

# MasterSCADA 4D™

## Руководство пользователя

Версия 1.2.X



2020

*Краткое описание руководства пользователя*

Данное руководство предназначено для освоения программного обеспечения Master-SCADA 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: 123298, г. Москва,

ул. Маршала Бирюзова, д. 1, корп. 3

Телефон: +7 (495) 989-22-49

Факс: +7 (499) 943-02-14

E-mail: [scada@insat.ru](mailto:scada@insat.ru)

Сайт: [www.insat.ru](http://www.insat.ru)

1. Установка .....	38
1.1. Состав MasterSCADA 4D .....	38
1.2. Системные требования среды разработки .....	40
1.3. Системные требования для среды исполнения .....	40
1.4. Поддерживаемые контроллеры .....	43
1.5. Установка MasterSCADA 4D.....	44
1.6. Установка среды разработки MasterSCADA 4D.....	45
1.6.1. Описание ярлыков MasterSCADA 4D.....	49
1.6.1.1. Обновление версии на ОС Windows .....	50
1.6.1.2. Удаление MasterSCADA 4D .....	51
1.6.1.3. Запуск среды разработки.....	51
1.6.1.4. Установка дополнительных компонентов .....	51
1.6.1.4.1. Установка OPC Core Components Redistributable .....	51
1.6.1.4.2. Установка Microsoft.NET Framework .....	55
1.6.2. Установка среды исполнения MasterSCADA 4D .....	58
1.6.2.1. Установка на ОС Windows.....	58
1.6.2.1.1. Установка исполнительной системы MasterSCADA 4D RT.....	59
1.6.2.1.2. Описание ярлыков MasterSCADA 4D RT.....	63
1.6.2.1.3. Обновление версии исполнительной системы на ОС Windows .....	64
1.6.2.1.4. Удаление MasterSCADA 4D RT .....	64
1.6.2.1.5. Установка драйвера ключа защиты .....	65
1.6.2.2. Установка на ОС Linux или Эльбрус.....	67
1.6.2.2.1. Установка на ОС Linux или Эльбрус.....	67
1.6.2.2.2. Обновление среды исполнения на ОС Linux или Эльбрус .....	72
1.6.2.2.3. Удаление среды исполнения на ОС Linux или Эльбрус.....	73
1.6.2.3. Установка на ОС Windows Embedded.....	73

1.6.2.4.	Установка на ОС WinCE .....	73
1.6.2.5.	Установка на ОС QNX.....	75
1.6.3.	Установка и удаление клиента на ОС Windows .....	75
1.7.	Лицензионная защита.....	79
1.8.	Совместимость с антивирусами и файрволами.....	79
2.	Быстрый старт.....	82
2.1.	БС. Урок 1 .....	82
2.1.1.	БС. Урок 1. Создание проекта.....	82
2.1.2.	БС. Урок 1. Конфигурирование дерева системы.....	85
2.1.3.	БС. Урок 1. Создание логической части проекта.....	92
2.1.4.	БС. Урок 1. Создание окна управления.....	104
2.1.5.	БС. Урок 1. Запуск .....	117
2.2.	БС. Урок 2 .....	122
2.2.1.	БС. Урок 2. Конфигурирование всплывающего окна.....	122
2.2.2.	БС. Урок 2. Запуск .....	129
2.3.	БС. Урок 3 .....	129
2.3.1.	БС. Урок 3. Объектный подход при создании .....	130
	проекта.....	130
2.3.2.	БС. Урок 3. Запуск .....	144
2.3.3.	БС. Урок 3. Операции с библиотеками .....	144
3.	Обзор MasterSCADA 4D .....	151
3.1.	Основные характеристики.....	151
3.2.	Понятия .....	154
3.3.	Состав MasterSCADA 4D .....	159
3.4.	Описание интерфейса.....	161
3.5.	Демонстрационные проекты .....	162
3.6.	Лицензионная политика .....	164

3.7.	Принятые сокращения и обозначения .....	166
4.	Принципы проектирования .....	166
4.1.	Анализ объекта автоматизации .....	167
4.2.	Разработка типовых элементов .....	167
4.3.	Создание структуры проекта .....	170
4.4.	Разработка и отладка программ .....	181
4.5.	Разработка окон .....	183
4.6.	Конфигурирование связей.....	185
4.7.	Формирование задач узлов.....	187
4.8.	Автономная отладка в реальном времени .....	188
4.9.	Компиляция проекта и загрузка конфигураций в узлы.....	190
5.	Интерфейс редактора проекта .....	192
5.1.	Стартовое меню.....	195
5.2.	Панель инструментов.....	196
5.2.1.	Настройки редактора .....	197
5.2.1.1.	Настройки среды .....	199
5.2.1.1.1.	Вкладка Внешний вид.....	200
5.2.1.1.2.	Вкладка Компиляция.....	202
5.2.1.1.3.	Вкладка Клиент HMI .....	203
5.2.1.1.4.	Вкладка Лог .....	204
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.....	207
5.2.2.	Панель отмены/возврата действий .....	208
5.2.2.1.	Управление проектами .....	210
5.2.3.	История версий.....	213
5.2.4.	Вкладка инструментов Проект .....	215

5.2.5.	Вкладка инструментов Исполнение .....	216
5.2.6.	Вкладка инструментов Отладка .....	218
5.2.6.1.	Редактор набора наблюдений .....	218
5.2.6.2.	Окно Ошибки проекта .....	219
5.2.7.	Панель управления интерфейсом .....	220
5.2.8.	Панель инструментов Справка .....	221
5.3.	Дерево MasterSCADA .....	222
5.4.	Панель редакторов .....	227
5.5.	Контекстное меню .....	228
5.6.	Контекстная панель .....	230
5.7.	Панель свойств .....	232
5.7.1.	Панель свойств. Права .....	236
5.7.2.	Панель свойств. События .....	237
5.7.3.	Панель свойств. Отображение свойств, задействованных в связях .....	237
5.7.4.	Панель свойств. Отображение связей библиотечного элемента .....	238
5.7.5.	Панель свойств. Отображение свойств, имеющих структуру .....	240
5.8.	Панель веток дерева .....	241
5.9.	Легенда .....	242
5.10.	Типовые инструменты и операции .....	242
5.10.1.	Связывание переменных .....	242
5.10.2.	Элементарные инструменты .....	243
5.10.3.	Типовые инструменты окон .....	248
5.10.4.	Типовые операции .....	250
5.10.5.	Типовые инструменты дерева .....	254
5.11.	Операции в редакторе проекта .....	255
5.11.1.	Операции с проектами .....	255
5.11.1.1.	Диалог открытия проекта .....	256

5.11.1.1.1.	Диалог создания проекта .....	262
5.11.1.2.	Диалог сохранения проекта .....	264
5.11.1.3.	Автоматическое сохранение действий разработчика.....	265
5.11.1.4.	Параллельная разработка проекта .....	266
5.11.2.	Операции с элементами .....	267
5.11.2.1.	Диалог установки наследования.....	267
5.11.2.2.	Работа с именами элементов.....	269
5.11.2.3.	Выделение элементов .....	270
5.11.2.4.	Автоматическая прокрутка .....	270
5.11.2.5.	Перетаскивание с удержанием ПК .....	270
5.11.2.6.	Дублирование и копирование .....	271
5.11.2.7.	Перемещение элементов в дереве .....	273
5.12.	Служебные диалоговые окна .....	273
6.	Проект в MasterSCADA 4D.....	274
6.1.	Типы и экземпляры MasterSCADA 4D .....	275
6.2.	Методы разработки типов элементов.....	281
6.3.	Дерево системы.....	283
6.3.1.	Узел.....	283
6.3.1.1.	Панель свойств Узла .....	287
6.3.1.1.1.	Свойство Параметры запуска RT .....	292
6.3.1.2.	Параметры узла .....	294
6.3.1.3.	Графический интерфейс.....	295
6.3.1.3.1.	Шаблон экрана .....	296
6.3.1.3.1.1.	Панель свойств Шаблона экрана .....	296
6.3.1.3.2.	Медиа .....	300
6.3.1.3.2.1.	Элемент Медиа.....	300
6.3.1.3.3.	Источники данных экрана .....	300

6.3.1.3.3.1.	Элемент JSON .....	300
6.3.1.3.4.	Окна .....	301
6.3.1.4.	Программы.....	301
6.3.1.5.	Сообщения .....	302
6.3.1.6.	Службы .....	303
6.3.1.6.1.	Сохранение состояния .....	304
6.3.1.7.	Архивы.....	306
6.3.1.7.1.	Основной архив данных .....	308
6.3.1.7.2.	Основной архив сообщений .....	313
6.3.1.7.3.	Элемент Архив данных .....	317
6.3.1.7.4.	Базы данных.....	318
6.3.1.7.4.1.	SQLITE.....	319
6.3.1.7.4.2.	POSTGRESQL .....	321
6.3.1.7.4.3.	MS SQL .....	323
6.3.1.7.5.	Конфигурирование архива данных.....	325
6.3.1.7.5.1.	Слой данных.....	328
6.3.1.7.5.2.	Шаблоны архивирования .....	328
6.3.1.7.5.3.	Передача архивных данных .....	331
6.3.1.8.	Задачи.....	332
6.3.1.9.	Отчеты .....	334
6.3.1.10.	Внешние каналы .....	335
6.3.1.11.	Протоколы.....	335
6.3.1.12.	Встроенные каналы .....	337
6.3.1.13.	Встроенные модули .....	338
6.3.1.14.	Канал .....	339
6.3.2.	Безопасность.....	344
6.3.2.1.	Панель свойств Безопасность.....	346



6.3.2.2.	Пользователи .....	349
6.3.2.3.	Роли .....	350
6.3.2.4.	Права доступа .....	352
6.3.2.5.	Архивы.....	353
6.3.2.6.	Настройки пользователей .....	356
6.3.2.7.	Настройка прав пользователей в режиме.....	358
	разработки.....	358
6.3.2.7.1.	Получение параметров текущего пользователя.....	363
6.3.2.7.2.	Задание начального окна для пользователя .....	364
6.3.2.7.3.	Журнал действий пользователей.....	365
6.3.2.7.4.	Настройка пользователя по умолчанию.....	367
6.3.2.8.	Работа с пользователями в режиме исполнения .....	368
6.3.2.9.	Active Directory.....	371
6.3.3.	Получение и отправка данных .....	380
6.3.3.1.	Стандартные протоколы .....	381
6.3.3.1.1.	Modbus .....	382
6.3.3.1.1.1.	Modbus TCP .....	383
6.3.3.1.1.2.	Modbus RTU.....	389
6.3.3.1.1.3.	Свойства модулей Modbus TCP и Modbus RTU .....	395
6.3.3.1.1.4.	Свойства каналов Modbus .....	399
6.3.3.1.1.5.	Modbus RTU over TCP.....	401
6.3.3.1.2.	DCON.....	404
6.3.3.1.2.1.	Свойства протокола DCON .....	406
6.3.3.1.2.2.	Модуль DCON.....	407
6.3.3.1.3.	OPC UA .....	410
6.3.3.1.3.1.	Свойства OPC UA.....	413
6.3.3.1.3.2.	Диалог подключения к серверу OPC UA .....	417

6.3.3.1.4.	OPC DA .....	421
6.3.3.1.4.1.	Свойства OPC DA .....	424
6.3.3.1.4.2.	Диалог подключения к серверу OPC DA .....	426
6.3.3.1.5.	OPC HDA.....	428
6.3.3.1.5.1.	Свойства OPC HDA.....	430
6.3.3.1.5.2.	Диалог подключения к серверу OPC HDA.....	433
6.3.3.2.	Дополнительные протоколы .....	434
6.3.3.2.1.	SiemensPLC .....	437
6.3.3.2.1.1.	Свойства SiemensPLC .....	441
6.3.3.2.1.2.	Свойства каналов SiemensPLC .....	444
6.3.3.2.1.3.	Диалог импорта каналов SiemensPLC .....	446
6.3.3.2.2.	Тепловычислители Теплоком.....	449
6.3.3.2.2.1.	VKT9 .....	449
6.3.3.2.2.2.	VKT7 .....	460
6.3.3.2.3.	IEC61850 .....	471
6.3.3.2.3.1.	Свойства IEC61850 .....	476
6.3.3.2.4.	Меркурий 230 .....	479
6.3.3.2.4.1.	Mercury_230_AR .....	481
6.3.3.2.4.2.	Mercury_230_ART2_F.....	482
6.3.3.2.4.3.	Свойства модулей Меркурий 230 .....	484
6.3.3.2.4.4.	Свойства протокола Меркурий 230 .....	485
6.3.3.2.5.	MQTT.....	486
6.3.3.2.5.1.	Свойства MQTT .....	490
6.3.3.2.6.	SNMP Client .....	492
6.3.3.2.6.1.	Свойства SNMP Client .....	495
6.3.3.2.6.2.	Свойства канала SNMP .....	498
6.3.3.2.7.	IEC104 .....	499

6.3.3.2.7.1.	Свойства IEC104 .....	502
6.3.3.2.7.2.	Свойства канала IEC104.....	505
6.3.3.2.8.	Протоколы для работы с базами данных.....	507
6.3.3.2.8.1.	Протокол PostgreSQL.....	507
6.3.3.2.8.2.	Протокол MSSQL.....	510
6.3.3.2.8.3.	Протокол MySQL.....	512
6.3.3.2.8.4.	Свойства протоколов для работы с базами данных.....	514
6.3.3.2.8.5.	Свойства Запроса к базе данных.....	516
6.3.3.3.	MasterSCADA 4D в роли SLave.....	518
6.3.3.3.1.	Внешние каналы.....	518
6.3.3.3.1.1.	MasterSCADA 4D как сервер OPC DA.....	522
6.3.3.3.1.2.	MasterSCADA 4D в роли Modbus Slave.....	529
6.3.3.3.1.3.	IEC104 SLave.....	534
6.3.3.3.2.	MasterSCADA 4D как сервер OPC UA.....	537
6.3.3.3.3.	IEC61850 Slave.....	545
6.3.3.4.	Типовые каналы обмена.....	551
6.3.3.5.	Отказ.....	552
6.3.3.6.	Новые типы протоколов и модулей.....	552
6.3.3.7.	Рекомендации по настройке DCOM.....	552
6.4.	Дерево объектов.....	552
6.4.1.	Объект.....	553
6.4.1.1.	Панель свойств Объекта.....	557
6.4.2.	Экземпляр объекта.....	558
6.4.3.	Параметры объектов.....	560
6.4.4.	Папка.....	561
6.4.5.	Тег.....	561
6.4.5.1.	Панель свойств Тега.....	562

6.4.5.2.	Стандартные теги .....	563
6.4.5.2.1.	Вещественный тег.....	564
6.4.5.2.2.	Целый тег .....	564
6.4.5.2.3.	Логический тег .....	564
6.4.5.3.	Новый тип тега .....	564
6.4.6.	Ресурсы.....	564
6.4.6.1.	Программы.....	565
6.4.6.2.	Окна .....	565
6.4.6.3.	Сообщения .....	566
6.4.6.4.	Отчеты .....	566
6.4.7.	Установка пароля на содержимое объектов .....	566
6.5.	Дерево библиотек .....	569
6.5.1.	Библиотека Стандартная .....	571
6.5.1.1.	Типы данных .....	573
6.5.1.1.1.	SYSTEM_LREAL_PARAM .....	574
6.5.1.1.2.	SYSTEM_BOOL_PARAM .....	574
6.5.1.1.3.	SYSTEM_DINT_PARAM.....	574
6.5.1.1.4.	SYSTEM_STRING_PARAM.....	574
6.5.1.1.5.	SYSTEM_COUNTER_PARAM.....	574
6.5.1.1.6.	StatusCode .....	575
6.5.2.	Пользовательская библиотека .....	578
6.5.2.1.	Создание/подключение библиотеки.....	578
6.5.2.2.	Работа с типом Объекта .....	581
6.5.3.	Диалог Обновить библиотеки .....	584
6.5.4.	Библиотека BaseObjects .....	585
6.5.4.1.	BaseObjects.ФБ.Системные .....	585
6.5.4.1.1.	SetDateAndTime .....	586

6.5.4.1.3.	READ_ARCHIVE_DATA.....	593
6.5.4.1.4.	PlaySound.....	595
6.5.4.1.5.	SEND_SMS.....	596
6.5.4.1.6.	ReportFB.....	599
6.5.4.1.7.	GET_CURRENT_USERNAME .....	602
6.5.4.1.8.	SysProcessCreate .....	603
6.5.4.1.9.	SysProcessCreateWithResult.....	604
6.5.4.1.10.	READ_SMS.....	606
6.5.4.1.11.	GetObjectName.....	608
6.5.4.1.12.	SysProcessCreateAsync .....	609
6.5.4.1.13.	SysProcessCreateWithResultAsync .....	610
6.5.4.1.14.	COMPort.....	612
6.5.4.1.15.	COMPortByte .....	614
6.5.4.1.16.	COMPortByteSync .....	616
6.5.4.1.17.	EventsCounter .....	618
6.5.4.1.18.	WriteArchiveData.....	619
6.5.4.1.19.	FireBaseEvent .....	621
6.5.4.2.	BaseObjects.ФБ.Users .....	622
6.5.4.2.1.	UsersGet.....	623
6.5.4.2.2.	UsersAdd .....	625
6.5.4.2.3.	UsersSetGroups.....	626
6.5.4.2.4.	UsersDelete .....	628
6.5.4.2.5.	UsersGetGroups .....	629
6.5.4.2.6.	UsersSetPassword .....	630
6.5.4.2.7.	UsersRename .....	632
6.5.4.2.8.	UsersGetSettings.....	633
6.5.4.2.9.	UsersSetSettings.....	635

6.5.4.2.10.	UsersChangeOwnPassword.....	638
6.5.4.2.11.	UsersGetADUserGroups.....	640
6.6.	Параметры.....	641
6.6.1.	Панель свойств Параметров.....	642
6.6.2.	Окно настройки параметра.....	643
6.6.3.	Создание нового типа данных.....	647
6.6.3.1.	Новый тип данных Массив.....	649
6.6.3.2.	Новый тип данных Перечисление.....	653
6.6.3.3.	Новый тип данных Диапазон.....	654
6.6.3.4.	Новый тип данных Структура.....	655
6.7.	Работа со шкалами.....	656
6.7.1.	Шкала AI.....	657
6.7.2.	Шкала DI.....	661
6.7.3.	Шкала.....	663
6.7.4.	Динамическая шкала.....	664
6.7.5.	Новый тип шкалы.....	664
6.8.	Свойства.....	664
6.8.1.	Категория свойств Общие.....	664
6.8.2.	Категория Задача.....	669
6.8.3.	Категория Масштабирование.....	673
6.8.4.	Категория свойств Служебные.....	675
6.8.5.	Категория свойств Разрешения.....	677
6.9.	Работа с сообщениями.....	680
6.9.1.	Типы стандартных сообщений.....	682
6.9.1.1.	Тревога.....	682
6.9.1.2.	Контроль границ.....	686
6.9.1.3.	Контроль скорости изменения.....	687

6.9.1.4.	Базовое сообщение.....	687
6.9.1.5.	Действие пользователей .....	688
6.9.2.	Новый тип тревоги.....	689
6.9.3.	Архивирование сообщений.....	690
6.10.	Настройка сетевых проектов .....	691
6.10.1.	Резервирование узлов .....	692
7.	Программирование в MasterSCADA 4D .....	696
7.1.	Общие элементы программ .....	697
7.1.1.	Параметры программ .....	697
7.1.2.	Ключевые слова.....	698
7.1.3.	Обработка En/EnO .....	699
7.1.4.	Типы данных .....	699
7.1.4.1.	Элементарные типы данных .....	699
7.1.4.2.	Производные типы данных .....	700
7.1.4.3.	Специальные типы данных.....	701
7.2.	Общие средства редактирования программ .....	702
7.2.1.	Структура программы в дереве .....	702
7.2.2.	Клеммники.....	703
7.2.2.1.	Клеммник входов .....	703
7.2.2.2.	Клеммник выходов .....	704
7.2.3.	Легенда.....	705
7.2.3.1.	Легенда. Вкладка Локальные .....	705
7.2.3.2.	Легенда. Вкладка Параметры.....	705
7.2.3.3.	Легенда. Вкладка Заголовков.....	706
7.2.3.3.1.	Легенда. Вкладка Ошибки .....	706
7.2.3.4.	Легенда. Вкладка Скомпилированный .....	706
7.2.3.5.	Легенда. Вкладка Элементы .....	707

7.2.3.6.	Легенда. Вкладка Связи .....	707
7.2.3.7.	Легенда. Вкладка Палитра .....	707
7.2.3.8.	Легенда. Вкладка Действия .....	708
7.2.3.9.	Легенда. Вкладка Условия .....	708
7.2.3.10.	Легенда. Вкладка Надписи .....	708
7.2.4.	Редактирование графических языков .....	708
7.2.4.1.	Редактирование диаграмм в дереве .....	708
7.2.4.2.	Масштабирование диаграмм .....	709
7.2.4.3.	Прокрутка диаграмм .....	709
7.2.4.4.	Миникарта .....	710
7.2.4.5.	Связывание переменных на FBD- и LD-диаграммах .....	710
7.3.	Редактор ST .....	711
7.3.1.	Вызов функций и функциональных блоков .....	711
7.3.2.	Функции преобразования типа .....	715
7.3.3.	ST-подсказчик .....	716
7.3.4.	Поиск и замена в тексте ST .....	717
7.3.5.	Экспорт и импорт программ ST .....	717
7.3.6.	Синтаксис Master ST .....	718
7.3.6.1.	Операторы Master ST .....	719
7.3.6.1.1.	Оператор RETURN .....	719
7.3.6.1.2.	Оператор IF .....	719
7.3.6.1.3.	Оператор CASE .....	720
7.3.6.1.4.	Оператор WHILE .....	721
7.3.6.1.5.	Оператор REPEAT .....	722
7.3.6.1.6.	Оператор FOR .....	722
7.3.6.1.7.	Оператор EXIT .....	723
7.3.6.1.8.	Операторы определения переменных .....	723



7.3.6.1.9.	Оператор индексирования элементов массива .....	727
7.3.6.2.	Инфиксные операторы.....	728
7.3.6.3.	Особенности вычислений.....	729
7.3.6.3.1.	Особенности битовых строк .....	729
7.3.6.3.2.	Работа с отдельными битами.....	729
7.3.6.4.	Константы Master ST.....	730
7.3.6.4.1.	Числовые константы .....	730
7.3.6.4.2.	Временные константы .....	731
7.3.6.4.3.	Строковые константы.....	732
7.3.6.5.	Лексическая структура .....	734
7.3.6.5.1.	Идентификаторы .....	735
7.3.6.5.2.	Разделители .....	735
7.3.6.5.3.	Комментарии .....	735
7.4.	Редактор FBD .....	735
7.4.1.	Редактирование FBD-диаграммы .....	736
7.4.2.	FBD-блок.....	739
7.4.2.1.	Панель свойств FBD-блока.....	740
7.4.3.	Слои диаграммы.....	741
7.4.4.	Операции со связями FBD-диаграммы.....	741
7.4.4.1.	Отображение связей .....	741
7.4.4.2.	Маршрутные точки.....	742
7.4.5.	Панель инструментов FBD-редактора .....	743
7.4.5.1.	Панель Слои .....	743
7.4.5.2.	Панель Режимы .....	744
7.4.5.3.	Панель Сетка .....	745
7.4.5.4.	Панель Выравнивание .....	746
7.4.5.5.	Порядок исполнения FBD-блоков .....	746

7.5.	Редактор LD.....	748
7.5.1.	Слои LD-диаграммы .....	749
7.5.2.	Редактирование LD-диаграммы.....	749
7.5.2.1.	Размещение элементов алгоритма на LD-диаграмме .....	750
7.5.2.2.	Перемещение элемента LD-алгоритма .....	750
7.5.2.3.	Изменение типа контакта/катушки после размещения.....	751
7.5.2.4.	Операции со связанными переменными.....	751
7.5.3.	Панель инструментов LD-редактора.....	752
7.5.4.	Палитра LD .....	752
7.5.4.1.	Силовая линия .....	753
7.5.4.2.	LD-блок .....	753
7.5.4.3.	LD-ветвление.....	754
7.5.4.4.	LD-терминатор .....	754
7.5.4.5.	Другие элементы LD .....	755
7.6.	Редактор SFC .....	757
7.6.1.	Слои SFC-диаграммы.....	758
7.6.2.	Редактирование SFC-диаграммы .....	758
7.6.2.1.	Размещение элементов алгоритма на SFC-диаграмме.....	758
7.6.3.	Панель инструментов SFC-редактора .....	759
7.6.4.	Палитра SFC.....	759
7.6.4.1.1.	Библиотека SFC .....	759
7.6.4.1.1.1.	SFC-ветка .....	760
7.6.4.1.2.	SFC-шаг .....	760
7.6.4.1.3.	SFC-условие .....	761
7.6.4.1.4.	SFC-переход .....	761
7.6.4.1.5.	SFC-терминатор .....	761
7.6.4.1.6.	SFC-ветвление выбора .....	762

7.6.4.1.7.	Параллельное SFC-ветвление.....	763
7.6.5.	Задание SFC-действий.....	763
7.6.5.1.	Диалог редактирования SFC-действия .....	764
7.6.6.	Задание SFC-условий.....	767
7.6.6.1.	Диалог редактирования SFC-условия .....	768
7.7.	Функции и ФБ библиотеки Стандартная .....	769
7.7.1.	Библиотека Стандартные функции .....	769
7.7.1.1.	Булевы функции .....	769
7.7.1.2.	Математические функции .....	771
7.7.1.3.	Функции сдвига .....	774
7.7.1.4.	Функции сравнения.....	775
7.7.1.5.	Тригонометрические функции .....	777
7.7.1.6.	Функции работы со строками.....	779
7.7.1.7.	Функции работы с датой-временем .....	782
7.7.1.7.1.	CONCAT_DATE .....	791
7.7.1.7.2.	CONCAT_TOD .....	792
7.7.1.7.3.	CONCAT_DT.....	793
7.7.1.7.4.	SPLIT_DATE .....	794
7.7.1.7.5.	SPLIT_TOD .....	795
7.7.1.7.6.	SPLIT_DT.....	796
7.7.1.8.	Логарифмические функции .....	797
7.7.1.9.	Функции выбора .....	797
7.7.1.10.	Функции конвертации.....	799
7.7.1.11.	Функции работы с массивами .....	800
7.7.1.12.	Служебные функции .....	804
7.7.2.	Библиотека Стандартные ФБ.....	805
7.7.2.1.	ФБ 'Детекторы фронтов'.....	805

7.7.2.2.	ФБ 'Таймеры' .....	806
7.7.2.3.	ФБ 'Триггеры' .....	807
7.7.2.4.	ФБ 'Счетчики' .....	807
8.	Создание окон для клиента визуализации .....	809
8.1.	Окно .....	813
8.1.1.	Структура окна .....	815
8.1.2.	Свойства Окна .....	817
8.2.	Редактор HMI .....	818
8.2.1.	Панель инструментов редактора HMI .....	819
8.2.1.1.	Панель Группа .....	820
8.2.1.2.	Панель Просмотр .....	821
8.2.1.3.	Панель Z-порядок .....	822
8.2.1.4.	Панель Поворот .....	823
8.2.2.	Легенда в редакторе HMI .....	823
8.2.2.1.	Легенда HMI. Вкладка Палитра .....	824
8.2.2.2.	Легенда HMI. Вкладка Локальные .....	824
8.2.2.3.	Легенда HMI. Вкладка Параметры .....	825
8.2.2.4.	Легенда HMI. Вкладка Ошибки .....	825
8.2.2.5.	Легенда HMI. Вкладка Элементы .....	826
8.2.2.6.	Легенда HMI. Вкладка Связи .....	826
8.2.2.7.	Легенда HMI. Вкладка События .....	827
8.2.2.8.	Легенда HMI. Вкладка Медиа .....	827
8.2.2.9.	Легенда HMI. Вкладка Триггеры .....	827
8.2.3.	HMI. Размещение элементов .....	828
8.2.3.1.	Размещение Тега, Канала, Объекта в окне .....	829
8.2.3.2.	Размещение параметра в окне .....	830
8.2.3.2.1.	Перетаскивание параметра в окно правой кнопкой мыши .....	831

8.2.3.2.1.1.	Перетаскивание параметра в окно левой кнопкой мыши.....	832
8.2.4.	НМІ. Программирование в окнах.....	833
8.2.4.1.	Динамизация .....	834
8.2.4.1.1.	Конвертация значений.....	835
8.2.4.1.1.1.	Конвертер значений.....	837
8.2.4.1.1.2.	Неявная конвертация значений .....	839
8.2.4.2.	НМІ. Действия по условию .....	839
8.2.4.3.	НМІ. Действия по событию .....	842
8.2.4.3.1.	НМІ. События .....	842
8.2.4.3.2.	НМІ. Действия .....	846
8.2.4.3.2.1.	Действие Вызвать Программу .....	847
8.2.4.3.2.2.	Действие Открыть окно.....	848
8.2.4.3.2.3.	Действие Закрyть окно .....	848
8.2.4.3.2.4.	Действие Открыть всплывающее окно.....	849
8.2.4.3.2.5.	Действие Установить параметр.....	854
8.2.4.3.2.6.	Действие Показать сообщение .....	854
8.2.4.3.2.7.	Действие Завершить сессию.....	855
8.2.4.3.2.8.	Действие Сменить пользователя .....	855
8.2.4.3.2.9.	Действие Вперед .....	856
8.2.4.3.2.10.	Действие Назад.....	856
8.2.4.3.2.11.	Действие Войти в полноэкранный режим.....	857
8.2.4.3.2.12.	Действие Выйти из полноэкранного режима.....	857
8.2.4.3.2.13.	Действие Открыть окно печати .....	857
8.2.4.3.2.14.	Действие Открыть адрес .....	858
8.2.4.3.2.15.	Действие Показать строку статуса.....	859
8.2.4.3.2.16.	Действие Вызвать событие .....	859
8.2.4.3.2.17.	Действие Вызвать метод.....	860

8.2.5.	Клеммники в редакторе HMI.....	860
8.2.6.	Диалоговое окно Выбор цвета .....	862
8.2.7.	Создание цвета пользователя .....	866
8.3.	Библиотека HMI.....	869
8.3.1.	Категория Диалог .....	870
8.3.1.1.	Текст.....	871
8.3.1.2.	Текстовый ввод.....	872
8.3.1.3.	Пароль .....	874
8.3.1.4.	Инкремент.....	876
8.3.1.5.	Кнопка .....	878
8.3.1.6.	Кнопка с фиксацией .....	880
8.3.1.7.	Радиокнопка .....	883
8.3.1.8.	Флаг .....	886
8.3.1.9.	Выпадающий список .....	887
8.3.1.10.	Часы .....	889
8.3.1.11.	ДатаВремя.....	891
8.3.1.12.	Время.....	896
8.3.1.13.	Интервал .....	900
8.3.1.14.	Горизонтальный ползунок.....	904
8.3.1.15.	Вертикальный ползунок .....	907
8.3.1.16.	Горизонтальный прогресс .....	908
8.3.1.17.	Вертикальный прогресс .....	910
8.3.1.18.	Кнопка выбора цвета .....	911
8.3.1.19.	Стрелочный прибор .....	914
8.3.2.	Категория Контролы.....	917
8.3.2.1.	Тренд .....	918
8.3.2.2.	Свойства Тренда .....	920

8.3.2.3.	График XY .....	928
8.3.2.3.1.	Свойства Графика XY .....	931
8.3.2.4.	Журнал и Архивный журнал .....	937
8.3.2.4.1.	Настройка журнала .....	941
8.3.2.4.1.1.	Окно настройки журнала. Столбцы таблицы .....	942
8.3.2.4.1.2.	Окно настройки журнала. Столбцы легенды .....	950
8.3.2.4.1.3.	Окно настройки журнала. Фильтры .....	950
8.3.2.4.1.4.	Окно настройки журнала. Стиль строки .....	954
8.3.2.4.1.5.	Окно настройки журнала. Стиль заголовка .....	956
8.3.2.4.1.6.	Свойства Журнала .....	957
8.3.2.4.1.7.	Методы для работы с журналом .....	961
8.3.2.4.2.	Работа с журналом в клиенте визуализации .....	962
8.3.2.4.2.1.	Журнал в клиенте визуализации при HMI v1 .....	963
8.3.2.4.2.2.	Журнал в клиенте визуализации при HMI v2 .....	968
8.3.2.5.	Индикатор событий .....	974
8.3.2.6.	Таблица данных .....	977
8.3.2.6.1.	Редактор таблицы данных. Столбцы таблицы .....	982
8.3.2.6.2.	Редактор таблицы данных. Стиль строки .....	985
8.3.2.6.3.	Редактор таблицы данных. Стили ячейки .....	987
8.3.2.6.4.	Редактор таблицы данных. Стиль заголовка .....	990
8.3.2.6.5.	Событие Клик по ячейке .....	991
8.3.3.	Категория Панели .....	994
8.3.3.1.	Общие инструменты панелей .....	994
8.3.3.2.	Панель .....	995
8.3.3.3.	Стековая панель .....	997
8.3.3.4.	Панель док .....	1003
8.3.3.5.	Панель вкладок .....	1007

8.3.3.6.	Контейнер окна .....	1010
8.3.4.	Категория Примитивы .....	1016
8.3.4.1.	Линия .....	1017
8.3.4.2.	Граф .....	1018
8.3.4.3.	Ломаная линия .....	1022
8.3.4.4.	Полигон .....	1025
8.3.4.5.	Овал, Прямоугольник, Треугольник .....	1029
8.3.4.6.	Крест .....	1032
8.3.4.7.	Стрелка .....	1033
8.3.4.8.	Картинка .....	1033
8.3.4.9.	Графический элемент Видео .....	1035
8.3.5.	Темы библиотеки HMI .....	1037
8.3.6.	Категория BaseObjects .....	1038
8.3.6.1.	BaseObjects. Элементы функциональных схем .....	1040
8.3.7.	Категория Images .....	1040
8.4.	HMI. Свойства .....	1041
8.4.1.	Свойство Формат значений .....	1041
8.4.2.	Свойство Маска ввода .....	1041
8.4.3.	Категория свойств Экран .....	1043
8.4.4.	Категория свойств Рамка/Линия .....	1045
8.4.5.	Категория свойств Клавиатура .....	1046
8.4.6.	Категория свойств Внешний вид .....	1047
8.4.7.	Категория свойств Расположение .....	1050
8.4.8.	Категория свойств Текст .....	1051
8.4.9.	Категория свойств Трансформация .....	1053
8.5.	Рекомендации по созданию окон .....	1054
8.5.1.	Создание произвольных фигур .....	1055



8.5.2.	Стартовое окно .....	1056
8.5.3.	Кеширование окон .....	1058
8.5.4.	Особенности создания окон в библиотеке .....	1062
8.5.5.	Особенности работы с элементами Кнопка и Кнопка с фиксацией.....	1066
8.5.6.	Массивы структур в окнах.....	1071
8.5.7.	Создание пользовательской библиотеки примитивов .....	1082
8.5.8.	Работа с веб-камерой.....	1087
9.	Работа в режиме исполнения .....	1088
9.1.	Работа со средой исполнения .....	1088
9.1.1.	Рабочая папка среды исполнения .....	1088
9.1.2.	Подготовка проекта к запуску в среде исполнения .....	1091
9.1.3.	Автоматическая загрузка проекта .....	1093
9.1.4.	Ручная загрузка проекта в среду исполнения .....	1096
9.1.5.	Восстановление работы узла.....	1099
9.1.6.	MasterSCADA 4D Monitor .....	1099
9.1.6.1.	Окно настройки среды исполнения .....	1103
9.1.6.2.	Работа с процессом plc .....	1107
9.1.7.	Запуск исполнительной системы .....	1109
9.1.8.	Запуск нескольких узлов на одном компьютере .....	1109
9.1.9.	Обновление проекта в новой версии среды исполнения.....	1110
9.1.10.	Выбор опций среды исполнения .....	1111
9.1.11.	Коды ошибок среды исполнения.....	1113
9.1.12.	Панель узла .....	1115
9.1.12.1.	Вкладка управления узлом.....	1115
9.1.12.2.	Вкладка Конфигурация .....	1117
9.1.13.	Работа с файлом данных для горячего рестарта .....	1120
9.2.	Работа с клиентом визуализации .....	1120

9.2.1.	Выбор версии HMI .....	1124
9.2.2.	Окно Список ошибок .....	1125
9.2.3.	MasterSCADA 4D Client Monitor .....	1126
9.2.4.	Особенности первой автоматической загрузки проекта .....	1128
9.2.5.	Окно авторизации пользователей .....	1129
9.2.6.	Задание принтера и пути к сохраняемым файлам по умолчанию .....	1130
9.2.7.	Горячие клавиши в клиенте визуализации .....	1133
9.2.8.	Многомониторный режим .....	1134
9.3.	Отладка проекта .....	1135
9.3.1.	Режимы отладки.....	1135
9.3.1.1.	Непрерывная отладка .....	1136
9.3.2.	Отладка с шагом 1 цикл .....	1137
9.3.3.	Как получить диагностическую информацию среды исполнения .....	1137
9.3.4.	Подключение среды разработки к исполняемому узлу .....	1144
9.3.5.	Эмуляция устройства.....	1146
9.3.6.	Имитация значений параметров.....	1146
9.3.7.	Сообщения об ошибках .....	1147
10.	Библиотека OSCAT .....	1151
10.1.	OSCAT.Функции .....	1152
10.1.1.	OSCAT.Функции.Математика .....	1157
10.1.1.1.	D_TRUNC (OSCAT).....	1157
10.1.1.2.	DEC1 (OSCAT) .....	1157
10.1.1.3.	EVEN (OSCAT) .....	1157
10.1.1.4.	FLOOR (OSCAT).....	1158
10.1.1.5.	FLOOR2 (OSCAT).....	1158
10.1.1.6.	FRACT (OSCAT) .....	1158
10.1.1.7.	INC (OSCAT).....	1158

10.1.1.8.	INC1 (OSCAT).....	1159
10.1.1.9.	INC2 (OSCAT).....	1159
10.1.1.10.	MID3 (OSCAT).....	1160
10.1.1.11.	MODR (OSCAT).....	1160
10.1.1.12.	MODR_old (OSCAT).....	1161
10.1.1.13.	RAD (OSCAT).....	1161
10.1.1.14.	RDM (OSCAT) .....	1162
10.1.1.15.	SIGN_R (OSCAT) .....	1162
10.1.2.	OSCAT.Функции.Арифметические функции .....	1163
10.1.2.1.	F_LIN (OSCAT) .....	1163
10.1.2.2.	F_LIN2 (OSCAT) .....	1163
10.1.2.3.	F_POLY (OSCAT) .....	1164
10.1.2.4.	F_POWER (OSCAT) .....	1164
10.1.2.5.	F_QUAD (OSCAT).....	1164
10.1.2.6.	FRMP_B (OSCAT).....	1165
10.1.2.7.	LINEAR_INT (OSCAT) .....	1165
10.1.2.8.	POLYNOM_INT (OSCAT) .....	1166
10.1.3.	OSCAT.Функции.Геометрические функции.....	1167
10.1.3.1.	CIRCLE_A (OSCAT) .....	1167
10.1.3.2.	CIRCLE_C (OSCAT) .....	1168
10.1.3.3.	CIRCLE_SEG (OSCAT) .....	1168
10.1.3.4.	CONE_V (OSCAT) .....	1169
10.1.3.5.	ELLIPSE_A (OSCAT) .....	1169
10.1.3.6.	ELLIPSE_C (OSCAT) .....	1170
10.1.3.7.	SPHERE_V (OSCAT).....	1170
10.1.3.8.	TRIANGLE_A (OSCAT) .....	1170
10.1.4.	OSCAT.Функции.Дата и время .....	1171

10.1.4.1.	DAY_OF_WEEK (OSCAT) .....	1171
10.1.4.2.	DAY_OF_YEAR (OSCAT) .....	1171
10.1.4.3.	LEAP_OF_DATE (OSCAT) .....	1171
10.1.4.4.	MONTH_OF_DATE (OSCAT) .....	1172
10.1.4.5.	YEAR_OF_DATE (OSCAT) .....	1172
10.1.5.	OSCAT.Функции.Логические модули .....	1173
10.1.5.1.	BCDC_TO_INT (OSCAT).....	1173
10.1.5.2.	BIT_COUNT (OSCAT).....	1173
10.1.5.3.	BIT_LOAD_B (OSCAT) .....	1173
10.1.5.4.	BIT_LOAD_B2 (OSCAT) .....	1173
10.1.5.5.	BIT_LOAD_DW (OSCAT) .....	1174
10.1.5.6.	BIT_LOAD_DW2 (OSCAT) .....	1174
10.1.5.7.	BIT_LOAD_W (OSCAT).....	1175
10.1.5.8.	BIT_LOAD_W2 (OSCAT).....	1175
10.1.5.9.	BIT_OF_DWORD (OSCAT).....	1175
10.1.5.10.	BIT_TOGGLE_B (OSCAT) .....	1176
10.1.5.11.	BIT_TOGGLE_DW (OSCAT) .....	1176
10.1.5.12.	BIT_TOGGLE_W (OSCAT).....	1176
10.1.5.13.	BYTE_OF_BIT (OSCAT) .....	1177
10.1.5.14.	BYTE_OF_DWORD (OSCAT) .....	1177
10.1.5.15.	BYTE_TO_GRAY (OSCAT) .....	1178
10.1.5.16.	CHECK_PARITY (OSCAT) .....	1178
10.1.5.17.	DWORD_OF_BYTE (OSCAT).....	1178
10.1.5.18.	DWORD_OF_WORD (OSCAT) .....	1179
10.1.5.19.	GRAY_TO_BYTE (OSCAT) .....	1179
10.1.5.20.	INT_TO_BCDC (OSCAT).....	1179
10.1.5.21.	MUX_2 (OSCAT) .....	1179

10.1.5.22.	MUX_4 (OSCAT) .....	1180
10.1.5.23.	PARITY (OSCAT) .....	1180
10.1.5.24.	REFLECT (OSCAT) .....	1181
10.1.5.25.	REVERSE (OSCAT) .....	1181
10.1.5.26.	SHL1 (OSCAT) .....	1181
10.1.5.27.	SHR1 (OSCAT) .....	1181
10.1.5.28.	SWAP_BYTE (OSCAT).....	1182
10.1.5.29.	SWAP_BYTE2 (OSCAT).....	1182
10.1.5.30.	WORD_OF_BYTE (OSCAT) .....	1182
10.1.5.31.	WORD_OF_DWORD (OSCAT) .....	1183
10.1.6.	OSCAT.Функции.Преобразования .....	1183
10.1.6.1.	BFT_TO_MS (OSCAT).....	1183
10.1.6.2.	C_TO_F (OSCAT) .....	1183
10.1.6.3.	C_TO_K (OSCAT).....	1184
10.1.6.4.	F_TO_C (OSCAT) .....	1184
10.1.6.5.	F_TO_OM (OSCAT) .....	1184
10.1.6.6.	F_TO_PT (OSCAT) .....	1185
10.1.6.7.	GEO_TO_DEG (OSCAT).....	1185
10.1.6.8.	K_TO_C (OSCAT).....	1185
10.1.6.9.	KMH_TO_MS (OSCAT).....	1186
10.1.6.10.	MS_TO_BFT (OSCAT).....	1186
10.1.6.11.	MS_TO_KMH (OSCAT).....	1186
10.1.6.12.	OM_TO_F (OSCAT) .....	1186
10.1.6.13.	PT_TO_F (OSCAT) .....	1187
10.1.7.	OSCAT.Функции.Обработка сигналов .....	1187
10.1.7.1.	AIN (OSCAT).....	1187
10.1.7.2.	AOUT (OSCAT) .....	1189

10.1.7.3.	AOUT1 (OSCAT) .....	1190
10.1.7.4.	BYTE_TO_RANGE (OSCAT) .....	1191
10.1.7.5.	MIX (OSCAT) .....	1191
10.1.7.6.	MUX_R2 (OSCAT) .....	1192
10.1.7.7.	MUX_R4 (OSCAT) .....	1192
10.1.7.8.	OFFSET (OSCAT) .....	1193
10.1.7.9.	OFFSET2 (OSCAT) .....	1193
10.1.7.10.	VERRIDE (OSCAT) .....	1194
10.1.7.11.	RANGE_TO_BYTE (OSCAT).....	1195
10.1.7.12.	RANGE_TO_WORD (OSCAT).....	1195
10.1.7.13.	SCALE (OSCAT).....	1195
10.1.7.14.	SCALE_B (OSCAT) .....	1196
10.1.7.15.	SCALE_B2 (OSCAT) .....	1196
10.1.7.16.	SCALE_B4 (OSCAT) .....	1199
10.1.7.17.	SCALE_B8 (OSCAT) .....	1201
10.1.7.18.	SCALE_D (OSCAT) .....	1202
10.1.7.19.	SCALE_R (OSCAT) .....	1202
10.1.7.20.	SCALE_X2 (OSCAT).....	1203
10.1.7.21.	SCALE_X4 (OSCAT).....	1204
10.1.7.22.	SCALE_X8 (OSCAT).....	1204
10.1.7.23.	STAIR (OSCAT) .....	1205
10.1.7.24.	WORD_TO_RANGE (OSCAT).....	1206
10.1.8.	OSCAT.Функции.Датчики.....	1207
10.1.8.1.	MULTI_IN (OSCAT) .....	1207
10.1.8.2.	RES_NI (OSCAT) .....	1208
10.1.8.3.	RES_NTC (OSCAT) .....	1209
10.1.8.4.	RES_PT (OSCAT).....	1209

10.1.8.5.	RES_SI (OSCAT).....	1210
10.1.8.6.	SENSOR_INT (OSCAT) .....	1211
10.1.8.7.	TEMP_NI (OSCAT).....	1212
10.1.8.8.	TEMP_NTC (OSCAT).....	1212
10.1.8.9.	TEMP_PT (OSCAT) .....	1213
10.1.8.10.	TEMP_SI (OSCAT) .....	1214
10.1.9.	OSCAT.Функции.Модули измерения.....	1214
10.1.9.1.	T_PLC_MS (OSCAT) .....	1214
10.1.9.2.	T_PLC_US (OSCAT) .....	1215
10.1.10.	OSCAT.Функции.Модули регулирования.....	1216
10.1.10.1.	BAND_B (OSCAT) .....	1216
10.1.10.2.	CTRL_IN (OSCAT) .....	1217
10.1.10.3.	DEAD_BAND (OSCAT) .....	1218
10.1.10.4.	DEAD_ZONE (OSCAT) .....	1219
10.1.11.	OSCAT.Функции.Управление устройствами .....	1219
10.1.11.1.	MANUAL (OSCAT).....	1219
10.2.	OSCAT.Функциональные блоки .....	1220
10.2.1.	OSCAT.ФБ.Арифметические функции .....	1226
10.2.1.1.	FT_AVG (OSCAT).....	1226
10.2.1.2.	FT_MIN_MAX (OSCAT) .....	1227
10.2.1.3.	FT_RMP (OSCAT) .....	1227
10.2.2.	OSCAT.ФБ.Модули памяти .....	1228
10.2.2.1.	FIFO_16 (OSCAT) .....	1228
10.2.2.2.	FIFO_32 (OSCAT) .....	1230
10.2.2.3.	STACK_16 (OSCAT) .....	1230
10.2.2.4.	STACK_32 (OSCAT) .....	1230
10.2.3.	OSCAT.ФБ.Генераторы импульсов.....	1231

10.2.3.1.	A_TRIG (OSCAT).....	1231
10.2.3.2.	B_TRIG (OSCAT).....	1231
10.2.3.3.	CLICK_CNT (OSCAT).....	1232
10.2.3.4.	CLICK_DEC (OSCAT).....	1233
10.2.3.5.	CLK_DIV (OSCAT).....	1234
10.2.3.6.	CLK_N (OSCAT).....	1235
10.2.3.7.	CLK_PRG (OSCAT).....	1236
10.2.3.8.	CLK_PULSE (OSCAT).....	1236
10.2.3.9.	CYCLE_4 (OSCAT).....	1238
10.2.3.10.	D_TRIG (OSCAT).....	1239
10.2.3.11.	GEN_BIT (OSCAT).....	1239
10.2.3.12.	GEN_SQ (OSCAT).....	1241
10.2.3.13.	SCHEDULER (OSCAT).....	1242
10.2.3.14.	SCHEDULER_2 (OSCAT).....	1244
10.2.3.15.	SEQUENCE_4 (OSCAT).....	1245
10.2.3.16.	SEQUENCE_64 (OSCAT).....	1251
10.2.3.17.	SEQUENCE_8 (OSCAT).....	1253
10.2.3.18.	TMAX (OSCAT).....	1254
10.2.3.19.	TMIN (OSCAT).....	1255
10.2.3.20.	TOF_1 (OSCAT).....	1255
10.2.3.21.	TONOF (OSCAT).....	1256
10.2.3.22.	TP_1 (OSCAT).....	1257
10.2.3.23.	TP_1D (OSCAT).....	1258
10.2.3.24.	TP_X (OSCAT).....	1259
10.2.4.	OSCAT.ФБ.Логические модули.....	1259
10.2.4.1.	BYTE_TO_BITS (OSCAT).....	1259
10.2.4.2.	DEC_2 (OSCAT).....	1260



10.2.4.3.	DEC_4 (OSCAT) .....	1261
10.2.4.4.	DEC_8 (OSCAT) .....	1262
10.2.4.5.	MATRIX (OSCAT).....	1263
10.2.5.	OSCAT.ФБ.Триггеры .....	1266
10.2.5.1.	COUNT_BR (OSCAT) .....	1266
10.2.5.2.	COUNT_BR_plcd.....	1267
10.2.5.3.	COUNT_DR (OSCAT) .....	1268
10.2.5.4.	COUNT_DR_plcd .....	1268
10.2.5.5.	FF_D2E (OSCAT) .....	1269
10.2.5.6.	FF_D4E (OSCAT) .....	1270
10.2.5.7.	FF_DRE (OSCAT) .....	1270
10.2.5.8.	FF_JKE (OSCAT).....	1271
10.2.5.9.	FF_RSE (OSCAT).....	1271
10.2.5.10.	LTCH (OSCAT) .....	1272
10.2.5.11.	LATCH4 (OSCAT) .....	1273
10.2.5.12.	SELECT_8 (OSCAT) .....	1273
10.2.5.13.	SHR_4E (OSCAT) .....	1274
10.2.5.14.	SHR_4UDE (OSCAT) .....	1275
10.2.5.15.	SHR_8PLE (OSCAT) .....	1276
10.2.5.16.	SHR_8UDE (OSCAT) .....	1277
10.2.5.17.	STORE_8 (OSCAT) .....	1277
10.2.5.18.	TOGGLE (OSCAT) .....	1278
10.2.6.	OSCAT.ФБ.Генераторы сигналов.....	1278
10.2.6.1.	_RMP_B (OSCAT).....	1278
10.2.6.2.	_RMP_NEXT (OSCAT) .....	1281
10.2.6.3.	_RMP_W (OSCAT) .....	1282
10.2.6.4.	GEN_PULSE (OSCAT) .....	1283

10.2.6.5.	GEN_PW2 (OSCAT).....	1284
10.2.6.6.	GEN_RDM (OSCAT) .....	1285
10.2.6.7.	GEN_RDT (OSCAT).....	1286
10.2.6.8.	GEN_RMP (OSCAT).....	1286
10.2.6.9.	GEN_SIN (OSCAT) .....	1287
10.2.6.10.	GEN_SQR (OSCAT).....	1289
10.2.6.11.	PWM_DC (OSCAT).....	1290
10.2.6.12.	PWM_PW (OSCAT).....	1291
10.2.6.13.	RMP_B (OSCAT).....	1292
10.2.6.14.	RMP_SOFT (OSCAT).....	1295
10.2.6.15.	RMP_W (OSCAT) .....	1297
10.2.7.	OSCAT.ФБ.Обработка сигналов .....	1298
10.2.7.1.	AIN1 (OSCAT).....	1298
10.2.7.2.	DELAY (OSCAT) .....	1300
10.2.7.3.	DELAY_4 (OSCAT) .....	1302
10.2.7.4.	FADE (OSCAT).....	1304
10.2.7.5.	FILTER_DW (OSCAT).....	1306
10.2.7.6.	FILTER_I (OSCAT).....	1306
10.2.7.7.	FILTER_MAV_DW (OSCAT).....	1307
10.2.7.8.	FILTER_MAV_W (OSCAT) .....	1308
10.2.7.9.	FILTER_W (OSCAT) .....	1309
10.2.7.10.	FILTER_WAV (OSCAT).....	1309
10.2.7.11.	SEL2_OF_3 (OSCAT).....	1311
10.2.7.12.	SEL2_OF_3B (OSCAT) .....	1312
10.2.7.13.	SH (OSCAT) .....	1312
10.2.7.14.	SH_1 (OSCAT) .....	1313
10.2.7.15.	SH_2 (OSCAT) .....	1314

10.2.7.16.	SH_T (OSCAT) .....	1316
10.2.7.17.	STAIR2 (OSCAT) .....	1317
10.2.7.18.	TREND (OSCAT) .....	1319
10.2.7.19.	TREND_DW (OSCAT) .....	1319
10.2.8.	OSCAT.ФБ.Модули измерения.....	1320
10.2.8.1.	ALARM_2 (OSCAT).....	1320
10.2.8.2.	BAR_GRAPH (OSCAT) .....	1321
10.2.8.3.	CALIBRATE (OSCAT) .....	1324
10.2.8.4.	CALIBRATE_mplc .....	1326
10.2.8.5.	CYCLE_TIME (OSCAT) .....	1327
10.2.8.6.	DT_SIMU (OSCAT) .....	1328
10.2.8.7.	FLOW_METER (OSCAT) .....	1329
10.2.8.8.	M_D (OSCAT).....	1331
10.2.8.9.	M_T (OSCAT) .....	1332
10.2.8.10.	M_TX (OSCAT) .....	1333
10.2.8.11.	ONTIME (OSCAT) .....	1334
10.2.8.12.	TC_MS (OSCAT) .....	1334
10.2.8.13.	TC_S (OSCAT).....	1335
10.2.8.14.	TC_US (OSCAT) .....	1335
10.2.8.15.	METER_STAT (OSCAT) .....	1336
10.2.9.	OSCAT.ФБ.Преобразования .....	1337
10.2.9.1.	ASTRO (OSCAT).....	1337
10.2.9.2.	ENERGY (OSCAT) .....	1338
10.2.9.3.	LENGTH (OSCAT) .....	1339
10.2.9.4.	PRESSURE (OSCAT) .....	1340
10.2.9.5.	SPEED (OSCAT) .....	1341
10.2.9.6.	TEMPERATURE (OSCAT) .....	1342

10.2.10.	OSCAT.ФБ.Модули регулирования.....	1343
10.2.10.1.	CONTROL_SET1 (OSCAT) .....	1343
10.2.10.2.	CONTROL_SET2 (OSCAT) .....	1345
10.2.10.3.	CTRL_OUT (OSCAT).....	1348
10.2.10.4.	CTRL_PI (OSCAT).....	1349
10.2.10.5.	CTRL_PID (OSCAT) .....	1351
10.2.10.6.	CTRL_PWM (OSCAT).....	1353
10.2.10.7.	DEAD_BAND_A (OSCAT).....	1355
10.2.10.8.	DEAD_ZONE2 (OSCAT) .....	1356
10.2.10.9.	FT_DERIV (OSCAT) .....	1357
10.2.10.10.	FT_IMP (OSCAT) .....	1358
10.2.10.11.	FT_INT (OSCAT) .....	1360
10.2.10.12.	FT_PD (OSCAT) .....	1361
10.2.10.13.	FT_PDT1 (OSCAT) .....	1363
10.2.10.14.	FT_PI (OSCAT).....	1363
10.2.10.15.	FT_PID (OSCAT) .....	1365
10.2.10.16.	FT_PIDW (OSCAT).....	1367
10.2.10.17.	FT_PIDWL (OSCAT) .....	1368
10.2.10.18.	FT_PIW (OSCAT) .....	1369
10.2.10.19.	FT_PIWL (OSCAT) .....	1371
10.2.10.20.	FT_PT1 (OSCAT).....	1372
10.2.10.21.	FT_PT2 (OSCAT).....	1373
10.2.10.22.	FT_TN16 (OSCAT) .....	1375
10.2.10.23.	FT_TN64 (OSCAT) .....	1375
10.2.10.24.	FT_TN8 (OSCAT) .....	1376
10.2.10.25.	HYST (OSCAT) .....	1376
10.2.10.26.	HYST_1 (OSCAT) .....	1378

10.2.10.27.	HYST_2 (OSCAT) .....	1378
10.2.10.28.	HYST_3 (OSCAT) .....	1379
10.2.10.29.	INTEGRATE (OSCAT) .....	1380
10.2.11.	OSCAT.ФБ.Управление устройствами .....	1381
10.2.11.1.	DRIVER_1 (OSCAT).....	1381
10.2.11.2.	DRIVER_4 (OSCAT).....	1382
10.2.11.3.	DRIVER_4C (OSCAT) .....	1382
10.2.11.4.	FLOW_CONTROL (OSCAT) .....	1384
10.2.11.5.	FT_PROFILE (OSCAT) .....	1385
10.2.11.6.	INC_DEC (OSCAT) .....	1387
10.2.11.7.	INTERLOCK_4 (OSCAT) .....	1389
10.2.11.8.	INTERLOCK (OSCAT) .....	1390
10.2.11.9.	MANUAL_1 (OSCAT).....	1391
10.2.11.10.	MANUAL_2 (OSCAT).....	1392
10.2.11.11.	MANUAL_4 (OSCAT).....	1393
10.2.11.12.	PARSET (OSCAT) .....	1394
10.2.11.13.	PARSET2 (OSCAT) .....	1396
10.2.11.14.	SIGNAL (OSCAT).....	1397
10.2.11.15.	SIGNAL_4 (OSCAT).....	1398
10.2.11.16.	SRAMP (OSCAT).....	1399
10.2.11.17.	TUNE (OSCAT) .....	1401
10.2.11.18.	TUNE2 (OSCAT) .....	1403

# 1. Установка

## 1.1. Состав MasterSCADA 4D

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

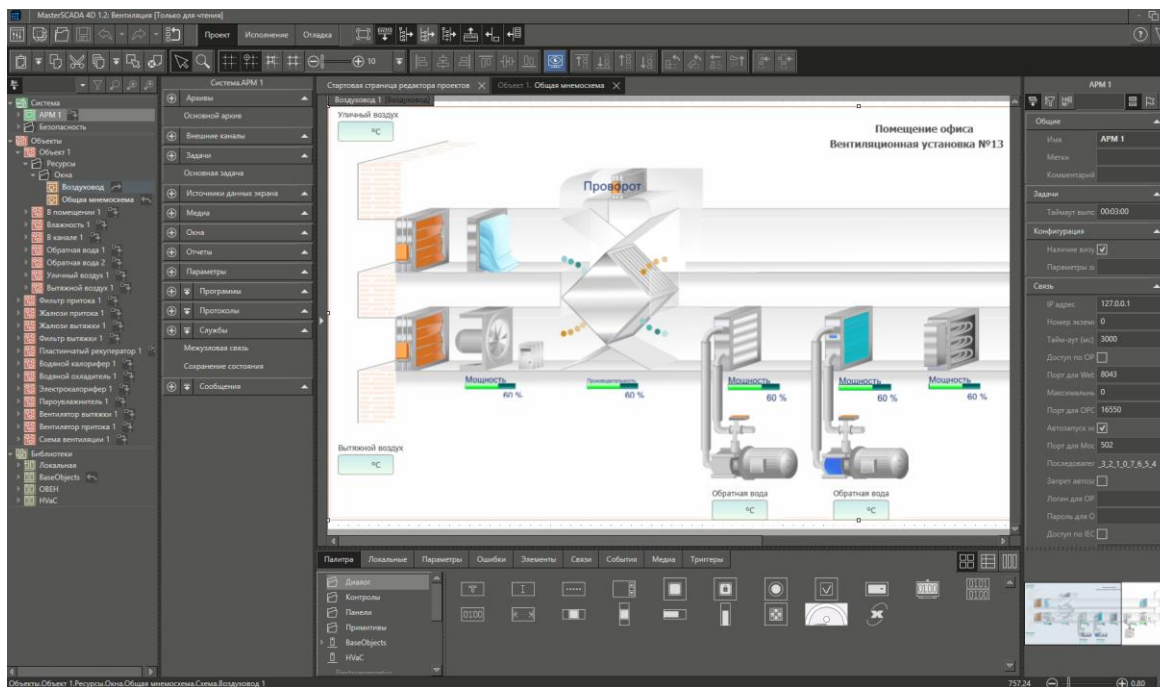
- инструментальная система/среда разработки;
- исполнительная система, состоящая из набора модулей;
- клиент визуализации.

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.

## 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 – современный одноядерный;
- ОЗУ – 4 Гб;
- дисплей – 1280x1024, цвет – 32 бита;
- свободное место на диске – 1 Гб (без учета свободного места для проектов пользователя и при условии, что ранее уже установлен Microsoft.NET Framework);
- последовательный порт, мышь.

Рекомендуемые системные требования:

- ОС – Windows 10 или старше;
- процессор – современный многоядерный;
- ОЗУ – не менее 8 Гб;
- дисплей – 1920x1080, цветность – 32 бита;
- свободное место на диске – 1 Гб (без учета свободного места для проектов пользователя и при условии, что ранее уже установлен Microsoft.NET Framework);
- последовательный порт, мышь.
- Для работы **MasterSCADA 4D** требуются **Microsoft.NET Framework** (версия 4.5) и **OPC Core Components Redistributable**.

**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: <b>Modbus</b> , <b>DCON</b> и др.
<b>OPC UA</b> клиент/сервер	Возможность получения и передачи данных по кроссплатформенному стандарту <b>OPC UA</b>
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 "Ин-САТ", так и другого производителя, например, браузер Chrome 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
<b>OPC UA</b> 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	XP SP3	
Win CE	5.0	Существуют ограничения по использованию клиента визуализации. Особенности работы уточняйте по адресу технической поддержки.
Windows Embedded Standart	XP SP3	
Windows Modern API (WinRT)	8.0	
Linux (в том числе AstraLinux, Debian и др.)	2.6	Как правило, на Linux-подобные системы устанавливается штатная среда исполнения, однако для некоторых сборок операционной системы, возможно, потребуются доработки.
QNX	6.5	
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 "ИнСАТ" может предоставить свою разработку. Данный компонент входит в среду исполнения для данной ОС
Эльбрус		
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
Vtune 6000	Linux
Платформы на базе микропроцессоров Эльбрус 4С и Эльбрус 8С	Эльбрус
Платформы на базе микропроцессоров Baikal T1 и Baikal M	Debian GNU/Linux (возможно также использование QNX - по запросу)
Fastwel МК150	Linux
ОВЕН PLC110M02	Linux
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

## 1.5. Установка MasterSCADA 4D

### ОС Windows

Существуют два инсталлятора продуктов MasterSCADA 4D (© 2008-2017 InSAT):

- *MasterSCADA4D.exe* – мастер установки среды разработки (редактора проекта) 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. 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.

## Другие ОС

Способ установки исполнительной системы зависит от операционной системы устройства.

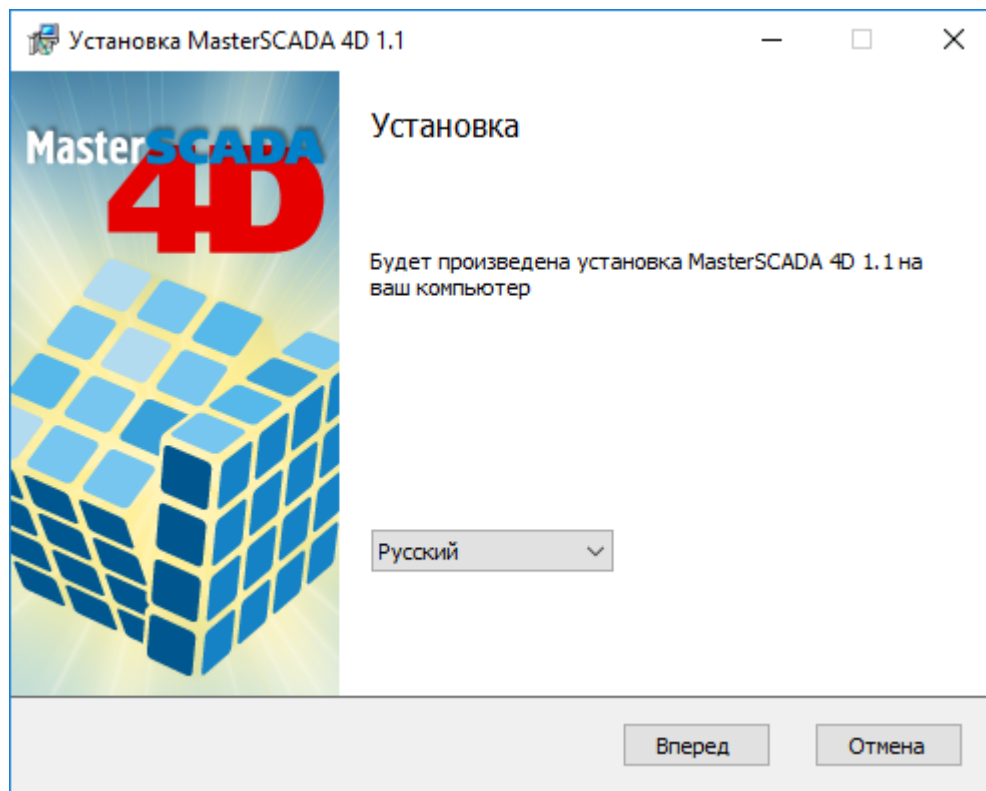
- Linux
- Эльбрус
- QNX
- Win CE
- Windows Embedded

## 1.6. Установка среды разработки MasterSCADA 4D

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

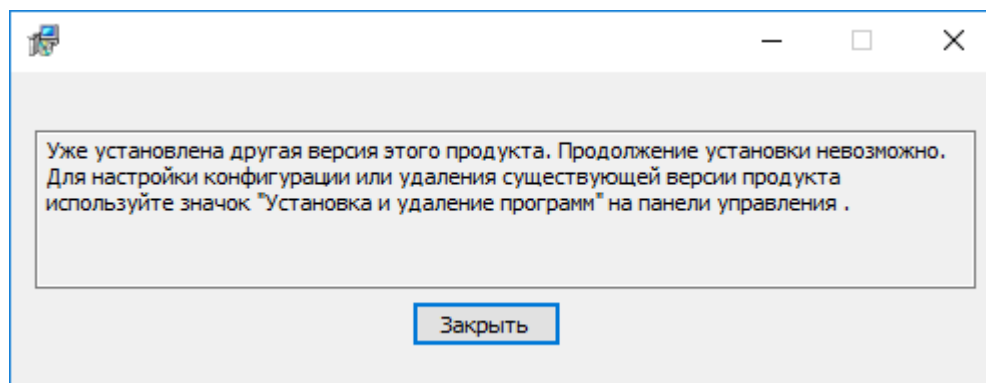


Экран-заставка отображается в течение нескольких секунд, после чего он автоматически исчезает, и открывается начальный диалог мастера установки:

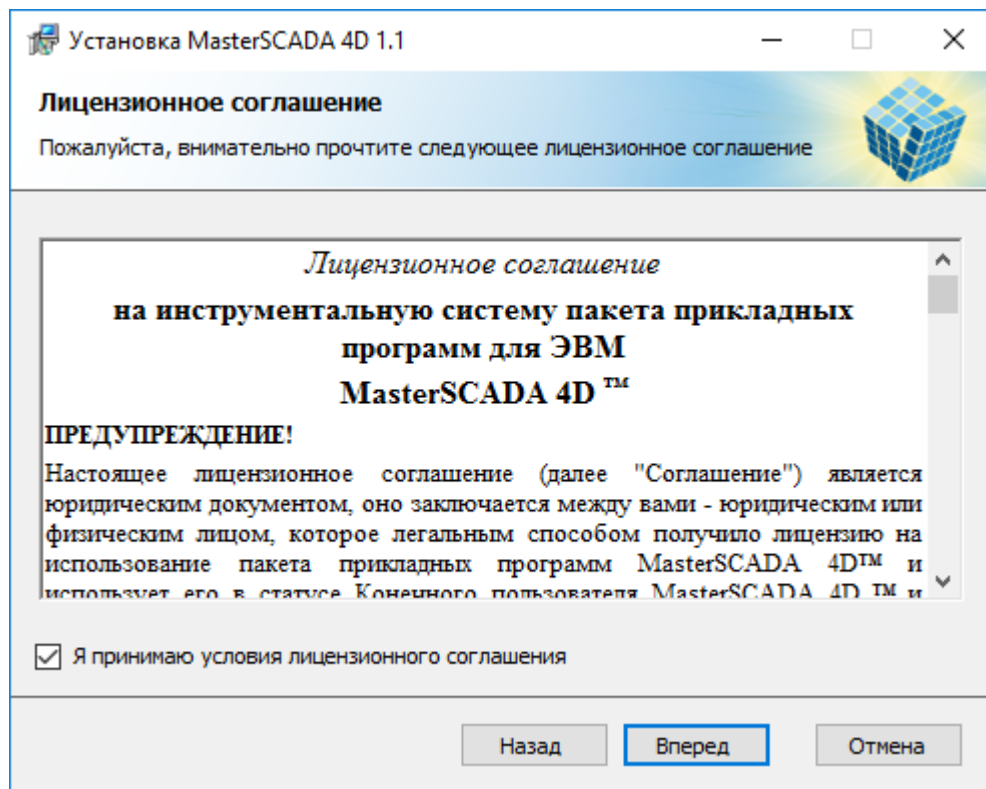


В диалоге выбирается локализация продукта (поддерживаются русский и английский языки).

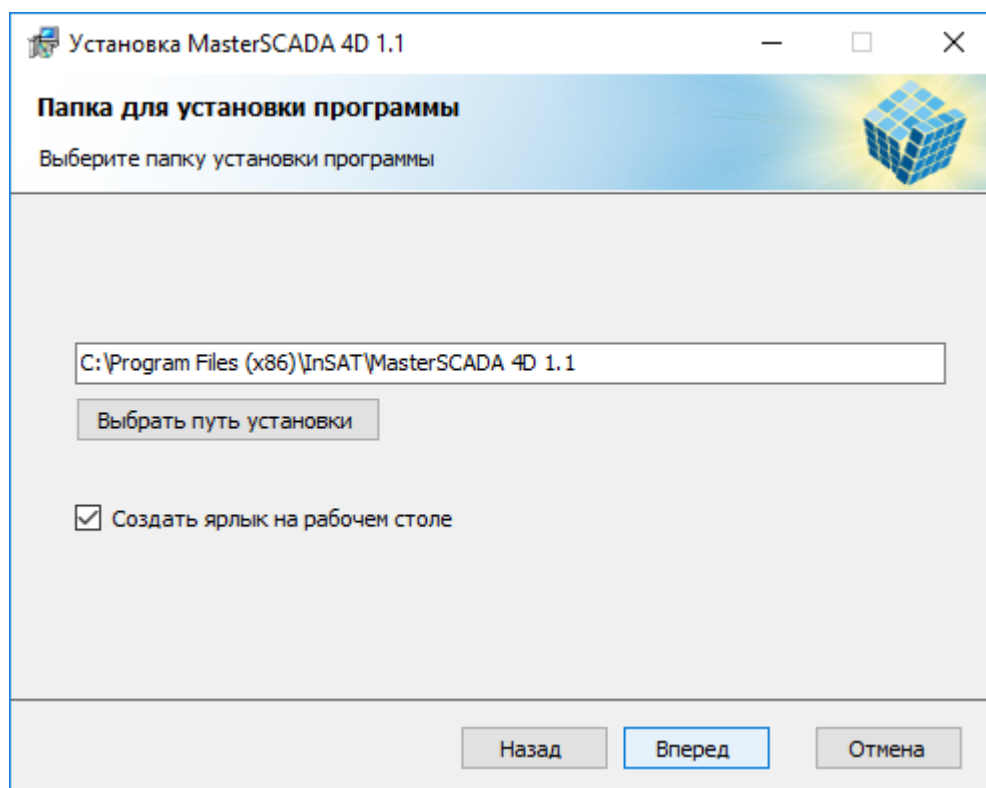
Если среда разработки **MasterSCADA 4D** уже была установлена ранее, то появится сообщение о невозможности продолжения установки:



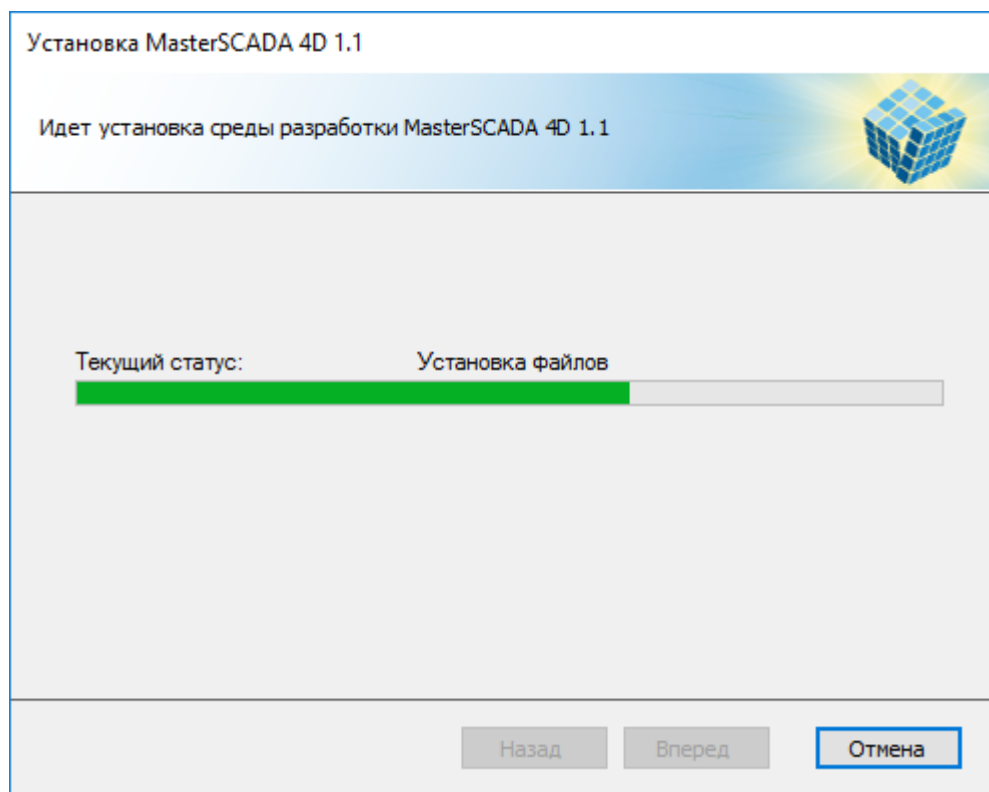
По команде Вперед мастера открывается диалог лицензионного соглашения:



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

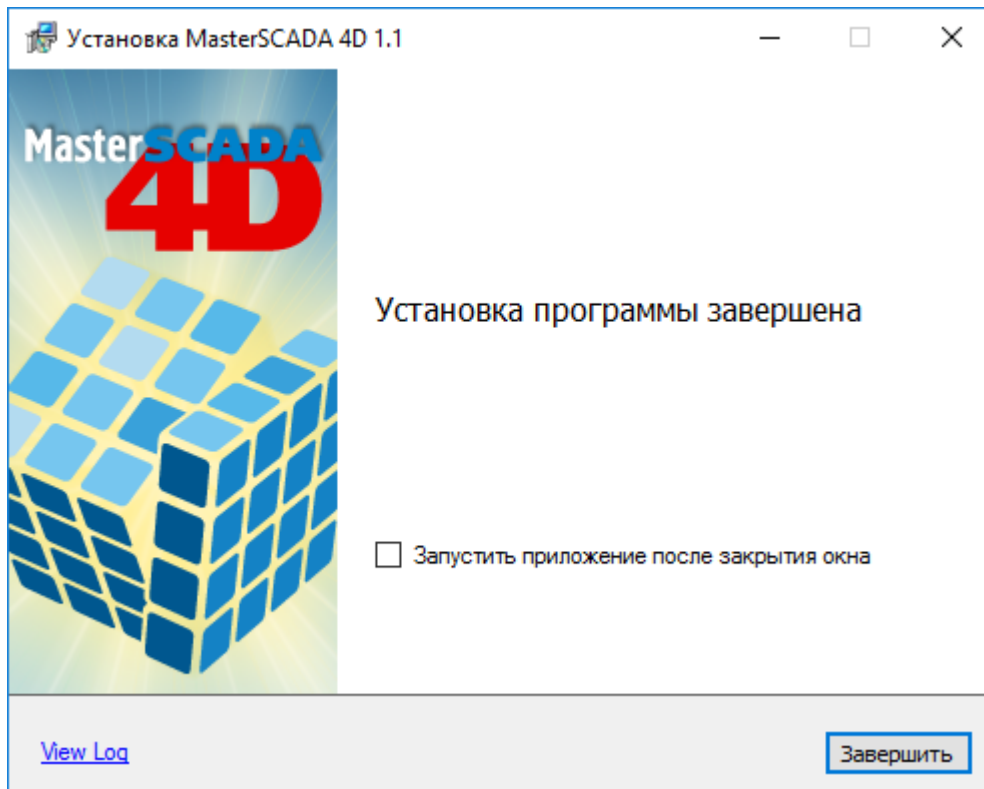


В этом диалоге можно изменить папку установки продукта, воспользовавшись кнопкой **Выбрать путь установки**, а также задать создание ярлыка программы на рабочем столе, отметив пункт **Создать ярлык на рабочем столе**. Нажмите кнопку **Вперед**, и откроется диалог, отображающий процесс установки:



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**.

