mBOOL then exit; end_if;
  out2:=out2+1;
end_for;
out2:=out2+1;
end_for;
```

7.3.8.1.8. ОПЕРАТОРЫ ОПРЕДЕЛЕНИЯ ПЕРЕМЕННЫХ

- Модификаторы операторов определения переменных

При создании переменных в дереве (см. Операции с переменными) или конфигурировании их в редакторе (см. Редактор переменной) операторы определения переменных добавляются в текст программного компонента автоматически.

Для задания операторов определения переменных вручную может быть использована Легенда. Вкладка 'Заголовок ST' .

Кроме того, операторы определения переменных могут быть заданы вручную непосредственно в тексте программного компонента ST, однако для переменных, объявленных подобным образом, нельзя задать связи.

Определены следующие операторы данного типа:

var

{определение переменной}

...

{определение переменной}

end_var

Объявляет локальную переменную с доступом Local. Значение такой переменной сохраняется между вызовами программного компонента.

var_temp

{определение переменной}

...

{определение переменной}

end_var

Объявляет локальную переменную с доступом Temp. Значение такой переменной не сохраняется между вызовами программного компонента.

var_input

{определение переменной}

...

{определение переменной}

end_var

Объявляет аргумент с доступом Input (предназначен для приема значений).

```

    var_output
    {определение переменной}
    ...
    {определение переменной}
end_var

```

Объявляет аргумент с доступом Output (предназначен для передачи значения).

```

    var_inout
    {определение переменной}
    ...
    {определение переменной}
end_var

```

Объявляет аргумент с доступом InOut (предназначен для передачи и приема значений).

После ключевого слова end_var точка с запятой не обязательна.

Выражение {определение переменной} имеет вид:

```
{имя переменной}: {тип переменной};
```

Переменная инициализируется значением по умолчанию для указанного типа (это значение отображается на вкладке Легенда. Вкладка 'Скомпилированный код'):

```
var aaa: dt; end_var (*aaa=dt#0001-01-01-00:00:00*)
```

Декларация переменной типа ФБ:

```
var fbST_1:Локальная.fbST; end_var
```

При декларации переменной типа ФБ, содержащегося в библиотеке Стандартная, указание библиотеки не обязательно:

```
var R_TRIG_1:R_TRIG; end_var
```

Декларация переменной структурного типа:

```
mSTRUCT:SYSTEM_BOOL_PARAM := (Value := False, TimeStamp := DT#0001-01-01-00:00, Quality := 192);
```

```
{имя переменной}: {тип переменной}:= {значение};
```

При инициализации переменной присваивается указанное значение:

```
var PI:REAL := 3.14159274; end_var
```

```
{имя переменной}: array [{размерности массива}] of {тип переменной};
```

Декларация массива с инициализацией элементов значениями по умолчанию для указанного типа:

```
var ar1:ARRAY [1..3] OF REAL; end_var
```

```
(*ar[1]=ar[2]=ar[3]=0*)
```

```
{имя переменной}: array [{размерности массива}] of {тип переменной}:={начальные значения};
```

Декларация массива с инициализацией элементов указанными значениями:

```
var ar1:ARRAY[1..2,1..2] OF REAL:=[1,2(2),3]; end_var
```

```
(*ar[1,1]=1, ar[1,2]=ar[2,1]=2, ar[2,2]=3*)
```

Выражения {размерности массива} задаются в виде диапазонов изменения индексов массива, разделенных запятой. Диапазон изменения индексов массива имеет вид

```
{нижняя граница} .. {верхняя граница}
```

Выражение {начальные значения} задается в виде строки начальных значений, разделенных запятой, а также может содержать конструкцию

```
{целочисленная константа} ({значение})
```

где {целочисленная константа} задает количество элементов, которым присваивается указанное значение.

Модификаторы операторов определения переменных

Определены следующие модификаторы операторов определения переменных:

RETAIN

Используется с операторами VAR, VAR_INPUT или VAR_OUTPUT. Задает сохранение переменной в дампа в энергонезависимой памяти контроллера и восстановление значения переменной из дампа при "холодном" рестарте:

```
var RETAIN bbb: REAL; end_var
```

В редакторе переменной модификатору RETAIN соответствует значение Сохранять=Да (см. Редактор переменной.'Сохранять').

NON_RETAIN

Используется с операторами VAR, VAR_INPUT или VAR_OUTPUT.

В редакторе переменной модификатору NON_RETAIN соответствует значение Сохранять=Нет (см. Редактор переменной.'Сохранять').

CONSTANT

Используется с оператором VAR (кроме объявления переменной типа ФБ). Задает константу (значение константы не может быть изменено):

```
var CONSTANT PI:REAL := 3.14159274; end_var
```

В редакторе переменной модификатору CONSTANT соответствует значение Сохранять=Константа (см. Редактор переменной.'Сохранять').

NONE

В редакторе переменной модификатору NONE соответствует значение Сохранять=Наследуется (см. Редактор переменной.'Сохранять').

7.3.8.1.9. ОПЕРАТОР ИНДЕКСИРОВАНИЯ ЭЛЕМЕНТОВ МАССИВА

Синтаксис:

```
{имя} [ {индекс 1}, ... {индекс N}]
```

где {имя} – имя переменной или функции, возвращающей массив, а {индекс k} – целое неотрицательное число или целочисленная переменная. Количество индексов зависит от размерности массива.

Алгоритм: возвращает ссылку на элемент массива:

```
locREAL:=ar[3,i]+1;
```

7.3.8.2. ИНФИКСНЫЕ ОПЕРАТОРЫ

Инфиксные операторы представляют собой обозначения/знаки операций, выполняемых над операндами. В качестве операндов могут выступать:

- константы;
- имена переменных (в т.ч. элементов массивов);
- вызовы функций;
- выражения, заключенные в скобки;
- уточненные имена элементов структур.

Из операндов и инфиксных операторов строятся выражения, а в случае использования оператора присвоения – предложения.

При вычислении выражения вызовы функций и выражения, заключенные в скобки, вычисляются в первоочередном порядке. Инфиксные операторы вычисляются в соответствии с их приоритетом. В таблице ниже операторы представлены в порядке убывания приоритета. Операторы равного приоритета вычисляются по порядку их расположения в выражении слева направо. Например, если INT-переменные A, B, C и D имеют значения соответственно 1, 2, 3 и 4, то $A+B-C*ABS(D) = -9$, а $(A+B-C)*ABS(D) = 0$.

Операция	Обозначение	Приоритет
Отрицание (смена знака)	-	Высший
Побитовое НЕ	NOT	
Возведение в степень	**	
Умножение	*	
Деление	/	
Деление по модулю	MOD	
Сложение	+	
Вычитание	-	
Сравнение:	< , > , <= , >=	
на равенство	=	
на неравенство	<>	
Побитовое И	&	
Побитовое И	AND	
Побитовое исключающее ИЛИ	XOR	
Побитовое ИЛИ	OR	

Операция	Обозначение	Приоритет
Присвоение	:=	Низший

7.3.8.3. ОСОБЕННОСТИ ВЫЧИСЛЕНИЙ

7.3.8.3.1. ОСОБЕННОСТИ БИТОВЫХ СТРОК

Присвоение

Если a – переменная типа данных LWORD, DWORD, WORD или BYTE, а MAX – максимальное число для данного типа данных (например, 255 для BYTE), то при записи в a значения $N > MAX$ $a = N - MAX - 1$. Например, BYTE-переменная $a = 0$ после записи в нее значения 256.

Вычитание

Если a и b – переменные типа данных LWORD, DWORD, WORD или BYTE, и $b < a$, то $b - a = MAX - a + b + 1$, где MAX – максимальное значение для данного типа данных. В случае, например, DWORD-переменных $b - a = 16\#FFFFFF - a + b + 1$.

В частности, это свойство используется для корректного вычисления длины временного интервала по отсчетам, полученным с помощью функции TIME, таймер ПЛК.

7.3.8.3.2. РАБОТА С ОТДЕЛЬНЫМИ БИТАМИ

В Master ST поддерживается чтение и запись отдельных битов целочисленных переменных:

VAR

BOOL1:BOOL := False;

BOOL2:BOOL := False;

UINT1:UINT := 16#FFFF;

INT2:INT := 0;

REAL1:REAL := 0;

END_VAR

(*чтение бита 9 UINT1 и запись в бит 0 INT2*)

INT2.0 := UINT1.9;

(*запись в биты INT2*)

```
INT2.1 := BOOL1;
```

```
INT2.2 := TRUE;
```

(*чтение бита 1 UINT1 и запись в BOOL-переменную*)

```
BOOL2 := UINT1.1;
```

(*использование в логическом выражении*)

```
if UINT1.15 then REAL1:=25; end_if;
```

(*использование в качестве операнда*)

```
UINT1.0 := UINT1.0 xor 1;
```

7.3.8.4. КОНСТАНТЫ MASTER ST

В редакторе ST можно использовать константы, которые задаются непосредственно в тексте программы.

Форма записи показана в разделах:

- Числовые константы
- Временные константы
- Строковые константы

Кроме того можно использовать константы, которые определены в библиотеках проекта. Подробное описание в разделе: Работа с Константами

7.3.8.4.1. ЧИСЛОВЫЕ КОНСТАНТЫ

При задании числовой константы допускаются одиночные символы подчеркивания между цифрами.

Булевы числа представляются 0 или 1 или ключевыми словами FALSE или TRUE соответственно.

Десятичные целочисленные константы состоят из ненулевой цифры, за которой следует последовательность десятичных цифр:

-312, 123_361

Вещественные десятичные константы состоят из целой и дробной части, разделенной точкой. Либо целая, либо дробная часть может отсутствовать (вместе с точкой). Числа могут задаваться в формате с плавающей точкой, при этом они сопровождаются суффиксом e (или E) с указанием десятичного порядка:

1.3, 13, 0.13, -0.13E2, 0.13e+2, 123_234e-5

Десятичные числа и их порядки могут иметь знак (+ или -).

Двоичные целочисленные константы начинаются с префикса 2#, за которым следует двоичное число:

2#1001, 2#1111_1111

Восьмеричные целочисленные константы начинаются с префикса 8#, за которым следует восьмеричное число:

8#777, 8#1_750

Шестнадцатеричные константы начинаются с префикса 16#, за которым следует шестнадцатеричное число (буквы a...f можно задавать как в нижнем, так и в верхнем регистре):

16#eF7, 16#FF_FF

Для изменения представления констант формата DEC, BIN, HEX и OCT (или явного задания типа данных) используются *буквальные константы*, которые начинаются с префиксов BOOL#, SINT#, INT#, DINT#, LINT#, USINT#, UINT#, UDINT#, ULINT#, REAL# и LREAL#. Например, UINT#16#20 – это UINT представление HEX-числа 16#20.

7.3.8.4.2. ВРЕМЕННЫЕ КОНСТАНТЫ

- Временные интервалы
- Абсолютное время

Временные интервалы

Временной интервал состоит из префикса t# или time#, за которым следует запись в виде

<дни>d<часы>h<минуты>m<секунды>s<миллисекунды>ms

Любая составляющая может быть опущена (например, запись t#5h30s3.5ms является корректной и означает 5 часов 30 секунд 3.5 миллисекунды).

Последняя оставшаяся может содержать дробное число. Разрешается "переполнение" старшей составляющей и разделение составляющих символами подчеркивания: t#27h_13m_13s.

Поскольку интервал по смыслу – это разность двух абсолютных времен, для него может быть задано, в том числе, отрицательное значение: t#-5h.

Абсолютное время

Дата состоит из префикса d# или date#, за которым следует запись в виде уууу-mm-dd (год, месяц, день).

Время дня состоит из префикса `tod#` или `time_of_day#`, за которым следует запись в виде `hh:mm:ss.ms` (час, минуты, секунды, миллисекунды).

Константа "дата и время" состоит из префикса `dt#` или `date_and_time#`, за которым следует запись в виде `уууу-мм-дд-hh:mm:ss` (год, месяц, день, часы, минуты, секунды).

7.3.8.4.3. СТРОКОВЫЕ КОНСТАНТЫ

Строковые константы включают 1-байтовые и 2-байтовые символы.

Строка 1-байтовых символов – это последовательность 0 или более символов из строки 00 набора ISO/IEC 10646, которая предваряется и заканчивается одиночной кавычкой (`'`). В строке 1-байтовых символов трехсимвольная комбинация знака доллара (`$`) и двух последующих шестнадцатеричных цифр интерпретируется как шестнадцатеричное представление 8-битового кода символа. (см. свойство 1 в таблице ниже).

Строка 2-байтовых символов – это последовательность 0 или более символов из набора ISO/IEC 10646, которая предваряется и заканчивается двойной кавычкой (`"`). В строке 2-байтовых символов пятисимвольная комбинация знака доллара (`$`) и четырех последующих шестнадцатеричных цифр интерпретируется как шестнадцатеричное представление 16-битового кода символа (см. свойство 2 в таблице ниже).

Свойства символьных строк

NN	Пример	Описание
1		Строки 1-байтовых символов
	<code>"</code>	Пустая строка (нулевой длины)
	<code>'A'</code>	Строка длины 1, содержащая символ A
	<code>' '</code>	Строка длины 1, содержащая символ пробела
	<code>'\$'</code>	Строка длины 1, содержащая символ одиночной кавычки
	<code>'"'</code>	Строка длины 1, содержащая символ двойной кавычки
	<code>'\$R\$L'</code>	Строка длины 2, содержащая символы CR и LF
	<code>'\$0A'</code>	Строка длины 1, содержащая символ LF

NN	Пример	Описание
	'\$\$1.00'	Строка длины 5, которая печатается как "\$1.00"
	'ÄË' '\$C4\$CB'	Эквивалентные строки длины 2
2	Строки 2-байтовых символов	
	""	Пустая строка (нулевой длины)
	"A"	Строка длины 1, содержащая символ A
	" "	Строка длины 1, содержащая символ пробела
	""	Строка длины 1, содержащая символ одиночной кавычки
	"\$""	Строка длины 1, содержащая символ двойной кавычки
	"\$R\$L"	Строка длины 2, содержащая символы CR и LF
	"\$\$1.00"	Строка длины 5, которая печатается как "\$1.00"
	"ÄË" "\$00C4\$00CB"	Эквивалентные строки длины 2
3	Буквальные строки 1-байтовых символов	
	STRING#'OK'	Строка длины 2, содержащая два 1-байтовых символа
4	Буквальные строки 2-байтовых символов	
	WSTRING#'OK'	Строка длины 2, содержащая два 2-байтовых символа

Двухсимвольные комбинации, начинающиеся со знака доллара, интерпретируются как показано в таблице ниже, когда они встречаются в символьной строке.

Двухсимвольные комбинации

NN	Комбинация	Интерпретация
2	\$\$	Знак доллара
3	\$'	Одиночная кавычка
4	\$L или \$l	LF (перевод строки)
5	\$N или \$n	Новая строка
6	\$P или \$p	Новая страница
7	\$R или \$r	CR (перевод каретки)
8	\$T или \$t	Табуляция
9	\$"	Двойная кавычка

ПРИМ. 2. Комбинация '\$' допускается только внутри строк в одиночных кавычках.

ПРИМ. 3. Комбинация '\$"' допускается только внутри строк в двойных кавычках.

7.3.8.5. ЛЕКСИЧЕСКАЯ СТРУКТУРА

В алфавит языка Master ST входят:

- прописные и строчные буквы;
- цифры 0, 1, ...9;
- специальные символы:

() + - # . ' \$: = [] , ; % * / < > &

Из символов алфавита формируются лексемы языка:

- идентификаторы;
- ключевые слова;
- числовые константы;

- временные константы;
- строковые константы;
- инфиксные операторы;
- разделители;
- комментарии.

7.3.8.5.1. ИДЕНТИФИКАТОРЫ

Идентификаторы (имена программ, параметров, функций и функциональных блоков и т.п.) могут состоять из заглавных и строчных букв, цифр и знака подчеркивания '_'.

Первым символом имени не может быть цифра.

Символ подчеркивания не может быть последним символом идентификатора. В начале и внутри идентификатора нельзя использовать два и более символа подчеркивания подряд.

Длина идентификатора не ограничена.

Идентификаторы не чувствительны к регистру, т.е. 'AAA' и 'aaa' являются идентичными именами.

7.3.8.5.2. РАЗДЕЛИТЕЛИ

В качестве разделителей в Master ST используются следующие лексемы:

(* *) + - # . e E ' \$: := () [] , ; .. % => ** * / < > <= >= = <> &

7.3.8.5.3. КОММЕНТАРИИ

Блочный комментарий начинается с символов (* и продолжается до символов *).

Строчный комментарий начинается с символов // и продолжается до конца строки.

Вложенные комментарии не допускаются.

7.4. РЕДАКТОР FBD


Master FBD – язык общего назначения, но менее гибок, чем Master ST.

Программа на языке FBD представляет собой диаграмму, состоящую из последовательно выполняемых FBD-блоков (см. FBD-блок) и связей между параметрами (см. Отображение связей, а также все подразделы разделов Общие элементы программ и Общие средства редактирования программ).

7.4.1. РЕДАКТИРОВАНИЕ FBD-ДИАГРАММЫ

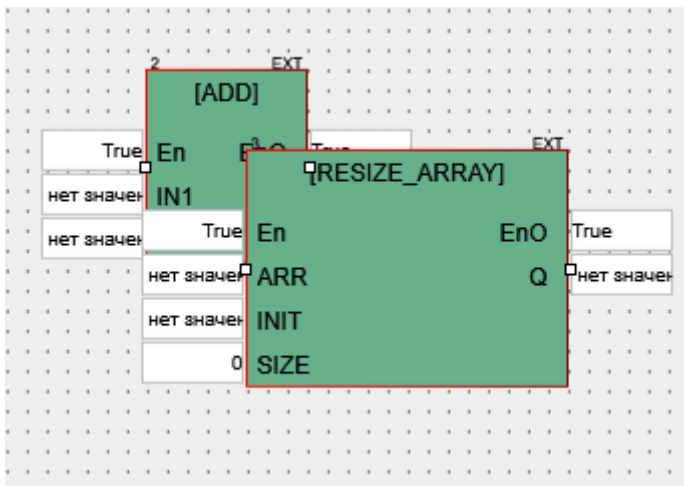
На языке FBD могут быть разработаны программы, функции и функциональные блоки.

Для разработки FBD-элемента нужно выполнить следующие операции:

- перетащить, удерживая левую кнопку мыши, из Палитры или из дерева библиотек нужный элемент в рабочее поле FBD-редактора. При перетаскивании курсор имеет форму , при этом стрелка указывает точку размещения левого верхнего угла FBD-блока.

Для перемещения блока (группы блоков) на диаграмме после размещения также необходимо перетащить его, удерживая ЛК мыши в новое место

Если границы FBD-блоков пересекаются, блоки подсвечиваются красной рамкой (только для предупреждения, на функциональность это не влияет):



- связать нужные входы и выходы блоков, образовав единую диаграмму (алгоритм) (см. Связывание переменных на FBD- и LD-диаграммах);
- связать входы/выходы FBD-диаграммы (входы/выходы блоков) с входами/выходами разрабатываемого FBD-компонента (см. Связывание переменных на FBD- и LD-диаграммах);
- проверить FBD-компонент на наличие ошибок (см. Команда проверки программного компонента).

Для редактирования FBD-диаграммы предусмотрены следующие средства:

- меню FBD-редактора;
- клеммники

- легенда
- инструменты и операции масштабирования (см. Масштабирование в редакторе проекта и Масштабирование диаграмм);
- контекстное меню;
- прокрутка диаграмм;
- операции с надписями;
- операции со связями FBD-диаграммы (см. Отображение связей и Маршрутные точки);
- Панель свойств – в этом окне в разделе Разметка отображаются/редактируются следующие свойства выделенного FBD-блока:

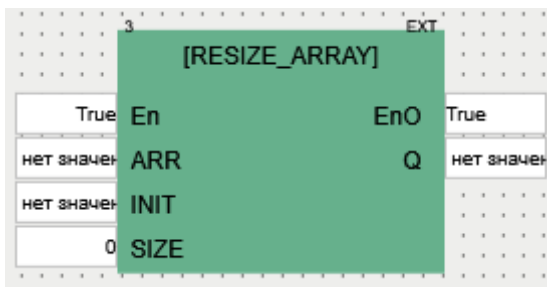
NOT	
Общие	
Имя	NOT
Метки	
Комментарий	НЕ (отрицание, инверсия)
Программное имя	NOT
Полное имя	Объекты.Объект 1.Ресурсы.Программы.Программ
Служебные	
Параметры	
En	<input checked="" type="checkbox"/>
EnO	<input checked="" type="checkbox"/>
Q	
IN	
Разметка	
Координата X	170
Координата Y	280
Заданная ширина	80
Отношения	
Ссылается	Библиотеки.Стандартная.Стандартные функции.Б

Название	Назначение
Координата X	Определяет положение верхнего левого угла блока в рх. Если установлено значение 0, то это значит блок занимает крайнее левое положение. Положение FBD-блока можно также изменять с помощью мыши
Координата Y	Определяет положение верхнего левого угла блока в рх. Если установлено значение 0, то это значит блок занимает крайнее верхнее положение. Положение FBD-блока можно также изменять с помощью мыши
Заданная ширина	Определяет ширину блока в рх. Ширину FBD-блока можно также изменять с помощью мыши

7.4.2. FBD-БЛОК

FBD-блок – это графическое изображение вызова встроенного или созданного пользователем программного элемента: функции, функционального блока

Вид FBD-блока показан на следующем рисунке.



В верхней части блока в квадратных скобках отображается его имя (RESIZE_ARRAY на рисунке).

Параметры слева (En, ARR, INIT, SIZE на рисунке) обозначают входы блока, параметры справа (EnO, Q на рисунке) – выходы. Около входов/выходов находятся поля начальных значений переменных. В панели свойств блока можно изменить начальные значения.

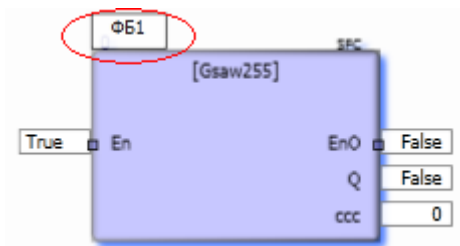
Буквы над блоком справа указывают язык, на котором написан вызываемый блок (например, ST, FBD, LD, SFC). Для встроенных элементов (библиотечных) справа над блоком выводится EXT (как на рисунке).

Число над блоком слева (3 на рисунке) указывает порядковый номер блока в процессе последовательного выполнения. Нумерации блоков начинается с 0.

Цвет блока на диаграмме указывает его разновидность.;

- зеленый – вызов функции;
- фиолетовый – вызов ФБ, программы;

При выделении блока-вызова ФБ над блоком слева отображается имя локальной переменной-декларации этого ФБ в программном компоненте (выполнив переход к редактированию, это имя можно изменить):



7.4.2.1. ПАНЕЛЬ СВОЙСТВ FBD-БЛОКА

Панель свойств FBD-блоков имеет вид:

NOT	
Общие	
Имя	NOT
Метки	
Комментарий	НЕ (отрицание, инверсия)
Программное имя	NOT
Полное имя	Объекты.Объект 1.Ресурсы.Программы.Программ
Служебные	
Параметры	
En	<input checked="" type="checkbox"/>
EnO	<input checked="" type="checkbox"/>
Q	
IN	
Разметка	
Координата X	170
Координата Y	280
Заданная ширина	80
Отношения	
Ссылается	Библиотеки.Стандартная.Стандартные функции.Б

Описание значимых свойств:

Название	Назначение
Группа Параметры	Количество полей в данной категории зависит от количества входов и выходов блока. Используется для установки значений по умолчанию. Также можно установить связь между свойством и параметром проекта, перетаскив одно на другое левой кнопкой мыши, в этом случае связь отобразится и на схеме.
Группа Разметка	
Координат а X	Определяет положение верхнего левого угла блока в рх. Если установлено значение 0, то это значит блок занимает крайнее левое

Название	Назначение
	положение. Положение FBD-блока можно также изменять с помощью мыши
Координата Y	Определяет положение верхнего левого угла блока в рх. Если установлено значение 0, то это значит блок занимает крайнее верхнее положение. Положение FBD-блока можно также изменять с помощью мыши
Заданная ширина	Определяет ширину блока в рх. Ширину FBD-блока можно также изменять с помощью мыши

7.4.3. СЛОИ ДИАГРАММЫ

Графически диаграмма состоит из следующих слоев:

- слой алгоритма (блоков);
- слой связей;
- слой надписей;
- слой отладочных значений. Этот слой добавляется на диаграмму в режиме отладки.

Слой может находиться в активном и неактивном состоянии. Элементы неактивных слоев не реагируют на действия мыши.

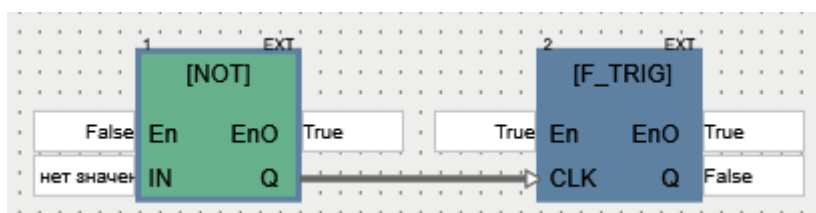
Для управления видимостью и активностью слоев предусмотрены инструменты, которые содержит панель Слой.

7.4.4. ОПЕРАЦИИ СО СВЯЗЯМИ FBD-ДИАГРАММЫ

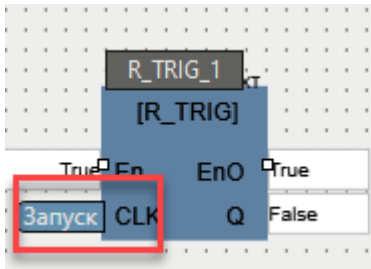
7.4.4.1. ОТОБРАЖЕНИЕ СВЯЗЕЙ

Связи между блоками отображаются как во вкладке Связи легенды редактора FBD, так и в графическом виде на схеме. Каждая межблочная связь на FBD-диаграмме может быть независимо отображена двумя способами :

- линия со стрелкой (стрелка указывает направление передачи значения)



- ссылка:



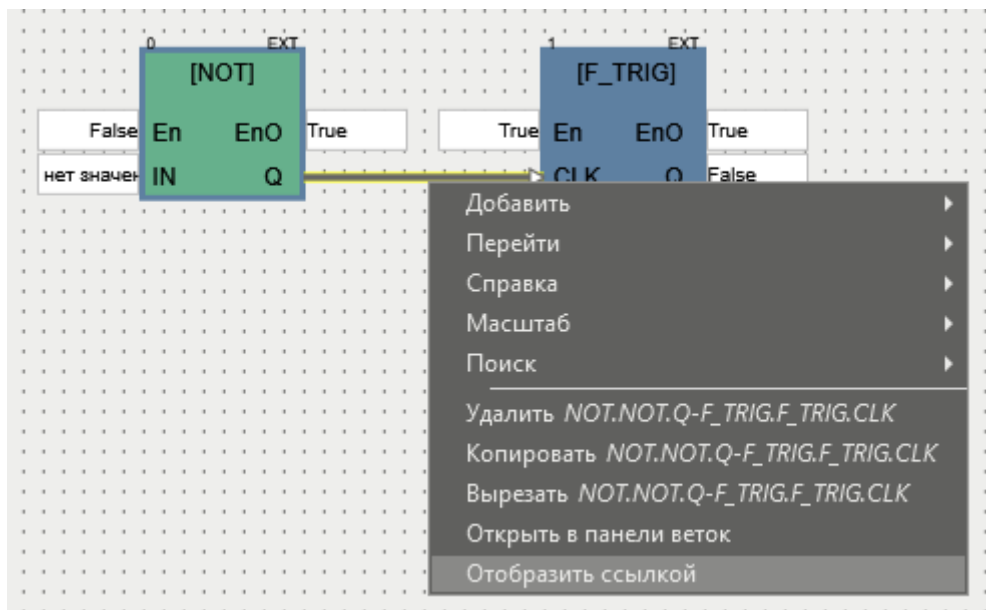
Ширина ножки, в которой отображается ссылка зависит от настройки среды разработки параметра Максимальная ширина связи POU.

Выделенная связь типа Линия подсвечивается желтым.

Выделенная связь типа Ссылка выделяется черным.

Для переключения отображения в контекстном меню связи нужно воспользоваться соответствующей командой:

- Отобразить ссылкой, если связь отображается как линия
- Отобразить линией если связь отображается как ссылка



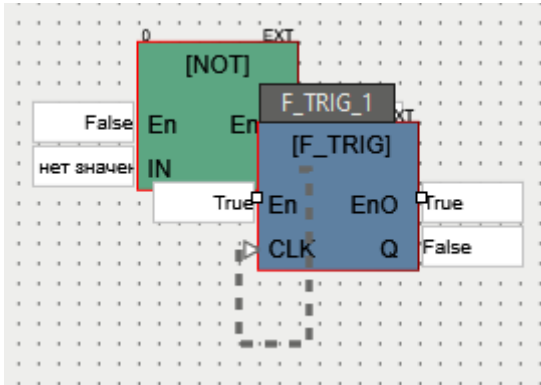
Также для решения этой задачи можно воспользоваться Легендой вкладкой Связи. Нужно или установить или снять флаг Разъединить:

Палитра	Локальные	Параметры	Ошибки	Заголовок ST
Скомпилированный код		Элементы	Надписи	Связи
Название		Источник	Приемник	Сохранять
NOT.NOT.Q-F_TRIG.F_TRIG.CLK	→ ←	NOT.Q	F_TRIG.CLK	нет текста
Программа_FBD_1.Запуск-R_TRIG.R_TRIG.CLK	→ ←	Запуск	R_TRIG.CLK	нет текста

Важно! Связь с параметром программы всегда отображается как ссылка.

Удалить связь можно при помощи пункта контекстного меню Удалить.

Связь отображается пунктирной линией, если редактору не удалось проложить ее корректно:



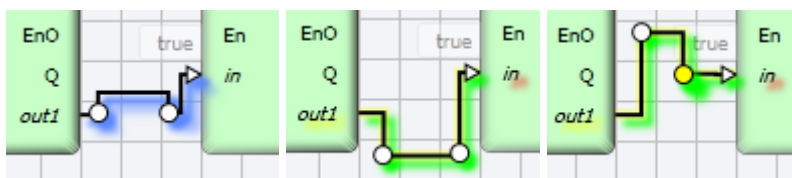
Для исправления подобной ситуации нужно использовать маршрутные точки.

7.4.4.2. МАРШРУТНЫЕ ТОЧКИ

- 'Добавить.Маршрутная точка'

Маршрутные точки используются для изменения расположения связей на диаграмме. Если связь отображается как линия, для установки маршрутной точки нужно выполнить команду 'Добавить.Маршрутная точка' из контекстного меню связи.

Перетаскивая маршрутные точки, можно получить требуемое расположение связи:



Маршрутные точки связи отображаются при наведении курсора на связь.

Маршрутные точки снабжены контекстным меню, содержащим общие команды редактирования.

Кроме типовых средств редактирования, определены следующие операции для работы с маршрутными точками:

- двойной щелчок некоторой точки связи – установка маршрутной точки в данную точку связи;
- двойной щелчок маршрутной точки – удаление этой маршрутной точки.

'Добавить.Маршрутная точка'






Добавить маршрутную точку на связь в точку вызова контекстного меню.

7.4.5. ПАНЕЛЬ ИНСТРУМЕНТОВ FBD-РЕДАКТОРА

При открытии FBD-редактора на вкладку инструментов Проект добавляется панель инструментов редактора FBD:



Она состоит из следующих групп:


Вид	Описание
	Панель Режимы . Переключает режимы работы в редакторе
	Панель Сетка. Служит для работы с сеткой редактора
	Панель Выравнивание. Служит для распределения элементов по схеме.
	Запускает проверку текущей программы.
	Панель Слои.



Доступные инструменты/команды панелей Слои, Сетка и Выравнивание отображаются также в контекстном меню элементов, расположенных на схеме FBD- редактора.

7.4.5.1. ПАНЕЛЬ СЛОИ

Служит для управления видимостью различных слоев схемы. Если иконка нажата, то данный слой отображается в обычном режиме. Если иконка отжата, то слой отображается на схеме бледно. При этом элемент не реагирует на нажатия мыши.



Вид	Назначение
	Управляет видимостью слоя алгоритма (блоков)






Вид	Назначение
	Управляет видимостью слоя связей
	Управляет видимостью слоя надписей

7.4.5.2. ПАНЕЛЬ РЕЖИМЫ

Панель определяет поведение схемы при работе с мышью








Доступны следующие режимы:

Режим	Описание
Выделение рамкой	<p>Чтобы перейти в этот режим, нужно щелкнуть кнопку  (кнопка перейдет в "утопленное" положение , и одновременно кнопка режима масштабирования рамкой перейдет в отключенное положение ).</p> <p>В этом режиме доступно выделение рамкой.</p>
Масштабирование рамкой	<p>Чтобы перейти в этот режим, нужно щелкнуть кнопку  (кнопка перейдет в "утопленное" положение , и одновременно кнопка режима выделения рамкой перейдет в обычное положение ).</p> <p>В этом режиме, если выполняется выделение рамкой, выделенная часть диаграммы увеличивается так, чтобы занять всю видимую область рабочего поля редактора.</p> <p>Для перехода в данный режим на короткое время нужно нажать и удерживать левую кнопку SHIFT.</p>

Режим	Описание
Связывание	 – переключатель режима связывания (включен/отключен). Если режим отключен, связывание переменных невозможно
Отображение отладочных значений	 – переключатель видимости слоя отладочных значений (см. Слои диаграммы). Если слой видим, в режиме отладки около входов/выходов блока отображаются их текущие значения: 









7.4.5.3. ПАНЕЛЬ СЕТКА



Вид	Описание
	Определяет видимость сетки
	Если кнопка нажата, то верхний левый угол элементов схемы будет привязан к узлам сетки. При изменении размеров элементов мышью ширина и высота также будет зависеть от шага сетки.
	Выравнивает выделенные элементы по сетке. По этой команде элемент привязывается к границам ячейки, ближайшим к его левому верхнему углу.
	Выравнивает размеры выделенного элемента по сетке.
	Данный инструмент работает так же, как и комбинированный инструмент масштабирования, но задает размер ячейки сетки (в px).

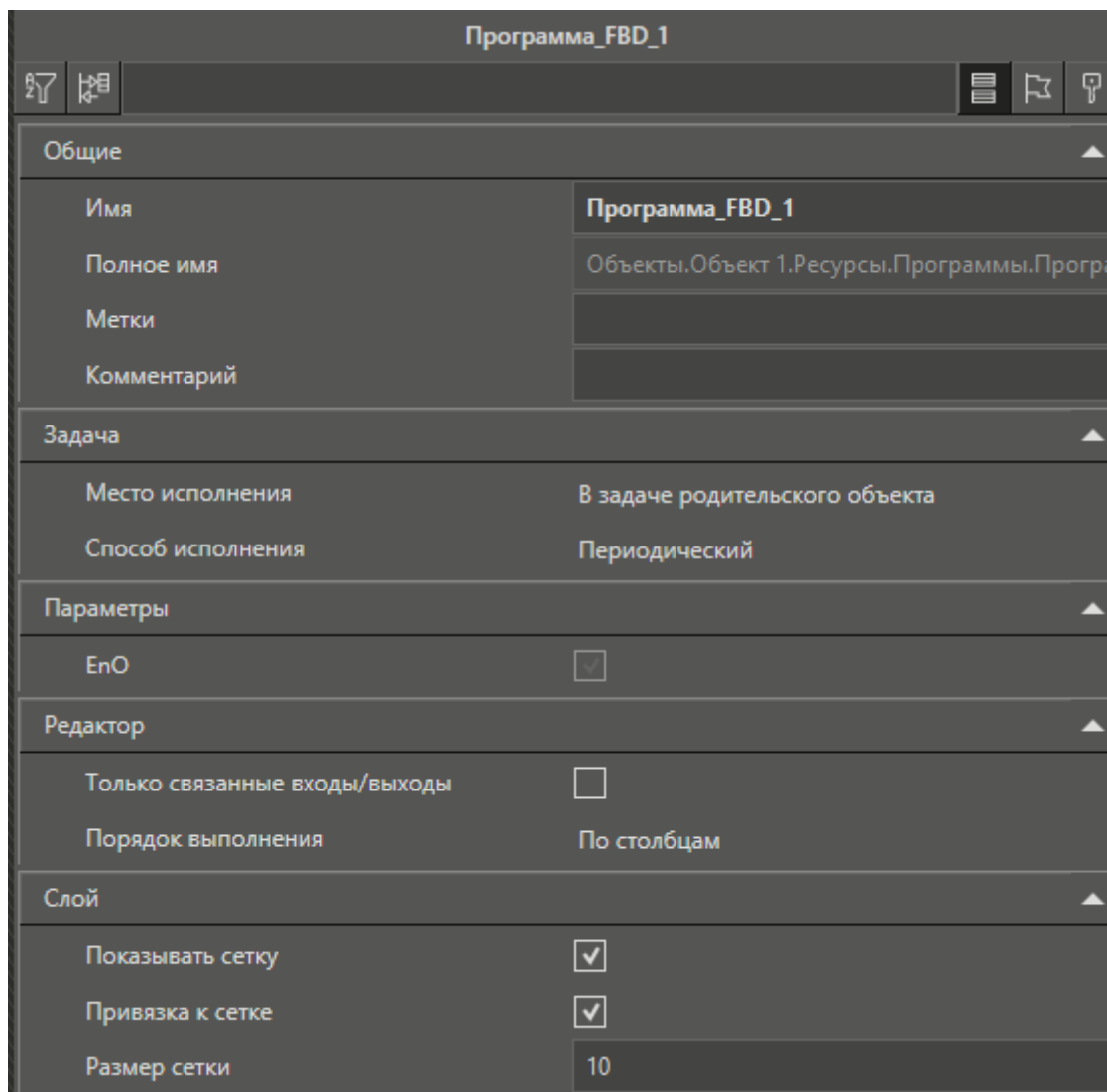
7.4.5.4. ПАНЕЛЬ ВЫРАВНИВАНИЕ

Инструменты этой панели доступны при выделении не менее двух элементов схемы.

Вид	Описание
	По этой команде левые края всех элементов выравниваются по левому краю элемента, который выделен первым
	По этой команде вертикальные оси симметрии всех элементов выравниваются по вертикальной оси симметрии элемента, который выделен первым
	По этой команде правые края всех элементов выравниваются по правому краю элемента, который выделен первым
	По этой команде верхние края всех элементов выравниваются по верхнему краю элемента, который выделен первым.
	По этой команде горизонтальные оси симметрии всех элементов выравниваются по горизонтальной оси симметрии элемента, который выделен первым.
	По этой команде нижние края всех элементов выравниваются по нижнему краю элемента, который выделен первым.
	По этой команде выделенные элементы располагаются с равными промежутками между собой по горизонтали. Иконка становится активной, если выделено три и более элемента
	По этой команде выделенные элементы располагаются с равными промежутками между собой по вертикали. Иконка становится активной, если выделено три и более элемента

7.4.6. ПОРЯДОК ИСПОЛНЕНИЯ FVD-БЛОКОВ

Порядок исполнения FBD-блоков зависит от их положения на схеме и от значения свойства программы Порядок выполнения:

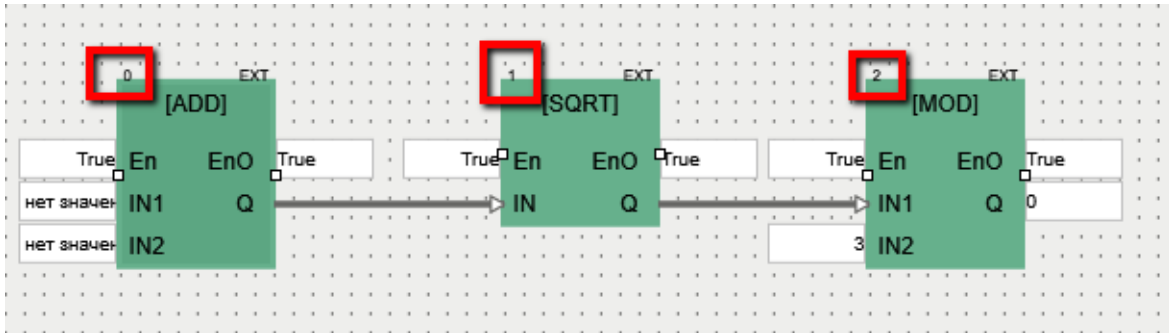


Если значение свойства По столбцам, то FBD-блоки исполняются сверху вниз слева направо. Таким образом элемент, у которого значение координаты X меньше других, вычисляется раньше. В случае если элементы имеют одинаковые координаты по оси X, то первым вычисляется тот, у которого меньше значение координаты Y. Верхний левый угол схемы соответствует координате (0,0). Данный способ используется по умолчанию.

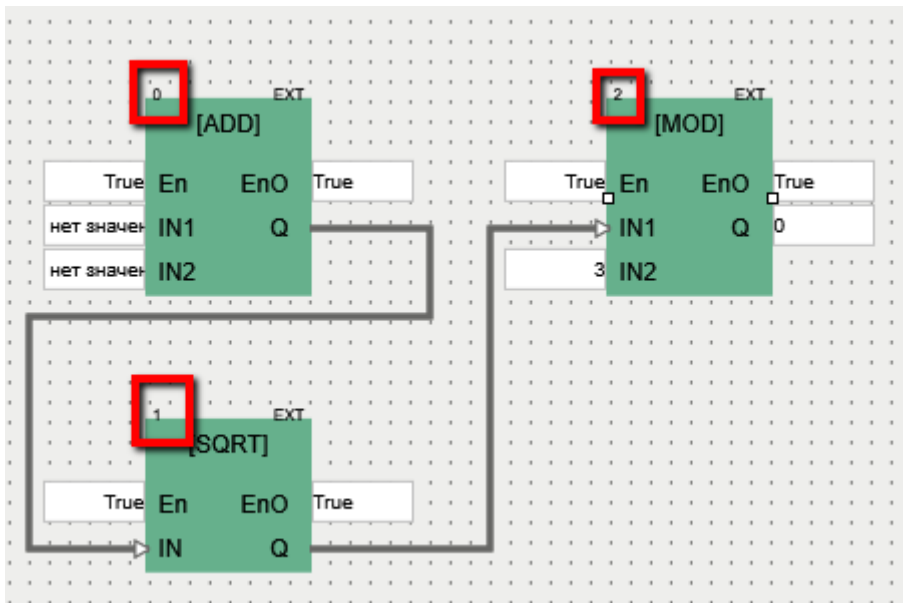
Если значение свойства По строкам, то FBD-блоки исполняются слева направо сверху вниз. Таким образом элемент, у которого значение координаты Y меньше других, вычисляется раньше. В случае если элементы имеют одинаковые координаты по оси Y, то первым вычисляется тот, у которого меньше значение координаты X. Верхний левый угол схемы соответствует координате (0,0).

Чтобы изменить значение свойства для всего проекта нужно в панели свойств проекта выбрать настройку Порядок выполнения FBD по умолчанию

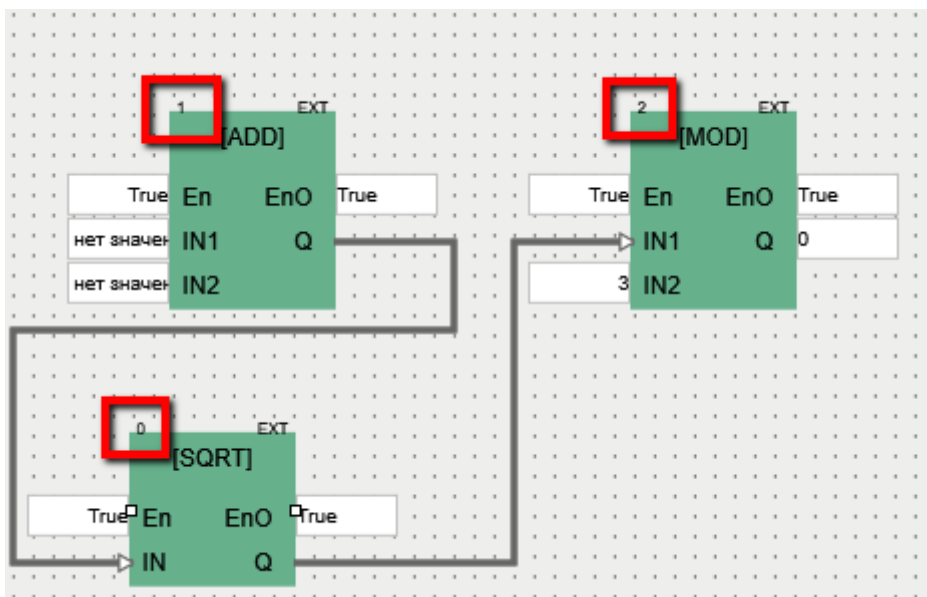
Порядковый номер исполнения блока отображается над блоком слева:



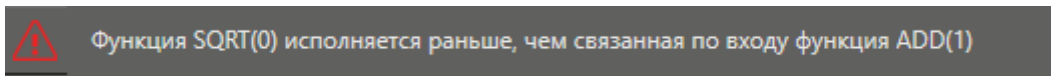
На приведенном выше рисунке свойство Порядок исполнения равно Слева направо и видно, что сначала выполняется блок ADD, затем SQRT и далее MOD. Такой же порядок исполнения будет и в таком случае:



Если блок SQRT переместить левее, то порядок блоков изменится.



Видно, что блок Sqrt будет исполняться до блока ADD, а т.к. выход блока ADD является источником данных для входа Sqrt, то при проверке программы в легенде на вкладке Ошибки получим:



В тексте ошибки в скобках после имени блока указывается его порядковый номер исполнения. Чтобы исправить ошибку необходимо переместить блок Sqrt правее, таким образом, чтобы его порядковый номер исполнения стал больше, чем у блока ADD

7.5. РЕДАКТОР LD

Язык Master LD предназначен в основном для создания логических программ.

Программа языке LD представляет собой диаграмму, состоящую из одной или нескольких основных силовых линий, заключенных между левой и правой шинами и содержащих последовательно выполняемые LD- и FBD-блоки.

Для программирования на языке LD используются следующие элементы алгоритма (содержатся в библиотеке Стандартная.LD):

- силовая линия;
- встроенная LD-функция (LD-блок);
- LD-ветвление;
- LD-терминатор.

Кроме того, для разработки программных компонентов на языке LD могут использоваться не только встроенные LD-функции, но и все остальные встроенные или определенные пользователем функции, программы или функциональные блоки. Вызов такого программного компонента отображается на LD-диаграмме как FBD-блок.

7.5.1. СЛОИ LD-ДИАГРАММЫ

Графически LD-диаграмма содержит те же слои, что и FBD-диаграмма (см. Слои диаграммы).

7.5.2. РЕДАКТИРОВАНИЕ LD-ДИАГРАММЫ

На языке LD может быть разработан любой программный компонент – программа, функция и функциональный блок.

Для разработки LD-компонента нужно выполнить следующие операции:

- задать переменные (см. Операции с переменными);
- выполнить размещение элементов алгоритма на LD-диаграмме;
- задать связанные переменные LD-блоков (см. Операции со связанными переменными);
- связать нужные входы и выходы FBD-блоков, образовав единую диаграмму (алгоритм) (см. Связывание переменных на FBD- и LD-диаграммах);
- связать входы/выходы LD-диаграммы (входы/выходы блоков) с входами/выходами разрабатываемого LD-компонента (см. Связывание переменных на FBD- и LD-диаграммах);
- проверить LD-компонент на наличие ошибок (см. Команда проверки программного компонента).

Для редактирования LD-диаграммы предусмотрены следующие средства:

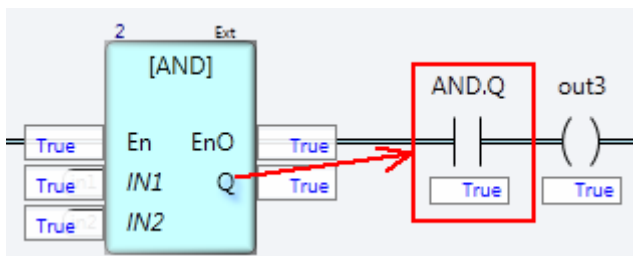
- меню LD-редактора;
- инструменты и операции масштабирования (см. Масштабирование в редакторе проекта и Масштабирование диаграмм);
- общие команды редактирования;
- прокрутка диаграмм;
- операции с надписями;
- операции со связями FBD-диаграммы (см. Отображение связей и Маршрутные точки);
- окно 'Свойства';
- различные редакторы, клеммники и легенда.

7.5.2.1. РАЗМЕЩЕНИЕ ЭЛЕМЕНТОВ АЛГОРИТМА НА LD-ДИАГРАММЕ

Для размещения элемента алгоритма на LD-диаграмме используются следующие инструменты и операции (см. также Редактирование диаграмм в дереве и Операции с переменными):

- команда 'Добавить' контекстного меню редактора или силовой линии;
- перетаскивание из любого окна/редактора, в котором элемент отображается, – например, перетаскивание на силовую линию программного компонента из дерева структуры, перетаскивание на диаграмму силовой линии из палитры (см. Легенда. Вкладка 'Палитра' и Панель 'Палитра LD') и т.п.;

- перетаскивание на силовую линию входа или выхода данного LD-компонента (из дерева структуры, клеммника и т.п.). Перетаскивание входа приводит к размещению LD-блока Контакт ||, перетаскивание выхода – к размещению LD-блока Катушка (). В обоих случаях перетаскиваемый вход/выход автоматически назначается связанной переменной;
- перетаскивание на силовую линию входа или выхода FBD-блока, размещенного на этой же LD-диаграмме. Перетаскивание входа приводит к размещению LD-блока Катушка (), перетаскивание выхода – к размещению LD-блока Контакт ||. В обоих случаях перетаскиваемый вход/выход автоматически назначается связанной переменной:

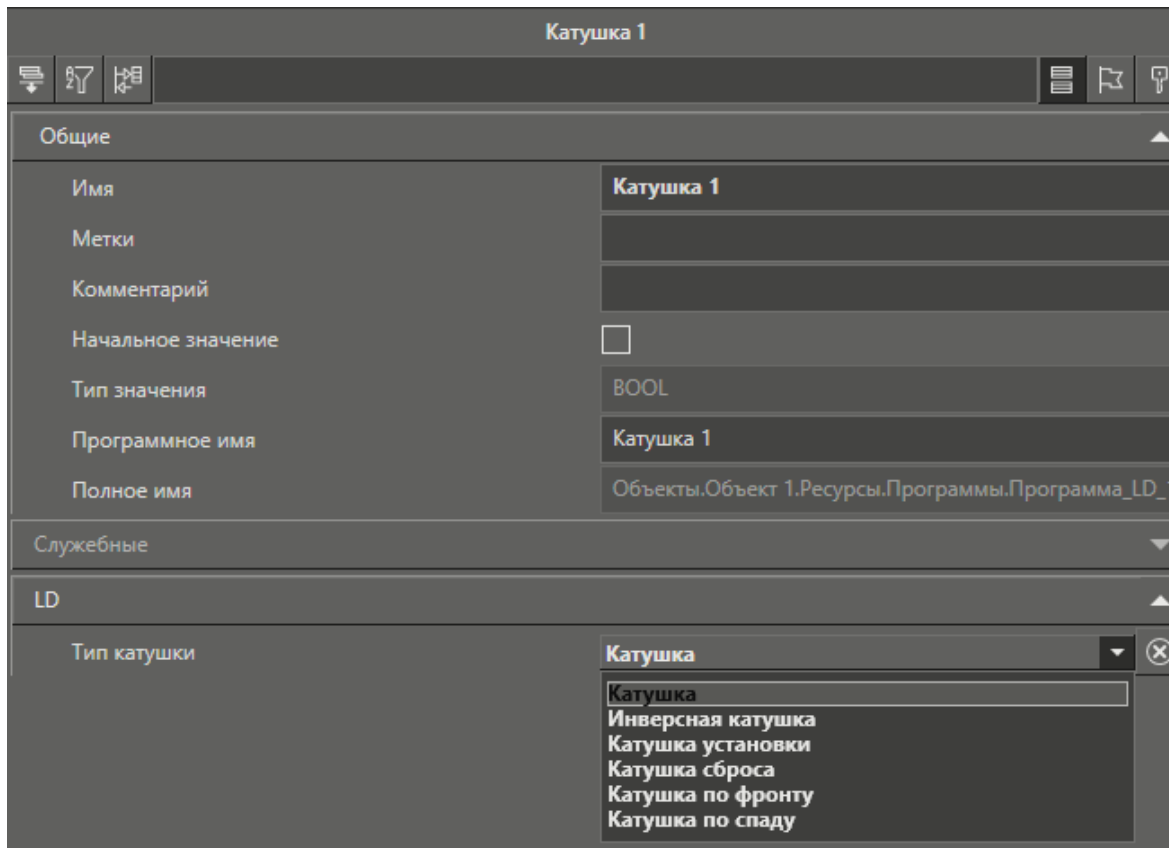


7.5.2.2. ПЕРЕМЕЩЕНИЕ ЭЛЕМЕНТА LD-АЛГОРИТМА

Для перемещения элемента алгоритма после его размещения на диаграмме используется перетаскивание.

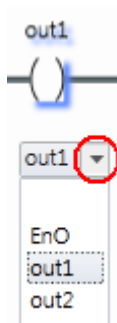
7.5.2.3. ИЗМЕНЕНИЕ ТИПА КОНТАКТА/КАТУШКИ ПОСЛЕ РАЗМЕЩЕНИЯ


Для изменения типа выделенного контакта или катушки (см. выделение) используется панель Свойства (параметр Тип контакта/Тип катушки раздела LD):



7.5.2.4. ОПЕРАЦИИ СО СВЯЗАННЫМИ ПЕРЕМЕННЫМИ

При выделении LD-блока (см. выделение) отображается выпадающий список входов/выходов редактируемого LD-компонента, доступных для использования в качестве связанной переменной блока. Для задания/замены связанной переменной нужно щелкнуть переменную в этом списке (чтобы удалить связанную переменную, нужно щелкнуть в списке пустую позицию):



Для замены связанной переменной можно также выполнить перетаскивание выделенной переменной на блок (курсор в этом случае принимает форму ).

Описанные выше методы не работают, если в качестве связанной переменной требуется назначить переменную FBD-блока, размещенного на этой же LD-диаграмме.

В этом случае нужно удалить блок и создать его заново (см. Размещение элементов алгоритма на LD-диаграмме).

7.5.3. ПАНЕЛЬ ИНСТРУМЕНТОВ LD-РЕДАКТОРА

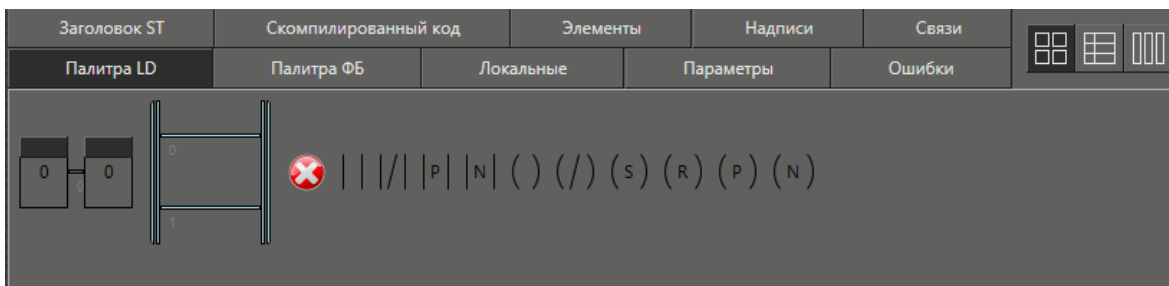
Панель инструментов редактора имеет вид:



Вид	Описание
	Панель Режимы . Переключает режимы работы в редакторе
	Запускает проверку текущей программы.
	Панель Слои.

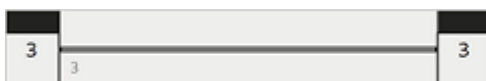
7.5.4. ПАЛИТРА LD

Легенда LD-редактора содержит вкладку Палитра LD, которая содержит все элементы библиотеки Стандартная, категория LD.

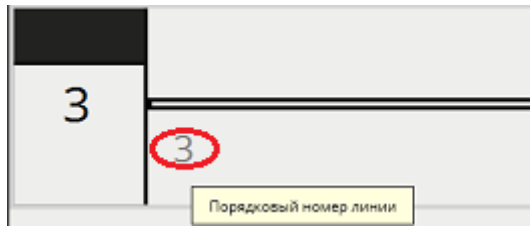


7.5.4.1. СИЛОВАЯ ЛИНИЯ

Основная силовая линия отображается на LD-диаграмме в виде горизонтальной линии, заключенной между левой и правой шинами; шины отображаются в виде вертикальных полос, на которых указывается номер силовой линии:



Номер основной силовой линии отображается также в области, показанной на рисунке ниже (при наведении курсора номер увеличивается, а также отображается всплывающая подсказка):

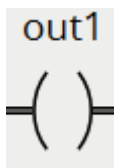


Значение левой шины всегда равно 1 (аналог положительной шины питания).

Существуют также вспомогательные силовые линии, их содержит элемент алгоритма LD-ветвление.

7.5.4.2. LD-БЛОК

LD-блок – это графическое изображение вызова встроенной LD-функции (встроенные LD-функции содержатся в библиотеке Стандартная.LD – см. Библиотека 'LD'), при этом в качестве изображения используется обозначение выполняемой функции (() на рисунке):



Отрезок слева обозначает вход блока, отрезок справа – выход. Все встроенные LD-функции имеют один вход (in) и один выход (out).

В FBD-блоках на LD-диаграмме в качестве in используется En, а в качестве out – EnO.

Над блоком отображается имя связанной переменной (out1 на рисунке выше).

Связанной переменной называется переменная, от значения которой зависит выполняемое блоком действие или значение которой устанавливается в процессе выполняемого блоком действия. Связанная переменная задается пользователем.

7.5.4.3. LD-ВЕТВЛЕНИЕ



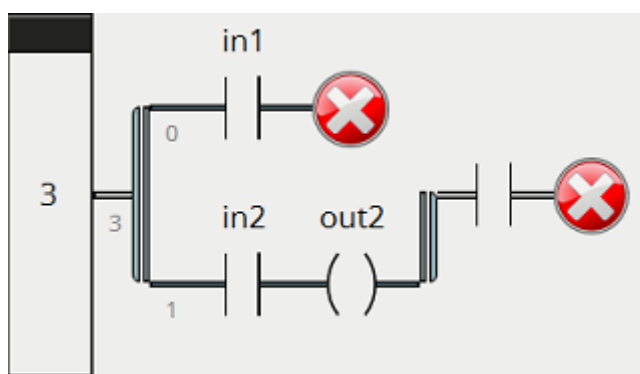
Ветвление содержит две параллельные вспомогательные силовые линии, заключенные между левой и правой вспомогательными шинами (см. силовая линия). Вспомогательная силовая линия имеет номер <номер основной силовой линии>.<номер вспомогательной силовой линии> (2.2 на рисунке выше).

Значение правой вспомогательной шины равно результату логического сложения (OR) значений выходов блоков, связанных с этой шиной.

7.5.4.4. LD-ТЕРМИНАТОР



Терминатор запрещает выполнение алгоритма силовой линии, на которой располагается, справа от себя (до конца ветвления при размещении на вспомогательной силовой линии или до правой шины при размещении на основной силовой линии). При размещении терминатора запрещаемый алгоритм скрывается (но не удаляется, и после удаления терминатора восстанавливается):



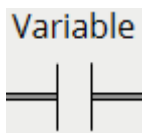
7.5.4.5. ДРУГИЕ ЭЛЕМЕНТЫ LD

- Контакт ||
- Замкнутый контакт |/|
- Контакт по фронту |P|

- Контакт по спаду |N|
- Катушка ()
- Инверсная катушка (/)
- Катушка установки (S)
- Катушка сброса (R)
- Катушка по фронту (P)
- Катушка по спаду (N)

Помимо элементов алгоритма силовая линия, LD-ветвление и LD-терминатор (см. Редактор LD), библиотека Стандартная.LD содержит встроенные LD-функции.

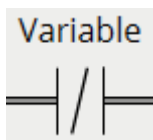
Контакт ||



Если Variable $\diamond 0$, то out = in.

Если Variable = 0, то out = 0.

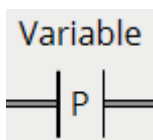
Замкнутый контакт |/|



Если Variable = 0, то out = in.

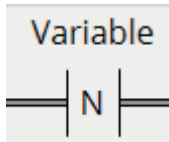
Если Variable $\diamond 0$, то out = 0.

Контакт по фронту |P|



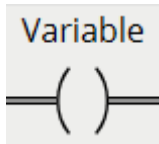
Если in = 1, а Variable меняет свое значение с 0 на любое ненулевое, то на один (следующий) такт out = 1. Во всех остальных случаях out = 0.

Контакт по спаду |N|



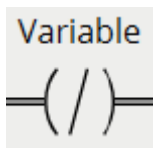
Если $in = 1$, а Variable меняет свое значение с любого ненулевого на 0, то на один (следующий) такт $out = 1$. Во всех остальных случаях $out = 0$.

Катушка ()



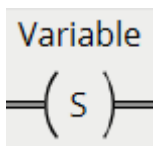
Variable = out = in.

Инверсная катушка (/)



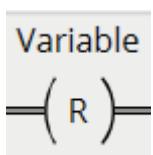
out = in, Variable = NOT in.

Катушка установки (S)



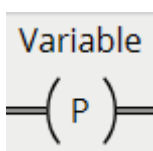
Переменная Variable принимает значение 1 при подаче 1 на вход и в дальнейшем может быть сброшена только с помощью катушки сброса. Во всех случаях $out = 1$.

Катушка сброса (R)



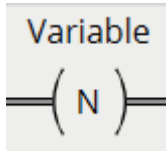
Переменная Variable принимает значение 0 при подаче 1 на вход и в дальнейшем может быть установлена только с помощью катушки установки. Во всех случаях $out = 1$.

Катушка по фронту (P)



Если in изменяется с 0 на любое ненулевое, то на один (следующий) такт $Variable = 1$.
Во всех случаях $out = in$.

Катушка по спаду (N)



Если in изменяется с любого ненулевого на 0, то на один (следующий) такт $Variable = 1$. Во всех случаях $out = in$.

7.6. РЕДАКТОР SFC

Язык SFC позволяет создавать программные компоненты в виде алгоритма, состоящего из SFC-шагов и SFC-условий (условных переходов), взаимодействующих между собой по прямым связям (см. Редактирование SFC-диаграммы).

Для SFC-шагов задаются выполняемые действия, для SFC-условий – условия переходов к элементам алгоритма.

Для программирования на языке SFC используются следующие элементы алгоритма (содержатся в библиотеке Стандартная.SFC):

- SFC-ветка;
- SFC-шаг;
- SFC-условие;
- SFC-переход;
- SFC-терминатор;
- SFC-ветвление выбора;
- параллельное SFC-ветвление.

7.6.1. СЛОИ SFC-ДИАГРАММЫ

За исключением слоя связей, SFC-диаграмма содержит те же графические слои, что и FBD-диаграмма (см. Слои диаграммы).

7.6.2. РЕДАКТИРОВАНИЕ SFC-ДИАГРАММЫ

Элементы SFC требуют хранения информации о состоянии, поэтому на языке SFC может быть разработана программа или функциональный блок (функция не может быть разработана на SFC).

Для редактирования SFC-диаграммы предусмотрены следующие средства:

- меню SFC-редактора;
- инструменты и операции масштабирования (см. Масштабирование в редакторе проекта и Масштабирование диаграмм);
- общие команды редактирования;
- прокрутка диаграмм;
- операции с надписями;
- различные редакторы, клеммники и легенда.

7.6.2.1. РАЗМЕЩЕНИЕ ЭЛЕМЕНТОВ АЛГОРИТМА НА SFC-ДИАГРАММЕ

Для размещения элемента алгоритма на SFC-диаграмме используются следующие инструменты и операции (см. также Редактирование диаграмм в дереве):

- команда 'Добавить' контекстного меню редактора или SFC-ветки;
- перетаскивание элемента алгоритма на SFC-ветку из библиотеки Стандартная.SFC или палитры (см. Легенда. Вкладка 'Палитра' и Панель 'Палитра').

7.6.3. ПАНЕЛЬ ИНСТРУМЕНТОВ SFC-РЕДАКТОРА

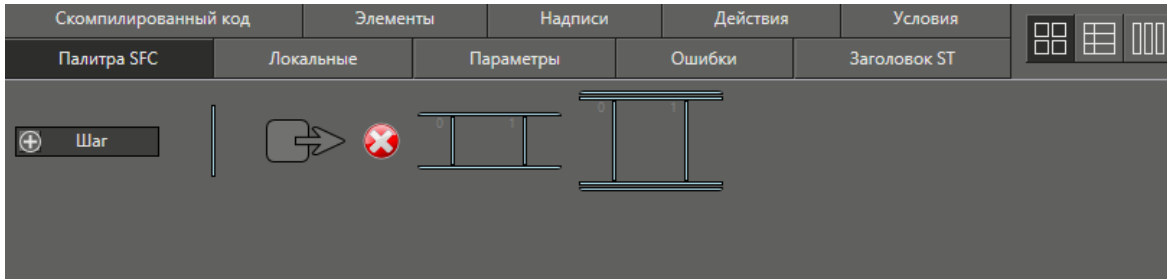
Панель инструментов редактора имеет вид:



Вид	Описание
	Панель Режимы . Переключает режимы работы в редакторе
	Запускает проверку текущей программы.
	Панель Слои.

7.6.4. ПАЛИТРА SFC

Легенда SFC-редактора содержит вкладку Палитра SFC, которая содержит все элементы библиотеки Стандартная, категория SFC.



7.6.4.1. БИБЛИОТЕКА SFC

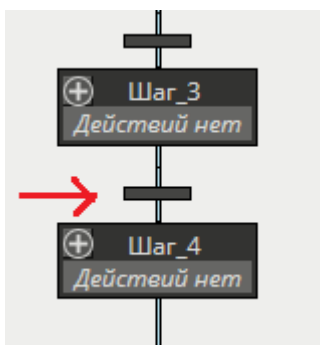
Содержит элементы SFC-алгоритма:

- SFC-ветка;
- SFC-шаг;
- SFC-переход;
- SFC-терминатор;
- SFC-ветвление выбора;
- параллельное SFC-ветвление.


7.6.4.1.1. SFC-ВЕТКА

SFC-ветка – это часть SFC-алгоритма с непосредственными связями между элементами (связи образуются автоматически при размещении элементов алгоритма на ветке).

На рисунке показана ветка (вертикальная линия) при ее размещении на диаграмме (ветка пуста) и ветка с элементами алгоритма:




SFC-алгоритм может содержать несколько веток.

При создании SFC-компонента автоматически создается ветка с начальным шагом (при создании SFC-шага всегда создается SFC-условие  перехода к этому шагу):




7.6.4.1.2. SFC-ШАГ


SFC-шаг (step) – это элемент алгоритма, который может содержать действия (см. Задание SFC-действий). При создании SFC-шага всегда создается SFC-условие  перехода к этому шагу.

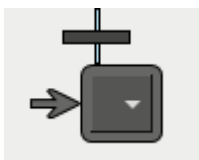
7.6.4.1.3. SFC-УСЛОВИЕ

SFC-условие (условный переход, transition) – это элемент алгоритма, который содержит условия перехода к последующему элементу алгоритма (см. Задание SFC-условий).

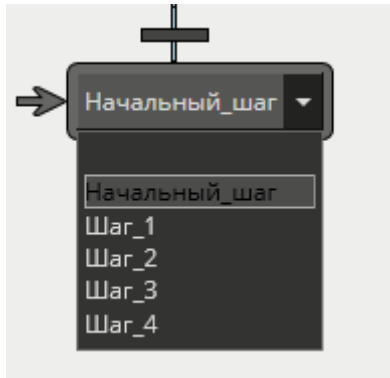
При размещении элементов алгоритма необходимые SFC-условия создаются автоматически, однако с помощью команды 'Добавить.Условие' контекстного меню ветки можно разместить дополнительное SFC-условие  ниже уже размещенного SFC-условия.

7.6.4.1.4. SFC-ПЕРЕХОД

SFC-переход (безусловный переход, jump) – это безусловный переход к указанному шагу, всегда сопровождается предшествующим SFC-условием  . При размещении SFC-переход имеет следующий вид:

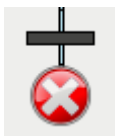


Для указания шага предназначен выпадающий список SFC-перехода:



При размещении SFC-перехода последующий алгоритм удаляется (но не уничтожается, и после удаления перехода восстанавливается).

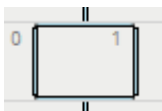
7.6.4.1.5. SFC-ТЕРМИНАТОР



Терминатор запрещает выполнение SFC-ветки, на которой располагается. Данный элемент всегда сопровождается предшествующим SFC-условием. При размещении терминатора последующий алгоритм удаляется (но не уничтожается, и после удаления терминатора восстанавливается).

7.6.4.1.6. SFC-ВЕТВЛЕНИЕ ВЫБОРА

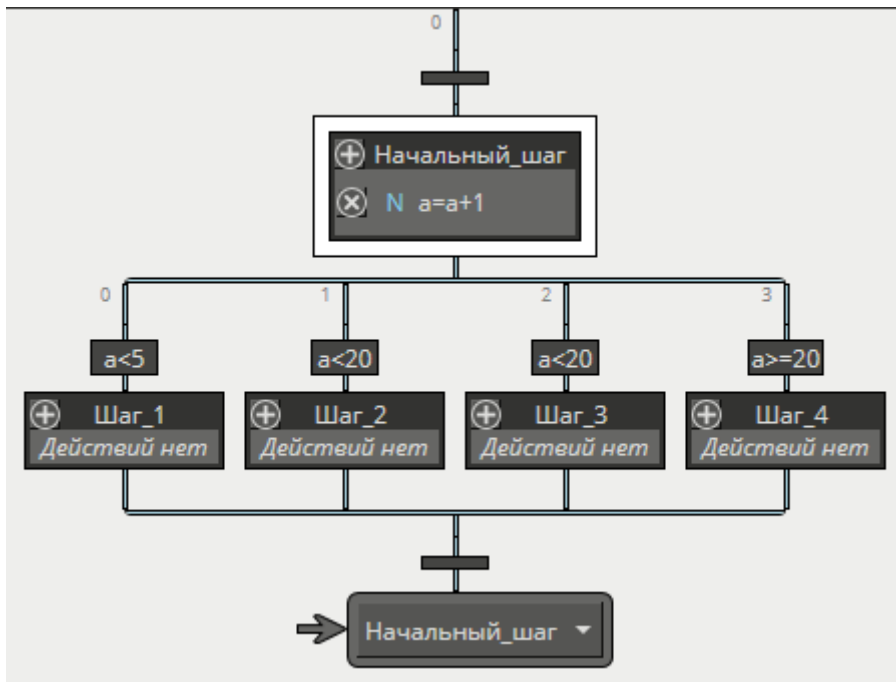
При размещении данный элемент имеет две пустые ветки (вертикальные линии):



С помощью команды 'Добавить.Ветка' контекстного меню горизонтальной линией элемента число веток может быть увеличено:



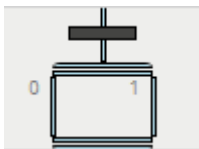
Ветки элемента анализируются слева направо, и выполняется первая (и единственная) ветка, для которой заданное условие истинно. Например, на следующей диаграмме переход к шагу 3 не произойдет никогда:



У SFC-ветвления выбора отсутствует предшествующее SFC-условие.

7.6.4.1.7. ПАРАЛЛЕЛЬНОЕ SFC-ВЕТВЛЕНИЕ

При размещении данный элемент имеет две пустые ветки (вертикальные линии):



С помощью команды 'Добавить.Ветка' контекстного меню горизонтальной линией элемента число веток может быть увеличено:



В отличие от SFC-ветвления выбора, параллельное SFC-ветвление всегда сопровождается предшествующим SFC-условием. Если заданные условия истинны, выполняются все ветки ветвления, в противном случае не выполняется ни одна из веток ветвления.

7.6.5. ЗАДАНИЕ SFC-УСЛОВИЙ

Задать SFC-условие можно одним из следующих способов.

Если не задано ни одно условие, считается, что условный переход всегда разрешен.

Способ 1

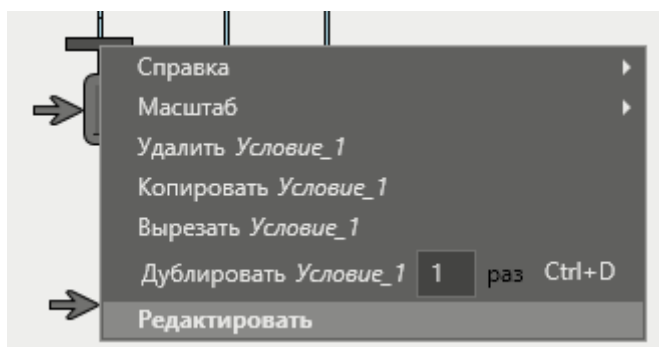
Для задания SFC-условия можно создать условие предварительно в корневой группе Условия программного компонента (см. Структура программы в дереве) и перетащить его на ветку или условие в диаграмме (в первом случае условие будет добавлено, во втором – заменено).

Способ 2

Для задания SFC-условия можно создать условие предварительно в корневой группе Условия программного компонента и перетащить его на группу Алгоритм.<ветка>.Шаг<n>.Ветвление условий И или на компонент Алгоритм.<ветка>.Шаг<n>.Ветвление условий И.<условие> (в первом случае условие будет добавлено, во втором – заменено).

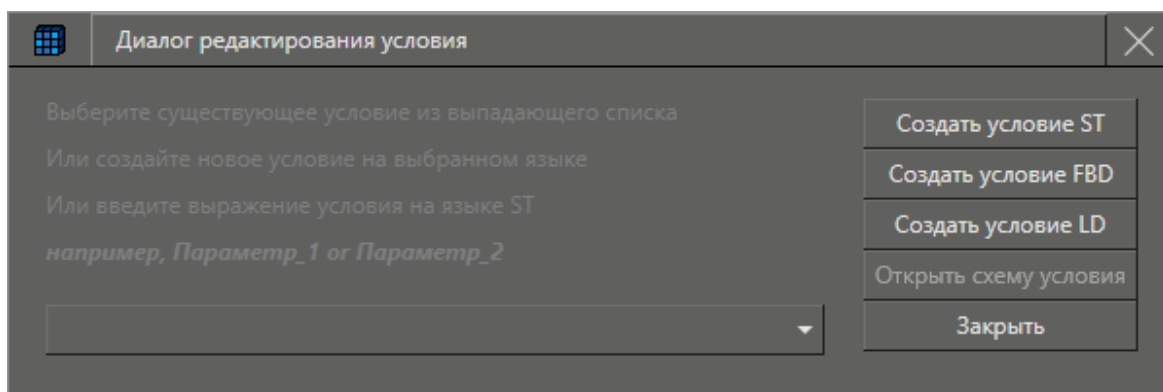
Способ 3

Для задания условия можно выполнить двойной клик мыши его на диаграмме – откроется диалог редактирования SFC-условия – и сконфигурировать условие в диалоге. Диалог открывается по команде Редактировать контекстного меню Условия:

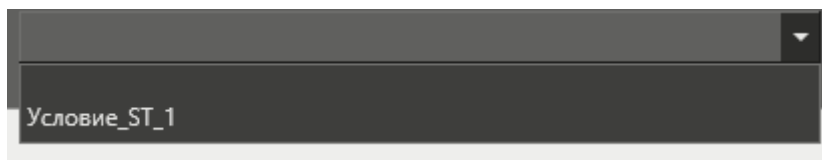


7.6.5.1. ДИАЛОГ РЕДАКТИРОВАНИЯ SFC-УСЛОВИЯ

Диалог имеет вид



Инструмент описания SFC-условия

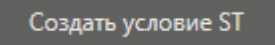


Выпадающий список SFC-условий, которые содержит корневая группа Условия программного компонента, позволяет использовать условия, созданные предварительно.

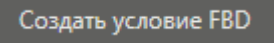
В группе Алгоритм.<ветка>.Шаг_<n>.Условия отображается ссылка на использованное условие.

Окно элемента управления списком представляет собой поле редактирования, в котором условие (логическое выражение) может быть задано вручную на языке ST.

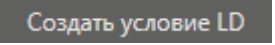
Кнопка создания условия ST

-  – создать SFC-условие на языке ST в корневой группе Условия программного компонента и использовать это условие.

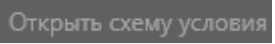
Кнопка создания условия FBD

-  – создать SFC-условие на языке FBD в корневой группе Условия программного компонента и использовать это условие.

Кнопка создания условия LD

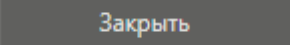

-  – создать SFC-условие на языке LD в корневой группе Условия программного компонента и использовать это условие.

Кнопка вызова редактора условия

-  – открыть выбранное SFC-условие в соответствующем редакторе.

Данная кнопка недоступна, если в качестве условия использовано выражение ST, заданное вручную.

Команда завершения редактирования условия

-  или  – завершить конфигурирование SFC-условия и закрыть диалог.

7.6.6. ЗАДАНИЕ SFC-ДЕЙСТВИЙ

Задать действие для SFC-шага можно одним из следующих способов.

Способ 1

Для задания SFC-действия можно создать действие предварительно в корневой группе Действия программного компонента (см. Структура программы в дереве) и перетащить его на шаг в диаграмме.

Этот способ работает только в том случае, если шаг уже содержит действия.

При перетаскивании действия из группы Действия на действие шага в диаграмме происходит замена действия.

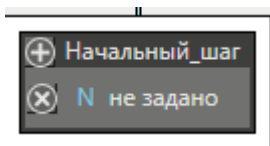
Способ 2

Для задания SFC-действия можно создать действие предварительно в корневой группе Действия программы и перетащить его на группу Действия шага в группе Алгоритм (см. также Редактирование диаграмм в дереве).

Способ 3

Для задания SFC-действия можно выполнить следующую последовательность операций:


- щелкнуть кнопку  шага в диаграмме – в шаг добавится строка действия:



Строка действия содержит инструмент удаления SFC-действия  ;

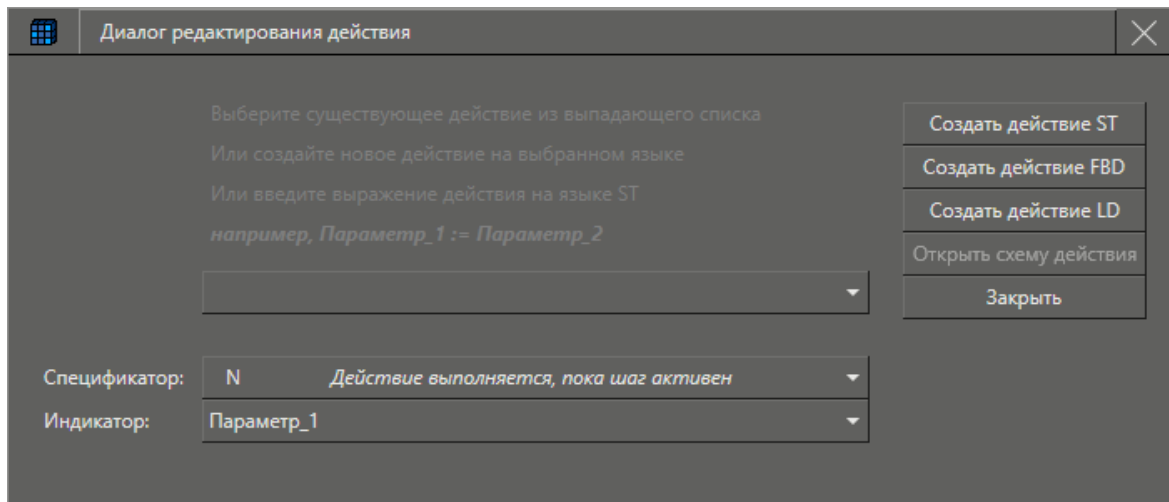
- выполнить двойной клик в строке действия поле описания (**не задано**) – откроется диалог редактирования SFC-действия – и сконфигурировать созданное действие в диалоге. (диалог открывается также по команде Редактировать контекстного меню Действия)

Инструмент удаления SFC-действия

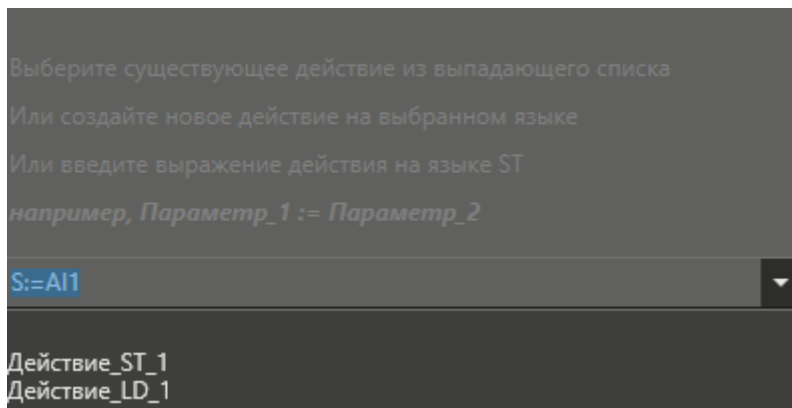
-  – удалить SFC-действие. Удаление SFC-действия помещается в стек UnDo и может быть отменено.

7.6.6.1. ДИАЛОГ РЕДАКТИРОВАНИЯ SFC-ДЕЙСТВИЯ

Диалог имеет вид:



Инструмент описания SFC-действия

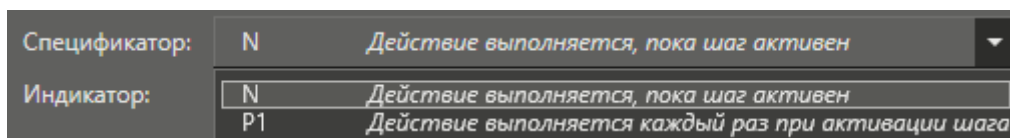


Выпадающий список действий, которые содержит корневая группа 'Действия' программного компонента, позволяет использовать действия, созданные предварительно.

В группе Алгоритм.<ветка>.Шаг_<n>.Действия отображается ссылка на использованное в шаге действие.

Окно элемента управления списком представляет собой поле редактирования, в котором действие может быть задано вручную на языке ST.

Инструмент выбора спецификатора действия

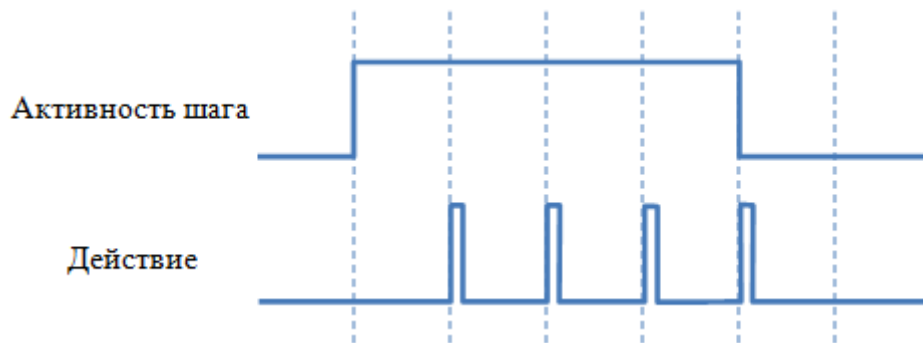


Выпадающий список, в котором выбирается спецификатор действия (action qualifier по IEC 61131-3).

Важно! В текущей версии доступны только спецификаторы N и P1.

Спецификатор N.

На первом цикле действие не выполняется. Далее выполняется на каждом цикле, пока шаг активен. После деактивации шага действие выполняется дополнительно один раз.

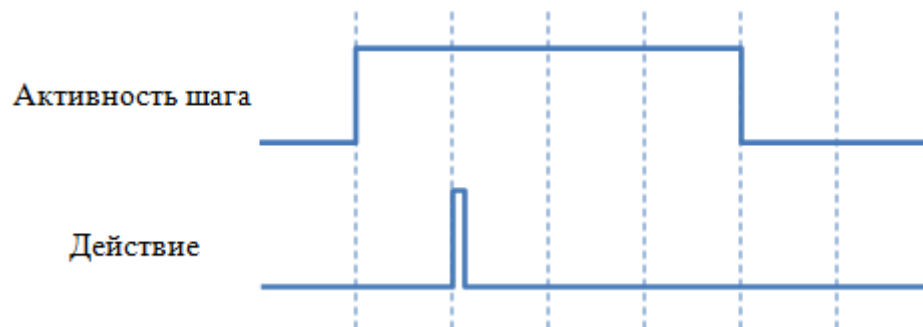


Работает в соответствии с определением стандарта, но со сдвигом на один цикл.

Сдвиг на один цикл будет устранен в следующем релизе системы.

Спецификатор P1.

Действие производится один раз при активации шага с задержкой на один цикл.



Работает в соответствии с определением стандарта, но со сдвигом на один цикл.

Сдвиг на один цикл будет устранен в следующем релизе системы.

Инструмент выбора индикатора действия



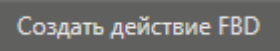
Выпадающий список, в котором выбирается индикатор действия (переменная).

Значение индикатора равно 1 (TRUE), если действие выполняется, или 0 (FALSE) в противном случае.

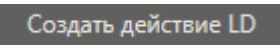
Кнопка создания действия ST

- Создать действие ST – создать SFC-действие на языке ST в корневой группе Действия программного компонента и использовать это действие.

Кнопка создания действия FBD

-  – создать SFC-действие на языке FBD в корневой группе Действия программного компонента и использовать это действие.

Кнопка создания действия LD

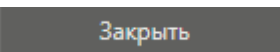

-  – создать SFC-действие на языке LD в корневой группе Действия программного компонента и использовать это действие.

Кнопка вызова редактора действия

-  – открыть выбранное SFC-действие в соответствующем редакторе.

Данная кнопка недоступна, если в качестве действия использовано выражение ST, заданное вручную.

Команда завершения редактирования действия

-  или  – завершить конфигурирование SFC-действия и закрыть диалог.

7.7. РЕДАКТОР C#

Редактор C# представляет собой текстовый редактор, в котором можно создавать программы на языке C#, которые будут работать в исполнительной системе. Если создать функциональный блок на языке C#, то его можно использовать в программах, написанных на других языках и других программах написанных на языке C#.

Важно! Поддерживается только в тех версиях исполнительной системы, которые работают под ОС Windows. Подробнее смотрите в разделе Программы C#. Системные требования

Возможности редактора C#:

- Работать с параметрами MasterSCADA 4D внутри программы, аналогично программам стандарта МЭК 61131-3. Можно устанавливать связи между параметрами T.e. можно использовать те же типы данных и устанавливать те же связи параметров программ и с другими элементами проекта, а также обращаться к глобальным переменным.
- Работать со сложными типами данных.
- Внутри программ C# можно использовать ФБ, созданные в других редакторах.
- Создавать ФБ и использовать в других программах C#
- Подключать и использовать сторонние dll

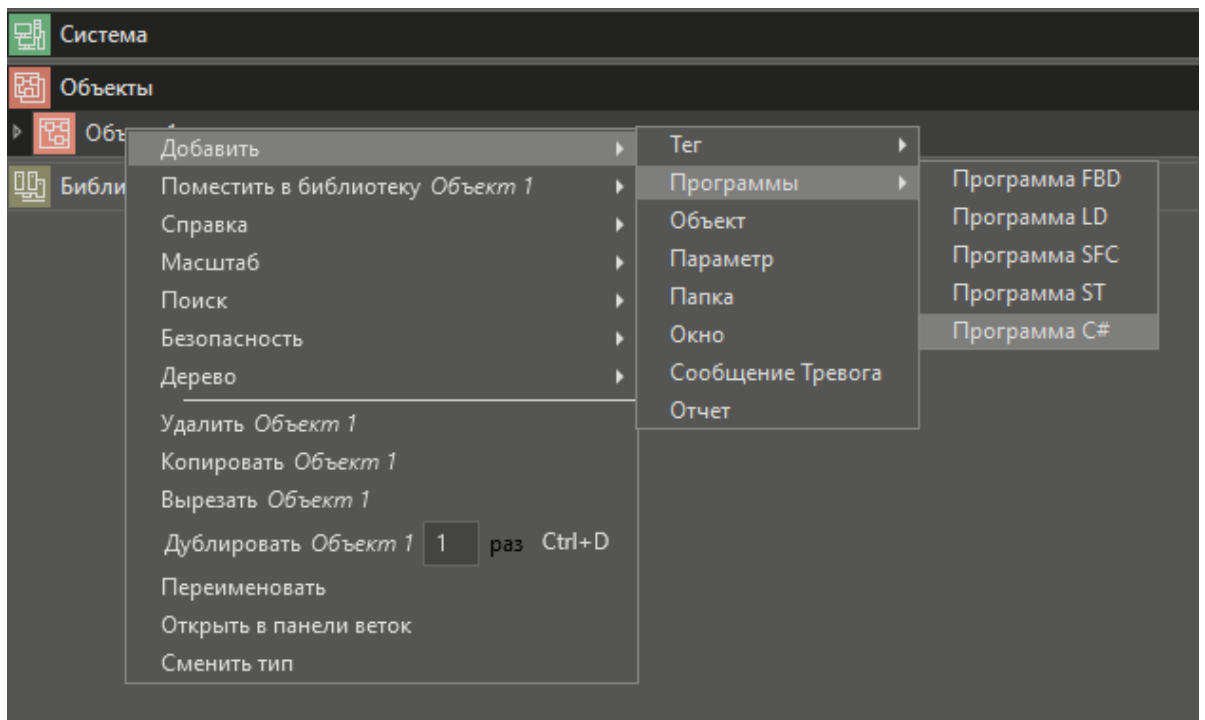
Для использования этих возможностей разработчик проекта должен обладать навыками программирования на языке C#.

Создание программы или ФБ

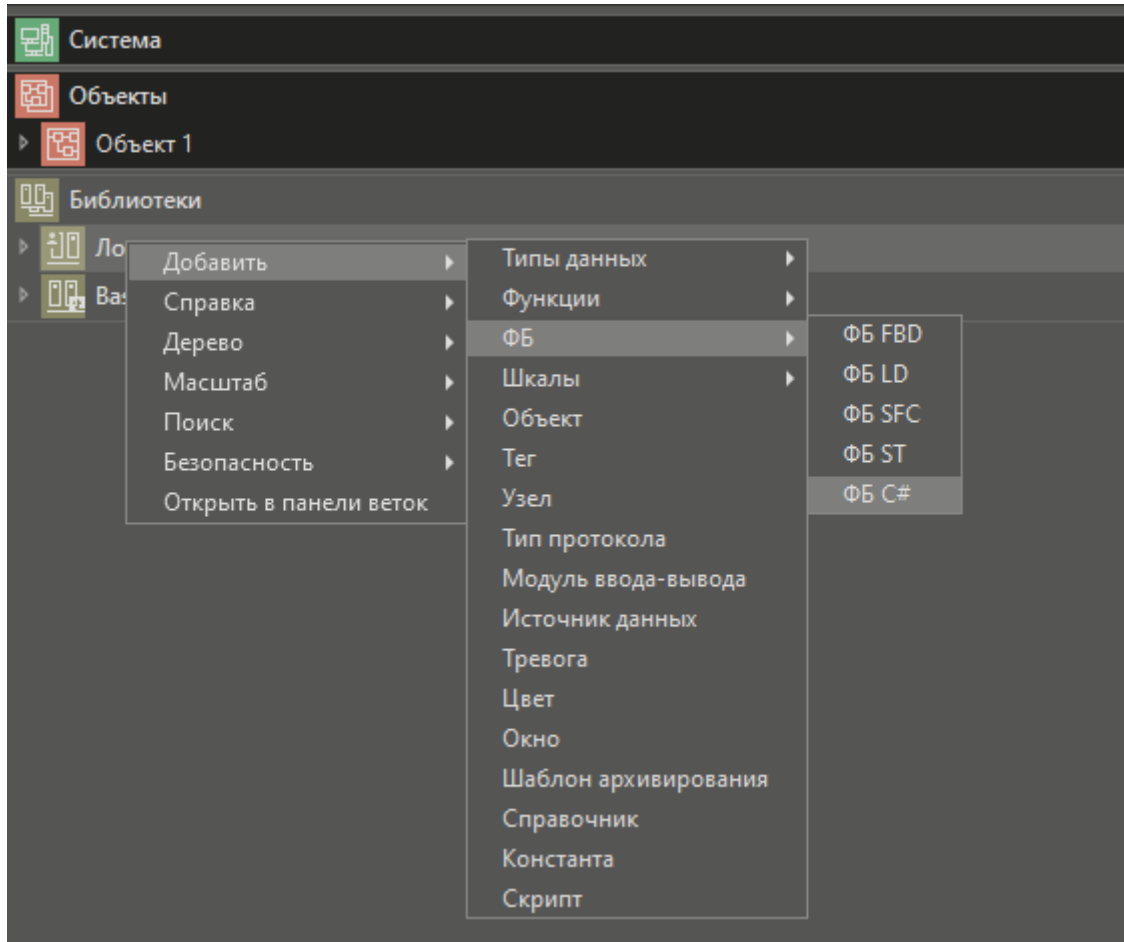
В проекте можно создать новую программу в дереве системы или в дереве объектов или новый ФБ в пользовательской библиотеке.

Для этого нужно выбрать соответствующий пункт контекстного меню того элемента, где планируется создавать новый элемент.

Контекстное меню объекта:

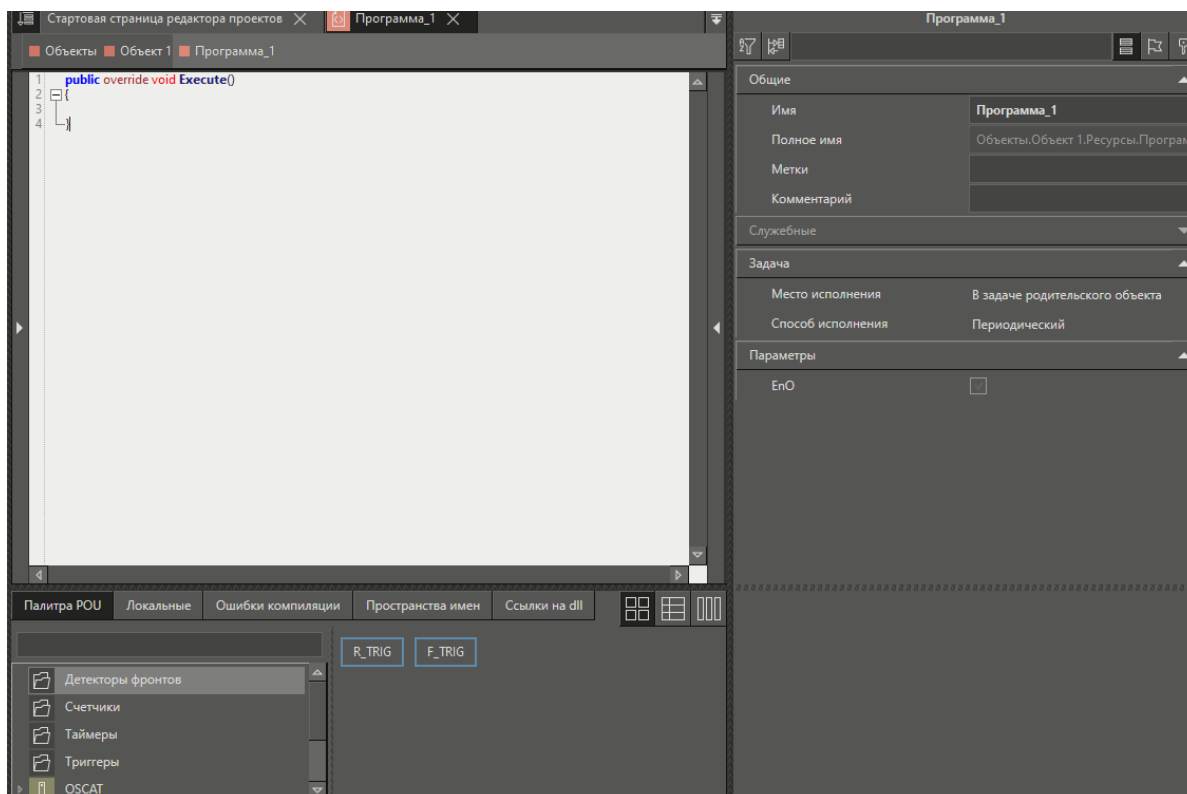


Контекстное меню библиотеки Локальная:



Добавится новый элемент. Имя элемента по умолчанию Программа 1 или ФБ 1 соответственно. Имя не будет включать название языка, т.к. в именах программ недопустимо использовать специальные символы. Имя затем можно будет сменить в панели свойств программы или ФБ. Автоматически откроется редактор.

Вид редактора:



В текстовом поле редактора создается программа.

Важно! Редактор C# регистрозависимый. В нем не используется TRUE, а только true.

По умолчанию добавлена в текст функция `public override void Execute()`, которая позволяет выполнять программы C# периодически, т.е. алгоритм внутри функции будет выполняться с заданным периодом задачи. По умолчанию программы C# назначены в задачу родительского объекта, т.е. в основную задачу. Место и способ исполнения программы можно поменять в панели свойств.

Важно! Программы и ФБ созданные в редакторе C# не будут работать в задаче экрана.

Дополнительные функции C#

Функция `public override void Init()` - будет вызываться один раз при старте, в отличии от функции `public override void Execute()`. Рассмотрим другие функции, которыми можно пользоваться внутри редактора C#:

- `LogMessage` - производит запись сообщения в лог исполнительной системы. Пример: `LogMessage("Текст сообщения");`
- `GetValue` - получает значение параметра по имени. Пример: `var value = (int)GetValue("Параметр_1");`
- `SetValue` - устанавливает значение параметра по имени. Пример: `SetValue("Параметр_1", 5);`

Связь с параметрами проекта

Связь с параметрами проекта и параметрами программы создается также как и в других редакторах, через клеммники.

При создании параметров важно следить за типами данных. В таблице ниже приведено соответствие типов параметров MasterSCADA 4D и параметров C#.

Тип данных MasterSCADA 4D	Тип данных C#
BOOL	bool
BYTE	byte
SINT	sbyte
нет	char
нет	decimal
LREAL	double
REAL	float
DINT	int
UDINT, DWORD	uint
LINT	long
ULINT	ulong
INT	short
UINT, WORD	ushort
нет	object
STRING	string

Тип данных MasterSCADA 4D	Тип данных C#
нет	dynamic
DATE	DateTime date1 = new DateTime(2015, 7, 20); // год - месяц - день
DATE_AND_TIME	DateTime date1 = new DateTime(2015, 7, 20, 18, 30, 25); // год - месяц - день - час - минута - секунда
TIME	TimeSpan
TIME_OF_DAY	TimeSpan
Перечисление	enum
Диапазон	нет
REF_TO	object
ANY	object

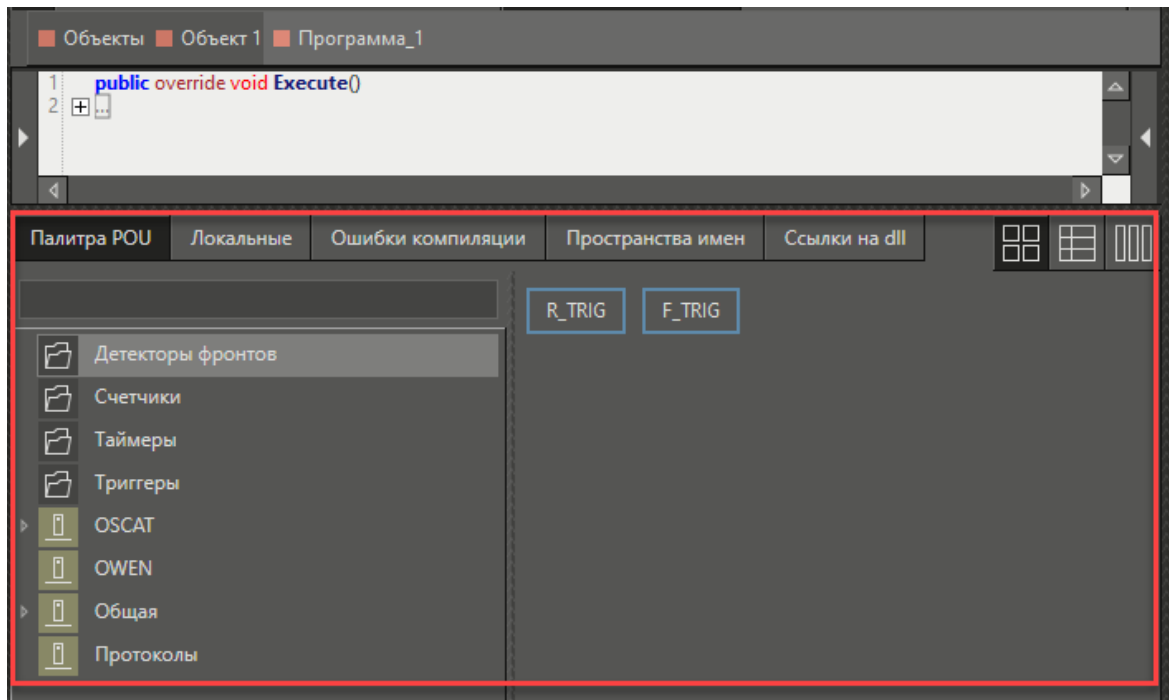
В тексте программ можно обращаться к глобальным параметрам узла, но обращение к ним будет отличаться от принятого обращения в языке ST. Подробное описание в разделе Программы C#. Использование глобальных переменных.


Порядок вызова ФБ, соответствующих стандарту МЭК 61131-3 описан в разделе Программы C#. Примеры

7.7.1. ЛЕГЕНДА РЕДАКТОРА C#

Легенда редактора C# имеет ряд отличий от других программ.

Вид легенды:



Вкладка	Описание
Палитра POU	Отображаются библиотеки функций и функциональных блоков, поставляемые по умолчанию и созданные разработчиком проекта. Элементы библиотек можно использовать в программе. Порядок вызова ФБ описан в примере Включение или отключение триггера с задержкой
Локальные	Отображаются экземпляры функциональных блоков, которые используются в программе. Порядок работы с локальными переменными и ФБ описан в примере Включение или отключение триггера с задержкой
Ошибка компиляции	Отображаются ошибки компиляции программы, которые возникли при проверке программы, если была нажата кнопка в панели инструментов редактора программ  . При компиляции проекта и обнаружении ошибок в программе, вкладка не заполняется (см. Компиляция проекта с программой C#)
Пространства имен	Указываются пространства имен, которые используются в программе. Вкладка представляет собой текстовый редактор.

Вкладка	Описание
Ссылки на dll	Указываются имена сторонних dll-файлов, которые нужно использовать в программе. Вкладка представляет собой текстовый редактор. Подробнее смотрите в разделе Программы С#.Подключение сторонних dll

7.7.2. ПРОГРАММЫ С#. СИСТЕМНЫЕ ТРЕБОВАНИЯ

Программы и функциональные блоки С# будут работать только в тех версиях исполнительной системы, которые установлены на ОС Windows. Причем операционная система должна поддерживать .NET 5

В приведенной ниже таблице содержится список поддерживаемых сейчас выпусков .NET и версий Windows, в которых они поддерживаются:

Операционная система	.NET Core 2.1	.NET Core 3.1	.NET 5
Windows 10 или Windows Server версии 20H2	✓	✓	✓
Windows 10 или Windows Server версии 2004	✓	✓	✓
Windows 10 или Windows Server версии 1909	✓	✓	✓
Windows 10 или Windows Server версии 1903	✓	✓	✓
Windows 10, версия 1809	✓	✓	✓
Windows 10, версия 1803	✓	✓	✓
Windows 10, версия 1709	✓	✓	✓
Windows 10 (версия 1607)	✓	✓	✓
Windows 8.1	✓	✓	✓
Windows 7 с пакетом обновления 1 (SP1), ESU	✓	✓	✓
Windows Server 2019 Windows Server 2016 Windows Server 2012 R2	✓	✓	✓
Windows Server Core 2012 R2	✓	✓	✓
Nano Server, версия 1809 и выше	✓	✓	✓
Nano Server, версия 1803	✓	✓	✗

При использовании версий Windows 7, Vista, 8.1, Server 2008 R2, Server 2012, R2 возможно потребуется установить дополнительные компоненты, а также последние обновления.

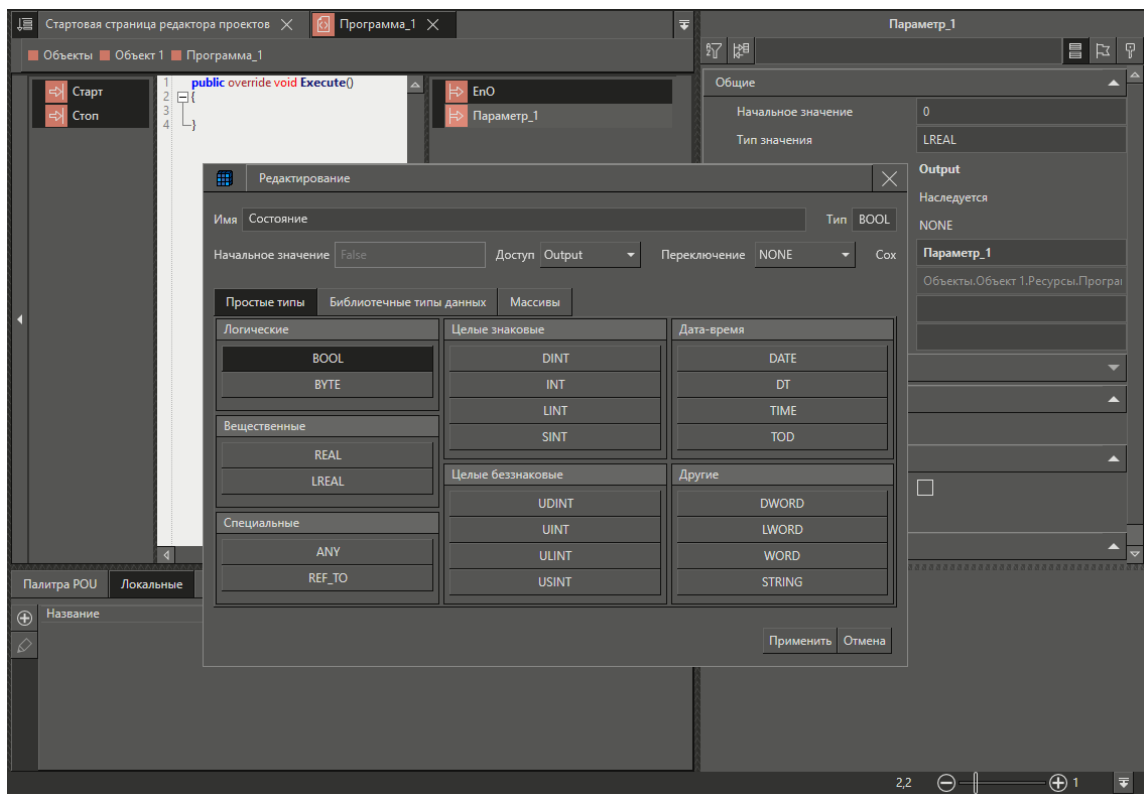
7.7.3. ПРОГРАММЫ C#. ПРИМЕРЫ

Рассмотрим основные особенности работы с редактором C# в MasterSCADA 4D на простых примерах.

Простейший триггер

В дереве объектов создадим программу C#. Добавим в нее три параметра типа BOOL:

- Старт и Стоп на клеммник входов
- Состояние на клеммник выходов



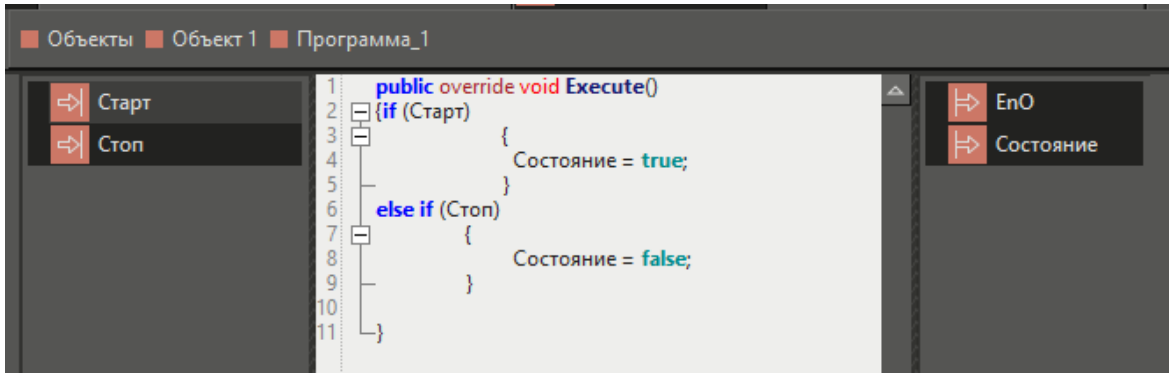
В теле функции, которая добавляется автоматически при создании программы пишем код:

```
{if (Старт)
```

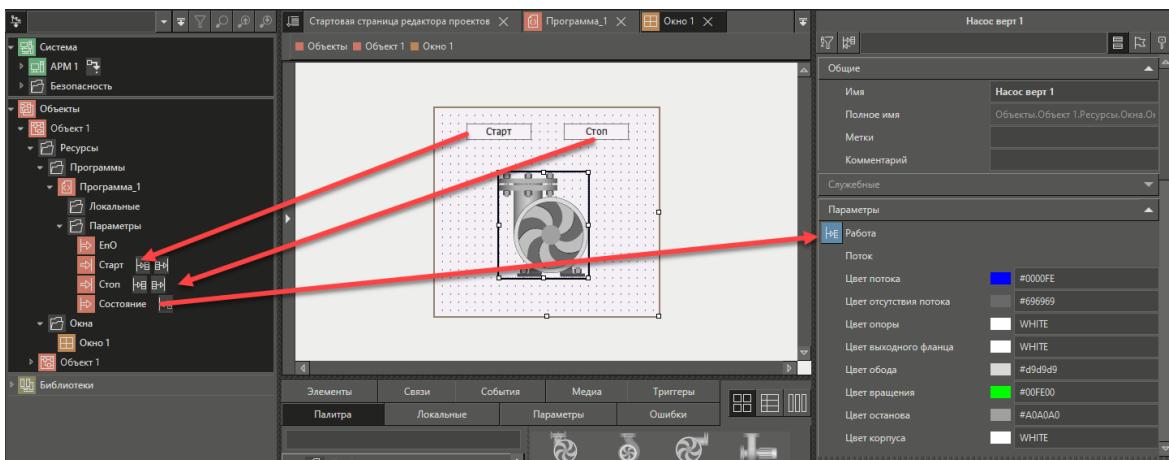
```
{  
    Состояние = true;  
}
```

```
else if (Стоп)
```

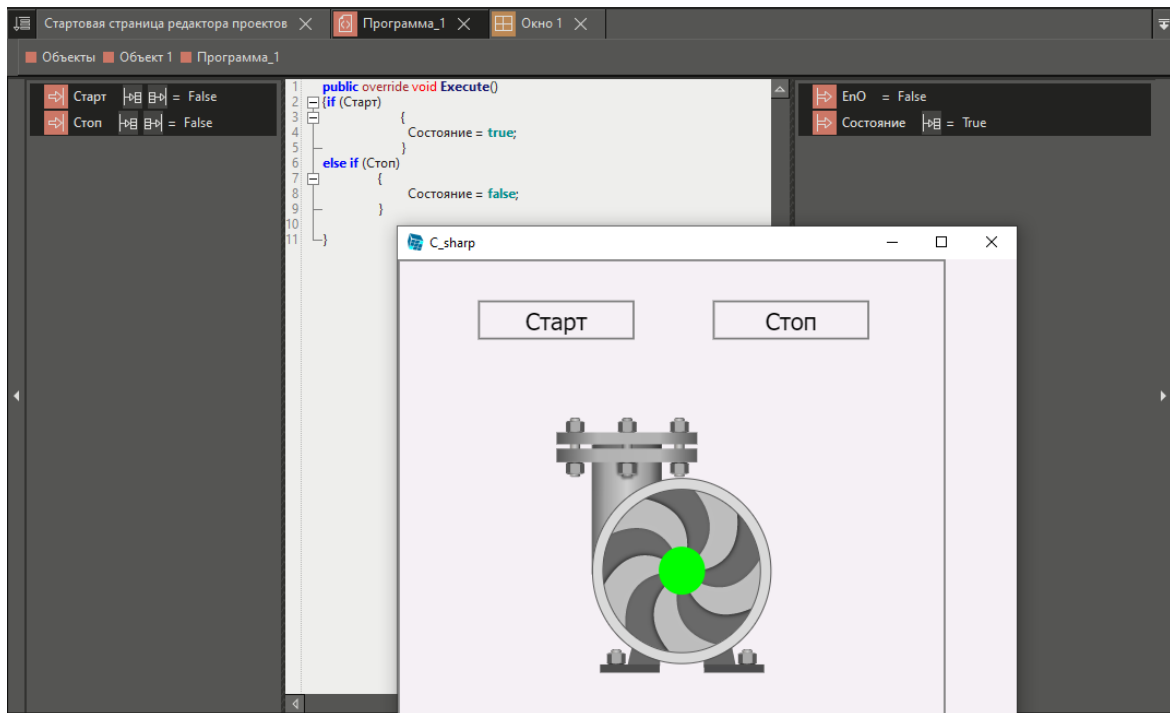
```
{  
    Состояние = false;  
}
```

Свяжем параметры программы с графическими элементами в окне:



В режиме исполнения получим результат:



Поменяем имя программы в ее панели свойств. Назовем ее Триггер.

Включение или отключение триггера с задержкой

Рассмотрим пример, когда переключение триггера должно происходить с задержкой, которая может быть изменена в режиме исполнения. В данном примере рассмотрим способ вызова ФБ, соответствующих стандарту МЭК 61131-3.

Продолжим работу с программой описанной в предыдущем примере.

Добавим во входной клеммник программы параметр Задержка типа Time.

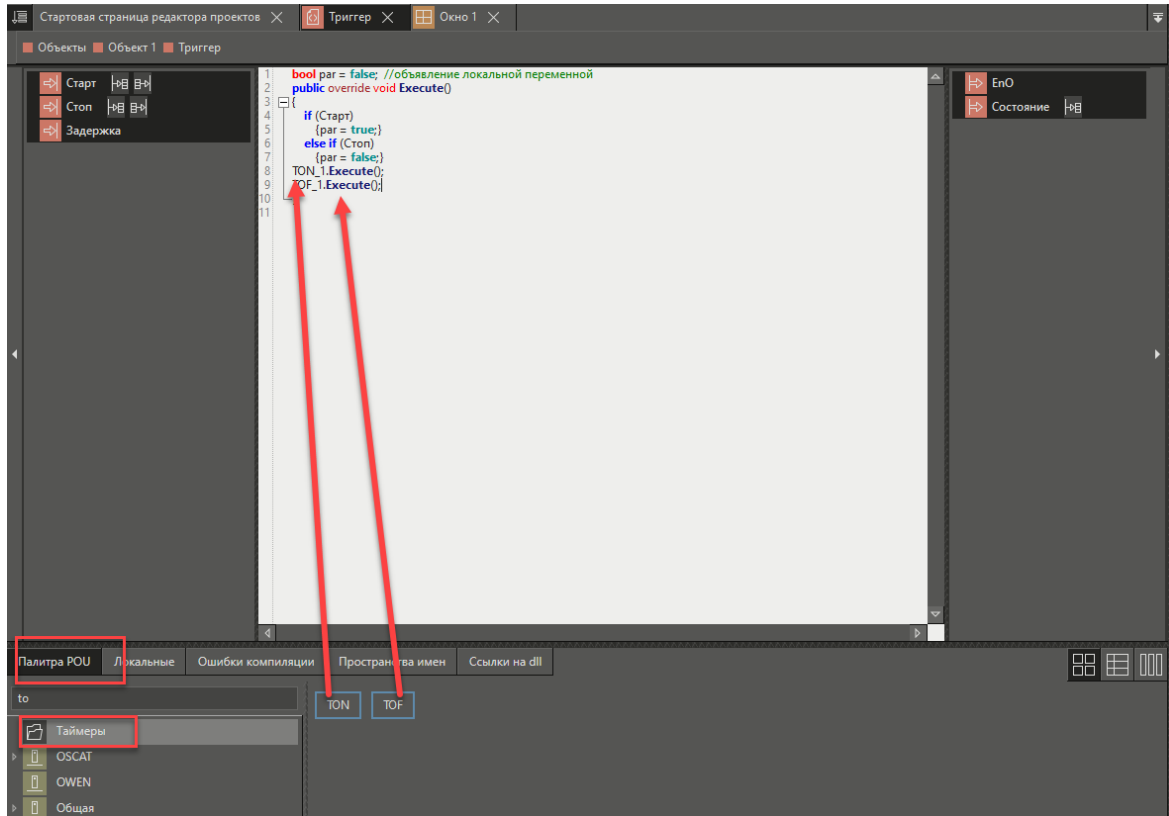
Для решения задачи нам потребуется ввести локальную переменную `rag` типа `BOOL`. В отличие от других редакторов, где локальная переменная объявляется на соответствующей вкладке легенды, в программе `C#`, они объявляются непосредственно в теле программы:

```
1  bool rag=false; //объявление локальной переменной
2  public override void Execute()
3  {
4      if (Старт)
5          {
6              Состояние = true;
7          }
8      else if (Стоп)
9          {
10             Состояние = false;
11         }
12 }
```

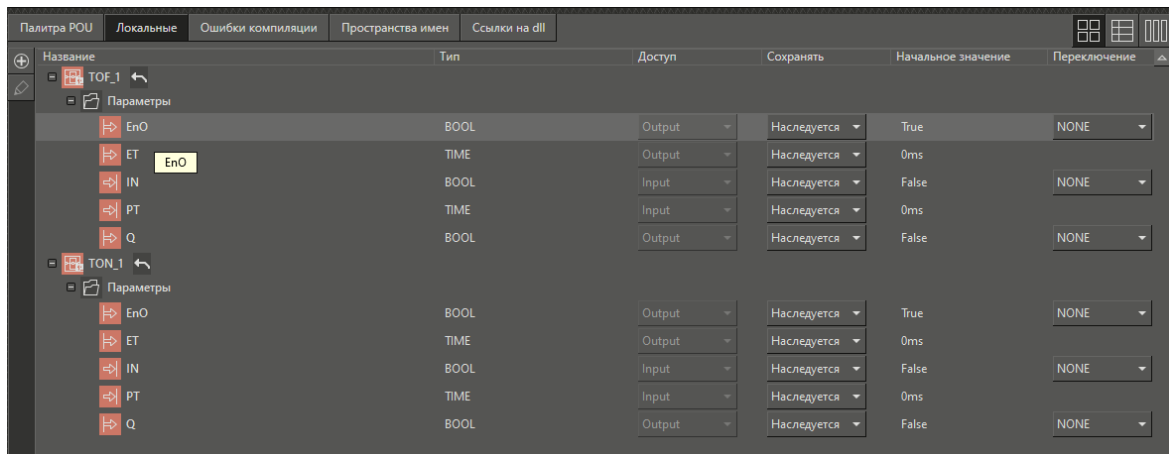
Важно! Локальные переменные объявляются вне функции `public override void Execute()`. В противном случае инициализации переменной будет происходить на каждом цикле программы.

Для решения поставленной задачи воспользуемся таймерами `TON`, `TOF`

Перетащим их в поле программы.



В тексте программы получим функцию обеспечивающую вызов функционального блока: имя_ФБ/Execute(). После этого ФБ появятся на вкладке Локальные:



Чтобы обратиться к тому или иному входу или выходы ФБ в программе C# нужно написать: Имя_ФБ.Имя_параметра

Далее изменим текст программы таким образом:

```

bool par = false; //объявление локальной переменной

public override void Execute()

{

```

```
if (Старт)
    {par = true;}

else if (Стоп)
    {par = false;}

TON_1.PT = Задержка; //присвоение входной переменной ФБ параметру
программы

TON_1.IN = par;

TON_1.Execute(); //выполнение ФБ

TOF_1.PT = Задержка;

TOF_1.IN = par;

TOF_1.Execute();

if (par) { Состояние = TON_1.Q;} //присвоение параметру программы выходной
переменной ФБ

else {Состояние = TOF_1.Q;}

}
```

7.7.4. ПРОГРАММЫ C#. РАБОТА СО СЛОЖНЫМИ ТИПАМИ ДАННЫХ

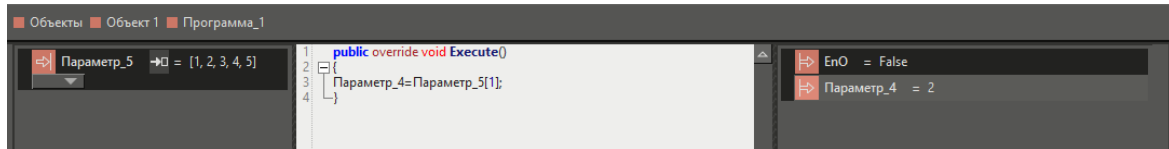
В программах C# можно работать со сложными типами данных проекта - перечислениями, структурами, массивами, массивами структур.

Чтобы обратиться к полю структуры нужно использовать запись `Имя_параметра.Имя_Поля`:

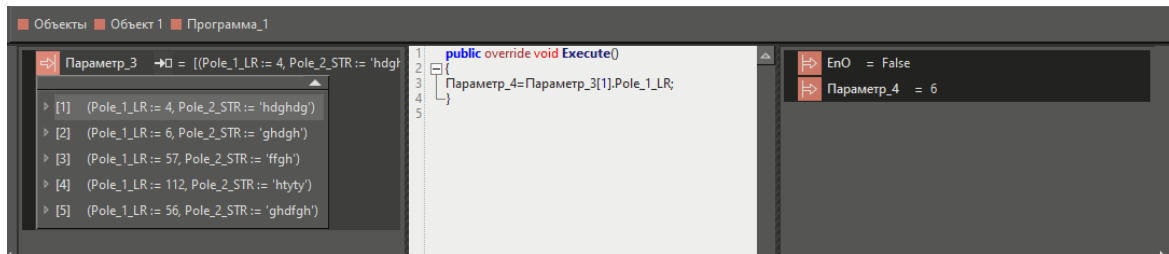


Важно! При создании типа данных Структура нужно в названиях полей использовать только латинские буквы и цифры.

В программе C# можно использовать только одномерные массивы. Для обращения к конкретному элементу нужно использовать запись: Имя_параметра[Номер элемента]. Нумерация начинается с 0:



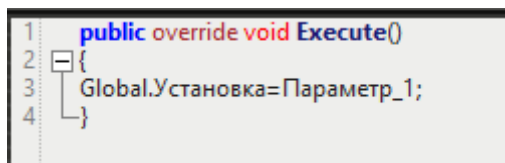
Чтобы обратиться к определенному полю массива структур необходимо использовать запись: Имя_параметра[Номер элемента].Имя_поля. Нумерация начинается с 0:



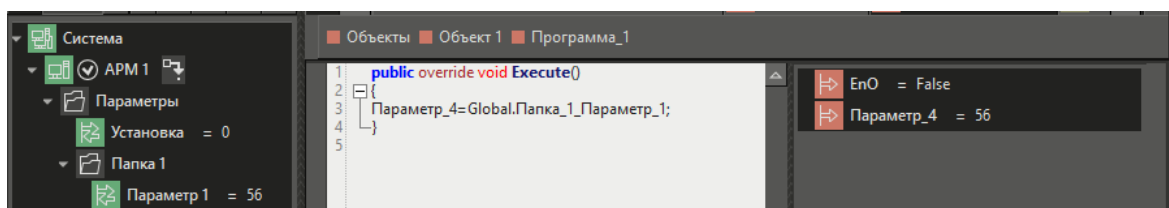
7.7.5. ПРОГРАММЫ C#. ИСПОЛЬЗОВАНИЕ ГЛОБАЛЬНЫХ ПЕРЕМЕННЫХ

В программах C# можно использовать глобальные параметры.

Для того, чтобы обратиться к глобальной переменной нужно добавить перед именем параметра Global.:



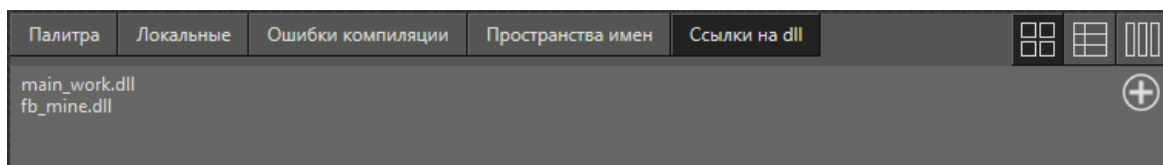
Если глобальный параметр находится в папке, то имя параметра будет содержать имя папки и имя параметра, все пробелы в названиях нужно заменить символом _ например:




7.7.6. ПРОГРАММЫ C#.ПОДКЛЮЧЕНИЕ СТОРОННИХ DLL

MasterSCADA 4D позволяет использовать сторонние dll в программах C#.

Добавить новые пользовательские библиотеки можно на вкладке Ссылки на dll:



Для добавления новой пользовательской библиотеки нужно нажать на кнопку , то откроется диалоговое окно Windows, в котором можно выбрать пользовательскую произвольную библиотеку, при этом она будет скопирована в папку проекта *[имя проекта]\dll*. При компиляции программы - автоматически загрузятся в виде assembly, то есть из программы можно ссылаться на классы, описанные в них (пользовательских dll).

7.7.7. КОМПИЛЯЦИЯ ПРОЕКТА С ПРОГРАММОЙ C#

В папке скомпилированного проекта появится дополнительная директория: *"c:\Users\[имя пользователя]\AppData\Roaming\MPSSoft\MasterSCADA4D[номер версии]\Debug_[имя проекта]\[имя узла]\netcore"*. В этой директории будет находиться проект программ C#, которые добавлены в задачи выбранного узла.

В директории *\Debug_[имя проекта]\APM 1\cfg\netcore* - будет содержаться скомпилированная программа, которая непосредственно будет исполняться в среде исполнения.

При компиляции проекта, ошибки, которые будут возникать будут отображаться непосредственно в окне Компиляция и в окне Ошибки проекта. При этом из-за особенностей реализации поддержки языка C# на вкладке Легенды программы Ошибки компиляции эта информация не попадет.

Вкладка Ошибки компиляции заполняется только при проверке одной программы, если

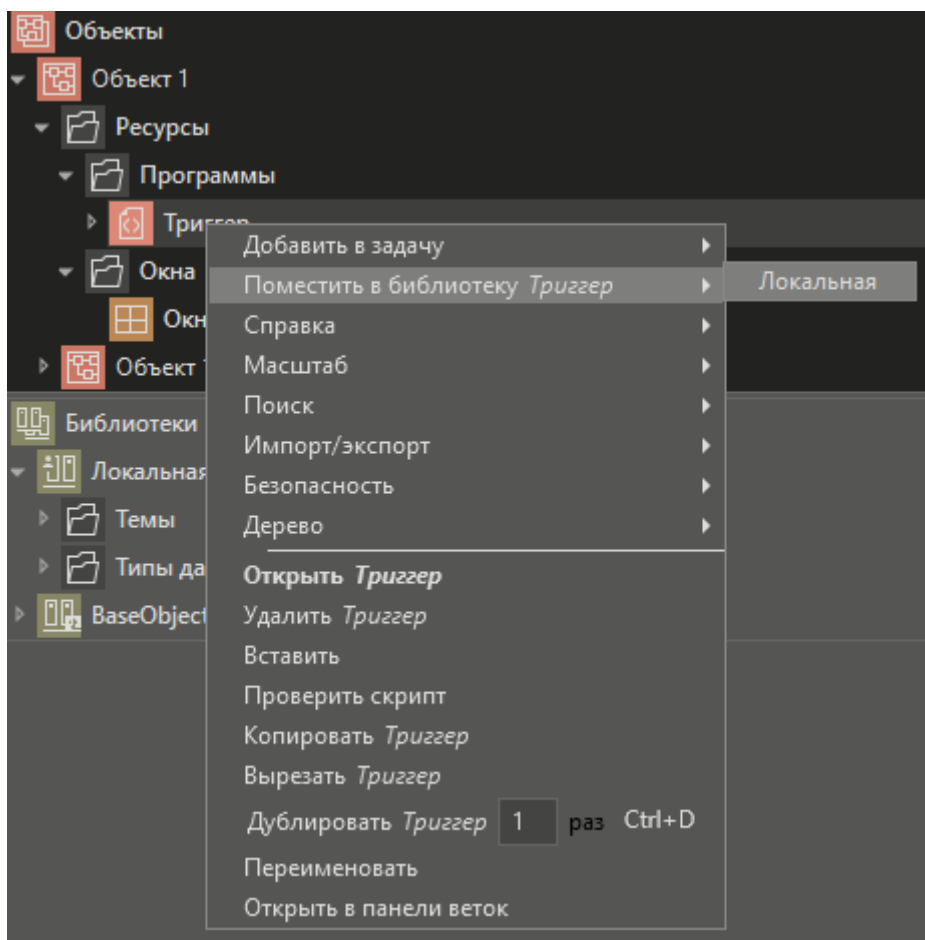
была нажата кнопка в панели инструментов редактора программ 

7.7.8. СОЗДАНИЕ ФБ C#

Функциональные блоки для различных языков как правильно создаются по пути от экземпляра к типу, т.к. сначала программу отлаживают в среде исполнения, и после чего сохраняют в библиотеку.

Рассмотрим на примере.

Выделим в дереве объектов программу, созданную в разделе Программы C#. Примеры. В контекстном меню выберем пункт Поместить в библиотеку:



После чего программа появится в выбранной библиотеки и станет типом ФБ, а программа в дереве объектов станет экземпляром.

Далее этот ФБ из библиотеки можно использовать в других частях текущего проекта, а если была выбрана библиотека, созданная пользователем и сохранена, то ФБ будет доступен и в других проектах. Его можно использовать как другие ФБ.

Использование ФБ С# в других программах С# и ФБ С#

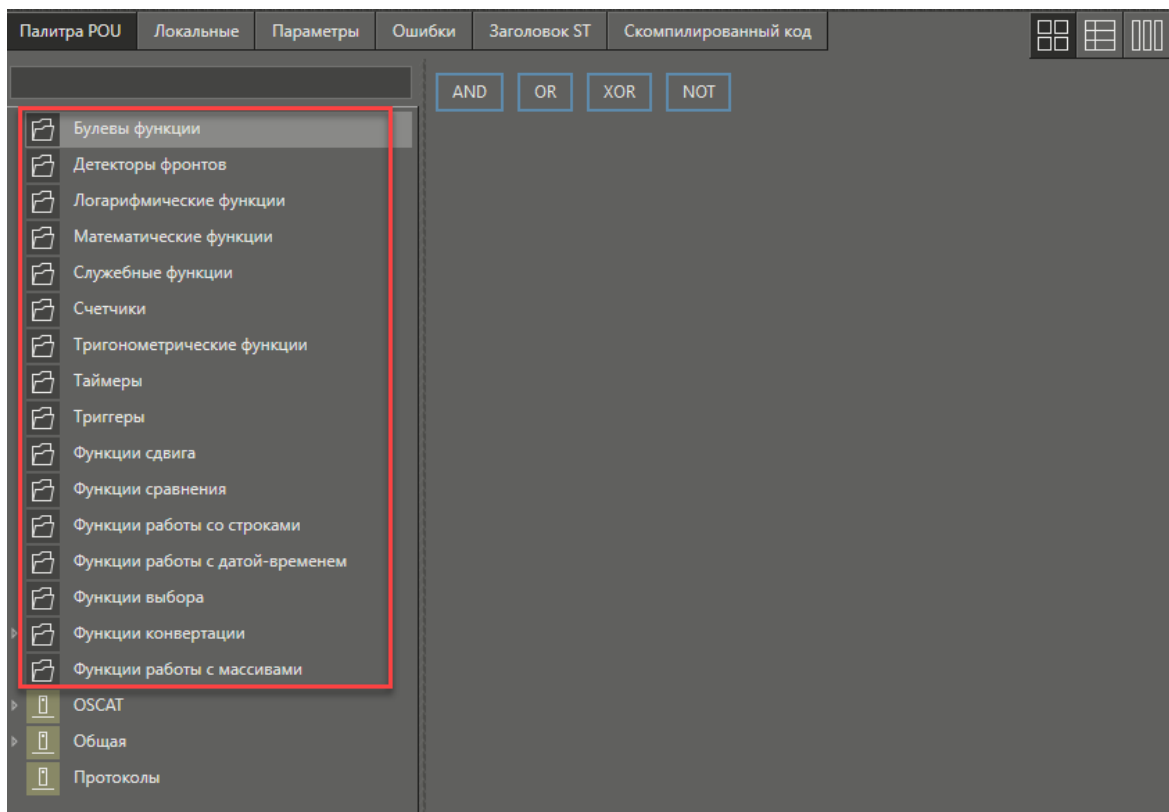
Создавать в других программах С# и ФБ С# экземпляры данных ФБ с помощью записи `new имя_библиотеки.имя_папки.имя_ФБ`. Аналогично можно использовать статические методы из этих ФБ

7.8. ФУНКЦИИ И ФБ БИБЛИОТЕКИ СТАНДАРТНАЯ

В стандартной библиотеки есть две подгруппы:

Название	Описание
Стандартные функции	Содержит стандартные функции для разработки схем на языке FBD и программ ST, согласно стандарту МЭК 61131-3.
Стандартные ФБ	Содержит стандартные функциональные блоки (ФБ) для разработки схем на языке FBD и программ ST, согласно стандарту МЭК 61131-3.

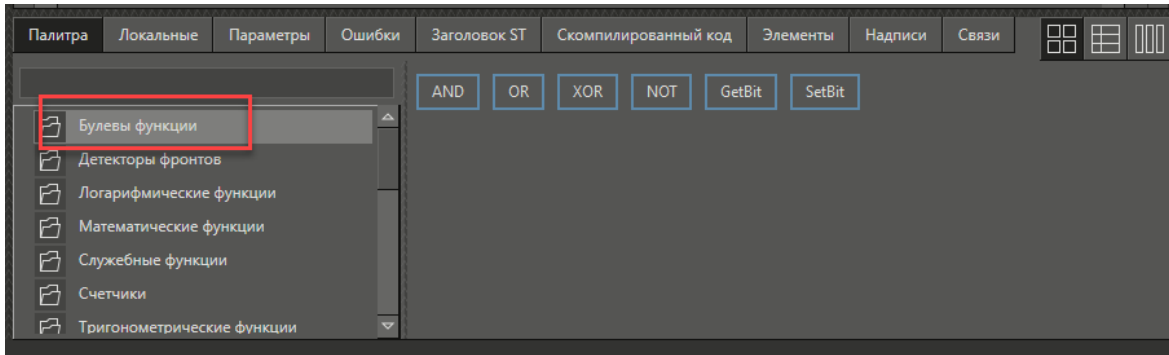
В палитре языков программирования элементы из этих библиотек находятся в верхней части:



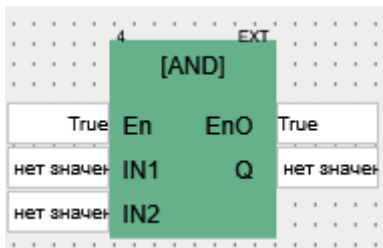
7.8.1. БИБЛИОТЕКА СТАНДАРТНЫЕ ФУНКЦИИ

7.8.1.1. БУЛЕВЫ ФУНКЦИИ

Вид в палитре



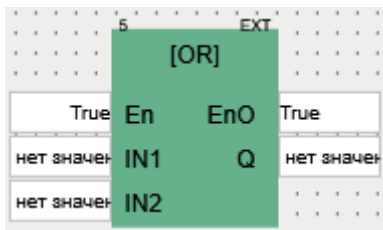
AND, побитовое И



$$Q := IN1 \& IN2$$

Например, $6 \& 10 = 2$.

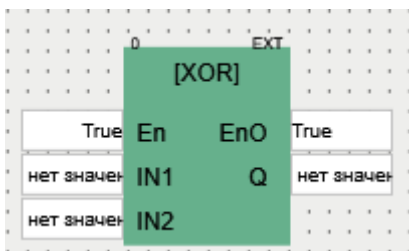
OR, побитовое ИЛИ



$$Q := IN1 \text{ or } IN2$$

Например, $6 \text{ or } 3 = 7$.

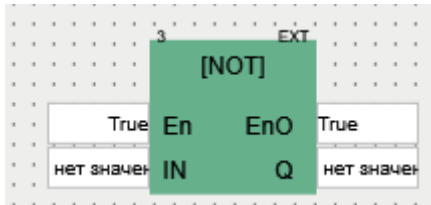
XOR, побитовое исключающее ИЛИ



$$Q := IN1 \text{ xor } IN2$$

Например, $6 \text{ xor } 3 = 5$.

NOT, побитовое НЕ



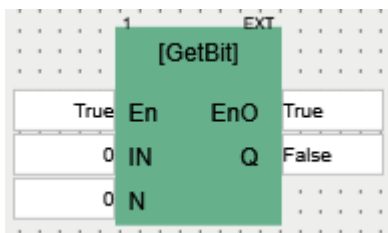
$Q := \text{not } IN$

Например (см. рис.), если тип данных in1 – UINT и in1=0, то out=16#FFFF (тип данных out – UINT, UDINT или ULINT).

7.8.1.1.1. GETBIT

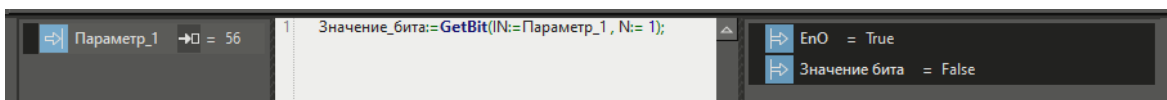
Функция GetBit служит для получения бита из целого числа. Находится в библиотеке Стандартные функции.Булевы функции.

Вид функции в редакторе FBD:



На вход IN передается число, целого типа, из которого нужно выделить бит, а на вход N - номер бита, нумерация начинается с нуля. 0 -младший бит. На выход Q передается значение бита. Выход имеет тип BOOL.

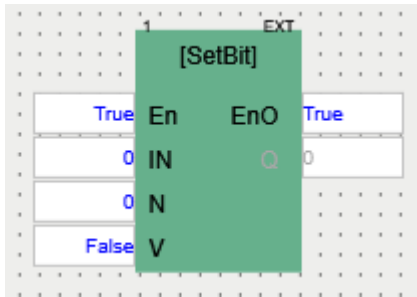
Пример:



7.8.1.1.2. SETBIT

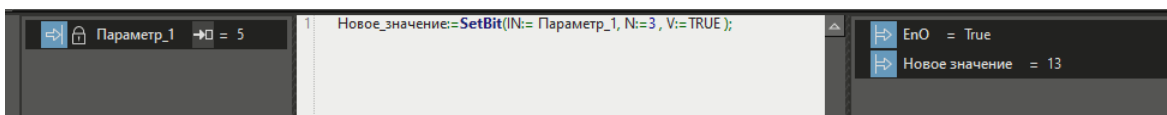
Функция SetBit служит для изменения значения бита в целом числе. Находится в библиотеке Стандартные функции.Булевы функции.

Вид функции в редакторе FBD:



На вход IN передается число, целого типа в котором нужно изменить бит, а на вход N - номер изменяемого бита, нумерация начинается с нуля. 0 -младший бит, на вход V, типа BOOL подается новое значение бита. На выход Q передается новое целое число.. Выход имеет тип LINT.

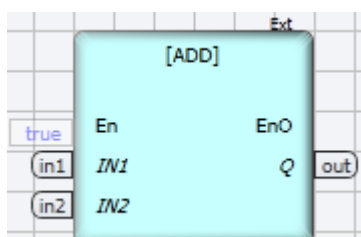
Пример:



7.8.1.2. МАТЕМАТИЧЕСКИЕ ФУНКЦИИ

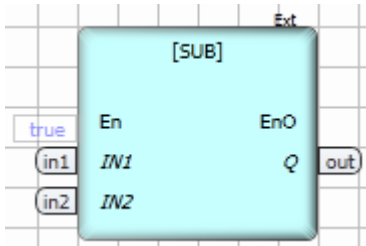
- ADD, сложение
- SUB, вычитание
- MUL, умножение
- DIV, деление
- MOD, остаток от деления
- EXPT, возведение в степень
- MOVE, передача значения
- ABS, абсолютная величина
- TRUNC, целая часть к нулю
- SQRT, квадратный корень

ADD, сложение



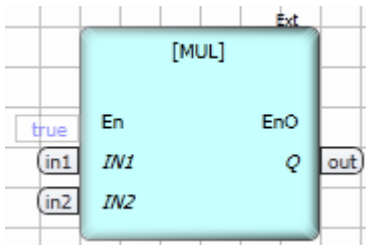
$Q := IN1 + IN2$

SUB, вычитание



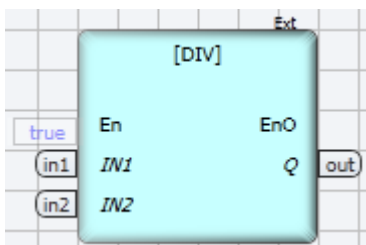
$Q := IN1 - IN2$

MUL, умножение



$Q := IN1 * IN2$

DIV, деление

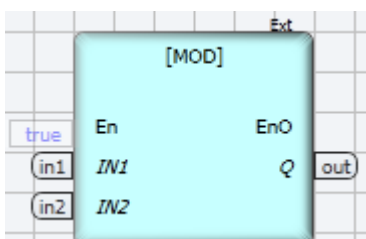


$Q := IN1 / IN2$

Например (см. рис.), если $in1=3$, $in2=2$ и тип данных out – REAL:

- ▶ $out=1$, если оба входа имеют целочисленный тип данных;
- ▶ $out=1.5$, если хотя бы один вход имеет вещественный тип данных.

MOD, остаток от деления

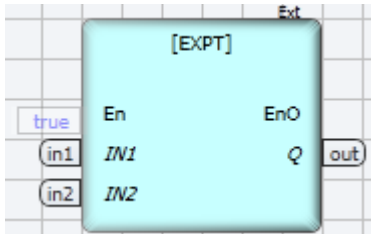


Функция возвращает остаток от деления целых чисел (IN1 на IN2):

$Q := \text{mod}(\text{IN1}, \text{IN2})$

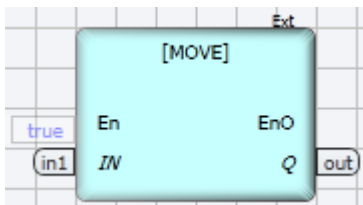
Например, $\text{mod}(8, 3) = 2$.

EXPT, возведение в степень



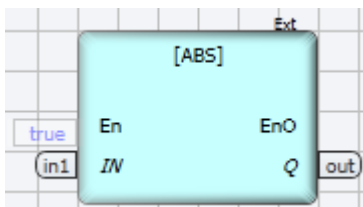
$Q := \text{IN1} ** \text{IN2}$

MOVE, передача значения



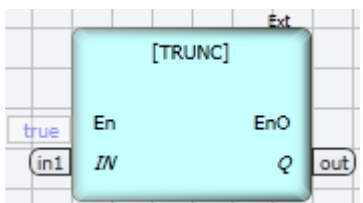
$Q := \text{IN}$

ABS, абсолютная величина



$Q := \text{abs}(\text{IN})$

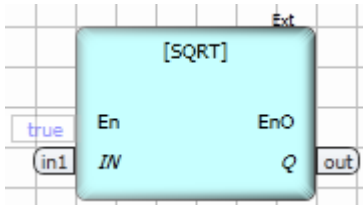
TRUNC, целая часть к нулю



$Q := \text{trunc}(\text{IN})$

Например, $\text{trunc}(5.9) = \text{trunc}(5.1) = 5$; $\text{trunc}(-5.9) = \text{trunc}(-5.1) = -5$.

SQRT, квадратный корень

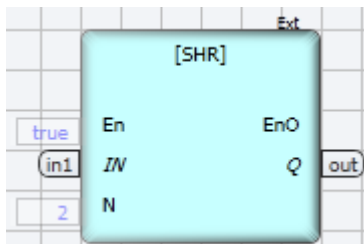


$Q := \text{sqrt}(IN)$

7.8.1.3. ФУНКЦИИ СДВИГА

- SHR, сдвиг вправо
- SHL, сдвиг влево
- ROR, циклический сдвиг вправо
- ROL, циклический сдвиг влево

SHR, сдвиг вправо

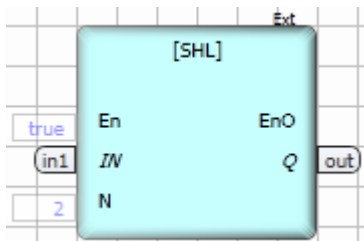


Сдвиг IN вправо на N разрядов, слева число дополняется нулевыми разрядами:

$Q := \text{shr}(IN, N)$

Например, $\text{shr}(8, 2) = 2$.

SHL, сдвиг влево

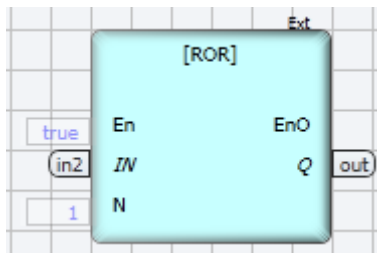


Сдвиг IN влево на N разрядов, справа число дополняется нулевыми разрядами:

$Q := \text{shl}(IN, N)$

Например, $\text{shl}(2, 2) = 8$.

ROR, циклический сдвиг вправо

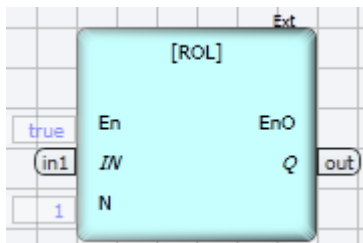


Сдвиг IN вправо на N разрядов. Слева число дополняется разрядами, которые при сдвиге "выбывают" справа:

$$Q := \text{ror}(IN, N)$$

Например (см. рис.), если тип данных in2 – USINT и in2=1, то out=128.

ROL, циклический сдвиг влево



Сдвиг IN влево на N разрядов. Справа число дополняется разрядами, которые при сдвиге "выбывают" слева:

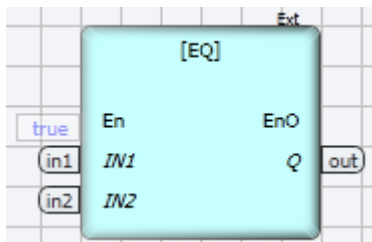
$$Q := \text{rol}(IN, N)$$

Например (см. рис.), если тип данных in1 – UINT и in1=16#8000, то out=1.

7.8.1.4. ФУНКЦИИ СРАВНЕНИЯ

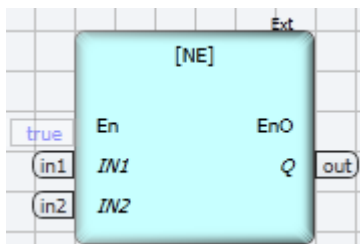
- EQ, равенство
- NE, неравенство
- LT, меньше
- LE, меньше или равно
- GT, больше
- GE, больше или равно

EQ, равенство



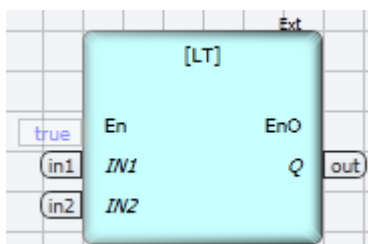
Если $in1=in2$, то $out=1$, в противном случае $out=0$.

NE, неравенство



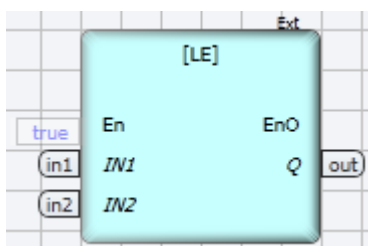
Если $in1 \neq in2$, то $out=1$, в противном случае $out=0$.

LT, меньше



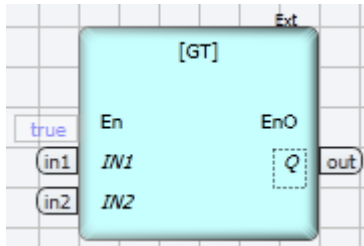
Если $in1 < in2$, то $out=1$, в противном случае $out=0$.

LE, меньше или равно



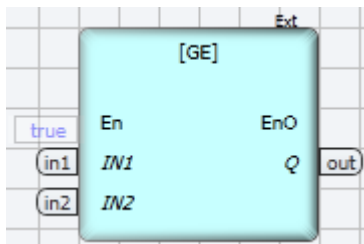
Если $in1 \leq in2$, то $out=1$, в противном случае $out=0$.

GT, больше



Если $in1 > in2$, то $out=1$, в противном случае $out=0$.

GE, больше или равно



Если $in1 \geq in2$, то $out=1$, в противном случае $out=0$.

7.8.1.5. ТРИГОНОМЕТРИЧЕСКИЕ ФУНКЦИИ

- SIN, синус
- COS, косинус
- TAN, тангенс
- ASIN, арксинус
- ACOS, арккосинус
- ATAN, арктангенс

Значение входа IN прямых тригонометрических функций задается как угол в радианах.

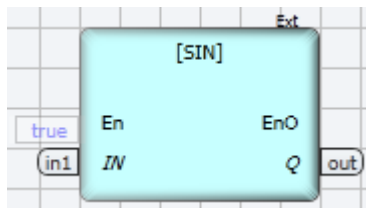
Обратные тригонометрические функции возвращают главное значение соответствующих функций в радианах:

$$-\frac{\pi}{2} \leq \arcsin x \leq \frac{\pi}{2}$$

$$0 \leq \arccos x \leq \pi$$

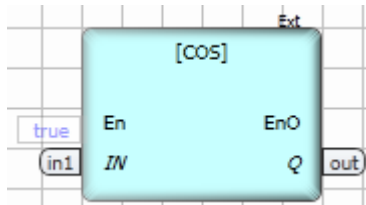
$$-\frac{\pi}{2} \leq \arctg x \leq \frac{\pi}{2}$$

SIN, синус



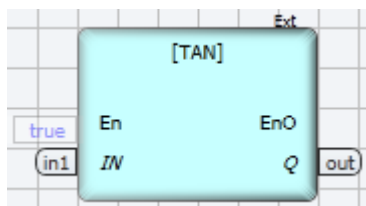
$Q := \sin(IN)$

COS, косинус



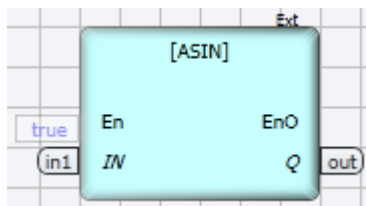
$Q := \cos(IN)$

TAN, тангенс



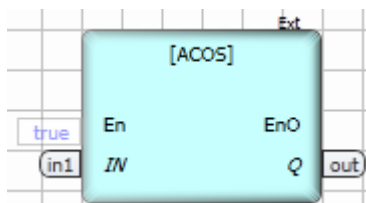
$Q := \text{tg}(IN)$

ASIN, арксинус



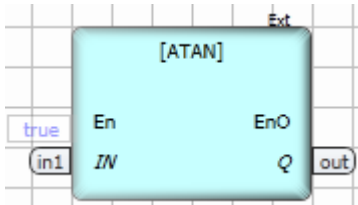
$Q := \text{asin}(IN)$

ACOS, арккосинус



$Q := \text{acos}(IN)$

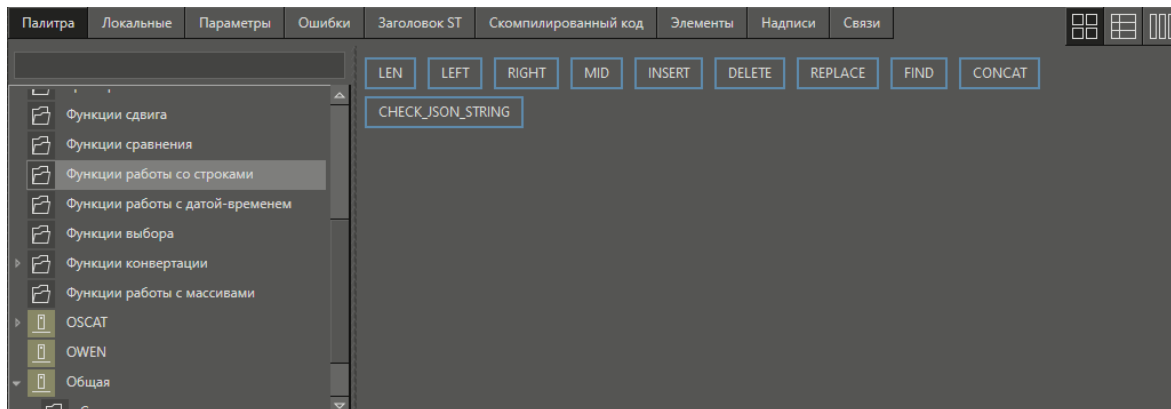
ATAN, арктангенс



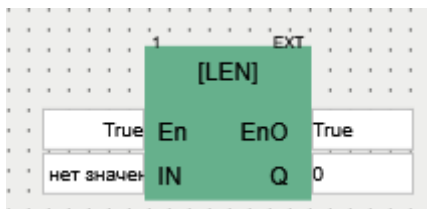
Q := atan(IN)

7.8.1.6. ФУНКЦИИ РАБОТЫ СО СТРОКАМИ

Вид категории в палитре языков программирования:



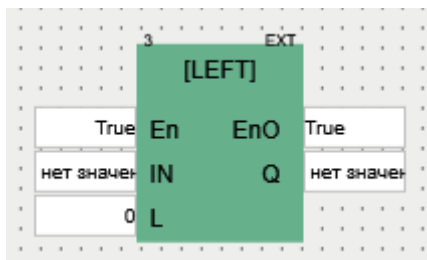
LEN, длина строки



Функция возвращает число символов в строке:

Q := len(IN)

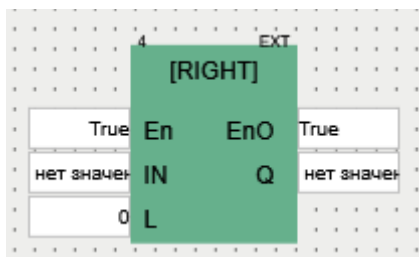
LEFT, первые символы



Функция возвращает L первых символов строки IN:

Q := left(IN, L)

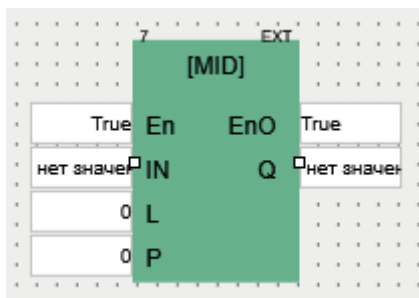
RIGHT, последние символы



Функция возвращает L последних символов строки IN:

$Q := \text{right}(\text{IN}, L)$

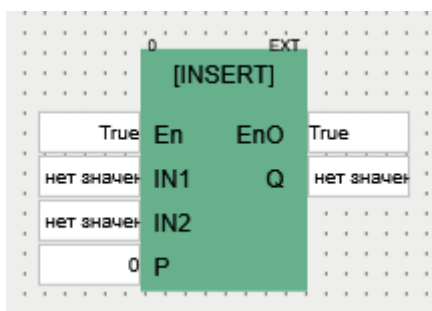
MID, подстрока



Функция возвращает L символов из строки IN, начиная с P-го символа:

$Q := \text{mid}(\text{IN}, L, P)$

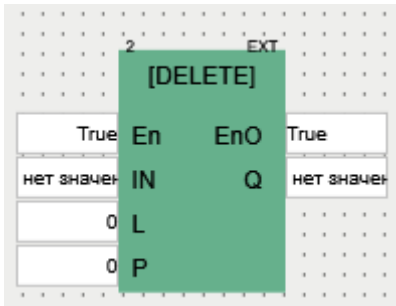
INSERT, вставка строки



Вставка строки IN2 в строку IN1 после P-го символа:

$Q := \text{insert}(\text{IN1}, \text{IN2}, P)$

DELETE, удаление подстроки

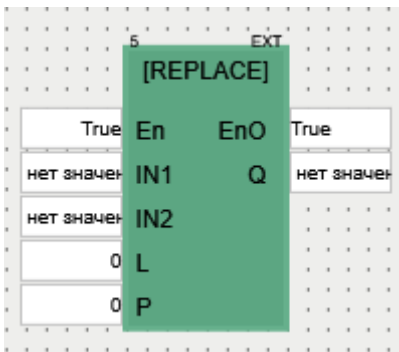


Удаление L символов из строки IN, начиная с P-го символа:

$Q := \text{delete}(\text{IN}, L, P)$

Например (см. рис.), если $mSTR1 = \text{"ABCDEF"}$, $in1 = 4$ и $in2 = 2$, то $mSTR = \text{"AF"}$.

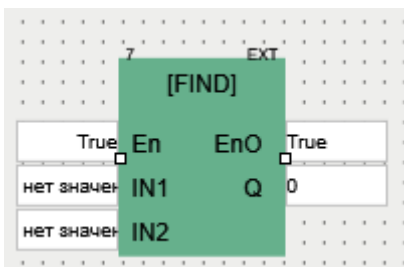
REPLACE, замена подстроки



Замена в строке IN1 L символов, начиная с P-го символа, строкой IN2:

$Q := \text{replace}(\text{IN1}, \text{IN2}, L, P)$

FIND, поиск подстроки

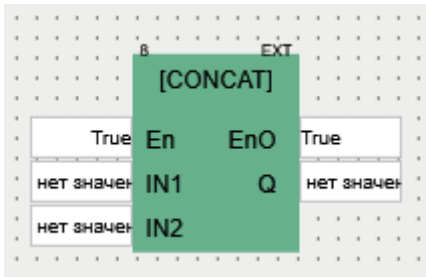


Функция возвращает номер символа, начиная с которого в строке IN1 располагается первое вхождение подстроки IN2:

$Q := \text{find}(\text{IN1}, \text{IN2})$

Например (см. рис.), если $mSTR1 = \text{"Product of MPS SOFT Company"}$, $mSTR2 = \text{"MPS SOFT"}$, то $out = 12$.

CONCAT, конкатенация строк

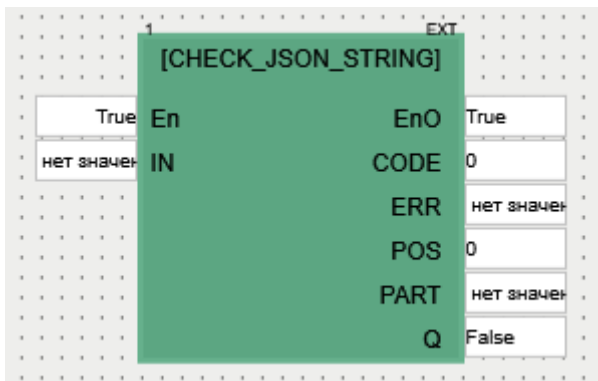


$Q := \text{concat}(\text{IN1}, \text{IN2})$

Например (см. рис.), если $mSTR1 = \text{"MPS SOFT"}$, $mSTR2 = \text{" Company"}$, то $mSTR = \text{"MPS SOFT Company"}$.

CHECK_JSON_STRING

Функция служит для проверки Json-строки на наличие ошибок:



Название	Тип	Назначение
Входы		
In	STRING	Задается строка, которую нужно проверить.
Выходы		
CODE	INT	Код ошибки. См таблицу Коды ошибок ниже
ERR	STRING	Текстовое описание кода ошибки. См таблицу Коды ошибок ниже
POS	DINT	Выдается позиция в строке с ошибкой

Название	Тип	Назначение
PART	STRING	Часть Json-строки, которая содержит ошибку.
Q	BOOL	Показывает есть ли ошибки в строке. Если значение TRUE, то ошибок нет.

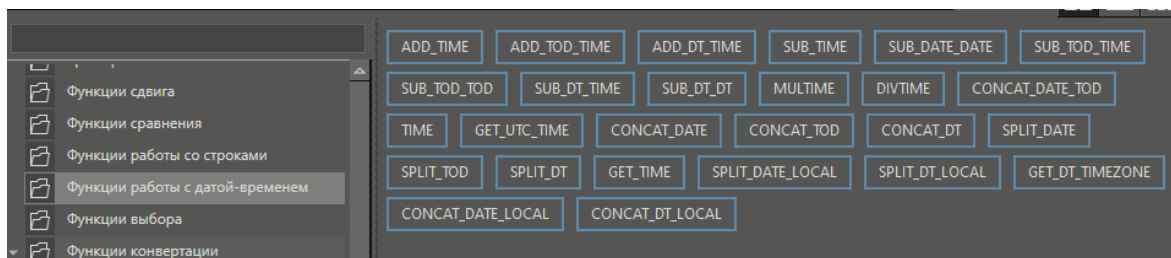
Коды ошибок:

Номер	Текст
0	No error.
1	The document is empty.
2	The document root must not be followed by other values.
3	Invalid value.
4	Missing a name for object member.
5	Missing a colon after a name of object member.
6	Missing a comma or '}' after an object member.
7	Missing a comma or ']' after an array element.
8	Incorrect hex digit after \\u escape in string.
9	The surrogate pair in string is invalid.
10	Invalid escape character in string.
11	Missing a closing quotation mark in string.
12	Invalid encoding in string.

Номер	Текст
13	Number too big to be stored in double.
14	Miss fraction part in number.
15	Miss exponent in number.
16	Terminate parsing due to Handler error.
17	Unspecific syntax error.
18	Unknown error.

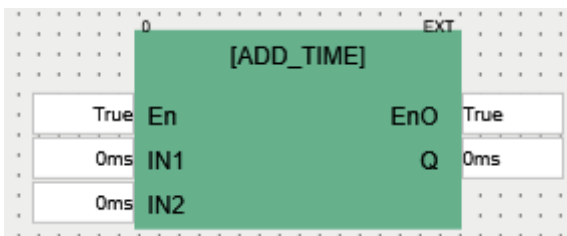
7.8.1.7. ФУНКЦИИ РАБОТЫ С ДАТОЙ-ВРЕМЕНЕМ

Вид категории в палитре языков программирования:



ADD_TIME

Сложение переменных типа TIME



Входы/Выходы	Тип	Описание
IN1	TIME	Вход функции, 1 слагаемое.
IN2	TIME	Вход функции, 2 слагаемое

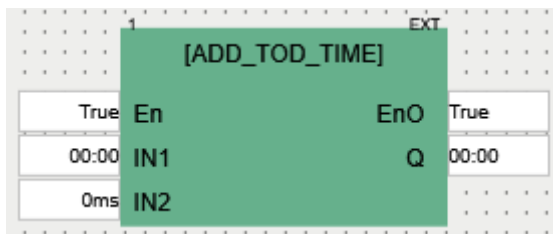
Входы/Выходы	Тип	Описание
Q	TIME	Выход функции, сумма времен поданных на входы функции

Пример в ST

```
Q := add_time(IN1, IN2)
```

ADD_TOD_TIME

Сложение переменных типа TOD и TIME



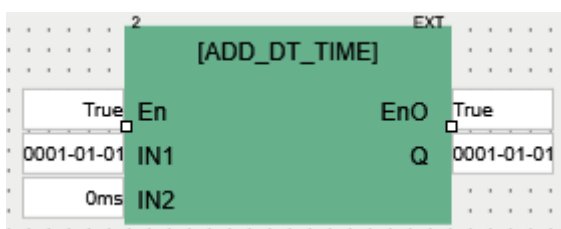
Входы/Выходы	Тип	Описание
IN1	TOD	Вход функции, 1 слагаемое.
IN2	TIME	Вход функции, 2 слагаемое
Q	TOD	Выход функции, сумма времен поданных на входы функции

Пример в ST

```
Q := add_tod_time(IN1, IN2)
```

ADD_DT_TIME

Сложение переменных типа DT и TIME



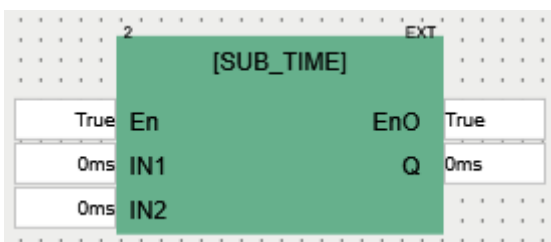
Входы/Выходы	Тип	Описание
IN1	DT	Вход функции, 1 слагаемое.
IN2	TIME	Вход функции, 2 слагаемое
Q	DT	Выход функции, сумма времен поданных на входы функции

Пример в ST

```
Q := add_dt_time(IN1, IN2)
```

SUB_TIME

Вычитание переменных типа TIME



Входы/Выходы	Тип	Описание
IN1	TIME	Вход функции, уменьшаемое
IN2	TIME	Вход функции, вычитаемое
Q	TIME	Выход функции, разность времен поданных на входы функции

Пример в ST

```
Q := sub_time(IN1, IN2)
```

SUB_DATE_DATE

Вычитание переменных типа DATE

Входы/Выходы	Тип	Описание
IN1	DATE	Вход функции, уменьшаемое
IN2	DATE	Вход функции, вычитаемое
Q	TIME	Выход функции, разность времен поданных на входы функции

Входы/Выходы	Тип	Описание
IN1	DATE	Вход функции, уменьшаемое
IN2	DATE	Вход функции, вычитаемое
Q	TIME	Выход функции, разность времен поданных на входы функции

Пример в ST

```
Q := sub_date_date(IN1, IN2)
```

SUB_TOD_TIME

Вычитание переменных типа TOD и TIME

Входы/Выходы	Тип	Описание
IN1	TOD	Вход функции, уменьшаемое
IN2	TIME	Вход функции, вычитаемое
Q	TOD	Выход функции, разность времен поданных на входы функции

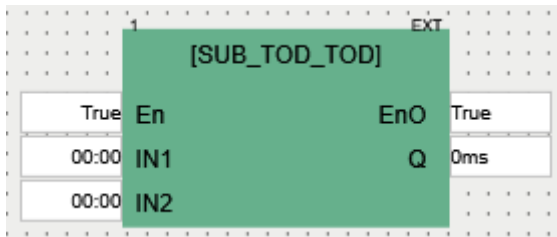
Входы/Выходы	Тип	Описание
IN1	TOD	Вход функции, уменьшаемое
IN2	TIME	Вход функции, вычитаемое
Q	TOD	Выход функции, разность времен поданных на входы функции

Пример в ST

```
Q := sub_tod_time(IN1, IN2)
```

SUB_TOD_TOD

Вычитание переменных типа TOD



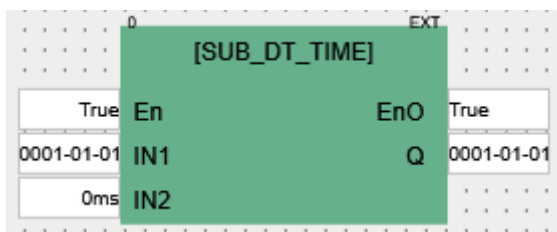
Входы/Выходы	Тип	Описание
IN1	TOD	Вход функции, уменьшаемое
IN2	TOD	Вход функции, вычитаемое
Q	TIME	Выход функции, разность времен поданных на входы функции

Пример в ST

```
Q := sub_tod_tod(IN1, IN2)
```

SUB_DT_TIME

Вычитание переменных типа DT и TIME



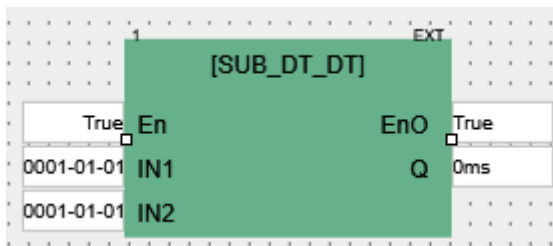
Входы/Выходы	Тип	Описание
IN1	DT	Вход функции, уменьшаемое
IN2	TIME	Вход функции, вычитаемое
Q	DT	Выход функции, разность времен поданных на входы функции

Пример в ST

```
Q := sub_dt_time(IN1, IN2)
```

SUB_DT_DT

Вычитание переменных типа DT



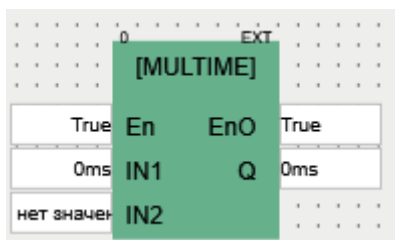
Входы/Выходы	Тип	Описание
IN1	DT	Вход функции, уменьшаемое
IN2	DT	Вход функции, вычитаемое
Q	TIME	Выход функции, разность времен поданных на входы функции

Пример в ST

```
Q := sub_dt_dt(IN1, IN2)
```

MULTIME

Умножение времени на число



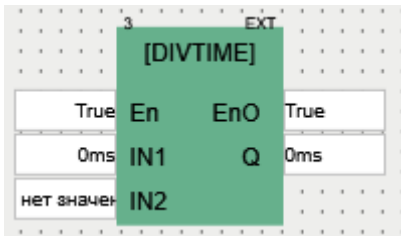
Входы/Выходы	Тип	Описание
IN1	TIME	Вход функции, множимое
IN2	ANY_NUM	Вход функции, множитель
Q	TIME	Выход функции, произведение

Пример в ST

$Q := \text{multime}(\text{IN1}, \text{IN2})$

DIVTIME

Деление времени на число



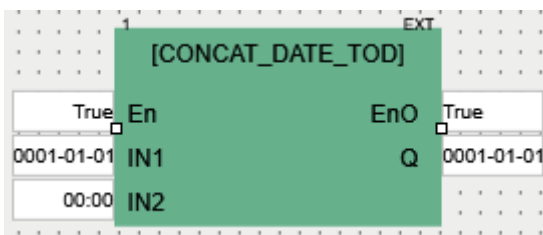
Входы/Выходы	Тип	Описание
IN1	TIME	Вход функции, делимое
IN2	ANY_NUM	Вход функции, делитель
Q	TIME	Выход функции, частное

Пример в ST

$Q := \text{divtime}(\text{IN1}, \text{IN2})$

CONCAT_DATE_TOD

Добавление времени к дате



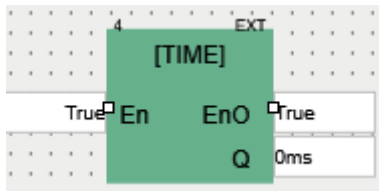
Входы/Выходы	Тип	Описание
IN1	DATE	Вход функции, дата
IN2	TIME	Вход функции, время
Q	DT	Выход функции, результат

Пример в ST

```
Q := concat_date_tod(IN1, IN2)
```

TIME

Таймер исполнительной системы.



Тип данных Q должен быть TIME.

Функция возвращает текущее значение аппаратного таймера контроллера. Аппаратный таймер не следует путать с модулем часов реального времени, который, возможно, имеется в контроллере. Для аппаратного таймера не предусмотрены операции сброса и задания начального значения; таймер всегда начинает отсчет времени с момента включения питания.

Пример в ST

```
Q:=time()
```

Стандартный таймер

Значение стандартного таймера – целое 32-разрядное число миллисекунд. т.е. максимальное значение составляет $(2^{32} - 1)$ мс = t#49d17h02m47s295ms, после чего таймер начинает новый цикл отсчета с 0 (т.н. переполнение таймера). При переполнении разность $\text{timeN}() - \text{timeN-1}()$ отрицательна (N – номер цикла, на котором возникло переполнение). Для корректного вычисления длины временного интервала (что требуется, например, при интегрировании) нужно использовать предварительный перевод значений функции $\text{time}()$ в тип данных DWORD (см. Особенности битовых строк). Следующий код корректно вычисляет длительность цикла ΔT :

```
var deltaT, T_current, T_last:TIME; end_var
```

```
T_current:=time();
```

```
deltaT:=dword_to_time(time_to_dword(T_current)-time_to_dword(T_last));
```

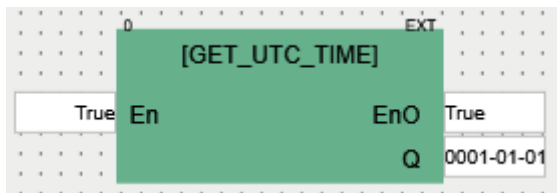
```
T_last:=T_current;
```

Нестандартный таймер

Если значение таймера – целое N-разрядное число миллисекунд, то для корректного вычисления длины временного интервала нужно использовать предварительный перевод значений функции $\text{time}()$ в битовую строку, разрядность которой меньше или равна N. Например, для интервалов, заведомо меньших, чем 255 мс, можно использовать перевод значений функции $\text{time}()$ в тип данных BYTE.

GET_UTC_TIME

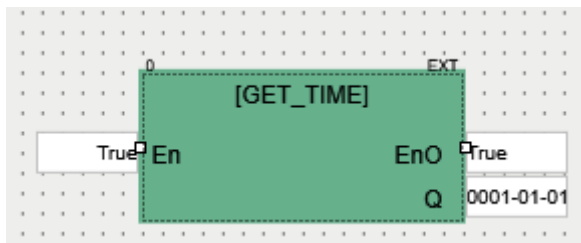
Выдает дату и время UTC заданную в устройстве



Тип данных Q должен быть DATE_AND_TIME.

GET_TIME

Выдает дату и время в том формате, который является стандартным для места исполнения функции: если выполняется в задаче экрана, то выдается локальное время, если работает в исполнительной системе, то в UTC.



Тип данных Q должен быть DATE_AND_TIME.

Пример в ST

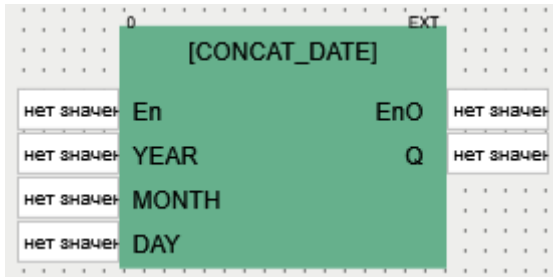
```
Q := get_utc_time()
```

7.8.1.7.1. CONCAT_DATE/CONCAT_DATE_LOCAL

Соединение отдельных значений в формат DATE. Функция находится в категории Функции работы с датой-временем.

Функция CONCAT_DATE_LOCAL сначала переводит параметры отвечающие за дату в локальные, а только потом начинает работу, в отличие от функции CONCAT_DATE. Например, для функции CONCAT_DATE_LOCAL дата 2021.01.01, будет переведена в 2020.12.31 - 21:00:00.

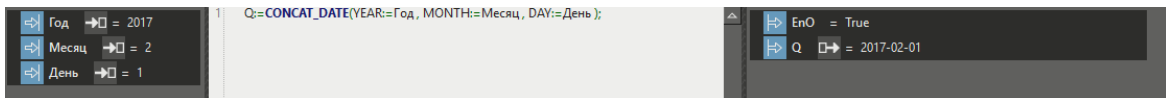
Вид функции в редакторе FBD:



Входы/Выходы	Тип	Описание
YEAR	UINT	Вход функции, отвечающий за год
MONTH	USINT	Вход функции, отвечающий за месяц
DAY	USINT	Вход функции, отвечающий за день
Q	DT	Выход функции, результат

Важно! Если значения параметров не могут быть преобразованы в дату (например, на входе MONTH установлено значение 14), то на выходе функции будет отображаться 0001-01-01

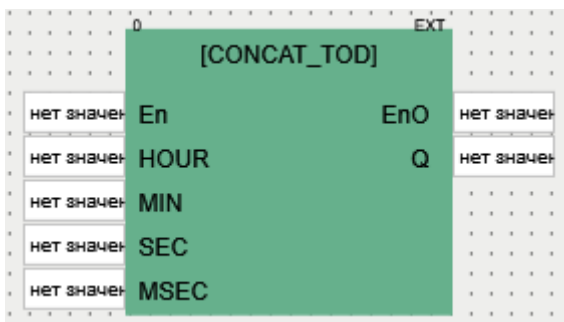
Пример в ST



7.8.1.7.2. CONCAT_TOD

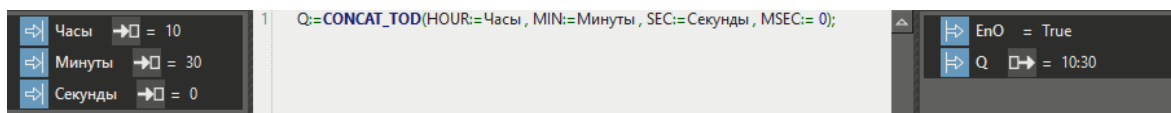
Соединение отдельных значений формат TOD, показывает сколько прошло времени с начала суток. Функция находится в категории Функции работы с датой-временем.

Вид функции в редакторе FBD:



Входы/Выходы	Тип	Описание
HOUR	USINT	Вход функции, отвечающий за часы
MIN	USINT	Вход функции, отвечающий за минуты
SEC	USINT	Вход функции, отвечающий за секунды
MSEC	USINT	Вход функции, отвечающий за миллисекунды
Q	TOD	Выход функции, результат

Пример в ST

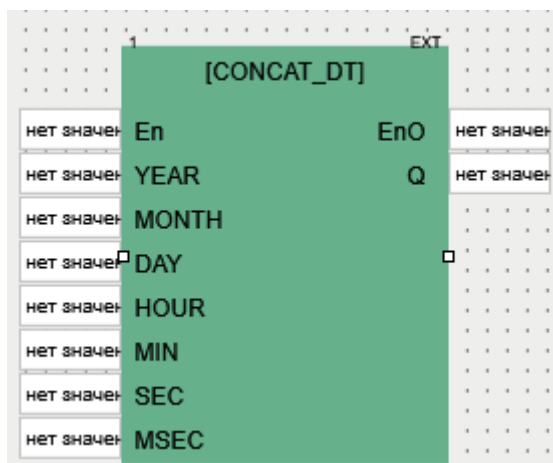


7.8.1.7.3. CONCAT_DT/CONCAT_DT_LOCAL

Соединение отдельных значений формат DT. Функция находится в категории Функции работы с датой-временем.

Функция CONCAT_DT_LOCAL сначала переводит параметры отвечающие за дату и время в локальные, а только потом начинает работу, в отличии от функции CONCAT_DT.

Вид функции в редакторе FBD:



Входы/Выходы	Тип	Описание
YEAR	UINT	Вход функции, отвечающий за год
MONTH	USINT	Вход функции, отвечающий за месяц
DAY	USINT	Вход функции, отвечающий за день
HOUR	USINT	Вход функции, отвечающий за часы
MIN	USINT	Вход функции, отвечающий за минуты
SEC	USINT	Вход функции, отвечающий за секунды
MSEC	USINT	Вход функции, отвечающий за миллисекунды
Q	TOD	Выход функции, результат

Важно! Если значения параметров не могут быть преобразованы в дату или время (например, на входе MONTH установлено значение 14), то на выходе функции будет отображаться 0001-01-01-00:00

Пример в ST

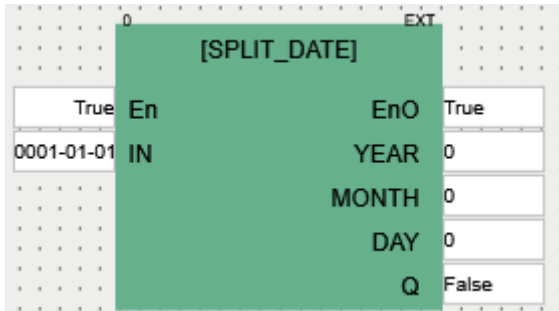
```
Q:=CONCAT_DT(YEAR:=Год , MONTH:=Месяц , DAY:=День , HOUR:=Часы ,
MIN:=Минуты , SEC:=Секунды , MSEC:= 0);
```

7.8.1.7.4. SPLIT_DATE/SPLIT_DATE_LOCAL

Разделение значения DATE на отдельные составляющие. Функции находятся в категории Функции работы с датой-временем.

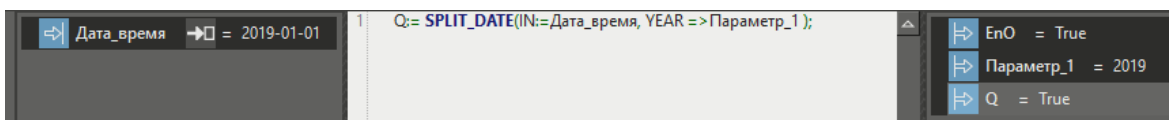
Функция SPLIT_DATE_LOCAL сначала переводит дату в локальную, а только потом начинает разбор, в отличие от функции SPLIT_DATE

Вид функций в редакторе FBD:



Входы/Выходы	Тип	Описание
IN	DATE	Вход функции, задается дата, которую нужно разобрать на составные части
YEAR	UINT	Выход функции, отвечающий за год
MONTH	USINT	Выход функции, отвечающий за месяц
DAY	USINT	Выход функции, отвечающий за день
Q	BOOL	Служебный выход функции

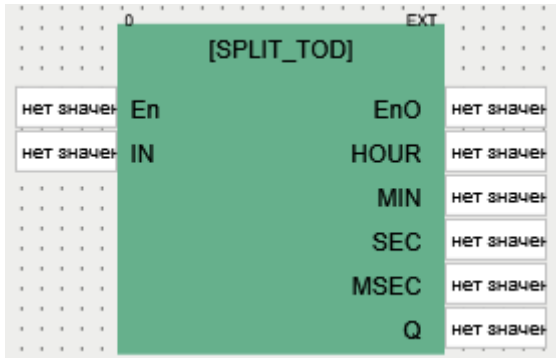
Пример в ST



7.8.1.7.5. SPLIT_TOD

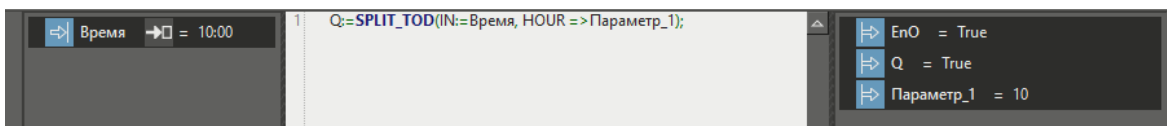
Разделение значения TOD на отдельные составляющие. Функция находится в категории Функции работы с датой-временем.

Вид функции в редакторе FBD:



Входы/Выходы	Тип	Описание
IN	TOD	Вход функции, задается время, которое нужно разобрать на составные части.
HOUR	USINT	Выход функции, отвечающий за часы
MIN	USINT	Выход функции, отвечающий за минуты
SEC	USINT	Выход функции, отвечающий за секунды
MSEC	USINT	Выход функции, отвечающий за миллисекунды
Q	BOOL	Служебный выход функции

Пример в ST

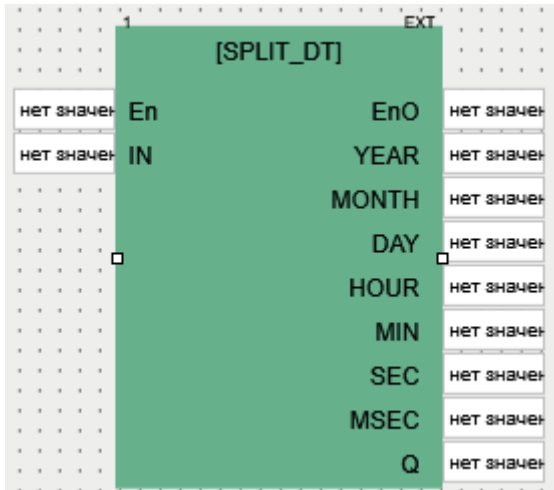


7.8.1.7.6. SPLIT_DT/SPLIT_DT_LOCAL

Разделение значения DT на отдельные составляющие. Функция находится в категории Функции работы с датой-временем.

Функция SPLIT_DT_LOCAL сначала переводит дату и время в локальные, а только потом начинает разбор, в отличие от функции SPLIT_DT

Вид функции в редакторе FBD:



Входы/Выходы	Тип	Описание
IN	TOD	Вход функции, задается время, которое нужно разобрать на составные части.
YEAR	UINT	Выход функции, отвечающий за год
MONTH	USINT	Выход функции, отвечающий за месяц
DAY	USINT	Выход функции, отвечающий за день
HOUR	USINT	Выход функции, отвечающий за часы
MIN	USINT	Выход функции, отвечающий за минуты
SEC	USINT	Выход функции, отвечающий за секунды
MSEC	USINT	Выход функции, отвечающий за миллисекунды
Q	BOOL	Служебный выход функции

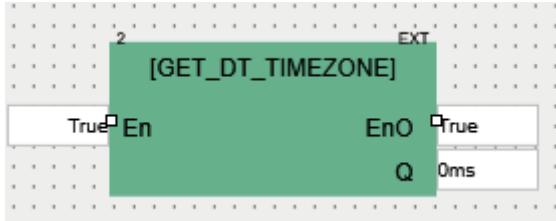
Пример в ST

```
Q:=SPLIT_DT(IN:=Время, HOUR =>Параметр_1, MONTH=>Параметр_2);
```

7.8.1.7.7. GET_DT_TIMEZONE

Функция GET_DT_TIMEZONE служит для получения смещения временной зоны относительно UTC.

Вид функции в редакторе FBD:



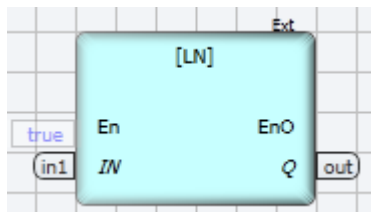
Выход Q имеет тип данных TIME.

Важно! При изменении часового пояса требуется перезагрузка ОС

7.8.1.8. ЛОГАРИФМИЧЕСКИЕ ФУНКЦИИ

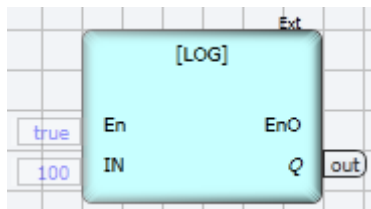
- LN, натуральный логарифм
- LOG, десятичный логарифм
- EXP, экспонента

LN, натуральный логарифм



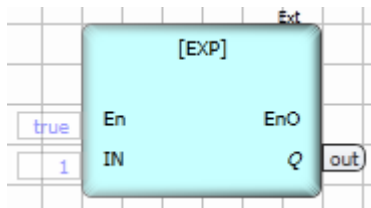
$Q := \ln(IN)$

LOG, десятичный логарифм



$Q := \log(IN)$

EXP, экспонента

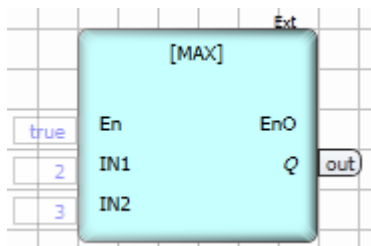


$Q := e ** IN$

7.8.1.9. ФУНКЦИИ ВЫБОРА

- MAX, максимальное из двух
- MIN, минимальное из двух
- SEL, выбор из двух
- MUX, мультиплексор
- LIMIT, ограничение

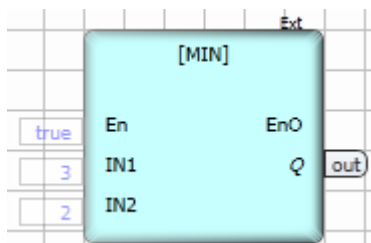
MAX, максимальное из двух



Функция возвращает наибольшее из двух значений:

$Q := \max(IN1, IN2)$

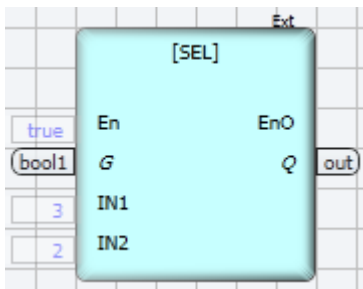
MIN, минимальное из двух



Функция возвращает наименьшее из двух значений:

$Q := \min(IN1, IN2)$

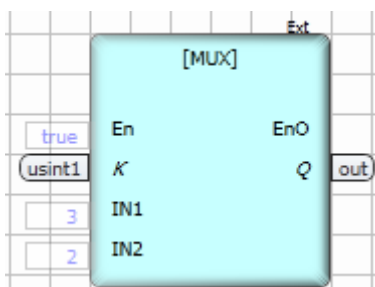
SEL, выбор из двух



$Q := \text{sel}(G, \text{IN1}, \text{IN2})$

Если $G=\text{FALSE}$, $Q=\text{IN1}$, если $G=\text{TRUE}$, $Q=\text{IN2}$

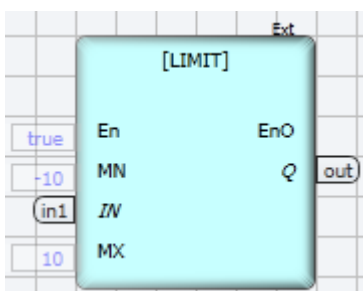
MUX, мультиплексор



$Q := \text{mux}(K, \text{IN1}, \text{IN2})$

Если $K=0$, $Q=\text{IN1}$, если $K=1$, $Q=\text{IN2}$

LIMIT, ограничение



$Q := \text{limit}(\text{MN}, \text{IN}, \text{MX})$

Функция работает по следующему алгоритму:

$Q := \min(\max(\text{IN}, \text{MN}), \text{MX})$

Если $\text{MN} \leq \text{IN} \leq \text{MX}$, $Q := \text{IN}$.

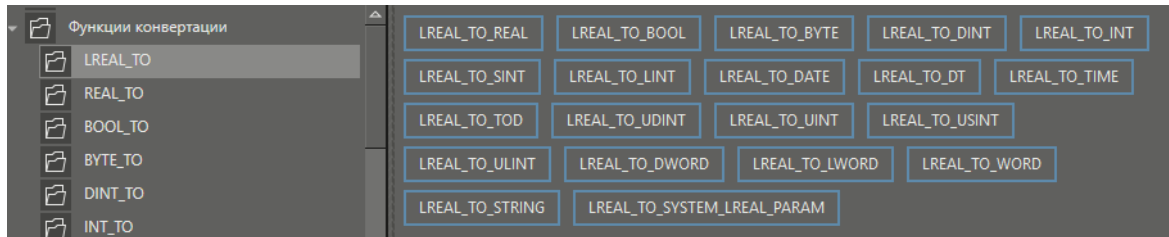
Если $\text{IN} < \text{MN}$, $Q := \text{MN}$.

Если $\text{IN} > \text{MX}$, $Q := \text{MX}$.

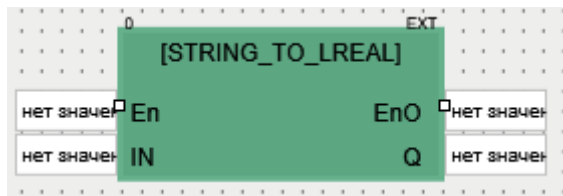
7.8.1.10. ФУНКЦИИ КОНВЕРТАЦИИ

Данная категория содержит внутри себя различные группы функций предназначенных для конвертации из одних типов данных в другие. Работа функций не зависит от места исполнения: могут работать как в задачах узла, так и в задаче экрана.

Группа LREAL_TO:



Каждый блок данного раздела имеет схожий внешний вид и набор входов и выходов:

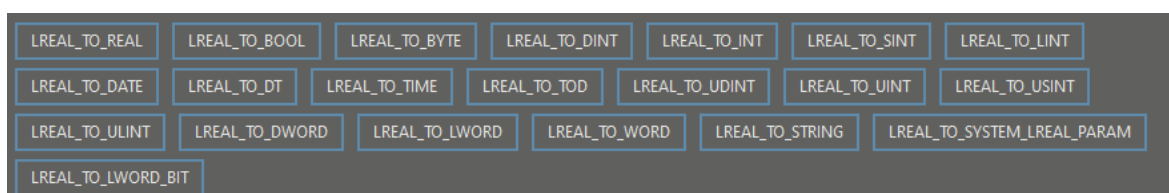


Входы/Выходы	Назначение
IN	Вход функции конвертации. Тип входа зависит от назначения функции. Например, для функций начинающихся со SRTING_TO тип будет STRING, а для функций LREAL_TO тип будет LREAL.
Q	Выход функции. Тип выхода зависит от назначения функции. Например, для функций оканчивающихся на TO_LREAL тип будет LREAL, а для функций TO_DATE тип будет DATE

7.8.1.10.1.LREAL_TO

Группа категории палитры Функции конвертации содержит функции предназначенные для конвертации в различные типы LREAL-параметров.

Вид категории в палитре:



Доступны следующие функции:

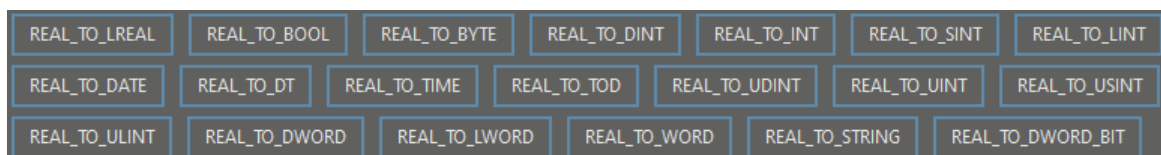
Название	Особенности работы
LREAL_TO_REAL	Преобразование с округлением
LREAL_TO_BOOLEAN	
LREAL_TO_BYTE	
LREAL_TO_DINT	Преобразование с округлением
LREAL_TO_INT	Преобразование с округлением
LREAL_TO_SINT	Преобразование с округлением
LREAL_TO_LINT	Преобразование с округлением
LREAL_TO_DATE	
LREAL_TO_DT	
LREAL_TO_TIME	Возвращает количество миллисекунд
LREAL_TO_TOD	
LREAL_TO_UDINT	Преобразование с округлением
LREAL_TO_UINT	Преобразование с округлением
LREAL_TO_USINT	Преобразование с округлением

Название	Особенности работы
LREAL_TO_ULINT	Преобразование с округлением
LREAL_TO_DWORD	
LREAL_TO_LWORD	Передача двоичных данных
LREAL_TO_WORD	
LREAL_TO_STRING	
LREAL_TO_SYSTEM_LREAL_PARAMETER	К текущему значению добавляется метка времени равная времени преобразованию и признак признака качества GOOD.
LREAL_TO_LWORD_BIT	Преобразует LREAL в LWORD побитово

7.8.1.10.2.REAL_TO

Группа категории палитры Функции конвертации содержит функции предназначенные для конвертации в различные типы REAL-параметров.

Вид категории в палитре:



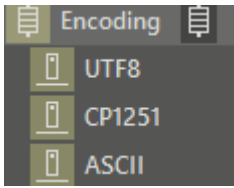
Доступны следующие функции:

Название	Особенности работы
REAL_TO_REAL	
REAL_TO_BOOL	
REAL_TO_BYTE	
REAL_TO_DINT	
REAL_TO_INT	
REAL_TO_SINT	
REAL_TO_LINT	
REAL_TO_DATE	
REAL_TO_DT	
REAL_TO_TIME	Возвращает количество миллисекунд
REAL_TO_TOD	
REAL_TO_UDINT	
REAL_TO_UINT	
REAL_TO_USINT	
REAL_TO_ULINT	
REAL_TO_DWORD	

Название	Особенности работы
REAL_TO_LWORD	Передача двоичных данных
REAL_TO_WORD	
REAL_TO_STRING	
REAL_TO_DWORD_BIT	Преобразует LREAL в DWORD побитово
REAL_TO_SYSTEM_REAL_PARAMETER	К текущему значению добавляется метка времени равная времени преобразованию и признак признака качества GOOD.

7.8.1.10.3. BYTE_TO

Группа категории палитры Функции конвертации содержит функцию предназначенную для конвертации параметров типа BYTE в различные типы данных:

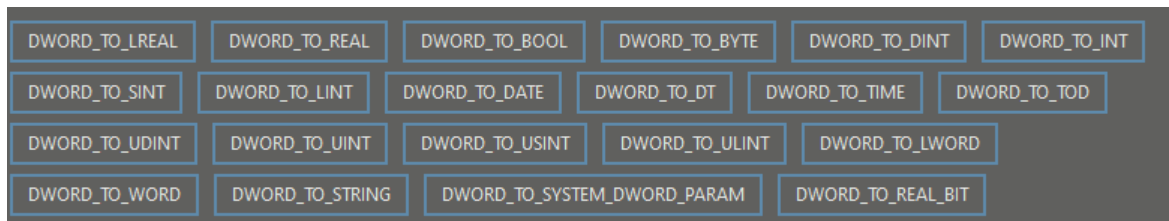
Название	Назначение
BYTE_ARRAY_TO_STRING	<p>Вход функции имеет тип ARRAY [*] OF BYTE на выходе получим значение STRING. Каждый байт массива отвечает за один символ строки, согласно таблице символов согласно выбранному значению на входе функции Encoding. Вход функции имеет одноименный тип данных. Тип данных находится в категории библиотеки Стандартная. Типы</p>  <p>данных. Системные. Encoding:</p> <p>Для обратного преобразования нужно использовать функцию STRING_TO_BYTE_ARRAY</p>

Название	Назначение
BYTE_TO_SYSTEM_BYTE_PARAMETER	К текущему значению добавляется метка времени равная времени преобразованию и признак признака качества GOOD.

7.8.1.10.4.DWORD_TO

Группа категории палитры Функции конвертации содержит функции предназначенные для конвертации в различные типы DWORD-параметров.

Вид категории в палитре:



Доступны следующие функции:

Название	Особенности работы
DWORD_TO_REAL	
DWORD_TO_REAL	
DWORD_TO_BOOL	
DWORD_TO_BYTE	
DWORD_TO_DINT	
DWORD_TO_INT	

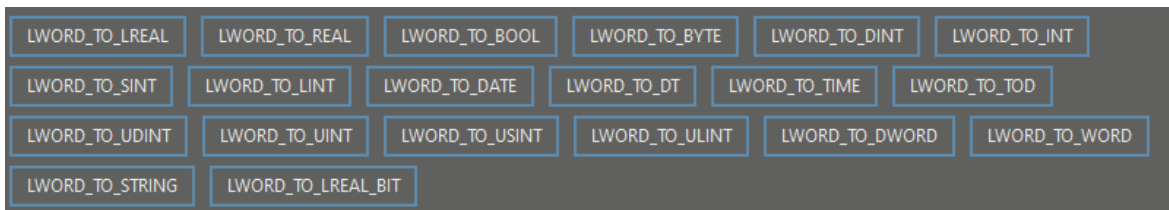
Название	Особенности работы
DWORD_TO_SINT	
DWORD_TO_LINT	
DWORD_TO_DATE	
DWORD_TO_DT	
DWORD_TO_TIME	
DWORD_TO_TOD	
DWORD_TO_UDINT	
DWORD_TO_UINT	
DWORD_TO_USINT	
DWORD_TO_ULINT	
DWORD_TO_WORD	
LREAL_TO_WORD	
DWORD_TO_STRING	

Название	Особенности работы
DWORD_TO_SYSTEM_PARAM	К текущему значению добавляется метка времени равная времени преобразованию и признак качества GOOD.
DWORD_TO_REAL_BIT	Преобразует DWORD в REAL побитово

7.8.1.10.5.LWORD_TO

Группа категории палитры Функции конвертации содержит функции предназначенные для конвертации в различные типы LWORD-параметров.

Вид категории в палитре:



Доступны следующие функции:

Название	Особенности работы
LWORD_TO_REAL	
LWORD_TO_REAL	
LWORD_TO_BOOL	
LWORD_TO_BYTE	
LWORD_TO_DINT	

Название	Особенности работы
LWORD_TO_INT	
LWORD_TO_SINT	
LWORD_TO_LINT	
LWORD_TO_DATE	
LWORD_TO_DT	
LWORD_TO_TIME	
LWORD_TO_TOD	
LWORD_TO_UDINT	
LWORD_TO_UINT	
LWORD_TO_USINT	
LWORD_TO_ULINT	
LWORD_TO_DWORD	
LREAL_TO_WORD	
LWORD_TO_STRING	

Название	Особенности работы
LWORD_TO_LREAL_BIT	Преобразует LWORD в LREAL побитово
LWORD_TO_SYSTEM_LWORD_PARAM	К текущему значению добавляется метка времени равная времени преобразованию и признак признака качества GOOD.

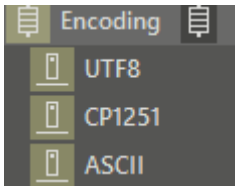
7.8.1.10.6.STRING_TO

Группа категории палитры Функции конвертации содержит функции предназначенные для конвертации в различные типы STRING-параметров.

Доступны следующие функции:

Название	Особенности работы
STRING_TO_LREAL	
STRING_TO_REAL	
STRING_TO_BOOLEAN	
STRING_TO_BYTE	
STRING_TO_DINT	
STRING_TO_INT	
STRING_TO_SINT	

Название	Особенности работы
STRING_TO_LINE	
STRING_TO_DATE	
STRING_TO_DT	Принимает строку с временем в UTC формате и возвращает время в UTC, далее полученное время для вывода пользователю преобразуется в локальное, т.е. например, для московского региона добавляется +3 часа
STRING_TO_TIME	
STRING_TO_TO	
STRING_TO_UNICODE	
STRING_TO_UINT	
STRING_TO_USINT	
STRING_TO_ULINT	
STRING_TO_WORD	
STRING_TO_LWORD	
STRING_TO_WORD	

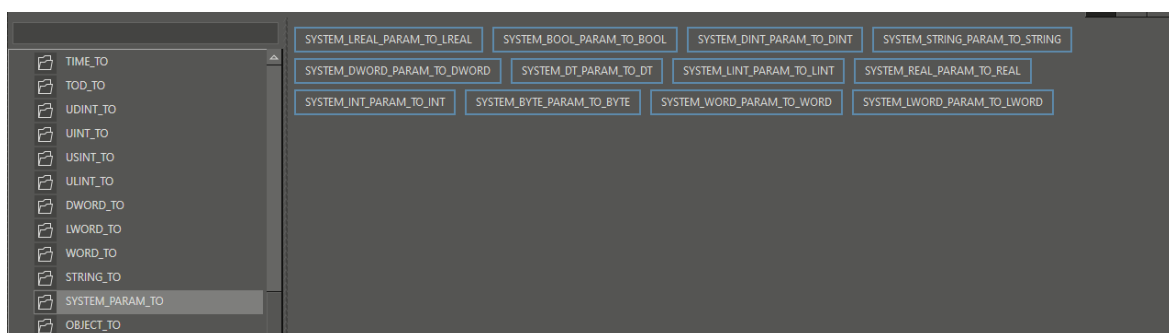
Название	Особенности работы
LREAL_TO_SYSTEM_LREAL_PARAMETER	К текущему значению добавляется метка времени равная времени преобразованию и признак качества GOOD.
STRING_TO_OBJECT	<p>На вход функции подается значение типа STRING в формате JSON, на выходе функции получим значение ANY. В дальнейшем выход функции надо присвоить значению параметра, тип которого соответствует полученному объекту. Правила формирования типов данных для работы с JSON смотрите в разделе Соответствие элементов JSON и MasterSCADA 4D</p>
STRING_TO_BYTE_ARRAY	<p>Преобразует строку в массив байт, каждый символ преобразуется в соответствующий код, согласно выбранному указанному на входе Encoding значению. Тип выхода функции ARRAY [*] OF BYTE. Первый символ строки занимает первый элемент массива. Для обратного преобразования нужно использовать BYTE_ARRAY_TO_STRING</p> <p>Пример: Out := BYTE_ARRAY_TO_STRING(IN := Arr, ENCODING := Encod#ASCII), где Arr - это строка, а Encod это параметр, который имеет тип данных, как и вход функции ENCODING. Тип данных находится в категории библиотеки Стандартная.Типы</p> <div data-bbox="1054 1406 1289 1592" style="text-align: right;">  </div> <p>данных.Системные.Encoding:</p>
STRING_TO_DATE_LOCAL	
STRING_TO_DATETIME_LOCAL	<p>Принимает строку с временем в локальном (относительно контроллера) формате, и преобразует его в UTC, в случае для московской часовой зоны отнимает 3 час и возвращает время в UTC, далее полученное время для вывода пользователю в среде разработки, подключенной к среде</p>

Название	Особенности работы
	<p>исполнения преобразуется в локальное, т.е. например, для московского региона добавляется +3 часа</p> <p>В случае когда на вход <code>STRING_TO_DT_LOCAL</code> подаётся не строка, а <code>DT</code> оно преобразуется в строку через функцию <code>DT_TO_STRING</code>, которое возвращает строку в с временем в UTC формате</p> <p>Пример: контроллер находится во Владивостоке (GMT+10), среда разработки подключена к исполняемому проекту в Москве (GMT+3), а клиент визуализации работает в Тюмени (GMT+5). Тогда при вызове функции <code>STRING_TO_DT_LOCAL("2020-01-01-00:00")</code> получим:</p> <ul style="list-style-type: none"> • В среде разработки будет отображено время 2019-12-31-17:00 ("2020-01-01-00:00" - 10ч + 3ч) • В клиенте визуализации будет отображено время 2019-12-31-19:00 ("2020-01-01-00:00" - 10ч + 5ч)

7.8.1.10.7.SYSTEM_PARAM_TO

Группа категории палитры Функции конвертации содержит функции предназначенные для конвертации сложных типов данных, структуры вида `SYSTEM*_PARAM`, содержащей значение, метку времени и признак качества, в различные простые типы.

Вид в палитре:



Доступны следующие функции:

- `SYSTEM_LREAL_PARAM_TO_LREAL`
- `SYSTEM_BOOL_PARAM_TO_BOOL`
- `SYSTEM_DINT_PARAM_TO_DINT`
- `SYSTEM_STRING_PARAM_TO_STRING`
- `SYSTEM_DWORD_PARAM_TO_DWORD`

- SYSTEM_DT_PARAM_TO_DT
- SYSTEM_LINT_PARAM_TO_LINT
- SYSTEM_REAL_PARAM_TO_REAL
- SYSTEM_INT_PARAM_TO_INT
- SYSTEM_BYTE_PARAM_TO_BYTE
- SYSTEM_WORD_PARAM_TO_WORD
- SYSTEM_LWORD_PARAM_TO_LWORD

Если связать в проекте два параметра например, типа SYSTEM_LREAL_PARAM и LREAL, то конвертация сработает автоматически.

Если связать в проекте два параметра например, типа SYSTEM_LREAL_PARAM и SYSTEM_INT_PARAM, то конвертация сработает автоматически. Поле Value будет конвертировано LREAL в INT, а другие поля будут скопированы без изменения.

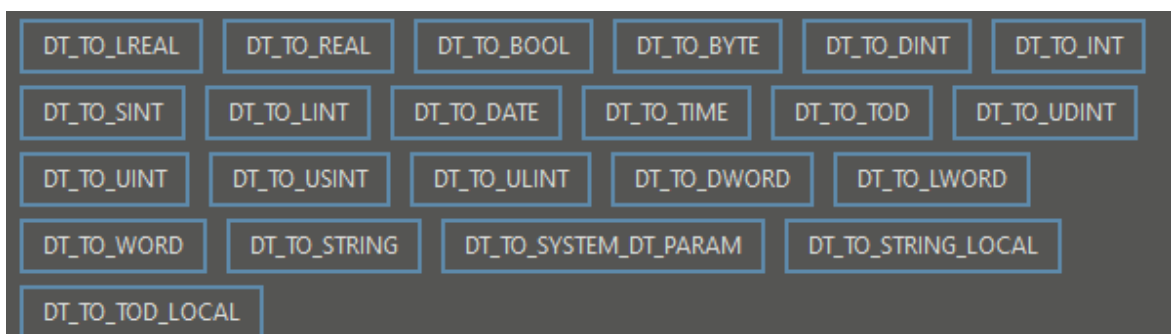
7.8.1.10.8.OBJECT_TO

Группа категории палитры Функции конвертации содержит функцию предназначенную для конвертации сложных типов данных (массив или структура) в строку, в формате JSON

Название	Назначение
OBJECT_TO_STRING	Вход функции имеет тип ANY, на него можно подать массив или структуру(массив или структура), на выходе получим значение STRING в формате JSON

7.8.1.10.9.DT_TO

Группа категории палитры Функции конвертации содержит функции предназначенные для конвертации в различные типы DT-параметров:



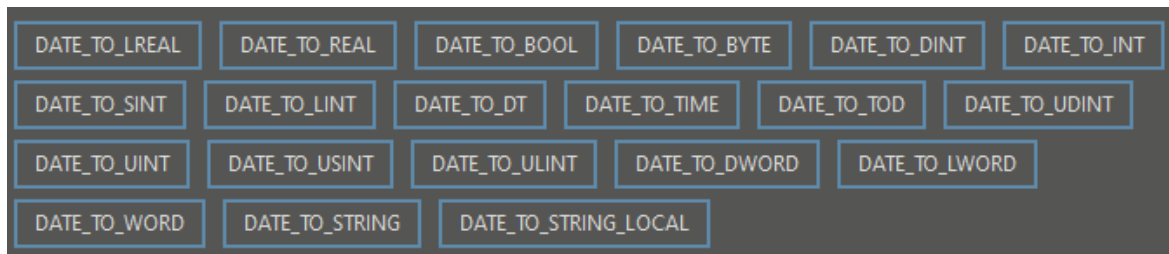
Доступны следующие функции:

Название	Назначение
DT_TO_LREAL	
DT_TO_REAL	
DT_TO_BOOL	
DT_TO_BYTE	
DT_TO_DINT	
DT_TO_INT	
DT_TO_SINT	
DT_TO_LINT	
DT_TO_DATE	
DT_TO_TIME	
DT_TO_TOD	
DT_TO_UDINT	
DT_TO_UINT	
DT_TO_USINT	
DT_TO_ULINT	
DT_TO_DWORD	
DT_TO_LWORD	

Название	Назначение
DT_TO_WORD	
DT_TO_STRING	
DT_TO_SYSTEM_DT_PARAM	
DT_TO_STRING_LOCAL	На вход принимает дату и время UTC, но перед преобразованием конвертируют время в локальное, и преобразование уже осуществляется над локальными значениями.
DT_TO_TOD_LOCAL	

7.8.1.10.10. DATE_TO

Группа категории палитры Функции конвертации содержит функции предназначенные для конвертации в различные типы DATE-параметров:



Доступны следующие функции:

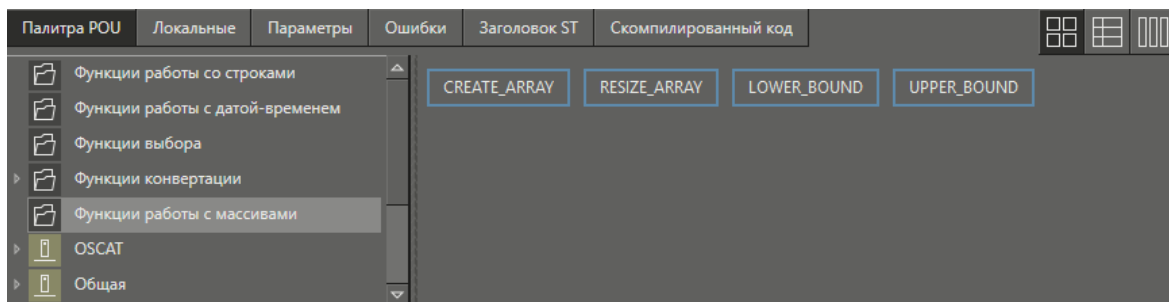
Название	Назначение
DATE_TO_REAL	
DATE_TO_REAL	

Название	Назначение
DATE_TO_BOOLEAN	
DATE_TO_BYTE	
DATE_TO_DATE	
DATE_TO_INTEGER	
DATE_TO_SHORT_INTEGER	
DATE_TO_LONG_INTEGER	
DATE_TO_DATETIME	
DATE_TO_TIME	
DATE_TO_TOD	
DATE_TO_UNICODE	
DATE_TO_UNICODE_INTEGER	
DATE_TO_UNICODE_SHORT_INTEGER	
DATE_TO_UNICODE_LONG_INTEGER	
DATE_TO_UNICODE_DATETIME	
DATE_TO_UNICODE_TIME	
DATE_TO_UNICODE_TOD	

Название	Назначение
DATE_TO_LWORD	
DATE_TO_WORD	
DATE_TO_STRING	
DATE_TO_STRING_LOCAL	На вход принимает дату UTC, но перед преобразованием конвертируют ее в локальную, и преобразование уже осуществляется над локальными значениями.

7.8.1.11. ФУНКЦИИ РАБОТЫ С МАССИВАМИ

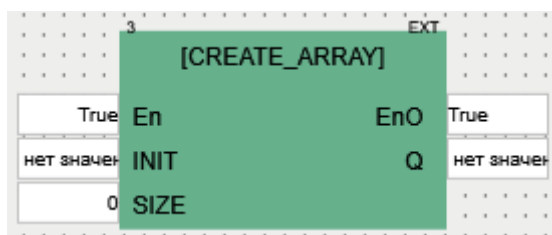
Вид категории в палитре языков программирования:



Данная категория, как правило, используется для параметров, для которых назначен тип данных Массив с динамическим размером осей.

Функция CREATE_ARRAY

Функция создает одномерный массив



Входы/Выходы	Тип	Описание
INIT	ANY	Вход функции, где задается начальное значение элементов массива
SIZE	ANY_INT	Вход функции, где задается количество элементов в массиве
Q	ANY	Выход функции, который можно связать с параметром, у которого назначен тип данных Массив с динамическим размером осей

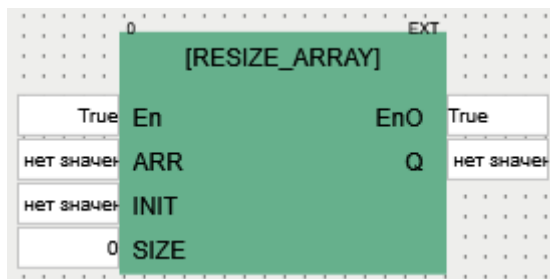
Пример в ST



Выход программы Параметр_1 имеет тип Массив с динамическим размером осей

Функция RESIZE_ARRAY

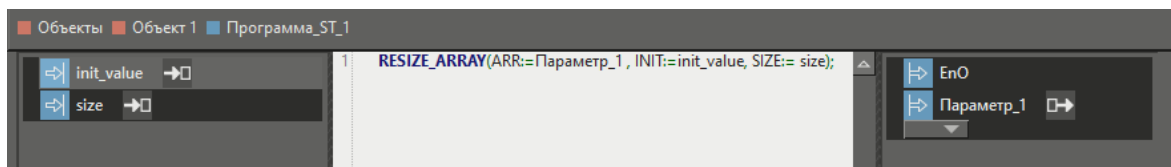
Функция изменяет размер ранее созданного одномерного массива



Входы/Выходы	Тип	Описание
ARR	ANY	Вход функции, который необходимо связать с параметром, у которого тип данных установлен Массив с динамическим размером осей
INIT	ANY	Вход функции, где задается начальное значение новых элементов массива.

Входы/Выходы	Тип	Описание
SIZE	ANY_INT	Вход функции служит для задания нового размера массива.
Q	ANY	Выдается нижний индекс массива. Как правило, нижний индекс массива будет равен 1, за исключением тех случаев, когда при создании массива, указали иной начальный индекс

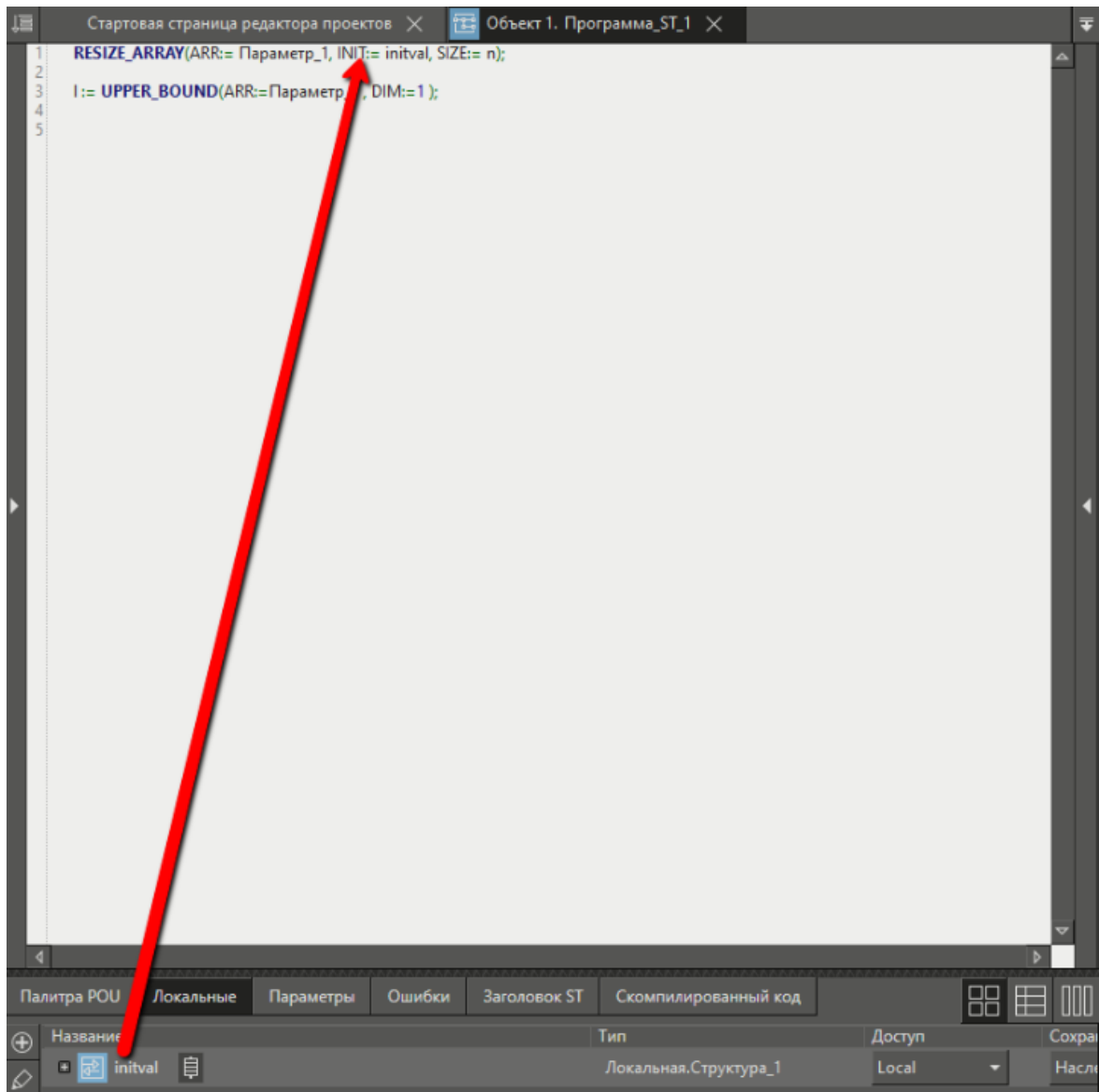
Пример в ST для массива чисел



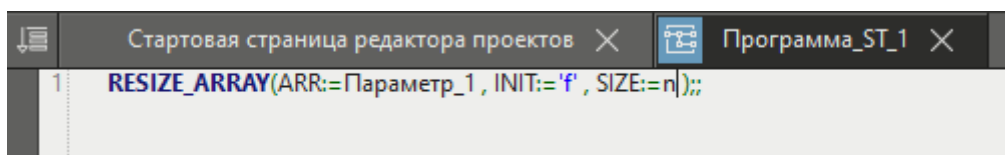
В данном случае функция изменит размер массива, который указан в качестве аргумента. Новые значения массива будут равны значению переменной `init_value`, а размер массива будет зависеть от значения входа программы `new_size`.

Пример в ST для массива структур

Сперва нужно на вкладке Локальные создать переменную, с типом данных Структура. Затем эту переменную нужно перетащить в программу:



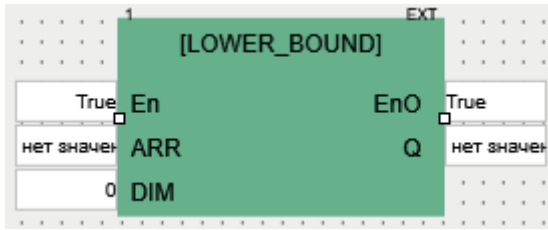
Пример в ST для строк



Порядок работы такой-же как и для числовых одномерных массивов.

Функция LOWER_BOUND

Функция получает нижнюю границу массива (минимальный индекс).



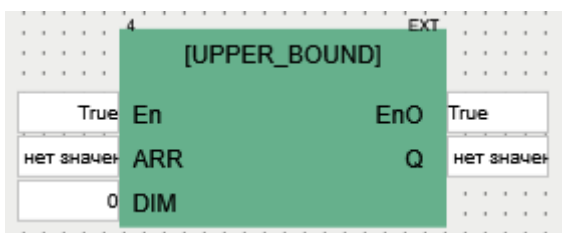
Входы/Выходы	Тип	Описание
ARR	ANY	Вход функции, который необходимо связать с массивом, границу которого нужно проверить.
DIM	ANY_ INT	Вход функции, где задается номер измерения (оси), индекс которого необходимо получить. В случае работы с двумерным массивом если установить 1, то получится минимальный индекс по вертикали, а если 2, то по горизонтали.
Q	ANY	Выдается нижний индекс массива. Как правило, нижний индекс массива будет равен 1, за исключением тех случаев, когда при создании массива, указали иной начальный индекс

Пример в ST

`low_1:=LOWER_BOUND(Массив,1);` - получение значения индекса нижней границы первой оси

Функция UPPER_BOUND

Функция получает верхнюю границу массива (максимальный индекс).



Входы/Выходы	Тип	Описание
ARR	ANY	Вход функции, который необходимо связать с массивом, границу которого нужно проверить.

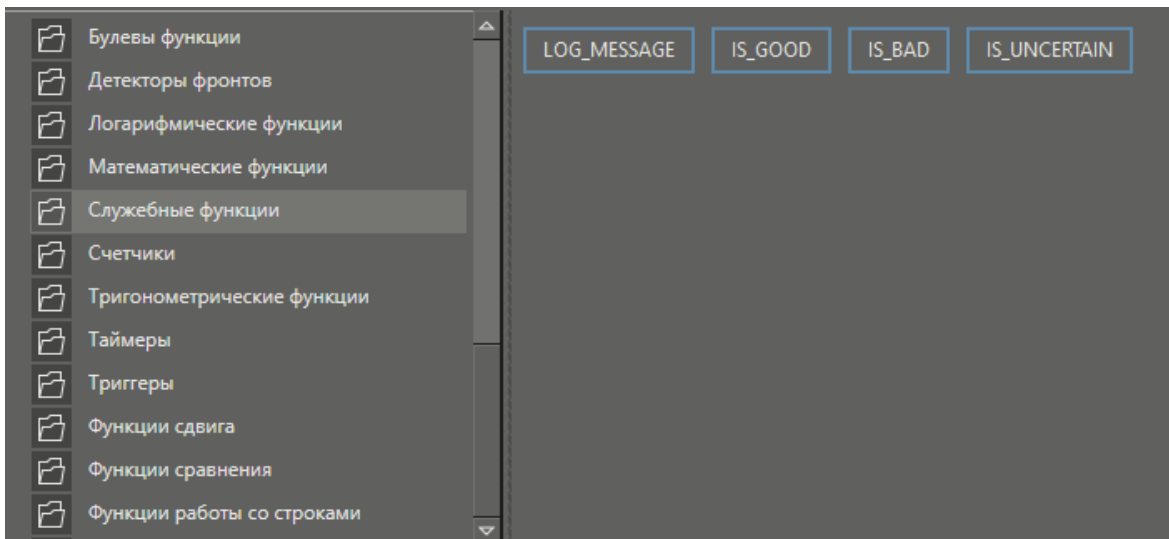
Входы/Выходы	Тип	Описание
DIM	ANY_ INT	Вход функции, где задается номер измерения (оси), индекс которого необходимо получить. В случае работы с двумерным массивом если установить 1, то получится максимальный индекс по вертикали, а если 2, то по горизонтали.
Q	ANY	Выдается верхний индекс массива.

Пример в ST



7.8.1.12. СЛУЖЕБНЫЕ ФУНКЦИИ

Вид категории в палитре языков программирования:



Функция LOG_MESSAGE

Функция используется для формирования сообщений, которые сохраняются в лог исполнительной системы



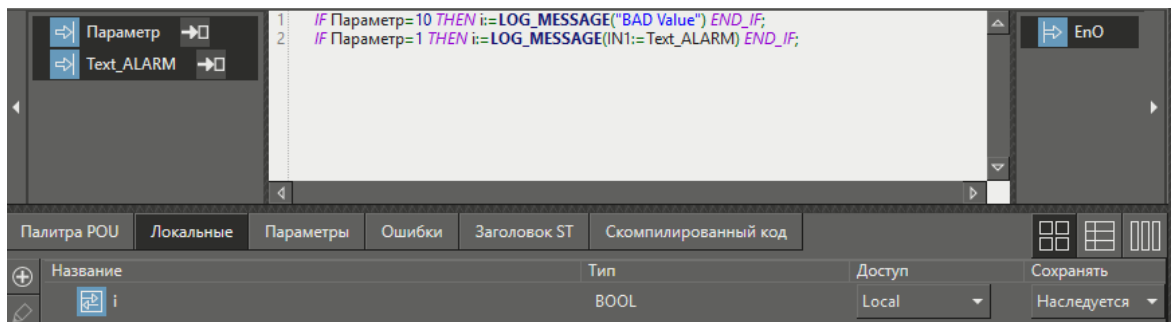
Специальные входы и выходы.

Название	Назначение
IN1	Вход. Тип STRING. Служит для формирования текста сообщения, который запишется в лог исполнительной системы
Q	Выход. Тип BOOL. Принимает значение TRUE, когда сообщение было сформировано

Сообщение будет записываться в лог исполнительной системы после вызова в программе, например, в ST.

Функция может работать в программе, которая выполняется в задаче экрана, в этом случае лог-файле будет зафиксирован IP-адрес клиента, с которого пришло сообщение.

Пример использования в ST:

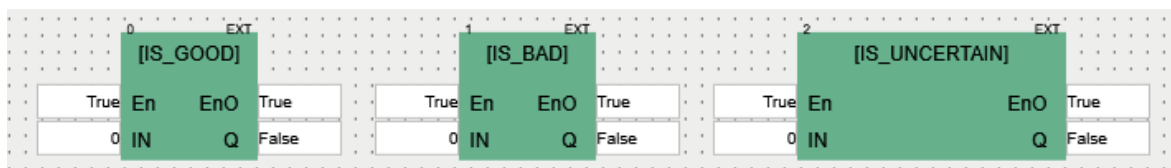


В первой строке показан пример вызова функции, когда сообщение не может быть изменено в режиме исполнения. Во второй строке текст сообщения зависит от строкового параметра Text_ALARM

Локальная переменная i принимает значение выхода функции Q, после того как сообщение будет сформировано.

Функции IS_GOOD, IS_BAD, IS_UNCERTAIN

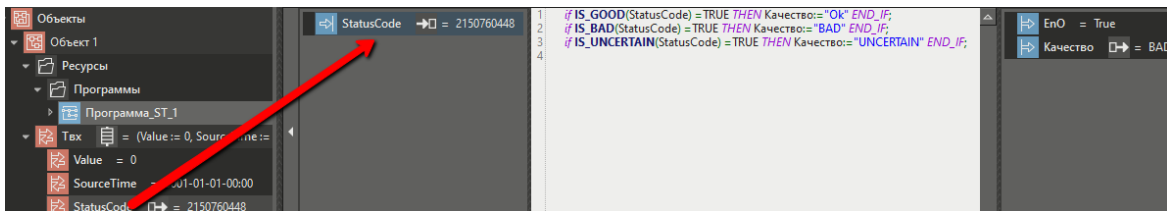
Данные функции необходимы для обработки признаков качества параметров, которые имеют тип SYSTEM_*_PARAM.



Специальные входы и выходы.

Название	Назначение
IN	Вход. Тип UDINT. Данный вход нужно связать с полем StatusCode структуры SYSTEM_*_PARAM
Q	Выход. Тип BOOOL. Принимает значение TRUE, если StatusCode принимает значение, соответствующее проверяемому признаку.

Пример использования в ST:

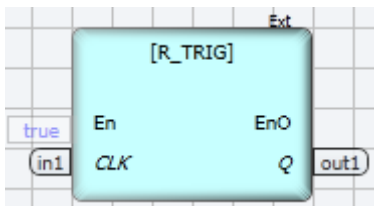


7.8.2. БИБЛИОТЕКА СТАНДАРТНЫЕ ФБ

7.8.2.1. ФБ 'ДЕТЕКТОРЫ ФРОНТОВ'

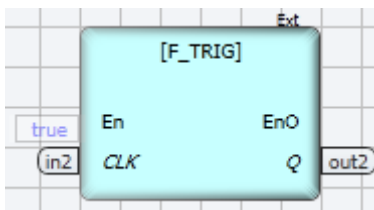
- R_TRIG, импульс по переднему фронту
- F_TRIG, импульс по заднему фронту

R_TRIG, импульс по переднему фронту



Q=1 в течение 1 такта при изменении значения CLK с 0 на любое ненулевое.

F_TRIG, импульс по заднему фронту

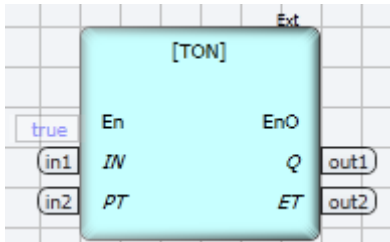


Q=1 в течение 1 такта при изменении значения CLK с любого ненулевого на 0.

7.8.2.2. ФБ 'ТАЙМЕРЫ'

- TON, задержка включения
- TOF, задержка выключения
- TP, импульс произвольной длительности

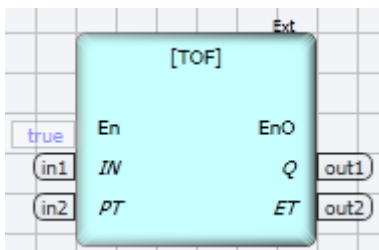
TON, задержка включения



При изменении значения IN с 0 на любое ненулевое Q принимает значение 1 с задержкой в PT, при этом ET индицирует время, прошедшее с момента изменения IN.

Чтобы обнулить Q и ET, нужно подать 0 на вход IN.

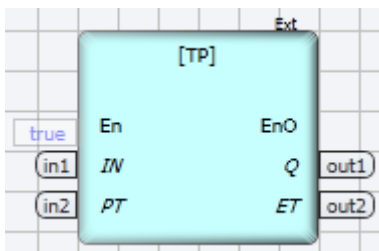
TOF, задержка выключения



Если $IN \neq 0$, то $Q = 1$ и $ET = 0$.

При изменении значения IN с любого ненулевого на 0 Q принимает значение 0 с задержкой в PT, при этом ET индицирует время, прошедшее с момента изменения IN.

TP, импульс произвольной длительности



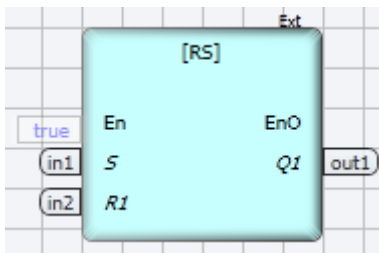
При изменении значения IN с 0 на любое ненулевое $Q=1$ на время PT. ET индицирует время, прошедшее с начала формирования импульса.

7.8.2.3. ФБ 'ТРИГГЕРЫ'

- RS, триггер с приоритетом по сбросу

- SR, триггер с приоритетом по установке

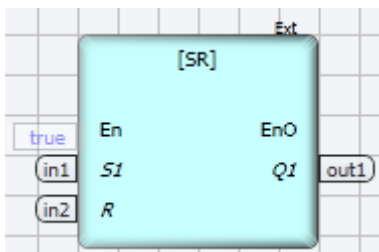
RS, триггер с приоритетом по сбросу



Если $R1 \neq 0$, $Q1=0$.

$Q1=1$, если $S \neq 0$ и $R1=0$.

SR, триггер с приоритетом по установке



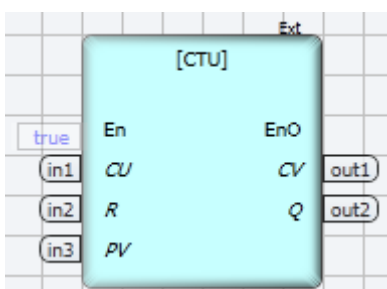
Если $S1=0$ и $R \neq 0$, $Q1=0$.

$Q1=1$, если $S1 \neq 0$.

7.8.2.4. ФБ 'СЧЕТЧИКИ'

- СТУ, счетчик прямой
- СТД, счетчик обратный
- СТУД, счетчик комбинированный

СТУ, счетчик прямой



Входы:

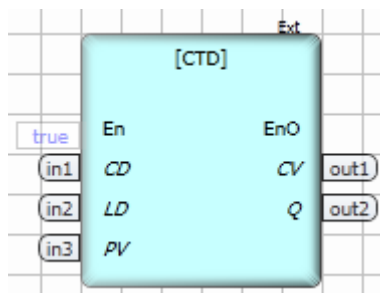
- CV (BOOL) – по переднему фронту на этом входе CV увеличивается на 1;

- R (BOOL) – если TRUE, то CV сбрасывается в 0;
- PV (INT) – верхний предел значений CV.

Выходы:

- Q (BOOL) – становится TRUE, когда счетчик достигнет значения заданного PV;
- CV (INT) – значение, которое увеличивается на 1, пока не достигнет PV.

CTD, счетчик обратный



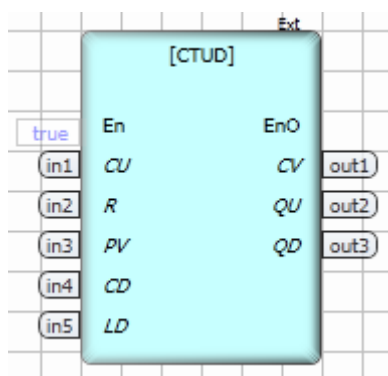
Входы:

- CD (BOOL) – по переднему фронту на этом входе CV уменьшается на 1;
- LD (BOOL) – если TRUE, CV сбрасывается в заданное значение верхнего предела (PV);
- PV (INT) – верхний предел, т.е. начальное значение декремента.

Выходы:

- Q (BOOL) – становится TRUE, когда CV становится 0;
- CV (INT) – значение, которое уменьшается на 1, начиная с PV и до 0.

CTUD, счетчик комбинированный



Входы:

- CU (BOOL) – по переднему фронту на этом входе CV увеличивается на 1;

- CD (BOOL) – по переднему фронту на этом входе CV уменьшается на 1;
- R (BOOL) – если TRUE, то CV сбрасывается в 0;
- LD (BOOL) – если TRUE, то CV сбрасывается в заданное значение верхнего предела (PV);
- PV (INT) – верхний предел значений CV.

Выходы:

- QU (BOOL) – TRUE, если CV больше или равен PV;.
- QD (BOOL) – TRUE, если CV равен 0;
- CV (INT) – изменяемое значение.

7.9. ВАЖНО ЗНАТЬ

В разделе находятся статьи, в которых содержатся особенности создания программ в MasterSCADA 4D.

Список статей:

Ограничения использования функций и ФБ в задаче экрана

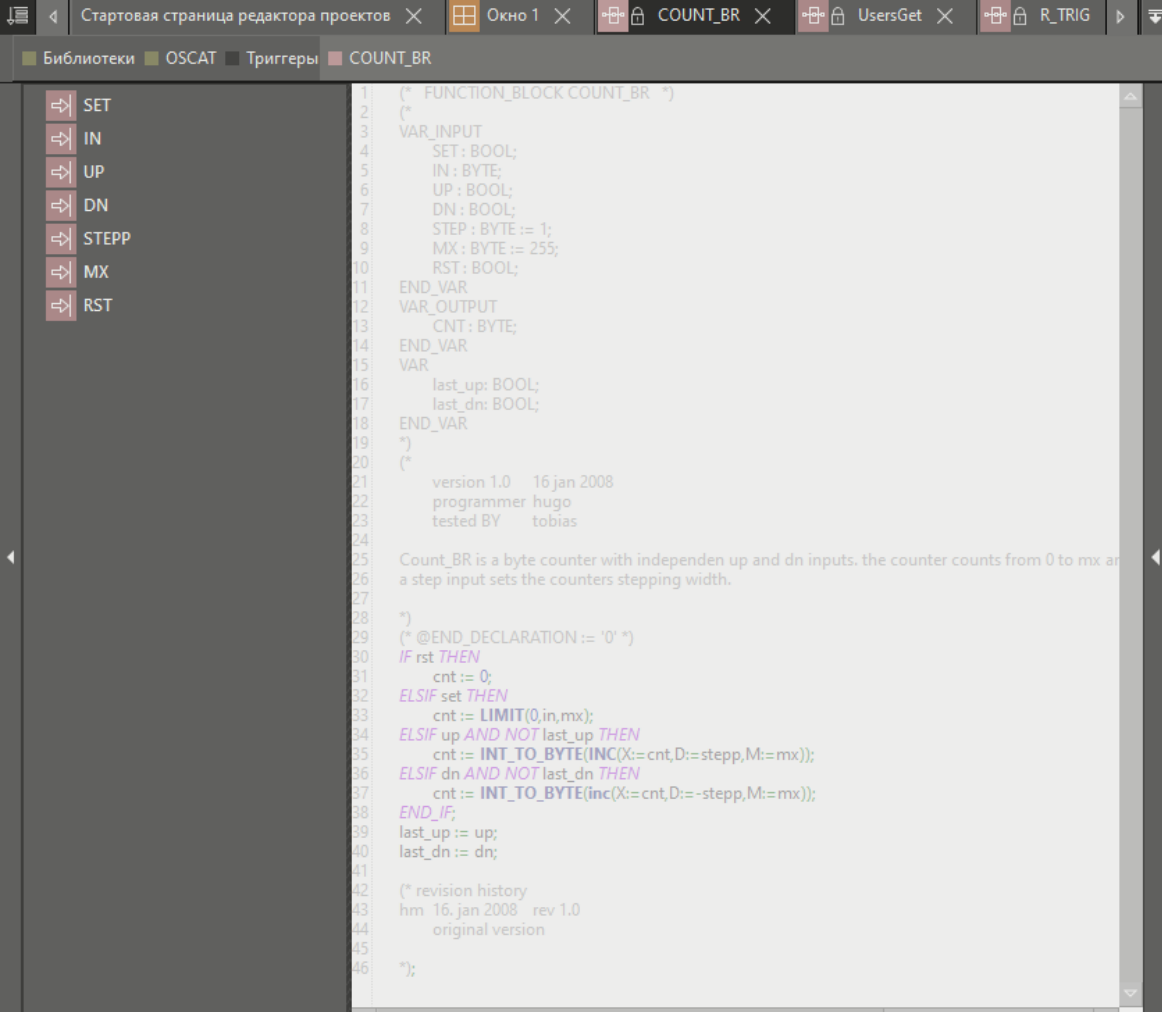
Работа с асинхронными ФБ

7.9.1. ОГРАНИЧЕНИЯ ИСПОЛЬЗОВАНИЯ ФУНКЦИЙ И ФБ В ЗАДАЧЕ ЭКРАНА

Работа не всех функций и функциональных блоков поддерживается в задаче экрана. Некоторые функции и функциональные блоки, которые используются в программах или самостоятельно в дереве проекта работать не будут, если их назначили в задачу экрана.

В общем случае в задаче экрана работать будут те элементы, у которых можно просмотреть код ST, который открывается по двойному клику по этому ФБ в центральной части редактора.

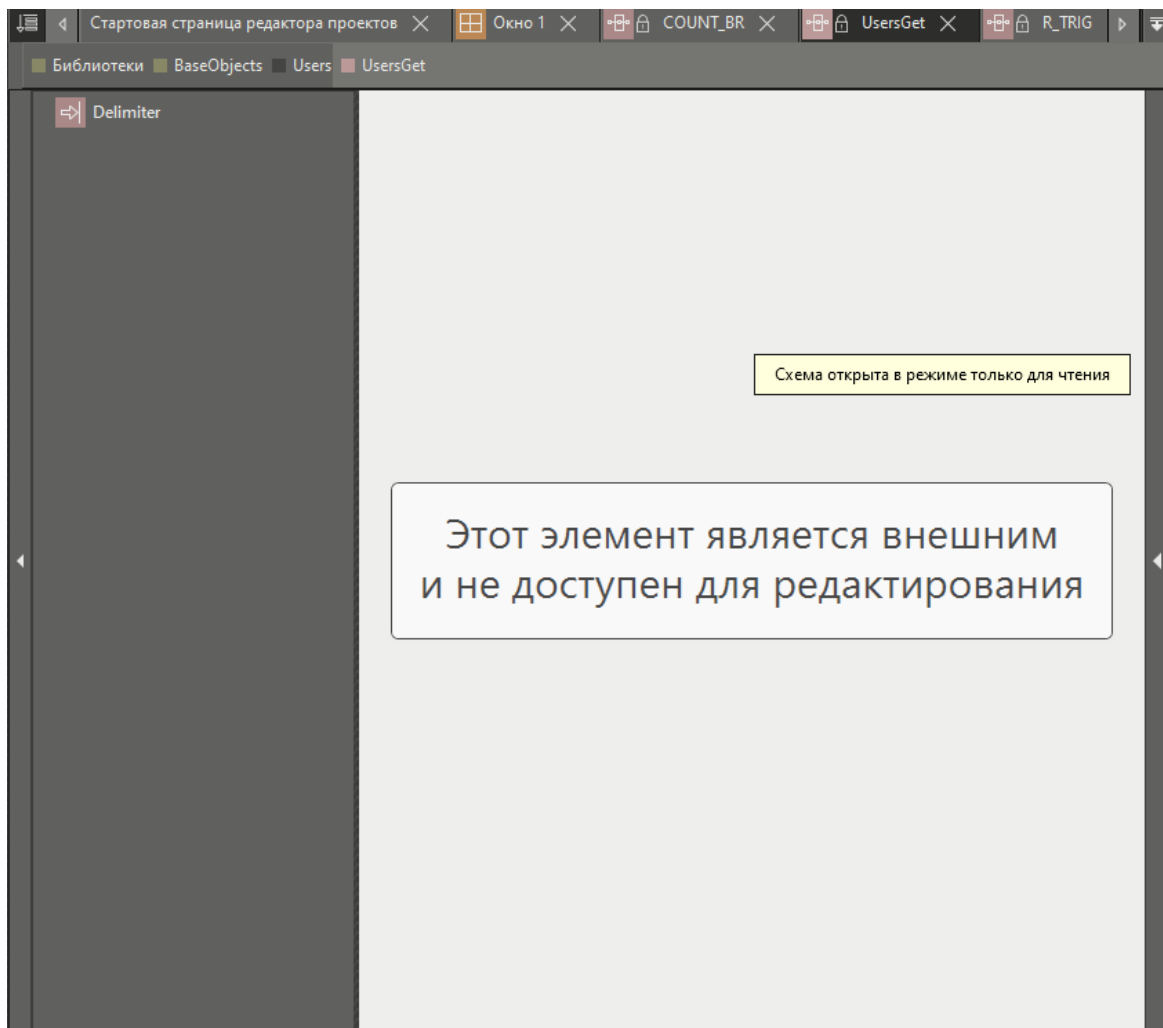
Пример такого ФБ:



```
1 (* FUNCTION_BLOCK COUNT_BR *)
2 (*
3 VAR_INPUT
4   SET : BOOL;
5   IN : BYTE;
6   UP : BOOL;
7   DN : BOOL;
8   STEP : BYTE := 1;
9   MX : BYTE := 255;
10  RST : BOOL;
11 END_VAR
12 VAR_OUTPUT
13   CNT : BYTE;
14 END_VAR
15 VAR
16   last_up : BOOL;
17   last_dn : BOOL;
18 END_VAR
19 *)
20 (*
21   version 1.0   16 jan 2008
22   programmer hugo
23   tested BY   tobias
24 *)
25 Count_BR is a byte counter with independent up and dn inputs. the counter counts from 0 to mx and
26 a step input sets the counters stepping width.
27 *)
28 (* @END_DECLARATION := '0' *)
29 IF rst THEN
30   cnt := 0;
31 ELSIF set THEN
32   cnt := LIMIT(0, in, mx);
33 ELSIF up AND NOT last_up THEN
34   cnt := INT_TO_BYTE(inc(X:=cnt, D:=stepp, M:=mx));
35 ELSIF dn AND NOT last_dn THEN
36   cnt := INT_TO_BYTE(inc(X:=cnt, D:=-stepp, M:=mx));
37 END_IF;
38 last_up := up;
39 last_dn := dn;
40 *)
41 (* revision history
42   hm 16. jan 2008 rev 1.0
43   original version
44 *)
45 *)
46 );
```

Если по двойному клику по функции или ФБ открывается вкладка без кода ST, то такой ФБ может не поддерживаться.

Вид вкладки ФБ, который возможно не будет поддерживаться в задаче экрана:



Такие функциональные блоки и функции имеют подключаемый код.

Поддерживаемые ФБ и функции в задаче экрана с подключаемым кодом

Поддерживается работа функциональных блоков и функций с подключаемым кодом, которые находятся в категории Стандартная библиотеки MasterSCADA 4D, за исключением функций CREATE_ARRAY, RESIZE_ARRAY из категории Функции работы с массивом

Из категории Общая (BaseObjects) в задаче экрана работают только ФБ с подключаемым кодом GET_CURRENT_USERNAME, PlaySound.

Элементы из категории Протоколы работать в задаче экрана не будет.

7.9.2. РАБОТА С АСИНХРОННЫМИ ФБ

Особенность асинхронных функциональных блоков в том, что они исполняются независимо от других задач. У подобных ФБ есть вход Run (или другой, подходящий по смыслу), который запускает его работу и выходы, которые сигнализируют о том, что ФБ завершил свою работу. Возможные специальные выходы асинхронных ФБ:

- Running/Building - принимает значение TRUE, пока выполняется запрошенная операция.
- Completed/Done - принимает значение TRUE на один такт, после того как ФБ завершил свою работу.
- State - меняет свое значение в зависимости от состояния ФБ

Асинхронные функциональные блоки могут обладать как обоими выходами, так и только одним из них. Если необходимо использовать результат работы блока на том же цикле, что и поступили данные на вход или запустить исполнение других программ после исполнения этого блока, нужно в программах использовать эти специальные выходы.

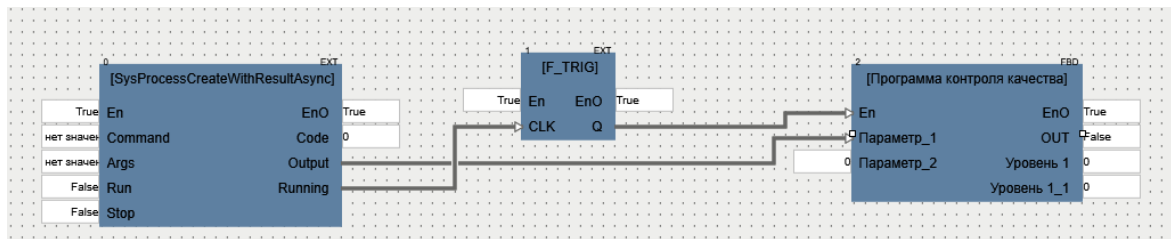
Примеры использования

Рассмотрим несколько примеров.

Способ исполнения - Периодически

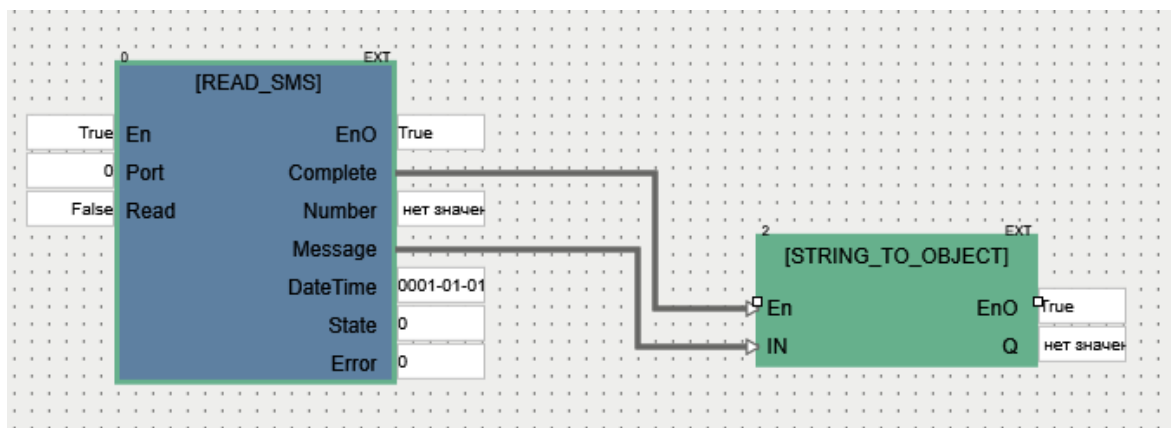
Чаще всего асинхронные ФБ используются в программах FBD, у которых в настройке Способ исполнения установлено значение Периодически. Рассмотрим примеры использования специальных выходов.

Выход Running:



В данном случае программа со всеми ФБ работает периодически (см. Способ исполнения), функциональный блок F_TRIG следит за задним фронтом выхода Running асинхронного блока SysProcessCreateWithResultAsync и после того, как он завершит свою программу будет вызвана другая программа проекта.

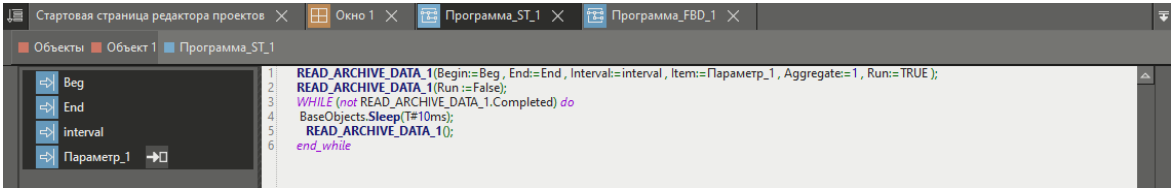
Выход Completed:



На схеме видно, что после того, как сообщение будет получено в блоке **READ_SMS** запустится функциональный блок **STRING_TO_OBJECT**

Способ исполнения - По вызову

Рассмотрим пример, когда функциональный блок необходимо включать в работу **по вызову** (см. **Способ исполнения**). Если необходимо получить результат вычисления на том же цикле, что и был запущен функциональный блок, то можно использовать следующий прием в программе ST.