### 1.6.1. Описание ярлыков MasterSCADA 4D

По завершении установки мастер создает программную группу **Пуск - Все программы - InSAT - MasterSCADA 4D**, содержащую следующие пункты меню:

Имени	Дата изменения	Тип	Размера
Doc	23.03.2018 12:26	Папка с файлами	
MasterSCADA 4D 1.2	23.03.2018 12:26	Ярлык	3 КБ
MasterSCADA 4D Client	23.03.2018 12:26	Ярлык	2 КБ
MasterSCADA 4D RT	23.03.2018 12:26	Ярлык	3 КБ
Остановить сервер	23.03.2018 12:26	Ярлык	2 КБ
Что нового	23.03.2018 12:26	Ярлык	2 КБ

**Doc** – 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);

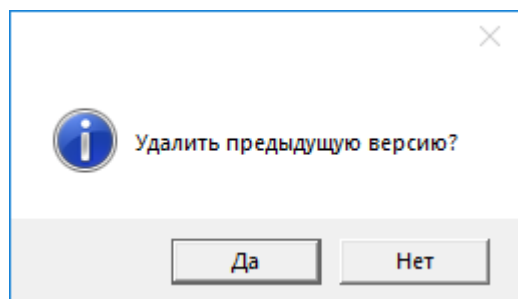
**MasterSCADA 4D RT** – запускает исполнительную систему (серверы *mplc.exe* и *node\_ms4d.exe*, веб-сервер *nginx.exe* и др.) из папки установки продукта (команда <папка установки MasterSCADA 4D>\bin\Config\MasterPLC\WIN32\run\_hmi.bat). Рабочая папка, в которой хранятся конфигурация проекта, сайт и протоколы работы сервера,  
– %<профиль> \AppData\Roaming\InSAT\MasterSCADA4D1.1\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.

### 1.6.1.1. Обновление версии на ОС Windows

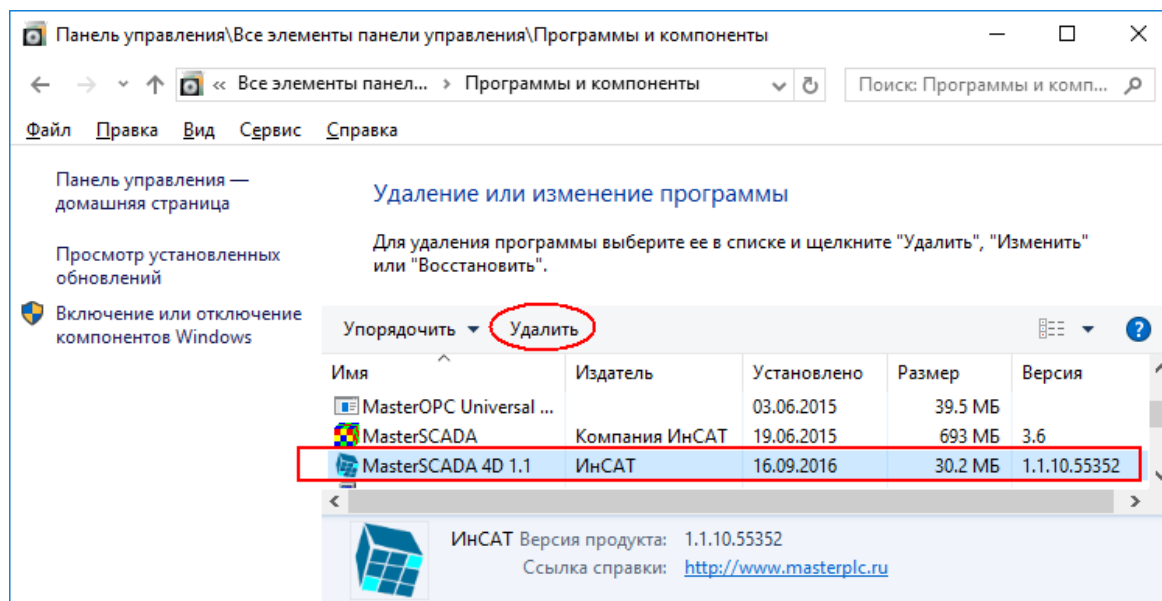
Для обновления версии среды разработки **MasterSCADA 4D** необходимо запустить инсталлятор *MasterSCADA4D.exe*, а для обновления версии исполнительной системы **MasterSCADA 4D RT** необходимо запустить инсталлятор *MasterSCADA4DRT.exe*. Так как на компьютере уже установлена предыдущая версия продукта,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



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

### 1.6.1.2. Удаление MasterSCADA 4D

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



### 1.6.1.3. Запуск среды разработки

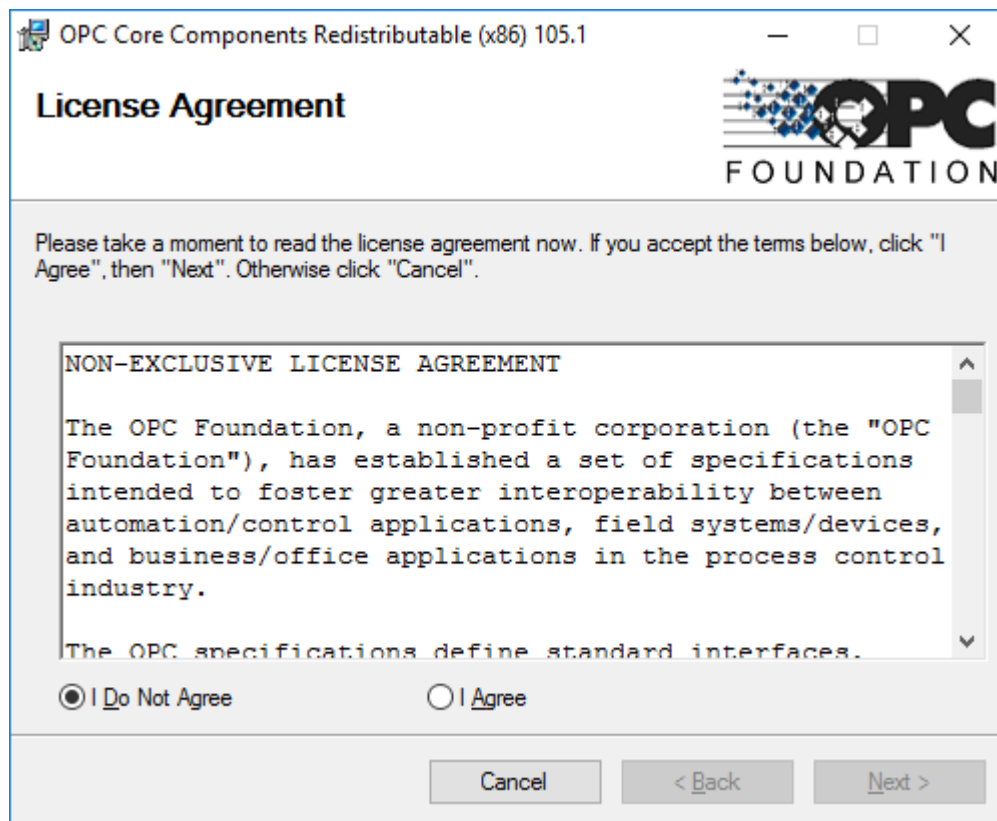
Для запуска среды разработки (редактора проекта) следует выбрать в меню **Пуск** пункт, соответствующий версии **MasterSCADA 4D**, либо дважды нажать левой кнопкой мыши на ярлык на рабочем столе **MasterSCADA 4D [номер версии]**, либо открыть файл `<папка установки продукта>\bin\ProjectEditor.exe` в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

При старте редактора проекта в директории `%профиль пользователя%\AppData\Roaming\InSAT\MasterSCADA4D\` создается папка **sys\_log**, содержащая протоколы работы (файлы с именами `yy_mm_dd HH_MM_SS.log`).

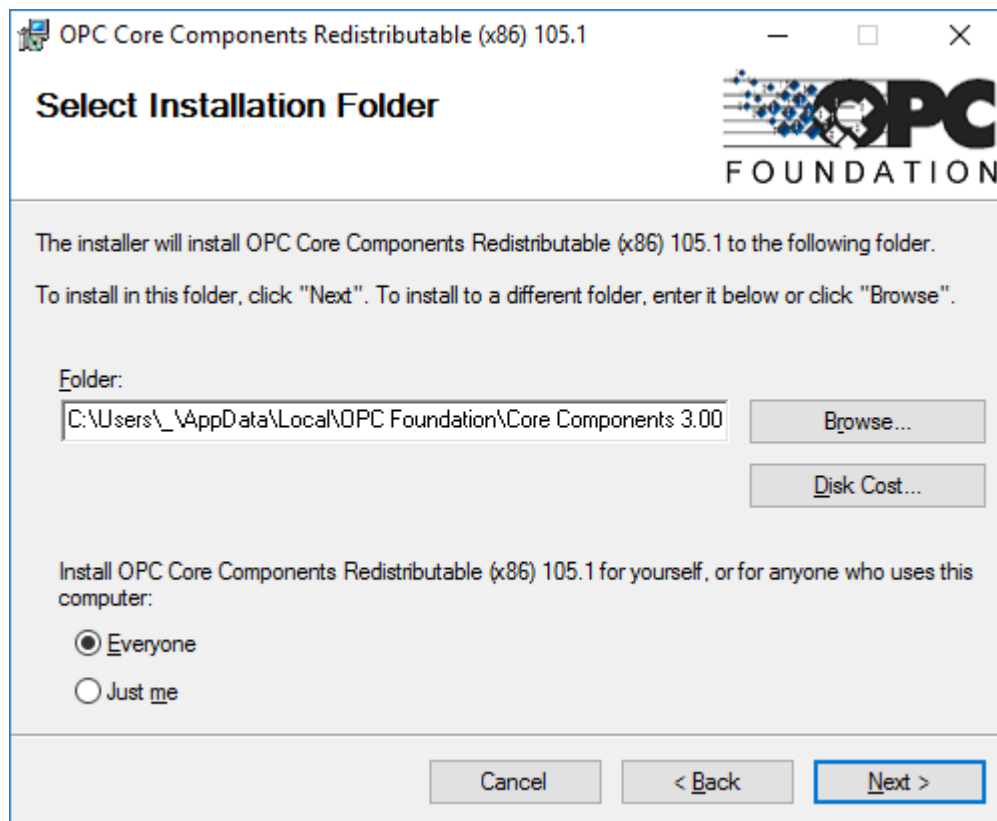
### 1.6.1.4. Установка дополнительных компонентов

#### 1.6.1.4.1. Установка OPC Core Components Redistributable

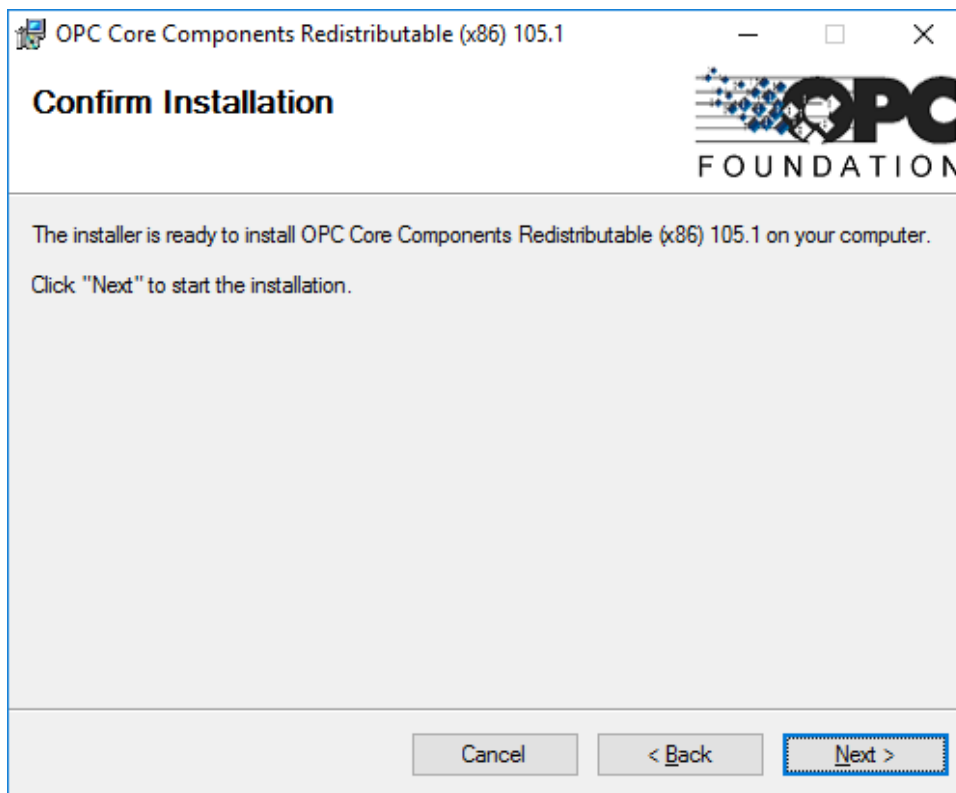
Мастер установки продукта **MasterSCADA 4D** запускает процедуру установки **OPC Core Components Redistributable** автоматически, если этот продукт еще не установлен на ПК.



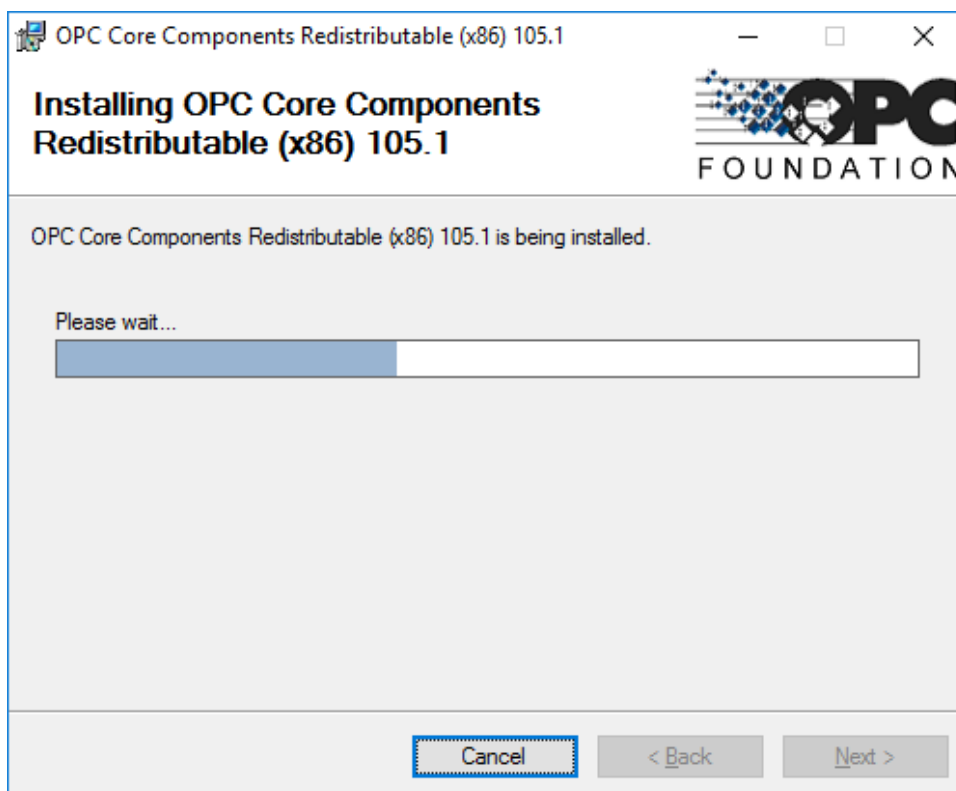
В первом диалоге мастера установки **OPC Core Components Redistributable** – диалоге лицензионного соглашения – необходимо выбрать пункт **I Agree** и нажать **Next** – откроется следующий диалог мастера:



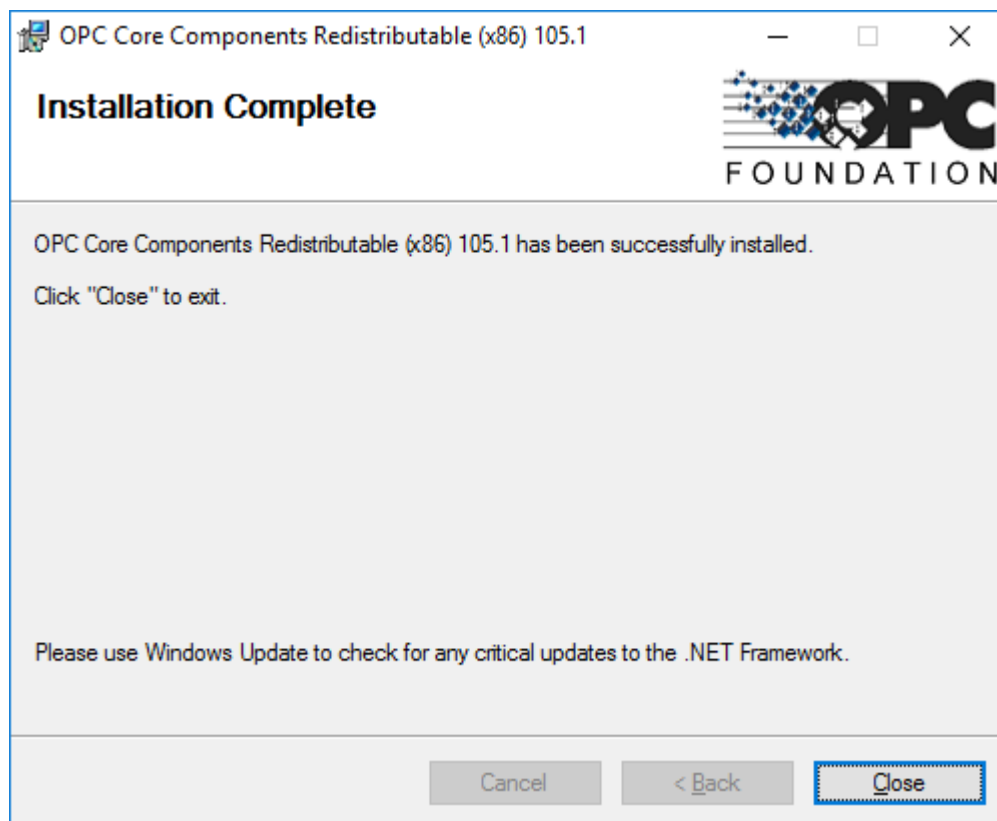
В этом диалоге можно изменить папку установки продукта (кнопка **Browse**) и задать вид установки: только для себя (**Just me**) или для всех (**Everyone**). Нажмите кнопку **Next** – откроется диалог подтверждения установки:



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



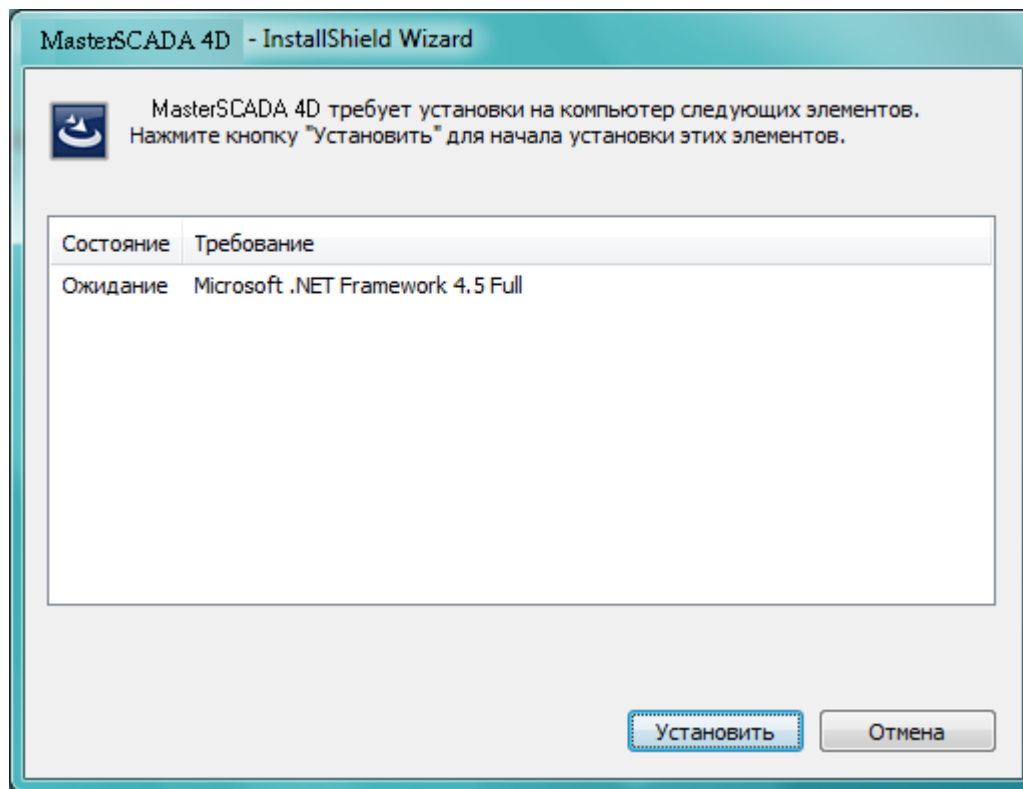
По окончании установки отобразится заключительный диалог мастера:



Для завершения работы мастера требуется нажать **Close**.

#### 1.6.1.4.2. Установка Microsoft.NET Framework

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

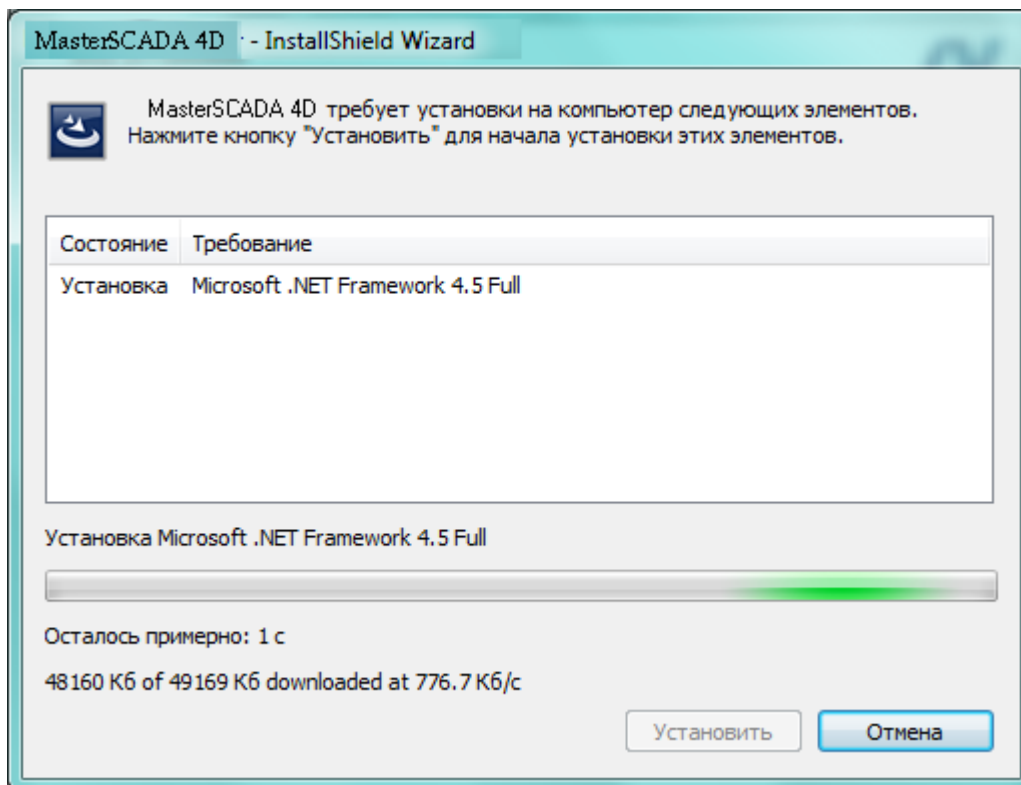
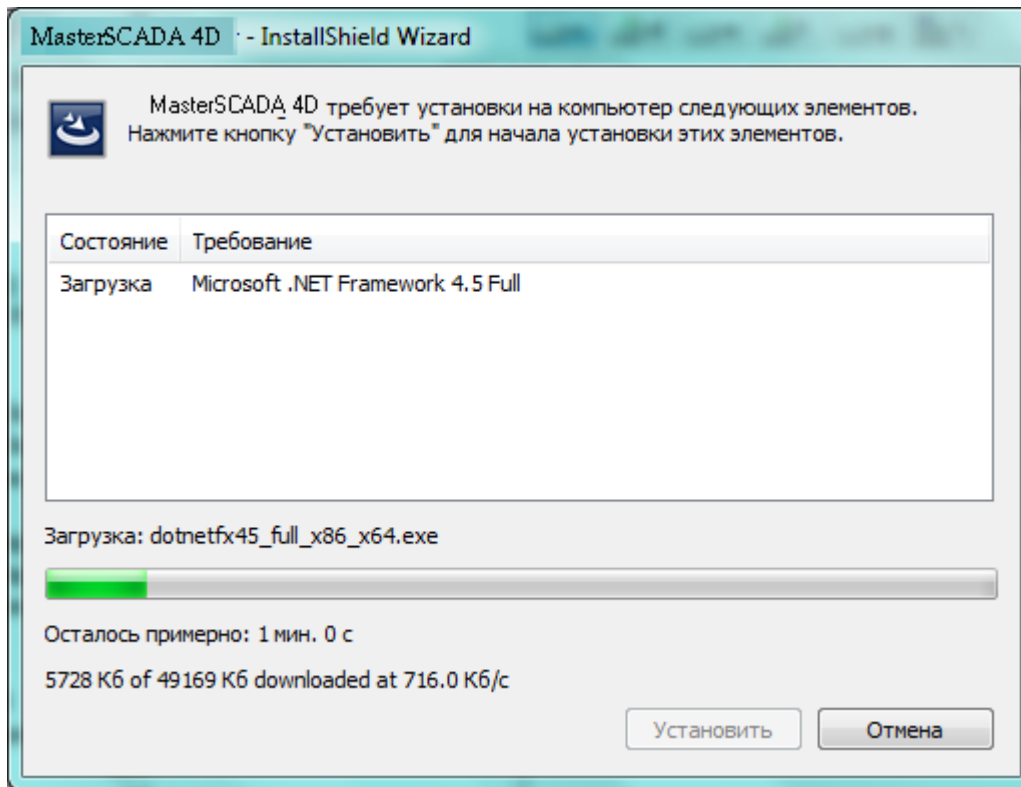


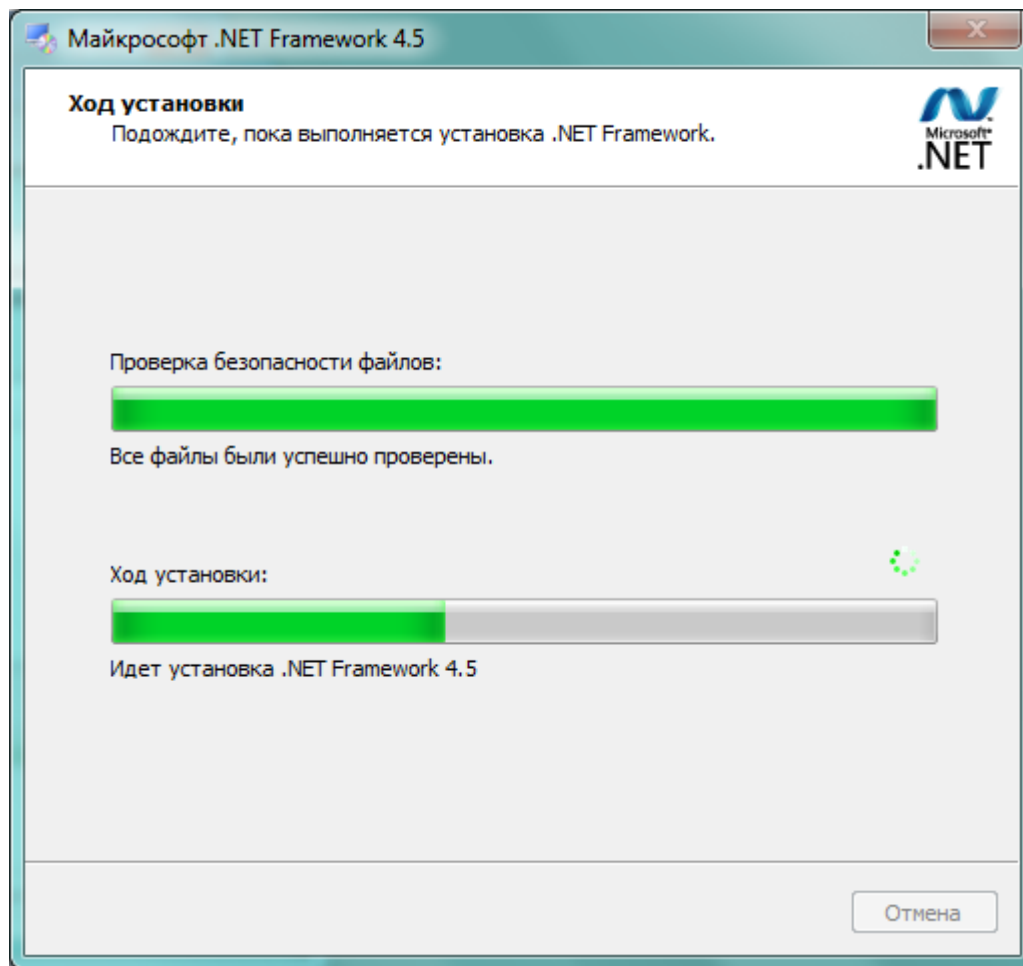
Инструменты диалога:

- **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.

## 1.6.2. Установка среды исполнения MasterSCADA 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**
- **Win CE**
- **Windows Embedded**

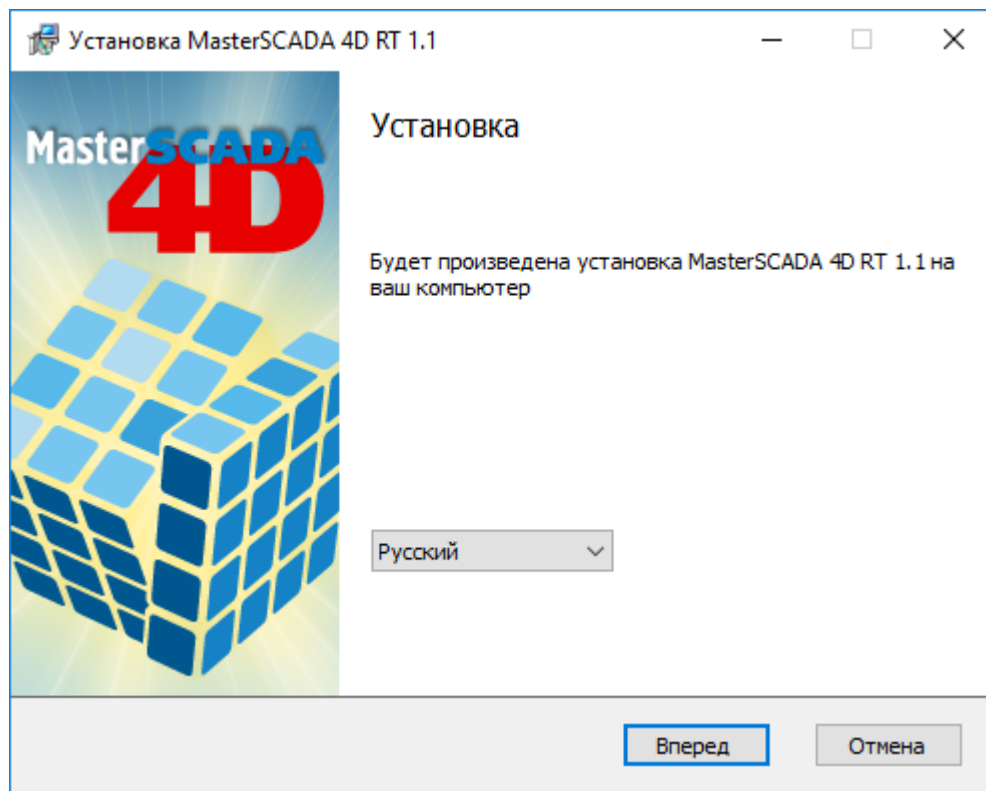
### 1.6.2.1. Установка на ОС Windows

### 1.6.2.1.1. Установка исполнительной системы MasterSCADA 4D RT

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

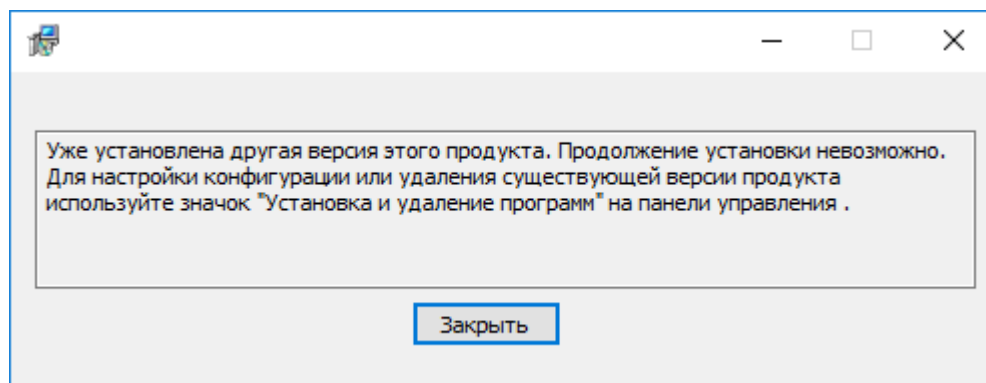


Экран-заставка отображается в течение нескольких секунд, после чего автоматически исчезает, и открывается начальный диалог мастера установки:

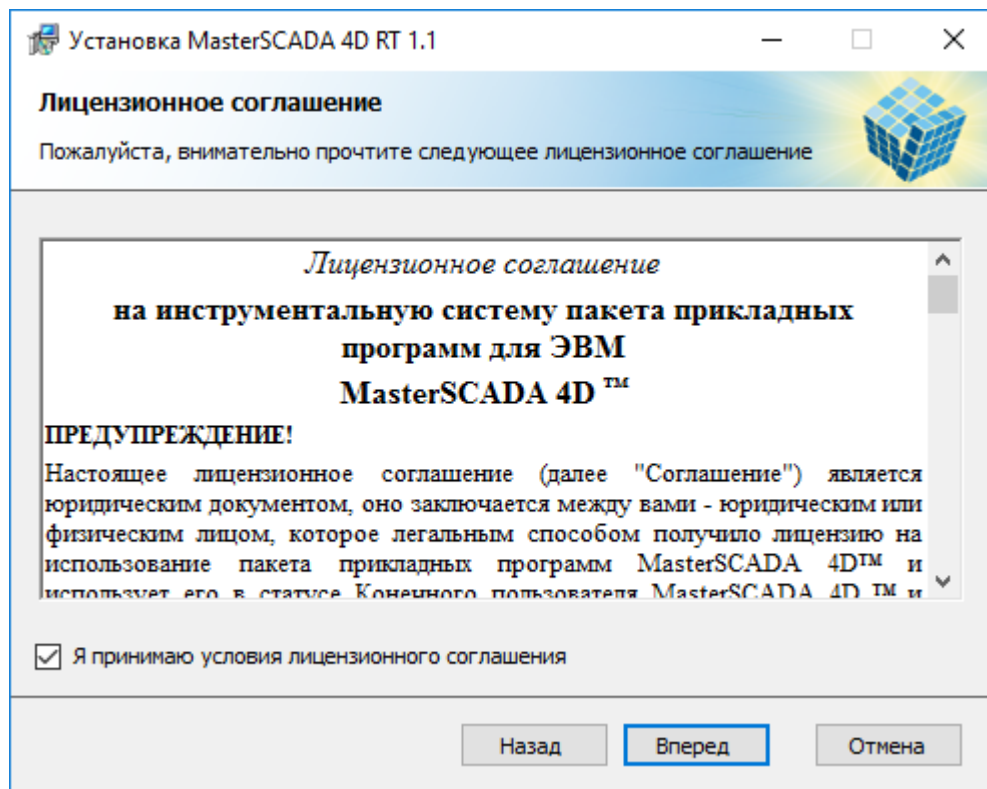


В диалоге выбирается локализация продукта (поддерживаются русский и английский языки).

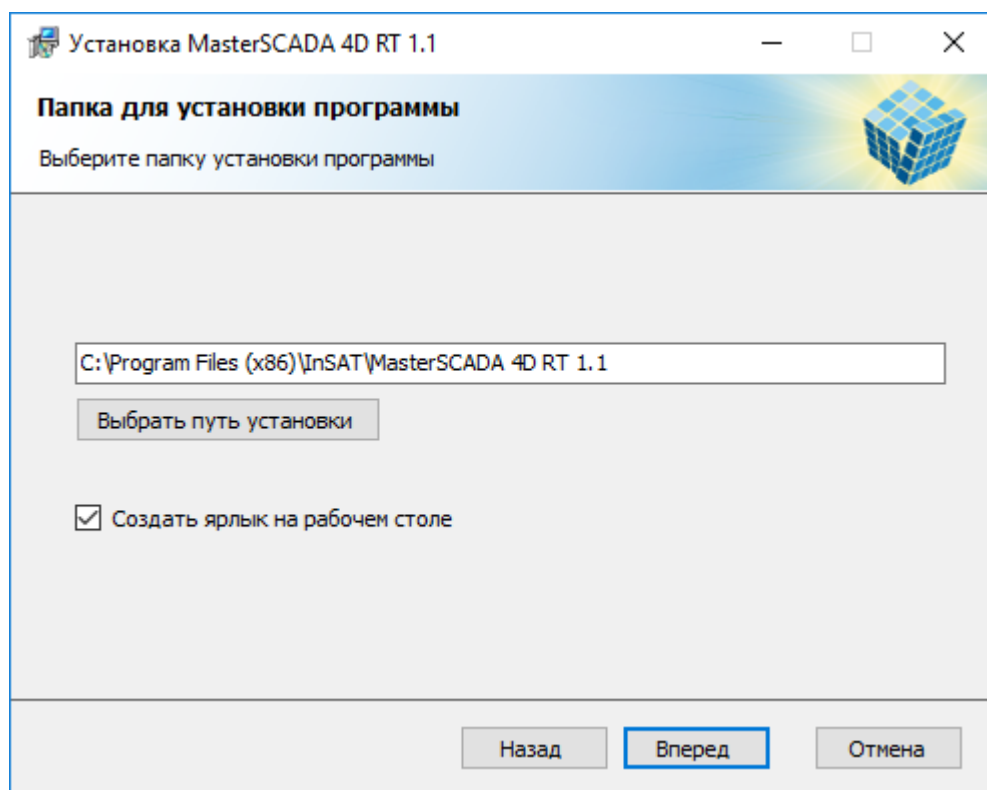
Если исполнительная система **MasterSCADA 4D RT** уже была установлена ранее, то появится сообщение о невозможности продолжения установки:



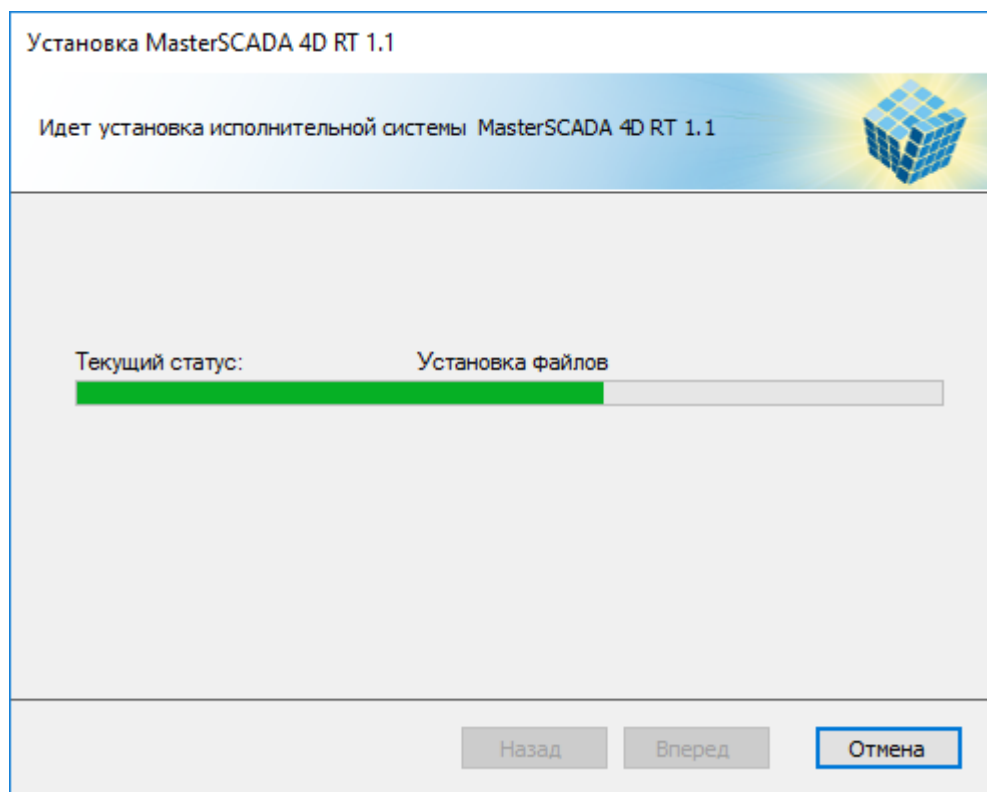
По команде **Вперед** мастера открывается диалог лицензионного соглашения:



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



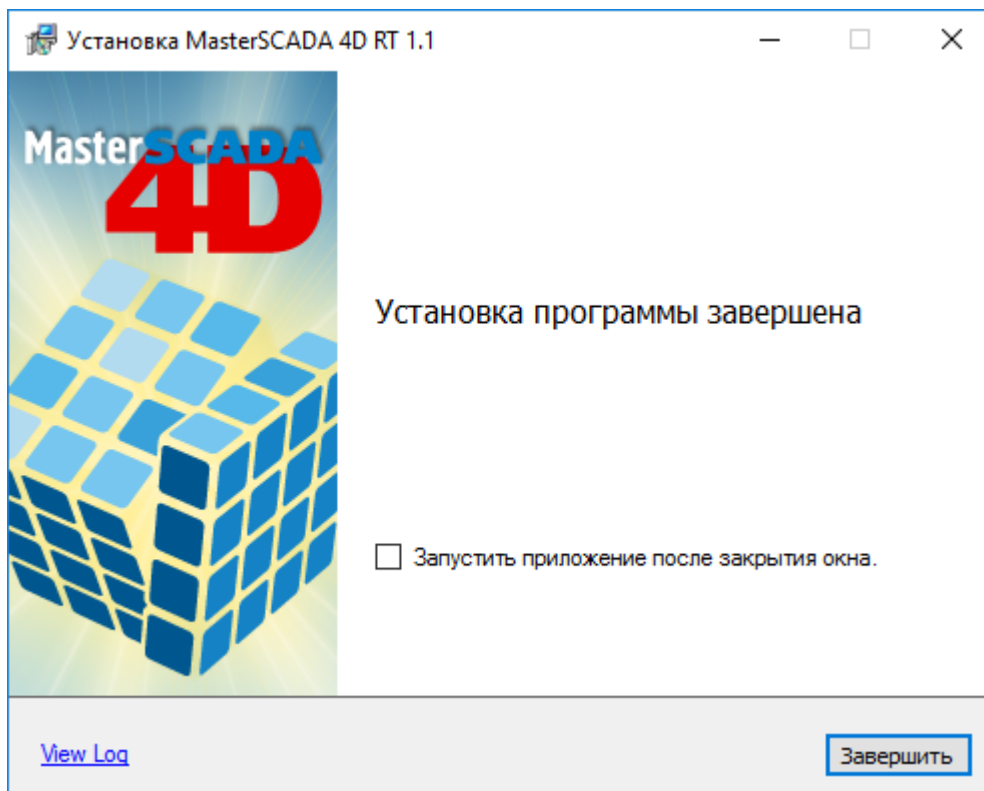
В этом диалоге можно изменить папку установки продукта, воспользовавшись кнопкой **Выбрать путь установки**, а также задать создание ярлыка программы на рабочем столе, отметив пункт **Создать ярлык на рабочем столе**. Нажмите кнопку **Вперед** – откроется диалог, отображающий процесс установки:



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

**Важно! Установка драйвера ключа защиты произойдет только в случае установки коммерческой версии программного продукта.**

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**.

### 1.6.2.1.2. Описание ярлыков MasterSCADA 4D RT

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

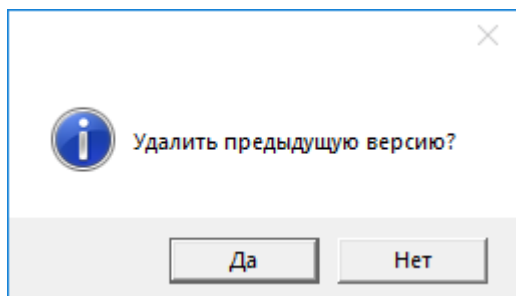
Имени	Дата изменения	Тип	Размера
MasterSCADA 4D Client	20.04.2018 9:55	Ярлык	2 КБ
MasterSCADA 4D RT	20.04.2018 9:55	Ярлык	3 КБ

- **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\InSAT\MasterSCADA4DRT1.1\Server\`;

### 1.6.2.1.3. 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:

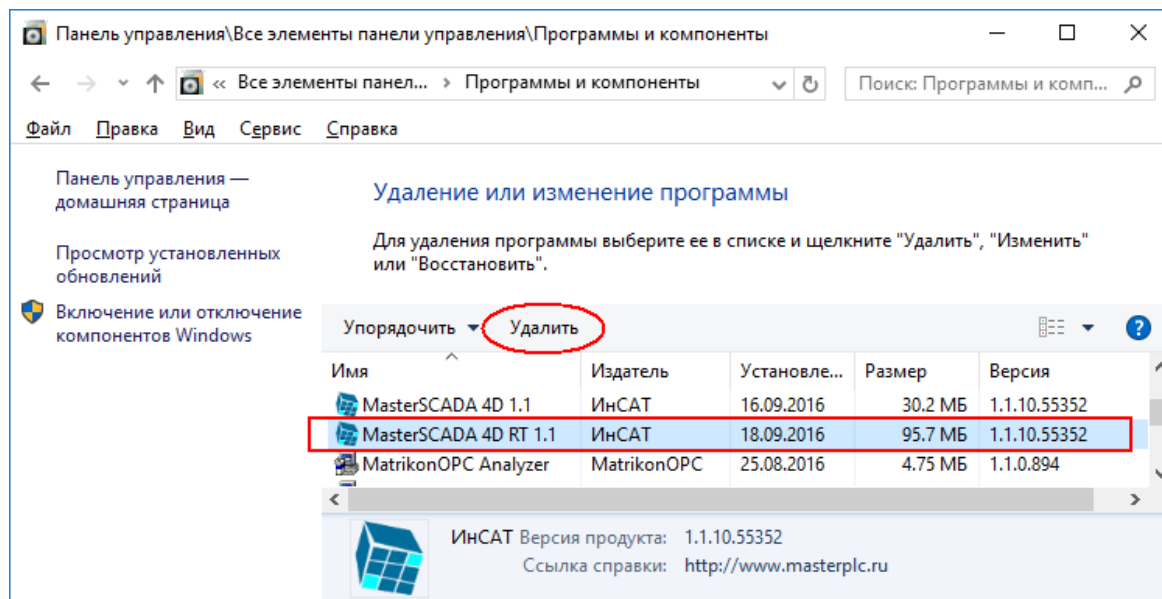


Если нажать кнопку **Да**, то ранее установленная версия удалится, и запустится установка новой.

### 1.6.2.1.4. Удаление MasterSCADA 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**:





### 1.6.2.1.5. Установка драйвера ключа защиты

Коммерческие версии исполнительной системы **MasterSCADA 4D RT** для ОС Windows защищаются USB-ключом. Для его работы, помимо USB-порта в компьютере, необходима установка драйвера.

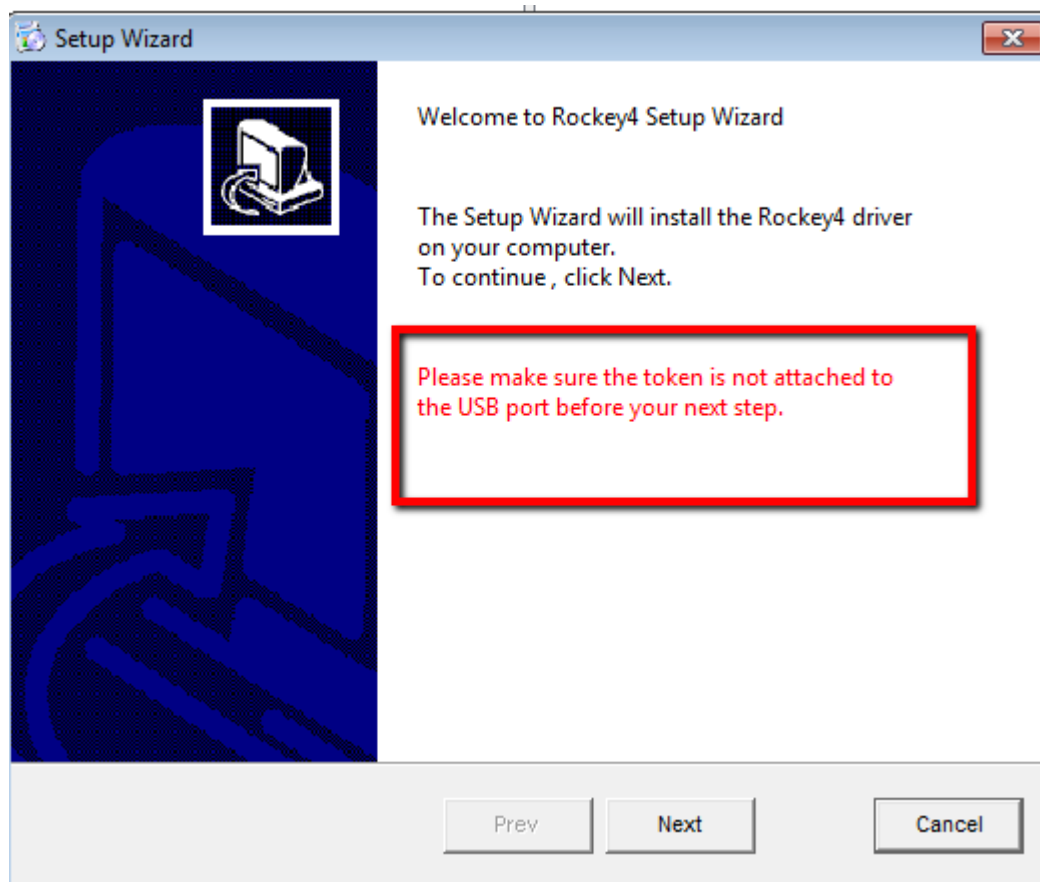
По умолчанию драйвер ключа устанавливается автоматически, нужно следовать командам установщика программ.

В случае если драйвер не установился автоматически, его нужно установить вручную.

Дистрибутив поставляется на диске вместе со средой исполнения.

В зависимости от типа операционной системы нужно запустить файл *rockey\_setup.exe* для 32-битной, либо *rockey\_setup\_x64.exe* — для 64-битной.

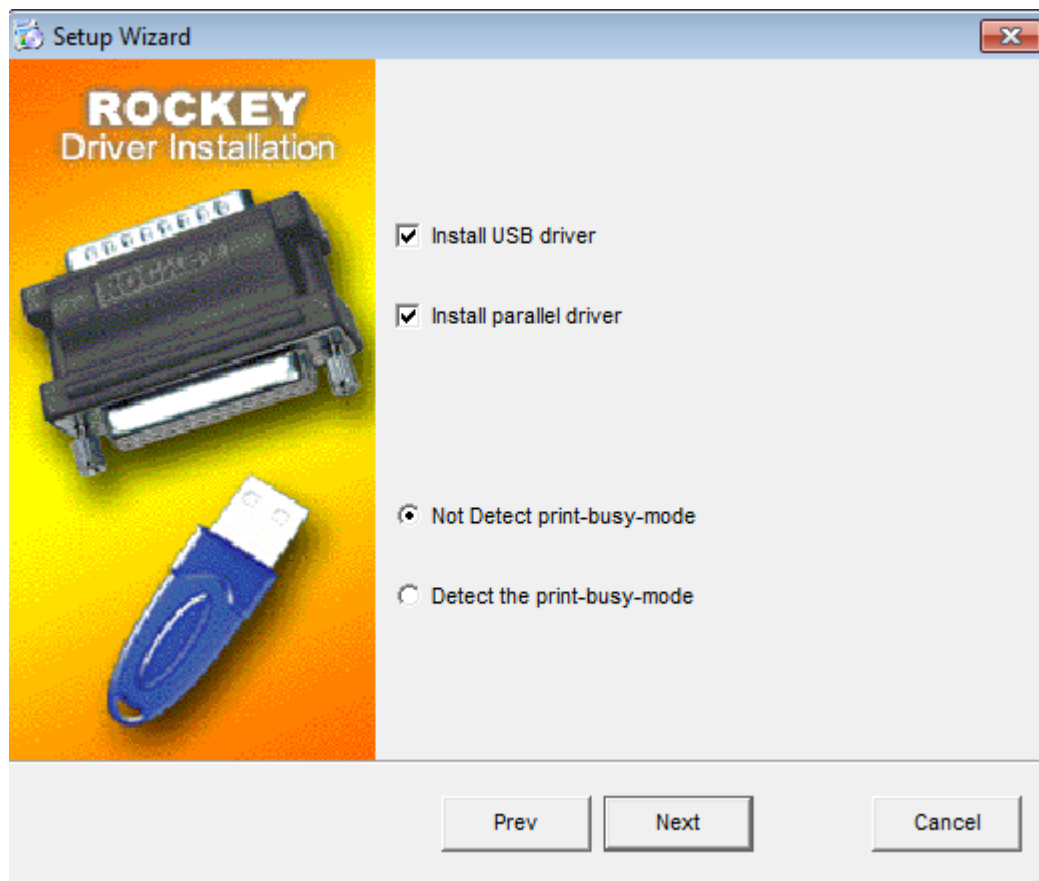
Откроется диалоговое окно:



**Важно! Перед тем как перейти к следующему шагу убедитесь, что ключ извлечен из USB-порта.**

Нажмите кнопку **Next**

Далее следуйте рекомендациям мастера:



Установка завершена.

USB-ключ имеет индикатор, который поможет диагностировать его работу после установки в порт:

- **Индикатор мигает** — драйвер ключа не установлен или установлен некорректно.
- **Индикатор горит постоянно** — ключ готов к работе.

### 1.6.2.2. 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**

#### 1.6.2.2.1. 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 "ИнСАТ". Доступ предоставляется по запросу. На FTP-сервере технической поддержки 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**:

**mpic** — основной процесс исполнительной системы;

**mpic\_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:

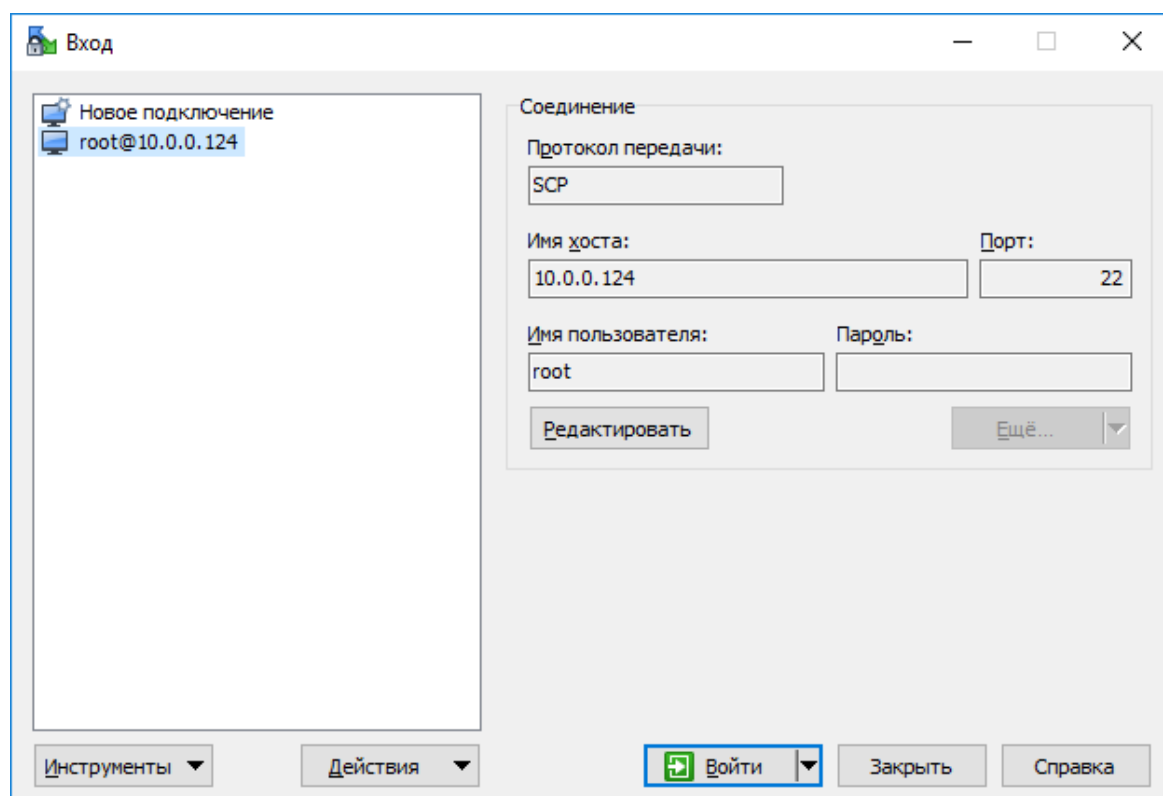
**PuTTY**

**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

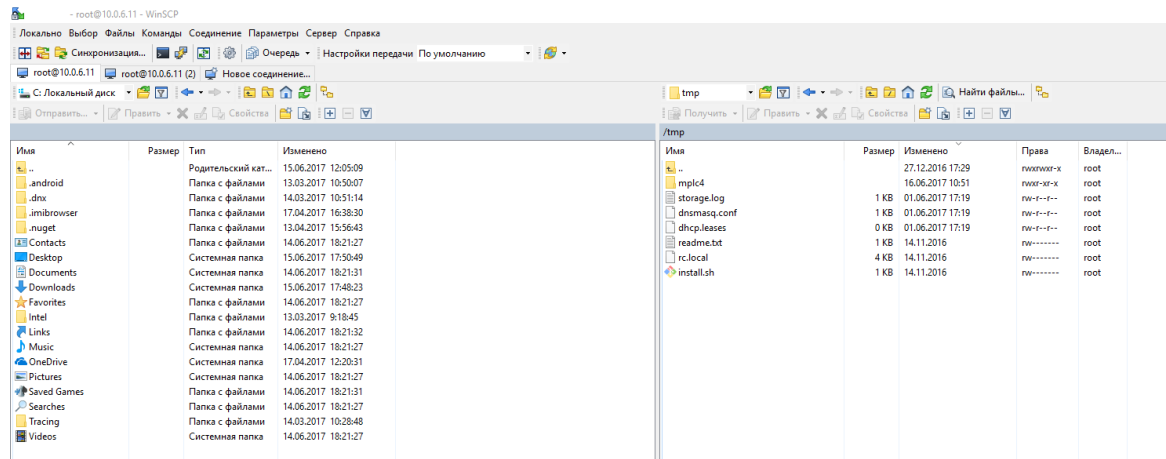


2. Подключить патчкорд к контроллеру и свитчу/компьютеру.

3. Включить питание устройства и нажать кнопку **Войти**.

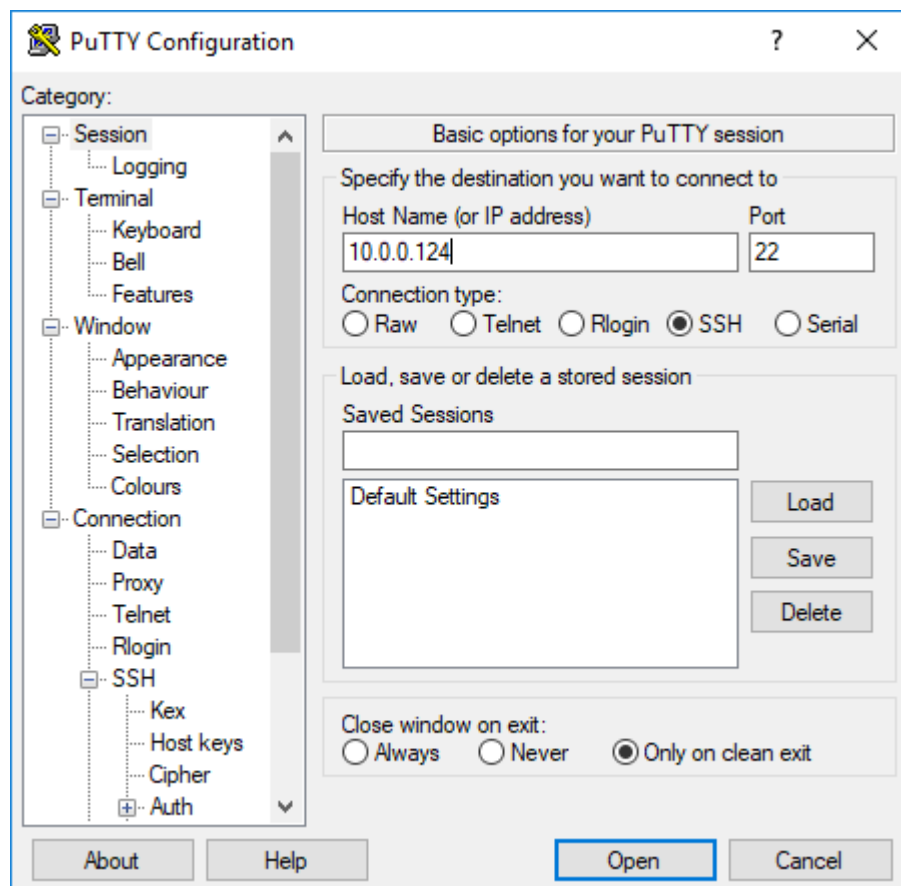
4. При входе необходимо будет указать пароль. Если контроллер подключается впервые, то в появившемся окне нажмите кнопку **Да**. Если появится окно с ошибкой, то необходимо нажать **ОК**.

### 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=&lt;path&gt;</code>	Путь, по которому требуется установить RT (по умолчанию: <code>/opt/mplc4</code> )
<code>--platform=&lt;NAME&gt;</code>	Задаёт тип платформы (Доступные значения PLC110, REGUL)
<code>--http-port=&lt;port&gt;</code>	Изменяет номер порта веб-сервера (по умолчанию: 80)
<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 <code>nginx.tar.gz</code> .
<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 <code>/nowdt</code>
<code>--config-only</code>	Устанавливает опцию обновления только конфигураций без обновления файлов <code>mplc</code> и <code>nginx</code> . С этой опцией можно запустить инсталлятор без <code>mplc.tar.gz</code> и <code>nginx.tar.gz</code>
<code>--start-delay=&lt;seconds&gt;</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 произойдёт.

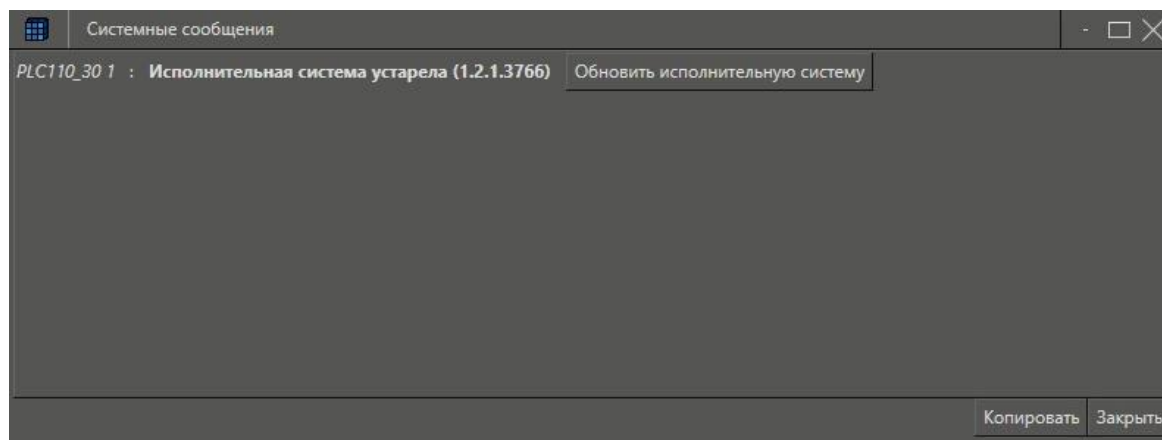
N - количество запускаемых сервисов mplc

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

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

### 1.6.2.2.2. 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**.





### 1.6.2.2.3. 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**.

### 1.6.2.3. 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**.

### 1.6.2.4. Установка на ОС WinCE

В **MasterSCADA 4D** поддерживается контроллер **WinPac** с предустановленной операционной системой WinCE. В этом разделе дано описание установки именно на этот контроллер, однако порядок действий для других устройств с такой же операционной системой будет похожим.

#### Загрузка с использованием USB Flash накопителя

Для подготовки узла к работе потребуются подключенные к контроллеру: монитор с разъемом VGA D-SUB, USB-мышь и клавиатура (желательно), а также ПК с выходом в Интернет.

Перечень шагов:

1. Записать на USB Flash накопитель файлы первоначальной установки (поставляются отдельно **по запросу** в техническую поддержку).
2. Дополнительно из папки установки **MasterSCADA 4D** из дочерней директории с названием контроллера (по умолчанию `c:\Program Files (x86)\InSAT\MasterSCADA 4D 1.1\bin\Config\MasterPLC\PLC\WinPac`) переписать все файлы на USB Flash накопитель.
3. Подключить USB Flash накопитель к контроллеру:

- 3.1. Создать папку: *\Micro\_SD\mplc4*.
- 3.2. Скопировать файлы из USB Flash накопителя в *\Micro\_SD\Mplc4*.

#### 4. В WinPAC Utility:

- 4.1. Перейти на закладку **Auto-Execution**, в поле **Programs1** ввести путь к файлу исполнительной системы (если сделано по инструкции выше, то: *\Micro\_SD\Mplc4\mplc\_service.exe*), который был загружен ранее. Нажать кнопку **Set**.
- 4.2. Перейти на закладку **Network Settings**, во вкладке **FTP Settings** выставить все переключатели в положение **Enable** и в поле **Change FTP default download directory \Temp to:** ввести **\**. Нажать кнопку **Set**.
- 4.3. Перейти во вкладку **Users and Passwords**, ввести в поле **User name** **sa**, в поле **Password** — **1234**. Далее нажать кнопку **Add**. И затем нажать кнопку **Set**.
- 4.4. Запустить **Explorer**, перейти в папку *\Micro\_SD\Mplc4* и запустить BAT-файл *ISAPI\_Registrator.bat* (либо скопировать файл *InSATHMI.dll* из папки *\Micro\_SD\Mplc4* в корень системного диска *\System\_Disk* и через утилиту **Run** в пуске выполнить команду *\Micro\_SD\mplc4\ISAPI.Registrator.exe -f\System\_Disk\InSATHMI.dll*). Закрыть окно консоли после отработки скрипта.
- 4.5. В окне программы **Winpac Utility** в меню выбрать **File-Save and reboot**. Перезагрузить контроллер вручную, если этого не произошло автоматически.

### Загрузка через FTP

Для подготовки узла к работе потребуются подключенные к контроллеру: монитор с разъемом VGA D-SUB, USB-мышь и клавиатура (желательно), а также ПК с выходом в Интернет, Ethernet (TCP/IP) подключение между компьютером и контроллером.

Перечень шагов:

1. Запустить контроллер и настроить его IP-адрес, маску подсети (настройка аналогична обычной **Windows XP**).
2. В контроллере из меню **Programs** запустить **WinPAC Utility**:
  - 2.1. Перейти на закладку **Network Settings**, во вкладке **FTP Settings** выставить все переключатели в положение **Enable** и в поле **Change FTP default download directory \Temp to:** прописать **\**. Нажать кнопку **Set**.
  - 2.2. В меню выбрать **File-Save and reboot**.
3. После перезагрузки контроллера подключиться FTP-клиентом (например, можно использовать **Total Commander**).

4. Загрузить в контроллер файлы первоначальной установки (поставляются отдельно), а также все файлы, которые находятся в дочерней директории с названием контроллера папки установки **MasterSCADA 4D** (по умолчанию `c:\Program Files (x86)\InSAT\MasterSCADA 4D 1.1\bin\Config\MasterPLC\PLC\WinPac`).
5. Переместить файлы системы в `\Micro_SD\Mplc4`.
6. Опять запустить **WinPAC Utility**:
  - 6.1. Перейти на закладку **Auto-Execution**, в поле **Programs1** указать место файла исполнительной системы (если сделано по инструкции выше, то расположен он тут: `\Micro_SD\Mplc4\mplc_service.exe`), который был загружен ранее. Нажать кнопку **Set**.
  - 6.2. Перейти во вкладку **Users and Passwords**, ввести в поле **User name** **sa**, в поле **Password** **-1234**. Далее нажать кнопку **Add**. И затем нажать кнопку **Set**.
  - 6.3. Запустить **Explorer**, перейти в папку `\Micro_SD\Mplc4` и запустить BAT-файл `ISAPI_Registrator.bat` (либо скопировать файл `InSATHMI.dll` из папки `\Micro_SD\Mplc4` в корень системного диска `\System_Disk` и через утилиту **Run** в пуске выполнить команду `\Micro_SD\mplc4\ISAPI.Registrator.exe -f\System_Disk\InSATHMI.dll`). Закрыть окно консоли после отработки скрипта.
  - 6.4. В окне программы **Winpac Utility** в меню выбрать **File-Save and reboot**. Перезагрузить контроллер вручную, если это не произошло автоматически.

По вопросам установки обращайтесь по **адресу технической поддержки**.

### 1.6.2.5. Установка на ОС QNX

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

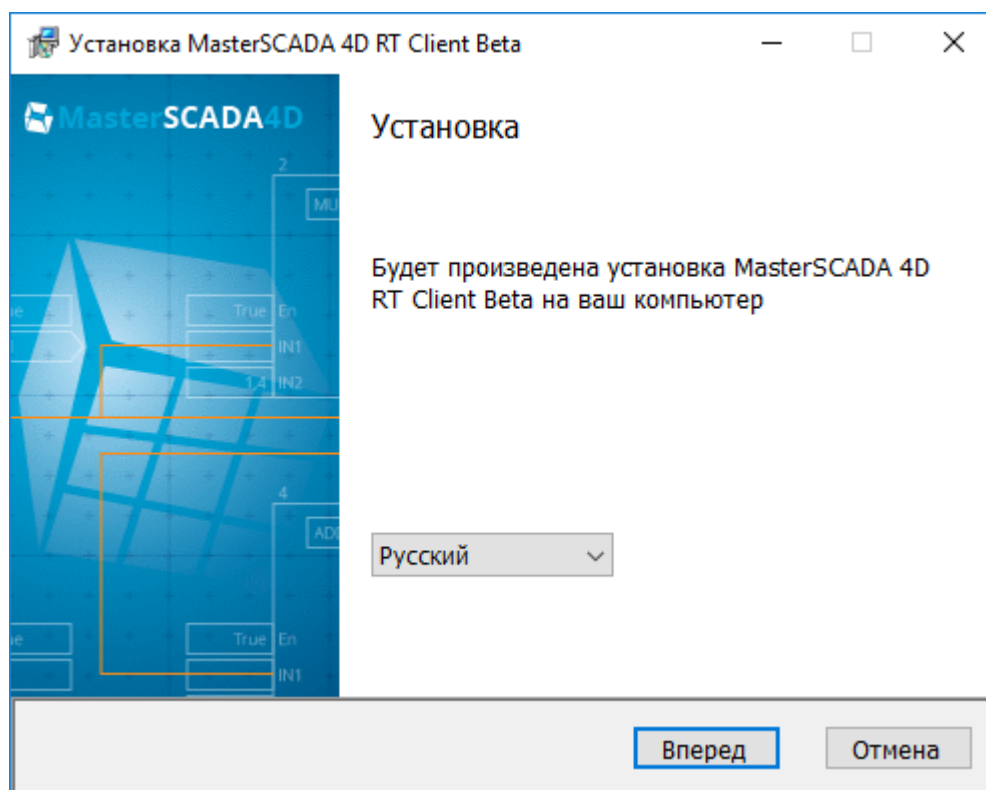
Однако, существует отличие для контроллеров фирмы ТРЕИ с предустановленной операционной системой QNX, у которых для подключения к файловой системе используется протокол **FTP**, а не **SSH**. Для загрузки файлов в память устройства можно использовать различные файловые менеджеры, например **Far Manager**.

### 1.6.3. Установка и удаление клиента на ОС Windows

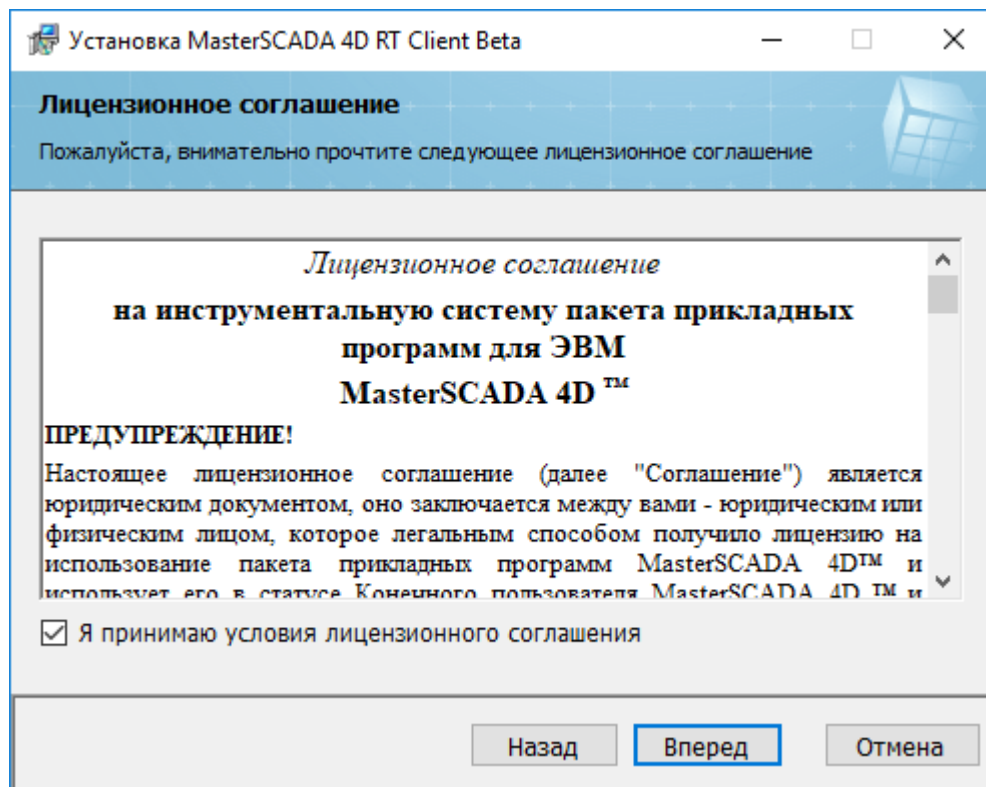
**MasterSCADA 4D Client** работает на ОС Windows и устанавливается в случае, если необходимо подключаться к исполнительной системе, работающей на другом компьютере.

Для того чтобы начать установку, необходимо запустить файл - `MasterSCADA4DRTClient.exe`

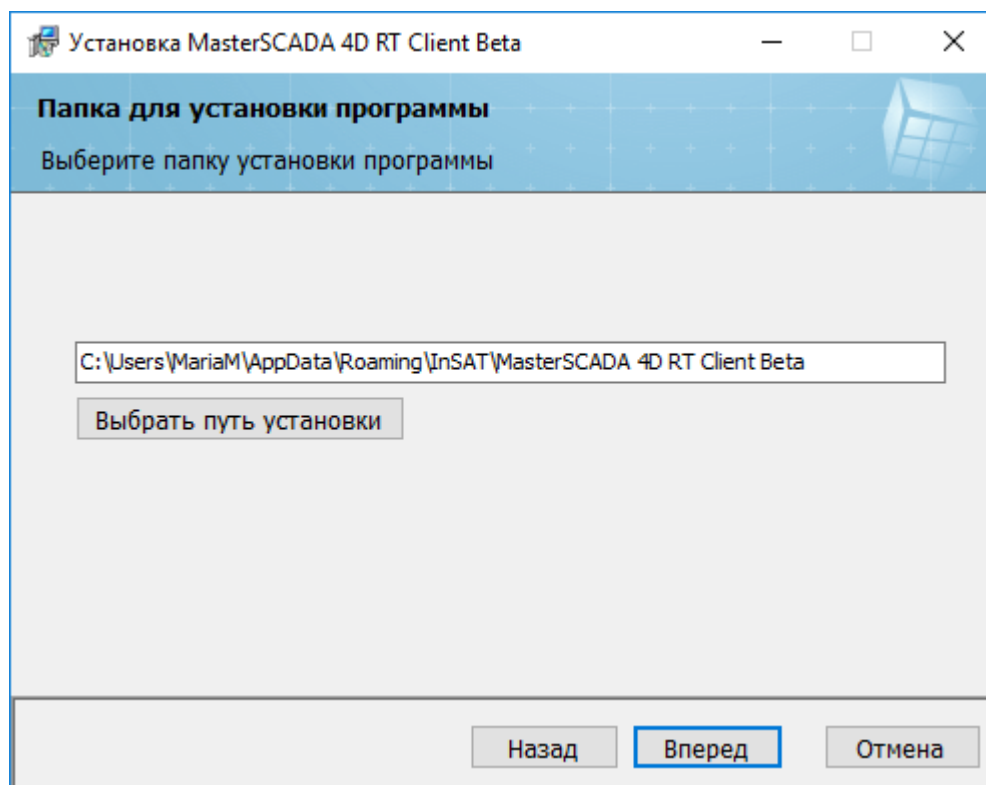
В появившемся диалоговом окне требуется выбрать язык клиента (по умолчанию **Русский**) и нажать кнопку **Вперед**:



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



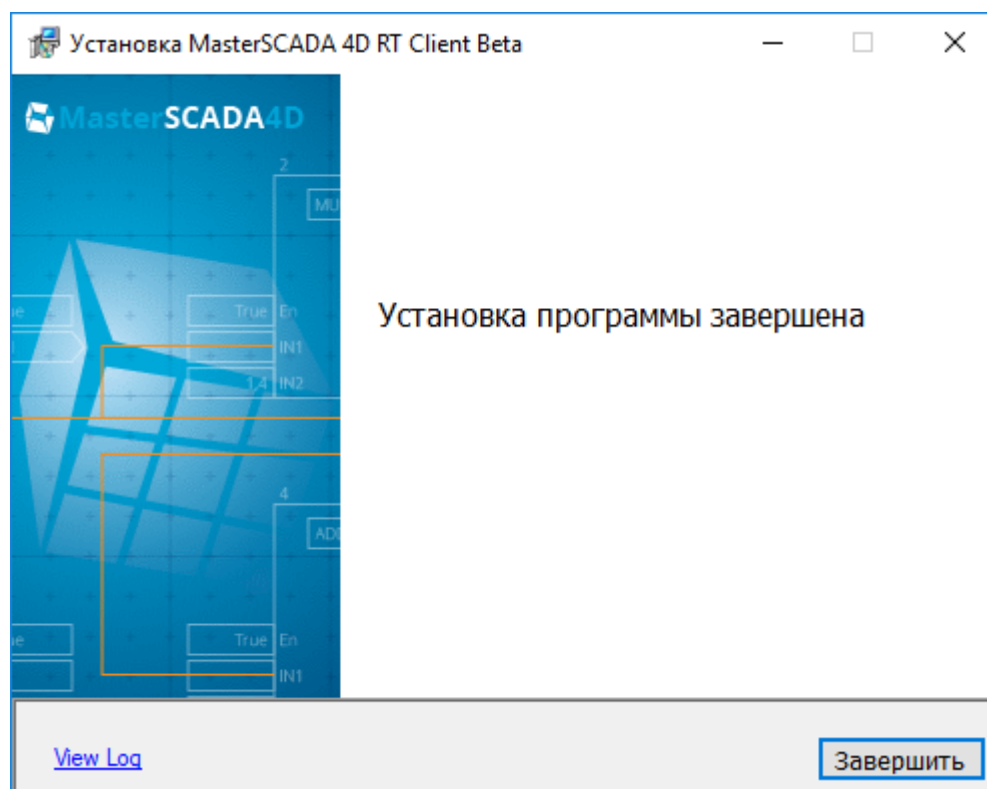
Затем выбирается папка установки приложения:



По умолчанию это папка: *C:\Users\User\AppData\Roaming\InSAT\MasterSCADA 4D RT Client [номер версии]*

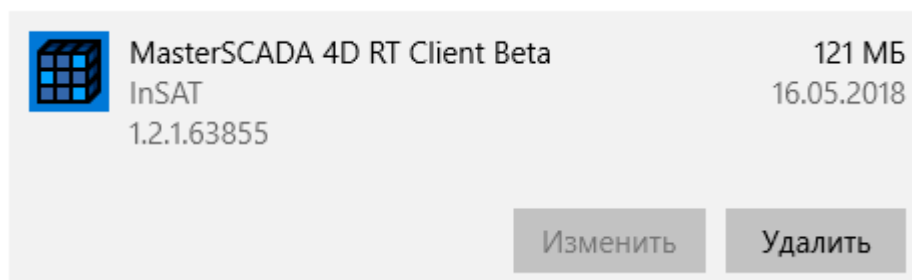
**Важно! Установка приложения в данную папку позволяет использовать приложение пользователям с ограниченными правами**

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**:



## 1.7. Лицензионная защита

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

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

**Активация** применяется для исполнительной системы, работающей на любой операционной системе, кроме Windows. 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 **scada@insat.ru**. В ответ Вы получите файл для активации.

**USB-ключ защиты** используется для ОС Windows и Linux (в особых случаях). Для корректной работы исполнительной системы с ключом защиты необходима **установка драйвера**.

Для выбора версии среды исполнения обратитесь по электронному адресу **scada@insat.ru**.

## 1.8. 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\MasterSCADA4D1.1\*.
- Назначить права на чтение и редактирование папки, в которой хранятся подготовленные для загрузки в среду исполнения проекты: *c:\Users\[имя пользователя]\AppData\Roaming\InSAT\*.
- 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\InSAT\MasterSCADA4DRT1.1*.
- Разрешить TCP-порт для работы с **веб-server**, по умолчанию это **8043** или тот, который задан в файле *c:\Program Files (x86)\InSAT\MasterSCADA 4D RT 1.1\nginx\conf\nginx-mplc.conf*.
- 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 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 + <номер экземпляра mplc>**



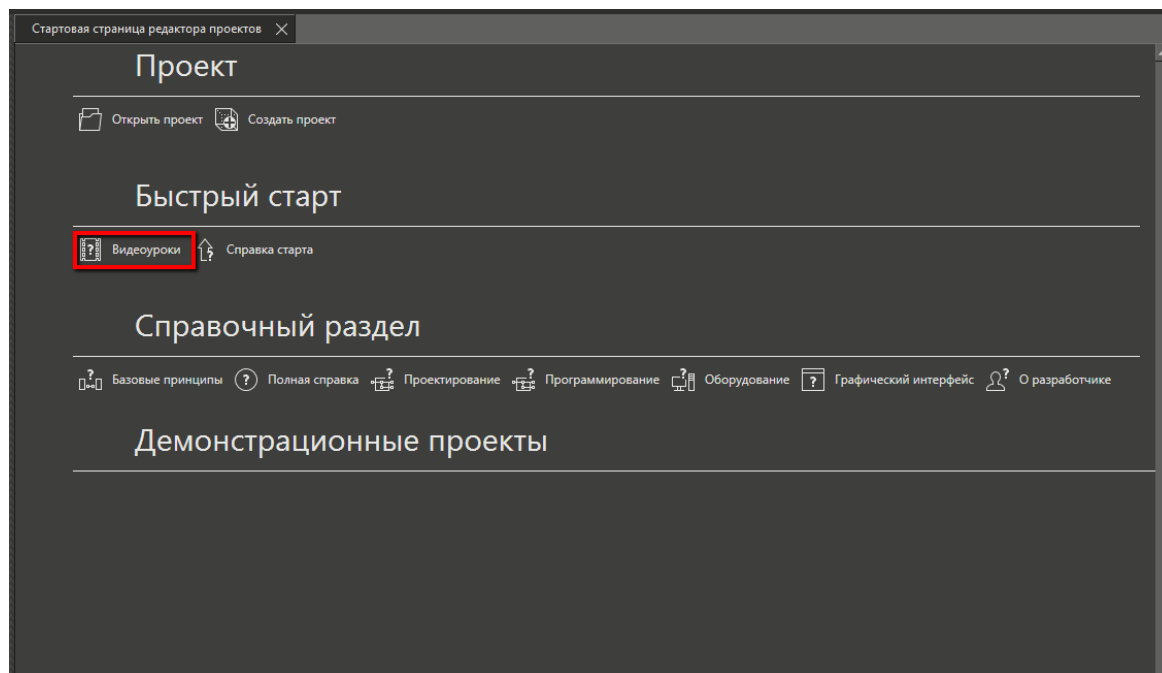
**Для клиента визуализации**

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

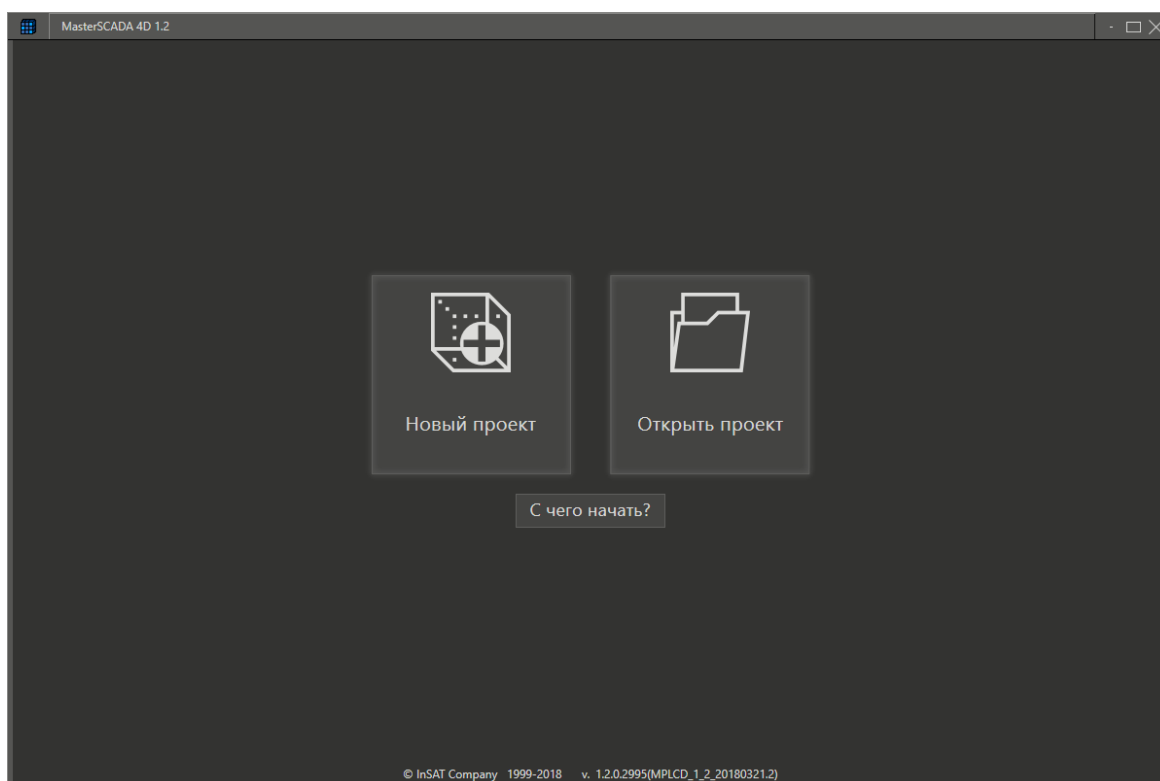
Рассмотрим создание проекта, реализующего управление включением и выключением устройства двумя кнопками – Пуск и Стоп. При нажатии на кнопку Пуск устройство должно включаться. При нажатии на кнопку Стоп устройство должно выключаться.

- 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

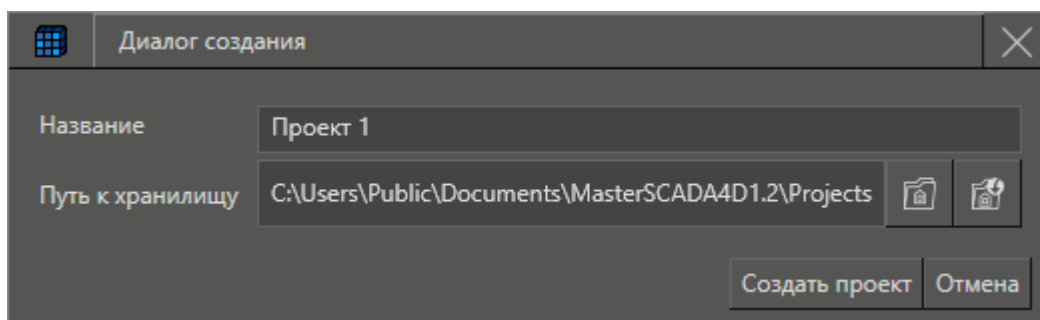
#### 2.1.1.БС. Урок 1. Создание проекта

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

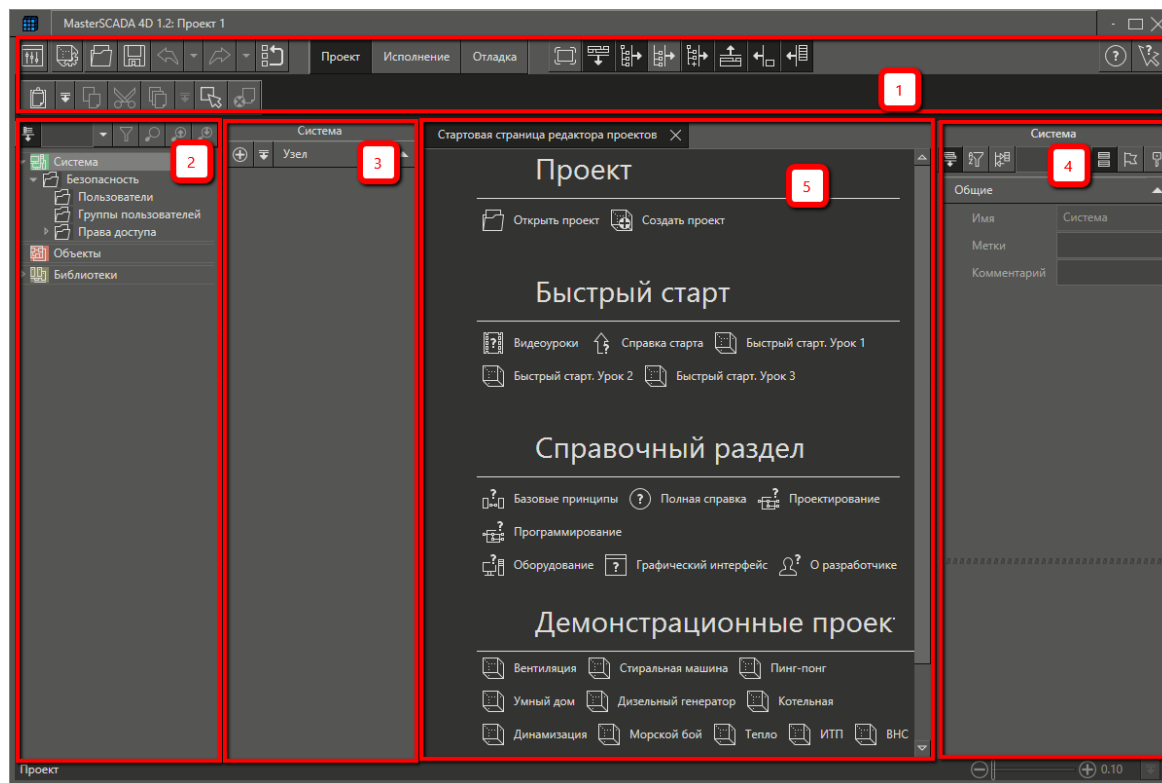
выполнив пункт меню Windows Пуск\Все программы\InSAT\MasterSCADA 4D\MasterSCADA 4D.



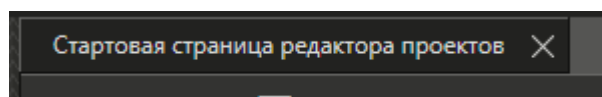
После запуска MasterSCADA 4D нам будет предложено либо создать новый проект, либо открыть существующий. Создадим новый проект.



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

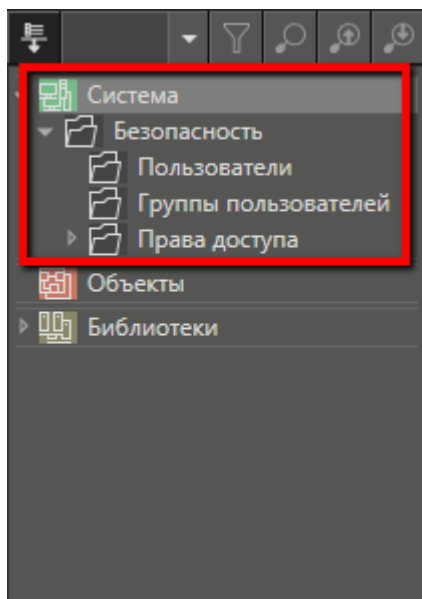


Интерфейс среды разработки состоит из нескольких элементов. В верхней части интерфейса располагается панель инструментов (1). Слева – дерево проекта (2) и контекстная панель (3). В правой части – панель свойств (4). В центральной части интерфейса располагается поле (панель) редакторов (5). В этом поле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

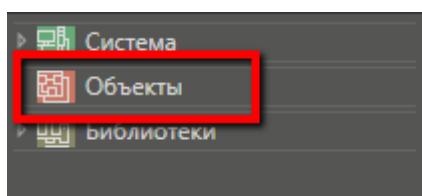


Дерево проекта состоит из трёх частей.

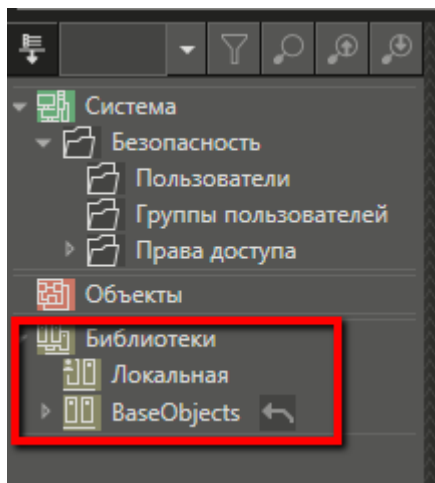
Первая часть – дерево системы. В нём задаётся физическая конфигурация Вашей системы, т.е. указываются устройства, протоколы, каналы связи.



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



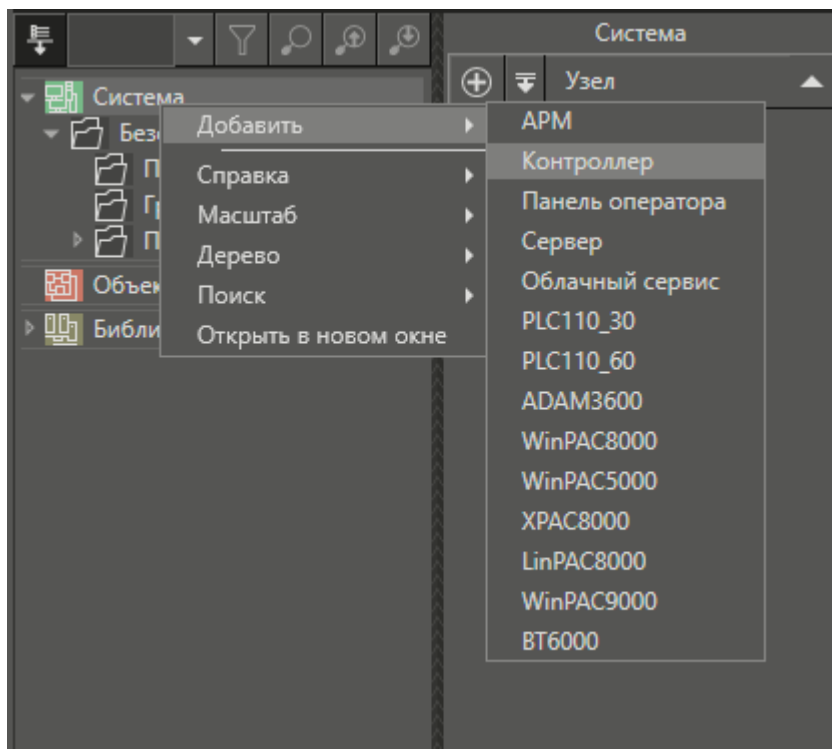
Третья часть дерева содержит библиотеки, как поставляемые со средой разработки, так и созданные пользователем.



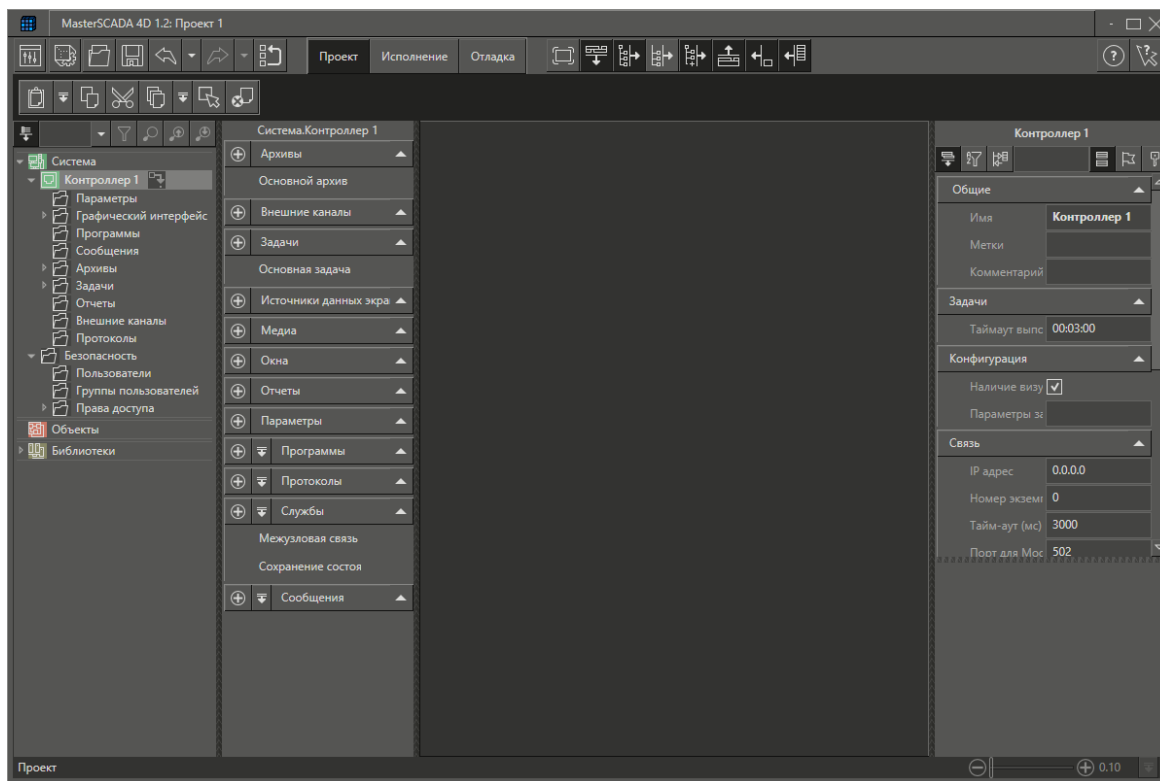
## 2.1.2.БС. Урок 1. Конфигурирование дерева системы

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.

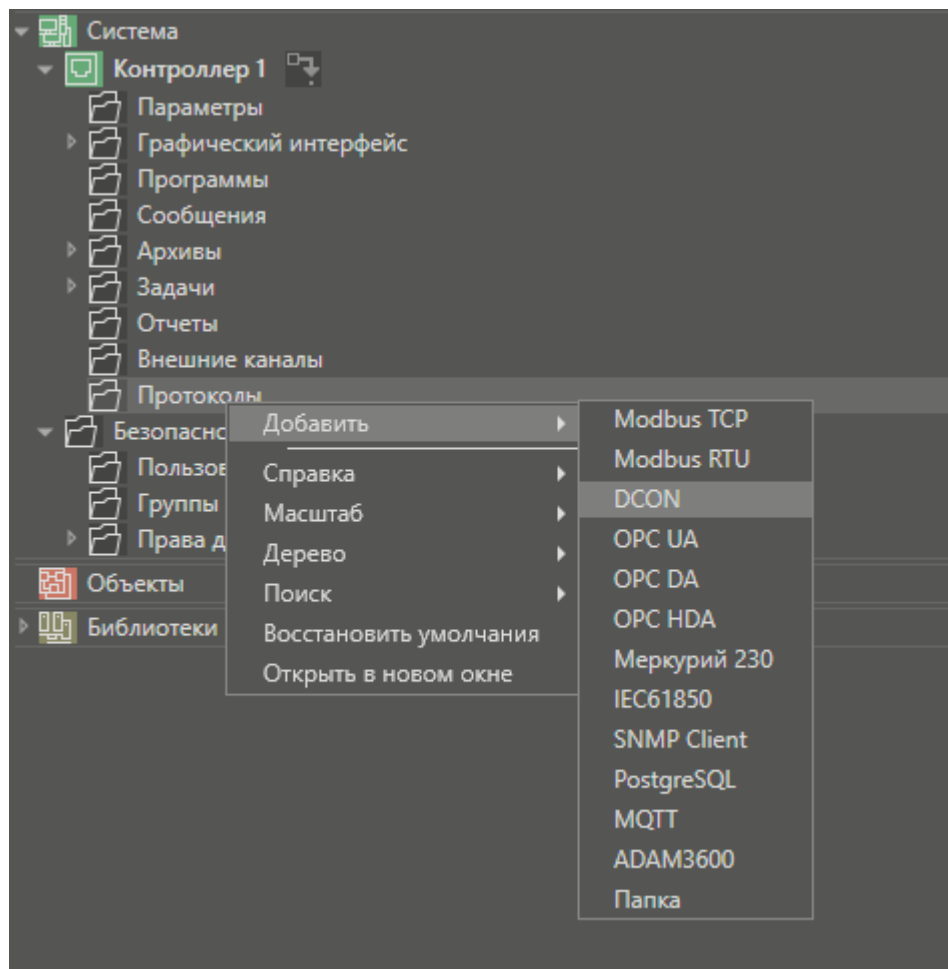


В дерево добавится Контроллер, содержащий свойственные для него элементы:



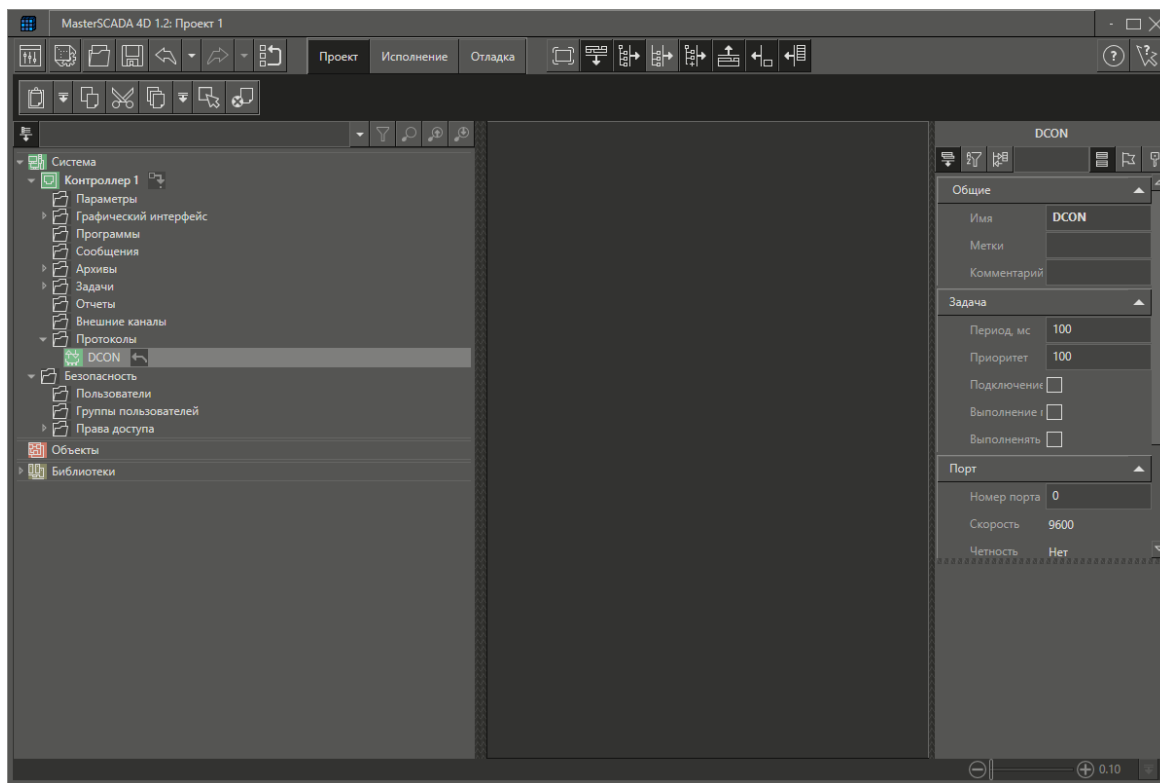
В рамках поставленной задачи нам потребуется один дискретный выход для управления устройством. Некоторые контроллеры имеют встроенные входы/выходы. Однако, в выбранном контроллере таких выходов нет, и мы будем использовать внешний модуль дискретного выхода, подключенный к контроллеру по последовательному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

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"/>

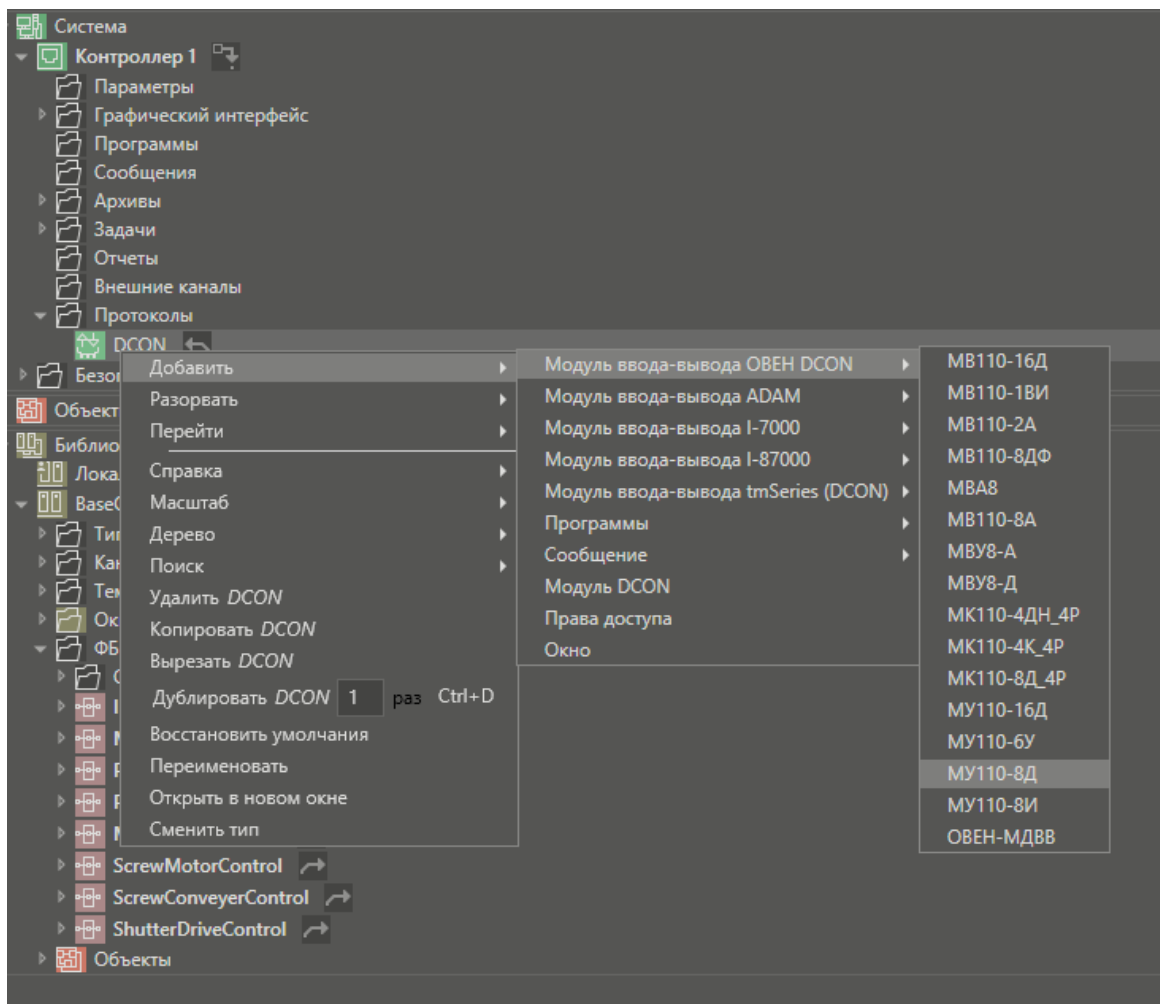
Порт	
Номер порта	4
Скорость	9600
Четность	Нет
Стоп бит	1
Разрядность символа	8

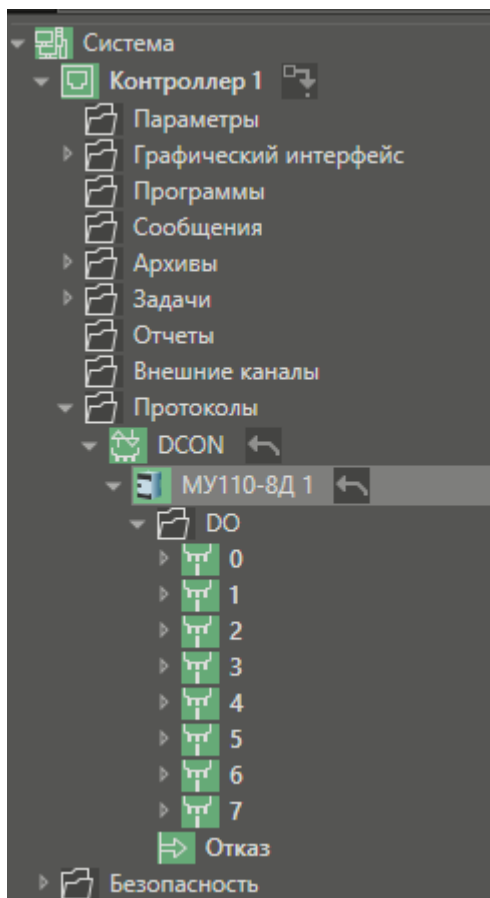
Протокол	
Таймаут	100
Контрольная сумма	<input type="checkbox"/>

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.



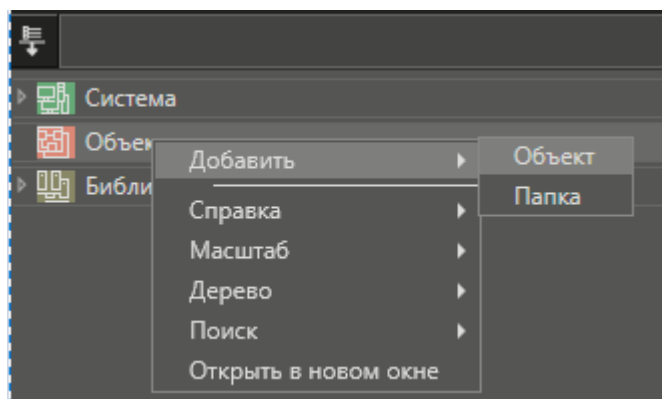
При этом в дерево системы добавится соответствующий элемент, который по умолчанию уже будет иметь правильный набор каналов.

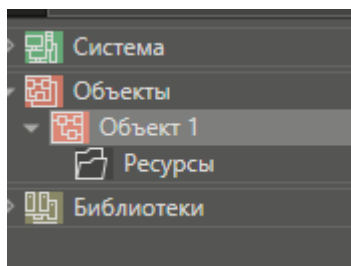


На этом конфигурирование физической структуры системы закончено. Перейдём к созданию логической части проекта.

### 2.1.3.БС. Урок 1. Создание логической части проекта

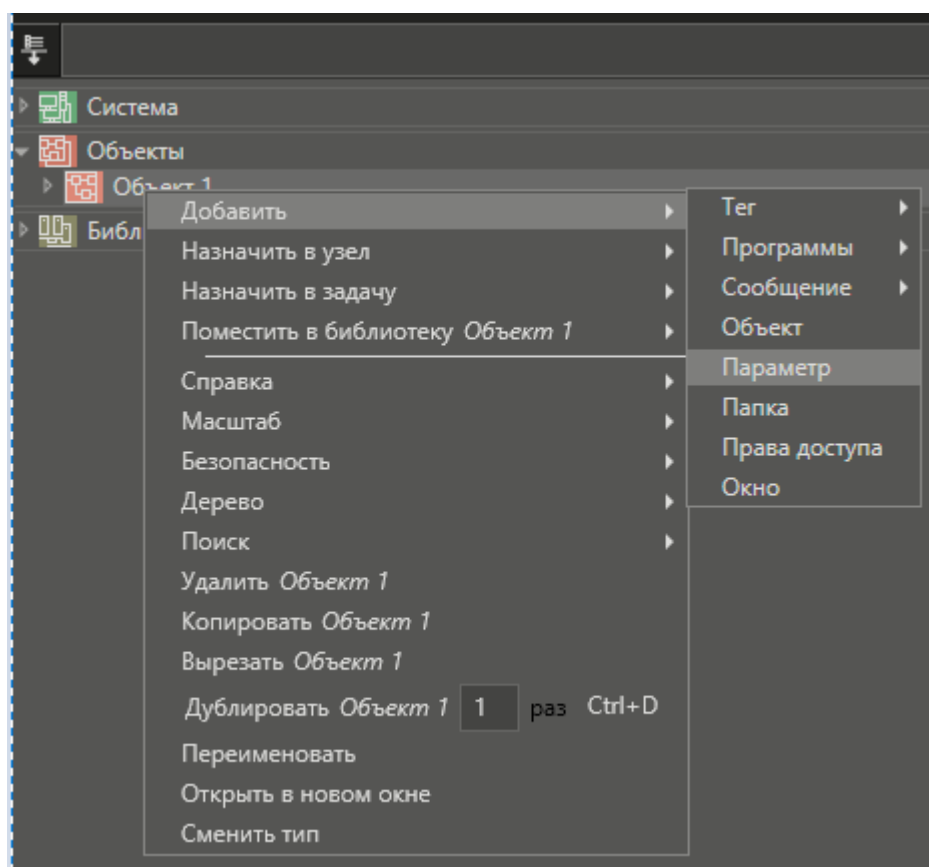
Для начала вызовем контекстное меню дерева объектов, нажав правую кнопку мыши, и выполним пункт Добавить - Объект.



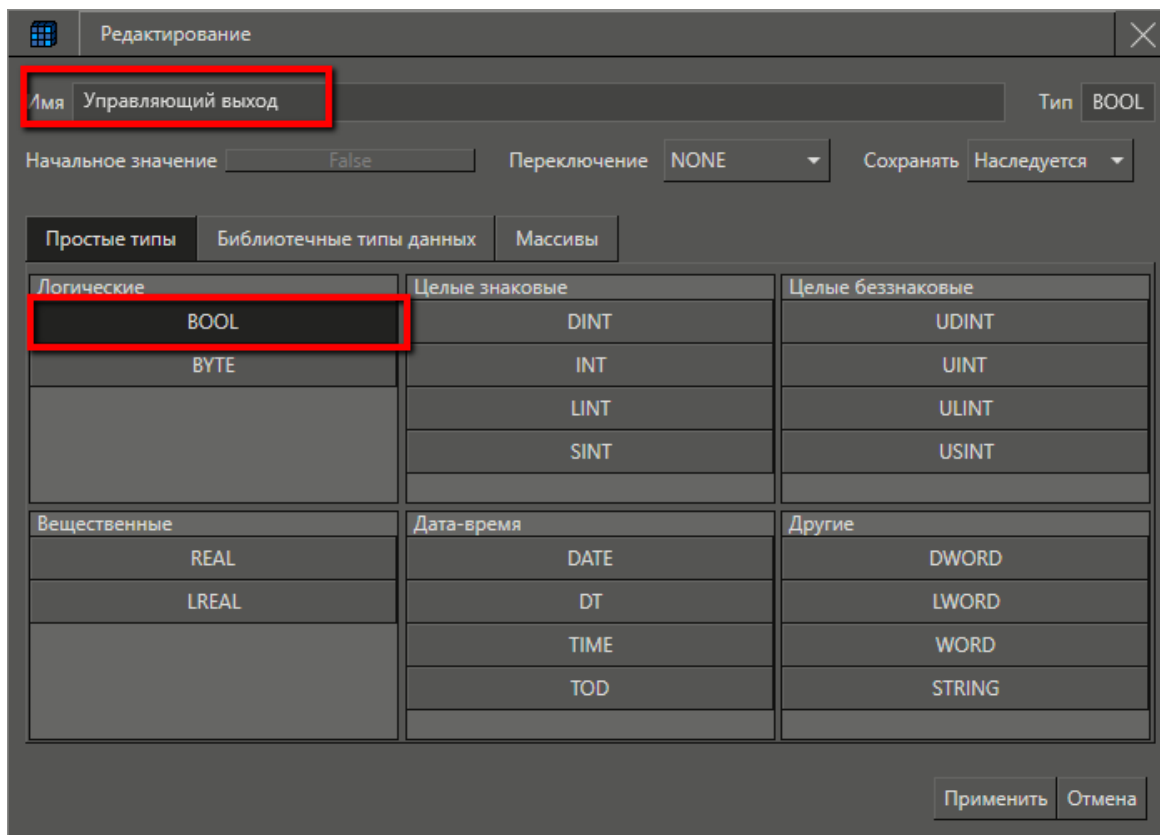


Помимо параметров, в объект можно добавлять программы для обработки данных, графические окна и другие элементы.

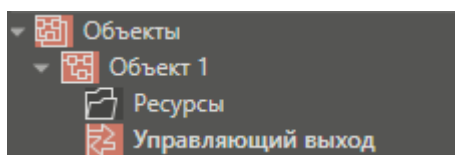
В рамках поставленной задачи у нас будет один управляющий сигнал. Нажмем правой кнопкой мыши на объект, и из контекстного меню выберем Добавить - Параметр.



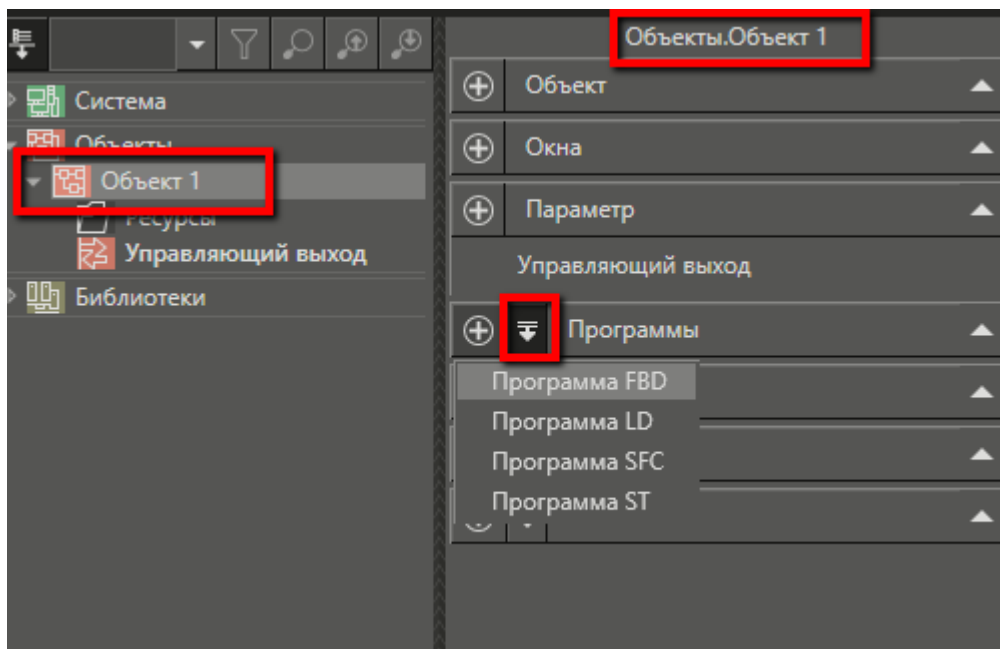
В открывшемся окне свойств параметра зададим ему имя, например, Управляющий Выход и укажем тип параметра – BOOL (логический).



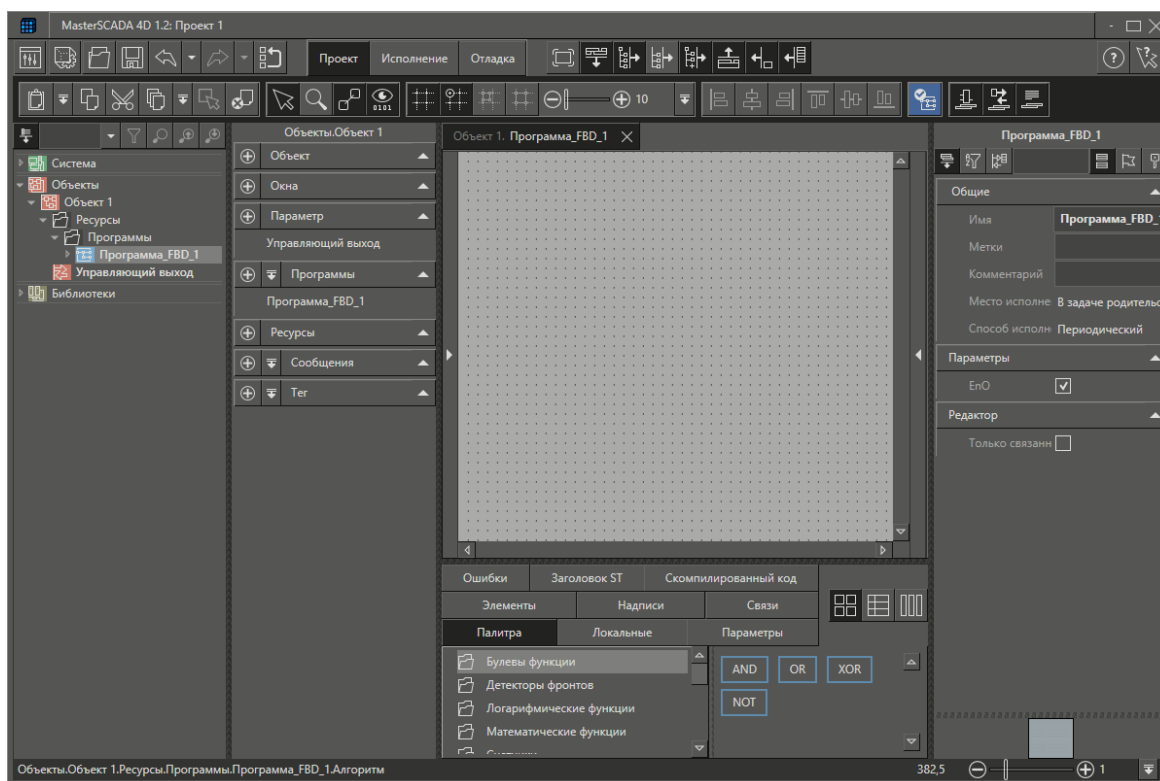
В результате получим:



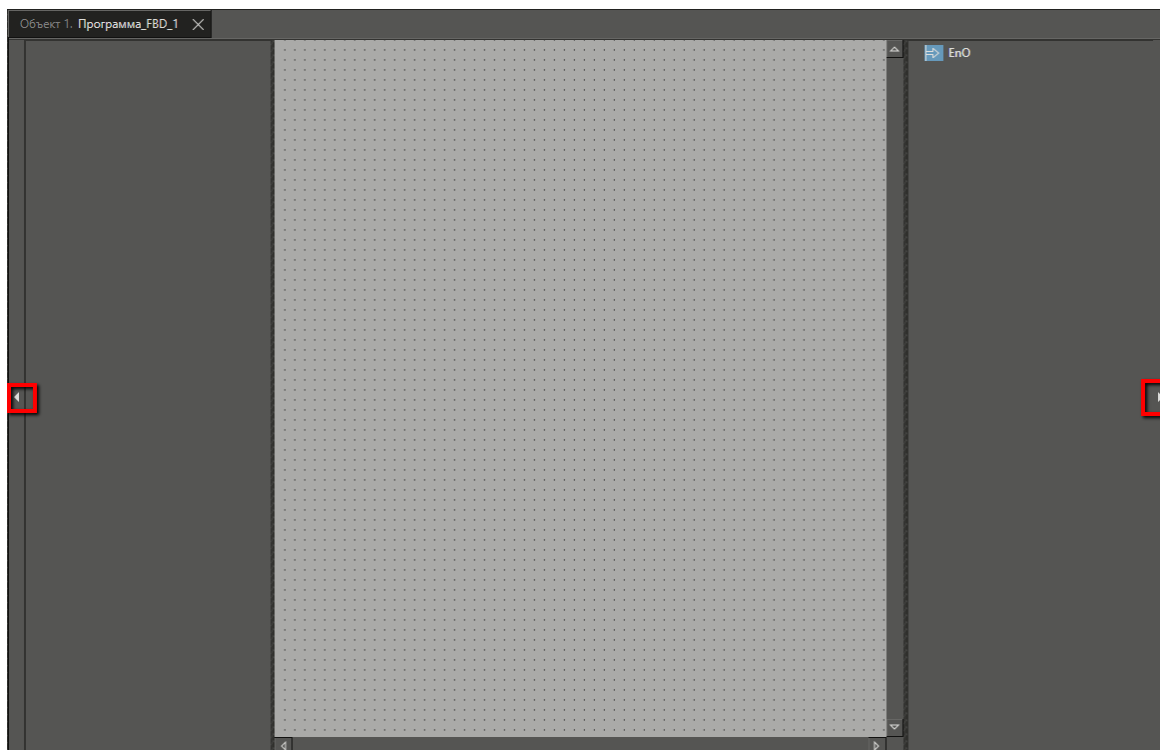
Теперь создадим управляющую программу. Для этого воспользуемся контекстной панелью элемента дерева Объект 1. На рисунке ниже показано, что данная панель дает возможность добавить программу на одном из четырех языков стандарта МЭК 61131-3. Выберем программу на языке функциональных блоков – FBD.



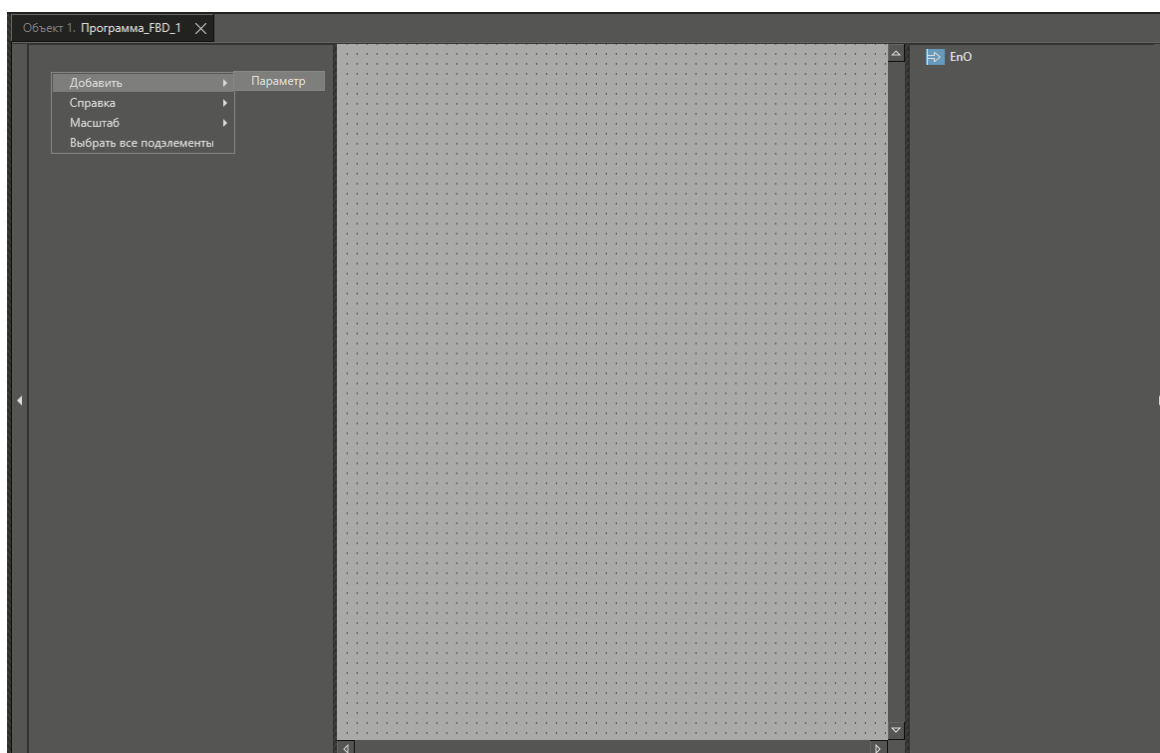
При этом откроется окно редактора этого языка.



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.

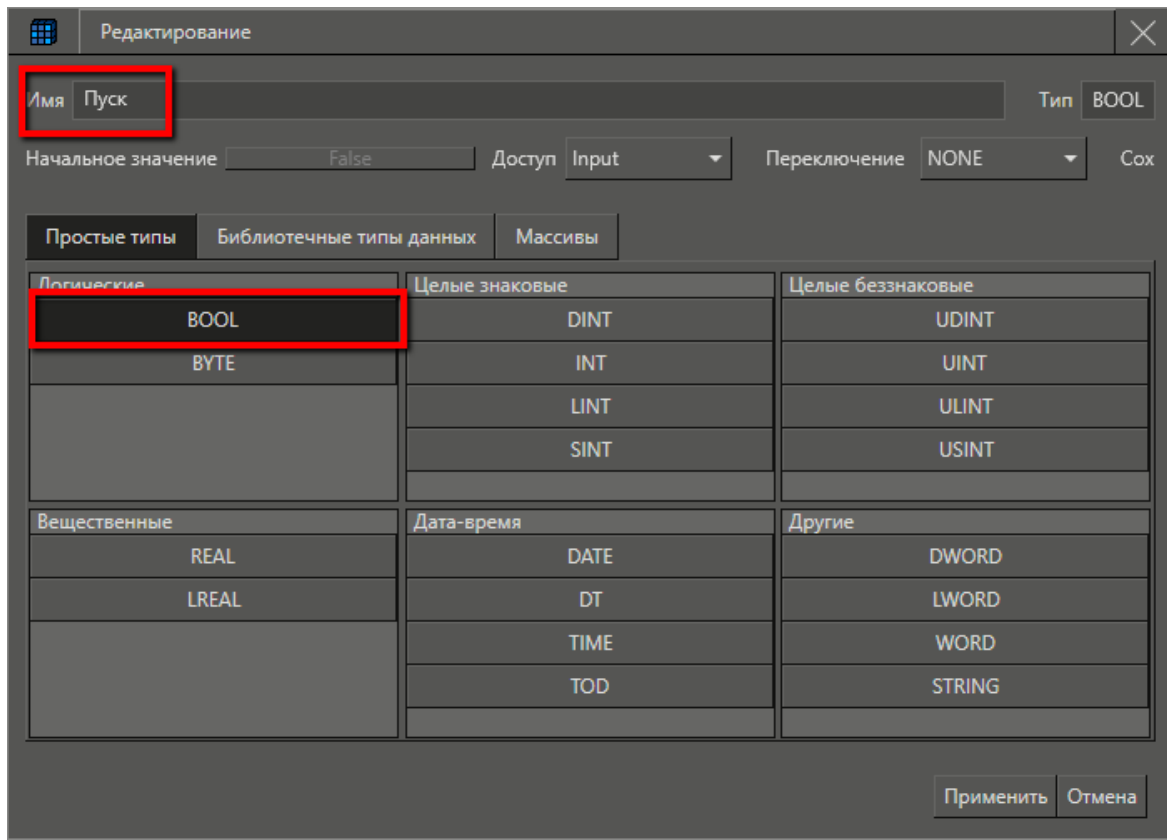


Создадим в программе два входных параметра: Пуск и Стоп . Для этого нажмем правой кнопкой мыши в любое место входного клеммника, и из контекстного меню выберем пункт Добавить - Параметр.

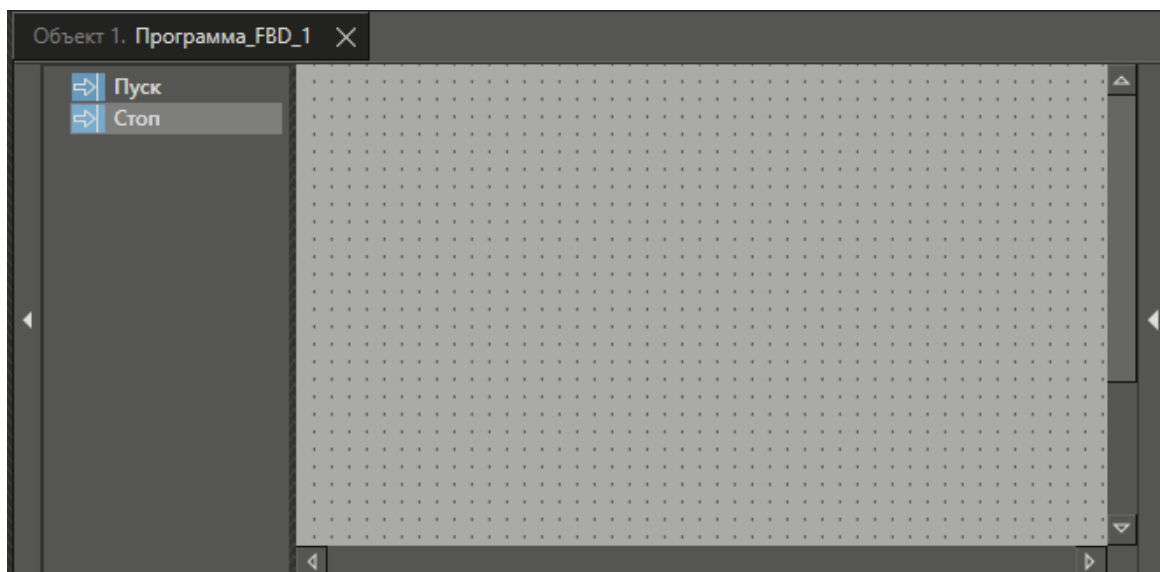
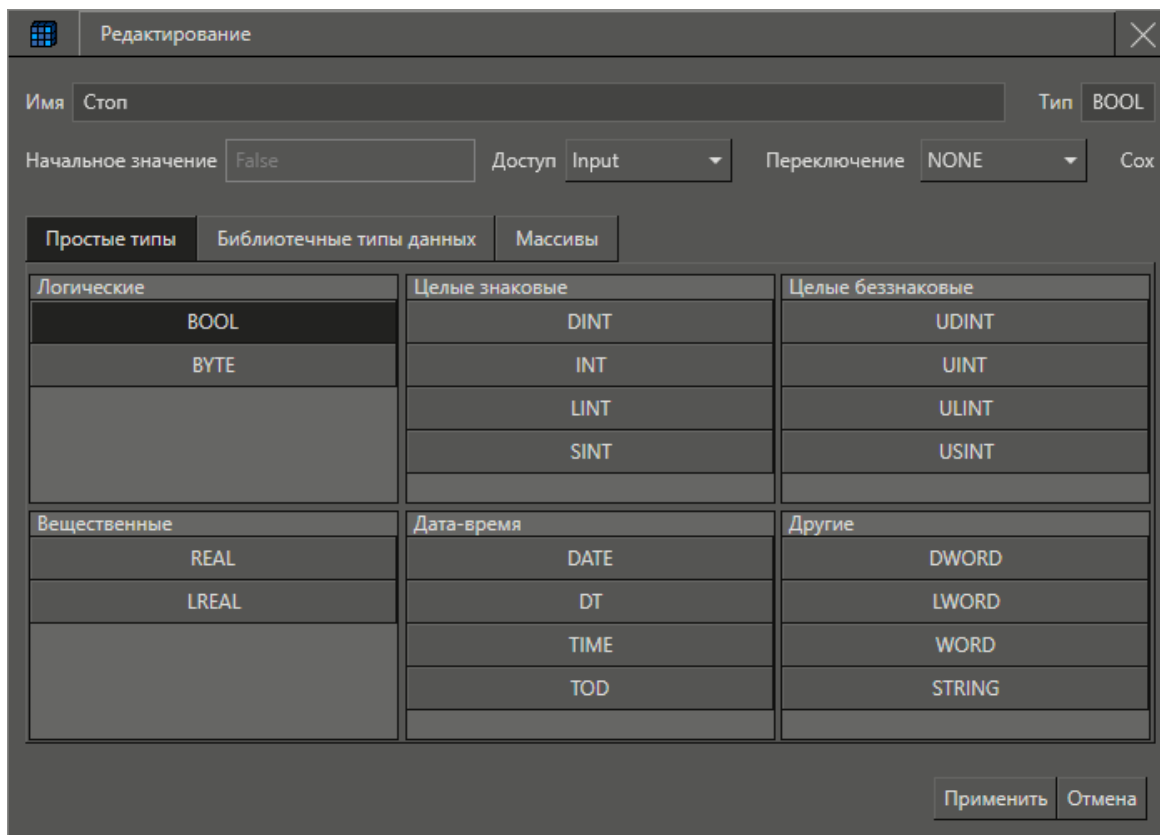


В открывшемся окне свойств параметра изменим его имя на Пуск, а тип – на BOOL.

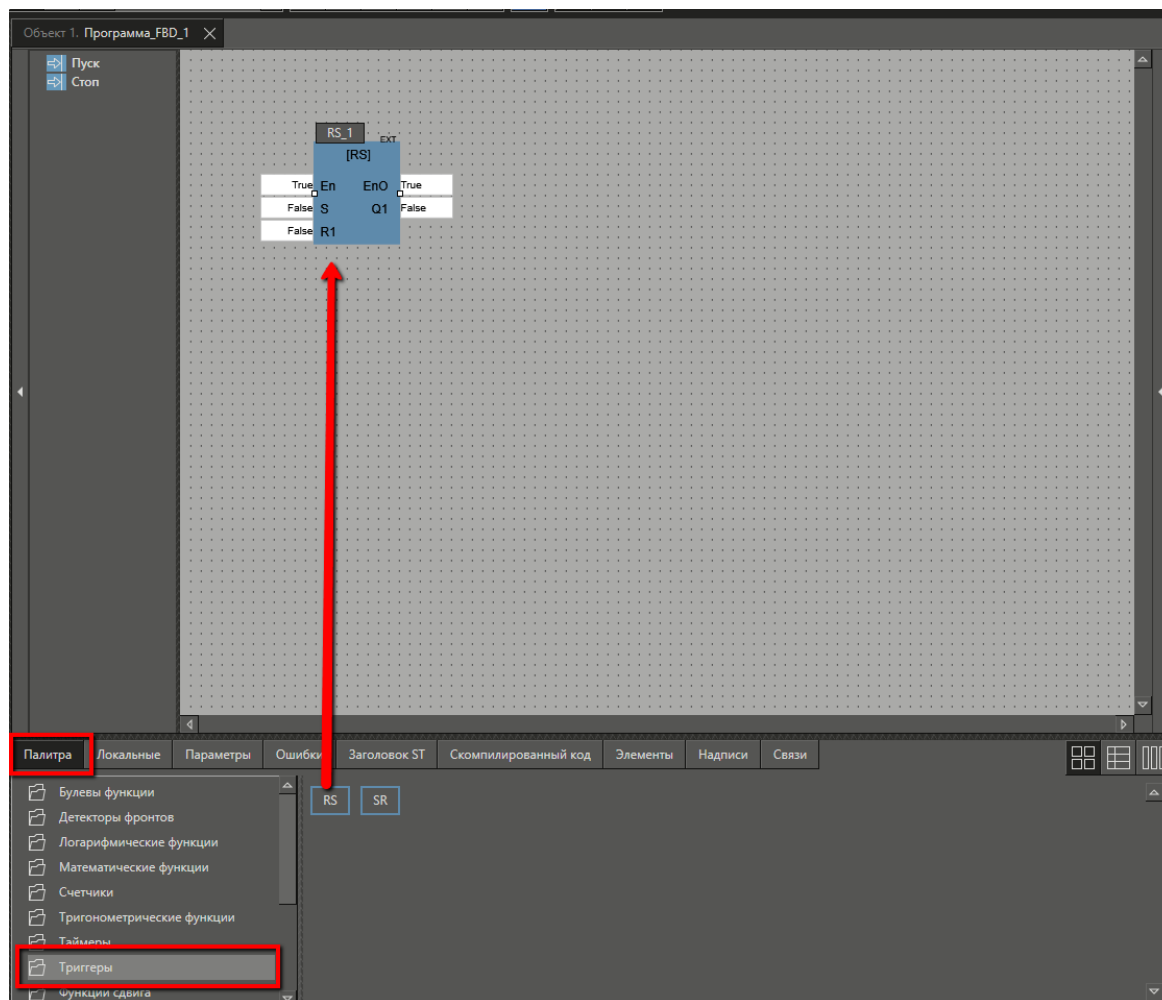




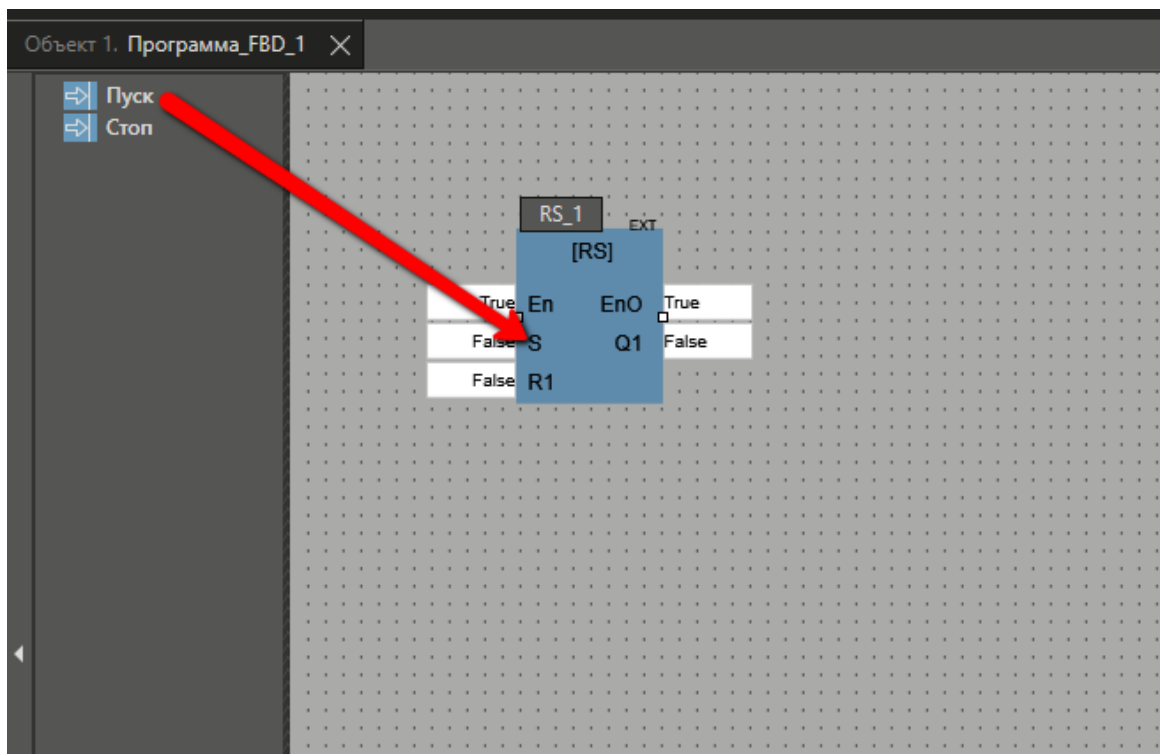
Аналогично создадим параметр Стоп.