```
1 READ_ARCHIVE_DATA_1(Begin:=Beg , End:=End , Interval:=interval , Item:=Параметр_1 , Aggregate:=1 , Run:=TRUE );
2 READ_ARCHIVE_DATA_1(Run := False);
3 WHILE (not READ_ARCHIVE_DATA_1.Completed) do
4   BaseObjects.Sleep(T# 10ms);
5   READ_ARCHIVE_DATA_1;
6 end_while
```

Рассмотрим порядок работы программы представленной на рисунке. Вызываем в программе функциональный блок - он начинает работать. В следующей строке, мы сбрасываем значение запускающего входа **RUN**. Далее до тех пор пока выход функционального блока **Completed** не примет значение **TRUE** на 10 миллисекунд система будет приостанавливаться, а затем ФБ будет вызываться снова для обновления его выходов. Если вызывать функциональный блок без задержки, то нагрузка на исполнительную систему будет значительно увеличена, что может привести к нежелательным последствиям.

Недостаток такого подхода в том, что при длительных операциях могут произойти большие задержки в работе программ, но для быстрых операций такой подход оправдан.

8. СОЗДАНИЕ ОКОН ДЛЯ КЛИЕНТА ВИЗУАЛИЗАЦИИ

Графический интерфейс оператора создается в редакторе HMI, который входит в состав среды разработки MasterSCADA 4D, в режиме исполнения созданные окна открываются в клиенте визуализации.

Разработчики MasterSCADA 4D стремятся достичь следующих целей:

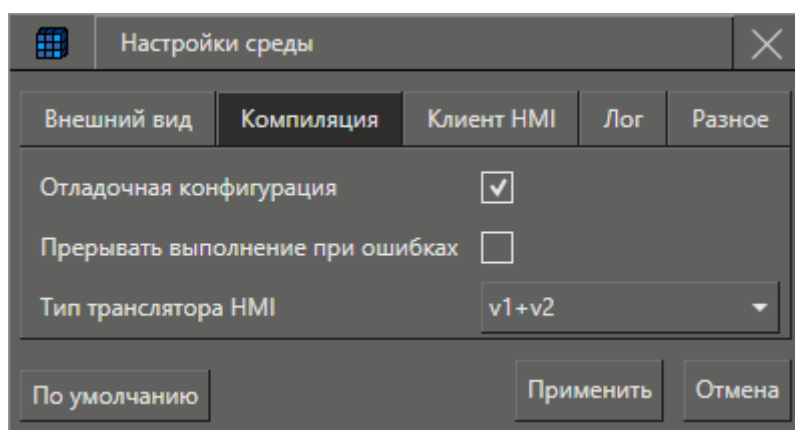
- В программе должны использоваться все возможные новые мировые тренды и технологии;
- Должен обеспечиваться максимально возможный функционал программы даже в самой узкоспециализированной ОС.

К сожалению, добиться этого одновременно или достичь разумного компромисса в настоящее время невозможно, поэтому было принято решение выпустить две версии транслятора HMI:

v1 - которая поддерживается во всех исполнительных системах и браузерах, но в которой, к сожалению, при этом многие современные функции не могут быть реализованы.

v2 - в которой используется более прогрессивная технология, но которая, по этой причине, поддерживается не во всех браузерах. При применении этой версии транслятора рекомендуется в качестве клиента визуализации использовать MasterSCADA 4D Client, а также браузеры Chrome и Firefox.

Выбрать версию, которая будет применяться для подготовки файлов, необходимых для работы клиента визуализации, можно в диалоговом окне Настройка среды во вкладке Компиляция:



В поле Тип транслятора HMI необходимо выбрать номер версии. Возможные варианты:

- v1 - при компиляции формируется только HMI v1;
- v1+v2 - при компиляции формируется обе версии HMI. По умолчанию, в клиенте открывается HMI v1, чтобы получить доступ к HMI v2 необходимо добавить /test

в адресной строке браузера или в настройках приложений MasterSCADA 4D Monitor и MasterSCADA 4D Client Monitor;

- v2 - при компиляции формируется только HMI v2.

Важно! Если MasterSCADA 4D была установлена впервые, то по умолчанию настройка будет v2. Если на компьютере была установлена версия, где использовался транслятор v1, то при установке новой версии появится диалоговое окно, содержащее следующий текст: "У Вас используется устаревший транслятор HMI v1. Перейти на новый транслятор HMI v2? Данную настройку можно изменить в настройках приложения на вкладке Компиляция." Если нажать кнопку Да, то установится значение v2, если Нет, то останется предыдущий вариант.

Различия окон, созданных с использованием разных типов трансляторов HMI:

Элемент/Действие	HMI v1	HMI v2
Элемент Док панель	Есть	Нет
Виртуальная клавиатура	Есть	Нет
Работа Шаблона экрана	Особая обработка элементов	Приведено к типу Контейнер
Действия Назад, Вперед в браузере	Работает, у окон есть свой URL, есть история переходов между окнами	Не работает, у окон есть URL, но в истории страница всегда одна
Действие Печать	Печать возможно только в случае, если окно было полностью загружено.	Печать только текущего открытого окна
Элемент Выбор цвета	Применяется сразу же	По кнопке Применить
Закрытие всплывающего окна	Закрытие только по крестик в углу.	Кроме нажатия на крестик в правом верхнем углу к закрытию приведет повторный вызов действия, которое открыло это окно. Т.е. если нажать на кнопку, которая открыла окно, то это приведет к его закрытию.

Элемент/Действие	HMI v1	HMI v2
Элемент График XY	Нет	Есть
Элемент Дерево множественного выбора	Нет	Есть
Элемент Дерево одиночного выбора	Нет	Есть
Добавление и удаление перьев в Тренде в RT	Нет	Есть
Масштабирование по оси Y	Нет	Есть
Минилегенда, панель инструментов, всплывающая подсказка в Тренде	Стандартная встроенная.	Собственная (html5)
Методы панели инструментов в Тренде	Доступны только в панели инструментов	Можно вызывать отдельно
Стрелочный прибор без образа	Не работает	Работает
Все всплывающие окна приведены к одному типу	Сообщение о потере связи с сервером, сообщение из действия Показать сообщение и всплывающие окна имеют разный вид	Есть
Модальность и неподвижность всплывающих окон	Нет	Есть
Восстановление связи с исполнительной системой	Только с перезагрузкой страницы	Без перезагрузки

Элемент/Действие	HMI v1	HMI v2
Права доступа	Принадлежит контролю	Принадлежит связи или действию
Автоматический логин на всех экземплярах встроенного браузера по первому входу	Нет	Есть
Количество динамизируемых элементов в каждом узле	Не более 1000	Без ограничений
Стыковка объемных элементов (трубы, задвижки и др)	Возможна нестыковка в режиме исполнения на 1-2 пикселя	Стыковка в соответствии с проектом

8.1. ОКНО

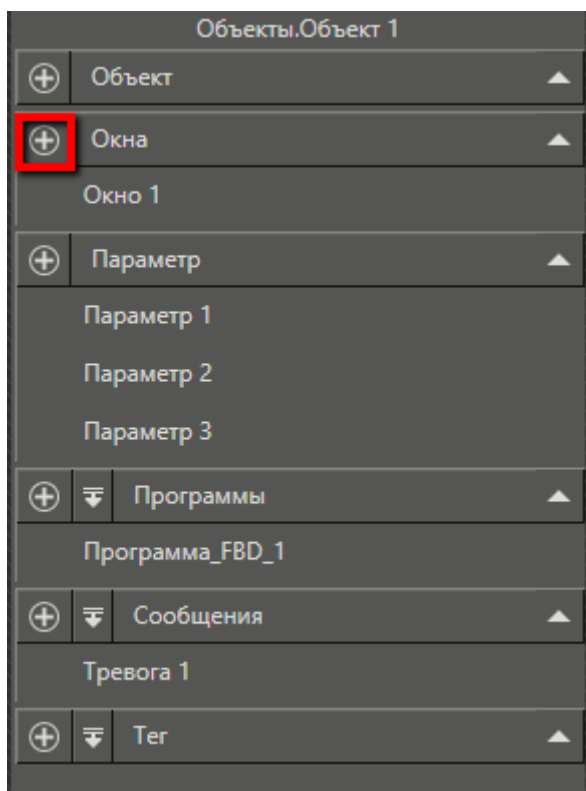
Окно – это элемент проекта. Оно может использоваться как самостоятельно, так и в качестве составной части другого окна.

Окна создаются в группах Ресурсы.Окна следующих элементов:

- объект;
- канал;
- тег.


Кроме того, они могут быть созданы в группе Окна у узлов.

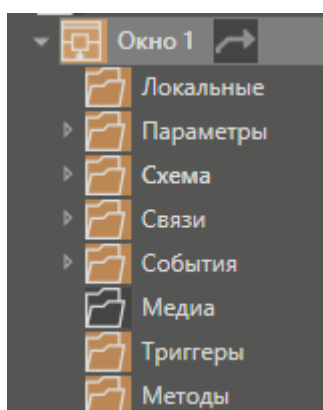
Уникальные окна, т.е. встречающиеся в проекте один раз, добавляются в контекстном меню родительского элемента при помощи команды Добавить.Окно или через контекстную панель:



Окна, которые встречаются в проекте многократно, необходимо типизировать, т.е. создать библиотечное окно, которое можно будет использовать в дальнейшем, перетаскивая его из библиотеки в нужное место проекта.

Вид элемента Окно в деревьях MasterSCADA 4D зависит от режима отображения дерева.

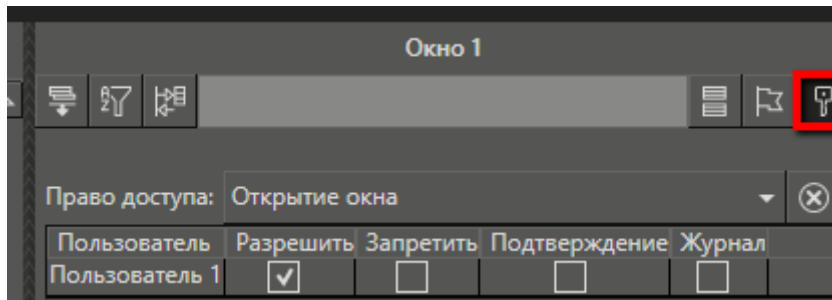
Если при разработке проекта используется упрощенное дерево, то окно представляется в виде простого элемента:  Окно 1. А в случае, если используется полное дерево, окно имеет вид:



Основная работа с окном осуществляется в редакторе НМІ, поэтому внутренняя структура окна в дереве проекта используется редко. Если полностью раскрыть внутреннюю структуру окна, то можно увидеть, что его составные части соответствуют одноименным вкладкам легенды редактора НМІ.

После добавления окна в проект, редактор НМІ откроется автоматически.

Если окно будет содержать различные элементы управления, то необходимо убедиться в том, что в нем корректно назначены права доступа пользователей. Если для окна требуется настроить специальные права доступа, отличные от тех, которые заданы у родительского элемента, то необходимо отметить или снять соответствующие флаги в панели свойств окна:



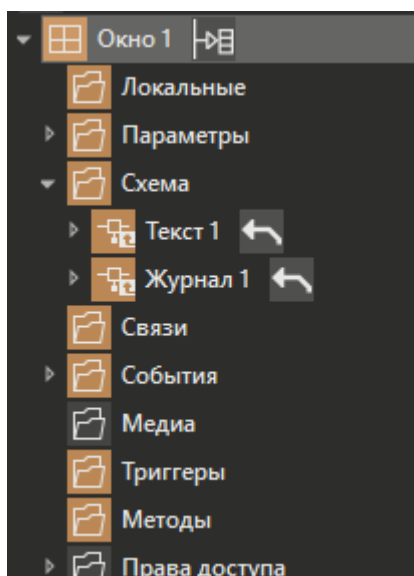
В этом случае появится дополнительная группа Права доступа в структуре окна в дереве.

8.1.1. СТРУКТУРА ОКНА

Разработка графического интерфейса проекта основана на том, что один элемент Окно может быть вставлен в другой элемент Окно. Для упрощения описания, элемент Окно, используемый в качестве графического интерфейса в клиенте визуализации в режиме исполнения, называется мнемосхема, а вставляемый в него элемент Окно называется графический элемент. В этом смысле мнемосхема создается методом размещения на ней графических элементов, как библиотечных, так и созданных пользователем. Пользовательский графический элемент создается в библиотеке таким же образом как мнемосхема, т.е. представляет собой комбинацию нескольких библиотечных графических элементов.

Мнемосхема, вставленная в другую мнемосхему, становится графическим элементом.


элемент Окно имеет predetermined structure, которая видна только в полном дереве:



Назначение групп:

Название	Назначение
Локальные	Группа содержит локальные параметры окна, используемые для обеспечения внутренней логики работы окна. Соответствует вкладке легенды Локальные.
Параметры	Содержит параметры окна. Соответствует вкладке легенды Параметры.
Схема	Содержит элементы, которые находятся в окне. Каждый элемент может иметь внутри себя дополнительные группы, которые применимы для того или иного элемента. Соответствует вкладке легенды Элементы.
Связи	Содержит связи между элементами окна и параметрами проекта. Соответствует вкладке легенды Связи. Если дважды кликнуть левой кнопкой мыши по какому-либо элементу группы, то откроется соответствующий этому элементу Конвертер значений.
События	Содержит настроенные события для окна. Соответствует вкладке легенды События.
Медиа	Показывает медиа, добавленные в окно. Соответствует вкладке легенды Медиа.
Триггеры	Показывает триггеры, добавленные в окно. Соответствует вкладке легенд Триггеры.
Методы	В текущей версии не поддерживается.
Права доступа	Эта группа появляется в случае, если для окна были настроены права доступа.

8.1.2. СВОЙСТВА ОКНА

Вид панели свойств элемента Окно при нажатой и отжатой кнопке :

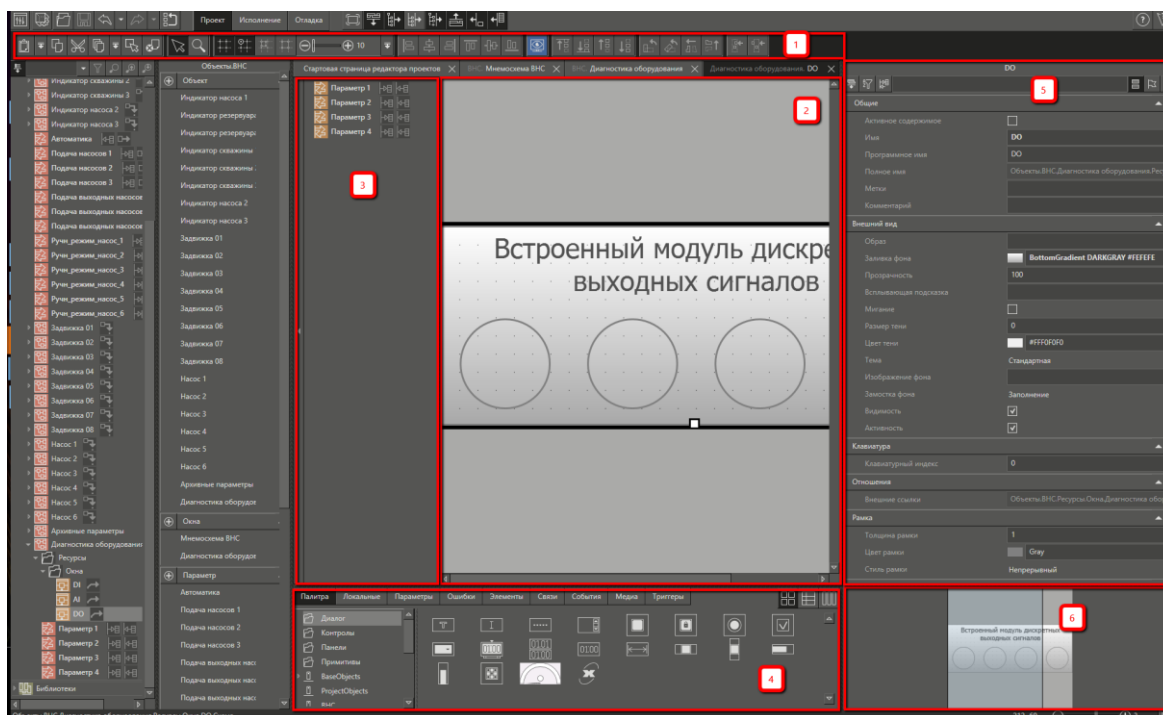
Простой режим	Полный режим
	
Название	Рекомендации
Категория Общие	Как правило, в этой категории настраивается только свойство Имя. Подробное описание данной категории приведено в соответствующем разделе.
Категория Внешний вид	В этой категории чаще всего настраивается свойство Заливка фона. Для увеличения скорости загрузки окон рекомендуется настроить свойство Прозрачность (см. Кеширование окон). Подробное описание данной категории приведено в соответствующем разделе.
Категория Рамка	Эта категория свойств настраивается в том случае, если окно является частью другого окна. Подробное описание данной категории приведено в соответствующем разделе.
Категория Расположение	В этой категории чаще всего настраиваются свойства Ширина и Высота. MasterSCADA 4D поддерживает автоматическое изменение размеров окон в зависимости от размеров окна клиента визуализации, но желательно, чтобы при разработке были указаны размеры, максимально близкие к размерам тех мониторов, которые будут применяться. Это позволит наиболее эффективно

Название	Рекомендации
	использовать всю площадь экрана. Подробное описание данной категории приведено в соответствующем разделе.
Категория Трансформация	Подробное описание этой категории приведено в соответствующем разделе.
Категория Клавиатура	Подробное описание этой категории приведено в соответствующем разделе.
Категория Отношения	В этой категории отображается окно, наследником или экземпляром которого является настраиваемое окно. Для того чтобы изменить это свойство, необходимо воспользоваться пунктом контекстного меню элемента Сменить тип.
Категория Экран	Подробное описание этой категории приведено в соответствующем разделе.

8.2. РЕДАКТОР НМІ

При создании элемента Окно редактор НМІ открывается автоматически в виде вкладки в рабочей области среды разработки, либо двойным нажатием левой кнопкой мыши на уже созданный элемент Окно в дереве. Одновременно может быть открыто несколько окон (вкладок).

Интерфейс редактора НМІ выглядит следующим образом:



Интерфейс редактора HMI состоит из следующих областей:

- 1 – панель инструментов;
- 2 – рабочая область редактора HMI - в этой рабочей области создается внешний вид окна (мнемосхемы), в котором оно будет отображаться в клиенте визуализации в режиме исполнения ;
- 3 – клеммники;
- 4 – легенда;
- 5 – панель свойств;
- 6 – миникарта - полное уменьшенное изображение окна (мнемосхемы), на котором прямоугольной рамкой выделена зона, отображаемая в рабочей области редактора HMI . Путем перемещения рамки по миникарте выбирается требуемая для редактирования зона. Используется для работы с мнемосхемами, размеры которых превышают размеры рабочей области редактора HMI).








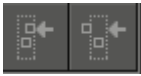
8.2.1. ПАНЕЛЬ ИНСТРУМЕНТОВ РЕДАКТОРА HMI

При открытии редактора HMI, на вкладку инструментов Проект добавляется ряд кнопок, необходимых для работы с элементами, расположенными в рабочей области редактора HMI.

Панель инструментов имеет вид :



Назначение кнопок:

Вид	Название панели и ссылка на описание
	Режимы (доступны только инструменты Выделение рамкой и Масштабирование рамкой)
	Сетка
	Выравнивание
	Уравнять
	Просмотр
	Z-порядок
	Поворот
	Группа

Все доступные инструменты/команды всех панелей, кроме панели Режимы, отображаются также в контекстных меню редактора НМІ или выделенных графических элементов. Команды контекстного меню, отсутствующие в меню редактора НМІ, описаны в разделе Контекстное меню НМІ.

8.2.1.1. ПАНЕЛЬ ГРУППА

В редакторе НМІ панель имеет вид :



Панель содержит следующие кнопки:

Вид	Название	Назначение
• ■	Поместить в группу	Выделенные элементы объединяются в группу (группируются) и создается графический элемент Панель, содержащий эти элементы (свойство Активное содержимое = False; см. также Общие инструменты панелей).
• ■	Вынести из группы	Элементы, выделенные внутри панели, выносятся из группы. В случае выделения панели или всех элементов в ней, элементы полностью разгруппировываются, и графический элемент Панель удаляется.

8.2.1.2. ПАНЕЛЬ ПРОСМОТР

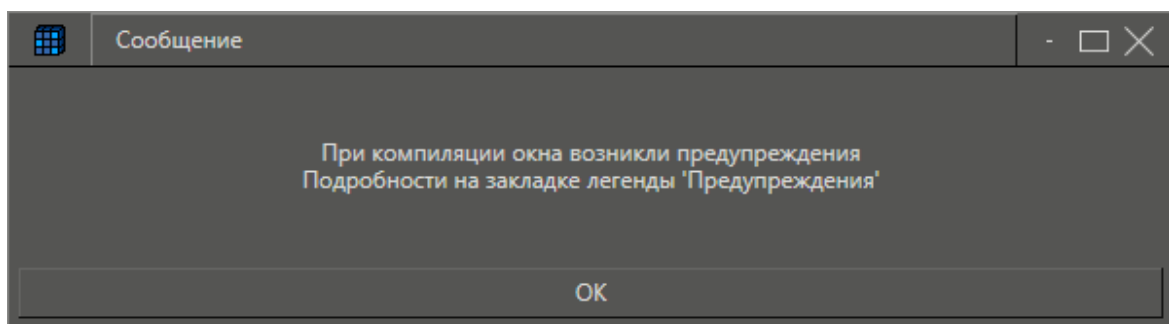
При нажатии на кнопку Просмотр панели инструментов редактора НМІ появляется возможность без запуска среды исполнения предварительно оценить внешний вид, который окно (мнемосхема) будет иметь в клиенте визуализации.



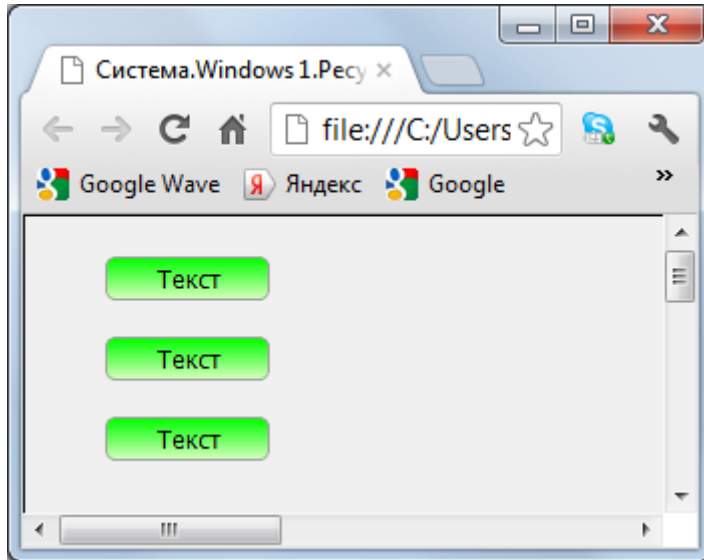
По команде  выполняются следующие действия:

- мнемосхема в виде файла *index.html* и файлов ресурсов сохраняется в папке *%TEMP%/MasterSCADA4D*;
- выполняется проверка мнемосхемы.

При наличии предупреждений появляется соответствующее сообщение, а подробную информацию отображает Легенда НМІ. Вкладка Предупреждения:



- мнемосхема (файл *index.html*) в статическом режиме открывается в браузере по умолчанию:



Графические элементы в данном режиме могут отображаться в упрощенном виде, особенно такие сложные, как тренд.




В данном режиме мнемосхема отображается без масштабирования, т.е. размеры и взаиморасположение графических элементов мнемосхемы соответствуют исходно заданным и не изменяются при изменении размеров окна браузера. Тем не менее, при этом сохраняется возможность масштабирования мнемосхемы в браузере, т.к. свойство Запрет масштабирования для предварительного просмотра всегда передается как False.


8.2.1.3. ПАНЕЛЬ Z-ПОРЯДОК

В редакторе НМІ панель имеет вид:



Панель содержит следующие кнопки:

Вид	Название	Назначение
	Переместить наверх	Перемещает выделенные элементы на передний план.
	Переместить вниз	Перемещает выделенные элементы на задний план.
	Переместить на шаг вверх	Перемещает выделенные элементы на шаг вверх по оси Z.





Вид	Название	Назначение
	Переместить на шаг вниз	Перемещает выделенные элементы на шаг вниз по оси Z.

8.2.1.4. ПАНЕЛЬ ПОВОРОТ

В редакторе НМІ панель имеет вид:



Панель содержит следующие кнопки:





Вид	Название	Назначение
	Повернуть на 90°	Поворачивает выделенные элементы на 90° по часовой стрелке.
	Повернуть на 45°	Поворачивает выделенные элементы на 45° по часовой стрелке.
	Отразить горизонтально	Переворачивает выделенные элементы вокруг вертикальной оси.
	Отразить вертикально	Переворачивает выделенные элементы вокруг горизонтальной оси.

8.2.1.5. ПАНЕЛЬ УРАВНЯТЬ

В редакторе НМІ панель имеет вид :



Панель содержит следующие кнопки:

Вид	Название	Назначение
	Уравнять ширину	<p>Если нажать на левую кнопку, то ширина всех выделенных элементов будет равна тому, который был выделен первым.</p> <p>Если нажать на кнопку , то откроется окно, в котором можно указать точное значение ширины, которое применится для всех выделенных элементов.</p>
	Уравнять высоту	<p>Если нажать на левую кнопку, то высота всех выделенных элементов будет равна тому, который был выделен первым.</p> <p>Если нажать на кнопку , то откроется окно, в котором можно указать точное значение высоты, которое применится для всех выделенных элементов.</p>

Кнопки панели становятся активны, если выделено больше одного графического элемента.

8.2.2. НМІ. РАЗМЕЩЕНИЕ ЭЛЕМЕНТОВ

Графические элементы, которые можно использовать для создания окон, находятся в легенде редактора НМІ (см. Легенда НМІ. Вкладка Палитра). Допускается также использовать элементы дерева, которые имеют predetermined графическое отображение.

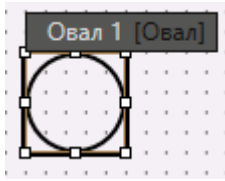
Список элементов дерева, имеющих графическое отображение:

- Объекты, содержащие окна или параметры
- Окна
- Теги
- Каналы
- Параметры

Для того чтобы разместить графический элемент в рабочей области редактора НМІ, необходимо выполнить перетаскивание его из легенды или из деревьев MasterSCADA 4D в нужное место рабочей области.

Графический элемент, уже находящийся в рабочей области, после выделения может быть перемещен на нужное место левой кнопкой мыши или клавишами курсора.

Если элемент выделен в окне, то отображается ограничивающий его прямоугольник (толщина линии – 1 px):

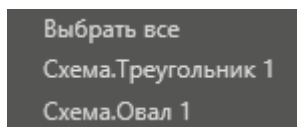


Установить произвольный размер можно потянув за узлы рамки левой кнопкой мыши. Если нажать клавишу CTRL и потянуть за углы рамки левой кнопкой мыши, то фигура будет изменять свои размеры пропорционально.

Над выделенным элементом отображается окно, содержащее имя элемента и его тип (Овал 1 на картинке выше).

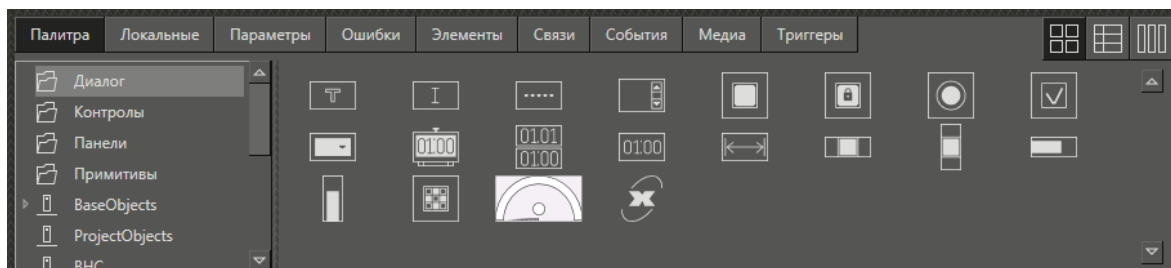
Для работы с элементами рекомендуется использовать контекстное меню элементов или группы элементов, а также панель инструментов.

Если несколько элементов перекрывают друг друга, то при нажатии левой кнопкой мыши на область перекрытия открывается меню выбора выделяемого элемента:



8.2.2.1.1. ЛЕГЕНДА В РЕДАКТОРЕ НМІ

Если в окне документов и редакторов активен редактор НМІ, то легенда имеет вид:



Легенда содержит следующие вкладки:

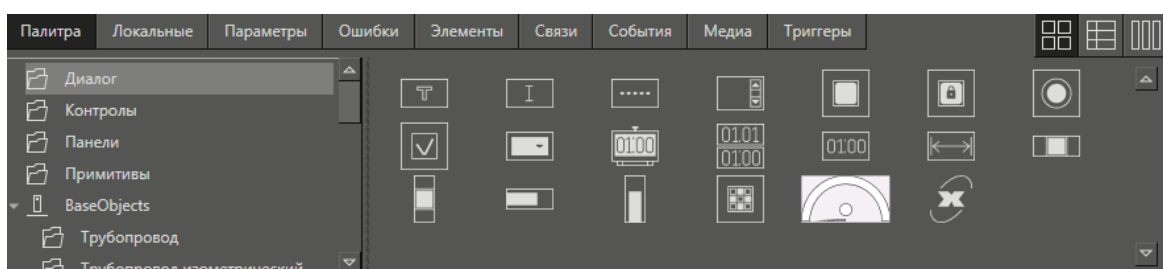
- Легенда НМІ. Вкладка Палитра;
- Легенда НМІ. Вкладка Локальные;
- Легенда НМІ. Вкладка Параметры;
- Легенда НМІ. Вкладка Ошибки
- Легенда НМІ. Вкладка Элементы;
- Легенда НМІ. Вкладка Связи;
- Легенда НМІ. Вкладка События;
- Легенда НМІ. Вкладка Медиа;
- Легенда НМІ. Вкладка Триггеры;

Видимостью легенды управляет переключатель Легенда (см. Панель управления видимостью окон).

8.2.2.1.2. ЛЕГЕНДА НМІ. ВКЛАДКА ПАЛИТРА

На этой вкладке отображаются встроенные графические элементы, находящиеся в библиотеке (например, НМІ , BaseObjects, Images и др.), и которые можно разместить в рабочей области редактора НМІ.

В левой части вкладки отображаются категории графических элементов, а в правой - элементы, которые могут быть перетащены в окно.

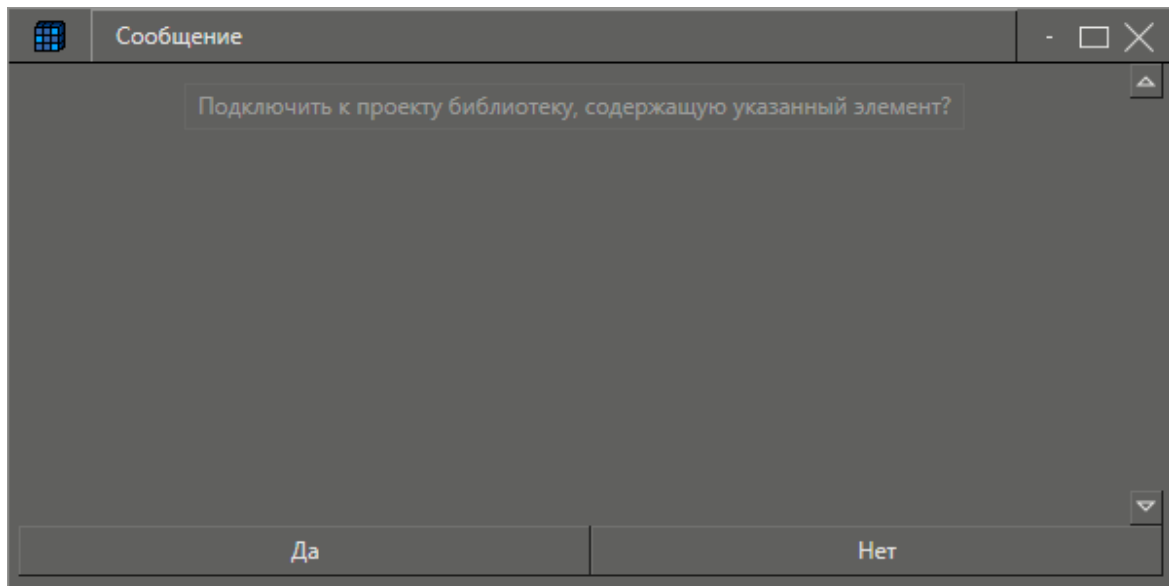


По умолчанию к проекту подключены не все имеющиеся библиотеки. Однако, все имеющиеся библиотеки и их содержимое будет отображаться на вкладке Палитра более бледным шрифтом.

Вид неподключенной библиотеки в палитре:



Если перетащить элемент из такой библиотеки в рабочую область графического редактора, то появится сообщение:



Если нажать кнопку Да, то библиотека будет подключена. И в дальнейшем ее элементы можно использовать обычным образом.

Способ отображения элементов в палитре

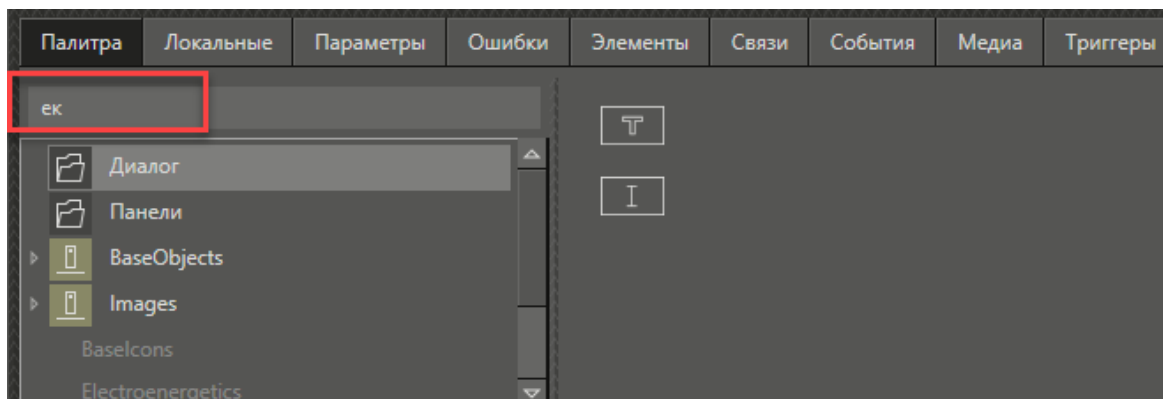
В палитре можно выбрать способ отображения элементов в палитре при помощи кнопок



- Вид списком - элементы располагаются друг за другом слева направо, сверху вниз. При наведении на имя элемента отображается путь к нему в библиотеке и комментарий.
- Табличный вид - элементы располагаются в таблице, где в первом столбце отображается имя, а во втором комментарий. При наведении на имя элемента отображается путь к нему в библиотеке и комментарий
- Вид колонок - элементы располагаются друг за другом сверху вниз слева направо. При наведении на имя элемента отображается путь к нему в библиотеке и комментарий

Фильтрация элементов в палитре

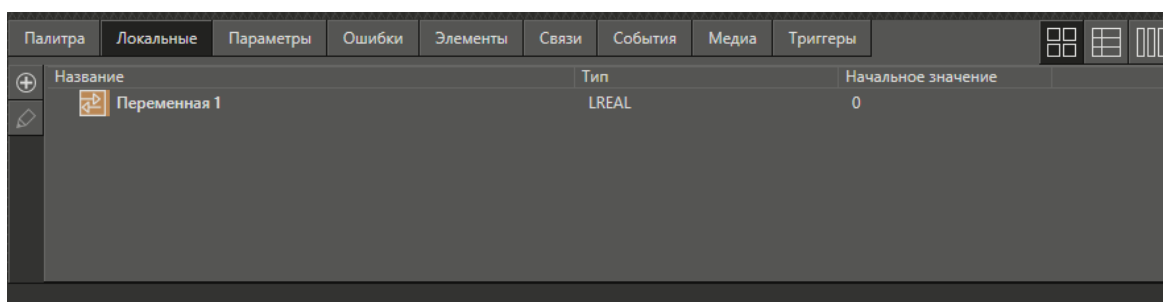
В палитре можно отфильтровать элементы по имени. Если известно имя элемента или часть имени, то можно ввести символы в строке:



В этом случае останутся только те категории и элементы, в которых содержится данная последовательность символов.

8.2.2.1.3. ЛЕГЕНДА HMI. ВКЛАДКА ЛОКАЛЬНЫЕ

На этой вкладке отображается редактор локальных переменных окна:



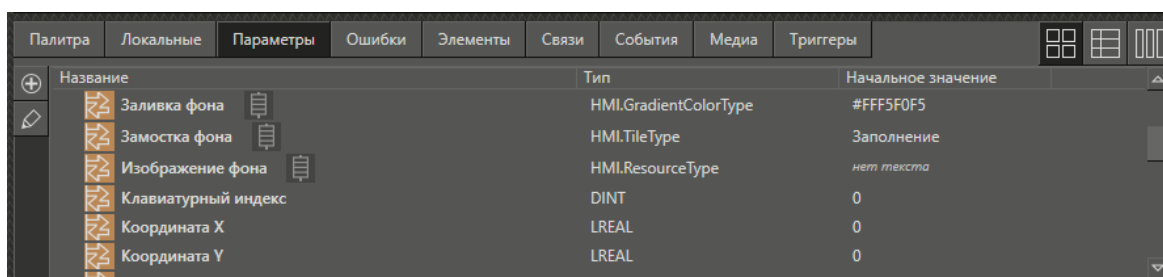
Редактор выполнен так же, как редактор локальных переменных программы (см. Легенда. Вкладка Локальные), но содержит только столбцы Название, Тип и Начальное значение.

Локальные переменные окна используются для обеспечения внутренней логики его работы.

Важно! Локальные переменные окон не доступны в их экземплярах. Как правило, локальные переменные окон используются только для обеспечения внутренних задач. Если необходимо настраивать связь между параметром, который рассчитывается в окне и другими параметрами других окон или параметрами сервера, то следует использовать клеммники окна.

8.2.2.1.4. ЛЕГЕНДА HMI. ВКЛАДКА ПАРАМЕТРЫ

На этой вкладке отображается редактор Параметров окна:



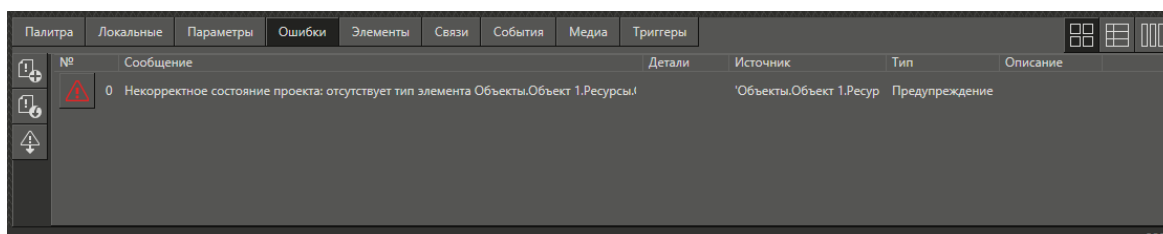
Т.е. в данной вкладке перечислены все стандартные параметры окна, значения которых могут быть изменены в режиме исполнения, а также параметры, созданные пользователем в окне клеммников

Редактор параметров выполнен так же, как редактор локальных переменных программы (см. Легенда. Вкладка Локальные), но содержит только столбцы Название, Тип и Начальное значение.

Параметры, добавленные в данной вкладке, появляются также в панели клеммников.


8.2.2.1.5. ЛЕГЕНДА НМІ. ВКЛАДКА ОШИБКИ

На этой вкладке отображаются Ошибки проекта. Вкладка содержит Ошибки только текущей мнемосхемы (см. Предварительный просмотр мнемосхемы и Команда проверки проекта):



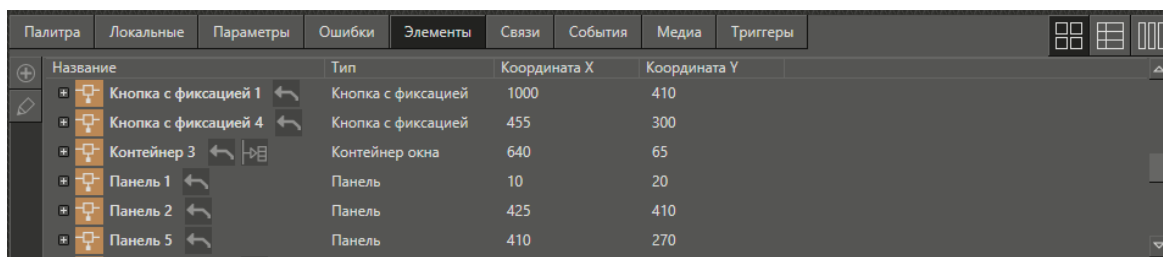
В отличие от таблицы, в окне ошибок строки на вкладке ошибок не имеют контекстного меню.

В случае затруднений с самостоятельным выявлением причин возникших ошибок, имеется

возможность при помощи кнопки  сформировать отчет об ошибках для последующей отправки его в службу технической поддержки.

8.2.2.1.6. ЛЕГЕНДА НМІ. ВКЛАДКА ЭЛЕМЕНТЫ

На этой вкладке отображается список графических элементов окна.

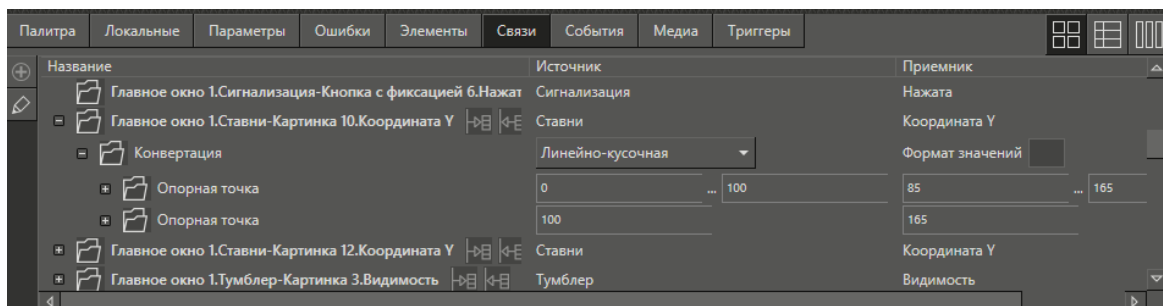


Элементы в списке можно отсортировать, нажав левой кнопкой мыши на заголовок какого-либо столбца.

Элемент, выделенный в панели, выделяется в рабочей области редактора НМІ, а в панели свойств отображается свойство выделенного элемента.

8.2.2.1.7. ЛЕГЕНДА НМІ. ВКЛАДКА СВЯЗИ

На этой вкладке отображаются все Связи элементов окна, используемые для динамизации. Кроме того, на ней можно отредактировать существующие Связи.

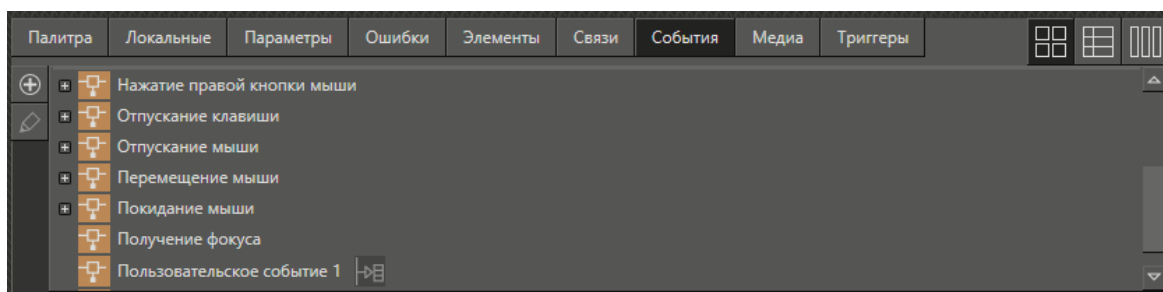


Связь редактируется таким же образом как и в панели свойств: сначала определяется тип конвертации, а затем указываются опорные точки, для которых задается зависимость между значением источника и приемника.

Если дважды кликнуть левой кнопкой мыши по названию какого-либо свойства или названию его дочернего элемента Конвертация, то откроется соответствующий этому элементу Конвертер значений.

8.2.2.1.8. ЛЕГЕНДА НМІ. ВКЛАДКА СОБЫТИЯ

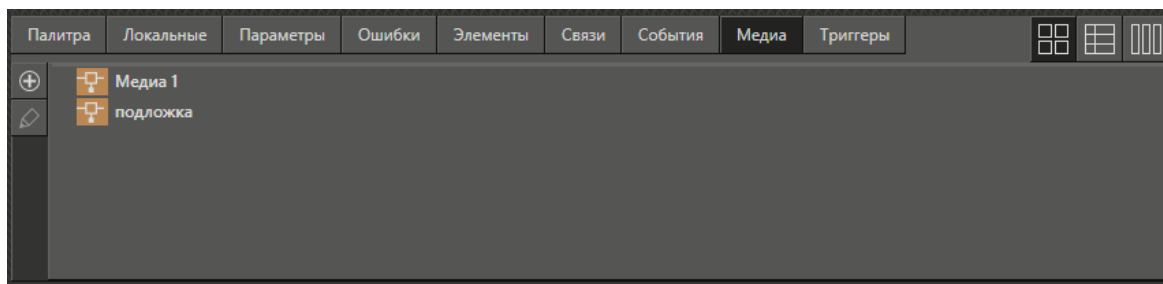
На этой вкладке отображается редактор группы событий окна:



На данной вкладке перечислены стандартные события окна. Кроме того, на ней имеется возможность создать пользовательские события. Пользовательские события окон необходимы при создании библиотечных окон.

8.2.2.1.9. ЛЕГЕНДА НМІ. ВКЛАДКА МЕДИА

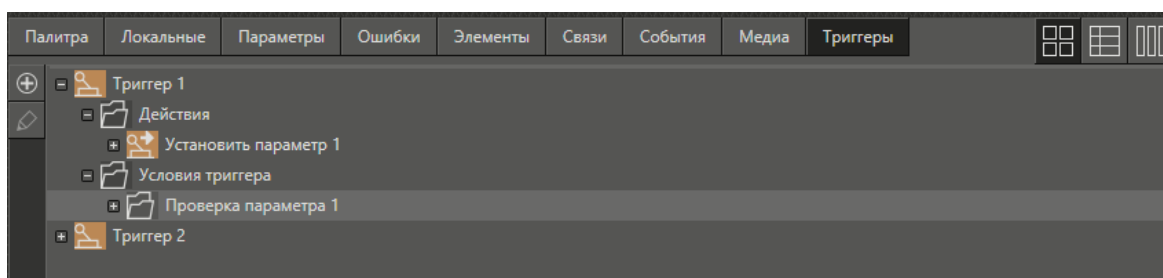
На этой вкладке отображаются медиа, которые уже используются в окне, например, в элементе Картинка. Кроме того, вкладка дает возможность добавлять медиа-файлы, которые еще только планируется задействовать:



Если выполнить двойное нажатие левой кнопкой мыши на элемент, то откроется диалоговое окно выбора медиа-ресурса, позволяющее выбрать графический медиа-файл для дальнейшего использования или замены уже используемого.

8.2.2.1.10.ЛЕГЕНДА НМІ. ВКЛАДКА ТРИГГЕРЫ

На этой вкладке отображается редактор группы триггеров окна

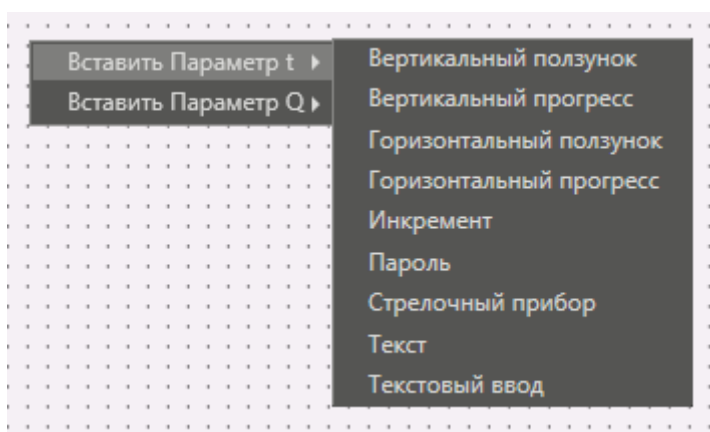


8.2.2.2. РАЗМЕЩЕНИЕ ТЕГА, КАНАЛА, ОБЪЕКТА В ОКНЕ

Перетащить данные элементы в окно можно непосредственно из дерева MasterSCADA 4D, при этом образуется графический элемент, имеющий все необходимые связи для динамической работы в режиме исполнения.

Тег/канал/объект не содержит окон, перетаскивание правой кнопкой мыши

Если в теге/канале/объекте отсутствуют собственные окна, то при перетаскивании такого тега/канала/объекта в окно правой кнопкой мыши открывается меню выбора графического элемента. Например, если объект содержит в себе два параметра, то после перетаскивания получим:



Выбранный для параметра элемент вставляется в окно, и при этом образуется двунаправленная связь параметр<=>свойство элемента управления по умолчанию.

Набор элементов в меню зависит от типа данных параметров, входящих в состав тега/канала/объекта. На примере выше показан список элементов для параметра с типом данных LREAL.

Тег/канал/объект не содержит окон, перетаскивание левой кнопкой мыши

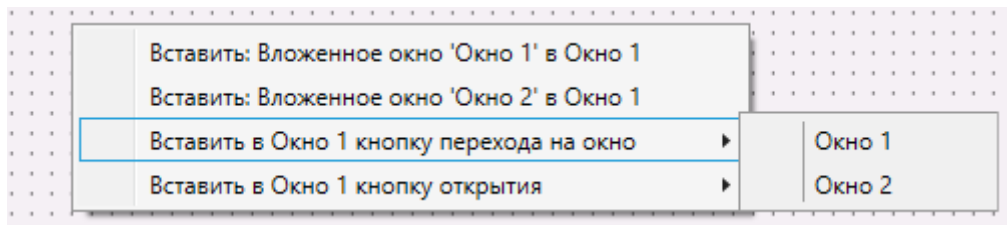
При перетаскивании левой кнопкой мыши в окно тега/канала/объекта, не содержащего собственных окон, для первого параметра такого тега/канала/объекта создается соответствующий графический элемент, в зависимости от типа данных параметра (см. Перетаскивание параметра в окно левой кнопкой мыши).

Тег/канал/объект содержит окна, перетаскивание левой кнопкой мыши

При перетаскивании левой кнопкой мыши тега/канала/объекта в окно, в него вставляется первое окно тега/канала/объекта.

Тег/канал/объект содержит окна, перетаскивание правой кнопкой мыши

При перетаскивании тега/канала/объекта в окно правой кнопкой мыши, открывается меню, которое обеспечивает вставку в него окон тега/канала/объекта, кнопок их открытия или кнопок перехода на них.



Вставить вложенное окно

По этой команде в окно вставляется соответствующее окно тега/канала/объекта, т.е. новый элемент будет динамически изменяемой частью другого окна.

Важно! В меню не формируются команды вставки окон в том случае, если эти окна в теге/канале/объекте созданы в папках.

Вставить кнопку перехода на окно

По этой команде в окно вставляется кнопка, для которой автоматически задается событие Клик мыши и действие Открыть окно .

Вставить кнопку открытия окна

По этой команде в окно вставляется кнопка, для которой автоматически задается Событие Клик мыши и Действие Открыть всплывающее окно .

8.2.2.3. РАЗМЕЩЕНИЕ ПАРАМЕТРА В ОКНЕ

При перетаскивании параметра в окно из дерева MasterSCADA 4D создается графический элемент, который имеет все необходимые связи с элементом дерева. Благодаря этому, создание окон занимает кратчайшее время.

Возможны способы перетаскивания параметра в окно указателем мыши с нажатием и удержанием как левой, так и правой кнопки мыши.

Эти способы определены как перетаскивание параметра в окно левой кнопкой мыши и как перетаскивание параметра в окно правой кнопкой мыши, соответственно.

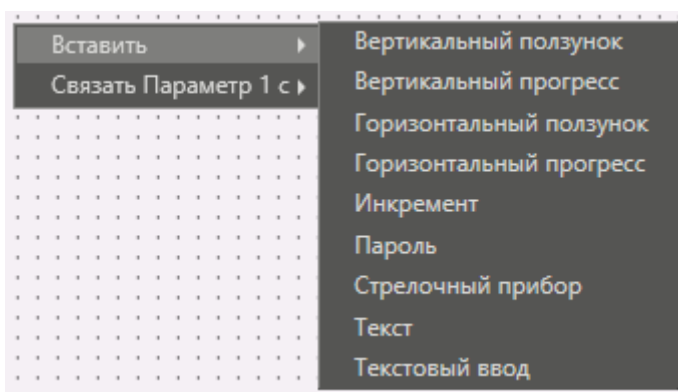
8.2.2.3.1. ПЕРЕТАСКИВАНИЕ ПАРАМЕТРА В ОКНО ПРАВОЙ КНОПКОЙ МЫШИ

При перетаскивании параметра в окно правой кнопкой мыши открываются меню Вставить и Связать:

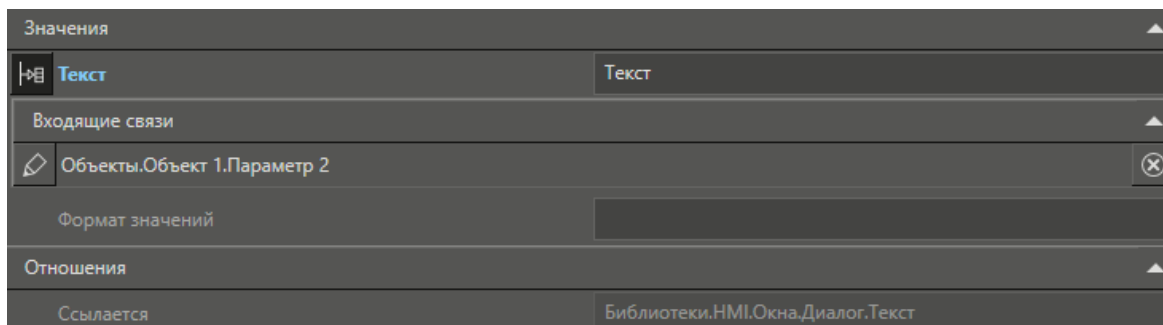
Меню Вставить

Это меню содержит команды вставки графического элемента в окно. В зависимости от выбранного элемента, при вставке может автоматически задаваться связь и начальное значение для свойства, у которого Свойство по умолчанию имеет значение TRUE, а также могут автоматически конфигурироваться другие свойства элемента.

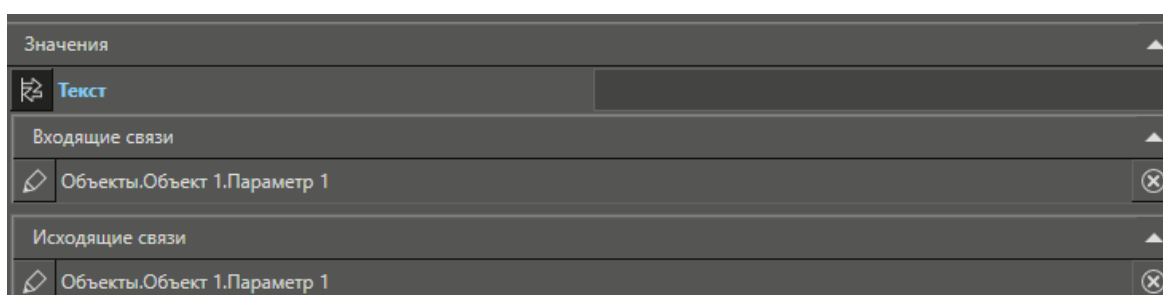
Набор элементов, доступных для вставки, зависит от типа данных перетаскиваемого параметра.



Если добавляется элемент, который допускает только получение данных в окне, то образуется входящая связь между свойством и параметром дерева. Например, для элемента Текст:



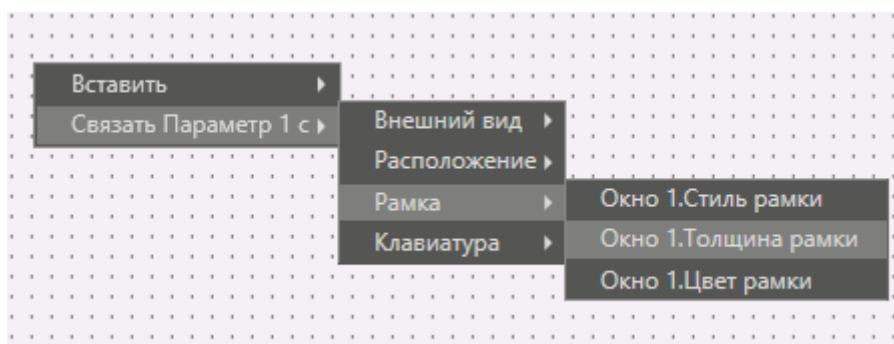
Для элементов типа Кнопка, Текстовый ввод, которые могут не только отображать полученные значения из дерева, но и изменять их, автоматически создаются две связи: входящая и исходящая:



Разработчик проекта может отредактировать автоматически созданную связь, либо удалить ее, нажав на соответствующие элементы управления.

Меню Связать

Это меню содержит команды связывания параметра со свойством графического элемента, на который выполнено перетаскивание параметра правой кнопкой мыши, либо, если ни один элемент не находится в зоне перетаскивания, то со свойствами самого окна:



Таким образом, при изменении значения параметра будет меняться и значение выбранного свойства в режиме исполнения.

8.2.2.3.2. ПЕРЕТАСКИВАНИЕ ПАРАМЕТРА В ОКНО ЛЕВОЙ КНОПКОЙ МЫШИ

Перетаскивание на пустое место окна

При перетаскивании параметра в окно левой кнопкой мыши создается соответствующий графический элемент в зависимости от типа данных параметра:

- **BOOL** – графический элемент Кнопка , настроенный так, чтобы при удержании кнопки в нажатом положении значение параметра было TRUE, а во всех остальных случаях – False, т.е. создается кнопка без фиксации;
- **DT** – графический элемент ДатаВремя
- **TIME** – графический элемент Интервал (если свойство Доступ параметра имеет значение INPUT, то создаются прямая и обратная связи, а если OUTPUT, то только прямая);
- **TOD** – графический элемент Время (если свойство Доступ параметра имеет значение INPUT, то создаются прямая и обратная связи, а если OUTPUT, то только прямая);
- все остальные типы данных – графический элемент Текст.

Перетаскивание на ранее созданный элемент

При обычном перетаскивании параметра, у которого свойство Доступ имеет значение INPUT или INOUT, на ранее созданный графический элемент автоматически выполняются следующие действия:

- на графические элементы Текст, Кнопка , Горизонтальный прогресс, Вертикальный прогресс, Кнопка выбора цвета, Картинка, Видео – задание однонаправленной связи параметр=>свойство по умолчанию (свойство по умолчанию – это главное свойство элемента);
- на графические элементы Текстовый ввод, Пароль, Инкремент, Кнопка с фиксацией, Радиокнопка, Флаг, ДатаВремя, Время, Интервал , Горизонтальный ползунок, Вертикальный ползунок – задание двунаправленной связи параметра со свойством по умолчанию;

При перетаскивании параметра, у которого свойства Доступ имеет значение OUTPUT, вместо двунаправленной связи задается однонаправленная связь параметр=>свойство по умолчанию

- на графический элемент График – создание пера графика и задание однонаправленной связи параметр=>значение пера.

При обычном перетаскивании параметра на другие встроенные графические элементы никаких действий не выполняется.

8.2.2.4. РАЗМЕЩЕНИЕ ПРОГРАММЫ В ОКНЕ

При перетаскивании программы или функционального блока в окно из дерева MasterSCADA 4D создается графический элемент Кнопка, у которой настроено у события

Клик мыши действие Вызвать программу. Подпись кнопки будет совпадать с именем переставляемого элемента.

Возможны способы перетаскивания программы или функционального блока в окно указателем мыши с нажатием и удержанием как левой, так и правой кнопки мыши - результат будет одинаковым.

8.2.3. HMI. ПРОГРАММИРОВАНИЕ В ОКНАХ

Для того, чтобы программировать в редакторе HMI, не требуется знания специальных языков программирования. Большинство действий выполняются при помощи drag-n-drop, интуитивно понятных диалоговых окон и панелей свойств.

При создании окон перед разработчиком проекта наиболее часто стоят следующие задачи:

- Изменение значения свойства графического элемента в клиенте визуализации в режиме исполнения в зависимости от значения параметра MasterSCADA 4D (организация работы индикаторов, графиков и т.п.);
- Изменение значение параметра в дереве с использованием элемента управления, размещенного в окне (ввод уставок, запуск оборудования, например, насосов, вентиляторов и т.п.);
- Организация навигации по окнам (переключение между окнами, открытие всплывающих окон и пр.).

Эти задачи могут быть решены при помощи следующих механизмов:

- Динамизация
- Действия по событию
- Действия по условию

8.2.3.1. ДИНАМИЗАЦИЯ

Динамизация - это связь со свойствами графического элемента.

Связи со свойствами графического элемента могут быть как входящими, так и исходящими.

Входящие связи (входная динамизация)

Если установлена входящая связь, то свойство графического элемента является приемником данных и меняется в зависимости от изменения источника. Например, если необходимо чтобы в окне клиента визуализации цвет индикатора менялся в зависимости от значения параметра, то следует установить связь между требуемым параметром и свойством графического элемента Цвет. Для этого нужно перетащить параметр на свойство, удерживая левую кнопку мыши. После установления связи с параметром должна быть настроена конвертация.

Конвертация - это преобразование передаваемых значений.

Исходящие связи (выходная динамизация)

Если установлена исходящая связь, то свойство графического элемента является источником данных и приводит к изменению связанного элемента. Например, если необходимо, чтобы значение, введенное в клиенте визуализации в элементе Текстовый ввод, передавалось бы какому-либо параметру, то следует свойство Текст из панели свойств перетащить на параметр. При необходимости следует настроить конвертацию (для исходящих связей конвертацию значений настраивают реже, чем для входящих). Как правило, исходящая связь не используется при работе с кнопками, т.к. короткие изменения свойства (например, короткое нажатие кнопки без фиксации) может не успеть передаться по связи, и в этом случае используется другой механизм - действие по событию.

Настраивая выходную динамизацию, необходимо учитывать, что при открытии окна после запуска клиента визуализации по связи передается заданное в свойстве элемента значение, что может приводить к ложным срабатываниям (особенно, если клиентов визуализации несколько). В этом случае следует устанавливать и входящую и исходящую связи с одним и тем же параметром. При одновременно установленных входящей и исходящей связях сначала обрабатывается входящая связь, а затем исходящая.

Важно! В случае НМІ v1 в загружаемом в узел проекте может быть использовано не более 1000 элементов с динамизацией. В случае НМІ v2 ограничения на количество используемых в одном проекте элементов с динамизацией отсутствуют.

8.2.3.1.1. КОНВЕРТАЦИЯ ЗНАЧЕНИЙ

При связи свойства мнемосхемы или графического элемента с некоторым параметром (или другим свойством) преобразование передаваемых значений может быть сконфигурировано при передаче параметр=>свойство, так и при передаче свойство=>параметр.

Настройки конвертации могут быть заданы при использовании следующих средств:

- Легенда НМІ. Вкладка Связи ;
- Панель свойств.

Следует учитывать следующие характеристики:

- тип конвертации – определяет алгоритм вычисления значения приемника между опорными точками;
- опорные точки – значения источника (X_i) и приемника (Y_i) используемые для конвертации значений.


Типы конвертации

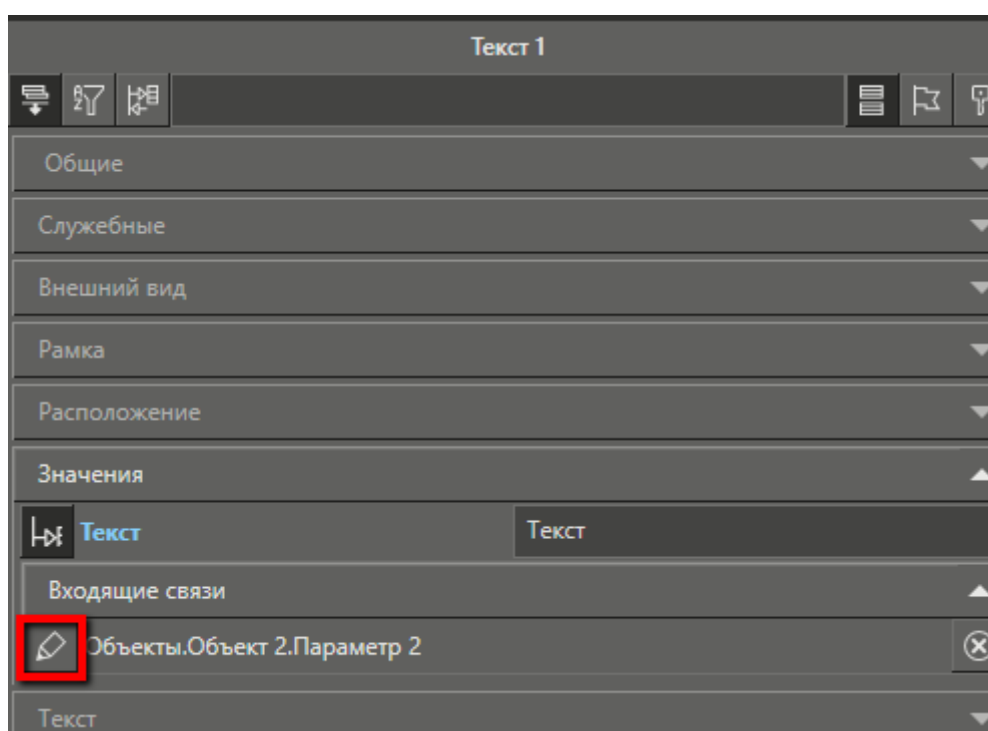
MasterSCADA 4D имеет следующие типы конвертации:

Тип	Описание	Примеры
Прямая	Значение приемника всегда равно значению источника	Данный тип конвертации используется по умолчанию при стандартном отображении параметров в окне
Точечная	<p>Одному значению источника, соответствует одно значение приемника.</p> <p>Если $X=X_i$ (X – текущее значение источника), то $Y=Y_i$ (Y – текущее значение приемника).</p> <p>Если X лежит между опорными точками, то Y имеет Значение по умолчанию, заданное в конвертере значений значений.</p>	<p>Такой тип конвертации удобно использовать для работы с параметрами и свойствами, которые имеют тип данных</p> <p>Перечисление</p>
Линейно-кусочная	<p>Если $X \leq X_{\min}$, то Y имеет Значение по умолчанию, заданное в конвертере значений</p> <p>Если $X \geq X_{\max}$, то $Y=Y_{\max}$.</p> <p>Если $X_i < X < X_{i+1}$, то Y линейно интерполируется:</p> $Y := Y_i + \frac{(Y_{i+1} - Y_i)}{(X_{i+1} - X_i)}(X - X_i)$	Используется для преобразования одной шкалы в другую, например, если свойство элемента может изменяться от 0 до 1, а связанный параметр меняется от 0 до 100.
Ступенчатая	<p>Если $X \leq X_{\min}$, то Y имеет Значение по умолчанию, заданное в конвертере значений.</p> <p>Если $X \geq X_{\max}$, то $Y=Y_{\max}$.</p> <p>Если значение источника лежит между опорными точками, приемник сохраняет свое значение.</p>	Используется, например, в случае, если требуется обеспечить следующий алгоритм: индикатор имеет синий цвет когда значение меньше 30, желтый цвет когда значение больше 30, но меньше 70, и красный цвет, когда значение равно или больше 70.

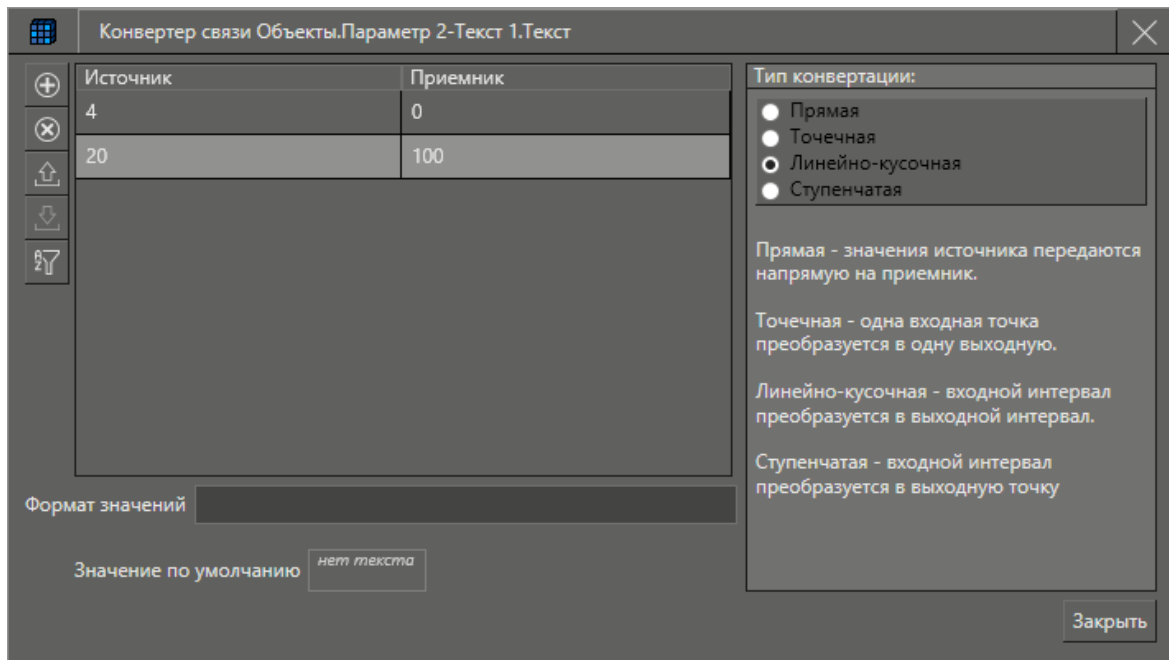
Тип	Описание	Примеры
Бит	Выделяет бит из целого значения. Нумерация битов начинается с нуля. Может использоваться при связи свойства типа BOOL и целого значения. Используется только для входящей связи. Если бит равен 1, то связанное свойство имеет значение TRUE, если бит равен 0, то свойство имеет значение False	Возможно есть набор флагов, которые должны отображать состояние битов целого числа, в котором например, передаются значения настроек оборудования.

8.2.3.1.1.1. КОНВЕРТЕР ЗНАЧЕНИЙ

Если между свойством и параметром (или другим свойством) установлена связь, то при нажатии на кнопку  открывается диалоговое окно конвертера в панели свойств:








Диалоговое окно имеет вид:



В центральной части находится таблица, в которой задается зависимость между источником и приемником данных.

Элементы диалогового окна:

Элемент	Описание
Источник	Задаются значения опорных точек источника данных (X_i , где i - номер опорной точки), значение заданное в верхней строке X_1 , значение, заданное в нижней строке - X_{max} . Если это входная динамизация, то источником является параметр (или другое свойство), если выходная динамизация, то значение свойства.
Приемник	Задаются значения опорных точек приемника данных (Y_i , где i - номер опорной точки), значение заданное в верхней строке Y_1 , значение, заданное в нижней строке - Y_{max}). Если это входная динамизация, то приемником является свойство, если выходная динамизация, то значение параметра (или другого свойства)
Тип конвертации	Выбирается зависимость между источником данных и приемником. Типы конвертации описаны в разделе Конвертация значений
	Добавляет опорную точку.
	Удаляет опорную точку.

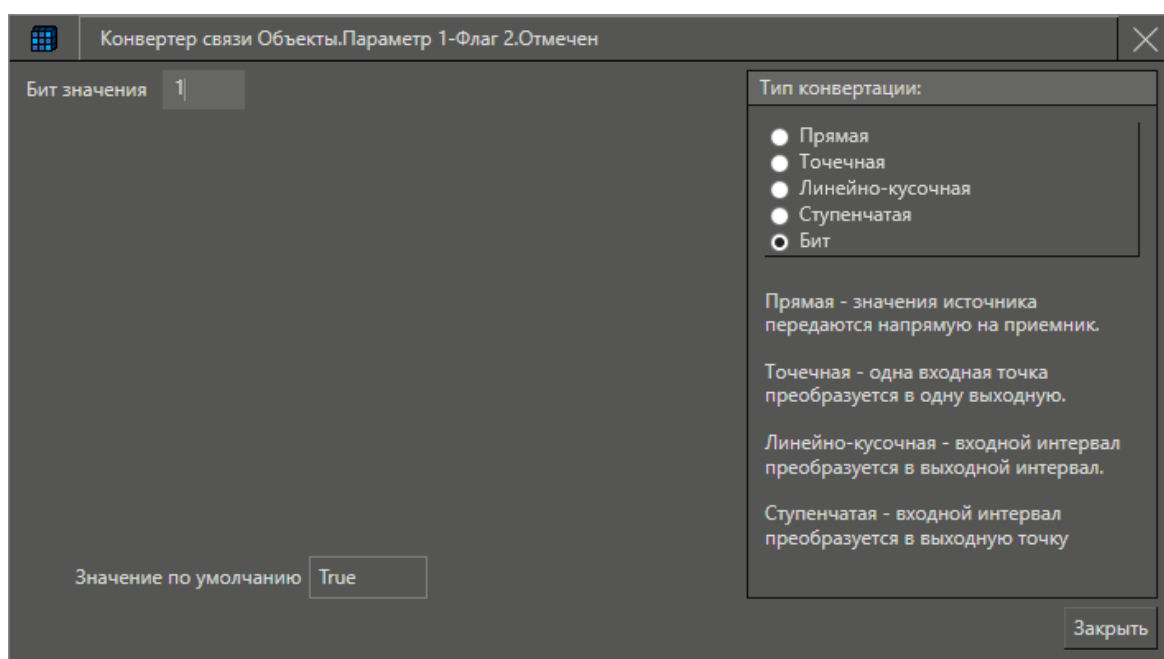
Элемент	Описание
	Перемещает опорную точку вверх.
	Перемещает опорную точку вниз.
	Размещает опорные точки по алфавиту, в зависимости от источника.
Формат значений	Назначает формат отображения значения, полученного в результате конвертации
Значение по умолчанию	Задаёт значение по умолчанию. Если конвертация настраивается для исходящей связи, то это значение будет передано по связи в момент открытия окна в клиенте визуализации. Используется также для расчета значения при конвертации.

Данное диалоговое окно используется также при работе с журналом для настройки зависимости стиля строки от какого-либо свойства сообщения.

Особенности работы

Работа с дискретными свойствами

Если свойство, которое имеет тип BOOL, связать с целым параметром, то окно конвертации кроме стандартных типов конвертации появится еще одна - Бит:



В поле Бит значения нужно указать тот бит входного значения, который будет отвечать за данное свойство. Нумерация битов начинается с нуля.

Важно! Выделение битов из целого числа работает только для входящей связи. Исходящую связь при данном типе конвертации использовать нельзя.

Если бит равен 1, то связанное свойство имеет значение TRUE, если бит равен 0, то свойство имеет значение False

8.2.3.1.1.2. НЕЯВНАЯ КОНВЕРТАЦИЯ ЗНАЧЕНИЙ

В MasterSCADA 4D поддерживается неявная конвертация значений при связях параметров с разными типами данных:

- BOOL → цвет; False → #00000000, TRUE → #FFFFFFF;
- BOOL → INT (все числовые); False → 0, TRUE → 1;
- BOOL → STRING; False → "false", TRUE → "true"

По умолчанию значение даты и времени выводится в формате dd-ММ-уууу HH:mm:ss.fff (если формат не задан явно). Данный формат используется для даты и времени при всех неявных преобразованиях.


8.2.3.2. НМІ. ДЕЙСТВИЯ ПО УСЛОВИЮ

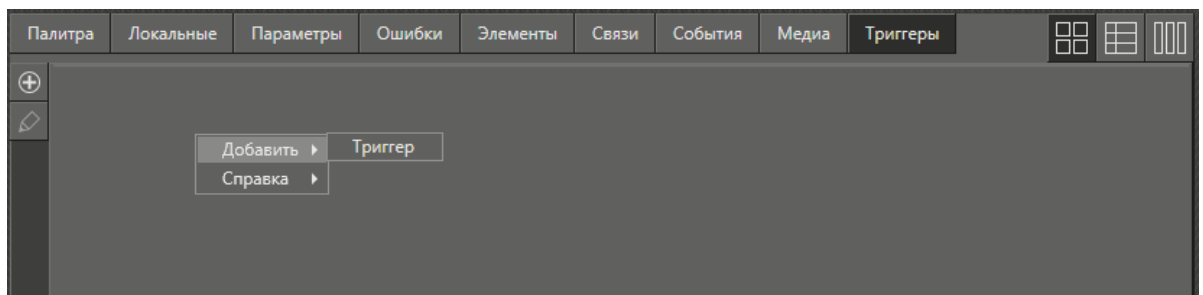
Динамизация свойства окна или графического элемента может быть сконфигурирована с помощью Триггеров, которые создаются и настраиваются в легенде редактора НМІ на вкладке Триггеры.

Триггер используется в тех случаях, когда требуется проверить какое-либо условие, и при его удовлетворении выполнить действия.

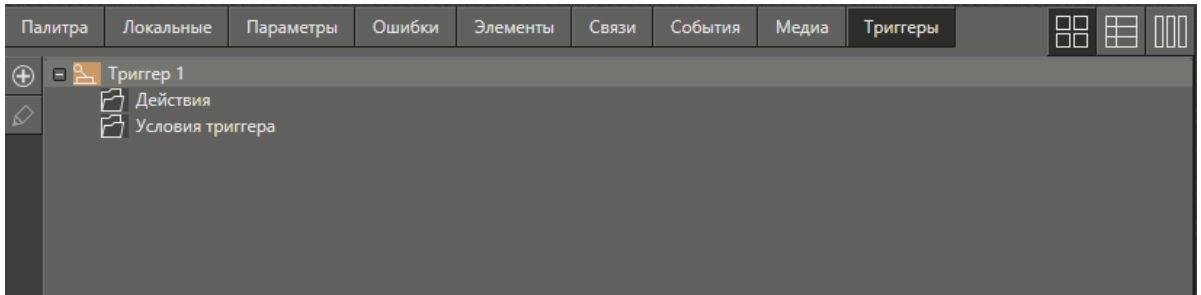
Важно! Триггер будет проверять условие, только если окно открыто! В случае если окно открыто в нескольких клиентах, триггеры в каждом клиенте будут работать независимо .

Для того чтобы создать Триггер, необходимо нажать правой кнопкой мыши в любое место вкладки Триггеры, и в появившемся контекстном меню выполнить пункт

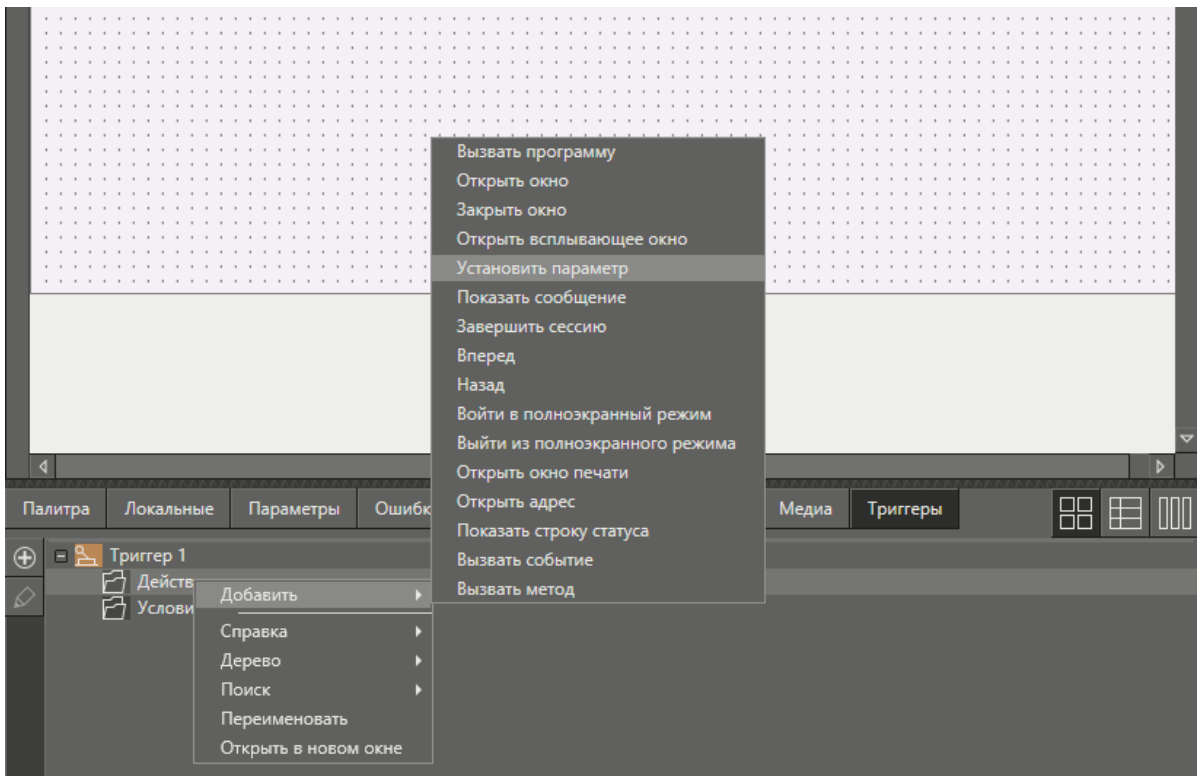
Добавить, либо нажать кнопку 



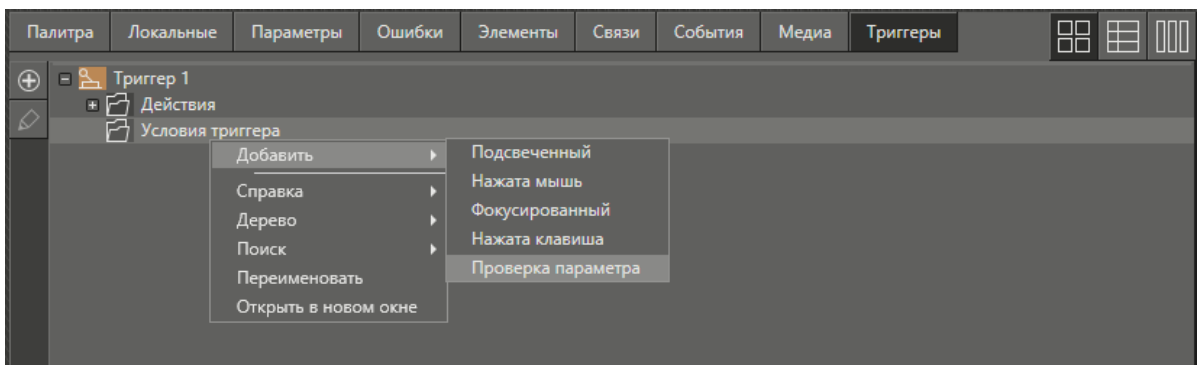
В результате получим:



В группу Действия добавляются действия, которые необходимо выполнить, когда условие Триггера будет удовлетворено.



В группу Условия триггера, соответственно, добавляются условия, при удовлетворении которых необходимо выполнить действия:



Проверены могут быть как браузерные условия, так и состояния параметров.

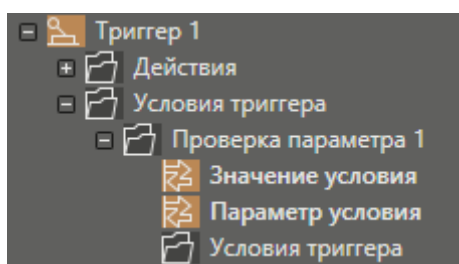
Браузерные условия

К таким условиям относятся:

- Подсвеченный – то же, что и событие Захват мыши;
- Нажата мышь – то же, что и событие Нажатие мыши;
- Фокусированный – то же, что и событие Получение фокуса;
- Нажата клавиша – то же, что и событие Нажатие клавиши.

Условие Проверка параметра

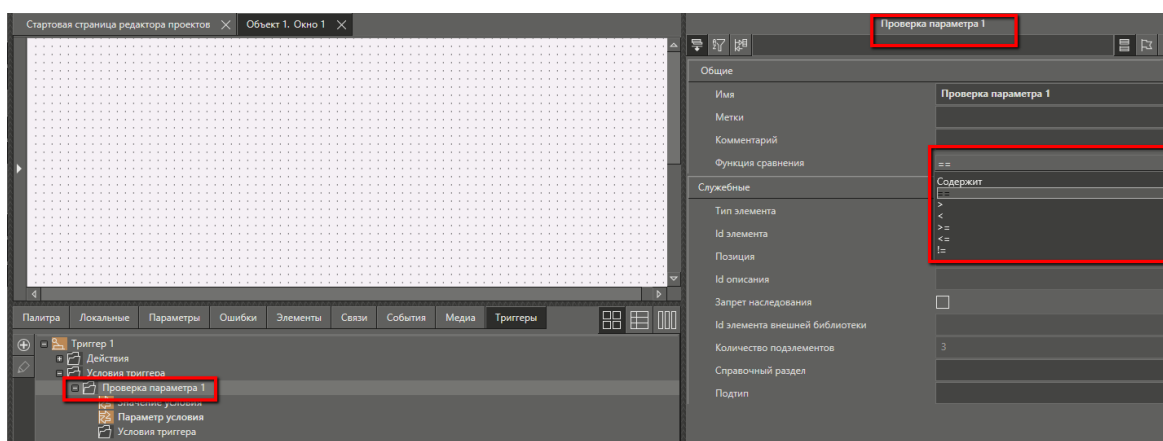
Данное условие конфигурируется с помощью следующих параметров:



В панели свойств параметра Триггера Значение условия в поле Начальное значение может быть задана константа, либо может быть установлена связь этого параметра с каким-нибудь другим параметром.

Параметр условия связывается с каким-либо параметром, значения которого нужно отслеживать (для установления связи следует перетащить контролируемый параметр на параметр Триггера).

В панели свойств Проверка параметра 1, необходимо выбрать функцию сравнения:



Если два условия идут друг за другом, то триггер сработает, если выполнилось хотя бы одно условие.

Если в условие добавлены другие условия, то триггер сработает, если выполнены оба условия.

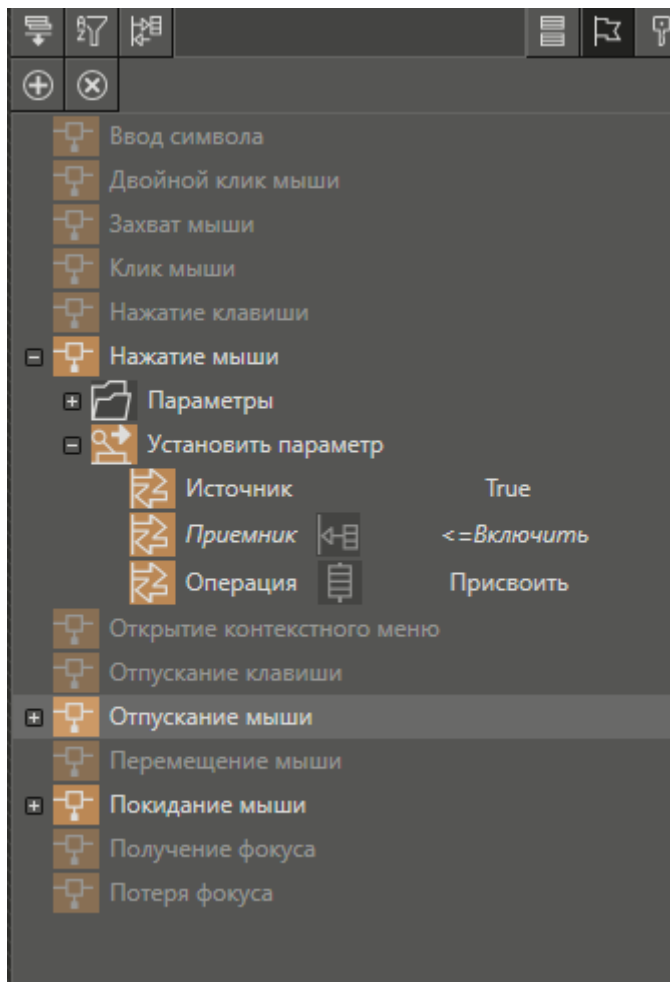
8.2.3.3. HMI. ДЕЙСТВИЯ ПО СОБЫТИЮ

Для мнемосхемы или графического элемента могут быть заданы predeterminedные действия, автоматически выполняемые при возникновении указанных predeterminedных событий (см. HMI. События и HMI. Действия).

8.2.3.3.1. HMI. СОБЫТИЯ

Предeterminedные события имеются как у окна, так и у каждого отдельного элемента. У разных элементов список возможных событий будет разным.

Они отображаются в панели свойств, в режиме События:

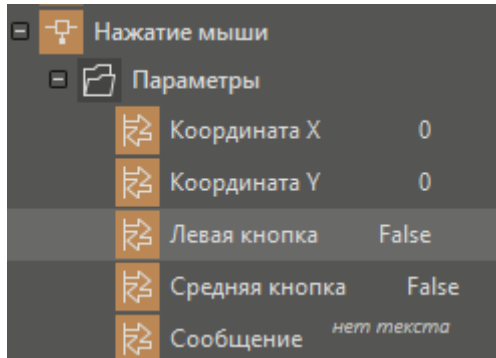


В редких случаях для выполнения операций с событиями может быть использована группа Окно. Группа События мнемосхемы или графического элемента. Кроме того, для выполнения операций с событиями окна предназначена вкладка Легенда HMI. Вкладка События.

Для мнемосхемы или графического элемента может быть задано по одному событию каждого вида.

Группа события содержит, как правило, подгруппу Параметры. Количество и назначение параметров зависит от типа события.

Например, для событий отслеживающих работу мыши, применимы следующие параметры:



Особое внимание стоит уделить параметру Сообщение. Если задано значение сообщения, то для пользователя, для которого установлены флаги Подтверждение и Журнал в настройках безопасности, данный текст будет выдаваться во всплывающем окне подтверждения вместо стандартного, а также будет фиксироваться в журнале сообщений.

Важно! Значения параметров можно использовать только внутри событий. Соединить напрямую параметр события и параметр проекта нельзя.

Для некоторых событий подгруппа Параметры может содержать и другие параметры, указанные в описании таких событий в таблице ниже.

Предопределенные события:

Название	Описание
Ввод символа	<p>Ввод символа при фокусе на мнемосхеме или графическом элементе. При вводе символа мнемосхема или графический элемент теряет фокус. Группа Параметры содержит:</p> <ul style="list-style-type: none"> Символ - выводится значение введенного символа; Код клавиши-числовые коды нажатой клавиши, используются для работы со служебными клавишами. <p>Как правило, используется для вызова программы, в которой анализируется нажатая клавиша на клавиатуре (т.е. параметр события связывается с параметром программы).</p>

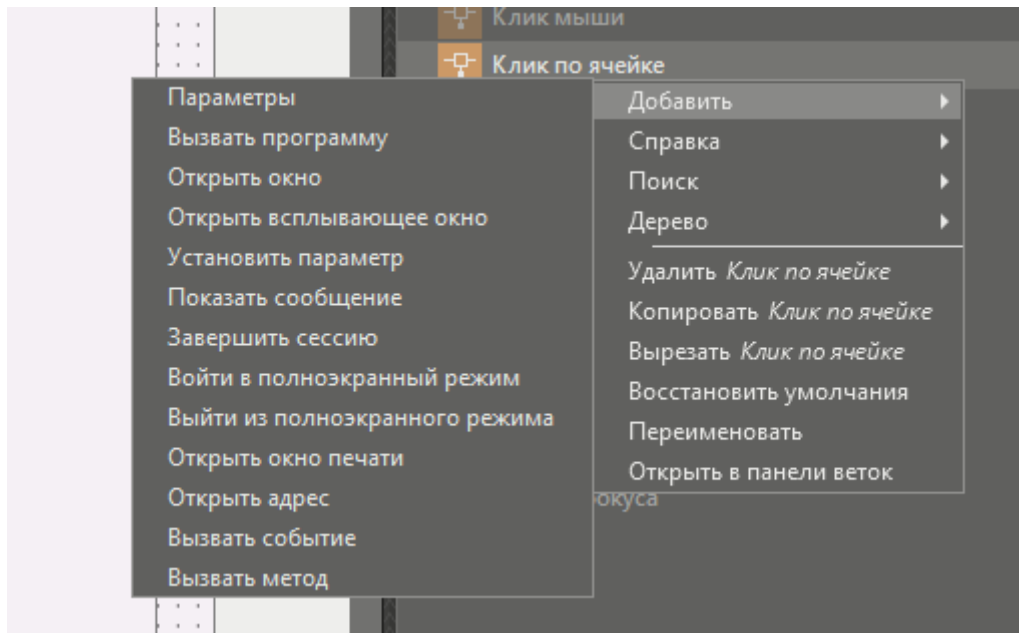
Название	Описание
	Если в клиенте визуализации клавиша была нажата и не отпущена , то введенный символ повторяется периодически пока клавиша остается нажатой.
Двойной клик мыши	Нажатие и отпускание левой кнопки мыши, выполненные дважды с интервалом не более 0.5 с при нахождении мыши в пределах мнемосхемы или графического элемента.
Захват мыши	Наведение мыши на мнемосхему или графический элемент.
Клик мыши	Нажатие и отпускание левой кнопки мыши при нахождении мыши в пределах мнемосхемы или графического элемента.
Нажатие клавиши	<p>Нажатие клавиши клавиатуры при фокусе на мнемосхеме или графическом элементе. Событием является нажатие не только клавиши ввода символа, но и, например, клавиши CTRL. При нажатии клавиши клавиатуры мнемосхема или графический элемент теряет фокус.</p> <p>Группа Параметры содержит:</p> <ul style="list-style-type: none"> • Символ - выводится значение введенного символа; • Код клавиши-числовые коды нажатой клавиши, используются для работы со служебными клавишами.
Нажатие мыши	Нажатие левой кнопки мыши при нахождении мыши в пределах мнемосхемы или графического элемента.
Открытие контекстного меню	Нажатие правой кнопки мыши. Если назначить на это событие некоторое действие (например, открытие некоторого всплывающего окна), то стандартный обработчик контекстного меню не будет выполнен.
Отпускание клавиши	Отпускание клавиши клавиатуры при фокусе на мнемосхеме или графическом элементе. Событием

Название	Описание
	является отпускание не только клавиши ввода символа, но и, например, клавиши CTRL. Подгруппа Параметры содержит те же параметры, что и в событии Событие Ввод символа.
Отпускание мыши	Отпускание левой кнопки мыши при нахождении мыши в пределах мнемосхемы или графического элемента.
Перемещение мыши	Перемещение мыши в пределах мнемосхемы или графического элемента.
Покидание мыши	Выведение мыши за пределы мнемосхемы или графического элемента.
Получение фокуса	Перевод фокуса на мнемосхему или графический элемент. Группа этого события не содержит подгруппу Параметры.
Потеря фокуса	Перевод фокуса на объект вне мнемосхемы или графического элемента. Группа этого события не содержит подгруппу Параметры.
Клик по ячейке	События используется в элементах Таблица данных и Справочник. Порядок работы сморите в разделе: Событие Клик по ячейке

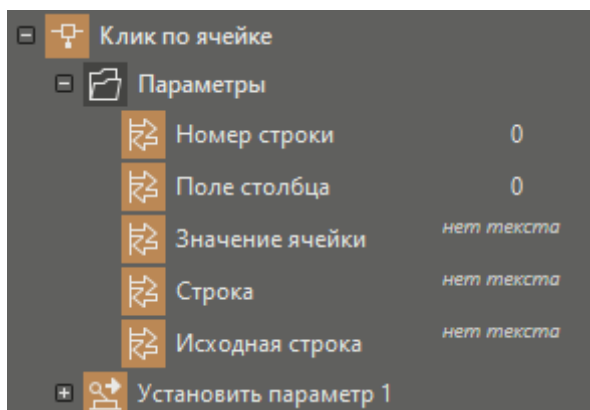
8.2.3.3.1.1. СОБЫТИЕ КЛИК ПО ЯЧЕЙКЕ

Элемент Таблица данных и Справочник имеют уникальное событие Клик по ячейке.

Для данного события доступны следующие действия:



После добавления какого-либо действия в событие добавляется группа Параметры:



Название	Назначение
Номер строки	Позволяет получить номер строки, в которой находится ячейка, на которой был выполнен клик.
Поле столбца	Позволяет получить поле столбца, в котором находится ячейка, на которой был выполнен клик.
Значение ячейки	Выдается текущее значение ячейки, на которой был выполнен клик. Тип передаваемого значения всегда будет STRING. Независимо от того, как тип имеет поле структуры, к которому относится ячейка
Строка	Выдается значение всей строки.

Название	Назначение
	Для справочника в случае, если используются поля из зависимых таблиц. Например есть поля idAddress.Street и idAddress.City, то в параметре Строка будет оба поля - idAddress_Street и idAddress_City, которые содержат значения зависимых полей, которые отображаются в самой таблице справочника (. заменяется в именах на _)
Исходная строка	Выдает тоже значение, что и Строка у элемента Таблица данных. Для Справочника в случае, если используются поля из зависимых таблиц в параметре Исходная строка передается одно поле, например, idAddress, в котором передается значение поля idAddress исходной таблицы.

Эти параметры можно использовать при формировании действий.

Пример

Пусть необходимо при нажатии на ячейку выдавать информацию о ее номерах строки и столбца, а также ее текущее значение, открывать окно, в котором отображено в графическом виде все элементы, входящие в структуру. Пусть уже сформирован массив структур и настроена Таблица данных

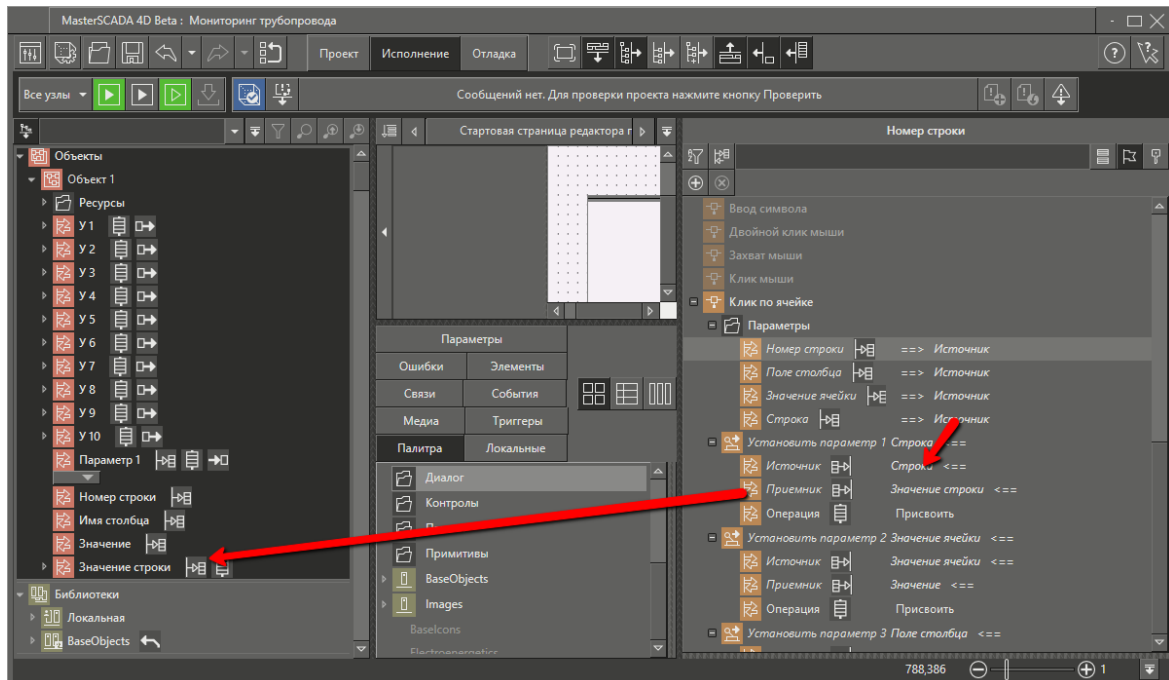
Получение информации о ячейке

В проекте создаем четыре параметра:

Название	Тип
Номер строки	LREAL
Имя столбца	STRING
Значение	STRING
Значение строки	Структура, которая является элементом массива, отображаемом в таблице данных

Для элемента Таблица данных добавляем четыре действие Установить параметр в событие Клик ячейки.

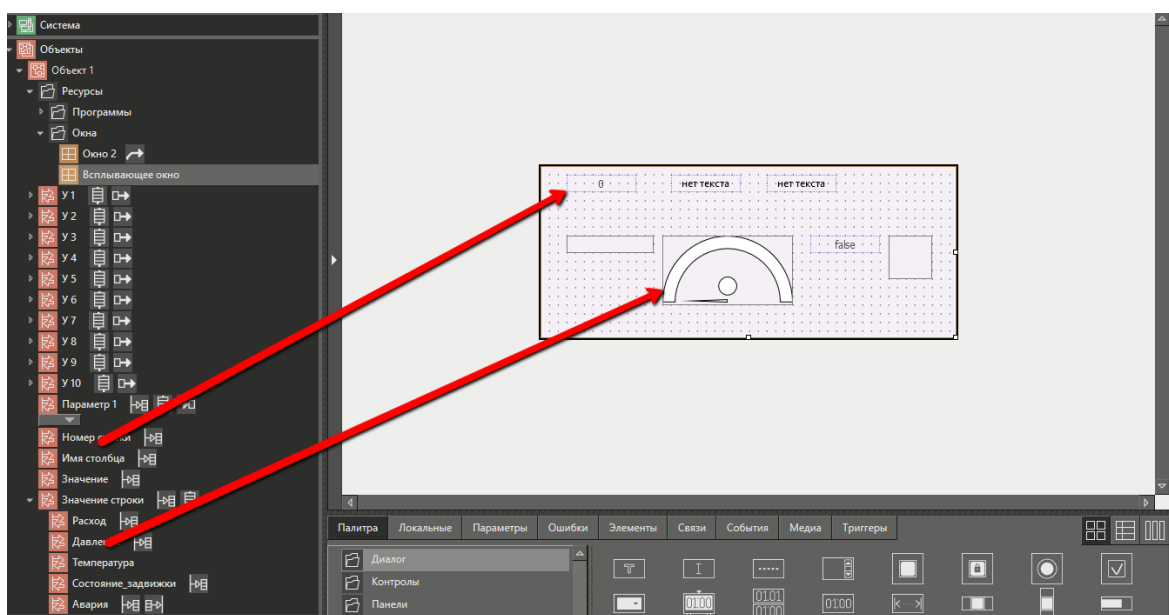
В качестве источника данных будут параметры события, а в качестве приемника добавленные параметры:



Создание всплывающего окна

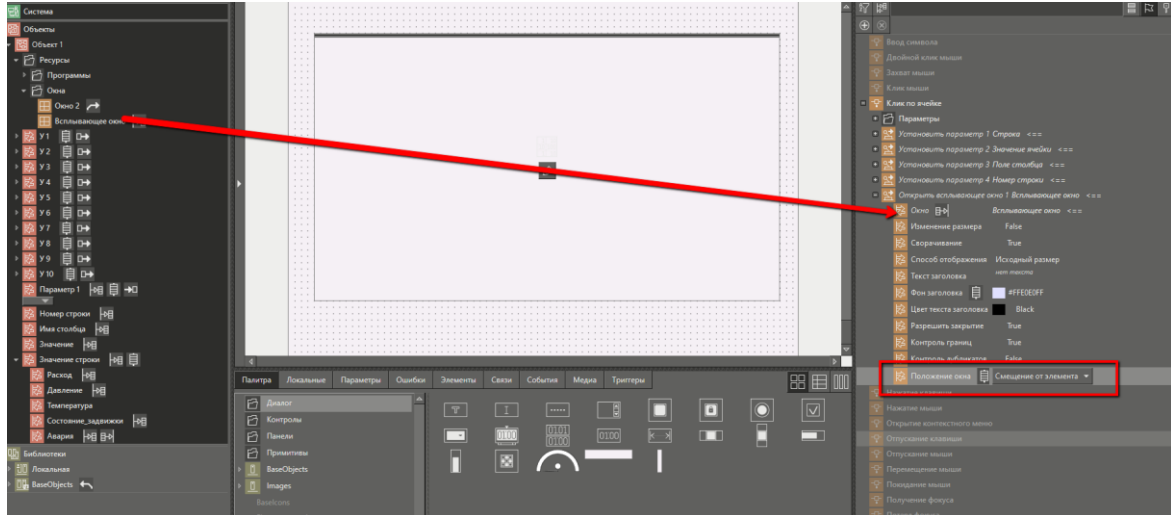
Создаем окно, которое должно открываться при нажатии на ячейку. Пусть оно называется Всплывающее окно.

Далее необходимо или перетащить параметры Номер строки, Имя столбца, Значение., а также поля структуры Значение строки в окно левой или правой кнопкой мыши, либо использовать их для динамизации произвольных элементов:



Открытие всплывающего окна

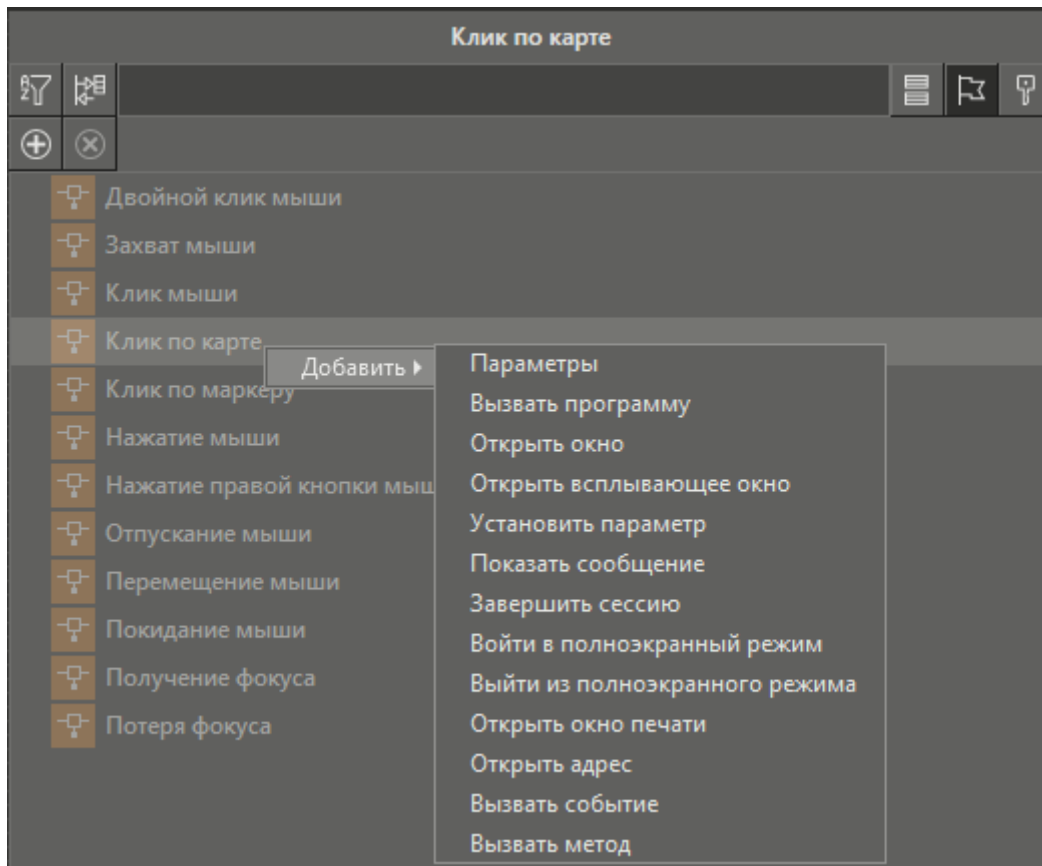
Для элемента Таблица данных добавляем действие Открыть всплывающее окно в событие Клик ячейки. Параметр действия Окно нужно связать с созданным окном. Если установить значение параметра Положение окна Смещение от элемента, то окно будет открываться рядом с ячейкой, по которой был выполнен клик:



8.2.3.3.1.2. СОБЫТИЕ КЛИК ПО КАРТЕ

Элемент Карта имеют уникальное событие Клик по карте.

Для данного события доступны следующие действия:



После добавления действия становятся доступны для настройки параметры события.

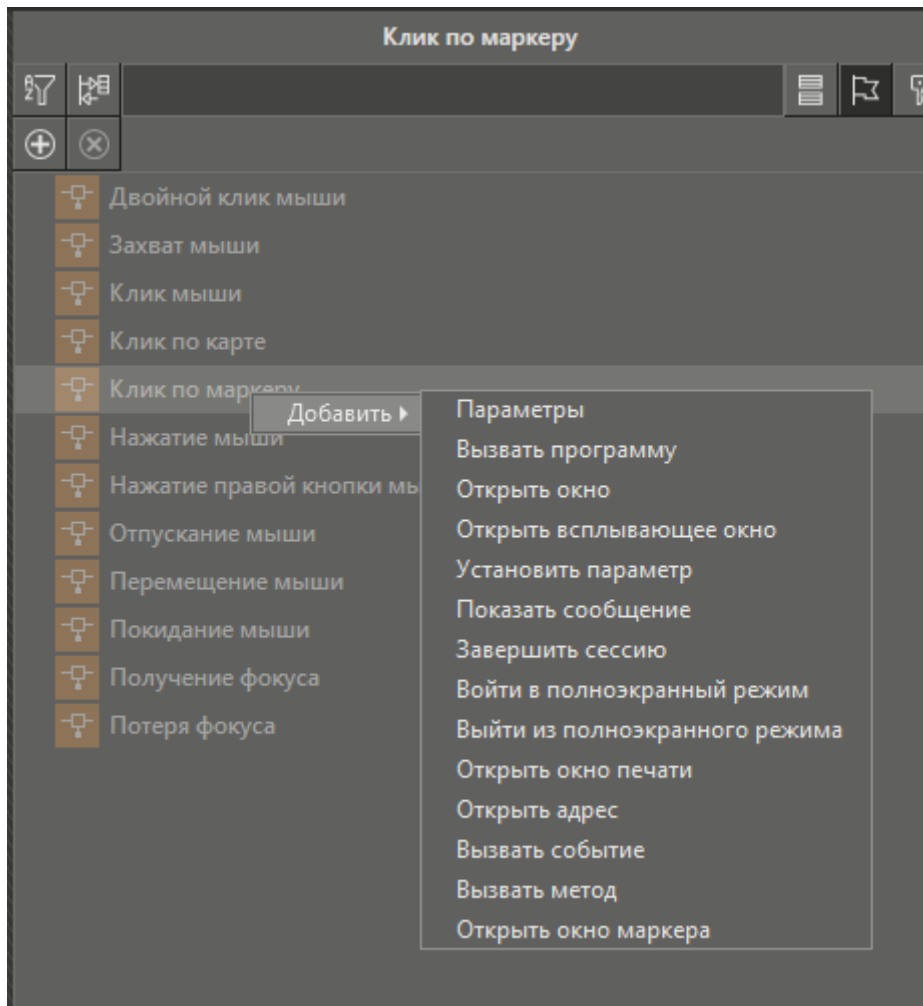
Параметры события только передают значения. Т.е при клике по какому-то маркеру на связанные переменные с данными параметрами можно получить:

Название	Назначение
Координата X	Выдает координаты места клика в координатной системе карты, долготу
Координата Y	Выдает координаты места клика в координатной системе карты, широту

8.2.3.3.1.3. СОБЫТИЕ КЛИК ПО МАРКЕРУ

Элемент Карта имеют уникальное событие Клик по маркеру.

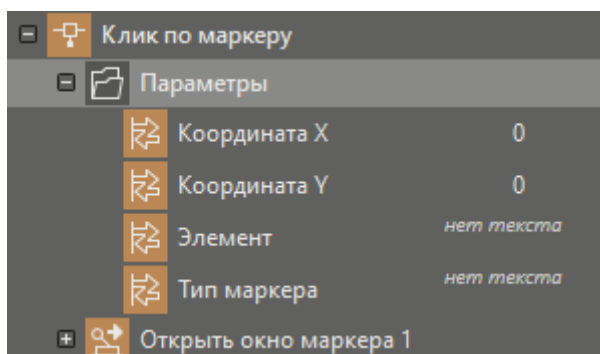
Для данного события доступны следующие действия:



Кроме стандартных действий для данного события существует специальное - Открыть окно маркера.

Параметры события

После добавления действия становятся доступны для настройки параметры события:

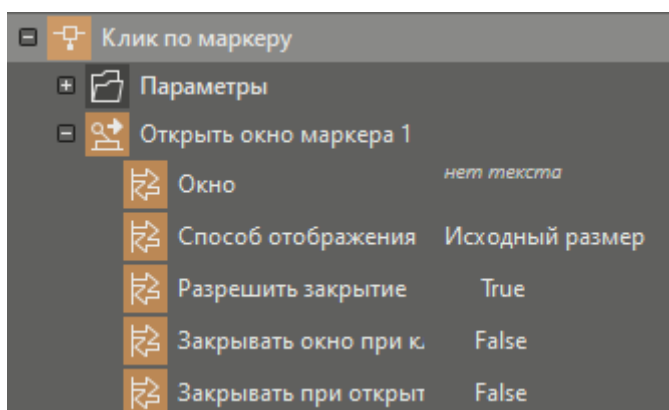


Параметры события только передают значения. Т.е при клике по какому-то маркеру на связанные переменные с данными параметрами можно получить:

Название	Назначение
Координата X	Выдает значение, которое соответствует полю структуры, отвечающему в источнике данных за долготу маркера, по которому выполнили клик.
Координата Y	Выдает значение, которое соответствует полю структуры, отвечающему в источнике данных за широту маркера, по которому выполнили клик.
Элемент	Выдает тот элемент массива структур, который характеризует маркер, по которому выполнили клик. Т.е. этот параметр нужно связать с параметром проекта, который будет иметь такой же тип Структура, из которого состоит массив структур источника данных карты.
Тип маркера	Выдает значение, которое соответствует полю структуры, отвечающему в источнике данных за тип маркера, по которому выполнили клик.

Данные, полученные в параметрах события можно использовать для создания окна, которое будет отрываться при нажатии на маркер. Порядок формирования такого ока описан в разделе Всплывающее окно маркера карты.

Действие Открыть окно маркера



Функционал похож на стандартное действие Открыть всплывающее окно, но в данном случае, что всплывающее окно открывается всегда рядом с маркером, и значит у него меньше параметров для настройки:

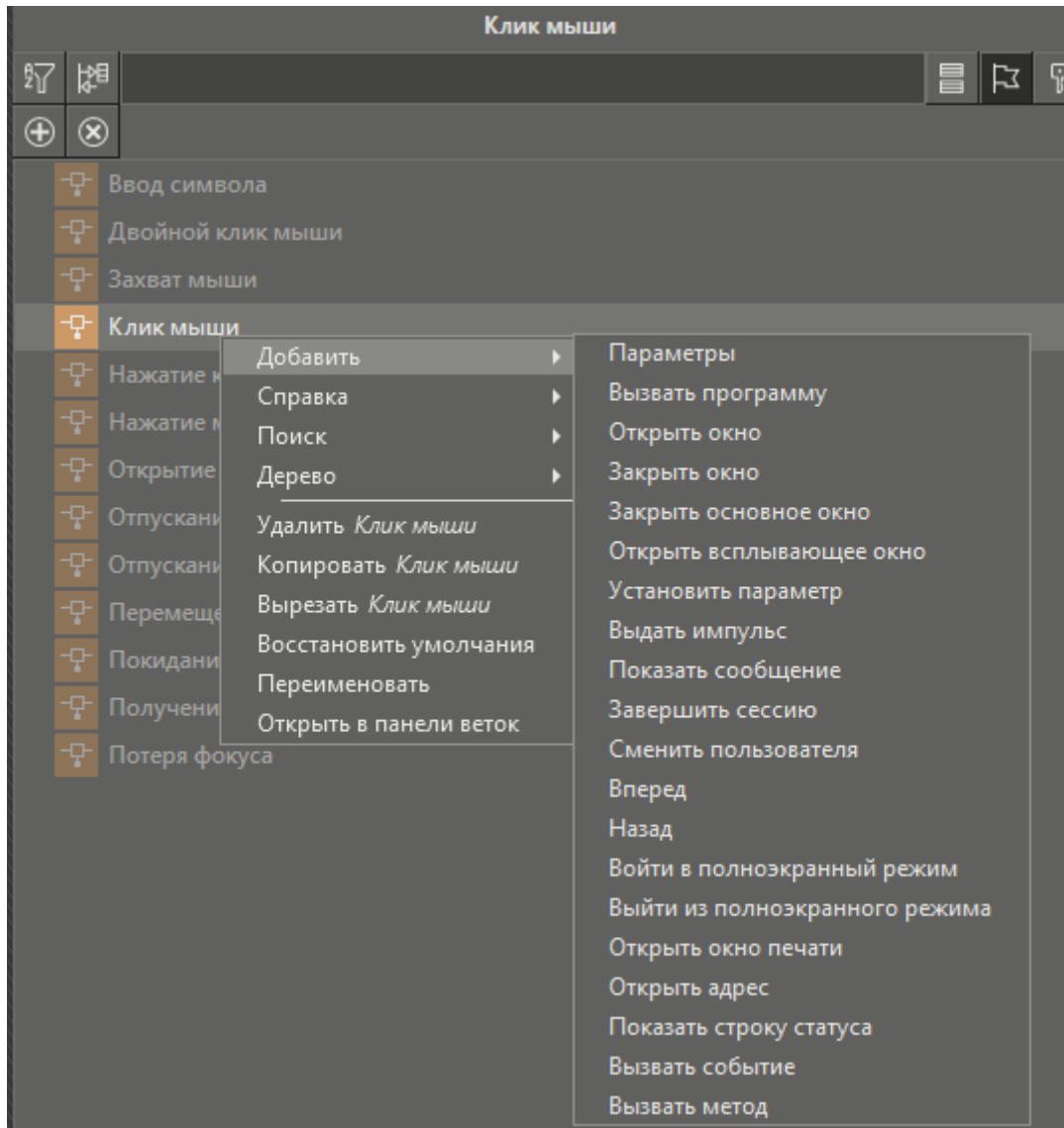
Название	Назначение
Окно	Устанавливает связь с окном, которое требуется открыть. Для установлении связи необходимо перетащить окно из дерева проекта на свойство.

Название	Назначение
Способ отображения	<p>Определяет способ подгонки окна просмотра и отображаемого окна. Возможные варианты:</p> <ul style="list-style-type: none"> • Исходный размер - размер всплывающего окна будет определяться размерами открываемого окна. Если свойства всплывающего окна Ширина/Высота были задинамизированы, то в режиме исполнения, если будут изменяться связанные параметры, то будет и изменяться размер всплывающего окна <ul style="list-style-type: none"> • Подогнать - содержимое окна будет уменьшено/увеличено таким образом, чтобы все элементы были видны в окне, размер которого определяется параметрами Ширина и Высота, которые появляются, если был выбран данный способ отображения. • Обрезать - часть содержимого окна будет скрыта, если не все элементы уместятся в окне, размер которого определяется параметрами Ширина и Высота, которые появляются, если был выбран данный способ отображения. Будут скрыты элементы, которые находятся внизу и справа <ul style="list-style-type: none"> • Прокрутить- появится линейка прокрутки снизу/справа, если не все элементы могут отобразиться в окне, размер которого определяется параметрами Ширина и Высота, которые появляются, если был выбран данный способ отображения.
Разрешить закрытие	Определяет возможность закрытия окна в клиенте визуализации. Если TRUE, то появится в окне кнопка-крестик для закрытия окна.
Закрывать окно при клике на карту или потере фокуса	Определяет будет ли окно автоматически закрыто при наступлении указанных событий.
Закрывать при открытии другого окна маркера	Определяет будет ли текущее окно закрыто, если будет открыто окно другого маркера. Если TRUE, то в один момент времени может быть открыто только одно окно.

8.2.3.3.2. HMI. ДЕЙСТВИЯ

Для каждого события (см. HMI. События) может быть задано одно или более предопределенных действий.

Для задания действий используется команда Добавить, контекстного меню события, а также команда удаления выделенных элементов:



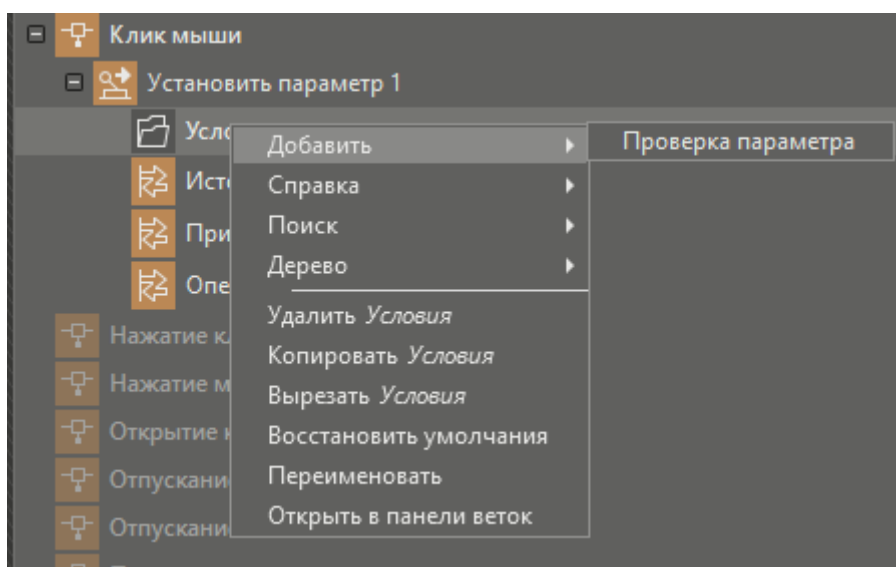
Список доступных действий:

- Действие Вызвать Программу
- Действие Открыть окно
- Действие Закреть окно
- Действие Закреть основное окно
- Действие Открыть всплывающее окно
- Действие Установить параметр

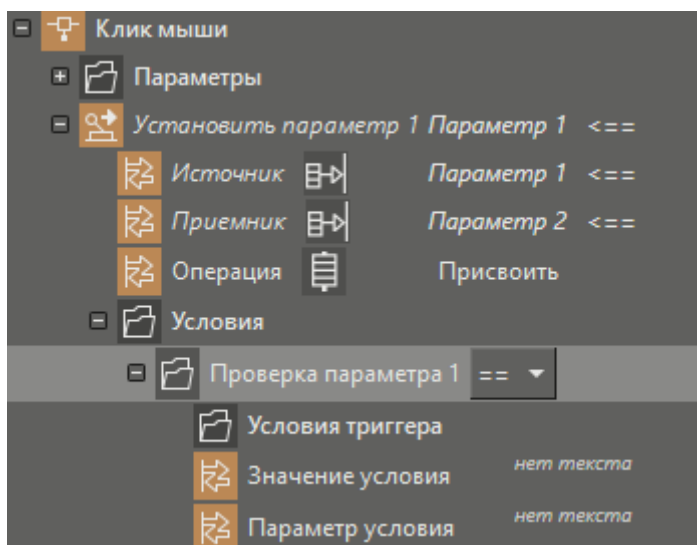
- Действие Выдать импульс
- Действие Показать сообщение
- Действие Завершить сессию
- Действие Сменить пользователя
- Действие Вперед
- Действие Назад
- Действие Войти в полноэкранный режим
- Действие Выйти из полноэкранного режима
- Действие Открыть окно печати
- Действие Открыть адрес
- Действие Показать строку статуса
- Действие Вызвать событие
- Действие Вызвать метод

В каждое действие можно добавить дополнительное условие выполнения. Можно добавлять только условия на проверку параметров.

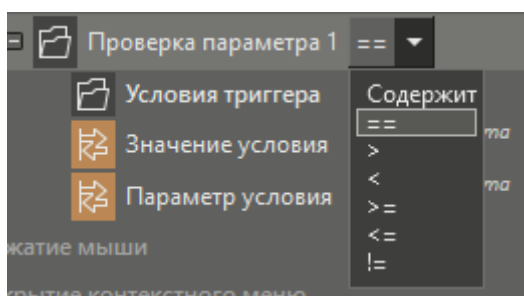
Каждое действие содержит папку Условия. В эту папку при помощи контекстного меню можно добавить условие Проверка параметра:



Далее необходимо связать Параметр условия, с каким-либо параметром проекта, и настроить Значение условия (можно установить связь с параметром или задать константу:



В этом случае, после выполнения клика мыши по элементу в клиенте визуализации действие выполнится только в том случае, если Параметр условия будет равен Значению условия. Т.е. по умолчанию проверяется условие равенства. Если необходимо изменить условие проверки параметра, то необходимо нажать на кнопку рядом с группой Проверка параметра и выбрать нужный вариант функции сравнения:

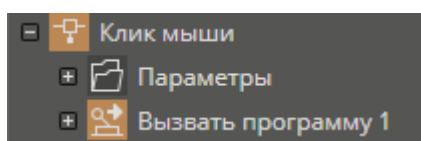


Если два условия идут друг за другом, то действие сработает, если выполнилось хотя бы одно условие.

Если в условии добавлены другие условия, то действие сработает, если выполнены оба условия.

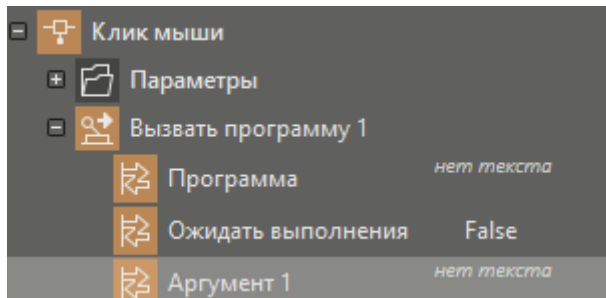
8.2.3.3.2.1. ДЕЙСТВИЕ ВЫЗВАТЬ ПРОГРАММУ

При задании этого действия группа события принимает следующий вид (на рисунке показано событие Клик мыши в окне свойств; см. также НМІ. Действия):



К параметру Программа привязывается программа, при этом в группу Вызвать Программу добавляются аргументы программы. К этим аргументам могут быть привязаны параметры

(во время вызова программы с помощью данного действия такие связи переопределяют связи, заданные для аргументов в программе):



Действие заключается в однократном вызове программы (в т.ч. допускается вызов программы объекта с параметрами свойство Место исполнения = В задаче экрана и свойство Способ исполнения = По вызову).

Если параметр Ожидать выполнения имеет значение TRUE, то перед выполнением следующих действий ожидается ответ исполнительной системы на вызов программы.

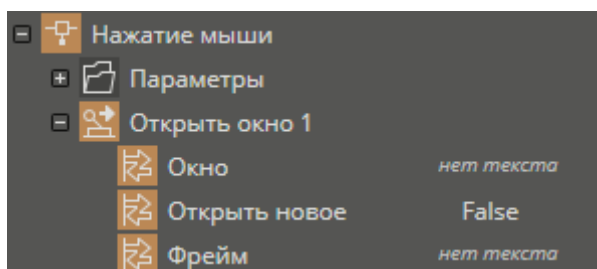
По команде Добавить контекстного меню действия Вызвать Программу в него может быть добавлен аргумент.

Вызов программы задачи экрана

Действие может быть использовано для вызова программ, которые работают в задаче экрана, однако асинхронный режим не поддерживается, т.е. необходимо установить параметр Ожидать выполнение в состояние True

8.2.3.3.2.2. ДЕЙСТВИЕ ОТКРЫТЬ ОКНО

При задании этого действия группа события принимает следующий вид (на рисунке показано событие Нажатие мыши в окне свойств; см. также HMI. Действия):



Действие заключается в открытии окна, привязанного к параметру Окно. Способы установки связи между параметром действия и окном, которое должно открываться показаны в разделе Действие Открыть всплывающее окно.

Для конфигурирования действия используются дополнительные параметры:

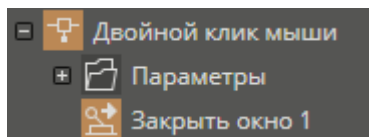
- Открыть новое – если параметр имеет значение TRUE, то привязанное к параметру окно открывается на новой вкладке браузера, а если False, то на той же. С помощью данной опции может быть открыто не более одной новой вкладки браузера;

- Фрейм – если к этому параметру привязан графический элемент Контейнер текущего окна, то привязанное к параметру окно открывается в нем, а параметр Открыть новое при этом игнорируется.

По команде Добавить в контекстном меню действия Открыть окно в него могут быть добавлены параметры окна (в т.ч. созданные пользователем).

8.2.3.3.2.3. ДЕЙСТВИЕ ЗАКРЫТЬ ОКНО

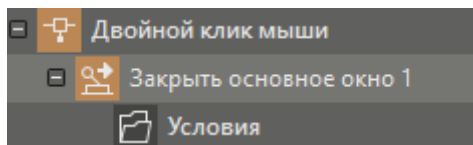
При задании этого действия группа события принимает следующий вид (на рисунке показано событие Двойной клик мыши, см. также НМІ. Действия):



Действие заключается в закрытии текущего всплывающего окна.

8.2.3.3.2.4. ДЕЙСТВИЕ ЗАКРЫТЬ ОСНОВНОЕ ОКНО

При задании этого действия группа события принимает следующий вид (на рисунке показано событие Двойной клик мыши, см. также НМІ. Действия):

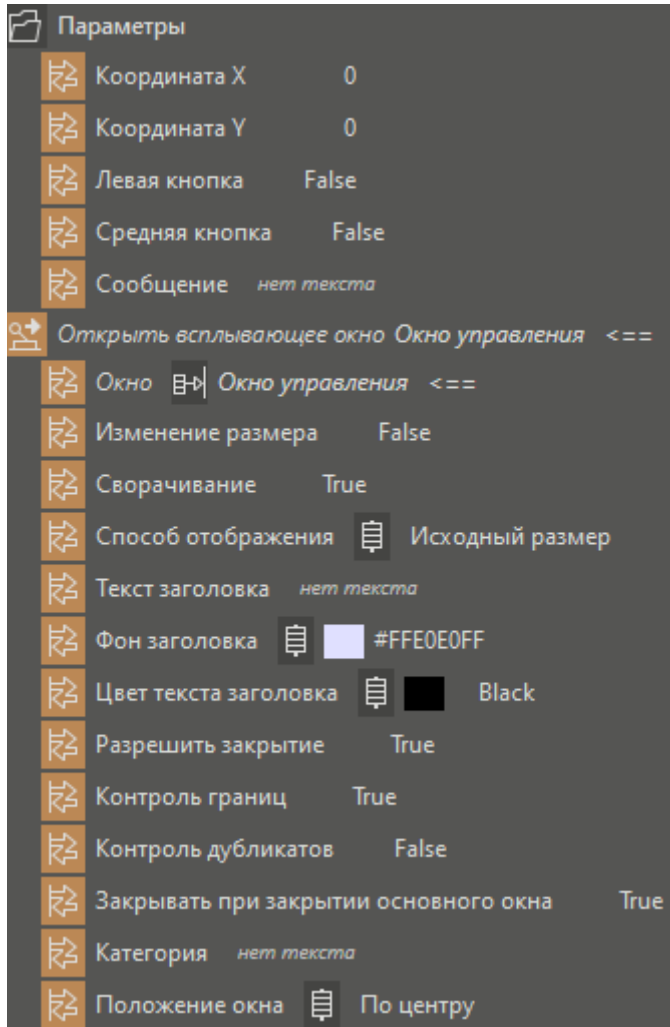


Действие заключается в закрытии окна клиента визуализации. Если в качестве клиента используется браузер стороннего производителя, то закроется только вкладка, при этом появится запрос от браузера.

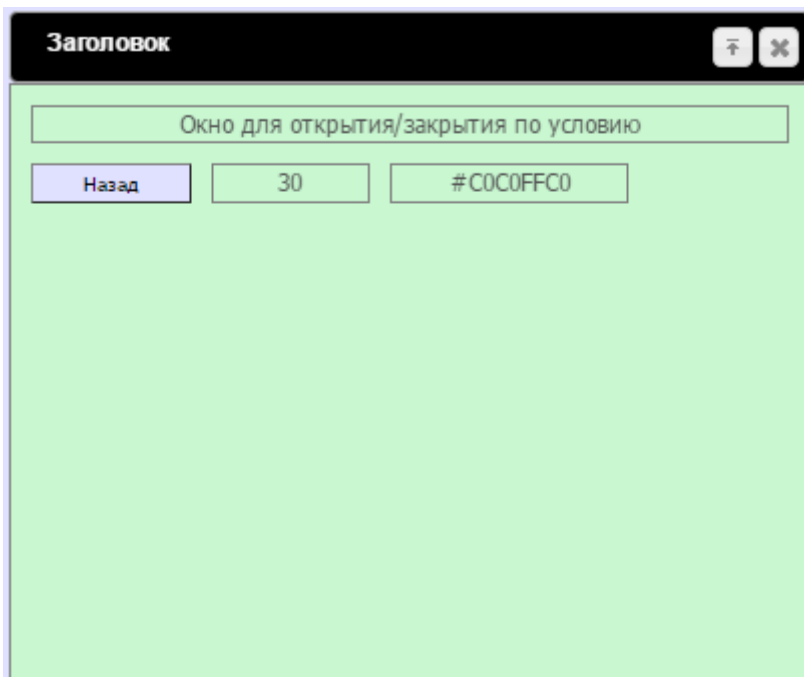
Действие может быть вызвано из любого окна, в том числе и из всплывающего.

8.2.3.3.2.5. ДЕЙСТВИЕ ОТКРЫТЬ ВСПЛЫВАЮЩЕЕ ОКНО

При задании этого действия группа события принимает следующий вид (на рисунке показано событие Нажатие мыши в окне свойств; см. также НМІ. Действия):

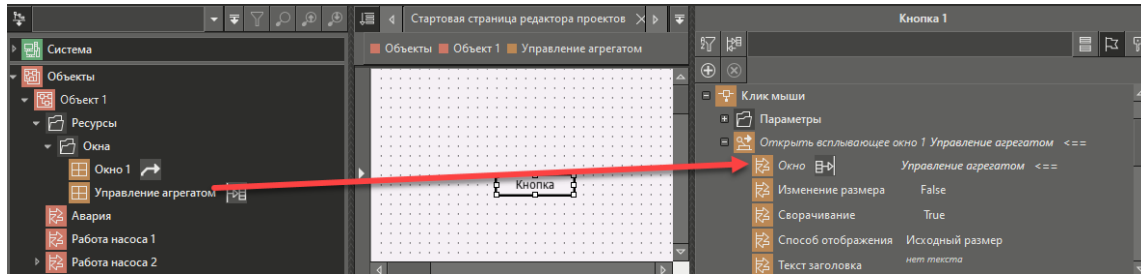


Действие позволяет открыть всплывающее окно, привязанное к параметру Окно:

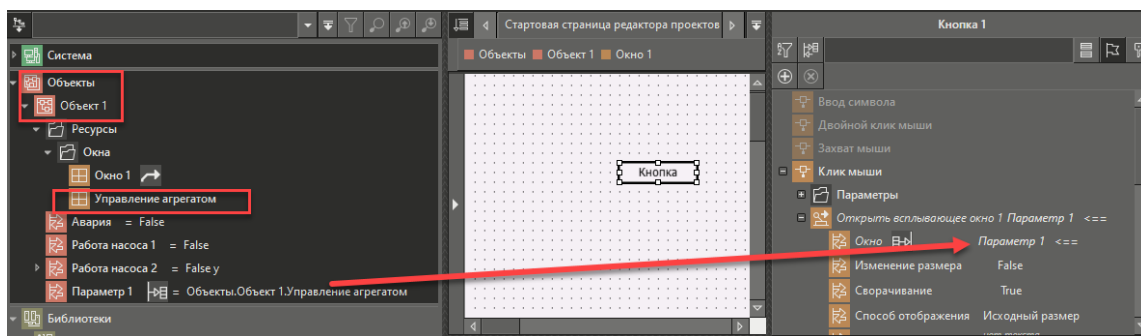


Для того чтобы привязать параметр действия Окно и то окно проекта, которое должно открываться, существует два способа.

1. Можно перетащить нужное окно из дерева проекта на свойство:






2. Параметр действия можно связать со строковой переменной, значение которой будет содержать путь к окну в виде: путь_к_объекту.имя_окна:



Важно! В НМИ v2 повторный вызов действия приведет к закрытию окна.


Всплывающее окно содержит заголовок и окно просмотра. В окне просмотра поддерживается прокрутка. После наведения мыши на заголовок всплывающее окно можно перемещать левой кнопкой мыши в пределах текущего окна браузера. При закрытии окна браузера всплывающее окно тоже закрывается.



Заголовок всплывающего окна содержит следующие инструменты:

-   – переключатель отображения области просмотра;
-  – закрыть всплывающее окно.

Для конфигурирования действия используются дополнительные параметры:

Параметр	Назначение
Окно	Устанавливает связь с окном, которое требуется открыть. Для установления связи необходимо перетащить окно из дерева проекта на свойство.

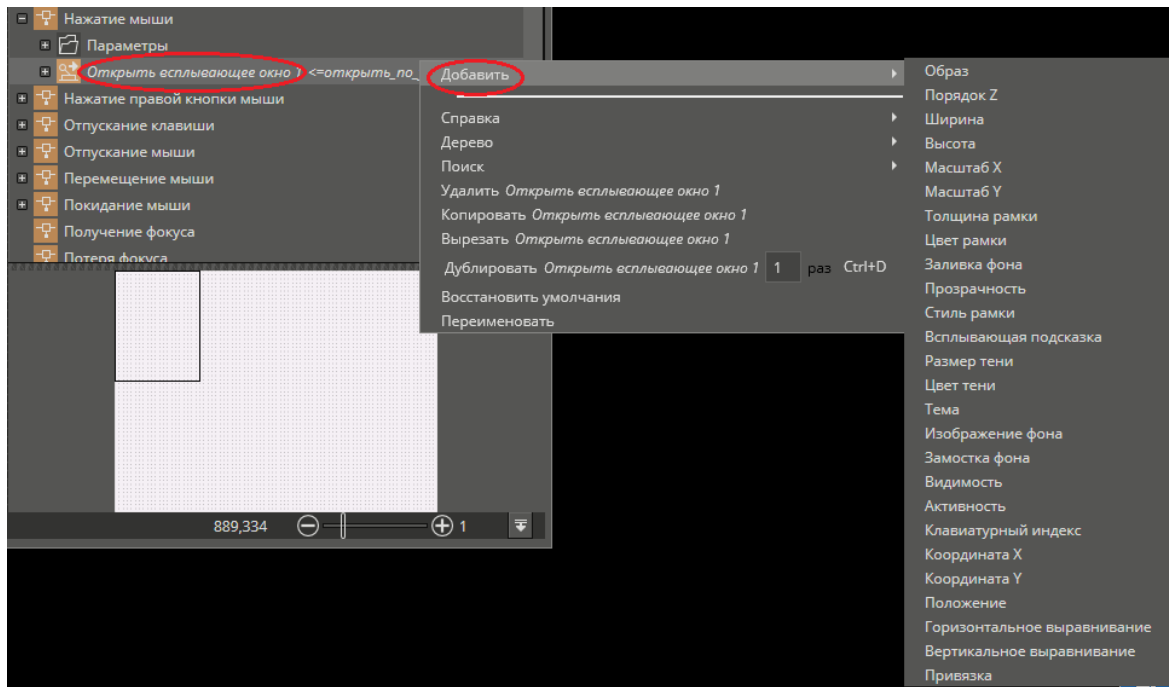
Параметр	Назначение
Изменение размера	<p>Определяет возможность изменения размера окна в клиенте визуализации. Если параметр имеет значение TRUE, то разрешено изменение размеров всплывающего окна (растягивание рамки разрешено при условии, что тип ширины/высоты в настройках открываемого окна - относительный).</p>
Сворачивание	<p>Определяет возможность сворачивания окна в клиенте визуализации. Если параметр имеет значение False, то в заголовке будет отсутствовать инструмент .</p>
Способ отображения	<p>Определяет способ подгонки окна просмотра и отображаемого окна. Возможные варианты:</p> <ul style="list-style-type: none"> • Исходный размер - размер всплывающего окна будет определяться размерами открываемого окна. Если свойства всплывающего окна Ширина/Высота были задинамизированы, то в режиме исполнения, если будут изменяться связанные параметры, то будет и изменяться размер всплывающего окна <ul style="list-style-type: none"> • Подогнать - содержимое окна будет уменьшено/увеличено таким образом, чтобы все элементы были видны в окне, размер которого определяется параметрами Ширина и Высота, которые появляются, если был выбран данный способ отображения. • Обрезать - часть содержимого окна будет скрыта, если не все элементы уместятся в окне, размер которого определяется параметрами Ширина и Высота, которые появляются, если был выбран данный способ отображения. Будут скрыты элементы, которые находятся внизу и справа <ul style="list-style-type: none"> • Прокрутить - появится линейка прокрутки снизу/справа, если не все элементы могут отобразиться в окне, размер которого определяется параметрами Ширина и Высота, которые появляются, если был выбран данный способ отображения.

Параметр	Назначение
Текст заголовка	<p>Определяет текст заголовка всплывающего окна. Если текст заголовка не задан или не установлена связь с параметром, то используется название объекта/узла, которому принадлежит окно.</p>
Фон заголовка	<p>Определяет цвет фона заголовка. При нажатии на кнопку , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.</p>
Цвет текста заголовка	<p>Определяет цвет текста заголовка. При нажатии на кнопку , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.</p>
Разрешить закрытие	<p>Определяет возможность закрытия окна в клиенте визуализации.</p>
Контроль границ	<p>Определяет возможность перемещения/открытия окна за границами видимости клиента визуализации.</p>
Контроль дубликатов	<p>Определяет необходимость закрытия ранее открытых копий и экземпляров данного окна.</p> <p>Если задано значение TRUE и при этом открывается окно объекта, то будут закрыты все ранее открытые копии этого окна.</p> <p>Если задано значение TRUE и при этом открывается окно экземпляра объекта, то будут закрыты все ранее открытые окна любых экземпляров этого объекта.</p>
Категория	<p>Задаёт категорию открываемому окну. При открытии окна ему присваивается указанная категория и если будет открыто другое окно с такой же категорией, то ранее открытое окно с такой же категорией будет закрыто. Например, в проекте есть несколько экземпляров задвижек, у всех есть окна управления, при открытии которых им присваивается одна и та же категория, заданная в</p>

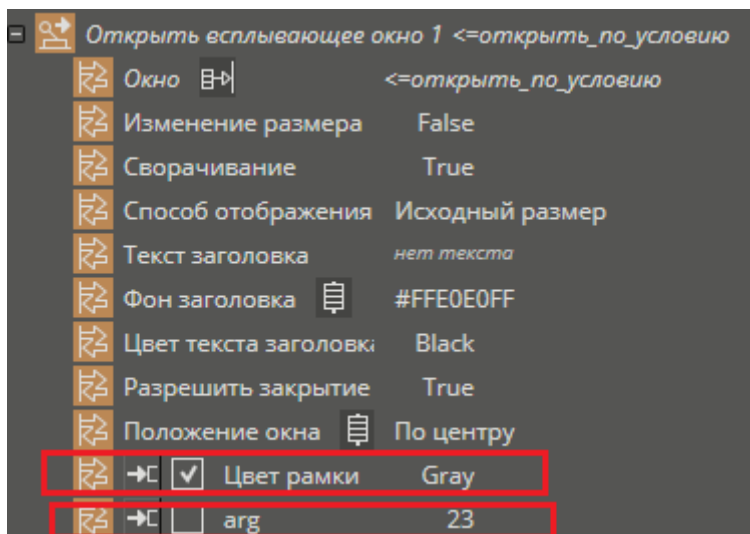
Параметр	Назначение
	библиотечном типе. Тогда в один момент времени может быть открыто только одно окно управления какой-либо задвижкой.
Положение окна	<p>Определяет место открытия окна. Возможные варианты:</p> <ul style="list-style-type: none"> • По центру экрана; • По месту указателя мыши (по умолчанию); • По заданным координатам; • С заданным смещением относительно элемента; • С заданным смещением относительно окна элемента. <p>В случае выбора трёх последних вариантов, появляются поля ввода координат, в которых либо может быть задана константа, либо эти поля могут быть связаны с параметрами проекта.</p>
Ширина	<p>Определяет ширину всплывающего окна, если при настройке параметра Способ отображения был выбран способ отличный от Исходный размер. Значение параметра проверяется только в момент открытия окна. Если параметр изменился после того, как окно было открыто размер окна не изменится.</p>
Высота	<p>Определяет высоту всплывающего окна, если при настройке параметра Способ отображения был выбран способ отличный от Исходный размер. Значение параметра проверяется только в момент открытия окна. Если параметр изменился после того, как окно было открыто размер окна не изменится.</p>

Дополнительные параметры открытия окна

По команде Добавить контекстного меню действия Открыть окно, в действие можно добавить параметры (аргументы) окна (в т.ч. созданные пользователем) и задать для них либо связи, либо постоянные значения:



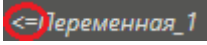
В результате выполнения команды **Добавить** в группу события добавляется строка аргумента:



Инструменты строки аргумента:

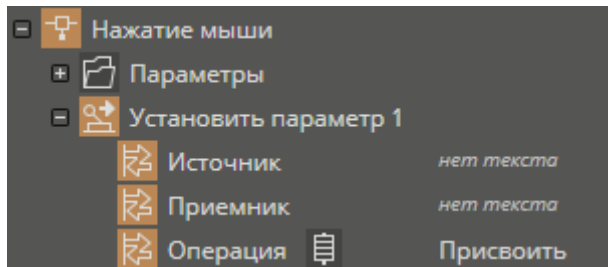
- Аргумент вызова / ⇨ Результат вызова – переключатель типа аргумента. Соответственно: входной аргумент/выходной аргумент;
- / – способ передачи. Соответственно: однократно/по изменению;
- 23 – в этом поле для аргумента задается постоянное значение, а если для аргумента задается связь (связь имеет более высокий приоритет по сравнению с постоянным значением), то вместо поля отображается имя привязанного параметра с указанием направления передачи значения:

 — из параметра в аргумент;

 — из аргумента в параметр.

8.2.3.3.2.6. ДЕЙСТВИЕ УСТАНОВИТЬ ПАРАМЕТР

При задании этого действия группа события принимает следующий вид (на рисунке показано событие Нажатие мыши в окне свойств; см. также НМІ. Действия):

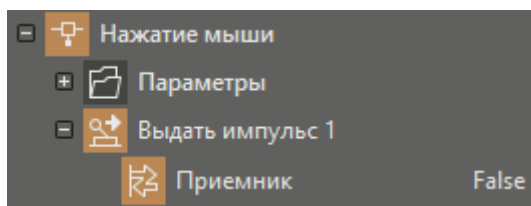


Действие заключается в выполнении математического действия, заданного параметром Операция:

- если Операция=Присвоить, то Приемник := Источник;
- при других операциях Приемник := Приемник <операция> Источник (например, при Операция=Разделить Приемник := Приемник/Источник).

8.2.3.3.2.7. ДЕЙСТВИЕ ВЫДАТЬ ИМПУЛЬС

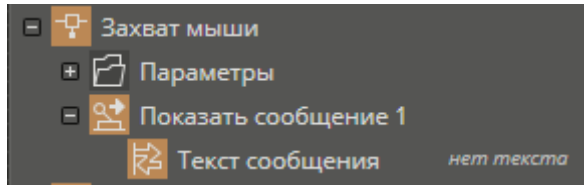
При задании этого действия группа события принимает следующий вид (на рисунке показано событие Нажатие мыши в окне свойств; см. также НМІ. Действия):



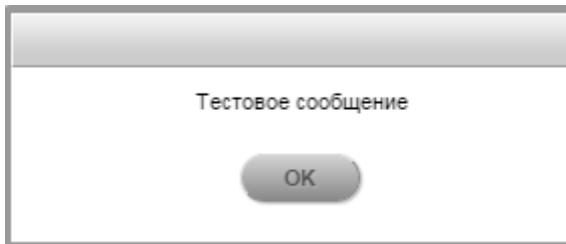
Действие заключается в установлении значения True параметра, связанного с Приемником, на один цикл, на следующем цикле связанный параметр примет значение False. Цикл зависит от периода задачи, в которой выполняется связанный параметр.

8.2.3.3.2.8. ДЕЙСТВИЕ ПОКАЗАТЬ СООБЩЕНИЕ

При задании этого действия группа события принимает следующий вид (на рисунке показано событие Захват мыши в окне свойств; см. также НМІ. Действия):

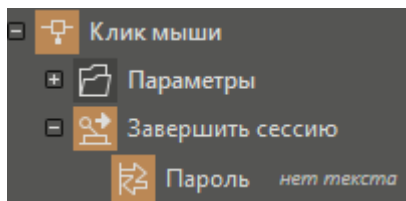


Ниже показан вид сообщения при возникновении события в реальном времени (чтобы закрыть сообщение, нужно нажать на кнопку ОК):



8.2.3.3.2.9. ДЕЙСТВИЕ ЗАВЕРШИТЬ СЕССИЮ

При задании этого действия группа события принимает следующий вид (на рисунке показано событие Нажатие мыши , см. также НМІ. Действия):



Если действие задано, то при работе в реальном времени при возникновении события сессия завершается, и открывается окно авторизации:

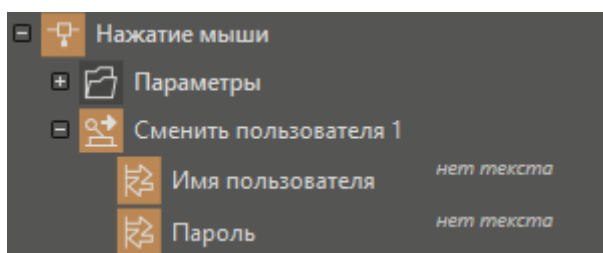
A white login window with a blue border. It has a label 'Логин' (Login) above a dropdown menu showing 'Пользователь 1' (User 1). Below it is a label 'Пароль' (Password) above an empty text input field. At the bottom is a large blue button with the text 'Вход' (Login).

Если в настройках Безопасности установлен флаг Запрашивать пароль при выходе из системы, то завершение сессии и открытие окна авторизации произойдет только в том случае, если значение параметра действия Пароль будет совпадать с паролем текущего

пользователя. Если пароль пользователя не будет совпадать с параметром действия, то останется авторизованным текущий пользователь. Авторизоваться, например, при помощи кнопки с настроенным действием Сменить пользователя сможет только тот пользователь, у которого есть право доступа Изменение пользователей. При этом в журнале появится сообщение - Принудительный выход подтвержден <новый_пользователь>.

8.2.3.3.2.10. ДЕЙСТВИЕ СМЕНИТЬ ПОЛЬЗОВАТЕЛЯ

При задании этого действия группа события принимает следующий вид (на рисунке показано событие Нажатие мыши, см. также НМІ. Действия):



Параметры Имя пользователя и Пароль могут быть связаны с параметрами проекта, а также заданы в виде констант.

Если действие задано, то при возникновении события в реальном времени сессия текущего пользователя завершится, и начнется сессия пользователя, данные которого переданы в параметры действия. Стандартное окно авторизации открываться не будет. Если параметры действия примут значения, которые не соответствуют существующим пользователям, то откроется окно авторизации, при этом сессия предыдущего пользователя будет продолжена до тех пор, пока не пройдет авторизация нового.

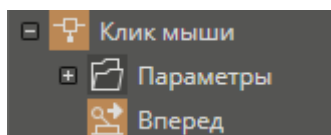
После логина нового пользователя остаются окна открытые ранее. В случае, если прав на изменение разных элементов открытых окон у пользователя нет, то эти элементы будут недоступны для управления. Если у нового пользователя нет прав на открытие ранее


открытых всплывающих окон, то эти окна будут закрыты. Если основное окно недоступно для открытия, то откроется стартовое окно.

Важно! Действие Сменить пользователя работает только в НМІ v2.

8.2.3.3.2.11. ДЕЙСТВИЕ ВПЕРЕД

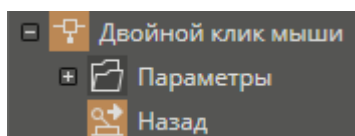
При задании этого действия группа события принимает следующий вид (на рисунке показано событие Клик мыши , см. также НМІ. Действия):




Если действие задано, то при работе в реальном времени при возникновении события выполняется команда  Вперед браузера (т.е. если в браузере существует история переходов по страницам, то данная команда – это переход на следующую страницу согласно истории переходов).

8.2.3.3.2.12. ДЕЙСТВИЕ НАЗАД

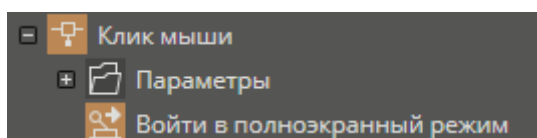
При задании этого действия группа события принимает следующий вид (на рисунке показано событие Двойной клик мыши , см. также НМІ. Действия):



Если действие задано, то при работе в реальном времени при возникновении события выполняется команда  Назад браузера (т.е. если в браузере существует история переходов по страницам, то данная команда – это переход на предыдущую страницу согласно истории переходов).

8.2.3.3.2.13. ДЕЙСТВИЕ ВОЙТИ В ПОЛНОЭКРАННЫЙ РЕЖИМ

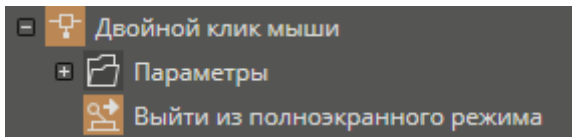
При задании этого действия группа события принимает следующий вид (на рисунке показано событие Клик мыши, см. также НМІ. Действия):



Если действие задано, то при работе в реальном времени при возникновении события применяется полноэкранный режим отображения.

8.2.3.3.2.14. ДЕЙСТВИЕ ВЫЙТИ ИЗ ПОЛНОЭКРАННОГО РЕЖИМА

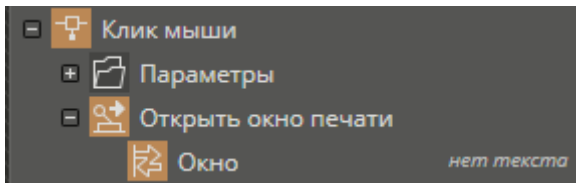
При задании этого действия группа события принимает следующий вид (на рисунке показано событие Двойной клик мыши , см. также НМІ. Действия):



Если действие задано, то при работе в реальном времени при возникновении события применяется обычный (не полноэкранный) режим отображения.

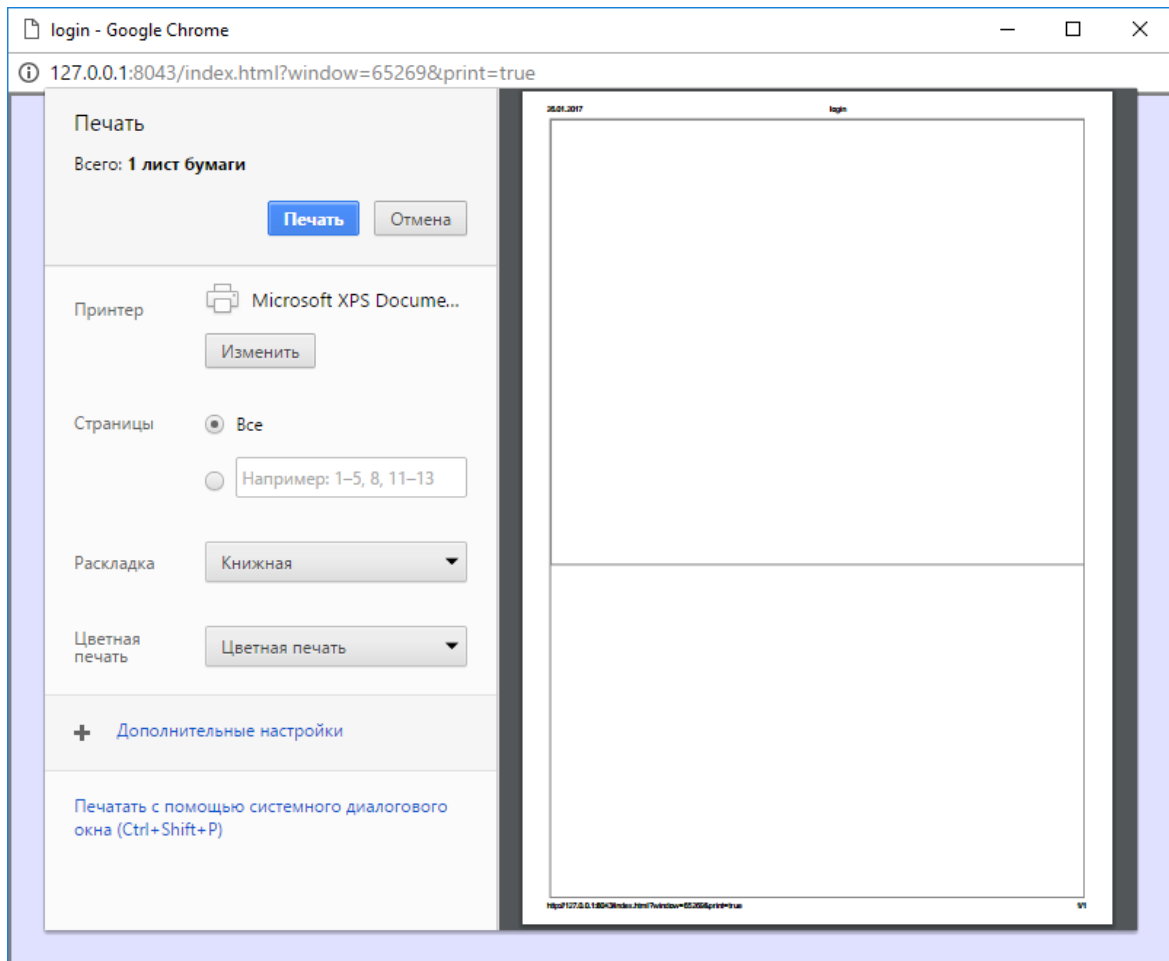
8.2.3.3.2.15. ДЕЙСТВИЕ ОТКРЫТЬ ОКНО ПЕЧАТИ

При задании этого действия группа события принимает следующий вид (на рисунке показано событие Клик мыши в окне свойств; см. также НМІ. Действия):



Действие заключается в выводе на печать окна, которое привязано к параметру Окно.

Появление окна печати определяется настройками, выполненными в MasterSCADA 4D Monitor или MasterSCADA 4D Client Monitor. Пример вида окна печати:



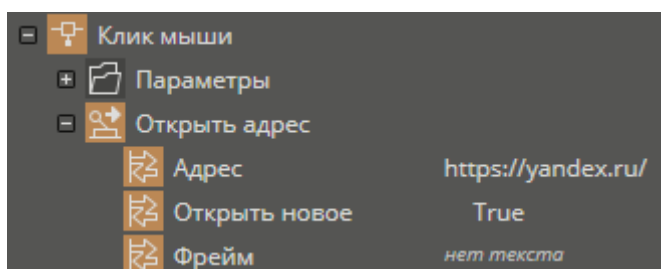
Работа в разных версиях НМІ

v1 - можно напечатать любое окно, которое ранее было открыто. Если окно не было открыто в клиенте, то печать не происходила.

v2 - можно напечатать только текущее окно.

8.2.3.3.2.16. ДЕЙСТВИЕ ОТКРЫТЬ АДРЕС

При задании этого действия группа события принимает следующий вид (на рисунке показано событие Клик мыши в окне свойств; см. также НМІ. Действия):

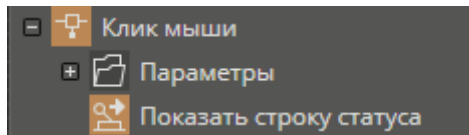


Данное действие – это действие Открыть окно, совмещенное с отображением а нем ресурса с адресом URL, заданным параметром Адрес.

Данное действие используется также для того, чтобы открывать отчеты, сгенерированные генератором отчетов MasterSCADA 4D.

8.2.3.3.2.17. ДЕЙСТВИЕ ПОКАЗАТЬ СТРОКУ СТАТУСА

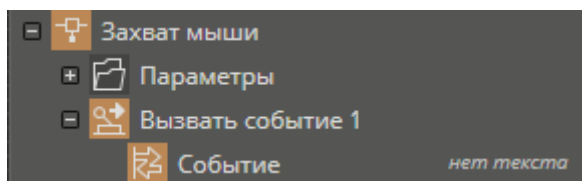
При задании этого действия группа события принимает следующий вид (на рисунке показано событие Клик мыши , см. также НМІ. Действия):



Если действие задано, то при работе в реальном времени при возникновении события отображается строка статуса (о строке статуса см. Отображение мнемосхем в браузере).

8.2.3.3.2.18. ДЕЙСТВИЕ ВЫЗВАТЬ СОБЫТИЕ

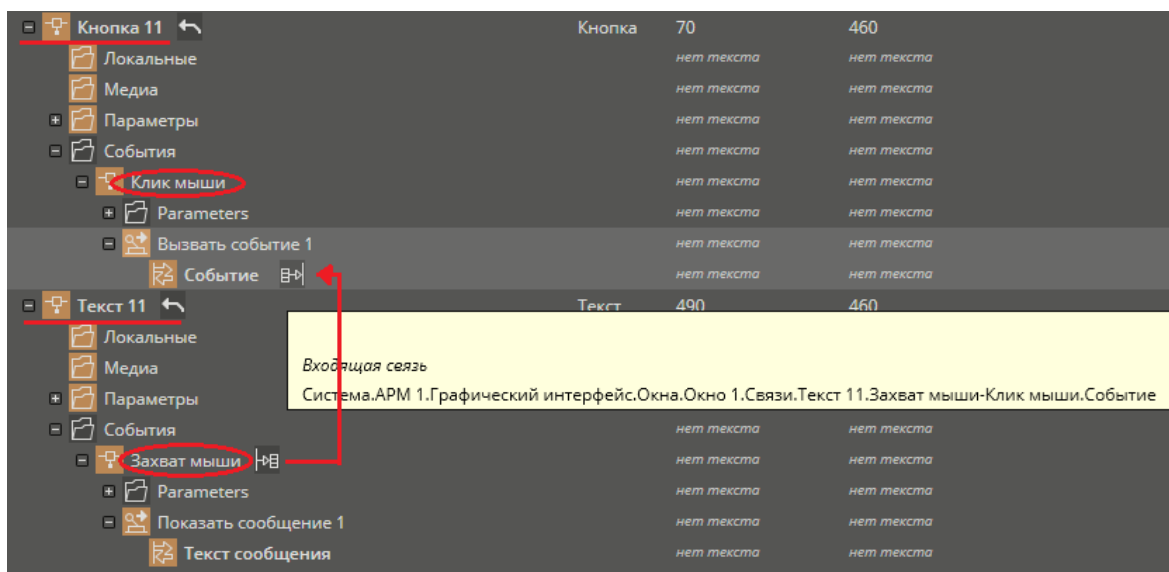
При задании этого действия группа события принимает следующий вид (на рисунке показано событие Захват мыши , см. также НМІ. Действия):



К параметру Событие привязывается некоторое событие.

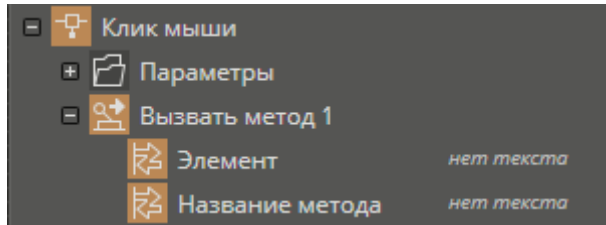
Пример

В примере ниже при событии Клик мыши кнопки Кнопка 11 наступает событие Захват мыши элемента Текст 11 (и в результате выполняется действие Показать сообщение).

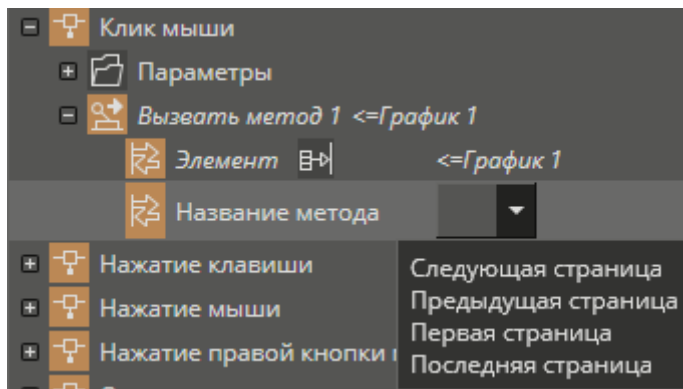


8.2.3.3.2.19. ДЕЙСТВИЕ ВЫЗВАТЬ МЕТОД

При задании этого действия группа события принимает следующий вид (на рисунке показано событие Клик мыши в окне свойств; см. также НМІ. Действия):



К параметру Элемент привязывается графический элемент, после чего в списке Название метода выбирается один из методов, определенных в привязанном графическом элементе:

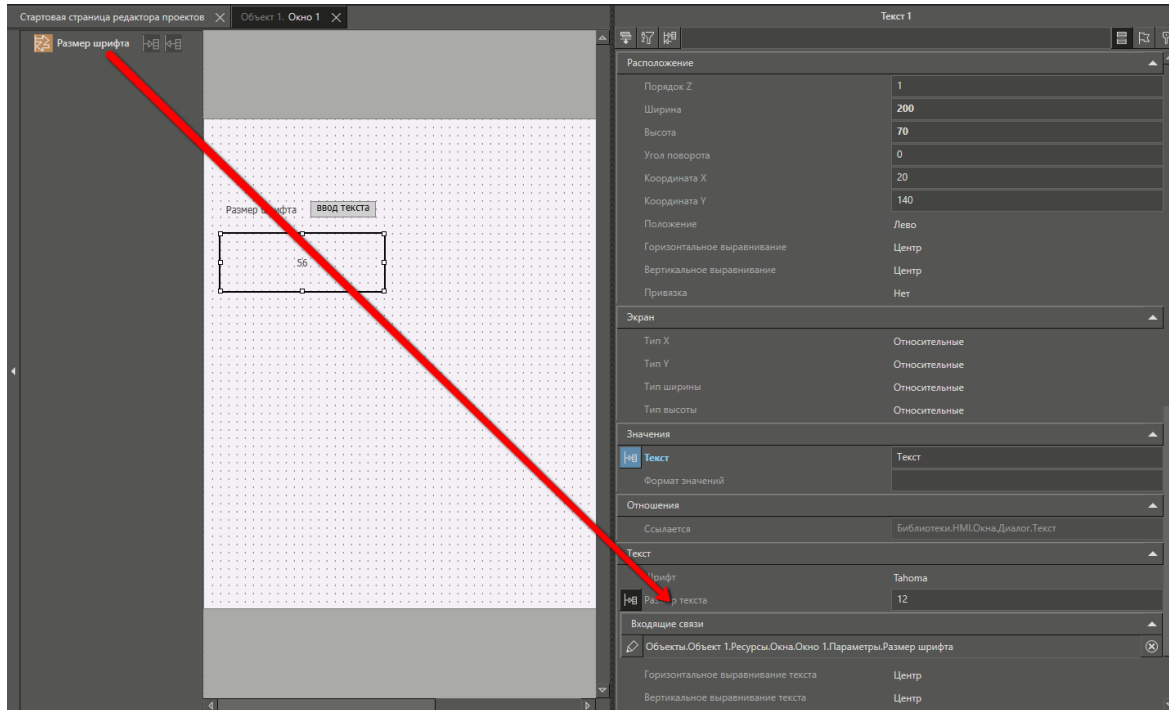


По команде Добавить контекстного меню действия Вызвать метод в действие можно добавить аргумент.

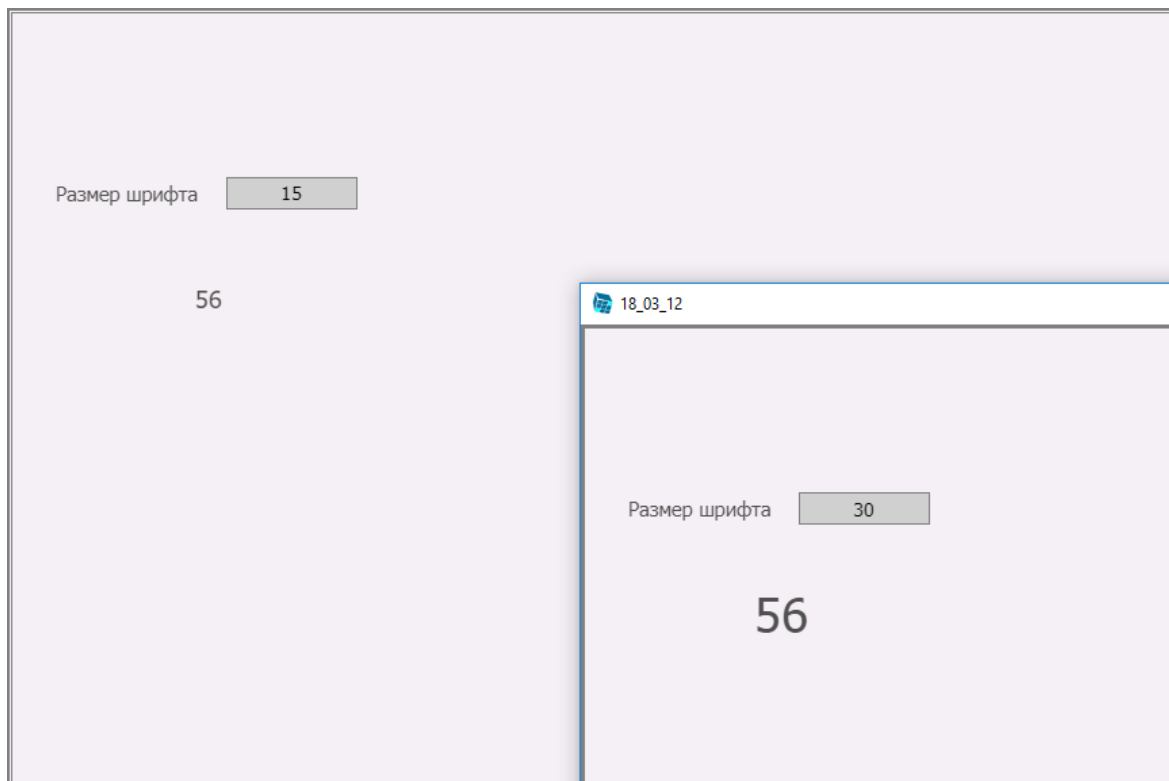
8.2.4. КЛЕММНИКИ В РЕДАКТОРЕ НМІ

Клеммник входов – это группа Параметры мнемосхемы (содержит параметры, созданные разработчиком мнемосхемы). Эти параметры обрабатываются в клиенте визуализации. Если параметры не имеют входящей связи от сервера, то может сложиться ситуация, когда у разных клиентов визуализации параметры будут иметь различные значения.

Например, свойство Размер текста связано с параметром окна Размер шрифта.



Если в режиме исполнения открыто несколько клиентов визуализации одновременно, то в каждом из них могут быть заданы разные размеры шрифта.



Кроме того, Клеммники окна используются при создании библиотечных окон. Когда в библиотеке создается окно, то определяются параметры, которые необходимо использовать для динамизации. Затем, когда экземпляр окна добавляется в объект, настраивается связь между клеммником окна и параметром объекта.

8.2.5. МЕНЕДЖЕР РЕСУРСОВ НМІ

Ресурсы - это дополнительные файлы, которые используются при создании графического интерфейса для пользователей. Например, звуки, рисунки, видео.

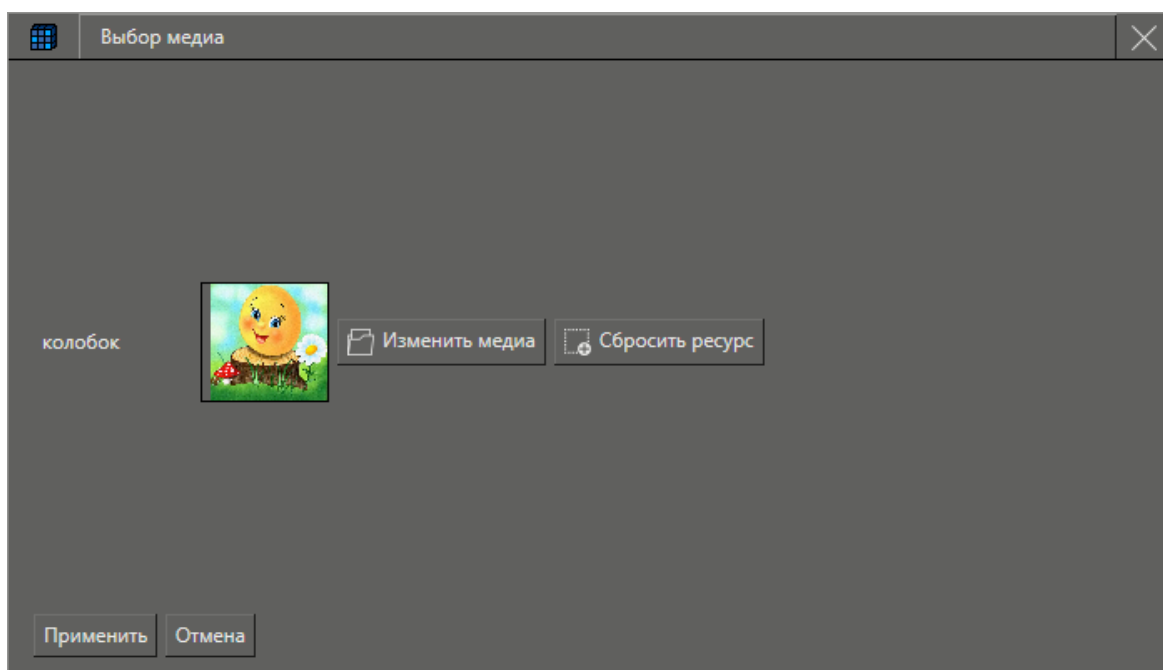
Разделяют три уровня добавления ресурсов в проект: в Узел, в Окно, в Элемент.

Ресурсы узла и окна

Ресурсы узла добавляются в элемент Медиа группы Графический интерфейс в дереве системы. Ресурсы узла видны и могут использоваться в любом графическом элементе проекта.

Ресурсы окна, как правило, добавляются в Легенде во вкладке Медиа. Ресурсы добавленные в окно будут доступны только для элементов текущего окна.

При добавлении ресурсов в узел или в окно автоматически откроется диалоговое окно Windows, в котором нужно выбрать медиа-файл. Добавится новое медиа, имя медиа будет совпадать с именем файла. Файл медиа добавится в базу данных проекта. Отредактировать добавленное медиа можно дважды кликнув по нему. Откроется диалоговое окно:




Имя медиа можно отредактировать дважды кликнув по его названию.

Кнопка Изменить медиа позволяет изменить выбранный файл. Кнопка Сбросить ресурс удалит файл из базы проекта.

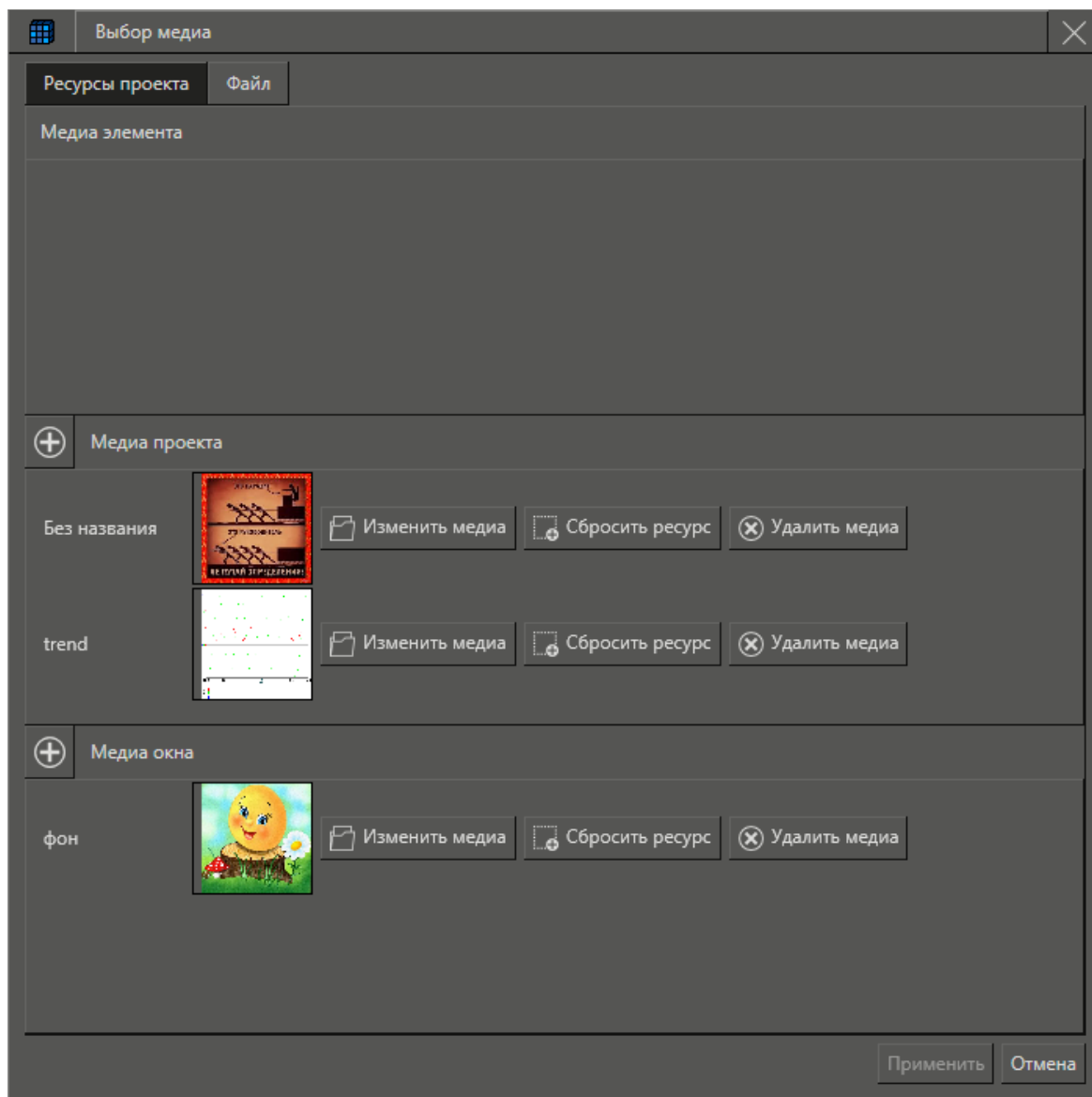
Ресурсы элементов

Работа с ресурсами элемента производится в соответствующих его свойствах. Например, в свойстве Изображение/Изображение фона категории Внешний вид.


После нажатия на кнопку  в строке свойства, откроется диалоговое окно, состоящее из двух вкладок: Ресурсы проекта и Файл.

Вкладка Ресурсы проекта

Вид вкладки:

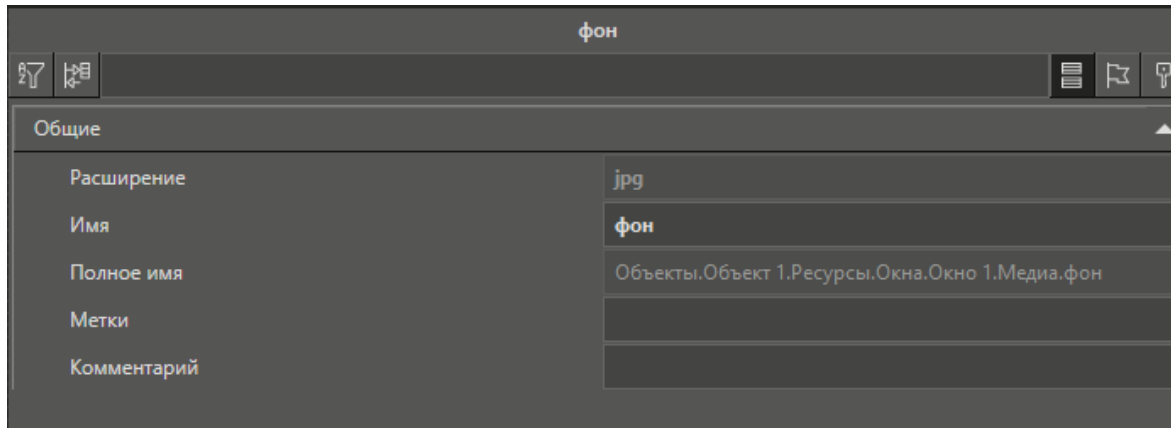


Позволяет выбрать медиа ранее добавленные в узел (группа Медиа проекта) или окно (группа Медиа окна) или добавить новое медиа, которое будет относиться только к текущему элементу. Если в проект добавлено несколько узлов, то группе Медиа проекта отображаются медиа того узла, в задаче которого будет работать объект, которому принадлежит окно с настраиваемым элементом.

В данном окне можно изменить, переопределить или вовсе удалить выбранные ранее медиа файлы. Можно добавить новые медиа, нажав на кнопку  в нужной группе. Новые медиа появятся соответственно или в узле, или в окне.

Чтобы определить, какое медиа будет использоваться в настраиваемом свойстве элемента, нужно выделить это медиа левой кнопкой мыши и нажать кнопку Применить.

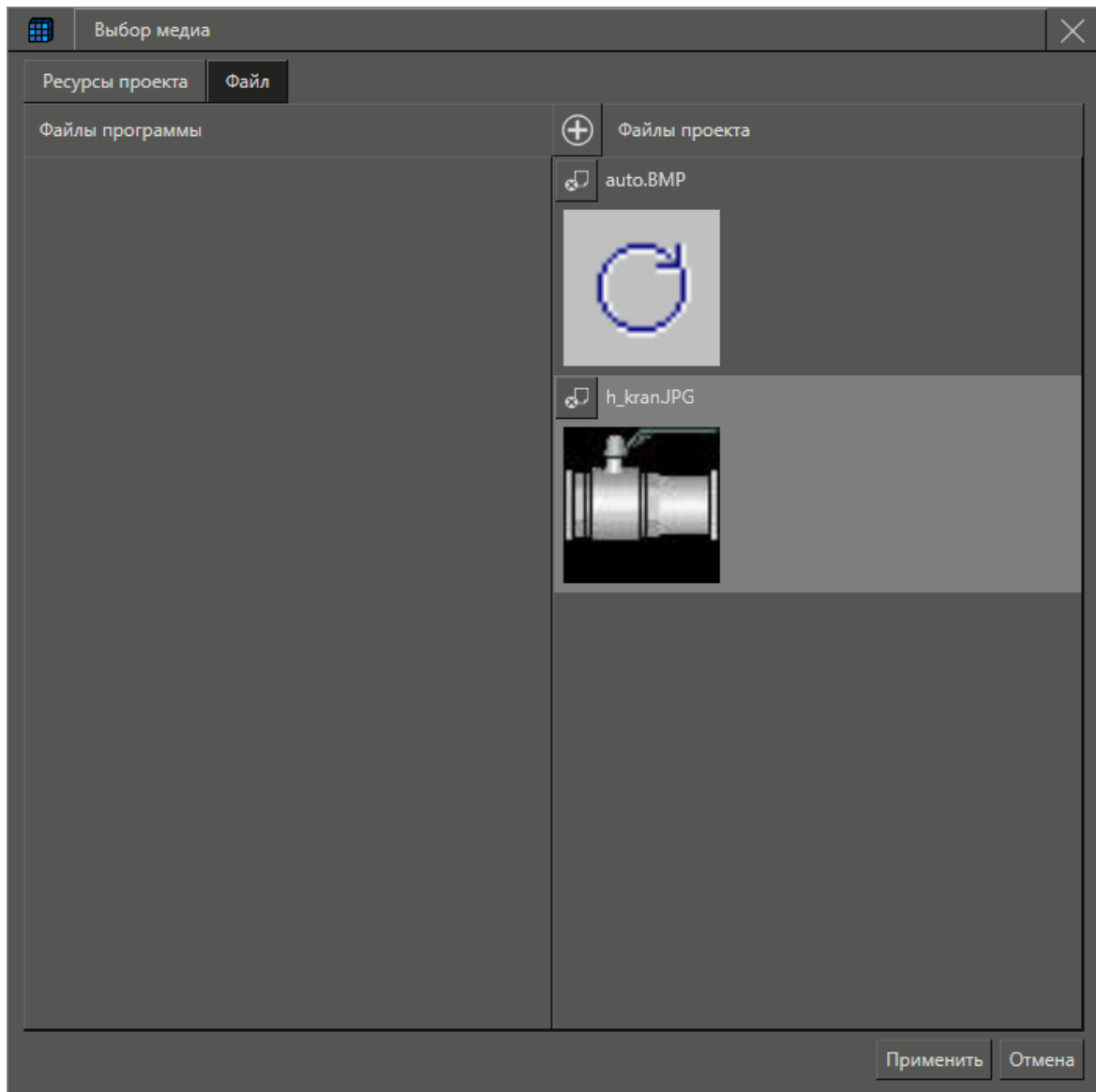
При выделении медиа в окне, в панели свойств отображаются его свойства, где можно посмотреть расширение, ранее добавленного файла:



После компиляции проекта ресурсы будут добавлены в папку htdocs подпапку *resources* скомпилированного проекта. Имена файлов не будут совпадать с именами добавляемых файлов.


Вкладка Файл

Вид вкладки:



Группа Файлы программы в текущей версии не используется.

В группе файлы проекта находится список файлов добавленных в папку *c:\Users\Public\Documents\MasterSCADA4D [номер версии]\Projects\[имя проекта]\Res*

После нажатия на кнопку  откроется окно Windows для выбора произвольного файла. Выбранный в диалоговом окне файл будет добавлен в папку *c:\Users\Public\Documents\MasterSCADA4D [номер версии]\Projects\[имя проекта]\Res*

Чтобы определить, какой файл будет использоваться в настраиваемом свойстве элемента, нужно выделить этот файл левой кнопкой мыши и нажать кнопку Применить.

После компиляции проекта файлы будут добавлены в папку htdocs подпапку *resources* скомпилированного проекта. Имена файлов будут совпадать с именами добавляемых файлов.

Использование не графических файлов

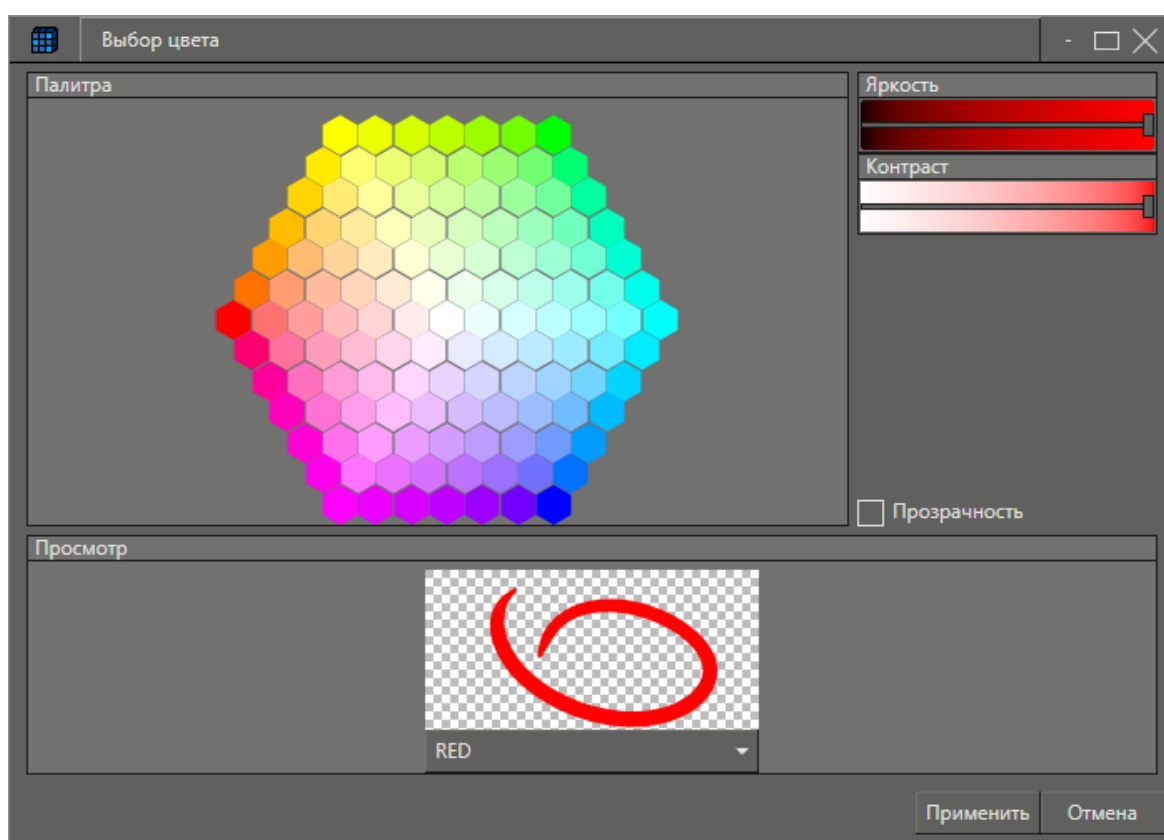
В ресурсы можно добавить произвольные файл, например, pdf или json. При компиляции проекта они помещаются в корень папки htdocs с исходными именами. Можно добавлять как на вкладке Файл, так и во вкладке Ресурсы проекта.

8.2.6. ДИАЛОГОВОЕ ОКНО ВЫБОР ЦВЕТА

В зависимости от настраиваемого свойства может открыться либо простой диалог выбора цвета, либо диалог выбора цвета с выбором градиента.

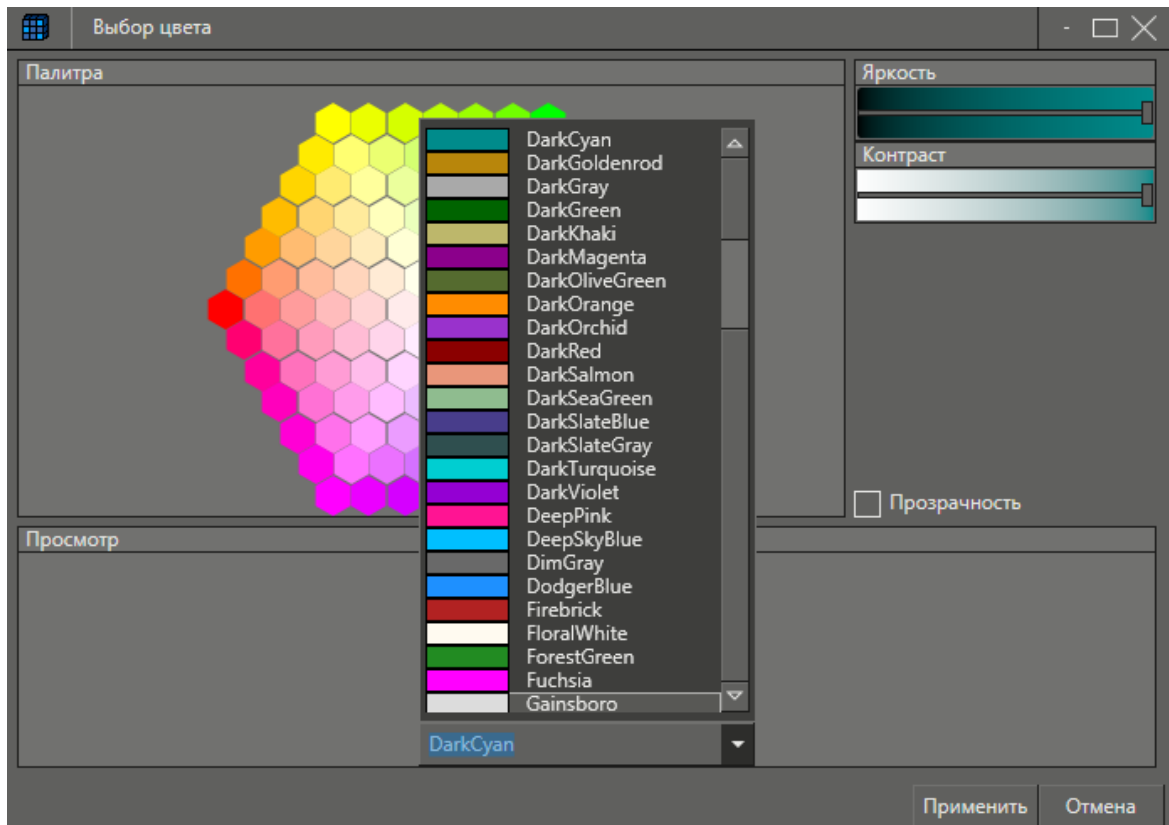
Простой диалог

Простой диалог выбора цвета открывается для настройки цветов линий и текстов, и позволяет выбрать необходимый цвет, например, рамки элемента:



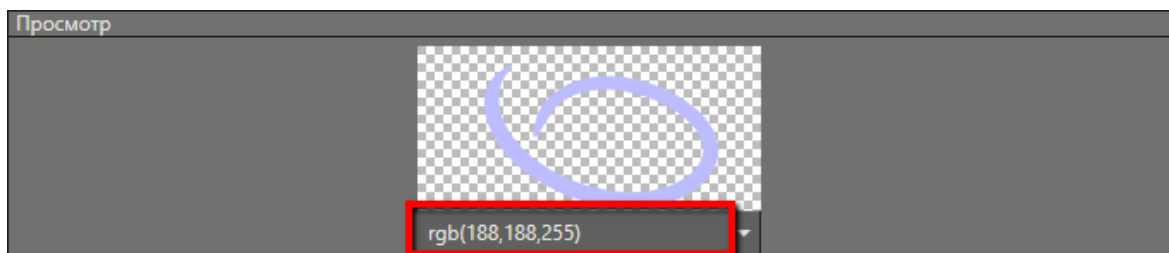
Порядок работы

Сначала необходимо выбрать базовый цвет при помощи области Палитра, либо воспользовавшись выпадающим списком в области Просмотр:



Затем, при необходимости, настраиваются Яркость и Контраст при помощи соответствующих ползунков.

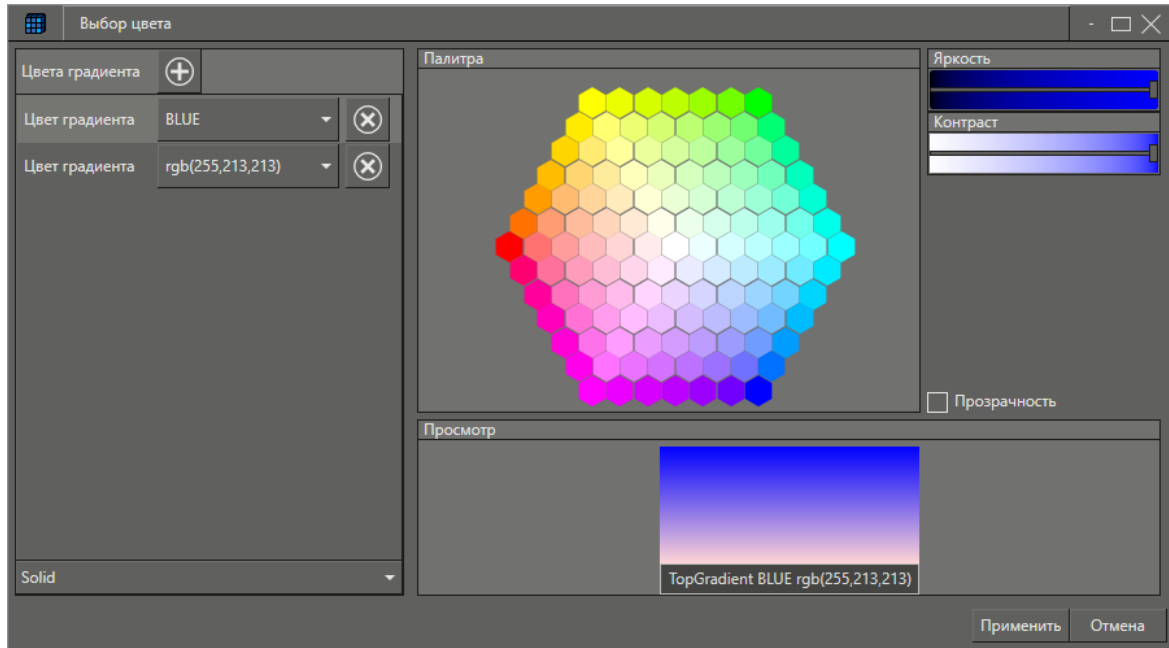
Кроме того, имеется возможность задать конкретный цвет в формате RGB. Для этого в поле Просмотр следует ввести требуемое значение цвета:



В том случае, если требуется задать цвет элемента, указанный в техническом задании в формате HEX, необходимо воспользоваться таблицами соответствия цветов, которые можно найти в сети Интернет.

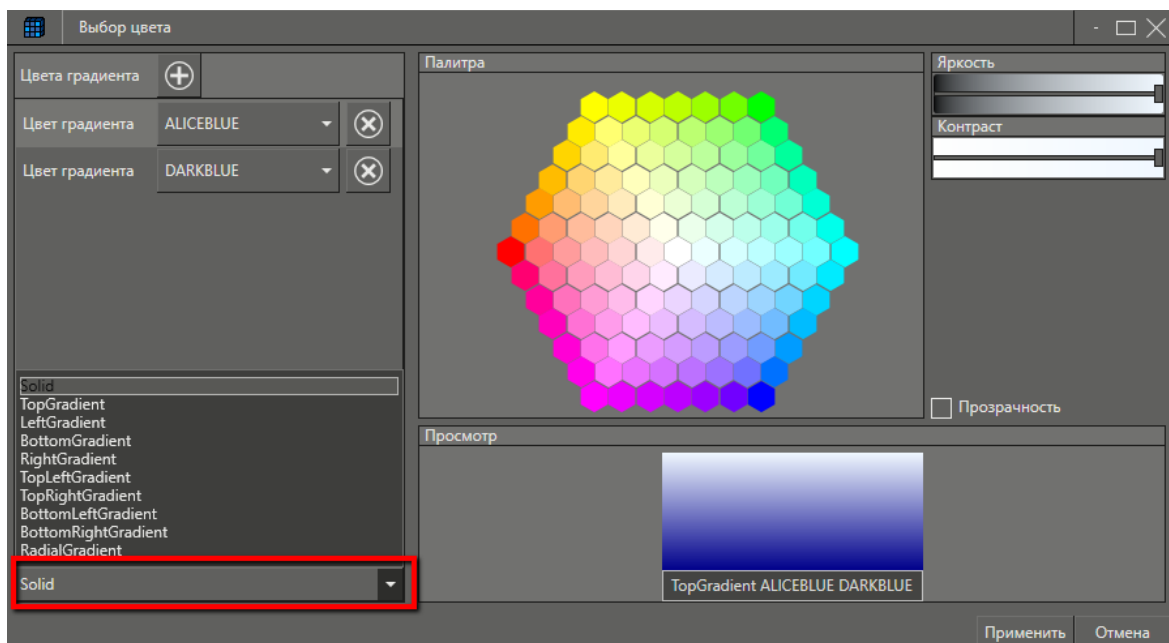
Диалог выбора цвета с настройкой градиента

Диалог выбора цвета с настройкой градиента, позволяет задать, например, заливку фона в случае, когда необходимо обеспечить плавный переход от одного цвета к другому.



Если фон требуется задать однотонным, то в левой части в поле Цвет градиента выбирается базовый цвет, так же как в простом диалоге в поле Просмотр, или при помощи поля Палитра, а затем, при необходимости, настраивается Яркость и Контраст.

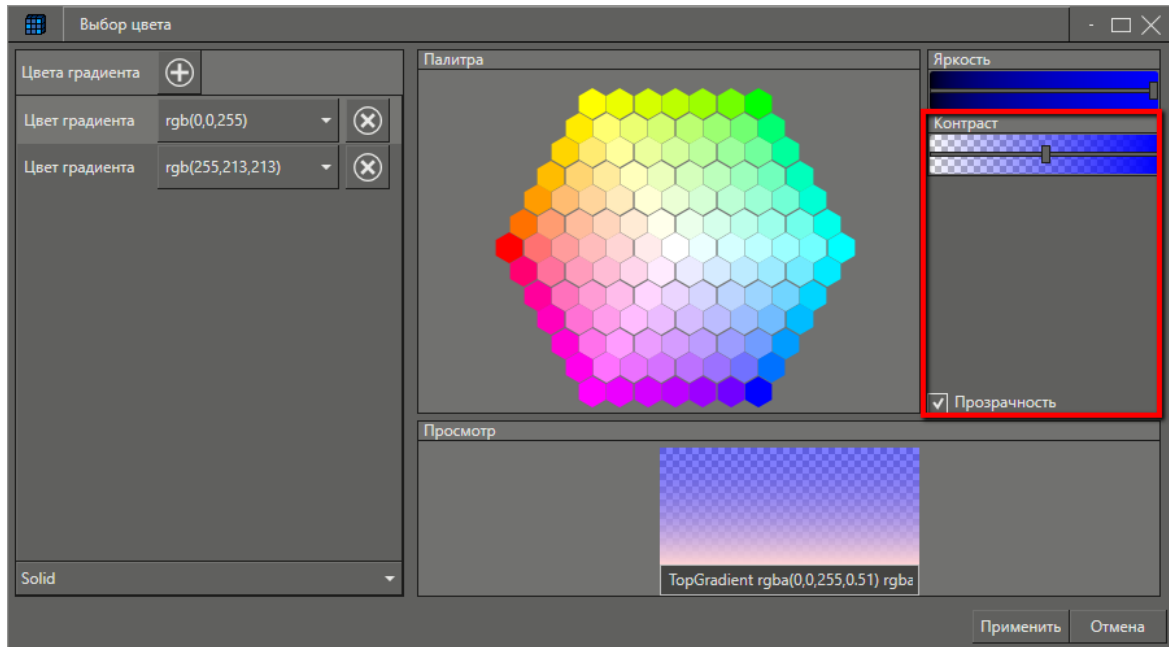
Если требуется реализовать градиент, то в левой части добавляется необходимое количество цветов градиента, а в выпадающем списке задается способ перехода от одного цвета к другому:



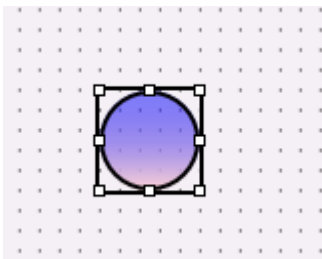
Настройка прозрачности

Если требуется сделать цвет прозрачным, то необходимо отметить флаг Прозрачность в правой части окна, в этом случае ползунок Контраст будет отвечать за прозрачность.

Кроме того, значение прозрачности можно указать в группе Просмотр. На рисунке ниже значение прозрачности равно 0.51, и при этом цвет наполовину прозрачный:

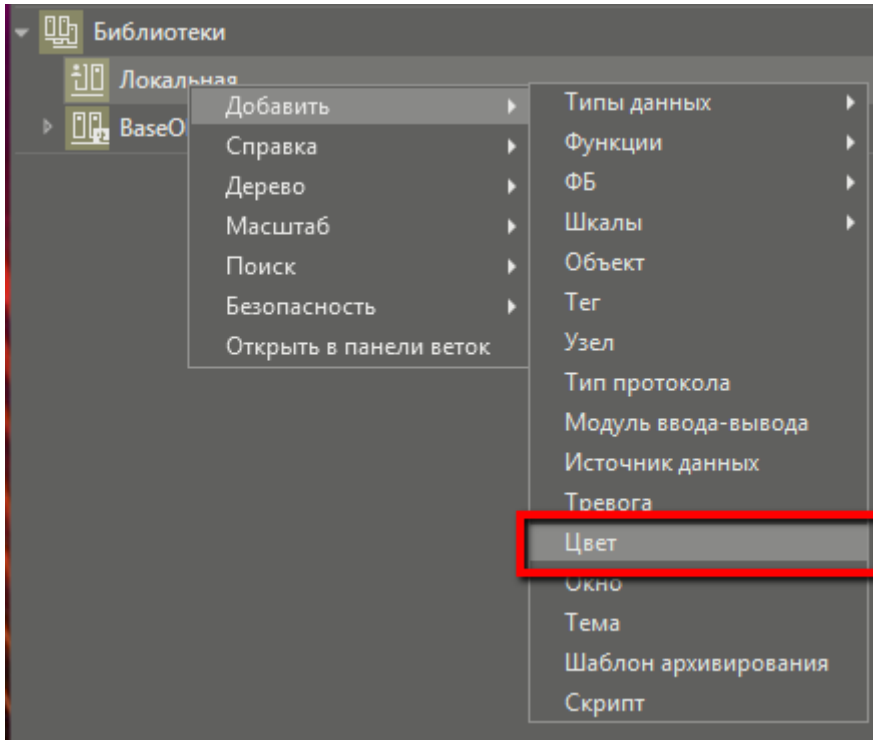


Результат:

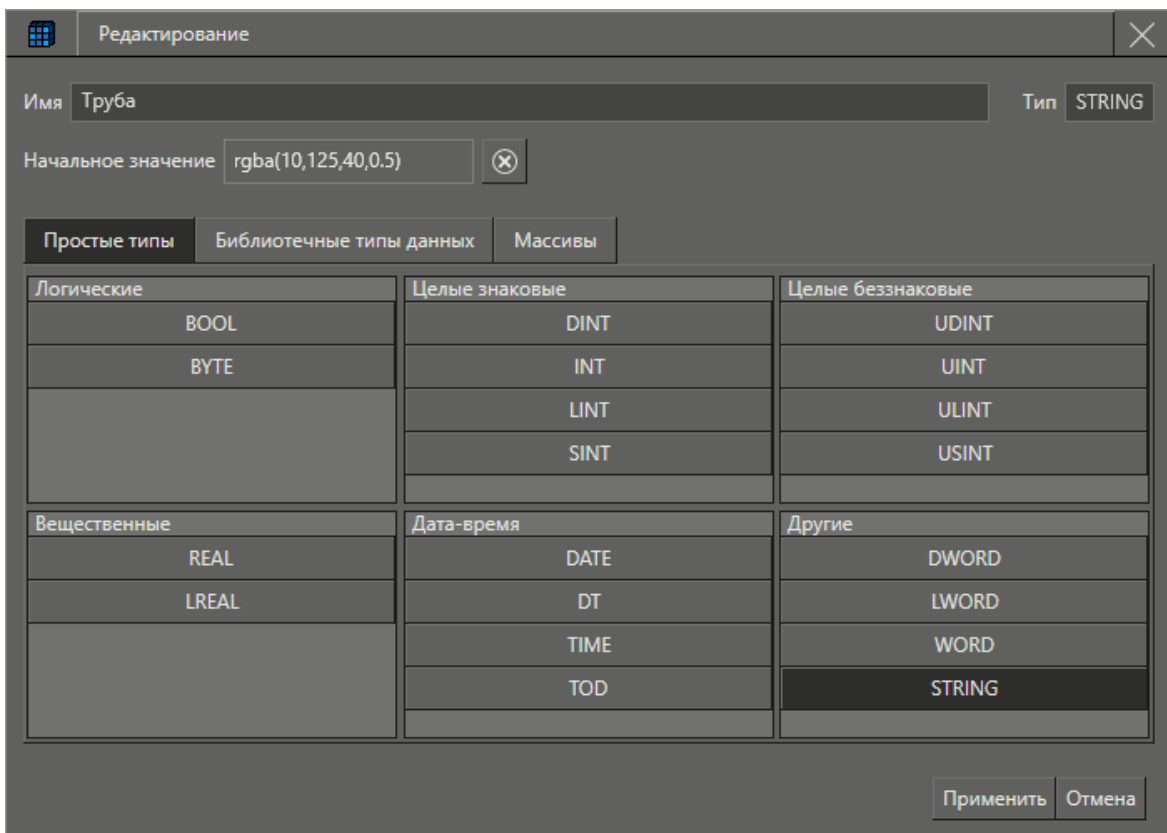


8.2.7. СОЗДАНИЕ ЦВЕТА ПОЛЬЗОВАТЕЛЯ

Для того чтобы создать цвет пользователя, необходимо в пользовательскую библиотеку добавить элемент Цвет:



При этом откроется диалоговое окно настройки параметра, в котором в поле Начальное значение следует указать значение цвета в формате RGBA:



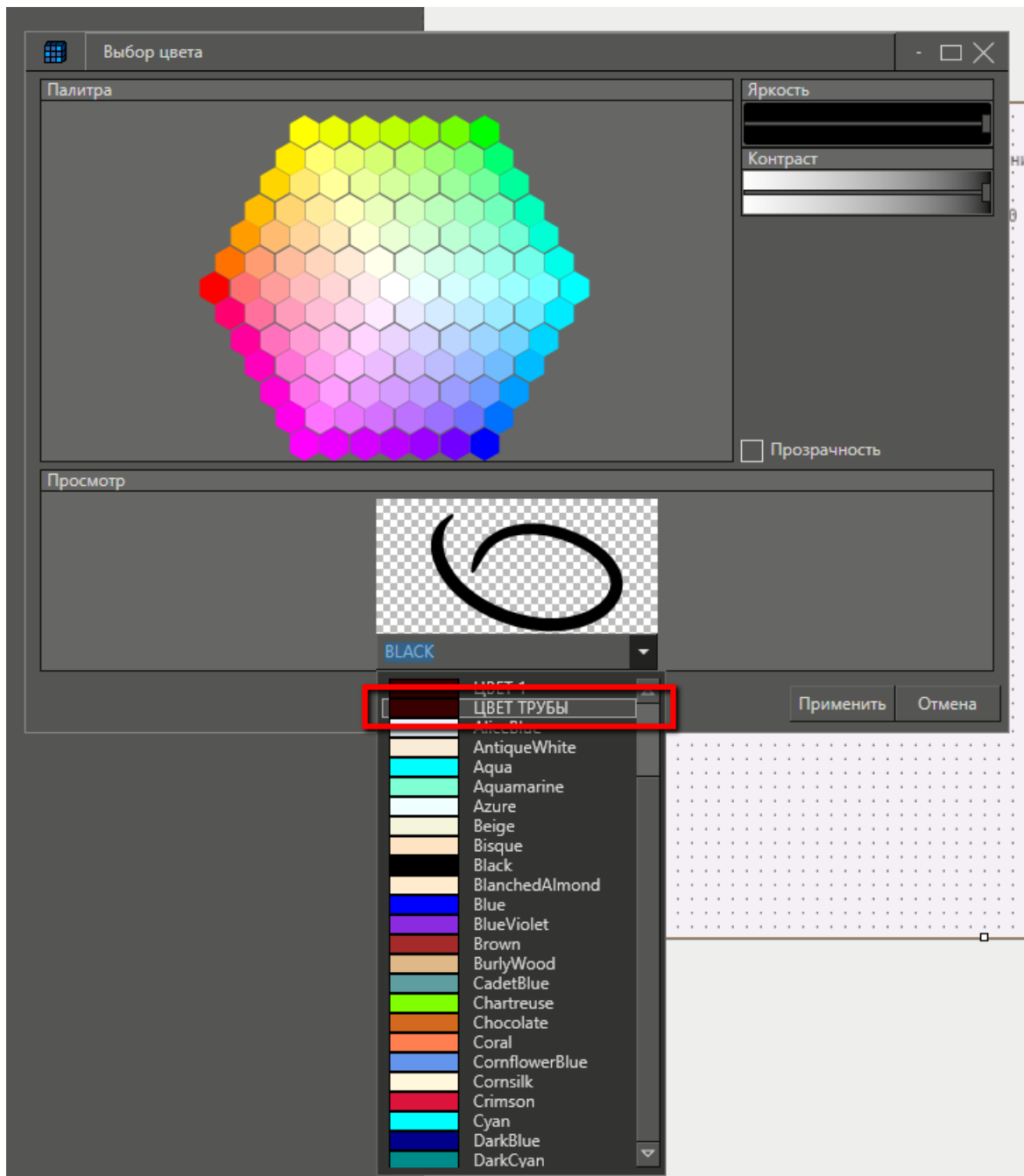
RGBA расшифровывается как Red Green Blue Alpha. Alpha - это значение прозрачности, при 1 цвет полностью не прозрачный, при 0 - абсолютно прозрачный.

При необходимости, измените поле Имя, чтобы в дальнейшем цветом было проще пользоваться.

Для того чтобы назначить цвет пользователя свойству элемента, необходимо написать имя цвета в соответствующем поле:



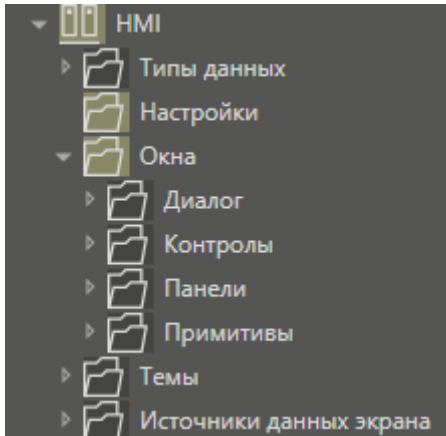
Либо выбрать в окне Выбор цвета:



Важно! Недопустимо перетаскивать элемент из библиотеки в поле свойства.

8.3. БИБЛИОТЕКА HMI

Библиотека HMI – это библиотека шаблонов графических и вспомогательных элементов, необходимых для работы в редакторе HMI.

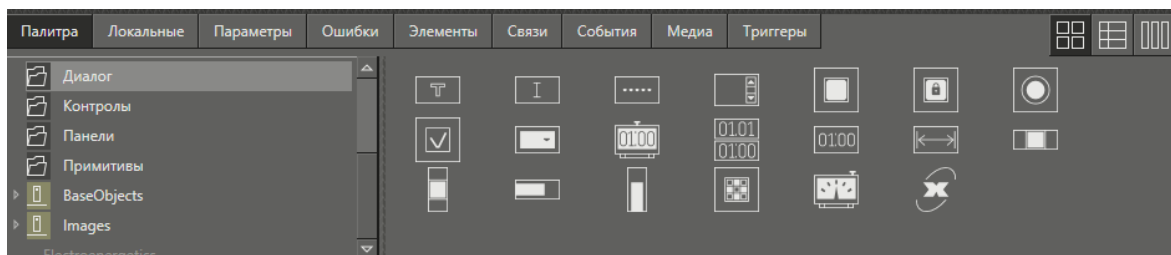


Элементы, содержащиеся в данной библиотеке доступны на вкладке Палитра в легенде редактора HMI, поэтому данная библиотека не отображается в упрощенном дереве.

8.3.1. КАТЕГОРИЯ ДИАЛОГ

Категория Диалог палитры редактора HMI содержит элементы, часто используемые для отображения полученных данных и для ввода данных в клиенте визуализации.

В палитре редактора HMI эта категория имеет вид:



Большинство элементов данной категории добавляются в окно при перетаскивании в него какого-либо параметра из дерева или из панели клемников. В этом случае все необходимые связи со свойствами элемента будут установлены автоматически. Если же элемент добавляется из палитры, то необходимо вручную установить входящие/исходящие связи между свойствами элементов и параметрами проекта.

Категория Диалог включает в себя следующие элементы:

- Текст
- Текстовый ввод
- Пароль
- Инкремент

- Кнопка
- Кнопка с фиксацией
- Радиокнопка
- Флаг
- Выпадающий список
- Часы
- ДатаВремя
- Время
- Интервал
- Горизонтальный ползунок
- Вертикальный ползунок
- Горизонтальный прогресс
- Вертикальный прогресс
- Кнопка выбора цвета
- Стрелочный прибор

8.3.1.1. ТЕКСТ

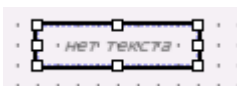
Данный предназначен для отображения значений параметров простых типов, а также констант (например, надписей, заголовков и т.п.). Чаще всего элементы Текст в окне создаются путем перетаскивания параметра в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.

В палитре редактора НМІ элемент Текст находится в категории Диалог.

Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора НМІ:

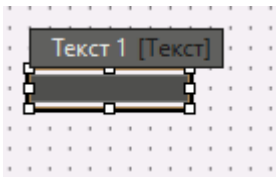


Если курсор на элементе не установлен, то он имеет рамку, которая будет не видна в клиенте визуализации, рамка служит для обозначения границ элемента:

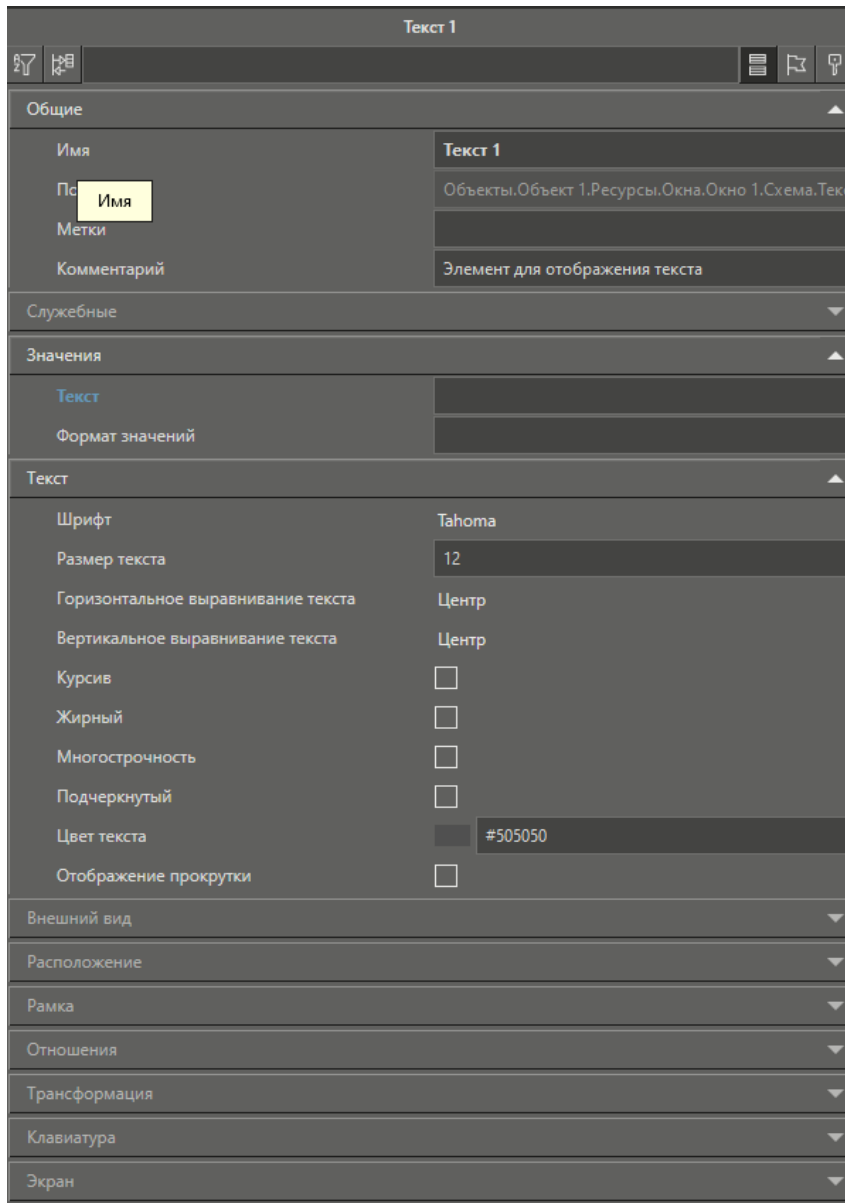


Если дважды кликнуть по элементу, то появится курсор и можно ввести текст, который будет отображаться в клиенте визуализации, т.е. таким образом можно изменить значение


свойства элемента Текст. Также режим редактирования включается после нажатием на кнопку клавиатуры F2. Вид элемента во время ввода текста:



Вид панели свойств элемента:



Описание основных свойств графического элемента Текст:

Название	Описание
Категория Значения	
Текст	В данном свойстве задается текст, который будет отображаться в клиенте визуализации. Допускается задавать константу. Если элемент получился в результате перетаскивания какого-либо параметра проекта в окно, либо если элемент сначала был добавлен из библиотеки, а затем параметр проекта был перетащен на свойство, то данная категория будет содержать дополнительные строки, в которых будет находиться информация о связях.
Формат значений	У данного элемента свойство доступно только при отображении всех свойств в панели (кнопка  отжата). Подробное описание свойства смотрите в разделе Формат значений.

Описание других свойств элемента смотрите в разделе НМІ.Свойства

Внешний вид элемента не отличается в НМІ v1 и в НМІ v2, и порядок работы с элементом одинаков.

8.3.1.2. ТЕКСТОВЫЙ ВВОД

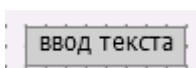
Данный элемент предназначен для ввода в клиенте визуализации значений параметров простых типов при помощи реальной или виртуальной клавиатуры. Чаще всего элементы Текстовый ввод в окне создаются путем перетаскивания параметра в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.

В палитре редактора НМІ элемент Текстовый ввод находится в категории Диалог.

Вид элемента в палитре:



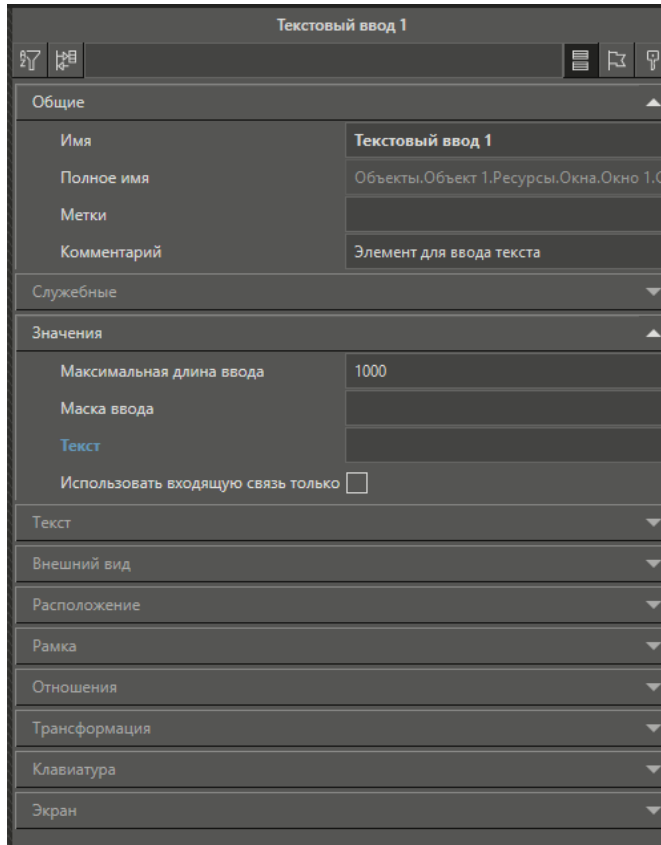
Вид элемента после добавления в рабочую область редактора НМІ:



Если дважды кликнуть по элементу, то появится курсор и можно ввести текст, который будет отображаться в клиенте визуализации, т.е. таким образом можно изменить значение свойства элемента Текст. Также режим редактирования включается после нажатием на кнопку клавиатуры F2. Вид элемента во время ввода текста:



Вид панели свойств элемента:



Описание основных свойств графического элемента **Текстовый ввод**:

Название	Описание
Категория Значения	
Максимальная длина ввода	Это свойство имеет тип INT и задает максимальное число символов во вводимой строке.
Маска ввода	Определяет символы, допустимые для ввода в данном поле (только цифры, только буквы или их определенная комбинация) Подробное описание свойства смотрите в соответствующем разделе.
Текст	Данное свойство должно быть связано с каким-либо параметром проекта. Если элемент был добавлен в окно из палитры редактора НМІ, то связи необходимо установить вручную. Если параметр

Название	Описание
	был перетащен правой кнопкой мыши в окно, и в контекстном меню был выбран пункт Текстовый ввод, то данное поле заполнится автоматически. После установки связи данное свойство будет содержать дополнительные строки, в которых будет находиться информация о связях.
Использовать входящую связь только для отображения	Определяет будет ли поступившее на вход свойства значение передано на выход. Если флаг установлен, то если изменяется значение по входной связи, то элемент отображается в соответствии с ним, но по выходной связи оно не идет. А если пользователь сам изменит текст, то значение передается по выходной связи, даже если значение совпадает с тем, что отправлялось на сервер в прошлый раз

Описание других свойств элемента смотрите в разделе НМІ.Свойства

Внешний вид элемента не отличается в НМІ v1 и в НМІ v2, и порядок работы с элементом одинаков.

8.3.1.3. ПАРОЛЬ

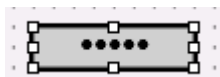
Данный элемент предназначен для ввода пароля в клиенте визуализации при помощи реальной или виртуальной клавиатуры. В отличие от других элементов, позволяющих вводить значения параметров в поле ввода, здесь вводимые данные маскируются точками. Чаще всего элементы Пароль в окне создаются путем перетаскивания параметра в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.

В палитре редактора НМІ элемент Пароль находится в категории Диалог.

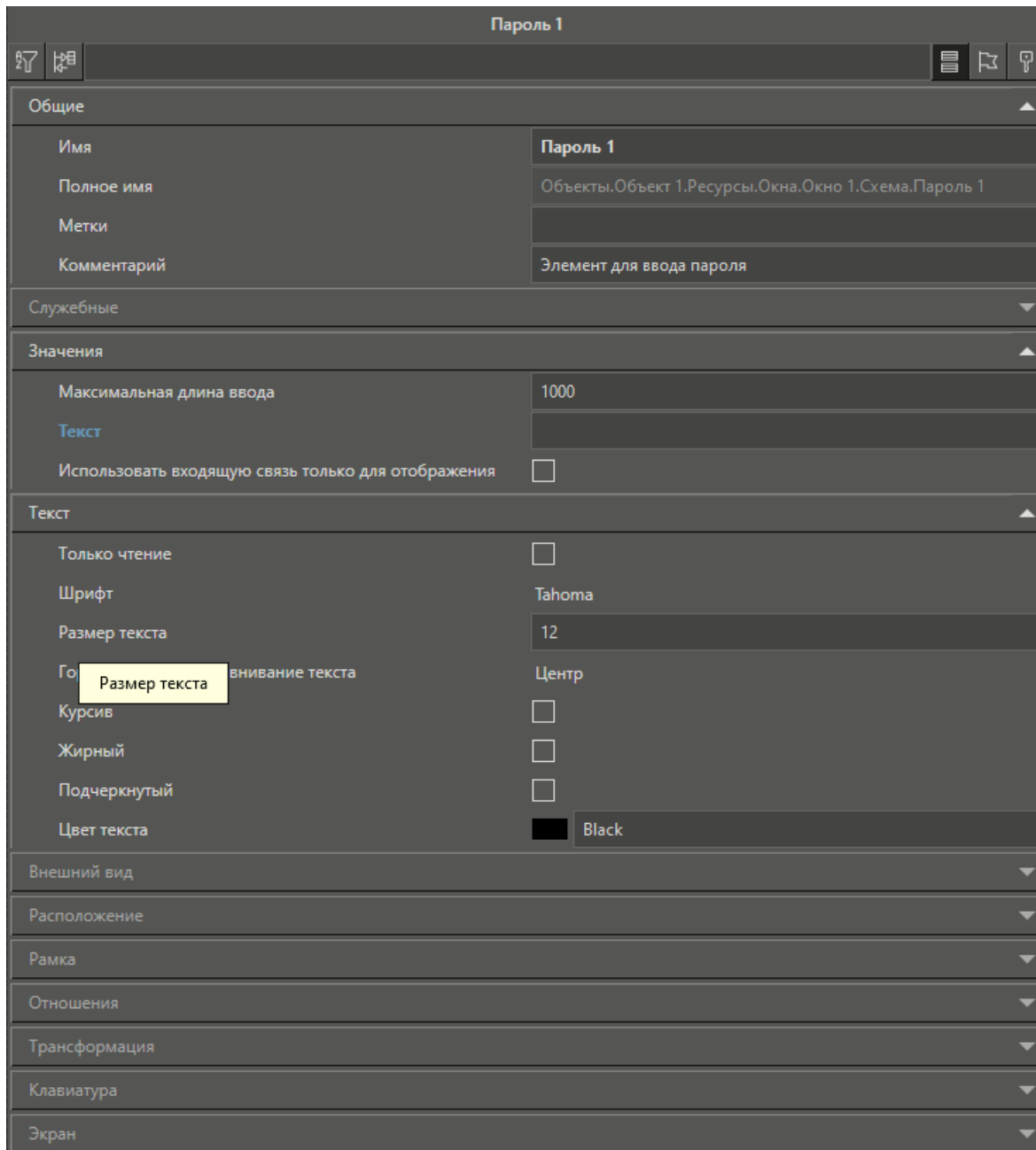
В палитре элемент имеет вид:



Вид элемента после добавления в рабочую область редактора НМІ:



Вид панели свойств элемента:



Описание основных свойств графического элемента Пароль:

Название	Описание
Категория Значения	
Максимальная длина ввода	Это свойство имеет тип INT и задает максимальное число символов во вводимой строке.

Название	Описание
Маска ввода	Определяет, какие символы допустимо вводить в данном поле. Подробное описание свойства смотрите в соответствующем разделе.
Текст	Данное свойство должно быть связано с каким-либо параметром проекта. Если элемент был добавлен в окно из палитры редактора НМІ, то связи необходимо установить вручную. Если параметр был перетащен правой кнопкой мыши в окно, и в контекстном меню был выбран пункт Пароль, то данное поле заполнится автоматически. После установки связи данное свойство будет содержать дополнительные строки, в которых будет находиться информация о связях.
Использовать входящую связь только для отображения	Определяет будет ли поступившее на вход свойства значение передано на выход. Если флаг установлен, то если изменяется значение по входной связи, то элемент отображается в соответствии с ним, но по выходной связи оно не идет. А если пользователь сам изменит текст, то значение передается по выходной связи, даже если значение совпадает с тем, что отправлялось на сервер в прошлый раз

Описание других свойств элемента смотрите в разделе НМІ.Свойства

Внешний вид элемента не отличается в НМІ v1 и в НМІ v2, и порядок работы с элементом одинаков.

8.3.1.4. ИНКРЕМЕНТ

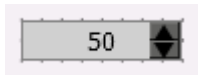
Данный элемент предназначен для установки в клиенте визуализации числовых значений параметров простых типов при помощи реальной или виртуальной клавиатур, а также при помощи нажатия левой кнопкой мыши на стрелки вверх и вниз или на символы + и -. Чаще всего элементы Инкремент в окне создаются путем перетаскивания аналогового параметра в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.

В палитре редактора НМІ элемент Инкремент находится в категории Диалог.

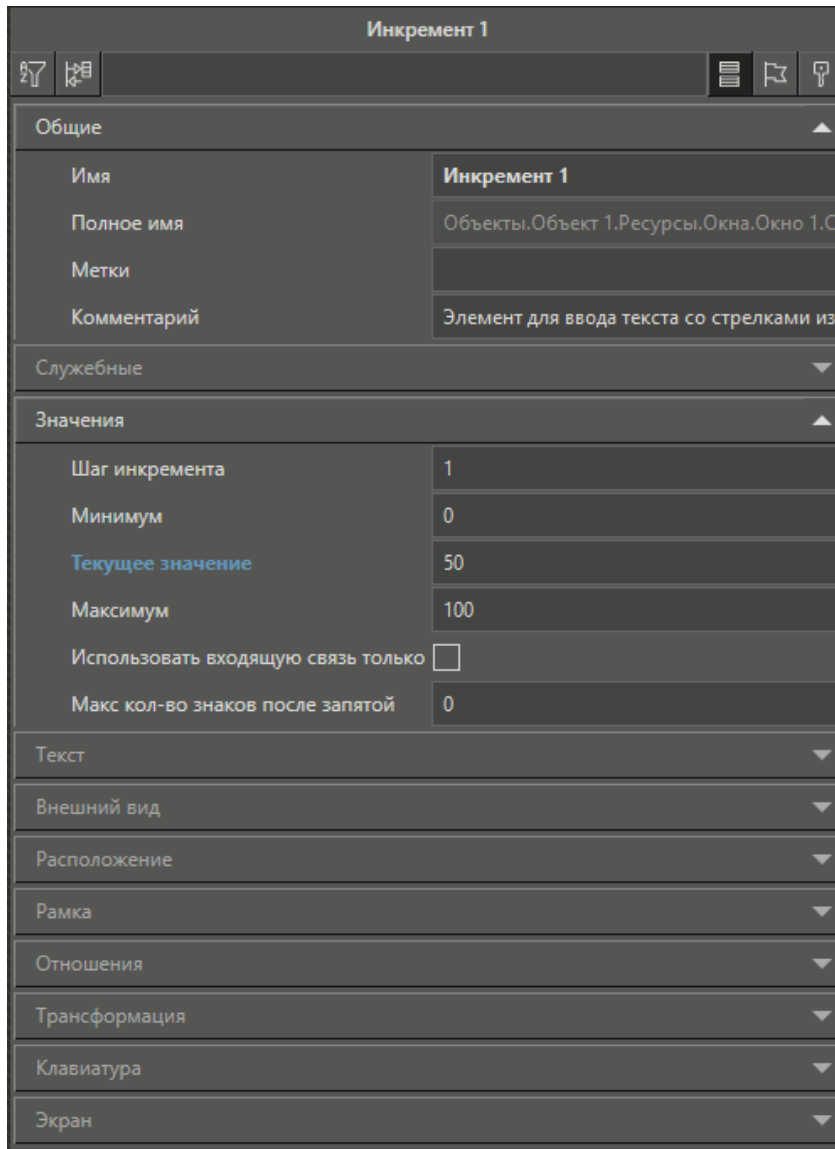
В палитре элемент имеет вид:



Вид элемента после добавления в рабочую область редактора HMI:



Вид панели свойств элемента:



Описание основных свойств графического элемента Инкремент:

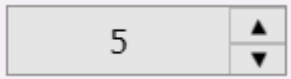
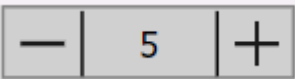
Название	Описание
Категория Внешний вид	
Показать кнопки	Определяет будут ли отображаться кнопки с символами + и - в клиенте визуализации. Если кнопки не отображаются, то

Название	Описание
	ввод значений может осуществляться только при помощи клавиатуры (только для HMI v2).
Категория Значения	
Шаг инкремента	Величина, на которую будет увеличиваться или уменьшаться текущее значение при нажатии на стрелки вверх и вниз или на символы + и -.
Минимум	Задается минимально возможное текущее значение.
Текущее значение	Данное свойство должно быть связано с каким-либо параметром проекта, текущее значение которого предполагается изменять в режиме исполнения. Свойство может иметь входящую и исходящую связи. Если элемент был добавлен в окно из палитры редактора HMI, то связи необходимо установить вручную. Если аналоговый параметр был перетащен в окно правой кнопкой мыши, и в контекстном меню был выбран пункт Инкремент, то данное поле заполнится автоматически. После установки связи данное свойство будет содержать дополнительные строки, в которых будет находиться информация о связях.
Максимум	Задается максимально возможное текущее значение.
Использовать входящую связь только для отображения	Определяет будет ли поступившее на вход свойства значение передано на выход. Если флаг установлен, то если изменяется значение по входной связи, то элемент отображается в соответствии с ним, но по выходной связи оно не идет. А если пользователь сам изменит значение, то значение передается по выходной связи, даже если значение совпадает с тем, что отправлялось на сервер в прошлый раз
Макс кол-во знаков после запятой	Определяет максимальное количество знаков после запятой, которые могут быть введены.
Категория Текст	

Название	Описание
Запрет ручного ввода	Определяет можно ли вводить значения в клиенте визуализации при помощи клавиатуры. Если флаг установлен, то изменение значение возможно только при помощи кнопок с символами + и - (только для HMI v2).

Описание других свойств элемента смотрите в разделе HMI.Свойства

В клиенте визуализации элемент имеет вид :

HMI v1	HMI v2
	

Особенности работы

Рассмотрим пример, когда с элементом Инкремент связан параметр, у которого установлено начальное значение 0, а у Инкремента свойство Минимум имеет значение 0.1. Шаг инкремента равен тоже 0.1 В окне клиента визуализации при старте режима исполнения в Инкременте отобразится значение 0.1, а значение связанного параметра будет 0. После увеличения оператором значения на один шаг, значение инкремента изменится с 0.1 до 0.2 и именно это значение будет присвоено связанному параметру, т.к. инкремент передает значение по связи после первого изменения пользователем.

8.3.1.5. КНОПКА

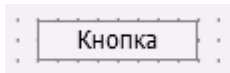
Данный элемент предназначен для выполнения сконфигурированных действий (см. Действия по условию и Действия по событию) либо для изменения значения дискретного параметра, связанного со свойством Нажата Чаще всего элементы Кнопка создаются в окне путем перетаскивания дискретного параметра, объекта или окна в рабочей области редактора HMI правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически. При нажатии левой кнопкой мыши на Кнопку в клиенте визуализации, она перейдет в положение Нажата, и будет находиться в этом положении до тех пор, пока левая кнопка мыши не будет отпущена.

В палитре редактора HMI элемент Кнопка находится в категории Диалог.

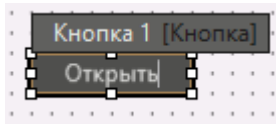
Вид элемента в палитре:



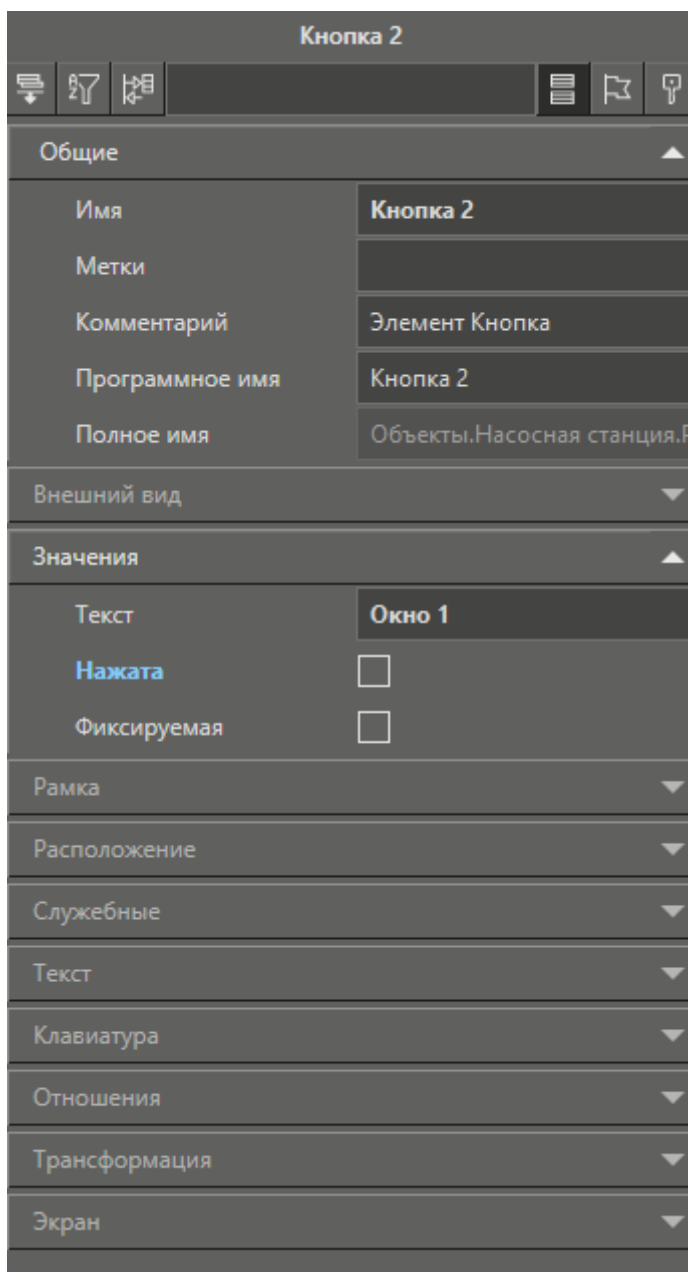
Вид элемента после добавления в рабочую область редактора HMI:



Если дважды кликнуть по элементу, то появится курсор и можно ввести текст, который будет отображаться в клиенте визуализации, т.е. таким образом можно изменить значение свойства элемента Текст. Также режим редактирования включается после нажатием на кнопку клавиатуры F2. Вид элемента во время ввода текста:



Вид панели свойств элемента:



Описание основных свойств графического элемента Кнопка:

Название	Описание
Категория Значения	
Текст	Задается текст надписи на кнопке. Если дискретный параметр был перетащен в окно правой кнопкой мыши, и в контекстном меню был выбран пункт Кнопка, то данное поле заполнится автоматически - текст будет совпадать с именем связанного параметра.
Нажата	Основное свойство кнопки, неразрывно связанное с её положением. Может иметь входящую и исходящую связи. Если элемент был добавлен в окно из палитры редактора НМІ, то связи необходимо установить вручную. Если дискретный параметр был перетащен в окно правой кнопкой мыши, и в контекстном меню был выбран пункт Кнопка, то данное поле заполнится автоматически. После установки связи данное свойство будет содержать дополнительные строки, в которых будет находиться информация о связях.
Фиксируемая	У данного элемента это свойство доступно только при отображении всех свойств в панели (кнопка  отжата). По умолчанию, у элемента Кнопка это свойство имеет значение False. Если перевести значение свойства в TRUE, то элемент кнопка по функционалу будет совпадать с элементом кнопка с фиксацией.

Описание других свойств элемента смотрите в разделе НМІ.Свойства

Внешний вид элемента не отличается в НМІ v1 и в НМІ v2, и порядок работы с элементом одинаков.

8.3.1.6. КНОПКА С ФИКСАЦИЕЙ

Как правило, данный элемент используется для установки значения дискретного параметра, связанного со свойством Нажата: если кнопка нажата, то параметр принимает значение TRUE, если кнопка отжата, то False. Чаще всего элементы Кнопка с фиксацией в окне создаются путем перетаскивания дискретного параметра в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически. При каждом нажатии левой кнопкой мыши на Кнопку с фиксацией в клиенте визуализации, она меняет свое положение.

В палитре редактора HMI элемент Кнопка с фиксацией находится в категории Диалог.

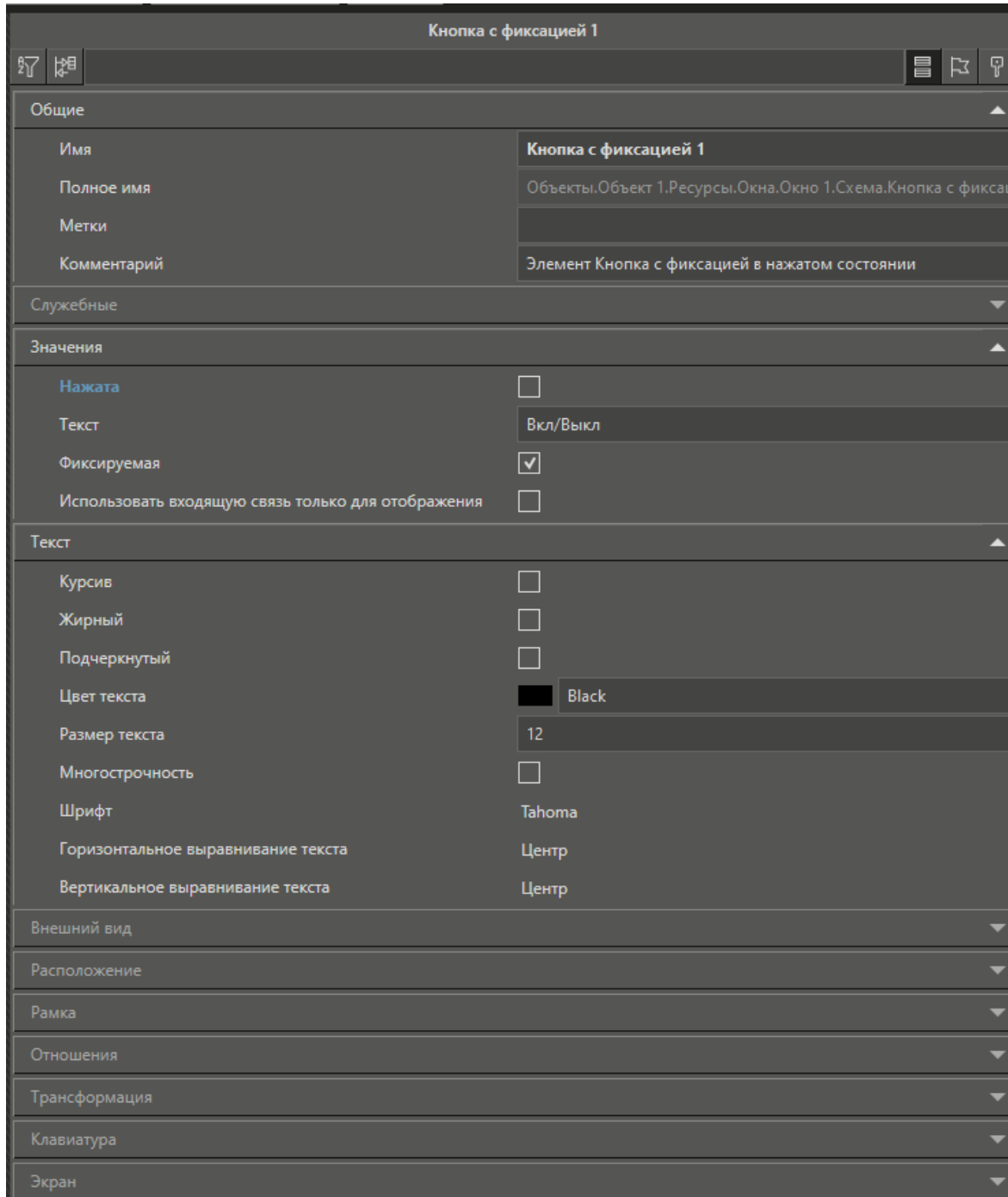
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:




Вид панели свойств элемента:



Описание основных свойств графического элемента Кнопка с фиксацией:

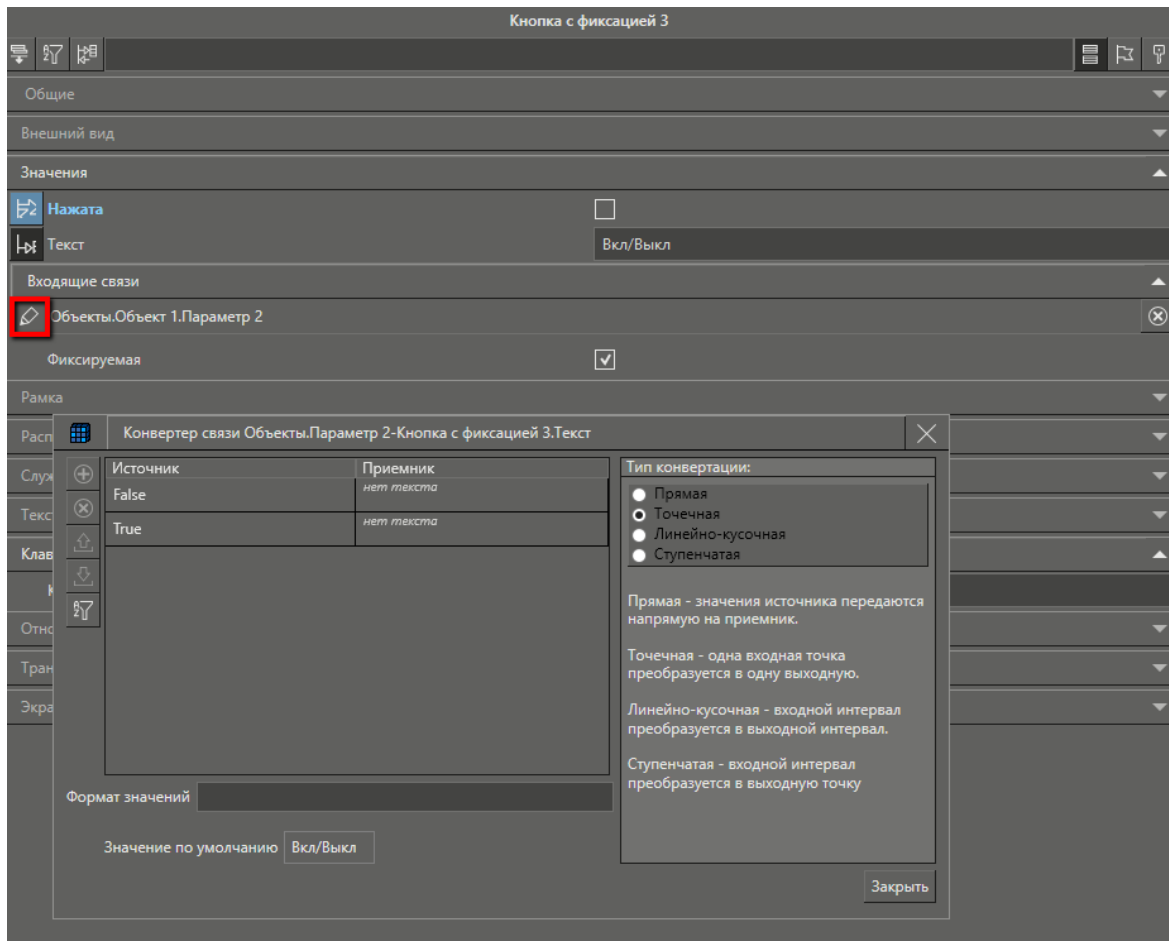
Название	Описание
Категория Значения	
Текст	Задается текст надписи на кнопке с фиксацией. Если дискретный параметр был перетасчен в окно правой кнопкой мыши, и в

Название	Описание
	<p>контекстном меню был выбран пункт Кнопка с фиксацией, то данное поле заполнится автоматически - текст будет совпадать с именем связанного параметра. Рекомендации по настройке читайте ниже в этом разделе.</p>
Нажата	<p>Основное свойство кнопки с фиксацией, неразрывно связанное с её положением. Может иметь входящую и исходящую связи. Если элемент был добавлен в окно из палитры редактора НМІ, то связи необходимо установить вручную. Если дискретный параметр был перетащен в окно правой кнопкой мыши, и в контекстном меню был выбран пункт Кнопка с фиксацией, то данное поле заполнится автоматически. После установки связи данное свойство будет содержать дополнительные строки, в которых будет находиться информация о связях.</p>
Фиксируемая	<p>У данного элемента свойство доступно только при отображении всех свойств в панели (кнопка  отжата). По умолчанию, у элемента кнопка с фиксацией это свойство находится в состоянии TRUE. Если перевести значение свойства в состояние False, то элемент кнопка с фиксацией по функционалу будет совпадать с элементом кнопка.</p>
Использовать входящую связь только для отображения	<p>Определяет будет ли поступившее на вход свойства значение передано на выход. Если флаг установлен, то если изменяется значение по входной связи, то элемент отображается в соответствии с ним, но по выходной связи оно не идет. А если пользователь сам изменит значение, то значение передается по выходной связи, даже если значение совпадает с тем, что отправлялось на сервер в прошлый раз</p>

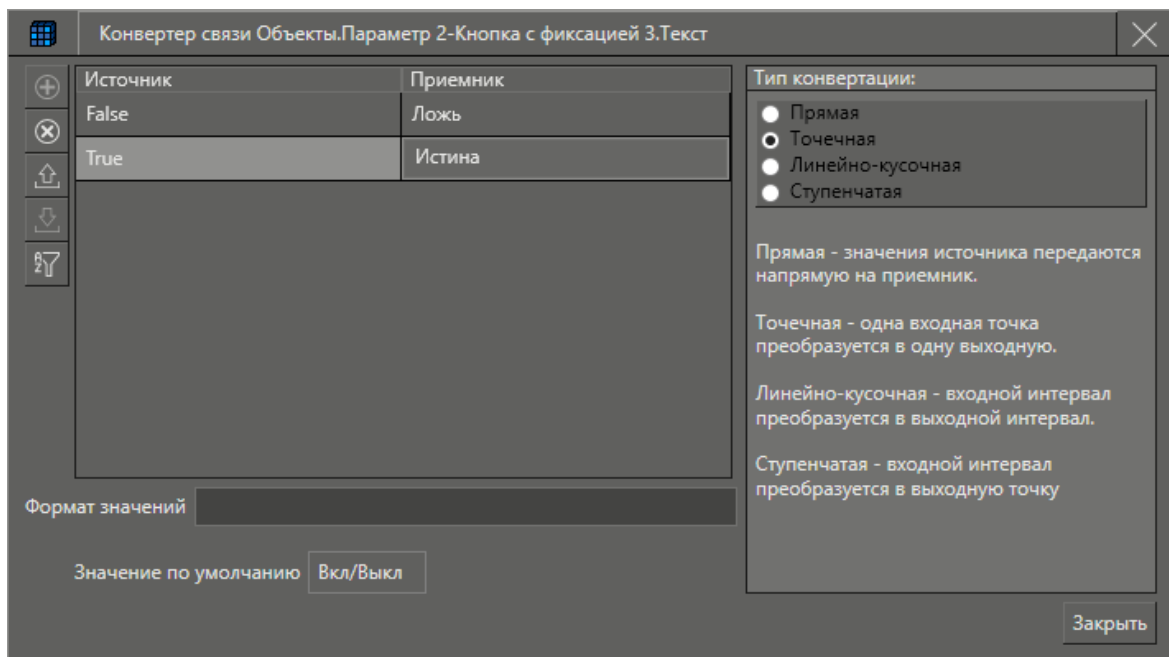
Описание других свойств элемента смотрите в разделе НМІ.Свойства

Настройка свойства Текст

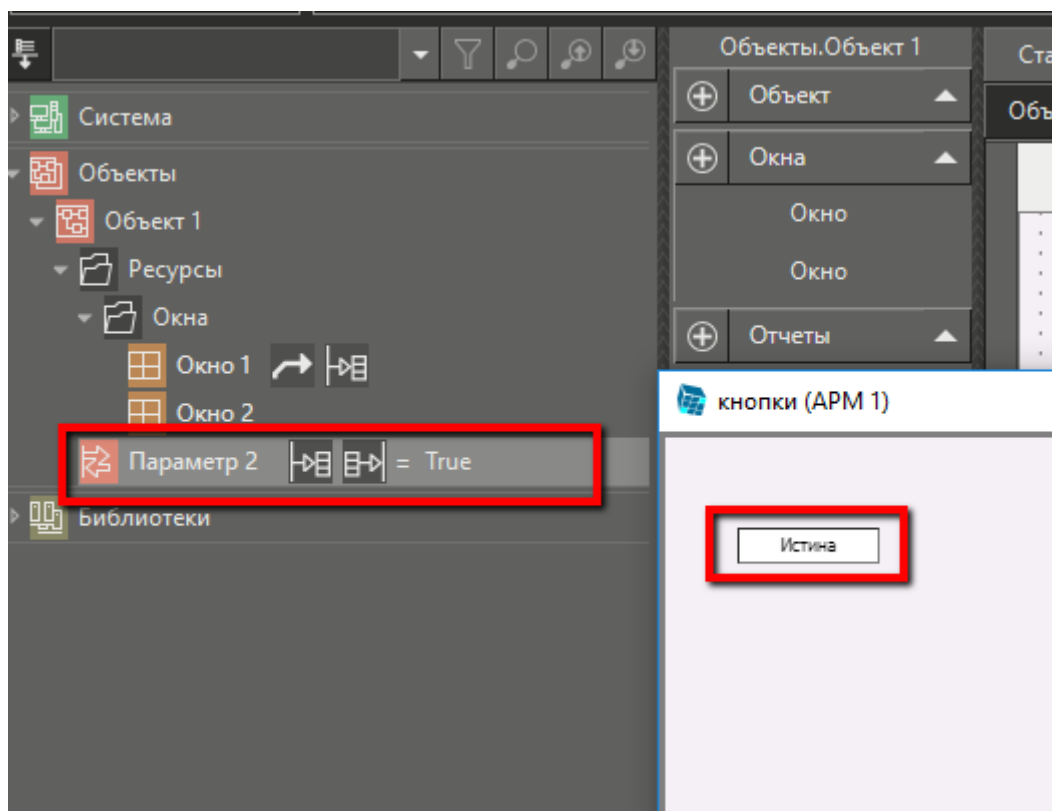
Для того чтобы текст на кнопке отображал состояние параметра, необходимо перетащить параметр на свойство текст и открыть конвертер образовавшейся связи (для точечной передаточной характеристики этой связи поля значений источника будут уже заполнены как TRUE и False):



Затем следует заполнить соответствующие поля приемника и нажать кнопку Закреть:



В результате получим в клиенте визуализации:



Внешний вид элемента не отличается в НМІ v1 и в НМІ v2, и порядок работы с элементом одинаков.

8.3.1.7. РАДИОКНОПКА

Данный элемент предназначен для установки в клиенте визуализации значения дискретного параметра, связанного со свойством отмечен. Если радиокнопка отмечена, то параметр имеет значение TRUE; если отметка снята, то False. Для того чтобы отметить радиокнопку, необходимо в клиенте визуализации нажать на неё левой кнопкой мыши, а чтобы снять отметку – на какую-либо другую радиокнопку, находящуюся в этом же окне или в панели. Таким образом, данный элемент можно использовать в ситуации, когда имеется набор дискретных параметров, а значение TRUE может иметь только один из них.

Чаще всего элементы Радиокнопка в окне создаются путем перетаскивания дискретного параметра в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.


В палитре редактора НМІ элемент Радиокнопка находится в категории Диалог.

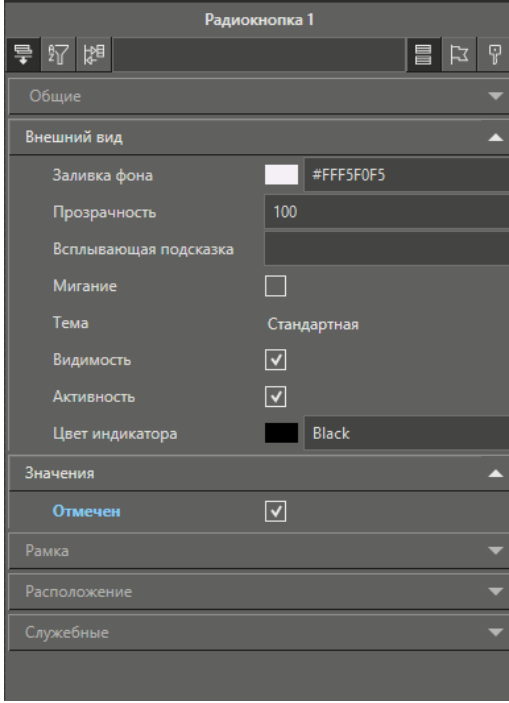
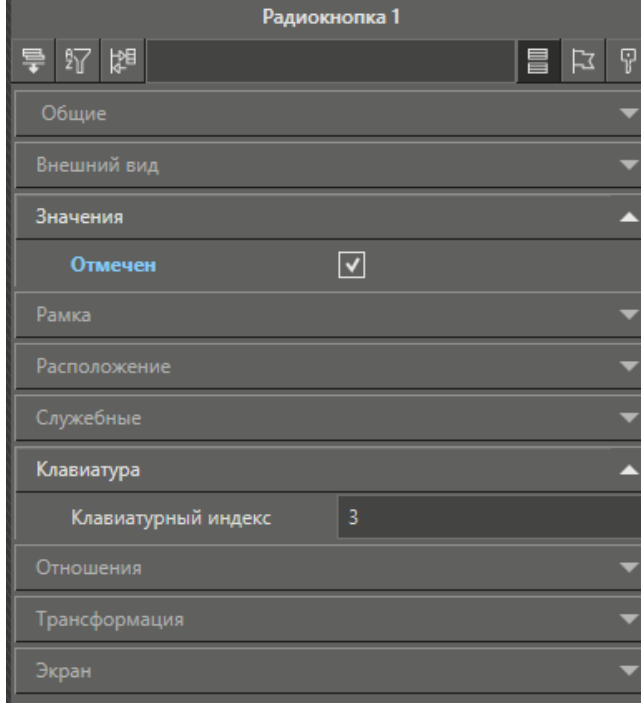
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора НМІ:



Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
	

Описание основных свойств графического элемента Радиокнопка:

Название	Описание
<p>Категория Значения</p>	
<p>Отмечен</p>	<p>Основное свойство радиокнопки, неразрывно связанное с наличием или отсутствием у неё отметки. Может иметь входящую и исходящую связи. Если элемент был добавлен в окно из палитры редактора HMI, то связи необходимо установить вручную. Если дискретный параметр был перетасчен в окно правой кнопкой мыши, и в контекстном меню был выбран пункт Радиокнопка, то данное поле заполнится автоматически. После установки связи данное свойство будет содержать дополнительные строки, в которых будет находиться информация о связях.</p>

Название	Описание
	Важно! Допускается чтобы только у одной радиокнопки в окне или панели это свойство имело значение TRUE по умолчанию.

Описание других свойств элемента смотрите в разделе НМІ.Свойства

Внешний вид элемента не отличается в НМІ v1 и в НМІ v2, и порядок работы с элементом одинаков.

8.3.1.8. ФЛАГ

Данный элемент предназначен для установки значения дискретного параметра, связанного со свойством Отмечен. Если флаг отмечен, то параметр имеет значение TRUE, а если флаг снят, то False. Каждое нажатии левой кнопкой мыши на Флаг в клиенте визуализации, либо отмечает флаг, либо снимает с него отметку, и свойство Отмечен меняет свое значение. Чаще всего элементы Флаг в окне создаются путем перетаскивания дискретного параметра в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.

В палитре редактора НМІ элемент Флаг находится в категории Диалог.

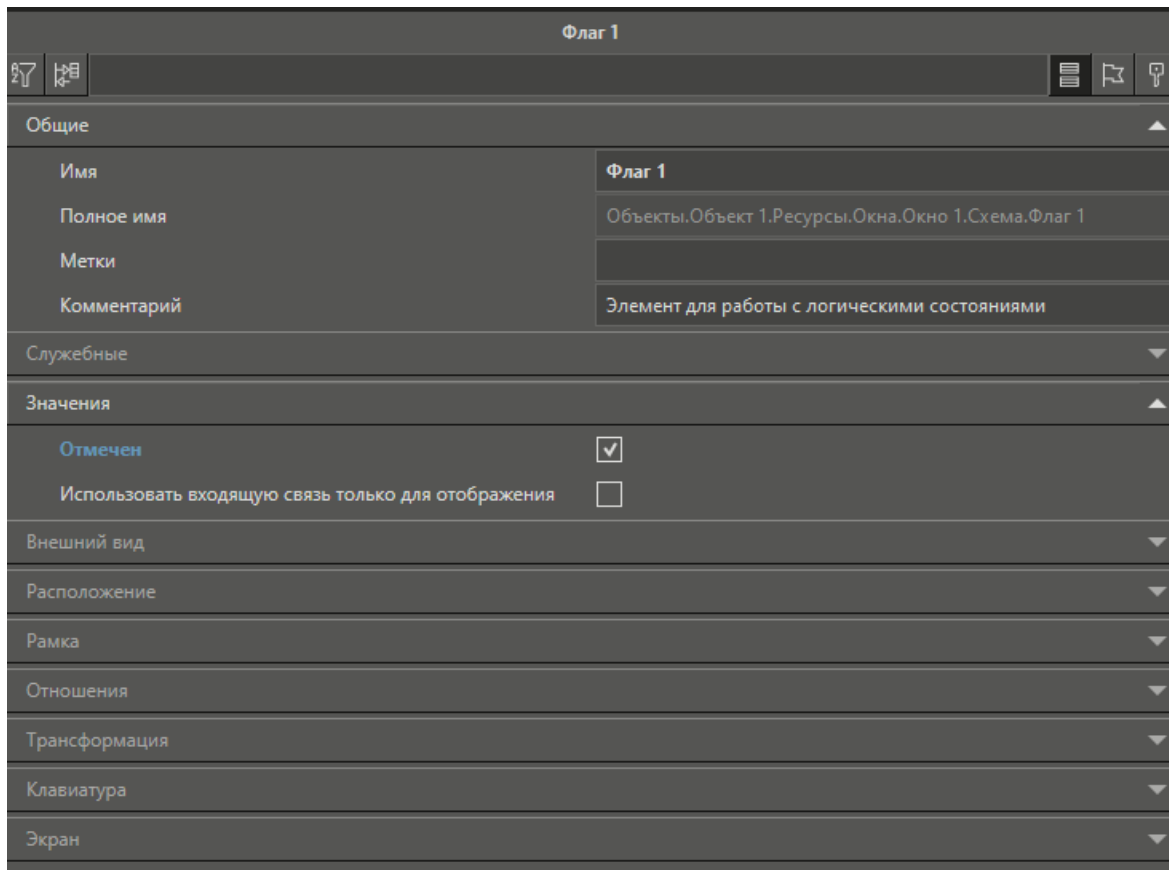
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора НМІ:



Вид панели свойств элемента:



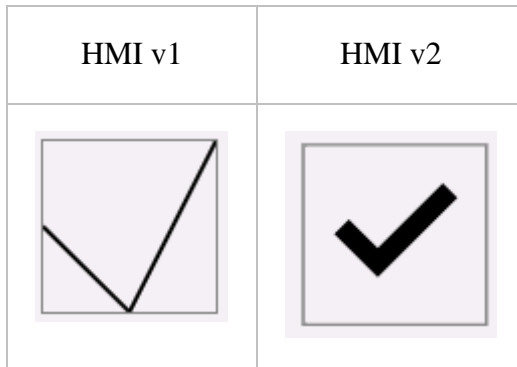
Описание основных свойств графического элемента флаг:

Название	Описание
Категория Значения	
Отмечен	Основное свойство флага, неразрывно связанное с наличием или отсутствием у него отметки. Может иметь входящую и исходящую связи. Если элемент был добавлен в окно из палитры редактора НМІ, то связи необходимо установить вручную. Если дискретный параметр был перетащен в окно правой кнопкой мыши, и в контекстном меню был выбран пункт Флаг, то данное поле заполнится автоматически. После установки связи данное свойство будет содержать дополнительные строки, в которых будет находиться информация о связях.
Использовать входящую связь только для отображения	Определяет будет ли поступившее на вход свойства значение передано на выход. Если флаг установлен, то если изменяется значение по входной связи, то элемент отображается в соответствии с ним, но по выходной связи оно не идет. А если пользователь сам изменит значение, то значение передается по выходной связи, даже

Название	Описание
	если значение совпадает с тем, что отправлялось на сервер в прошлый раз

Описание других свойств элемента смотрите в разделе НМІ.Свойства

В клиенте визуализации элемент имеет вид :



8.3.1.9. ВЫПАДАЮЩИЙ СПИСОК

Данный элемент позволяет выбрать из набора строк одну. Список строк может быть определен как в среде разработки, так и изменен в процессе работы среды исполнения. При перетаскивании параметра типа перечисление в окно правой кнопкой мыши, одним из возможных вариантов отображения будет выпадающий список.

В палитре редактора НМІ элемент Выпадающий список находится в категории Диалог.

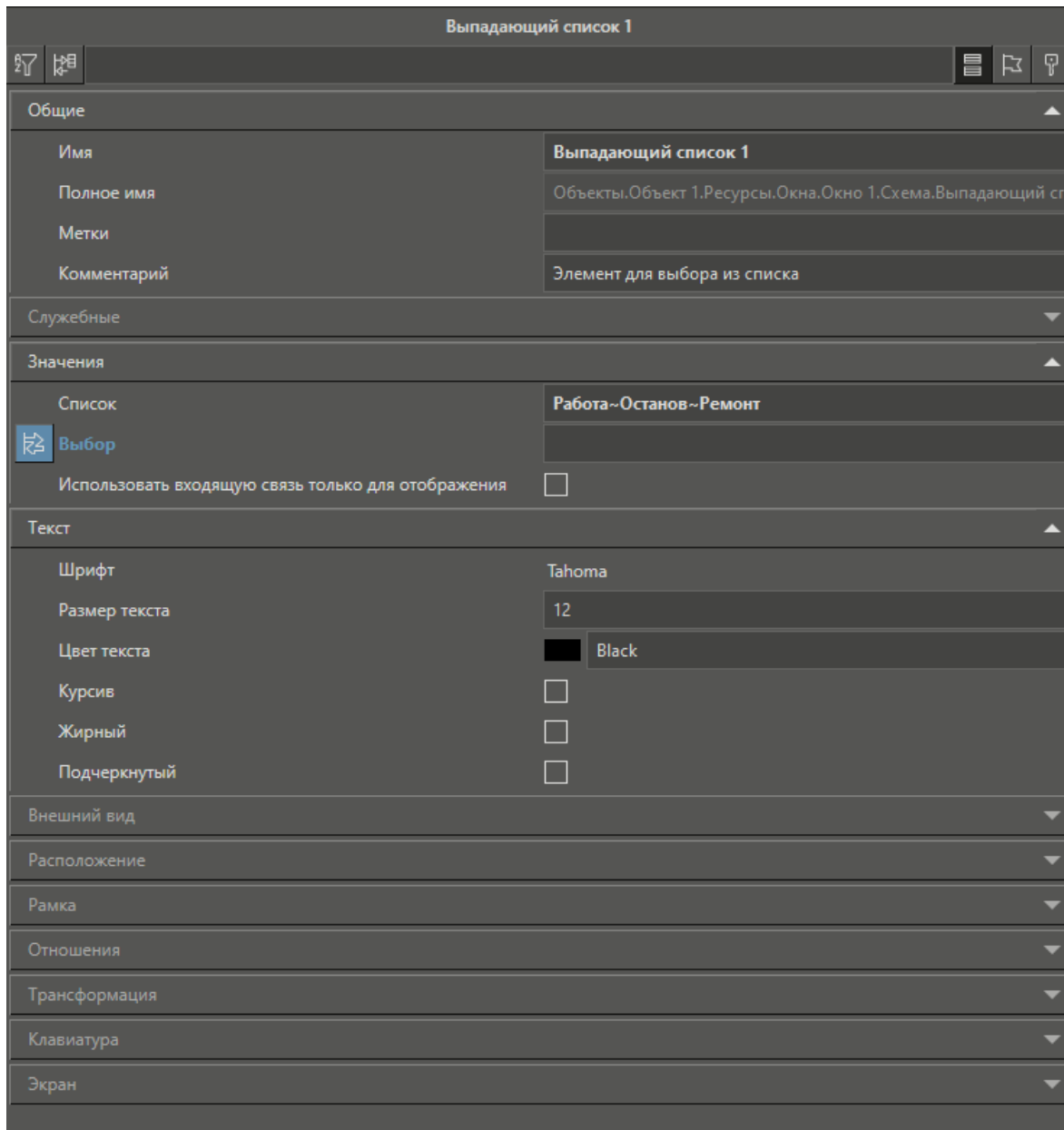
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора НМІ:



Вид панели свойств элемента:



Описание основных свойств графического элемента выпадающий список:

Название	Описание
Категория Значения	Подробное описание свойств данной категории смотрите в разделе Внешний вид справочной системы.
Список	Это свойство имеет тип <code>STRING</code> и определяет список доступных строк в выпадающем списке. Строки в списке разделяются символом ~. При установке входящей связи список строк может быть изменен в результате работы среды

Название	Описание
	исполнения. Если параметр типа перечисление был перетащен в окно правой кнопкой мыши, и в контекстном меню был выбран пункт Выпадающий список, то данное свойство заполнится автоматически.
Выбор	Это свойство имеет тип STRING и определяет, какая строка выбрана в списке. Свойство может иметь входящую и исходящую связи. Если элемент выпадающий список был создан в окне в результате перетаскивания параметра типа перечисление в окно правой кнопкой мыши, то связи будут установлены автоматически.
Использовать входящую связь только для отображения	Определяет будет ли поступившее на вход свойства значение передано на выход. Если флаг установлен, то если изменяется значение по входной связи, то элемент отображается в соответствии с ним, но по выходной связи оно не идет. А если пользователь сам изменит выбор, то значение передается по выходной связи, даже если значение совпадает с тем, что отправлялось на сервер в прошлый раз

Описание других свойств элемента смотрите в разделе НМІ.Свойства

Внешний вид элемента не отличается в НМІ v1 и в НМІ v2, и порядок работы с элементом одинаков.

8.3.1.10. ЧАСЫ

Данный элемент отображает текущее время и/или дату устройства, на котором запускается клиент визуализации. Для его работы не требуется связь с параметрами дерева. Если одновременно запущено несколько клиентов на разных устройствах, то время на них может отличаться. В случае, если необходимо, чтобы на всех клиентах отображалось время, отсчитываемое на устройстве с работающей средой исполнения, следует соответствующий параметр, обрабатываемый в задаче узла, перетащить в окно правой кнопкой мыши, и выбрать способ отображения ДатаВремя.


В палитре редактора НМІ элемент Часы находится в категории Диалог.

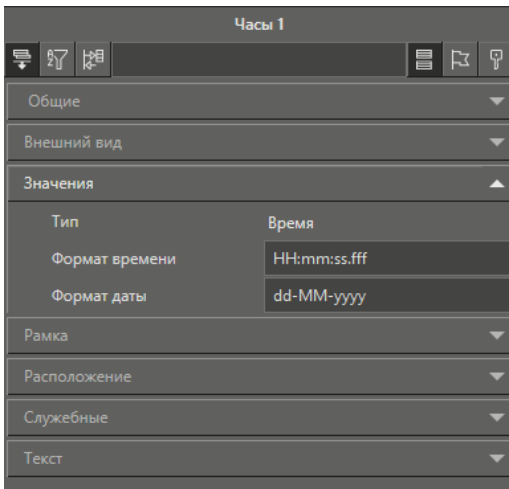
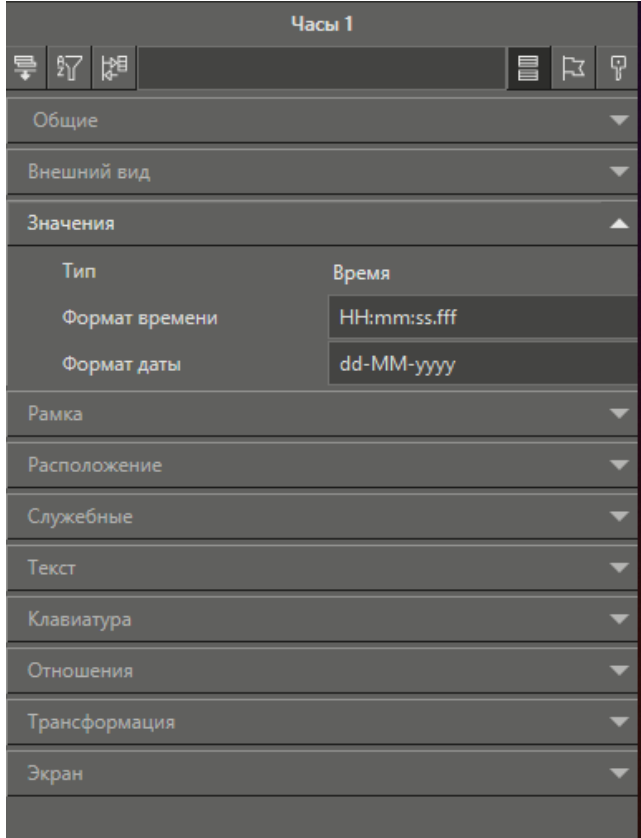
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:



Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
	

Описание основных свойств графического элемента часы:

Название	Описание
Категория Значения	
Тип	Задает, что необходимо отображать: <ul style="list-style-type: none"> • дата; • дата и время;

Название	Описание
	<ul style="list-style-type: none"> • время и дата; • время.
Формат времени	Определяет формат времени. По умолчанию свойство принимает значение HH:mm:ss.fff, где: HH – это часы, mm – минуты, ss – секунды, fff – миллисекунды. Например, если формат времени будет иметь вид HH:mm, то в поле часов будет отображаться только часы и минуты.
Формат даты	Определяет формат даты. По умолчанию свойство принимает значение dd-ММ-уууу, где: dd – число, ММ – месяц, уууу – год. Например, если формат даты будет иметь вид dd-ММ, то в поле часов будет отображаться только число и месяц.

Описание других свойств элемента смотрите в разделе НМІ.Свойства

Внешний вид элемента не отличается в НМІ v1 и в НМІ v2, и порядок работы с элементом одинаков.

8.3.1.11. ДАТАВРЕМЯ

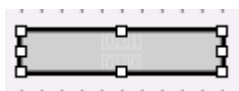
Данный элемент предназначен для отображения и/или установки значений параметров типа DT. Чаще всего элементы ДатаВремя в окне создаются путем перетаскивания параметра типа DT в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.


В палитре редактора НМІ элемент ДатаВремя находится в категории Диалог.

Вид элемента в палитре:

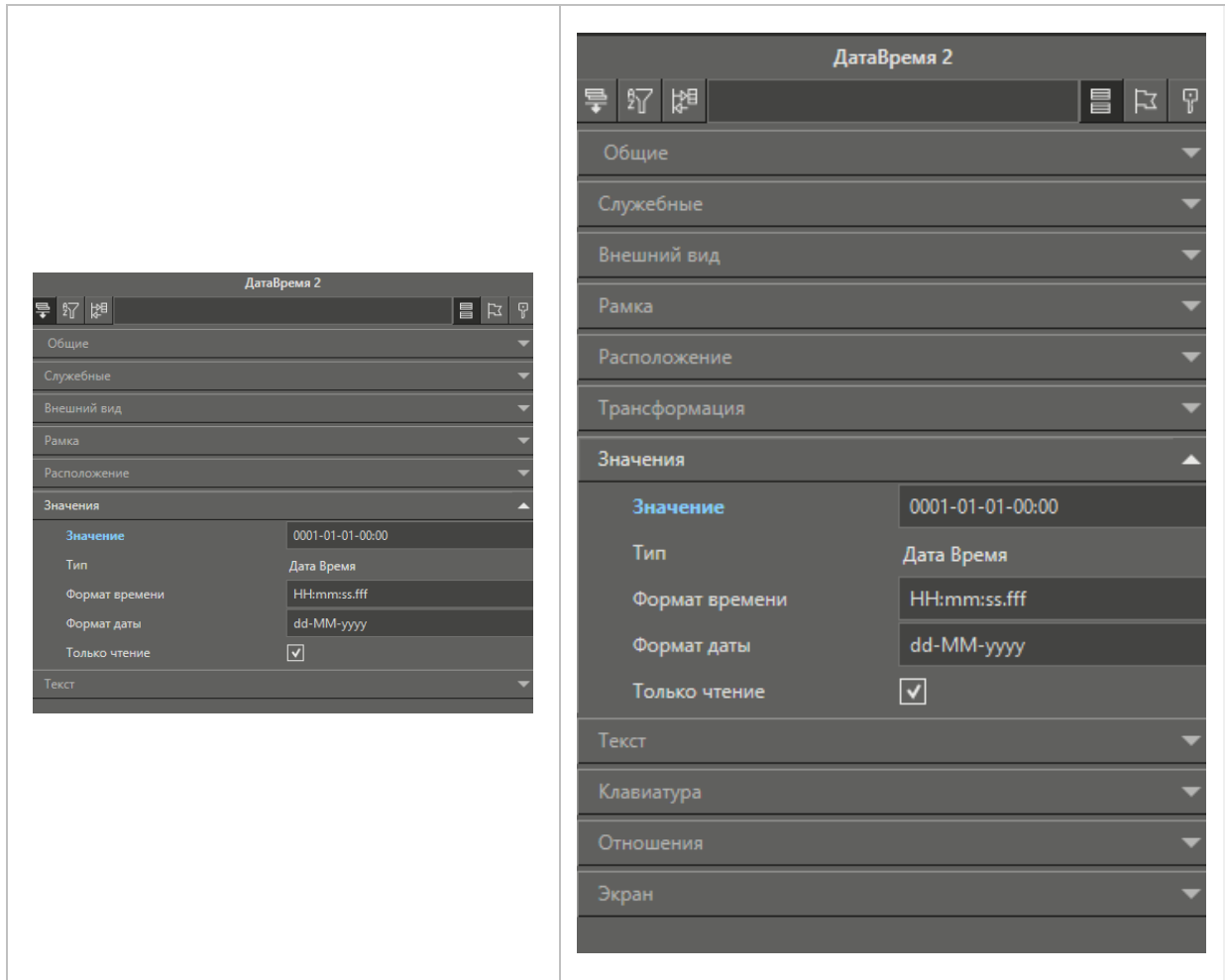


Вид элемента после добавления в рабочую область редактора НМІ:



Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
---------------	--------------



Описание основных свойств графического элемента ДатаВремя:

Название	Описание
Категория Значения	
Значение	Основное свойство элемента. Это свойство должно быть связано с параметром, значение которого необходимо устанавливать или значение которого требуется отображать в клиенте визуализации. Соответственно, свойство может иметь входящую и исходящую связи. Если элемент был добавлен в окно из палитры редактора НМІ , то связи необходимо установить вручную. Если параметр типа DT был перетащен в окно правой кнопкой мыши, и в контекстном меню был выбран пункт ДатаВремя, то данное поле заполнится автоматически. После установки связи данное свойство будет содержать дополнительные строки, в которых будет находиться информация о связях.

Название	Описание
Категория Значения	
Тип	<p>Задаёт, что именно и в каком порядке необходимо отображать. Выпадающий список содержит следующие варианты:</p> <ul style="list-style-type: none"> • дата; • дата и время; • время и дата; • время.
Формат времени	<p>Определяет формат времени. По умолчанию свойство принимает значение HH:mm:ss.fff, где: HH – это часы, mm – минуты, ss – секунды, fff – миллисекунды. Например, если формат времени будет иметь вид HH:mm, то в поле часов будут отображаться только часы и минуты.</p>
Формат даты	<p>Определяет формат даты. По умолчанию свойство принимает значение dd-ММ-уууу, где: dd – число, ММ – месяц, уууу – год. Например, если формат даты будет иметь вид dd-ММ, то в поле часов будут отображаться только число и месяц.</p>
Только чтение	<p>Если свойство имеет значение TRUE, то ввод значения параметра в клиенте визуализации запрещен. Если свойство принимает значение False, то при нажатии левой кнопки мыши на элемент в клиенте визуализации откроется окно ввода значения параметра.</p>

Описание других свойств элемента смотрите в разделе НМІ.Свойства

Установка даты/времени в клиенте визуализации

Тип транслятора НМІ v1

В НМІ v1 элемент имеет вид:

31-01-2014 23:00:00.000

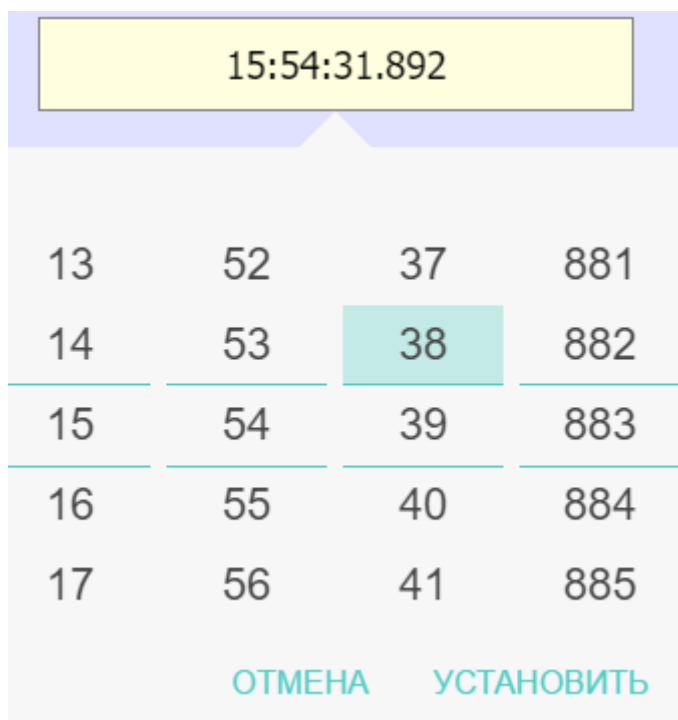
Для того чтобы отобразить инструменты установки даты/времени, необходимо нажать левой кнопкой мыши на элемент (набор инструментов зависит от свойства тип):

31-01-2017 03:04:08.011						
29	11	2015	01	02	06	009
30	12	2016	02	03	07	010
31	01	2017	03	04	08	011
01	02	2018	04	05	09	012
02	03	2019	05	06	10	013
						ОТМЕНА УСТАНОВИТЬ

Для изменения отдельных элементов даты/времени с помощью показанных выше инструментов могут быть использованы различные способы:

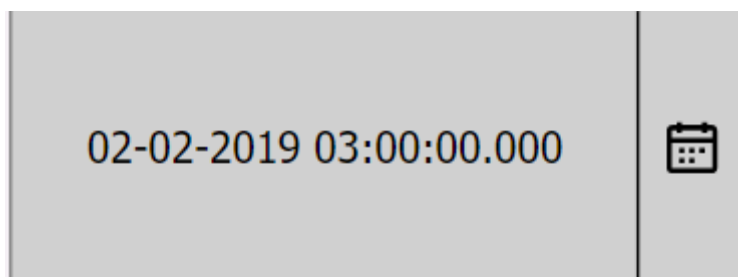
- Навести мышь на отдельный элемент даты или времени и нажать левой кнопкой мыши на нужное значение;
- Навести мышь на отдельный элемент даты или времени и вращением колеса мыши перемещать его до нужного значения;
- Навести мышь на отдельный элемент даты или времени и перемещать его левой кнопкой мыши вверх или вниз до нужного значения.

Для применения выполненных изменений следует нажать кнопку УСТАНОВИТЬ. По команде ОТМЕНА инструменты установки даты/времени скрываются, при этом выполненные изменения игнорируются. Для того чтобы скрыть инструменты установки даты/времени можно также нажать левой кнопкой мыши на произвольное место мнемосхемы:



Тип транслятора НМІ v2

В НМІ v2 элемент имеет вид::



Для изменения даты/времени можно воспользоваться одним из следующих способов:

- Нажать левой кнопкой мыши перед тем символом, значение которого необходимо изменить. При этом вокруг отображаемого времени появится синяя рамка и курсор. Затем, используя клавиатуру, можно ввести необходимое значение. При вводе можно перемещать курсор стрелками влево/вправо, использовать клавиши DEL, Backspace. Значение будет применено после нажатия на клавишу Enter, либо нажатия любой кнопки мыши за границей элемента.
- Нажать левой кнопкой мыши на правую область элемента. При этом откроется окно, в котором можно изменять отдельные части даты/времени, используя соответствующие стрелки, либо покрутив колесо мыши в нужном поле:

День	Месяц	Год
▲	▲	▲
04	/ 02	/ 2019
▼	▼	▼

Часы	Мин.	Сек.	Милисек.
▲	▲	▲	▲
04	/ 08	/ 30	/ 780
▼	▼	▼	▼

Для применения выполненных изменений даты/времени следует нажать кнопку **ВЫБРАТЬ**. По команде **ОТМЕНА** инструменты установки даты/времени скрываются, при этом выполненные изменения игнорируются. Для того чтобы скрыть инструменты установки даты/времени можно также нажать левой кнопкой мыши на произвольное место мнемосхемы.

8.3.1.12. ВРЕМЯ

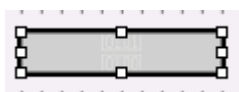
Данный элемент предназначен для отображения и/или установки значений параметров типа TOD. Чаще всего элементы Время в окне создаются путем перетаскивания параметра типа TOD в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.


В палитре редактора НМІ элемент Время находится в категории Диалог.

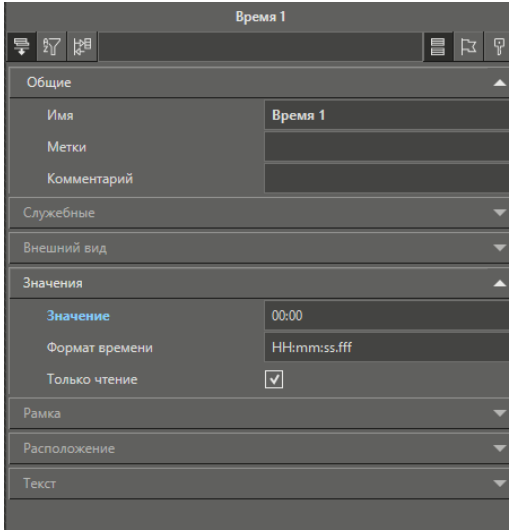
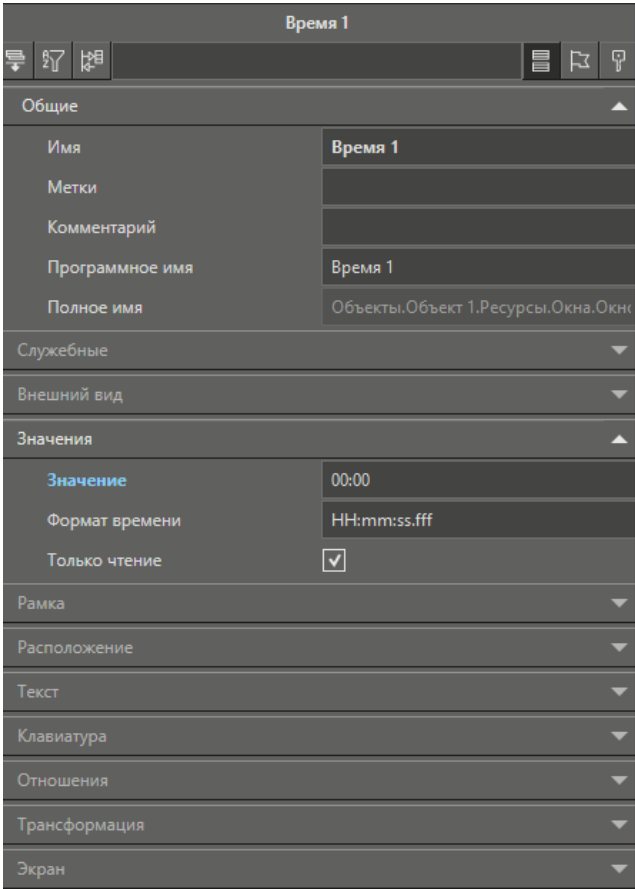
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора НМІ:



Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
	

Описание основных свойств графического элемента **Время**:

Название	Описание
<p>Категория Значения</p>	
<p>Значение</p>	<p>Основное свойство элемента. Это свойство должно быть связано с параметром, значение которого необходимо устанавливать или значение которого требуется отображать в клиенте визуализации. Соответственно, свойство может иметь входящую и исходящую связи. Если элемент был добавлен в окно из палитры редактора НМІ , то связи необходимо установить вручную. Если параметр типа TOD был перетащен в окно правой кнопкой мыши, и в контекстном меню был выбран пункт Время, то данное поле заполнится автоматически. После установки связи данное свойство будет содержать дополнительные строки, в которых будет находиться информация о связях.</p>

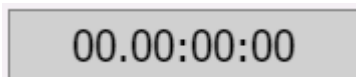
Название	Описание
Категория Значения	
Формат времени	Определяет формат времени. По умолчанию значение свойства равно HH:mm:ss.fff, где: HH – это часы, mm – минуты, ss – секунды, fff – миллисекунды. Например, если формат времени будет иметь вид HH:mm, то в поле часов будут отображаться только часы и минуты.
Только чтение	Если свойство имеет значение TRUE, то ввод значения параметра в клиенте визуализации запрещен. Если свойство принимает значение False, то при нажатии левой кнопки мыши на элемент в клиенте визуализации откроется окно ввода значения параметра.

Описание других свойств элемента смотрите в разделе НМІ.Свойства

Установка времени в клиенте визуализации

Тип транслятора НМІ v1

В НМІ v1 элемент имеет вид:



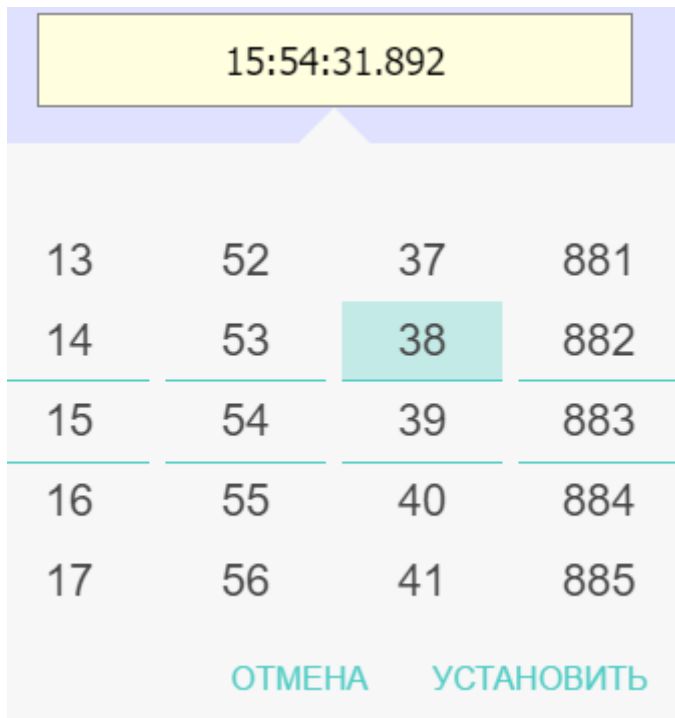
Для того чтобы отобразить инструменты задания времени, необходимо нажать левой кнопкой мыши на элемент:

15:54:31.892			
13	52	29	890
14	53	30	891
15	54	31	892
16	55	32	893
17	56	33	894
ОТМЕНА		УСТАНОВИТЬ	

Для установки отдельных элементов времени с помощью показанных выше инструментов можно использовать различные методы:

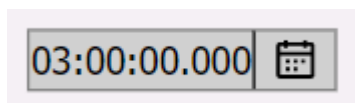
- Навести мышь на отдельный элемент даты или времени и нажать левой кнопкой мыши на нужное значение;
- Навести мышь на отдельный элемент даты или времени и вращением колеса мыши перемещать его до нужного значения;
- Навести мышь на отдельный элемент даты или времени и перемещать его левой кнопкой мыши вверх или вниз до нужного значения.

Для применения выполненных изменений следует нажать кнопку УСТАНОВИТЬ. По команде ОТМЕНА инструменты установки даты/времени скрываются, при этом выполненные изменения игнорируются. Для того чтобы скрыть инструменты установки времени можно также нажать левой кнопкой мыши на произвольное место мнемосхемы:



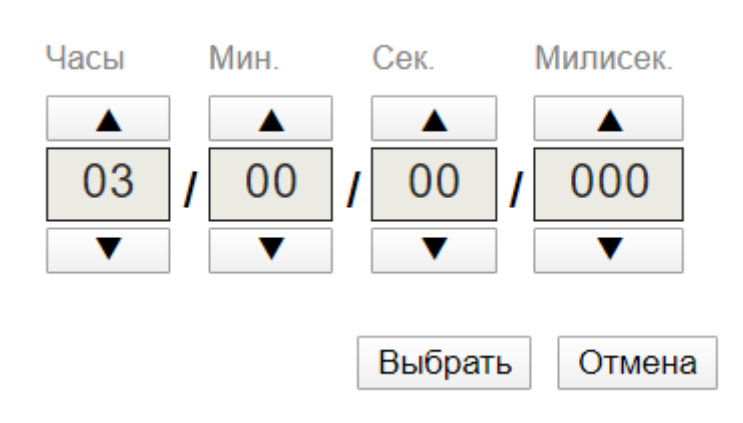
Тип транслятора НМІ v2

В НМІ v2 элемент имеет вид:



Для изменения времени можно воспользоваться одним из следующих способов:

- Нажать левой кнопкой мыши перед тем символом, значение которого необходимо изменить. При этом вокруг отображаемого времени появится синяя рамка и курсор. Затем, используя клавиатуру, можно ввести необходимое значение. При вводе можно перемещать курсор стрелками влево/вправо, использовать клавиши DEL, Backspace. Значение будет применено после нажатия на клавишу Enter, либо нажатия любой кнопки мыши за границей элемента.
- Нажать левой кнопкой мыши на правую область элемента. При этом откроется окно, в котором можно изменять отдельные части времени, используя соответствующие стрелки, либо покрутив колесо мыши в нужном поле:



Для применения выполненных изменений времени следует нажать кнопку ВЫБРАТЬ. По команде ОТМЕНА инструменты установки времени скрываются, при этом выполненные изменения игнорируются. Для того чтобы скрыть инструменты установки времени можно также нажать левой кнопкой мыши на произвольное место мнемосхемы.

8.3.1.13. ИНТЕРВАЛ

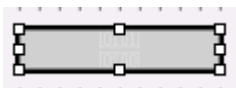
Данный элемент предназначен для отображения и/или установки значений типа TIME. Чаще всего элементы Интервал в окне создаются путем перетаскивания параметра типа TIME в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.


В палитре редактора НМІ элемент Интервал находится в категории Диалог.

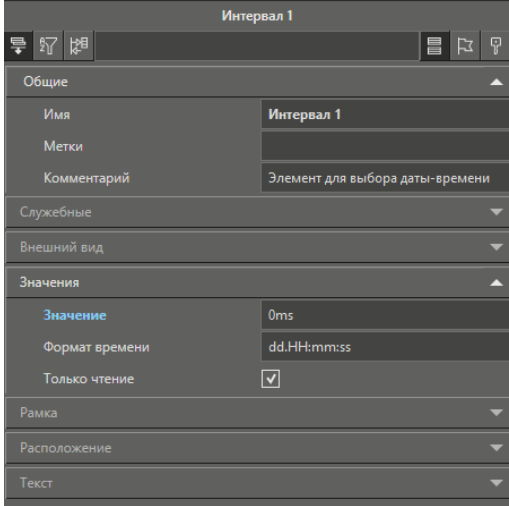
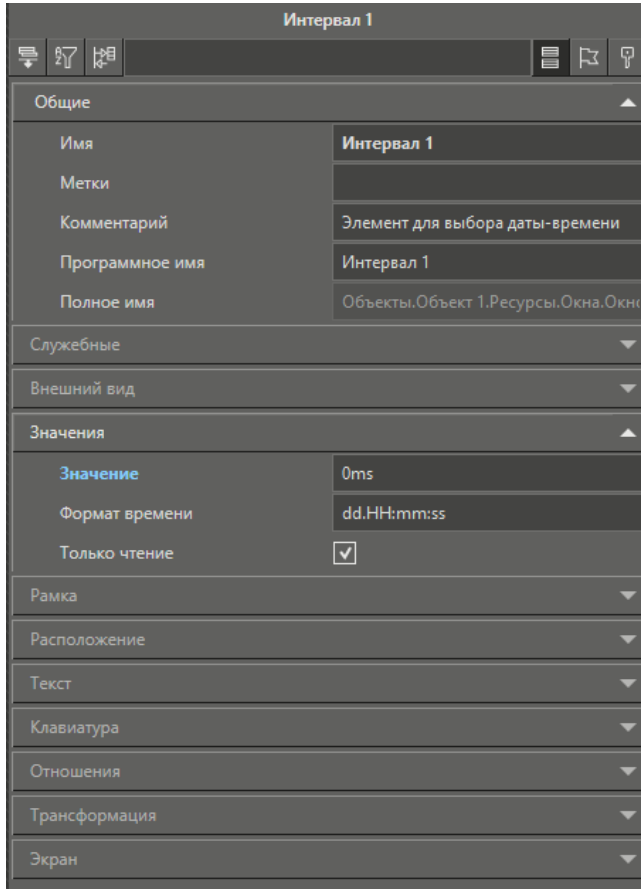
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора НМІ:



Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
	

Описание основных свойств графического элемента Интервал:

Название	Описание
<p>Категория Значения</p>	
<p>Значение</p>	<p>Основное свойство элемента. Это свойство должно быть связано с параметром, значение которого необходимо устанавливать или значение которого требуется отображать в клиенте визуализации. Соответственно, свойство может иметь входящую и исходящую связи. Если элемент был добавлен в окно из палитры редактора НМІ, то связи необходимо установить вручную. Если параметр типа TIME был перетащен в окно правой кнопкой мыши, и в контекстном меню был выбран пункт ДатаВремя, то данное поле заполнится автоматически. После установки связи данное свойство будет содержать дополнительные строки, в которых будет находиться информация о связях.</p>

Название	Описание
Категория Значения	
Формат времени	Определяет формат времени. По умолчанию свойство принимает значение HH:mm:ss.fff, где: HH – это часы, mm – минуты, ss – секунды, fff – миллисекунды. Например, если формат времени будет иметь вид HH:mm, то в поле часов будут отображаться только часы и минуты.
Только чтение	Если свойство имеет значение TRUE, то ввод значения параметра в клиенте визуализации запрещен. Если свойство принимает значение False, то при нажатии левой кнопкой мыши на элемент в клиенте визуализации откроется окно ввода значения параметра.

Описание других свойств элемента смотрите в разделе HMI.Свойства

Установка значения в клиенте визуализации

Тип транслятора HMI v1

В HMI v1 элемент имеет вид:

03.22:00:00

Для того чтобы отобразить инструменты задания интервала, необходимо нажать левой кнопкой мыши на элемент.

ДЕНЬ	ЧАСЫ	МИНУТЫ	СЕКУНДЫ
0	22	58	58
1	23	59	59
2	00	00	00
3	01	01	01
4	02	02	02
ОТМЕНА		УСТАНОВИТЬ	

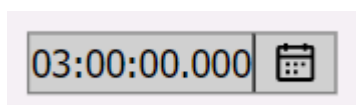
Для установки отдельных элементов интервала с помощью показанных выше инструментов можно использовать различные методы:

- Навести мышь на отдельный элемент интервала и нажать левой кнопкой мыши на нужное значение;
- Навести мышь на отдельный элемент интервала и вращением колеса мыши перемещать его до нужного значения;
- Навести мышь на отдельный элемент интервала и перемещать его левой кнопкой мыши вверх или вниз до нужного значения.

Для применения выполненных изменений следует нажать кнопку УСТАНОВИТЬ. По команде ОТМЕНА инструменты установки интервала скрываются, при этом выполненные изменения игнорируются. Для того чтобы скрыть инструменты установки интервала можно также нажать левой кнопкой мыши на произвольное место мнемосхемы:

Тип транслятора НМІ v2

В НМІ v2 элемент имеет вид:



Для изменения интервала можно воспользоваться одним из следующих способов:

- Нажать левой кнопкой мыши перед тем символом, значение которого необходимо изменить. При этом вокруг отображаемого времени появится синяя рамка и курсор. Затем, используя клавиатуру, можно ввести необходимое значение. При вводе можно перемещать курсор стрелками влево/вправо, использовать клавиши DEL, Backspace. Значение будет применено после нажатия на клавишу Enter, либо нажатия любой кнопки мыши за границей элемента.
- Нажать левой кнопкой мыши на правую область элемента. При этом откроется окно, в котором можно изменять отдельные части интервала, используя соответствующие стрелки, либо покрутив колесо мыши в нужном поле:

Для применения выполненных изменений интервала следует нажать кнопку ВЫБРАТЬ. По команде ОТМЕНА инструменты установки интервала скрываются, при этом выполненные изменения игнорируются. Для того чтобы скрыть инструменты установки интервала можно также нажать левой кнопкой мыши на произвольное место мнемосхемы.

8.3.1.14. ГОРИЗОНТАЛЬНЫЙ ПОЛЗУНОК

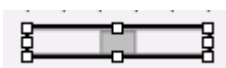
Данный элемент предназначен для установки значения аналогового параметра при помощи линейного регулятора в клиенте визуализации. Для изменения установленного значения следует передвинуть ползунок в нужное место левой кнопкой мыши. Чаще всего элементы Горизонтальный ползунок в окне создаются путем перетаскивания аналогового параметра в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.

В палитре редактора НМІ элемент Горизонтальный ползунок находится в категории Диалог.

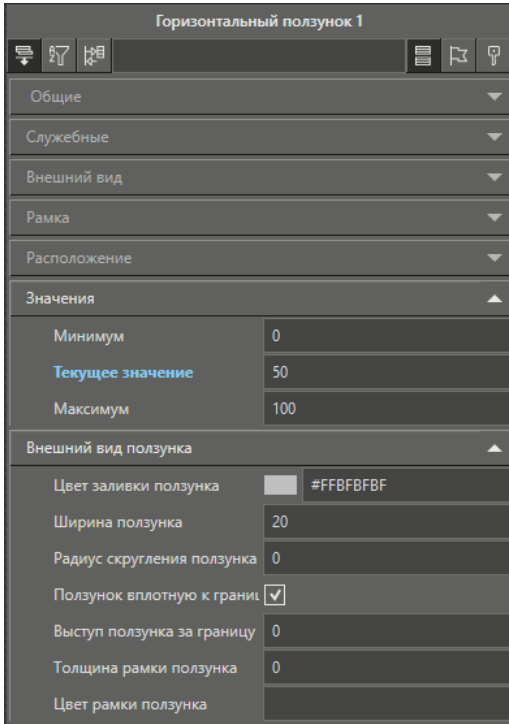
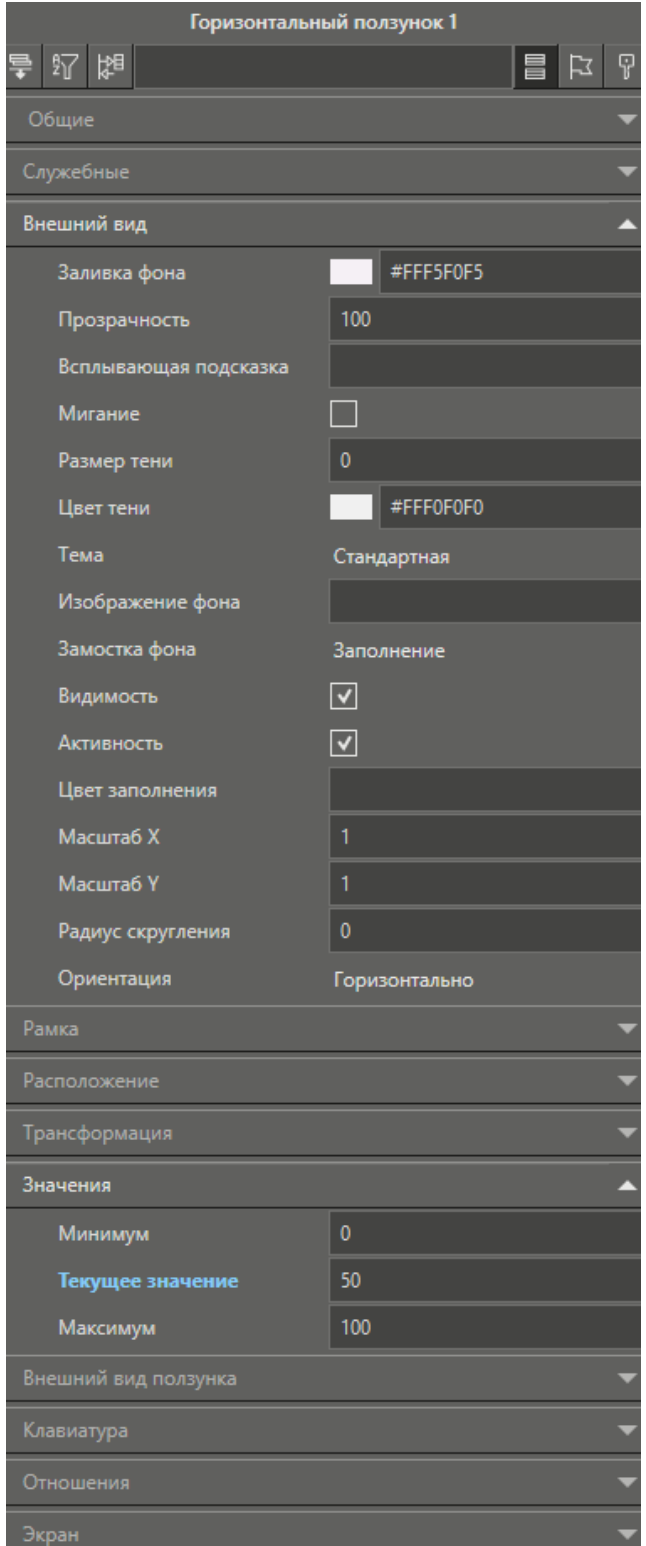
Вид элемента в палитре:





Вид элемента после добавления в рабочую область редактора НМІ:



Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
	

Описание основных свойств графических элементов Горизонтальный ползунок и Вертикальный ползунок:

Название	Описание
Категория Внешний вид	
Цвет заполнения	Определяет цвет закрашенной области ползунка. При наведении мыши на свойство появляется кнопка  , при нажатии на которую открывается диалоговое окно Выбор цвета.
Ориентация	Имеет два значения: Горизонтально и Вертикально. Элементы Горизонтальный ползунок и Вертикальный ползунок отличаются только значением этого свойства, заданным для них по умолчанию.
Категория Значения	
Минимум	Определяет минимально возможное значение, которое может быть установлено ползунком в клиенте визуализации.
Текущее значение	Данное свойство должно быть связано с каким-либо параметром проекта, текущее значение которого предполагается изменять в режиме исполнения. Свойство может иметь входящую и исходящую связи. Если элемент был добавлен в окно из палитры редактора НМІ, то связи необходимо установить вручную. Если аналоговый параметр был перетащен в окно правой кнопкой мыши, и в контекстном меню был выбран пункт Горизонтальный ползунок, то данное поле заполнится автоматически. После установки связи данное свойство будет содержать дополнительные строки, в которых будет находиться информация о связях.
Максимум	Определяет максимально возможное значение, которое может быть установлено ползунком в клиенте визуализации.
Категория Внешний вид ползунка	
Цвет заливки ползунка	Определяет цвет ползунка. При наведении мыши на свойство появляется кнопка  , при нажатии на которую открывается диалоговое окно Выбор цвета.
Ширина ползунка	Это свойство задает ширину ползунка (в px).

Название	Описание
Категория Внешний вид	
Радиус скругления ползунка	Задаёт радиус скругления углов ползунка (в px).
Ползунок вплотную к границе	Если у свойства задано состояние TRUE, то максимальное и минимальное значения будут достигнуты при касании ползунком края элемента. Если у свойства задано состояние False, то эти значения будут достигнуты при выходе ползунка за край элемента на величину, равную ширине ползунка.
Выступ ползунка за границу	Определяет величину выступа ползунка за границу сверху и снизу элемента (для элемента вертикальный ползунок – справа и слева).
Толщина рамки ползунка	Определяет толщину рамки ползунка.
Цвет рамки ползунка	Определяет цвет рамки ползунка. При наведении мыши на свойство появляется кнопка:  , при нажатии на которую открывается диалоговое окно Выбор цвета.

Описание других свойств элемента смотрите в разделе HMI.Свойства

Рекомендации по использованию

Для отображения устанавливаемого значения можно рядом с ползунком установить элемент текст, который будет иметь связь с тем же параметром, что и свойство текущее значение:



8.3.1.15. ВЕРТИКАЛЬНЫЙ ПОЛЗУНОК

Графический элемент Вертикальный ползунок аналогичен графическому элементу Горизонтальный ползунок, у которого свойство Ориентация имеет значение Вертикально.

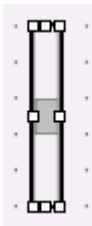
Чаще всего элементы Вертикальный ползунок в окне создаются путем перетаскивания аналогового параметра в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.

В палитре редактора HMI элемент Вертикальный ползунок находится в категории Диалог.

Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:



8.3.1.16. ГОРИЗОНТАЛЬНЫЙ ПРОГРЕСС

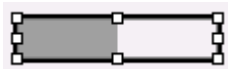
Данный элемент предназначен для отображения текущего значения аналогового параметра в виде показаний линейного индикатора. Чаще всего элементы Горизонтальный прогресс в окне создаются путем перетаскивания аналогового параметра в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.


В палитре редактора HMI элемент Горизонтальный прогресс находится в категории Диалог.

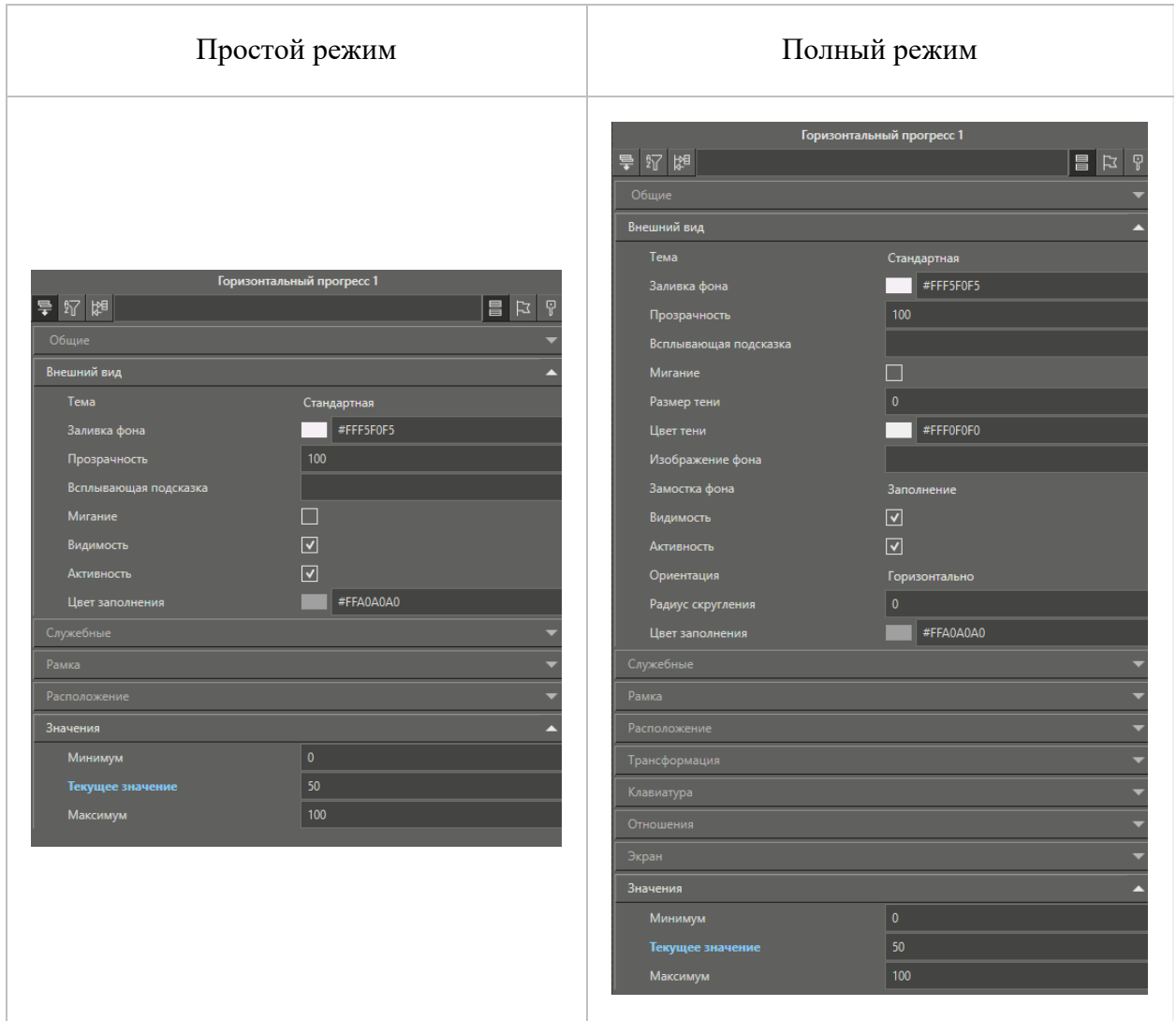
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:




Вид панели свойств элемента при нажатой и отжатой кнопке :



Описание основных свойств графического элемента Горизонтальный прогресс и Вертикальный прогресс:

Название	Описание
Категория Внешний вид	
Ориентация	<p>Имеет два значения: Горизонтально и Вертикально. Элементы Горизонтальный прогресс и Вертикальный прогресс отличаются только значением этого свойства, заданным для них по умолчанию.</p> <p>Цвет столбика элемента задается при помощи свойства цвет заполнения.</p>

Название	Описание
Категория Внешний вид	
Цвет заполнения	Определяет цвет закрашенной области прогресса. При наведении мыши на свойство появляется кнопка  , при нажатии на которую открывается диалоговое окно Выбор цвета.
Категория Значения	
Минимум	Задается минимально возможное значение, которое может отобразить элемент.
Текущее значение	Данное свойство должно быть связано с каким-либо параметром проекта, изменяющимся в режиме исполнения, за текущим значением которого предполагается следить в клиенте визуализации. Свойство может иметь только входящую связь. Свойство может иметь входящую и исходящую связи. Если элемент был добавлен в окно из палитры редактора НМІ, то связи необходимо установить вручную. Если аналоговый параметр был перетащен в окно правой кнопкой мыши, и в контекстном меню был выбран пункт Горизонтальный прогресс, то данное поле заполнится автоматически. После установки связи данное свойство будет содержать дополнительные строки, в которых будет находиться информация о связях.
Максимум	Задается максимально возможное значение, которое может отобразить элемент.

Описание других свойств элемента смотрите в разделе НМІ.Свойства

8.3.1.17. ВЕРТИКАЛЬНЫЙ ПРОГРЕСС

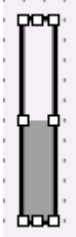
Графический элемент Вертикальный прогресс аналогичен графическому элементу горизонтальный прогресс, у которого свойство ориентация имеет значение вертикально. Чаще всего элементы Вертикальный прогресс в окне создаются путем перетаскивания аналогового параметра в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.

В палитре редактора НМІ элемент Вертикальный прогресс находится в категории Диалог.

Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:



8.3.1.18. КНОПКА ВЫБОРА ЦВЕТА

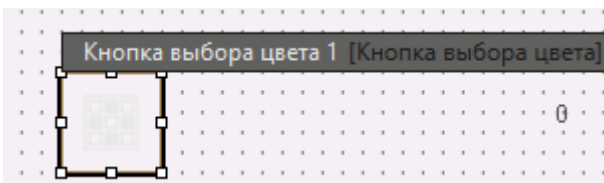
Данный элемент предназначен для выбора цвета в клиенте визуализации. Если параметр проекта будет иметь входящую связь со свойством Значение кнопки и исходящую связь с каким-либо свойством графического элемента, отвечающего за цвет, то появится возможность менять цвет элементов в клиенте визуализации, например, цвет пера графика.


В палитре редактора HMI элемент Кнопка выбора цвета находится в категории Диалог.

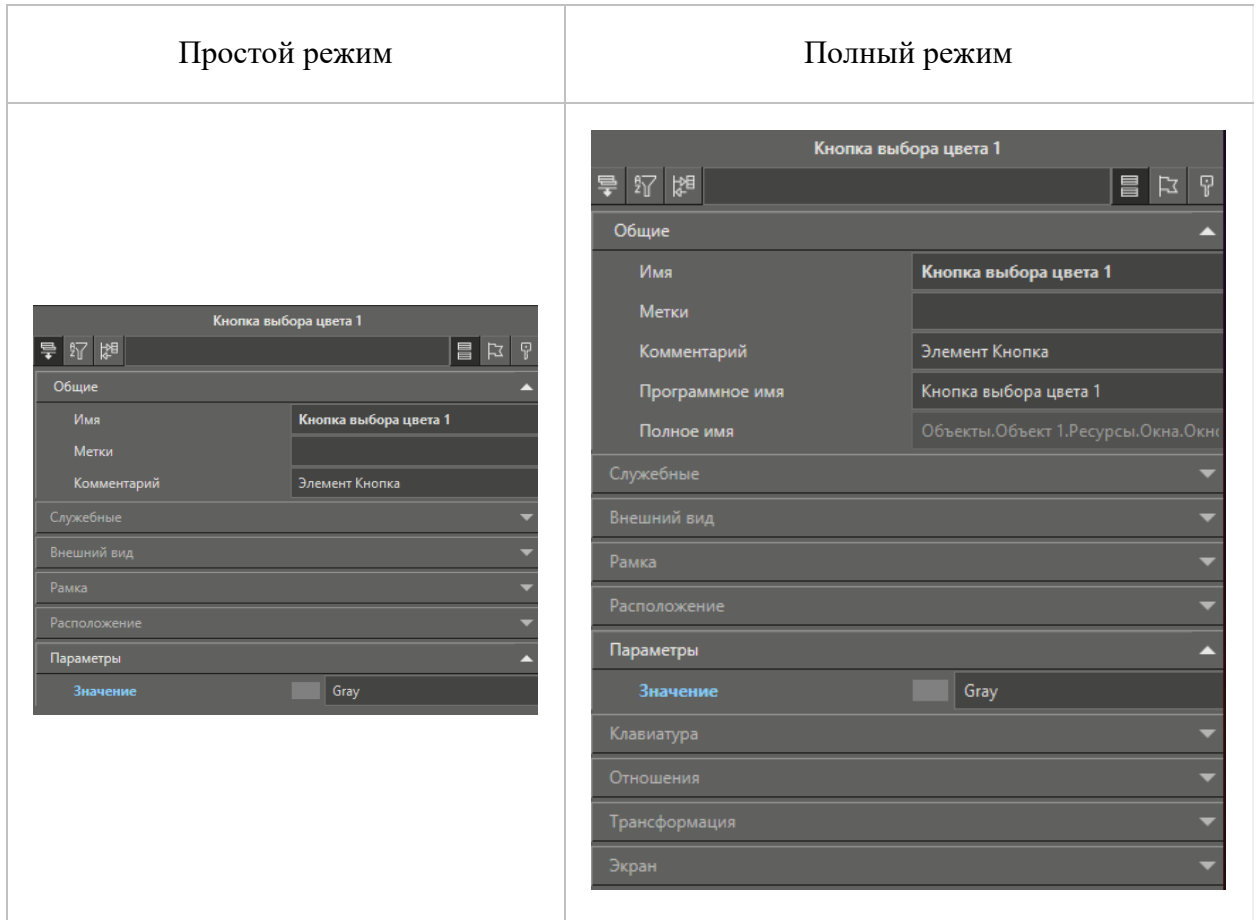
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:



Вид панели свойств элемента при нажатой и отжатой кнопке :



Описание основных свойств графического элемента Кнопка выбора цвета:

Название	Описание
<p>Категория Параметры</p>	
<p>Значения</p>	<p>Основное свойство элемента, которое необходимо связать с параметром типа STRING. В этом свойстве будет указан цвет (RGB-значение цвета), выбранный в клиенте визуализации. Может иметь входящую и исходящую связи. Если кнопка связана с параметром окна, не имеющим связи с параметрами объектов и узлов, то допускается выбрать в разных клиентах разное цветовое оформление.</p>

Описание других свойств элемента смотрите в разделе HMI.Свойства

Элемент в клиенте визуализации

В клиенте визуализации элемент имеет вид:



Цвет кнопки, зависит от выбранного в ней цвета.

После нажатия на кнопку открывается диалоговое окно, в котором можно выбрать требуемый цвет. В зависимости от выбранной версии HMI вид диалогового окна будет разный.

Окно выбора цвета в HMI v1

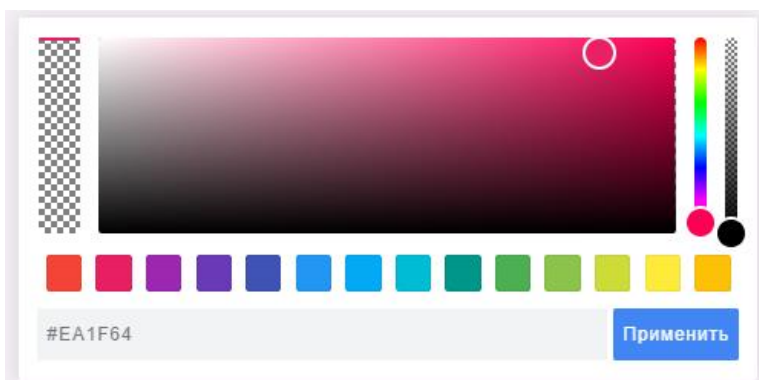
В HMI v1 окно имеет вид:



Изменения вступят в силу незамедлительно после нажатия левой кнопкой мыши на выбранный цвет.

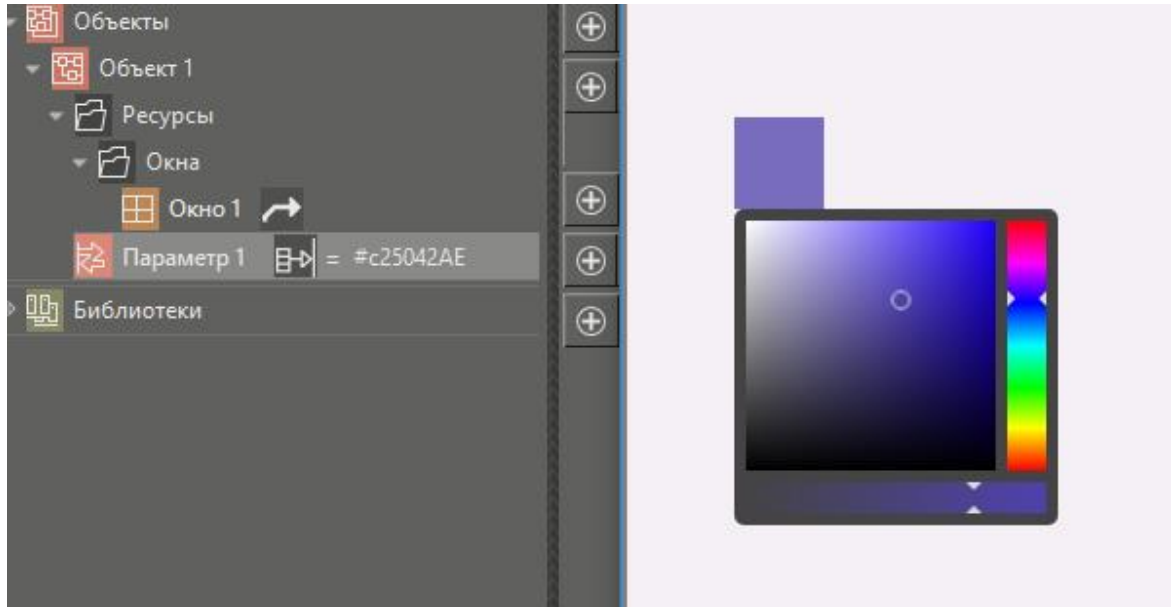
Окно выбора цвета в HMI v2

В HMI v1 окно имеет вид:



Изменения вступят в силу после нажатия на кнопку Применить.

После выбора цвета значение связанного параметра отобразится в формате HEX:



8.3.1.19. СТРЕЛОЧНЫЙ ПРИБОР

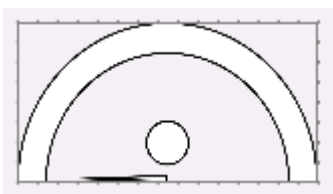
Данный элемент предназначен для отображения значения аналогового параметра в виде показаний стрелочного индикатора. Чаще всего элементы Стрелочный прибор в окне создаются путем перетаскивания аналогового параметра в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.


В палитре редактора НМІ элемент Стрелочный прибор находится в категории Диалог.

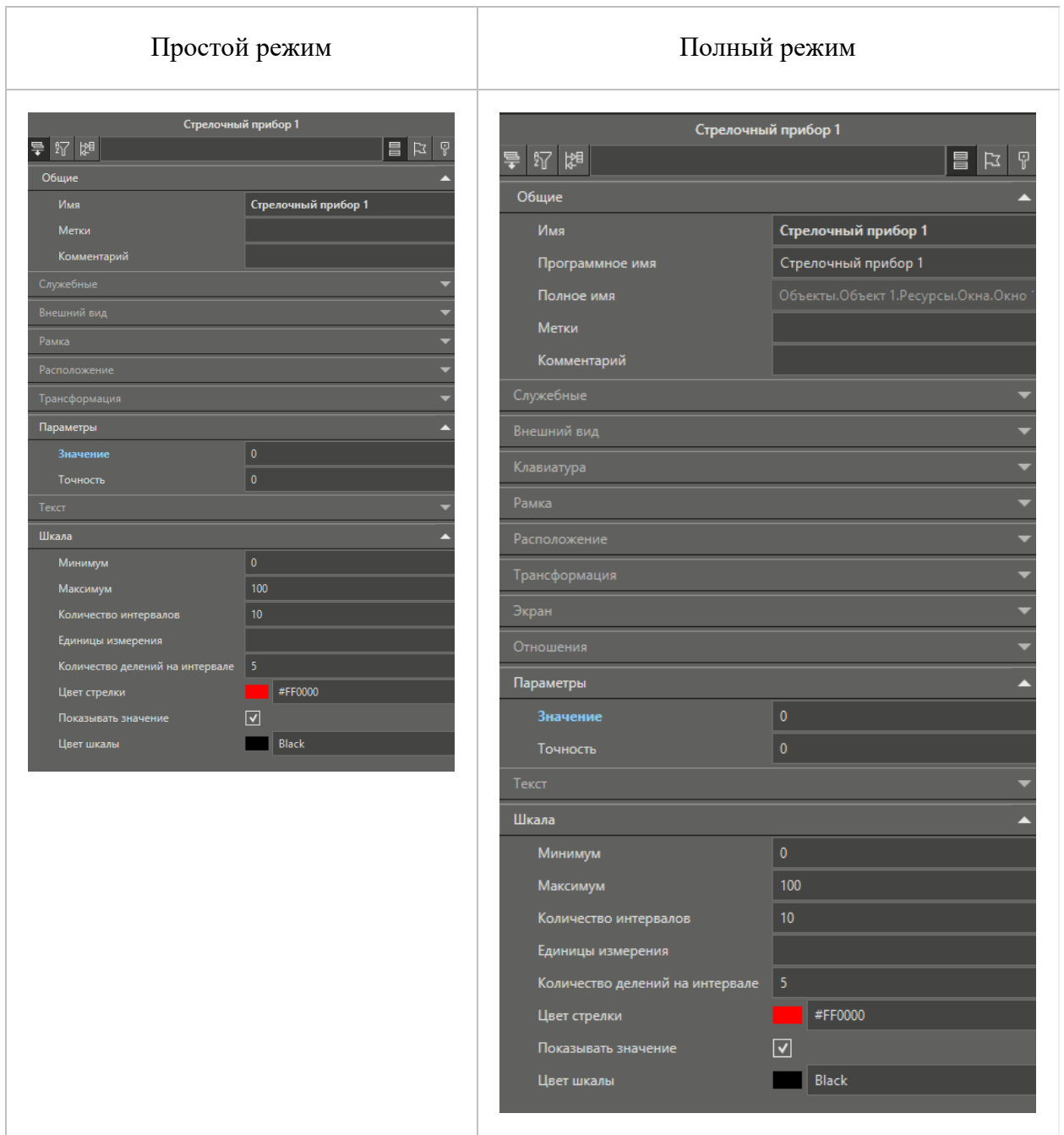
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора НМІ:





Вид панели свойств элемента при нажатой и отжатой кнопке  :



Описание основных свойств графического элемента Стрелочный прибор:

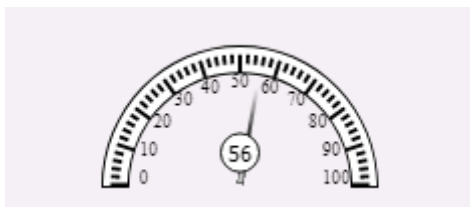
Название	Описание
Категория Шкала	
Максимум	Максимальное значение, которое может показывать элемент.
Минимум	Минимальное значение, которое может показывать элемент.

Название	Описание
Количество интервалов	Это свойство имеет тип INT и задает количество больших делений шкалы.
Единицы измерения	Это свойство имеет тип STRING и задает единицы измерения шкалы. Влияет только на отображение символов шкалы в графическом элементе, т.е. если у параметра назначена шкала и задана единица измерения, например, в Па, а в графическом элементе задали значение свойства в мм. рт.ст., то конвертация значения параметра производится не будет.
Количество делений на интервале	Это свойство имеет тип INT и задает количество маленьких делений в одном большом делении шкалы.
Цвет стрелки	Определяет цвет стрелки элемента. При нажатии на кнопку  , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.
Показывать значение	Это свойство имеет тип BOOL. Если оно имеет значение TRUE, то на шкале отображается численное значение параметра и единицы измерения, заданные свойством Единицы измерения.
Цвет шкалы	Определяет цвет шкалы элемента. При нажатии на кнопку  , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.
Категория Параметры	
Значение	Данное свойство должно быть связано с каким-либо параметром проекта, изменяющимся в режиме исполнения, за значением которого предполагается следить в клиенте визуализации. Свойство может иметь только входящую связь. Если элемент был добавлен в окно из палитры редактора НМІ, то связи необходимо установить вручную. Если аналоговый параметр был перетащен в окно правой кнопкой мыши, и в контекстном меню был выбран пункт Стрелочный прибор, то данное поле заполнится автоматически. После установки связи данное свойство будет

Название	Описание
	содержать дополнительные строки, в которых будет находиться информация о связях.
Точность	Это свойство имеет тип LREAL. Определяет количество знаков после запятой будет отображаться в цифровом табло стрелочного прибора.

Описание других свойств элемента смотрите в разделе HMI.Свойства

Вид элемента клиенте визуализации:



8.3.1.20. ГОРИЗОНТАЛЬНЫЙ ПОЛЗУНОК ИНТЕРВАЛА

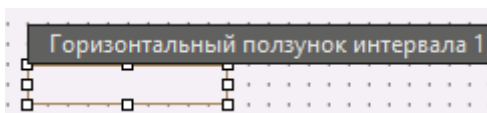
Данный элемент предназначен для установки интервала в клиенте визуализации. В элементе можно установить интервал и перемещать его влево и вправо. Для изменения установленного значения следует передвинуть ползунок в нужное место левой кнопкой мыши.

В палитре редактора HMI элемент Горизонтальный ползунок интервала находится в категории Диалог.

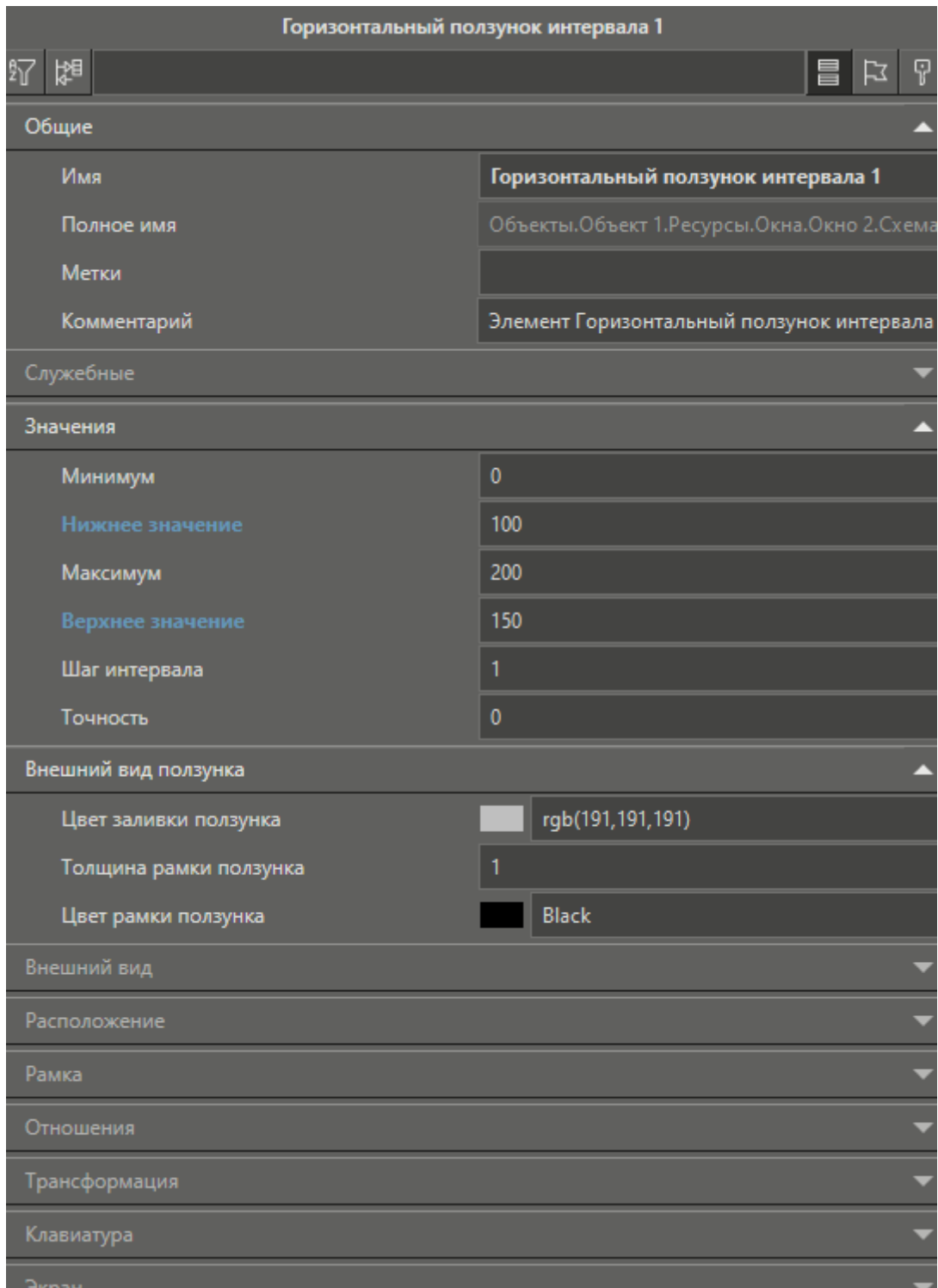
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:




Вид панели свойств элемента:



Описание основных свойств графических элементов Горизонтальный ползунок интервала и Вертикальный ползунок интервала:

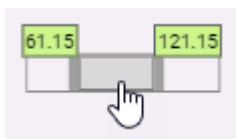
Название	Назначение
Категория Значения	
Минимум	Определяет минимально возможное Нижнее значение, которое может быть установлено ползунком в клиенте визуализации.
Нижнее значение	Определяет нижнее значение интервала. Данное свойство должно быть связано с каким-либо параметром проекта, отвечающим за нижнее значение интервала, значение которого предполагается изменять в режиме исполнения. Связи необходимо установить вручную. После установки связи данное свойство будет содержать дополнительные строки, в которых будет находиться информация о связях.
Максимум	Определяет максимально возможное Верхнее значение, которое может быть установлено ползунком в клиенте визуализации.
Верхнее значение	Определяет верхнее значение интервала. Данное свойство должно быть связано с каким-либо параметром проекта, отвечающим за нижнее значение интервала, значение которого предполагается изменять в режиме исполнения. Связи необходимо установить вручную. После установки связи данное свойство будет содержать дополнительные строки, в которых будет находиться информация о связях.
Шаг интервала	Минимально возможная разница между верхним и нижним значениями интервала.
Точность	Определяет, какое количество знаков после запятой будет использоваться для указания нижнего и верхнего значения.
Категория Внешний вид ползунка	
Цвет заливки ползунка	Определяет цвет ползунка. При наведении мыши на свойство появляется кнопка  , при нажатии на которую открывается диалоговое окно Выбор цвета.

Название	Назначение
Категория Значения	
Толщина рамки ползунка	Определяет толщину рамки ползунка.
Цвет рамки ползунка	Определяет цвет рамки ползунка. При наведении мыши на свойство появляется кнопка:  , при нажатии на которую открывается диалоговое окно Выбор цвета.
Категория Внешний вид	
Ориентация	Имеет два значения: Горизонтально и Вертикально. Элементы Горизонтальный ползунок интервала и Вертикальный ползунок интервала отличаются только значением этого свойства, заданным для них по умолчанию.

Описание других свойств элемента смотрите в разделе НМІ.Свойства

Вид элемента в клиенте визуализации

При наведении мыши на ползунок над элементом отобразится текущее значение интервала. Указатель мыши будет иметь вид:



В таком режиме ползунок можно перемещать влево и вправо - разница между верхним и нижним значением останется неизменной, но верхнее и нижнее значение будут изменяться.



Если подвести мышь к границе ползунка, то указатель мыши примет вид горизонтальной двунаправленной стрелки, в этом случае нажимая и удерживая левую кнопку мыши будет изменяться разница между верхним и нижним значением.

8.3.1.21. ВЕРТИКАЛЬНЫЙ ПОЛЗУНОК ИНТЕРВАЛА

Графический элемент Вертикальный ползунок интервала аналогичен графическому элементу Горизонтальный ползунок интервала, у которого свойство Ориентация имеет

значение Вертикально. В элементе можно установить интервал и перемещать его вверх и вниз. Для изменения установленного значения следует передвинуть ползунок в нужное место левой кнопкой мыши.

В палитре редактора HMI элемент Вертикальный ползунок находится в категории Диалог.

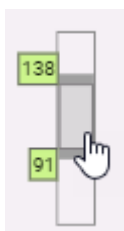
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:



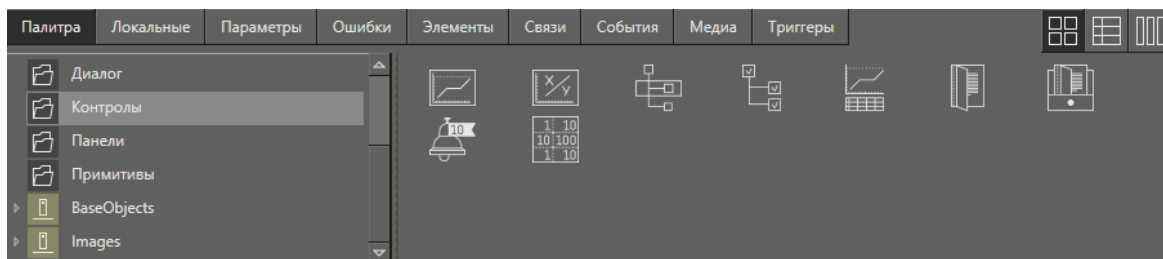
Вид в клиенте визуализации:



8.3.2. КАТЕГОРИЯ КОНТРОЛЫ

Категория Контролы палитры редактора HMI содержит элементы, предназначенные для работы с архивными и текущими данными, сообщениями, а также для отображения сложных типов данных в клиенте визуализации.

В палитре редактора HMI эта категория имеет вид:



Категория Контролы включает в себя следующие элементы:

- Тренд

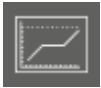
- График XY
- Дерево одиночного выбора
- Дерево множественного выбора
- Легенда
- Журнал и Архивный журнал
- Индикатор событий
- Таблица данных
- Справочник

8.3.2.1. ТРЕНД

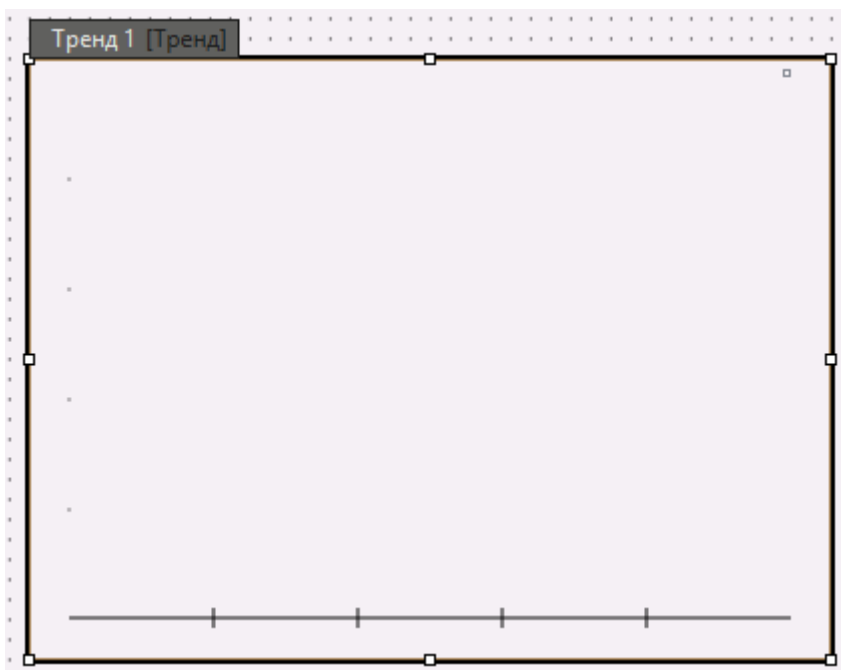
Графический элемент Тренд предназначен для отображения изменения значения параметра во времени. Данный элемент может работать как с архивными, так и с текущими значениями.

В палитре редактора НМІ элемент Тренд находится в категории Контролы.

Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора НМІ:



Для того чтобы добавить параметры, которые должны отображаться на тренде, необходимо перетащить параметр из дерева в поле тренда. При этом добавится перо. Кроме того,

изменится внешний вид элемента: появится минилегенда в верхнем левом углу элемента. Одновременно с этим в панели свойств добавится группа свойств, относящаяся к добавленному перу.

Важно! На Тренд можно добавить несколько выделенных параметров одновременно.

Если параметр, связанный с пером, будет удален, или если будет разорвана связь между параметром и пером в контекстном меню параметра, то перо при этом останется в тренде, а все настройки, например, цвет, тип линии сохранятся. Для удаления созданного пера следует воспользоваться панелью свойств тренда.

Вид тренда в клиенте визуализации и способы работы с ним зависят от типа транслятора НМІ. Рекомендуется использовать тип транслятора НМІ v2.

Внешний вид тренда может быть изменен при помощи панели свойств. Любое свойство может быть задинамизировано, таким образом, разработчик проекта может создать те элементы управления трендом, которые необходимы пользователю.

Настройки внешнего вида будут применены в клиенте визуализации в режиме исполнения. Для того, чтобы оценить изменения, не загружая конфигурацию в узел, можно воспользоваться кнопкой Просмотр панели инструментов

8.3.2.1.1. СВОЙСТВА ТРЕНДА

Вид панели свойств элемента Тренд:

Тренд 1

Общие

Служебные

График

Интервал	00:01:00
Титул	
Отображать легенду	<input checked="" type="checkbox"/>
Автопрокрутка	<input checked="" type="checkbox"/>
Период обновления данных	100
Минимальный вид	<input type="checkbox"/>
Заливка графика	<input type="checkbox"/> White
+ Перья	<input type="checkbox"/> <input type="checkbox"/>
Конец	00:00:00
Ширина минилегенды	250
Дерево выбора перьев	<input type="checkbox"/>
Максимальное количество перьев	100
Печать минилегенды	<input type="checkbox"/>

Ось X

Ось Y

Текст

Параметры

Курсор выбранной позиции

Сетка

Архив

Начало архива	0001-01-01-00:00
Конец архива	0001-01-01-00:00

Внешний вид

Расположение

Рамка

Отношения


Трансформация



Клавиатура


Экран


Описание основных свойств графического элемента Тренд:

Название	Описание
Категория Архив	
Начало архива	Используется, если на тренде отображается архивируемый параметр. Данное свойство имеет тип DT и отображает начало выборки из архива.
Конец архива	Используется, если на тренде отображается архивируемый параметр. Данное свойство имеет тип DT и отображает конец выборки из архива.
Категория График	
Интервал	Это свойство имеет тип TIME и задает для тренда временной интервал между крайними точками поля тренда по оси X.
Титул	Свойство определяет заголовок тренда. Если свойство задано, то заголовок окна выбора перьев будет иметь вид: Выбор перьев - [Титул]
Автопрокрутка	Данное свойство имеет тип BOOL и может принимать значение TRUE (автоматическая прокрутка отображаемого элементом тренд содержимого включена) или False (автоматическая прокрутка выключена). Если автоматическая прокрутка включена, то на тренде будут отображаться значения параметров за Интервал, предшествующий текущему моменту времени.
Период обновления данных	Это свойство имеет тип INT и задает период (в мс) обновления данных на тренде, если включена автопрокрутка.
Минимальный вид	Если данное свойство имеет значение TRUE, то на тренде скрываются все элементы, кроме поля тренда, которое, в этом случае, занимает всю площадь тренда. При этом управление трендом можно осуществлять элементами, созданными разработчиком проекта. Например, для изменения интервала

Название	Описание
	можно использовать текстовый ввод, который связан с параметром, передающим данные в свойство тренда Интервал.
Заливка графика	Определяет цвет заливки поля тренда. При наведении мыши на свойство появляется кнопка:  , после нажатия на которую открывается диалоговое окно Выбор цвета.
Группа Перья	После нажатия на кнопку  можно добавлять перья на тренд. Данная группа содержит подгруппы, количество подгрупп соответствует количеству перьев на тренде. Описание свойств подгрупп описано в таблице ниже.
Конец	Данное свойство имеет тип DT. Крайняя точка справа по оси X.
Дерево выбора перьев	Определяет возможность добавлять перья в клиенте визуализации. Если флаг установлен, то в панели инструментов появляется иконка вызывающая вызывающей элемент, позволяющий добавлять параметры на тренд в режиме исполнения
Максимальное количество перьев	Определяет максимальное количество перьев, которые могут располагаться на тренде одновременно.
Категория Курсор выделенной позиции	
Цвет курсора	Определяет цвет курсора. При наведении мыши на свойство появляется кнопка:  , после нажатия на которую открывается диалоговое окно Выбор цвета.
Толщина курсора	Определяет толщину линии курсора. Задается в пикселях.
Видимость курсора	Это свойство имеет тип BOOL и задает видимость курсора выбранной позиции на тренде.
Категория Ось X	

Название	Описание
Подпись X	Подпись оси X. Если задан текст, то он отобразится в клиенте визуализации около оси X правее поля тренда.
Формат значений X	Задаёт формат значений временной оси X тренда. Например, dd.MM.yyyy HH:mm:ss:fff - в этом случае будут отображаться на тренде в виде - 25.11.2017 22:30:01:129.
Количество делений X	Задаётся требуемое количество делений оси. При построении тренда среда исполнения будет выбирать значение этого параметра по возможности близко к заданному.
Толщина делений X	Определяет внешний вид деления шкалы.
Длина делений X	Определяет внешний вид деления шкалы.
Цвет делений X	Определяет цвет делений оси. При наведении мыши на свойство появляется кнопка:  , после нажатия на которую открывается диалоговое окно Выбор цвета.
Верхний отступ делений X	Определяет расстояние от оси до значения, соответствующего тому или иному делению.
Цвет оси X	Определяет цвет оси. При наведении мыши на свойство появляется кнопка:  , после нажатия на которую открывается диалоговое окно Выбор цвета.
Толщина оси X	Определяет толщину оси.
Категория Ось Y	
Подпись Y	Подпись оси Y. Если задан текст, то он отобразится в клиенте визуализации около оси Y над полем тренда.
Длина делений Y	Определяет внешний вид деления шкалы

Название	Описание
Правый отступ делений Y	Определяет расстояние от оси до значения, соответствующего тому или иному делению.
Цвет делений Y	Определяет цвет делений оси. При наведении мыши на свойство появляется кнопка:  , после нажатия на которую открывается диалоговое окно Выбор цвета.
Толщина оси Y	Определяет толщину оси.
Толщина делений Y	Определяет внешний вид деления шкалы.
Автоматически создавать оси значений	Определяет количество осей Y (осей значений). Если флаг установлен, то для каждого пера, отображаемого в данный момент на тренде, может отображаться отдельная ось значений (в зависимости от настроек, каждого отдельного пера). Оси удалять нельзя. Если флаг снят, то отображается одна ось, которая отображает диапазон выделенного в данный момент пера.
Масштабировать по всем осям	Определяет поведение графиков на тренде при использовании вертикального ползунка. Если флаг установлен, то при изменении положения ползунка будут масштабироваться все графики, находящиеся в поле тренда.
Категория Параметры	
Шаг прореживания	Данную настройку полезно использовать в случае, когда на тренде необходимо отобразить большее количество данных, чем может поместиться физически. Поскольку в одном пикселе отображается только одно значение, то не рационально считать из архива данных больше, чем может быть использовано на тренде. Настройка определяет, на какое число пикселей должна приходиться группа из не более чем четырех значений. Для этого вычисляется интервал данных по формуле $= \text{Интервал времени видимой области графика} / (\text{Ширина графика (в px)} * \text{Шаг прореживания})$. Выборка из архива делается с тем условием, что за каждый такой интервал данных из него придет не более четырех значений (первое, последнее,

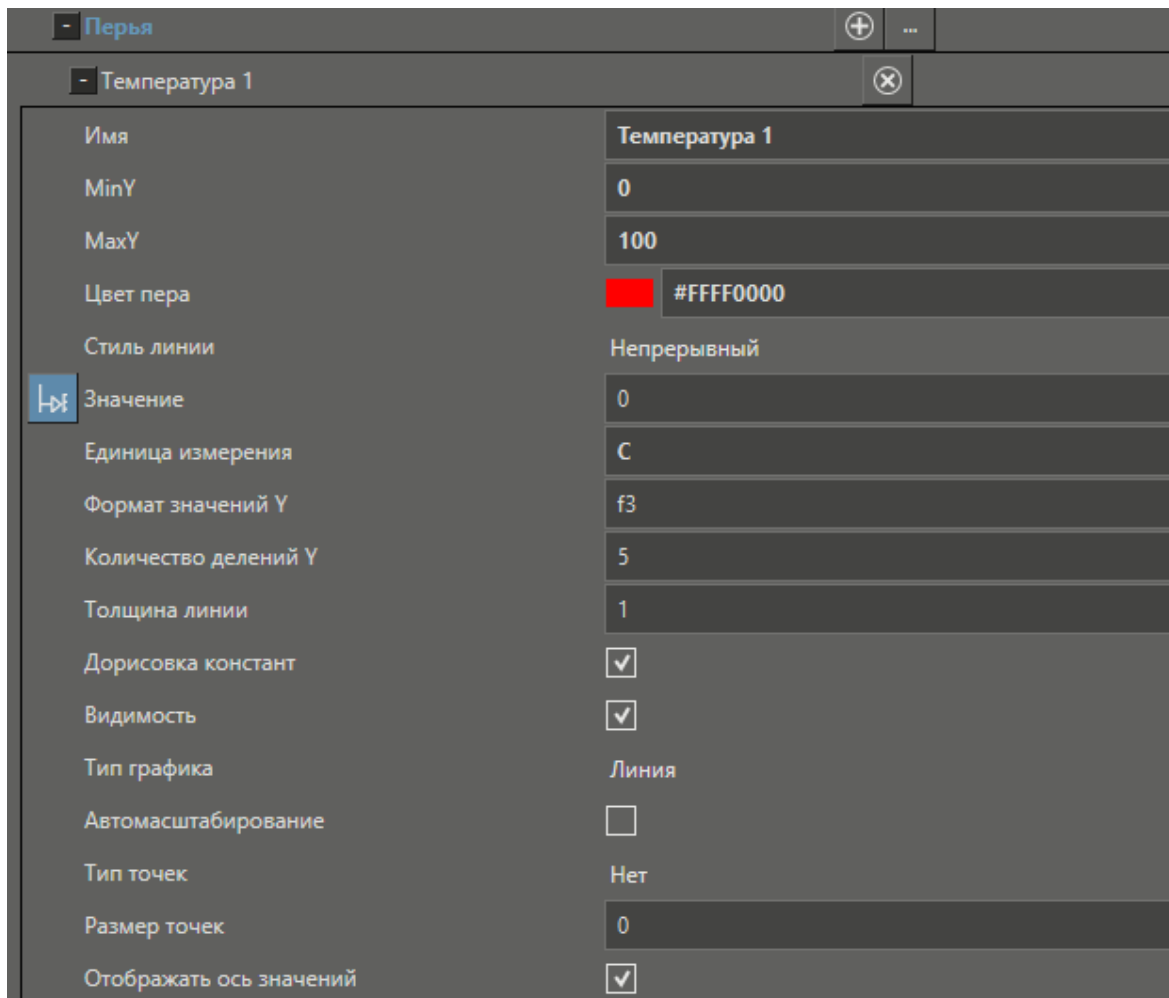
Название	Описание
	минимальное, максимальное). По умолчанию Шаг прореживания равен 1.
Интервал Листания	Это свойство имеет тип LREAL и в процентах указывает, на какую часть интервала необходимо смещать окно при листании тренда вперед/назад, при использовании действия Вызвать метод.
Категория Сетка	
Цвет сетки	Определяет цвет линий сетки. При наведении мыши на свойство появляется кнопка:  , после нажатия на которую открывается диалоговое окно Выбор цвета.
Категория Минилегенда	
Отображать легенду	Данное свойство имеет тип BOOL и может принимать значение TRUE (отображать легенду тренда) или False (не отображать).
Ширина минилегенды	Определяет ширину минилегенды в клиенте визуализации
Печать минилегенды	Определяет необходимость включения минилегенды в изображение при печати тренда или при его сохранении в файл. Если флаг установлен, то при сохранении/печати в изображение включается минилегенда, даже если была отключена на момент печати.
Прозрачность минилегенды	Задаёт степень прозрачности минилегенды. Значение изменяется в пределах от 0 до 100, где 0 - элемент полностью прозрачный, 100 - элемент полностью не прозрачный


Свойства группы Перья


Количество настроек соответствует типу данных TrendPen, который входит в состав библиотеки HMI. Типы данных.

Набор перьев представляет собой библиотечный тип TrendPens. Эту информацию можно использовать для создания трендов, на которых набор перьев будет меняться при помощи программ в клиенте визуализации в режиме исполнения.

Вид:





Новая группа добавится автоматически, после того как разработчик перетащит новый параметр в поле тренда. Кроме того, группа настроек может быть добавлена вручную с использованием кнопки .

Для удаления группы настроек необходимо нажать на кнопку: .

Описание настроек:

Название	Описание
Имя	Определяет название пера, которое будет отображаться в Легенде.
MaxY	Максимальное значение оси Y.
MinY	Минимальное значение оси Y.

Название	Описание
Автомасштабирование	Автоматически определяет значения MaxY и MaxX таким образом, чтобы тренд занимал всю площадь поля тренда.
Видимость	Определяет видимость пера в тренде. Данное свойство может быть задинамизировано, т.е. зависеть от других элементов управления, созданных разработчиком проекта, либо от какой-то программы. Если свойство пера примет значение False, то скроется не только перо, но и его ось значений.
Дорисовка констант	Определяет как будет рисоваться тренд, если значение параметра не будет меняться. Если флаг отмечен, то будет дорисовываться прямая линия. Если флаг снят, то график рисоваться не будет, до тех пор, пока не будет получено новое значение.
Единица измерения	Определяется единица измерения отображаемого параметра. Заданное значение будет отображаться в Минилегенде. Если параметр перетаскивали в поле тренда и образовалось перо, и у этого параметра настроена шкала, то эта настройка пера примет значение установленное в шкале параметра.
Значение	Настраивается связь между параметром и пером. В данное поле необходимо перетащить параметр, который требуется отображать на тренде в случае, если перо было создано вручную. Если параметр был перетащен в поле тренда, то данное свойство заполнится автоматически. Т.е. данное свойство обязательно должно быть задинамизировано.
Количество делений Y	Определяет количество делений оси Y.
Стиль линии	Определяет стиль линии тренда. Выпадающий список содержит следующие возможные значения: Непрерывный, Пунктир, Точка, Нет.
Тип графика	Определяет способ рисования тренда между точками тренда. Возможные варианты:

Название	Описание
	<ul style="list-style-type: none"> • Линия - между двумя точками будет рисоваться прямая линия, т.е. считается, что значение параметра менялось плавно. • Ступенька - между двумя точками будет рисоваться ступенька, т.е. считается, что значение изменилось скачкообразно, т.е. до тех пор, пока значение не обновилось, система будет считать, что оно было равно предыдущему значению. <p>Разработчик проекта должен выбирать это свойство, исходя из особенностей технологического процесса. На рисунке у красного пера свойство имеет значение Линия, а у зеленого - Ступенька:</p> 
Толщина линии	Определяет толщину линии пера.
Формат значений Y	Определяет формат значений оси Y., Подробное описание свойства смотрите в разделе Формат значений.
Цвет пера	Определяет цвет пера тренда. При наведении мыши на свойство появляется кнопка:  , после нажатия на которую открывается диалоговое окно Выбор цвета. Если перетащить в поле тренда несколько параметров, то у связанных с ними перьев автоматически выберутся разные цвета.

Название	Описание
Тип точек	<p>Определяет тип точек, нанесенных на тренд. Точка рисуется в местах излома тренда:</p> <ul style="list-style-type: none"> • Нет • Квадрат • Круг • Треугольник • Ромб • Крест
Размер точек	Определяет размер точек.
Отображать ось значений	Определяет видимость оси пера, если установлен флаг в свойстве Автоматически создавать оси значений в категории свойств Ось Y

Описание других свойств элемента смотрите в разделе НМІ.Свойства.

Настройки внешнего вида будут применены в клиенте визуализации в режиме исполнения. Чтобы оценить изменения, не загружая конфигурацию в узел, можно воспользоваться кнопкой Просмотр панели инструментов.

8.3.2.1.2. ТРЕНД В КЛИЕНТЕ ВИЗУАЛИЗАЦИИ

Вид Тренда в клиенте визуализации и способы работы с ним зависят от типа транслятора НМІ:

Тренд в RT при типе транслятора НМІ v1

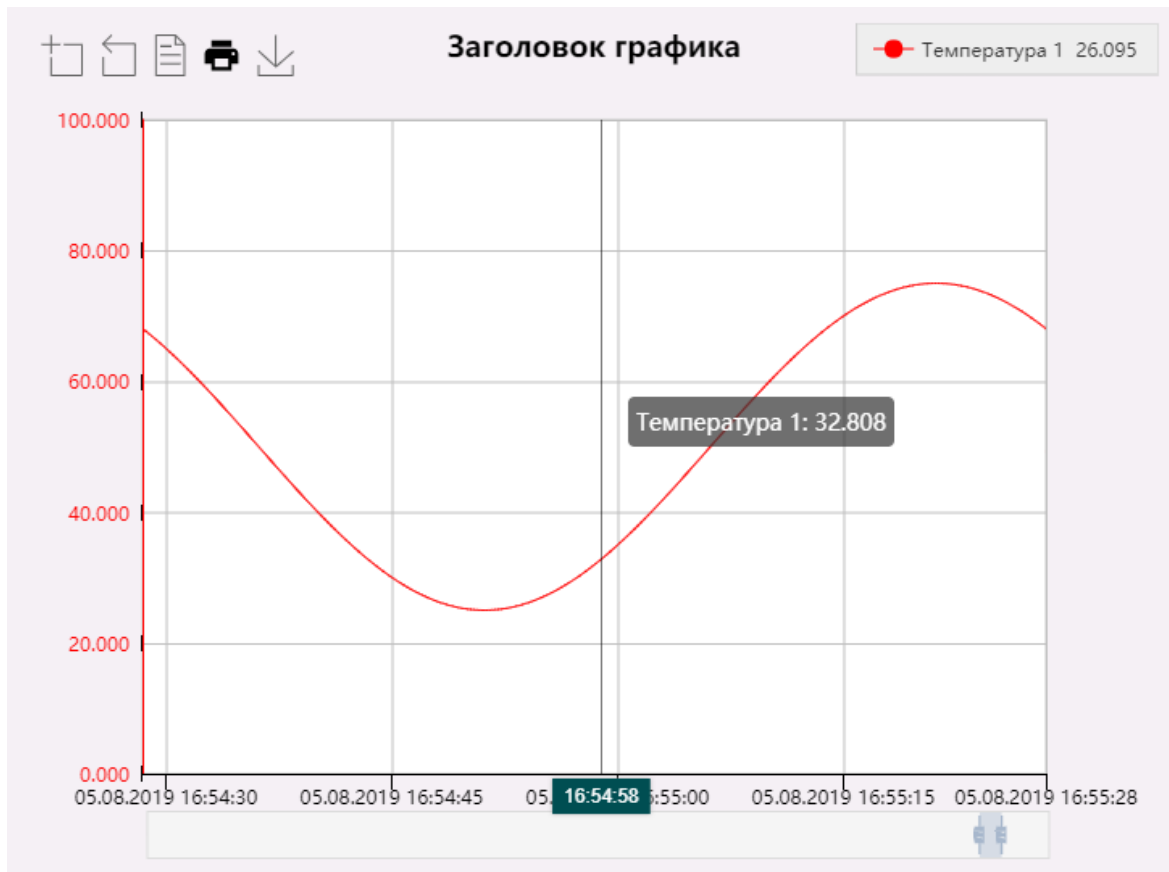
Тренд в RT при типе транслятора НМІ v2

8.3.2.1.2.1. ТРЕНД В КЛИЕНТЕ ВИЗУАЛИЗАЦИИ ПРИ НМІ V1






Данный графический элемент отображает зависимость значений параметров (ось Y) от времени (ось X) и содержит следующие области/элементы:


Вид элемента в клиенте визуализации зависит от его настроек.

Возможный вид графика:



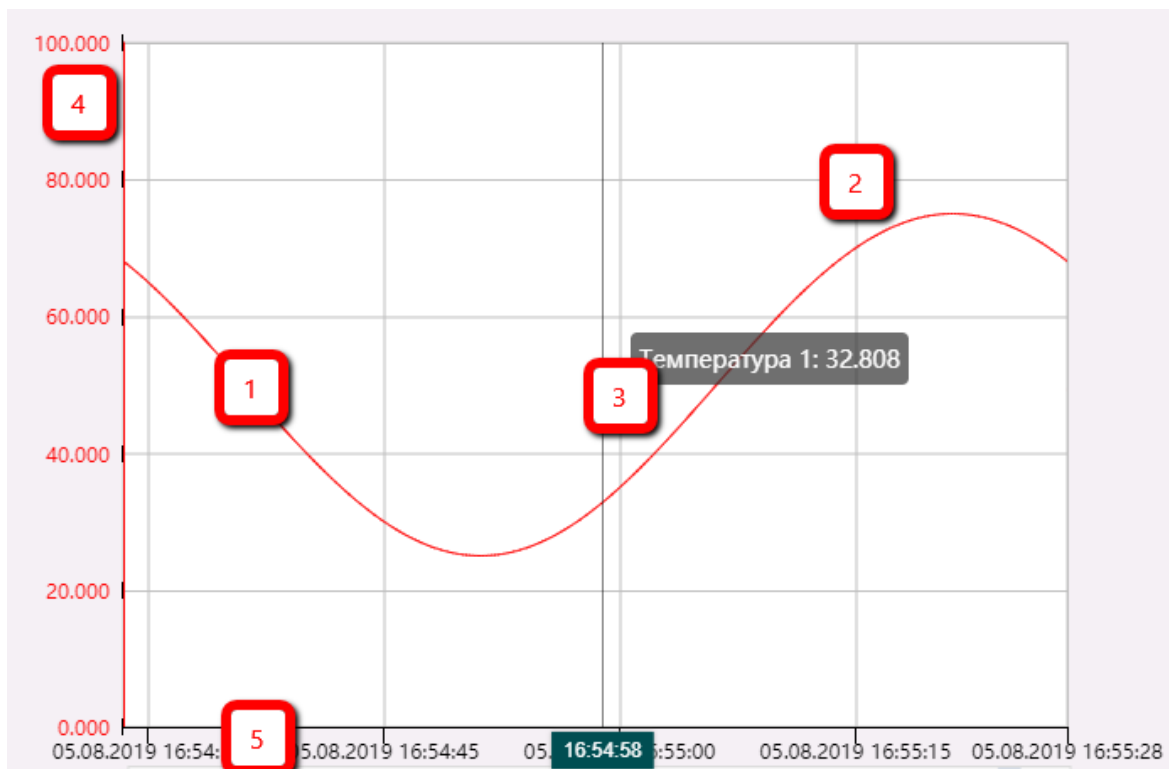
В левом верхнем углу располагается панель инструментов. Описание элементов:

Элемент	Описание
 Масштабировани е	Включает режим Масштабирование. В данном режиме возможно увеличить нужную часть графика, выделив ее мышью. Если режим включен, то курсор мыши на экране имеет вид: 
 Вернуть назад	Отменяет последнее действие сделанное в режиме Масштабирование. после отмены самого первого действия включится режим автопрокрутки
 Таблица данных	Представляет видимую часть графика в табличном виде.
 Распечатать	Позволяет распечатать видимую часть графика. По умолчанию откроется диалоговое окно выбора принтера

Элемент	Описание
 Сохранить png	Сохраняет видимую область графика в графический файл. По умолчанию откроется диалоговое окно выбора места сохранения файла.

В верхней средней части находится заголовок графика, но только если он был ранее настроен в свойствах графика Титул. В верхнем правом углу находится Минилегенда, в которой показывается список перьев, которые отображаются на графике, цвет пера, а также последнее полученное значение средой исполнения.

В средней части элемента находится собственно само поле графика, область отображения кривых. Состав поля графика:



Элемент	Рекомендации
1 Кривая	
2. Сетка	

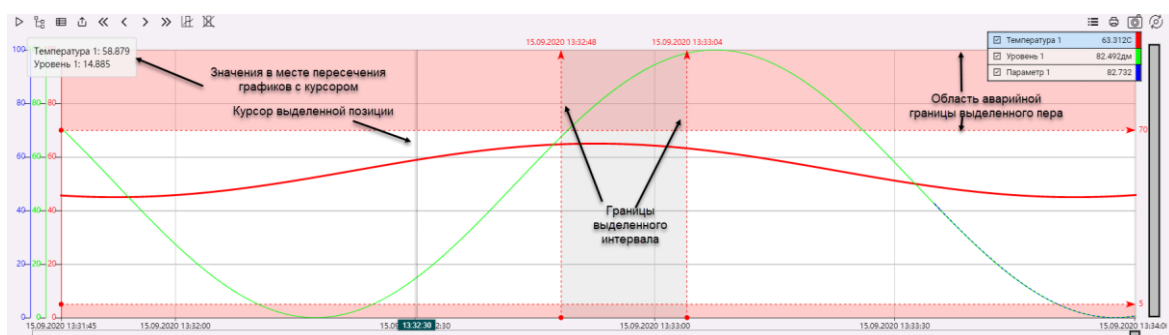
Элемент	Рекомендации
3. Курсор выбранной позиции	
4. Ось Y	
5. Ось X	

- область отображения кривых, в которой располагаются следующие элементы:
- собственно кривые;
- сетка;
- курсор текущего времени;
- курсор выбранной позиции;
- ось времени (X);
- ось значений (Y).

8.3.2.1.2.2. ТРЕНД В КЛИЕНТЕ ВИЗУАЛИЗАЦИИ ПРИ НМІ V2

При открытии окна, которое содержит тренд, клиент визуализации шлет запрос к исполнительной системе на получение архивных данных связанных с перьями параметров. Исполнительная система предоставляет данные для тренда для параметров, которые архивируются из базы данных или из оперативной памяти (если данные еще не были переданы в базу данных). После запроса из архива данные некоторое время хранятся в кеше клиента визуализации, на случай повторных запросов (для оптимизации работы с архивом). Для параметров, которые не архивируются, данные накапливаются непосредственно в клиенте визуализации и могут быть доступны только пока открыто окно с трендом, в течении непродолжительного времени.

Возможный внешний вид тренда в клиенте визуализации:




Панель инструментов тренда

В верхней части тренда находится панель инструментов. Назначение иконок панели инструментов:

Элемент	Название	Назначение
Левая сторона панели инструментов		
	Остановить автопрокрутку/Автопрокрутка	Если автопрокрутка включена (запущена), то отображаются последние значения. Если автопрокрутка выключена (остановлена), то пользователь самостоятельно выбирает за какой отрезок времени будут отображаться графики.
	Дерево объекта	Открывает окно, в котором в виде дерева отображаются архивируемые параметры.
	Таблица	Открывает окно для отображения значений в табличном виде.
	Экспорт	Сохраняет данные, отображаемые на тренде в табличном виде в формате csv.
	Первая страница	Открывает самую первую страницу тренда, с самыми ранними значениями, которые имеются в базе данных, при этом текущий интервал отображаемых данных остается неизменным, но меняется конец отображаемых данных. При нажатии на данную иконку автопрокрутка будет отключена.
	Предыдущая страница	Открывает предыдущую страницу тренда. При этом текущий интервал отображаемых данных остается неизменным, но меняется конец отображаемых данных. При нажатии на данную иконку автопрокрутка будет отключена.
	Следующая страница	Открывает предыдущую страницу тренда. При этом текущий интервал отображаемых данных остается неизменным, но меняется конец отображаемых

Элемент	Название	Назначение
		данных . При нажатии на данную иконку автопрокрутка будет отключена.
	Последняя страница	Открывает самую последнюю страницу тренда, с самыми поздними значениями, которые имеются в базе данных. При этом текущий интервал отображаемых данных остается неизменным, но меняется конец отображаемых данных. При нажатии на данную иконку автопрокрутка будет отключена.
	Установить/Зафиксировать	Позволяет выделить область на тренде
	Сбросить	Убирает выделение области на тренде
Правая сторона панели инструментов		
	Показать минилегенду	Определяет видимость минилегенды
	Печать	Запускает механизм печати графика
	Экспорт графика	Позволяет сохранить тренд в виде графического файла
	Сброс	Отменяет настройки внешнего вида тренда, сделанные пользователем в клиенте визуализации.

Минилегенда тренда

Под панелью инструментов справа может отображаться полупрозрачная минилегенда. Ее видимость определяется соответствующим свойством тренда. Самый простой способ включить и выключить минилегенду - нажать на иконку  в панели инструментов.

Вид минилегенды:

<input checked="" type="checkbox"/> Уровень 1	3.458дм	Red
<input checked="" type="checkbox"/> Температура 1	60.426С	Green
<input checked="" type="checkbox"/> Параметр 1	30.000	Blue
<input checked="" type="checkbox"/> Параметр 3	60.000	Cyan

Каждому перу соответствует строка в минилегенде. Устанавливая и снимая флаг рядом с именем пера, можно управлять видимостью перьев на тренде (значения из архива будут поступать независимо от видимости пера на тренде). В минилегенде также отображается текущее значение пера, единицы измерения и цвет, которым он рисуется на тренде. В минилегенде можно выделить перо, кликнув по его строке - выделенное перо отмечается синей подсветкой в минилегенде

Ось Y

Слева находятся оси Y (ось значений). Количество осей зависит от количества перьев и их настроек. Если в настройках тренда у свойства Автоматически создавать оси значений снят флаг, то будет одна ось, которая отображает диапазон выделенного в данный момент пера. Если флаг установлен, то для каждого пера, у которого в свойстве Отображать ось значений отмечен флаг отобразится отдельная ось значений.

Цвет оси значений совпадает с цветом пера.

Другие параметры оси Y можно задать в настройках тренда.

В зависимости от количества символов используемых в шкале положения осей могут смещаться левее/правее, таким образом, чтобы были видны все значения шкалы.

Ось X

Ось X (ось времени) находится под полем графиков, на ней отмечены время начала и окончания отображаемого интервала и несколько промежуточных точек. Внешний вид оси X можно определить в настройках тренда.

Поле графиков

Перья рисуются в центральной части элемента. Внешний вид перьев и правила их отображения настраиваются в свойствах тренда.

Выделенное перо на тренде рисуется более жирной линией (перо можно выделить в минилегенде).

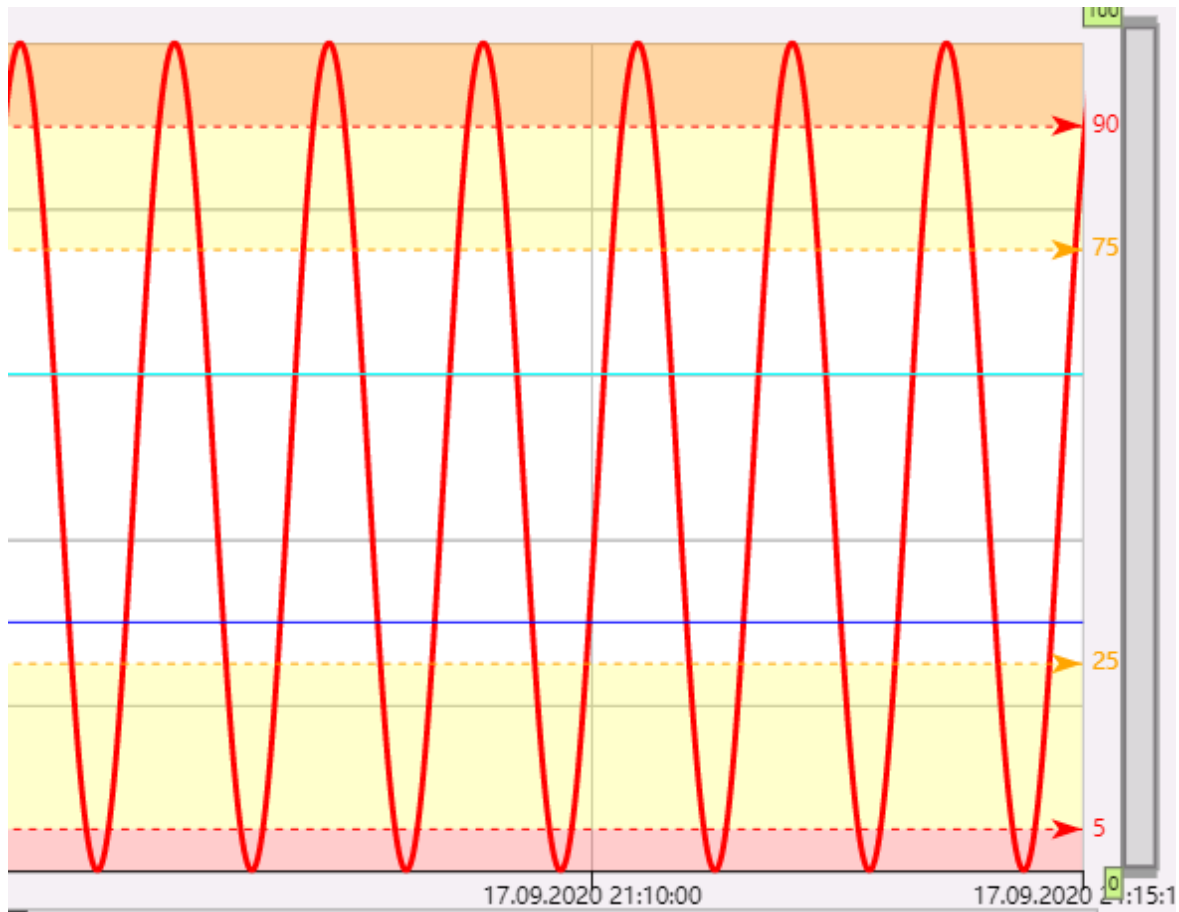
Во время выполнения загрузки архивных данных в нижнем правом углу поля графика появляется индикатор подгрузки. Как только с сервера придут все данные, индикатор пропадает. При смене интервала или добавлении перьев, когда возникает новый запрос к архиву, индикатор появляется снова. Он возникает при задержке более чем 1 секунда.

Отображение предупредительных и аварийных границ

Если для связанного с пером параметра назначена шкала, и такое перо становится выделенным, то в поле графиков отрисовываются его границы:

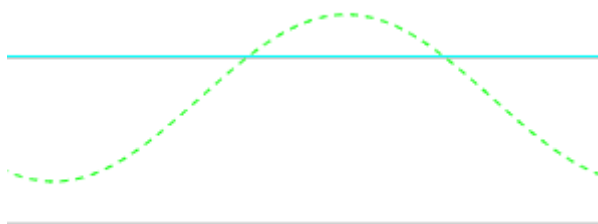
- красным цветом - область аварийных границ
- желтым цветом - область предупредительных границ

С правой стороны поля графиков указывается значение границ:



Отображение недостоверных данных

Если приходят недостоверные данные (признак качества параметров типа SYSTEM_*_PARAM не равен значению GOOD), то на тренде такие параметры будут отображаться пунктирной линией:



Если исполнительная система выключена (признак качества параметров в том случае будет `OpcUa_GoodShutdownEvent 0x00A80000`), то на графике будет разрыв.

Горизонтальный ползунок интервала

Находится под осью X, служит для определения интервала и конца отображаемой области в поле графика. Порядок работы с элементом в клиенте визуализации описан в разделе, посвященном одноименному независимому графическому элементу.

Вертикальный ползунок интервала

Находится слева от поля графиков. Служит для масштабирования графиков и для смещения графиков по оси значений. Порядок работы с элементом в клиенте визуализации описан в разделе, посвященном независимому графическому элементу Горизонтальный ползунок интервала.

Курсор выделенной позиции

Курсор выделенной позиции - это вертикальная линия, пересекающая всё поле графиков, появляется при наведении мышью на поле графиков. Под курсором отображается значение времени в месте установки курсора. Слева под панелью инструментов появляется область, в которой отображаются значения перьев в точке установки курсора. Внешний вид курсора выделенной позиции указывается в настройках тренда.

Минимальный вид тренда

Вид тренда, если в настройках тренда установлен флаг в свойстве Минимальный вид:

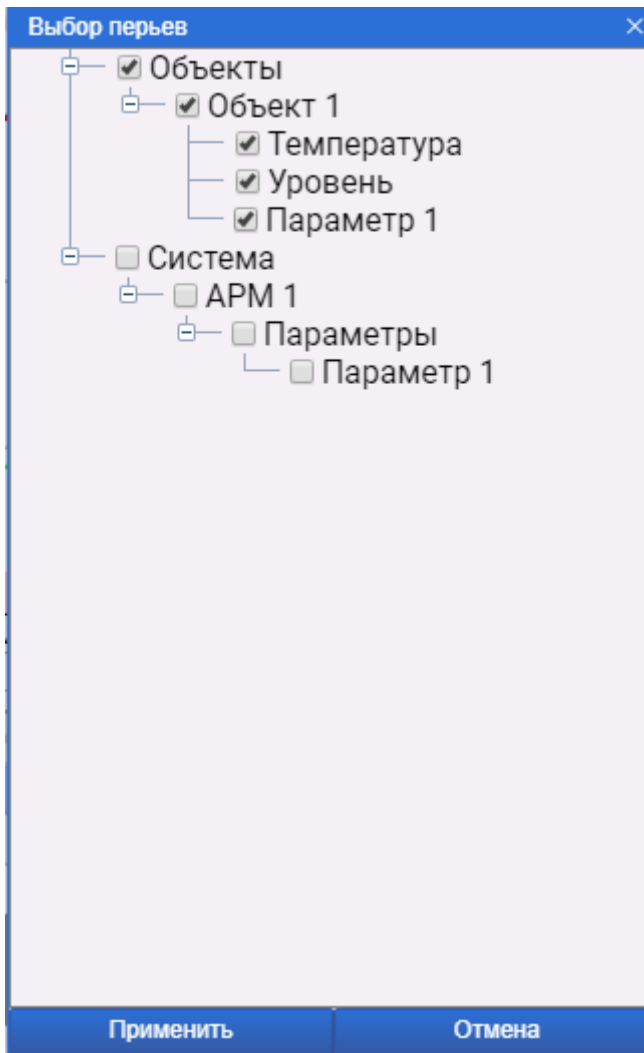
Рекомендации по использованию тренда

Тренд часто используют совместно с графическим элементом Легенда

ДОБАВЛЕНИЕ/УДАЛЕНИЕ ПЕРЬЕВ ТРЕНДА В КЛИЕНТЕ ВИЗУАЛИЗАЦИИ

Возможность добавлять и удалять перья тренда в клиенте визуализации определяет разработчик проекта настройкой Дерево выбора перьев. Если флаг установлен, то в панели инструментов тренда будет иконка Дерево объекта.

Для добавления нового пера на тренд необходимо нажать на иконку Дерево Объекта. Откроется диалоговое окно, в котором содержатся в виде дерева все архивируемые параметры, которые находятся в исполнительной системе, к которой подключен клиент визуализации:




Если свойство Титул задано в настройках, то заголовок окна выбора перьев будет иметь вид: Выбор перьев - [Титул]

Флагами отмечены параметры, которые связаны с перьями тренда. Устанавливая и снимая флаги можно изменить состав перьев на тренде. На тренд может быть добавлено перьев не больше чем указано в настройке Максимальное количество перьев

Для добавления/удаления перьев у пользователя должны быть настроены соответствующие права доступа

Также возможно добавлять перья на тренд в режиме исполнения используя методы тренда

ТАБЛИЧНОЕ ПРЕДСТАВЛЕНИЕ ДАННЫХ ТРЕНДА

Чтобы открыть таблицу значений перьев необходимо нажать на иконку в панели инструментов . Тогда в левой части графика откроется таблица:


Время	Уровень 1	Температура 1	Параметр 1	Параметр 3
16.09.2020 11:46:12	-	-	30.000	60.000
16.09.2020 12:30:09	86.545	53.412	30.000	60.000
17.09.2020 12:06:47	50.000	55.000	30.000	60.000
17.09.2020 12:06:48	53.146	55.524	30.000	60.000
17.09.2020 12:06:49	56.273	56.046	30.000	60.000
17.09.2020 12:06:50	59.378	56.566	30.000	60.000
17.09.2020 12:06:51	62.447	57.081	30.000	60.000
17.09.2020 12:06:52	65.466	57.591	30.000	60.000
17.09.2020 12:06:53	68.421	58.093	30.000	60.000
17.09.2020 12:06:54	71.306	58.587	30.000	60.000
17.09.2020 12:06:55	74.110	59.071	30.000	60.000
17.09.2020 12:06:56	76.813	59.544	30.000	60.000
17.09.2020 12:06:57	79.412	60.004	30.000	60.000
17.09.2020 12:06:58	81.895	60.451	30.000	60.000

Ширина таблицы будет зависеть от количества перьев.

В левом столбце отображается время изменения данных. Если одно перо изменилось в одно время, а другое перо в другое время, то в таблице сформируется две строки: для каждого времени изменения. Одно значение точное полученное в этот момент, а другое будет рассчитываться. Значение параметра, которое не менялось будет зависеть от настройки пера Тип графика. Если установлено значение Ступенька, то будет предыдущее значение, если Линия, то будет рассчитано в зависимости от полученных данных.

В таблице будут показаны значения за интервал времени, который отображается на тренде.

При необходимости данные таблицы можно экспортировать.


Закрывать таблицу можно повторно нажав на иконку 

ЭКСПОРТ И ПЕЧАТЬ ТРЕНДА

Экспорт тренда

Тренд можно экспортировать в CSV, либо сохранить как картинку.


Экспорт тренда в CSV

Для экспорта тренда в табличном виде в формате csv необходимо нажать на иконку в панели инструментов 


Откроется диалоговое окно, в котором необходимо выбрать имя сохраняемого файла и путь к нему. Если недопустимо открытие подобных окон, то можно указать место сохранения файлов в приложении MasterSCADA 4D Client Monitor (если клиент запускается на устройстве отличном от исполнительной системы) или в приложении MasterSCADA 4D Monitor в поле Параметры запуска клиента.

Кодировка сохраняемого файла зависит от настройки Шаблона экрана Кодировка сохраняемых csv файлов

Экспорт тренда в графический файл

Для сохранения текущего состояния тренда в графический файл необходимо нажать на иконку  в панели инструментов. Откроется диалоговое окно, в котором необходимо выбрать имя сохраняемого файла и путь к нему. Если недопустимо открытие подобных окон, то можно указать место сохранения файлов в приложении MasterSCADA 4D Client Monitor (если клиент запускается на устройстве отличном от исполнительной системы) или в приложении MasterSCADA 4D Monitor в поле Параметры запуска клиента.

Печать тренда

Для начала печати тренда необходимо нажать на иконку панели инструментов . Откроется диалоговое окно, в котором необходимо выбрать принтер для печати. Если недопустимо открытие подобных окон, то можно указать принтер в приложении MasterSCADA 4D Client Monitor (если клиент запускается на устройстве отличном от исполнительной системы) или в приложении MasterSCADA 4D Monitor в поле Параметры запуска клиента.

При печати тренда или экспорте сформируется сообщение в журнал:

Печать графика. Текст "Печать графика на принтер <имя принтера>"

Экспорт графика. Текст "Сохранение графика в файл <имя файла с путем>"

Также можно для выполнения экспорта и печати использовать методы для работы с трендом

СМЕЩЕНИЕ ГРАФИКОВ ПО ОСИ X

Смещение графиков по оси X

Просматривать графики, смещая их по оси времени влево и вправо, можно при помощи следующих инструментов:

- нажимая иконки в панели инструментов
- используя мышь
- используя горизонтальный ползунок интервала, который находится под осью X
- используя стандартные жесты (для touchscreen)
- используя методы для работы с трендом

В любом случае при смещении графиков автопрокрутка будет отключена.

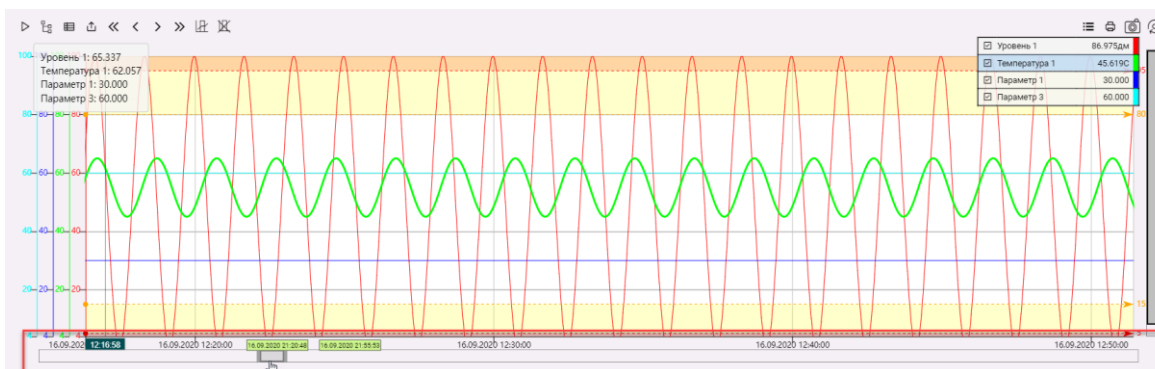
Ознакомиться с иконками в панели инструментов можно в таблице выше.

Смещение графика при помощи мыши

Перемещение мыши в поле графика с нажатой левой клавишей приведет к плавному смещению графиков. При этом текущий интервал отображаемых данных останется неизменным, но будет изменяться конец отображаемых данных .

Смещение графика при помощи горизонтального ползунка интервала

Для перемещения ползунка на него нужно нажать левой кнопкой мыши и передвигать мышь влево/вправо. При перемещении ползунка будет отображаться начало и окончание времени, которое отобразится на тренде.



Тренд перерисовывается только после завершения перемещения ползунка, когда пользователь отпустит левую кнопку мыши.

ИЗМЕНЕНИЕ ОТОБРАЖАЕМОГО ИНТЕРВАЛА

Изменить интервал отображаемых данных (изменить масштаб по оси времени) можно при помощи следующих инструментов

- используя горизонтальный ползунок интервала, который находится под осью X
- используя мышь
- используя стандартные жесты (для touchscreen)

Изменение отображаемого интервала при помощи горизонтального ползунка интервала

Для изменения отображаемого интервала необходимо потянуть левой кнопкой мыши за границы ползунка (при этом индикатор мыши примет форму стрелочки). Тренд перерисовывается только после окончания изменения границы ползунка

Изменение отображаемого интервала при помощи мыши

Вращение колесика мыши в области графиков приведет к плавному изменению интервала:

- вращение вверх - уменьшит интервал, т.е. график, можно будет изучить более детально, в каждый момент времени
- вращение вниз - увеличит интервал, т.е. отобразится график за более длительное время на экране.


ВЫДЕЛЕНИЕ ОБЛАСТИ НА ТРЕНДЕ

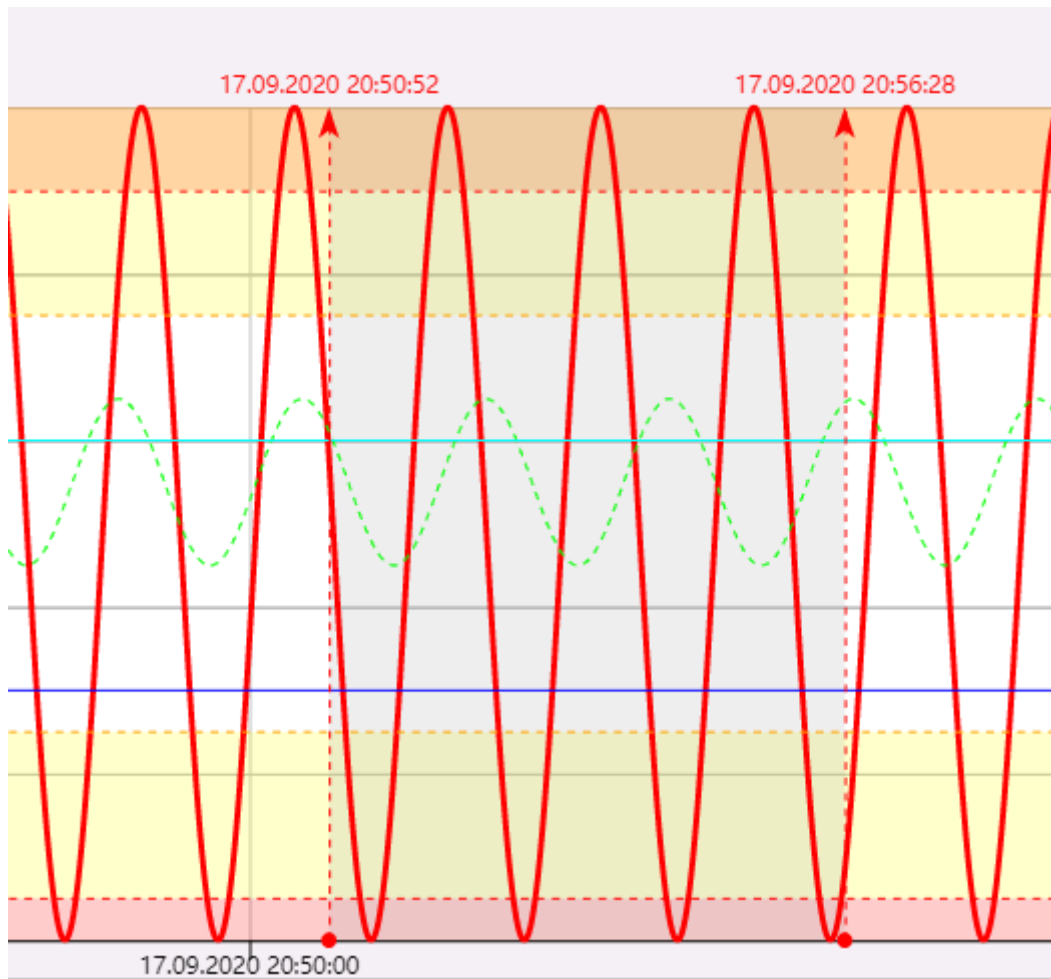
Для выделения области на тренде необходимо в панели инструментов нажать на иконку




Затем левой кнопкой мыши в области графиков выделить нужный интервал:



Затем повторно нажать на иконку  - выделенная область будет зафиксирована. Вид зафиксированной области:



Появятся вертикальные стрелки с указанием, времени начала и временем окончания интервала. Если к тренду подключена Легенда, то в ней отобразится дополнительная информация по этому интервалу, например, максимальные и минимальные значения в этом промежутке времени. и др.

Может быть выбран только один выделенный интервал на тренде. Для снятия выделенного интервала необходимо нажать на иконку в панели инструментов .

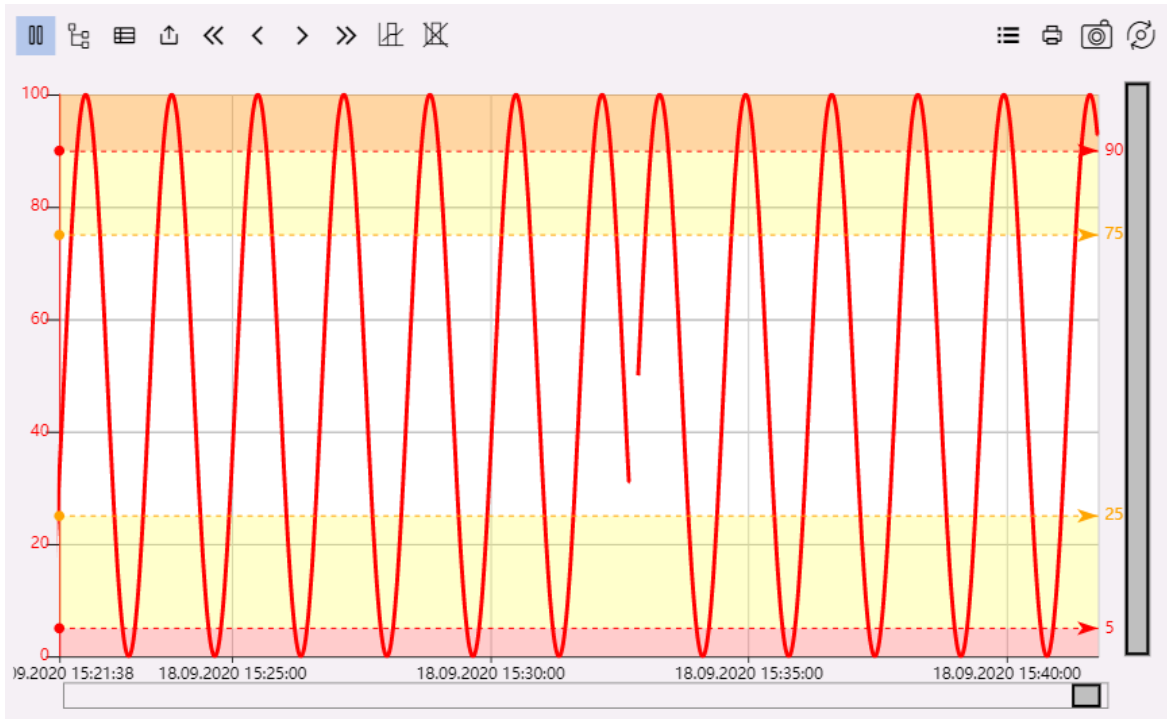
Также можно для выделения области использовать методы для работы с трендом

МАСШТАБИРОВАНИЕ ПО ОСИ ЗНАЧЕНИЙ

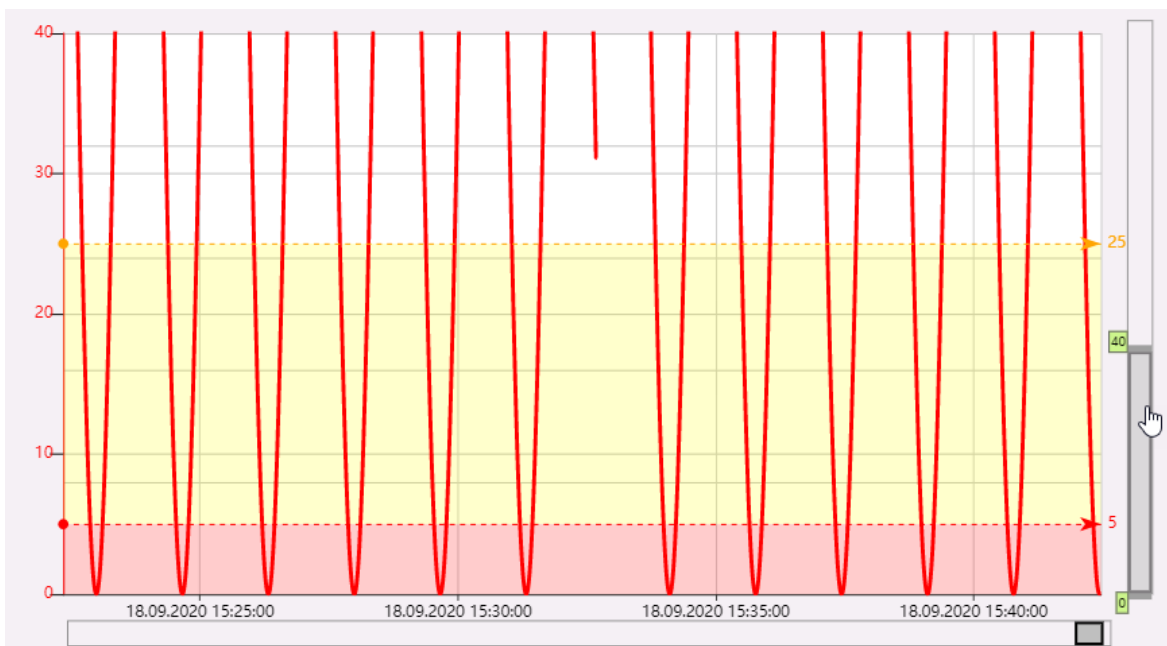
Изменение видимой части графиков по оси Y (масштабирование по оси значений) производится при помощи вертикального ползунка интервала, который находится справа от поля графиков.

Если в настройках тренда в свойстве Масштабировать по всем осям установлен флаг, то при работе с вертикальным ползунком будут изменяться все перья тренда. Если флаг снят, то масштабирование будет производиться только выделенного пера.

По умолчанию для всех перьев отображается вся шкала. Вид ползунка в этот момент:

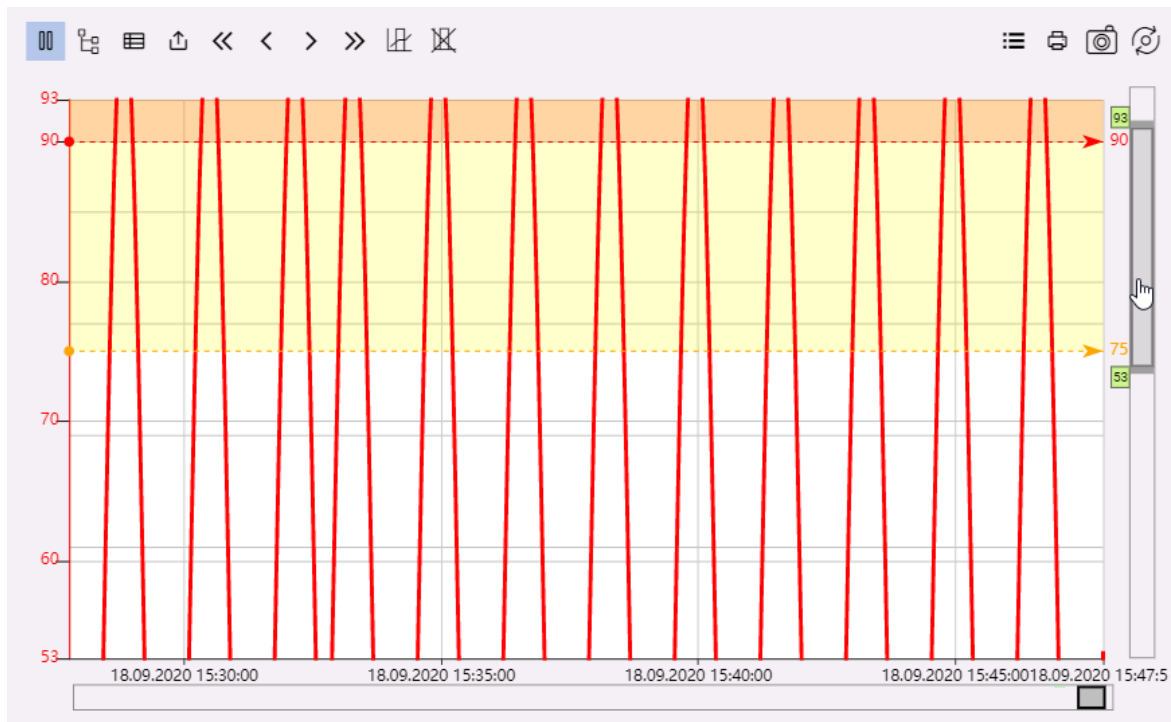


Потянув за верхнюю границу ползунка вниз или за нижнюю границу вверх изменятся значения по оси Y. График можно будет рассмотреть более детально:



Видно, что интервал отображаемый в ползунке совпадает с максимальным и минимальным значением оси Y.

Теперь, если переместить ползунок вверх, то будет возможность изменить отображаемый диапазон:



8.3.2.1.3. ТРЕНД. ПРАВА ДОСТУПА

Для работы с трендом в режиме исполнения предусмотрены специальные права доступа.

В дереве системы в группе Безопасность настраиваются умолчания для всех добавляемых в проект Трендов. У каждого отдельного Тренда в его панели свойств можно изменить сделанные умолчания.

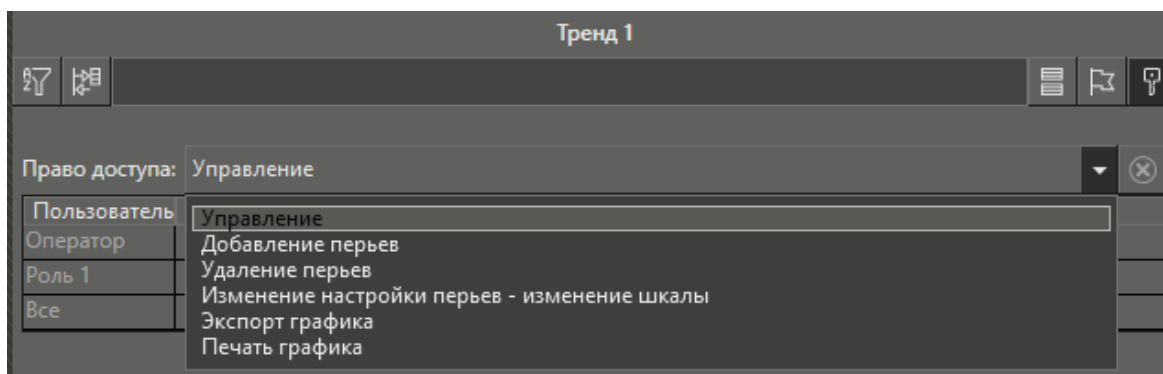
Вид панели свойств Тренда в режиме редактирования Прав:

Тренд 1

Право доступа: Управление

Пользователь	Разрешить	Запретить	Подтверждение	Журнал
Оператор	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Роль 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Все	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Доступные для настройки права доступа:

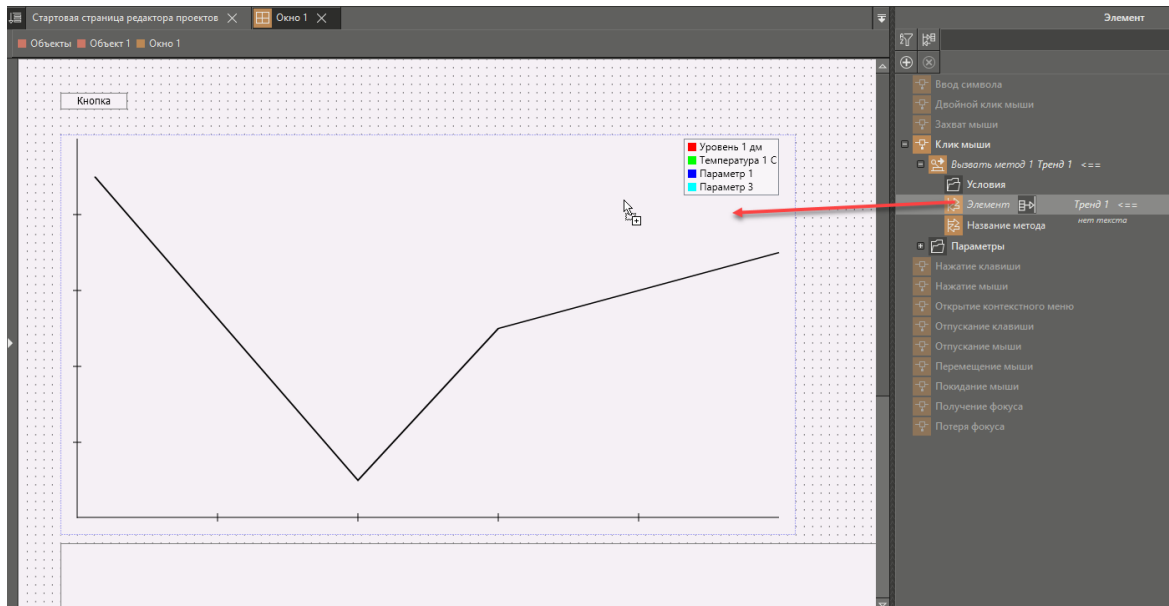


Право	Назначение
Управление	Определяет возможность пользователя работать с трендом в режиме исполнения. Если право не предоставлено, то тренд будет полностью отключен (полностью недоступен для управления). Пользователь не сможет воспользоваться панелью инструментов, а также изменить масштаб или интервал отображаемых данных при помощи мыши. Как правило, это право пользователям предоставляется. Даже если на другие графические элементы это право не распространяется.
Добавление перьев	Определяет возможность добавлять новые перья на тренд в клиенте визуализации.
Удаление перьев	Определяет возможность удалять перья с поля графика.
Изменение настройки перьев - изменение шкалы	Определяет возможность пользователя менять значения нижней и верхней границы отображения графика по оси Y
Экспорт графика	Определяет возможность экспорта тренда в табличном виде или в графическом. Если место сохранения файлов указано в приложении MasterSCADA 4D Client Monitor (если клиент запускается на устройстве отличном от исполнительной системы) или в приложении MasterSCADA 4D Monitor в поле Параметры запуска клиента, и установлен флаг Журнал, то в журнале сформируется сообщение Экспорт графика. Текст "Сохранение графика в файл <имя файла и путь>"

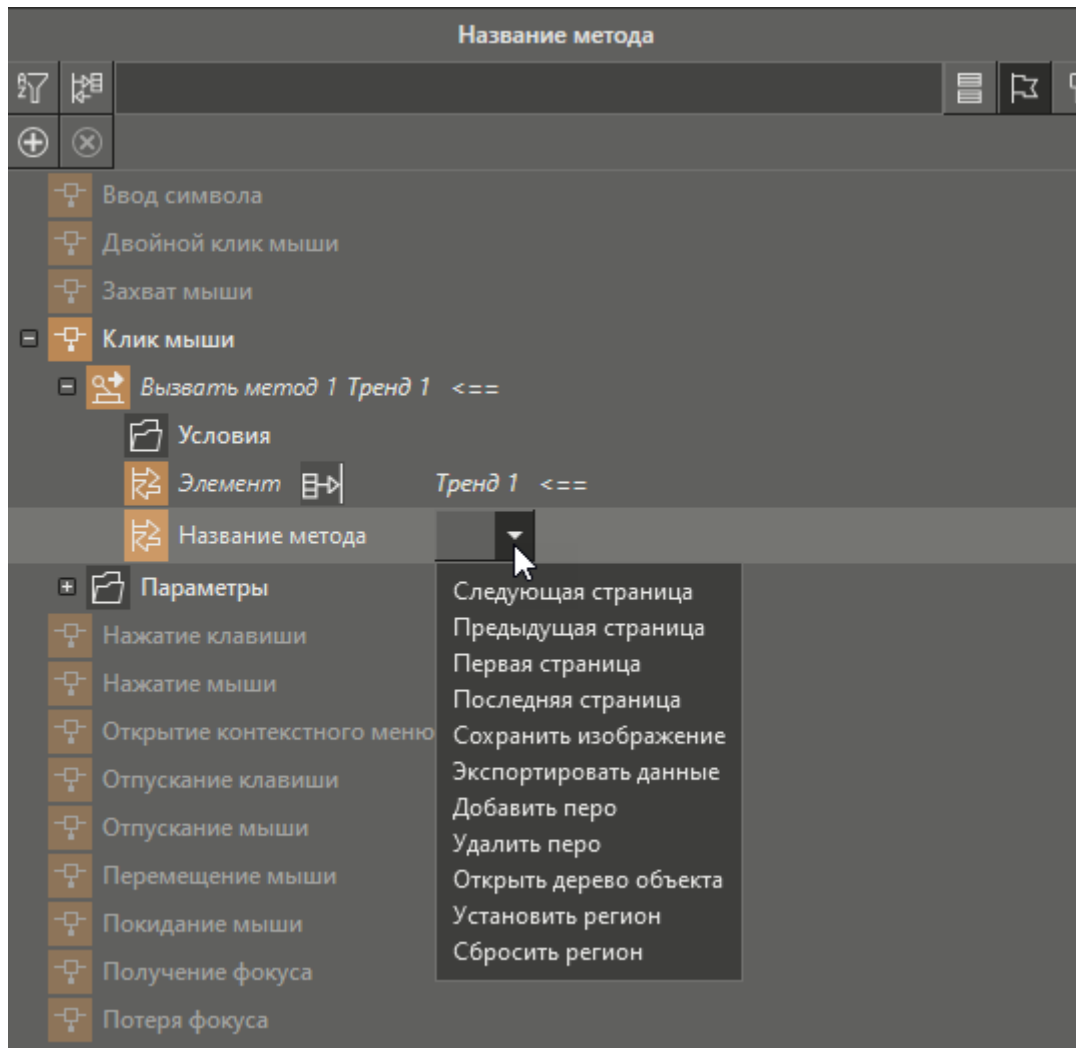
Право	Назначение
Печать графика	<p>Определяет возможность пользователю печатать тренд в клиенте визуализации, используя методы и кнопки панели инструментов. Если принтер указан в приложении MasterSCADA 4D Client Monitor (если клиент запускается на устройстве отличном от исполнительной системы) или в приложении MasterSCADA 4D Monitor в поле Параметры запуска клиента, и установлен флаг Журнал то в журнале сформируется сообщение Печать графика. Текст "Печать графика на принтер <имя принтера>"</p>

8.3.2.1.4. МЕТОДЫ ДЛЯ РАБОТЫ С ТРЕНДОМ

Для работы с трендом можно создать собственные элементы управления, например кнопки, и настроить события, например, в событие Клик мыши добавить действие Вызвать метод, связать параметр Элемент с трендом (для этого можно потянуть параметр метода на тренд.):



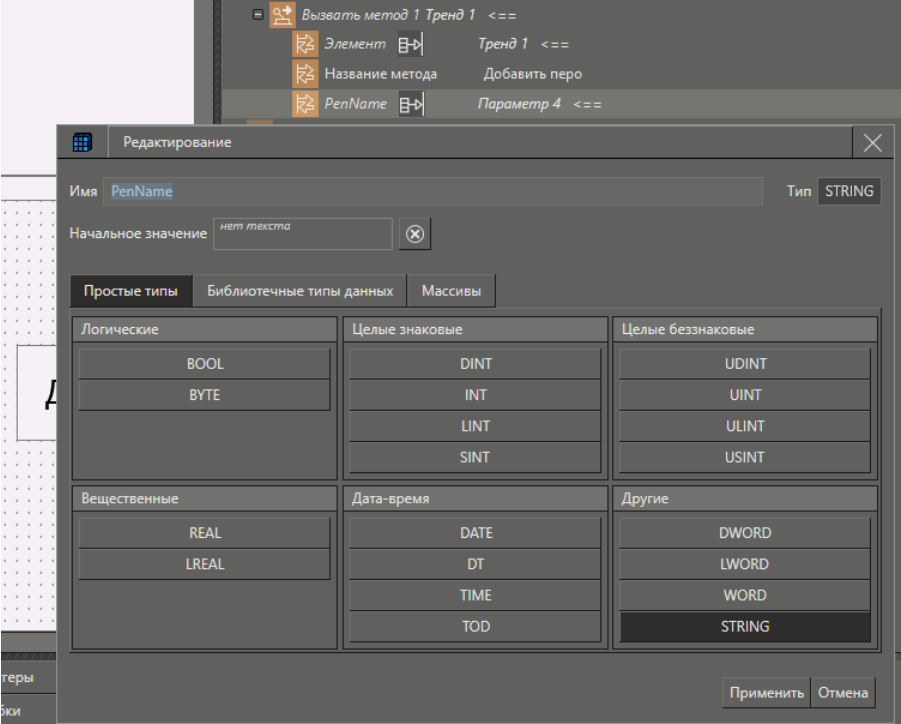
Затем указать какое действие с трендом должно выполняться при нажатии на элемент управления:



Возможные методы:

Название	Описание
Следующая страница	Переходит на следующую страницу журнала.
Предыдущая страница	Переходит на предыдущую страницу журнала.
Первая страница	Открывает первую страницу журнала, т.е. самые ранние сообщения в архивном журнале, либо самые ранние сообщения в журнале.
Последняя страница	Открывает последнюю страницу журнала.

Название	Описание
Сохранить изображение	<p>Заменяет иконку панели инструментов тренда Экспорт графика. Сохраняет текущее изображение тренда в графический файл. В результате работы метода откроется диалоговое окно, в котором необходимо выбрать имя сохраняемого файла и путь к нему. Если недопустимо открытие подобных окон, то можно указать место сохранения файлов в приложении MasterSCADA 4D Client Monitor (если клиент запускается на устройстве отличном от исполнительной системы) или в приложении MasterSCADA 4D Monitor в поле Параметры запуска клиента.</p>
Экспортировать данные	<p>Заменяет иконку панели инструментов тренда Экспорт. Сохраняет данные в csv. В результате работы метода откроется диалоговое окно, в котором необходимо выбрать имя сохраняемого файла и путь к нему. Если недопустимо открытие подобных окон, то можно указать место сохранения файлов в приложении MasterSCADA 4D Client Monitor (если клиент запускается на устройстве отличном от исполнительной системы) или в приложении MasterSCADA 4D Monitor в поле Параметры запуска клиента. Кодировка сохраняемого файла зависит от настройки Шаблона экрана Кодировка сохраняемых csv файлов</p>
Добавить перо	<p>Позволяет добавить перо на тренд в клиенте визуализации. Для этого в метод нужно добавить Аргумент (через контекстное меню метода). Указать его имя - PenName, и тип данных - STRING. Этот аргумент нужно связать с параметром, через который будет подаваться полное имя параметра, который нужно отобразить на тренде.</p>

Название	Описание
	 <p>При добавлении цвет перу присваивается из пула различных цветов. Пул рассчитан на 20 цветов.</p> <p>Важно! При обновлении страницы клиента визуализации все добавленные/удаленные перья сбрасываются, остаются только перья, заданные в проекте.</p>
Удалить перо	<p>Позволяет удалить перо с тренда в клиенте визуализации. Для этого в метод нужно добавить Аргумент. Указать его имя - PenName, и тип данных - STRING. Этот аргумент нужно связать с параметром, через который будет подаваться полное имя параметра, который нужно удалить с тренда. Если на Тренде есть перья с связанные с одним и тем же параметром, то удалятся все перья, имеющие одинаковую связь.</p> <p>Важно! При обновлении страницы клиента визуализации все добавленные/удаленные перья сбрасываются, остаются только перья, заданные в проекте.</p>
Открыть дерево объекта	<p>Заменяет иконку в панели инструментов Дерево объекта. Открывает окно, в котором находятся архивируемые параметры проекта. Устанавливая и снимая флаги можно изменить список отображаемых перьев на тренде.</p>

Название	Описание
Установить регион	Заменяет иконку в панели инструментов Установить/Зафиксировать.
Сбросить регион	Заменяет иконку в панели инструментов Сбросить.

8.3.2.2. ДИСКРЕТНЫЙ ТРЕНД

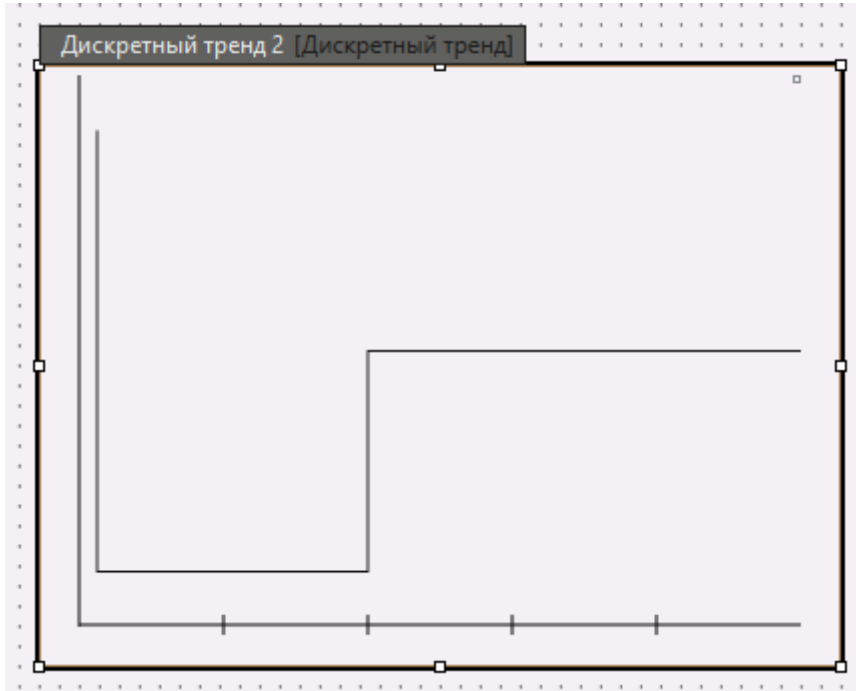
Графический элемент Дискретный тренд предназначен для отображения изменения значения параметра типов BOOL и Перечисление во времени. Данный элемент может работать как с архивными, так и с текущими значениями. При добавлении перьев в режиме исполнения в окне выбора перьев будут доступны только параметры данных типов.

В палитре редактора HMI элемент Дискретный тренд находится в категории Контролы.

Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:



Поддерживается только при использовании типа транслятора HMI v2.

Внешний вид дискретного тренда может быть изменен при помощи панели свойств. Любое свойство может быть задинамизировано, таким образом, разработчик проекта может создать те элементы управления трендом, которые необходимы пользователю.

Настройки внешнего вида будут применены в клиенте визуализации в режиме исполнения. Для того, чтобы оценить изменения, не загружая конфигурацию в узел, можно воспользоваться кнопкой Просмотр панели инструментов

Добавление на тренд параметра типа BOOL

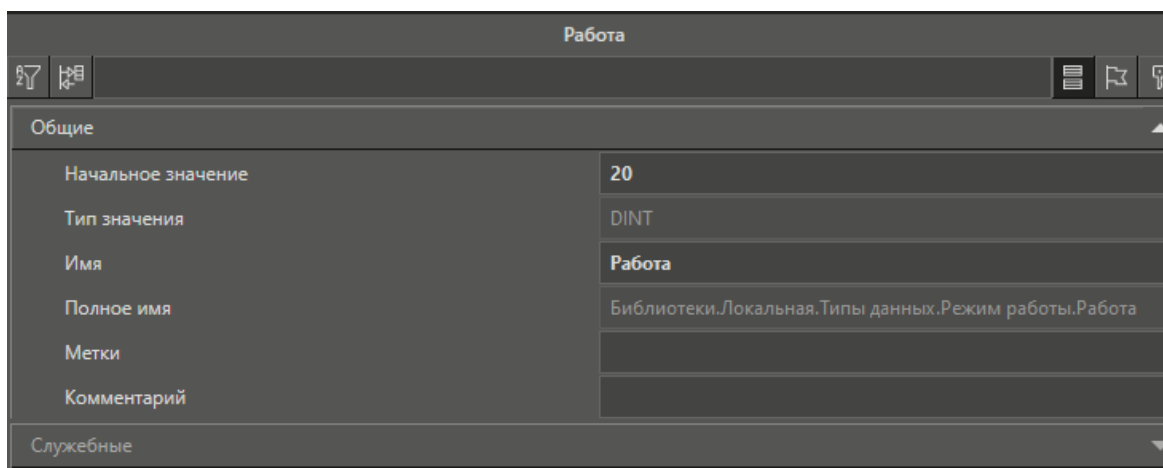
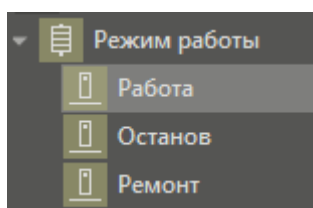
Для того чтобы добавить параметры, которые должны отображаться на дискретном тренде, необходимо перетащить параметр из дерева в поле дискретного тренда. При этом добавится перо. Кроме того, изменится внешний вид элемента: появится минилегенда в верхнем левом углу элемента. Одновременно с этим в панели свойств добавится группа свойств, относящаяся к добавленному перу.

Важно! На Дискретный тренд можно добавить несколько выделенных параметров одновременно.

Если параметр, связанный с пером, будет удален, или если будет разорвана связь между параметром и пером в контекстном меню параметра, то перо при этом останется в дискретном тренде, а все настройки, например, цвет, тип линии сохранятся. Для удаления созданного пера следует воспользоваться панелью свойств элемента.

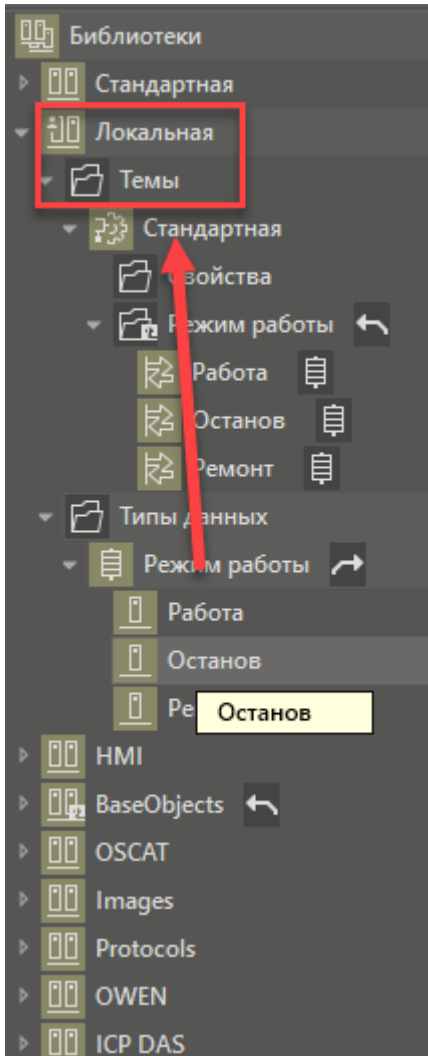
Добавление на тренд параметра типа Перечисление

Пусть есть параметр тип данных Перечисление вида:

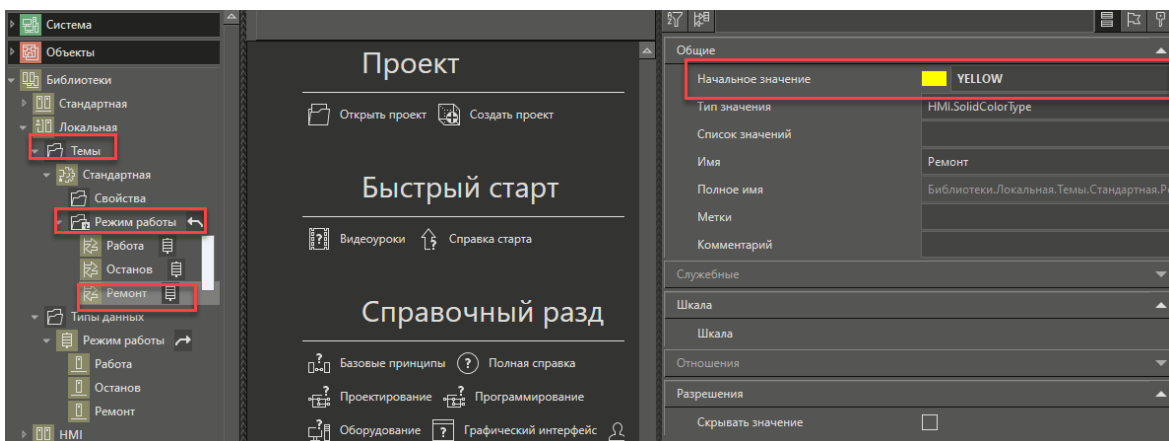


Далее, чтобы иметь возможность отображать перечислимое на дискретном тренде необходимо в Тему, которая используется в проекте перетащить это перечислимое.

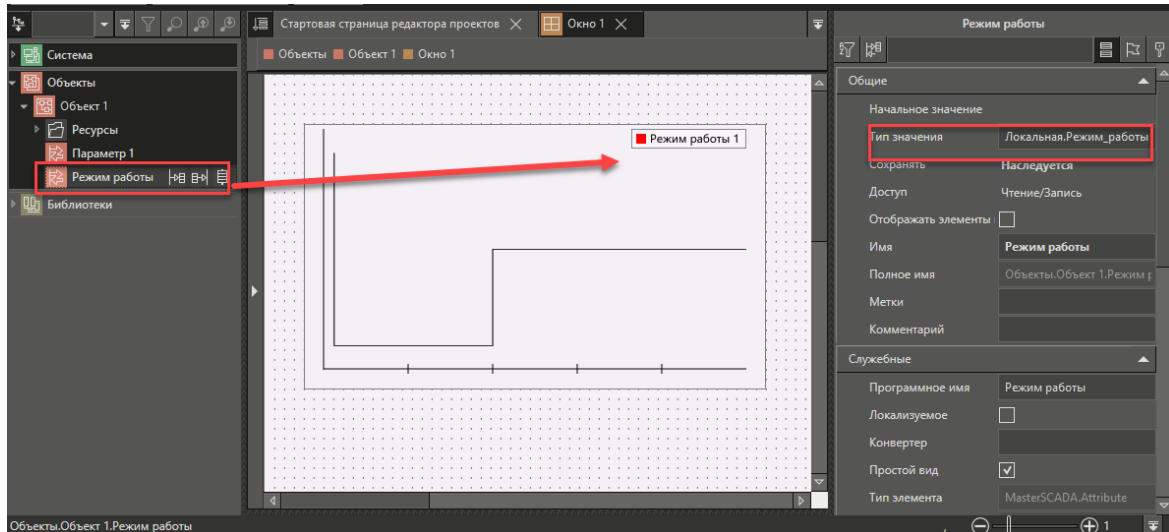
По умолчанию используется тема Стандартная. В локальной библиотеке есть возможность расширить ее настройками разработчика проекта:



Далее для каждого значения перечислимого, которое появилось в теме нужно задать цвет, которым он будет отображаться на тренде:



Если параметру назначить этот тип данных и добавить его на тренд, то появится нужное перо:

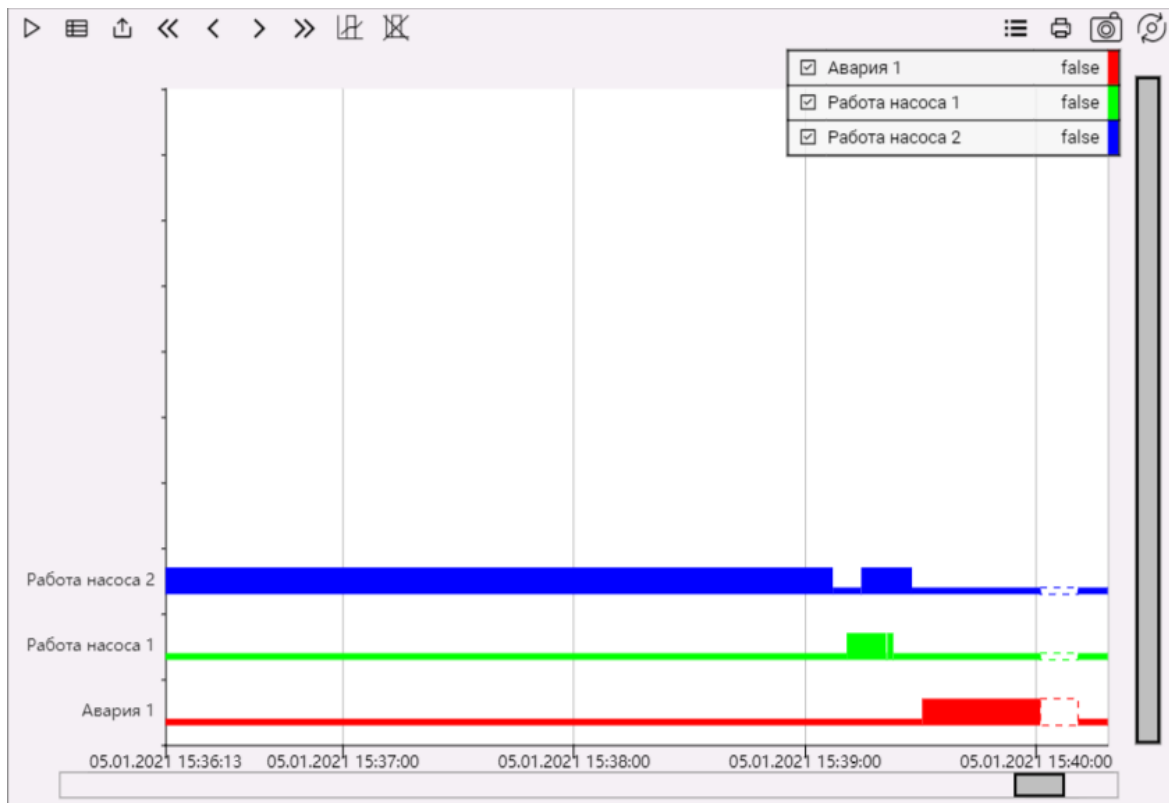


Дискретный тренд в клиенте визуализации

Вид дискретного тренда в клиенте визуализации зависит от его настроек. Каждое перо, добавленное на тренд, будет рисоваться на своем уровне, таким образом не будет перекрываться один график другим.

Параметры типа BOOL

Вид дискретного тренда с дискретными параметрами:



Высота (толщина) графика если значение равно TRUE или False определяется в настройках элемента.

Перечислимые параметры

Вид дискретного тренда с перечислимым:

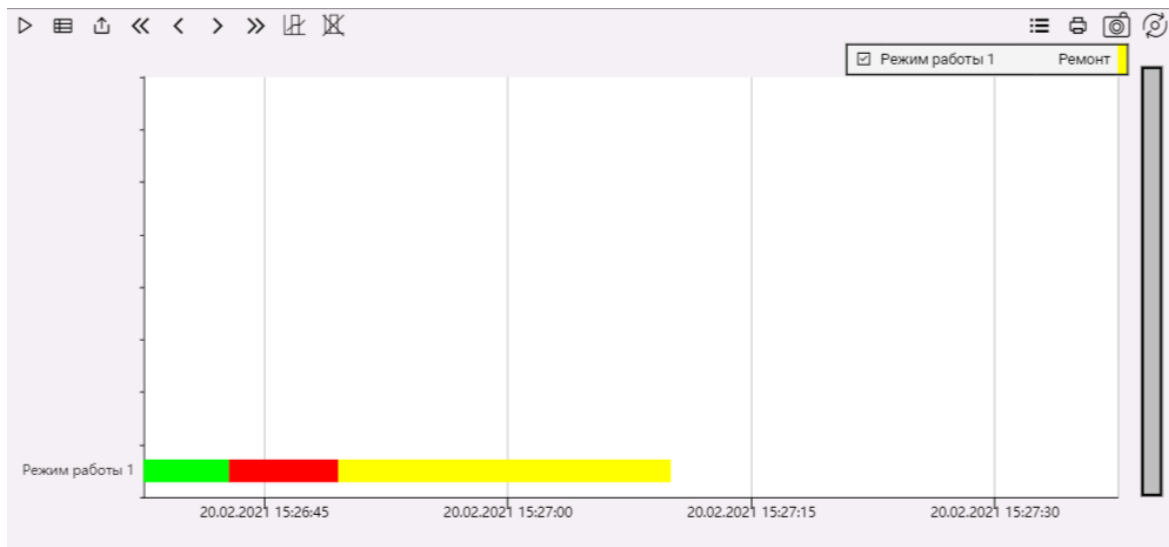


График будет менять свой цвет в зависимости от значения, которое принимает перечислимое. График будет рисоваться тем цветом, которое выбрано в теме для данного значения. В легенде будет отображаться цвет, которым в данный момент рисуется график.

Порядок работы с элементом в клиенте визуализации такой же как и с элементом Тренд:

8.3.2.2.1. СВОЙСТВА ДИСКРЕТНОГО ТРЕНДА

Вид панели свойств элемента Дискретный тренд:

Дискретный тренд 1

Общие

Служебные

График

Интервал	00:01:00
Титул	
Автопрокрутка	<input checked="" type="checkbox"/>
Период обновления данных	100
Минимальный вид	<input type="checkbox"/>
Заливка графика	<input type="checkbox"/> White
+ Перья	<input type="checkbox"/> <input type="button" value="+"/> <input type="button" value="..."/>
Конец	00:00:00
Дерево выбора перьев	<input type="checkbox"/>
Максимальное количество перьев	100
Уровень логической единицы	50
Уровень логического нуля	-50

Ось X

Ось Y

Текст

Параметры

Курсор выбранной позиции

Сетка

Архив

Внешний вид

Расположение

Рамка

Отношения

Трансформация

Клавиатура



Экран



Минилегенда

Описание основных свойств графического элемента Дискретный тренд:

Название	Описание
Категория Архив	
Начало архива	Используется, если на тренде отображается архивируемый параметр. Данное свойство имеет тип DT и отображает начало выборки из архива.
Конец архива	Используется, если на тренде отображается архивируемый параметр. Данное свойство имеет тип DT и отображает конец выборки из архива.
Категория График	
Интервал	Это свойство имеет тип TIME и задает для тренда временной интервал между крайними точками поля тренда по оси X.
Титул	Свойство определяет заголовок тренда. Если свойство задано, то заголовок окна выбора перьев будет иметь вид: Выбор перьев - [Титул]
Автопрокрутка	Данное свойство имеет тип BOOL и может принимать значение TRUE (автоматическая прокрутка отображаемого элементом тренд содержимого включена) или False (автоматическая прокрутка выключена). Если автоматическая прокрутка включена, то на тренде будут отображаться значения параметров за Интервал, предшествующий текущему моменту времени.
Период обновления данных	Это свойство имеет тип INT и задает период (в мс) обновления данных на тренде, если включена автопрокрутка.
Минимальный вид	Если данное свойство имеет значение TRUE, то на тренде скрываются все элементы, кроме поля тренда, которое, в этом случае, занимает всю площадь тренда. При этом управление трендом можно осуществлять элементами, созданными разработчиком проекта. Например, для изменения интервала можно использовать текстовый ввод, который связан с параметром, передающим данные в свойство тренда Интервал.

Название	Описание
Заливка графика	Определяет цвет заливки поля тренда. При наведении мыши на свойство появляется кнопка:  , после нажатия на которую открывается диалоговое окно Выбор цвета.
Группа Перья	После нажатия на кнопку  можно добавлять перья на тренд. Данная группа содержит подгруппы, количество подгрупп соответствует количеству перьев на тренде. Описание свойств подгрупп описано в таблице ниже.
Конец	Данное свойство имеет тип DT. Крайняя точка справа по оси X.
Дерево выбора перьев	Определяет возможность добавлять перья в клиенте визуализации. Если флаг установлен, то в панели инструментов появляется иконка вызывающая вызывающей элемент, позволяющий добавлять параметры на тренд в режиме исполнения
Максимальное количество перьев	Определяет максимальное количество перьев, которые могут располагаться на тренде одновременно.
Уровень логической единицы	Определяет толщину линии пера, если связанный параметр равен TRUE
Уровень логического нуля	Определяет толщину линии пера, если связанный параметр равен False
Категория Курсор выделенной позиции	
Цвет курсора	Определяет цвет курсора. При наведении мыши на свойство появляется кнопка:  , после нажатия на которую открывается диалоговое окно Выбор цвета.
Толщина курсора	Определяет толщину линии курсора. Задается в пикселях.
Видимость курсора	Это свойство имеет тип BOOL и задает видимость курсора выбранной позиции на тренде.

Название	Описание
Категория Ось X	
Подпись X	Подпись оси X. Если задан текст, то он отобразится в клиенте визуализации около оси X правее поля тренда.
Формат значений X	Задаёт формат значений временной оси X тренда. Например, dd.MM.yyyy HH:mm:ss:fff - в этом случае будут отображаться на тренде в виде - 25.11.2017 22:30:01:129.
Количество делений X	Задаётся требуемое количество делений оси. При построении тренда среда исполнения будет выбирать значение этого параметра по возможности близко к заданному.
Толщина делений X	Определяет внешний вид деления шкалы.
Длина делений X	Определяет внешний вид деления шкалы.
Цвет делений X	Определяет цвет делений оси. При наведении мыши на свойство появляется кнопка:  , после нажатия на которую открывается диалоговое окно Выбор цвета.
Верхний отступ делений X	Определяет расстояние от оси до значения, соответствующего тому или иному делению.
Цвет оси X	Определяет цвет оси. При наведении мыши на свойство появляется кнопка:  , после нажатия на которую открывается диалоговое окно Выбор цвета.
Толщина оси X	Определяет толщину оси.
Категория Ось Y	
Подпись Y	Подпись оси Y. Если задан текст, то он отобразится в клиенте визуализации около оси Y над полем тренда.

Название	Описание
Цвет делений Y	Определяет цвет делений оси. При наведении мыши на свойство появляется кнопка:  , после нажатия на которую открывается диалоговое окно Выбор цвета.
Толщина оси Y	Определяет толщину оси.
Толщина делений Y	Определяет внешний вид деления шкалы.
Категория Параметры	
Шаг прореживания	Данную настройку полезно использовать в случае, когда на тренде необходимо отобразить большее количество данных, чем может поместиться физически. Поскольку в одном пикселе отображается только одно значение, то не рационально считывать из архива данных больше, чем может быть использовано на тренде. Настройка определяет, на какое число пикселей должна приходиться группа из не более чем четырех значений. Для этого вычисляется интервал данных по формуле $= \text{Интервал времени видимой области графика} / (\text{Ширина графика (в px)} * \text{Шаг прореживания})$. Выборка из архива делается с тем условием, что за каждый такой интервал данных из него придет не более четырех значений (первое, последнее, минимальное, максимальное). По умолчанию Шаг прореживания равен 1.
Интервал Листания	Это свойство имеет тип LREAL и в процентах указывает, на какую часть интервала необходимо смещать окно при листании тренда вперед/назад, при использовании действия Вызвать метод.
Категория Сетка	
Цвет сетки	Определяет цвет линий сетки. При наведении мыши на свойство появляется кнопка:  , после нажатия на которую открывается диалоговое окно Выбор цвета.
Категория Минилегенда	

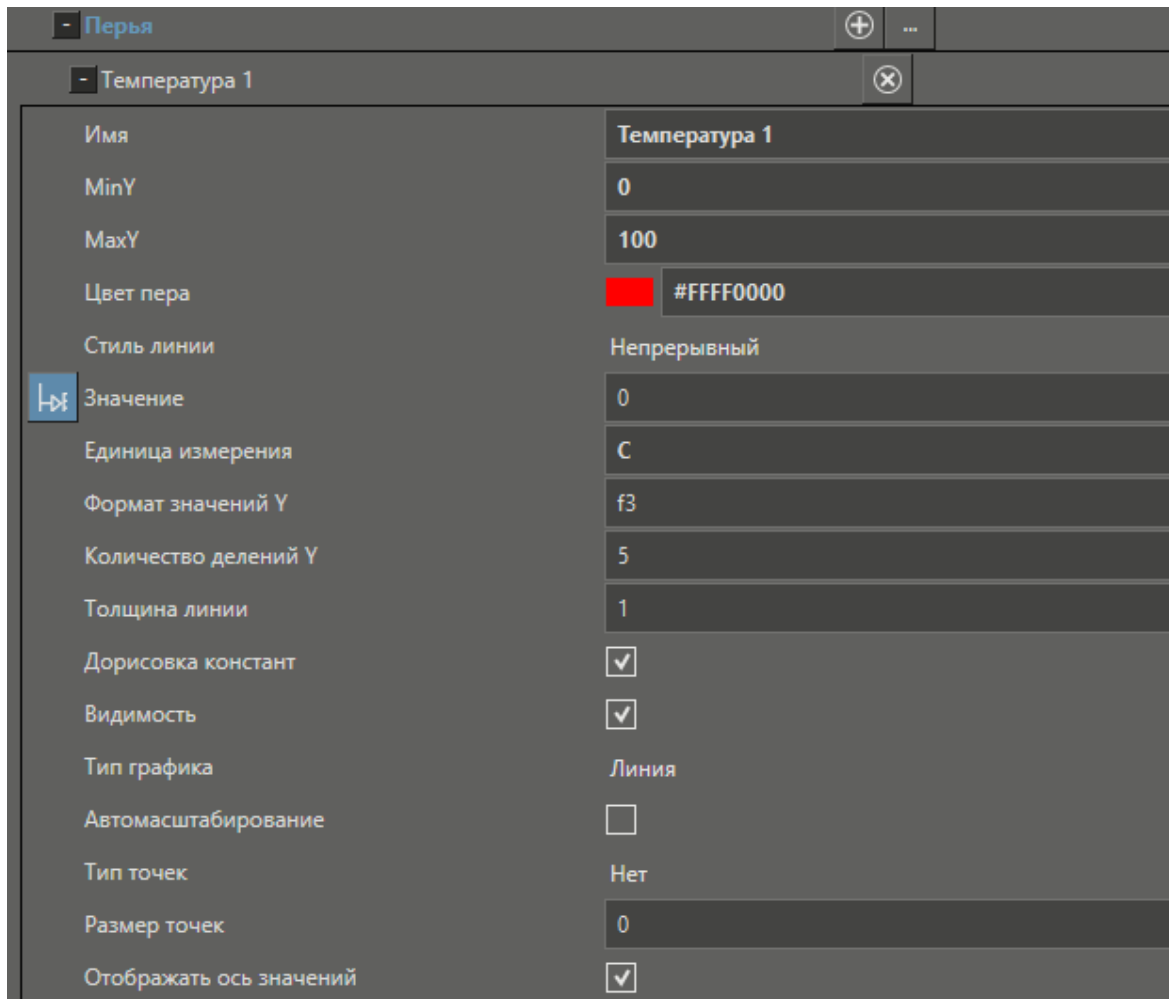
Название	Описание
Отображать легенду	Данное свойство имеет тип BOOL и может принимать значение TRUE (отображать легенду тренда) или False (не отображать).
Ширина минилегенды	Определяет ширину минилегенды в клиенте визуализации
Печать минилегенды	Определяет необходимость включения минилегенды в изображение при печати тренда или при его сохранении в файл. Если флаг установлен, то при сохранении/печати в изображение включается минилегенда, даже если была отключена на момент печати.
Прозрачность минилегенды	Задаёт степень прозрачности минилегенды. Значение изменяется в пределах от 0 до 100, где 0 - элемент полностью прозрачный, 100 - элемент полностью не прозрачный


Свойства группы Перья


Количество настроек соответствует типу данных TrendPen, который входит в состав библиотеки HMI. Типы данных.

Набор перьев представляет собой библиотечный тип TrendPens. Эту информацию можно использовать для создания трендов, на которых набор перьев будет меняться при помощи программ в клиенте визуализации в режиме исполнения.

Вид:





Новая группа добавится автоматически, после того как разработчик перетащит новый параметр в поле тренда. Кроме того, группа настроек может быть добавлена вручную с использованием кнопки .

Для удаления группы настроек необходимо нажать на кнопку: .

Описание настроек:

Название	Описание
Имя	Определяет название пера, которое будет отображаться в Легенде.
MaxY	Максимальное значение оси Y.
MinY	Минимальное значение оси Y.

Название	Описание
Автомасштабирование	Автоматически определяет значения MaxY и MaxX таким образом, чтобы тренд занимал всю площадь поля тренда.
Видимость	Определяет видимость пера в тренде. Данное свойство может быть задинамизировано, т.е. зависеть от других элементов управления, созданных разработчиком проекта, либо от какой-то программы. Если свойство пера примет значение False, то скроется не только перо, но и его ось значений.
Дорисовка констант	Определяет как будет рисоваться тренд, если значение параметра не будет меняться. Если флаг отмечен, то будет дорисовываться прямая линия. Если флаг снят, то график рисоваться не будет, до тех пор, пока не будет получено новое значение.
Единица измерения	Определяется единица измерения отображаемого параметра. Заданное значение будет отображаться в Минилегенде. Если параметр перетащили в поле тренда и образовалось перо, и у этого параметра настроена шкала, то эта настройка пера примет значение установленное в шкале параметра.
Значение	Настраивается связь между параметром и пером. В данное поле необходимо перетащить параметр, который требуется отображать на тренде в случае, если перо было создано вручную. Если параметр был перетащен в поле тренда, то данное свойство заполнится автоматически. Т.е. данное свойство обязательно должно быть задинамизировано.
Количество делений Y	Определяет количество делений оси Y.
Стиль линии	Определяет стиль линии тренда. Выпадающий список содержит следующие возможные значения: Непрерывный, Пунктир, Точка, Нет.
Тип графика	Определяет способ рисования тренда между точками тренда. Возможные варианты:

Название	Описание
	<ul style="list-style-type: none"> • Линия - между двумя точками будет рисоваться прямая линия, т.е. считается, что значение параметра менялось плавно. • Ступенька - между двумя точками будет рисоваться ступенька, т.е. считается, что значение изменилось скачкообразно, т.е. до тех пор, пока значение не обновилось, система будет считать, что оно было равно предыдущему значению. <p>Разработчик проекта должен выбирать это свойство, исходя из особенностей технологического процесса. На рисунке у красного пера свойство имеет значение Линия, а у зеленого - Ступенька:</p>  <p>На графике показаны две линии на координатной сетке. Красная линия (Линия) плавная, она начинается на низком уровне, повышается до пика, а затем плавно снижается. Зеленая линия (Ступенька) ступенчатая, она начинается на среднем уровне, скачком повышается до высокого уровня, а затем скачком снижается до среднего уровня.</p>
Толщина линии	Определяет толщину линии пера.
Формат значений Y	Определяет формат значений оси Y. Подробное описание свойства смотрите в разделе Формат значений.
Цвет пера	<p>Определяет цвет пера тренда. При наведении мыши на свойство появляется кнопка: , после нажатия на которую открывается диалоговое окно Выбор цвета. Если перетащить в поле тренда несколько параметров, то у связанных с ними перьев автоматически выберутся разные цвета.</p>

Название	Описание
Тип точек	<p>Определяет тип точек, нанесенных на тренд. Точка рисуется в местах излома тренда:</p> <ul style="list-style-type: none"> • Нет • Квадрат • Круг • Треугольник • Ромб • Крест
Размер точек	Определяет размер точек.
Отображать ось значений	Определяет видимость оси пера, если установлен флаг в свойстве Автоматически создавать оси значений в категории свойств Ось Y

Описание других свойств элемента смотрите в разделе НМІ.Свойства.

Настройки внешнего вида будут применены в клиенте визуализации в режиме исполнения. Чтобы оценить изменения, не загружая конфигурацию в узел, можно воспользоваться кнопкой Просмотр панели инструментов.

8.3.2.3. ГРАФИК XY

Графический элемент График XY предназначен для отображения массива типа ARRAY [*..2] OF [тип данных элементов], где по оси X будут находиться значения первой строки данных, а по оси Y значения второй.

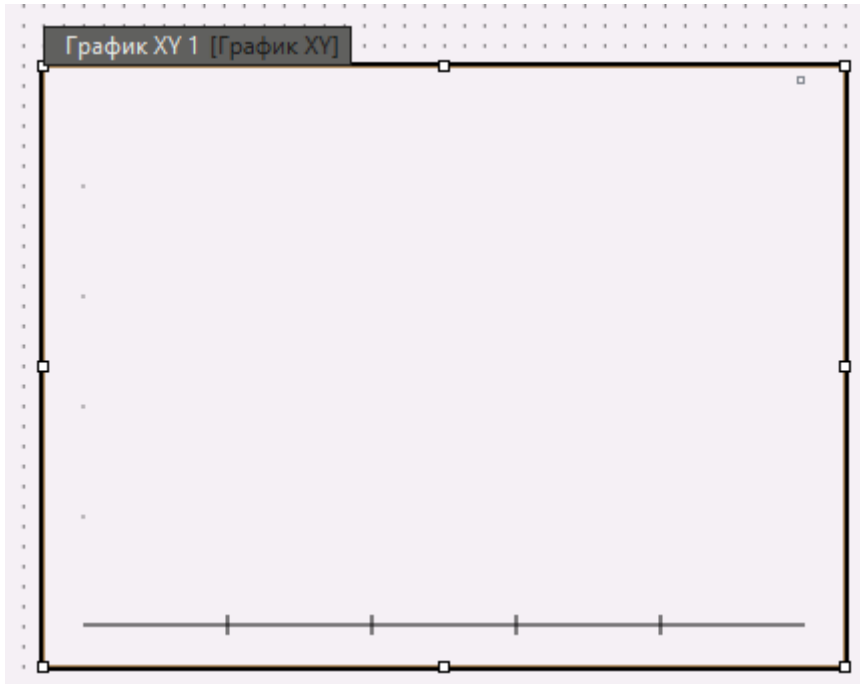
Важно! Элемент поддерживается только в НМІ v2

В палитре редактора НМІ элемент График XY находится в категории Контролы.

Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора НМІ:



Для того, чтобы добавить параметр с типом данных массив на График ХУ, необходимо перетащить параметр из дерева в поле графика. При этом внешний вид элемента изменится: в верхнем левом углу элемента появится минилегенда, одновременно с этим в панели свойств добавится группа свойств, относящаяся к добавленному перу.

Если параметр, связанный с пером, будет удален, или в контекстном меню параметра будет разорвана связь между параметром и пером, то перо останется в графике, и все настройки, например, цвет, тип линии сохранятся. Для того чтобы удалить созданное перо, следует воспользоваться панелью свойств графика.

Вид графика в клиенте визуализации:

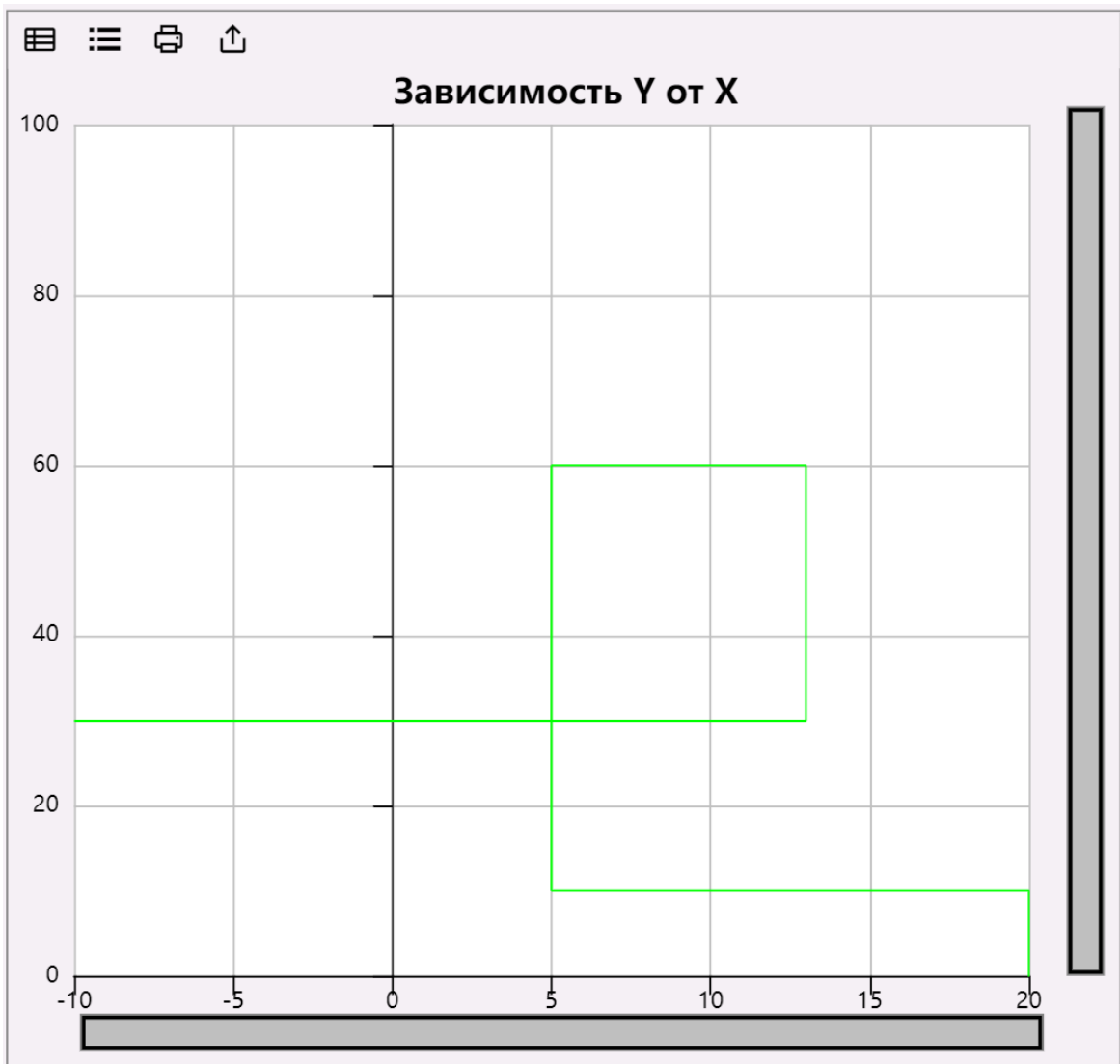


График в клиенте визуализации имеет следующие элементы:

- Панель инструментов
- Поле графика - место, где отображается линия графика. Поле графика может содержать сетку
- Ось X- ось значений, на которой отображаются значения первого столбца массива
- Ось Y -ось значений. Если в график добавлено несколько параметров, то отображается несколько осей Y.
- Курсор - вертикальная черта, появляющаяся при нажатии левой кнопкой мыши на поле графика, рядом с которой отображается значение X, соответствующее пересечению этой чертой оси X.
- Минилегенда - - список имен перьев, их обозначений и текущих значений, находящийся в верхней части графика. Отображается только при соответствующей настройке в панели свойств

- Вертикальный/горизонтальный ползунок интервала - инструменты для изменения отображаемого интервала. Находятся справа от поля графика и под ним.

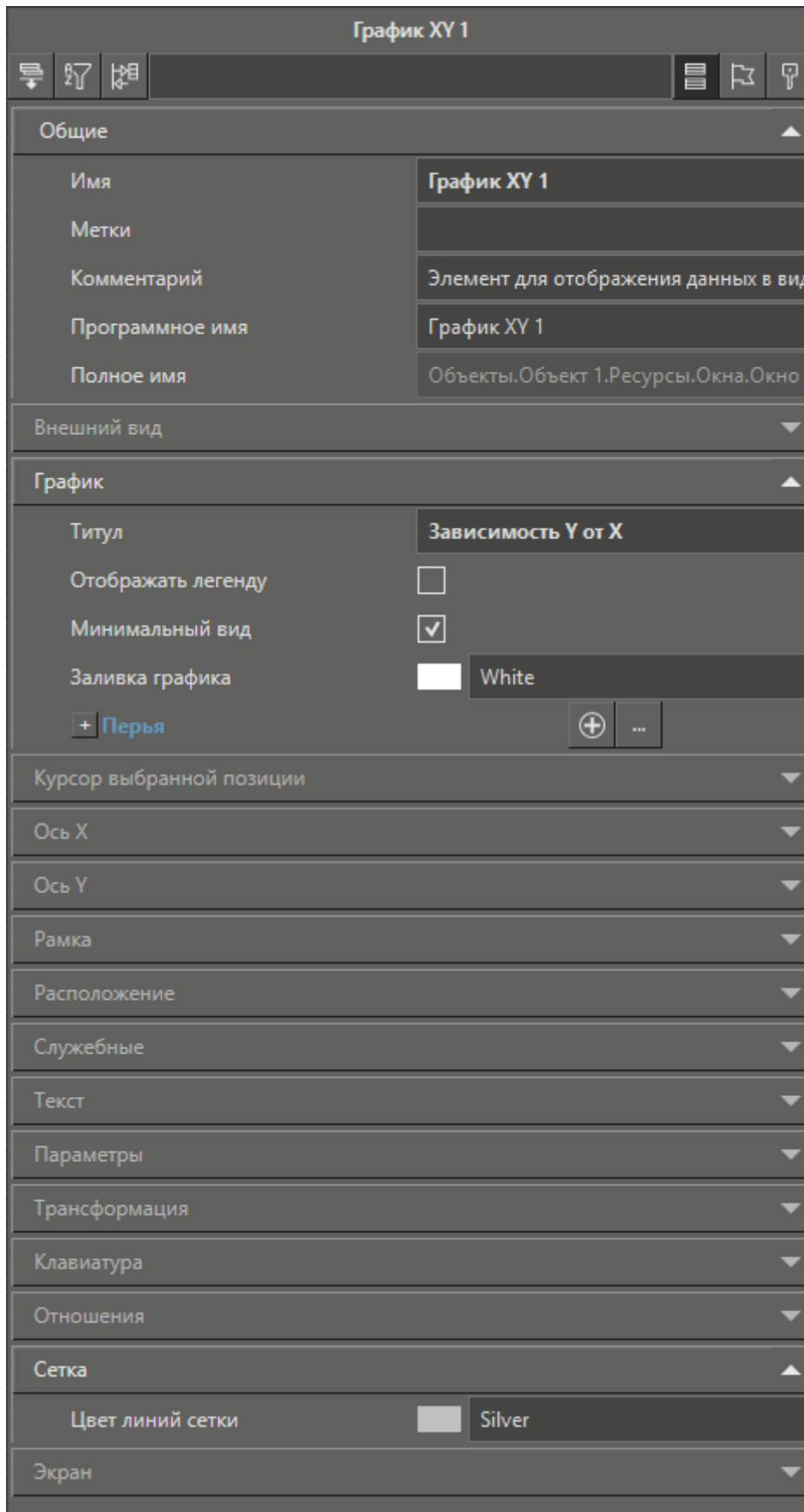
Внешний вид графика может быть изменен при помощи панели свойств. Любое свойство может быть задинамизировано, и таким способом разработчик проекта может создать те элементы управления графиком, которые необходимы пользователю.

Настройки внешнего вида будут применены в клиенте визуализации в режиме исполнения, Для того чтобы оценить изменения, не загружая конфигурацию в узел, можно воспользоваться кнопкой Просмотр панели инструментов




Порядок работы с элементами управления графика, такой же как и с элементами управления тренда.



8.3.2.3.1. СВОЙСТВА ГРАФИКА XY




Вид панели свойств элемента:




Описание основных свойств графического элемента График ХУ:

Название	Описание
Категория График	
Титул	Свойство определяет заголовок графика.
Отображать легенду	Данное свойство имеет тип BOOL и может принимать значение TRUE (отображать легенду графика) или False (не отображать легенду графика).
Минимальный вид	Если данное свойство имеет значение TRUE, то на графике скрываются все элементы, кроме Поля графика, которое, в этом случае, занимает всю площадь Графика. При этом управление графиком может осуществляться элементами управления, созданными разработчиком проекта. Например, для изменения интервала может использоваться текстовый ввод, связанный с параметром, передающим данные в свойство графика Интервал.
Заливка графика	Определяет цвет заливки Поля графика. При нажатии на кнопку  , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.
Группа Перья	При нажатии на кнопку  могут быть добавлены перья на график. Данная группа содержит подгруппы, количество подгрупп соответствует количеству перьев на графике. Описание свойств подгрупп описано в таблице ниже.
Категория Курсор выделенной позиции	
Цвет курсора	Определяет цвет вертикального курсора. При нажатии на кнопку  , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.
Толщина курсора	Определяет толщину курсора.

Название	Описание
Видимость	Это свойство имеет тип BOOL и задает видимость курсора выбранной позиции на графике.
Категория Ось X	
Подпись X	Подпись оси X. Если задан текст, то он отобразится в клиенте визуализации около оси X правее поля графика.
Формат значений X	Задает формат значений оси X графика.
Количество делений X	Задается желаемое количество делений оси. При построении графика среда исполнения будет выбирать значение этого параметра по возможности близкое к заданному.
Толщина делений X	Определяет внешний вид деления шкалы.
Длина делений X	Определяет внешний вид деления шкалы.
Цвет делений X	Определяет цвет делений оси. При нажатии на кнопку  , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.
Верхний отступ делений X	Определяет расстояние от оси до значения, соответствующего тому или иному делению.
Цвет оси X	Определяет цвет оси. При нажатии на кнопку  , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета..
Толщина оси X	Определяет толщину оси.
Категория Ось Y	
Подпись Y	Подпись оси Y. Если задан текст, то он отобразится в клиенте визуализации около оси Y над полем графика.

Название	Описание
Длина делений Y	Определяет внешний вид деления шкалы.
Правый отступ делений Y	Определяет расстояние от оси до значения, соответствующего тому или иному делению.
Цвет делений Y	Определяет цвет делений оси. При нажатии на кнопку  , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.
Цвет оси Y	Определяет цвет оси. При нажатии на кнопку  , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.
Толщина оси Y	Определяет толщину оси.
Толщина делений Y	Определяет внешний вид деления шкалы
Категория Текст	
Цвет текста	Определяет цвет текста графика. При нажатии на кнопку  , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.
Категория Параметры	
Шаг прореживания	Данную настройку полезно использовать в случае, когда на графике необходимо отобразить большее количество данных, чем может поместиться физически. Поскольку в одном пикселе отображается только одно значение, то не рационально считывать из массива данных больше, чем может быть использовано на графике. Настройка определяет, на какое число пикселей должна приходиться группа из не более чем четырех значений. Для этого вычисляется интервал данных по формуле = Интервал времени видимой области графика / (Ширина графика (в px) * Шаг прореживания). Выборка из массива делается с тем условием, что за каждый такой интервал данных из него придет не более четырех

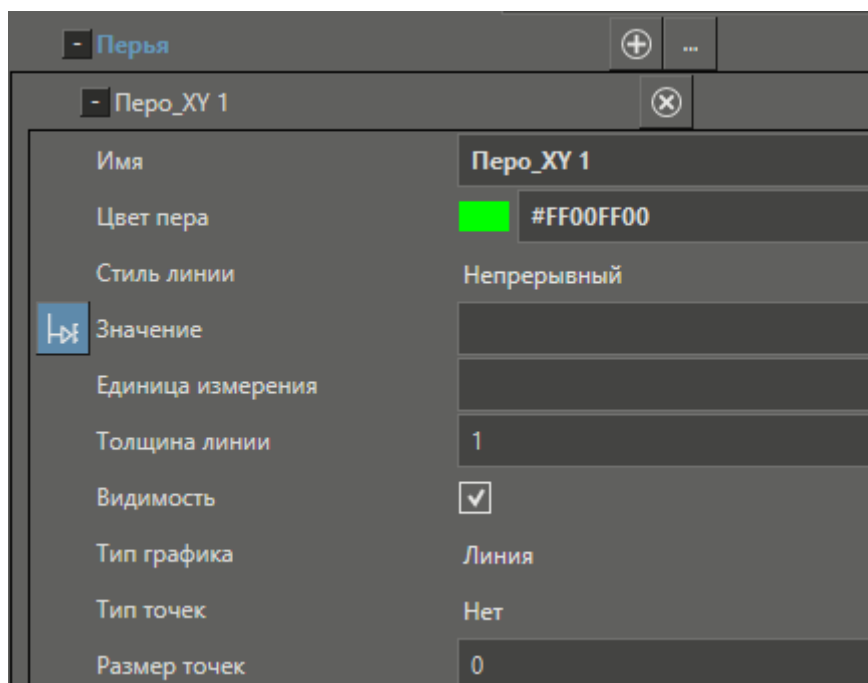
Название	Описание
	значений (первое, последнее, минимальное, максимальное). По умолчанию Шаг прореживания равен 1.
Категория Сетка	
Цвет сетки	Определяет цвет линий сетки. При нажатии на кнопку  , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.


Свойства группы Перья


Количество настроек соответствует типу данных ChartPen, который входит в состав библиотеки HMI. Типы данных.

Набор перьев представляет собой библиотечный тип ChartPens. Эту информацию можно использовать для создания графиков, на которых набор перьев будет меняться в клиенте визуализации

Вид группы в клиенте визуализации:





Новая группа добавится автоматически, после того как разработчик перетащит новый параметр в поле графика. Кроме того, группа настроек может быть добавлена вручную с использованием кнопки .

Для удаления группы настроек необходимо нажать на кнопку .

Описание настроек:

Название	Описание
Имя	Определяет название пера, которое будет отображаться в Легенде.
Видимость	Определяет видимость пера в графике. Данное свойство может быть задинамизировано, т.е. зависеть от других элементов управления, созданных разработчиком проекта, либо от какой-то программы.
Единица измерения	Определяет единицу измерения отображаемого параметра. Заданное значение будет отображаться в Легенде.
Значение	Настраивает связь между параметром и пером. В данное поле необходимо перетащить параметр, который требуется отображать на графике, в случае, если перо было создано вручную. Если параметр был перетащен в поле графика, то данное свойство заполнится автоматически. Т.е. данное свойств обязательно должно быть задинамизировано.
Стиль линии	Определяет стиль линии графика. Выпадающий список содержит следующие возможные значения: Непрерывный, Пунктир, Точка, Нет.
Тип графика	<p>Определяет способ рисования графика между точками графика. Возможные варианты:</p> <ul style="list-style-type: none"> • Линия - между двумя точками будет рисоваться прямая линия, т.е. будет считаться, что значение параметра менялось плавно. • Ступенька - между двумя точками будет рисоваться ступенька, т.е. будет считаться, что значение изменилось скачкообразно, т.е. до тех пор, пока значение не обновилось, система будет считать, что оно было равно предыдущему значению. <p>Разработчик проекта должен выбирать это свойство, исходя из особенностей технологического процесса. На рисунке у</p>

Название	Описание
	<p>красного пера свойство имеет значение Линия, а у зеленого - Ступенька:</p> 
Толщина линии	Определяет толщину линии пера.
Цвет пера	<p>Определяет цвет пера графика. При нажатии на кнопку , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета. Если перетащить в поле графика несколько параметров, то у связанных с ними перьев автоматически выберутся разные цвета.</p>
Тип точек	<p>Определяет тип точек, нанесенных на график. Точка рисуется в местах излома графика.</p> <ul style="list-style-type: none"> • Нет • Квадрат • Круг • Треугольник • Ромб • Крест
Размер точек	Определяет размер точек.

Описание других свойств элемента смотрите в разделе НМІ.Свойства

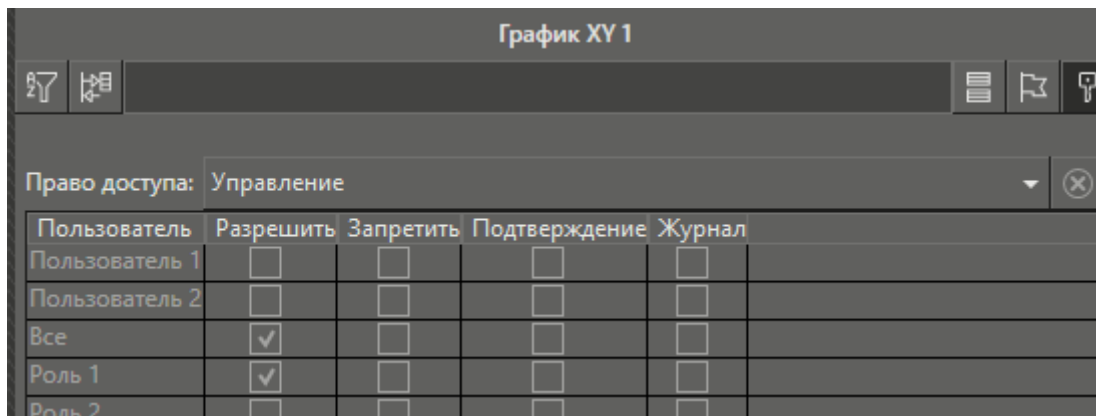
Настройки внешнего вида будут применены в клиенте визуализации в режиме исполнения. Для того чтобы оценить изменения, не загружая конфигурацию в узел, можно воспользоваться кнопкой Просмотр панели инструментов.

8.3.2.3.2. ГРАФИК ХУ. ПРАВА ДОСТУПА

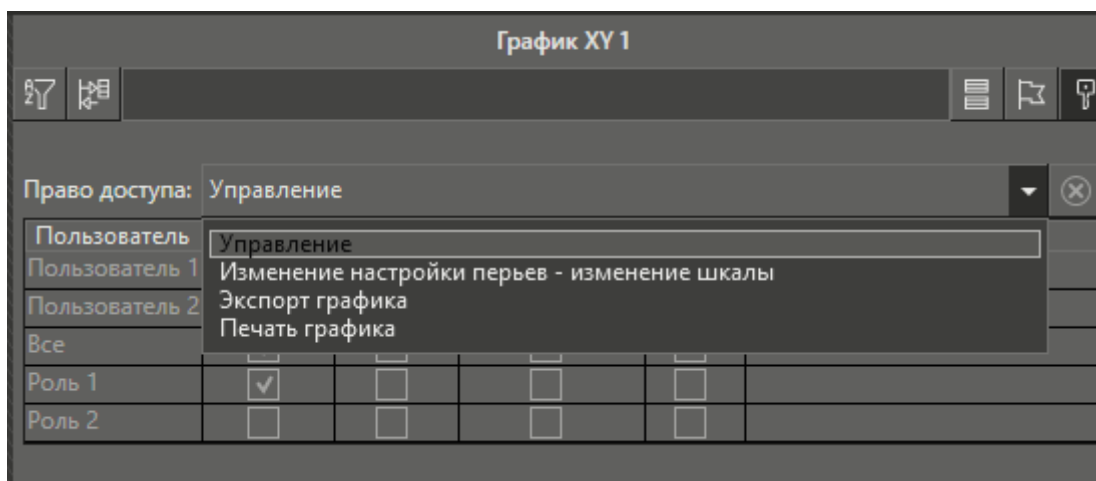
Для работы с Графиком ХУ в режиме исполнения предусмотрены специальные права доступа.

В дереве системы в группе Безопасность настраиваются умолчания для всех добавляемых в проект График ХУ. У каждого отдельного Графика ХУ в его панели свойств можно изменить сделанные умолчания.

Вид панели свойств График ХУ в режиме редактирования Прав:



Доступные для настройки права доступа:



Право	Назначение
Управление	Определяет возможность пользователя работать с графиком в режиме исполнения. Если право не предоставлено, то график будет полностью отключен

Право	Назначение
	(полностью недоступен для управления). Пользователь не сможет воспользоваться панелью инструментов, а также изменить масштаб или интервал отображаемых данных при помощи мыши. Как правило, это право пользователям предоставляется. Даже если на другие графические элементы это право не распространяется.
Изменение настройки перьев - изменение шкалы	Определяет возможность пользователя менять значения нижней и верхней границы отображения графика по оси Y
Экспорт графика	Определяет возможность экспорта графика в табличном виде или в графическом. Если место сохранения файлов указано в приложении MasterSCADA 4D Client Monitor (если клиент запускается на устройстве отличном от исполнительной системы) или в приложении MasterSCADA 4D Monitor в поле Параметры запуска клиента, и установлен флаг Журнал, то в журнале сформируется сообщение Экспорт графика. Текст "Сохранение графика в файл <имя файла и путь>"
Печать графика	Определяет возможность пользователю печатать график в клиенте визуализации. Если принтер указан в приложении MasterSCADA 4D Client Monitor (если клиент запускается на устройстве отличном от исполнительной системы) или в приложении MasterSCADA 4D Monitor в поле Параметры запуска клиента, и установлен флаг Журнал то в журнале сформируется сообщение Печать графика. Текст "Печать графика на принтер <имя принтера>"

8.3.2.4. ДЕРЕВО ОДИНОЧНОГО/МНОЖЕСТВЕННОГО ВЫБОРА

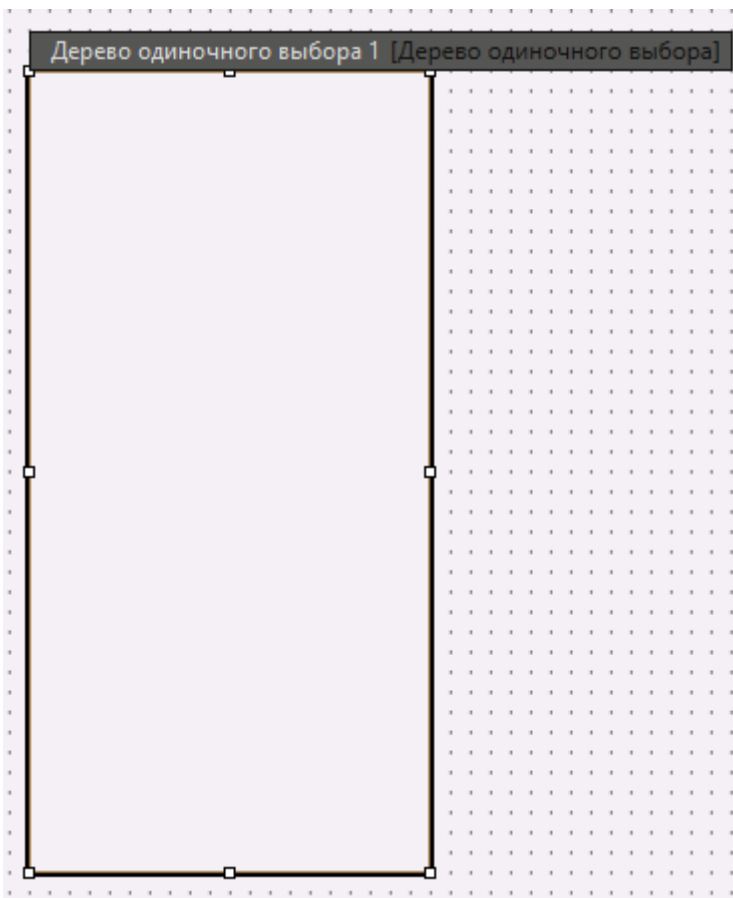
Элемент Дерево одиночного выбора отображает данные в виде дерева и позволяет выбрать один компонент. Элемент Дерево множественного выбора отображает данные в виде дерева и позволяет выбрать несколько компонентов

В палитре редактора НМІ элемент Дерево одиночного выбора находится в категории Контролы.

Вид элементов в палитре:

Дерево одиночного выбора	Дерево множественного выбора
	

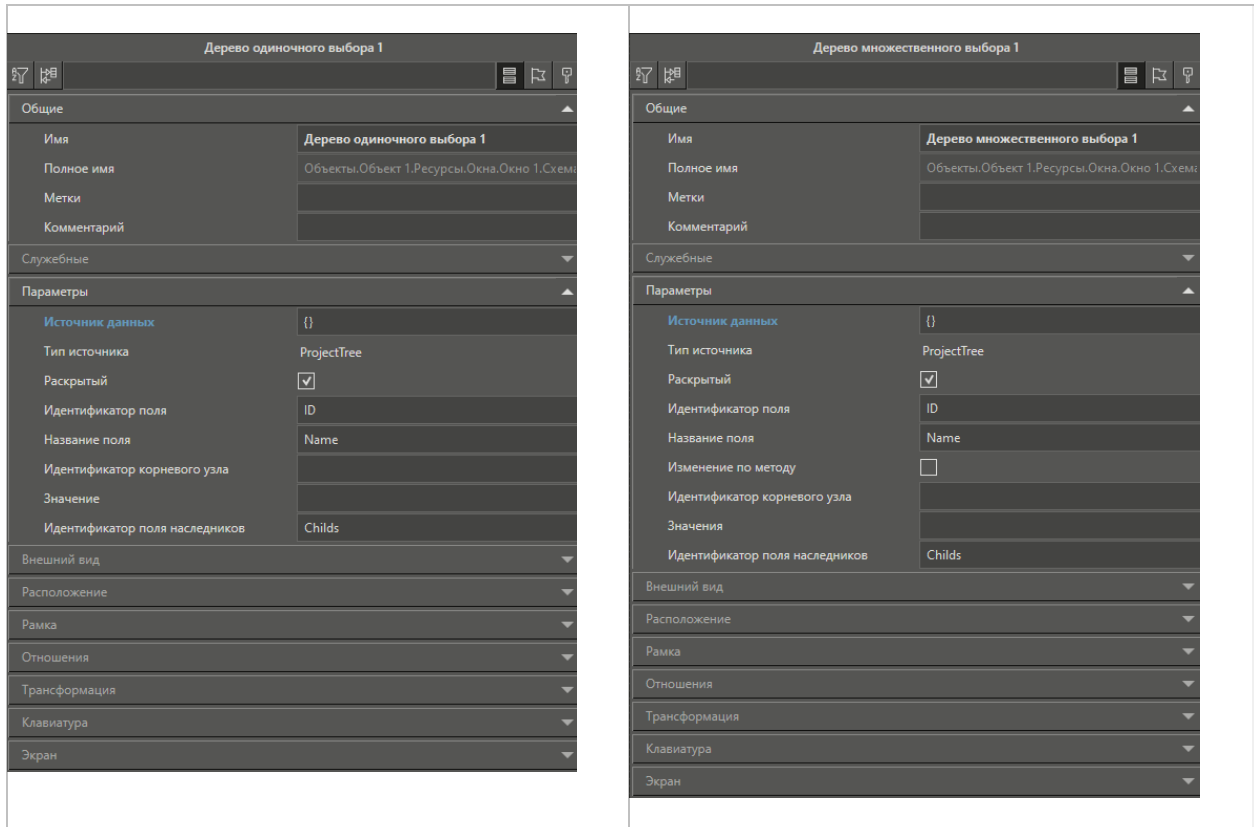
Вид элемента после добавления в рабочую область редактора HMI:



Какие данные будут отображаться в деревьях зависит от настроек элемента.

Вид панели свойств:

Дерево одиночного выбора	Дерево множественного выбора
--------------------------	------------------------------



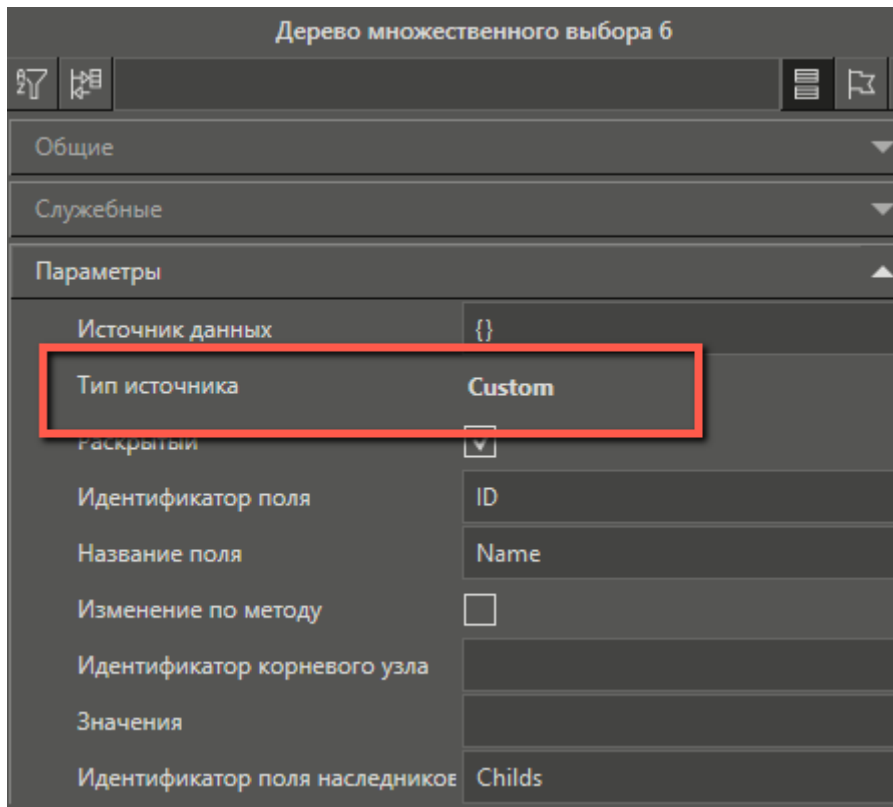
Описание основных свойств графического элемента Дерево одиночного выбора:

Название	Назначение
Источник данных	<p>Определяет содержимое дерева. Нужно или задать константу, или связать с параметром типа STRING. Задается разработчиком проекта только в том случае, если свойство Тип источника имеет значение Custom. Представляет собой строку JSON. Порядок формирования строки JSON смотрите в разделе Источник данных для дерева одиночного/множественного выбора</p>
Тип источника	<p>Определяет способ формирования источника данных. Возможные варианты:</p> <ul style="list-style-type: none"> ProjectTree - отображаются дерево проекта с параметрами, группами и другими важными компонентами. Objects - отображаются основные компоненты проекта. Custom - отображаются данные поступающие на свойство Источник данных.

Название	Назначение
Раскрытый	Флаг определяет, в каком состоянии будет дерево. Если флаг установлен, то дерево будет раскрыто.
Идентификатор поля	Поле влияет на то, какое значение будет выводить в свойствах Значение (для Дерева одиночного выбора) и Значения (для Дерева множественного выбора). По умолчанию установлено значение ID - в этом случае будет указаны ID элементов выделенных в дереве. Менять можно только для типа источника Custom, т.е. можно указать имя атрибута JSON, в котором записан идентификатор узла.
Название поля	Имя атрибута JSON, в котором записано отображаемое имя для узла дерева.
Изменение по методу	Если флаг установлен, то изменения сделанные пользователем в клиенте визуализации будут применяться, только после использования методов, предназначенных для работы с деревом. Используется только для Дерева множественного выбора.
Идентификатор корневого узла	Определяет, что будет отображаться только указанный данный объект и его элементы дочерние.
Значение/Значения	Показывает выделенные поля в дереве. Тип информации выдаваемой в данном свойстве зависит от значения свойства Идентификатор поля.
Идентификатор поля наследников	Определяет символ, которым будет определяться дочерние элементы

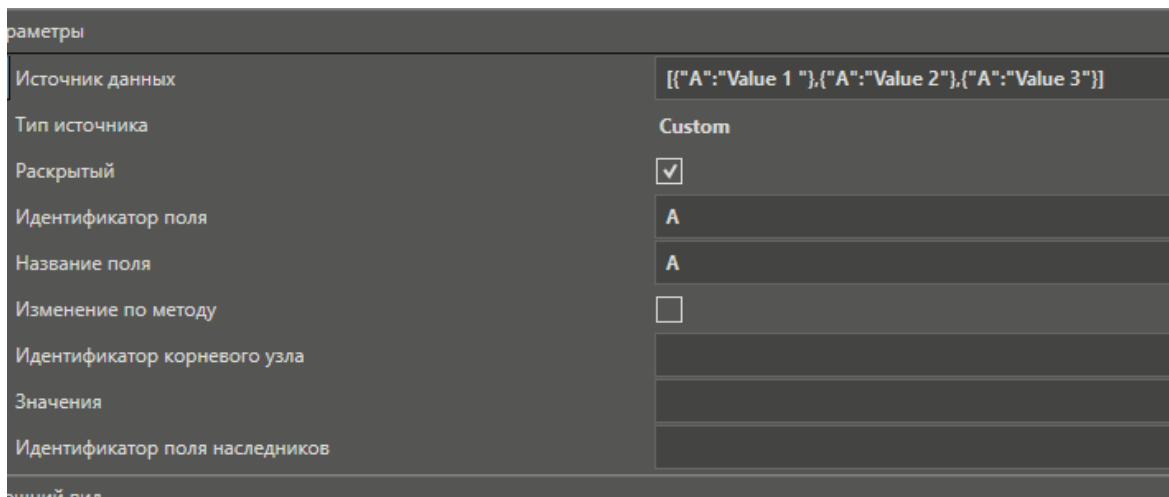
8.3.2.4.1. ИСТОЧНИК ДАННЫХ ДЛЯ ДЕРЕВА ОДИНОЧНОГО/МНОЖЕСТВЕННОГО ВЫБОРА

Данные Дерева одиночного/множественного выбора можно формировать вручную. В этом случае Тип источника будет Custom:

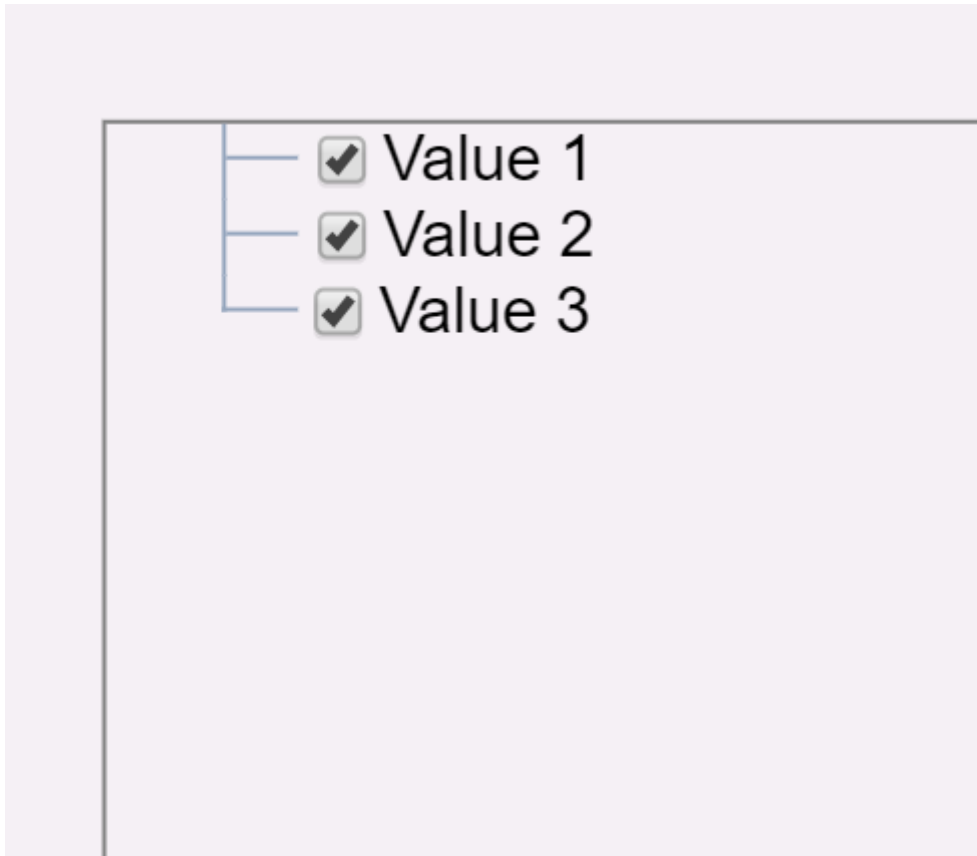


В этом случае строку для свойства Источник данных нужно сформировать в формате JSON. Для первого уровня вложенности формат должен быть в таком виде:

```
[{"A": "Value 1 "}, {"A": "Value 2"}, {"A": "Value 3"}]
```



В клиенте визуализации получим:



Если необходимо добавить в дерево значения второго уровня вложенности, то JSON немного поменяется и будет в таком виде:

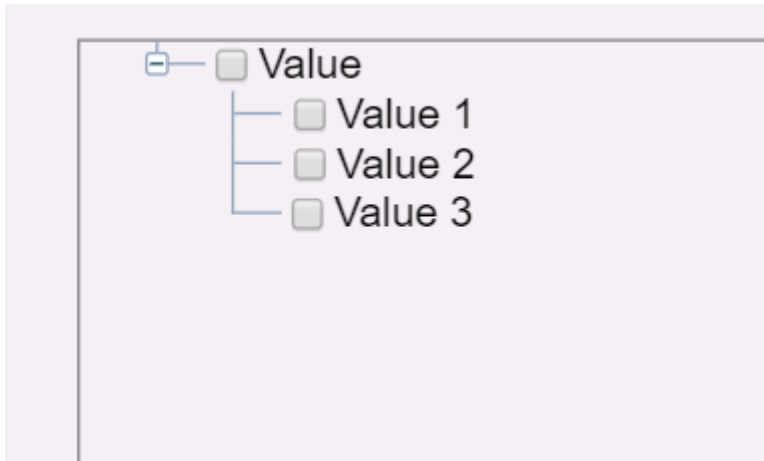
```
[{"B":[{"A":"Value 1 "},{ "A":"Value 2"}, {"A":"Value 3"}], "A":"Value"}]
```

Тогда Идентификатором поля будет значение А, а Идентификатором поля наследников - В:

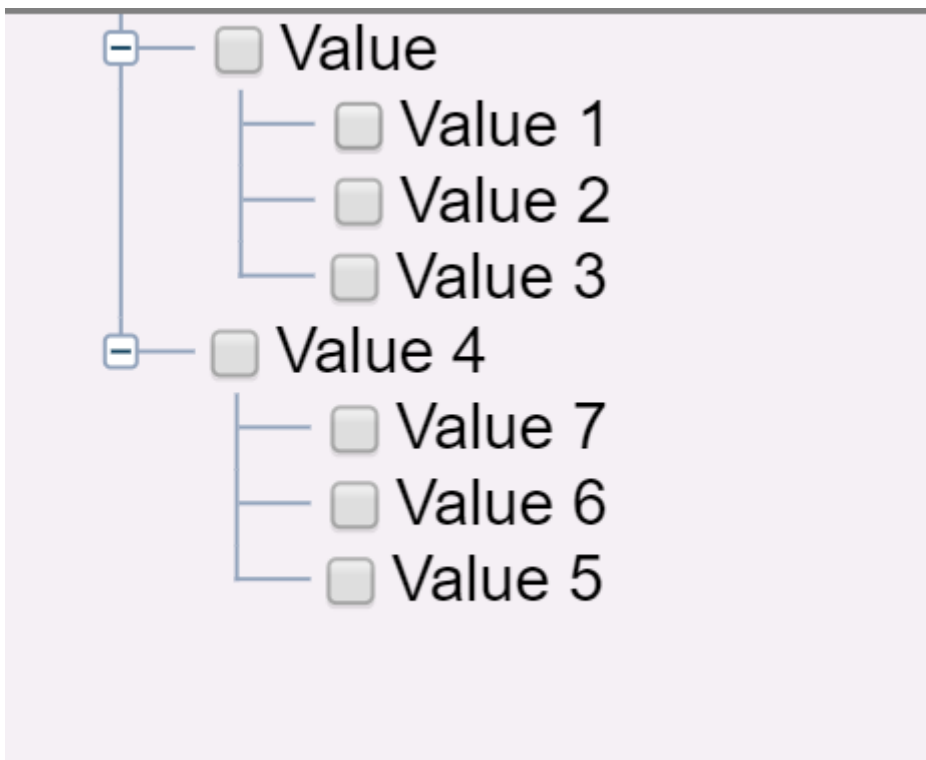
Параметры	
Источник данных	[{"B":[{"A":"Value 1 "},{ "A":"Value 2"}, {"A":"Value 3"}], "A":"Value"}]
Тип источника	Custom
Раскрытый	<input checked="" type="checkbox"/>
Идентификатор поля	A
Название поля	A
Изменение по методу	<input type="checkbox"/>
Идентификатор корневого узла	
Значения	
Идентификатор поля наследников	B
Внешний вид	

В клиенте структура дерева будет в таком виде:

```
[{"B":[{"A":"Value 1 "},{ "A":"Value 2"}, {"A":"Value 3"}], "A":"Value"}, {"B":[{"A":"Value 7"}, {"A":"Value 6"}, {"A":"Value 5"}], "A":"Value 4"}]
```

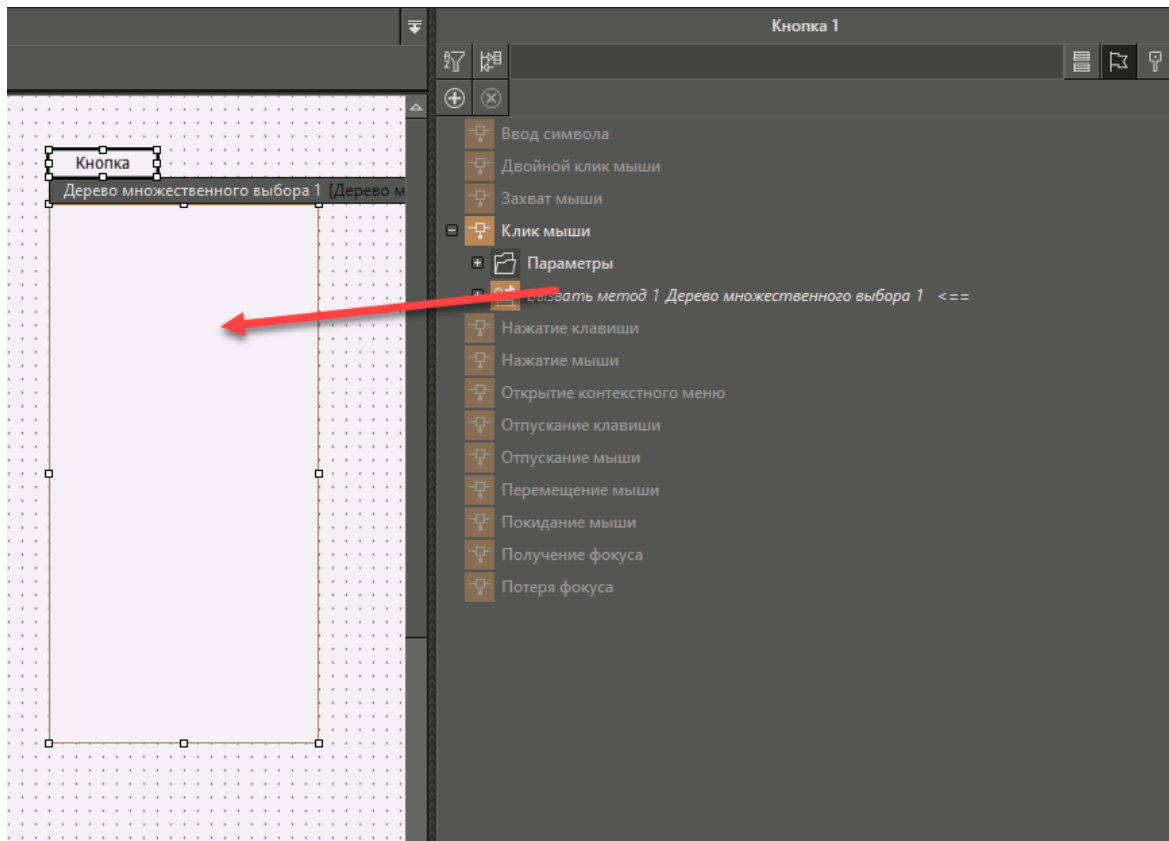


Для такой строки JSON [{"B":[{"A":"Value 1 "}, {"A":"Value 2"}, {"A":"Value 3"}], "A":"Value"}, {"B":[{"A":"Value 7 "}, {"A":"Value 6"}, {"A":"Value 5"}], "A":"Value 4"}] результат будет уже таким:

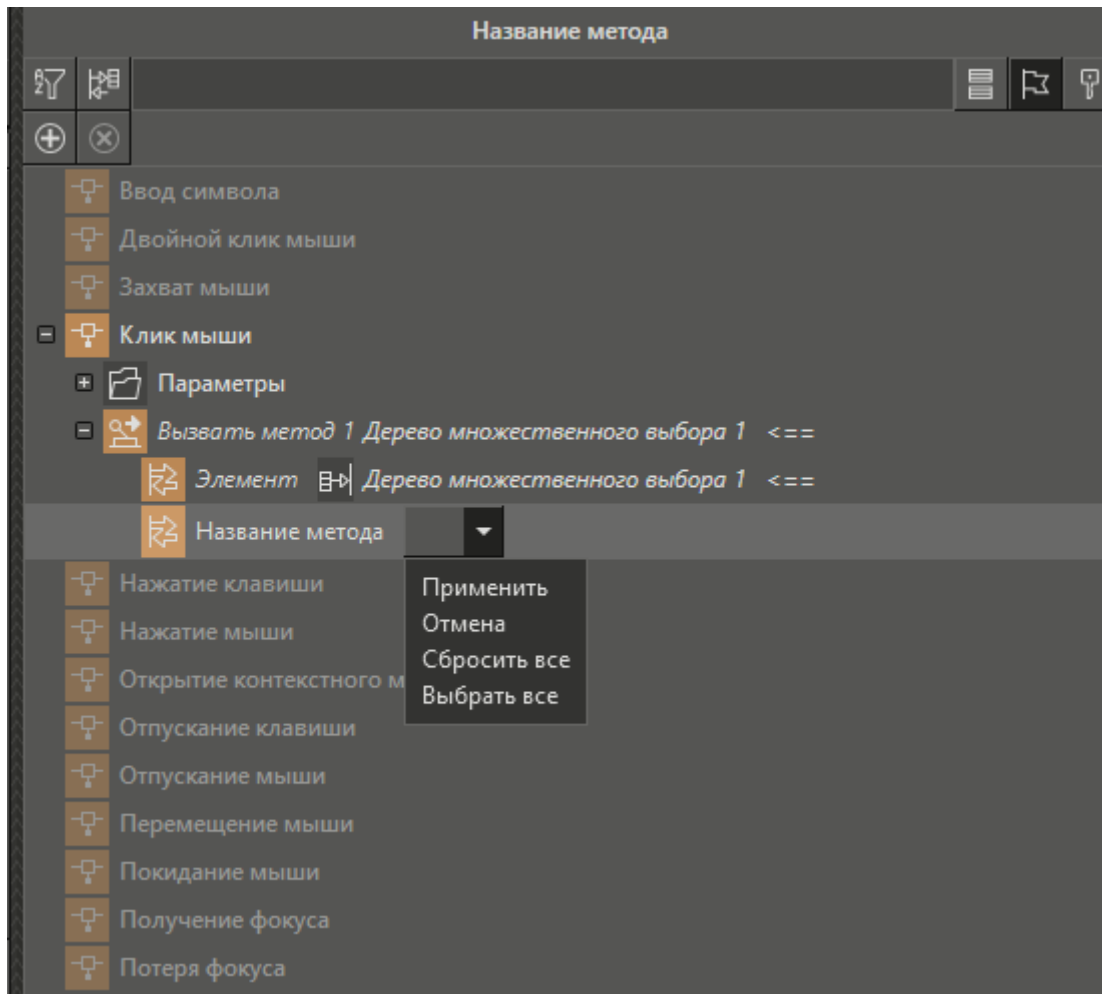


8.3.2.4.2. МЕТОДЫ ДЛЯ РАБОТЫ ДЕРЕВОМ МНОЖЕСТВЕННОГО ВЫБОРА

Для работы с деревом множественного выбора можно создать элементы управления, например кнопки, и настроить события, например, в событие Клик мыши добавить действие Вызвать метод, связать параметр Элемент с Дерево множественного выбора (для этого можно потянуть параметр метода на элемент):



Затем указать какое действие с трендом должно выполняться при нажатии на элемент управления:



Возможные методы:

Название	Описание
Применить	Передает информацию о выделенных полях. Используется только если установлен флаг Изменение по методу в свойствах элемента.
Отмена	Отменяет последнее выполненное действие. Используется только если снят флаг Изменение по методу в свойствах элемента.
Сбросить все	Убирает выделения со всех элементов.
Выбрать все	Устанавливает выделение у всех элементов.

8.3.2.5. ЛЕГЕНДА

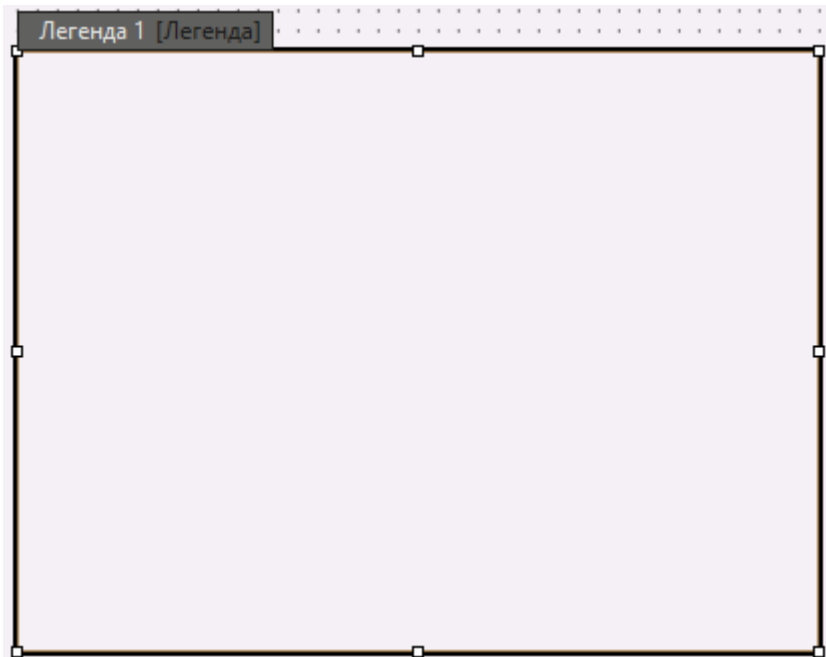
Графический элемент Легенда является вспомогательным элементом тренда, служит для предоставления пользователю дополнительной информации о перьях добавленных на тренд.

В палитре редактора HMI элемент Легенда находится в категории Контролы.

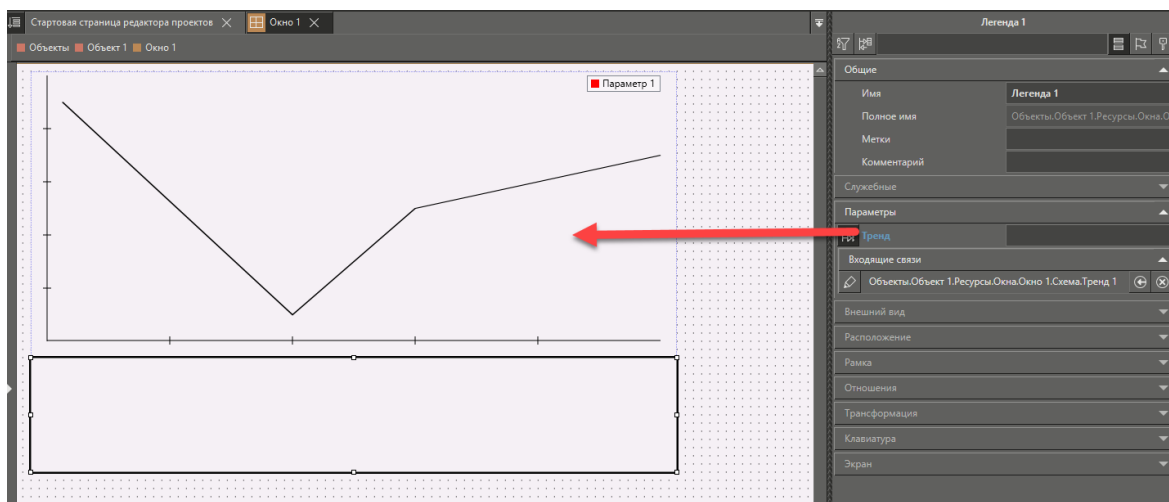
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:



Самый быстрый способ связать Легенду и Тренд это перетащить соответствующее свойство элемента на Тренд:



После этого в Легенде в клиенте визуализации будут отображаться дополнительная информация о перьях тренда.

При необходимости можно настроить панель свойств элемента.

8.3.2.5.1. ЛЕГЕНДА. ПРАВА ДОСТУПА

Легенда позволяет изменять настройки отображения перьев на тренде в клиенте визуализации. Возможность пользователя выполнять эти действия определяются правами доступа Легенды.

Задать значения прав доступа по умолчанию для всех элементов проекта можно в дереве системы, в настройках элемента Безопасность. Уточнить права доступа для конкретного элемента можно в его панели свойств.

Вид панели свойств Легенды в режиме Права доступа:



Название	Назначение
Управление	Определяет возможность пользователя работать с легендой в режиме исполнения. Если право не предоставлено, то легенда будет полностью отключена (полностью недоступна для управления)
Изменение настройки перьев-изменение шкалы	Определяет возможность пользователя менять значения нижней и верхней границы отображения графика по оси Y (значения в столбцах легенды Шк.Мин., Шк.Макс.)

8.3.2.5.2. ЛЕГЕНДА В КЛИЕНТЕ ВИЗУАЛИЗАЦИИ

Легенда после установки связи с трендом в клиенте визуализации будет иметь вид:

Вид	Перо	Имя	Источник	Ед...	Значение	Вр.посл.	Кач...	Ш...	Шк.Макс.	НАУ	НПУ	ВПУ	В...	Нач.знач.	Мин.знач.	Макс.знач.	Кон.
<input checked="" type="checkbox"/>		Уровень	Объекты.Об...	дм	3.621873102...	18.02.2021,...	0	0	100	5	25	75	90	84.71206929...	84.71206929...	84.712069295...	84.7
<input checked="" type="checkbox"/>		Темпер...	Объекты.Об...	С	46.06746221...	18.02.2021,...	0	0	100	5	15	80	95	64.18024721...	64.18024721...	64.180247214...	64.1
<input checked="" type="checkbox"/>		Парам...	Объекты.Об...		30.000	18.02.2021,...	0	0	100					30.000	30.000	30.000	30.0

Каждому перу соответствует строка в легенде. Выделенное перо на тренде в Легенде подсвечено синим цветом.

Пользователь может изменить ширину столбцов потянув мышью за их границы, а также отсортировать перья по любому столбцу. Изменить порядок сортировки можно кликнув по заголовку столбца. Справа от названия столбца размещается индикатор, в виде треугольника. Возможный внешний вид индикатора:

- Треугольник бледно-серого цвета - сортировки по данному столбцу нет.
- Треугольник яркий, направлен вершиной вниз - сортировка идет от большего значения к меньшему.
- Треугольник яркий, направлен вершиной вверх - сортировка идет от меньшего значения к большему

Столбцы легенды:

Название	Назначение
Вид	Флаг определяет видимость пера на тренде. В заголовке столбца есть флаг, который позволяет отключить/включить видимость всех перьев на связанном тренде
Перо	Отображает тип и цвет пера в тренде. Позволяет изменить параметры, отвечающие за внешний вид пера.
Имя	Отображает имя пера
Источник	Отображает полное имя параметра, с которым связано перо.
Ед.изм.	Отображаются единицы измерения. Если параметр перетаскили в поле тренда и образовалось перо, и у этого параметра настроена шкала, то эта настройка пера примет значение установленное в шкале параметра (если она не переопределена в режиме разработки в настройках пера).
Значение	Отображает последнее полученное значение.
Вр.посл.	Отображает время получения последнего значения.

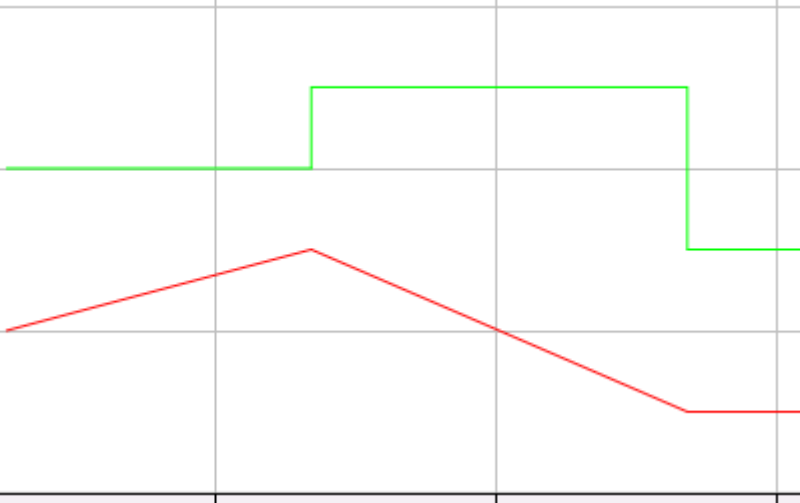
Название	Назначение
Кач-во	Отображается признак качества полученного значения. Если значение 0 - признак качества хороший.
Шк.Мин	Отображает минимальное значение, которое может отобразиться на тренде. Эту настройку можно изменить, дважды кликнув в соответствующей ячейке.
Шк.Макс	Отображает максимальное значение, которое может отобразиться на тренде. Эту настройку можно изменить, дважды кликнув в соответствующей ячейке.
НАУ	Отображается значение нижней аварийной границы, если оно было задано в настройках шкалы связанного параметра
НПУ	Отображается значение нижней предупредительной границы, если оно было задано в настройках шкалы связанного параметра
ВПУ	Отображается значение верхней предупредительной границы, если оно было задано в настройках шкалы связанного параметра
ВАУ	Отображается значение верхней аварийной границы, если оно было задано в настройках шкалы связанного параметра
Нач.знач.	Отображает первое значение выделенного интервала на тренде. Если интервал не выбран, то поле пустое
Мин.знач.	Отображает минимальное значение выделенного интервала на тренде. Если интервал не выбран, то поле пустое
Макс.знач.	Отображает максимальное значение выделенного интервала на тренде. Если интервал не выбран, то поле пустое
Кон.знач.	Отображает последнее значение выделенного интервала на тренде. Если интервал не выбран, то поле пустое

Изменение внешнего вида пера

Для изменения внешнего вида пера необходимо дважды кликнуть в столбце Перо в строке отвечающем за график, который необходимо изменить. Откроется диалоговое окно:

Тип графика	Линия ▼
Толщина	— 1 +
Цвет	
Стиль линии	Линия ▼
Вид точки	Нет ▼
Размер точки	— 1 +
Видимость шкалы	<input checked="" type="checkbox"/>
<input type="button" value="Применить"/> <input type="button" value="Отмена"/>	

Свойство	Описание
Тип графика	<p>Определяет способ рисования тренда между точками тренда. Возможные варианты:</p> <ul style="list-style-type: none"> • Линия - между двумя точками будет рисоваться прямая линия, т.е. считается, что значение параметра менялось плавно. • Ступенька - между двумя точками будет рисоваться ступенька, т.е. считается, что значение изменилось скачкообразно, т.е. до тех пор, пока значение не обновилось, система будет считать, что оно было равно предыдущему значению. <p>Пользователь может выбирать это свойство, исходя из особенностей технологического процесса. На рисунке у красного пера свойство имеет значение Линия, а у зеленого - Ступенька:</p>

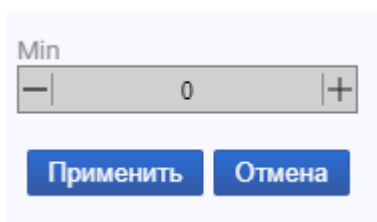
Свойство	Описание
	
Толщина	Определяет толщину линии пера.
Цвет	Позволяет выбрать цвет. При нажатии на цвет, откроется всплывающее окно, в котором можно определить цвет. Внешний вид окна соответствует виду элемента Кнопка выбора цвета
Стиль линии	Определяет стиль линии тренда. Выпадающий список содержит следующие возможные значения: Непрерывный, Пунктир, Точка, Нет.
Вид точки	<p>Определяет тип точек, нанесенных на тренд. Точка рисуется в местах излома тренда:</p> <ul style="list-style-type: none"> • Нет • Квадрат • Круг • Треугольник • Ромб • Крест
Размер точки	Определяет размер точек.
Видимость шкалы	Определяет видимость оси пера, если установлен флаг в свойстве Автоматически создавать оси значений в категории свойств Ось Y

Изменения вступят в силу после нажатия на кнопку Применить

Изменение границ оси Y

Важно! Для изменения границ оси Y у пользователя должны быть настроены Права Доступа

Если дважды кликнуть в ячейку столбцов Шк.Мин и Шк.Макс, то откроется диалоговое окно, позволяющее изменить данные настройки:



Изменение ширины столбцов легенды

Ширина столбцов легенды настраивается только в клиенте визуализации. Изменения сохраняются после перезагрузки проекта, если Используется свойство Сохранить состояние в категории свойств Внешний вид

8.3.2.6. ЖУРНАЛ И АРХИВНЫЙ ЖУРНАЛ

Графические элементы Журнал и Архивный журнал служат для отображения в табличном виде сообщений (тревог, созданных в среде разработки, сообщений о нарушении границ и действий пользователей), появляющихся в среде исполнения во время работы. Кроме этого, журналы предоставляют возможность работать с сообщениями, например, квитиовать их. Порядок настройки у обоих элементов совпадает.

Элемент Журнал необходимо использовать для работы с текущими сообщениями. Текущие сообщения в Журнале, по умолчанию, сортируются по полю Время активации. При появлении нового сообщения создается новая строка таблицы Журнала. Количество столбцов таблицы и её внешний вид зависят от настроек Журнала.

Важно! По умолчанию у Журнала настроен фильтр Активные сообщения, а также снят флаг Использовать архив. Изменение этих настроек может привести к некорректной работе элемента.

Элемент Архивный журнал используется для работы с архивными сообщениями. Архивные сообщения в Архивном журнале, по умолчанию, сортируются по полю Время. При каждом изменении состояния сообщения создается новая строка в журнале. Таким образом, по каждому условному сообщению может быть создано три строки: при формировании сообщения (активации), при квитиовании сообщения, и при деактивации сообщения, когда сообщение становится не актуальным.

В режиме исполнения клиент отправляет на сервер запрос данных, в котором указываются все фильтры и другие настройки журналов. На сервере производится фильтрация данных, после чего запрашиваемая информация отправляется клиенту.

В палитре редактора НМІ элементы Журнал и Архивный журнал находятся в категории Контролы.

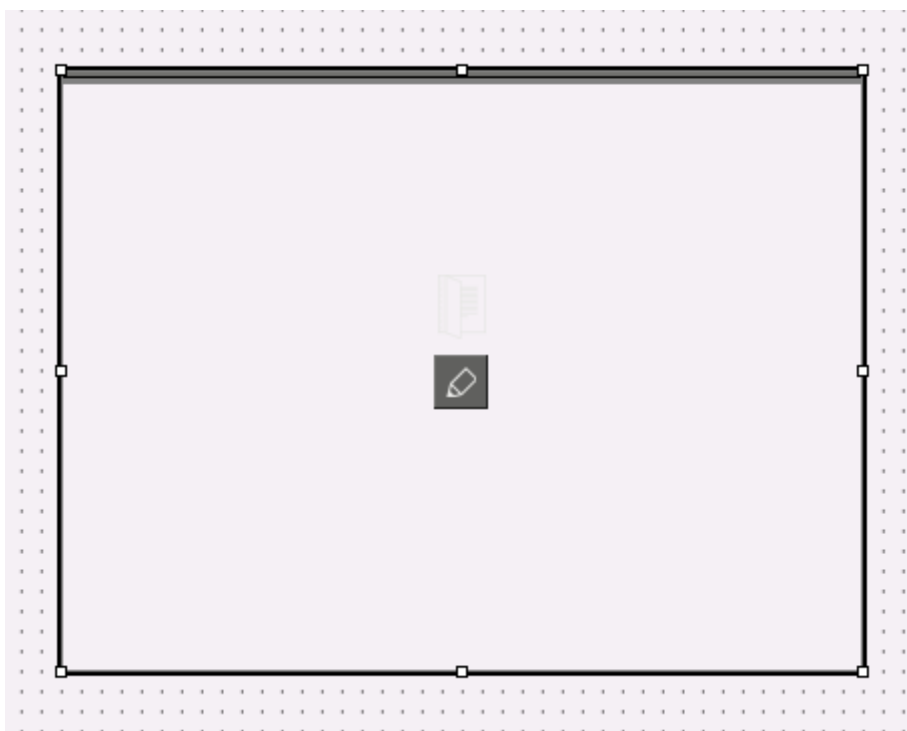
Вид элемента Журнал в палитре:




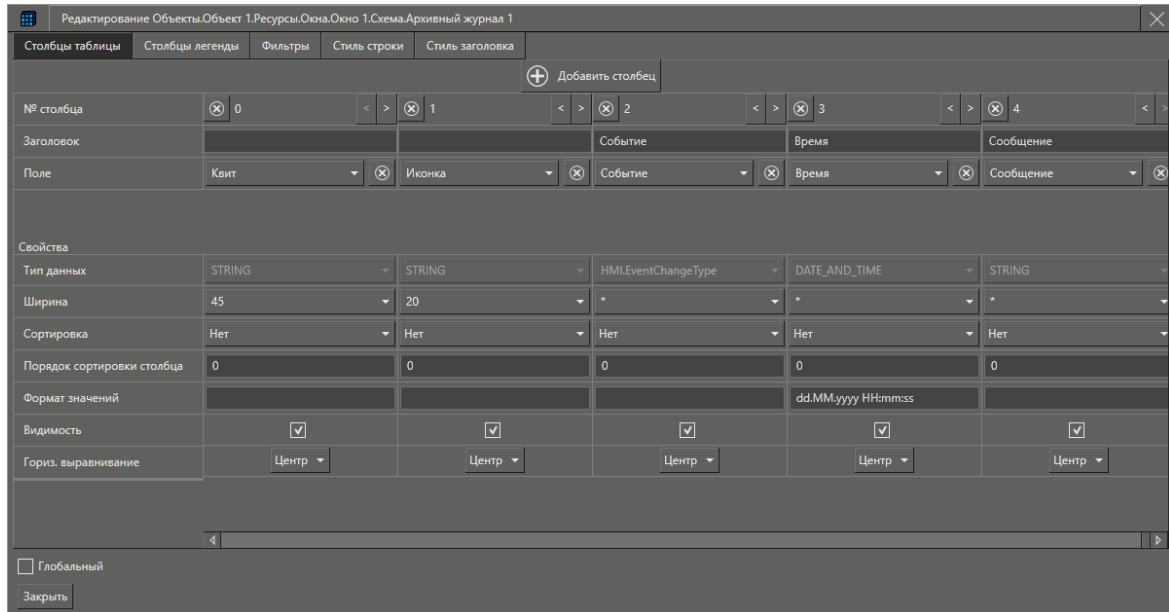
Вид элемента Архивный журнал в палитре:



Вид элементов после добавления в рабочую область редактора НМІ:

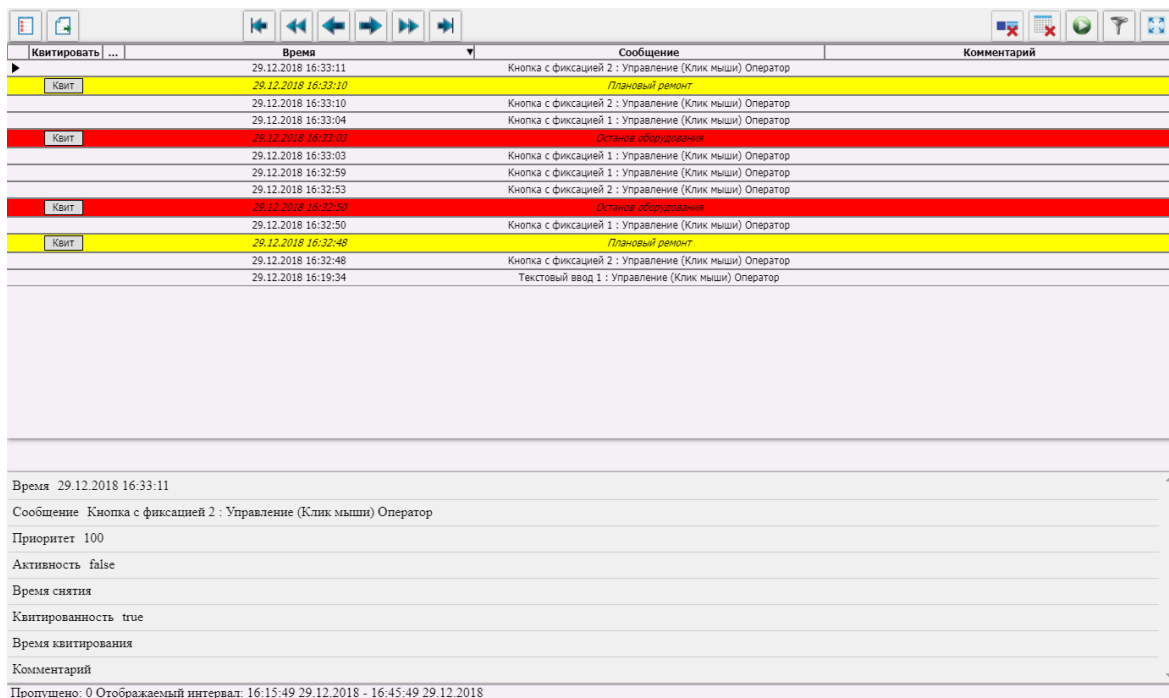


Для того чтобы начать настройку журналов, необходимо нажать кнопку:  либо выполнить двойной клик по элементу, либо нажать клавишу клавиатуры F2. При этом откроется диалоговое окно, в котором настраивается содержимое журнала, а также его внешний вид:



Подробное описание диалогового окна смотрите в разделе [Настройка журнала](#).

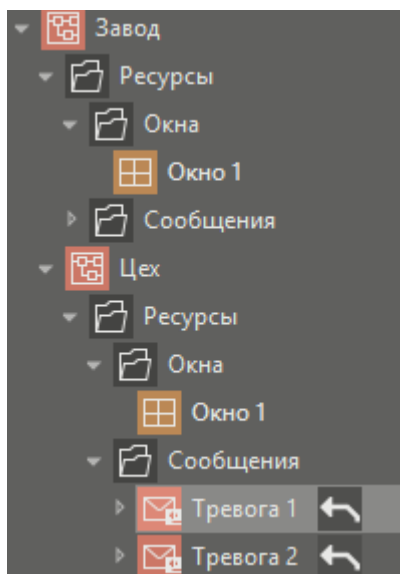
В зависимости от настроек, произведенных в данном окне, в клиенте визуализации журнал будет иметь вид:



Журналы можно настроить таким образом, чтобы в них отображались не все сообщения, а только те, которые удовлетворяют различным условиям (например, отображались сообщения от нужного источника, или сообщения разной важности отображались разным цветом, например, с высоким приоритетом - красным, а с низким - белым).

Источники, сообщения от которых попадают в журнал

Допустим, имеется объект Завод с дочерним объектом Цех, и в обоих объектах созданы Тревоги.



Окна каждого из объектов содержат журналы, у которых флаг Глобальный в окне настройки журнала не отмечен..


В этом случае, журнал, который находится в окне объекта Цех может содержать в себе сообщения только этого объекта. Журнал, расположенный в окне объекта Завод будет содержать в себе сообщения, которые сформировались в обоих объектах. Т.е. для отображения сообщения имеет значение местоположение элемента Журнал в проекте.

Если же требуется, чтобы журнал, расположенный в одном объекте отображал сообщения другого объекта, необходимо в панели свойств журнала настроить свойство Объект. В этом случае местоположение журнала не будет иметь значения.

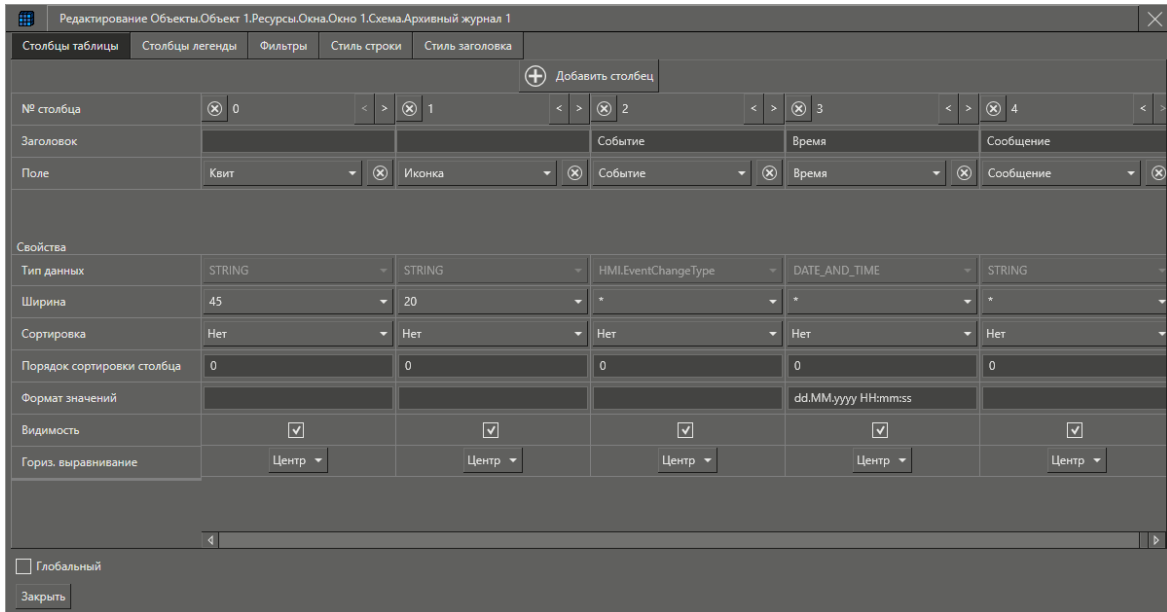
В случае, если флаг Глобальный отмечен, то журнал может отображать сообщения узла, на котором исполняется объект, а также всех других объектов, исполняемых на этом же узле.

В дальнейшем можно сделать выборку из сообщений, отображаемых в журнале, и ввести дополнительные фильтры.

8.3.2.6.1. НАСТРОЙКА ЖУРНАЛА

После добавления журнала в окно можно приступить к его настройке. Начальная настройка производится в диалоговом окне, которое открывается при нажатии на кнопку , находящуюся в центре элемента.

Откроется диалоговое окно, которое содержит пять вкладок:



Назначение вкладок:

Название и ссылка	Назначение
Столбцы таблицы	Служит для определения количество столбцов в журнале, в каждом столбце отображается информация о сообщении.
Столбцы легенды	Определяет количество информации, выдаваемой в легенде журнала.
Фильтры	<p>Дает возможность управлять тем, какие сообщения будут отображаться в журнале в клиенте визуализации в режиме исполнения.</p> <p>Для того чтобы обеспечить возможность в режиме исполнения выбирать, какие сообщения будут отображаться в клиенте визуализации, разработчик проекта должен создать необходимое количество фильтров.</p>
Стиль строки	Определяет внешний вид строки (фона, текста) журнала в клиенте визуализации. Внешний вид строки сообщения может зависеть от значений свойств (параметров) тревоги.
Стиль заголовка	Определяет внешний вид заголовка журнала.

Флаг Глобальный определяет, какие сообщения будут попадать в текущий журнал. Если флаг отмечен, то в журнале будут отображаться все сообщения узла. Если флаг снят, то содержимое журнала зависит от его местоположения в проекте и от настроек панели свойств. Подробнее этот вопрос рассмотрен в разделе Журнал.

После нажатия на кнопку Закрыть окно настройки журнала будет закрыто.

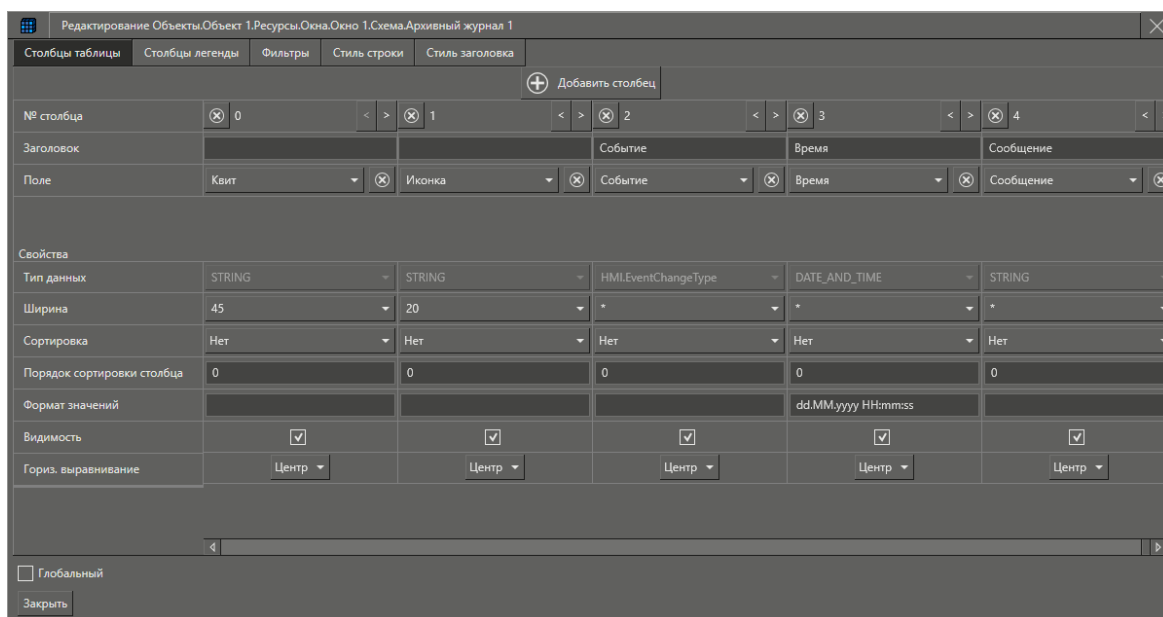
В панели свойств журнала, как правило, настраивается период обновления данных. При необходимости, некоторые свойства можно использовать для создания собственных инструментов управления журналом, например, для настройки интервала отображаемых сообщений и т.п.

8.3.2.6.1.1. ОКНО НАСТРОЙКИ ЖУРНАЛА. СТОЛБЦЫ ТАБЛИЦЫ

Настройки, описанные в данном разделе, применимы к элементам Журнал и Архивный журнал.

Вкладка Столбцы таблицы определяет количество, внешний вид и содержимое столбцов журнала.

Вкладка имеет вид :







При помощи кнопки Добавить столбец задается количество столбцов. Содержимое столбцов определяется в таблице окна настройки журнала.

В случае если все добавленные столбцы не помещаются в видимой части таблицы, в ее нижней части появляется линейка прокрутки.

Важно! Количество столбцов журнала не влияет на архив сообщений. Столбцы могут быть добавлены/удалены в любой момент разработки проекта.

Для каждого столбца можно определить следующие свойства:

Название	Описание
№ столбца	<p>Определяет порядок столбцов. В клиенте визуализации порядок столбцов будет таким же, как и в данной таблице.</p> <p>Чтобы переместить столбец влево или вправо следует воспользоваться соответствующими стрелками -  .</p> <p>Нажатие на кнопку  приведет к удалению столбца.</p>
Заголовок	<p>Определяет текст заголовка столбца журнала. Существующий текст заголовка можно редактировать после двойного нажатия на него левой кнопкой мыши :  Квितिp</p>
Поле	<p>Определяет содержимое столбца, т.е. информацию, которая будет отображаться в данном столбце, например, время возникновения сообщения или тревоги, источник сообщения и др. Список возможных вариантов значения Поля смотрите ниже. Если для каких-то сообщений данное поле не применимо, то ячейка останется пустой.</p>
Свойства	
Тип данных	<p>Показывает тип данных свойства Поле. Определяется автоматически после настройки строки Поле. Носит информативный характер и не может быть изменен при помощи окна настройки журнала.</p>
Ширина	<p>Определяет ширину столбца журнала. Возможны следующие варианты значения свойства:</p> <ul style="list-style-type: none"> • 20 - числовое значение ширины столбца (в px). Может принимать любое значение. Если сумма ширин всех столбцов, окажется больше ширины журнала, то в клиенте визуализации в нижней части таблицы сообщений появится линейка прокрутки. • Auto - ширина столбца будет определяться автоматически, в зависимости от содержимого заголовка столбца. Если сумма ширин всех столбцов, окажется больше ширины журнала, то в клиенте визуализации в нижней части таблицы сообщений появится линейка прокрутки.

Название	Описание
	<ul style="list-style-type: none"> • Символы * - в этом случае ширина столбца будет зависеть от размера экрана монитора и рассчитываться автоматически. Если у одних столбцов будет задан один символ *, а у других - 2*, то вторые будут в два раза шире первых. Если сумма ширин столбцов, значения которых заданы числовыми настройками или Auto, окажется больше ширины журнала, то в клиенте визуализации в нижней части таблицы сообщений появится линейка прокрутки, а на столбцы, у которых ширина задана символами *, будет выделено минимальное количество пикселей, достаточное только для того, чтобы обеспечить возможность растягивания таких столбцов в режиме исполнения для просмотра их содержимого.
Сортировка	<p>Определяет тип сортировки в столбце в таблице сообщений в клиенте визуализации по умолчанию. Может принимать значения:</p> <ul style="list-style-type: none"> • Нет - сортировка в столбце не используется • По возрастанию - сортировка в столбце используется, первыми в таблице располагаются сообщения, у которых поле имеет меньшее значение. • По убыванию - сортировка в столбце используется, первыми в таблице располагаются сообщения, у которых свойство имеет большее значение <p>Поддерживается только в элементе Журнал.</p>
Порядок сортировки столбцов	<p>Определяет приоритет сортировки столбцов в случае, когда необходима сортировка по нескольким столбцам (множественная сортировка).</p> <p>Поддерживается только в элементе Журнал.</p>
Формат значений	<p>Определяет формат отображения выводимого значения. Описание этого свойства смотрите в разделе Свойство формат значений. Например, можно задать это свойство таким образом, чтобы в столбце таблицы, имеющим тип данных DT,</p>

Название	Описание
	<p>отображались дата и время с точностью до миллисекунды: dd.MM.yyyy HH:mm:ss.fff.</p> <p>Формат значений можно редактировать после двойного нажатия на поле этого свойства левой кнопкой мыши.</p> <p>Если для столбца Квит в этом свойстве задано, например, dd.MM.yyyy HH:mm:ss, то в нем будут отображаться дата и время квитирования (только для НМІ v2).</p>
Видимость	<p>Определяет видимость столбца в таблице сообщений. Как правило, в данном окне не настраивается. В том случае, если требуется управлять видимостью, то необходимо задинамизировать нужное свойство в панели свойств журнала.</p>
Гориз. выравнивание	<p>Определяет положение текста в столбце. Возможные варианты:</p> <ul style="list-style-type: none"> • Центр • Лево • Право

Возможные значения свойства Поле по умолчанию.

Значение (Программное имя поля тревоги)	Описание	Тип сообщений
Квит	<p>Этот столбец дает возможность пользователю квитировать сообщение, а также оценить, какие сообщения требуют квитирования.</p> <p>Если для этого столбца в свойстве Формат значений задано, например, dd.MM.yyyy HH:mm:ss, то в нем будут отображаться дата и время квитирования (только для НМІ v2).</p>	Условные сообщения
Иконка	<p>Отображает графическое изображение сообщения.</p> <p>Для того чтобы назначить иконку Тревоге, необходимо в контекстном меню библиотечного</p>	Условные сообщения

Значение (Программное имя поля тревоги)	Описание	Тип сообщений
	элемента выбрать пункт Дерево.Установить иконку. В этом случае у всех экземпляров, добавленных в дерево объектов, будет отображаться соответствующая иконка.	
Полное имя объекта (EventFullObject Name)	Отображает путь к родительскому объекту источника сообщения в проекте. Например, Объекты. Завод. Цех 1.Линия	Все
Объект (Object)	Отображает имя родительского объекта источника сообщения. Например, Линия	Все
Источник	Отображает имя источника сообщения. Если сообщение появилось в результате нарушения границ, то отобразится имя параметра (тега, канала), для которого назначена шкала	Все
Состояние	<p>Отображает, в каком состоянии находится сообщение.</p> <p>Возможны следующие варианты:</p> <p>0 - не активно, не квитировано;</p> <p>1 - не активно, квитировано;</p> <p>2 - активно, не квитировано;</p> <p>3 - активно, квитировано.</p> <p>Соответственно, если состояние сообщение изменится (например, если пользователь квитировал сообщение), то содержимое данного столбца тоже изменится.</p> <p>Для безусловных сообщений значение столбца всегда будет равно 1.</p>	Все

Значение (Программное имя поля тревоги)	Описание	Тип сообщений
Время активации (ActiveTime)	Отображает время появления сообщения.	Все
Пользователь	Отображает имя пользователя, который квитирует сообщение. Если сообщение появилось в журнале в результате действия пользователя, то в данном поле будет отображаться, какой пользователь выполнил это действие, например, нажал кнопку.	Все
IP-адрес	Отображает IP-адрес устройства, с которого пользователь квитирует сообщение. Если сообщение появилось в журнале в результате действия пользователя, то в данном поле будет отображаться IP-адрес устройства, с которого пользователь выполнил действие, например, открыл окно.	Все
Событие	<p>Отображает причину возникновения новой строки в Архивном журнале. При каждом изменении состояния сообщения формируется новая строка в журнале. Таким образом, по каждому условному сообщению может быть сформировано три строки - при формировании сообщения (активации), при квитировании сообщения, и при деактивации сообщения, т.е когда сообщение становится не актуальным.</p> <p>Возможны следующие значения свойства:</p> <ol style="list-style-type: none"> 1. Появление 2. Исчезновение 3. Квитирование 	Все

Значение (Программное имя поля тревоги)	Описание	Тип сообщений
Сообщение (Message)	Отображает текст сообщения. У пользовательских тревог в журнал попадает значение свойства Сообщение.	Все
Приоритет (Severity)	<p>Отображает приоритет сообщения. У пользовательских тревог значение приоритета настраивается. Предопределенные сообщения имеют следующие приоритеты:</p> <p>100 - сообщение о действиях пользователя</p> <p>750 - сообщения о нарушении аварийных границ - значение установленное по умолчанию, его можно изменить в настройках шкалы</p> <p>500 - сообщения о нарушении предупредительных границ - значение установленное по умолчанию, его можно изменить в настройках шкалы</p>	Все
Активность (Active)	Указывает, активно ли сообщение в настоящий момент. Может принимать значение TRUE и False. Для пользовательских тревог принимает значение свойства Активность. Для сообщений, появляющихся в результате нарушения границ, значение столбца будет равно TRUE, пока условия соблюдаются, и False, когда значение контролируемого параметра выйдет за указанную границу.	Условные сообщения
Время деактивации (InactiveTime)	Отображает момент времени, в который сообщение потеряло свою актуальность.	Условные сообщения
Квитированность (Acked)	Отображает, было ли квитировано сообщение. Если сообщение было квитировано пользователем в журнале или программно (изменено свойство пользовательской тревоги), то значение ячейки	Все

Значение (Программное имя поля тревоги)	Описание	Тип сообщений
	<p>столбца будет TRUE, а если сообщение не было квити́ровано, то False.</p> <p>Для сообщений о действиях пользователей это свойство всегда принимает значение TRUE</p>	
Время квити́рования (AckedTime)	Отображает момент времени, в который сообщение было квити́ровано.	Условные сообщения
Комментарий (Comment)	Отображает значение свойства Комментарий.	Тревога
HiHi	Отображает значение заданной границы HiHi в момент возникновения сообщения	Сообщения о нарушении границ
Hi	Отображает значение заданной границы Hi в момент возникновения сообщения	Сообщения о нарушении границ
Lo	Отображает значение заданной границы Lo в момент возникновения сообщения	Сообщения о нарушении границ
LoLo	Отображает значение заданной границы LoLo в момент возникновения сообщения	Сообщения о нарушении границ
Значение	Отображает значение контролируемого параметра в момент возникновения сообщения.	Сообщения о нарушении границ
Максимальная скорость изменения	Отображает значение заданного свойства Максимальная скорость изменения в момент возникновения сообщения.	Сообщения о превышении

Значение (Программное имя поля тревоги)	Описание	Тип сообщений
		скорости изменения
Время (ChangeTime)	Отображает момент времени, в который произошло изменение состояния сообщения.	Все

Полный перечень программных имен параметров тревоги можно найти в разделе Список программных имен параметров тревог

В случае, если разработчик проекта в библиотеке создаст Тревогу, которая кроме стандартных параметров будет иметь и дополнительные параметры (свойства), то в Поле появятся и дополнительные значения, соответствующие новым параметрам.

Важно! Если разработчик работает с тревогами в программе, то необходимо использовать программные имена полей тревог

8.3.2.6.1.2. ОКНО НАСТРОЙКИ ЖУРНАЛА. СТОЛБЦЫ ЛЕГЕНДЫ

Настройки, описанные в данном разделе, применимы к элементам Журнал и Архивный журнал.

Вкладка Столбцы легенды определяет содержимое и внешний вид легенды журнала в клиенте визуализации.

Столбцы таблицы	Столбцы легенды	Фильтры	Стиль строки	Стиль заголовка	Добавить столбец				
Заголовок	<input checked="" type="checkbox"/> Время	<input checked="" type="checkbox"/> Сообщение	<input checked="" type="checkbox"/> Приоритет	<input checked="" type="checkbox"/> Время активации	<input checked="" type="checkbox"/> Время снятия	<input checked="" type="checkbox"/> Время квитирова	<input checked="" type="checkbox"/> Пользователь	<input checked="" type="checkbox"/> Комментарий	
Поле	Время ▾ <input checked="" type="checkbox"/>	Сообщение ▾ <input checked="" type="checkbox"/>	Приоритет ▾ <input checked="" type="checkbox"/>	Время акти ▾ <input checked="" type="checkbox"/>	Время деак ▾ <input checked="" type="checkbox"/>	Время квит ▾ <input checked="" type="checkbox"/>	Пользовате ▾ <input checked="" type="checkbox"/>	Комментар ▾ <input checked="" type="checkbox"/>	
Тип поля	DATE_AND_TIME ▾	STRING ▾	DINT ▾	DATE_AND_TIME ▾	DATE_AND_TIME ▾	DATE_AND_TIME ▾	STRING ▾	STRING ▾	
Формат значений	dd.MM.yyyy HH:mm:ss	нет текста	нет текста	dd.MM.yyyy HH:mm:ss	нет текста	нет текста	нет текста	нет текста	
Видимость	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Гориз. выравнивание	▾	▾	▾	▾	▾	▾	▾	▾	
	< >	< >	< >	< >	< >	< >	< >	< >	

Настройка легенды журнала соответствует настройке таблицы сообщений.

В легенде по каждому сообщению может отображаться любая информация, представленная в свойстве Поле.

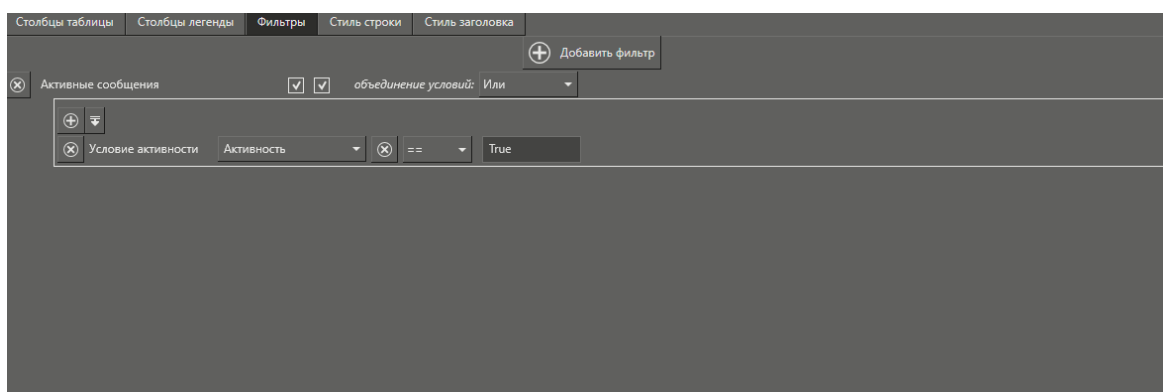
8.3.2.6.1.3. ОКНО НАСТРОЙКИ ЖУРНАЛА. ФИЛЬТРЫ

Настройки, описанные в данном разделе, применимы к элементам Журнал и Архивный журнал.

Вкладка **Фильтры** предназначена для настройки фильтров, которые могут быть использованы в клиенте визуализации.

Местоположение журнала, значение свойства **Объект** в панели свойств, а также состояние флага **Глобальный** определяют, какие сообщения, в принципе, смогут отображаться в журнале. Для того, чтобы обеспечить выбор в клиенте визуализации только части из них, разработчику необходимо настроить вкладку **Фильтры**. В этом случае, при нажатии в клиенте визуализации на кнопку **Показать фильтр** откроется список созданных фильтров.

По умолчанию, вкладка имеет вид:










По умолчанию, данная вкладка содержит фильтр **Активные сообщения**. Это означает, что если фильтр включен, то в журнале будут отображаться только те сообщения, которые актуальны в настоящий момент. Если фильтр выключен, то будут отображаться все сообщения.

Для того чтобы добавить новый фильтр, необходимо нажать на кнопку **Добавить фильтр**.

Появится группа настроек:



Элемент	Назначение
	Удаляет существующий фильтр
	Название фильтра. Для его изменения необходимо дважды нажать левой кнопкой мыши на существующее название
	Флаги, определяющие состояние фильтров по умолчанию и возможность управления фильтрами в клиенте визуализации. Левый флаг определяет, включен или выключен фильтр по







Элемент	Назначение
	<p>умолчанию после запуска клиента визуализации. Правый флаг определяет наличие возможности включения и выключения фильтра. Если флаг снят, то управлять использованием этого фильтра в клиенте визуализации невозможно.</p>
	<p>Фильтр срабатывает когда сообщения удовлетворяют условиям. Данное поле определяет работу фильтра том в случае, если условий несколько. Возможные варианты:</p> <p>Или - фильтр работает при выполнении хотя бы одного условия.</p> <p>И - фильтр работает только при выполнении всех условий.</p>
	<p>Инструменты добавления условий срабатывания фильтра. Существуют следующие типы условий:</p> <p>Сравнение - позволяет сравнить любое свойство сообщения, описанное в Поле, с каким-либо значением. Данное условие используется чаще остальных.</p> <p>Проверка типа сообщения - сообщение будет отображено в случае, если оно соответствует нужному типу.</p> <p>Проверка отношения - сообщение будет отображено в случае, если оно сформировалось в нужном узле, объекте, теге и т.п.</p> <p>При нажатии на кнопку  добавится условие Сравнение. При нажатии на кнопку  появится выбор из всех трех вариантов.</p>

Работа с условием Сравнение

После добавления условия появится строка:



На примере показано условие фильтра, при котором в журнале будут отображаться только те сообщения, которые активны и не были квитированы.

Элемент	Назначение
	Удаляет существующее условие.
	Название условия. Для его изменения необходимо дважды нажать левой кнопкой мыши на существующее название.
	Выпадающий список содержит все поля (характеристики) сообщения, причем поле может быть использовано в фильтре даже в том случае, если это поле не добавлено ни в Легенду, ни в Таблицу сообщений. Кнопка  очистит содержимое свойства.
	<p>Определяет, какая функция сравнения будет использована.</p> <p>Возможные варианты:</p> <p>Содержит - условие сработает, если выбранная характеристика сообщения содержит в себе символы, введенные в соседнее поле;</p> <p>== - условие сработает, если выбранная характеристика сообщения равна значению, введенному в соседнее поле;</p> <p>> - условие сработает, если выбранная характеристика сообщения больше значения, введенного в соседнее поле;</p> <p><- условие сработает, если выбранная характеристика сообщения меньше значения, введенного в соседнее поле;</p> <p>>= - условие сработает, если выбранная характеристика сообщения больше или равна значению, введенному в соседнее поле;</p> <p><= - условие сработает, если выбранная характеристика сообщения меньше или равна значению, введенному в соседнее поле;</p> <p>!= - условие сработает, если выбранная характеристика сообщения не равна значению, введенному в соседнее поле.</p>
	Поле, в которое вводится значение, которое сравнивается с выбранной характеристикой сообщения.

Работа с условием Проверка типа сообщения

После добавления условия появится строка:



В поле строки необходимо перетащить нужный тип сообщения из библиотеки. Если сообщения были сформированы при помощи ФБ FireBaseEvent или его наследников, то в поле строки нужно перетащить этот ФБ или его наследник.

Работа с условием Проверка отношения

После добавления условия появится строка:



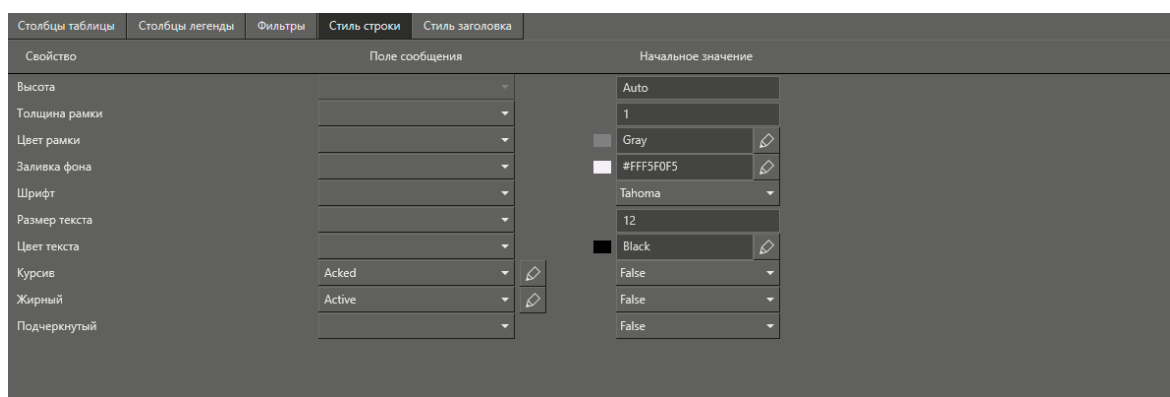
В поле строки необходимо перетащить объект, тег, канал, узел, т.е. тот элемент, сообщения которого необходимо отображать при использовании этого фильтра.

8.3.2.6.1.4. ОКНО НАСТРОЙКИ ЖУРНАЛА. СТИЛЬ СТРОКИ

Настройки, описанные в данном разделе, применимы к элементам Журнал и Архивный журнал.

Вкладка Стиль строки определяет внешний вид строки журнала. При этом любое свойство строки может меняться в зависимости от того, какое сообщение в ней располагается. Так, например, по умолчанию вкладка настроена таким образом, что все активные сообщения отображаются жирным шрифтом, а не квитируемые сообщения отображаются курсивом.

Вкладка имеет вид :



На вкладке находится таблица, состоящая из трех столбцов:

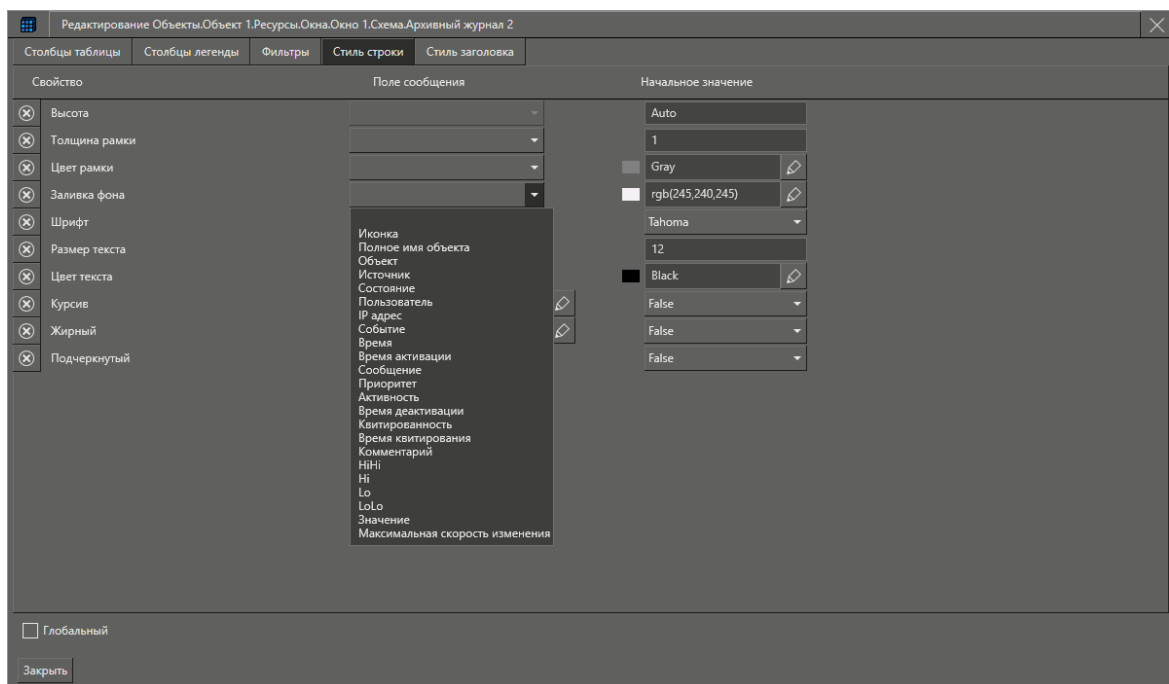
Название	Назначение
Свойство	Название свойства строки.


Название	Назначение
Поле сообщения	Служит для настройки зависимости свойства строки от поля (характеристики) сообщения. В этом случае свойство будет меняться в режиме исполнения: для одних сообщений оно будет принимать одно значение, а для других - другое.
Начальное значение	Служит для настройки свойства строки в случае, когда его не требуется менять в режиме исполнения.

Пример настройки стиля строки

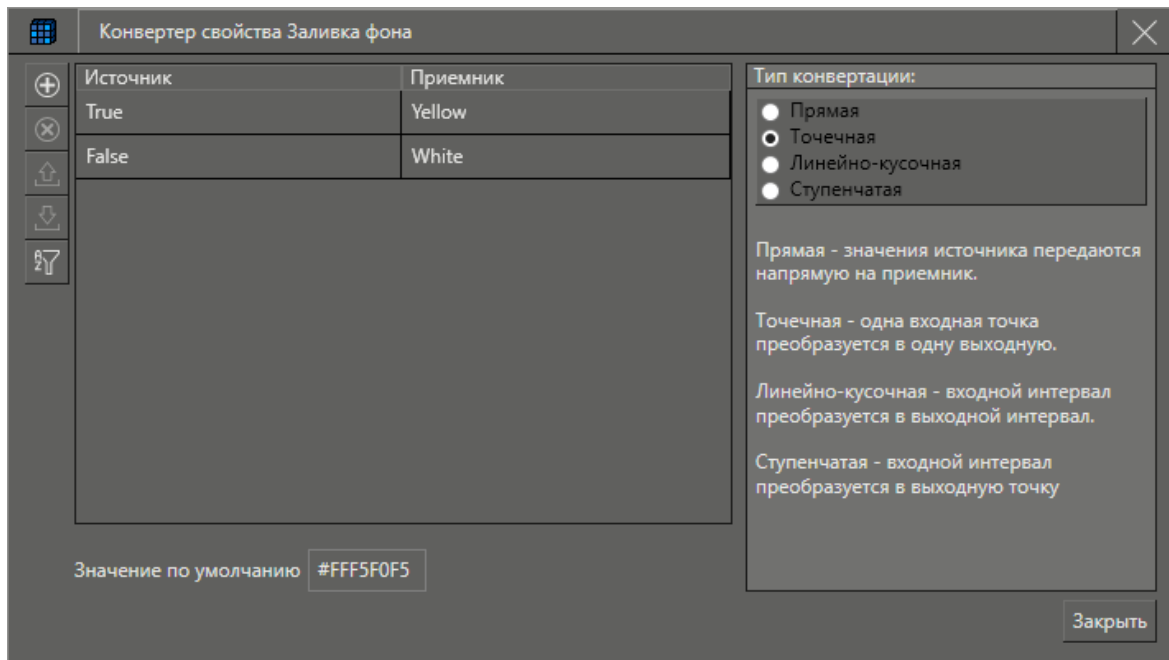
Допустим, что необходимо активные сообщения отображать на желтом, а сообщения, уже потерявшие свою актуальность, - на белом.

В этом случае необходимо настроить свойство Заливка фона. В выпадающем списке Поле сообщения следует выбрать поле Активность (Active):



Для настройки зависимости цвета фона строки от поля сообщения Активность необходимо нажать на кнопку .

Откроется диалоговое окно Конвертер значений, где в столбце Источник задаются значения, которые может принимать поле, а в столбце Приемник задаются значения, которые должно принимать свойство строки. Например:

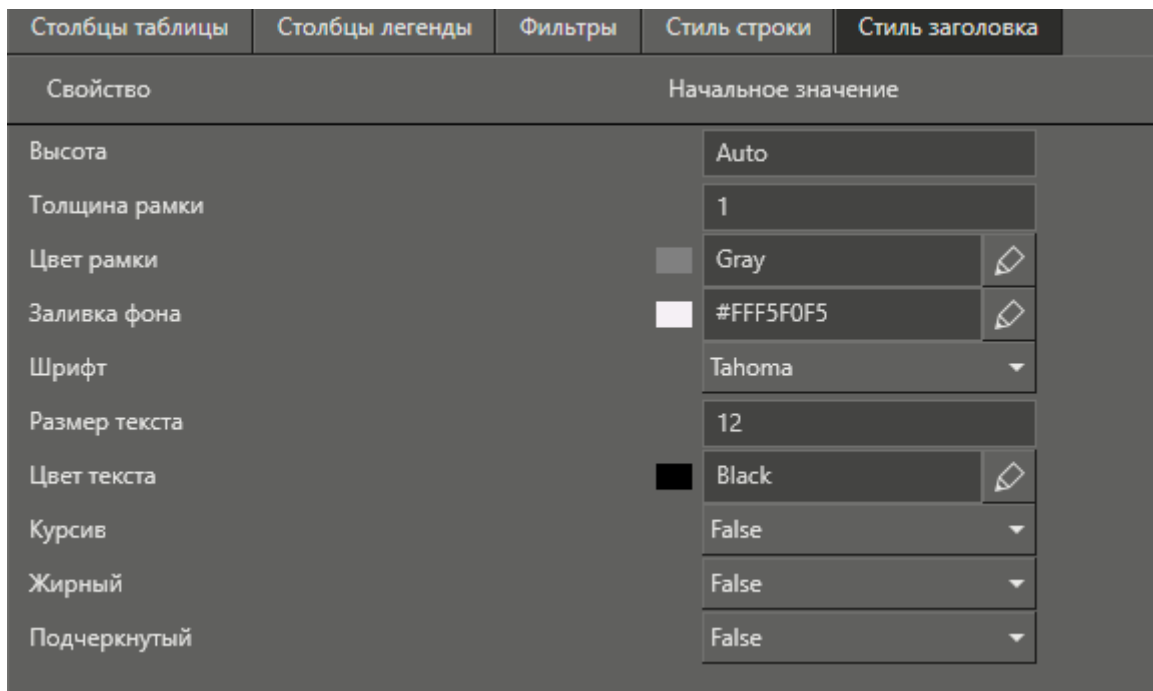


8.3.2.6.1.5. ОКНО НАСТРОЙКИ ЖУРНАЛА. СТИЛЬ ЗАГОЛОВКА

Настройки, описанные в данном разделе, применимы к элементам Журнал и Архивный журнал.

Вкладка **Стиль заголовка** определяет внешний вид заголовка таблицы сообщений.

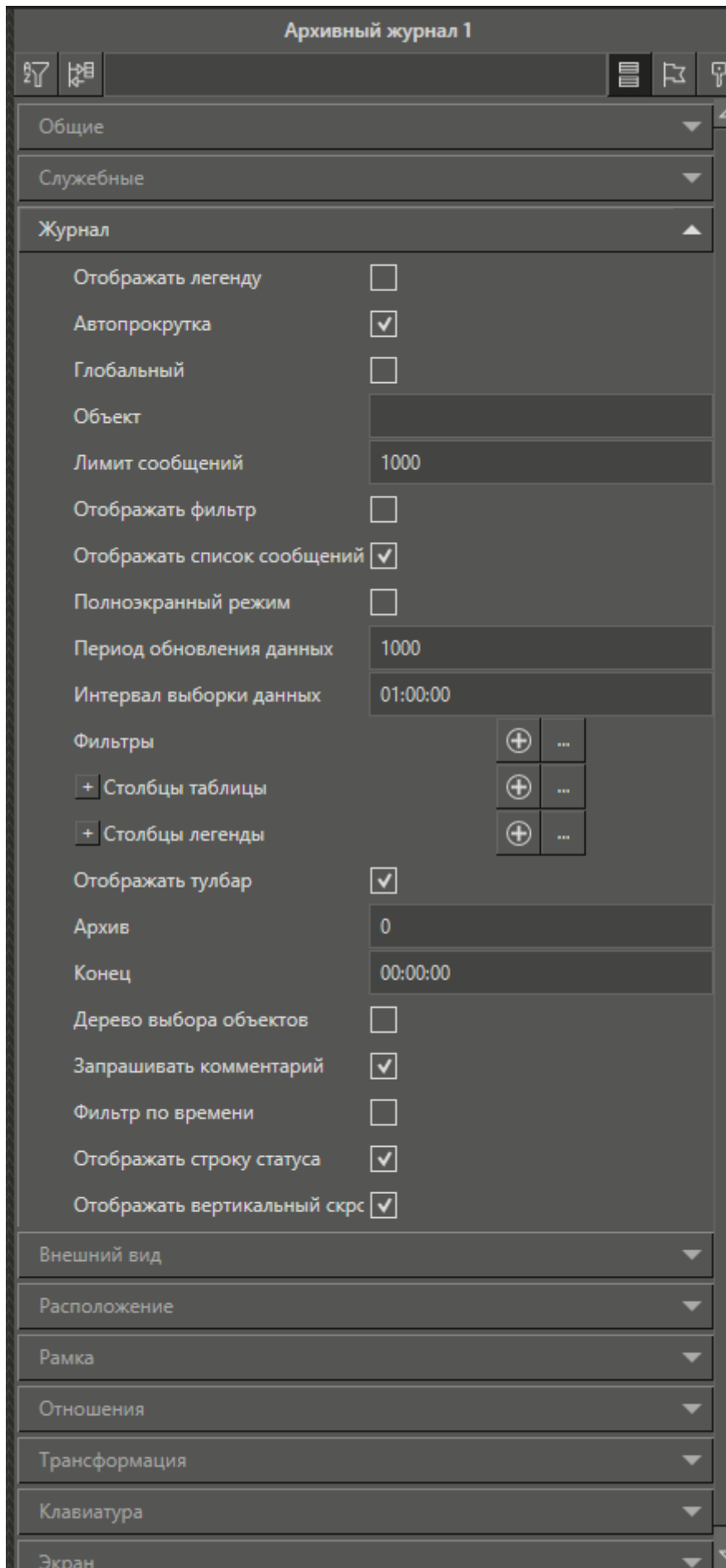
Вкладка имеет вид:



В столбце **Начальное значение** задаются значения параметров строки.




8.3.2.6.1.6. СВОЙСТВА ЖУРНАЛА

Вид панели свойств элементов Журнал и Архивный журнал :









Многие Свойства журнала, представленные в панели свойств, удобнее настраивать в диалоговом окне настройки журнала.

Описание основных свойств графического элемента Журнал:

Название	Описание
Категория Внешний вид	
Заливка легенды	Определяет цвет фона легенды журнала. При нажатии на кнопку  , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.
Заливка тулбара	Определяет цвет фона панели инструментов журнала. При нажатии на кнопку  , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.
Заливка фильтров	Определяет цвет фона панели фильтров журнала. При нажатии на кнопку  , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.
Категория Журнал	
Отображать легенду	Определяет, будет ли отображаться легенда после открытия окна, содержащего журнал. В клиенте визуализации имеется возможность включать/выключать легенду, используя панель инструментов журнала. Кроме того, данное свойство может быть задинамизировано, т.е. зависеть от других элементов управления, созданных разработчиком проекта, либо от какой-то программы.
Автопрокрутка	Определяет будет ли включена автопрокрутка журнала после открытия окна, содержащего журнал. В клиенте визуализации имеется возможность включать/выключать автопрокрутку, используя панель инструментов журнала. Кроме того, данное свойство может быть задинамизировано, т.е. зависеть от других элементов управления, созданных разработчиком проекта, либо от какой-то программы.

Название	Описание
Глобальный	<p>Определяет, сообщения от каких источников будут попадать в журнал. Если флаг отмечен, то в журнале могут отображаться все сообщения, которые будут формироваться в узле, а также в объектах, исполняемых в нем. Если флаг снят, то в журнале будут отображаться либо сообщения элемента (объекта, тега, канала, узла), в котором находится окно, содержащее журнал, либо того элемента, с которым связано свойство журнала Объект .</p>
Объект	<p>Определяет сообщения от какого источника будут попадать в журнал. Если флаг Глобальный снят, а в данное поле будет перетащен какой-либо элемент проекта (объект, тег, канал), то в журнале станут отображаться сообщения этого элемента, независимо от местоположения журнала.</p>
Лимит сообщений	<p>Задаёт максимальное число сообщений, извлекаемое из БД (только для НМІ v2).</p>
Отображать фильтр	<p>Определяет будет ли отображаться панель фильтров после открытия окна, содержащего журнал. В клиенте визуализации имеется возможность включать/выключать панель фильтров, используя панель инструментов журнала . Кроме того, данное свойство может быть задинамизировано, т.е. зависеть от других элементов управления, созданных разработчиком проекта, либо от какой-то программы.</p>
Отображать список сообщений	<p>Определяет, будет ли отображаться таблица сообщений после открытия окна, содержащего журнал. В клиенте визуализации имеется возможность включать/выключать таблицу сообщений, используя панель инструментов журнала. Кроме того, данное свойство может быть задинамизировано, т.е. зависеть от других элементов управления, созданных разработчиком проекта, либо от какой-то программы.</p>
Полноэкранный режим	<p>Определяет, будет ли отображаться элемент журнал в полноэкранном режиме после открытия окна, содержащего журнал. В клиенте визуализации имеется возможность включать/выключать полноэкранный режим, используя панель инструментов журнала. Кроме того, данное свойство может</p>

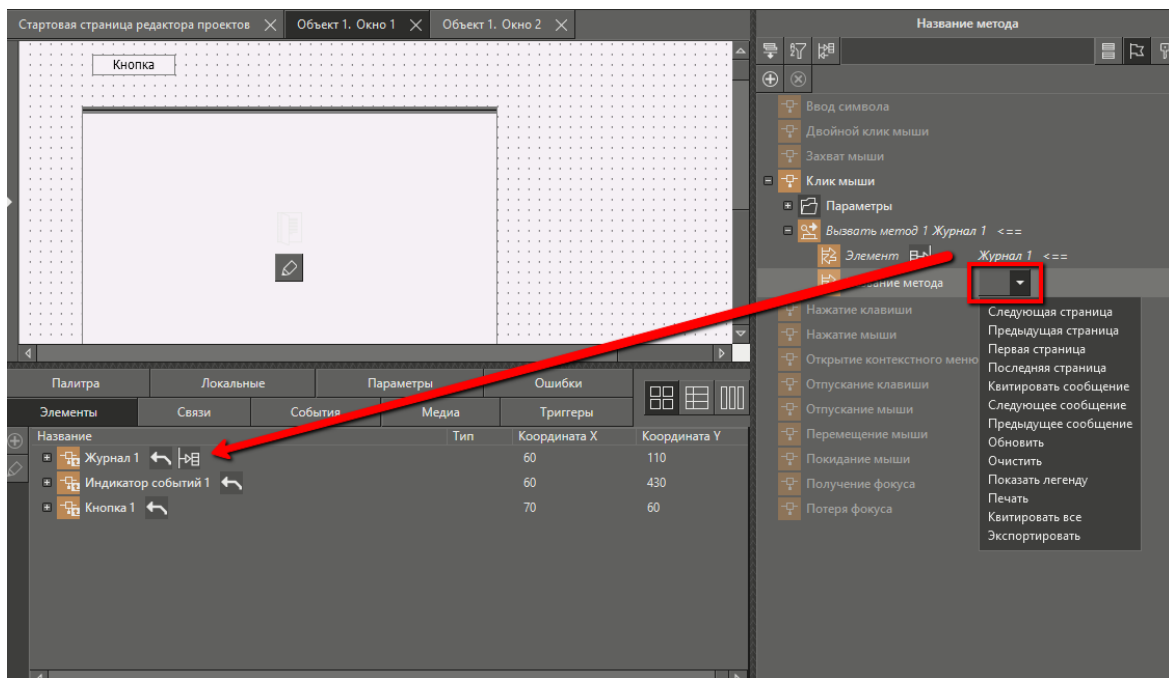
Название	Описание
	<p>быть задинамизировано, т.е. зависеть от других элементов управления, созданных разработчиком проекта, либо от какой-то программы.</p>
<p>Период обновления данных</p>	<p>Задаёт период автоматического обновления данных при включенной автопрокрутке.</p>
<p>Интервал выборки данных</p>	<p>Задаёт интервал, который будет отображаться на одной странице журнала. Свойство может быть задинамизировано, т.е. зависеть от других элементов управления, созданных разработчиком проекта, либо от какой-то программы.</p>
<p>Фильтры</p>	<p>Группа настроек, отображающих ранее созданные фильтры в окне настройки журнала. При помощи кнопок  можно добавлять новые фильтры и условия в них, а также изменять настройки ранее созданных фильтров. Рекомендуется настраивать фильтры журнала при помощи вкладки Фильтры окна настройки журнала, которая откроется при нажатии на кнопку .</p>
<p>Столбцы таблицы</p>	<p>Группа настроек, отображающих ранее созданные столбцы в окне настройки журнала. При помощи кнопки  можно добавлять новые столбцы, а также изменять настройки ранее созданных столбцов. Рекомендуется настраивать столбцы журнала при помощи вкладки Столбцы таблицы окна настройки журнала, которая откроется при нажатии на кнопку .</p>
<p>Столбцы легенды</p>	<p>Группа настроек, отображающих ранее созданные строки в окне настройки журнала. При помощи кнопки  можно добавлять новые строки, а также изменять настройки ранее созданных строк. Рекомендуется настраивать легенду журнала при помощи вкладки Столбцы легенды окна настройки журнала, которая откроется при нажатии на кнопку .</p>
<p>Отображать тулбар</p>	<p>Определяет, будет ли отображаться панель инструментов журнала. Если флаг снят, и панель инструментов не</p>

Название	Описание
	отображается, то разработчик проекта должен создать свои собственные элементы управления журналом.
Использовать архив сообщений	Определяет источник, из которого будет осуществляться выборка сообщений. Если флаг снят, то источником будут сообщения, которые находятся в оперативной памяти. Если флаг отмечен, то для работы с журналом сервер, помимо сообщений из оперативной памяти, будет использовать архив сообщений из базы данных.
Конец	Определяет время конца интервала, за который запрашиваются сообщения с сервера. Свойство может быть задинамизировано, т.е. зависеть от других элементов управления, созданных разработчиком проекта, либо от какой-то программы. Если свойство не задинамизировано, то управлять концом интервала в режиме исполнения можно, используя панель инструментов. Кроме того, разработчик может создать свои собственные инструменты управления с настроенными методами
Запрашивать комментарий	Определяет необходимость ввода комментария во время квитирования сообщения. Если флаг отмечен, то откроется диалоговое окно для ввода комментария, который будет записан в базу данных сообщений (только для НМІ v2).
Дерево выбора объектов	Позволяет без добавления специальных фильтров выбрать нужные для отображения источники сообщений. Если флаг установлен, то в панели инструментов появится иконка Дерево объекта
Фильтр по времени	Позволяет загрузить из базы данных только те сообщения, которые сформировались в указанный интервал времени. Если флаг установлен, то в клиенте визуализации в панели инструментов появится кнопка  .
Отображать строку статуса	Определяет видимость строки статуса в клиенте визуализации

Название	Описание
Отображать вертикальный скроллбар	Определяет видимость вертикальной линейки прокрутки в клиенте визуализации

8.3.2.6.1.7. МЕТОДЫ ДЛЯ РАБОТЫ С ЖУРНАЛОМ

Для работы с журналом можно создать собственные элементы управления, например кнопки, и настроить события, например, в событие Клик мыши добавить действие Вызвать метод, где указать какое действие с журналом должно выполняться при нажатии на элемент управления:



Возможные методы:

Название	Описание
Следующая страница	Переходит на следующую страницу журнала.
Предыдущая страница	Переходит на предыдущую страницу журнала.
Первая страница	Открывает первую страницу журнала, т.е. самые ранние сообщения в архивном журнале, либо самые ранние сообщения в журнале.

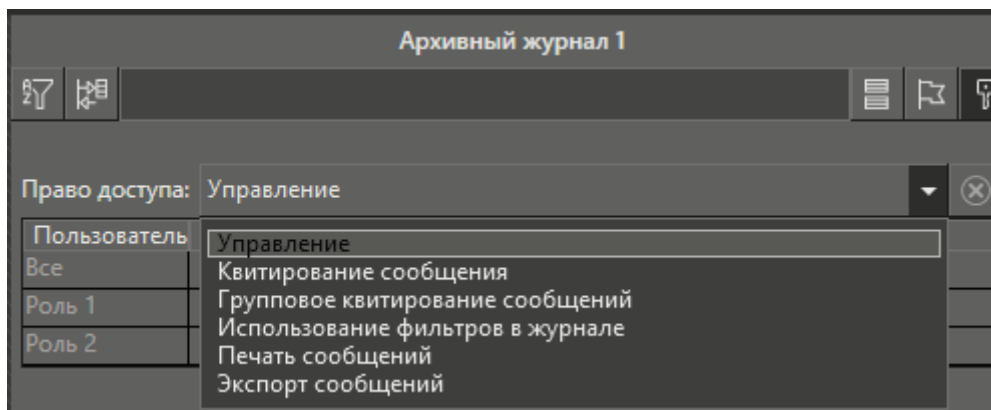
Название	Описание
Последняя страница	Открывает последнюю страницу журнала.
Квитировать сообщение	Квитирует выделенное сообщение.
Следующее сообщение	Переводит курсор на следующую строку сообщения (на строку вверх).
Предыдущее сообщение	Переводит курсор на предыдущую строку сообщения (на строку вниз).
Обновить	
Очистить	Очищает содержимое журнала (работает только для текущего журнала).
Показать легенду	Отображает легенду журнала.
Печать	Производит печать журнала.
Квитировать все	Квитирует все видимые сообщения на странице.
Экспортировать	Производит экспорт загруженных сообщений журнала в файл.

8.3.2.6.1.8. ЖУРНАЛ. ПРАВА ДОСТУПА

Для работы со Справочником в режиме исполнения предусмотрены специальные права доступа.

В дереве системы в группе Безопасность настраиваются умолчания для всех добавляемых в проект Журналов. У каждого отдельного Журнала в его панели свойств можно изменить сделанные умолчания.

Вид панели свойств Журнала в режиме редактирования Прав:



Доступные для настройки права доступа:

Право	Назначение
Управление	Определяет возможность пользователя работать со справочником в режиме исполнения. Если право не предоставлено, то справочник будет полностью отключен (полностью недоступен для управления). Пользователь не сможет воспользоваться панелью инструментов и другими инструментами.
Квитирование сообщения	Определяет право оператора квитировать сообщения. Если установлен флаг Журнал, то в журнале сформируется сообщение Квитировано : <время активации> <текст сообщения> (если в журнале нет столбца с текстом сообщения, только время будет)
Групповое квитирование сообщений	Определяет право оператора квитировать несколько сообщений одновременно. Если установлен флаг Журнал, то в журнале сформируется сообщение Квитировано N сообщений
Использование фильтров в журнале	Определяет право пользователя использовать фильтры в журнале. Если установлен флаг Журнал, то в журнале сформируется сообщения: "Включен фильтр <имя фильтра>", "Отключен фильтр <имя фильтра>", "Включена фильтрация столбца <имя столбца> : <параметры фильтрации>", "Отключена фильтрация столбца <имя столбца>"
Печать сообщений	Определяет возможность пользователю печатать журнал в клиенте визуализации, используя методы и

Право	Назначение
	кнопки панели инструментов. Если принтер указан в приложении MasterSCADA 4D Client Monitor (если клиент запускается на устройстве отличном от исполнительной системы) или в приложении MasterSCADA 4D Monitor в поле Параметры запуска клиента, и установлен флаг Журнал то в журнале сформируется сообщение Печать журнала. Текст "Печать журнала на принтер <имя принтера>"
Экспорт сообщений	Определяет возможность экспорта сообщений журнала. Если место сохранения файлов указано в приложении MasterSCADA 4D Client Monitor (если клиент запускается на устройстве отличном от исполнительной системы) или в приложении MasterSCADA 4D Monitor в поле Параметры запуска клиента, и установлен флаг Журнал, то в журнале сформируется сообщение Экспорт журнала. Текст "Сохранение журнала в файл <имя файла и путь>"

8.3.2.6.2. РАБОТА С ЖУРНАЛОМ В КЛИЕНТЕ ВИЗУАЛИЗАЦИИ

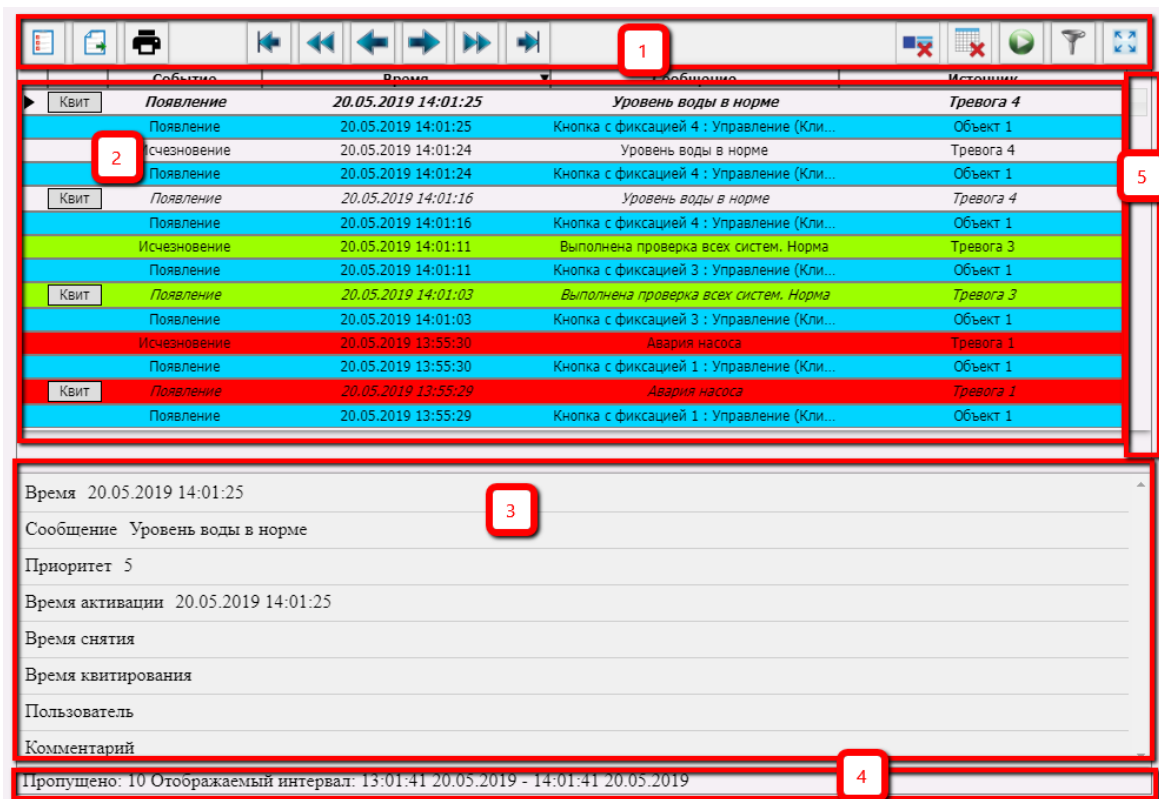
Вид журнала в клиенте визуализации и способы работы с ним зависят от типа транслятора НМІ:

Журнал в клиенте визуализации при НМІ v1

Журнал в клиенте визуализации при НМІ v2

8.3.2.6.2.1. ЖУРНАЛ В КЛИЕНТЕ ВИЗУАЛИЗАЦИИ ПРИ НМІ V1

Вид журнала в клиенте визуализации:



Где:



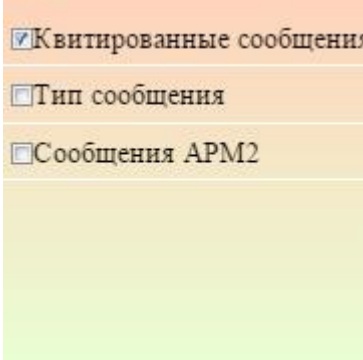

- 1 - Панель инструментов
- 2 - Таблица сообщений
- 3 - Легенда сообщений
- 4 - Строка статуса
- 5 - Линейка прокрутки

Панель инструментов

Панель содержит следующие инструменты:

Элемент	Название	Назначение
	Квитировать все	Квитирует все сообщения на странице.
	Экспортировать	Открывает диалог сохранения текстового файла, в который экспортируются сообщения на текущей странице (имя по умолчанию – <i>journal.txt</i>).

Элемент	Название	Назначение
	Печать	Открывает диалог печати. Будут выведены на печать все загруженные в журнал сообщения.
	Первая страница	Открывает первую страницу журнала. Если автопрокрутка была включена, то при нажатии на эту кнопку она выключится.
	Предыдущая страница	Открывает предыдущую страницу журнала. Если автопрокрутка была включена, то при нажатии на эту кнопку она выключится.
	Предыдущее сообщение	Переводит указатель на предыдущую позицию в таблице. Если автопрокрутка была включена, то при нажатии на эту кнопку она выключится.
	Следующее сообщение	Переводит указатель на следующую позицию в таблице.
	Следующая страница	Открывает следующую страницу журнала.
	Последняя страница	Открывает последнюю страницу журнала. Если автопрокрутка была выключена, то при нажатии на кнопку она включится.
	Скрыть/показать легенду	Включает и выключает отображение легенды.
	Остановить/запустить автопрокрутку	Если автопрокрутка включена (запущена), то показывается последняя страница журнала сообщений. При этом новые сообщения отображаются в верхней части таблицы (автоматически включается сортировка по полю Время активации по убыванию). Если автопрокрутка выключена (остановлена), то пользователь самостоятельно выбирает какие сообщения просматривать. Для того чтобы пользователь не пропустил важные сообщения, в строке статуса отображается, сколько сообщений

Элемент	Название	Назначение
		появилось в журнале за то время, пока автопрокрутка была остановлена.
	Скрыть грид	Скрывает и отображает таблицу сообщений в журнале.
	Скрыть/показать фильтр	<p>Включает и выключает отображение панели фильтров. Данный инструмент отображается только в том случае, если фильтры были созданы в среде разработки. В панели отображаются только фильтры, созданные разработчиком проекта.</p>  <p>Каждый фильтр отображается на панели в отдельной строке. Строка содержит имя фильтра и флаг <input type="checkbox"/> / <input checked="" type="checkbox"/> применения фильтра. Таблица сообщений отображает только те сообщения, которые удовлетворяют одновременно всем примененным фильтрам. Флаг применения фильтра может быть недоступен (см. описание окна настройки журнала). Цветом панели фильтров управляет свойство Заливка фильтров.</p>
	Полноэкранный режим	Включает полноэкранный режим, когда размер журнала станет равным размеру окна, в котором он находится.

Разработчик проекта может выключить отображение панели инструментов журнала, используя свойство Отображать тулбар.

Для работы с журналом могут быть созданы собственные элементы управления, например кнопки, и настроены события, например, в событие Клик мыши добавлено действие Вызвать метод, где будет указано какое действие с журналом должно выполняться по клику мыши на элемент управления:

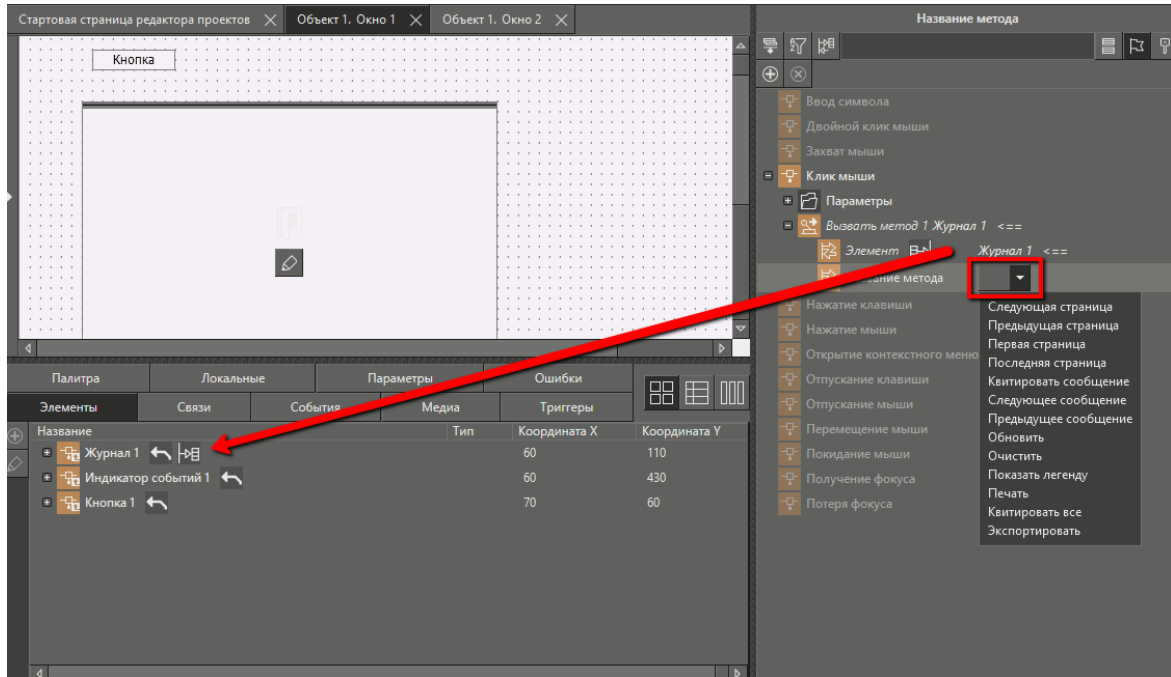


Таблица сообщений






Таблица сообщений отображает сообщения журнала, которые удовлетворяют примененным фильтрам.

На одной странице журнала отображаются все сообщения, удовлетворяющие свойству Интервал выборки данных. При переходе к следующей странице будут отображаться сообщения за следующий интервал. Если сообщений за интервал окажется больше, чем умещается на видимой части страницы журнала, то появится линейка прокрутки.


Вид:

Квитировать	Время	Сообщение	Объект
	30.01.2019 14:19:21	Кнопка с фиксацией 1 : Управл...	Объект 1
Квит	30.01.2019 14:19:21	Останов оборудования	Объект 1
	30.01.2019 14:19:21	Кнопка с фиксацией 1 : Управл...	Объект 1
	30.01.2019 14:19:20	Кнопка с фиксацией 1 : Управл...	Объект 1
Квит	30.01.2019 14:19:20	Останов оборудования	Объект 1
	30.01.2019 14:19:20	Кнопка с фиксацией 1 : Управл...	Объект 1
	30.01.2019 14:19:19	Плановый ремонт	Объект 1
	30.01.2019 14:19:19	Кнопка с фиксацией 2 : Управл...	Объект 1
	30.01.2019 14:19:18	Кнопка с фиксацией 2 : Управл...	Объект 1
Квит	30.01.2019 14:19:17	Плановый ремонт	Объект 1
	30.01.2019 14:19:17	Кнопка с фиксацией 2 : Управл...	Объект 1
Квит	30.01.2019 14:19:01	Минимальное аварийное	
Квит	30.01.2019 14:16:50	Плановый ремонт	Объект 1
	30.01.2019 14:16:50	Кнопка с фиксацией 2 : Управл...	Объект 1
	30.01.2019 14:16:49	Кнопка с фиксацией 2 : Управл...	Объект 1
	30.01.2019 14:16:49	Останов оборудования	Объект 1
	30.01.2019 14:16:49	Кнопка с фиксацией 1 : Управл...	Объект 1
	30.01.2019 14:16:49	Кнопка с фиксацией 1 : Управл...	Объект 1

Инструменты таблицы:

Элемент	Описание
	Заголовок таблицы. Вид заголовка зависит от настроек, заданных на вкладке Стилль заголовка окна настройки журнала.
	Строка таблицы. В каждой строке отображается информация о том или ином сообщении. Количество информации в строке таблицы зависит от настроек, заданных на вкладке Столбцы таблицы окна настройки журнала. Внешний вид строки зависит от настроек заданных на вкладке Стилль строки окна настройки журнала. На рисунке выше видно, что активные сообщения отображаются жирным шрифтом, а не квитируемые сообщения - курсивом.
	Указатель сообщения, параметры которого должны отображаться в легенде. Для перемещения указателя в требуемую строку необходимо выполнить клик мыши по этой строке или использовать панель инструментов журнала.
	Кнопка квитирувания сообщения. Для квитируемых сообщений, а также для сообщений, не требующих квитирувания, не отображается.
	Указатель сортировки. Располагается справа от заголовка столбца и показывает по какому столбцу таблицы осуществляется сортировка сообщений, а также направление сортировки - по убыванию или по возрастанию.

Легенда сообщений

Легенда отображается в нижней части журнала в том случае, если свойство журнала Отображать легенду принимает значение TRUE. Включить легенду можно при помощи панели инструментов журнала или связав свойство журнала с каким-либо элементом управления, например, кнопкой. В легенде отображаются параметры сообщения, на которое установлен указатель  в таблице сообщений. Количество информации в легенде зависит от настроек, выполненных на вкладке Столбцы легенды окна настройки журнала.

Вид:

Время	04.02.2019 15:19:45
Сообщение	Плановый ремонт
Приоритет	1
Активность	true
Время снятия	
Квитированность	true
Время квитирования	Mon Feb 04 2019 15:19:56 GMT+0300 (RTZ 2 (зима))
Комментарий	

Строка статуса

Строка статуса показывает интервал, за который отображаются сообщения на странице в текущий момент. Если автопрокрутка выключена, то в строке статуса отображается количество сообщений, появившихся с момента выключения автопрокрутки.

Вид:

Пропущено: 2 Отображаемый интервал: 16:06:37 29.12.2018 - 16:36:37 29.12.2018

8.3.2.6.2.2. ЖУРНАЛ В КЛИЕНТЕ ВИЗУАЛИЗАЦИИ ПРИ НМІ V2

Вид журнала в клиенте визуализации:

Квит	Событие	Время	Сообщение	Источник
КВИТ	Появление	23.10.2020 10:17:02	Плановый ремонт	Тревога 2
	Появление	23.10.2020 10:17:02	Управление: Кнопка с фиксацией 2	Объект 1
	Появление	23.10.2020 10:16:22	Вход в систему	Reports
23.10.2020 10:18:14	Появление	23.10.2020 10:16:10	Нижнее аварийное	Параметр 2
	Исчезновение	23.10.2020 10:10:19	Останов оборудования	Тревога 1
	Появление	23.10.2020 10:10:19	Управление: Кнопка с фиксацией 1	Объект 1
КВИТ	Появление	23.10.2020 10:10:18	Останов оборудования	Тревога 1
	Появление	23.10.2020 10:10:18	Управление: Кнопка с фиксацией 1	Объект 1
Время		23.10.2020 10:16:10		
Сообщение		Нижнее аварийное		
Приоритет		750		
Время активации		23.10.2020 10:16:10		
Время снятия		0		
Пропущено: 13	Отображаемый интервал: 23.10.2020 10:18:14 - 20.02.2020 15:25:09			

Где:

1 - Панель инструментов

2 - Таблица сообщений






3 - Легенда сообщений

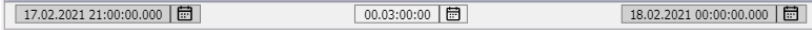


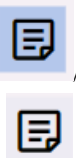
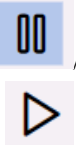
4 - Строка статуса (видимость определяется настройками журнала)





5 - Линейка прокрутки (видимость определяется настройками журнала)

Панель инструментов

Панель содержит следующие инструменты:

Элемент	Название	Назначение
	Квитувать все	Квитурует все сообщения на странице. Если установлен флаг в свойстве Запрашивать комментарий, то после нажатия на кнопку откроется диалоговое окно для ввода комментария, который будет применен ко всем сообщениям и записан в базу данных сообщений. По окончании квитирования будет заполнен столбец Пользователь (в случае если он используется в таблице журнала).
	Экспортировать	Открывает диалог сохранения текстового файла, в который экспортируются сообщения на текущей странице (имя по умолчанию – <i>journal.txt</i>).
	Печать	Открывает диалог печати. Будут выведены на печать все загруженные в журнал сообщения.
	Дерево объектов	Открывает окно, в котором в виде дерева отображаются объекты проекта. Флагами можно отметить те объекты, сообщения которых нужно отображать в журнале.
	Фильтр по времени	<p>Позволяет загрузить из базы данных только те сообщения, которые сформировались в определенный интервал времени. Кнопка появляется в Архивном журнале, если установлен флаг в соответствующем свойстве журнала. Интервал определяется так:</p> <ul style="list-style-type: none"> Начало интервала - это время последнего сообщения, которое видно в журнале, минус интервал Конец интервала - это время последнего сообщения, которое видно в журнале

Элемент	Название	Назначение
		<p>После нажатия на кнопку автопрокрутка отключается, в нижней части журнала появляются поля выбора границ и интервала:</p>  <p>Поведение журнала при изменении значений в полях:</p> <ul style="list-style-type: none"> • При смене правой границы меняется левая, интервал сохраняется • При смене левой границы меняется правая, интервал сохраняется • При смене интервала меняется левая граница, правая сохраняется
	Приоритет	<p>Позволяет выбрать максимальный и минимальный приоритет отображаемых в журнале сообщений. У пользовательских тревог значение приоритета настраивается. Предопределенные сообщения имеют следующие приоритеты:</p> <ul style="list-style-type: none"> • 100 - сообщение о действиях пользователя • 750 - сообщения о нарушении аварийных границ • 500 - сообщения о нарушении предупредительных границ
	Перейти к дате	<p>Позволяет перейти к сообщениям за любую дату. После нажатия на кнопку открывается окно выбора даты. Переход осуществляется после нажатия на кнопку Применить. Автопрокрутка будет выключена.</p>
	Скрыть/показать легенду	Включает и выключает отображение легенды.
	Остановить/запустить автопрокрутку	<p>Если автопрокрутка включена (запущена), то отображается последняя страница журнала сообщений. При этом новые сообщения размещаются в верхней части таблицы (автоматически включается сортировка по полю Время активации по убыванию). Если автопрокрутка выключена (остановлена), то пользователь самостоятельно выбирает</p>

Элемент	Название	Назначение
		<p>какие сообщения просматривать. Для того чтобы пользователь не пропустил важные сообщения, в строке статуса отображается, сколько сообщений появилось в журнале за то время, пока автопрокрутка была остановлена,</p>
	<p>Скрыть/показать фильтр</p>	<p>Включает и выключает отображение панели фильтров. Данный инструмент отображается только в том случае, если фильтры были созданы в среде разработки. В панели отображаются только фильтры, созданные разработчиком проекта.</p> <div data-bbox="783 779 1302 976" style="border: 1px solid #ccc; padding: 5px; background-color: #f0f0f0; margin: 10px auto; width: fit-content;"> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Пользователь <input checked="" type="checkbox"/> Объект <input checked="" type="checkbox"/> Только АРМ диспетчера </div> <p>Каждый фильтр отображается на панели в отдельной строке. Строка содержит имя фильтра и флаг <input type="checkbox"/> / <input checked="" type="checkbox"/> применения фильтра. Таблица сообщений отображает только те сообщения, которые удовлетворяют одновременно всем примененным фильтрам. Флаг применения фильтра может быть недоступен (см. описание окна настройки журнала). Цветом панели фильтров управляет свойство Заливка фильтров.</p>
	<p>Сброс</p>	<p>Отменяет настройки внешнего вида журнала, сделанные пользователем в клиенте визуализации.</p>
	<p>К первой странице</p>	<p>Открывает первую страницу журнала. Если автопрокрутка была включена, то при нажатии на эту кнопку она выключится.</p>
	<p>Последняя страница</p>	<p>Открывает последнюю страницу журнала. Если автопрокрутка была выключена, то при нажатии на эту кнопку она включится.</p>

Разработчик проекта может выключить отображение панели инструментов журнала, используя свойство Отображать тулбар.

Для работы с журналом могут быть созданы собственные элементы управления, например кнопки, и настроены события, например, в событие Клик мыши добавлено действие Вызвать метод, где будет указано какое действие с журналом должно выполняться по клику мыши на элемент управления:

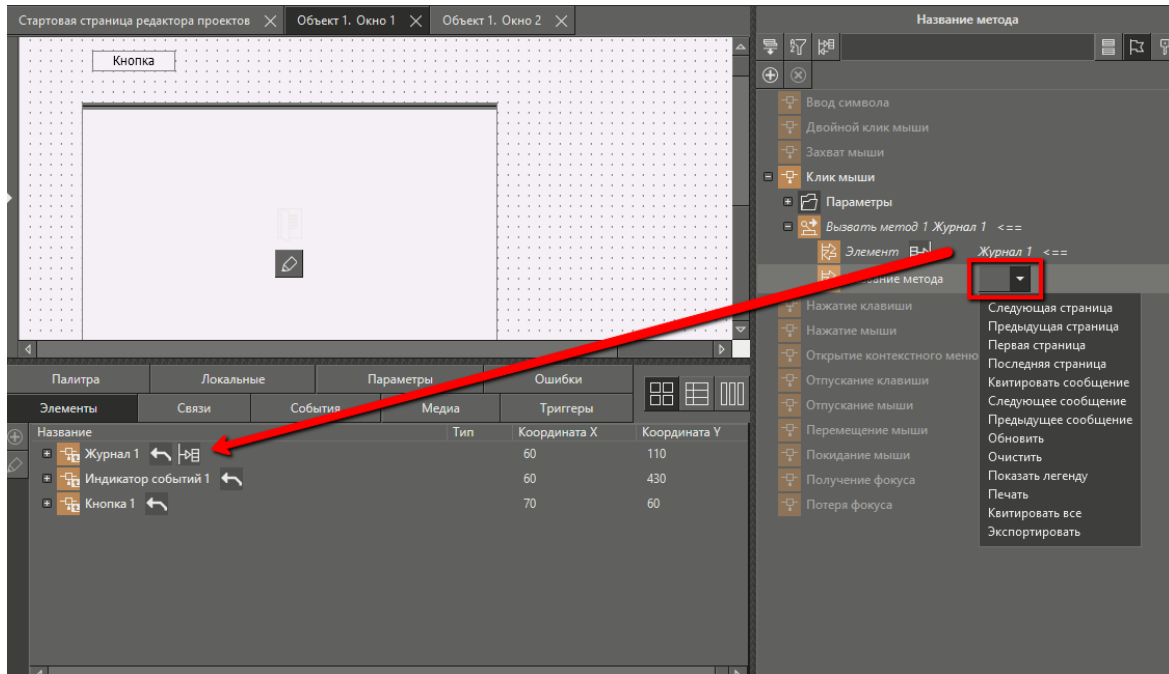


Таблица сообщений

Таблица сообщений отображает сообщения журнала, которые удовлетворяют примененным фильтрам.






На одной странице журнала отображаются все сообщения, удовлетворяющие свойству Интервал выборки данных. При переходе к следующей странице будут отображаться сообщения за следующий интервал. Если сообщений за интервал окажется больше, чем умещается на видимой части страницы журнала, то появится линейка прокрутки.


Вид:

		Событие	Время	Сообщение	Пользователь
		Появление	20.02.2020 15:41:38	Управление: Кнопка ...	Оператор
		Появление	20.02.2020 15:41:37	Управление: Кнопка ...	Оператор
		Появление	20.02.2020 15:41:37	Управление: Кнопка ...	Оператор
		Появление	20.02.2020 15:41:36	Управление: Кнопка ...	Оператор
		Появление	20.02.2020 15:41:36	Управление: Кнопка ...	Оператор
		Квитирование	20.02.2020 15:40:11	Останов оборудования	Оператор
		Появление	20.02.2020 15:40:03	Останов оборудов...	Оператор
		Появление	20.02.2020 15:40:03	Управление: Кнопка ...	Оператор
		Появление	20.02.2020 15:40:01	Управление: Текстов...	Оператор
		Появление	20.02.2020 15:39:56	Управление: Кнопка ...	Оператор
		Появление	20.02.2020 15:39:25	Вход в систему	Оператор
		Квитирование	20.02.2020 15:38:05	Минимальное аварий...	Оператор
		Появление	20.02.2020 15:37:51	Вход в систему	Оператор
		Появление	20.02.2020 15:25:09	Минимальное ава...	Оператор

Пропущено: 0 Отображаемый интервал: 20.02.2020 16:13:54 - 20.02.2020 15:25:09

Инструменты таблицы:

Элемент	Описание
	Заголовок таблицы. Вид заголовка зависит от настроек, заданных на вкладке Стиль заголовка окна настройки журнала. Изменить ширину столбца можно, передвигая левой кнопкой мыши разделительную линию между названиями столбцов. При нажатии на кнопку  рядом с заголовком столбца, откроется окно настройки фильтра, где можно указать, какие элементы необходимо отображать в таблице.
	Строка таблицы. В каждой строке отображается информация о том или ином сообщении. Количество информации в строке таблицы зависит от настроек, заданных на вкладке Столбцы таблицы окна настройки журнала. Внешний вид строки зависит от настроек заданных на вкладке Стиль строки окна настройки журнала. На рисунке выше видно, что активные сообщения отображаются жирным шрифтом, а не квитированные сообщения - курсивом.
	Указатель сообщения, параметры которого должны отображаться в легенде. Для перемещения указателя в требуемую строку необходимо выполнить клик мыши по этой строке или использовать панель инструментов журнала.
	Кнопка квитирования сообщения. Для квитированных сообщений, а также для сообщений, не требующих квитирования, не отображается.

Элемент	Описание
	<p>Если отмечен флаг в свойстве Запрашивать комментарий, то после нажатия на кнопку откроется диалоговое окно для ввода комментария, который будет записан в базу данных сообщений. По окончании квитирования будет заполнен столбец Пользователь (в случае если он используется в таблице журнала).</p> <p>Квитировать сообщения можно также при помощи функционального блока EventsCounter</p>
	<p>Указатель сортировки. Располагается справа от заголовка столбца и показывает по какому столбцу таблицы осуществляется сортировка сообщений, а также направление сортировки - по убыванию или по возрастанию. Доступно только в элементе Журнал, элемент Архивный журнал данный механизм не поддерживает</p>

Легенда сообщений

Легенда отображается в нижней части журнала в том случае, если свойство журнала Отображать легенду принимает значение TRUE. Включить легенду можно при помощи панели инструментов журнала или связав свойство журнала с каким-либо элементом управления, например, кнопкой. В легенде отображаются параметры сообщения, на которое установлен указатель ► в таблице сообщений. Количество информации в легенде зависит от настроек, выполненных на вкладке Столбцы легенды окна настройки журнала.

Строка статуса

Строка статуса показывает интервал, за который отображаются сообщения на странице в текущий момент. Если автопрокрутка выключена, то в строке статуса отображается количество сообщений, появившихся с момента выключения автопрокрутки.

Вид:

Пропущено: 2 Отображаемый интервал: 16:06:37 29.12.2018 - 16:36:37 29.12.2018

8.3.2.7. ИНДИКАТОР СОБЫТИЙ

Элемент Индикатор событий предназначен для индикации количества пропущенных активных сообщений. Данный элемент принимает активный вид, если с момента начала сессии в журнал сообщений записано хотя бы одно сообщение. В этом случае, в верхнем правом углу будет отображаться количество пропущенных сообщений, удовлетворяющих заданным для элемента фильтрам. В противном случае, элемент имеет "пассивный" вид. Счетчик элемента будет уменьшаться, если разработчиком проекта будет добавлен, например, такой фильтр, чтобы индикатор отображал только неквитированные сообщения. В этом случае, если сообщение будет квитировано, то значение счетчика уменьшится.


В палитре редактора HMI элемент Индикатор событий находится в категории Контролы.

Вид элемента в палитре:

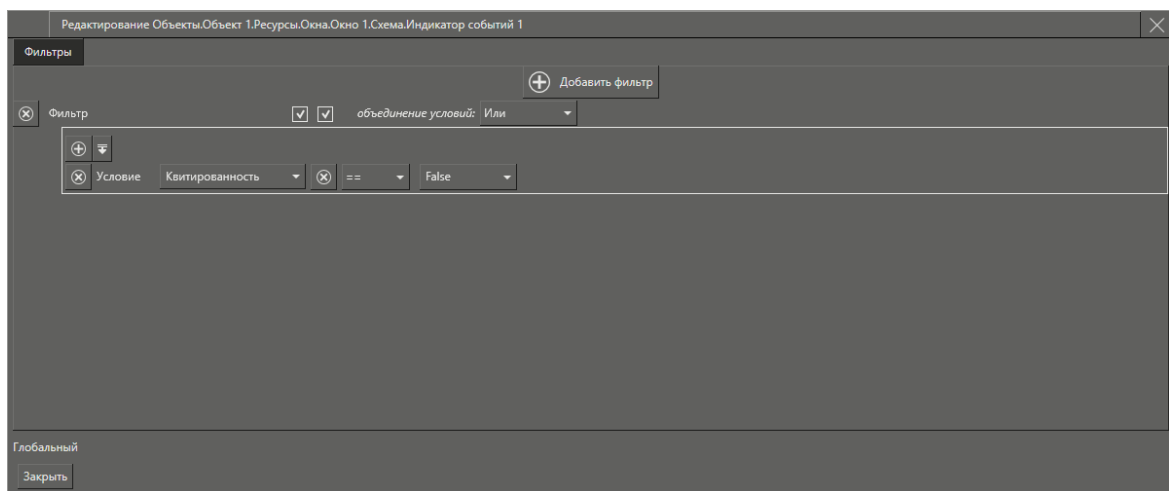


Вид элемента после добавления в рабочую область редактора HMI:



При наведении мыши на элемент становится активной кнопка , при нажатии на которую открывается окно настройки индикатора. Окно настройки также откроется, если выполнить двойной клик по элементу либо нажать клавишу клавиатуры F2

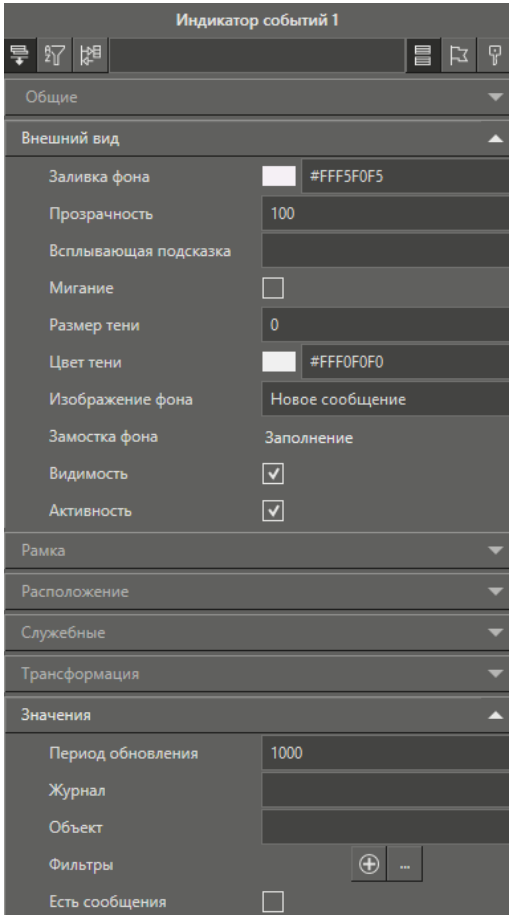
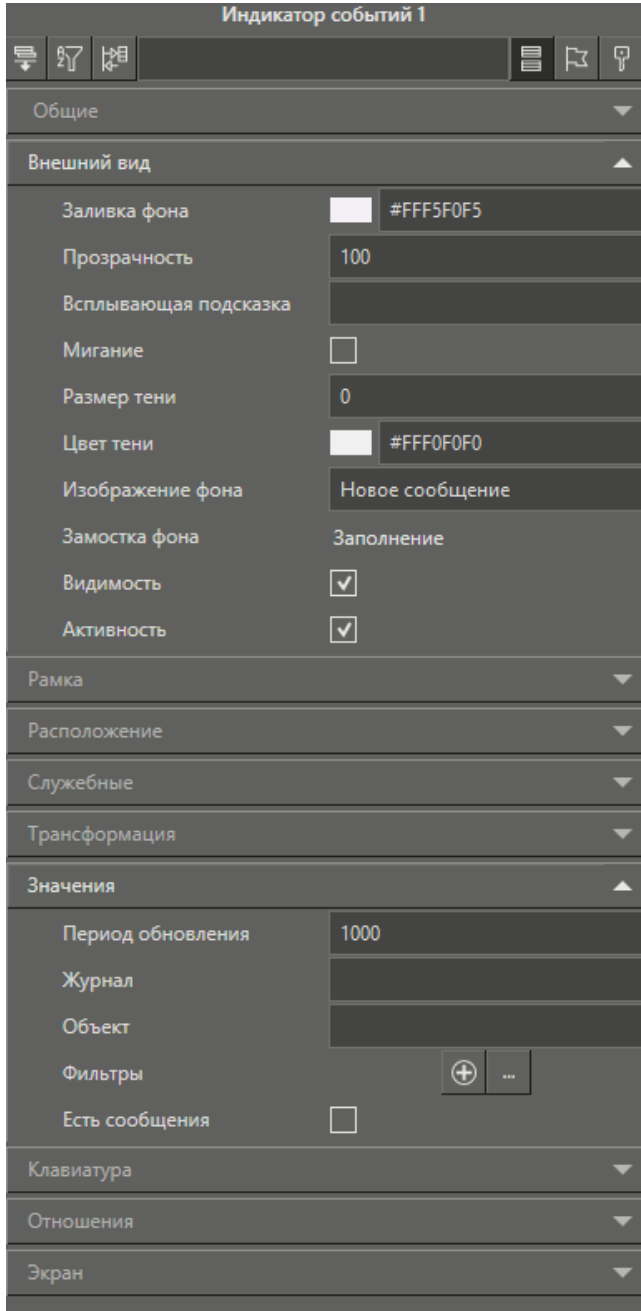
Окно настройки индикатора



Правила работы с этим окном такие же как и с аналогичным окном элемента журнал.


Свойства Индикатора событий



Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
	



Описание основных свойств графического элемента Индикатор событий:

Название	Ссылка на описание
Категория Внешний вид	

Название	Ссылка на описание
Изображение фона	<p>Определяет фоновое изображение элемента в клиенте визуализации. При нажатии на кнопку , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора медиа-ресурса. В качестве фонового изображения элемента может быть использован произвольный графический файл.</p>
Замостка фона	<p>Задаёт тип использования фонового изображения. Используется совместно со свойством Изображение фона. Может принимать значения:</p> <ul style="list-style-type: none"> • Заполнение - размер фонового изображения будет изменен по границам элемента (используется по умолчанию); • Мозаика - размер фонового изображения будет исходным, и повторен столько раз, сколько необходимо чтобы замостить всю площадь элемента; • Центр - размер фонового изображения будет исходным, изображение будет центрировано; • Нет - фоновое изображение использоваться не будет.
Категория Значения	
Период обновления	Период с которым происходит проверка новых сообщений на сервере.
Журнал	<p>Позволяет перейти к журналу после нажатия на индикатор в клиенте визуализации для дальнейшей работы с сообщениями. Если в данное поле перетащить из дерева проекта окно, содержащее в себе журнал, то при нажатии на индикатор данное окно откроется автоматически. Параметры открытия окна в этом случае изменить нельзя. Если требуется указать место открытия окна и другие параметры, то лучше воспользоваться событиями индикатора.</p>
Объект	<p>Определяет связь с объектом, тегом или каналом. Если в данное поле из дерева проекта перетащить объект (тег или канал), то индикатор будет активным в случае, если в этом объекте появятся новые сообщения. Если свойство объект не настроено, то индикатор</p>

Название	Ссылка на описание
	будет контролировать сообщения того объекта, в котором он находится.
Группа фильтры	Группа настроек, в которой отображаются фильтры, ранее созданные в окне настройки индикатора. При помощи кнопок  можно добавить новые фильтры и условия в них, а также изменить настройки ранее созданных. Рекомендуется настраивать фильтры журнала при помощи окна настройки окна настройки журнала. Оно откроется при нажатии на кнопку  .
Есть сообщения	Определяет состояние индикатора в клиенте визуализации в режиме исполнения. Имеет тип BOOL. Данное свойство можно связать с любыми другими свойствами элементов для дополнительной индикации наличия активных событий, удовлетворяющих условиям фильтра индикатора. Если состояние принимает значение TRUE, то индикатор имеет активный вид. Если состояние принимает значение False, то пассивный.

Вид в клиенте визуализации

Пассивный вид	Активный вид
	

8.3.2.8. ТАБЛИЦА ДАННЫХ

Данный элемент предназначен для отображения массивов структур в табличном виде, когда каждый элемент массива структур представлен в виде строки. Количество строк соответствует количеству элементов в массиве.

В палитре редактора HMI элемент Таблица данных находится в категории Контролы .

Важно! Этот элемент рекомендуется использовать в HMI v2


Вид элемента в палитре:




Вид элемента после добавления в рабочую область редактора HMI:



Порядок работы с элементом Таблица данных

1. Элемент добавляется в рабочую область редактора HMI. Задается его размер.
2. Устанавливается связь Источника данных в параметром проекта с использованием панели свойств элемента.
3. Добавляется необходимое количество столбцов (количество столбцов должно соответствовать количеству полей в структуре). При необходимости, задаются настройки столбцов (название, размер и т.п.). Столбцы могут быть добавлены при помощи диалогового окна (рекомендовано), которое открывается при нажатии на кнопку , либо с использованием панели свойств.
4. При необходимости можно настроить Событие Клик по ячейке

Редактор таблицы данных

Чтобы открыть редактор, необходимо нажать на кнопку  после размещения элемента Таблица данных в окне, или выполнить двойной клик по элементу, или нажать клавишу клавиатуры F2:

Редактирование Объекты.Объект 1.Ресурсы.Окно.Окно 2.Схема.Таблица данных 1

Столбцы таблицы | Стиль строки | Стили ячейки | Стиль заголовка

+ Добавить столбец

№ столбца	⊗ 0	< > ⊗ 1	< > ⊗ 2	< > ⊗ 3
Заголовок	Расход	Давление	Температура	Состояние_здвижки
Поле	Расход ⊗	Давление ⊗	Температура ⊗	Состояние_здвижки ⊗


Свойства

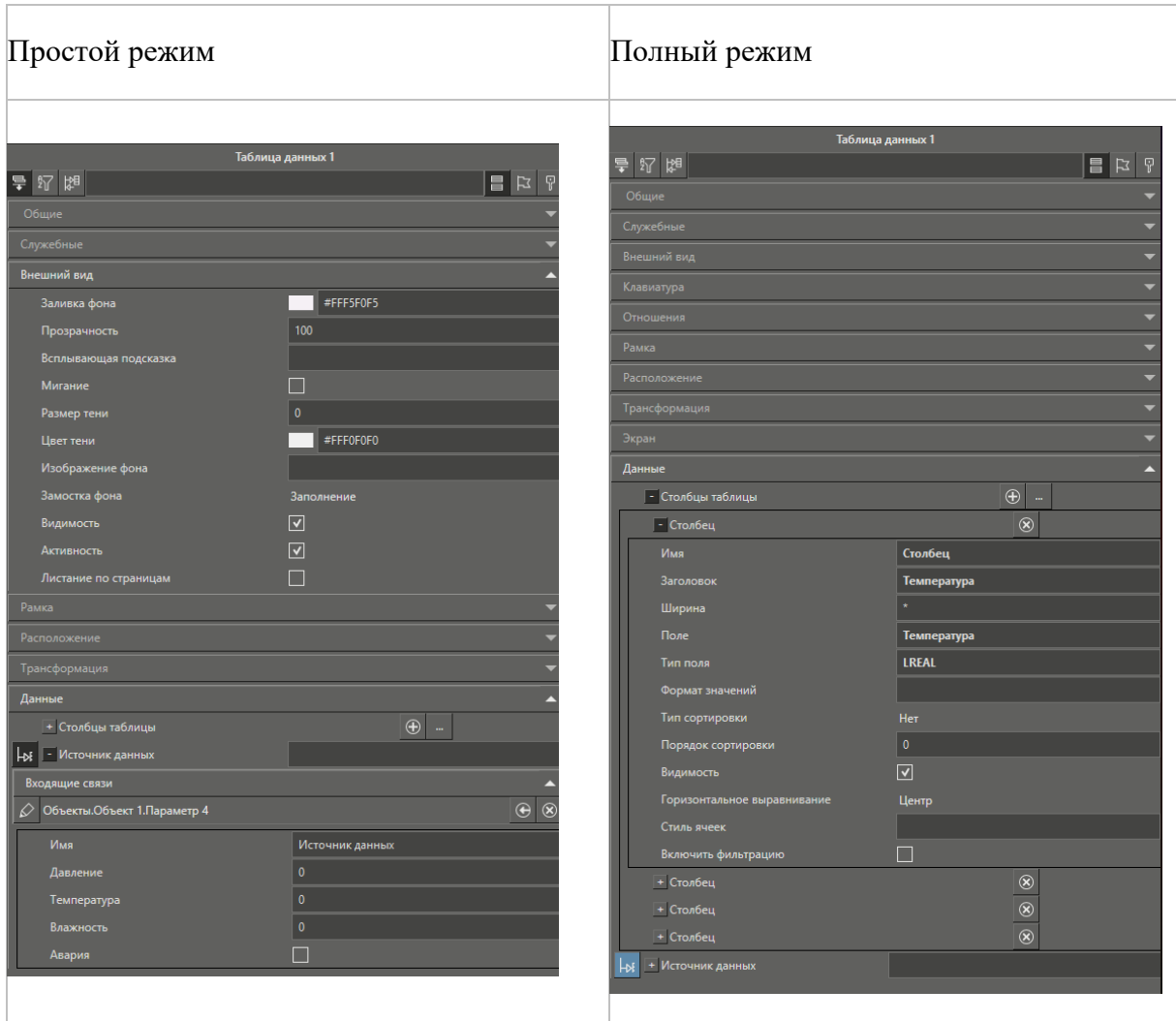
Тип данных	LREAL	LREAL	LREAL	BOOL
Ширина	*	*	*	*
Сортировка	Нет	Нет	Нет	Нет
Порядок сортировки столбца	0	0	0	0
Формат значений				
Видимость	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Гориз. выравнивание	Центр	Центр	Центр	Центр
Стиль ячеек столбца				
Включить фильтрацию столбца	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Запрет редактирования	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Заккрыть

Редактор содержит четыре вкладки:




Название и ссылка	Описание
Столбцы таблицы	Служит для определения содержания и внешнего вида столбцов таблицы данных.
Стиль строки	Определяет внешний вид строки (фона, текста) таблицы в клиенте визуализации. Внешний вид строки может зависеть от значений полей массива структур.
Стили ячейки	Позволяет сформировать зависимость какого-либо свойства ячейки (шрифта, фона и т.п.) от значения поля массива структур.
Стиль заголовка	Определяет внешний вид заголовка таблицы.

Вид панели свойств элемента при нажатой и отжатой кнопке :



Описание основных свойств графического элемента Таблица данных:

Название	Описание
Категория Внешний вид	
Листание по страницам	<p>Используется в случае, если количество строк таблицы больше, чем вмещается на одну страницу. Если свойство имеет значение TRUE, то в клиенте визуализации в нижней части таблицы будет отображаться инструмент для навигации по страницам:</p> <div style="text-align: center; border: 1px solid gray; padding: 5px; margin: 10px 0;"> Предыдущая 1 2 3 Следующая </div> <p>Если свойство имеет значение False, то таблице не будет разделяться на страницы, но появится линейка прокрутки.</p>

Название	Описание
Категория Данные	
Столбцы таблицы	<p>При нажатии на кнопку  в таблицу добавится столбец . В панели свойств добавится группа свойств Элемент массива.</p> <p>При нажатии на кнопку  откроется Редактор таблицы данных.</p>
Столбец	<p>Группа, которая содержит в себе свойства для настройки каждого столбца. Для удаления столбца необходимо нажать на кнопку  . Состав свойств в группе соответствует строкам в редакторе таблицы данных.</p> <p>Столбцы таблицы рекомендуется настраивать с использованием Редактора таблицы данных.</p>
Источник данных	<p>Основное свойство элемента. Его необходимо связать с параметром типа массив структур. После этого появится возможность настраивать столбцы таблицы.</p>

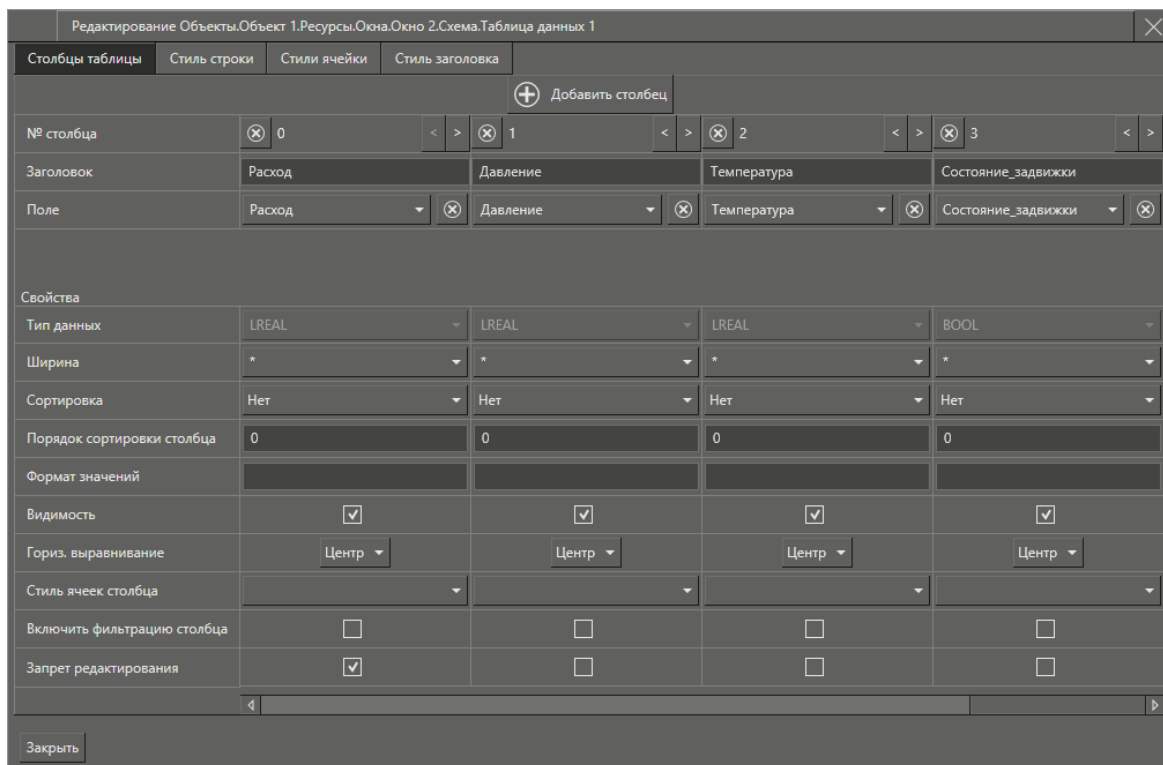
Описание других свойств элемента смотрите в разделе НМІ.Свойства

Важно! Если в клиенте визуализации требуется управлять сортировкой, видимостью столбцов и другими свойствами Таблицы данных, то следует связать нужные свойства с параметрами объекта или узла (если необходимо, чтобы сортировка применялась на всех клиентах) или с параметром окна (если необходимо обеспечить у разных пользователей разное представление данных)

8.3.2.8.1. РЕДАКТОР ТАБЛИЦЫ ДАННЫХ. СТОЛБЦЫ ТАБЛИЦЫ

Вкладка Столбцы таблицы определяет содержание и внешний вид столбцов таблицы данных.




Вкладка имеет вид:



Для добавления столбца необходимо нажать на кнопку **Добавить столбец** в верхней части вкладки. При этом появится столбец, в котором можно установить его свойства.

В случае если все добавленные столбцы не помещаются в видимой части таблицы, в ее нижней части появляется линейка прокрутки.

Описание строк таблицы:

Название	Описание
№ столбца	<p>Определяет порядок столбцов. В клиенте визуализации порядок столбцов будет таким же, как и в данной таблице.</p> <p>Чтобы переместить столбец влево или вправо следует воспользоваться соответствующими стрелками -  </p> <p>Нажатие на кнопку  приведет к удалению столбца</p>
Заголовок	<p>Определяет заголовок столбца. По умолчанию, после настройки строки Поле, Заголовок принимает значение имени связанного поля структуры.</p>
Поле	<p>Определяет имена полей, значения которых будут отображаться в данных столбцах. В выпадающем списке, содержащем имена полей структур, из которых состоит массив, требуется выбрать необходимое. Если эту строку оставить</p>

Название	Описание
	пустой, то порядок столбцов распределится случайным образом (не рекомендуется).
Свойства	
Тип данных	Показывает тип данных поля структуры. Определяется автоматически после настройки строки Поле. Носит информативный характер и не может быть изменен при помощи редактора таблицы данных.
Ширина	<p>Определяет ширину столбца таблицы. Возможны следующие варианты значения свойства:</p> <ul style="list-style-type: none"> • 20 - числовое значение ширины столбца (в px). Может принимать любое значение. Если сумма ширин всех столбцов окажется больше ширины таблицы данных, то в клиенте визуализации в нижней части таблицы появится линейка прокрутки. • Auto - ширина столбца будет определяться автоматически, в зависимости от содержимого заголовка столбца. Если сумма ширин всех столбцов окажется больше ширины таблицы данных, то в клиенте визуализации в нижней части таблицы появится линейка прокрутки. • Символы * - в этом случае ширина столбца будет зависеть от размера экрана монитора и рассчитываться автоматически. Если у одних столбцов будет задан один символ *, а у других - два, то вторые будут в два раза шире первых. Если сумма ширин столбцов, значения которых заданы числовыми настройками или Auto, окажется больше ширины таблицы данных, то в клиенте визуализации в нижней части таблицы появится линейка прокрутки, а на столбцы, у которых ширина задана символами *, будет выделено минимальное количество пикселей, достаточное только для того, чтобы обеспечить возможность растягивания таких столбцов для просмотра их содержимого.

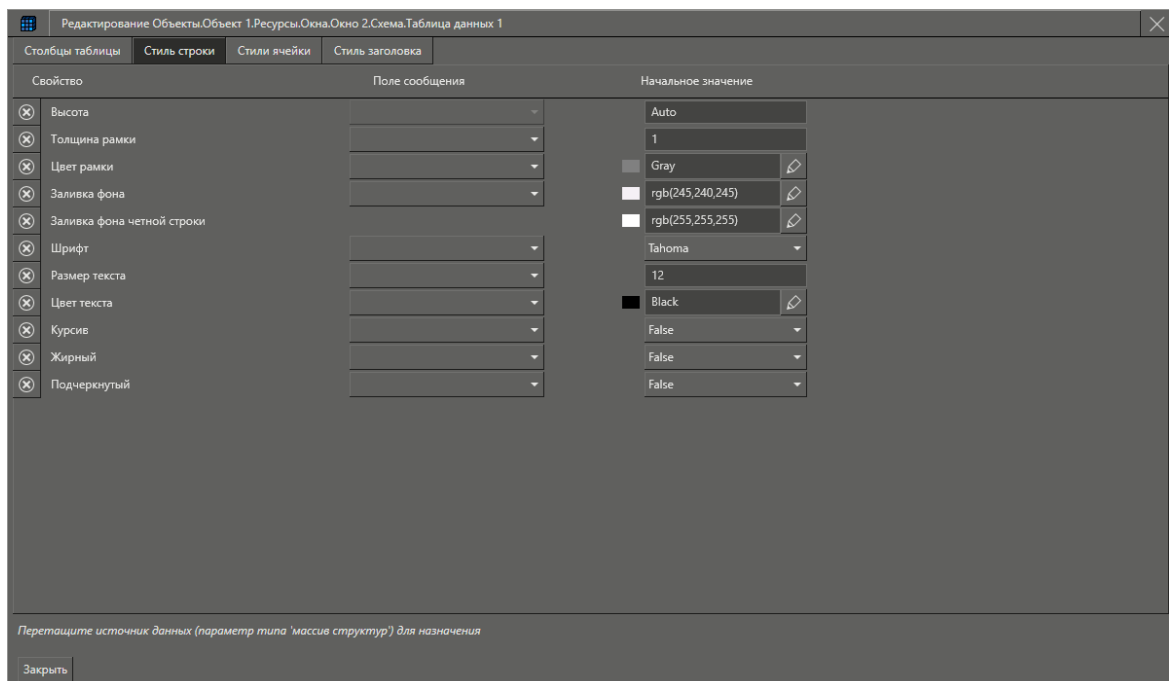
Название	Описание
Сортировка	<p>Определяет тип сортировки в столбце в таблице данных в клиенте визуализации по умолчанию. Может принимать значения:</p> <ul style="list-style-type: none"> • Нет - сортировка в столбце не используется. • По возрастанию - сортировка в столбце используется, первыми в таблице располагаются строки, у которых поле имеет меньшее значение. • По убыванию - сортировка в столбце используется, первыми в таблице располагаются строки, у которых поле имеет большее значение.
Порядок сортировки столбцов	<p>Определяет приоритет сортировки столбцов в случае, когда необходима сортировка по нескольким столбцам (множественная сортировка). Принимает числовые значения (0,1,2, и т.д.) Например, если у двух столбцов используется сортировка, то сначала происходит сортировка по тому столбцу, в котором значение свойства Порядок сортировки столбцов ниже.</p> <p>В клиенте визуализации можно менять множественную сортировку, кликая по столбцам левой кнопкой мыши с нажатой на клавиатуре клавишей Ctrl.</p> <p>Важно! Множественная сортировка поддерживается только в HMI v2</p>
Формат значений	<p>Определяет формат отображения выводимого значения. Описание этого свойства смотрите в разделе Свойство Формат значений.</p>
Видимость	<p>Флаг, определяющий видимость столбца. Если флаг снят, то для соответствующего столбца место в таблице не выделяется. Данное свойство имеет смысл динамизировать, используя панель свойств элемента.</p>
Гориз. выравнивание	<p>Определяет выравнивание заголовка и самого столбца.</p>

Название	Описание
Стиль ячеек столбца	Позволяет выбрать стиль ячеек столбца, отличный от стиля ячеек всей строки. В выпадающем списке появятся возможные стили для ячеек, которые ранее были добавлены в проект.
Включить фильтрацию столбца	Включает фильтрацию столбца в клиенте визуализации.
Запрет редактирования	Определяет возможность редактирования ячеек столбца в клиенте визуализации.

8.3.2.8.2. РЕДАКТОР ТАБЛИЦЫ ДАННЫХ. СТИЛЬ СТРОКИ

Вкладка Стиль строки определяет внешний вид строки таблицы данных. При этом, любое свойство строки может изменяться в зависимости от того, какое значение поля структуры связанного параметра в ней располагается.

Вкладка имеет вид:



На вкладке находится таблица, состоящая из трех столбцов:

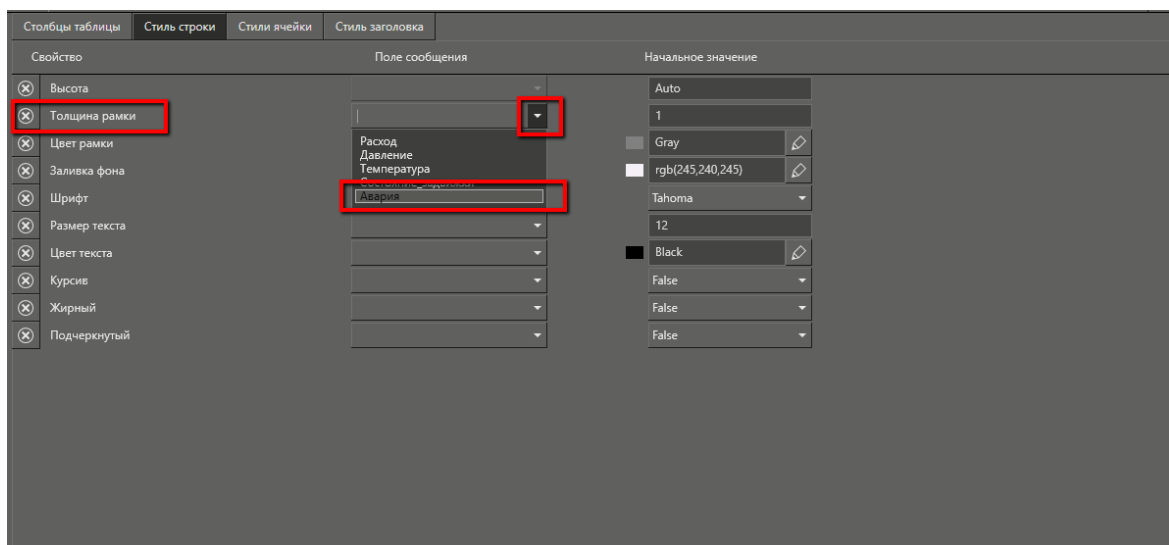
Название	Назначение
Свойство	Название свойства строки.

Название	Назначение
Поле сообщения	Служит для настройки зависимости свойства строки от значения поля структуры связанного параметра. В этом случае свойство будет меняться в режиме исполнения: для одних строк оно будет принимать одно значение, а для других другое.
Начальное значение	Служит для настройки свойства строки в случае, когда его не требуется менять в клиенте визуализации в режиме исполнения.


Пример настройки стиля строки

Допустим, что в таблице необходимо выделять строки, в которых поле Авария принимает значение TRUE .

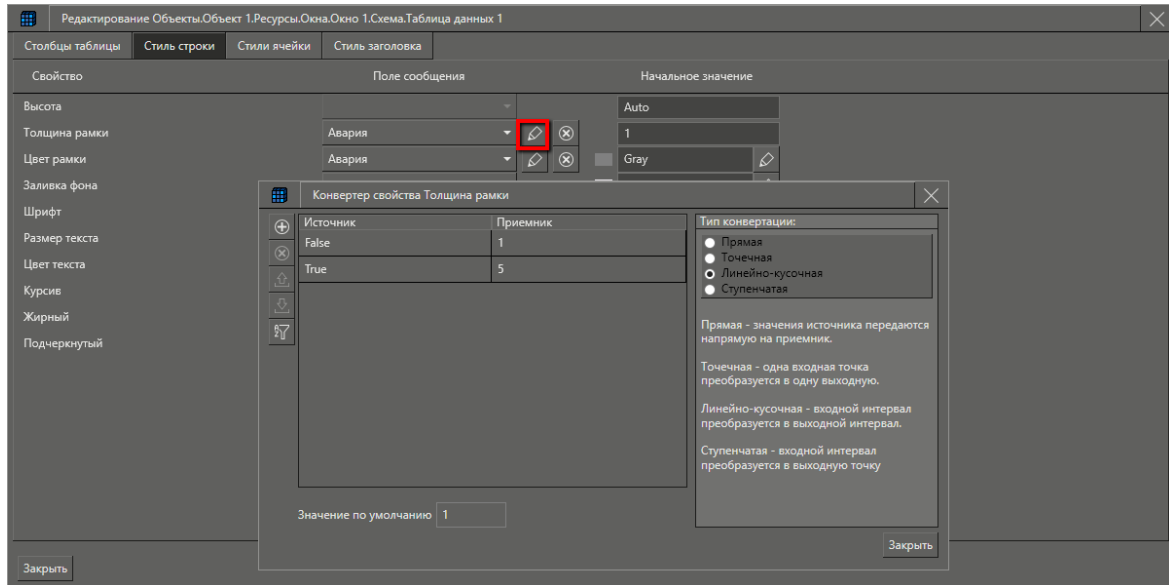
В этом случае требуется настроить свойства Толщина рамки и Цвет рамки. В выпадающем списке Поле сообщения следует выбрать поле Авария:



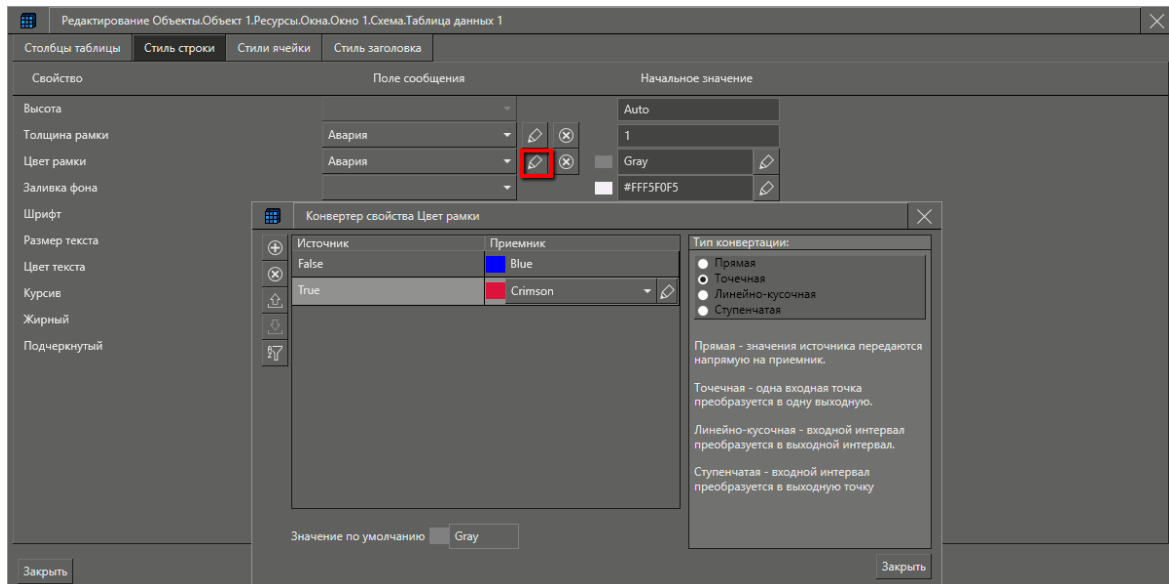
Важно! Варианты в выпадающем списке в столбце Поле сообщения зависят от созданного библиотечного элемента Массив структур

Для настройки зависимости толщины рамки и цвета рамки строки от поля Авария необходимо нажать на кнопку .

Откроется диалоговое окно Конвертер значений, где в столбце Источник задаются значения, которые может принимать поле, а в столбце Приемник задаются значения, которые должно принимать свойство строки. Например, для свойства Толщина рамки:



И для свойства Цвет рамки:



Заливка фона четной строки

Данный параметр нельзя динамизировать.

Если у параметра стиля строки Заливка фона установлена динамизация, то параметр Заливка фона четной строки игнорируется. Если не задана, то для нечетных строк применяется Заливка фона, для четных строк - Заливка фона четной строки

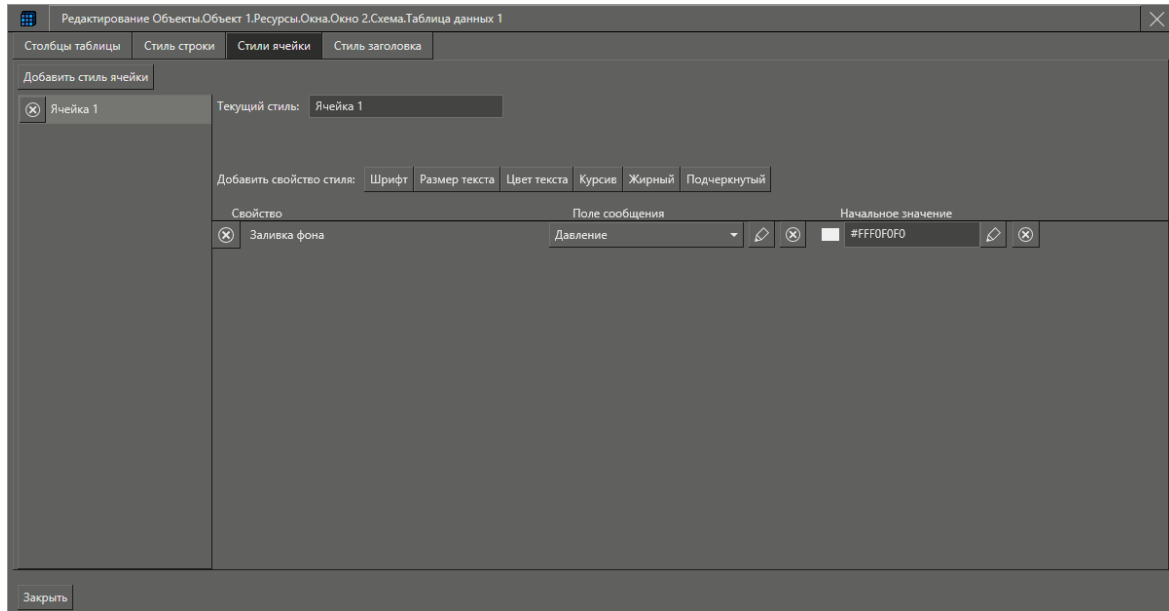
Если у параметра стиля ячейки Заливка фона установлена динамизация, то для столбцов, у которых установлен данный стиль ячейки всегда используется данная динамизация и для этого столбца настройка Заливка фона четной строки игнорируется.


8.3.2.8.3. РЕДАКТОР ТАБЛИЦЫ ДАННЫХ. СТИЛИ ЯЧЕЙКИ

Вкладка Стили ячейки определяет внешний вид ячейки. Используется в случае, когда в таблице необходимо выделить ячейку независимо от всей строки.

Важно! Данный функционал поддерживается только в HMI v2.


Вкладка имеет вид:



Для добавления стиля необходимо нажать на кнопку **Добавить стиль ячейки** в верхней части вкладки. При этом слева будет формироваться список стилей. Нажатие на кнопку  , находящуюся слева от стиля, приведет к его удалению. Выбранный стиль может быть отредактирован с помощью элементов управления, расположенных с правой стороны вкладки.

Описание элементов управления:

Название	Описание
Текущий стиль	В поле отображается текущее имя выбранного стиля, которое , при необходимости, может быть изменено.
Добавить свойство стиля	Набор кнопок, который позволяет определить свойство ячейки, которое будет изменяться в клиенте визуализации в режиме исполнения независимо от стиля строки.
Таблица настроек	После нажатия на кнопку в области Добавить свойство стиля , в таблице появится соответствующая строка, отвечающая за то или

Название	Описание
	иное свойство. Нажатие на кнопку  , находящуюся слева от свойства, приведет к его удалению.

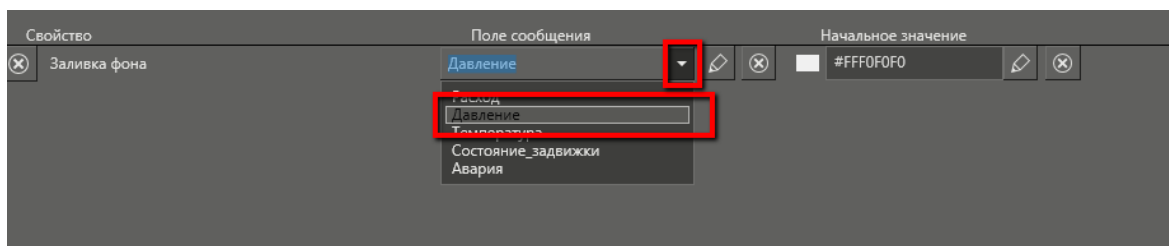
Описание таблицы настроек:

Название	Назначение
Свойство	Название свойства ячейки.
Поле сообщения	Служит для настройки зависимости свойства ячейки от значения поля структуры связанного параметра. Если настройки заданы, то в режиме исполнения свойство будет меняться: для одних строк оно будет принимать одно значение, а для других другое.
Начальное значение	Служит для настройки свойства строки в случае, когда его не требуется менять в клиенте визуализации в режиме исполнения.


Пример настройки стиля строки

Допустим, что в таблице необходимо выделять ячейки, в которых поле Давление находится в определенном диапазоне значений.

Нажмем кнопку Заливка фона в области Добавить свойство стиля. В выпадающем списке Поле сообщения таблицы следует выбрать поле Давление:

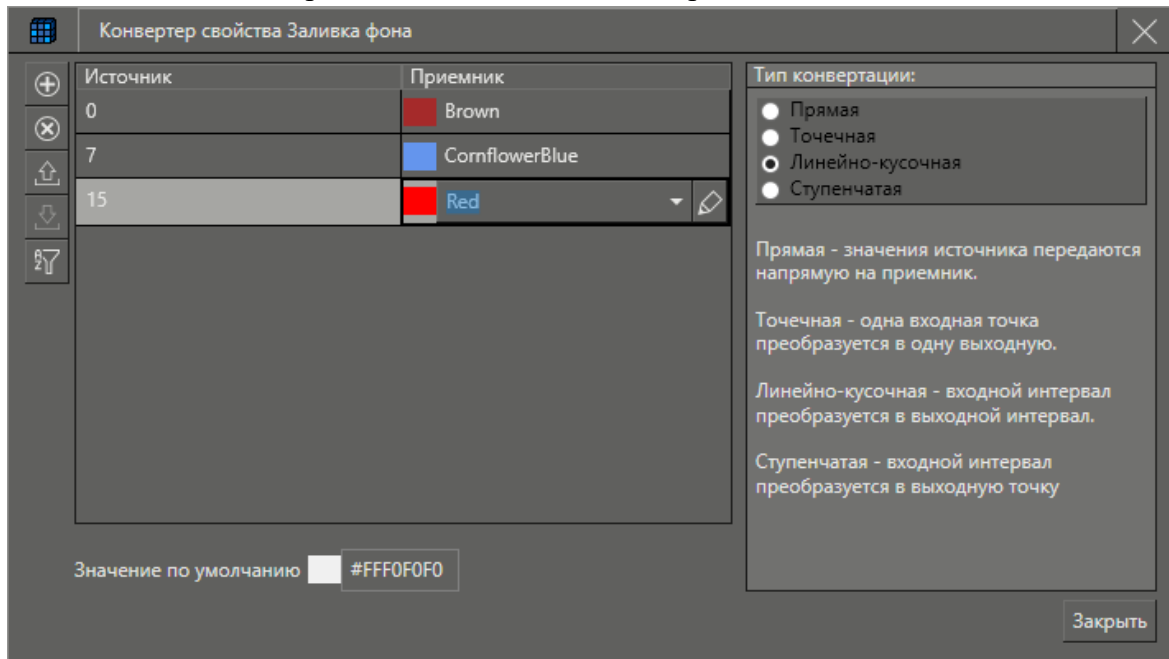


Важно! Варианты в выпадающем списке в столбце Поле сообщения зависят от созданного библиотечного элемента Массив структур.

Для настройки зависимости заливки фона ячейки от поля Давление необходимо нажать на кнопку .

При этом откроется диалоговое окно Конвертер значений, где в столбце Источник могут быть заданы значения, которые может принимать поле, а в столбце Приемник задаются

значения, которые должно принимать свойство ячейки:

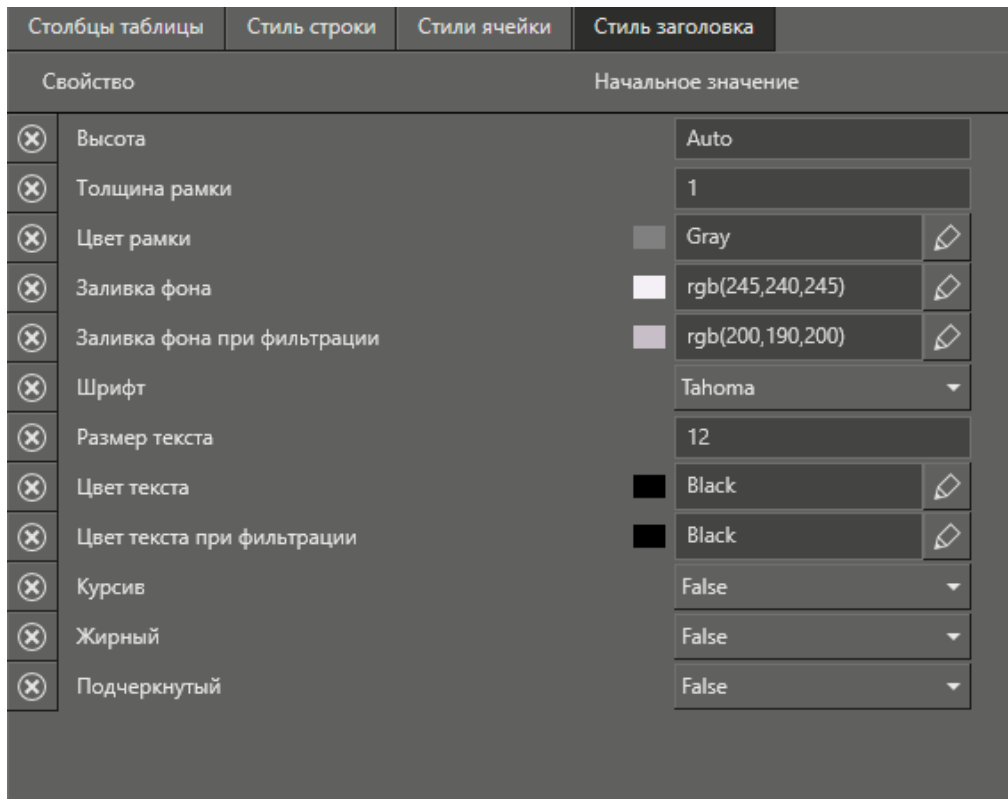


В приведенном примере выбран тип конвертации Линейно-кусочная, а это означает, что в диапазоне значений от 7 до 15 заливка фона ячейки будет голубой, но если значение опустится ниже 7, то заливка фона ячейки станет коричневой, а если повысится выше 15, то красной.

8.3.2.8.4. РЕДАКТОР ТАБЛИЦЫ ДАННЫХ. СТИЛЬ ЗАГОЛОВКА

Вкладка Стиль заголовка определяет внешний вид заголовка таблицы данных.

Вкладка имеет вид:



Важно! Стиль заголовка определяется только в диалоговом окне. В панели свойств стиль заголовка не настраивается.

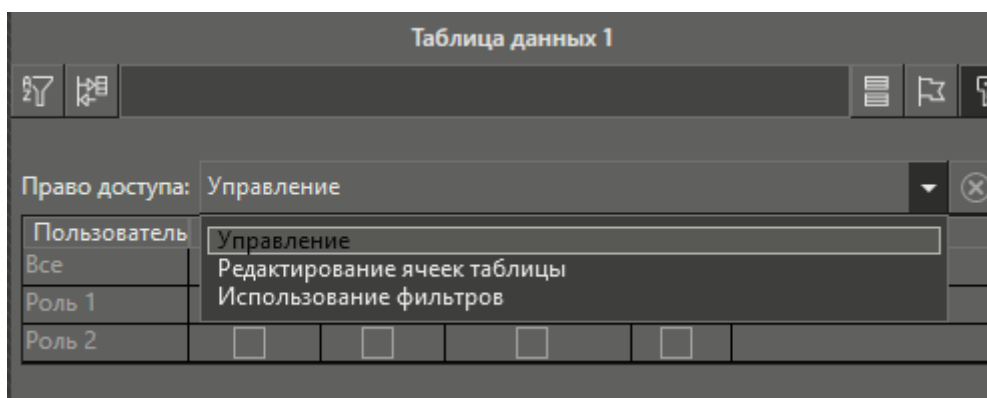
Свойства Заливка фона при фильтрации и Цвет текста при фильтрации относится только к тем столбцам, к которым применена фильтрация

8.3.2.8.5. ТАБЛИЦА ДАННЫХ. ПРАВА ДОСТУПА

Для работы со Таблицей данных в режиме исполнения предусмотрены специальные права доступа.

В дереве системы в группе Безопасность настраиваются умолчания для всех добавляемых в проект Таблиц данных. У каждого отдельного элемента Таблица данных в его панели свойств можно изменить сделанные умолчания.

Вид панели свойств Таблицы данных в режиме редактирования Прав:



Доступные для настройки права доступа:

Право	Назначение
Управление	<p>Определяет возможность пользователя работать со справочником в режиме исполнения. Если право не предоставлено, то справочник будет полностью отключен (полностью недоступен для управления). Пользователь не сможет воспользоваться панелью инструментов и другими инструментами.</p>
Редактирование ячеек таблицы	<p>Определяет право пользователя редактировать ячейки таблицы. Если установлен флаг Журнал, то в журнале сформируется сообщение Изменение записи : <поле1>=<значение1>,<поле2>=<значение2>..." (перечисляются новые измененные значения полей записи)</p>
Использование фильтров	<p>Определяет право пользователя использовать фильтры в таблице данных. Если установлен флаг Журнал, то в журнале сформируются сообщения Включена фильтрация столбца <имя столбца> : <параметры фильтрации>", "Отключена фильтрация столбца <имя столбца></p>

8.3.2.8.6. ТАБЛИЦА ДАННЫХ В КЛИЕНТЕ ВИЗУАЛИЗАЦИИ

Пример отображения Таблицы данных в клиенте визуализации:

▼ Расход ▲	Давление ▲	▼ Температура ▲	Состояние_зад... ▲	Авария ▲
14	14	14	false	false
14	14	14	false	false
14	14	14	false	false
14	14	14	false	false
14	14	14	false	false
14	14	14	false	false
14	14	14	false	false
14	14	14	false	false
14	14	14	false	false
14	14	14	false	false
14	14	14	false	false

Выделенная строка в таблице подсвечивается. Таблица служит для отображения данных в реальном времени.

В случае, если необходимо изменить значение ячеек, то необходимо выполнить двойной клик по строке. Откроется диалоговое окно ввода новых значений. Внешний вид окна формируется автоматически. Для каждого поля, у которого в настройках не установлен флаг Запрет редактирования создается своя строка. В зависимости от типа данных поля будет зависеть элемент для ввода значений. Так например, для ввода даты и времени, будет добавлен в окно элемент ДатаВремя.

Пример окна:

Редактирование ×


Расход



Давление

Температура

Состояние_зadвижки

Введенные значения будут отправлены после нажатия на кнопку Применить.

Если в настройках установлен флаг Включить фильтрацию столбца, то в заголовке, рядом с именем столбца появится кнопка . При нажатии на эту кнопку рядом с заголовком столбца, откроется окно настройки фильтра, где можно указать, какие элементы необходимо отображать в таблице.

Рядом с именем заголовка столбца находится указать сортировки  . Показывает по какому столбцу таблицы осуществляется сортировка сообщений, а также направление сортировки - по убыванию или по возрастанию.

8.3.2.9. СПРАВОЧНИК

Данный элемент позволяет в клиенте визуализации отображать, редактировать, добавлять и удалять данные таблицы сторонней базы данных. В отличие от контрола Таблица данных, источником данных которой является внутренний параметр типа массив структур, Справочники обращаются к базе напрямую. Также справочники позволяют работать со столбцами таблиц базы данных, связанных между собой первичным и внешним ключом

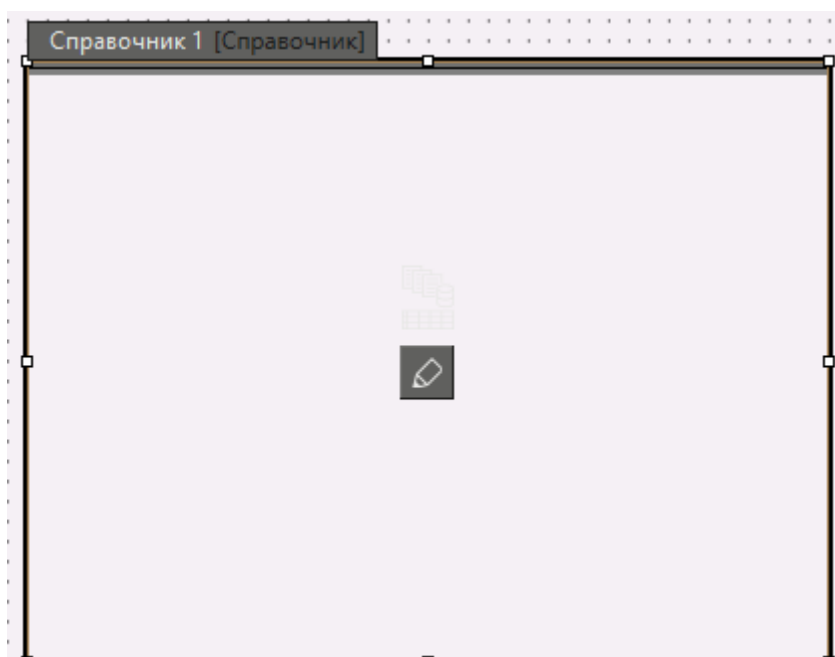
В палитре редактора НМІ элемент Справочник находится в категории Контролы .

Важно! Этот элемент рекомендуется использовать в НМІ v2


Вид элемента в палитре:




Вид элемента после добавления в рабочую область редактора НМІ:

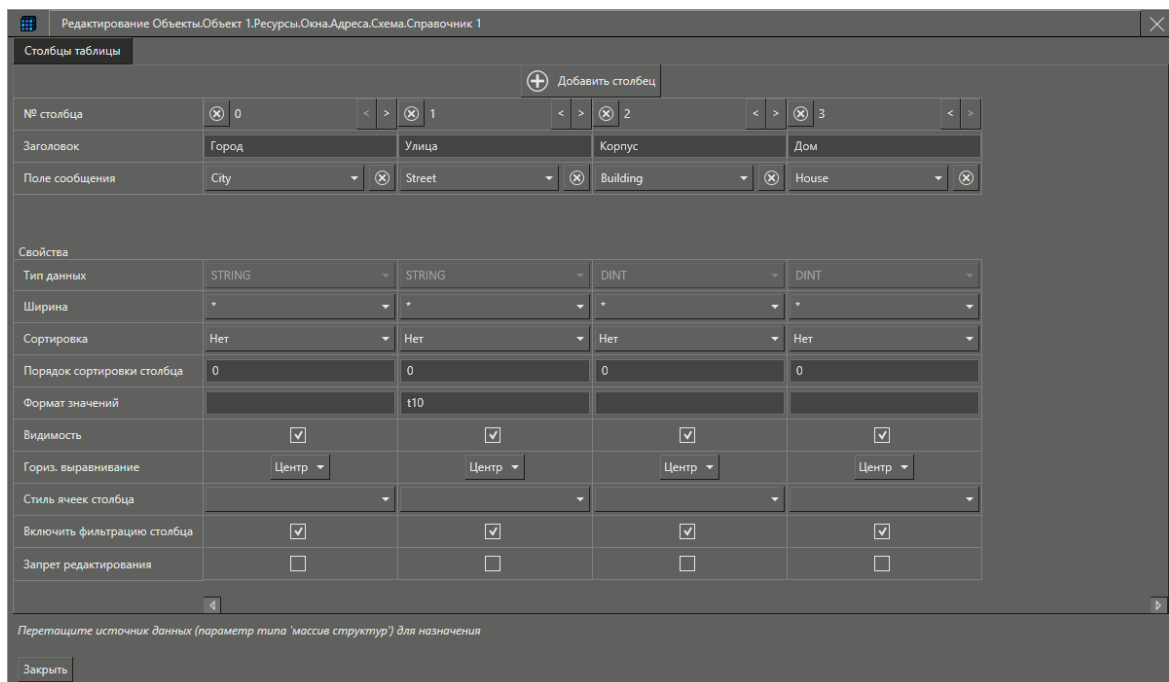


Порядок работы с элементом Справочник

1. Создается база данных, с которой будет работать справочник.
2. В пользовательской библиотеке создается элемент Справочник, в котором определяются поля, которые необходимо отобразить в клиенте визуализации, в графическом элементе Справочник.
3. В дерево системы добавляется элемент БД справочников, в котором настраивается связь с базой данных, с которой нужно взаимодействовать.
4. Элемент добавляется в рабочую область редактора HMI. Задается его размер.
5. Устанавливается связь между библиотечным элементом Справочник и данным графическим элементом с использованием панели свойств элемента. Необходимо перетащить элемент из библиотеки в свойство графического элемента Источник данных
6. Добавляется необходимое количество столбцов (количество столбцов должно соответствовать количеству полей в Справочнике). При необходимости, задаются настройки столбцов (название, размер и т.п.). Столбцы могут быть добавлены при помощи диалогового окна (рекомендовано), которое открывается при нажатии на кнопку , либо с использованием панели свойств.
7. При необходимости можно настроить Событие Клик по ячейке

Редактор Справочника

Чтобы открыть редактор, необходимо нажать на кнопку  после размещения графического элемента Справочник в окне:



Редактор содержит одну вкладку Столбцы таблицы.

Вид панели свойств элемента:

Справочник 1

Общие

Имя	Справочник 1
Полное имя	Объекты.Объект 1.Ресурсы.Окна.Адреса.Схема.
Метки	
Комментарий	Тип окна

Служебные

Данные

+ Столбцы таблицы

Источник данных: 0

Фильтр

Внешний вид

Заливка фона	rgb(245,240,245)
Прозрачность	100
Всплывающая подсказка	
Мигание	<input type="checkbox"/>
Размер тени	0
Цвет тени	rgb(240,240,240)
Изображение фона	
Замостка фона	Заполнение
Видимость	<input checked="" type="checkbox"/>
Активность	<input checked="" type="checkbox"/>
Сохранять состояние	
Листание по страницам	<input type="checkbox"/>
Запрет добавления	<input type="checkbox"/>
Запрет удаления	<input type="checkbox"/>
Запрет редактирования	<input type="checkbox"/>
Тип линий сетки	Вертикальные

Расположение

Рамка





Отношения

Трансформация

Клавиатура

Экран

Описание основных свойств графического элемента Таблица данных:

Название	Описание
Категория Внешний вид	
Листание по страницам	<p>Используется в случае, если количество строк таблицы больше, чем вмещается на одну страницу. Если свойство имеет значение TRUE, то в клиенте визуализации в нижней части таблицы будет отображаться инструмент для навигации по страницам:</p>  <p>Если свойство имеет значение False, то таблице не будет разделяться на страницы, но появится линейка прокрутки.</p>
Запрет добавления	Определяет возможность добавления новой строки в таблицу.
Запрет удаления	Определяет возможность удаления строк из таблицы.
Запрет редактирования	Определяет возможность редактировать строки таблицы.
Категория Данные	
Столбцы таблицы	<p>При нажатии на кнопку  в таблицу добавится столбец . В панели свойств добавится группа свойств Элемент массива. При нажатии на кнопку  откроется Редактор справочника.</p>
Столбец	<p>Группа, которая содержит в себе свойства для настройки каждого столбца. Для удаления столбца необходимо нажать на кнопку  . Состав свойств в группе соответствует строкам в Редакторе справочника.</p> <p>Столбцы таблицы рекомендуется настраивать с использованием Редактора справочника.</p>

Название	Описание
Источник данных	<p>Основное свойство элемента. Его необходимо связать с библиотечным элементом Справочник. После этого появится возможность настраивать столбцы таблицы.</p>
Фильтры	<p>Задается в виде текста условие для выборки данных для справочника. Условия фильтра записываются через запятую.</p> <p>При составлении текста фильтра можно использовать -</p> <ul style="list-style-type: none"> • Имена полей справочника, задается имя поля основной таблицы в фигурных скобках, например, {A} = 5 <ul style="list-style-type: none"> • Имена зависимых полей в формате {имя_поля_основной_таблицы.имя_поля_зависимой_таблицы}, например (A.B) = 5 • Операции = > < != >= <= or and not • Константы - числа, true/false, строковые величины. Если строковая константа содержит в себе символ ', то его надо заменять на \ <p>При открытии контрол передает данную строку фильтра в запросе к базе данных.</p> <p>Пример записи фильтра: {idAddress.City} != 'Москва' and {Meter2} >= 33</p> <p>Важно! Если после открытия справочника меняется параметр Фильтр, то необходимо обновить таблицу. Если во время обновления был открыт диалог добавления/удаления записи, и в новой выборке нет записи с данным ключом, диалог надо закрыть без сохранения.</p>

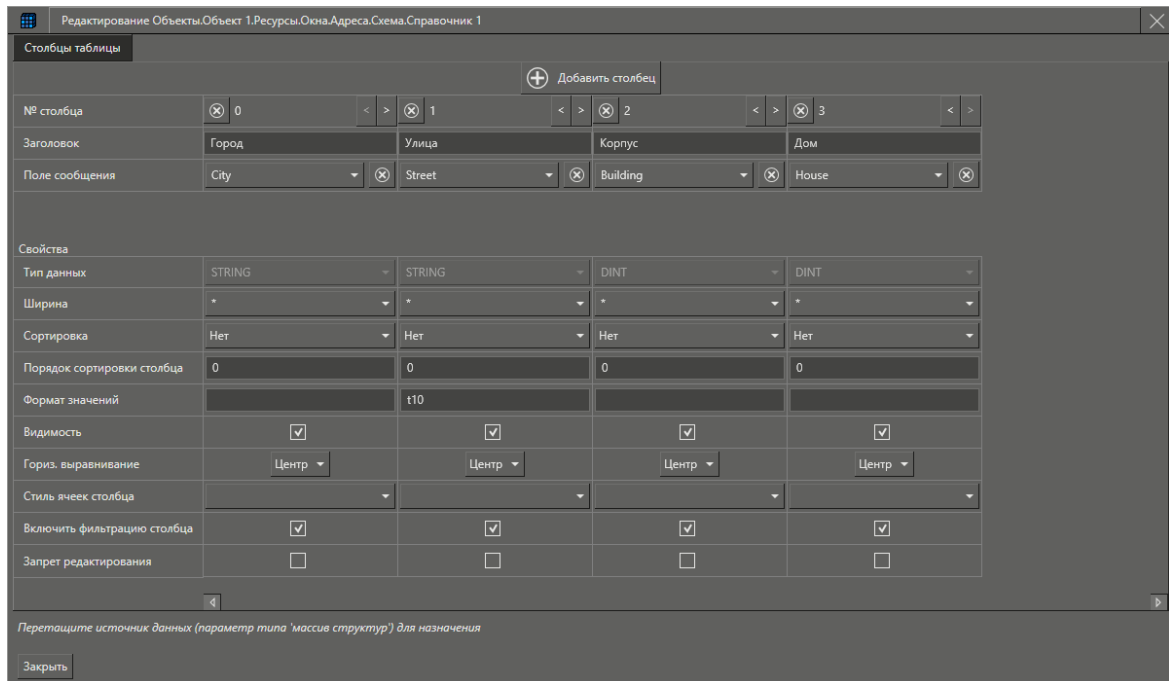
Описание других свойств элемента смотрите в разделе НМІ.Свойства

Важно! Если в клиенте визуализации требуется управлять сортировкой, видимостью столбцов и другими свойствами Справочника, то следует связать нужные свойства с параметрами объекта или узла (если необходимо, чтобы сортировка применялась на всех клиентах) или с параметром окна (если необходимо обеспечить у разных пользователей разное представление данных)

8.3.2.9.1. РЕДАКТОР СПРАВОЧНИКА. СТОЛБЦЫ ТАБЛИЦЫ

Вкладка Столбцы таблицы определяет содержание и внешний вид столбцов Справочника.




Вкладка имеет вид:



Для добавления столбца необходимо нажать на кнопку **Добавить столбец** в верхней части вкладки. При этом появится столбец, в котором можно установить его свойства.

В случае если все добавленные столбцы не помещаются в видимой части таблицы, в ее нижней части появляется линейка прокрутки.

Описание строк таблицы:

Название	Описание
№ столбца	<p>Определяет порядок столбцов. В клиенте визуализации порядок столбцов будет таким же, как и в данной таблице.</p> <p>Чтобы переместить столбец влево или вправо следует воспользоваться соответствующими стрелками -  </p> <p>Нажатие на кнопку  приведет к удалению столбца</p>
Заголовок	<p>Определяет заголовок столбца. По умолчанию, после настройки строки Поле, Заголовок принимает значение имени связанного поля структуры.</p>
Поле	<p>Определяет имена полей, значения которых будут отображаться в данных столбцах. В выпадающем списке, содержащем имена полей библиотечного элемента Справочник, требуется выбрать необходимое. Если эту строку</p>

Название	Описание
	<p>оставить пустой, то порядок столбцов распределится случайным образом (не рекомендуется). Если выбранное поле является ссылкой на другой справочник, то по умолчанию отображается то поле справочника, в котром отмечено свойство Отображаемое имя, если такого нет, то будет отображаться Ключевое поле, на который идет ссылка. Кроме того можно добавить дополнительные поля для отображения данных справочников, на которые идут ссылки из данного (путь к ним задается в форме имя_поля.имя_поля_внешнего_справочника, например:</p> <div data-bbox="643 741 1038 801" style="border: 1px solid black; padding: 2px; display: inline-block;"> idAddress.City ▼ ✕ </div> <p>Может быть иерархическая ссылка.</p>
Свойства	
Тип данных	<p>Показывает тип данных поля. Определяется автоматически после настройки строки Поле. Носит информативный характер и не может быть изменен при помощи редактора справочника.</p>
Ширина	<p>Определяет ширину столбца таблицы. Возможны следующие варианты значения свойства:</p> <ul style="list-style-type: none"> • 20 - числовое значение ширины столбца (в px). Может принимать любое значение. Если сумма ширин всех столбцов окажется больше ширины справочника, то в клиенте визуализации в нижней части таблицы появится линейка прокрутки. • Auto - ширина столбца будет определяться автоматически, в зависимости от содержимого заголовка столбца. Если сумма ширин всех столбцов окажется больше ширины справочника, то в клиенте визуализации в нижней части таблицы появится линейка прокрутки. • Символы * - в этом случае ширина столбца будет зависеть от размера экрана монитора и рассчитываться автоматически. Если у одних столбцов будет задан один символ *, а у других - два, то вторые будут в два раза шире первых. Если сумма ширин столбцов, значения которых заданы

Название	Описание
	<p>числовыми настройками или Auto, окажется больше ширины справочника, то в клиенте визуализации в нижней части таблицы появится линейка прокрутки, а на столбцы, у которых ширина задана символами *, будет выделено минимальное количество пикселей, достаточное только для того, чтобы обеспечить возможность растягивания таких столбцов для просмотра их содержимого.</p>
Сортировка	<p>Определяет тип сортировки в столбце в таблице данных в клиенте визуализации по умолчанию. Может принимать значения:</p> <ul style="list-style-type: none"> • Нет - сортировка в столбце не используется. • По возрастанию - сортировка в столбце используется, первыми в таблице располагаются строки, у которых поле имеет меньшее значение. • По убыванию - сортировка в столбце используется, первыми в таблице располагаются строки, у которых поле имеет большее значение.
Порядок сортировки столбцов	<p>Определяет приоритет сортировки столбцов в случае, когда необходима сортировка по нескольким столбцам (множественная сортировка). Принимает числовые значения (0,1,2, и т.д.) Например, если у двух столбцов используется сортировка, то сначала происходит сортировка по тому столбцу, в котором значение свойства Порядок сортировки столбцов ниже.</p> <p>В клиенте визуализации можно менять множественную сортировку, кликая по столбцам левой кнопкой мыши с нажатой на клавиатуре клавишей Ctrl.</p>
Формат значений	<p>Определяет формат отображения выводимого значения. Описание этого свойства смотрите в разделе Свойство Формат значений.</p>
Видимость	<p>Флаг, определяющий видимость столбца. Если флаг снят, то для соответствующего столбца место в таблице не выделяется. Данное свойство имеет смысл динамизировать, используя панель свойств элемента.</p>

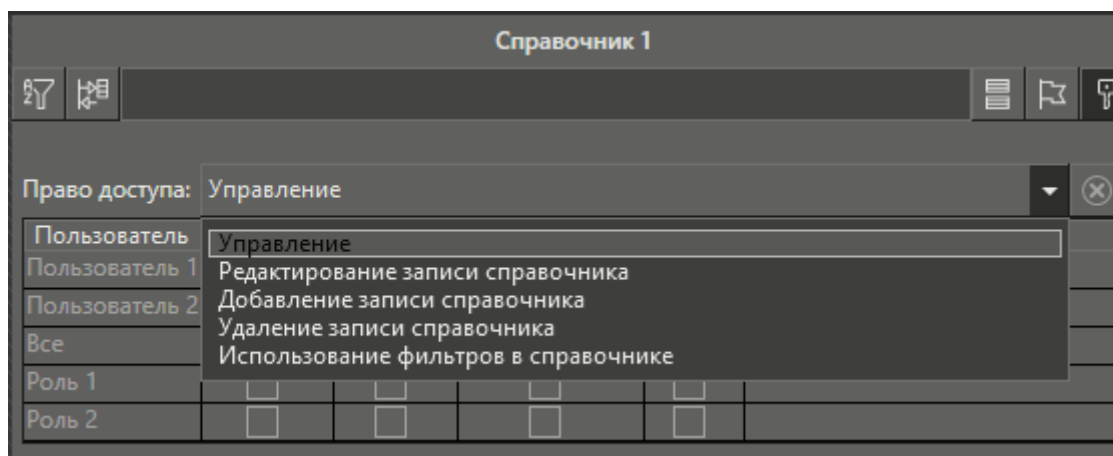
Название	Описание
Гориз. выравнивание	Определяет выравнивание заголовка и самого столбца.
Стиль ячеек столбца	Не используется в текущей версии
Включить фильтрацию столбца	Включает фильтрацию столбца в клиенте визуализации.
Запрет редактирования	Определяет возможность редактирования ячеек столбца в клиенте визуализации.

8.3.2.9.2. СПРАВОЧНИК. ПРАВА ДОСТУПА

Для работы со Справочником в режиме исполнения предусмотрены специальные права доступа.

В дереве системы в группе Безопасность настраиваются умолчания для всех добавляемых в проект Справочников. У каждого отдельного Справочника в его панели свойств можно изменить сделанные умолчания.

Вид панели свойств Справочника в режиме редактирования Прав:



Доступные для настройки права доступа:

Право	Назначение
Управление	Определяет возможность пользователя работать со справочником в режиме исполнения. Если право не предоставлено, то справочник будет полностью отключен (полностью недоступен для управления).

Право	Назначение
	Пользователь не сможет воспользоваться панелью инструментов и другими инструментами.
Редактирование записи справочника	Определяет право пользователя редактировать строки таблицы. Если установлен флаг Журнал, то в журнале сформируется сообщение Изменение записи : <поле1>=<значение1>,<поле2>=<значение2>..." (перечисляются новые измененные значения полей записи)
Добавление записи справочника	Определяет право пользователя добавлять новые строки в таблицу. Если установлен флаг Журнал, то в журнале сформируется сообщение Добавление записи : <поле1>=<значение1>,<поле2>=<значение2>..." (перечисляются значения полей добавляемой записи)
Удаление записи справочника	Определяет право пользователя удалять строки из таблицы. Если установлен флаг Журнал, то в журнале сформируется сообщение Удаление записи : <поле1>=<значение1>,<поле2>=<значение2>..." (перечисляются значения полей удаляемой записи)
Использование фильтров в справочнике	Определяет право пользователя использовать фильтры в таблице справочника. Если установлен флаг Журнал, то в журнале сформируются сообщения Включена фильтрация столбца <имя столбца> : <параметры фильтрации>", "Отключена фильтрация столбца <имя столбца>

8.3.2.9.3. ПРИМЕР РАБОТЫ СО СПРАВОЧНИКОМ

Постановка задачи:

Заносить показания приборов на объектах в зависимости от тарифа. Для этого в базе данных MSSQL есть четыре таблицы:

- Address с адресами объектов,
- Models с наименованиями приборов,
- Rates с текущими тарифами,
- Main - показания.

Таблицы. Первичный и внешний ключ. Представления

Сначала обозначим требования к таблицам в MSSQL. В таблице должен быть столбец - первичный ключ (Primary key) с выставленной настройкой Спецификация идентификатора.

Таблица Address

В таблице Address это столбец idAddress, который будет увеличиваться на единицу при создании новой записи в базу. Это означает, что следующая запись в таблице будет иметь новое значение idAddress вне зависимости от удаления ранее добавленных записей. Первичные ключи нужны для связывания таблиц с помощью диаграмм баз данных.

Также в этой таблице есть другие столбцы, которые будем заполнять в уже работающем проекте:

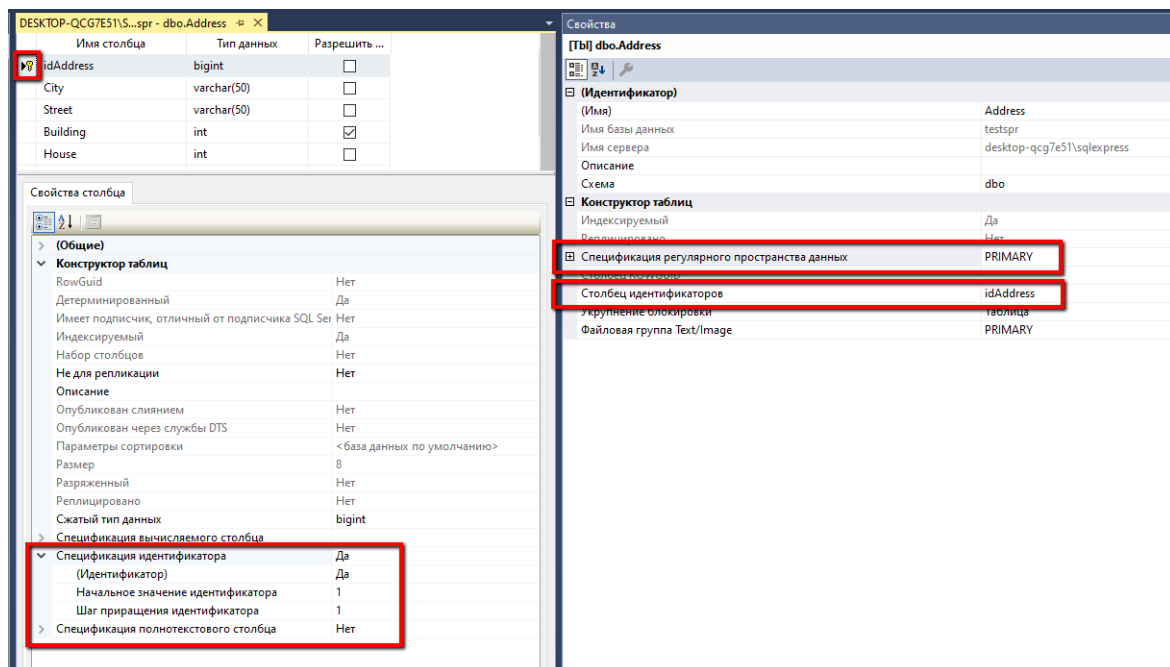
City - город,

Street- улица,

Building- корпус ,

House - номер дома .

Вид таблицы данных Address в MSSQL, где отображен столбец с первичным ключом и идентификатором.



Столбцу Building разрешено использовать нулевые значения, так как корпуса здания может и не быть:

Таблица Models

В таблице списка приборов Models первичным ключом с идентификатором является столбец IDModel, а также другие столбцы:

Name - это название прибора,

Number - его регистрационный номер - Number.

Вид таблицы:



	Имя столбца	Тип данных	Разрешить ...
	IDModel	int	<input type="checkbox"/>
	Name	varchar(50)	<input type="checkbox"/>
	Number	varchar(50)	<input type="checkbox"/>

Таблица Rates

В таблице тарифов Rates первичный ключ с идентификатором idRates и столбцы: Rate, Value1, Value2:



	Имя столбца	Тип данных	Разрешить ...
	idRates	int	<input type="checkbox"/>
	Rate	varchar(50)	<input type="checkbox"/>
	Min	int	<input checked="" type="checkbox"/>
	Max	int	<input checked="" type="checkbox"/>

Таблица Main

Главная таблица имеет следующие столбцы:

idMain - первичный ключ и идентификатор,

DateMeter - время снятия показания,

Meter 1, Meter 2 - значения самих показаний.

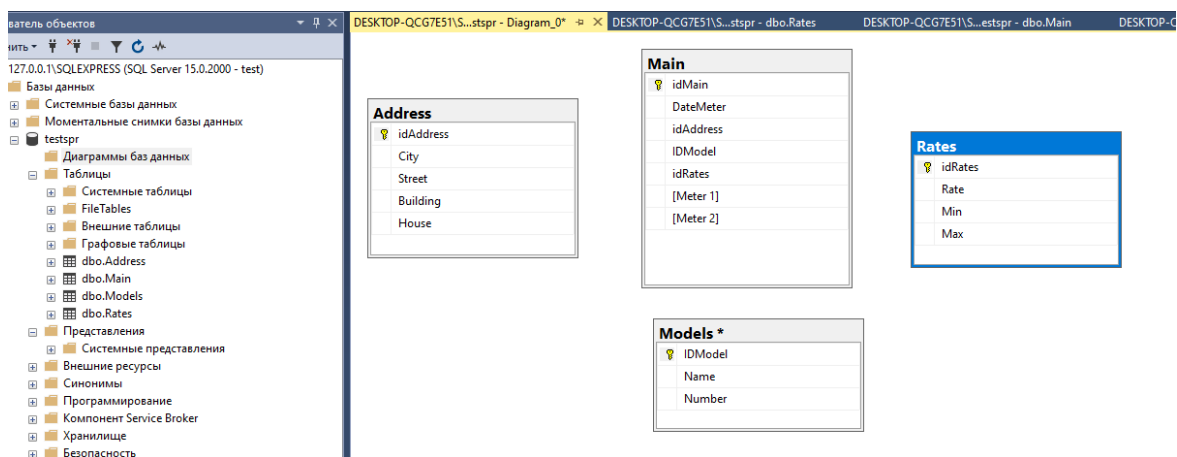
Важно! Остальные столбцы имеют те же названия, что и столбцы с первичным ключом остальных таблиц. Это сделано для того, чтобы в момент задания значений в главной таблице мы могли использовать и отображать данные из других таблиц.

Вид таблицы:

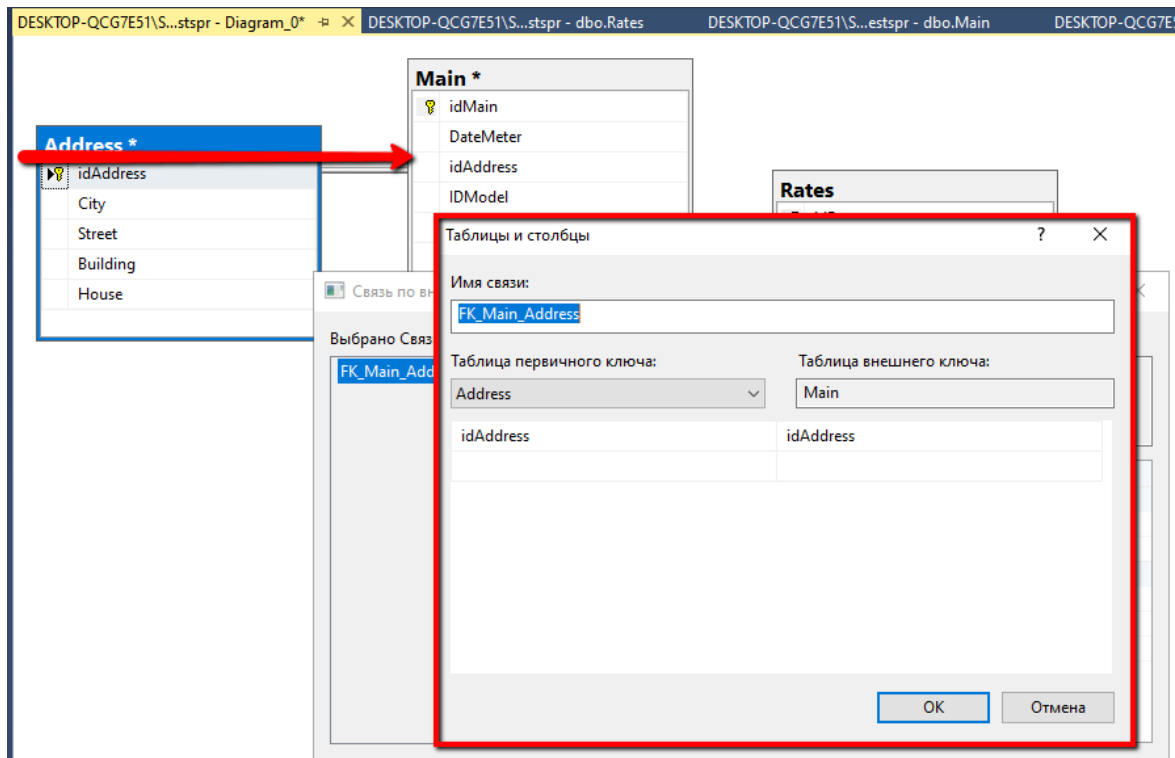
	Имя столбца	Тип данных	Разрешить ...
🔑	idMain	int	<input type="checkbox"/>
	DateMeter	datetime	<input type="checkbox"/>
	idAddress	bigint	<input type="checkbox"/>
	IDModel	int	<input type="checkbox"/>
	idRates	int	<input type="checkbox"/>
	[Meter 1]	int	<input checked="" type="checkbox"/>
▶	[Meter 2]	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Создание диаграммы

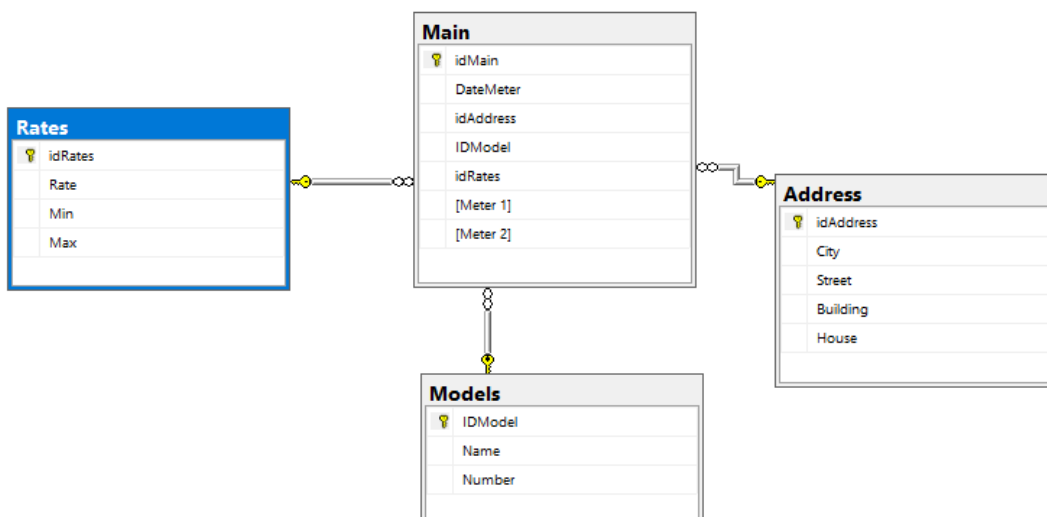
Для того, чтобы в момент задания значений в главной таблице можно было отображать данные из других таблиц необходимо таблицы добавить в диаграмму баз данных:



Связать столбцы одной таблицы с другой. Установить связи между внешним ключом одной таблицы и первичным ключом другой таблицы:



После этого столбцы таблицы Main, с которой была установлена связь, приобретут свойство внешнего ключа. Вид связанных таблиц в диаграмме базы данных:



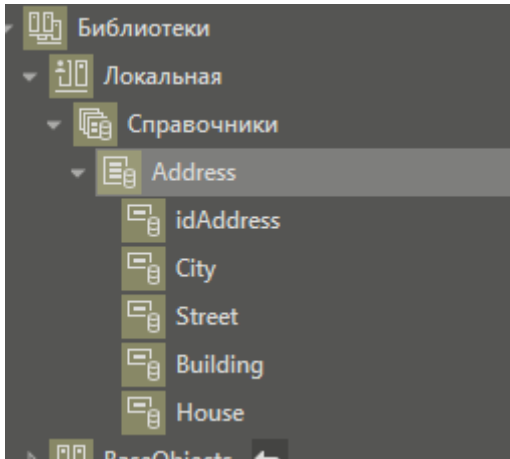
Таблицы для примера готовы. Можно перейти к их добавлению в проект MS4D.

Справочник как тип данных

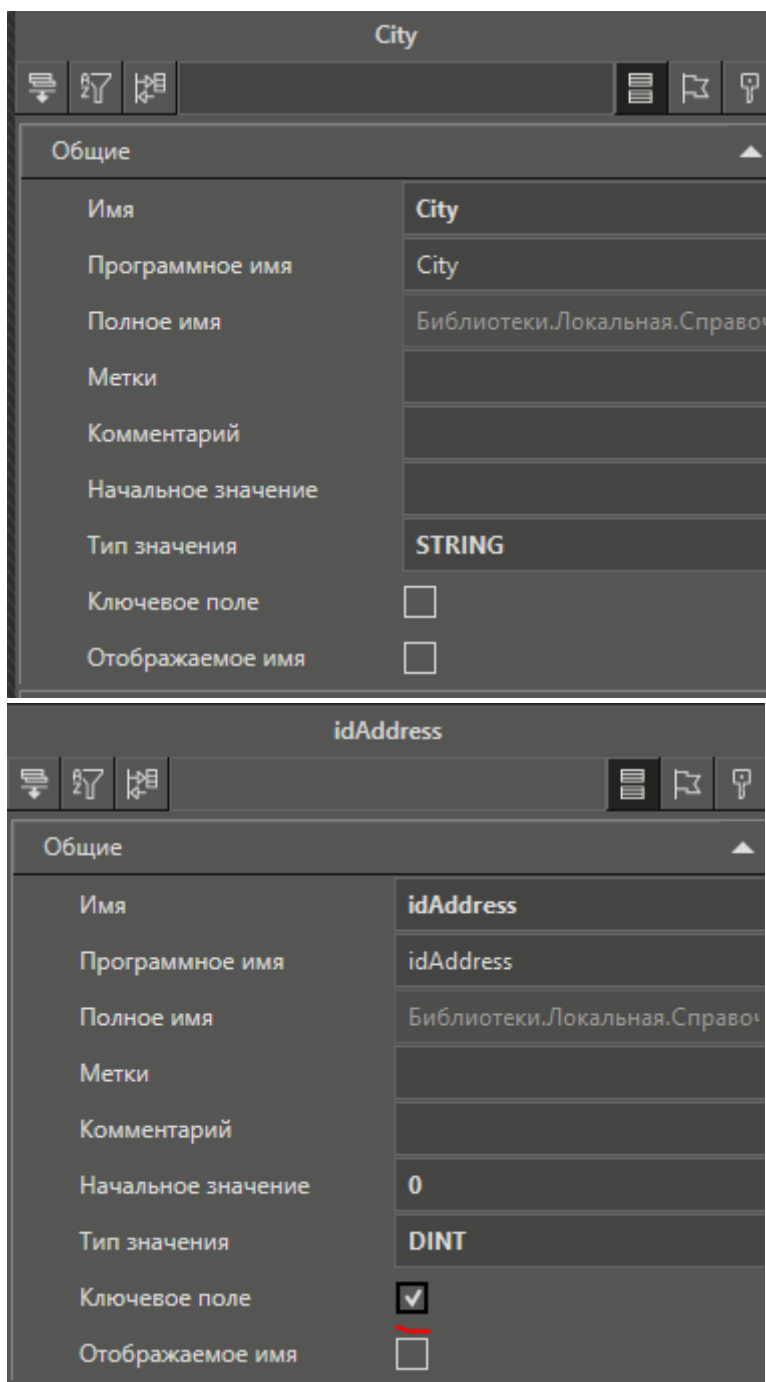
Для работы справочников в проекте MasterSCADA 4D нужно завести соответствующий элемент в пользовательской библиотеке. Элемент называется Справочник. Создаваемый в

библиотеке элемент должен называться точно также как таблица в базе данных, а поля справочника иметь такой же тип данных как столбцы этой таблицы. Для столбца с первичным ключом обязательно должна быть выставлена настройка Ключевое поле. Настройка Отображаемое имя требуется для отображения значения столбца удаляемой строки, а также отображения отладочного значения параметра типа Справочник, об этом позже.

Вид справочника после изменения имени и добавления полей:

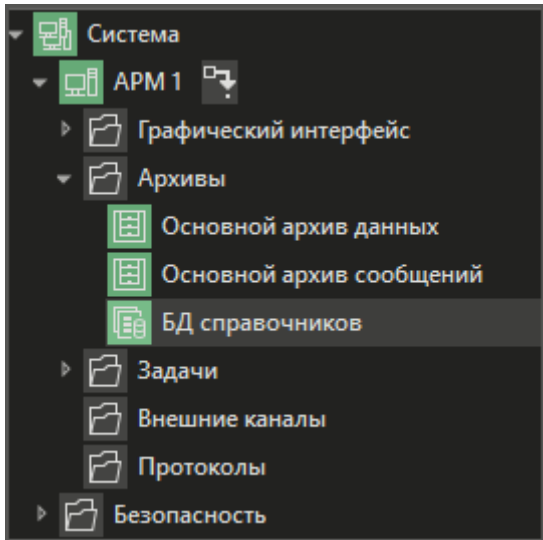


Панели свойств некоторых полей справочника:

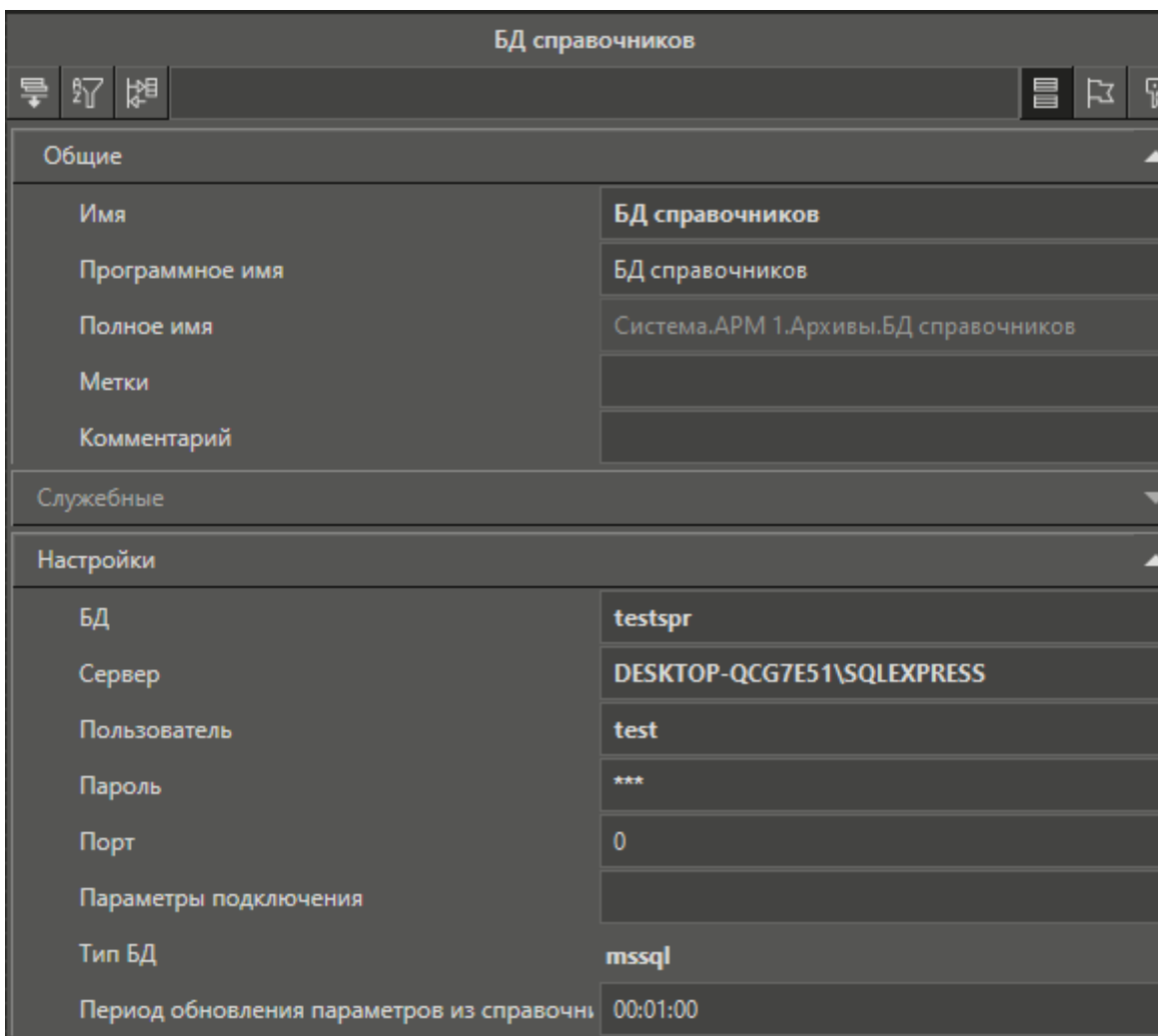


БД справочников.

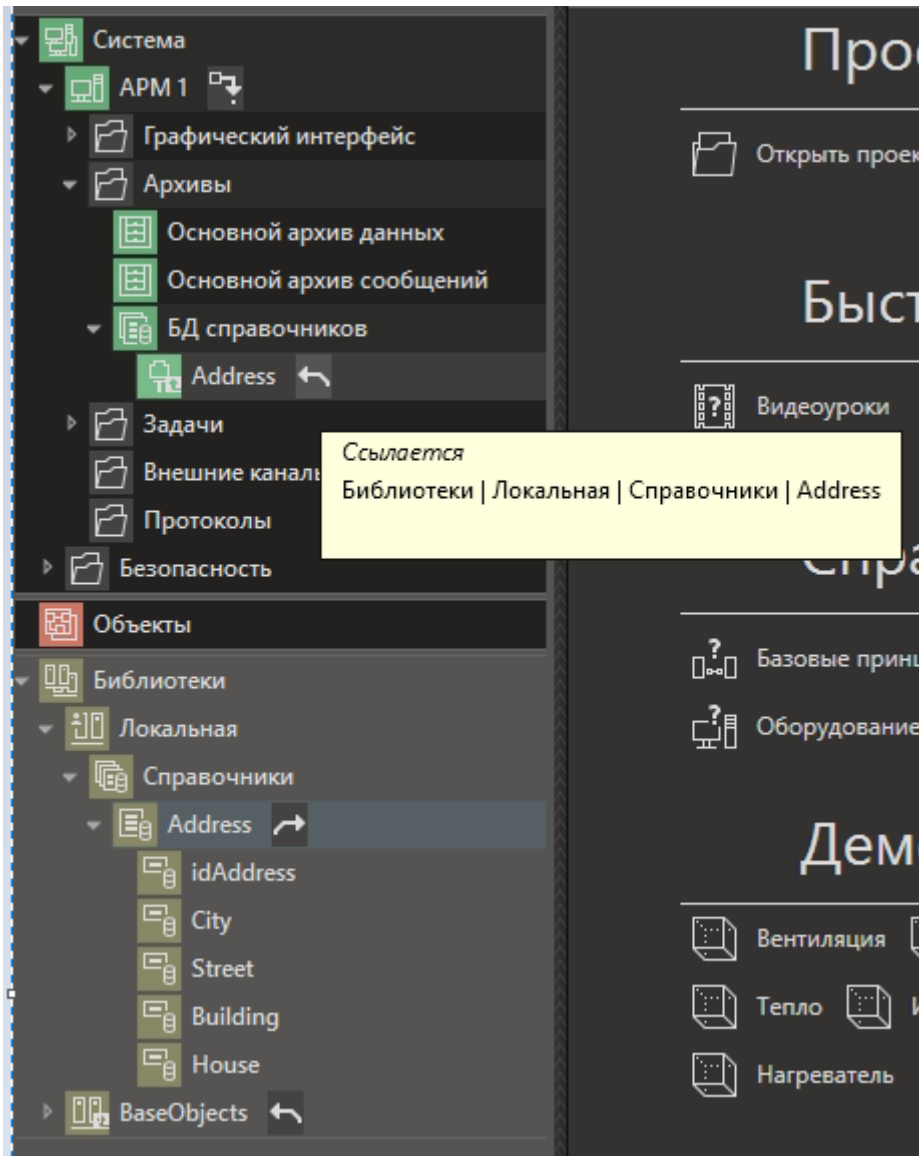
Для установления связи с сервером MSSQL нужно добавить БД справочников в папку Архивы узла:



В панели свойств указывается имя сервера, логин, пароль и другие параметры подключения:



Далее необходимо установить связь между созданным ранее Справочником в библиотеке и данным элементом. Для этого методом Drag&Drop нужно перетащить Справочник из дерева библиотек в БД справочников. В дереве системы появится ссылка:

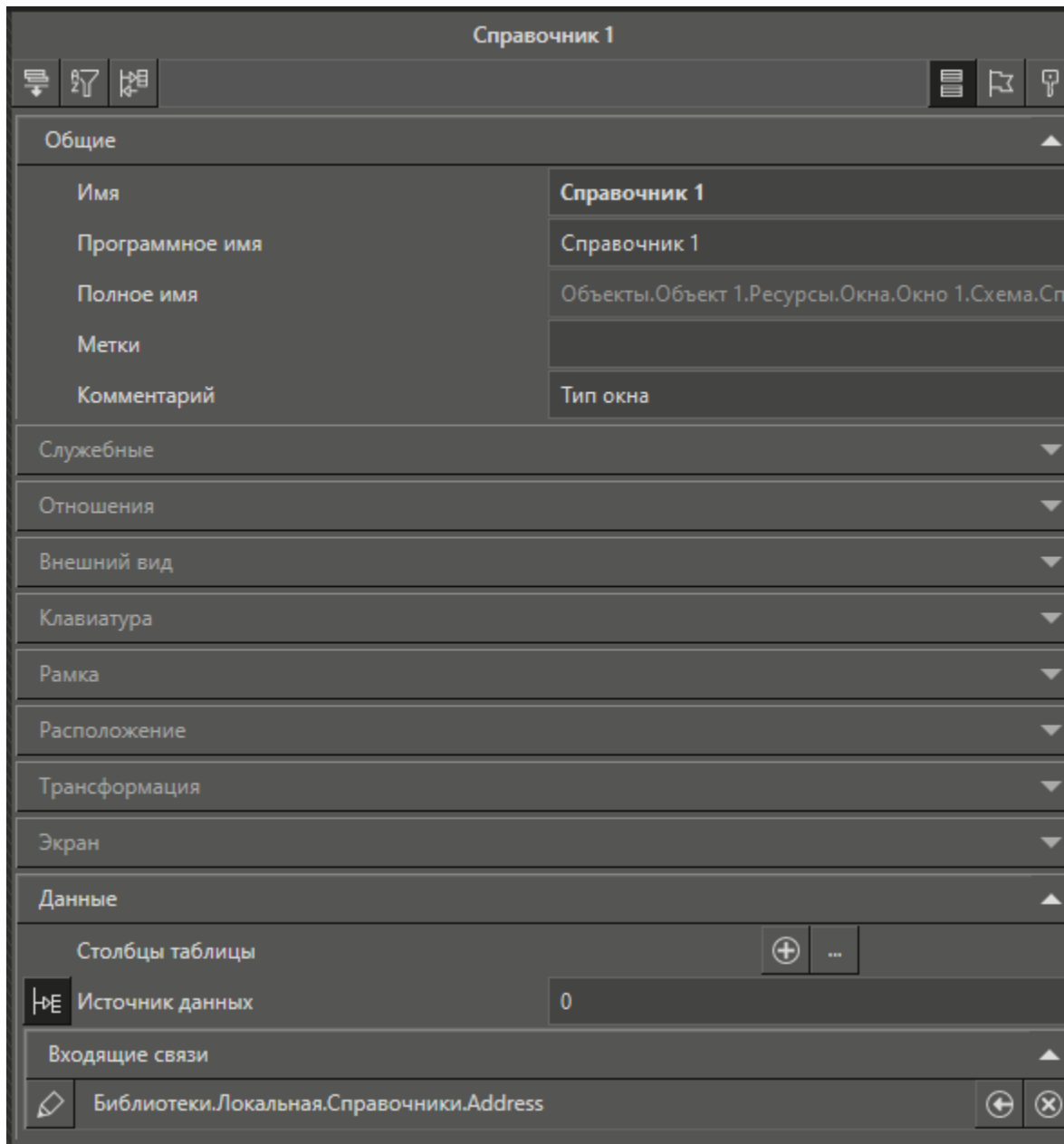


При необходимости нужно установить связь со всеми справочниками, которые необходимы для работы проекта

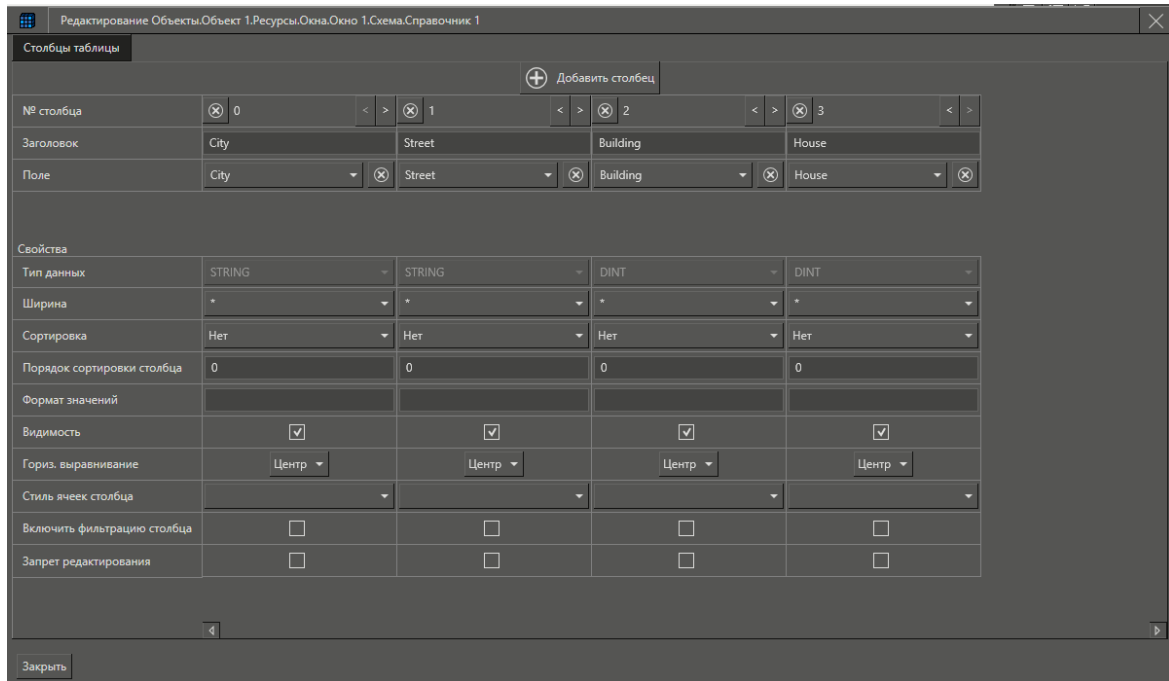
Графический элемент справочник

Для работы с таблицей данных в графическом клиенте нужен контрол Справочник.

В дереве объектов создадим объект, окно, назначим это окно стартовым. В это окно вытащим из категории палитры Контролы графический элемент Справочник. В панели свойств найдем параметр Источник данных и на него перетащим библиотечный элемент Справочник. Появится связь:



Откроем редактор справочника с помощью кнопки, расположенной по центру элемента. Добавим несколько столбцов и в строке Поле выберем нужные поля справочника:

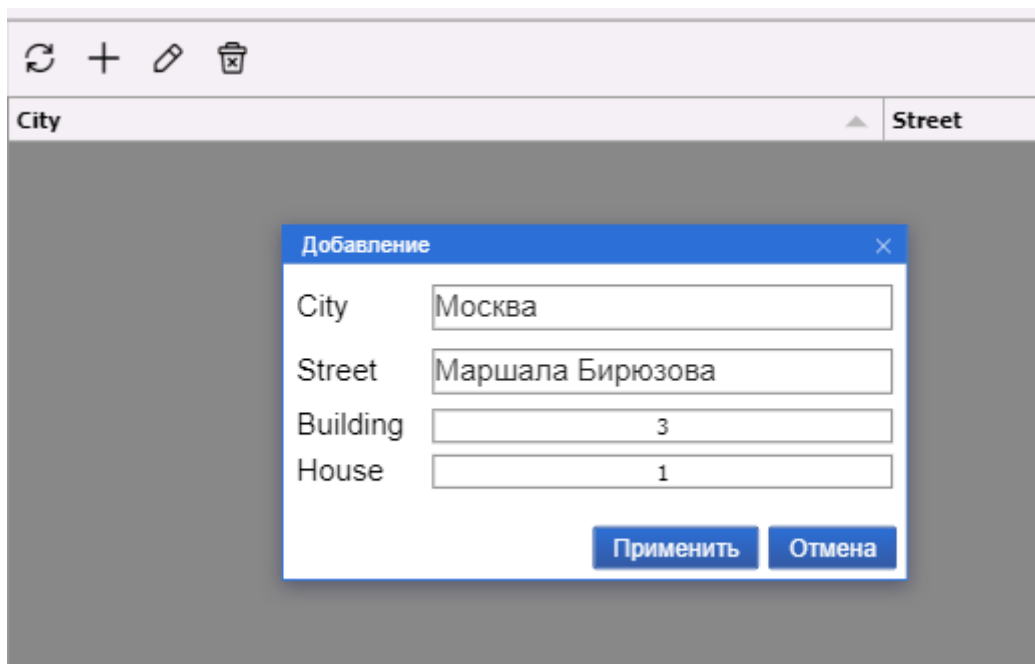


Работа со справочником в клиенте визуализации

Запустим режим исполнения с загрузкой конфигурации. Если данных в базе данных нет, то изначально отобразится пустая таблица:



Нажмем на кнопку "+". После этого появляется всплывающее окно, с помощью которого можно добавлять строки в базу:



После нажатия кнопки Применить запись появится как в справочнике, так и в базе данных.

Вид справочника:

City	Street	Building	House
Москва	Маршала Бирюзова	3	1
Норильск	Кирова		37
Обнинск	Студенческий городок	1	15
Москва	Москворечье	2	2

Вид БД:

SQLQuery1.sql - 127....S.testspr (test (57))