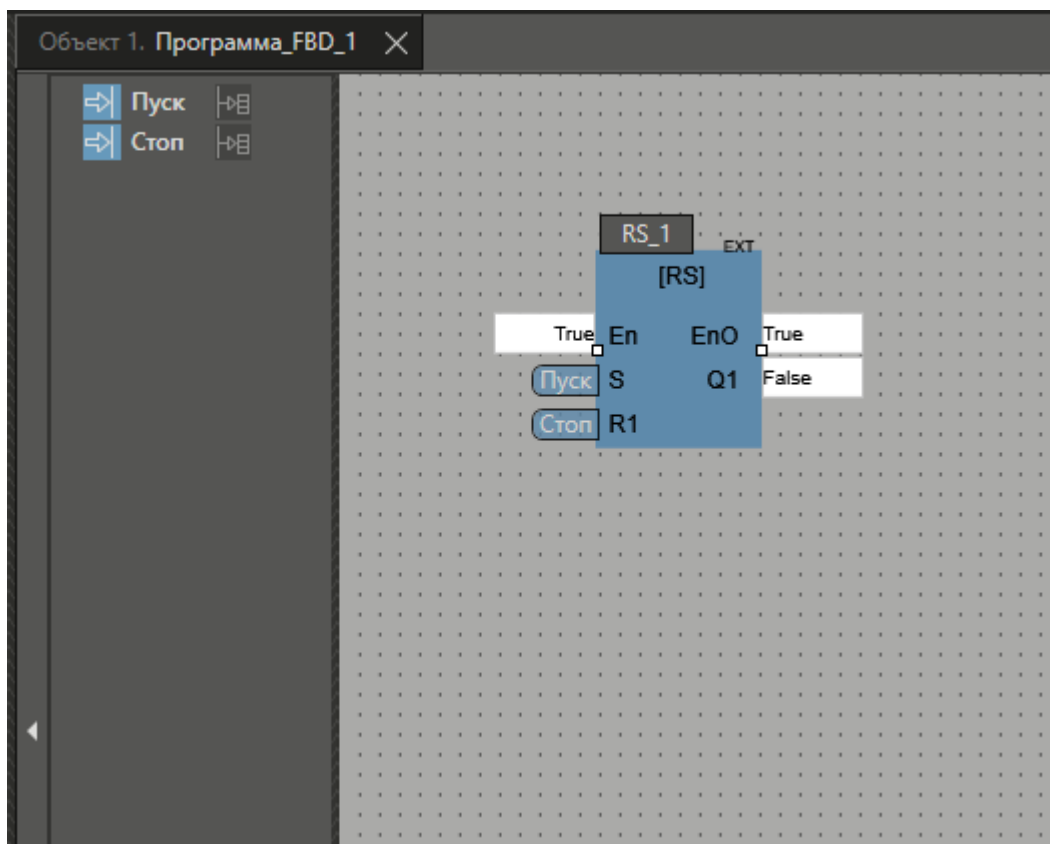
Для формирования логики включения используем обычный RS-триггер. Для того чтобы вставить на схему функциональный блок, достаточно просто перетащить его из Палитры в поле редактора. В Палитре найдем раздел Триггеры и перетащим элемент RS на схему.



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.

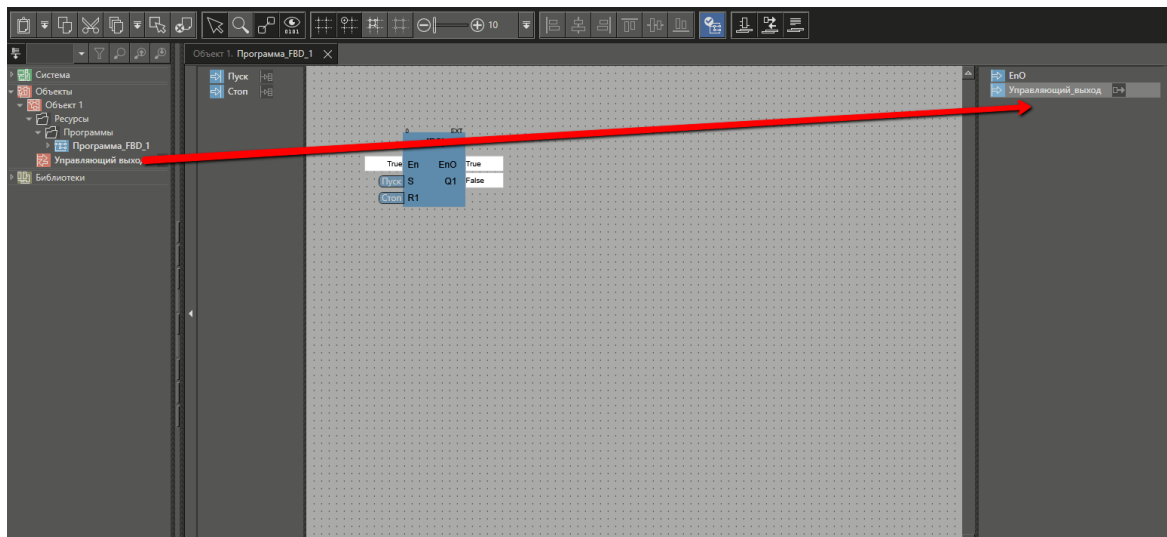


Аналогичным способом свяжем вход программы Стоп с входом триггера R1. И получим результат:

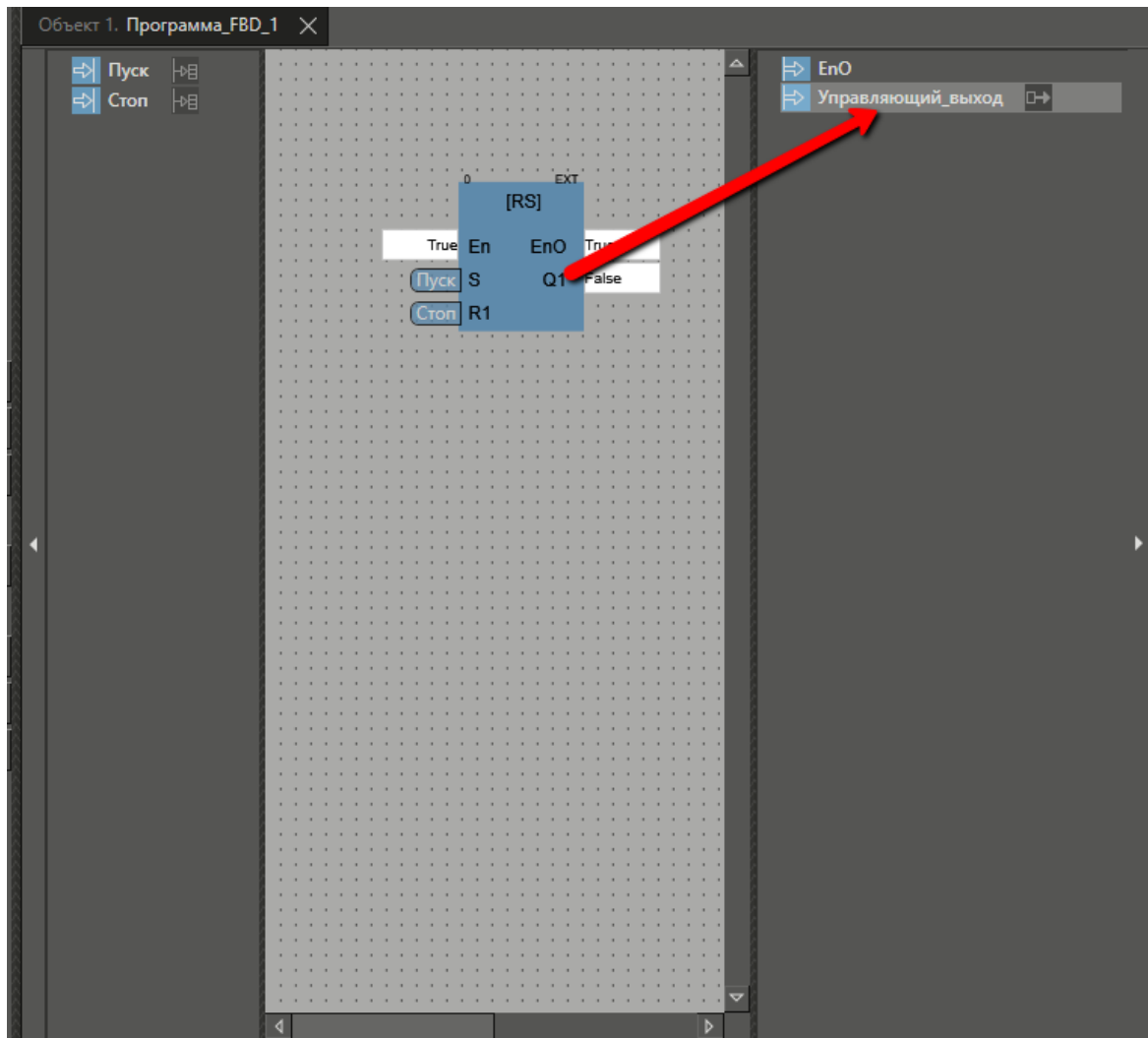


Мы ещё вернёмся к входным параметрам программы, а сейчас создадим выходной параметр.

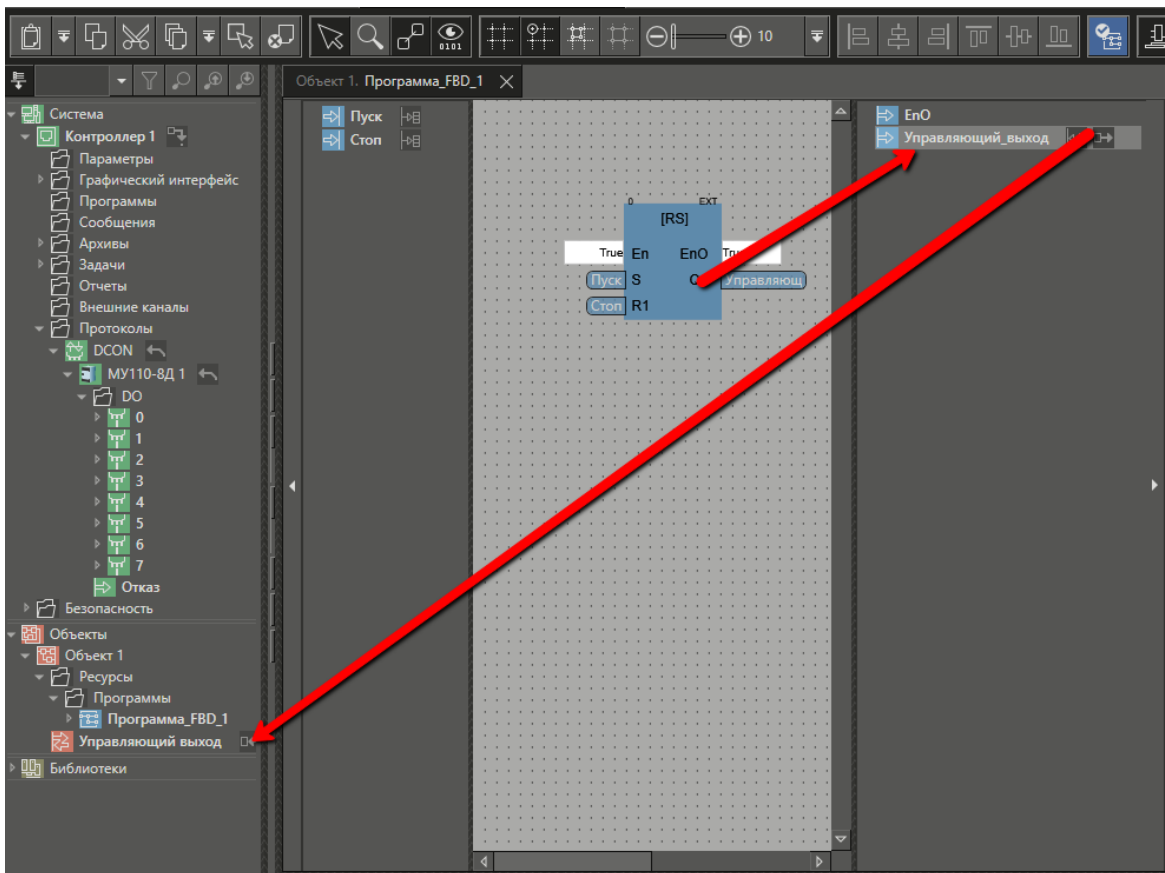
Для этого перетащим параметр объекта Управляющий Выход на выходной клеммник программы, и в результате будет создан выходной параметр программы (по умолчанию – одноимённый), а также связь с ним.



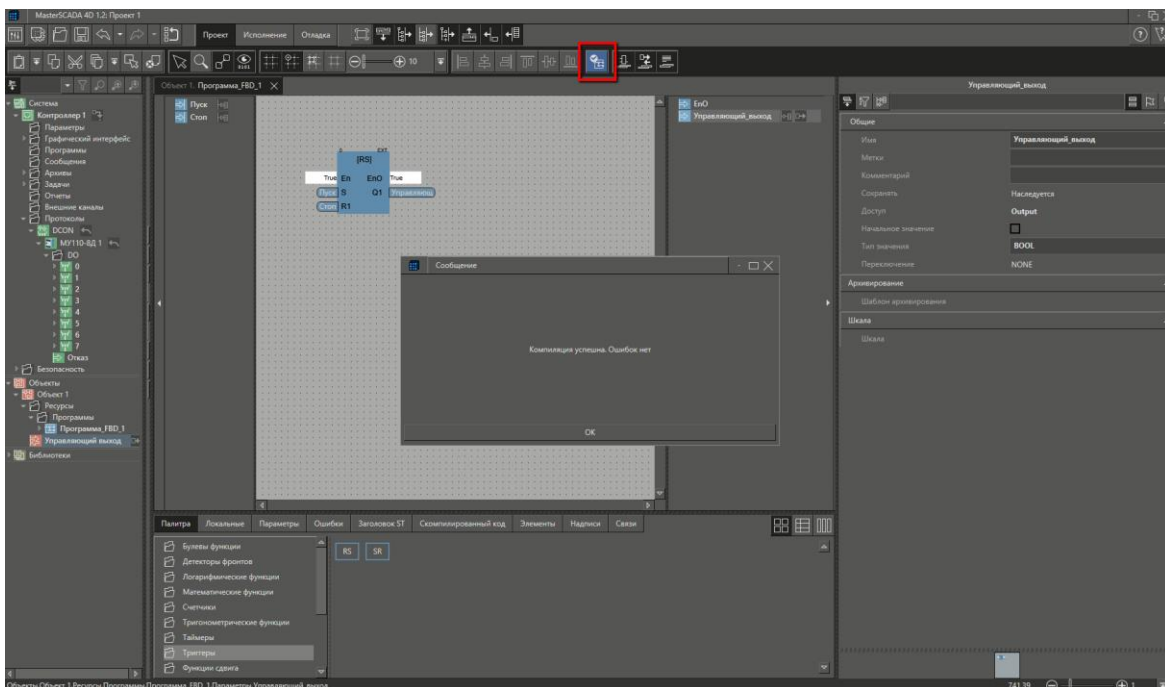
Наконец, свяжем выход триггера Q1 с выходным параметром Управляющий Выход.



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



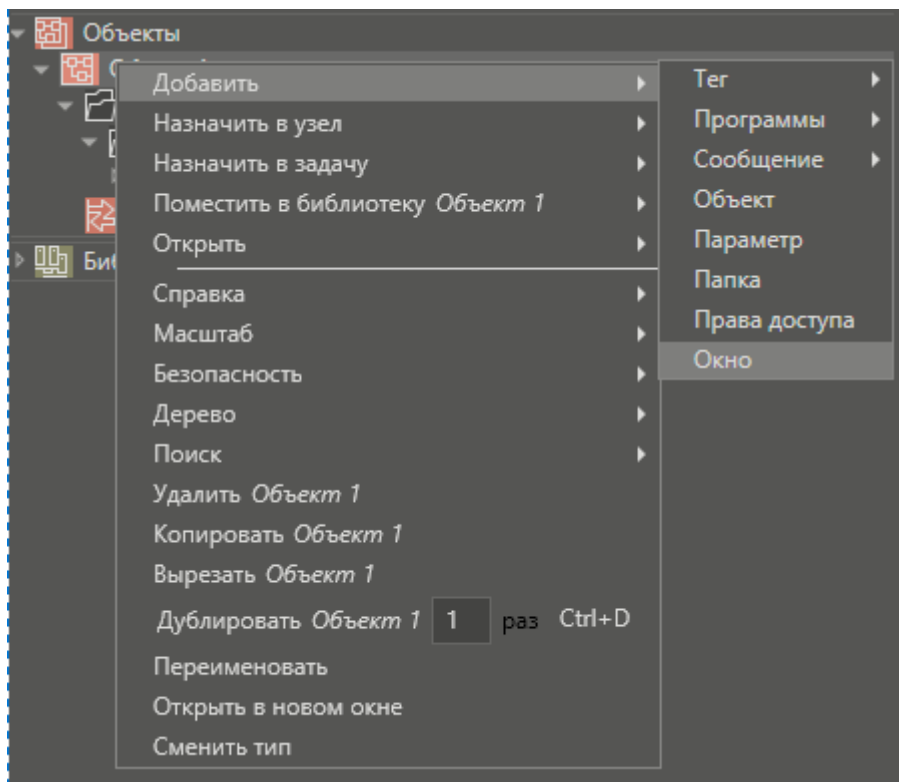
На этом формирование логики управления закончено. На всякий случай, нажмём кнопку Проверить и убедимся, что компиляция пройдет успешно.



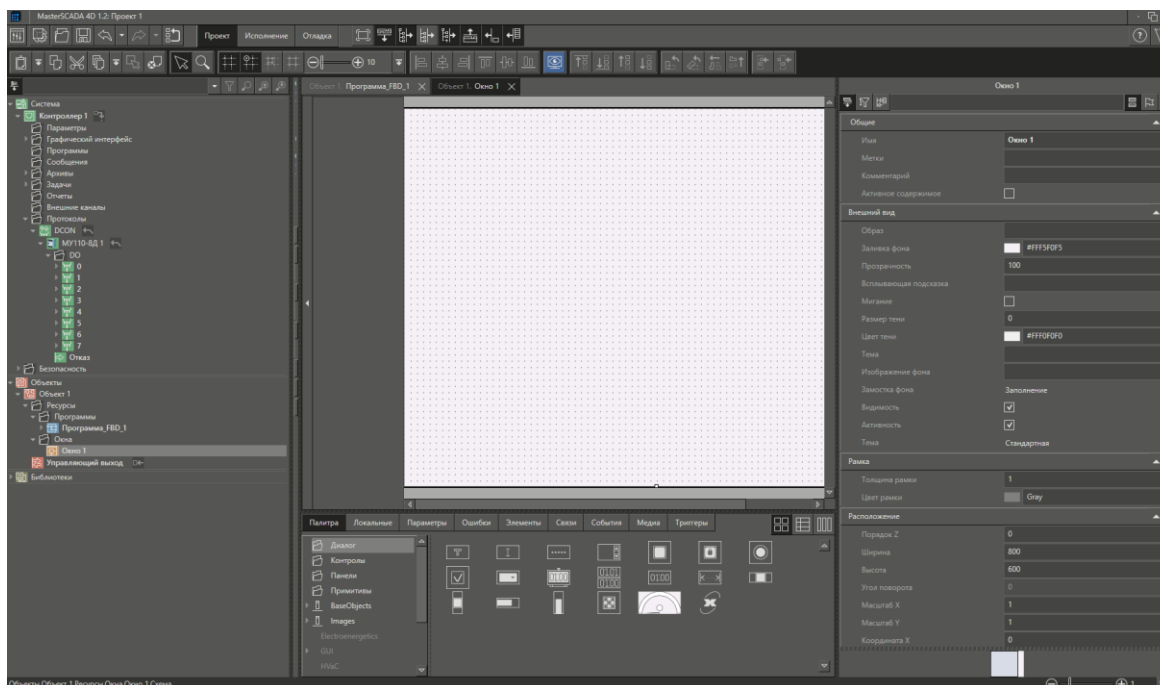
Следующий этап – создание визуализации.

## 2.1.4.БС. Урок 1. Создание окна управления

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



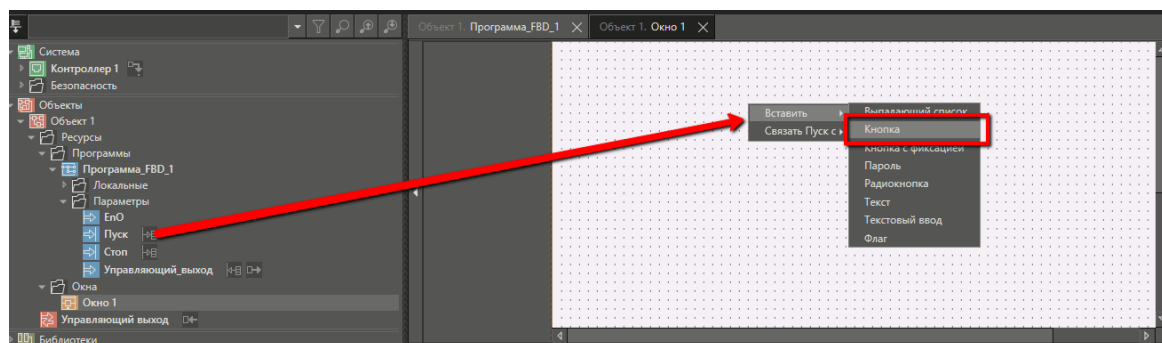
При этом у объекта появится окно, которое сразу откроется в поле редактора графического интерфейса.



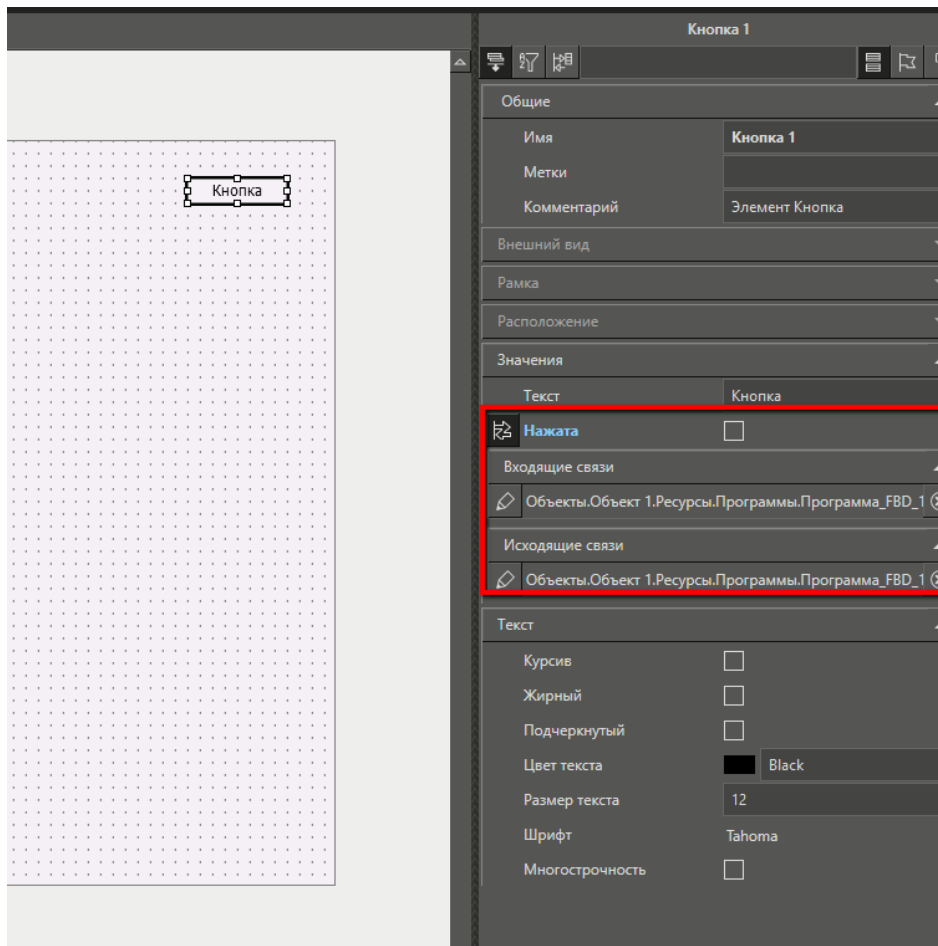


Наша задача заключается в том, чтобы разместить в этом окне две кнопки – Пуск и Стоп, нажатие которых будет передаваться на соответствующие входные параметры программы, а также создать индикацию состояния Управляющего выхода.

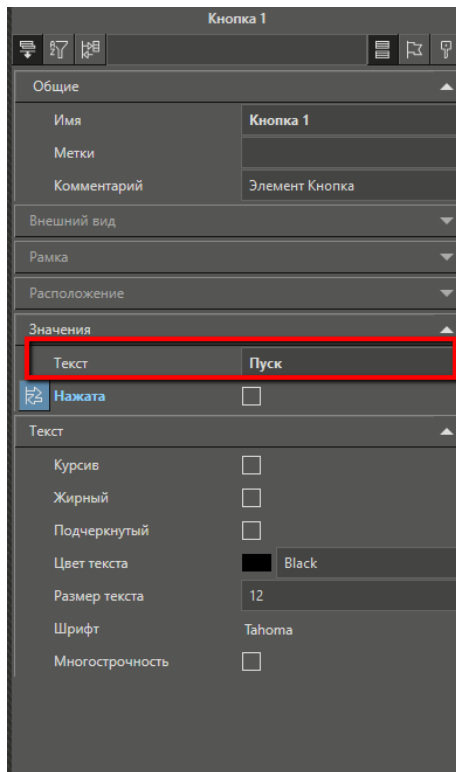
Из группы Параметры созданной нами программы перетащим правой кнопкой мыши параметр Пуск в окно. Из открывшегося контекстного меню выберем пункт Вставить.Кнопка.



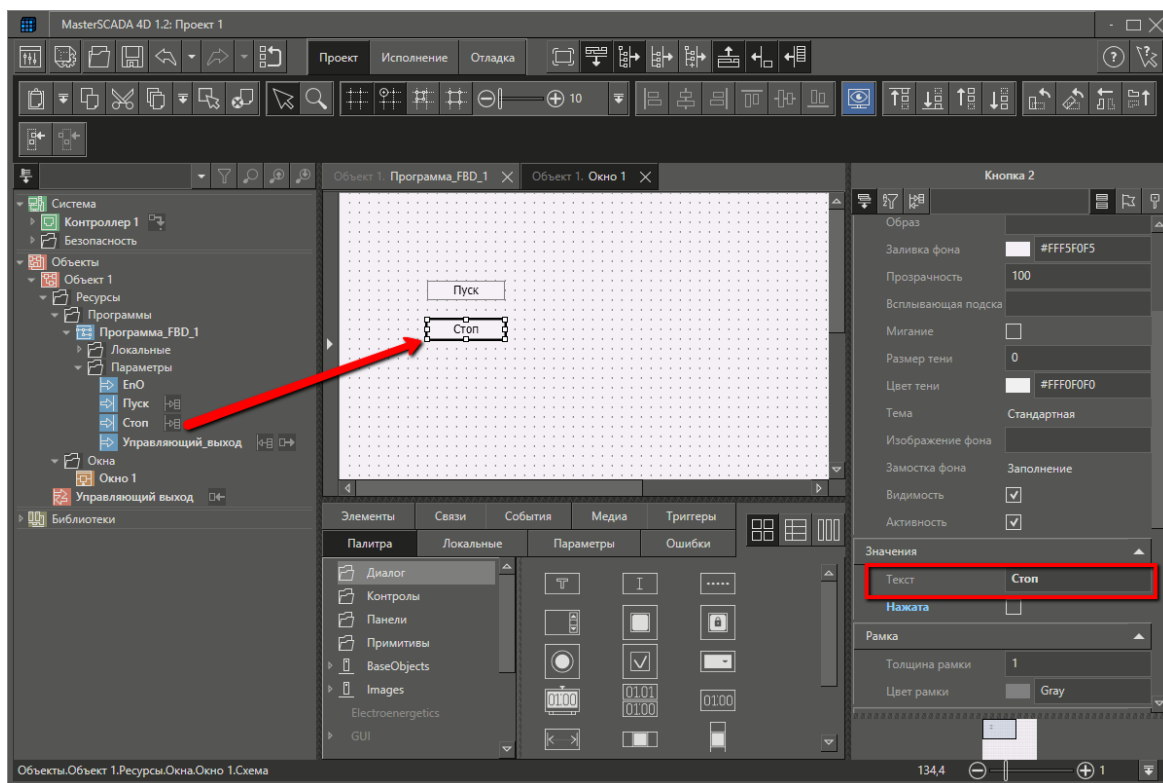
В окно будет вставлена кнопка с уже установленными связями между параметром Пуск и свойством кнопки Нажата. По нажатию кнопки параметр Пуск будет устанавливаться в TRUE, а при отпусканию кнопки этот же параметр будет установлен в False. Таким образом, параметр Пуск будет отражать состояние кнопки.



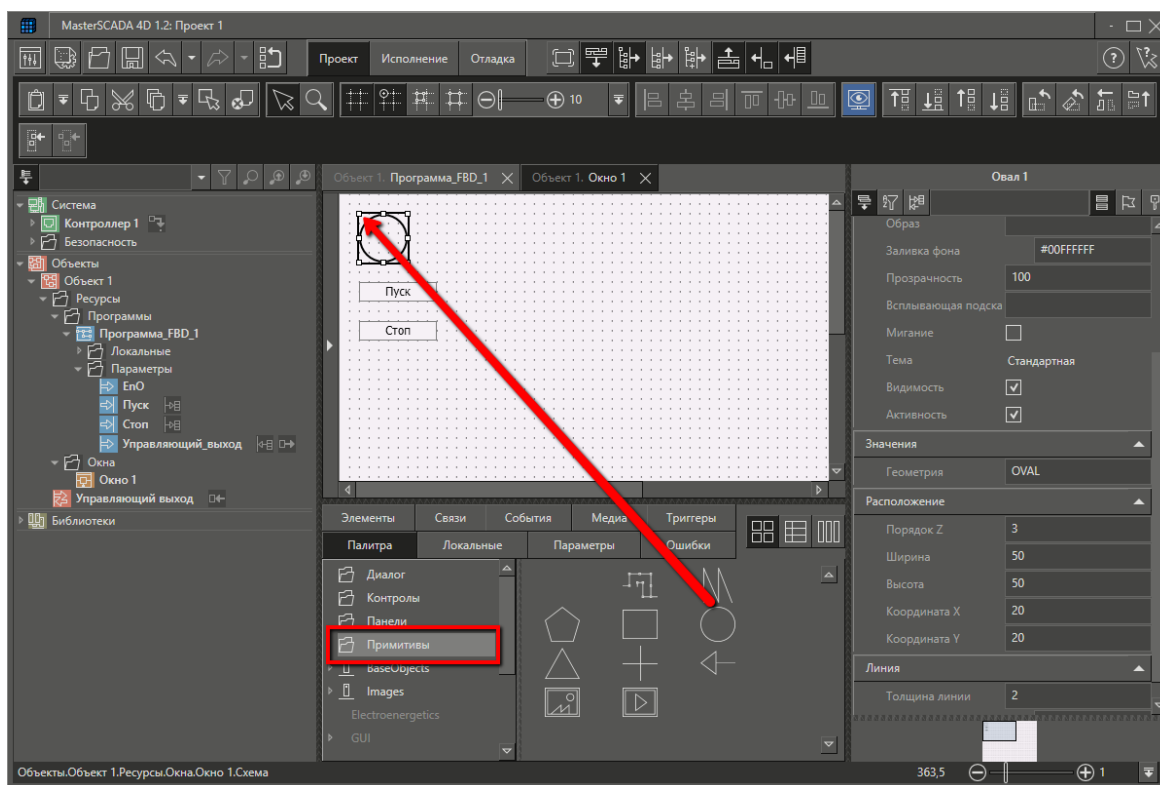
Назовем эту кнопку Пуск, набрав с клавиатуры это название в графе текст панели свойств:



Аналогичные действия произведём с параметром Стоп.

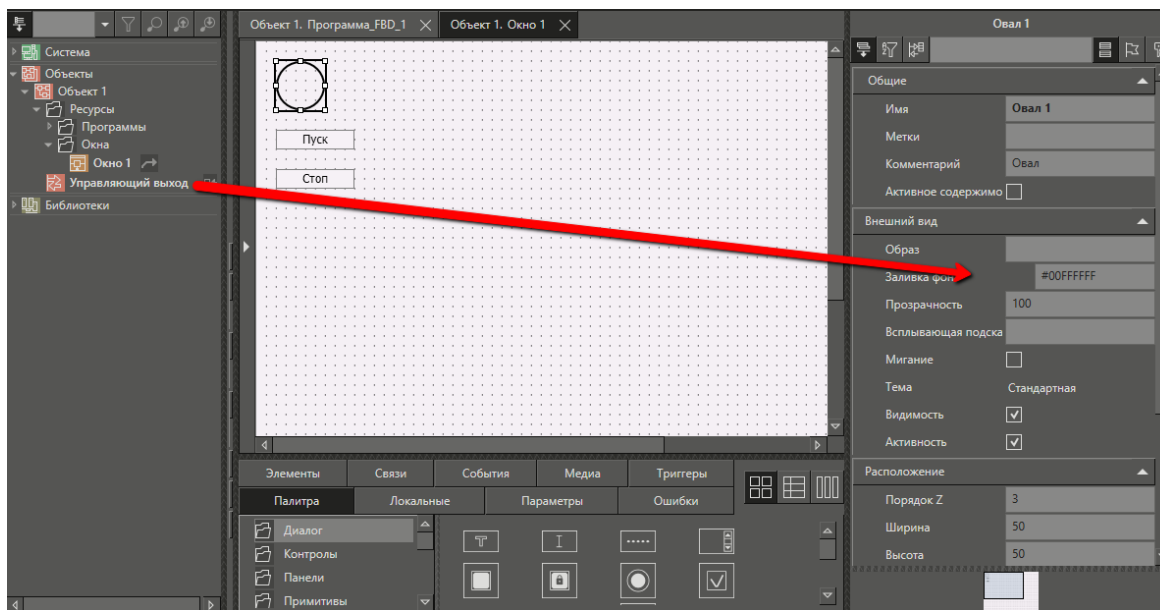



Теперь реализуем отображение состояния Управляющего выхода. Для этого перейдём в Легенду графического редактора, откроем вкладку Палитра, выберем категорию Примитивы, и перетащим левой кнопкой мыши элемент Овал в окно (мнемосхему).



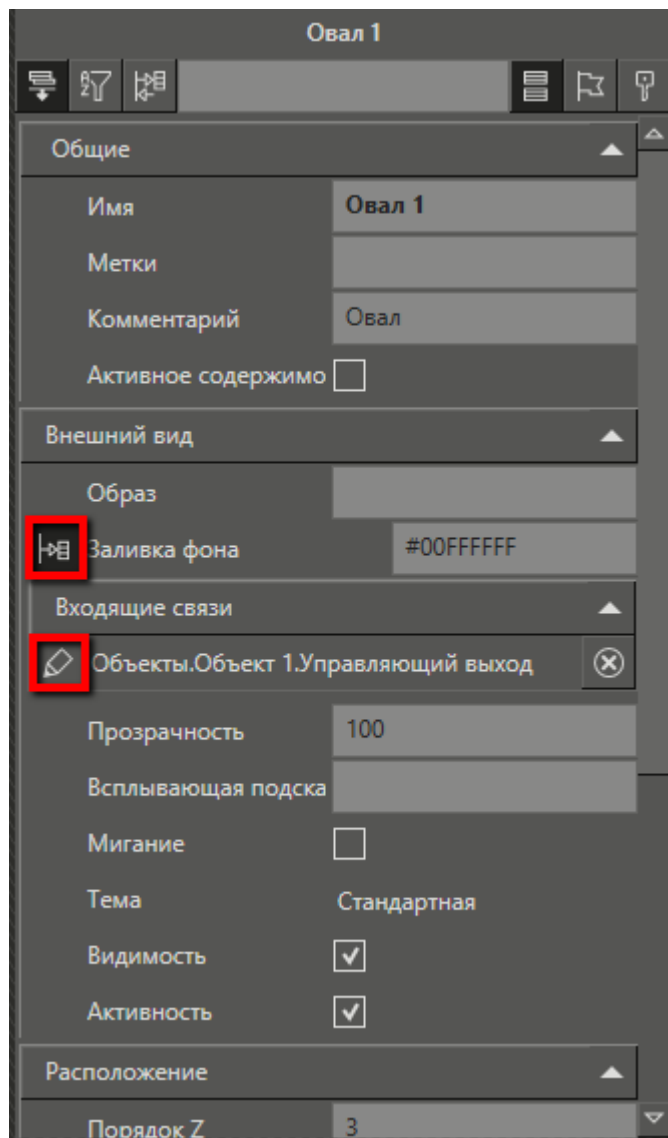
Сделаем так, чтобы цвет овала отображал текущее состояние выходного сигнала: включен или выключен.

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

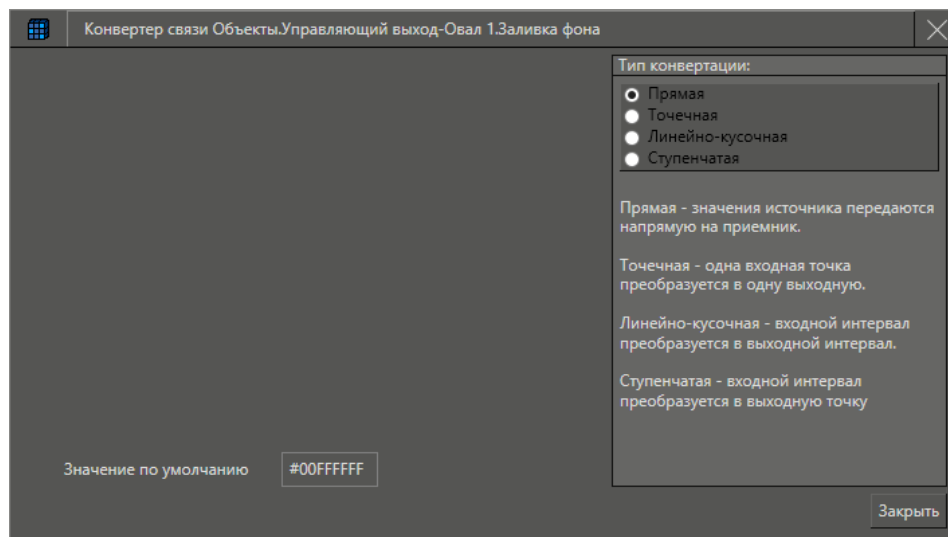


Справа от свойства появится значок связи . При нажатии на него левой кнопкой мыши откроется дополнительная информация о созданной связи. Нажмём на значок

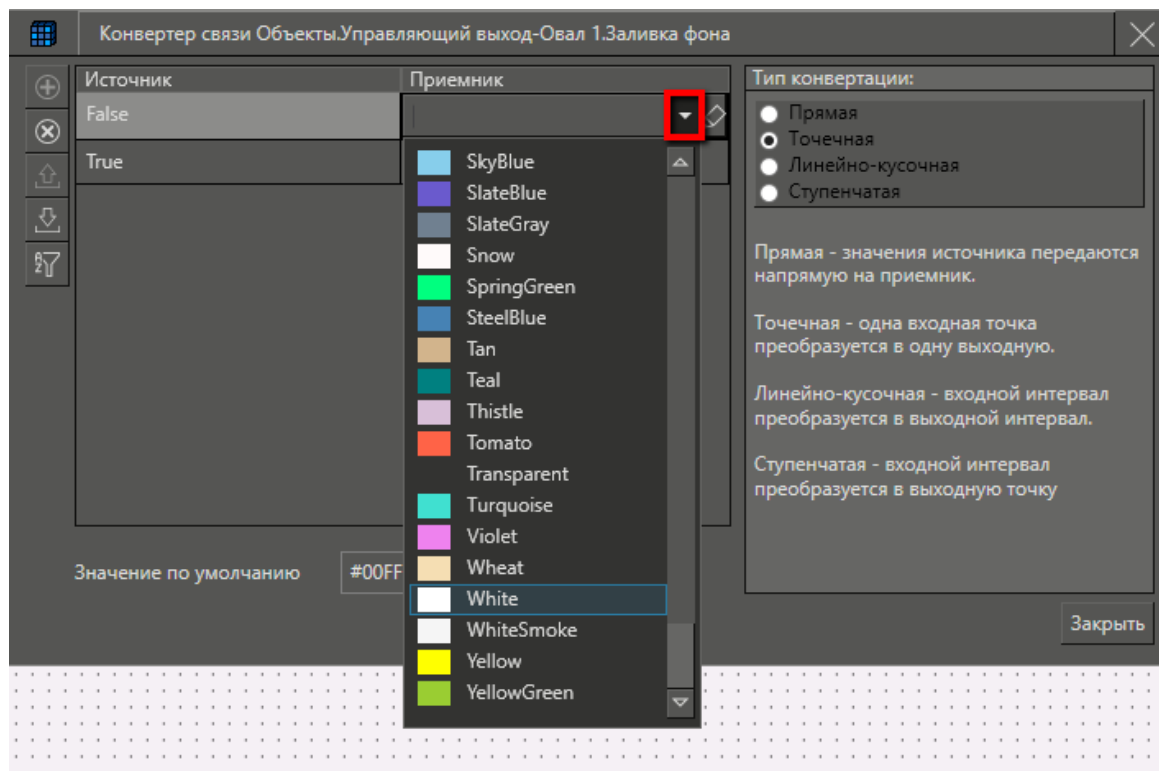




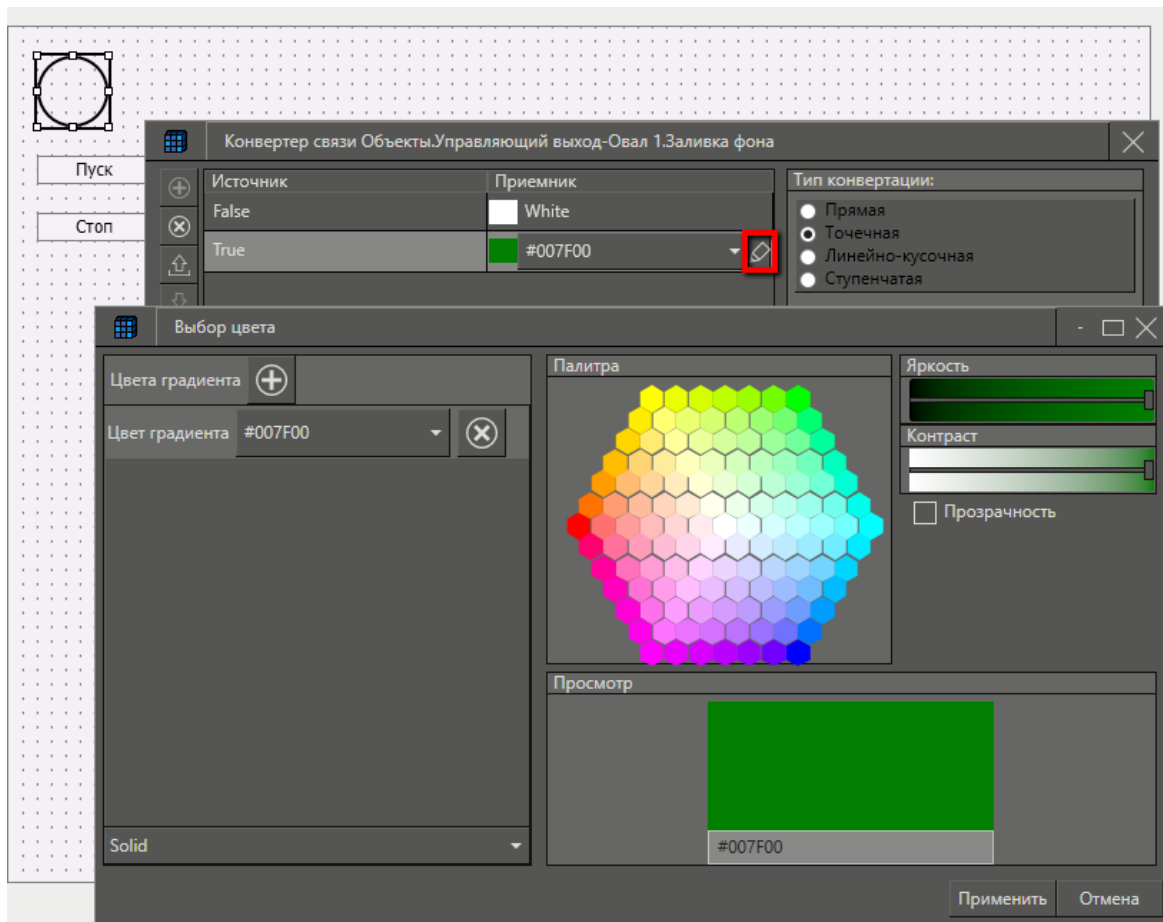
Откроется диалоговое окно:



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.

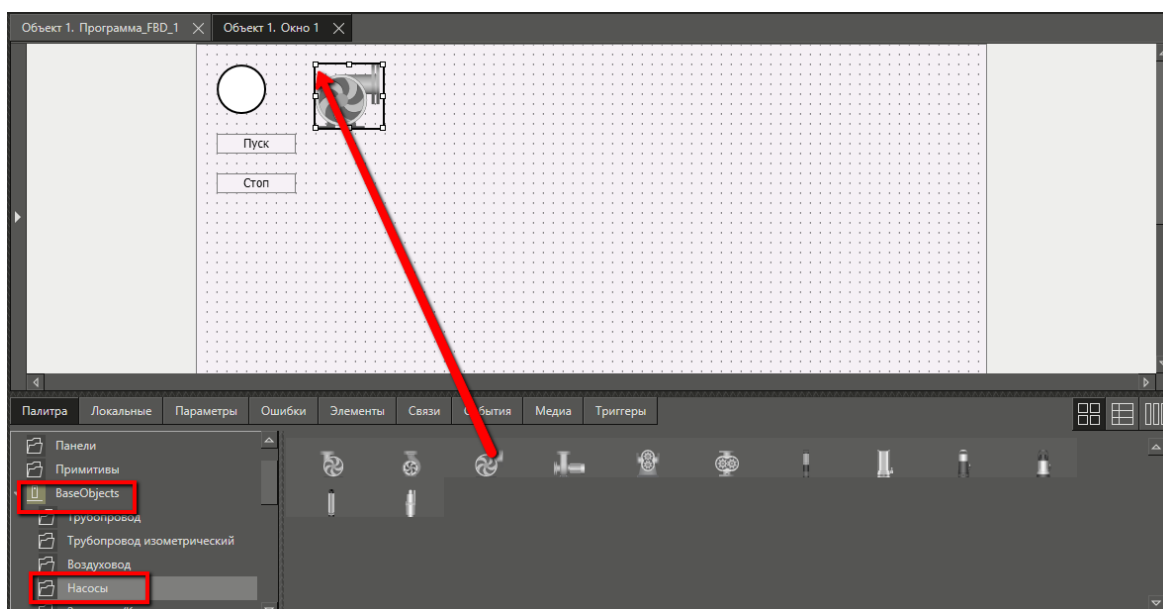


Для значения TRUE выберем зеленый цвет из палитры.



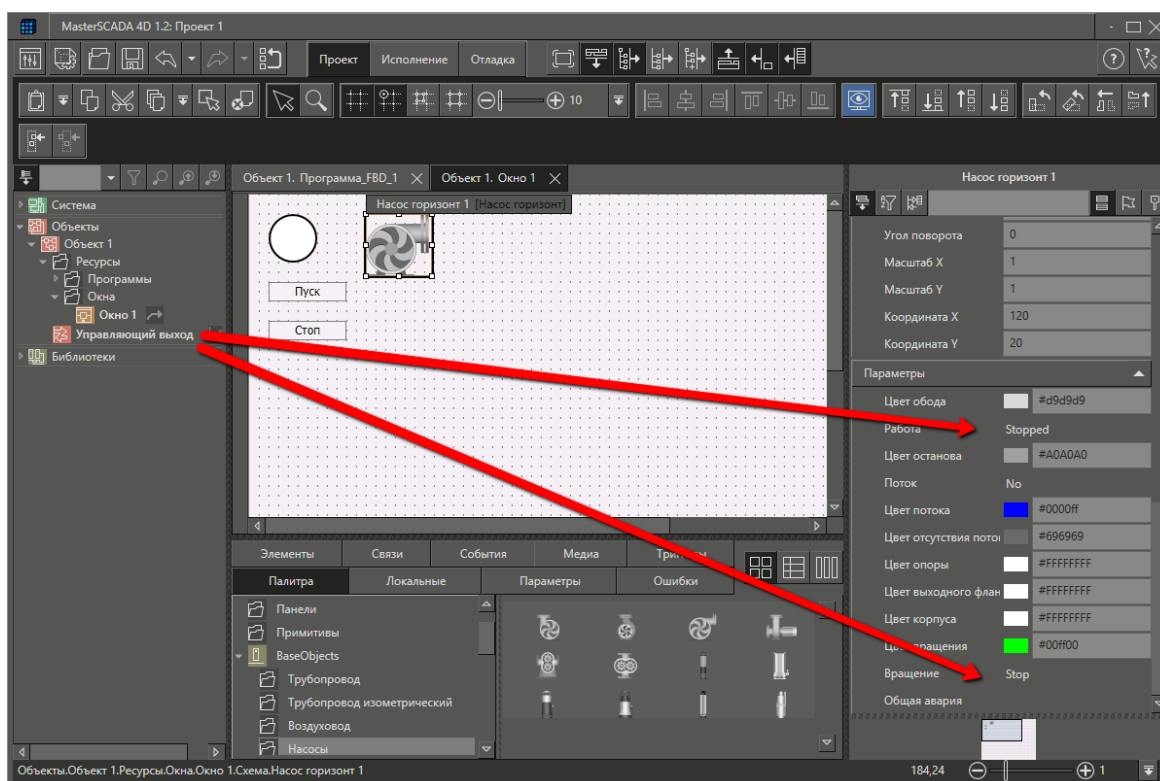
После этого закроем окно конвертации.

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



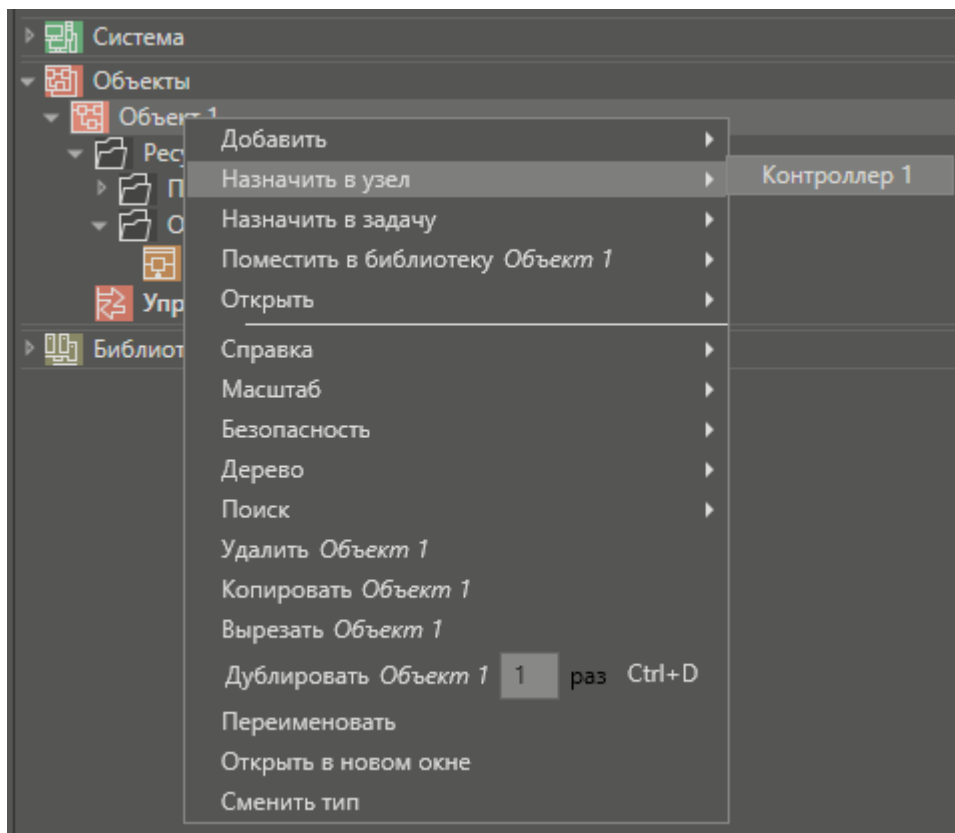


Наконец, свяжем параметр объекта Управляющий выход со свойствами насоса Вращение и Работа.

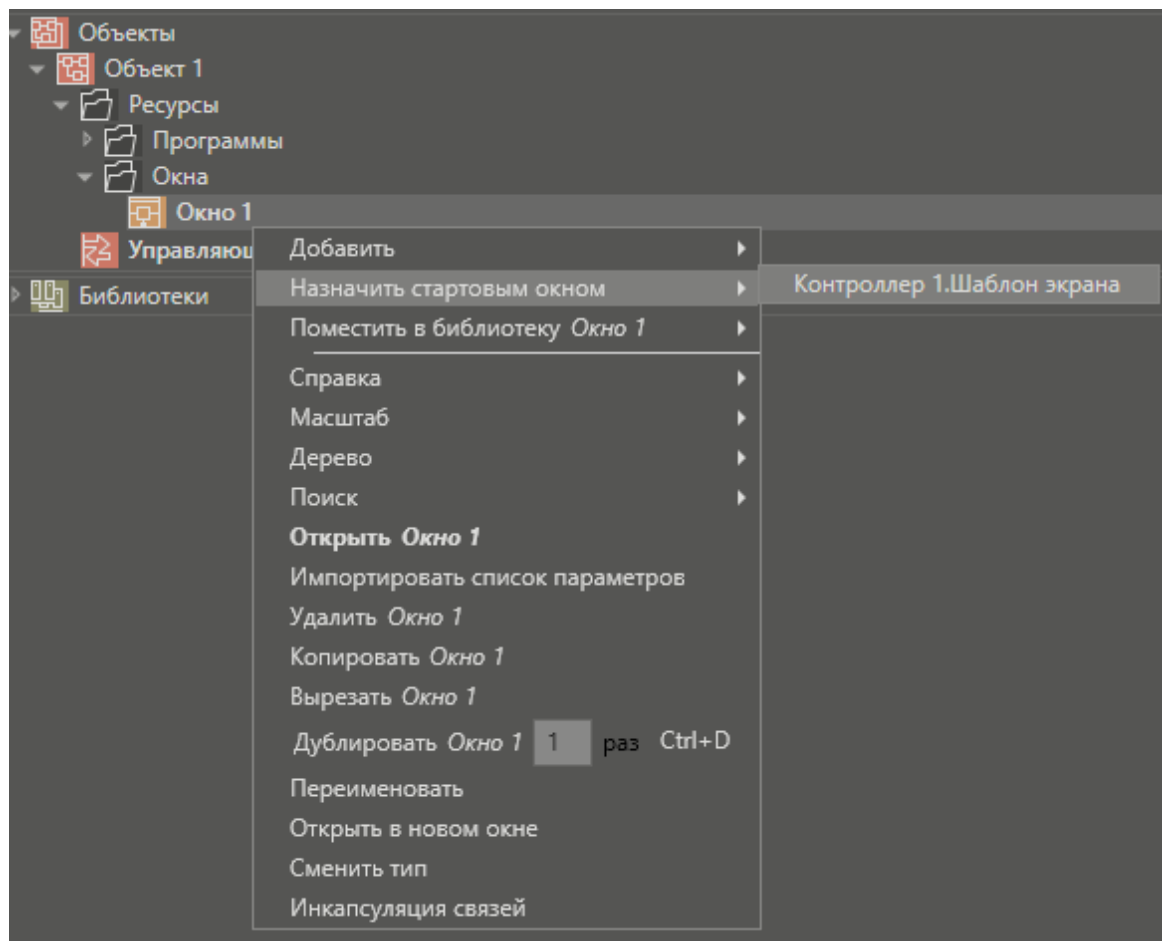


На этом формирование визуализации завершено.

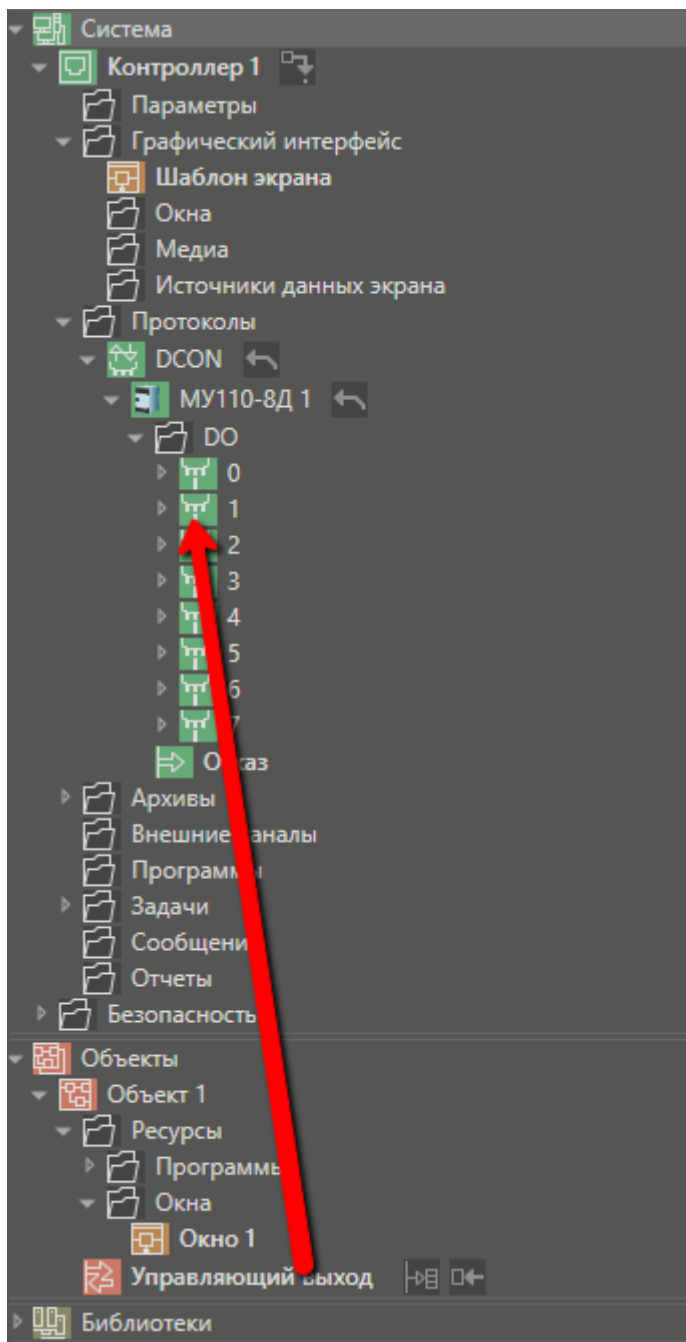
Теперь назначим объект для исполнения в узле контроллера. Это необходимо сделать потому, что узлов в проекте может быть много, и любой объект может быть назначен для исполнения в любом из них. Для этого нажмем правой кнопкой мыши на объект и из контекстного меню выберем Назначить в узел, а затем выберем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



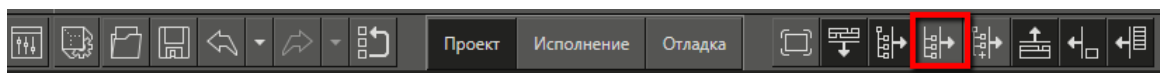
Итак, получился функционально законченный объект, содержащий программу управления, средства визуализации и имеющий один параметр для связи этого объекта с внешним миром. В нашем случае нам требуется при помощи этого параметра управлять дискретным выходом модуля. Для этого перетащим левой кнопкой мыши 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:

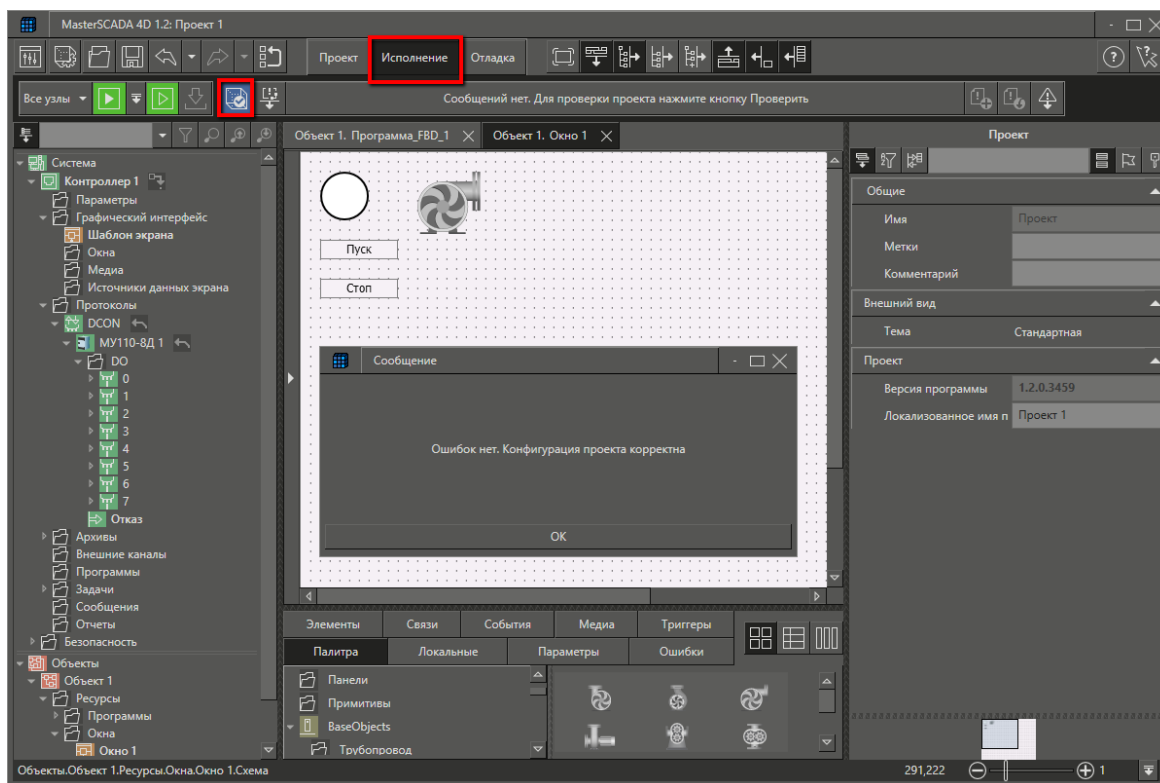
Если в какие-то моменты панель контекста не требуется в работе, то ее можно отключить в панели свойств, отжав соответствующую кнопку:



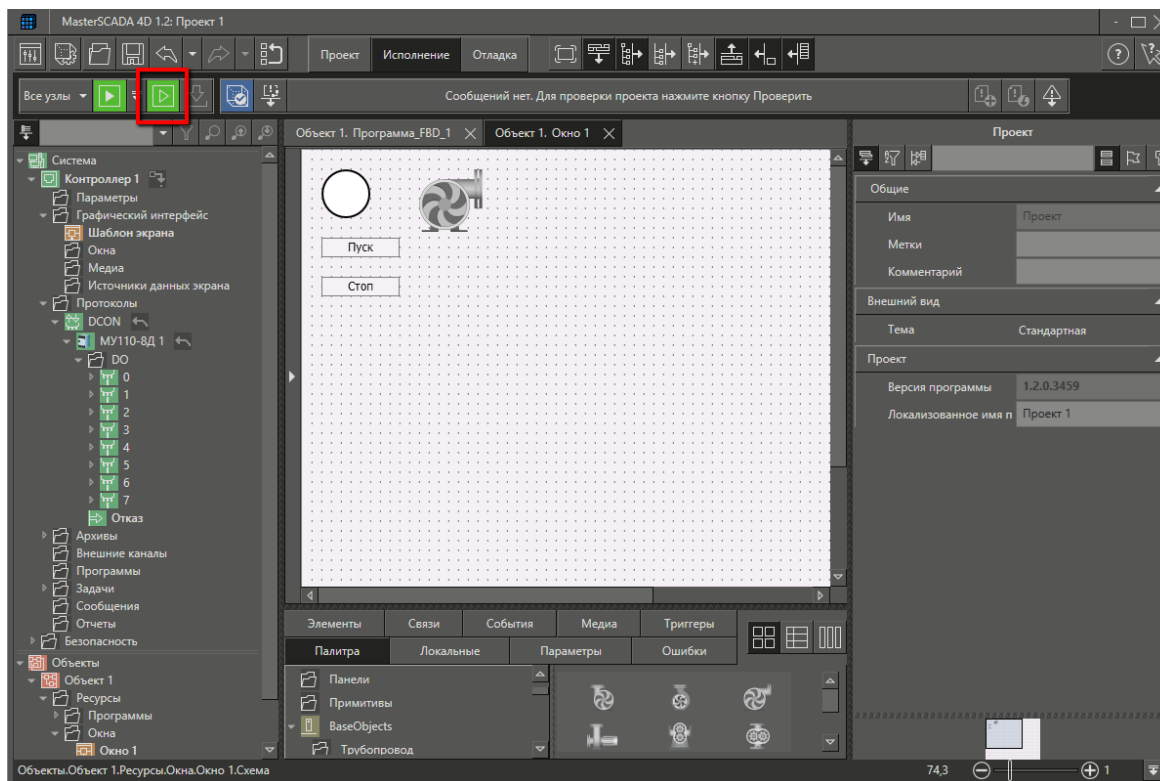
При необходимости, можно также уменьшить размер панели, перетащив ее правую границу к левой.

## 2.1.5.БС. Урок 1. Запуск

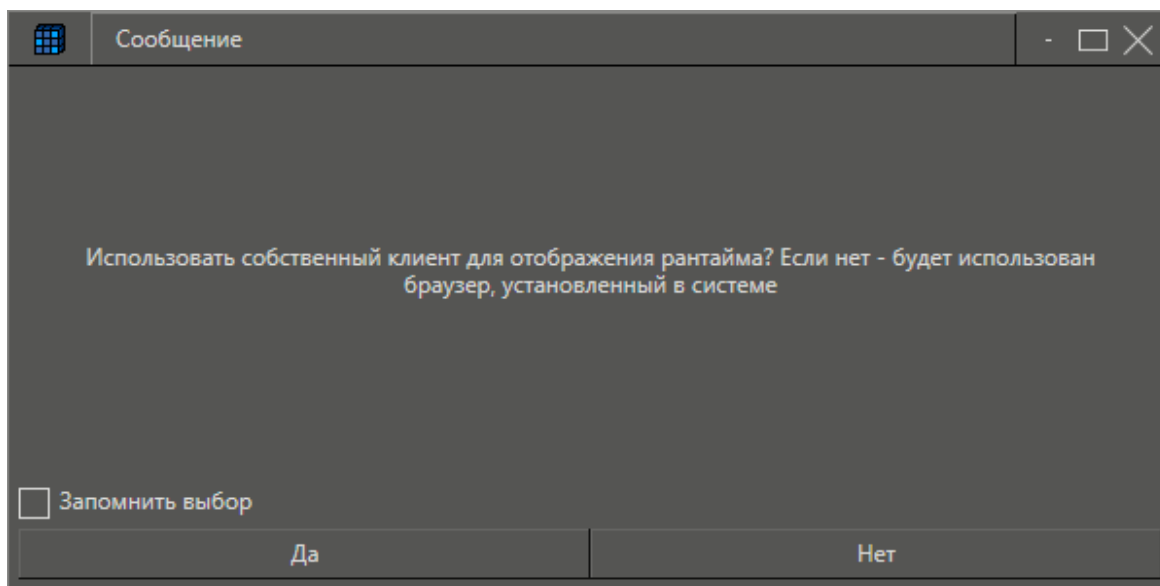
Сначала перейдём на вкладку Исполнение панели инструментов и нажмём кнопку Проверить:



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.

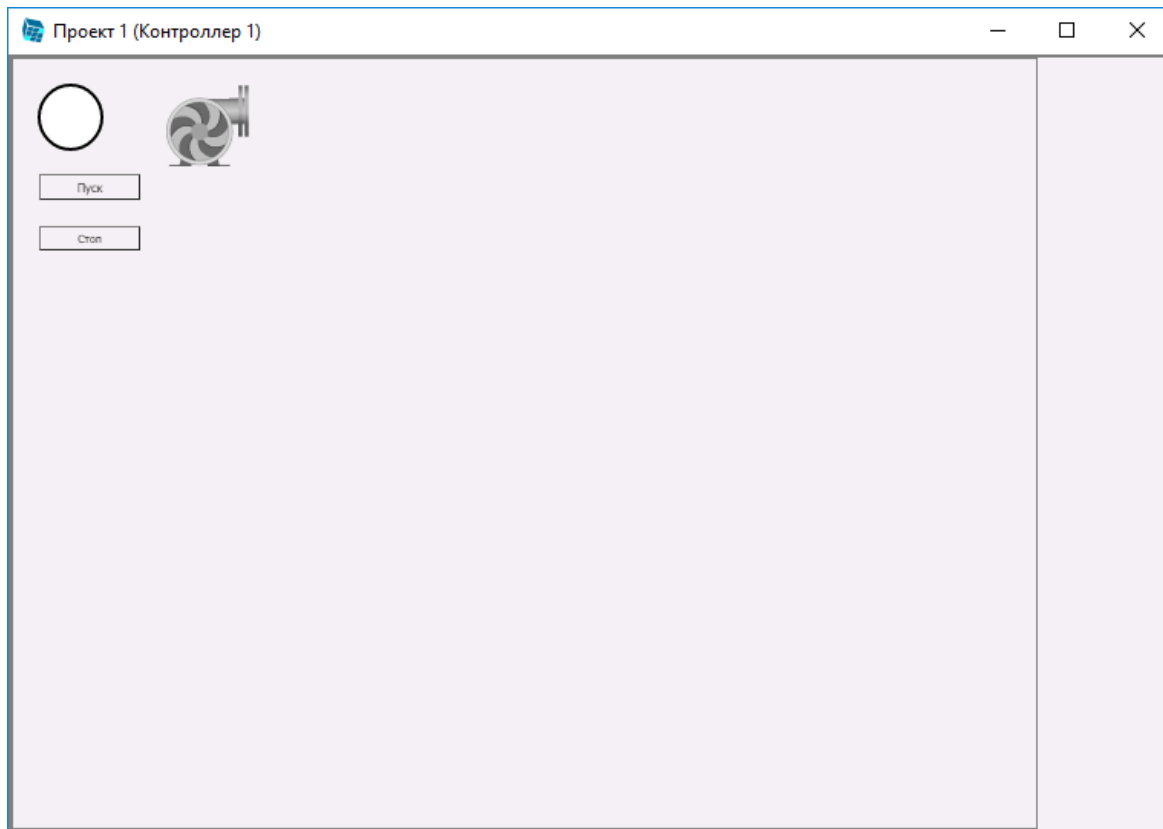


Если режим исполнения запускается впервые, то появится окно, предлагающее для отображения в режиме исполнения на выбор либо клиент визуализации, разработанный компанией "ИнСАТ", либо браузер, назначенный на компьютере по умолчанию.

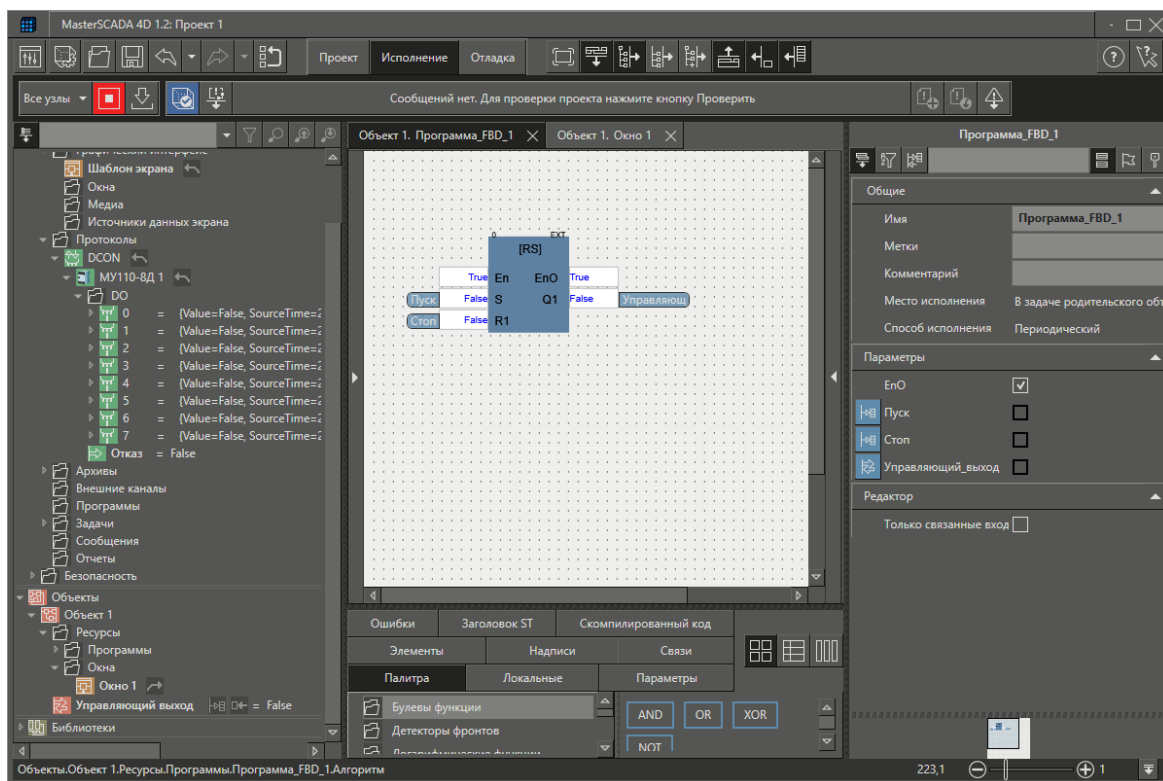


В нашем случае выбор может быть любым.

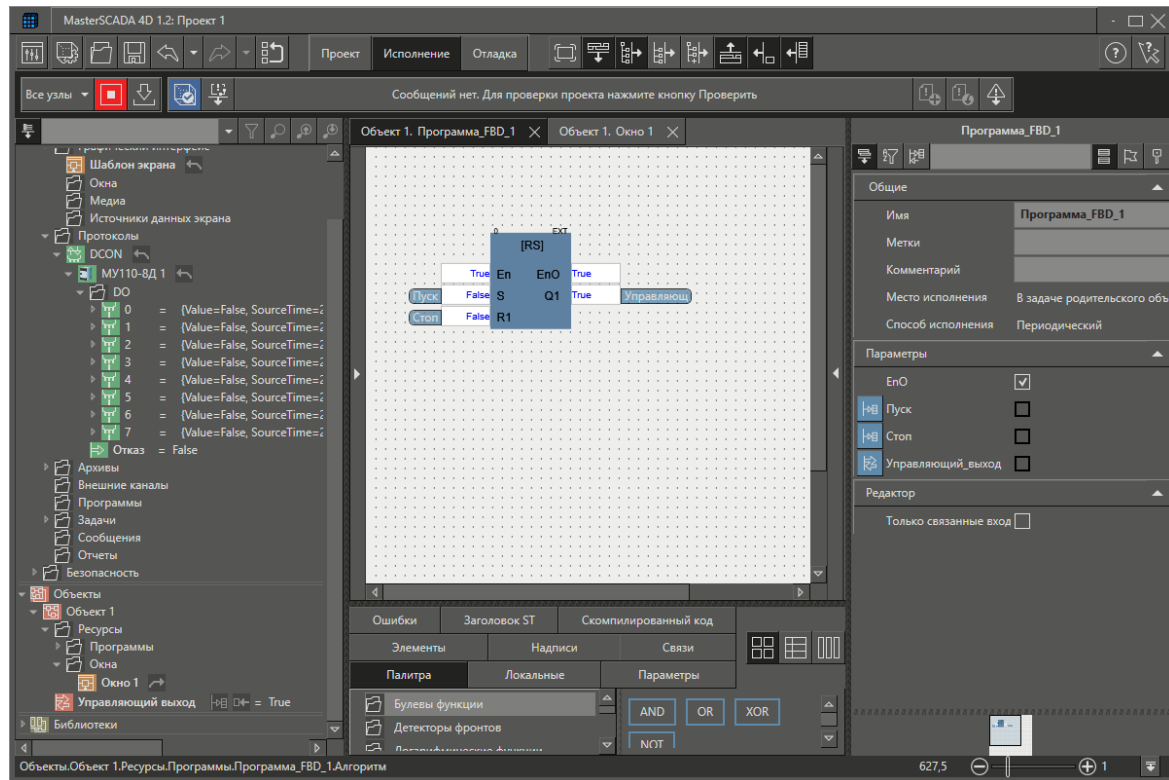
После успешного запуска проекта на исполнение, рядом с названием узла появится знак , и откроется встроенный клиент визуализации (или браузер) с созданной мнемосхемой.



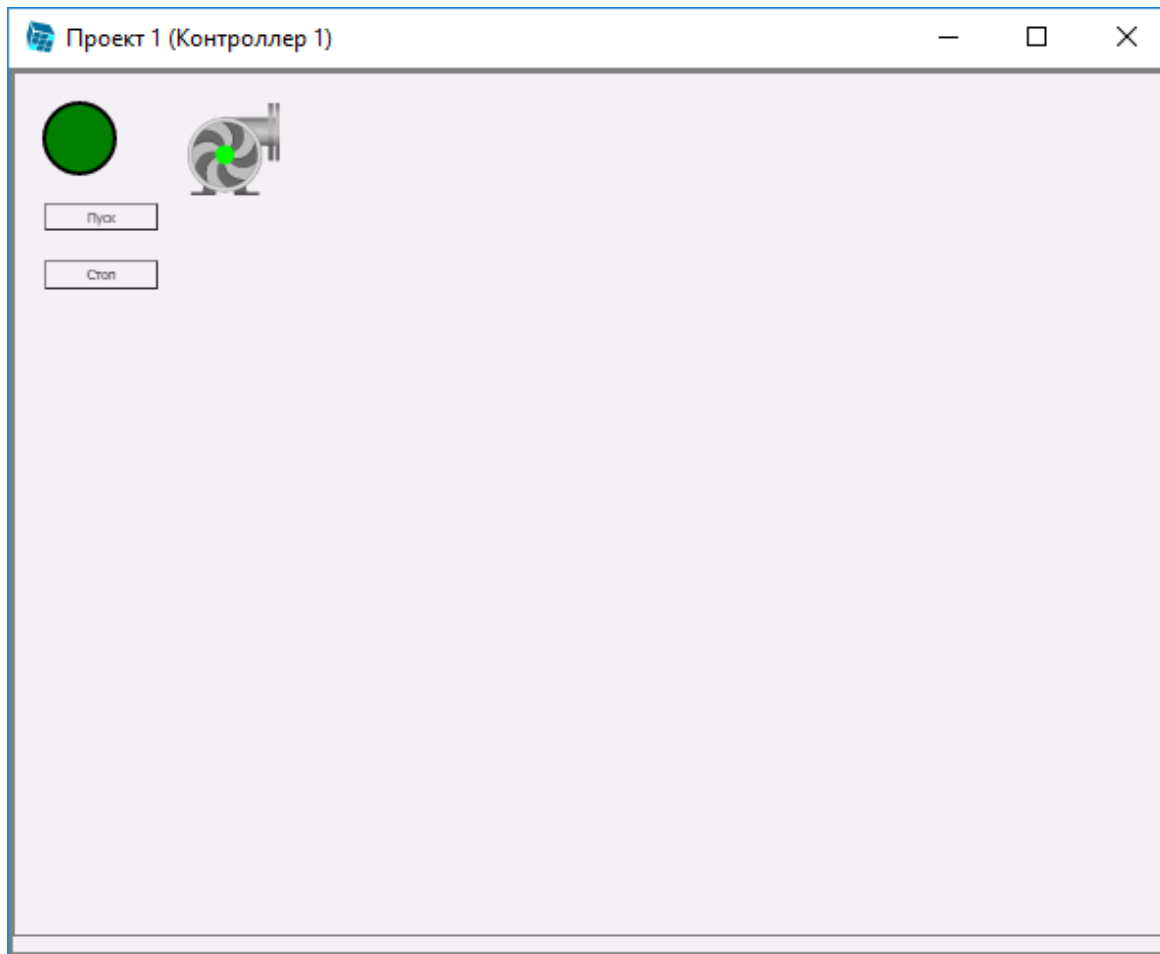
В среде разработки рядом с именами параметров появятся их текущие значения.



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.







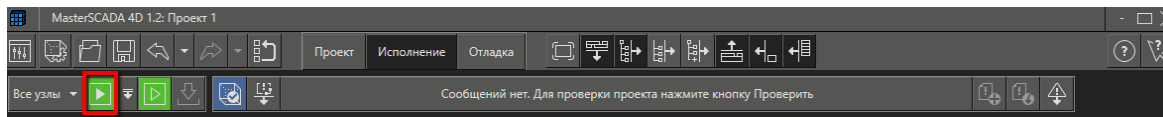
Нажмем кнопку Стоп. Управляющий выход примет значение False, индикатор на мнемосхеме окрасится в белый цвет, а насос остановится.

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.

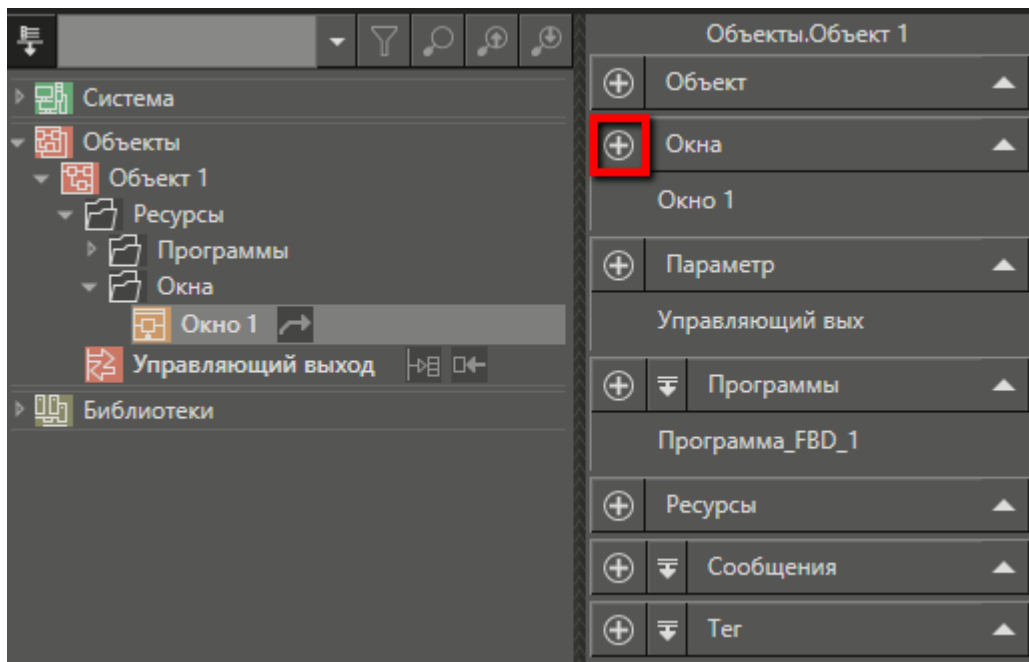
## 2.2. БС. Урок 2

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!

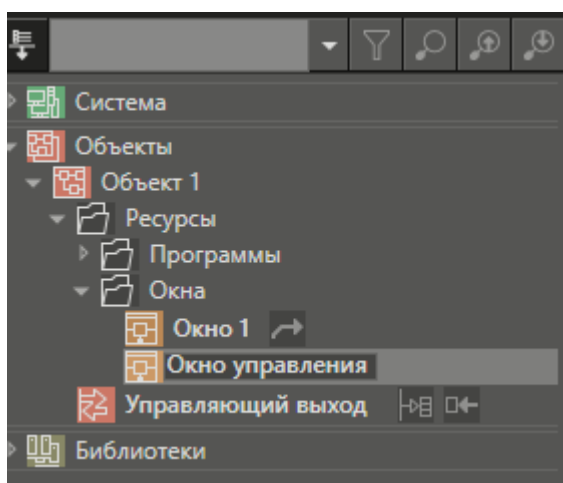
### 2.2.1.БС. Урок 2. Конфигурирование всплывающего окна

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

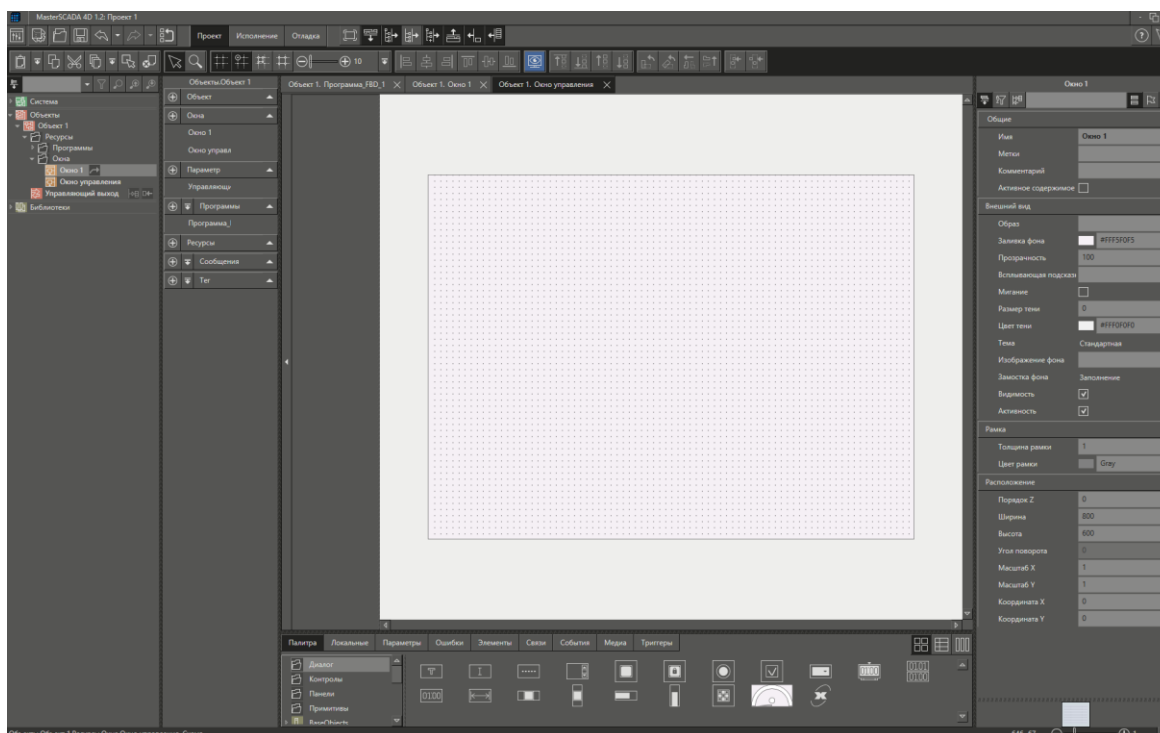


Пусть новый элемент называется Окно управления. Для изменения имени, присвоенного добавленному элементу по умолчанию, можно воспользоваться панелью

своих свойств, либо, дважды нажав мышью на элемент в дереве, ввести с помощью клавиатуры новое имя в появившееся поле:

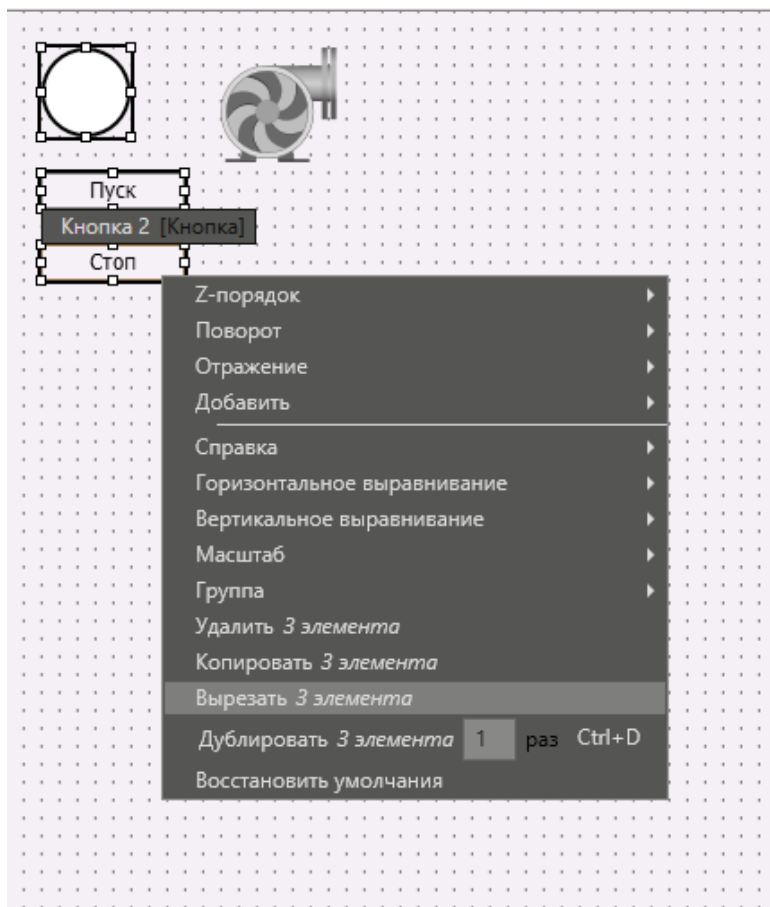
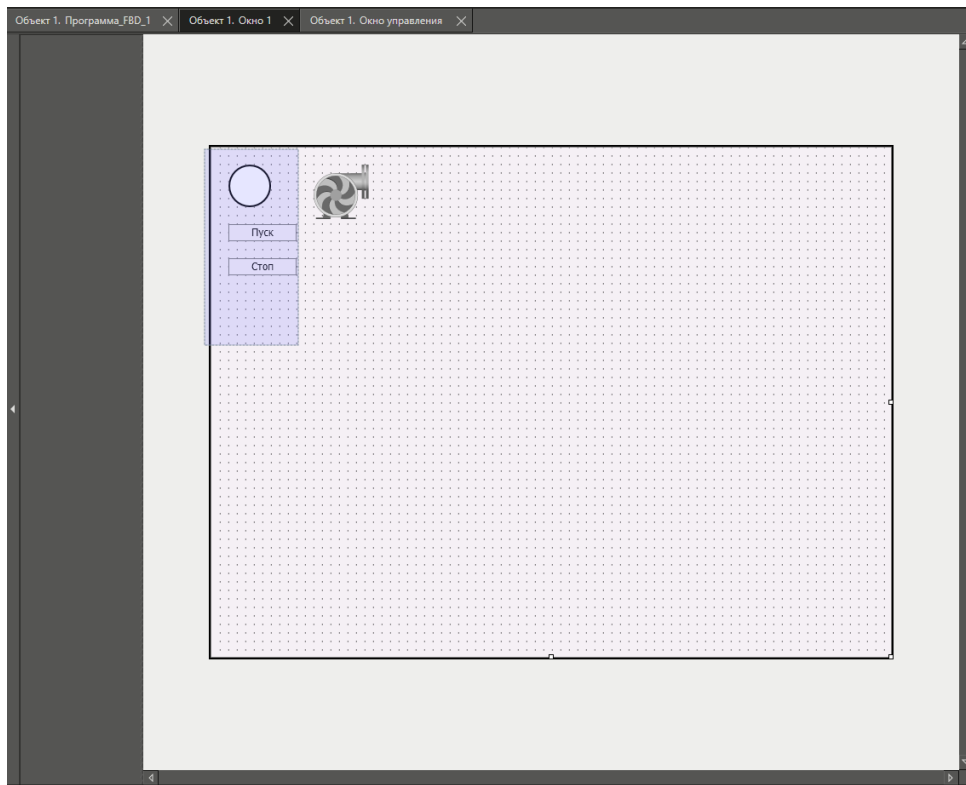


После создания окна автоматически откроется графический редактор:



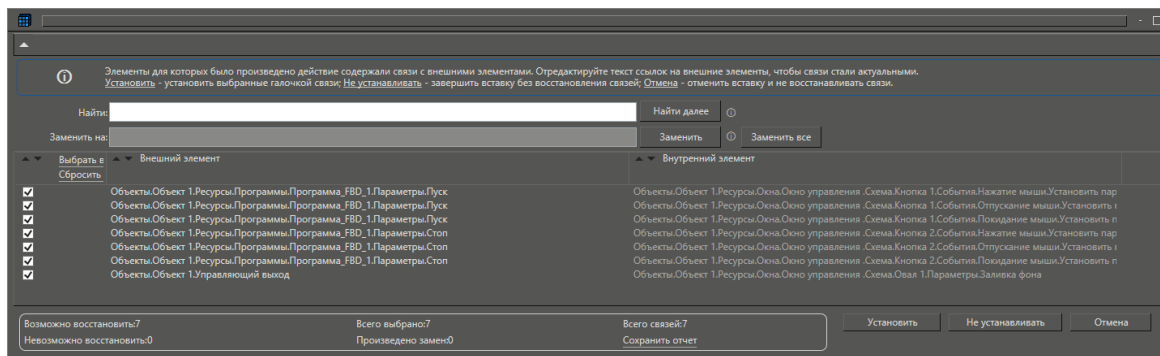
Перенесём в него элементы управления и индикации из основного окна.

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.

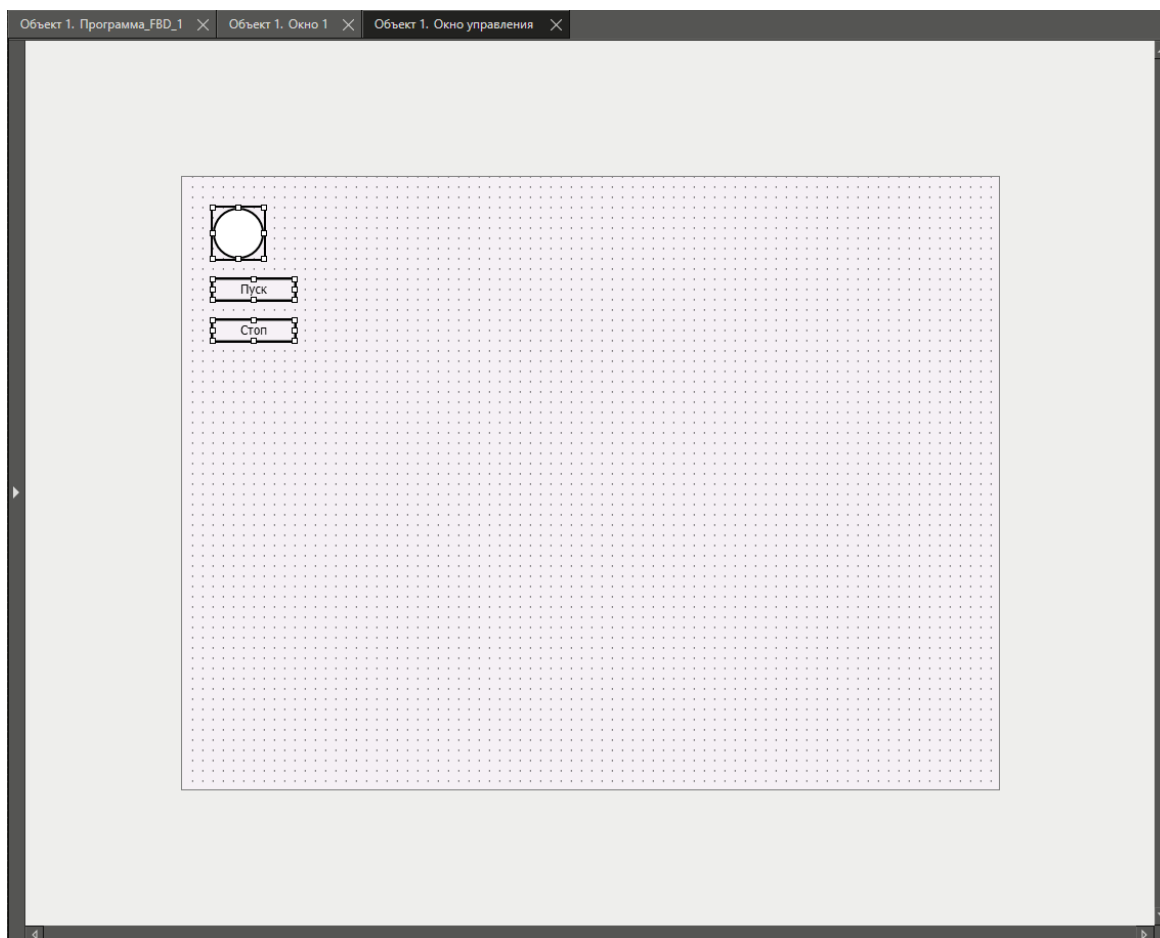


Теперь перейдём на вкладку Окно управления и, либо через контекстное меню Вставить, либо нажав комбинацию клавиш Ctrl+V, вставим туда вырезанные элементы.

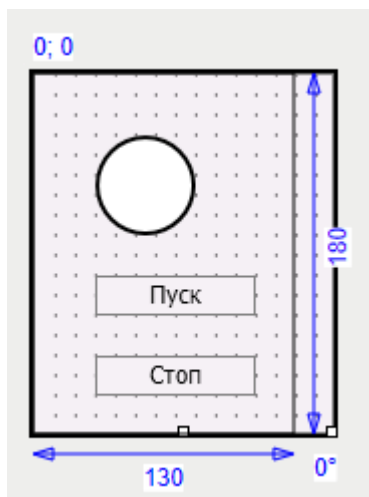
При этом появится диалоговое окно восстановления связей:



Нажмем кнопку Установить и получим результат:

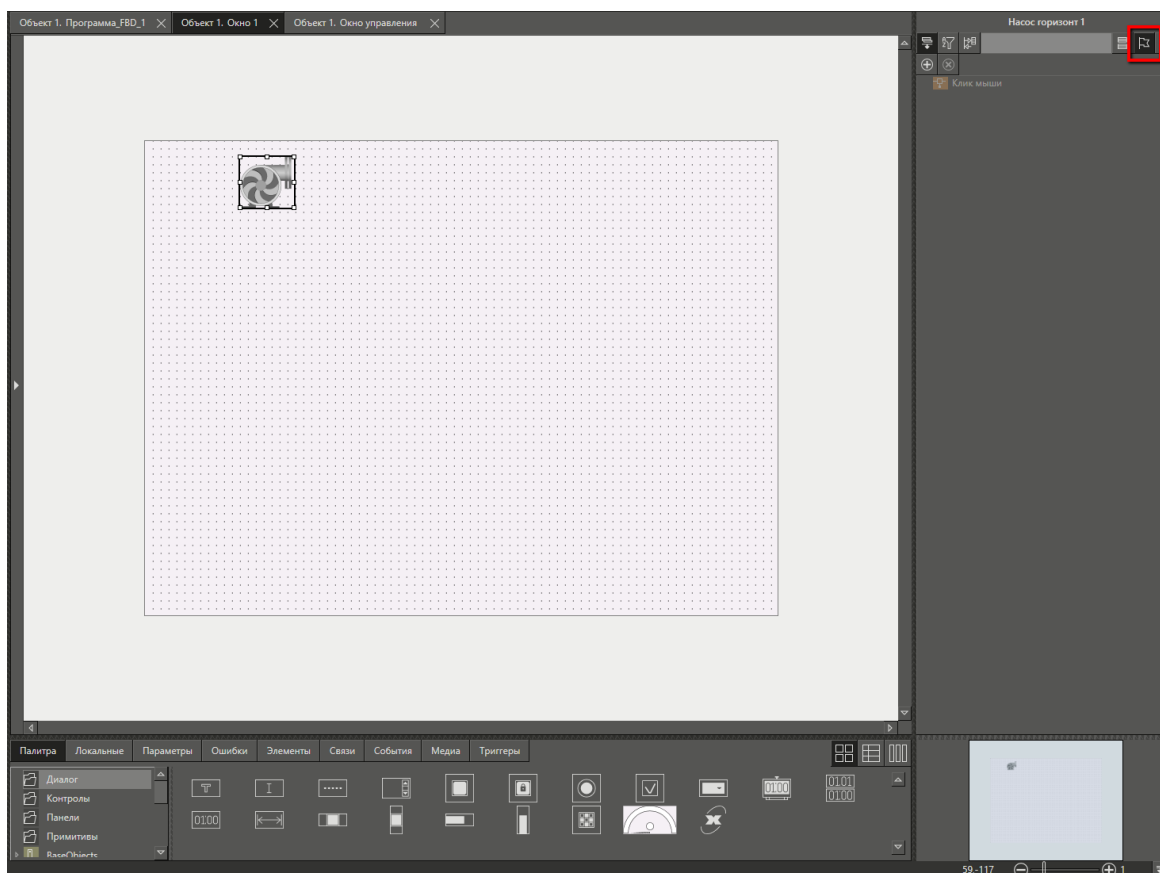


Затем уменьшим размер окна, потянув влево правую границу окна и вверх - нижнюю:

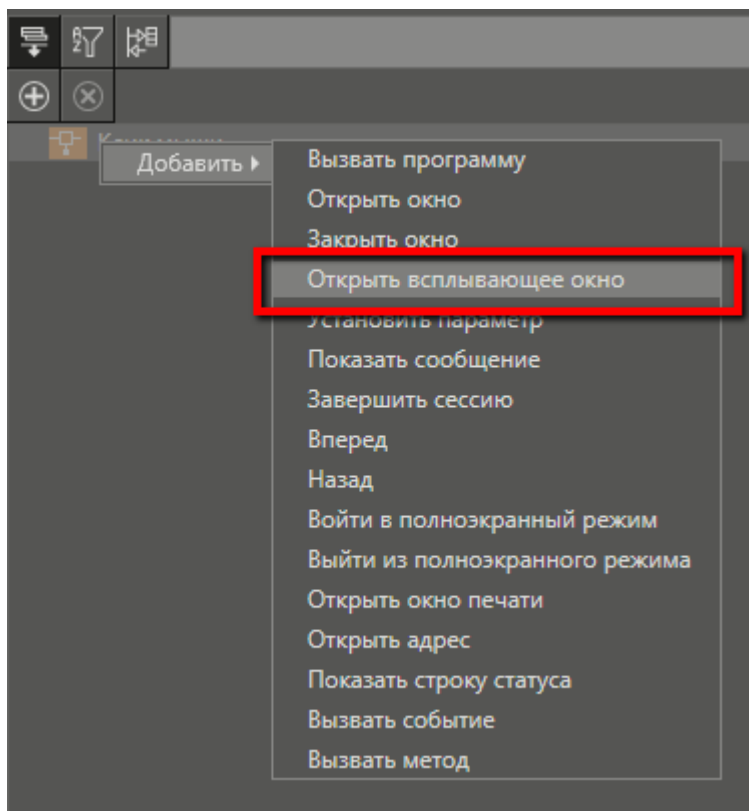


Теперь нам предстоит сделать так, чтобы это окно открывалось по клику мыши на изображении насоса.

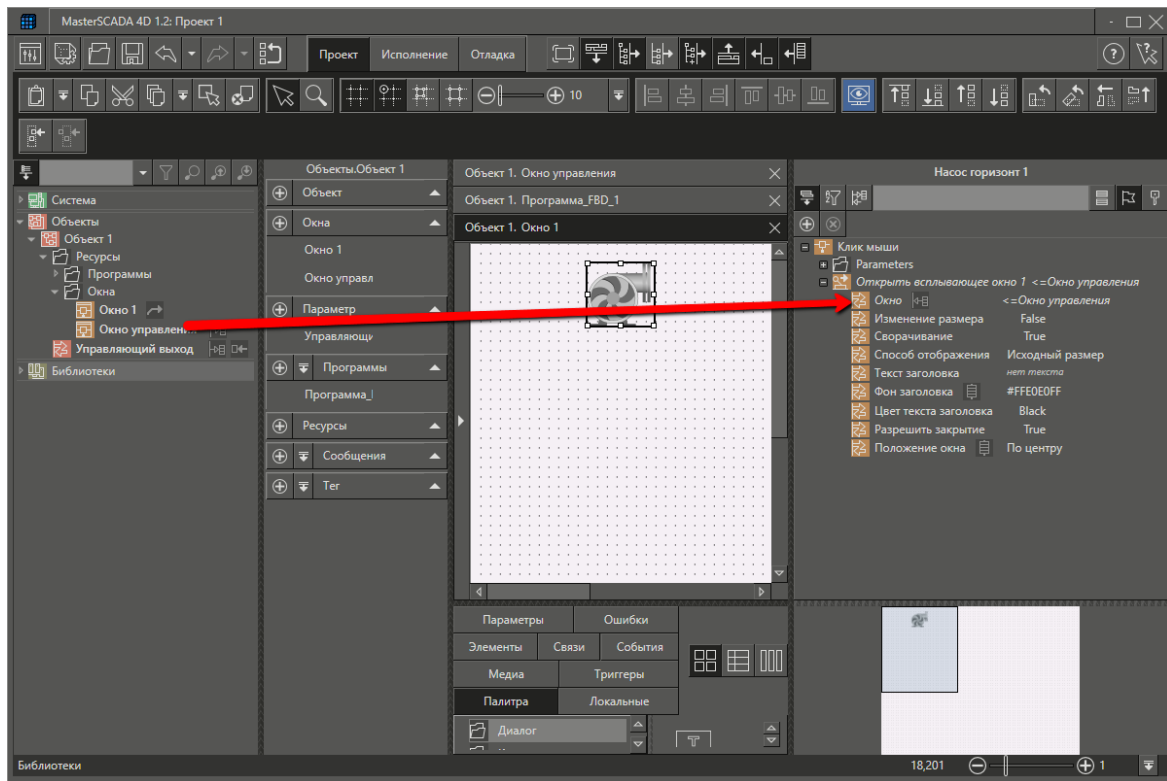
Для этого перейдём на вкладку Окно 1, которое является основным. Выделим в окне изображение насоса, нажав на него левой кнопкой мыши. Панель свойств переведём в режим отображения событий:



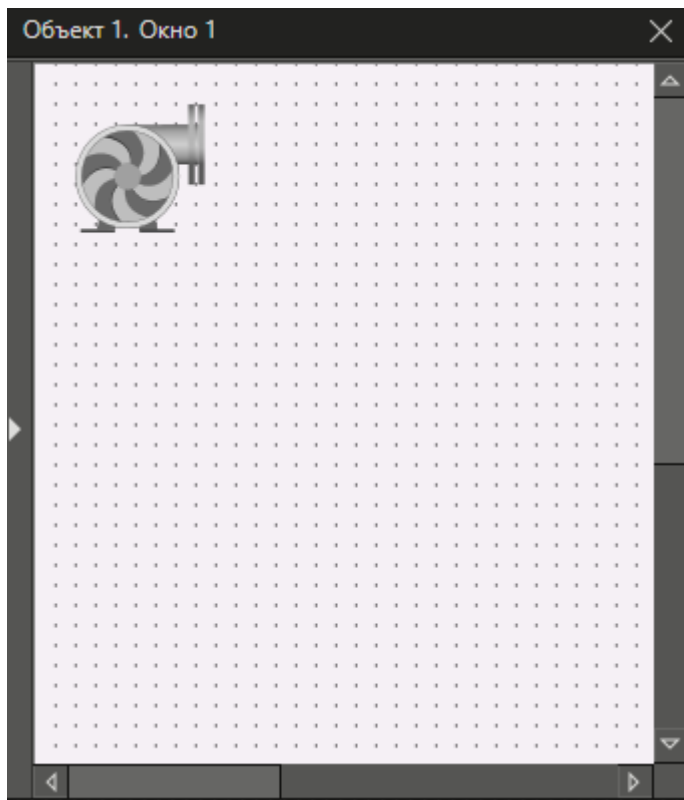
Нажмем правой кнопкой мыши на событие Клик мыши. Из контекстного меню выберем Добавить, а из появившегося списка доступных действий – Открыть всплывающее окно:



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:

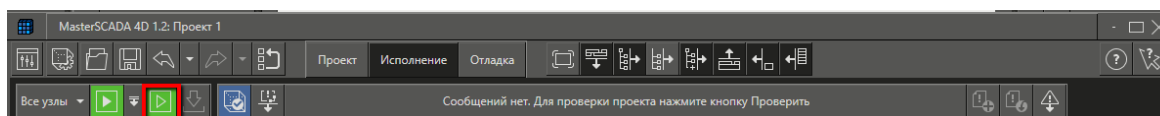


На этом конфигурирование нашего простого задания закончено.

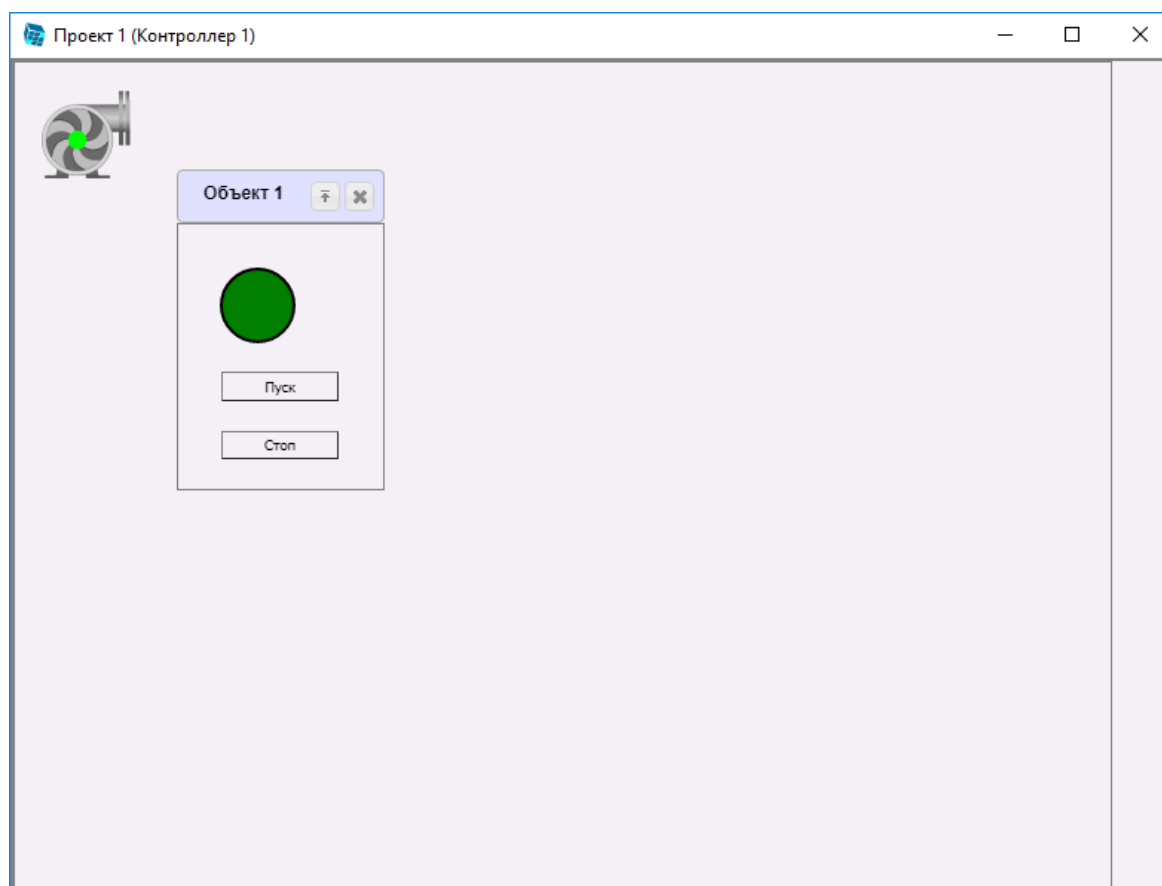


## 2.2.2.БС. Урок 2. Запуск

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.



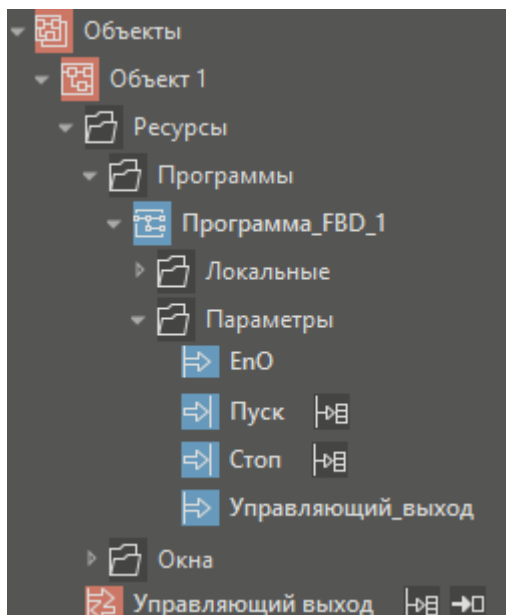
Нажмем кнопку Стоп – насос остановится. Всё, как и в уроке 1, за исключением того, что органы управления вынесены в отдельное окно.

## 2.3. БС. Урок 3

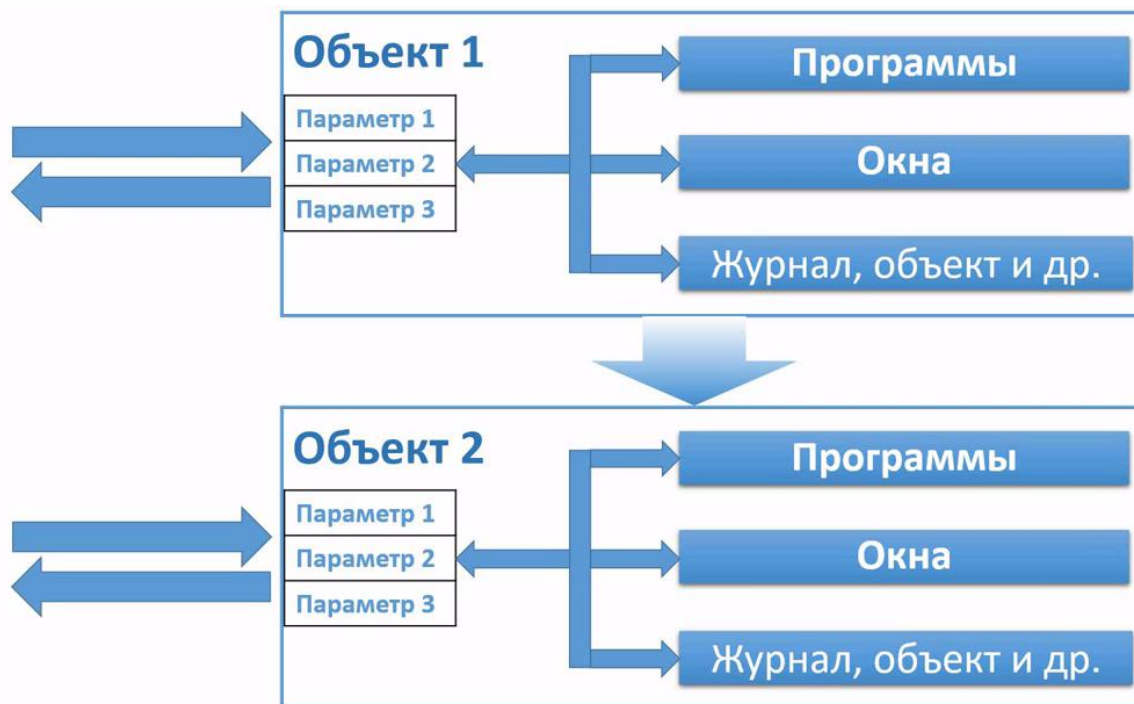
В предыдущих уроках мы осваивали объектный подход при создании проекта, но не объясняли, для чего это нужно. Теперь настало время остановиться на объектной идеологии подробнее.

### 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 создаётся с сохранением всей внутренней структуры, включая межэлементные связи внутри объекта. Разработчику останется лишь подключить внешние.

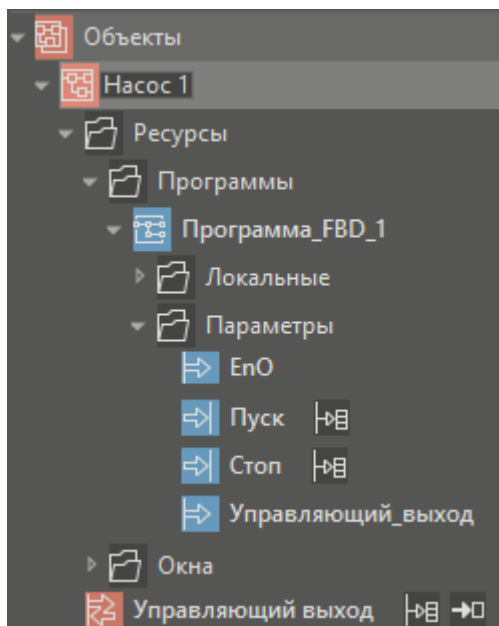


Теперь посмотрим как это можно применить на практике.

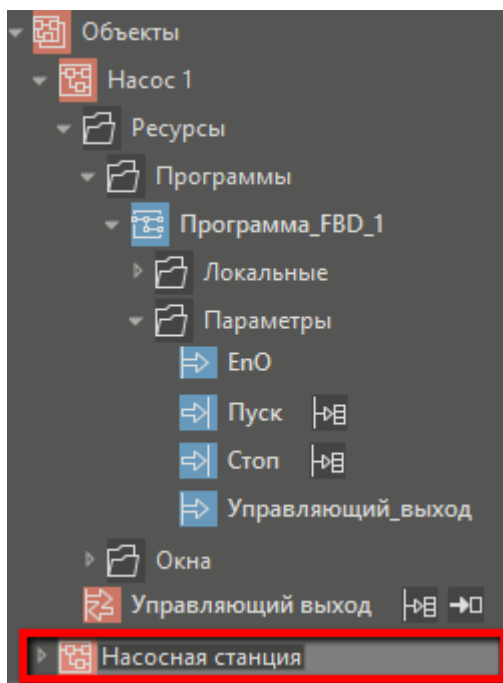
В предыдущем уроке мы создали объект, состоящий из окна с изображением насоса, окна управления этим насосом, вызываемого по нажатию мышью на изображении насоса, и программы, реализующей логику управления насосом. Объект имеет один параметр для связи с внешним миром – это управляющий выход, сигнал с которого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

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

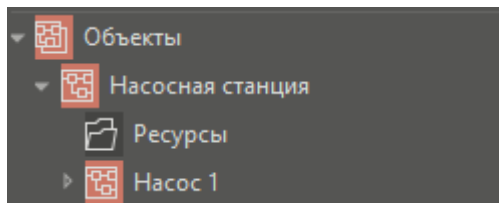
Переименуем созданный ранее объект в Насос 1.



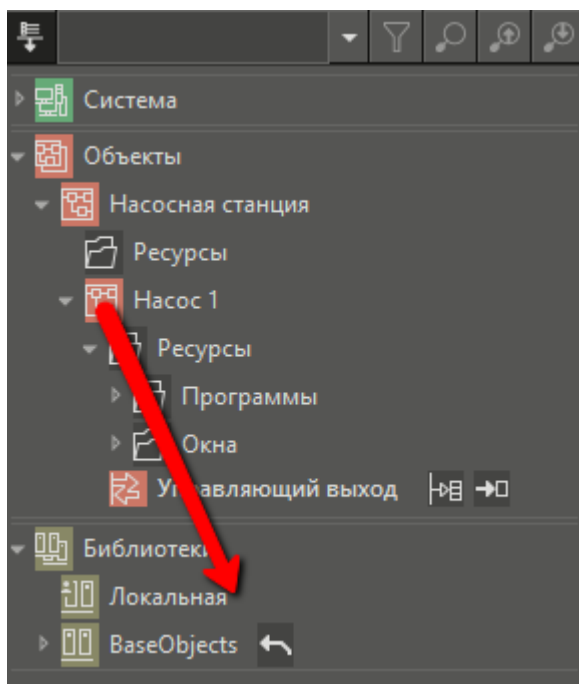
В дереве объектов создадим ещё один объект и назовём его Насосная станция.



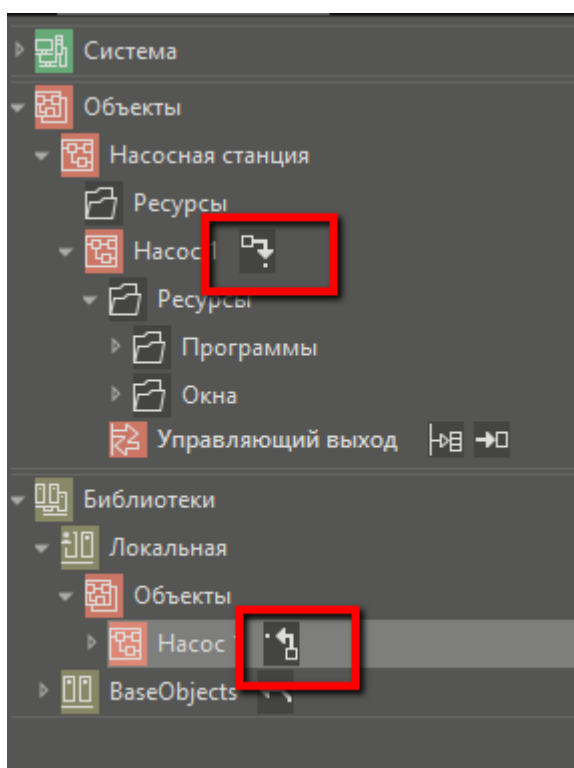
Теперь перетащим левой кнопкой мыши насос в насосную станцию. У нас получился корневой объект Насосная станция, у которого есть вложенный объект Насос 1.



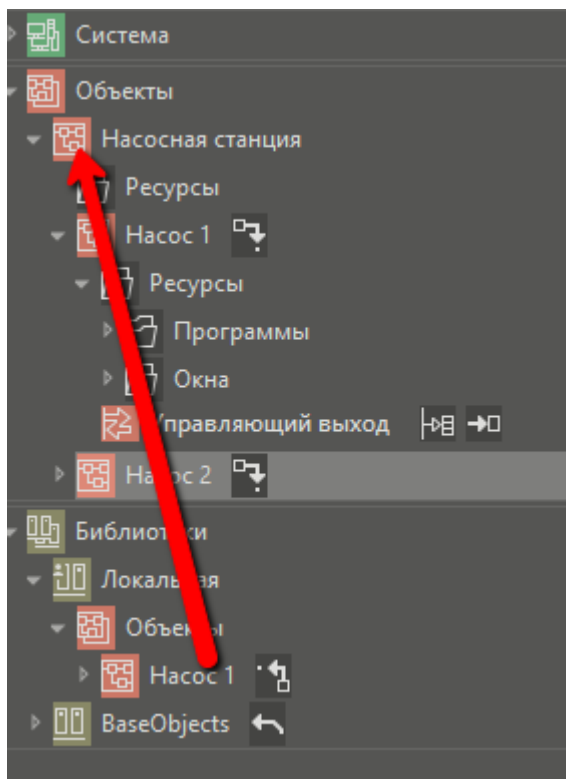
Перенесём его в библиотеку Локальная.



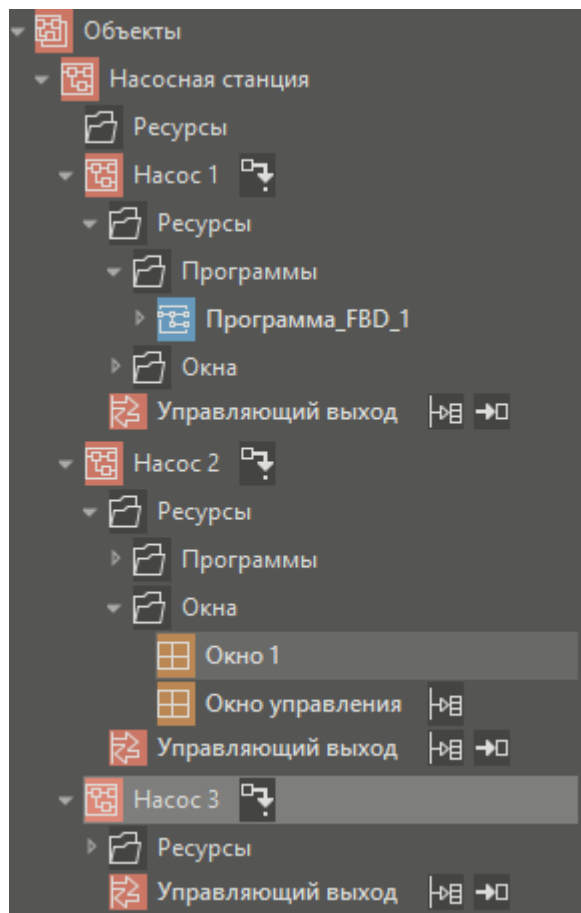
Теперь насос стал библиотечным элементом, а насос в дереве объектов – его наследником, о чём нам говорит специальный значок рядом с именем и всплывающая подсказка.



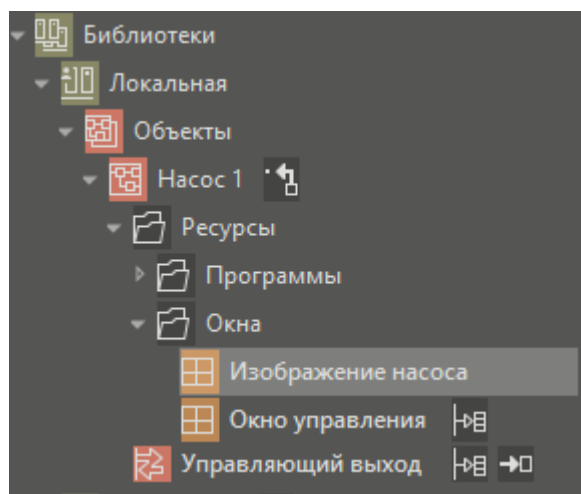
Перетащим библиотечный насос в насосную станцию. Получаем два насоса.



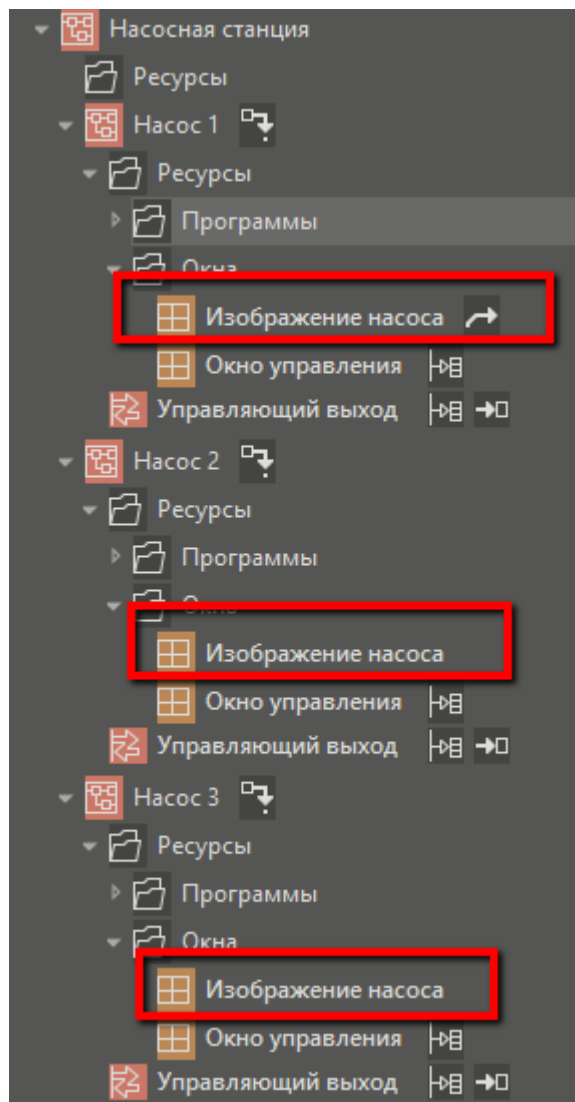
Повторим эту процедуру ещё раз – теперь насосная станция включает уже три насоса, каждый из которых имеет один параметр для связи с внешними элементами или устройствами (Управляющий выход), а внутри содержит окна и программу управления.



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:



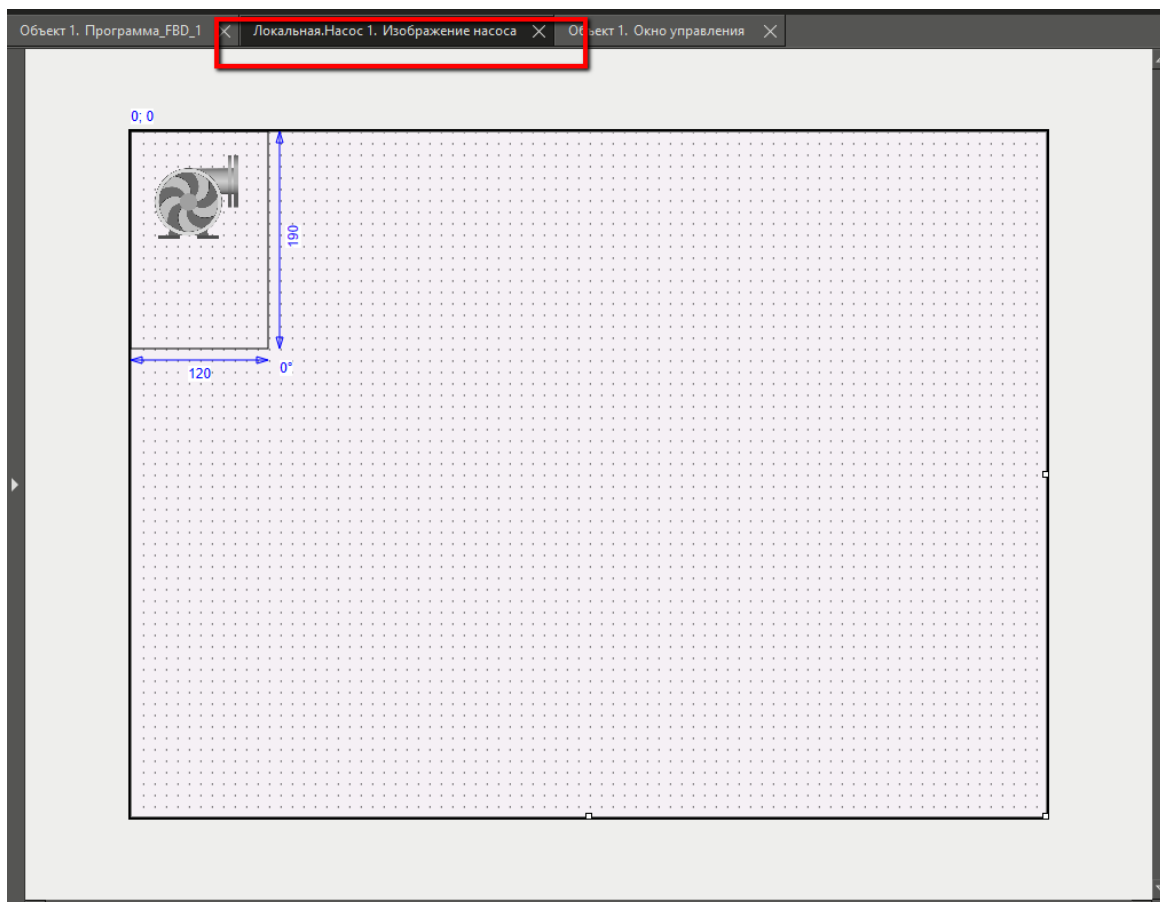
Увидим, что в дереве объектов это окно тоже переименовалось:



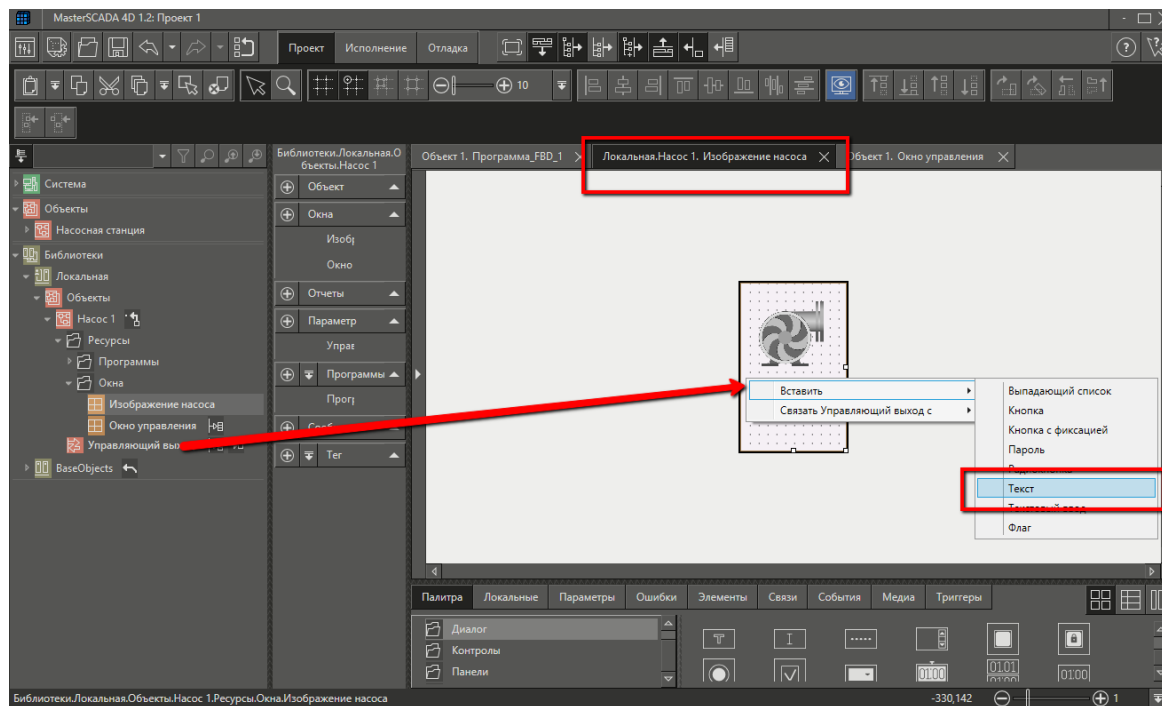
То есть изменение унаследовалось. Строго говоря, изменения будут наследоваться не всегда, а только до тех пор, пока не будет отредактирован сам наследник, и он не станет отличаться от библиотечного элемента. После этого наследование изменений будет сохраняться только для части наследника, совпадающей 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):

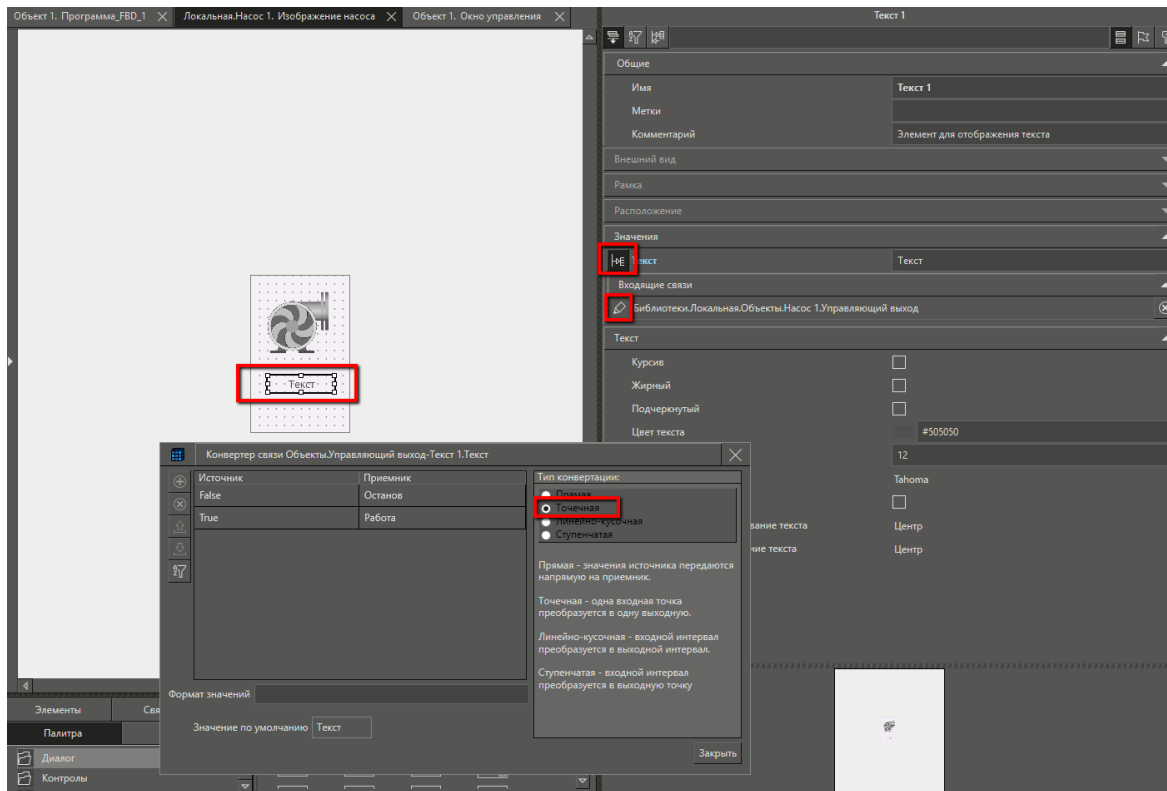




Добавим надпись, отражающую состояние насоса. Для этого перетащим правой кнопкой мыши параметр Управляющий выход из библиотечного насоса (мы сейчас работаем с библиотечным элементом), и из открывшегося контекстного меню выберем Вставить.Текст:

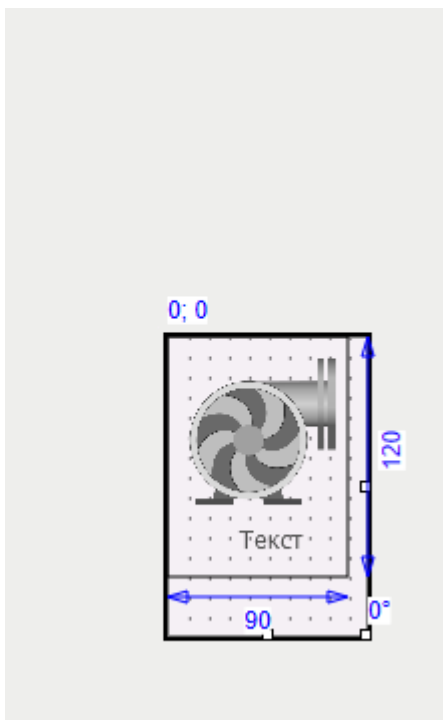


Очень часто рядом с изображением объекта отображают какой-нибудь основной параметр его работы, а иногда и несколько. Для клапана – это его положение; для двигателя с частотным преобразователем – частота, и так далее. Мы выведем просто состояние. Перейдем на панель свойств элемента Текст, откроем конвертор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

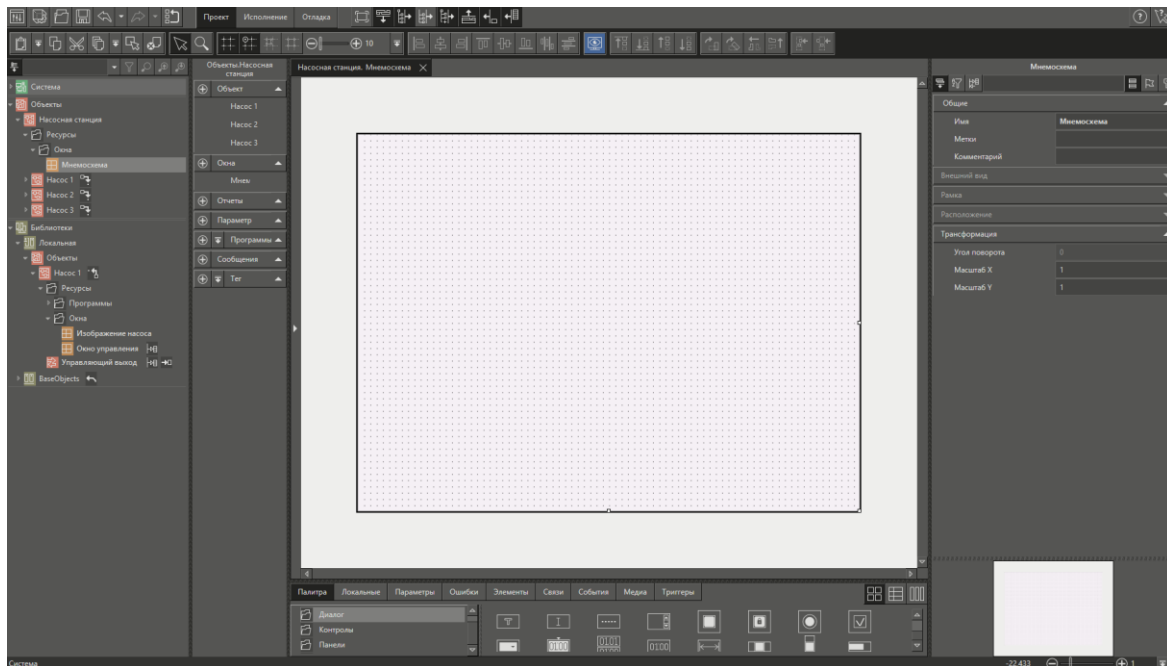


Закроем конвертор.