```

/***** Скрипт для команды SelectTopNRows из среды SSMS *****/
SELECT TOP (1000) [idAddress]
, [City]
, [Street]
, [Building]
, [House]
FROM [testspr].[dbo].[Address]

```

100 %

Результаты Сообщения

	idAddress	City	Street	Building	House
1	1	Москва	Маршала Бирюзова	3	1
2	2	Норильск	Кирова	0	37
3	3	Обнинск	Студенческий городок	1	15
4	4	Москва	Москворечье	2	2

Каждую запись можно редактировать и удалять, для этого можно воспользоваться панелью инструментов справочника:

Редактирование

City: Норильск

Street: Кирова

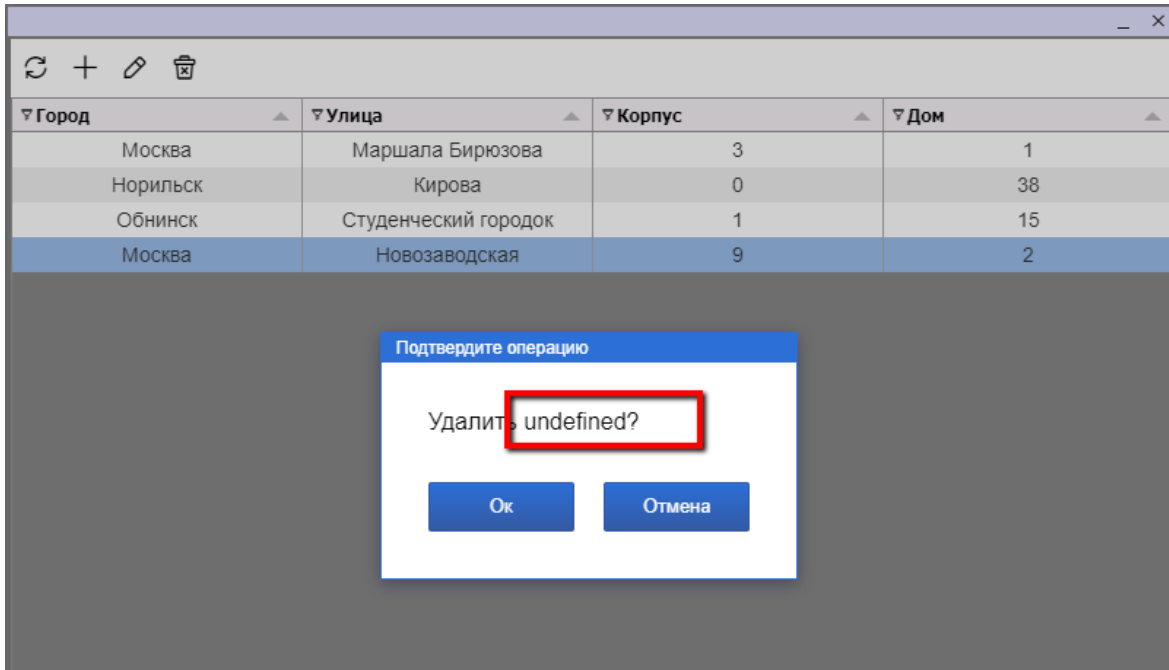
Building:

House: 38

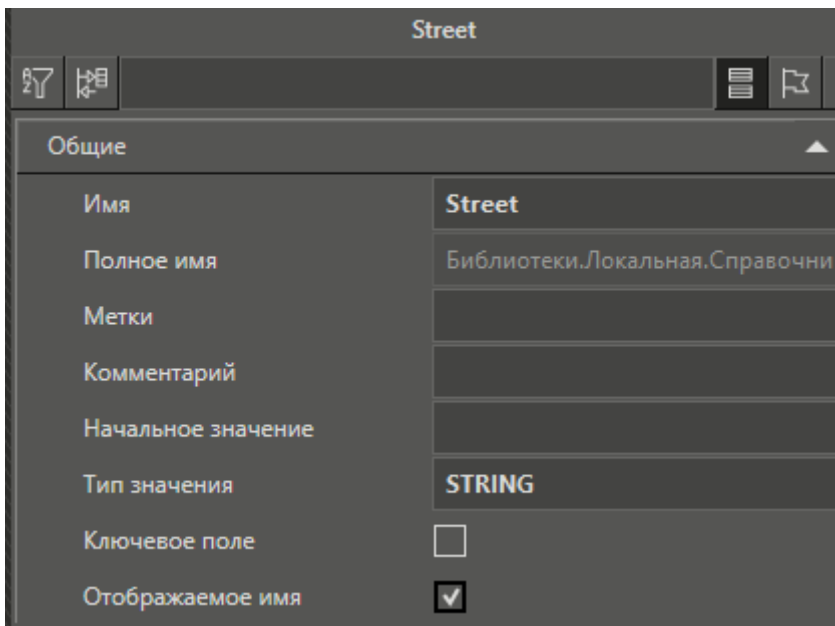
Применить Отмена

	1
	37
	15
	2

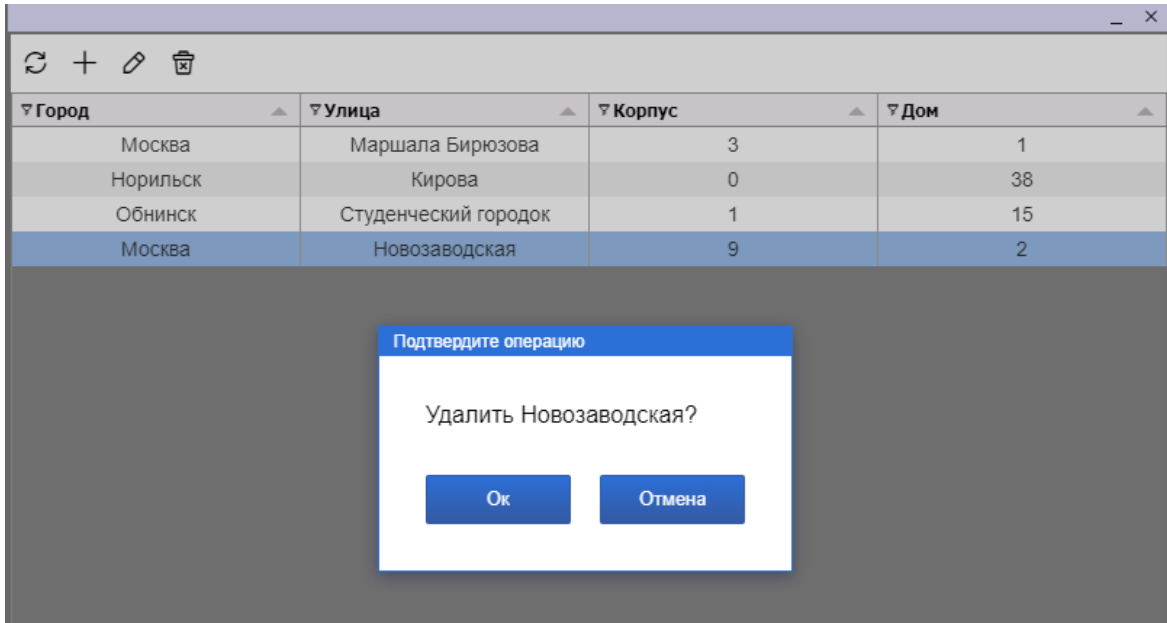
При попытке удалить строку из базы будет отображаться сообщение:



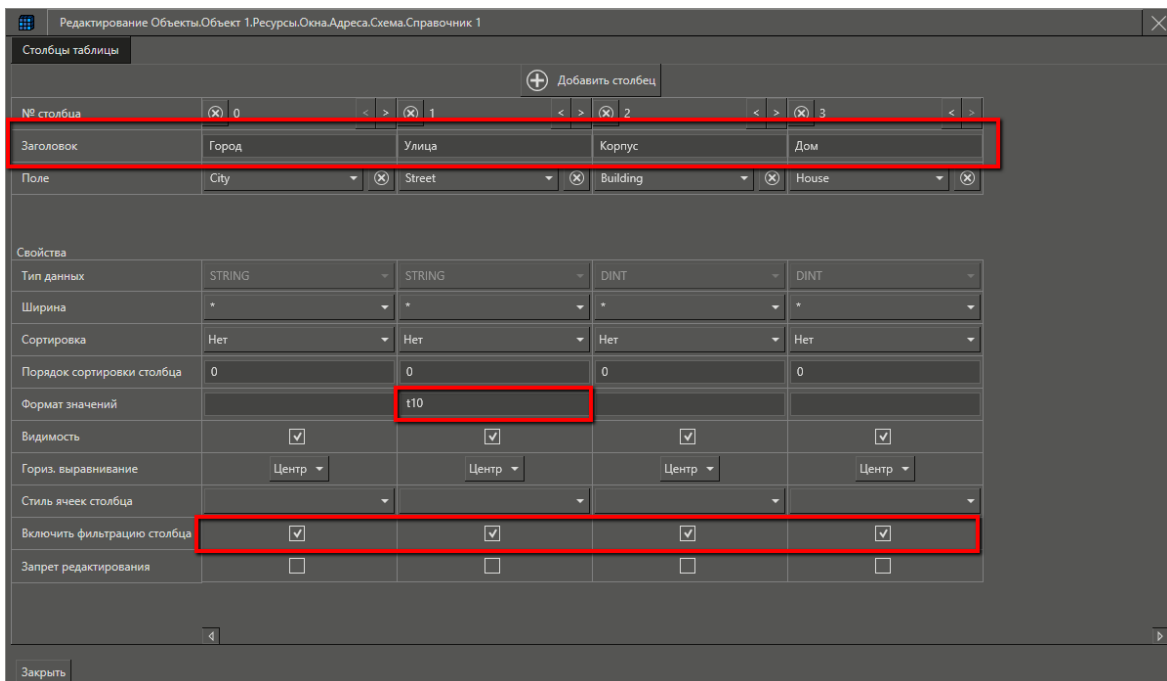
Слово undefined можно заменить на значение столбца удаляемой строки. Это делается с помощью настройки поля **Отображаемое имя**:



После этого в режиме исполнения при удалении строки будет отображаться нужное значение:



Внутри редактора справочника также можно менять различные настройки. Например, менять имена столбцов, включать/отключать фильтрацию значений столбцов, а также ограничить количество вводимых/выводимых значений с помощью формата значений tN, где N - количество вводимым символов:



В результате заголовок столбца отличается от имени поля справочника:

Город	Улица	Корпус	Дом
Обнинск	Студенческ	1	15
Норильск	Кирова	0	38
Москва	Маршала Би	3	1

Можно применить фильтр к столбцу, чтобы выделить только нужные значения в таблице:

Город	Улица	Корпус	Дом
Москва	Ма		1

Filter for Город

Применить Отмена

Значение столбца Улица ограничено выставленным форматом значений в 10 символов t10:

Город	Улица	Корпус	Дом
Обнинск	Студенческ	1	15
Норильск	Кирова	0	38
Москва	Маршала Би	3	1

Сделаем такие же Справочники и перенесем их в БД справочников для таблиц Models и Rates. В описываемом проекте окна со справочниками Адреса, Модели и Тарифы открываются во всплывающих окнах, так как на стартовом окне будет отображаться справочник Main:

The screenshot shows the 'Справочники' (Reference) window with three data tables:

- Адреса (Addresses):**

Город	Улица	Корпус	Дом
Москва	Маршала Бирисова	3	1
Норильск	Кирова	0	38
Обнинск	Студенческий городок	1	15
- Модели (Models):**

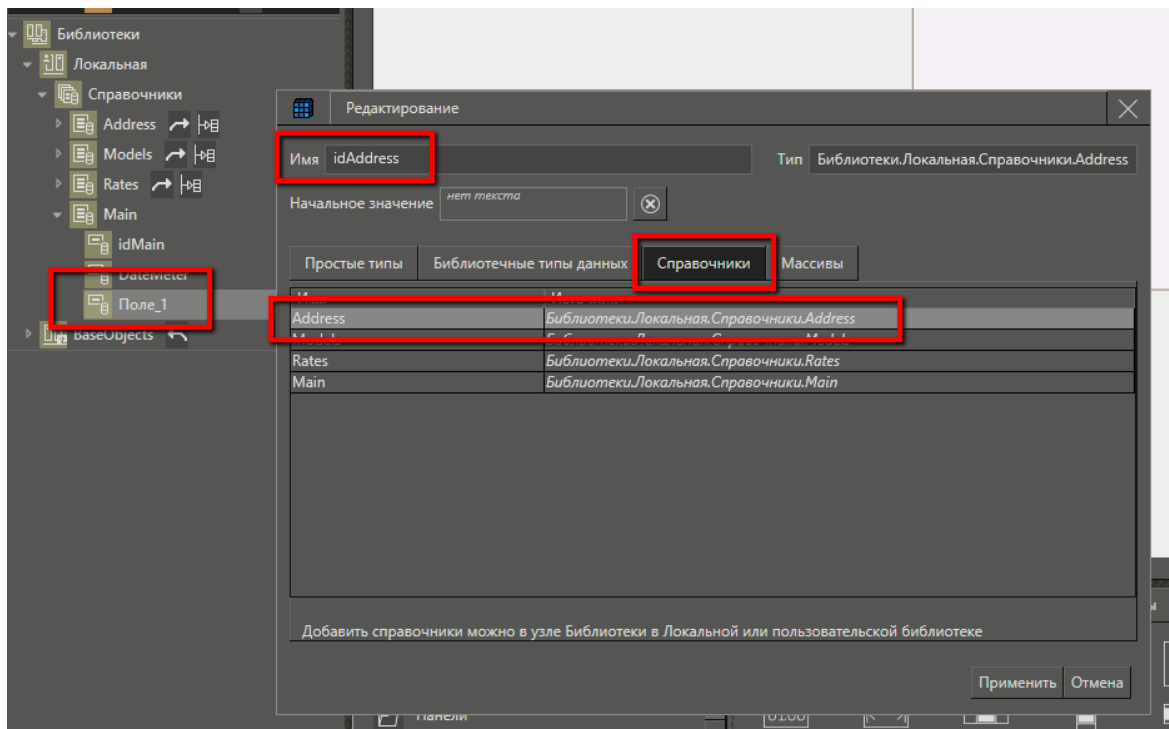
Название прибора	Регистрационный номер
Меркурий 230	550-043
Меркурий 230	550-041
Меркурий 230	550-601
Меркурий 232	551-012
Меркурий 232	551-013
Меркурий 230	550-791
- Тарифы (Rates):**

Тариф	День	Ночь
Зима 1	70	30
Зима 2	50	50
Лето 1	60	20
Лето 2	40	40

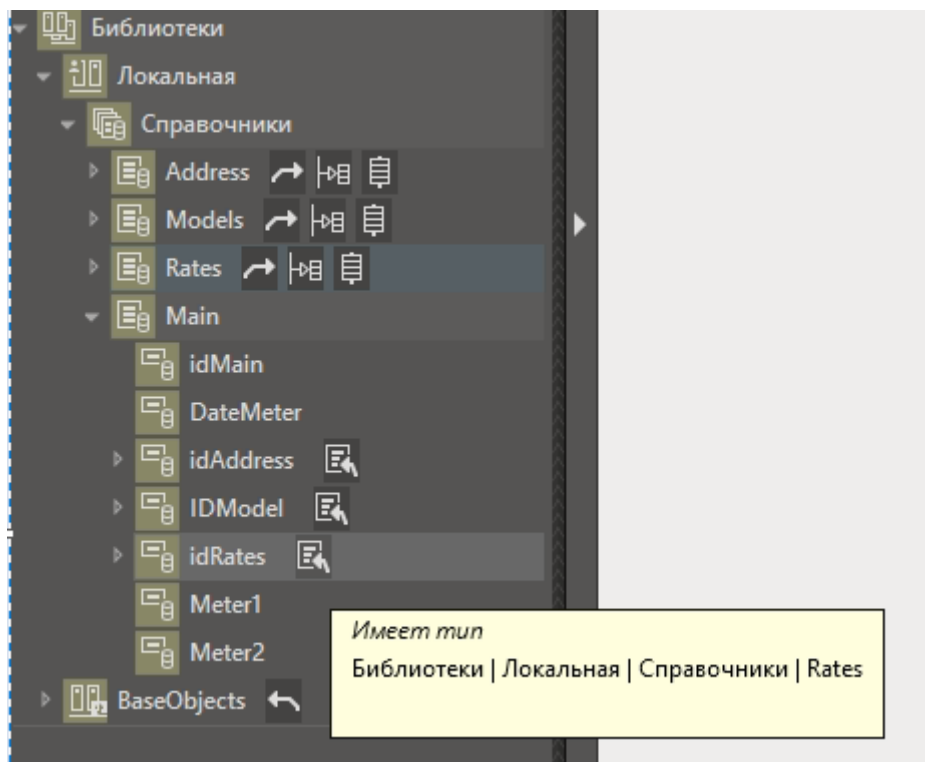
Отображение данных из других справочников

В главном справочнике Main мы будем вводить данные показаний счетчиков, при этом адрес, наименование прибора и данные по тарифу выбирать из уже готового списка, т.е. пользователю системы вводить все данные не потребуется. Создадим элемент Справочник Main аналогично остальным, но, так как эта таблица имеет связи с другими таблицами, типы

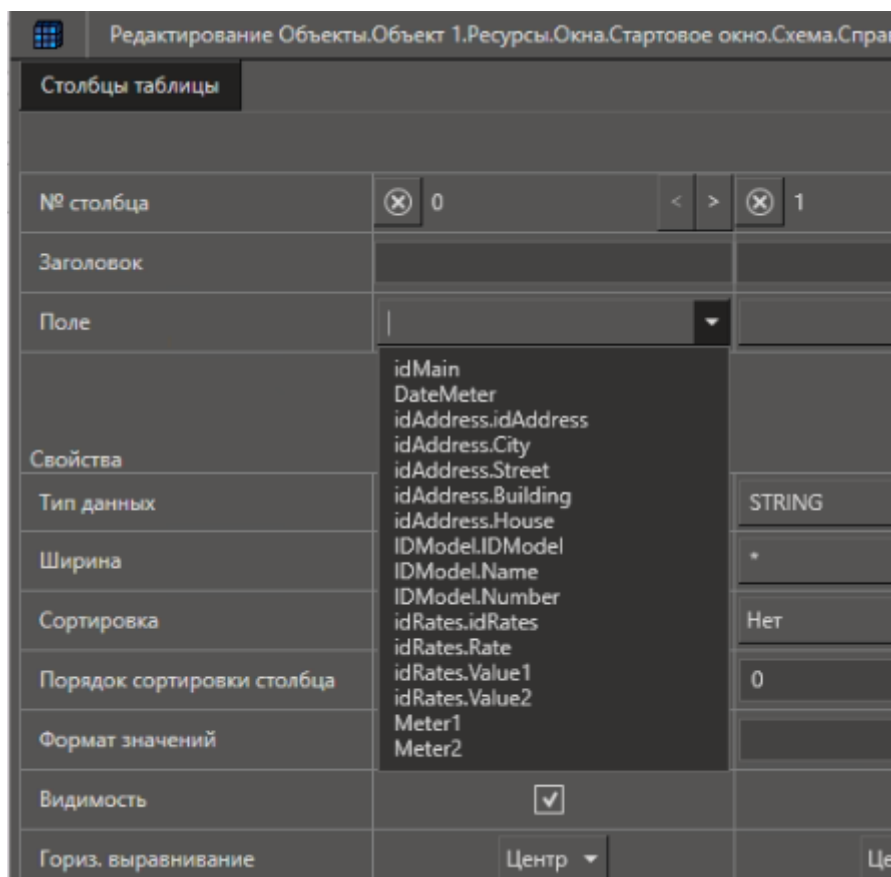
данных полей этого справочника должны иметь тип связанного с этим полем с помощью диаграммы другого справочника, например, со справочником Address



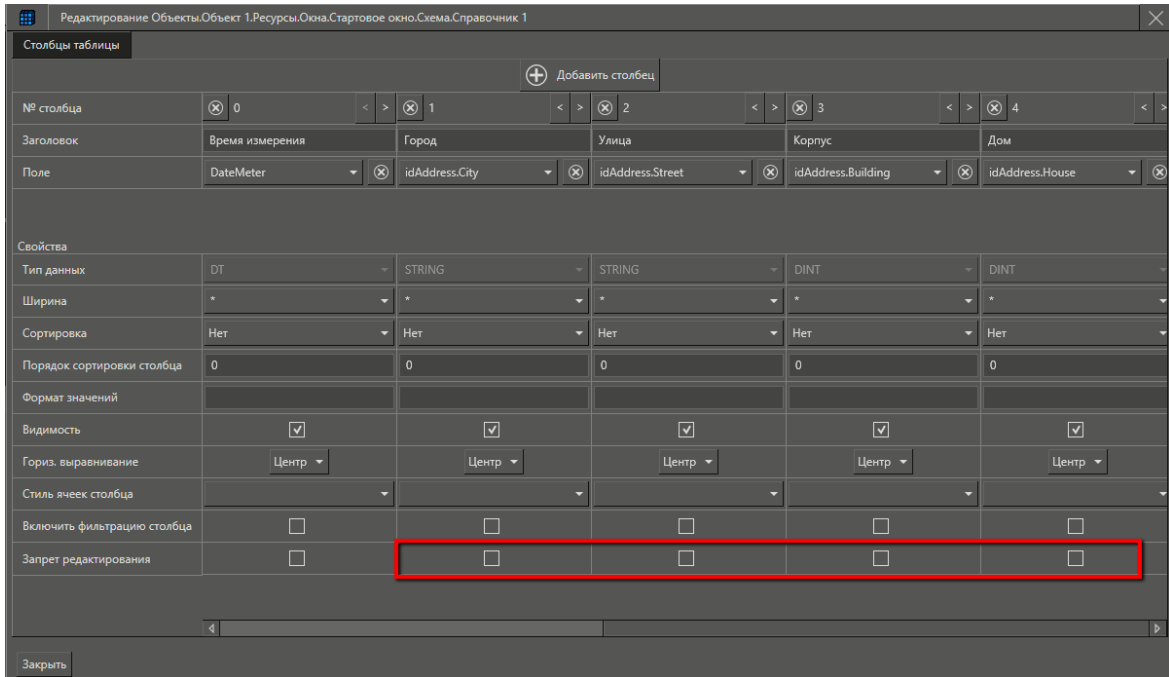
В результате справочник в пользовательской библиотеке будет выглядеть следующим образом:



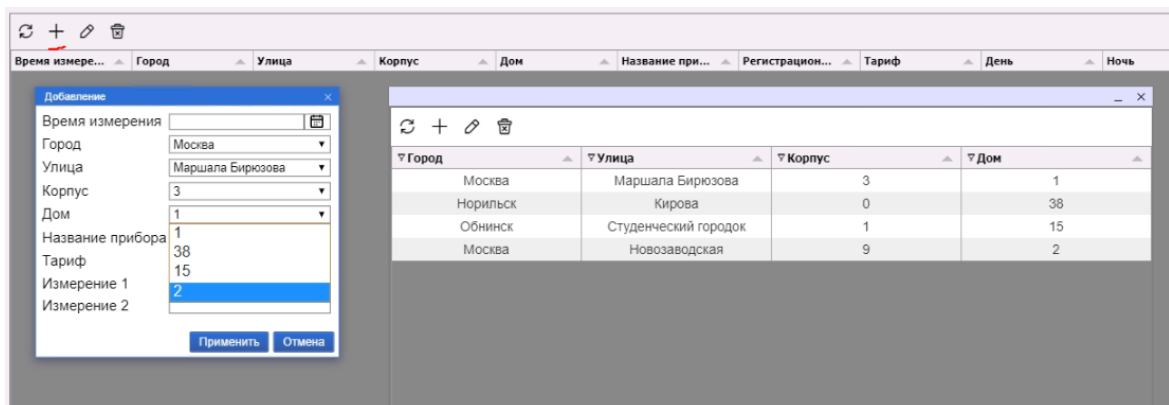
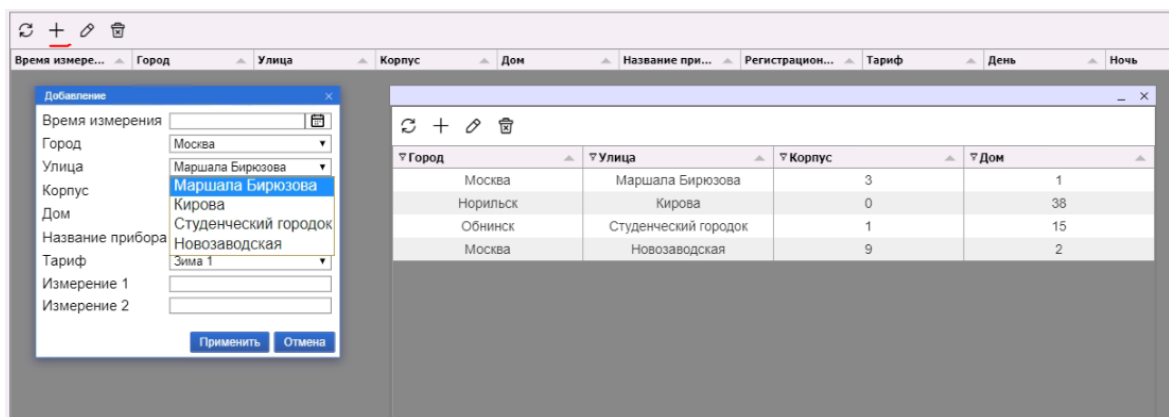
Переносим справочник Main в БД справочников. На стартовое окно из палитры вытаскиваем графический элемент Справочник, связываем источник данных этого справочника со справочником Main в библиотеке, открываем редактор справочника. Так как поля этого справочника имеют типы данных других справочников, то в поле столбца будут отображаться поля всех справочников:



Задавая поле, мы выбираем столбцы, которые хотим отобразить. В этом случае стоит учитывать, что если столбцы оставить для редактирования, то при добавлении новой строки в общую таблицу остается возможность добавлять значения из разных строк других связанных таблиц. На следующих рисунках явно отображается эта ситуация: в справочнике Main у столбцов City, Street, Building и House из справочника Address не выставлена настройка Запрет редактирования. Это приведет к тому, что в режиме исполнения при добавлении новой записи можно выбрать значения из разных строк таблицы Address

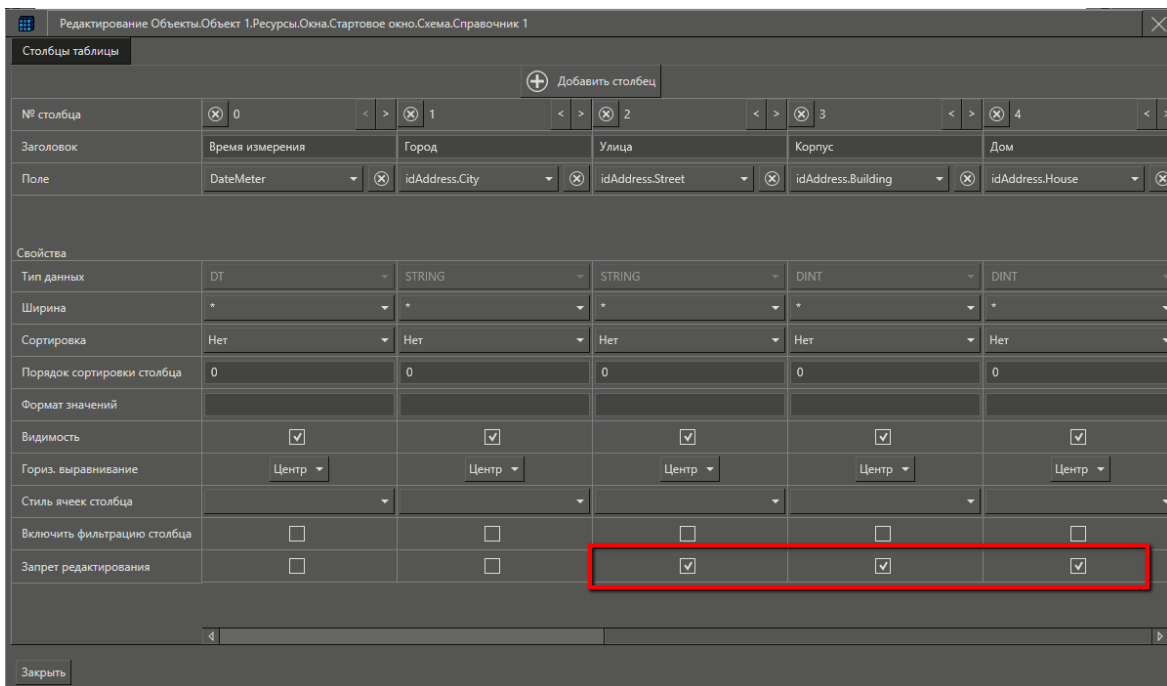


Вид окна Добавление к клиенте визуализации:



Т.е. в основную таблицу можно добавить как разные названия улицы, так и разные номера домов, хотя в этом примере, по условию так делать нельзя. Для того, чтобы можно было выбирать только значения одной строки, надо выставить настройку Запрет редактирования

у всех нужных столбцов. В этом примере это idAddress.Street, idAddress.Building, idAddress.House, IDModel.Number, idRates.Value1, idRates.Value2.



Стоит отметить, что во всех справочниках мы не отображаем столбцы id с первичным ключом и идентификаторов, так как в этом нет необходимости. В противном случае у этого столбца надо обязательно выставлять настройку Запрет редактирования.

В итоге после запуска режима исполнения мы получим таблицу в таком виде:

Время измере...	Город	Улица	Корпус	Дом	Название при...	Регистрацион...	Тариф	День	Ночь	Измерение 1	Измерение 2
27-04-2020 15:45...	Москва	Маршала Бирозова	3	1	Меркурий 232	551-012	Лето 1	60	20	31	55
28-04-2020 12:46...	Москва	Новозаводская	9	2	Меркурий 230	550-041	Лето 2	40	40	33	5
01-01-2021 00:00...	Норильск	Кирова	0	38	Меркурий 232	551-013	Зима 1	70	30	1	1

В самой таблице базы данных Main значения такие:

	idMain	DateMeter	idAddress	IDModel	idRates	Meter1	Meter2
1	1	2020-04-27 15:45:54.000	1	4	5	31	55
2	2	2020-04-28 12:46:33.000	6	2	6	33	5
3	3	2021-01-01 00:00:00.000	2	5	1	1	1

В справочнике видим данные из связанных таблиц. В самой же таблице Main в базу данных попали только введенные значения столбцов этой таблицы.

8.3.2.9.4. РАБОТА СО СПРАВОЧНИКОМ В КЛИЕНТЕ ВИЗУАЛИЗАЦИИ

Вид Справочника в клиенте визуализации:

Т Времена изме...	Т Город	Т Улица	Т Корпус	Т Дом	Т Название п...	Т Регистраци...	Т Тариф	Т День	Т Ночь	Т Измерение 1	Т Измерение 2
27-04-2020 15:45...	Москва	Улица Улица Улиц	3	1	Меркурий 230	550-043	Лето 1	55	35	31	55
28-04-2020 12:46...	Москва	Новозаводская	9	2	Меркурий 230	550-041	Лето 2	40	40	32	5
01-01-2021 00:00...	Норильск	Улица Улица Киро	0	30	Меркурий 232	551-013	Зима 1	70	30	1	1
28-04-2020 11:48...	Запорожье	Днепропетровская	0	17	Меркурий 230	550-043	Зима 1	70	30	888	44
13-05-2020 13:40...	Москва	Улица Улица Улиц	3	1	Меркурий 230	550-043	Лето 1	55	35	55	60
13-05-2020 13:42...	Москва	Улица Улица Улиц	3	1	Меркурий 230	550-043	Лето 1	55	35	44	77
13-05-2020 14:37...	Ухта	Советская	0	6	Меркурий 230	550-043	Лето 1	55	35	33	33
16-06-1971 00:00...	Норильск	Улица Улица Киро	0	30	Меркурий 230	550-043	Зима 1	70	30	0	0
21-10-2020 13:54...	Москва	Улица Улица Улиц	3	1	Меркурий 230	550-043	Зима 1	70	30	5	56

Панель инструментов

Элементы панели инструментов:








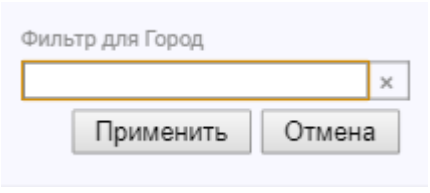
Элемент	Название	Назначение
	Обновить	Обновляет данные справочника.
	Добавить	Позволяет добавить новую строку в базу данных. После нажатия на кнопку откроется диалоговое окно (вид окна будет совпадать с видом окна для редактирования), в котором нужно ввести значения полей. Данные запишутся в базу данных после нажатия на кнопку окна Применить. Кнопка отображается, если в настройках справочника снят флаг в свойстве Запрет добавления. Действие совершится, если у пользователя есть соответствующие права доступа
	Редактировать	Позволяет отредактировать выделенную строку. После нажатия на кнопку откроется диалоговое окно, в котором нужно ввести новые значения полей (описание окна ниже). Данные запишутся в базу данных после нажатия на кнопку окна Применить. Кнопка отображается, если в настройках справочника снят флаг в свойстве Запрет редактирования. Действие совершится, если у пользователя есть соответствующие права доступа
	Удалить	Позволяет удалить выделенную строку. После нажатия на кнопку откроется диалоговое окно, в котором нужно подтвердить удаление. Команда будет передана в базу данных после нажатия на кнопку окна ОК. Кнопка отображается, если в настройках справочника снят флаг в свойстве Запрет удаления. Действие совершится, если у пользователя есть соответствующие права доступа
	Сброс	Отменяет настройки внешнего вида тренда, сделанные пользователем в клиенте визуализации.

Таблица справочника

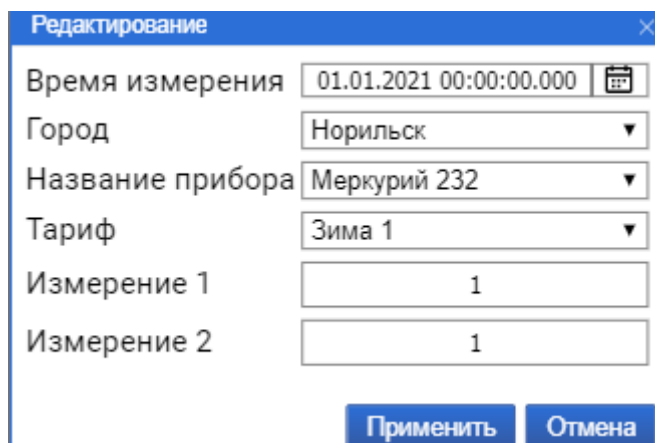
Элементы таблицы:

Элемент	Описание
Выделенная строка	Выделенная строка отмечена синим цветом. Для выделения строки нужно кликнуть по ней. Если дважды кликнуть по строке, то откроется диалоговое окно Редактирование.
Элементы Заголовка	
	Указатель сортировки. Располагается справа от заголовка столбца и показывает по какому столбцу таблицы осуществляется сортировка сообщений, а также направление сортировки - по убыванию или по возрастанию.
	<p>Открывает окно настройки фильтра, где можно указать, какие элементы необходимо отображать в таблице.</p> <p style="text-align: center;">Вид окна:</p>  <p>Кнопка отображается, если в настройках таблицы справочника установлен флаг для соответствующего столбца свойстве Включить фильтрацию столбца. Выборка производится среди тех элементов, которые поступили из базы данных. Действие совершится, если у пользователя есть соответствующие права доступа</p>

В таблице можно изменить последовательность столбцов. Для этого нужно нажать и удерживать левую кнопку мыши в заголовке того столбца, который нужно перенести, а затем переместить его на новое место и отпустить левую кнопку мыши.

Окно Редактирование/Добавление

Вид окна:



Время измерения	01.01.2021 00:00:00.000
Город	Норильск
Название прибора	Меркурий 232
Тариф	Зима 1
Измерение 1	1
Измерение 2	1

Применить Отмена

Внешний вид окна формируется автоматически. Для каждого поля, у которого в настройках не установлен флаг Запрет редактирования создается своя строка. В зависимости от типа данных поля будет зависеть элемент для ввода значений. Так например, для ввода даты и времени, будет добавлен в окно элемент ДатаВремя. А для ввода значений полей, соответствующих ссылкам на другие справочники, используется выпадающий список. Ссылки на поля внешних справочников редактировать нельзя. Если добавляется новая строка, при этом в таблице выведены не все возможные поля, то оставшимся полям устанавливаются значения по умолчанию. Если не выбрано ключевое поле, оно все равно читается с сервера (так как оно необходимо для сохранения изменений в таблице). Ключевое поле нельзя менять для существующих строк. Данные запишутся в базу данных после нажатия на кнопку окна Применить. Окно будет отображаться, если в настройках справочника снят флаг в свойстве Запрет редактирования, а также при наличии у пользователя прав доступа.

8.3.2.10. КАРТА

Элемент служит для отображения окон объектов MasterSCADA 4D на карте. Для этого необходимо настроить обращение к поисково-информационной картографической службе (тайловому серверу) и расположить на карте окна объектов (маркеры) и привязать их координаты. Поддерживается отработка различных действий, например, по клику мыши можно отобразить расширенную информацию по объекту.

Тайловый сервер - сервер, откуда можно получать тайлы.

Тайлы - фрагменты карты, из которых складывается полное изображение.

Принцип работы карты заключается в плиточном размещении специально подготовленных коллекций фрагментов карты для разных масштабов.

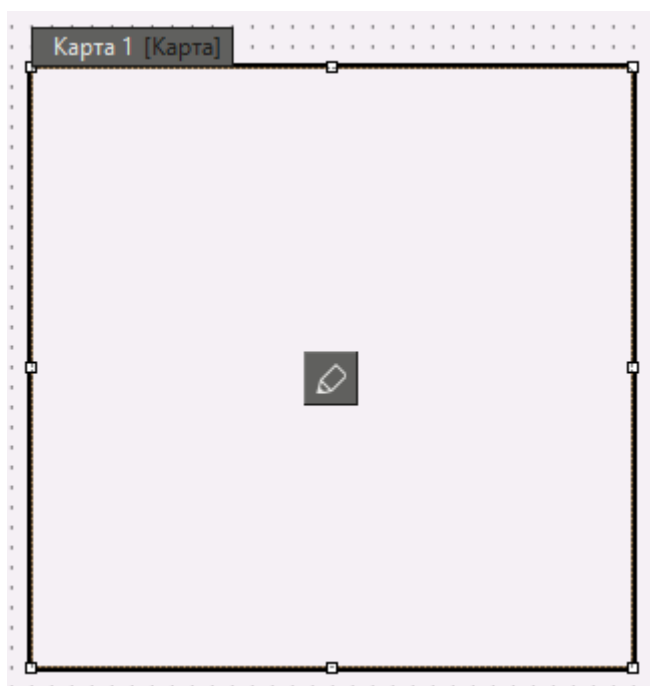
Важно! Использование тайловых серверов и фрагментов карт без соглашения с правообладателем - это нарушение авторских прав. Пример пользовательского соглашения рассмотрен в разделе Яндекс.Карты.

В палитре редактора HMI элемент Карта находится в категории Контролы.


Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:



Порядок работы с элементом Карта

1. После добавления элемента из палитры в окно необходимо настроить связь с тайловым сервером и указать основные настройки. Это можно сделать при помощи диалогового окна на вкладке Сервер, которое открывается при нажатии на кнопку , либо в категории Параметры панели свойств.
2. Настроить способ отображения маркеров на карте. Можно использовать стандартное отображение маркера, предусмотренное в программе по умолчанию, либо в качестве отображения можно использовать какое-либо окно, созданное ранее в библиотеке. Описания данного этапа работы смотрите в разделе Окна для отображения маркеров Карты.
3. Далее необходимо настроить связь между свойством карты Источник данных и массивом структур, где каждая структура описывает один маркер на карте. Порядок создания такого массива структур описан в разделе Источник данных для маркеров Карты

4. При необходимости можно настроить дополнительные действия, которые будут выполняться при клике по маркеру Карты.

5. Проверить работу в исполнительной системе.

8.3.2.10.1.РЕДАКТОР КАРТЫ

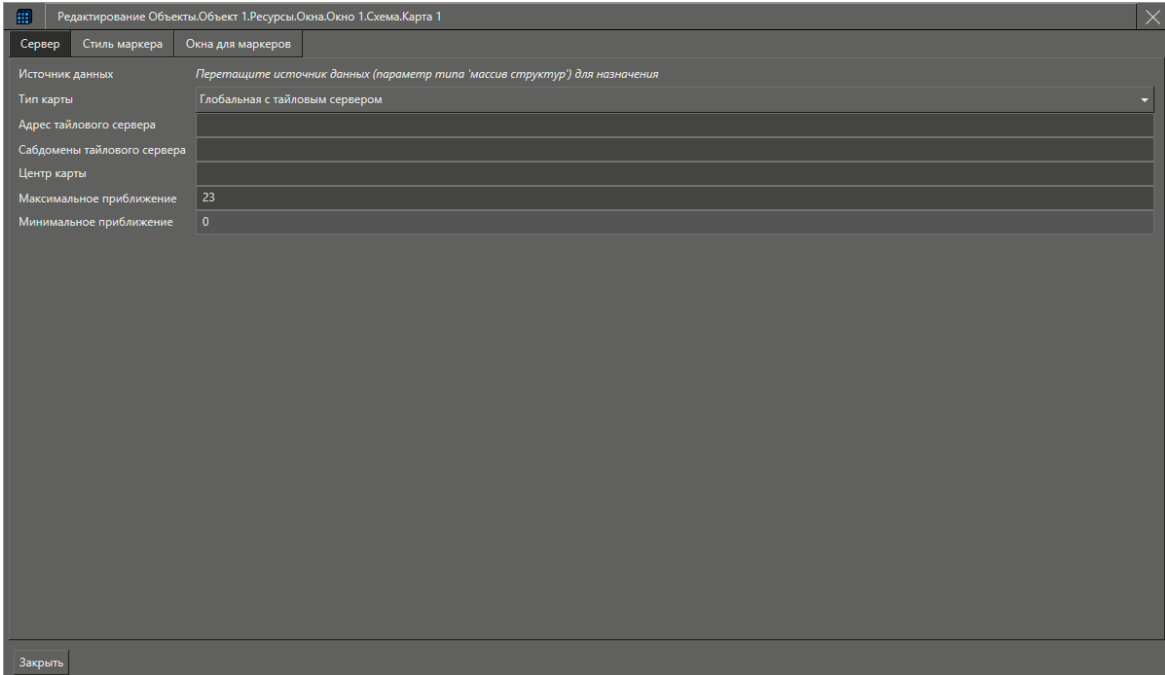
Редактор карты открывается при нажатии на кнопку  элемента.

Редактор состоит из трех вкладок:

Название	Назначение
Сервер	Настраиваются параметры связи с тайловым сервером
Стиль маркера	Определяется зависимость между внешним видом маркеров и значениями полей связанного источника данных
Окна для маркеров	Определяется список окон, которые могут быть использованы для отображения маркеров на карте.

Вкладка Сервер

Вид вкладки:



Свойство	Значение
Источник данных	Перетащите источник данных (параметр типа 'массив структур') для назначения
Тип карты	Глобальная с тайловым сервером
Адрес тайлового сервера	
Сабдомены тайлового сервера	
Центр карты	
Максимальное приближение	23
Минимальное приближение	0

Вкладка содержит таблицу, состоящую из двух столбцов, где в левом столбце отображается свойство, а в правом его значение.

Описание свойств:

Название	Описание
Источник данных	<p>Настраивается связь с источником данных для маркеров, которые будут располагаться на карте. В данное поле можно перетащить параметр типа массив структур, который будет содержать в себе все необходимые данные для отображения информации на маркерах карты. Порядок формирования типа данных и использование смотрите в разделе Источник данных для маркеров Карты</p>
Тип карты	<p>Задается способ получения тайлов карты. Возможные следующие варианты:</p> <p>Глобальная с тайловым сервером - данный тип карты создается с помощью подключения к внешнему источнику тайлов - тайловому серверу. Путь к тайловому серверу задается в параметре Адрес тайлового сервера</p> <p>Глобальная с ресурсами тайлов - в данном типе карты используются подготовленные локальные ресурсы тайлов. Но геолокация и позиционирование маркеров осуществляется в глобальных координатах (долгота\широта). Это может быть полезно для серверов без доступа к внешним интернет ресурсам. Для этого необходимо "нарезать" карту в хорошем разрешении на тайлы. Для генерации тайлов рекомендуется использовать утилиту от Яндекса (Подробнее о процессе генерации). Сгенерированную папку необходимо разместить в папке проекта Res, а в параметре Адрес тайлового сервера записать имя этой папки.</p> <p>Локальная -использует локальный ресурс, картинку, в качестве подложки карты с возможностью изменения масштаба и размещением на ней элементов. В текущей версии не поддерживается.</p>
Адрес тайлового сервера	<p>Если используется тип карты Глобальная с тайловым сервером, то указывается шаблон для формирования URL тайлового сервера. Например, для Stutnik - <code>Http://tilesspunik.ru/{z}/{x}/{y}.png</code>, OpenStreetMap указывается <code>https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png</code>, а для Яндекс.Карты: <code>yandex</code> (другие варианты описаны в разделе - Яндекс.Карты)</p>

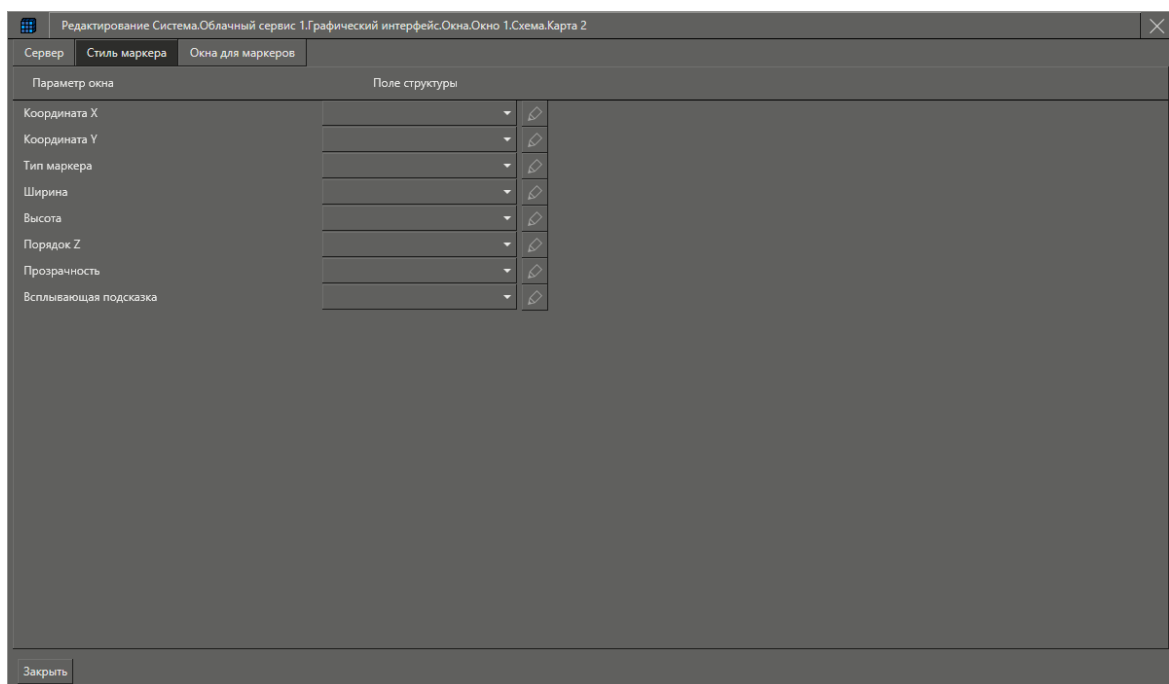
Название	Описание
	<p>где {z}, {x}, {y} - это zoom, долгота и широта, а {s} - это маска для сабдомена. Сабдомены - это некие символы для формирования разного url для резервных подключений.</p> <p>Если используется тип карты Глобальная с ресурсами тайлов, то в данном поле указывается имя папки, которая находится в папке <i>Res</i> разрабатываемого проекта, которая содержит в себе сохраненные тайлы.</p>
Сабдомены тайлового сервера	<p>Указываются сабдомены допустимые для указанного тайлового сервера. Например, для OpenStreetMap подставляются символы из строки a,b,c.</p>
Центр карты	<p>Указывается точка первоначальной загрузки карты, задается строкой в формате : y,x,z - где</p> <p>y - latitude - широта (значение 0 - экватор)</p> <p>x - longitude - долгота (значение 0 - нулевой меридиан)</p> <p>z - zoom, приближение</p> <p>Может использоваться как для настройки первоначальной точки загрузки карты, так и для динамического центрирования нужной точки карты, например при наступлении некоторого события, по триггеру и тп.</p>
Максимальное/ Минимальное приближение	<p>Определяется значение, на котором будет зафиксирован максимальный/минимальный zoom карты, нужно для ограничения зоны просмотра. Например при пользовательских тайлах невозможно будет просматривать фрагменты в неподготовленных участках карты . Так ограничив область zoom (а в дальнейшем планируется и ограничение по координатам), пользователь не будет выходить за рамки существующих тайлов. Для Яндекс.Карты это целое число от 0 до 23 - стандартные уровни zoom. Для других тайловых серверов они могут отличаться.</p> <p>Эти настройки нужны для ограничения зоны просмотра. Например при пользовательских тайлах невозможно будет просматривать фрагменты в неподготовленных участках карты . Так ограничив область zoom (а в дальнейшем планируется и ограничение по</p>

Название	Описание
	координатам), пользователь не будет выходить за рамки существующих тайлов.


На данной вкладке представлены основные свойства, необходимые для работы с Картой. Дополнительные свойства могут быть настроены в панели свойств.

Вкладка Стиль маркера


Вид вкладки по умолчанию:



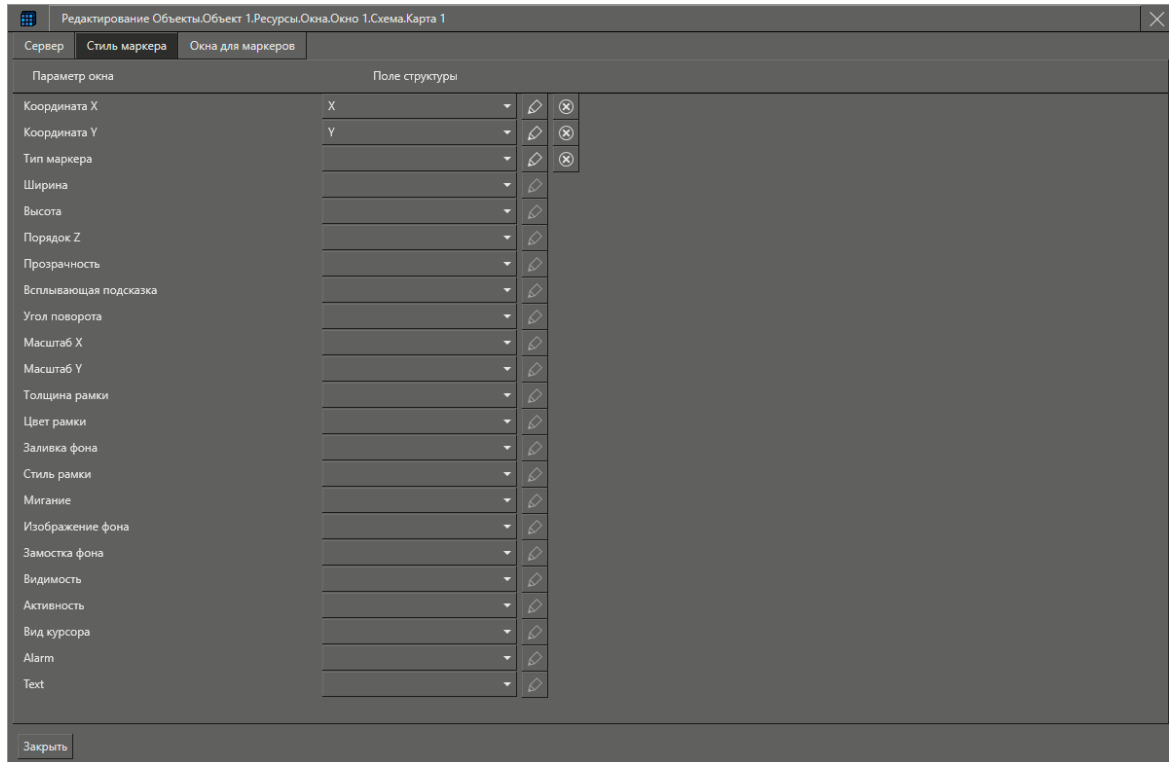
Важно! Вкладку нужно настраивать после того, как была установлена связь между картой и параметром нужного типа массив структур (см. Источник данных для маркеров Карты), а также после настройки вкладки Окна для маркеров

По умолчанию вкладка содержит основные параметры, которые не зависят от окна, которое может быть использовано для отображения маркеров. В выпадающем списке напротив нужного параметра будут отображаться поля структуры источника данных. Нужно выбрать нужное поле и настроить конвертацию нажав на на кнопку 

Название	Назначение
Координата X	Определяет положение маркера на карте. Нужно выбрать поле, которое отвечает за долготу. Параметр должен быть задан обязательно

Название	Назначение
Координата Y	Определяет положение маркера на карте. Нужно выбрать поле, которое отвечает за широту. Параметр должен быть задан обязательно
Тип маркера	Определяет, какое окно из тех, что были добавлены на вкладку Окно для маркеров будет использовано для отображения маркера. Если свойство не будет связано с полем структуры, то в качестве окна маркера будет использовано предопределенное обозначение, 
Ширина	Определяет ширину окна маркера.
Высота	Определяет высоту окна маркера.
Порядок Z	Определяет z-порядок маркера.
Прозрачность	Определяет прозрачность. Значение изменяется в пределах от 0 до 100, где 0 - элемент полностью прозрачный, 100 - элемент полностью не прозрачный
Всплывающая подсказка	Определяет всплывающую подсказку, которая появляется при наведении курсора на маркер

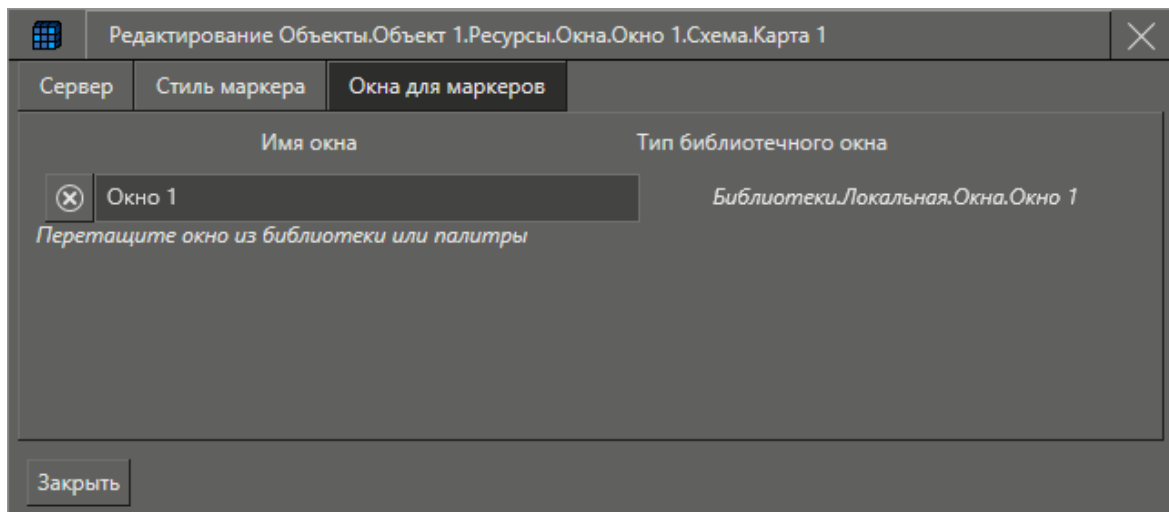
После добавления окон на вкладке Окна для маркеров, кроме стандартных параметров вкладка может содержать еще параметры этих окон (стандартные и добавленные пользователем)



Между параметрами и значениями полей структуры также можно настроить связь и зависимость.

Вкладка Окна для маркеров

Вид вкладки:



В пользовательской библиотеке можно создать окна, которые можно перетащить на данную вкладку и в дальнейшем использовать для отображения маркеров на карте: использовать их в качестве опорных точек при настройке конвертации параметра маркера Тип маркера на вкладке Стиль маркера.

8.3.2.10.2.СВОЙСТВА КАРТЫ

Вид панели свойств Карты:

Карта 1


Общие

Служебные



Параметры

Тип карты	Глобальная с тайловым сервером
Адрес тайлового сервера	http:// vec(s).maps.yandex.net/tiles?l=r
Сабдомены тайлового сервера	01,02,03
Центр карты	56.435193, 38.252585, 14
Максимальное приближение	23
Минимальное приближение	12
Система координат	Эллиптическая проекция Меркатора
Атрибуция	
Приближение по двойному клику	<input checked="" type="checkbox"/>

Данные

 + Источник данных

Внешний вид

Заливка фона	 rgb(245,240,245)
Прозрачность	100
Мигание	<input type="checkbox"/>
Размер тени	0
Цвет тени	 rgb(240,240,240)
Изображение фона	
Замостка фона	Заполнение
Видимость	<input checked="" type="checkbox"/>
Активность	<input checked="" type="checkbox"/>
Сохранять состояние	По умолчанию
Вид курсора	Стандартный

Расположение

Рамка

Отношения

Трансформация

Клавиатура

Экран

Маркер

Название	Назначение
Категория Параметры	
Тип карты	<p>Задается способ получения тайлов карты. Возможные следующие варианты:</p> <p>Глобальная с тайловым сервером - данный тип карты создается с помощью подключения к внешнему источнику тайлов - тайловому серверу. Путь к тайловому серверу задается в параметре Адрес тайлового сервера</p> <p>Глобальная с ресурсами тайлов - в данном типе карты используются подготовленные локальные ресурсы тайлов. Но геолокация и позиционирование маркеров осуществляется в глобальных координатах (долгота\широта). Это может быть полезно для серверов без доступа к внешним интернет ресурсам. Для этого необходимо "нарезать" карту в хорошем разрешении на тайлы. Для генерации тайлов рекомендуется использовать утилиту от Яндекса (Подробнее о процессе генерации). Сгенерированную папку необходимо разместить в папке проекта Res, а в параметре Адрес тайлового сервера записать имя этой папки.</p> <p>Локальная -использует локальный ресурс, картинку, в качестве подложки карты с возможностью изменения масштаба и размещением на ней элементов. В текущей версии не поддерживается.</p>
Адрес тайлового сервера	<p>Если используется тип карты Глобальная с тайловым сервером, то указывается шаблон для формирования URL тайлового сервера. Например, для Stutnik - <code>Http://tile Sputnik.ru/{z}/{x}/{y}.png</code>, OpenStreetMap указывается <code>https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png</code>, а для Яндекс.Карты: <code>yandex</code> (другие варианты описаны в разделе - Яндекс.Карты)</p> <p>где {z}, {x}, {y} - это zoom, долгота и широта, а {s} - это маска для сабдомена. Сабдомены - это некие символы для формирования разного url для резервных подключений.</p> <p>Если используется тип карты Глобальная с ресурсами тайлов, то в данном поле указывается имя папки, которая находится в папке Res разрабатываемого проекта, которая содержит в себе сохраненные тайлы.</p>

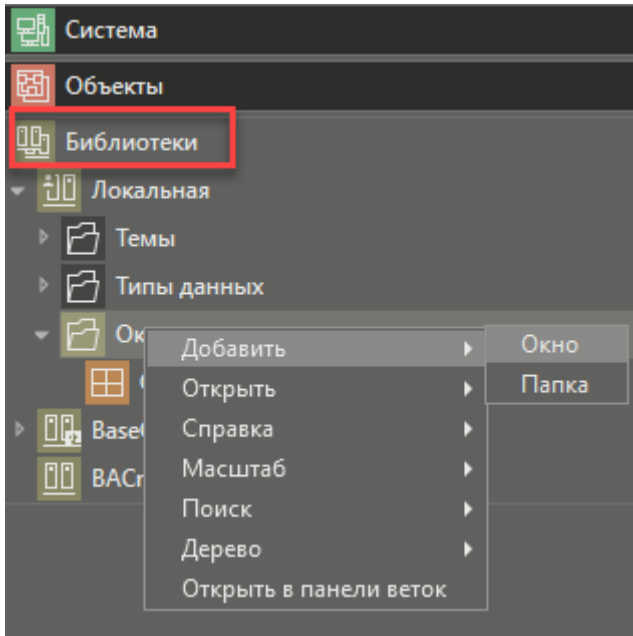
Название	Назначение
Сабдомены тайлового сервера	<p>Указываются сабдомены допустимые для указанного тайлового сервера. Например, для Яндекс.Карты массив сабдоменов выглядит как 01,02,03,04, для OpenStreetMap подставляются символы из строки a,b,c.</p>
Центр карты	<p>Указывается точка первоначальной загрузки карты, задается строкой в формате : y,x,z - где</p> <p>y - latitude - широта (значение 0 - экватор)</p> <p>x - longitude - долгота (значение 0 - нулевой меридиан)</p> <p>z - zoom, приближение</p> <p>Может использоваться как для настройки первоначальной точки загрузки карты, так и для динамического центрирования нужной точки карты, например при наступлении некоторого события, по триггеру и тп.</p>
Максимальное/ Минимальное приближение	<p>Определяется значение, на котором будет зафиксирован максимальный/минимальный zoom карты, нужно для ограничения зоны просмотра. Например при пользовательских тайлах невозможно будет просматривать фрагменты в неподготовленных участках карты . Так ограничив область zoom (а в дальнейшем планируется и ограничение по координатам), пользователь не будет выходить за рамки существующих тайлов. Для Яндекс.Карты это целое число от 0 до 23 - стандартные уровни zoom. Для других тайловых серверов они могут отличаться.</p> <p>Эти настройки нужны для ограничения зоны просмотра. Например при пользовательских тайлах невозможно будет просматривать фрагменты в неподготовленных участках карты . Так ограничив область zoom (а в дальнейшем планируется и ограничение по координатам), пользователь не будет выходить за рамки существующих тайлов.</p>
Система координат	<p>Реализует механику преобразования координат и их расположение в контейнере карты. Возможные значения:</p> <ul style="list-style-type: none"> • Эллиптическая проекция Меркатора - используется для Яндекс-тайлов

Название	Назначение
	<ul style="list-style-type: none"> • Сферическая проекция Меркатора - используется в Openstreetmap тайлах • Равнопрямоугольная проекция • Простая - используется при настройке типа карты "Локальная" - значения широты-долготы напрямую используются как X,Y, точка (0,0) переносится в левый нижний угол
Атрибуция	<p>Определяется текст в нижнем правом углу. Может использоваться для некой подписи, указания прав и лицензий или ссылки на ресурс тайлов, например карты Sputnik можно использовать только с указанием источника.</p> <p>Пример вставки гиперссылки в атрибуции: <code>MasterSCADA</code></p>
Приближение по двойному клику	Позволяет изменять масштаб при помощи двойного клика по карте в клиенте визуализации. По умолчанию флаг установлен.
Категория Данные	
Источник данных	<p>Настраивается связь с источником данных для маркеров, которые будут располагаться на карте. В данное поле можно перетащить параметр типа массив структур, который будет содержать в себе все необходимые данные для отображения информации на маркерах карты. Порядок формирования типа данных и использование смотрите в разделе Источник данных для маркеров Карты</p>

8.3.2.10.3. ОКНА ДЛЯ ОТОБРАЖЕНИЯ МАРКЕРОВ КАРТЫ

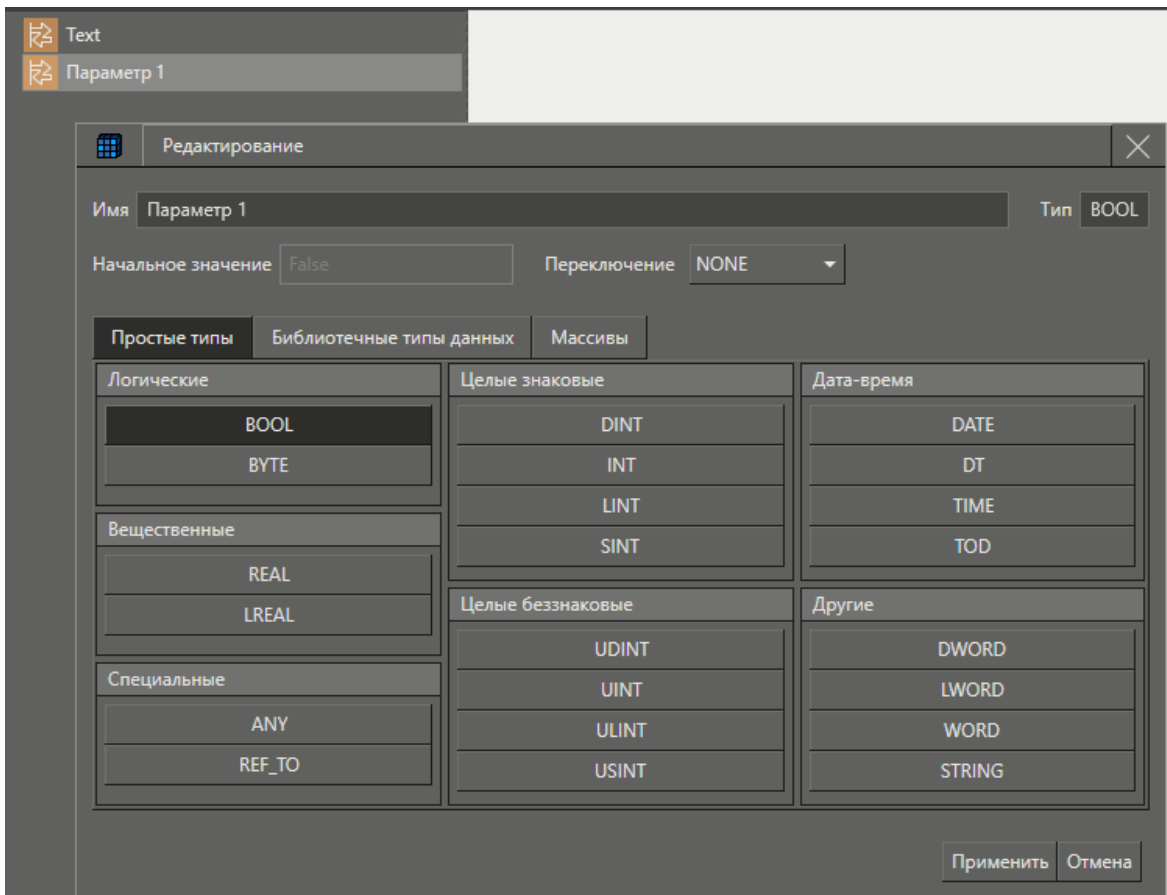
Создание окна

Окна, которые могут быть использованы для отображения маркеров нужно создавать в пользовательской библиотеке:

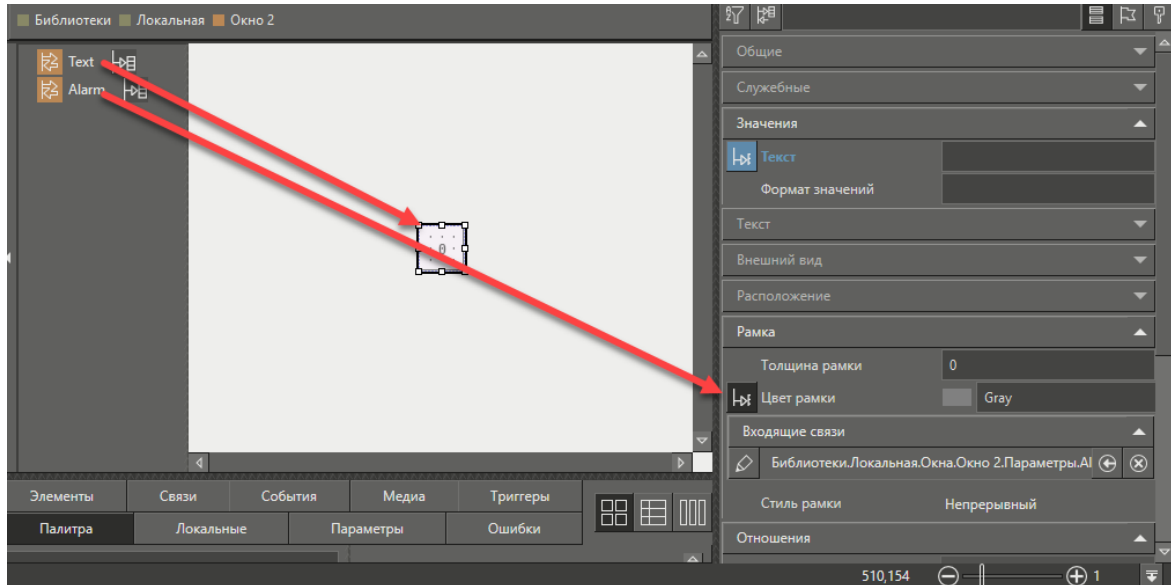


Размер окна изменяют, т.к. маркеры карты должны быть небольшого размера.

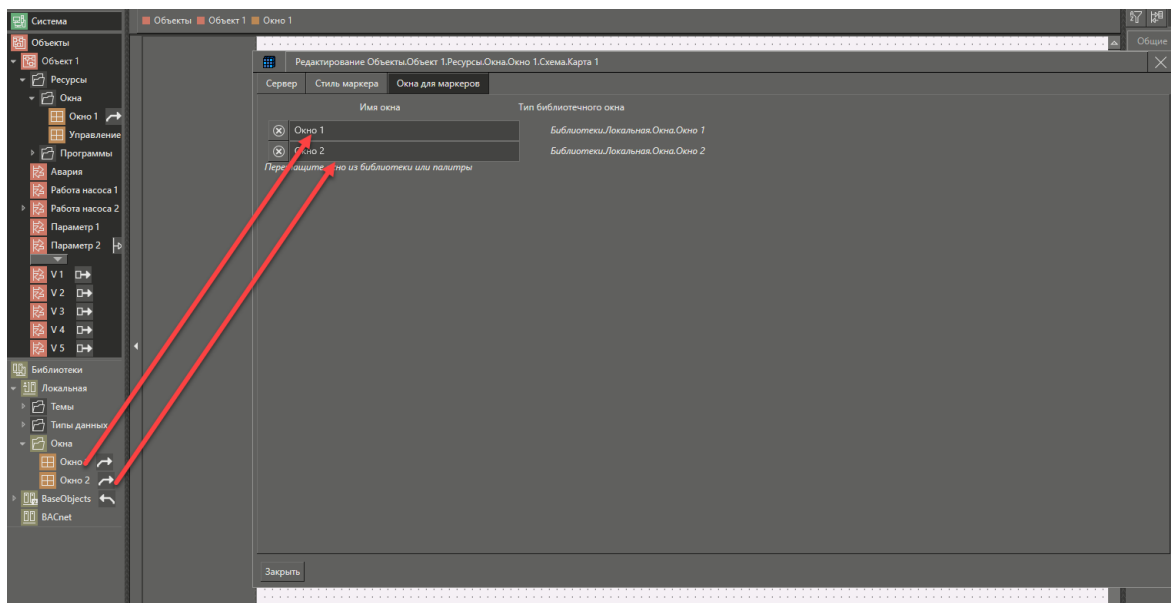
Далее нужно создать параметры окна, которые будут влиять на изменение внешнего вида маркера в режиме исполнения. Например:



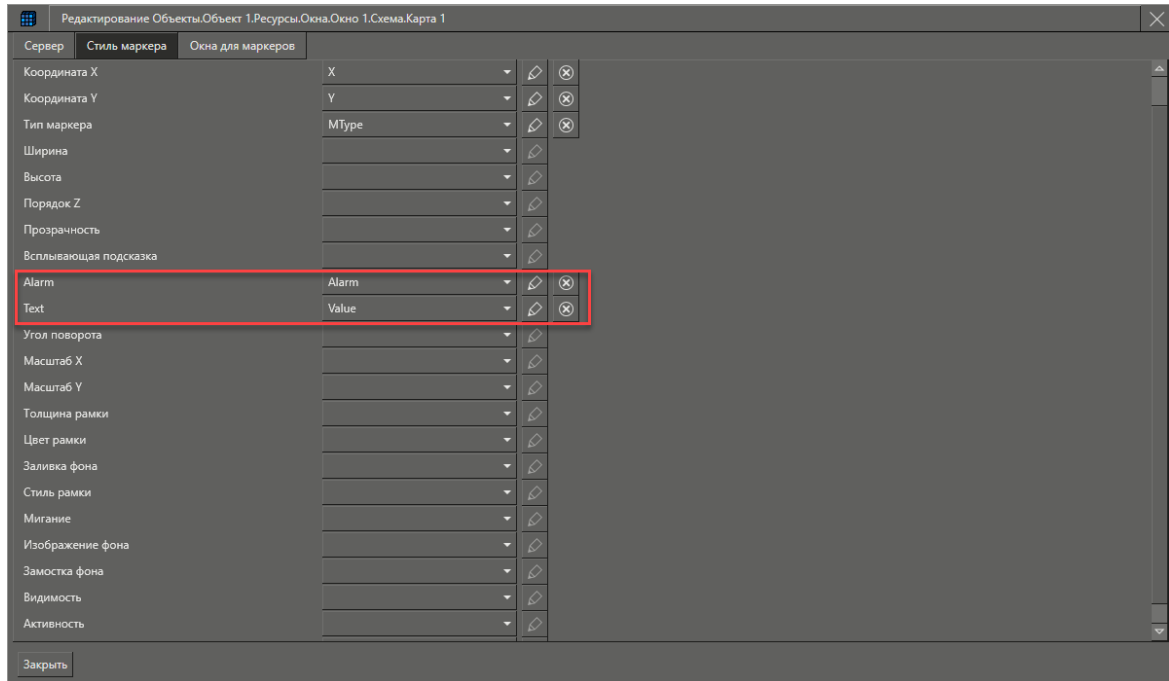
Создаем внешний вид будущего маркера. Например, можно использовать параметр Text для отображения важного параметра, а параметр Alarm, для изменения цвета и размера рамки:



После того как окно в библиотеке будет создано, можно будет перетащить его в редактор карты во вкладку Окна для маркеров:



После этого на вкладке маркеров Стиль маркера появятся дополнительные параметры окна, созданные вручную. Далее нужно выбрать поле структуры, от которого будет зависеть параметр Тип маркера и настроить конвертацию, и также связать дополнительные параметры окна с нужными полями структуры:



В клиенте визуализации получим:



В зависимости от значений, которые будут получены источником данных будут меняться значения, которые будут отображаться. в окне маркера и будет отображаться рамка.

Если необходимо, чтобы при нажатии на маркер открывалось другое окно, то нужно настроить действие карты Клик по маркеру.

8.3.2.10.4. ИСТОЧНИК ДАННЫХ ДЛЯ МАРКЕРОВ КАРТЫ

Источник данных карты - это массив структур, в котором каждая структура описывает один маркер на карте. В этой структуре обязательно должны быть поля X, Y, в которых записаны

координаты элемента, а также поле для отображения типа маркера. Помимо этих обязательных полей могут быть поля, описывающие любые другие свойства маркера – видимость, фон, пользовательские параметры. Источником данных может быть параметр узла, объекта или программы. Чтобы привязать этот параметр к карте, надо его перетащить в поле Источник данных в окне редактора карты на вкладке Сервер или в панели свойств.

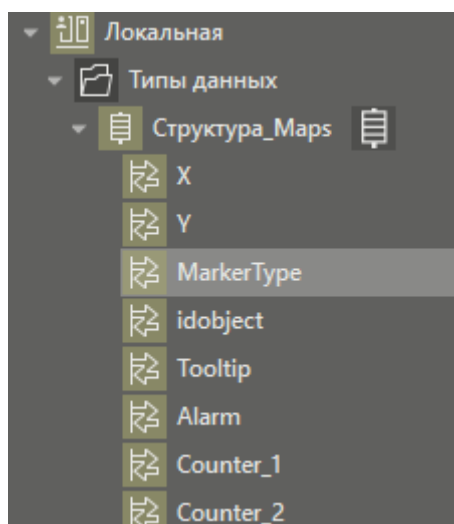
Рассмотрим основные этапы создания параметра нужного типа.

Создание структуры

В пользовательской библиотеке нужно создать структуру с нужными полями. Три поля должны быть обязательными:

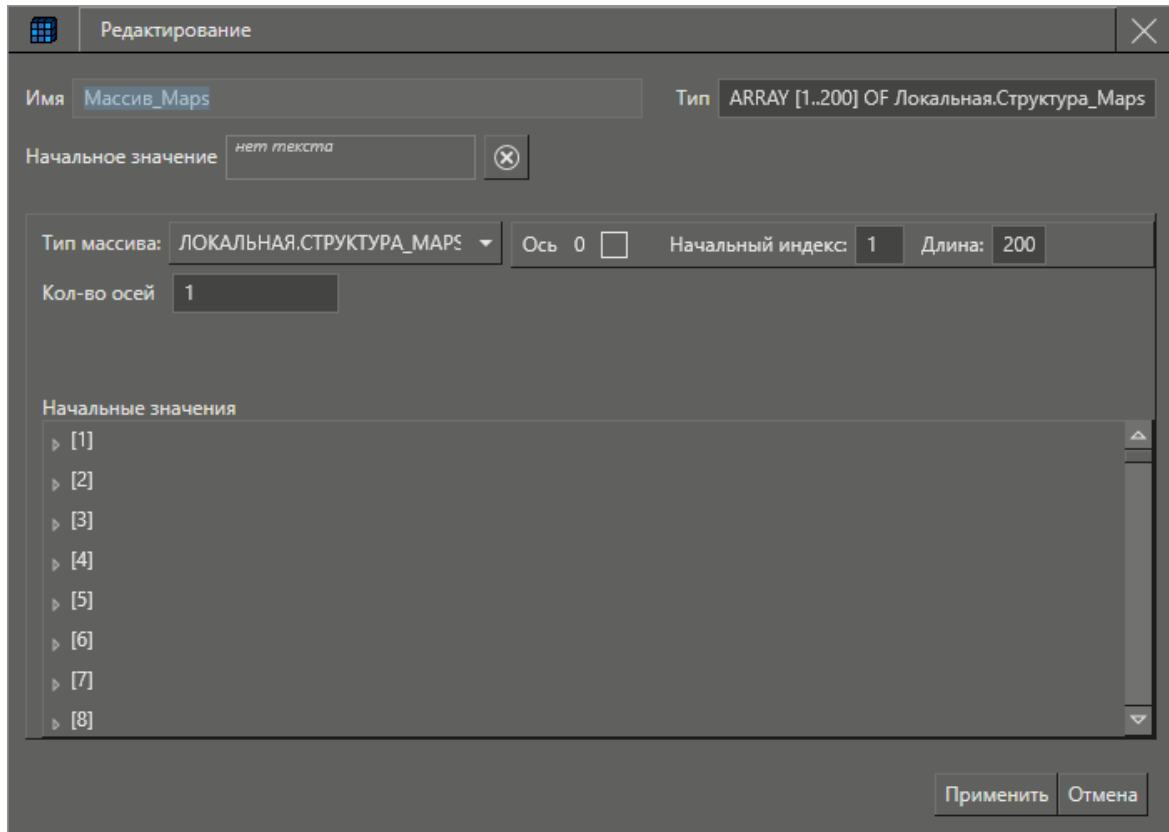
Название	Тип	Назначение
X	LREAL	Определяет долготу, на которой будет находиться маркер. Как правило, задается в градусах.
Y	LREAL	Определяет широту маркера, на которой будет находиться маркер. Как правило, задается в градусах
MarkerType	например , INT	Поле, которое будет определять, как будет отображаться маркер, какое окно, будет использовано, для отображения маркера. Если нужно на карте размещать три разных типа объектов, то параметр должен принимать три предопределенных значения, например, 1, 2 или 3.

Помимо обязательных параметров, могут быть добавлены и произвольные, которые также могут влиять на внешний вид маркеров (или отображаться в всплывающих окнах, при клике на маркер) например:



Создание массива структур

Далее в пользовательской библиотеке нужно создать массив структур, где в качестве элемента массива используется ранее созданная структура. Количество элементов массива определяется количеством маркеров, которые нужно отображать на карте:

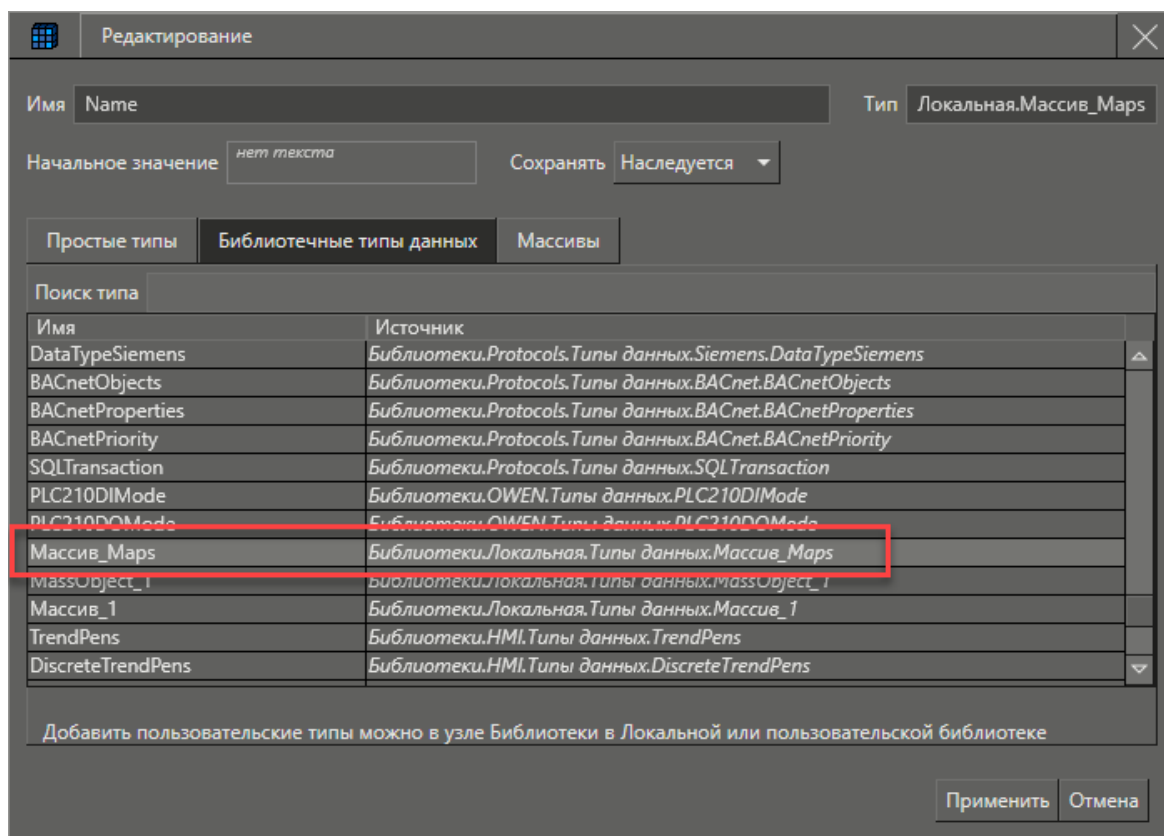


Массив может быть определенной на этапе разработки длины, либо изменяемой в режиме исполнения, т.е. динамической.

Если используется динамический массив, то в программе, который будет его формировать нужно использовать специальные функции, предназначенные для работы с такими массивами.

Параметр типа массив структур

При создании параметра нужно назначить ему созданный тип массив структур:

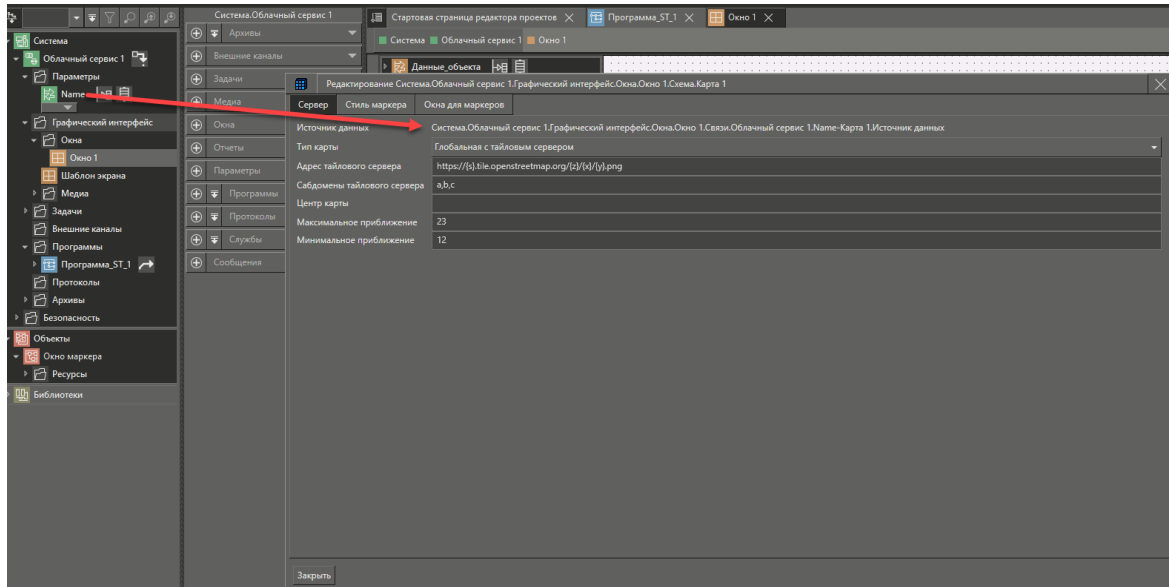


Чтобы значения полей структур массива изменялись в режиме исполнения нужно создать программу, где они будут связаны или с параметрами проекта, либо будут прочитаны из файла или из базы данных. Текст программы зависит от задач, которые решаются в проекте. Обычно для создания таких программ используют редактор ST.

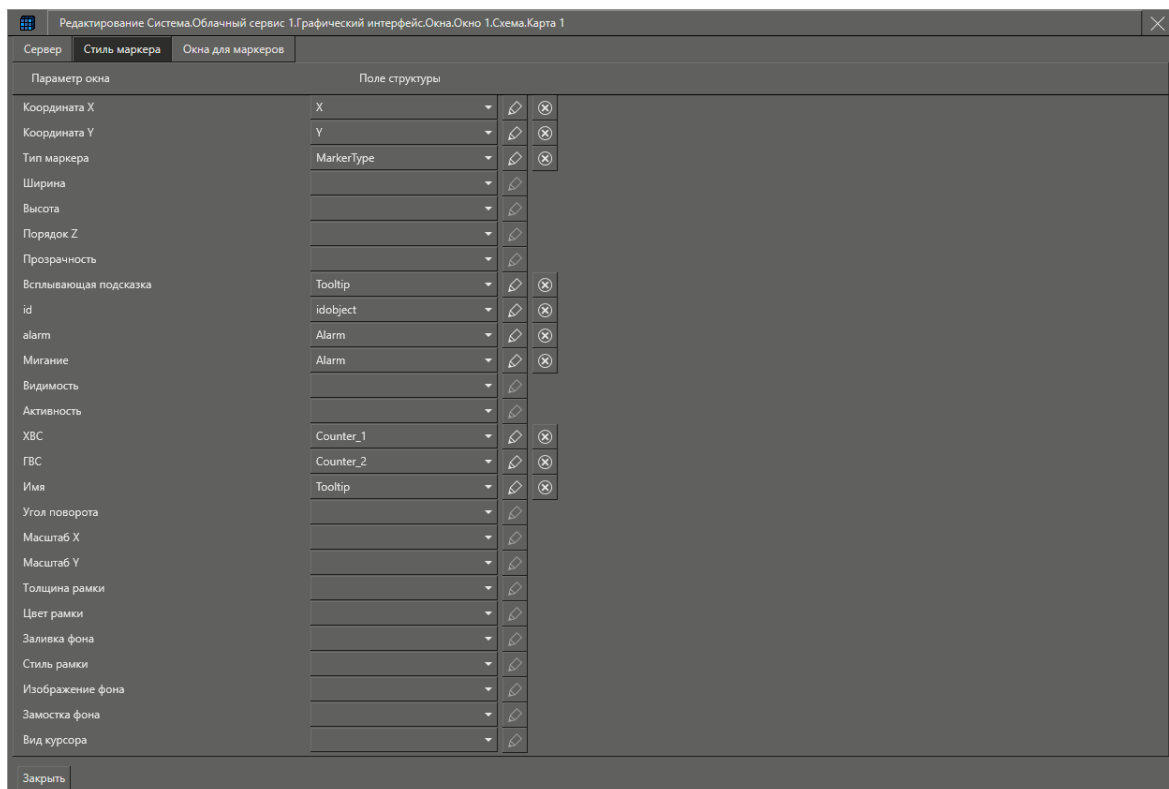
Чтобы обратиться к какому-либо полю массива структур можно использовать запись: Параметр[i]. поле, где Параметр - имя параметра, i - номер элемента массива, а поле, это имя поля структуры.

Связь между внешним видом маркера и полями параметра

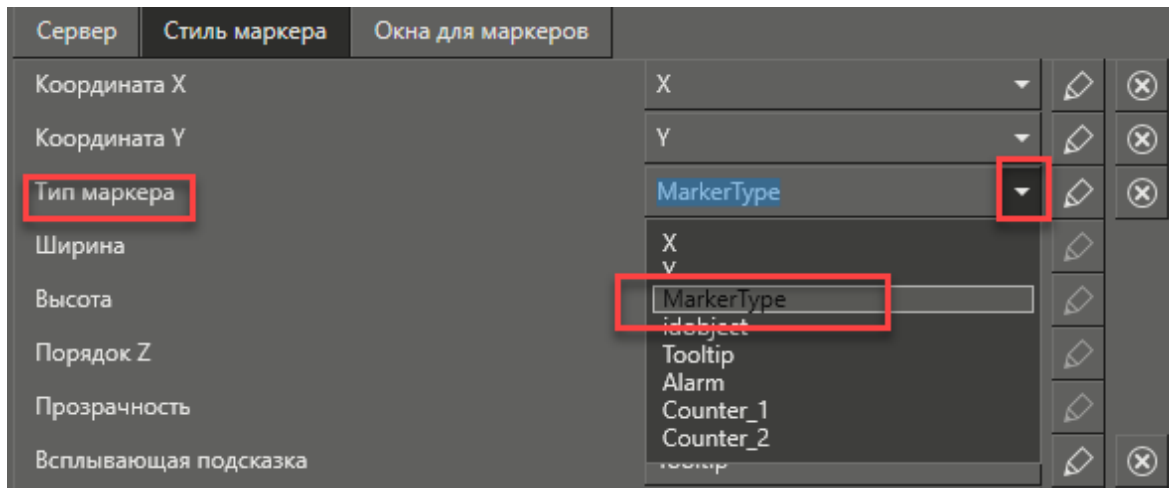
После того как параметр нужного типа создан, его нужно перетащить в поле Источник данных в окне редактора карты на вкладке Сервер или в панели свойств:




После этого на вкладке Стиль маркера редактора Карты появится возможность настроить зависимость внешнего вида маркера и полями источника данных:

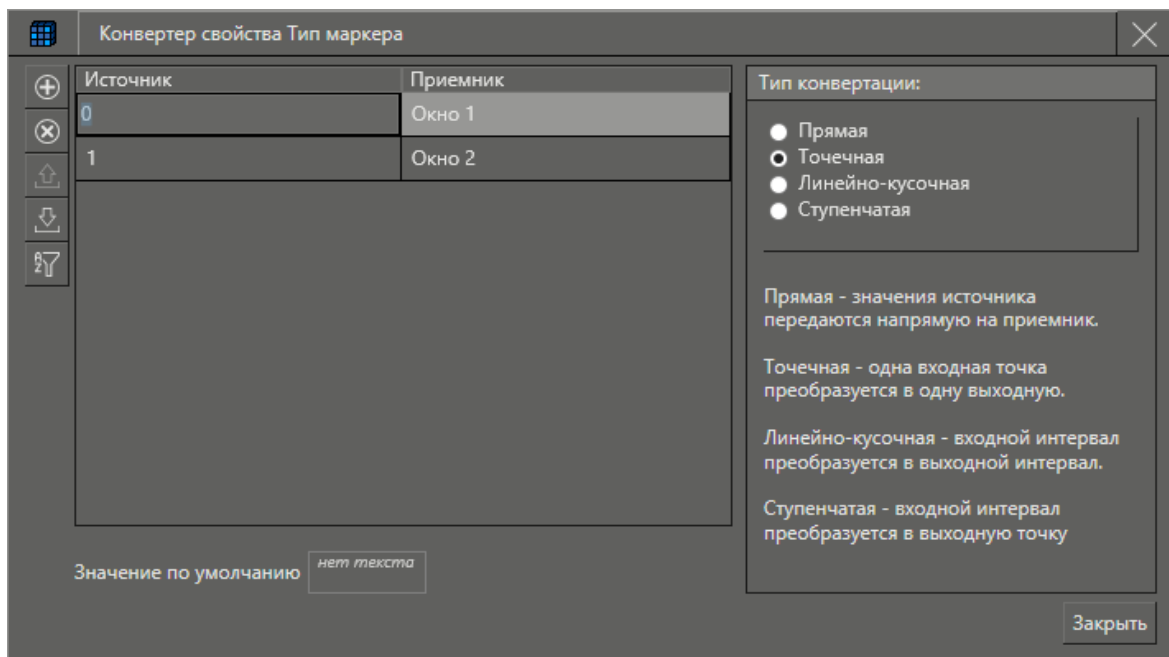


Рассмотрим порядок настройки на примере параметра Тип маркера. В выпадающем списке, который соответствует данному параметру выбираем поле связанной структуры MarkerType:



Далее необходимо настроить зависимость между значением поля структуры и значением свойства, для этого нужно нажать на кнопку  рядом с полем выбора. Появится окно настройки конвертации.

Далее нужно определить какое окно, из тех, что добавлены во вкладке Окна для маркеров будет использовано для каждого из возможных значений поля структуры. В данном случае рекомендуется использовать точечную конвертацию, для других параметров могут быть использованы другие способы конвертации, в зависимости от поставленных задач:



8.3.2.10.5. ВСПЛЫВАЮЩЕЕ ОКНО МАРКЕРА КАРТЫ

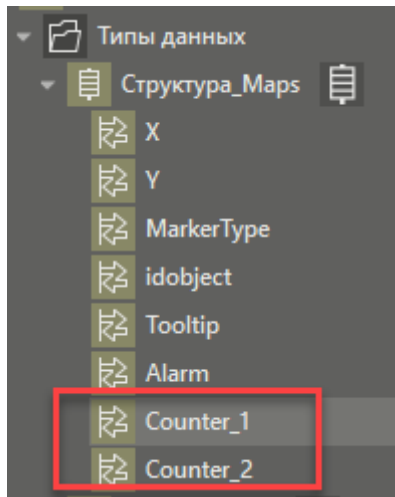
Если требуется выводить дополнительную информацию объекта из источника данных во всплывающее окно по клику на маркер или получить координаты клика, то нужно воспользоваться уникальными событиями карты.

Существует два специальных события:

- Событие Клик по карте
- Событие Клик по маркеру

Рассмотрим на примере.

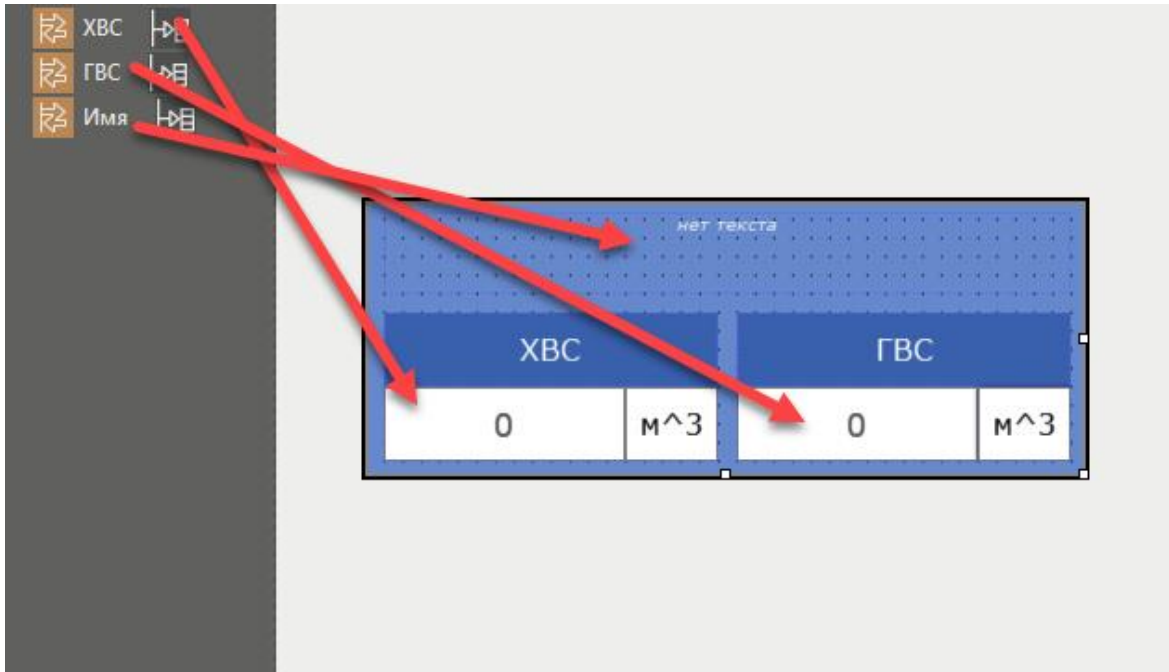
Пусть при клике на маркер, нужно открывать окно, в котором необходимо выводить значения двух параметров, которые относятся к конкретному маркеру. Для этого структура, которая является элементом массива типа параметра, связанного с источником данных, должна иметь эти два поля:



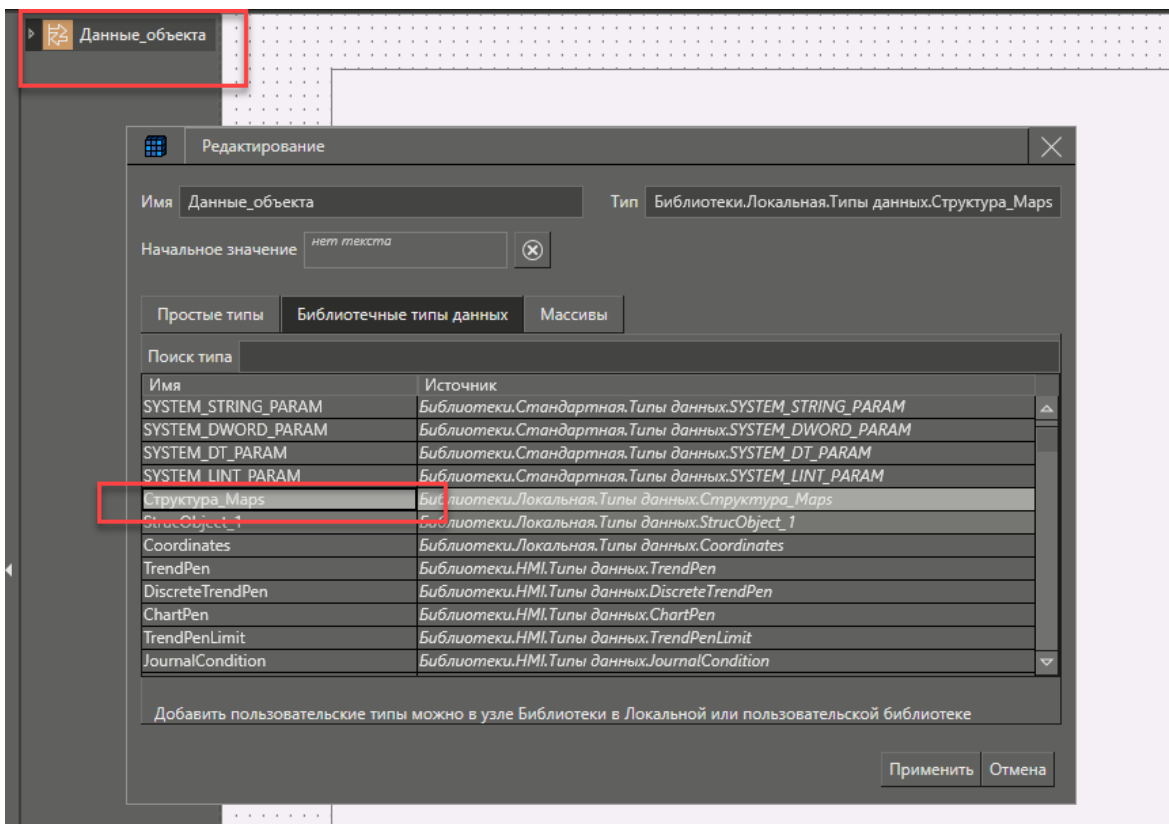
Пусть в нашем примере Counter_1 отвечает за значение счетчика ГВС, а Counter_2 - ХВС.

Далее создаем окно. Если подобное окно будет использоваться в проекте неоднократно, то нужно его создавать в пользовательской библиотеке, а затем добавлять в дерево проекта его экземпляр. Если окно будет уникальным и не потребуется для других проектов, то можно создавать его непосредственно в дереве системы или в дереве объектов.

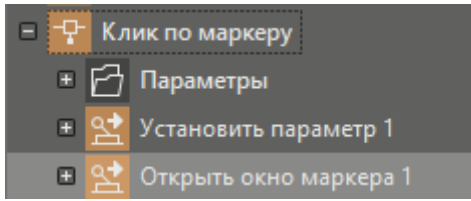
В окне создаем три параметра, и настраиваем внешний вид и связь между графическими элементами и параметрами окна:



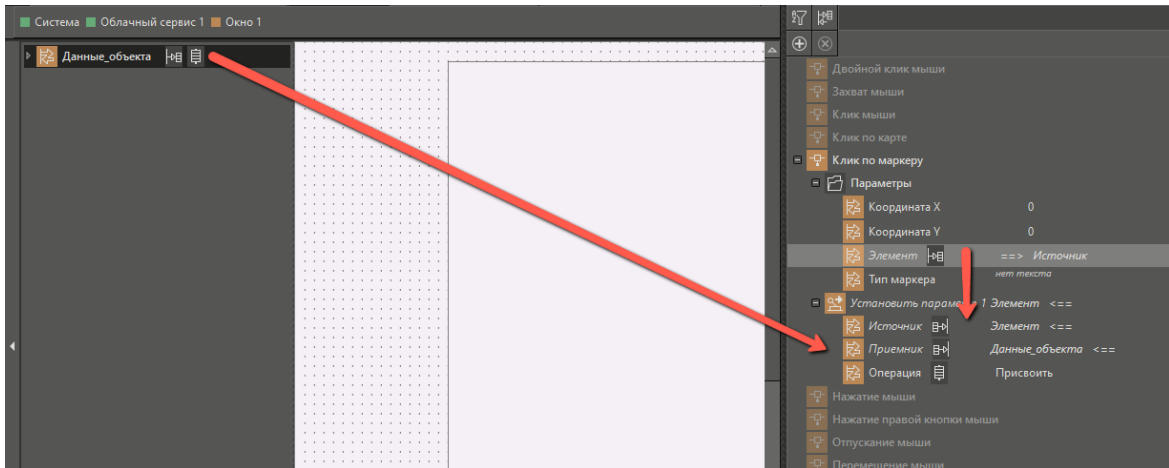
Далее нужно в окне, в котором содержится карта, добавить параметр типа структура. Такую структуру создавали для формирования источника данных для карты. Она легла в основу массива структур:



Далее выделяем элемент Карта и в панели свойств добавляем событие Клик по маркеру и в него два действия: Установить параметр и Открыть окно маркера:

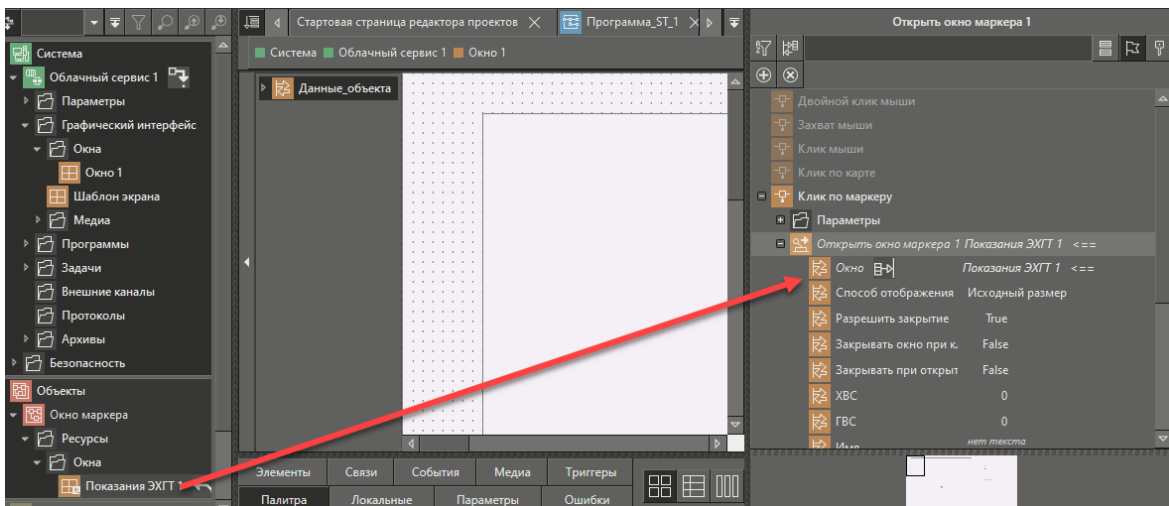


Настраиваем событие Установить параметр:

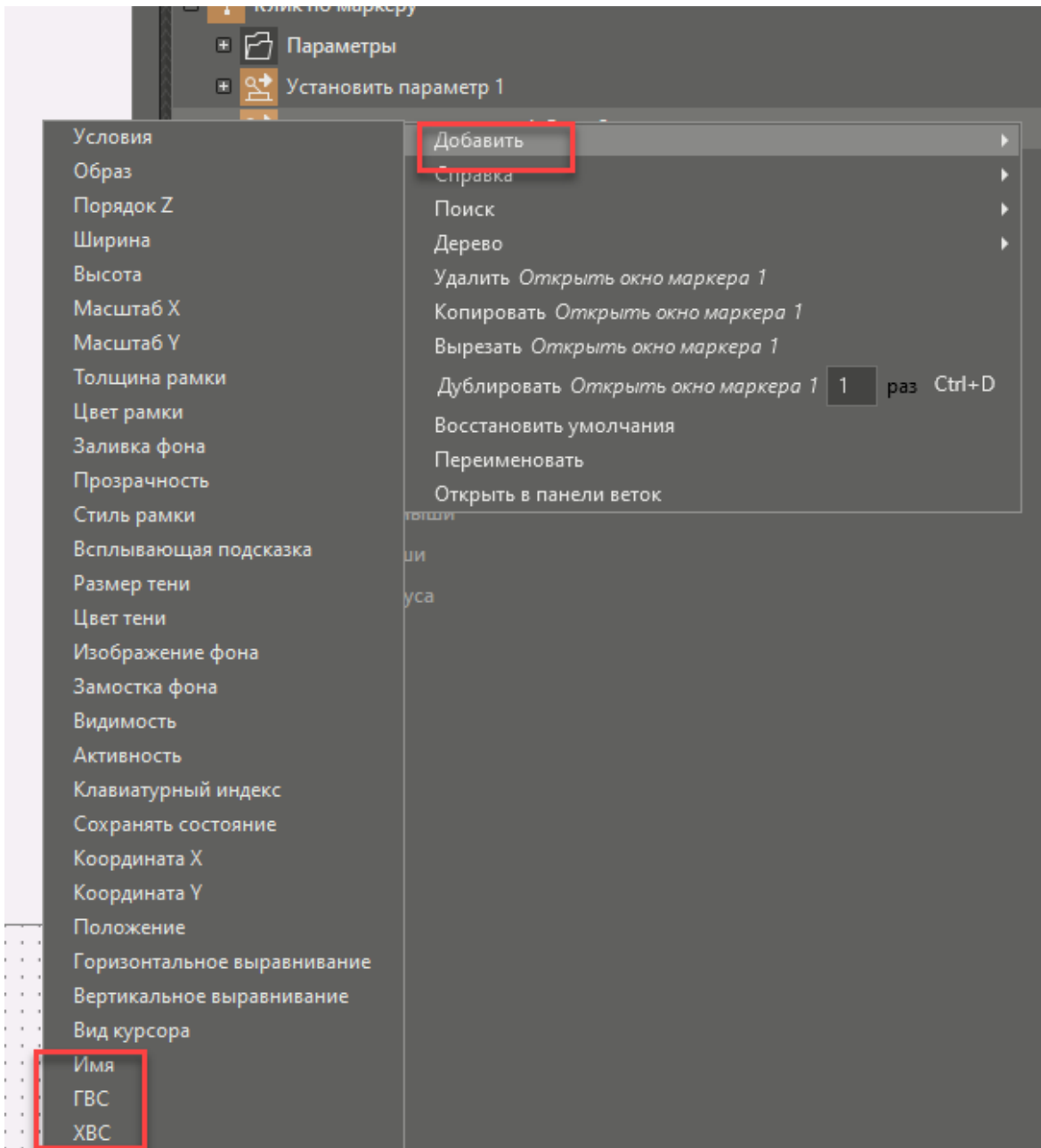


Затем настраиваем событие Открыть окно маркера:

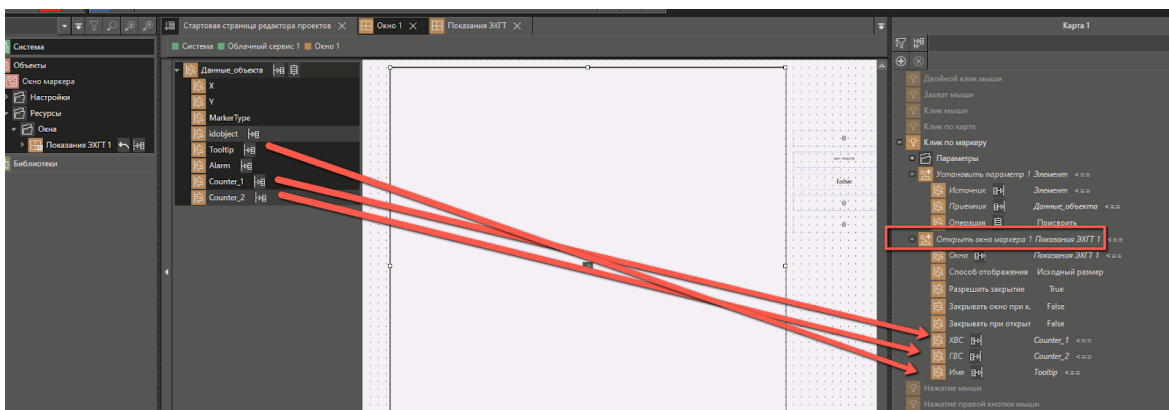
Перетаскиваем окно или экземпляр библиотечного окна из дерева объектов или из дерева системы, которое должно открываться при клике на маркер:



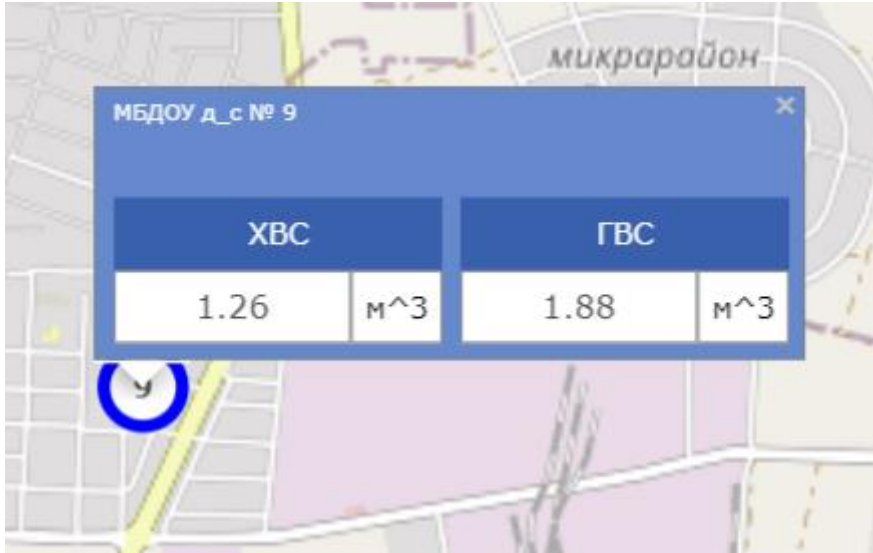
Затем добавляем параметры окна:



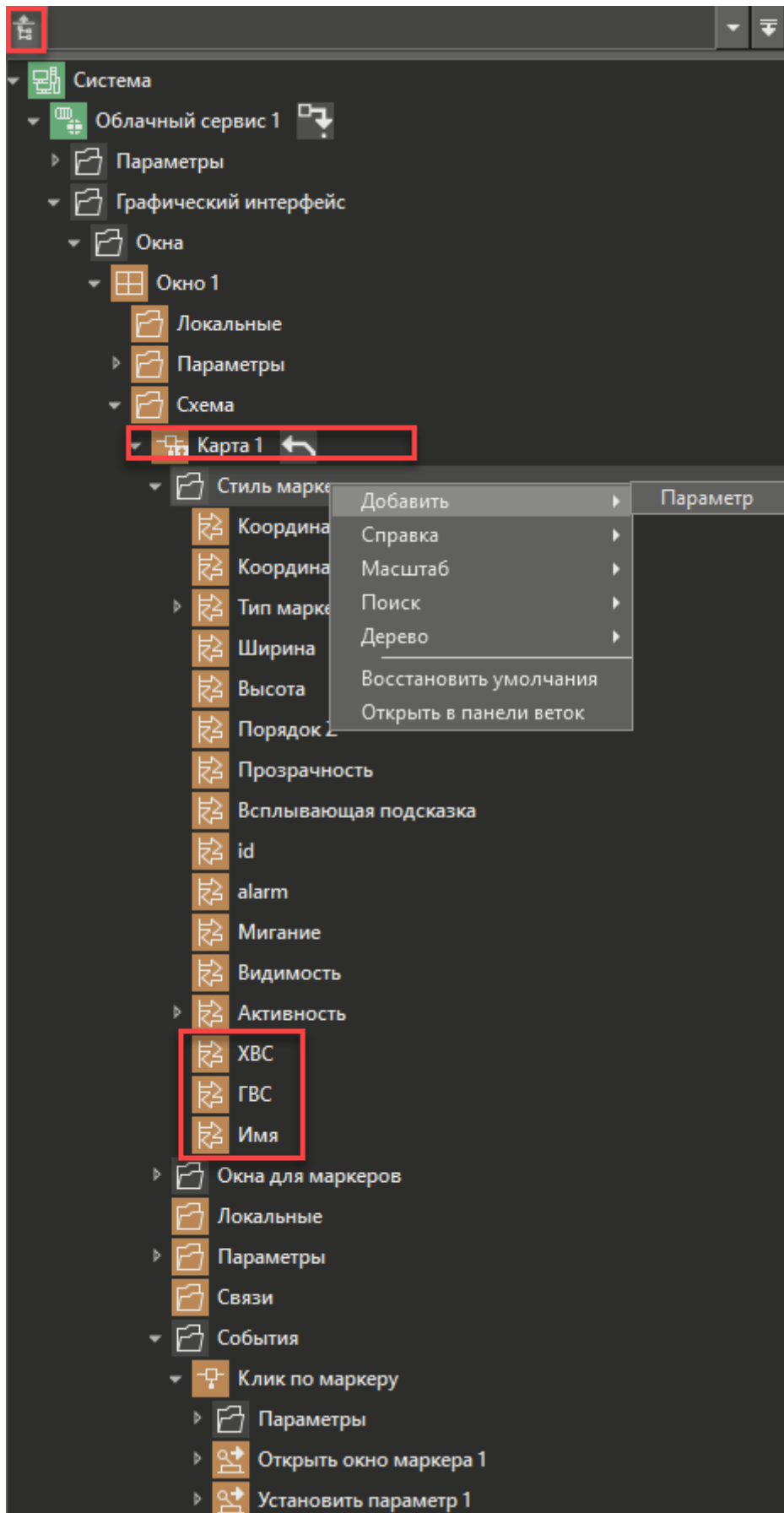
Далее устанавливаем связь между параметрами окна и полями структуры:



Т.е. сначала мы передаем данные от маркера в параметр окна типа Структура через действие Установить параметр, а затем из параметра этого окна во всплывающее окно. Таким образом получим, что в режиме исполнения если кликнуть по маркеру, то появится окно, в котором будут зафиксированы значения параметров в момент клика:

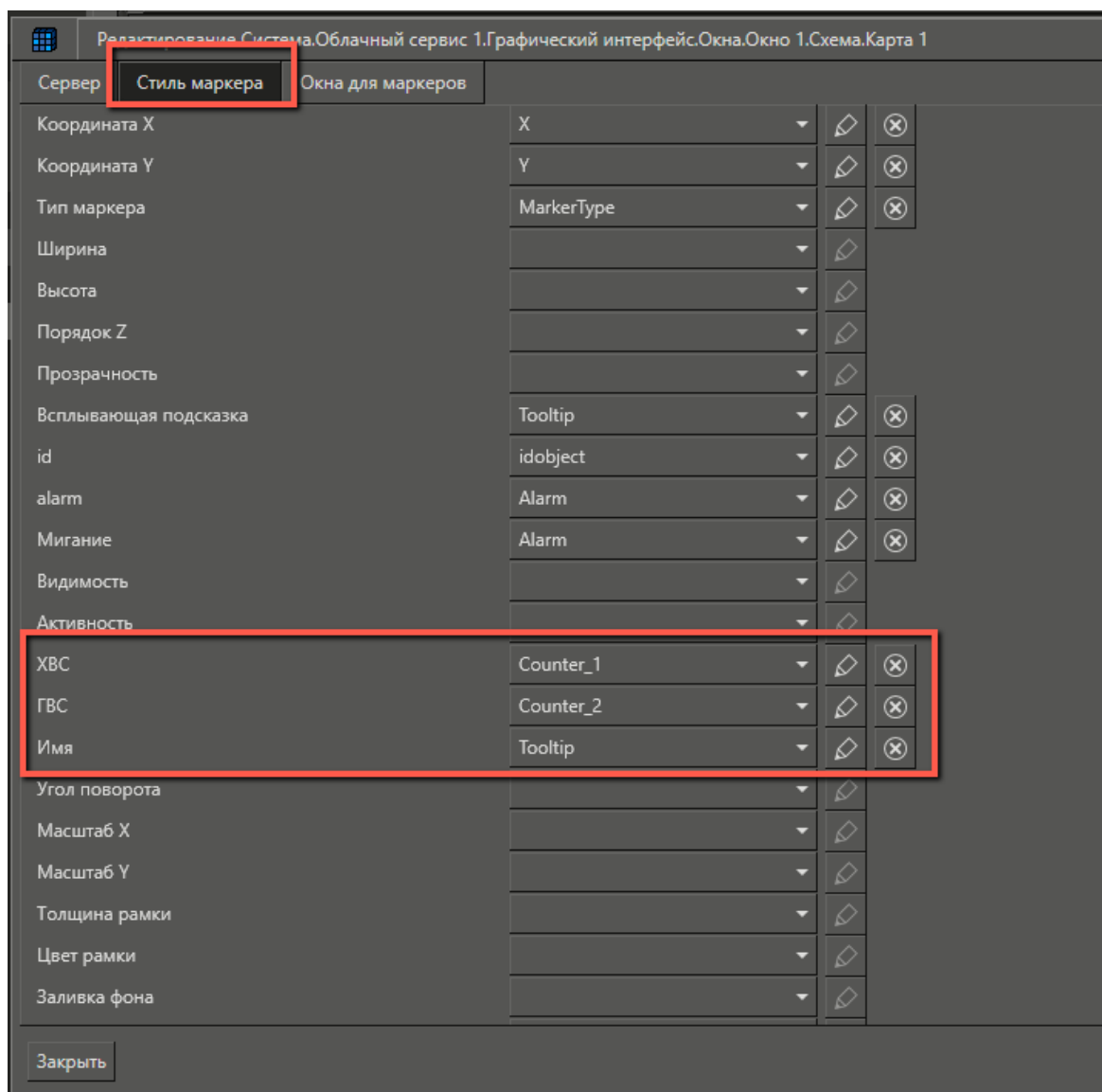


Если необходимо, чтобы данные изменялись в окне после клика, то необходимо включить отображение полного дерева проекта. Раскрыть в дереве содержимое элемента Карта, и в группу Стиль маркера добавить параметры, с такими же именами, как и параметры окна, которое должно открыться. В данном случае нужны параметры с именами: ГВС, ХВС и Имя:



Важно! Если имена параметров, созданных в группе Стиль маркера будут отличаться от имен параметров открываемого окна, то значение изменяться в открытом окне маркера в клиенте визуализации не будут.

Затем в редакторе карты на вкладке стиль маркера, нужно определить зависимость между параметрами и полями источника данных:



В результате, если окно маркера будет открыто, и если изменится поле структуры элемента массива источника данных, то это изменение отобразится в клиенте визуализации.

8.3.2.10.6. ЯНДЕКС.КАРТЫ

Подключение к сервисам Яндекса, согласно условиям использования, возможно только через API Яндекс карт. Поэтому настройка подключения имеет некоторые отличия от других серверов. Строку, задающую значение параметра Тайловый сервер необходимо составить по шаблону:

yandex#тип_карты#api_ключ, где:

Тип_карты - тип тайлов, используемых для подложки карты, возможные значения:

- map - классический векторный стиль тайлов с подписями, используется по умолчанию (можно не задавать, если не используется API-ключ)
- satellite - спутниковые снимки без подписей
- hybrid - спутниковые снимки с подписями, дорогами и др. объектами
- skeleton - только подписи

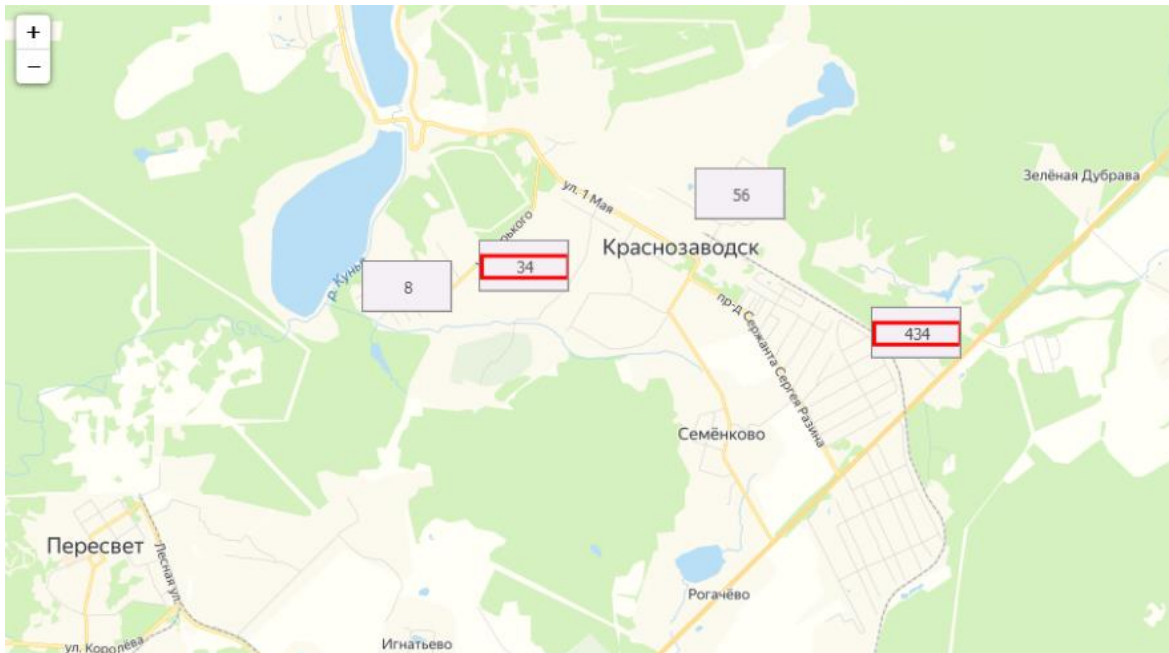
API_ключ - при наличии ключа, который необходим для легального коммерческого использования, если ключ не задан, то сервер может отказать в доступе, для проверки функций элемента, можно использоваться без ключа.

Примеры корректной настройки:

- yandex
- yandex#map#1234-1324-1243-1423
- yandex#hybrid

8.3.2.10.7.КАРТА В КЛИЕНТЕ ВИЗУАЛИЗАЦИИ

Вид карты в клиенте визуализации:



Положение маркеров будет зависеть от параметров X и Y, заданных в панели свойств или в редакторе карты.

Внешний вид маркеров определяется параметром Тип маркера.

Изменять масштаб карты можно используя колесико мыши. Также можно воспользоваться кнопками + и -, которые находятся в верхнем левом углу карты. Удерживая нажатой левую кнопку мыши можно изменять видимый участок на карте.

Смотреть также:

Карта

Редактор карты

Свойства Карты

Окна для отображения маркеров Карты

Источник данных для маркеров Карты

Всплывающее окно маркера карты

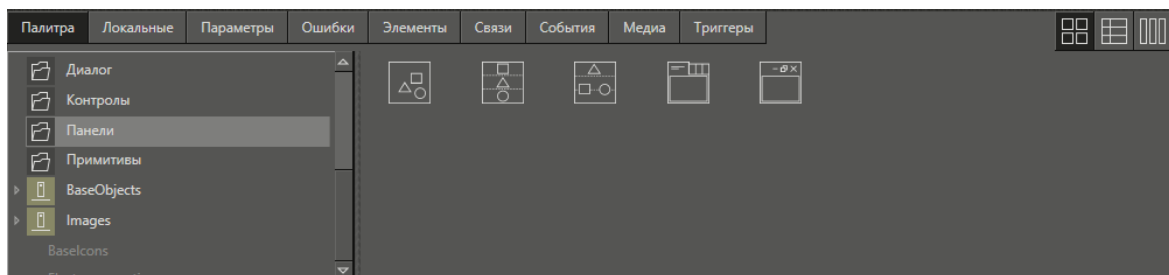
Яндекс.Карты

Карта в клиенте визуализации

8.3.3. КАТЕГОРИЯ ПАНЕЛИ

Категория Панели палитры редактора НМІ содержит элементы, предназначенные для группировки других элементов.

В палитре редактора НМІ эта категория имеет вид:



Категория Панели включает в себя следующие элементы:

Панель

Стековая панель

Панель док

Панель вкладок

Контейнер

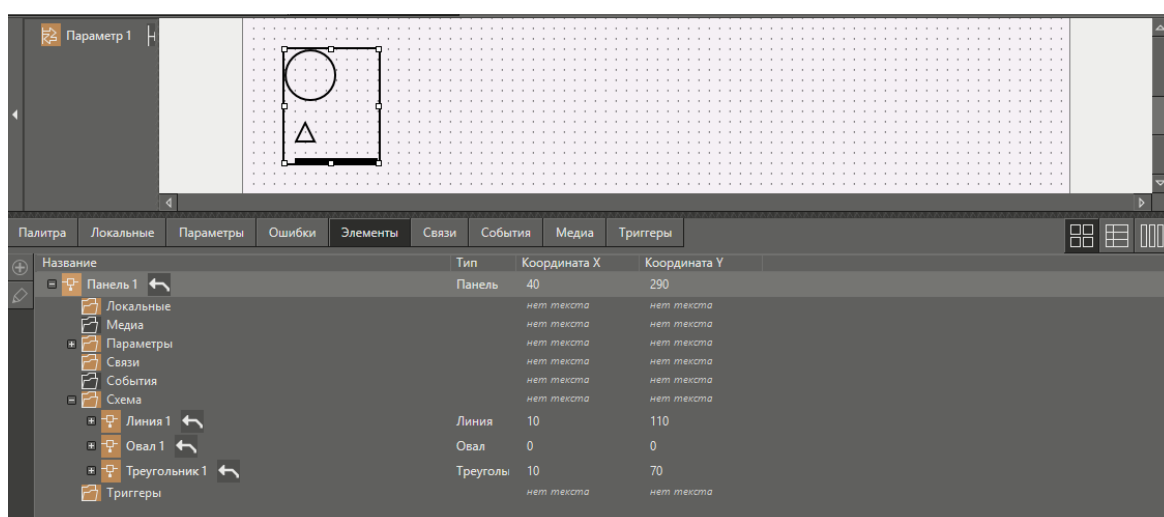
8.3.3.1. ОБЩИЕ ИНСТРУМЕНТЫ ПАНЕЛЕЙ

Элементы, расположенные на панели/вкладке, образуют вместе с этой панелью/вкладкой единый объект. Например, при перемещении панели соответствующим образом перемещаются и элементы, расположенные на ней.

Если свойство Активное содержимое категории Служебные панели/вкладки имеет значение TRUE, то доступно выделение отдельных элементов панели/вкладки. В противном случае, выделение отдельных элементов недоступно. Получить доступ к дочерним элементам можно через панель.

8.3.3.2. ПАНЕЛЬ

Данный элемент предназначен для группировки других элементов. Если выделить несколько элементов и воспользоваться кнопкой панели инструментов или пунктом контекстного меню Группа.Группировать, то появится Панель, содержащая все ранее выделенные элементы:



Для того, чтобы получить доступ к свойствам сгруппированных элементов необходимо на вкладке Элементы в легенде редактора HMI раскрыть дерево элемента Панель, и в группе Схема выбрать нужный элемент, после чего в панели свойств отобразятся свойства выделенного элемента. Панель внутри себя может содержать другие панели.

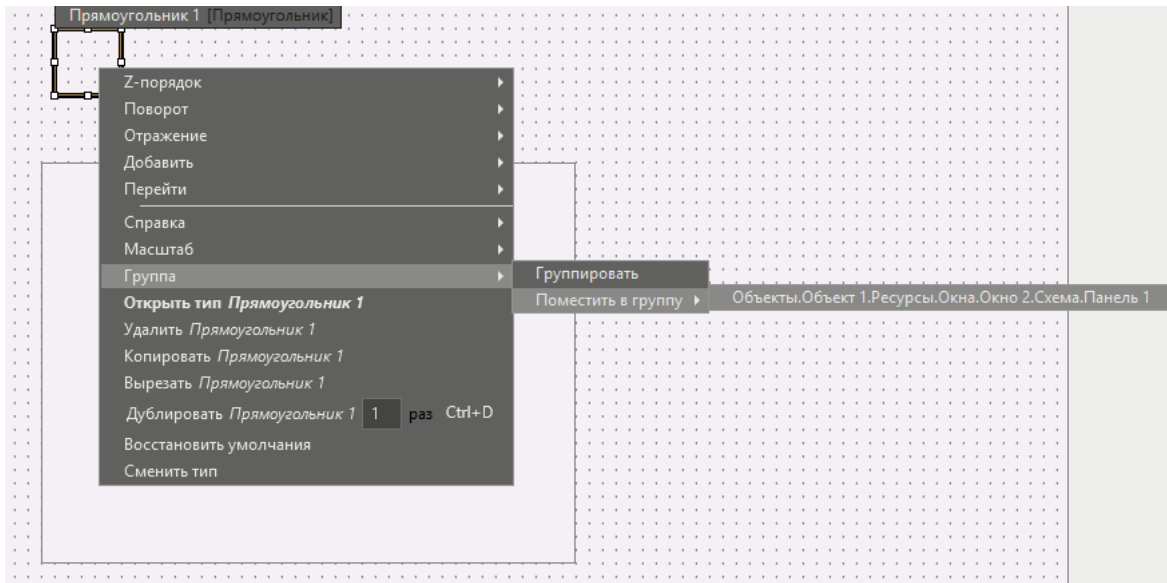
Важно! Изменение размеров панели не приводит к изменению размеров элементов размещенных в панели. Если необходимо, чтобы размеры элементов изменялись пропорционально изменяемой области, то элементы следует разместить в отдельном окне, а затем это окно перетащить в рабочую область редактора HMI.

В палитре редактора HMI элемент Панель находится в категории Панели.

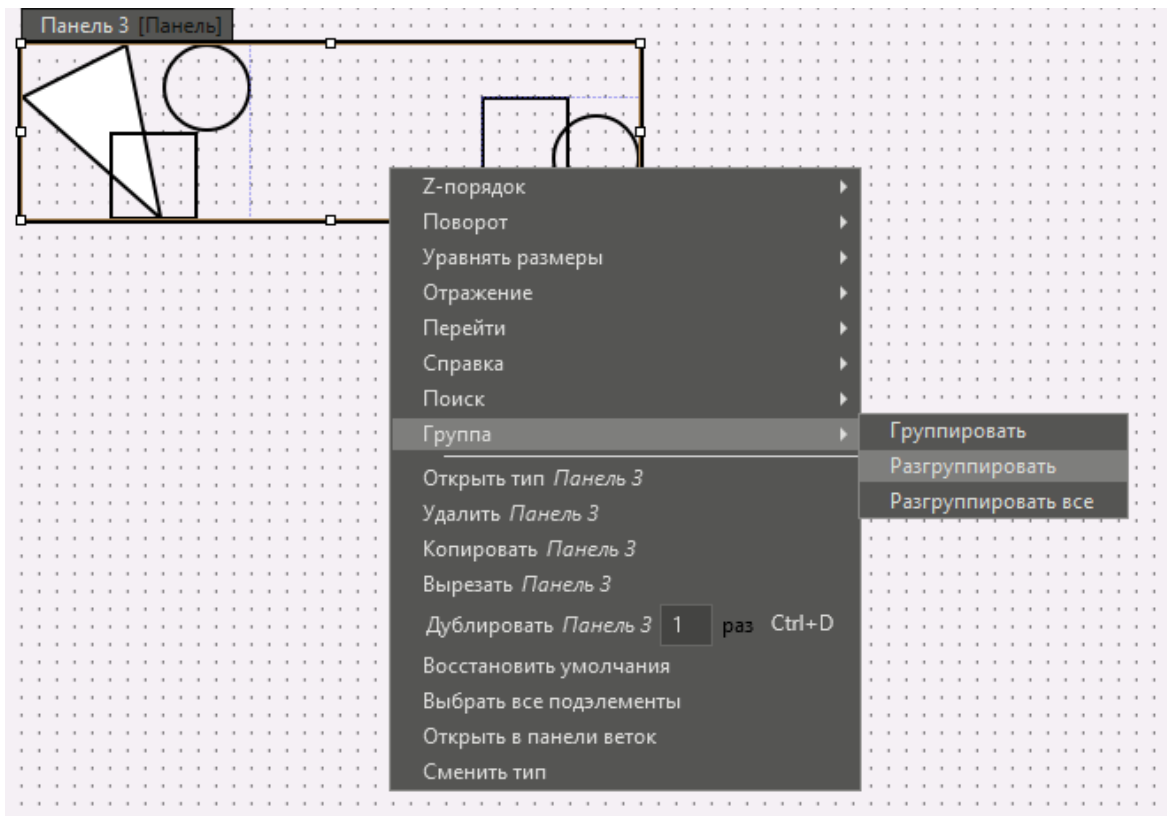
Вид элемента в палитре:



Элемент может быть добавлен из палитры в рабочую область редактора HMI. Для добавления элемента в созданную панель, необходимо выбрать элемент, и в контекстном меню выбрать пункт Группа.Поместить в группу.[путь к панели в дереве]:

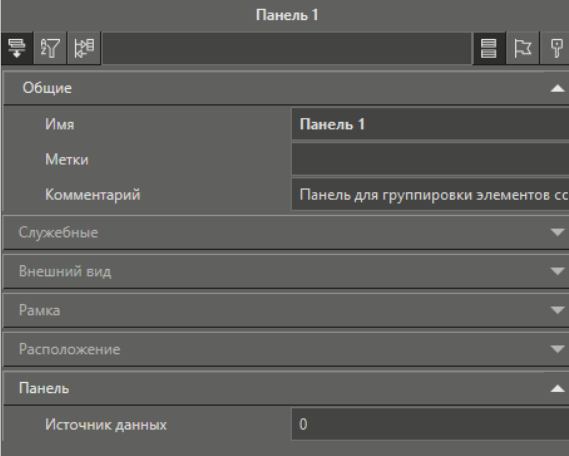
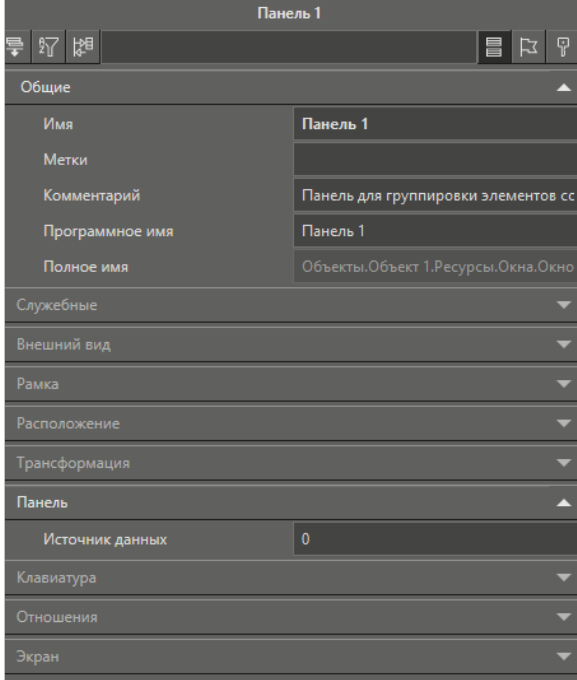


Чтобы разгруппировать элементы нужно воспользоваться пунктами контекстного меню:



- Разгруппировать - разгруппируются элементы на один уровень вложенности
- Разгруппировать все - если панель содержит в себе другие панели, то все они будут разгруппированы.

Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
	

Описание основных свойств графического элемента Панель:

Название	Описание
Категория Панель	
Источник данных	Используется для графического отображения массивов структур, по аналогии со Стековой панелью. Основное отличие, что в стековой панели координаты элементов зависят от свойства Ориентация, а в данном элементе разработчику необходимо задавать координаты самостоятельно.

Описание других свойств элемента смотрите в разделе НМІ.Свойства

8.3.3.3. СТЕКОВАЯ ПАНЕЛЬ

Данный элемент предназначен для отображения массивов структур в графическом виде, когда каждый элемент массива (структура) представлен в виде строки или столбца. Каждое поле структуры в столбце или в строке может выглядеть произвольным образом, т.е в виде любого графического элемента. В режиме разработки формируется шаблон одной строки

или столбца, который в режиме исполнения будет размножен, и каждая строка или столбец станут связаны с соответствующим элементом массива.

В палитре редактора НМІ элемент Стековая панель находится в категории Панели.

Вид элемента в палитре:

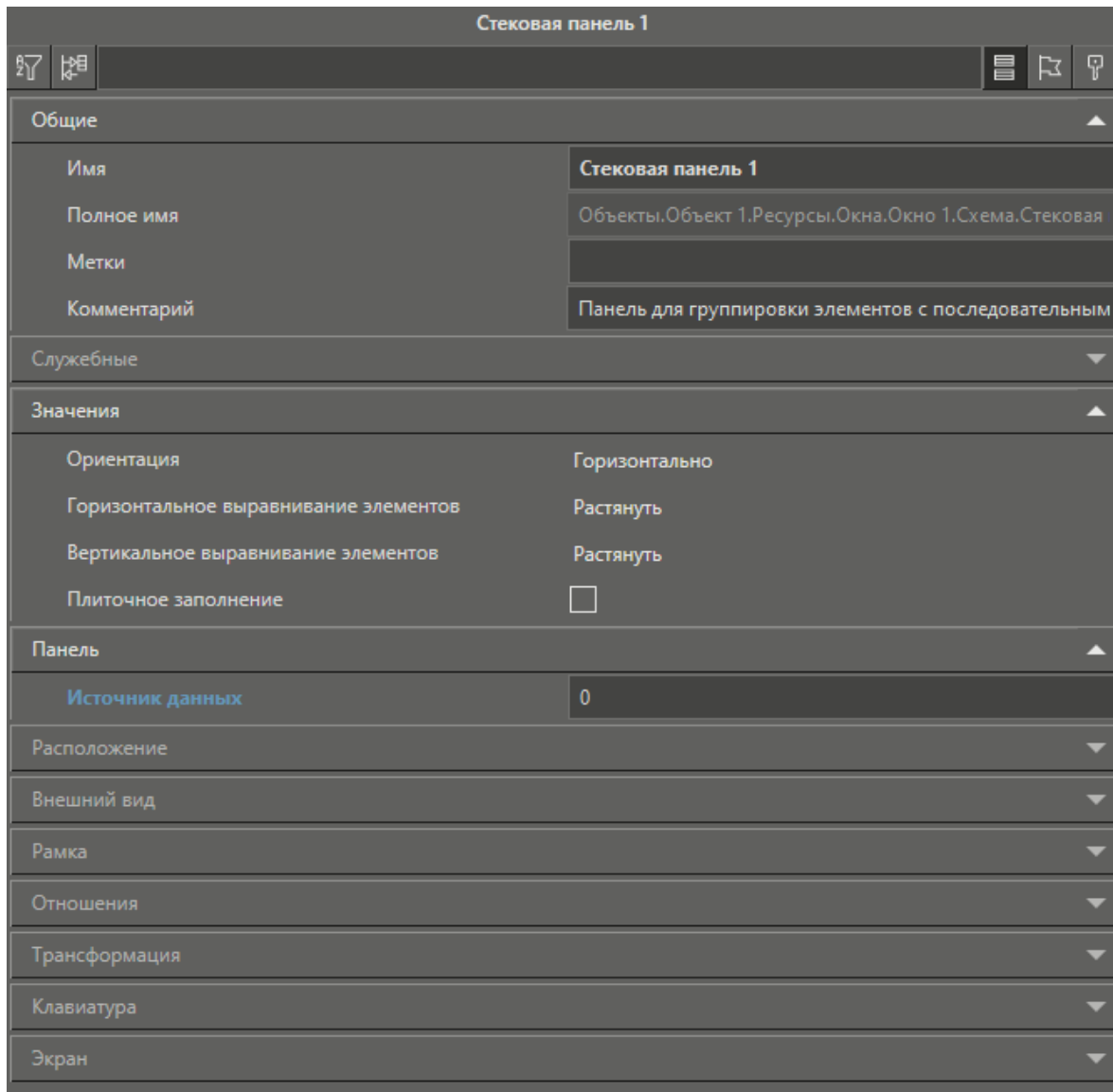


Вид элемента после добавления в рабочую область редактора НМІ:



Работа с элементом начинается с настройки его свойства Источник данных.

Вид панели свойств элемента:



Описание основных свойств графического элемента Панель:

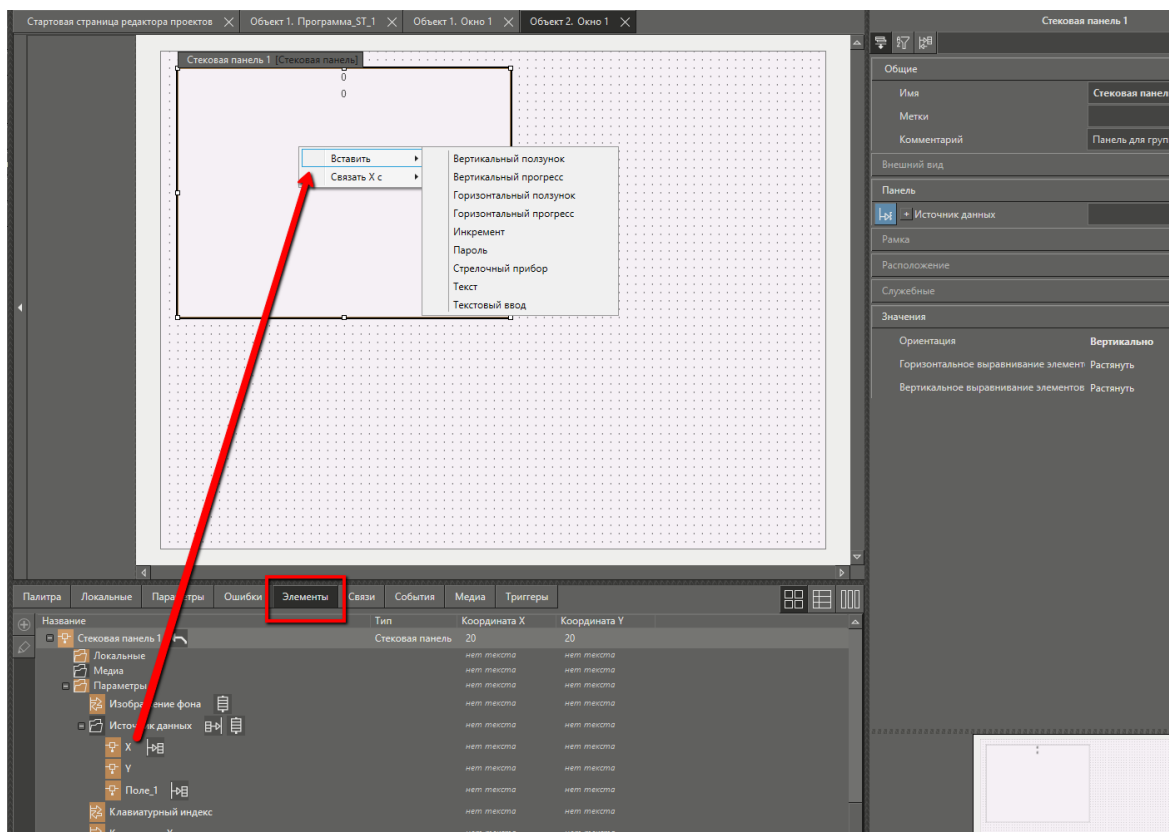
Название	Описание
Категория Значения	
Ориентация	<p>Определяет направление, в котором будут располагаться элементы массива. Возможные варианты:</p> <ul style="list-style-type: none"> Горизонтально - размещаемые на панели элементы последовательно (по порядку размещения) располагаются на панели горизонтально слева направо, и весь набор размещенных элементов прикрепляется к левой границе панели, образуя строку. В режиме исполнения элементы

Название	Описание
	<p>размножаться, количество строк будет совпадать с количеством элементов массива.</p> <ul style="list-style-type: none"> • Вертикально - размещаемые на панели элементы последовательно (по порядку размещения) располагаются на панели вертикально сверху вниз, и весь набор размещенных элементов прикрепляется к верхней границе панели, образуя столбец. В режиме исполнения элементы размножаются, количество строк будет совпадать с количеством элементов массива.
Горизонтальное выравнивание	<p>Определяет поведение итоговой таблицы элементов в панели в режиме исполнения, если размер итоговой таблицы не будет совпадать с размерами панели.</p>
Вертикальное выравнивание	
Плиточное заполнение	<p>Определяется способ заполнения панели элементами, если они не умещаются в одну строку (Ориентация -Горизонтально) или столбец (Ориентация-Вертикально). Если флаг установлен, то при заполнении по строкам высота каждой строки вычисляется как максимальная из высот элементов, которые поместились на этой строке, те что не поместились, начинают заполнять следующую строку и т.д. Если заполнение выполняется по столбцам - не уместившиеся элементы переносятся в следующий столбец.</p> <p>Настройки выравнивания элементов в случае установки флага не используются.</p> <p>Если флаг снят, то все элементы будут располагаться в одной строке или столбце.</p>
<p>Категория Панель</p>	
Источник данных	<p>Используется для графического отображения массивов структур. Данное свойство необходимо связать с параметром типа массив структур. Координаты элементов в итоговой таблице Стековой панели в режиме исполнения в клиенте визуализации зависят от свойства Ориентация</p>

Описание других свойств элемента смотрите в разделе НМІ.Свойства

Порядок работы с элементом Стековая панель

1. Элемент добавляется в рабочую область редактора HMI. Задается его размер.
2. Устанавливается связь Источника данных в параметром проекта.
3. Элементы размещаются в панели. Свойства элементов могут быть связаны только с параметрами Источника данных. Для этого параметры Источника данных могут быть переташены как в свойства различных элементов, так и в поле панели, правой или левой кнопкой мыши:



Доступ к параметрам источника данных может быть получен в легенде редактора HMI на вкладке Элементы, как показано на рисунке выше, либо в полном дереве проекта:

Важно! Если требуется, чтобы элементы в строке или в столбце были размещены произвольно относительно друг друга (не касались друг друга и границ панели), то необходимо объединить их в панель, которая займет положение либо вверху стековой панели, либо у левой границы (в зависимости от значения свойства Ориентация)

Пример работы со стековой панелью описан в разделе Массивы структур в окнах.

8.3.3.4. ПАНЕЛЬ ДОК

Данный элемент предназначен для организации размещения элементов. С её помощью размещаемый на панели элемент прикрепляется к какому-либо краю панели док в зависимости от его свойства Положение категории Расположение, а также аналогичных свойств других размещаемых элементов. Элементы, размещенные на панели, имеют собственные размеры, а последний размещенный элемент может занять всю свободную площадь панели.

Элемент Панель док используется также для формирования шаблона экрана узла в НМІ v1.

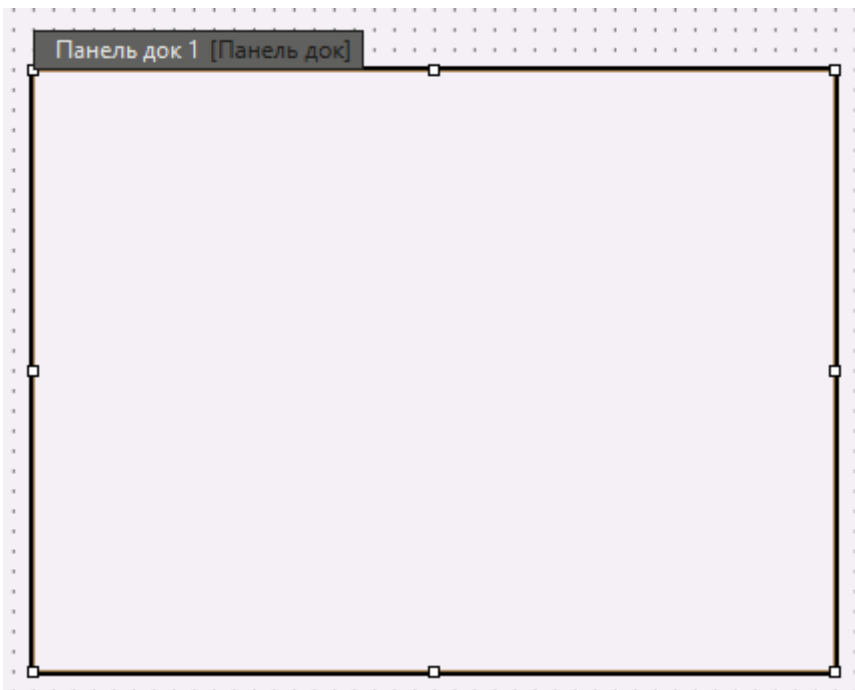
Важно! Данный элемент поддерживается только в НМІ v1.

В палитре редактора НМІ элемент Панель док находится в категории Панели.

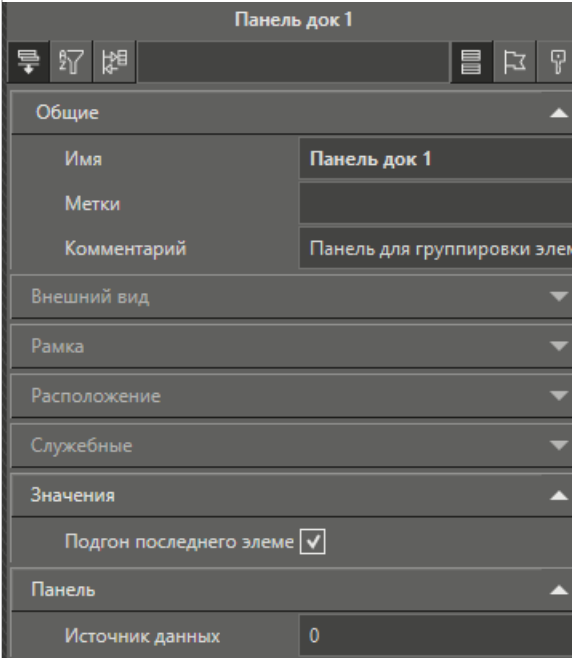
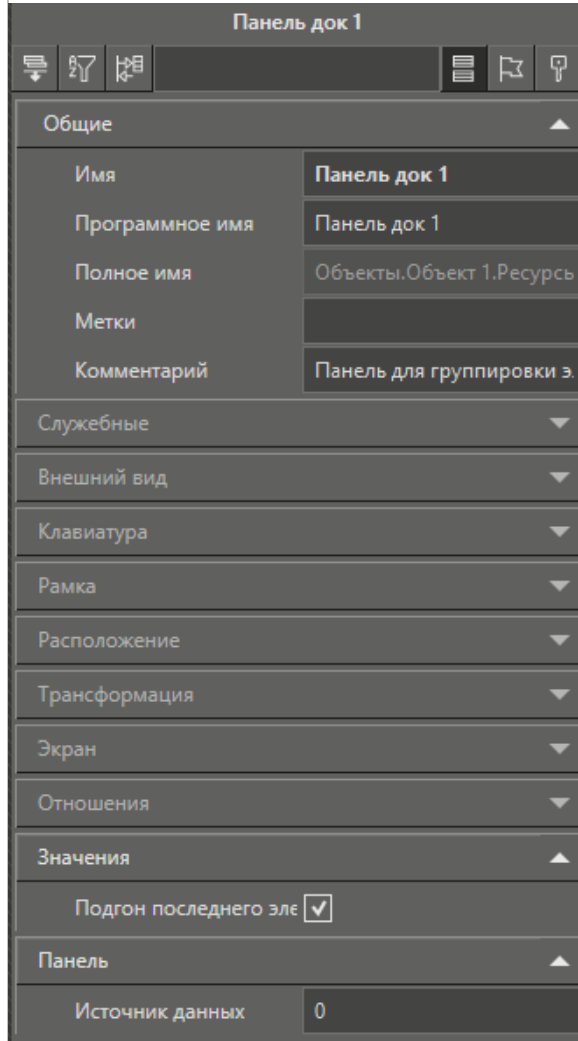
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора НМІ:



Вид панели свойств элемента при нажатой и отжатой кнопке .

Простой режим	Полный режим
	

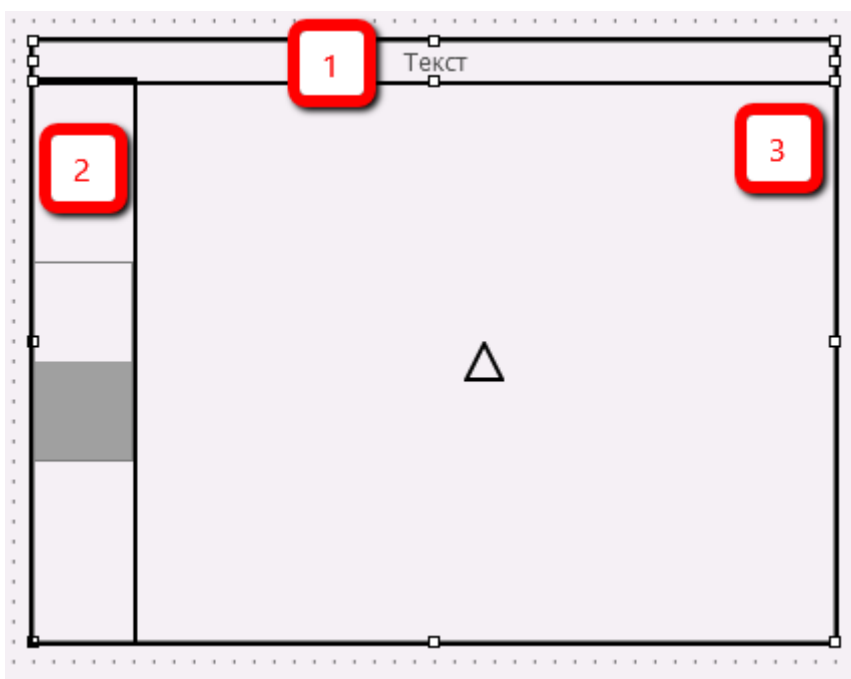
Описание основных свойств графического элемента Панель:

Название	Описание
Категория Значения	
Подгон последнего элемента	Если флаг отмечен, то последний добавленный элемент в панели будет занимать по возможности все свободное место. Если флаг снят, то последний добавленный элемент будет иметь размеры (пропорции), заданные разработчиком.
Категория Панель	

Название	Описание
Источник данных	Не используется.

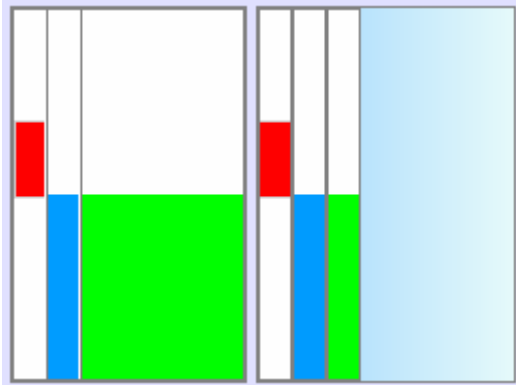
Правила размещения элементов в панели

Первый добавленный элемент займет все свободное место в панели. Размер второго добавленного элемента будет зависеть от свойства Положение категории Расположение как первого элемента, так и второго. Так, например, если свойство Положение первого элемента равно Верх, а второго - Лево, то первый элемент будет иметь ширину, равную ширине панели и произвольную высоту, а второй элемент будет иметь высоту равную высоте панели минус высота первого элемента, т.е. верхний край второго элемента будет упираться в первый и т.д.



Поведение в браузере

Ширина последнего размещенного элемента зависит от состояния флага свойства Подгон последнего элемента. На левом рисунке показано размещение в случае, когда флаг Подгон последнего элемента отмечен, а на правом - когда снят"



8.3.3.5. ПАНЕЛЬ ВКЛАДОК

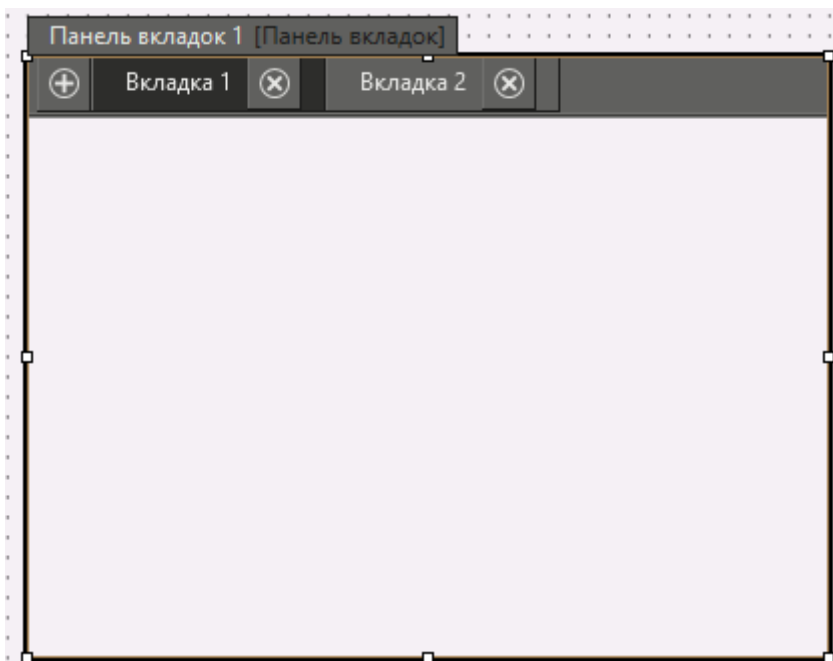
Данный элемент представляет собой набор вкладок и предназначен для объединения в группы других элементов.

В палитре редактора НМІ элемент Панель вкладок находится в категории Панели.

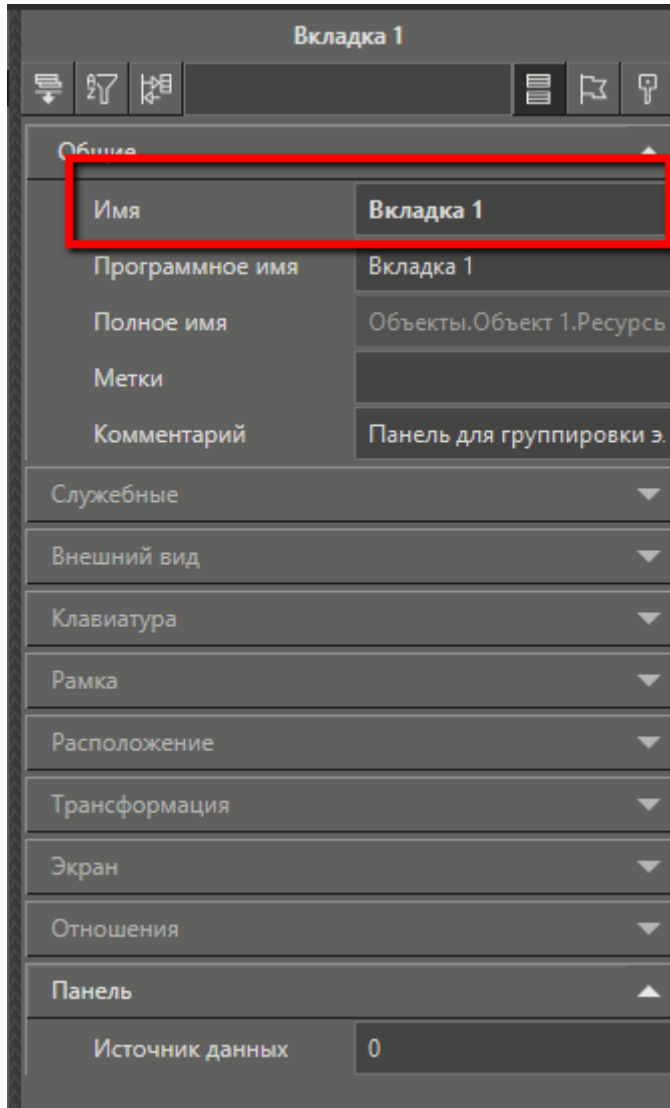
Вид элемента в палитре:





Вид элемента после добавления в рабочую область редактора НМІ:



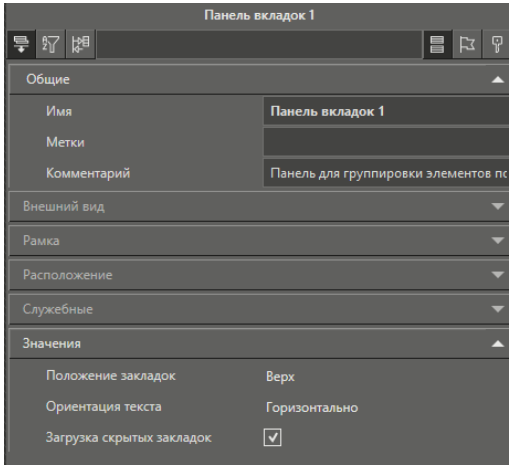
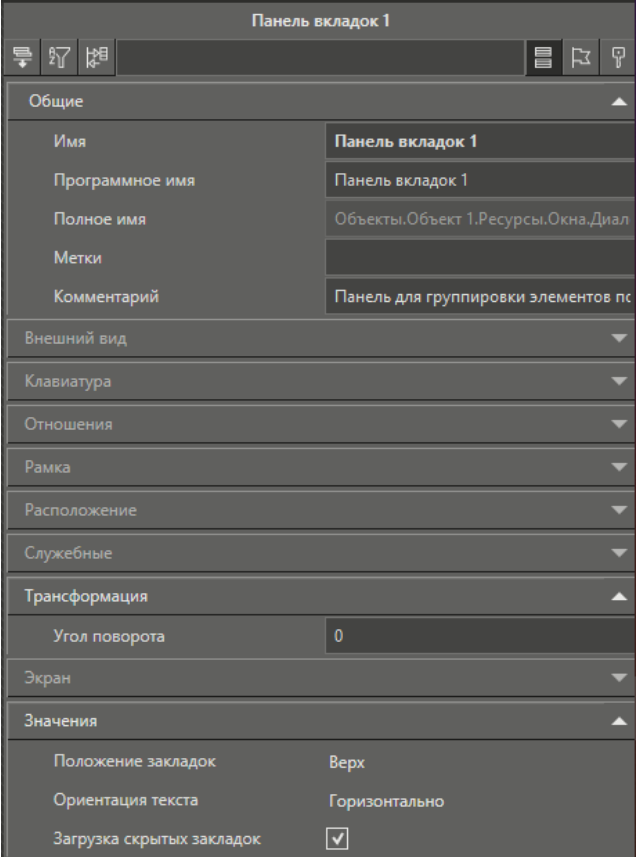
Каждая вкладка представляет собой элемент Панель. Для настройки свойств каждой отдельной вкладки, необходимо нажать левой кнопкой мыши на её название, а затем на требуемое поле панели. В этом случае, в панели свойств отобразятся свойства выделенной вкладки, где можно изменить имя вкладки:



Для добавления вкладки следует нажать на кнопку , а для удаления ранее созданной вкладки - на кнопку .

Для того чтобы на панели свойств отобразились свойства Панели вкладок, необходимо нажать левой кнопкой мыши на область названий вкладок.

Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
	

Описание основных свойств графического элемента Панель вкладок:

Название	Описание
Категория Значения	
Положение вкладок	Определяет расположение заголовков вкладок. Возможные варианты: Верх, Низ, Лево, Право.
Ориентация текста	Определяет направление текста в заголовках вкладок. Возможные варианты: Вертикально, Горизонтально.
Загрузка скрытых вкладок	Определяет необходимость загрузки данных, расположенных на скрытых вкладках в режиме исполнения.

Описание других свойств элемента смотрите в разделе НМІ.Свойства

8.3.3.6. КОНТЕЙНЕР ОКНА

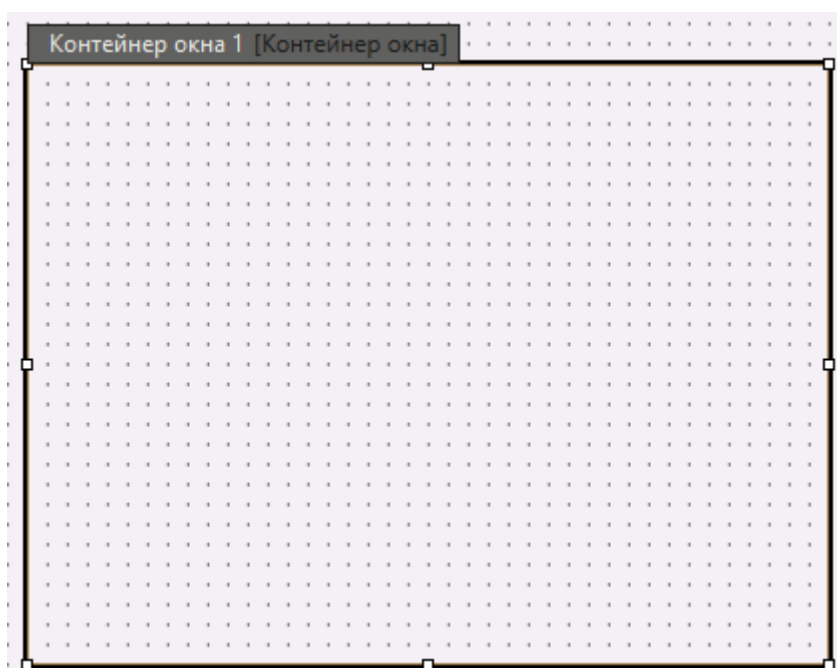
Данный элемент предназначен для отображения содержимого различных окон или страниц сайтов.

В палитре редактора HMI элемент Контейнер окна находится в категории Панели.

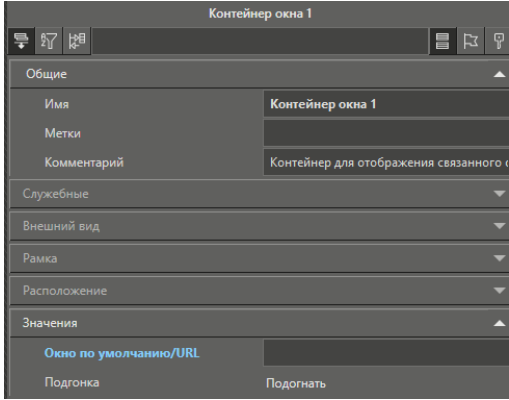
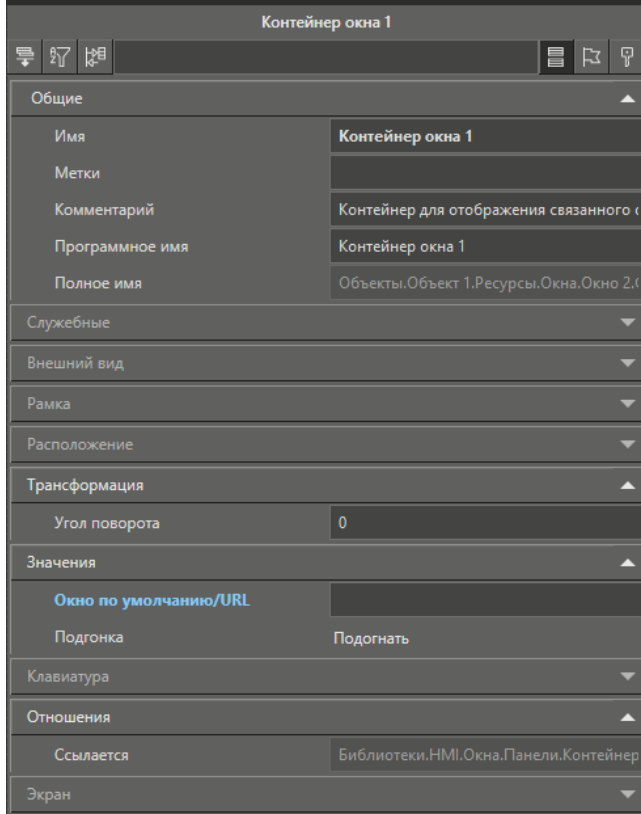
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:



Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
	

Описание основных свойств графического элемента Контейнер окна:

Название	Описание
<p>Категория Значения</p>	
<p>Окно по умолчанию/URL</p>	<p>Указывает, что должно отображаться в контейнере. Если перетащить в данное поле какое-либо окно из дерева, то в режиме исполнения в контейнере будет отображаться содержимое этого окна. Если в данном поле задать адрес страницы сайта, то в режиме исполнения в контейнере будет отображаться содержимое этой страницы (при наличии доступа к сайту).</p>
<p>Подгонка</p>	<p>Определяет, каким образом будет отображаться содержимое окна или страницы сайта в случае, если их размер отличается от размера контейнера. .</p> <p>Возможны следующие значения:</p> <ul style="list-style-type: none"> Исходный размер - не используется;

Название	Описание
	<ul style="list-style-type: none"> • Подогнать - отображаемое содержимое масштабируется под размеры контейнера; • Обрезать - в случае превышения размеров контейнера, содержимое при отображении будет совмещено с левым верхним углом контейнера и обрезано снизу и справа; • Прокрутить - если содержимое превышает размеры контейнера, то появляются линейки прокрутки.

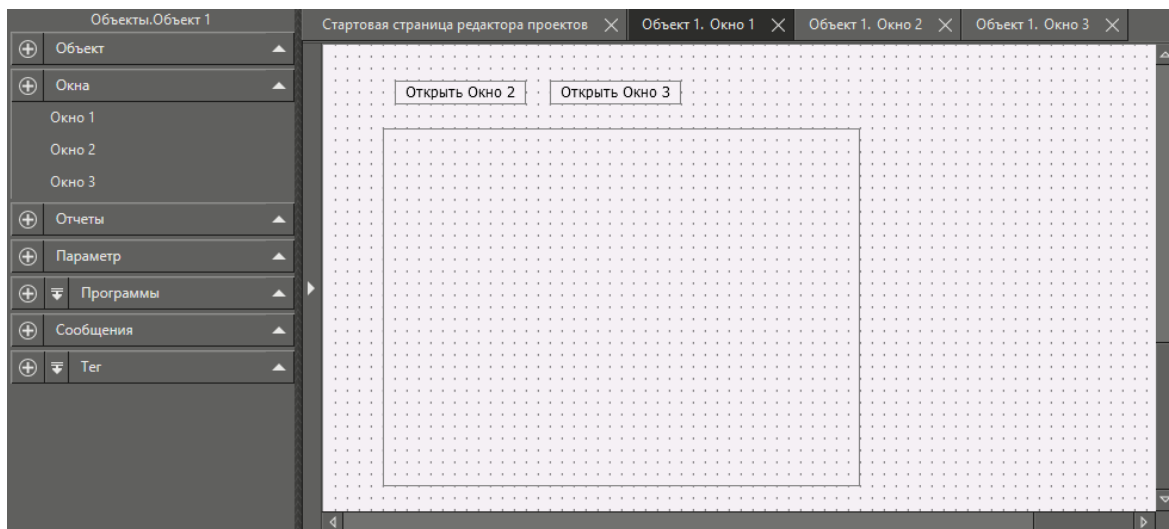
Описание других свойств элемента смотрите в разделе НМІ.Свойства

Рекомендации

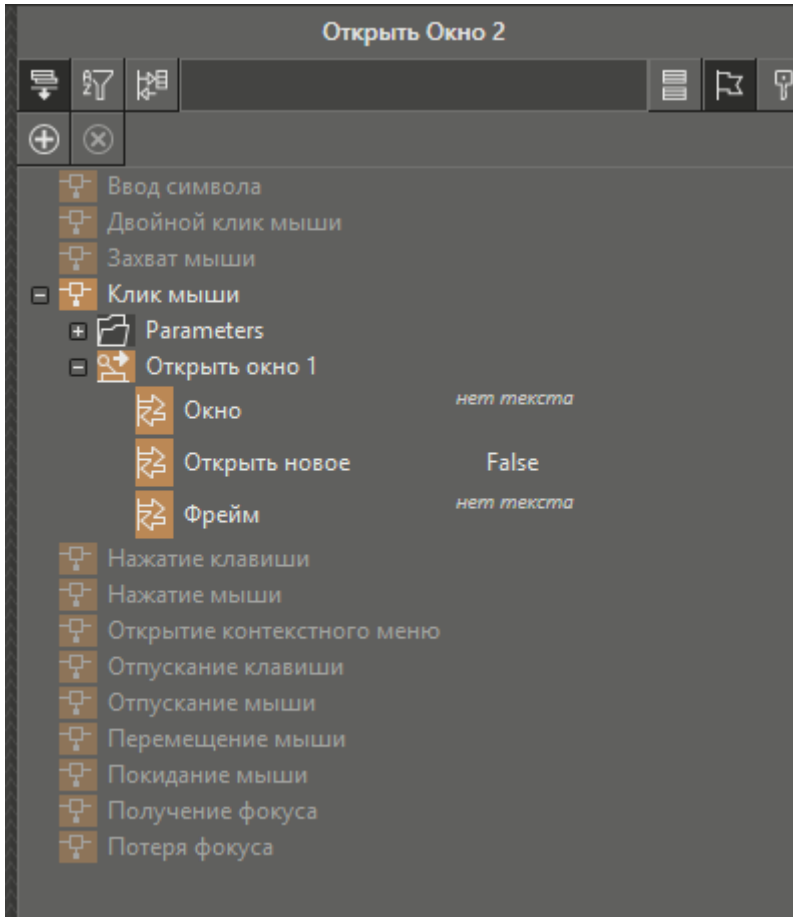
Отображение различных окон в одном контейнере

Рассмотрим пример, когда требуется при нажатии на кнопки открывать разные окна в одном контейнере.

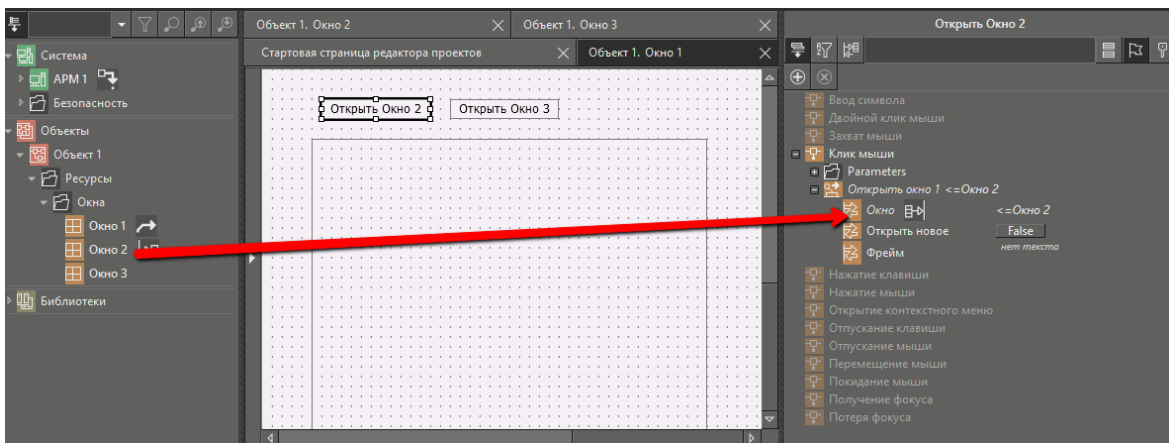
Допустим имеется три окна: Окно 1 - основное стартовое окно, которое содержит элемент Контейнер окна и две кнопки. Окно 2 и Окно 3 должны открываться при нажатии на эти кнопки.



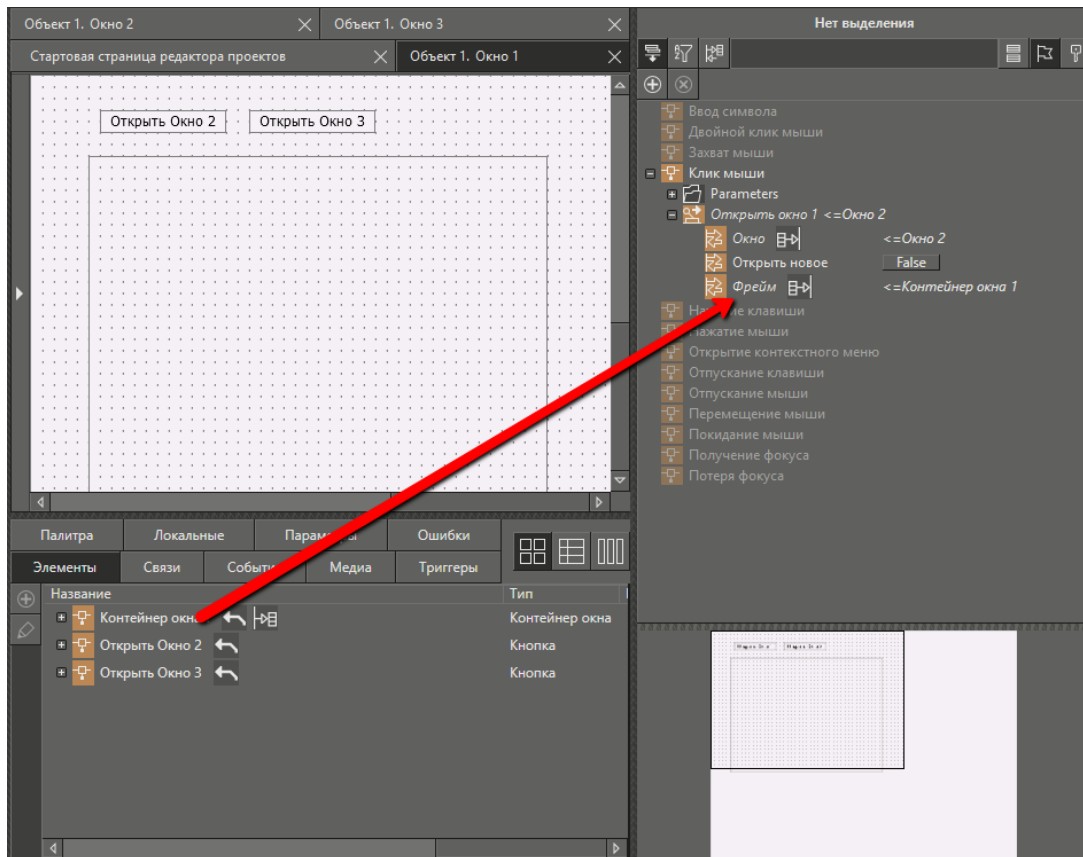
Выделим кнопку Открыть окно 2. Перейдем в ее панель свойств, переключимся в режим работы с событиями. И добавим в событие Клик мыши действие Открыть окно:



Затем перетащим левой кнопкой мыши Окно 2 из дерева в параметр действия Окно:



Откроем вкладку Элементы в легенде редактора HMI. Перетащим элемент Контейнер окна на параметр Фрейм действия:



Аналогичные операции необходимо проделать с кнопкой Открыть окно 3.

- Добавить в событие Клик мыши действие Открыть окно
- Перетащить Окно 3 на параметр действия Окно
- Перетащить Контейнер окна из легенды на свойство Фрейм.

Вид в режиме исполнения после нажатия на кнопку Открыть окно 2

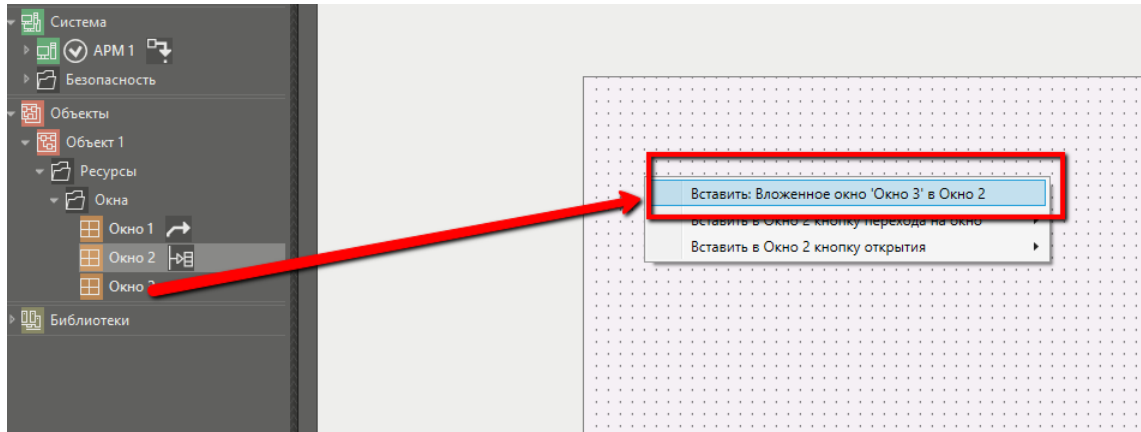


Отображение различных сайтов в одном контейнере

Допустим, что при нажатии на одну кнопку требуется отображать сайт 1, а при нажатии на другую - сайт 2. Для этого применим точно такой же алгоритм как и в описанном выше случае, но только вместо действия Открыть окно используется действие Открыть адрес

Отображение одного окна в другом окне

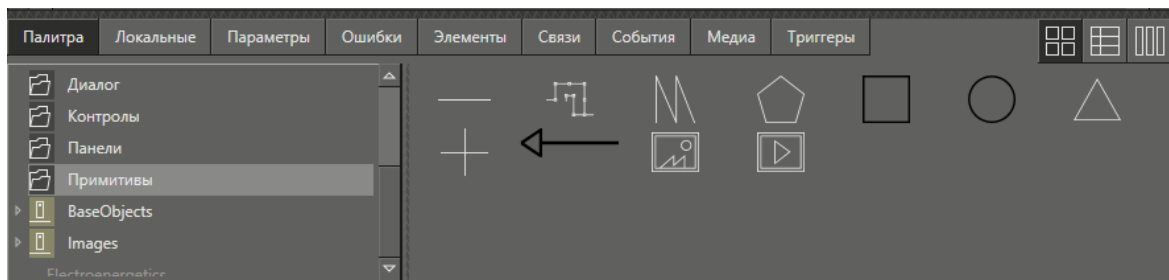
Если необходимо, чтобы в одном окне всегда отображалось одно и то же другое окно, то можно использовать контейнер окна, перетащив требуемое окно в свойство Окно по умолчанию/URL. Однако можно просто перетащить левой кнопкой мыши окно, требующее отображения, в нужное место другого окна. И в появившемся списке выбрать:



8.3.4. КАТЕГОРИЯ ПРИМИТИВЫ

Категория Прimitives палитры редактора НМІ содержит вспомогательные элементы, которые могут быть использованы как в статическом виде (свойства элементов не изменяются в клиенте визуализации в зависимости от каких-либо параметров проекта), так и в динамическом (все свойства элементов могут быть динамизированы, а также при нажатии на эти элементы могут выполняться различные действия).

В палитре редактора НМІ эта категория имеет вид:



Категория Прimitives включает в себя следующие элементы:

- Графический элемент Линия
- Графический элемент Граф
- Графический элемент Ломаная линия
- Графический элемент Полигон
- Графические элементы Овал, Прямоугольник, Треугольник
- Графический элемент Крест
- Графический элемент Стрелка
- Графический элемент Картинка
- Графический элемент Видео

8.3.4.1. ЛИНИЯ

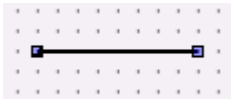
Данный элемент предназначен для рисования прямых линий.

В палитре редактора HMI элемент Линия находится в категории Прimitives.

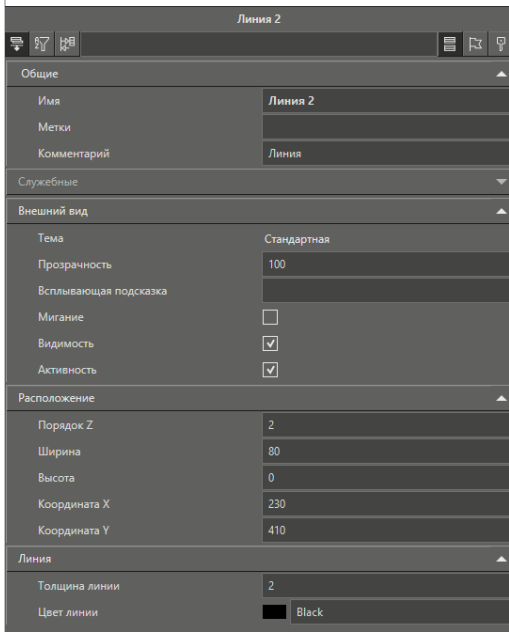
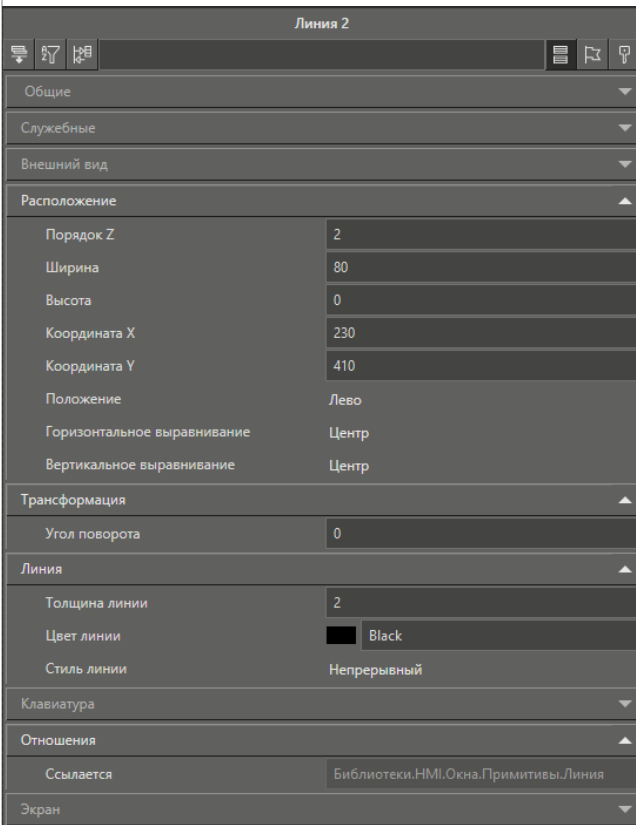
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:



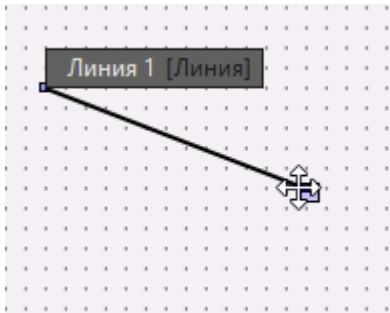
Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
	

Описание свойств элемента смотрите в разделе HMI.Свойства

Рекомендации

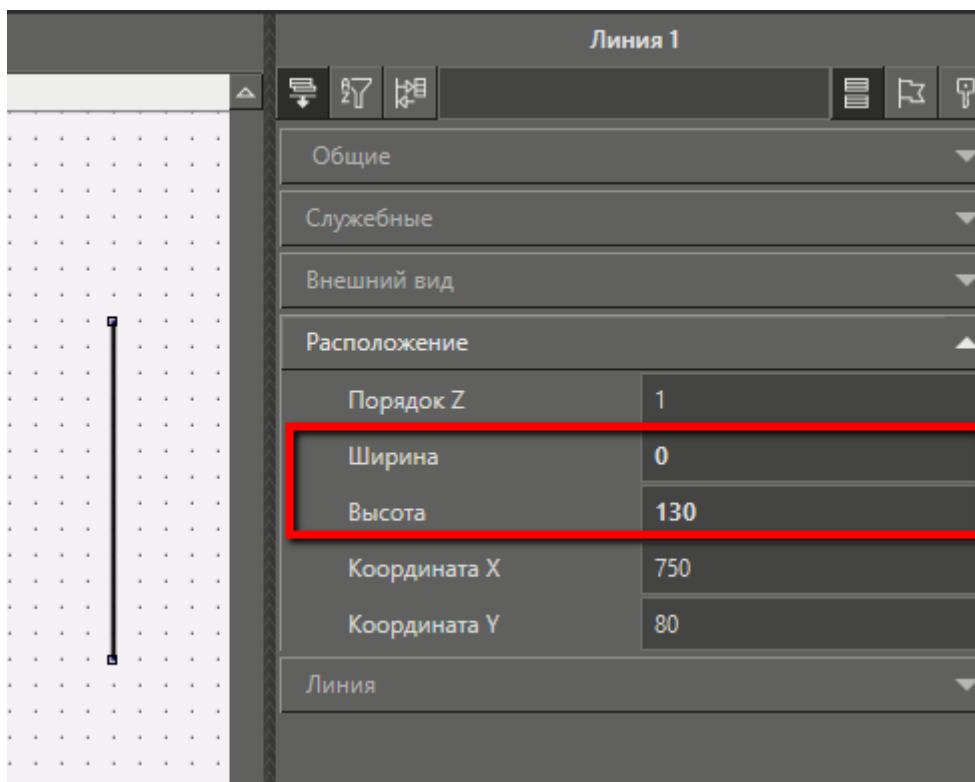
- Для того чтобы изменить угол наклона можно потянуть начало или конец линии левой кнопкой мыши. При этом указатель мыши будет изменен:



- Для того чтобы нарисовать вертикальную линию, можно установить следующие значения свойств в категории Расположение:

Ширина=0

Высота=[необходимая длина линии]



- Для того чтобы рисовать сложные геометрические фигуры из линий, необходимо использовать элементы Граф и Ломаная линия.

8.3.4.2. ГРАФ

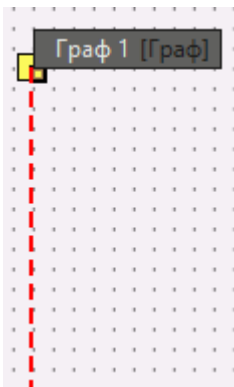
Данный элемент предназначен для рисования ломаных линий с ответвлениями, представляет собой SVG-фигуру.

В палитре редактора НМІ элемент Граф находится в категории Прimitives.

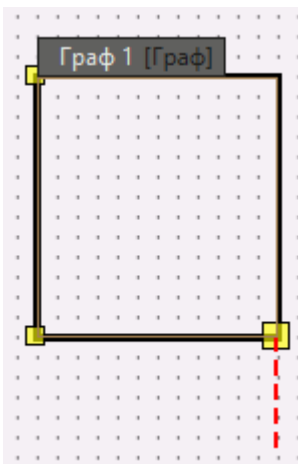
Вид элемента в палитре:



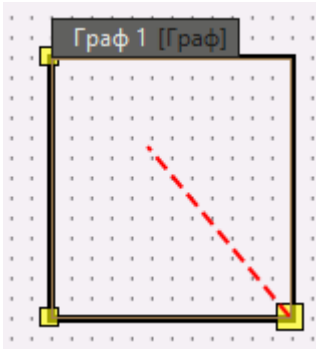
После добавления элемента в рабочую область редактора HMI граф будет находиться в режиме редактирования, в котором необходимо задать его конфигурацию. Сначала появится первая точка, из которой будет выходить пунктирная линия, направление которой будет следовать за движением мыши:



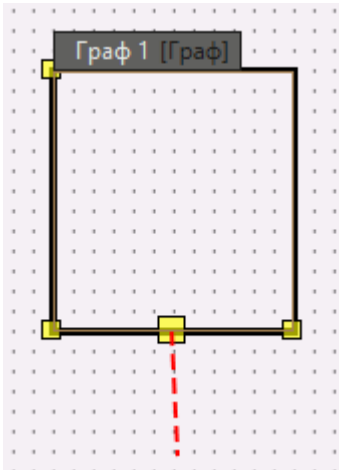
Если нажать левой кнопкой мыши на какое-либо место рабочей области, то будет создана вторая точка. Если переместить мышь и опять нажать левую кнопку, то будет создана третья точка:



В режиме редактирования точки графа обозначены желтыми квадратами. Точка, из которой выходит пунктирная линия, имеет больший размер. По умолчанию, флаг в свойстве Ортогональность отмечен, поэтому линия будет рисоваться под прямым углом. Это означает, что новая точка будет создаваться не точно в месте нажатия левой кнопки мыши, а максимально близко к нему, так чтобы получился прямой угол. В случае снятого флага Ортогональность новая точка будет создаваться точно в месте нажатия левой кнопки мыши:



Для того чтобы нарисовать линию из произвольного места графа (там где точка может быть поставлена, появится ее макет, граница которого будет задана пунктирной линией), необходимо нажать левую кнопку мыши, при этом в месте нажатия будет создана точка, и из неё будет выходить линия (ответвление).

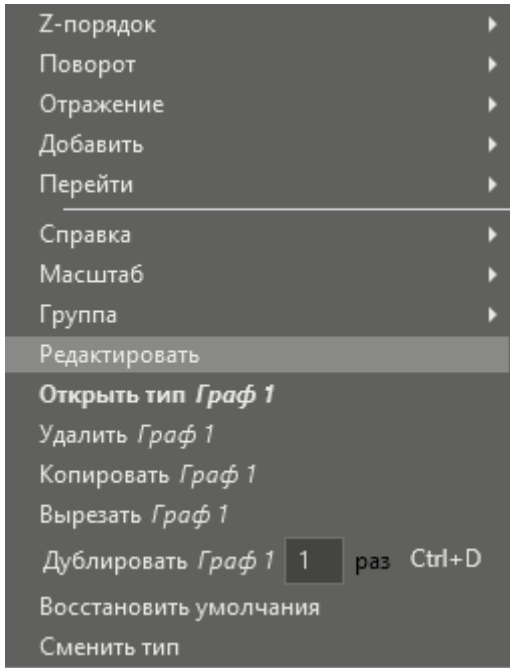


Для удаления существующей точки, нужно дважды нажать на неё левой кнопкой мыши, после чего точка удалится, а граф перерисуеться.

Чтобы изменить точку от которой рисуется новый участок необходимо выбрать ее левой кнопкой мыши, при этом эта точка станет больше остальных и из нее будет выходить красная пунктирная линия.

Для того чтобы прекратить редактирование графа, необходимо нажать клавишу ESC на клавиатуре или правую кнопку мыши.

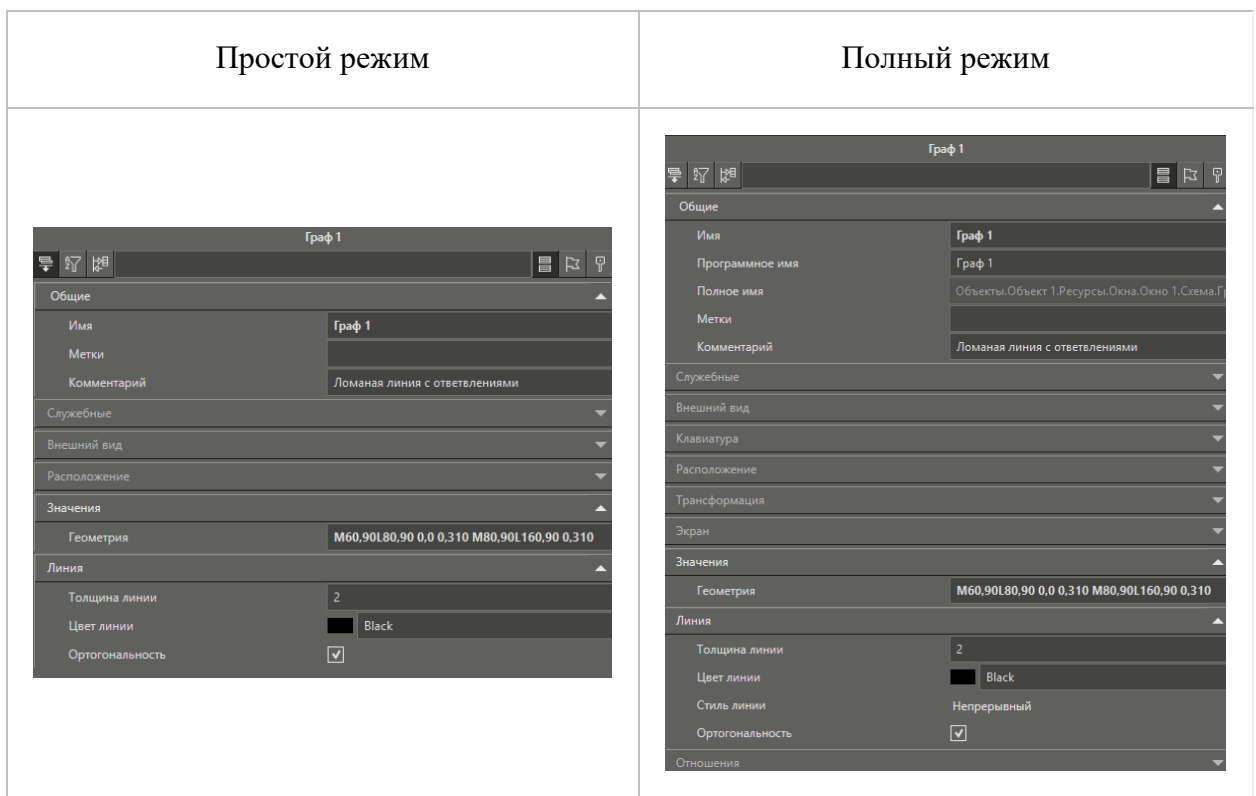
Для того чтобы войти в режим редактирования ранее нарисованного графа, необходимо нажать правую кнопку мыши, и в контекстном меню выбрать пункт Редактировать:



Редактирование продолжится с того места, на котором было закончено ранее.

Результат редактирования графа отобразится в свойстве Геометрия. При необходимости, можно отредактировать граф, используя данное свойство.

Вид панели свойств элементов при нажатой и отжатой кнопке :



Описание основных свойств графического элемента Граф:

Название	Ссылка на описание
Категория Значение	
Геометрия	Задается выражение, описывающее фигуру.
Категория Линия	
Ортогональность	Определяет способ рисования новой линии: под прямым или под произвольным углом.

Описание других свойств элементов смотрите в разделе НМІ.Свойства

8.3.4.3. ЛОМАНАЯ ЛИНИЯ

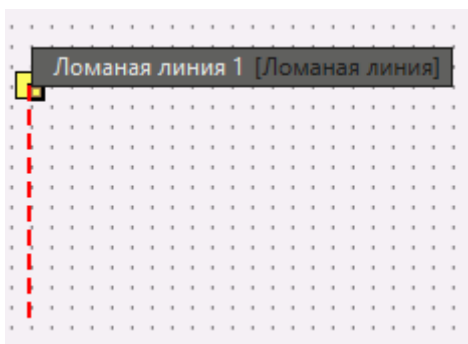
Данный элемент предназначен для рисования ломаных линий, представляет собой SVG-фигуру.

В палитре редактора НМІ элемент Ломаная линия находится в категории Прimitives.

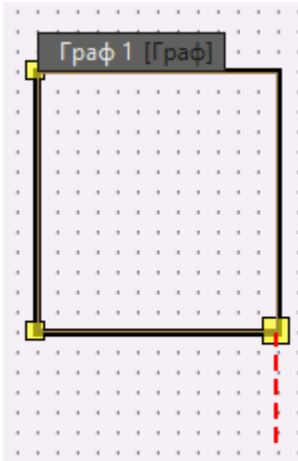
Вид элемента в палитре:



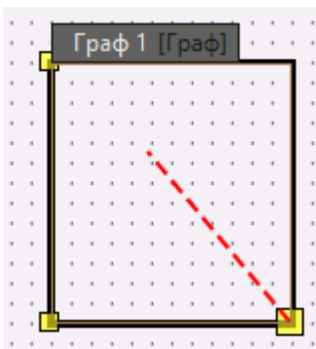
После добавления элемента в рабочую область редактора НМІ ломаная линия будет находиться в режиме редактирования, в котором необходимо задать её конфигурацию. Сначала появится первая точка и из нее будет выходить пунктирная линия, направление которой будет следовать за движением мыши:



Если нажать левой кнопкой мыши на какое-либо место рабочей области, то будет создана вторая точка. Если переместить мышь и опять нажать левую кнопку, то будет создана третья точка:

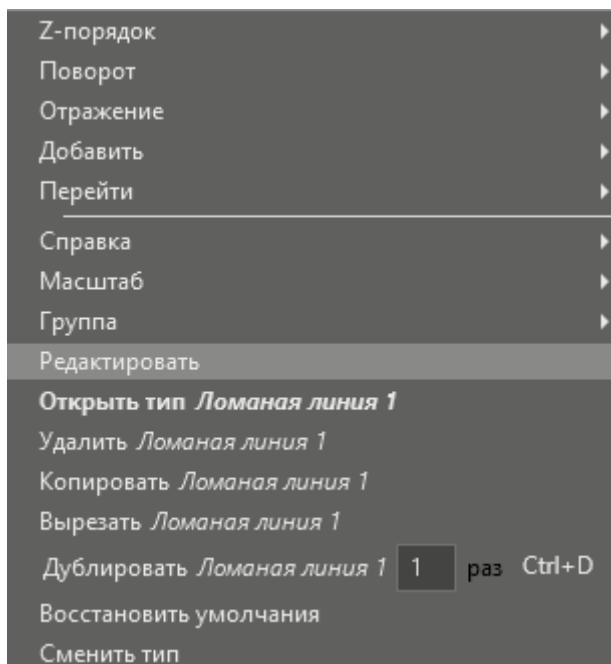


По умолчанию, флаг в свойстве Ортогональность отмечен, поэтому линия будет рисоваться под прямым углом. Это означает, что новая точка будет создаваться не точно в месте нажатия левой кнопки мыши, а максимально близко к нему, так чтобы получился прямой угол. В случае снятого флага Ортогональность новая точка будет создаваться точно в месте нажатия левой кнопки мыши:



Для того чтобы прекратить редактирование Ломаной линии необходимо нажать клавишу ESC на клавиатуре или правую кнопку мыши.

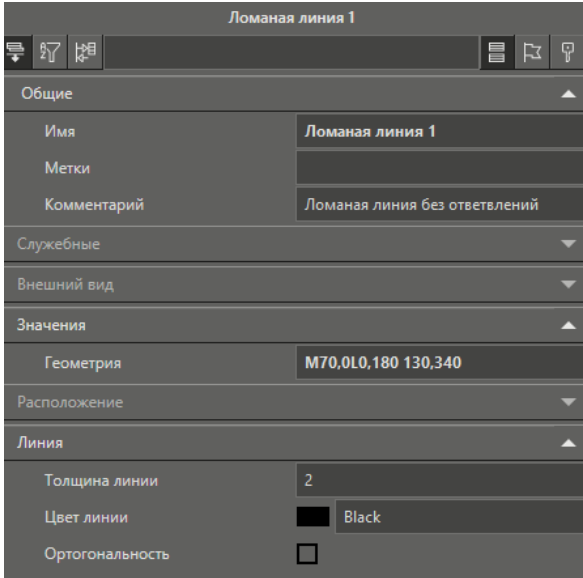
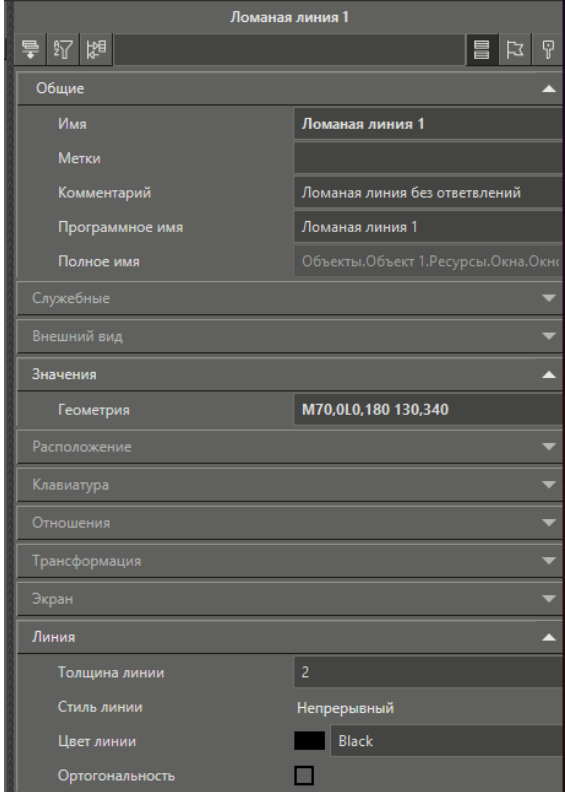
Для того чтобы войти в режим редактирования ранее нарисованной Ломаной линии, необходимо нажать правую кнопку мыши и в контекстном меню выбрать пункт Редактировать:



Для того чтобы добавить новую точку, необходимо дважды нажать на линию левой кнопкой мыши. В месте нажатия кнопки будет создан дополнительный излом линии (точка). Чтобы удалить точку, нужно дважды нажать на неё левой кнопкой мыши, после чего точка удалится, а Ломаная линия перерисуеться.

Результат редактирования Ломаной линии отобразится в свойстве Геометрия. При необходимости, Ломаную линию можно отредактировать, используя данное свойство.

Вид панели свойств элементов при нажатой и отжатой кнопке :

Простой режим	Полный режим
	

Описание основных свойств графического элемента Ломаная линия:

Название	Ссылка на описание
Категория Значение	
Геометрия	Задается выражение, описывающее фигуру.
Категория Линия	
Ортогональность	Определяет способ рисования новой линии: под прямым или под произвольным углом.

Описание других свойств элементов смотрите в разделе НМІ.Свойства

8.3.4.4. ПОЛИГОН

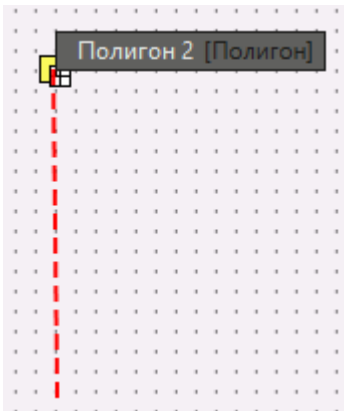
Данный элемент предназначен для рисования произвольных геометрических фигур, представляет собой SVG-фигуру.

В палитре редактора НМІ элемент Полигон находится в категории Прimitives.

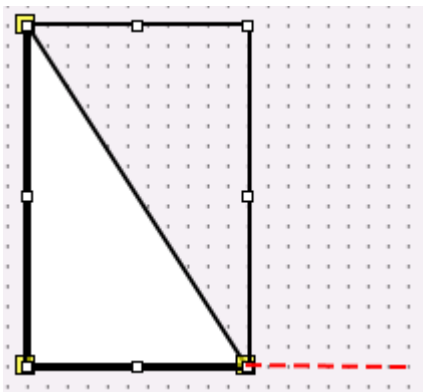
Вид элемента в палитре:



После добавления элемента в рабочую область редактора НМІ полигон будет находиться в режиме редактирования, в котором необходимо задать его конфигурацию. Сначала появится первая точка и из нее будет выходить пунктирная линия, направление которой будет следовать за движением мыши:



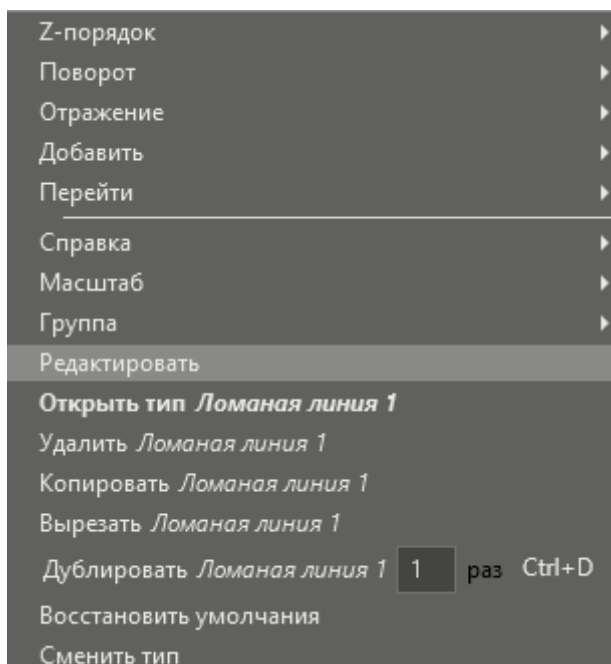
Если нажать левой кнопкой мыши на какое-либо место рабочей области, то будет создана вторая точка. Если переместить мышь и опять нажать левую кнопку, то будет создана третья точка:



По умолчанию, флаг в свойстве Ортогональность отмечен, поэтому линия будет рисоваться под прямым углом. Это означает, что новая точка будет создаваться не точно в месте нажатия левой кнопки мыши, а максимально близко к нему, так чтобы получился прямой угол. В случае снятого флага Ортогональность новая точка будет создаваться точно в месте нажатия левой кнопки мыши.

Для того чтобы прекратить редактирование Полигона необходимо нажать клавишу ESC на клавиатуре или правую кнопку мыши.

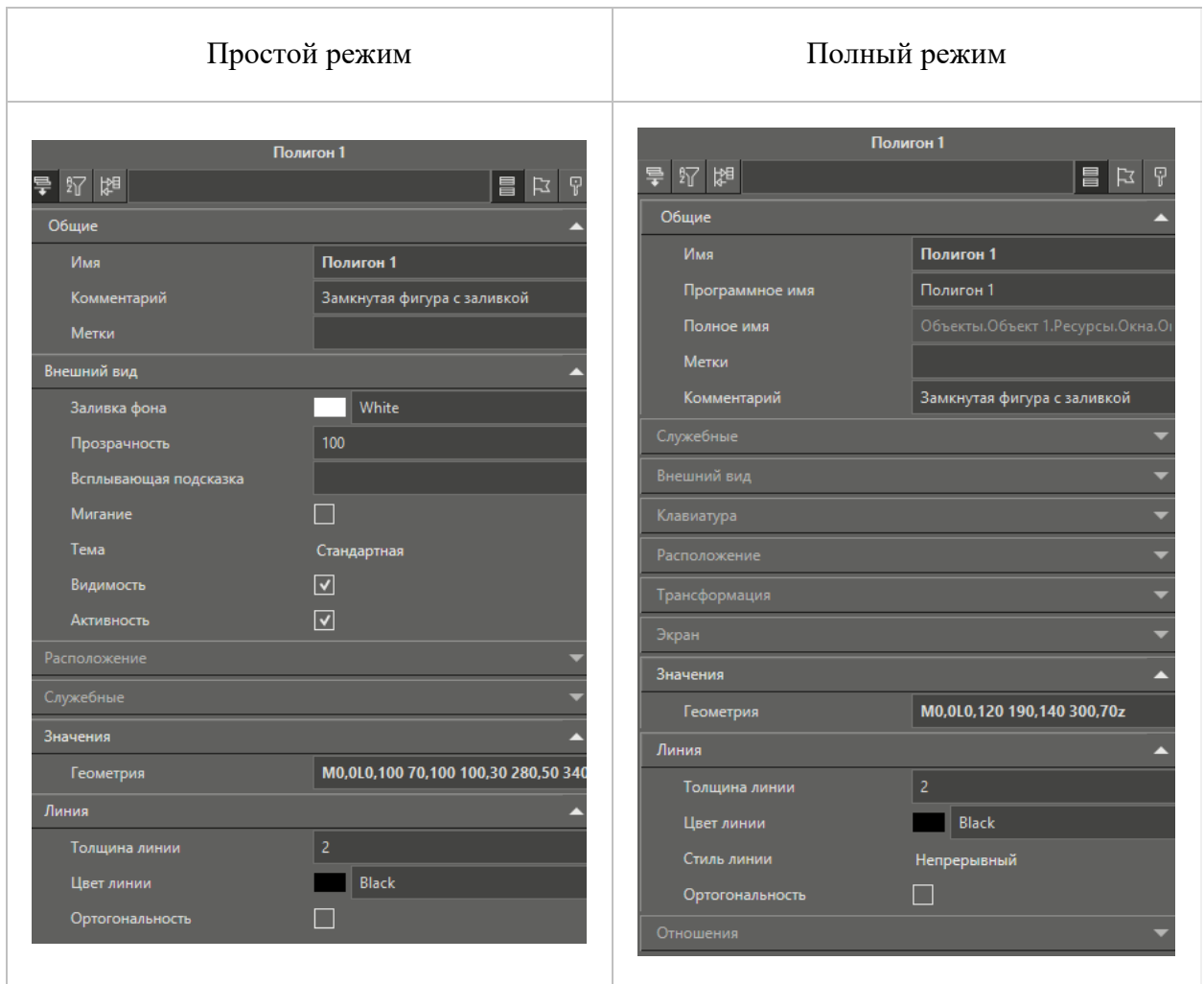
Для того чтобы войти в режим редактирования ранее нарисованной фигуры, необходимо нажать правую кнопку мыши и в контекстном меню выбрать пункт Редактировать:



Для того чтобы добавить новую точку, необходимо дважды нажать левой кнопкой мыши на нужное место периметра. В месте нажатия будет создан дополнительный излом границ (точка). Чтобы удалить точку, нужно дважды нажать на неё левой кнопкой мыши, после чего точка удалится, а Полигон перерисуеться.

Результат редактирования Полигона отобразится в свойстве Геометрия. При необходимости можно отредактировать фигуру, используя данное свойство.

Вид панели свойств элементов при нажатой и отжатой кнопке :



Описание основных свойств графического элемента Полигон:

Название	Ссылка на описание
Категория Значение	
Геометрия	Задается выражение, описывающее фигуру.
Категория Линия	
Ортогональность	Определяет способ рисования новой линии: под прямым или под произвольным углом.

Описание других свойств элементов смотрите в разделе НМІ.Свойства


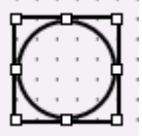

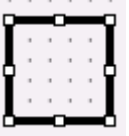


Создание пользовательской библиотеки примитивов


8.3.4.5. ОВАЛ, ПРЯМОУГОЛЬНИК, ТРЕУГОЛЬНИК

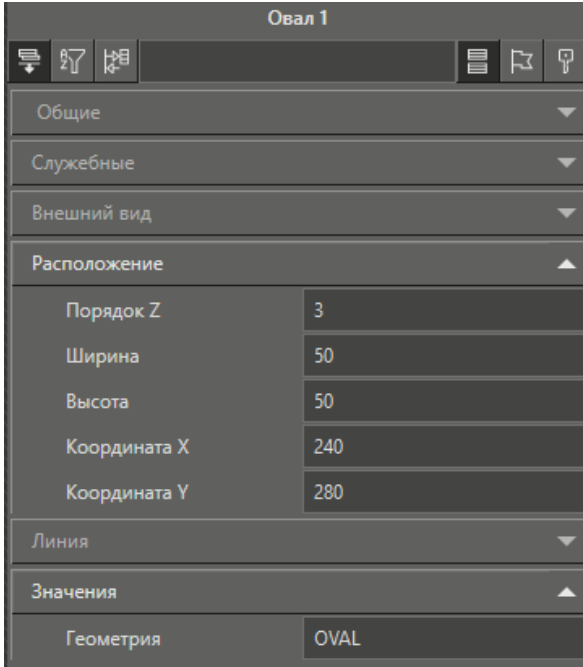
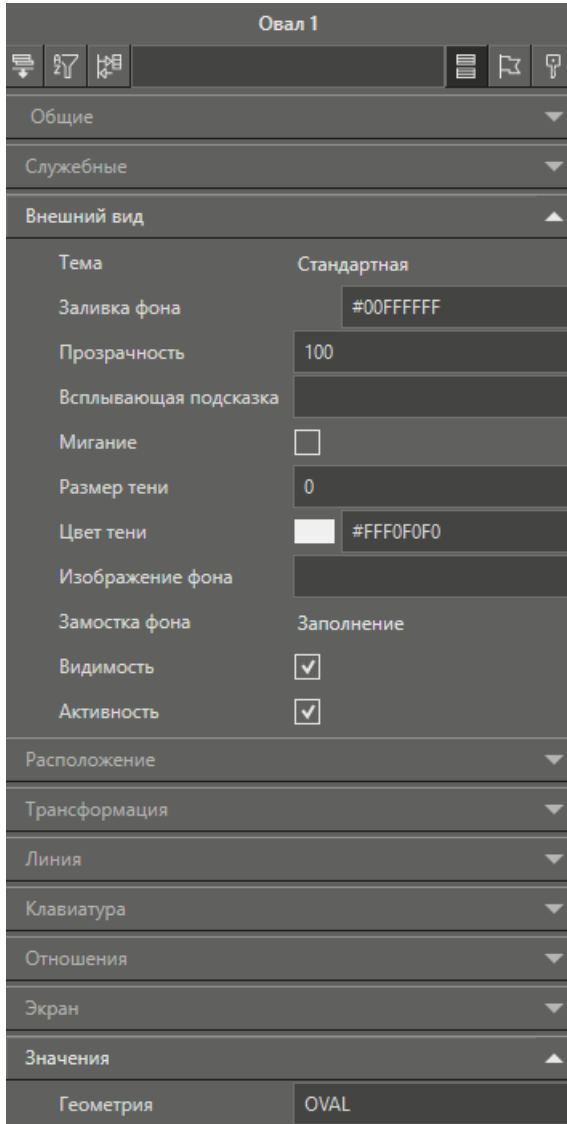
Данные элементы представляют собой SVG-фигуры, в настройках которых по умолчанию задана их геометрическая форма (тип фигуры). Чаще всего используются в окнах в виде индикаторов (т.е. меняют свой внешний вид в зависимости от значений параметров).

В палитре редактора HMI элементы Овал, Прямоугольник, Треугольник находятся в категории Прimitives.

Вид элементов в палитре:

Название	Значение свойства Геометрия	Вид в палитре	Вид в окне
Овал	OVAL		
Прямоугольник	RECT		
Треугольник	M0,0.5 L1,0 L1,1Z		

Вид панели свойств элементов при нажатой и отжатой кнопке :

Простой режим	Полный режим
	

Описание основных свойств графических элементов Овал, Прямоугольник, Треугольник:

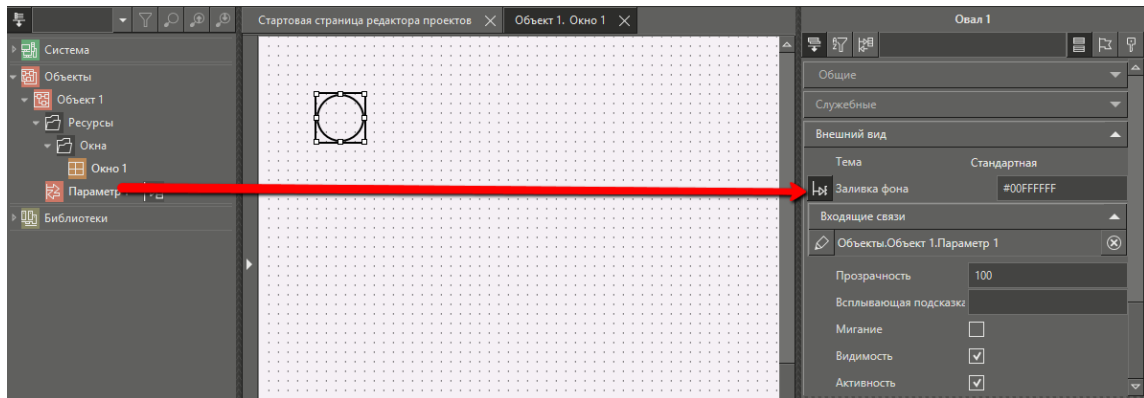
Название	Ссылка на описание
Категория Значение	
Геометрия	Задается тип фигуры. Настройку по умолчанию можно изменить, при этом соответственно изменится и тип фигуры.

Описание других свойств элементов смотрите в разделе НМІ.Свойства

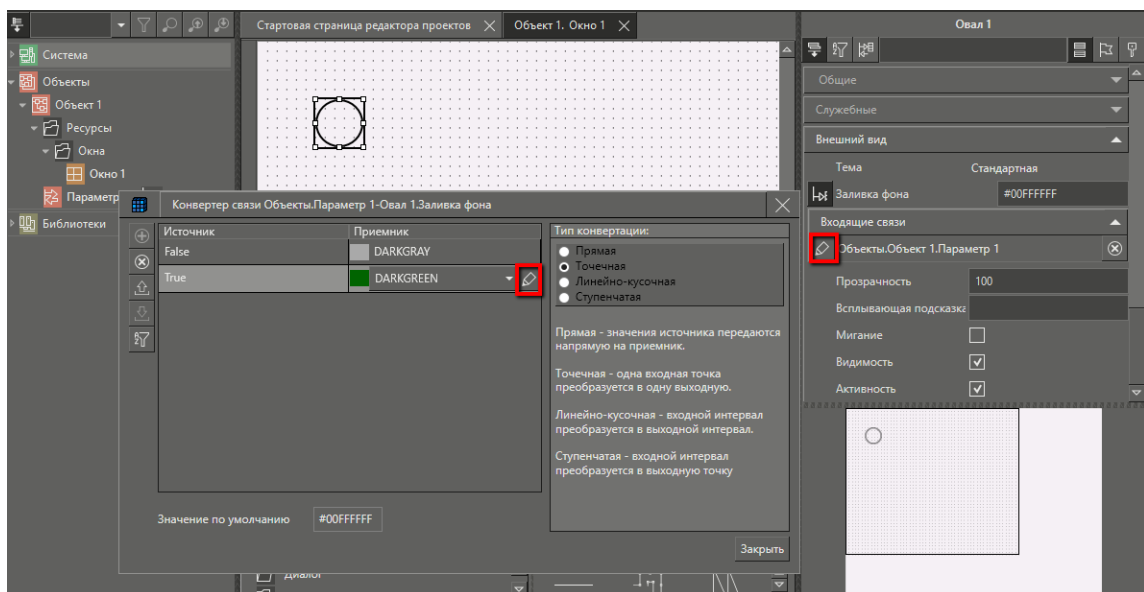
Рекомендации

Рассмотрим основные принципы работы с элементами на примере Овала

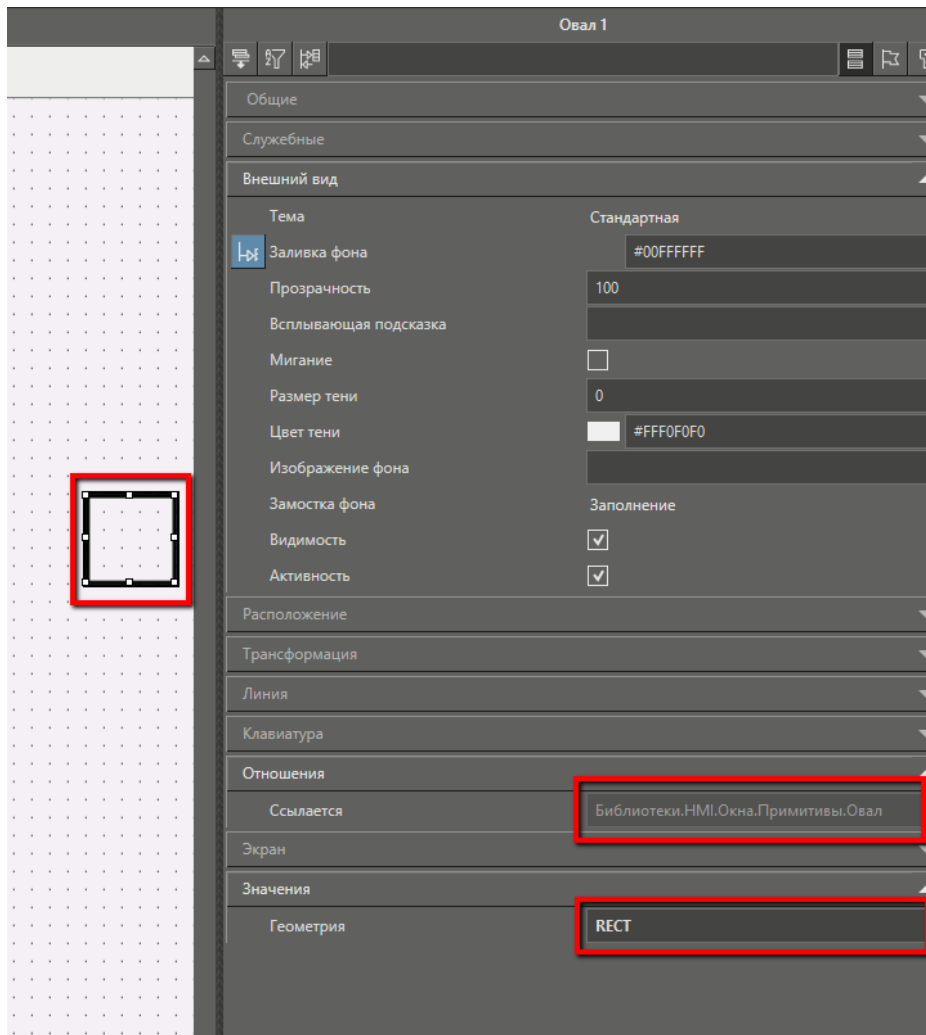
- Данный элемент часто используется для индикации состояния параметра типа BOOL. Например, если параметр имеет значение TRUE, то индикатор должен быть зеленым, а если False - серым. В этом случае параметр проекта необходимо перетащить в свойство Заливка фона.



А затем настроить конвертер значений:



- Если по требованиям заказчика сначала было необходимо отображать индикаторы круглыми, а затем требования изменились и стало необходимо использовать квадраты, то можно в свойстве Геометрия прописать другой тип фигуры, при этом все ранее сделанные настройки и связи сохранятся:



8.3.4.6. КРЕСТ

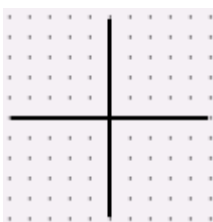
Данный элемент представляет собой SVG-фигуру, в настройках которой, по умолчанию, задано пересечение двух линий под прямым углом: M50,0L50,100M0,50L100,50.

В палитре редактора HMI элемент Крест находится в категории Примитивы.

Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:



Свойства элемента Крест идентичны свойствам элемента Овал.

8.3.4.7. СТРЕЛКА

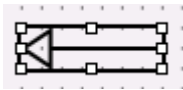
Данный элемент представляет собой SVG-фигуру, в настройках которой по умолчанию задан рисунок стрелки, указывающей влево: M0,0.5 L1,0 L1,1 L 0, 0.5 M1, 0.5 L5, 0.5.

В палитре редактора HMI элемент Стрелка находится в категории Примитивы.

Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:



Свойства элемента Стрелка идентичны свойствам элемента Овал.

8.3.4.8. КАРТИНКА

Данный элемент предназначен для отображения медиа-ресурса (изображение из файла, в т.ч. анимированное GIF- или SVG-изображение).


В палитре редактора HMI элемент Картинка находится в категории Примитивы.

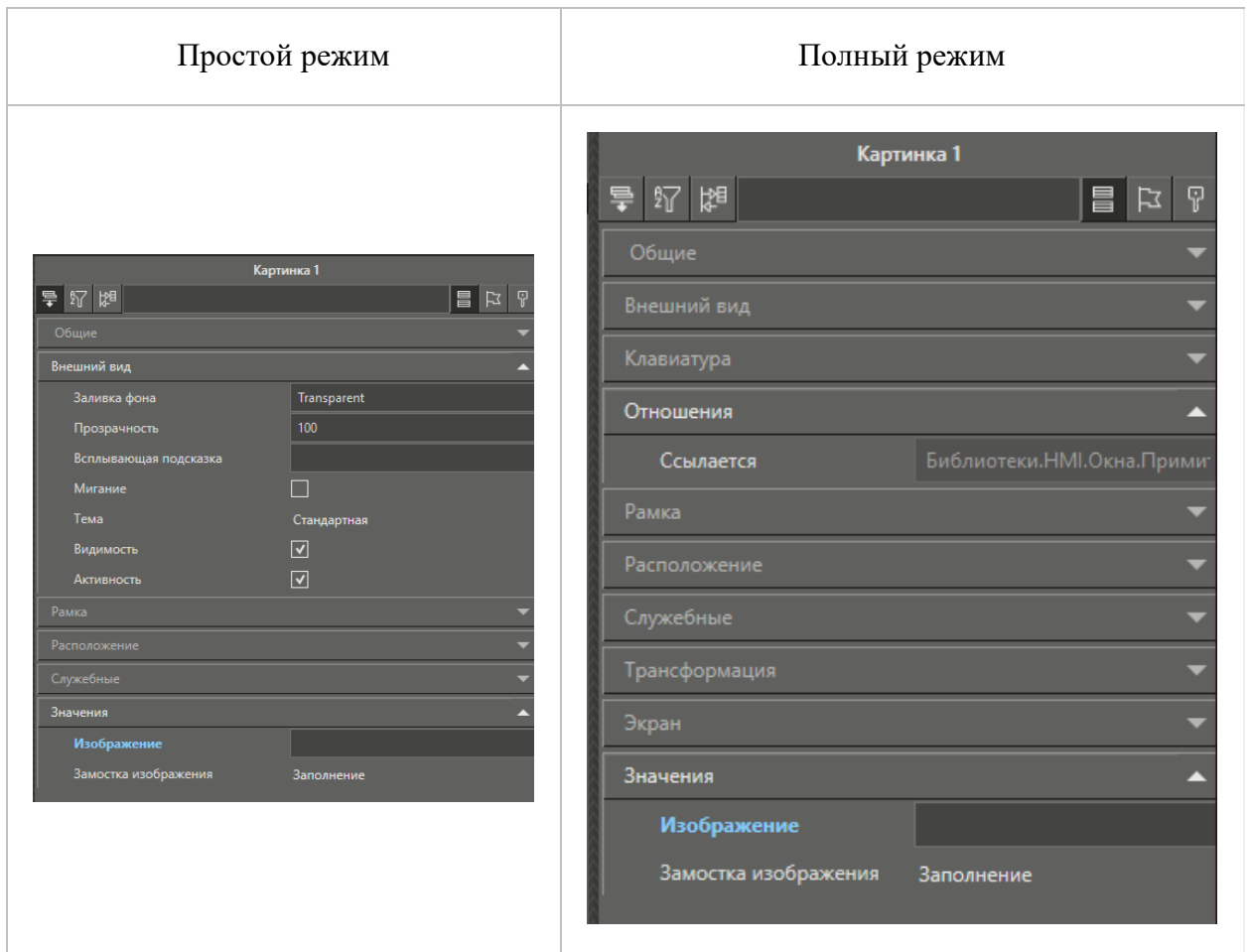
Вид элемента в палитре:




Вид элемента после добавления в рабочую область редактора HMI:

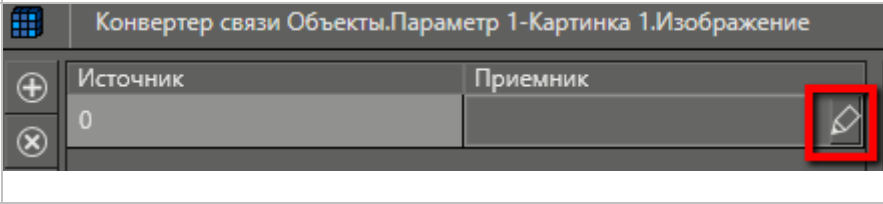


Вид панели свойств элемента при нажатой и отжатой кнопке :



Описание основных свойств графического элемента Картинка:

Название	Ссылка на описание
Категория Значения	
Изображение	<p>Определяет изображение элемента, которое будет отображаться в клиенте визуализации. При нажатии на кнопку , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора медиа-ресурса. В качестве изображения элемента может быть использован произвольный графический файл.</p> <p>Если требуется, чтобы в клиенте визуализации отображались разные изображения, то следует связать свойство с параметром проекта, и настроить конвертер значений, т.е. выбрать необходимый тип конвертации, и выбрать для каждой опорной точки свой медиа-ресурс:</p>

Название	Ссылка на описание
	
Замостка изображения	<p>Задаёт тип использования изображения элемента. Используется совместно со свойством Изображение. Может принимать значения:</p> <ul style="list-style-type: none"> • Заполнение - размер изображения будет изменен по границам элемента; • Мозаика - размер изображения будет исходным, и повторен столько раз, сколько необходимо чтобы замостить всю площадь элемента; • Центр - размер изображения будет исходным, изображение будет центрировано; • Нет - изображение использоваться не будет.

Описание других свойств элемента смотрите в разделе НМІ.Свойства.

8.3.4.9. ГРАФИЧЕСКИЙ ЭЛЕМЕНТ ВИДЕО

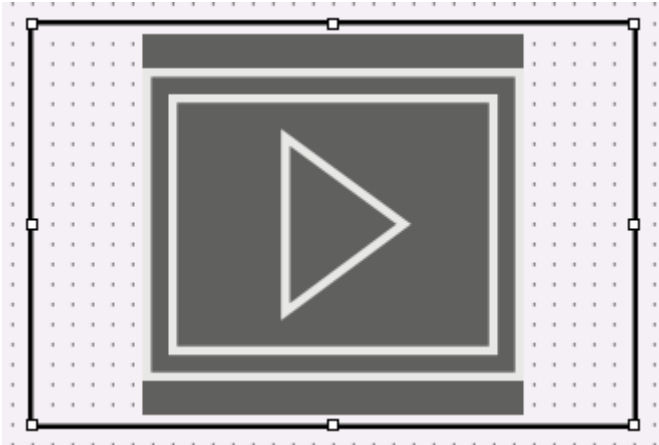
Данный элемент предназначен для проигрывания медиа-ресурса (видео из файла). Доступный формат файла: WebM.


В палитре редактора НМІ элемент Видео находится в категории Прimitives.

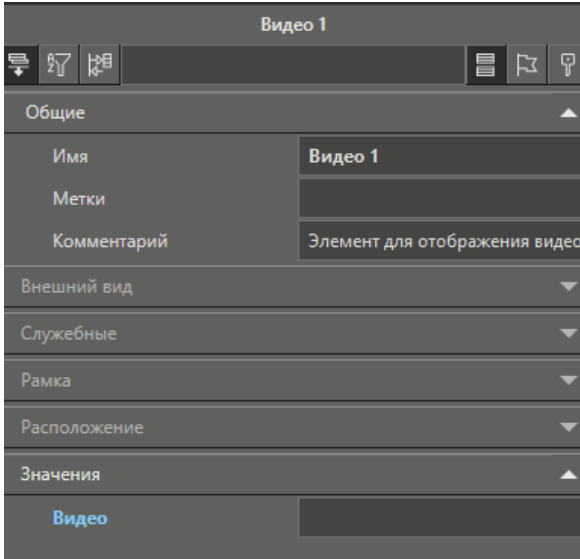
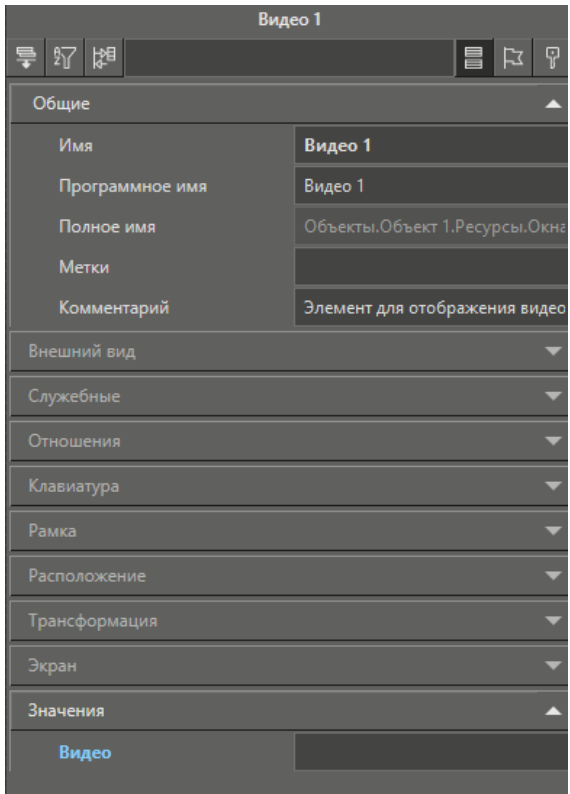
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора НМІ:


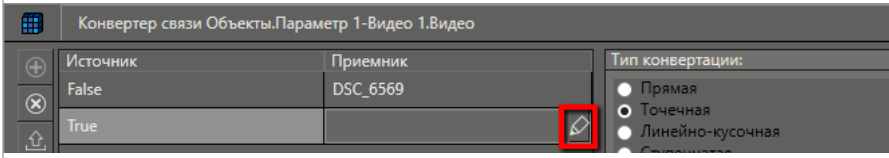


Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
	

Описание основных свойств графического элемента Видео:

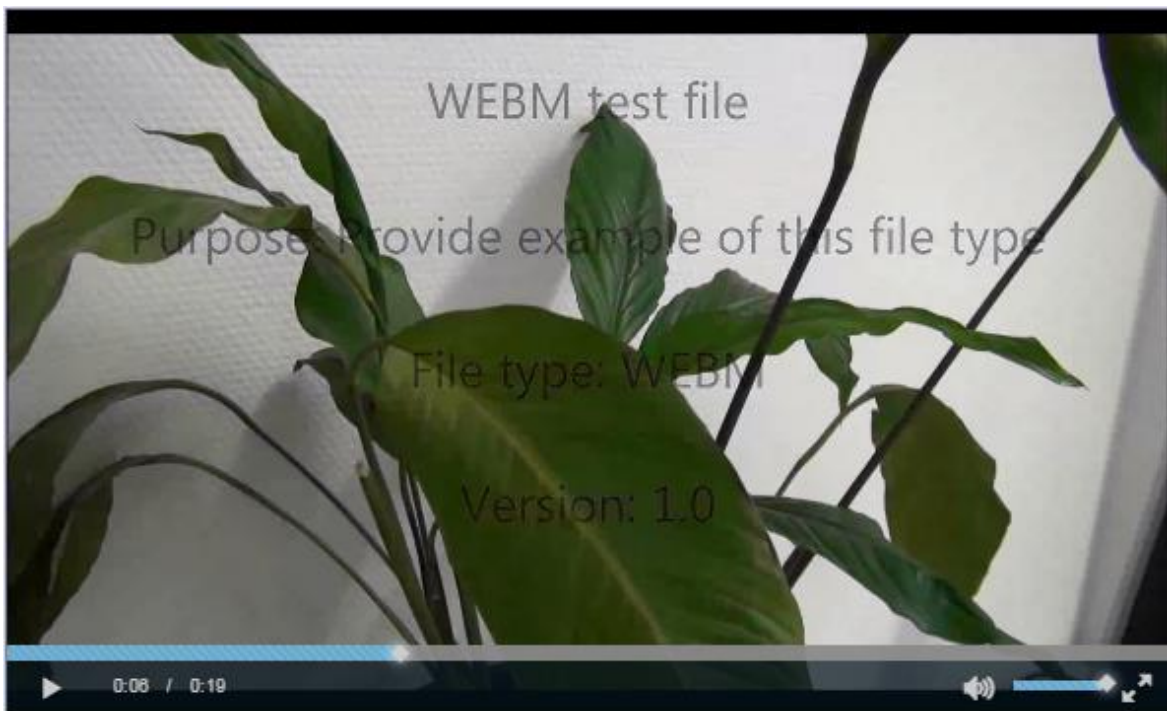
Название	Ссылка на описание
Категория Значения	

Название	Ссылка на описание
Видео	<p>Определяет видео, которое будет проигрываться в клиенте визуализации. При нажатии на кнопку , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора медиа-ресурса. В качестве видео может быть использован произвольный видео-файл (например, mp4).</p> <p>Если требуется, чтобы в клиенте визуализации проигрывались разные видео, то следует связать свойство с параметром проекта, т.е. выбрать необходимый тип конвертации, и выбрать для каждой опорной точки свой медиа-ресурс:</p> 

Описание других свойств элемента смотрите в разделе НМІ.Свойства.

Работа в режиме исполнения

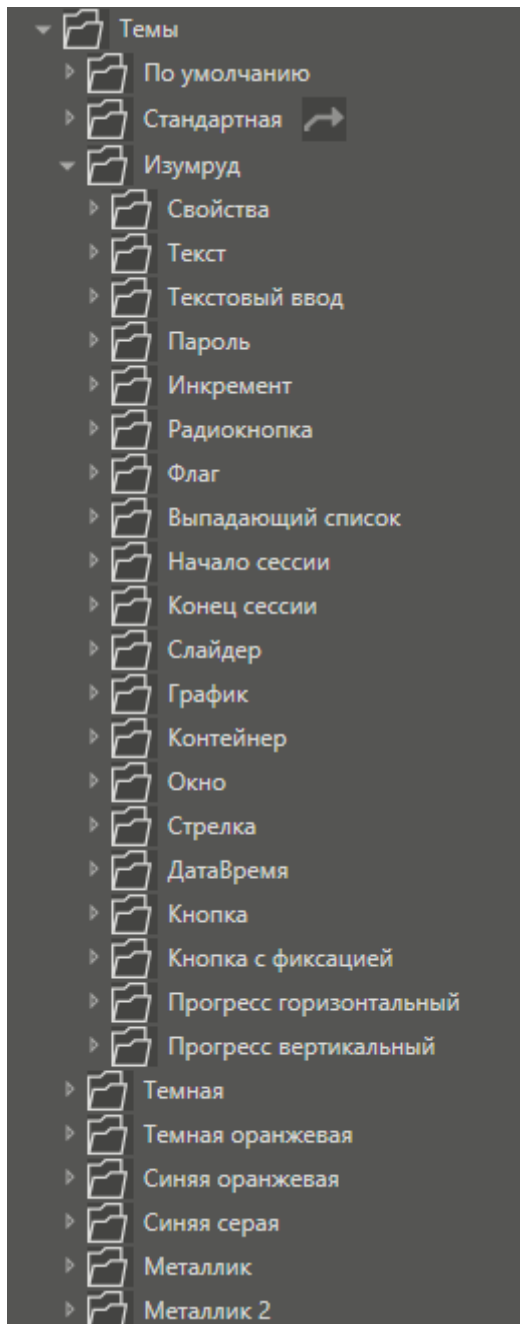
Вид элемента в браузере:



В режиме исполнения можно остановить и перемотать видео, управлять звуком и включить полноэкранный режим.

8.3.5. ТЕМЫ БИБЛИОТЕКИ HMI

Группа Темы библиотеки HMI содержит несколько тем. Каждая Тема содержит те же графические элементы, что и группа Окна библиотеки HMI.

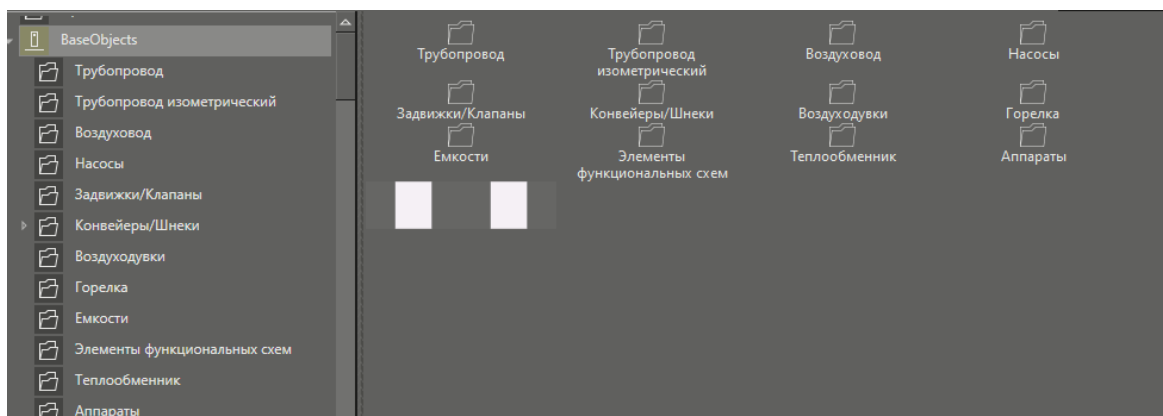


Изменение свойства графического элемента в применяемой теме (в библиотеке) автоматически изменяет данное свойство на мнемосхеме, если это свойство не редактировалось на мнемосхеме вручную.

8.3.6. КАТЕГОРИЯ BASEOBJECTS

Категория BaseObjects палитры редактора HMI содержит элементы, часто используемые для создания графического отображения технологических процессов в клиенте визуализации. Элементы могут быть использованы как в статическом виде (т.е. свойства элементов не изменяются в клиенте визуализации в зависимости от каких-либо параметров проекта), так и в динамическом (когда все свойства элементов могут быть динамизированы, т.е. при наступлении различных событий, связанных с этими элементами, могут выполняться различные действия).

В палитре редактора HMI эта категория имеет вид:



После добавления элемента из палитры могут быть установлены входящие/исходящие связи между свойствами элементов и параметрами проекта.

Категория BaseObjects имеет сложную структуру, т.к. содержит внутри себя не только отдельные элементы, но и вложенные категории:

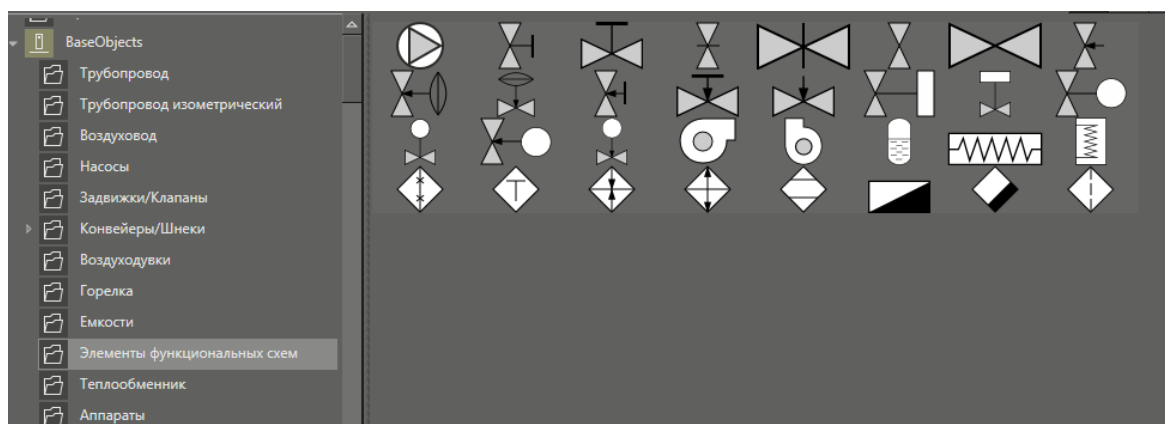
- BaseObjects. Трубопровод
- BaseObjects. Трубопровод изометрический
- BaseObjects. Воздуховод
- BaseObjects. Насосы
- BaseObjects. Задвижки/Клапаны
- BaseObjects. Конвейеры/Шнеки
- BaseObjects. Воздуходувки
- BaseObjects. Горелка
- BaseObjects. Емкости
- BaseObjects. Элементы функциональных схем
- BaseObjects. Теплообменник
- BaseObjects. Аппараты
- Индикатор
- Задатчик

- Метки

8.3.6.1. BASEOBJECTS. ЭЛЕМЕНТЫ ФУНКЦИОНАЛЬНЫХ СХЕМ

Категория Элементы функциональных схем палитры редактора НМІ содержит элементы, внешний вид которых соответствует условным графическим изображениям технологического оборудования на функциональных схемах автоматизации.

В палитре редактора НМІ эта категория имеет вид:



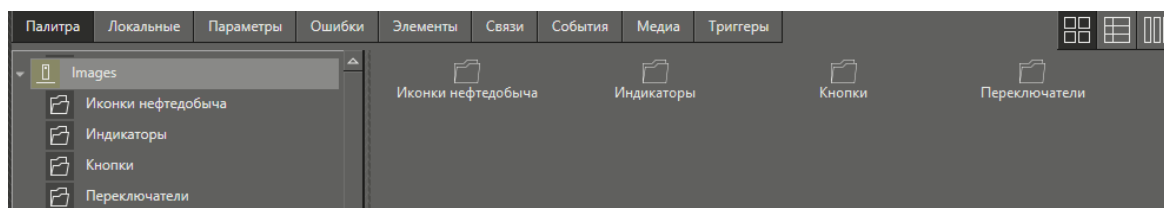
После добавления элемента из палитры могут быть установлены входящие/исходящие связи между свойствами элементов и параметрами проекта.

Категория Элементы функциональных схем включает в себя следующие элементы:

8.3.7. КАТЕГОРИЯ IMAGES

Категория Images палитры редактора НМІ содержит дополнительные вспомогательные элементы, которые могут быть использованы как в статическом виде (т.е. свойства элементов не изменяются в клиенте визуализации в зависимости от каких-либо параметров проекта), так и в динамическом (когда все свойства элементов могут быть динамизированы, т.е. при наступлении различных событий, связанных с этими элементами, могут выполняться различные действия).

В палитре редактора НМІ эта категория имеет вид:



После добавления элемента из палитры могут быть установлены входящие/исходящие связи между свойствами элементов и параметрами проекта.

Категория Images имеет сложную структуру, т.к. содержит внутри себя не только отдельные элементы, но и вложенные категории:

- Images. Иконки нефтедобыча
- Images. Индикаторы
- Images. Кнопки
- Images. Переключатели

8.4. НМІ. СВОЙСТВА

Свойство Формат значений

Свойство Маска ввода

Категория свойств Экран

Категория свойств Рамка/Линия

Категория свойств Клавиатура

Категория свойств Внешний вид

Категория свойств Расположение

Категория свойств Текст

8.4.1. СВОЙСТВО ФОРМАТ ЗНАЧЕНИЙ

Данное свойство имеет тип STRING и задает формат отображения значений при использовании конвертации и добавления в единиц измерения. Рассмотрим числовые и временные форматы.

Числовые форматы

Используются для отображения числовых значений параметров стандартные числовые форматы. Строка стандартных числовых форматов использует формат Axx, где:

A — это один буквенный символ, который называют описателем формата.

xx — это необязательное целое число, которое называют описателем точности. Спецификатор точности находится в диапазоне от 0 до 99 и влияет на число цифр в результате. Описатель точности может повлиять на округления числа к ближайшему значению

Значение	Описание
F или f	Используется для значений с фиксированной запятой. Описатель точности влияет на количество знаков после запятой. Например, число 25.560008 может быть представлено при значении свойства F0, как 26, а при F3 - 25.560
D или d	Используется для целочисленных значений. Минимальное количество знаков в выходной строке задается спецификатором точности. Недостающие знаки в строке заменяются нулями. Если описатель точности не задан, по умолчанию используется минимальное значение, позволяющее представить целое число без нулей в начале.
C или c	Используется для обозначения валюты. К значению добавляется региональное обозначение валюты. Описатель точности влияет на количество знаков после запятой
N или n	Преобразует число в строку вида -d,ddd,ddd.ddd... , где знак - при необходимости представляет знак отрицательного числа, знак d означает цифру (0-9), знак , — разделитель групп, а знак . — разделитель целой и дробной части. Требуемое число знаков дробной части задается спецификатором точности.
X или x	Используется для отображение чисел в шестнадцатеричном формате (X-буквенная часть будет указана заглавными буквами, x - прописными). Используется только для целочисленных типов. Описатель точности влияет на минимальное число цифр в результирующей строке. Недостающие знаки в строке заменяются нулями. Например, число 6787 может быть представлено при значении свойства X7, как 0001A83, а при x3 - 1a83.
E или e	Преобразует число в экспоненциальный вид. Описатель точности влияет на количество знаков после запятой.
P или p	Используется для вывода процентов. Исходное число будет умножено на 100 и будет добавлен символ %. Описатель точности влияет на количество знаков после запятой. По умолчанию указывается 6 знаков после запятой.

Для задания формата отображения переменных имеющих типа DT, TIME, DATA можно использовать стандартные форматы даты и времени, либо настраиваемые.

Стандартные форматы даты и времени

Строка стандартного формата даты и времени использует один символ в качестве описателя формата для определения текстового представления:

Значение	Описание	Ожидаемый результат
d	Короткий шаблон даты.	15/06/2009
D	Полный шаблон даты.	15 июня 2009 г.
f	Полный шаблон даты и времени (короткий шаблон времени).	15 июня 2009 г. 13:45
F	Полный шаблон даты и времени (полный шаблон времени).	15 июня 2009 г. 13:45:30
g	Общий шаблон даты и времени (короткий шаблон времени). Представляет сочетание краткого формата даты (d) и краткого формата времени (t), разделенных пробелом.	15/06/2009 13:45
G	Общий шаблон даты и времени (полный шаблон времени). Представляет сочетание краткого формата даты (d) и полного формата времени (T), разделенных пробелом.	15/06/2009 13:45:30
M, m	Шаблон дней месяца.	15 июня
O, o	Шаблон обратного преобразования даты и времени. Представляет строку настраиваемого формата даты и времени, используя шаблон, который сохраняет данные о часовом поясе и возвращает строковое значение, соответствующее стандарту ISO 8601	2009-06- 15T13:45:30.000 0000-07:00
R, r	Шаблон RFC1123.	Mon, 15 Jun 2009 20:45:30 GMT

Значение	Описание	Ожидаемый результат
s	Сортируемый шаблон времени и даты.	2009-06-15T13:45:30
t	Короткий шаблон времени.	13:45
T	Полный шаблон времени.	13:45:30
u	Универсальный сортируемый шаблон времени и даты.	2009-06-15 13:45:30Z
U	Универсальный полный шаблон даты и времени.	Пн, 15 июня 2009 г. 13:45:30
Y, y	Шаблон месяца года.	Июнь 2009 г.

Настраиваемые форматы даты и времени

Строка настраиваемого формата состоит из одного или нескольких настраиваемых описателей формата даты и времени. Каждому значению даты или времени соответствует свое обозначение. В качестве разделителей можно использовать символы (пробел, дефис, слеш и т.п.). Любая строка, не являющаяся строкой стандартного формата даты и времени, воспринимается как строка настраиваемого формата даты и времени. Примеры формирования настраиваемого формата времени:

Значение	Описание	Ожидаемый результат
dd-MM-yyuu	Шаблон даты в виде день-месяц-год	15-06-2009
HH:mm:ss.fff	Шаблон времени в виде часы:минуты:секунды.миллисекунды	13:45:30.000
dd MMMM yyuu HH:mm	Шаблон даты и времени, где месяц задается полным именем.	15 июня 2009 13:45
yyuu,	Шаблон задания года: год в виде четырехзначного числа;	2009

Значение	Описание	Ожидаемый результат
ууу,	год в виде как минимум трех цифр;	2009
уу,	год, в диапазоне от 00 до 99;	09
у	год, в диапазоне от 0 до 99.	9
d/M/yy HH:mm	Краткий шаблон даты и времени.	15/6/09 13:45
ddd	Краткий шаблон записи дня недели	Пн
dd MMMM	Шаблон задания дня и полного наименования месяца	15 июня
h:m:s.f	Шаблон задания времени, где h - Час в 12-часовом формате от 1 до 12, m - минуты, в диапазоне от 0 до 59, s - секунды, в диапазоне от 0 до 59, f - десятые доли секунды в значении даты и времени.	1:45:30.0
h	Формата задания часа: час в 12-часовом формате от 1 до 12;	1
hh	час в 12-часовом формате от 01 до 12;	01
HH	час в 24-часовом формате от 0 до 23;	13
H	час в 24-часовом формате от 00 до 23.	13
M	Формат задания месяца: месяц, в диапазоне от 1 до 12;	
MM	месяц, в диапазоне от 01 до 12;	
MMM	сокращенное название месяца;	
MMMM	полное название месяца;	
d	Формат задания дня: число в диапазоне от 1 до 31;	15
dd	число в диапазоне от 01 до 31;	15
ddd	число в диапазоне от 01 до 31;	Пн

Значение	Описание	Ожидаемый результат
dddd	сокращенное название дня недели, которому соответствует число; полное название дня недели, которому соответствует число.	Понедельник
Свойство не задано	Используется формат вывода даты и времени по умолчанию	

Добавление единицы измерения

Если необходимо, чтобы рядом с числом находилась единица измерения, то свойство формат значения можно указать в виде {обозначение формата} единицы измерения. Например, если будет указано значение {f2} м3, то в клиенте визуализации получим значение:

12.27 м3

Единицы измерения будут устанавливаться автоматически при использовании шкал.

8.4.2. СВОЙСТВО МАСКА ВВОДА

Данное свойство имеет тип STRING и указывает допустимые символы, вводимые в клиенте визуализации, например, при работе с элементом Текстовый ввод.

Маска ввода с обязательным числом символов

Общий вид маски: (999)(aa)(**), где цифра 9 обозначает, что на это место можно ввести только цифру, латинская буква а – только букву, символ * – букву или цифру.

Пример применения – поле для ввода телефона. Следующая маска позволит ввести только 7 цифр: (999)(99)(99):

Маска ввода с необязательным числом чисел

Для ввода значений с неопределенным заранее количеством цифр используется символ 0.

Если часть маски ввода содержит 0, то применяется следующий алгоритм подстановки -

В случае, если между необязательными символами оказываются не введенные, при обработке введенного значения пробелы будут удаляться.

Задача	Значения свойства	Вид поля ввода	Ввод в режиме исполнения	Результат
Допустим, требуется ограничить ввод чисел диапазоном от 0.0 до 999.99.	009.90	___._	__0.0_	0.0
			10.1	10.1
			_10.99	10.99
			1_1.1_	11.1
			1__.1_	т.к. не хватает обязательной цифры перед точкой, то введенное значение будет сброшено
Пусть можно вводить любое число от 0.001 до 99999.999	00000.000	_____.__	1____.____1	1.1
			____.____	Не будет являться числом (будут возникать ошибки при передаче значения в числовые параметры)
			_____.1_	0.1
			__1.____	1.0
			1_2_3_.4_	123.4
Пусть необходимо вводить и цифры и буквы и символы	9a90/90	_____/__	0x_1/11	Значение будет сброшено, т.к. не хватает обязательной цифры после буквы
			0x1_/_2	Значение будет сброшено, не хватает

Задача	Значения свойства	Вид поля ввода	Ввод в режиме исполнения	Результат
				обязательной цифры после символа /
			0d23/23	0d23/23
			0k2_/1_	0k2/1

Сложная маска ввода

Если разбить маску на отдельные фрагменты на серии из 09a*, то в рамках одного фрагмента нельзя задать 0 и * одновременно, так как это приводит к неоднозначному разбору. В этом случае выдается ошибка режиме исполнения.

Если фрагмент содержит 0, то применяется следующий алгоритм подстановки -

Идет подстановка букв/цифр слева направо. На каждой позиции -

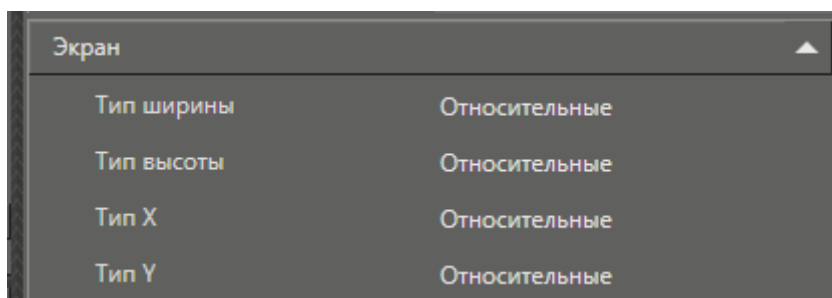
- если в маске a, подставляется буква
- если в маске 9, подставляется цифра
- если в маске 0, то цифра подставляется при соблюдении одного из условий -
 - оставшееся число цифр в маске меньше или равно чем оставшееся число цифр в тексте
 - на предыдущей позиции была заполнена цифра и оставшееся число 9 в маске меньше чем оставшееся число цифр в тексте

Например, если маска 090, то при вводе принимаются оба значения _45 и 45_, но при получении 45 по входящей связи будет выдано _45

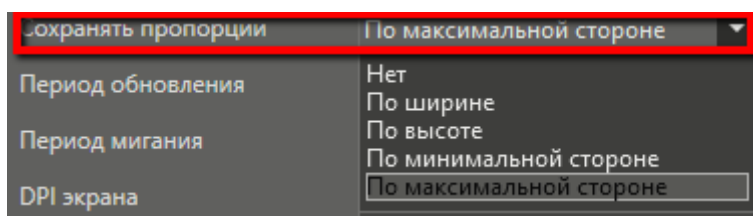
В случае маски 9009 значение 456 будет отображено как 45_6

8.4.3. КАТЕГОРИЯ СВОЙСТВ ЭКРАН

Категория свойств Экран определяет координаты и размер элемента в окне клиента визуализации и его поведение при масштабировании окна браузера. Помимо настроек, сделанных непосредственно в панели свойств элемента, на координаты и размер оказывают влияние настройки Шаблона экрана и родительского окна.



Значения свойств в данной категории могут иметь тип Относительные, либо Абсолютные. Если задан тип Относительные, то местоположение верхнего левого угла рамки, описывающей элемент, и размер элемента будут зависеть от размера окна браузера, т.е. элементы будут автоматически уменьшаться или увеличиваться. Пропорции размеров элементов будут зависеть от настройки Шаблона экрана Сохранять пропорции:



Если задан тип Абсолютные, то размер элемента и положение верхнего левого угла рамки, описывающей элемент, изменяться не будет, а будут использоваться размеры и координаты, заданные в категории свойств Расположение.

Рассмотрим влияние настроек заданных в данной категории у различных элементов проекта.

Допустим, есть проект, в котором создано два окна. Окно 2, содержащее элемент Овал, и Окно 1, которое назначено стартовым для шаблона экрана узла и содержит Окно 2. Рассмотрим, к чему приведет изменение настроек Тип ширины и Тип высоты одновременно в категории Экран, у Шаблона экрана, у элементов Окно 2 и Овал.

№	Шаблон экрана	Окно 2	Овал	Результат
1	Относительные	Относительные	Относительные	Размер стартового окна, размер Окна 2 и размер Овала масштабируются в зависимости от размера окна клиента визуализации.
2	Относительные	Относительные	Абсолютные	Размер стартового окна и размер Окна 2 масштабируются. Размер Овала внутри Окна 2 свои размеры не меняет.

№	Шаблон экрана	Окно 2	Овал	Результат
3	Относительные	Абсолютные	Относительные	Размер стартового окна масштабируется, а размер Окна 2 и размеры всех его дочерних элементов не масштабируются.
4	Абсолютные	Относительные	Относительные	Размер стартового окна не масштабируется, а значит и размеры всех других окон тоже не масштабируются. Если стартовое окно не умещается в окне браузера, то появляются линейки прокрутки.

Рассмотрим зависимость при изменении свойств Тип X, Тип Y у различных элементов

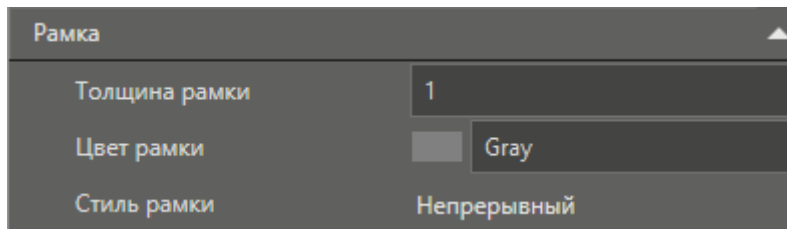
№	Шаблон экрана	Окно 2	Овал	Результат
1	Относительные	Относительные	Относительные	Положение стартового окна, положение Окна 2 и положение Овала масштабируются в зависимости от размера окна клиента визуализации.
2	Относительные	Относительные	Абсолютные	Независимо от размеров окна браузера, верхний левый угол Овала всегда удален от верхнего левого угла Окна 2.
3	Относительные	Абсолютные	Относительные	Верхний левый угол вложенного Окна 2 всегда находится на одинаковом расстоянии от верхнего левого угла стартового окна. Положение элементов внутри Окна 2 меняется пропорционально изменению размеров окна.
4	Абсолютные	Относительные	Относительные	Не используется

Важно! Данная настройка не влияет на всплывающие окна. Размер, местоположение и возможности масштабирования зависят от настроек, заданных в действии, которое приводит к открытию окна.

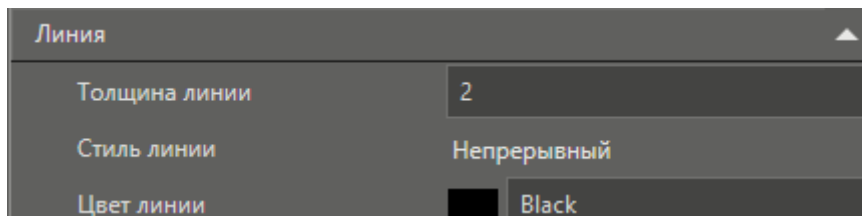
8.4.4. КАТЕГОРИЯ СВОЙСТВ РАМКА/ЛИНИЯ


Категория свойств Рамка определяет внешний вид прямоугольной рамки элемента. Категория свойств Линия определяет внешний вид линии, которая используется для создания элемента.

Вид категории в панели свойств:



либо:



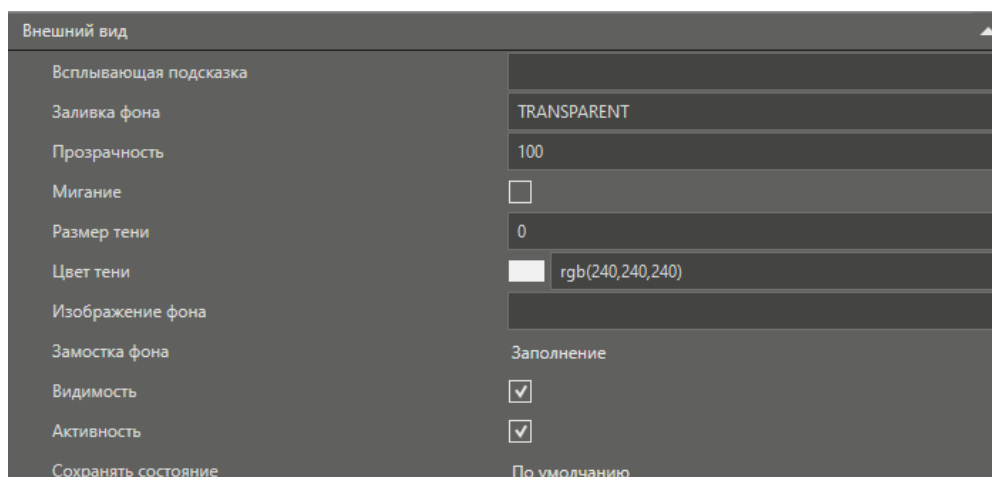
Название	Тип	Описание
Толщина рамки/линии	LREAL	Определяет толщину рамки элемента.
Цвет рамки/линии	HMI.SolidColor Type (STRING[15])	Определяет цвет рамки элемента. При нажатии на кнопку  , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.
Стиль рамки/линии	HMI.BorderStyleType (Перечисление)	Определяет стиль рамки. Выпадающий список содержит следующие возможные значения: Непрерывный, Пунктир, Точка, Нет.

8.4.5. КАТЕГОРИЯ СВОЙСТВ ВНЕШНИЙ ВИД


Данная категория свойств определяет внешний вид элемента при отображении его в окне клиента визуализации. В данном разделе перечислены свойства, которые находятся в панели свойств в категории Внешний вид и применимы для большинства графических



элементов. В данной категории у некоторых элементов могут содержаться дополнительные свойства, и описание таких свойств дано в разделе, посвященном конкретному элементу.


Вид категории в панели свойств:

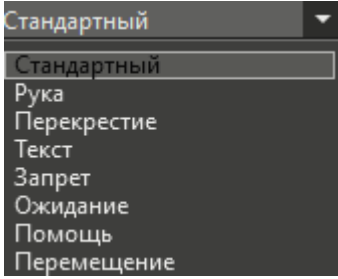


Описание:

Название	Описание
Замостка фона	<p>Задаёт тип использования медиа-ресурса в качестве фона графического элемента или фона окна. Используется совместно со свойством Изображение фона. Может принимать значения:</p> <ul style="list-style-type: none"> • Заполнение - размер фонового изображения будет изменен по границам элемента; • Мозаика - размер фонового изображения будет исходным, и повторен столько раз, сколько необходимо чтобы замостить всю площадь элемента; • Центр - размер фонового изображения будет исходным, изображение будет центрировано; • Нет - фоновое изображение использоваться не будет.
Заливка фона	<p>Определяет цвет фона элемента. При нажатии на кнопку , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.</p>
Прозрачность	<p>Задаёт степень прозрачности элемента. Значение изменяется в пределах от 0 до 100, где 0 - элемент полностью прозрачный, 100 -</p>

Название	Описание
	элемент полностью не прозрачный. Если элемент полностью прозрачный, то динамизация и Действия не исполняются.
Всплывающая подсказка	Данное свойство имеет тип <code>STRING</code> и задает текст, отображаемый во всплывающей подсказке элемента.
Мигание	Данное свойство имеет тип <code>BOOL</code> и может принимать значение <code>TRUE</code> (использовать мигание) или <code>False</code> (не использовать мигание). Период мигания задает свойство Период мигания в панели свойств Шаблона экрана.
Видимость	Данное свойство имеет тип <code>BOOL</code> и может принимать значение <code>TRUE</code> (элемент видим) или <code>False</code> (элемент невидим).
Активность	<p>Данное свойство имеет тип <code>BOOL</code>. Если свойство находится в состоянии <code>TRUE</code>, то свойства элементов в режиме исполнения принимают и отправляют значения (работает динамизация), а также выполняются Действия. Если свойство находится в состоянии <code>False</code>, то работает только входящая связь, а Действия и ввод значений исполняться не будут. Если свойство имеет значение <code>TRUE</code>, то элемент отображается серым цветом. На рисунке слева находится элемент, свойство которого принимает значение <code>TRUE</code>, а справа - <code>False</code>:</p>  <p>Данное свойство работает только в том случае, если установлена связь между этим свойством и каким-либо параметром. Если связь между свойством Активность и параметром проекта отсутствует, то элемент будет активным в клиенте визуализации независимо от состояния флага.</p>
Размер тени	Определяет величину тени элемента (тень элемента появляется снизу и справа).
Цвет тени	Определяет цвет тени элемента. При нажатии на кнопку  , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.

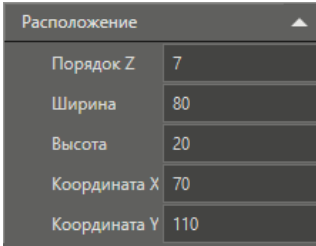
Название	Описание
Изображение фона	<p>Определяет фоновое изображение элемента. При нажатии на кнопку , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора медиа-ресурса. В качестве фона может быть использован произвольный графический файл.</p>
Радиус скругления	<p>Задаёт радиус скругления углов рамки элемента (в px).</p>
Сохранить состояние	<p>Определяет необходимость сохранения настроек, сделанных пользователем в режиме исполнения, после перезагрузки.</p> <p>Возможно следующие значения свойства:</p> <ul style="list-style-type: none"> • По умолчанию (Default) - наследуется от родительского окна. Если родительского окна нет, то означает, что сохранение включено <ul style="list-style-type: none"> • Да (Yes) - Сохранять • Нет (No) - Не сохранять <p>Настройка применяется если, у Шаблона экрана свойство Место хранения состояния окон имеет значение отличное от Не сохранять</p> <p>Настройка применяется для следующих элементов:</p> <p style="text-align: center;">Тренд</p> <ul style="list-style-type: none"> • Список перьев RT с их настройками <ul style="list-style-type: none"> • Диапазон по времени • Масштабы по отдельным осям Y <p style="text-align: center;">Журналы</p> <ul style="list-style-type: none"> • Настройки фильтров столбцов • Настройки сортировки столбцов • Список включенных фильтров • Выбранные в RT фильтры по источникам <p style="text-align: center;">Таблица данных</p> <ul style="list-style-type: none"> • Настройки фильтров столбцов • Настройки сортировки столбцов

Название	Описание
	<p style="text-align: center;">Справочник</p> <ul style="list-style-type: none"> • Настройки фильтров столбцов • Настройки сортировки столбцов <p style="text-align: center;">Легенда</p> <ul style="list-style-type: none"> • Ширина столбцов легенды
Тип линий сетки	<p>Определяет, как будут прорисовываться линии сетки таблиц элементов Справочник, Таблица данных, Легенда. Возможные варианты:</p> <ul style="list-style-type: none"> • Вертикальные (Vertical) - будут отображаться только вертикальные линии сетки • Горизонтальные (Horizontal) - будут отображаться только горизонтальные линии сетки • Обе (Both) - будут отображаться и горизонтальные и вертикальные линии сетки • Нет (None) - линии сетки отображаться не будут
Вид курсора	<p>Определяет вид курсора при наведении мыши на элемент в клиенте визуализации:</p> 

8.4.6. КАТЕГОРИЯ СВОЙСТВ РАСПОЛОЖЕНИЕ

Данная категория свойств определяет расположение элемента в родительском окне. В данном разделе перечислены свойства, которые находятся в панели свойств в категории свойств Расположении и применимы для большинства графических элементов. В данной категории у некоторых элементов могут содержаться дополнительные свойства, и описание таких свойств приведено в разделе, посвященном конкретному элементу.

Вид категории в панели свойств элемента, при нажатой и отжатой кнопке  :

Простой режим	Полный режим
	

Описание:

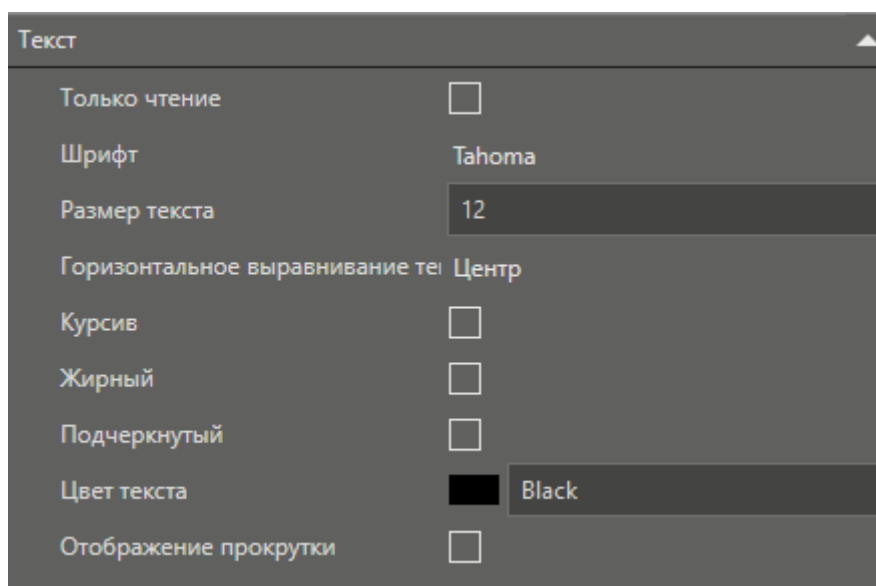
Название	Тип	Описание
Порядок Z	INT	Задаёт положение элемента по оси Z (0 – самое нижнее положение).
Ширина	REAL	Задаёт ширину элемента (в px) в редакторе HMI. У окна, открывающегося в клиенте визуализации, значение может быть изменено в зависимости от настроек, заданных в категории Экран.
Высота	REAL	Задаёт высоту элемента (в px) в редакторе HMI. У окна, открывающегося в клиенте визуализации, значение может быть изменено в зависимости от настроек, заданных в категории Экран.
Координата X	REAL	Задаёт абсциссу левого верхнего угла прямоугольника, ограничивающего графический элемент (в px) в редакторе HMI. Абсциссу 0 имеет левый верхний угол рабочей области редактора HMI. У окна, открывающегося в клиенте визуализации, значение может быть изменено в зависимости от настроек, заданных в категории Экран.
Координата Y	REAL	Задаёт ординату левого верхнего угла прямоугольника, ограничивающего графический элемент (в px) в рабочей области редактора HMI. Ординату 0 имеет левый верхний

Название	Тип	Описание
		угол рабочей области. У окна, открывающегося в клиенте визуализации, значение может быть изменено в зависимости от настроек, заданных в категории Экран.
Положение		Свойство используется только если элемент располагается внутри Панели док.
Горизонтальное выравнивание		Свойства используются только если элемент располагается внутри Стековой панели. Если свойство стековой панели Ориентация имеет значение Горизонтально, то используется Вертикальное выравнивание. Если свойство Ориентация имеет значение Вертикально, то используется Горизонтальное выравнивание.
Вертикальное выравнивание		

8.4.7. КАТЕГОРИЯ СВОЙСТВ ТЕКСТ


Категория свойств Текст имеется у элементов, в которых предполагается отображение полученного от параметра значения, а также ввод значений в клиенте визуализации с клавиатуры. Категория определяет внешний вид текста. В данном разделе перечислены свойства, которые находятся в панели свойств в категории Текст и применимы для большинства графических элементов. В данной категории у некоторых элементов могут содержаться дополнительные свойства. Описание таких свойств дано в разделе, посвященном конкретному элементу.

Вид категории в панели свойств:



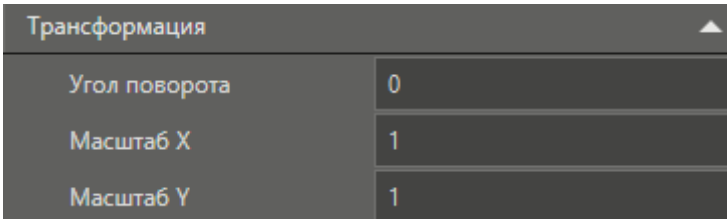
Описание:

Название	Описание
Только чтение	Это свойство имеет тип BOOL. Если свойство Только чтение имеет значение TRUE, то в элементе блокируется ввод значений.
Шрифт	Данное свойство имеет тип STRING и задает семейство шрифта, используемое в элементе.
Размер текста	Это свойство имеет тип BYTE и задает размер шрифта (в pt).
Горизонтальное выравнивание текста	Определяет положение текста по горизонтали относительно границ области ввода текста. Возможные значения - Лево, Право, Центр.
Вертикальное выравнивание текста	Определяет положение текста по вертикали относительно границ области ввода текста. Возможные значения - Вверх, Центр, Вниз.
Курсив	Данное свойство имеет тип BOOL и может принимать значение TRUE (использовать курсив) или False (использовать обычный шрифт).
Жирный	Данное свойство имеет тип BOOL и может принимать значение TRUE (использовать жирный шрифт) или False (использовать обычный шрифт)
Многострочность	Данное свойство имеет тип BOOL и может принимать значение TRUE (разрешить многострочность текста) или False (запретить многострочность текста). Если свойство Многострочность имеет значение TRUE, то в элементе выполняется автоматический перенос текста.
Подчеркнутый	Данное свойство имеет тип BOOL и может принимать значение TRUE (использовать подчеркнутый шрифт) или False (использовать обычный шрифт).

Название	Описание
Отображение прокрутки	Определяет будут ли появляться инструменты прокрутки. Если флаг установлен и в элементе необходимо отобразить многострочный текст, который превышает размеры элемента, то с правой стороны элемента появятся инструменты прокрутки (только для НМІ v2).
Цвет текста	Определяет цвет текста элемента. При нажатии на кнопку  , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.

8.4.8. КАТЕГОРИЯ СВОЙСТВ ТРАНСФОРМАЦИЯ

Категория свойств Трансформация определяет угол поворота и масштаб элемента относительно значений по умолчанию. Вид категории в панели свойств:



Трансформация	
Угол поворота	0
Масштаб X	1
Масштаб Y	1

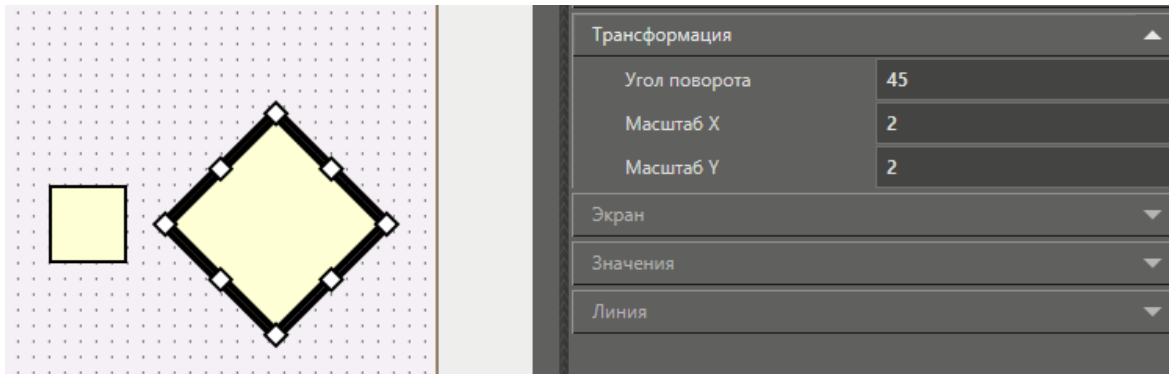
Описание:

Название	Описание
Угол поворота	Определяет угол поворота элемента относительно центра фигуры. Задается в градусах. Может принимать положительные и отрицательные значения.
Масштаб X	Позволяет масштабировать размер элемента по ширине. Может принимать положительные значения, целые и дробные.

Название	Описание
Масштаб Y	Позволяет масштабировать размер элемента по высоте. Может принимать положительные значения, целые и дробные.

Пример

На рисунке показаны два прямоугольника, у которых значения ширины и высоты в категории Расположение одинаковы. Но у левого прямоугольника значения свойств категории Трансформация установлены по умолчанию, а у правого так, как показано на рисунке.

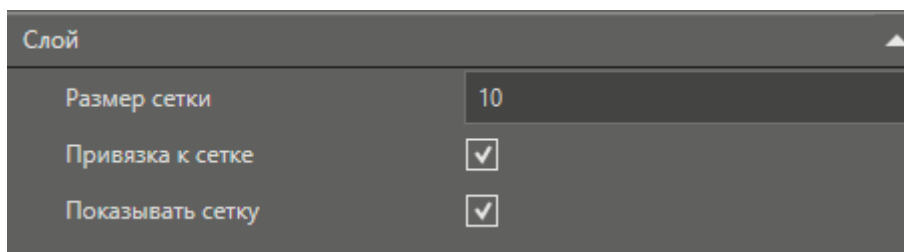


Можно видеть, что фактические размеры ширины и высоты у правого прямоугольника увеличены в два раза.

8.4.9. КАТЕГОРИЯ СВОЙСТВ СЛОЙ

Категория свойств Слой определяет видимость вспомогательной сетки, которая может быть использована во время разработки проекта для размещения и выравнивания элементов. Сетка видна только в среде разработки.

Вид категории в панели свойств:



Описание:

Название	Описание
Размер сетки	Задаёт размер ячейки сетки (в px).
Привязка к сетке	Если флаг установлен, то верхний левый угол элементов окна будет привязан к узлам сетки. При изменении размеров элементов мышью ширина и высота также будет зависеть от шага сетки
Показывать сетку	Определяет видимость сетки. Если флаг установлен, то сетка будет видна

Управление сеткой также может осуществляться при помощи панели инструментов.

8.5. РЕКОМЕНДАЦИИ ПО СОЗДАНИЮ ОКОН

Название раздела	Описание
Создание произвольных фигур	Описаны основные команды при для создания фигур в SVG.
Стартовое окно	Описано какое окно будет открываться по умолчанию в клиенте визуализации при старте режима исполнения.
Кеширование окон	Описано как ускорить открытие окон в режиме исполнения.
Особенности создания окон в библиотеке	Описаны особенности работы с параметрами при создании типов окон.
Особенности работы с элементами Кнопка и Кнопка с фиксацией	Описаны различные задачи, иллюстрирующие когда и как следует использовать данные графические элементы.
Массивы структур в окнах	Описан порядок работы с элементами редактора HMI, созданными для работы с массивами структур.

Название раздела	Описание
Создание пользовательской библиотеки примитивов	Порядок работы с библиотеками MasterSCADA 4D, а также со свойствами элементов при создании своих графических элементов.
Работа с веб-камерой	Описан наиболее распространенный способ работы с веб-камерой.

8.5.1. СОЗДАНИЕ ПРОИЗВОЛЬНЫХ ФИГУР

С использованием элементов Граф, Ломаная линия, Полигон можно создать произвольную фигуру, установив соответствующее значение в свойстве Геометрия. Рассмотрим основные правила работы. Это свойство имеет тип данных STRING и задает одну или несколько геометрических фигур.

Фигура описывается с помощью выражения следующего формата:

M X0,Y0 [<line1>] X1,Y1 .. [<lineN>] XN,YN[<format>]

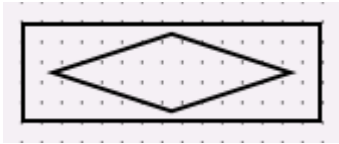
- M – разделитель фигур в строке;
- X_i, Y_i – координаты вершин фигуры. Координаты задаются относительно левого верхнего угла прямоугольника, ограничивающего элемент. Координаты задаются для размеров прямоугольника 160*160 px; фигуры строятся по этим координатам, а затем вписываются в заданные размеры элемента (масштабируются с сохранением пропорций между фигурами);
- line $_i$ – вид линии между вершинами [X_{i-1}, Y_{i-1}] и [X_i, Y_i];
- L – прямая (необязательный параметр, используется по умолчанию);
- format – формат задания координат, необязательный параметр:
 - не указан – в px;
 - Z – от 0 до 1, 1 соответствует 160 px.

Описания фигур в одной строке должны иметь один и тот же формат задания координат.

Например, две следующие строки задают одну и ту же комбинацию фигур, показанную на рисунке:

M 0,0 L 160,0 L 160,160 L 0,160 L 0,0 M 16,80 L 80,16 L 144,80 L 80,144 L 16,80

M 0,0 L 1,0 L 1,1 L 0,1 L 0,0Z M 0.1,0.5 L 0.5,0.1 L 0.9,0.5 L 0.5,0.9 L 0.1,0.5Z



Существуют следующие зарезервированные слова, заменяющие описание фигуры:

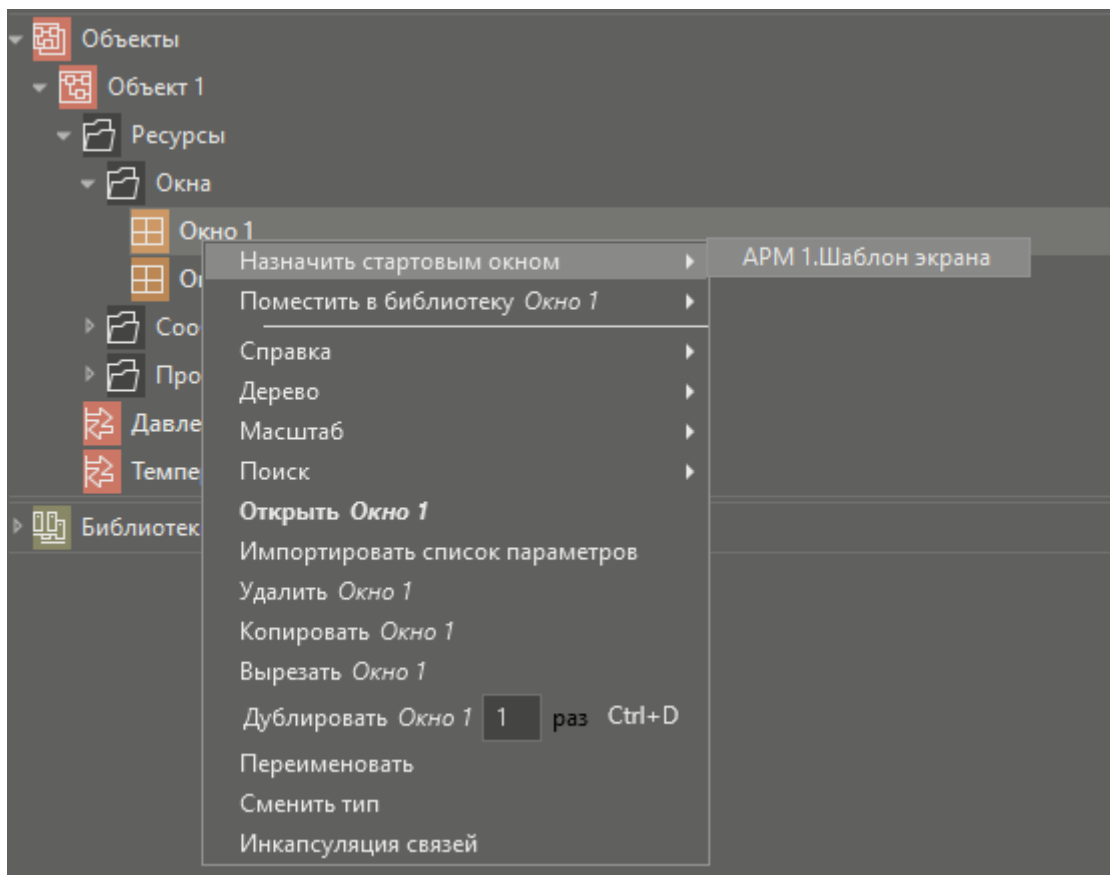
- RECT – прямоугольник;
- OVAL – овал.

В строке, содержащей зарезервированное слово, нельзя использовать никакие другие описания фигур, в т.ч. другие зарезервированные слова.

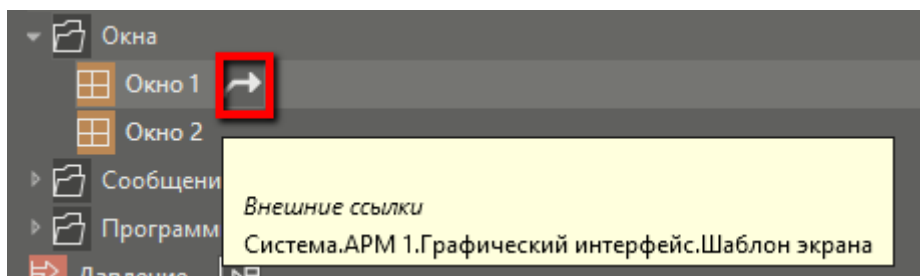
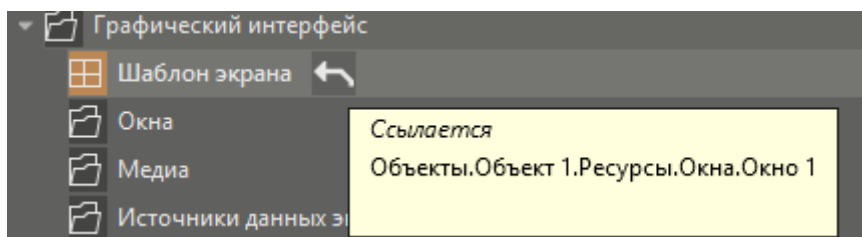
Заливка фигур при наложении вычисляется методом XOR.

8.5.2. СТАРТОВОЕ ОКНО

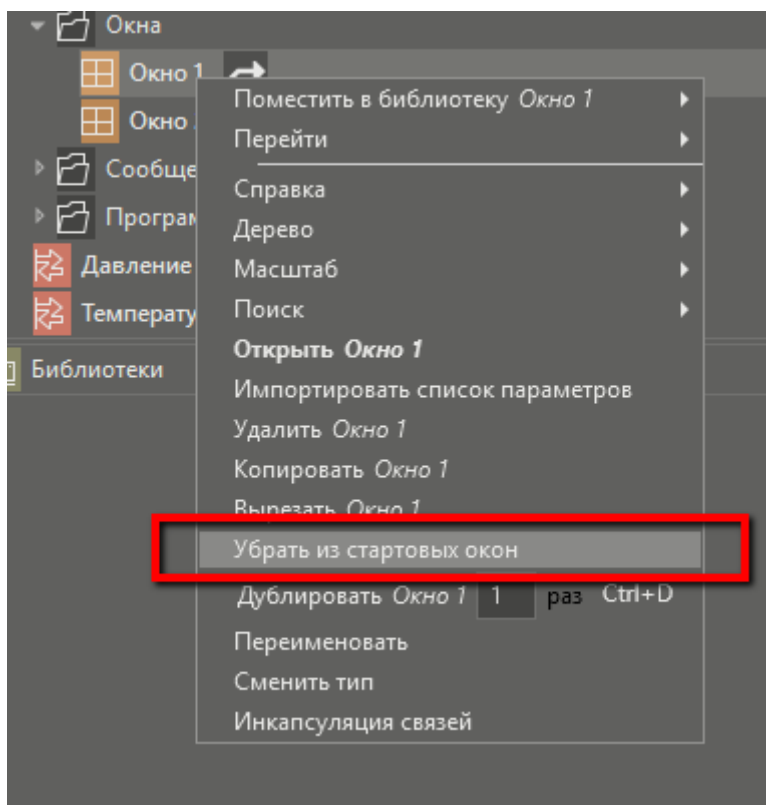
Стартовая окно – это окно, которое открывается в браузере при запуске узла в стартовом контейнере Шаблона экрана. Для назначения окна, которое откроется в контейнере стартового окна, необходимо в контекстном меню нужного окна выбрать пункт Назначить стартовым окном, и в списке выбрать шаблон экрана нужного узла.



При этом образуется ссылка у Шаблона экрана на окно и у окна - на шаблон экрана:



При помощи пункта контекстного меню Убрать из стартовых окон разрывается связь между окном и шаблоном экрана:



Важно! Если ни одно окно не назначено стартовым, то веб-сервер среды исполнения использоваться не будет. Если проект загружается в исполнительную систему автоматически, то клиент визуализации не запустится в автоматическом режиме.

8.5.3. КЕШИРОВАНИЕ ОКОН

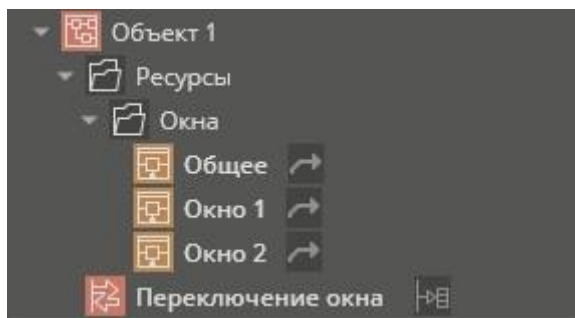
Скорость загрузки окон в режиме исполнения зависит от количества элементов, которые располагаются в окне: чем больше элементов, тем медленнее происходит загрузка окна.

Если пользователь в процессе работы закрывает открытые окна, то они выгружаются из оперативной памяти, и следующее открытие потребует повторной загрузки, что приводит к увеличению времени реакции пользователя на контролируруемую ситуацию.

Для того чтобы уменьшить время открытия окон в режиме исполнения, их можно кешировать, т.е. не выгружать из памяти; в этом случае повторное открытие будет происходить быстрее.

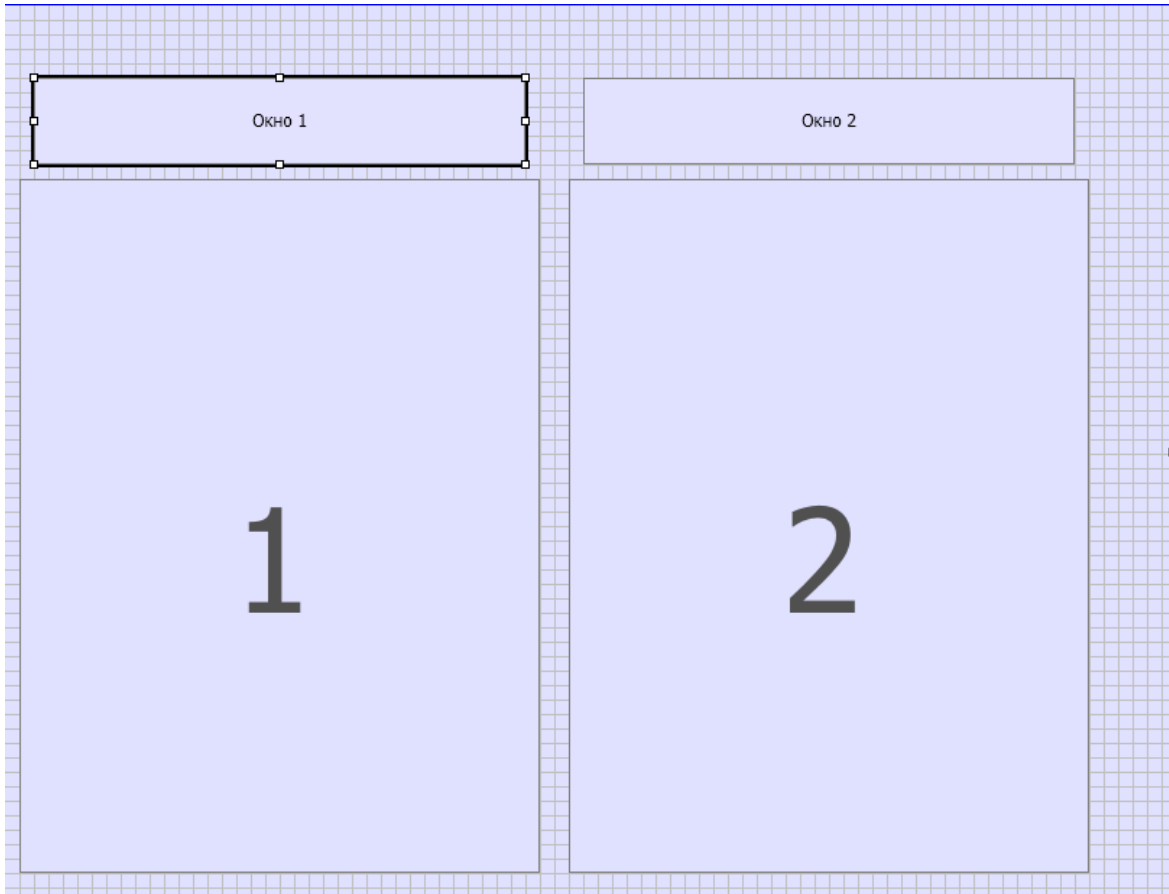
С этой целью можно использовать следующий прием.

Допустим, в проекте есть два окна, которые должны сменять друг друга в некотором общем окне. Для организации переключения потребуется параметр, имеющий тип целый и начальное значение 1.



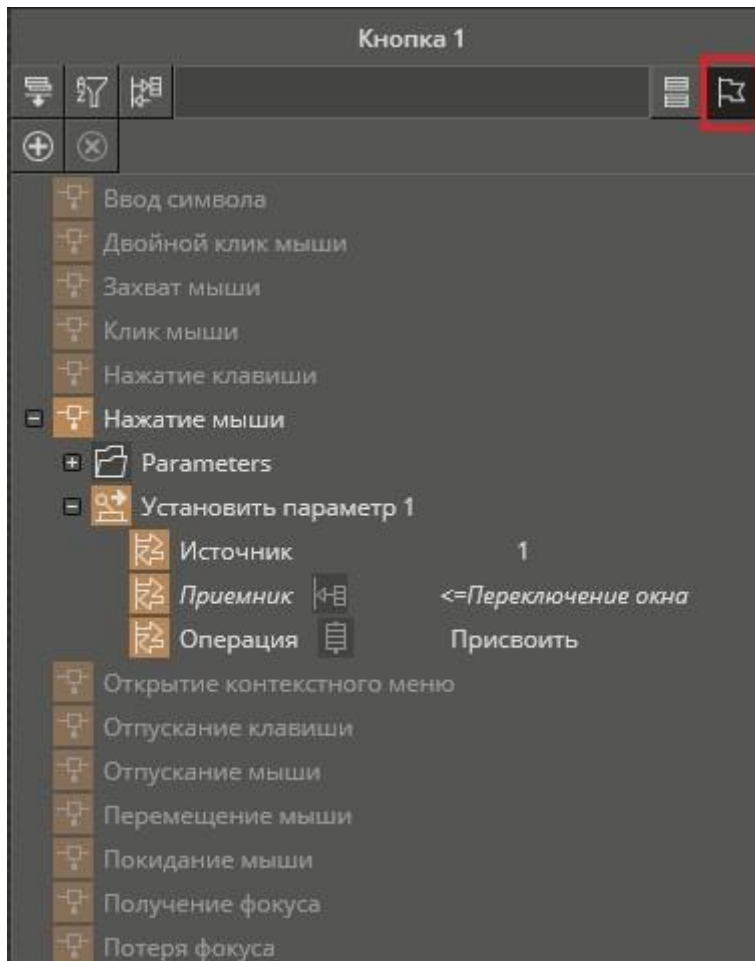
В общее окно перетащим левой кнопкой мыши элементы проекта Окно 1 и Окно 2.

И из категории палитры Диалог перетащим в рабочую область редактора НМІ две кнопки без фиксации, которые будут отвечать за открытие соответственно Окна 1, Окна 2:

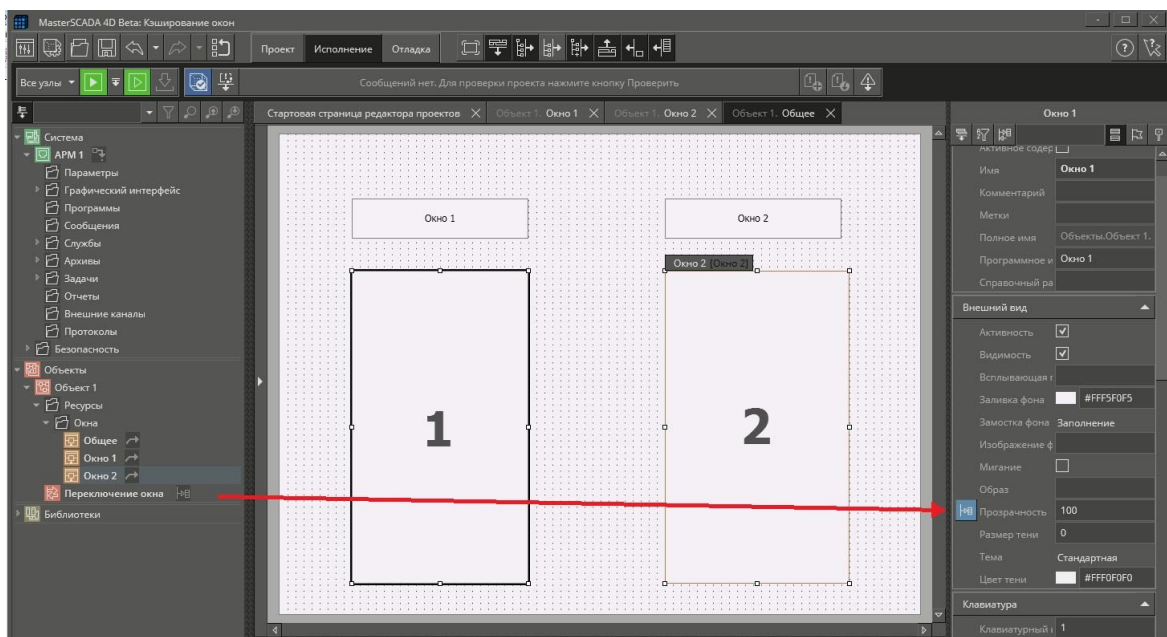


Необходимо, чтобы при нажатии на кнопку Окно 1 параметр Переключение окна принимал значение 1, на кнопку Окно 2 принимал значение 2. Для этого у каждой кнопки следует настроить событие, которое будет выполняться при нажатии мыши.

Для этого необходимо в панели настроек кнопок перейти в режим формирования событий. Затем нажать правой кнопкой мыши на элемент списка Нажатие мыши, и в контекстном меню выбрать пункт Установить параметр. В появившуюся группу настроек на элемент *Приемник* необходимо перетащить параметр Переключение окна. Источником является константа: 1 или 2.

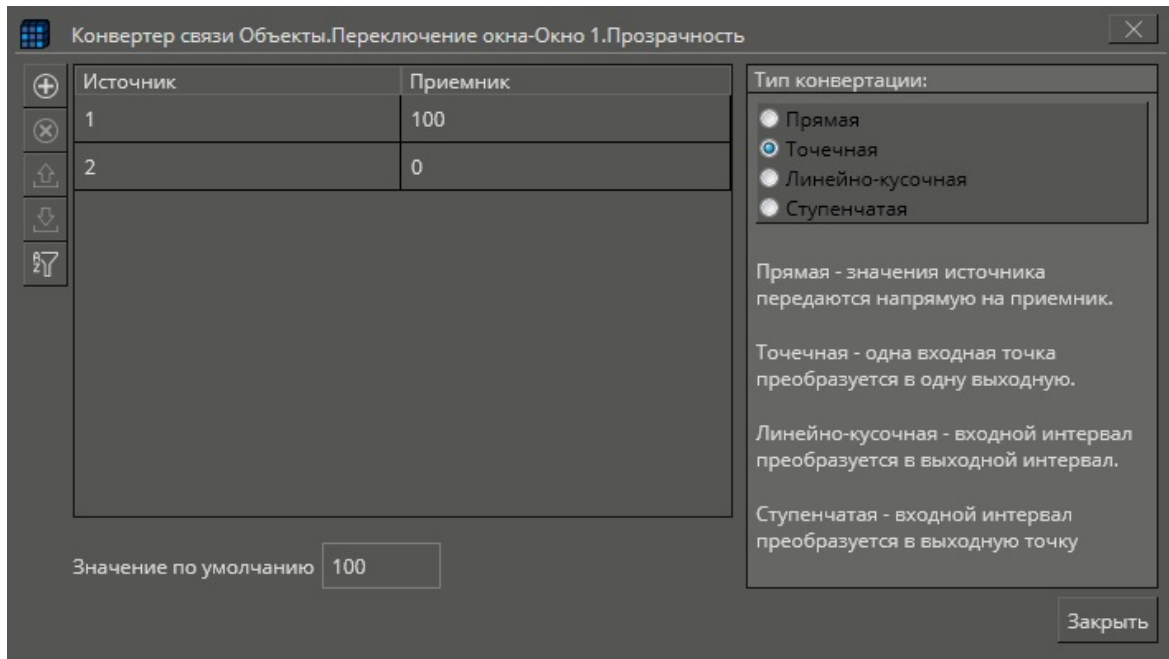


Далее, необходимо установить связь между параметром дерева и свойством Прозрачность окна. Для этого необходимо перетащить Переключение окна в поле нужного свойства.

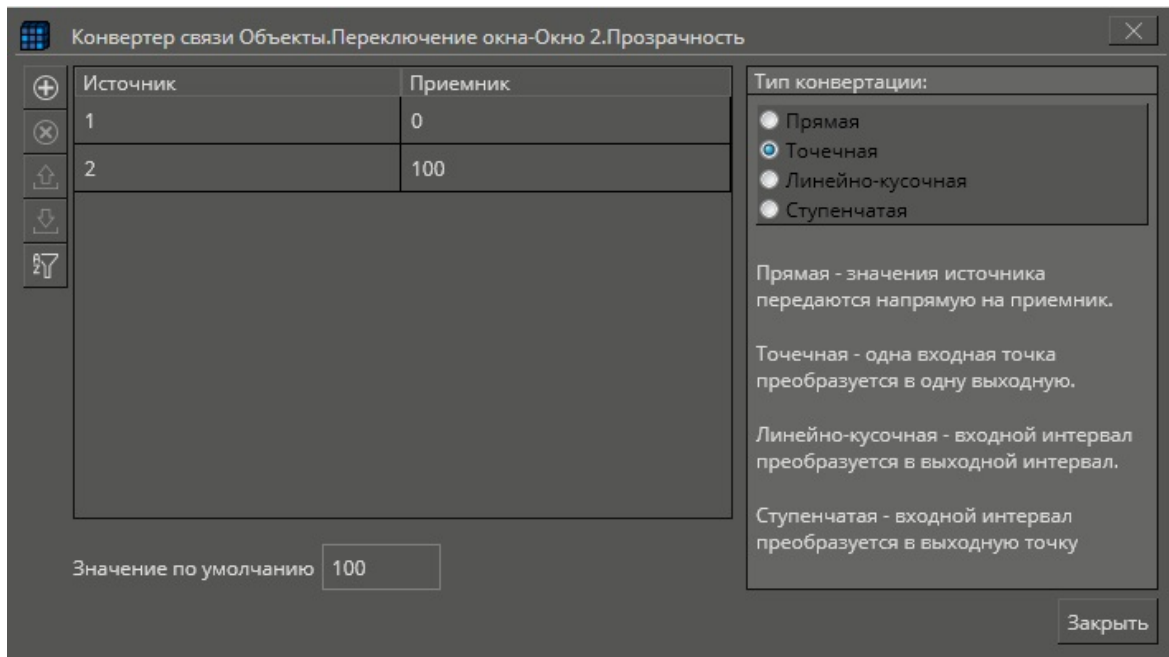


Открыв окно конвертера связи, необходимо настроить точечную конвертацию. Значение свойства 100 обозначает, что окно будет видно пользователю, а 0 - что окно будет скрыто.

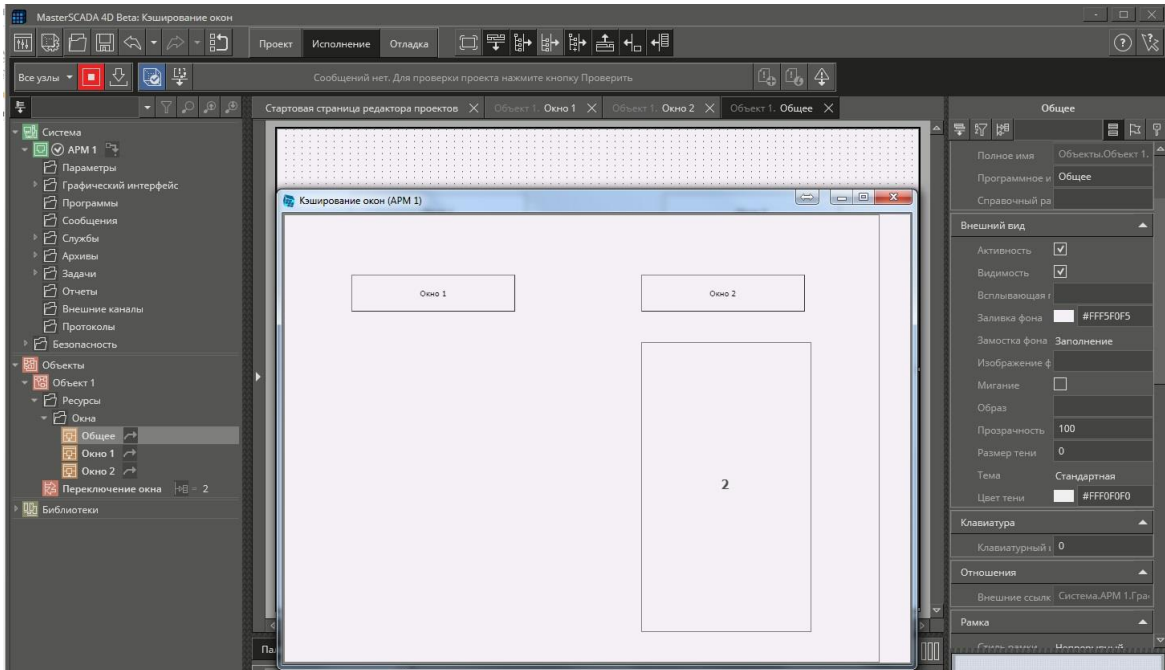
Настройки конвертации для Окна 1:



Настройки конвертации для Окна 2:



В режиме исполнения получим результат:



Расположив окна друг над другом, можно добиться эффекта, что в одном месте внутри стартового окна открываются разные окна.

8.5.4. ОСОБЕННОСТИ СОЗДАНИЯ ОКОН В БИБЛИОТЕКЕ

Если какая-то часть мнемосхемы повторяется в проекте несколько раз, то правильнее создать шаблон окна в библиотеке, а в проекте уже использовать его экземпляры.

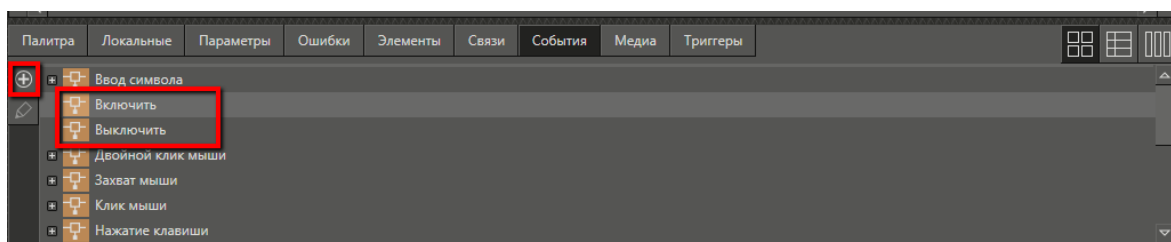
Однако, при таком подходе разработчик проектов должен учитывать ряд особенностей.

Работа с событиями

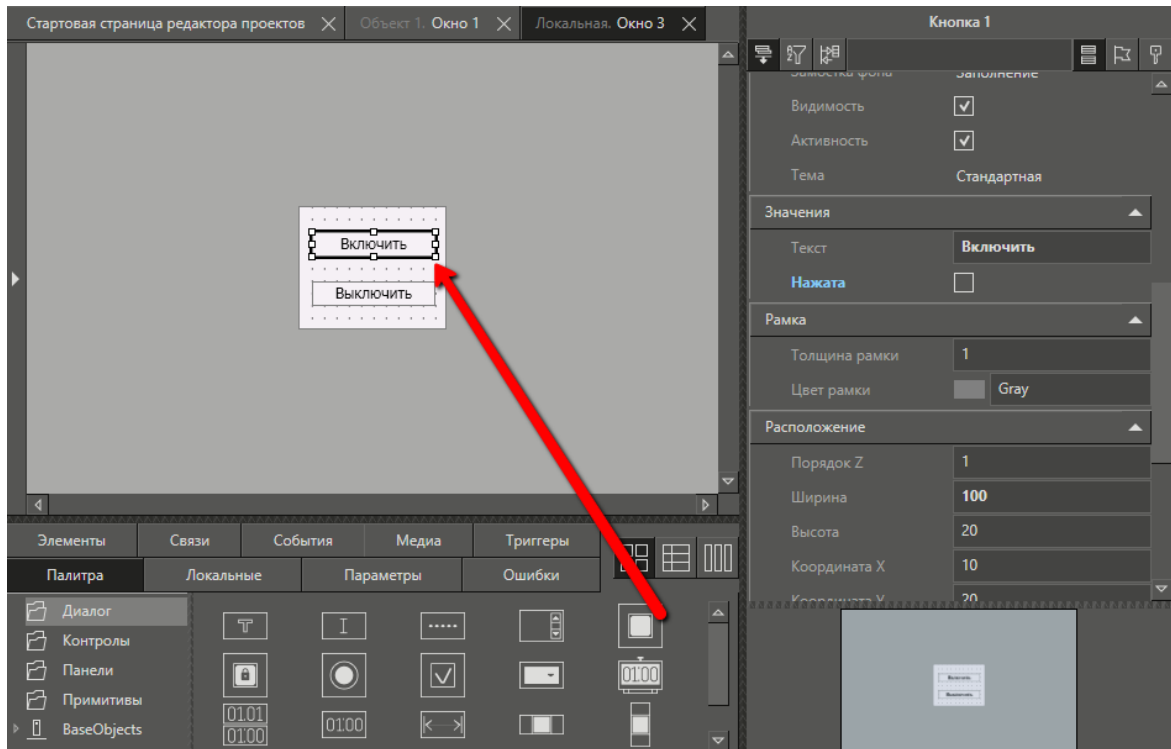
Важно! Разработчик проекта не имеет доступа к событиям отдельных элементов экземпляра окна, и может работать только с событиями окна.

Рассмотрим ситуацию, когда на одной и той же мнемосхеме, или на разных мнемосхемах, есть несколько групп элементов управления, которые выглядят одинаково, имеют один и тот же алгоритм работы, но должны быть связаны с разными параметрами. Например, одна группа работает с одним насосом, а другая группа - с другим насосом. В этом случае, создаем окно в библиотеке.

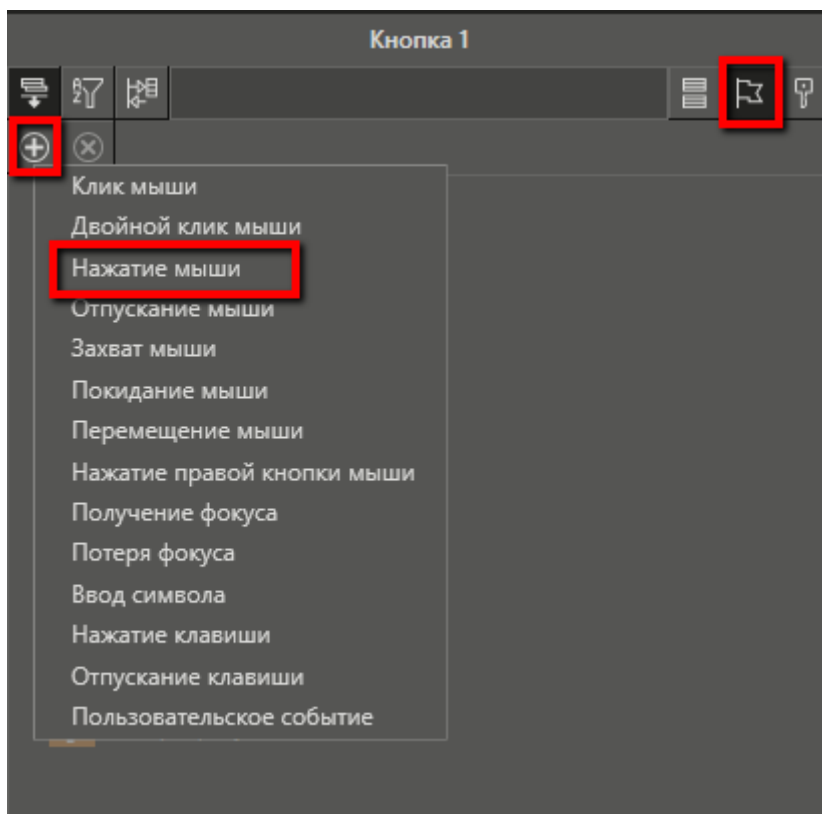
В легенде редактора НМІ на вкладке События создаем два события, назовем их Включить и Выключить:



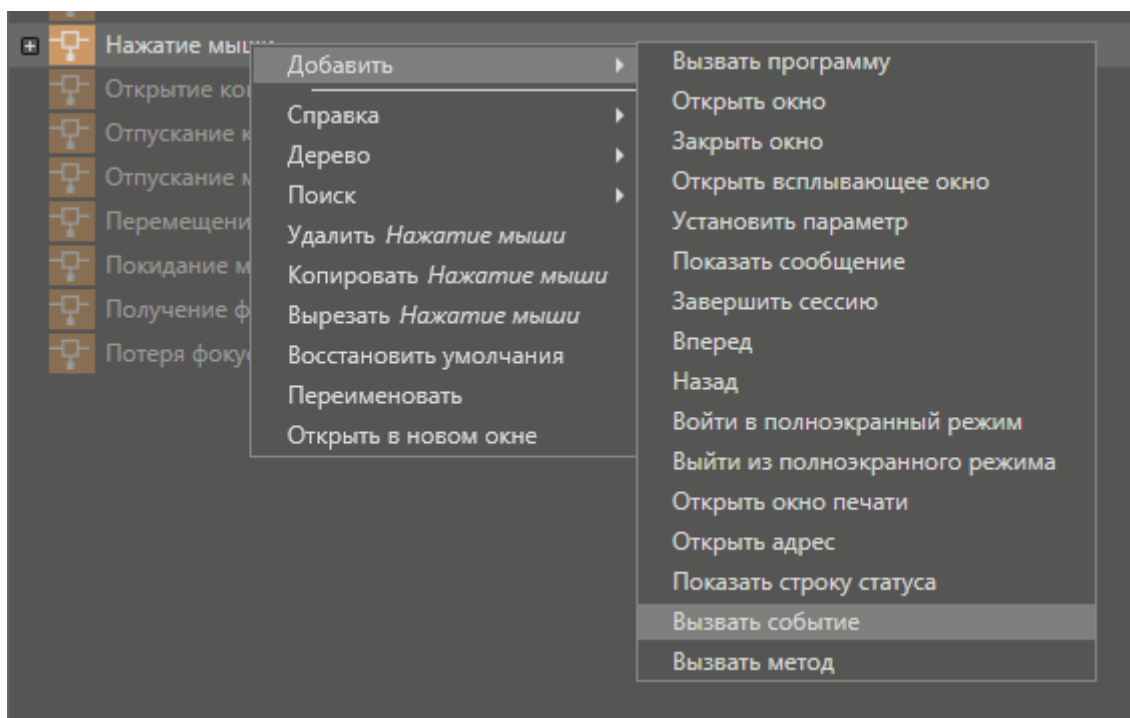
Добавляем в него элементы управления, например, две кнопки.



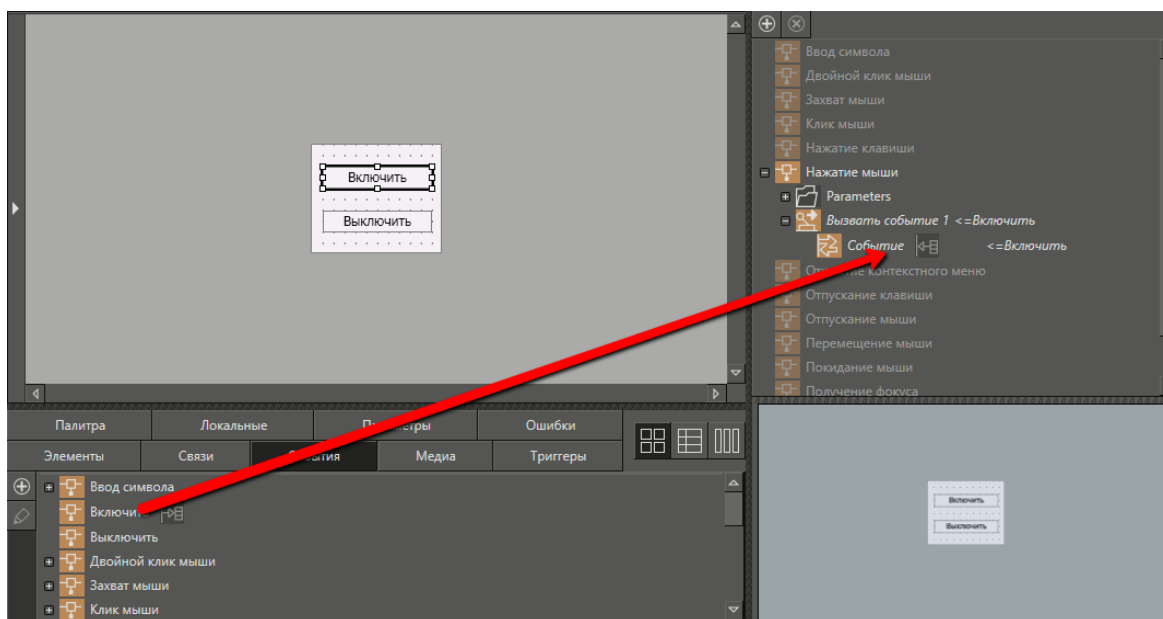
Затем необходимо сделать вызов событий окна из событий кнопок. Для этого выделим кнопку, и в её панели свойств переключимся в режим отображения, позволяющий работать с событиями. Добавим нужное событие, например, Нажатие мыши:



В это событие необходимо добавить действие Вызвать событие:



Затем перетащить пользовательское событие окна в параметры созданного действия:

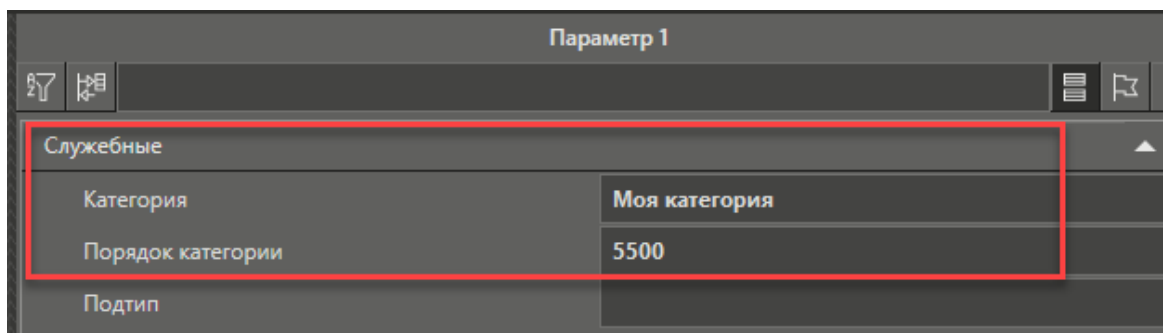


После этого, при помещении экземпляра окна в другое окно, можно будет определять действие, которое будет происходить при вызове того или иного события окна.

Порядок категории у параметров окна.

Параметры, добавленные в клеммники окон, появляются и в панели свойств окна в категории Параметры. Можно изменить название категории и ее порядок следования в панели свойств.

Для этого в настройках Параметра нужно задать два свойства в Категория и Порядок категории, которые находятся в в группе свойств Служебные.



Если в окно добавлено несколько параметров в одну и ту же категорию но с разными значениям свойств Порядок категории, то место категории в панели свойств определяется исходя из минимального значения, заданного у всех параметров.

Значения Порядок категории у некоторых стандартных категорий свойств:

```
<Category Category="Общие" CategoryOrder="100"/>
```

```
<Category Category="Служебные" CategoryOrder="200"/>
```

```
<Category Category="Значения" CategoryOrder="1000"/>
```

```
<Category Category="Ось X" CategoryOrder="1010"/>
```

```
<Category Category="Ось Y" CategoryOrder="1011"/>
```

```
<Category Category="Текст" CategoryOrder="1020"/>
```

```
<Category Category="Внешний вид ползунка" CategoryOrder="1020"/>
```

```
<Category Category="Параметры" CategoryOrder="1030"/>
```

```
<Category Category="Шкала" CategoryOrder="1040"/>
```

```
<Category Category="Курсор выбранной позиции" CategoryOrder="1070"/>
```

```
<Category Category="Сетка" CategoryOrder="1100"/>
```

```
<Category Category="Журнал" CategoryOrder="1110"/>
```

```
<Category Category="Данные" CategoryOrder="1120"/>
```

```
<Category Category="Панель" CategoryOrder="1130"/>
```

```
<Category Category="Линия" CategoryOrder="1140"/>
```

```
<Category Category="Архив" CategoryOrder="1150"/>
```

```
<Category Category="Внешний вид" CategoryOrder="2100"/>
```

```
<Category Category="Расположение" CategoryOrder="2200"/>
```

```
<Category Category="Рамка" CategoryOrder="2300"/>
```

```
<Category Category="Отношения" CategoryOrder="4000"/>
```

```
<Category Category="Трансформация" CategoryOrder="4100"/>
```

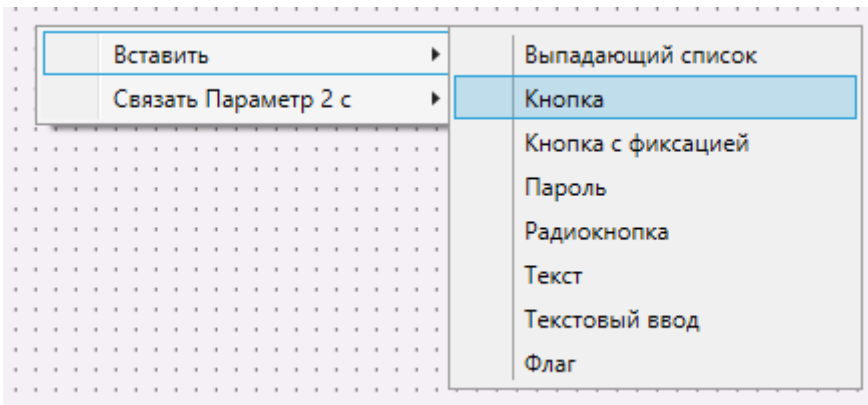
```
<Category Category="Клавиатура" CategoryOrder="4200"/>
```