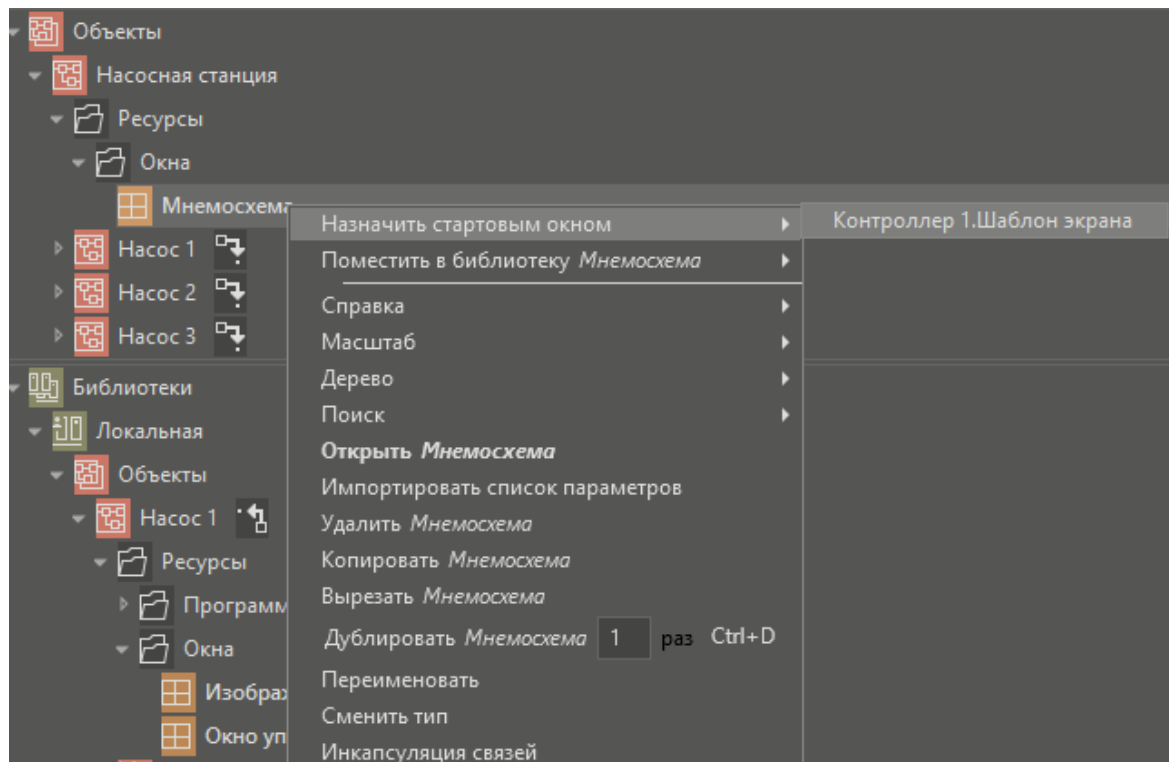
Уменьшим размер окна до размеров картиннки:



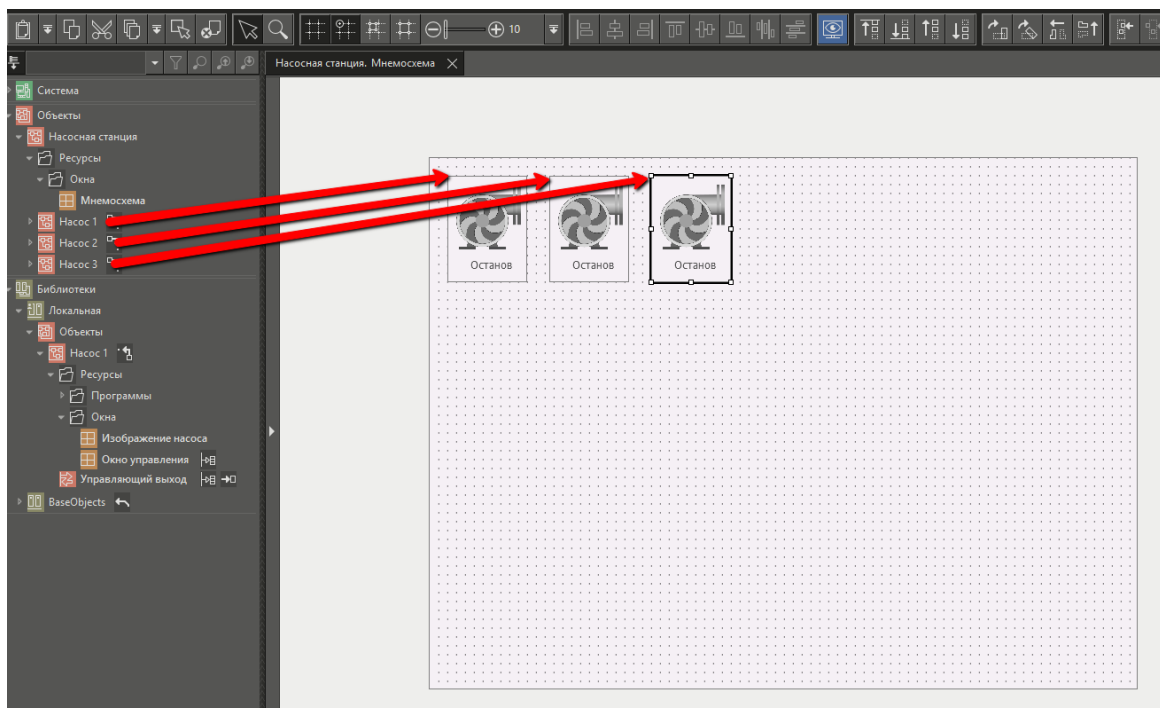
Добавим в насосной станции окно и назовём его Мнемосхема:



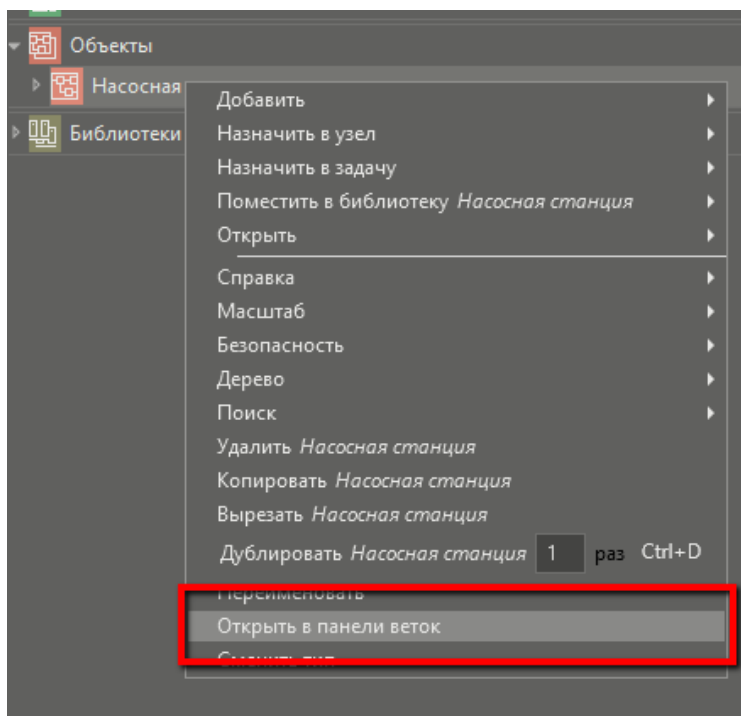
Назначим Мнемосхему стартовым окном узла при помощи контекстного меню:

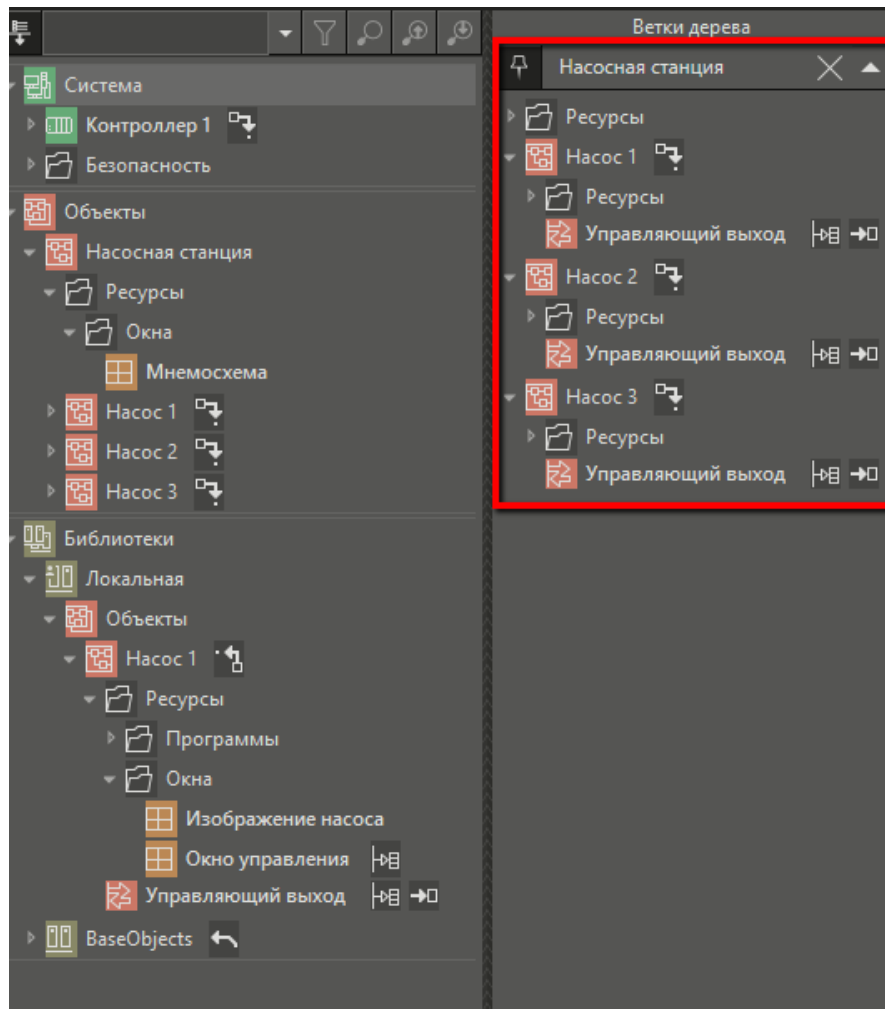


Теперь перетащим поочередно все насосы в окно:

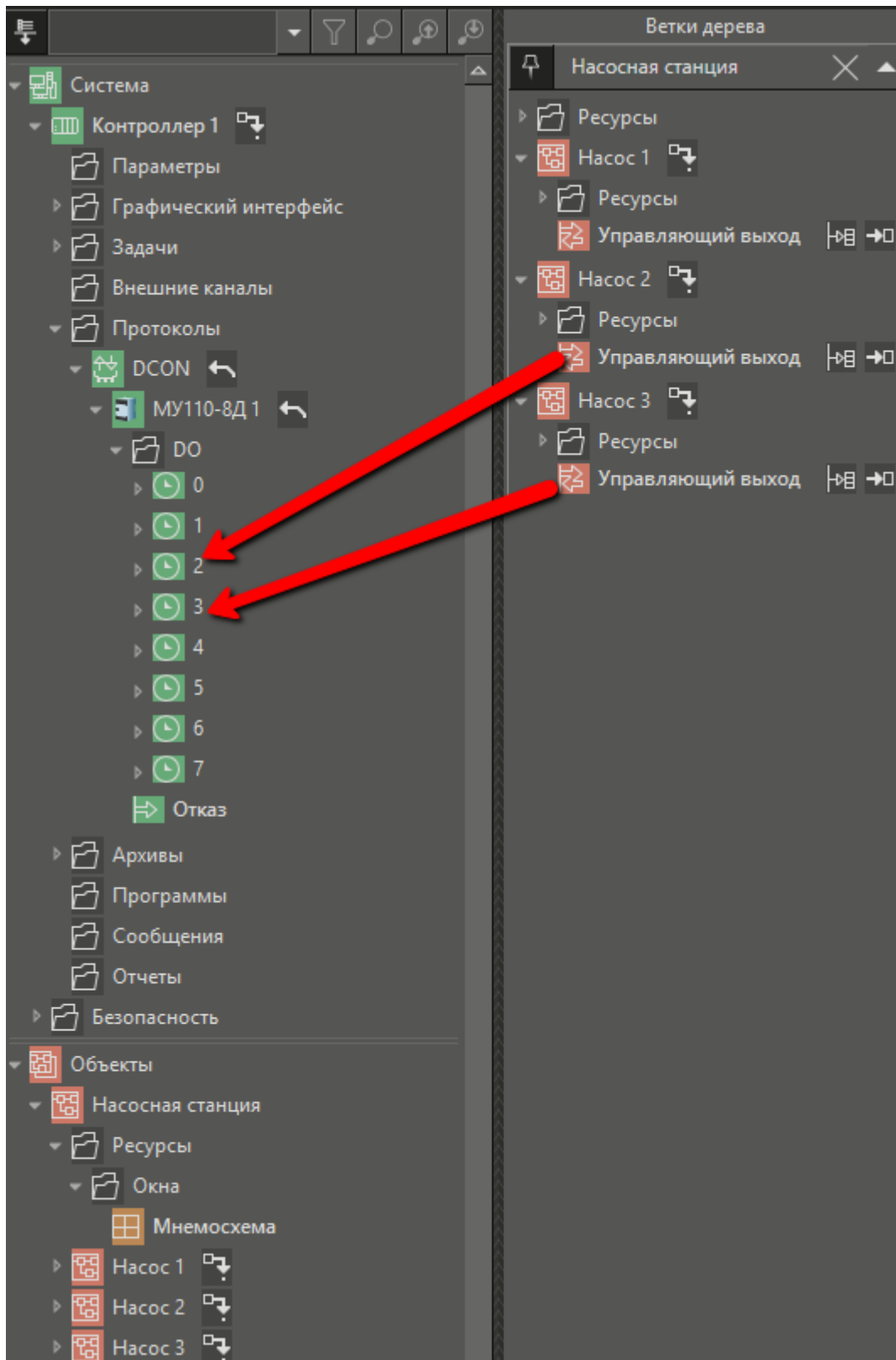


Свяжем выходы новых насосов с физическими выходами модулей ввода/вывода (мы подключили их в Уроке 1). Связывание удобнее производить, открыв деревья связываемых элементов в параллельных окнах. Нажмем правой кнопкой мыши на объект Насосная станция, и из контекстного меню выберем Открыть в панели веток:





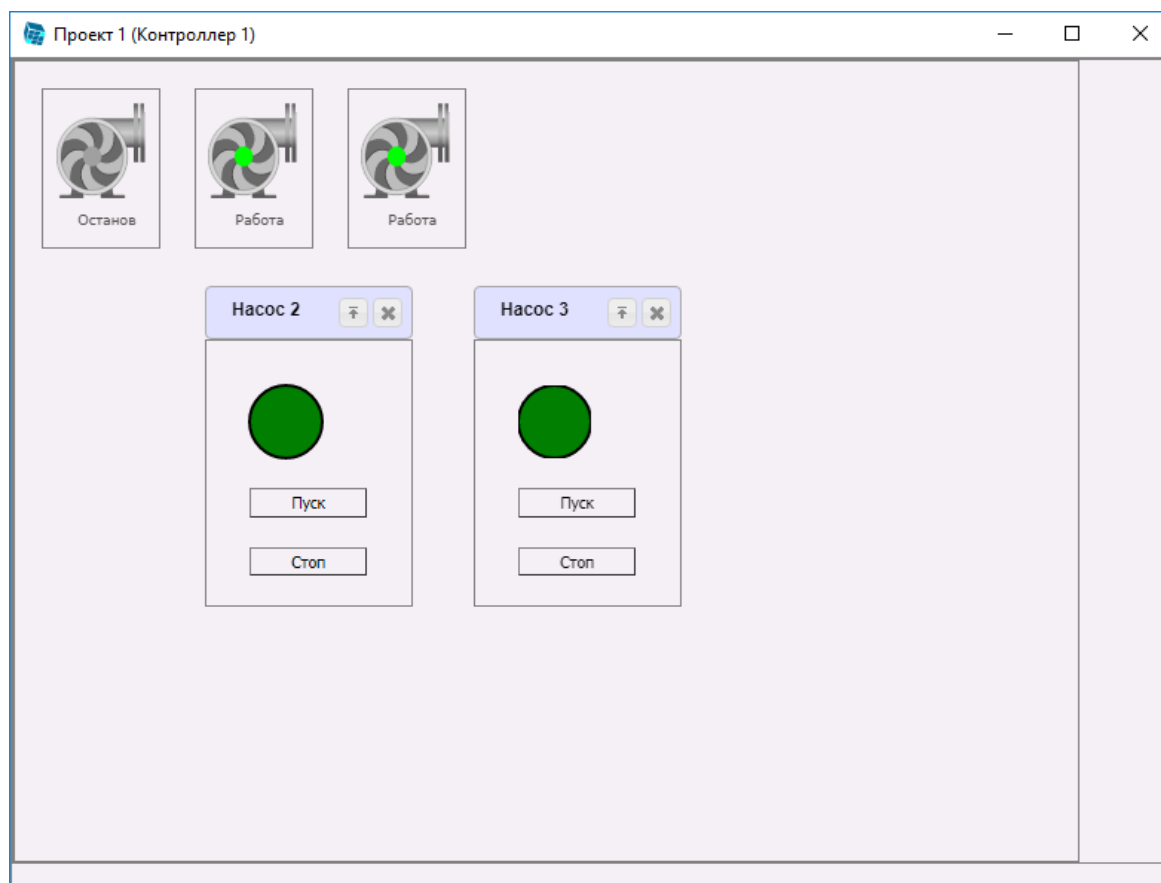
Первый насос у нас уже связан. Теперь свяжем параметр Управляющий выход второго и третьего насоса с последующими дискретными выходами модуля:



На этом наша насосная станция готова!

### 2.3.2.БС. Урок 3. Запуск

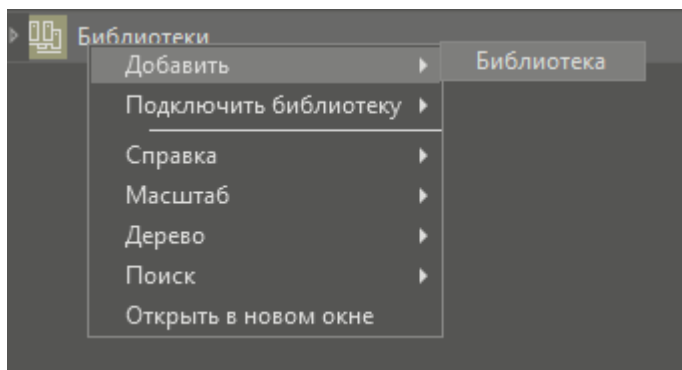
Запустим режим исполнения. В браузере мы увидим созданную нами насосную станцию с тремя насосами, каждый из которых управляется независимо от других. Насосы можно включать или выключать в любой последовательности. Можно также закрывать ненужные окна, а затем снова их открывать и т.п.



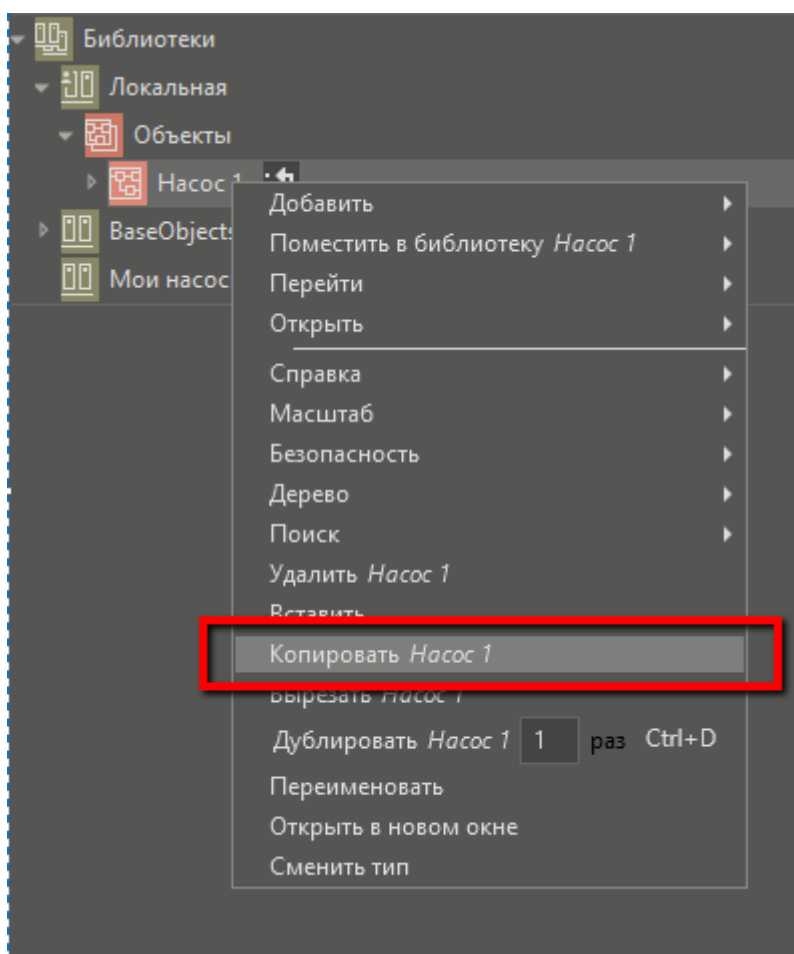
### 2.3.3.БС. Урок 3. Операции с библиотеками

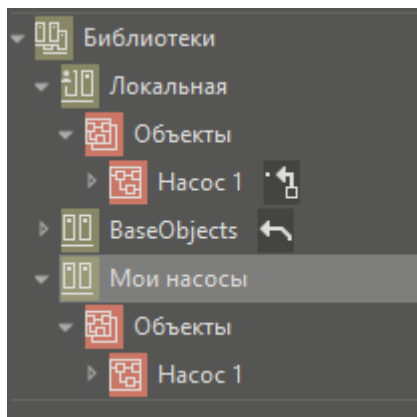
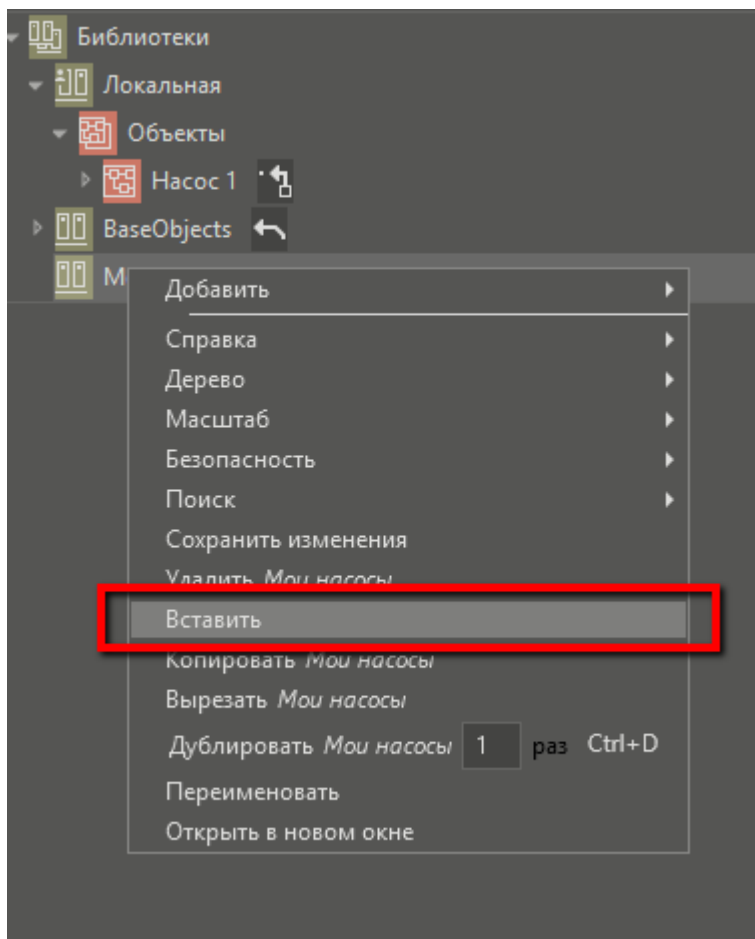
В рамках создания этого проекта мы поместили тиражируемый элемент Насос в библиотеку Локальная. Библиотека Локальная доступна только в том проекте, в котором она была создана. Если же требуется, чтобы библиотечный элемент был доступен для использования и в других проектах, то необходимо создать его в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



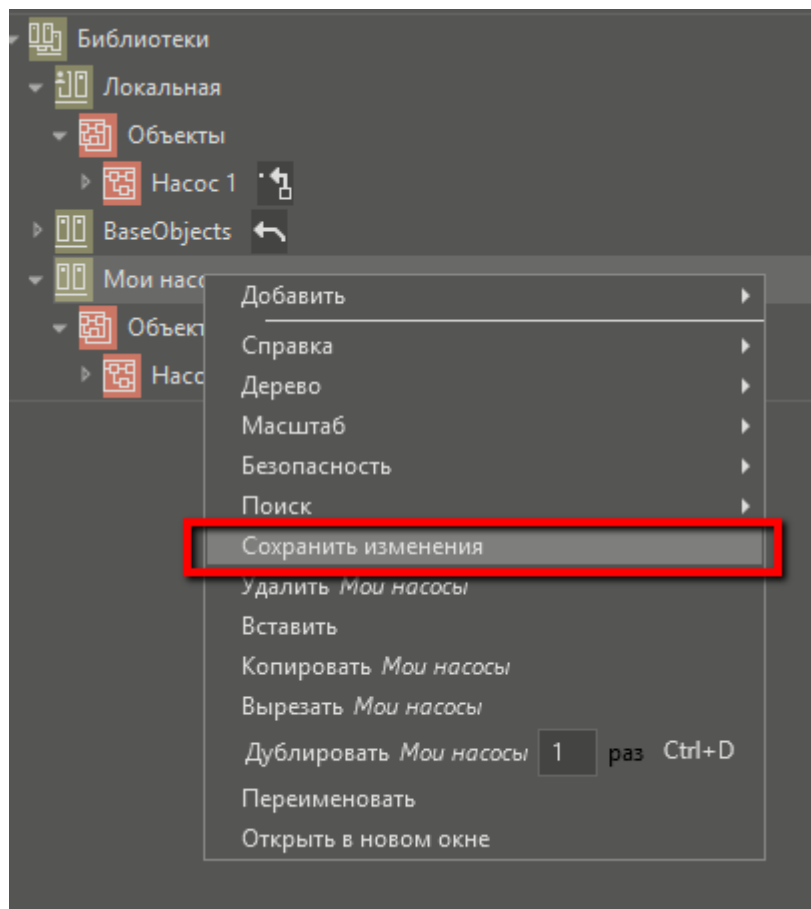


Назовём эту библиотеку Мои насосы. Скопируем созданный нами библиотечный насос из библиотеки Локальная и вставим его в библиотеку Мои насосы:

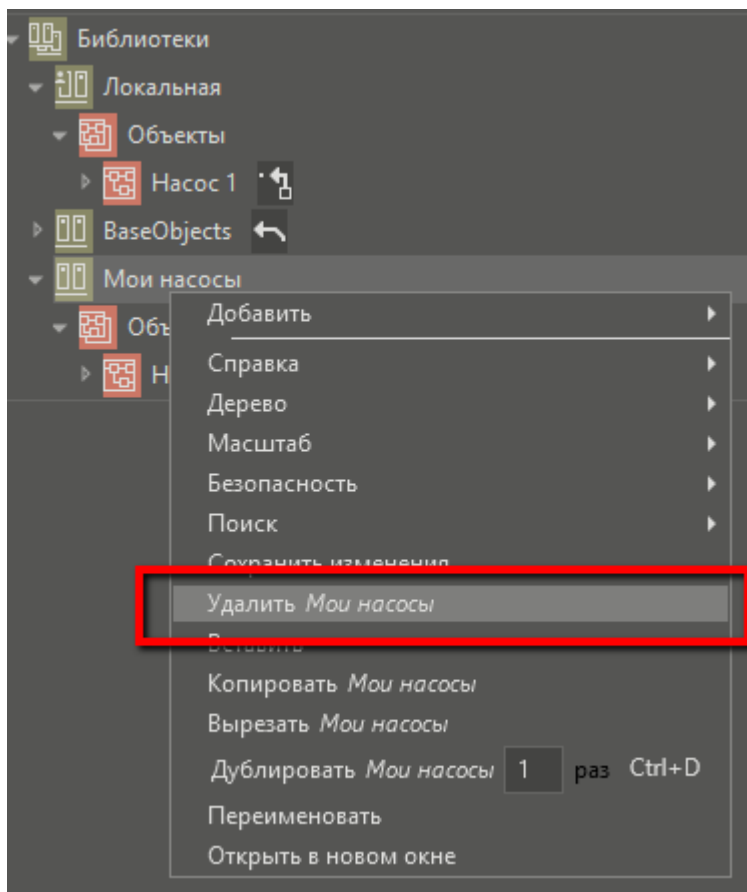




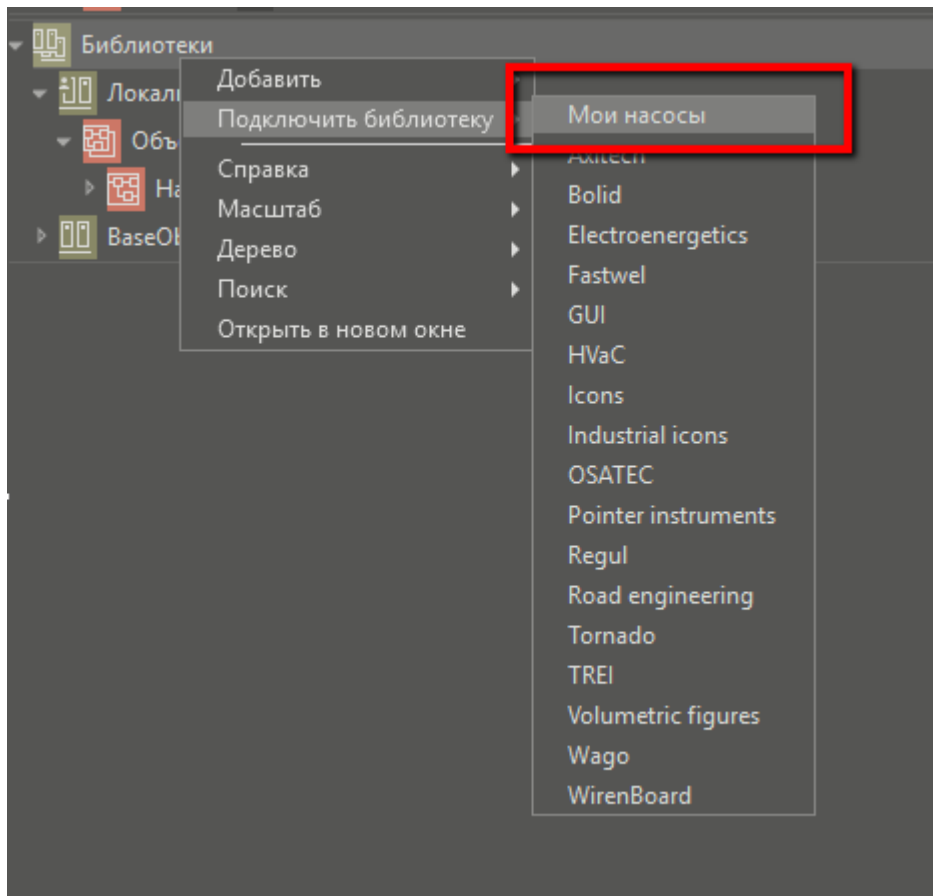
Затем нужно нажать правой кнопкой мыши на название библиотеки и выбрать пункт контекстного меню Сохранить изменения:



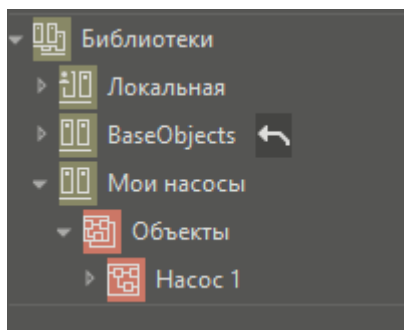
После этого библиотека сохранится на диске и будет доступна для подключения из других проектов через контекстное меню Подключить библиотеку. В данный момент она недоступна для подключения, т.к. уже подключена. Для того чтобы это продемонстрировать, удалим библиотеку Мои насосы из проекта через контекстное 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 MasterSCADA 4D

### 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 "ИнСАТ" 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 "ИнСАТ". Зарегистрированные пользователи нашего сайта могут найти их по ссылке: <http://insat.ru/products/?category=1755>

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 поддерживается 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 "ИнСАТ" 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.

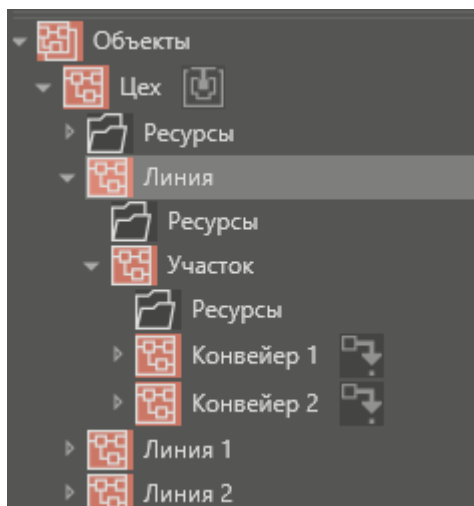
Скомпилированный проект – это один или несколько наборов файлов, полученных в результате преобразования (компиляции) проекта, которые загружаются в среду исполнения. Если, в случаях когда необходимо организовать межузловое взаимодействие, в среде разработки создается единый проект для нескольких устройств (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.

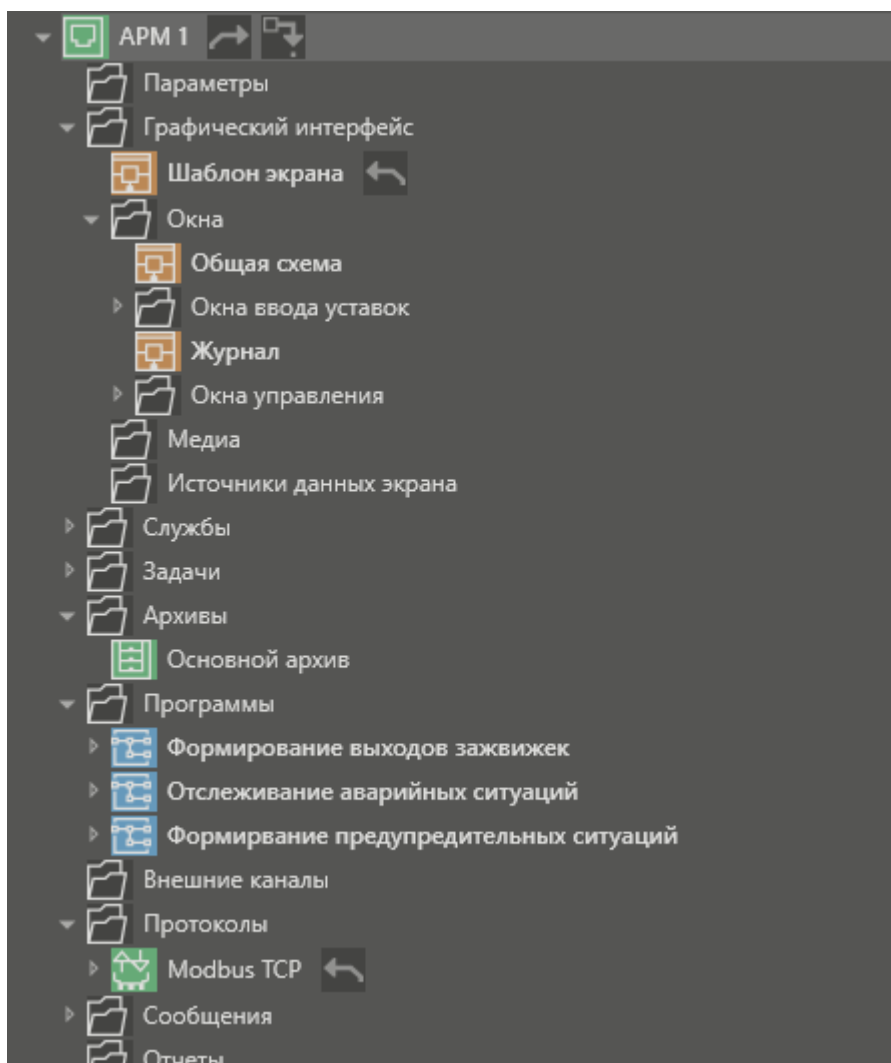
Наследование – механизм, обеспечивающий при изменении типа объекта в библиотеке соответствующее изменение и всех экземпляров этого объекта в проекте.

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. Состав MasterSCADA 4D

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

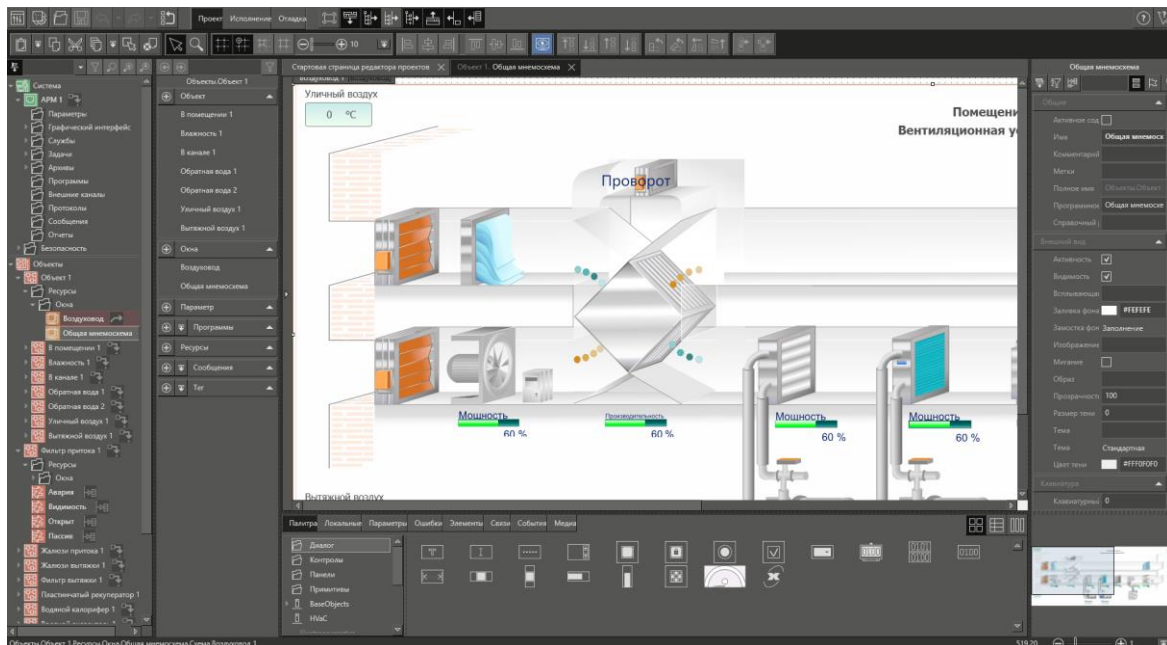
- инструментальная система/среда разработки;
- исполнительная система, состоящая из набора модулей;
- клиент визуализации.

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.

### **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

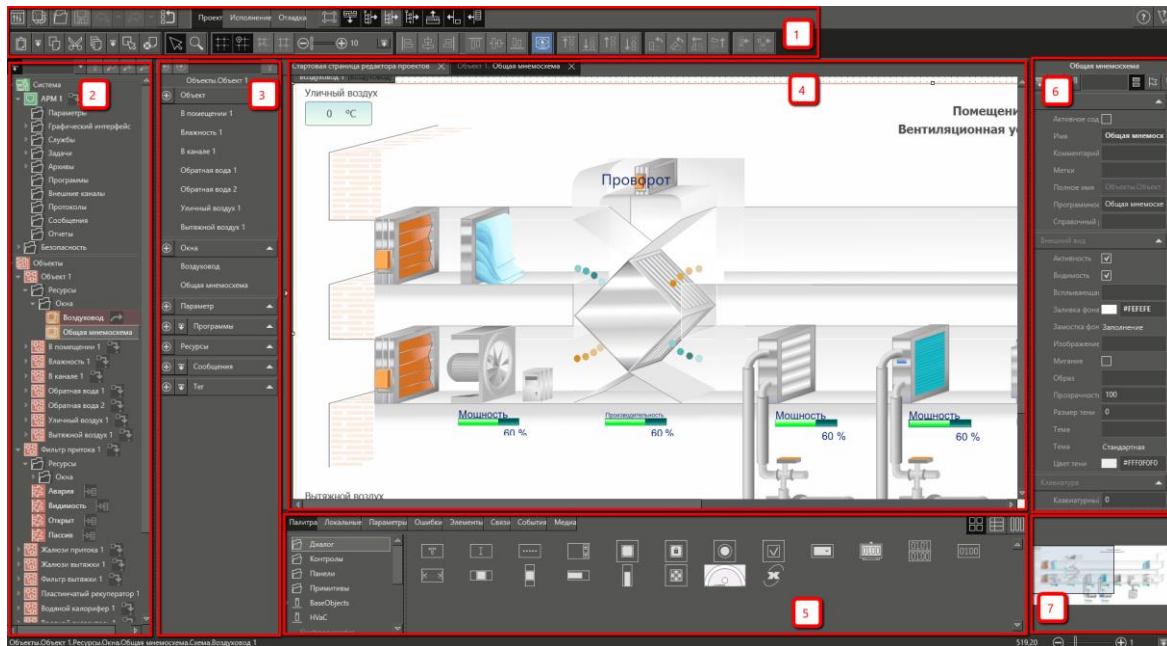


оператору на локальном АРМ. Естественно, это возможно только если был открыт доступ к данной информации или управлению.

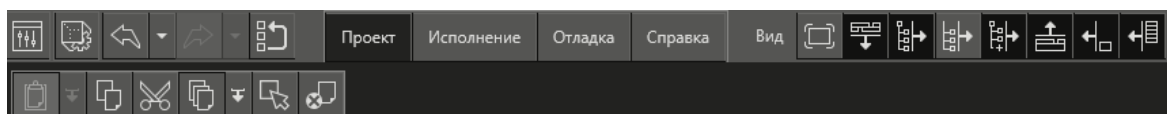
### 3.4. Описание интерфейса

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.

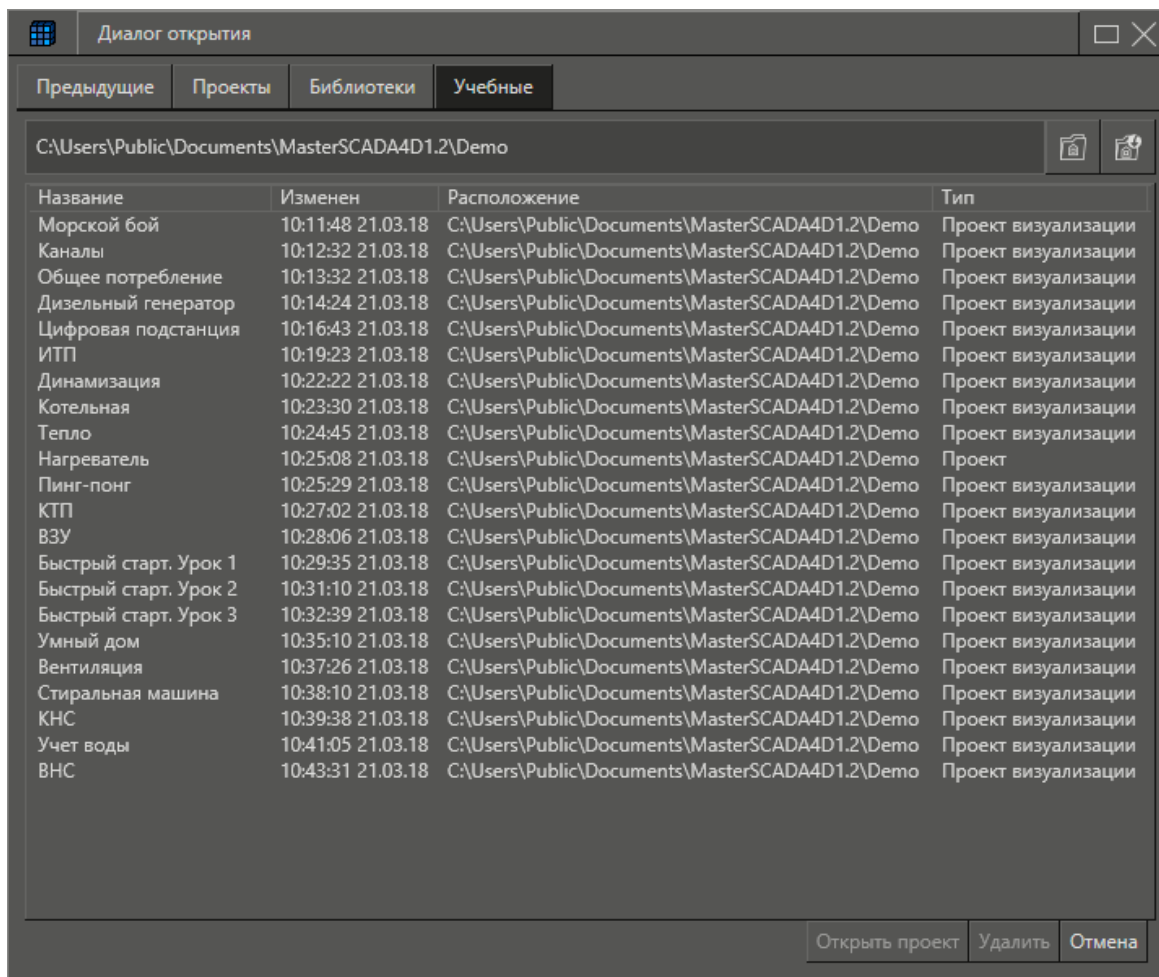
5. Легенда содержит элементы, которые используются в том или ином редакторе на закладке Палитра, и ряд вспомогательных закладок для отладки, анализа и работы. Набор закладок легенды меняется в зависимости от открытого редактора. Более подробно некоторые закладки рассматриваются в разделах, посвященных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

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

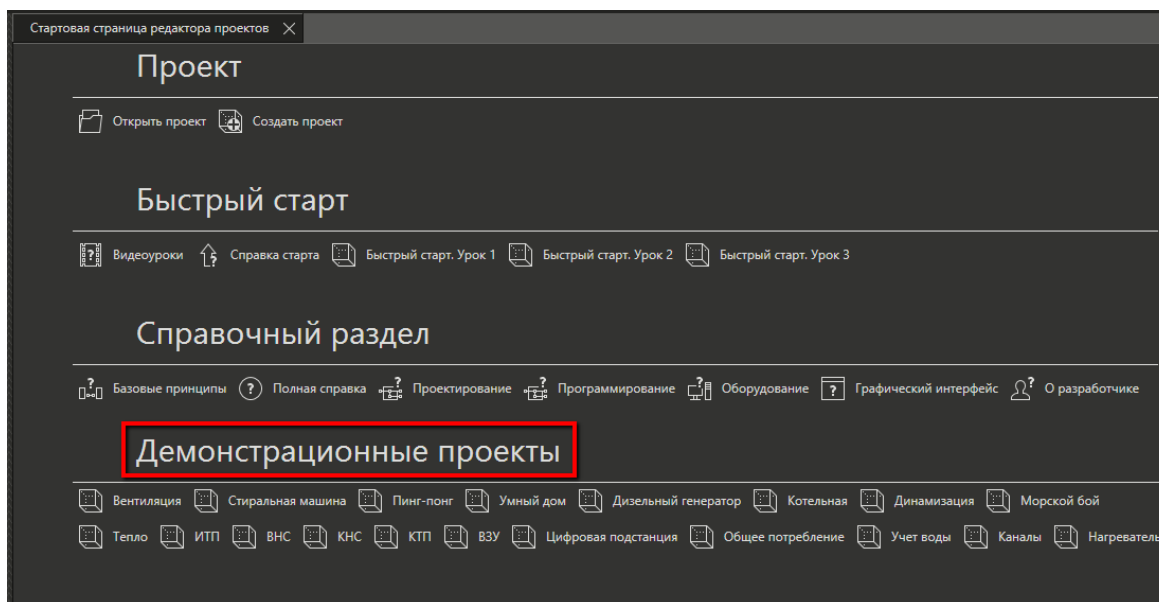
7. Миникарта – средство для навигации в окнах редакторов: FBD-, SFC-, LD-диаграмм и окон.

### **3.5. 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.

### 3.6. Лицензионная политика

Лицензионное соглашение на инструментальную систему MasterSCADA 4D™

#### ПРЕДУПРЕЖДЕНИЕ!

Настоящее лицензионное соглашение (далее - Соглашение) является юридическим документом, оно заключается между Вами – юридическим или физическим лицом, которое легальным способом получило лицензию на использование пакета прикладных программ MasterSCADA 4D™ и использует его в статусе конечного пользователя MasterSCADA 4D™, и обладателем исключительного права на MasterSCADA 4D™ (№ государственной регистрации 2010617624; № государственной регистрации 2012619676) (далее - Правообладатель).

MasterSCADA 4D™ включает в себя набор программ различной функциональной комплектности, записанный на электронных носителях или выложенный для скачивания в сети Интернет, печатные материалы и/или электронную документацию.

Устанавливая, копируя, запуская на выполнение или иным образом используя MasterSCADA 4D™, Вы тем самым принимаете на себя условия настоящего Соглашения. Если Вы не согласны с ними, то Правообладатель отказывается предоставить Вам право на какое-либо использование MasterSCADA 4D™ или лишает Вас такого права.

#### ЛИЦЕНЗИЯ НА MasterSCADA 4D™

MasterSCADA 4D™ является собственностью Правообладателя и охраняется законами и международными соглашениями об авторском праве, а также другими законами и договорами, относящимися к интеллектуальной собственности. MasterSCADA 4D™ лицензируется, а не продается.

**ОБЪЕМ ЛИЦЕНЗИИ.** Настоящее Соглашение дает Вам следующие права:

- \* Программное обеспечение. Вы имеете право использовать программное обеспечение MasterSCADA 4D™ в ознакомительных или учебных целях. Коммерческое применение разрешено исключительно с целью создания прикладных программ для тех программируемых контроллеров, графических электронных панелей, наименования и модели которых обозначены в штатном интерфейсе MasterSCADA 4D™.
- \* Ответственность за ущерб. MasterSCADA 4D™ предоставляется Вам «как есть», то есть Вы осознанно используете MasterSCADA 4D™ в своих целях, понимая, что в любом программном обеспечении возможны различные ошибки, как программного

кода, так и алгоритмические. Все риски за прямые или косвенные последствия, связанные с проявлением этих ошибок, Вы, используя MasterSCADA 4D™, осознанно берете на себя.

- \* **Хранение, копирование и распространение.** Вам разрешается хранить копии носителя, на котором поставляется MasterSCADA 4D™ 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. Передача носителя третьим лицам допустима только в ознакомительных целях, без права коммерческого использования и с уведомлением их об источнике получения данного носителя.
- \* **Техническая поддержка.** Техническую поддержку MasterSCADA 4D™ выполняет авторизованная для этих целей Правообладателем компания ООО «ИнСАТ» (далее – ИнСАТ). ИнСАТ предоставляет техническую поддержку в виде консультаций и ответов на вопросы. Техническая поддержка осуществляется через адрес электронной почты support.ms4d@insat.ru на русском языке.
- \* **Правообладатель сохраняет за собой все права, которые не передаются Вам в явном виде.**

**ОГРАНИЧЕНИЯ.** Настоящее Соглашение накладывает на Вас следующие ограничения:

- \* **Авторское право.** MasterSCADA 4D™, в том числе включенные примеры использования, электронная и печатная документация, графические и интерфейсные решения, отдельные графические изображения, исходные тексты программных примеров, сценариев, интерфейсов расширения, является субъектом авторского права. Копирование печатных материалов, сопровождающих MasterSCADA 4D™, **ЗАПРЕЩЕНО.**
- \* **Обратное конструирование, декомпиляция и дизассемблирование.** Запрещено предпринимать обратное конструирование, декомпиляцию и дизассемблирование MasterSCADA 4D™.
- \* **MasterSCADA 4D™ лицензируется как единый продукт.** Составляющие его части нельзя разделять для использования на нескольких компьютерах. Исключение составляют исходные тексты программных интерфейсов расширения, о которых в MasterSCADA 4D™ явно прописано, что они предназначены для разработки пользовательских программных решений, используемых исключительно в комплекте с MasterSCADA 4D™.
- \* **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 4D™ и все составляющие его части.

### 3.7. Принятые сокращения и обозначения

В данной документации используются следующие сокращения:

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

HMI – 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.

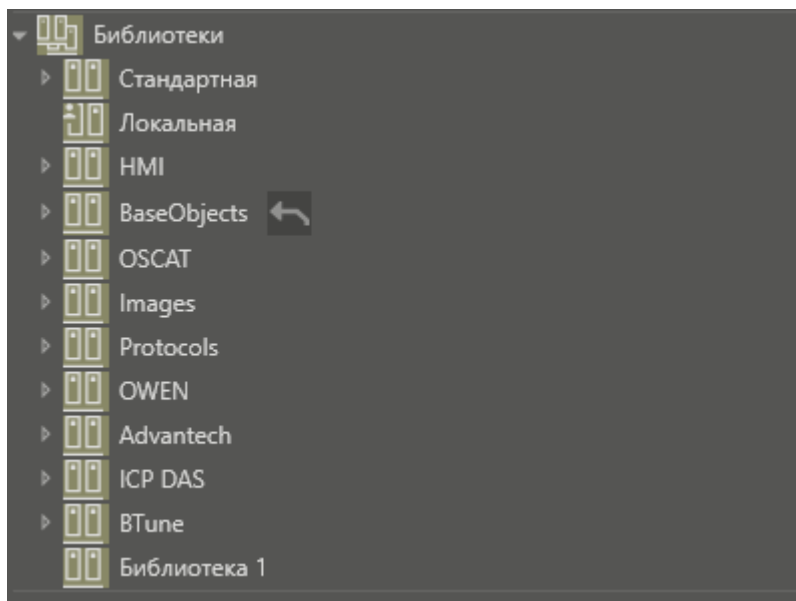
## 4.1. Анализ объекта автоматизации

Прежде чем приступить 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.

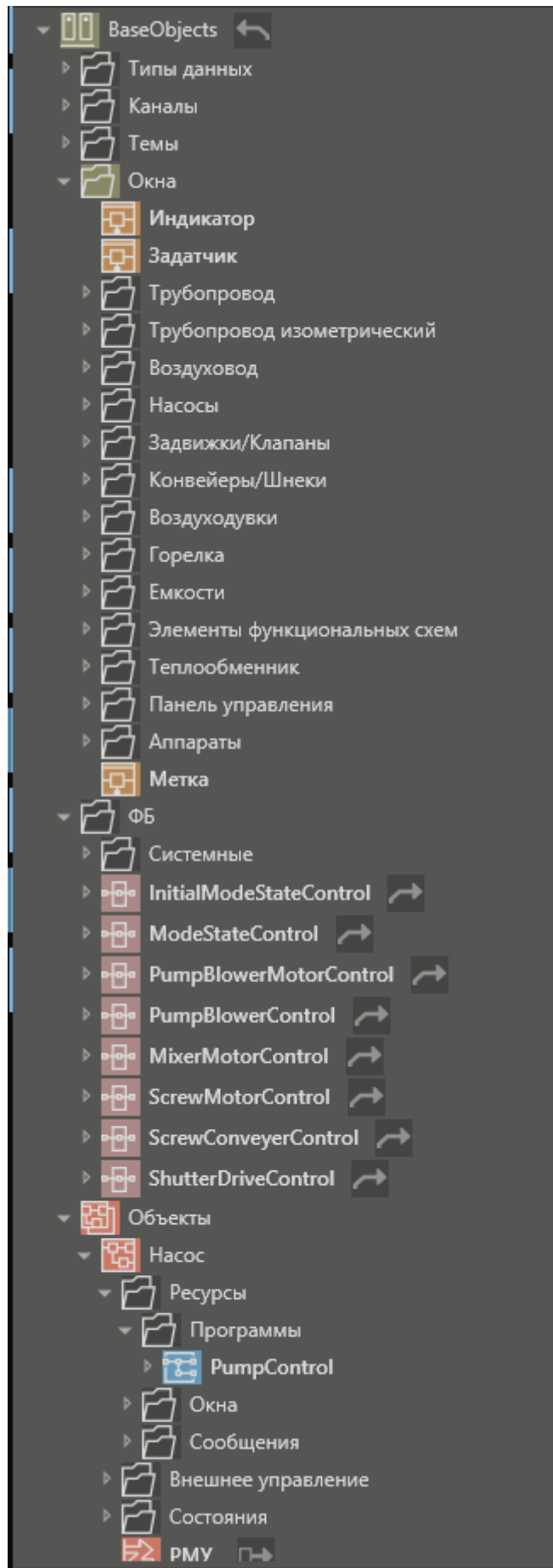


Если необходимо создать библиотечный объект, который включает в себя окна и программы, то сначала создается библиотечная программа, библиотечное окно, а затем уже библиотечный объект.

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

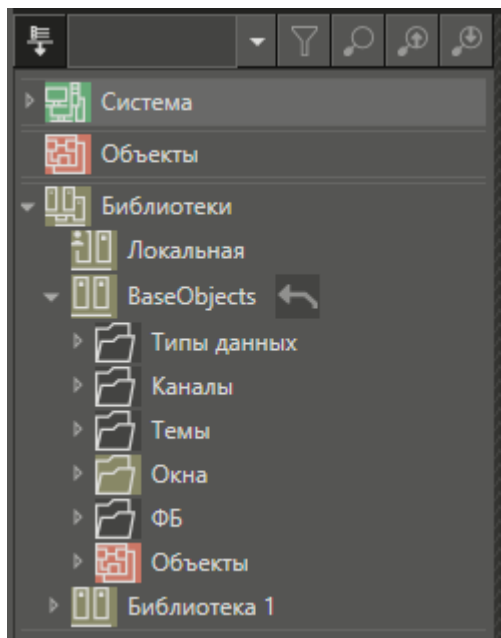
Так, например, если проанализировать объект Насос, то можно заметить, что сначала были созданы окна и ФБ, а затем из этих элементов собрали библиотечный объект:





### 4.3. Создание структуры проекта

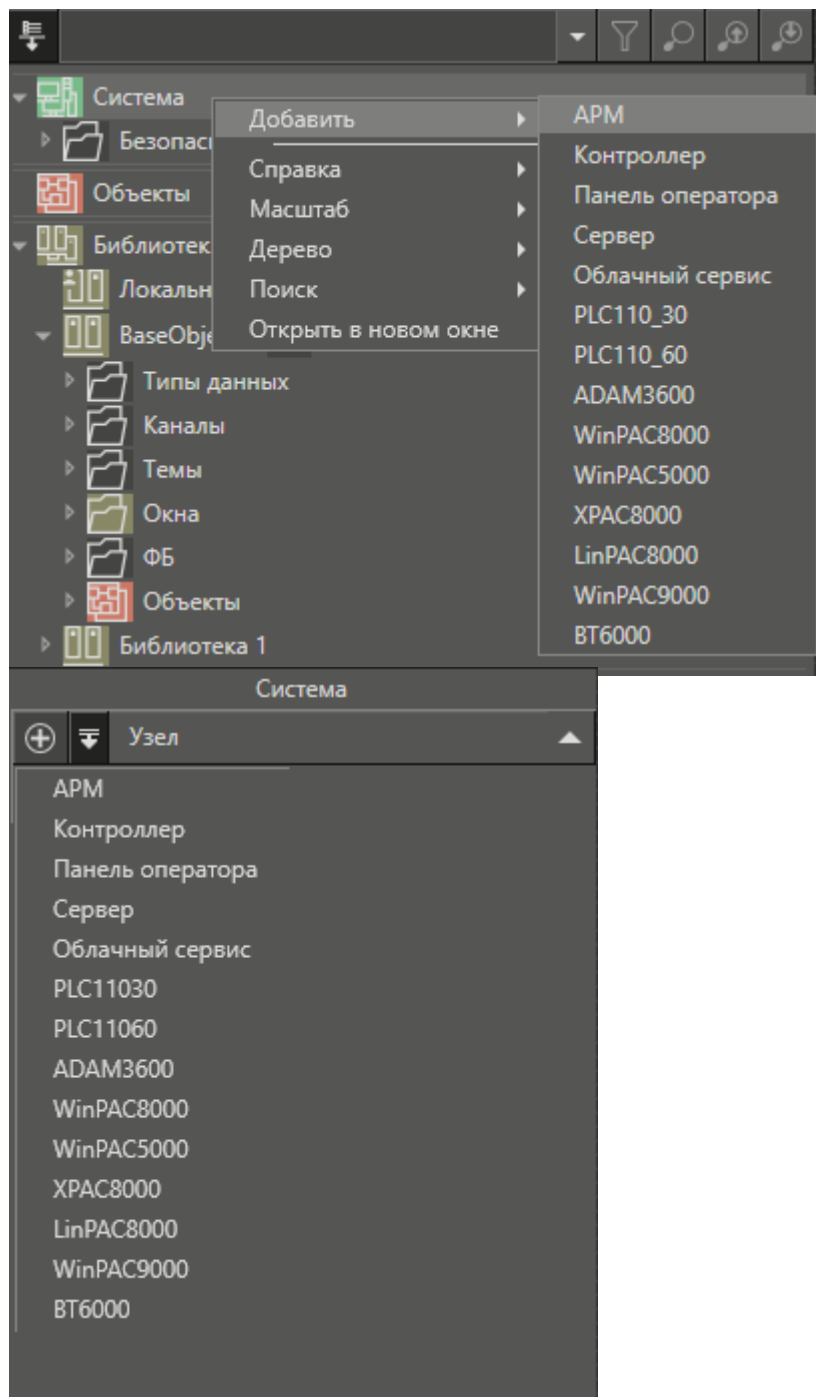
В MasterSCADA 4D проект создается из библиотечных элементов в дереве системы и дереве объектов.



#### Дерево системы

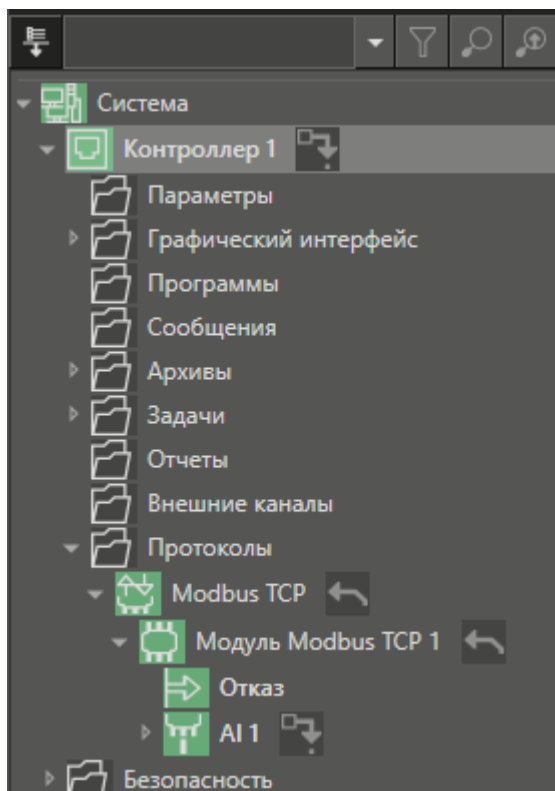
В этой группе создаются узлы – конфигурации, загружаемые для исполнения в реальные устройства проекта (АРМы, контроллеры и т. д.).

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):



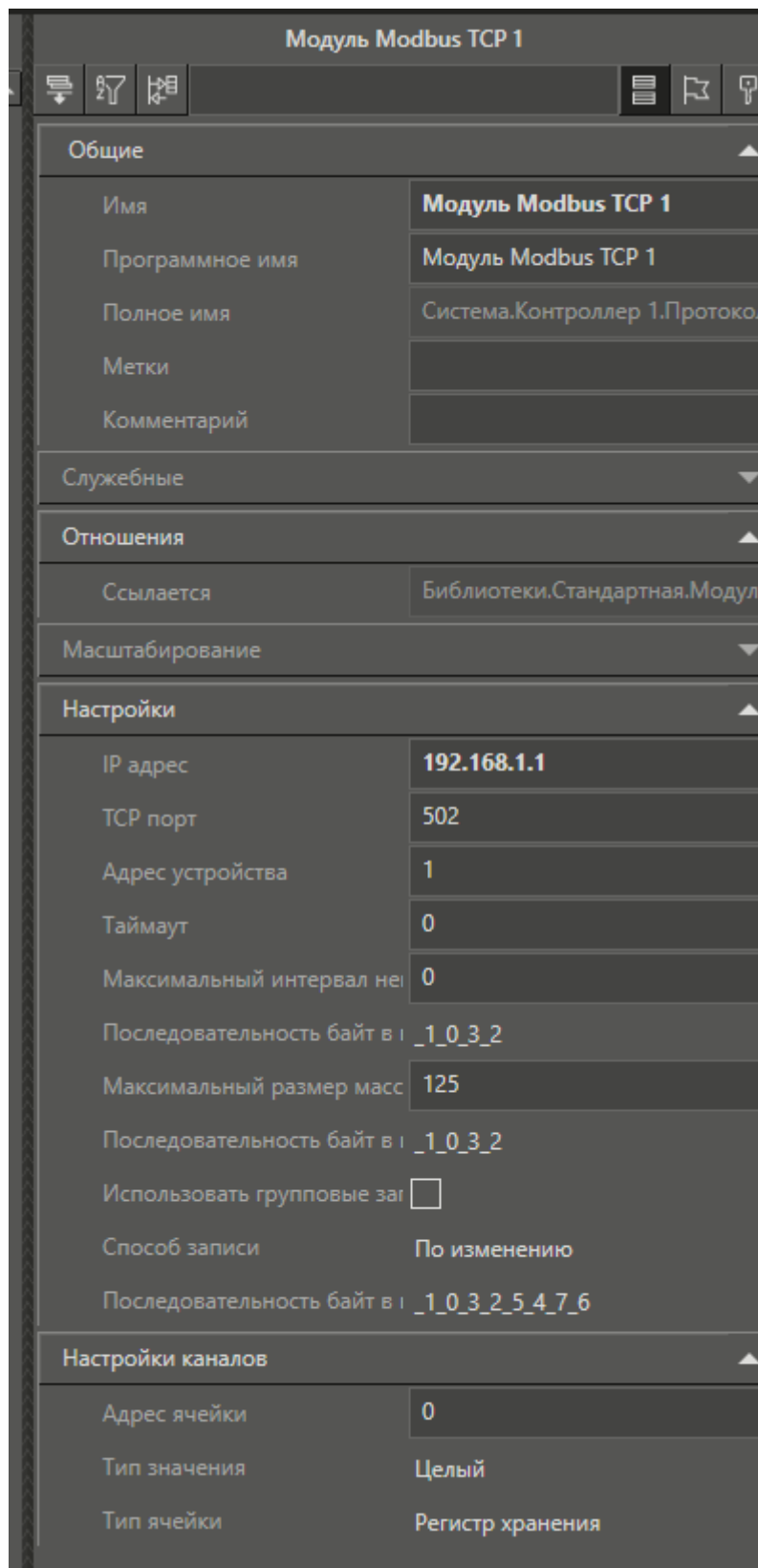
Для всех реальных аппаратных и программных элементов системы управления (АР-Мов, контроллеров, портов, протоколов и т.п.) в структуре проекта создаются соответствующие элементы проекта. Набор всех необходимых элементов содержится в библиотеках, откуда элементы могут быть вставлены в соответствующую группу 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:

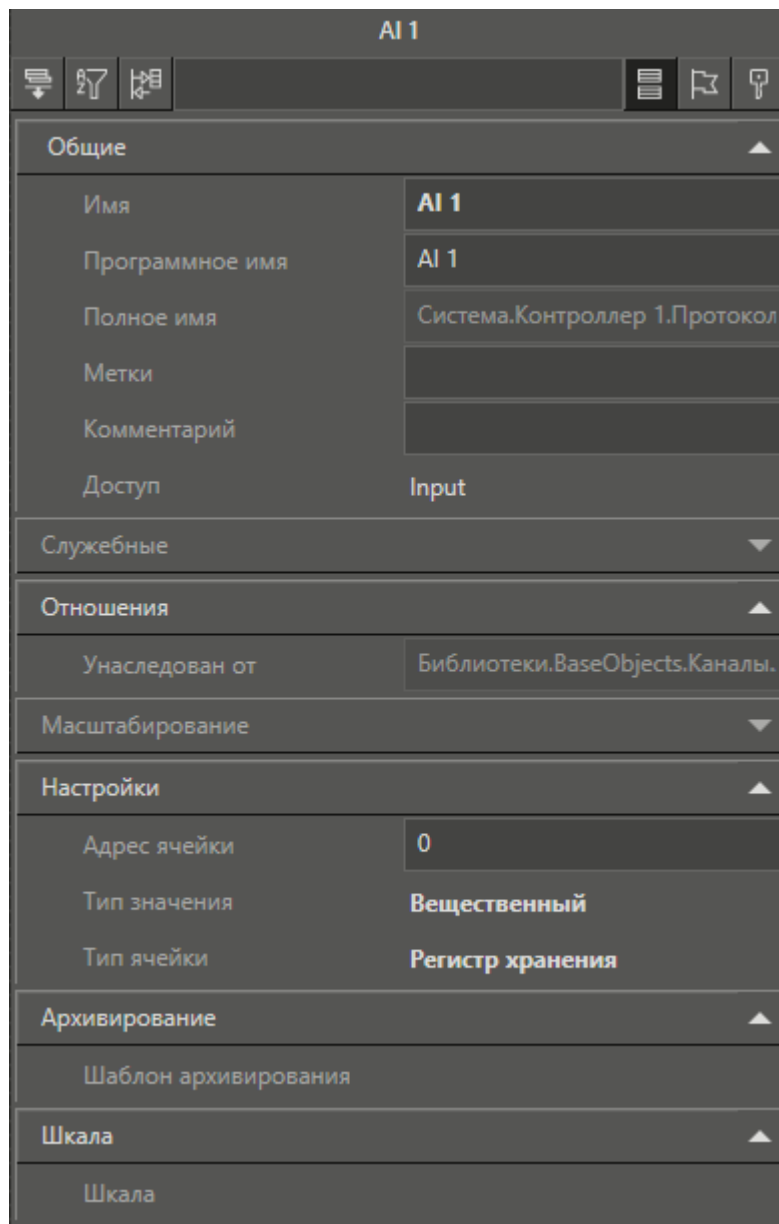
The screenshot shows the configuration interface for a Modbus TCP module. The window title is "Modbus TCP". At the top, there are several icons for navigation and actions. The configuration is organized into sections:

- 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:



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

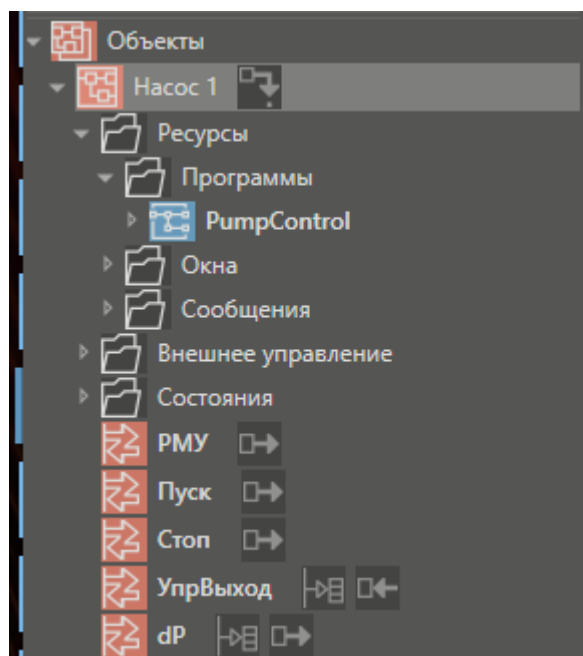


Канал – это структура, содержащая один или несколько параметров для работы с данными и ряд конфигурационных параметров. Например, канал обмена по протоколу Modbus содержит значение, прочитанное драйвером или поставленное в очередь на запись. Специфическими конфигурационными параметрами такого канала 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

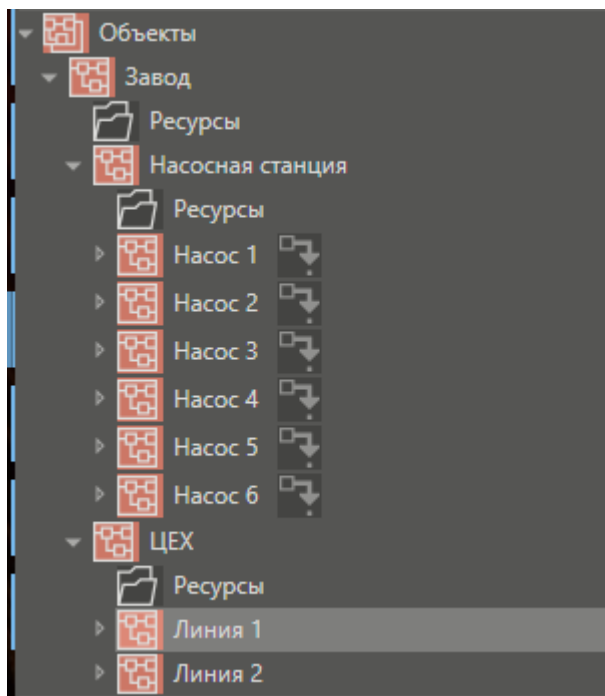
В этой группе задается вся логика проекта. При этом в MasterSCADA 4D основной идеологией при разработке проекта является объектно-ориентированный подход.

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:



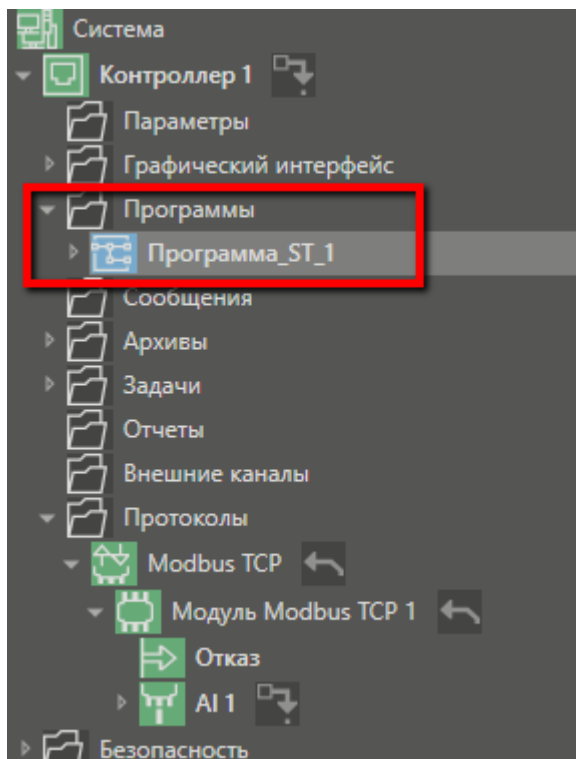


Важно! Типовые объекты разрабатываются заранее и помещаются в библиотеку, а впоследствии из них, как из кирпичиков, формируется дерево проекта.

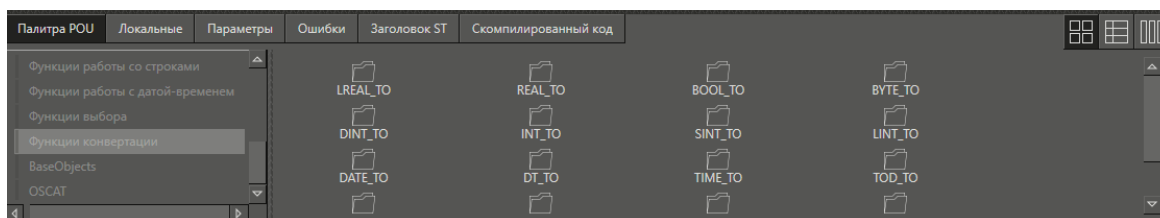
Если узел в проекте один, то никаких действий для добавления того или иного объекта в узел не требуется, все объекты будут назначены в него автоматически (т.е. в режиме исполнения написанная программа будет исполняться в системе единственного узла). Если узлов несколько, то для того чтобы задать исполнение 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):

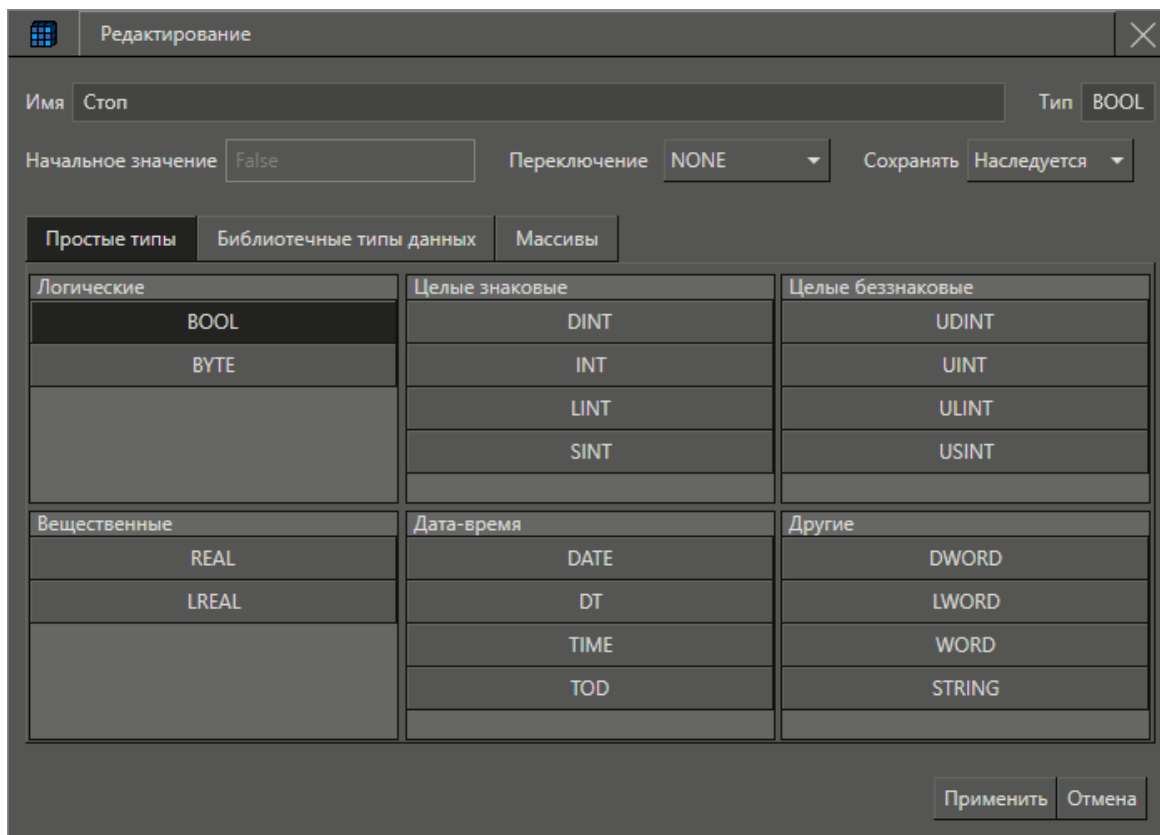


Для упрощения разработки программ в MasterSCADA 4D встроена библиотека Стандартная, а также поставляются различные подключаемые библиотеки (например, библиотека OSCAT), которые содержат большое количество функций и типов функциональных блоков. Предустановленные ФБ и функции доступны в легенде по закладкам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



### Параметры (переменные проекта)

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

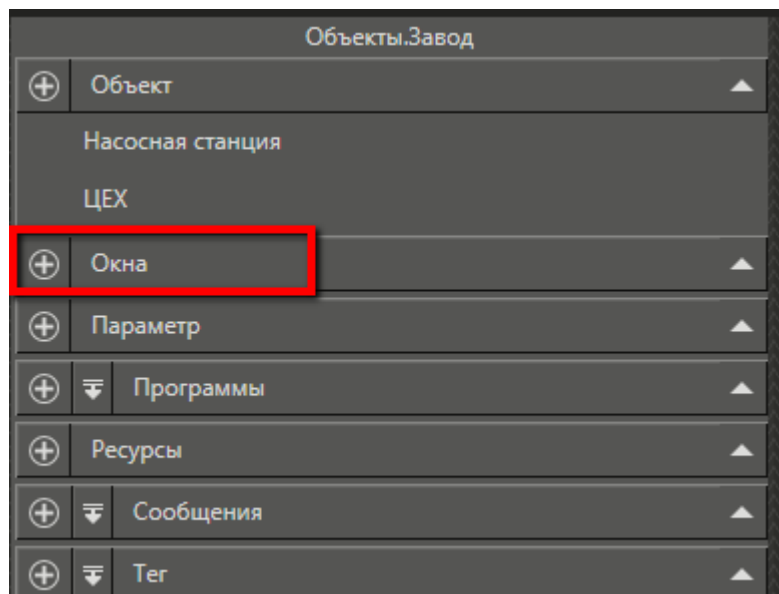


Типы параметров можно также задать в панели свойств:

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

## Окна

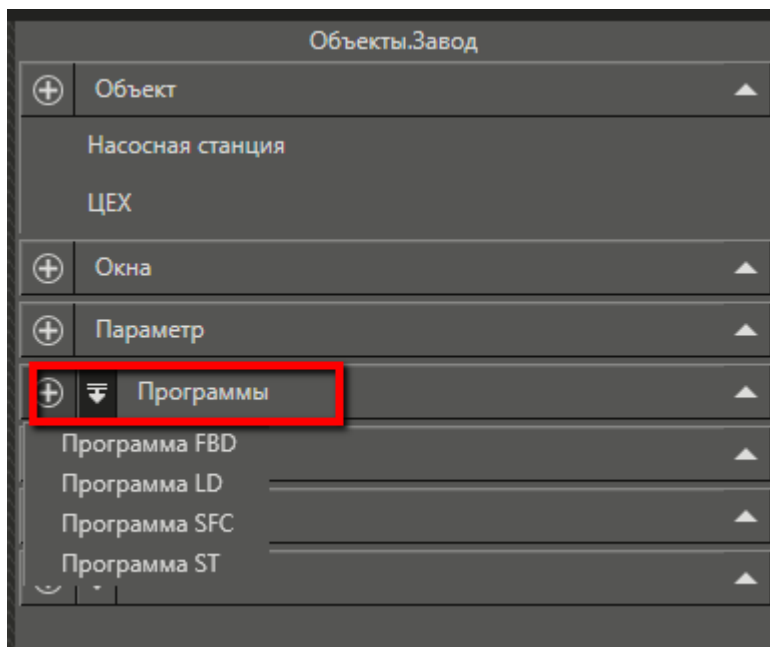
Окна создаются при помощи контекстного меню Объекта или Узла. При этом в центральной части интерфейса MasterSCADA 4D откроется редактор HMI.