```
<Category Category="Экран" CategoryOrder="4300"/>
```

8.5.5. ОСОБЕННОСТИ РАБОТЫ С ЭЛЕМЕНТАМИ КНОПКА И КНОПКА С ФИКСАЦИЕЙ

Изменение параметра типа BOOL

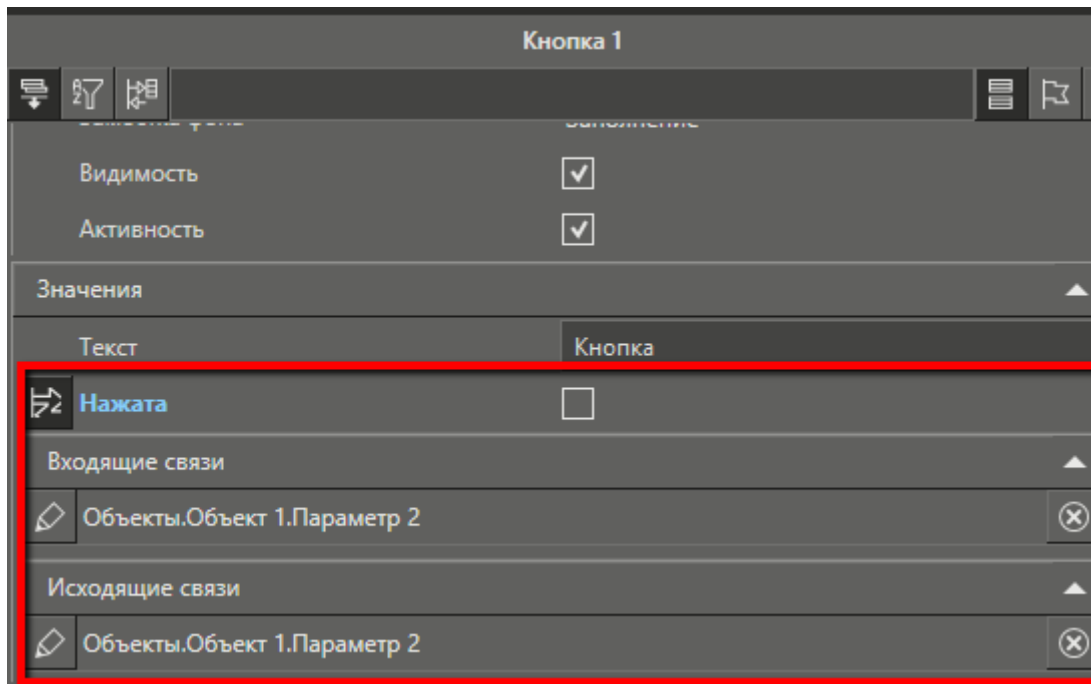
Если элементы Кнопка и Кнопка с фиксацией используются для изменения значения параметра типа BOOL, то необходимо перетащить этот параметр в нужное место рабочей области редактора НМІ правой кнопкой мыши, и в выпадающем списке выбрать требуемый элемент:



После этого в окне появится элемент Кнопка или Кнопка с фиксацией:



Кроме того, будет автоматически установлена двунаправленная связь между параметром проекта и свойством Нажата:

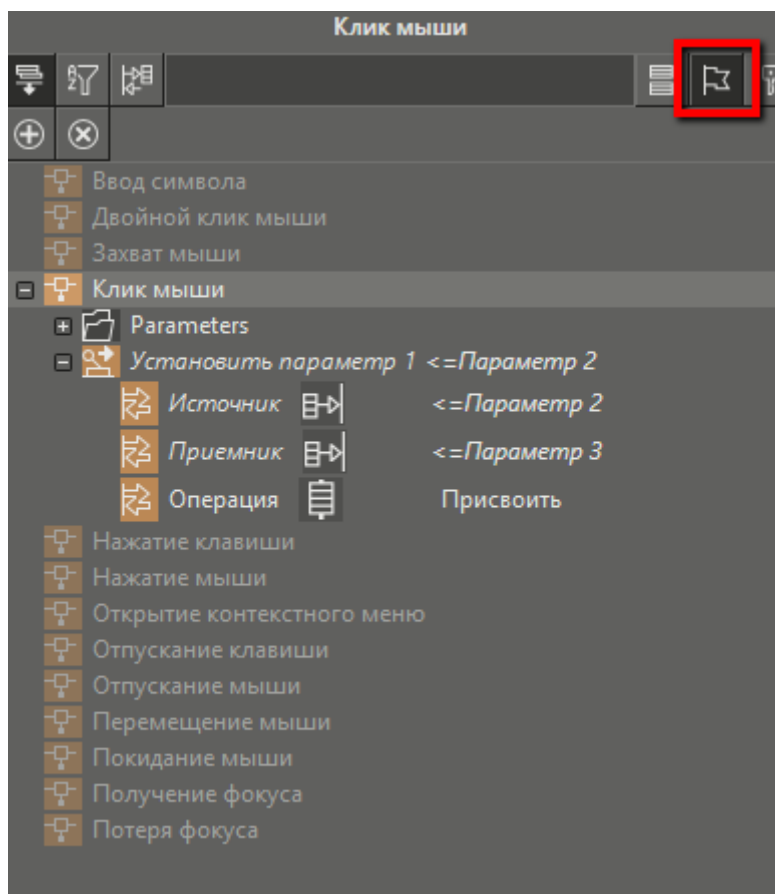


Если в клиенте визуализации в режиме исполнения нажать на Кнопку левой кнопкой мыши, то связанный параметр будет принимать значение TRUE до тех пор, пока Кнопка не будет отпущена.

Если в клиенте визуализации в режиме исполнения нажать на Кнопку с фиксацией левой кнопкой мыши, то связанный параметр будет принимать значение TRUE до тех пор, пока Кнопка с фиксацией не будет повторно нажата левой кнопкой мыши.

Выполнение произвольного действия

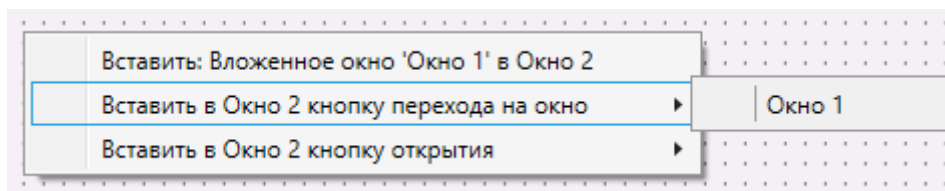
Если при нажатии на кнопку необходимо выполнить какое-либо действие, например, значение одного параметра присвоить значению другого, либо переключить окно, либо открыть всплывающее окно, то в окно необходимо добавить элемент Кнопка. Затем, переключившись в панели свойств в режим работы с событиями, в нужное событие добавить требуемое действие:



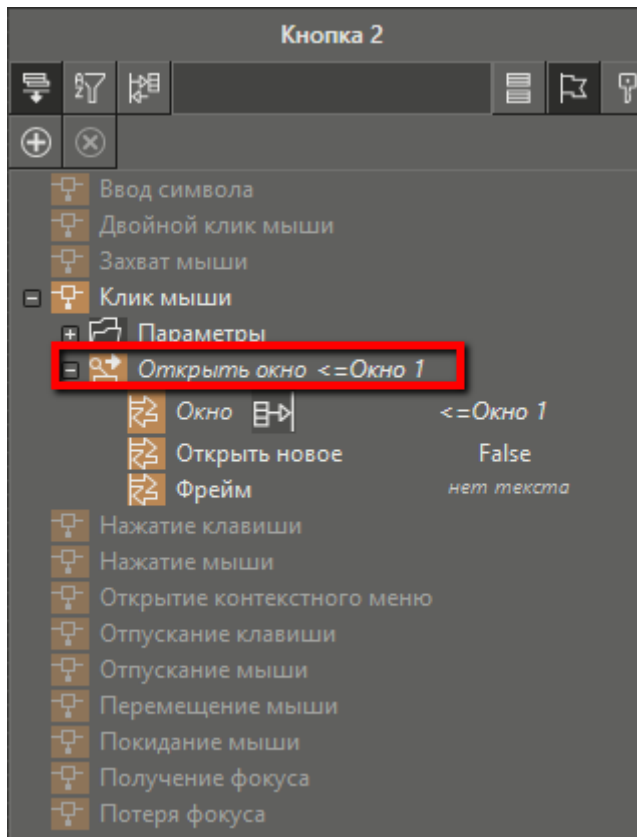
Переключение окон

Если необходимо, чтобы при нажатии на кнопку вместо текущего окна открывалось другое окно, то можно пойти по пути, описанному выше, т.е. добавить кнопку из библиотеки, переключить панель свойств в режим События и добавить на событие Клик мыши или Нажатие мыши действие Открыть окно.

Второй вариант более быстрый: можно перетащить правой кнопкой мыши объект, содержащий окно, либо непосредственно само окно в рабочую область редактора HMI, и в появившемся меню выбрать пункт Вставить в [имя редактируемого окна] кнопку перехода на окно - [Имя окна, на которое нужно выполнить переход]:



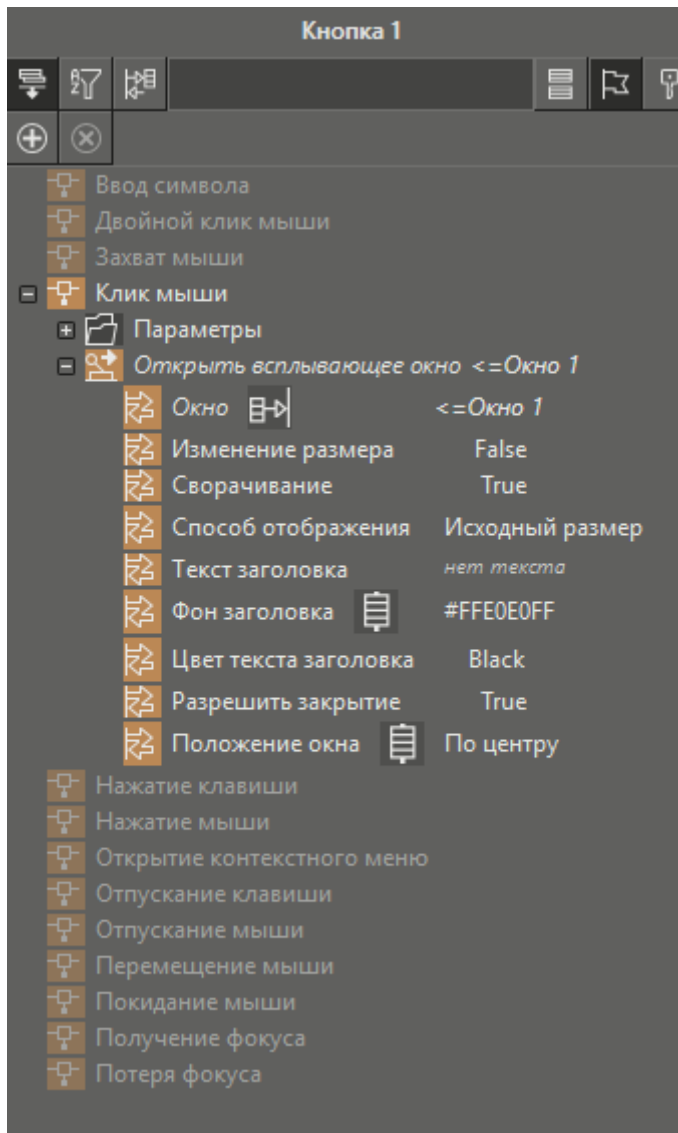
В этом случае кнопка с настроенным действием добавится автоматически:



Открытие всплывающего окна

Самый быстрый способ создать кнопку для открытия всплывающего окна - это перетащить правой кнопкой мыши объект, содержащий окно, либо непосредственно само окно в рабочую область редактора НМІ, и в появившемся меню выбрать пункт Вставить в [имя редактируемого окна] кнопку открытия - [Имя окна, на которое нужно выполнить переход].

В этом случае кнопка с настроенным действием добавится автоматически:



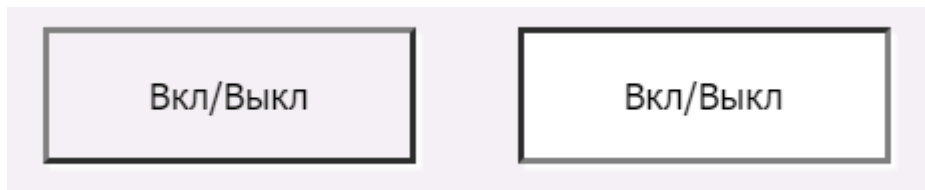
Как сделать нажатие кнопки более заметным

Вид кнопки в отжатом и нажатом положения в клиенте визуализации по умолчанию показан на рисунке:



Слева показана кнопка в отжатом положении, её цвет темнее, чем цвет нажатой кнопки, находящейся справа. Помимо цвета кнопки изменяется и рамка элемента: в отжатом положении нижняя и правая граница рамки жирнее, чем верхняя и левая, а в нажатом состоянии - наоборот.

Для того чтобы сделать нажатое положение кнопки более заметным, можно увеличить значение свойства Толщина рамки:



8.5.6. МАССИВЫ СТРУКТУР В ОКНАХ

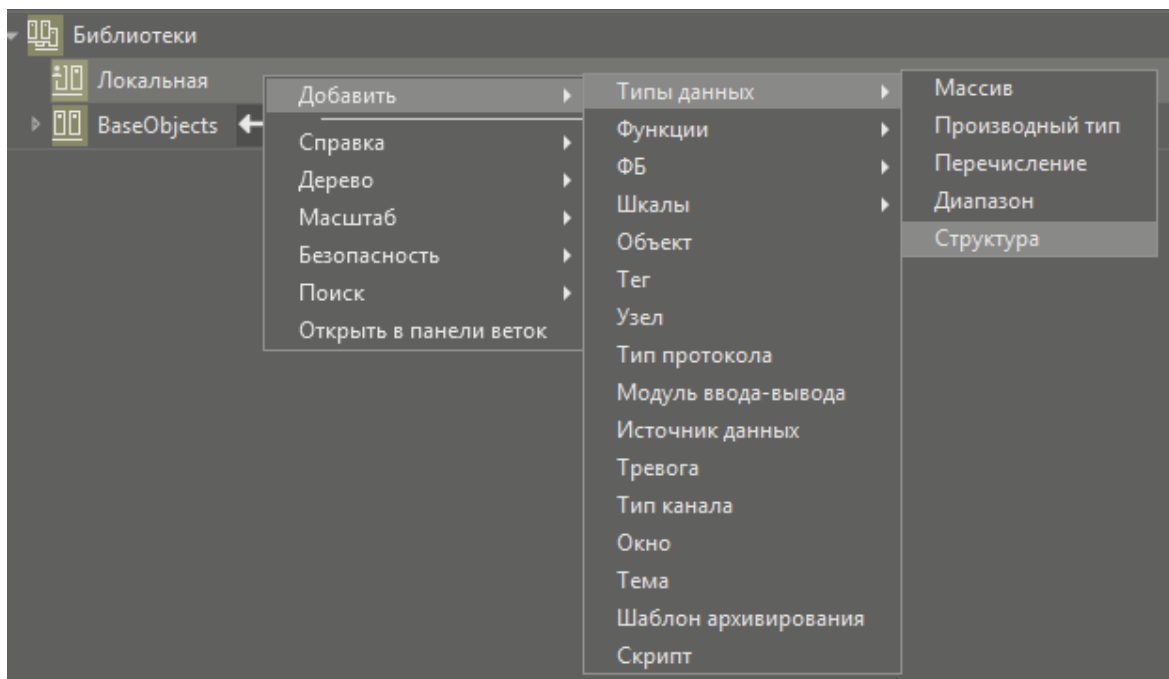
Постановка задачи

Проект предназначен для мониторинга участка трубопровода с 0 по 100 км. Через каждые 10 км на трубопроводе установлено оборудование, передающее в диспетчерскую следующие параметры: расход, давление, температуру, состояние задвижки, наличие или отсутствие аварии. В окне диспетчера каждый участок выглядит одинаково. Требуется найти наиболее оптимальное решение поставленной задачи.

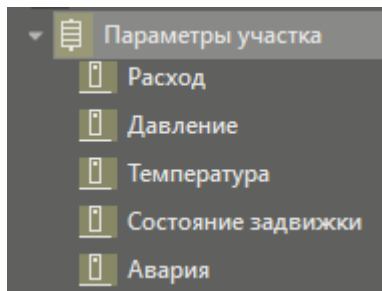
Формирование библиотечных элементов

Создание структуры

Представим набор параметров каждого из участков в виде структуры. Для этого в библиотеке создадим структуру:



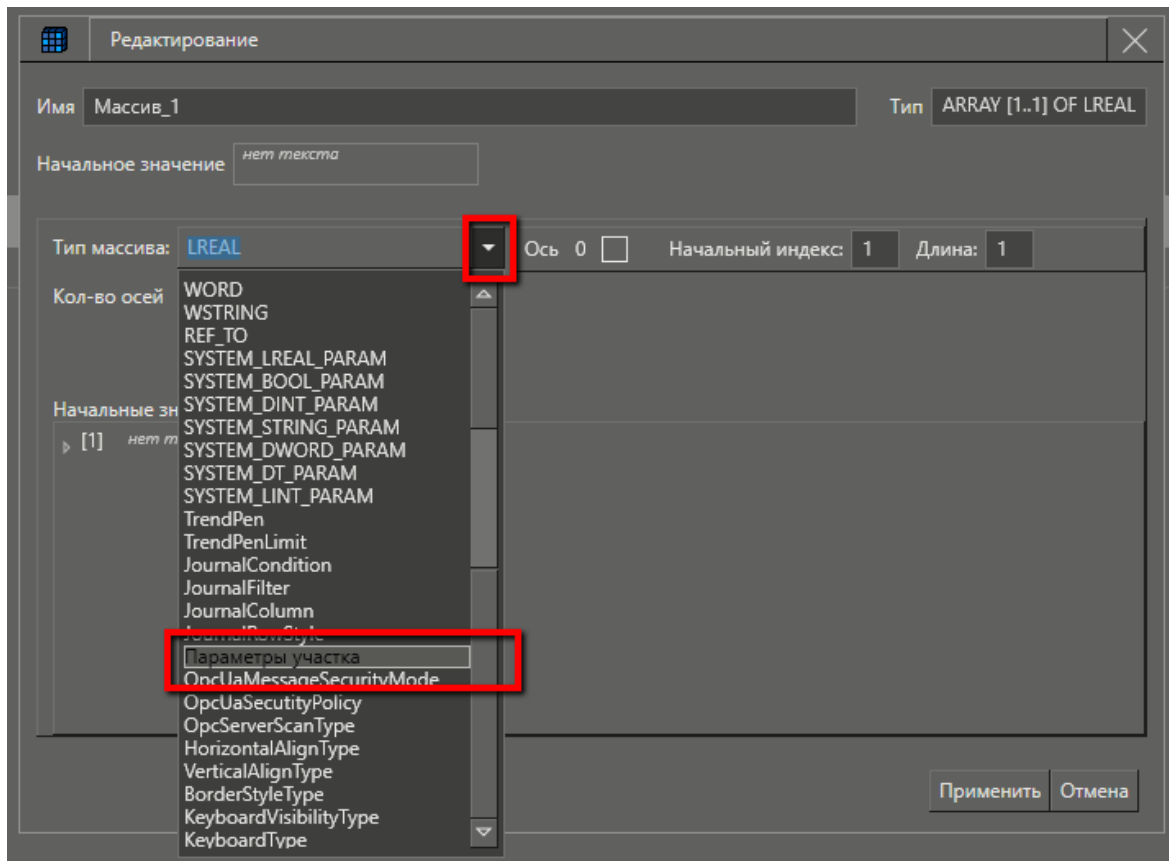
Зададим ее имя Параметры участка и добавим необходимые поля:



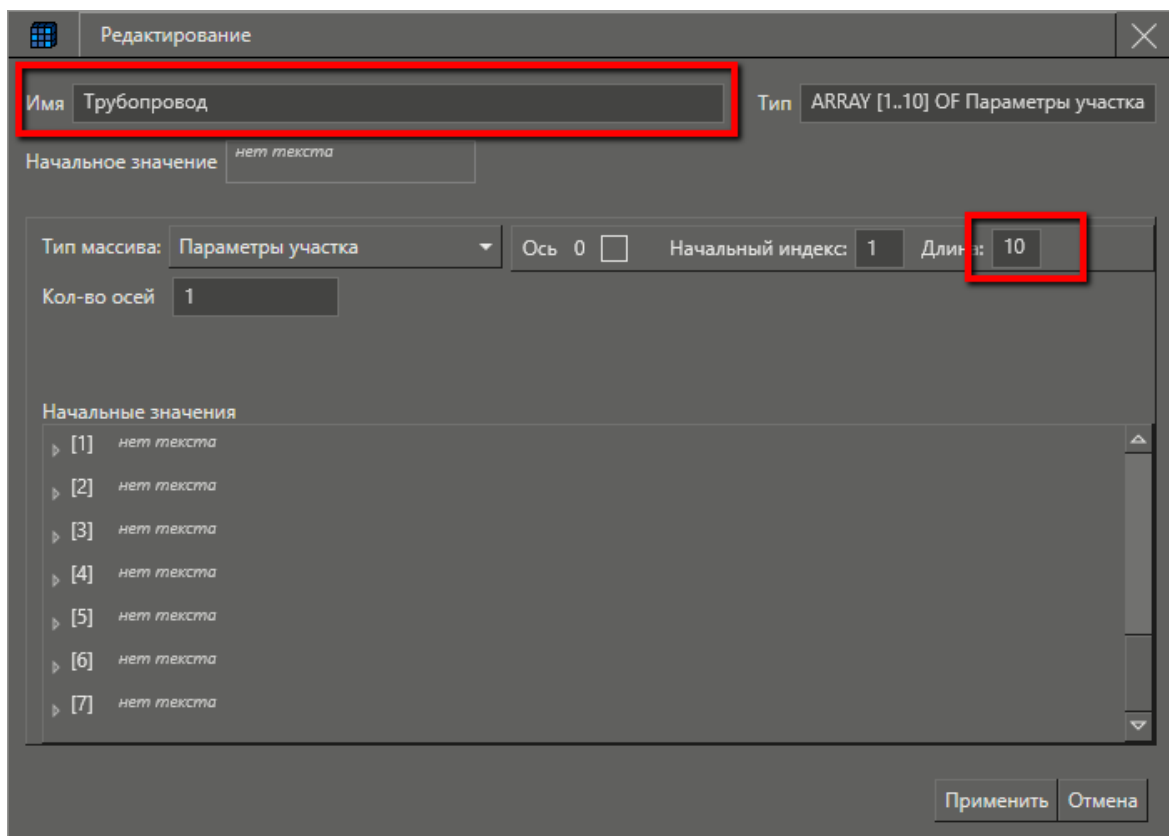
Поле	Тип
Расход	LREAL
Давление	LREAL
Температура	LREAL
Состояние_задвижки	BOOL
Авария	BOOL

Создание массива структур

На основе ранее созданной структуры в библиотеке создадим массив:

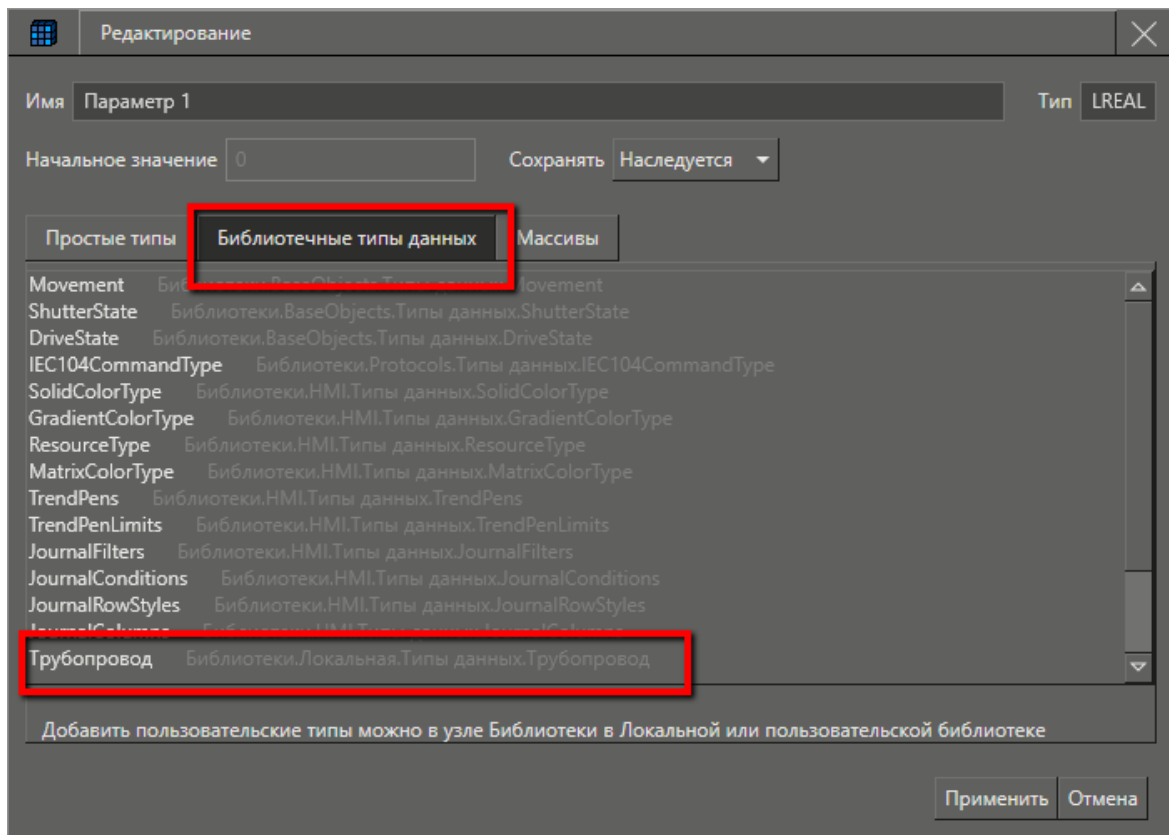


Установим длину массива 10 и изменим имя массива:



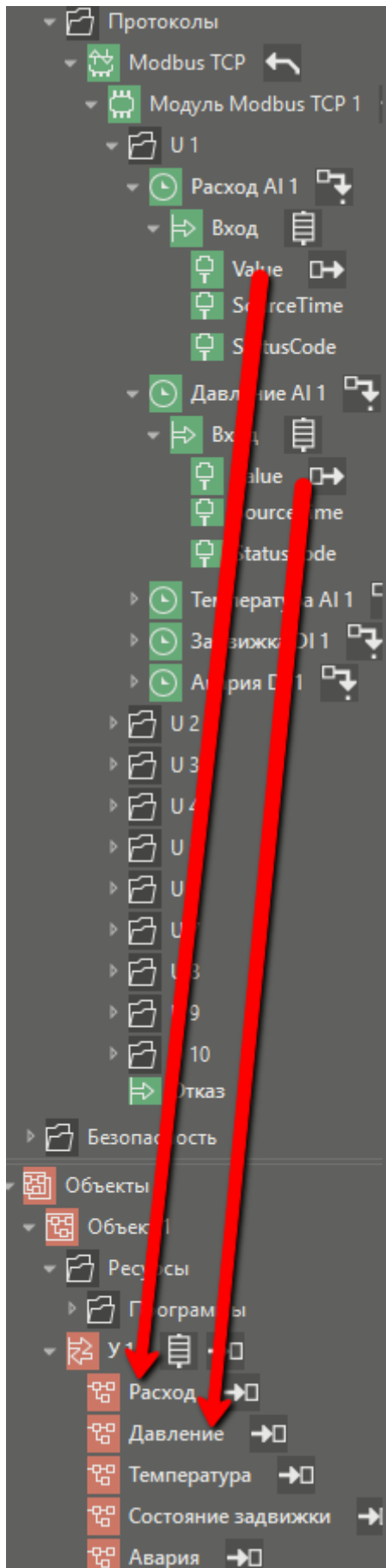
Создание дерева объектов

Добавим в дерево один параметр типа Параметры участка, назовем его У 1 и один Параметр 1 типа Трубопровод:

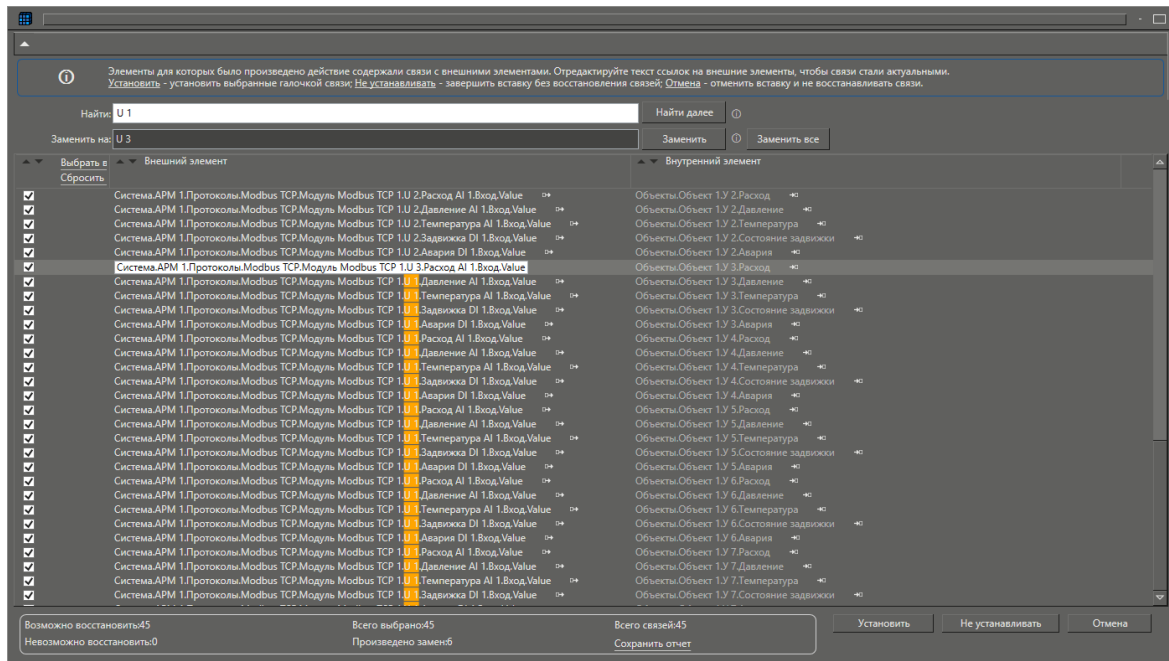


Теперь необходимо обеспечить присвоение данных из дерева система в объект. Допустим, что в примере данные для проекта поступают по протоколу Modbus TCP. Не будем здесь подробно описывать формирование дерева системы, а описание работы с протоколами изложено в разделе Получение и передача данных.

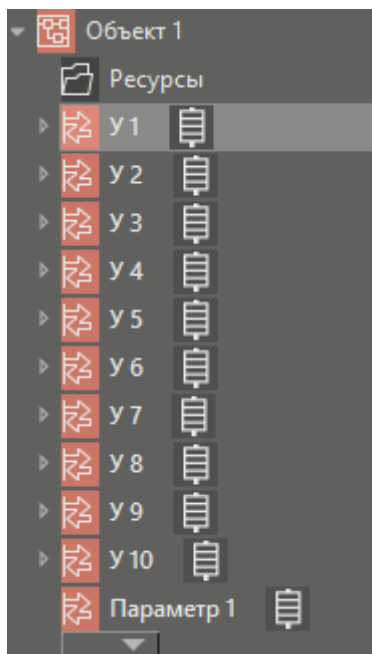
Параметры каналов свяжем с полями структуры параметра У 1:



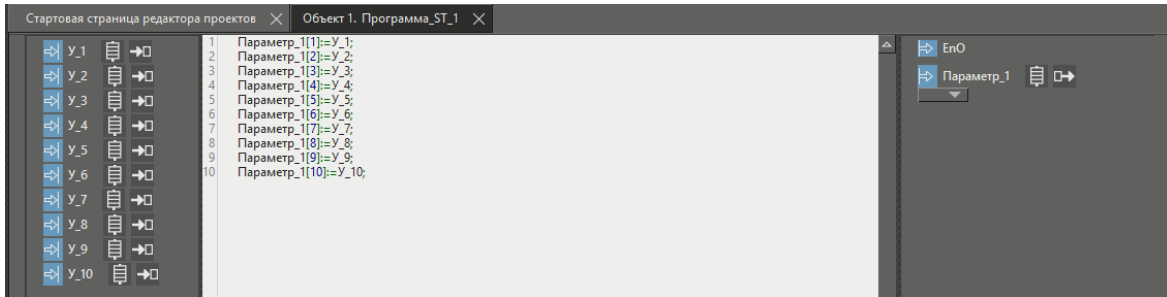
Затем продублируем параметр У 1. Требуется сделать 9 копий. В появившемся диалоговом окне, пользуясь его инструментами, необходимо отредактировать путь к параметрам каналов в дереве системы:



Получим 10 параметров типа структура, поля которых имеют связи параметрами каналов дерева системы:



Затем необходимо обеспечить передачу значений из параметров У 1-У 10 в Параметр 1. Проще всего это сделать, используя язык MasterST. Пример текста программы:

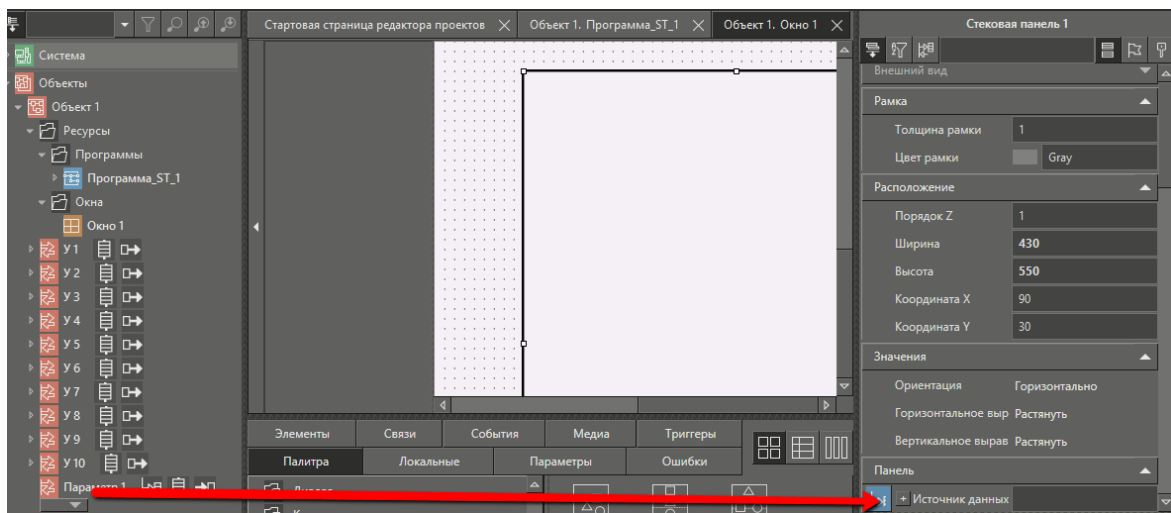


Создание окна

Добавим в объект Окно.

В окно вставляем элемент Стековая панель из категории Панели в палитре редактора HMI.

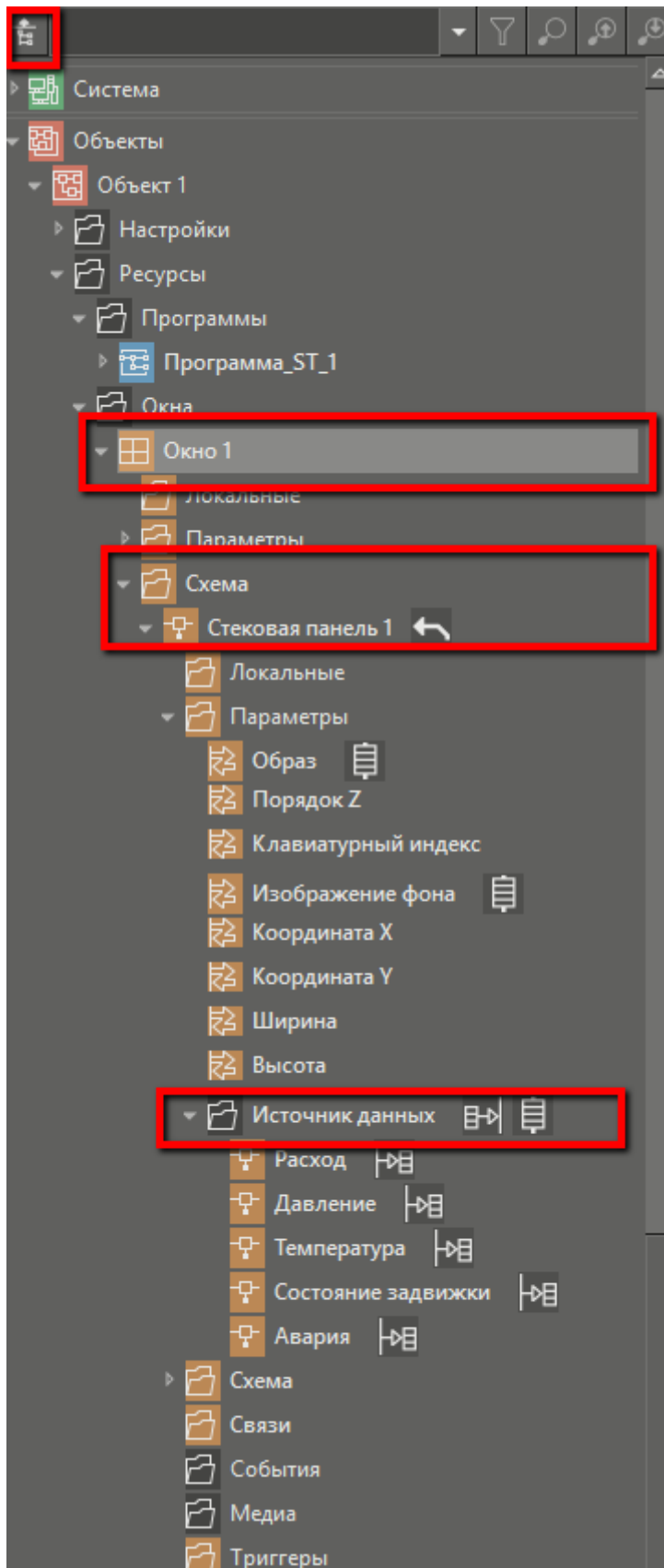
Настраиваем связь между Параметр 1 и свойством панели Источник данных:



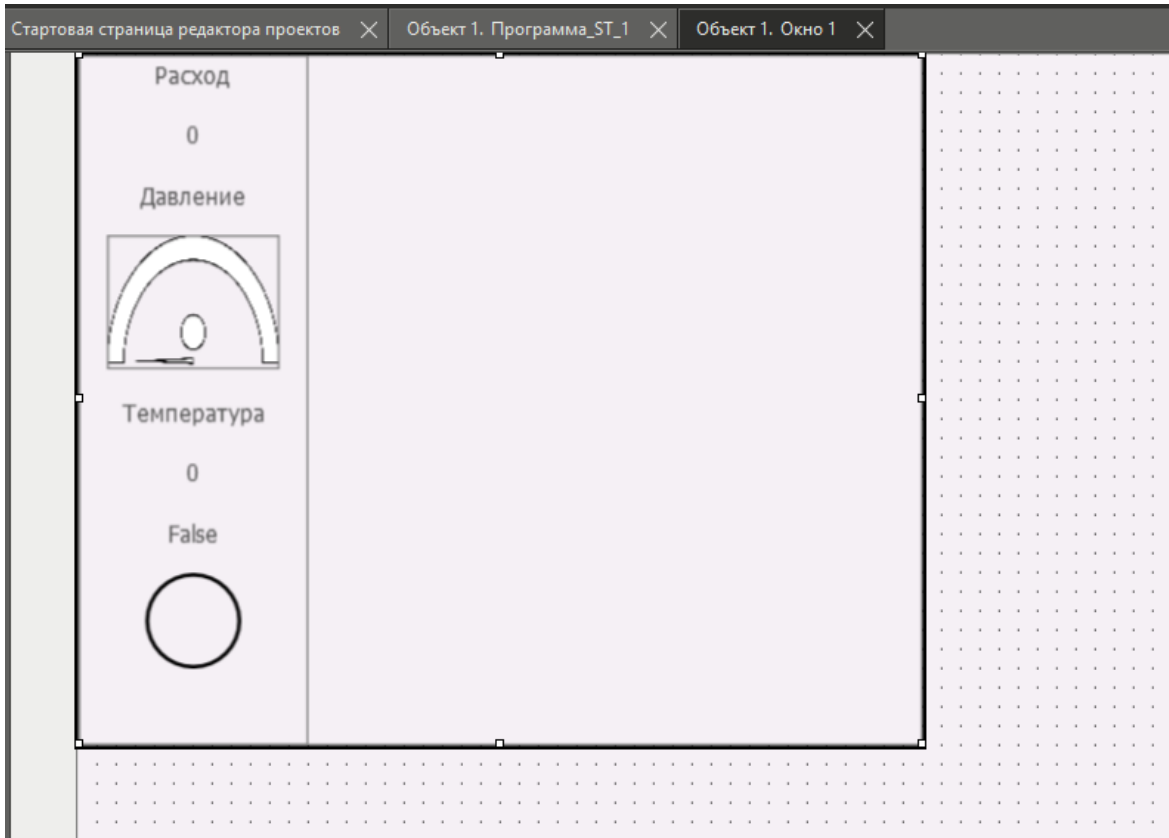
Затем в стековую панель добавляем элемент Панель.

Наконец, в созданном элементе начинаем формировать вид участка, т.е. подготавливаем в стековой панели шаблон, который при отображении в клиенте будет размножен. В зависимости от ориентации стековой панели, элементы в режиме исполнения будут размножаться или слева направо или сверху вниз.

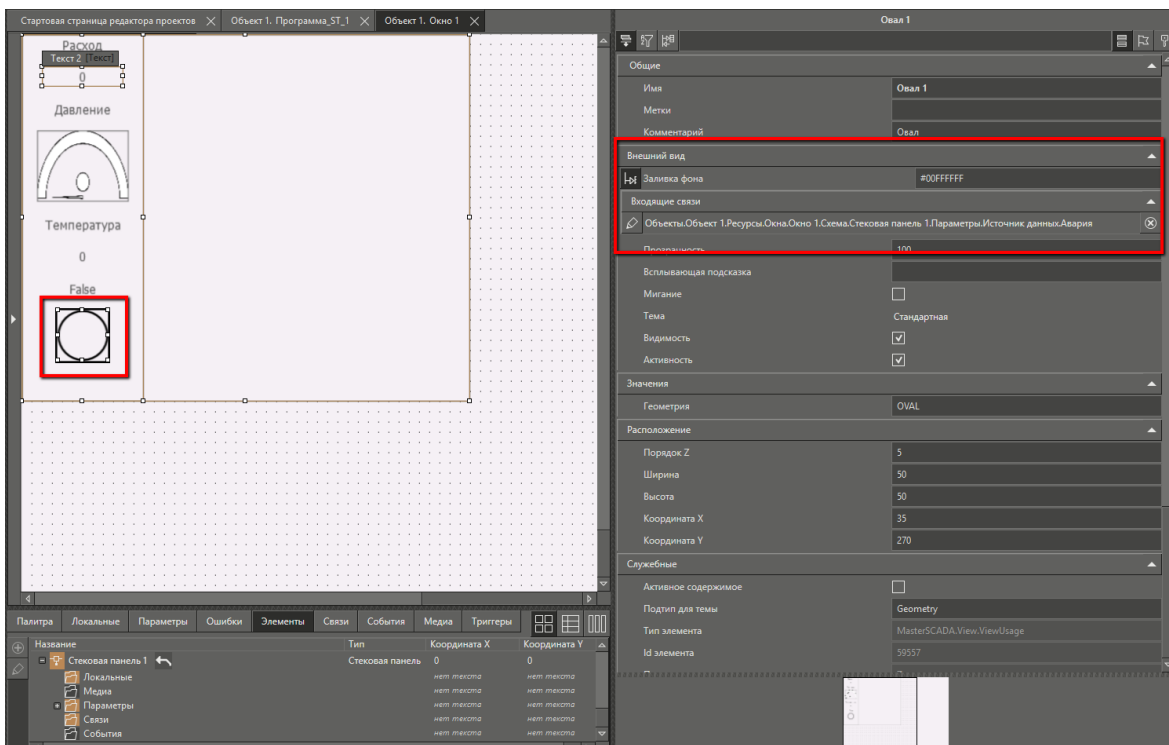
В панели можно использовать различные графические элементы, но при этом связи необходимо устанавливать только между свойствами элементов и параметрами Источника данных (доступны при отображении полного дерева):



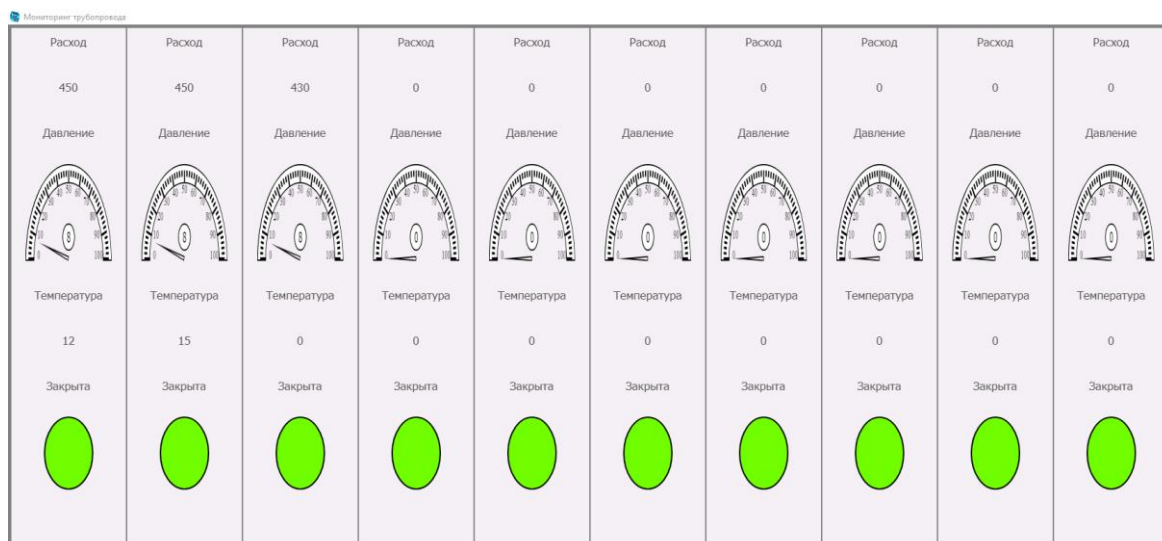
Получим результат:



Где все элементы связаны с параметрами Источника данных Стековой панели. Например:



В зависимости от свойства стековой панели Ориентация в клиенте визуализации мы получим результат, в котором каждая строка или столбец будет отвечать за свой участок трубопровода:



8.5.7. СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКОЙ БИБЛИОТЕКИ ПРИМИТИВОВ

Если необходимо модифицировать элементы из палитры редактора HMI, например, категории Прimitives для дальнейшего использования во многих окнах, то необходимо создать в пользовательской библиотеке элемент Окно, в которое добавить элемент и задать ему необходимые настройки.

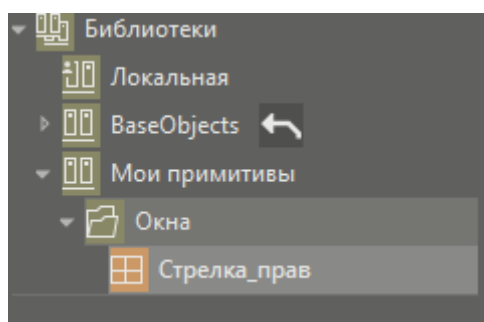
Рассмотрим порядок работы на примере элемента Стрелка. В палитре есть горизонтальная стрелка, указывающая влево. Перечислим порядок операций для создания стрелки указывающей направо.

Шаг 1

Создаем в пользовательскую библиотеку, назовем ее Мои примитивы.

Шаг 2

В новой библиотеке создаем Окно, назовем его Стрелка_прав:

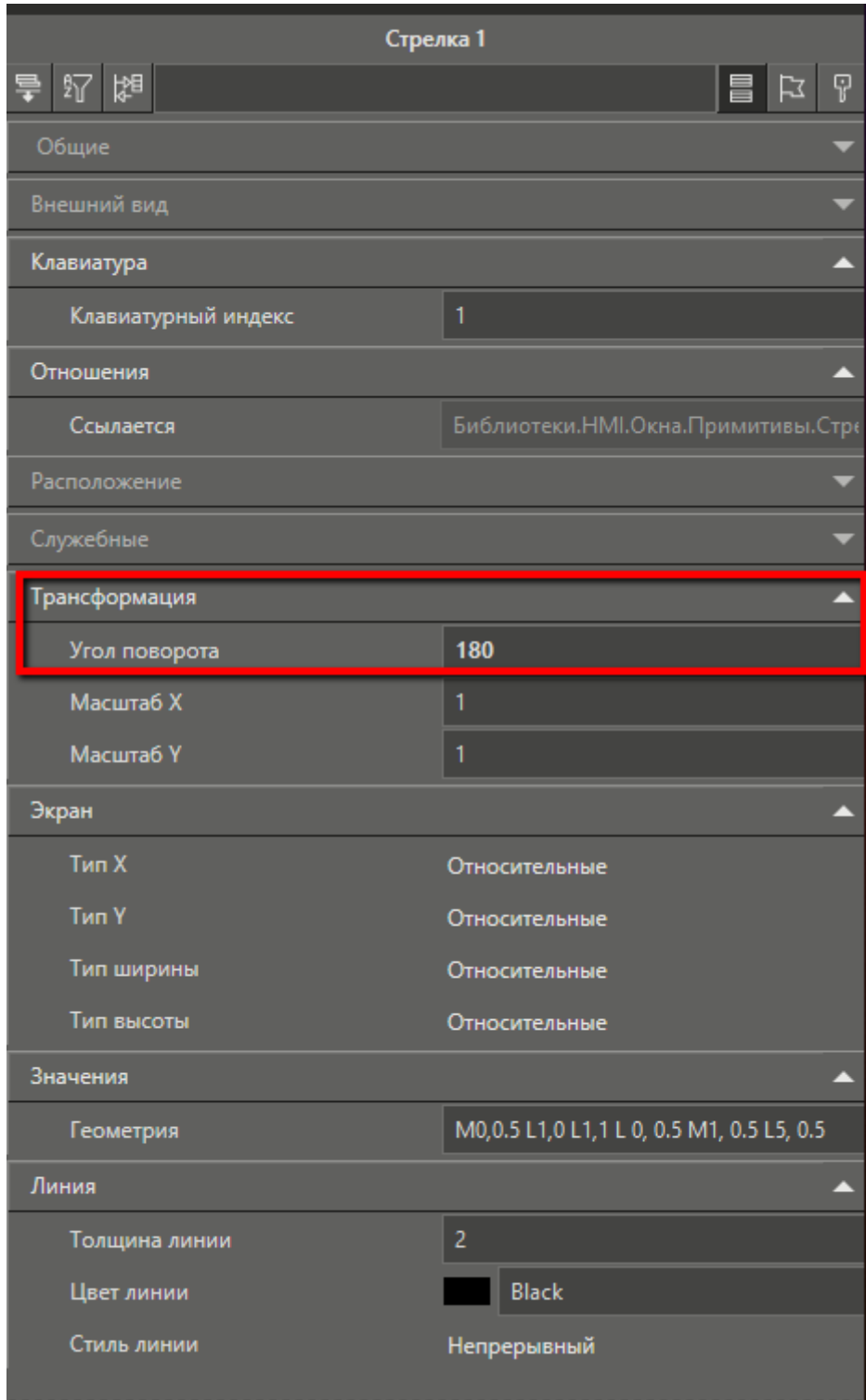


Шаг 3

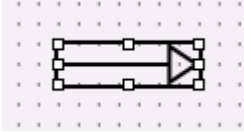
Добавляем в окно элемент Стрелка.

Шаг 4

Определяем необходимый внешний вид элемента. В панели свойств элемента задаем значение свойства Угол поворота, равное 180:



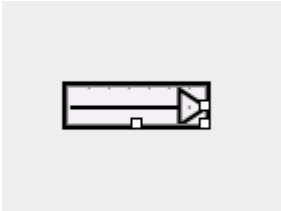
Получим результат:



Разместим элемент в верхнем левом углу.

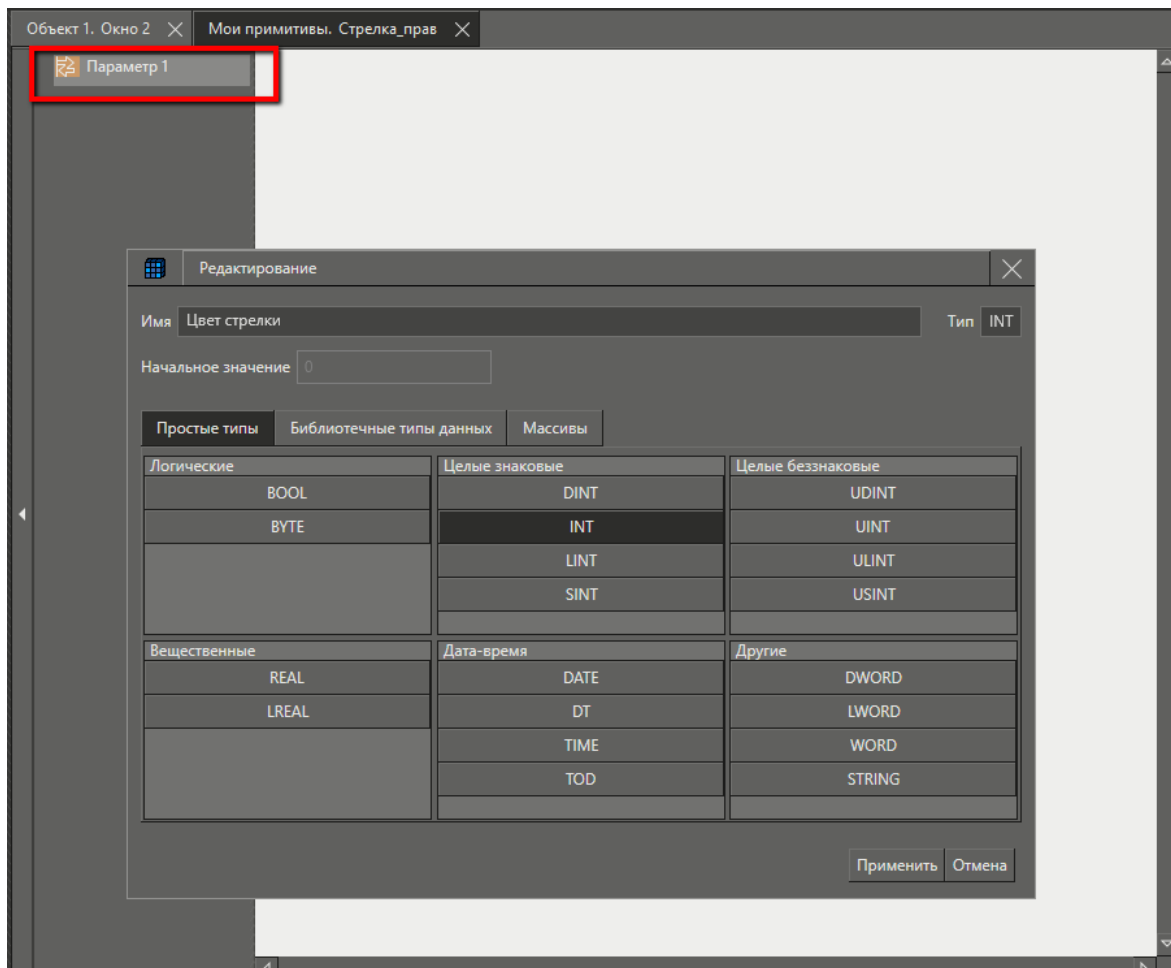
Шаг 5

Изменим размер окна так, чтобы он совпадал с размером стрелки:

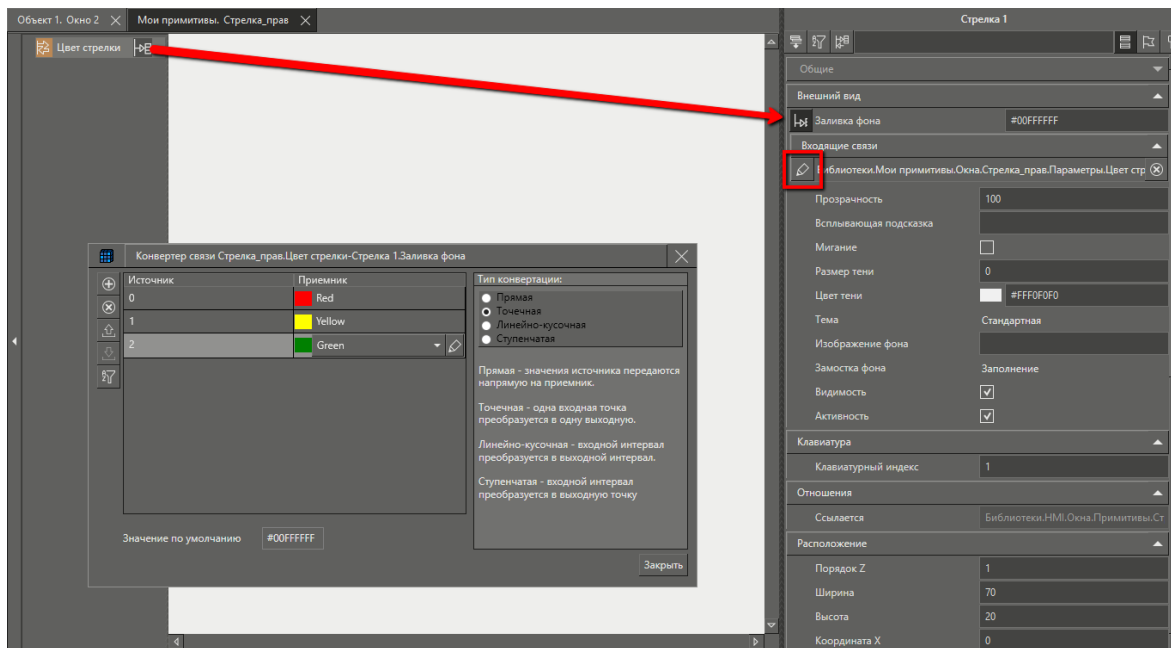


Шаг 6

Если в процессе работы необходимо менять какое-либо свойство стрелки, например, ее цвет, то следует добавить Параметр в клеммник окна. Например, добавим параметр типа INT, назовем его Цвет стрелки:

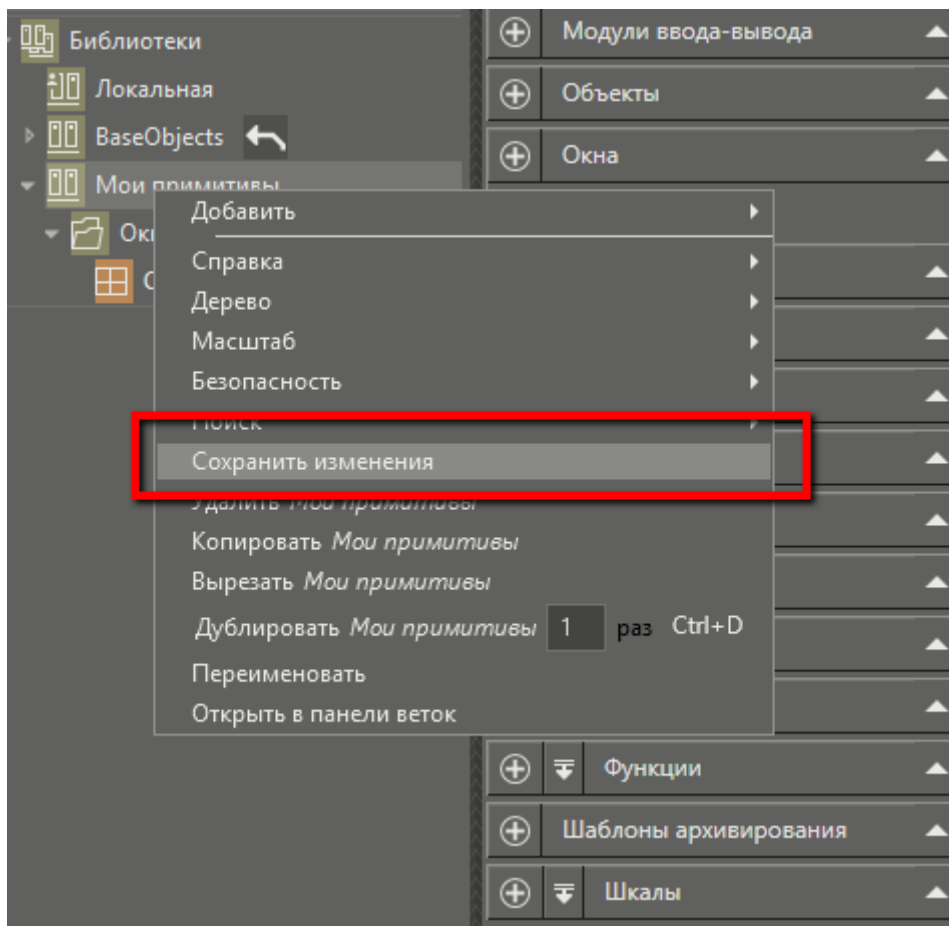


Затем установим связь между свойством стрелки Заливка фона и параметром, и настроим конвертер значений:

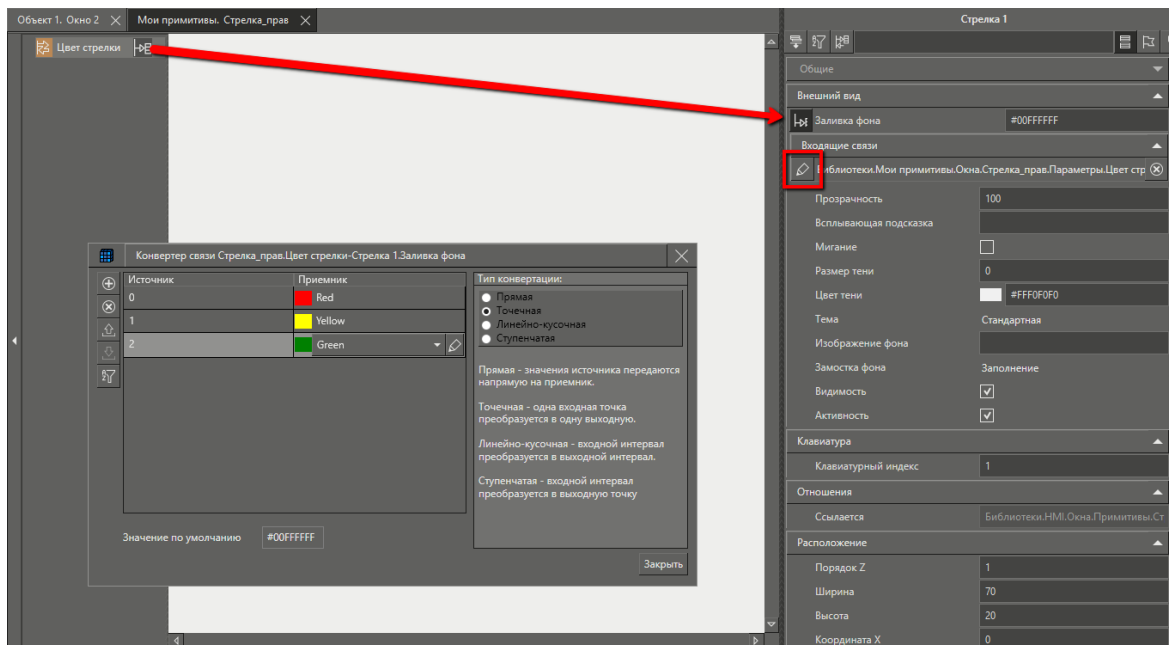


Шаг 7

Сохраняем библиотеку:



После этого можно добавлять элементы из пользовательской библиотеки в окна:

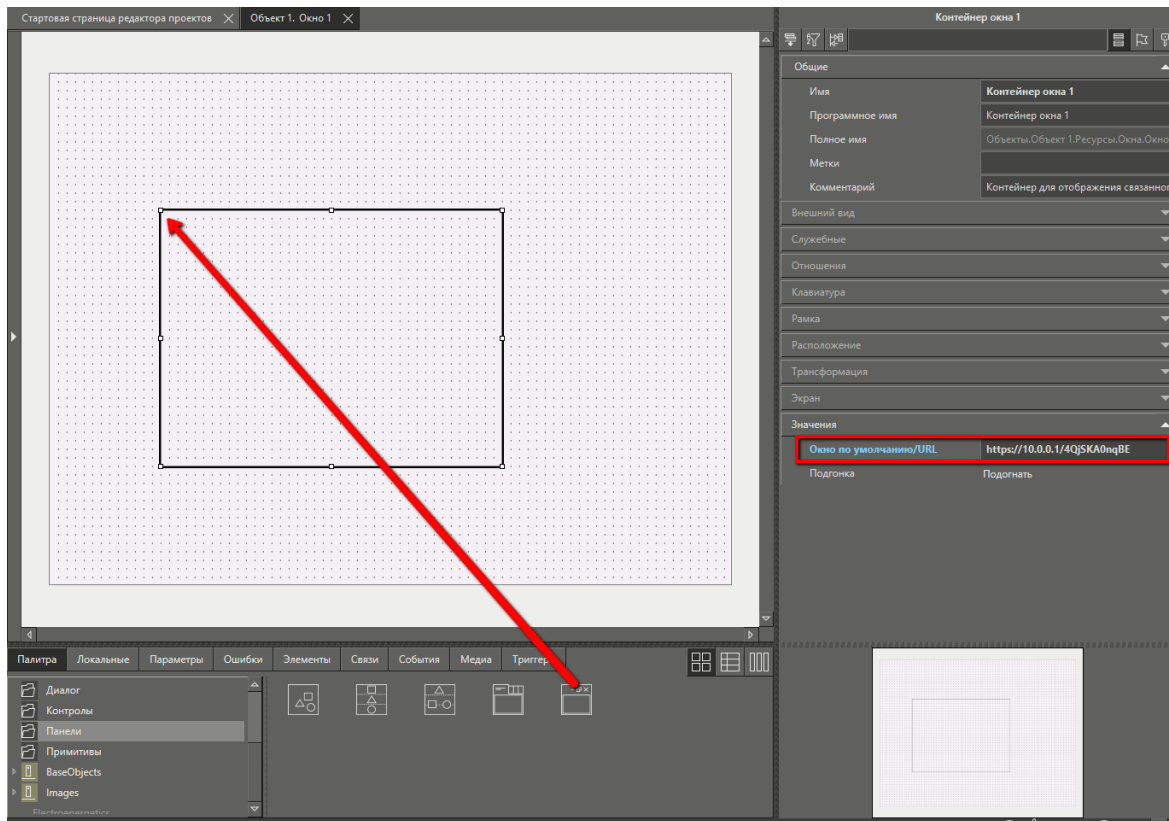


Параметр, добавленный в библиотечное окно, отображается как свойство элемента. Если установить связь между этим свойством и каким-либо параметром проекта, то в режиме исполнения стрелка будет менять свой цвет.

8.5.8. РАБОТА С ВЕБ-КАМЕРОЙ

Для работы с MasterSCADA 4D используют веб-камеры, которые транслируют видео на веб-сайт.

В этом случае в окно добавляется элемент Контейнер окна, и в его панели свойств задается адрес веб-камеры.

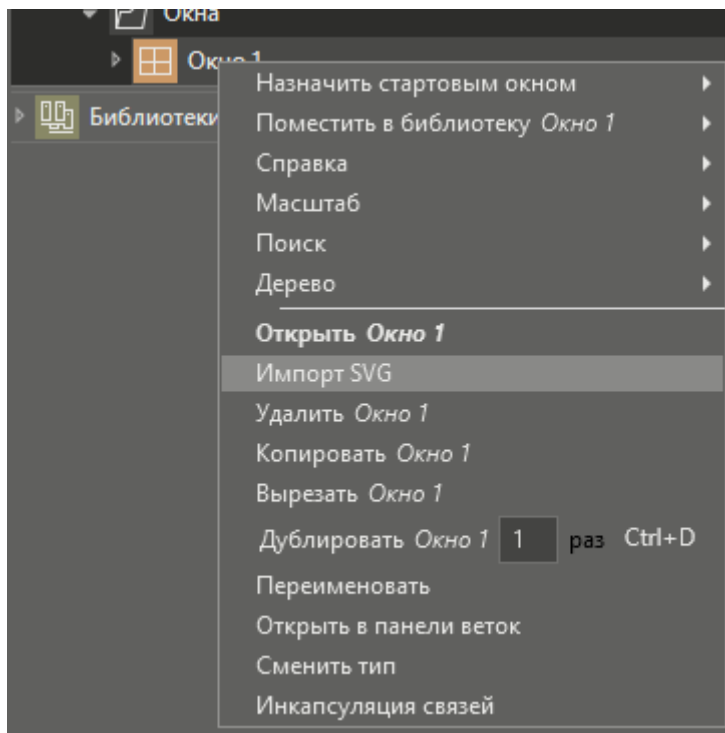


8.5.9. ИМПОРТ SVG В СРЕДУ РАЗРАБОТКИ MASTERSCADA 4D

Для создания окон в MasterSCADA 4D можно использовать сторонние SVG файлы с изображением. Рассмотрим порядок работы с такими элементами.

Сначала необходимо в одну папку положить сам svg-файл и xml-файл, в котором содержится описание параметров, если такой имеется.

Затем в дереве Библиотеки MasterSCADA 4D нужно создать окно, в которое затем будет добавлен сторонний SVG-элемент и в контекстном меню созданного окна нужно выбрать пункт Импорт SVG:

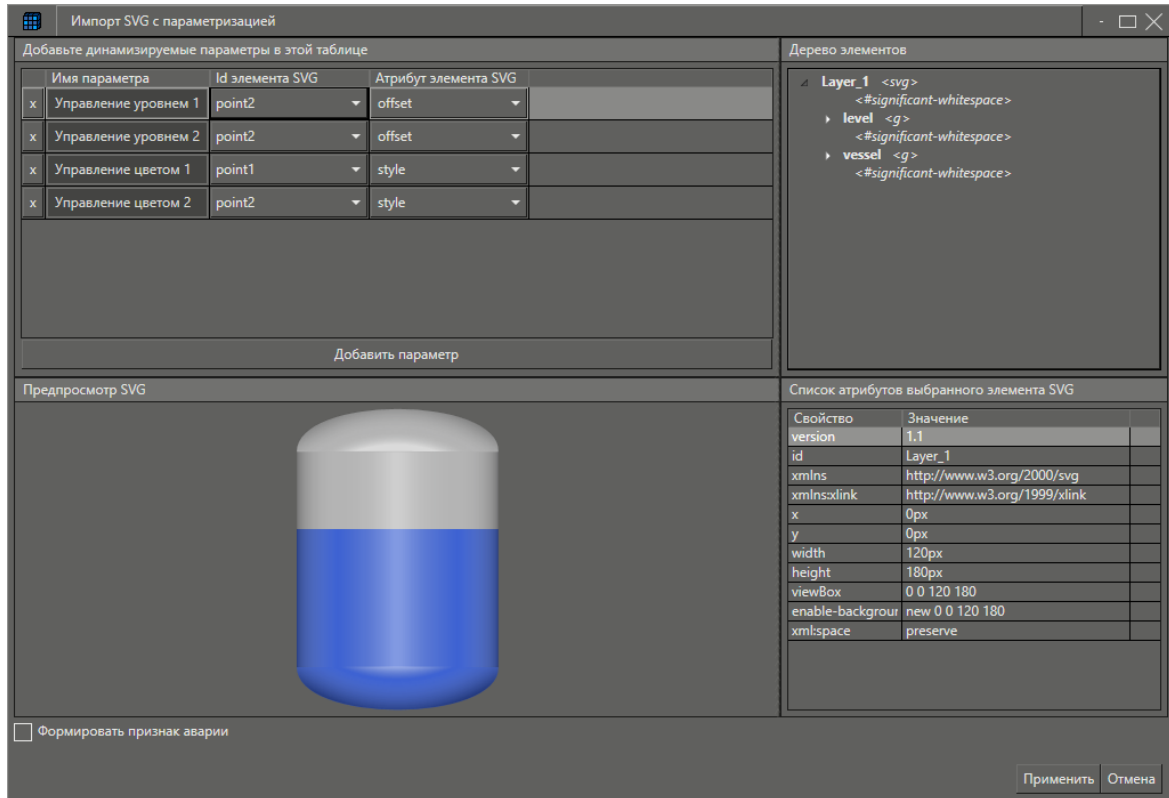


В открывшемся диалоговом окне можно выбрать SVG файл, тогда будет импортирован и он, и описание xml к нему, если оно имеется.

Если SVG файл был уже ранее добавлен в MasterSCADA 4D, то если выбрать файл xml, относящийся к этому SVG, то тогда в среде разработки обновится его описание. Если был выбран SVG файл и для него существует XML файл, то откроется диалоговое окно Импорт SVG с параметризацией.

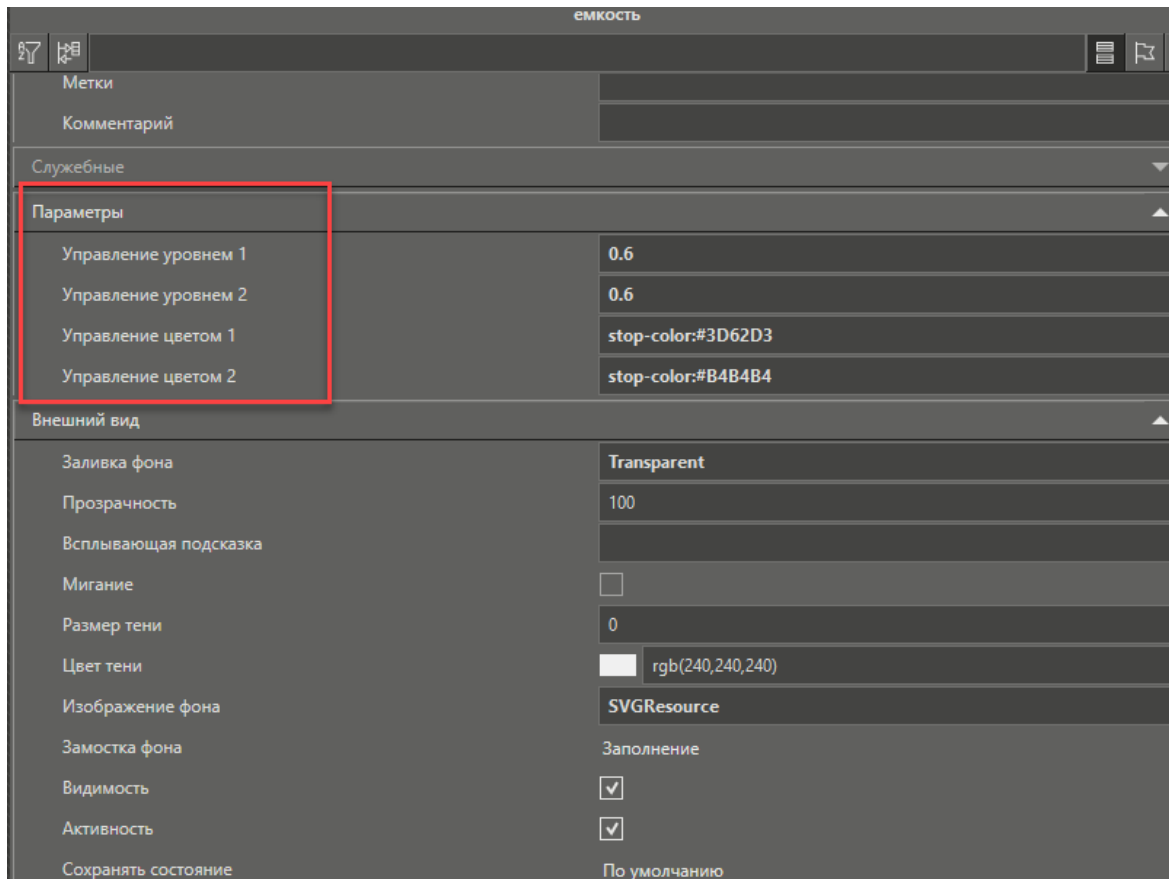
Импорт SVG с XML файлом

Вид окна Импорт SVG с параметризацией, если XML файл с описанием имеется:



В верхней левой части окна отобразятся параметры, прописанные в файле XML, которые можно динамизировать, т.е. изменять в режиме исполнения в зависимости от значений параметров в дереве или наоборот - менять значения параметров дерева в зависимости от значения параметров SVG. При необходимости их можно изменить, выбрав в таблице в выпадающем списке другое значение.

После нажатия на кнопку Применить, размер окна будет равен добавленному SVG, а в панели свойств будут параметры из таблицы:

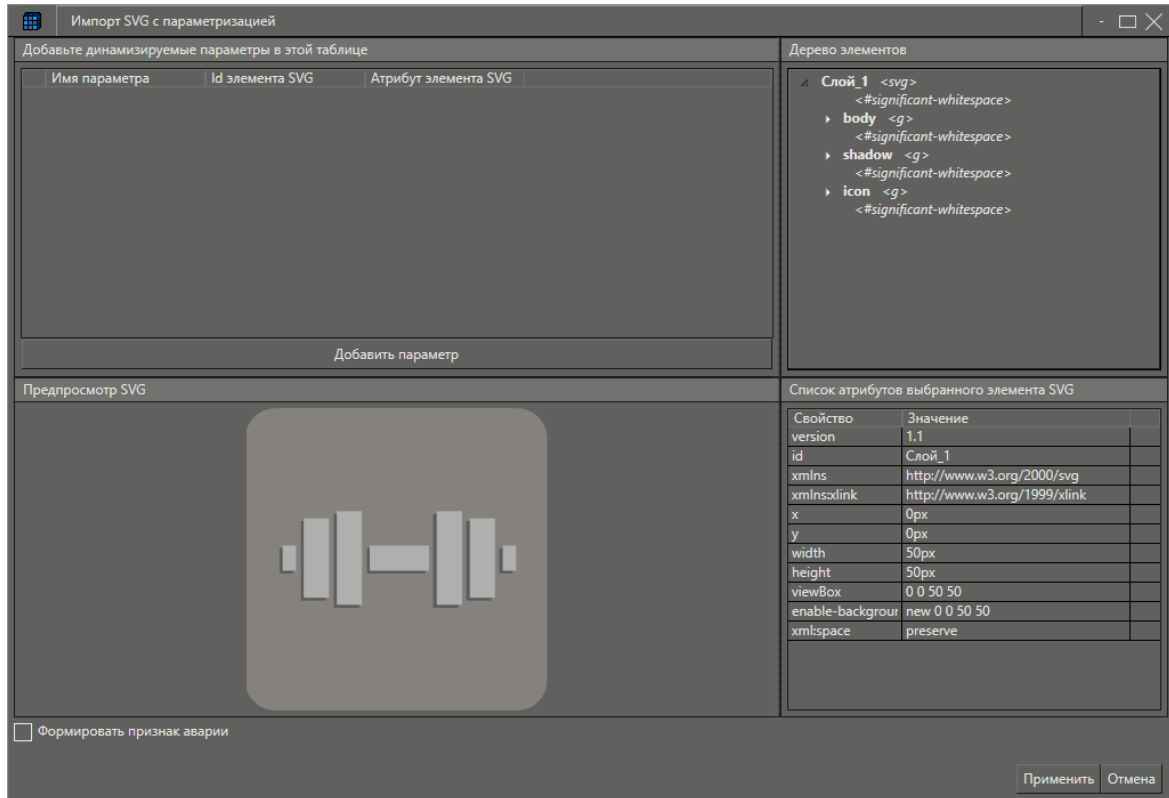


Если в окне импорта установить флаг Формировать признак аварии, то в категорию Параметры будет добавлено свойство Общая авария, динамизируя которое можно привлечь внимание пользователя в клиенте визуализации к элементу, в случае необходимости.

Импорт SVG без XML файла

Если XML файла с описанием параметров для динамизации нет, их можно добавить самостоятельно.

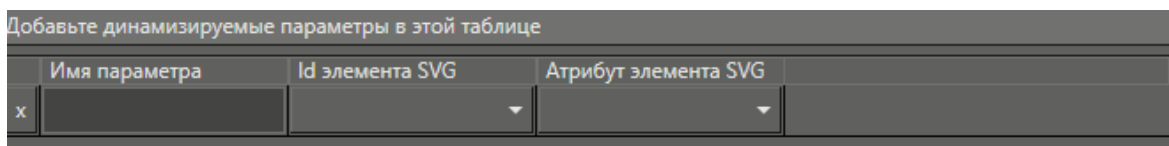
Вид окна Импорт SVG с параметризацией, если XML файл с описанием отсутствует:



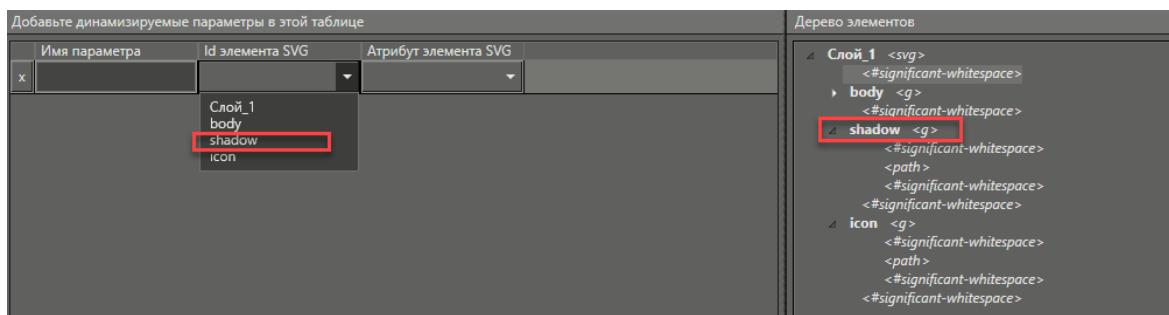
В поле **Дерево элементов** видно из каких частей состоит SVG.

Для добавления параметра, который можно будет динамизировать, нужно нажать на кнопку **Добавить параметр**.

В таблице появится строка, в которой можно выбрать ID элемента, атрибут и указать произвольное имя, которое будет отображаться в панели свойств:



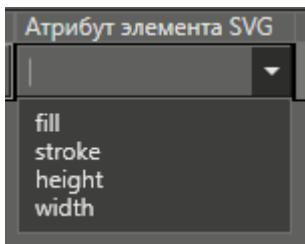
В выпадающем списке ID элемента SVG будут отображаться список составных частей элемента, которые были предусмотрены во время разработки SVG:



После выбора какого-либо ID, в поле Список атрибутов выбранного элемента SVG появится список атрибутов, которые характерны этой части SVG и были заданы разработчиком:

Список атрибутов выбранного элемента SVG	
Свойство	Значение
id	shadow
fill	#595758

Выпадающий список Атрибут элемента SVG содержит следующие строки:

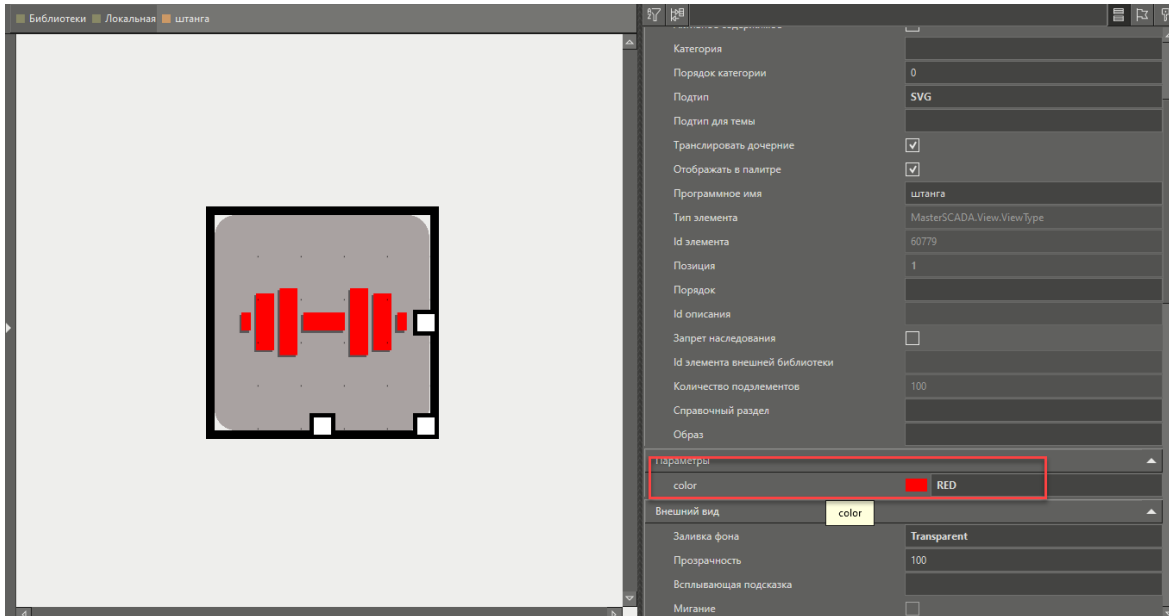


Название	Назначение
fill	Определяет цвет заливки части SVG
stroke	Определяет цвет контура части SVG
height	Определяет ширину части SVG
width	Определяет высоту части SVG

В списке отображаются самые популярные атрибуты.

Если атрибут отсутствует в списке, его можно добавить вручную. При клике мышью в поле атрибута появляется курсор и можно ввести любое значение атрибута, который может быть использован для выделенного типа фигуры (соответствие фигур и их атрибутов прописаны в стандарте SVG). Поддержаны все атрибуты стандарта SVG. (Рекомендуется ознакомиться с информацией на сайте <https://developer.mozilla.org/ru/docs/Web/SVG/Attribute>)

После нажатия на кнопку Применить в панели свойства появятся свойства, имена которых будут определены в таблице Имя параметра

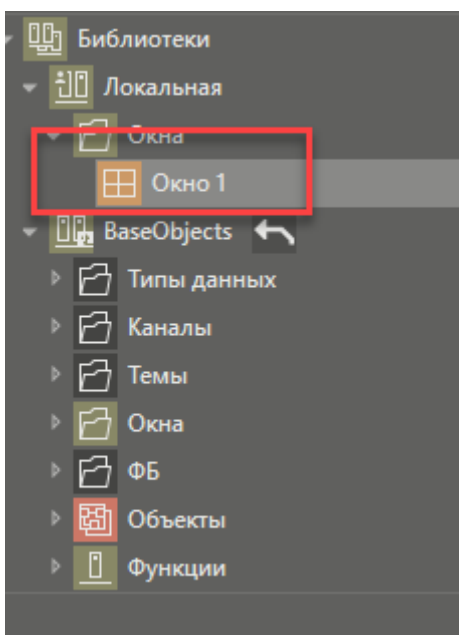


Важно! Если не получается динамизировать какой-либо атрибут для какого-либо ID, то возможно этот атрибут не поддерживается для таких фигур. Если ID присвоено сложной фигуре, то возможно ее стоит разделить на несколько более простых и каждой присвоить свой ID. Для работы с SVG элементами нужно знать основы стандарта SVG.

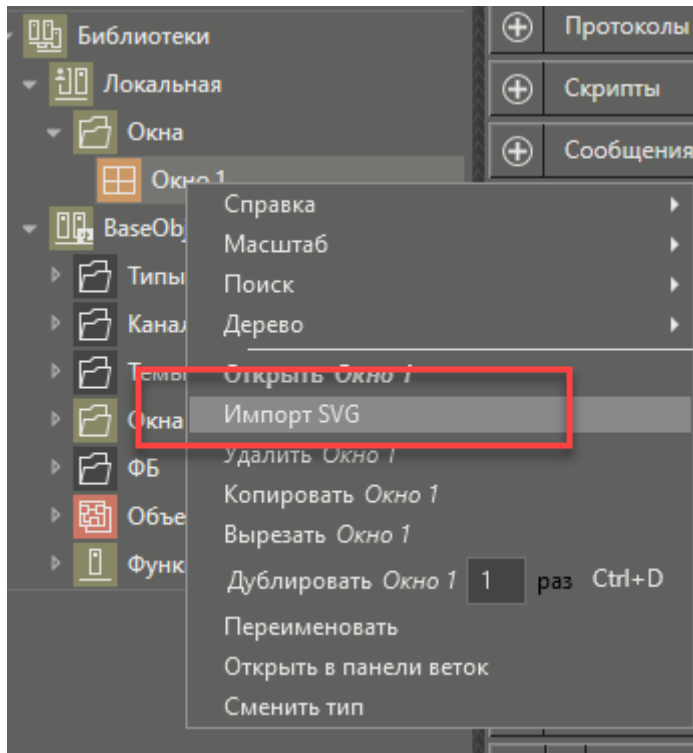
8.5.9.1. ПРИМЕР ИМПОРТА SVG

Рассмотрим пример импорта SVG файла без XML файла с описанием параметров в MasterSCADA 4D.

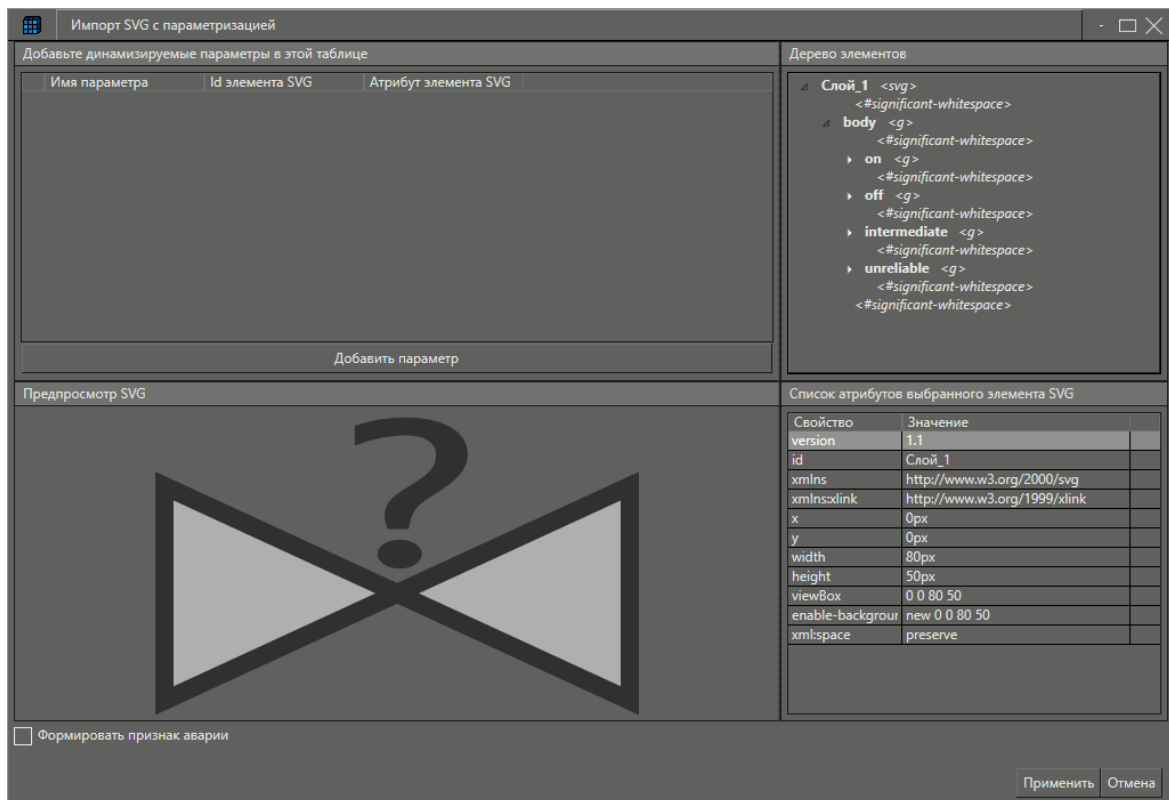
Добавим окно в пользовательскую библиотеку Локальная:



В контекстном меню выполним пункт Импорт SVG:



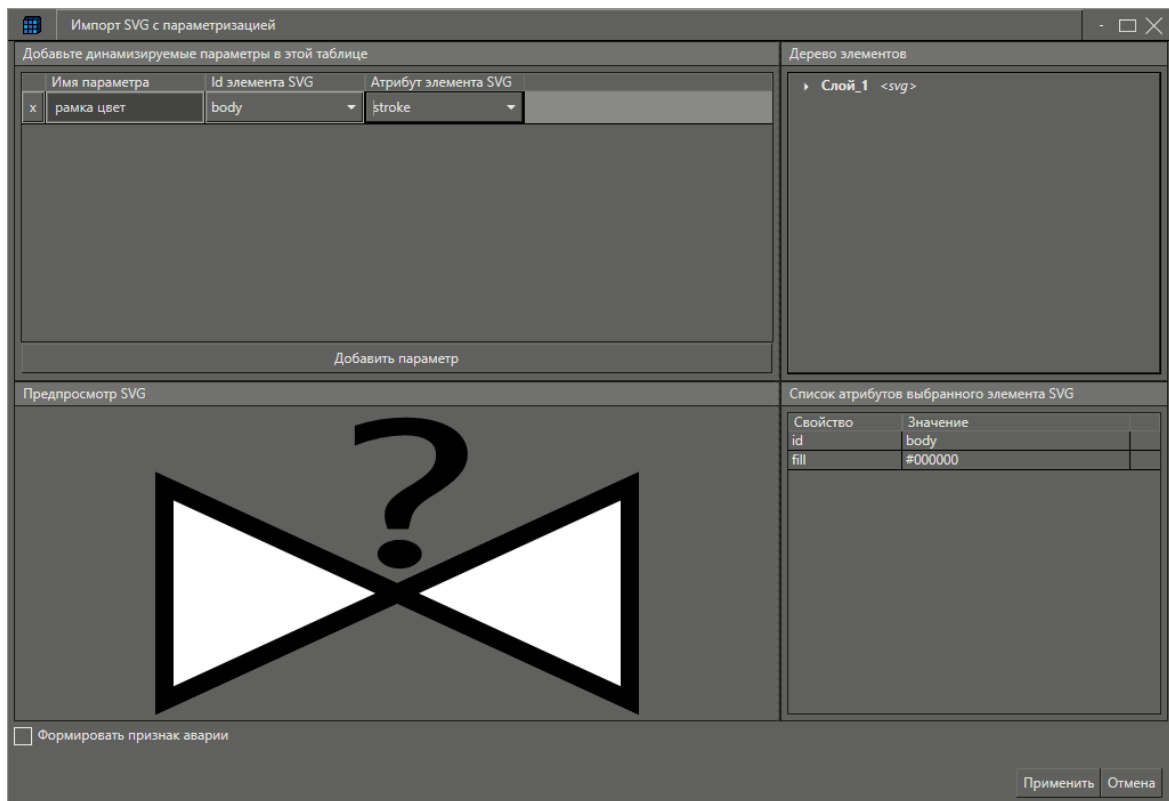
Откроется стандартное диалоговое окно Windows для выбора файла. После выбора файла с расширением SVG откроется окно MasterSCADA 4D:



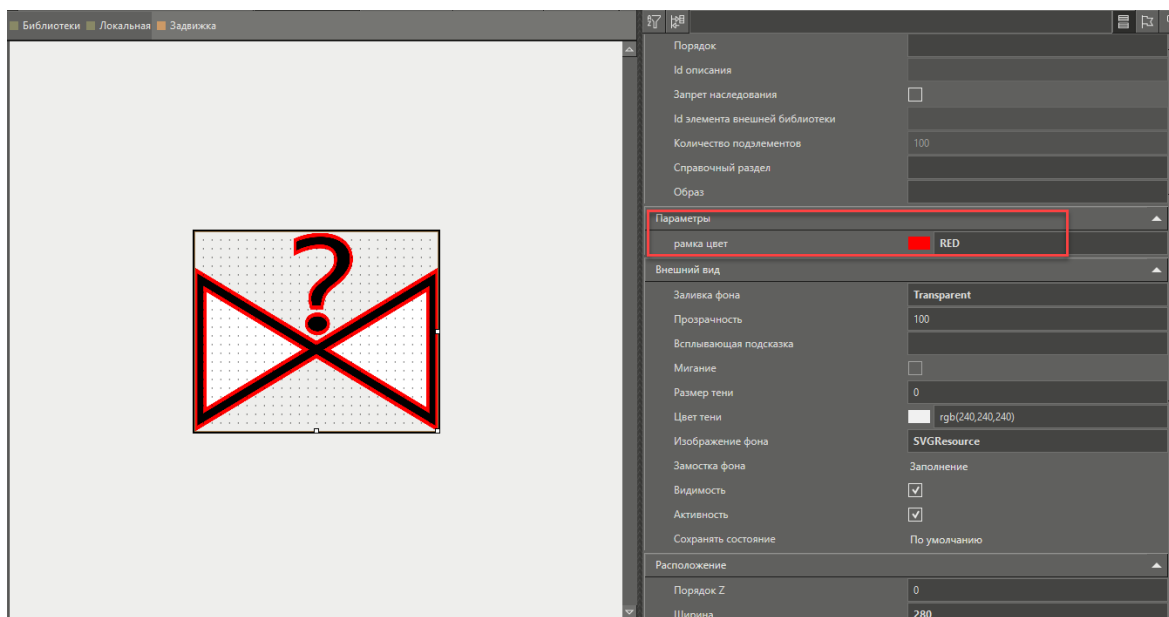
Нажмем кнопку **Добавить параметр**.

В таблице появится строка.

Заполним ее следующим образом:



Для ID элемента body атрибут stroke (цвет границы) не был прописан разработчиком, но это не значит, что его невозможно динамизировать в MasterSCADA 4D. После нажатия на кнопку Применить в созданном окне отобразится выбранный SVG файл, в панели свойств появится свойство, которое можно изменить. А также связать с каким-либо параметром проекта, для его изменения в режиме исполнения:



8.5.10. РАЗРАБОТКА SVG ФАЙЛА С ИЗОБРАЖЕНИЕМ

Для создания окон в MasterSCADA 4D можно использовать SVG файлы с изображением, разработанные самостоятельно в сторонних приложениях. Рассмотрим порядок разработки.

Общие требования

- SVG файл разрабатывается в стороннем графическом редакторе (например, Adobe Illustrator или Inkscape)
- Все элементы изображения должны входить в корневую группу (элемент g). Эта корневая группа должен иметь Id="root"
- Внутри корневой группы элементы должны быть сгруппированы в подгруппы для упрощения накладывания динамических фильтров. Требования по группировке должны быть оформлены в заданиях на конкретные элементы.
- У корневой группы должна быть возможность задавать стили, в соответствии с которыми меняется внешний вид всего контрола. Привязка стилей осуществляется через атрибут class корневой группы svg. Стили корневой группы могут комбинироваться (то есть одновременно применяться несколько классов от разных стилей). Названия классов должны быть осмысленными.
- Документ с SVG изображением должен иметь ширину/высоту по умолчанию, например равные 200*500 (или более удобное, но примерно такого порядка). Изображение должно полностью вписываться в область редактирования документа. По краям не должно быть пустых отступов.
- SVG может содержать в себе не одно цельное устройство, а несколько его частей. В этом случае все части должны иметь осмысленные Id, например:
 - Ротор (вентилятор) - Rotor
 - Двигатель (мотор) - Motor
 - Инвертер - Inverter
 - Шкала (прогресс) - Bar
- Так же должны быть осмысленными Id внутренних элементов контрола, к которым может быть предоставлен внешний доступ (то есть те, которые должны быть изменяемыми). Например, в контроле может быть текст, который прописан не в стиле, а в элементе SVG. Вот этот элемент должен иметь осмысленный Id для возможности программной обработки и программной смены данного атрибута в собственном редакторе. Другими словами - кастомизироваться программным путем должно все логически осмысленное, что только можно (цвета, текст и т.п.). Не все атрибуты всех элементов, а именно те, которые логично подстраивать в дальнейшем.

Разработка стилей

- Стили включаются в состав самого svg элемента в виде элемента style -

```
<style type="text/css">
```

```
<![CDATA[
```

```
....
```

```
]]></style>
```

- Каждый стиль должен содержать селектор, который может включать имя класса, к которому применяется стиль и/или Id элементов
- Для применения правил к вложенным элементам svg необходимо использовать селектор вида -

```
.имя_класса > #id_вложенного_элемента
```

- Переключение состояний элемента должно производиться набором классов у корневой группы SVG. Например, class="Running Avary NoReserve".

Данные классы должны быть описаны в следующем виде (для примера):

```
#Element.MotorAlarm {
filter:url('#alarm_filter');
}
```

```
#Element.Running #Rotor #Running {
display:block;}
```

Разработка файла с описанием параметров

Необходимо разработать XML-файл с описанием параметров SVG элемента, к которым может быть предоставлен внешний доступ.

Этот файл должен иметь то же имя, что и SVG файл и находится в той же папке.

Структура файла:

```
<?xml version="1.0"?>
```

```
<SvgDef xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
<ParameterDefs>
```

.... Здесь описания параметров SVG


```
</ParameterDefs>
```

```
</SvgDef>
```

В данном xml-файле требуется задать все параметры, которые указаны в техническом задании.

Формат описания параметра:

```
<ParameterDef DisplayName="..." ParamType="..." ItemId="..." Path="..." Type="...">
```

...

здесь может быть задан список значений параметра, если параметр является перечислимым (описано ниже)

...

```
</ParameterDef>
```

Если список значений не задается, то описание упрощается -

```
<ParameterDef DisplayName="..." ParamType="..." ItemId="..." Path="..." Type="..." />
```

- `DisplayName` - имя параметра для пользователя, например “Авария вращения ротора”
- `ItemId` - идентификатор элемента SVG, прописанный в его `Id`, например “Element”
- `Path` - атрибут данного элемента, который отвечает за значение данного параметра. Например: “class” для установки класса или “width” для установки ширины. Если параметр определяет внутреннее содержимое элемента (его контент), то “content”
- `ParamType` - указывает на то, где следует искать элемент с данным ID - среди стилей или среди элементов SVG. Может принимать значения:
 - “attribute” - устанавливается значение атрибута элемента
 - “style” - устанавливается значение элемента стиля. В этом случае в `Path` задается имя устанавливаемого в стиле атрибута. Например, для стиля

```
#Element #Bar #inner{  
    fill:#25d22b;  
}
```

`Path` будет равно “fill”

- `Type` = тип значения параметра. Определены следующие типы:

- “BOOL” - булевское значение да/нет (true/false)
- “STRING” - строковое значение, например текст или значение класса
- “LREAL” - числовое дробное значение
- “HMI.SolidColorType” - сплошной цвет
- HMI.MatrixColorType - для матрицы цветового фильтра

Примеры оформления цветовых параметров

```
<ParameterDef DisplayName="Цвет аварии" ParamType="attribute" ItemId=Rotor" Path="fill"
Type="HMI.SolidColorType">
```

Если параметр имеет фиксированное число значений, например значений класса, то для него следует описать эти значения в виде:

```
<ParameterDef DisplayName="..." ParamType="..." ItemId="..." Path="..." Type="...">
```

```
<Value DisplayName="Выкл"></Value>
```

```
<Value DisplayName="Вкл">RotorAlarm</Value>
```

```
</ParameterDef>
```

где каждый элемент <Value> описывает одно значение. Этот элемент имеет следующие атрибуты:

- DisplayName - пользовательское название значения, например “Вкл”
- Content - непосредственное значение элемента в SVG, например RotorAlarm (название класса в данном случае)

Далее созданный SVG можно импортировать в MasterSCADA 4D

Важно! Созданные SVG элементы могут содержать в себе ошибки. В случае возникновения неисправности в службу технической поддержки нужно предоставить файл в SVG, в работе которого обнаружилось нежелательное поведение.

8.5.11. ИСПОЛЬЗОВАНИЕ ТЕМ

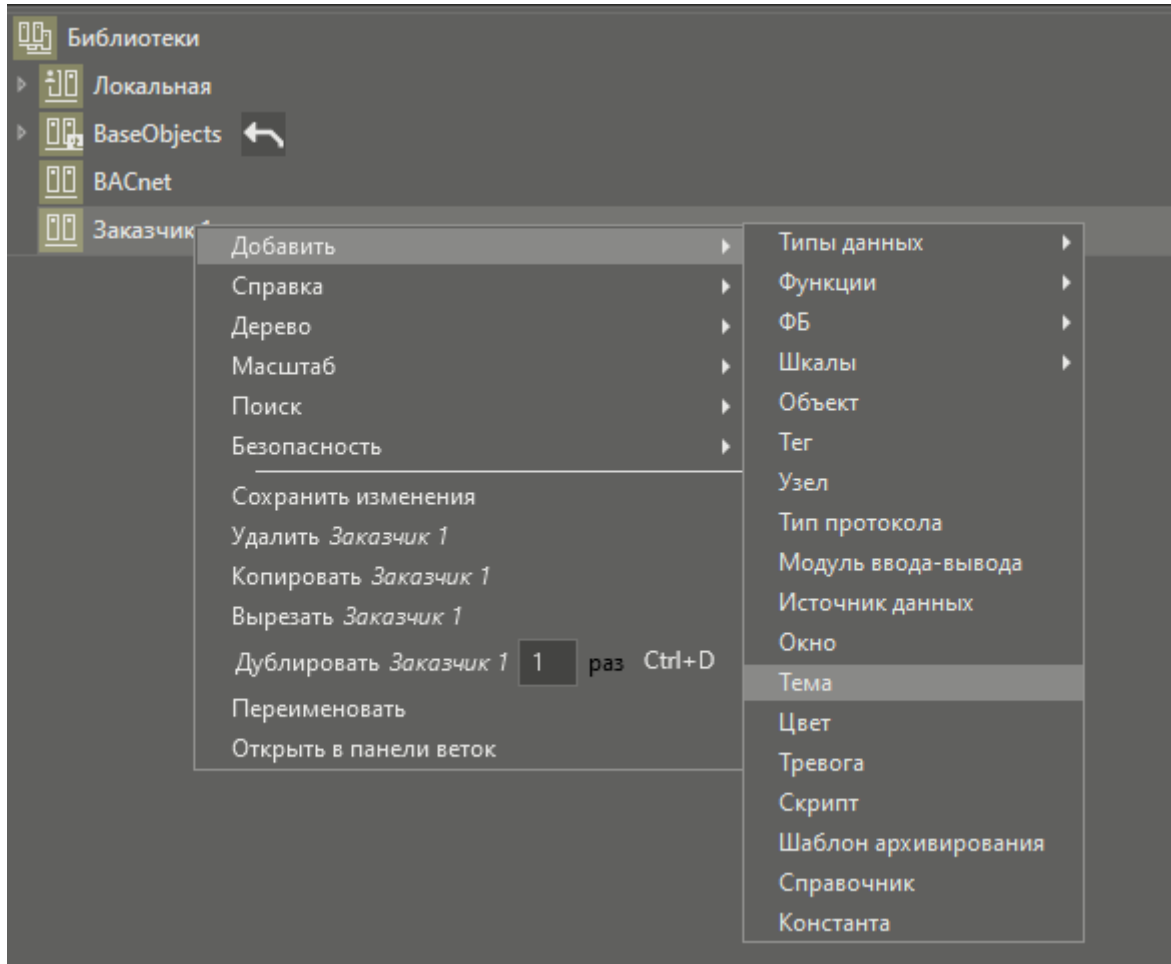
Тема это набор свойств элементов, которые могут быть применены в любой момент разработки проекта.

Например, по стандартам, принятым у разработчика проекта рамка элементов должна быть 3 pt, цвет фона текстов должен быть голубой, разработчик может создать тему, куда добавить все нужные элементы, и выполнить необходимые настройки. И применить тему в настройках проекта. Изменение темы приведет к изменению внешнего вида всех элементов, указанных в теме, например, в случае изменения требований заказчика.

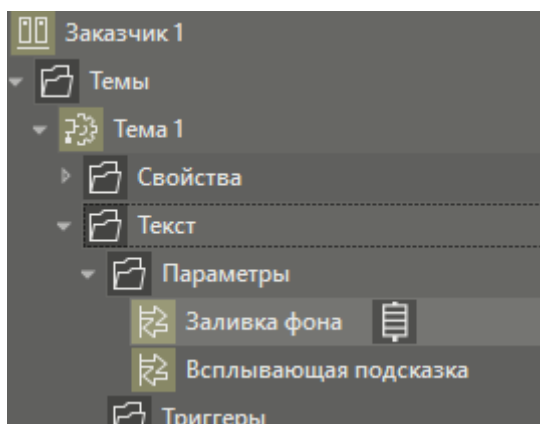
Можно использовать темы, которые по умолчанию присутствуют в библиотеке. Разработчик проекта может создать свою тему в пользовательской библиотеке. По умолчанию библиотека Локальная имеет тему Стандартная.

Пример

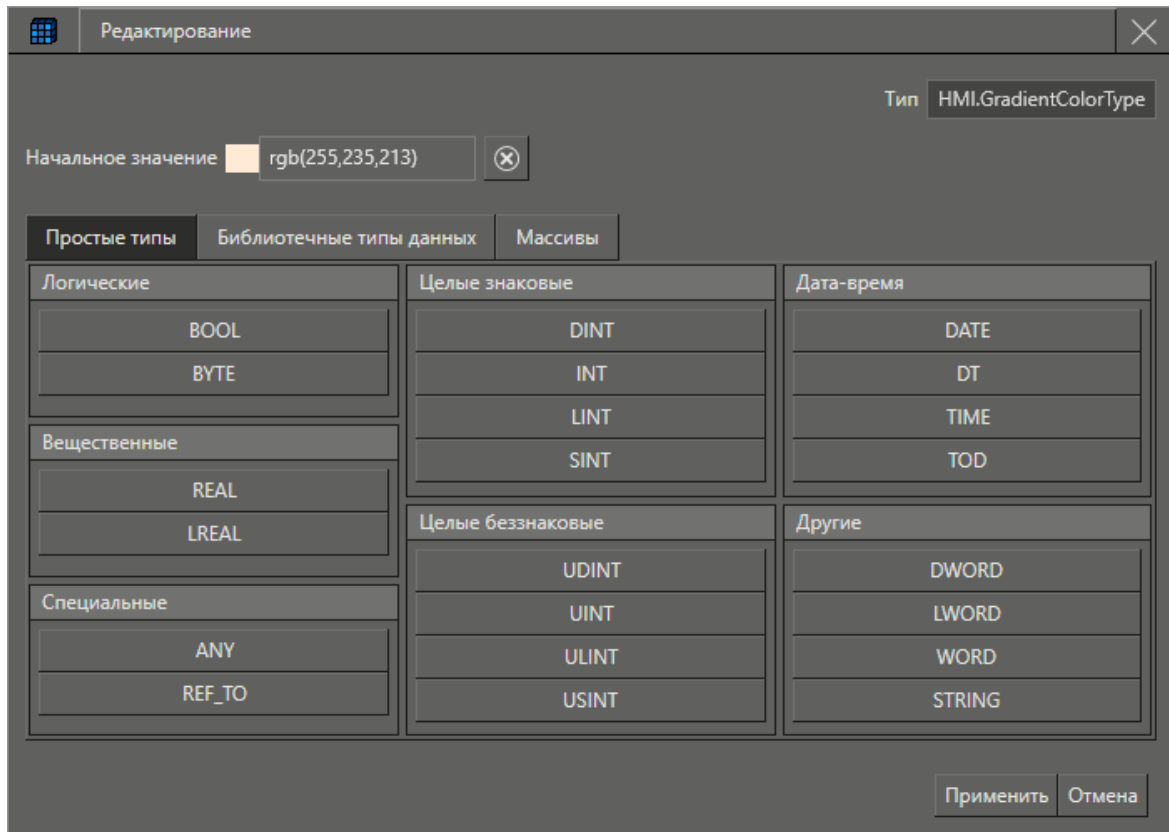
Добавим новую библиотеку Заказчик 1 и в нее новую тему:



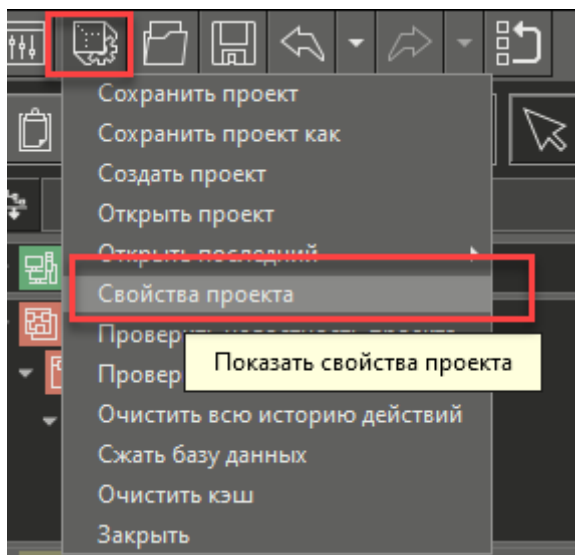
В тему добавим элемент Текст и в него параметр Заливка фона:



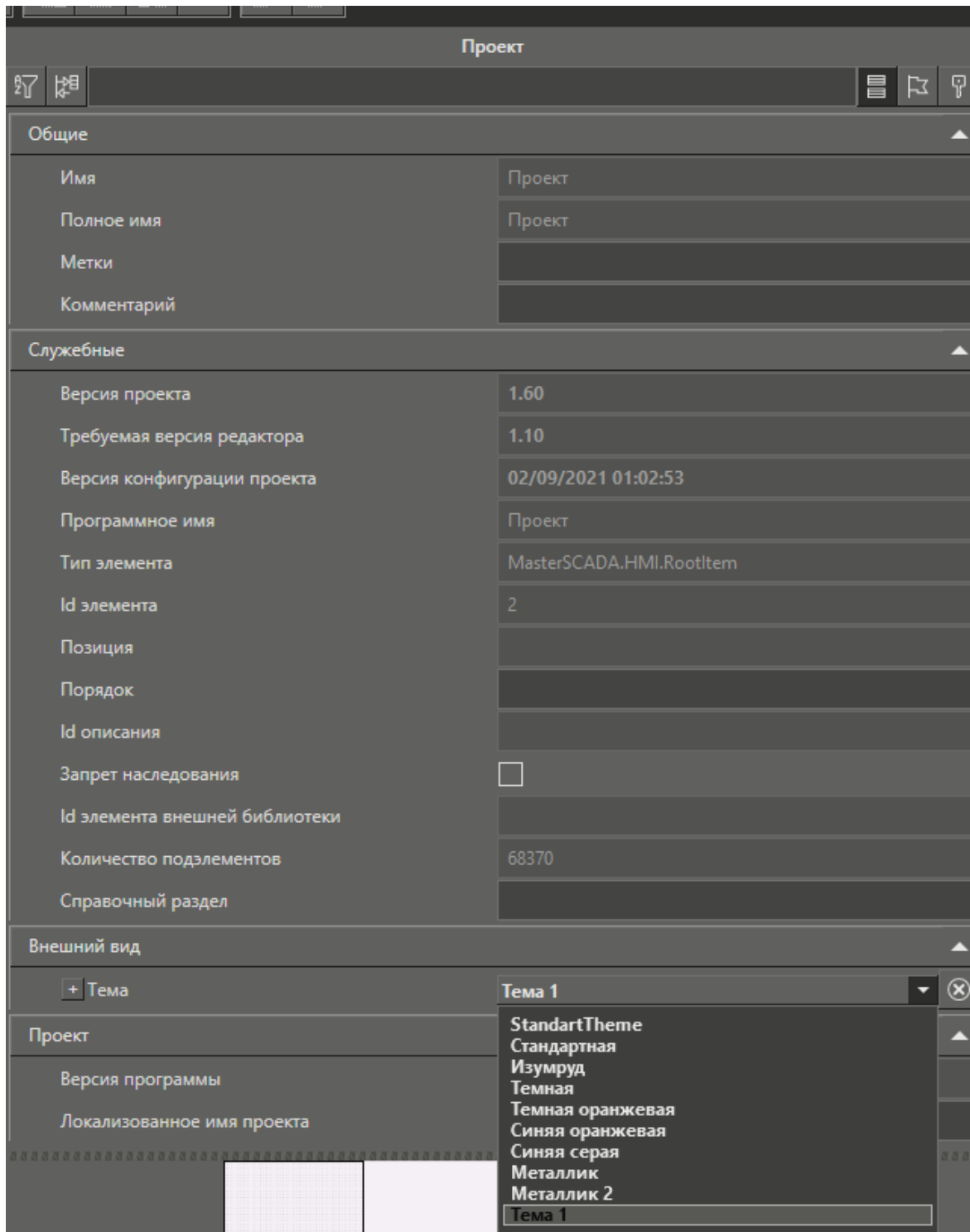
При добавлении параметра установим его значение:



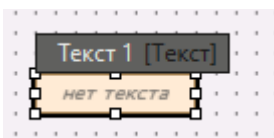
Затем вызовем свойства проекта:



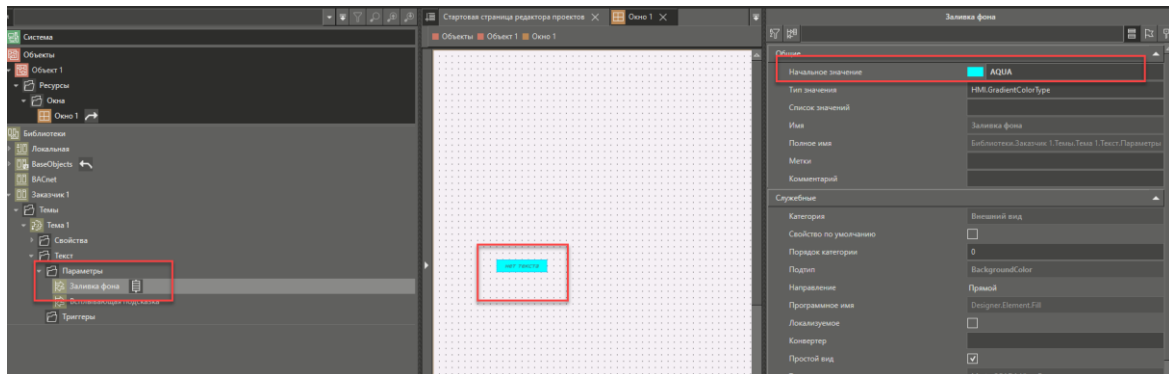
В панели свойств отобразятся свойства проекта, где нужно выбрать созданную тему:



Тогда добавляемые элементы Текст в окно будут сразу же иметь заданный ранее фон:



При изменении параметра темы в библиотеке будет изменен и ранее добавленный элемент, но только в том, случае, если разработчик проекта не менял это свойство, т.е. если используется значение свойства по умолчанию:



9. СОЗДАНИЕ ОТЧЕТОВ

MasterSCADA 4D имеет специальный генератор отчетов, который предназначен для формирования отчетов в режиме исполнения на основе данных из архива MasterSCADA 4D или других источников данных (сторонних баз PostgreSQL и MS SQL).

Важно! Модуль отчетов работает на исполнительных системах для ОС Windows, Linux. При использовании модуля в определенных контроллерах необходимо уточнить, поддерживается ли он в каждом конкретном типе.

Важно! В версии MasterSCADA 4D 1.2.12 есть возможность использовать технологию .NET Core вместо Node.JS. Рекомендуется ее использовать для проектов, которые будут работать в исполнительной системе под Windows. Включить ее можно в настройках среды разработки во вкладке Внешний вид

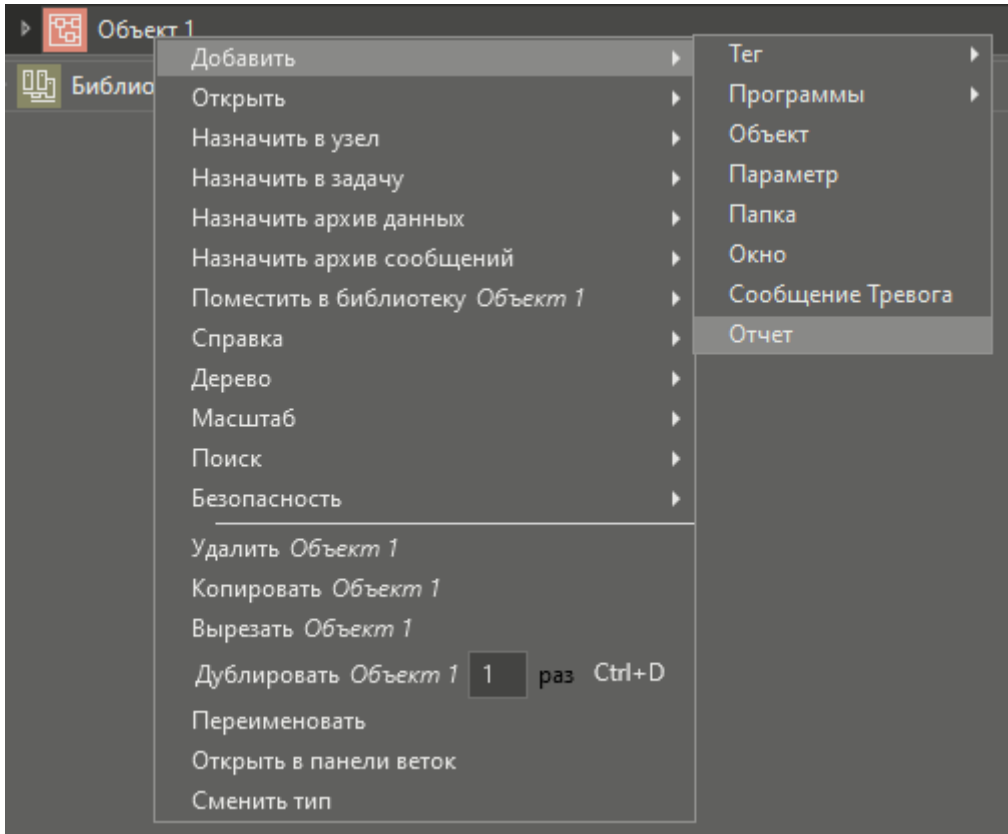
Можно создавать отчеты периодические, итоговые, отчеты по изменению значений. Добавлять в отчеты различные дополнительные элементы, как простейшие, например, номера страниц, имя и фамилию пользователя, при котором сформировался отчет, дату формирования, так и более сложные, например, штрих-коды, QR-коды, диаграммы и др.

Порядок создания отчета в MasterSCADA 4D:

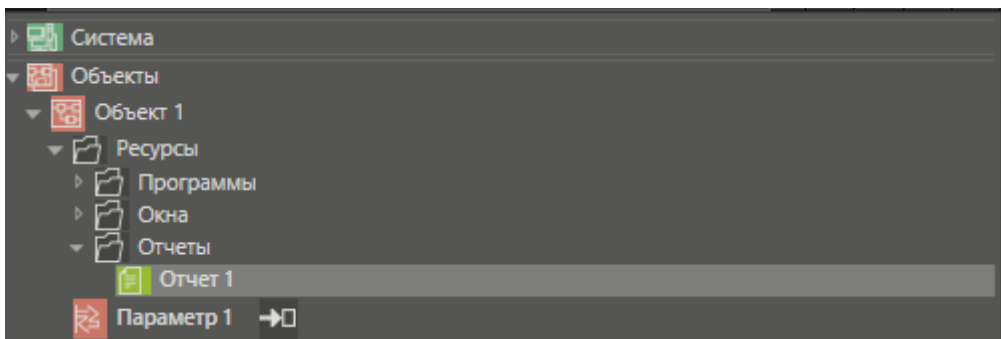
- Добавить отчет в дерево проекта, например, в объект или в узел
- Указать источник данных отчета
- Создать шаблон отчета, используя встроенный в редактор отчетов инструментарий
- Добавить в проект функциональный блок для управления созданием отчета и определить способы изменения его входов (программно или при помощи элементов управления в окне)
- При необходимости организовать отображение отчетов в окне.
- Проверить работу генератора отчетов в исполнительной системе.

Все основные этапы создания отчета описаны в разделе Пример создания отчета

Редактор открывается в центральной части окна MasterSCADA 4D автоматически после того, как в проект добавили элемент Отчет. Чтобы добавить отчет нужно воспользоваться контекстным меню узла или объекта:



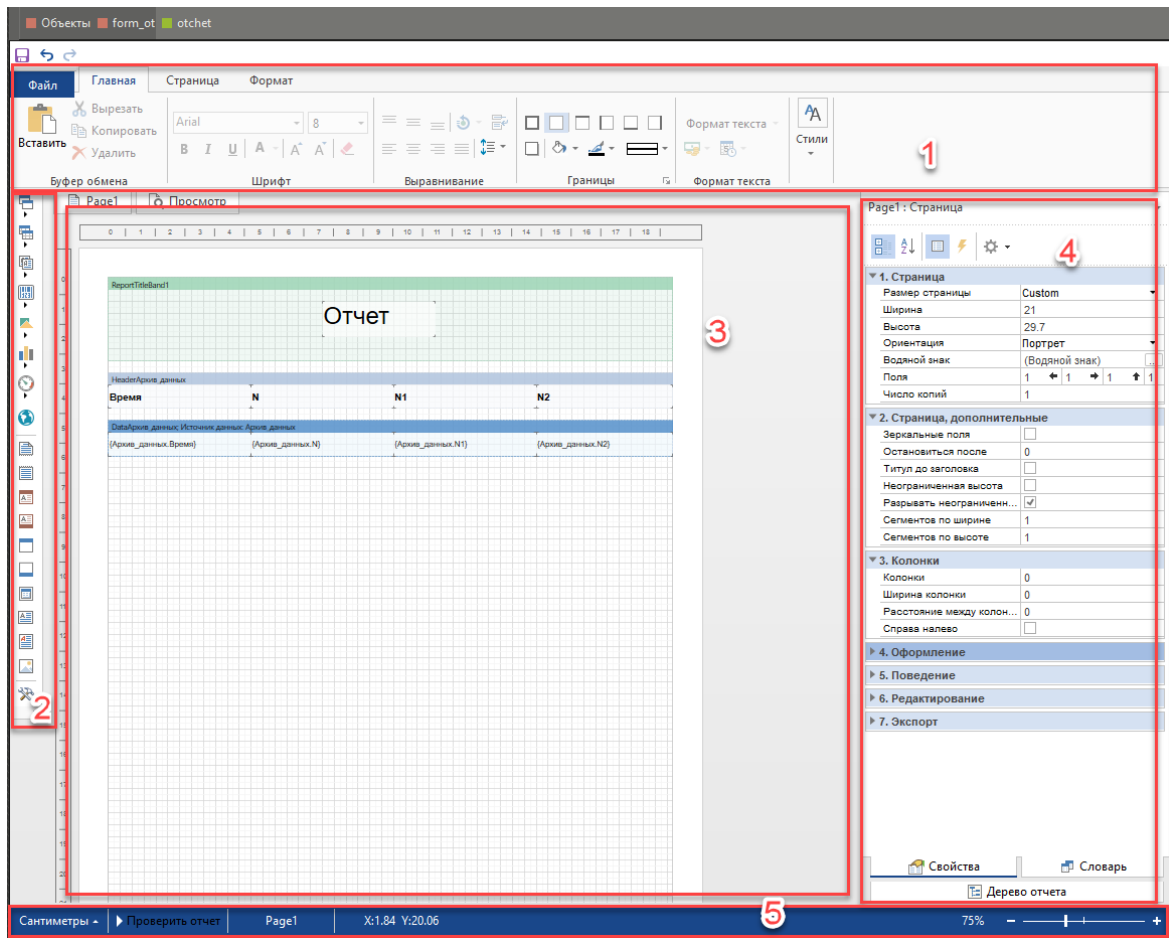
В дереве появится новый элемент и откроется редактор отчетов:



Открыть редактор для уже созданного ранее элемента проекта Отчет можно следующими способами:

- Дважды кликнуть по элементу Отчет в дереве
- Выделить элемент в дереве и нажать клавишу Enter
- Выполнить пункт контекстного меню элемента Редактировать...

Вид редактора отчетов:



Окно редактора отчетов состоит из следующих компонентов:

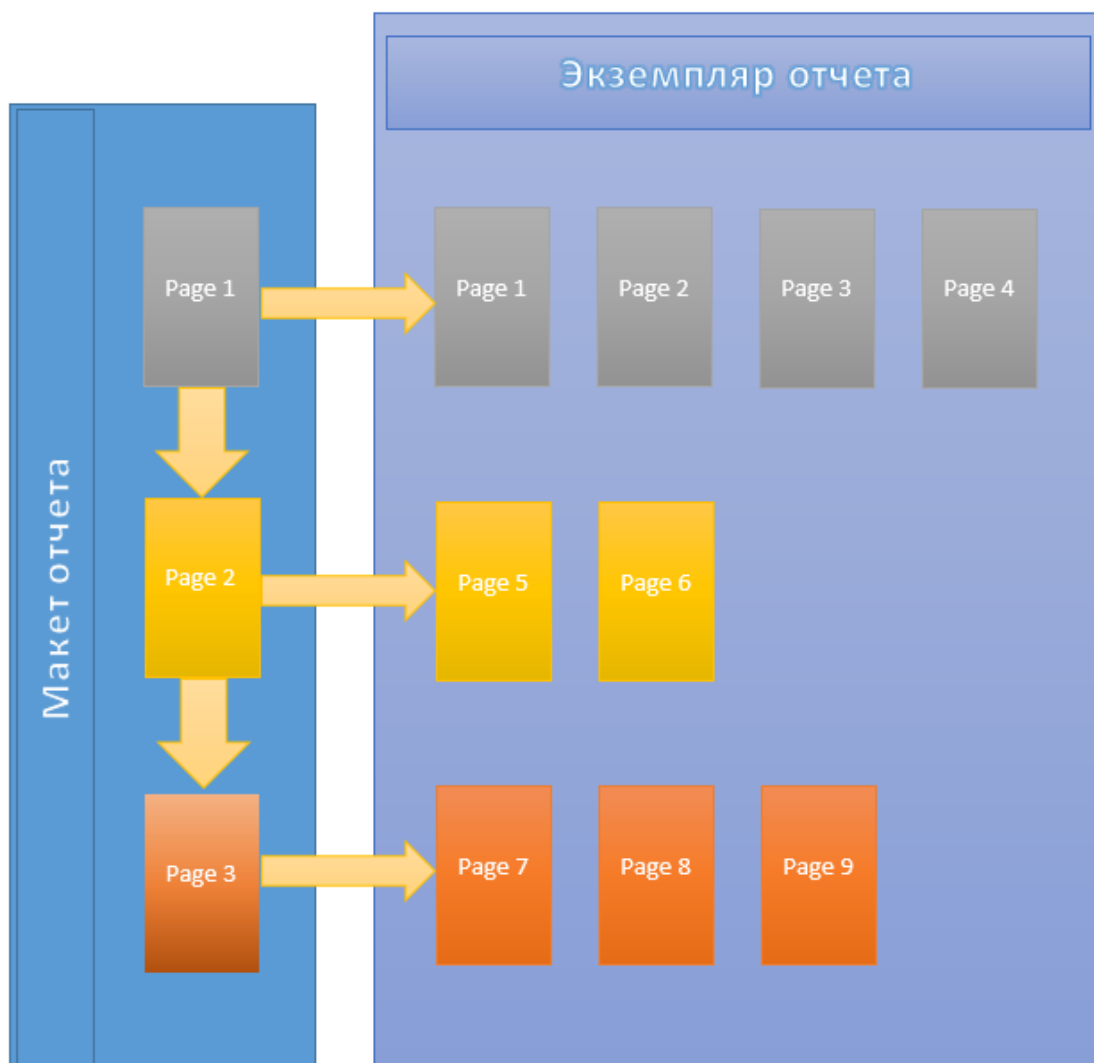
1. Панель форматирования - используя инструменты данной панели можно настроить внешний вид добавленных компонентов (изменить шрифт, заливку, включить границы и т.д.)
2. Инструментарий - в данной панели находятся компоненты отчета.
3. Рабочая область отчета (Страница) - в данной области создается шаблон будущего отчета.
4. Набор панелей, состоящий из трех панелей:
 - Словарь - в словаре располагаются источники данных, из которых происходит формирование отчета.
 - Свойства - свойства текущего (выделенного) компонента
 - Дерево отчета - панель выводит структуру сформированного отчета в виде дерева.
5. Панель статуса - содержит дополнительные элементы управления отчетом, масштабом отчета его проверки.

Основные положения

В генераторе отчетов существует три понятия отчетов:

1. Шаблон отчета - отчет с привязкой к переменным MasterSCADA.
2. Прототип отчета - отчет без привязки к переменным MasterSCADA.
3. Экземпляр отчета - конечный вид отчета.

Прототип/шаблон отчета делится на страницы. Каждая страница может иметь свои размеры и поля печати. Все компоненты в отчете располагаются на страницах. При построении отчета генератор последовательно обрабатывает все страницы отчета:




Все элементы шаблона отчета делятся на две категории: компоненты и контейнеры. Принципиальное отличие контейнера от компонента в том, что в контейнер можно вложить компонент или ещё один контейнер, в компонент вложить нельзя ничего. К примеру, к компонентам относится элемент Текст (Text). Текстовый компонент располагается на странице либо на другом контейнере. Но расположить какой-либо контейнер или компонент в текстовом компоненте нельзя. Контейнером, к примеру, может выступать Форма. Т.е. на любую форму можно разместить ещё какой-либо компонент или контейнер.

В отчеты приняты пять основных понятий:

1. Страницы (Page)
2. Бэнды
3. Контейнеры
4. Компоненты для отображения информации
5. Элементы управления

Комбинация представителей этих групп и образует отчет. Каждый представитель обладает набором свойств, событий и методов, которые отображаются в панели Свойства редактора отчетов. Изменяя свойства и задавая события, можно изменять поведение той или иной части отчета. Пример панели Свойства редактора отчетов:

HeaderАрхив_данных : Заголовок данных ▾



▼ 1. Разрыв страницы или колонки	
Новая страница до	<input type="checkbox"/>
Новая страница после	<input type="checkbox"/>
Новая колонка до	<input type="checkbox"/>
Новая колонка после	<input type="checkbox"/>
Новая, если больше чем	100
Пропустить первого	<input checked="" type="checkbox"/>

▼ 2. Расположение	
Высота	0.8
Макс. высота	0
Мин. высота	0

▶ 3. Оформление

▶ 4. Поведение

▶ 5. Редактирование

Все свойства и события сгруппированы в категории. Основные категории свойств:

Название	Назначение
Расположение	Положение компонента на странице определяется координатами X, Y его левого верхнего угла. Координаты X, Y и размер компонента описываются свойствами: Лево, Верх, Ширина и Высота. Значения задаются в текущей единице измерения, которая указана в панели статуса.
Поведение	Свойство Доступность позволяет включить обработку компонента при построении отчета. Если свойство равно Нет, то компонент будет полностью проигнорирован. Свойство Печатать, принимающее значение Да или Нет, определяет, будет ли компонент выведен на печать. Значение Нет указывает, что компонент не будет выведен на печать, при этом в окне предварительного

	просмотра он будет отображен. Значение Да указывает, что компонент, отображаемый в окне предварительного просмотра отчета, будет выводиться на печать. Свойства Может расти и Может сжиматься позволяют автоматически управлять изменениями размеров компонента в зависимости от его содержимого. Свойство Режим смещения позволяет избежать наложения компонентов при использовании средств автоматического изменения размеров компонента.
Редактирование	Название компонента задается свойством Наименование - это программное имя компонента. Если в свойстве Псевдоним указывается какой-либо текст, то везде, вместо наименования компонента будет отображаться этот текст. Свойство Глобализация используется при построении локализованных отчетов. Значение Да свойства Замок фиксирует положение и размер компонента. Рекомендуется использовать кнопку на панели форматирования на вкладке Формат для управления свойством Замок выделенных компонентов. Свойство Связь позволяет привязать компонент к контейнеру, в котором находится компонент. Если оно установлено в значение Да, то компонент не может переместиться в другой контейнер. Например, если привязанный компонент перетащить с одной секции на другую, привязанный компонент все равно будет печататься вместе с первой секцией. Рекомендуется использовать кнопку на панели форматирования на вкладке Формат для управления свойством Связь выделенных компонентов

Интерфейс дизайнера предоставляет пользователю огромный набор инструментов, компонентов и средств для разработки отчетов, их визуального оформления и предварительного просмотра. В данной теме будут рассмотрены основные сведения о редакторе отчетов, его основные элементы, горячие клавиши, панели, линейка инструментов, главное меню и т.д.

9.1. ПРИМЕР СОЗДАНИЯ ОТЧЕТА

Постановка задачи

Имеем контролируемые параметры, на основании которых нужно построить отчет. Оператор должен иметь возможность выбрать время начала составления отчета, время окончания, а также интервал, с которым будет появляться новая строка. Так, например, если стоит задача построить суточный отчет, с интервалом в 1 час, то оператору нужно задать время начала отчета, время окончания, которое отличается на 24 часа от значения предыдущего параметра, и интервал 60 минут. В этом случае в отчете должно появиться 24 строки. Значение, которое будет отображено в отчете будет зависеть от способа обработки архивных данных, например, может выбраться максимальное значение, за контролируемый час, или минимальное, или среднее, др. В данном случае отчет строится на основании

архивных данных, поэтому в проекте у контролируемых параметров должен быть установлен флаг Архивировать.

Важно! Модуль отчетов работает на исполнительных системах для ОС Windows, Linux. При использовании модуля в определенных контроллерах необходимо уточнить, поддерживается ли он в каждом конкретном типе.

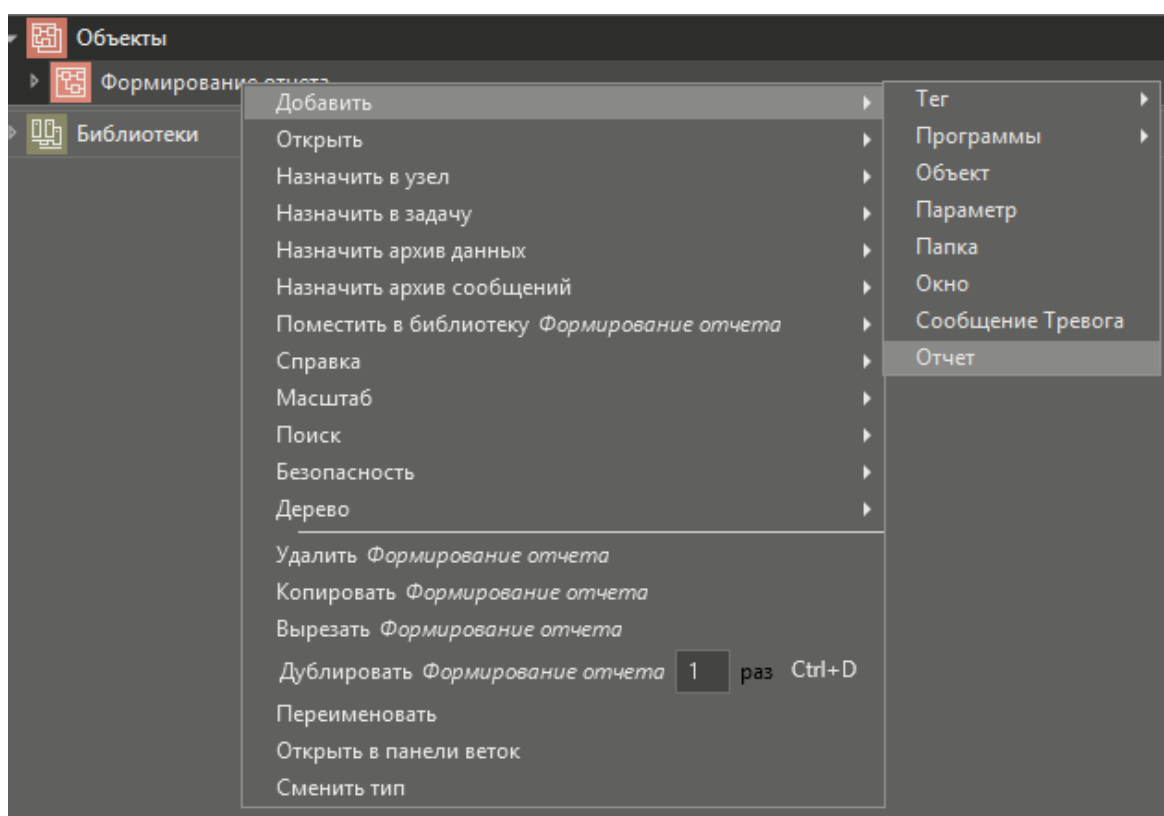
Реализация

Создание элемента отчет

Отчет может быть создан как в узле, так и в объекте. Рассмотрим создание отчета в объекте.

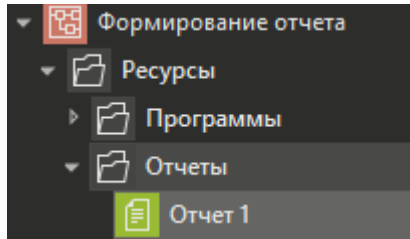
Чтобы добавить отчет в проект необходимо выполнить пункт контекстного меню объекта **Добавить.Отчет**.

Итак, создадим в проекте объект с именем **Формирование отчета**, в него добавим элемент **Отчет**:



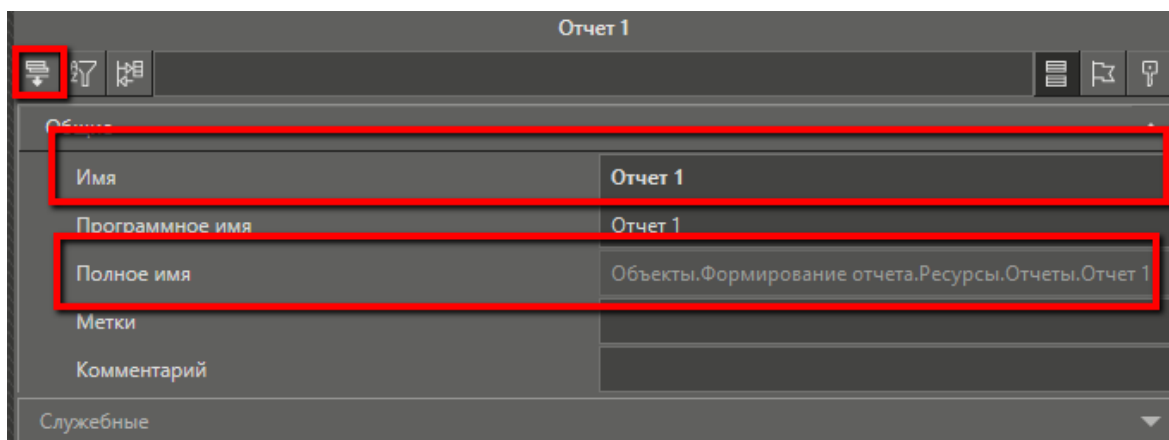
Редактор создания отчетов откроется автоматически.

В группе объекта **Ресурсы**, появится подгруппа **Отчеты**, в которой будет находиться созданный элемент.



При необходимости можно изменить имя отчета в его панели свойств.

Также стоит обратить внимание на свойство Полное имя, которое в будущем пригодится (оно отображается, если Простой режим в панели свойств отключен). Значение свойства Полное имя можно скопировать:

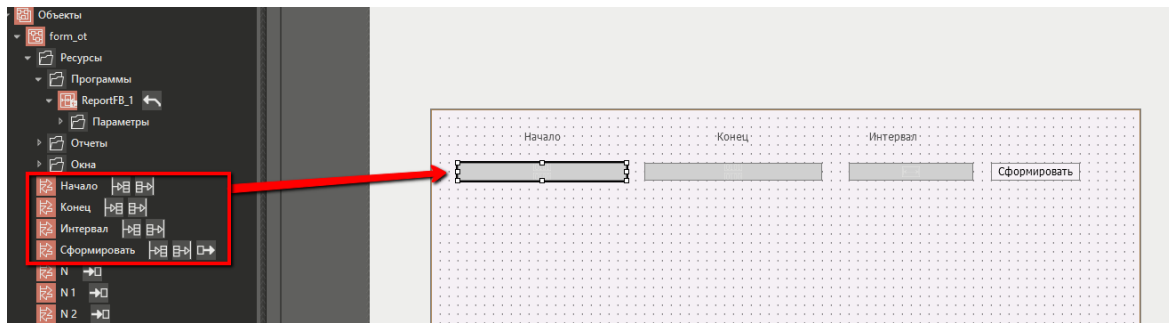


Параметры формирования отчета

В проект добавляем служебные параметры, необходимые для формирования отчета.

Название	Тип	Назначение
Начало	DT	Определяет время начала формирования отчета
Конец	DT	Определяет время окончания формирования отчета
Интервал	Time	Время определяющее через какой промежуток будет формироваться новая строка или столбец.
Сформировать	Bool	Команда оператора, после которой начнется формирование отчета.

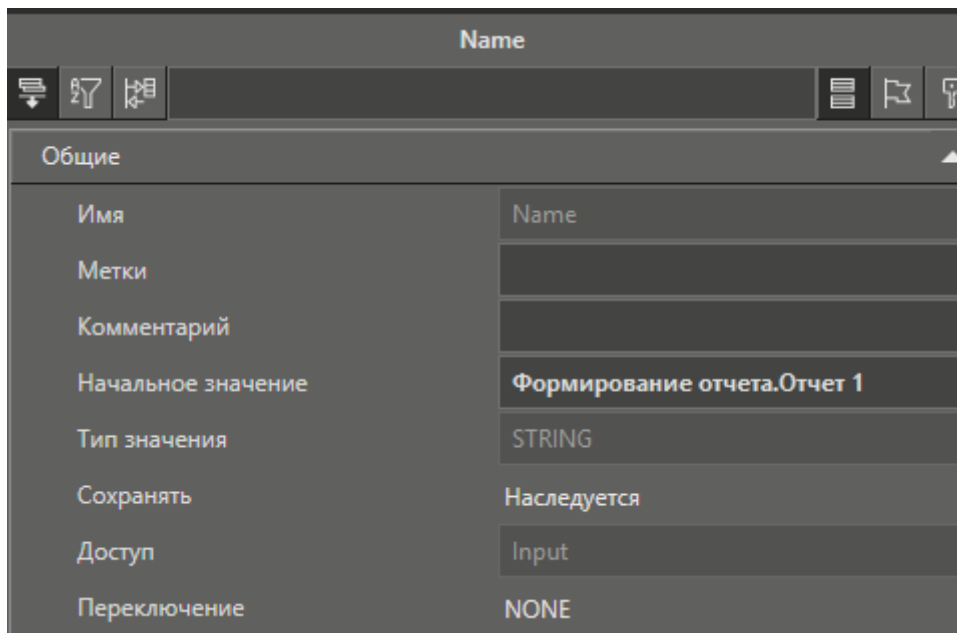
Параметры Начало, Конец, Интервал и Сформировать размещаем в окне, в удобном для оператора виде. Пример окна:



Если отчет должен формироваться автоматически, то данные параметры могут формироваться в результате работы программ, например, ST.

Управление формированием отчета

- Добавляем в проект служебный ФБ из библиотеки: BaseObjects.ФБ.Системные.ReportFB, который управляет формированием отчета.
- В настройках входа Name необходимо указать в качестве значения по умолчанию имя отчета, которым будет управлять данный ФБ. Имя отчета содержит в себе имя объекта и его родителей, в котором находится отчет, и имя самого отчета. Поэтому можно вставить ранее скопированное полное имя отчета и удалить оттуда несущественные элементы. Например, свойство Полное имя имеет значение: *Объекты.Формирование отчета.Ресурсы.Отчеты.Отчет 1*, нужно оставить - *Формирование отчета.Отчет 1*

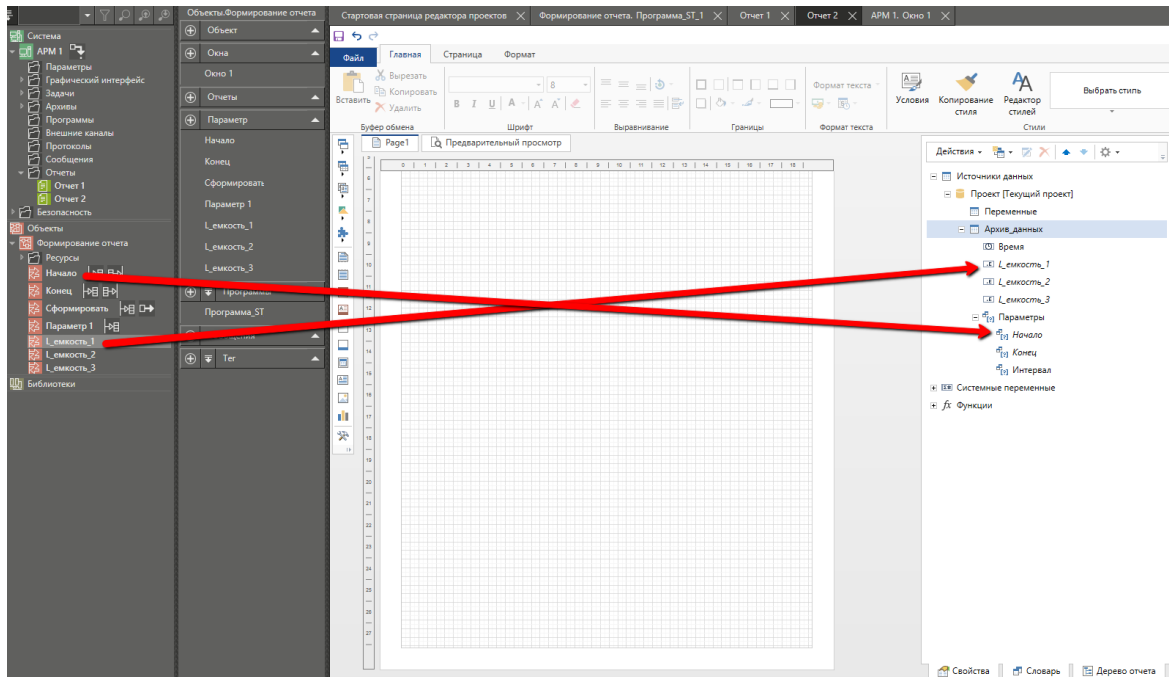


Если отчет создан в дереве системы, то в свойстве нужно указать только имя отчета. Имя узла, указывать не требуется.

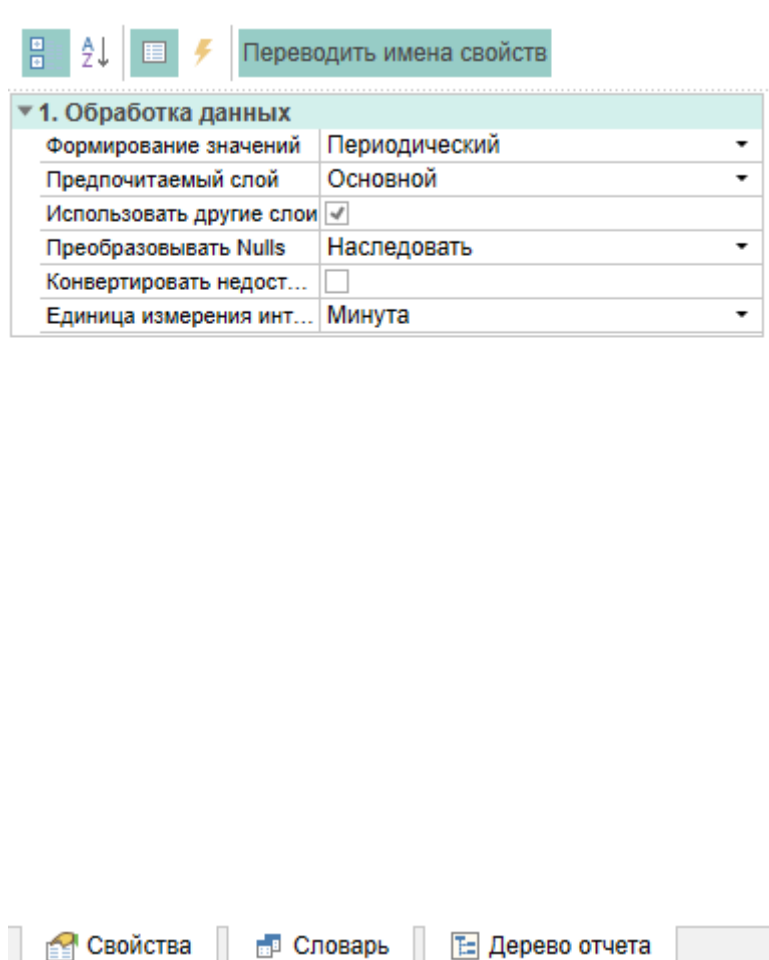
- Вход Execute соединить с параметром Сформировать

Редактор отчетов

- В редакторе отчетов, который открылся автоматически необходимо перейти на панель Словарь. В данной панели определяется какие параметры MasterSCADA будут участвовать при составлении отчета. В описываемом случае необходимо настроить группу словаря Архив данных.
- Перетаскиваем контролируемые параметры в группу Архив данных - появятся ссылки на соответствующие элементы. Затем необходимо связать элементы из подгруппы Параметры с соответствующими элементами дерева объектов. Для этого нужно переменные из дерева объектов перетащить на нужный параметр

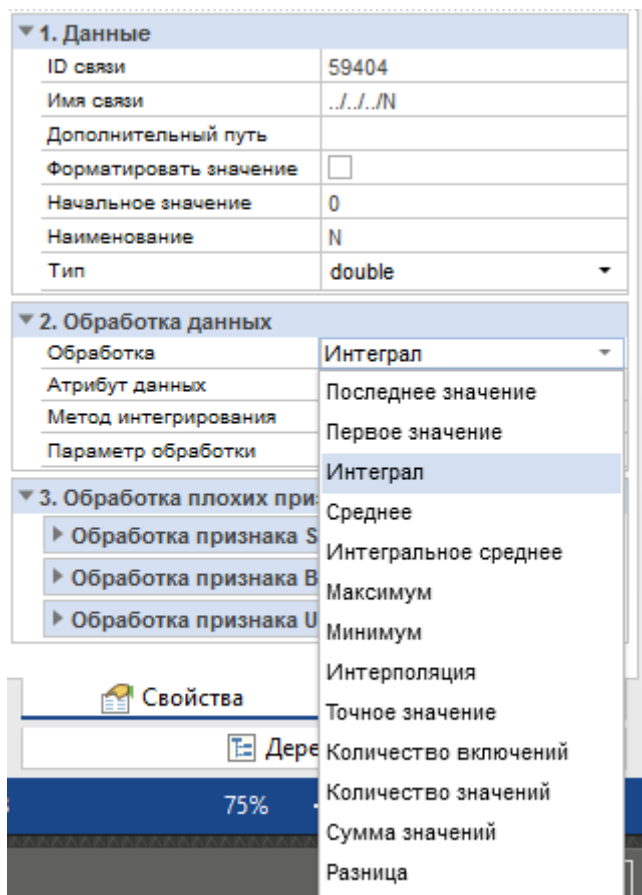


- Далее необходимо определить способ формирования отчета. Для этого выделим элемент Архив данных в Словаре отчета. и перейдем на панель Свойства в интерфейсе дизайнера отчетов.



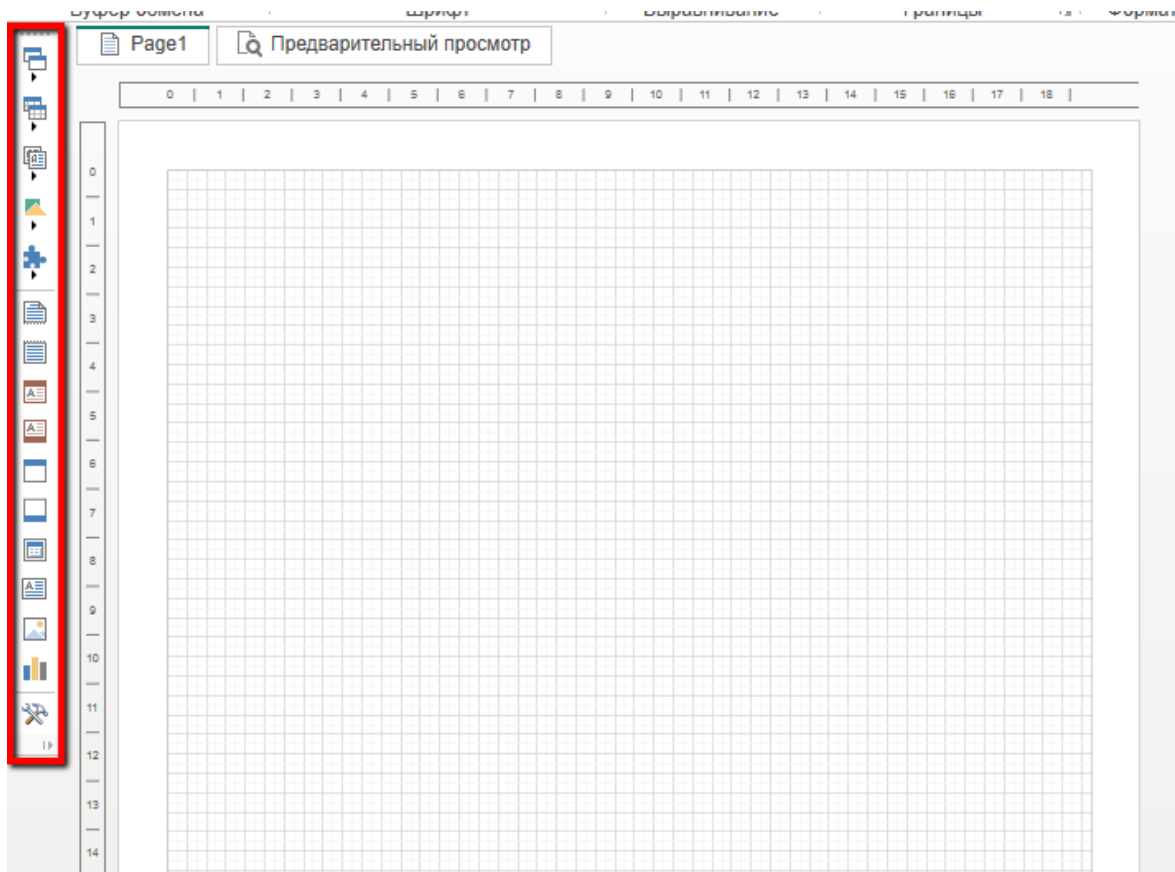
Для решения поставленной задачи укажем Способ формирования данных - Периодический.

- Далее необходимо выбрать способ обработки данных, т.е как рассчитывать значения для записи в отчет. Нужно выделить контролируемый параметр в словаре отчета. И перейти на панель свойств, где определить для каждого параметра свой способ обработки.



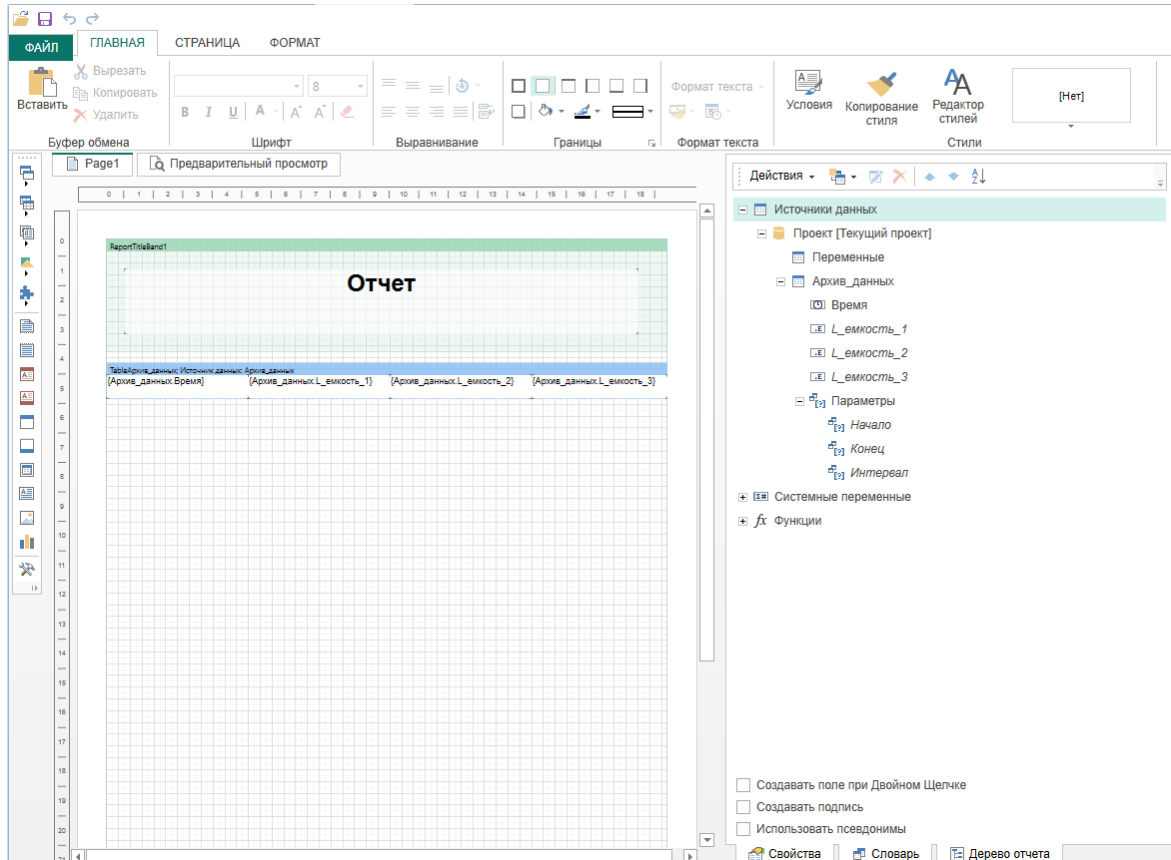
Например, если отчет будет формироваться раз в час, т.е. время начала будет 10:00, а время окончания 11:00, а Интервал будет 10, то в отчете будет 6 строк, и нужно определить какое число попадет в ячейку отчета, если выбрать способ обработки - минимальное, то в отчет попадет самое маленькое значение за 10 минут, если максимальное - то соответственно самое большое.

- Внешний вид отчета определяется его шаблоном, который задается в рабочей области, заголовок отчета, а также вид каждой новой страницы определяется элементами, которые располагаются с левой стороны поля отчета.



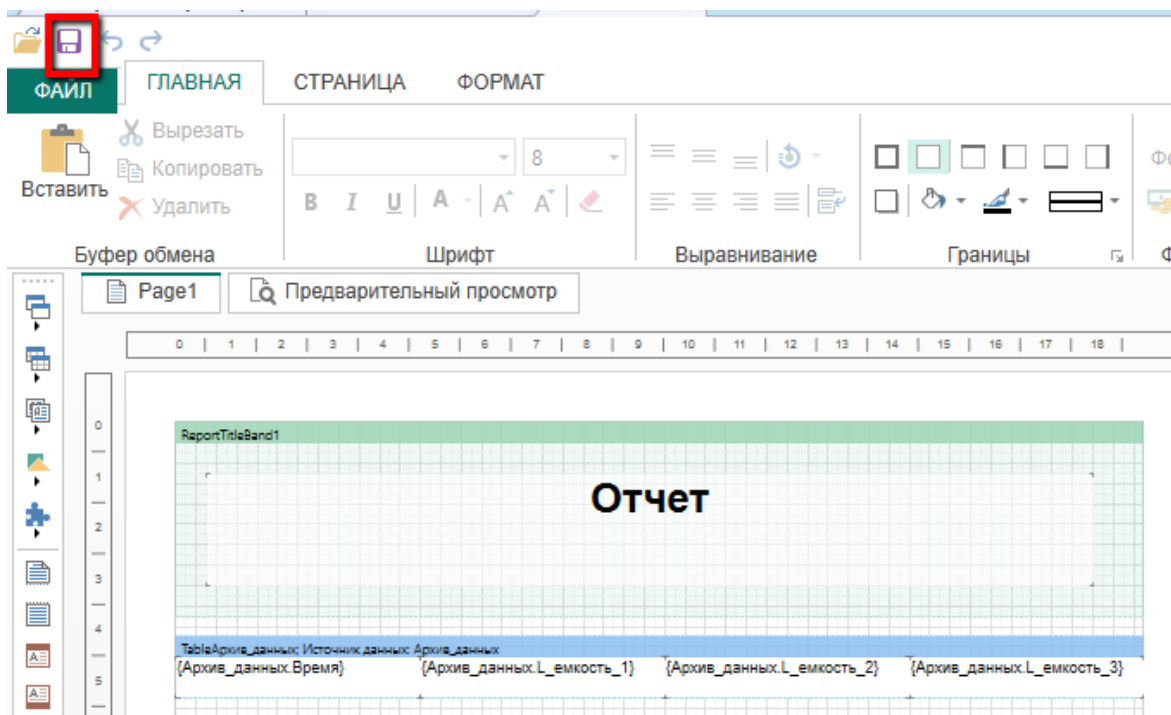
Добавим Заголовок отчета. Разместим в появившемся контейнере элемент Текст (свойства также настраиваются на соответствующей панели внутри редактора отчетов)

Перетащим из словаря элемент Архив данных на поле отчета. В диалоговом окне укажем, какие столбцы необходимы отображать в отчете.



Примечание: Если нужно добавить в отчет текущие значения параметров на момент формирования отчета, то их перетаскивают в Словарь в группу Переменные, а затем также перетаскивают в поле отчета в нужное место

- Сохраним отчет в редакторе отчетов



Созданный отчет попадет в базу данных проекта. После компиляции проекта, его запуске в исполнительной системе и формировании отчета (помните, что для формирования отчета нужны архивные данные) если не задавали дополнительных настроек отчет сохранится по умолчанию в папке:

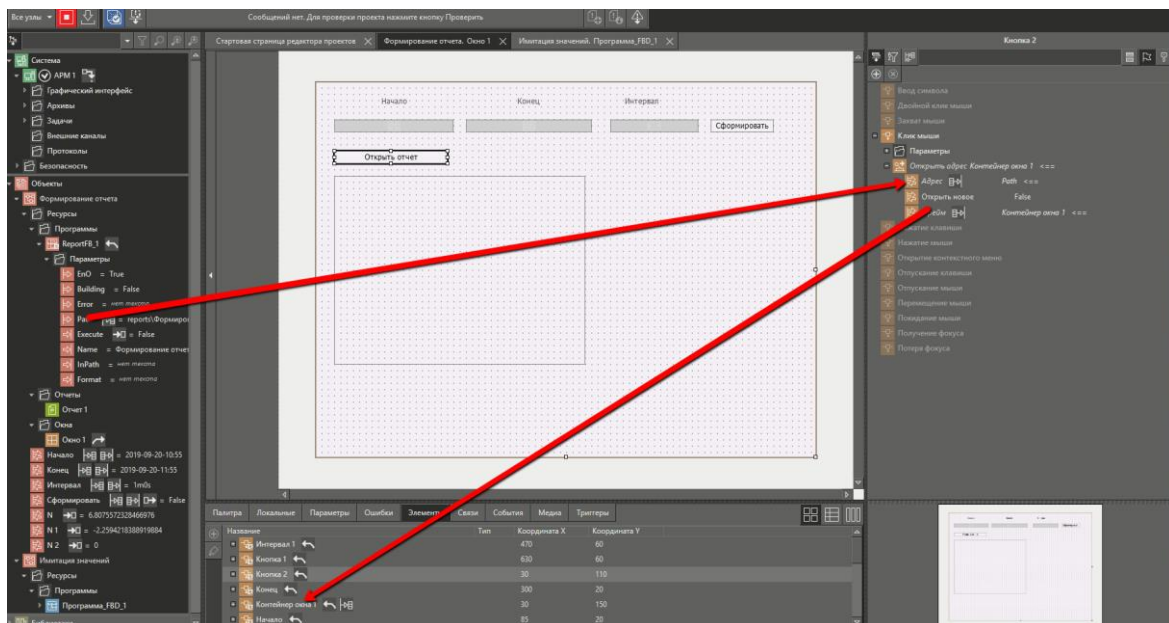
"c:\Users\[имя пользователя]\AppData\Roaming\MPSSoft\MasterSCADA4D1.I\Debug_[имя проекта]\[имя узла]\PLC\htdocs\reports\" - если работает среда исполнения, которая идет в комплекте со средой разработки

"c:\Users\[имя пользователя]\AppData\Roaming\MPSSoft\MasterSCADA4DRT1.I\Server\htdocs\reports\" - если среда исполнения устанавливалась отдельно

Открытие отчета в окне

Рассмотрим один из возможных способов открытия ранее сформированного отчета в окне:

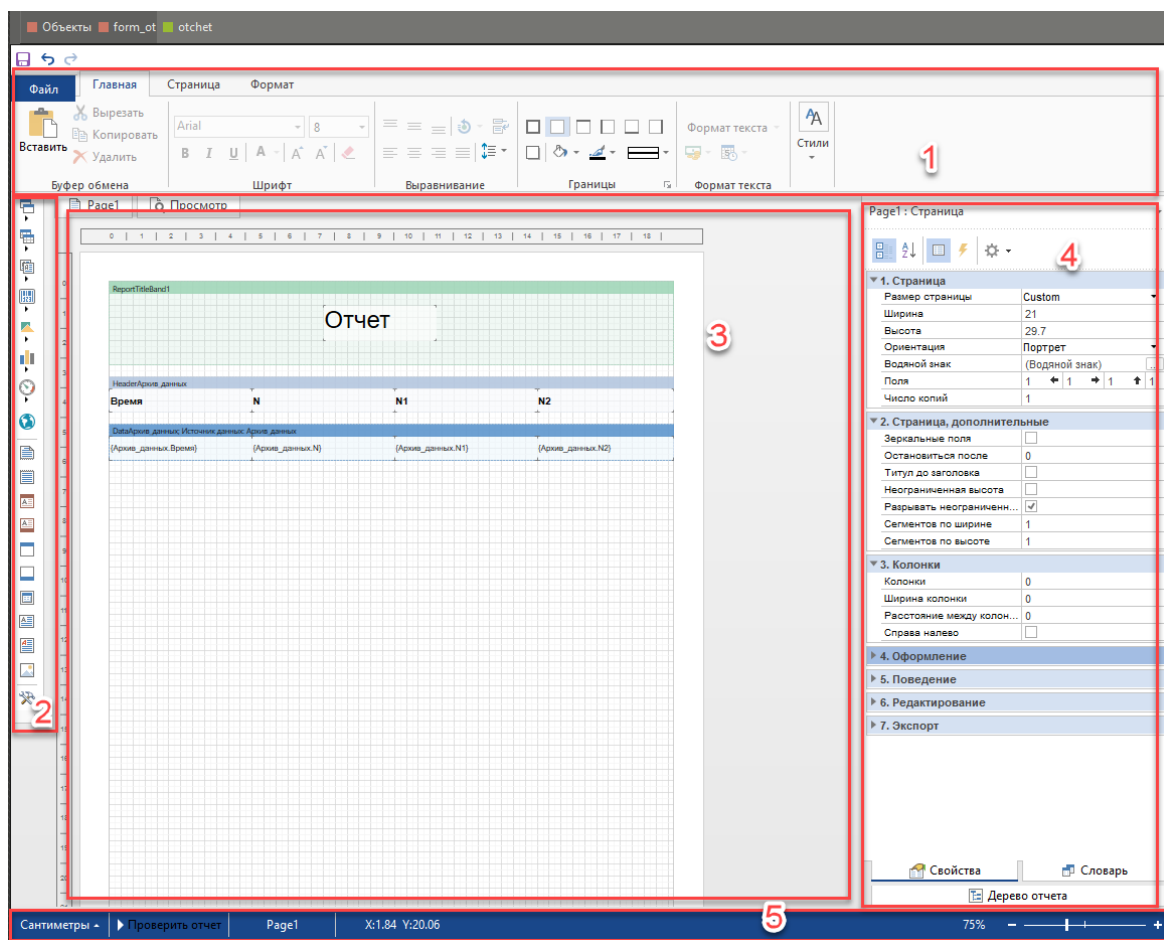
- Добавим в окно графический элемент Контейнер окна
- Добавим в окно графический элемент Кнопка.
- В панели свойств кнопки перейдем в режим События и в Клик мыши добавим действие Открыть адрес.
- Установим связь между выходом ФБ ReportFB Path и параметром действия Адрес.
- Откроем вкладку Элементы легенды графического редактора и перетащим параметр действия Фрейм на элемент Контейнер окна.



В результате в клиенте будет открываться последний созданный отчет.

9.2. РЕДАКТОР ОТЧЕТОВ

Вид редактора отчетов:



Окно редактора отчетов состоит из следующих компонентов:

1. Панель форматирования - используя инструменты данной панели можно настроить внешний вид добавленных компонентов (изменить шрифт, заливку, включить границы и т.д.)
2. Инструментарий - в данной панели находятся компоненты отчета.
3. Рабочая область отчета (Страница) - в данной области создается шаблон будущего отчета.
4. Набор панелей, состоящий из трех панелей:
 - Словарь - в словаре располагаются источники данных, из которых происходит формирование отчета.
 - Свойства - свойства текущего (выделенного) компонента
 - Дерево отчета - панель выводит структуру сформированного отчета в виде дерева.
5. Панель статуса - содержит дополнительные элементы управления отчетом, масштабом отчета его проверки.

9.2.1. СТРАНИЦА (PAGE)

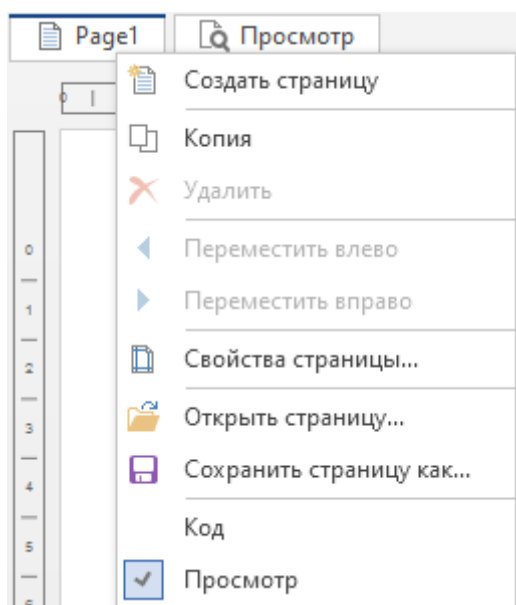
Страница (Page) это основное понятие отчета. Страница — любая из двух сторон листа бумаги. В генераторе отчетов, страница является основным понятием, а также рабочей областью редактора. Страница в редакторе отчетов выступает в роли контейнера, т.е. на ней можно положить другие компоненты генератора отчетов, но страницу нельзя поместить на какой-либо компонент. Именно на страницы добавляются бэнды, в которые в свою очередь вставляются, компоненты, отображающие информацию.

Для работы со страницей отчета предусмотрено несколько инструментов: Контекстное меню, Панель свойств, Окно настройки.

Контекстное меню страницы

Контекстное меню страницы открывается если нажать правую кнопку мыши на заголовке вкладки страницы, в верхней части рабочей области отчета.

Вид меню:



Название	Назначение
Создать страницу	Добавляет еще одну страницу в макет отчета. Количество страниц в экземпляре отчета, т.е. уже в итоговом отчете, который сформируется в режиме исполнения, будет зависеть от количества данных в отчете. Сначала в экземпляре отчета будут располагаться данные, которые получились в результате обработки первой страницы макета (их может получиться, как одна страница, так и произвольное количество), а затем данные второй страницы макета. Таким образом макет отчета может содержать в себе две страницы, а в итоге в отчете может быть гораздо больше страниц.

Название	Назначение
Копия	Добавляет еще одну страницу в макет отчета, копируя в нее все содержимое текущей страницы.
Удалить	Удаляет текущую страницу. Единственную страницу в отчете удалять нельзя.
Переместить влево	Перемещает вкладку страницы влево, тем самым определяется порядок формирования страниц в отчете. Сначала будут обрабатываться страницы макета, вкладки которых располагаются левее в редакторе отчетов
Переместить вправо	Перемещает вкладку страницы вправо, тем самым определяется порядок формирования страниц в отчете. Сначала будут обрабатываться страницы макета, вкладки которых располагаются левее в редакторе отчетов
Свойства страницы	Открывает окно настройки страницы.
Открыть страницу	
Сохранить страницу как...	
Код	Определяет наличие вкладки Код, где отчет представлен в виде XML
Просмотр	Определяет наличие вкладки Просмотр, которая служит для оперативного просмотра внешнего вида отчета, который получится в результате.

Окно настройки страницы

Окно настройки открывается при помощи пункта Свойства страницы контекстного меню.

Вид окна:

Свойства страницы

Размер страницы

Размер: По выбору

Ширина: 21

Высота: 29.7

Ориентация страницы

Портретная

Альбомная

Поля страницы

Поля страницы указываются в текущих единицах измерения отчета

Слева: 0

Справа: 2

Сверху: 2

Снизу: 2

Зеркальные поля

Бумага

Колонки

Водяной знак

OK Отмена

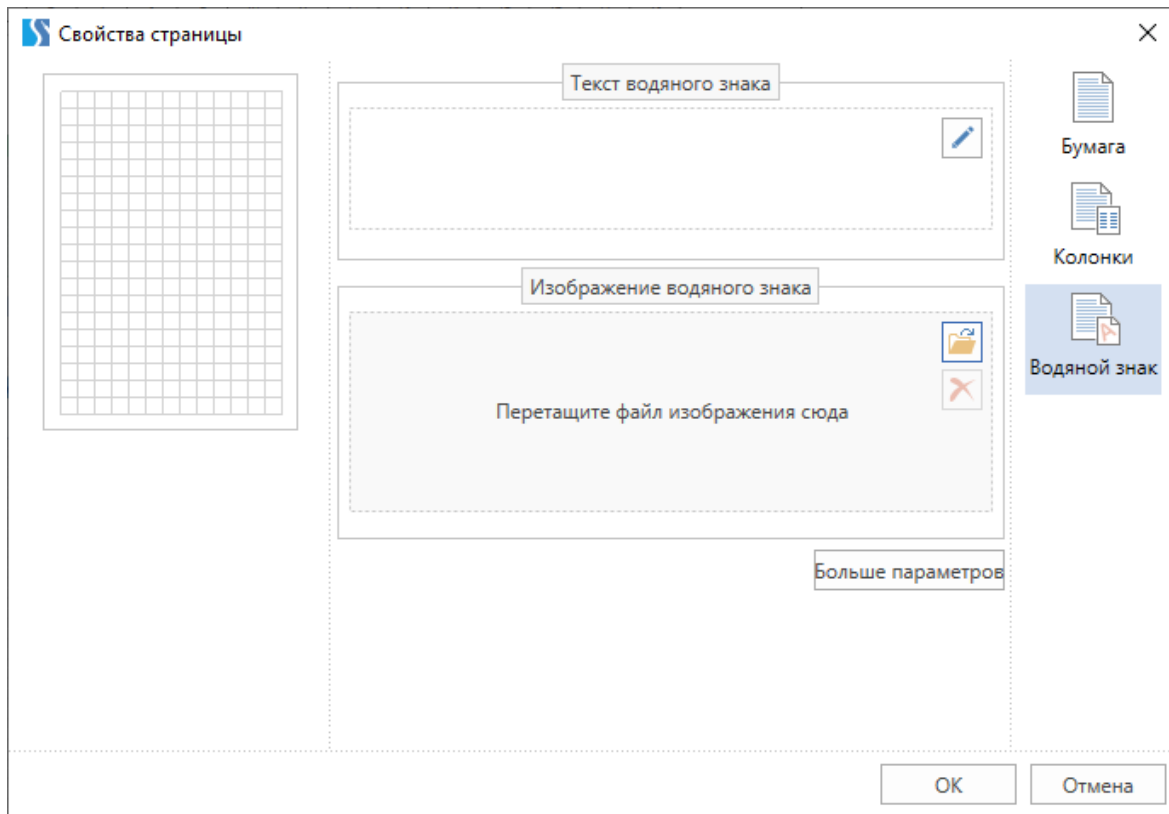
В окне можно задать произвольные значения свойств страницы. Окно имеет три вкладки: Бумага, Колонки, Водяной знак.

На вкладках Бумага и Колонки задаются настройки страницы, которые также можно настроить в панели свойств (описание ниже).

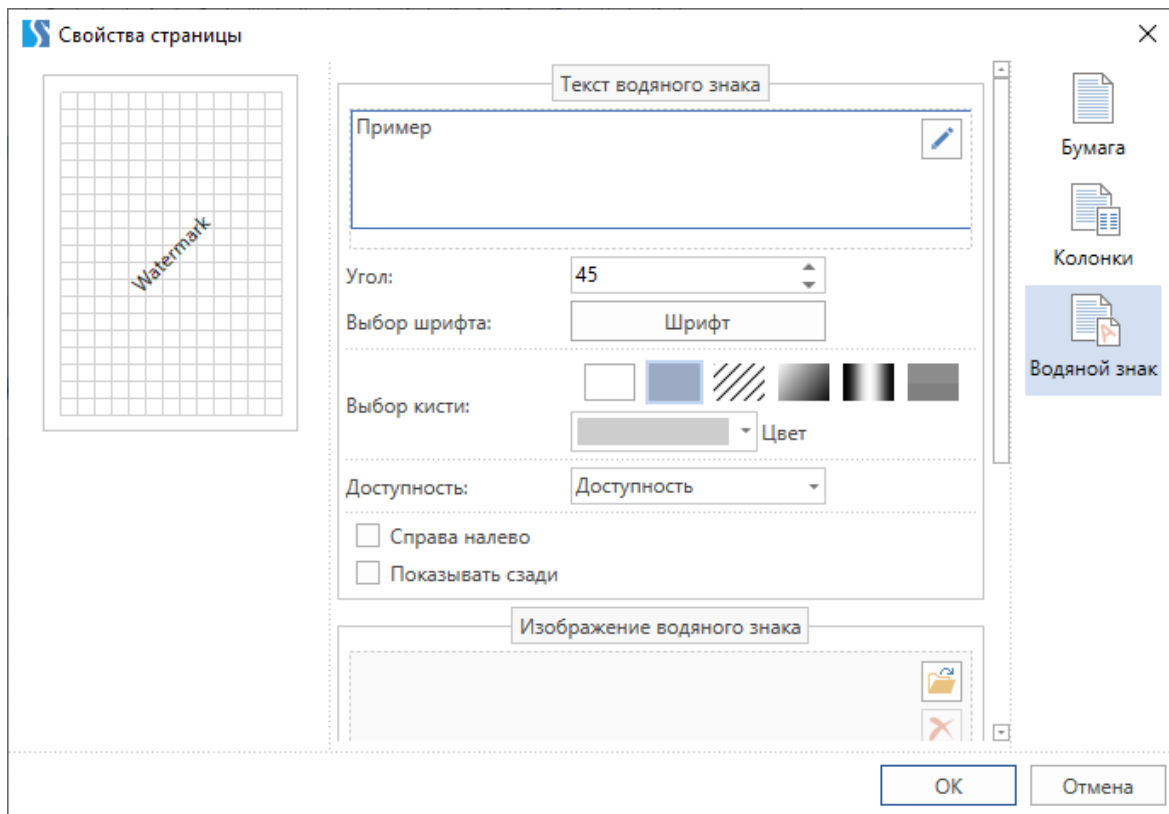
Вкладка Водяной знак

Позволяет выбрать файл или ввести текст, который будет использоваться в качестве водяного знака (текста или рисунка, изображаемого на заднем фоне отчета) при формировании отчета.

Вид вкладки:








Слева отображается предпросмотр водяного знака, а справа вводится текст или выбирается файл. Кнопка Больше параметров включает расширенные настройки:



Панель свойств Страницы

Вид панели:

Page1 : Страница

1. Страница

Размер страницы	Custom
Ширина	21
Высота	29.7
Ориентация	Портрет
Водяной знак	(Водяной знак) ...
Поля	0 ← 2 → 2 ↑ 2 ↓
Число копий	1

2. Страница, дополнительные

Зеркальные поля	<input type="checkbox"/>
Остановиться после	0
Титул до заголовка	<input type="checkbox"/>
Неограниченная высота	<input type="checkbox"/>
Разрывать неограниченн...	<input checked="" type="checkbox"/>
Сегментов по ширине	1
Сегментов по высоте	1

3. Колонки

Колонки	0
Ширина колонки	0
Расстояние между колон...	0
Справа налево	<input type="checkbox"/>

4. Оформление

5. Поведение



Доступность	Да
Взаимодействие	[Взаимодействие] ...
Печатать на предыдущей...	<input type="checkbox"/>
Использовать Заголовки...	<input type="checkbox"/>
Сбросить номер страниц	<input type="checkbox"/>

6. Редактирование

Наименование	Page1 ...
Псевдоним	...
Большая высота	<input type="checkbox"/>
Кэффициент большой в...	4


7. Экспорт

Лист Excel	...
------------	-----

 Свойства
  Словарь

Рассмотрим некоторые свойства страницы более детально:

Название категории	Описание
Страница	<ul style="list-style-type: none"> • Размер (Size) - это размер самой страницы. Этот параметр зависит от установленного принтера, т.е. доступны те размеры страницы, которые поддерживает установленный принтер. <ul style="list-style-type: none"> • Ширина/высота - служат для задания размера страницы вручную • Ориентация (Orientation) предоставляет возможность изменить ориентацию страницы. Страница может располагаться в вертикальном и горизонтальном формате, т.е. параметр Ориентация (Orientation) может принимать следующие значения: Портретная (Portrait) и Альбомная (Landscape). • Поля страницы предоставляет возможность изменить границы рабочей области на странице в редакторе. • Число копий (Number Of Copies) - предоставляет возможность построить одну и ту же страницу шаблона отчета необходимое количество раз. Значением данного свойства может быть любое целое положительное число. По умолчанию, данное свойство установлено в значение 1.
Страница дополнительные	<ul style="list-style-type: none"> • Зеркальные поля - позволяет распечатать на одном листе страницы таким образом, чтобы большее поле было с одной и той же стороны листа. Подробнее смотрите в разделе: Поля и границы • Остановиться после - используется, если, к примеру, есть необходимость прекратить построение отчета после определенной страницы, то сделать это можно при помощи свойства Остановиться после (Stop Before Print). Значением данного свойства будет порядковый номер страницы построенного отчета, после которой необходимо остановиться. <ul style="list-style-type: none"> • Титул до заголовка - позволяет ввести текст до Заголовка отчета • Разрывать неограниченный - определяет, что при построении многостраничных, кросс-отчетов и т.п. колонки и строки должны разрываться • Сегментов по ширине/высоте - если в отчете данные необходимо разместить на одной странице по ширине или по высоте, а размеры страницы малы, то можно добавить необходимое количество сегментов по ширине и/или по высоте. В этом случае, одним сегментом является целая страница, а итоговая страница складывается из нескольких сегментов по ширине или

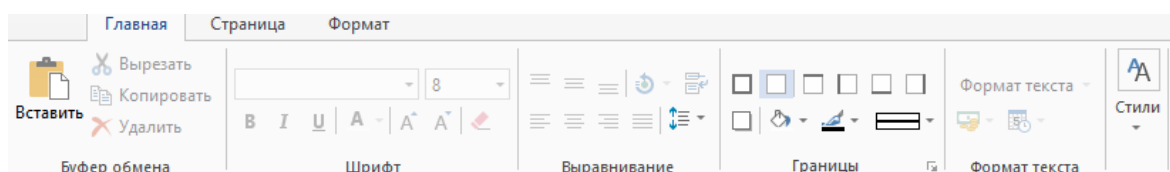
Название категории	Описание
	<p>высоте. На рисунке снизу представлены сегментарные страницы по ширине:</p> 
Колонки	<p>Если установить настройку Количество колонок больше чем 1, то данные будут выводиться в виде колонок. Кроме того необходимо задать еще два свойства: Ширина колонки (ColumnWidth) и Расстояние между колонками (ColumnGaps). В свойстве Ширина колонки (ColumnWidth) указывается ширина выводимой колонки. Свойство Справа налево определяет порядок вывода колонок.</p>
Поведение	<ul style="list-style-type: none"> Печатать на предыдущей странице - страницы шаблона отчета обрабатываются и выводятся на печать/сохранение последовательно, т.е. сначала обрабатывается первая страница шаблона, затем вторая и т.д. Порядок обработки страниц можно узнать на вкладке Дерево отчета (Report Tree), чем выше страница в дереве, тем выше её приоритет обработки. В случае, с копиями страницы, сначала будет обрабатываться и печататься исходная страница, затем её копии. Следует понимать, построение страницы шаблона отчета начинается с новой страницы в построенном отчете. Например, первая страница шаблона отчета была развернута на 14 с половиной страниц. В этом случае, построение второй страницы шаблона отчета будет начинаться с 15 страницы в построенном отчете. Использовать заголовки и итоги с предшествующей страницы - если шаблон отчета состоит из нескольких страниц, то каждая страница строится

Название категории	Описание
	<p>независимо, поэтому на каждой странице шаблона могут использоваться разные данные, заголовок и итог страницы, а свойство Использовать заголовки и итоги с предшествующей страницы (PrintHeadersAndFootersFromPreviousPage) предоставляет возможность использовать заголовок и итог с предыдущей страницы. Для этого, следует данное свойство установить в значение TRUE.</p> <ul style="list-style-type: none"> Сбросить номер страниц - позволяет сбросить номер страниц текущей страницы
Экспорт	<p>Свойство Лист Excel (Excel Sheet) предоставляет возможность экспортировать каждую страницу шаблона отчета на отдельную вкладку в Excel. Значением данного свойства будет название вкладки Excel, на которую будет экспортирована данная страница шаблона отчета. Если значения свойства Лист Excel (Excel Sheet) у разных страниц шаблона отчета совпадают, то эти страницы экспортируются в одну вкладку в Excel.</p> <p>Если же в качестве свойства указать выражение, например передать номер страницы:</p> <p style="text-align: center;">{PageNumber}</p> <p>то каждая страница отчета будет представлена в Excel как отдельная вкладка, с именем равным номеру страницу отчета.</p>

9.2.2. ПАНЕЛЬ ФОРМАТИРОВАНИЯ

Панель форматирования предназначена для настройки отображения компонентов отчета.

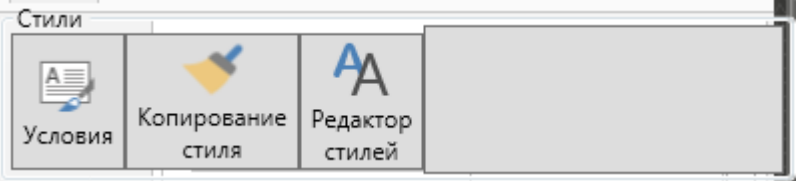
Вид панели:



Панель содержит три вкладки: Главная, Страница и Формат.

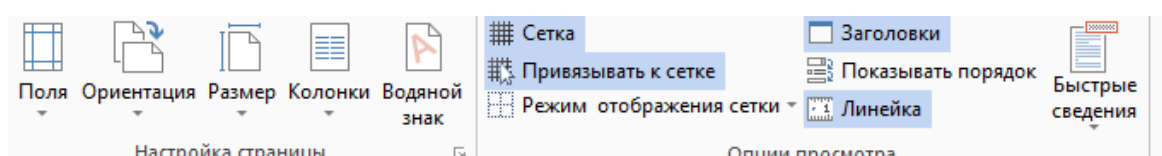
Вкладка Главная

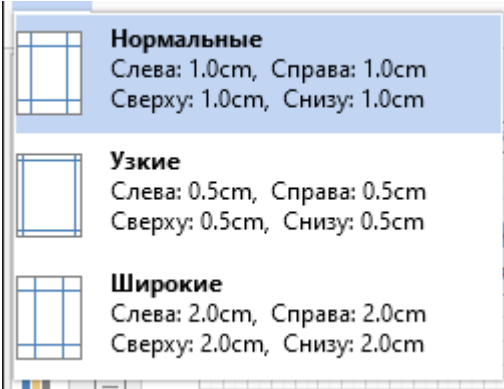
Вид вкладки показан на рисунке выше.

Название группы	Назначение
Буфер обмена	Группа предоставляет стандартные команды для работы с буфером обмена дизайнера отчетов: Вставить, Вырезать Копировать, Удалить
Шрифт	Группа управления компонентами, которые умеют выводить текст заданным шрифтом и цветом. Важно! При выборе шрифта убедитесь, что он присутствует на устройстве, где отчет будет формироваться. Рекомендуется использовать стандартные шрифты.
Выравнивание	Группа предназначена для управления горизонтальным и вертикальным выравниванием содержимого компонентов, которые поддерживают выравнивание содержимого. Кроме этого группа содержит команды для установки угла поворота текста и управления свойством Перенос текста (WordWrap).
Границы	В группе собраны команды для настройки границ компонента (наличие границ, цвета границ и заливки содержимого внутри границ)
Формат текста	Определяется формат вывода значений.
Стили	Группа управляет стилями и условиями, которые используются для автоматического оформления компонентов в отчете (Условное выделение). При нажатии на кнопку откроется вспомогательная панель: 

Вкладка Страница

Вид вкладки:

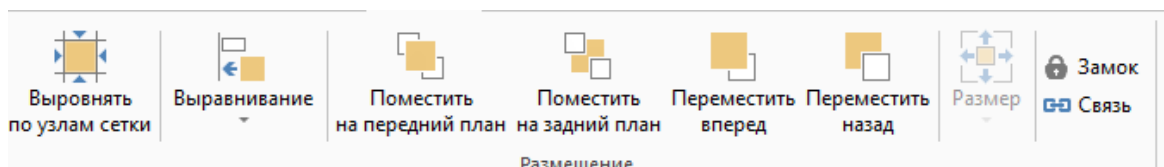


Название	Назначение
Группа Настройки страницы	
Поля	<p>Определяет ширину полей страницы. При нажатии на кнопку открывается выбор из трех вариантов:</p>  <p>Для задания других значений можно воспользоваться панелью свойств свтраницы или окном настройки</p>
Ориентация	Определяет ориентацию страницы - Портретная или Альбомная
Размер	Определят размер страницы. При нажатии на кнопку открывается выбор из стандартных размеров.
Колонки	Позволяет вывести данные отчета в виде колонок. При нажатии на кнопку открывается выбор: Одна, Две, Три. Чтобы задать дополнительные настройки колонок нужно воспользоваться панелью Свойства страницы или окном настройки.
Водяной знак	Открывает Окно настройки страницы, где можно указать файл, который будет использоваться в качестве водяного знака при отображении и печати отчета.
Группа Опции просмотра	
Сетка	Определяет видимость сетки при проектировании отчета
Привязывать к сетке	Определяет необходимость привязывать компоненты к узлам сетки

Название	Назначение
Режим отображения сетки	Задаёт режим отображения сетки (линии или точки)
Заголовки	Определяет отображение заголовков компонентов отчета
Показывать порядок	Определяет необходимость отображения на компонентах порядка их обработки при построении отчета.
Линейка	Определяет наличие линейки при построении отчета.
Быстрые сведения	Определят, какая информация будет отображаться на компонентах при проектировании.

Вкладка Формат

Вид вкладки:



Название	Назначение
Выровнять по узлам сетки	Выравнивает все выделенные компоненты по сетке страницы
Выравнивание	Выравнивает все выделенные компоненты согласно выбранной команде.
Группа Размещение	Определяет положения элемента по глубине, относительно други элементов. Определяет z-порядок.
Размер	Устанавливает такой же размер как и у первого выделенного компонента
Замок	Управляет свойством Замок (Lock).

Название	Назначение
Связь	Управляет свойством Связь (Link).

Изменить свойства элементов можно не только через панель форматирования, но и через панель Свойства

9.2.3. ИНСТРУМЕНТАРИЙ

Инструментарий (ToolBox) это панель элементов, в которой сосредоточены основные инструменты для создания отчета. Инструментарий располагается слева в окне дизайнера, в виде вертикальной панели.


Вид инструментария по умолчанию:

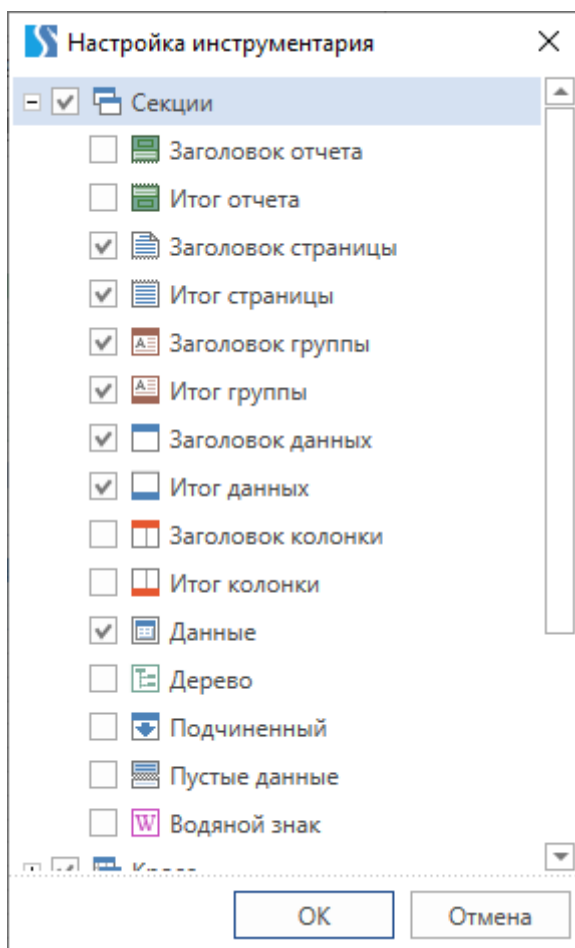


В верхней части панели находятся категории компонентов. Нажатие на кнопку панели приведет к открытию дополнительной панели, на которой будут располагаться все возможные элементы данной категории.

Категория	Назначение
 Секции	Категория содержит простые секции - базовые элементы для построения отчетов. Назначение, каждой секции категории смотрите в разделе Типы секций.
 Кросс	Категория содержит секции, которые нельзя расположить непосредственно на странице, но их можно положить на любую простую секцию. Это позволяет строить очень сложные кросс отчеты. Назначение, каждой секции категории смотрите в разделе Типы секций.
 Компоненты	Категория содержит простейшие компоненты для отображения данных.
 Штрих код	Категория содержит элементы, которые позволяет в отчете выводить штрих-коды, и QR-коды. Порядок работы со штрих-кодами смотрите в соответствующем разделе.
 Примитивы	Категория содержит простейшие элементы
 Диаграмма	Категория содержит элементы для построения различных диаграмм. Порядок работы с диаграммами смотрите в соответствующем разделе.
 Измерительный прибор	Категория содержит элементы для вывода значений в виде показаний измерительных приборов.
 Карта	Представляет собою инструмент визуализации данных с привязкой к географическому положению. При помощи карт можно отображать различные статистические данные по конкретному региону или в целом по миру. Порядок работы с картами смотрите в соответствующем разделе.

Под категориями находится панель быстрого доступа (Category Shortcut). В ней располагаются те элементы, которые выберет разработчик проекта. Как правило, в панель выводятся те элементы, которые используются наиболее часто.

Чтобы определить, какие элементы будут отображаться в панели быстрого доступа необходимо нажать кнопку  инструментария. Откроется окно, в котором флагами нужно выбрать нужные элементы:



Компоненты которые отмечены флагом в окне Настройка инструментария (Setup Toolbox) будут отображаться на инструментарии в категории быстрого доступа. Соответственно, компоненты которые не отмечены отображаться не будут.





9.2.3.1. ТИПЫ СЕКЦИЙ



В генераторе отчетов есть множество различных типов секций, каждый из которых обладает специфическими возможностями. Все секции, в свою очередь, разделяются на две категории: секции и кросс секции. Кросс-секции не могут быть расположены на странице, но зато их можно расположить на простых секциях. Секции доступны в Инструментарии. После нажатия на соответствующую кнопку появится список всех доступных секций (бэндов).






Простые секции





Простые секции доступны в Инструментарии. После нажатия на соответствующую кнопку появится список всех доступных секций (бэндов).

Простые секции являются базовым элементом для построения отчетов. Ниже в таблице приведены все простые секции.

Иконка	Название	Описание
	Заголовок отчета (ReportTitleBand)	Печатается один раз в начале отчета. Одним из способов создания заголовка отчета, является вариант с использованием бэнда Заголовок отчета (ReportTitleBand). Заголовок отчета будет выведен один раз в начале отчета. Бэнд Заголовок отчета (ReportTitleBand) располагается после бэнда Заголовок страницы (PageHeaderBand), перед бэндом Заголовок данных (HeaderBand). Какие-либо ограничения на количество размещенных бэндов, Заголовок отчета (ReportTitleBand), в генераторе отчетов отсутствуют.
	Итог отчета (ReportSummaryBand)	Печатается один раз в конце отчета. Используется для вывода итогов отчета.. Каких-либо ограничений на количество размещаемых бэндов Итог отчета (ReportSummaryBand) на странице шаблона нет. Если в шаблоне отчета более одной страницы, то бэнд Итог отчета (ReportSummaryBand) может располагаться на каждой странице шаблона отчета. При этом выводится он будет после каждой построенной шаблонной страницы.
	Заголовок страницы (PageHeaderBand)	Печатается сверху на каждой странице. Эта секция используется для вывода заголовка страницы. Секция Заголовок страницы выводится в самом верху страницы, один раз на каждой странице экземпляра отчета. Секция используется для вывода номеров страниц, даты построения отчета и т.д.- любую информацию, которую нужно отображать на каждой странице. Генератор отчетов не накладывает каких-либо ограничений на количество секций Заголовок страницы. На одной странице можно разместить два, три, десять заголовков.
	Итог страницы	Печатается внизу на каждой странице. Для вывода информации внизу каждой страницы экземпляра отчета используется секция Итог страницы. Секция Итог страницы выводится в самом низу страницы, один раз на каждой странице отчета. Генератор

Иконка	Название	Описание
		<p>отчетов не накладывает каких-либо ограничений на количество секций Итог страницы. Эта секция используется для вывода номеров страниц, итогов по странице и т.д.</p>
	<p>Заголовок группы</p>	<p>Печатается в начале группы. Секция Заголовок группы является основной секцией для построения отчетов с группировкой. Без этой секции выполнить группировку невозможно. На заголовке группы размещаются компоненты для отображения информации по группе. К примеру, это может быть наименование группы, дата, обозначение условия группировки и т.д. Секция Заголовок группы выводится один раз в начале каждой группы. Для создания группировки необходимо указать условие. Условие группировки задается в свойстве Условия этой секции или при помощи редактора секции Заголовок группы.</p> <p>Важно! Секция Заголовок данных всегда выводится выше секции Заголовок группы, в независимости от того, как эти секции расположены на странице в дизайнера.</p> <p>В момент построения отчета, генератор отчетов связывает заголовок группы и определенную секцию Данные. Секция Заголовок группы размещается на странице выше секции Данные, который выводит строки с данными. Секция Заголовок группы всегда относится к какой-то конкретной секции Данные. Обычно этой секцией является первая секция Данные, которая находится ниже секции Заголовок группы. Для построения отчета с группировкой необходима секция Данные.</p> <p>Секция Данные необходима, потому что строки с данными выводятся при помощи этой секции и именно на основе этих строк с данными строится группировка. Также у секции Данные задается сортировка строк, которая влияет на порядок построения группировки.</p>
	<p>Итог группы</p>	<p>Печатается в конце группы. Секция Итог группы используется для вывода различной информации по группе. К примеру, для того чтобы вывести количество строк в группе, достаточно разместить на секции Итог группы, текстовый компонент. В текстовом компоненте указать выражение: {Count()}. Эта секция размещается ниже секции Данные. Причем, имеется ввиду именно та секция Данные, с которой связана секция Итог группы. Каждая секция Итог группы, относится к какой то определенной</p>

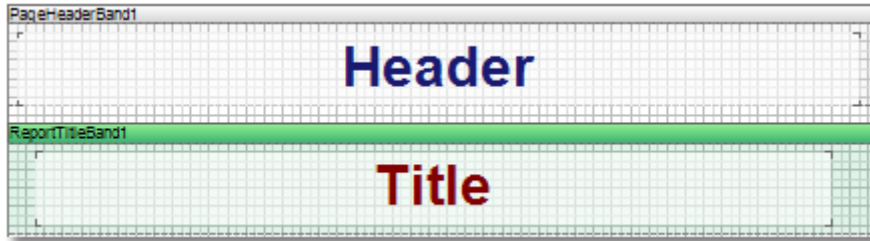
Иконка	Название	Описание
		секции Заголовок группы. Без секции Заголовок группы, секция Итог группы выводится не будет.
	Заголовок колонки	Печатается перед колонкой. Секция Заголовок колонки выводится один раз, но компоненты, расположенные на ней, будут повторно выведены над каждой колонкой. Причем секция Заголовок колонки используется только для колонок, расположенных на секции Данные.
	Итог колонки	Секция выводится один раз под каждой колонкой. Все компоненты, расположенные на этой секции, будут также выведены под каждой колонкой.
	Заголовок данных (HeaderBand)	Печатается перед данными. Для вывода наименований данных и другой информации перед данными используется специальная секция Заголовок данных. Она размещается на странице перед секцией Данные. При построении отчетов для одной секции Данные, можно создавать более одной секции Заголовок данных. К примеру, одна секция Заголовок данных может выводиться только в начале данных. А вторая и в начале данных и повторно на других страницах отчета. Секции Заголовок данных выводятся в том порядке, в каком они размещены на странице.
	Итог данных	Печатается после данных. Эта секция размещается после секции Данные. По умолчанию, секция Итог печатается один раз, после того как будут напечатаны все данные. На секцию итог можно например, вынести один текстовый компонент, который будет выводить количество выведенных на печать строк.
	Данные	Печатается столько раз, сколько записей в источнике данных. На основании данных полученных из источника данных формируется таблица отчета. Каждой секции Данные указывается источник данных (например источник Архив MasterSCADA). Источник данных фактически представляет из себя таблицу. В каждом источнике данных есть поля данных. Разместив на секции данных текстовые компоненты со ссылками на эти поля, можно вывести на печать таблицу. Один источник данных может указывать на неизвестное заранее количество строк с данными. Секция Данные выводится столько раз, сколько строк в указанном источнике данных. К примеру, если в

Иконка	Название	Описание
		<p>источнике данных есть 100 строк, то секция Данные будет выведена на печать ровно 100 раз. Если при выводе на печать заканчивается место на странице, то будет сформирована новая страница и печать будет продолжена.</p> <p>Иногда необходимо напечатать секцию Данные определенное количество раз, без указания ей источника данных. Для этого используется свойство Количество данных. По умолчанию, это свойство равно 0. Но если, например, установить его в 4, то секция Данные будет напечатана ровно 4 раза. Это можно использовать, например, для печати пустых граф. Стоит помнить, что в случае использования секции Данные как виртуальной, источник данных нужно не указывать. Указать количество элементов можно также при помощи редактора секции Данные. (Редактор секции Данные, открывается в момент установки их на странице, либо если выполнить двойной клик по элементу.)</p>
	Подчиненный	Печатается один раз сразу за бэндом, после которого он расположен
	Дерево	Бэнд выводится столько раз, сколько строк в источнике данных, который с ним связан. Данные выводятся в виде дерева.
	Пустые данные	Бэнд используется для заполнения пустого места в нижней части таблицы
	Водяной знак	Бэнд используется для вывода водяного знака

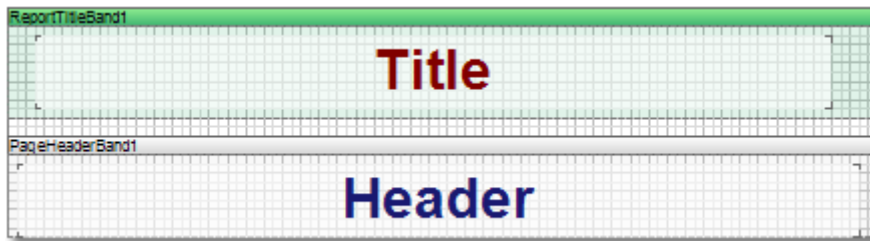
Для удобства визуального восприятия все секции имеют свой цвет. Это позволяет легко понимать структуру отчета:



По умолчанию, бэнд Заголовок страницы (PageHeaderBand) располагается выше бэнда Заголовок отчета (ReportTitleBand):



Но возможно вывести бэнд Заголовок отчета (ReportTitleBand) до бэнда Заголовок страницы (PageHeaderBand):




Для этого используется свойство страницы Титул до заголовка (TitleBeforeHeader). По умолчанию это свойство выключено, т.е. значение свойства равно ложь (false). Это означает, что бэнд Заголовок страницы (PageHeaderBand) располагается выше, чем бэнд Заголовок отчета (ReportTitleBand).

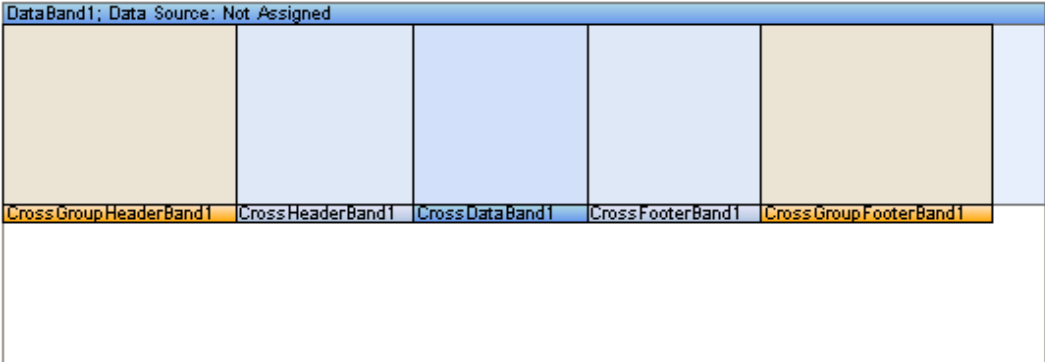
Кросс-секции

Вторая категория секций – это кросс-секции. Секции этой категории нельзя положить на страницу, но их можно положить на любую простую секцию. Это позволяет строить очень сложные кросс отчеты. Список кросс-секций приведен ниже.

Иконка	Название	Описание
	Кросс-заголовок группы	Печатается в начале группы
	Кросс-итог группы	Печатается в конце группы
	Кросс-заголовок данных	Печатается перед данными
	Кросс-итог данных	Печатается после данных




Иконка	Название	Описание
	Кросс-данные	Печатается столько раз, сколько записей в источнике данных





В отличие от простых секций, у кросс-секций заголовки расположены не сверху секции а снизу.



9.2.3.2. КОМПОНЕНТЫ

Компоненты доступны в Инструментарии. После нажатия на соответствующую кнопку появится список всех доступных компонентов:

Иконка	Название	Назначение
	Текст	Основной компонент для вывода текстовой информации в отчете. Компонент имеет большое количество настроек, для формирования нужного внешнего вида. Помимо этого можно использовать возможности условного выделения, когда в зависимости от значения отображаемого в тексте меняется и внешний вид компонента.
	Текст в ячейках	Компонент для вывода текста в виде ячеек. Обычно используется при создании бланков
	Rich текст	Компонент для вывода и редактирования текста RTF. Поддерживает загрузку сохранение файлов в формате RTF. Этот компонент умеет автоматически изменять свои размеры в зависимости от размеров RTF текста в нем, умеет обрабатывать выражения, поддерживает различные варианты оформления, обработку в конце построения отчета и т.д.

Иконка	Название	Назначение
	Изображение	Используется для вывода изображений. Поддерживает следующие графические форматы: BMP, JPG, JPEG, GIF, TIFF, PNG, ICO, WMF, EMF. Порядок добавления изображения в отчет смотрите в соответствующем разделе.
	Панель	Прямоугольная область, где другие компоненты могут быть размещены. При перемещении панели компоненты, размещенные на ней будут также перемещены. Панель может размещаться как на бэнде, так и на странице. Панель может быть клонирована. Альтернативное название компонента - Контейнер. См. также раздел Панели-контейнеры
	Клон	Используется для клонирования части отчета в другое место. Клонирование может быть применено только к компоненту панель.
	Флажок	Компонент для отображения флагов в отчете. Компонент может менять свой внешний вид в зависимости от условия, которое может принимать два значения: Включено, Выключено
	Вложенный отчет	Используется для вывода дополнительных данных в разных местах отчета
	Почтовый индекс	Используется для вывода почтового индекса
	Таблица	Представляет собой набор элементов данных в виде таблицы
	Кросс-таблица	Используется для представления структурированных данных в виде таблицы. (Когда по горизонтали и вертикали находятся разные колонки источника данных и можно установить зависимость между ними.)
	Спарклайн	Используется для создания спарклайн (минидиаграмм).

9.2.3.3. ПРИМИТИВЫ

Примитивы доступны в Инструментарии. После нажатия на соответствующую кнопку появится список всех доступных компонентов:

Иконка	Название	Назначение
	Геометрия	Используется для добавления в отчет геометрических фигур. Геометрическим фигурам можно указывать различное оформление – кисть заполнения, цвет границ и т.д. Этот компонент умеет вводить следующие типы геометрических фигур: прямоугольник, прямоугольник со скругленными углами, треугольник, овал, стрела и различные виды линий.
	Горизонтальная линия	Используется для визуального оформления отчетов
	Вертикальная линия	Используется для визуального оформления отчетов. Может начинаться в области одного бэнда, а заканчиваться в области другого
	Прямоугольник	Используется для визуального оформления отчетов. Верхний левый угол может находиться в области одного бэнда, а нижний правый в области другого.
	Закругленный прямоугольник	Используется для визуального оформления отчетов. Верхний левый угол может находиться в области одного бэнда, а нижний правый в области другого.

9.2.4. ПАНЕЛИ

В данном разделе будут рассмотрены основные панели редактора отчетов: Словарь данных (Dictionary), Дерево отчетов (Report Tree), Свойства (Properties). На этих панелях сосредоточены свойства, функции и команды для управления различными компонентами отчета, а также располагаются информационные поля для уведомления и подсказок пользователю.

Подробное описание панелей смотрите в соответствующих разделах:

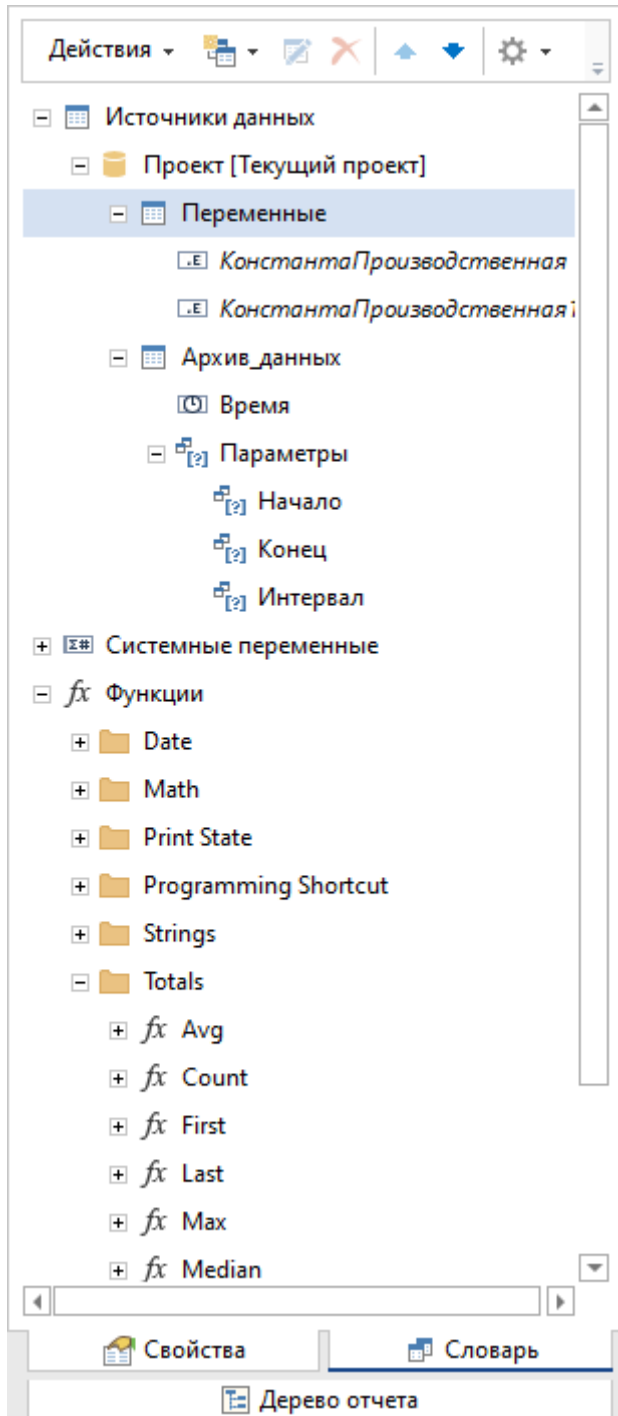
Словарь данных

Дерево отчетов

Свойства

9.2.4.1. СЛОВАРЬ ДАННЫХ

Панель Словарь (Dictionary) отображает подключенные данные, доступные источники данных, системные переменные и функции. Кроме этого в словаре можно создавать соединения и подключать новые источники данных, устанавливать связи между источниками данных. На рисунке снизу приведен пример Словаря (Dictionary):



Панель Словарь (Dictionary) включает в себя: Панель инструментов словаря (Dictionary ToolBar), Дерево данных (Data Tree), Панель описания (Description Panel), Панель настройки словаря (Dictionary Setting Panel). Рассмотрим эти составляющие более подробно.

Панель инструментов словаря (Dictionary ToolBar)

Представляет собой набор инструментов и команд для работы со Словарем (Dictionary).

Вид панели:

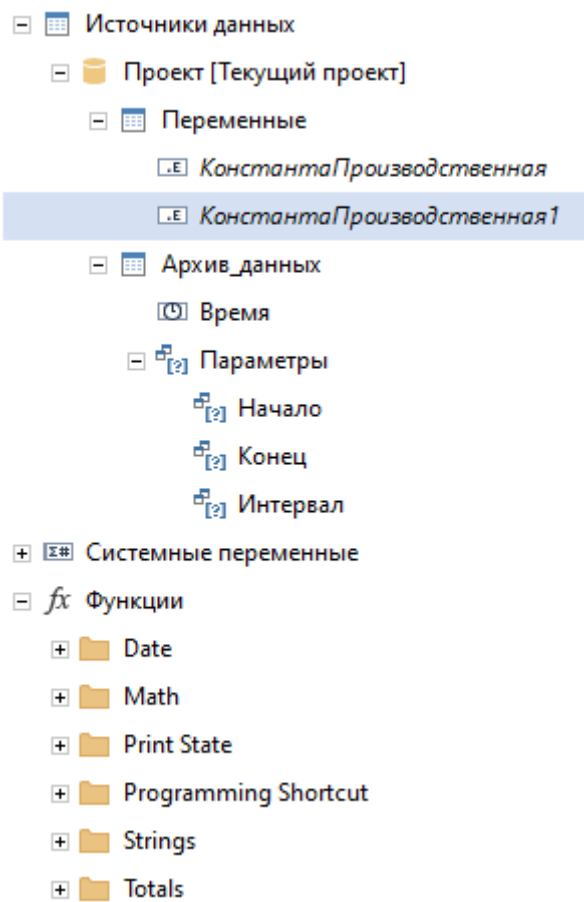


Элемент	Назначение
Действия (Actions)	Представляет собой набор команд для работы со словарем таких как: создание, открытие, сохранение словаря, добавление, экспорт и импорт словаря в виде XML-схемы, синхронизация данных представленных в словаре с данными зарегистрированными в хранилище данных. Как правило, данные команды при создании отчетов в MasterSADA 4D не используются.
Новый элемент (New Item)	Содержит команды для создания новых элементов: новое соединение, новый источник данных, новая связь, новая переменная, бизнес объекты. Подробное описание смотрите в разделе Меню Новый элемент
Изменить (Edit)	Предоставляет возможность редактировать созданные элементы. При нажатии кнопки запускается редактор выделенного элемента в дереве Словаря данных
Удалить (Delete)	Удаляет выделенный элемент в дереве Словаря данных
Вверх (Up), Вниз (Down)	Перемещает выделенный элемент в дереве Словаря данных вверх или вниз
Настройки	Открывает Панель настройки словаря (Dictionary Setting Panel).

Дерево данных (Tree Data)

Дерево данных (Tree Data) представляет собою список всех данных словаря, которые отображаются в виде дерева.

Вид дерева данных:



По умолчанию дерево содержит три узловых элемента:

- Источники данных - где указывается, откуда будут браться данные для построения отчета. По умолчанию в проект уже добавлен источник данных - текущий проект. Порядок работы с источниками данных смотрите в разделе Источники данных.
- Системные переменные - это переменные, которые сообщают, какую-либо информацию о текущем состоянии отчета. Список всех доступных переменных смотрите в разделе Системные переменные
- Функции - в этой категории представлены элементы, при помощи которых можно рассчитать определенный результат или вернуть необходимое значение. Все элементы категории Функции разделены по группам. В разделе Функции и расчет итогов представлен перечень функций, их описание и краткие примеры.

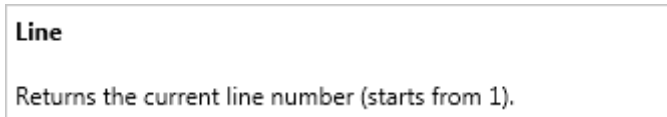
В процессе разработки отчета может быть добавлена еще одна категория Переменные. В эту категорию добавляются локальные переменные.

Важно! Если выделить элемент в дереве, то в панели свойств будут отображаться его свойства.

Панель описания (Description Panel)

Панель описания (Description Panel) отображает краткое описание выделенной системной переменной или функции.

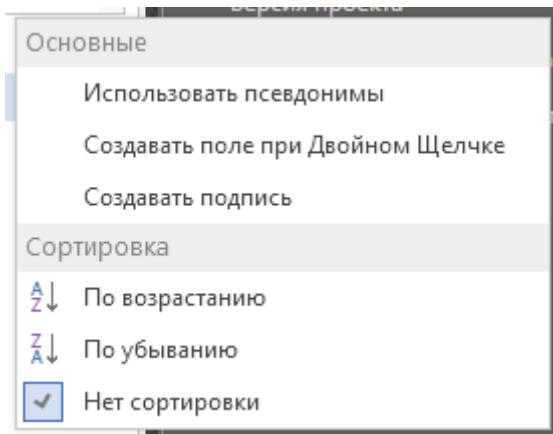
Вид панели для системной переменной Line:



Панель настройки словаря (Panel Setting Dictionary)

Панель настройки словаря (Panel Setting Dictionary) представляет собой панель с тремя параметрами, позволяющими оптимизировать работу со словарем и его содержимым. Панель открывается после нажатия на соответствующую иконку в панели инструментов словаря (описание выше).

Вид панели:



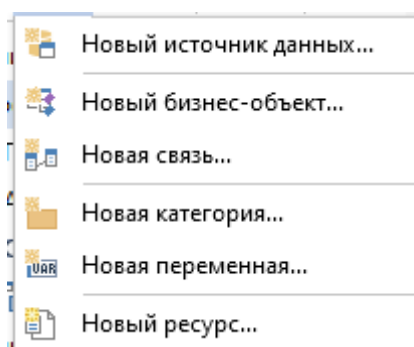
Команда	Назначение
Группа Основные	
Использовать псевдонимы (Use Aliases)	Предоставляет возможность вместо Имя (Name) отображать Псевдоним (Aliases) компонентов в шаблоне отчета
Создавать поле при Двойном Щелчке (Create Field on Double Click)	Предоставляет возможность создавать поля на бэнде Данные (DataBand). Поля создаются на том бэнде, у которого выбран источник данных, по полям которого производится двойной щелчок в словаре.

Команда	Назначение
Создавать подпись (Create Label)	Присоединяет заголовок колонки данных при её перетаскивании на шаблон отчета.
Группа Сортировка	Определяет порядок расположения элементов в словаре. Рекомендуется устанавливать значение Нет сортировки

9.2.4.1.1. МЕНЮ НОВЫЙ ЭЛЕМЕНТ

В меню Новый элемент (New Item) расположены команды, при помощи которых можно добавить новые элементы в словарь данных отчета.

Вид меню:

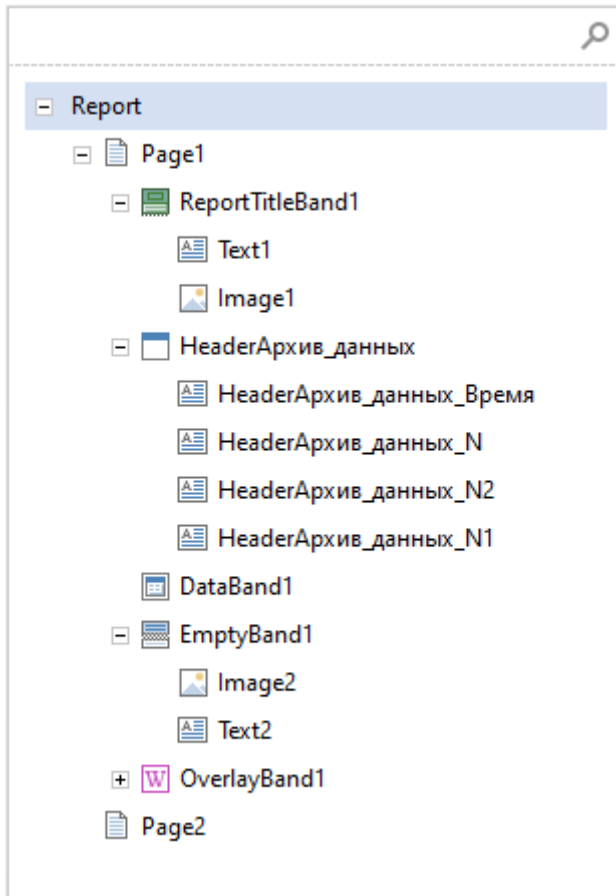


Название	Назначение
Новый источник данных... (New Data Source...)	Используется для добавления нового источника данных для отчета. Тип создаваемого источника данных зависит от типа соединения. При использовании данной команды запускает мастер создания нового источника данных, который предоставляет возможность добавить более одной таблицы данных в словарь данных отчета. Стоит учитывать, что это всего лишь метод описания источника данных.
Новый бизнес-объект... (New Business Object...)	Не используется

Название	Назначение
Новая связь... (New Relation...)	Добавляет новую связь. Стоит обратить внимание, что связи могут быть созданы только между источниками данных.
Новая категория... (New Category...)	Добавляет новую категорию локальных переменных. Все локальные переменные организованы в двухуровневую структуру, где переменная может находиться как в основном списке, так и в категории, которая находится в этом основном списке. Такую категорию можно создать при помощи данной команды.
Новая переменная... (New Variable...)	Добавляет новую локальную переменную. Если в момент вызова данной команды, была выделена какая-либо категория переменных в словаре данных, то переменная будет создана в этой категории. Если никакая категория в словаре данных не была выделена или был выделен элемент Переменная в словаре данных, то новая переменная будет создана в верхнем уровне списка локальных переменных переменных.
Новый ресурс	Добавляет графический ресурс в словарь данных. Затем можно использовать его в отчете.

9.2.4.2. ДЕРЕВО ОТЧЕТОВ

Панель Дерево отчета (Report Tree) отображает иерархию отчета, т.е. представляет все компоненты отчета в виде дерева. Кроме того, если у компонента добавлен обработчик событий, то он также будет отображен в иерархии отчета. На рисунке снизу приведен пример панели Дерево отчета (Report Tree):



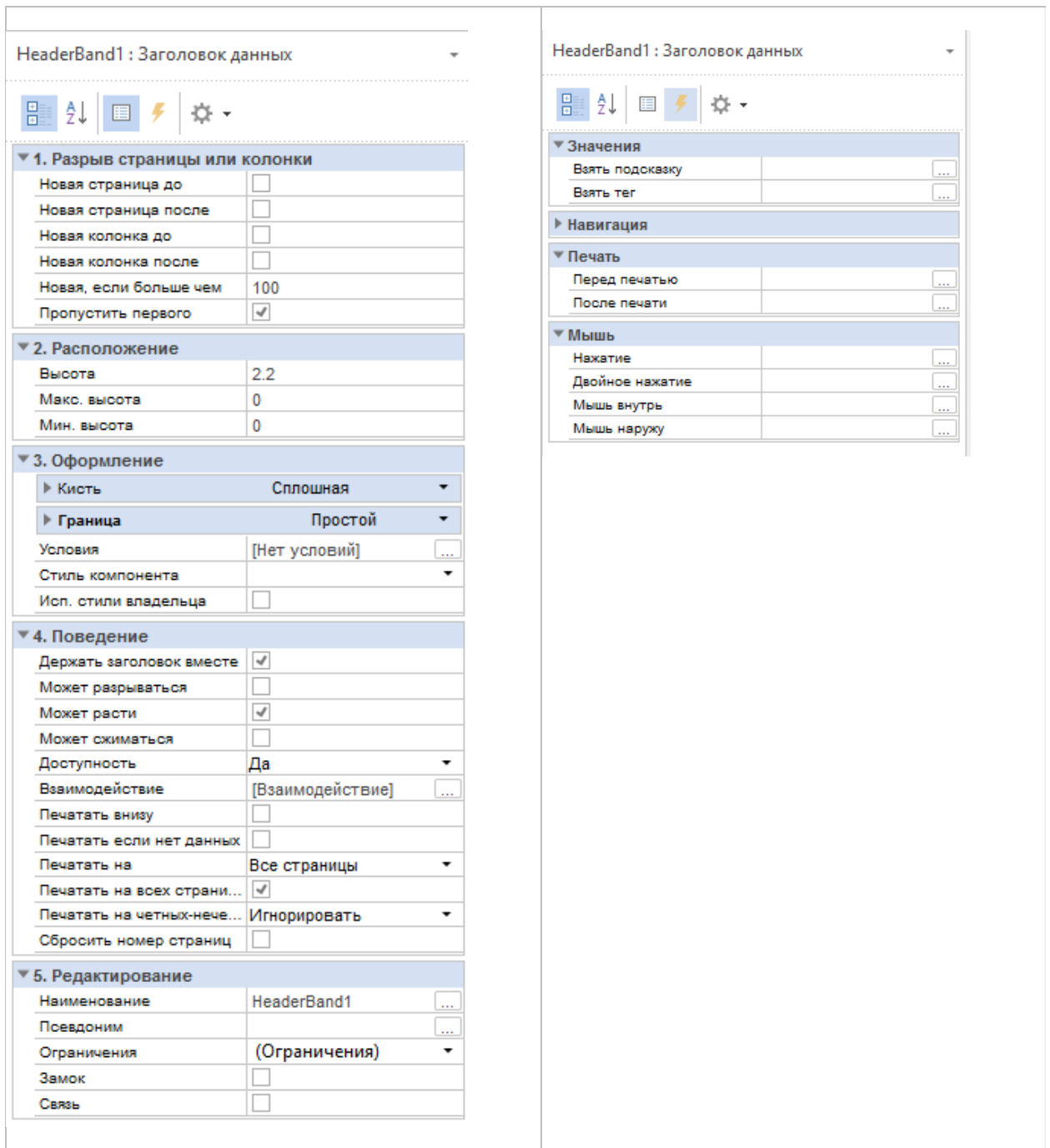
Как видно из рисунка, представлена иерархия отчета по принципу "вложенности". Панель Дерево отчета (Report Tree) предоставляет возможность визуально определить подчинение компонентов той или иной секции (бэнду)/панели. Первым дочерним элементом секции располагается тот элемент, который находится выше остальных относительно оси Z.

Если выделить элемент в дереве, то он выделится и на странице отчета и в панели свойств будут отображаться его свойства.

В верхней части панели есть строка поиска, если ввести в нее символы и нажать на кнопку поиска, то будет выделен ближайший к выделенному на момент начала поиска нижний элемент, содержащий данный набор символов.

9.2.4.3. СВОЙСТВА

Панель Свойства (Properties) отображает все свойства выбранного компонента, а также его события. На рисунке снизу показана панель Свойства (Properties), показывающая свойства бэнда Заголовок данных (рисунок слева) и его события (рисунок справа):

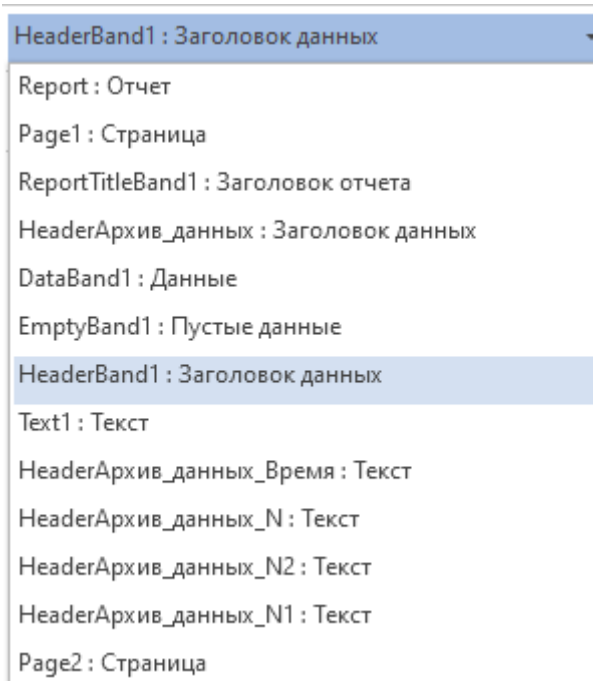


Панель Свойства (Properties) включает в себя: Выпадающий список компонентов (Dropdown Components List), Панель инструментов свойств (Properties ToolBar), Таблицу свойств (Properties Tab) или событий (Events Tab), Панель описания (Description Panel). Рассмотрим эти составляющие более подробно

Выпадающий список компонентов (Dropdown Components List)

Выпадающий список компонентов (Dropdown Components List) отображает список всех компонентов отчета, находится в верхней части панели.

Пример:



Список представлен двумя колонками. В первой колонке отображается Имя (Name) компонента, а во второй его тип. К примеру, строка Text1 : Text, где Text1 - это Имя (Name), Text - это тип. Если в данном списке выделить компонент, то на панели Свойств (Properties) отобразятся свойства и события выделенного компонента.

Панель инструментов свойств (Properties ToolBar)

Панель инструментов свойств (Properties ToolBar) предназначена для управления панелью Свойства (Properties).

Вид:




Кнопка	Назначение
Сортировка по Категориям (Categorized)	Если кнопка нажата, то представленный перечень свойств или событий сортируется по категориям
Сортировка по Алфавиту (Alphabetical)	Если кнопка нажата, то представленный перечень свойств или событий сортируется по алфавиту от А до Я (from A to Z)
Таблица свойств (Properties Tab)	Если кнопка нажата, то отображаются свойства выделенного элемента


Кнопка	Назначение
Таблица событий (Events Tab)	Если кнопка нажата, то отображаются события выделенного элемента
Настройки	<p>Определяет какой набор свойств будет отображаться. Возможные варианты:</p> <ul style="list-style-type: none"> • Базовый • Стандартный • Расширенный <p>Если установлено расширенный, то будут отображаться все возможные свойства (рекомендуется)</p>

Таблица Свойств (Properties Tab) / Событий (Events)

Таблица Свойств (Properties Tab) или Событий (Events) представляет собой таблицу с двумя колонками. В первой колонке отображается название свойства или события, во второй колонке вводятся или выбираются значения этого свойства или события. Количество строк зависит от количества свойств или событий, т.к. одно свойство или событие занимает одну строку. На рисунке снизу представлены таблица свойств (рисунок слева) и таблица событий (рисунок справа):

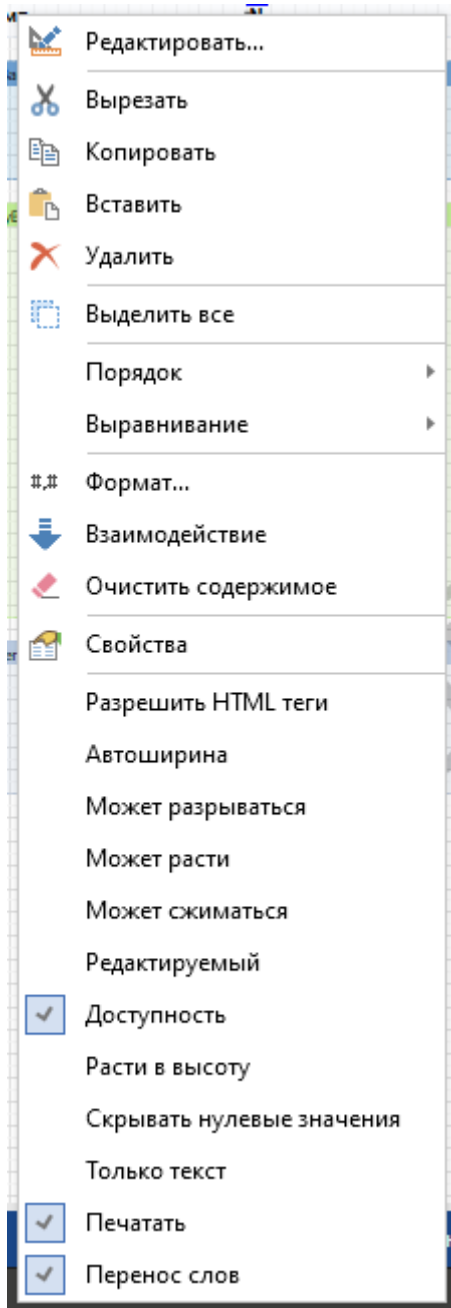
▼ Value Events	
Get Excel Value	...
Get Value	...
Get Tool Tip	...
Get Tag	...
▼ Navigation Events	
Get Hyperlink	...
Get Bookmark	...
▶ Print Events	
▶ Mouse Events	
▶ 5. Behavior	
▶ 6. Design	
▶ 7. Export	
▶ 8. Misc	
▼ Value Events	
Get Excel Value	...
Get Value	...
Get Tool Tip	...
Get Tag	...
▼ Navigation Events	
Get Hyperlink	...
Get Bookmark	...
▶ Print Events	
▶ Mouse Events	

Значение свойства можно ввести непосредственно в его строку, либо можно нажать на кнопку  и тогда откроется диалоговое окно для ввода свойства. Для каждого типа свойства будет свое окно.

В свойства отвечающие за вывод текста в отчете можно задавать выражения. Все выражения в Панели Свойств (Properties Panel) помечаются специальным значком .

9.2.5. КОНТЕКСТНОЕ МЕНЮ

Контекстное меню (Context Menu) в графическом интерфейсе пользователя — меню, открываемое, как правило, при нажатии правой кнопки мыши. В этом меню отображаются команды, которые предоставляет объект в текущем контексте, находившийся под указателем в момент его вызова, его свойства и общие команды. Наличие у объекта контекстного меню и его содержимое зависят от рабочей среды и конкретного компонента. На рисунке снизу приведен пример контекстного меню текстового компонента шаблона отчета:



Рассмотрим порядок расположения команд в меню:

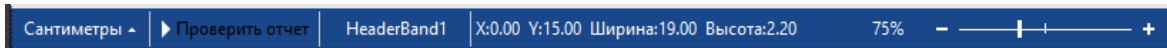
Название	Назначение
Общие команды - находятся в верхней части контекстного меню и доступны любому компоненту	
Редактировать (Design)	Вызывает редактор выделенного компонента. К примеру, если выделен текстовый компонент то будет вызван Редактор текста (Text Editor).

Название	Назначение
Вырезать (Cut)/Копировать (Copy)/ Вставить (Paste)	Стандартные команды для работы с буфером обмена
Удалить (Delete)	Удаляет выделенный компонент
Размер (Size)	Доступна при одновременном выделении двух или нескольких компонентов. Данная команда содержит подменю, в котором можно определить одинаковые параметры размеров для всех выделенных компонентов. Образцом размеров будут размеры компонента с которого было начато выделение. Если выделить все компоненты на странице, т.е. выбрать команду Выделить все (Select all) в контекстном меню, или использовать горячие клавиши Ctrl+A, то прототипом размеров для всех компонентов будут размеры того компонента, который располагается на более высоком уровне и выше остальных компонентов в дереве отчета на данном уровне.
Выделить все (Select All)	Выделяет все компоненты на текущей странице.
Порядок (Order)	Вызывает подменю, в котором можно определить место размещения выделенного компонента по оси Z.
Выравнивание (Align)	вызывает подменю, в котором можно различные режимы выравнивания
Индивидуальные команды - это команды, которые характерны для выделенного элемента. Находятся в средней части меню	
Команды отвечающие за свойства элементов - можно включать/выключать различные свойства, не закрывая контекстное меню выделенного компонента. Находятся в нижней части меню	

9.2.6. СТАТУСНАЯ ПАНЕЛЬ

Статусная панель располагается внизу окна дизайнера и представляет собой панель, на которой расположены элементы управления.

Вид:



Элементы управления:

Элемент	Назначение
Единицы измерения	Данное поле отображает используемые единицы измерения в отчете, а также позволяет изменять их.
Инспектор отчетов (Report Checker)	Проверяет отчет на наличие ошибок, уведомлений, сообщений и оповещает пользователя. См. раздел Инспектор отчета
Выделенный компонент	Отображает выделенный компонент
Значения курсора	Это поле отображает координаты курсора на странице шаблона отчета, а также координаты и размеры выделенного компонента. Причем, начало отсчета (X:0,0 и Y:0,0) совпадает с верхним левым углом пересечения границ на странице шаблона отчета.
Управление масштабом	Определяет масштаб страницы шаблона отчета. Двигая ползунок влево и вправо можно управлять масштабом.

9.2.6.1. ИНСПЕКТОР ОТЧЕТА



Для того, чтобы проверить отчет на наличие ошибок следует воспользоваться Инспектором отчета (Report Checker). Инспектор отчетов произведет анализ отчета, в результате чего выдаст сообщение об ошибках, замечаниях, неточностях найденных в данном отчете.

Вид:

❌ Ошибки - 0 | ⚠️ Предупреждения - 1 | ℹ️ Информационные сообщения - 0 | 🗨️ Сообщения построения отчета - 0

1. Внимание! Компонент частично за пределами бэнда.

Компонент 'Text2' частично или полностью находится за пределами бэнда. Компонент с таким расположением может быть выведен на печать неверно. Однако, в некоторых случаях такое расположение компонента допустимо. Также возможны проблемы с экспортом отчета в другие форматы.

Удалить |   Не показывать

9.2.7. "ГОРЯЧИЕ КЛАВИШИ" В РЕДАКТОРЕ ОТЧЕТОВ

Редактор отчетов поддерживает большое количество горячих клавиш" С использованием горячих клавиш можно достаточно сильно увеличить производительность построения отчетов. Некоторые горячие клавиши дублируются командами в контекстном меню.

Горячие клавиши	Действие
Ctrl+B	Установить стиль шрифта Жирный для выделенного текстового компонента
Ctrl+I	Установить стиль шрифта Наклонный для выделенного текстового компонента
Ctrl+U	Установить стиль шрифта Подчеркнутый для выделенного текстового компонента
Ctrl+"+"	Увеличить размер шрифта для выделенного текстового компонента
Ctrl+"-"	Уменьшить размер шрифта для выделенного текстового компонента
Ctrl+L	Установить выравнивание по левому краю для текста в текстовом компоненте
Ctrl+E	Установить выравнивание по центру для текста в текстовом компоненте
Ctrl+R	Установить выравнивание по правому краю для текста в текстовом компоненте
Ctrl+J	Установить выравнивание по ширине для текста в текстовом компоненте

Горячие клавиши	Действие
Ctrl+Shift+D	Активизировать панель “Dictionary”
Ctrl+Shift+M	Активизировать панель “Messages”
Ctrl+Shift+L	Активизировать панель “Report Tree”
F4 Shift+Enter	Активизировать панель “Properties”
Ctrl+C Ctrl+Insert	Копировать в буфер обмена
Delete Ctrl+Delete	Удалить выделенные компоненты
Ctrl+V Shift+Insert	Вставить из буфера обмена
Ctrl+X Shift+Delete	Вырезать в буфер обмена
Ctrl+A	Выделить все
Ctrl+Z	Отменить последнее действие
Ctrl+Y	Повторить отмененное действие
Ctrl+F2	Отобразить “Data Store”
Ctrl+F5	Отобразить “Services Configurator”
Ctrl+N	Создать новый отчет

Горячие клавиши	Действие
Ctrl+Shift+N	Добавить страницу в отчет
Ctrl+Shift+F	Добавить форму в отчет
Ctrl+O	Загрузить отчет из файла
Ctrl+Shift+O	Загрузить страницу из файла
Ctrl+S	Сохранить отчет
Ctrl+F12	Сохранить отчет как
Ctrl+Tab	Осуществляется переход между страницами в шаблоне отчета
F5	Предварительный просмотр отчета
Ctrl+Enter	Вызвать дизайнер по умолчанию для выбранного компонента
Enter	Вызвать текстовый редактор для выбранного компонента
F1	Выбрать инструмент “Select”
F3	Выбрать инструмент “Text Editor”
F6	Выбрать инструмент “Copy Style”
F8	Выбрать инструмент “Hand”
Cursor keys	Move selection
Shift+Cursor keys	Resize selected component (one step = grid size)
Shift+Alt+Cursor keys	Resize selected component (one step = half grid size)
Ctrl+Cursor keys	Move selected component (one step = grid size)

Горячие клавиши	Действие
Ctrl+Alt+Cursor keys	Move selected component (one step = half grid size)
Ctrl+Drag mouse	Copy selected components
Alt+Drag mouse	Ignore “Align to Grid” when moving and resizing
Ctrl+F4	Вызвать менеджер страниц

9.2.8. РАСПОЛОЖЕНИЕ КОМПОНЕНТОВ В ОТЧЕТЕ

Размещать элементы в отчете можно следующими способами:

- При помощи инструментария редактора отчетов. Для этого необходимо кликнуть нужный компонент или секцию/бэнд, а затем нажать и растянуть левой кнопкой мыши в том месте отчета, где планируется расположить компонент. Причем, если был выбран компонент, который имеет свое место в отчете, например, Заголовок страницы, то он автоматически установится на свое место. Таким способом устанавливаются все бэнды и статические элементы, т.е. элементы, которые не имеют связи с источником данных отчета.
- При помощи панели Словарь. Если взять и перетащить элемент из словаря на страницу, удерживая левую кнопку мыши, то на странице появится тот компонент или секция, который служит для отображения этого элемента словаря в отчете. Таким образом, как правило, в отчете отображают компоненты, связанные с источником данных отчета.

9.2.9. ПРЕДВАРИТЕЛЬНЫЙ ПРОСМОТР ОТЧЕТА В ДИЗАЙНЕРЕ

Редактор отчетов позволяет выполнить предварительный просмотр отчета перед запуском в режим исполнения, для того чтобы выявить возможные ошибки, недочеты.

Выполнить предварительный просмотр отчета можно с помощью закладки Предварительный просмотр (Preview). Также можно выполнить предварительный просмотр отчета можно в отдельном окне выювера. Для этого необходимо использовать горячую клавишу F5. Предварительный просмотр (Preview), в закладке либо отдельном окне, отображает построенный отчет согласно определенным параметрам, к примеру, тип бумаги, границы страницы и т.д.

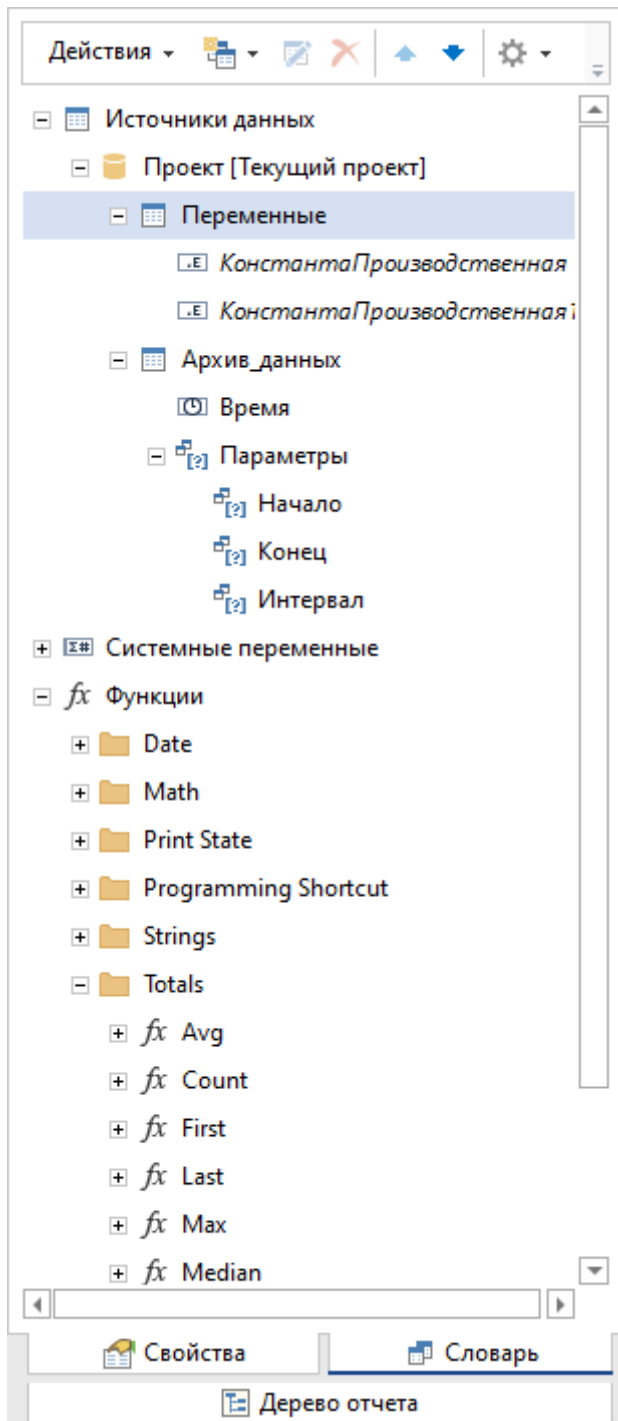
Рекомендуется проверку отчета осуществлять в исполнительной системе, совместно с функциональным блоком ReportFB

9.3. ИСТОЧНИКИ ДАННЫХ, ПЕРЕМЕННЫЕ И ФУНКЦИИ

В данном разделе представлена информация по настройке исходных данных поступающих в отчет - данных, на основе которых будут формироваться страницы отчета.

В данном разделе представлена информация по работе со Словарем данных отчета. На примерах показан процесс создания соединения с базой данных из дизайнера отчетов, создание источника данных, работа с переменными и их применение в отчете.

Каждый отчет содержит Словарь данных. Словарь данных содержит информацию о данных, которая используется для построения отчетов. К этой информации относятся: соединения с базами данных, источники данных и связи между ними и переменные. Также Словарь данных отчета может не иметь какой-либо информации о данных, но при этом отчет будет построен. Словарь данных отчета отображается на панели Словарь данных (Dictionary):



9.3.1. ИСТОЧНИКИ ДАННЫХ

Источник данных (DataSource) - это структурное описание данных, используемых для построения отчета. Источник данных (DataSource) является, как бы программной "прослойкой", которая обеспечивает получение данных из архива исполнительной системы или базы данных, преобразование их и передачу данных в генератор отчетов. Другими

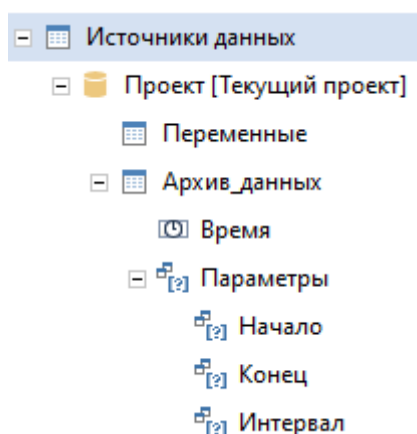
словами, Источник данных (DataSource) - это описание способов, параметров и методов доступа к данным.

Для построения отчета могут быть использованы следующие источники данных:

- данные из архива MasterSCADA 4D. Причем база данных для хранения архива данных может быть выбрана на любом этапе разработки проекта. Работа отчета от выбранной базы данных для хранения архива данных не зависит.
- данные из произвольной базы данных типа MSSQL и Postgre. В этом случае отчет независимо обращается напрямую к базе данных и получает из нее данные. Разработчик проекта должен сам создать базы данных и обеспечить доступ к этим базам данных.

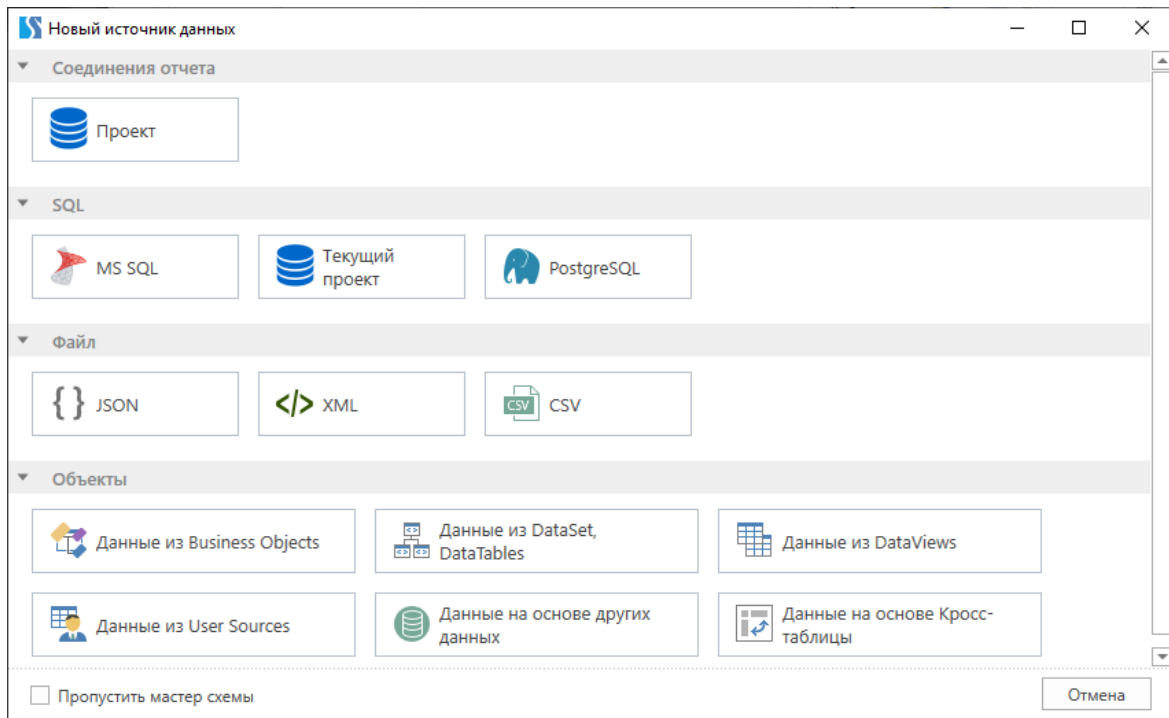
Важно! Описание данных не содержит реальных данных. Заполнение данными осуществляется в момент построения отчета в исполнительной системе

По умолчанию при создании отчета к нему уже подключена группа источников данных Проект [Текущие проект], которая умеет работать с архивными и текущими значениями параметров MasterSCADA 4D:



Порядок работы с этим источником данных описан в разделе Проект MasterSCADA.

Для создания другого источника данных следует выбрать команду Новый источник данных в меню Новый элемент словаря данных или из контекстного меню. Откроется окно:



В окне содержатся следующие категории

Название	Описание
Соединения отчета	Отображаются источники данных, которые уже добавлены в отчет
SQL	Отображаются базы данных, с которыми можно установить связь.
Файл	Перечислены файлы, из которых отчет может получать данные. При использовании этого источника данных разработчик проекта должен обеспечить доступность файлов на устройстве, где будет работать исполнительная система.
Объекты	Дополнительные источники данных.

Для работы некоторых источников данных например, для DataSet необходимо установить связи.

Для работы со сторонними базами данных необходимо выполнить подключение к ним и настроить запросы.

Флаг Пропустить мастер схемы (Skip Schema Wizard)

При создании источника данных, получить из хранилища данные можно следующими способами:

Получить схему данных. В этом случае, отобразится иерархический список данных в виде таблиц, представлений, хранимых процедур и т.д. Пользователю следует отметить необходимые источники флажками. При этом, если данных в хранилище огромное количество, то предпочтительней воспользоваться вторым способом;

Сформировать запрос для получения данных. Более подробнее прочитать о запросах можно [здесь](#).

Определить способ получения данных можно при помощи флага Пропустить мастер схемы (Skip Schema Wizard). Если необходимо получить схему данных, то следует снять флаг. Если же необходимо перейти к созданию запроса, установите флаг. Стоит учитывать, что из формы получения схемы данных, также можно перейти к созданию запроса, нажав кнопку Новый запрос (New Query).

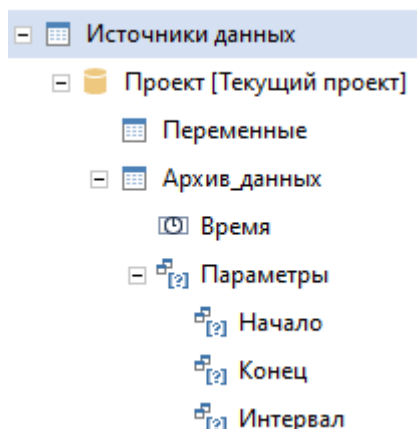
После того, как будет установлено соединение, в зависимости от типа источника данных и значения параметра Пропустить получение схемы данных (Skip Schema Wizard), будет вызвана форма создания источника данных.

9.3.1.1. ПРОЕКТ MASTERSCADA

Основным источником данных в редакторе отчетов является Проект MasterSCADA 4D.

Данный источник предоставляет интерфейс связи с накопленным архива данных и текущими значениями параметров. Место сохранения архивов значения не имеет для использования этого источника данных. Место хранения архивов может быть изменено в процессе работы, при этом изменения в отчет вносить не нужно.

По умолчанию при создании отчета к нему уже подключена группа источников данных Проект [Текущие проект], который умеет работать с архивными и текущими значениями параметров MasterSCADA 4D:



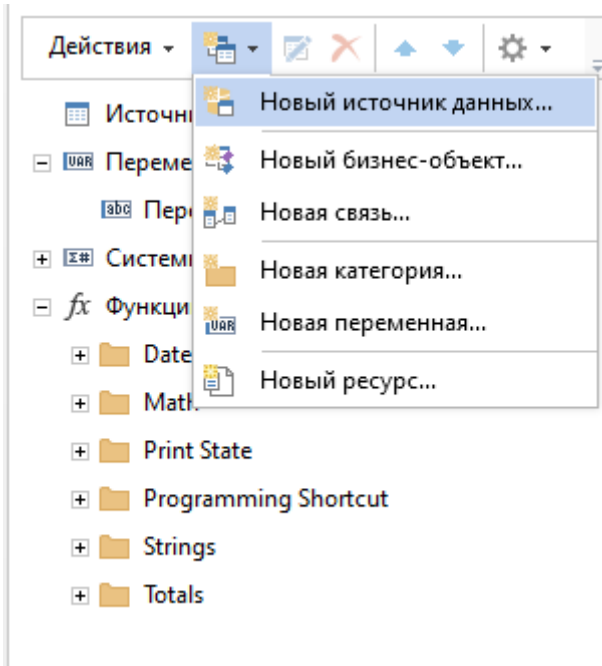
В Словарь добавляется группа, которая состоит из двух источников данных:

- Переменные - служит для установки связи с параметрами, когда в отчет нужно поместить текущее значение параметра на момент формирования отчета.
- Архив данных - используется для установки связи с параметрами, когда в отчет нужно поместить значение, рассчитанное или полученное на основе архива данных MasterSCADA 4D. При формировании отчета на основе архивных значений нужно особое внимание уделить настройкам элемента отчета Архив данных. Именно от настроек зависит способ формирования отчета (периодический, итоговый, по изменению и т.п.)

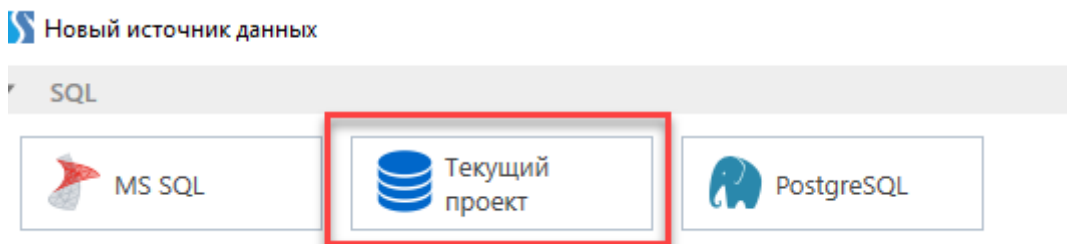
Также если в источник данных перетащить параметр, который имеет тип массив структур, то появится новый источник данных, имя его будет зависеть от имени связанного параметра. Подробное описание смотрите в разделе Массив структур в отчете.

9.3.1.1.1. СОЗДАНИЕ ИСТОЧНИКА ПРОЕКТ [ТЕКУЩИЙ ПРОЕКТ]

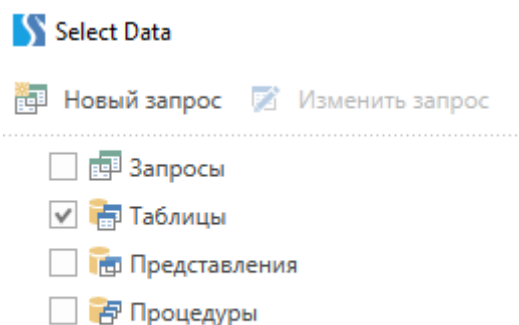
По умолчанию источник добавлен в словарь отчета. Если по каким-то причинам источник данных Проект [Текущий проект] был удален, то можно его заново добавить в отчет. Для этого нужно перейти на панель Словарь. В меню Новый элемент выбрать пункт Новый источник данных:



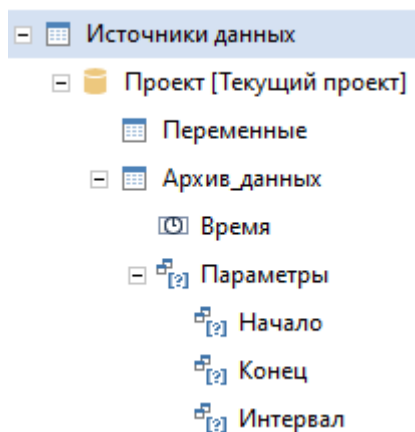
В появившемся окне нужно выбрать источник Текущий проект:



Появится окно, в котором нужно нажать на кнопку ОК:



В словарь будет добавлен источник данных:



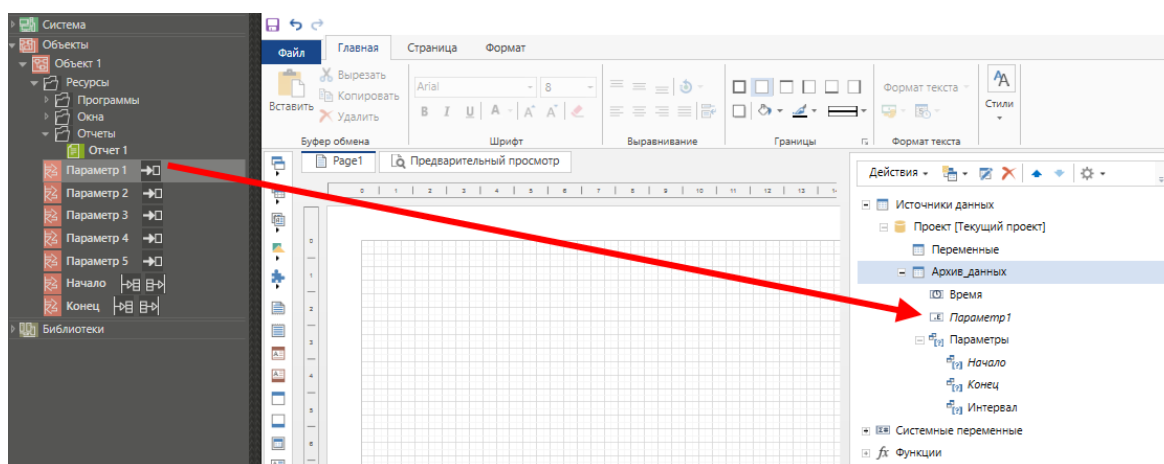
Источник создан, теперь в него можно добавлять переменные из дерева объектов.

9.3.1.1.2. АРХИВ MASTERSCADA

В данном разделе описаны основные части категории Архив данных источника данных Текущий проект отчета, и порядок работы этим элементом.

Добавление переменных

Чтобы установить связь между параметрами проекта и переменными источника данных нужно перетащить их из дерева проекта в словарь данных. Если нужны архивные значения в отчете, то переменные, у которых установлен флаг Архивировать нужно перетаскивать в Архив данных:

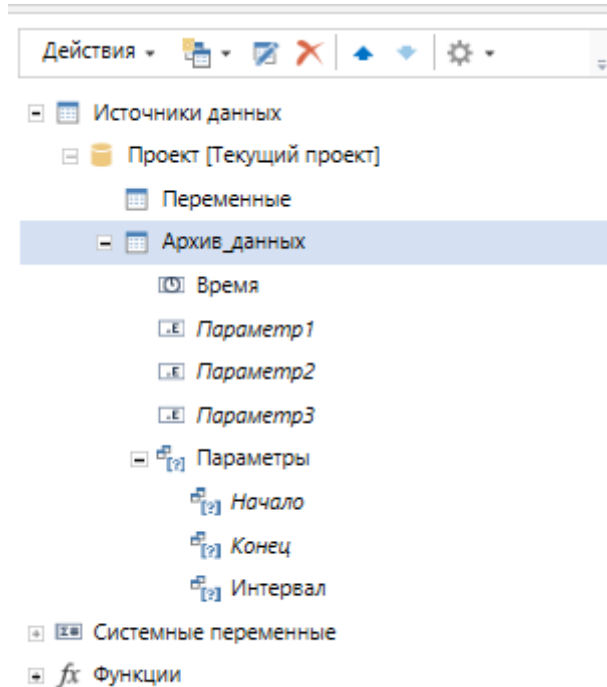


Важно! Рекомендуется добавлять переменные из объекта в котором располагается отчет или вложенных в него объектов. Это облегчает копирование объектов.

Если переменных много, допускается вариант группового переноса. Для этого выделите переменные в дереве (используя клавиши Shift и Ctrl) и также перетащите в источник. В отчет можно перетащить и отдельные элементы сложных типов данных, например, поля структуры.

В отчет можно перетащить один и тот же параметр несколько раз, но изменить для каждой полученной переменной способ обработки

Переменные добавятся в источник - теперь они называются Колонками:



По умолчанию в источник данных уже добавлена колонка **Время**, в которой будет отображаться время к которому относится та или иная строка в отчете.

После того как переменные добавлены в источник необходимо настроить их свойства.

Параметры **Начало**, **Конец**, **Интервал**

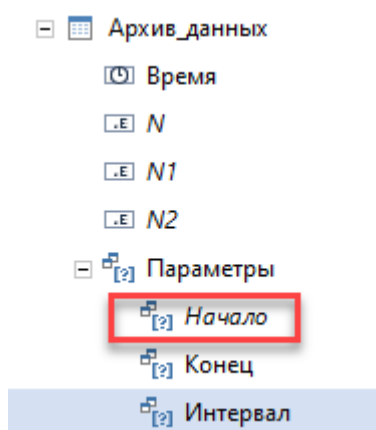
При запуске построения отчета, происходит выборка данных из архива MasterSCADA 4D. Для того чтобы ограничить данную выборку и построить отчет за нужный интервал времени предназначены параметры Источника данных - **Начало** и **Конец**. Также есть параметр **Интервал** - данный интервал предназначен для задания временного шага вывода строка, при периодическом способе формирования.

Данные параметры нужно связать с параметрами проекта, которые будут или рассчитываться в программе, или задаваться пользователем к клиенте визуализации:

Параметр	Рекомендации
Начало	Данный параметр нужно связать с параметром дерева системы или дерева объектов типа DT. Этот параметр будет влиять на то, с какого времени будет сделана выборка из архива данных

Параметр	Рекомендации
Конец	Данный параметр нужно связать с параметром дерева системы или дерева объектов типа DT. Этот параметр будет влиять на то, до какого времени будет сделана выборка из архива данных
Интервал	Можно установить связь с параметром дерева системы или дерева объектов типа Time. Либо для построения, например, годовых отчетов с интервалом месяц, можно установить константу в свойствах (Формирование отчетов по периодической выборке значений)

Для связи параметра проекта и параметра отчета нужно перетащить параметр проекта на параметр отчета. Если связь будет установлена, то изменится шрифт параметра отчета. Если шрифт наклонный, то связь установлена:



При наведении на параметр, если связь установлена, то появится всплывающая подсказка, в которой отобразится полное имя связанного параметра. Также связь будет отображаться в панели свойств параметра отчета (чтобы открыть панель свойств параметра нужно выделить его в словаре и перейти на панель свойств)






Важно! В дереве MasterSCADA 4D связь с параметрами отчета не отображается!

Свойства источника данных Архив MasterSCADA

Для того, чтобы определить тип отчета, какие значения из архива будут использованы необходимо настроить панель свойств источника данных. Чтобы перейти на панель свойств необходимо выделить элемент в словаре, а затем переключиться на панель свойств отчета.

Вид панели свойств:

Report : Отчет

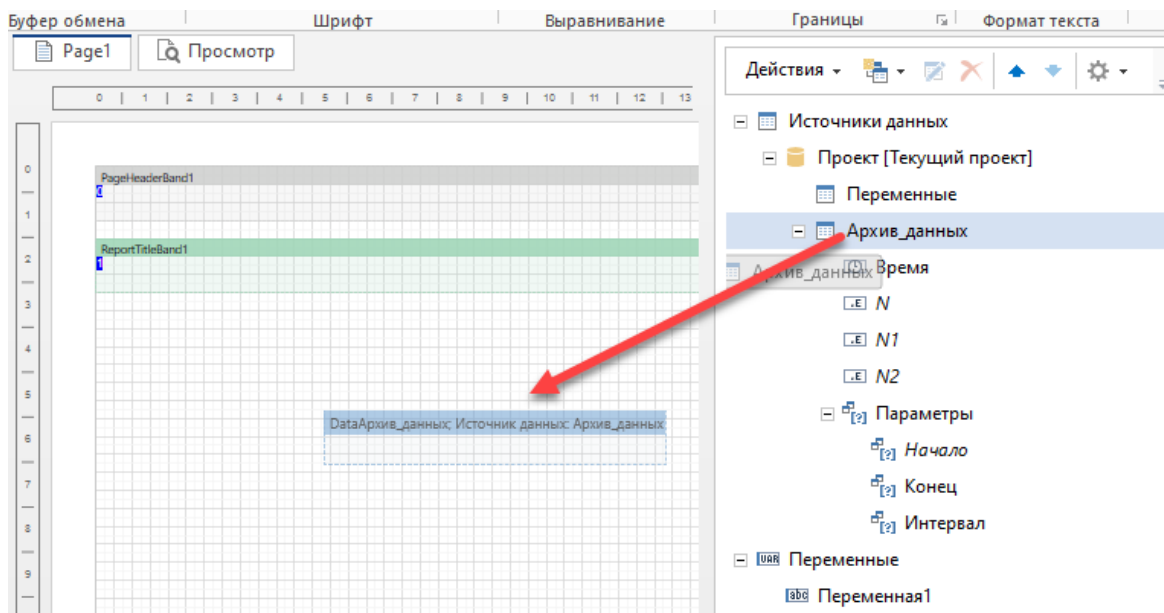
1. Обработка данных	
Формирование значений	По изменению
Предпочитаемый слой	Основной
Использовать другие слои	<input checked="" type="checkbox"/>
Преобразовывать Nulls	Наследовать
Конвертировать недоста...	<input type="checkbox"/>
Мертвая зона (с)	0

Название	Назначение
Формирование значений	<p>Определяет принцип формирования строк. Существует три способа формирования:</p> <ul style="list-style-type: none"> • По изменению – в этом случае каждой изменение каждой переменной, у которых включена настройка основной, формирует новую строку. Количество строк в отчете заранее предусмотреть нельзя. • Периодический – в этом случае строки формируются с заданным у параметра Интервал временем. Строк в отчете будет столько, сколько интервалов уместится на отрезке времени между значениями заданными на параметры Начало и Конец <ul style="list-style-type: none"> • Итоговый - является частным случаем периодического. Когда Интервал равен единице, это интервал от начала до конца формирования отчета, т.е. сформируется только одна строка.
Предпочитаемый слой	<p>При архивировании используются слои данных. В редакторе отчетов можно выбрать предпочитаемый слой, из которого будут получены данные.</p>
Использовать другие слои	<p>При использовании прореженных слоев (часовых, суточных), может возникнуть ситуация, что данных на необходимый момент времени нет. Настройка определяет – брать ли в этом случае редактору отчетов данные из другого слоя.</p>
Преобразовывать Nulls	<p>Определяет, нужно ли преобразовывать пустые значения Nulls источника данных в нули. По умолчанию</p>

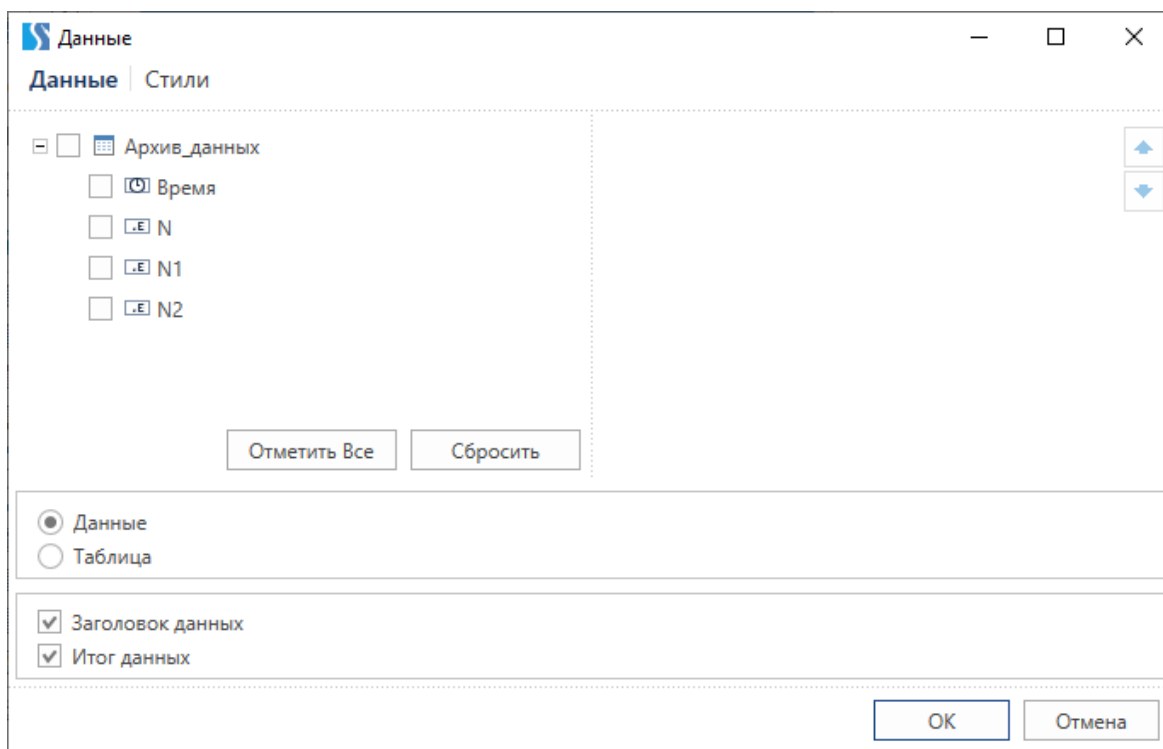
Название	Назначение
	используется режим Наследовать из отчета в этом случае настройка наследуется самого отчета.
Конвертировать недостоверные значения в Nulls	Если колонка имеет недостоверное значение, то при включении данной настройки результат колонки будет преобразован в Null (пустое значение).
Мертвая зона	Данная настройка настраивается только при способе формирования По изменению. Если несколько переменных были сохранены в архиве приблизительно в одно время, то для представления пользователю желательно считать эти измерения одновременными. Если время записей разных переменных не превышает величины мертвой зоны, то они считаются одним моментом времени, а сами эти значения будут записаны в одну строку таблицы.

Архив данных на странице отчета

Архив данных можно перетащить левой кнопкой мыши на страницу отчета:



В момент отпускания кнопки мыши откроется диалоговое окно настройки секции Данные, которое позволит автоматически создать все необходимые секции/бэнды и компоненты для формирования отчета на основе архивных данных в исполнительной системе:



В верхней части окна представлены колонки, которые могут отображаться в секции. Нужно флагами отметить те из них, которые необходимы.

В средней части окна определяется какой компонент будет использоваться Данные или Таблица (рекомендуется использовать Данные)

В нижней части определяется необходимость установки секции Заголовок данных и Итог данных на страницу отчета.

После нажатия на кнопку ОК все необходимые элементы будут добавлены на страницу шаблона отчета.

Дополнительные колонки в отчете

Если необходимо добавить дополнительные колонки, например, от других источников данных или переменных или рассчитываемые, то необходимо вручную добавить на бэнд Данные компонент Текст и в нем установить связь нужным элементом или выполнить расчет - сформировать выражение.

9.3.1.1.2.1. СВОЙСТВА ПЕРЕМЕННЫХ АРХИВА ДАННЫХ

Панель свойств для Переменных Архива данных источника данных отчета Текущий проект имеет вид:

1. Данные	
ID связи	59405
Имя связи	.././././form_ot/N 1
Дополнительный путь	
Форматировать значение	<input type="checkbox"/>
Начальное значение	0
Наименование	N1
Тип	double
2. Прочие	
Основной	<input checked="" type="checkbox"/>
3. Обработка данных	
Обработка	Последнее значение
Атрибут данных	Значение
Учитывать границы инте...	<input checked="" type="checkbox"/>
4. Обработка плохих признаков	
▶ Обработка признака Stop	
▶ Обработка признака Bad	
▶ Обработка признака Uncertain	

Чтобы открыть панель свойств переменной ее нужно выделить в словаре, а затем переключиться на панель свойств.

Добавленные в источник данных Архив данных переменные (их также называют колонками) имеют набор свойств позволяющих гибко настроить способы получения и обработки данных из архива MasterSCADA 4D, а также формирование строк данных:

Название	Назначение
Категория Данные	
ID связи	Указывается ID связи.
Имя связи	Указывается путь к параметру дерева проекта, состоящее из программных имен элементов проекта. Заполняется автоматически после того как параметр перетаскили в словарь
Дополнительный путь	Если в отчет перетаскили элемент сложного типа данных, например, поля структуры. То в данном поле будет отображаться путь к этому элементу внутри сложного типа

Название	Назначение
	данных. Заполняется автоматически после того как параметр перетащили в словарь
Форматировать значение	
Начальное значение	Можно указать начальное значение, которое будет отображаться в отчете, пока нет связи со связанным параметром в режиме исполнения
Наименование	Имя переменной, которое отображается в словаре.
Тип	Тип данных переменной (наследуется от связанного параметра проекта)
Категория Прочие	
Основной	Используется, если способ формирования установлен По изменению. Если данная настройка включена, то каждое изменение этой переменной приводит к созданию строки данных в отчете.
Категория Обработка данных	
Обработка	Определяется способ обработки значений параметра, полученных из архива. При построении периодического или итогового отчета на каждый конкретный интервал времени может приходиться от нуля до нескольких значений переменной, поэтому необходимо задать способ вычисления результата – какое из значений должна возвращать переменная источника (колонка данных). Возможные варианты значения свойства описаны ниже
Атрибут данных	<p>Определяет какой из атрибутов переменной выводится в отчет. Каждая переменная источника имеет три атрибута:</p> <ul style="list-style-type: none"> • Значение – значение переменной в архиве. • Время – метка времени переменной в архиве (в локальном времени)

Название	Назначение
	<ul style="list-style-type: none"> Признак качества - признак качества переменной в архиве.
Учитывать границы интервала	Определяет будут ли значения полученные на границах интервала учитываться при расчете значений.
Метод интегрирования	Определяет способ расчета для обработки Интеграл и Интегральное среднее (см. разделы Обработка Интеграл и Обработка Интеграл среднее)
Параметр обработки	Используется при Обработка Интеграл
Категория Обработка плохих признаков	Определяет поведение отчета если были получены недостоверные данные. Подробное описание в разделе Обработка плохих признаков качества

Свойство Обработка

Свойство Обработка может принимать следующие значения:

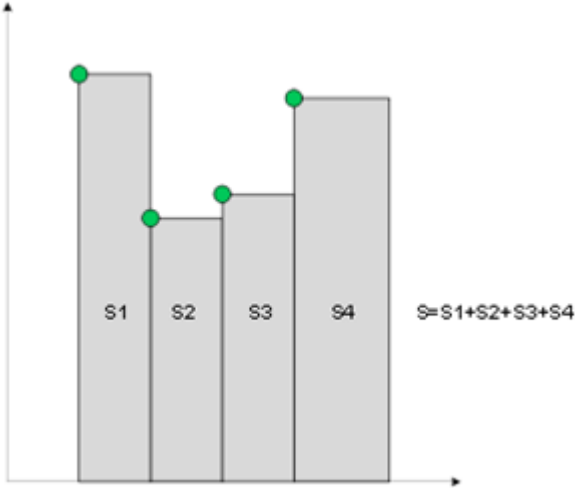
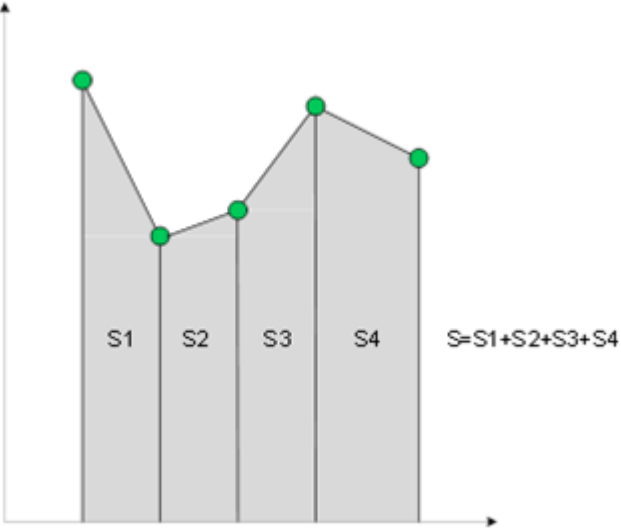
Значение	Описание
Последнее значение	Будет выбрано последнее из имеющихся значений
Первое значение	Будет выбрано первое из имеющихся значений
Интеграл	Будет вычислена площадь фигуры ограниченной графиком значений (см. раздел Обработка Интеграл). Данный способ обработки применяется, например, для вычисления суммарного расхода по действующему значению
Среднее	Будет вычислено среднее арифметическое значение
Интегральное среднее	Будет вычислено среднее интегральное значение (см. раздел Обработка Интегральное среднее).
Максимум	Будет выбрано максимальное значение на интервале

Значение	Описание
Минимум	Будет выбрано минимальное значение на интервале
Интерполяция	Не поддерживается в текущей версии
Точное значение	Используется для формирования отчета по изменению. Если выбран данный способ, то значение из этой колонки будет записано только в том случае, если оно в точности (до величины мертвой зоны) получено в этот момент времени. Иначе будет записан ноль или клетка таблицы останется пустой.
Количество включений	Определяет количество переходов из False в TRUE переменной на интервале времени
Количество значений	Будет возвращено количество всех значений на интервале
Сумма значений	Будет вычислена сумма всех значений на интервале
Разница	Будет вычислена разность между первым и последним значением на интервале. Данный способ обработки используется, например, если нужно вычислить значение энергии от счетчика с нарастающим итогом.

ОБРАБОТКА ИНТЕГРАЛ

Способ обработки Интеграл предназначен для вычисления площади фигуры ограниченной графиком из значений архива. Данный способ обработки применяется для вычисления суммарного расхода, по его действующему значению (например, вычисление расхода газа за час по действующему значению расхода).

При выборе данного способа обработки необходимо указать следующие настройки:

Настройка	Описание
Параметр обработки	<p>Если в данном поле указано значение отличное от нуля, то результат вычисления интеграла делится на это значение. Данная настройка предназначена для приведения значения к реальным единицам. Например, если расходомер возвращает значения расхода в м³/ч, то нужно указать в данном поле 3600, тогда колонка будет возвращать реальное значение расхода за интервал времени.</p>
Метод интегрирования	<p>Определяет способ вычисления интеграла. Иллюстрации работы программы ниже. Доступно два варианта:</p> <ul style="list-style-type: none"> • Метод левых прямоугольников – в этом случае интеграл считается как сумма прямоугольников, образованных точками значений в архиве.:  <ul style="list-style-type: none"> • Метод трапеций – в этом случае, между точками в архиве проводится линия. Интеграл считается как сумма образовавшихся трапеции 

Настройка	Описание

ОБРАБОТКА ИНТЕГРАЛЬНОЕ СРЕДНЕЕ

Способ обработки Интегральное среднее предназначен для вычисления среднего интегрального значения за интервал времени.

При выборе данного способа обработки происходит вычисление интеграла (см. Обработка Интеграл), а затем результат делится на длительность интервала в секундах.

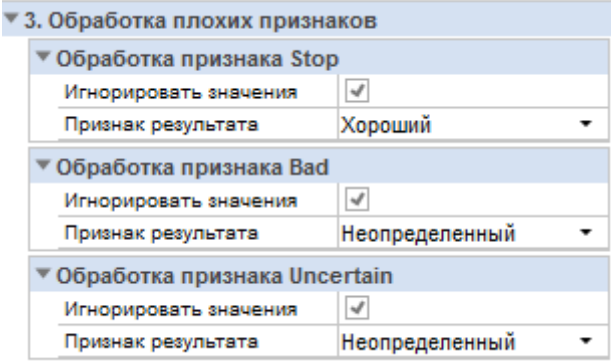
Интегральное среднее имеет такие же настройки что и обработка Интеграл, за исключением настройки Параметр обработки.

ОБРАБОТКА ПЛОХИХ ПРИЗНАКОВ КАЧЕСТВА

В реальных системах автоматизации от устройств могут поступать не только достоверные значения (с признаком качества Good), но и данные с "плохими" признаками качества (ошибка связи Bad, отказ датчика Device_Failure и т.д.), кроме того за какой-то интервал времени данных в архиве может не быть (например если исполнительная система была остановлена) - данные факторы нужно учитывать при формировании отчета.

Для каждой переменной источника данных можно настроить каким образом обрабатывать недостоверные признаки качества.

Вид категории в панели свойств переменной архива данных:



3. Обработка плохих признаков	
▼ Обработка признака Stop	
Игнорировать значения	<input checked="" type="checkbox"/>
Признак результата	Хороший ▼
▼ Обработка признака Bad	
Игнорировать значения	<input checked="" type="checkbox"/>
Признак результата	Неопределенный ▼
▼ Обработка признака Uncertain	
Игнорировать значения	<input checked="" type="checkbox"/>
Признак результата	Неопределенный ▼

Настройку обработки плохих признаков можно задать независимо для трех типов некачественных значений:

Stop – признак остановки исполнительной системы.

Bad – признак качества Ошибка. В данный признак качества входят ошибки - в него включены признаки:

- Значение недостоверно (Bad);
- Ошибка конфигурации (Config Error);
- Нет соединения (Not Connected),
- Ошибка устройства (Device Failure);
- Ошибка датчика (Sensor Failure),
- Последнее значение (Last Know);
- Нет связи (Comm Failure);
- Не обслуживается (Out of service).

Uncertain – признак качества Не определено. В этот признак входят признаки:

- Не определено (Uncertain);
- Датчик не откалиброван (Sensor Cal);
- Значение не допустимо (Egu Exceeded и SubNormal).

Для каждого типа некачественного значения существует две настройки:

Настройка	Назначение
Игнорировать значения	Если флаг установлен, то значения с недостоверным признаком (то есть отличным от Good), игнорируются при обработке интервала. Например, для обработки Максимум данная настройка определяет, какое из максимальных значений выдать - найденное недостоверное или максимальное среди достоверных.
Признак результата	<p>Задаёт, какой признак качества будет выставляться у результата, если на интервале находятся некорректные значения влияющие на результат.</p> <p>Доступны варианты:</p> <ul style="list-style-type: none"> • Хороший – результату устанавливается признак качества норма • Плохой – результату устанавливается признак качества ошибка • Неопределенный – результату устанавливается признак качества не определено. <p>Если на интервале обработки будет обнаружено значение влияющее на результат с данным признаком качества (Stop,Bad, Uncertain – в зависимости от группы настроек), то в результате колонка будет иметь признак определенный данной настройкой.</p>

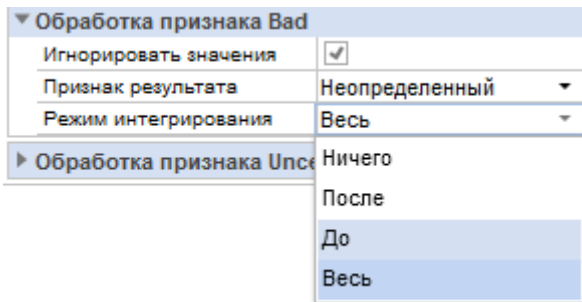
Настройка	Назначение
	<p>То есть, если в группе Обработка признаков Bad , настройка признака результата будет задана Uncertain, то при обнаружении на интервале значения с признаком качества Bad, результату колонки будет установлен признак Uncertain.</p> <p>Алгоритм определения признака результата отличается в зависимости от способа обработки параметра:</p> <p>Если используются способы обработки Первое значение , Последнее значение , Максимум , Минимум , то признак качества колонки выставляется на основании результата. Например, если способ обработки - Последнее значение , последнее значение на интервале имело признак, относящийся к Bad (Ошибка датчика , Ошибка устройства и т.д.), то признак результата определится настройкой группы настроек Bad . Если на интервале есть значение с признаком Bad, но оно не является последним, то на результат оно не влияет.</p> <p>Если используются способы обработки Среднее, Интеграл, Интегральное среднее, то признак качества колонки выставляется на основании всех значений на интервале - поскольку все значения влияют на результат. Например, если способ обработки Среднее , и на интервале было обнаружено значение с признаком качества Bad, то признак результата определится настройкой группы настроек Bad.</p> <p>Если на интервале будут обнаружены значения с разными недостоверными признаками (например: Uncertain и Bad), то результату присваивается наиболее недостоверное (группа Bad менее достоверно, чем группа Uncertain).</p>

Исключение составляет обработка Интеграл. В этом случае появляется еще одна настройка. Порядок работы и рекомендации описаны в разделе Обработка плохих признаков при вычислении интеграла

ОБРАБОТКА ПЛОХИХ ПРИЗНАКОВ ПРИ ВЫЧИСЛЕНИИ ИНТЕГРАЛА

Обработка Интеграл имеет особенности при обработке плохих признаков качества.

При включении обработки Интеграл в категории свойств Обработка плохих признаков переменной архива данных появляется еще одно свойство Режим интегрирования.



Режим интегрирования - определяет способ интегрирования плохих значений. Доступны следующие варианты значения свойства - Ничего, После, До, Весь

На интервале времени может возникнуть ситуация, когда на нем присутствуют значения с разными признаками достоверности – например, значения категории Bad и Uncertain, имеющие разные настройки. Общий алгоритм вычисления результата интегрирования следующий:

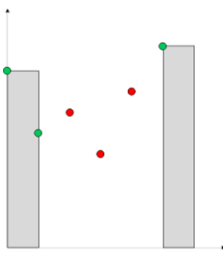
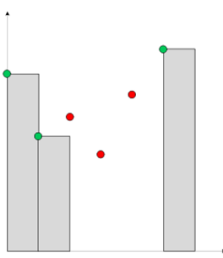
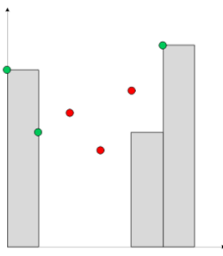
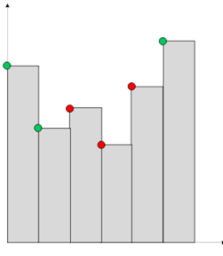
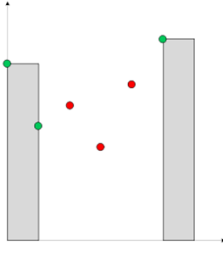
- 1) Если 2 соседние точки хорошие, то интеграл считается по этим значениям
- 2) Если одна точка плохая, другая хорошая, то учитывается настройка для плохого признака.
- 3) Если обе плохие, то область будет интегрироваться только, если настройки обеих точек указывают, что ее необходимо считать.

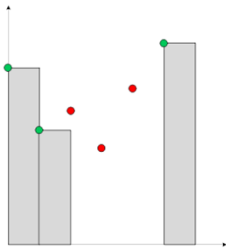
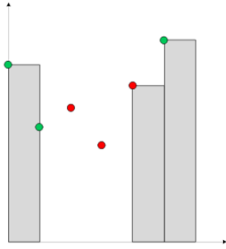
Далее приведены все возможные комбинации настроек Режим интегрирования (плохих значений) и Игнорировать значение,

Зелеными точками обозначены значения в архиве имеющие хороший признак качества, красными – плохой.

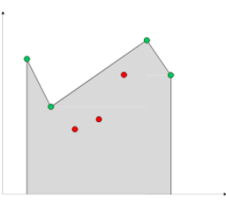
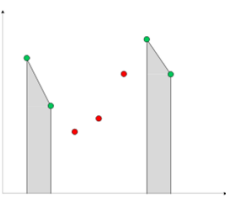
Способ вычисления – метод левого прямоугольника

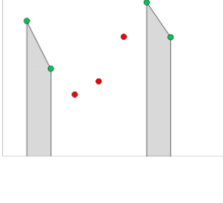
Значения настроек	Рисунок	Комментарий
Игнорировать значение - флаг установлен Режим интегрирования - Весь		В этом режиме, площадь интервала с плохими значениями считается по последнему корректному значению.

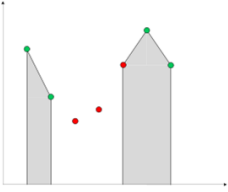
Значения настроек	Рисунок	Комментарий
Игнорировать значение - флаг установлен Режим интегрирования - Ничего		В этом режиме, площадь интервала с плохими значениями не вычисляется.
Игнорировать значение - флаг установлен Режим интегрирования - До		В этом режиме, вычисляется площадь интервала от последнего хорошего, до первого плохого значения. Площади интервалов с остальными плохими значениями не вычисляются.
Игнорировать значение - флаг установлен Режим интегрирования - После		В этом режиме, вычисляется площадь интервала от последнего плохого, до первого хорошего значения (по значению последнего хорошего значения). Площади интервалов с остальными плохими значениями не вычисляются.
Игнорировать значение - флаг снят Режим интегрирования - Весь		В этом режиме, вычисляется площадь всех интервалов независимо от признака качества.
Игнорировать значение - флаг снят Режим интегрирования - Ничего		В этом режиме, площадь интервала с плохими значениями не вычисляется.

Значения настроек	Рисунок	Комментарий
Игнорировать значение - флаг снят Режим интегрирования - До		В этом режиме, вычисляется площадь интервала от последнего хорошего, до первого плохого значения. Площади интервалов с остальными плохими значениями не вычисляются.
Игнорировать значение - флаг снят Режим интегрирования - После		В этом режиме, вычисляется площадь интервала от последнего плохого, до первого хорошего значения (по значению последнего плохого значения). Площади интервалов с остальными плохими значениями не вычисляются.

Способ вычисления – метод трапеций

Значения настроек	Рисунок	Комментарий
Игнорировать значение - флаг установлен Режим интегрирования - Весь		В этом режиме, между последним хорошим значением, и первым хорошим значения проводится прямая линия. Затем вычисляется площадь сформированной трапеции
Игнорировать значение - флаг установлен Режим интегрирования - Ничего		В этом режиме, интервалы с плохими значениями не вычисляются.

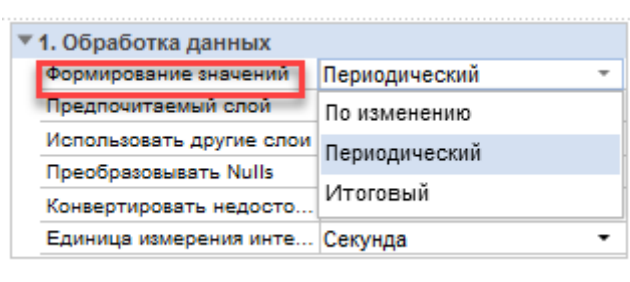
Значения настроек	Рисунок	Комментарий
<p>Игнорировать значение - флаг установлен</p> <p>Режим интегрирования - До</p>		<p>В этом режиме, между последним хорошим значением, и первым хорошим значением проводится прямая линия. Затем вычисляется площадь сформированной трапеции ограниченная последним плохим значением.</p>
<p>Игнорировать значение - флаг установлен</p> <p>Режим интегрирования - После</p>		<p>В этом режиме, между последним хорошим значением, и первым хорошим значением проводится прямая линия. Затем вычисляется площадь сформированной трапеции ограниченная последним плохим значением.</p>
<p>Игнорировать значение - флаг снят</p> <p>Режим интегрирования - Весь</p>		<p>В этом режиме, вычисляется площадь всех интервалов независимо от признака качества.</p>
<p>Игнорировать значение - флаг снят</p> <p>Режим интегрирования - Ничего</p>		<p>В этом режиме, площадь интервала с плохими значениями не вычисляется.</p>
<p>Игнорировать значение - флаг снят</p> <p>Режим интегрирования - До</p>		<p>В этом режиме, между последним хорошим значением, и первым плохим значением проводится наклонная линия, после чего вычисляется площадь сформированной трапеции. Остальные интервалы с плохими значениями игнорируются.</p>

Значения настроек	Рисунок	Комментарий
Игнорировать значение - флаг снят Режим интегрирования - После		В этом режиме, между последним плохим значением, и первым хорошим значением проводится наклонная линия, после чего вычисляется площадь сформированной трапеции. Остальные интервалы с плохими значениями игнорируются.

9.3.1.1.2.2. СПОСОБЫ ФОРМИРОВАНИЯ ОТЧЕТОВ

Существует три способа формирования отчетов по архивным данным значений переменной MasterSCADA 4D: по изменению, периодический и итоговый.

Способ формирования определяется настройкой Формирование значений элемента словаря Архив данных :



Чтобы открыть панель свойств элемента его нужно выделить в словаре, а затем переключиться на панель свойств.

ФОРМИРОВАНИЕ ОТЧЕТОВ ПО ИЗМЕНЕНИЮ ЗНАЧЕНИЙ

Мысленно представим архивные значения переменной в виде таблицы из двух столбцов, первый из которых содержит время прихода сигнала и фиксации значения, второй – соответствующее этому моменту времени значение переменной. При формировании отчета архивных значений одной переменной по изменению в данную таблицу будет добавляться новая строка каждый раз, когда будет поступать новое значение.

При формировании отчета по архивным данным двух или более переменных возникают некоторые сложности. Данные о значениях могут поступать в различные моменты времени (асинхронно), поэтому в таких случаях требуется задать некоторые условия, с помощью которых генератор отчетов сформирует итоговую таблицу, содержащую значения всех переменных.

Одним из таких условий является задание величины мертвой зоны. Если несколько переменных были сохранены в архиве приблизительно в одно время, то для представления пользователю желательно считать эти измерения одновременными. С этой целью времена записи, интервал времени между которыми не превышает величины мертвой зоны, считаются одним моментом времени, а сами эти значения будут записаны в одну строку таблицы. Для задания величины мертвой зоны необходимо в дереве словаря выделить Архивным данных и перейти к его свойствам. Далее ввести время в секундах в поле

1. Обработка данных	
Формирование значений	По изменению
Предпочитаемый слой	Основной
Использовать другие слои	<input checked="" type="checkbox"/>
Преобразовывать Nulls	Наследовать
Конвертировать недоста...	<input type="checkbox"/>
Мертвая зона (с)	0

Мертвая зона (с):

Формирование таблицы значений двух или более переменных будет при этом происходить следующим образом: каждый раз для построения новой строки таблицы из всех архивных значений, ещё незаписанных в таблицу, всех переменных генератор отчета будет выбирать момент времени максимально приближенный к началу формирования отчета (попросту самый ранний) и запишет его в первый столбец новой строки.

Может оказаться, что данному моменту времени соответствуют значения не всех переменных, и тогда в некоторых столбцах таблицы генератору отчетов будет неясно, какое же из имеющихся значений нужно записать. Для разрешения подобной ситуации необходимо задать один из двух вариантов для свойства Обработка каждой из колонок: Последнее значение и Точное значение. Если задан вариант Последнее значение, то будет выбрано последнее полученное (зафиксированное) к данному моменту времени значение переменной. Если оно отсутствует, то будет записан ноль, или клетка останется пустой.

Если же для свойства Обработка будет выбран вариант Точное значение, то в строку, соответствующую выбранному моменту времени значение из этой колонки будет записано только в том случае, если оно в точности (до величины мертвой зоны) получено в этот момент времени. Иначе будет записан ноль или клетка таблицы останется пустой.

Количество архивных значений для разных переменных может отличаться, у одной их может быть больше (значения фиксировались чаще), у другой меньше (фиксировались реже). В таком случае можно сделать так, чтобы метки времени значений одной из колонок игнорировались при выборе времени следующей строки. Для этого необходимо перейти к свойствам колонки и для свойства Основной снять флаг. Тогда данная колонка не участвует в определении времени следующей строки, и её значения будут записаны в таблицу в зависимости от настроек обработки.

▼ 1. Данные	
ID связи	82002
Имя связи	../././Параметр 2
Дополнительный путь	
Форматировать значение	<input type="checkbox"/>
Начальное значение	
Наименование	Параметр2
Тип	string ▼
▼ 2. Прочие	
Основной	<input checked="" type="checkbox"/>
▼ 3. Обработка данных	
Обработка	Последнее значение ▼
Атрибут данных	Значение ▼
Учитывать границы инте...	<input checked="" type="checkbox"/>
▼ 4. Обработка плохих признаков	
▶ Обработка признака Stop	
▶ Обработка признака Bad	
▶ Обработка признака Uncertain	

ФОРМИРОВАНИЕ ОТЧЕТОВ ПО ПЕРИОДИЧЕСКОЙ ВЫБОРКЕ ЗНАЧЕНИЙ

Частый способ формирования отчета является периодический. Для его задания необходимо для свойства Формирование значений элемента Архив данных выбрать Периодический. При таком способе строки таблицы значений будут сформированы по заданным интервалам времени. Нужный интервал времени задается с помощью параметра Интервал - можно задавать как константой, так и связать его с переменной дерева объекта типа Time.

Чтобы задать константу, например, формирование отчета раз в месяц, или раз в сутки, то нужно выделить Архив данных в словаре, а затем переключиться на панель свойств.

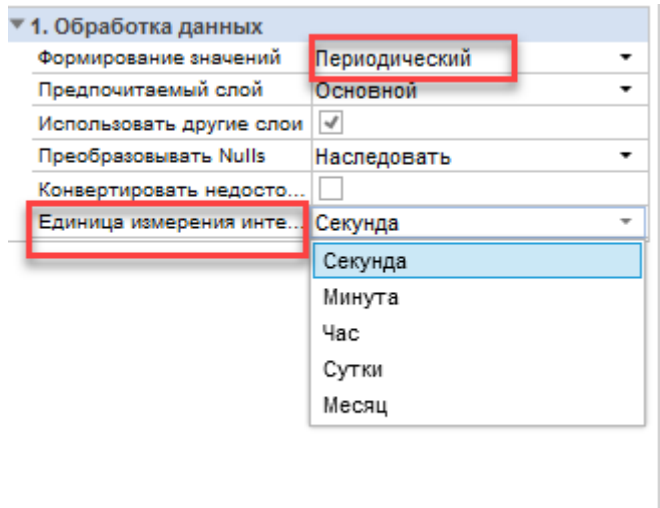
Интервал формирования новой строки отчета

Нужный интервал времени задается с помощью параметра Интервал - связать его с переменной дерева объекта типа Time. Либо использовать настройку Единица измерения интервала элемента словаря Архив данных.

Постоянный интервал

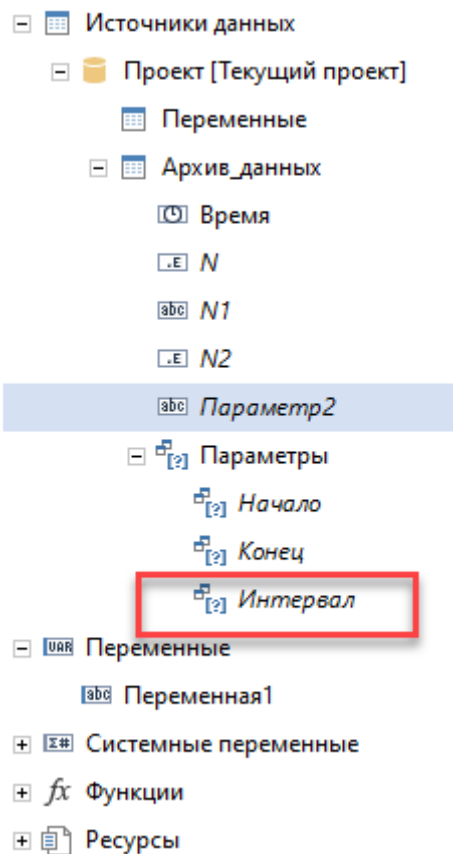
Чтобы задать константу, например, формирование отчета раз в месяц, или раз в сутки, то нужно выделить Архив данных в словаре, а затем переключиться на панель свойств.

В поле Единица измерения интервала можно выбрать из выпадающего списка нужное значение:

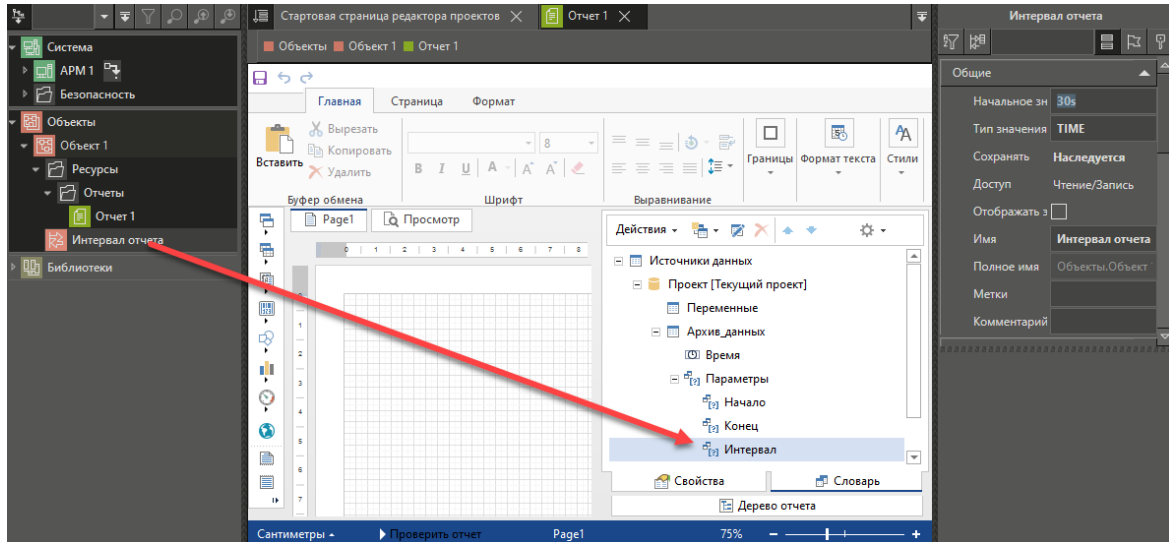


Изменяемый интервал

Если необходимо динамически изменять интервал формирования отчета, то нужно использовать параметр Интервал:



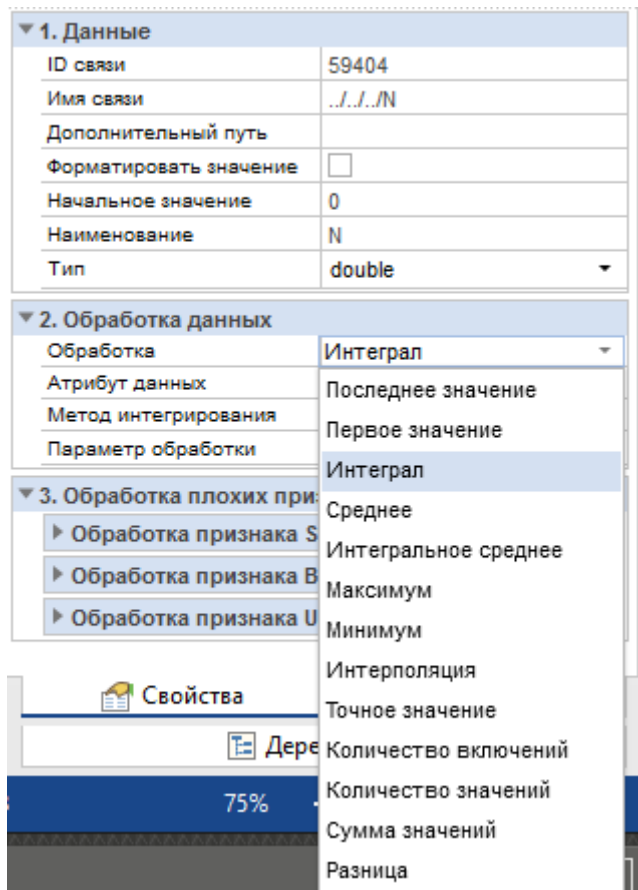
Параметр нужно связать с параметром проекта типа Time:



Важно! Если установлена связь с параметром Интервал, то настройка Единица измерения интервала не будет использоваться.

Обработка значений

При этом на каждый конкретный интервал времени может приходиться от нуля до нескольких значений переменной, поэтому необходимо задать способ вычисления значения. С этой целью нужно выбрать в дереве словаря колонку, перейти к ее свойству Обработка и присвоить ему одно из значений:



Обработка аналоговых переменных в периодическом отчете

- Последнее значение - будет выбрано последнее из имеющихся значений.
- Первое значение - будет выбрано первое из имеющихся значений.
- Интеграл - будет вычислена площадь фигуры ограниченной графиком значений (см. раздел Обработка Интеграл). Данный способ обработки применяется, например, для вычисления суммарного расхода по действующему значению.
- Среднее - будет вычислено среднее арифметическое значение.
- Интегральное среднее - будет вычислено среднее интегральное значение (интеграл разделенный на интервал времени).
- Максимум - будет выбрано максимальное значение на интервале.
- Минимум - будет выбрано минимальное значение на интервале.
- Количество значений - будет возвращено количество всех значений на интервале
- Сумма значений - будет вычислена сумма всех значений на интервале.
- Разница - будет вычислена разность между первым и последним значением на интервале. Данный способ обработки используется, например, если нужно вычислить значение энергии от счетчика с нарастающим итогом.

Обработка дискретных переменных в периодическом отчете

Для дискретных переменных помимо способов обработки Последнее значение, Минимум, Максимум используются способы обработки, как Количество включений.

Количество включений – определяет количество переходов из False в TRUE Ложь переменной на интервале времени.

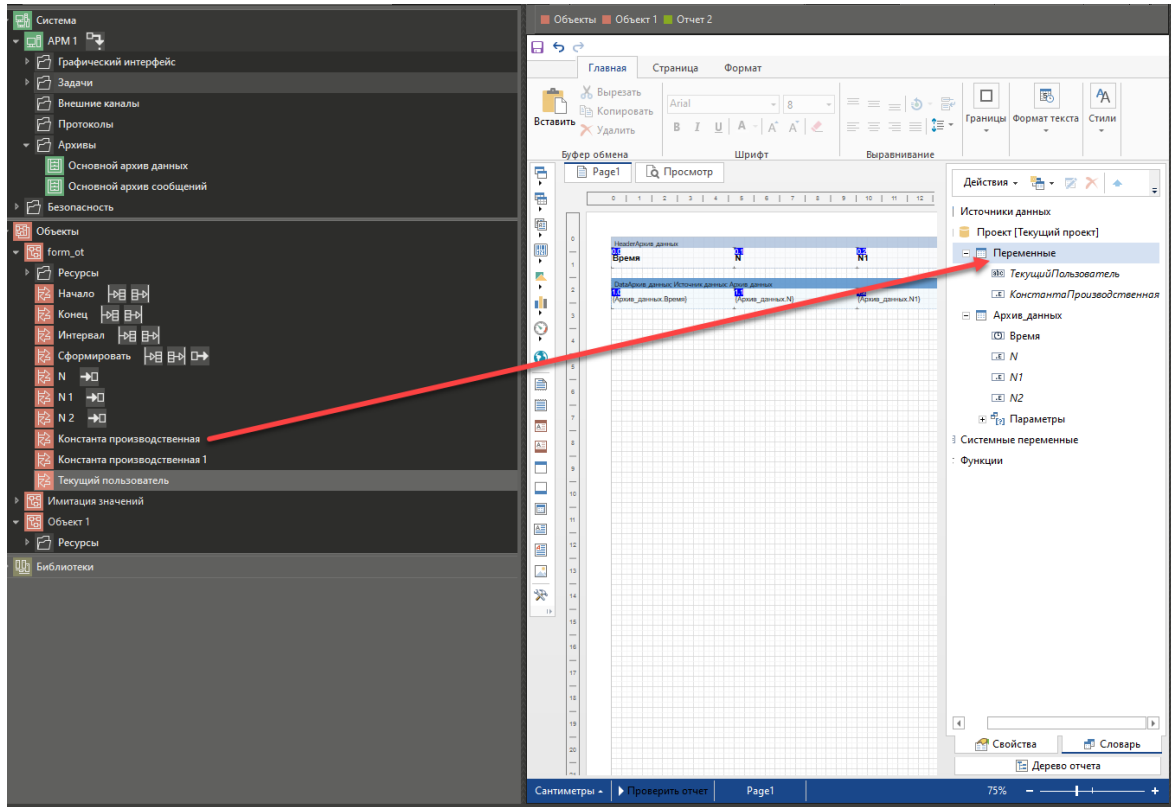
ИТОГОВЫЙ СПОСОБ ФОРМИРОВАНИЯ ОТЧЕТА

Итоговый способ формирования отчета является частным случаем периодического: это интервал от начала до конца формирования отчета. Для итогового отчета параметр Интервал не настраивается

Переменные источника данных Текущий проект

Добавление переменной/параметра в отчет

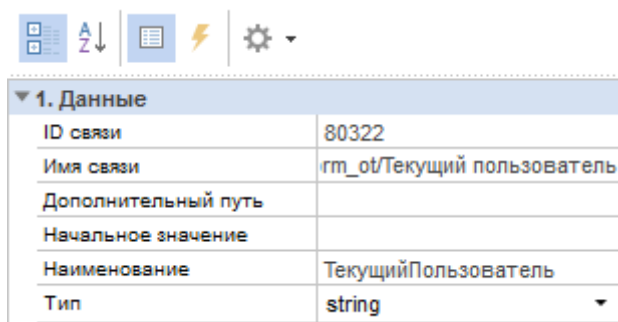
В словарь, в источник данных Проект [Текущий проект] в группу Переменные можно перетащить параметры MasterSCADA 4D:



Переменные можно добавлять как из дерева объектов, так и из дерева системы. Однако рекомендуется добавлять переменные расположенные в объекте, в котором находится отчет, и вложенных в него - это облегчает копирование объекта.

Свойства переменной в отчете

Чтобы перейти на панель свойств переменной необходимо выделить элемент в словаре, а затем переключиться на панель свойств отчета.



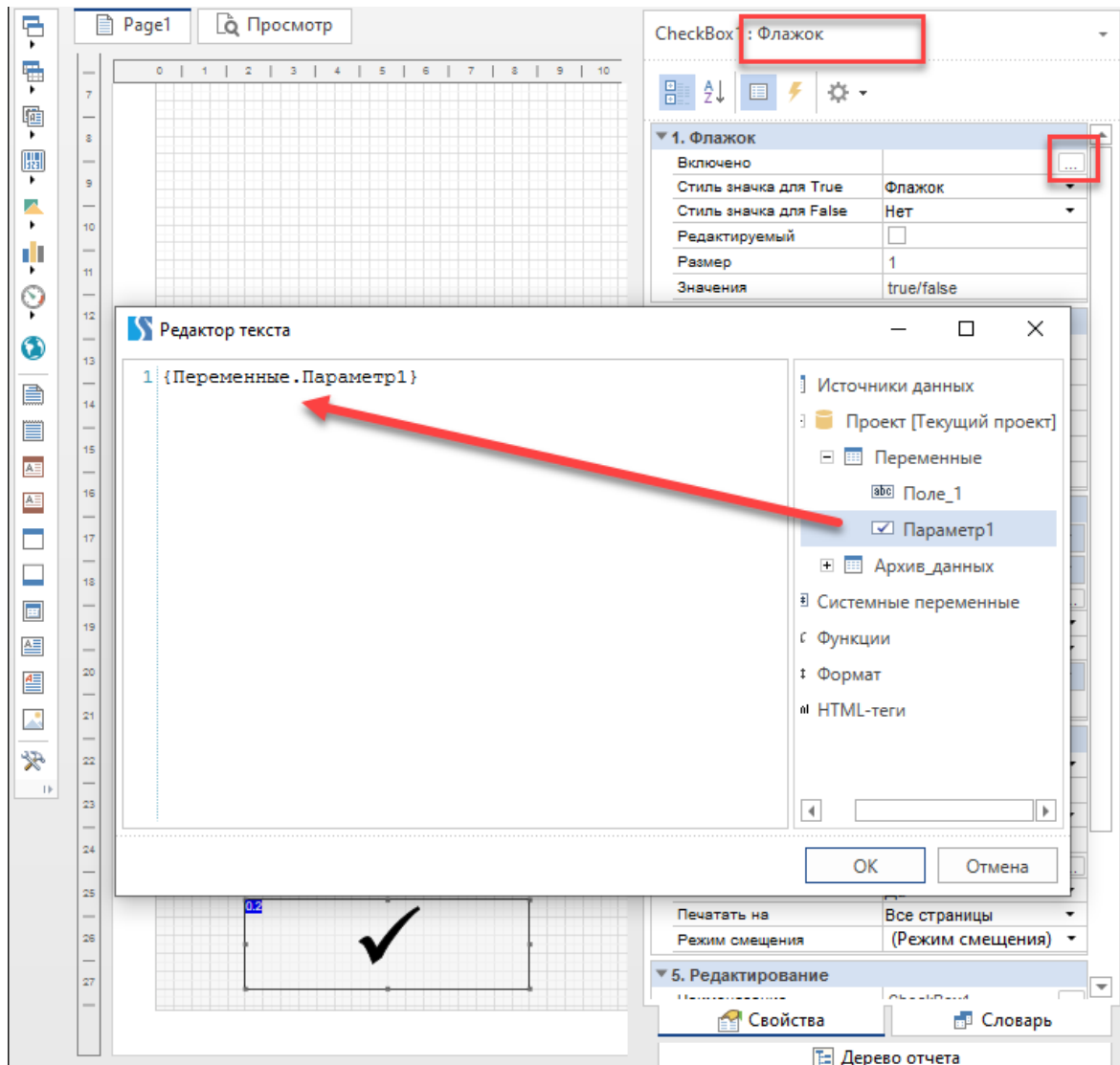
Свойство	Описание
ID связи	Указывается ID связи.
Имя связи	Указывается путь к параметру дерева проекта, состоящее из программных имен элементов проекта. Заполняется

	автоматически после того как параметр перетащили в словарь
Дополнительный путь	Если в отчет перетащили элемент сложного типа данных, например, поля структуры. То в данном поле будет отображаться путь к этому элементу внутри сложного типа данных. Заполняется автоматически после того как параметр перетащили в словарь
Начальное значение	Можно указать начальное значение, которое будет отображаться в отчете, пока нет связи со связанным параметром в режиме исполнения
Наименование	Имя переменной, которое отображается в словаре.
Тип	Тип данных переменной (наследуется от связанного параметра проекта)

Отображение переменных на странице отчета

Переменную отчета можно перетащить в ту секцию, в которой нужно ее отображать. Например, в Итог страницы. Если перетащить переменную на страницу отчета левой кнопкой мыши, то будет создан компонент Текст, и будет установлена связь между компонентом и переменной.

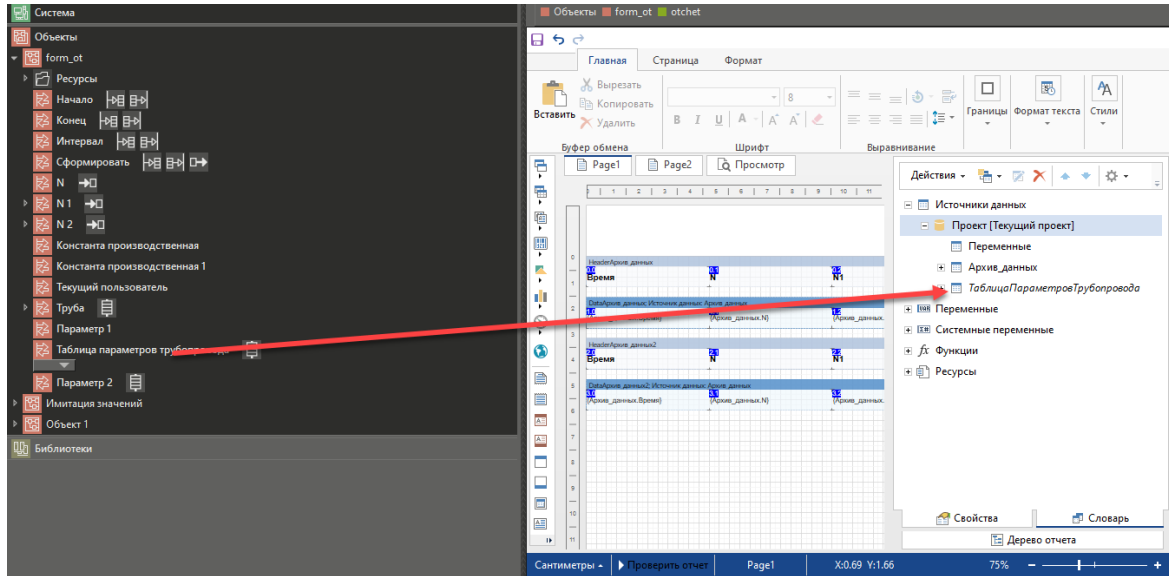
Если расположить сначала компонент, например, Флажок, то потом нужно будет установить связь вручную с переменной источника данных. Для этого в панели свойств нужно настроить свойство Включено:



В режиме исполнения будет отображаться значение переменной на момент формирования отчета.

9.3.1.1.3. МАССИВ СТРУКТУР В ОТЧЕТЕ

В отчет может быть выведен параметр типа массив структур. Если перетащить параметр данного типа в источник данных Проект [Текущий проект], то появится новая категория. Имя этой категории будет совпадать с именем параметра:



Новый элемент будет содержать столько дочерних элементов, сколько имеет полей структура. из которой состоит массив:

- ☐ **Таблица параметров трубопровода**
 - ☐ Поле_1
 - ☐ Поле_2

Далее, если перетащить эту категорию в отчет, то автоматически будет создана таблица. В режиме исполнения, при построении отчета она будет содержать столько строк, сколько элементов будет в массиве.

9.3.1.2. ПРОЧИЕ ИСТОЧНИКИ

Основным источником данных является источник Текущий проект. Однако редактор отчетов также позволяет работать и с другими различными источниками данных - различными СУБД, OLE DB, ODBC, текстовые файлы различных форматов (csv, XML, JSON). Указанные источники данных предназначены для решения специальных задач и применяются редко.

Рассмотрим некоторые из них:

Источник данных СУБД

Пример работы с БД MS SQL

Создание и редактирование колонки данных

Рассчитываемая колонка данных

Данные на основе других источников данных

Данные из кросс-таблиц

Пользовательские источники данных

Получить только используемые данные

9.3.1.2.1. ИСТОЧНИК ДАННЫХ СУБД

Для построения отчета можно использовать данные полученные из сторонних баз данных. Поддерживаются базы данных MS SQL и PostGre.

Для этого нужно выполнить следующие действия.

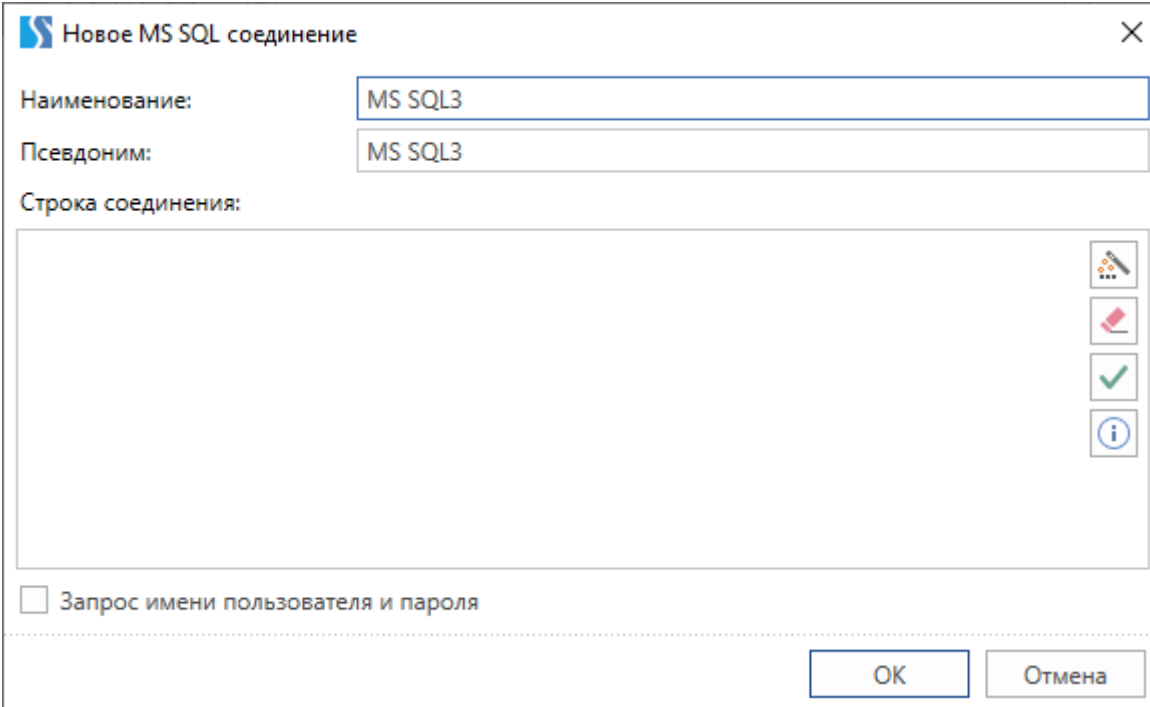
Добавление нового источника данных

Новый источник данных можно добавить разными способами. Рассмотрим самый растространенный

Сначала нужно выбрать в панели Словарь в меню Новый элемент пункт Новый источник данных.

В появившемся окне выбрать базу данных:

Далее появится окно, в котором нужно указать необходимые параметры для подключения:



Новое MS SQL соединение

Наименование: MS SQL3

Псевдоним: MS SQL3



Строка соединения:

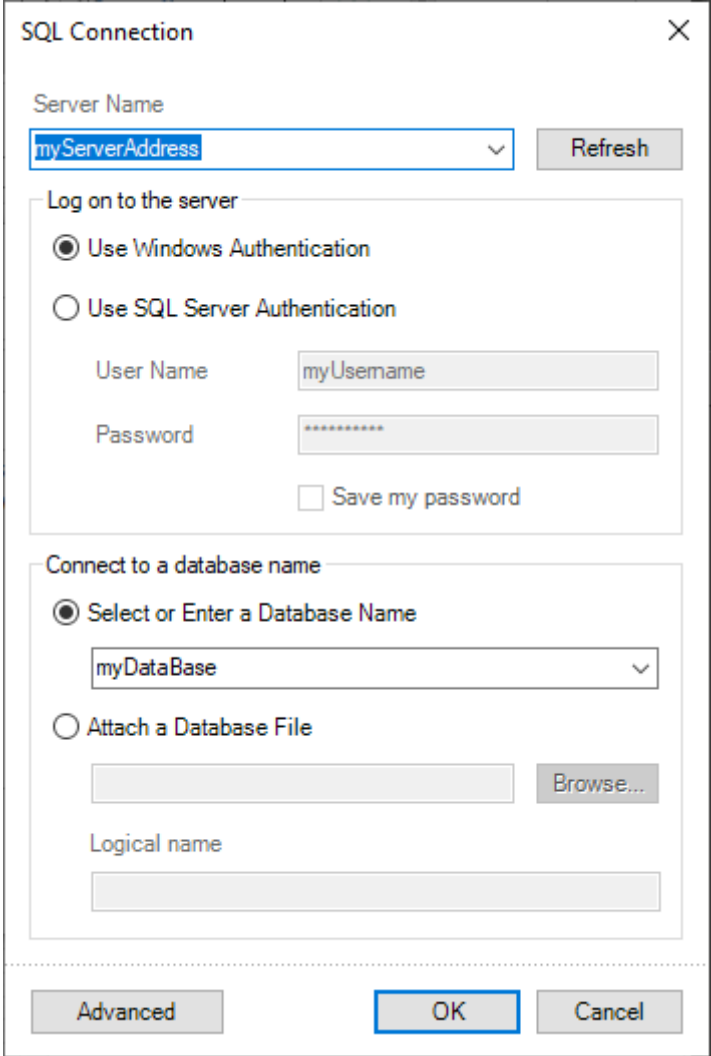
Запрос имени пользователя и пароля

ОК Отмена

Самое главное указать параметры подключения в поле строка соединения.

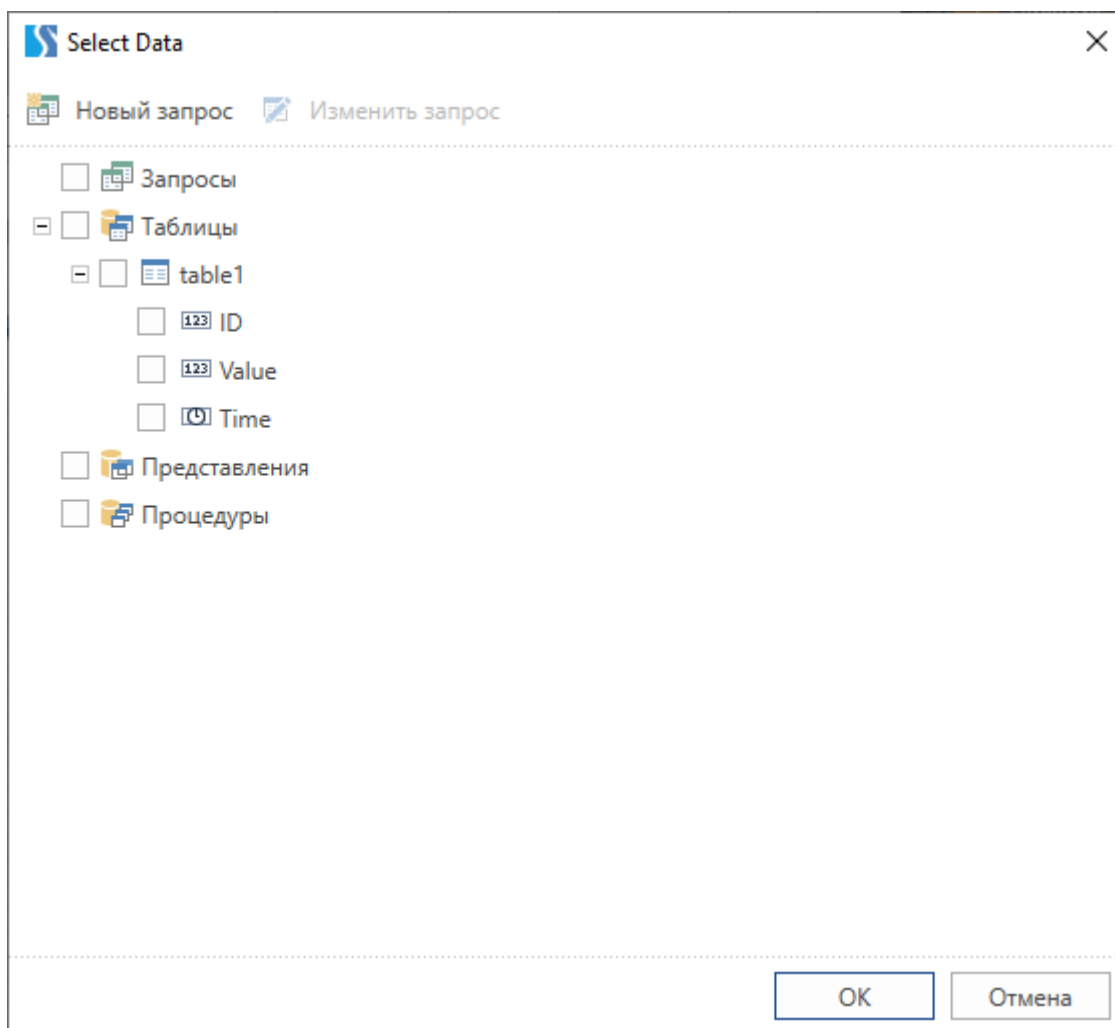
База	Шаблон строки соединения
MS SQL	Integrated Security=False; Data Source=myServerAddress;Initial Catalog=myDataBase; User ID=myUsername; Password=myPassword;
Postgre SQL	Server=myServerAddress; Port=5432; Database=myDataBase;User Id=myUsername; Password=myPassword

Чтобы вставить шаблон нужной строки подключения нужно нажать на кнопку  Затем шаблон можно отредактировать - указать нужные значения. Либо можно нажать на кнопку  и в диалоговом окне указать все параметры:



После нажатия на кнопку ОК параметры подключения добавятся в поле Строка соединения

Если все настройки были заданы верно, то произойдет подключение к базе данных, после нажатия на кнопку ОК. И откроется окно, в котором нужно указать, какие значения из базы нужно получить:



Если в отчете необходимо получить все значения из базы, то нужно выбрать в группе Таблицы нужные таблицы и столбцы (колонки) которые нужно будет отобразить в отчете.

Если необходимо сделать выборку из базы, то нужно сформировать Запрос. Для этого нужно нажать кнопку Новый запрос.

Запросы представляют собой текст скриптового вида, который служит для извлечения данных из таблиц и предоставления их в генератор отчетов. Особенность запросов состоит в том, что они получают данные из таблиц базы данных и создают на их основе временную таблицу. Данные во временной таблице будут отфильтрованы, сгруппированы, отсортированы и упорядочены, согласно параметрам запроса. Затем, временная таблица передается в генератор отчетов. Применение запросов предоставляет возможность избежать дублирования данных в таблицах и обеспечивает максимальную гибкость при поиске и отображении данных в базе данных. В основном запросы используются для выборки данных из базы данных и передачи их в генератор отчетов. Если тип источника данных поддерживает SQL запросы, то в окне Новый источник данных (New Data Source)

будет отображено поле Текст запроса (Query Text), в котором указывается запрос. На рисунке снизу представлено окно Новый источник данных (New Data Source), где в поле Текст запроса (Query Text) создан запрос на выборку:

Изменение источника данных

Наименование в источнике: MS SQL

Наименование: ИсточникДанных1

Псевдоним: ИсточникДанных1

! SQL Запустить построитель запросов

Текст запроса

```
select * from table1 where Value < @Max and Value>@Min
```

Тип: Запрос

Время ожидания запроса: 30 sec

Получить все колонки

Колонки

- ID
- Value
- Time

Параметры

- Min
- Max

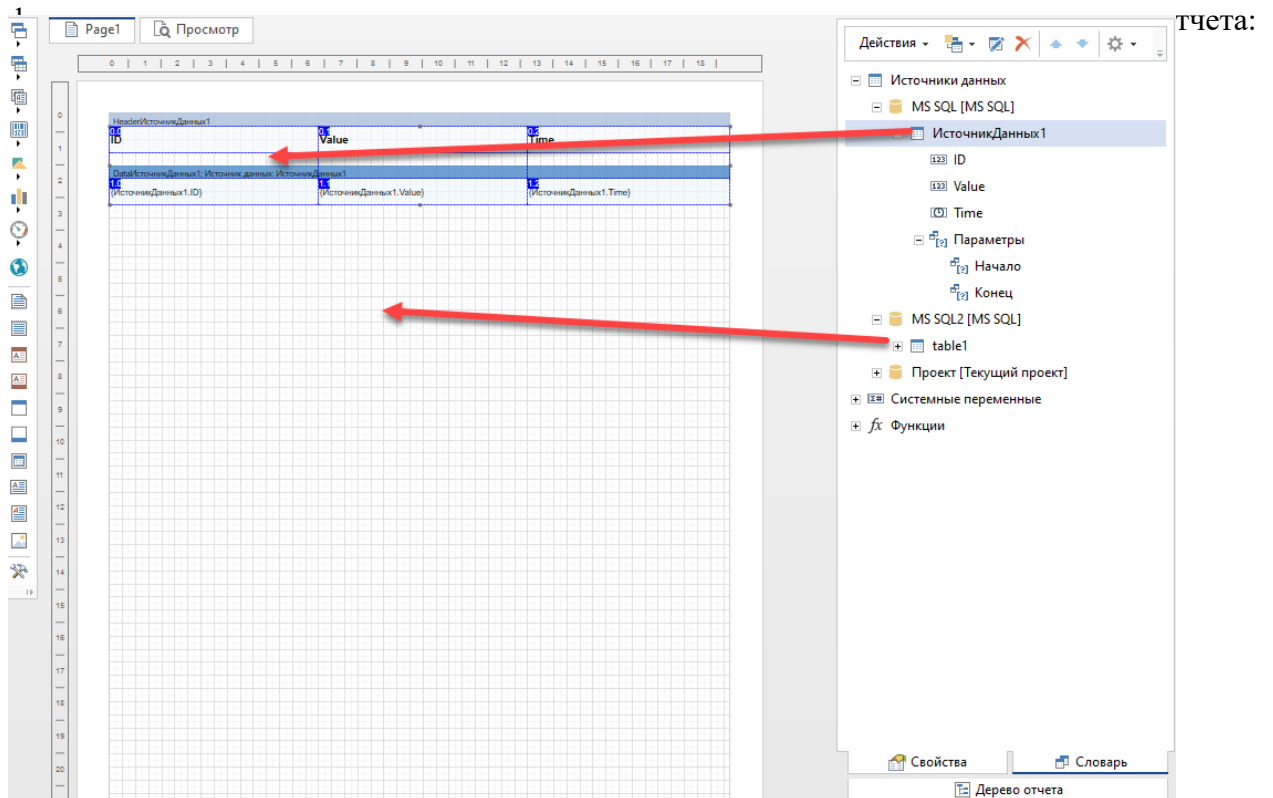
1. Данные	
Наименование	Max
Выражение	
Размер	0
Тип	Real

Сохранить копию

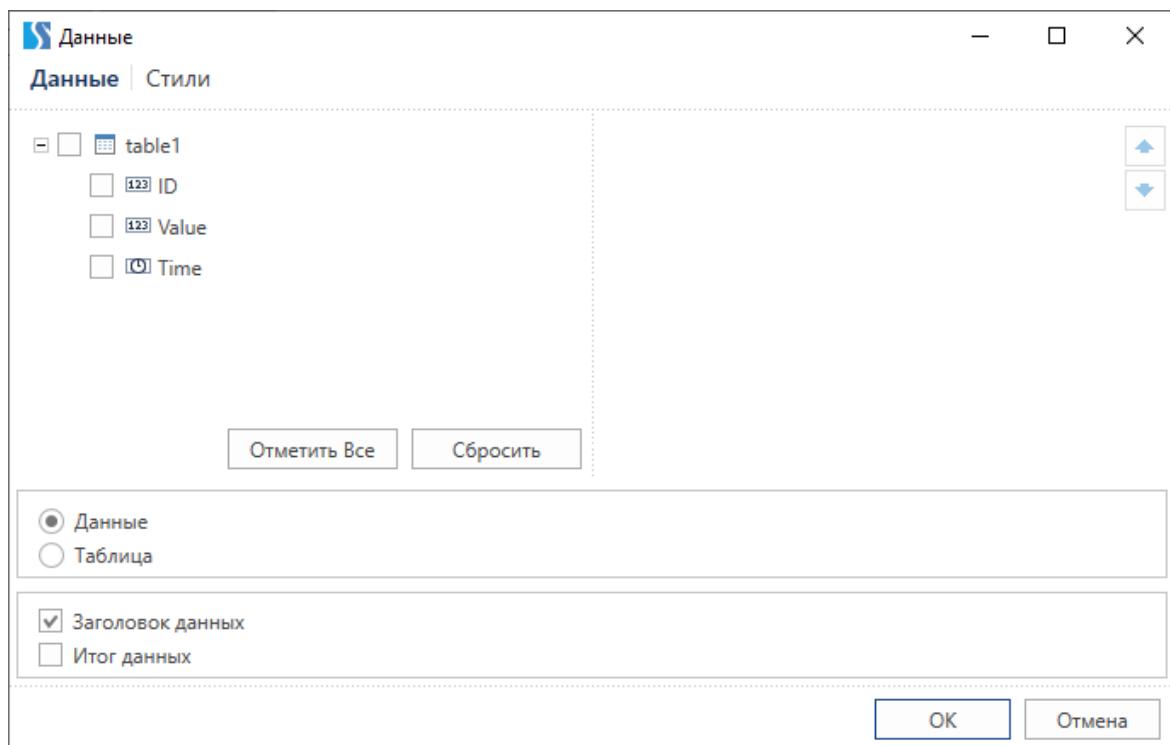
OK Отмена

- Пример работы с БД MS SQL
- Запросы

Далее после нажатия на кнопку ОК в словарь добавится новый элемент. Если перетащить его на страницу, то появится бэнд, который будет заполнен в режиме исполнения при



После отпускания мыши, откроется окно, в котором можно выбрать какие колонки нужно добавлять в отчет, а также какой бэнд использовать:

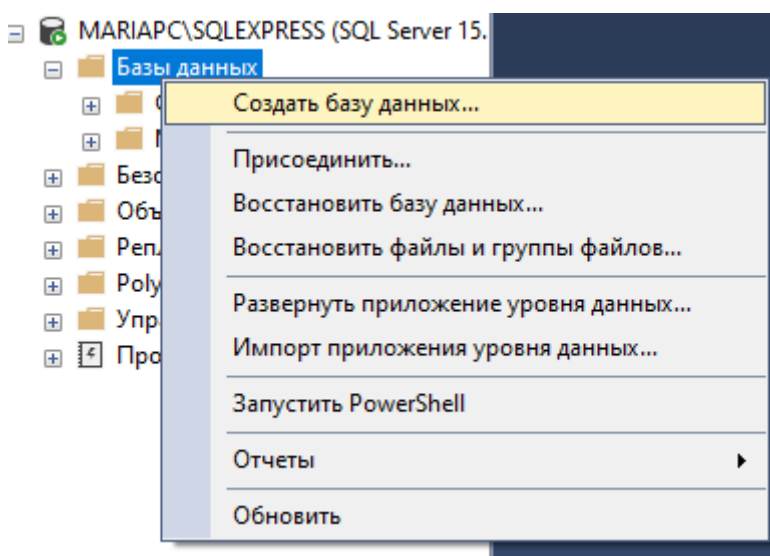


9.3.1.2.1.1. ПРИМЕР РАБОТЫ С БД MS SQL

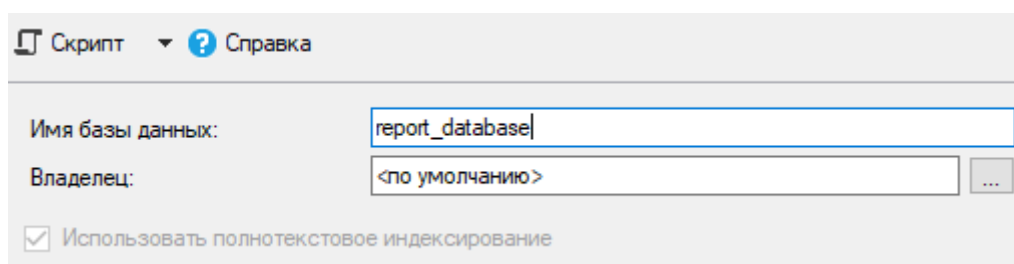
Создадим пример, в котором, используя редактор отчетов, мы получаем данные из SQL-сервера в зависимости от значения одного из столбцов. Отчет должен содержать таблицу следующего вида: ID (уникальный номер записи), Время, Величина значения. Все данные мы будем получать из SQL-сервера, переменные Min и Конец, для задания диапазона выборки значения, мы будем получать из дерева объектов MasterSCADA 4D. Для таких отчетов необходимо использовать источник данных Данные из SQL-соединения. В качестве SQL сервера будем использовать MS SQL. Сначала создадим базу данных и наполним ее данными.

Создание базы данных

Используя администратор баз Microsoft SQL Server Management Studio, создадим новую базу:



Укажем имя базы report_database:



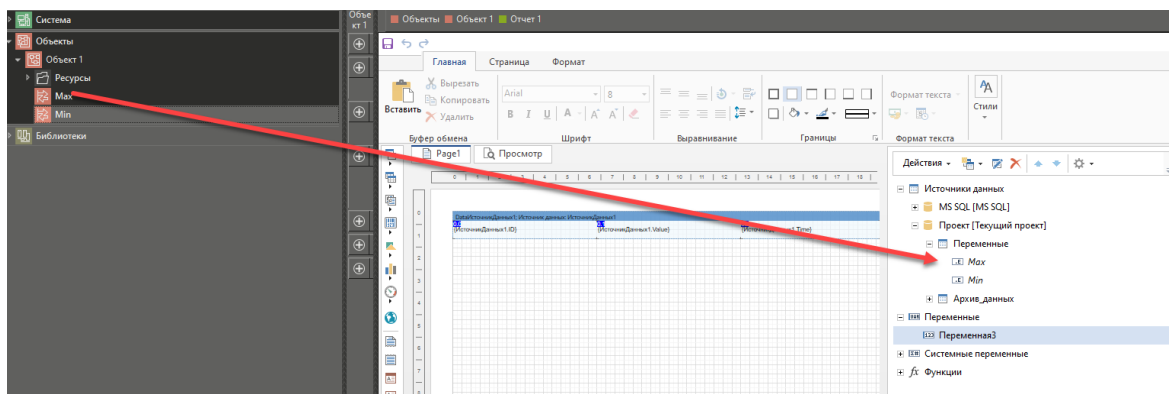
Добавим новую таблицу. Назовем ее table. Сделаем три поля: уникальный номер поля – ID (тип bigint, поле не может быть null), значение переменное – Value (тип int), и время переменной – Time (тип datetime). И зададим следующие значения:

	ID	Value	Time
	1	12	2019-09-20 13:55:00.000
	2	14	2019-09-20 13:59:00.000
	3	34	2019-09-20 14:05:00.000
	4	45	2019-09-20 14:15:00.000
	5	65	2019-09-20 14:20:00.000
	6	80	2019-09-20 14:25:00.000
	7	90	2019-09-20 14:30:00.000
	8	98	2019-09-20 14:35:00.000
	9	100	2019-09-20 14:40:00.000
	10	97	2019-09-20 14:45:00.000
✎	11	80	2019-09-20 14:50:00.000
*	NULL	NULL	NULL

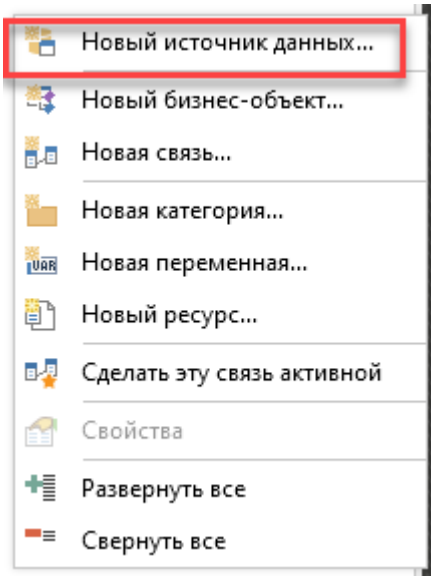
Создание отчета в MasterSCADA 4D

В проект MasterSCADA 4D в объект добавим два параметра Max и Min. Типа параметров Real. Можно указать Начальные значения параметров, ориентируясь на значения указанные в базе.

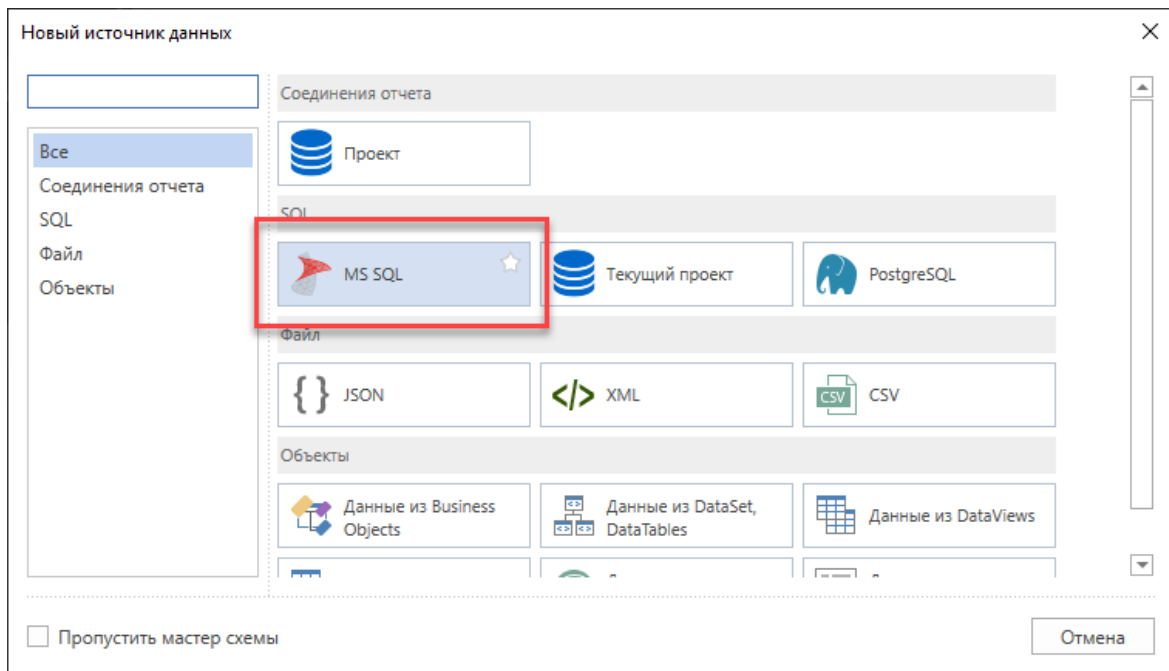
Далее в объект добавляем Отчет. В словарь перетащим переменные из дерева, в категорию Переменные:



Вспользуемся контекстным меню Словаря пункт Добавить новый источник данных:



Далее выберем:



В появившемся окне нужно указать имя нового источника данных и параметры подключения. Чтобы вставить шаблон нужно строки подключения нужно нажать на кнопку



Затем шаблон можно отредактировать - указать нужные значения. Либо можно нажать

на кнопку



и в диалоговом окне указать все параметры:

SQL Connection

Server Name
myServerAddress Refresh

Log on to the server

Use Windows Authentication

Use SQL Server Authentication

User Name myUsername

Password *****

Save my password

Connect to a database name

Select or Enter a Database Name

myDataBase

Attach a Database File

Browse...

Logical name

Advanced OK Cancel

После нажатия на кнопку ОК параметры подключения добавятся в окно:

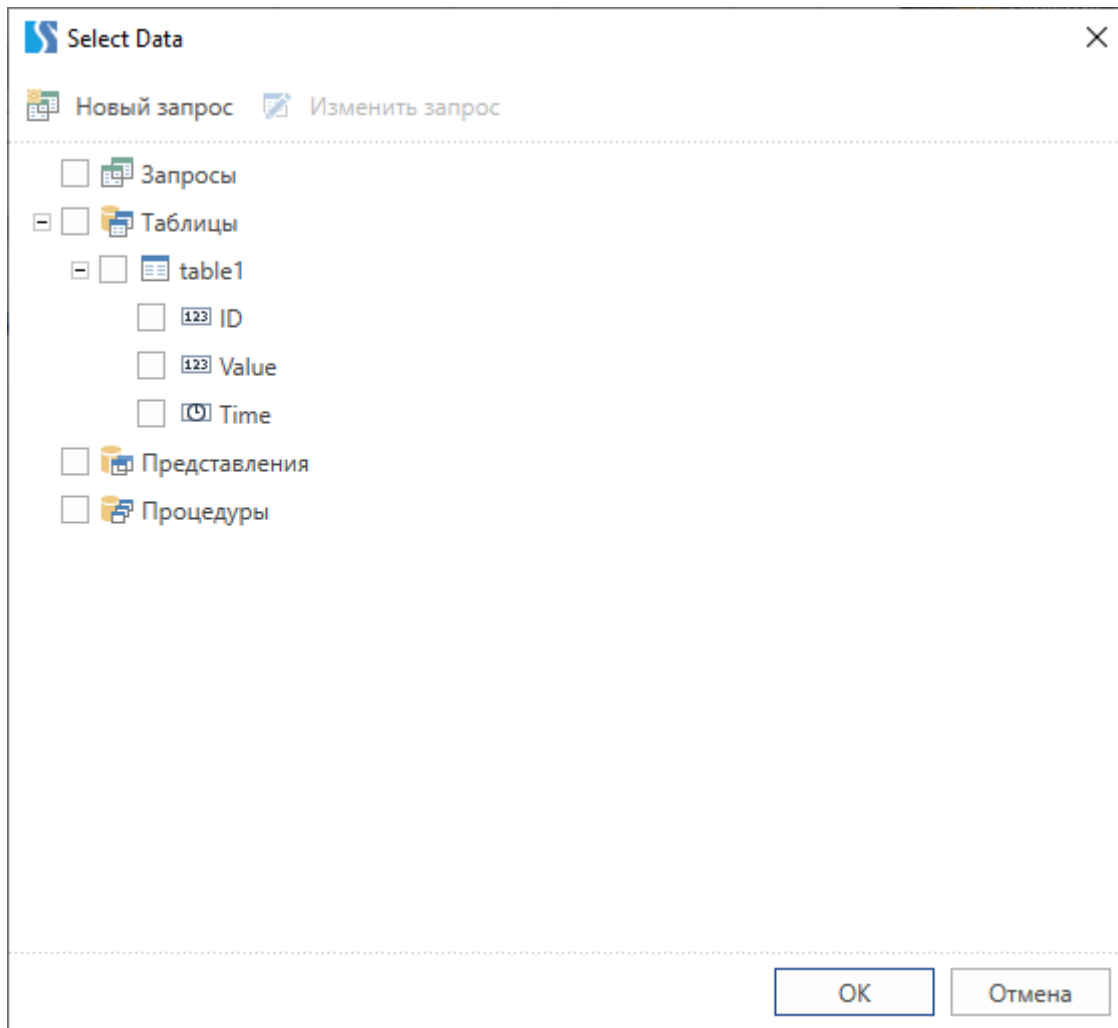
Строка соединения:

Data Source=MARIAPC\SQLEXPRESS;Initial Catalog=report_database;Integrated Security=True;User ID=;Password=

Запрос имени пользователя и пароля

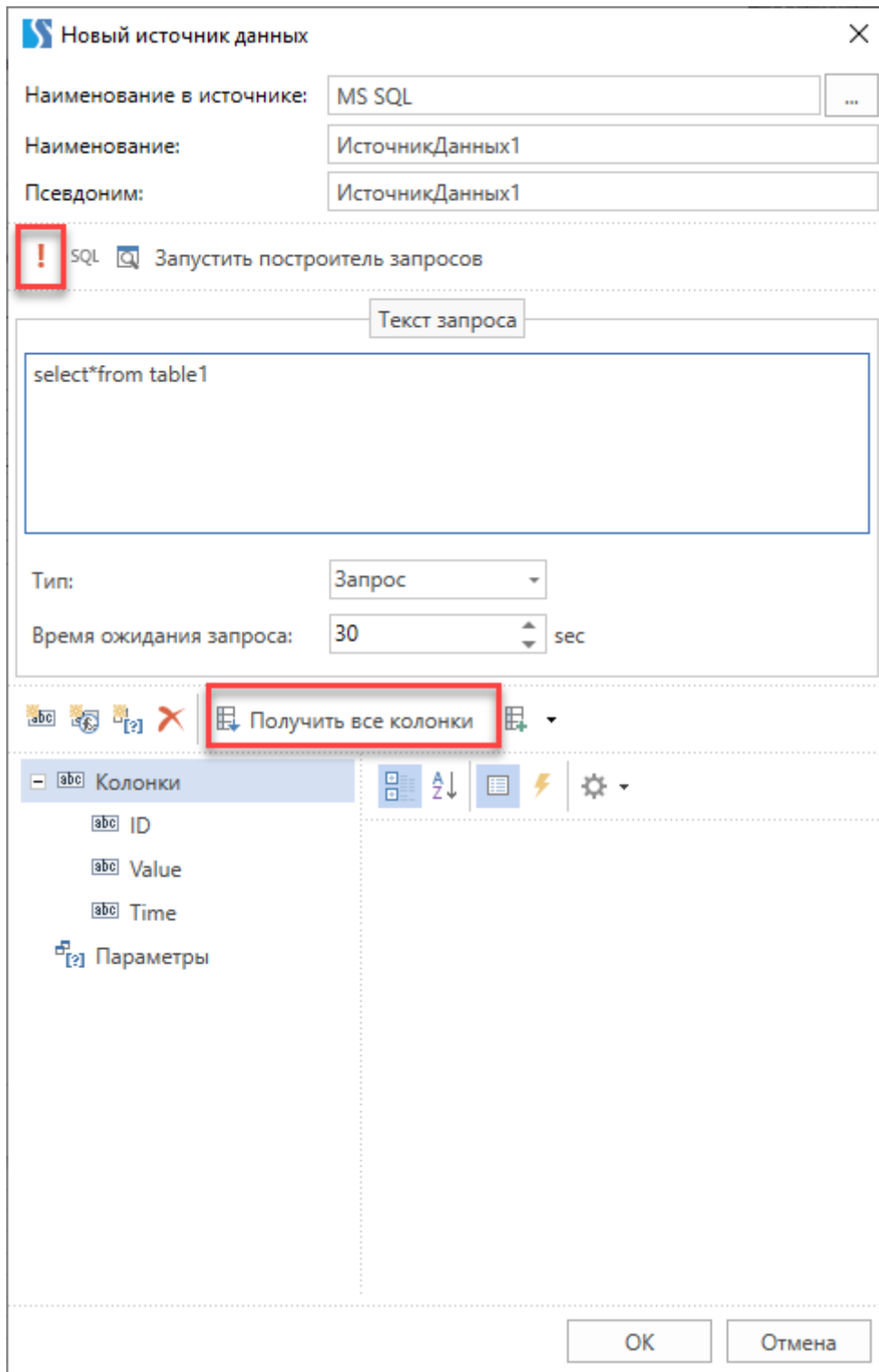
OK Отмена

Если все настройки были заданы верно, то произойдет подключение к базе данных, после нажатия на кнопку ОК. И откроется окно, в котором нужно указать, какие значения из базы нужно получить:




Чтобы из базы данных получить выборку значений нужно создать запрос. Для этого требуется нажать кнопку Новый запрос.

Откроется окно, где нужно указать текст запроса, проверить его, а затем получить все колонки:



необходимо ограничить выборку при помощи двух параметров времени – начало и конец.

Добавим эти параметры. Для этого в окне формирования запроса нажмем на кнопку .

В группе параметры появится новый элемент. Нужно указать его имя и тип данных DateTime. Таким образом нужно создать два параметра Начало и Конец.

Изменение источника данных
✕

Наименование в источнике: ...

Наименование:

Псевдоним:

SQL Запустить построитель запросов

Текст запроса

```
select * from table1
```

Тип:

Время ожидания запроса: sec

| Получить все колонки

Колонки

- ID
- Value
- Time

Параметры

- Min
- Max

▼ 1. Данные

Наименование	Max
Выражение	
Размер	0
Тип	Real ▼

После этого необходимо уточнить Текст запроса. Необходимо построить запрос выборки при помощи этих двух параметров. Для того чтобы использовать в запросе параметр, перед его именем нужно поставить символ @. Получим: `select * from table1 where Value < @Max and Value > @Min:`

Изменение источника данных [X]

Наименование в источнике: MS SQL [...]

Наименование: ИсточникДанных1

Псевдоним: ИсточникДанных1

! SQL [?] Запустить построитель запросов

Текст запроса