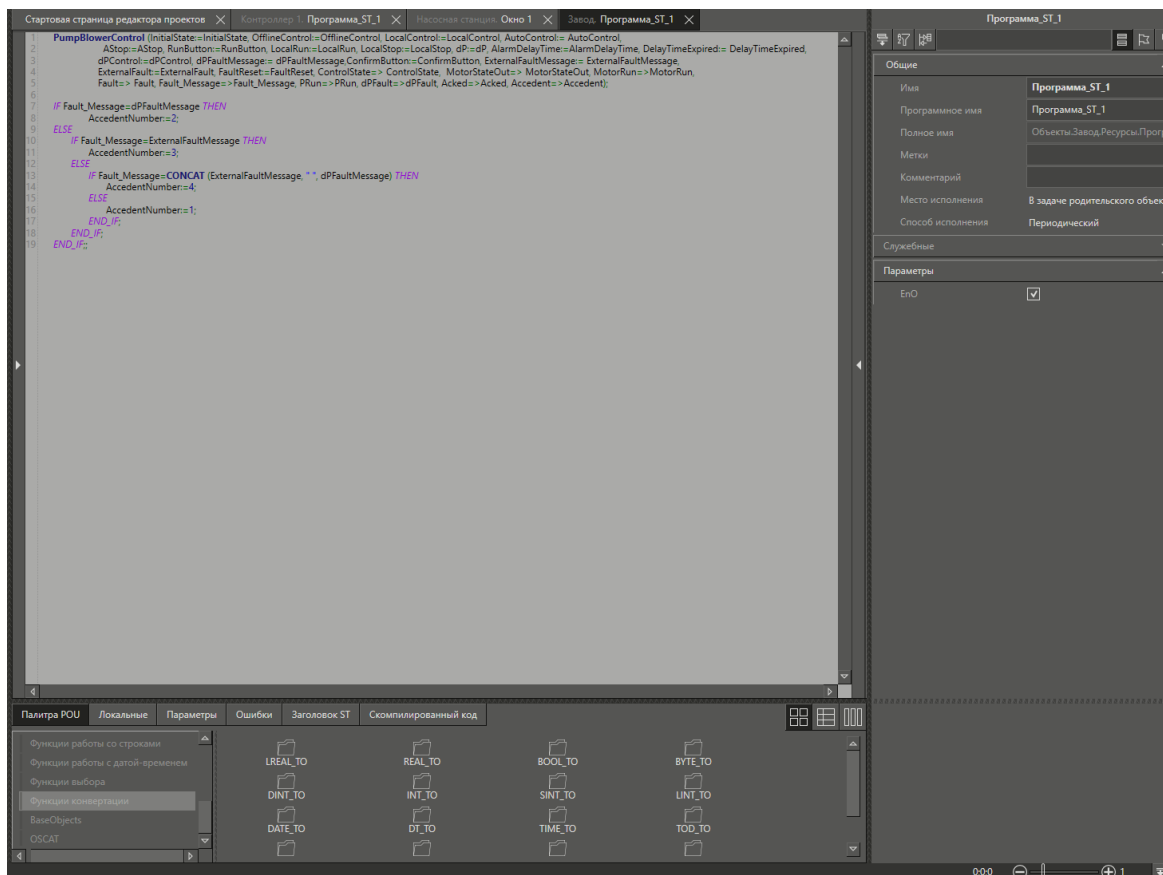
#### 4.4. Разработка и отладка программ

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:

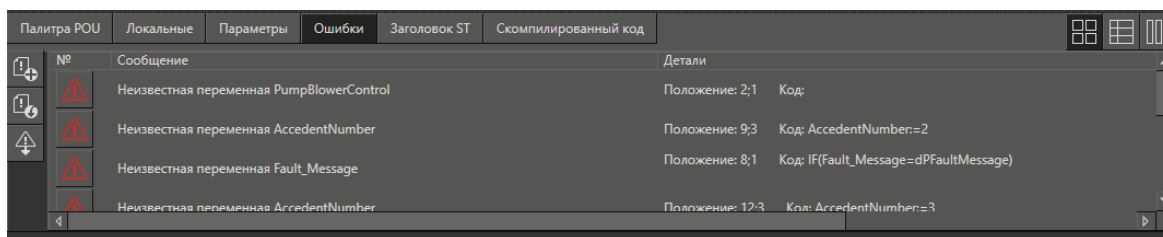


Для проверки программы на наличие ошибок следует нажать кнопку Проверить программу на панели инструментов Проверить вкладки Проект.



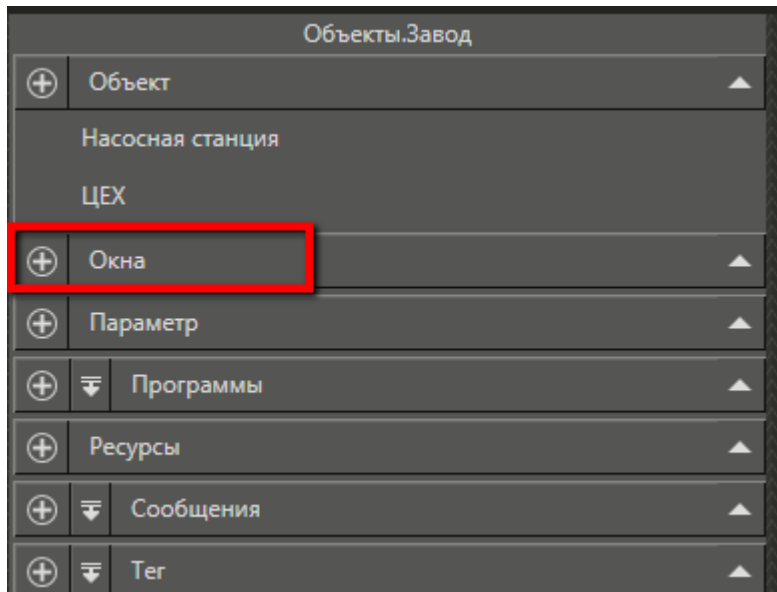
Если ошибок не обнаружено, то кнопка останется синей, а на вкладках Заголовок ST и Скомпилированный код легенды отобразятся соответственно заголовок и полный текст программного элемента на языке Master ST.

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

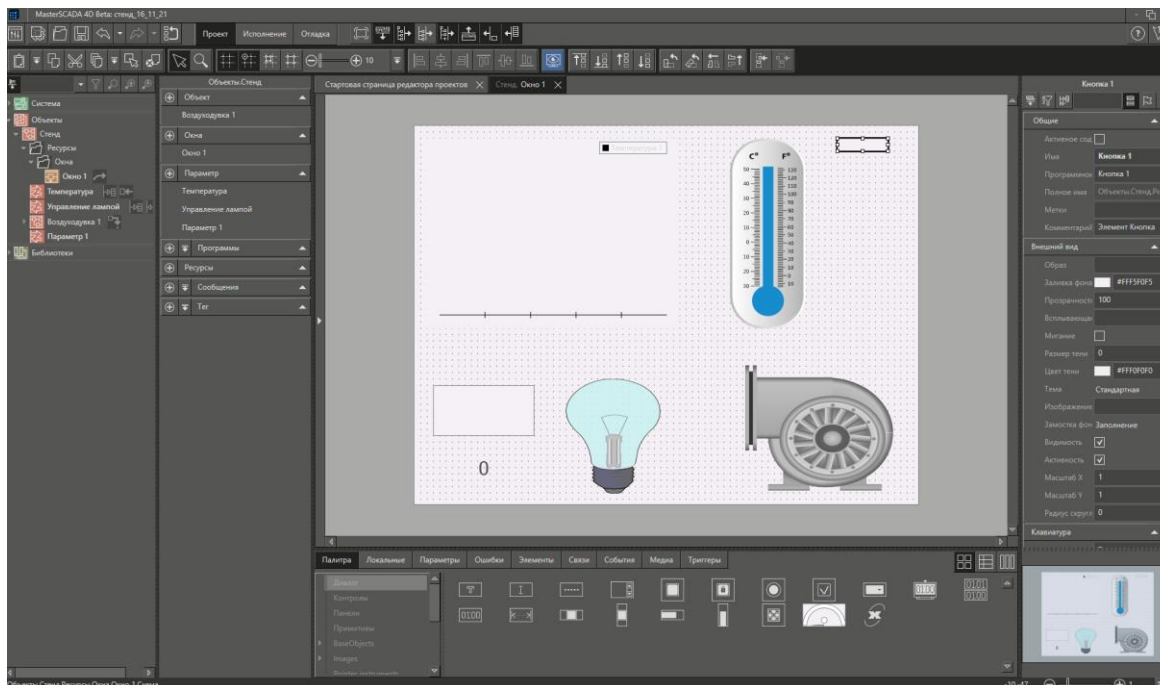


## 4.5. Разработка окон

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.




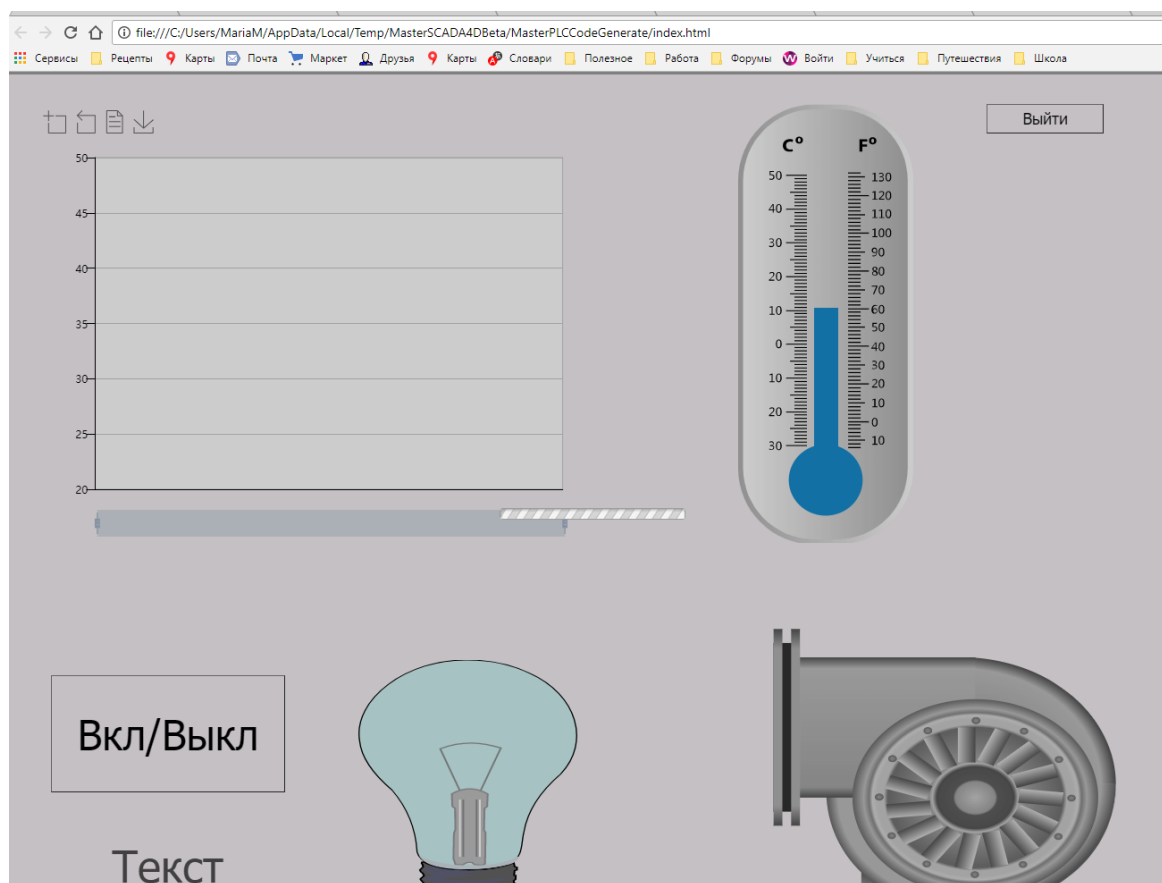
Для того чтобы открыть окно (мнемосхему) в редакторе, необходимо в дереве объектов дважды нажать ЛК на соответствующий элемент Окно. Редактор HMI открывается в виде вкладки в специальной панели (см. Панель документов и редакторов), при этом легенда отображает различную информацию мнемосхемы, а специфические 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 HMI, 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.

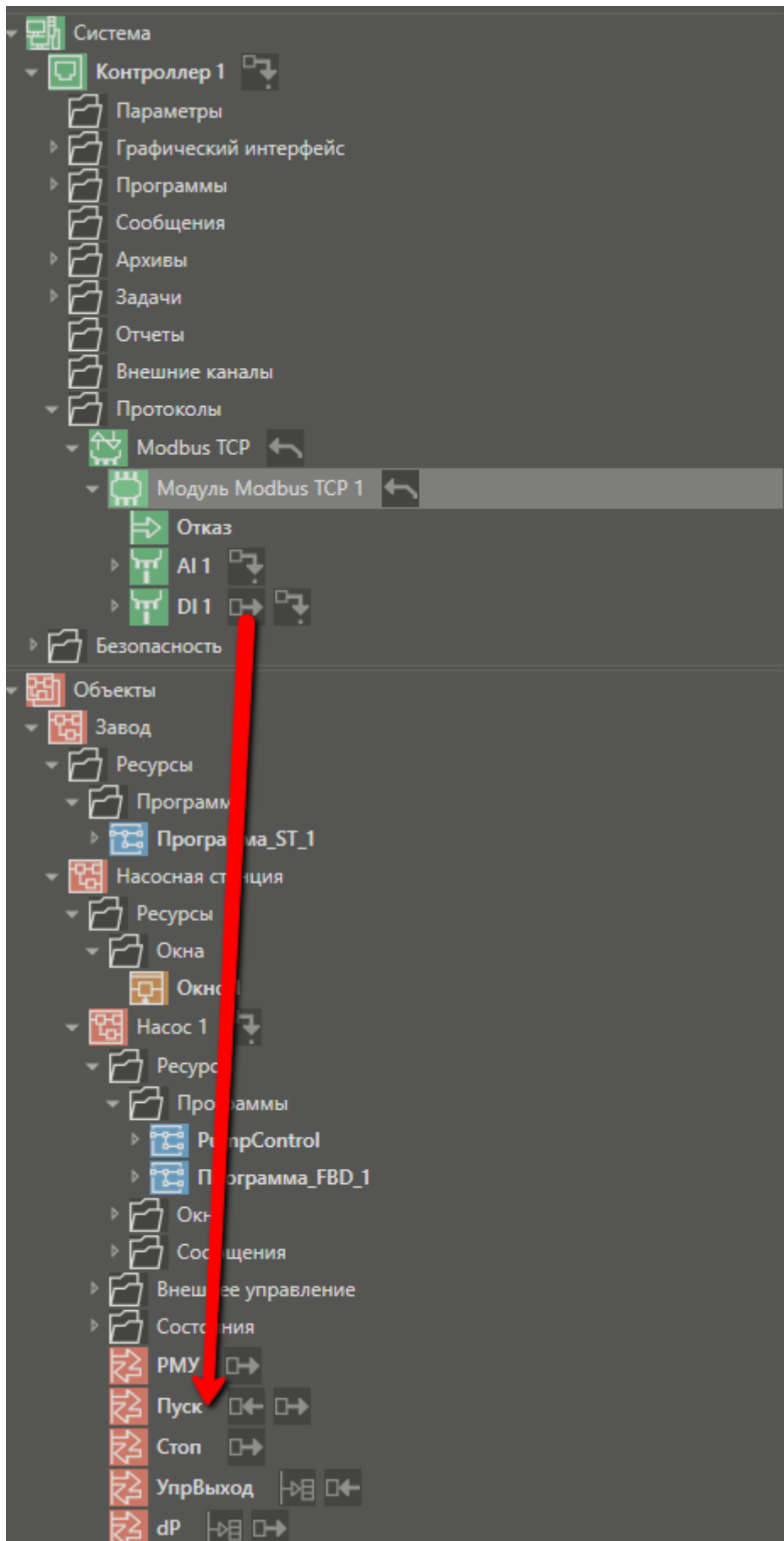
Для проверки и предварительного просмотра мнемосхемы в браузере (в статическом режиме) следует нажать кнопку  панели Просмотр графического редактора:



## 4.6. Конфигурирование связей

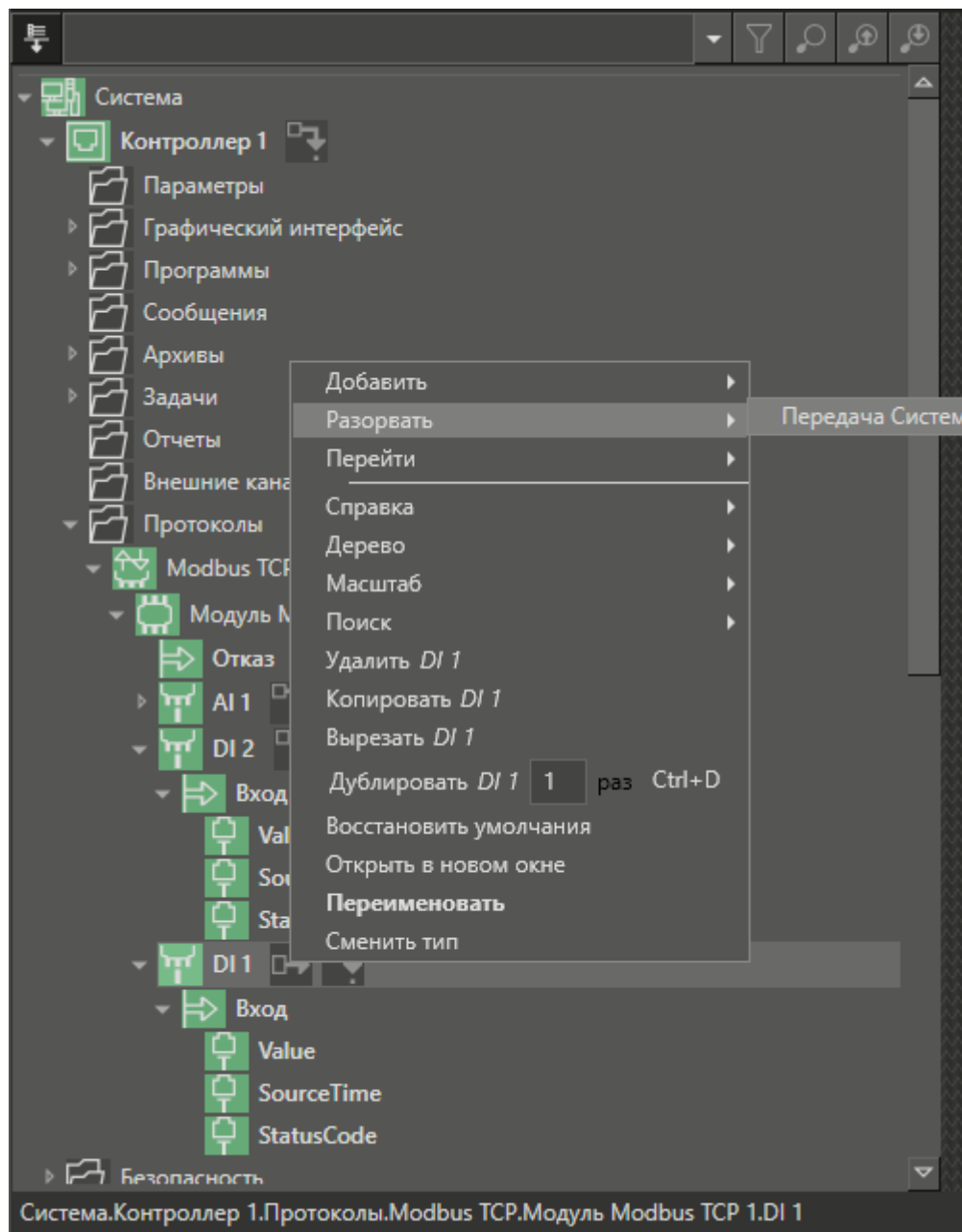
Связи в пределах одного узла (например, Параметр канала – Параметр программы) создаются с помощью перетаскивания элемента на элемент.

**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:

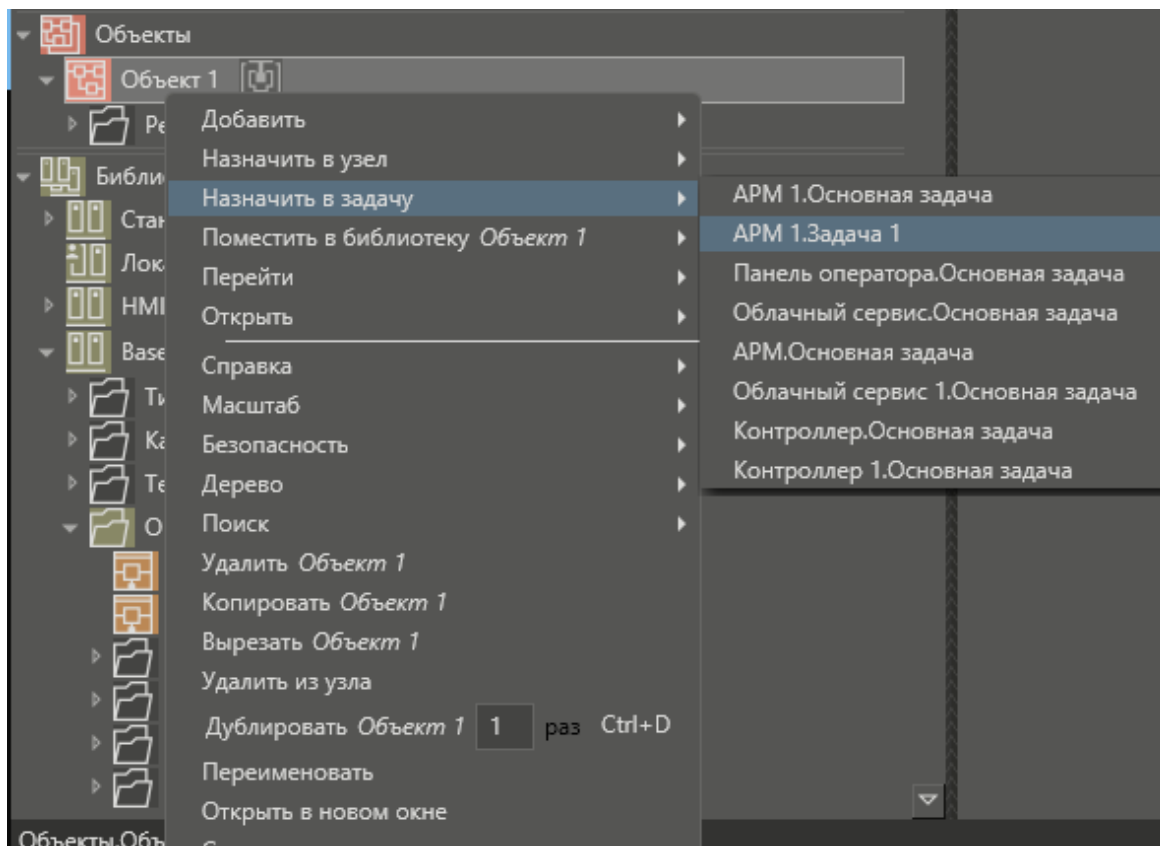
Для удаления связи используется команда Разорвать контекстного меню элемента.



## 4.7. Формирование задач узлов

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

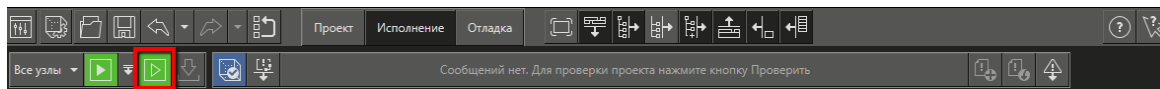
В случае если необходимо, чтобы часть объектов или программ исполнялись независимо от основной задачи, то создаются дополнительные задачи в узле в соответствующей группе. Затем тот или иной элемент проекта при помощи контекстного меню назначается в ту или иную задачу.



## 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).

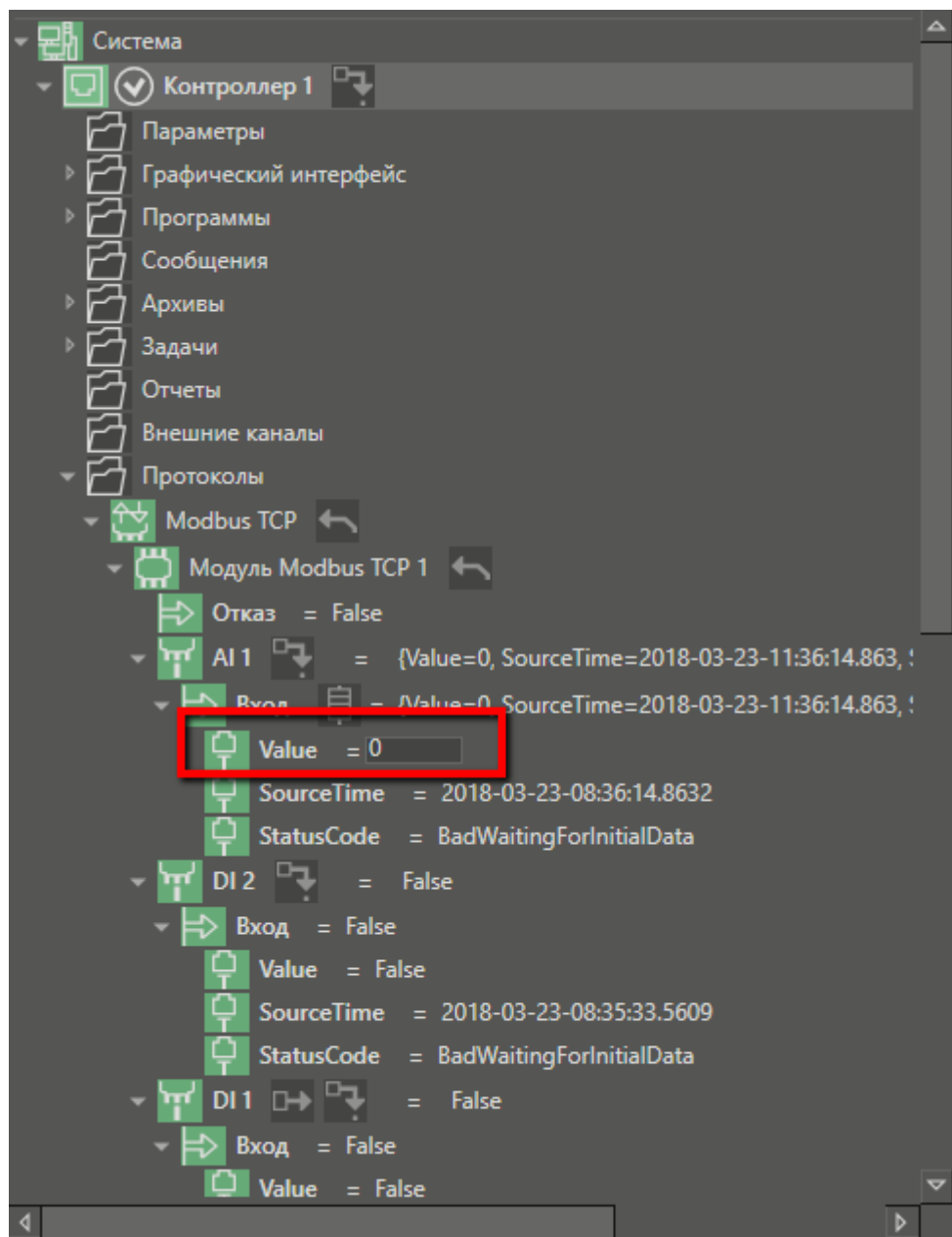
Для запуска эмуляции узла необходимо либо в контекстном меню выполнить пункт Узел.Подключить узел в режиме эмуляции, либо нажать на кнопку Эмуляция во вкладке инструментов Исполнение.



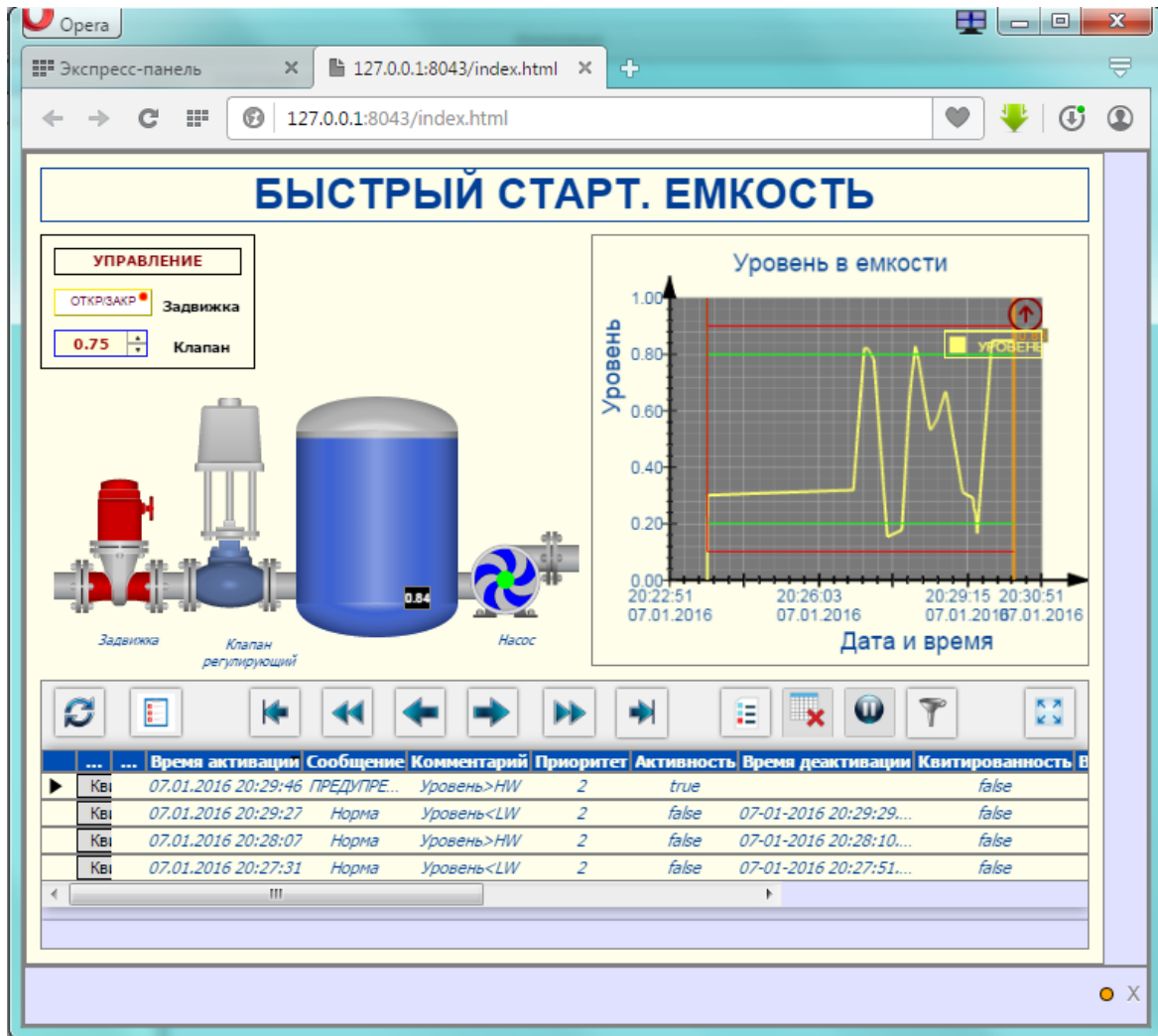
Важно! В режиме эмуляции опрос подключенных к узлу по различным протоколам устройств не происходит.


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:

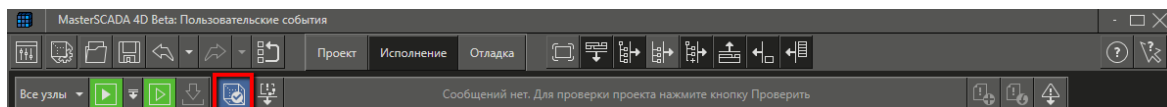


Для выхода из режима эмуляции необходимо нажать кнопку  Отключить панели инструментов вкладки Исполнение.

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.

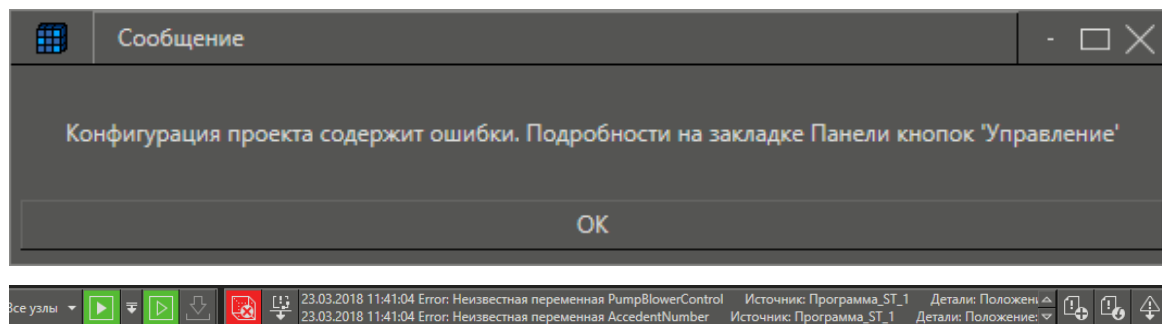


Для проверки конфигурации отдельного узла следует выполнить соответствующую команду контекстного меню в дереве системы узла.

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:



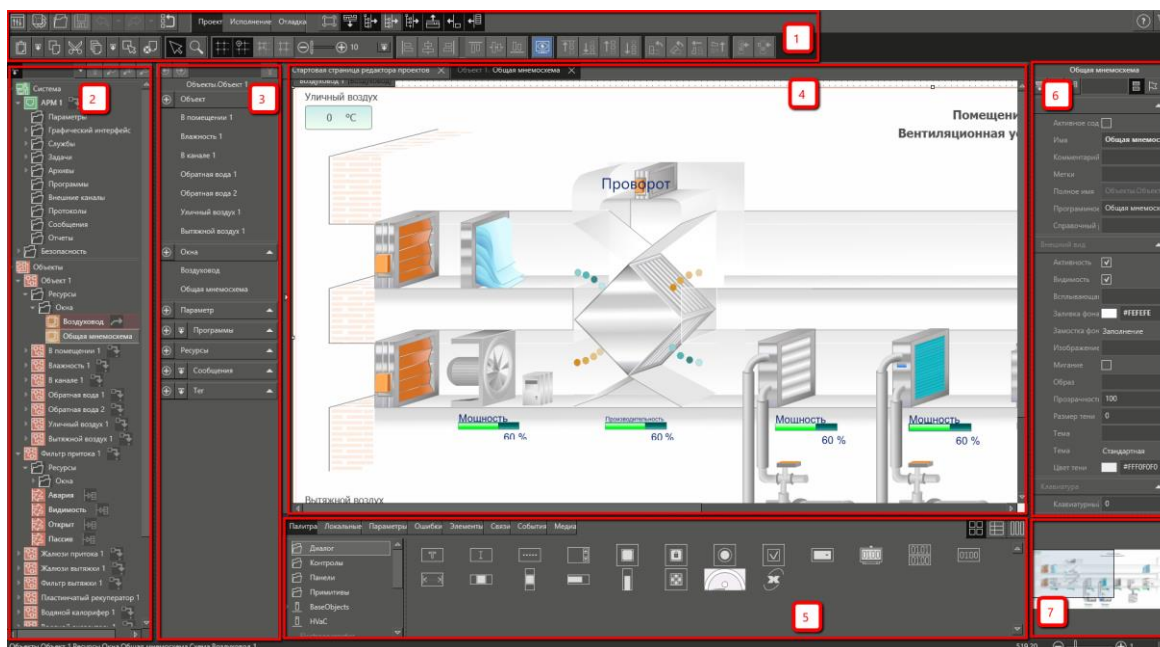
Для загрузки конфигурации в устройство следует выполнить соответствующую команду контекстного меню в дереве системы узла.

## 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 происходит редактирование элемента - программы или окна. Элемент для редактирования открывается двойным кликом мыши.

Подробное описание интерфейса приведено в разделе Меню редактора проекта.

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:

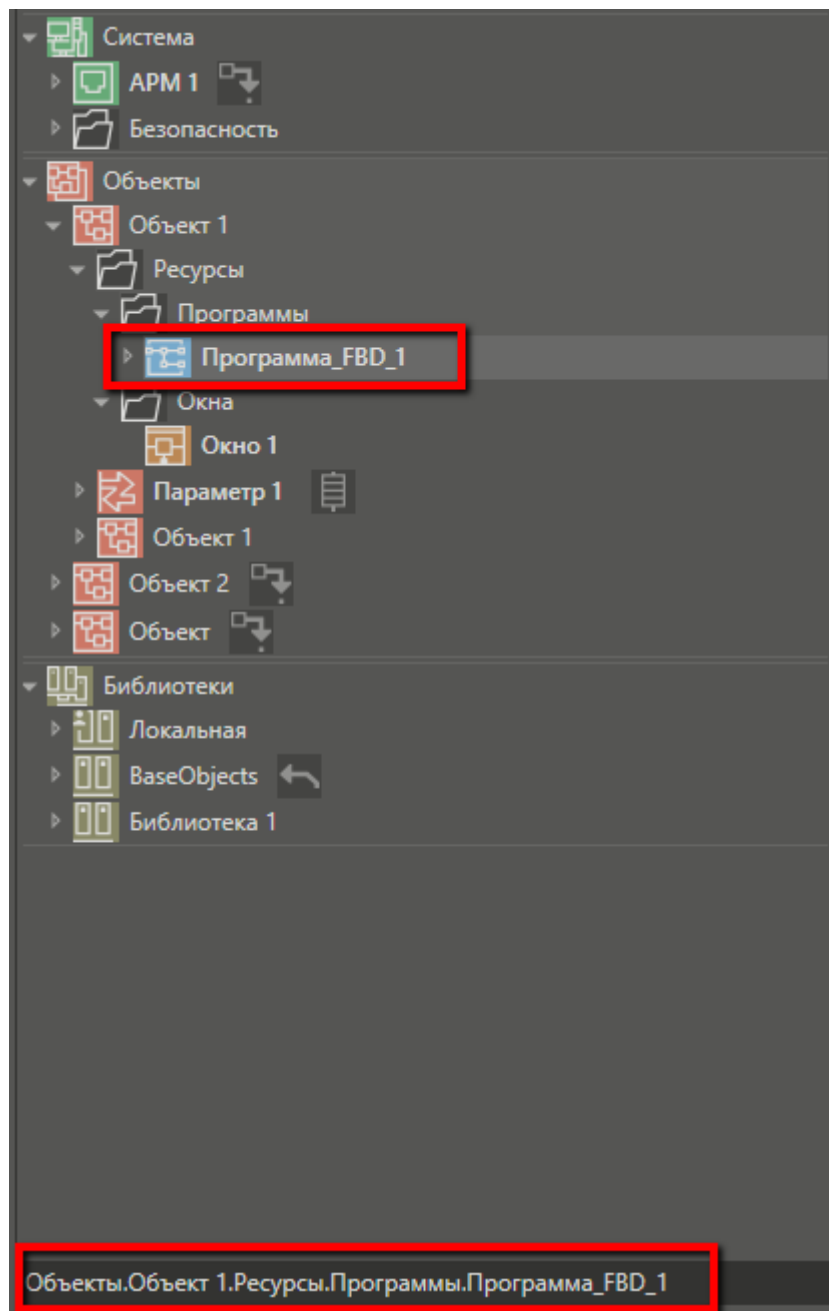
- строка статуса;
- комбинированный инструмент масштабирования:



- поле позиции курсора.

### **Строка статуса**

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



### Поле позиции курсора

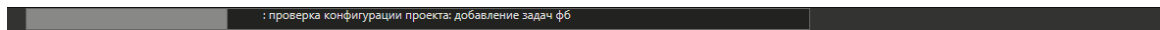
В этом поле отображается позиция курсора в тексте ST-программы. Поле расположено справа от комбинированного инструмента масштабирования. Позиция курсора отображается как <номер строки>;<знакоместо в строке>:



Данное поле отображается, если в окне документов и редакторов активен редактор ST.

## Выполнение длительных операций

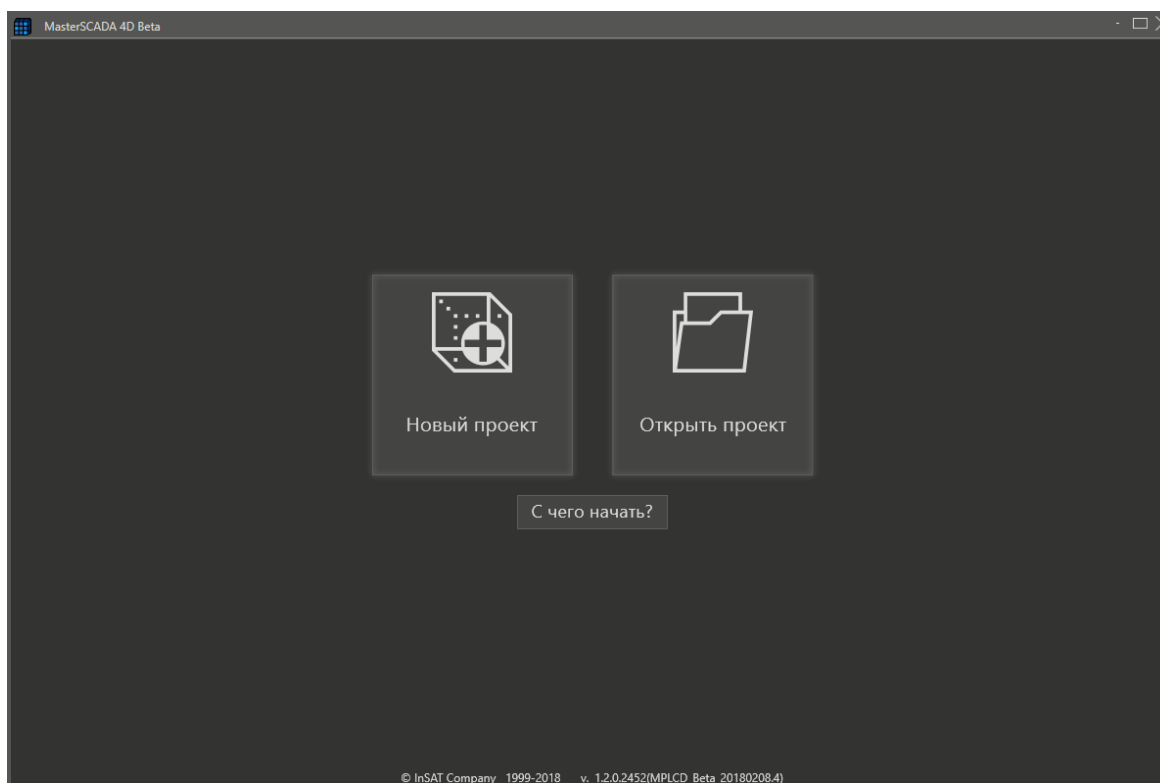
При выполнении отдельных операций, которые могут занимать длительное время, в строке статуса отображается индикатор выполнения (ProgressBar):



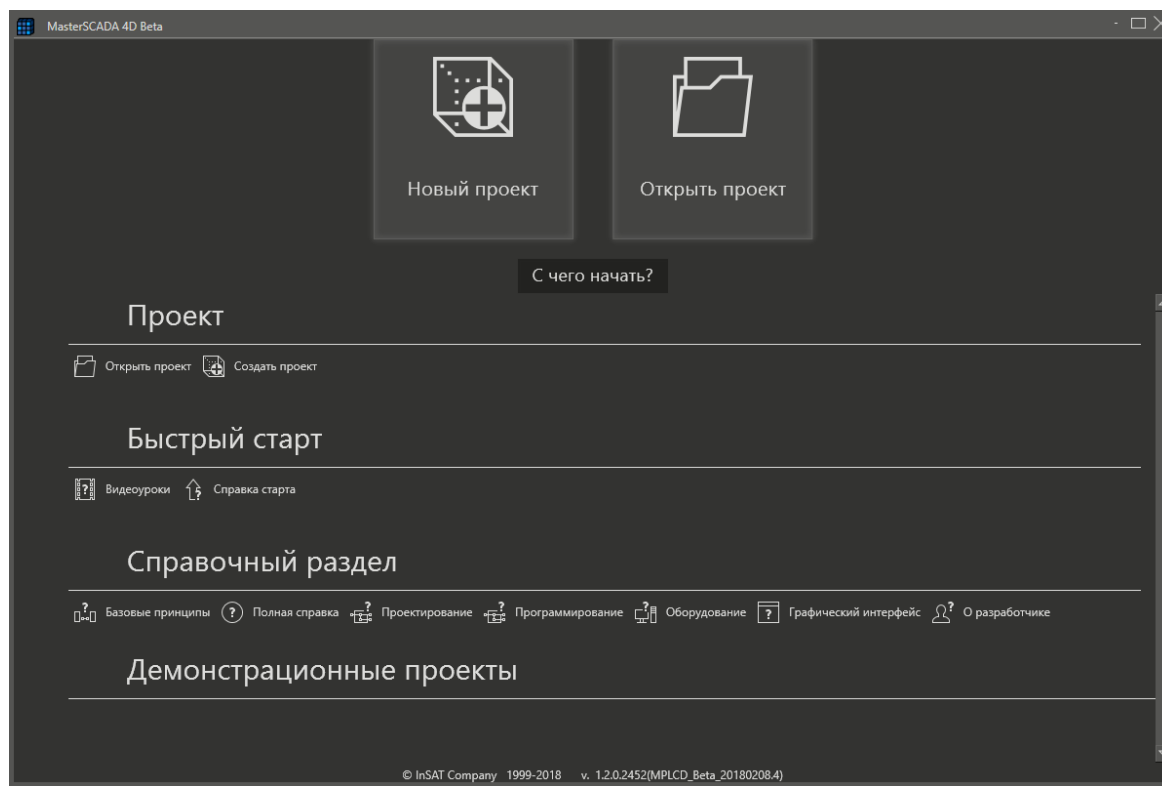
К подобным операциям относятся: подключение библиотек, обновление библиотек, копирование в буфер, вставка из буфера, удаление, проверка конфигурации проекта и др.

## 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

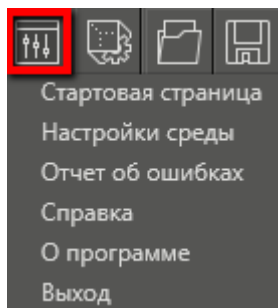
	действий через панель отмены/возврата действий. После выполнения этой команды процесс среды разработки может занимать меньший размер оперативной памяти
	Панель отмены и возврата действий позволяет вернуться к любому состоянию проекта до последнего сохранения.
Вкладка Проект	Открывает панель инструментов, 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.
	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

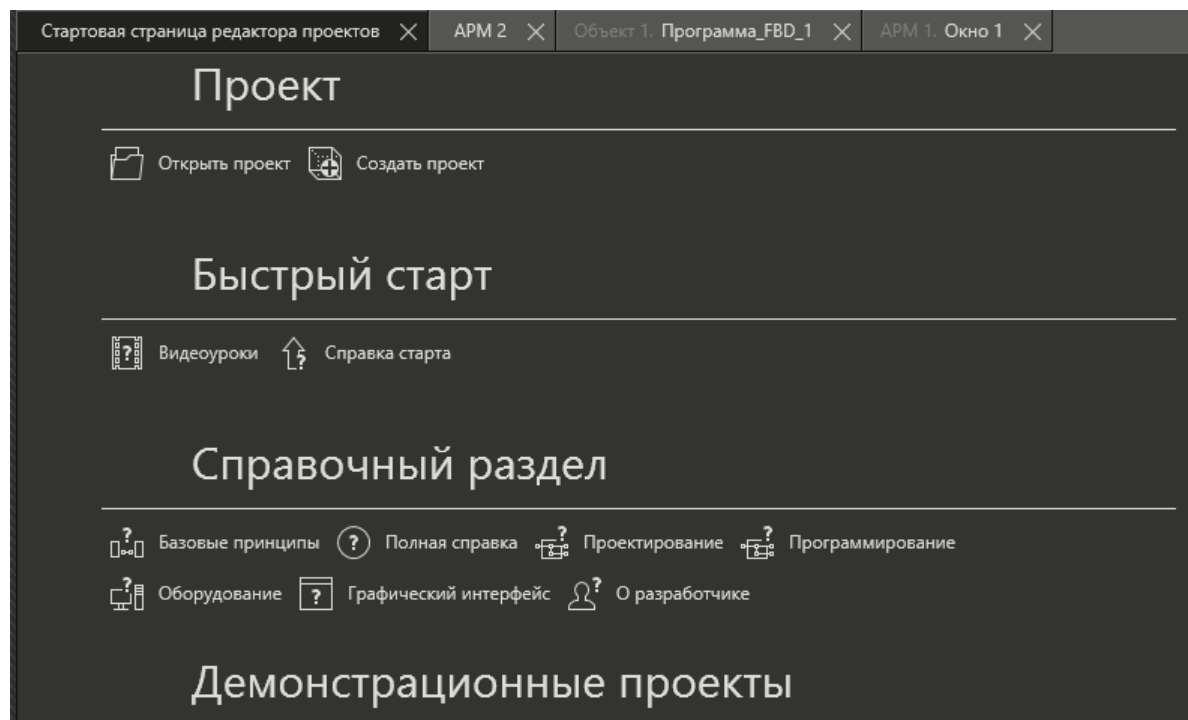
### 5.2.1. Настройки редактора

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**

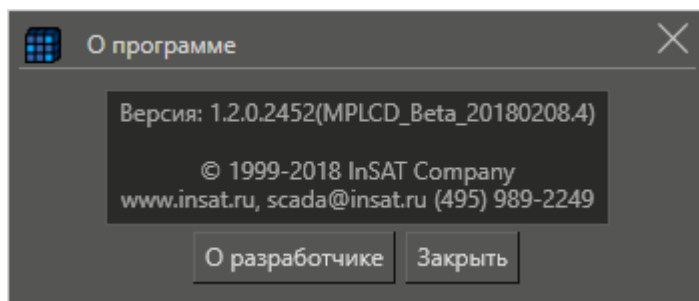
Открывает диалог параметров отчета об ошибках .

### **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 "ИнСАТ";
- Закреть – закрыть диалог.

### Пункт меню Выход

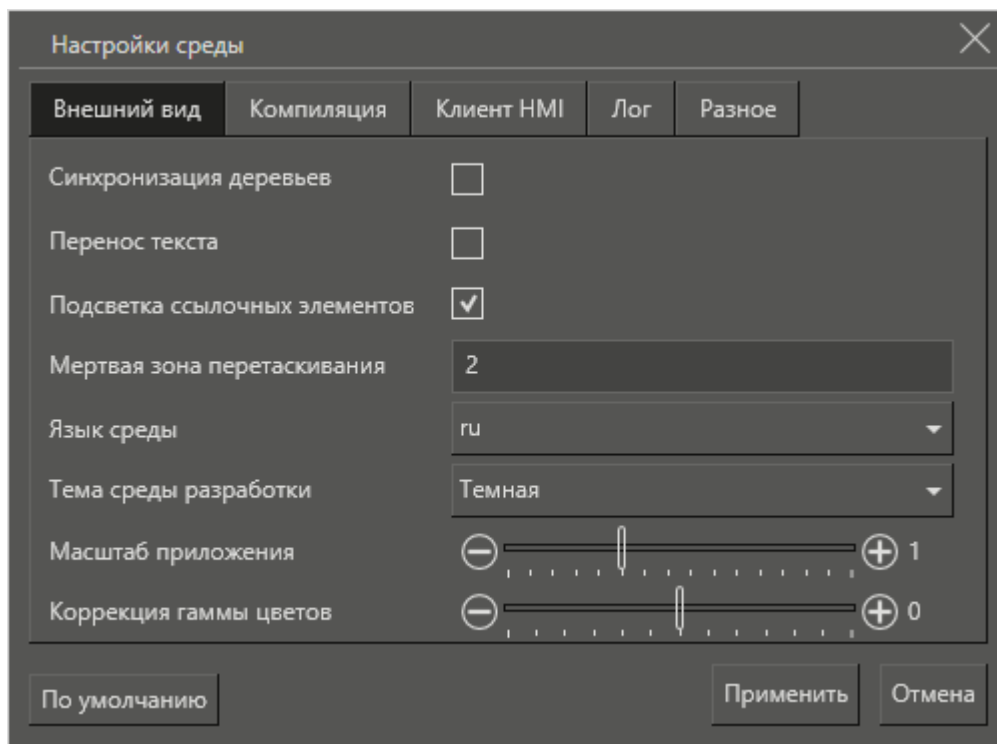
Закрывает текущее окно редактора проекта (аналог инструмента  – см. Типовые инструменты окон ).

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

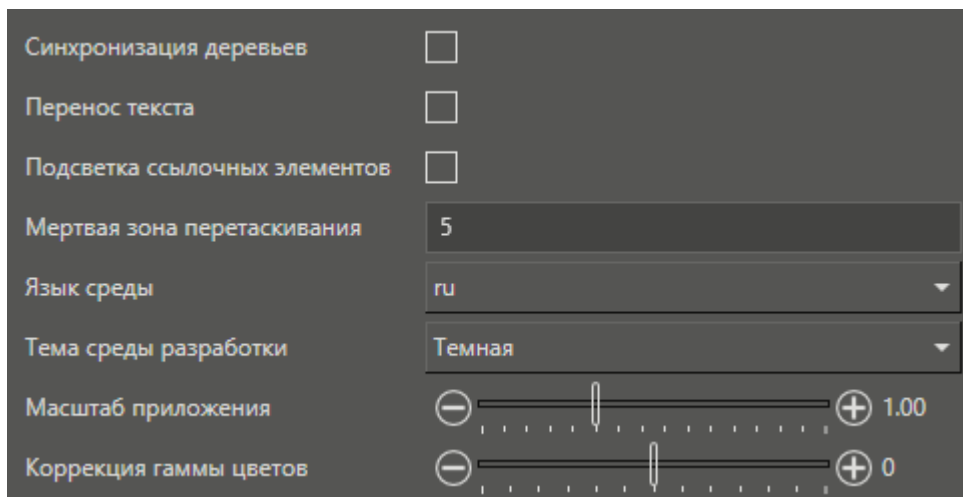
Изменения, сделанные в окне, будут применены после нажатия на кнопку Применить или после закрытия окна (после нажатия на крестик в верхнем правом углу).

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:





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>
Подсветка ссылочных элементов	<p>Если этот флаг установлен, то при наведении курсора на некоторый элемент А разрешена подсветка элементов, имеющих относительно элемента А Ссылку и Внешнюю ссылку .</p>
Мертвая зона перетаскивания	<p>Поле редактирования, в котором задается минимальное значение смещения (в пикселях) для операции перемещения. Если смещение выполняется на меньшую величину, то перемещение не производится.</p>

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

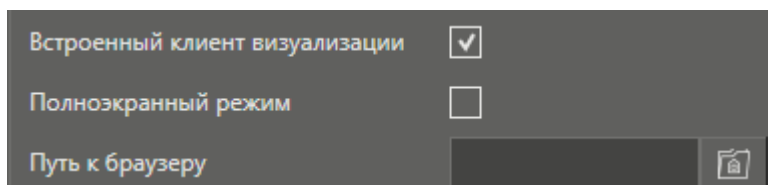
Вид вкладки:

Свойство	Описание
Отладочная конфигурация	Этот флаг следует установить в случае, если требуется вывод всех отладочных значений в редакторах FBD, LD, SFC.
Прерывать выполнение при ошибках	<p>Определяет поведение среды исполнения при возникновении ошибок.</p> <p>По умолчанию флаг снят. В этом случае, если при выполнении программы или ФБ происходит некорректная операция (например деление на 0 и т.п.), то выполнение ФБ/программы, где она произошла прерывается, но выполнение остальных программ в этой задаче продолжается.</p> <p>Если флаг установлен, то возникшая ошибка в программе фиксируется в логе и среда исполнения перезапускается. В логе указывается место возникновения ошибки в скомпилированном проекте.</p>

Тип транслятор НМІ	<p>Определяется, какая версия транслятора НМІ будет использоваться при компиляции окон. Возможные варианты:</p> <ul style="list-style-type: none"> <li>• v1 - при компиляции используется только версия транслятора НМІ v1.</li> <li>• 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 приложения MasterSCADA 4D Monitor и MasterSCADA 4D Client Monitor.</li> <li>• v2 - при компиляции используется только одна версия транслятора НМІ v2.</li> </ul>
--------------------	--

### 5.2.1.1.3. Вкладка Клиент НМІ

Вид вкладки:



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 "ИнСАТ". 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 исполнителями.

тельной 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

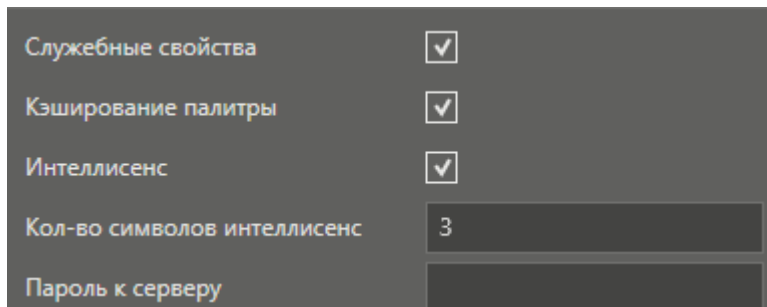
Вид вкладки:

Детализация лога	<input type="checkbox"/>
Проверка корректности состояния проекта	<input checked="" type="checkbox"/>
Журнал связи с узлом	<input checked="" type="checkbox"/>
Лог БД	<input type="checkbox"/>
Лог расхода памяти	<input checked="" type="checkbox"/>
Контроль утечек памяти	<input type="checkbox"/>
Показывать ошибки	<input type="checkbox"/>
Лог браузера	<input checked="" type="checkbox"/>
Количество логов	<input type="text" value="3"/>

**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

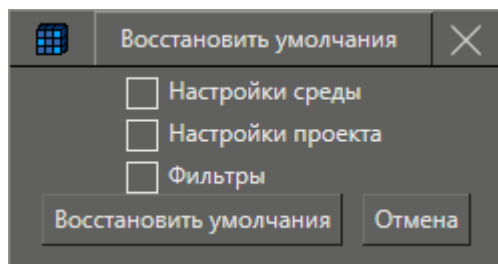
Вид вкладки:



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	Если в среде исполнения задан пароль на загрузку проекта, то загружать проект можно только из той среды разработки, в которой задан такой же пароль.

### 5.2.1.1.6. Диалог Восстановить умолчания

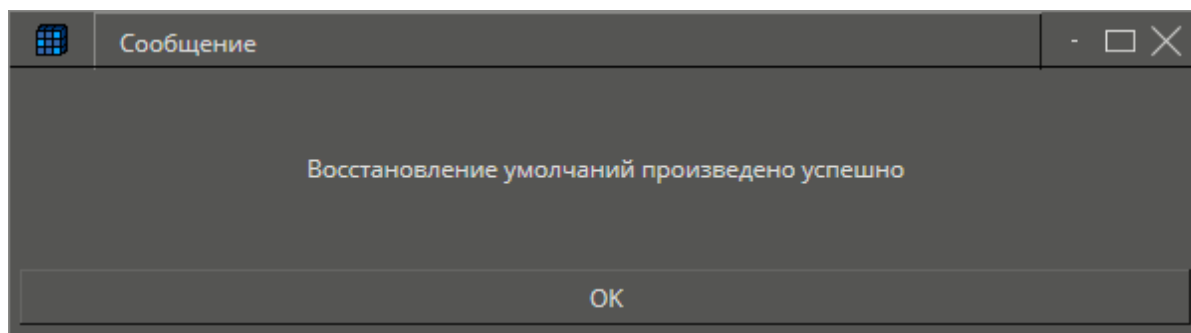
Для того, чтобы открыть диалоговое окно Восстановить умолчания, необходимо нажать на кнопку По умолчанию в диалоговом окне настройки среды.



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\InSAT\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\InSAT\MasterSCADA4D<версия>\ProjectsConfig\<имя проекта>\ProjectUserSettings.xml.

### **Флаг Фильтры**

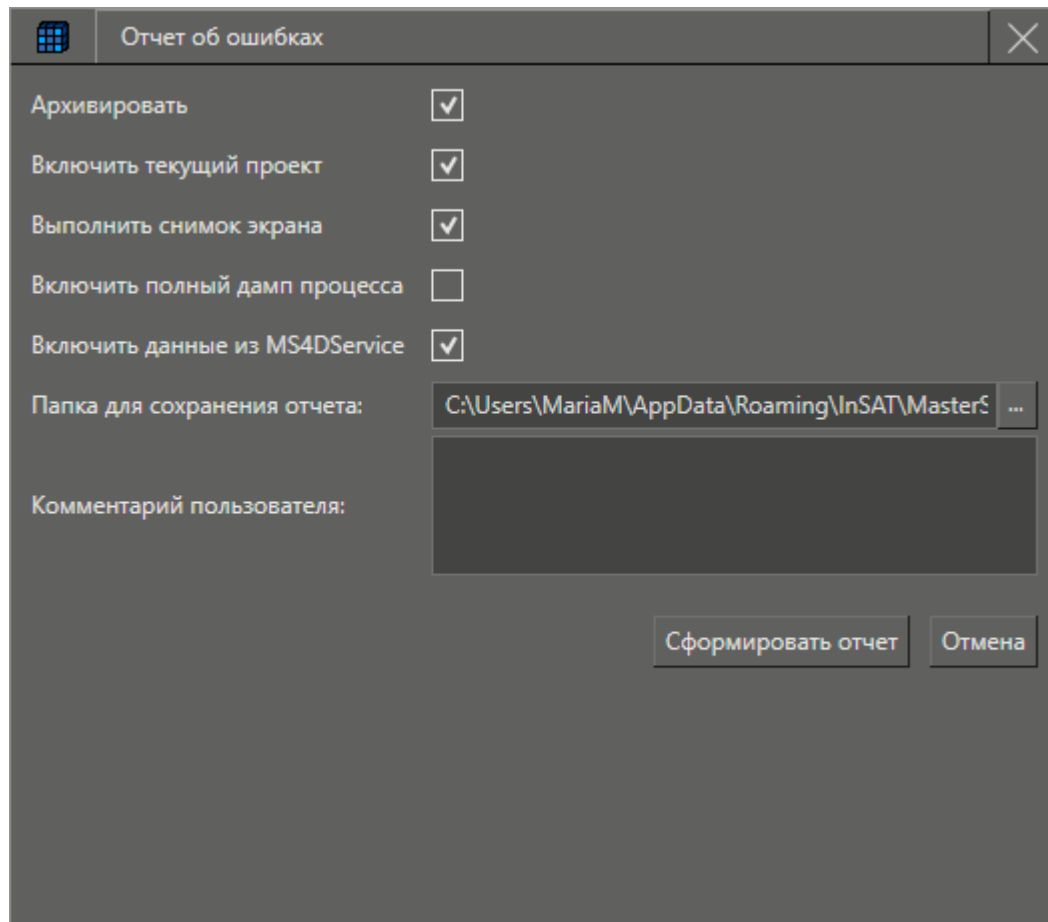
Если этот флаг установлен, то при нажатии на кнопку Восстановить умолчания удаляются пользовательские фильтры.

При восстановлении по этому флагу уничтожаются файлы <папка проекта>\Settings\Filters\InSAT.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.
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: C:\Users\[имя пользователя]\AppData\Roaming\InSAT\MasterSCADA4DBeta\ErrorReports.
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

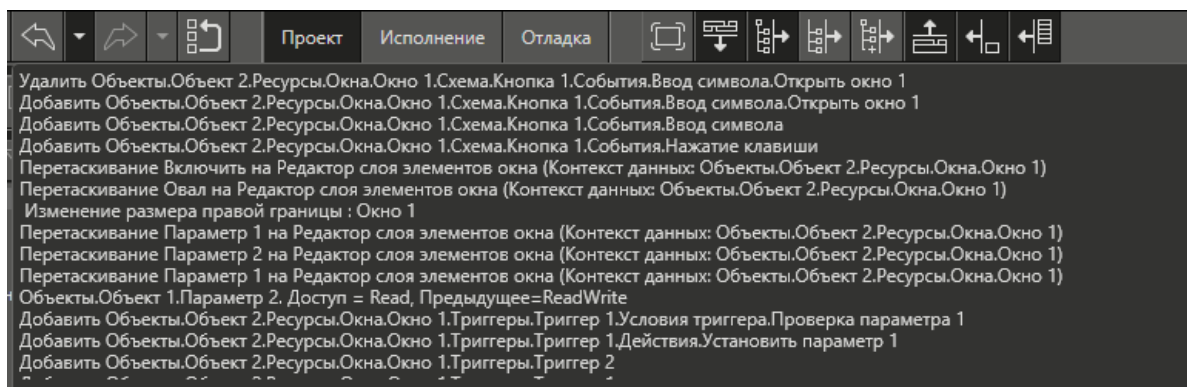
-  – инструменты отмены действий;



-  (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:

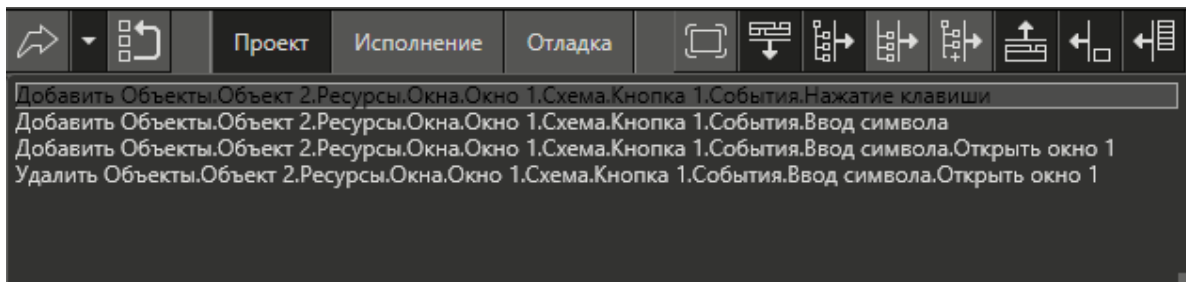


Отменяемое действие может быть макродействием, т.е. содержать несколько элементарных действий.

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

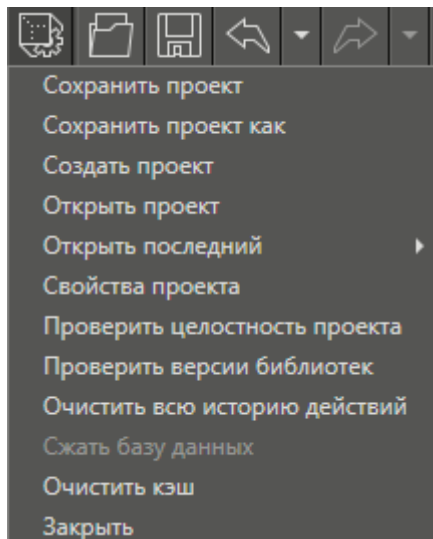
-  – открыть диалог История версий.

### 5.2.2.1. Управление проектами

Группа меню управления проектами открывается при нажатии на кнопку:



Вид меню:

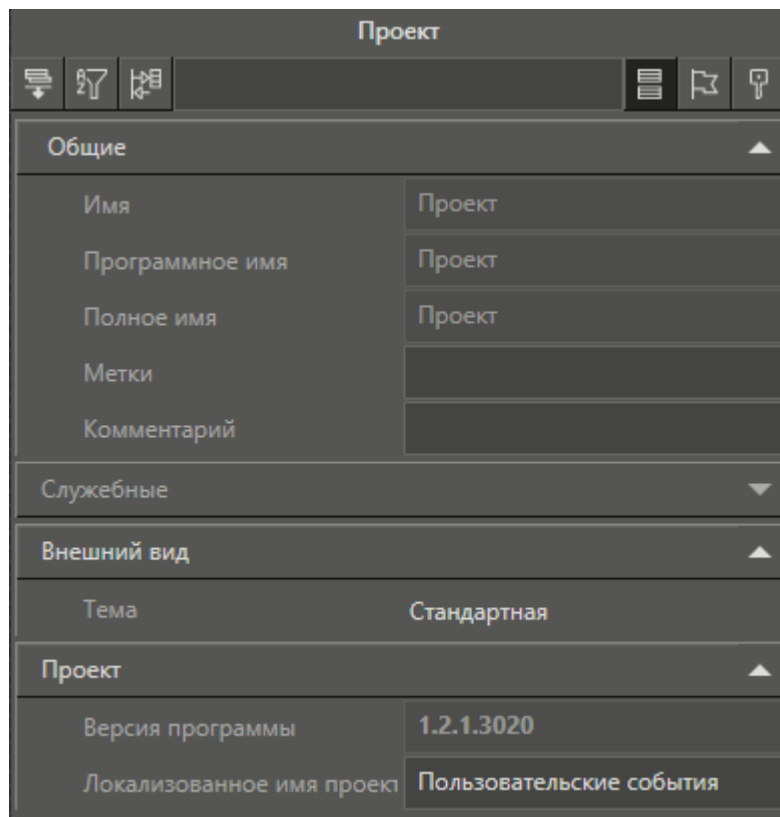


Название	Описание
Сохранить проект	Сохраняет изменения, сделанные в проекте с момента последнего сохранения по текущий момент. После выполнения этой команды теряется возможность отмены и возврата последних действий через панель отмены/возврата действий. После выполнения этой команды процесс среды

	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.
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:



Для проекта можно определить следующие настройки: Тема и Локализованное имя проекта.

### Пункт меню Тема

Определяет тему графических элементов. Тема влияет на фон, тени и цвета элементов.

Например, если сменить тему для учебного проекта Умный дом на Изумруд, то автоматически получится изображение:




Список возможных тем определяется в библиотеке.

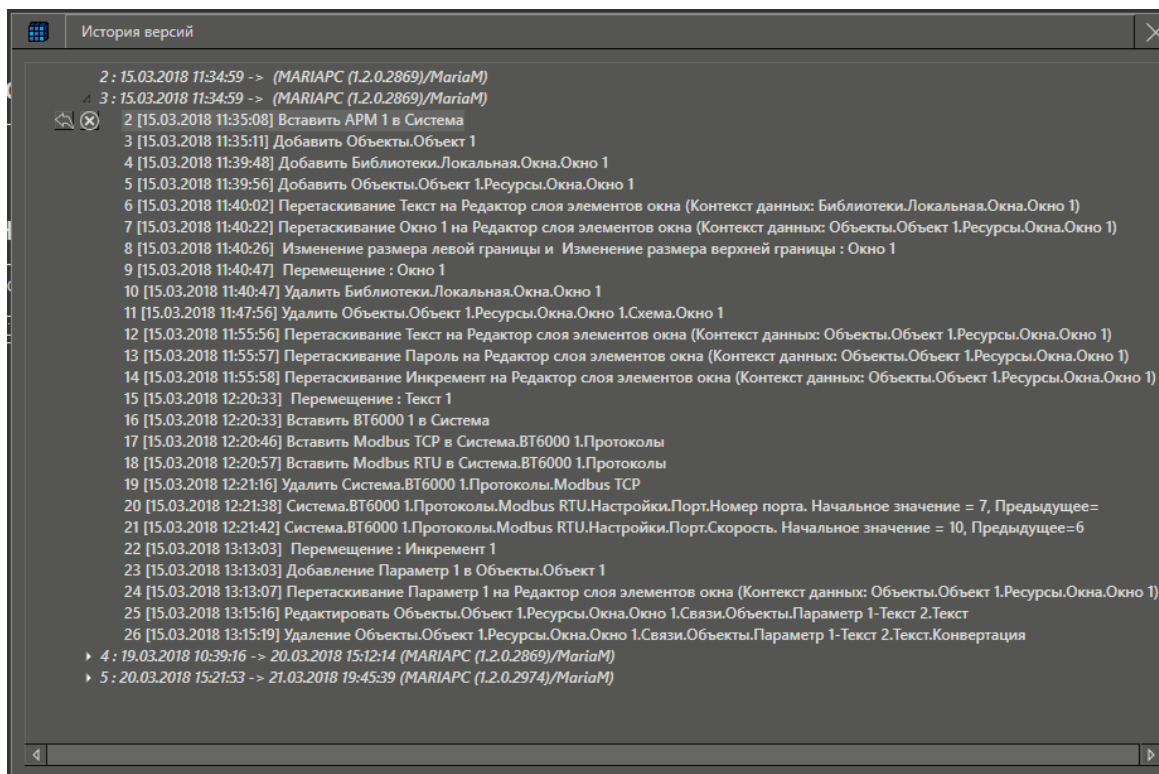
### Пункт меню Локализованное имя проекта

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.

### 5.2.3.История версий

Диалог История версий открывается при нажатии на иконку панели свойств .

В нем отображается история редактирования проекта в виде дерева:



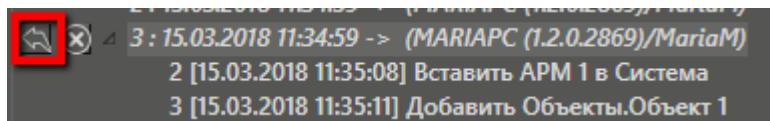
Корневые группы дерева – сессии редактирования проекта – имеют имена <ID сессии>: <время начала сессии> -> <время окончания сессии> (<имя компьютера> (<номер сборки MasterSCADA 4D>)/<имя пользователя>).

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

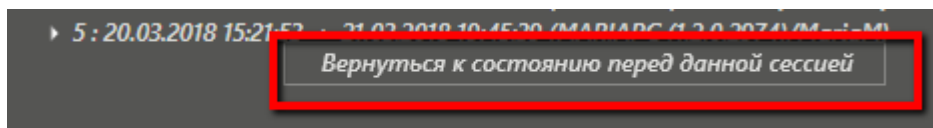
При раскрытии группы действий отображается список элементарных операций, которые выполняются программой в обеспечение этого действия.

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

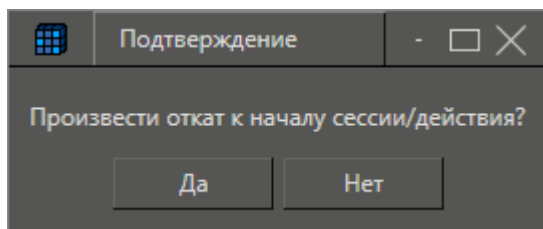
Для возврата к состоянию проекта перед каким-либо действием или перед всей сессией необходимо воспользоваться кнопкой, которая появляется при выделении строки данного диалога.