```
select * from table1 where Value < @Max and Value>@Min
```

Тип: Запрос

Время ожидания запроса: 30 sec

[abc] [?] [X] | [?] Получить все колонки [?]

[abc] Колонки

- [abc] ID
- [abc] Value
- [abc] Time

[?] Параметры

- [?] Min
- [?] Max

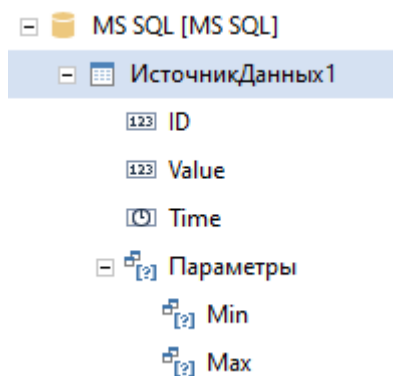
[?] [?] [?] [?] [?]

1. Данные	
Наименование	Max
Выражение	
Размер	0
Тип	Real

Сохранить копию [OK] [Отмена]

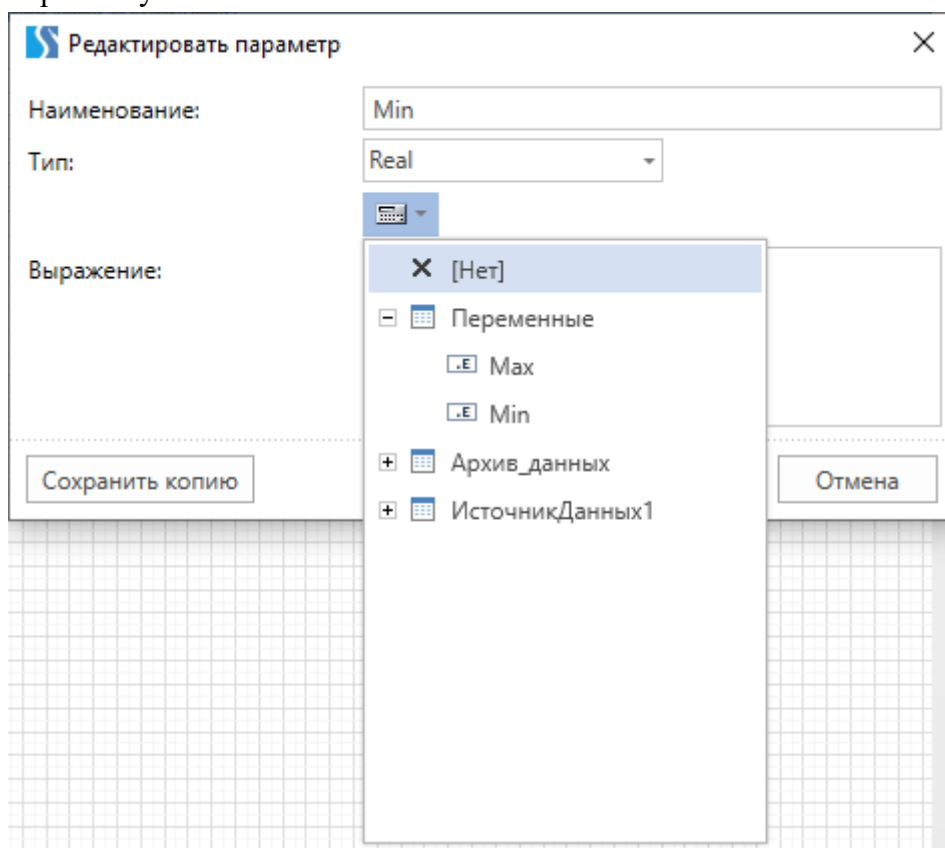
Нужно нажать кнопку ОК.

В Словаре получим:



Далее необходимо настроить связь между параметрами источника данных и переменными, которые были получены в источнике данных Текущий проект.

Нужно дважды кликнуть по параметру. Появится окно, в окне выбрать нужную переменную:



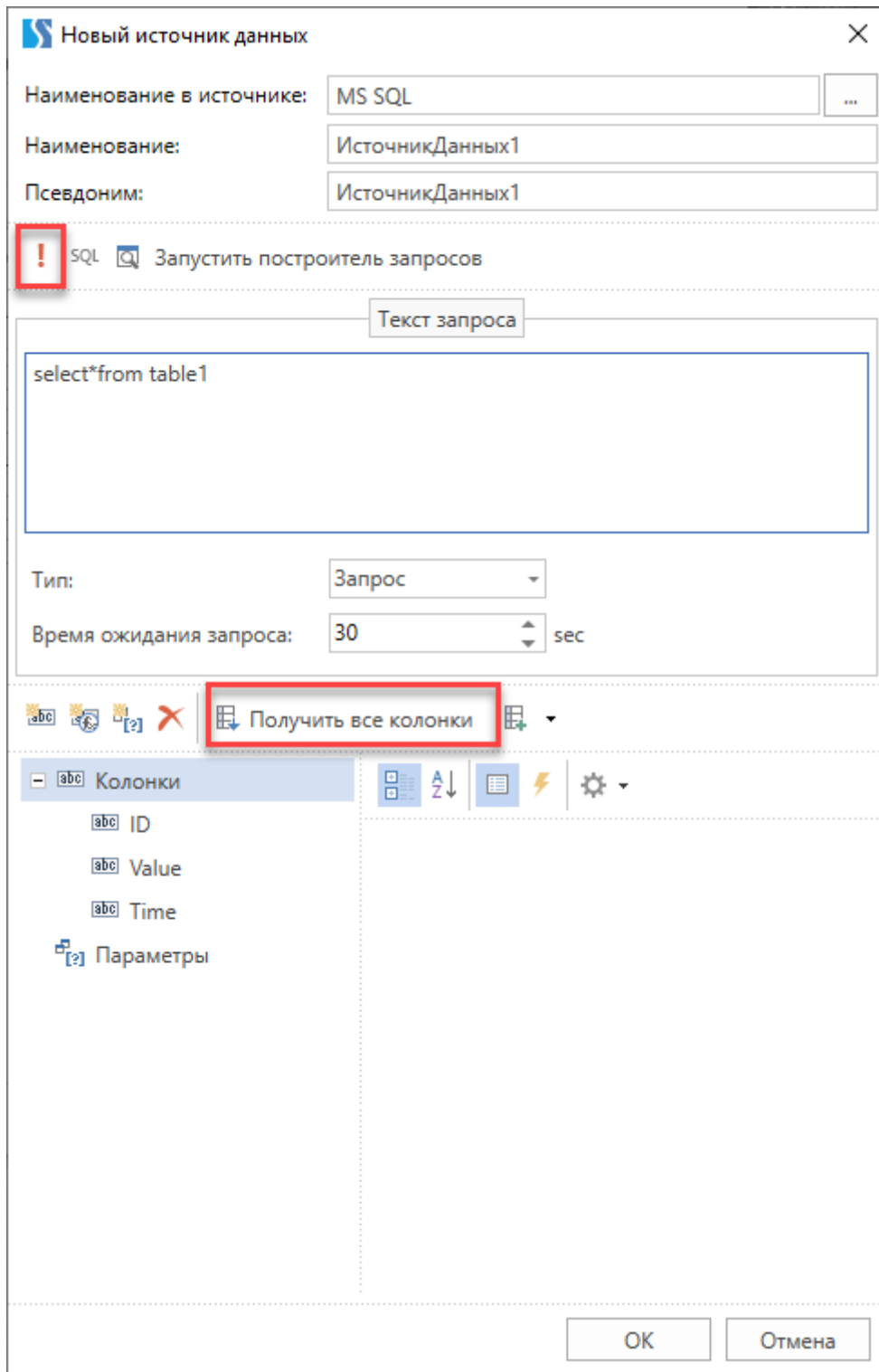
Элемент Источник данных можно перетащить на страницу отчета левой кнопкой мыши. Там появится бэнд Данные.

В исполнительной системе сформируется отчет на основании данных, которые хранятся в указанной базе. и с учетом заданных параметров

Важно! Следует обеспечить исполнительной системе доступ к базе данных.

















9.3.1.2.1.2. ЗАПРОСЫ

Запросы представляют собой текст скриптового вида, который служит для извлечения данных из таблиц и предоставления их в генератор отчетов. Особенность запросов состоит в том, что они получают данные из таблиц базы данных и создают на их основе временную таблицу. Данные во временной таблице будут отфильтрованы, сгруппированы, отсортированы и упорядочены, согласно параметрам запроса. Затем, временная таблица передается в генератор отчетов. Применение запросов предоставляет возможность избежать дублирования данных в таблицах и обеспечивает максимальную гибкость при поиске и отображении данных в базе данных. В основном запросы используются для выборки данных из базы данных и передачи их в генератор отчетов. Не все типы источников данных поддерживают SQL запросы. Если тип источника данных поддерживает SQL запросы, то в окне Новый источник данных (New Data Source) будет отображено поле Текст запроса (Query Text), в котором указывается запрос. На рисунке снизу представлено окно Новый источник данных (New Data Source), где в поле Текст запроса (Query Text) создан запрос на выборку:



Элементы окна

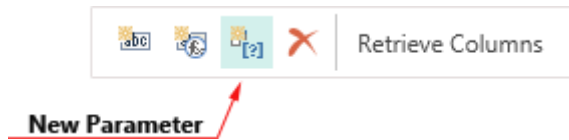
Название	Назначение
Наименование в источнике (Name in Source)	Определяется в каком источнике будет добавлен новый источник данных. В данном

Название	Назначение
	поле можно ввести имя или можно нажать кнопку  , для вызова списка имен.
Наименование (Name)	Имя источника данных
Псевдоним (Alias)	Псевдоним источника
 SQL  Запустить построитель запросов (Команды управления запросами)	На данной панели расположены основные элементы управления текстовыми запросами. Для того, чтобы запустить запрос на выполнение следует нажать кнопку Выполнить (Run).
Текст запроса (Query Text)	В данном поле указывается текст запроса
Тип	Указывается тип - запрос или хранимая процедура
     Получить все колонки  ▾ (Команды управления данными)	На этой панели представлены такие команды как создание новой колонки, новой рассчитываемой колонки, нового параметра. Также присутствует команда удаления выделенной колонки или параметра. Помимо всего прочего, на данной панели располагается команда Получить колонки данных (Retrieve Columns).
 Колонки  ID  Value  Time  Параметры  Начало  Конец	На данной панели отображаются колонки данных источника, а также параметры. Свойства выделенной колонки или параметра располагаются на панели свойств.

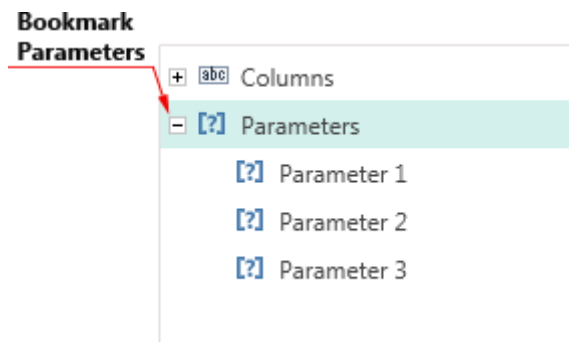
ПАРАМЕТРЫ

При создании запроса есть возможность использовать объект Параметр (Parameter). Данный объект предназначен для передачи в запрос дополнительных условий для выборки данных. К примеру, если требуется, чтобы запрос использовал какое-либо значение

вводимое пользователем при каждом выполнении запроса, то можно создать запрос с использованием параметров. Объект Параметр (Parameter) может быть использован только совместно с SQL источниками данных. Такие источники данных обычно имеют поле Текст запроса (Text Query). Для того, чтобы вставить параметр в запрос следует нажать кнопку Новый параметр (New Parameter):



После нажатия данной кнопки будет создан новый параметр, который будет отображен в группе Параметры (Parameters) на панели Колонки (Columns):



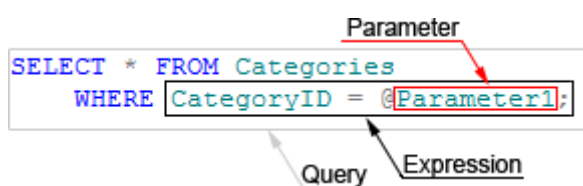
Каждый параметр имеет свойства, с помощью которых можно изменить его настройки. На рисунке снизу представлена свойства параметра:

▼ 1. Данные	
Наименование	Начало
Выражение	
Размер	0
Тип	DateTime ▼

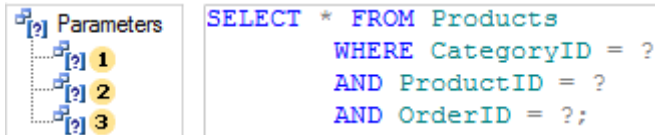
Название	Назначение
Наименование	Имя параметра. В случае именованного параметра
Выражение	Для каждого параметра можно указать значение, которое используется для заполнения параметра. Значение может быть выражением, const, выражение, ссылка на элемент словаря

Название	Назначение
	данных, к примеру переменная и т.д. К примеру, x+y или {variable}
Размер	Предоставляет возможность изменить размерность типа используемого в параметре. Следует помнить, что в базе данных каждый тип имеет свою размерность. Поэтому при использовании параметра в запросе, необходимо указывать корректную размерность типа. Для некоторых адаптеров баз данных размерность можно не указывать, но обычно если размерность не указана или указана неверно, то запросы с использованием таких параметров будут выполнены некорректно.
Тип	Значения данного свойства находятся в выпадающем списке, и представляет собой перечисление типов используемых в параметрах для конкретной базы данных. Стоит учитывать, что список типов отличается в зависимости от базы данных.

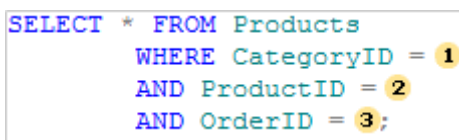
Также параметр необходимо указать и в самом запросе. Ниже представлен пример схематичного расположения параметров в запросе:



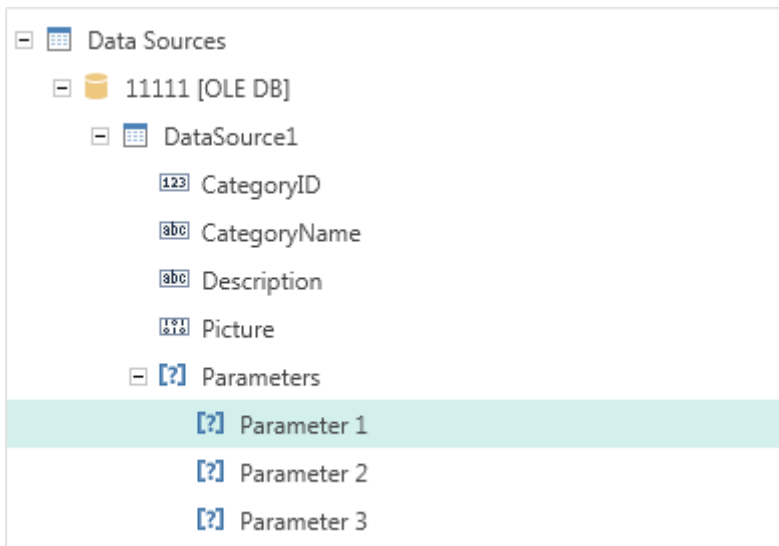
Как правило, для указания параметра в запросе используется символ @. Символ @ используется с именованными параметрами, т.е. после символа @ следует имя параметра. Но в некоторых базах данных (к примеру в OleDb), символ @ может не восприниматься адаптером базы данных и запросы с параметрами работать не будут. В данном случае, можно использовать неименованные параметры. Для указания в запросе неименованных параметров используется символ ?. После символа ? не указывается имя параметра. В этом случае, важен порядок параметров в группе Параметры (Parameters). По мере указания символов ? в запросе, параметры будут браться последовательно из группы Параметры (Parameters) в направлении "сверху-вниз". Рассмотрим на примере. Допустим есть три параметра, которые указываются в запросе:



Поскольку в данном случае используются неименованные параметры (обозначенные символом ?), то при выполнении запроса параметры будут браться из группы Параметры (Parameters) последовательно "сверху-вниз". На рисунке снизу схематично представлено сопоставление параметров из группы Параметры (Parameters) к параметрам в запросе:

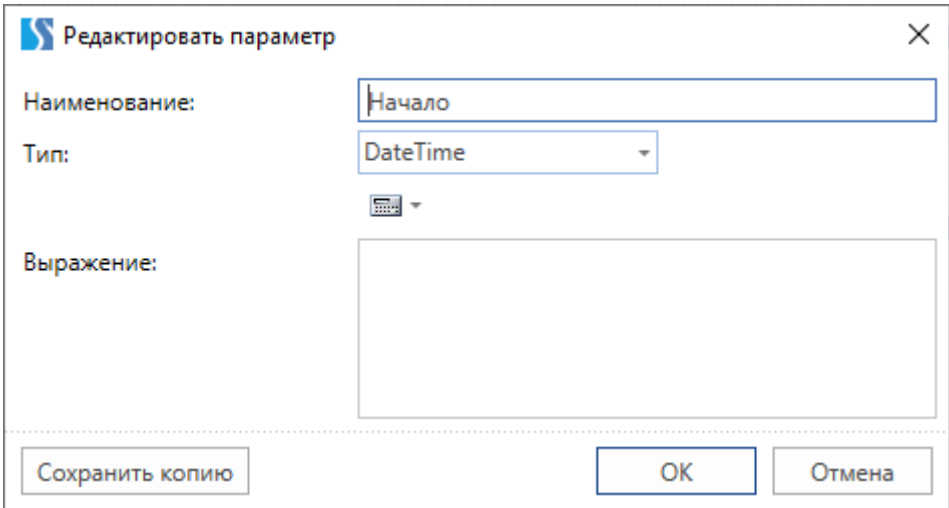


При этом, используемые параметры в данном примере, могут иметь имена, но при использовании символа ? они не играют роли. После того, как запрос с параметрами создан и выполнен, параметры также будут отображаться в Словаре (Dictionary), в созданном источнике данных, в группе Параметры (Parameters). На рисунке снизу представлен пример панели Словаря (Dictionary) и расположения в нем параметров:



Для того, чтобы отредактировать параметр отдельно от источника данных, следует выделить Параметр (Parameter) в словаре данных и дважды кликнуть по нему. Тогда будет отображено окно Редактирование параметра (Edit Parameter), в котором можно изменить

настройки выделенного параметра. На рисунке снизу представлен пример окна Редактирование параметра (Edit Parameter):



Скриншот диалогового окна «Редактировать параметр» (Edit Parameter). В окне отображены следующие элементы:

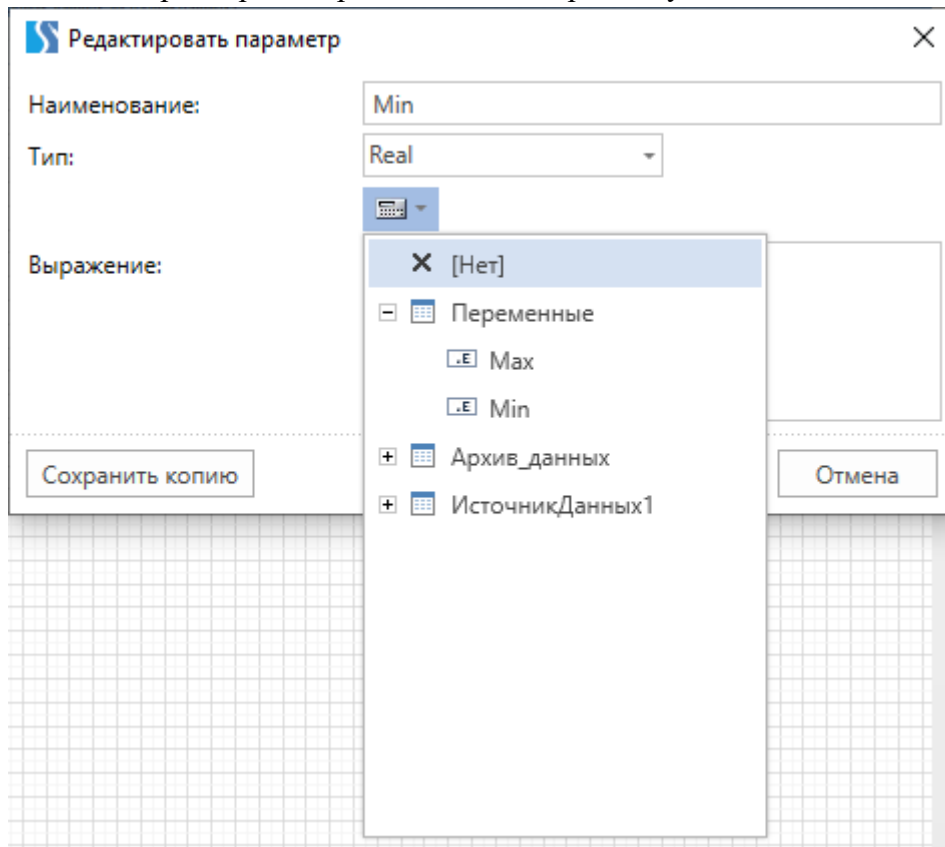
- Наименование:
- Тип:
- Выражение:

В нижней части окна расположены кнопки: «Сохранить копию», «ОК» и «Отмена».

Использование переменной как параметра

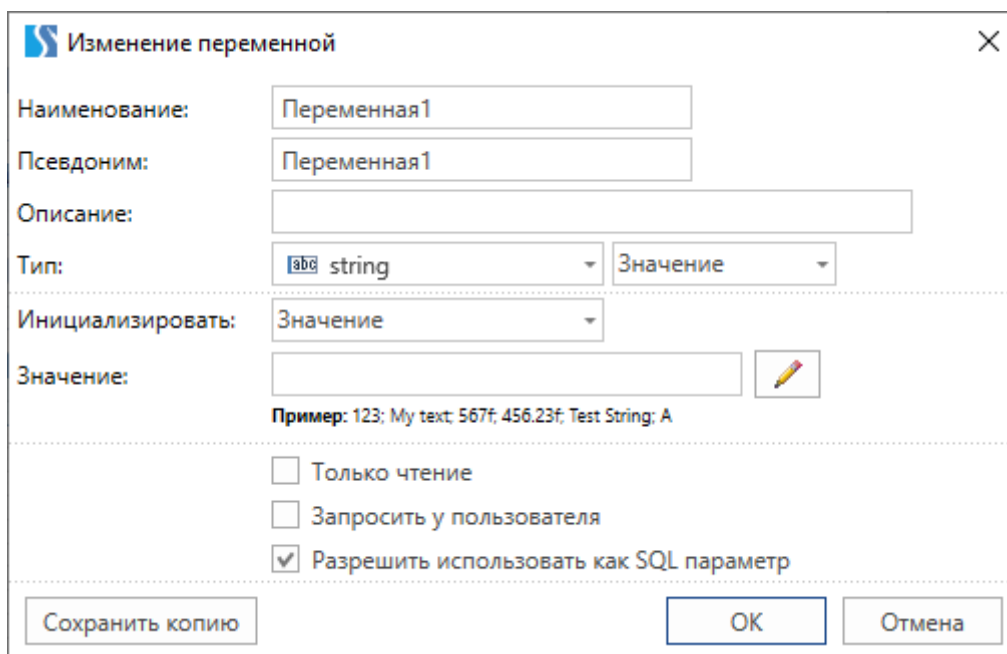
В параметре как значение можно указать переменную. В этом случае, значения переменной будут значениями параметра при запросе. Использовать переменную в запросе как параметр можно двумя способами:

- Создать переменную в словаре данных. Открыть на редактирование запрос. Создать параметр в запросе. Указать переменную как значение этого параметра:



Прописать параметр в тексте запроса.

- При создании или редактировании переменной, установить флаг Разрешить использовать как SQL параметр (Allow using as SQL parameter):

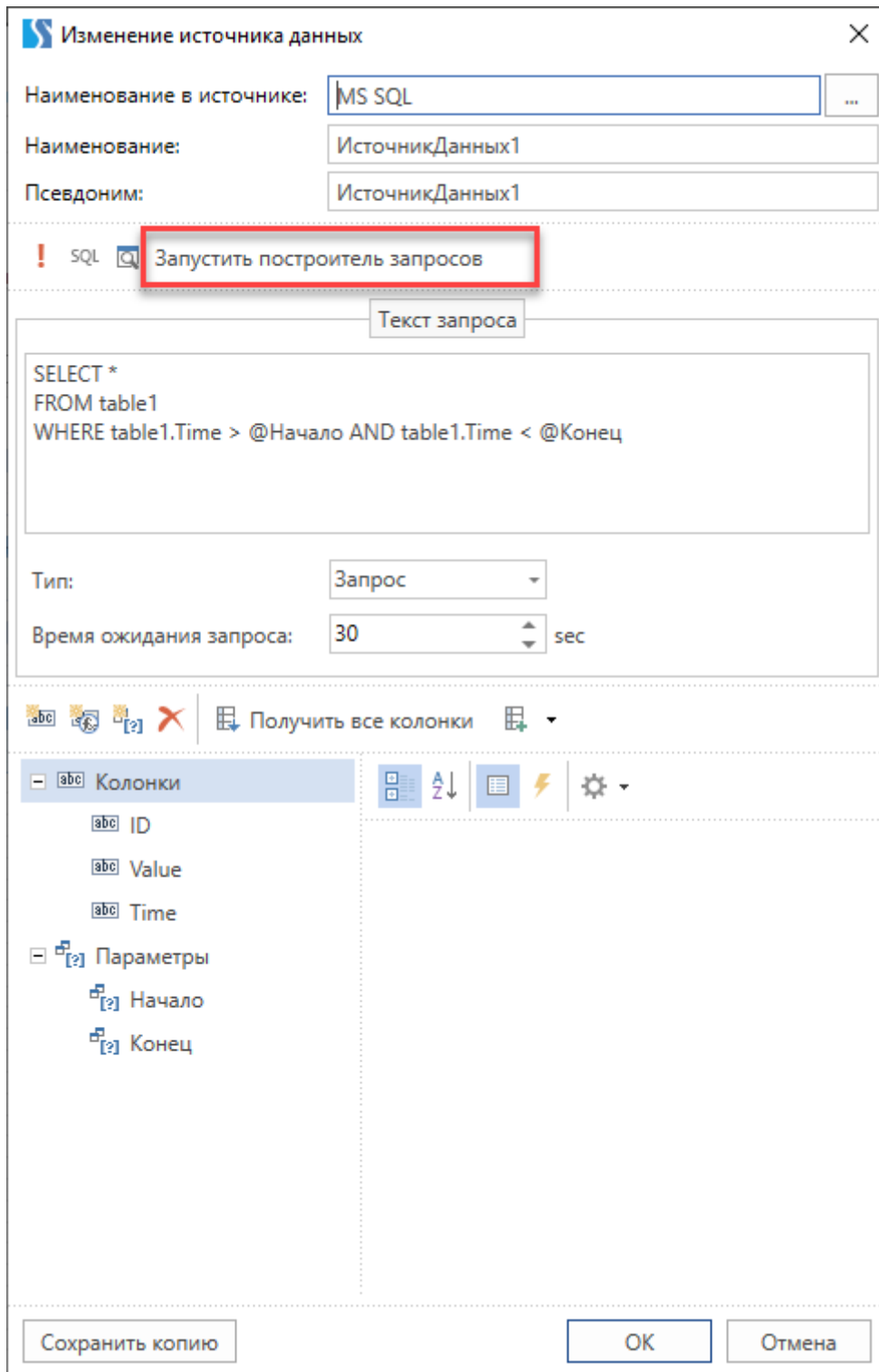


Прописать эту переменную в тексте запроса, используя специальный символ "@" перед именем переменной.

Нажмите кнопку ОК в редакторе запросов. Теперь переменная присутствует в источнике данных, и используется как параметр в запросе.

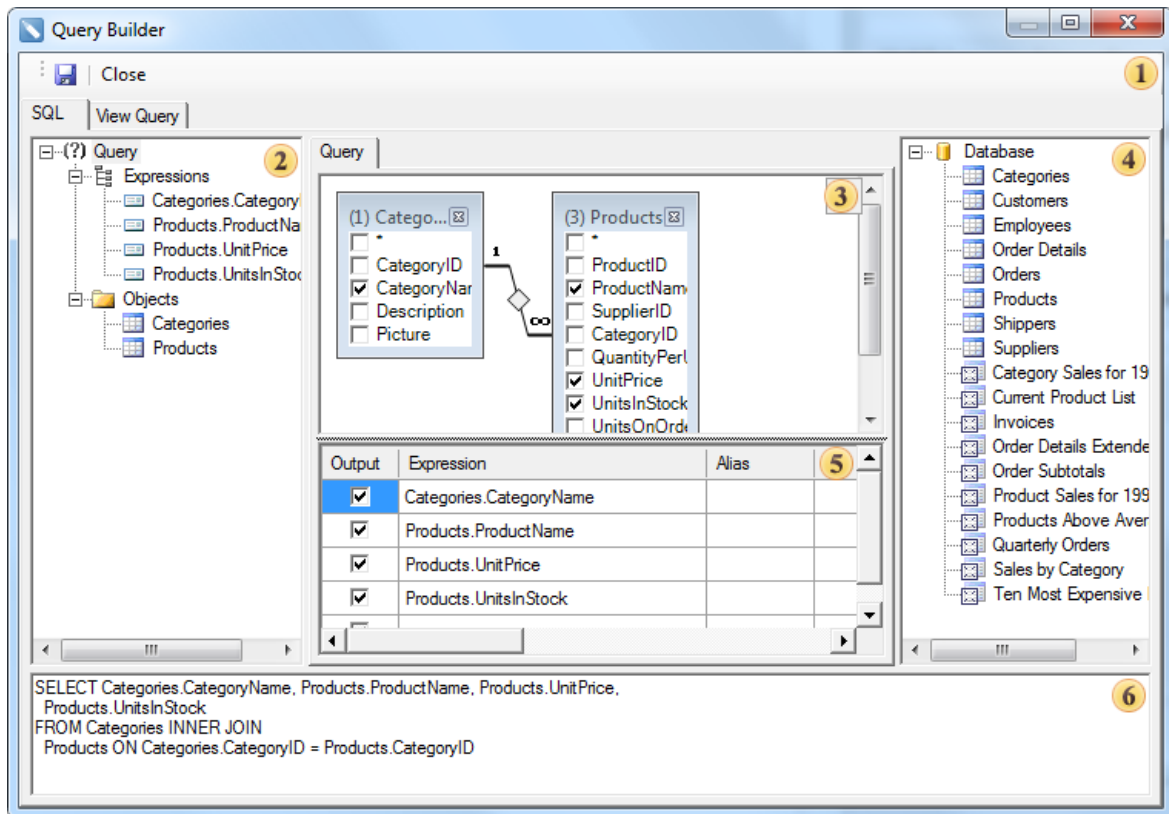
КОНСТРУКТОР (ПОСТРОИТЕЛЬ) ЗАПРОСОВ

Открыть Построитель запросов можно в окне настройки запроса:



Конструктор (Построитель) запросов (Query Builder) представляет собой визуальный компонент, с помощью которого можно создавать запросы визуальными средствами. Создание запроса с помощью конструктора запросов, обеспечивает полное управление параметрами запроса и построение сложных условий отбора данных при помощи простейших визуальных действий пользователя.

Вид окна Конструктора запросов (Query Builder):



1 Панель управления конструктором запросов. Представлена кнопками Сохранить (Save), сохраняет запрос, и Закреть (Close), закрывает конструктор запросов;

2 Панель Дерево запросов. На данной панели можно просмотреть дерево запросов.

3 Панель Конструкция запроса. Данная панель представляет собой область, в которой располагается визуальное представление запроса. В этой области можно определить исходную базу данных объектов и производные источники данных, а также определить связи между источниками данных, настроить свойства источника данных и ссылки.

4 Панель Базы данных. На этой панели отображается база данных и включенные в неё источники данных;

5 Панель таблица. На данной панели расположена таблица, в которой строки - это колонки данных, используемые в запросе, а столбцы - операции. В этой таблице можно определить колонки данных, псевдонимы, тип сортировки, порядок сортировки, группировку, критерии.

6 На данной панели отображается построенный, на панели **3**, запрос в синтаксическом виде.

Также в конструкторе запросов есть вкладка Посмотреть результаты запроса (View), которая предоставляет возможность отобразить колонки данных, выбранные с помощью запроса. Также будут учтены операции в запросе:

View Query


"(Name)"	Product Name	Unit Price	Units In Stock
Beverages	Chutreuse verte	18	69
Beverages	Chang	19	17
Beverages	Guaraná Fantástica	4,5	20
Beverages	Sasquatch Ale	14	111
Beverages	Stoutley Stout	18	20
Beverages	Chai	18	39
Beverages	Cote de Blaye	263,5	17
Beverages	Ipoih Coffee	46	17
Beverages	Laughing Lumberjack Lager	14	52
Beverages	Outback Lager	15	15
Beverages	Rhinbrau Kosterbier	7,75	125
Beverages	Lekkalkkuri	18	57
Condiments	Genen Shouyu	15,5	39
Condiments	Northwoods Cranberry Sauce	40	6
Condiments	Original Frankfurter gine Soße	13	32
Condiments	Grandma's Boysenberry Spread	25	120
Condiments	Gula Malacca	19,45	27
Condiments	Chef Anton's Gumbo Mix	21,35	0
Condiments	Chef Anton's Cajun Seasoning	22	53
Condiments	Aniseed Syrup	10	13
Condiments	Louisiana Fiery Hot Pepper Sauce	21,05	76
Condiments	Louisiana Hot Spiced Okra	17	4

Для того, чтобы добавить построенный запрос в поле Текст запроса (Query Text) следует в конструкторе запроса нажать кнопку Сохранить (Save).

9.3.1.2.2. СОЗДАНИЕ И РЕДАКТИРОВАНИЕ КОЛОНКИ ДАННЫХ

Создание колонки данных

Для создания новой колонки следует выделить источник данных, в который будет добавлена колонка данных, и выбрать пункт Новая колонка... (New Column...) в меню Новый элемент (New Item) или в контекстном меню выделенного источника данных. После выбора данного пункта, будет вызвано окно Новая колонка (New Column), в котором необходимо указать параметры новой колонки. На рисунке снизу представлено окно Новая колонка (New Column):

 Новая колонка ✕

Наименование в источнике:

Наименование:

Псевдоним:

Тип:

Название	Назначение
Наименование в источнике (Name in Source)	Указывается имя в источнике (не в отчете) данных

Название	Назначение
Наименование (Name)	Имя колонки, которое используется для обращения к данной новой колонке в отчете
Псевдоним (Alias)	Указывается псевдоним колонки
Тип (Type)	Указывается тип данных, которые будут содержаться в колонке

После нажатия кнопки ОК, будет создана новая колонка данных в выбранном источнике данных. Следует отметить, что созданная таким образом колонка данных, является лишь описанием (виртуальной) колонки данных и реальных данных она не содержит. Если в базе данных эта колонка отсутствует, то в момент обращения к базе данных, генератор отчетов выдаст ошибку.

Редактирование колонки данных

Колонку данных можно отредактировать. Для этого, следует выбрать пункт Редактировать (Edit) в контекстном меню выделенной колонки или нажать кнопку Редактировать (Edit) на панели инструментов в словаре данных. После этого, пользователю будет отображено окно Редактирование колонки (Edit Column), в котором можно изменить такие параметры как Наименование в источнике (Name in Source), Наименование (Name), Псевдоним (Alias) и Тип (Type) редактируемой колонки. Изменения будут приняты после нажатия кнопки Ок. На рисунке снизу представлено окно Редактирование колонки (Edit Column):

Кнопка Сохранить копию (Save a Copy) сохраняет копию редактируемой колонки данных, с присвоением постфикса Copy в имени колонки данных.

Важно! Нельзя вручную создавать колонку данных в источнике данных Проект [Текущий проект]

9.3.1.2.2.1. РАССЧИТЫВАЕМАЯ КОЛОНКА ДАННЫХ

Рассчитываемая колонка данных вычисляется на основе выражения, в котором могут использоваться другие колонки данных в существующем источнике данных. Выражение может быть именем не вычисляемой колонки, константой, функцией или любым их сочетанием, соединенным одним или несколькими операторами. Выражение не может быть вложенным запросом. Рассчитываемая колонка данных является виртуальной колонкой, которая физически в источнике данных не хранится. Значения рассчитываемой колонки данных вычисляются каждый раз при обращении к ним запроса. Также, значения рассчитываемой колонки обновляются каждый раз при изменении колонок, входящих в вычисляемое выражение. Прежде чем добавить рассчитываемую колонку, необходимо подключить хотя бы один источник данных. Рассмотрим создание рассчитываемой колонки данных в источнике данных Auto. В данном источнике находятся колонки: Rank, Country, Year2000, Year2005, Year2009. В колонках Year2000, Year2005, Year2009 содержатся данные о количестве выпускаемых автомобилей в 2000, 2005 и 2009 году. Создадим рассчитываемую колонку данных, которая будет содержать данные о росте выпускаемых автомобилей в 2009 по отношению к 2000 году, результат отобразим в процентах. На рисунке снизу представлены колонки данных Year2000 и Year2009:

Year2000	Year2009
2069069	13790994
10140796	7934516
12799857	5711823
5526615	5209853
3114998	3512916
1681517	3182617
801360	2632694
3032874	2170078
3348361	2049762
1935527	1557290

Для того, чтобы создать новую рассчитываемую колонку необходимо вызвать окно Новая рассчитываемая колонка (New Calculated Column) и заполнить поля в диалоговой форме. Вызвать окно можно из контекстного меню источника данных или из меню Новый элемент.

Вид окна:

Название	Назначение
Наименование (Name)	Имя колонки, которое используется для обращения к данной рассчитываемой колонке в отчете
Псевдоним (Alias)	Псевдоним колонки
Тип (Type)	Кнопка содержит выпадающее меню в котором отображается структура словаря данных. В данном меню можно выбрать колонки данных, бизнес-объекты, системные или переменные, которые будут добавлены в выражение вычисления рассчитываемой колонки данных.
Словарь данных (Dictionary)	Кнопка содержит выпадающее меню в котором отображается структура словаря данных. В данном меню можно выбрать колонки данных, бизнес-объекты, системные или переменные, которые будут добавлены в выражение вычисления рассчитываемой колонки данных.
Значение (Value)	Указывается выражение вычисления значений новой рассчитываемой колонки данных

В данном примере, выражение вычисления будет содержать колонки данных Year2000 и Year2009 из источника Auto, а тип данных в новой рассчитанной колонке будет double. После того, как колонка будет создана, следует разместить в отчете на бэнде Данные (DataBand) текстовый компонент с ссылкой на эту колонку данных. В описанном примере, текстовый компонент будет содержать ссылку {Auto.NewCalculatedColumn1}. Поскольку,

результат вычисления необходимо отобразить в процентах, то у данного текстового компонента следует изменить формат, т.е. установить формат Процентный (Percentage). На рисунке снизу представлен отчет с рассчитываемой колонкой данных:

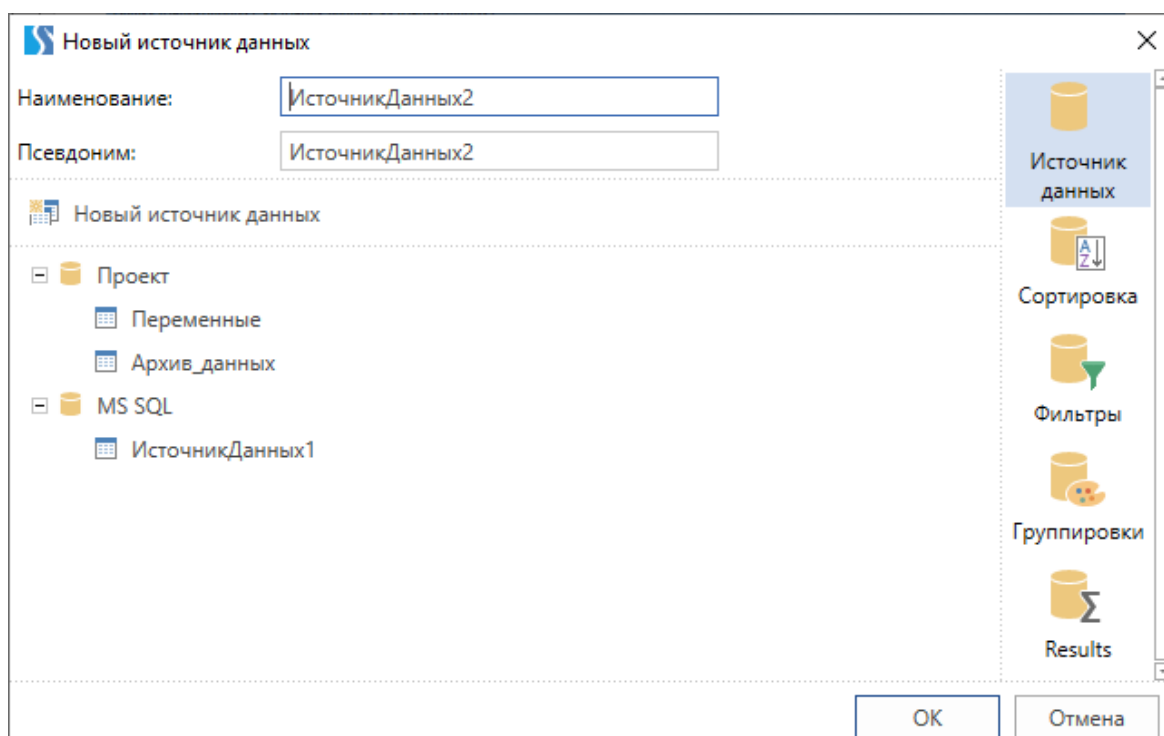
Year2000	Year2009	NewCalculatedColumn1
2069069	13790994	15,00%
10140796	7934516	127,81%
12799657	5711823	224,09%
5526615	5209853	106,08%
3114998	3512916	88,67%
1681517	3182617	52,83%
801380	2632894	30,44%
3032874	2170078	139,76%
3348361	2049762	163,35%
1935527	1557290	124,29%

9.3.1.2.3. ДАННЫЕ НА ОСНОВЕ ДРУГИХ ИСТОЧНИКОВ ДАННЫХ

В генераторе отчетов можно создать источник данных на основе уже существующих (добавленных ранее в отчет) источников данных. Источник данных Данные на основе других источников (Data from other DataSource) предоставляет аналогические возможности, что и Запрос к базе данных. При создании такого источника данных можно с помощью визуального интерфейса, в процессе создания источника данных, произвести сортировку, группировку, фильтрацию и вычисление итогов с помощью агрегатных функций. Рассмотрим пример создания данных на основе других источников. Допустим, есть Master-Detail отчет, в котором каждой категории соответствуют некоторое количество продуктов. На рисунке снизу представлена страница Master-Detail отчета (представлена частично):

Beverages		
Chai	18	39
Chang	19	17
Guaranó Fantástica	4.5	20
Sasquatch Ale	14	111
Steeleye Stout	18	20
Côte de Blaye	263.5	17
Chartreuse verte	18	69
Ipon Coffee	46	17
Laughing Lumberjack Lager	14	52
Outback Lager	15	15
Rhinbräu Klosterbier	7.75	125
Lakkalikööri	18	57

Как видно из рисунка, в отчете отображается название категории, имя продукта (относящегося к данной категории) и цена продукта. Если необходимо создать отчет, в котором будут отображаться название категории и общая стоимость всех продуктов входящих в данную категорию, то это можно выполнить различными способами. Но наиболее простым способом будет создание источника данных на основе других данных. Для этого, следует в окне Новый источник данных (New Data Source) выбрать пункт Данные на основе других данных (Data from other Data Source) и в следующей диалоговой форме окна новый источник данных, произвести настройку создаваемого источника данных. На рисунке снизу представлена вторая форма окна Новый источник данных (New Data Source):



Как видно из рисунка, процесс создания данных на основе других источников состоит из нескольких этапов. Каждый этап настройки производится последовательно на вкладках, которые находится справа в окне:

Название вкладки	Порядок работы
Источник данных (Data Source)	На данном этапе необходимо указать Наименование (Name) нового источника данных и его Псевдоним (Alias). Также, следует выбрать источник данных, на основе которого будет создан новый. Данный этап является обязательным.
Сортировка (Sort)	На данном этапе следует указать колонку данных, по которой будет осуществляться сортировка, и выбрать направление сортировки. Данный этап не является обязательным.

Название вкладки	Порядок работы						
Фильтры (Filters)	Для фильтрации данных необходимо добавить фильтр, указать выражение или условие, согласно которому будет осуществляться фильтрация. Данный этап не является обязательным.						
Группировка (Groups)	Для группировки данных следует указать колонку данных, по которой будут данные группироваться, и выбрать направление расположения групп. Колонка данных, по которой будет осуществляться группировка, будет присутствовать в новом источнике данных. Данный этап не является обязательным.						
Результаты (Results)	<p>На данном шаге можно произвести вычисление по колонке данных с помощью агрегатных функций:</p> <div data-bbox="453 949 1426 1048" style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 40%; border-bottom: 1px solid #ccc;">Колонка</td> <td style="width: 20%; border-bottom: 1px solid #ccc;">Агрегатная функци:</td> <td style="width: 40%; border-bottom: 1px solid #ccc;">Наименование</td> </tr> <tr> <td style="border: 1px solid #ccc;">Начало</td> <td style="border: 1px solid #ccc;">Sum</td> <td style="border: 1px solid #ccc;">Начало.Sum</td> </tr> </table> </div> <p>На данной вкладке следует указать следующие параметры:</p> <ul style="list-style-type: none"> • Поле Колонка (Column) колонку данных, которая будет присутствовать в новом источнике данных или из которой будут браться данные для расчета агрегатной. Данное поле является обязательным для заполнения.. • Поле Агрегатная Функция (Aggregate Function) представляет собой список агрегатных функций, которые могут быть использованы для вычисления выбранной колонки данных. Агрегатную функцию можно не указывать, в этом случае колонка данных будет содержать данные, такие же как в колонке данных, которая берется за основу. • В поле Наименование (Name) указывается имя колонки, которое используется для обращения к данной рассчитываемой колонке в отчете. 	Колонка	Агрегатная функци:	Наименование	Начало	Sum	Начало.Sum
Колонка	Агрегатная функци:	Наименование					
Начало	Sum	Начало.Sum					

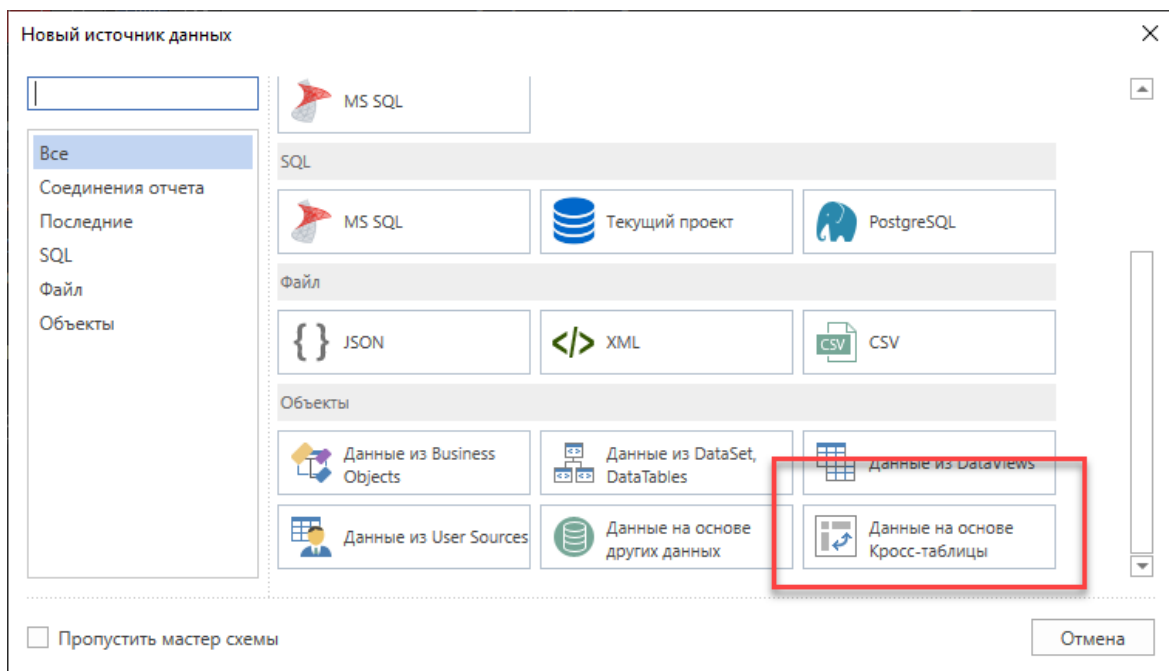
9.3.1.2.4. ДАННЫЕ ИЗ КРОСС-ТАБЛИЦ

В генераторе отчетов есть возможность создать источник данных на основе кросс-таблицы, т.е. можно создать новый источник, колонками которого будут являться колонки построенной кросс-таблицы, а строками - строки построенной кросс-таблицы. Рассмотрим

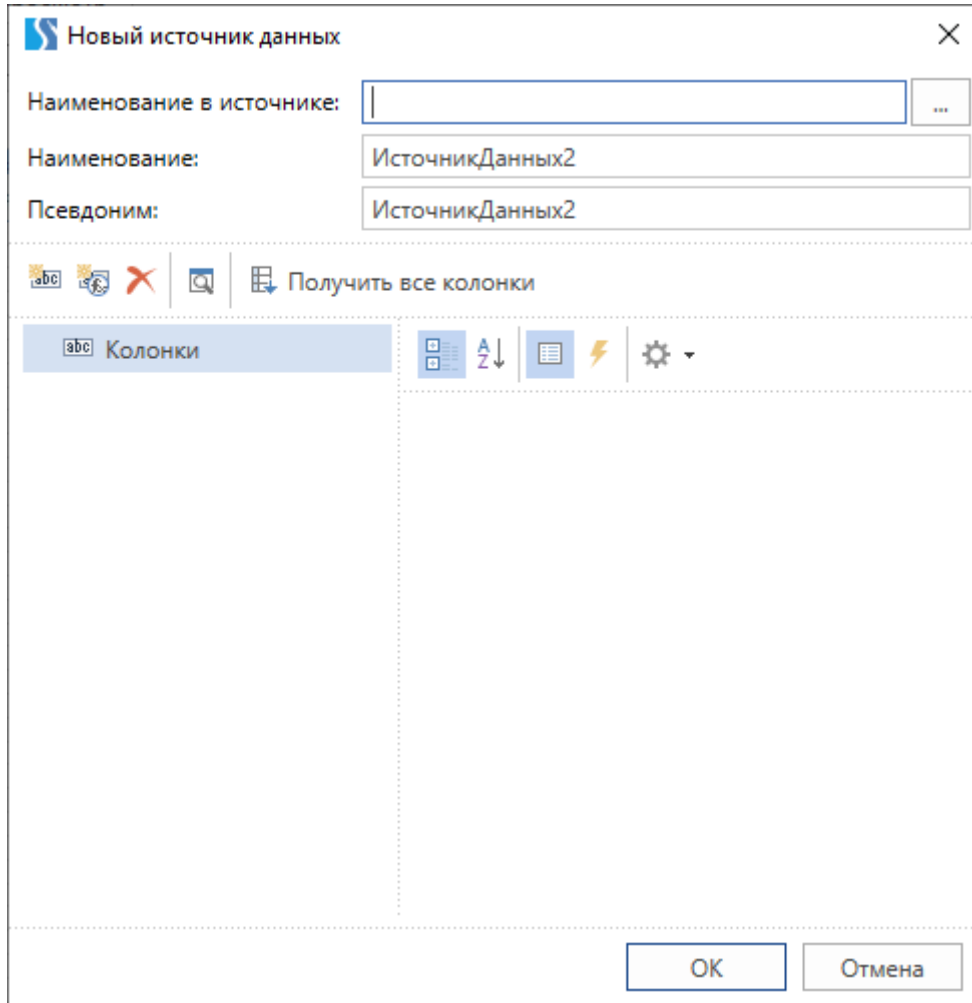
на примере, создание источника данных на основе кросс-таблицы. На рисунке снизу представлена страница отчета с построенной кросс-таблицей:

Products	CategoryName								
Country	Beverages	Condiments	Confections	Dairy Products	Grains/Cereals	Meat/Poultry	Produce	Seafood	Total
Australia	15	43.9	17.45		7	71.8	53	62.5	270.65
Brazil	4.5								4.5
Canada		28.5	49.3			31.45			109.25
Denmark								21.5	21.5
Finland	18		36.25						54.25
France	281.5			89				13.25	383.75
Germany	7.75	13	89.13		33.25	123.79	45.6	25.89	338.41
Italy				79.3	57.5				136.8
Japan		15.5				97	33.25	37	182.75
Netherlands			22.25						22.25
Norway				60.0					60.0
Singapore	46	19.45			14				79.45
Spain				59					59
Sweden								60	60
Sweden					30				30
UK	37	10	112.7						159.7
USA	46	146.40					30	28.05	250.45
Total	455.75	276.75	327.08	287.3	141.75	324.04	161.65	248.19	2222.71

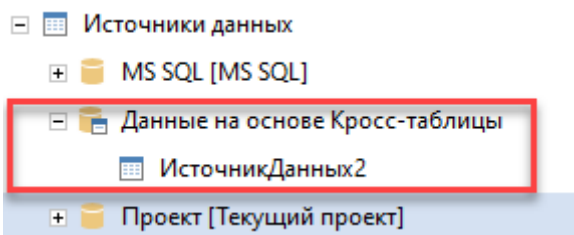
Для того, чтобы создать источник данных на основе кросс-таблицы, следует вызвать окно Новый источник данных (New Data Source) и выбрать пункт Данные на основе Кросс-таблицы (Data from Cross-Tab). На рисунке снизу представлено окно Новый источник данных (New Data Source):



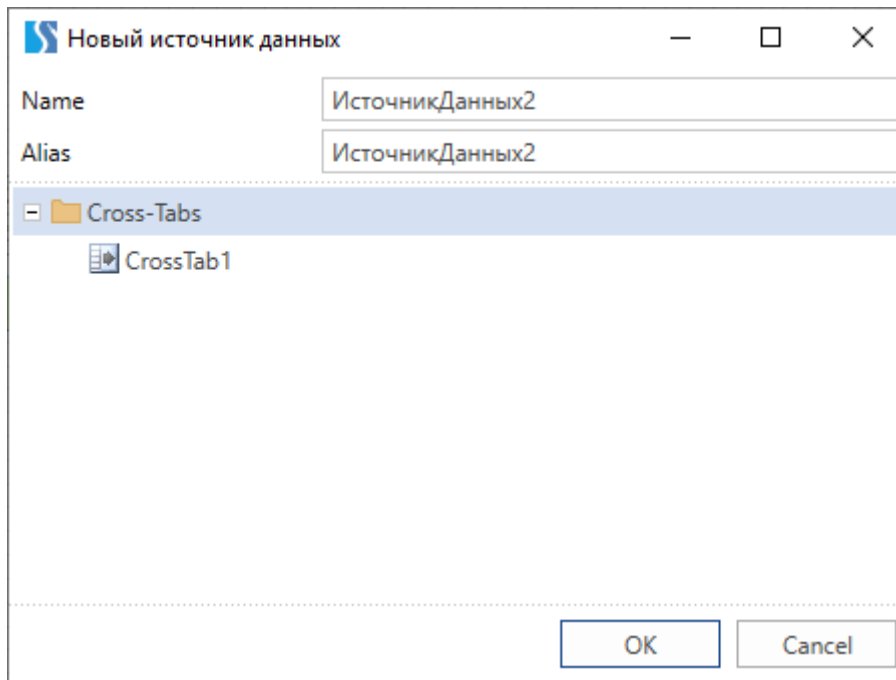
Откроется окно Новый источник данных (New Data Source), следует указать Наименование (Name) нового источника данных



После нажатия кнопки ОК получим результат в словаре:



Далее необходимо дважды кликнуть по новому источнику данных. Откроется вторая форма окна Новый источник данных (New Data Source):



В окне выбирается нужная кросс-таблица, которая ранее была добавлена в отчет:

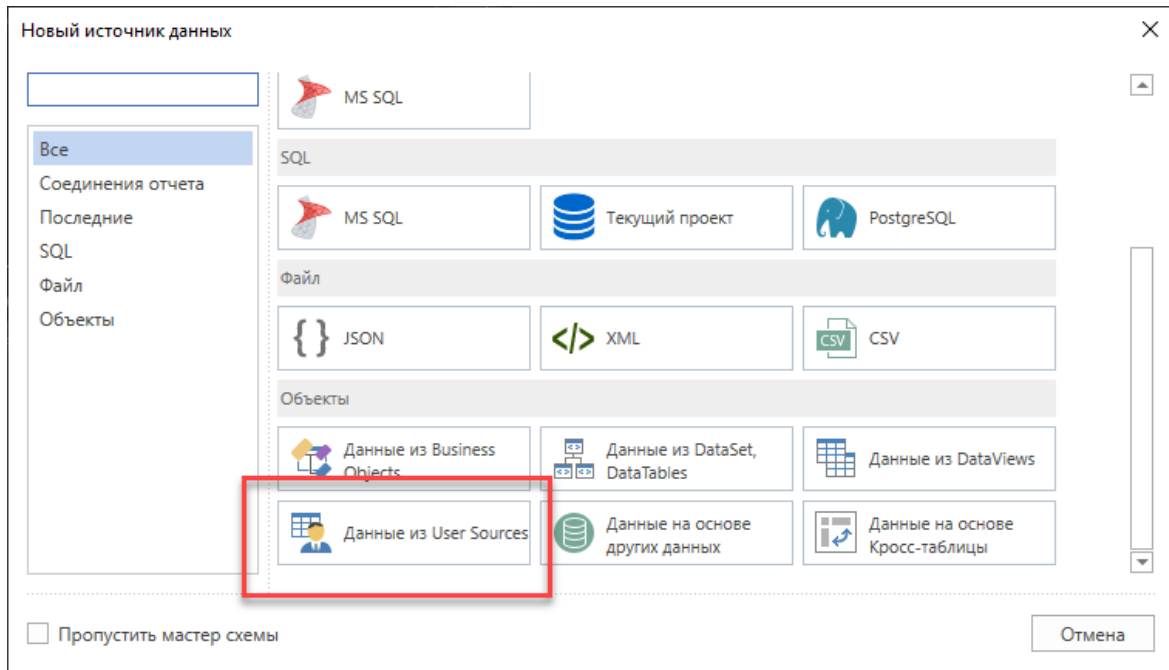
После нажатия кнопки ОК, будет создан источник данных, который будет содержать в себе колонки, которые указаны в кросс-таблице источнике. Источник данных на основе кросс-таблицы представляет собой виртуальный источник данных, который не содержит реальные данные. Заполнение этого источника происходит в момент построения кросс-таблицы. Поэтому отчет, в котором будет использоваться этот источник данных, к примеру, для построения отчета со списком, должен содержать кросс-таблицу, на основе которой создан данный источник:

UK	Beverages	37
UK	Condiments	10
USA	Condiments	146.40
USA	Produce	30
Japan	Meat/Poultry	97
Japan	Seafood	37
Spain	Dairy Products	59
Japan	Produce	33.25
Japan	Condiments	15.5
Australia	Confections	17.45
Australia	Meat/Poultry	71.8

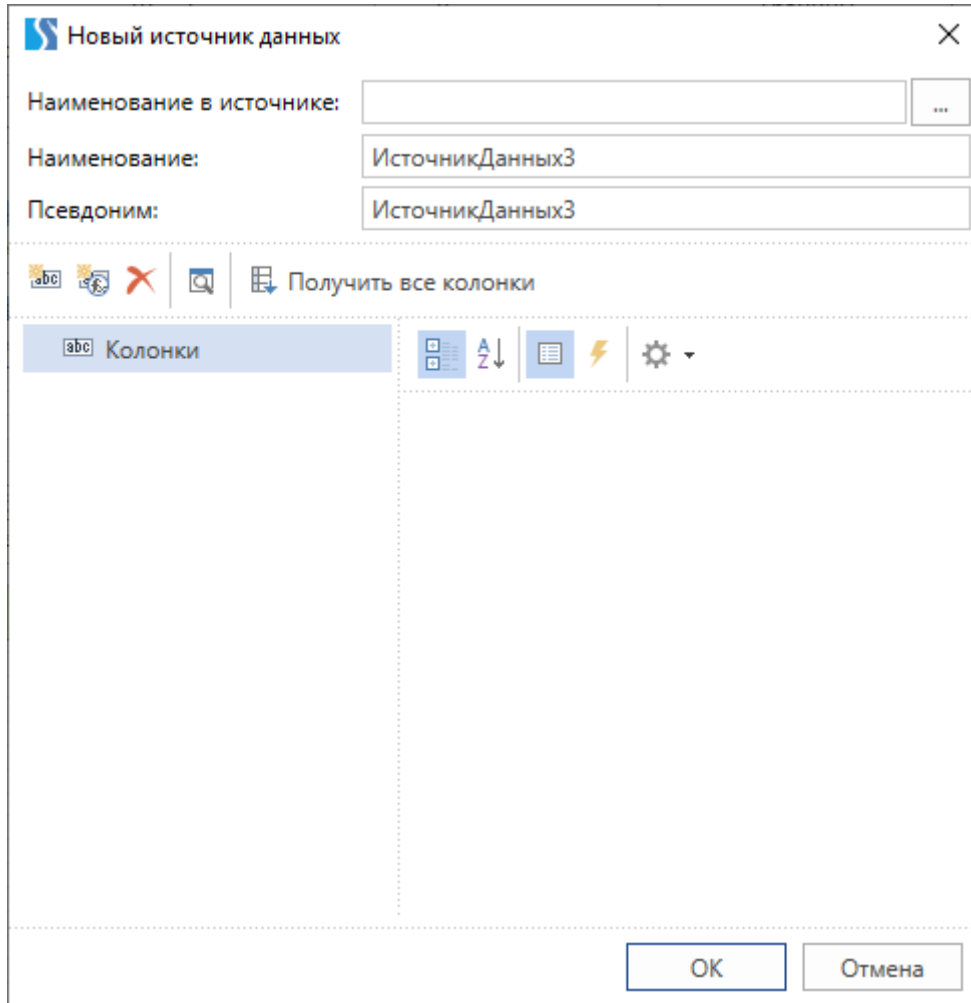
При построении отчета, генератор отчетов заполнил созданный источник данных данными из кросс-таблицы и отобразил данные в виде списка.

9.3.1.2.5. ПОЛЬЗОВАТЕЛЬСКИЕ ИСТОЧНИКИ ДАННЫХ

Если необходимо построить отчет на основе данных пользователя, то в генераторе отчетов есть возможность создания данных на основе пользовательских источников. Для этого, вызвать окно Новый источник данных (New Data Source) и в нем выбрать пункт Данные из UserSources (Data from User Sources):



В следующем окне произвести настройку пользовательского источника данных:



В верхней части окна находятся основные элементы управления, предназначенные для работы с источником данных. В нижней части, находится список добавленных колонок. Если какая-либо колонка выделена, то справа от нее указываются ее свойства.

Элементы управления:

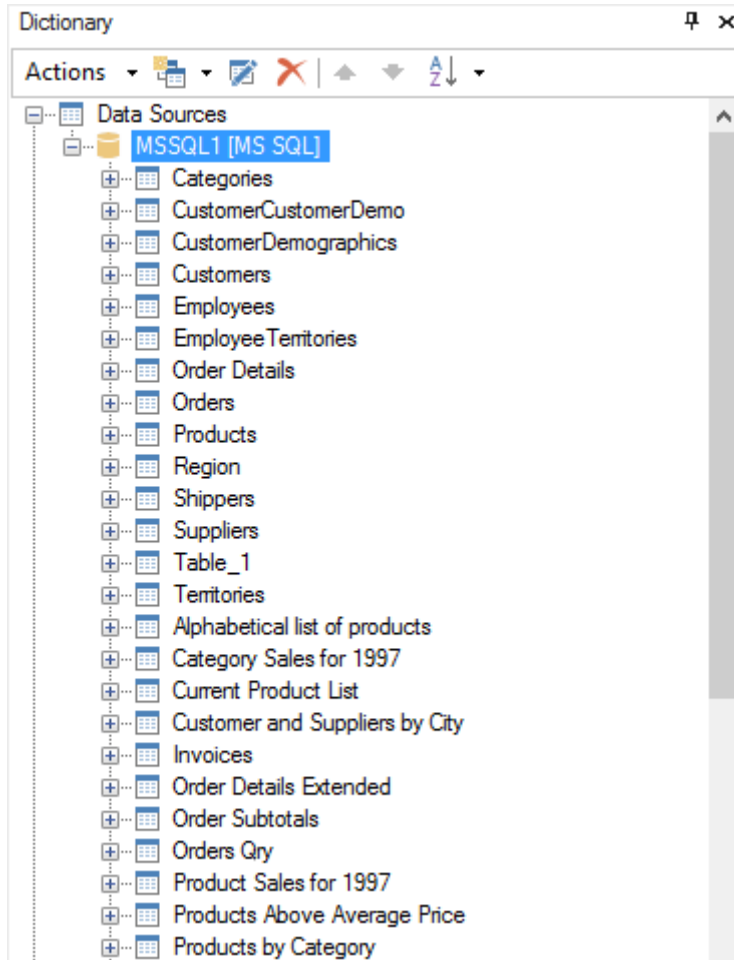
Элемент	Назначение
Наименование в источнике (Name in Source)	При создании данных на основе пользовательских источников данное не является обязательным для заполнения.
Наименование (Name)	Имя источника, которое используется для обращения в отчете
Псевдоним (Alias)	Псевдоним источника

Элемент	Назначение
Новая колонка (New Column)	Можно добавить новую колонку в источник данных
Новая рассчитываемая колонка (New Calculated Column)	Добавляет рассчитываемую колонку
Удалить (Delete)	Удаляет ранее созданную колонку, которая выделена в нижней части окна.
Команда просмотра запроса	Не используется
Получить все колонки (Retrieve Columns)	Не используется

9.3.1.2.6. ПОЛУЧИТЬ ТОЛЬКО ИСПОЛЬЗУЕМЫЕ ДАННЫЕ

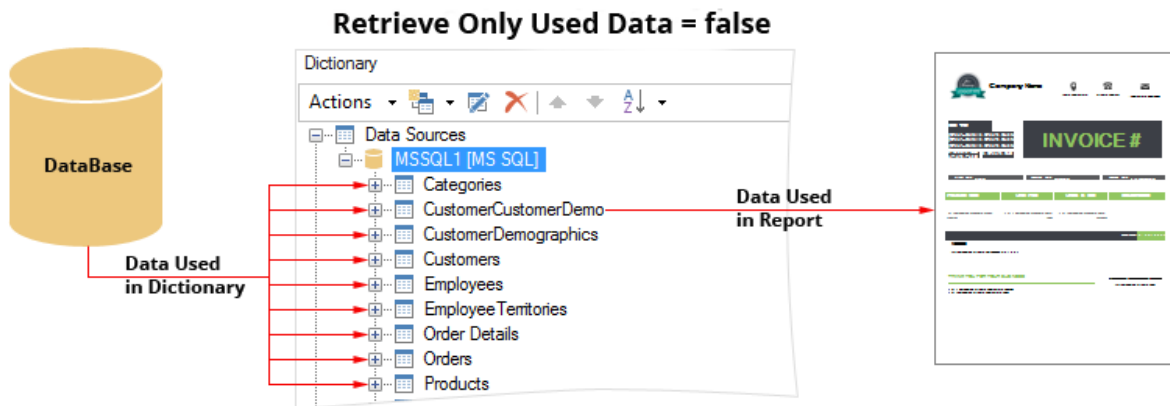
Иногда бывает достаточно изменить значение одного свойства, чтобы значительно увеличить скорость построения отчета. При работе с шаблоном отчета, словарь данных не содержит реальных данных. В словаре расположено лишь описание структуры данных. Выполнение всех запросов и передача данных из хранилища осуществляется в момент построения отчета. При этом, происходит заполнение всей структуры словаря реальными данными. Иначе говоря, если в словаре создано 200 источников данных, то во все эти источники передаются реальные данные из хранилища. Чем больше данных необходимо передать из хранилища в словарь, тем дольше время построения отчета. Однако, не всегда все источники данных используются в самом отчете. Для того чтобы значительно сократить время построения отчета, получив реальные данные только для используемых источников данных в отчете, следует установить свойство отчета `Retrieve Only Used Data` в значение `true`.

Рассмотрим пример. Допустим, используется MS SQL база данных, которая содержит таблицы данных, хранимые процедуры и представления. Например,



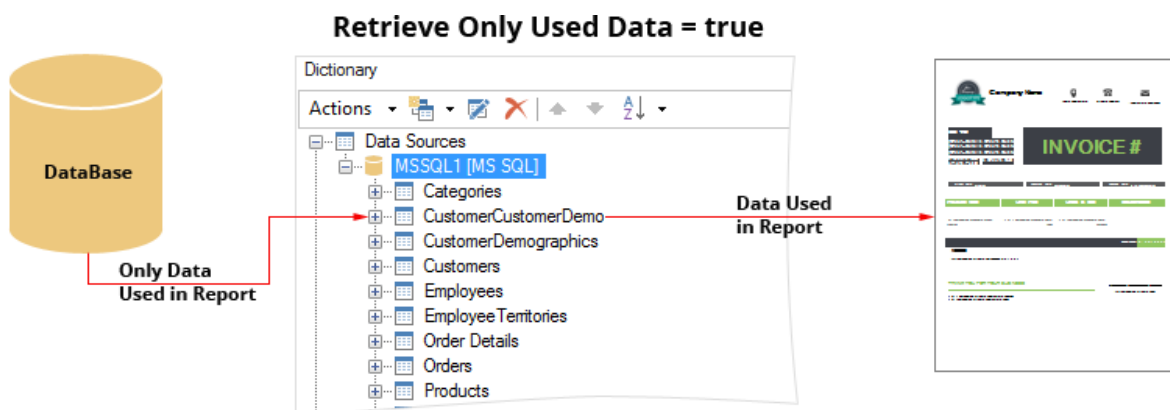
Каждая таблица данных содержит от одного до множества колонок данных, минимум с одной строкой данных. Допустим, в отчете будет использоваться только источник данных CustomerCustomerDemo.

Свойство отчета Retrieve Only Used Data установлено в значение false. В этом случае, при построении отчета, данные будут передаваться из базы данных для каждой таблицы в словаре данных, а затем из словаря в сам отчет. Иначе говоря, каждая таблица будет заполняться реальными данными. Далее, генератор отчетов, выберет данные используемые в самом отчете и отобразит их в структурированном виде. Время построения отчета зависит от скорости передачи данных и от их объема. Чем быстрее данные будут переданы, тем быстрее будет построен отчет. На рисунке снизу, схематично представлена передача данных, если свойство Retrieve Only Used Data установлено в значение false:



Свойство `Retrieve Only Used Data` установлено в значение `true`. В этом случае, при построении отчета, генератор отчетов проанализирует структуру отчета и только для используемых в этом отчете таблиц, будут переданы данные. В текущем примере, данные будут переданы только для таблицы `CustomerCustomerDemo`. Время построения отчета, при этом будет значительно меньше. Если же в отчете будет использоваться несколько таблиц, то данные будут передаваться только для этих нескольких таблиц. На рисунке снизу, схематично представлена передача данных, если свойство `Retrieve Only Used Data` установлено в

значение `true`



Альтернативный вариант является способ удаления из словаря данных неиспользуемые источники данных. Однако, иногда необходимо, чтобы присутствовала вся структура. Например, для дальнейшей разработки отчета или, скажем, когда один и тот же словарь используется для множества отчетов.

9.3.2. ПЕРЕМЕННЫЕ

В редакторе отчетов можно использовать Переменные (Variables) в отчете. Переменная (Variable) представляет собой возможность для размещения и использования при построении отчета какого-либо значения. Значения могут быть различного типа: строка,

дата, время, число, массив, коллекция, диапазон и т.д. Все переменные, после их создания, хранятся в словаре данных. Пока ни одна переменная не создана, то соответствующая группа может отсутствовать в словаре.

При этом переменные могут быть связанными с переменными дерева объектов MasterSCADA или быть локальными - созданными непосредственно в редакторе отчетов:

- Переменные источника данных Текущий проект
- Локальные переменные

В отчетах также существуют системные переменные. Системные переменные - это переменные, которые сообщают, какую либо информацию о текущем состоянии отчета, постоянно находятся в словаре отчета

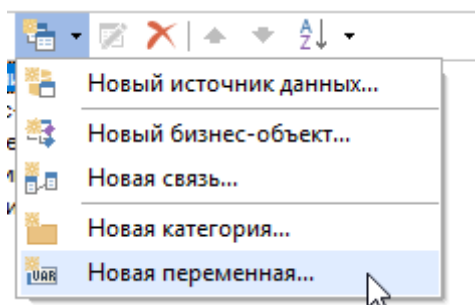
9.3.2.1. ЛОКАЛЬНЫЕ ПЕРЕМЕННЫЕ

Локальные переменные - переменные создаваемые непосредственно в отчете и не связанные с переменными MasterSCADA 4D

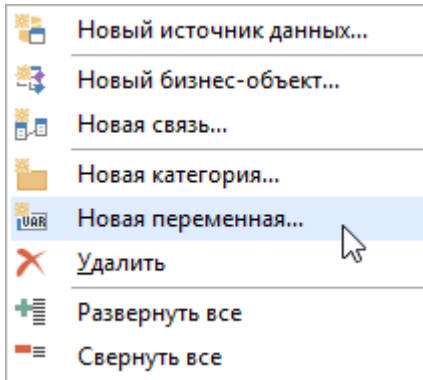
Предназначение данных переменных - промежуточные вычисления. В данных переменных, используя События, можно сохранять значения колонок (например находить максимальное значение или значение на определенный момент времени), а также выполнять предварительную обработку - например, по переменной MasterSCADA 4D передающей месяц, за который нужно сформировать отчет, вычислить начало и конец интервала для источника данных.

9.3.2.1.1. СОЗДАНИЕ ПЕРЕМЕННЫХ

Добавить локальную переменную можно выбрав пункт Новая переменная... (New Variable...) в меню Новый элемент (New Item) словаря данных:

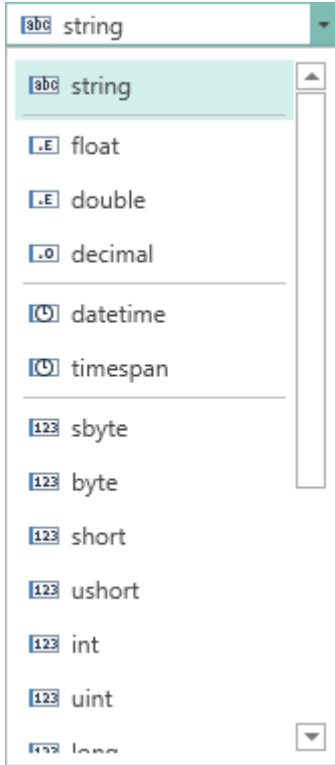
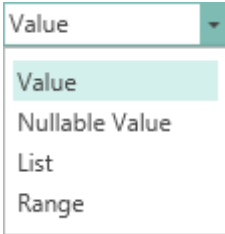


Также создать новую переменную можно выбрав пункт Новая переменная... (New Variable...) в контекстном меню Словаря данных (Dictionary):



После выбора данного пункта будет отображено окно Новая переменная (New Variable), в котором необходимо определить параметры переменной:

Название	Назначение
Наименование (Name)	Указывается имя переменной используемое в отчете
Псевдоним (Alias)	Указывается имя переменной, которое выводится пользователю
Описание (Description)	Можно указать пояснения к переменной
Тип (Type)	Позволяет изменить тип данных, которые будут размещены в переменной, и вид переменной. Данное поле представлено двумя полями с выпадающими списками.:

Название	Назначение
	<ul style="list-style-type: none"> Первый список перечень всех доступных типов данных разбитых на категории:  Второй список содержит перечень видов переменной. В зависимости от вида переменной могут отображаться или не отображаться некоторые дополнительные поля параметров. Перечень видов переменной представлен во втором списке поля Тип (Type) (см. рисунок выше). На рисунке снизу представлен список видов переменной:  <p>Переменная может быть следующих видов: Значение (Value), Значение с null (Nullable Value), Список (List), Диапазон (Range).</p>
Инициализировать	Определяет способ инициализации.

Название	Назначение
	<ul style="list-style-type: none"> • Значение - переменная инициализируется значением заданным в поле Значение. • Выражение - в этом режиме инициализация выполняется кодом на языке C#. В этом режиме есть возможность работать с другими переменными, то есть можно инициализировать переменную результатом расчета над другой переменной. Например, можно прибавить в коде к переменной времени 1 месяц и получить дату конца отчета. <p>Важно! При инициализации выражением нужно помнить, что если переменная в коде инициализации обращается к другим переменным, то у нее необходимо установить флаг Только чтение. Изменить эту переменную в процессе построения отчета нельзя (то есть в нее нельзя будет что-либо записывать в событиях. Если же вам требуется изменять значение переменной в процессе построения отчета через события, то используйте способ инициализации Значение, а затем присваивайте переменной значение, используя событие Начало построения отчета (см. События)</p>
Только чтение (Read Only)	<p>Устанавливает режим только чтения, т.е. в данном случае будет возвращаться значение хранимое в переменной и изменить его пользователь не сможет. При этом, если значение инициализируется как выражение, то в момент обращения к нашей переменной, выражение будет вычисляться каждый раз заново.</p>
Запросить у пользователя (Request from User)	<p>Устанавливает режим, при котором возвращаемое значение может быть изменено пользователем. Следует отметить, что если установлен параметр Запросить у пользователя (Request from User), то будет отображаться дополнительная панель. На этой панели расположены настройки переменной, которые определяют возможности взаимодействия с пользователем, в случае использования данной переменной в отчете. Кроме того, окно Новая переменная (New Variable) может видоизменяться.</p>

Название	Назначение
	Важно! При построении отчетов в исполнительной системе рекомендуется значения всех переменных задавать при помощи элементов окна визуализации. Т.е. из окна передавать значение в параметр дерева MasterSCADA 4D, а затем в переменную источника данных Текущий проект
Использовать как параметр SQL запроса (Allow using as SQL parameter)	Предоставляет возможность использовать переменную как параметр в запросе при выборке из базы данных.

При редактировании ранее созданной переменной, в окне будет отображена кнопка Сохранить копию (Save a Copy). При нажатии на данную кнопку, произойдет создание копии редактируемой переменной, с присвоением постфикса Copy в имени переменной.


9.3.2.1.2. ПАНЕЛЬ ЗАПРОСИТЬ У ПОЛЬЗОВАТЕЛЯ

На панели Запросить у пользователя (Request from user) располагаются параметры. Данные параметры определяют возможности пользовательского участия при использовании переменной в отчете. Некоторые параметры могут присутствовать или отсутствовать, в зависимости от значения поля Источник данных (Data Source). На рисунке снизу представлена панель Запросить у пользователя (Request from user), если в поле Источник данных (Data Source) выбрано значение Колонки данных (Data Columns):

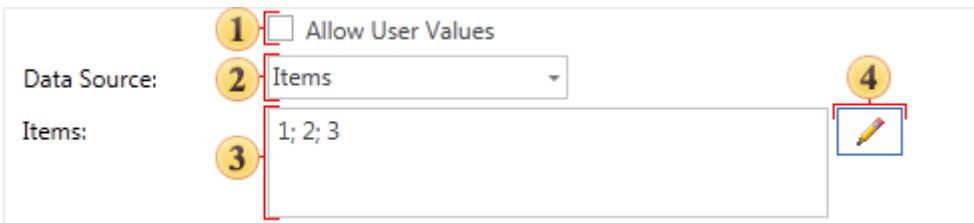
1 Параметр Разрешить значения пользователя (Allow User Values) предоставляет возможность установить режим диалога, т.е. при использовании данной переменной в отчете от пользователя может потребоваться ввод значения.

2 Поле Источник данных (Data Source) содержит выпадающий список значений. В зависимости от выбранного значения: Элементы (Items) или Колонки данных (Data Columns), на данной панели будут поле Элементы (Items) или поля Ключи (Keys) и Значения (Values).

3 В поле Ключи (Keys), с помощью кнопки , выбирается колонка данных, записи которой будут являться ключами.

4 В поле Значения (Values), с помощью кнопки , выбирается колонка данных, записи которой будут являться значениями.

Если в поле Источник данных (Data Source) выбрано значение Элементы (Items), то на панели Запросить у пользователя (Request from user) будут располагаться другие параметры. На рисунке снизу представлена панель Запросить у пользователя (Request from user):



1 Параметр Разрешить значения пользователя (Allow User Values) предоставляет возможность установить режим диалога, т.е. при использовании данной переменной в отчете от пользователя может потребоваться ввод значения.

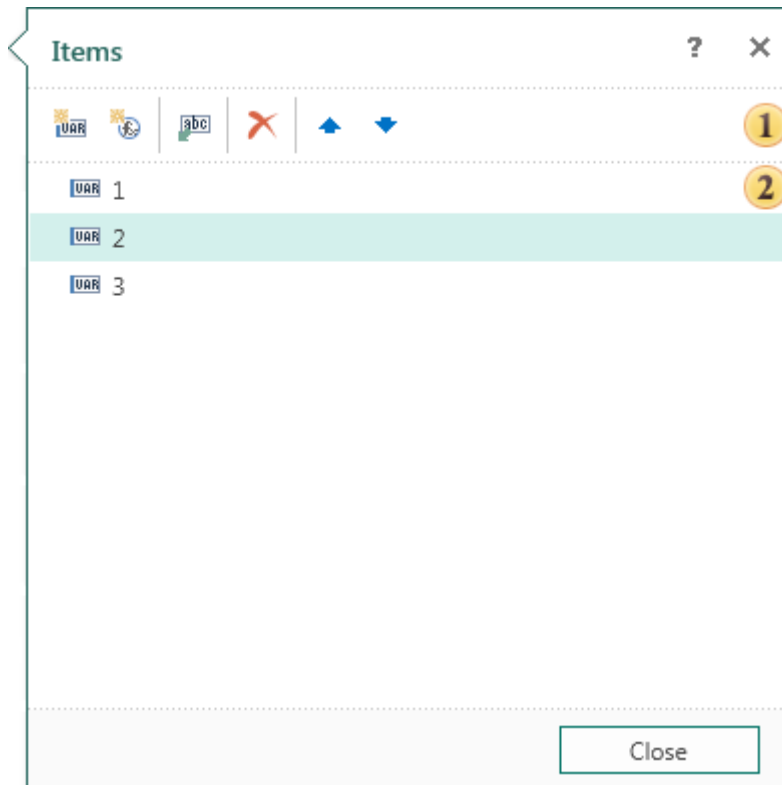
2 Поле Источник данных (Data Source) содержит выпадающий список значений. В зависимости от выбранного значения: Элементы (Items) или Колонки данных (Data Columns), на данной панели будут располагаться поле Элементы (Items) или поля Ключи (Keys) и Значения (Values).

3 В поле Элементы (Items) отображается список созданных элементов переменной. Если элементы не созданы, то данное поле будет пустым. Следует отметить, что очередность элементов в данном списке зависит от их очередности на панели списка в окне Элементы (Items), чем элемент выше - тем он левее в данном списке, и наоборот.

4 Кнопка Редактор (Editor) вызывает окно Элементы (Items), в котором можно создать новые элементы, удалить существующие или отредактировать их.

9.3.2.1.2.1. ОКНО ЭЛЕМЕНТЫ

В окне Элементы (Items) осуществляется создание, удаление, редактирование элементов (значений, выражений). Данное окно вызывается после нажатия кнопки Редактор (Editor) в окне Переменные (Variable). На рисунке снизу представлено окно Элементы (Items):



Как видно из рисунка, в данном окне расположены следующие панели:

- 1 Панель управления. На данной панели расположены кнопки управления элементами.
- 2 На панели элементов отображается список созданных элементов (значений, выражений). Следует помнить, что очередность элементов в данном списке влияет на очередность элементов в поле Элементы (Items) на панели Запросить у пользователя (Request from User).
- 3 Панель свойств. На данной панели отображаются свойства выбранного элемента. У элемента есть два свойства: Ключ (Key) и Значение (Value).
- 4 Tool Tip поля Значение (Value). Содержит описание поля.

Панель управления

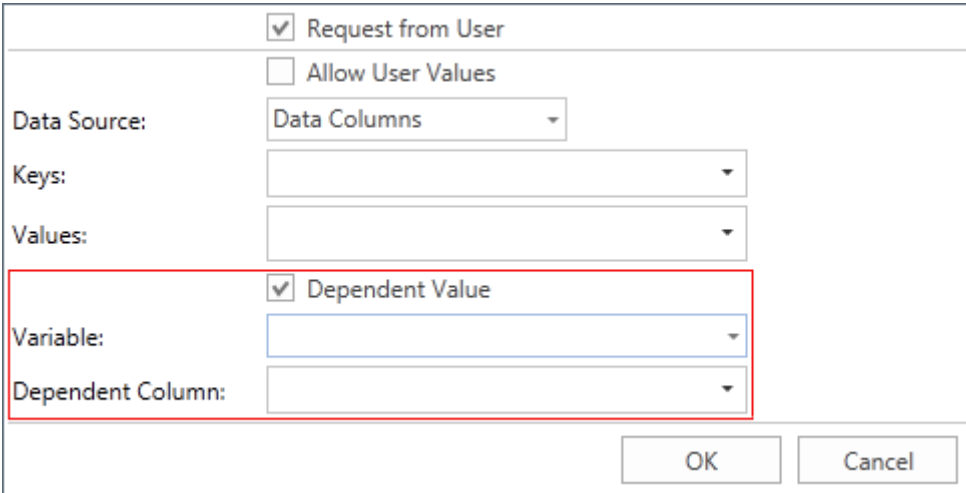
Как уже говорилось выше, на данной панели расположены кнопки управления элементами. Рассмотрим более подробно эти команды. На рисунке снизу представлена панель управления:



- 1 Кнопка Новое значение (New Value) создает новый элемент типа значение;
- 2 Кнопка Новое выражение (New Expression) создает новый элемент типа выражение;
- 3 Кнопка Выбор колонок (Select Columns) вызывает диалоговое окно, в котором следует указать в качестве ключей и значений колонки данных;
- 4 Кнопка Удалить (Remove) удаляет выделенный элемент.
- 5 Кнопки перемещения. Предоставляют возможность перемещать выделенный элемент вверх-вниз на панели элементов.

9.3.2.1.2.2. ЗАВИСИМЫЕ ПЕРЕМЕННЫЕ

При создании отчетов с параметрами, можно использовать зависимые переменные. В этом случае, одна переменная будет независимая, а остальные могут подчиняться ей или представлять собой иерархию, т.е. каждая последующая переменная подчиняется предыдущей. Для того чтобы переменная стала зависимой необходимо установить флажок Зависимое значение (расположен на панели Запросить у пользователя, при выборе источника данных Колонки данных). После установки флажка, будут отображаться два поля: Переменная и Зависимая колонка. В первом поле следует выбрать переменную, которая будет главной, т.е. от которой данная переменная будет зависеть. Во втором поле следует указать колонку данных, по которой будет осуществляться связь с главной переменной.



The screenshot shows a dialog box with the following elements:

- Request from User
- Allow User Values
- Data Source: Data Columns (dropdown)
- Keys: (empty dropdown)
- Values: (empty dropdown)
- Dependent Value (highlighted in red)
- Variable: (empty dropdown, highlighted in red)
- Dependent Column: (empty dropdown, highlighted in red)
- OK button
- Cancel button

Данная возможность, зависимость между переменными, удобна при применении параметров в отчете, к примеру, в Master-Detail отчетах. Допустим, есть список категорий, в каждую категория входит несколько продуктов, а каждый продукт обладает детальной информацией. В этом случае, при использовании параметров в отчете, переменная по

продуктам и по детальным данным будет содержать огромный список значений (полностью весь перечень продуктов и весь список детальной информации), и если необходимо произвести выборку продукта или какой-то конкретной детальной информации, придется затратить не мало времени. Если взаимосвязь между переменными будет отсутствовать, то список значений категорий будет содержать 8 категорий, продуктов - 77 записей, а детальных данных до нескольких сотен. Отыскать необходимый продукт и уж тем более детальные данные в таком случае практически невозможно. На рисунках снизу приведены примеры списков значений без взаимосвязи между переменными:

Categories

- Beverages
- Condiments
- Confections
- Dairy Products
- Grains/Cereals
- Meat/Poultry
- Produce
- Seafood

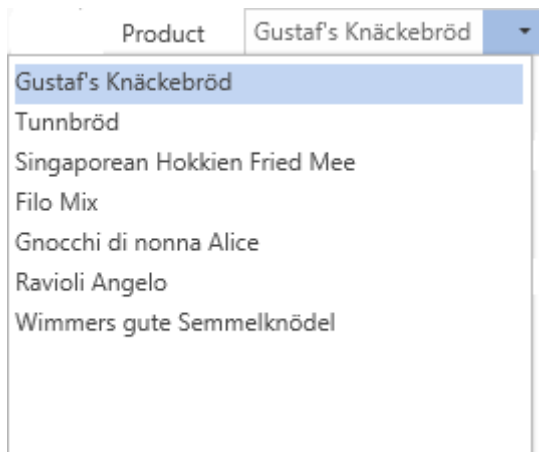
Product

- Chai
- Chang
- Aniseed Syrup
- Chef Anton's Cajun Seasoning
- Chef Anton's Gumbo Mix
- Grandma's Boysenberry Spread
- Uncle Bob's Organic Dried Pears
- Northwoods Cranberry Sauce
- Mishi Kobe Niku
- Ikura

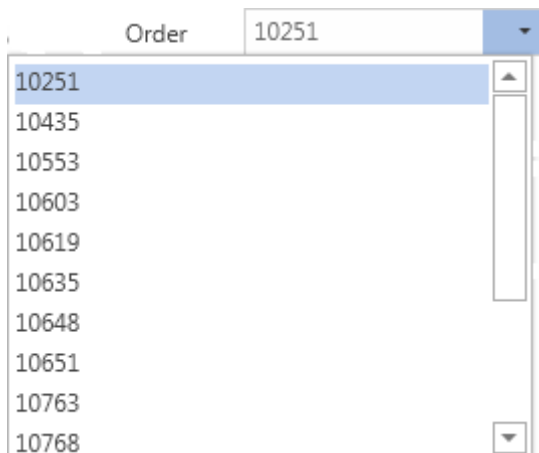
Order

- 11
- 14
- 41
- 22
- 20
- 31
- 24
- 2
- 53
- 27

Возможность зависимых переменных предоставляет возможность сократить списки значений переменных. Иными словами, между переменными можно организовать связь, что приведет к фильтрации списка значений в зависимости от значения главной переменной. Например, в зависимости от выбранной категории, сформируется список значений переменной по продуктам, а в зависимости от выбранного продукта сформируется список детальной информации. К примеру, будет выбрана категория Condiments, тогда список продуктов отфильтруется и будет вида:



Теперь выберем продукт Genen Shouyu, и тогда список детальных данных будет вида:

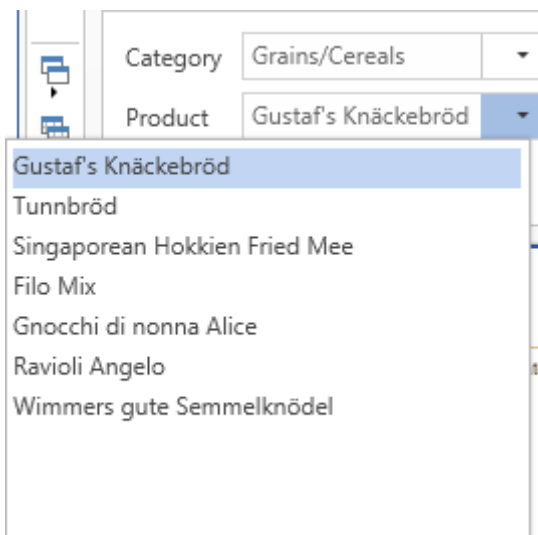


Рассмотрим создание и использование переменных в отчете. Создадим две переменных, одна из которых будет содержать список категорий, вторая список продуктов. Причем список продуктов будет зависеть от выбранной категории. Пример рассмотрим на источниках данных Demo.

Создадим переменные Category и Product, типа Значение с инициализацией данных integer. В главной переменной (Category), ключи выберем Categories.CategoryID, а значения Categories.CategoryName.

Примечание: Ключ - это уникальный идентификатор записи (строки) в источнике данных. В данном случае для категорий, колонкой содержащей ключи, будет CategoryID, а для продуктов - ProductID. Именно при помощи ключей организуется зависимость (связь) между источниками данных. Важно понимать, что разные ключи продуктов могут относиться к одному и тому же ключу категории.

В зависимой переменной ключи укажем Products.ProductID, а значения - Products.ProductName. Установим флажок Зависимое значение, выберем Category как главную переменную, а колонку данных Products.CategoryID, как зависимую колонку. Теперь перейдем на вкладку Предварительного просмотра, как видно на рисунке ниже, отображается два параметра. В первом списке выбирается категория, а второй список (продукты) формируется в зависимости от выбранной категории:



Как видно из рисунка, вторая переменная, которая представляет собой список значений, отображает не полностью весь список хранимых значений, а лишь те значения, которые относятся к выбранной категории.

► Добавим третью переменную в словарь данных. Переменная будет с именем All, типа Значение с инициализацией данных bool.

► Теперь используем зависимость переменных в отчете. Допустим есть Master-Detail отчет, где в каждой категории по несколько продуктов. Для того, чтобы при помощи параметров выбирать определенный продукт или продукты определенной категории, добавим на бэнды Данные в шаблоне отчета, фильтры с выражениями:

- ✓Первый фильтр будет на Master бэнде Данные, т.е. на бэнде при помощи которого формируется список категорий в отчете. Здесь необходимо отфильтровать категории, в зависимости от выбранного параметра отчета, поэтому выражение будет вида (Category == Categories.CategoryID).
- ✓Далее необходимо добавить второй фильтр на Detail бэнд Данные, т.е. на бэнд при помощи которого формируется список продуктов в отчете. Фильтр будет с выражением (Product == Products.ProductID).

►Осуществляем переход на вкладку предварительного просмотра. В параметрах отчета необходимо выбрать категорию, затем продукт, применить параметры в фильтрации данных отчета:

CategoryID	ProductID	Quantity per unit	Price	Units in stock
5	22	Gustaf's Knäckebröd	21,00p.	104,00

Count: 1

Как видно из рисунка, была выбрана категория Grains/Cereals (обратите внимание, ключ категории = 5) и продукт Gustaf's Knackebrod (ключ продукта = 22). Другими словами, отобразятся все категории с ключом 5 и все продукты с ключом 22.

►Для того чтобы в отчете отобразить полный список продуктов относящихся к выбранной категории, необходимо использовать третью переменную (All). Поэтому следует изменить выражение фильтра на бэнде Данные, при помощи которого формируется список продуктов (Product == Products.ProductID || All). В этом случае, в зависимости от значения третьей переменной (установлен флажок или не установлен), будет осуществляться фильтрация. Если флажок не установлен, то фильтрация будет по ключам продуктов, т.е. в отчете отобразится тот продукт, чей ключ совпадет. Если же флажок будет установлен, то отображаться будут все продукты выбранной категории:

Category: Grains/Cereals

Product: Gustaf's Knäckebröd

All:

Reset Submit

Grains/Cereals
Breads, crackers, pasta, and cereal

Products

CategoryID	ProductID	Quantity per unit	Price	Units in stock
5	52	16 - 2 kg boxes	7,00p.	38,00
5	56	24 - 250 g pkgs.	38,00p.	21,00
5	22	24 - 500 g pkgs.	21,00p.	104,00
5	57	24 - 250 g pkgs.	19,50p.	36,00
5	42	32 - 1 kg pkgs.	14,00p.	26,00 ✓
5	23	12 - 250 g pkgs.	9,00p.	61,00
5	64	20 bags x 4 pieces	33,25p.	22,00

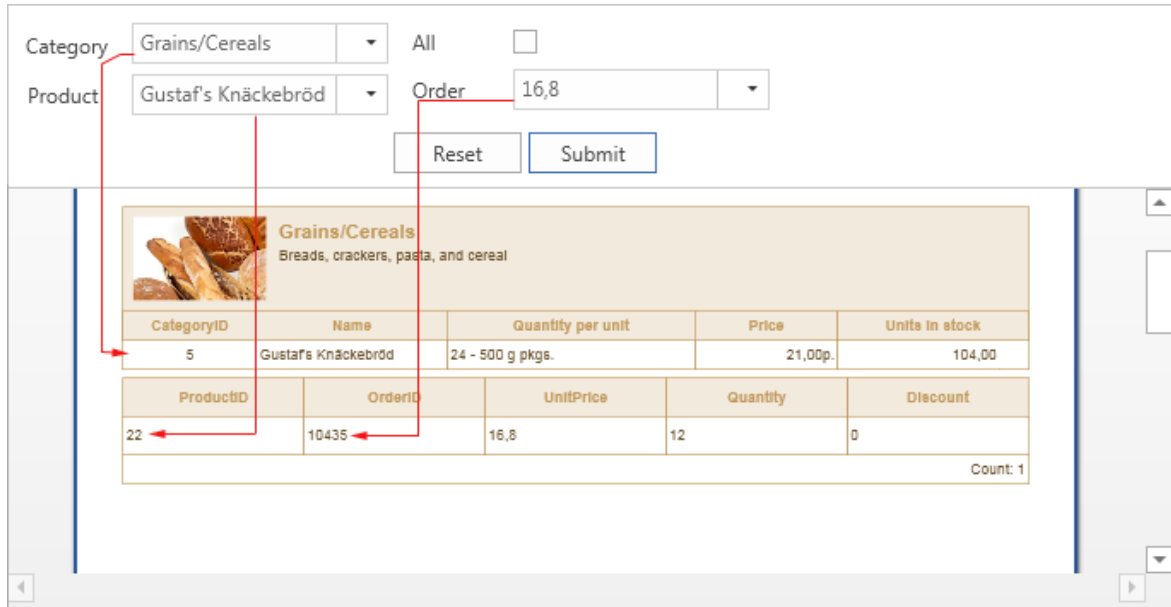
One Category

Как видно из рисунка, отображена одна категория (ключ = 5) и все продукты относящиеся к ней, с разными ключами.

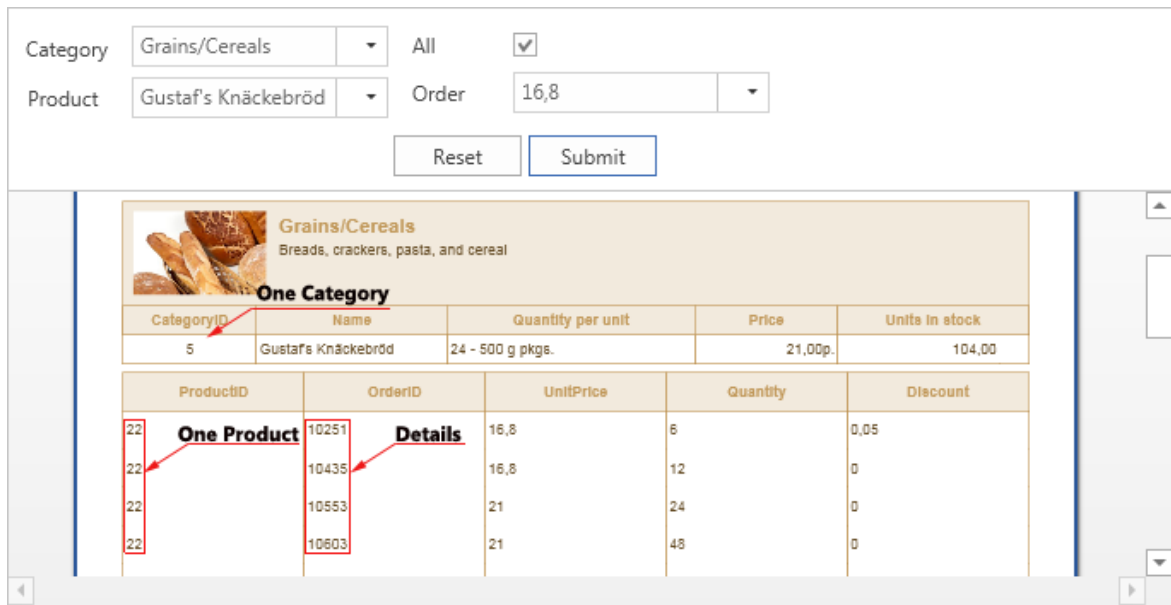
Пример, который был рассмотрен выше, представляет собой одноуровневую зависимость. Теперь рассмотрим более сложный пример, из одноуровневой сделаем двухуровневую зависимость. Оставим категории, продукты относящиеся к ним, и добавим детальные данные уже по каждому продукту. Для этого создаем переменную Order, типа Значение с инициализацией данных integer. Далее, установим флажок Запросить у пользователя, выбираем источник данных как колонка данных.

- ✓ Колонкой с ключами OrderDetails.OrderID, со значениями - OrderDetails.UnitPrice.
- ✓ Далее необходимо установить взаимосвязь с продуктами, для этого в качестве главной переменной следует выбрать Product, зависимой колонкой будет OrderDetails.ProductID.
- ✓ Теперь в шаблон отчета, следует добавить бэнд Данные с детальной информацией по продуктам. В данном примере, у бэнда Данные следует выбрать Order Details как источник данных. Master компонентом будет бэнд Данные с продуктами, и также необходимо указать связь между источниками данных.
- ✓ Добавить фильтр с выражением Order == Order_Details.OrderID на бэнд Данные, который содержит детальные данные по продуктам.
- ✓ Осуществляем переход на вкладку предварительного просмотра.

В отчете выбираем категорию, после чего фильтруется список продуктов. Выбираем продукт, после чего фильтруется список детальных данных выбранного продукта. Выбираем детальное значение, нажмем кнопку Применить:



Если необходимо, чтобы можно было отображать всю детальную информацию по выбранному продукту, то следует изменить выражение фильтрации на бэнде Данные с детальными данными продуктов. Выражение будет с использованием Variable3 и примет вид `Order == Order_Details.OrderID || All`. Теперь, можно просто указать категорию, выбрать продукт и получить все детальные данные по нему:



Как видно из рисунка отобразилась одна категория, один продукт и все детали поэтому продукту. Стоит также отметить, что количество уровней вложенности так же не ограничен.

9.3.2.1.3. ВИД ЗНАЧЕНИЕ

Переменная первого вида предоставляет возможность разместить какое-либо простое значение любого доступного типа данных или выражение. Рассмотрим пример создания переменной данного вида. Для этого вызовем команду Новая переменная...(New Variable...) после чего будет открыто диалоговое окно, в котором следует определить параметры переменной. Вид переменной Значение (Value) используется по умолчанию. На рисунке снизу представлено окно Новая переменная (New Variable):

Диалоговое окно "Новая переменная" (New Variable) с полями для ввода:

- Наименование: Переменная1
- Псевдоним: Переменная1
- Описание:
- Тип: int (выпадающий список), Значение (выпадающий список)
- Инициализировать: 1 Значение (выпадающий список)
- Значение: 2 (текстовое поле)
- Пример: 123; My text; 567f; 456.23f; Test String; A
- Чекбоксы: Только чтение, Запросить у пользователя, Разрешить использовать как OAD параметр
- Кнопки: OK, Отмена

1 Поле Инициализировать (Init by) содержит меню с выпадающим списком. В зависимости от выбранного пункта данного меню определяется тип значения в переменной: Значение (Value) или Выражение (Expression), т.е. выбирается способ инициализации переменной как значение или как выражение. В этом примере, переменная инициализирована как Значение (Value).

2 В данном поле указывается значение, которое будет храниться в переменной. Следует учитывать, что данное поле может отсутствовать. Если, к примеру, в поле Инициализировать (Init by) выбран пункт Выражение (Expression), то данное поле отсутствует, а вместо него присутствует поле Выражение (Expression). В этом случае, в поле Выражение (Expression) следует указать выражение, которое будет храниться в переменной. В данном примере, в переменной размещено значение 2.

После нажатия кнопки **Ok** будет создана переменная с именем **Variable1**. Рассмотрим на примере использование переменной вида **Значение (Value)** в отчете. Допустим есть отчет, в котором содержится информация о сотрудниках. На рисунке снизу представлен отчет с информацией о сотрудниках:

EmployeeID	LastName	City	Country
1	Davolio	Seattle	USA
2	Fuller	Tacoma	USA
3	Leverling	Kirkland	USA
4	Peacock	Redmond	USA
5	Buchanan	London	UK
6	Suyama	London	UK
7	King	London	UK
8	Callahan	Seattle	USA
9	Dodsworth	London	UK

Добавим фильтр с выражением **Employees.EmployeeID == UNN** на бэнд **Данные (DataBand)**. Теперь при построении отчета, будет отображена информация о тех сотрудниках, у которых **EmployeeID** равно значению хранимому в переменной. В данном примере, это **EmployeeID = 2**. На рисунке снизу представлен отчет с условием фильтрации:

EmployeeID	LastName	City	Country
2	Fuller	Tacoma	USA

9.3.2.1.4. ВИД ЗНАЧЕНИЕ С NULL

Переменная вида **Значение с null (Nullable Value)** предоставляет возможность размещать как простые значения, так и значения равные **null**. Если в отчете необходимо возвращать значение равное **null**, то при использовании переменной другого вида, произойдет ошибка компиляции отчета. На рисунке снизу представлено окно **Новая переменная (New Variable)** типа **Значение с null (Nullable Value)**:

Новая переменная

Наименование:

Псевдоним:

Описание:

Тип:

Инициализировать:

Значение:

Пример: 123; My text; 567f; 456.23f; Test String; A

Только чтение

Запросить у пользователя

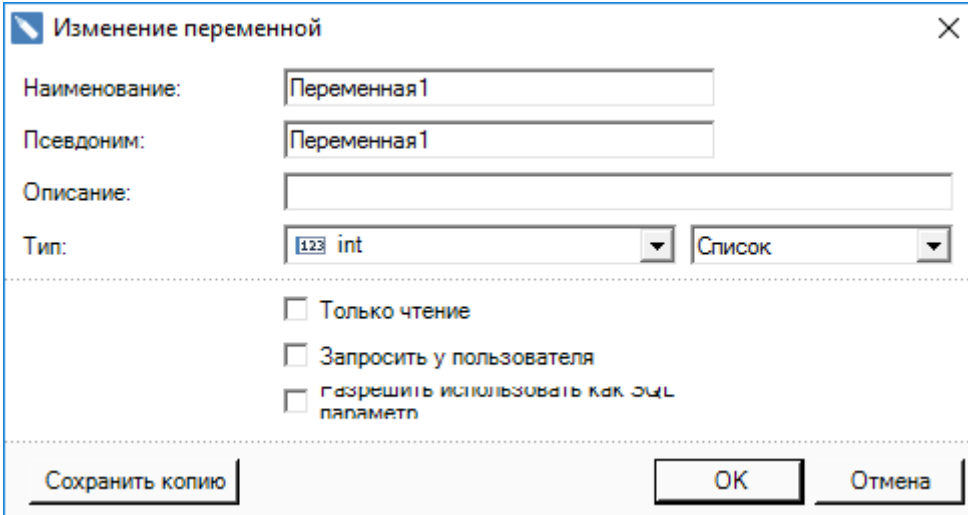
Разрешить использовать как ЭЦЛ параметр

1 Поле Инициализировать (Init by) содержит меню с выпадающим списком. В зависимости от выбранного пункта данного меню определяется тип значения в переменной: Значение (Value) или Выражение (Expression), т.е. выбирается способ инициализации переменной как значение или как выражение. В этом примере, переменная инициализирована как Значение (Value).

2 В данном поле указывается значение, которое будет храниться в переменной. Следует учитывать, что данное поле может отсутствовать. Если, к примеру, в поле Инициализировать (Init by) выбран пункт Выражение (Expression), то данное поле отсутствует, а вместо него присутствует поле Выражение (Expression). В этом случае, в поле Выражение (Expression) следует указать выражение, которое будет храниться в переменной. В данном примере, в переменной размещено значение с null.

9.3.2.1.5. ВИД СПИСОК

Переменная вида Список (List) предоставляет возможность разместить список значений любого доступного типа данных. В отличие, от переменной вида Значение (Value), в данном случае при построении отчета переменная содержит список значений. На рисунке снизу представлено окно переменной с выбранным типом Список (List):



Изменение переменной

Наименование:

Псевдоним:

Описание:

Тип:

Только чтение

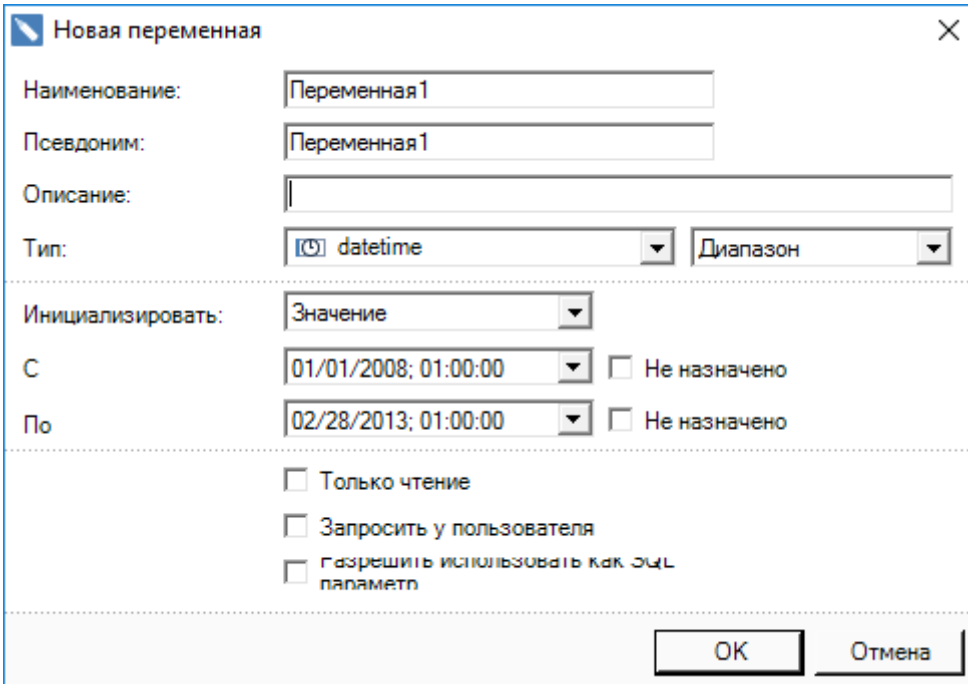
Запросить у пользователя

Разрешить использовать как ЭЦЛ параметр

Список можно сформировать вручную или добавлять значения в Событиях используя метод Add (например `Переменная.Add(10)`)

9.3.2.1.6. ВИД ДИАПАЗОН

При использовании переменной данного вида в отчете, можно работать с диапазонами значений. На рисунке снизу, представлено окно переменной вида Диапазон (Range):



Новая переменная

Наименование:

Псевдоним:

Описание:

Тип:

Инициализировать:

С: Не назначено

По: Не назначено

Только чтение

Запросить у пользователя

Разрешить использовать как ЭЦЛ параметр

Поле Инициализировать (Init by) содержит меню с выпадающим списком. В зависимости от выбранного пункта данного меню определяется тип переменной: Значение (Value) или Выражение (Expression), т.е. выбирается способ инициализации переменной как значение или как выражение. В этом примере, переменная инициализирована как Значение (Value).

В поле С (From) указывается начальное значение диапазона. Причем, указанное значение в данном поле включается в диапазон значений. В данном случае, указана дата 01/01/2008; 01:00:00.

В поле По (To) указывается конечное значение диапазона. Причем, указанное значение в данном поле включается в диапазон значений. В данном случае, указана дата 02/28/2013; 1:0:0.

После нажатия кнопки Ок, переменная будет создана. Рассмотрим пример использования переменной данного вида в отчете. Допустим есть отчет, в котором содержится информация о заказах: страна, имя доставки и дата. На рисунке снизу представлена страница отчета:

ShipCountry	ShipName	ShippedDate
Germany	Die Wandernde Kuh	12/15/2008 1:00:00 PM
Germany	Die Wandernde Kuh	5/23/2010 1:00:00 PM
Germany	Die Wandernde Kuh	10/23/2009 1:00:00 PM
Germany	Die Wandernde Kuh	11/2/2008 1:00:00 PM
Germany	Die Wandernde Kuh	5/28/2009 1:00:00 PM
Germany	Die Wandernde Kuh	9/18/2009 1:00:00 PM
Germany	Die Wandernde Kuh	9/27/2009 1:00:00 PM
Germany	Die Wandernde Kuh	10/17/2008 1:00:00 PM
Germany	Die Wandernde Kuh	12/27/2008 1:00:00 PM
Germany	Drachenblut Delikatessen	6/4/2010 1:00:00 PM

Если необходимо отобразить информацию о заказах, которые совершались в 2008-2013 годах, то используем созданную переменную в отчете. Для этого, добавим фильтр на бэнд Данные (DataBand) с выражением `Orders.ShippedDate > Variable1.FromDate && Orders.ShippedDate < Variable1.ToDate`. При построении отчета, отобразится только та информация о заказах, которые были осуществлены в 2008-2013 годах. На рисунке снизу представлен отчет с заказами в 2008-2013 годах:

ShipCountry	ShipName	ShippedDate
Germany	Die Wandernde Kuh	12/15/2008 1:00:00 PM
Germany	Die Wandernde Kuh	11/2/2008 1:00:00 PM
Germany	Die Wandernde Kuh	10/17/2008 1:00:00 PM
Germany	Die Wandernde Kuh	12/27/2008 1:00:00 PM

Стоит отметить, что при обращении к начальному/конечному значению диапазона, если используется тип данных DateTime, то для избежания дополнительных преобразований, можно обращаться VariableName.FromDate (или VariableName.FromTime если используется тип данных TimeSpan) и VariableName.ToDate (или VariableName.ToTime если используется тип данных TimeSpan).

9.3.2.1.7. ТРИ РЕЖИМА РАБОТЫ ПЕРЕМЕННОЙ

В зависимости от выбранных параметров переменная в отчете может работать в следующих режимах: автономный, пользовательский с выбором значений, пользовательский с вводом значениями. Рассмотрим более подробно эти режимы работы.

Автономный

Данный режим переменной будет применяться, если параметр Запросить у пользователя (Request from User) выключен, т.е. при использовании переменной в отчете, от пользователя не будет требоваться каких-либо действий. Создадим переменную, в которой будет храниться значение 2 типа integer с именем UNN. Используем эту переменную в отчете. На рисунке снизу представлен пример построенного отчета:

CategoryID	CategoryName
1	Beverages
2	Condiments
3	Confections
4	Dairy Products
5	Grains/Cereals
6	Meat/Poultry
7	Produce
8	Seafood

Допустим, добавим фильтр на бэнд Данные (DataBand), где в качестве условия фильтрации, укажем выражение Categories.CategoryID == UNN. Теперь при построении отчета, генератор отчетов учтет условия фильтрации, и отобразит только те записи, у которых значения в колонке CategoryID будут равны значениям хранимым в переменной. В данном случае, запись Condiments. На рисунке снизу представлен пример отчета с использованием переменной в фильтрации данных:

CategoryID	CategoryName
2	Condiments

В данном случае, при построении отчета от пользователя не потребуется каких-либо действий.

Пользовательский с выбором значений

Этот режим работы переменной будет применяться, если параметр Запросить у пользователя (Request from User) включен, а параметр Разрешить пользовательские значения (Allow Users Values) выключен. При использовании данной переменной в отчете, от пользователя могут потребоваться действия по выбору значений из списка переменной. Создадим переменную с именем UNN, в которой будет храниться список элементов от 1 до 8. Используем эту переменную в отчете. На рисунке снизу представлен пример построенного отчета:

CategoryID	CategoryName
1	Beverages
2	Condiments
3	Confections
4	Dairy Products
5	Grains/Cereals
6	Meat/Poultry
7	Produce
8	Seafood

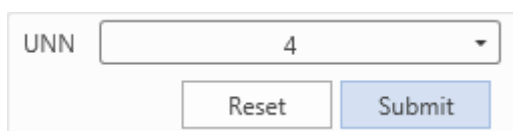
Допустим, добавим фильтр на бэнд Данные (DataBand), где в качестве условия фильтрации, укажем выражение Categories.CategoryID == UNN. Теперь при построении отчета, в окне вьюера необходимо будет выбрать значение из списка. На рисунке снизу представлен список значений переменной:



UNN 1, 2, 3, 4, 5, 6, 7, 8

- ✓ [1]
- ✓ [2]
- ✓ [3]
- ✓ [4]
- ✓ [5]
- ✓ [6]
- ✓ [7]
- ✓ [8]

После выбора значения, следует нажать кнопку Применить (Submit), для применения выбранного значения или кнопку Сбросить (Reset), для сброса на начальное значение в списке. На рисунке снизу представлена панель переменной в отчете:



UNN 4

Reset Submit

После нажатия кнопки Применить (Submit), генератор отчетов отфильтрует данные, и отобразит те данные, у которых CategoryID равен выбранному значению. На рисунке снизу представлен пример отчета с выбранным значением 4:

CategoryID	CategoryName
4	Dairy Products

Кроме кнопки Применить (Submit), также есть кнопка Сбросить (Reset), которая сбрасывает текущее значение и устанавливает первое сверху значение из списка переменной.

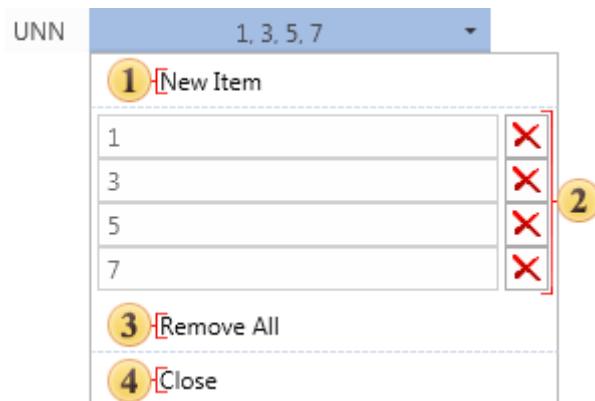
Пользовательский с вводом значений

Данный режим работы переменной будет применяться, если параметр Запросить у пользователя (Request from User) и Разрешить пользовательские значения (Allow Users Values) включены. При использовании в данном режиме переменной в отчете, от

пользователя потребуется выбор или ввод значения в поле переменной. Создадим переменную вида Список (List) с именем UNN, и укажем колонку CategoryID как ключи и значения. На рисунке снизу представлен пример построенного отчета:

CategoryID	CategoryName
1	Beverages
2	Condiments
3	Confections
4	Dairy Products
5	Grains/Cereals
6	Meat/Poultry
7	Produce
8	Seafood

Допустим, добавим фильтр на бэнд Данные (DataBand), где в качестве условия фильтрации, укажем выражение UNN.Contains(Categories.CategoryID). Теперь при построении отчета, в окне вывера необходимо отредактировать список значений переменной (удалить ненужные элементы, или изменить ключ в поле элемента, или создать новый элемент). На рисунке снизу представлен отредактированный список значений переменной:



- 1 Кнопка Новый элемент (New Item) создает новый элемент с полем, в котором следует указать ключ;
- 2 Кнопки Удалить (Remove) удаляет элемент, которому она принадлежит. У каждого элемента в списке есть данная кнопка.
- 3 Кнопка Удалить все (Remove All) удаляет все элементы из списка;
- 4 Кнопка Закрыть (Close) закрывает данное меню с сохранением элементов и введенных ключей.

После этого, следует нажать кнопку Применить (Submit). Теперь генератор отчетов отфильтрует данные, и отобразит данные, у которых CategoryID равен ключам указанным в полях элементов в списке значений переменной. На рисунке снизу представлен отфильтрованный отчет :

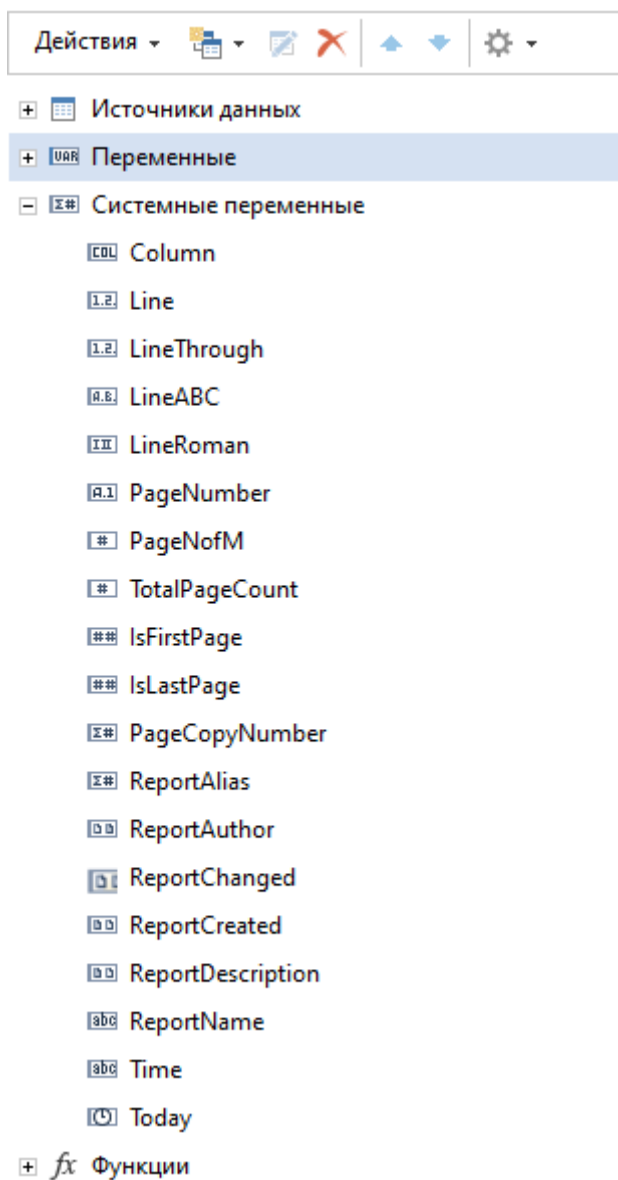
CategoryID	CategoryName
1	Beverages
3	Confections
5	Grains/Cereals
7	Produce

Кроме кнопки Применить (Submit), также есть кнопка Сбросить (Reset), которая в данном случае сбрасывает текущий список значений на первоначальный.

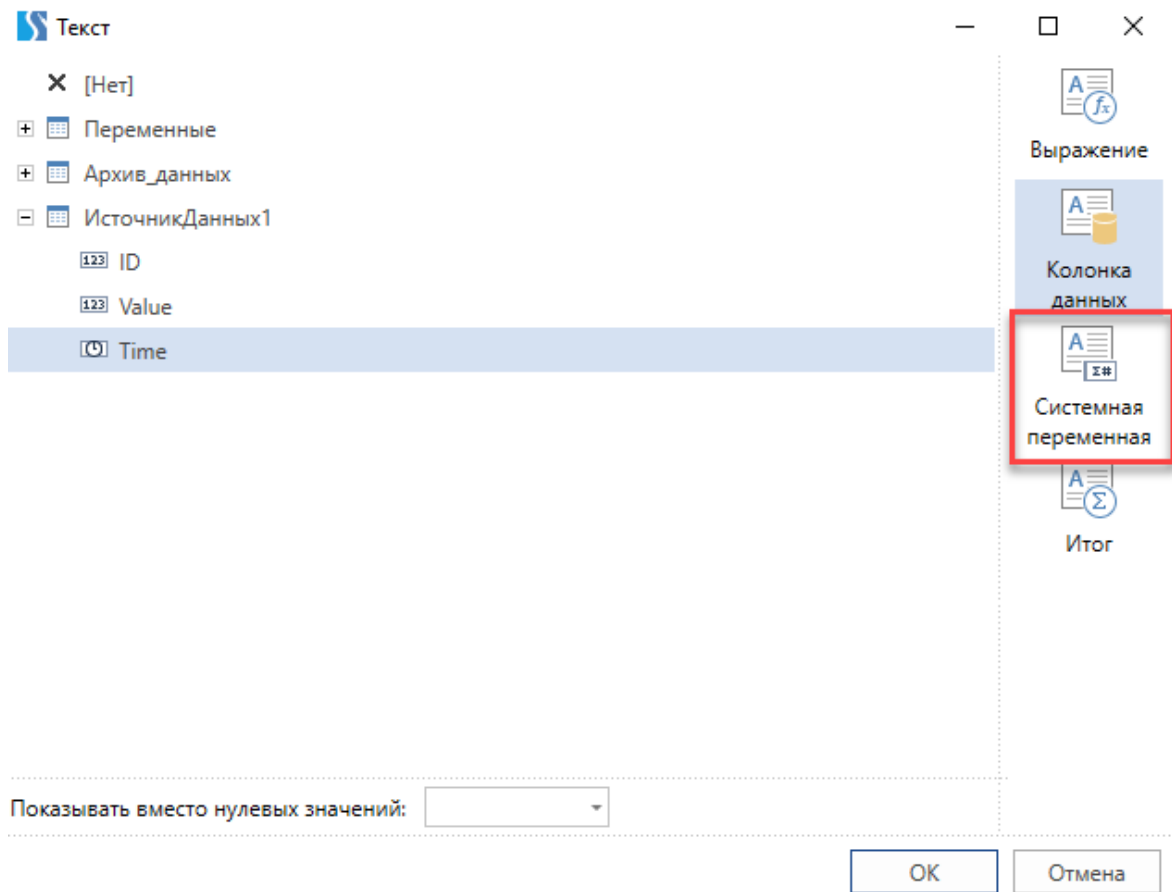
9.3.3. СИСТЕМНЫЕ ПЕРЕМЕННЫЕ

Редактор отчетов предлагает к использованию в выражениях системные переменные. Системные переменные - это переменные, которые сообщают, какую либо информацию о текущем состоянии отчета.

Список всех системных переменных отображается в словаре:



Также при настройке компонентов, где может потребоваться использование системных переменных будет возможность добавить ее, например, при редактировании компонента Текст:



Рассмотрим доступные системные переменные:

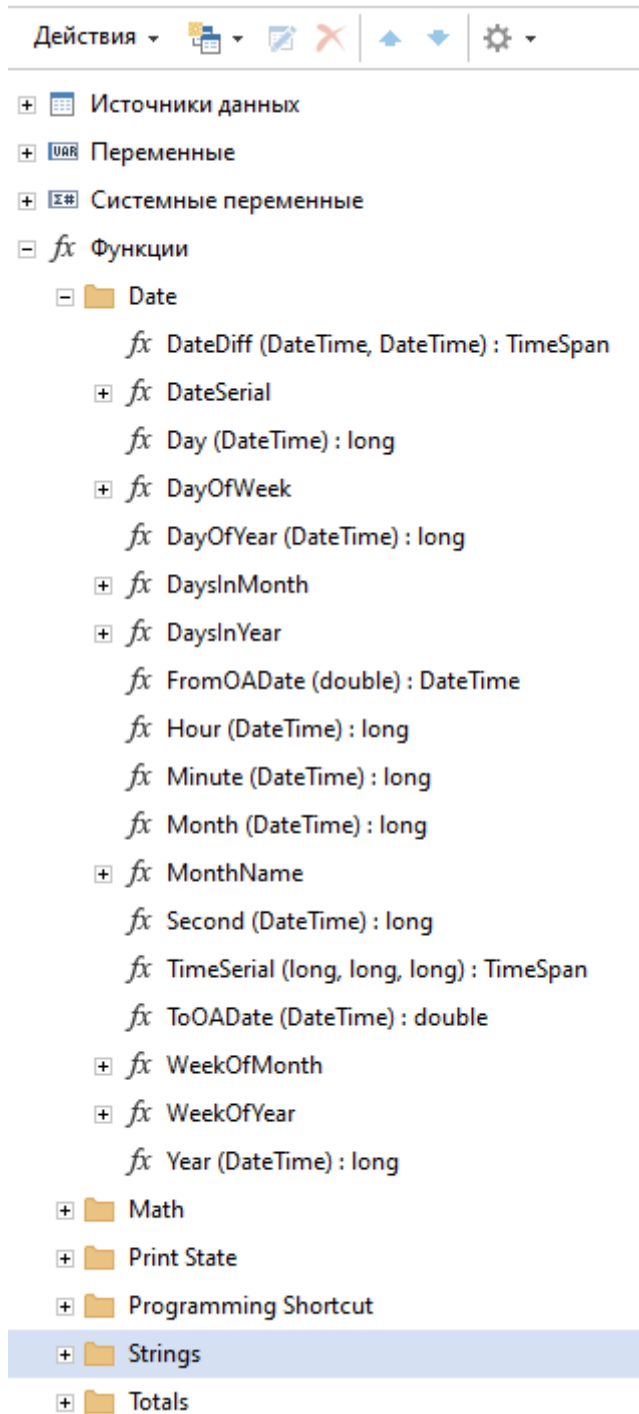
Название	Назначение
Column	Возвращает текущий номер колонки. Используется для нумерации колонок в отчете
Line	Возвращает текущий номер линии. Используется для нумерации строк в отчетах. Нумерация начинается с 1. Нумерация для каждой группы происходит отдельно
LineThrough	Возвращает сквозной номер линии. В отличие от Line возвращает номер линии с самого начала отчета, без учета группировок отчета. Нумерация начинается с 1
LineABC	Возвращает алфавитный аналог номера линии
LineRoman	Возвращает текущий номер линии группы. Начинается с 1

Название	Назначение
PageNumber	Возвращает текущий номер страницы. Нумерация страниц начинается с 1. Используется для нумерации страниц
PageNofM	<p>возвращает строку по следующему шаблону: Страница {PageNumber} из {TotalPageCount}</p> <p>Данная переменная представляет собой комбинирование системных переменных Номер страницы (PageNumber) и Общее количество страниц (TotalPageCount), т.е. будет выводить порядковый номер страницы в отношении общего количества страниц.</p>
TotalPageCount	Возвращает общее количество страниц в отчете
IsFirstPage	Возвращает значение истина (true), если в текущий момент следующая страница печатается;
sLastPage	Возвращает значение истина (true), если в текущий момент предыдущая страница печатается. Для корректного расчета этой переменной необходимо выполнить два прохода, т.е. отчет должен быть построен два раза. В первый раз отчет строится и запоминаются некоторые параметры, затем отчет строится второй раз, но уже с учетом этих параметров;
PageCopyNumber	Возвращает номер копии страницы. Начиная с 1.
Report Alias	Возвращает псевдоним отчета
Report Author	Возвращает автора отчета
Report Changed	Возвращает дату когда был последний раз изменен отчет
Report Created	Возвращает дату когда был создан отчет
Report Description	Возвращает описание отчета

Название	Назначение
Report Name	Возвращает текущее имя отчета. Имя отчета можно изменить при помощи свойства Имя отчета (Report Name);
Time	Возвращает текущее время
Today	Возвращает текущую дату

9.3.4. ФУНКЦИИ И РАСЧЕТ ИТОГОВ

В словаре данных представлена отдельная категория Функции. В этой категории представлены элементы, при помощи которых можно рассчитать определенный результат или вернуть необходимое значение. Все элементы категории Функции разделены по группам:



В таблице ниже представлен перечень функций, их описание и краткие примеры.

✖Посмотреть таблицу функций:

Функция	Описание	Пример
Date:		
{DateDiff(.)}	<p>Вычисляет интервал между указанными датами.</p> <ul style="list-style-type: none"> ▶ Аргументы должны быть типа DateTime. ▶ Возвращает значение TimeSpan 	<p>{DateDiff(DateSerial(2016,1,30),DateSerial(2016,1,1))} - результат, в этом случае, будет 29.00:00:00, т.е. 29 дней.</p> <p>{DateDiff(DataSource.Column1,DataSource.Column2)} - в этом случае, результат будет вычислен для каждого значения в Column1</p>
{DateSerial(.,.)}	<p>Предоставляет возможность указать дату.</p> <ul style="list-style-type: none"> ▶ В аргументах указывается год, месяц, день. ▶ Возвращает значение DateTime. 	<p>{DateSerial(2016,1,30)} - в этом случае, результат будет 1/30/2016 12:00:00 AM</p> <p>Функция возвращает значение DateTime, однако если необходимо отображение только даты, то следует применить к текстовому компоненту форматирование текста.</p>
{Day()}	<p>Предоставляет возможность отобразить день из указанной даты.</p> <ul style="list-style-type: none"> ▶ Аргументы должны быть типа DateTime. ▶ Возвращает значение long 	<p>{Day(DateSerial(2016,1,30))} - в этом случае, результат будет 30, т.к. в аргументах указано тридцатое января 2016 года.</p> <p>{Day(DataSource.Column)} - в этом случае, результат будет рассчитан для каждого значения Column</p>
{DayOfWeek()}	<p>Предоставляет возможность отобразить день недели из указанной даты в текстовом виде.</p> <ul style="list-style-type: none"> ▶ В аргументах указывается: <ul style="list-style-type: none"> 1 Дата (тип DateTime) 2 Культура (тип string). 3 Значение true or false (тип bool), для отображения результата с заглавной буквы или со строчной. 	<p>{DayOfWeek(DateSerial(2016,1,30))} - в этом случае, результат будет saturday.</p> <p>{DayOfWeek(DataSource.Column)} - в этом случае, для каждого значения будет рассчитан день недели</p> <p>{DayOfWeek(DateSerial(2016,1,30),"de")} - в этом случае, результат будет samstag, потому как установлена культура de.</p> <p>{DayOfWeek(DataSource.Column,"de")} - в этом случае, все результаты будут</p>

Функция	Описание	Пример
	<p>4 Значение true or false (тип bool), в зависимости от которого будет использоваться системная культура или локализация дизайнера</p> <p>▶ Возвращает значение string</p>	<p>соответствовать на культуре de</p> <p>{DayOfWeek(DateSerial(2016,1,30),"en", false)} - в этом случае, день недели будет со строчной буквы, т.е. saturday.</p> <p>{DayOfWeek(DataSource.Column,"de", true)} - в этом случае, дни недели будут с заглавной буквы, т.е. например Samstag.</p> <p>{DayOfWeek(DateSerial(2016,1,30), false)} - в этом случае, день недели будет отображен на культуре, которая используется системой.</p> <p>{DayOfWeek(DataSource.Column, true)} - в этом случае, день недели будет отображен на культуре, которая используется в дизайнера отчетов.</p>
{DayOfYear() }	<p>Предоставляет возможность отобразить день года</p> <p>▶ В аргументах указывается дата (тип DateTime)</p> <p>▶ Возвращает значение long</p>	<p>{DayOfYear(DateSerial(2016,2,14))} - в этом случае, результат будет 45, т.к. 14 февраля это 45-ый день года.</p> <p>{DayOfYear(DataSource.Column)} - в этом случае, для каждого значения Column будет рассчитан день года.</p>
{DaysInMonth() }	<p>Предоставляет возможность отобразить количество дней в месяце:</p> <p>▶ В аргументах указывается</p> <p>1 Дата (тип DateTime)</p> <p>2 Год и месяц (тип long)</p> <p>▶ Возвращает значение long</p>	<p>{DaysInMonth(DateSerial(2016,2,1))} - в этом случае, результат будет 29, т.к. 2016 год является високосным и в феврале 29 дней.</p> <p>{DaysInMonth(DataSource.Column)} - в этом случае, для каждого значения будет рассчитано количество дней в месяце.</p> <p>{DaysInMonth(2016,3)} - в этом случае, результат будет 31, поскольку в марте месяце 31 день.</p>
{DaysInYear() }	<p>Предоставляет возможность отобразить количество дней в году:</p>	<p>{DaysInYear(2016)} - в этом случае, результат будет 366 дней, поскольку 2016 год является високосным годом.</p>

Функция	Описание	Пример
	<ul style="list-style-type: none"> ▶ В аргументах указывается дата (тип DateTime) или год (тип long) ▶ Возвращает значение long 	{DaysInYear(DataSource.Column)} - в этом случае, для каждого значения Column будет рассчитано количество дней в году.
{Hour() }	<p>Предоставляет возможность отобразить час из времени:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается время (тип DateTime) ▶ Возвращает значение long 	{Hour(DataSource.Column)} - в этом случае, из каждого значения будет отображен час. Например, если время 16:22:36, то результатом будет 16.
{Minute() }	<p>Предоставляет возможность отобразить минуты из времени:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается время (тип DateTime) ▶ Возвращает значение long 	{Minute(DataSource.Column)} - в этом случае, из каждого значения будут отображены минуты. Например, если время 16:22:36, то результатом будет 22.
{Month() }	<p>Предоставляет возможность отобразить месяц из даты:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается время (тип DateTime) ▶ Возвращает значение long 	{Month(DateSerial(2016,12,1))} - в этом случае, результатом будет 12, поскольку установлена дата первое декабря 2016. {Month(DataSource.Column)} - в этом случае, из каждого значения Column будет отображен месяц.
{MonthName() }	<p>Предоставляет возможность отобразить название месяца из указанной даты</p> <ul style="list-style-type: none"> ▶ В аргументах указывается: <ul style="list-style-type: none"> 1 Дата (тип DateTime) и культура (тип string). 2 Культура (тип string), 3 Значение true or false (тип bool), для отображения результата с заглавной буквы или со строчной. 4 Значение true or false (тип bool), в зависимости от которого будет использоваться системная 	{MonthName(DateSerial(2016,1,1))} - в этом случае, результатом будет January, поскольку установлена дата первое января 2016 года. {MonthName(DataSource.Column)} - в этом случае, результатом будет название месяца для каждого значения Column {MonthName(DateSerial(2016,2,1),"de")} - в этом случае, результат будет соответствовать культуре de, т.е. Februar. {MonthName(DataSource.Column,"en")} - в этом случае, все названия месяцев

Функция	Описание	Пример
	<p>культура или локализация дизайнера</p> <ul style="list-style-type: none"> ▶ Возвращает значение string 	<p>будут соответствовать культуре en.</p> <p>{MonthName(DateSerial(2016,1,1), false)} - в этом случае, название месяца будет со строчной буквы.</p> <p>{MonthName(DataSource.Column, true)} - в этом случае, название месяцев будет с заглавной буквы.</p> <p>{MonthName(DateSerial(2016,1,1), false)} - в этом случае, название месяца будет соответствовать культуре, которая используется системой.</p> <p>{MonthName(DataSource.Column, true)} - в этом случае, названия месяцев будут соответствовать культуре, которая соответствует культуре дизайнера отчетов.</p>
{Second()}	<p>Предоставляет возможность отобразить секунды из времени:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается время (тип DateTime) ▶ Возвращает значение long 	<p>{Second(DataSource.Column)} - в этом случае, из каждого значения будут отображены секунды. Например, если время 16:22:36, то результатом будет 36.</p>
{TimeSerial(,,)}	<p>Предоставляет возможность отобразить время:</p> <ul style="list-style-type: none"> ▶ В аргументах указываются часы, минуты, секунды (тип long) ▶ Возвращает значение TimeSpan 	<p>{TimeSerial(1,14,20)} - в этом случае, результат будет 01:14: 20, т.е. один час, 14 минут, 20 секунд.</p>
{Year()}	<p>Предоставляет возможность отобразить год:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается дата (тип DateTime) ▶ Возвращает значение long 	<p>{Year(DateSerial(2016,1,2))} - в этом случае, результатом будет 2016, поскольку указана дата второе января 2016 года.</p> <p>{Year(DataSource.Column)} - в этом</p>

Функция	Описание	Пример
		случае, будут отображены года из каждого значения Column.
Math:		
{Abs()}	<p>Предоставляет возможность возратить абсолютно число.</p> <ul style="list-style-type: none"> ▶ В аргументах указывается число (тип double,decimal,long) ▶ Возвращает соответственно значение типа double,decimal,long 	<p>{Abs(-42)} - в этом случае, результат будет 42</p> <p>{Abs(DataSource.Column1)} - в этом случае, результатами будут абсолютные числа из значений Column1, т.е. без учет знака числа.</p>
{Acos()}	<p>Предоставляет возможность отобразить угол в радианах.</p> <ul style="list-style-type: none"> ▶ В аргументах значение cos (тип double) ▶ Возвращает значение угла в радианах типа double 	<p>{Acos(-1)} - в этом случае, будет вычислен угол в радианх для значения $\cos = -1$, т.е. угл будет равен ~ 3.14.</p> <p>{Acos(DataSource.Column1)} - в этом случае, для всех значений cos будет вычислен угол в радианах.</p>
{Asin()}	<p>Предоставляет возможность отобразить угол в радианах.</p> <ul style="list-style-type: none"> ▶ В аргументах значение sin (тип double) ▶ Возвращает значение угла в радианах типа double 	<p>{Asin(0)} - в этом случае, будет вычислен угол в радианх для значения $\sin = 0$, т.е. угл будет равен 0.</p> <p>{Asin(DataSource.Column1)} - в этом случае, для всех значений sin будет вычислен угол в радианах.</p>
{Atan()}	<p>Предоставляет возможность отобразить угол в радианах.</p> <ul style="list-style-type: none"> ▶ В аргументах значение tan (тип double) ▶ Возвращает значение угла в радианах типа double 	<p>{Atan(-1)} - в этом случае, будет вычислен угол в радианх для значения $\tan = -1$, т.е. угл будет равен ~ -0.79</p> <p>{Atan(DataSource.Column1)} - в этом случае, для всех значений tan будет вычислен угол в радианах.</p>
{Ceiling()}	Предоставляет возможность отобразить максимальное	<p>{Ceiling(25.124)} - в этом случае, результат будет 26. Стоит отметить, что при применении данной функции не происходит округление числа.</p>

Функция	Описание	Пример
	<p>целочисленное значение к указанному числу</p> <ul style="list-style-type: none"> ▶ В аргументах указывается значение (тип double, decimal) ▶ Возвращает значение типа double, decimal 	<p>{Ceiling(25.9)} - в этом случае, результат будет 26</p> <p>{Ceiling(DataSource.Column)} - в этом случае, для всех значений Column будут найдены и отображены ближайшие максимальные целые числа.</p>
{Cos() }	<p>Предоставляет возможность вычислить и отобразить значение cos:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается значение угла в радианах (тип double) ▶ Возвращает значение типа double, decimal 	<p>{Cos(0)} - в этом случае, результат будет 1.</p> <p>{Cos(DataSource.Column1)} - в этом случае, для всех значений будет вычислен cos угла.</p>
{Div() }	<p>Предоставляет возможность отобразить результат деления одного аргумента на другой:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается: <ul style="list-style-type: none"> 1 Делимое и делитель (тип double, decimal, long). 2 Делимое, делитель и значение, которое будет результатом, если делитель равен 0. ▶ Возвращает значение типа double, decimal, long 	<p>{Div(2,1)} - в этом случае, результат будет 2, т.к. $2 / 1 = 2$</p> <p>{Div(2,0,4)} - в этом случае, результат будет 4, т.к. делитель равен 0 и отображаться будет третий аргумент.</p> <p>{Div(DataSource.Column1,DataSource.Column2,DataSource.Column3)} - в этом случае, отображаться будет результат деления значений Column1 на значения Column2. При этом, если Column2 содержит нулевые значения, то вместо результата деления, в данной строке будут отображаться значения из Column3.</p>
{Exp() }	<p>Предоставляет возможность отобразить результат возведения в указанную степень числа e:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается степень, в которую необходимо возвести число e (тип long) 	<p>{Exp(4)} - в этом случае, число e будет возведено в 4-ую степень.</p> <p>{Exp(DataSource.Column1)} - в этом случае, каждое значение из Column1 будет представлять собой степень, в которую будет возводиться число e.</p>

Функция	Описание	Пример
	<ul style="list-style-type: none"> ▶ Возвращает значение типа double 	
{Floor()}	<p>Предоставляет возможность отобразить минимальное целочисленное значение к указанному числу</p> <ul style="list-style-type: none"> ▶ В аргументах указывается значение (тип double, decimal) ▶ Возвращает значение типа double, decimal 	<p>{Floor(123.59)} - в этом случае, результат будет 123, т.к. это ближайшее минимальное целое число. Стоит отметить, что данная функция не производит округление.</p> <p>Floor(101.99)} - в этом случае, результат будет 101</p> <p>{Floor(DataSource.Column1)} - в этом случае, для всех значений Column1 будут найдены и отображены ближайшие минимальные целые числа.</p>
{Log()}	<p>Предоставляет возможность вычислить натуральный логарифм:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается значение (тип double) ▶ Возвращает значение типа double 	<p>{Log(x)}, где x - это число или выражение, результатом будет вычисление натурального логарифма.</p>
{Maximum(,)}	<p>Предоставляет возможность сравнить два значения и отобразить максимальное:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается два значения (тип long, decimal, double) ▶ Возвращает значение типа long, decimal, double 	<p>{Maximum(5,9)} - в этом случае, результатом будет число 9.</p> <p>{Maximum(DataSource.Column1,DataSource.Column2)} - в этом случае, все значения Column1 сравнятся со значениями Column2. В отчете будут отображены максимальные числа.</p>
{Minimum(,)}	<p>Предоставляет возможность сравнить два значения и отобразить минимальное:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается два значения (тип long, decimal, double) 	<p>{Minimum(5,9)} - в этом случае, результатом будет число 5.</p> <p>{Minimum(DataSource.Column1,DataSource.Column2)} - в этом случае, все значения Column1 сравнятся со значениями Column2. В отчете будут</p>

Функция	Описание	Пример
	<ul style="list-style-type: none"> ▶ Возвращает значение типа long, decimal, double 	отображены минимальные числа.
{Round() }	<p>Предоставляет возможность округлить значение до целого числа или до определенного количества знаков дробной части:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается: <ul style="list-style-type: none"> 1 Значение (тип decimal, double), 2 Число знаков, до которого следует округлить дробную часть (тип int) ▶ Возвращает значение типа decimal, double 	<p>{Round(7.56)} - в этом случае, результат будет 8</p> <p>{Round(DataSource.Column1)} - в этом случае, все значения Column1 будут округлены по правилам математического округления.</p> <p>{Round(5.7896541897,3)} - в этом случае, результат будет 5.789</p> <p>{Round(DataSource.Column1,2)} - в этом случае, все значения из колонки данных будут округлены правилам математического округления до двух знаков в дробной части.</p>
{Sign() }	<p>Предоставляет возможность отобразить индикатор со знаком числа. Для положительных чисел 1, 0 - для всех нулевых значений, -1 - для отрицательных значений:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается значение (тип long, decimal, double). ▶ Возвращает значение типа long 	<p>{Sign(256)} - в этом случае, результат будет 1.</p> <p>{Sign(0)} - в этом случае, результат будет 0.</p> <p>{Sign(-157)} - в этом случае, результат будет -1.</p> <p>{Sign(DataSource.Column1)} - в этом случае, каждому значению из Column1, в зависимости от знака числа, будет присвоен индикатор.</p>
{Sin(0) }	<p>Предоставляет возможность вычислить sin угла:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается значение угла в радианах (тип double). ▶ Возвращает значение типа long 	<p>{Sin(0)} - в этом случае, результат будет 0.</p> <p>{Sin(DataSource.Column1)} - в этом случае, для всех значений будет вычислен sin угла.</p>

Функция	Описание	Пример
{Sqrt()}	<p>Предоставляет возможность вычислить квадратный корень из числа:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается число (тип double). ▶ Возвращает значение типа double 	<p>{Sqrt(4)} - в этом случае, результат будет 2, т.к. квадратный корень из 4 равен 2.</p> <p>{Sqrt(DataSource.Column1)} - в этом случае, для всех значений Column1 будет вычислен квадратный корень.</p>
{Tan()}	<p>Предоставляет возможность вычислить tg угла:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается значение угла в радианах (тип double). ▶ Возвращает значение типа long 	<p>{Tan(90)} - в этом случае, результат будет ~ -1.995</p> <p>{Tan(DataSource.Column1)} - в этом случае, для всех значений будет вычислен tan угла.</p>
{Truncate()}	<p>Предоставляет возможность отобразить только целую часть без округления:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается значение (тип double, decimal). ▶ Возвращает значение типа double, decimal 	<p>{Truncate(Sqrt(5))} - в этом случае, результат будет число 2, т.к. квадратный корень из 5 равен ~ 2.236, т.е. целая часть в этом числе и есть 2.</p> <p>{Truncate(DataSource.Column1)} - в этом случае, будет отображена только целая часть от всех значений Column1.</p>
Print State:		
{IsNull(,)}	<p>Предоставляет возможность выявить значения null в указанной колонке данных. Если будет присутствовать значение null, то результат будет true, инача - false.</p> <ul style="list-style-type: none"> ▶ В аргументах указывается: <ul style="list-style-type: none"> 1 Источник данных (тип object) 	<p>{Next(DataSource.Column)} - в этом случае, в построенном отчете, вместо значений null будет отображаться значение true, а вместо других значений - false.</p>

Функция	Описание	Пример
	<p>2 Имя колонки данных (тип string).</p> <p>▶ Возвращает значение типа bool</p>	
{Next(,)}	<p>Предоставляет возможность отобразить значение из следующей строки. Если значения следующей строки null, то результатом будет 0.</p> <p>▶ В аргументах указывается источник данных (тип object) и имя колонки данных (тип string).</p> <p>▶ Возвращает значение типа object</p>	<p>Например, колонка данных Column содержит значения 2, 5, 9. Тогда используя функцию {Next(DataSource,"Column")}, в результате первое значение будет 5, второе 9, третье - пусто.</p>
{NextIsNull(,)}	<p>Предоставляет возможность сравнить значение строки со значением следующей строки. Если значение следующей строки будет 0 или null, то результат будет true, иначе - false.</p> <p>▶ В аргументах указывается:</p> <p>1 Источник данных (тип object)</p> <p>2 Имя колонки данных (тип string).</p> <p>▶ Возвращает значение типа bool</p>	<p>Например, колонка данных Column содержит значения 2, 0, 9. Тогда используя функцию {NextIsNull(DataSource,"Column")}, в результате первое значение будет true, второе - false, третье - true.</p>
{Previous(,)}	<p>Предоставляет возможность отобразить значение из предыдущей строки. Если значения следующей строки null, то результатом будет 0.</p>	<p>Например, колонка данных Column содержит значения 2, 5, 9. Тогда используя функцию {Previous(DataSource,"Column")}, в</p>

Функция	Описание	Пример
	<p>▶ В аргументах указывается:</p> <ul style="list-style-type: none"> 1 Источник данных (тип object) 2 Имя колонки данных (тип string). <p>▶ Возвращает значение типа bool</p>	результате первое значение будет пусто, второе 2, третье - 5.
{PreviousIsNull(,) }	<p>Предоставляет возможность сравнить значение строки со значением предыдущей строки. Если значение предыдущей строки будет 0 или null, то результат будет true, иначе - false.</p> <p>▶ В аргументах указывается:</p> <ul style="list-style-type: none"> 1 Источник данных (тип object) 2 Имя колонки данных (тип string). <p>▶ Возвращает значение типа bool</p>	<p>Например, колонка данных Column содержит значения 2, 9, 0. Тогда используя функцию {PreviousIsNull(DataSource,"Column")}, в результате первое значение будет true, второе - false, третье - false.</p>
Programming Shortcut:		
{Choose() }	<p>Предоставляет возможность отобразить значение по индексу.</p> <p>▶ В аргументах указывается индекс и значения.</p> <p>▶ Возвращает значения по индексу</p>	<p>Все группы товаров сгруппированы по категориям: дорогие товары, товары средней цены, дешевые товары. Каждой группе присвоен индекс: дорогие - индекс 1, средние - индекс 2, дешевые - индекс 3 . В отчете следует отобразить вместо их индекса - категорию. В этом случае, можно воспользоваться функцией Choose.</p>

Функция	Описание	Пример
		<p>{Choose(DataSource.Column1, "expensive", "average", "cheap")} - вместо индекса 1 будет отображаться значение expensive, вместо индекса 2 - average, вместо индекса 3 - cheap.</p>
{IFF(,,)}	<p>Предоставляет возможность использовать отобразить то или иное значение, в зависимости от условия:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается условие, значение если условие выполнилось (true) и значение если условие не выполнилось (false) ▶ Возвращает значение типа object 	<p>В отчете со складскими запасами необходимо отслеживать количество товаров. Задача логиста состоит в том, что когда количество товаров приближается к нулю (меньше б), необходимо произвести заказ этих товаров. Для того чтобы выделять критические позиции в отчете визуально можно воспользоваться функцией {IFF(,,)}</p> <p>{IFF(DataSource.Column1 > б,"Minimum","Normal")}, где DataSource.Column1 - колонка со значениями количества товаров, б - предельное количество товаров, Minimum - значение, которое будет отображено если запас товаров меньше б, Normal - значение, которое будет отображено, если запас товаров б и больше.</p>
{Switch()}	<p>Предоставляет возможность присвоить заданное значение при выполнении определенного условия:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается условие и значение, которое будет присвоено, если условие выполнится. Таких пар условие-значение можно указываться несколько 	<p>Например, в отчете отображается список сотрудников, и необходимо отобразить их должность: Nancy - ведущий менеджер проекта, Andrew - главный разработчик, остальные сотрудники (их б человек) - Junior. В этом случае, функция Switch будет иметь три пары аргументов "условие-значение":</p> <p>{Switch(Employees.FirstName == "Nancy", "Manager", Employees.FirstName == "Andrew",</p>

Функция	Описание	Пример
	▶ Возвращает значение типа object	"Developer", Employees.FirstName != "", "Junior")}
Strings:		
{Arabic() }	<p>Предоставляет возможность конвертировать указанные цифры в цифры на арабском:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается значение типа string или int ▶ Возвращает значение типа string 	<p>{Arabic(2)} - в этом случае, цифра 2 будет иметь арабское написание.</p> <p>{Arabic(DataSource.Column1)} - в этом случае, все цифры из Column1 будут иметь арабское написание.</p>
{DateToStr() }	<p>Предоставляет возможность сконвертировать дату в текстовое значение:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается: <ul style="list-style-type: none"> 1 Дата (тип DateTime) 2 Булевские значения (true or false) для вывода отображения заголовка с заглавной буквы или со строчной. ▶ Возвращает значение типа string 	<p>{DateToStr(DataSource.Column1)} - в этом случае, все даты из Column1 будут отображаться в текстовом виде.</p> <p>{DateToStrPl(DataSource.Column1,true)} - в этом случае, даты будут отображаться в текстовом виде, на польском языке и первый символ с заглавной буквы.</p> <p>{DateToStrPl(DataSource.Column1,false)} - в этом случае, даты будут отображаться в текстовом виде, на польском языке и первый символ со строчной буквы.</p> <p>{DateToStrPtBr(DataSource.Column1)} - в этом случае, даты будут отображаться в текстовом виде на бразильском языке.</p>
{Insert(,)}	Предоставляет возможность вставить значение после	{Insert("25",2," dollars")} - в этом случае, в значение 25, после второго символа будет вставлено значение dollars, т.е. результат будет 25 dollars.

Функция	Описание	Пример
	<p>определенное символа в другое значение:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается: <ul style="list-style-type: none"> 1 значение , в которое будет производиться вставка текста (тип string), 2 номер символа, после которого будет вставлено значение (тип int), 3 значение для вставки (тип string) ▶ Возвращает значение типа string 	<p>{Insert(DataSource.Column1,2,DataSource.Column2)} - в этом случае, в значения Column1, после второго символа, будут вставлены значения Column2. Например, Column1 - содержит значение Category, Column2 - Products, тогда результат будет CaProductstegrity.</p>
{Left() }	<p>Предоставляет возможность отобразить указанное количество символов от левой стороны значения:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается значение типа string и количество символов, которое необходимо отобразить (тип int) ▶ Возвращает значение типа string 	<p>{Left("Beverages", 4)} - в этом случае, будет отображено только четыре символа из значения Beverages, т.е. результат будет Beve.</p> <p>{Left(DataSource.Column1, 2)} - в этом случае, будет отображено только первые два символа для каждого значения Column1.</p>
{Length() }	<p>Предоставляет возможность отобразить количество символов для указанного значения:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается значение (тип string) ▶ Возвращает значение типа int 	<p>{Length("Beverages") } - в этом случае, результат будет число 9, т.к. значение Beverages состоит из девяти символов.</p> <p>{Length(DataSource.Column1)} - в этом случае. для каждого значения Column1 будет рассчитано количество символов и этот результат будет отображен.</p>
{Mid() }	<p>Предоставляет возможность отобразить символы из значения. При этом можно задать позицию отсчета:</p>	<p>{Mid("Beverages",2,3)} - в этом случае, будет отображено три символа, после первых двух, т.е. результат будет ver.</p> <p>Mid(DataSource.Column1,3,2)} - в этом</p>

Функция	Описание	Пример
	<p>▶ В аргументах указывается:</p> <ul style="list-style-type: none"> 1 Значение (тип string) 2 Индекс позиции отсчета (тип int) 3 Количество символов, которое необходимо отобразить (тип int) <p>▶ Возвращает значение типа int</p>	случае, будет отображено 2 символа после первых трех для всех значений
{Persian()}	<p>Предоставляет возможность конвертировать указанные цифры в цифры на персидском:</p> <p>▶ В аргументах указывается значение типа string или int</p> <p>▶ Возвращает значение типа string</p>	<p>{Persian(5)} - в этом случае, цифра 2 будет иметь персидское написание.</p> <p>{Persian(DataSource.Column1)} - в этом случае, все цифры из Column1 будут иметь персидское написание.</p>
{Remove()}	<p>Предоставляет возможность удалить указанное количество символов с индекса определенной позиции:</p> <p>▶ В аргументах указывается:</p> <ul style="list-style-type: none"> 1 Значение (тип string) 2 Индекс позиции отсчета (тип int) 3 Количество символов, которое необходимо удалить (тип int) <p>▶ Возвращает значение типа int</p>	<p>{Remove("Beverages",2,3)} - в этом случае, после второго символа будет удалено три символа, т.е. результат будет Beages.</p> <p>{Remove(DataSource.Column1,3,2)} - в этом случае, для всех значений из Column1 будет удалено два символа после первых трех.</p>
{Replace(,,)}	<p>Предоставляет возможность заменить определенные символы или их сочетание на другие символы:</p> <p>▶ В аргументах указывается:</p> <ul style="list-style-type: none"> 1 Значение (тип string), в котором будет производиться замена 	<p>{Replace("Beverages","ver","NEW")} - в этом случае, в значении Beverages, символы ver будут заменены на символы NEW, т.е. результат будет BeNEWages</p> <p>{Replace(DataSource.Column1,"gex","sum")} - в этом случае, для значений Column1, в которых встречается сочетание символов gex,</p>

Функция	Описание	Пример
	<p>2 Символы, которые необходимо заменить (тип string)</p> <p>3 Символы, которые необходимо вставить (тип string)</p> <p>▶ Возвращает значение типа string</p>	<p>будет осуществлена замена gex на sum. В значениях, где отсутствует сочетание gex, замена производиться не будет.</p>
{Right()}	<p>Предоставляет возможность отобразить указанное количество символов от правой стороны значения:</p> <p>▶ В аргументах указывается значение типа string и количество символов, которое необходимо отобразить (тип int)</p> <p>▶ Возвращает значение типа string</p>	<p>{Right("Beverages",3)} - в этом случае, будет отображено три символа от правой стороны значения. т.е. ges.</p> <p>{Right(DataSource.Column1,4)} - в этом случае, для каждого значения Column1 будет отображено четыре символа от правой стороны.</p>
{Roman()}	<p>Предоставляет возможность сконвертировать арабские цифры в римские:</p> <p>▶ В аргументах указывается число (тип int)</p> <p>▶ Возвращает значение типа string</p>	<p>{Roman(4)} - в этом случае, цифра 4 будет иметь римское написание.</p> <p>{Roman(DataSource.Column1)} - в этом случае, все цифры из Column1 будут иметь римское написание.</p>
{Substring()}	<p>Предоставляет возможность отобразить определенное число символов с указанной позиции:</p> <p>▶ В аргументах указывается:</p> <p>1 Значение (тип string), из которого будут отображаться символы</p> <p>2 Индекс позиции (тип int), т.е. сколько символов пропускается</p> <p>3 Количество символов, которые необходимо отобразить (тип int)</p>	<p>{Substring("Beverages",6,3)} - в этом случае, первые шесть символов пропускаются и будет отображаться три символа, т.е. результат будет ges.</p> <p>{Substring("Beverages",0,2)} - в этом случае, будет отображаться два символа начиная с нулевого, т.е. результат будет Be</p> <p>{Substring(DataSource.Column1,1,4)} - в этом случае, пропускается первый символ и начиная со второго</p>

Функция	Описание	Пример
	<p>▶ Возвращает значение типа string</p>	<p>отсчитывается четыре. Это и будет результатом для каждого значения Column1, который отобразится в отчете.</p>
<p>{ToCurrencyWords(s)}</p>	<p>Предоставляет возможность отобразить валютную стоимость текстом:</p> <p>▶ В аргументах указывается:</p> <ol style="list-style-type: none"> 1 Числовое значение (тип double, decimal, long) 2 Также могут указываться значения (true or false) для отображения с заглавной буквы и отображения центов (тип bool) 3 Могут указывать форматы единственного числа и множественного для валюты и центов (тип string) 4 Есть возможность указать ISO код (тип string) 5 Также можно указать базовую единицу измерения для целой части и дробной <p>Кроме этого, возможно различные комбинации аргументов. Также есть виды этой функции, поддерживающие различные культуры.</p> <p>▶ Возвращает значение типа string</p>	<p>{ToCurrencyWords(100)} - по умолчанию, используемая валюта dollars of USA, поэтому результат будет One hundred dollars and zero cents.</p> <p>{ToCurrencyWords(Products.UnitPrice)} - в этом случае, все цены продуктов будут отображаться текстом.</p> <p>{ToCurrencyWords(100,true,false)} - в этом случае, результат будет отображаться с заглавным первым символом (поскольку установлено значение true) и без отображения центов (поскольку, установлен значение false), т.е. результат будет One hundred dollars.</p> <p>{ToCurrencyWordsEnGb(1.25,"EUR",2)} - в этом случае, применится ISO код EUR и результат будет one euro and twenty-five cents.</p> <p>{ToCurrencyWordsEnIn("dollars","cents",1.25M,0,true)} - в этом случается, указана базовая единица для целой части как dollars, для дробной части - cents, число для конвертации 1.25, затем указано колчество знаков после запятой, которое необходимо конвертировать и значение true означает, что запись будет с заглавной буквы.</p>

Функция	Описание	Пример
{ToLowerCase()}	<p>Предоставляет возможность отобразить значение в нижнем регистре:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается значение (тип string) ▶ Возвращает значение типа string 	<p>{ToLowerCase("EURO")} - в этом случае, результат будет euro.</p> <p>{ToLowerCase(DataSource.Column1)} - все значения этой колонки будут отображены в нижнем регистре.</p>
{ToOrdinal()}	<p>Предоставляет возможность преобразовать число в порядковое числительное:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается значение (тип long) ▶ Возвращает значение типа string 	<p>{ToOrdinal(25)} - - в этом случае, результат будет 25th.</p> <p>{ToOrdinal(DataSource.Column1)} - - в этом случае, все значения этой колонки будут преобразованы в порядковые числительные.</p>
{ToProperCase()}	<p>Предоставляет возможность преобразовать текст в формат первый символ заглавный, остальные в нижнем регистре:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается значение (тип string) ▶ Возвращает значение типа string 	<p>{ToProperCase("dOllars")} - - в этом случае, результат будет Dollars.</p> <p>{ToProperCase("dollars")} - - в этом случае, результат будет Dollars.</p> <p>{ToProperCase("dOLLARS")} - - в этом случае, результат будет Dollars.</p> <p>{ToProperCase(DataSource.Column1)} - - в этом случае, все значения из этой колонки будут с первым заглавным символом и остальными в нижнем регистре.</p>
{ToUpperCase()}	<p>Предоставляет возможность отобразить значение в верхнем регистре:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается значение (тип string) ▶ Возвращает значение типа string 	<p>{ToUpperCase("dollars")} - в этом случае, результат будет DOLLARS.</p> <p>{ToUpperCase("dOllars")} - в этом случае, результат будет DOLLARS.</p> <p>{ToUpperCase("dOLLARS")} - в этом случае, результат будет DOLLARS.</p> <p>{ToProperCase(DataSource.Column1)} - в этом случае, все значения будут написаны в верхнем регистре.</p>

Функция	Описание	Пример
{ToWords()}	<p>Предоставляет возможность отобразить число текстом:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается: <ul style="list-style-type: none"> 1 Числовое значение, которое будет преобразовано в текст (decimal, double, long) 2 Значение true or false для отображения с заглавное буквы для первого символа или нет 3 Значение true or false для возврата нулевых и пустых значений 4 Также возможен вариант указать true or false для предания женской формы для результата ▶ Возвращает значение типа string 	<p>{ToWords(100)} - в этом случае, результат будет one hundred. {ToWords(100, true)} - в этом случае, результат будет One hundred. {ToWordsEnIn(0,false)} - в этом случае, результат будет Zero. {ToWordsEnIn(0,true)} - в этом случае, результата не будет. {ToWordsEs(100,true,true)} - в этом случае, результат будет с заглавной буквы и в женской форме, т.е. cien. {ToProperCase(DataSource.Column1)} - в этом случае, все значения будут отображены текстом.</p>
{Trim()}	<p>Предоставляет возможность обрезать пробелы в начале или конце строки:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается значение (тип string) ▶ Возвращает значение типа string 	<p>{Trim(" <1 dollars> ")} - в этом случае результат в этом случае будет <1 dollars>". {Trim(DataSource.Column1)} - в этом случае, пробелы перед каждым значением и после каждого значения будут обрезаны.</p>
{TryParseDecimal() {TryParseDouble() {TryParseLong()}	<p>Предоставляют возможность проверить значение для конвертации в тип decimal, double,long :</p> <ul style="list-style-type: none"> ▶ В аргументах указывается значение (тип string) ▶ Возвращает значение типа bool. Если true - то конвертация будет успешной, иначе будет значение false. 	<p>{TryParseLong("100")} - в этом случае результат будет true, т.е. значение можно будет сконвертировать в long. {TryParseLong(" { 100")} - в этом случае, результат будет false, т.е. значение нельзя будет сконвертировать в long. {TryParseLong(DataSource.Column1)} - в этом случае, каждое значение будет проверено на возможность конвертации в тип long.</p>

Функция	Описание	Пример
Totals. Более подробно		

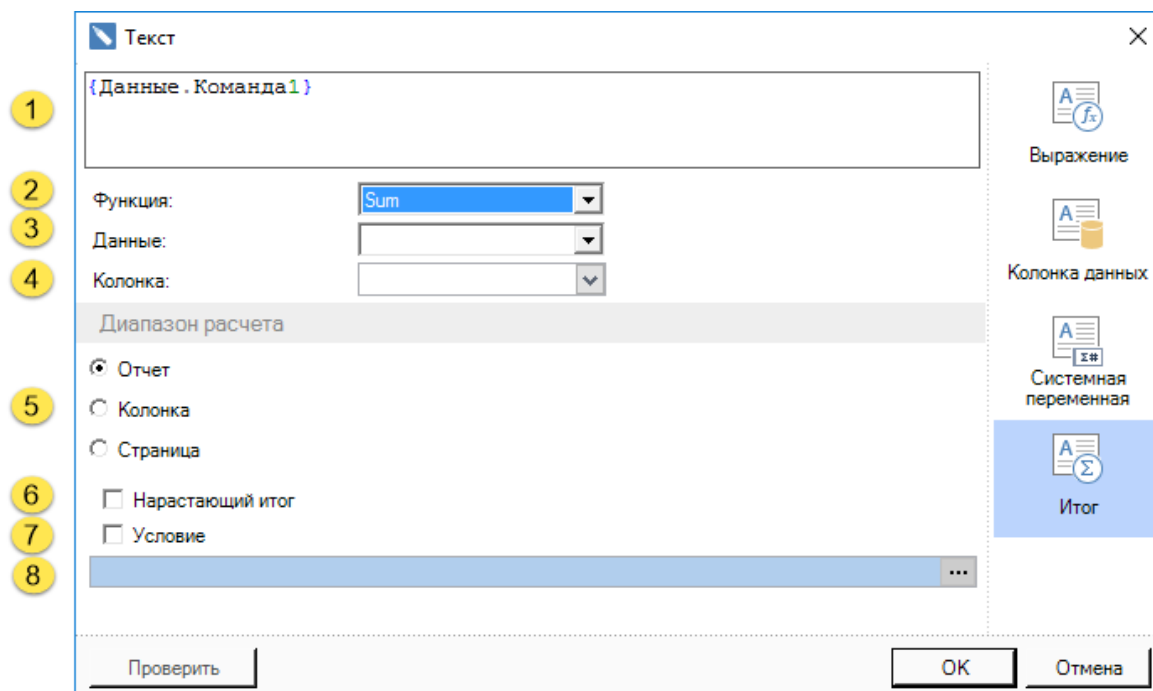
9.3.4.1. ИТОГИ

В отчете, кроме списка данных и их заголовка, как правило, присутствуют итоги. Это могут быть суммы, количество, минимальное, среднее, максимальное значение по определенному источнику, бэнду, странице. В зависимости от необходимого результата, следует выбирать тип функции итогов. Все функции итогов можно поделить условно на два типа: Связанные с бэндами. В этом случае, итоги рассчитываются во время построения отчета. Каждый раз, когда осуществляется одна операция с бэндом Данные (DataBand), производится расчет одного значения. Соответственно, текстовый компонент с итогом, следует располагать на каком-либо бэнде, который относится непосредственно к какому-либо бэнду Данные.

Не связанные с бэндами. В этом случае, расчет итогов не связан с операцией построения бэнда Данные. Следовательно, текстовый компонент с итоговой функцией может располагаться в любом месте отчета. Стоит отметить, что все функции имеют префикс Totals, т.е. формат записи {Totals.Functions()}.

9.3.4.1.1. ИТОГИ СВЯЗАННЫЕ С БЭНДАМИ

Для того чтобы рассчитать и отобразить итог, следует разместить текстовый компонент в отчете, вызвать редактор и перейти на вкладку Итог:



1 Поле выражения. В этом поле указывается выражение расчета итогов. Выражение можно указать вручную, либо оно будет сформировано автоматически, в зависимости от других параметров.

2 Поле Итоговая функция. В данном поле выбирается функция расчета итогов.

3 В данном поле определяется бэнд Данные, по которому будет рассчитан итог.

4 В данном поле определяется колонка данных, значения которой будут использоваться для расчета итога.

5 При помощи переключателей, устанавливается объект расчета итогов:

- ▶ Отчет, т.е. итог будет рассчитан по всему отчету.
- ▶ Колонка, т.е. итог будет рассчитан по каждой колонке в отчете.
- ▶ Страница, т.е. итог будет рассчитан по каждой странице построенного отчета.

6 Параметр расчета нарастающего итога. Если флажок установлен, то итог будет рассчитан как нарастающий. Если же флажок снят, то расчет итогов будет только по объекту (отчет, колонка, страница).

7 Параметр Условие. Если флажок установлен, то при расчете итогов будет учитываться условие. Если же флажок снят, то расчет итогов будет без учета условия.

8 Поле, в котором указывается выражение условия.

Тип результата итоговой функции

По умолчанию, функции для расчета итогов, возвращают значение типа Decimal (за исключением функций Count и CountDistinct). Однако, также можно производить расчеты с использованием и других двух типов данных - Double и Int64. Для того чтобы функция возвращала результат расчета, используя тип данных Double, добавьте латинскую букву D в верхнем регистре к названию функции. Для расчетов с использованием типа Int64 необходимо добавить латинскую букву I в верхнем регистре. Такое разделение позволит избежать потерь при расчетах итогов.

Функция	Тип возвращаемого значения
Sum()	Decimal
SumD()	Double
SumI()	Int64

⚠ОБРАТИТЕ ВНИМАНИЕ: Буквы I, D можно добавлять к любым функциям, за исключением двух: Count и CountDistinct. Эти функции всегда возвращают тип Int64.

Немного о синтаксисе функций на примере Sum и Count

При использовании языка программирования C#, все функции следует писать строго с соблюдением регистра.

- ▶ Sum(expression) - расчет суммы по автоматически определяемому объекту.
- ▶ Sum(band, expression) - расчет суммы по определенному объекту.
- ▶ SumIf(band, expression, condition) - расчет суммы по объекту с условием.

- ▶ expression – выражение для расчета;
- ▶ band – название бэнда для выполнения расчета;
- ▶ condition – условие включения выражения в расчет.

В случае с расчетами по Странице или контейнеру, синтаксис тот же за исключением добавления латинской буквы *s* как префикса к имени функции:

- ▶ `cSum(expression)` - расчет суммы по странице или контейнеру;
- ▶ `cSum(band, expression)` - расчет суммы по странице или контейнеру и объекту на ней;
- ▶ `cSumIf(band, expression, condition)` - расчет суммы по странице или контейнеру и объекту на ней, по определенному условию.

Для расчета итогов по колонке, к имени функции добавляется префикс `col`:

- ▶ `colSum(expression)` - расчет суммы по колонке;
- ▶ `colSum(band, expression)` - расчет суммы по колонке и объекту в ней;
- ▶ `colSumIf(band, expression, condition)` - расчет суммы по колонке и объекту в ней, по определенному условию.

Функция `Count` отличается от остальных функций тем, что она не имеет выражения для расчета. Синтаксис для этой функции приведен ниже.

- ▶ `Count()` - рассчитывает количество строк;
- ▶ `CountIf(condition)` - рассчитывает количество строк по условию;
- ▶ `Count(band)` - рассчитывает количество строк по объекту;
- ▶ `CountIf(band, condition)` - рассчитывает количество строк по объекту и условию;
- ▶ `cCount()` - рассчитывает количество строк по странице или контейнеру;
- ▶ `cCount(band)` - рассчитывает количество строк по странице (контейнеру) и объекту на ней;
- ▶ `cCountIf(band, condition)` - рассчитывает количество строк по странице (контейнеру) и объекту на ней, по определенному условию;
- ▶ `colCount()` - рассчитывает количество строк по колонке;
- ▶ `colCount(band)` - рассчитывает количество строк по колонке и определенному объекту в этой колонке;
- ▶ `colCountIf(band, condition)` - рассчитывает количество строк по колонке и определенному объекту в этой колонке, по определенному условию.

✖ Вывод итога в любом месте

Обычно компоненты, в текстовом выражении которых указан вызов функции, располагаются на итоговых бэндах или на бэнде Данные (Data). В редакторе отчетов есть несколько видов итоговых бэндов:

- ▶ ReportSummaryBand - бэнд используется для вывода итогов по всему отчету;
- ▶ PageFooterBand - бэнд используется для вывода итогов по странице;
- ▶ FooterBand - бэнд используется для вывода итогов по списку;
- ▶ GroupFooterBand - бэнд используется для вывода итогов по группе.
- ▶ ColumnFooterBand - бэнд используется для вывода итогов по колонке.

Расположение компонентов с функциями на любом из указанных выше бэндов позволяет генератору отчетов точно определить, к какому бэнду Данные относится эта функция. Также компонент с функций можно разместить и на самом бэнде Data. В этом случае на каждой строке данных будет выведен результат расчета функции по всем строкам.

Если же требуется вывести итог, например, на бэнде Заголовок данных (Header), то это выполняется при помощи скрипта. Однако в редактор отчетов компонент с функцией может быть расположен на любом бэнде отчета.

⚠ ОБРАТИТЕ ВНИМАНИЕ: Компоненты с функциями можно размещать в любом месте отчета.

Также допускается размещение компонента с функцией на странице и на других страницах шаблона отчета. К примеру, можно рассчитывать сумму значений по списку и вывести его в заголовке списка. Еще один пример, рассчитать количество строк в списке и вывести это значение в начале страницы. При этом, есть ограничение: необходимо указать бэнд Данные, по которому будет рассчитываться результат:

▶ {Sum(DataBand1, Products.UnitsInStock)}. В этом случае, будет рассчитана сумма значений колонки Products.UnitsInStock для каждой строки бэнда DataBand1.

▶ {Count(DataBand1)}. В этом случае будет, рассчитано количество строк бэнда DataBand1.

✖ Выражения с функциями

Для расчета итогов, в выражении можно не указывать дополнительные аргументы. Например, для функции Count необязательно или можно задать для функции Sum только один аргумент - выражение, которое должно быть рассчитано. Все это возможно, если генератор отчетов может определить, к какому бэнду Data относятся эти функции.

⚠ОБРАТИТЕ ВНИМАНИЕ: Генератор отчетов может определить взаимосвязь функции и конкретного бэнда Данные, если компонент с этой функцией располагается на связанных бэндах с этим бэндом Данные. Иначе говоря, компонент с функцией расположен на бэндах заголовков и итогов, которые относятся к этому бэнду Данные.

В противном случае, в аргументах следует указать источник данных или бэнд Данные, по которому необходимо рассчитать итог. В выражениях может быть указано:

- ▶ Просто объект, значения которого будут рассчитываться - {Sum(DataSource.Column)}
- ▶ Объект и различные математические операции с ним - {100 + Sum(DataSource.Column) * 2}

✖Расчет итогов по странице

Для того чтобы рассчитать итог по странице или панели следует к добавить латинскую букву с в нижнем регистре, как префикс к названию функции:

- ▶ {cCount(DataBand1)} - генератор отчетов рассчитает количество строк на одной странице или панели.

📌Примечание: Расчет итогов по странице идет по такому же принципу, как и для панели.

При расчете итогов по панели или по странице, желательно указывать бэнд Data, по которому производится расчет агрегатной функции. Это необходимо, потому что на одной странице может находиться более одного бэнда Data.

На одной странице или панели может использоваться любое количество агрегатных функций. MasterReport не налагает никаких ограничений на это. Допускается комбинирование итогов по странице с условием. К примеру:

▶ {cCountIf(DataBand1, Products.UnitsInStock = 0)} - генератор рассчитает количество элементов на этой странице равных нулю.

Расчет итогов по колонке

Для того чтобы рассчитать итог по колонке необходимо добавить префикс col (от слова column) в нижнем регистре к названию функции. К примеру:

▶ {colCount()} - генератор отчетов рассчитает количество строк по каждой колонке.

⚠ОБРАТИТЕ ВНИМАНИЕ: Расчет итогов по колонке в MasterReport имеет одно ограничение. Итоги можно рассчитывать только по колонкам на странице. Рассчитывать итоги по колонкам на бэнде Data не допускается.

При расчете итогов по колонке желательно, текстовый компонент с функцией размещать на бэндах ColumnHeader, ColumnFooter, Header или Footer. Можно рассчитывать неограниченное количество итогов по колонке. Каких-либо ограничений не накладывается. Также допускается комбинирование итогов по колонке с условием:

▶ {colCountIf(DataBand1, Products.UnitsInStock = 0)} - генератор отчетов рассчитает количество строк в каждой колонке, где условие выполниться.

Расчет итогов в коде событий

При помощи MasterReport можно произвести расчет функций в коде событий отчета. Это предоставляет возможность вычислять более сложные функции. Также при этом, можно обращаться к рассчитываемому значению из кода в процессе расчет и влиять на этот процесс. Для того чтобы произвести такой расчет необходимо создать переменную в словаре данных, которая будет хранить значение функции.

🕒Примечание: Не допускается использование переменных, объявленных в коде, для хранения результата расчета функций. Необходимо использовать переменные из словаря данных.

При создании переменной указывается тип данных переменной, к примеру, Decimal, и первоначальное значение, к примеру, 0. Затем следует, у бэнда Данные в событии Rendering указать выражение для приращения переменной. К примеру, если необходимо посчитать сумму значений по полю Products.UnitPrice, то выражение будет следующего вида:

▶Variable += Products.ItemsInStock;

Для того, чтобы отобразить результат расчета, следует в шаблоне отчета разместить текстовый компонент с выражением:

▶{Variable}

Также, необходимо у текстового компонента с выражением {Variable}, установить свойство Process At в значение End of Report. Это необходимо, чтобы генератор отчетов рассчитал значение переменной после обработки остальных компонентов.

✖Расчет итогов с условием

Иногда при расчете итогов, необходимо считать определенные значения. В этом случае, задается условие в функции расчета итогов. Например, необходимо просуммировать значения, которые больше нуля. Для добавления условия к функции расчета итогов, следует к названию функции добавить суффикс If латинскими буквами, и дополнительный аргумент с условием:

▶{SumIf(Products.UnitsInStock, Products.UnitsInStock > 1)}. В этом случае, будет рассчитана сумма значений Products.UnitsInStock, которые больше 1.

▶{CountIf(Products.UnitsInStock == 0)}. В этом случае, будет рассчитано количество строк, с нулевым значением в колонке UnitsInStock

🕒Примечание: Если необходимо произвести расчет с использованием типа Double или Int64, то сначала добавляем латинскую букву D или I, а затем слово If. К примеру: {SumDIf(Products.UnitsInStock, Products.UnitsInStock > 0)}.

Итоги и автоматическое изменение размеров компонента

⚠️ ОБРАТИТЕ ВНИМАНИЕ: При построении отчета, в тот момент, когда определяется размер компонента, результат расчета функции итога еще неизвестен. Это следует учитывать при установке автоматического изменения размеров для компонентов, в которых производится расчет итогов. В противном случае, может возникнуть ситуация, когда размер компонента не будет корректным по отношению к результату расчета функции итога.

Итоги по выключенным бэндам Данные

Бэнд Данные в отчете можно выключить различными способами. Например, он может быть выключенным по определенному условию или иметь нулевую высоту. По умолчанию, при построении отчета генератор отчетов не будет учитывать выключенные бэнды Данные и обрабатывать их. Однако, если необходимо по выключенному бэнду Данные произвести расчет итогов, то следует, у этого бэнда свойство Вычислять невидимые установить в значение true. В этом случае, в отчете будут выводиться только включенные бэнды Data, а расчет итогов выполниться с учетом выключенного бэнда Данные.

Расчет итогов в Master-Detail отчетах

При расчете итогов в иерархических отчетах есть особенности вычисления результата. Рассмотрим пример на основе Master-Detail отчета. Предположим в отчете выводится список продуктов по категориям. Категории, в данном случае, это master записи, а продукты - detail:

The sample demonstrates how to create a Master-Detail report. Date: July 2016



Beverages

Soft drinks, coffees, teas, beers, and ales

Name	Quantity per unit	Price	Units in stock
1 Chai	10 boxes x 20 bags	\$18.00	39.00
2 Chang	24 - 12 oz bottles	\$19.00	17.00
3 Chartreuse verte	750 cc per bottle	\$18.00	89.00
4 Côte de Blaye	12 - 75 cl bottles	\$283.50	17.00
5 Guaraná Fantástica	12 - 355 ml cans	\$4.50	20.00 ✓
6 Ipoh Coffee	16 - 500 g tins	\$46.00	17.00
7 Lakkalikööri	500 ml	\$18.00	57.00
8 Laughing Lumberjack Lager	24 - 12 oz bottles	\$14.00	52.00
9 Outback Lager	24 - 355 ml bottles	\$15.00	15.00
10 Rhönbräu Klosterbier	24 - 0.5 l bottles	\$7.75	125.00
11 Sasquatch Ale	24 - 12 oz bottles	\$14.00	111.00
12 Steeleye Stout	24 - 12 oz bottles	\$18.00	20.00



Condiments

Sweet and savory sauces, relishes, spreads, and seasonings

Допустим, необходимо посчитать количество продуктов, которое выводится в отчете. Если добавить бэнд Итог данных с функцией Count() к бэнду со списком продуктов (detail записи), то по каждой категории (master записи) будет произведен расчет итога:



Beverages

Soft drinks, coffees, teas, beers, and ales

Name	Quantity per unit	Price	Units in stock
1 Chai	10 boxes x 20 bags	\$18.00	39.00
2 Chang	24 - 12 oz bottles	\$19.00	17.00
3 Chartreuse verte	750 cc per bottle	\$18.00	89.00
4 Côte de Blaye	12 - 75 cl bottles	\$283.50	17.00
5 Guaraná Fantástica	12 - 355 ml cans	\$4.50	20.00 ✓
6 Ipoh Coffee	16 - 500 g tins	\$46.00	17.00
7 Lakkalikööri	500 ml	\$18.00	57.00
8 Laughing Lumberjack Lager	24 - 12 oz bottles	\$14.00	52.00
9 Outback Lager	24 - 355 ml bottles	\$15.00	15.00
10 Rhönbräu Klosterbier	24 - 0.5 l bottles	\$7.75	125.00
11 Sasquatch Ale	24 - 12 oz bottles	\$14.00	111.00
12 Steeleye Stout	24 - 12 oz bottles	\$18.00	20.00
			Count: 12

Если же добавить бэнд Итог данных с функцией Count() к бэнду с категориями, то результатом будет количество master записей в отчете, т.е. число категорий. Однако, в Master-Detail отчете можно рассчитать итоги и сразу для всех detail записей. В этом

случае, необходимо указать названия обоих (master и detail) бэндов в функции через двоеточие: Count(MasterBand: DetailBand).

The sample demonstrates how to create a Master-Detail report. Date: July 2016



Beverages
Soft drinks, coffees, teas, beers, and ales

Name	Quantity per unit	Price	Units in stock
1 Chai	10 boxes x 20 bags	\$18.00	39.00
2 Chang	24 - 12 oz bottles	\$19.00	17.00
3 Chartreuse verte	750 cc per bottle	\$18.00	89.00
4 Côte de Blaye	12 - 75 cl bottles	\$283.50	17.00
5 Guaraná Fantástica	12 - 355 ml cans	\$4.50	20.00 ✓
6 Ipoh Coffee	16 - 500 g tins	\$46.00	17.00
7 Lakkalikööri	500 ml	\$18.00	57.00
8 Laughing Lumberjack Lager	24 - 12 oz bottles	\$14.00	52.00
9 Outback Lager	24 - 355 ml bottles	\$15.00	15.00
10 Rhönbräu Klosterbier	24 - 0.5 l bottles	\$7.75	125.00
11 Sasquatch Ale	24 - 12 oz bottles	\$14.00	111.00
12 Steeleye Stout	24 - 12 oz bottles	\$18.00	20.00

77

Итогом функции Count(MasterBand:DetailBand) является число продуктов по всем категориям.

9.3.4.1.2. ИТОГИ НЕ СВЯЗАННЫЕ С БЭНДАМИ

Расчет итогов в отчете можно произвести указав выражение, к примеру, {Sum(DataBand1)}. При этом, данные итоги считаются во время построения отчета: каждый раз, когда осуществляется одна операция с бэндом Данные (DataBand), производится расчет одного значения. Затем, все рассчитанные значения суммируются и в результате получится итоговое значение, которое будет отображено. В этом случае, итоги связаны с бэндами. В MasterReport расчет итогов можно произвести другим способом - моментально. Иначе говоря, рассчитать итог не связанный с бэндами. Для этого следует использовать специальный префикс Totals перед функцией с разделителем ".", например, {Totals.Sum(DataBand1)}. Расчет функций с префиксом Totals происходит в месте вызова функции, в отличие от итогов связанных с бэндами, расчет которых выполняется в ходе построения отчета.

✖Функции Totals:

Функция	Описание	Пример
{Avg()}	<p>Предоставляет возможность высчитать среднее арифметическое значение:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается объект или два объекта ▶ Возвращает значения различного типа (double, decimal, long, DateTime, TimeSpan), в зависимости от выбранной функции. 	<p>{Avg(DataSource.Column1)} - в этом случае, будет вычислено среднее арифметическое по колонке Column1.</p> <p>{AvgDate(DataSource.ColumnDate)} - в этом случае, будет вычислено среднее значение даты по колонке ColumnDate.</p> <p>{AvgTime(DataSource.ColumnTime)} - в этом случае, будет вычислено среднее значение времени по колонке ColumnTime.</p> <p>Во всех функциях может быть указано два аргумента. Например, в отчете используется несколько бэндов Данные. Необходимо рассчитать среднее арифметическое по колонке для первого бэнда Данные. В этом случае, первым аргументом указывается бэнд, вторым объект для вычисления - {Avg(DataBand1,DataSource.Column2)}. В результате будет рассчитано среднее арифметическое по Column2, но только для значений DataBand1.</p>
{Count()}	<p>Предоставляет возможность рассчитать количество значений или количество уникальных значений:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается значение, объект или два объекта ▶ Возвращает значения различного типа long 	<p>{Count()} - в этом случае, результатом будет количество записей в источнике данных.</p> <p>{Count(DataBand1,DataSource.Column1)} - в этом случае, результатом будет количество записей в колонке Column1 для DataBand1.</p>

Функция	Описание	Пример
		<p>{CountDistinct(DataSource.Column1)} - в этом случае, результатом будет количество уникальных записей в DataSource.Column1.</p> <p>{CountDistinct(DataBand2, DataSource.Column2)} - в этом случае, результатом будет количество уникальных записей в колонке Column2 для DataBand2.</p>
{First()}	<p>Предоставляет возможность отобразить первое значение из указанного объекта:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается объект или два объекта ▶ Возвращает значения различного типа object 	<p>{First(DataSource1.Column1)} - в этом случае, результатом будет первое значение Column1 из источника DataSource1.</p> <p>{First(DataBand2, DataSource.Column2)} - в этом случае, результатом будет первое значение Column2 бэнда DataBand2.</p>
{Last()}	<p>Предоставляет возможность отобразить последнее значение из указанного объекта:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается объект или два объекта ▶ Возвращает значения различного типа object 	<p>{Last(DataSource1.Column1)} - в этом случае, результатом будет последнее значение Column1 из источника DataSource1.</p> <p>{Last(DataBand2, DataSource.Column2)} - в этом случае, результатом будет последнее значение Column2 бэнда DataBand2.</p>
{Max()}	<p>Предоставляет возможность отобразить максимальное значение из указанного объекта:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается объект или два объекта ▶ Возвращает значения различного 	<p>{Max(DataSource1.Column1)} - в этом случае, результатом будет максимальное значение из Column1 источника DataSource1.</p> <p>{MaxDate(DataSource1.ColumnDa</p>

Функция	Описание	Пример
	<p>типа double, decimal, long, DateTime, TimeSpan, string, в зависимости от выбранной функции.</p>	<p>te)} - в этом случае, результатом будет максимальная дата из ColumnDate источника DataSource1.</p> <p>{MaxTime(DataSource1.ColumnTime)} - в этом случае, результатом будет максимальное время из ColumnTime источника DataSource1.</p> <p>{MaxStr(DataSource1.Column1)} - в этом случае, все значения будут отсортированы в алфавитном порядке. Результатом будет последнее значение.</p> <p>{Max(DataBand2, DataSource.Column2)} - в этом случае, результатом будет максимальной значение Column2 бэнда DataBand2.</p>
<p>{Median() }</p>	<p>Предоставляет возможность отобразить среднее (не арифметическое) значение из списка:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается объект или два объекта ▶ Возвращает значения различного типа double, decimal, long, в зависимости от выбранной функции. 	<p>Допустим, Column1 содержит 5 значений: 2, 5, 6,1,7. Функция {Median(DataSource1.Column1)} отобразит среднее значения из этого списка, т.е. результатом будет значение 6.</p> <p>{Median(DataBand2, DataSource.Column2)} - в этом случае, результатом будет среднее значение Column2 бэнда DataBand2.</p>

Функция	Описание	Пример
{Min()}	<p>Предоставляет возможность отобразить максимальное значение из указанного объекта:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается объект или два объекта ▶ Возвращает значения различного типа double, decimal, long, DateTime, TimeSpan, string, в зависимости от выбранной функции. 	<p>{Min(DataSource1.Column1)} - в этом случае, результатом будет минимальное значение из Column1 источника DataSource1.</p> <p>{MinDate(DataSource1.ColumnDate)} - в этом случае, результатом будет минимальное дата из ColumnDate источника DataSource1.</p> <p>{MinTime(DataSource1.ColumnTime)} - в этом случае, результатом будет минимальное время из ColumnTime источника DataSource1.</p> <p>{MinStr(DataSource1.Column1)} - в этом случае, все значения будут отсортированы в алфавитном порядке. Результатом будет первое значение.</p> <p>{Min(DataBand2, DataSource.Column2)} - в этом случае, результатом будет минимальное значение Column2 бэнда DataBand2.</p>
{Mode()}	<p>Предоставляет возможность отобразить значение, которое наиболее часто встречается в списке значений:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается объект или два объекта ▶ Возвращает значения различного типа double, decimal, long, в зависимости от выбранной функции. 	<p>{Mode(DataSource1.Column1)}. Допустим Column1 содержит список значений: 2, 2, 6, 7, 7, 8, 7, 6, 5, 9, 4. В этом случае, результатом будет число 7, т.к. оно повторяется в списке значений наиболее часто.</p> <p>{Mode(DataBand2,</p>

Функция	Описание	Пример
		DataSource.Column2}} - в этом случае, результатом будет значение из Column2 бэнда DataBand2, которое наиболее часто встречается.
{Rank(,)}	<p>Предоставляется возможность отобразить ранг значения. Обязателен префикс Totals:</p> <p>▶ В аргументах указывается:</p> <ol style="list-style-type: none"> 1 Объекты для обработки и присвоения ранга (тип object) 2 Значение (true or false) для присвоения плотного или не плотного ранга 3 Направление сортировки значений. <ul style="list-style-type: none"> ▶ Возвращает значения различного типа long 	<p>{Totals.Rank(DataBand1,DataSource.Column1)}. Допустим, колонка Column1 содержит список значений: 44, 9, 36, 55, 71. В этом случае, значения будут отсортированы по возрастанию, т.е. 9, 36, 44, 55, 71 и каждому из них будет присвоен ранг. Число 9 получит ранг 1, 36 - ранг 2, 44 - ранг 3, 55 - ранг 4, 71 - ранг 5. По умолчанию, рассчитывает плотный ранг и сортировка значений для присвоения ранга выполняется по возрастанию</p> <p>{Totals.Rank(DataBand1,DataSource.Column1, true, StiRankOrder.Desc)} - в этом случае, будет плотный ранг поскольку установлено значение true. При присвоении ранга, значения будут отсортированы по убыванию, т.к. StiRankOrder установлено в Desc. Для сортировки по возрастанию (используется по умолчанию), следует установить в значение Asc (StiRankOrder.Asc).</p> <p>Пример неплотного ранга - {Totals.Rank(DataBand1,DataSource.Column1, false, StiRankOrder.Asc)}. Допустим Column1 содержит список</p>

Функция	Описание	Пример
		<p>значений: 44, 9, 44, 9, 31, 64,68, 71. Значения присвоены по возрастанию, т.е. 9, 9, 31, 44, 44, 44, 68, 71. В этом случае, ранги будут следующими: 9 - ранг 1, 9 - ранг 1, 31 - ранг 3, 44 - ранг 4, 44 - ранг 4, 44 - ранг 4, 68 - ранг 7, 71 - ранг 8. Иначе говоря при присвоении ранга числу, учитывается ранг предыдущего значения и количество значений с этим рангом.</p>
{Sum() }	<p>Предоставляет возможность отобразить результат суммы значений:</p> <ul style="list-style-type: none"> ▶ В аргументах указывается: 1 Объекты для обработки и присвоения ранга (тип object) 2 Условие 3 Выражение суммирования ▶ Возвращает значения различного типа long, decimal, double, TimeSpan 	<p>{Sum(DataSource1.Column1)} - в этом случае, результатом будет сумма всех значений Column1 в источнике DataSource1.</p> <p>{SumDistinct(DataSource1.Column1)} - в этом случае, результатом будет сумма всех уникальных значений Column1 в источнике DataSource1.</p> <p>SumTime(DataSource1.Column1) - в этом случае, результатом будет сумма времени из колонки Column1 в источнике DataSource1.</p> <p>{Sum(DataBand2,DataSource2.Column2)} - в этом случае, результатом будет сумма значений из Column2 бэнда DataBand2.</p>

Функция	Описание	Пример
		{SumDistinct(DataBand1,DataSource.Column1, DataSource.Column2)} - в этом случае, результатом будет сумма значений Column2, которые соответствуют уникальным значениям из Column1 бэнда DataBand2.

✖Рассмотрим расчет итогов не связанный с бэндами. О

Допустим, есть Master-Detail отчет, который представляет собой список продуктов по категориям:



Beverages Soft drinks, coffees, teas, beers, and ales			
Name	Quantity per unit	Price	Units in stock
1 Chai	10 boxes x 20 bags	\$18.00	39.00
2 Chang	24 - 12 oz bottles	\$19.00	17.00
3 Chartreuse verte	750 cc per bottle	\$18.00	69.00
4 Côte de Blaye	12 - 75 cl bottles	\$283.50	17.00
5 Guaraná Fantástica	12 - 355 ml cans	\$4.50	20.00
6 Ipoh Coffee	16 - 500 g tins	\$46.00	17.00
7 Lakkalikööri	500 ml	\$18.00	57.00
8 Laughing Lumberjack Lager	24 - 12 oz bottles	\$14.00	52.00
9 Outback Lager	24 - 355 ml bottles	\$15.00	15.00
10 Rhönbräu Klosterbier	24 - 0.5 l bottles	\$7.75	125.00
11 Sasquatch Ale	24 - 12 oz bottles	\$14.00	111.00
12 Steeleye Stout	24 - 12 oz bottles	\$18.00	20.00

В этом отчете, итог может быть рассчитан по каждой категории, по всему отчету. Также можно рассчитать долю каждой категории из общей суммы. Для начала рассчитаем сумму продуктов категории. Для этого, в шаблон отчета следует добавить бэнд Итог данных, разместить на нем текстовый компонент с выражением расчета итога {Sum(DataBand2,Products.UnitPrice)}. Для суммирования значений применяется функция Sum, в её аргументах указывается объект по которому будут считаться итоги и колонка данных, значения которой будут суммироваться. Поскольку, необходимо рассчитать сумму

продуктов по каждой категории, то объектом для расчета итогов будет детальный бэнд Данные, т.е. DataBand2. Значения в колонке UnitPrice обозначают стоимость каждого продукта, поэтому сумма этих значений и будет итог для категории:



Name	Quantity per unit	Price	Units in stock
1 Chai	10 boxes x 20 bags	\$18.00	39.00
2 Chang	24 - 12 oz bottles	\$19.00	17.00
3 Chartreuse verte	750 cc per bottle	\$18.00	69.00
4 Côte de Blaye	12 - 75 cl bottles	\$283.50	17.00
5 Guaraná Fantástica	12 - 355 ml cans	\$4.50	20.00
6 Ipoh Coffee	16 - 500 g tins	\$46.00	17.00
7 Lakkalikööri	500 ml	\$18.00	57.00
8 Laughing Lumberjack Lager	24 - 12 oz bottles	\$14.00	52.00
9 Outback Lager	24 - 355 ml bottles	\$15.00	15.00
10 Rhönbräu Klosterbier	24 - 0.5 l bottles	\$7.75	125.00
11 Sasquatch Ale	24 - 12 oz bottles	\$14.00	111.00
12 Steeleye Stout	24 - 12 oz bottles	\$18.00	20.00
Total by Beverages:			\$455.75

В этом случае, итог является связанным с бэндом Данные. Для расчета итога по отчету, в этом случае, следует использовать функции не связанные с бэндами. Для этого, к функции необходимо добавить префикс Totals, через разделитель ".". В качестве объекта, следует указать источник данных. Выражение расчета итога по отчету, в этом случае, будет `{Totals.Sum(Products,Products.UnitPrice)}`. Результат отобразим на master бэнде:

Beverages		Total by Report: \$2,222.71		
Soft drinks, coffees, teas, beers, and ales				
Name	Quantity per unit	Price	Units in stock	
1 Chai	10 boxes x 20 bags	\$18.00	39.00	
2 Chang	24 - 12 oz bottles	\$19.00	17.00	
3 Chartreuse verte	750 cc per bottle	\$18.00	69.00	
4 Côte de Blaye	12 - 75 cl bottles	\$263.50	17.00	
5 Guaraná Fantástica	12 - 355 ml cans	\$4.50	20.00	
6 Ipoh Coffee	16 - 500 g tins	\$46.00	17.00	
7 Lakkalikööri	500 ml	\$18.00	57.00	
8 Laughing Lumberjack Lager	24 - 12 oz bottles	\$14.00	52.00	
9 Outback Lager	24 - 355 ml bottles	\$15.00	15.00	
10 Rhönbräu Klosterbier	24 - 0.5 l bottles	\$7.75	125.00	
11 Sasquatch Ale	24 - 12 oz bottles	\$14.00	111.00	
12 Steeleye Stout	24 - 12 oz bottles	\$18.00	20.00	
		Total by Beverages: \$455.75		

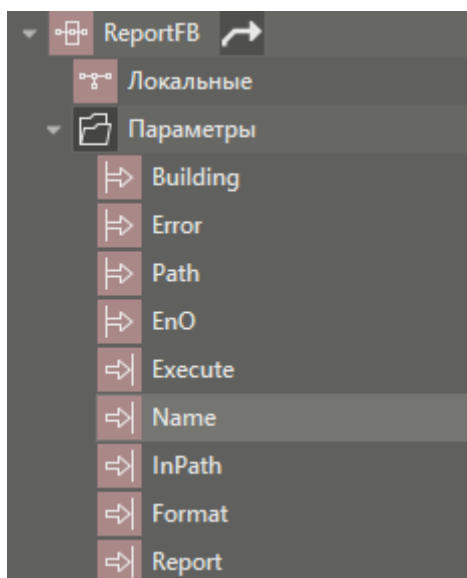
Каждый раз, когда в отчете будет печататься master бэнд, будет выводиться итог по отчету. Используя результаты расчетов, можно вычислить долю каждой категории из общего итога. Результат отобразим в процентах. Для расчета доли, следует итог по категории разделить на итог по отчету - $\{(Sum(DataBand2,Products.UnitsInStock) / Totals.Sum(Products, Products.UnitsInStock))\}$. Для текстового компонента, в котором будет выводиться доля, следует установить процентный формат. Результат отобразим на master бэнде:

Beverages		Total by Report: \$2,222.71		
Soft drinks, coffees, teas, beers, and ales		Percent: 17.92 %		
Name	Quantity per unit	Price	Units in stock	
1 Chai	10 boxes x 20 bags	\$18.00	39.00	
2 Chang	24 - 12 oz bottles	\$19.00	17.00	
3 Chartreuse verte	750 cc per bottle	\$18.00	69.00	
4 Côte de Blaye	12 - 75 cl bottles	\$263.50	17.00	
5 Guaraná Fantástica	12 - 355 ml cans	\$4.50	20.00	
6 Ipoh Coffee	16 - 500 g tins	\$46.00	17.00	
7 Lakkalikööri	500 ml	\$18.00	57.00	
8 Laughing Lumberjack Lager	24 - 12 oz bottles	\$14.00	52.00	
9 Outback Lager	24 - 355 ml bottles	\$15.00	15.00	
10 Rhönbräu Klosterbier	24 - 0.5 l bottles	\$7.75	125.00	
11 Sasquatch Ale	24 - 12 oz bottles	\$14.00	111.00	
12 Steeleye Stout	24 - 12 oz bottles	\$18.00	20.00	
		Total by Beverages: \$455.75		

Таким образом, рассчитать можно любой итог в отчете. Для расчета итога не связанного с бэндами следует использовать префикс Totals к имени функции, через разделитель ".".

9.4. ФОРМИРОВАНИЕ ОТЧЕТА В ИСПОЛНИТЕЛЬНОЙ СИСТЕМЕ

Для того чтобы иметь возможность формировать отчет в режиме исполнения необходимо во время разработки проекта добавить в него служебный ФБ ReportFB из библиотеки: BaseObjects.ФБ.Системны, который управляет формированием отчета:

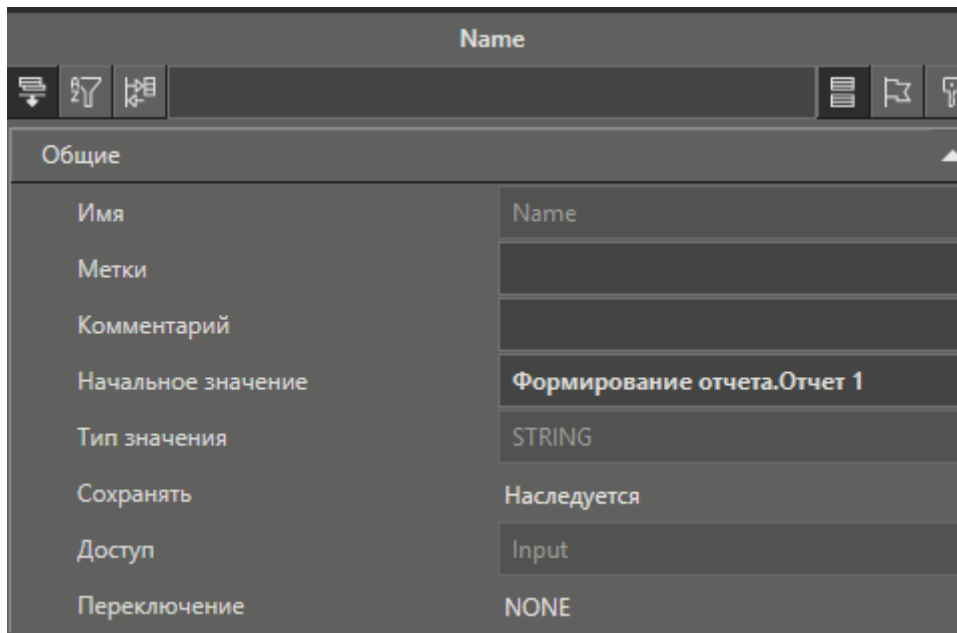


Настройка связи между ФБ и отчетом

Чтобы указать, каким отчетом будет управлять данный ФБ нужно настроить или вход Name или вход Report

Если принято решение использовать вход Name

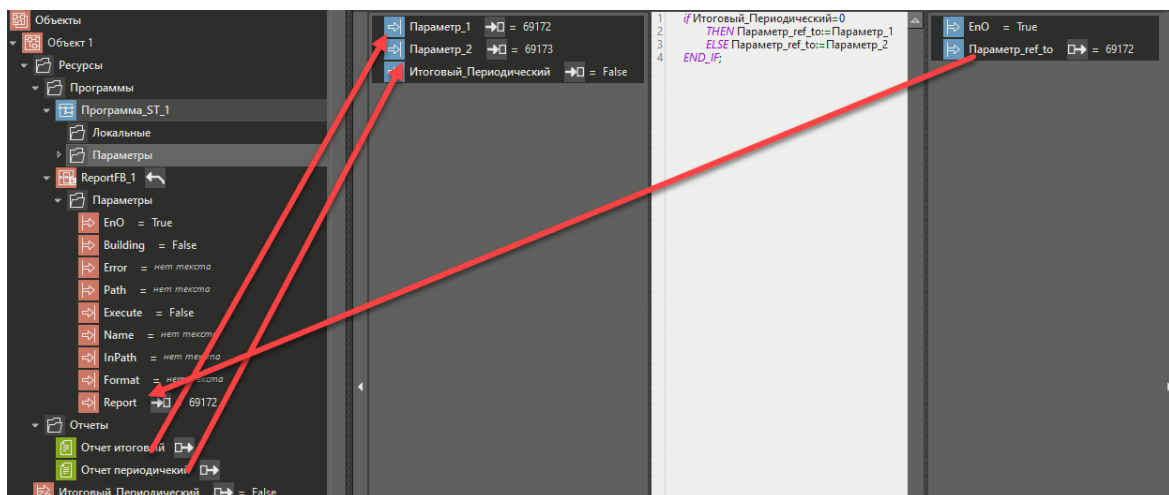
В настройках входа Name необходимо указать в качестве значения по умолчанию имя отчета, которым будет управлять данный ФБ. Имя отчета содержит в себе имя объекта и его родителей, в котором находится отчет, и имя самого отчета. Поэтому можно вставить ранее скопированное полное имя отчета и удалить оттуда несущественные элементы. Например, свойство Полное имя имеет значение: *Объекты.Формирование отчета.Ресурсы.Отчеты.Отчет 1*, нужно оставить - *Формирование отчета.Отчет 1*



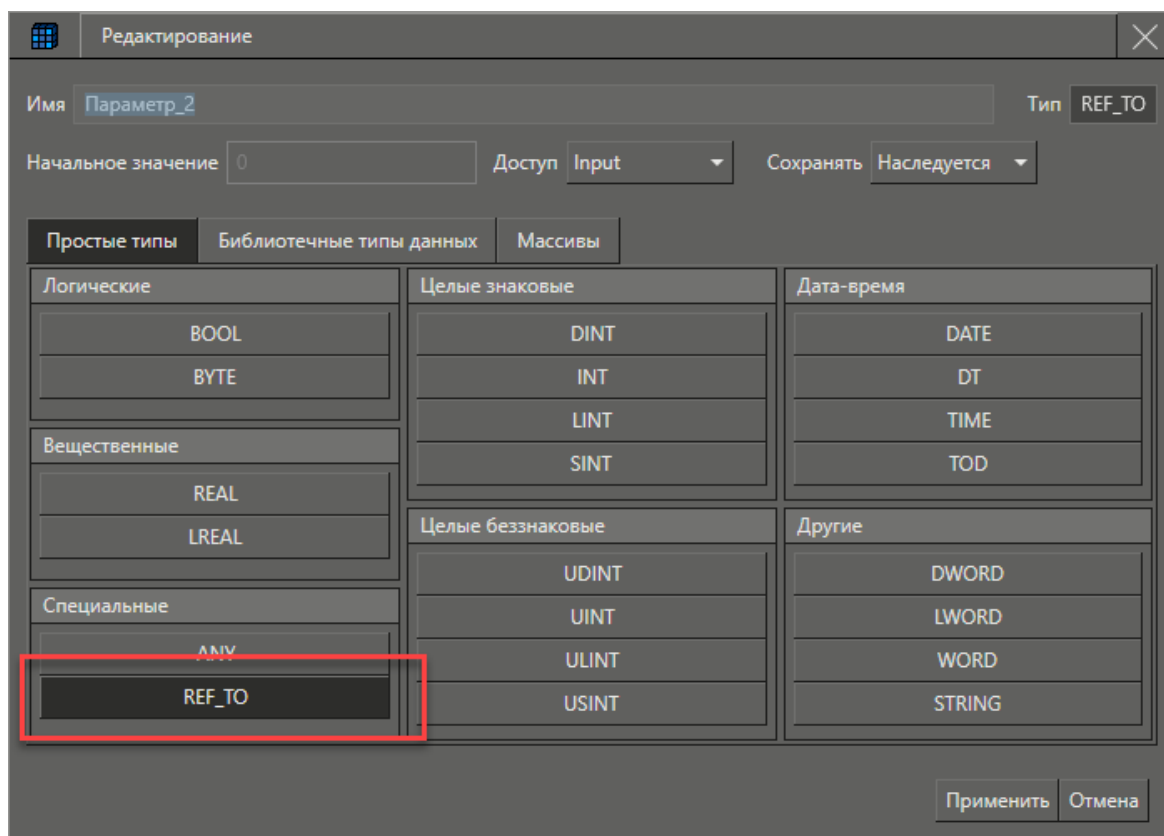
Если отчет создан в дереве системы, то в свойстве нужно указать только имя отчета. Имя узла, указывать не требуется.

Если принято решение использовать вход Report

Нужно на вход Report передать ID отчета, который нужно сформировать. Это можно сделать перетаскив на вход Report нужный отчет. Если один ФБ служит для формирования несколькими отчетами, то тогда вход нужно определять в программе. В программу добавляется параметр типа REF_TO:



На рисунке переменные программы Параметр_1, Параметр_2 и Параметр_ref_to имеют специальный тип REF_TO:



Каждый отчет связан с входным параметром. Какой именно отчет формировать определяется переменной Периодический_Итоговый типа BOOL (в окне клиента визуализации она может быть связана с Кнопкой с фиксацией). Далее в программе определяем значения выхода программы, который в свою очередь связываем в входом Report функционального блока, который управляет формированием отчета.

Формирование выбранного отчета

Чтобы запустить формирование отчета нужно на вход Execute функционального блока, который управляет формированием отчета подать значение TRUE.

Просмотр отчета в окне визуализации.

Для просмотра отчета можно использовать элемент Контейнер окна

Один из возможных вариантов описан в разделе Пример создания отчета

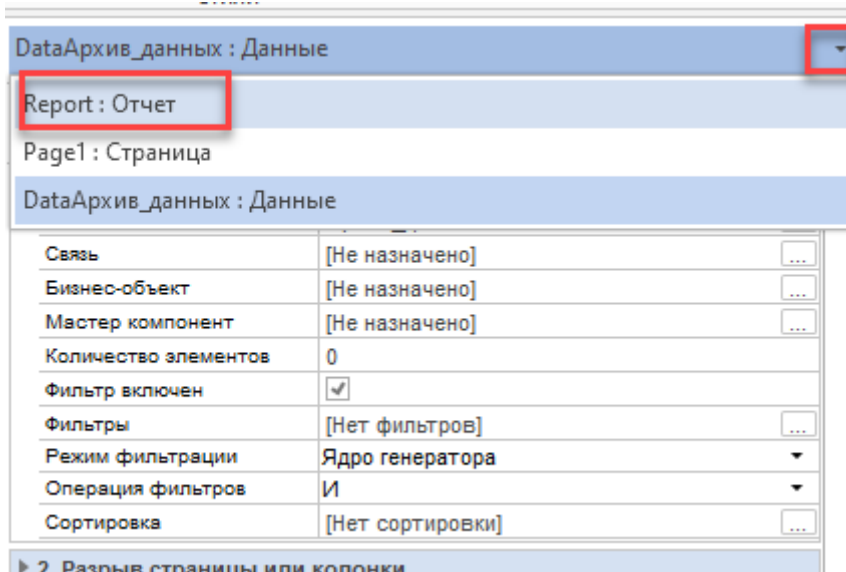
Выбор режима вычисления

При разработке отчетов и панелей индикаторов, выражения могут обрабатываться в режиме Интерпретация (Interpretation) или Компиляция (Compilation).

В режиме Компиляция для вычисления выражений используется CSharp компилятор. В этом случае, допускается использование событий, различных методов и функций платформы .NET Framework. Однако, время построения отчета или панелей индикаторов замедляется, а также требуется больший объем оперативной памяти.

В режиме Интерпретация для вычисления выражений используется собственный интерпретатор. Это ускоряет построение отчета или панели индикаторов, снижает требуемый объем оперативной памяти. Однако, в отчете или на панели индикаторов можно использовать только встроенные функции и методы. Не допускается использование событий и сторонних скриптов.

По умолчанию используется режим Компиляция. Чтобы сменить режим нужно открыть свойства отчета:



Изменить свойство Режим вычислений:

Report : Отчет

⌘ ⌵ ⌘ ⚡ ⚙

▼ 1. Описание

Наименование отчета	Report	...
Псевдоним отчета	Report	...
Автор отчета		...
Описание отчета		...
Иконка отчета		
Изображение отчета	[Не назначено]	...

▼ 2. Данные

▼ 3. Глобализация

▼ 4. Ядро

Кэшировать Totals-функции	<input type="checkbox"/>	
Режим вычислений	Компиляция	▼
Сортировка страниц	Компиляция	
Версия ядра	Интерпретация	
Количество проходов	Один проход	
Ссылки на сборки	(Ссылки на сборки)	...
Кэш отчета	Выключен	▼
Единица измерения	Сантиметры	▼
Скриптовый язык	CSharp	▼
Остановиться после стр...	0	
Стили	(Стили)	...

▼ 5. Вид

10. РАБОТА В РЕЖИМЕ ИСПОЛНЕНИЯ

10.1. РАБОТА СО СРЕДОЙ ИСПОЛНЕНИЯ

Среда исполнения (исполнительная система) – программа, в которой работает (исполняется) проект, предварительно созданный и скомпилированный в среде разработки. Среда исполнения может быть установлена и запущена отдельно, как независимое приложение, либо может входить в состав среды разработки.

Если в проекте в дереве системы добавлено несколько узлов, то количество запускаемых исполнительных систем должно соответствовать количеству узлов.

Проект можно загрузить в среду исполнения как автоматически, используя команды среды разработки, так и вручную, самостоятельно переместив нужные файлы в рабочую папку среды исполнения, либо используя возможности MasterSCADA 4D Monitor.

Важно! При загрузке новой конфигурации с использованием MasterSCADA 4D Monitor произойдет перезагрузка среды исполнения.

Если среда исполнения запущена, то запускаются 4 процесса:

- `mplc` – основной процесс исполнительской системы;
- `nginx` – процесс веб-сервера;
- `node_ms4d` – процесс генератора отчетов;
- `mplc_service` – вспомогательный процесс, контролирующий работу основного. Если используется отдельный инсталлятор среды исполнения для Windows, то вместо `mplc_service` при старте компьютера запускается Windows-служба `MS4DService.WinService.exe`; при вводе логина пользователя запускается `MS4DMonitor.exe` для управления работой сервиса.

10.1.1. РАБОЧАЯ ПАПКА СРЕДЫ ИСПОЛНЕНИЯ

Рабочая папка независимо установленной среды исполнения для ОС Windows носит название *Server*, а в других ОС – *mplc4*.

Местонахождение рабочей папки может быть задано в настройках сервиса. По умолчанию, она находится в:

- для ОС Windows: `C:\ProgramData\MPSSoft\MasterSCADA4DRT[номер версии]/Server`. Эта папка является общей для всех пользователей, так как сервис запускается при старте компьютера;
- для других ОС – `/opt/mplc4`.

Важно! В разных контроллерах путь к рабочей папке может отличаться. Информация о местонахождении рабочей папки смотрите в сопроводительной документации на контроллер.

Важно! Путь к рабочей папке среды исполнения не может содержать в себе буквы русского алфавита.

Структура рабочей папки

Один экземпляр исполнительной системы

Если на устройстве запускается один экземпляр исполнительной системы, то для нулевого экземпляра в корне *Server (mplc)* структура будет следующей:

Имя	Дата изменения	Тип	Размер
cfg	29.11.2019 15:10	Папка с файлами	
CltCertificateStore	29.11.2019 15:10	Папка с файлами	
htdocs	29.11.2019 15:10	Папка с файлами	
IEC61850	29.11.2019 15:11	Папка с файлами	
log	29.11.2019 15:10	Папка с файлами	
logs	29.11.2019 15:10	Папка с файлами	
temp	29.11.2019 15:10	Папка с файлами	
data	29.11.2019 15:11	Data Base File	44 КБ
EventsData	29.11.2019 15:11	Data Base File	40 КБ
session.bin	29.11.2019 15:11	Файл "BIN"	1 КБ
Users-3dd4606d-8247-4ae6-b8de-b18ea6...	29.11.2019 15:11	JSON File	1 КБ

Название	Назначение
Папки	
cfg	<p>Содержит конфигурацию проекта (кроме файлов, необходимых для работы клиента визуализации):</p> <ul style="list-style-type: none"> • <i>config.bin</i> – конфигурация; • <i>VMInfo.json</i> – дерево проекта с Id - элементов и типами значений; • <i>_files.xml</i> – список хеш-сумм загруженных файлов; • подпапка <i>templates</i> – содержит шаблоны отчетов
CltCertificateStore	Сертификат UA-клиента

Название	Назначение
CertificateStore	Сертификат UA-сервера
htdocs	Содержит файлы, необходимые для работы клиента визуализации
IEC61850	<p>Используется, если включена настройка Использовать подключение TLS для протокола IEC61840. Ключи и сертификаты TLS хранятся в папке <i>IEC61850</i>. Ключи и сертификаты сервера хранятся в папке <i>IEC61850/srv</i>. Ключи и сертификаты клиента хранятся в папке <i>IEC61850/clt</i>.</p> <p>Имя файла ключа клиента должно быть <i>MasterSCADA4Dclt_key.pem</i>. Имя файла сертификата клиента должно быть <i>MasterSCADA4Dclt.cer</i>.</p> <p>Имя файла ключа сервера должно быть <i>MasterSCADA4Dsrv_key.pem</i>. Имя файла сертификата сервера должно быть <i>MasterSCADA4Dsrv.cer</i>.</p> <p>Сертификаты CA должны храниться в папке <i>IEC61850/clt/root</i> для клиента и <i>IEC61850/srv/root</i> для сервера. Сертификаты CA могут иметь любое имя и расширение <i>.cer</i>.</p>
log\0	<p>Содержит лог-файлы исполнительной системы. Состав папки:</p> <p>Подпапка <i>ErrorDumps</i> содержит дампы, которые формируются автоматически при возникновении ошибки.</p> <p>Содержит лог-файлы типа <i>log_18_01_07.txt</i></p>
logs	Хранятся лог-файлы клиента визуализации
Temp	<p>Содержит временные файлы nginx.</p> <p>Также содержатся файлы с расширением <i>pdb</i> с соответствием строк программ ST проекта строкам LUA кода, который формируется после компиляции программы. Количество файлов соответствует количеству задач. Структура таких файлов представляет собой набор строк, каждая строка разделена пробелом на 4 части:</p>

Название	Назначение
	<ol style="list-style-type: none"> 1. Номер строки в Lua. Указываются только строки, в которых есть код, полезный для отладки. 2. ID элемента в проекте. Значение 0 соответствует функции программы задачи, которую нельзя отнести к конкретной программе проекта. 3. Номер строки в ST. Пока не используется для функции программы задачи, которую нельзя отнести к конкретной программе проекта. 4. Номер столбца в ST. Пока не используется для функции программы задачи, которую нельзя отнести к конкретной программе проекта.
Файлы, создаваемые в режиме исполнения	
data.db	Файл содержащий архив данных (для бд SQLite)
EventsData.db	Файл содержащий архив сообщений (для бд SQLite)
session.bin	Файл данных для горячего рестарта
Users-[уникальный GUID проекта]	Файл, в котором хранится информация о пользователях, добавленных в проект.

Если в проекте используются программы C#, то будет добавлена папка netcore

Несколько экземпляров исполнительной системы

Если на одном устройстве запускается несколько экземпляров исполнительной системы одновременно, то в рабочую папку добавляются подпапки и файлы для ненулевых экземпляров.

Рассмотрим структуру рабочей папки на примере исполнительной системы для ОС Windows. К существующим файлам и папкам для нулевого экземпляра в рабочую папку будут добавлены следующие подпапки и файлы:

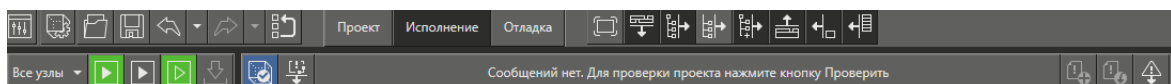
Папка	Назначение
Server\N, где N номер экземпляра	Содержит папку с конфигурацией проекта <i>cfg</i> и файлы, создаваемые в режиме исполнения для узла

Папка	Назначение
	загружаемого в экземпляр N (подробное описание файлов содержится в предыдущей таблице)
Server\htdocs\N, где N номер экземпляра	Содержит файлы, необходимые для работы клиента визуализации с узлом загруженным в экземпляр N
\Server\log\N, где N номер экземпляра	Содержит лог-файлы типа <i>log_18_01_07.txt</i> , соответствующие работе узла, загруженного в экземпляр N, а также подпапки <i>ErrorDumps</i> и <i>logs</i>

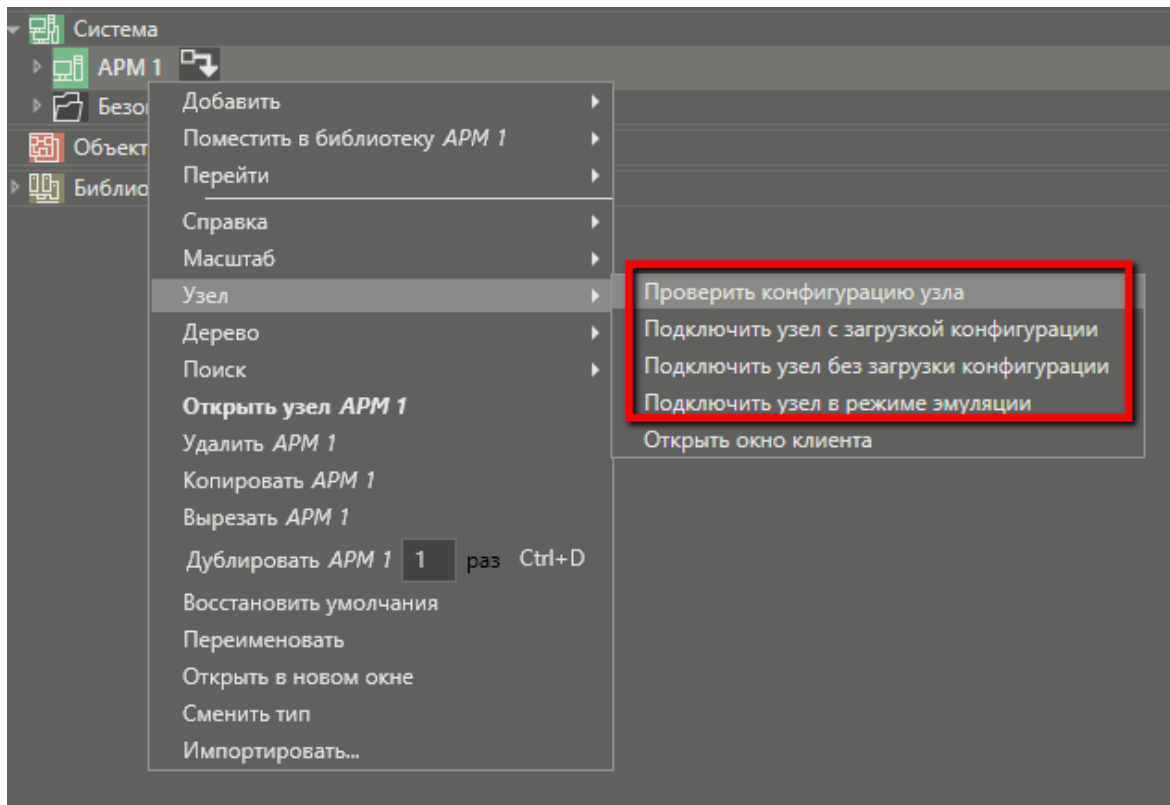
10.1.2. ПОДГОТОВКА ПРОЕКТА К ЗАПУСКУ В СРЕДЕ ИСПОЛНЕНИЯ

С точки зрения файловой системы, разрабатываемый проект представляет собой папку, содержащую файлы (в т.ч. файл БД **.fdb*) и вложенные папки. Не рекомендуется переименовывать эти элементы средствами ОС. Для переименования проекта должен использоваться инструмент Сохранить как. В случае если проект не отображается в диалоге Открытия проекта, возможно, папка проекта и файл БД имеют разные имена.

Для компиляции проекта необходимо выполнить любую из команд вкладки Исполнение на панели инструментов Среды разработки: Подключить, Эмуляция или Проверить.



После выполнения команд контекстного меню узла (см. рис. ниже) произойдет компиляция той части проекта, которая относится к данному узлу:

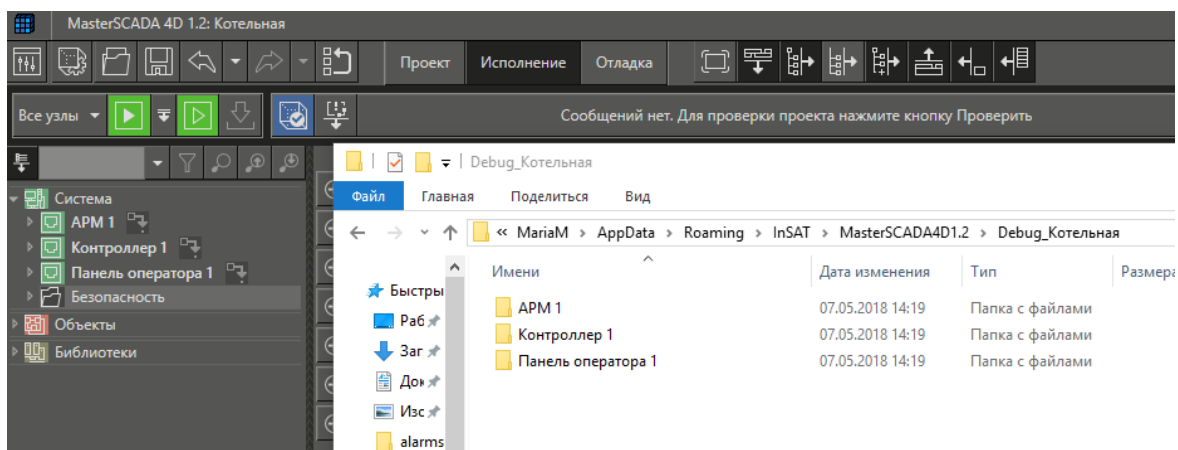


Если узел имеет межузловые связи, то будут также скомпилированы части проекта, относящиеся к связанным узлам. Нужные библиотеки функциональных блоков подгружаются в ту задачу узла, в которую ее заложил разработчик проекта.

Место хранения скомпилированных проектов: `c:\Users\[имя пользователя]\AppData\Roaming\MPSSoft\MasterSCADA4D[версия]\`.

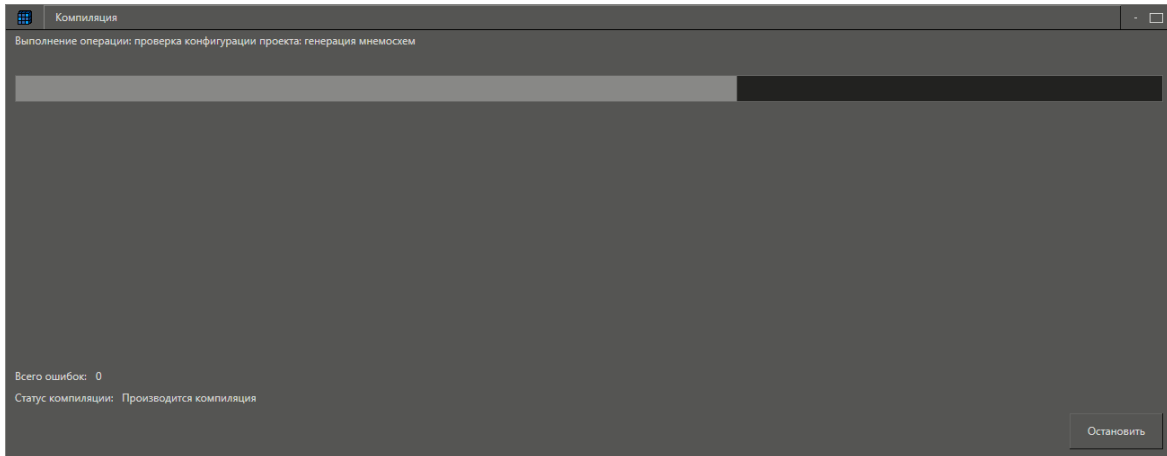
Каждый проект сохраняется под именем: `Debug_[имя проекта]`.

Внутри данной папки может быть несколько вложенных папок. Их количество и названия соответствуют количеству узлов в проекте. В каждой вложенной папке создается проект для каждого отдельного узла, который будет загружаться в свою исполнительную систему автоматически или вручную.

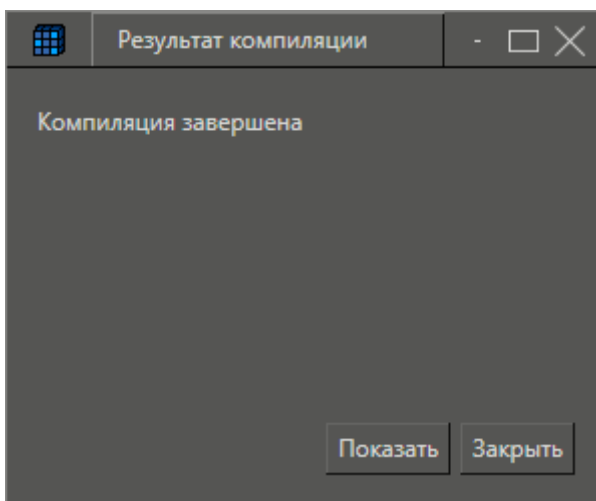


Важно! Скомпилированный проект из папки Debug_[имя проекта] невозможно открыть в среде разработки.

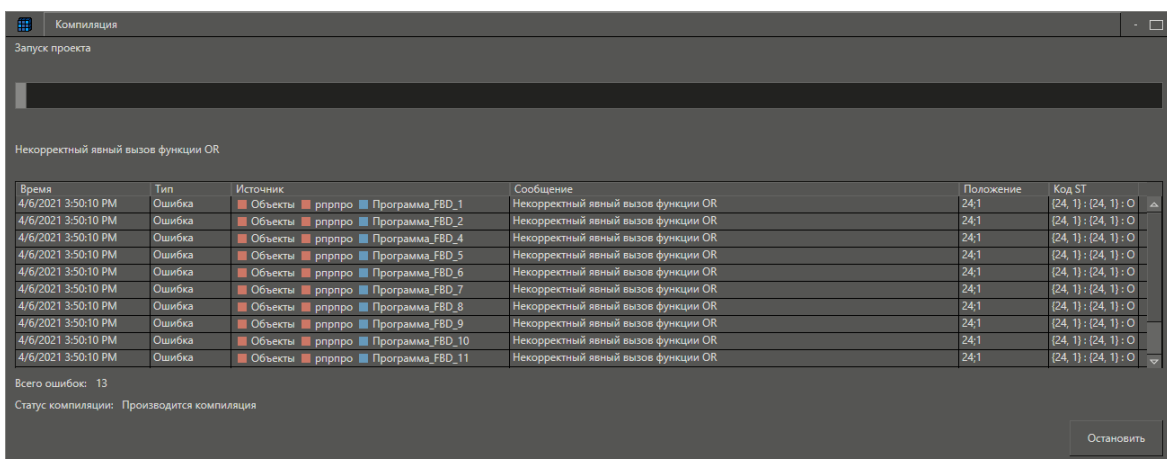
Информация о ходе компиляции указывается в окне:



После окончания появится окно:



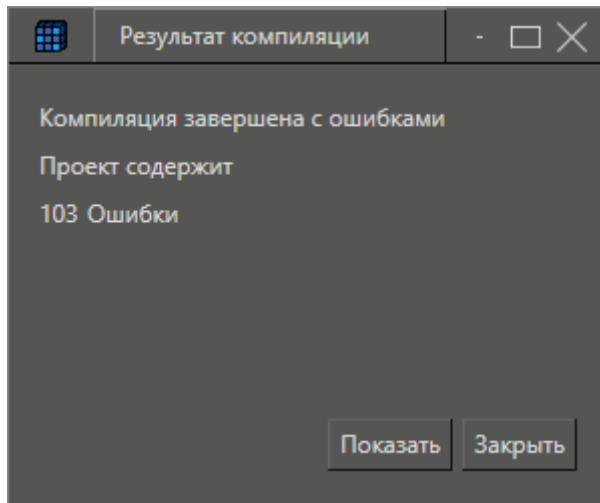
Если возникнут ошибки, то в окне отобразятся ошибки:



Если ошибка возникает в типе, то в поле источник будет указан тип этого элемента.

Если нажать кнопку Остановить, то компиляция проекта прекратится.

После окончания компиляции появится окно:



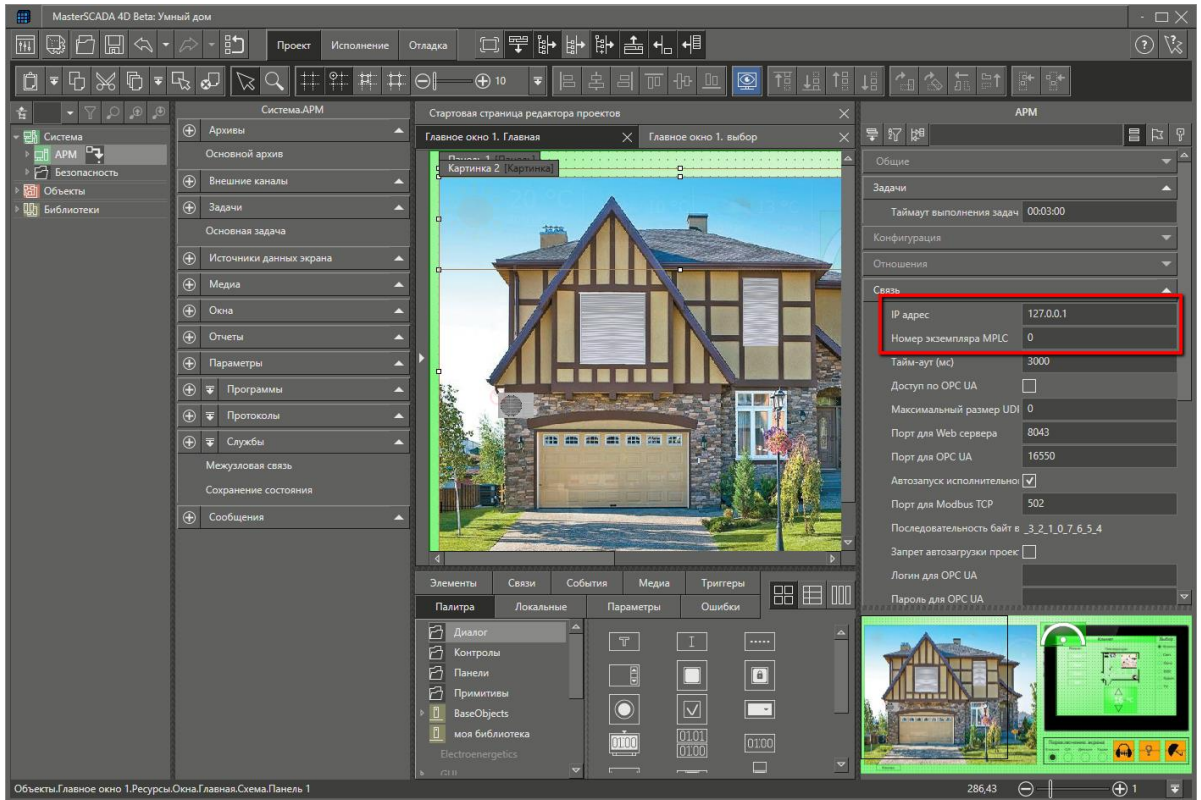
При нажатии на кнопку Показать откроется окно Ошибки проекта

10.1.3. АВТОМАТИЧЕСКАЯ ЗАГРУЗКА ПРОЕКТА

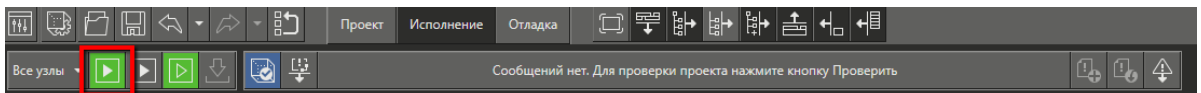
Рассмотрим два варианта работы: в первом случае исполнительная система и среда разработки работают на одном компьютере; во втором – на разных устройствах.

Исполнительная система и среда разработки работают на одном компьютере

Для примера допустим, что в проекте имеется единственный узел с такими параметрами связи, которые показаны на рисунке ниже (для узла задан IP-адрес текущего компьютера или 127.0.0.1, свойство Номер экземпляра MPLC равно 0):



Для того чтобы произвести автозапуск исполнительной системы, выполняем команду Подключить:



После этого начинается подготовка проекта к запуску. Сразу после завершения компиляции проекта MasterSCADA 4D попытается подключиться к уже запущенной среде исполнения. Если среда исполнения не запущена и в настройках узла установлен флаг Автозапуск исполнительной системы, то среда разработки запустит входящую в ее состав версию исполнительной системы. Если флаг не установлен и среда исполнения не запущена, то появится сообщение, что связь с исполнительной системой (узлом) установить не удалось:



Если в проекте созданы окна и одно из них назначено стартовым, то после загрузки проекта в среду исполнения автоматически запустится клиент визуализации.

Среда исполнения будет подключена к среде разработки. В интерфейсе среды разработки отражаются текущие значения параметров. При необходимости их можно заменить значениями, введенными разработчиком проекта.

Среда разработки отключится от среды исполнения после выполнения команды Отключить в панели инструментов, либо в контекстном меню узла:



Если среда исполнения была запущена средой разработки, то после выполнения команды Отключить она закроется. Если же среда исполнения запускалась независимо, то она не закроется и проект продолжит исполняться в ней.

Важно! Флаг Автозапуск исполнительной системы работает только для узлов, в настройках которых задан локальный IP-адрес (127.0.0.1).

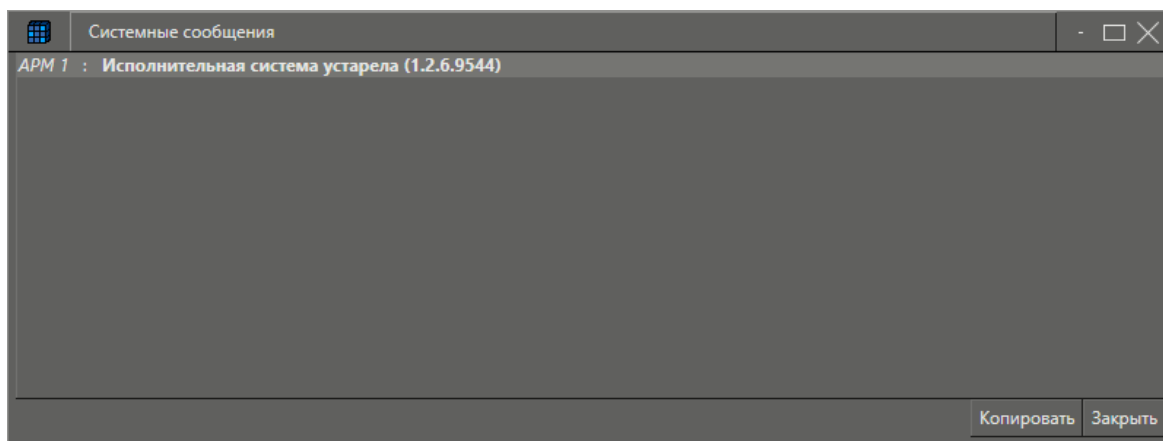
Среда разработки и среда исполнения работают на разных устройствах

В случае если в настройках узла указан IP-адрес, на котором установлена среда разработки MasterSCADA 4D, то следует убедиться, что на удаленном устройстве (компьютере, контроллере, панели и т.п.) установлена среда исполнения MasterSCADA 4D RT.

После выполнения команды Подключить произойдет компиляция проекта и его подключение к уже запущенной среде исполнения, а также загрузка в нее проекта. Если запущенная среда исполнения не будет найдена, то появится сообщение об ошибке. В этом случае среда разработки не будет пытаться запустить среду исполнения.

Несоответствие версии среды разработки и среды исполнения

Если версия среды разработки выше, чем версия среды исполнения, то появится сообщение:



В этом случае необходимо обновить среду исполнения. Среда исполнения для различных контроллеров и операционных систем, поставляются в комплекте со средой разработки. и

находятся в папке `c:\Program Files (x86)\MPSSoft\MasterSCADA 4D 1.2\bin\Config\MasterPLC\PLC`. Если среду исполнения не активировать, то она будет работать с ограничениями демоверсии.

Скачать демоверсию среды исполнения и получить информацию о способах покупки коммерческой версии можно на сайте компании "МПС Софт".

10.1.3.1. ИСПОЛЬЗОВАНИЕ ПАРОЛЯ ДЛЯ ЗАГРУЗКИ ПРОЕКТА

В среде исполнения можно задать пароль, который будет проверяться при загрузке проекта и обновлении версии ПО из среды разработки.

Пароль в среде исполнения можно указать в настройках MasterSCADA 4D Monitor или в свойстве Параметры запуска RT узла.

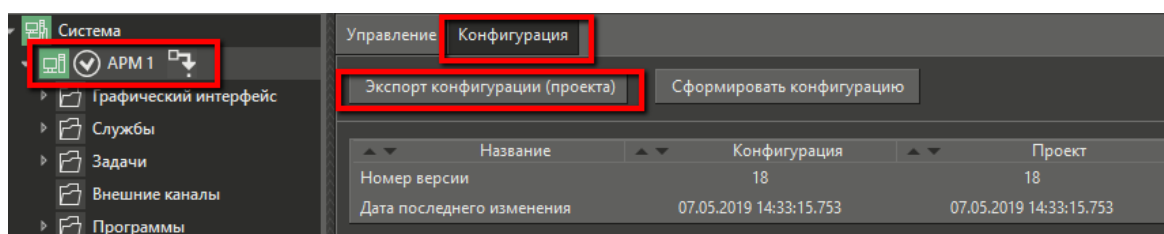
Если в исполнительной системе задан пароль, то загружать в него проект можно только, если в настройках среды разработки во вкладке Разное задан такой же пароль. Как правило используется для защиты публичного сервера от загрузки в него конфигурации пользователями.

10.1.4. РУЧНАЯ ЗАГРУЗКА ПРОЕКТА В СРЕДУ ИСПОЛНЕНИЯ

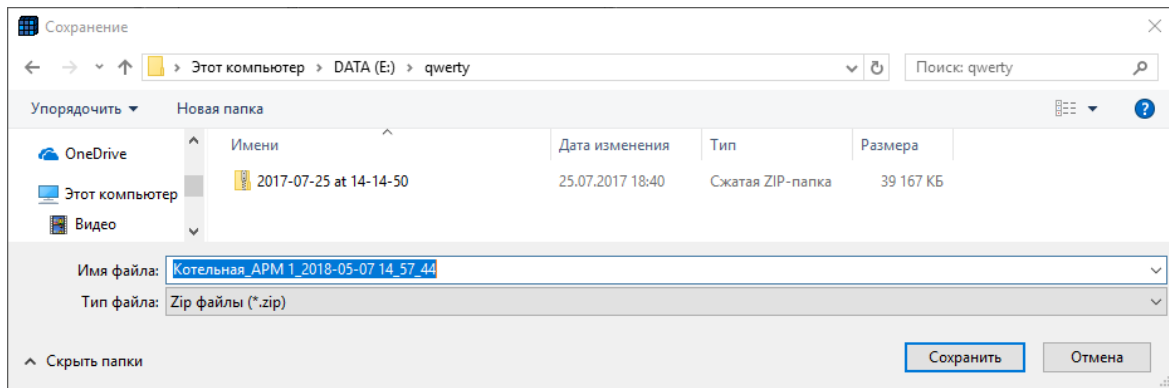
На тот случай, если автоматическая загрузка проекта в среду исполнения через среду разработки невозможна, предусмотрена его ручная загрузка путем копирования необходимых файлов в рабочую папку среды исполнения.

Среда исполнения автоматически запускает последний загруженный в нее проект. Под проектом понимается папка, которая создается в результате компиляции проекта. Для каждого узла создается своя папка.

Для того чтобы подготовить файлы для копирования в рабочую папку исполнительной системы, необходимо дважды нажать левой кнопкой мыши на узел в дереве системы. В средней части интерфейса откроется вкладка управления узлом, содержащая несколько разделов, в том числе раздел Конфигурация. Сначала необходимо нажать кнопку Сформировать конфигурацию, а затем Экспорт конфигурации (проекта).



После нажатия на кнопку Экспорт проекта откроется диалоговое окно Windows, предлагающее выбрать папку для сохранения архива проекта, предназначенного для работы в среде исполнения выбранного узла:



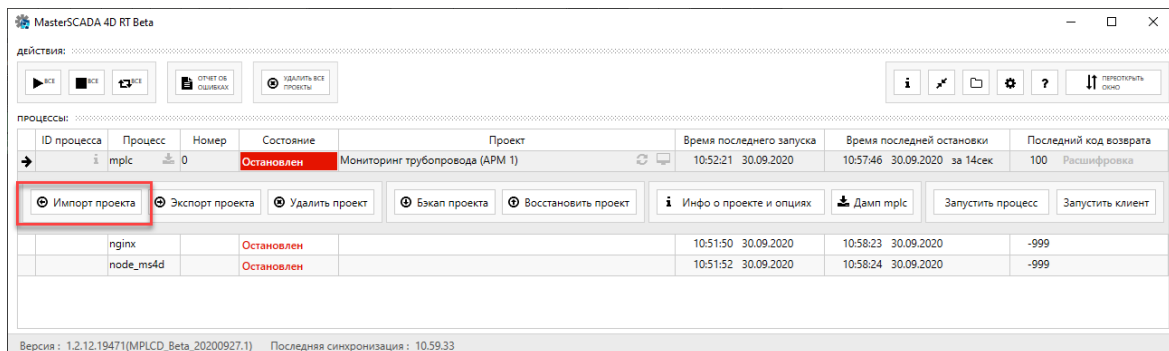
После нажатия на кнопку Сохранить создается архив, имеющий, по умолчанию, имя : *[название проекта]_[имя узла]_[дата и время экспорта]*.

После этого содержимое архива необходимо разместить в рабочей папке среды исполнения.

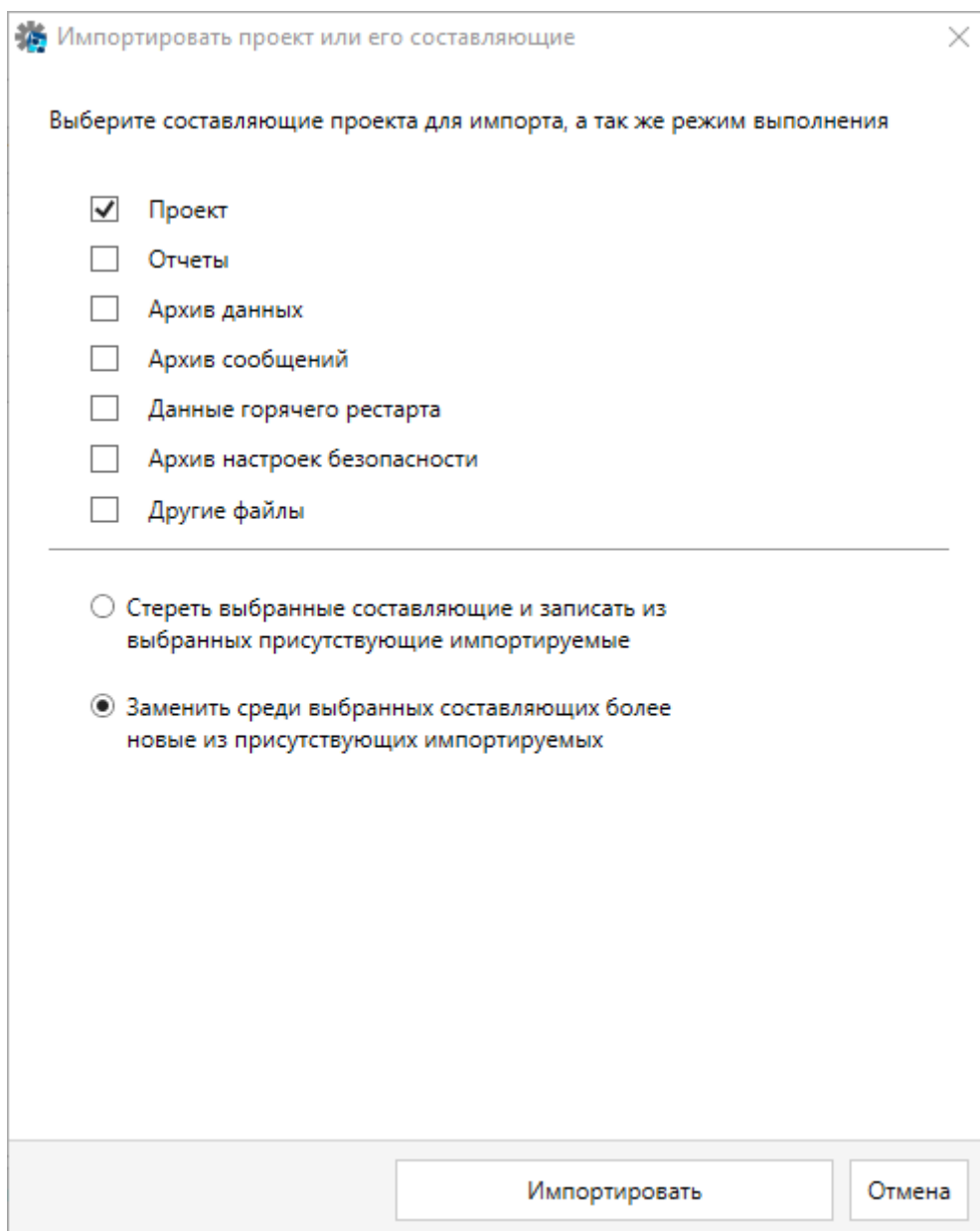
OS Windows

При помощи окна MasterSCADA 4D Monitor можно выполнить импорт ранее экспортированного проекта.

Для осуществления импорта в этом окне необходимо остановить процессы mpls и nginx и нажать на строку процесса mpls, после чего откроются необходимые дополнительные инструменты:



При нажатии кнопки Импорт проекта откроется диалоговое окно:



Окно позволит выбрать, какие части проекта необходимо импортировать. А также определить необходимо ли стереть все содержимое рабочей папки и заменить на содержимое папки импорта, или заменить только те файлы, что новее в папке импорта.

После нажатия на кнопку Импортировать откроется окно Windows, в котором требуется выбрать путь к архивному файлу, полученному в результате экспорта проекта из среды исполнения. В результате этого в рабочую папку MasterSCADA 4D RT загрузятся все необходимые файлы. После перезагрузки среды исполнения нужный проект запустится автоматически. По умолчанию, в настройках сервиса задан следующий путь к рабочей папке: *C:\ProgramData\MPSSoft\MasterSCADA4DRT[номер версии]*.

Кроме этого, не прибегая к процедуре импорта и экспорта, в эту папку можно также записать файлы проекта, скопировав содержимое папки *Debug_[имя проекта]/[имя узла]*, получившейся после компиляции проекта, в рабочую папку среды исполнения.

При старте исполнительной системы она автоматически запустит загруженный проект, но при условии, что исполнительная система не была запущена ранее при помощи bat-файлов, позволяющих выполнять проект из любого места дискового пространства.

Другие ОС

Загрузка проекта в исполнительные системы, работающие в других ОС, принципиально не отличается от загрузки проекта в исполнительную систему под ОС Windows: необходимо точно также переместить часть скомпилированного проекта, предназначенного для данного узла, в рабочую папку среды исполнения.

Необходимые файлы можно получить через экспорт проекта из среды исполнения, либо скопировав их непосредственно из папки: *Users\[имя пользователя]\AppData\Roaming\MPSSoft\MasterSCADA4DRT[номер версии]*.

Рабочая папка для других ОС носит название *MPLC4*. Как правило, она находится по адресу */opt/mplc4*, но путь к ней может и отличаться в разных контроллерах.

10.1.5. ВОССТАНОВЛЕНИЕ РАБОТЫ УЗЛА

Работу проекта в среде исполнения обеспечивают три процесса:

- *mplc* – основной процесс исполнительной системы;
- *nginx* – процесс веб-сервера;
- *node_ms4d* – процесс генератора отчетов;

Исполнительная система имеет средства самодиагностики, для реализации которых запускается еще один процесс - *mplc_service*, контролирующий работу основного процесса. В случае если процесс *mplc* завершается в результате ошибки, то процесс *mplc_service* перезапускает его. Если используется отдельный инсталлятор среды исполнения для Windows, то вместо *mplc_service* при старте компьютера запускается Windows-служба *MS4DService.WinService.exe*, а при вводе логина пользователя запускается *MS4DMonitor.exe* для управления работой сервиса.


При потере связи со средой исполнения клиент визуализации сообщит оператору об ошибке.

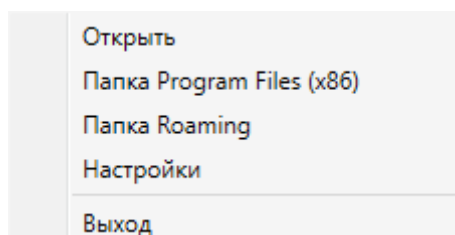
После восстановления связи с узлом, клиент визуализации переподключится к среде исполнения автоматически.


Управлять работой процессов среды исполнения можно при помощи MasterSCADA 4D Monitor.

10.1.6. MASTERSCADА 4D MONITOR

MasterSCADA 4D Monitor запускается автоматически после установки среды исполнения MasterSCADA 4D RT. Имя процесса в диспетчере задач - *MS4DMonitor.exe*. Позволяет работать со средой исполнения и сопутствующими процессами, а также удалять, импортировать и экспортировать скомпилированные проекты, а также собирать диагностическую информацию.

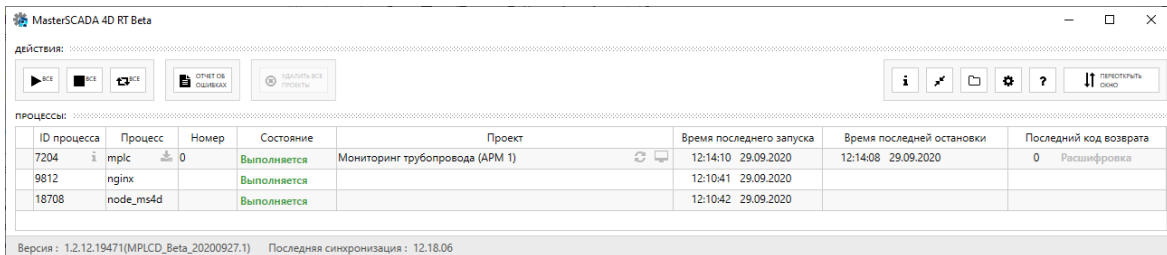
После установки и запуска среды исполнения, в области уведомлений панели задач появляется значок  (этот значок присутствует всегда, когда запущена среда исполнения, за исключением случая, когда пользователь вышел из приложения MasterSCADA 4D Monitor в текущей сессии работы среды исполнения). При нажатии правой кнопки мыши открывается контекстное меню:




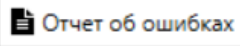


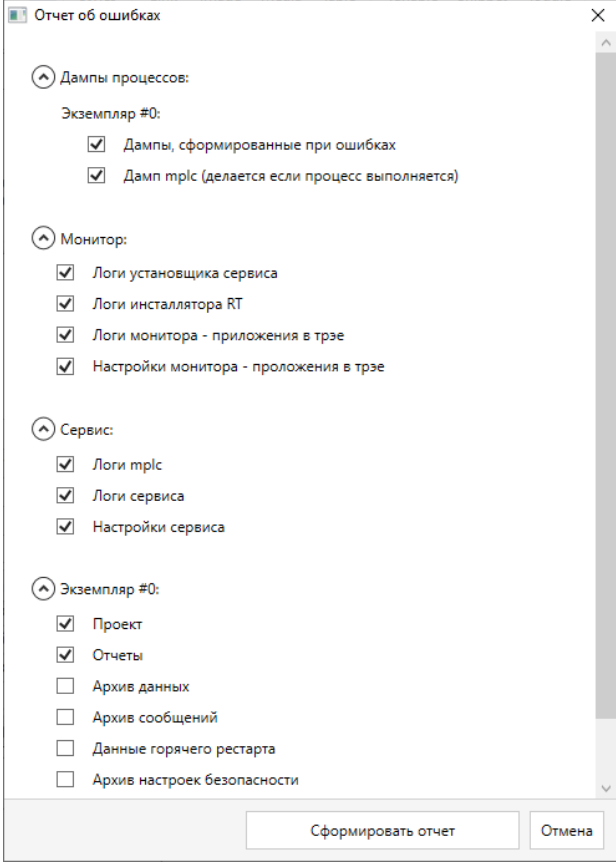

Пункт меню	Назначение
Открыть	Открывает приложение MasterSCADA 4D Monitor, предназначенное для контроля и управления MasterSCADA 4D RT. Приложение можно открыть также двойным нажатием левой кнопкой мыши на значок  .
Папка Program Files (x86)	Открывает папку установки среды исполнения в проводнике Windows. <i>C:\Program Files (x86)\MPSSoft\MasterSCADA 4D RT</i>
Папка Roaming	Открывает папку, в которой хранится вспомогательная информация <i>C:\Users\[имя пользователя]\AppData\Roaming\MPSSoft\MasterSCADA4DRT</i>
Настройки	Открывает окно настройки среды исполнения
Выход	Осуществляется выход из приложения MasterSCADA 4D Monitor. Процессы среды исполнения продолжают свою работу. После перезапуска среды исполнения MasterSCADA 4D Monitor запустится снова. Для того чтобы запустить приложение вручную, необходимо запустить exe-файл <i>c:\Program Files (x86)\MPSSoft\MasterSCADA 4D RT\MS4DMonitor.exe</i>

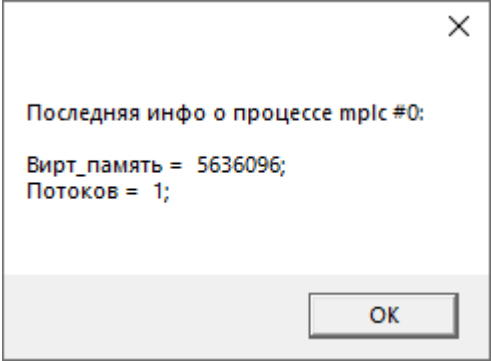
Вид приложения




Окно приложения имеет вид:



Элемент	Описание
	Запускает все процессы, необходимые для работы среды исполнения.
	Останавливает все процессы, перечисленные в таблице.
	Перезапускает все процессы, перечисленные в таблице.
	Открывает окно для формирования отчета об ошибках среды исполнения., в котором нужно установить какие именно лог-файлы необходимо включить в отчет:

Элемент	Описание
	 <p>Затем необходимо нажать кнопку Сформировать отчет. После завершения формирования отчета папка с отчетом откроется в проводнике Windows. Полученную информацию необходимо отправить в службу технической поддержки. В сопроводительном письме необходимо указать причину формирования отчета.</p>
Кнопка Удалить все проекты	Кнопка позволяет удалить из всех исполнительных систем проекты. Кнопка становится активной, если все процессы остановлены
Таблица Процессы	<p>Содержит информацию о запущенных процессах, относящихся к работе среды исполнения:</p> <ul style="list-style-type: none"> • ID процесса – отображает ID процесса в диспетчере задач, при нажатии на значок , откроется окно, в котором отобразится дополнительная информация о процессе, например, количество используемых

Элемент	Описание
	<p>потоков и объем занимаемом памяти</p>  <ul style="list-style-type: none"> • Процесс – имя процесса; • Номер – номер экземпляра среды исполнения. Если запущено несколько сред исполнения на одном компьютере, то в данном поле указывается номер экземпляра среды исполнения, который должен совпадать с номером, заданным в среде разработки в свойствах узла; • Состояние – состояние процесса: Выполняется, Остановлен и т.п.; • Проект - показывает имя работающего проекта. Справа от имени проекта отображаются две иконки, позволяющие обновить имя проекта и открыть клиент визуализации; • Время запуска, Время остановки – указывается время работы процесса; • Последний код возврата – указывает номер ошибки, если процесс запустить не удалось или в процессе работы возникли проблемы. При нажатии на слово Расшифровка в поле откроется информационное окно, в котором отобразится уточняющая информация, с текстовым описанием ошибки; <p>При нажатии на строку процесса, открываются дополнительные строки, позволяющие выполнять действия с процессом, например, остановить его. Для процесса mpls есть возможность осуществлять работу с проектом.</p>

Элемент	Описание
	Позволяет скопировать в буфер обмена дополнительную информацию о выделенном процессе, при нажатии на правую кнопку мыши
	Сворачивает дополнительные строки, появляющиеся после нажатия на строку процесса в таблице.
	Открывает окно настройки среды исполнения.
	Открывает справочную систему.
	Позволяет переоткрыть окно
Строка статуса	Указывает номер версии среды исполнения, а также время последней проверки состояния процессов среды исполнения. Находится в нижней части окна. Номер версии может быть скопирован, если нажать правую кнопку мыши на кнопку 

Возможные коды возврата:

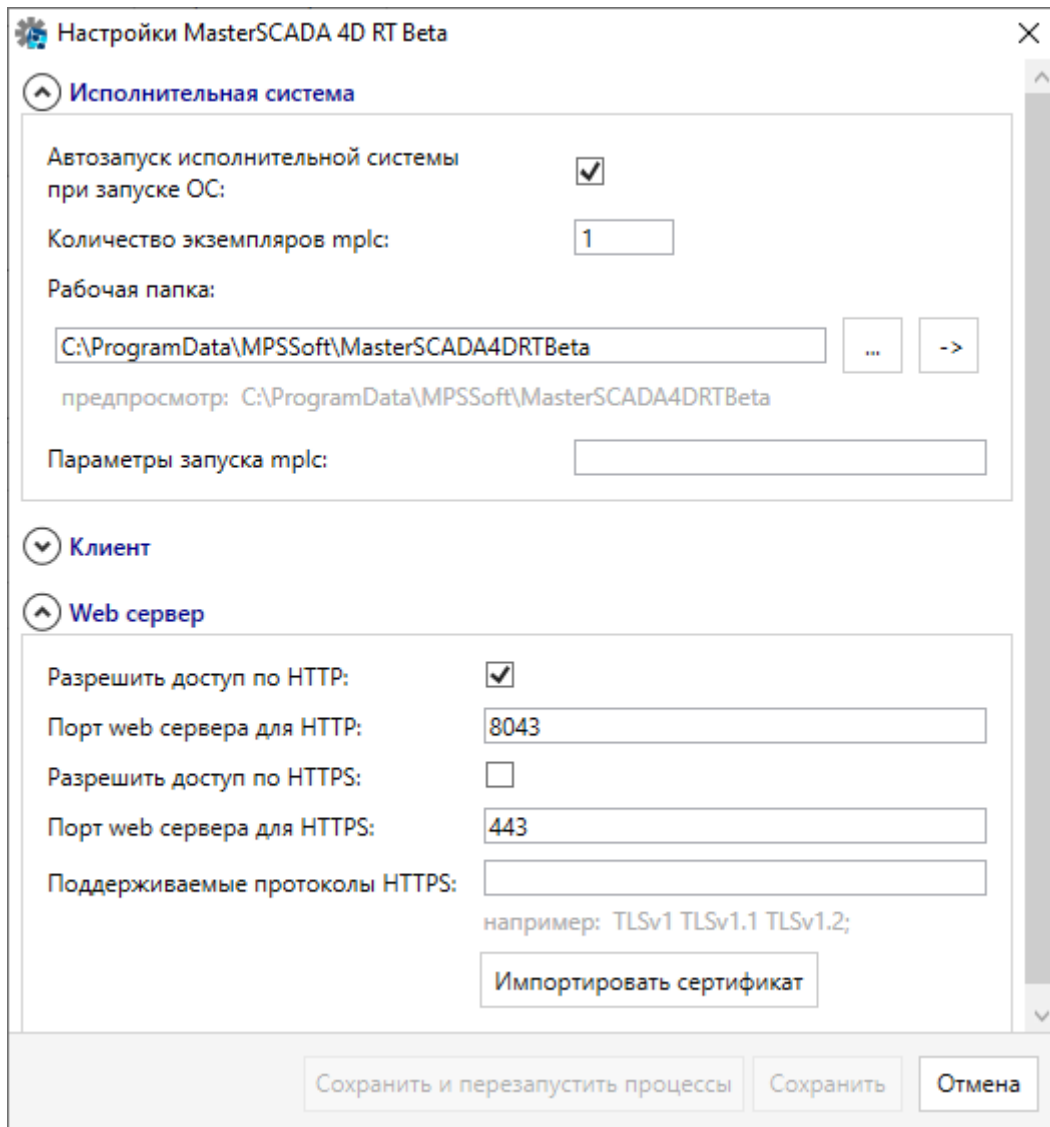
Код возврата	Расшифровка кода возврата
Ошибки, выдаваемые в процессе работы. При этих ошибках автоматически формируется дамп памяти:	
2	Аварийное завершение
3	Ошибка памяти
4	Ошибка интерпретатора кода
5	Ошибка интерпретатора кода
6	Зависание задачи

Код возврата	Расшифровка кода возврата
Ошибки, выдаваемые при загрузке и старте конфигурации:	
-1	Ошибка конфигурации
-5	Ошибка сети
-6	Ключ защиты не обнаружен
-7	Используется неподдерживаемый протокол
-8	Ошибка открытия основного UDP-сокета
-9	Нехватка памяти

10.1.6.1. ОКНО НАСТРОЙКИ СРЕДЫ ИСПОЛНЕНИЯ

Если в приложении MasterSCADA 4D Monitor нажать кнопку , то откроется окно управления средой исполнения.

Вид окна:



Элемент	Описание
Исполнительная система	
Автозапуск исполнительной системы при запуске ОС	Если флаг установлен, то среда исполнения запускается автоматически при старте ОС.
Количество экземпляров mpls	Если требуется запустить несколько исполнительных систем на одном компьютере, то в данном поле необходимо указать их количество.

Элемент	Описание
Рабочая папка	<p>Указывается место хранения рабочей папки <i>Server</i>. Место хранения может быть изменено.</p> <p>Важно! Путь к рабочей папке среды исполнения не может содержать в себе буквы русского алфавита.</p>
Параметры запуска <i>mrpc</i>	Можно задать те же ключи запуска среды исполнения, что и в среде разработке в свойстве узла Параметры запуска RT.
Клиент	
Автозапуск клиента	Если флаг установлен, то клиент запустится автоматически при старте ОС.
Адрес сервера	<p>Задается путь к веб-серверу: <i>[IP]:[порт TCP/IP] /[номер экземпляра]/</i>.</p> <p>Для того чтобы при запуске клиента не появлялось окно авторизации, можно ввести в адресную строку: <i>127.0.0.1:8043/index.html?user=sa&password=123</i>, где 127.0.0.1 - IP-адрес узла, 8043 - порт, sa - имя оператора, а 123 - его пароль.</p> <p>Если в конце адресной строки добавить <i>/test</i>, и при этом в настройках среды указано Тип транслятора НМІ - v1+v2, то откроется клиент визуализации выполненный в версии НМІ v2.</p> <p>Есть возможность указать несколько адресов для подключения. Адреса вводятся через запятую. Например, <i>http://192.168.110.1:8043,http://192.168.110.12:8043</i></p> <p>При автозапуске или ручном запуске открывается первый доступный адрес в порядке следования.</p>
Полноэкранный режим для клиента	Если флаг установлен, то клиент запустится во весь экран. В полноэкранном режиме клавиша F11 не работает
Параметры запуска клиента	Могут быть заданы следующие ключи:

Элемент	Описание
	<p>-d – если ключ установлен, то в процессе работы по кнопке F12 отрываются средства разработчика;</p> <p>-t "text" – устанавливает заголовок окна. Заголовком окна будет текст, указанный в кавычках. Если текст не содержит пробелы, то кавычки допускается не ставить;</p> <p>--enable-logging – обеспечивает вывод некоторых ошибок в консоль.</p> <p>Ключи для настройки положения и размера окна клиента:</p> <p>-x [число] - устанавливает координату X (в пикселях) в диапазоне от 0 (координата левого края монитора) до Xmax (координата правого край монитора, значение определяется разрешением экрана). Например, -x 100 обеспечит смещение от левого края монитора вправо на 100 пикселей;</p> <p>-y [число] - устанавливает координату Y (в пикселях) в диапазоне от 0 (координата верхнего края монитора) до Ymax (координата нижнего края монитора, значение определяется разрешением экрана). Например, -y 50 обеспечит смещение от верхнего края монитора вниз на 50 пикселей;</p> <p>-w [число] - устанавливает ширину окна (в пикселях);</p> <p>-h [число] - устанавливает высоту окна (в пикселях);</p> <p>Если данные параметры не заданы, то окно открывается в верхнем левом углу с наибольшими шириной и высотой.</p> <p>Для переключения в полноэкранный режим можно воспользоваться клавишей F11.</p> <p>-f - включает полноэкранный режим. В полноэкранном режиме клавиша F11 не работает;</p> <p>Ключи, которые будут работать только в случае если выбран тип транслятора HMI v2:</p> <p>-sp -p "имя принтера в системе" - устанавливает принтер по умолчанию;</p> <p>-s "путь к месту хранения" - устанавливается папка для сохранения файлов по умолчанию, например, при работе с</p>

Элемент	Описание
	<p>журналом сообщений. Если путь к файлу не содержит пробелы, то кавычки допускается не ставить;</p> <p>-с - открывает конфигурационное окно, позволяющее установить папку для сохранения файлов по умолчанию, принтер по умолчанию;</p> <p>-m [число] - задает число мониторов, на которых необходимо открыть окна клиента визуализации. Если к компьютеру подключено меньшее количество мониторов, чем указано в настройке, то лишние окна клиента визуализации открываться не будут. Окно авторизации пользователя будет появляться только на одном мониторе - на том, который указан первым в операционной системе. В процессе работы оператор может в каждом мониторе осуществлять независимые действия (открывать окна, работать с графиками, журналами и т.п.). Например, -m 2 - обеспечит запуск клиента визуализации на двух мониторах.</p> <p>--noframe создает окно без рамки и заголовка. Может применяться, чтобы создавать полноэкранное окно, растянутое на несколько мониторов, в этом случае надо указать такие опции (пример для 2 мониторов FullHD):</p> <p style="text-align: center;">-l 0 -t 0 -w 3840 -h 1080 --noframe</p>
Запустить клиент	По нажатию на кнопку откроется клиент визуализации, разработанный компанией "МПС Софт".
Web сервер	
Разрешить доступ по HTTP	Определяет необходимость использование протокола HTTP для передачи данных между сервером и клиентом
Порт web сервера для HTTP	Указывается порт для WEB-сервера. По умолчанию используется 8043
Разрешить доступ по HTTPS	Определяет возможность использовать протокол HTTPS. Для замены сертификата необходимо подменить в папке <i>"c:\Program Files\MPSSoft\MasterSCADA 4D RT 1.2\nginx\conf"</i> файлы <i>server.crt</i> и <i>server.key</i>

Элемент	Описание
Порт web сервера для HTTPS	Указывается порт для WEB-сервера. По умолчанию используется 443
Поддерживаемые протоколы HTTPS	Определяет поддерживаемые протоколы шифрования. По умолчанию свойство не задано. Если указано, то строка применяется, вставляется в <i>nginx-mplc.conf</i> .
Импортировать сертификат	<p>Позволяет импортировать сертификат ssl. После нажатия на кнопку открывается стандартное окно Windows, в котором нужно указать путь к папке, которая содержит два файла: <i>server.crt</i> и <i>server.key</i></p> <p>Если файлы в папке есть, то после нажатия а кнопку Да, происходит их импорт (передается текст файлов сервису, далее файлы копируются в папку настроек сервиса), если файлы не найдены или произошла какая-либо ошибка, то импорт не производится.</p> <p>Если при старте сервиса в рабочей папке нет файлов, то они копируются из папки установки исполнительной системы ("c:\Program Files\MPSSoft\MasterSCADA 4D RT 1.2\nginx\conf")</p>
Сохранить и перезапустить процессы	Останавливает ранее запущенные процессы и запускает их вновь. Количество запускаемых процессов с именем <i>mplc</i> соответствуют значению, введенному в поле Количество экземпляров <i>mplc</i> .
Сохранить	Сохраняет введенные настройки.
Отмена	Закрывает окно настроек, введенные изменения не сохраняются.

10.1.6.2. РАБОТА С ПРОЦЕССОМ MPLC

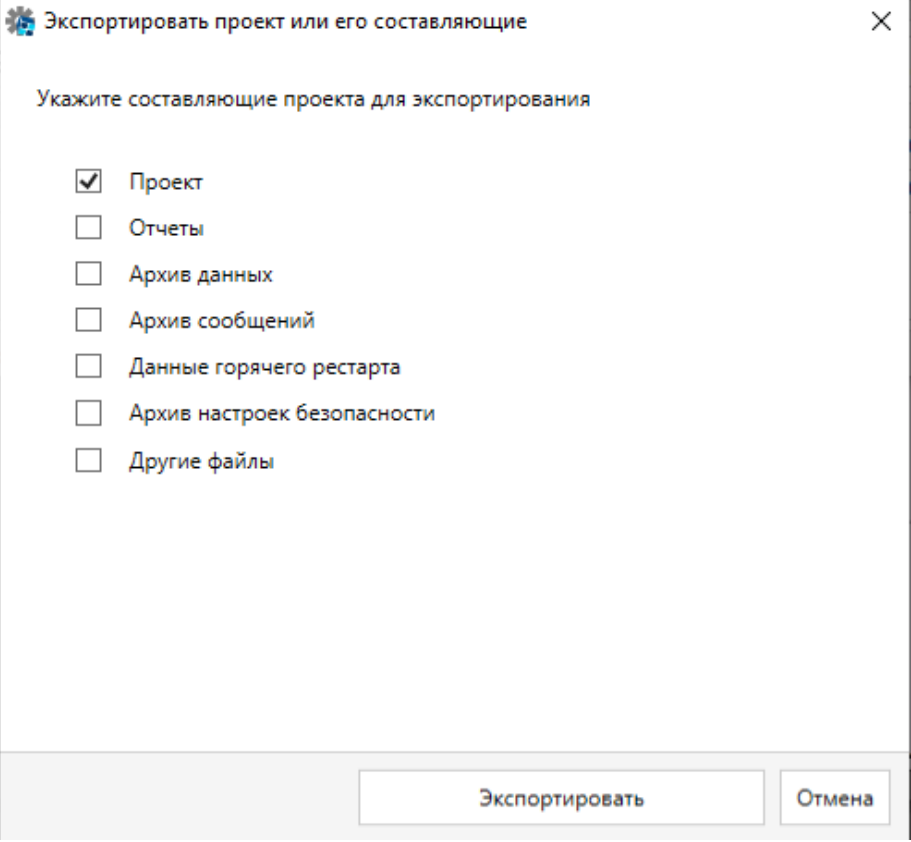
Процесс *mplc* – основной процесс исполнительной системы.

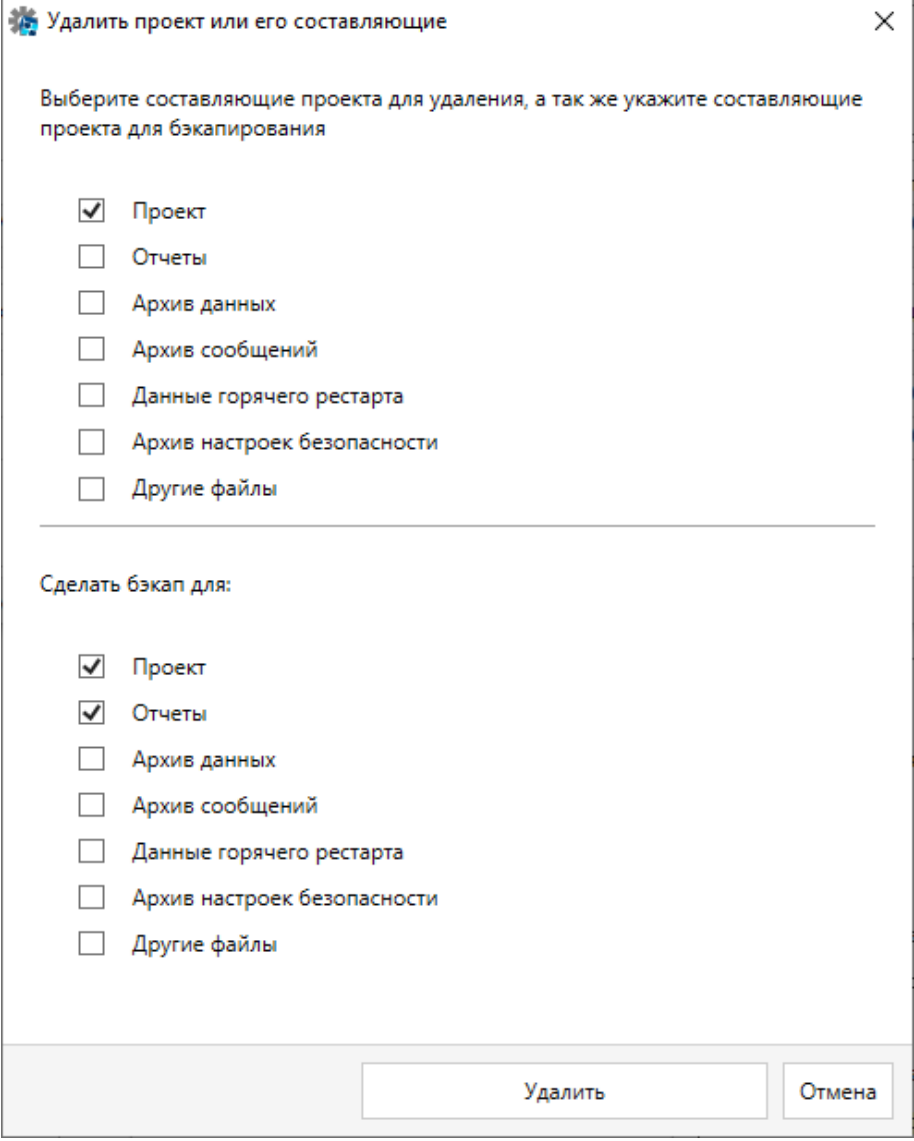
Если в приложении MasterSCADA 4D Monitor в таблице процессов выделить *mplc*, то откроются дополнительные строки, содержащие элементы управления:

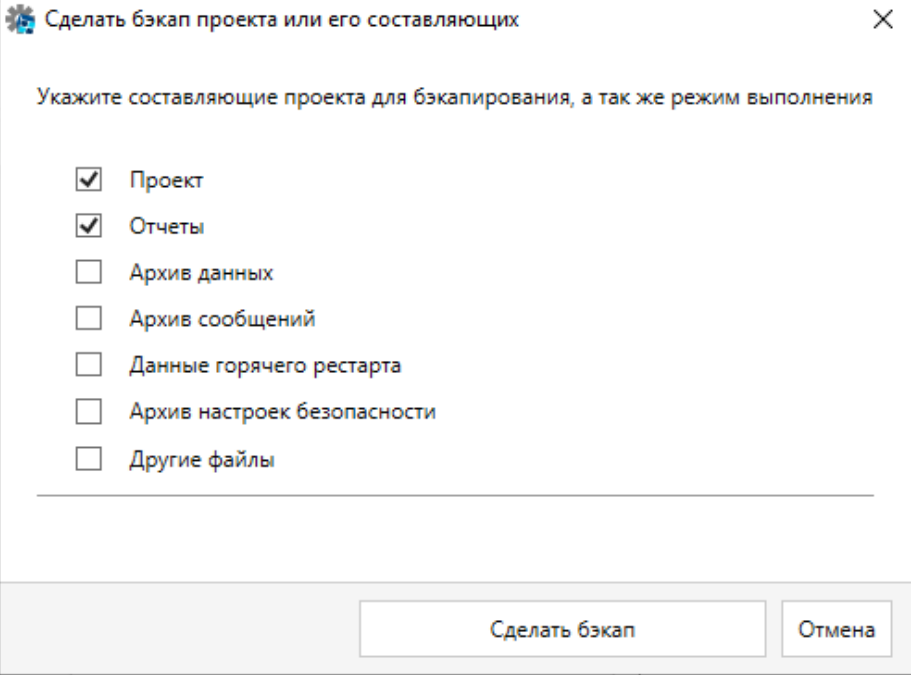
ID процесса	Процесс	Номер	Состояние	Проект	Время последнего запуска	Время последней остановки	Последний код возврата
→	mpic	0	Остановлен	Мониторинг трубопровода (АРМ 1)	12:14:10 29.09.2020	12:23:38 29.09.2020 за 13сек	100 Расшифровка

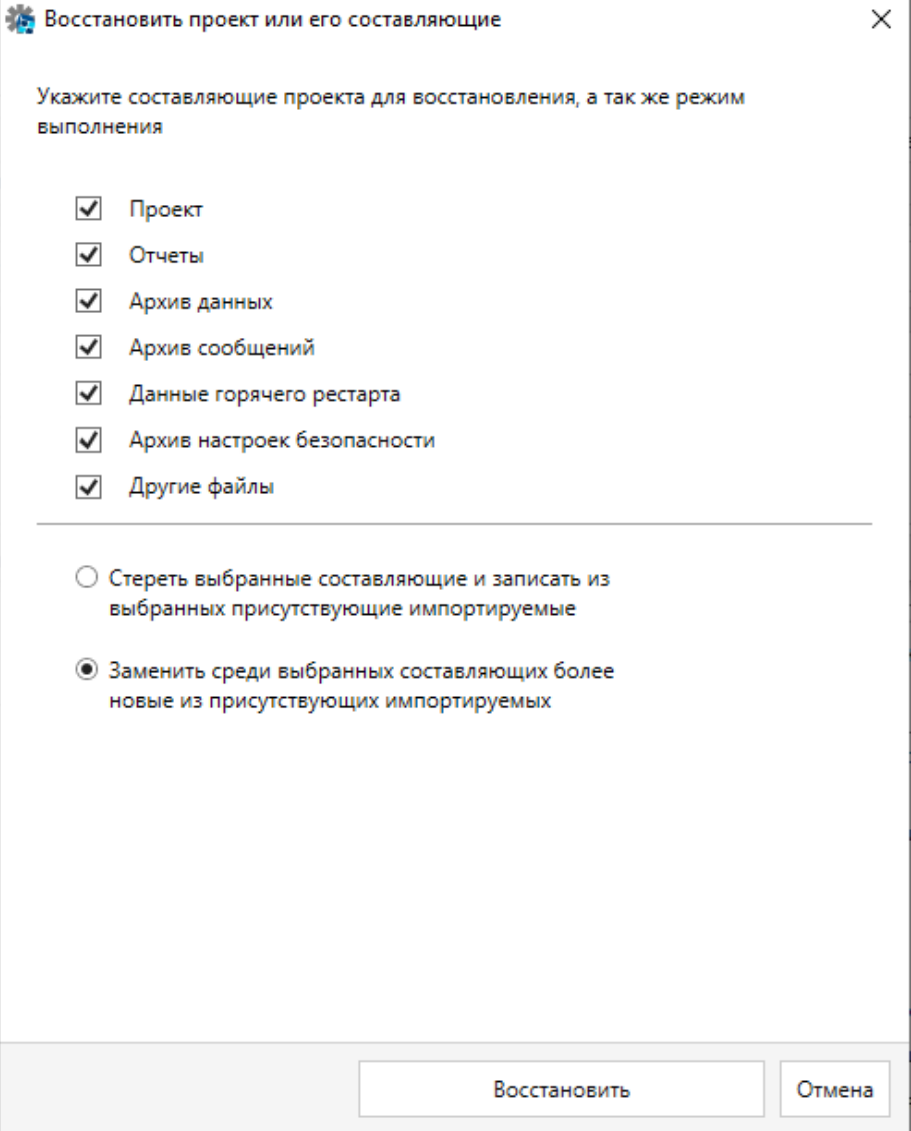
Описание элементов управления:

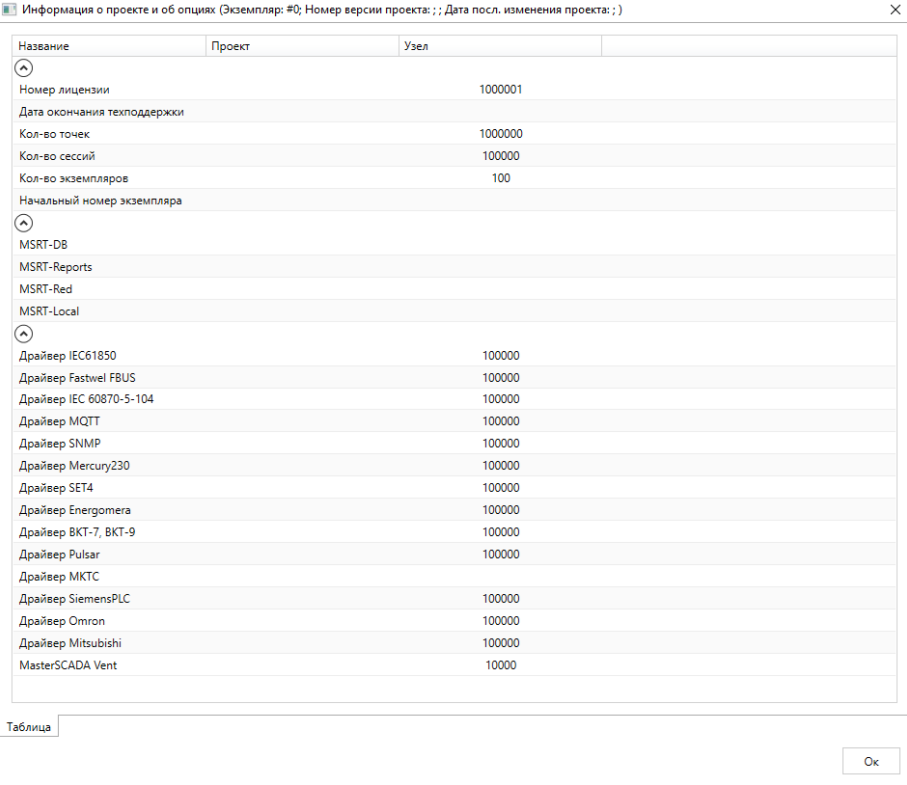
Элемент	Описание
Импорт проекта	Кнопка позволяет осуществлять импорт скомпилированного проекта для запуска в среде исполнения. Подробное описание в разделе Ручная загрузка проекта в среду исполнения. Кнопка активна только если все процессы остановлены. При нажатии на кнопку откроется диалоговое окно, аналогичное окну, которое открывается при нажатии на кнопку Восстановить. Окно позволит выбрать, какие части проекта необходимо импортировать. А также определить необходимо ли стереть все содержимое рабочей папки и заменить на содержимое папки импорта, или заменить только те, что новее в папке импорта.
Экспорт проекта	Кнопка позволяет экспортировать загруженный проект в какое-либо место, указанное пользователем. После нажатия на кнопку открывается диалоговое окно, в котором необходимо выбрать, какие компоненты проекта необходимо экспортировать.

Элемент	Описание
	 <p>После нажатия на кнопку Экспортировать откроется стандартное диалоговое окно Windows , с помощью которого можно выбрать место экспорта проекта.</p>
Удалить проект	Кнопка позволяет удалять загруженный проект или отдельные его части. Также при удалении позволяет сделать бекап загруженного проекта. После нажатия на кнопку откроется диалоговое окно:

Элемент	Описание
	 <p>Удалить проект или его составляющие</p> <p>Выберите составляющие проекта для удаления, а так же укажите составляющие проекта для бэкапирования</p> <ul style="list-style-type: none"><input checked="" type="checkbox"/> Проект<input type="checkbox"/> Отчеты<input type="checkbox"/> Архив данных<input type="checkbox"/> Архив сообщений<input type="checkbox"/> Данные горячего рестарта<input type="checkbox"/> Архив настроек безопасности<input type="checkbox"/> Другие файлы <p>Сделать бэкап для:</p> <ul style="list-style-type: none"><input checked="" type="checkbox"/> Проект<input checked="" type="checkbox"/> Отчеты<input type="checkbox"/> Архив данных<input type="checkbox"/> Архив сообщений<input type="checkbox"/> Данные горячего рестарта<input type="checkbox"/> Архив настроек безопасности<input type="checkbox"/> Другие файлы <p>Удалить Отмена</p>
Бэкап проекта	Выполняет бэкап проекта. После нажатия на кнопку откроется диалоговое окно, которое позволит выбрать для каких частей проекта необходимо выполнить бэкап.

Элемент	Описание
	 <p>Сделать бэкап проекта или его составляющих</p> <p>Укажите составляющие проекта для бэкапирования, а так же режим выполнения</p> <ul style="list-style-type: none"><input checked="" type="checkbox"/> Проект<input checked="" type="checkbox"/> Отчеты<input type="checkbox"/> Архив данных<input type="checkbox"/> Архив сообщений<input type="checkbox"/> Данные горячего рестарта<input type="checkbox"/> Архив настроек безопасности<input type="checkbox"/> Другие файлы <p>Сделать бэкап Отмена</p> <p>После нажатия на кнопку Сделать бэкап. Рядом с рабочей папкой создается папка <i>Backups</i>.</p>
Восстановить проект	<p>Позволяет восстановить проект из папки <i>Backups</i>. После нажатия на кнопку откроется диалоговое окно, которое позволит выбрать, какие части проекта необходимо восстановить. А также определить необходимо ли стереть все содержимое рабочей папки и заменить на содержимое бекапа, или заменить только те, что новее в папке <i>Backups</i>.</p>

Элемент	Описание
	 <p>Восстановить проект или его составляющие</p> <p>Укажите составляющие проекта для восстановления, а так же режим выполнения</p> <ul style="list-style-type: none"><input checked="" type="checkbox"/> Проект<input checked="" type="checkbox"/> Отчеты<input checked="" type="checkbox"/> Архив данных<input checked="" type="checkbox"/> Архив сообщений<input checked="" type="checkbox"/> Данные горячего рестарта<input checked="" type="checkbox"/> Архив настроек безопасности<input checked="" type="checkbox"/> Другие файлы <hr/> <ul style="list-style-type: none"><input type="radio"/> Стереть выбранные составляющие и записать из выбранных присутствующие импортируемые<input checked="" type="radio"/> Заменить среди выбранных составляющих более новые из присутствующих импортируемых <p>Восстановить Отмена</p> <p>Проект восстановится после нажатия на кнопку Восстановить</p>
Инфо о проекте и опциях	Кнопка открывает окно, в котором отображается дополнительная информация о проекте и о возможностях, установленной в узел исполнительной системы.

Элемент	Описание
	 <p data-bbox="550 1137 1437 1256">В таблице перечислены имеющиеся опции, в столбце Узел указываются доступные точки, а в столбце Проект указываются используемые.</p>
Дамп mrlc	Позволяет сделать дамп процесса независимо от отчета об ошибках. После нажатия на кнопку, открывается диалоговое окно Windows, в котором можно выбрать место сохранения дампа.
Остановить процесс/Запустить процесс	Кнопка останавливает и запускает процесс mrlc.
Запустить клиент	Кнопка запускает клиент визуализации.

10.1.7. ЗАПУСК ИСПОЛНИТЕЛЬНОЙ СИСТЕМЫ

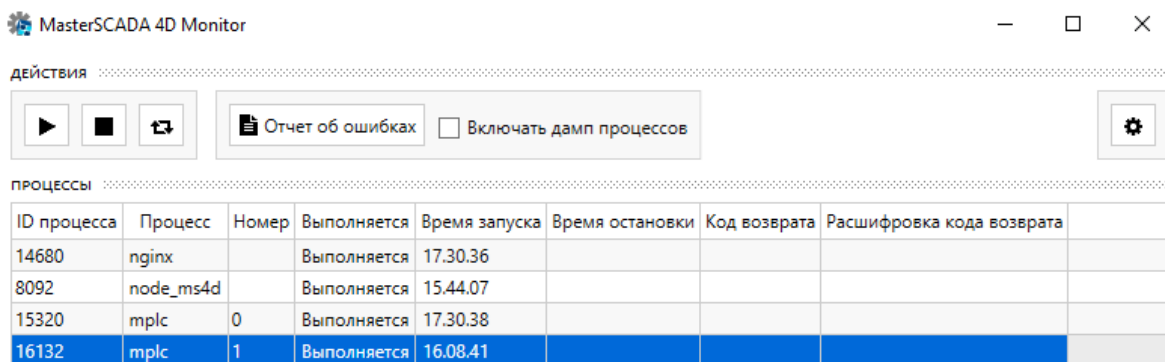
Если исполнительная система установлена как независимое приложение, то при старте операционной системы, по умолчанию, среда исполнения запускается автоматически.

Среду исполнения можно запустить также при помощи приложения MasterSCADA 4D Monitor, либо при помощи ярлыков в меню Пуск – MasterSCADA 4D RT.

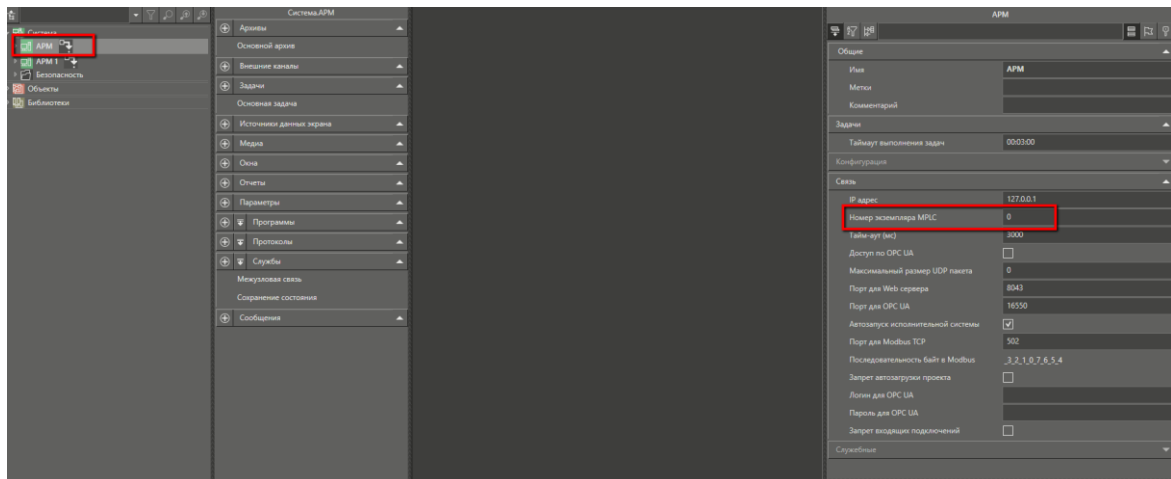
10.1.8. ЗАПУСК НЕСКОЛЬКИХ УЗЛОВ НА ОДНОМ КОМПЬЮТЕРЕ

Для запуска нескольких узлов на одном компьютере, работающем под ОС Windows, можно воспользоваться приложением MasterSCADA 4D Monitor, в настройках которого предварительно устанавливается количество экземпляров исполнительной системы, которые требуется запустить одновременно. При этом в дереве системы также должно быть создано необходимое количество узлов. Затем в настройках узлов в свойстве Номер экземпляра mpls следует установить номера процессов, которые указаны в приложении MasterSCADA 4D Monitor.

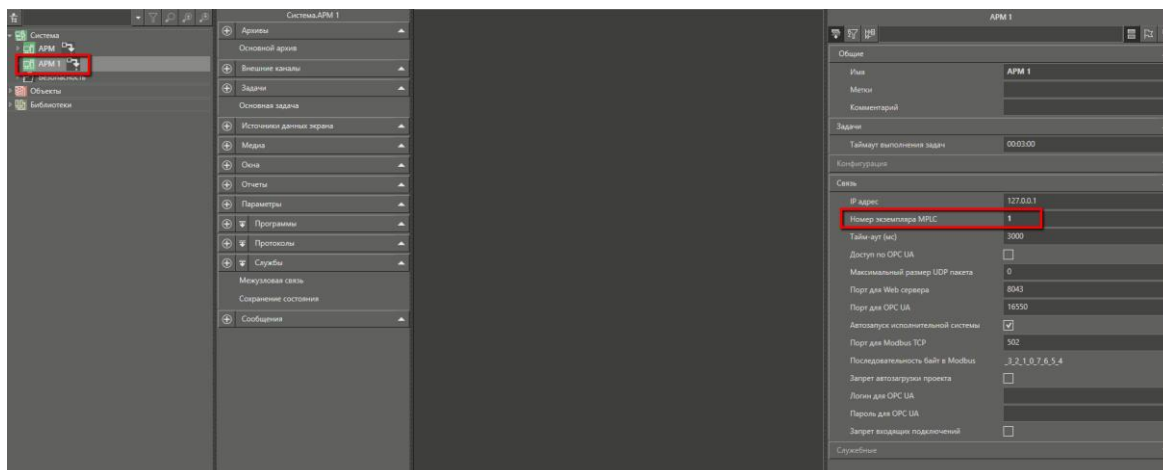
Вид MasterSCADA 4D Monitor:



Настройки для АРМ, который должен запускаться в нулевом процессе:



Настройки АРМ 1, который должен запускаться в первом экземпляре:

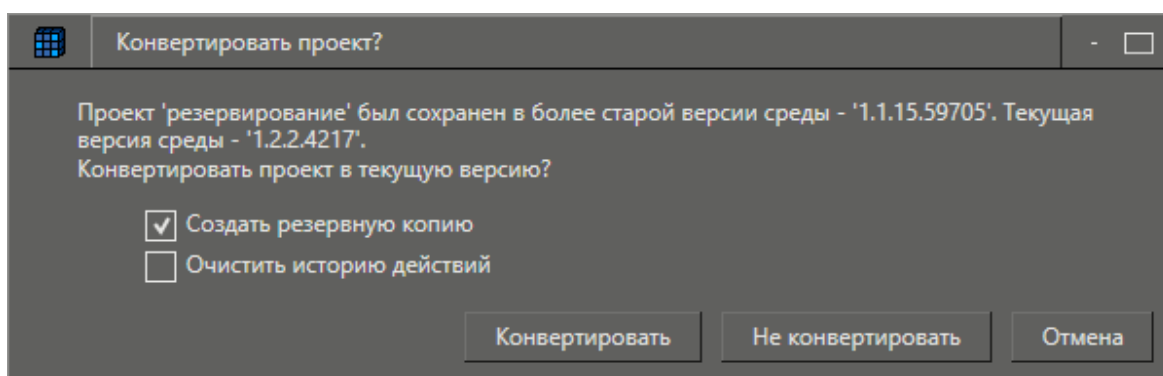


Загрузить проект в среды исполнения можно как автоматически, так и вручную.

10.1.9. ОБНОВЛЕНИЕ ПРОЕКТА В НОВОЙ ВЕРСИИ СРЕДЫ ИСПОЛНЕНИЯ

Если среда исполнения в устройстве обновлена до новой версии, например, до этого была установлена версия Y.X, а затем установили версию Y.X+1, то необходимо также обновить и работающий в среде исполнения проект.

Для этого исходный проект, созданный в среде разработки версии Y.X, необходимо открыть в среде разработки, соответствующей новой версии среды исполнения. При открытии такого проекта появится окно:



Далее следует нажать кнопку Конвертировать и дождаться окончания работы процесса конвертации.

Затем требуется выполнить проверку проекта в режиме Эмуляции, чтобы убедиться в том, что все части системы работают корректно.

После этого необходимо загрузить проект в обновленную среду исполнения автоматически (рекомендуется) либо вручную.

Важно! Архивы, накопленные в устройстве ранее, будут сохранены. Для дополнительной надежности можно перед обновлением сделать резервную копию рабочей папки среды исполнения.

10.1.10. ВЫБОР ОПЦИЙ СРЕДЫ ИСПОЛНЕНИЯ

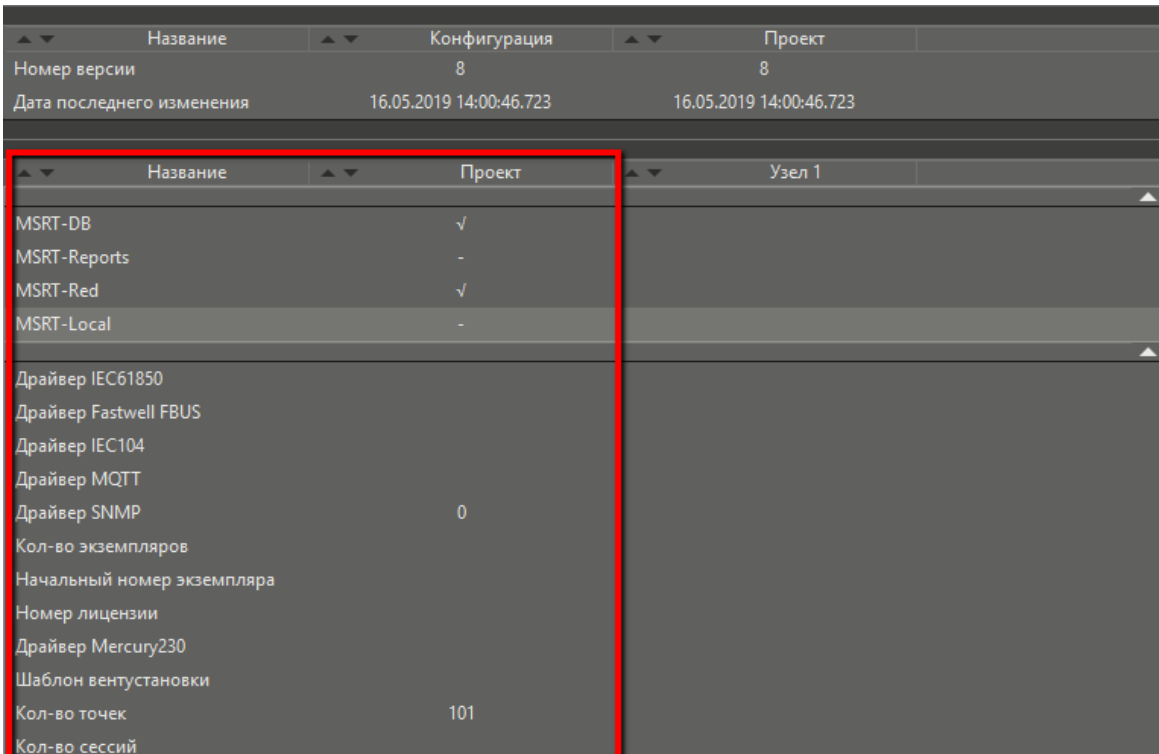
Среда разработки позволяет разработать любой проект любой сложности, а демоверсия среды исполнения дает возможность отладить все функции разработанного проекта.

После того как проект разработан и отлажен, возникает вопрос о том, как выбрать коммерческую версию среды исполнения: какие протоколы уже входят в базовый комплект, а какие необходимо включить в счет поставки дополнительно, какое количество точек будет задействовано в проекте, какие дополнительные опции необходимы.

Для каждого узла, добавленного в проект, должна быть приобретена своя исполнительная система.

Получение списка опций в среде разработки

1. Подключить среду разработки к работающей среде исполнения, например, к демоверсии, входящей в состав среды разработки.
2. Дважды нажать левой кнопкой мыши на узел, для которого необходимо получить список опций, при этом в панели вкладок откроется панель управления узлом.
3. Перейти на вкладку Конфигурация - в нижней таблице, в столбце Проект будет отмечено, какие опции необходимы.



Название	Конфигурация	Проект
Номер версии	8	8
Дата последнего изменения	16.05.2019 14:00:46.723	16.05.2019 14:00:46.723
Название	Проект	Узел 1
MSRT-DB	√	
MSRT-Reports	-	
MSRT-Red	√	
MSRT-Local	-	
Драйвер IEC61850		
Драйвер Fastwell FBUS		
Драйвер IEC104		
Драйвер MQTT		
Драйвер SNMP	0	
Кол-во экземпляров		
Начальный номер экземпляра		
Номер лицензии		
Драйвер Mercury230		
Шаблон вентустановки		
Кол-во точек	101	
Кол-во сессий		

Получение списка опций в MasterSCADA 4D Monitor

1. Открыть сервис MasterSCADA 4D Monitor
2. Выделите процесс mrlc - откроется дополнительная информация.
3. Нажмите кнопку Инфо о проекте и опциях - откроется окно, в котором отображаются опции, необходимые для корректной работы проекта - столбец Проект, и опции доступны в данной исполнительной системе - столбец Узел

Информация о проекте и об опциях (Экземпляр: #0; Номер версии проекта: ;; Дата посл. изменения проекта: ;)

Название	Проект	Узел
Номер лицензии		1000001
Дата окончания техподдержки		
Кол-во точек		1000000
Кол-во сессий		100000
Кол-во экземпляров		100
Начальный номер экземпляра		
MSRT-DB		
MSRT-Reports		
MSRT-Red		
MSRT-Local		
Драйвер IEC61850		100000
Драйвер Fastwel FBUS		100000
Драйвер IEC 60870-5-104		100000
Драйвер MQTT		100000
Драйвер SNMP		100000
Драйвер Mercury230		100000
Драйвер SET4		100000
Драйвер Energomera		100000
Драйвер BKT-7, BKT-9		100000
Драйвер Pulsar		100000
Драйвер MKTC		
Драйвер SiemensPLC		100000
Драйвер Omron		100000
Драйвер Mitsubishi		100000
MasterSCADA Vent		10000

Таблица

Ok

Какие точки учитываются при подсчете

Точка - это задействованный канал, какого-либо протокола, а также все каналы группы Внешние каналы. Задействованным каналом считается тот, который имеет связи или архивируются.

Если протокол лицензируется по количеству точек, то они считаются по такому же правилу.

Если среда разработки не подключена к среде исполнения, то достоверно определить количество задействованных каналов невозможно, поэтому подсчет точек следует производить только при установленной связи со средой исполнения.

Важно! В случае возникновения сложностей с выбором версии среды исполнения, пришлите информацию о необходимых опциях и количестве точек, полученную в среде разработки или в MasterSCADA 4D Monitor по адресу sales@masterscada.ru

10.1.11. КОДЫ ОШИБОК СРЕДЫ ИСПОЛНЕНИЯ

В случае возникновения неисправности, исполнительная системы выдает код ошибки. В исполнительной системе для ОС Windows код ошибки отображается в приложении MasterSCADA 4D Monitor. В других версиях исполнительных систем подобная информация попадает в лог-файлы, которые можно получить, например, сформировав отчет об ошибках в среде разработки, которая подключена к среде исполнения.

Код ошибки	Текст	Примечания
1	Ключ не найден	
Ошибки, выдаваемые в процессе работы (при этих ошибках автоматически формируется дамп памяти)		
2	Аварийное завершение	
3	Ошибка памяти	
4	Ошибка интерпретатора кода	
5	Ошибка интерпретатора кода	
6	Зависание задачи	
Ошибки, выдаваемые при загрузке и старте конфигурации		
7	Превышение допустимого объема памяти	
-1	Ошибка конфигурации	
-5	Ошибка сети	
-6	Ключ защиты не обнаружен	

Код ошибки	Текст	Примечания
-7	Используется неподдерживаемый протокол	
-8	Ошибка открытия основного UDP-сокета	
-9	Нехватка памяти	
-10	Ошибка создания потока	
Ошибки ключа		
Значения от -101 до -208		Ошибки формируются ключом защиты. Для решения проблемы обратитесь в техническую поддержку.
Завершение работы		
-999	Принудительно остановлено	Процесс удален ("убит").
0	Остановлено	Процесс остановлен самопроизвольно.
100	Остановлено	

10.1.12. ПАНЕЛЬ УЗЛА

Панель узла служит для работы с узлом в режиме исполнения, для подготовки проекта к загрузке в среду исполнения, расположенную на другом устройстве, а также для выбора опций, необходимых для корректной работы проекта.

Панель можно открыть, нажав дважды левой кнопкой мыши на узел в дереве системы, либо выбрав в контекстном меню узла пункт Открыть узел.

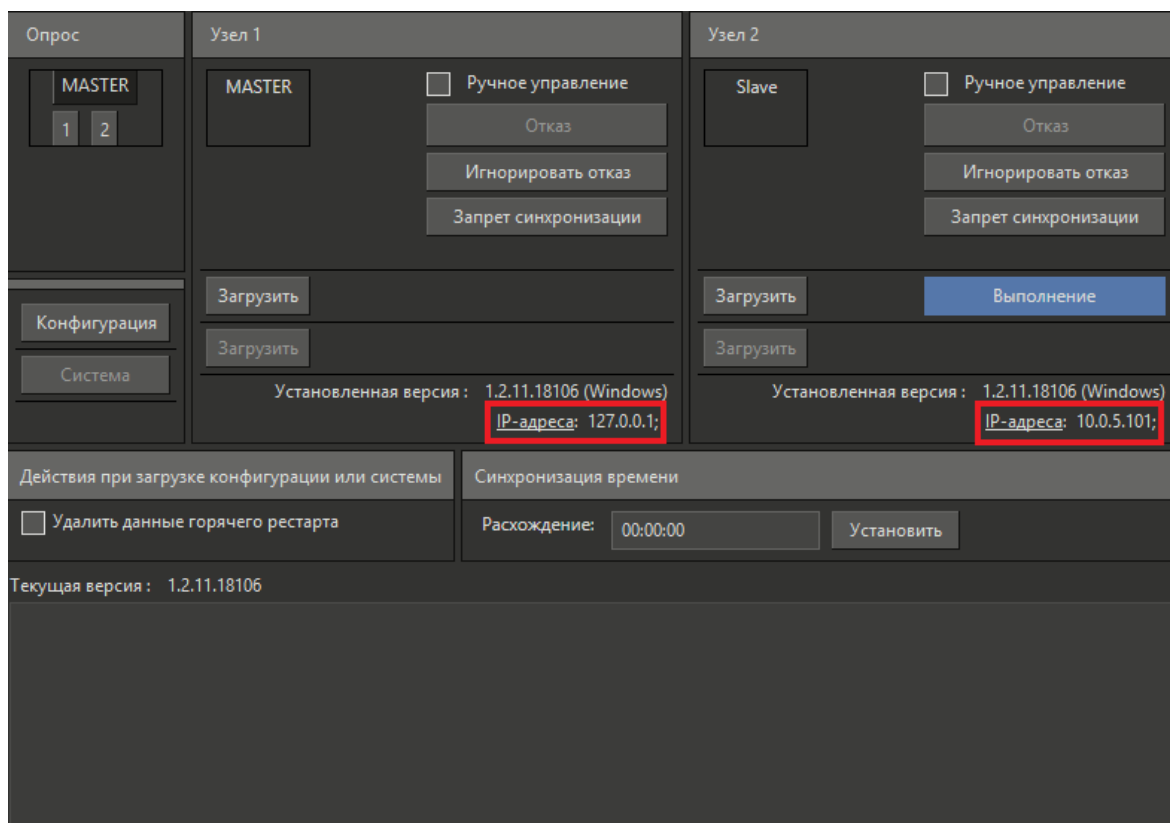
Панель содержит две вкладки:

- Управление
- Конфигурация

10.1.12.1. ВКЛАДКА УПРАВЛЕНИЯ УЗЛОМ

Вкладка панели узла Управление (далее Панель управления узлом) предоставляет возможность оператору в процессе работы проекта получать информацию о состоянии узла и о загруженном в него программном обеспечении. Это особенно необходимо для диагностики и управления узлами при использовании службы резервирования. Панель управления узлом открывается в среде разработки, подключенной к среде исполнения, по двойному нажатию левой кнопкой мыши на узел.

Вид панели управления (при использовании службы резервирования):



В случае если узел не резервируется, на панели управления находится только одна группа Узел, а также отсутствует группа Опрос.

Элементы управления:

Название	Описание
Группа Опрос	Определяет, какой из двух резервируемых узлов опрашивается средой разработки, значения которого отображаются в интерфейсе среды разработки.

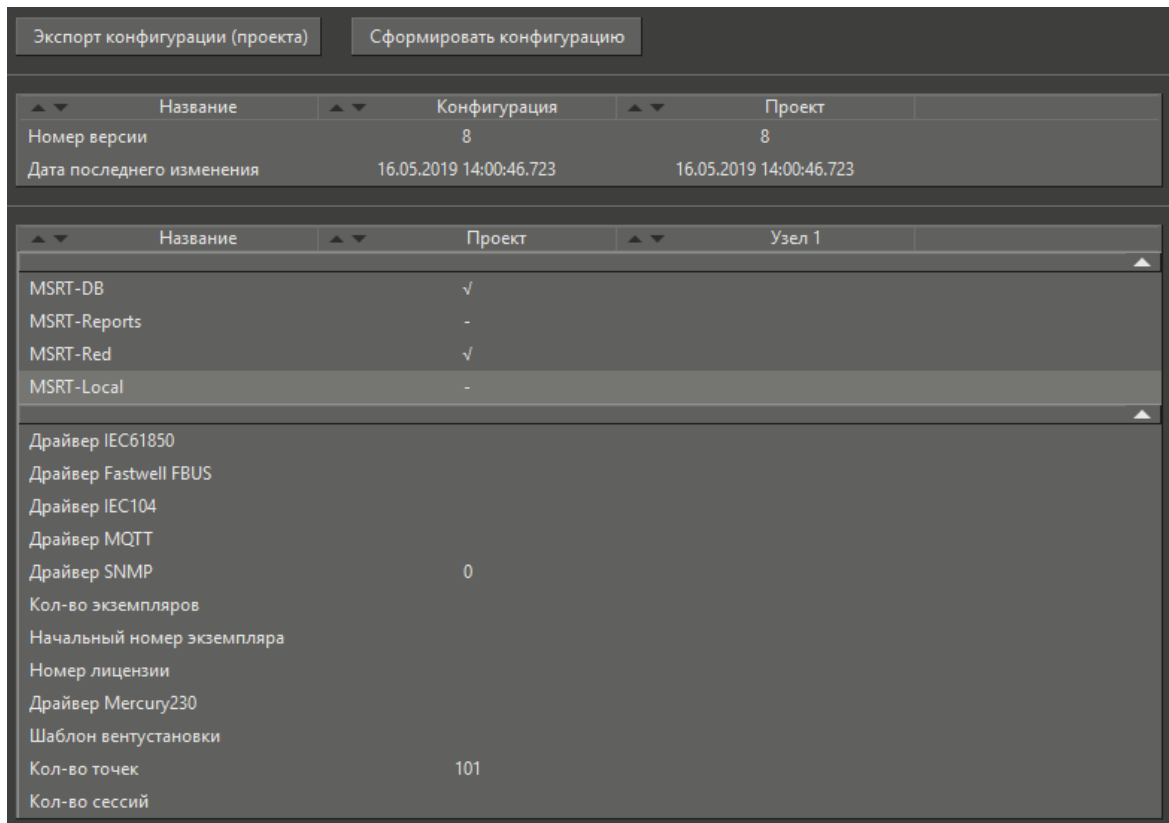
Название	Описание
Кнопка Конфигурация	<p>После нажатия кнопки происходит последовательная загрузка конфигурации проекта в оба узла по очереди. Сначала обновится тот узел, который работает в режиме Резервный, затем произойдет переключение на него и обновится другой узел. При подключении среды разработки к узлу, проверяется конфигурация, загруженная в узел. Если конфигурация в среде разработки отличается от конфигурации, загруженной в узел, то в группе этого узла справа от кнопки появится текст "Несоответствие". Если конфигурация в среде разработки отличается от конфигурации, загруженной в узел, то в группе этого узла справа от кнопки появится текст "Несоответствие"</p>
Кнопка Система	<p>После нажатия на кнопку происходит последовательная загрузка исполнительной системы в оба узла по очереди. Сначала обновится тот узел, который работает в режиме Резервный, затем произойдет переключение на него и обновится другой узел. При подключении среды разработки к узлу проверяется версия ПО, загруженная в узел. Если версия среды разработки отличается от версии исполнительной системы, загруженной в узел, то в группе этого узла справа от кнопки появится текст "Несоответствие".</p>
Группа Узел 1 (Узел 2)	<p>В группе отображается информация о состоянии узла, а также предоставляется возможность управления им. Если установить флаг Ручное, то нажатием/отжатием кнопки Отказ можно изменить статус программного отказа контроллера. В случае программного или аппаратного отказа контроллера появляется надпись "Отказ". После нажатия на кнопку Игнорировать отказ узел может остаться в режиме основного. Кнопка Запрет синхронизации отключает синхронизацию данных между основным и резервным узлами. Рисунок слева обозначает, в каком режиме находится данный узел: если узел основной, то его изображение подсвечивается и находится на переднем плане; если резервный, то остается темным на заднем плане. .</p>

Название	Описание
	<p>Зеленая подсветка означает, что с узлом установлено соединение. Красная подсветка означает, что соединение с узлом отсутствует. Кнопки Загрузить, находящиеся напротив кнопок Конфигурация и Система, позволяют загрузить проект (конфигурацию) в узел и исполнительную систему соответственно. Загрузка произойдет только в текущий узел. Кнопки Откат загружают предыдущую версию проекта и ПО. Также отображается информация об установленной версии исполнительной системы, а также указываются IP-адреса устройств, на которых установлена исполнительная система. Если задано несколько адресов, они все будут перечислены, но активный IP-адрес будет выделен более ярко.</p>
Поле Синхронизация времени	<p>Выдается отклонение системного времени узла, находящегося в режиме основного от системного времени компьютера, на котором установлена среда разработки. По нажатию кнопки Установить происходит принудительная синхронизация системного времени узла с системным временем компьютера.</p>
Поле Действия при загрузке конфигурации или системы	<p>При установленном флаге Удалить данные горячего рестарта работа узла после загрузки новой конфигурации или обновления ПО начнется с начальных значений, заданных разработчиком проекта.</p>
Нижняя часть панели управления	<p>Если загрузка невозможна из-за отсутствия каких-либо модулей, не включенных в состав исполнительной системы, в данном поле появится соответствующее сообщение.</p>

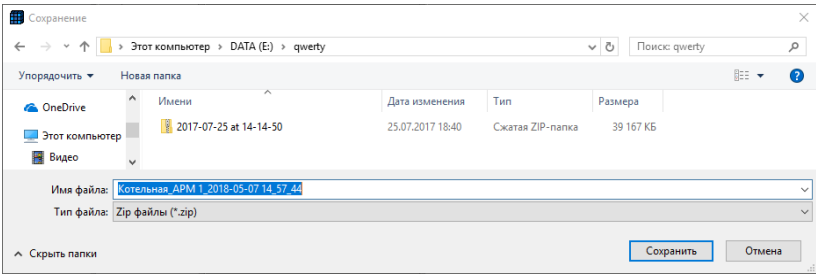
10.1.12.2. ВКЛАДКА КОНФИГУРАЦИЯ

Вкладка панели узла Конфигурация служит для подготовки проекта для загрузки в среду исполнения, расположенную на другом устройстве, а также для выбора опций, необходимых для корректной работы проекта.

Вид вкладки:



Описание:

Элемент	Назначение
<p>Экспорт конфигурации (проекта)</p>	<p>Кнопка позволяет сохранить ранее созданную конфигурацию (т.е. проект, скомпилированный для работы в среде исполнения) в произвольном месте на компьютере. После нажатия на кнопку откроется диалоговое окно Windows, предлагающее выбрать папку для сохранения архива проекта, предназначенного для работы в среде исполнения выбранного узла:</p>  <p>После нажатия на кнопку Сохранить, создастся архив со следующим именем по умолчанию: <i>[название проекта]_[имя узла]_[дата и время экспорта]</i>.</p>

Элемент	Назначение
Сформировать конфигурацию	Кнопка позволяет сформировать конфигурацию (т.е скомпилировать проект для работы в среде исполнения). Проект сохранится в папке <i>c:\Users\[имя пользователя]\AppData\Roaming\MPSSoft\MasterSCADA4DBeta\Debug_[имя проекта]\[имя узла]</i>
Таблица с информацией о проекте	Служит для получения информации о загруженном проекте в среду исполнения, и о разрабатываемом проекте в среде разработки. В столбце Конфигурация показывается номер версии и время последнего изменения проекта, который был скомпилирован и загружен в среду исполнения. В столбце Проект показывается номер версии проекта среды разработки и время его последнего сохранения.
Таблица с информацией об используемых опциях	Служит для определения списка необходимых и доступных опций для корректной работы созданного проекта. Каждая строка отвечает за одну из опций среды исполнения. В столбце Проект показывается, какие опции необходимы для корректной работы проекта. В столбце Узел 1 показывается какие опции присутствуют в используемой среде исполнения. В случае, если используется резервирование, то появится столбец Узел 2, где будет информация о исполнительной системе, установленной на резервном устройстве.

Важно! Информация об опциях и загруженном проекте в среду исполнения доступна только при подключении среды разработки к среде исполнения.

10.1.13. РАБОТА С ФАЙЛОМ ДАННЫХ ДЛЯ ГОРЯЧЕГО РЕСТАРТА

Как правило, данные для горячего рестарта хранятся в файлах *session.bin*, *session2.bin*. Запись в них идет поочередно. При старте сравниваются времена обновления файлов, вначале происходит попытка загрузки из более нового. Если контрольная сумма не совпадает, идет попытка загрузить из другого. Основное место расположения файлов зависит от настроек проекта, а также от типа контроллера, на котором работает исполнительная система. Исключение составляют контроллеры с энергонезависимой памятью, у которых данные хранятся непосредственно в этой памяти.

Часто возникает задача произвести импорт/экспорт данных для горячего рестарта.

Экспорт данных для горячего рестарта в файл для контроллеров с энергонезависимой памятью

В контроллерах с энергонезависимой памятью (например, PLC110, M903) для экспорта данных для горячего рестарта в файл необходимо запустить `mpIc` с опцией `/export`.

После этого в рабочей папке исполнительной системы `/opt/mpIc4` сформируется файл `session.bin`, который затем может быть использован для импорта.

Импорт файла данных для горячего рестарта для всех контроллеров/серверов

Для импорта файла данных для горячего рестарта необходимо скопировать файл `session.bin` в рабочую папку исполнительной системы `/opt/mpIc4/import`. При очередной перезагрузке, необходимые данные считываются из этого файла, после чего он будет удален, а сами данные сохранятся в основном месте хранения.

Для резервирования (backup'a) файла данных для горячего рестарта можно скопировать файл `session.bin` в папку `/opt/mpIc4/backup`. При перезагрузке, данные для горячего рестарта, как обычно, будут считываться из основного места хранения, но, в случае возникновения ошибки чтения, смогут быть получены из этого файла

Удаление файла данных для горячего рестарта

Для удаления файла данных для горячего рестарта необходимо запустить `mpIc` с опцией `/delhr`.

10.1.14. СОЗДАНИЕ БЕКАПА ЧЕРЕЗ КОМАНДНУЮ СТРОКУ

Бекап работающего проекта можно сделать в режиме исполнения через приложение MasterSCADA 4D Monitor. Также это можно сделать при помощи специальной команды MS4DMonitor.Prompt введенной с командную строку в ОС Windows.

Команда содержит 4 аргумента:

- имя выполняемого действия - `backup`
- номер `mpIc` - если запущено на компьютере только одна исполнительная система, то по умолчанию нужно указать 0
- Путь к рабочей папки. По умолчанию `C:\Users\Public\Documents\MPSSoft\MasterSCADA4DRTBeta`, имя самой рабочей папки `Server` указывать не нужно.
- Путь к папке, в которой нужно разместить бекап, например, `C:\Users\User\Documents\wood\TryBackupCmd3`

Пример текста команды:

```
>MS4DMonitor.Prompt backup 0  
C:\Users\Public\Documents\MPSSoft\MasterSCADA4DRTBeta  
C:\Users\User\Documents\wood\TryBackupCmd3
```

Важно! Если путь к папкам содержит пробелы, то их в тексте команды нужно заменить на символ *.

Эту команду можно использовать для формирования бекапа проекта сторонними средствами.

10.2. РАБОТА С КЛИЕНТОМ ВИЗУАЛИЗАЦИИ

В качестве клиента визуализации может выступать браузер, поддерживающий HTML5, а также специальное приложение для ОС Windows или Android, поставляемое компанией "МПС Софт".

Важно! Браузеры Internet Explorer и Microsoft Edge не поддерживаются. Версия браузера Google Chrome должна быть не ниже v.66

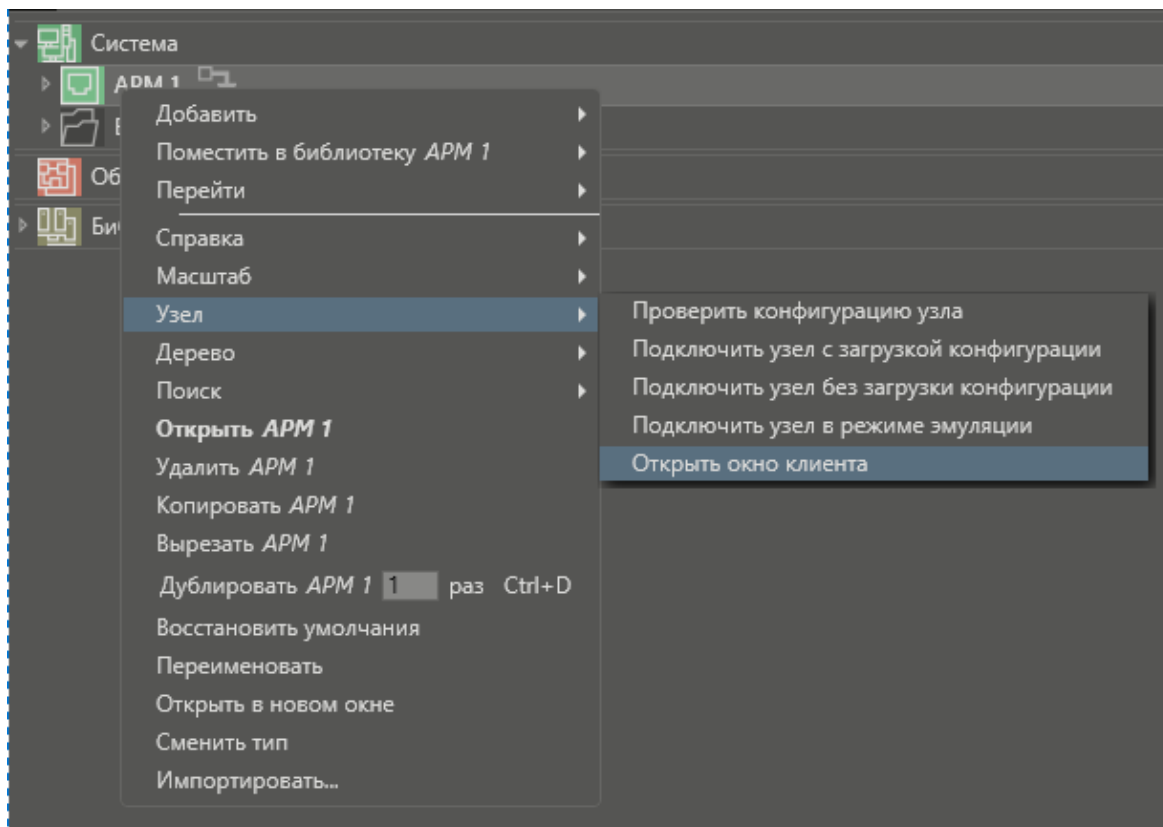
При подключении клиента к среде исполнения загрузится окно, назначенное стартовым при разработке проекта.

При выборе коммерческой версии среды исполнения следует учитывать количество одновременно подключаемых клиентов, а также место запуска клиента (на том же устройстве, что и среда исполнения, или на другом).

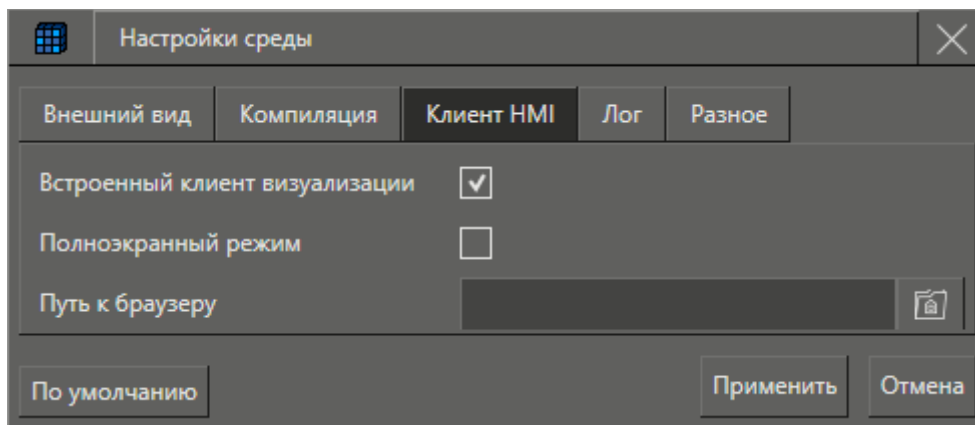
Важно! При подсчете клиентов, подключенных к среде исполнения, их происхождение не имеет значения: это могут быть как специальные приложения компании "МПС Софт", так и браузеры сторонних производителей .

Запуск клиента через среду разработки

Если среда разработки подключена к среде исполнения, то при помощи пункта Узел.Открыть окно клиента контекстного меню любого Узла можно вызвать клиент визуализации.



В качестве клиента откроется то приложение, которое выбрано в настройках среды:



Если установлен флаг Встроенный клиент визуализации, то откроется клиент, разработанный компанией "МПС Софт" – MasterSCADA 4D Client, который для ОС Windows идет в комплекте как со средой разработки, так и со средой исполнения. Если флаг не установлен, то откроется браузер, путь к которому задан в соответствующем поле.

Если необходимо открыть окно клиента на весь экран, то в настройках Шаблона экрана следует включить настройку Полноэкранный режим.

Важно! Свойство Полноэкранный режим в настройках Шаблона экрана влияет только на клиент, открытый из среды разработки.

Подключение клиента без использования среды разработки

Для подключения любого клиента к проекту необходимо указать параметры подключения: IP-адрес устройства, на котором работает среда исполнения; порт TCP/IP, который задан в конфигурационных файлах среды исполнения (по умолчанию, для ОС Windows 8043) и номер экземпляра. Если на одном устройстве запущено несколько исполнительных систем одновременно (актуально для проектов, работающих в облаке), то следует указать, к какому именно экземпляру Исполнительной системы клиент будет подключаться. Номер экземпляра задается в настройках узла.

Запуск стороннего Клиента

Если в качестве клиента используется браузер стороннего производителя, то указанные параметры вводятся в адресную строку: `[IP]:[порт TCP/IP] /[#номер экземпляра]/`.

Если необходимо, чтобы при запуске клиента не появлялось окно авторизации, то можно ввести в адресную строку: `127.0.0.1:8043/index.html?user=sa&password=123`, где 127.0.0.1 - IP-адрес узла, 8043 - TCP-порт, sa - имя оператора, а 123 - его пароль.

Запуск MasterSCADA 4D Client для Windows

Запускать клиент на компьютере с установленной средой исполнения можно при помощи сервиса MasterSCADA 4D Monitor. Кроме того, имеется возможность установить в настройках сервиса флаг Автозапуск клиента, и тогда клиент будет запускаться при старте ОС.

Если среда разработки подключается к уже работающей исполнительной системе, то запуск клиента будет происходить автоматически, при условии, что в настройках узла установлен флаг Автозапуск клиента визуализации, и в качестве клиента в настройках среды выбран Встроенный клиент визуализации. В том случае, когда флаг Автозапуск клиента визуализации снят, клиент может быть запущен при помощи контекстного меню узла Узел.Открыть окно клиента.

На случай запуска клиента на компьютере, на котором не установлены ни среда исполнения, ни среда разработки, предусмотрена установка независимого приложения MasterSCADA 4D Client, и в этом случае параметры запуска указываются в его сервисе.

Работа клиента при резервировании

При первом обращении к резервируемой паре устройств с установленной средой исполнения MasterSCADA 4D RT, клиент должен подключаться к тому узлу, который находится в режиме основного. Если связь с этим узлом будет потеряна, то страница клиента переподключится автоматически к тому узлу, который находится в состоянии основного.

Восстановление подписки на данные

В случае если клиент потеряет связь с исполнительной системой, то он будет периодически пытаться восстановить подключение.

В случае восстановления связи поведение клиента будет разным в версиях HMI v1 и v2:

- v1 - произойдет обновление страницы;
- v2 - переключением произойдет без перезагрузки.

Изменение порта TCP/IP

По умолчанию, для работы клиента и сервера для Windows-версий используется TCP/IP порт 8043, а для большинства контроллеров - порт 80. При необходимости его изменения требуется отредактировать файл *nginx-mplc.conf*, найдя и изменив в его тексте строку 8043 default_server или 80 default_server, в зависимости от типа используемой среды. Если для отладки используется демо-версия Среды исполнения, включенная в комплект Среды разработки, то файл находится по адресу: *c:\Program Files (x86)\MPSSoft\MasterSCADA 4D 1.1\bin\Config\MasterPLC\WIN32\nnginx\conf*. В исполнительной системе для ОС Linux, по умолчанию, файл находится в папке: */opt/mplc4/nginx/conf*.

10.2.1. ВЫБОР ВЕРСИИ HMI

Исполнительная система MasterSCADA 4D способна работать на компьютерах, контроллерах, платах, операционные системы которых обладают разным уровнем возможностей.

Графический интерфейс оператора создается в редакторе HMI, который входит в состав среды разработки MasterSCADA 4D, в режиме исполнения созданные окна открываются в клиенте визуализации.

Разработчики MasterSCADA 4D стремятся достичь следующих целей:

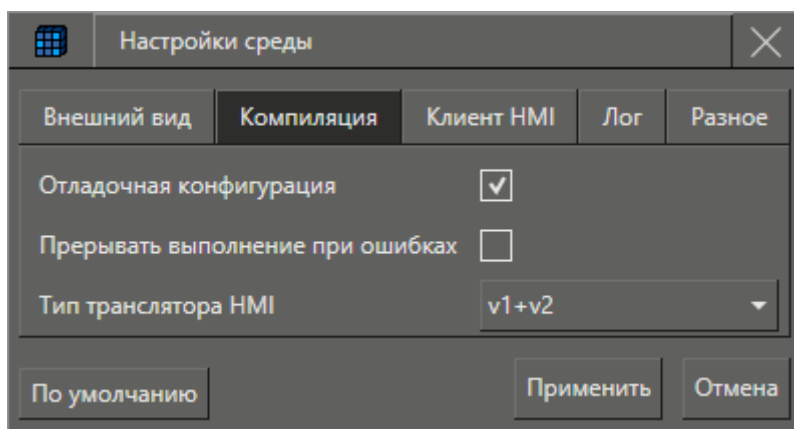
- В программе должны использоваться все возможные новые мировые тренды и технологии;
- Должен обеспечиваться максимально возможный функционал программы даже в самой узкоспециализированной ОС.

К сожалению, добиться этого одновременно или достичь разумного компромисса в настоящее время невозможно, поэтому было принято решение выпустить две версии транслятора HMI:

v1 - которая поддерживается во всех исполнительных системах и браузерах, но в которой, к сожалению, при этом многие современные функции не могут быть реализованы.

v2 - в которой используется более прогрессивная технология, но которая, по этой причине, поддерживается не во всех браузерах. При применении этой версии транслятора рекомендуется в качестве клиента визуализации использовать MasterSCADA 4D Client, а также браузеры Chrome и Firefox.

Выбрать версию, которая будет применяться для подготовки файлов, необходимых для работы клиента визуализации, можно в диалоговом окне Настройка среды во вкладке Компиляция:



В поле Тип транслятора HMI необходимо выбрать номер версии. Возможные варианты:


- v1 - при компиляции формируется только HMI v1;
- v1+v2 - при компиляции формируется обе версии HMI. По умолчанию, в клиенте открывается HMI v1, чтобы получить доступ к HMI v2 необходимо добавить /test в адресной строке браузера или в настройках приложений MasterSCADA 4D Monitor и MasterSCADA 4D Client Monitor;
- v2 - при компиляции формируется только HMI v2.

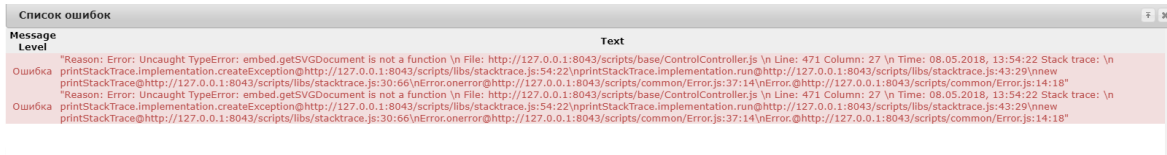
Если MasterSCADA 4D была установлена впервые, то по умолчанию настройка будет v2. Если на компьютере была установлена версия, где использовался транслятор v1, то при установке новой версии появится диалоговое окно, содержащее следующий текст: "У Вас используется устаревший транслятор HMI v1. Перейти на новый транслятор HMI v2? Данную настройку можно изменить в настройках приложения на вкладке Компиляция." Если нажать кнопку Да, то установится значение v2, если Нет, то останется предыдущий вариант. Рекомендуем использовать v2 для создания новых проектов. В дальнейшем поддержка v1 будет завершена.

Отличие версий описано в разделе Создание окон для графического интерфейса операторов.

10.2.2. ОКНО СПИСОК ОШИБОК

Компилятор HMI v1


В случае возникновения ошибок при работе клиента со средой исполнения, в нижнем правом углу клиента отображается круглый индикатор . После нажатия на него левой кнопкой мыши откроется окно, содержащее таблицу из двух столбцов:

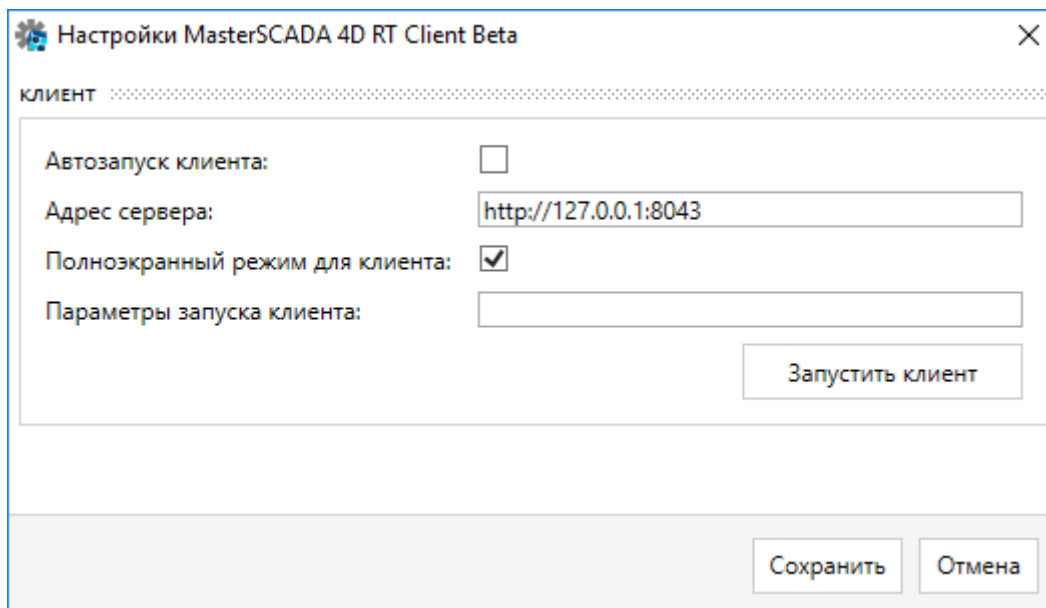


- Message Level – уровень конфликта (ошибка/предупреждение);
- Text – описание конфликта.

В случае если с причиной ошибки не получается разобраться самостоятельно, отправьте скриншот с окном сообщений в службу технической поддержки, приложив к письму текущий проект и описание условий, при которых проявляются данные ошибки (например, при старте, при вводе значений, при получении каких-то данных и др.).

10.2.3. MASTERSCADА 4D CLIENT MONITOR

Если MasterSCADA 4D Client установлен на компьютере не в составе исполнительной системы, а как независимое приложение, разработанное компанией "МПС Софт", то в области уведомлений панели задач появляется значок , при двойном нажатии на который левой кнопкой мыши открывается окно, предназначенное для настройки этого приложения.



Название	Описание
Автозапуск клиента	Если флаг установлен, то клиент запускается автоматически при старте ОС.
Адрес сервера	Задается путь к веб-серверу: <i>[IP]:[порт TCP/IP] [/номер экземпляра]/</i>

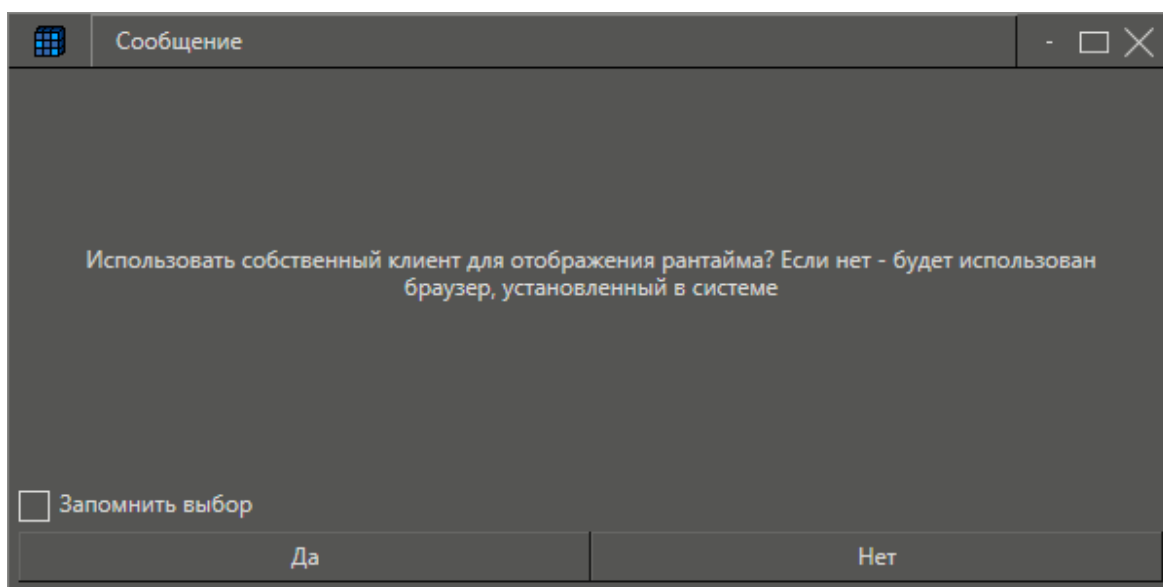
Название	Описание
	<p>Если требуется обеспечить запуск клиента без процедуры авторизации, то следует ввести в адресную строку: <i>127.0.0.1:8043/index.html?user=sa&password=123</i>, где 127.0.0.1 - IP-адрес узла, 8043 - порт, sa - имя оператора, а 123 - его пароль.</p> <p>Если в конце адресной строки добавить /test, и при этом в настройках среды указано Тип транслятора HMI - v1+v2, то откроется клиент визуализации выполненный в версии HMI v2.</p>
<p>Полноэкранный режим для клиента</p>	<p>Если этот флаг установлен, то MasterSCADA 4D Client запускается во весь экран.</p>
<p>Параметры запуска клиента</p>	<p>Могут быть заданы следующие ключи:</p> <p>-d – если ключ установлен, то в процессе работы по кнопке F12 отрываются средства разработчика;</p> <p>-t "text" – устанавливает заголовок окна. Заголовком окна будет текст, указанный в кавычках. Если текст не содержит пробелы, то кавычки допускается не ставить;</p> <p>--enable-logging – обеспечивает вывод некоторых ошибок в консоль.</p> <p>Ключи для настройки положения и размера окна клиента:</p> <p>-x [число] - устанавливает координату X (в пикселях) в диапазоне от 0 (координата левого края монитора) до Xmax (координата правого край монитора, значение определяется разрешением экрана). Например, -x 100, обеспечит смещение от левого края монитора вправо на 100 пикселей.</p> <p>-y [число] - устанавливает координату Y (в пикселях) в диапазоне от 0 (координата верхнего края монитора) до Ymax (координата нижнего края монитора, значение определяется разрешением экрана). Например, -y 50, обеспечит смещение от верхнего края монитора вниз на 50 пикселей.</p> <p>-w [число] - устанавливает ширину окна (в пикселях).</p> <p>-h [число] - устанавливает высоту окна (в пикселях).</p> <p>Если данные ключи не заданы, то окно открывается в верхнем левом углу с наибольшими шириной и высотой. Для</p>

Название	Описание
	<p>переключения в полноэкранный режим можно воспользоваться клавишей F11.</p> <p>-f - включает полноэкранный режим. В полноэкранном режиме клавиша F11 не работает</p> <p>Ключи, которые будут работать только в случае если выбран тип транслятора HMI v2:</p> <p>-sp -p "имя принтера в системе" - устанавливает принтер по умолчанию</p> <p>-s "путь к месту хранения" - устанавливается папка для сохранения файлов по умолчанию, например, при работе с журналом сообщений. Если путь к файлу не содержит пробелы, то кавычки допускается не ставить .</p> <p>-c - открывает конфигурационное окно, позволяющее установить папку для сохранения файлов по умолчанию, принтер по умолчанию</p> <p>-m [число] - задает число мониторов, на которых необходимо открыть окна клиента визуализации. Если к компьютеру подключено меньшее количество мониторов, чем указано в настройке, то лишние окна клиента визуализации открываться не будут. Окно авторизации пользователя будет появляться только на одном мониторе - на том, который указан первым в операционной системе. В процессе работы оператор может в каждом мониторе осуществлять независимые действия (открывать окна, работать с графиками, журналами и т.п.). Например, -m 2 - обеспечит запуск на двух мониторах.</p> <p>--noframe создает окно без рамки и заголовка. Может применяться, чтобы создавать полноэкранный режим, растянутое на несколько мониторов, в этом случае надо указать такие опции (пример для 2 мониторов FullHD):</p> <p>-l 0 -t 0 -w 3840 -h 1080 --noframe</p>
Запустить клиент	По нажатию на кнопку откроется MasterSCADA 4D Client, разработанный компанией "МПС Софт".
Сохранить	Сохраняет введенные настройки.

Название	Описание
Отмена	Закрывает окно настроек, введенные изменения не сохраняются.

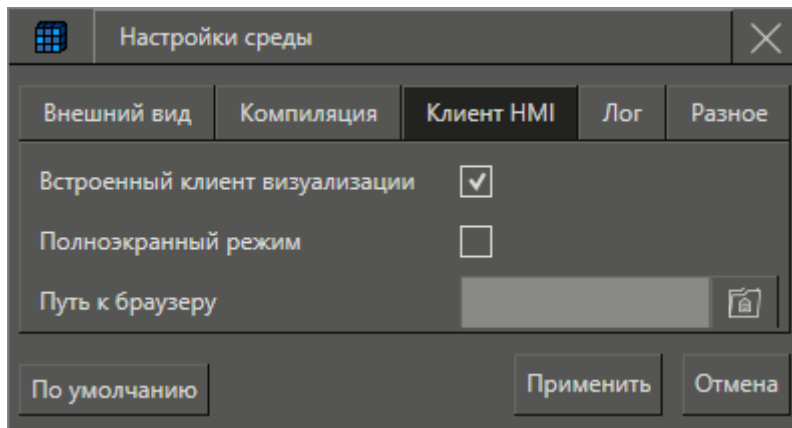
10.2.4. ОСОБЕННОСТИ ПЕРВОЙ АВТОМАТИЧЕСКОЙ ЗАГРУЗКИ ПРОЕКТА

Если узел запускается в первый раз после установки продукта (т.е. до того как выполнено конфигурирование редактора проекта, и, соответственно, пока еще отсутствует файл *%профиль пользователя%\AppData\Roaming\MPSSoft\MasterSCADA4D<версия>\MasterPLCUserSettings.xml*), то открывается следующий диалог:



Инструменты диалога:

- Кнопка Да – для подтверждения, что отображение окон(мнемосхем) в режиме исполнения будет выполняться клиентом визуализации MasterSCADA 4D Client;
- Кнопка Нет – отображение окон(мнемосхем) в режиме исполнения будет выполняться браузером, назначенным в ОС по умолчанию;
- Флаг Запомнить выбор – если этот флаг установлен, то по командам Да и Нет формируется *MasterPLCUserSettings.xml* и в дальнейшем будет использоваться сделанный выбор. Изменить настройку можно в окне Настройки среды:

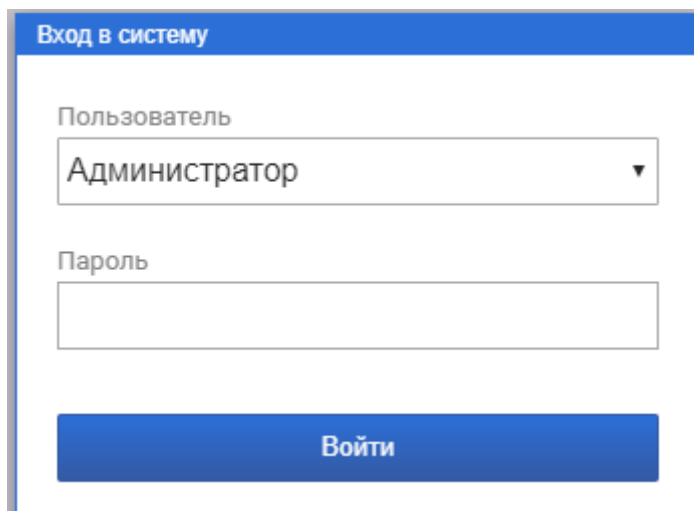


10.2.5. ОКНО АВТОРИЗАЦИИ ПОЛЬЗОВАТЕЛЕЙ

В данном разделе описано окно автори

До тех пор, пока в проекте не созданы пользователи, авторизация не производится и права доступа не проверяются.

После того как в проекте создан хотя бы один пользователь, при запуске узла открывается диалог авторизации:



Для того чтобы начать сессию, необходимо выбрать пользователя в списке Пользователь необходимую позицию (этот список содержит пользователей, созданных в группе Безопасность. Пользователи), указать Пароль выбранного пользователя, и нажать кнопку Войти. Если будет указан неверный пароль, то в диалоге отобразится соответствующее сообщение.

Вход в систему

Пользователь
Администратор ▼

Пароль
.....

**Неуспешная попытка логина:
неверный логин или пароль**

Войти

Для запуска стороннего клиента без использования окна авторизации, необходимо прописать в адресной строке браузера `127.0.0.1:8043/index.html?user=sa&password=123`, где 127.0.0.1 - IP-адрес узла, 8043 - порт, sa - имя оператора, а 123 - его пароль.

Для запуска клиента визуализации, разработанного компанией "МПС Софт", требуется выполнить необходимые настройки в MasterSCADA 4D Monitor, или в MasterSCADA 4D Client Monitor (если используется клиент, установленный независимо от среды исполнения)

Если у пользователя установлено, что при попытке логина необходимо сменить пароль, то после попытки авторизации со старым паролем окно авторизации будет иметь вид:

Вход в систему

<p>Пользователь Петров ▼</p> <p>Пароль</p> <p>Необходимо сменить пароль.</p> <p>Новый пароль <input type="text"/></p> <p>Повторить пароль <input type="text"/></p> <p>Войти</p>	<p>Информация</p> <p>Длительность сессии: 0 Минимальная длина пароля: 0 Количество неповторяемых паролей: 0 Срок действия пароля: 0 Использовать сложный пароль Нет</p> <p>Отмена</p>
--	---

В нем нужно ввести новый пароль, соответствующий ограничениям указанным в поле Информация. Если в поле информация для какого-либо пункта установлено значение 0, то это значит ограничений нет.

Если пользователю не будет предоставлено право Изменение собственного пароля, то в окне появится сообщение:

**Неуспешная попытка смены пароля:
недостаточно прав**

10.2.6. ЗАДАНИЕ ПРИНТЕРА И ПУТИ К СОХРАНЯЕМЫМ ФАЙЛАМ ПО УМОЛЧАНИЮ

Данный раздел посвящен такой настройке приложения клиента, которая позволила бы использовать функции печати, а также сохранения трендов и журналов в файл автоматически (без открытия диалогового окна с параметрами печати и без использования проводника операционной системы).

Важно! Функция поддерживается только в версии HMI v2

Важно! Данная функция используется для сохранения и печати только тех элементов MasterSCADA 4D, которые входят в ее состав по умолчанию.

Для настройки MasterSCADA 4D Client (для Widows)

Существует два способа настройки.

Вариант 1

При помощи настройки приложения MasterSCADA 4D Monitor, а в случае если используется клиент визуализации, установленный независимо от исполнительной системы, то MasterSCADA 4D Client Monitor.

В поле Параметры запуска клиента необходимо указать следующие ключи:

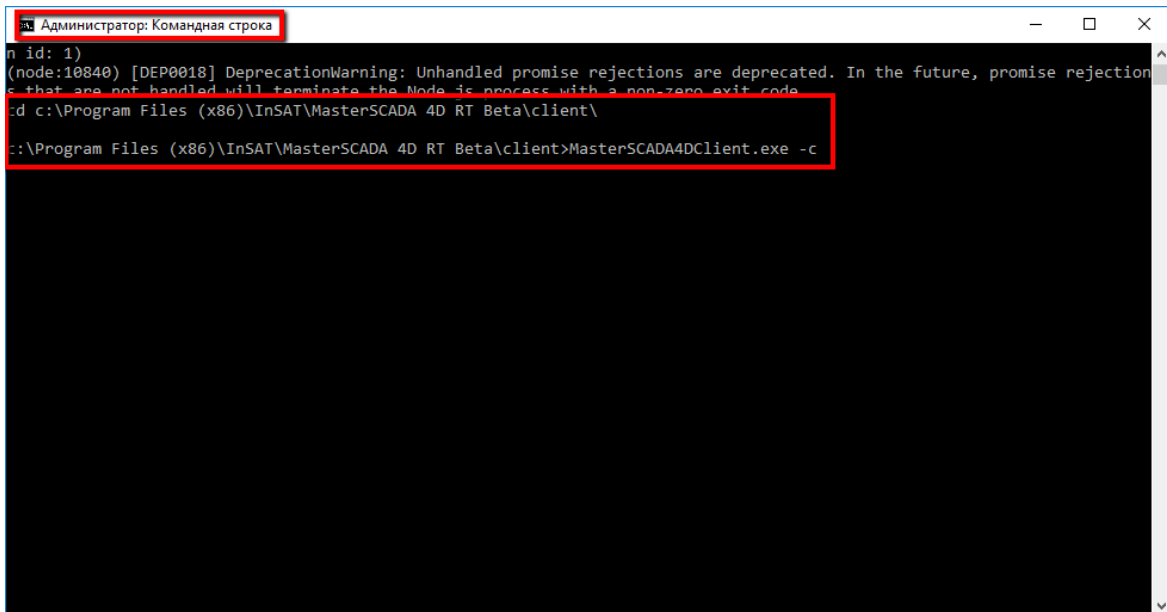
-cp -p "имя принтера в системе" - устанавливает принтер по умолчанию

-s " путь к месту хранения" - устанавливается папка для сохранения файлов по умолчанию, например, при работе с журналом сообщений.

Вариант 2

Путем внесения изменений в конфигурационный файл *options.json*. Файл находится в той же папке, что и *MasterSCADA4DClient.exe*. Если используется клиент, входящий с состав среды исполнения, то файл находится в папке: *c:\Program Files (x86)\MPSSoft\MasterSCADA 4D RT[номер версии]\client*

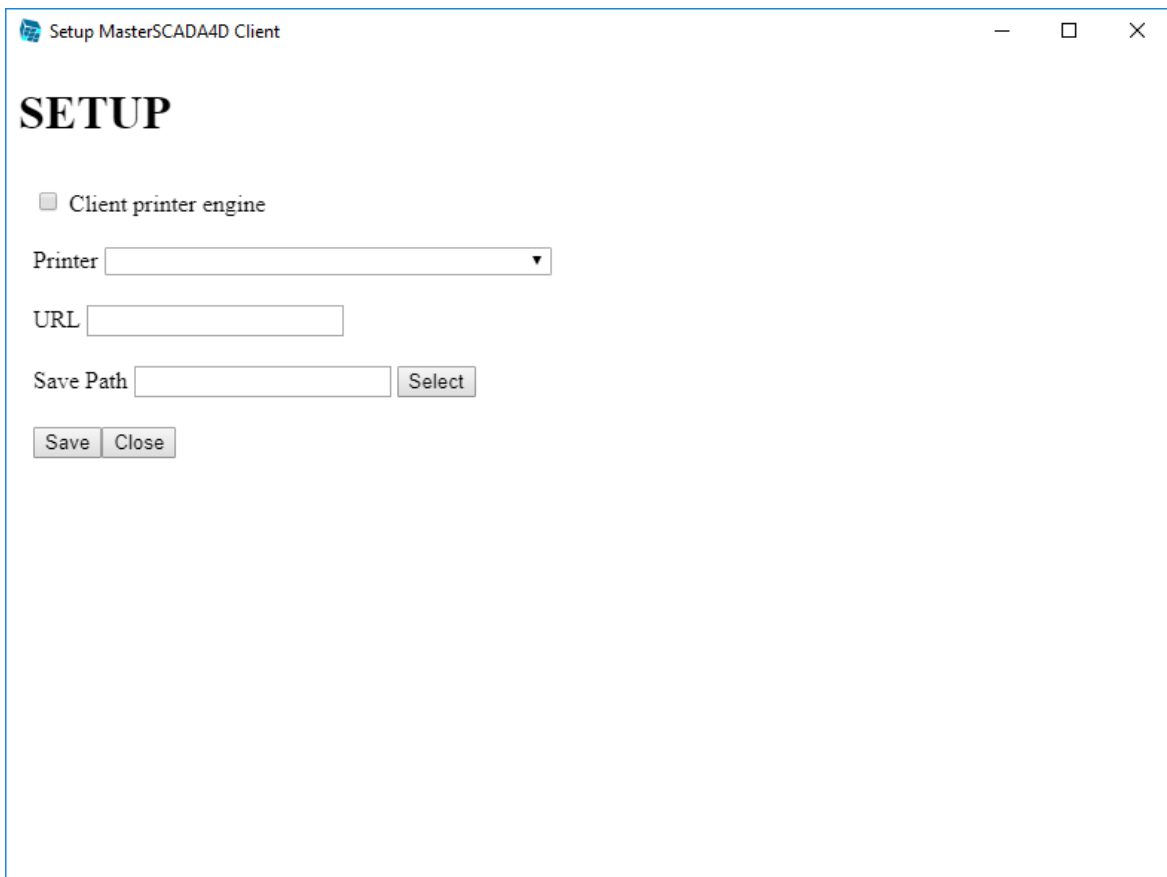
Для получения возможности редактирования этого файла необходимо запустить приложение с ключем -c. Например, можно воспользоваться стандартным приложением Windows Командная строка:



```
Администратор: Командная строка
n id: 1)
(node:10840) [DEP0018] DeprecationWarning: Unhandled promise rejections are deprecated. In the future, promise rejections
s that are not handled will terminate the Node.js process with a non-zero exit code
cd c:\Program Files (x86)\InSAT\MasterSCADA 4D RT Beta\client\
c:\Program Files (x86)\InSAT\MasterSCADA 4D RT Beta\client>MasterSCADA4DClient.exe -c
```

Важно! Приложение должно быть запущено с правами Администратора.

В результате откроется диалоговое окно:



Элемент	Описание
Client printer engine	Включает режим установки принтера по умолчанию, если флаг не установлен, то настройка Printer будет проигнорирована
Printer	Позволяет выбрать нужный принтер среди установленных на компьютере.
URL	Аналог поля Адрес сервера в приложениях MasterSCADA 4D Monitor и MasterSCADA 4D Client Monitor
Save Path	Позволяет указать место сохранения файлов по умолчанию.
Save	Сохраняет сделанные настройки.
Close	Закрывает диалоговое окно.

После выполнения указанных действий появится файл options.json.

Важно! После обновления приложения настройки будут сохранены.

Сторонние клиенты

При использовании сторонних клиентов, для установки принтера и места сохранения файлов по умолчанию, необходимо действовать согласно инструкции, поставляемой с данными приложениями.

10.2.7. ГОРЯЧИЕ КЛАВИШИ В КЛИЕНТЕ ВИЗУАЛИЗАЦИИ

В клиенте визуализации MasterSCADA 4D Client доступны следующие горячие клавиши:

Клавиши	Назначение	Версия НМІ
ALT+ курсор влево	Аналог стандартному действию в браузерах Назад. Открывает предыдущее окно.	
ALT+ курсор вправо	Аналог стандартному действию в браузерах Вперед. Открывает следующее окно.	

Клавиши	Назначение	Версия НМІ
F11	Включает/отключает полноэкранный режим. Если изначально клиент был запущен в данном режиме, то отключить его при помощи этой клавиши невозможно.	Все
F12	Открывает инструменты разработчика. Клавиша активна только в том случае, если в параметрах запуска клиента установлен ключ -d. Ключ можно установить приложениях MasterSCADA 4D Monitor, MasterSCADA 4D Client Monitor (если клиент запускается независимо от среды исполнения)	Все

Использование горячих клавиш в других клиентах визуализации (браузерах) определяется их возможностями.

10.2.8. МНОГОМОНИТОРНЫЙ РЕЖИМ

Клиент визуализации MasterSCADA 4D Client поддерживает многомониторный режим работы.

Важно! Функция поддерживается только в версии НМІ v2

Для того чтобы включить этот режим, необходимо в приложениях MasterSCADA 4D Monitor, MasterSCADA 4D Client Monitor (если клиент запускается независимо от среды исполнения) установить в поле Параметры запуска клиента ключ -m [число мониторов]. В этом случае на компьютере откроется указанное количество окон MasterSCADA 4D Client, но не больше, чем количество физически подключенных мониторов. На каждом мониторе откроется свое окно клиента.

Важно! Окно авторизации пользователя будет появляться только на одном мониторе - на том, который указан первым в операционной системе. Завершение сессии работы в исполнительной системе оператора на одном мониторе приведет к такому же действию на всех окнах, открытых на других мониторах, т.е. в многомониторном режиме для оператора создается одна сессия.

В процессе работы оператор может в каждом мониторе осуществлять независимые действия (открывать окна, работать с графиками, журналами и т.п.).

Важно! Несмотря на то, что при такой работе открывается одновременно несколько окон MasterSCADA 4D Client, считается, что к исполнительной системе подключен только один клиент.

10.2.9. ОЧИСТКА КЕША MASTERSCADА 4D CLIENT

Кеш MasterSCADA 4D Client можно очистить вручную.

В ОС Windows он хранится по умолчанию в папке *C:\Users\<user>\AppData\Roaming\MasterSCADA4DClient*

В ОС Linux он хранится по умолчанию в папке: */home/<user>/.config/MasterSCADA4DClient*

10.3. ОТЛАДКА ПРОЕКТА

Необходимые средства и режимы отладки обеспечивают инструменты, которые содержатся в панели Устройства и вкладке инструментов Отладка.

10.3.1. РЕЖИМЫ ОТЛАДКИ

В MasterSCADA 4D предусмотрены следующие виды отладки проекта:

- непрерывная отладка;
- отладка с шагом 1 цикл;

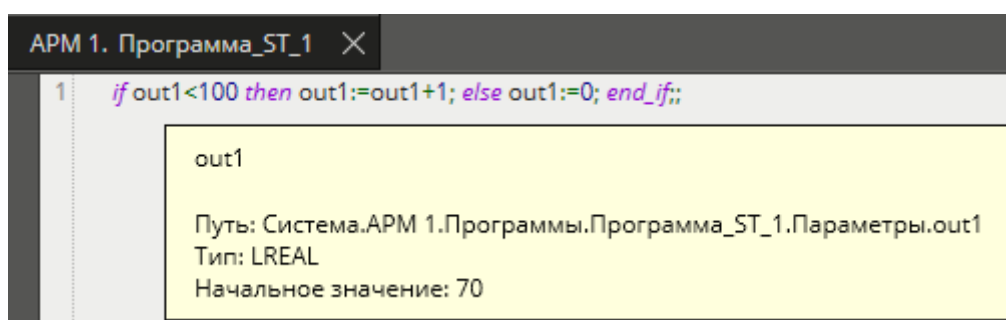
Для индикации/изменения значений параметров в режиме отладки в редакторе проекта используются:

- Дерево (Окно структуры проекта, Клеммники, Легенда и т.д.);
- Наблюдатель.

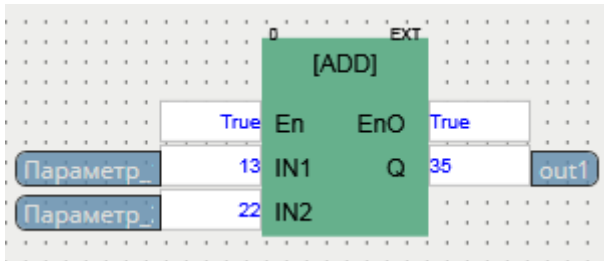
Чтобы выбрать режим отладки, нужно воспользоваться вкладкой инструментов Отладка.

Значения параметров также индицируются и могут быть изменены в редакторах программ:

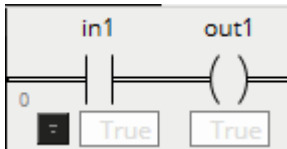
- в редакторе ST – во всплывающей подсказке параметра (только индикация):



- в FBD- и LD-редакторе:
 - значения входов и выходов FBD-блока – в полях редактирования около входов/выходов (если дважды щелкнуть пользовательский ФБ на диаграмме, откроется алгоритм ФБ, в котором отображаются значения параметров этого ФБ):

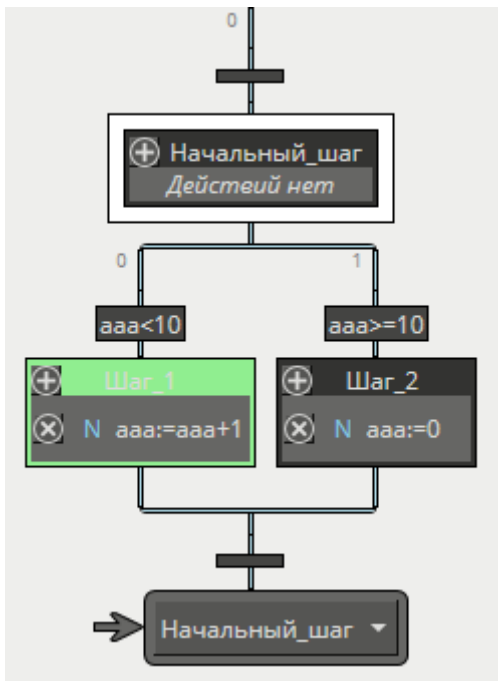


- значение связанного параметра LD-блока – в поле редактирования под блоком:




В режиме отладки в редакторах программ используется цветовая индикация конструкций языка:

- на SFC-диаграмме зеленым цветом подсвечиваются разрешенные переходы и активный шаг:



10.3.1.1. НЕПРЕРЫВНАЯ ОТЛАДКА

Для перехода в режим непрерывной отладки следует перевести кнопку Пауза  в отжатое положение.

Для задания периода опроса узла в отладке используется период службы Межузловая связь (по умолчанию 100мс).

10.3.1.2. ОТЛАДКА С ШАГОМ 1 ЦИКЛ

В этом режиме значения каналов и параметров, связанных с каналами, вычисляются непрерывно с шагом, равным заданному периоду выполнения соответствующей задачи.

Значения всех остальных параметров вычисляются по каждому нажатию кнопки Цикл



, при этом код программ выполняется целиком.

10.3.2. КАК ПОЛУЧИТЬ ДИАГНОСТИЧЕСКУЮ ИНФОРМАЦИЮ СРЕДЫ ИСПОЛНЕНИЯ

Среда исполнения для ОС Windows

Во время работы среды исполнения лог-файлы формируются автоматически. Место хранения лог-файлов зависит от типа используемой среды исполнения.

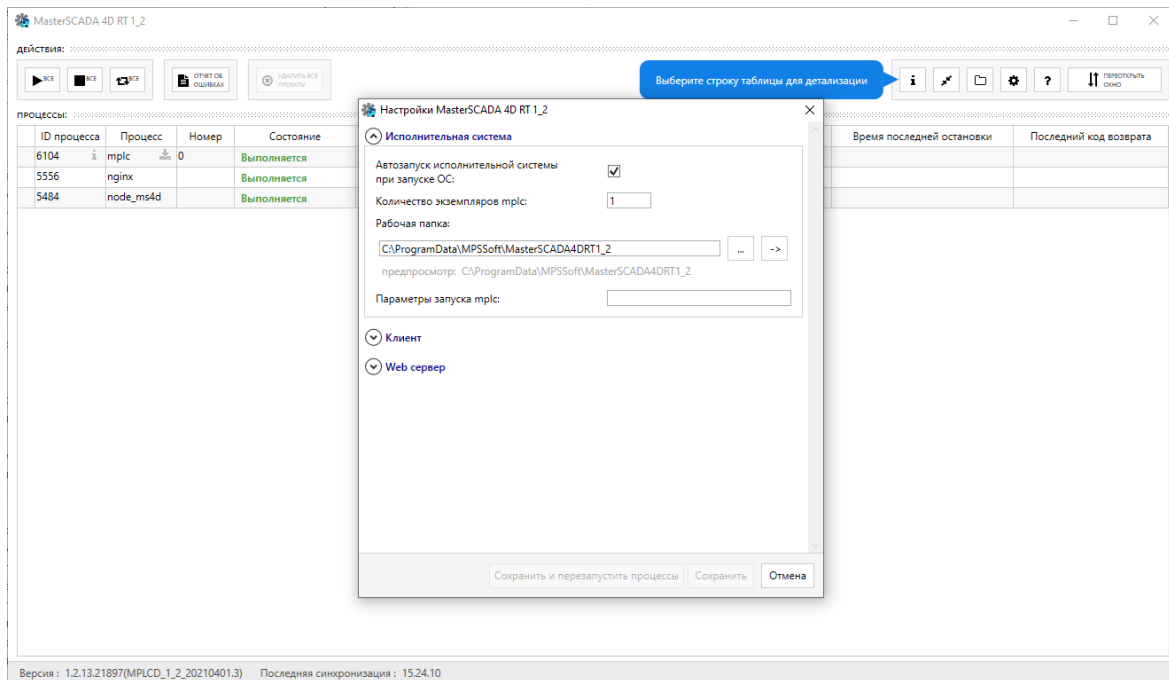
Среда исполнения, входящая в комплект среды разработки

В случае если используется среда исполнения, входящая в состав среды разработки, файлы, по умолчанию, хранятся в папке: `"c:\Users\[имя пользователя]\AppData\Roaming\MPSSoft\MasterSCADA4D[номер версии]\Debug_[имя проекта]\[имя узла]\PLC\logs\log_[дата формирования файла].txt"`

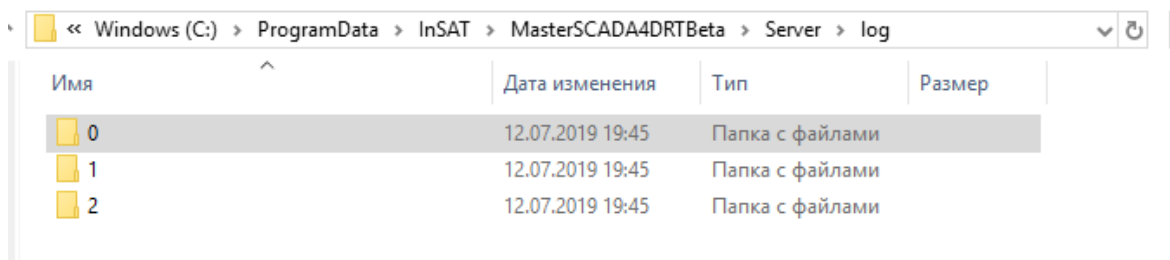
Установленная независимо среда исполнения

В случае если используется исполнительная система, установленная как независимое приложение, лог-файлы хранятся в рабочей папке среды исполнения.

Быстрый доступ к рабочей папке можно получить при помощи приложения MasterSCADA 4D Monitor. Для этого необходимо открыть окно настройки и перейти к рабочей папке:



Далее необходимо открыть папку *Server/Log*, в которой может содержаться несколько вложенных папок, в зависимости от того, сколько экземпляров запускалось на данном компьютере:



Важно! Диагностическая информация о периодах работы среды исполнения выдается в конце файла.

Любая среда исполнения

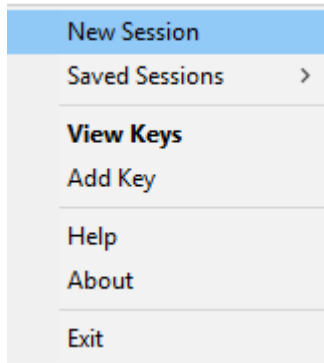
Работающая среда исполнения выдает диагностическую информацию по TCP/IP через порт 31550. Для чтения этой информации можно воспользоваться любым программным эмулятором терминала, например HyperTerminal, PuTTY и др..

Рассмотрим запись лога с отладочной информацией при помощи свободно распространяемой программы PuTTY – <http://www.putty.org>

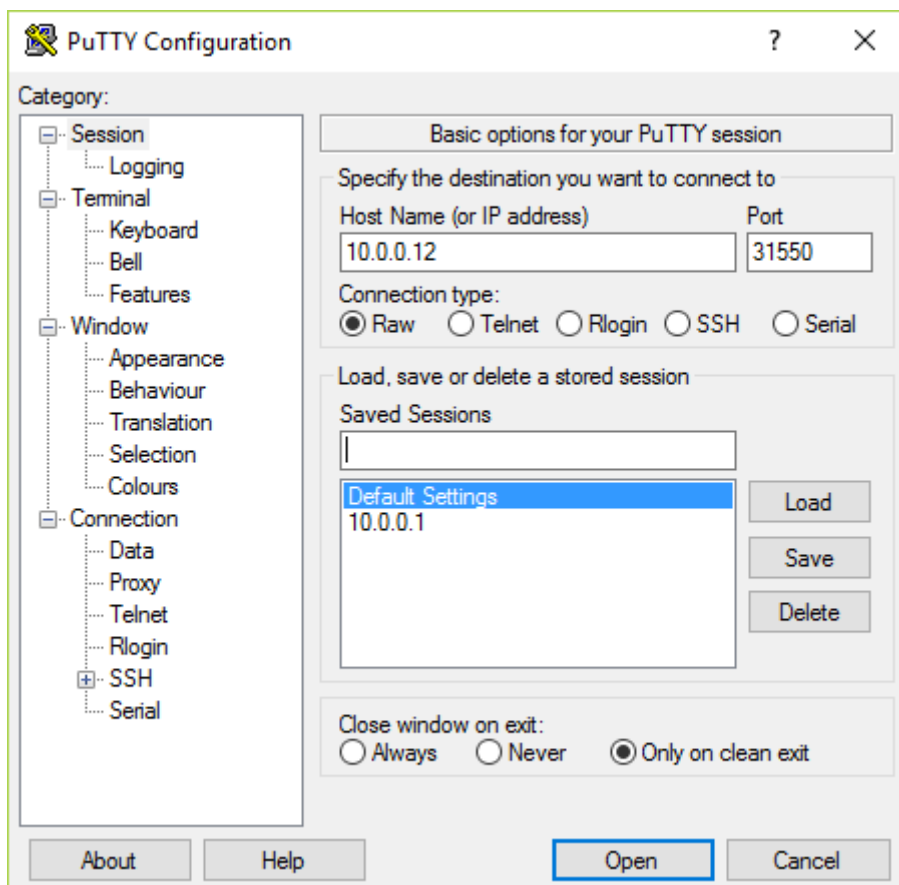
Запустить программу можно с помощью ярлыка



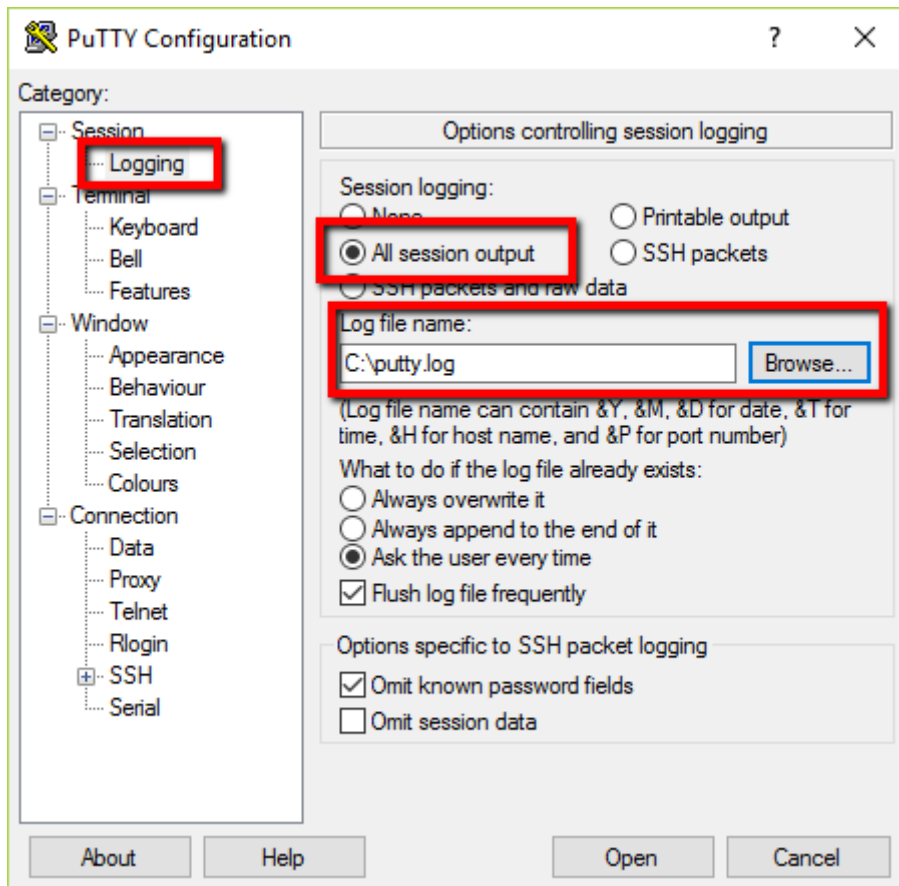
По умолчанию, программа минимизируется при запуске в панели задач. Для выполнения настройки программы необходимо нажать правой кнопкой мыши на её ярлык в панели задач, и в контекстном меню выбрать пункт New Session:



При этом откроется окно настроек, в котором следует задать IP-адрес узла с установленной средой исполнения, порт и тип соединения.

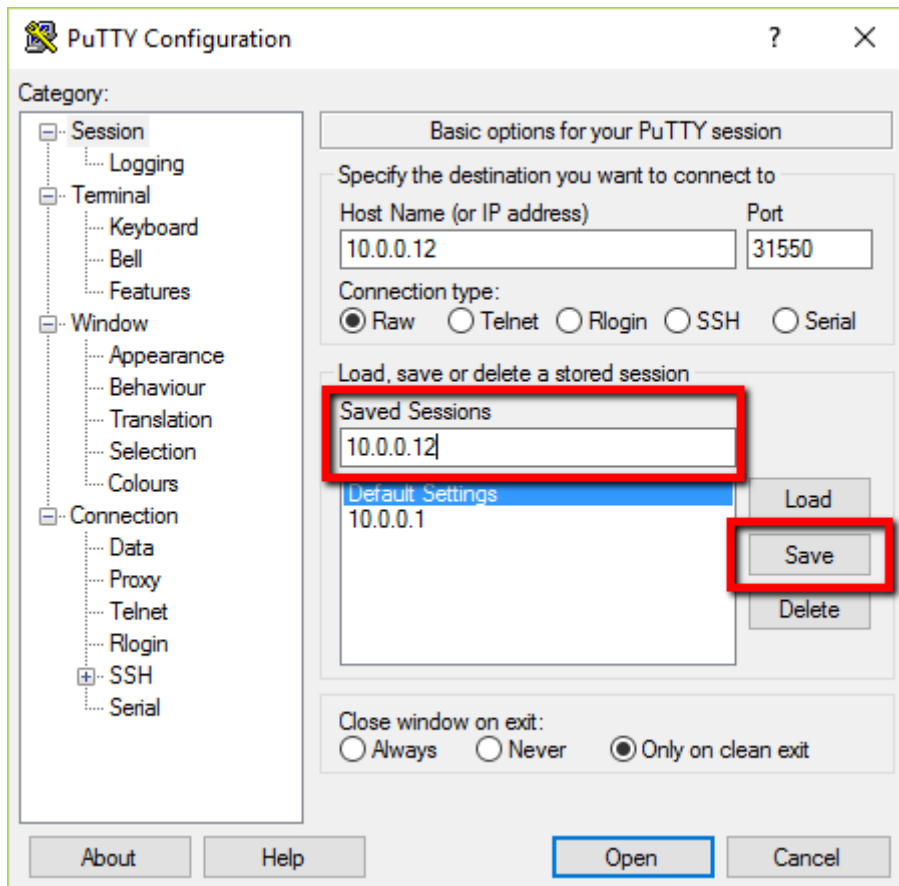


После этого требуется определить место хранения лог-файлов:



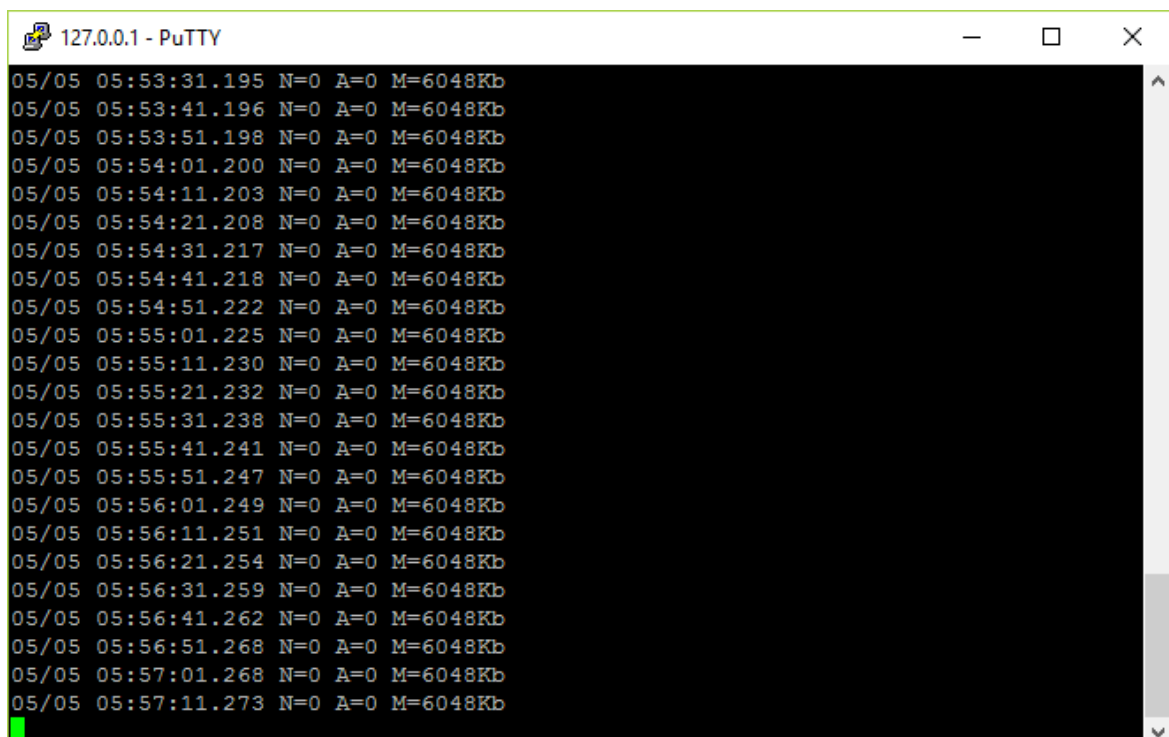
Место, в которое планируется сохранить лог-файл, должно быть доступно для записи.

Рекомендуется сохранить данную сессию, для того чтобы в дальнейшем иметь возможность оперативно подключиться к среде исполнения без ввода настроек подключения.



Затем необходимо нажать кнопку Open.

Откроется окно, в которое будет выводиться отладочная информация:



Один раз в 10 секунд будут выводиться сообщения со статистикой всех задач. Информация о работе системы архивирования будет выдаваться один раз в минуту. Файл с сохраненной отладочной информацией необходимо предоставить в службу технической поддержки по электронной почте support.ms4d@masterscada.ru для анализа.

Пример отладочных сообщений

Периоды выполнения задач

Каждые 10 сек отображается диагностическое сообщение со статистикой по всем задачам -

```
M14/04    20:28:39.771    N=880550(1,71)    N2=34707    A=0    O0=41552(0,e0)
U0=167423(100,100,422)(90,83,115)(,0,42,46)13797Kb
U1=16746(1000,1000,1454)(1,1,1454)(,0,0,1)525Kb    S0=314891,e0/0(53,49,57)(2,1,27)368Kb
HR=16603(6) R=828487(0,0,e1) M=110060Kb
```

где:

- В начале строки признак M - Master, S - Slave
- дата - время
- N - количество обработанных запросов UDP (за исключением отдельных типов, которые считаются в N2). В скобках указывается - <максимальное время обработки запроса, N функции протокола по которой была самая долгая обработка>
- N2 - количество обработанных запросов UDP, которые используются для взаимодействия со средой разработки при отладки программ при указанной опции /udp2, а также при синхронизации архивов (данные запросы обрабатываются в отдельном потоке)
- M=<кол-во обработанных запросов> - в случае Modbus RTU Slave задачи
- A=<количество обработанных запросов от HMI клиентов>
- O<индекс контроллера>=<кол-во циклов>(<время последнего цикла опроса>, <общее кол-во ошибок>) - статистика задачи межузловой связи (для связи с каждым контроллером отдельная задача).
- U<индекс задачи>=<кол-во циклов>(<среднее цикла>, <мин цикла>, <макс цикла>)(<среднее реального выполнения>, <мин>, <макс>)(, <время чтения входных данных в последнем цикле>, <время выполнения задачи в последнем цикле>, <время записи выходных данных в последнем цикле>)<объем памяти виртуальной машины>Kb - статистика задачи объектов
- S<индекс задачи>=<кол-во циклов>,e<общее кол-во ошибок>/<кол-во ошибок в последнем цикле>(<среднее цикла>, <мин цикла>, <макс цикла>)(<среднее реального выполнения>, <мин>, <макс>)<объем памяти виртуальной машины>Kb - статистика задачи протоколов
- HR=<кол-во сохранений горячего рестарта>(<время, затраченное на последнее сохранение>)

- R =<кол-во циклов задачи резервирования>(<время последнего цикла>, <время последней синхронизации данных>, <макс время ответа от другого контроллера>, <кол-во ошибочных ответов>). В режиме SLAVE на каждом цикле выполняется проверка состояния второго контроллера, а также получение данных, если с прошлого получения прошел заданный период задачи. В конце каждого цикла дополнительно пауза 20мс
- M=<используемый объем памяти процессом>Kb

Для получения расширенной диагностической информации требуется нажать клавишу клавиатуры *o*, а затем *Enter*

В этом случае на экран будет выдаваться трассировка:

```

127.0.0.1 - PuTTY
2171953: End get (count=15) dt=0 q=155 [addr=127.0.0.1 port=62340]
2171980: err=-3 Task0 ModbusRead module=0(A=1), request=0, StartAddr=0, Count=9
1 3
0ModbusTCP read 010.000.006.010
0ModbusTCP write 010.000.006.010
0ModbusTCP write 010.000.006.010
2172055: End get (count=15) dt=0 q=155 [addr=127.0.0.1 port=62340]
2172087: err=-3 Task=0 ModbusWrite module=1(A=2), request=0, StartAddr=0, Count=
1 1 0
2172155: End get (count=15) dt=0 q=155 [addr=127.0.0.1 port=62340]
2172187: err=-3 Task=0 ModbusWrite module=1(A=2), request=1, StartAddr=1, Count=
1 2 0
2172256: End get (count=15) dt=0 q=155 [addr=127.0.0.1 port=62340]
2172288: err=-3 Task=0 ModbusWrite module=1(A=2), request=2, StartAddr=2, Count=
1 3 0
Trace disabled !
2174562: Error TCP connect : 0 (port=502 state=0) 010.000.006.010
2177849: Error TCP connect : 0 (port=502 state=0) 010.000.006.010
11/05 06:06:44.822 N=492 A=0 O0=570(0,e0) U0=587(100,100,104)(0,0,1) S0=97,e97(6
04,458,2409)(599,452,2404) HR=1(4) M=7360Kb
MT 0.00 MR 0 NRq 0 NI 0 R(0 0) Rq(0 0) WI(-0.00 0.00 0.00) TWII(-0.00 0.00 0.00)
TWIT(-0.00 0.00 0.00) RqI(-0.00 0.00 0.00) RqT(-0.00 0.00 0.00) DR(0 0) TDII(-
0.00 0.00 0.00) TDIT(-0.00 0.00 0.00)

```

Статистика архивирования

Статистика архивирования выдается один раз в минуту. Строка с информацией имеет вид:

```
Stat: { Archive Info }
R(Tm:float, Ct:int, Av:int) W(Tm:float, Ct:int, Av:int, Wt:int, Ps: int, Ls:int) D(Tm:float, Ct:int, Av:int)
```

Где:

R - Чтение

W - Запись

D - Удаление

Tm - Число секунд, затраченных на обработку R/W/D

Ct - Количество записей обработанных за последнюю минуту R/W/D

Av - Средняя скорость в сек процессорного времени R/W/D

Wt - Количество записей в очереди на запись в БД

Ps: - Количество записей в сек реального времени

Ls: - Количество удаленных записей в результате прореживания (по переполнению очереди записи)

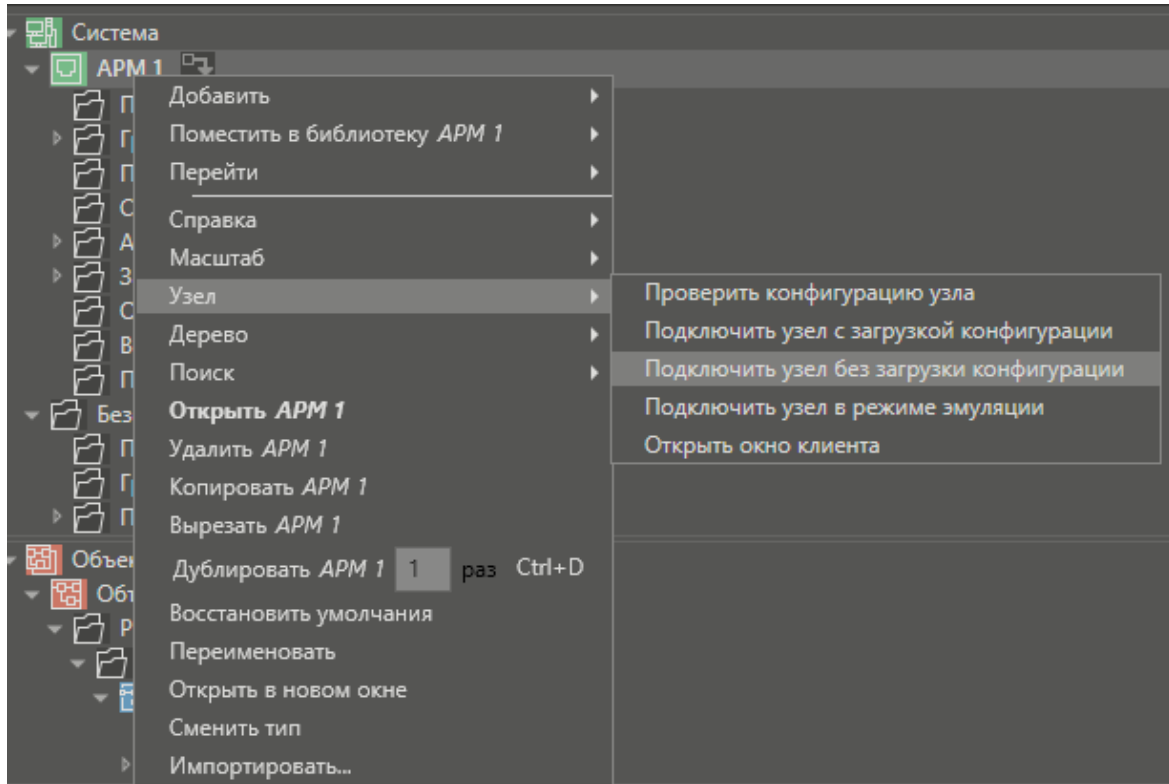
Получение диагностической информации при помощи функциональных блоков

Для получения диагностической информации встроенными средствами MasterSCADA 4D можно использовать специальные функциональные блоки библиотеки BaseObjects:

- GetSystemInfo - получение информации об установленной версии исполнительной системы и загруженном проекте
- RedundancyControl - управление резервированием.
- FileIntegrityControl - управление контролем целостности ПО и проекта.
- GetOpcUaServerInfo - получения информации и статистики по работе OPC UA сервера.
- GetDataArchiveStatistic - получения информации и статистики по работе архива данных.
- GetEventsArchiveStatistic - получения информации и статистики по работе архива данных.
- GetTaskStatistics - получение статистики работы задач.

10.3.3. ПОДКЛЮЧЕНИЕ СРЕДЫ РАЗРАБОТКИ К ИСПОЛНЯЕМОМУ УЗЛУ

Если узел уже находится в процессе исполнения, то для подключения к этому узлу среды разработки необходимо выполнить команду Узел.Подключить узел без загрузки конфигурации:



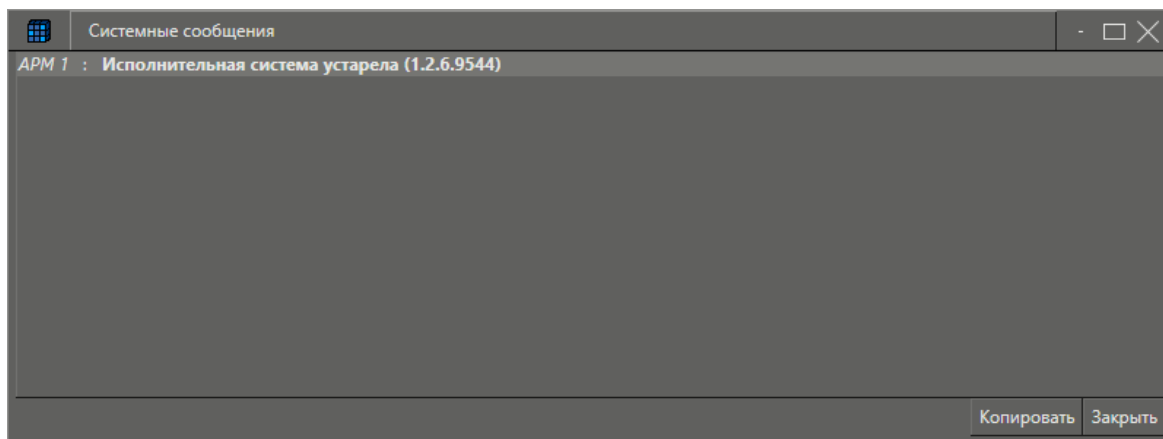
В результате этого, в интерфейсе среды разработки отобразятся текущие значения параметров, которые допускается принудительно изменить.

Отключение среды разработки от исполняемого проекта, выполненное при помощи контекстного меню или вкладки Исполнение, не приводит к остановке исполнения проекта.

При взаимодействии среды разработки и среды исполнения UDP-запросы на получение/запись данных могут обрабатываться в отдельном потоке. Для включения обработки данных запросов в отдельном потоке в параметрах запуска исполнительной системы надо задать опцию /udr2 (по умолчанию отключено, так как в некоторых сетях не доходят ответы при отправке с другого UDP сокета). Данные типы запросов в статистике отображаются как N2

Несоответствие версии среды разработки и среды исполнения

Если версия среды разработки выше, чем версия среды исполнения, то при попытке подключения к узлу появится сообщение:



В этом случае необходимо обновить среду исполнения.

Для predetermined контроллеров окно сообщения будет иметь кнопку Обновить исполнительную систему, при нажатии на которую обновление версии произойдет автоматически. Исполнительные системы, защищаемые USB-ключом, необходимо обновить вручную. Для формирования новой версии исполнительной системы обратитесь по электронному адресу sales@masterscada.ru.

Среда исполнения для различных контроллеров и операционных систем поставляется в комплекте со средой разработки и находится в папке *c:\Program Files (x86)\MPSSoft\MasterSCADA 4D 1.2\bin\Config\MasterPLC\PLC*.

Скачать демоверсию среды исполнения и получить информацию о способах покупки коммерческой версии можно на сайте компании "МПС Софт".

10.3.4. ЭМУЛЯЦИЯ УСТРОЙСТВА

Проект MasterSCADA 4D можно отлаживать даже если реальное оборудование еще не подключено. Для этого необходимо на вкладке Исполнение панели инструментов нажать кнопку Эмуляция.

В этом случае запустится версия среды исполнения, входящей в состав среды разработки. Опрос устройств по различным протоколам производиться не будет, также не будут работать функциональные блоки SEND_EMAIL и HttpClient. Однако все значения можно установить вручную.

Если проект содержит несколько узлов, то в этом режиме автоматически запустится несколько экземпляров среды исполнения, однако номера экземпляров будут отличаться.

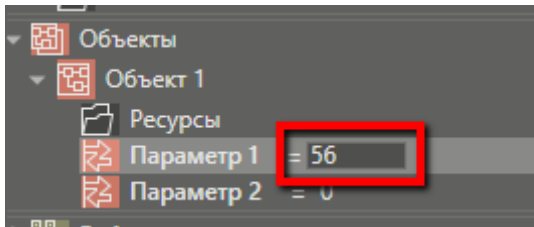
Если проект содержит окна, то будет открыто несколько окон клиентов по числу узлов.

Важно! На одном компьютере не может быть запущено две среды исполнения, имеющие одинаковые номера экземпляров. Поэтому, если на компьютере уже запущена среда исполнения, установленная как независимое от среды разработки приложение, то проект будет загружаться в нее, даже в случае нажатия на кнопку Эмуляция на вкладке Исполнение.

10.3.5. ИМИТАЦИЯ ЗНАЧЕНИЙ ПАРАМЕТРОВ

Установка произвольного значения параметра в дереве

Если среда разработки подключена к среде исполнения, то в дереве, в панели клеммников и т.п. можно установить значение параметра. Это значение будет использоваться при работе программы в среде исполнения.



При взаимодействии среды разработки и среды исполнения UDP-запросы на получение/запись данных могут обрабатываться в отдельном потоке. Для включения обработки данных запросов в отдельном потоке в параметрах запуска исполнительной системы надо задать опцию /udr2 (по умолчанию отключено, так как в некоторых сетях не доходят ответы при отправке с другого UDP сокета). Данные типы запросов в статистике отображаются как N2


Поддерживается отображение в среде разработки, подключенной к среде исполнения, значений параметров различных типов, в том числе динамических массивов (массивов переменной длины), а также сохранение длинных строк (более 256 символов), при условии, что параметр имеет тип STRING

Особенности отображения параметров специальных типов

Значения параметров типа ANY отображаются только если в режиме исполнения принимают значения атомарных типов. Массивы и структуры отображаться не будут.


При вводе значения в среде разработки оно передается как STRING.

Блокировка параметра в дереве

После ввода значения параметра, рядом с параметром появится знак . Это означает, что введенное значение будет заблокировано для изменений по связи, а данные, приходящие по связи, будут игнорироваться.

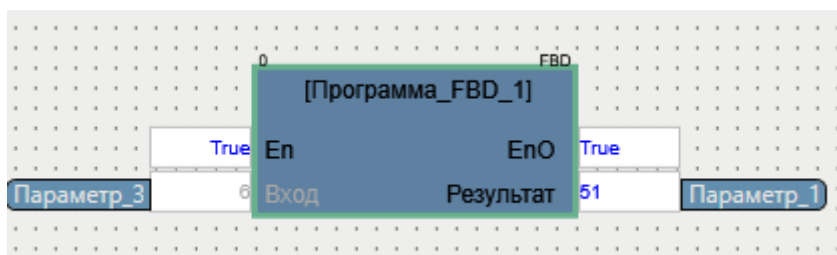
В случае если необходимо зафиксировать какое-либо значение параметра, то можно также воспользоваться соответствующими инструментами вкладки Отладка панели инструментов.

Снятие блокировки параметра в дереве

Для того чтобы снять блокировку с параметра, необходимо либо нажать левой кнопкой мыши на знак  рядом с параметром, либо воспользоваться соответствующими инструментами вкладки Отладка панели инструментов.

Установка/Блокировка/Снятие блокировки на схеме FBD

Для установки значения в программе FBD, если среда разработки подключена к среде исполнения, нужно ввести значение в окно, рядом с ножкой входа или выхода. После нажатия клавиши Enter или перевода курсора в любое место редактора значение будет установлено, и включится блокировка. Если блокировка включена, то имя входа/выхода и его значение будет отображаться серым цветом:



Чтобы снять блокировку нужно выполнить клик мыши по имени входу/выходу. Чтобы установить блокировку, не вводя при этом новое значение, нужно также выполнить клик по имени входа/выхода функционального блока.

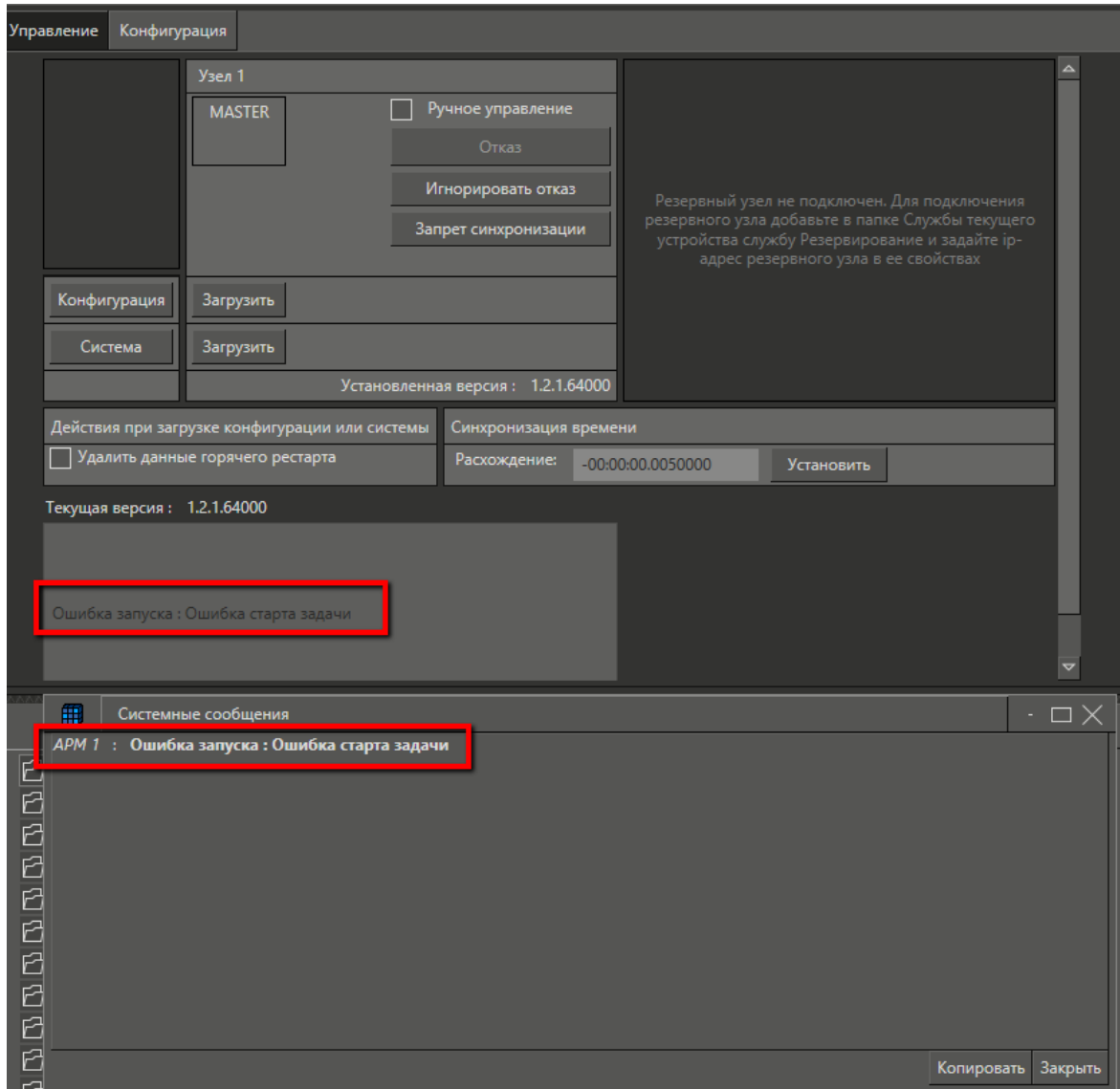
Автоматическая имитация значений каналов

Для автоматическом имитации значений каналов протоколов, в которые добавляются модули ввода-вывода, например, Modbus, DCON, можно использовать ключ /imit, который необходимо задать в свойстве Параметры запуска RT. В этом случае аналоговые параметры будут пилообразно менять значения от 0 до 100, а дискретные параметры будут менять значения с 0 на 1 и обратно.

10.3.6. СООБЩЕНИЯ ОБ ОШИБКАХ


В среде разработки, подключенной к среде исполнения могут появиться сообщения, сигнализирующие о возникновении каких-либо проблем в работе.

Сообщения отображаются в окне системных сообщений, которое открывается автоматически, а также в панели узла, во вкладке Управление.




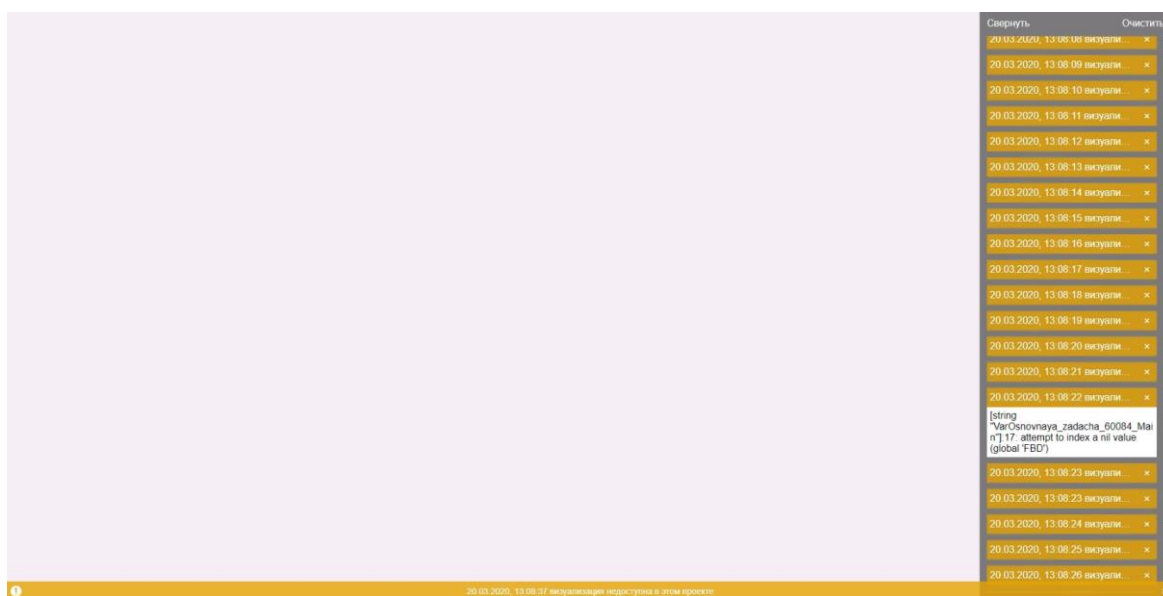
Сообщение	Причина	Комментарии
<p>Превышено разрешенное число экземпляров</p>	<p>На одном устройстве запущено большее, чем позволяет лицензия, число экземпляров среды исполнения.</p>	<p>Возможно, при тестировании проекта использовалась демонстрационная версия, и разные узлы запускались на одном компьютере. В этом случае в настройках узлов следует указать соответствующие IP-адреса устройств, на</p>

Сообщение	Причина	Комментарии
		<p>которых установлены среды исполнения, и указать корректные значения свойства Номер экземпляра. В том случае, если планировалось запускать несколько экземпляров сред исполнения на одном компьютере, необходимо приобрести соответствующую лицензию.</p>
Не загружена конфигурация	Среда разработки подключена к среде исполнения, в которую не загружена конфигурация (проект)	<p>Выполните загрузку конфигурации доступным способом:</p> <ul style="list-style-type: none"> • Автоматическая загрузка проекта • Ручная загрузка проекта в среду исполнения
Ошибка конфигурации	Ошибка в файле конфигурации	Повторно подготовьте проект к запуску в среде исполнения и загрузите его в исполнительную систему.
Ошибка загрузки конфигурации	Ошибка при чтении файла конфигурации	Повторно подготовьте проект к запуску в среде исполнения и загрузите его в исполнительную систему.

Сообщение	Причина	Комментарии
Неподдерживаемый формат конфигурации	В узел загружена конфигурация проекта, созданная в более поздней версии среды разработки, чем версия исполнительной системы (учитываются первые 3 числа, например 1.2.8 и 1.2.9)	Необходимо обновить версию исполнительной системы
Неподдерживаемая платформа в конфигурации	Конфигурация создана для другой платформы. В текущей версии не используется	
Ошибка старта задачи	<p>Возникает, если в проекте используется ФБ, неподдерживаемый в данной платформе, а также в случае использования отчетов в исполнительных системах, предназначенных для работы на контроллерах.</p> <p>В тексте ошибки выдается имя элемента проекта, который стал причиной проблемы.</p>	Удалите из проекта проблемный элемент. В случае возникновения сложностей при выборе альтернативного решения обратитесь в службу технической поддержки по электронной почте support.ms4d@masterscada.ru
Ошибка старта протокола	<p>В узел добавлен протокол, не поддерживаемый на данной платформе. Пример текста сообщения:</p>  <p>Protocol 3 - это порядковый номер протокола в дереве системы</p>	Удалите протокол из проекта.
Превышено ограничение для [название протокола/драйвера]: Доступное - X, Требуемое - Y	В проекте задействовано больше каналов, чем позволяет используемая лицензия (X<Y). При этом загрузки конфигурации и подключения к узлу происходить не будет.	Задействованным каналом считается тот, который имеет связи или архивируется. Если все каналы необходимы для функционирования проекта, то необходимо увеличить количество

Сообщение	Причина	Комментарии
		точек данного протокола.

В случае, если клиент визуализации подключается к узлу, который не был запущен (например, если проект не запущен из-за ошибки проверки целостности или в задаче использован ФБ, не подключенный к исполнительной системе), то в строке статуса будет отображаться последняя возникшая ошибка. Если нажать на знак  в строке статуса, то откроется панель, в которой будут отображаться все текущие ошибки:



10.4. ПОРЯДОК ИСПОЛНЕНИЯ И ПЕРЕДАЧИ ДАННЫХ

В исполнительной системе можно определить следующие задачи.

Задачи	Настройка	Назначение
Задачи узла	Панель свойств элементов Основная задача, Задача группы узла Задачи.	Созданный пользователем в среде разработки проект (программы, параметры, объекты и т.п.) перед загрузкой в среду исполнения преобразуется (компилируется). В результате компиляции вся логика работы проекта, независимо от того на каком языке она была написана разработчиком проекта, преобразуется в специальную программу, которая будет загружаться в устройство и там циклически исполняться. Количество

Задачи	Настройка	Назначение
		<p>программ зависит от количества задач, созданных в дереве системы в данной группе.</p> <p>По умолчанию, у каждого узла при его создании создается только одна задача - Основная задача. Разработчик проекта может создать столько задач, сколько необходимо для работы конкретного проекта. Особенности вычисления, например, цикл вычисления, будет зависеть от настроек той или иной задачи. Каждая задача будет работать независимо от других задач в устройстве.</p> <p>Создание нескольких задач будет целесообразно в том случае, если необходимо обеспечить разные циклы вычисления разных программ проекта.</p>
Задачи протоколов	Панель свойств Протоколов, категория Задача	<p>Для получения и отправки данных по каждому из протоколов создается отдельная задача. Каждая задача будет работать со своим периодом. Если в протоколе была создана программа, то период работы будет также зависеть от периода настроенного в протоколе.</p>
Задачи SLAVE-протоколов	Включение задач производится в панели свойств Узла	<p>Если MasterSCADA 4D выступает в роли сервера, например, передает данные по протоколу OPC UA, то создается для каждого протокола своя задача. Исполнительная система постоянно слушает запросы клиента и отправляет данные согласно этим запросам, т.е. период задачи в MasterSCADA 4D не настраивается.</p>
Сохранение состояния	Панель свойств элемента Сохранение состояния узла, категория Задача	<p>Служебная задача Сохранение состояния - используется для восстановления значений после перезагрузки среды исполнения. Как правило, применяется для того, чтобы значения уставок после перезагрузки проекта в среде исполнения приняли значения, введенные оператором либо полученные в</p>

Задачи	Настройка	Назначение
		<p>результате работы программ в предыдущей сессии. Сохраняются те значения параметров, в настройках которых установлено свойство Сохранять=Да. Задача работает независимо от других задач. Период сохранения зависит от настроек, сделанных в панели свойств службы. Служебная задача</p>
Межузловая связь	<p>Панель свойств элемента Межузловая связь узла, категория Задача</p>	<p>Служебная задача используется для передачи данных между разными узлами, на которых установлена исполнительная система MasterSCADA 4D RT и настроена связь между параметрами.</p> <p>Если есть связь между узлами (по прямой связи параметров в проекте или сгенерированная автоматически при использовании в окне одного узла параметров другого узла), то при чтении служба в узле проверяет наличие изменения или величины значения или метки времени. Если изменилась любая из этих характеристик, то только тогда значение передается по межузловой связи.</p>
Резервирование	<p>Панель свойств элемента Резервирование узла, категория Задача</p>	<p>Служебная задача используется для обмена данными при резервировании серверов. Задача работает независимо от других задач, в зависимости от настроек сделанных в службе</p>
Архивирование	<p>Панели свойств элементов, которые находятся в группе Архивы узла</p>	<p>Задачи создаются исполнительной системой для работы с базой данных, в которой хранятся архивы, накопленные во время работы. Для работы с базой, где хранятся данные и базой, где хранятся сообщения, создаются отдельные подключения. Помимо этого количество задач (потоков подключения к БД) зависит от настройки Максимальное количество одновременных подключений, каждого из элементов находящихся в группе Архивы узла. Задача</p>

Задачи	Настройка	Назначение
		работает независимо от других задач, в зависимости от настроек сделанных в настройках архивов.

Каждая задача работает в одном или нескольких потоках ОС. Потоки распределяются на разные ядра процессора, либо ОС переключает потоки через определенные кванты времени.

Порядок исполнения и обработка данных в рамках одной задачи

Сначала выполнится присвоение значений параметров связанных напрямую в дереве объектов. Затем произойдет вызов программ: чтение данных, необходимых для работы, затем отработка алгоритма программы, запись значений в связанные параметры параметров

Если элементы исполняются в одной и той же задаче узла, то первым исполнится тот элемент, который находится выше в дереве MasterSCADA 4D: присвоение параметров в дереве произойдет сверху вниз, а затем выполнятся программы сверху вниз. Изменить порядок элементов в дереве можно при помощи контекстного меню элемента и горячих клавиш CTRL+ ↓, CTRL+ ↑. Подробное описание смотрите в разделе Дерево MasterSCADA 4D.

Если объект имеет дочерние объекты, то сначала выполнятся присвоения связанных параметров по всем объектам (родительскому и дочерним), а затем выполнятся по очереди все программы, принадлежащие этим объектам - сначала родительские сверху вниз, а затем дочерних тоже сверху вниз.

Передача данных между задачами

Обмен данными между задачами осуществляется либо через параметры узла (глобальные переменные), либо через глобальный массив.

Передача данных через глобальный массив

Глобальный массив - это область памяти, где хранятся текущие значения определенных параметров:

- параметры у которых есть связи с параметрами из других задач (как правило, если параметр канала протокола связан с параметром объекта),
- параметры на которые ссылаются внешние каналы
- параметры, которые используются в других узлах

В начале цикла каждая задача считывает значения из глобального массива, в конце цикла - записывает значения указанных выше параметров в глобальный массив.

Глобальный массив используется задачей межузлового обмена, задачами SLAVE-протоколов, которые используют для передачи данных внешние каналы (протоколы, которые используются все дерево MasterSCADA формируют данные для сервера в рамках своих задач), а также задачами узла для передачи данных простых типов.

Глобальные переменные

Глобальные переменные (параметры узла), доступные всегда для всех задач, могут в проекте использоваться двумя способами:

- непосредственная связь между глобальным параметром и другим параметром проекта - в этом случае они всегда будут обрабатываться вначале цикла задачи, связанного параметра.
- прямой вызов в программе ST - в этом случае обработка произойдет непосредственно в момент чтения или записи

Важно! В текущей версии MasterSCADA 4D прямая передача сложных типов данных (структур, массивов) между задачами невозможна. Например, если в задаче протокола было получено значение типа Структура, то его нужно связать сначала с глобальным параметром Узла, а затем уже глобальный параметр связывать напрямую с параметром задачи узла или непосредственно использовать в программах ST

Порядок получения и передачи данных между клиентом визуализации и исполнительной системой

При открытии окна клиент подписывается к исполнительной системе на получение необходимых данных. Исполнительная система будет присылать данные в зависимости от периода той задачи, в которой запрашиваемые данные обрабатываются. Обновляться данные в окне будут с периодом, указанных в настройках Шаблона экрана.

10.5. ОШИБКИ КЛЮЧА ЗАЩИТЫ

В качестве ключей защиты используются продукты Guardant. Независимо от того используется ли реальный порт USB (локальный или сетевой), либо программный ключ защиты (в этом случае создаются виртуальные USB-порты и эмулируется работа реального ключа) исполнительная система проверяет периодически доступность ключа. В случае его отсутствия формируются сообщения. Проверка по умолчанию производится раз в 30 минут (если на производстве коммерческой версии исполнительной системы не было установлено другое значение). В случае если три проверки подряд были неудачными, то работа в режиме исполнения прекращается. При каждой неудачной проверке формируется сообщение, в котором указывается оставшееся время, которое исполнительная система может проработать без ключа.

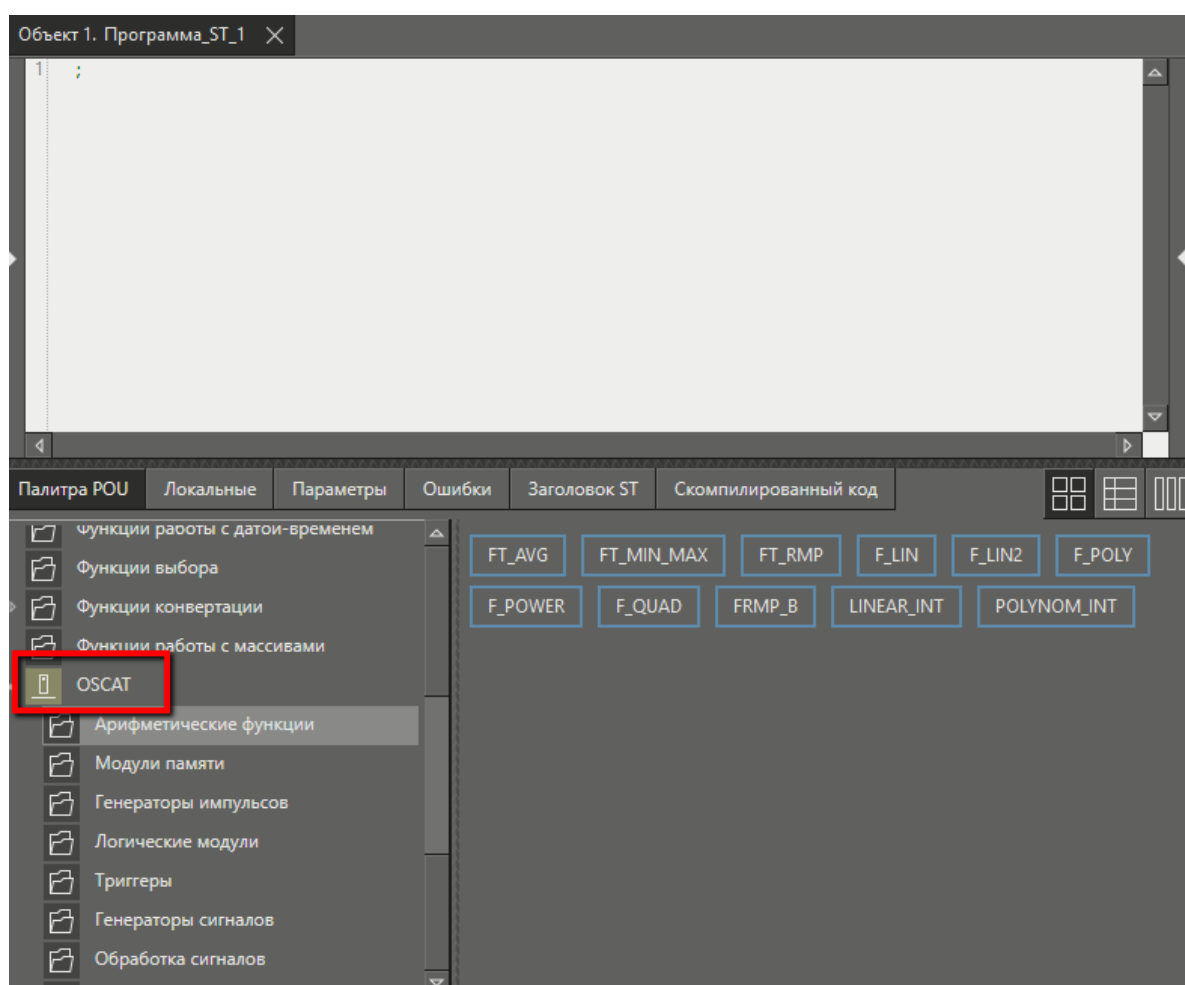
При ошибках в работе ключа формируются сообщения типа Системное сообщение

11. БИБЛИОТЕКА OSCAT

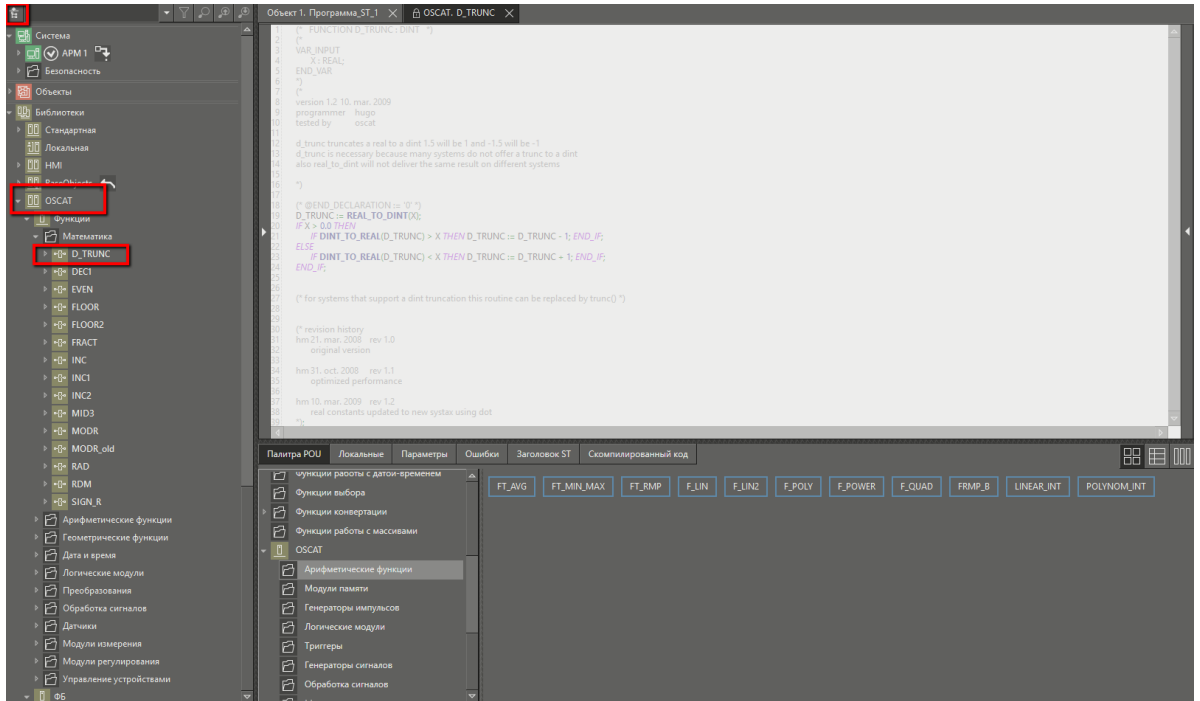
В состав MasterSCADA 4D включена библиотека компании OSCAT (Open Source Community for Automation Technology; библиотека и ее описание доступны для свободной загрузки на сайте <http://www.oscat.de>). Библиотека содержит функции с открытым исходным кодом.

Библиотека была протестирована, код функций и функциональных блоков был откорректирован.

Библиотека OSCAT доступна по умолчанию для использования в проектах MasterSCADA 4D.

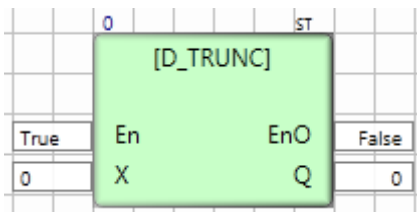


Чтобы открыть ST-код функции/ФБ библиотеки OSCAT, нужно дважды щелкнуть функцию/ФБ в дереве. ST-код содержит краткое описание Ф/ФБ, сделанное программистом (на английском языке).



11.1.1. OSC.AT.ФУНКЦИИ.МАТЕМАТИКА

11.1.1.1. D_TRUNC (OSC.AT)

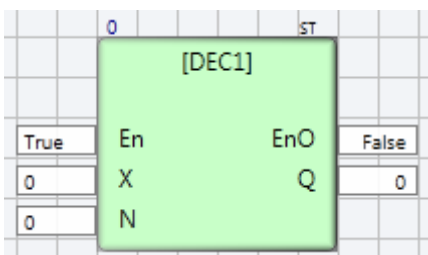


Тип данных X – REAL, Q – DINT.

Аналог TRUNC, целая часть к нулю.

Например, $d_trunc(5.9)=d_trunc(5.1)=5$; $d_trunc(-5.9)=d_trunc(-5.1)=-5$.

11.1.1.2. DEC1 (OSC.AT)

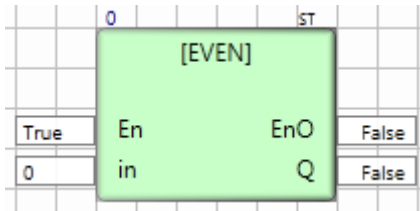


Тип данных X, N и Q – INT.

Если $X=0$, то $Q=N-1$, иначе $Q=X-1$.

При связи $Q \rightarrow X$ и $N > 0$ функция работает как генератор убывающего пилообразного сигнала ($N-1, N-2 \dots 0, N-1 \dots$).

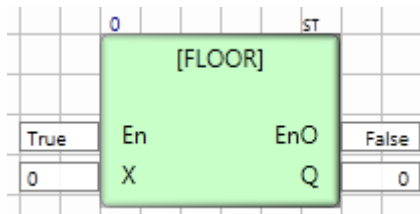
11.1.1.3. EVEN (OSCAT)



Тип данных IN – DINT, Q – BOOL.

Если IN – четное (в т.ч. 0), Q=TRUE, иначе Q=FALSE.

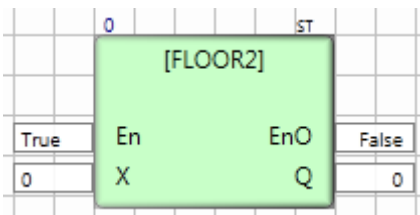
11.1.1.4. FLOOR (OSCAT)



Тип данных X – REAL, Q – INT.

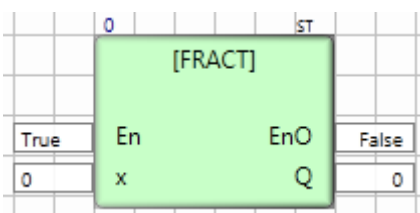
Функция возвращает наибольшее целое число, меньшее или равное X:
 $\text{floor}(5.1)=\text{floor}(5.9)=5$, $\text{floor}(-5.1)=\text{floor}(-5.9)=-6$.

11.1.1.5. FLOOR2 (OSCAT)



Аналог FLOOR (OSCAT), но тип данных Q – DINT.

11.1.1.6. FRACT (OSCAT)

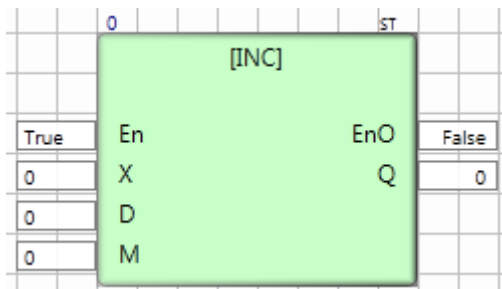


Тип данных X – REAL, Q – REAL.

Если $\text{abs}(X) < 2.14e9$, функция возвращает дробную часть X ($\text{fract}(3.14) = 0.14$, $\text{fract}(-3.14) = -0.14$), в противном случае функция возвращает 0.

Данная функция использует функцию D_TRUNC (OSCAT).

11.1.1.7. INC (OSCAT)



Тип данных X , D , M и Q – INT.

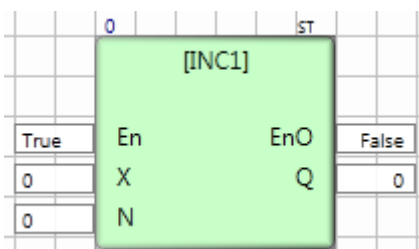
Функция используется при обращении к элементам массива, она увеличивает X на D и предотвращает выход индекса $(X+D)$ за M :

- если $0 \leq (X+D) \leq M$, $Q := X+D$;
- при выходе значения $(X+D)$ за границы диапазона $[0, M]$ значение Q вычисляется по правилам вычитания битовых строк (см. Особенности битовых строк): $Q := X+D-M-1$.

Примеры: $\text{INC}(3, 2, 5) = 5$, $\text{INC}(4, 2, 5) = 0$.

При любых неотрицательных X , D и M значение Q лежит в диапазоне $[0, M]$.

11.1.1.8. INC1 (OSCAT)

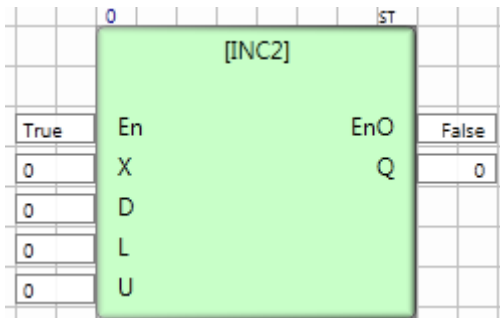


Тип данных X , N и Q – INT.

Если $X \geq N-1$, $Q=0$, иначе $Q=X+1$ (т.е. выход Q может принять одно из N различных значений от 0 до $N-1$).

При связи $Q \rightarrow X$ и $N > 0$ функция работает как генератор пилообразного сигнала (0, 1, 2 .. $N-1$, 0, 1 .. – ср. DEC1 (OSCAT)).

11.1.1.9. INC2 (OSCAT)



Тип данных X, D, L, U и Q – INT.

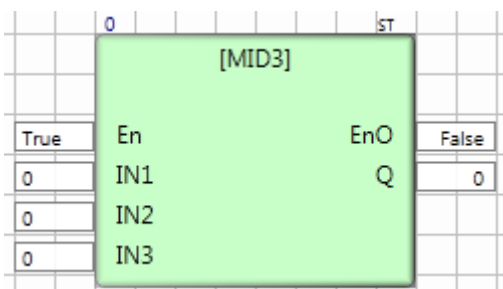
Функция используется при обращении к элементам массива, она увеличивает X на D и предотвращает выход индекса (X+D) за U:

- если $L \leq (X+D) \leq U$, $Q := X+D$;
- если $(X+D) > U$, $Q := L + (X+D-U-1)$ (ср. Особенности битовых строк);
- если $(X+D) < L$, $Q := U - (L-X-D+1)$ (ср. Особенности битовых строк).

Примеры: $INC2(9, 2, 10, 20) = 11$, $INC2(19, 2, 10, 20) = 10$.

При любых неотрицательных X, D, L и U ($U > L$) Q лежит в диапазоне [L, M].

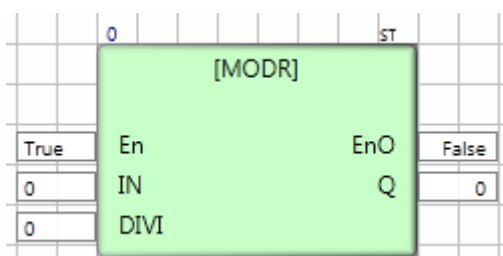
11.1.1.10. MID3 (OSCAT)



Тип данных IN1, IN2, IN3 и Q – REAL.

Функция возвращает значение входа, расположенное на числовой оси между двумя другими значениями входов. Например, $mid(-3, 9, 2) = 2$.

11.1.1.11. MODR (OSCAT)



Тип данных IN, DIVI и Q – REAL.

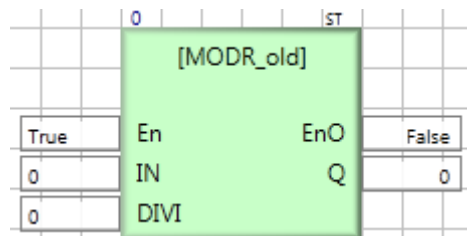
Функция работает по алгоритму, который используется в ФБ-генераторах для сдвига генерируемого сигнала по фазе (см., например, GEN_SIN (OSCAT)):

$$Q := IN - DIVI * FLOOR2(IN/DIVI)$$

Если DIVI=0, Q=0.

До замены в алгоритме D_TRUNC (OSCAT) на FLOOR2 (OSCAT) функция MODR возвращала остаток от деления значений FLOAT. После замены MODR возвращает корректный остаток от деления IN на DIVI только в том случае, если IN и DIVI имеют одинаковый знак. Чтобы сохранить в библиотеке алгоритм взятия остатка от деления значений FLOAT, добавлена восстановленная функция – MODR_old (OSCAT).

11.1.1.12. MODR_OLD (OSCAT)



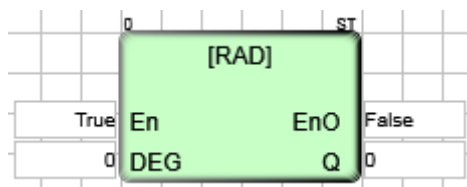
Тип данных IN, DIVI и Q – REAL.

Функция возвращает остаток от деления значений FLOAT (IN/DIVI). Например, MODR_old(5.75, -2.5):=0.75.

Если DIVI=0, Q=0.

MODR_old использует функцию D_TRUNC (OSCAT).

11.1.1.13. RAD (OSCAT)



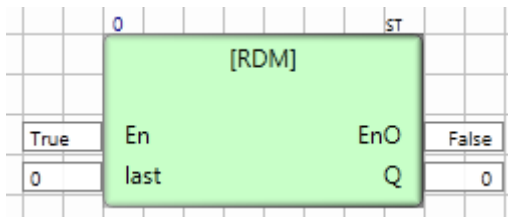
Тип данных входа DEG и выхода Q – REAL.

Функция переводит угловые градусы ($00 \leq DEG < 3600$) в радианы (Q).

Если $DEG \geq 3600$, перед конвертацией из DEG необходимое число раз вычитается 3600 – до тех пор, пока конвертируемое значение не окажется в диапазоне $[00, 3600)$. Таким образом, $RAD(3600) = RAD(00) = 0$, $RAD(9000) = RAD(1800) = \pi$.

Используемые функции: MODR (OSCAT).

11.1.1.14. RDM (OSCAT)



Тип данных LAST и Q – REAL.

Функция возвращает псевдослучайное число в интервале (0, 1).

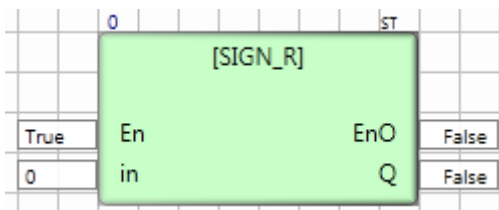
Если в программе требуется иметь несколько независимых генераторов псевдослучайных чисел, функцию RDM нужно вызывать с различными значениями входа LAST ($0 \leq \text{LAST} < 1$):

```
r116:=RDM(last:=0);
```

```
r117:=RDM(last:=0.5);
```

Используемые функции: FRACT (OSCAT), BIT_COUNT (OSCAT), T_PLC_MS (OSCAT).

11.1.1.15. SIGN_R (OSCAT)

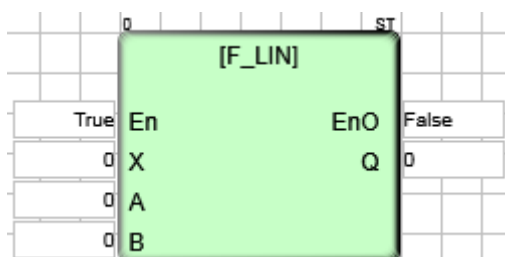


Тип данных IN – REAL, Q – BOOL.

Если $IN < 0$, $Q = \text{TRUE}$, иначе $Q = \text{FALSE}$.

11.1.2. OSCAT.ФУНКЦИИ.АРИФМЕТИЧЕСКИЕ ФУНКЦИИ

11.1.2.1. F_LIN (OSCAT)

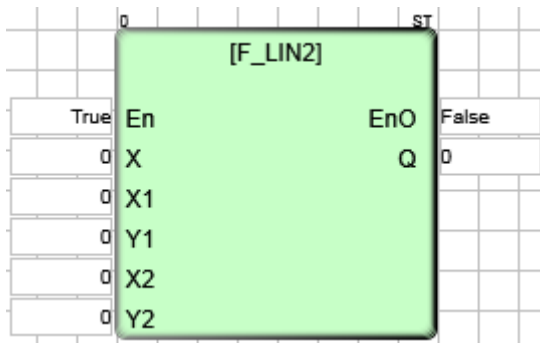


Тип данных входов X, A и B и выхода Q – REAL.

Функция F_LIN выполняет стандартное линейное преобразование:

$$Q := A * X + B$$

11.1.2.2. F_LIN2 (OSCAT)

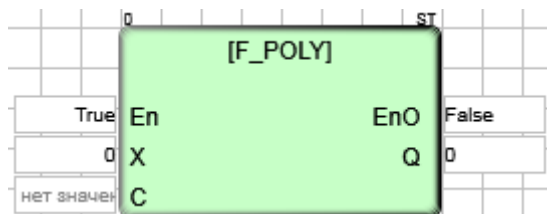


Тип данных входов и выхода – REAL.

Функция возвращает ординату точки прямой по абсциссе (X); прямая определяется двумя точками – (X1, Y1) и (X2, Y2):

$$F_LIN2 := (Y2 - Y1) / (X2 - X1) * (X - X1) + Y1$$

11.1.2.3. F_POLY (OSCAT)



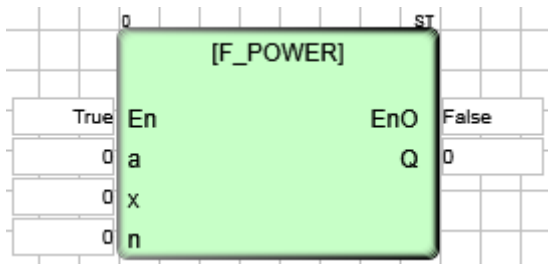
Тип данных входа X и выхода Q – REAL.

Вход C – массив 8 переменных REAL (индексы – 0 .. 7).

Функция возвращает значение полинома 7-й степени от X:

$$Q := C[7] \cdot x^7 + C[6] \cdot x^6 + C[5] \cdot x^5 + C[4] \cdot x^4 + C[3] \cdot x^3 + C[2] \cdot x^2 + C[1] \cdot x + C[0]$$

11.1.2.4. F_POWER (OSCAT)

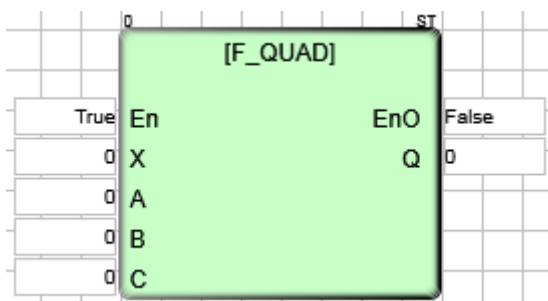


Тип данных входов и выхода – REAL.

$$Q := A \cdot X^N$$

Используемые функции: EXPT, возведение в степень.

11.1.2.5. F_QUAD (OSCAT)

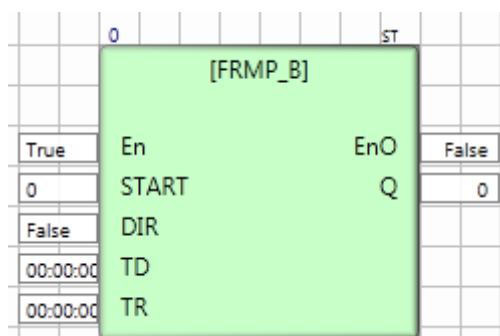


Тип данных входов и выхода – REAL.

Функция возвращает значение полинома 2-й степени от X:

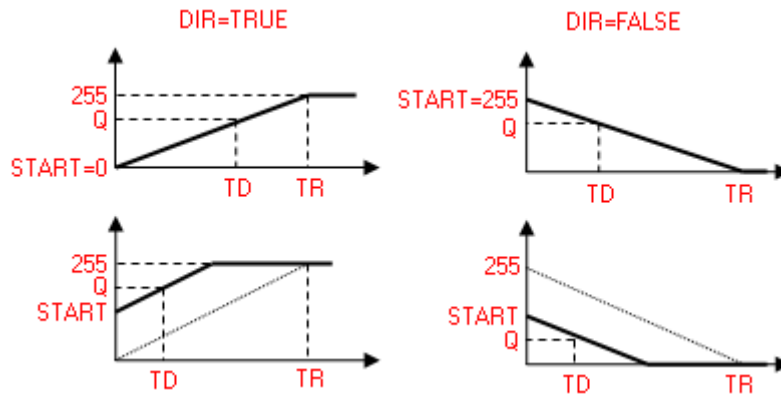
$$Q := A \cdot X^2 + B \cdot X + C$$

11.1.2.6. FRMP_B (OSCAT)

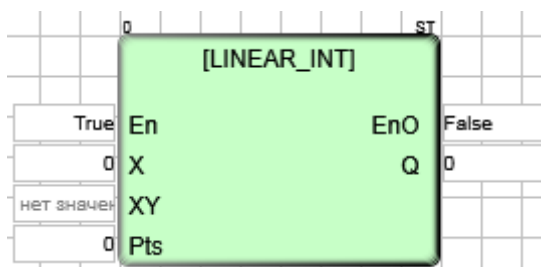


Тип данных START и Q – BYTE, DIR – BOOL, TD и TR – TIME.

FRMP_B вычисляет значение пилообразного сигнала в заданное время TD. TR задает период пилы (время изменения от 0 до 255), TD – истекшее время. Во всех случаях выходное значение ограничивается и лежит в диапазоне [0, 255]. Если DIR=TRUE, вычисляется возрастающая пила, а если DIR=FALSE – убывающая. START задает начальное значение.



11.1.2.7. LINEAR_INT (OSCAT)



Тип данных входа X и выхода Q – REAL.

Вход XY – двумерный массив 20-ти точек – (XY[1,0], XY[1,1]) .. (XY[20,0], XY[20,1]). Тип данных координат – REAL. XY[i,0] должно быть больше XY[i-1,0].

Тип данных входа PTS – INT (PTS должно быть целым и удовлетворять условию $3 \leq \text{PTS} \leq 20$).

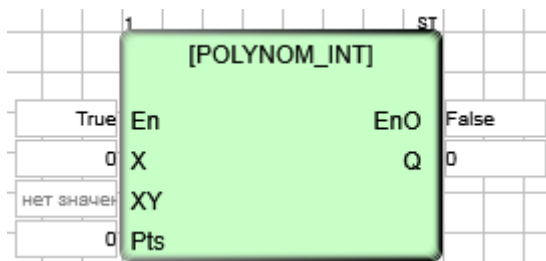
С помощью XY и PTS отрезок (XY[1,0], XY[PTS,0]) оси X разбивается на (PTS-1) интервалов (при максимально возможном PTS число интервалов равно 19). В каждом интервале LINEAR_INT работает так же, как F_LIN2 (OSCAT), т.е. вычисляет линейно-интерполированное значение ординаты по абсциссе (X).

При $X < \text{XY}[1,0]$ функция LINEAR_INT возвращает ординату как результат экстраполяции по прямой, заданной на первом интервале, т.е. точками (XY[1,0], XY[1,1]) и (XY[2,0], XY[2,1]).

При $X > \text{XY}[\text{PTS},0]$ функция LINEAR_INT возвращает ординату как результат экстраполяции по прямой, заданной на последнем интервале, т.е. точками (XY[PTS-1,0], XY[PTS-1,1]) и (XY[PTS,0], XY[PTS,1]). Например, при PTS=3 и следующих точках XY функция LINEAR_INT(30) возвращает 70, а не 50:

xy	
▼ [1]	
[1,0]	0
[1,1]	0
▼ [2]	
[2,0]	10
[2,1]	10
▼ [3]	
[3,0]	20
[3,1]	40
▼ [4]	
[4,0]	30
[4,1]	50
▼ [5]	
[5,0]	40
[5,1]	60
▶ [6]	

11.1.2.8. POLYNOM_INT (OSCAT)



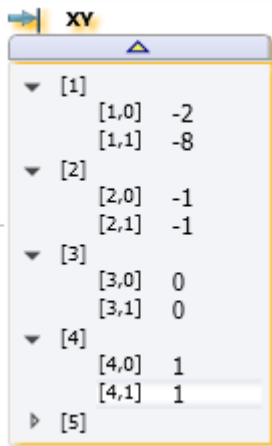
Тип данных входа X и выхода Q – REAL.

Вход XY – двумерный массив 5-ти точек – (XY[1,0], XY[1,1]) .. (XY[5,0], XY[5,1]). Тип данных координат – REAL. XY[i,0] должно быть больше XY[i-1,0].

Тип данных входа PTS – INT (PTS может принимать только три значения: 3, 4 или 5).

POLYNOM_INT – аналог LINEAR_INT (OSCAT), но если передаточная характеристика LINEAR_INT – ломаная, то передаточная характеристика POLYNOM_INT – полином степени (PTS-1).

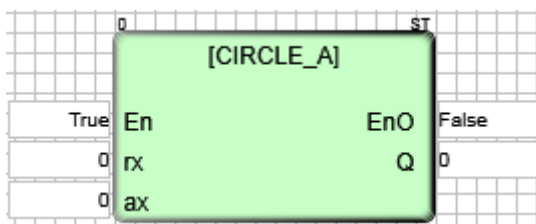
Например, при PTS=4 по следующим точкам XY строится передаточная функция $Q:=X^3$ и $POLYNOM_INT(2)=8$:



Заметим, что вне диапазона, заданного с помощью XY, весьма вероятны осцилляции передаточной характеристики (особенно при больших PTS), поэтому следует избегать подобного использования функции POLYNOM_INT.

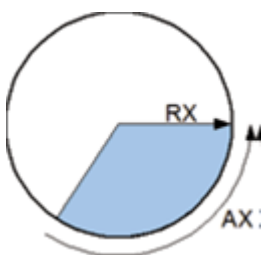
11.1.3. OSCAT.ФУНКЦИИ.ГЕОМЕТРИЧЕСКИЕ ФУНКЦИИ

11.1.3.1. CIRCLE_A (OSCAT)



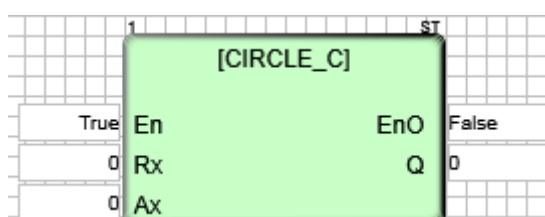
Тип данных входов и выхода – REAL.

Функция вычисляет площадь сектора круга радиуса RX:



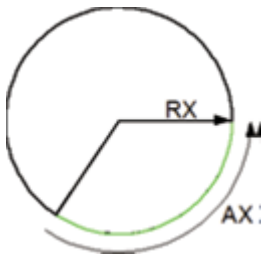
AX задает угол сектора (в градусах).

11.1.3.2. CIRCLE_C (OSCAT)



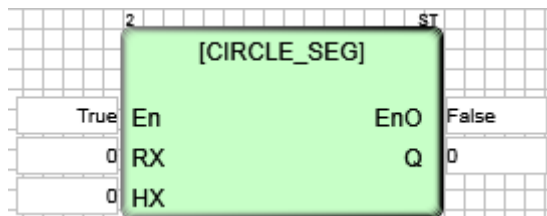
Тип данных входов и выхода – REAL.

Функция вычисляет длину дуги окружности радиуса RX:



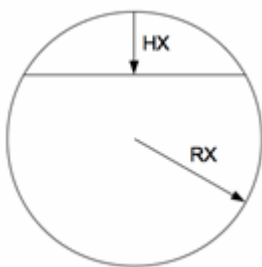
AX задает угол дуги (в градусах).

11.1.3.3. CIRCLE_SEG (OSCAT)

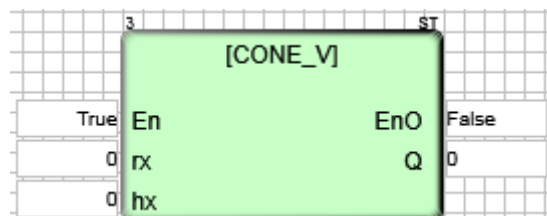


Тип данных входов и выхода – REAL.

Функция вычисляет площадь сегмента круга. RX – радиус круга, HX – высота сегмента:

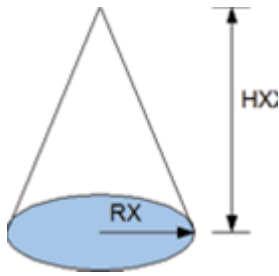


11.1.3.4. CONE_V (OSCAT)

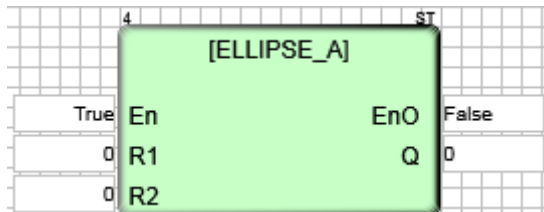


Тип данных входов и выхода – REAL.

Функция вычисляет объем конуса. RX – радиус основания, HX – высота конуса:

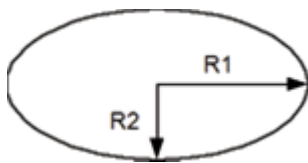


11.1.3.5. ELLIPSE_A (OSCAT)

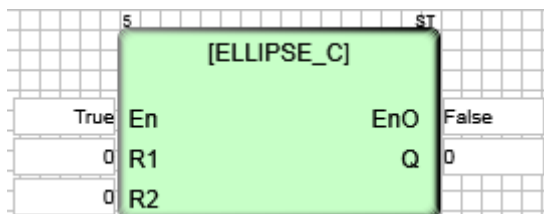


Тип данных входов и выхода – REAL.

Функция вычисляет площадь эллипса по большой (R1) и малой (R2) полуосям:

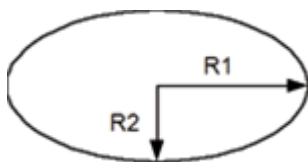


11.1.3.6. ELLIPSE_C (OSCAT)

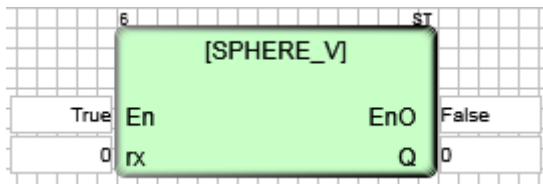


Тип данных входов и выхода – REAL.

Функция вычисляет длину эллипса по большой (R1) и малой (R2) полуосям:



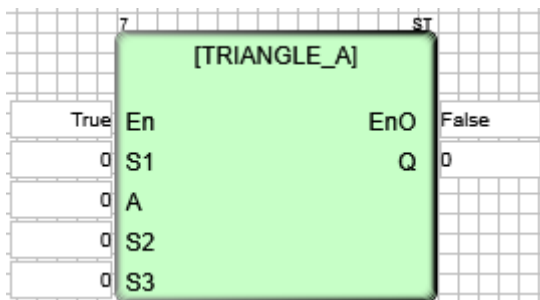
11.1.3.7. SPHERE_V (OSCAT)



Тип данных входа и выхода – REAL.

Функция вычисляет объем шара радиуса RX.

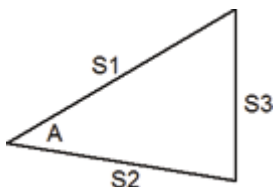
11.1.3.8. TRIANGLE_A (OSCAT)



Тип данных входов и выхода – REAL.

Функция вычисляет площадь произвольного треугольника, используя два алгоритма (S1, S2 и S3 – длины сторон, A – угол в градусах между S1 и S2):

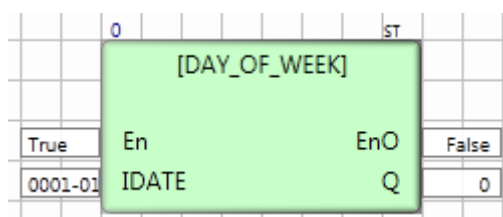
- ▶ если $A=0$ – по трем сторонам;
- ▶ если $A \neq 0$ – по двум сторонам S1 и S2 и углу A между ними.



Используемые функции: RAD (OSCAT).

11.1.4. OSCAT.ФУНКЦИИ.ДАТА И ВРЕМЯ

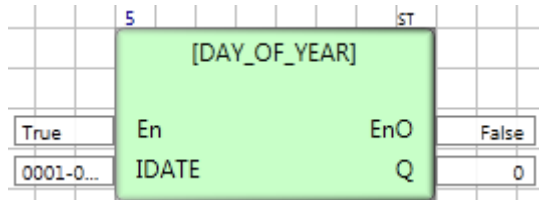
11.1.4.1. DAY_OF_WEEK (OSCAT)



Тип данных IDATE – DATE, Q – INT.

Функция вычисляет день недели по дате IDATE в соответствии с ISO8601 (1 – понедельник .. 7 – воскресенье). Например, DAY_OF_WEEK(D#2007-1-8) = 1.

11.1.4.2. DAY_OF_YEAR (OSCAT)

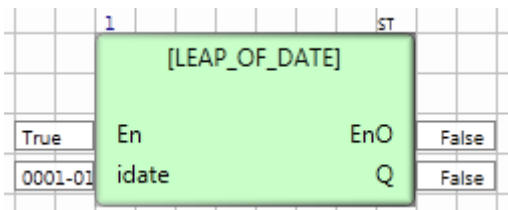


Тип данных IDATE – DATE, Q – INT.

Функция возвращает день года по дате с учетом високосных годов: DAY_OF_YEAR(D#2007-12-31) = 365, DAY_OF_YEAR(D#2008-12-31) = 366.

Диапазон допустимых годов – 1970- 2099.

11.1.4.3. LEAP_OF_DATE (OSCAT)

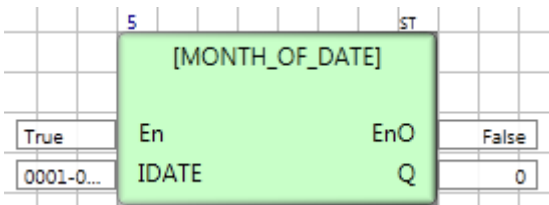


Тип данных IDATE – DATE, Q – BOOL.

Если год, заданный IDATE, – високосный, то Q=TRUE, иначе Q=FALSE.

Пример: LEAP_OF_DATE(D#2004-01-12) = TRUE.

11.1.4.4. MONTH_OF_DATE (OSCAT)

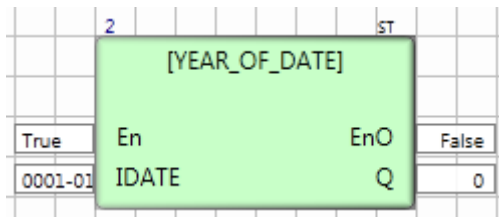


Тип данных входа IDATE – DATE, выхода Q – INT.

Функция возвращает месяц по дате в диапазоне годов 1970 - 2099.

Например, MONTH_OF_DATE(D#2007-12-31)=12, MONTH_OF_DATE(D#2006-1-1)=1.

11.1.4.5. YEAR_OF_DATE (OSCAT)



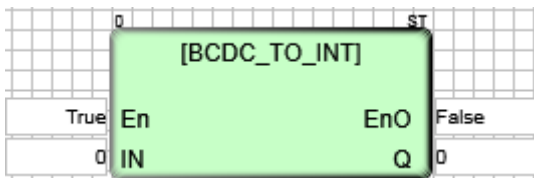
Тип данных IDATE – DATE, Q – INT.

Функция возвращает год по дате IDATE (1970 .. 2099).

Пример: YEAR_OF_DATE(D#2007-12-31) = 2007.

11.1.5. OSCAT.ФУНКЦИИ.ЛОГИЧЕСКИЕ МОДУЛИ

11.1.5.1. BCDC_TO_INT (OSCAT)

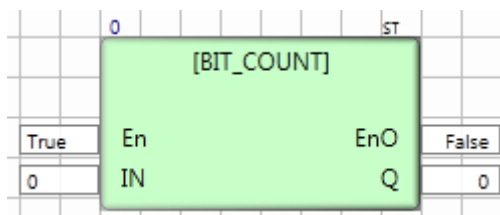


Тип данных входа IN – BYTE, на вход подается значение в формате BCD (не более 2 BCD-разрядов).

Тип данных выхода Q – INT.

BCDC_TO_INT обратна функции INT_TO_BCDC (OSCAT), т.е. преобразует формат значения из BCD в INT. Например, BCDC_TO_INT(231)=147 (231=2#1110_0111).

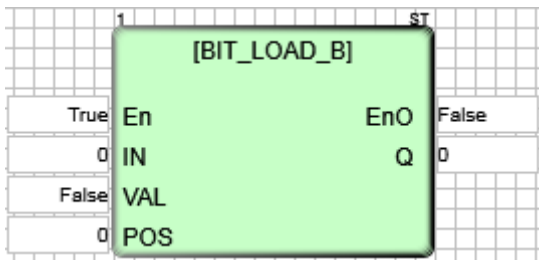
11.1.5.2. BIT_COUNT (OSCAT)



Тип данных IN – DWORD, WORD или BYTE, Q – INT.

Функция возвращает число установленных бит IN. Например, bit_count(6)=2.

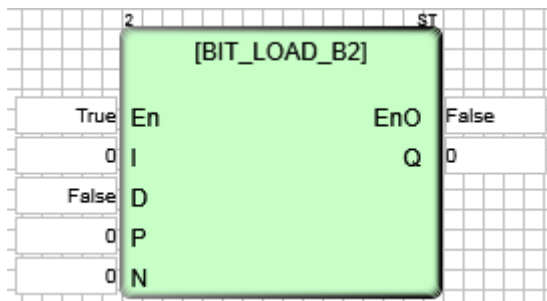
11.1.5.3. BIT_LOAD_B (OSCAT)



Тип данных входа IN – BYTE, VAL – BOOL, POS – INT, выхода Q – BYTE.

BIT_LOAD_B устанавливает бит POS значения IN в состояние VAL. Нумерация битов начинается с 0. Например, $\text{bit_load_b}(0, \text{true}, 3) = 8$.

11.1.5.4. BIT_LOAD_B2 (OSCAT)



Тип данных входа I – BYTE, D – BOOL, P и N – INT, выхода Q – BYTE.

BIT_LOAD_B2 предназначена для установки произвольного числа битов значения I в положение D. P задает начальный бит (нумерация битов начинается с 0), N – число изменяемых битов (если N=0, не изменяется ни один бит).

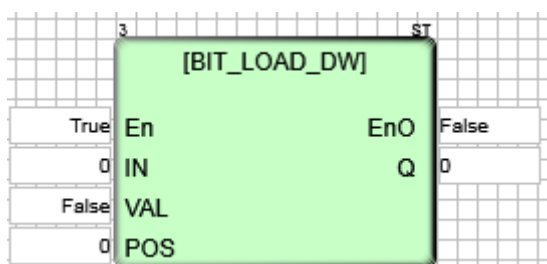
Если комбинация P и N задает номер бита больше 7, из этого номера нужное количество раз вычитается 8.

Примеры:

$\text{BIT_LOAD_B2}(2\#1111_0000, \text{TRUE}, 1, 2) = 2\#1111_0110$

$\text{BIT_LOAD_B2}(2\#1111_1111, \text{FALSE}, 7, 2) = 2\#0111_1110$

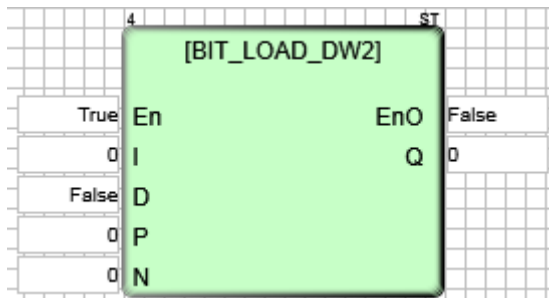
11.1.5.5. BIT_LOAD_DW (OSCAT)



Тип данных входа IN – DWORD, VAL – BOOL, POS – INT, выхода Q – DWORD.

BIT_LOAD_DW – аналог BIT_LOAD_B (OSCAT) для значений DWORD.

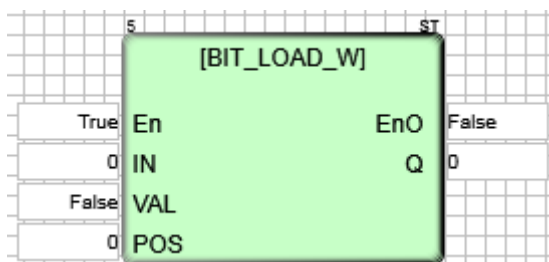
11.1.5.6. BIT_LOAD_DW2 (OSCAT)



Тип данных входа I – DWORD, D – BOOL, P и N – INT, выхода Q – DWORD.

BIT_LOAD_DW2 – аналог BIT_LOAD_B2 (OSCAT) для значений DWORD.

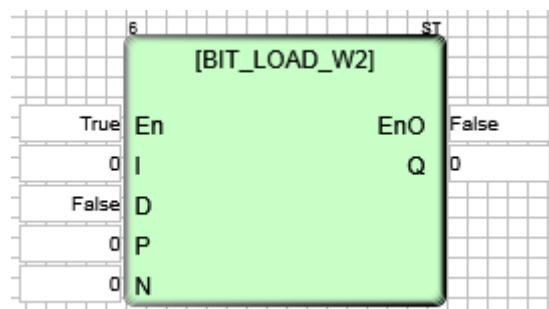
11.1.5.7. BIT_LOAD_W (OSCAT)



Тип данных входа IN – WORD, VAL – BOOL, POS – INT, выхода Q – WORD.

BIT_LOAD_W – аналог BIT_LOAD_B (OSCAT) для значений WORD.

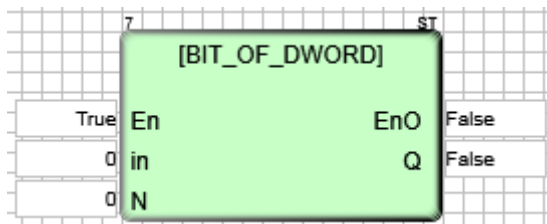
11.1.5.8. BIT_LOAD_W2 (OSCAT)



Тип данных входа I – WORD, D – BOOL, P и N – INT, выхода Q – WORD.

BIT_LOAD_W2 – аналог BIT_LOAD_B2 (OSCAT) для значений WORD.

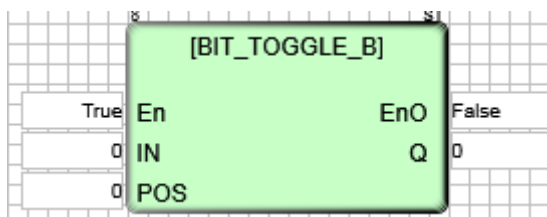
11.1.5.9. BIT_OF_DWORD (OSCAT)



Тип данных входа IN – DWORD, N – INT, выхода Q – BOOL.

Функция передает на выход Q значение бита N значения IN. Нумерация битов начинается с 0.

11.1.5.10. BIT_TOGGLE_B (OSCAT)



Тип данных входа IN – BYTE, POS – INT, выхода Q – BYTE.

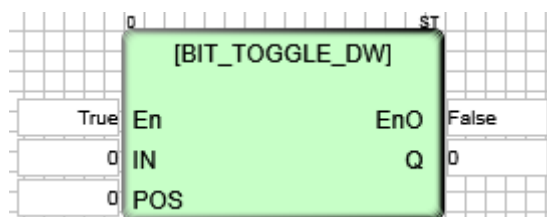
BIT_TOGGLE_B инвертирует бит POS в IN.

Примеры:

$\text{BIT_TOGGLE_W}(2\#0000_1111, 2) = 2\#0000_1011$

$\text{BIT_TOGGLE_W}(2\#0000_1111, 7) = 2\#1000_1111$

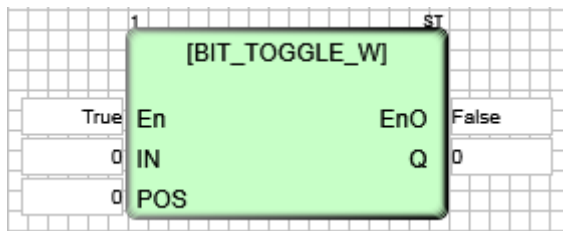
11.1.5.11. BIT_TOGGLE_DW (OSCAT)



Тип данных входа IN – DWORD, POS – INT, выхода Q – DWORD.

BIT_TOGGLE_DW – аналог BIT_TOGGLE_B (OSCAT) для значений DWORD.

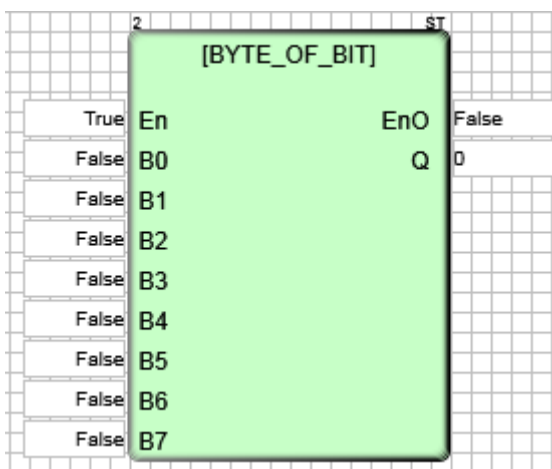
11.1.5.12. BIT_TOGGLE_W (OSCAT)



Тип данных входа IN – WORD, POS – INT, выхода Q – WORD.

BIT_TOGGLE_W – аналог BIT_TOGGLE_B (OSCAT) для значений WORD.

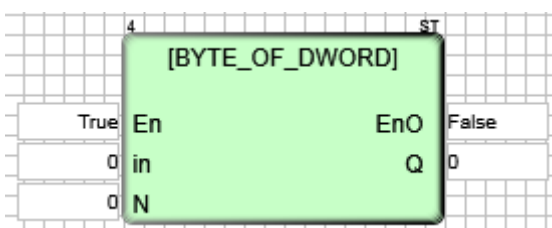
11.1.5.13. BYTE_OF_BIT (OSCAT)



Тип данных входов B0..B7 – BOOL, выхода Q – BYTE.

Значения B0..B7 задают значения соответствующих битов Q, т.е. BYTE_OF_BIT – это функция упаковки битов в байт.

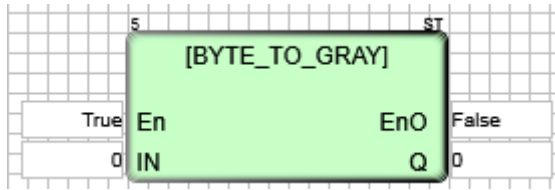
11.1.5.14. BYTE_OF_DWORD (OSCAT)



Тип данных IN – DWORD, выхода Q – BYTE.

Функция возвращает значение байта N значения IN (N может иметь значение 0, 1, 2 или 3).

11.1.5.15. BYTE_TO_GRAY (OSCAT)

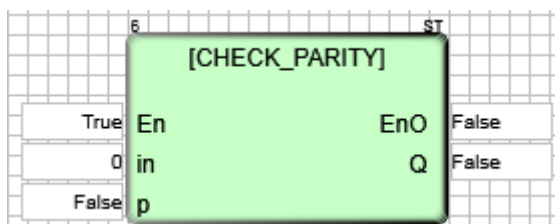


Тип данных входа IN и выхода Q – BYTE.

Функция конвертирует IN в код Грея:

$Q := IN \text{ XOR } SHR(IN, 1);$

11.1.5.16. CHECK_PARITY (OSCAT)



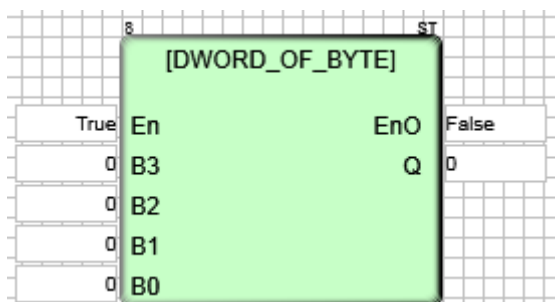
Тип данных входа IN – DWORD, P – BOOL, выхода Q – BOOL.

Функция анализирует биты IN и бит P как единую группу.

Если число установленных битов в группе – нечетное, Q:=FALSE.

Если число установленных битов в группе – четное, Q:=TRUE.

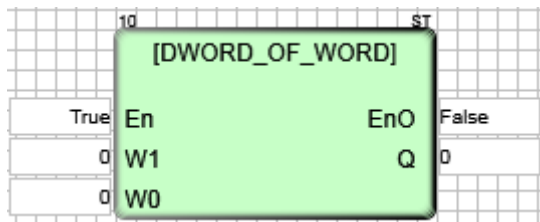
11.1.5.17. DWORD_OF_BYTE (OSCAT)



Тип данных входов B0..B3 – BYTE, выхода Q – DWORD.

Входы B0..B3 задают значения соответствующих байтов Q, т.е. DWORD_OF_BYTE – это функция упаковки байтов в двойное слово.

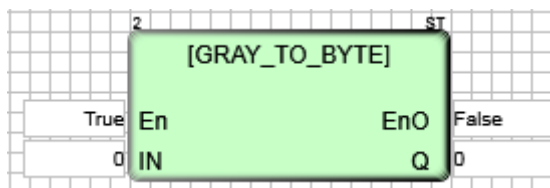
11.1.5.18. DWORD_OF_WORD (OSCAT)



Тип данных входов W0 и W1 – WORD, выхода Q – DWORD.

Входы W0 и W1 задают значения соответствующих слов Q, т.е. DWORD_OF_WORD – это функция упаковки слов в двойное слово.

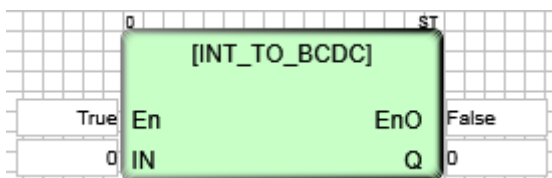
11.1.5.19. GRAY_TO_BYTE (OSCAT)



Тип данных входа IN и выхода Q – BYTE.

GRAY_TO_BYTE обратна функции BYTE_TO_GRAY (OSCAT), т.е. декодирует код Грея.

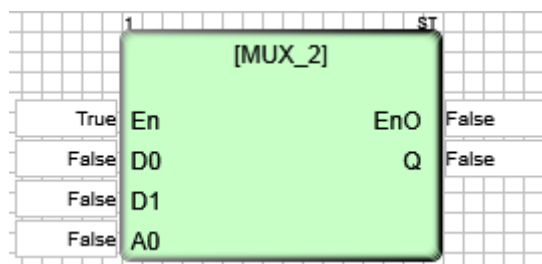
11.1.5.20. INT_TO_BCD (OSCAT)



Тип данных входа IN – INT, выхода Q – BYTE.

INT_TO_BCD конвертирует значение IN в формат BCD (для обратного преобразования используется BCD_TO_INT (OSCAT)).

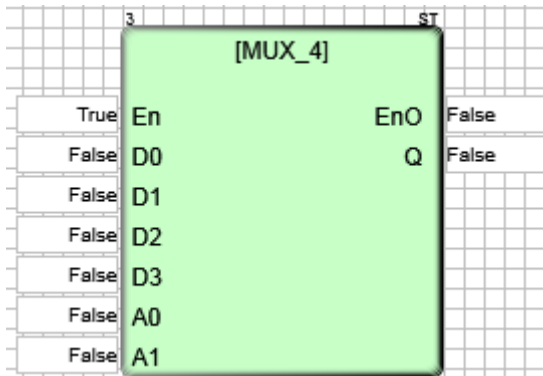
11.1.5.21. MUX_2 (OSCAT)



Тип данных входов и выхода – BOOL.

MUX_2 – это функция SEL, выбор из двух. На выход передается D0, если A=FALSE, и D1, если A=TRUE. Аналогом является также MUX_R2 (OSCAT).

11.1.5.22. MUX_4 (OSCAT)

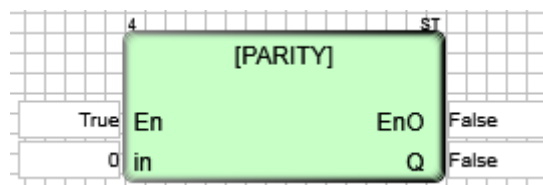


Тип данных входов и выхода – BOOL.

MUX_4 – это аналог MUX_R4 (OSCAT) для цифровых сигналов:

- при A1=FALSE: если A0=FALSE, Q=D0, если A0=TRUE, Q=D1.
- при A1=TRUE: если A0=FALSE, Q=D2, если A0=TRUE, Q=D3.

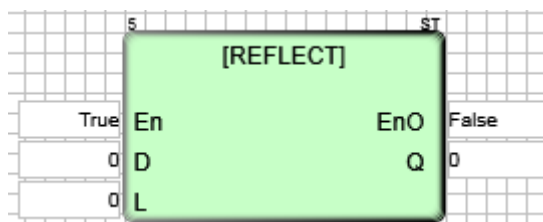
11.1.5.23. PARITY (OSCAT)



Тип данных IN – DWORD, выхода Q – BOOL.

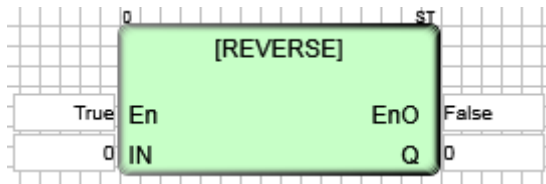
Q=TRUE, если число установленных бит в IN – нечетное (проверка на нечетность).

11.1.5.24. REFLECT (OSCAT)



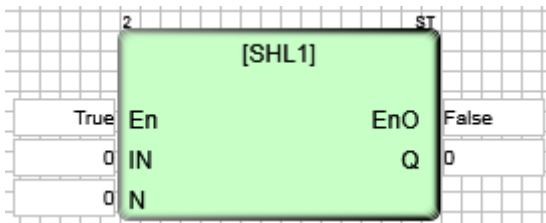
Тип данных выхода D – DWORD, L – INT, выхода Q – DWORD.

Функция зеркально изменяет порядок L младших битов в D. Например, REFLECT(2#1_0011_0001, 8) = 2#1_1000_1100.

11.1.5.25. REVERSE (OSCAT)

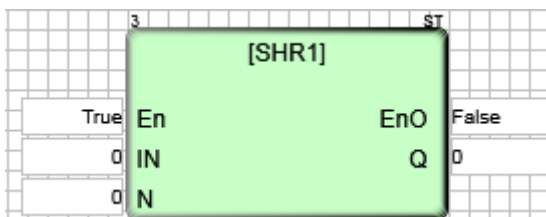
Тип данных входа IN и выхода Q – BYTE.

REVERSE зеркально изменяет порядок битов в IN: бит 7 становится битом 0, бит 6 – битом 1 и т.д. Например, $REVERSE(2\#1001_1110) = 2\#0111_1001$.

11.1.5.26. SHL1 (OSCAT)

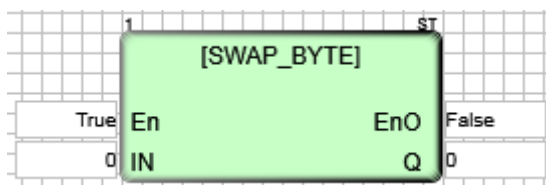
Тип данных входа IN – DWORD, N – INT, выхода Q – DWORD.

В отличие от стандартной функции SHL, сдвиг влево, функция SHL1 сдвигает значение IN на N бит влево, дополняя число справа единичными разрядами. Например, если $shl(1, 3)=8$, то $shl_1(1, 3)=15$.

11.1.5.27. SHR1 (OSCAT)

Тип данных входа IN – DWORD, N – INT, выхода Q – DWORD.

В отличие от стандартной функции SHR, сдвиг вправо, функция SHR1 сдвигает значение IN на N бит вправо, дополняя число слева единичными разрядами. Например, если $shr(16\#FFFFFFFF, 31)=1$, то $shr_1(16\#FFFFFFFF, k)=16\#FFFFFFFF$ при любом k.

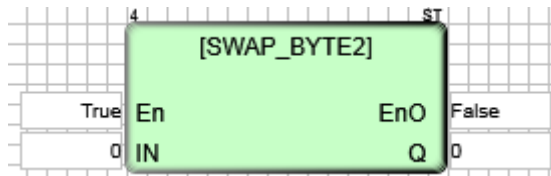
11.1.5.28. SWAP_BYTE (OSCAT)

Тип данных входа IN и выхода Q – WORD.

Функция выполняет перестановку байтов в IN.

Например, SWAP_BYTE(16#33df) = 16#df33.

11.1.5.29. SWAP_BYTE2 (OSCAT)

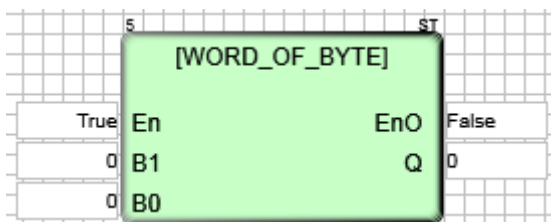


Тип данных входа IN и выхода Q – DWORD.

Функция зеркально изменяет порядок байтов в IN.

Например, SWAP_BYTE2(16#33df1122) = 16#2211df33.

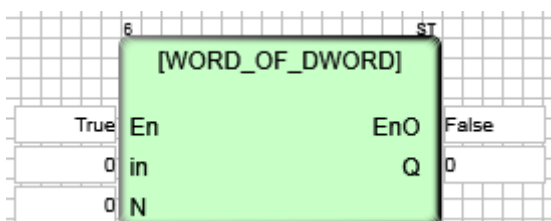
11.1.5.30. WORD_OF_BYTE (OSCAT)



Тип данных входов B0 и B1 – BYTE, выхода Q – WORD.

B0 и B1 задают значения соответствующих байтов выхода Q, т.е. WORD_OF_BYTE – это функция упаковки байтов в слово.

11.1.5.31. WORD_OF_DWORD (OSCAT)

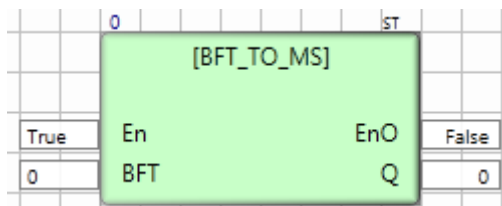


Тип данных входа IN – DWORD, N – BYTE, выхода Q – WORD.

Функция возвращает значение слова N значения IN (N = 0 или 1).

Например, WORD_OF_DWORD(65537, N)=1 при любом N.

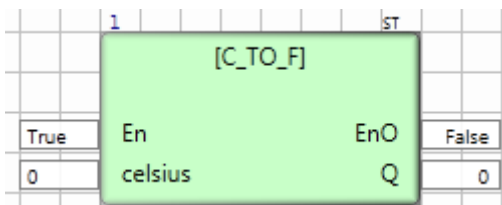
11.1.6. OSCAT.ФУНКЦИИ.ПРЕОБРАЗОВАНИЯ

11.1.6.1. BFT_TO_MS (OSCAT)

Тип данных BFT – INT, Q – REAL.

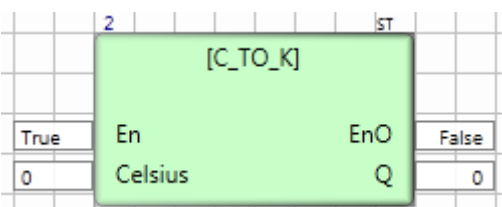
Функция переводит скорость ветра по бофоровой шкале в м/с по формуле

$$BFT_TO_MS = 0.836 * BFT^{**}(3/2) \text{ [m/s]}$$

11.1.6.2. C_TO_F (OSCAT)

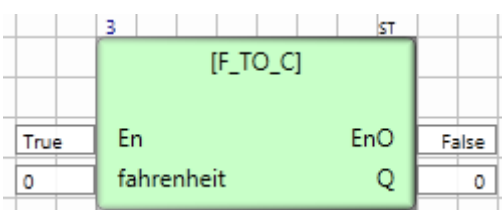
Тип данных CELSIUS и Q – REAL.

Функция переводит температуру по Цельсию в температуру по Фаренгейту.

11.1.6.3. C_TO_K (OSCAT)

Тип данных CELSIUS и Q – REAL.

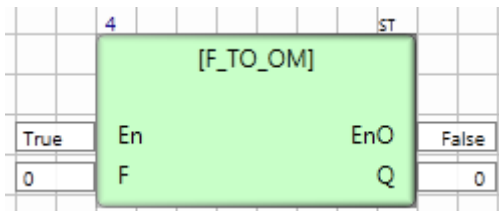
Функция переводит температуру по Цельсию в температуру по Кельвину.

11.1.6.4. F_TO_C (OSCAT)

Тип данных FAHRENHEIT и Q – REAL.

Функция переводит температуру по Фаренгейту в температуру по Цельсию.

11.1.6.5. F_TO_OM (OSCAT)

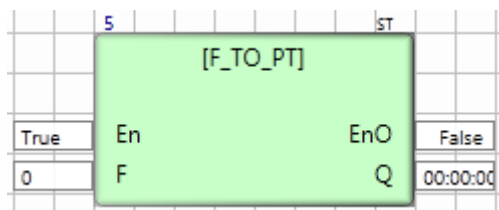


Тип данных F и Q – REAL.

Функция вычисляет циклическую частоту по заданной частоте:

$$Q := 2\pi F$$

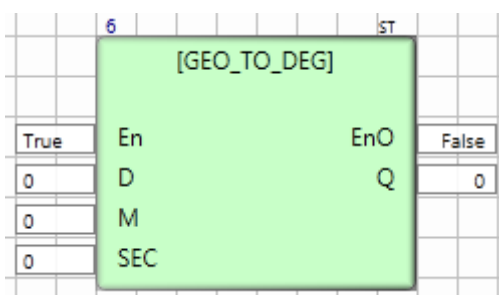
11.1.6.6. F_TO_PT (OSCAT)



Тип данных F – REAL, Q – TIME.

Функция вычисляет период по заданной частоте в герцах.

11.1.6.7. GEO_TO_DEG (OSCAT)



Тип данных D и M – INT, SEC и Q – REAL.

Функция переводит угловые меры, заданные в угловых градусах (D), минутах (M) и секундах (SEC), в угловые градусы с дробной частью. Например, GEO_TO_DEG(5,30,900.0)=5.75.

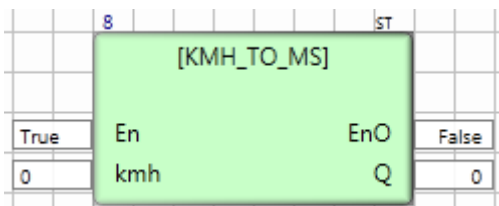
11.1.6.8. K_TO_C (OSCAT)



Тип данных KELVIN и Q – REAL.

Функция переводит температуру по Кельвину в температуру по Цельсию.

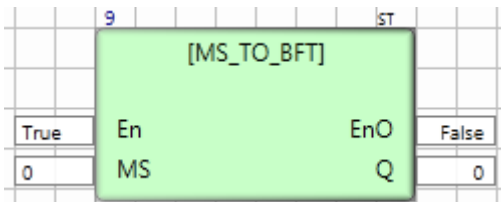
11.1.6.9. КМН_TO_MS (OSCAT)



Тип данных КМН и Q – REAL.

Функция переводит скорость в км/ч в м/с.

11.1.6.10. MS_TO_BFT (OSCAT)

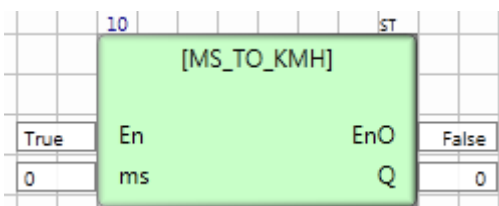


Тип данных MS – REAL, Q – INT.

Функция переводит скорость ветра в м/с в бофорову шкалу по формуле

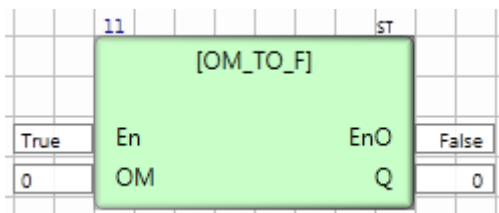
$$MS_TO_BFT = (MS * 1.196172)^{(2/3)}$$

11.1.6.11. MS_TO_KMH (OSCAT)



Тип данных MS и Q – REAL.

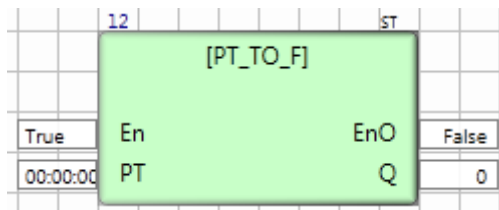
Функция переводит скорость в м/с в км/ч.

11.1.6.12. OM_TO_F (OSCAT)

Тип данных OM и Q – REAL.

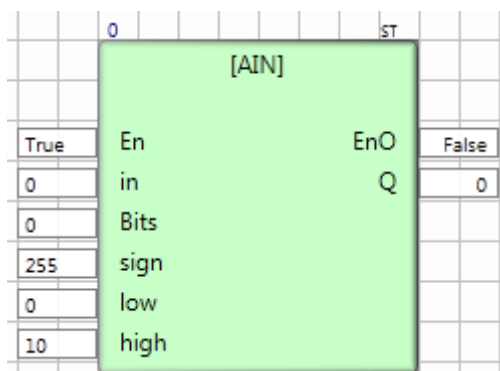
Функция вычисляет частоту по заданной циклической частоте:

$$Q := \frac{OM}{2\pi}$$

11.1.6.13. PT_TO_F (OSCAT)

Тип данных PT – TIME. Q – REAL.

Функция вычисляет частоту в герцах по заданному периоду PT:

11.1.7. OSCAT.ФУНКЦИИ.ОБРАБОТКА СИГНАЛОВ**11.1.7.1. AIN (OSCAT)**

Входы функции:

- IN (тип данных DWORD) – значение от АЦП;
- BITS (тип данных BYTE) – число бит, 16 для полного слова;
- SIGN (тип данных BYTE) – бит знака, 15 для бита 15;

- LOW (тип данных REAL) – нижняя граница диапазона АЦП;
- HIGH (тип данных REAL) – верхняя граница диапазона АЦП.

Выход функции:

- Q (тип данных REAL).

Выход АЦП обычно 16-разрядный (WORD) или 32-разрядный (DWORD), однако само аналого-цифровое преобразование, как правило, не обладает разрешением в 16 или 32 бита.

Кроме того, в результате аналого-цифрового преобразования входной диапазон (например, -10 .. +10 В) превращается в диапазон цифровых значений 0 .. 65535 (в случае 16-разрядного выхода АЦП).

Функция AIN – модель ЦАП, настраивается таким образом, чтобы преобразовать значение выхода АЦП в реально измеренную физическую величину. Функция может также извлечь и преобразовать знаковый бит из любого разряда.

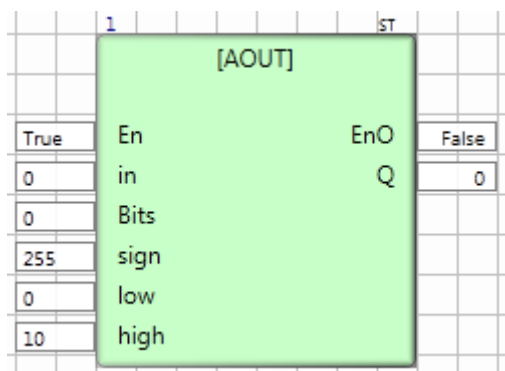
Конфигурационные параметры:

- BITS задает число бит входа IN, которые нужно обрабатывать. Для 12-разрядного АЦП должно быть задано BITS=12, что означает обработку только битов 0-11;
- SIGN задает, требуется ли обработка знакового бита, и где этот бит расположен в IN. SIGN=255 (значение по умолчанию) означает, что знаковый бит в IN отсутствует, SIGN=15 означает, что знаковый бит содержится в 15-м бите IN;
- LOW и HIGH ограничивают выходное значение функции. Если обработка знакового бита задана (SIGN < 255), LOW и HIGH должны быть положительными, в противном случае могут быть как положительными, так и отрицательными.

Примеры конфигурирования (см. также пример в описании ФБ AIN1 (OSCAT)):

- 12-разрядный АЦП без знакового бита с диапазоном [0, 10] – BITS=12, SIGN=255, LOW=0, HIGH=10;
- 14-разрядный АЦП со знаковым битом в 14-м разряде и диапазоном [-10, 10] – BITS=14, SIGN=14, LOW=0, HIGH=10;
- 24-разрядный АЦП без знакового бита с диапазоном [-10, 10] – BITS=24, SIGN = 255, LOW=-10, HIGH=10.

11.1.7.2. AOUT (OSCAT)



Входы функции:

- IN (тип данных REAL) – входное значение;
- BITS (тип данных BYTE) – число бит, 16 для полного слова;
- SIGN (тип данных BYTE) – бит знака, 15 для бита 15;
- LOW (тип данных REAL) – минимальное допустимое значение IN (нижняя граница диапазона ЦАП);
- HIGH (тип данных REAL) – максимальное допустимое значение IN (верхняя граница диапазона ЦАП).

Выходы функции:

- Q (тип данных DWORD) – значение для подачи на вход ЦАП.

Вход ЦАП обычно 16-разрядный (WORD) или 32-разрядный (DWORD), однако само цифро-аналоговое преобразование, как правило, не обладает разрешением в 16 или 32 бита.

ЦАП генерирует фиксированный выходной диапазон аналоговых значений (например, -10 .. +10 В), которому соответствует входной диапазон цифровых значений 0 .. 65535 (в случае 16-разрядного входа ЦАП).

Функция AOOUT – модель АЦП, конфигурируется для преобразования входа IN таким образом, чтобы на выходе AOOUT сформировалось цифровое значение, обеспечивающее на выходе ЦАП значение, равное IN. Кроме того, функция может вставить в любой разряд Q знаковый бит, если это требуется для ЦАП.

Конфигурационные параметры:

- BITS задает разрядность ЦАП. Для 12-разрядного ЦАП должно быть задано BITS=12;
- SIGN задает, требуется ли вставка знакового бита, и где этот бит должен быть расположен в Q. SIGN=255 (значение по умолчанию) означает, что знаковый бит не требуется, SIGN=15 означает, что знаковый бит будет вставлен в 15-й бит Q;

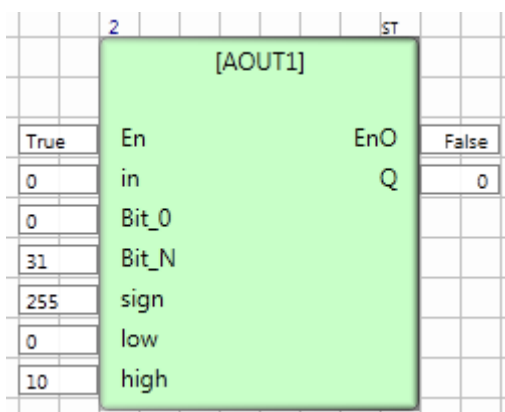
- LOW и HIGH ограничивают входное значение функции. Если обработка знакового бита задана (SIGN < 255), LOW и HIGH должны быть положительными, в противном случае могут быть как положительными, так и отрицательными.

Примеры конфигурирования:

- 12-разрядный ЦАП без знакового бита с выходным диапазоном [0, 10] – BITS=12, SIGN=255, LOW=0, HIGH=10;
- 14-разрядный ЦАП со знаковым битом в 14-м разряде и выходным диапазоном [-10, 10] – BITS=14, SIGN=14, LOW=0, HIGH=10;
- 24-разрядный ЦАП без знакового бита с выходным диапазоном [-10, 10] – BITS=24, SIGN = 255, LOW=-10, HIGH=10.

Используемые функции: SIGN_R (OSCAT).

11.1.7.3. AOUT1 (OSCAT)



AOUT1 – это расширенный аналог AOUT (OSCAT) .

Входы функции:

- IN (тип данных REAL) – входное значение;
- BIT_0 (тип данных INT) – позиция LSB в выходных данных;
- BIT_N (тип данных INT) – позиция MSB в выходных данных;
- SIGN (тип данных INT) – бит знака, 15 для бита 15;
- LOW (тип данных REAL) – минимальное допустимое значение IN (нижняя граница диапазона ЦАП);
- HIGH (тип данных REAL) – максимальное допустимое значение IN (верхняя граница диапазона ЦАП).

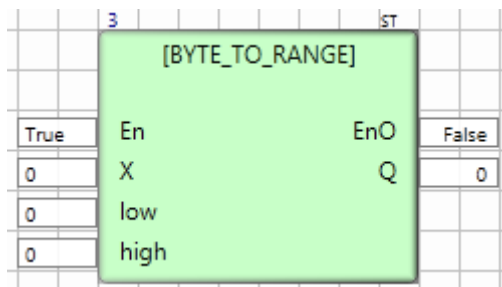
Выходы функции:

- Q (тип данных DWORD) – значение для подачи на вход ЦАП.

AOUT1 – модель АЦП, преобразует входное значение (REAL) в цифровое выходное значение (DWORD) для ЦАП или других цифровых устройств (см. пример в описании ФБ AIN1 (OSCAT)). С помощью конфигурационных параметров выходное значение функции может быть адаптировано для различных целей. Входное значение (IN) обрабатывается с использованием параметров LOW и HIGH, для выходных данных с помощью параметров BIT_0 и BIT_N устанавливается длина, равная BIT_N - BIT_0 + 1 (от LSB до MSB включительно, длина должна соответствовать разрядности ЦАП). Если позиция знакового бита задана (с помощью параметра SIGN), знак входного значения копируется в указанный разряд выходных данных.

Используемые функции: SIGN_R (OSCAT).

11.1.7.4. BYTE_TO_RANGE (OSCAT)

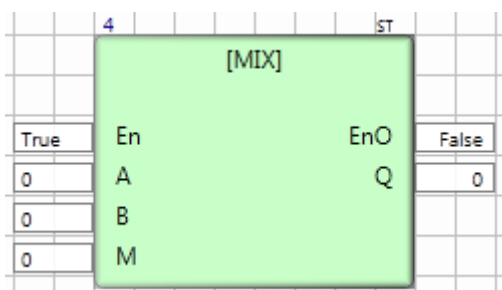


Тип данных X – BYTE, LOW, HIGH и Q – REAL.

Функция преобразует BYTE-значение входа X в значение REAL с масштабированием в диапазон [LOW, HIGH].

Например, для перевода всего диапазона BYTE (от 0 до 255) в проценты (от 0 до 100) можно использовать BYTE_TO_RANGE(X,0,100).

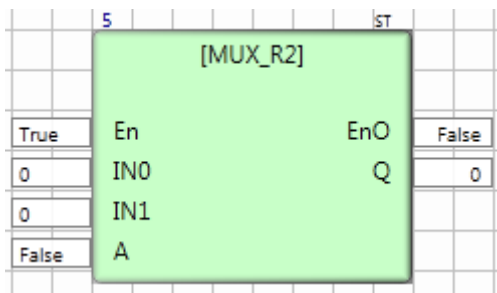
11.1.7.5. MIX (OSCAT)



Тип данных A, B, M и Q – REAL.

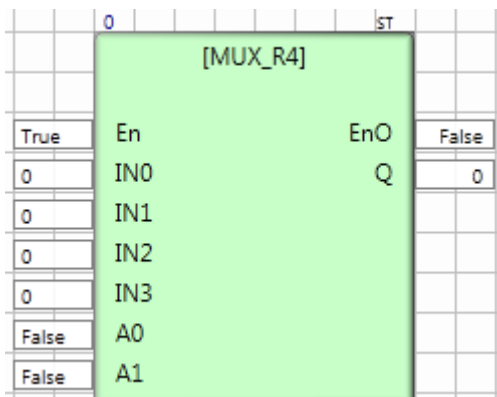
Функция представляет собой аналоговый микшер:

$$Q := (1 - M) * A + M * B;$$

11.1.7.6. MUX_R2 (OSCAT)

Тип данных A – BOOL, IN0, IN1 и Q – REAL.

Функция представляет собой аналоговый мультиплексор: если A=FALSE, Q=IN0, если A=TRUE, Q=IN1. Аналог функции SEL, выбор из двух.

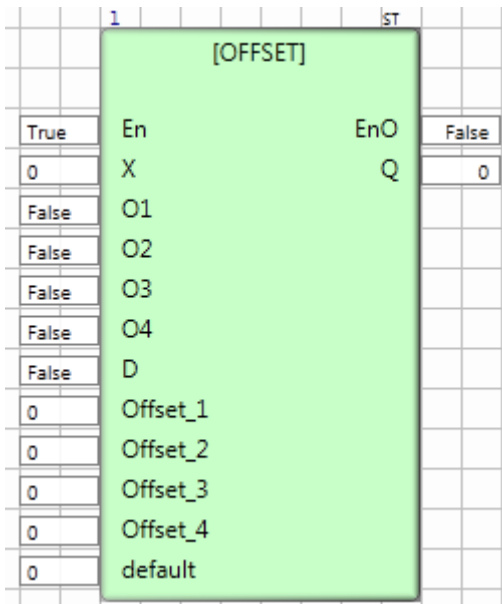
11.1.7.7. MUX_R4 (OSCAT)

Тип данных IN0, IN1, IN2, IN3 и Q – REAL, A0 и A1 – BOOL.

Функция представляет собой аналоговый мультиплексор:

- при A1=FALSE: если A0=FALSE, Q=IN0, если A0=TRUE, Q=IN1.
- при A1=TRUE: если A0=FALSE, Q=IN2, если A0=TRUE, Q=IN3.

11.1.7.8. OFFSET (OSCAT)

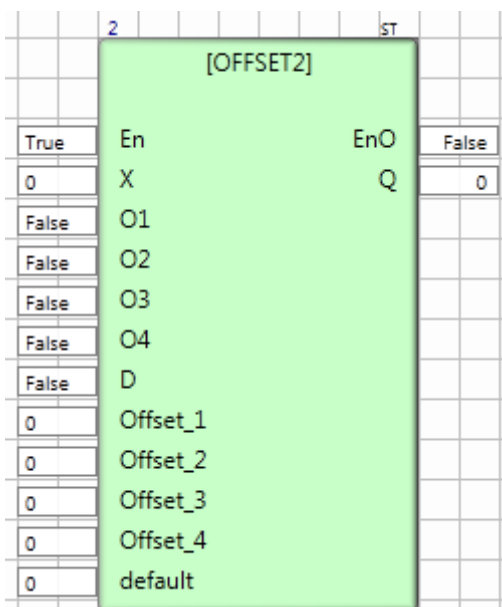


Тип данных X, OFFSET_1, OFFSET_2, OFFSET_3, OFFSET_4, DEFAULT и Q – REAL, O1, O2, O3, O4 и D – BOOL.

Функция работает по следующему алгоритму:

- если D=FALSE, Q равно сумме X и тех входов OFFSET_<i>, для которых O<i>=TRUE;
- если D=TRUE, Q равно сумме DEFAULT и тех входов OFFSET_<i>, для которых O<i>=TRUE.

11.1.7.9. OFFSET2 (OSCAT)

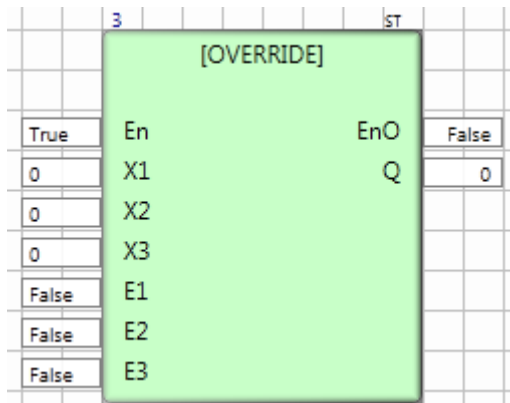


Тип данных X, OFFSET_1, OFFSET_2, OFFSET_3, OFFSET_4, DEFAULT и Q – REAL, O1, O2, O3, O4 и D – BOOL.

Функция работает по следующему алгоритму:

- если D=FALSE, Q равно сумме X и входа OFFSET с максимальным индексом, для которого соответствующий вход O=TRUE;
- если D=TRUE, Q равно сумме DEFAULT и входа OFFSET с максимальным индексом, для которого соответствующий вход O=TRUE.

11.1.7.10. OVERRIDE (OSCAT)

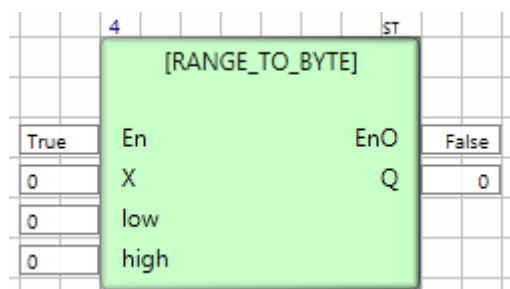


Тип данных X1, X2, X3 и Q – REAL, E1, E2 и E3 – BOOL.

Входы E1..E3 разрешают (TRUE) или запрещают (FALSE) обработку входов X1..X3 (соответственно).

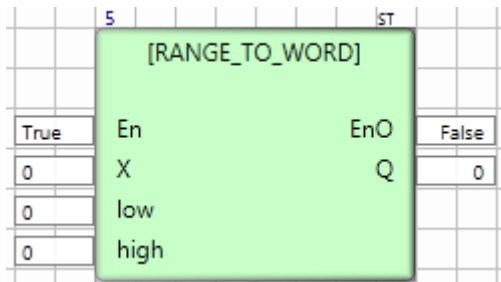
На выход Q передается значение входа X с максимальным абсолютным значением среди обрабатываемых входов X.

11.1.7.11. RANGE_TO_BYTE (OSCAT)



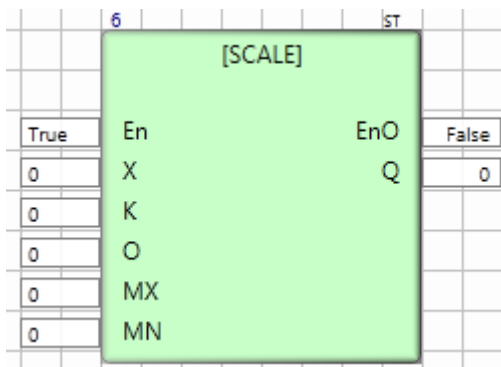
Эта функция обратна функции BYTE_TO_RANGE (OSCAT).

11.1.7.12. RANGE_TO_WORD (OSCAT)



Эта функция обратна функции WORD_TO_RANGE (OSCAT).

11.1.7.13. SCALE (OSCAT)

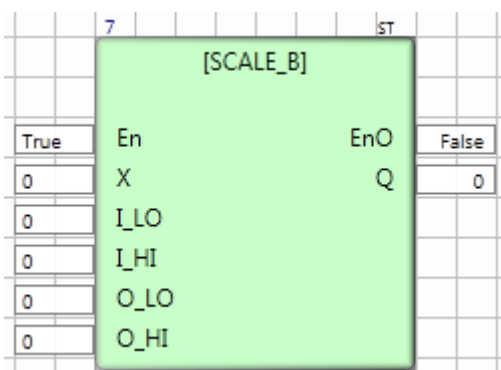


Тип данных входов и выхода – REAL.

SCALE умножает X на K и прибавляет смещение O. Результат ограничивается значениями MN и MX и присваивается выходу Q:

$$\text{SCALE} = \text{LIMIT}(\text{MN}, \text{X} * \text{K} + \text{O}, \text{MX})$$

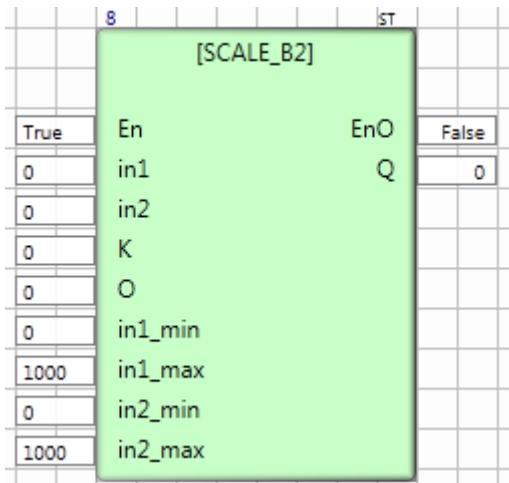
11.1.7.14. SCALE_B (OSCAT)



Тип данных X, I_LO и I_HI – BYTE, O_LO, O_HI и Q – REAL.

Ограниченное параметрами I_LO и I_HI входное BYTE-значение X масштабируется (диапазон [I_LO, I_HI] переводится в диапазон [O_LO, O_HI]) и преобразуется в значение REAL.

Например, SCALE_B(75,50,100,-100,200):=50.

11.1.7.15. SCALE_B2 (OSCAT)

Входы функции:

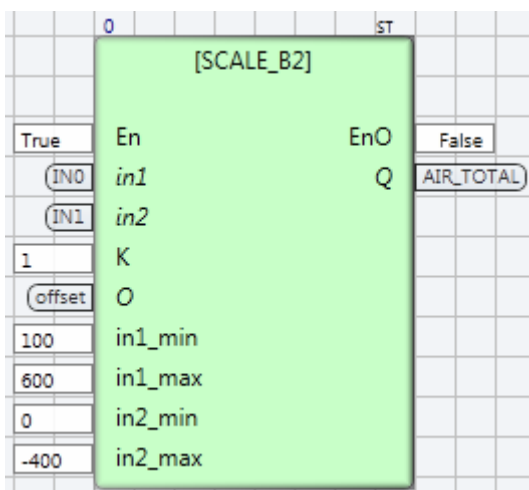
- IN1 (тип данных BYTE) – первый вход;
- IN2 (тип данных BYTE) – второй вход;
- K (тип данных REAL) – множитель;
- O (тип данных REAL) – смещение;
- IN1_MIN (тип данных REAL) – нижняя граница диапазона, в который масштабируется IN1;
- IN1_MAX (тип данных REAL) – верхняя граница диапазона, в который масштабируется IN1;
- IN2_MIN (тип данных REAL) – нижняя граница диапазона, в который масштабируется IN2;
- IN2_MAX (тип данных REAL) – верхняя граница диапазона, в который масштабируется IN2.

SCALE_B2 масштабирует IN1 и IN2, суммирует результаты, затем умножает сумму на множитель K и прибавляет смещение O:

$$Q := (in1 * (in1_max - in1_min) / 255 + in1_min + in2 * (in2_max - in2_min) / 255 + in2_min) * K + O$$

SCALE_B2 может быть использована, например, для вычисления расхода воздуха в вентиляционной системе, а также в любой системе, в которой есть смеситель, и требуется вычислить некоторое результирующее значение.

Примеры



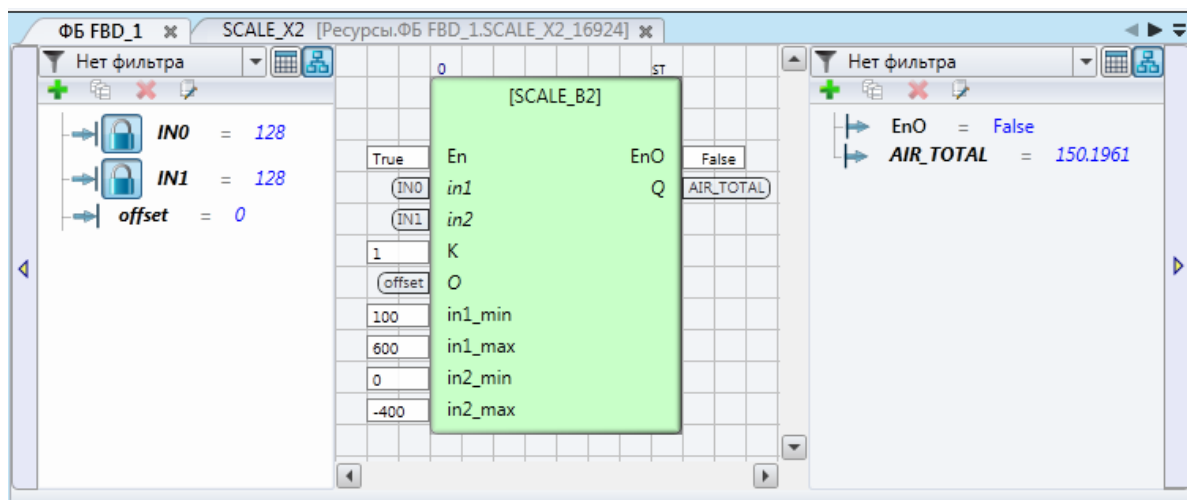
IN0 – воздушный клапан, который регулирует приток воздуха в диапазоне от 100м³/ч до 600м³/ч при изменении IN0 в диапазоне [0, 255].

IN1 – вытяжка, от 0м³/ч до 400м³/ч при изменении IN1 в диапазоне [0, 255].

Для данного случая конфигурация функции должна быть следующей: IN1_MIN = 100, IN1_MAX = 600, IN2_MIN = 0, IN2_MAX = -400.

Результирующий расход воздуха при K = 1 и O = 0 будет варьироваться от -300 (IN0 = 0 и IN1 = 255) до +600 (IN0 = 255 и IN1 = 0).

Результат для входных значений IN0 = 128 (задвижка 50%) и IN1 = 128 (вентилятор 50%) показан на рисунке:



Входное смещение может быть использовано для каскадного подключения функции (см. пример в описании функции SCALE_X2 (OSCAT)).

11.1.7.16. SCALE_B4 (OSCAT)

0		ST	
[SCALE_B4]			
True	En	EnO	False
0	in1	Q	0
0	in2		
0	in3		
0	in4		
0	K		
0	O		
0	in1_min		
1000	in1_max		
0	in2_min		
1000	in2_max		
0	in3_min		
1000	in3_max		
0	in4_min		
1000	in4_max		

Входы функции:

- IN1 (тип данных BYTE) – первый вход;
- IN2 (тип данных BYTE) – второй вход;
- IN3 (тип данных BYTE) – третий вход;
- IN4 (тип данных BYTE) – четвертый вход;
- K (тип данных REAL) – множитель;
- O (тип данных REAL) – смещение;
- IN1_MIN (тип данных REAL) – нижняя граница диапазона, в который масштабируется IN1;
- IN1_MAX (тип данных REAL) – верхняя граница диапазона, в который масштабируется IN1;
- IN2_MIN (тип данных REAL) – нижняя граница диапазона, в который масштабируется IN2;
- IN2_MAX (тип данных REAL) – верхняя граница диапазона, в который масштабируется IN2.
- IN3_MIN (тип данных REAL) – нижняя граница диапазона, в который масштабируется IN3;

- IN3_MAX (тип данных REAL) – верхняя граница диапазона, в который масштабируется IN3.
- IN4_MIN (тип данных REAL) – нижняя граница диапазона, в который масштабируется IN4;
- IN4_MAX (тип данных REAL) – верхняя граница диапазона, в который масштабируется IN4.

SCALE_B4 масштабирует IN1..IN4, суммирует результаты, затем умножает сумму на множитель K и прибавляет смещение O:

$$Q:=(in1*(IN1_MAX-IN1_MIN)/255 + IN1_MIN + in2*(IN2_MAX-IN2_MIN)/255 + IN2_MIN + in3*(IN3_MAX-IN3_MIN)/255 + IN3_MIN + in4*(IN4_MAX-IN4_MIN)/255 + IN4_MIN)*K + O$$

Как и SCALE_B2 (OSCAT), SCALE_B4 может быть использована, например, для вычисления расхода воздуха в вентиляционной системе, а также в любой системе, в которой есть смеситель, и требуется вычислить некоторое результирующее значение.

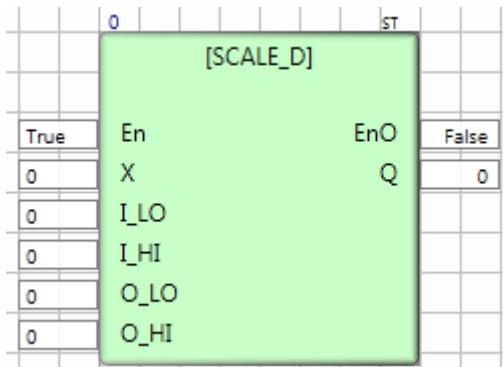
11.1.7.17. SCALE_B8 (OSCAT)

True	En	EnO	False
0	in1	Q	0
0	in2		
0	in3		
0	in4		
0	in5		
0	in6		
0	in7		
0	in8		
0	K		
0	O		
0	in1_min		
1000	in1_max		
0	in2_min		
1000	in2_max		
0	in3_min		
1000	in3_max		
0	in4_min		
1000	in4_max		
0	in5_min		
1000	in5_max		
0	in6_min		
1000	in6_max		
0	in7_min		
1000	in7_max		
0	in8_min		
1000	in8_max		

То же, что и SCALE_B4 (OSCAT), но обрабатывает 8 входов:

$$Q := (in1*(IN1_MAX-IN1_MIN)/255 + IN1_MIN + in2*(IN2_MAX-IN2_MIN)/255 + IN2_MIN + in3*(IN3_MAX-IN3_MIN)/255 + IN3_MIN + in4*(IN4_MAX-IN4_MIN)/255 + IN4_MIN + in5*(IN5_MAX-IN5_MIN)/255 + IN5_MIN + in6*(IN6_MAX-IN6_MIN)/255 + IN6_MIN + in7*(IN7_MAX-IN7_MIN)/255 + IN7_MIN + in8*(IN8_MAX-IN8_MIN)/255 + IN8_MIN)*K + O$$

11.1.7.18. SCALE_D (OSCAT)



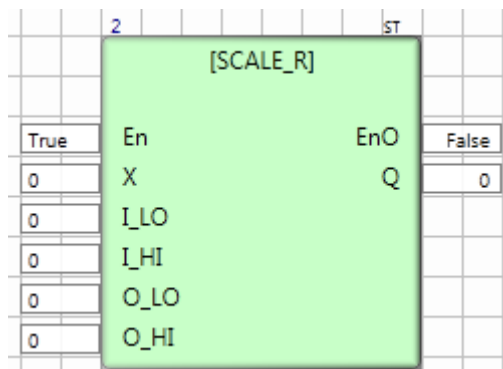
Тип данных X, I_LO и I_HI – DWORD, O_LO, O_HI и Q – REAL.

Ограниченное параметрами I_LO и I_HI входное DWORD-значение X масштабируется (диапазон [I_LO, I_HI] переводится в диапазон [O_LO, O_HI]) и преобразуется в значение REAL. Например, SCALE_D(X, 0, 8191, 0, 100) переводит значение битов 0-12 входа X в диапазон [0, 100].

SCALE_D может возвращать отрицательные значения. Например, SCALE_D (250, 0, 1000, -100, 100)=-50.

Функция работает корректно, если I_LO < I_HI.

11.1.7.19. SCALE_R (OSCAT)



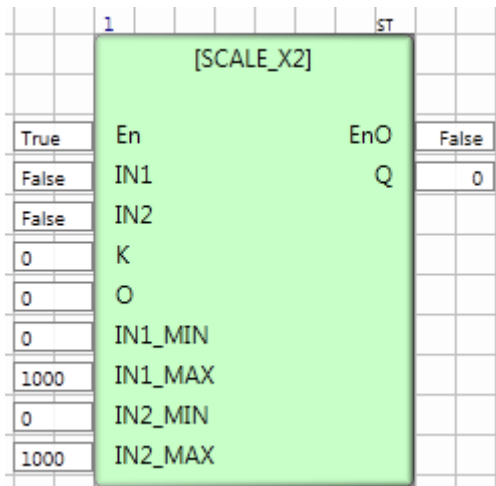
Тип данных входов и выхода – REAL.

Ограниченное параметрами I_LO и I_HI входное REAL-значение X масштабируется (диапазон [I_LO, I_HI] переводится в диапазон [O_LO, O_HI]). Например, SCALE_R(X,4,20,0,100) переводит значения 4..20mA в выход 0..100.

SCALE_R может возвращать отрицательные значения. Например, SCALE_R(100, -100, 200, -300, 0)=-100.

Функция работает корректно, если I_LO < I_HI.

11.1.7.20. SCALE_X2 (OSCAT)



Тип данных IN1 и IN2 – BOOL, остальных входов и выхода – REAL.

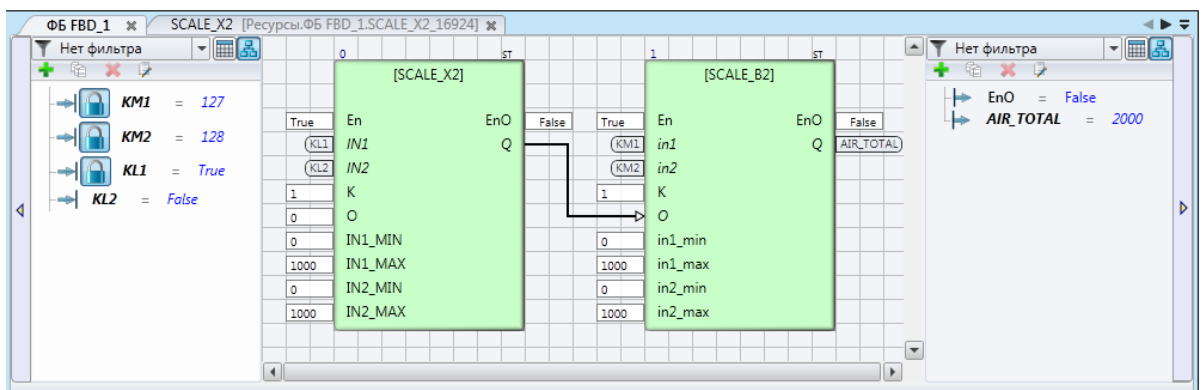
SCALE_X2 использует две функции SEL, выбор из двух, суммирует 2 выбранные значения, умножает сумму на множитель K и прибавляет смещение O.

Первое значение для суммирования выбирается следующим образом: если IN1=FALSE, выбирается IN1_MIN, если IN1=TRUE, выбирается IN1_MAX.

Второе значение для суммирования выбирается аналогично: если IN2=FALSE, выбирается IN2_MIN, если IN2=TRUE, выбирается IN2_MAX.

SCALE_X2 может использоваться, например, для вычисления расхода воздуха в вентиляционных системах, а также в любых системах, в которых используются управляемые заслонки, и требуется вычислять некоторые результирующие величины. С входным смещением, SCALE_X2 могут использоваться каскадно.

В следующем примере показаны две регулируемые заслонки KM1 и KM2 и две заслонки типа ОТКРЫТО/ЗАКРЫТО KL1 и KL2, и вычисляется общий расход воздуха:

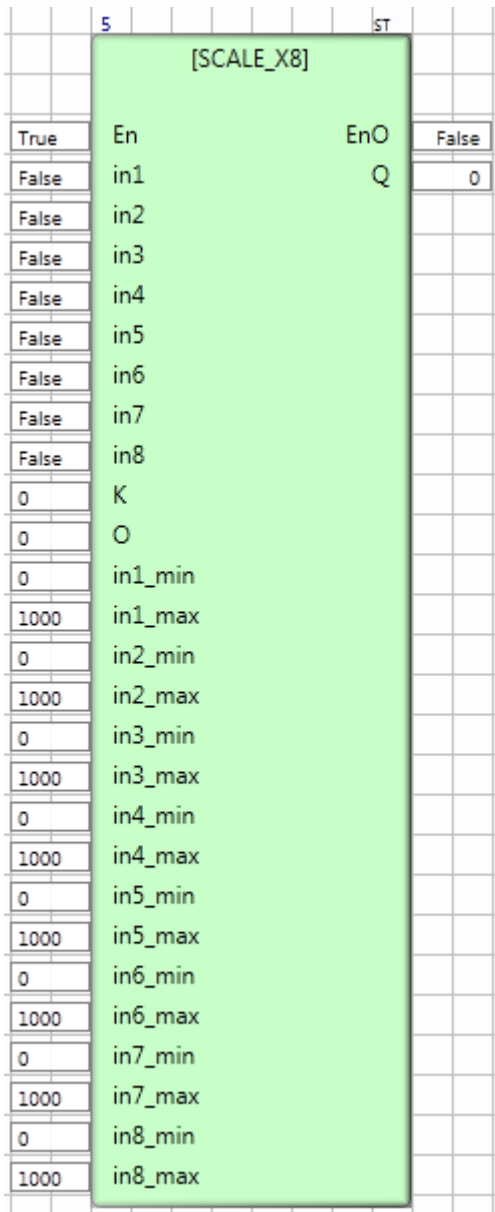


11.1.7.21. SCALE_X4 (OSCAT)

4				ST
[SCALE_X4]				
True	En	EnO	False	
False	IN1	Q	0	
False	IN2			
False	IN3			
False	IN4			
0	K			
0	O			
0	IN1_MIN			
1000	IN1_MAX			
0	IN2_MIN			
1000	IN2_MAX			
0	IN3_MIN			
1000	IN3_MAX			
0	IN4_MIN			
1000	IN4_MAX			

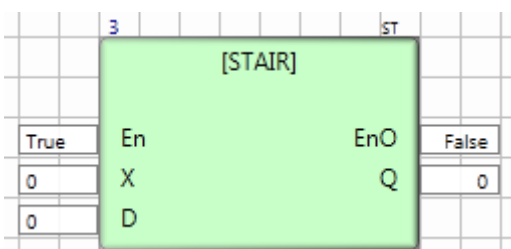
То же, что и SCALE_X2 (OSCAT), но суммирует 4 значения (в SCALE_X4 используется 4 функции SEL).

11.1.7.22. SCALE_X8 (OSCAT)



То же, что и SCALE_X2 (OSCAT), но суммирует 8 значений (в SCALE_X8 используется 8 функций SEL).

11.1.7.23. STAIR (OSCAT)

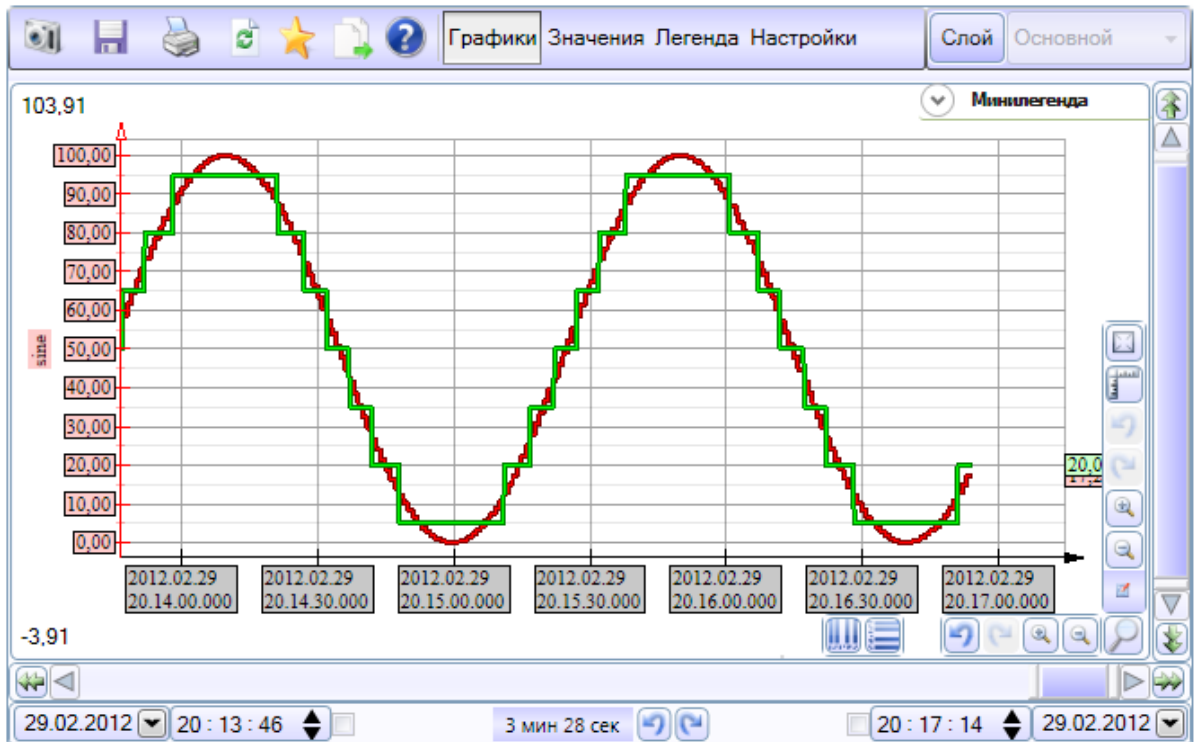


Тип данных входов и выхода функции – REAL.

Если $D \leq 0$, $Q := X$.

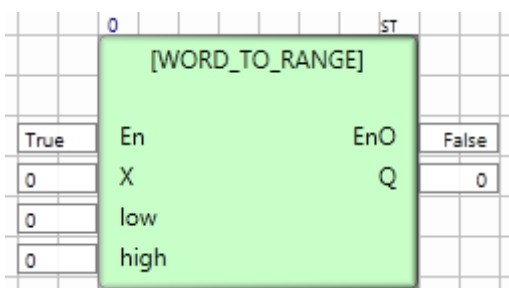
Если $D > 0$, функция преобразует гладкий сигнал, поданный на вход X , в ступенчатый с величиной шага D по оси значений (выход Q принимает только значения, кратные D).

Следующий график иллюстрирует работу функции STAIR на синусоидальном сигнале с амплитудой 1 при $D=0.3$ (коэффициент масштабирования кривых на тренде – 50):



В случае малых флуктуаций X около значений $n \cdot D/2$ (n – любое целое число) выход нестабилен, т.е. переключается между двумя соседними значениями (следствие использования округления в алгоритме функции).

11.1.7.24. WORD_TO_RANGE (OSCAT)

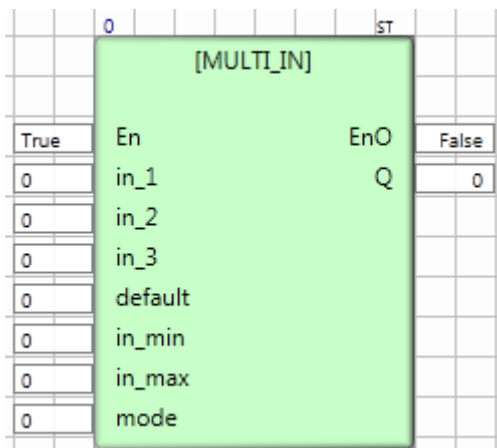


Тип данных X – WORD, LOW, HIGH и Q – REAL.

Функция преобразует WORD-значение входа X в значение REAL с масштабированием в диапазон [LOW, HIGH]. Например, для перевода всего диапазона WORD (от 0 до 65535) в проценты (от 0 до 100) можно использовать $WORD_TO_RANGE(X,0,100)$.

11.1.8. OSCAT.ФУНКЦИИ.ДАТЧИКИ

11.1.8.1. MULTI_IN (OSCAT)



Тип данных входа MODE – BYTE, остальных входов и выхода – REAL.

MULTI_IN принимает до 3 сигналов датчиков (IN_1, IN_2, IN_3), проверяет их и, в зависимости от заданного режима работы (MODE), вычисляет выходное значение (Q).

Значения MODE задают следующие режимы работы:

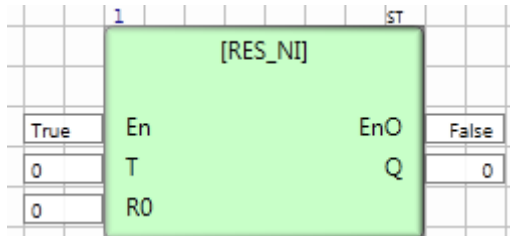
- 0 – MULTI_IN возвращает среднее от входов IN_1.. IN_3;
- 1 – MULTI_IN возвращает IN_1;
- 2 – MULTI_IN возвращает IN_2;
- 3 – MULTI_IN возвращает IN_3;
- 4 – MULTI_IN возвращает DEFAULT;
- 5 – MULTI_IN возвращает наименьшее значение среди входов IN_1..IN_3;
- 6 – MULTI_IN возвращает наибольшее значение среди входов IN_1..IN_3;
- 7 – MULTI_IN возвращает значение входа, расположенное на числовой оси между двумя другими значениями входов;
- >7 – MULTI_IN возвращает 0.

Вне зависимости от режима работы, игнорируются входные значения, большие или равные IN_MAX или меньшие или равные IN_MIN. Например, при вычислении среднего (MODE=0) $Q:=(IN_2+IN_3)/2$, если IN_2 и IN_3 лежат в интервале (IN_MIN, IN_MAX), а IN_1 находится за его пределами.

Если в заданном режиме работы вычисления невозможны, Q:=DEFAULT. Например, Q:=DEFAULT, если при MODE=3 IN_3 находится за пределами интервала (IN_MIN, IN_MAX).

MULTI_IN используется в том случае, когда несколько датчиков измеряют одно и то же значение, и требуются высокая безопасность и надежность. Возможное применение – измерение наружной температуры в различных точках или контроль исправности датчика или кабеля.

11.1.8.2. RES_NI (OSCAT)



Тип данных входов T (температура, 0C) и R0 (сопротивление при 00C) и выхода Q (сопротивление при T) – REAL.

Auszug aus DIN 43760 für Ni100

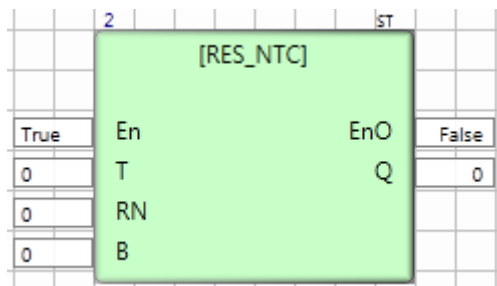
°C	R	°C	R	°C	R	°C	R	°C	R
-60	69,5	-10	94,6	40	123,0	90	154,9	140	190,9
-55	71,9	-5	97,3	45	126,0	95	158,3	145	194,8
-50	74,3	0	100,0	50	129,1	100	161,8	150	198,7
-45	76,7	5	102,8	55	132,2	105	165,3	155	202,6
-40	79,1	10	105,6	60	135,3	110	168,8	160	206,6
-35	81,6	15	108,4	65	138,5	115	172,4	165	210,7
-30	84,2	20	111,2	70	141,7	120	176,0	170	214,8
-25	86,7	25	114,1	75	145,0	125	179,6	175	219,0
-20	89,3	30	117,1	80	148,3	130	183,3	180	223,2
-15	91,9	35	120,0	85	151,6	135	187,1		

Функция вычисляет сопротивление никелевого терморезистивного датчика в диапазоне температур [-60, +180]0C по следующей формуле:

$$RES_NI := R0 + A * T + B * T^2 + C * T^4$$

где A = 0.5485, B = 0.665e-3 и C = 2.805e-9.

11.1.8.3. RES_NTC (OSCAT)



Тип данных входов и выхода – REAL.

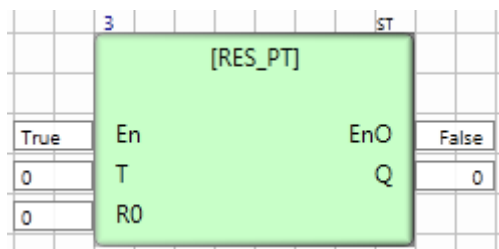
RES_NTC вычисляет сопротивление терморезистивного NTC-датчика в зависимости от значений входов T (температура в 0C) и RN (сопротивление при 250C). Вход B – константа, значение которой должно быть взято из описания датчика.

Вычисления производятся по следующей формуле:

$$R_H = R_N \cdot e^{B \cdot \left(\frac{1}{T+273.15} - \frac{1}{298.15} \right)}$$

Данная формула обеспечивает достаточную точность для низких температур, например, в диапазоне [0, 100]0C. Для более высоких температур рекомендуется использовать формулу Стейнхарта.

11.1.8.4. RES_PT (OSCAT)



Тип данных входов T (температура, 0C) и R0 (сопротивление при 00C) и выхода Q (сопротивление при T) – REAL.

Auszug aus DIN 43760 für Pt100

°C	R	°C	R	°C	R	°C	R	°C	R	°C	R
-200	18,49	-100	60,25	0	100,00	100	138,50	200	175,84	300	212,02
-195	20,65	-95	62,28	5	101,95	105	140,39	205	177,68	305	213,80
-190	22,80	-90	64,30	10	103,90	110	142,29	210	179,51	310	215,57
-185	24,94	-85	66,31	15	105,85	115	144,17	215	181,34	315	217,35
-180	27,08	-80	68,33	20	107,79	120	146,06	220	183,17	320	219,12
-175	29,20	-75	70,33	25	109,73	125	147,94	225	184,99	325	220,88
-170	31,32	-70	72,33	30	111,67	130	149,82	230	186,82	330	222,65
-165	33,43	-65	74,33	35	113,61	135	151,70	235	188,63	335	224,41
-160	35,33	-60	76,33	40	115,54	140	153,58	240	190,45	340	226,17
-155	37,63	-55	78,32	45	117,47	145	155,45	245	192,26	345	227,92
-150	39,71	-50	80,31	50	119,40	150	157,31	250	194,07	350	229,67
-145	41,79	-45	82,29	55	121,32	155	159,18	255	195,88	355	231,42
-140	43,87	-40	84,27	60	123,24	160	161,04	260	197,69	360	233,17
-135	45,94	-35	86,25	65	125,16	165	162,90	265	199,49	365	234,91
-130	48,00	-30	88,22	70	127,07	170	164,76	270	201,29	370	236,65
-125	50,06	-25	90,19	75	128,98	175	166,61	275	203,08	375	238,39
-120	52,11	-20	92,16	80	130,89	180	168,46	280	204,88	380	240,13
-115	54,15	-15	94,12	85	132,80	185	170,31	285	206,67	385	241,86
-110	56,19	-10	96,09	90	134,70	190	172,16	290	208,45	390	243,59
-105	58,22	-5	98,04	95	136,60	195	174,00	295	210,24	395	245,31

Функция вычисляет сопротивление платинового терморезистивного датчика в диапазоне температур [-200, +850]0C.

При $T > 00C$ используется формула

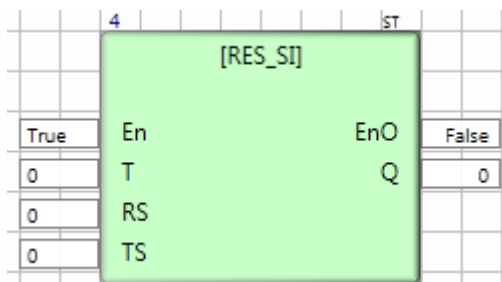
$$RES_PT = R0*(1 + A*T + B*T^2)$$

При $T < 00C$ используется формула

$$RES_PT = R0*(1 + A*T + B*T^2) + C*(T - 100)*T^3$$

где $A = 3.90802e-3$, $B = -5.80195e-7$, $C = -427350e-12$.

11.1.8.5. RES_SI (OSCAT)



Тип данных входов и выхода – REAL.

RES_SI вычисляет сопротивление силиконового терморезистивного датчика в диапазоне температур $[-50, +150]0C$ в зависимости от значений входов T (температура в $0C$) и RS (сопротивление при температуре TS $0C$).

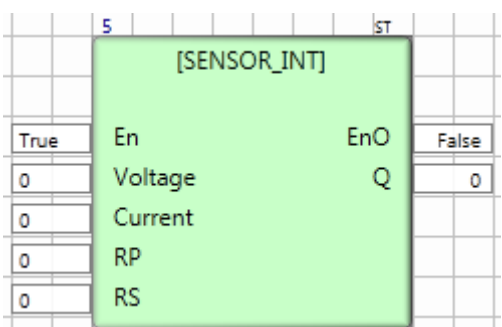
В отличие от функций RES_NI (OSCAT), RES_NTC (OSCAT) и RES_PT (OSCAT), для которых реперное значение сопротивления задается при фиксированной температуре, сопротивление RS может быть задано при произвольной температуре TS (например, при $250C$ для KTY-10).

Для вычислений используется следующая формула:

$$RES_SI = RS + A*(T - TS) + B*(T - TS)^2$$

где $A = 7.64e-3$, $B = 1.66e-5$

11.1.8.6. SENSOR_INT (OSCAT)



Входы функции:

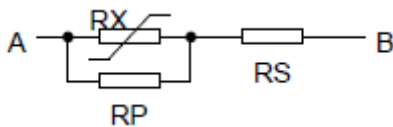
- VOLTAGE (тип данных REAL) – напряжение, В;
- CURRENT (тип данных REAL) – ток, А;
- RP (тип данных REAL) – паразитное параллельное сопротивление, Ом;
- RS (тип данных REAL) – паразитное последовательное сопротивление, Ом.

Выход функции:

- Q (тип данных REAL) – сопротивление датчика.

SENSOR_INT вычисляет сопротивление датчика с учетом наличия паразитных сопротивлений, которые часто сопутствуют измерениям.

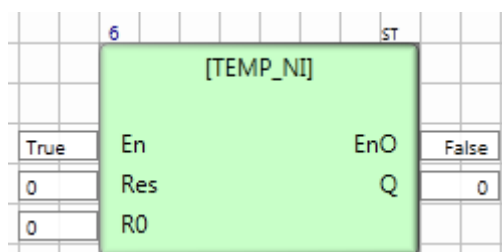
Например, АЦП измеряет или ток при фиксированном напряжении или напряжение при фиксированном токе. Сопротивление, вычисленное по измеренным напряжению и току, – это не только сопротивление датчика, оно включает также два паразитных сопротивления RS и RP:



Т.к. паразитные сопротивления остаются постоянными, они могут быть скомпенсированы, и может быть вычислено реальное сопротивление датчика.

Далее, при использовании соответствующей функции (TEMP_NI (OSCAT), TEMP_NTC (OSCAT), TEMP_PT (OSCAT) или TEMP_SI (OSCAT)), может быть вычислено точное значение температуры.

11.1.8.7. TEMP_NI (OSCAT)



Входы функции:

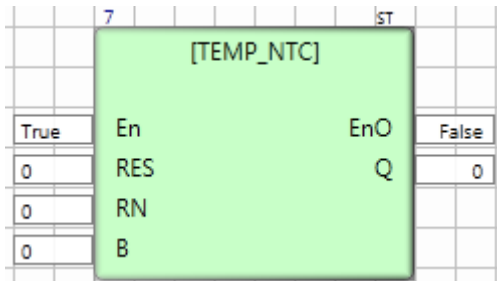
- RES (тип данных REAL) – измеренное сопротивление Ni-термистора, Ом;
- R0 (тип данных REAL) – сопротивление Ni-термистора при 00С.

Выход функции:

- Q (тип данных REAL) – температура в 0С, соответствующая RES.

Данная функция обратна функции RES_NI (OSCAT).

11.1.8.8. TEMP_NTC (OSCAT)



Входы функции:

- RES (тип данных REAL) – измеренное сопротивление NTC-термистора, Ом;
- RN (тип данных REAL) – сопротивление NTC-термистора при 250С;
- B (тип данных REAL) – константа, значение которой должно быть взято из документации NTC-датчика.

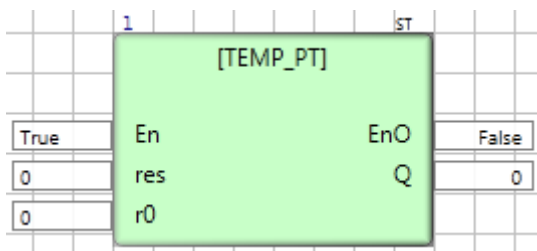
Выход функции:

- Q (тип данных REAL) – температура в 0С, соответствующая RES.

Данная функция обратна функции RES_NTC (OSCAT):

$$Q := B * 298.15 / (B + \ln(RES/RN) * 298.15) - 273.15$$

11.1.8.9. TEMP_PT (OSCAT)



Входы функции:

- RES (тип данных REAL) – измеренное сопротивление Pt-термистора, Ом;
- R0 (тип данных REAL) – сопротивление Pt-термистора при 00С.

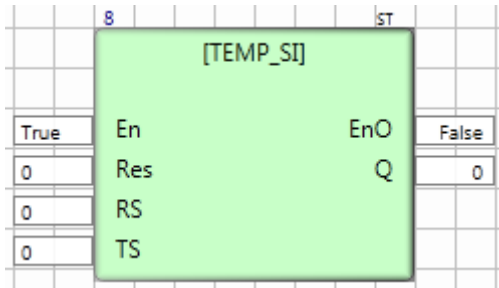
Выход функции:

- Q (тип данных REAL) – температура в 0С, соответствующая RES.

Данная функция обратна функции RES_PT (OSCAT).

Важно! TEMP_PT переписана без использования указателей.

11.1.8.10. TEMP_SI (OSCAT)



Входы функции:

- RES (тип данных REAL) – измеренное сопротивление силиконового термистора, Ом;
- RS (тип данных REAL) – сопротивление силиконового термистора при TS 0С;
- TS (тип данных REAL) – температура в 0С для RS.

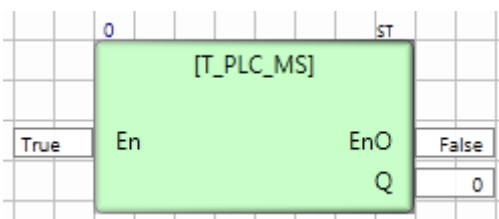
Выход функции:

- Q (тип данных REAL) – температура в 0С, соответствующая RES.

Данная функция обратна функции RES_SI (OSCAT).

11.1.9. OSCAT.ФУНКЦИИ.МОДУЛИ ИЗМЕРЕНИЯ

11.1.9.1. T_PLC_MS (OSCAT)



Тип данных выхода Q – DWORD (значение SPS-таймера PLC в мс).

Код функции:

```
FUNCTION T_PLC_MS : DWORD
```

```
VAR CONSTANT
```

```
DEBUG : BOOL := FALSE;
```

```

N : INT := 0;

OFFSET := 0;

END_VAR

VAR

TEMP : DWORD := 1;

END_VAR

T_PLC_MS := TIME_TO_DWORD(TIME());

IF debug THEN

T_PLC_MS := (SHL(T_PLC_MS,N) OR SHL(DWORD#1,N)-1) + OFFSET;

END_IF;

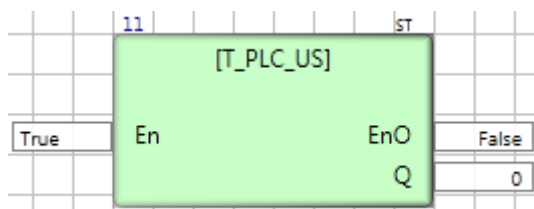
```

Функция работает в двух режимах – нормальном и отладочном.

В нормальном режиме (DEBUG=FALSE, значение по умолчанию) T_PLC_MS использует функцию *time()* (см. TIME, таймер ПЛК) и возвращает текущее время стандартного таймера контроллера как целое число миллисекунд. При использовании T_PLC_MS длина временного интервала вычисляется корректно, в том числе при переполнении таймера.

В отладочном режиме (DEBUG=TRUE) T_PLC_MS используется для тестирования поведения ПО при переполнении, не дожидаясь 49 суток. С помощью константы N скорость нарастания выходного значения функции может быть увеличена в 2,4,8,16 и т.д. раз, константа OFFSET добавляется к полученному значению. Константы DEBUG, N и OFFSET специально не выполнены в виде входов функции, чтобы предотвратить случайное некорректное использование.

11.1.9.2. T_PLC_US (OSCAT)



Тип данных выхода Q – DWORD (значение SPS-таймера PLC в мкс).

Код функции:

```

FUNCTION T_PLC_US : DWORD

VAR CONSTANT

```

```

debug:BOOL := False; N:INT := 0; offset:DWORD := 0;

END_VAR

VAR tx:TIME := T#0ms; END_VAR

tx := TIME();

T_PLC_US := TIME_TO_DWORD(Tx)*1000;

IF debug THEN

T_PLC_US := (SHL(T_PLC_US,N) OR SHL(DWORD#1,N)-1) + OFFSET;

END_IF;

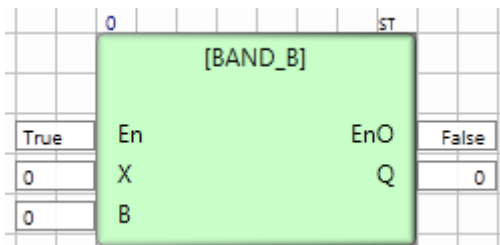
END_FUNCTION

```

Данная функция только предоставляет интерфейс для функции чтения таймера с разрешением меньше 1мс. В текущей редакции T_PLC_US является аналогом функции T_PLC_MS (OSCAT), она возвращает текущее время таймера ПЛК в микросекундах с точностью 1 миллисекунда.

11.1.10. OSCAT.ФУНКЦИИ.МОДУЛИ РЕГУЛИРОВАНИЯ

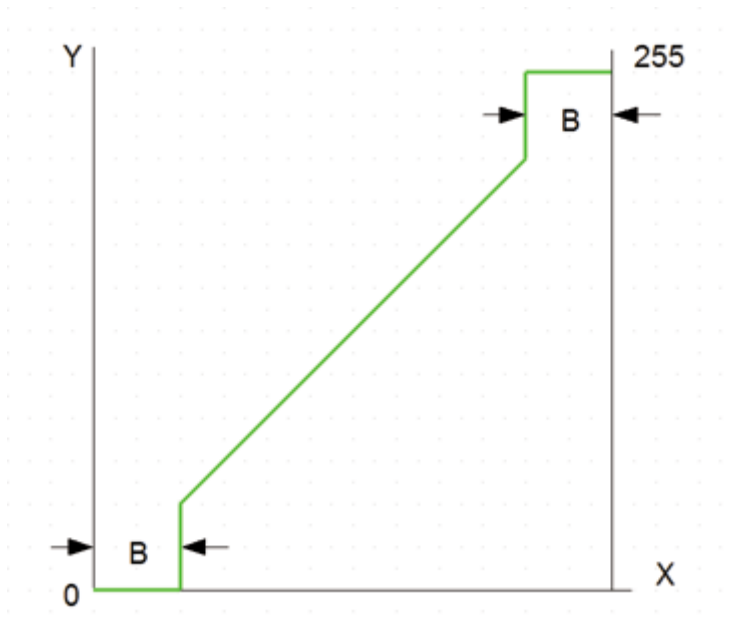
11.1.10.1. BAND_B (OSCAT)



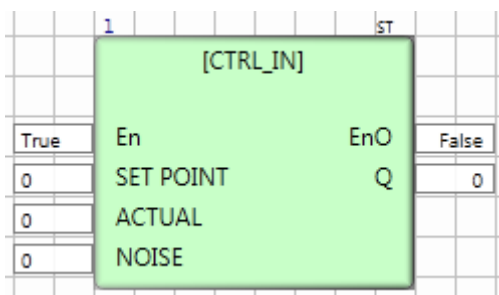
Тип данных входов и выхода – BYTE.

Выход Q равен значению входа X везде, кроме следующих областей:

- [0, B) – в этой области значений X Q=0;
- (255-B, 255] – в этой области значений X Q=255.



11.1.10.2. CTRL_IN (OSCAT)



Входы функции:

- SET POINT (тип данных REAL) – уставка процесса;
- ACTUAL (тип данных REAL) – значение процесса;
- NOISE (тип данных REAL) – порог.

Выход функции:

- Q (тип данных REAL) – девиация процесса.

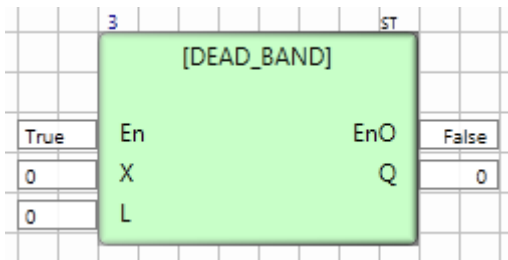
CTRL_IN вычисляет отклонение процесса от уставки (SET_POINT - ACTUAL). Если разница меньше NOISE, выход остается равным 0.

Код CTRL_IN:

```
CTRL_IN:=DEAD_BAND(X:=SET_POINT-ACTUAL, L:=NOISE);
```

Важно! В CTRL_IN функция DEAD_ZONE (OSCAT) заменена на DEAD_BAND (OSCAT).

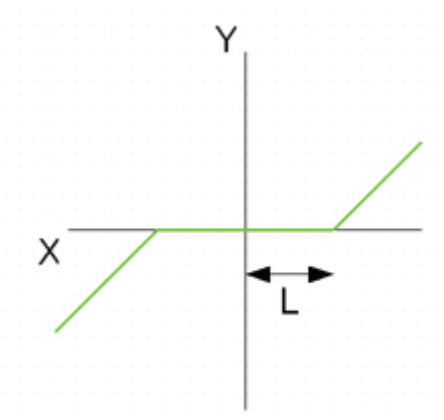
11.1.10.3. DEAD_BAND (OSCAT)



Тип данных входов и выхода – REAL.

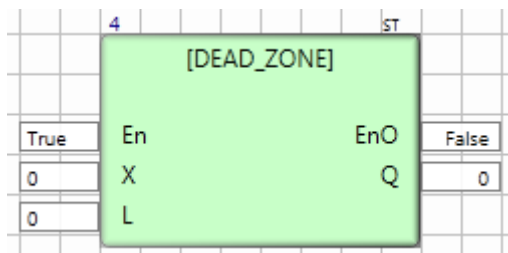
DEAD_BAND – безударная линейная передаточная функция с зоной нечувствительности шириной $2L$ ($L > 0$):

- если $X < -L$, $Q := X + L$;
- если $X > L$, $Q := X - L$;
- если $-L \leq X \leq L$, $Q := 0$.



DEAD_BAND используется для фильтрации шумов квантования в сигнале – например, для предотвращения постоянного переключения контроллера при малых изменениях сигнала, что приводит к перегрузкам и преждевременному старению исполнительного механизма.

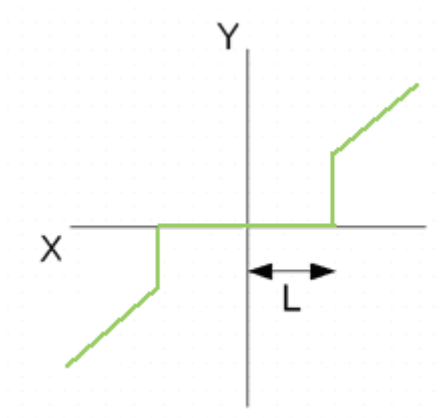
11.1.10.4. DEAD_ZONE (OSCAT)



Тип данных входов и выхода – REAL.

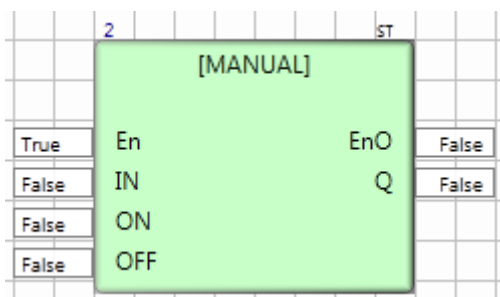
DEAD_ZONE – ударная линейная передаточная функция с зоной нечувствительности шириной $2L$ ($L > 0$):

- если $X < -L$ или $X > L$, $Q := X$;
- если $-L \leq X \leq L$, $Q = 0$.



11.1.11. OSCAT.ФУНКЦИИ.УПРАВЛЕНИЕ УСТРОЙСТВАМИ

11.1.11.1. MANUAL (OSCAT)

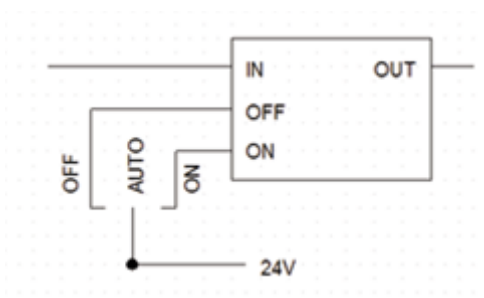


Тип данных входов и выхода функции – BOOL.

MANUAL может заменить значение цифрового входа IN на TRUE или FALSE:

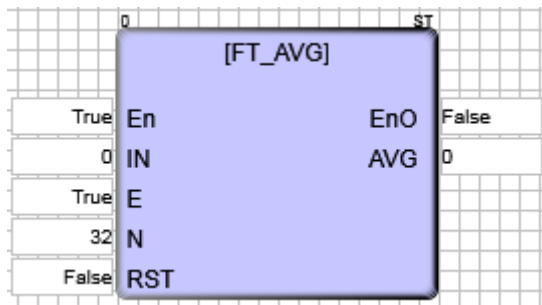
- если $ON = FALSE$ и $OFF = FALSE$, $Q := IN$;
- если $OFF = TRUE$, $Q := FALSE$;
- если $OFF = FALSE$ и $ON = TRUE$, $Q := TRUE$.

Типичное использование функции MANUAL – трехпозиционный переключатель OFF/AUTO/ON, где AUTO соответствует $ON = FALSE$ и $OFF = FALSE$:



11.1.12. OSCAT.ФБ.АРИФМЕТИЧЕСКИЕ ФУНКЦИИ

11.1.12.1. FT_AVG (OSCAT)



Входы ФБ:

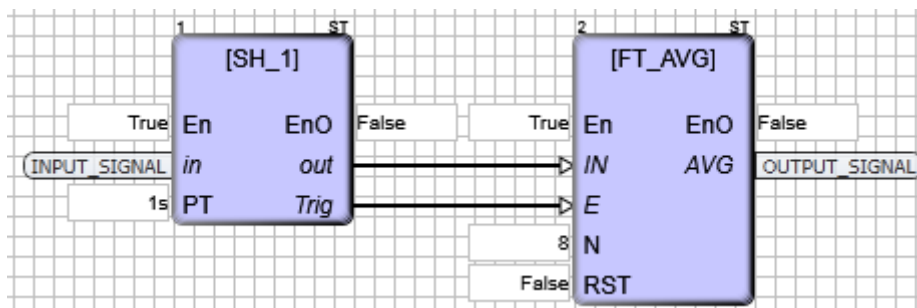
- IN (REAL) – входной сигнал;
- E (BOOL) – разрешение (TRUE) или запрет (FALSE) вычислений;
- N (UINT) – число значений в буфере;
- RST (BOOL) – сброс.

Выходы ФБ:

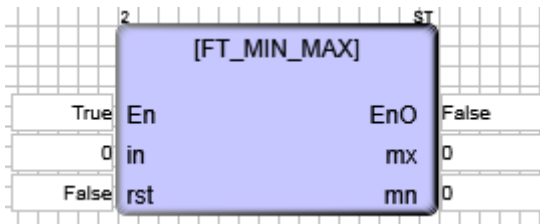
- AVG (REAL) – выходной сигнал.

FT_AVG – это аналог ФБ FILTER_MAV_DW (OSCAT) вычисления скользящего среднего для данных REAL с дополнительным входом E. Значение IN помещается в буфер только при E=TRUE, что позволяет делать выборки из входного сигнала в нужные моменты времени.

FT_AVG может использоваться совместно с SH_1. В примере ниже ФБ SH_1 делает выборки из входного сигнала с периодом 1с, а ФБ FT_AVG вычисляет скользящее среднее по восьми выборкам:



11.1.12.2. FT_MIN_MAX (OSCAT)



Входы ФБ:

- IN (REAL) – входной сигнал;
- RST (BOOL) – сброс.

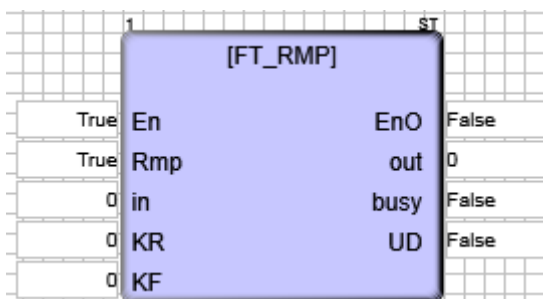
Выходы ФБ:

- MX (REAL) – максимальное значение входного сигнала;
- MN (REAL) – минимальное значение входного сигнала.

ФБ вычисляет максимальное и минимальные значения входного сигнала с момента RST=FALSE.

При RST=TRUE MX=MN=IN.

11.1.12.3. FT_RMP (OSCAT)



Входы ФБ:

- RMP (BOOL) – разрешение (TRUE) или запрет (FALSE) сглаживания;
- IN (REAL) – входной сигнал;
- KR (REAL) – скорость нарастания при сглаживании, [с-1], положительное число;
- KF (REAL) – скорость убывания при сглаживании, [с-1], положительное число.

Выходы ФБ:

- OUT (REAL) – выходной сигнал;
- BUSY (BOOL) – индикатор процесса сглаживания;

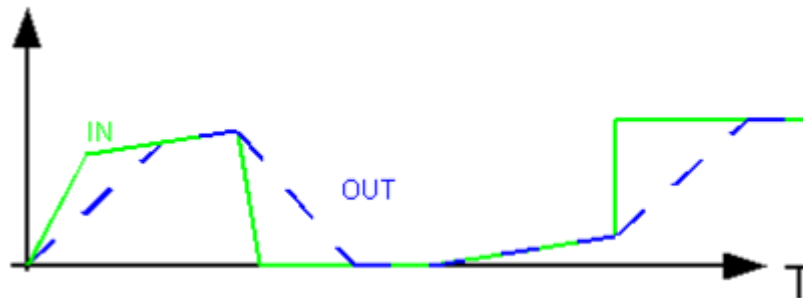
- UD (BOOL) – индикатор нарастания/убывания при сглаживании.

При RMP=FALSE OUT:=IN.

При RMP=TRUE ФБ линейно сглаживает входной сигнал IN. В процессе сглаживания BUSY=TRUE, а когда OUT достигает IN, BUSY:=FALSE. Выход UD равен TRUE при нарастании выходного сигнала и FALSE – при убывании.

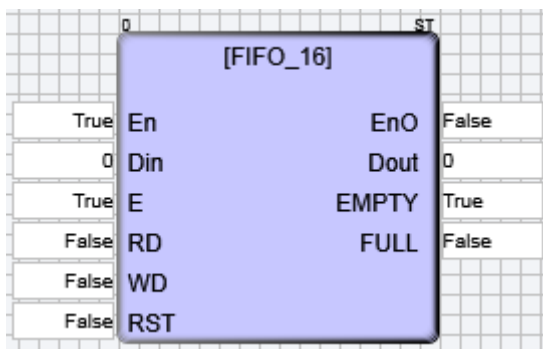
Строго говоря, сглаживание работает в том случае, если $T_{цикла} < |IN_n - OUT_{n-1}| / K$ (K равно KR при нарастании и KF при убывании), в противном случае OUT:=IN. Для упрощения понимания можно использовать более мягкое условие начала сглаживания: скорость возрастания (убывания) IN должна быть больше KR (KF), где под скоростью возрастания/убывания IN понимается отношение изменения IN за 1 цикл к длительности цикла.

Пример работы ФБ:



11.1.13. OSCAT.ФБ.МОДУЛИ ПАМЯТИ

11.1.13.1. FIFO_16 (OSCAT)



Входы ФБ:

- DIN (DWORD) – вход;
- E (BOOL) – разрешение выполнения команд RD и WD;
- RD (BOOL) – RD=TRUE – команда извлечения из стека;

- WD (BOOL) – WD=TRUE – команда помещения в стек;
- RST (BOOL) – сброс.

Выходы ФБ:

- DOUT (DWORD) – значение, извлеченное из стека;
- EMPTY (BOOL) – EMPTY=TRUE означает, что стек пуст;
- FULL (BOOL) – FULL=TRUE означает, что стек заполнен полностью.

FIFO_16 – это стек FIFO , содержащий 16 позиций для данных DWORD.

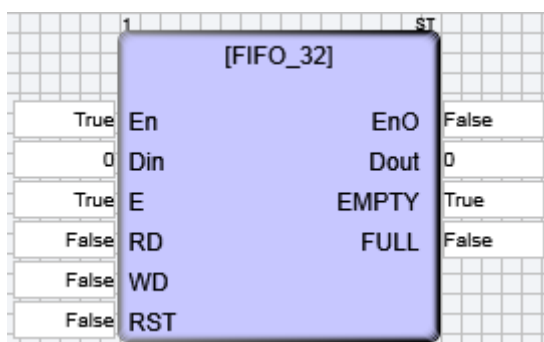
Если данные помещены во все свободные позиции стека, FULL=TRUE, и помещение новых значений в стек невозможно. EMPTY=TRUE индицирует, что стек пуст (все позиции свободны, и невозможно извлечение значений из стека).

Значение DIN помещается в стек в каждом цикле, в котором WD=E=TRUE. Значение из стека извлекается в каждом цикле, в котором RD=E=TRUE.

При RST=TRUE освобождаются все позиции стека (EMPTY=TRUE). Позиция также освобождается при извлечении ее значения (после извлечения значений из всех доступных позиций EMPTY=TRUE). Например, если поместить в пустой стек n значений ($n \leq 16$), а затем извлечь k значений ($k \leq n$), то для дальнейшего извлечения доступны оставшиеся (n-k) позиций, а свободны (16-n+k) позиций. Учитывая данное обстоятельство, при необходимости можно конфигурировать одновременное помещение и извлечение (выполняется в одном и том же цикле).

Используемые функции: INC1 (OSCAT).

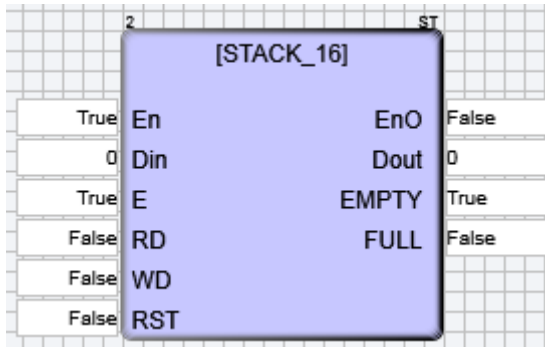
11.1.13.2. FIFO_32 (OSCAT)



FIFO_32 – это аналог FIFO_16 (OSCAT), но стек содержит 32 позиции.

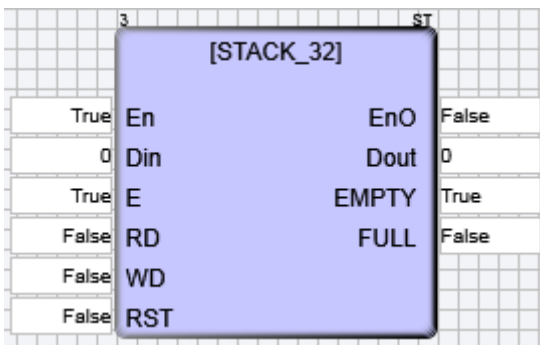
Используемые функции: INC1 (OSCAT).

11.1.13.3. STACK_16 (OSCAT)



STACK_16 – это стек LIFO, содержащий 16 позиций для данных DWORD. За исключением порядка извлечения, данный ФБ работает так же, как FIFO_16 (OSCAT).

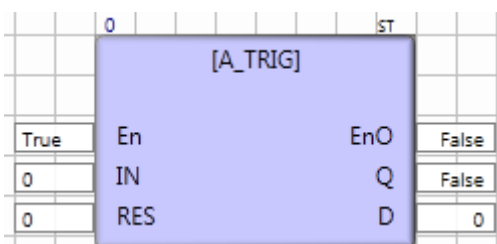
11.1.13.4. STACK_32 (OSCAT)



STACK_32 – это аналог STACK_16 (OSCAT), но стек содержит 32 позиции.

11.1.14. OSCAT.ФБ.ГЕНЕРАТОРЫ ИМПУЛЬСОВ

11.1.14.1. A_TRIG (OSCAT)



Входы ФБ:

- IN (тип данных REAL) – входной сигнал;
- RES (тип данных REAL) – порог.

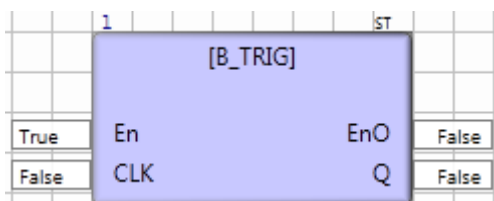
Выходы ФБ:

- Q (тип данных BOOL) – выход, на котором вырабатывается импульс;

- D (тип данных REAL) – последнее изменение входного сигнала или 0 в случае выработки импульса.

Помимо входов и выходов, в A_TRIG существует внутренняя переменная LAST_IN. На нулевом цикле (при старте исполнения программы) LAST_IN=0. В последующих циклах LAST_IN остается равной 0 до тех пор, пока абсолютная величина выхода D (в D записывается разность текущего значения входа IN и LAST_IN) не превысит RES. В случае превышения в LAST_IN записывается текущее значение IN (и, таким образом, D обнуляется), а на выходе Q генерируется одиночный импульс длиной 1 цикл. Далее алгоритм ФБ повторяется с новым значением LAST_IN.

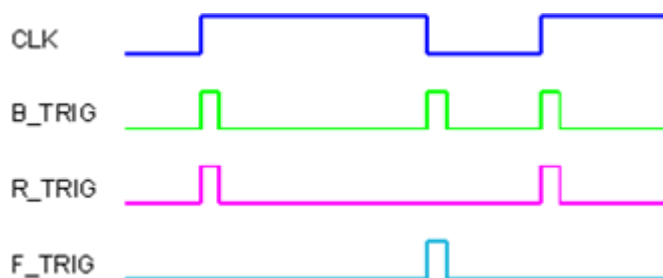
11.1.14.2. B_TRIG (OSCAT)



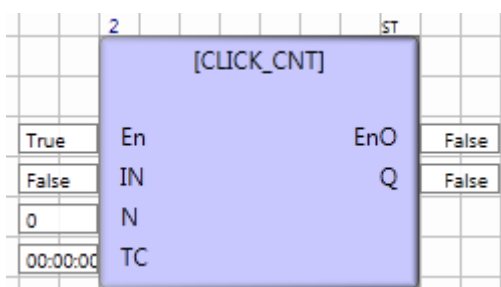
Тип данных входа и выхода – BOOL.

На выходе Q генерируется одиночный импульс длиной 1 цикл по любому фронту входного цифрового сигнала CLK.

На рисунке ниже показаны отличия B_TRIG от функций R_TRIG, импульс по переднему фронту и F_TRIG, импульс по заднему фронту, которые генерируют импульс соответственно только по переднему или только по заднему фронту:



11.1.14.3. CLICK_CNT (OSCAT)



Входы ФБ:

- IN (тип данных BOOL) – входной сигнал;
- N (тип данных INT) – число импульсов;
- TC (тип данных TIME) – время, в течение которого должны пройти N импульсов.

Выход ФБ:

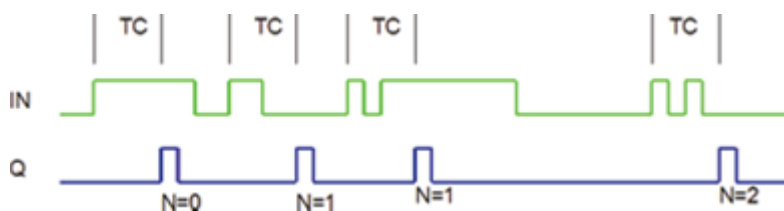
- Q (тип данных BOOL) – выход, на котором вырабатывается импульс.

ФБ содержит внутренний таймер, который запускается по первому переднему фронту в сигнале IN и останавливается по истечении времени TC, затем по первому переднему фронту в IN вновь запускается на время TC и т.д.

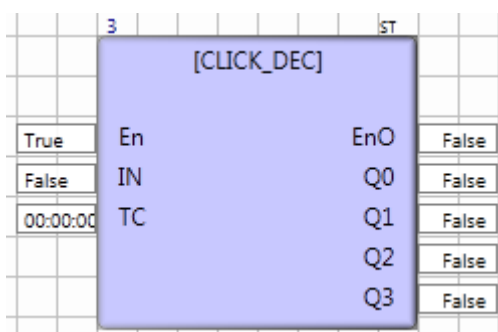
При $N > 0$ CLICK_CNT генерирует на выходе Q одиночный импульс длиной 1 цикл, если во входном сигнале в течение одного периода работы таймера прошли N полных импульсов, т.е. $M \geq N$ передних и ровно N задних фронтов.

При $N = 0$ CLICK_CNT генерирует на выходе Q одиночный импульс длиной 1 цикл, если в течение одного периода работы таймера во входном сигнале прошел только один передний фронт (который, собственно, и запустил таймер).

Следующая диаграмма иллюстрирует алгоритм ФБ:



11.1.14.4. CLICK_DEC (OSCAT)



Входы ФБ:

- IN (тип данных BOOL) – входной сигнал;
- TC (тип данных TIME) – время, в течение которого должны пройти импульсы;

Выходы ФБ:

- Q0 (тип данных BOOL) – выходной сигнал в случае переднего фронта в IN в течение ТС;
- Q1 (тип данных BOOL) – выходной сигнал в случае 1 импульса в IN в течение ТС;
- Q2 (тип данных BOOL) – выходной сигнал в случае 2 импульсов в IN в течение ТС;
- Q3 (тип данных BOOL) – выходной сигнал в случае 3 импульсов в IN в течение ТС.

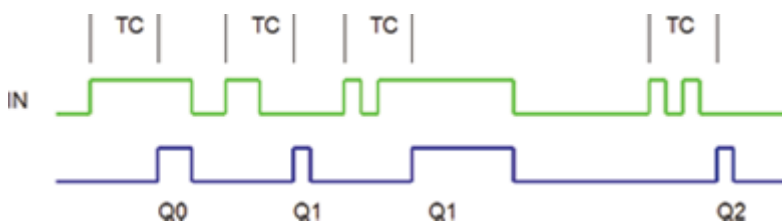
ФБ содержит внутренний таймер, который запускается по первому переднему фронту в сигнале IN и останавливается по истечении времени ТС, затем по первому переднему фронту в IN вновь запускается на время ТС и т.д.

Если в течение одного периода работы таймера во входном сигнале IN прошел только один передний фронт (который, собственно, и запустил таймер), Q0 принимает значение 1 и остается равным 1 до тех пор, пока IN не примет значение FALSE.

Если в течение одного периода работы таймера во входном сигнале IN прошло ровно $n=1,2,3$ полных импульсов (ровно n передних и ровно n задних фронтов), на соответствующем выходе $Q<n>$ генерируется одиночный импульс длиной 1 цикл.

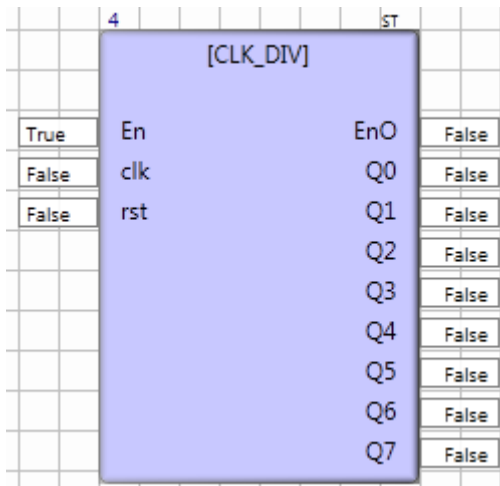
Если в течение одного периода работы таймера во входном сигнале IN прошло $(n+1)$ передних и n задних фронтов (т.е. $n=1,2,3$ полных импульсов и передний фронт), выход $Q<n>$ принимает значение 1 и остается равным 1 до тех пор, пока IN не примет значение FALSE.

Следующая диаграмма иллюстрирует алгоритм ФБ:



CLICK_DEC может использоваться для детектирования повторяющегося нажатия клавиш.

11.1.14.5. CLK_DIV (OSCAT)



Входы ФБ:

- CLK (тип данных BOOL) – на этот вход подается сигнал тактового генератора с частотой следования импульсов F (ширина импульса в этом сигнале не может быть больше, чем 1 цикл);
- RST (тип данных BOOL) – при RST=1 выходы ФБ обнуляются.

Выходы ФБ:

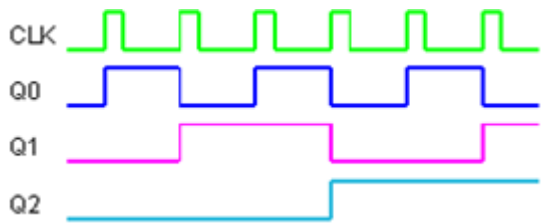
- Q0 (тип данных BOOL) – меандр с частотой F/2;

Важно! Меандр – прямоугольный сигнал со скважностью 2. Скважность – отношение периода следования к ширине импульса. При скважности, равной 2, ширина импульса равна паузе между импульсами и равна половине периода следования.

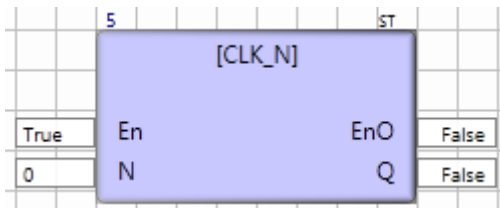
- Q1 (тип данных BOOL) – меандр с частотой F/4;
- Q2 (тип данных BOOL) – меандр с частотой F/8;
- Q3 (тип данных BOOL) – меандр с частотой F/16;
- Q4 (тип данных BOOL) – меандр с частотой F/32;
- Q5 (тип данных BOOL) – меандр с частотой F/64;
- Q6 (тип данных BOOL) – меандр с частотой F/128;
- Q7 (тип данных BOOL) – меандр с частотой F/256.

ФБ используется как делитель частоты.

Временная диаграмма ФБ показана на рисунке:



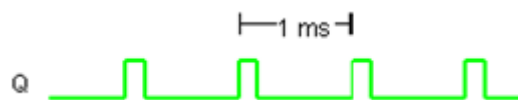
11.1.14.6. CLK_N (OSCAT)



Тип данных входа N – INT, выхода Q – BOOL.

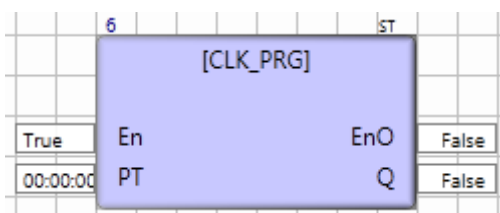
С помощью SPS-таймера ПЛК (разрешение этого таймера – 1 мс) ФБ генерирует импульс шириной ровно 1 цикл каждые 2N миллисекунд (т.е. при N=0 период следования импульсов равен 1мс, при N=1 – 2мс, при N=2 – 4мс и т.д.).

Следующая диаграмма иллюстрирует работу ФБ при N=0:



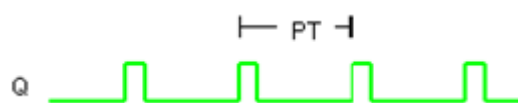
Используемые функции: T_PLC_MS (OSCAT).

11.1.14.7. CLK_PRG (OSCAT)



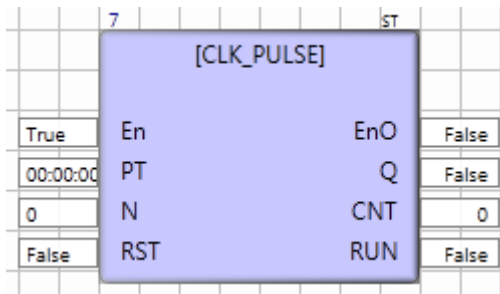
Тип данных входа PT – TIME, выхода Q – BOOL.

ФБ генерирует на выходе Q импульс шириной 1 цикл с периодом PT (при старте программы импульс генерируется в первом же цикле):



Используемые функции: T_PLC_MS (OSCAT).

11.1.14.8. CLK_PULSE (OSCAT)



Входы ФБ:

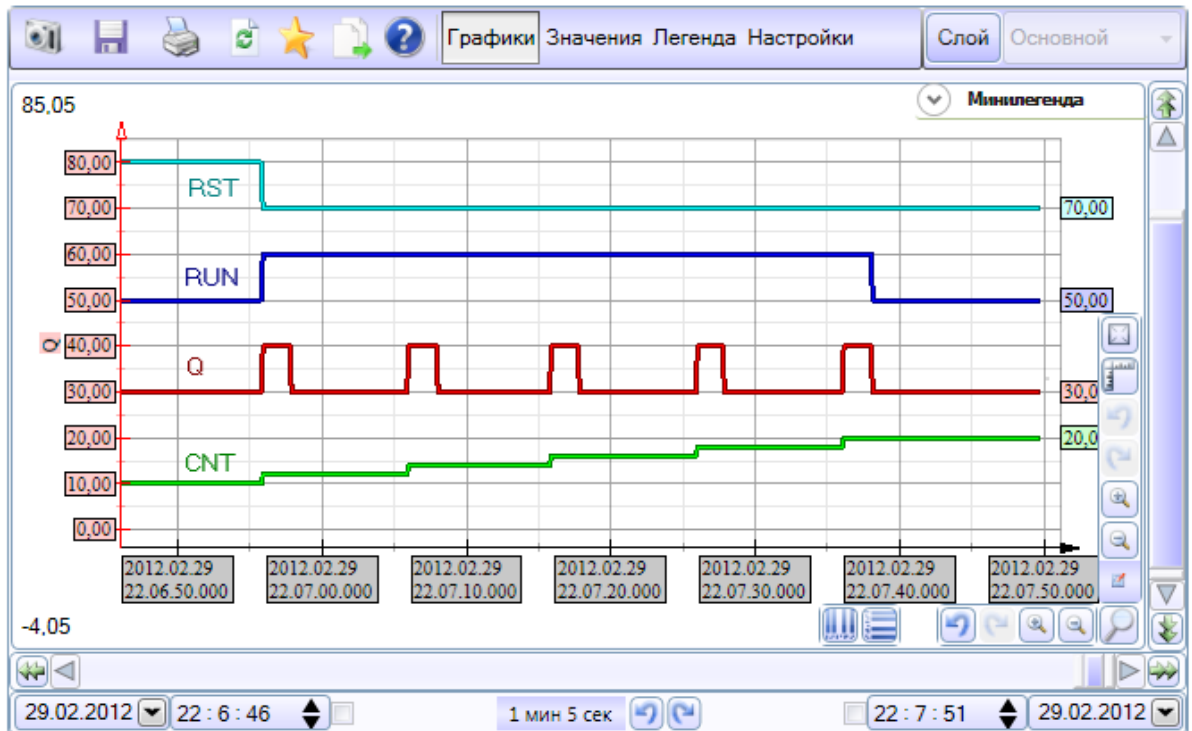
- PT (тип данных TIME) – период следования генерируемых импульсов;
- N (тип данных INT) – число генерируемых импульсов (1-32767). Если N=0, генерируется бесконечная последовательность импульсов;
- RST (тип данных BOOL) – сброс.

Выходы ФБ:

- Q (тип данных BOOL) – генерируемые импульсы;
- CNT (тип данных INT) – текущее число сгенерированных импульсов;
- RUN (тип данных BOOL) – индикатор генерации импульсов (в течение генерации импульсов RUN=TRUE).

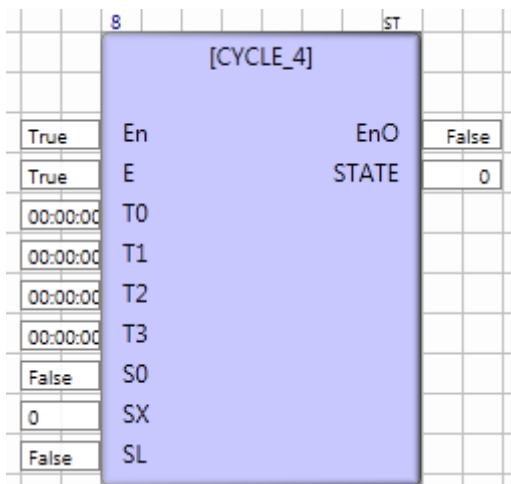
CLK_PULSE генерирует на выходе Q заданное число импульсов шириной 1 цикл с заданным периодом следования PT. Генерация начинается при RST=FALSE (первый импульс генерируется в том же цикле, в котором RST принимает значение FALSE). Для отмены генерации нужно присвоить RST значение TRUE (по этой команде обнуляются Q, CNT и RUN до следующего сброса RST).

Следующая диаграмма иллюстрирует алгоритм ФБ при PT=10с и N=5 при периоде выполнения задачи (цикле) 2с:



Используемые функции: T_PLC_MS (OSCAT).

11.1.14.9. CYCLE_4 (OSCAT)



Входы ФБ:

- E (тип данных BOOL) – разрешение (TRUE) или запрет (FALSE) работы алгоритма ФБ;
- T0 .. T3 (тип данных TIME) – времена пребывания ФБ в состояниях 0 .. 3 соответственно;
- S0 (тип данных BOOL) – разрешение (TRUE) или запрет (FALSE) циклического прохождения состояний;

- SX (тип данных INT) – состояние, в которое переходит блок, если SL=TRUE;
- SL (тип данных BOOL) – разрешение (TRUE) или запрет (FALSE) отработки SX.

Выход ФБ:

- STATE (тип данных INT) – текущее состояние ФБ.

При SL=FALSE ФБ предназначен для последовательного прохождения состояний 0-3, при этом T0 .. T3 определяют длительность состояний.

Если SL=TRUE, блок переходит в состояние SX (ручной режим).

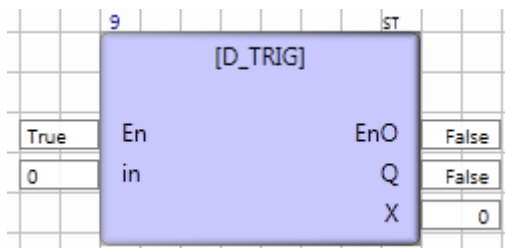
Если SL=FALSE и SO=FALSE, ФБ проходит все состояния от начального до 3, и на этом переходы прекращаются.

Если SL=FALSE и SO=TRUE, включается режим циклического прохождения состояний (начальное, .. 3, 0, .. 3, 0 и т.д.).

При E=FALSE STATE:=0.

Используемые функции: T_PLC_MS (OSCAT).

11.1.14.10. D_TRIG (OSCAT)



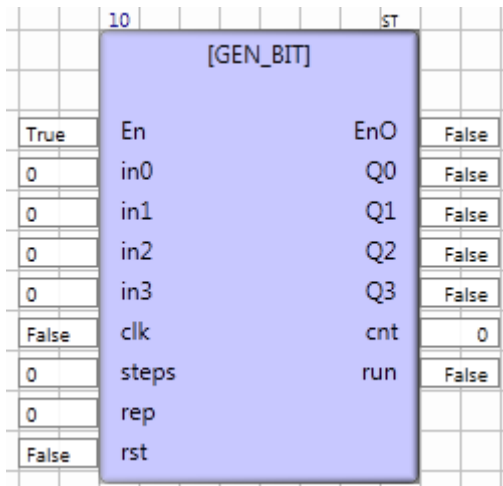
Тип данных входа IN – DWORD, выхода Q – BOOL, выхода X – DWORD.

К входу IN может быть привязана переменная любого числового типа данных или BOOL (приводится к DWORD).

При любом изменении приведенного значения IN на выходе Q генерируется импульс шириной 1 цикл, а выход X индицирует, также в течение 1 цикла, последнее изменение приведенного значения IN.

Замечание по выходу X. Т.к. DWORD – беззнаковый тип данных, при изменении (-1) приведенного значения IN выход X принимает значение не (-1), а 232-2. Чтобы получить отрицательные изменения в корректном виде, нужно привести тип данных X к типу данных со знаком – например, с помощью стандартной функции DWORD_TO_INT.

11.1.14.11. GEN_BIT (OSCAT)



Входы ФБ:

- IN0 (тип данных DWORD) – образец для последовательности Q0;
- IN1 (тип данных DWORD) – образец для последовательности Q1;
- IN2 (тип данных DWORD) – образец для последовательности Q2;
- IN3 (тип данных DWORD) – образец для последовательности Q3;
- CLK (BOOL) – сигнал тактового генератора (ширина импульса сигнала генератора не должна быть больше, чем 1 цикл). Если требуется передавать биты с частотой циклов, достаточно присвоить 1 входу CLK;
- STEPS (INT) – число бит образцов, используемых для генерации;
- REP (INT) – число повторений заданных битов образцов в генерируемых последовательностях;
- RST (BOOL) – сброс.

Выходы ФБ:

- Q0 (BOOL) – генерируемая последовательность по образцу IN0;
- Q1 (BOOL) – генерируемая последовательность по образцу IN1;
- Q2 (BOOL) – генерируемая последовательность по образцу IN2;
- Q3 (BOOL) – генерируемая последовательность по образцу IN3;
- CNT (INT) – текущий номер передаваемого бита;
- RUN (BOOL) – 1 при генерации. 0 после 1 означает окончание генерации.

GEN_BIT – это программируемый генератор прямоугольных сигналов. Начиная с битов 0, биты образцов последовательно передаются на соответствующие выходы

синхронно с сигналом тактового генератора, и таким образом на выходах формируются прямоугольные сигналы. После первого импульса тактового генератора на выход Q0 передается бит 0 IN0, на выход Q1 – бит 0 IN1 и т.д. При следующем импульсе тактового генератора на выходы Q0 .. Q3 передаются биты 1 соответственно IN0 .. IN3, и т.д. до тех пор, пока заданные последовательности не будут сформированы на выходах полностью.

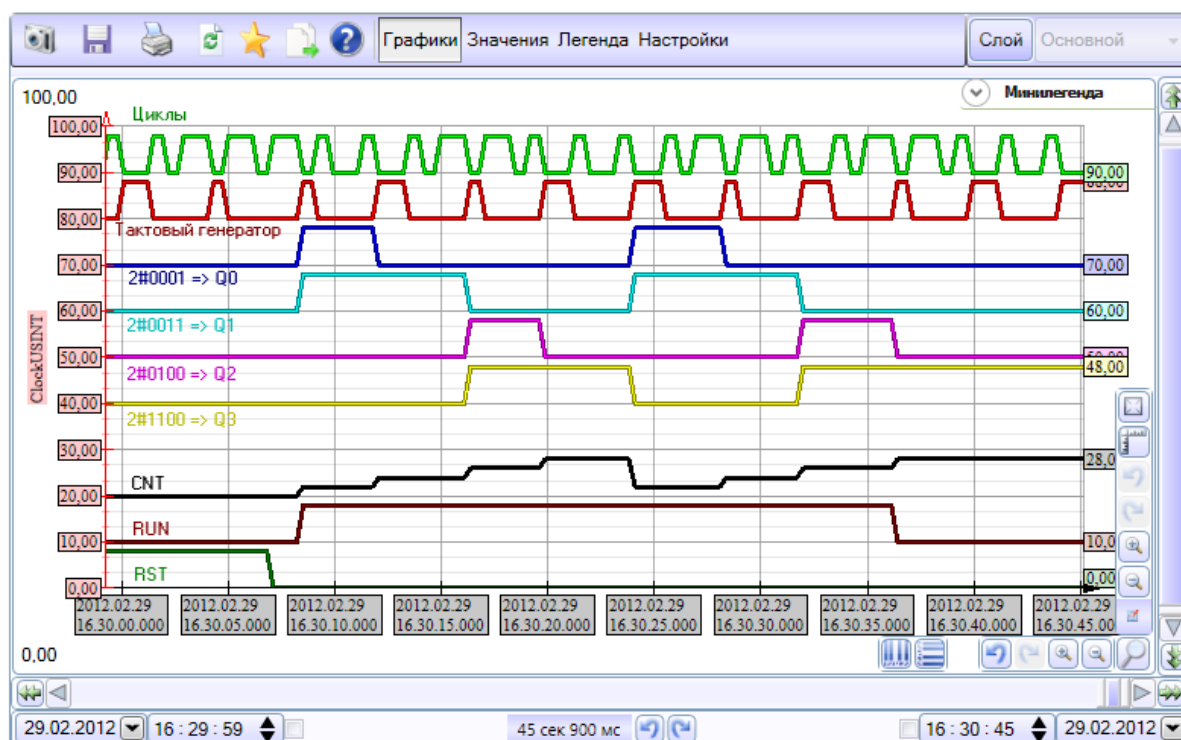
Значение STEPS задает число бит (начиная с младшего), передаваемых из образцов на выходы для формирования последовательностей.

Вход REP определяет число повторений образцов в генерируемых последовательностях. Если REP=0, образцы повторяются бесконечно.

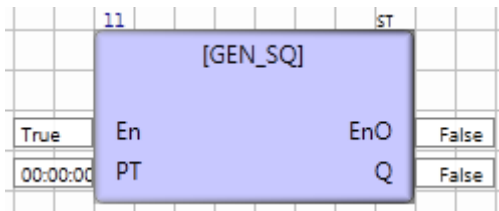
После окончания генерации, как и после каждого переданного бита, выходы не обнуляются и сохраняют последние принятые значения. Это, в частности, приводит к тому, что по единичным битам, идущим в образце подряд, генерируется один импульс соответствующей ширины.

Для прерывания генерации и обнуления выходов нужно присвоить значение TRUE входу RST. Для запуска генерации нужно присвоить FALSE входу RST (после RST=TRUE).

В качестве примера на рисунке ниже показаны последовательности, генерируемые при STEPS=4 и REP=2 (образцы указаны на рисунке):

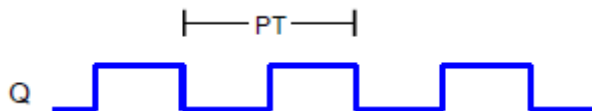


11.1.14.12. GEN_SQ (OSCAT)



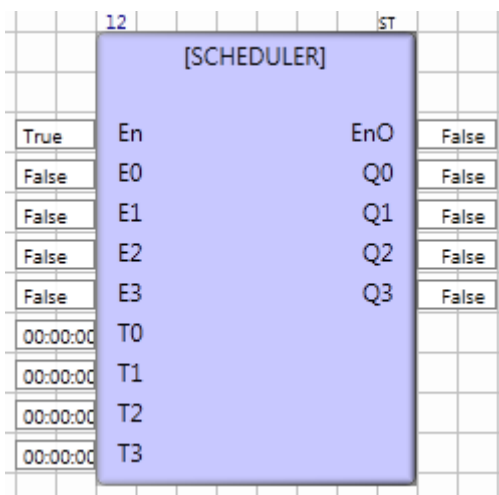
Тип данных входа PT – TIME, выхода Q – BOOL.

GEN_SQ генерирует на выходе Q меандр с заданным периодом PT:



Используемые функции: T_PLC_MS (OSCAT).

11.1.14.13. SCHEDULER (OSCAT)

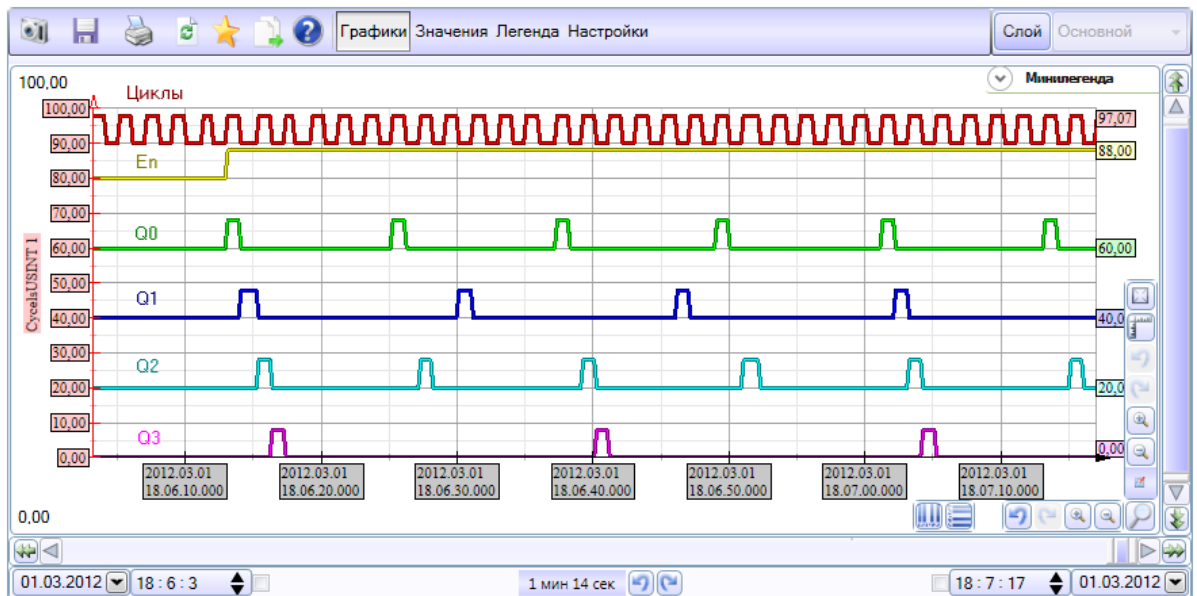


Тип данных входов E0 .. E3 и соответствующих им выходов Q0 .. Q3 – BOOL, тип данных входов T0 .. T3 – TIME.

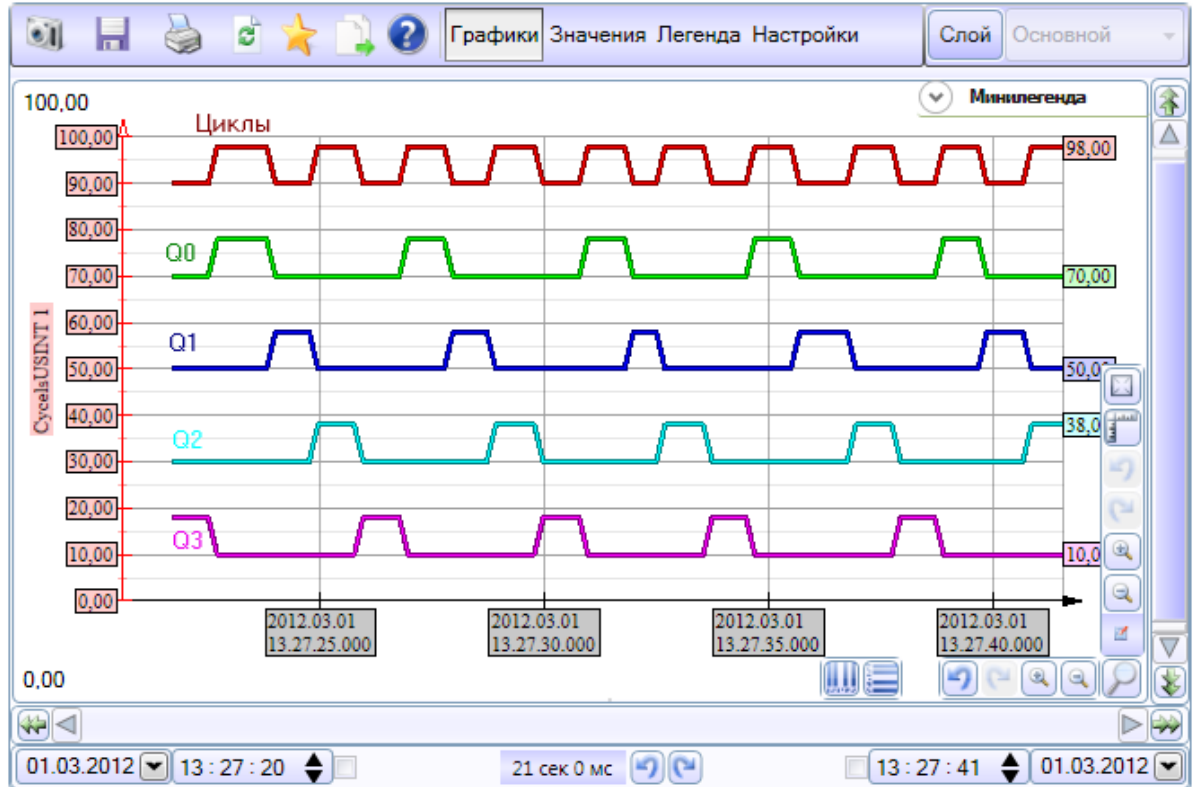
ФБ SCHEDULER, предназначенный для вызова программ по расписанию, работает по следующему алгоритму:

- при старте ФБ в течение 4 циклов последовательно передает значения входов E0 .. E3 соответственно на выходы Q0.. Q3;
- в дальнейшем ФБ передает на выходы Q0 .. Q1 значения соответствующих входов E0 .. E3 с соответствующим периодом T0 .. T3 с соблюдением следующих условий:
- выход может быть равен 1 в течение только одного цикла;
- в течение одного цикла совершается только одна передача (т.е., только один выход может быть равен 1 в течение одного цикла).

На рисунке ниже показаны выходные сигналы после старта ФБ (после $E_n=1$) при $E_0=E_1=E_2=E_3=1$, $T_0=10\text{с}$, $T_1=15\text{с}$, $T_2=10\text{с}$ и $T_3=20\text{с}$ (длительность цикла – 1с):



Условие равенства единице только одного выхода в течение цикла может приводить к увеличению $T_{<n>}$. Например, при длительности цикла 1с и всех $T_{<n>} = 0$ реальный период составит 4с:



Используемые функции: T_PLC_MS (OSCAT).

11.1.14.14. SCHEDULER_2 (OSCAT)

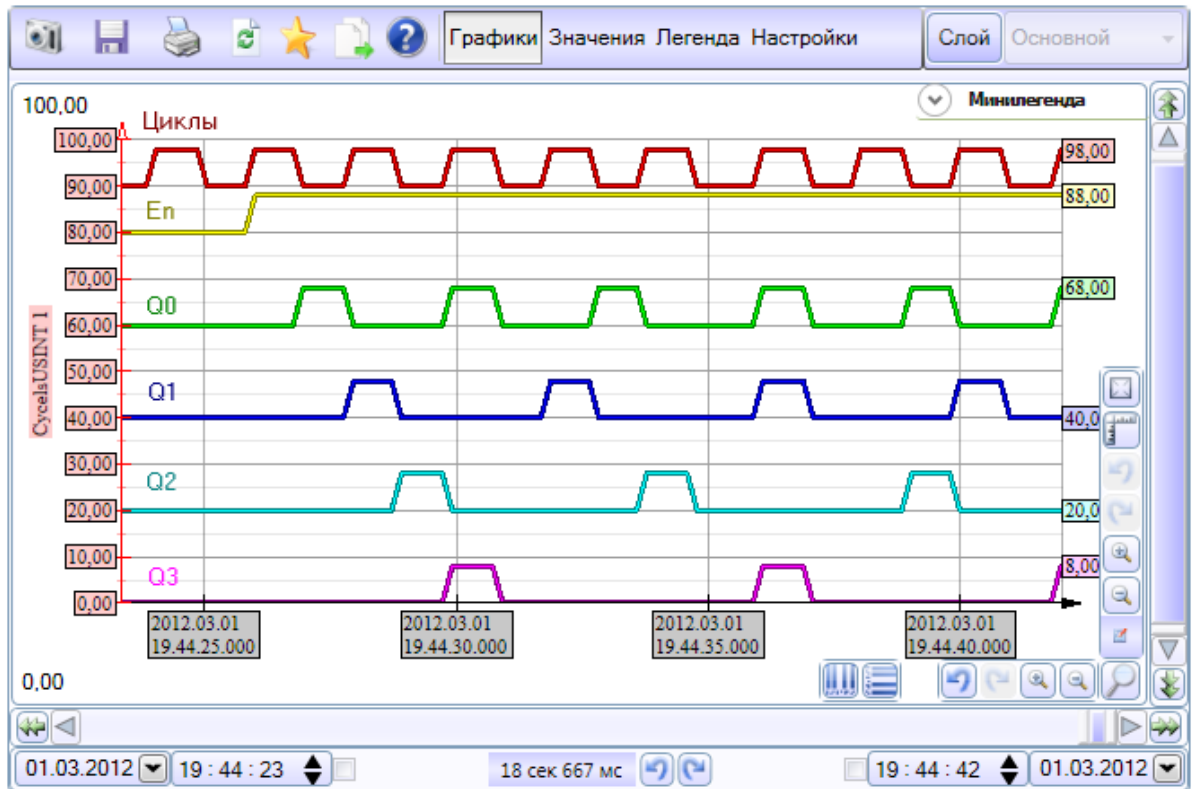
	0		ST	
	[SCHEDULER_2]			
False	En		EnO	False
True	E0		Q0	False
True	E1		Q1	False
True	E2		Q2	False
True	E3		Q3	False
3	C0			
4	C1			
5	C2			
6	C3			
1	O0			
2	O1			
3	O2			
4	O3			

Тип данных входов E0 .. E3 и соответствующих им выходов Q0 .. Q3 – BOOL, тип данных входов C0 .. C3 и O0 .. O3 – UINT.

SCHEDULER_2 используется для вызова программ на определенных циклах. При $E_{<n>} = 1$ этот ФБ активировывает соответствующий выход $Q_{<n>}$ в зависимости от $C_{<n>}$ и $O_{<n>}$.

C0 .. C3 определяют период активации (в циклах) соответствующих выходов Q0 .. Q3, а с помощью O0 .. O3 задается сдвиг (в циклах) при старте.

При конфигурации ФБ, показанной на рисунке выше, после $E_n=TRUE$ на выходах формируются следующие сигналы:



11.1.14.15. SEQUENCE_4 (OSCAT)

[SEQUENCE_4]		ST
True	En	EnO True
True	in0	Q0 False
True	in1	Q1 False
True	in2	Q2 False
True	in3	Q3 False
False	start	QX False
False	rst	run False
00:00:05	wait0	stepp -1
00:00:05	delay0	status 110
00:00:05	wait1	
00:00:02	delay1	
00:00:05	wait2	
00:00:03	delay2	
00:00:05	wait3	
00:00:02	delay3	
False	stop_on_erro	

Входы ФБ:

- IN0..IN3 (BOOL) – разрешающие входы соответственно для выходов Q0..Q3;

- START (BOOL) – запуск/перезапуск алгоритма ФБ;
- RST (BOOL) – сброс;
- WAIT0..WAIT3 (TIME) – время ожидания 1 соответственно на входе IN0..IN3;
- DELAY0..DELAY3 (TIME) – длительность генерируемого импульса соответственно на выходе Q0..Q3;
- STOP_ON_ERROR (BOOL) – запрет/разрешение останова алгоритма в случае отсутствия 1 на входе IN<n>.

Выходы ФБ:

- Q0..Q3 (BOOL) – генерируемые импульсы;
- QX (BOOL) – TRUE, если один из выходов Q0..Q3 равен 1;
- RUN (BOOL) – TRUE, если алгоритм ФБ запущен;
- STEPP (INT) – номер обрабатываемого входа;
- STATUS (BYTE) – ESR-совместимый выход статуса.
- ФБ вырабатывает на выходах Q0..Q3 импульсы в зависимости от своей конфигурации.

Алгоритм SEQUENCE_4 запускается по переходу FALSE=>TRUE на входе START; для перезапуска нужно выполнить на этом входе переход TRUE=>FALSE и затем FALSE=>TRUE.

Если RST=TRUE, алгоритм ФБ не выполняется, выходы обнуляются, а команды запуска/перезапуска алгоритма игнорируются. Если RST=FALSE после RST=TRUE, для возобновления работы требуется перезапуск алгоритма ФБ.

Если алгоритм ФБ запущен, RUN=TRUE, и ФБ ожидает 1 на IN0 в течение WAIT0. Если 1 детектируется, Q0=1 на время DELAY0. По окончании импульса на Q0 ФБ ожидает 1 на входе IN1 в течение WAIT1 и т.д. (таким образом, в каждом цикле только один выход может быть равен 1).

Если 1 на каком-либо входе не детектируется в течение соответствующего времени WAIT<n>, дальнейшее поведение ФБ зависит от значения STOP_ON_ERROR:

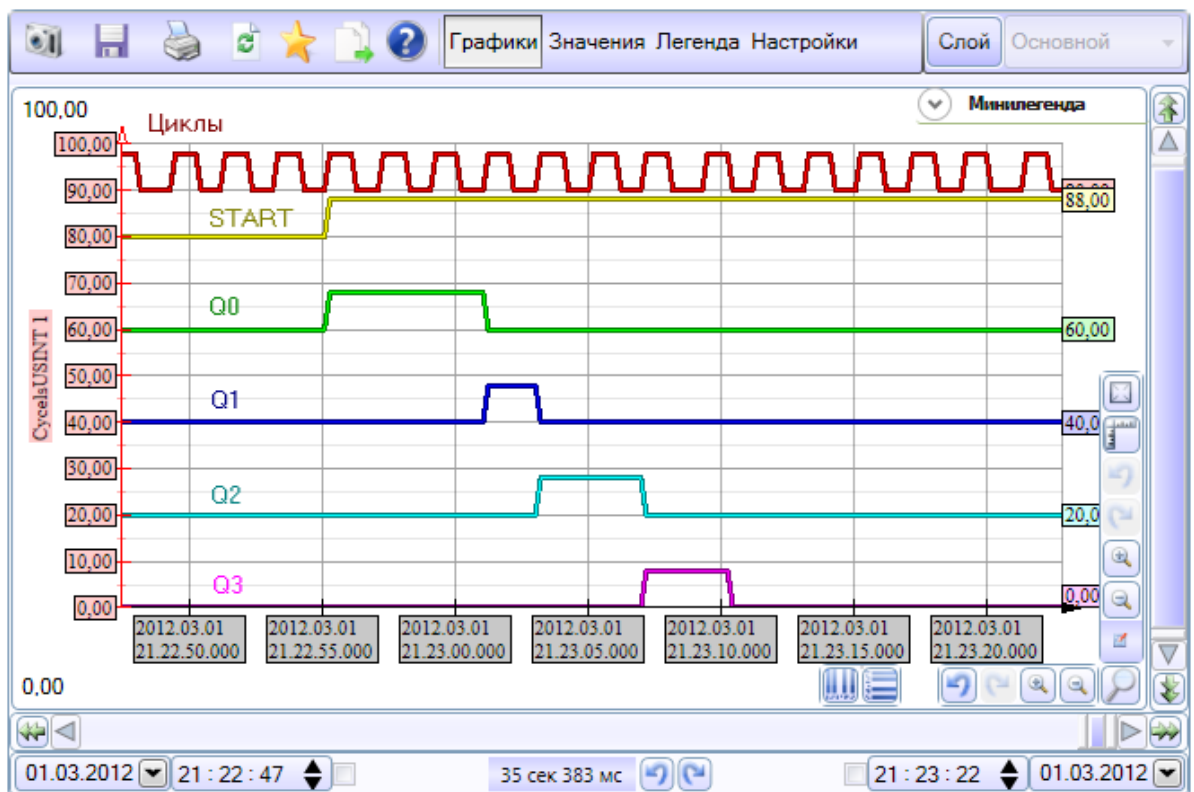
- если STOP_ON_ERROR=TRUE, алгоритм ФБ завершает работу (RUN=FALSE), выход STEPP индицирует номер входа с ошибкой, а STATUS=STEPP+1;
- если STOP_ON_ERROR=FALSE, алгоритм ФБ продолжает работу (так же, как если бы 1 на соответствующем входе детектирована в самом конце интервала ожидания, но без генерации импульса на Q<n>).

Выход STATUS индицирует 110, если алгоритм остановлен, и 111, если алгоритм запущен.

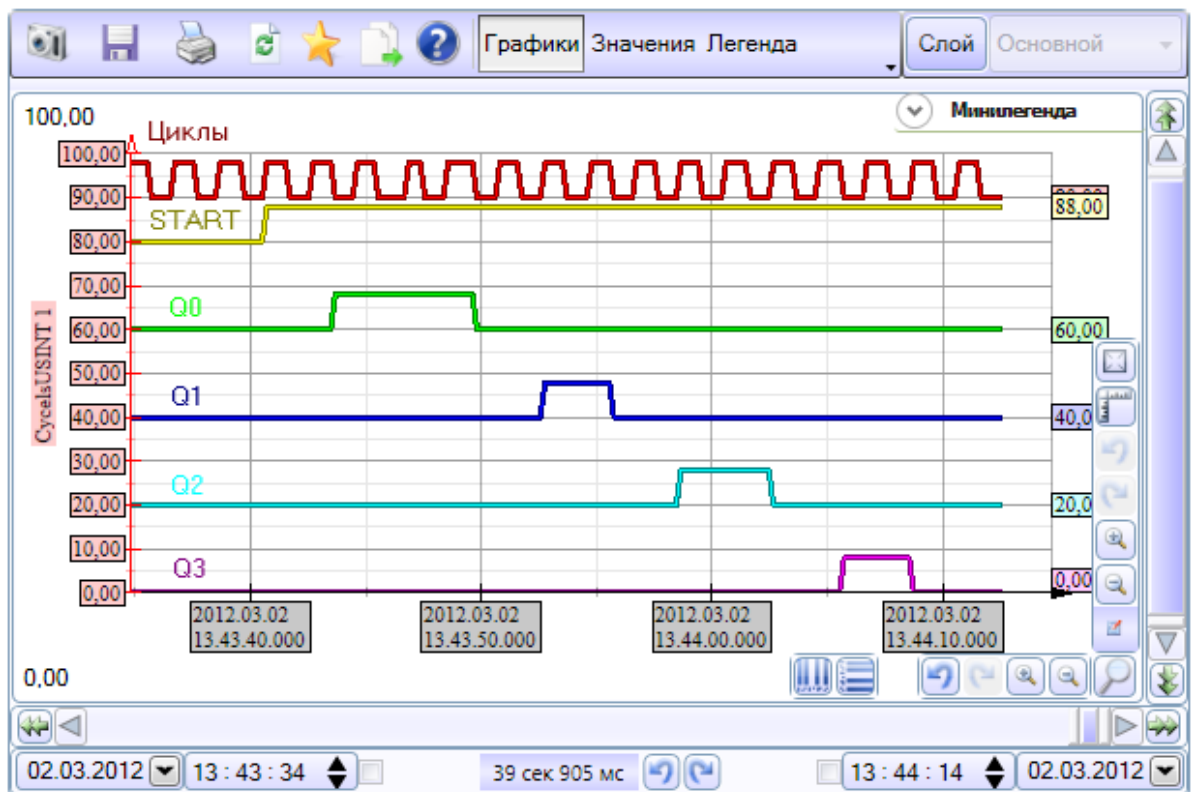
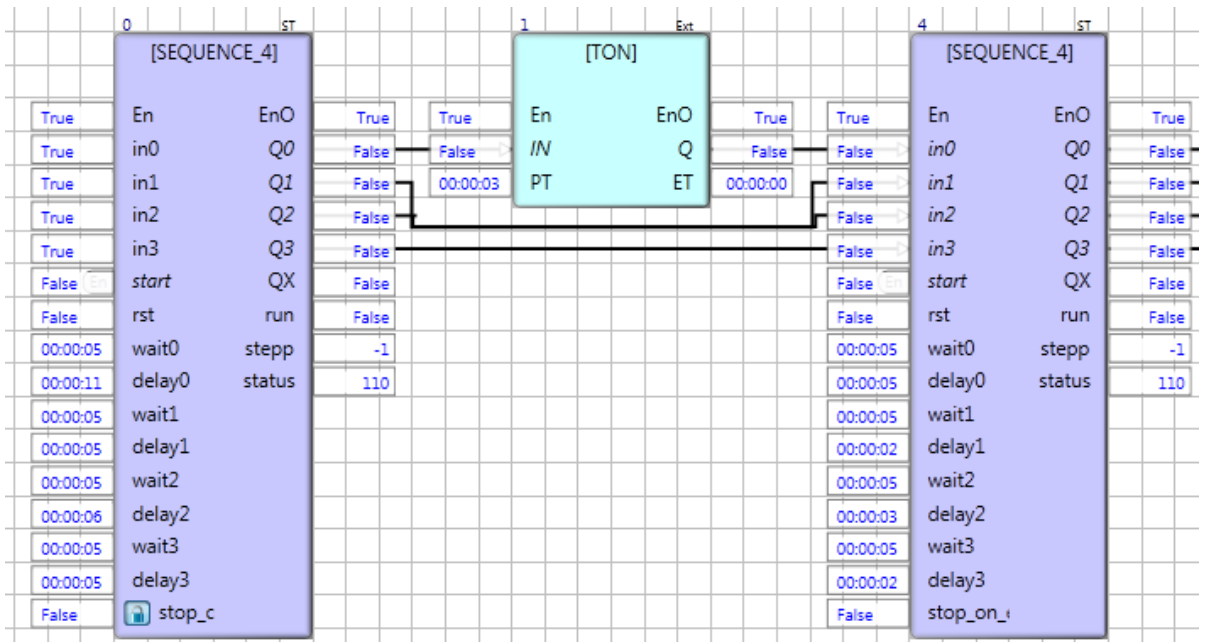
После последнего шага, на котором генерируется импульс на Q3, ФБ возвращается в исходное состояние.

Используемые функции: T_PLC_MS (OSCAT).

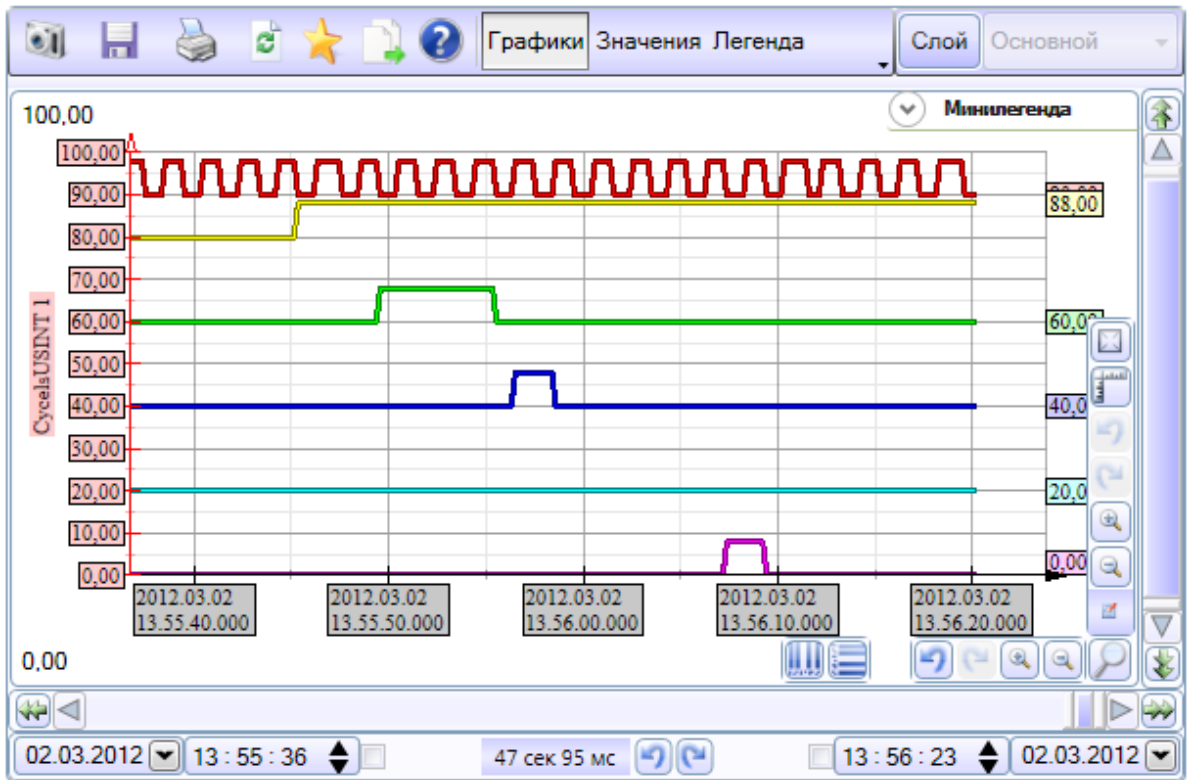
При конфигурации ФБ, показанной на рисунке выше, после START=TRUE на выходах формируются следующие сигналы (длительность цикла – 1с):



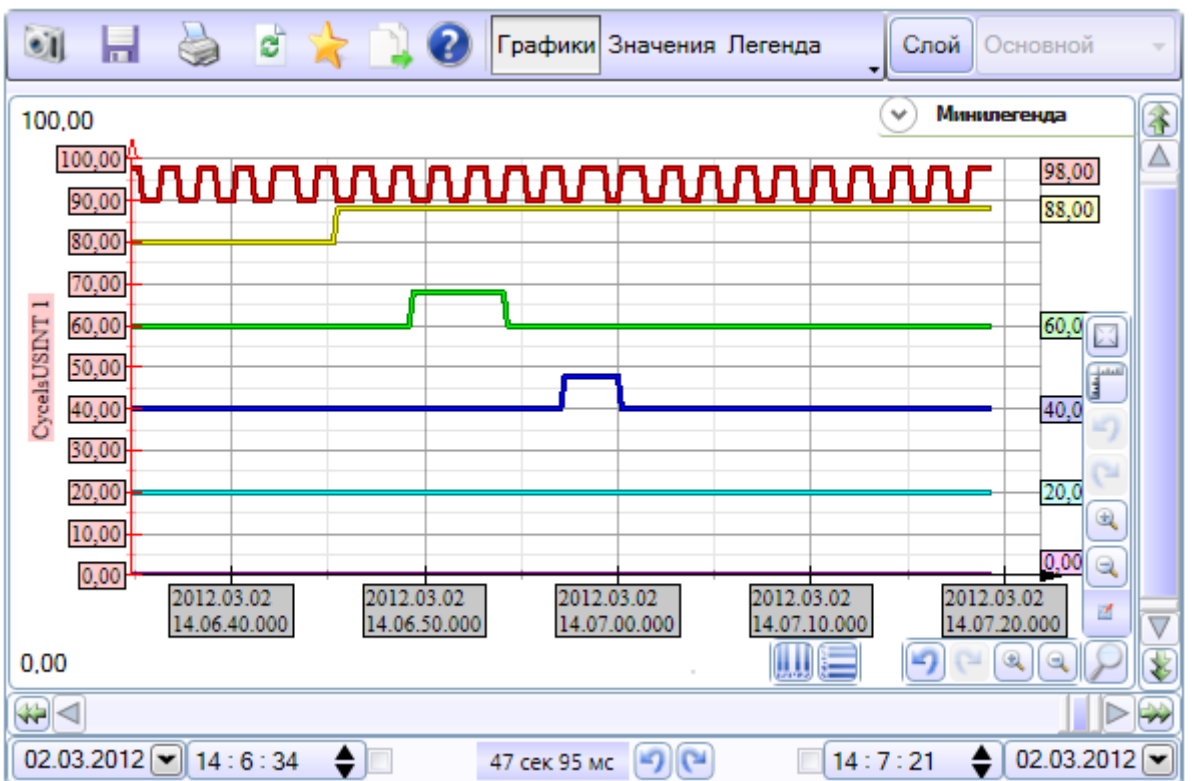
В следующей программе левый ФБ SEQUENCE_4 и ФБ TON, задержка включения имитируют задержку в 3с появления 1 на каждом входе правого ФБ SEQUENCE_4 относительно заднего фронта импульса на предыдущем выходе:

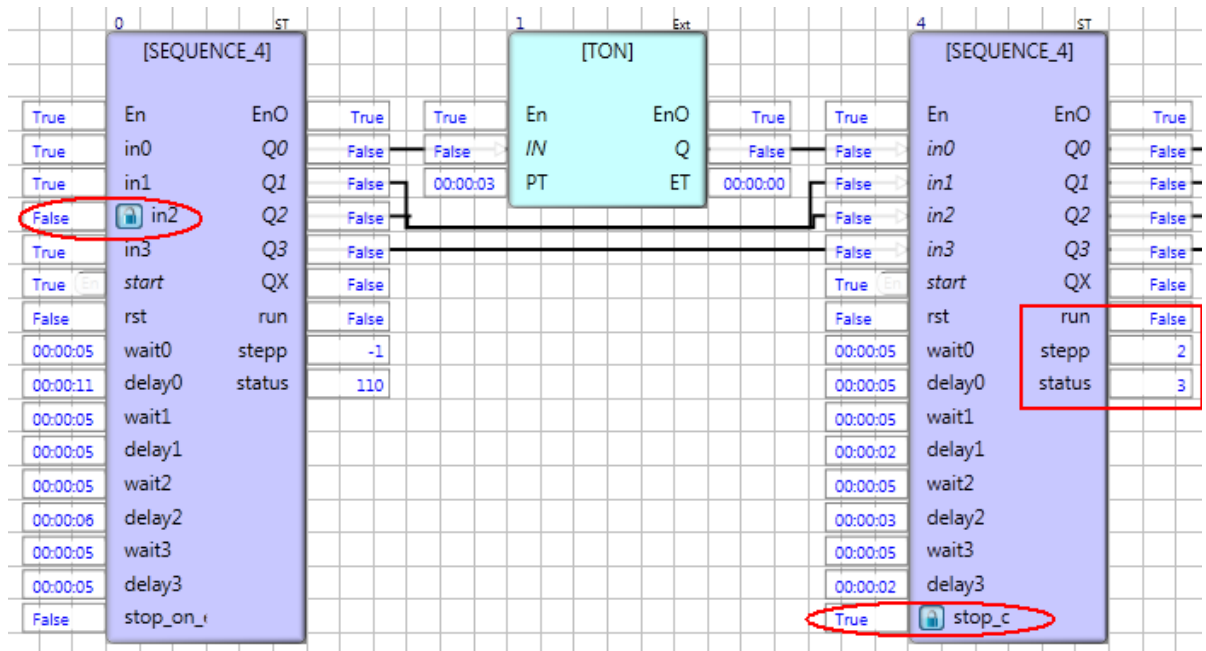


Следующая диаграмма демонстрирует игнорирование отсутствия 1 на IN2 (IN2 левого ФБ SEQUENCE_4 – FALSE, STOP_ON_ERROR обоих ФБ – FALSE):

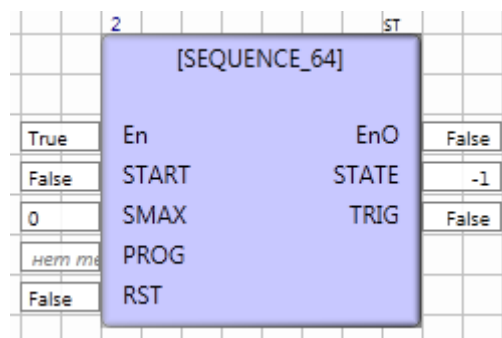


Следующая диаграмма демонстрирует останов генерации при отсутствии 1 на IN2 (IN2 левого ФБ SEQUENCE_4 – FALSE, STOP_ON_ERROR правого ФБ SEQUENCE_4 – TRUE):





11.1.14.16. SEQUENCE_64 (OSCAT)



Входы ФБ:

- START (BOOL) – передний фронт на этом входе запускает алгоритм ФБ;
- SMAX (INT) – последнее состояние ФБ;
- PROG (ARRAY [0..63] OF TIME) – длительности состояний;
- RST (BOOL) – сброс.

Выходы ФБ:

- STATE (INT) – текущее состояние;
- TRIG (BOOL) – индикатор изменения состояния.

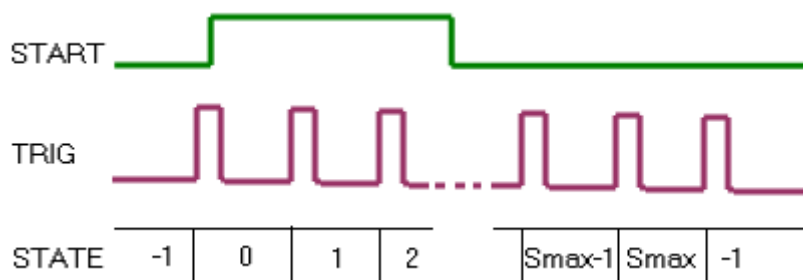
SEQUENCE_64 предназначен для последовательного прохождения состояний 1..N (NMAX=64). В последнем состоянии STATE = -1 ("спящее" состояние ФБ, т.е. алгоритм остановлен, но готов к старту).

Передний фронт на входе START запускает алгоритм, и выход STATE обнуляется. После времени ожидания PROG[0] STATE = 1, затем, после времени ожидания PROG[1], STATE = 2 и т.д. до тех пор, пока выход STATE не достигнет значения SMAX. Далее, после времени ожидания PROG[SMAX], ФБ переходит в "спящее" состояние (STATE = -1).

При переходе в новое состояние (т.е. при изменении STATE) TRIG:=TRUE в течение 1 цикла (выход TRIG предназначен для контроля работы ФБ).

При RST=TRUE ФБ переходит в начальное состояние.

Временная диаграмма SEQUENCE_64:

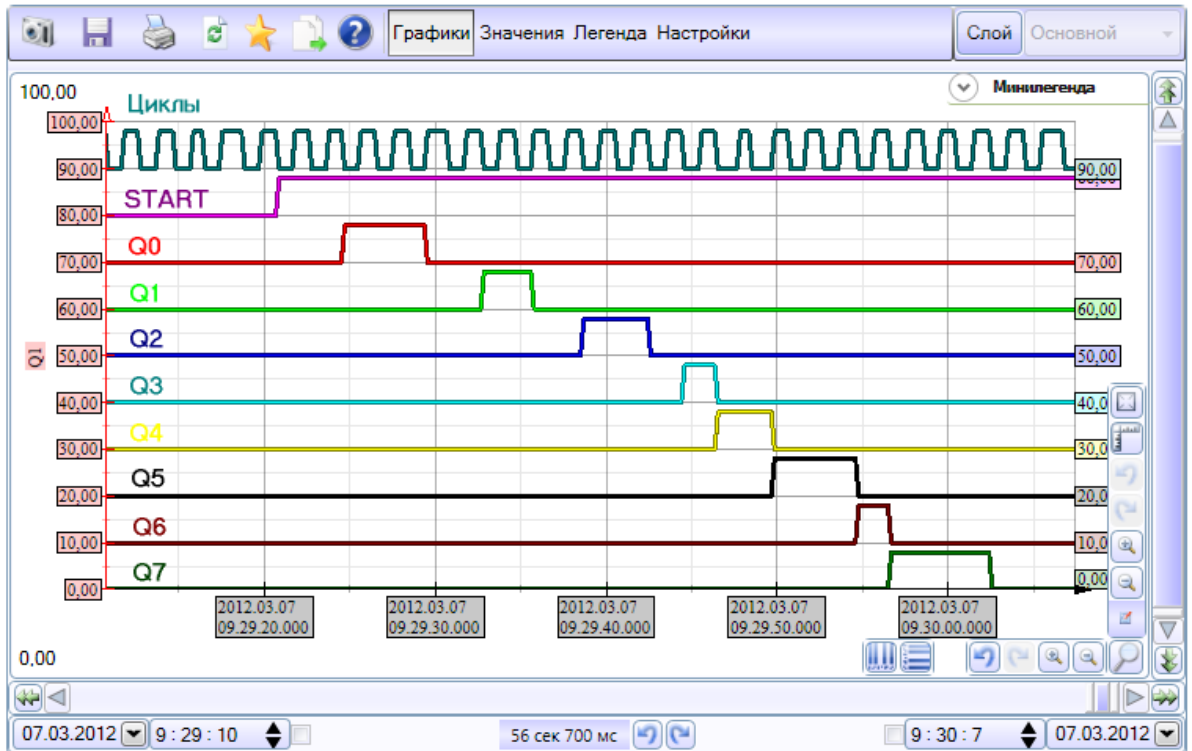


Используемые функции: T_PLC_MS (OSCAT), INC2 (OSCAT).

11.1.14.17. SEQUENCE_8 (OSCAT)

	0		ST	
		[SEQUENCE_8]		
True	En		EnO	False
True	in0		Q0	False
True	in1		Q1	False
True	in2		Q2	False
True	in3		Q3	False
True	in4		Q4	False
True	in5		Q5	False
True	in6		Q6	False
True	in7		Q7	False
False	start		QX	False
False	rst		run	False
00:00:00	wait0		stepp	-1
00:00:00	delay0		status	0
00:00:00	wait1			
00:00:00	delay1			
00:00:00	wait2			
00:00:00	delay2			
00:00:00	wait3			
00:00:00	delay3			
00:00:00	wait4			
00:00:00	delay4			
00:00:00	wait5			
00:00:00	delay5			
00:00:00	wait6			
00:00:00	delay6			
00:00:00	wait7			
00:00:00	delay7			
False	stop_on_error			

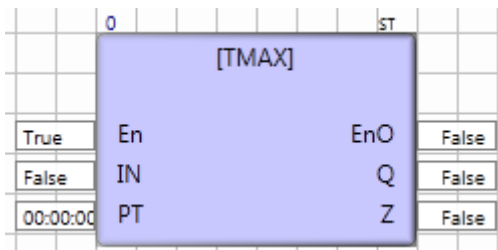
Данный ФБ представляет собой 8-канальный аналог ФБ SEQUENCE_4 (OSCAT):



Используемые функции: T_PLC_MS (OSCAT).

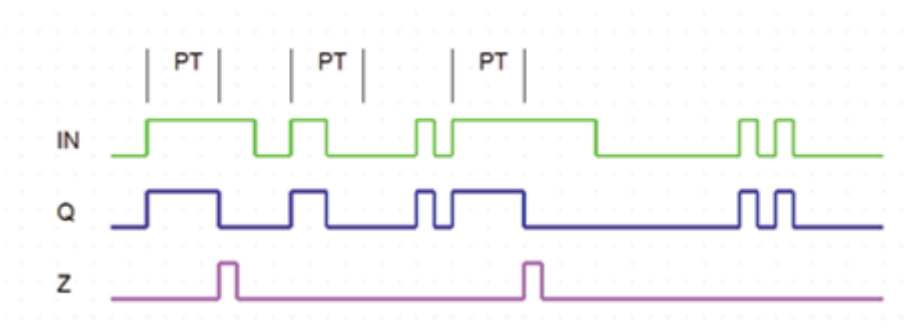
SEQUENCE_8 используется в ФБ Legionella (OSCAT).

11.1.14.18. TMAX (OSCAT)



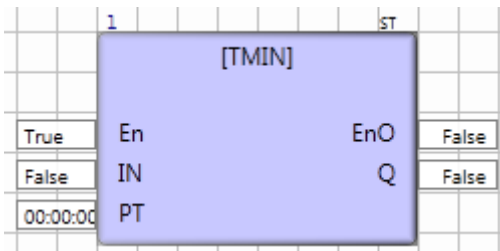
Тип данных входа PT – TIME, входа IN и выходов Q и Z – BOOL.

Если импульсы, подаваемые на вход IN, короче, чем PT, ФБ передает их на выход Q без изменения, в противном случае обрезает длительность импульса до PT. При каждом обрезании Z=TRUE в течение 1 цикла.



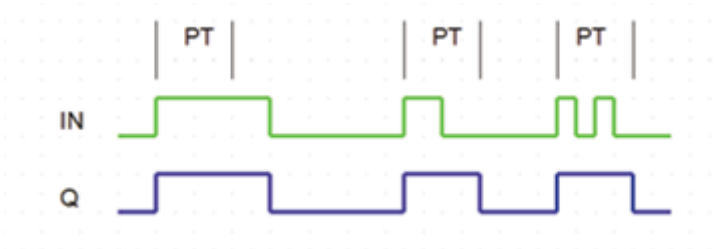
Используемые функции: T_PLC_MS (OSCAT).

11.1.14.19. TMIN (OSCAT)

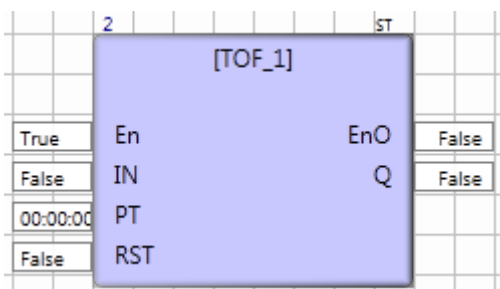


Тип данных входа PT – TIME, входов IN и выхода Q – BOOL.

Если импульс, подаваемый на вход IN, длиннее, чем PT, ФБ передает его на выход Q без изменения, в противном случае увеличивает длительность импульса до PT. В случае нескольких импульсов на входе IN за время PT на выходе Q формируется один импульс.

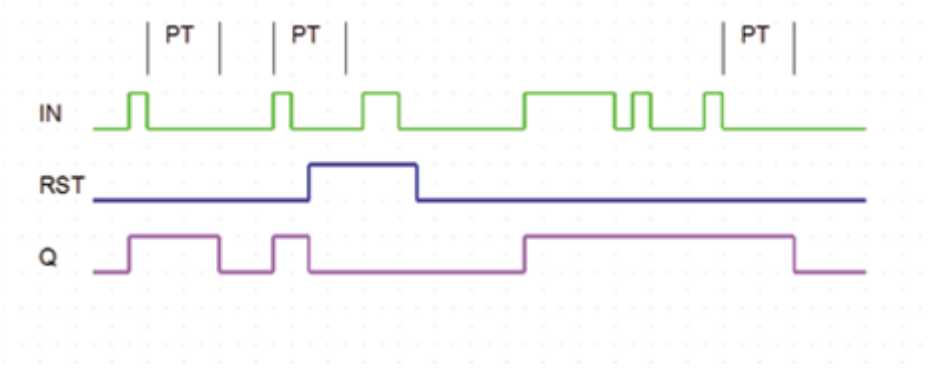


11.1.14.20. TOF_1 (OSCAT)



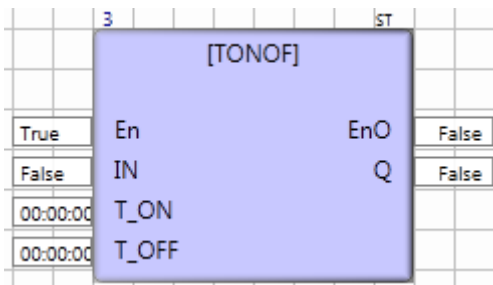
Тип данных входа PT – TIME, входов IN и RST и выхода Q – BOOL.

TOF_1 – это аналог стандартного ФБ TOF, задержка выключения с возможностью сброса импульса на выходе Q по команде RST=TRUE.



Используемые функции: T_PLC_MS (OSCAT).

11.1.14.21. TONOF (OSCAT)

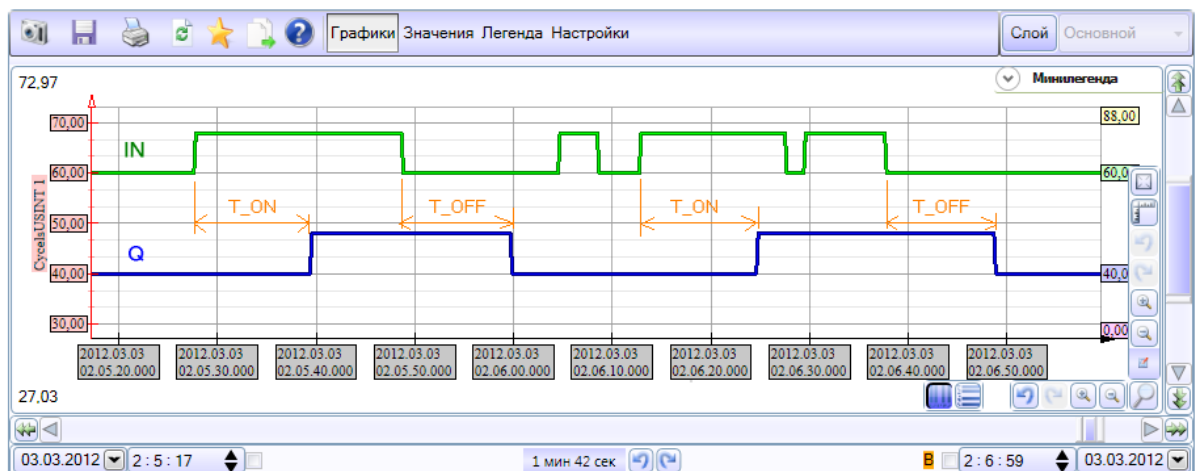


Тип данных входов T_ON и T_OFF – TIME, входа IN и выхода Q – BOOL.

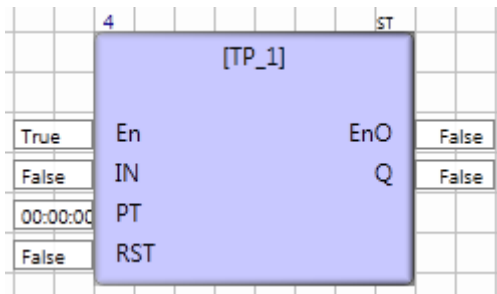
TONOF задерживает включение (передний фронт на IN) на T_ON и выключение (задний фронт) – на T_OFF.

Для корректной работы ФБ ширина импульса включения должна быть больше T_ON, а после заднего фронта выключения не должно быть импульсов.

На рисунке ниже T_ON=T_OFF=10с:



11.1.14.22. TP_1 (OSCAT)

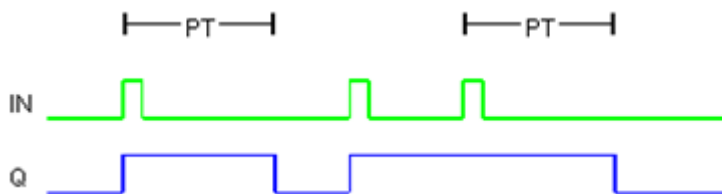


Тип данных входа PT – TIME, входов IN и RST и выхода Q – BOOL.

TP_1 по переднему фронту на входе IN генерирует на выходе Q импульс шириной PT. Если в течение генерируемого импульса (в некоторый момент t_0) на входе IN вновь детектируется передний фронт, длительность генерируемого импульса увеличивается (с момента t_0 она будет равна PT).

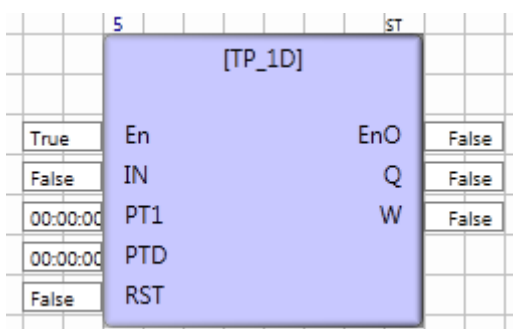
Для останова алгоритма нужно присвоить TRUE входу RST (выход Q при этом обнуляется).

Временная диаграмма ФБ:



Используемые функции: T_PLC_MS (OSCAT).

11.1.14.23. TP_1D (OSCAT)



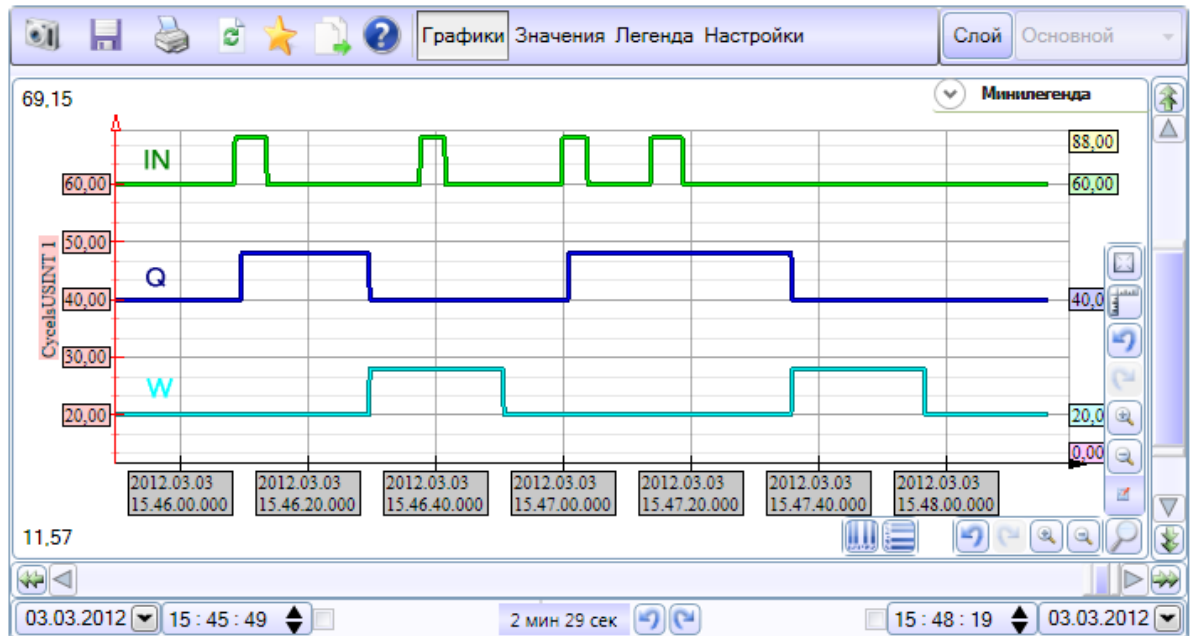
Тип данных входов IN и RST и выходов Q и W – BOOL, входов PT1 и PTD – TIME.

TP_1D по переднему фронту на входе IN генерирует на выходе Q импульс шириной PT1. Если в течение генерируемого импульса (в некоторый момент t_0) на входе IN вновь детектируется передний фронт, длительность генерируемого импульса увеличивается (с момента t_0 она будет равна PT1). По окончании генерации импульса алгоритм генерации блокируется на время PTD (при блокировке W=TRUE).

При $RST=TRUE$ алгоритм генерации блокируется на неопределенное время (до $RST=FALSE$); выходы Q и W при этом обнуляются.

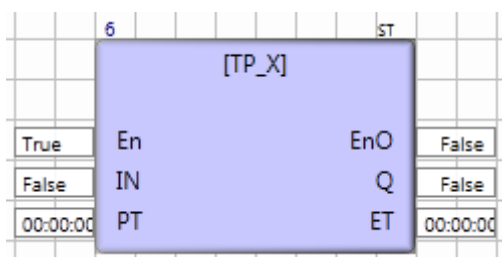
Важно! Данный ФБ модифицирован, в нем отсутствует автоматический сброс IN и RST .

Временная диаграмма ФБ:



Используемые функции: T_PLC_MS (OSCAT).

11.1.14.24. TP_X (OSCAT)

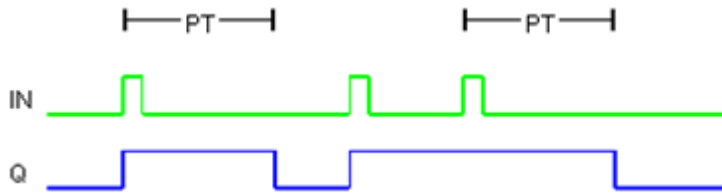


Тип данных входа IN и выхода Q – $BOOL$, входа PT и выхода ET – $TIME$.

TP_X – аналог стандартного ФБ TP , импульс произвольной длительности с возможностью увеличения длительности генерируемого импульса: если в течение генерируемого импульса (в некоторый момент t_0) на входе IN вновь детектируется передний фронт, длительность генерируемого импульса увеличивается (с момента t_0 она будет равна PT).

Выход ET индицирует время, истекшее с момента детектирования последнего переднего фронта на входе IN .

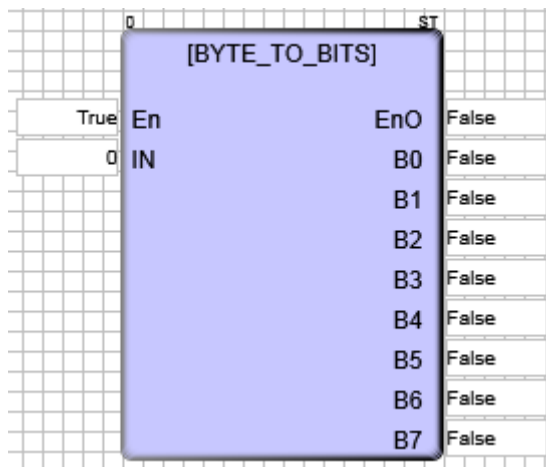
Временная диаграмма ФБ:



Используемые функции: T_PLC_MS (OSCAT).

11.1.15. OSCAT.ФБ.ЛОГИЧЕСКИЕ МОДУЛИ

11.1.15.1. BYTE_TO_BITS (OSCAT)

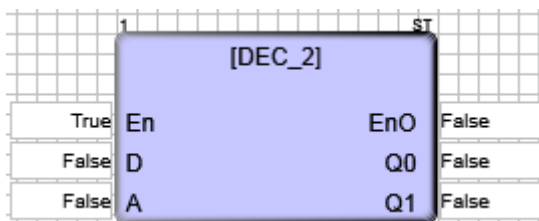


Тип данных входа IN – BYTE, выходов B0..B7 – BOOL.

Выходы B0..B7 индицируют значения битов 0..7 значения IN.

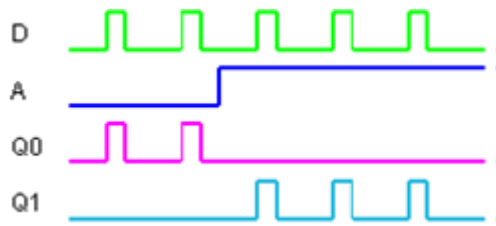
К входу IN может быть привязана переменная DWORD, WORD или BYTE (во всех случаях распаковываются биты 0..7).

11.1.15.2. DEC_2 (OSCAT)

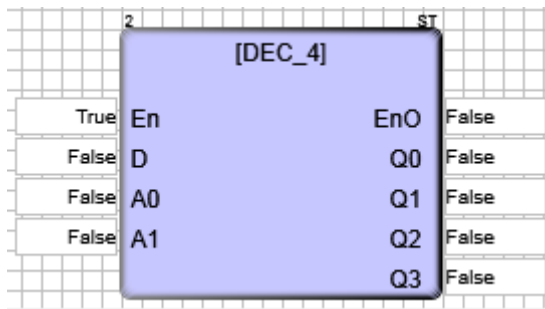


Тип данных входов и выходов – BOOL.

DEC_2 – это двухпозиционный переключатель. Если A=0, значение входа D передается на выход Q0 (при этом Q1 принимает значение FALSE), если A=1 – на выход Q1 (при этом Q0 принимает значение FALSE).



11.1.15.3. DEC_4 (OSCAT)

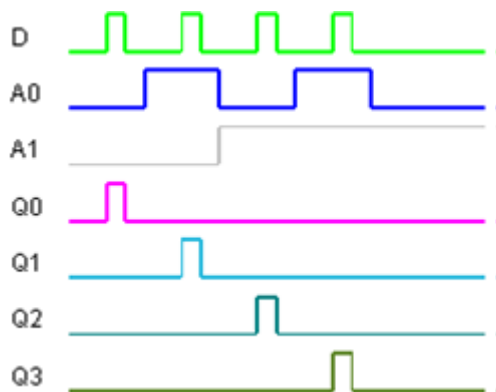


Тип данных входов и выходов – BOOL.

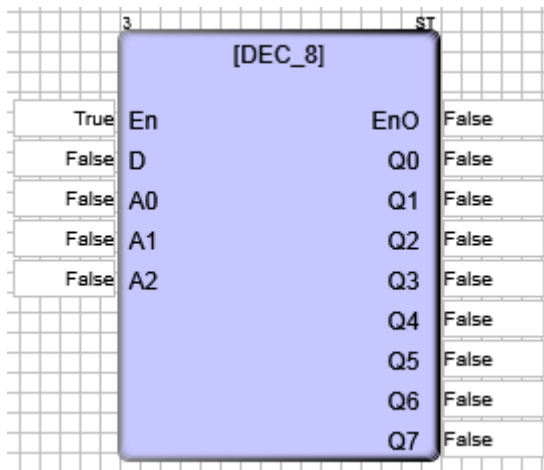
DEC_4 – это 4-позиционный переключатель:

- если $A1=A0=0$, значение входа D передается на выход Q0;
- если $A1=0$ и $A0=1$, значение входа D передается на выход Q1;
- если $A1=1$ и $A0=0$, значение входа D передается на выход Q2;
- если $A1=A0=1$, значение входа D передается на выход Q3.

Выходы, на которые значение не передается, принимают значение FALSE.



11.1.15.4. DEC_8 (OSCAT)

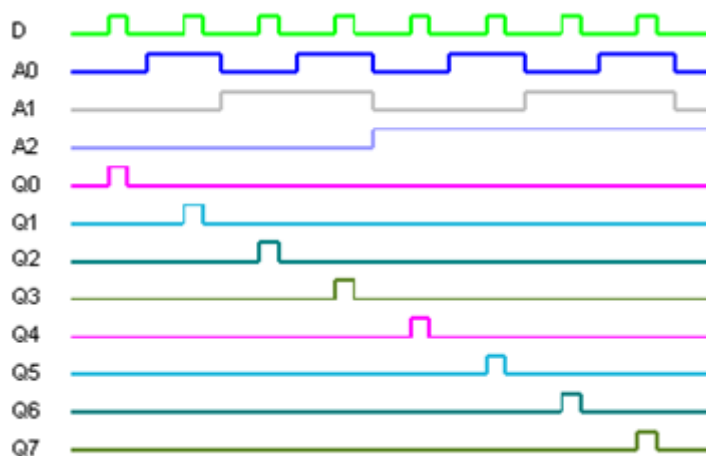


Тип данных входов и выходов – BOOL.

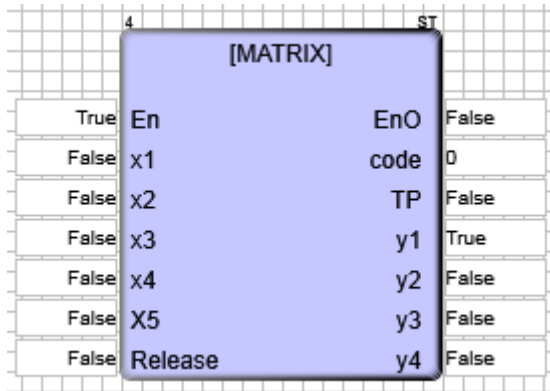
DEC_8 – это 8-позиционный переключатель:

- если $A2=A1=A0=0$, значение входа D передается на выход Q0;
- если $A2=A1=0$ и $A0=1$, значение входа D передается на выход Q1;
- если $A2=0$, $A1=1$ и $A0=0$, значение входа D передается на выход Q2;
- если $A2=0$ и $A1=A0=1$, значение входа D передается на выход Q3;
- если $A2=1$ и $A1=A0=0$, значение входа D передается на выход Q4;
- если $A2=1$, $A1=0$ и $A0=1$, значение входа D передается на выход Q5;
- если $A2=1$, $A1=1$ и $A0=0$, значение входа D передается на выход Q6;
- если $A2=A1=A0=1$, значение входа D передается на выход Q7.

Выходы, на которые значение не передается, принимают значение FALSE.



11.1.15.5. MATRIX (OSCAT)



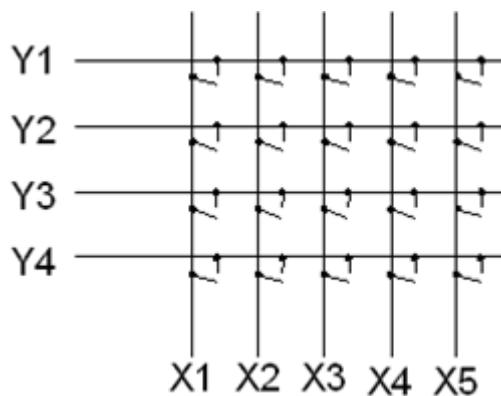
Входы ФБ:

- X1 .. X5 (BOOL) – входы линий Y1 .. Y4;
- RELEASE (CONSTANT BOOL) – если RELEASE=TRUE, генерируется код нажатия и отжатия клавиши, в противном случае генерируется только код нажатия.

Выходы ФБ:

- CODE (BYTE) – выход (генерируемый код);
- TP (BOOL) – TRUE в течение 1 цикла при изменении CODE;
- Y1 .. Y4 (BOOL) – TRUE, если линия считывается.

MATRIX – это матричный контроллер (энкодер) клавиатуры (4 линии/столбца Y1 .. Y4 и 5 входов/строк X1 .. X5):



Контроллер опрашивает линии последовательно, поэтому в течение 1 цикла только один из выходов Y1 .. Y4 имеет значение TRUE:



При опросе линии считываются входы X1 .. X5, и если клавиша нажата, соответствующий код генерируется на выходе CODE. При этом, если значение CODE изменяется, выход TP на 1 цикл принимает значение TRUE. Если RELEASE=TRUE, спустя 4 цикла на выходе дополнительно генерируется код отжатия клавиши.

Значение CODE формируется следующим образом:

- бит 7 – 1, если клавиша нажата, 0 – если отжата;
- бит 6 – бит 2 номера линии (*);
- бит 5 – бит 1 номера линии (*);
- бит 4 – бит 0 номера линии (*);

(*) Номер линии line принимает значение от 0 до 3. При опросе линии 0 Y1=1, линии 1 – Y2=1 и т.д.

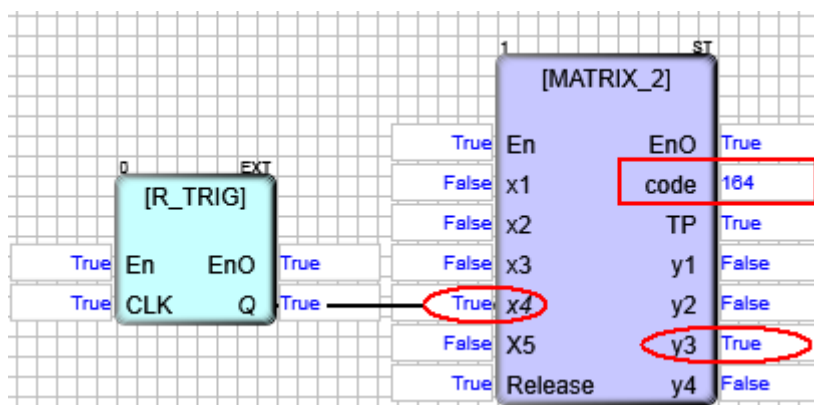
- бит 3 – всегда 0;
- бит 2 – бит 2 номера строки (**);
- бит 1 – бит 1 номера строки (**);
- бит 0 – бит 0 номера строки (**).

(**) Номер строки row принимает значение от 0 до 4. Для строки X1 row=0, для X2 row=1 и т.д.

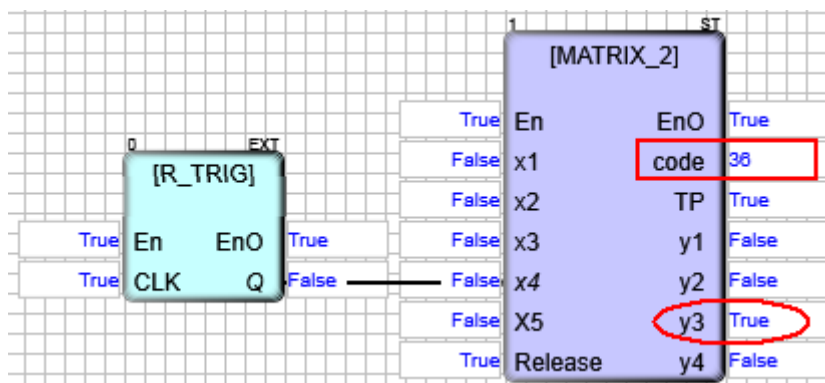
Важно! Текущий код контроллера не анализирует удержание клавиши, т.е. длина импульса от нажатия клавиши не должна превышать длительность цикла, в противном случае будут "считываться" соседние по строке клавиши. При нажатии двух клавиш одной линии корректно считывается клавиша с наименьшим номером строки.

Пример

Пусть на некотором цикле N при опросе линии 2 (line=2, Y3=1) детектируется X4=1. В этом случае CODE принимает значение 164:

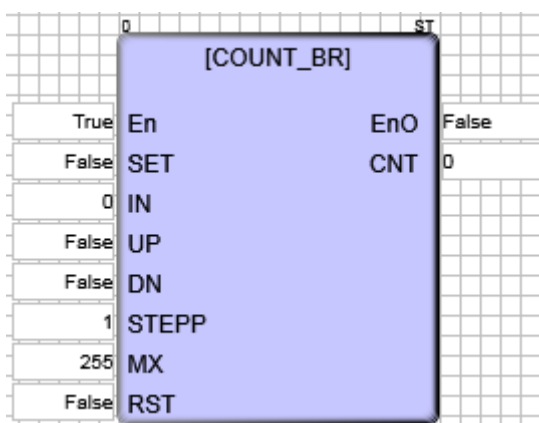


На циклах N+1, N+2 и N+3 CODE=0, а на цикле N+4 CODE=36, поскольку RELEASE=TRUE:



11.1.16. OSCAT.ФБ.ТРИГГЕРЫ

11.1.16.1. COUNT_BR (OSCAT)



Тип данных входов SET, UP, DN и RST – BOOL, входов IN, STEPP, MX и выхода CNT – BYTE.

COUNT_BR выполняет функции возрастающего/убывающего счетчика передних фронтов на входе UP/DN. Приоритетным является возрастающий счетчик (если передние фронты детектируются на входах UP и DN одновременно, выход CNT увеличивается).

Вход STEPP задает инкремент (шаг увеличения/уменьшения выхода CNT).

Если $(CNT+STEPP)>MX$ или $(CNT-STEPP)<0$, значение CNT вычисляется по правилам битовых строк (см. Особенности битовых строк).

Если SET=TRUE, CNT:=IN (если $IN \leq MX$) или CNT:=MX (если $IN > MX$), и после SET:=FALSE счетчик начинает считать с установленного значения CNT.

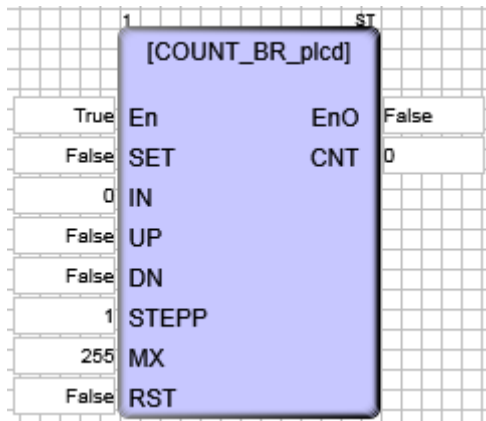
Если RST=TRUE, CNT=0.

Используемые функции: INC (OSCAT).

Пример

Пусть $MX=50$, $STEPP=10$ и детектируются передние фронты только на входе UP (возрастающий счетчик). В этом случае выход CNT будет последовательно принимать следующие значения: 0, 10, 20, 30, 40, 50, 9, 19 ...

11.1.16.2. COUNT_BR_PLCD



Тип данных входов SET, UP, DN и RST – BOOL, входов IN, STEPP, MX и выхода CNT – BYTE.

COUNT_BR_plcd выполняет функции возрастающего/убывающего счетчика передних фронтов на входе UP/DN. Приоритетным является возрастающий счетчик (если передние фронты детектируются на входах UP и DN одновременно, выход CNT увеличивается).

Вход STEPP задает инкремент (шаг увеличения/уменьшения выхода CNT).

В случае возрастающего счетчика: при достижении $CNT > MX$ счетчик начинает считать с 0. В случае убывающего счетчика: при достижении $CNT < 0$ счетчик начинает считать с MX.

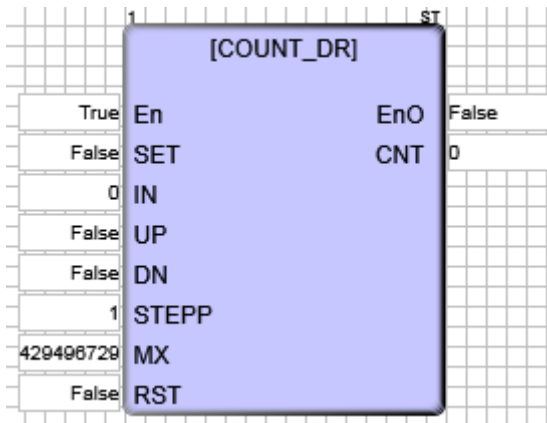
Если SET=TRUE, CNT:=IN (если $IN \leq MX$) или CNT:=MX (если $IN > MX$), и после SET:=FALSE счетчик начинает считать с установленного значения CNT.

Если RST=TRUE, CNT=0.

Пример

Пусть $MX=50$, $STEPP=10$ и детектируются передние фронты только на входе UP (возрастающий счетчик). В этом случае выход CNT будет последовательно принимать следующие значения: 0, 10, 20, 30, 40, 50, 0, 10 ...

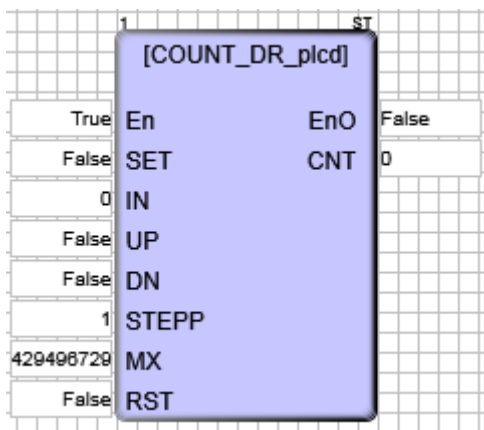
11.1.16.3. COUNT_DR (OSCAT)



Тип данных входов SET, UP, DN и RST – BOOL, входов IN, STEPP, MX и выхода CNT – DWORD.

COUNT_DR – аналог COUNT_BR (OSCAT) для DWORD.

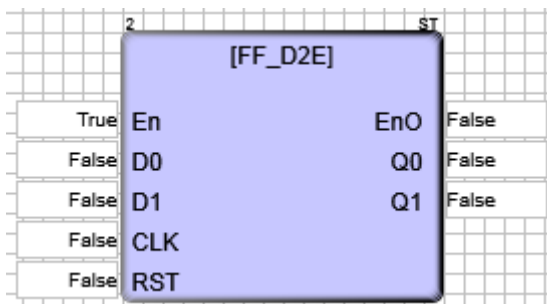
11.1.16.4. COUNT_DR_PLCD



Тип данных входов SET, UP, DN и RST – BOOL, входов IN, STEPP, MX и выхода CNT – DWORD.

COUNT_DR_plcd – аналог COUNT_BR_plcd для DWORD.

11.1.16.5. FF_D2E (OSCAT)

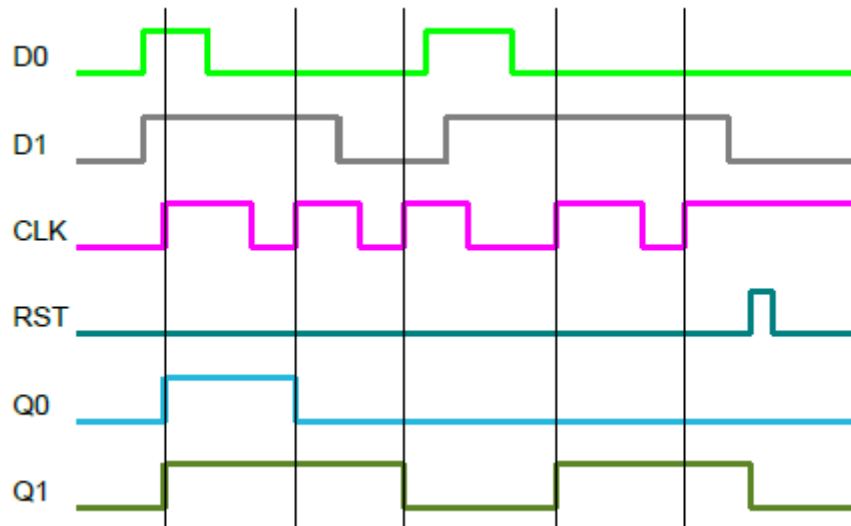


Тип данных входов и выходов – BOOL.

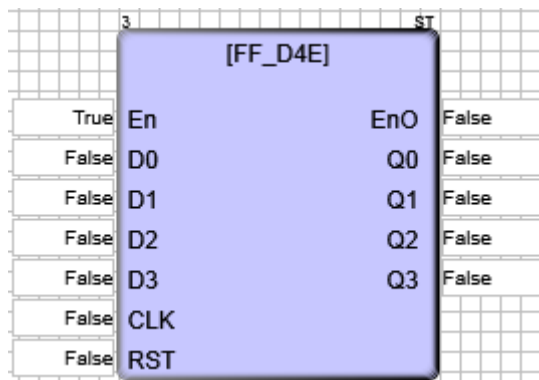
FF_D2E – это D-триггер.

При RST=FALSE и детектировании переднего фронта на входе CLK значения D0 и D1 записываются соответственно в Q0 и Q1.

При RST=TRUE Q0=Q1=0.



11.1.16.6. FF_D4E (OSCAT)



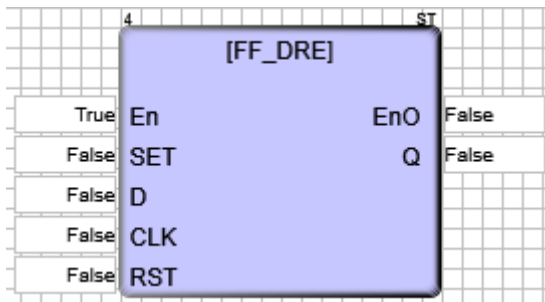
Тип данных входов и выходов – BOOL.

FF_D4E – это D-триггер.

При RST=FALSE и детектировании переднего фронта на входе CLK значения D0, D1, D2 и D3 записываются соответственно в Q0, Q1, Q2 и Q3.

При RST=TRUE Q0=Q1=Q2=Q3=0.

11.1.16.7. FF_DRE (OSCAT)



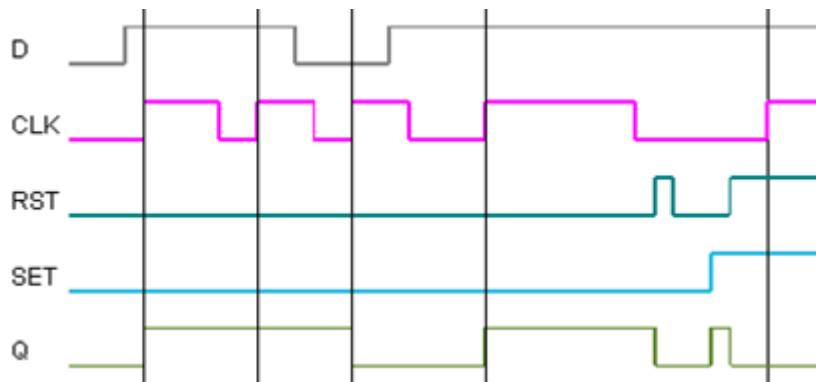
Тип данных входов и выхода – BOOL.

FF_DRE – это D-триггер.

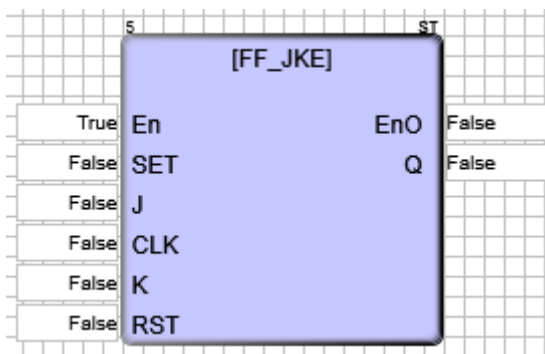
При RST=SET=FALSE и детектировании переднего фронта на входе CLK значение D записывается в Q.

При RST=FALSE и SET=TRUE Q:=TRUE (принудительная установка выхода).

При RST=TRUE Q:=FALSE (принудительный сброс выхода).



11.1.16.8. FF_JKE (OSCAT)



Тип данных входов и выхода – BOOL.

FF_JKE – это JK-триггер.

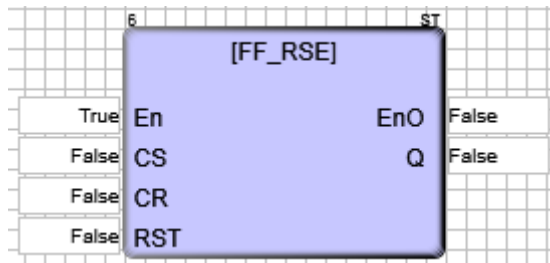
При RST=SET=FALSE, $J \neq K$ и детектировании переднего фронта на входе CLK $Q := J$.

При RST=SET=FALSE, $J = K$ и детектировании переднего фронта на входе CLK $Q := Q \text{ xor } K$.

При RST=FALSE и SET=TRUE Q:=TRUE (принудительная установка выхода).

При RST=TRUE Q:=FALSE (принудительный сброс выхода).

11.1.16.9. FF_RSE (OSCAT)



Тип данных входов и выхода – BOOL.

FF_RSE – это RS-триггер.

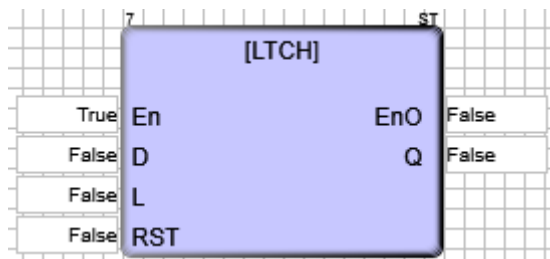
При детектировании переднего фронта на входе CS Q:=TRUE.

При детектировании переднего фронта на входе CR Q:=FALSE.

При детектировании передних фронтов одновременно на CS и CR Q:=FALSE.

При RST=TRUE Q:=FALSE.

11.1.16.10. LTCH (OSCAT)

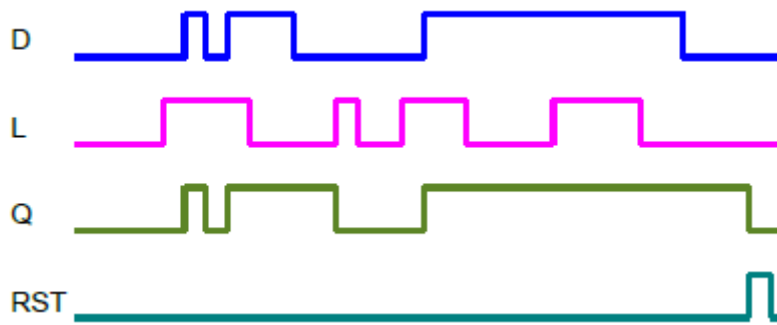


Тип данных входов и выхода – BOOL.

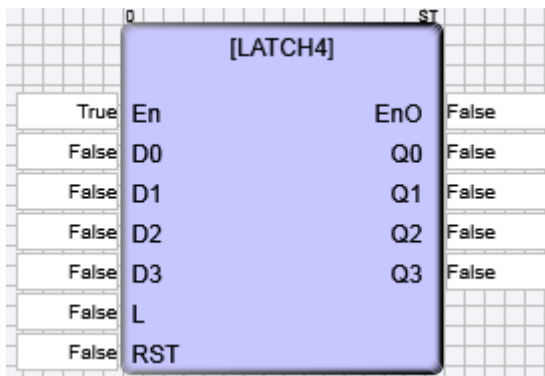
LTCH – т.н. прозрачная защелка.

До тех пор, пока L=TRUE, значение D передается в Q. При L=FALSE передача блокируется, а Q сохраняет свое значение.

Если RST=TRUE, Q:=FALSE (принудительный сброс выхода).



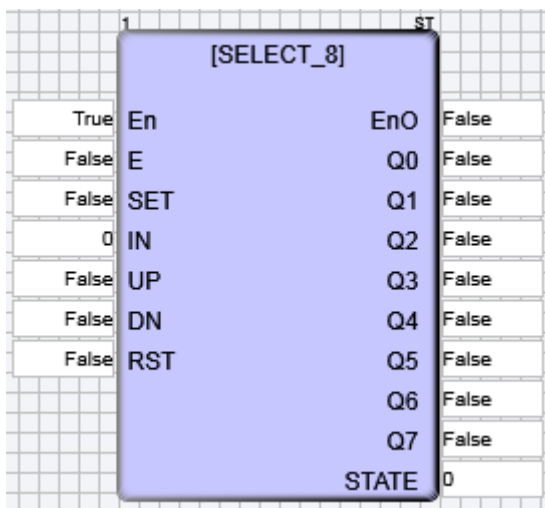
11.1.16.11. LATCH4 (OSCAT)



Тип данных входов и выхода – BOOL.

LATCH4 представляет собой 4 ФБ LTCH (OSCAT) (D0 записывается в Q0 и т.д.).

11.1.16.12. SELECT_8 (OSCAT)



Тип данных входа IN – BYTE, выхода STATE – INT, остальных входов и выходов – BOOL.

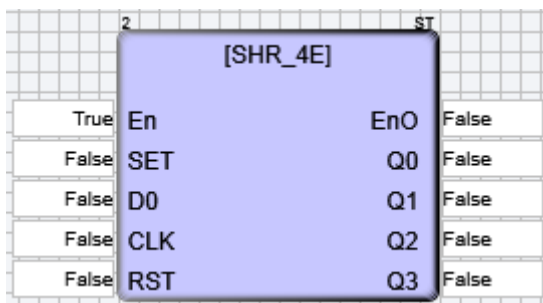
При E=TRUE SELECT_8 устанавливает выход Q с номером STATE, сбрасывая остальные выходы Q. Алгоритм:

- если $RST=SET=FALSE$, передний фронт на входе UP устанавливает следующий по номеру выход ($STATE:=STATE+1$, Q0 после Q7), а передний фронт на входе DN – предыдущий ($STATE:=STATE-1$, Q7 после Q0); увеличение STATE имеет более высокий приоритет;
- если $RST=FALSE$, а $SET=TRUE$, $STATE:=IN$ (установка заданного выхода);
- если $RST=TRUE$, $STATE:=0$ (т.е. устанавливается Q0).

При $E=FALSE$ все выходы Q обнуляются вне зависимости от значения STATE.

Используемые функции: INC (OSCAT).

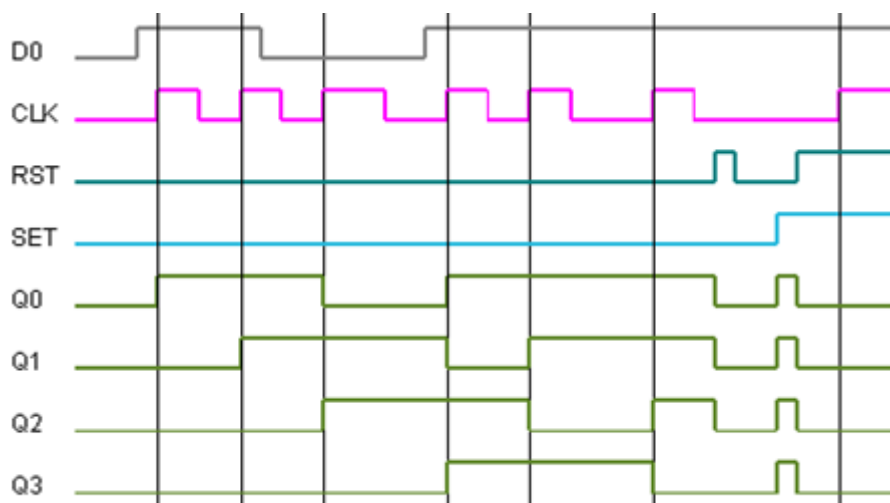
11.1.16.13. SHR_4E (OSCAT)



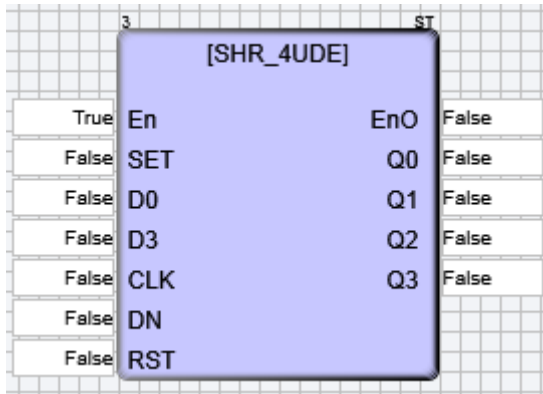
Тип данных входов и выходов – BOOL.

SHR_4E – это 4-разрядный сдвиговый регистр с асинхронной установкой и сбросом. Передний фронт на входе CLK записывает Q2 в Q3, Q1 в Q2, Q0 в Q1 и D0 в Q0.

При $RST=FALSE$ и $SET=TRUE$ все выходы (Q0.. Q3) устанавливаются, при $RST=TRUE$ – сбрасываются.



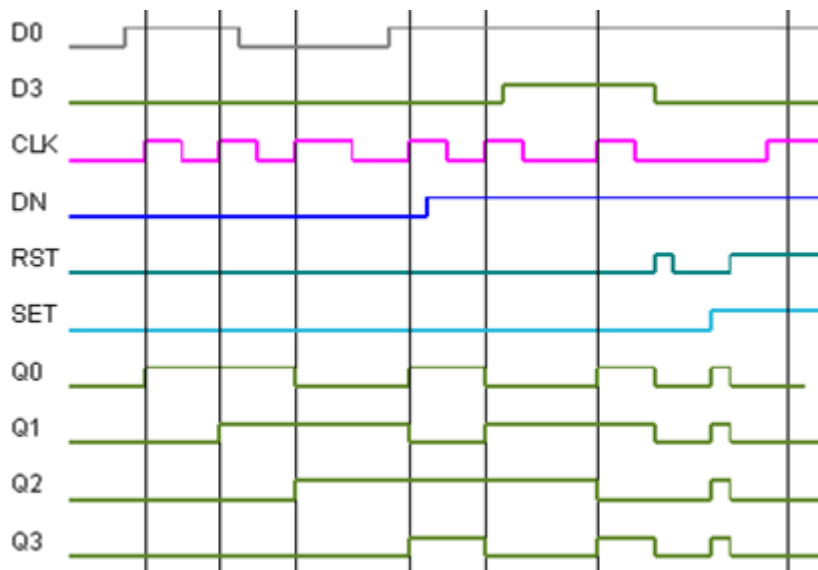
11.1.16.14. SHR_4UDE (OSCAT)



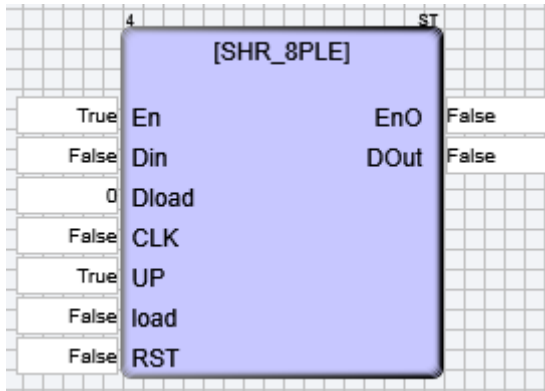
Тип данных входов и выходов – BOOL.

SHR_4UDE – это аналог SHR_4E (OSCAT) с возможностью изменения направления сдвига:

- если DN=FALSE, передний фронт на входе CLK записывает Q2 в Q3, Q1 в Q2, Q0 в Q1 и D0 в Q0 (как SHR_4E (OSCAT));
- если DN=TRUE, передний фронт на входе CLK записывает Q1 в Q0, Q2 в Q1, Q3 в Q2 и D3 в Q3.



11.1.16.15. SHR_8PLE (OSCAT)



Тип данных входа DLOAD – BYTE, остальных входов и выхода – BOOL.

SHR_8PLE – это 8-разрядный сдвиговый регистр (внутренняя переменная с типом данных BYTE) с параллельной загрузкой и асинхронным сбросом. Направление сдвига может быть изменено с помощью входа UP.

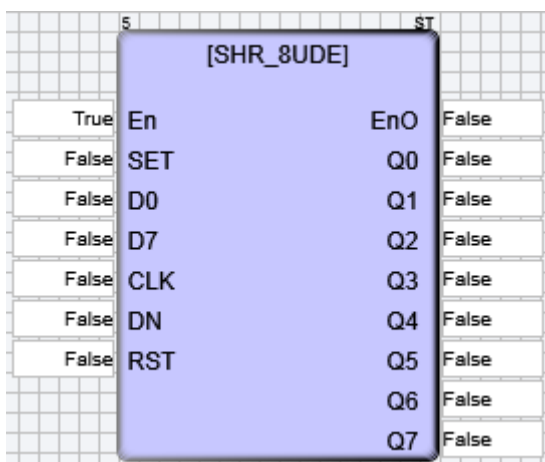
При RST=LOAD=FALSE:

- если UP=TRUE, передний фронт на входе CLK сдвигает регистр на 1 разряд влево, в освободившийся разряд 0 регистра записывается DIN, а в DOUT записывается разряд 7 регистра;
- если UP=FALSE, передний фронт на входе CLK сдвигает регистр на 1 разряд вправо, в освободившийся разряд 7 регистра записывается DIN, а в DOUT записывается разряд 0 регистра.

При RST=FALSE и LOAD=TRUE передний фронт на входе CLK сначала записывает DLOAD в регистр, а затем разряд 7 регистра (если UP=TRUE) или разряд 0 регистра (если UP=FALSE) – в DOUT (принудительная установка).

Если RST=TRUE, регистр обнуляется и DOUT:=FALSE (принудительный сброс).

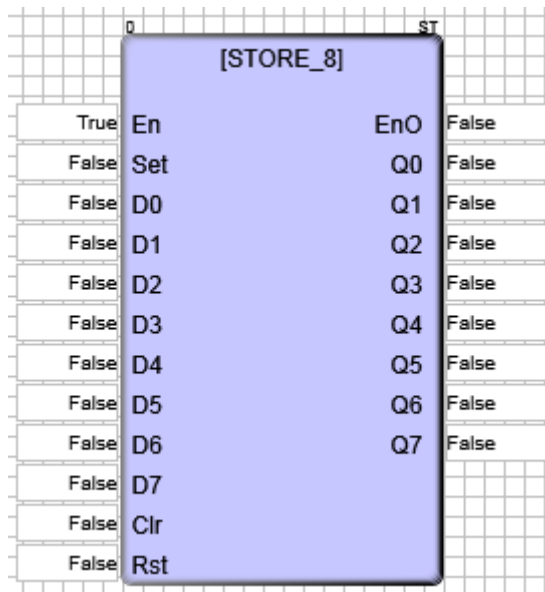
11.1.16.16. SHR_8UDE (OSCAT)



Тип данных входов и выходов – BOOL.

SHR_8UDE – это 8-разрядный аналог SHR_4UDE (OSCAT).

11.1.16.17. STORE_8 (OSCAT)



Тип данных входов и выходов – BOOL.

STORE_8 – это элемент памяти восьми событий.

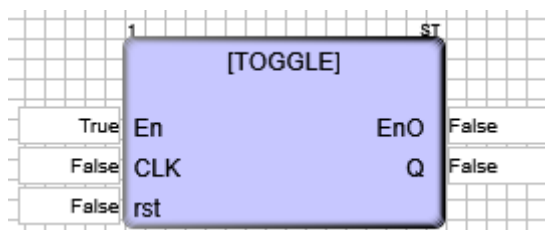
При RST=SET=FALSE:

- значение TRUE входов D0..D7 записывается соответственно в выходы Q0..Q7 (значение FALSE входа не передается на соответствующий выход);
- при D0=D1=...=D7=FALSE передний фронт на входе CLR сбрасывает младший по номеру выход Q.

При RST=FALSE и SET=TRUE Q0=Q1=...=Q7=TRUE (принудительная установка выходов).

При RST=TRUE Q0=Q1=...=Q7=FALSE (принудительный сброс выходов).

11.1.16.18. TOGGLE (OSCAT)



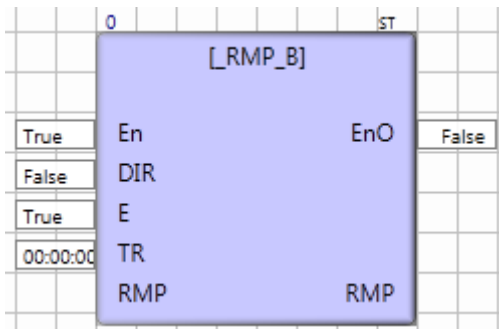
Тип данных входов и выхода – BOOL.

При RST=FALSE передний фронт на входе CLK инвертирует значение выхода Q.

При RST=TRUE Q:=FALSE (сброс).

11.1.17. OSCAT.ФБ.ГЕНЕРАТОРЫ СИГНАЛОВ

11.1.17.1. _RMP_V (OSCAT)



Тип данных входов DIR и E – BOOL, TR – TIME, входа-выхода RMP – BYTE.

_RMP_V – генератор пилообразного сигнала в диапазоне 0..255.

Вход DIR задает направление пила (DIR=TRUE/FALSE – возрастающая/убывающая пилы), вход E – разрешение/запрет работы (E=TRUE/FALSE), вход TR – время полного, от 0 до 255, изменения значения сигнала (период пила).

После достижения конечного значения пила это значение сохраняется на входе-выходе RMP.

Вход-выход RMP используется для задания начального значения (при E=FALSE).

Используемые функции: T_PLC_MS (OSCAT), FRMP_V (OSCAT).

Примеры:

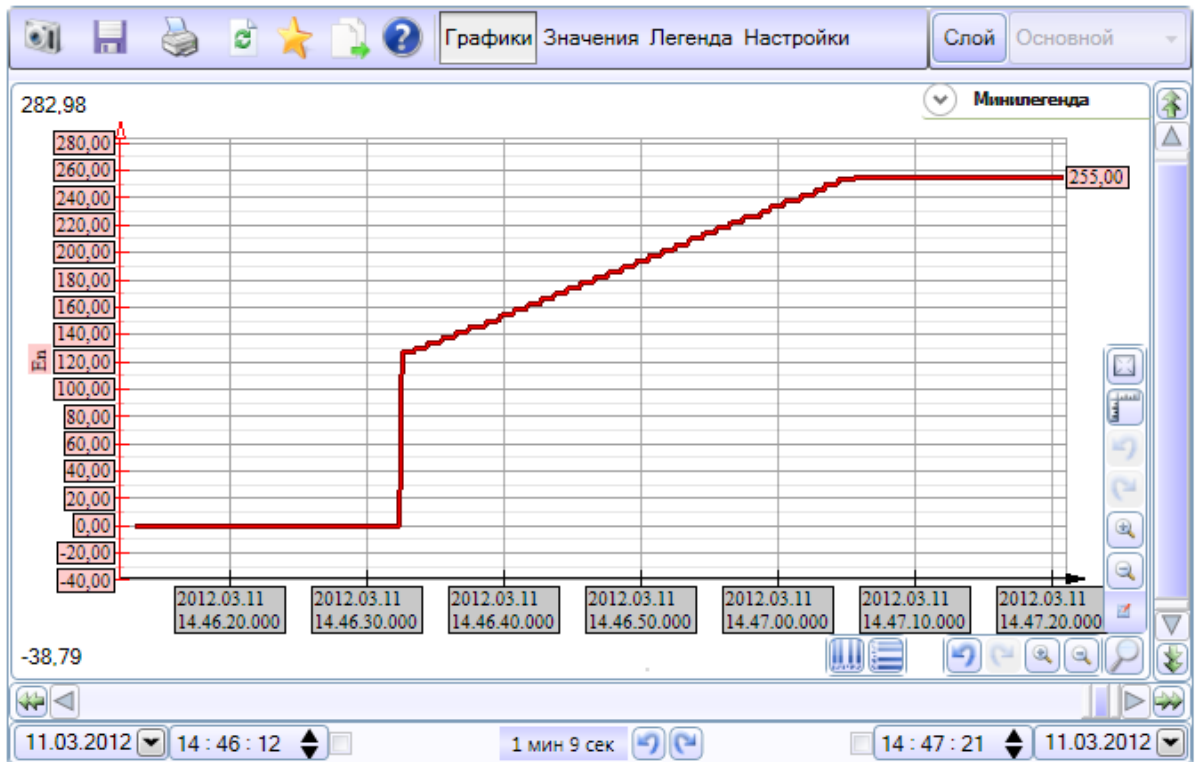
Пусть ST-программа с выходом BYTE_out типа BYTE и локальной переменной _RMP_V_1 типа _RMP_V имеет следующий код:

```
var mBYTE1:BYTE:=127; end_var;
```

```
_RMP_V_1(DIR:=true,E:=true,TR:=t#1m4s,RMP:=mBYTE1);
```

```
BYTE_out:=mBYTE1;
```

Данный код приводит к генерации на выходе BYTE_out второй половины периода возрастающей пила (от 127 до 255):



Для генерации бесконечного треугольного сигнала, начинающегося, например, со значения 127, нужно изменить код следующим образом:

```
var
```

```
GGG:=bool:=TRUE;
```

```
mBYTE1:BYTE:=127;
```

```
end_var;
```

```
if GGG=TRUE and mBYTE1=255 then
```

```
GGG:=FALSE;
```

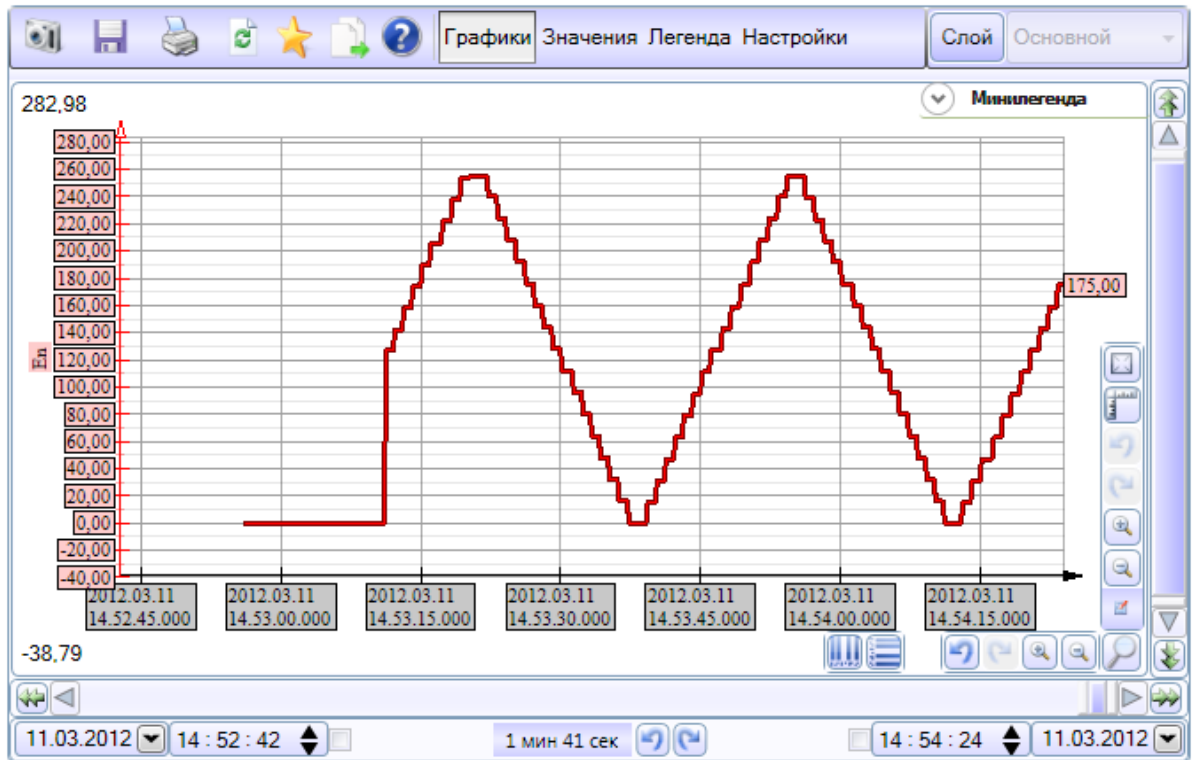
```
elsif GGG=FALSE and mBYTE1=0 then
```

```
GGG:=TRUE;
```

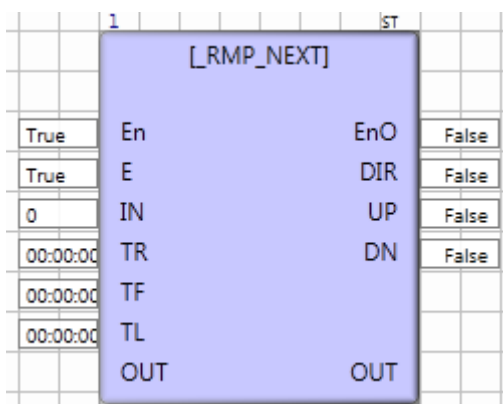
```
end_if;
```

```
_RMP_B_1(DIR:=GGG, E:=true, TR:=t#16s, RMP:=mBYTE1);
```

```
BYTE_out := mBYTE1;
```

11.1.17.2. _RMP_NEXT (OSCAT)



Тип данных входов TR, TF и TL – TIME, E – BOOL, IN – BYTE, выходов DIR, UP и DN – BOOL, входа-выхода OUT – BYTE.

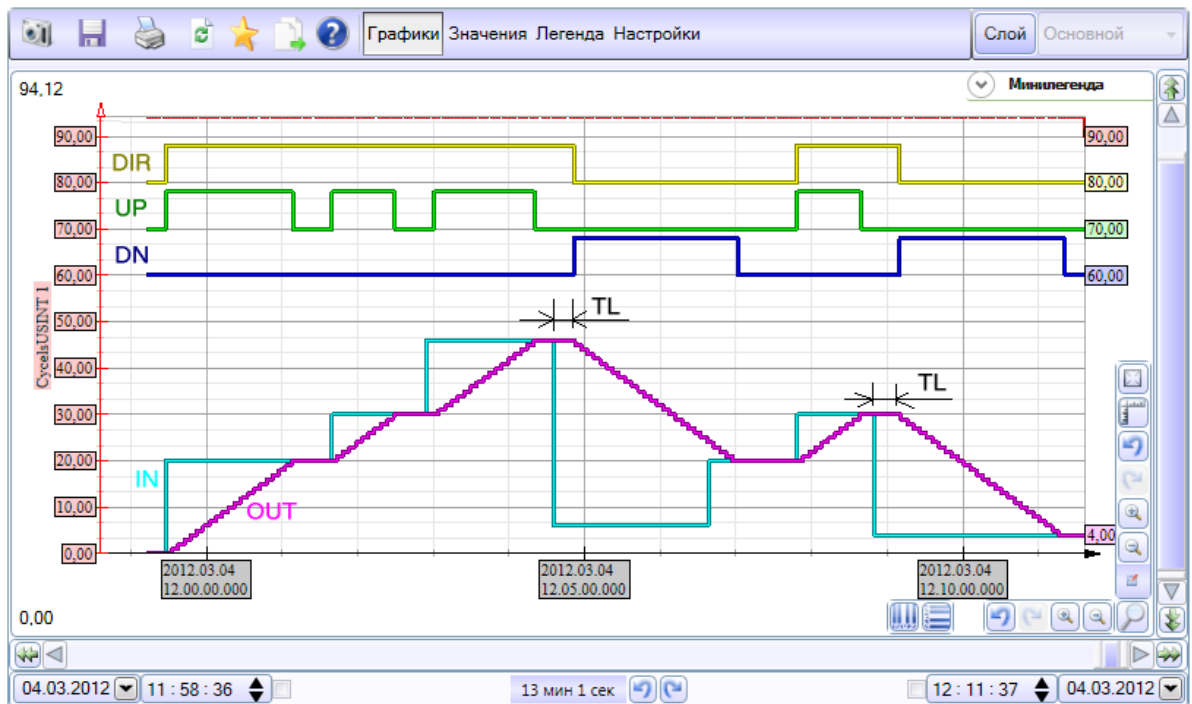
Вход E разрешает (TRUE) или приостанавливает (FALSE) исполнение алгоритма ФБ. При E=FALSE выходы ФБ сохраняют свои значения.

При скачкообразном изменении IN выход OUT изменяется плавно, по линейному закону. Скорость изменения определяется параметрами TR (время возрастания от 0 до 255) и TF (время убывания от 255 до 0). При достижении IN выход OUT сохраняет свое значение.

DIR=TRUE/FALSE при положительном/отрицательном изменении IN.

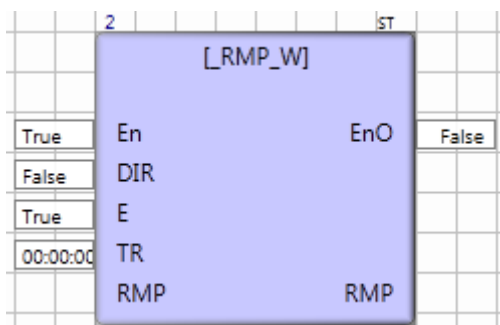
В течение возрастания/убывания OUT UP:=TRUE/FALSE и DN:=FALSE/TRUE.

При изменении направления с возрастания на убывание процедура задерживается на время TL.



Используемые функции: RMP_B (OSCAT), TREND_DW (OSCAT).

11.1.17.3. _RMP_W (OSCAT)

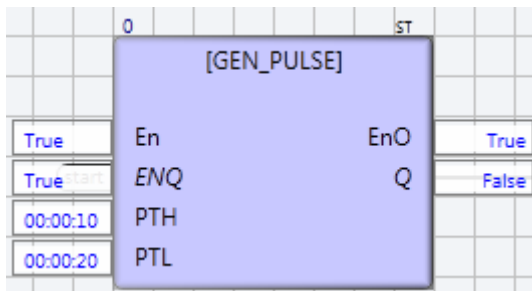


Тип данных входов DIR и E – BOOL, TR – TIME, входа-выхода RMP – WORD.

_RMP_W – это 16-битовый аналог _RMP_B (OSCAT), т.е. данный ФБ генерирует пилообразный сигнал в диапазоне 0..65535.

Используемые функции: T_PLC_MS (OSCAT).

11.1.17.4. GEN_PULSE (OSCAT)



Тип данных входов PTH и PTL – TIME, входа ENQ и выхода Q – BOOL.

На выходе Q данного ФБ генерируется прямоугольный сигнал со следующими параметрами: длительность импульса – PTH, длительность паузы между импульсами – PTL.

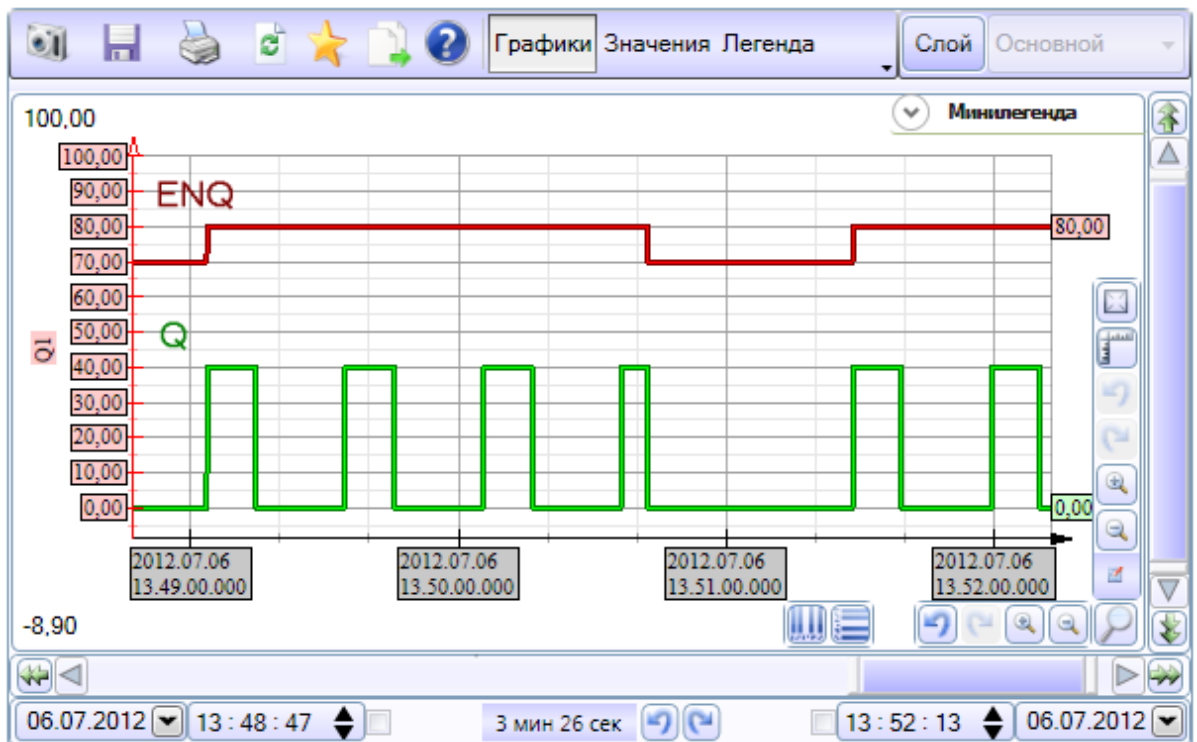
Команда запуска генератора – ENQ:=TRUE, при запуске вырабатывается первый импульс. Генерация импульсов продолжается до тех пор, пока ENQ:=TRUE.

PTH и PTL ограничиваются снизу значением цикла. Например, при следующем вызове генерируется сигнал с шириной импульса 1 цикл и паузой между импульсами 1 цикл:

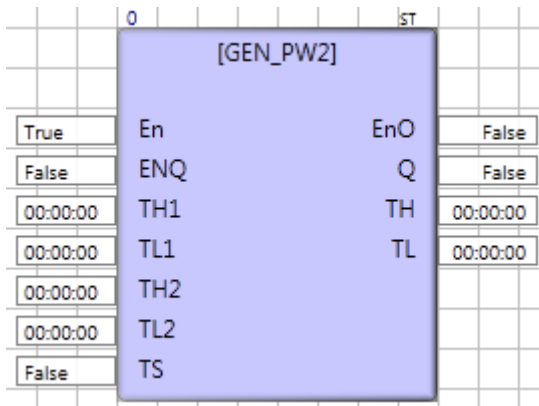
```
GEN_PULSE_1(ENQ:=TRUE, PTH:=T#0s, PTL:=T#0s);
```

Используемые функции: T_PLC_MS (OSCAT).

Следующая временная диаграмма демонстрирует работу ФБ в конфигурации, показанной на рисунке выше:



11.1.17.5. GEN_PW2 (OSCAT)



Тип данных ENQ и Q – BOOL, остальных входов и выходов – TIME.

При ENQ=TRUE ФБ генерирует на выходе Q один из двух возможных прямоугольных сигналов, которые задаются с помощью следующих параметров:

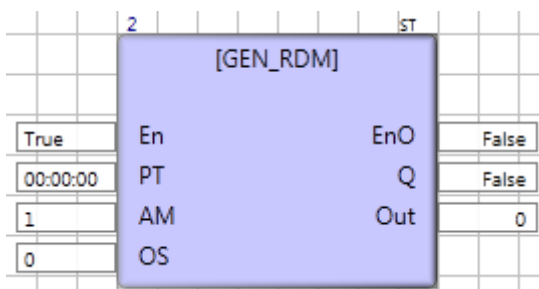
- TH1 – ширина импульса в сигнале 1;
- TL1 – пауза между импульсами в сигнале 1;
- TH2 – ширина импульса в сигнале 2;
- TL2 – пауза между импульсами в сигнале 2;

Если TS=FALSE, генерируется сигнал 1, в противном случае – сигнал 2.

Выход TH показывает время с начала текущего импульса, TL – с начала текущей паузы.

Используемые функции: T_PLC_MS (OSCAT).

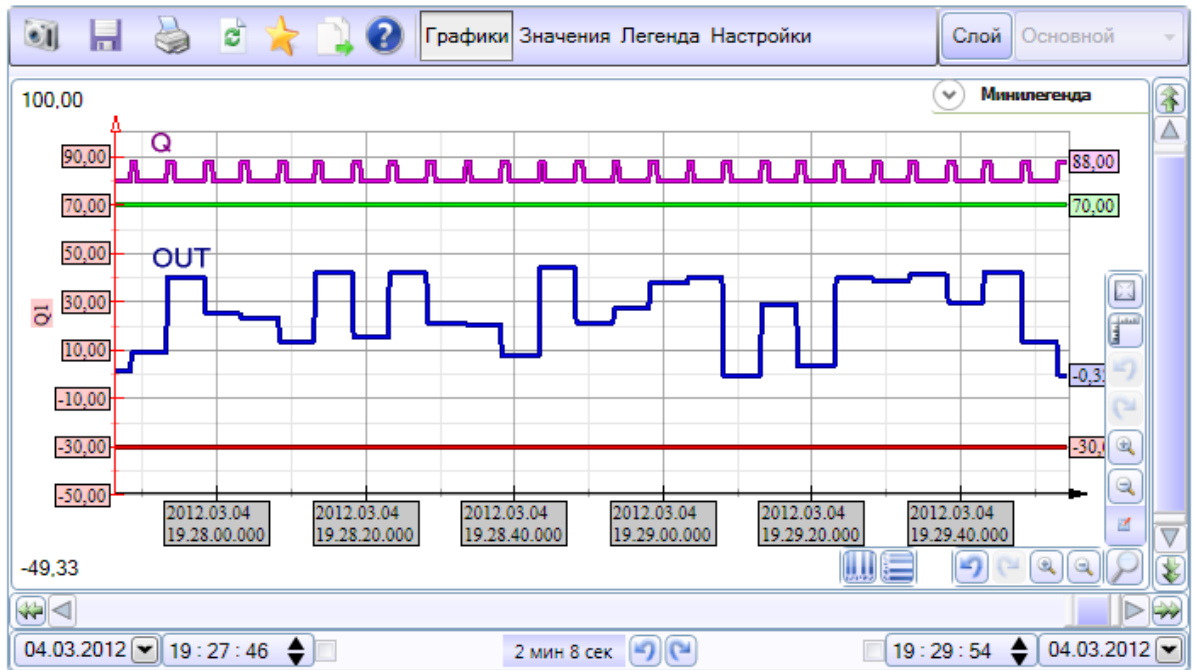
11.1.17.6. GEN_RDM (OSCAT)



Тип данных входа PT – TIME, входов AM и OS и выхода OUT – REAL, выхода Q – BOOL.

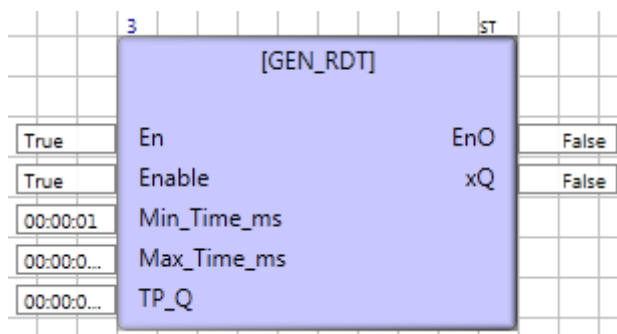
GEN_RDM генерирует на выходе OUT случайное число в интервале $(-0.5*AM+OS, 0.5*AM+OS)$ с частотой 1 раз за время PT. При изменении OUT Q=1 в течение 1 цикла.

Временная диаграмма ФБ при AM=50 и OS=20:



Используемые функции: T_PLC_MS (OSCAT), RDM (OSCAT).

11.1.17.7. GEN_RDT (OSCAT)

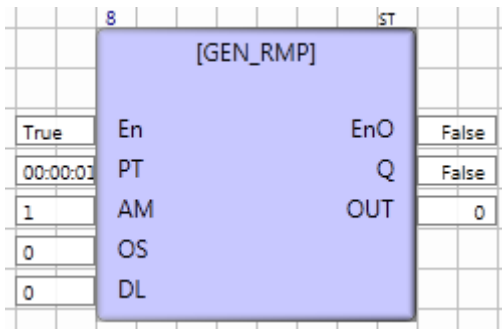


Тип данных входов MIN_TIME_MS, MAX_TIME_MS и TP_Q – TIME, входа ENABLE и выхода XQ – BOOL.

GEN_RDT генерирует прямоугольный сигнал со следующими параметрами: ширина импульсов – TP_Q, пауза между импульсами – случайное число в интервале (MIN_TIME_MS, MAX_TIME_MS).

ФБ генерирует сигнал только при ENABLE=TRUE.

11.1.17.8. GEN_RMP (OSCAT)



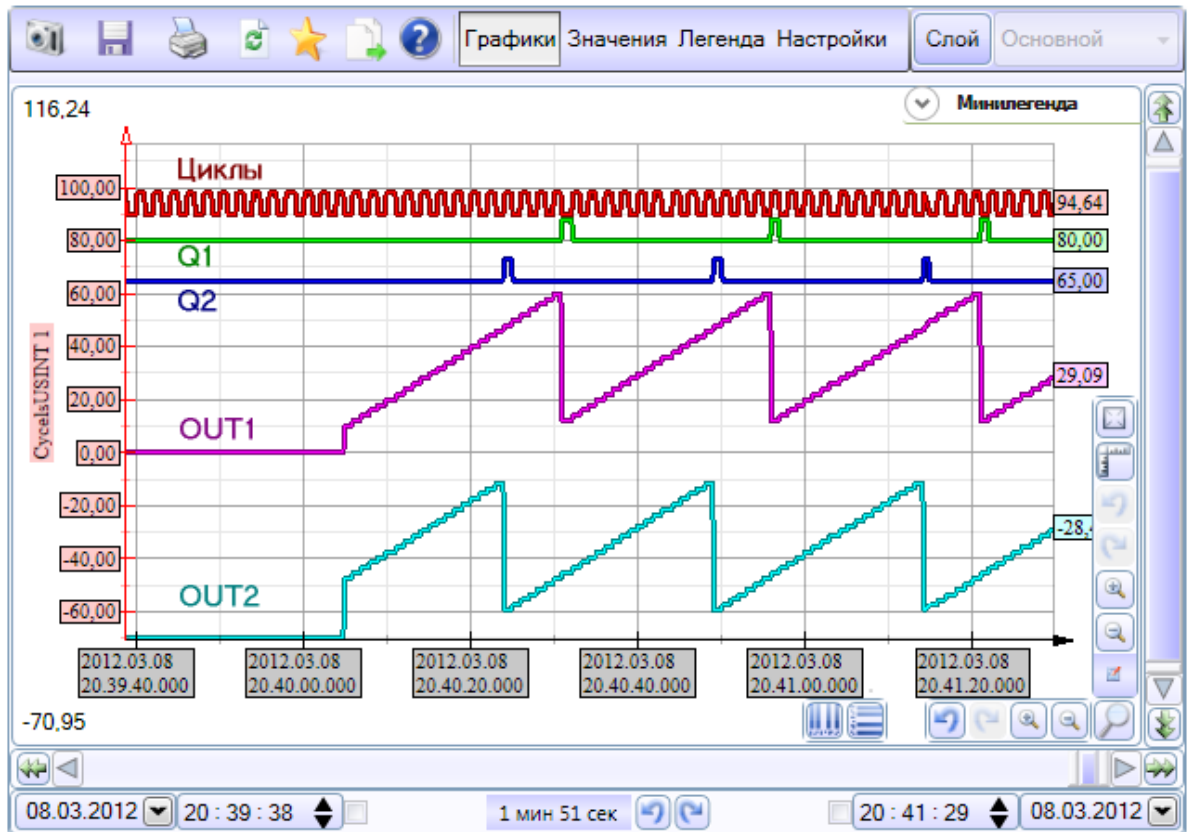
Тип данных входа PT – TIME, выхода Q – BOOL, остальных входов и выхода OUT – REAL.

GEN_RMP генерирует на выходе OUT пилообразный сигнал в диапазоне [OS, AM+OS] с периодом PT. Число шагов в периоде – $PT / \langle \text{время цикла} \rangle$, шаг по оси значений – $AM * \langle \text{время цикла} \rangle / PT$. По завершении каждого периода Q=1 на 1 цикл.

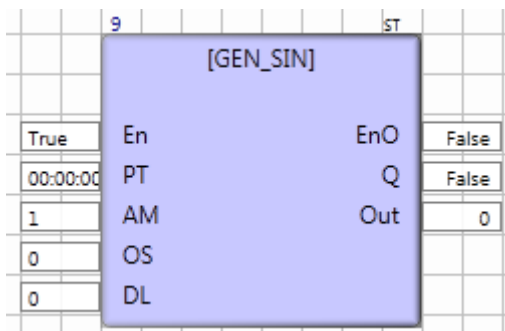
Вход DL задает смещение сигнала по фазе. Значение этого входа должно находиться в диапазоне [-1, 1], что соответствует фазовому сдвигу на величину от (-2π) (отставание на период (PT)) до 2π (опережение на период (PT)).

Используемые функции: T_PLC_MS (OSCAT), MODR (OSCAT), FRACT (OSCAT).

Ниже на рисунке показана временная диаграмма двух ФБ GEN_RMP с момента начала их работы; нижний сигнал (OUT2) опережает верхний (OUT1) по фазе на $\frac{\pi}{2}$ (т.е. на четверть периода, DL ФБ верхнего сигнала равно 0, нижнего – +0.25):



11.1.17.9. GEN_SIN (OSCAT)



Тип данных входа PT – TIME, выхода Q – BOOL, остальных входов и выхода OUT – REAL.

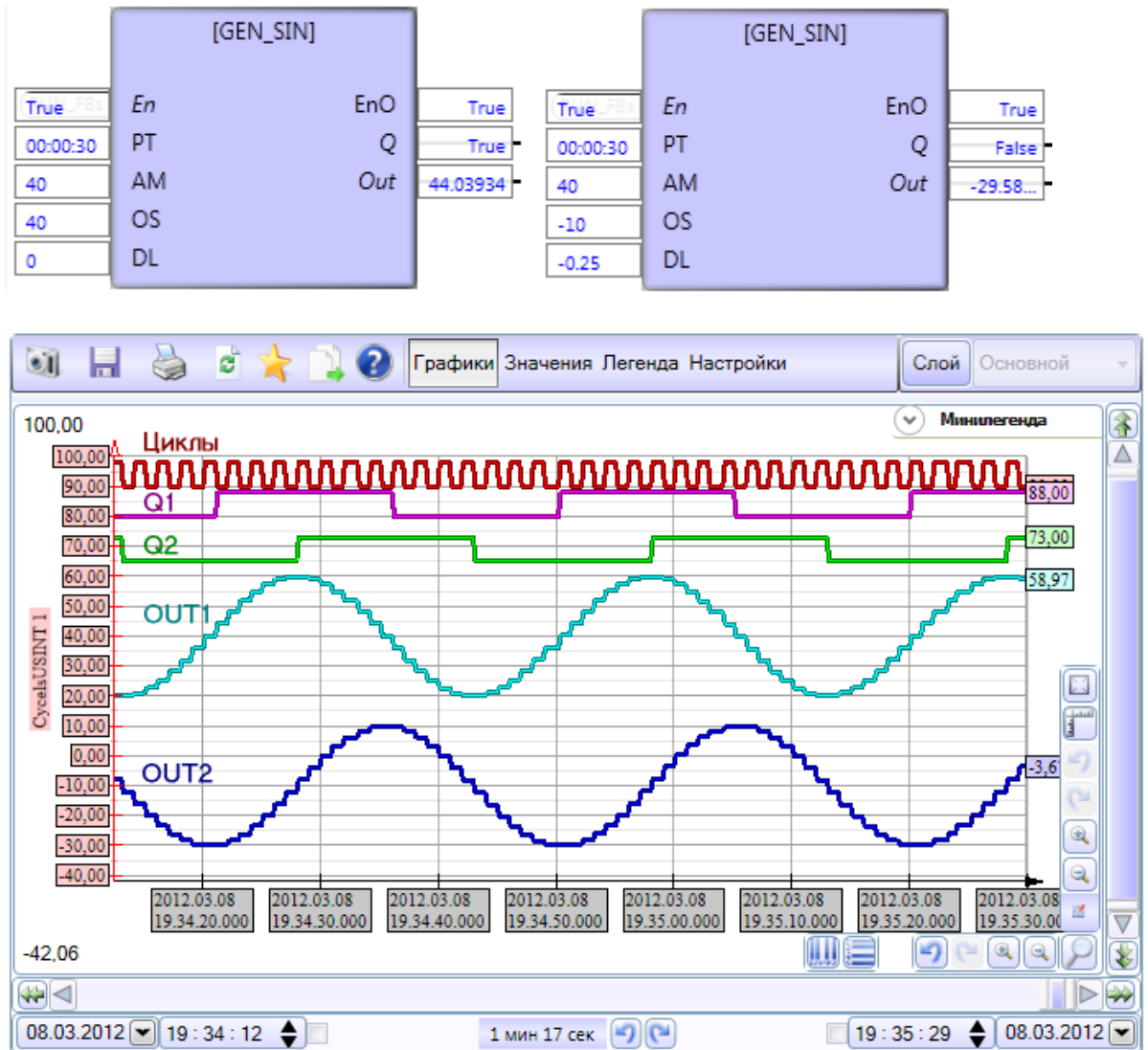
GEN_SIN генерирует на выходе OUT синусоидальный сигнал в диапазоне $[-0.5 \cdot AM + OS, 0.5 \cdot AM + OS]$ с периодом PT.

Вход DL задает смещение сигнала по фазе. Значение этого входа должно находиться в диапазоне $[-1, 1]$, что соответствует фазовому сдвигу на величину от (-2π) (отставание на период (PT)) до 2π (опережение на период (PT)).

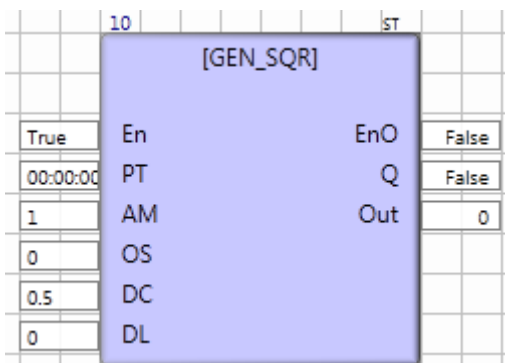
$Q:=TRUE$ в первом полупериоде генерируемого сигнала (фаза принадлежит диапазону $[\pm 2\pi, \pi \pm 2\pi]$) и $Q:=FALSE$ во втором полупериоде (фаза принадлежит диапазону $[\pi \pm 2\pi, 2\pi \pm 2\pi]$).

Используемые функции: T_PLC_MS (OSCAT), MODR (OSCAT), SIGN_R (OSCAT).

Ниже на рисунках показаны конфигурации двух ФБ GEN_SIN и их временные диаграммы (OUT2 отстает по фазе от OUT1 на $\frac{\pi}{2}$ (четверть периода)):



11.1.17.10. GEN_SQR (OSCAT)



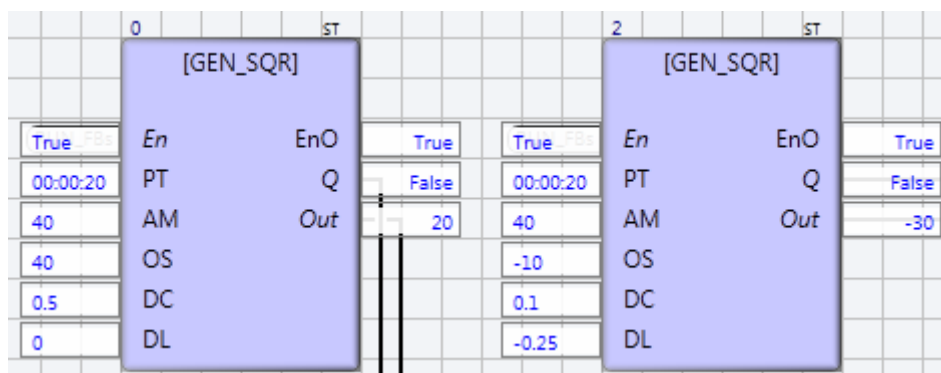
Тип данных входа PT – TIME, выхода Q – BOOL, остальных входов и выхода OUT – REAL.

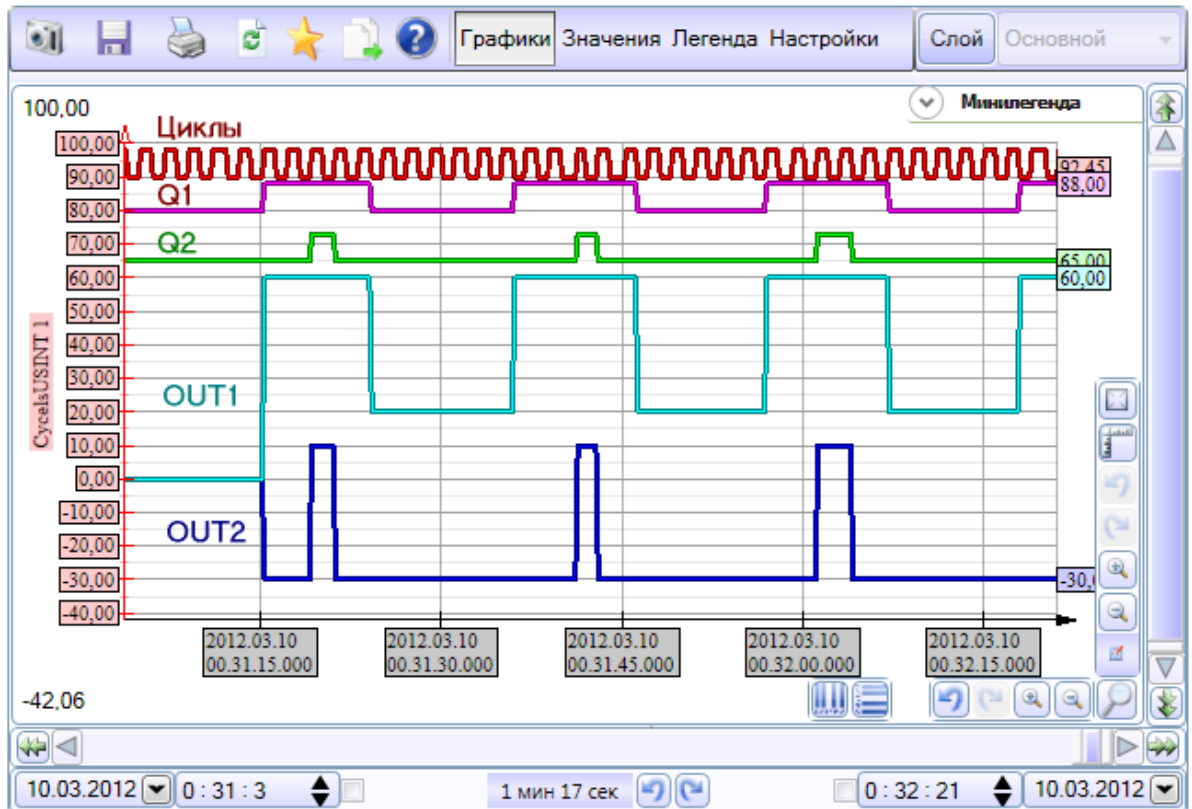
GEN_SQR генерирует на выходе OUT прямоугольный сигнал в диапазоне $[-0.5*AM + OS, 0.5*AM+OS]$, а на выходе Q – аналогичный цифровой сигнал с периодом следования импульсов PT.

Ширина импульса равна $DC*PT$, поэтому значение DC должно лежать в диапазоне $[0,1]$. Если $DC \leq 0$, OUT всегда равен $(-0.5*AM + OS)$ и Q всегда FALSE, если $DC \geq 1$, OUT всегда равен $(0.5*AM+OS)$ и Q всегда TRUE.

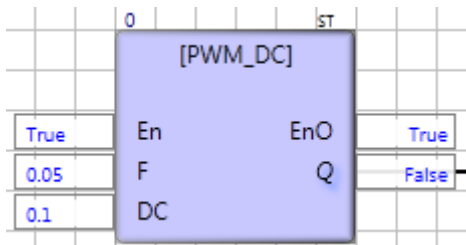
Дробная часть значения входа DL задает смещение сигнала по фазе (-1 соответствует фазовому сдвигу на -3600 (отставание на период PT), +1 – фазовому сдвигу на +3600 (опережение на период PT)).

Ниже на рисунках показаны конфигурации двух ФБ GEN_SQR и их временные диаграммы (OUT2 отстает по фазе от OUT1 на четверть периода):





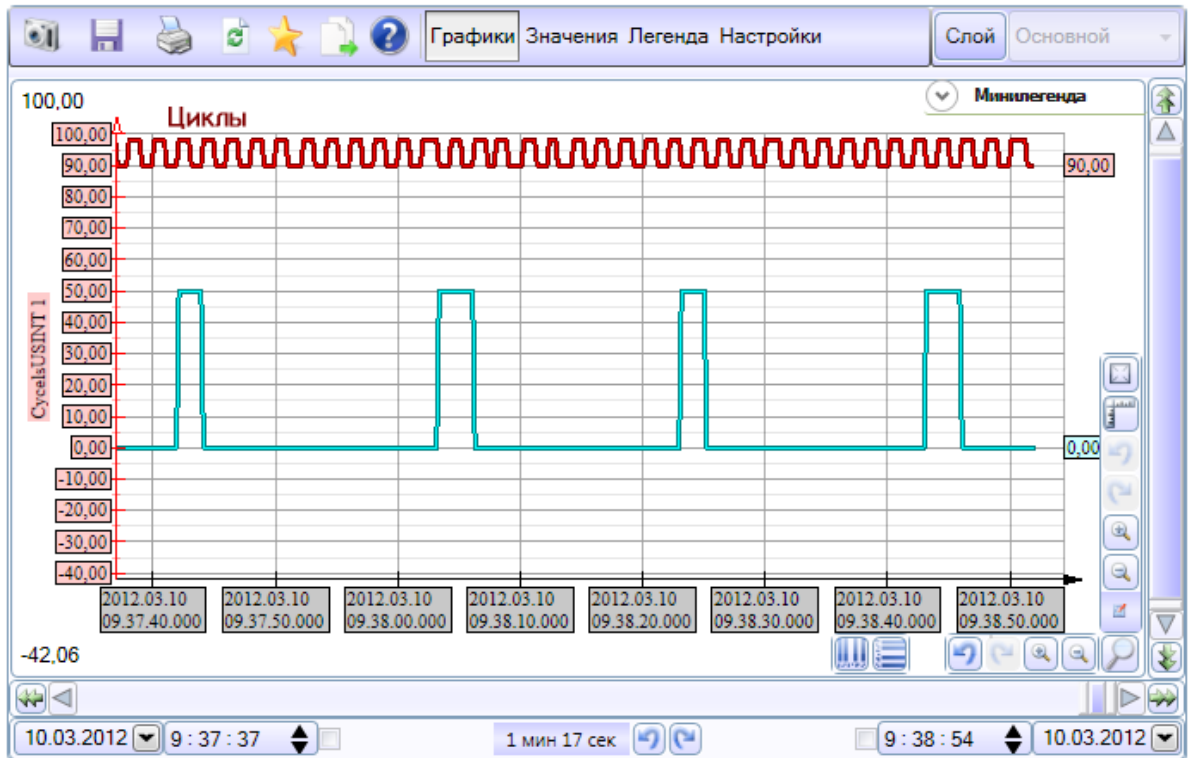
11.1.17.11. PWM_DC (OSCAT)



Тип данных входов – REAL, выхода – BOOL.

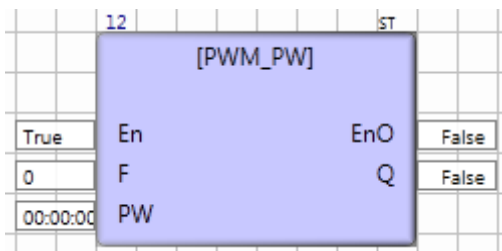
PWM_DC генерирует прямоугольный сигнал с заданной частотой F, Гц и длительностью импульса (DC/F), с. Значение DC должно лежать в диапазоне [0, 1].

Например, при показанной выше конфигурации генерируются импульсы шириной 2с и периодом 20с:



Используемые функции: CLK_PRG (OSCAT), TP_X (OSCAT).

11.1.17.12. PWM_PW (OSCAT)

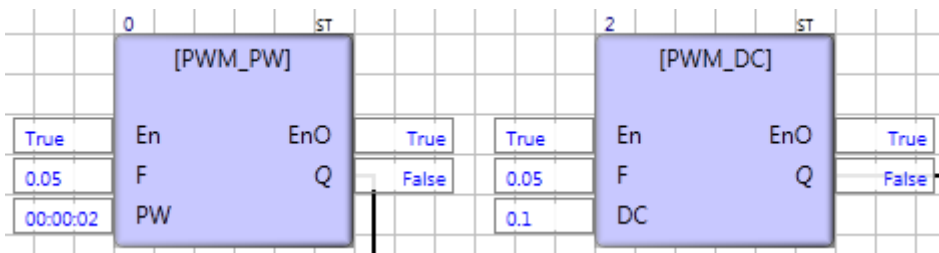


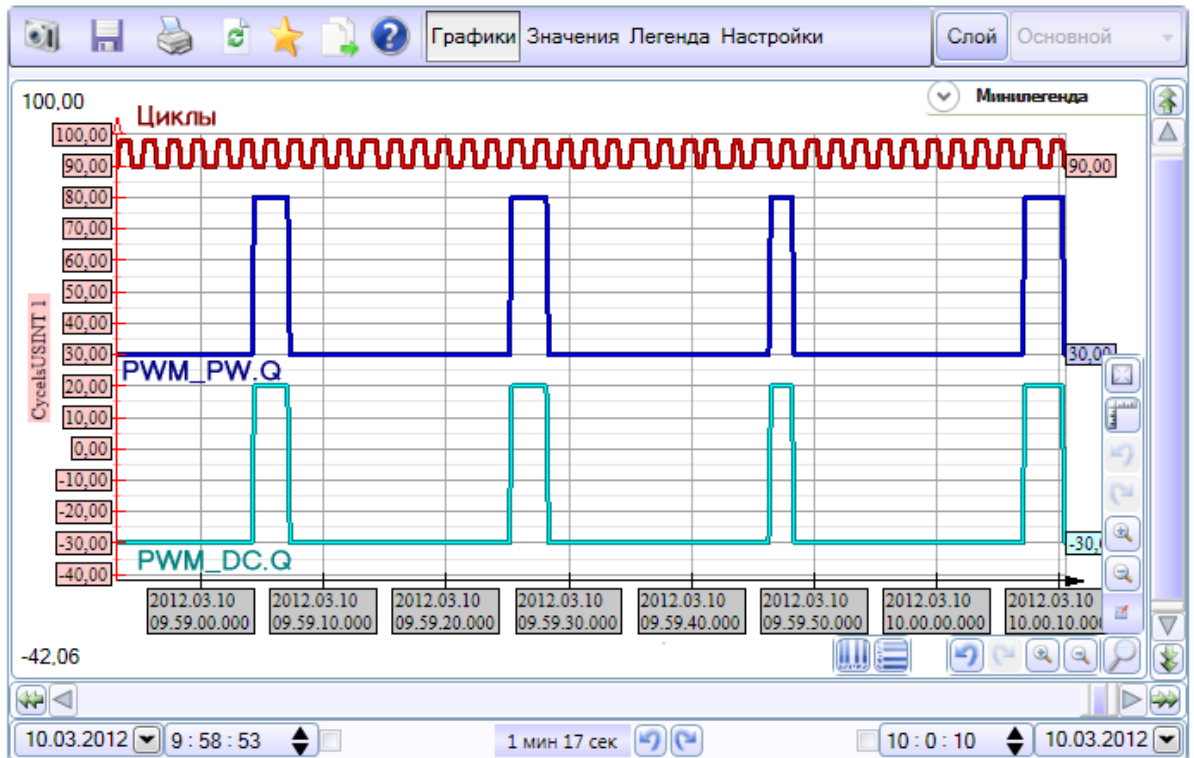
Тип данных входа F – REAL, входа PW – TIME, выхода Q – BOOL

PWM_PW – это аналог PWM_DC (OSCAT), генерирует прямоугольный сигнал с заданной частотой F, Гц и длительностью импульса PW.

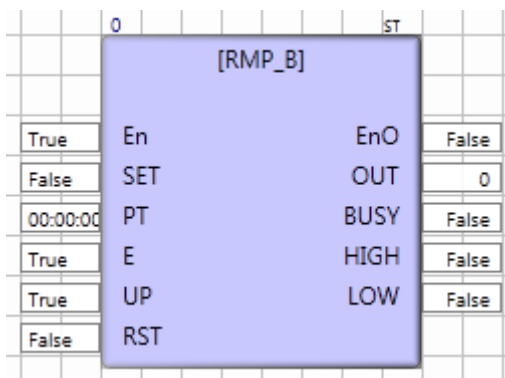
Используемые функции: CLK_PRG (OSCAT), TP_X (OSCAT).

Ниже для сравнения показаны оба ФБ, генерирующие одинаковые сигналы (ширина импульса – 2с, период – 20с):





11.1.17.13. RMP_B (OSCAT)



Тип данных входа PT – TIME, выхода OUT – BYTE, остальных входов и выходов – BOOL.

RMP_B генерирует пилообразный сигнал, используя ФБ _RMP_B (OSCAT).

Вход UP задает направление пила (UP=TRUE/FALSE – возрастающая/убывающая пилы), вход E – разрешение/запрет работы (E=TRUE/FALSE), вход PT – время полного, от 0 до 255, изменения значения сигнала (период пила).

Если SET=TRUE при RST=FALSE, OUT:=255.

Если RST:=TRUE, OUT:=0.

SET и RST используются для задания начального значения (при E=FALSE). При генерации пилы (при E=TRUE) SET и RST только маскируют (скрывают) генерируемое значение OUT.

Пока $OUT:=0$, $LOW:=TRUE$, в остальных случаях $LOW:=FALSE$.

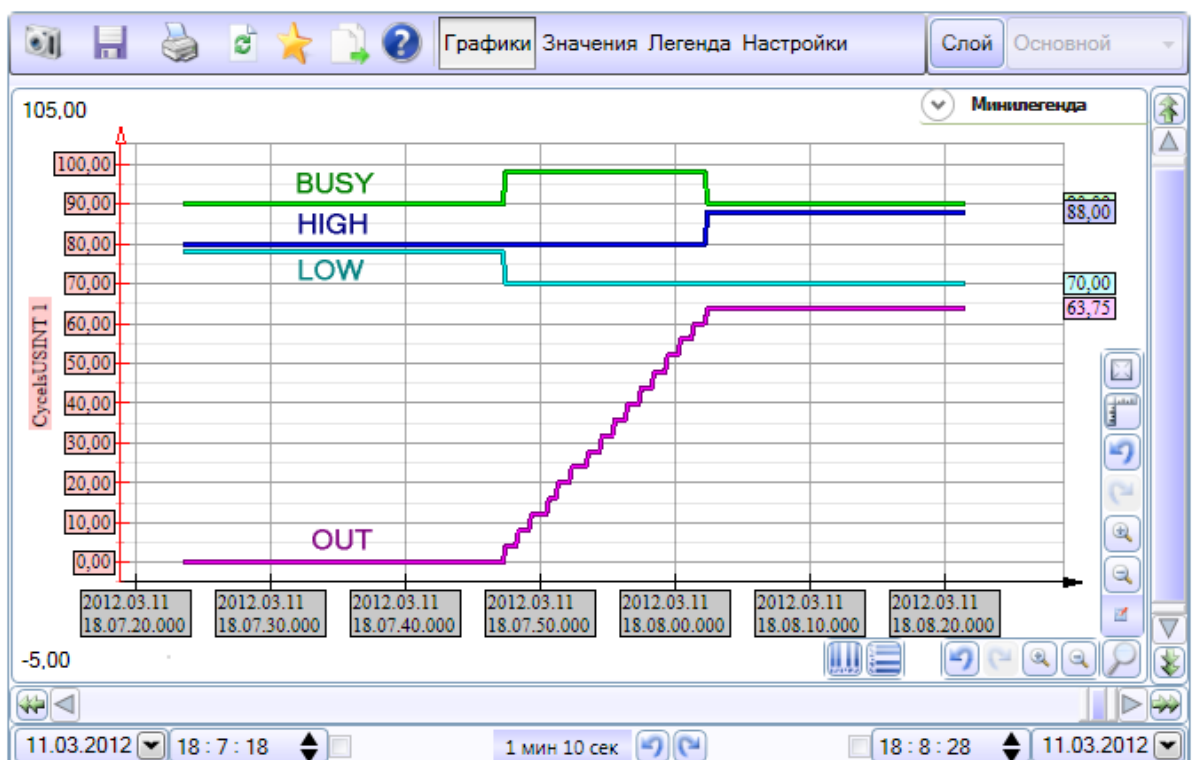
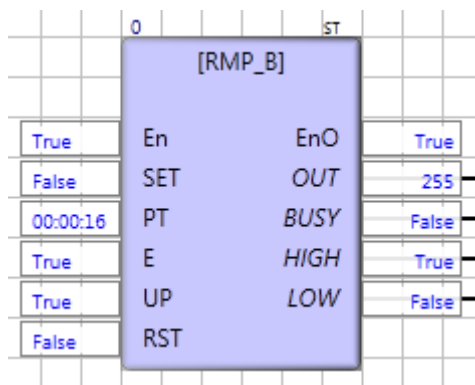
Пока $OUT:=255$, $HIGH:=TRUE$, в остальных случаях $HIGH:=FALSE$.

$BUSY:=TRUE$ при генерации пилы, т.е. если $E:=TRUE$ и $0<OUT<255$.

Число шагов в периоде – $PT/<время\ цикла>$, шаг по оси значений – $255*\langle время\ цикла\rangle/PT$.

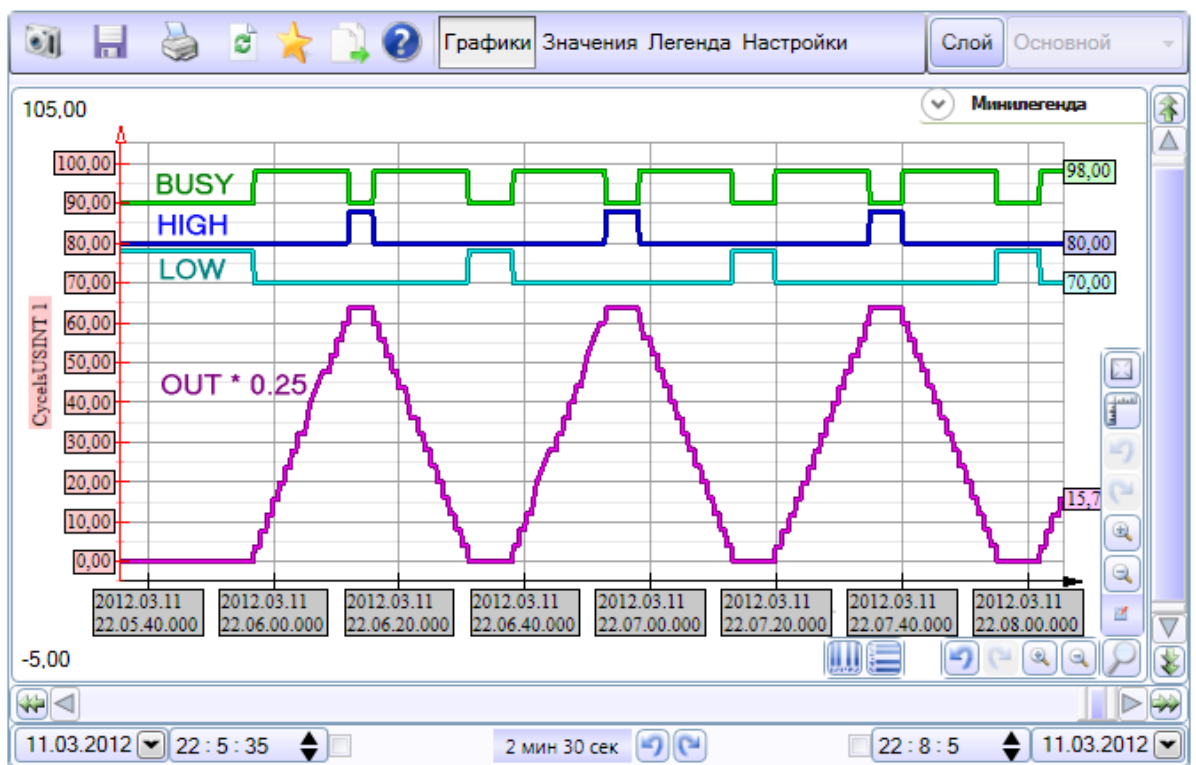
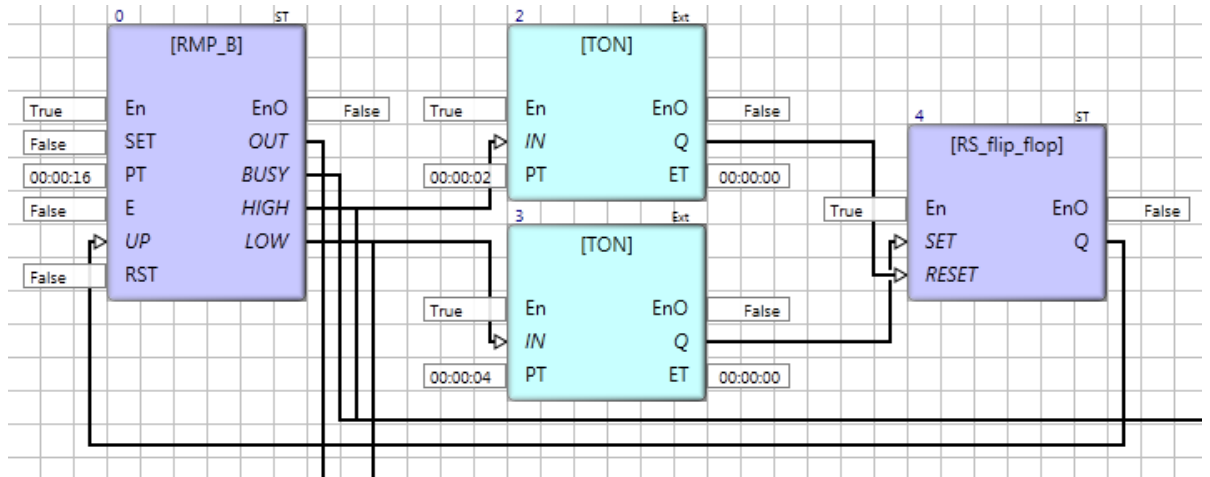
Пример 1

Следующая программа при неизменной конфигурации RMP_B генерирует один период возрастающей пилы:



Пример 2

Следующая программа после RMP_V.E=TRUE генерирует бесконечный треугольный сигнал с заданными задержками в моменты переключения направления пилы (с возрастания на убывание – 2с, с убывания на возрастание – 4с, задержки задаются с помощью двух ФБ TON, задержка включения):

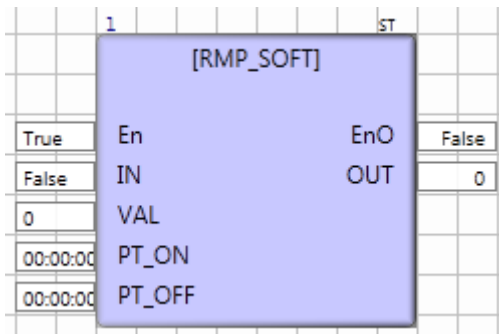


Код пользовательского ФБ RS_flip_flop:

```
if SET and NOT RESET then Q:=TRUE; end_if;
```

```
if RESET then Q:=FALSE; end_if;
```

11.1.17.14. RMP_SOFT (OSCAT)



Тип данных входов PT_ON и PT_OFF – TIME, входа VAL и выхода OUT – BYTE, входа IN – BOOL.

При IN=TRUE RMP_SOFT линейно сглаживает фронты прямоугольного сигнала, поданного на вход VAL:

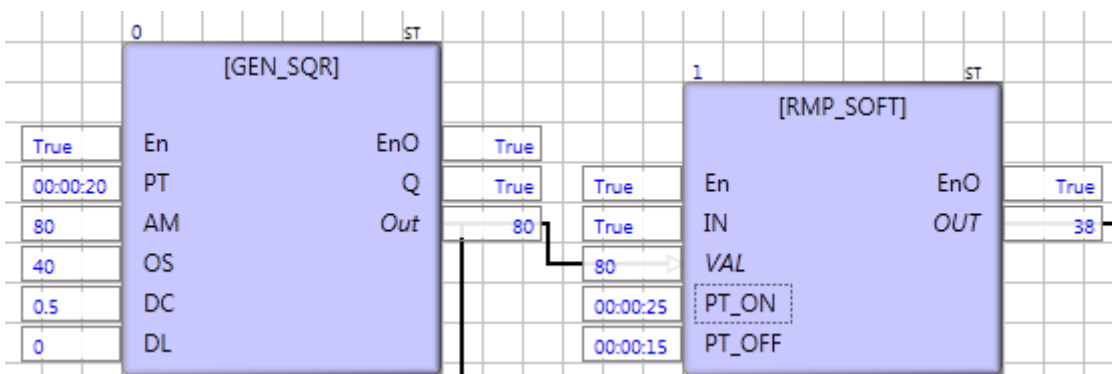
- сглаживание переднего фронта: PT_ON задает время нарастания сигнала от 0 до 255; если VAL<255, время нарастания сигнала на выходе OUT будет соответственно меньше;
- сглаживание заднего фронта: PT_OFF задает время убывания сигнала от 255 до 0; если VAL<255, время убывания сигнала на выходе OUT будет соответственно меньше.

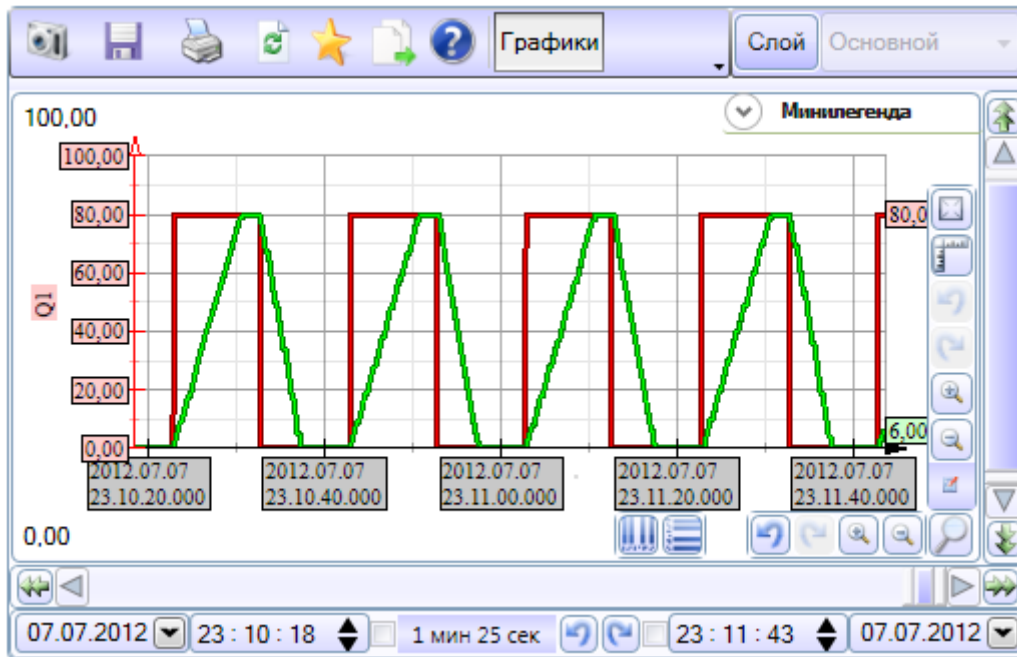
При IN=FALSE OUT=0.

Используемые функции: _RMP_B (OSCAT).

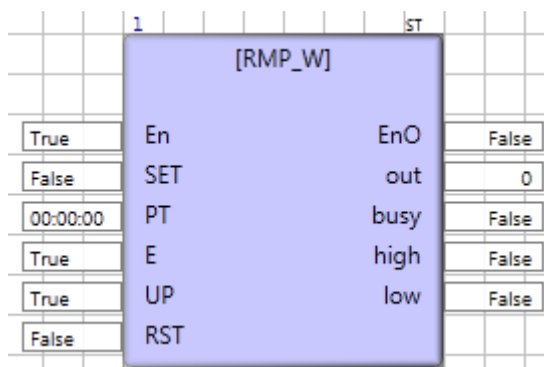
Пример

В качестве генератора прямоугольного сигнала в данном примере используется ФБ GEN_SQR (OSCAT).





11.1.17.15. RMP_W (OSCAT)



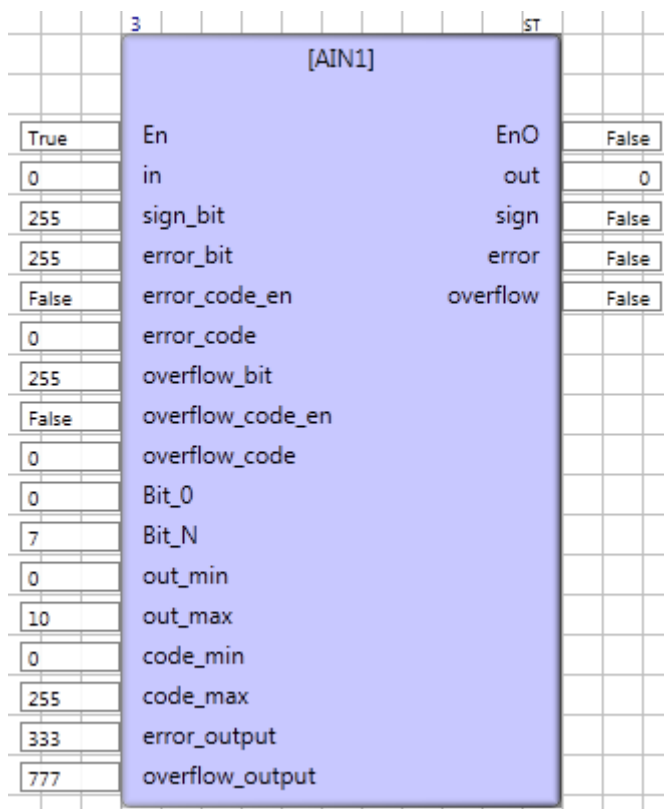
RMP_W – это 16-битовый аналог RMP_B (OSCAT), т.е. данный ФБ генерирует пилообразный сигнал в диапазоне 0..65535.

Используемые функции: _RMP_W (OSCAT).

Заметим, что в случае необходимости возрастания/убывания значения пилы с шагом 1 при длительности цикла 5мс в качестве PT потребуется задать величину $65536 * 5 = 327$ секунд. При меньшем значении PT шаг изменения значения пилы пропорционально возрастет.

11.1.18. OSCAT.ФБ.ОБРАБОТКА СИГНАЛОВ

11.1.18.1. AIN1 (OSCAT)



AIN1 преобразует дискретное значение выхода АЦП в значение REAL, соответствующее измеряемой величине (поданной на вход АЦП). С помощью своих настроек ФБ может быть сконфигурирован для работы с широким кругом АЦП, в том числе для индикации ошибок.

Дискретное значение выхода АЦП подается на вход IN ФБ AIN1 (тип данных IN – DWORD).

Вход SIGN_BIT (тип данных INT) должен быть равен номеру бита, в котором АЦП передает знак. SIGN_BIT=255 (значение по умолчанию) означает, что АЦП не передает знаковый бит. Наличие/отсутствие знакового бита индицируется на BOOL-выходе SIGN – соответственно TRUE/FALSE.

INT-входы BIT_0 и BIT_N определяют соответственно позиции LSB и MSB во входных данных (LSB – начало данных, MSB – конец данных).

Значения IN масштабируются – диапазон [CODE_MIN, CODE_MAX] переводится в диапазон [OUT_MIN, OUT_MAX] (тип данных CODE_MIN и CODE_MAX – DWORD, OUT_MIN и OUT_MAX – REAL). Если знаковый бит используется (SIGN=TRUE), значение OUT инвертируется.

Обработка ошибок

Поскольку АЦП могут сигнализировать о наличии ошибки с помощью передачи флага ошибки в некотором бите или с помощью передачи некоторого фиксированного значения (вне рабочего диапазона), ФБ AIN1 поддерживает оба эти механизма как для

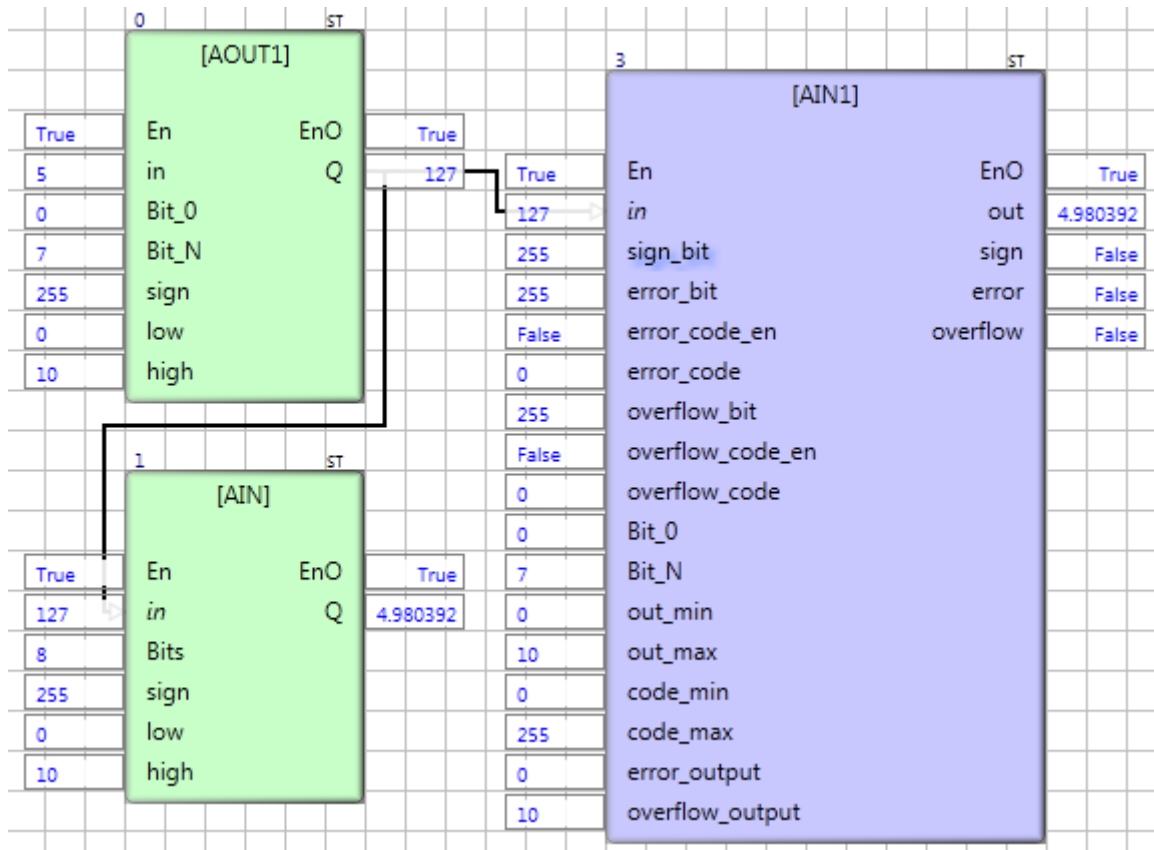
общей ошибки АЦП, так и для ошибки переполнения АЦП. По умолчанию, ошибки не обрабатываются (значения `ERROR_BIT=255`, `ERROR_CODE_EN=FALSE`, `OVERFLOW_BIT=255` и `OVERFLOW_CODE_EN=FALSE` блокируют все 4 механизма обработки ошибок).

Обработка общей ошибки

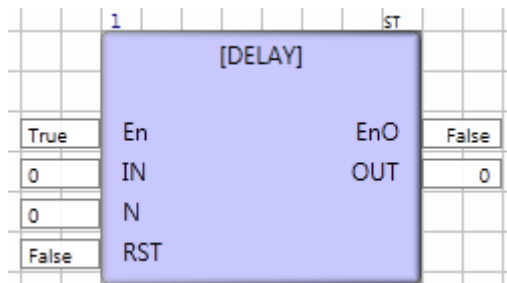
- Для обработки общей ошибки по флагу INT-вход `ERROR_BIT` должен быть равен номеру бита, в котором АЦП передает флаг ошибки. Кроме того, нужно задать необходимое значение REAL-входу `ERROR_OUTPUT`. Если значение IN содержит установленный флаг ошибки, BOOL-выход `ERROR` принимает значение `TRUE`, а выход `OUT` – значение `ERROR_OUTPUT`.
- Для обработки общей ошибки по фиксированному значению нужно присвоить это значение DWORD-входу `ERROR_CODE`, разрешить обработку (присвоить BOOL-входу `ERROR_CODE_EN` значение `TRUE`) и задать необходимое значение REAL-входу `ERROR_OUTPUT`. Если `IN=ERROR_CODE`, то `ERROR=TRUE` и `OUT=ERROR_OUTPUT`.
- Ошибка переполнения обрабатывается аналогично:
- Для обработки ошибки переполнения по флагу INT-вход `OVERFLOW_BIT` должен быть равен номеру бита, в котором АЦП передает флаг переполнения. Кроме того, нужно задать необходимое значение REAL-входу `OVERFLOW_OUTPUT`. Если значение IN содержит установленный флаг переполнения, BOOL-выход `OVERFLOW` принимает значение `TRUE`, а выход `OUT` – значение `OVERFLOW_OUTPUT` (если `ERROR=FALSE`; в противном случае `OUT=ERROR_OUTPUT`).
- Для обработки ошибки переполнения по фиксированному значению нужно присвоить это значение DWORD-входу `OVERFLOW_CODE`, разрешить обработку (присвоить BOOL-входу `OVERFLOW_CODE_EN` значение `TRUE`) и задать необходимое значение REAL-входу `OVERFLOW_OUTPUT`. Если `IN=OVERFLOW_CODE`, то `OVERFLOW=TRUE` и `OUT=OVERFLOW_OUTPUT` (если `ERROR=FALSE`; в противном случае `OUT=ERROR_OUTPUT`).

Пример

В данном примере показано конфигурирование ФБ `AIN1` для 8-разрядного АЦП без передачи знакового бита и без обработки ошибок. В качестве модели АЦП используется функция `AOUT1 (OSCAT)`, функция `AIN (OSCAT)` показана для сравнения с ФБ `AIN1`.



11.1.18.2. DELAY (OSCAT)



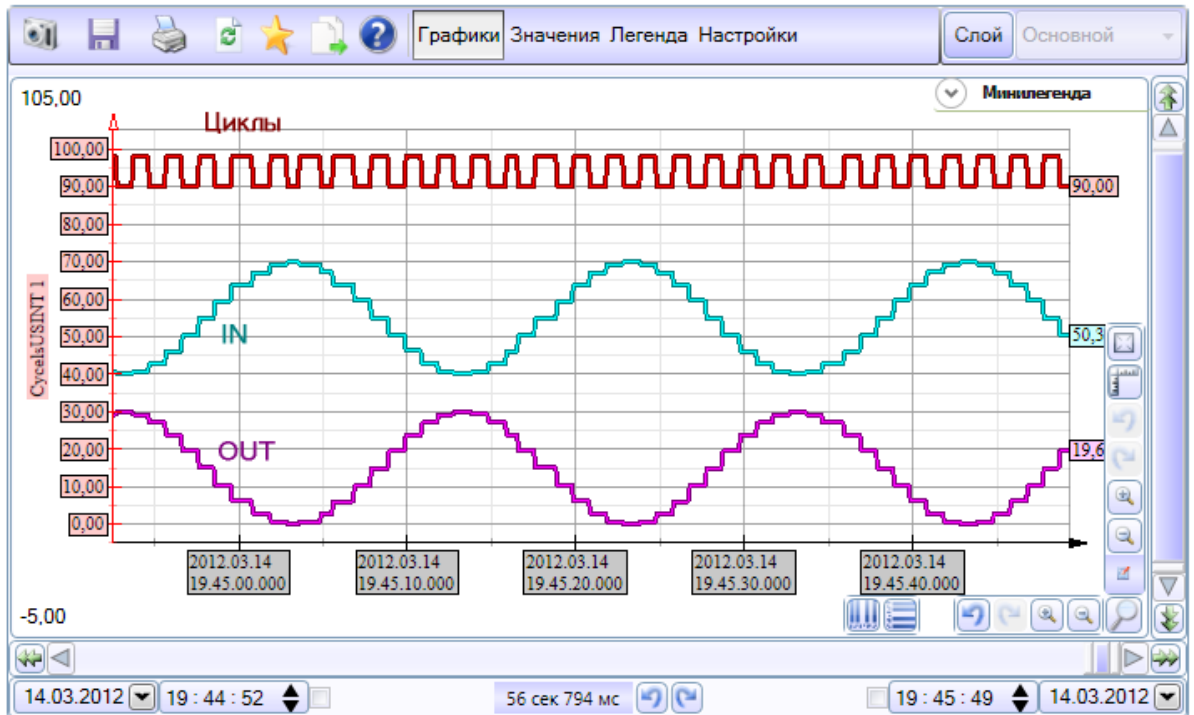
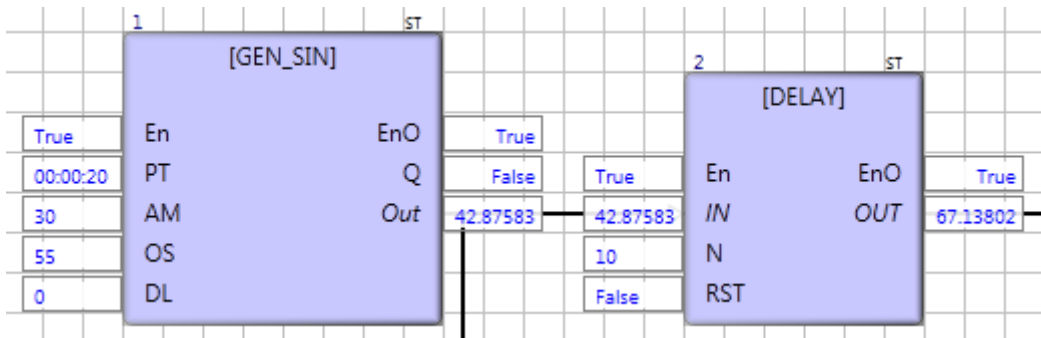
ФБ DELAY задерживает сигнал, поданный на REAL-вход IN, на N циклов (тип данных N – INT). Задержанный сигнал передается на REAL-выход OUT.

Для сброса буфера ФБ входу RST нужно присвоить значение TRUE (если RST=TRUE, OUT=IN).

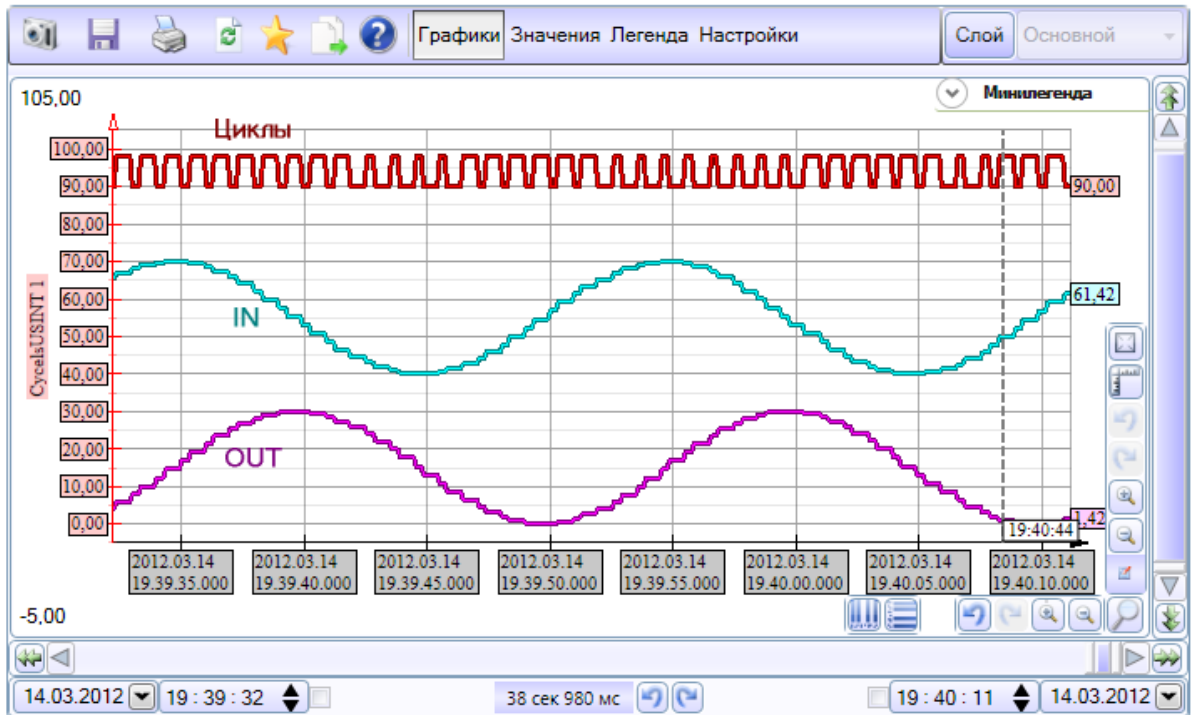
Используемые функции: INC1 (OSCAT).

Пример

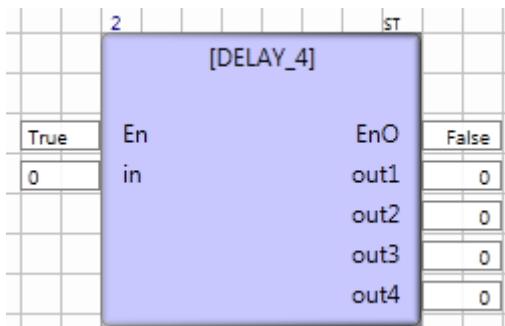
В данном примере длительность цикла – 1с, задержка N=10 циклов, поэтому синусоида с периодом 20с задерживается на половину периода (10*1=10с). Для генерации синусоиды используется ФБ GEN_SIN (OSCAT).



В той же программе при времени цикла 0.5с синусоида задерживается на четверть периода ($10 \cdot 0.5 = 5\text{с}$):



11.1.18.3. DELAY_4 (OSCAT)

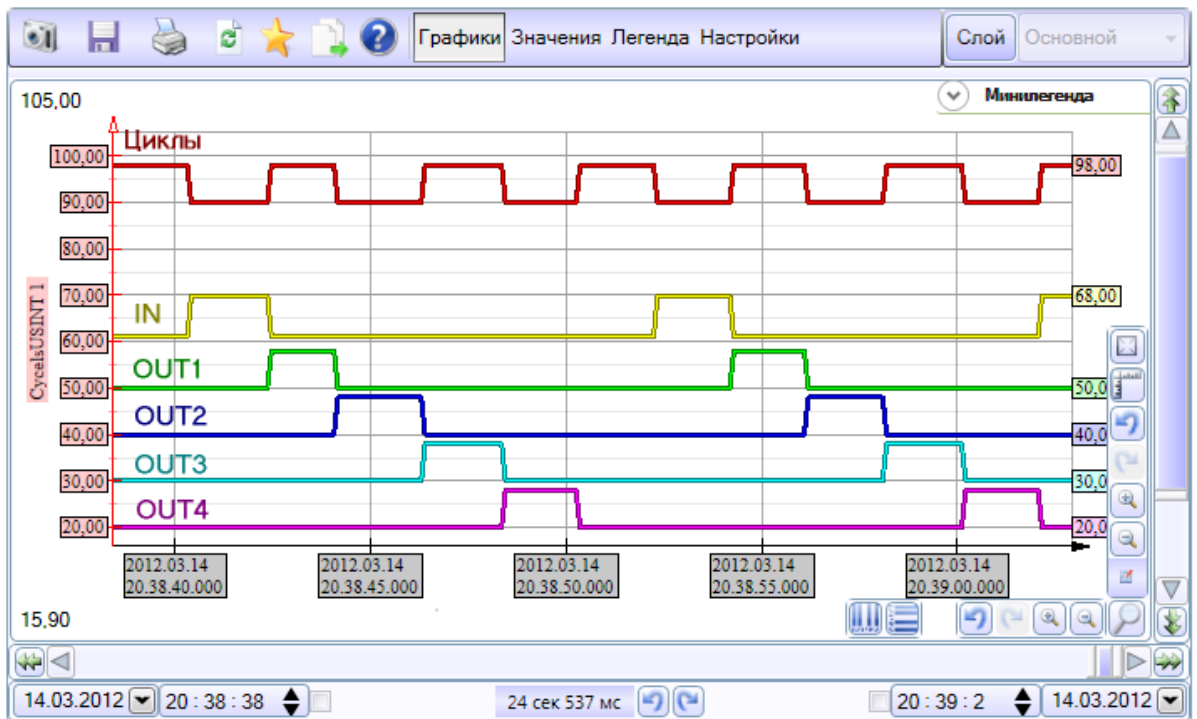
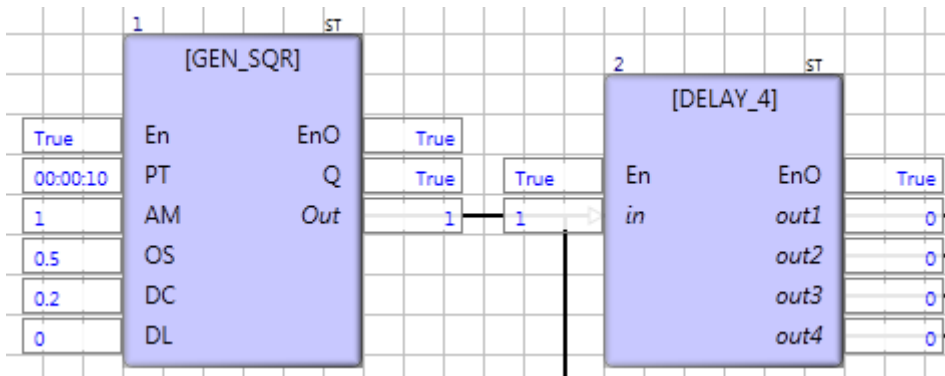


DELAY_4 передает сигнал, поданный на REAL-вход IN, на REAL-выходы с фиксированными задержками:

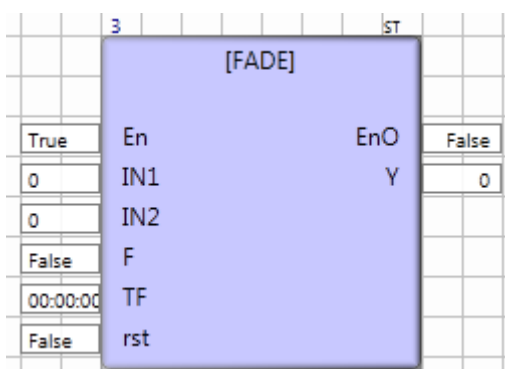
- на OUT1 – с задержкой 1 цикл;
- на OUT2 – с задержкой 2 цикла;
- на OUT3 – с задержкой 3 цикла;
- на OUT4 – с задержкой 4 цикла.

Пример

В данном примере временная диаграмма ФБ DELAY_4 получена при длительности цикла 2с. В качестве генератора прямоугольного сигнала использован ФБ GEN_SQR (OSCAT).



11.1.18.4. FADE (OSCAT)



Входы ФБ:

- IN1 (REAL) – вход 1;
- IN2 (REAL) – вход 2;

- F (BOOL) – выбор входа (TRUE соответствует IN2);
- TF (TIME) – время перехода;
- RST (BOOL) – сброс.

Выходы ФБ:

- Y (REAL) – выход ФБ.

FADE используется для безударного переключения между входами IN1 и IN2 (используется линейное сглаживание). Вход TF задает время перехода.

По команде RST=TRUE Y=IN1 (если F=FALSE) или Y=IN2 (если F=TRUE), при этом выход принимает значение входа без задержки.

Операция переключения начинается при изменении значения F и продолжается в течение времени TF. В ходе переключения с IN_M на IN_N (M, N = 1, 2) выход Y вычисляется по следующей формуле (TU – это время, истекшее с начала переключения):

$$Y = (1 - TU/TF) * IN_M + TU/TF * IN_N$$

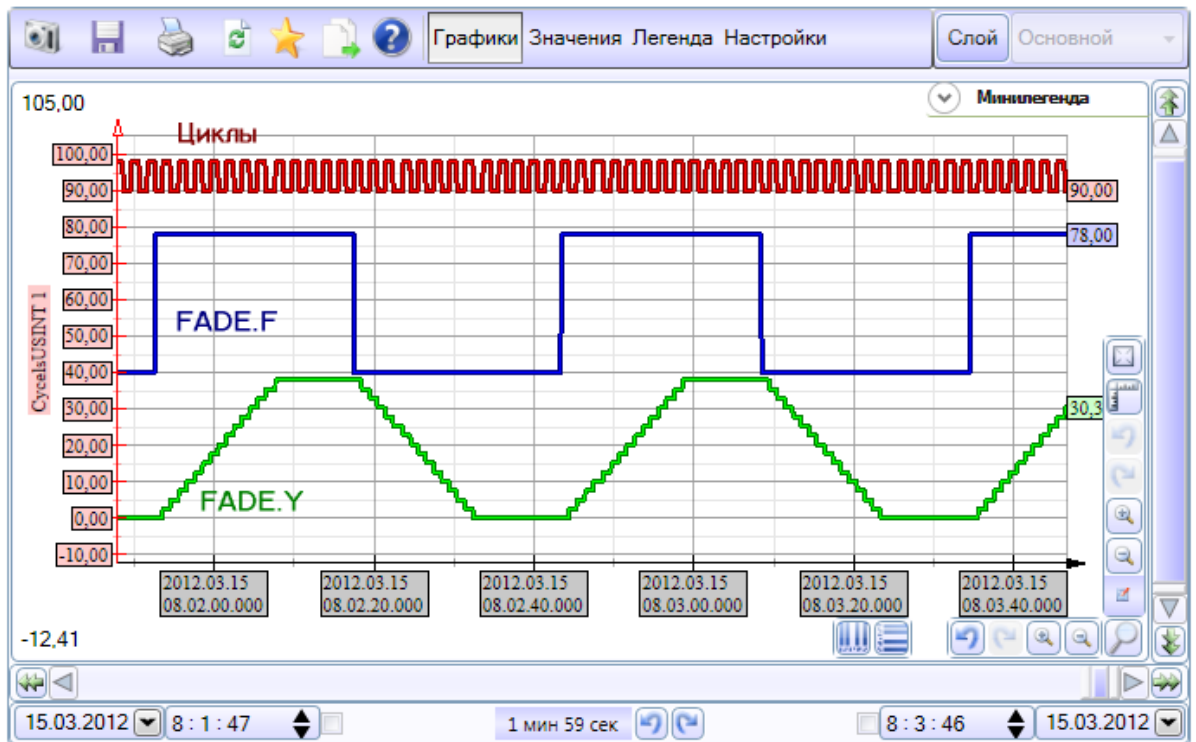
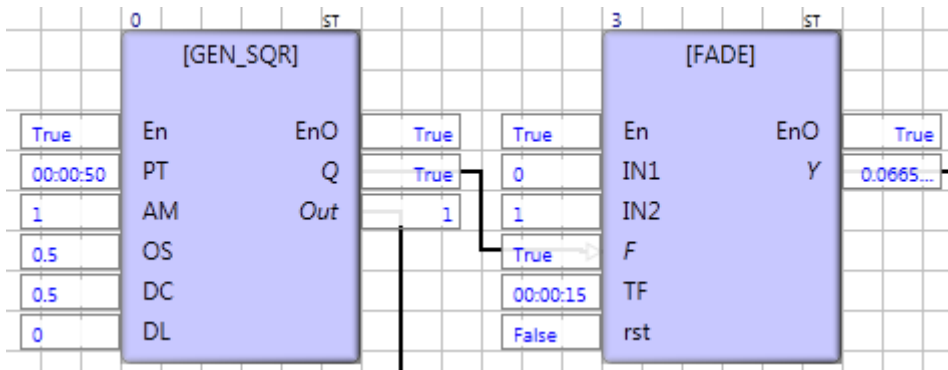
Т.е. в промежуток времени TF значения двух входов смешиваются: в начале на выход передается 0% нового значения и 100% старого, по истечении половины времени перехода (TF/2) – 50% каждого входа ($Y := IN1 * 0.5 + IN2 * 0.5$), по истечении TF – 100% нового значения. Если F принимает значение TRUE (FALSE), то, спустя время TF, Y=IN2 (IN1).

Т.к. выход вычисляется динамически, ФБ FADE может быть также использован для переключения динамических сигналов. В принципе, переход может быть разбит на 65535 шагов, однако длительность шага не может быть меньше, чем время цикла, поэтому, например, при времени цикла 10мс и TF=1с получится только $1с/10мс = 100$ шагов.

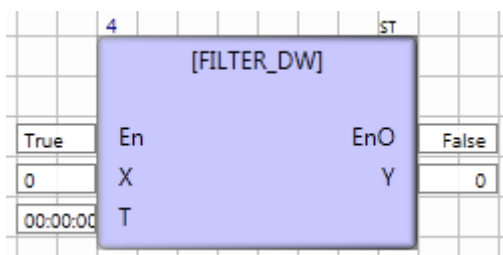
Используемые функции: RMP_W (OSCAT).

Пример

В данном примере для переключения входов (для изменения значения F) используется ФБ GEN_SQR (OSCAT). Время цикла – 1с, поэтому переход выполняется за $1с/1с=1$ шаг.



11.1.18.5. FILTER_DW (OSCAT)

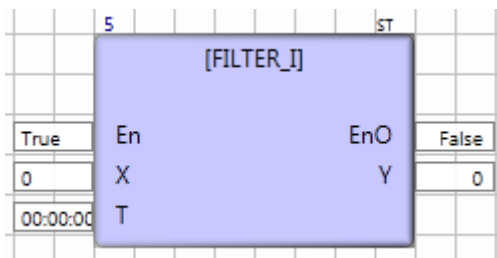


Тип данных входа T – TIME, входа X и выхода Y – DWORD.

FILTER_DW – это фильтр 1-го порядка для 32-битовых данных (DWORD). Основное назначение – фильтрация сигнала датчика для уменьшения шумов.

Алгоритм фильтра 1-го порядка рассматривается в описании ФБ FT_PT1 (OSCAT).

Используемые функции: T_PLC_MS (OSCAT).

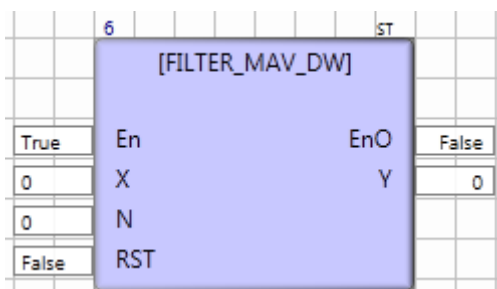
11.1.18.6. FILTER_I (OSCAT)

Тип данных входа T – TIME, входа X и выхода Y – INT.

FILTER_1 – это фильтр 1-го порядка для 16-битовых данных (INT). Основное назначение – фильтрация сигнала датчика для уменьшения шумов.

Алгоритм фильтра 1-го порядка рассматривается в описании ФБ FT_PT1 (OSCAT).

Используемые функции: T_PLC_MS (OSCAT).

11.1.18.7. FILTER_MAV_DW (OSCAT)

Входы ФБ:

- X (DWORD) – входной сигнал;
- N (UINT) – число значений в буфере;
- RST (BOOL) – сброс.

Выходы ФБ:

- Y (DWORD) – выходной сигнал.

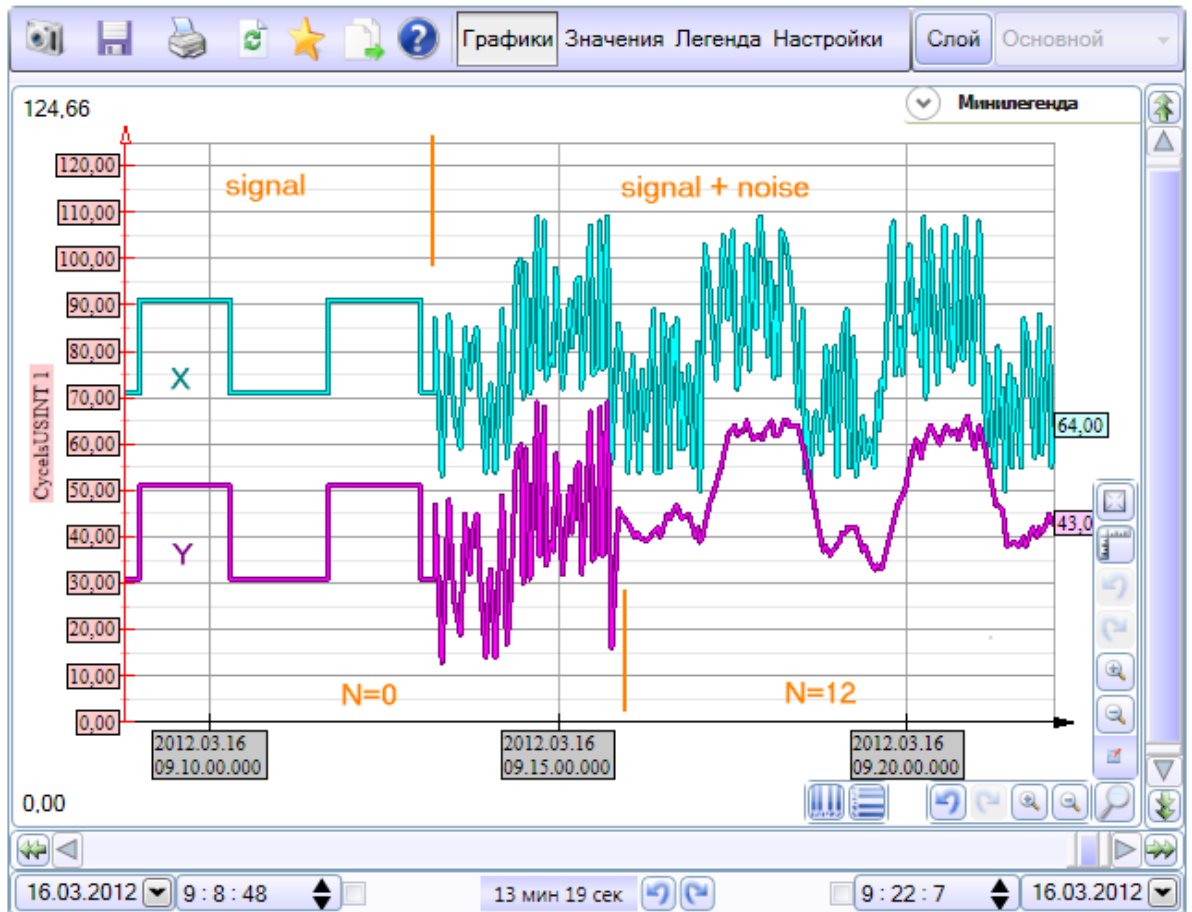
Если $N=0$ или 1, $Y:=X$, в противном случае ФБ вычисляет скользящее среднее (метод сглаживания колебаний данных):

$$Y := (X_0 + X_1 + \dots + X_{N-1}) / N$$

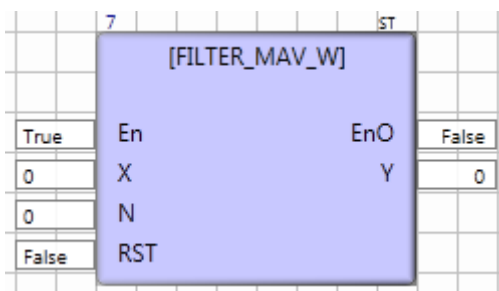
В формуле X_0 – это значение входа X в текущем цикле, X_1 – значение входа X в предыдущем цикле и т.д. Число значений, используемых в усреднении, определяет вход N ($N \leq 32$).

Если $RST=TRUE$ или $N_i \neq N_{i-1}$, $Y:=X$ и все данные в буфере заменяются значениями X.

На рисунке ниже показана фильтрация зашумленного прямоугольного сигнала с помощью данного ФБ:

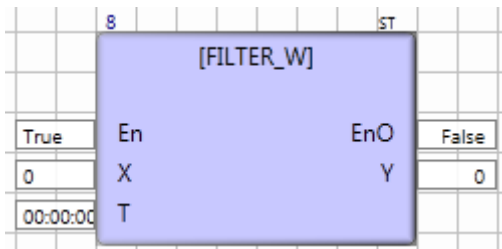


11.1.18.8. FILTER_MAV_W (OSCAT)



$FILTER_MAV_W$ – это аналог ФБ $FILTER_MAV_DW$ (OSCAT) вычисления скользящего среднего для данных WORD (тип данных X и Y – WORD).

11.1.18.9. FILTER_W (OSCAT)



Входы ФБ:

- X (WORD) – входной сигнал;
- T (TIME) – постоянная времени фильтра.

Выходы ФБ:

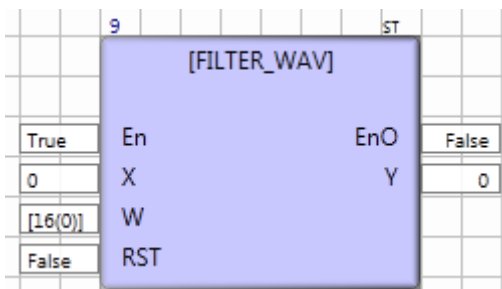
- Y (WORD) – отфильтрованный сигнал.

FILTER_W – это фильтр 1-го порядка для 16-битовых данных (WORD). Основное назначение – фильтрация сигнала датчика для уменьшения шумов.

Алгоритм фильтра 1-го порядка рассматривается в описании ФБ FT_PT1 (OSCAT).

Используемые функции: T_PLC_MS (OSCAT).

11.1.18.10. FILTER_WAV (OSCAT)



Входы ФБ:

- X (REAL) – входной сигнал;
- W (массив 16-ти переменных REAL ([0...15]) – весовые коэффициенты;
- RST (BOOL) – сброс.

Выходы ФБ:

- Y (REAL) – выходной сигнал.

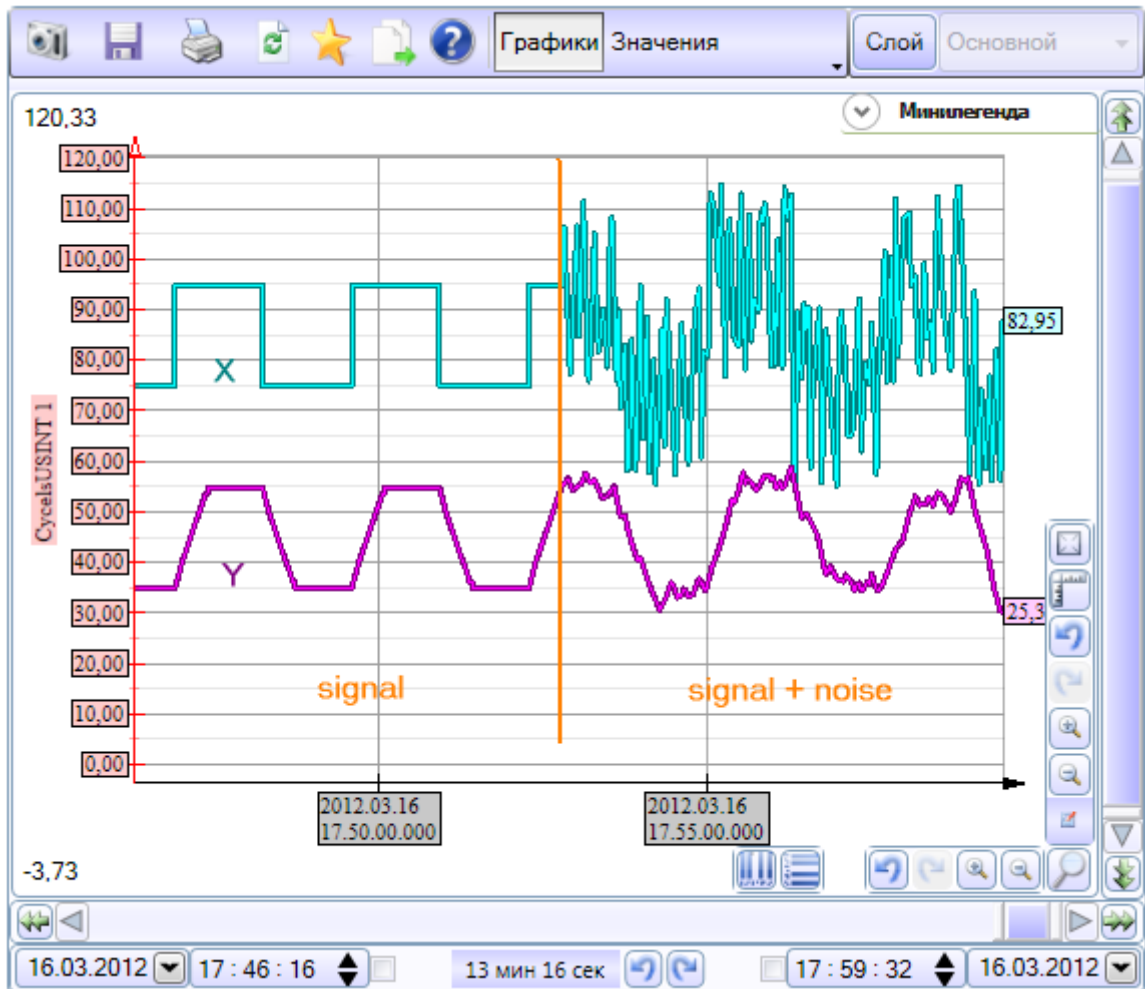
FILTER_WAV вычисляет скользящую сумму с весом:

$$Y := X_0 * W_0 + X_1 * W_1 + .. + X_{15} * W_{15}$$

В формуле X_0 – это значение входа X в текущем цикле, X_1 – значение входа X в предыдущем цикле и т.д.

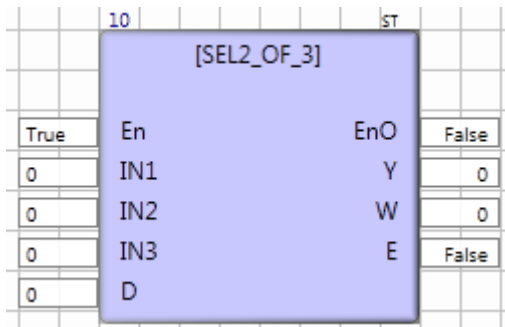
Применение `FILTER_WAV` имеет смысл, если набор весовых коэффициентов определен с помощью методов, соответствующих задаче.

Если все весовые коэффициенты равны $1/16$, `FILTER_WAV` представляет собой аналог ФБ `FILTER_MAV_DW (OSCAT)` с фиксированным буфером (16) для работы с данными REAL:



Используемые функции: `INC1 (OSCAT)`, `DEC1 (OSCAT)`.

11.1.18.11. SEL2_OF_3 (OSCAT)



Входы ФБ:

- IN1 (REAL) – вход 1;
- IN2 (REAL) – вход 2;
- IN3 (REAL) – вход 3;
- D (REAL) – отклонение.

Выходы ФБ:

- Y (REAL) – выход;
- W (INT) – предупреждение;
- E (BOOL) – индикатор ошибки.

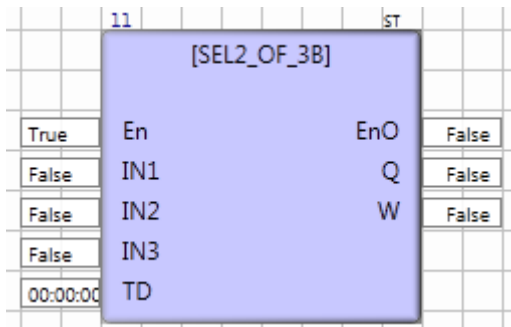
SEL2_OF_3 сравнивает абсолютные значения разностей (IN1-IN2), (IN1-IN3) и (IN2-IN3) с заданным отклонением D. Если, по меньшей мере, две разности меньше или равны D, выход Y равен среднему трех входов.

Если только одна разность меньше или равна D, то выход Y равен среднему двух входов, образующих эту разность, а выход W индицирует номер оставшегося входа.

Если все разности больше D, E=TRUE, W=4, а выход Y не изменяется и сохраняет свое значение.

Типичное применение ФБ – прием данных от трех датчиков, измеряющих одну и ту же величину с целью повышения достоверности измерений.

11.1.18.12. SEL2_OF_3B (OSCAT)



Входы ФБ:

- IN1 (BOOL) – вход 1;
- IN2 (BOOL) – вход 2;
- IN2 (BOOL) – вход 3;
- TD (TIME) – задержка для выхода W.

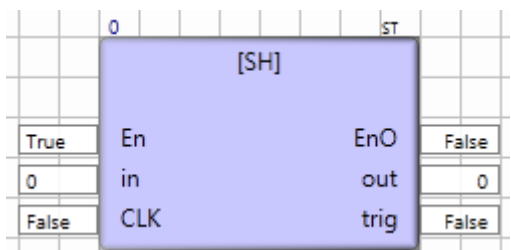
Выходы ФБ:

- Q (BOOL) – выход;
- W (BOOL) – предупреждение.

Выход Q принимает то значение, которое имеют, по меньшей мере, два входа из IN1 .. IN3.

Если одинаковое значение имеют только два входа из IN1 .. IN3, W принимает значение TRUE с задержкой TD.

11.1.18.13. SH (OSCAT)

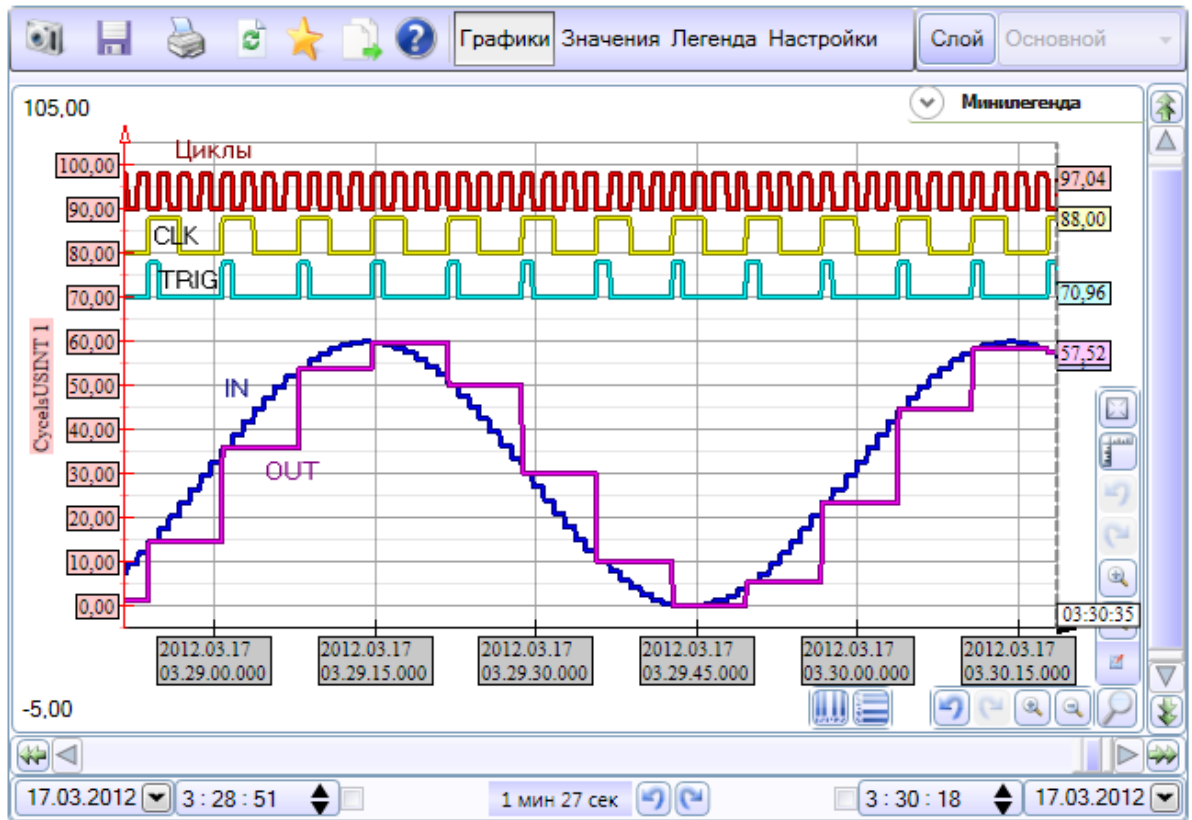


Тип данных входа IN и выхода OUT – REAL, входа CLK и TRIG – BOOL.

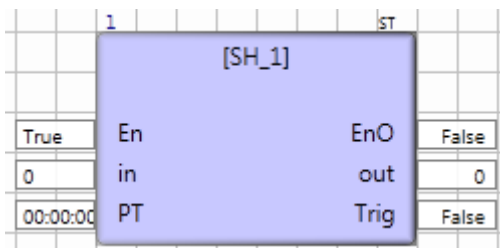
SH предназначен для выборки значений из сигнала, подаваемого на вход IN. По переднему фронту импульса на входе CLK значение IN записывается в выход OUT и сохраняется на выходе до следующего переднего фронта на CLK.

При каждой записи в OUT TRIG=TRUE в течение 1 цикла.

Следующая временная диаграмма иллюстрирует алгоритм ФБ:



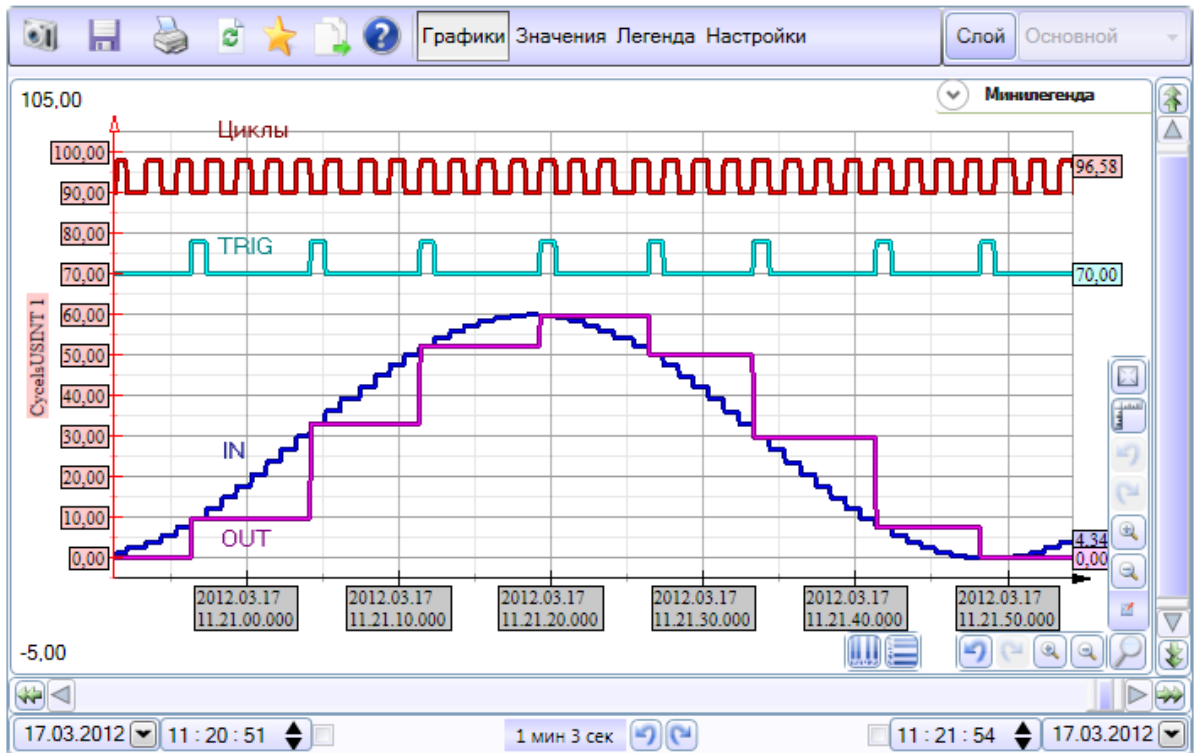
11.1.18.14. SH_1 (OSCAT)



Тип данных входа IN и выхода OUT – REAL, входа PT – TIME, выхода TRIG – BOOL.

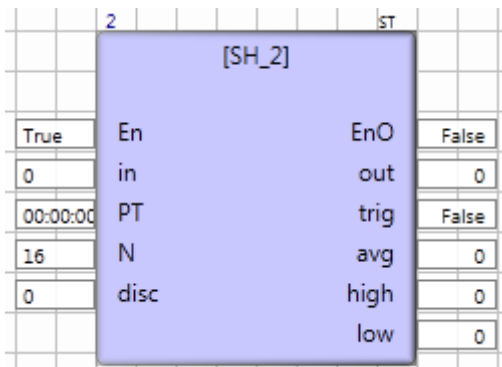
SH_1 предназначен для периодической выборки значений из сигнала, подаваемого на вход IN. Период задается значением входа PT. Один раз за период значение IN записывается в выход OUT и сохраняется на выходе до следующей выборки. При каждой записи в OUT TRIG=TRUE в течение 1 цикла.

Следующая временная диаграмма иллюстрирует алгоритм ФБ:



Используемые функции: T_PLC_MS (OSCAT).

11.1.18.15. SH_2 (OSCAT)



Тип данных входа IN и выходов OUT, AVG, HIGH и LOW – REAL, входов N и DISC – INT, входа PT – TIME, выхода TRIG – BOOL.

SH_2 предназначен для периодической выборки значений из сигнала, подаваемого на вход IN, и вычисления ряда статистических характеристик.

Вход PT задает период выборки. Один раз за период значение IN записывается в выход OUT и сохраняется на выходе до следующей выборки. При каждой записи в OUT TRIG=TRUE в течение 1 цикла.

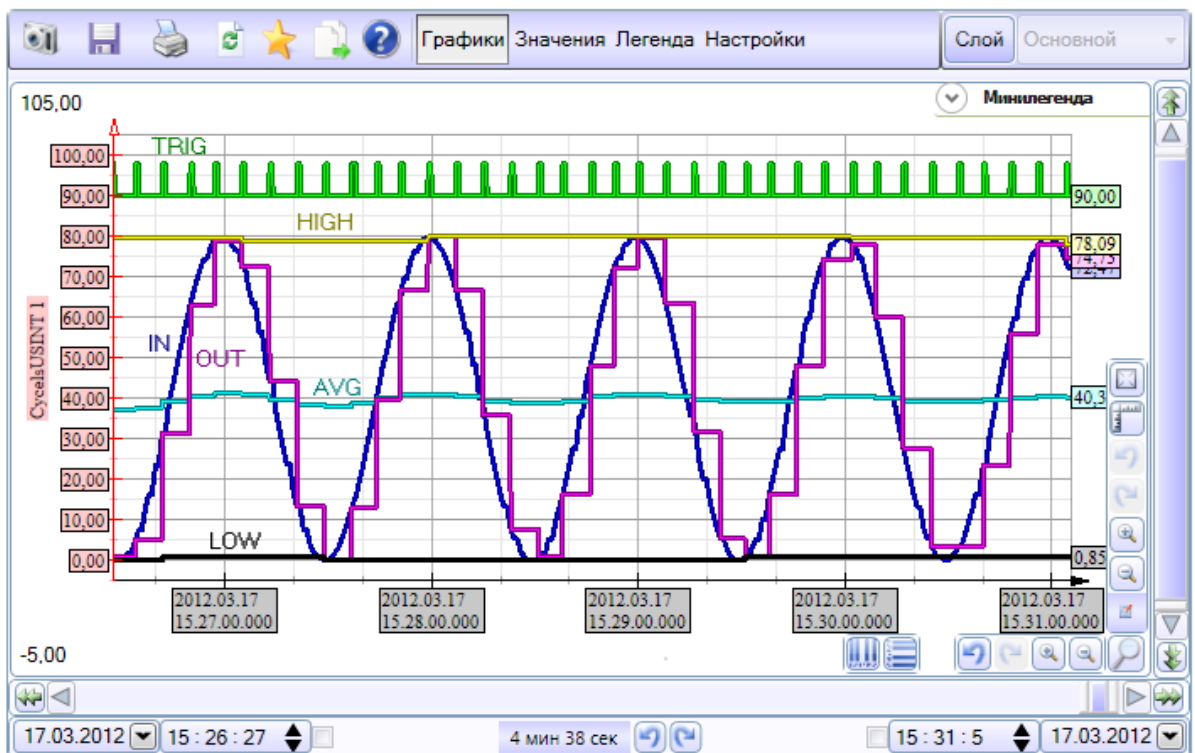
Вход N задает число последних выбранных значений (не более 16), по которым вычисляются среднее, максимальное и минимальное значения. Алгоритм вычислений зависит от входа DISC:

- если $DISC=0$, в вычислениях участвуют все N значений;
- если $DISC=1$, в вычислениях участвуют $N-1$ значений (игнорируется наименьшее значение);
- если $DISC=2$, в вычислениях участвуют $N-2$ значений (игнорируются наименьшее и наибольшее значения);
- если $DISC=3$, в вычислениях участвуют $N-3$ значений (игнорируются два наименьших и одно наибольшее значения);
- и т.д.

Например, если $N=5$ и $DISC=2$, из последних 5 выбранных значений отбрасываются наименьшее и наибольшее, и по оставшимся трем значениям вычисляются среднее, максимальное и минимальное значения.

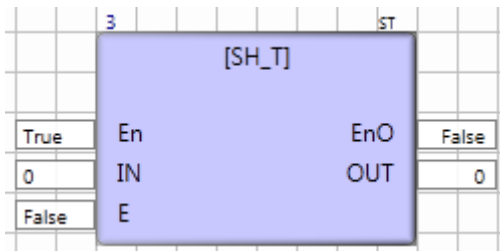
Среднее, максимальное и минимальное значения записываются соответственно в выходы AVG , $HIGH$ и LOW .

Следующая временная диаграмма иллюстрирует работу ФБ при $DISC=0$:



Используемые функции: T_PLC_MS (OSCAT).

11.1.18.16. SH_T (OSCAT)

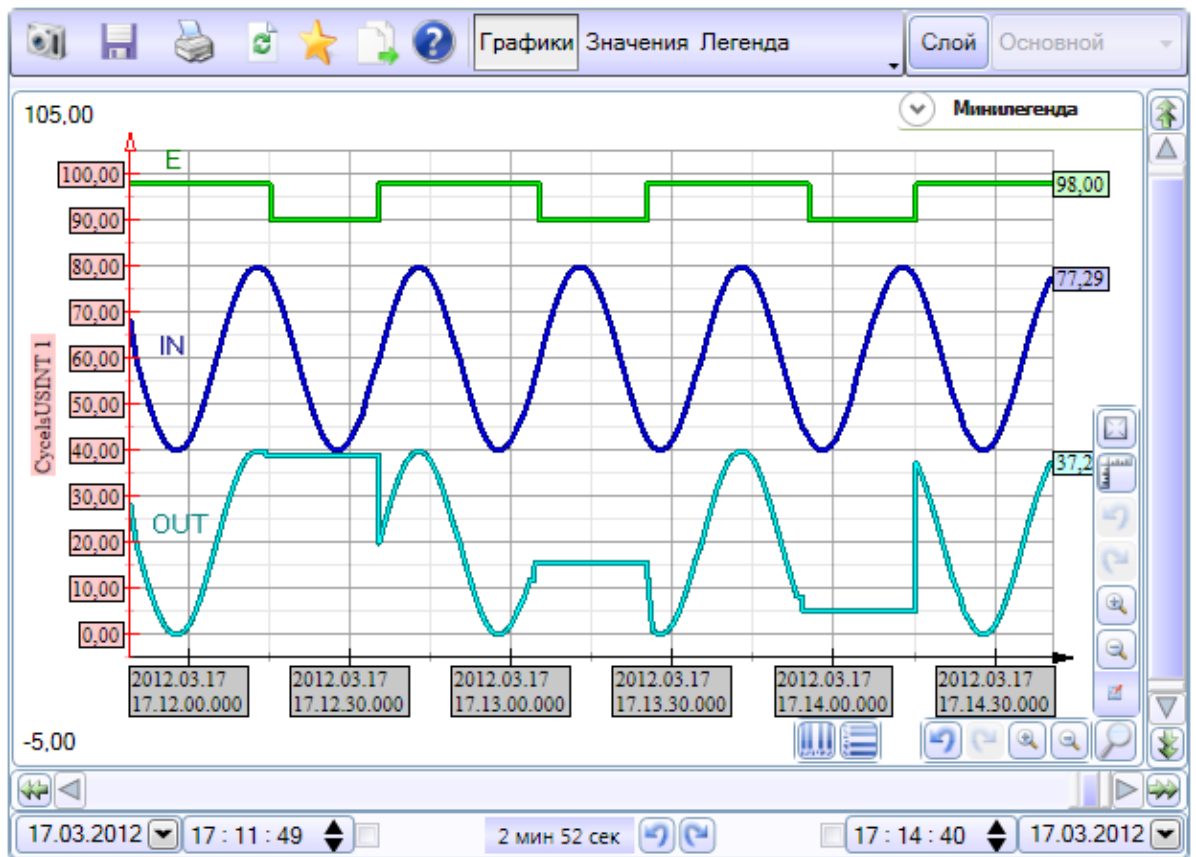


Тип данных входа IN и выхода OUT – REAL, входа E – BOOL.

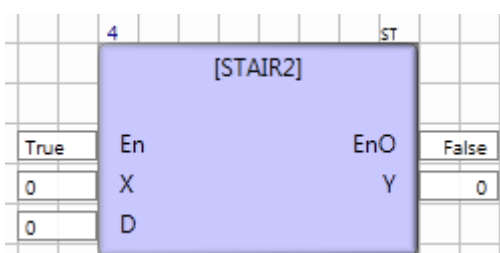
При E=TRUE SH_T передает значения IN на выход OUT.

По заднему фронту на входе E передача IN->OUT останавливается, а OUT сохраняет последнее значение до тех пор, пока передача IN->OUT не возобновится (т.е. пока E не будет присвоено значение TRUE).

Следующая временная диаграмма иллюстрирует работу ФБ:



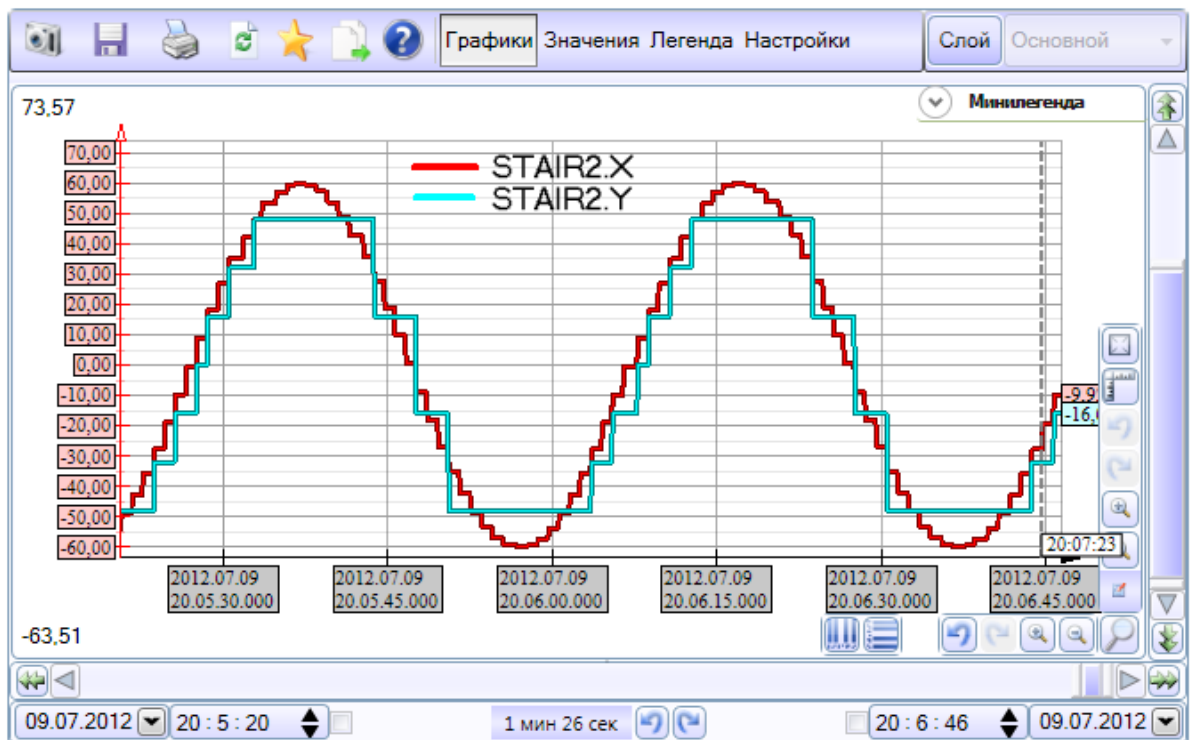
11.1.18.17. STAIR2 (OSCAT)



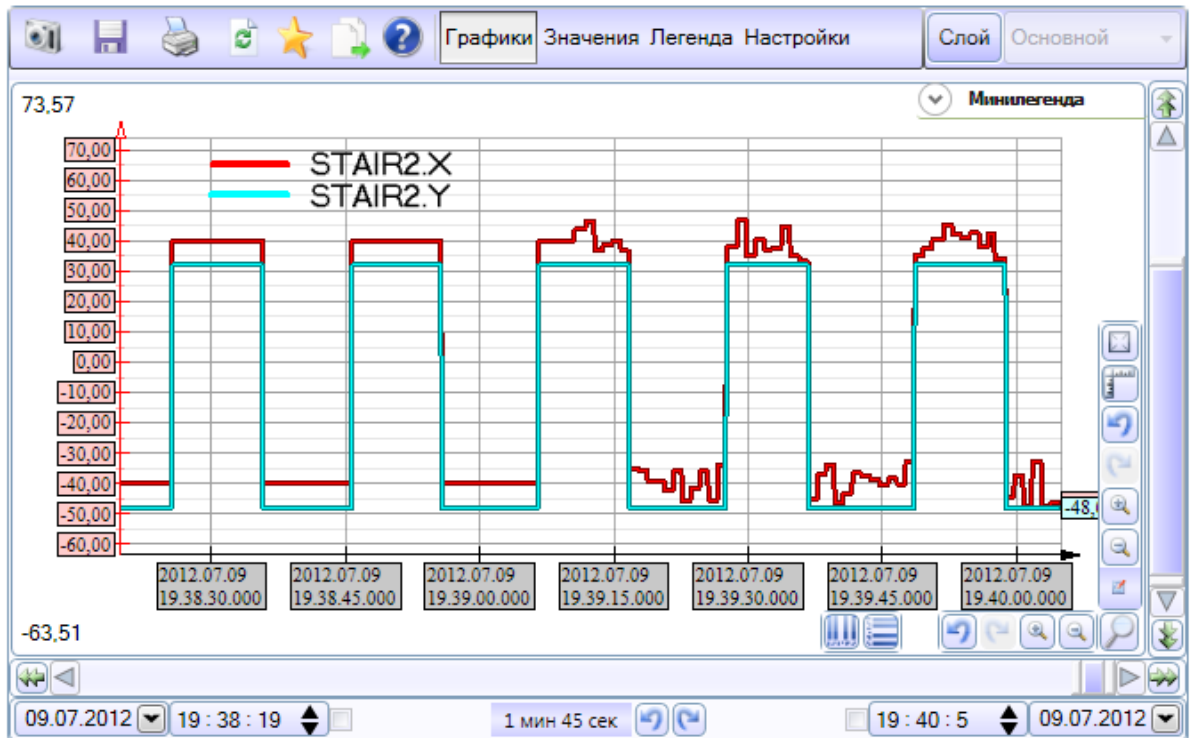
Тип данных входов и выхода функции – REAL.

Если $D \leq 0$, $Y = X$.

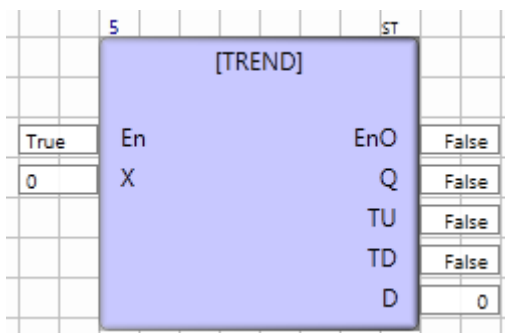
Если $D > 0$, ФБ преобразует гладкий сигнал, поданный на вход X, в ступенчатый (выход Q принимает только значения $n \cdot D$, где n – любое целое число), при этом передаточная характеристика ФБ имеет гистерезис D . Следующий график иллюстрирует работу ФБ STAIR2 на синусоидальном сигнале при $D=16$ (вследствие использования в алгоритме функции FLOOR (OSCAT) при уменьшении входного сигнала шаги выходного сигнала могут быть равны $2D$):



В некоторых случаях ФБ STAIR2 может использоваться как входной фильтр:



11.1.18.18. TREND (OSCAT)

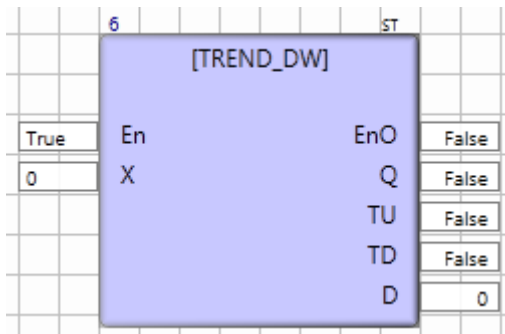


Тип данных входа X и выхода D – REAL, выходов Q, TU и TD – BOOL.

TREND сравнивает значения входа X в текущем и предыдущем циклах. По результатам сравнения выходы ФБ принимают следующие значения:

- $D := X_{\text{текущее}} - X_{\text{предыдущее}}$;
- если $X_{\text{текущее}} > X_{\text{предыдущее}}$, $TU := \text{TRUE}$, в противном случае $TU := \text{FALSE}$;
- если $X_{\text{текущее}} < X_{\text{предыдущее}}$, $TD := \text{TRUE}$, в противном случае $TD := \text{FALSE}$;
- если $X_{\text{текущее}} \neq X_{\text{предыдущее}}$, $Q := \text{TRUE}$, в противном случае $Q := \text{FALSE}$.

11.1.18.19. TREND_DW (OSCAT)



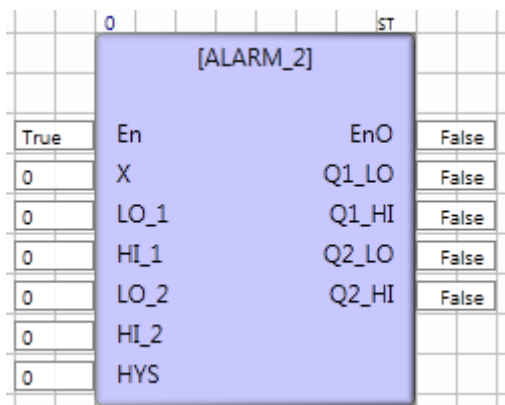
Тип данных входа X и выхода D – DWORD, выходов Q, TU и TD – BOOL.

TREND_DW сравнивает значения входа X в текущем и предыдущем циклах. По результатам сравнения выходы ФБ принимают следующие значения:

- $D := \text{abs}(X_{\text{текущее}} - X_{\text{предыдущее}})$;
- если $X_{\text{текущее}} > X_{\text{предыдущее}}$, $Q = \text{TRUE}$, $TU = \text{TRUE}$, $TD = \text{FALSE}$;
- если $X_{\text{текущее}} < X_{\text{предыдущее}}$, $Q = \text{FALSE}$, $TU = \text{FALSE}$, $TD = \text{TRUE}$;
- если $X_{\text{текущее}} = X_{\text{предыдущее}}$, $TU = \text{FALSE}$, $TD = \text{FALSE}$, а Q сохраняет предыдущее значение.

11.1.19. OSCAT.ФБ.МОДУЛИ ИЗМЕРЕНИЯ

11.1.19.1. ALARM_2 (OSCAT)



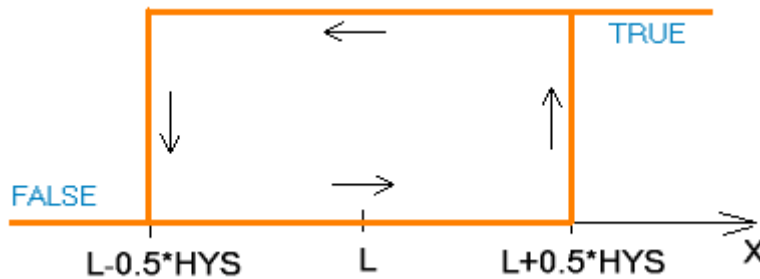
Тип данных входов – REAL, выходов – BOOL.

ФБ устанавливает/сбрасывает выход в зависимости от положения значения входа X относительно соответствующей границы по следующему алгоритму (выход Q1_LO соответствует границе LO_1, Q1_HI – HI_1, Q2_LO – LO_2, Q2_HI – HI_2) :

- если выход имеет значение FALSE, то он примет значение TRUE только после того, как X станет больше величины $(L + 0.5 * \text{HYS})$ (здесь L – граница, соответствующая выходу);

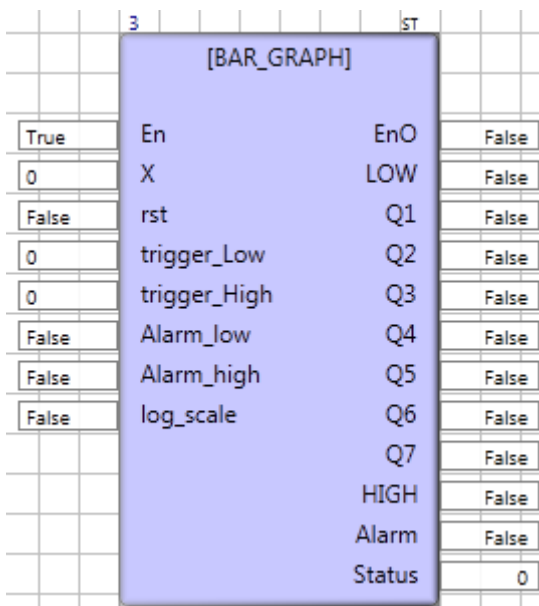
- если выход имеет значение TRUE, то он примет значение FALSE только после того, как X станет меньше величины $(L-0.5*HYS)$.

Т.е. значение X переходит границу по гистерезису:



Вход HYS задает гистерезис одновременно для всех границ, его значение должно быть больше или равно нулю. Если $HYS=0$, гистерезис отсутствует (X сравнивается с самой границей).

11.1.19.2. BAR_GRAPH (OSCAT)



Входы ФБ:

- X (REAL) – аналоговый сигнал;
- RST (BOOL) – сброс;
- TRIGGER_LOW (REAL) – нижняя граница;
- TRIGGER_HIGH (REAL) – верхняя граница;
- ALARM_LOW (BOOL) – разрешение установки ALARM при $X < TRIGGER_LOW$;

- **ALARM_HIGH (BOOL)** – разрешение установки **ALARM** при $X \geq \text{TRIGGER_HIGH}$;
- **LOG_SCALE (BOOL)** – вид разбиения диапазона $[\text{TRIGGER_LOW}, \text{TRIGGER_HIGH}]$ на поддиапазоны.

Выходы ФБ:

- **LOW (BOOL)** – индикатор выхода X за нижнюю границу ($X < \text{TRIGGER_LOW}$);
- **Q1..Q7 (BOOL)** – индикаторы нахождения X в поддиапазонах;
- **HIGH (BOOL)** – индикатор выхода X за верхнюю границу ($X \geq \text{TRIGGER_HIGH}$);
- **ALARM (BOOL)** – индикатор аварии;
- **STATUS (Byte)** – статус ESR.

BAR_GRAPH – это многоуровневый компаратор. Пороговые значения для выходов **LOW** и **HIGH** задаются соответственно входами **TRIGGER_LOW** и **TRIGGER_HIGH** (**TRIGGER_HIGH** должно быть больше **TRIGGER_LOW**). **LOW=TRUE**, если $X < \text{TRIGGER_LOW}$, и **HIGH=TRUE**, если $X \geq \text{TRIGGER_HIGH}$.

Если **ALARM_LOW=TRUE** и/или **ALARM_HIGH=TRUE**, то **ALARM=TRUE**, если соответственно $X < \text{TRIGGER_LOW}$ или $X \geq \text{TRIGGER_HIGH}$.

Вне зависимости от X , **LOW**, **HIGH** и **ALARM** сохраняют значение **TRUE** до сброса (**RST=TRUE**).

Диапазон $[\text{TRIGGER_LOW}, \text{TRIGGER_HIGH}]$ разбит на 7 поддиапазонов по линейному (**LOG_SCALE=FALSE**) или логарифмическому (**LOG_SCALE=TRUE**) закону. Нахождение X в поддиапазоне индицирует соответствующий выход **Q1..Q7** (одновременно только один из выходов **Q1..Q7** может иметь значение **TRUE**). Выходы **Q1..Q7** игнорируют команду сброса (**RST=TRUE**).

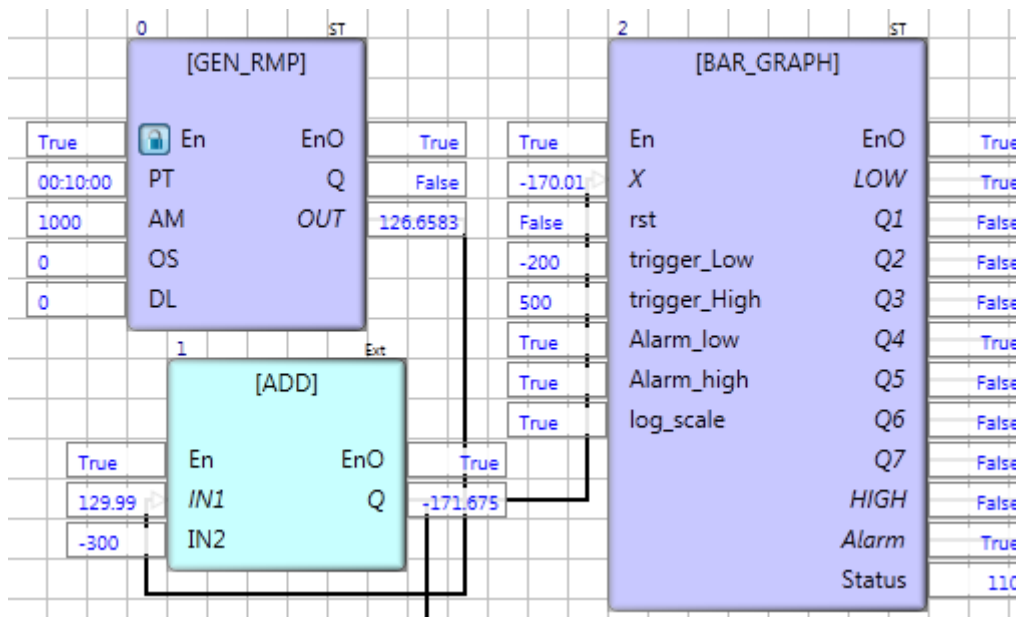
Выход **STATUS** индицирует ESR-совместимый статус:

СТАТУС	
110	$\text{TRIGGER_LOW} < X < \text{TRIGGER_HIGH}$
111	$X < \text{TRIGGER_LOW}$ (LOW=TRUE)
112	$X > \text{TRIGGER_HIGH}$ (HIGH = TRUE)
1	$X < \text{TRIGGER_LOW}$ (LOW=TRUE) и ALARM_LOW=TRUE

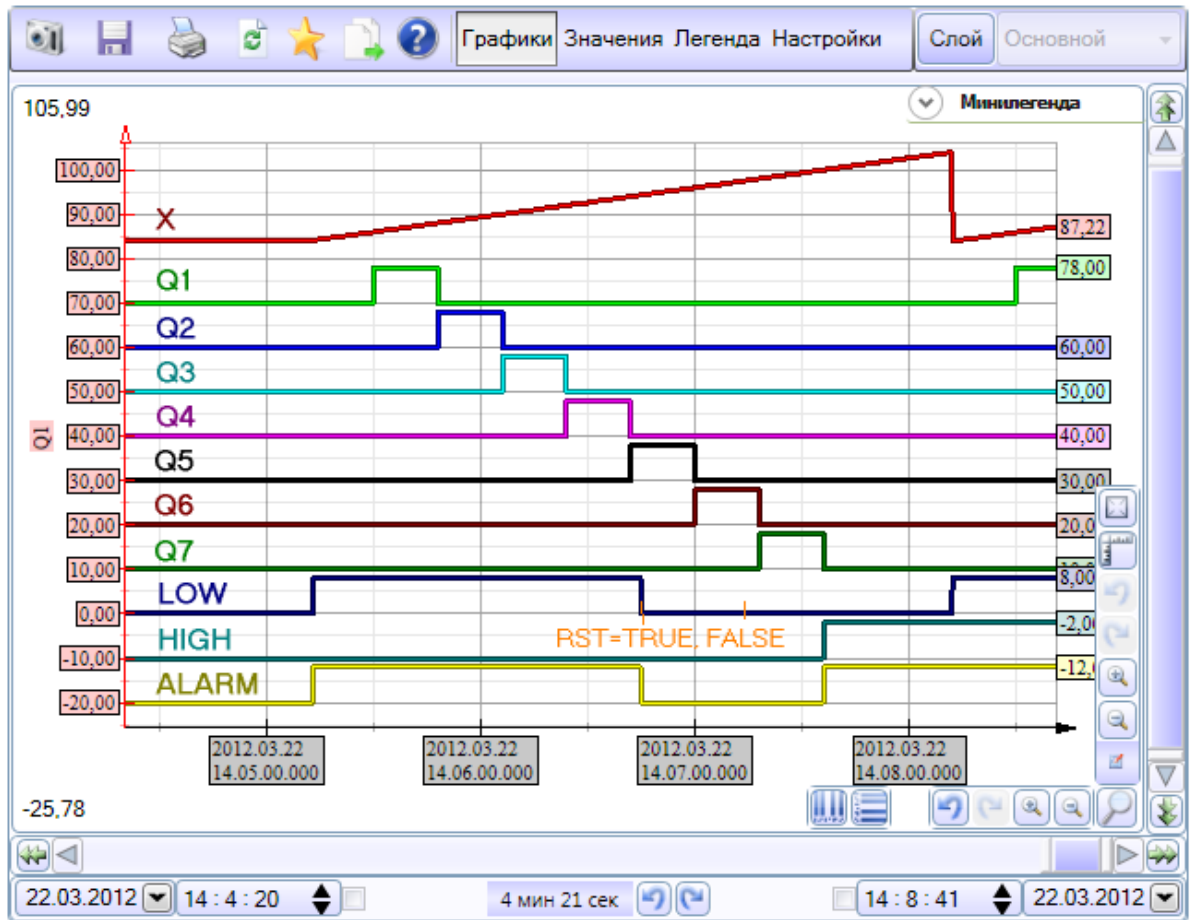
СТАТУС	
2	X >= TRIGGER_HIGH (HIGH = TRUE) и ALARM_HIGH=TRUE

Примеры

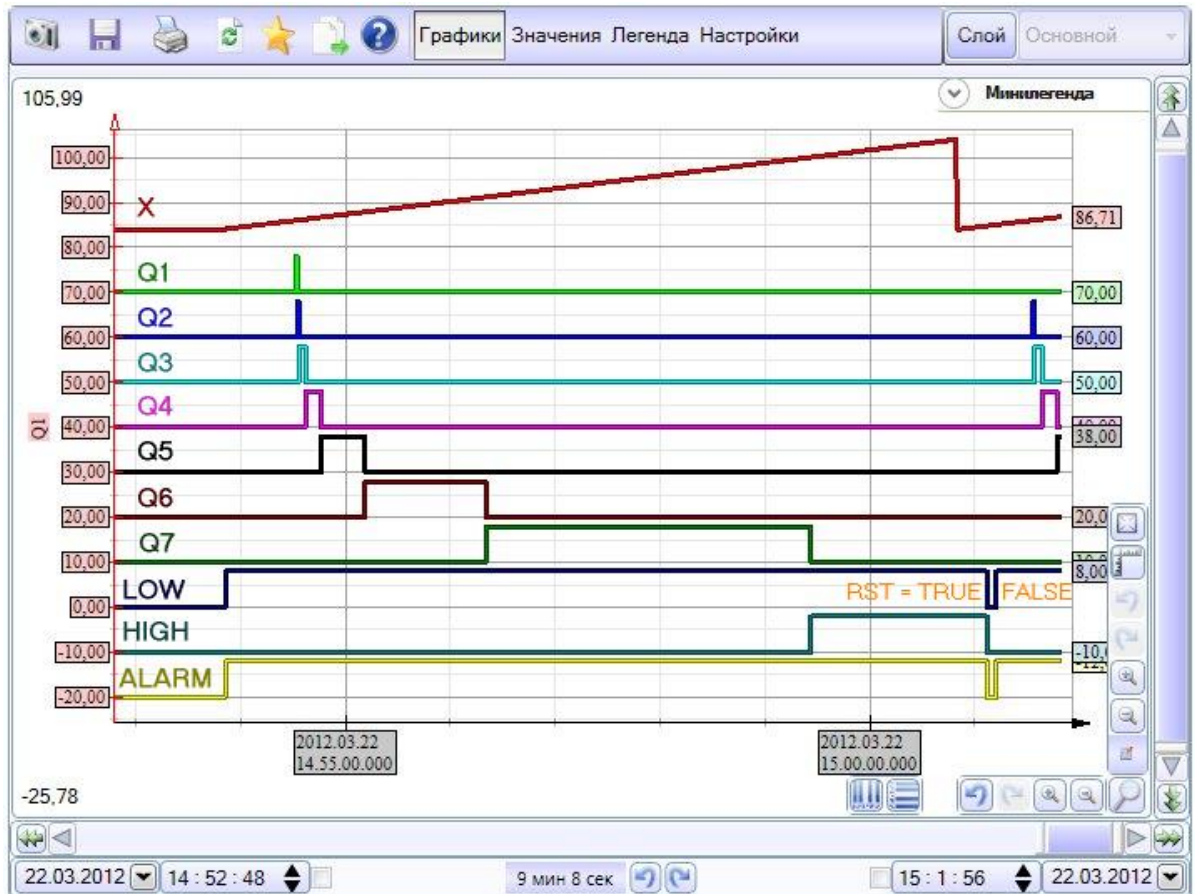
Приведенные ниже временные диаграммы ФБ BAR_GRAPH получены с помощью следующей программы (для генерации пилообразного сигнала используется ФБ GEN_RMP (OSCAT)):



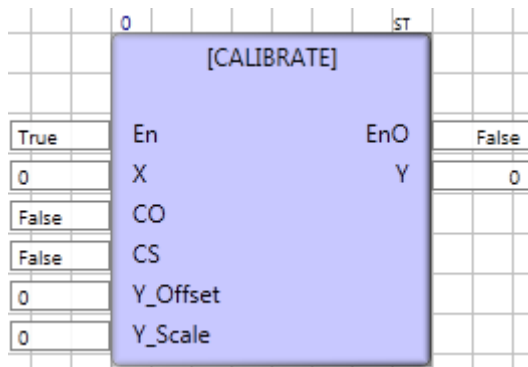
Временная диаграмма при линейном разбиении диапазона [TRIGGER_LOW, TRIGGER_HIGH] (LOG_SCALE=FALSE):



Временная диаграмма при логарифмическом разбиении диапазона [TRIGGER_LOW, TRIGGER_HIGH] (LOG_SCALE=TRUE):



11.1.19.3. CALIBRATE (OSCAT)



Входы ФБ:

- X (REAL) – входной сигнал;
- CO (BOOL) – импульс для сохранения смещения;
- CS (BOOL) – импульс для сохранения множителя;
- Y_OFFSET (REAL) – нижняя граница Y (константа, может быть задана только до старта);

- Y_SCALE (REAL) – верхняя граница Y (константа, может быть задана только до старта).

Выходы ФБ:

- Y (REAL) – масштабированный сигнал.

CALIBRATE масштабирует входной сигнал X по формуле

$$Y := (X + OFFSET) * SCALE$$

где $OFFSET$ – смещение, а $SCALE$ – множитель (оба параметра вычисляются при предварительной калибровке ФБ).

Данная формула некорректна, и ФБ может применяться только при $Y_OFFSET=0$. Для корректного масштабирования следует использовать ФБ CALIBRATE_mplc.

Для калибровки ФБ нужно выполнить следующую последовательность действий:

- присвоить входу X значение $XMIN$, при котором Y должен быть равен Y_OFFSET , и подать импульс на вход CO для вычисления и запоминания смещения $OFFSET$;
- присвоить входу X значение $XMAX$, при котором Y должен быть равен Y_SCALE , и подать импульс на вход CS для вычисления и запоминания множителя $SCALE$.

После калибровки ФБ автоматически готов к работе.

Смещение $OFFSET$ и множитель $SCALE$ сохраняются в энергонезависимой памяти ПЛК, поэтому не утрачиваются при перезагрузке или выключении питания ПЛК.

Пример

Пусть термопара работает линейно в диапазоне температур от 0 до 700С. Ток термопары при 00С – 4mA, при 700С – 20mA. Сигнал термопары подается на вход X ФБ CALIBRATE, который должен преобразовывать входные значения в температуру.

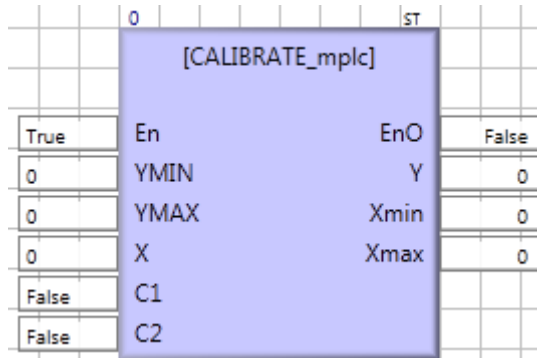
Для решения задачи сначала нужно откалибровать ФБ. Для калибровки нужно выполнить следующую последовательность действий:

- задать $Y_OFFSET=0$ и $Y_SCALE=70$;
- при температуре термопары 00С подать импульс на вход CO для вычисления и запоминания смещения $OFFSET$;
- при температуре термопары 700С подать импульс на вход CS для вычисления и запоминания множителя $SCALE$.

После калибровки значение выхода Y соответствует температуре, измеряемой термопарой.

Для решения данной задачи ФБ CALIBRATE может быть использован, поскольку $Y_OFFSET=0$.

11.1.19.4. CALIBRATE_MPLC



Входы ФБ:

- X (REAL) – входной сигнал;
- C1 (BOOL) – импульс для сохранения XMIN;
- C2 (BOOL) – импульс для сохранения XMAX;
- YMIN (REAL) – нижняя граница требуемого диапазона;
- YMAX (REAL) – верхняя граница требуемого диапазона.

Выходы ФБ:

- Y (REAL) – масштабированный сигнал;

CALIBRATE предназначен для линейного масштабирования входного сигнала. Параметры масштабирования вычисляются при калибровке ФБ, в ходе которой произвольный диапазон входных значений [XMIN, XMAX] преобразуется в требуемый диапазон [YMIN, YMAX] по линейному закону.

Для калибровки ФБ необходимо выполнить следующую последовательность действий, после которой ФБ автоматически переходит в рабочий режим:

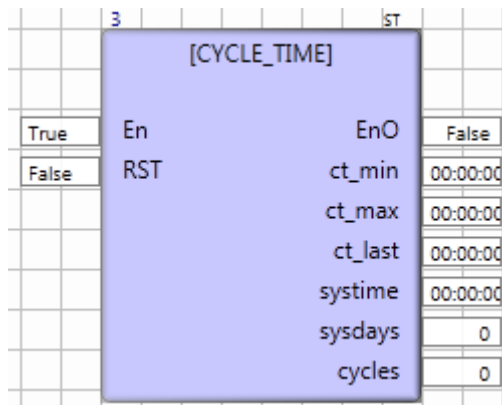
- подать XMIN на вход X;
- подать импульс на вход C1 для запоминания XMIN во внутренней переменной ФБ;
- подать XMAX на вход X;
- подать импульс на вход C2 для запоминания XMAX во внутренней переменной ФБ.

Минимальное и максимальное значения входного диапазона сохраняются в энергонезависимой памяти ПЛК, поэтому не утрачиваются при перезагрузке или

выключении питания ПЛК. ФБ отображает сохраненные значения на выходах Xmin и Xmax.

ФБ не работает, если $X_{max} = X_{min}$.

11.1.19.5. CYCLE_TIME (OSCAT)



Входы ФБ:

- RST (BOOL) – сброс.

Выходы ФБ:

- CT_MIN (TIME) – минимальная длительность цикла;
- CT_MAX (TIME) – максимальная длительность цикла;
- CT_LAST (TIME) – длительность предыдущего цикла;
- SYSTIME (TIME) – время, прошедшее с начала работы;
- SYSDAYS (INT) – число целых суток, прошедших с начала работы;
- CYCLES (DWORD) – число циклов с начала работы.

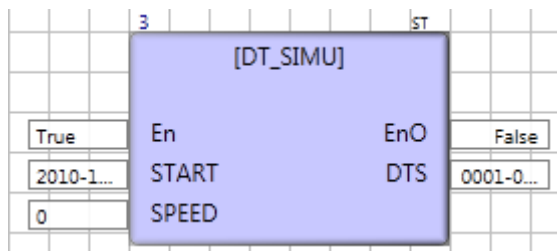
CYCLE_TIME отслеживает реальную длительность цикла ПЛК и индицирует информацию, связанную с этим параметром, а также информацию об общем времени работы ПЛК. Число циклов с начала работы также измеряется.

CT_MIN, CT_MAX и CT_LAST вычисляются корректно в том числе и при переполнении таймера – см. описание функции TIME, таймер ПЛК.

По этой информации ФБ управления могут сообщать об ошибках (например, длительность цикла слишком велика, и поэтому корректное управление не может быть гарантировано).

Используемые функции: T_PLC_MS (OSCAT).

11.1.19.6. DT_SIMU (OSCAT)



Входы ФБ:

- START (DT) – начальное показание симулируемых часов;
- SPEED (REAL) – коэффициент ускорения ($SPEED > 1$) или замедления ($0 < SPEED < 1$) хода симулируемых часов.

Выходы ФБ:

- DTS (DT) – симулируемые часы.

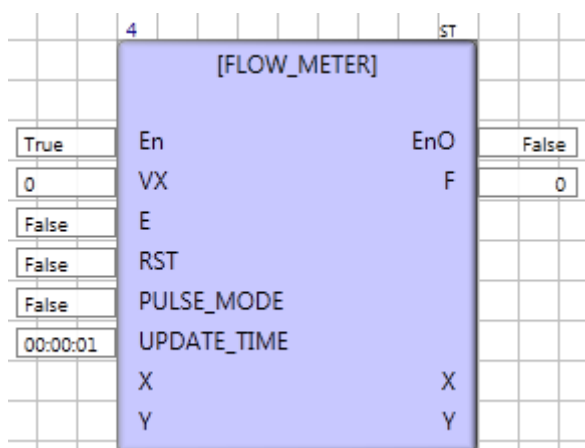
DT_SIMU – это симулятор часов с произвольной скоростью хода, которую определяет вход SPEED. Например, при $SPEED = 0.25$ симулируемые часы ходят в 4 раза медленнее астрономических часов (1с за 4 астрономические секунды), а при $SPEED = 4$ – в 4 раза быстрее (4с за 1 астрономическую секунду). Если $SPEED = 0$, скорость хода симулируемых часов – 1с/цикл, если $SPEED = 1$, симулируемые часы ходят так же, как астрономические.

Вход START задает начальное показание симулируемых часов.

ФБ может быть использован при отладке в качестве симулятора RTC, а также для настройки скорости RTC при тестировании.

Используемые функции: T_PLC_MS (OSCAT).

11.1.19.7. FLOW_METER (OSCAT)



Входы ФБ:

- VX (REAL) – входной сигнал;
- E (BOOL) – вспомогательный вход;
- RST (BOOL) – сброс;
- PULSE_MODE (BOOL) – режим ФБ;
- UPDATE_TIME (TIME) – интервал усреднения вычисляемой скорости расхода.

Входы/выходы ФБ:

- X (REAL) – дробная часть суммарного расхода;
- Y:UDINT – целая часть суммарного расхода.

Выходы ФБ:

- F (REAL) – вычисленное значение скорости расхода.

ФБ FLOW_METER вычисляет суммарный расход и усредненную скорость расхода в двух режимах.

Текущее значение суммарного расхода выводится на выходы Y (целая часть) и X (дробная часть). В программе, использующей ФБ FLOW_METER, переменные, связанные с выходами X и Y ФБ, могут быть объявлены сохраняемыми в энергонезависимой памяти контроллера (с помощью модификатора RETAIN оператора VAR); в этом случае после сбоя питания вычисления начнутся с сохраненной величины расхода.

По значениям Y и X вычисляется и выводится на выход F скорость расхода в час, усредненная по интервалу UPDATE_TIME (выход F индицирует среднюю скорость расхода в час за предыдущий интервал UPDATE_TIME). Значение UPDATE_TIME должно быть отлично от 0, в противном случае F=0.

По команде RST=TRUE производится полный сброс с обнулением всех выходов, и в дальнейшем, после RST=FALSE, вычисления начнутся с нулевой величины расхода.

Для переключения режима ФБ нужно выполнить следующую последовательность команд:

- RST:=TRUE;
- PULSE_MODE:=TRUE или FALSE;
- RST:=FALSE.

Режим 1 (PULSE_MODE=FALSE)

В первом режиме ($PULSE_MODE=FALSE$) на вход VX подается скорость расхода в час (функция времени).

По команде $E=TRUE$ запускается алгоритм ФБ, и расход вычисляется методом интегрирования входного сигнала.

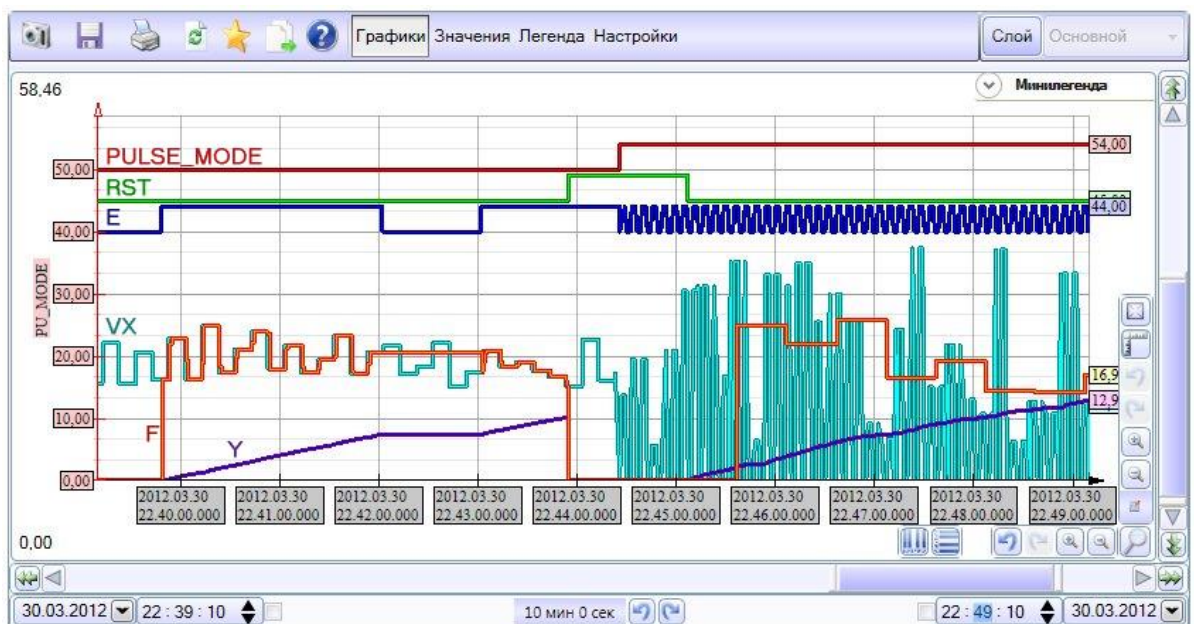
По команде $E=FALSE$ интегрирование приостанавливается, а X и Y сохраняют свои значения, и при последующем возобновлении интегрирования (по команде $E=TRUE$) сохраненные значения X и Y определяют начальное значение интеграла. При $E=FALSE$ выход F также сохраняет свое значение.

Режим 2 ($PULSE_MODE=TRUE$)

Во втором режиме ($PULSE_MODE=TRUE$) по переднему фронту на входе E к текущему значению расхода прибавляется значение VX.

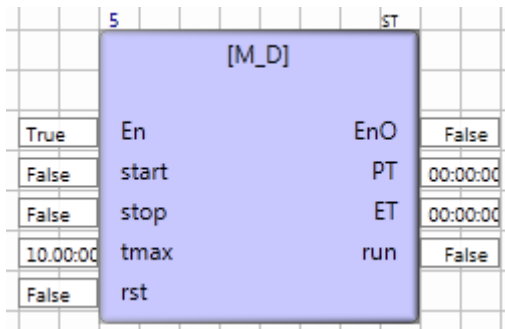
Пример

Следующая временная диаграмма иллюстрирует функционал FLOW_METER:



Во втором режиме имитируется сбор данных с 15-ти расходомеров (на вход VX подается сигнал с мультиплексора), и ФБ вычисляет расход и скорость расхода суммарно по всем 15-ти линиям, на которых установлены эти расходомеры. На опрос одного расходомера в примере затрачивается 2с, поэтому интервал усреднения – 30с. Время цикла в примере – 100мс.

11.1.19.8. M_D (OSCAT)



Тип данных входов START, STOP и RST и выхода RUN – BOOL, входа TMAX и выходов PT и ET – TIME.

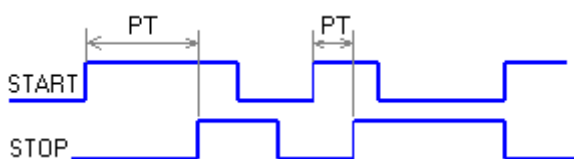
M_D измеряет интервал времени между передним фронтом на входе START и передним фронтом на входе STOP. По окончании измерения результат выводится на выход PT. В течение измерения выход ET индицирует время, прошедшее с момента детектирования переднего фронта на входе START.

Передний фронт на входе START обрабатывается (и измерение начинается) только после того, как STOP=FALSE и RST=FALSE. Если START=TRUE при вызове ФБ, то это также не рассматривается как передний фронт.

В течение измерения RUN=TRUE. По окончании измерения выходы PT и ET сохраняют свои значения до начала нового измерения.

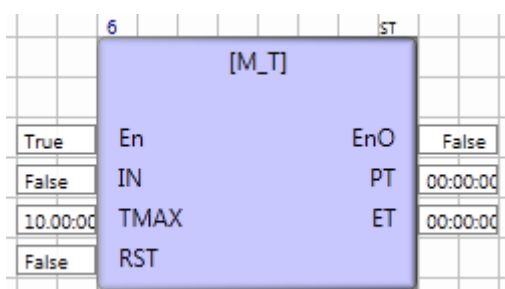
По команде RST=TRUE измерение прерывается и выходы обнуляются.

Измерение также прерывается и выходы обнуляются, если ET достигает значения TMAX (TMAX задает максимально возможную длительность измеряемого интервала времени). По умолчанию TMAX=T#10d.



Используемые функции: T_PLC_MS (OSCAT).

11.1.19.9. M_T (OSCAT)



Тип данных входов IN и RST – BOOL, входа TMAX и выходов PT и ET – TIME.

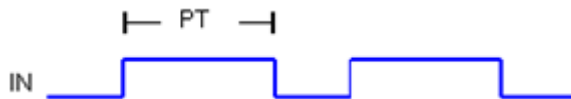
M_T измеряет длительность импульса на входе IN (интервал времени между передним и задним фронтом). По окончании измерения результат выводится на выход PT. В течение измерения выход ET индицирует время, прошедшее с момента детектирования переднего фронта на входе IN.

Если IN=TRUE при вызове ФБ, это не рассматривается как передний фронт.

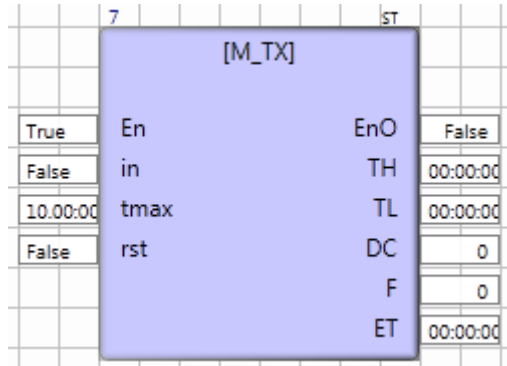
По окончании измерения выходы PT и ET сохраняют свои значения до начала нового измерения.

По команде RST=TRUE измерение прерывается и выходы обнуляются.

Измерение также прерывается и выходы обнуляются, если ET достигает значения TMAX (TMAX задает максимально возможную длительность импульса). По умолчанию TMAX=T#10d.



11.1.19.10. M_TX (OSCAT)



Тип данных входов IN и RST – BOOL, входа TMAX и выходов TH, TL и ET – TIME, выходов DC и F – REAL.

M_TX измеряет следующие характеристики прямоугольного сигнала, поданного на вход IN:

- текущая ширина импульса – выводится на выход TH;
- текущая длительность паузы между импульсами – выводится на выход TL.

Выход ET индицирует время, прошедшее с начала текущего измерения (ET сбрасывается при достижении величины (TH+TL)).

Вход TMAX задает максимальный измеряемый интервал (TH+TL). Значение TMAX по умолчанию – t#10d. Если ET достигает TMAX, выполняется перезапуск ФБ с обнулением выходов.

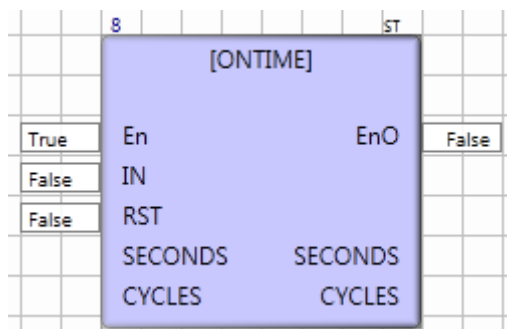
По измеренным величинам вычисляются следующие параметры сигнала:

- частота следования импульсов в герцах ($1/(TH+TL)$) – выводится на выход F. Минимальная измеряемая частота – $1/TMAX$;
- отношение ширины импульса к периоду следования, $TH/(TH+TL)$ (эта величина обратна скважности сигнала) – выводится на выход DC.

Если RST=TRUE, выходы ФБ равны 0.

Используемые функции: T_PLC_MS (OSCAT).

11.1.19.11. ONTIME (OSCAT)



Тип данных входов IN и RST – BOOL, входов-выходов SECONDS и CYCLES – UDINT.

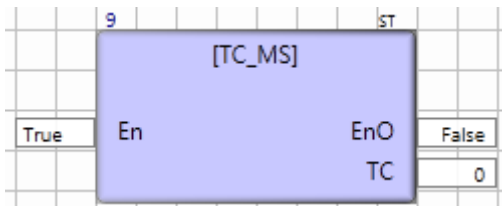
ONTIME суммирует время, в течение которого IN=TRUE. Текущее значение суммы (от 1с до 136 лет) выводится как число секунд на вход-выход SECONDS. Отсчет начинается с момента старта или с момента последнего RST=FALSE (при RST=TRUE входы-выходы равны 0).

Число передних фронтов на входе IN за время вычисления выводится на вход-выход CYCLES.

В программе, использующей ФБ ONTIME, переменные, связанные с входами-выходами SECONDS и CYCLES ФБ, могут быть объявлены сохраняемыми в энергонезависимой памяти контроллера (с помощью модификатора RETAIN оператора VAR); в этом случае после сбоя питания вычисления начнутся с сохраненных величин.

Используемые функции: T_PLC_MS (OSCAT).

11.1.19.12. TC_MS (OSCAT)

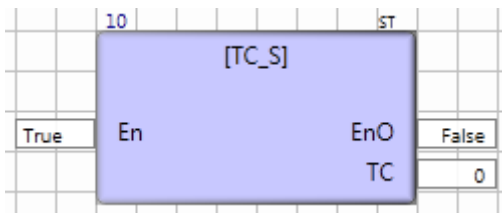


Тип данных выхода TC – DWORD.

ФБ возвращает длительность последнего цикла как целое число миллисекунд (точность – 1 миллисекунда).

Используемые функции: T_PLC_MS (OSCAT).

11.1.19.13. TC_S (OSCAT)

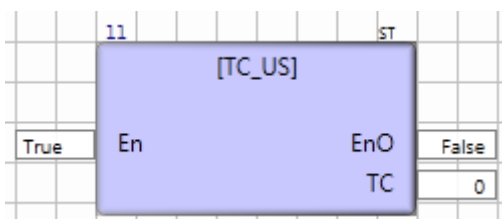


Тип данных выхода TC – REAL.

ФБ возвращает длительность последнего цикла как float-число секунд (точность – 1 миллисекунда).

Используемые функции: T_PLC_MS (OSCAT).

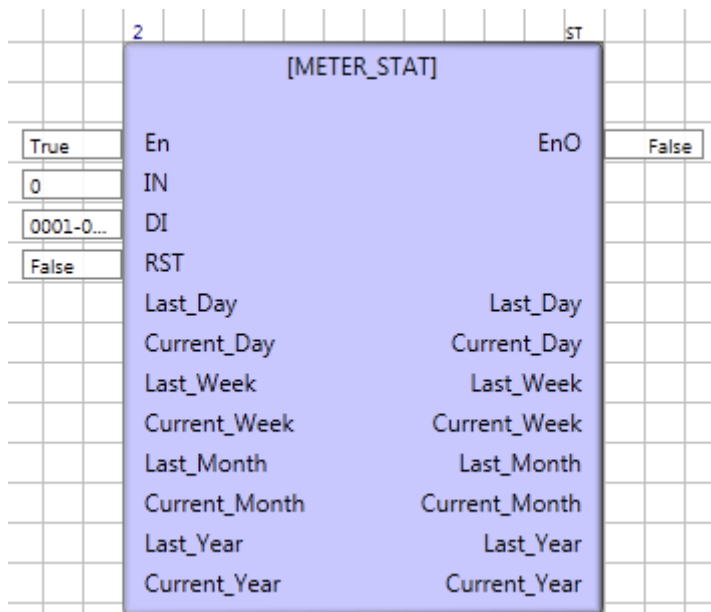
11.1.19.14. TC_US (OSCAT)



Тип данных выхода TC – DWORD.

ФБ TC_US использует функцию T_PLC_US (OSCAT), и в текущей редакции этой функции возвращает длительность последнего цикла как целое число микросекунд с точностью 1 миллисекунда.

11.1.19.15. METER_STAT (OSCAT)



Входы ФБ:

- IN (REAL) – вход;
- DI (DATE) – дата;
- RST (BOOL) – сброс.

Входы-выходы ФБ:

- LAST_DAY (REAL) – расход за предыдущий день;
- CURRENT_DAY (REAL) – расход с начала текущего дня;
- LAST_WEEK (REAL) – расход за предыдущую неделю;
- CURRENT_WEEK (REAL) – расход с начала текущей недели;
- LAST_MONTH (REAL) – расход за предыдущий месяц;
- CURRENT_MONTH (REAL) – расход с начала текущего месяца;
- LAST_YEAR (REAL) – расход за предыдущий год;
- CURRENT_YEAR (REAL) – расход с начала текущего года.

METER_STAT вычисляет расход за текущий день, неделю, месяц и год, а также индицирует расход за соответствующий предыдущий период. Общий суммарный расход подается на вход IN, дата – на вход DI.

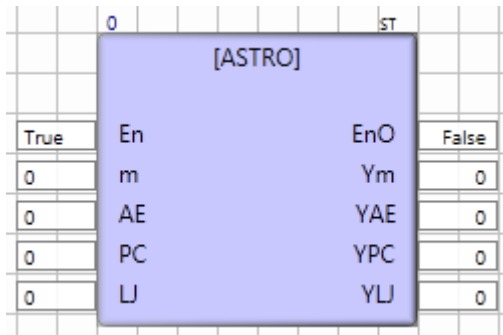
Чтобы перезапустить ФБ, нужно выполнить последовательность команд RST:=TRUE, RST:=FALSE.

Необходимые параметры ФБ сохраняются в энергонезависимой памяти контроллера.

Используемые функции: YEAR_OF_DATE (OSCAT), MONTH_OF_DATE (OSCAT), DAY_OF_YEAR (OSCAT), DAY_OF_WEEK (OSCAT).

11.1.20. OSCAT.ФБ.ПРЕОБРАЗОВАНИЯ

11.1.20.1. ASTRO (OSCAT)



Входы ФБ:

- M (REAL) – расстояние в метрах;
- AE (REAL) – расстояние в астрономических единицах (а.е.);
- PC (REAL) – расстояние в парсеках;
- LJ (REAL) – расстояние в световых годах.

Выходы ФБ:

- YM (REAL) – расстояние в метрах;
- YAE (REAL) – расстояние в астрономических единицах (а.е.);
- YPC (REAL) – расстояние в парсеках;
- YLJ (REAL) – расстояние в световых годах.

ASTRO конвертирует значения входов в а.е., суммирует полученные величины и выводит сумму, конвертированную соответствующим образом, на выходы.

$$1 \text{ а.е.} = 149,597870 * 10^9 \text{ м}$$

$$1 \text{ световой год} = 9,460530 * 10^{15} \text{ м} = 63240 \text{ а.е.} = 0,30659 \text{ парсека}$$

$$1 \text{ парсек} = 206265 \text{ а.е.} = 30,857 * 10^{15} \text{ м} = 3,261622 \text{ световых года}$$

11.1.20.2. ENERGY (OSCAT)

1		[ENERGY]		ST
True	En	EnO	False	
0	J	YJ	0	
0	C	YC	0	
0	Wh	YWh	0	

Входы ФБ:

- J (REAL) – энергия в джоулях (Дж);
- C (REAL) – энергия в калориях (кал);
- WH (REAL) – энергия в ватт-часах (Вт*ч).

Выходы ФБ:

- YJ (REAL) – энергия в джоулях (Дж);
- YC (REAL) – энергия в калориях (кал);
- YWH (REAL) – энергия в ватт-часах (Вт*ч).

ENERGY конвертирует значения входов в джоули, суммирует полученные величины и выводит сумму, конвертированную соответствующим образом, на выходы.

$$1 \text{ Дж} = 1 \text{ Вт} \cdot \text{с} = 1 \text{ н} \cdot \text{м} \text{ (ньютон} \cdot \text{метр)}$$

$$1 \text{ кал} = 4,1868 \text{ Дж} = 1,163 \cdot 10^{-3} \text{ Вт} \cdot \text{ч}$$

$$1 \text{ Вт} \cdot \text{ч} = 3,6 \cdot 10^3 \text{ Дж} = 860 \text{ кал}$$

11.1.20.3. LENGTH (OSCAT)

2		[LENGTH]		ST
True	En	EnO	False	
0	m	Ym	0	
0	p	Yp	0	
0	in	Yin	0	
0	ft	Yft	0	
0	yd	Yyd	0	
0	mile	Ymile	0	
0	sm	Ysm	0	
0	fm	Yfm	0	

Входы ФБ:

- M (REAL) – длина в метрах (м);
- P (REAL) – длина в типографских точках (пт);
- IN (REAL) – длина в дюймах;
- FT (REAL) – длина в футах;
- YD (REAL) – длина в ярдах;
- MILE (REAL) – длина в милях;
- SM (REAL) – длина в морских милях;
- FM (REAL) – длина в морских саженьях.

Выходы ФБ:

- YM (REAL) – длина в метрах (м);
- YP (REAL) – длина в типографских точках (пт);
- YIN (REAL) – длина в дюймах;
- YFT (REAL) – длина в футах;
- YYD (REAL) – длина в ярдах;
- YMILE (REAL) – длина в милях;
- YSM (REAL) – длина в морских милях;
- YFM (REAL) – длина в морских саженьях.

ФБ конвертирует значения входов в метры, суммирует полученные величины и выводит сумму, конвертированную соответствующим образом, на выходы.

1 пт = 0.376065 мм (типографская единица измерения)

1 дюйм = 25,4 мм

1 фут = 0,3048 м

1 ярд = 0,9144 м

1 миля = 1609,344 м

1 мор. миля = 1852 м

1 мор. сажень = 1,829 м

11.1.20.4. PRESSURE (OSCAT)

3		ST	
[PRESSURE]			
True	En	EnO	False
0	mws	Ymws	0
0	torr	Ytorr	0
0	att	Yatt	0
0	atm	Yatm	0
0	pa	Ypa	0
0	bar	Ybar	0

Входы ФБ:

- MWS (REAL) – давление в метрах водяного столба (м вод.ст.);
- TORR (REAL) – давление в миллиметрах ртутного столба (мм рт.ст.);
- ATT (REAL) – давление в технических атмосферах (тех.атм);
- ATM (REAL) – давление в физических атмосферах (физ.атм);
- PA (REAL) – давление в паскалях (Па);
- BAR (REAL) – давление в барах (бар).

Выходы ФБ:

- YMWS (REAL) – давление в метрах водяного столба (м вод.ст.);
- YTORR (REAL) – давление в миллиметрах ртутного столба (мм рт.ст.);
- YATT (REAL) – давление в технических атмосферах (тех.атм);
- YATM (REAL) – давление в физических атмосферах (физ.атм);
- YPA (REAL) – давление в паскалях (Па);
- YBAR (REAL) – давление в барах (бар).

ФБ конвертирует значения входов в бары, суммирует полученные величины и выводит сумму, конвертированную соответствующим образом, на выходы.

1 м вод.ст. = 0.0980665 бар

1 Торр = 1 мм.рт.ст. = 101325/760 Па

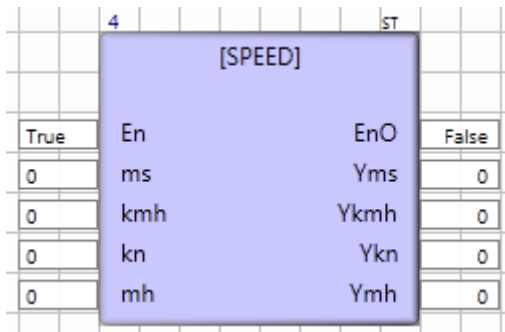
1 тех.атм = 1 кПа/см² = 0.980665 бар

1 физ.атм = 1.01325 бар

1 Па = 1 н/м²

1 бар = 105 Па

11.1.20.5. SPEED (OSCAT)



Входы ФБ:

- MS (REAL) – скорость в метрах в секунду (м/с);
- КМН (REAL) – скорость в километрах в час (км/ч);
- КН (REAL) – скорость в узлах (1 узел=1 мор.миля/ч);
- МН (REAL) – скорость в милях в час.

Выходы ФБ:

- YMS (REAL) – скорость в метрах в секунду (м/с);
- YКМН (REAL) – скорость в километрах в час (км/ч);
- YКН (REAL) – скорость в узлах (1 узел=1 мор.миля/ч);
- YМН (REAL) – скорость в милях в час.

ФБ конвертирует значения входов в м/с, суммирует полученные величины и выводит сумму, конвертированную соответствующим образом, на выходы.

$$1 \text{ м/с} = 3.6 \text{ км/ч}$$

$$1 \text{ узел} = 0.5144 \text{ м/с}$$

$$1 \text{ миля/ч} = 0.44704 \text{ м/с}$$

11.1.20.6. TEMPERATURE (OSCAT)

S		[TEMPERATURE]		ST	
True	En		EnO	False	
0	K		YK	0	
-273.15	C		YC	0	
-459.67	F		YF	0	
-218.52	Re		YRe	0	
0	Ra		YRa	0	

Входы ФБ:

- K (REAL) – температура в градусах Кельвина (K);
- C (REAL) – температура в градусах Цельсия (0C);
- F (REAL) – температура по шкале Фаренгейта (0F);
- RE (REAL) – температура по шкале Реомюра (Re);
- RA (REAL) – температура по шкале Рэнкина (Ra).

Выходы ФБ:

- YK (REAL) – температура в градусах Кельвина (K);
- YC (REAL) – температура в градусах Цельсия (0C);
- YF (REAL) – температура по шкале Фаренгейта (0F);
- YRE (REAL) – температура по шкале Реомюра (Re);
- YRA (REAL) – температура по шкале Рэнкина (Ra).

ФБ конвертирует значения входов в градусы Кельвина, суммирует полученные величины и выводит сумму, конвертированную соответствующим образом, на выходы.

$$T K = T 0C - 273.15$$

$$T 0F = 1.8 * T 0C + 32$$

$$T Re = 0.8 * T 0C$$

$$T Ra = 1.8 * T K$$

11.1.21. OSCAT.ФБ.МОДУЛИ РЕГУЛИРОВАНИЯ

11.1.21.1. CONTROL_SET1 (OSCAT)

0	[CONTROL_SET1]		ST
True	En	EnO	False
0	Kt	KP	0
0	Tt	TN	0
False	PI	TV	0
False	PID	KI	0
0.5	P K	KD	0
0.45	PI K		
0.83	PI TN		
0.6	PID K		
0.5	PID TN		
0.125	PID TV		

Входы ФБ:

- КТ (REAL) – критический КР;
- ТТ (REAL) – период критической вибрации;
- PI (BOOL) и PID (BOOL) – устанавливаются в соответствии с типом регулятора, для которого определяются параметры:
- П – PI=PID=0;
- ПИ – PI=1, PID=0;
- ПИД – PI=0, PID=1;
- P_K (REAL) – константа (0.5 по умолчанию) для вычисления КР П-регулятора;
- PI_K (REAL) – константа (0.45 по умолчанию) для вычисления КР ПИ-регулятора;
- PI_TN (REAL) – константа (0.83 по умолчанию) для вычисления TN ПИ-регулятора;
- PID_K (REAL) – константа (0.6 по умолчанию) для вычисления КР ПИД-регулятора;
- PID_TN (REAL) – константа (0.5 по умолчанию) для вычисления TN ПИД-регулятора;
- PID_TV (REAL) – константа (0.125 по умолчанию) для вычисления TV ПИД-регулятора.

Выходы ФБ:

- КР (REAL) – КР;

- $TN (REAL) - TN$;
- $TV (REAL) - TV$;
- $KI (REAL) - KI$;
- $KD (REAL) - KD$.

ФБ CONTROL_SET1 вычисляет настроечные параметры П-, ПИ- и ПИД-регуляторов по методу Циглера-Николса (Ziegler-Nichols) для замкнутых контуров управления.

В соответствии с этим методом, регулятор переводится в режим П-регулятора (например, с помощью обнуления коэффициентов интегральной и дифференциальной составляющих), в контуре управления создается малое возмущение (например, путем изменения уставки), а затем коэффициент пропорциональной части KP регулятора линейно увеличивается от 0 до тех пор, пока при некотором критическом $KP=KT$ на выходе регулятора не возникают вибрации с постоянной амплитудой и периодом TT . По KT и TT ФБ вычисляет настроечные параметры регулятора:

Тип регулятора	KP	TN	TV
П	P_K*KT		
ПИ	PI_K*KT	PI_TN*TT	
ПИД	PID_K*KT	PID_TN*TT	PID_TV*TT

Если используются константы по умолчанию, вычисления производятся по следующим формулам:

Тип регулятора	KP	TN	TV
П	$0.5*KT$		
ПИ	$0.45*KT$	$0.83*TT$	
ПИД	$0.6*KT$	$0.5*TT$	$0.125*TT$

Если $TN > 0$, то $KI = KP / TN$, в противном случае $KI = 0$.

KD вычисляется по формуле $KD = KP * TV$.

Метод Циглера-Николса для замкнутых контуров применим не ко всем регуляторам, а также занимает длительное время в случае медленных регуляторов.

11.1.21.2. CONTROL_SET2 (OSCAT)

True	En	EnO	False
0	KS	KP	0
0	TU	TN	0
0	TG	TV	0
False	PI	KI	0
False	PID	KD	0
1	P K		
0.9	PI K		
3.33	PI TN		
1.2	PID K		
2	PID TN		
0.5	PID TV		

Входы ФБ:

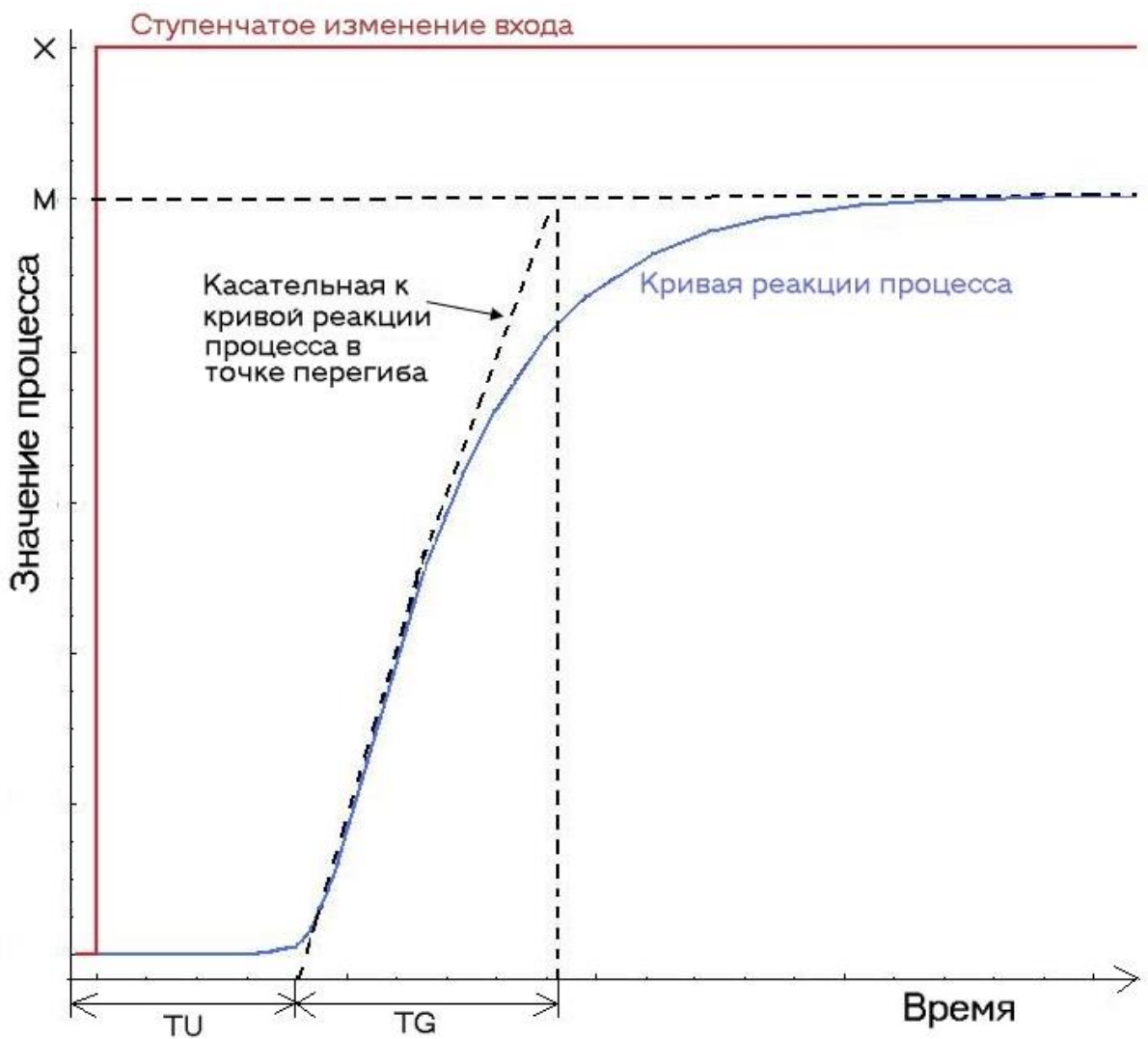
- KS (REAL), TU (REAL) и TG (REAL) – параметры, определяемые по кривой реакции процесса (см. рис. ниже);
- PI (BOOL) и PID (BOOL) – устанавливаются в соответствии с типом регулятора, для которого определяются параметры:
- П – PI=PID=0;
- ПИ – PI=1, PID=0;
- ПИД – PI=0, PID=1;
- P_K (REAL) – константа (1 по умолчанию) для вычисления KP П-регулятора;
- PI_K (REAL) – константа (0.9 по умолчанию) для вычисления KP ПИ-регулятора;
- PI_TN (REAL) – константа (3.33 по умолчанию) для вычисления TN ПИ-регулятора;
- PID_K (REAL) – константа (1.2 по умолчанию) для вычисления KP ПИД-регулятора;
- PID_TN (REAL) – константа (2 по умолчанию) для вычисления TN ПИД-регулятора;
- PID_TV (REAL) – константа (0.5 по умолчанию) для вычисления TV ПИД-регулятора.

Выходы ФБ:

- $KP (REAL) - KP$;
- $TN (REAL) - TN$;
- $TV (REAL) - TV$;
- $KI (REAL) - KI$;
- $KD (REAL) - KD$.

ФБ CONTROL_SET2 вычисляет настроечные параметры П-, ПИ- и ПИД-регуляторов по методу Циглера-Николса (Ziegler-Nichols) для открытых контуров управления (этот метод называется также методом реакции процесса (Process Reaction Method)).

В соответствии с этим методом, по кривой реакции процесса определяются параметры TU , TG и $KS=M/X$:



По TU , TG и KS ФБ вычисляет настроечные параметры регулятора:

Тип регулятора	KP	TN	TV
П	$P_K * TG / TU / KS$		
ПИ	$PI_K * TG / TU / KS$	$PI_TN * TU$	
ПИД	$PID_K * TG / TU / KS$	$PID_TN * TU$	$PID_TV * TU$

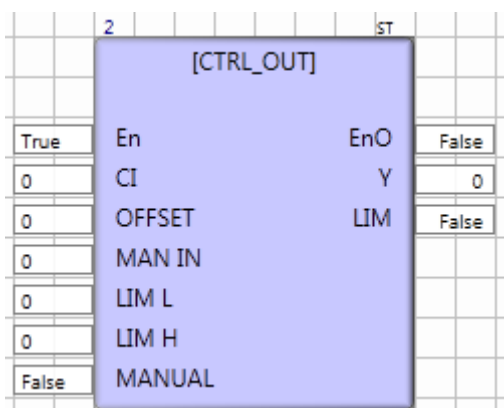
Если используются константы по умолчанию, вычисления производятся по следующим формулам:

Тип регулятора	KP	TN	TV
П	$1 * TG / TU / KS$		
ПИ	$0.9 * TG / TU / KS$	$3.33 * TU$	
ПИД	$1.2 * TG / TU / KS$	$2 * TU$	$0.5 * TU$

Если $TN > 0$, то $KI = KP / TN$, в противном случае $KI = 0$.

KD вычисляется по формуле $KD = KP * TV$.

11.1.21.3. CTRL_OUT (OSCAT)



Входы ФБ:

- CI (REAL) – входной сигнал (сигнал регулятора);
- OFFSET (REAL) – смещение выхода;
- MAN_IN (REAL) – вход ручного управления;

- LIM_L (REAL) – нижний предел выхода;
- LIM_H (REAL) – верхний предел выхода;
- MANUAL (BOOL) – вход разрешения (TRUE) или запрета (FALSE) ручного управления.

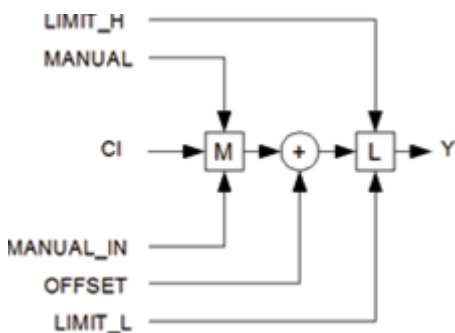
Выходы ФБ:

- Y (REAL) – выходной (управляющий) сигнал;
- LIM (BOOL) – этот выход принимает значение TRUE, если выходной сигнал достигает предела.

CTRL_OUT прибавляет к значению входа CI значение OFFSET и присваивает результат выходу Y, если MANUAL = FALSE. Если MANUAL=TRUE, выходу Y присваивается значение (MAN_IN+OFFSET).

Y всегда ограничивается значениями LIM_L и LIM_H. Если Y достигает одной из границ, выход LIM принимает значение TRUE.

Функциональная схема CTRL_OUT:



11.1.21.4. CTRL_PI (OSCAT)

3		ST	
[CTRL_PI]			
True	En	EnO	False
0	ACT	Y	0
0	SET	DIFF	0
0	SUP	LIM	False
0	OFS		
0	M_I		
False	MAN		
False	RST		
1	KP		
1	KI		
-1000	LL		
1000	LH		

Входы ФБ:

- ACT (REAL) – вход (текущее значение процесса);
- SET (REAL) – уставка процесса;
- SUP (REAL) – порог;
- OFS (REAL) – смещение выхода Y;
- M_I (REAL) – значение для ручного режима;
- MAN (BOOL) – переключатель ручного режима;
- RST (BOOL) – сброс значения интеграла;
- KP (REAL) – коэффициент пропорциональной составляющей;
- KI (REAL) – коэффициент интегральной составляющей;
- LL (REAL) – нижний предел выхода Y;
- LH (REAL) – верхний предел выхода Y.

Выходы ФБ:

- Y (REAL) – выход;
- DIFF (REAL) – девиация процесса. Если $(SET-ACT) < -SUP$, $DIFF := SET-ACT+SUP$; если $(SET-ACT) > SUP$, $DIFF := SET-ACT-SUP$; если $-SUP \leq (SET-ACT) \leq SUP$, $DIFF := 0$;
- LIM (BOOL) – TRUE, если выход Y достигает границы.

CTRL_PI – это пропорционально-интегральный регулятор (ПИ-регулятор) с вычислением девиации процесса и ограничением выходного сигнала, а также возможностью ручного управления.

Математически ФБ работает по следующей формуле:

$$Y(t) := KP \cdot DIFF(t) + KI \int_{t_0}^t DIFF(t)dt + OFS$$

Для вычисления девиации процесса ФБ CTRL_PI использует функцию CTRL_IN (OSCAT), для ограничения выхода – ФБ CTRL_OUT (OSCAT), в качестве собственно регулятора – ФБ FT_PIWL (OSCAT):

$Y := CTRL_OUT(FT_PIWL(DIFF, KP, KI, LL, LH, RST), OFS, M_I, LL, LH, MAN)$

где

$DIFF := CTRL_IN(SET, ACT, SUP)$

В ручном режиме (MAN=TRUE) значение Y продолжает вычисляться по приведенной выше формуле, но заменяется на $Y := M_I + OFS$.

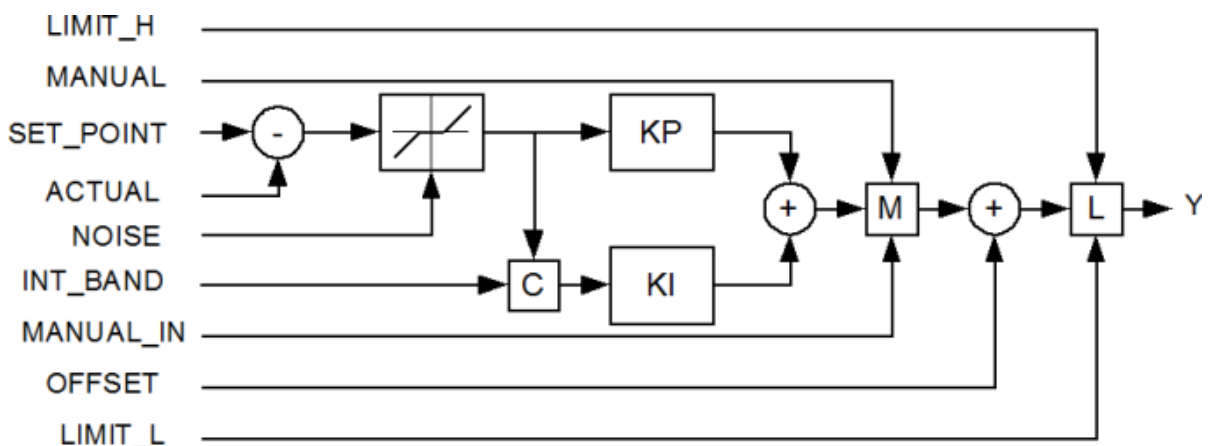
Значение выхода Y всегда ограничивается значениями LL и LH: $LL \leq Y \leq LH$.

Для сброса значения интеграла нужно подать импульс на вход RST.

По умолчанию входы ФБ имеют следующие значения:

$KP=KI=1, LIMIT_L=-1000, LIMIT_H=+1000$

Блок-схема ФБ CTRL_PI:



11.1.21.5. CTRL_PID (OSCAT)

4				ST
[CTRL_PID]				
True	En	EnO	False	
0	ACT	Y	0	
0	SET	DIFF	0	
0	SUP	LIM	False	
0	OFS			
0	M I			
False	MAN			
False	RST			
1	KP			
1	TN			
1	TV			
-1000	LL			
1000	LH			

Входы ФБ:

- ACT (REAL) – значение процесса;
- SET (REAL) – уставка процесса;
- SUP (REAL) – порог;
- SFO (REAL) – смещение выхода Y;
- M_I (REAL) – значение для ручного режима;
- MAN (BOOL) – переключатель ручного режима;
- RST (BOOL) – сброс значения интеграла;
- KP (REAL) – коэффициент усиления ФБ;
- TN (REAL) – этот параметр должен быть отличен от 0. Величина $1/TN$ передается в используемый ПИД-регулятор FT_PIDWL (OSCAT) для формирования в нем коэффициента интегральной части;
- TV (REAL) – этот параметр передается в используемый ПИД-регулятор FT_PIDWL (OSCAT) для формирования в нем коэффициента дифференциальной части;
- LL (REAL) – нижняя граница выхода Y;
- LH (REAL) – верхняя граница выхода Y.

Выходы ФБ:

- Y (REAL) – выход;

- **DIFF (REAL)** – девиация процесса. Если $(SET-ACT) < -SUP$, $DIFF := SET-ACT+SUP$; если $(SET-ACT) > SUP$, $DIFF := SET-ACT-SUP$; если $-SUP \leq (SET-ACT) \leq SUP$, $DIFF := 0$;
- **LIM (BOOL)** – TRUE, если значение выхода Y достигает границы.

CTRL_PID – это пропорционально-интегрально-дифференциальный регулятор (ПИД-регулятор) с вычислением девиации процесса и ограничением выходного сигнала, а также возможностью ручного управления.

Математически ФБ работает по следующей формуле:

$$Y(t) := KP \cdot \left(DIFF(t) + TV \frac{d(DIFF(t))}{dt} + \frac{1}{TN} \int_{t_0}^t DIFF(t) dt \right) + OFS$$

Для вычисления девиации процесса ФБ CTRL_PID использует функцию CTRL_IN (OSCAT), для ограничения выхода – ФБ CTRL_OUT (OSCAT), в качестве собственно регулятора – ФБ FT_PIDWL (OSCAT):

$Y := CTRL_OUT(FT_PIDWL(DIFF, KP, TN, TV, LL, LH, RST), OFS, M_I, LL, LH, MAN)$

где

$DIFF := CTRL_IN(SET, ACT, SUP)$

В ручном режиме ($MAN=TRUE$) значение Y продолжает вычисляться по приведенной выше формуле, но заменяется на $Y := M_I + OFS$.

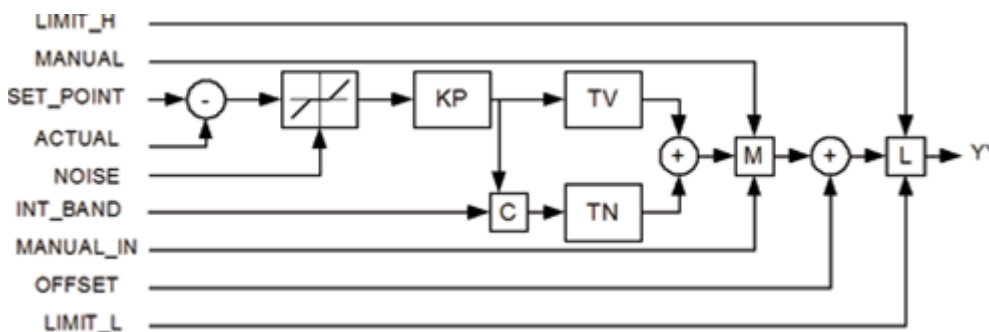
Значение выхода Y всегда ограничивается значениями LL и LH: $LL \leq Y \leq LH$.

Если $TN=0$, интегрирование приостанавливается (значение интеграла сохраняется). Для сброса значения интеграла нужно подать импульс на вход RST.

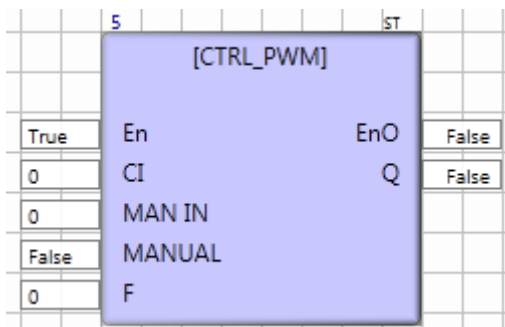
По умолчанию входы ФБ имеют следующие значения:

$KP=TN=TV=1$, $LIMIT_L=-1000$, $LIMIT_H=+1000$

Блок-схема ФБ CTRL_PID:



11.1.21.6. CTRL_PWM (OSCAT)



Входы ФБ:

- CI (REAL) – входной (модулирующий) сигнал; значение CI должно лежать в диапазоне (0, 1);
- MAN_IN (REAL) – входной (модулирующий) сигнал; значение MAN_IN должно лежать в диапазоне (0, 1);
- MANUAL (BOOL) – переключатель входов;
- F (REAL) – частота следования импульсов, Гц.

Выходы ФБ:

- Q (BOOL) – выходной ШИМ-сигнал.

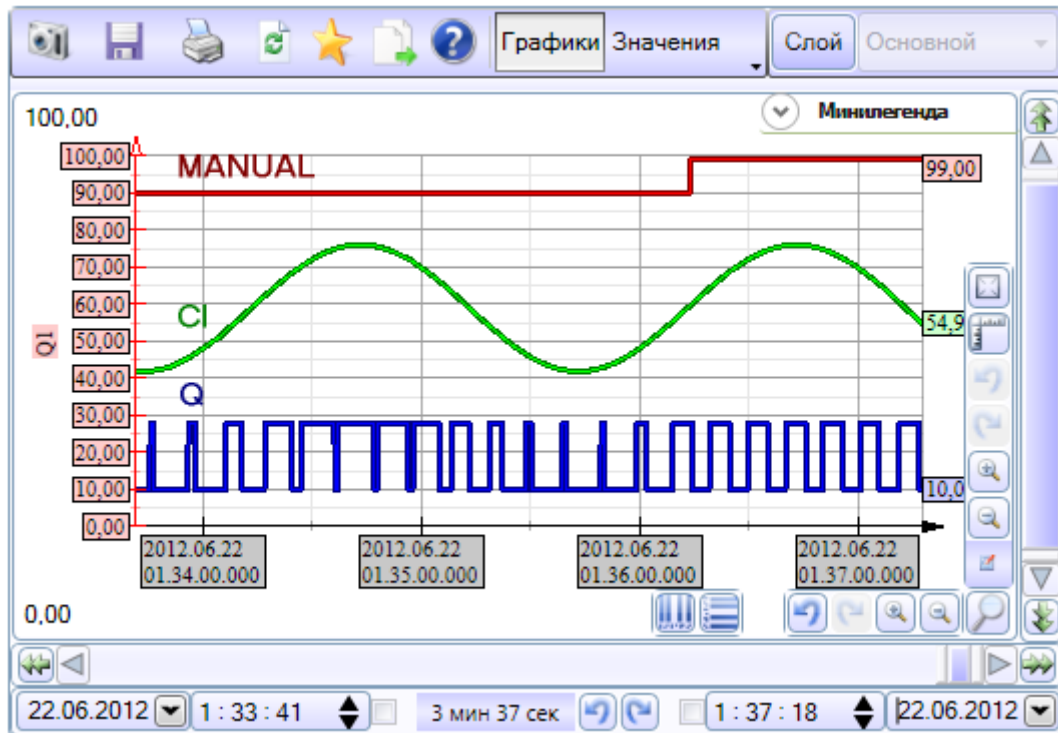
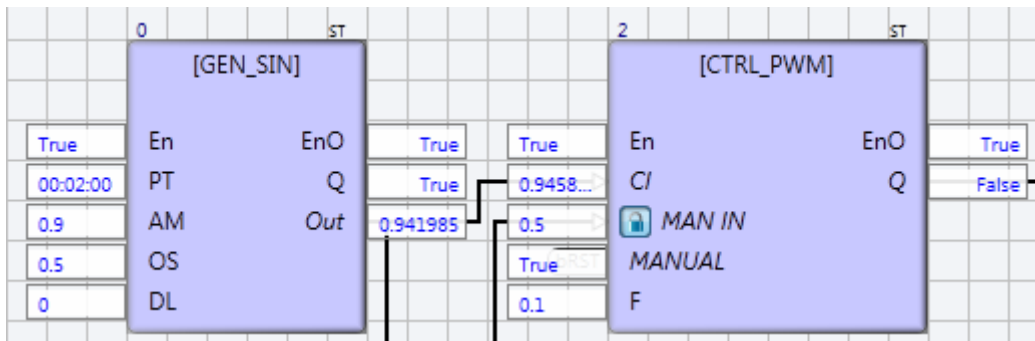
CTRL_PWM генерирует на выходе Q широтно-модулированный импульсный сигнал с частотой следования импульсов F. Для генерации используется ФБ PWM_DC (OSCAT).

Если MANUAL=FALSE, в качестве модулирующего выбирается сигнал CI, и ширина импульсов в этом случае равна CI/F.

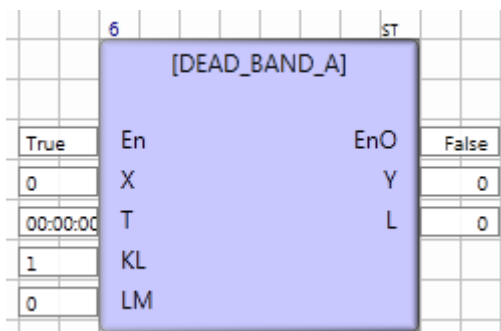
Если MANUAL=TRUE, в качестве модулирующего выбирается сигнал MAN_IN, и ширина импульсов в этом случае равна MAN_IN/F.

Любой из входов CI и MAN_IN может быть использован для ручного задания ширины импульсов.

Следующий пример демонстрирует широтно-импульсную модуляцию синусоидальным сигналом, подаваемым на вход CI, и переход в ручной режим по команде MANUAL=TRUE (в примере MAN_IN=0.5):



11.1.21.7. DEAD_BAND_A (OSCAT)



Входы ФБ:

- X (REAL) – входной сигнал;
- T (TIME) – постоянная времени низкочастотного фильтра;
- KL (REAL) – коэффициент усиления фильтра;

- LM (REAL) – максимальная амплитуда высокочастотной составляющей (половина максимально допустимой величины зоны нечувствительности).

Выходы ФБ:

- Y (REAL) – выходной сигнал;
- L (REAL) – половина реальной величины зоны нечувствительности.

$DEAD_BAND_A$ – безударная линейная передаточная функция с адаптивной зоной нечувствительности (L).

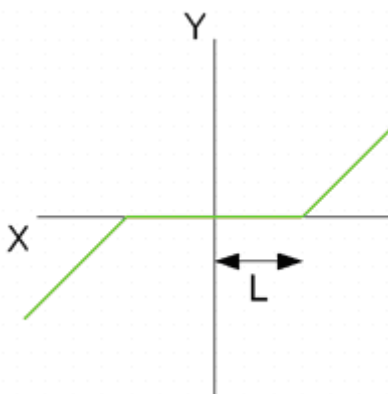
Функция смещает положительную часть кривой на $-L$, а отрицательную – на $+L$. $DEAD_BAND_A$ используется для фильтрации шумов во входном сигнале – например, для предотвращения постоянного переключения контроллера при малых изменениях сигнала, что приводит к перегрузкам и преждевременному старению исполнительного механизма.

Величина L вычисляется из амплитуды высокочастотной составляющей входного сигнала X :

$$L = \langle \text{амплитуда ВЧ} \rangle * KL$$

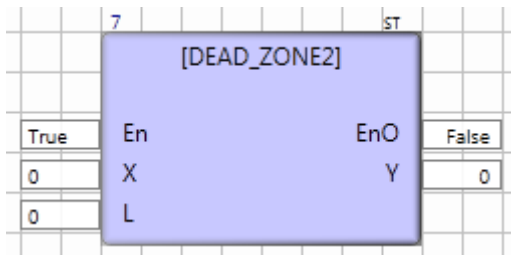
Для выделения ВЧ-составляющей используется фильтр с постоянной времени T . Для изменения чувствительности алгоритма используется множитель KL . Значение KL по умолчанию – 1, допустимые значения KL – от 1 до 5.

Поскольку алгоритм должен оставаться работоспособным даже в экстремальных условиях, величина L ограничивается значением входа LM .



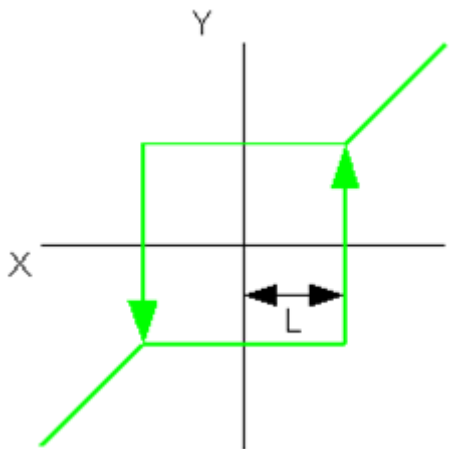
Используемые функции: FT_PT1 (OSCAT)

11.1.21.8. DEAD_ZONE2 (OSCAT)

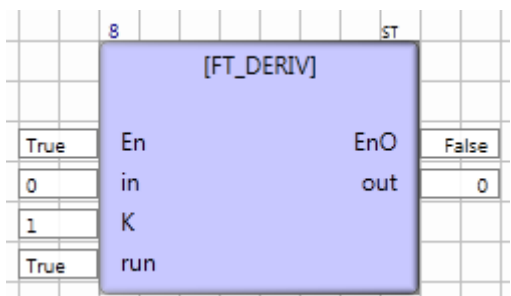


Тип данных выходов X, L и выхода Y – REAL.

DEAD_ZONE2 – ударная линейная передаточная функция с зоной нечувствительности и гистерезисом:



11.1.21.9. FT_DERIV (OSCAT)



Тип данных входов IN, K и выхода OUT – REAL, входа RUN – BOOL.

При RUN=TRUE ФБ вычисляет производную (по двум точкам) от входного сигнала IN и умножает результат на K:

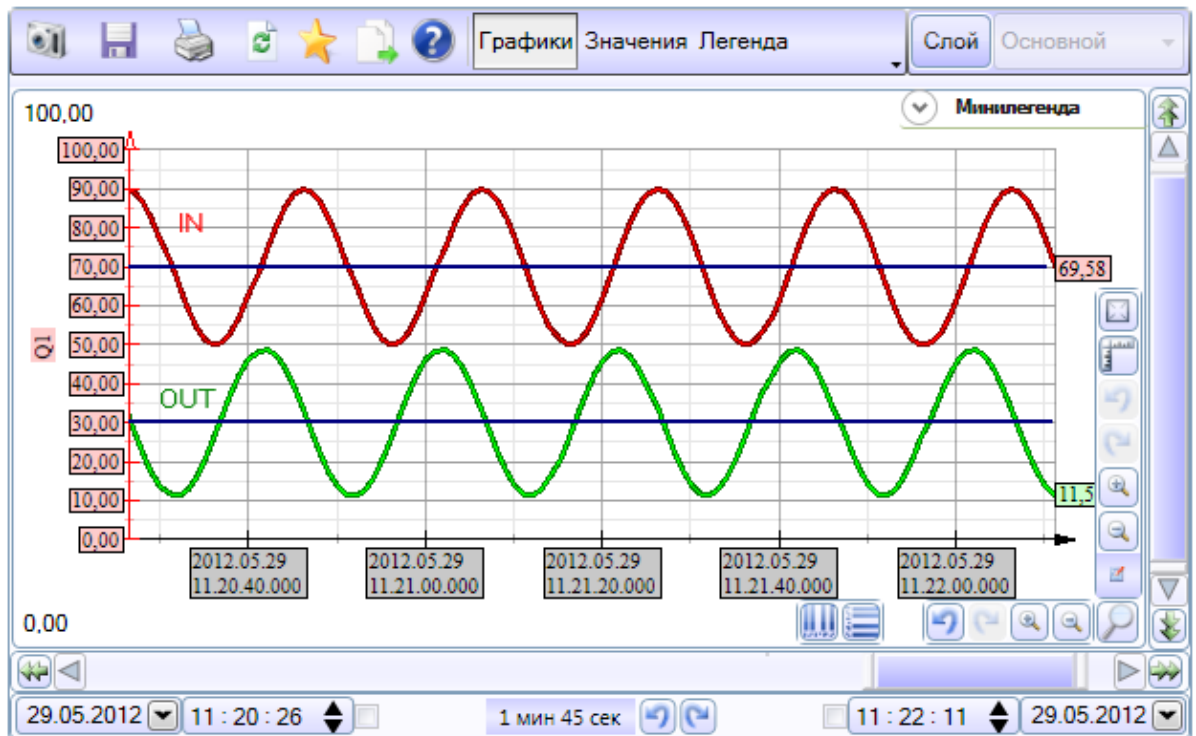
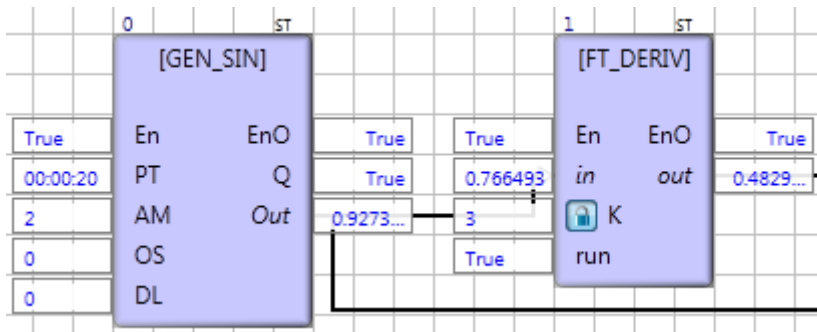
$$\text{out}(t) := K * \text{IN}'(t)$$

При K=1 размерность выхода равна [$\text{размерность IN} / \text{с}$]

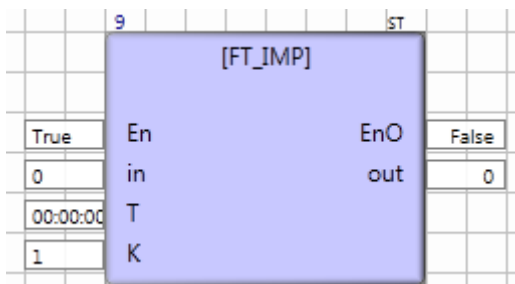
Если RUN=FALSE, OUT:=0.

Используемые функции: T_PLC_MS (OSCAT).

Пример



11.1.21.10. FT_IMP (OSCAT)

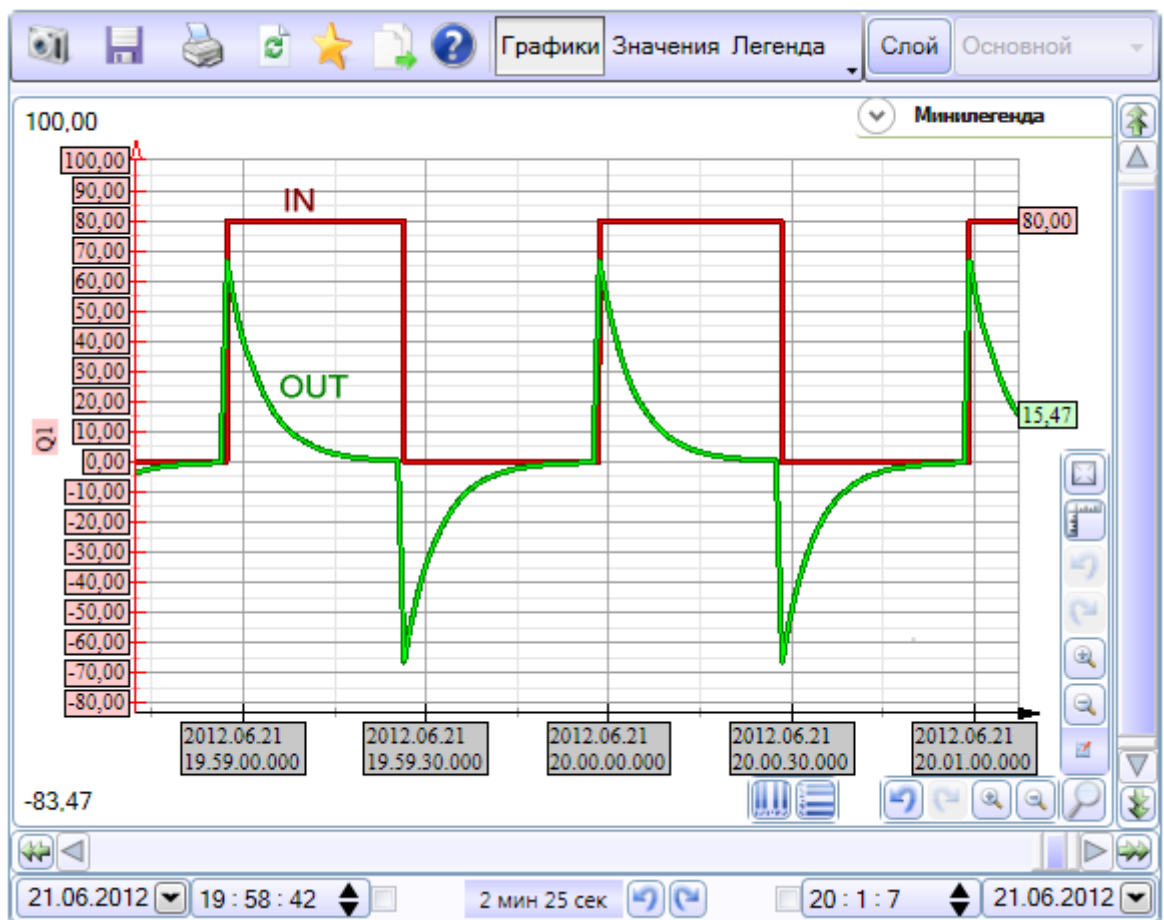
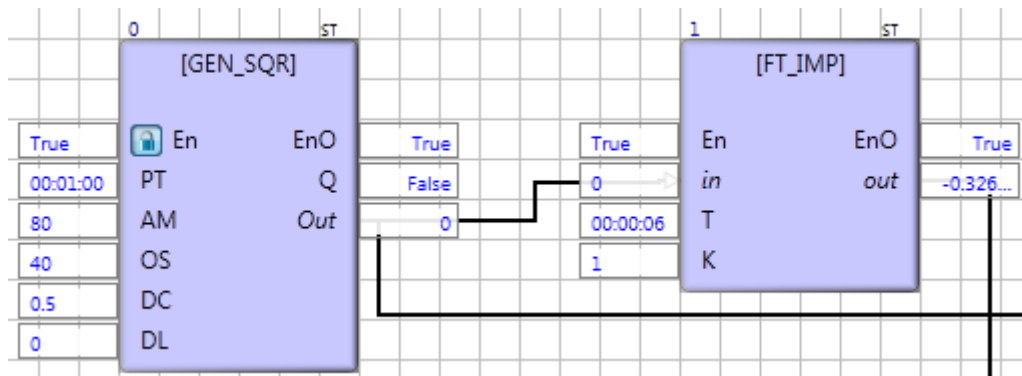


Тип данных входов IN, K и выхода OUT – REAL, выхода OUT – TIME.

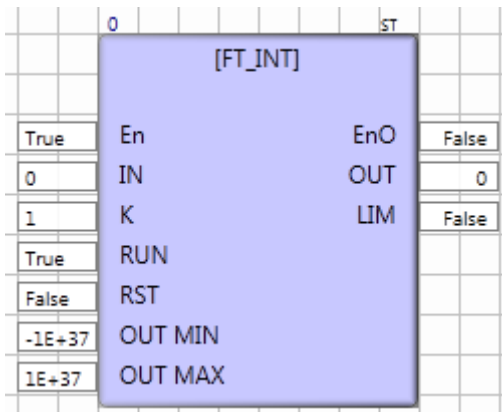
Импульсный (высокочастотный) фильтр FT_IMP работает по следующему алгоритму:

- входной сигнал IN фильтруется с помощью ФБ FT_PT1 (OSCAT);
- вычисляется разность между IN и отфильтрованным IN;
- полученная разность умножается на K и присваивается выходу OUT.

Ниже показан отклик FT_IMP на прямоугольный сигнал (время цикла – 1с, в качестве генератора прямоугольного сигнала использован ФБ GEN_SQR (OSCAT)):



11.1.21.11. FT_INT (OSCAT)



Входы ФБ:

- IN (REAL) – входной сигнал;
- K (REAL) – коэффициент (передается в используемый ФБ INTEGRATE (OSCAT));
- RUN (BOOL) – разрешение/приостановка интегрирования;
- RST (BOOL) – реинициализация алгоритма (сброс значения интеграла);
- OUT_MIN (REAL) – нижняя граница значения интеграла (выхода OUT);
- OUT_MAX (REAL) – верхняя граница значения интеграла (выхода OUT).

Выходы ФБ:

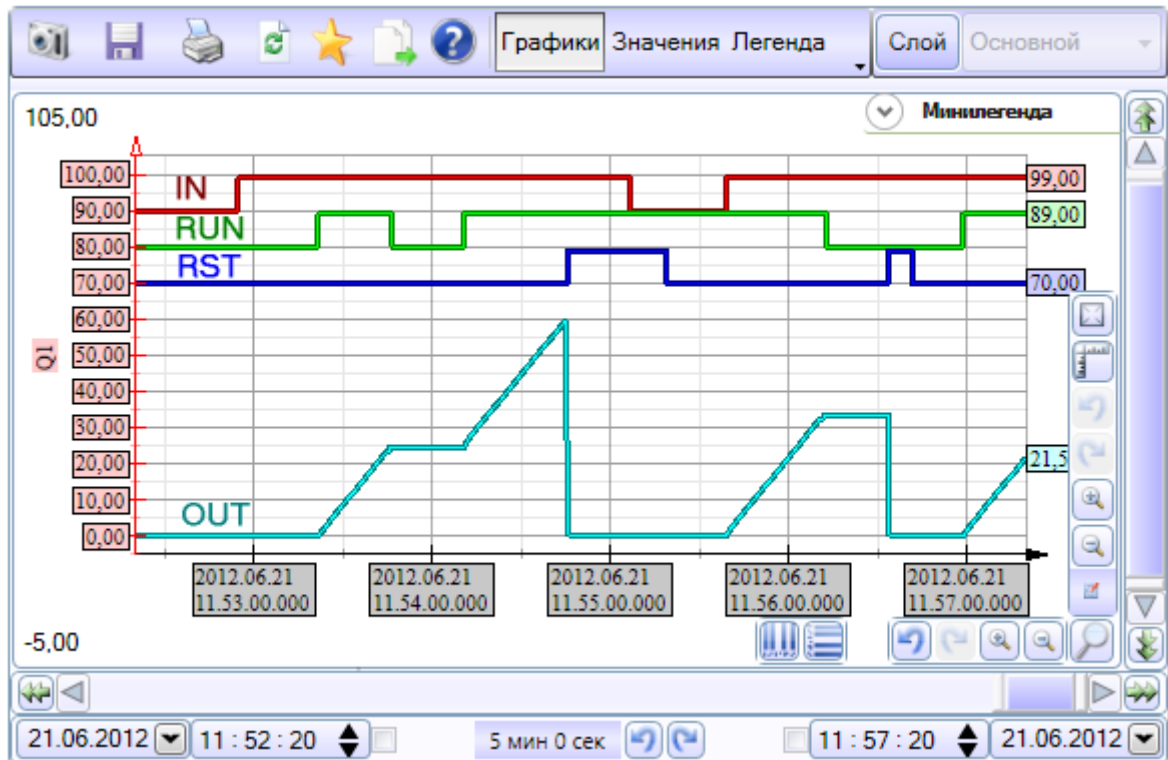
- OUT (REAL) – значение интеграла;
- LIM (BOOL) – TRUE, если значение интеграла достигает границы.

FT_INT интегрирует входной сигнал (IN) с помощью ФБ INTEGRATE (OSCAT).

Определенный интеграл от входного сигнала $IN(t)$ вычисляется с момента времени, когда RUN впервые принимает значение TRUE после старта или реинициализации (если $RUN:=FALSE$ после $RUN:=TRUE$, интегрирование приостанавливается, а OUT сохраняет свое значение, и при последующем возобновлении интегрирования по команде $RUN:=TRUE$ начальное значение интеграла будет равно сохраненному OUT).

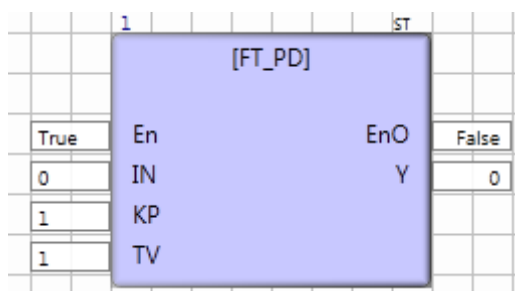
Для реинициализации алгоритма нужно последовательно выполнить команды $RST:=TRUE$ и $RST:=FALSE$ – после этого начальное значение интеграла равно 0, а интегрирование начнется при $RUN:=TRUE$.

Следующая временная диаграмма ФБ FT_INT демонстрирует управление интегрированием с помощью RUN и RST:



Фундаментальная проблема интегратора – его разрешение (точность). Выход типа REAL имеет точность 7-8 разрядов. Пусть на каждом шаге интегрирования прибавляемое значение (шаг) равно 1, а значение интеграла достигло величины более ста миллионов (1E8). В этом случае шаг не может быть прибавлен к значению интеграла, поскольку находится ниже предела точности в 8 разрядов для типа данных REAL. Важно помнить про это ограничение, особенно если FT_INT используется как счетчик коммунальных услуг или в других подобных задачах.

11.1.21.12. FT_PD (OSCAT)



Входы ФБ:

- IN (REAL) – входной сигнал (девиация процесса);
- KP (REAL) – коэффициент усиления ФБ;
- TV (REAL) – коэффициент К используемого ФБ FT_DERIV (OSCAT).

Выходы ФБ:

- Y (REAL) – выходной сигнал.

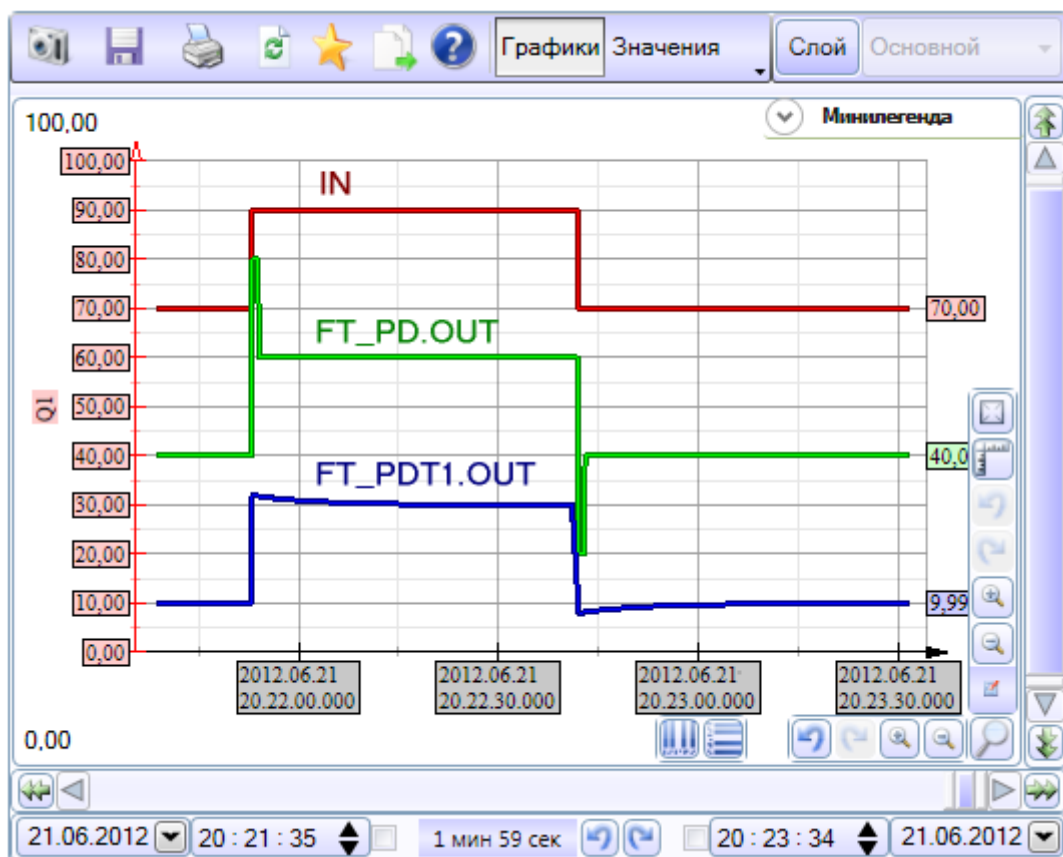
Пропорционально-дифференциальный регулятор (ПД-регулятор) FT_PD работает по следующей формуле:

$$Y := KP * (IN + FT_DERIV(IN, TV))$$

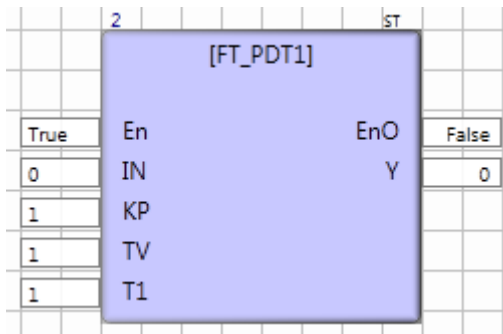
Т.е. коэффициент пропорциональной части равен KP, а коэффициент при производной – KP*TV.

На вход IN подается выходной сигнал функции CTRL_IN (OSCAT), вычисляющей девиацию процесса.

Ниже показаны отклики ФБ FT_PD и, для сравнения, ФБ FT_PDT1 (OSCAT) на прямоугольный импульс:



11.1.21.13. FT_PDT1 (OSCAT)



Данный ФБ представляет собой аналог ФБ FT_PD (OSCAT) с использованием ФБ FT_PT1 (OSCAT) в качестве НЧ-фильтра после вычисления производной:

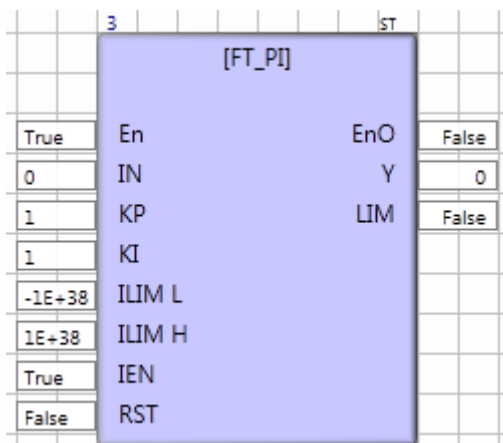
$$Y := KP * (IN + FT_PT1(FT_DERIV(IN,TV), T1))$$

Постоянная времени НЧ-фильтра задается входом T1 как число миллисекунд (тип данных TV1 – REAL).

На вход IN подается выходной сигнал функции CTRL_IN (OSCAT), вычисляющей девиацию процесса.

Используемые функции: FT_DERIV (OSCAT), FT_PT1 (OSCAT).

11.1.21.14. FT_PI (OSCAT)



Входы ФБ:

- IN (REAL) – входной сигнал (девиация процесса);
- KP (REAL) – коэффициент пропорциональной составляющей;
- KI (REAL) – коэффициент интегральной составляющей;
- ILIM_L (REAL) – нижний предел интегратора;
- ILIM_H (REAL) – верхний предел интегратора;
- IEN (BOOL) – разрешение (TRUE) или приостановка (FALSE) работы интегратора;

- RST (BOOL) – реинициализация интегратора.

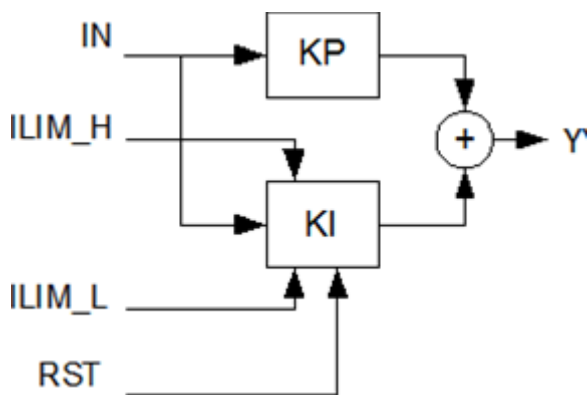
Выходы ФБ:

- Y (REAL) – выходной сигнал;
- LIM (BOOL) – TRUE, если значение интегратора достигает границы.

Пропорционально-интегральный регулятор (ПИ-регулятор) FT_PI работает по следующей формуле (в качестве интегратора используется ФБ FT_INT (OSCAT)):

$$Y := KP*IN + FT_INT(IN, KI, IEN, RST, ILIM_L, ILIM_H)$$

Следующая блок-схема поясняет алгоритм ФБ FT_PI:



Интегратор вычисляет определенный интеграл от входного сигнала $IN(t)$ с момента времени, когда IEN впервые принимает значение TRUE после старта или реинициализации (если IEN:=FALSE после IEN:=TRUE, интегрирование приостанавливается, а выход интегратора сохраняет свое значение, и при последующем возобновлении интегрирования по команде IEN:=TRUE начальное значение интеграла будет равно сохраненному).

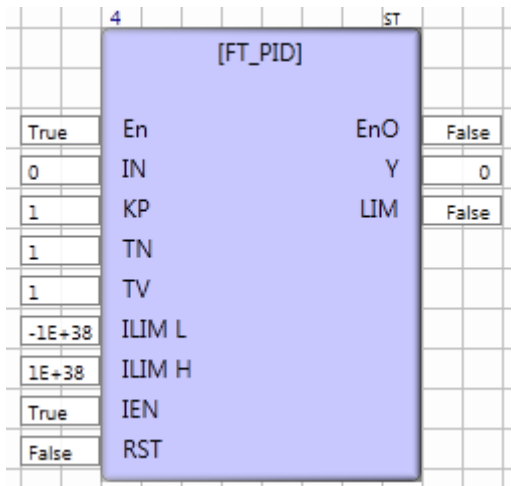
Для реинициализации интегратора нужно последовательно выполнить команды RST:=TRUE и RST:=FALSE – после этого начальное значение интеграла равно 0, а интегрирование начнется при IEN:=TRUE.

На вход IN ФБ FT_PI подается выходной сигнал функции CTRL_IN (OSCAT), вычисляющей девиацию процесса. Для ограничения выходного значения регулятора используется ФБ CTRL_OUT (OSCAT).

По умолчанию входы ФБ имеют следующие значения:

$$KP=1; KI=1; ILIM_L=-1E38; ILIM_H=+1E38; IEN=TRUE$$

11.1.21.15. FT_PID (OSCAT)



Входы ФБ:

- IN (REAL) – входной сигнал (девиация процесса);
- KP (REAL) – коэффициент усиления ФБ;
- TN (REAL) – этот параметр должен быть больше 0. Величина 1/TN передается в качестве коэффициента К в интегратор (в качестве интегратора используется FT_INT (OSCAT));
- TV (REAL) – коэффициент К используемого ФБ FT_DERIV (OSCAT);
- ILM_L (REAL) – нижний предел интегратора;
- ILM_H (REAL) – верхний предел интегратора;
- IEN (BOOL) – разрешение/приостановка работы интегратора;
- RST (BOOL) – реинициализация интегратора (сброс значения интеграла).

Выходы ФБ:

- Y (REAL) – выходной сигнал;
- LIM (BOOL) – TRUE, если значение интегратора достигает границы.

FT_PID – это пропорционально-интегрально-дифференциальный регулятор (ПИД-регулятор), который работает по следующей формуле (в качестве дифференциатора используется ФБ FT_DERIV (OSCAT), в качестве интегратора – ФБ FT_INT (OSCAT)):

$$Y := KP * [IN + FT_DERIV(IN, TV) + FT_INT(IN, 1/TN, IEN, RST, ILM_L, ILM_H)]$$

Т.е. коэффициент пропорциональной части равен KP, коэффициент при производной (KD) – KP*TV, коэффициент при интеграле (KI) – KP/TN.

Интегратор вычисляет определенный интеграл от входного сигнала IN(t) с момента времени, когда IEN впервые принимает значение TRUE после старта или

реинициализации (если $IEN:=FALSE$ после $IEN:=TRUE$, интегрирование приостанавливается, а выход интегратора сохраняет свое значение, и при последующем возобновлении интегрирования по команде $IEN:=TRUE$ начальное значение интеграла будет равно сохраненному).

Для реинициализации интегратора нужно последовательно выполнить команды $RST:=TRUE$ и $RST:=FALSE$ – после этого начальное значение интеграла равно 0, а интегрирование начнется при $IEN:=TRUE$.

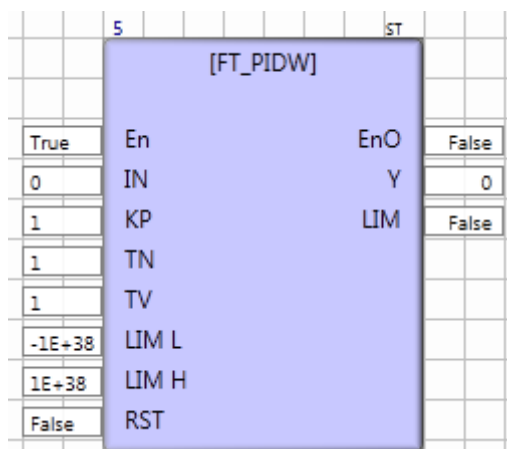
На вход IN ФБ FT_PID подается выходной сигнал функции CTRL_IN (OSCAT), вычисляющей девиацию процесса. Для ограничения выходного значения регулятора используется ФБ CTRL_OUT (OSCAT).

По умолчанию входы ФБ имеют следующие значения:

$KP=TN=TV=1$, $ILIM_L=-1E38$; $ILIM_H=+1E38$; $IEN=TRUE$

Если TN меньше или равно 0, данные в интегратор не передаются, а интегратор продолжает работать с последними сохраненными в нем параметрами IN и $1/TN$. В этом состоянии управление интегратором с помощью IEN и RST невозможно.

11.1.21.16. FT_PIDW (OSCAT)



Входы ФБ:

- IN (REAL) – входной сигнал (девиация процесса);
- KP (REAL) – коэффициент усиления ФБ;
- TN (REAL) – этот параметр должен быть отличен от 0. Величина $1/TN$ передается в качестве коэффициента K в интегратор (в качестве интегратора используется ФБ INTEGRATE (OSCAT));
- TV (REAL) – коэффициент K используемого ФБ FT_DERIV (OSCAT);
- LIM_L (REAL) – нижний предел выхода Y;

- LIM_H (REAL) – верхний предел выхода Y;
- RST (BOOL) – реинициализация интегратора (сброс значения интеграла).

Выходы ФБ:

- Y (REAL) – выходной сигнал;
- LIM (BOOL) – TRUE, если сумма значений пропорциональной и интегральной частей выходит за границы диапазона [LIM_L, LIM_H].

FT_PIDW – это пропорционально-интегрально-дифференциальный регулятор (ПИД-регулятор) с ограничением выхода. ФБ работает по следующей формуле (в качестве дифференциатора используется ФБ FT_DERIV (OSCAT), в качестве интегратора – ФБ INTEGRATE (OSCAT)):

$$Y := KP * (IN + FT_DERIV(IN, TV) + INTEGRATE(IN, 1/TN, NOT LIM))$$

Т.е. коэффициент пропорциональной части равен KP, коэффициент при производной (KD) – KP*TV, коэффициент при интеграле (KI) – KP/TN.

Интегратор вычисляет определенный интеграл от входного сигнала IN(t) с момента старта или реинициализации интегратора. Если LIM=TRUE, интегрирование приостанавливается (выход интегратора сохраняет свое значение). Для сброса значения интеграла нужно выполнить команду RST:=TRUE или TN=0.

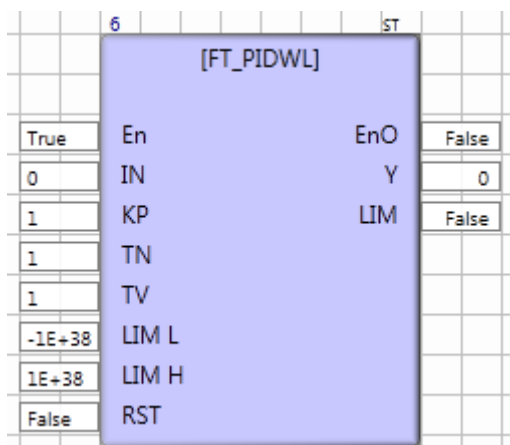
Значение выхода Y ограничивается значениями LIM_L и LIM_H: $LIM_L \leq Y \leq LIM_H$.

На вход IN ФБ FT_PIDW подается выходной сигнал функции CTRL_IN (OSCAT), вычисляющей девиацию процесса.

По умолчанию входы ФБ имеют следующие значения:

$$KP=TN=TV=1, LIM_L=-1E38; LIM_H=+1E38$$

11.1.21.17. FT_PIDWL (OSCAT)



Входы ФБ:

- IN (REAL) – входной сигнал (девиация процесса);
- KP (REAL) – коэффициент усиления ФБ;
- TN (REAL) – этот параметр должен быть отличен от 0. Величина $1/TN$ передается в ПИ-вычислитель для формирования в нем коэффициента интегральной части (в качестве ПИ-вычислителя используется ФБ FT_PIWL (OSCAT));
- TV (REAL) – коэффициент K используемого ФБ FT_DERIV (OSCAT);
- LIM_L (REAL) – нижний предел ПИ-вычислителя и одновременно выхода Y;
- LIM_H (REAL) – верхний предел ПИ-вычислителя и одновременно выхода Y;
- RST (BOOL) – сброс значения интеграла.

Выходы ФБ:

- Y (REAL) – выходной сигнал;
- LIM (BOOL) – TRUE, если значение выхода Y выходит за границы диапазона [LIM_L, LIM_H].

FT_PIDWL – это пропорционально-интегрально-дифференциальный регулятор (ПИД-регулятор) с ограничением выхода. ФБ работает по следующей формуле (в качестве дифференциатора используется ФБ FT_DERIV (OSCAT), в качестве ПИ-вычислителя – ФБ FT_PIWL (OSCAT)):

$$Y := FT_DERIV(IN, KP * TV) + FT_PIWL(IN * KP, 1, 1/TN, LIM_L, LIM_H)$$

Т.е. коэффициент пропорциональной части равен KP, коэффициент при производной (KD) – $KP * TV$, коэффициент при интеграле (KI) – KP / TN .

Если $TN=0$, интегрирование приостанавливается (значение интеграла сохраняется). Для сброса значения интеграла нужно подать импульс на вход RST.

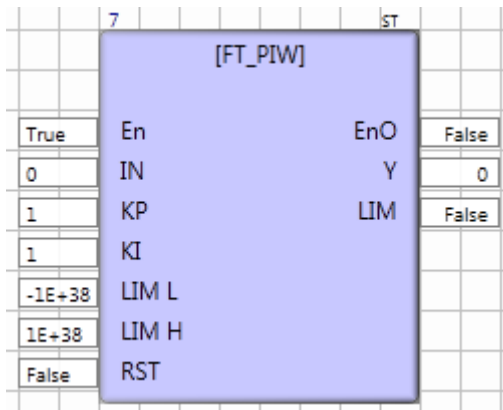
Значение выхода Y ограничивается значениями LIM_L и LIM_H: $LIM_L \leq Y \leq LIM_H$.

На вход IN ФБ FT_PIDWL подается выходной сигнал функции CTRL_IN (OSCAT), вычисляющей девиацию процесса.

По умолчанию входы ФБ имеют следующие значения:

$$KP = TN = TV = 1, LIM_L = -1E38, LIM_H = +1E38$$

11.1.21.18. FT_PIW (OSCAT)



Входы ФБ:

- IN (REAL) – входной сигнал (девиация процесса);
- KP (REAL) – коэффициент пропорциональной составляющей;
- KI (REAL) – коэффициент интегральной составляющей;
- LIM_L (REAL) – нижний предел выхода Y;
- LIM_H (REAL) – верхний предел выхода Y;
- RST (BOOL) – реинициализация интегратора.

Выходы ФБ:

- Y (REAL) – выходной сигнал;
- LIM (BOOL) – TRUE, если значение Y выходит за границы диапазона [LIM_L, LIM_H].

FT_PIW – это пропорционально-интегральный регулятор (ПИ-регулятор) с ограничением выхода. ФБ работает по следующей формуле (в качестве интегратора используется ФБ FT_INT (OSCAT)):

$$Y := KP*IN + FT_INT(IN, KI, NOT LIM, RST)$$

Интегратор вычисляет определенный интеграл от входного сигнала IN(t) с момента старта или реинициализации. Если LIM=TRUE, интегрирование приостанавливается (выход интегратора сохраняет свое значение). Для сброса значения интеграла нужно выполнить команду RST:=TRUE.

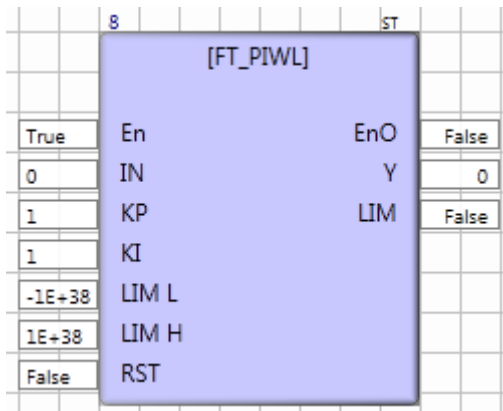
На вход IN ФБ FT_PI подается выходной сигнал функции CTRL_IN (OSCAT), вычисляющей девиацию процесса.

Значение выхода Y ограничивается значениями LIM_L и LIM_H: $LIM_L \leq Y \leq LIM_H$.

По умолчанию входы ФБ имеют следующие значения:

KP=1; KI=1; LIM_L=-1E38; LIM_H=+1E38

11.1.21.19. FT_PIWL (OSCAT)



Входы ФБ:

- IN (REAL) – входной сигнал (девиация процесса);
- KP (REAL) – коэффициент пропорциональной составляющей;
- KI (REAL) – коэффициент интегральной составляющей;
- LIM_L (REAL) – нижний предел выхода Y;
- LIM_H (REAL) – верхний предел выхода Y;
- RST (BOOL) – сброс значения интеграла.

Выходы ФБ:

- Y (REAL) – выходной сигнал;
- LIM (BOOL) – TRUE, если значение Y достигает границы.

FT_PIWL – это пропорционально-интегральный регулятор (ПИ-регулятор) с ограничением выхода. ФБ работает по формуле $Y := KP * IN + I$, где интегральная составляющая I вычисляется методом трапеций (n – номер цикла, t – время внутреннего таймера ПЛК в миллисекундах (используется функция T_PLC_MS (OSCAT))):

$$I_n := I_{n-1} + KI * \frac{IN_n + IN_{n-1}}{2 * 1000} * (t_n - t_{n-1})$$

Т.е. коэффициент пропорциональной части равен KP, коэффициент при интеграле – KI.

Определенный интеграл от входного сигнала IN(t) вычисляется с момента старта или сброса значения интеграла. Для сброса значения интеграла нужно последовательно выполнить команды RST=TRUE и RST=FALSE (т.е. значение интеграла сбрасывается импульсом).

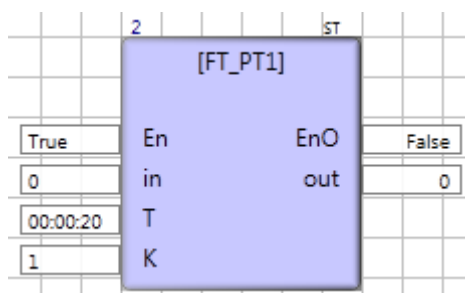
Значение выхода Y ограничивается значениями LIM_L и LIM_H : $LIM_L \leq Y \leq LIM_H$. Если Y достигает границы, значение интегральной части I ограничивается соответственно значениями $LIM_H - KP * IN$ и $LIM_L - KP * IN$.

На вход IN ФБ FT_PIWL подается выходной сигнал функции $CTRL_IN$ ($OSCAT$), вычисляющей девиацию процесса.

По умолчанию входы ФБ имеют следующие значения:

$KP=1$; $KI=1$; $LIM_L=-1E38$; $LIM_H=+1E38$

11.1.21.20. FT_PT1 (OSCAT)



Входы ФБ:

- IN (REAL) – входной сигнал;
- T (TIME) – постоянная времени;
- K (REAL) – коэффициент усиления.

Выходы ФБ:

- OUT (REAL) – выходной сигнал.

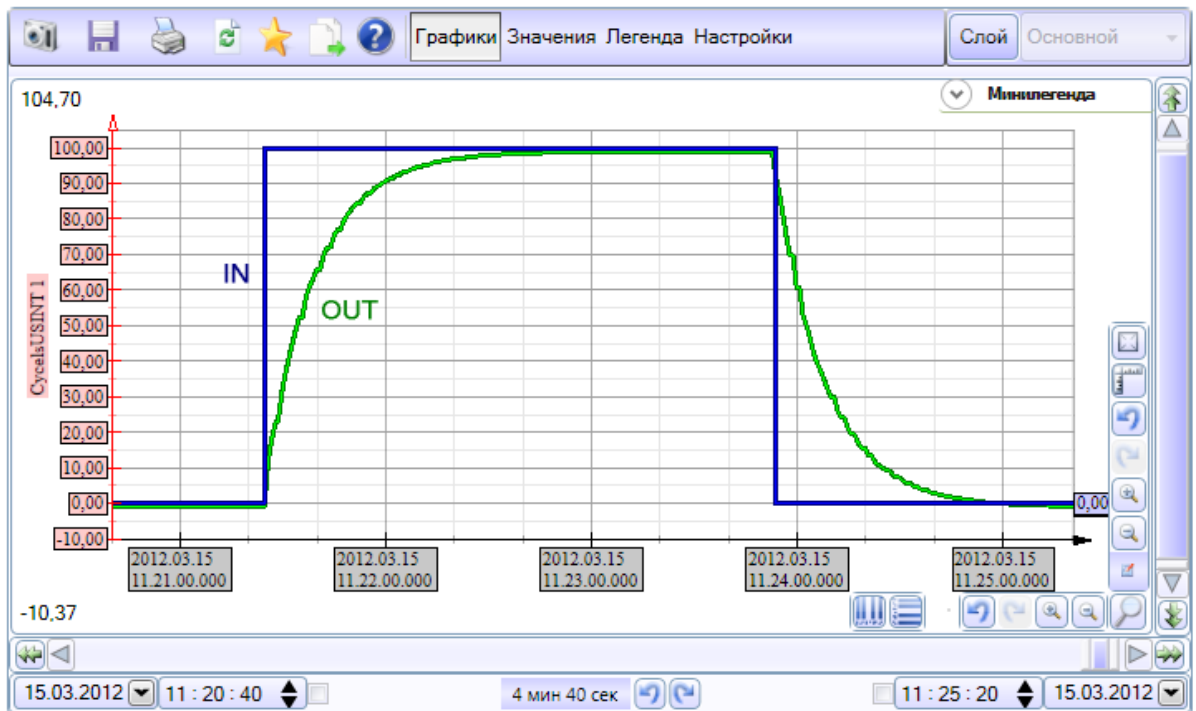
FT_PT1 – это низкочастотный фильтр 1-го порядка.

FT_PT1 сглаживает изменения величины ($K * IN$): выходной сигнал изменяется за время T на 63% от изменения ($K * IN$), за время $3 * T$ – на 95%, а затем асимптотически приближается к ($K * IN$).

Если $T=t\#0s$, $OUT := K * IN$ (т.е. сглаживания нет).

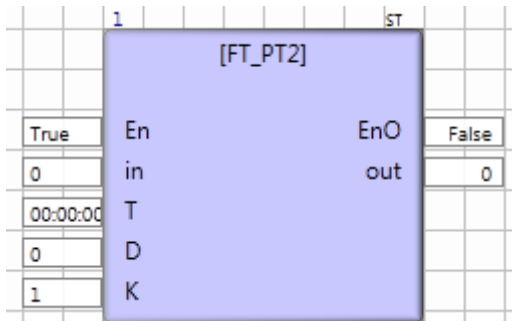
При старте ФБ $OUT := K * IN$.

Ниже показаны отклики ФБ FT_PT1 на изменения IN при $T = 15c$ (время цикла – $1c$):



Используемые функции: T_PLC_MS (OSCAT).

11.1.21.21. FT_PT2 (OSCAT)



Входы ФБ:

- IN (REAL) – входной сигнал;
- T (TIME) – постоянная времени;
- D (REAL) – декремент затухания;
- K (REAL) – коэффициент усиления.

Выходы ФБ:

- OUT (REAL) – выходной сигнал.

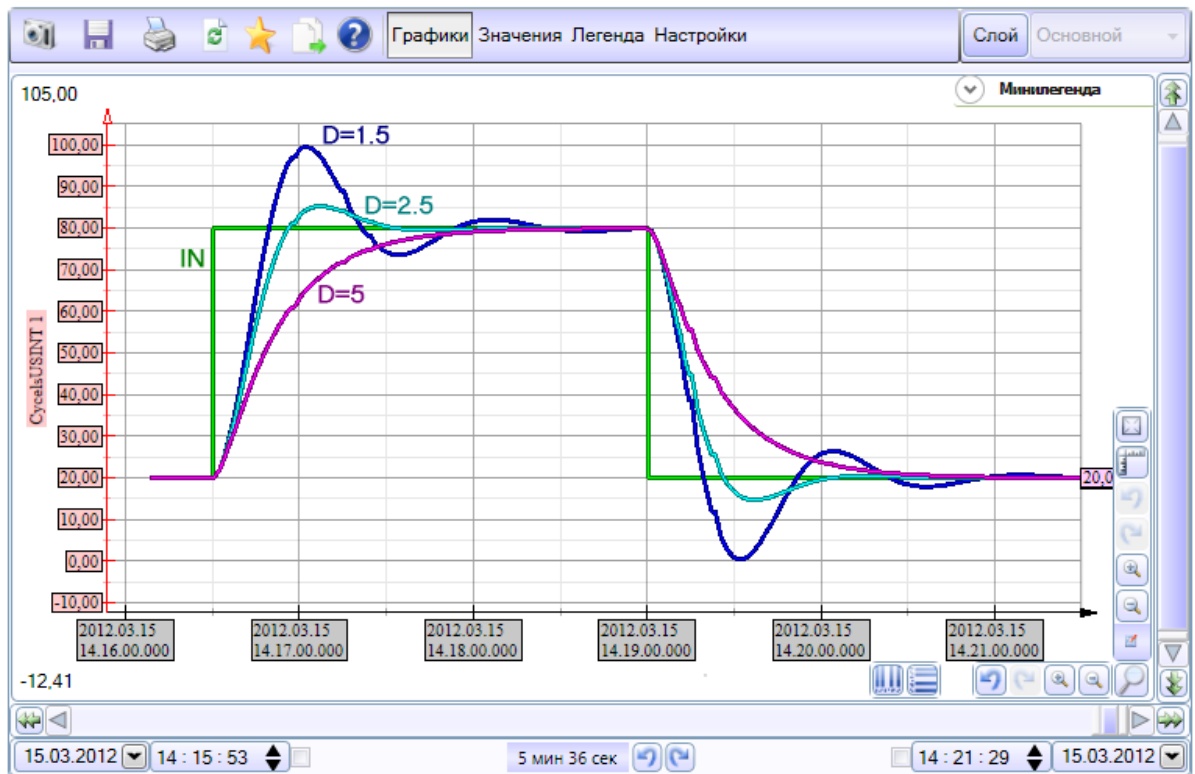
FT_PT2 – это низкочастотный фильтр 2-го порядка.

Если $T=t\#0s$, $OUT = K * IN$.

Связь между переменными ФБ выражается следующим дифференциальным уравнением:

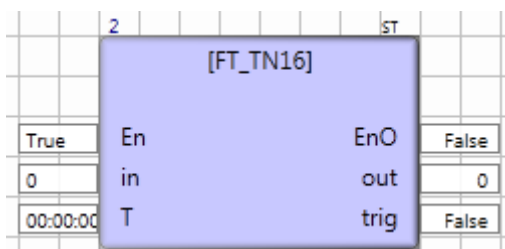
$$T2*OUT''(T) + 2*D*T*OUT'(T) + OUT(T) = K*in(T)$$

Ниже показаны отклики ФБ FT_PT2 на изменения IN при T=10с и D = 1.5 / 2.5 / 5 (время цикла – 1с):



Используемые функции: INTEGRATE (OSCAT)_

11.1.21.22. FT_TN16 (OSCAT)



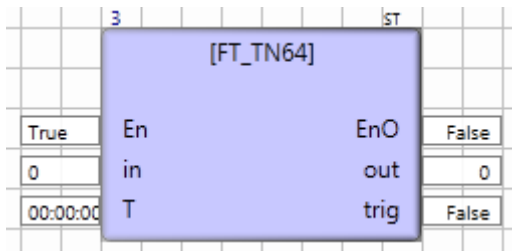
Тип данных входа IN и выхода OUT – REAL, входа T – TIME, выхода TRIG – BOOL.

FT_TN16 представляет собой аналог FT_TN8 (OSCAT), ФБ передает на выход OUT выборки из входного сигнала IN с задержкой T. Выборка из IN производится 1 раз за время T/16 (т.е. за T выбирается 16 значений). При каждой выборке/передаче TRIG=TRUE в течение 1 цикла.

Еще одним ФБ подобного типа является FT_TN64 (OSCAT).

Используемые функции: T_PLC_MS (OSCAT).

11.1.21.23. FT_TN64 (OSCAT)



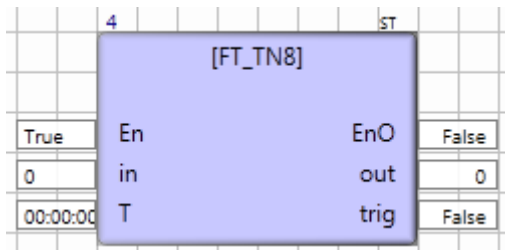
Тип данных входа IN и выхода OUT – REAL, входа T – TIME, выхода TRIG – BOOL.

FT_TN64 представляет собой аналог FT_TN8 (OSCAT), ФБ передает на выход OUT выборки из входного сигнала IN с задержкой T. Выборка из IN производится 1 раз за время T/64 (т.е. за T выбирается 64 значения). При каждой выборке/передаче TRIG=TRUE в течение 1 цикла.

Еще одним ФБ подобного типа является FT_TN16 (OSCAT).

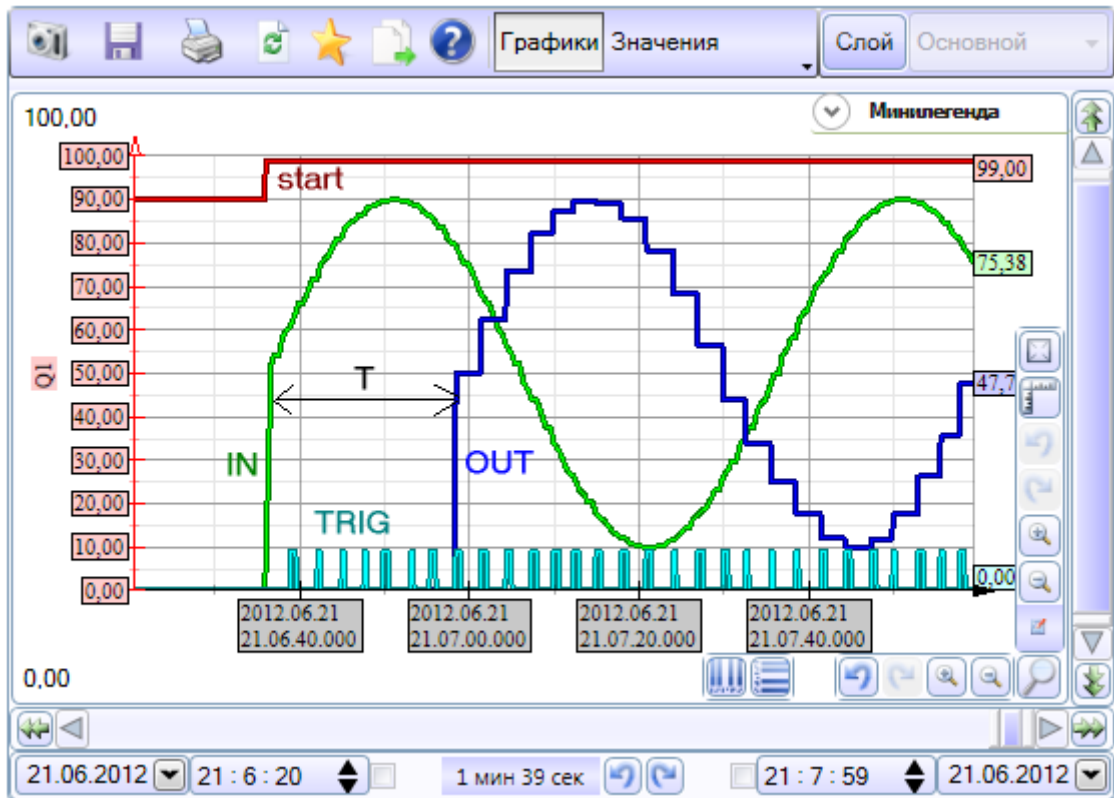
Используемые функции: T_PLC_MS (OSCAT).

11.1.21.24. FT_TN8 (OSCAT)



Тип данных входа IN и выхода OUT – REAL, входа T – TIME, выхода TRIG – BOOL.

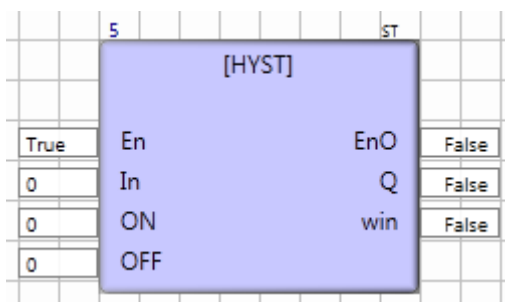
FT_TN8 передает на выход OUT выборки из входного сигнала IN с задержкой T. Выборка из IN производится 1 раз за время T/8 (т.е. за T выбирается 8 значений). При каждой выборке/передаче TRIG=TRUE в течение 1 цикла:



Аналогами FT_TN8 являются ФБ FT_TN16 (OSCAT) и FT_TN64 (OSCAT).

Используемые функции: T_PLC_MS (OSCAT).

11.1.21.25. HYST (OSCAT)



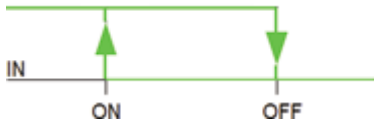
Тип данных входов IN, ON и OFF – REAL, выходов Q и WIN – BOOL.

HYST реализует гистерезис, зависящий от соотношения между значениями входов ON и OFF:

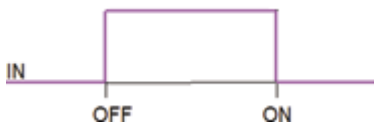
- ON >= OFF:
- если Q=FALSE, то Q становится TRUE при IN>ON; если Q=TRUE, то Q становится FALSE при IN<OFF:



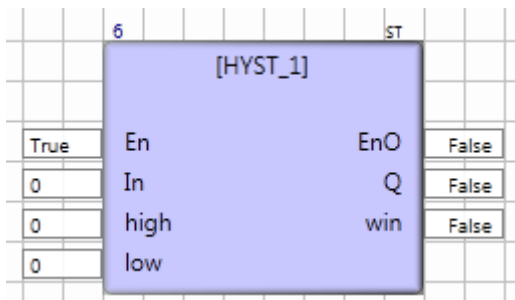
- ON<OFF:
- если Q=FALSE, то Q становится TRUE при IN<ON; если Q=TRUE, то Q становится FALSE при IN>OFF:



В обоих случаях WIN:=TRUE, если ON<=IN<=OFF; если IN лежит вне диапазона [ON, OFF], WIN:=FALSE:

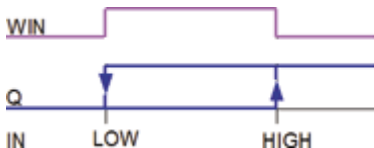


11.1.21.26. HYST_1 (OSCAT)

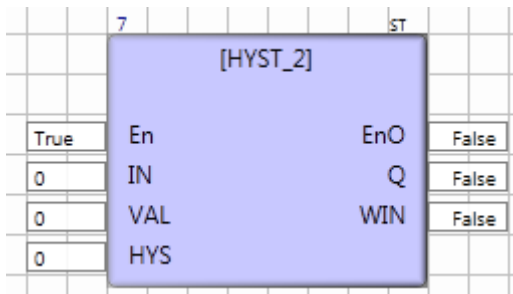


Тип данных входов IN, HIGH и LOW – REAL, выходов Q и WIN – BOOL.

При HIGH>LOW HYST_1 реализует гистерезис: если Q:=FALSE, то Q становится TRUE при IN>HIGH; если Q:=TRUE, то Q становится FALSE при IN<LOW. Если LOW<=IN<=HIGH, то WIN:=TRUE, иначе WIN:=FALSE.



11.1.21.27. HYST_2 (OSCAT)



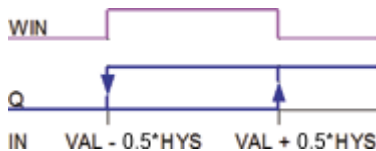
Тип данных входов IN, VAL и HYS – REAL, выходов Q и WIN – BOOL.

HYST_2 реализует гистерезис.

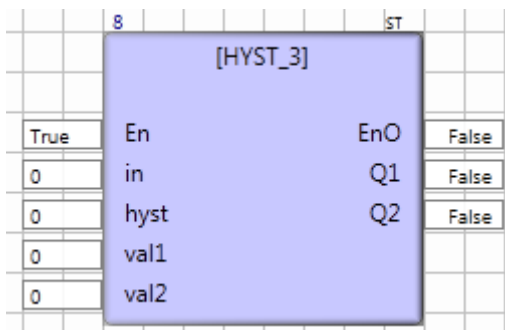
Если Q:=FALSE, то Q становится TRUE при $IN > VAL + 0.5 * HYS$.

Если Q:=TRUE, то Q становится FALSE при $IN < VAL - 0.5 * HYS$.

Если $(VAL - 0.5 * HYS) \leq IN \leq (VAL + 0.5 * HYS)$, то WIN:=TRUE, иначе WIN:=FALSE.



11.1.21.28. HYST_3 (OSCAT)

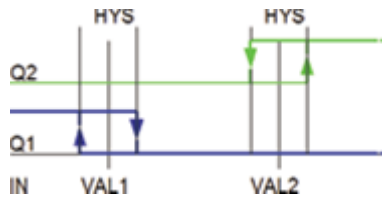


Тип данных входов IN, HYST, VAL1 и VAL2 – REAL, выходов Q1 и Q2 – BOOL.

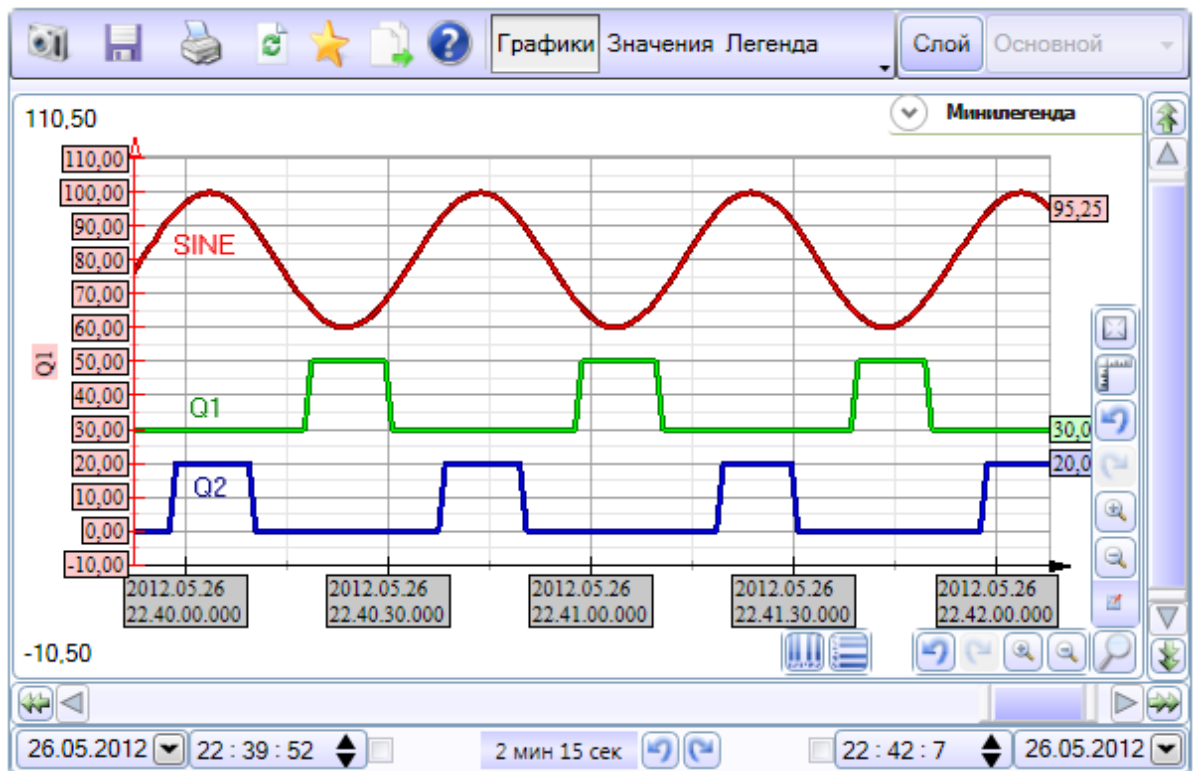
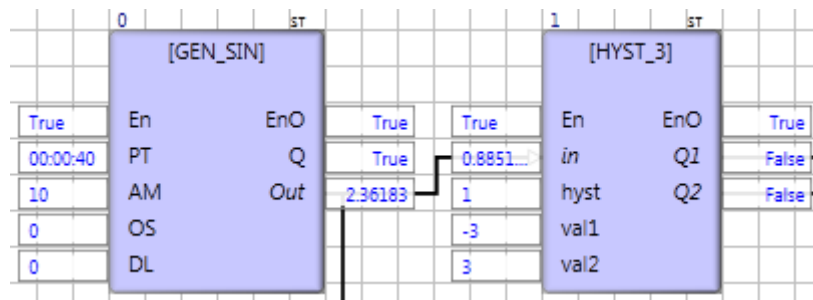
HYST_3 – это 3-точечный контроллер. 3-точечный контроллер содержит две функции гистерезиса.

Если Q1=FALSE, то Q1 становится TRUE при $IN < (VAL1 - HYST/2)$. Если Q1=TRUE, то Q1 становится FALSE при $IN > (VAL1 + HYST/2)$.

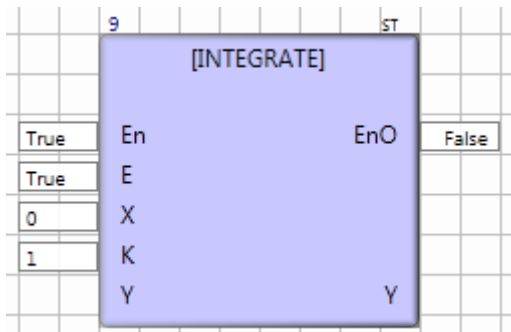
Если Q2=FALSE, то Q2 становится TRUE при $IN > (VAL2 + HYST/2)$. Если Q2=TRUE, то Q2 становится FALSE при $IN < (VAL2 - HYST/2)$.



3-точечный контроллер используется, например, для управления приводом клапана. Следующий пример демонстрирует сигналы выходов 3-точечного контроллера:



11.1.21.29. INTEGRATE (OSCAT)



Входы ФБ:

- E (BOOL) – запуск/останов интегрирования (TRUE/FALSE);
- X (REAL) – интегрируемый сигнал;
- K (REAL) – коэффициент, определяющий размерность интеграла.

Входы-выходы ФБ:

- Y (REAL) – текущее значение интеграла.

INTEGRATE вычисляет определенный интеграл от входного сигнала X(t) с момента времени, когда E впервые принимает значение TRUE после старта или сброса (если E:=FALSE после E:=TRUE, интегрирование приостанавливается, а Y сохраняет свое значение, и при последующем возобновлении интегрирования по команде E:=TRUE начальное значение интеграла будет равно сохраненному Y).

Для вычисления интеграла используется метод трапеций (n – номер цикла, t – время внутреннего таймера ПЛК в миллисекундах):

$$Y_n := Y_{n-1} + K \cdot \frac{X_n + X_{n-1}}{2 \cdot 1000} \cdot (t_n - t_{n-1})$$

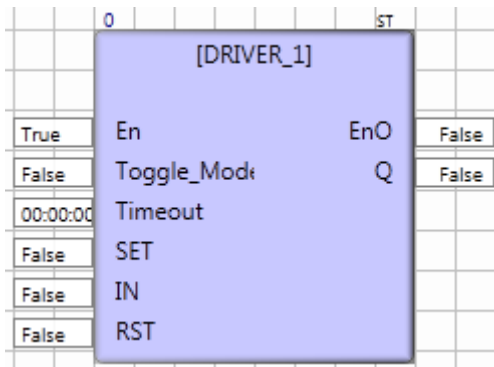
Временные метки определяются с помощью функции T_PLC_MS (OSCAT).

При K=1 размерность полученного интеграла равна [<размерность X>*с]. Чтобы получить интеграл с другими единицами времени в размерности, нужно задать соответствующий коэффициент. Например, при K=1/3600 размерность интеграла равна [<размерность X>*ч].

Для сброса значения интеграла нужно присвоить 0 входу-выходу Y.

11.1.22. OSCAT.ФБ.УПРАВЛЕНИЕ УСТРОЙСТВАМИ

11.1.22.1. DRIVER_1 (OSCAT)



Тип данных входа TIMEOUT – TIME, остальных входов и выхода Q – BOOL.

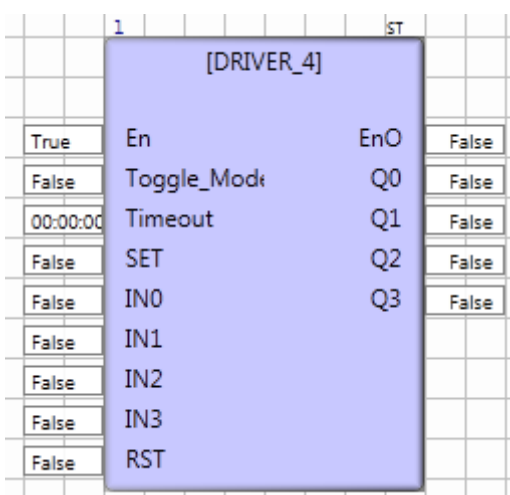
Алгоритм ФБ при SET=RST=FALSE:

- TOGGLE_MODE=FALSE:
- если IN=TRUE, Q=TRUE (если TIMEOUT>t#0s, Q=TRUE в течение TIMEOUT);
- если IN=FALSE, Q=FALSE;
- TOGGLE_MODE=TRUE:
- передний фронт на входе IN переключает значение Q (если TIMEOUT>t#0s, Q=TRUE в течение TIMEOUT).

Если SET=TRUE при RST=FALSE, Q=TRUE (принудительная установка Q).

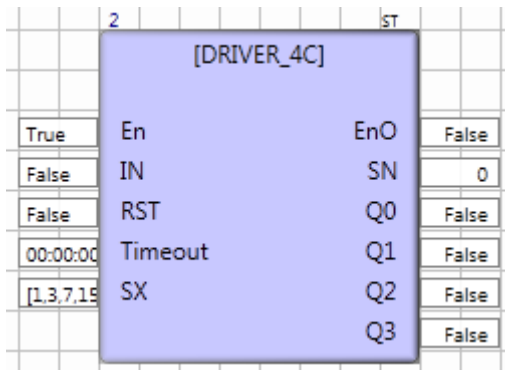
Если RST=TRUE, Q=FALSE (принудительный сброс Q).

11.1.22.2. DRIVER_4 (OSCAT)



Данный ФБ представляет собой совокупность четырех ФБ DRIVER_1 (OSCAT).

11.1.22.3. DRIVER_4C (OSCAT)



Тип данных входов IN и RST – BOOL, TIMEOUT – TIME, SX (массив 7 переменных, индексация с 1) – BYTE, выходов Q0..Q3 – BOOL, SN – INT.

ФБ предназначен для последовательного перехода по состояниям (не более 7). Переход из состояния в состояние производится по переднему фронту на входе IN. Номер состояния индицируется на выходе SN. Значения выходов Q0..Q3 в состоянии SN равны соответственно значениям битов 0..3 элемента массива SX[SN]. При старте SN=0 и Q0=Q1=Q2=Q3=FALSE.

Последовательность переходов прерывается в следующих случаях:

- вне зависимости от значения TIMEOUT;
- если SX[SN]=0;
- если производится переход в состояние SN>7;
- если RST=TRUE;
- при TIMEOUT>t#0s;
- если в течение TIMEOUT переход не выполняется.

Во всех указанных случаях SN=0 и Q0=Q1=Q2=Q3=FALSE.

Пример

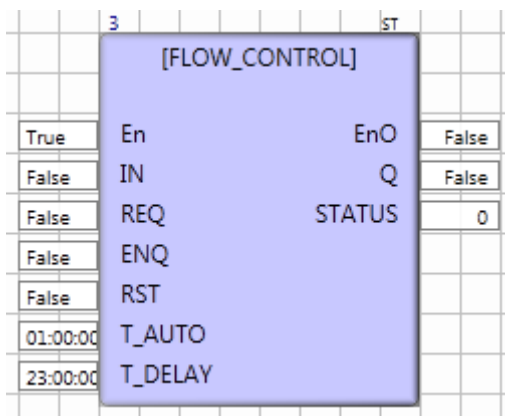
Если SX = [1,3,7,15,7,3,1], то при переходах выходы Q3, Q2, Q1 и Q0 будут принимать следующие значения:

- SN=0 (старт) – 0000;
- SN=1 – 0001;
- SN=2 – 0011;
- SN=3 – 0111;
- SN=4 – 1111;

- SN=5 – 0111;
- SN=6 – 0011;
- SN=7 – 0001;
- SN=0 – 0000;
- SN=1 – 0001;
- и т.д.

Если TIMEOUT=t#0s, ФБ может находиться в любом состоянии произвольное время.

11.1.22.4. FLOW_CONTROL (OSCAT)



Тип данных входов IN, REQ, ENQ, RST и выхода Q – BOOL, входов T_AUTO и T_DELAY – TIME, выхода STATUS – BYTE.

ФБ вырабатывает сигнал включения устройства (Q=TRUE) двумя способами:

- с помощью входа IN: если IN=TRUE/FALSE при ENQ=TRUE, Q=TRUE/FALSE (управление вручную);
- с помощью входа REQ: если при ENQ=TRUE на входе REQ детектируется передний фронт, Q принимает значение TRUE на время T_AUTO, после чего в течение времени T_DELAY включение с помощью входа REQ блокируется (однако управление с помощью IN возможно).

Для реинициализации алгоритма нужно присвоить входу RST значение TRUE (по этой команде выходы обнуляются), а затем FALSE.

Выход STATUS индицирует ESR-совместимый статус:

- 100 – готов;
- 101 – IN=TRUE;

- 102 – REQ=TRUE;
- 103 – RST=TRUE.

Используемые функции: TP_1D (OSCAT).

11.1.22.5. FT_PROFILE (OSCAT)

1	[FT_Profile]		ST
True	En	EnO	False
2	K	Y	0
10	O	RUN	False
2	M	ET	0ms
False	E		
0	value_0		
5000ms	time_1		
10	value_1		
10000ms	time_2		
10	value_2		
15000ms	time_3		
30	value_3		
20000ms	time_10		
30	value_10		
25000ms	time_11		
15	value_11		
30000ms	time_12		
15	value_12		
35000ms	time_13		
0	value_13		

Тип данных входа E и выхода RUN – BOOL, входов K, O, M, VALUE_N и выхода Y – REAL, входов TIME_N и выхода ET – TIME.

ФБ предназначен для задания временного профиля некоторого параметра – например, температуры в печи.

Алгоритм запускается по переднему фронту на входе E. В этот момент RUN принимает значение TRUE, Y – VALUE_0, ET обнуляется и начинается отсчет времени работы алгоритма, и начинает обрабатываться первый участок профиля.

Каждый участок профиля характеризуется парой значений (TIME_<N>, VALUE_<N>). Участки обрабатываются по очереди в последовательности 1-2-3-10-11-12-13, при этом участки 11-12-13 обрабатываются только после того, как E примет значение FALSE.

Длительность участка N составляет

$M * (TIME_<N> - \text{<время начала отработки участка } N>)$

Временем начала отработки участка является либо время окончания отработки предыдущего участка, либо время выполнения необходимого условия ($E=TRUE/FALSE$).

При отработке участка N значение Y изменяется по линейному закону:

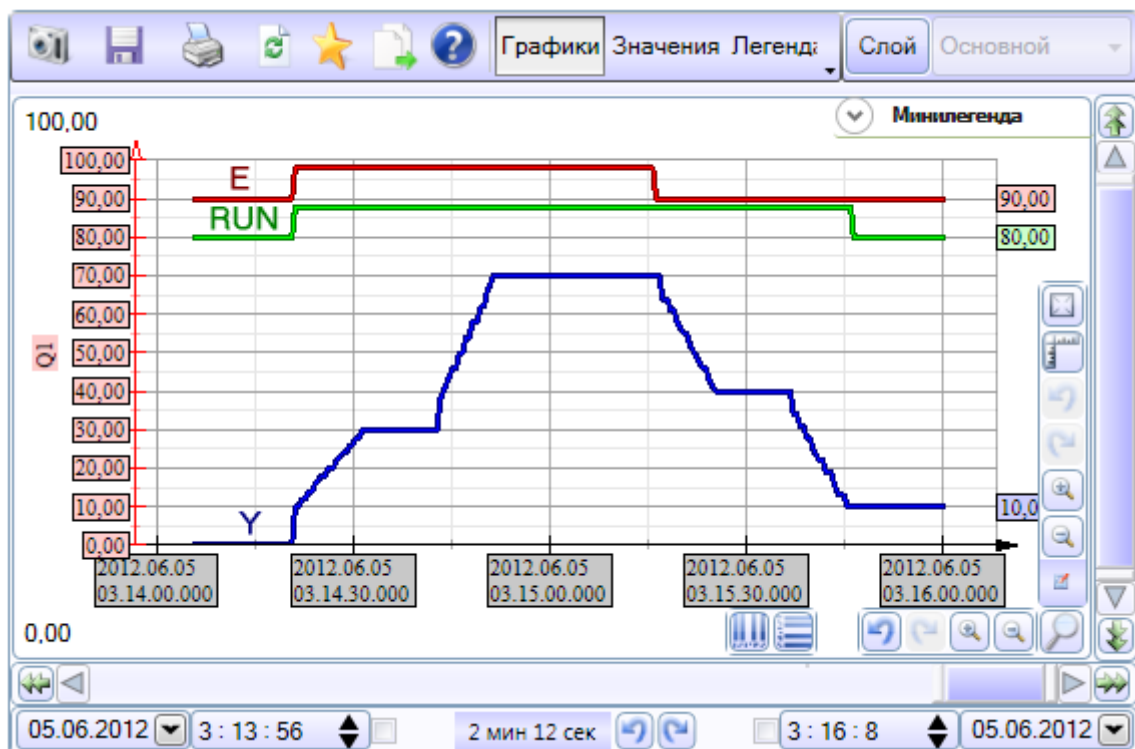
$$Y := K * [(VALUE_<N> - VALUE_<предыдущее>) * \text{<время с начала участка } N> / \text{<длительность участка } N>] + O;$$

По окончании отработки всех участков $RUN=FALSE$, $Y=VALUE_13$, а ET индицирует время, затраченное на отработку.

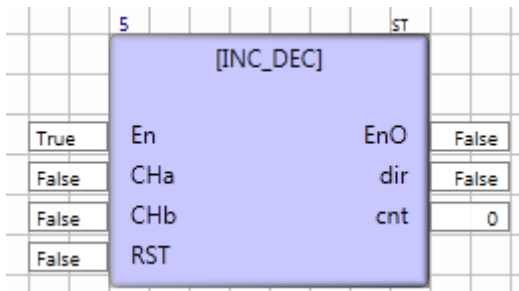
Используемые функции: T_PLC_MS (OSCAT).

Пример

Следующая временная диаграмма демонстрирует профиль при конфигурации ФБ, показанной на рисунке выше:



11.1.22.6. INC_DEC (OSCAT)



Входы ФБ:

- CHA (BOOL) – канал А;
- CHB (BOOL) – канал В;
- RST (BOOL) – сброс.

Выходы ФБ:

- DIR (BOOL) – направление вращения;
- CNT (INT) – счетчик.

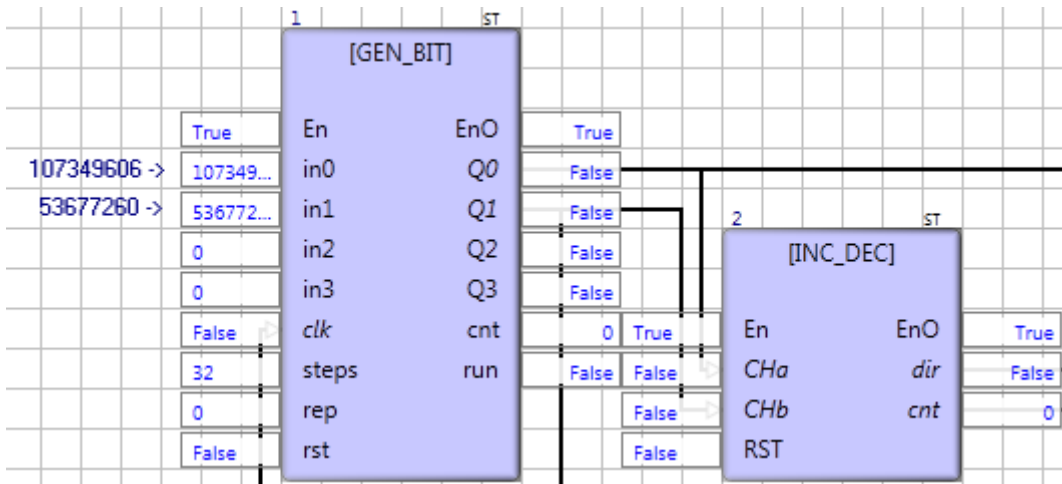
INC_DEC – это декодер сигналов кодера вращения.

Два сигнала кодера вращения, называемые каналами А и В, – это импульсы, частично перекрывающиеся друг друга (шаги поворота). По этим сигналам INC_DEC декодирует направление и угол поворота. Выход DIR индицирует направление вращения, а выход CNT – учетверенное число шагов поворота (INC_DEC детектирует каждый фронт в каналах кодера).

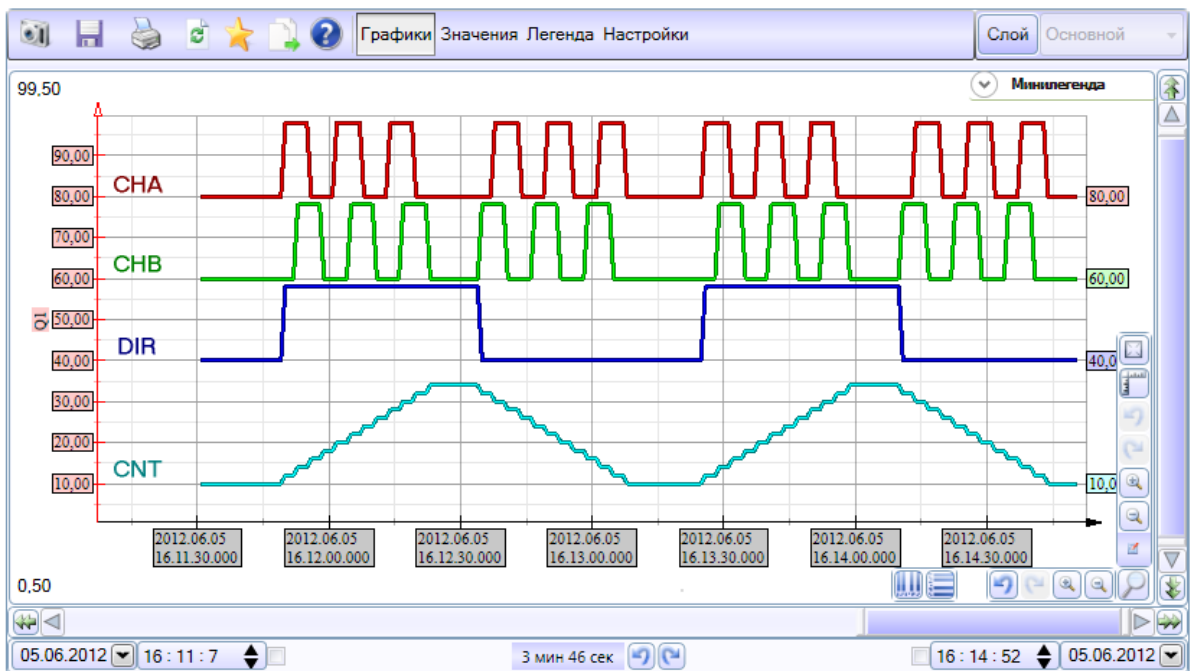
Команда RST=TRUE сбрасывает счетчик. Счетчик нарастающий, если DIR=TRUE, и убывающий, если DIR=FALSE.

Пример

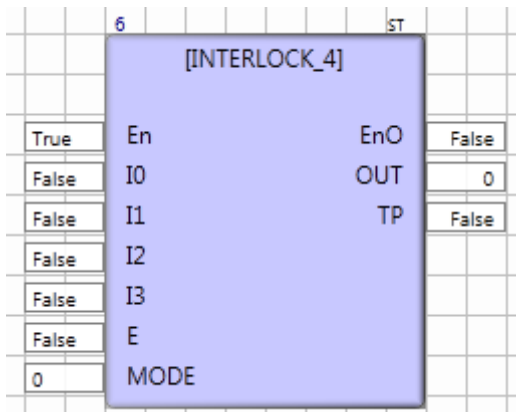
В данном примере в качестве имитатора кодера вращения используется ФБ GEN_BIT (OSCAT); ФБ имитирует 3 шага поворота по часовой стрелке и 3 шага в обратном направлении (107349606 = 2#0000_0110_0110_0110_0000_0110_0110_0110, 53677260 = 2#0000_0011_0011_0011_0000_1100_1100_1100):



Временная диаграмма программы показывает разбиение 3 шагов поворота на 12 и детектирование направления вращения:



11.1.22.7. INTERLOCK_4 (OSCAT)



Входы ФБ:

- I0 (BOOL) – вход 0;
- I1 (BOOL) – вход 1;
- I2 (BOOL) – вход 2;
- I3: BOOL – вход 3;
- E (BOOL) – разрешение/запрет работы алгоритма ФБ;
- MODE (INT) – режим работы ФБ.

Выходы ФБ:

- OUT (BOOL) – выход;
- TP (BOOL) – индикатор изменения OUT.

INTERLOCK_4 хранит значения входов I0..I3 в битах 0..3 выхода OUT.

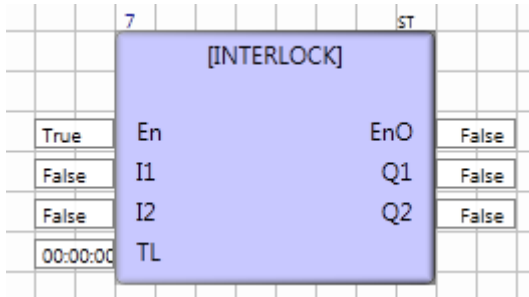
При каждом изменении OUT TP=TRUE в течение 1 цикла.

Если E=FALSE, выходы ФБ обнуляются.

Значение MODE задает режим работы ФБ:

- 0 – все входы передаются в OUT. Например, если I0=1, I1=0, I2=1 и I3=0, OUT=2#0000_0101=5;
- 1 – в OUT передается только старший по номеру вход со значением TRUE. Например, если I0=1, I1=0, I2=1 и I3=0, OUT=2#0000_0100=4;
- 2 – в OUT передается только вход, который принял значение TRUE последним. Если одновременно несколько входов принимают значение TRUE, в OUT передается старший по номеру;
- 3 – в OUT передается только вход, который принял значение TRUE первым. Если при переходе в данный режим несколько входов имеют значение TRUE, в OUT передается старший по номеру.

11.1.22.8. INTERLOCK (OSCAT)



Входы ФБ:

- I1 (BOOL) – вход 1;
- I2 (BOOL) – вход 2;
- TL (TIME) – время блокировки.

Выходы ФБ:

- Q1 (BOOL) – выход 1;
- Q2 (BOOL) – выход 2.

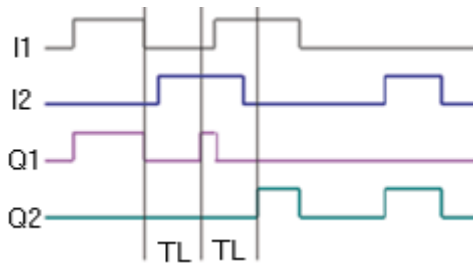
В зависимости от сочетания значений входов I1 и I2 выходы Q1 и Q2 принимают значения, показанные в таблице ниже, при соблюдении следующего условия:

- выход Q<i> может принять значение TRUE только в том случае, если значение соответствующего входа I<i> равно TRUE, а с момента времени, когда другой выход принял значение FALSE, прошло не менее TL.

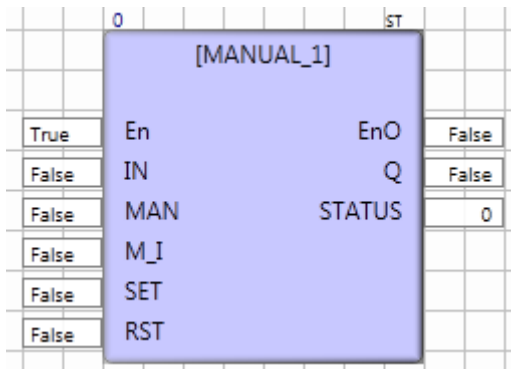
Как следствие, оба выхода не могут быть равны TRUE одновременно.

I1	I2	Q1	Q2
0	0	0	0
0	1	0	1
1	0	1	0
1	1	0	0

Следующая временная диаграмма поясняет алгоритм ФБ:



11.1.22.9. MANUAL_1 (OSCAT)



Входы ФБ:

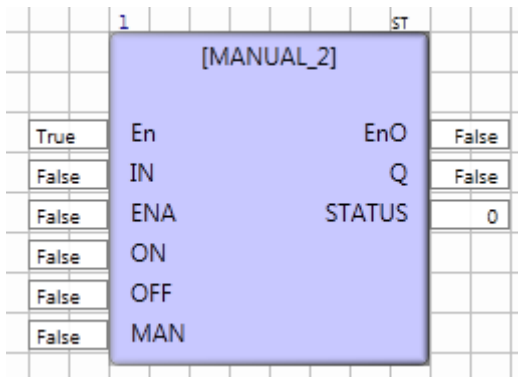
- IN (BOOL) – сигнал;
- MAN (BOOL) – включение/отключение ручного режима;
- M_I (BOOL) – сигнал в ручном режиме;
- SET (BOOL) – установка в ручном режиме;
- RST (BOOL) – сброс в ручном режиме.

Выходы ФБ:

- Q (BOOL) – выход;
- STATUS (BYTE) – ESR-совместимый статус.

MANUAL_1 позволяет заменить цифровой сигнал в ручном режиме:

- если MAN=FALSE, то Q:=IN и STATUS:=100;
- если MAN=TRUE:
 - если SET=RST=FALSE, то Q:=M_I и STATUS:=103 (ручной режим);
 - если SET=TRUE и RST=FALSE, то Q:=TRUE и STATUS:=101 (принудительная установка выхода);
 - если RST=TRUE, то Q:=FALSE и STATUS:=102 (принудительный сброс выхода).

11.1.22.10. MANUAL_2 (OSCAT)

Входы ФБ:

- IN (BOOL) – сигнал;
- ENA (BOOL) – разрешение/запрет работы алгоритма ФБ;
- ON (BOOL) – принудительная установка выхода;
- OFF (BOOL) – принудительный сброс выхода;
- MAN (BOOL) – сигнал в ручном режиме.

Выходы ФБ:

- Q (BOOL) – выход;
- STATUS (BYTE) – ESR-совместимый статус.

MANUAL_2 позволяет заменить цифровой сигнал в ручном режиме:

- если ENA=FALSE, то Q:=FALSE и STATUS:=104;
- если ENA=TRUE:
- если ON=OFF=FALSE, то Q:=IN и STATUS:=100;
- если ON=TRUE и OFF=FALSE, то Q:=TRUE и STATUS:=101 (принудительная установка выхода);
- если ON=FALSE и OFF=TRUE, то Q:=FALSE и STATUS:=102 (принудительный сброс выхода);
- если ON=OFF=TRUE, то Q:=MAN и STATUS:=103 (ручной режим).

11.1.22.11. MANUAL_4 (OSCAT)

2		ST	
[MANUAL_4]			
True	En	EnO	False
False	I0	Q0	False
False	I1	Q1	False
False	I2	Q2	False
False	I3	Q3	False
False	MAN	STATUS	0
False	STP		
False	M0		
False	M1		
False	M2		
False	M3		

Входы ФБ:

- I0..I3 (BOOL) – сигналы;
- MAN (BOOL) – включение/отключение ручного режима;
- M0..M3 (BOOL) – сигналы в ручном режиме;
- STP (BOOL) – шаговый режим в ручном режиме.

Выходы ФБ:

- Q0..Q3 (BOOL) – выходы;
- STATUS (BYTE) – ESR-совместимый статус.

MANUAL_4 позволяет заменить цифровой сигнал в ручном режиме:

- если MAN=FALSE, то $Q<i>:=I<i>$ ($i=0,1,2,3$) и STATUS:=100;
- если MAN=TRUE (ручной режим):
- если STP=FALSE, то $Q<i>:=M<i>$ ($i=0,1,2,3$) и STATUS:=101;
- первый передний фронт на STP переводит ФБ в шаговый режим, при этом Q0:=TRUE, остальные выходы сбрасываются, а STATUS:=110. По второму переднему фронту на STP активируется только Q1 (STATUS:=111), по третьему – только Q2 (STATUS:=112), по четвертому – только Q3 (STATUS:=113), по пятому – снова Q0 (STATUS:=110) и т.д. Для выхода из шагового режима нужно выйти из ручного режима.

Используемые функции: INC (OSCAT).

11.1.22.12. PARSET (OSCAT)

True	En	EnO	False
False	A0	P1	0
False	A1	P2	0
0	X01	P3	0
0	X02	P4	0
0	X03		
0	X04		
0	X11		
0	X12		
0	X13		
0	X14		
0	X21		
0	X22		
0	X23		
0	X24		
0	X31		
0	X32		
0	X33		
0	X34		
00:00:00	TC		

Тип данных входов X и выходов P – REAL, входов A0 и A1 – BOOL, входа TC – TIME.

В зависимости от сочетания значений A0 и A1, на выходы P передаются значения входов X в соответствии с таблицей:

A0 A1	P1	P2	P3	P4
00	X01	X02	X03	X04
01	X11	X12	X13	X14
10	X21	X22	X23	X24
11	X31	X32	X33	X34

Изменение значений входов X после первого запуска ФБ не поддерживается. Изменение TC в реальном времени поддерживается.

Если $TC=t\#0s$, переход к новым значениям выходов выполняется ступенчато, если $TC>t\#0s$ – по линейному закону за время TC.

Используемые функции: T_PLC_MS (OSCAT).

11.1.22.13. PARSET2 (OSCAT)

True	En	EnO	False
0	X	P1	0
0	X01	P2	0
0	X02	P3	0
0	X03	P4	0
0	X04		
0	X11		
0	X12		
0	X13		
0	X14		
0	X21		
0	X22		
0	X23		
0	X24		
0	X31		
0	X32		
0	X33		
0	X34		
0	L1		
0	L2		
0	L3		
00:00:00	TC		

Тип данных входа TC – TIME, остальных входов и выходов – REAL.

В зависимости от абсолютного значения X относительно значений L1, L2 и L3, на выходы P передаются значения входов X<i><k> в соответствии с таблицей:

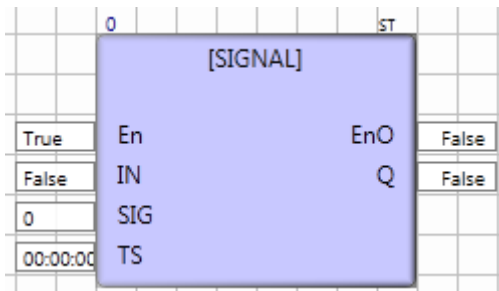
X	P1	P2	P3	P4
$\text{abs}(X) < L1$	X01	X02	X03	X04
$L1 \leq \text{abs}(X) < L2$	X11	X12	X13	X14
$L2 \leq \text{abs}(X) < L3$	X21	X22	X23	X24
$\text{abs}(X) \geq L3$	X31	X32	X33	X34

Изменение значений входов $X_{i < k >}$ и TC после первого запуска ФБ не поддерживается. Изменение границ $L1$, $L2$ и $L3$ в реальном времени поддерживается.

Если $TC=t\#0s$, переход к новым значениям выходов выполняется ступенчато, если $TC>t\#0s$ – по линейному закону за время TC .

Используемые функции: PARSET (OSCAT).

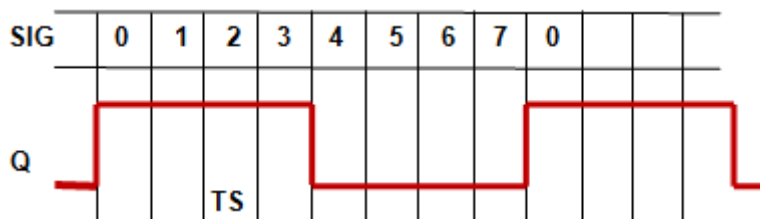
11.1.22.14. SIGNAL (OSCAT)



Тип данных входа IN и выхода Q – BOOL, входа SIG – BYTE, входа TS – TIME.

SIGNAL генерирует на выходе Q прямоугольный сигнал по битовому образцу SIG. При $IN=TRUE$ биты SIG последовательно передаются на выход со скоростью 1 бит за время TS , если $TS>t\#0s$; в противном случае – со скоростью 1 бит за 128 мс.

Например, по образцу $2\#10101010$ генерируются импульсы шириной TS и такой же паузой между импульсами (частота следования импульсов – $1/(2*TS)$). По образцу $2\#11110000$ генерируются импульсы шириной $4*TS$ и такой же паузой между импульсами (частота следования импульсов – $1/(8*TS)$):

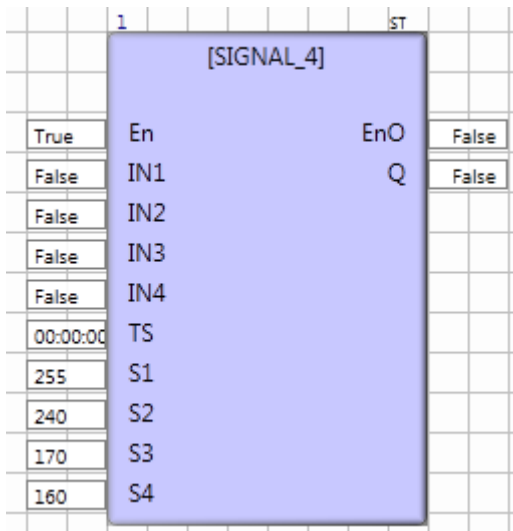


При старте генерации выходного сигнала первым передается случайный бит образца, далее передаются старшие биты, если $TS>t\#0s$, или младшие в противном случае.

Типичное применение ФБ – генерация сигнала для сирен или сигнальных ламп.

Используемые функции: T_PLC_MS (OSCAT).

11.1.22.15. SIGNAL_4 (OSCAT)



Используя ФБ SIGNAL (OSCAT), ФБ SIGNAL_4 может генерировать на выходе Q прямоугольный сигнал по битовым образцам S1..S4.

TS определяет скорость передачи битов (см. SIGNAL (OSCAT)).

Входы IN1..IN4 имеют приоритет. Если IN1=TRUE, то, вне зависимости от значений входов IN2..IN4, сигнал генерируется по образцу S1. Если IN1=FALSE и IN2=TRUE, то, вне зависимости от значений входов IN3 и IN4, сигнал генерируется по образцу S2. Соответственно, сигнал по образцу S3 генерируется при IN3=TRUE и IN1=IN2=FALSE, а сигнал по образцу S4 – при IN4=TRUE и IN1=IN2=IN3=FALSE.

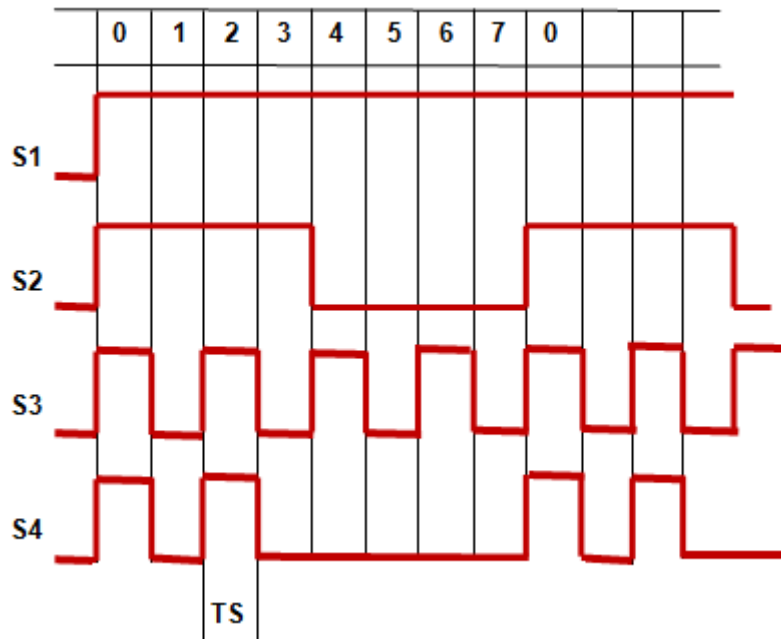
В ФБ заданы по умолчанию следующие образцы (могут быть изменены, в том числе в реальном времени):

S1 = 2#1111_1111

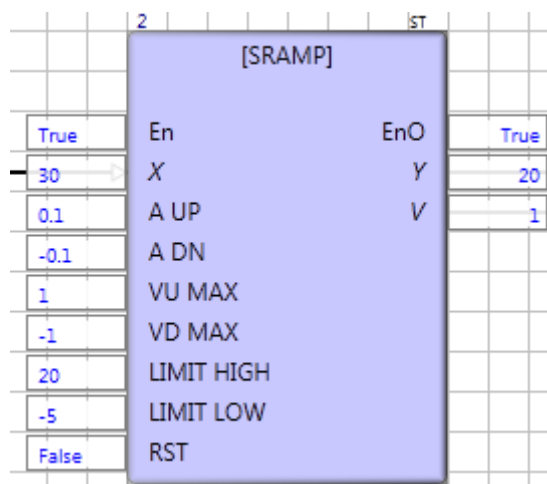
S2 = 2#1111_0000

S3 = 2#1010_1010

S4 = 2#1010_0000



11.1.22.16. SRAMP (OSCAT)



Входы ФБ:

- X (REAL) – вход;
- A_UP (REAL) – максимальное ускорение нарастания;
- A_DN (REAL) – максимальное ускорение убывания;
- VU_MAX (REAL) – максимальная скорость нарастания;
- VD_MAX (REAL) – максимальная скорость убывания;
- LIMIT_HIGH (REAL) – верхний предел выходного сигнала Y;
- LIMIT_LOW (REAL) – нижний предел выходного сигнала Y;

- RST (BOOL) – асинхронный сброс.

Выходы ФБ:

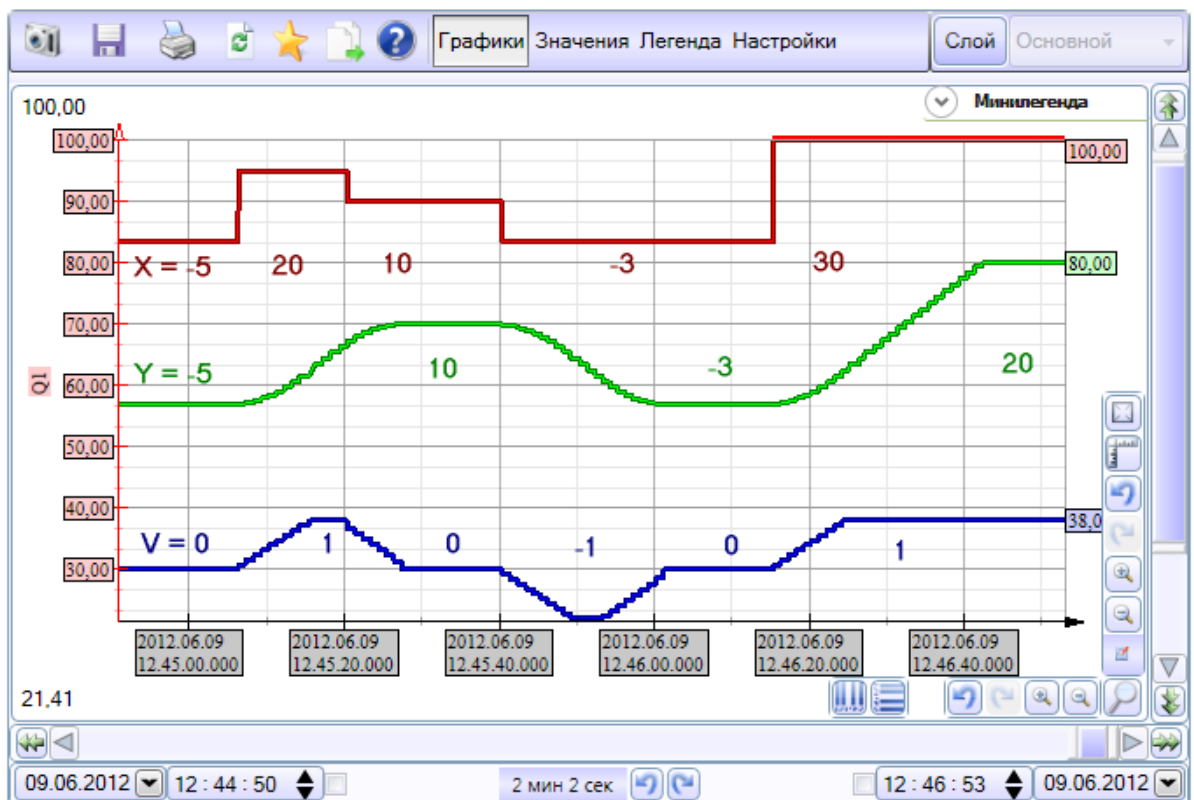
- Y (REAL) – выходной сигнал;
- V (REAL) – текущая скорость выходного сигнала.

SRAMP генерирует выходной сигнал по входному (X), ограничивая скорость (VU_MAX и VD_MAX), значение (LIMIT_LOW и LIMIT_HIGH) и ускорение (A_UP и A_DN). SRAMP используется, например, для управления двигателем.

Значения для A_UP и VU_MAX должны быть положительными, для A_DN и VD_MAX – отрицательными.

Выход V индицирует текущую скорость выходного сигнала Y.

На рисунке ниже приведена временная диаграмма ФБ SRAMP, конфигурация которого показана на рисунке выше:



Используемые функции: TC_S (OSCAT),

11.1.22.17. TUNE (OSCAT)

3		ST	
[TUNE]			
True	En	EnO	False
False	SET	Y	0
False	SU		
False	SD		
False	RST		
0.1	SS		
0	Limit_L		
100	LIMIT_H		
0	RST_val		
100	SET_val		
00:00:00	T1		
00:00:02	T2		
2	S1		
10	S2		

Входы ФБ:

- SET (BOOL) – командный вход установки выхода;
- SU (BOOL) – командный вход разрешения (TRUE) или запрета (FALSE) увеличения;
- SD (BOOL) – командный вход разрешения (TRUE) или запрета (FALSE) уменьшения;
- RST (BOOL) – командный вход установки выхода;
- SS (REAL) – шаг дискретного изменения;
- LIMIT_L (REAL) – нижний предел выхода;
- LIMIT_H (REAL) – верхний предел выхода;
- RST_VAL (REAL) – значение выхода, устанавливаемое по команде RST=TRUE;
- SET_VAL (REAL) – значение выхода, устанавливаемое по команде SET=TRUE при RST=FALSE;
- T1 (TIME) – таймаут 1;
- T2 (TIME) – таймаут 2;
- S1 (REAL) – скорость непрерывного увеличения;
- S2 (REAL) – скорость непрерывного уменьшения.

Выходы ФБ:

- Y (REAL) – выходной сигнал.

TUNE устанавливает выходной сигнал Y в дискретном и непрерывном режимах.

В дискретном режиме каждый импульс на входе SU (SD) шириной, меньшей или равной T1 (T2), увеличивает (уменьшает) значение выхода на величину SS (увеличение/уменьшение производится по заднему фронту импульса на соответствующем входе).

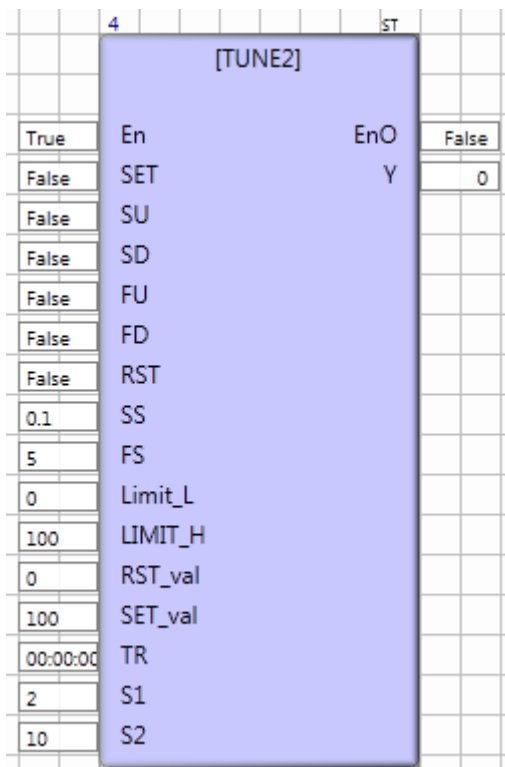
Если значение входа SU (SD) равно TRUE дольше, чем T1 (T2), ФБ переходит в непрерывный режим, в котором значение Y увеличивается (уменьшается) со скоростью S1 (S2) в секунду.

SU имеет более высокий приоритет по сравнению с SD. SU и SD могут использоваться только при SET=RST=FALSE.

Значение Y ограничивается значениями LIMIT_L и LIMIT_H.

Используемые функции: T_PLC_MS (OSCAT).

11.1.22.18. TUNE2 (OSCAT)



Входы ФБ:

- SET (BOOL) – командный вход установки выхода;
- SU (BOOL) – командный вход разрешения (TRUE) или запрета (FALSE) увеличения с малым шагом;

- SD (BOOL) – командный вход разрешения (TRUE) или запрета (FALSE) уменьшения с малым шагом;
- FU (BOOL) – командный вход разрешения (TRUE) или запрета (FALSE) увеличения с большим шагом;
- FD (BOOL) – командный вход разрешения (TRUE) или запрета (FALSE) уменьшения с большим шагом;
- RST (BOOL) – командный вход установки выхода;
- SS (REAL) – величина малого шага;
- FS (REAL) – величина большого шага;
- LIMIT_L (REAL) – нижний предел выхода;
- LIMIT_H (REAL) – верхний предел выхода;
- RST_VAL (REAL) – значение выхода, устанавливаемое по команде RST=TRUE;
- SET_VAL (REAL) – значение выхода, устанавливаемое по команде SET=TRUE при RST=FALSE;
- TR (TIME) – таймаут перехода в непрерывный режим;
- S1 (REAL) – скорость изменения в непрерывном режиме 1;
- S2 (REAL) – скорость изменения в непрерывном режиме 2.

Выходы ФБ:

- Y (REAL) – выходной сигнал.

TUNE2 устанавливает выходной сигнал Y в дискретном и непрерывном режимах.

Дискретный режим:

- каждый импульс на входе SU (SD) шириной, меньшей или равной TR, увеличивает (уменьшает) значение выхода на величину SS (увеличение/уменьшение производится по заднему фронту импульса на соответствующем входе);
- каждый импульс на входе FU (FD) шириной, меньшей или равной TR, увеличивает (уменьшает) значение выхода на величину FS (увеличение/уменьшение производится по заднему фронту импульса на соответствующем входе).

Если значение входа SU, SD, FU или FD равно TRUE дольше, чем TR, ФБ переходит в непрерывный режим. В этом режиме:

- SU=TRUE – Y увеличивается со скоростью S1 в секунду;

- SD=TRUE – Y уменьшается со скоростью S1 в секунду;
- FU=TRUE – Y увеличивается со скоростью S2 в секунду;
- FD=TRUE – Y уменьшается со скоростью S2 в секунду.

Командные входы имеют приоритет. В следующем списке командные входы расположены по убыванию приоритета слева направо: RST, SET, SU, SD, FU, FD.

Значение Y ограничивается значениями LIMIT_L и LIMIT_H.

Используемые функции: T_PLC_MS (OSCAT).