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.



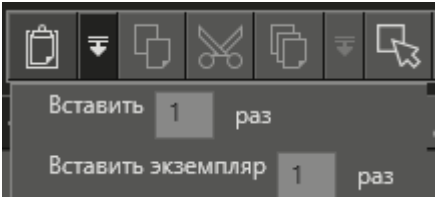

После удаления истории действий рекомендуется выполнить действие Сжать базу данных в меню Управление проектом.







## 5.2.4. 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: 
	Копировать (CTRL+C)	Копирует выделенное в буфер обмена.

	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.

Смотрите также:

Панель инструментов графического редактора

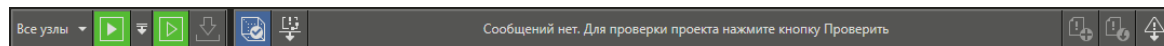
Панель инструментов редактора FBD

Панель инструментов редактора SFC

Панель инструментов редактора LD

## 5.2.5. Вкладка инструментов Исполнение


Вкладка инструментов Исполнение имеет вид:







Вкладка содержит панель Устройства и панель Компиляция:

Панель Устройства



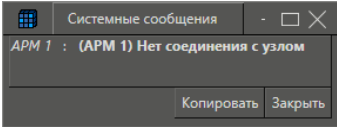




Назначение элементов:

Вид	Описание
	Отображает, в какой узел будет происходить загрузка проекта или производиться имитация проекта, а также позволяет выбрать необходимый узел.



	<p>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.</p> <p>При нажатии на кнопку  можно выбрать способ подключения с загрузкой конфигурации и без загрузки конфигурации.</p>
	<p>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.</p>
	<p>Осуществляет загрузку конфигурации в уже подключенный узел.</p>

### Панель Компиляция

Вид	Описание
	<p>Кнопка проверки конфигурации проекта на ошибки. При проверке происходит компиляция проекта.</p>
	<p>Открывает системное окно сообщений:</p> 
	<p>Поле отображает ошибки, найденные при компиляции проекта.</p>
	<p>В случае затруднений с самостоятельным выявлением причин возникших ошибок, имеется возможность при помощи данной кнопки сформировать отчет об ошибках для последующей отправки его в службу технической поддержки.</p>
	<p>Копирует текст ошибок, возникших при компиляции, в буфер. В дальнейшем его можно вставить в какой-либо текстовый файл для анализа.</p>
	<p>Открывает окно, в котором отображаются все ошибки, возникшие при компиляции.</p>

## 5.2.6. 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.

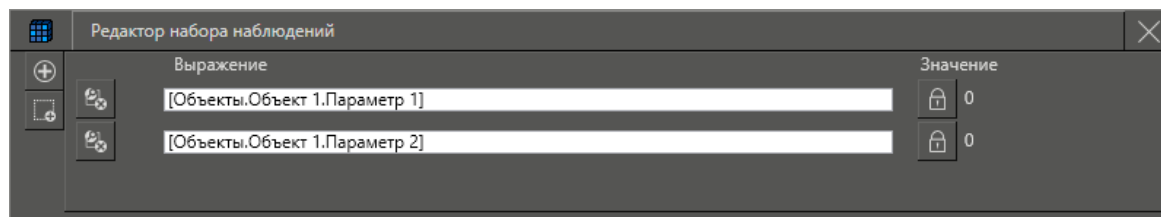


Вид	Назначение
	Если кнопка нажата, то программа перестает выполняться в среде исполнения.
	Запускает исполнение программы в исполнительной системе. Каждая задача узла обрабатывает 1 цикл и останавливается.
	Кнопка блокирует значения выбранных параметров. В этом режиме устанавливается текущее значение, а значения, пришедшие по связи, игнорируются.
	Снимает блокировку с выбранных параметров.
	Открывает диалоговое окно Редактор набора наблюдений.








### 5.2.6.1. 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.

Окно имеет табличный вид:



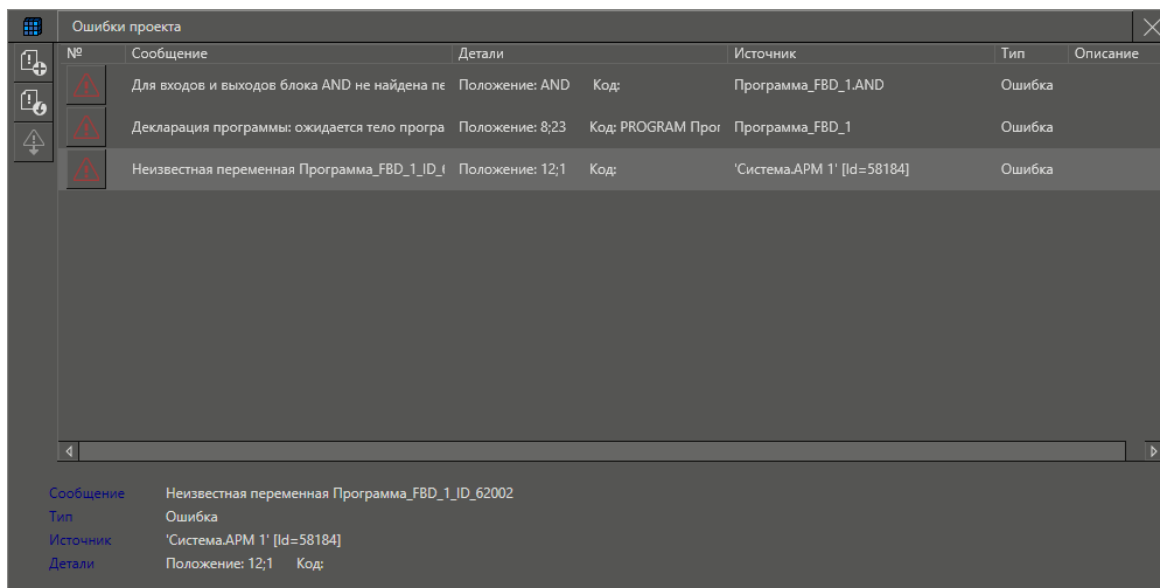
Элементы:

Элемент	Назначение
Столбец Выражение	В данном столбце отображается имя и путь к контролируемому параметру в проекте. Может быть отредактировано вручную.
Столбец Значение	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.
	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: 

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

### 5.2.6.2. 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	<p>В случае ошибки программного элемента столбец содержит ее детальное описание. Место ошибки указывает параметр 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. 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.

## 5.2.8. 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

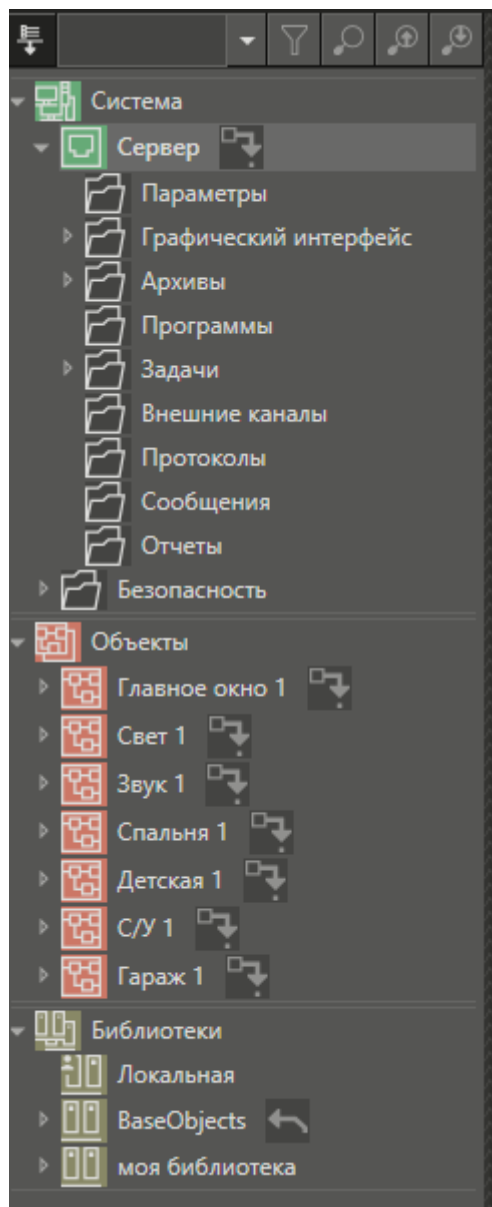
Если режим контекстной справки включен, то курсор имеет вид , и нажатие ЛК на элементе редактора приводит к открытию меню выбора контекстной справки.

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.

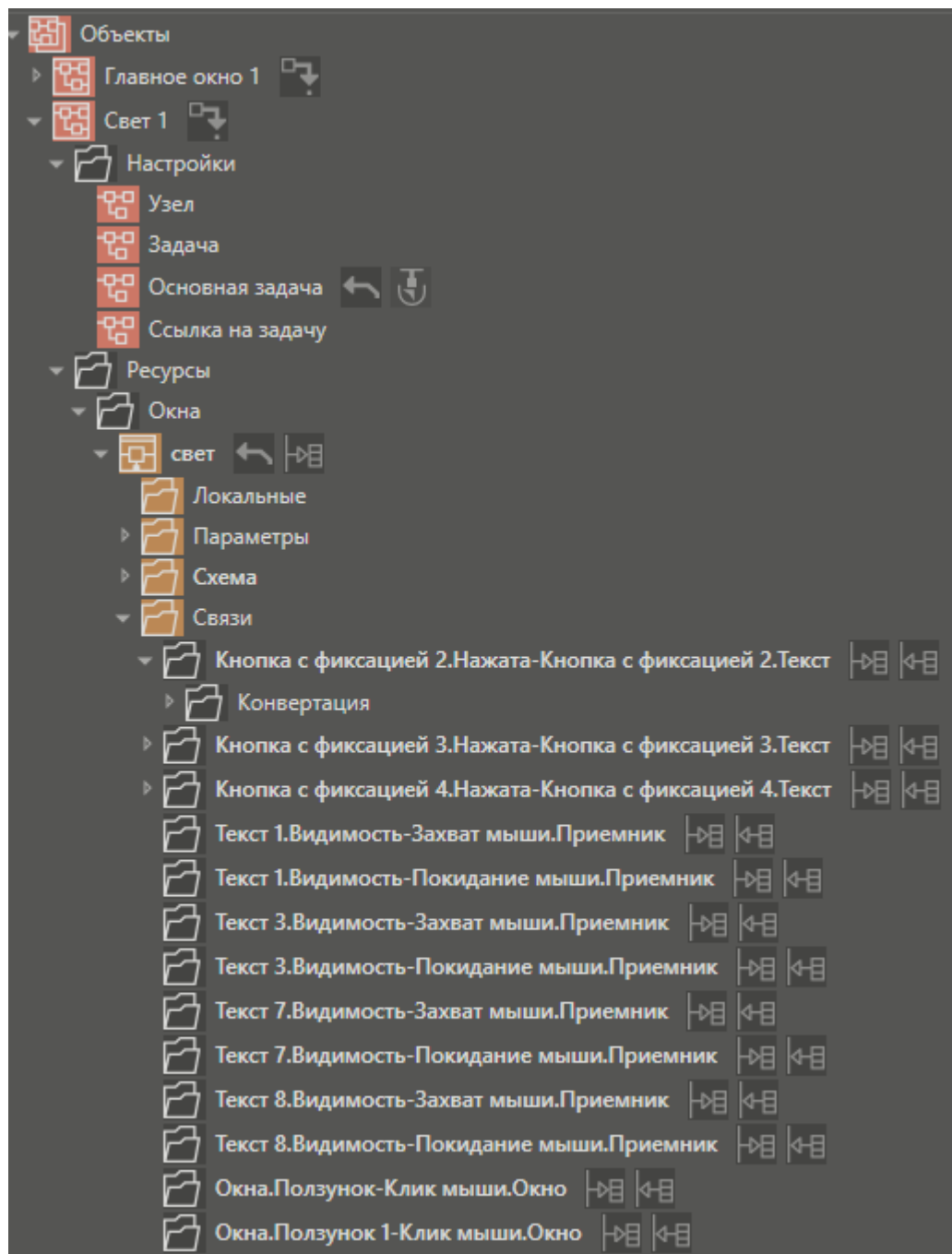
Кроме того, в контекстном меню любого элемента есть пункт Справка, в котором можно выбрать раздел справочной системы, относящийся к выделенному элементу.

### 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.






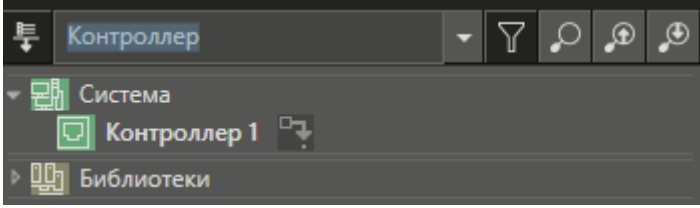



Панель инструментов дерева

В верхней части Древа находится Панель инструментов дерева:



Элементы **Панели инструментов**:



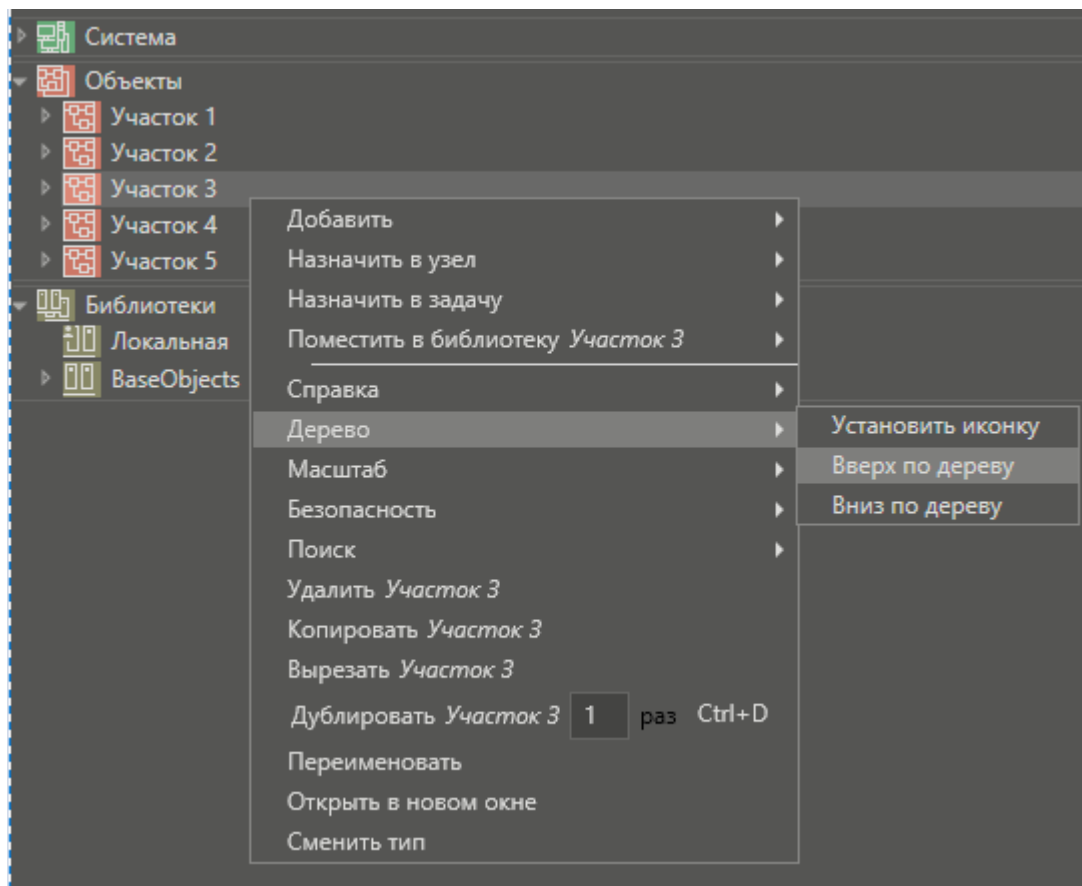
Вид	Описание
	<p>Кнопка управления видимостью элементов в Дереве. Если кнопка нажата, то Деревья отображаются в упрощенном виде. Это означает, что в них присутствуют только те элементы, которые разработчик проекта создает явно, либо те элементы программ и окон, которые в дальнейшем будет логично связать 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.</p>
	<p>Включает фильтр Древа. 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.</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 SHIFT+↓, SHIFT+↑, 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.

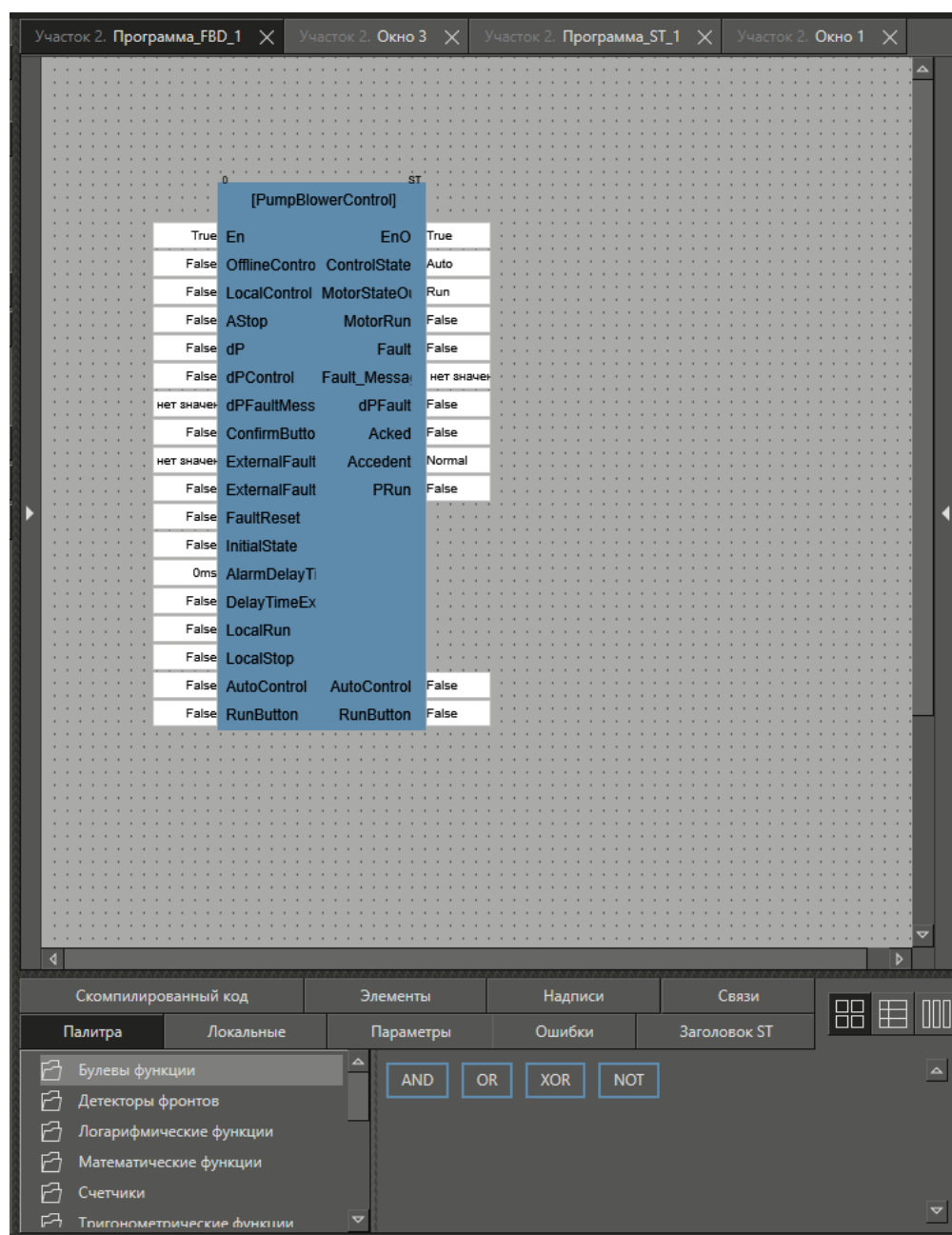
При помощи команды контекстного меню Дерево-Удалить Иконку восстанавливаются иконки по умолчанию для одного элемента или выделенной группы элементов.

Ветки дерева


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

## 5.4. Панель редакторов

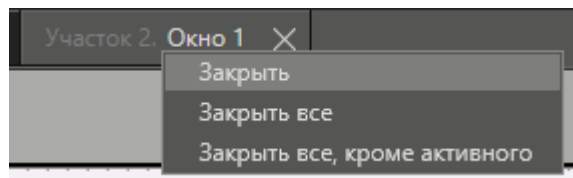
Редакторы программ и окон открываются в центральной части интерфейса как вкладки:



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

При работе с проектом обращайте 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:

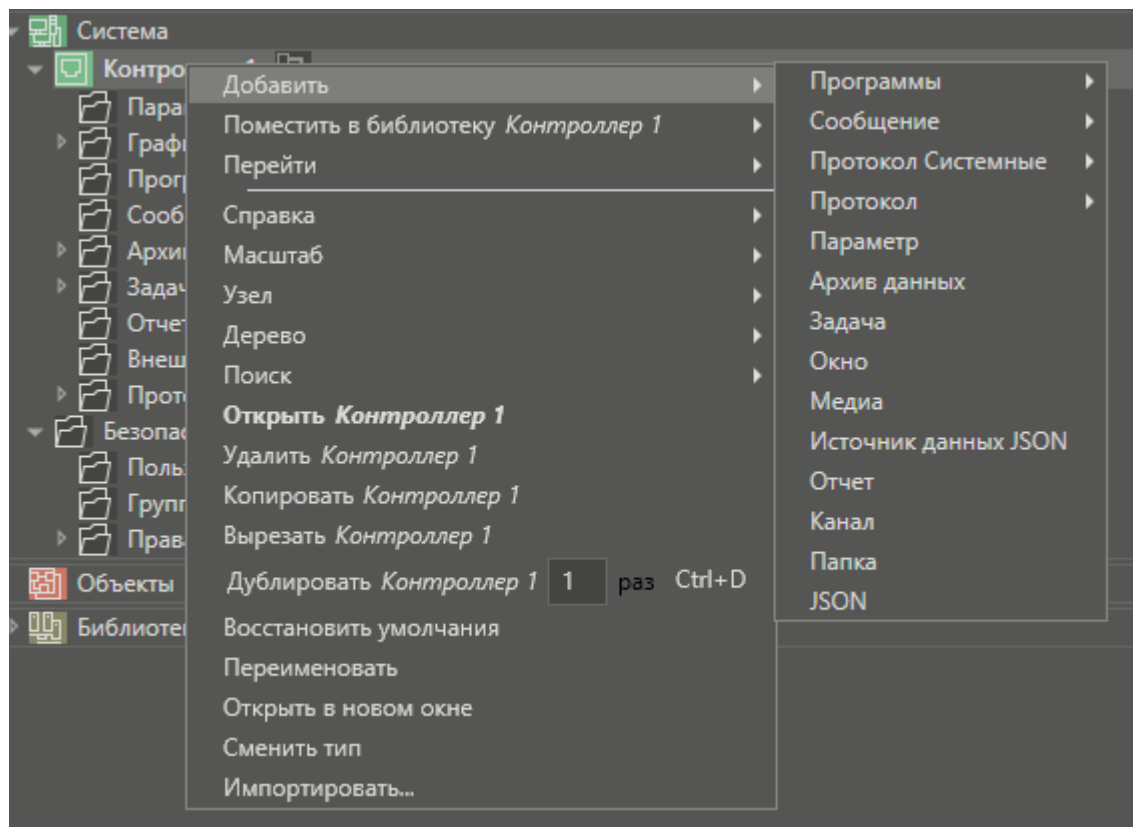


## 5.5. Контекстное меню

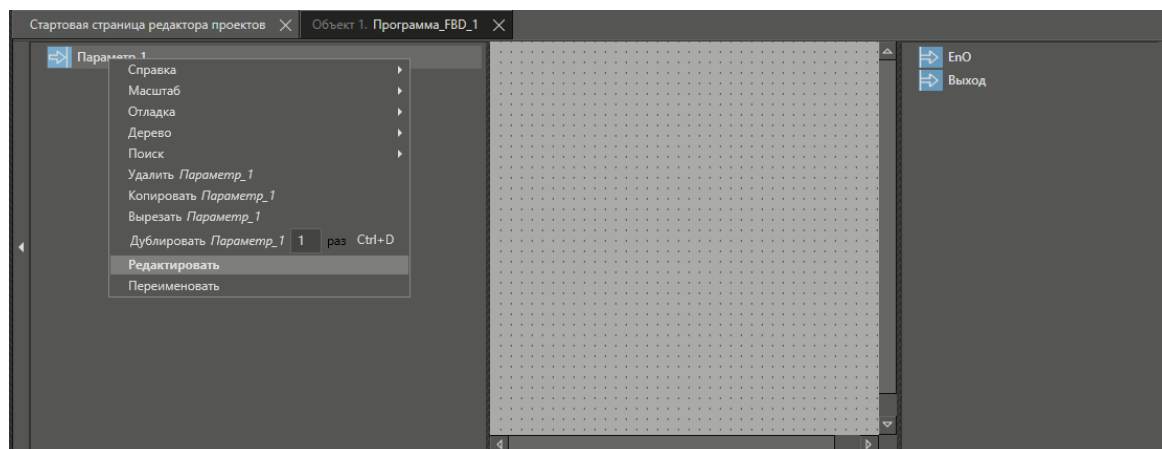
Контекстное меню – это инструмент программы, представляющий собой список возможных команд для выбранного элемента.

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.

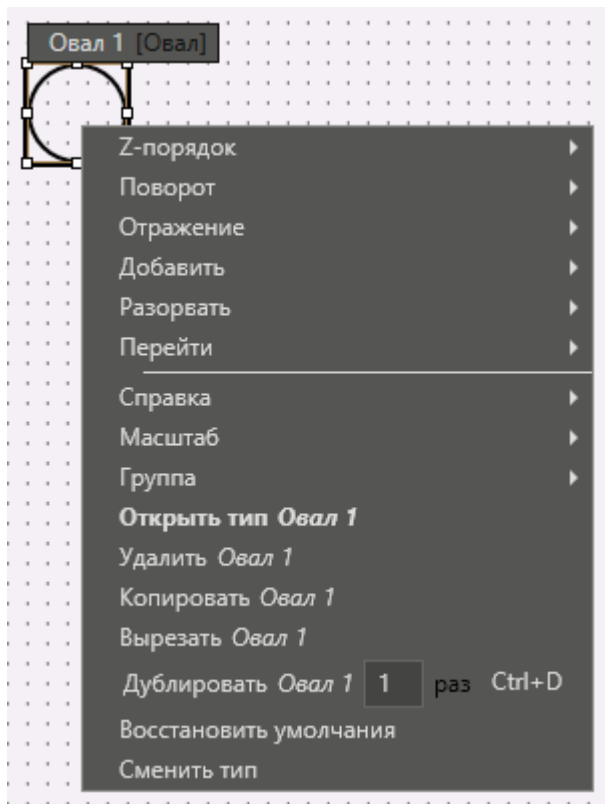
Контекстное меню узла имеет вид:



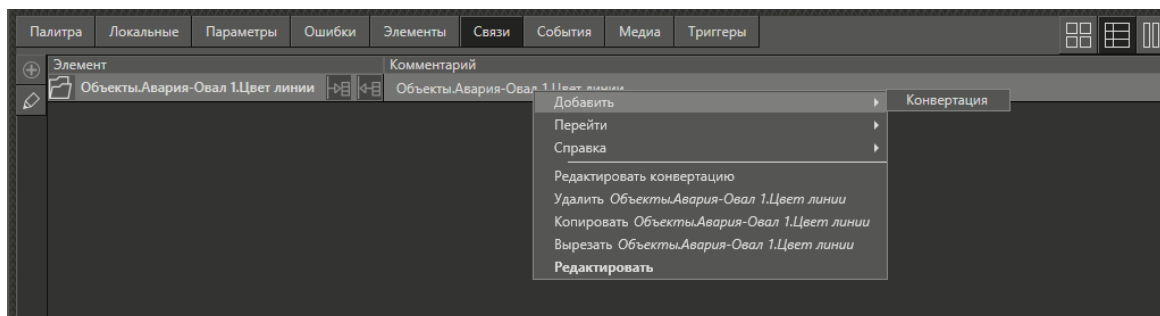
Контекстное меню клеммника программы FBD имеет вид:



Контекстное меню графического элемента имеет вид:



Контекстное меню связи в Легенде графического редактора имеет вид:

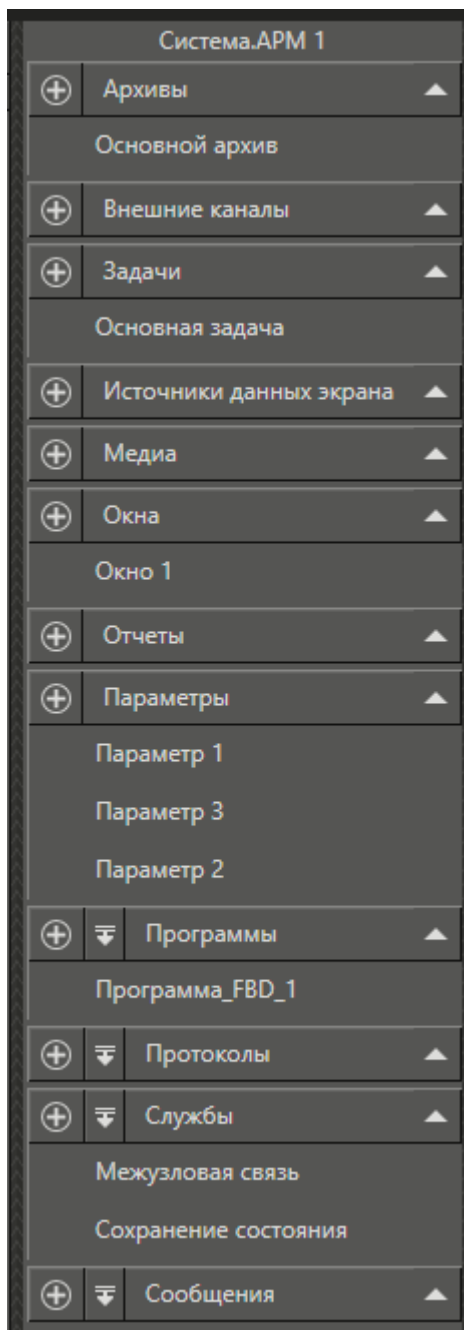


## 5.6. Контекстная панель

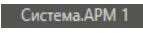

Контекстная панель находится справа от дерева MasterSCADA и обеспечивает альтернативный способ добавления элементов в проект. Контекстная панель открывается при выделении элемента в дереве. Если элемент не имеет собственной контекстной панели, то откроется контекстная панель родительского элемента.


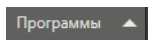



Собственную контекстную панель имеют следующие элементы: дерево системы, дерево объектов, дерево библиотек, Узел, Объект, Библиотека.

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 .

	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):



APM 1

Общие

Имя	<b>APM 1</b>
Программное имя	<b>APM 1</b>
Полное имя	Система.APM 1
Метки	
Комментарий	

Служебные

Задачи

Конфигурация

Наличие визуализации	<input checked="" type="checkbox"/>
Параметры запуска RT	

Отношения

Унаследован от	Библиотеки.Стандартная.Узлы.APM
Внешние ссылки	Объекты.Объект 1.Настройки.Узел

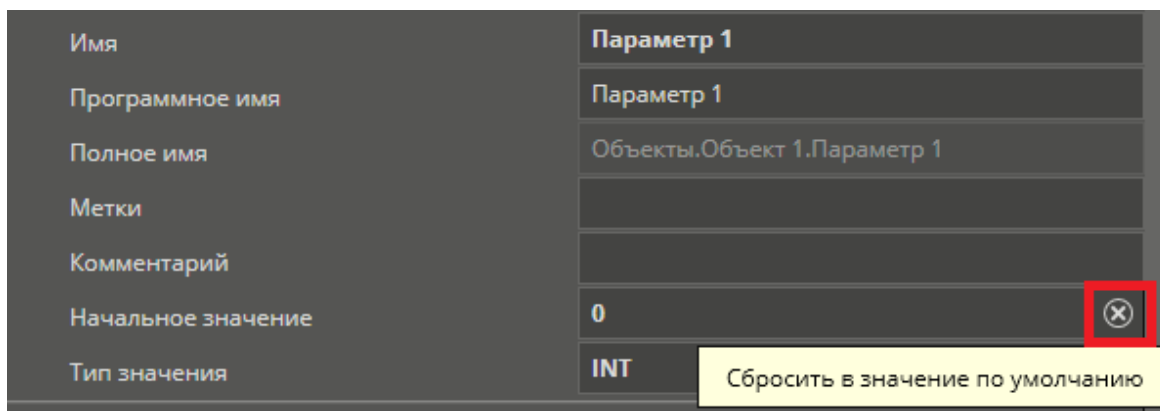
Связь

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	<b>_3_2_1_0_7_6_5_4</b>
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.APM 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 строике свойства:



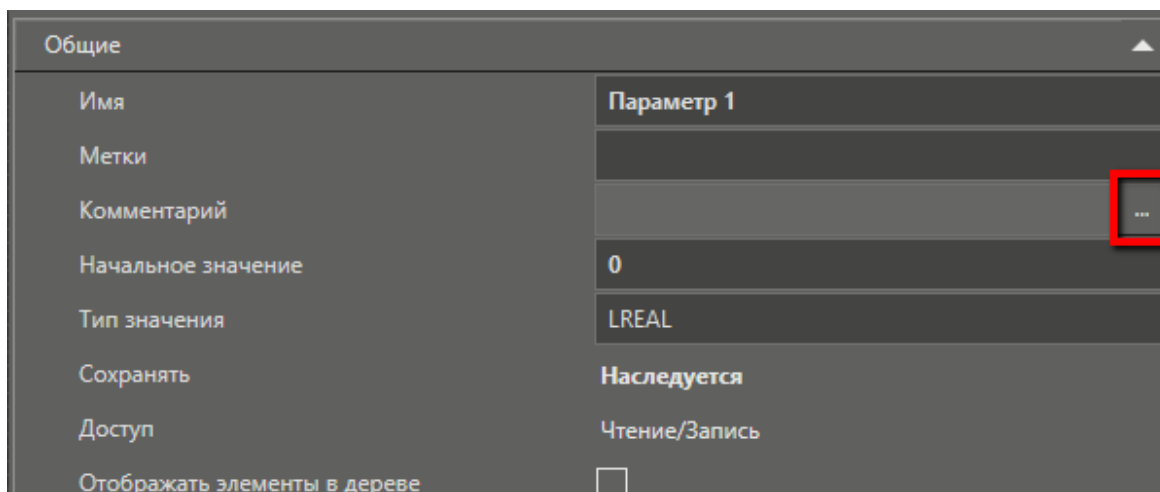
Свойство по умолчанию графических элементов отображается жирным шрифтом.

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.

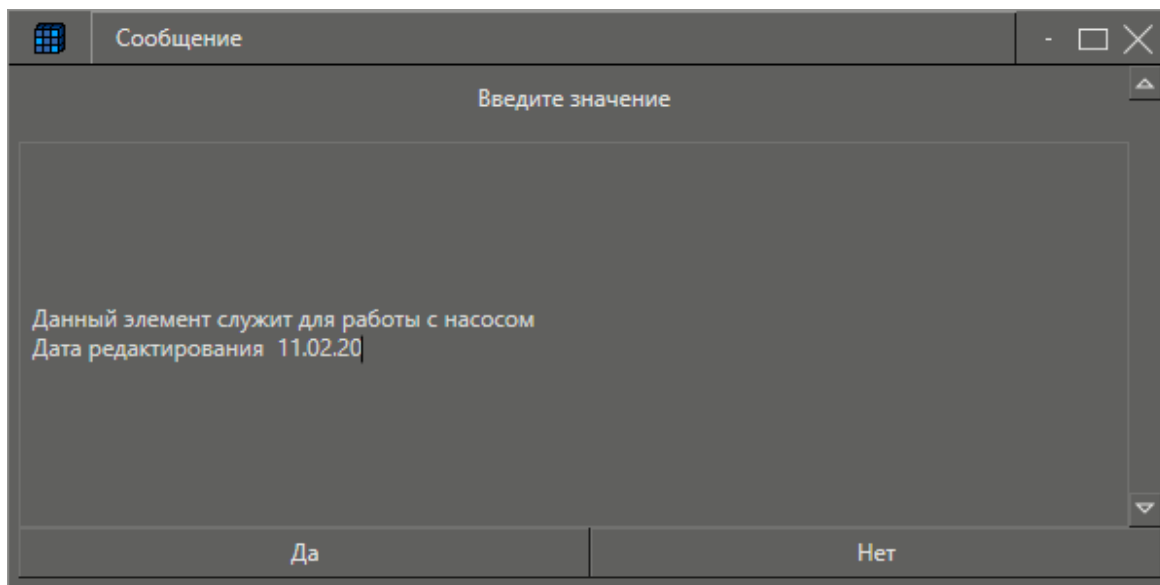
Для открытия диалогового окна ввода значения необходимо нажать на кнопку:



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:

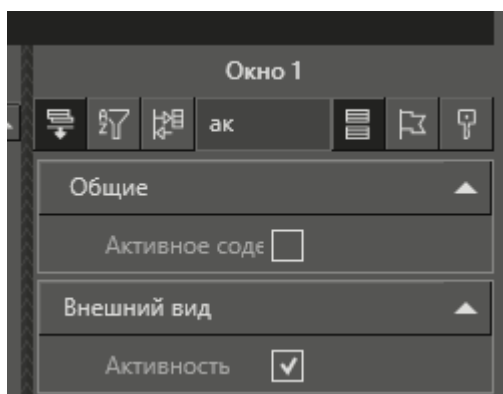


Инструменты панели:


-  – 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).

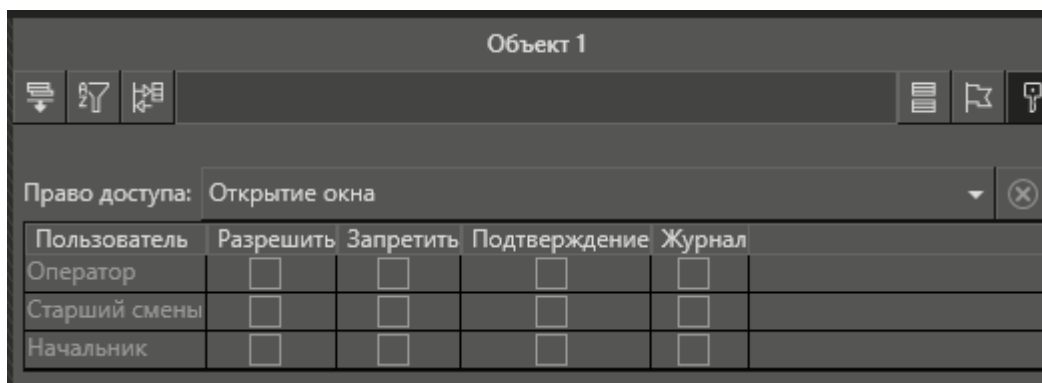
**Важно!** Изображения Панели свойств в справочной системе соответствуют отжатому состоянию кнопок.


-  – 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

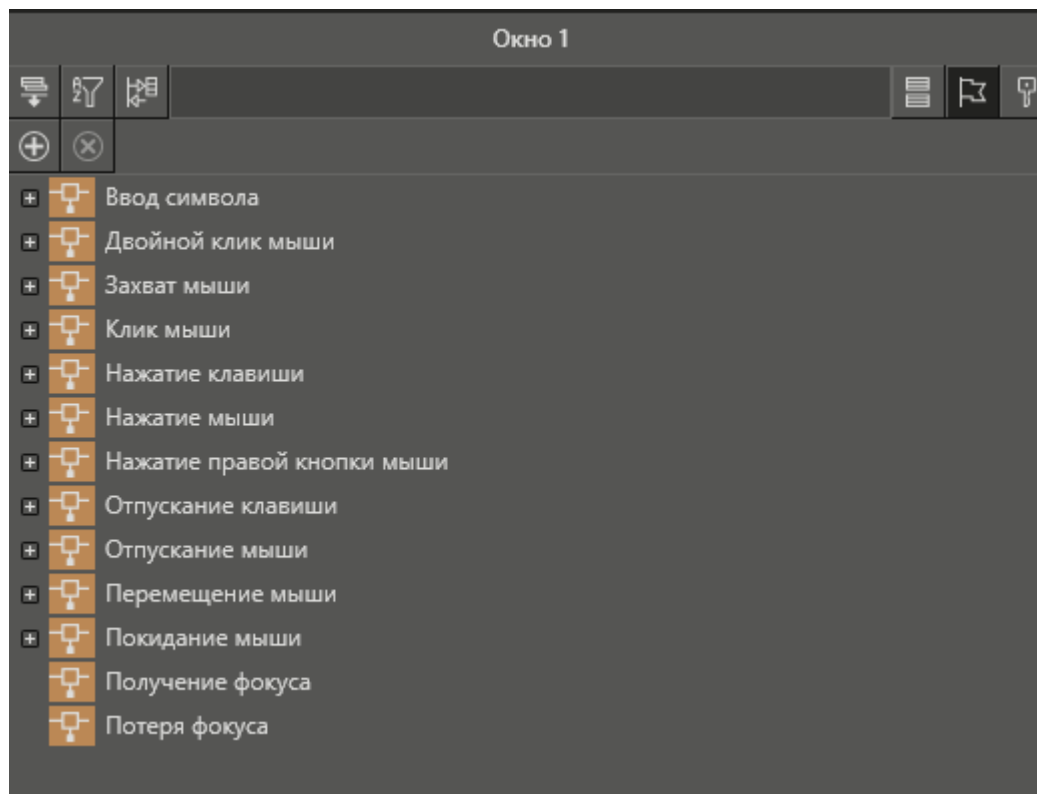
Если нажата кнопка , то Панель свойств позволяет задать права пользователей проекта для выделенного элемента проекта (см. Задание прав пользователей ).




В данном случае кнопка  и инструменты поиска и фильтра недоступны.

### 5.7.2. Панель свойств. События

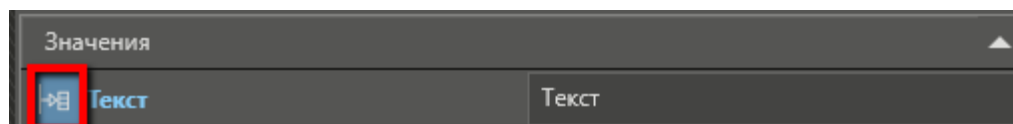
Если нажата кнопка , то Панель свойств представляет собой редактор событий выделенного графического окна или графического элемента – см. НМІ. События :





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):



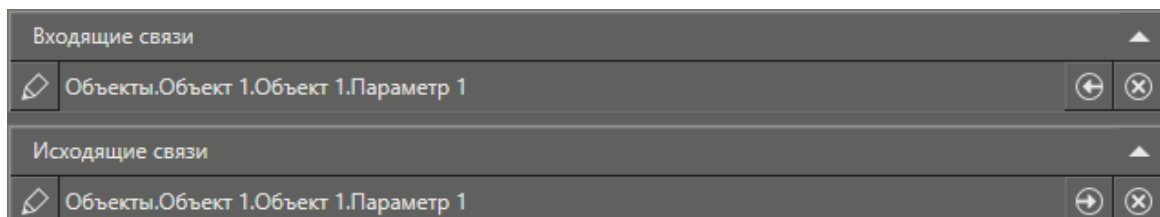
Внешний вид кнопки зависит от направления связи:

-  – 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:






Индикатор

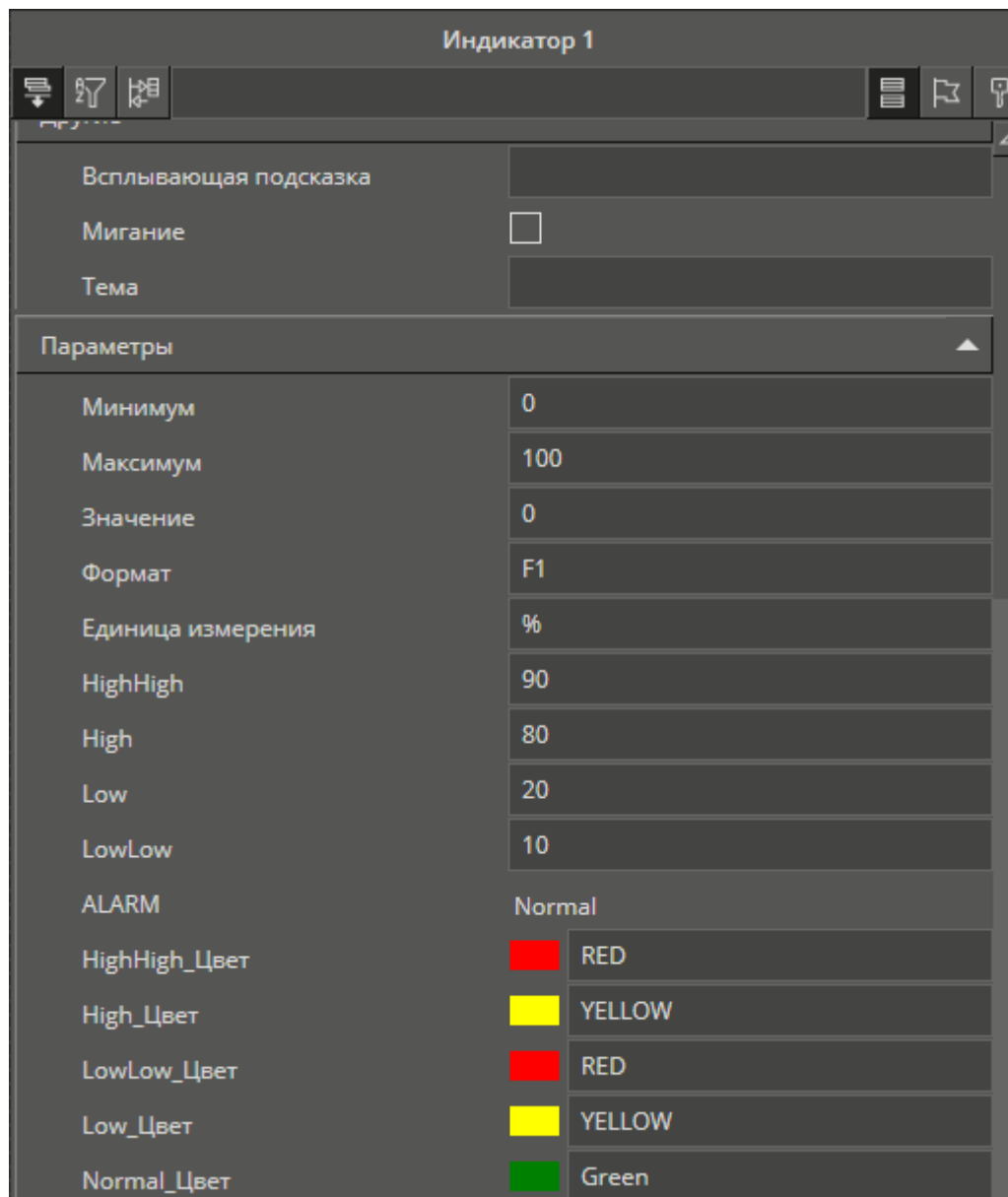
Всплывающая подсказка

Мигание

Тема

Параметры

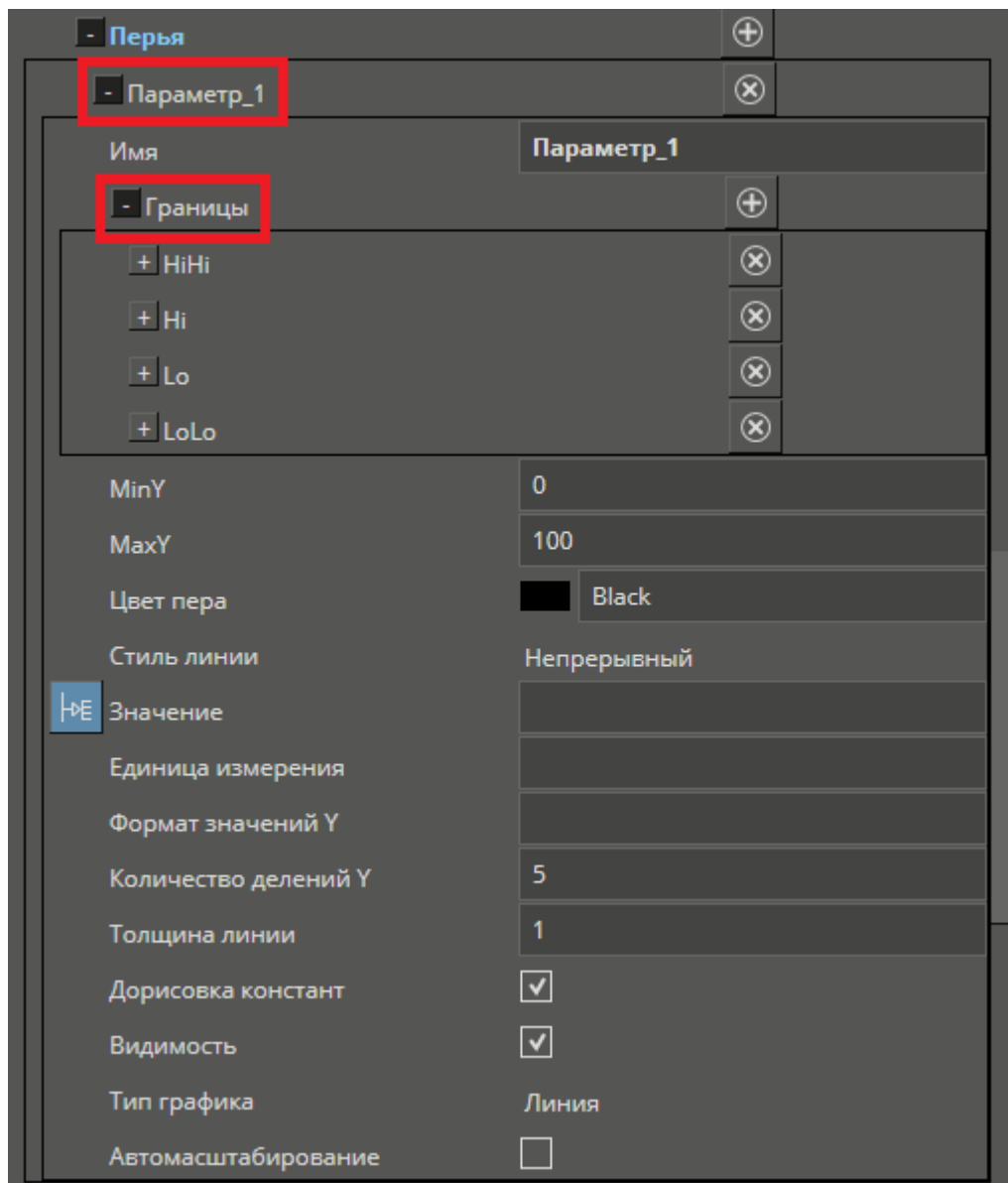
Минимум	0
Максимум	100
Значение	0
Формат	F1
Единица измерения	%
HighHigh	90
High	80
Low	20
LowLow	10
ALARM	Normal
HighHigh_Цвет	 RED
High_Цвет	 YELLOW
LowLow_Цвет	 RED
Low_Цвет	 YELLOW
Normal_Цвет	 Green



### 5.7.5. Панель свойств. Отображение свойств, имеющих структуру

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





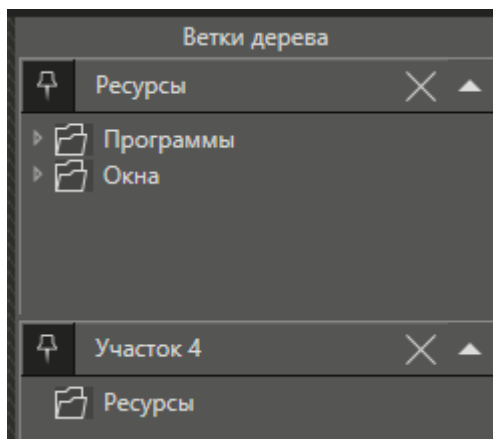
Чтобы раскрыть/свернуть структуру параметра/свойства, необходимо использовать кнопку **+** / **-**, расположенную слева от имени параметра/свойства.

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

### 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 .

## 5.10.2. Элементарные инструменты

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;

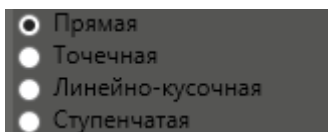


– нажатое (утопленное) положение.

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.

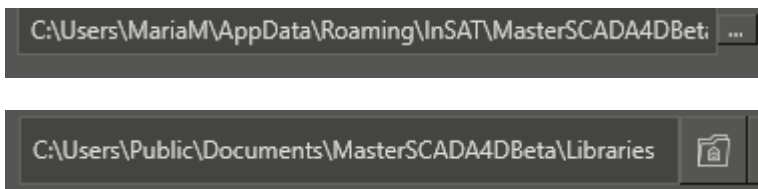
Типовые операции в текстовом редакторе:

- 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

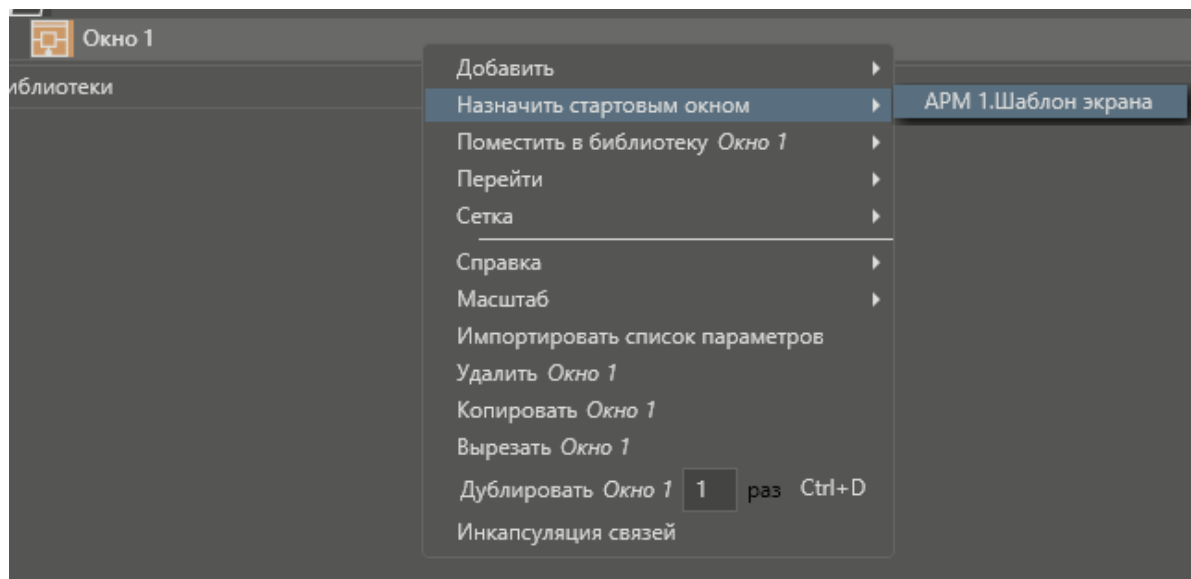
Контекстное меню – это список действий/команд, которые могут быть выполнены над объектом (группой объектов), если такие действия/команды определены для этого объекта (группы объектов).

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:



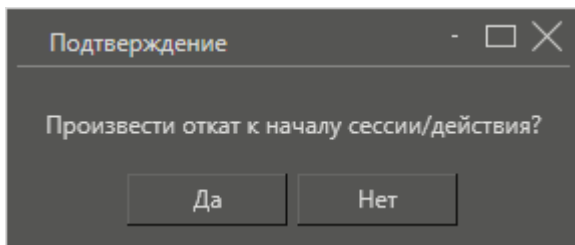
Для того, чтобы закрыть контекстное меню, необходимо нажать на любое свободное место редактора или нажать клавишу Esc. Если были открыты подменю, то однократное нажатие клавиши Esc последовательно закрывает подменю, начиная с самого нижнего уровня, а при всех закрытых подменю закрывает само меню.

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:

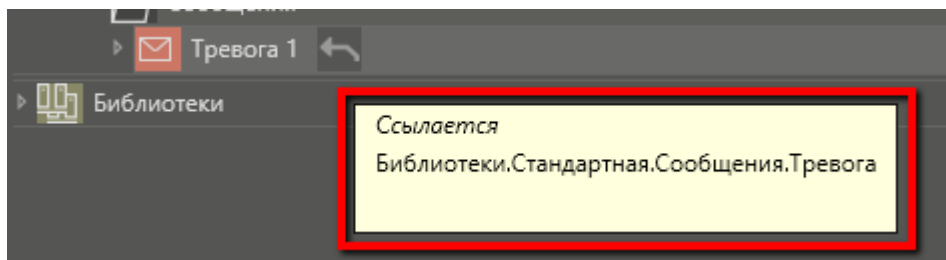


Инструменты диалога:

- 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

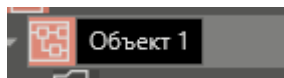
Открытие элемента в редакторе

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.

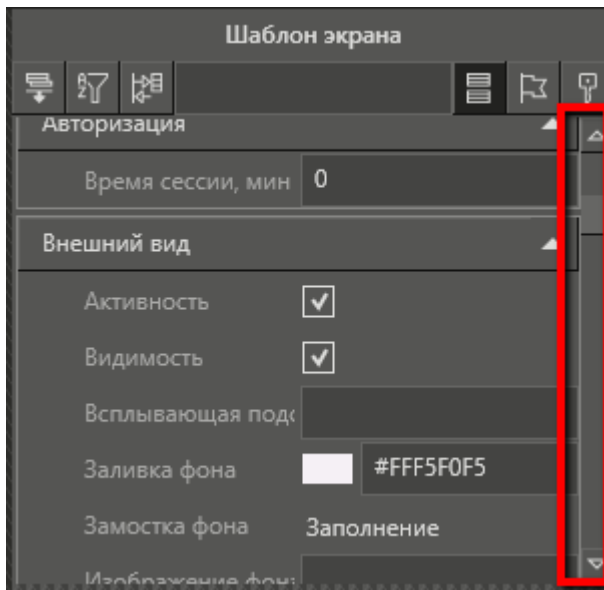
По этой команде на месте значения открывается инструмент его задания (выпадающий список, поле редактирования и т.п.). В поле редактирования текущее значение выделено:



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.

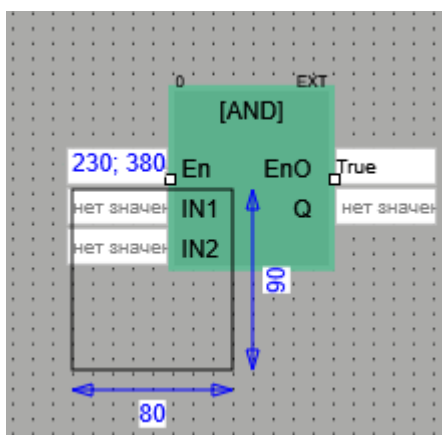
Кроме того, для прокрутки FBD-, LD- и SFC-диаграмм предусмотрены специфические средства (см. Прокрутка диаграмм).

### 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

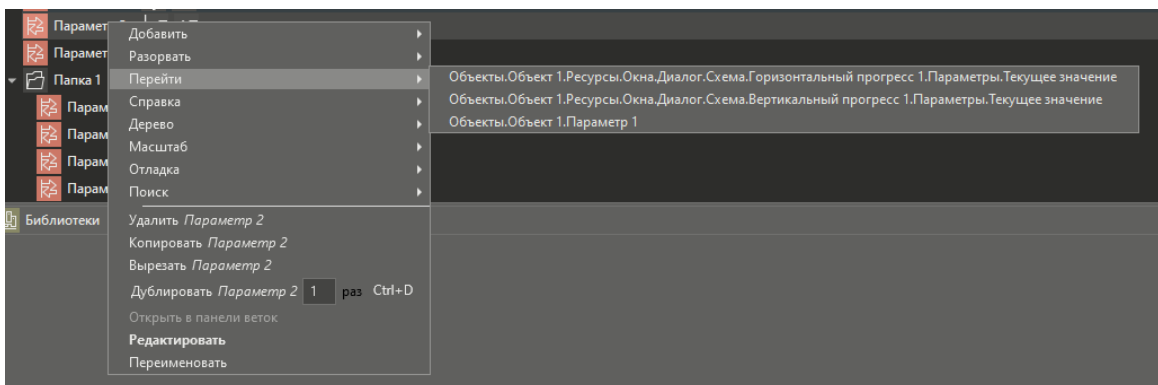
Для изменения размеров выделенного объекта необходимо установить курсор в вершину прямоугольника, ограничивающего объект (курсор при этом принимает вид  или ), нажать ЛК и, удерживая кнопку нажатой, переместить курсор в нужную точку, после чего кнопку мыши отпустить. При одновременном с этим удержанием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

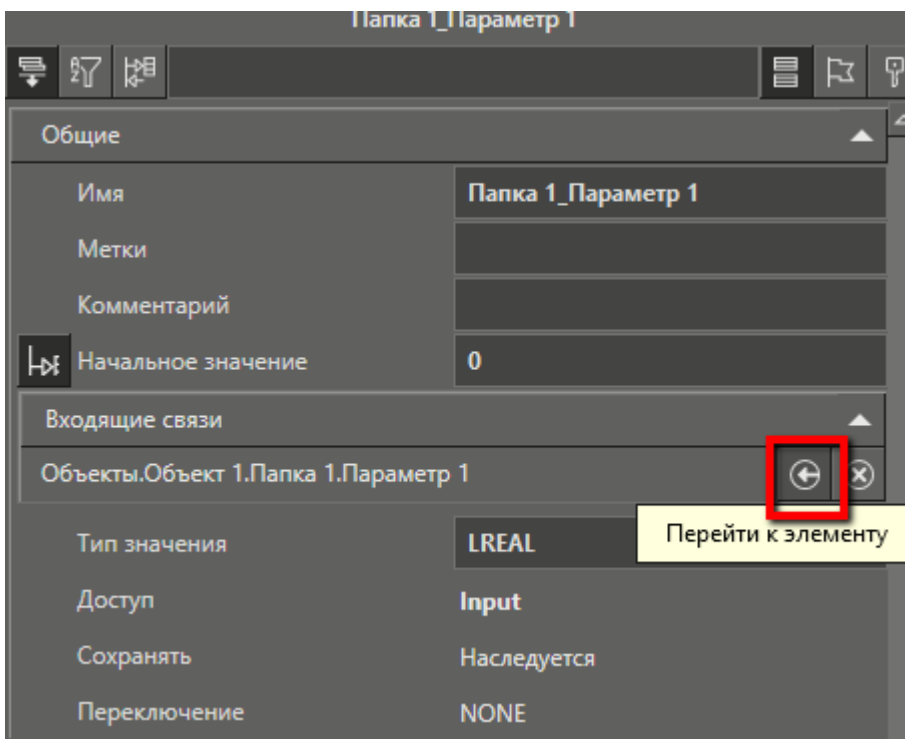
Таким же способом можно изменять размеры окон и диалогов.

### 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:



### 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.

## 5.11. Операции в редакторе проекта

### 5.11.1. Операции с проектами

#### 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

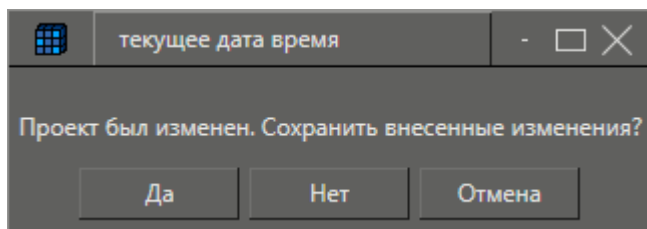
Важно! При сохранении проекта очищаются стеки отмены/возврата действий (стеки UnDo/ReDo). Откатить изменения можно, используя кнопку панели инструментов История версий.

#### Закрытие проекта

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

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 :

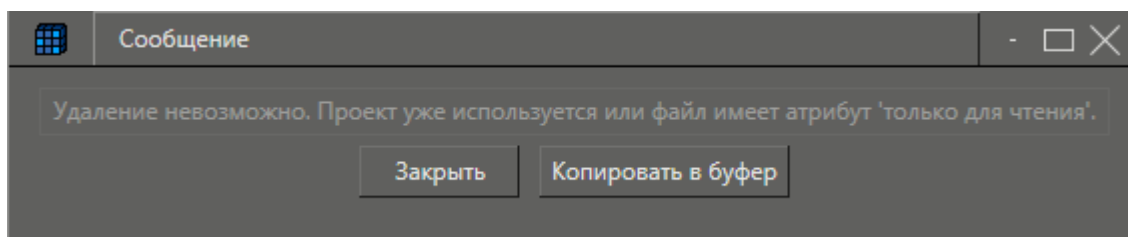


После закрытия проекта открывается стартовое меню .

### 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. Диалог открытия проекта

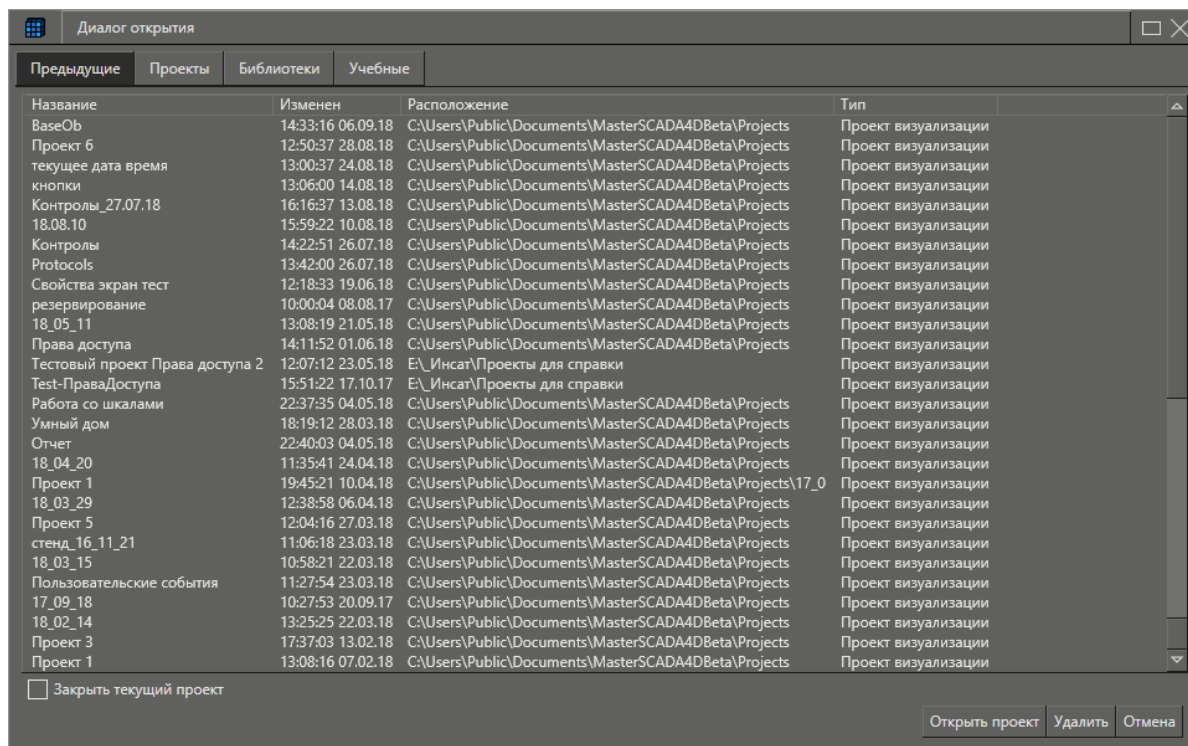
С точки зрения файловой системы, проект представляет собой папку, содержащую файлы (в т.ч. файл БД \*.fdb) и вложенные папки. Переименовывать эти элементы средствами ОС не рекомендуется. Для переименования проекта следует использовать инструмент Сохранить как ).

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.

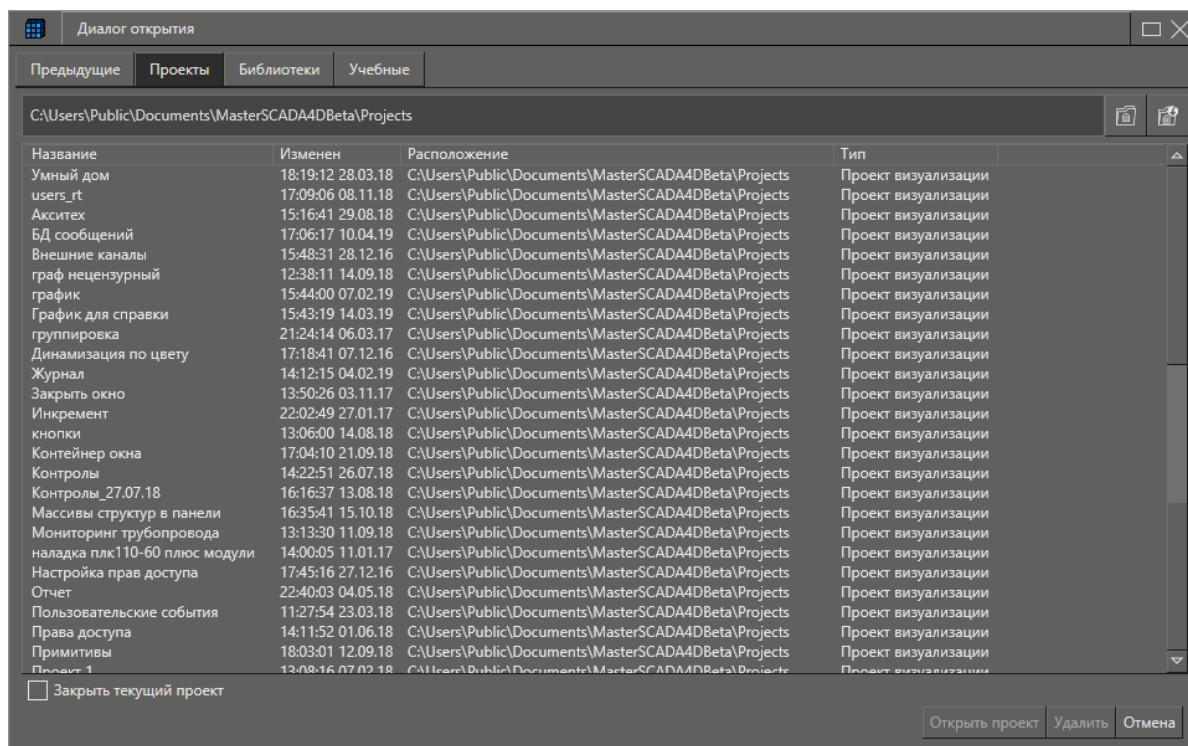
Вкладка Предыдущие





На этой вкладке отображается таблица параметров последних открытых проектов.

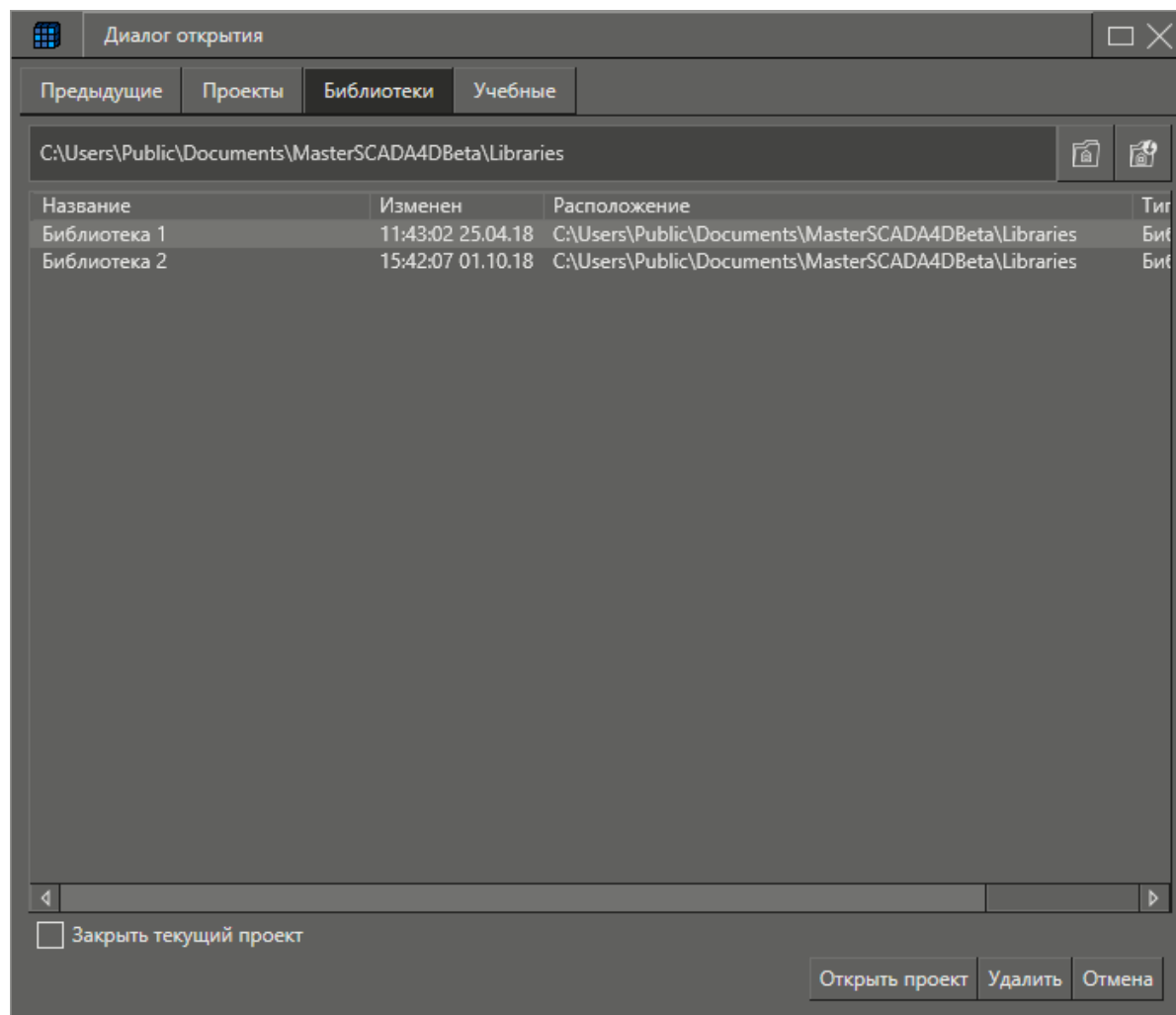
## Вкладка Проекты



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

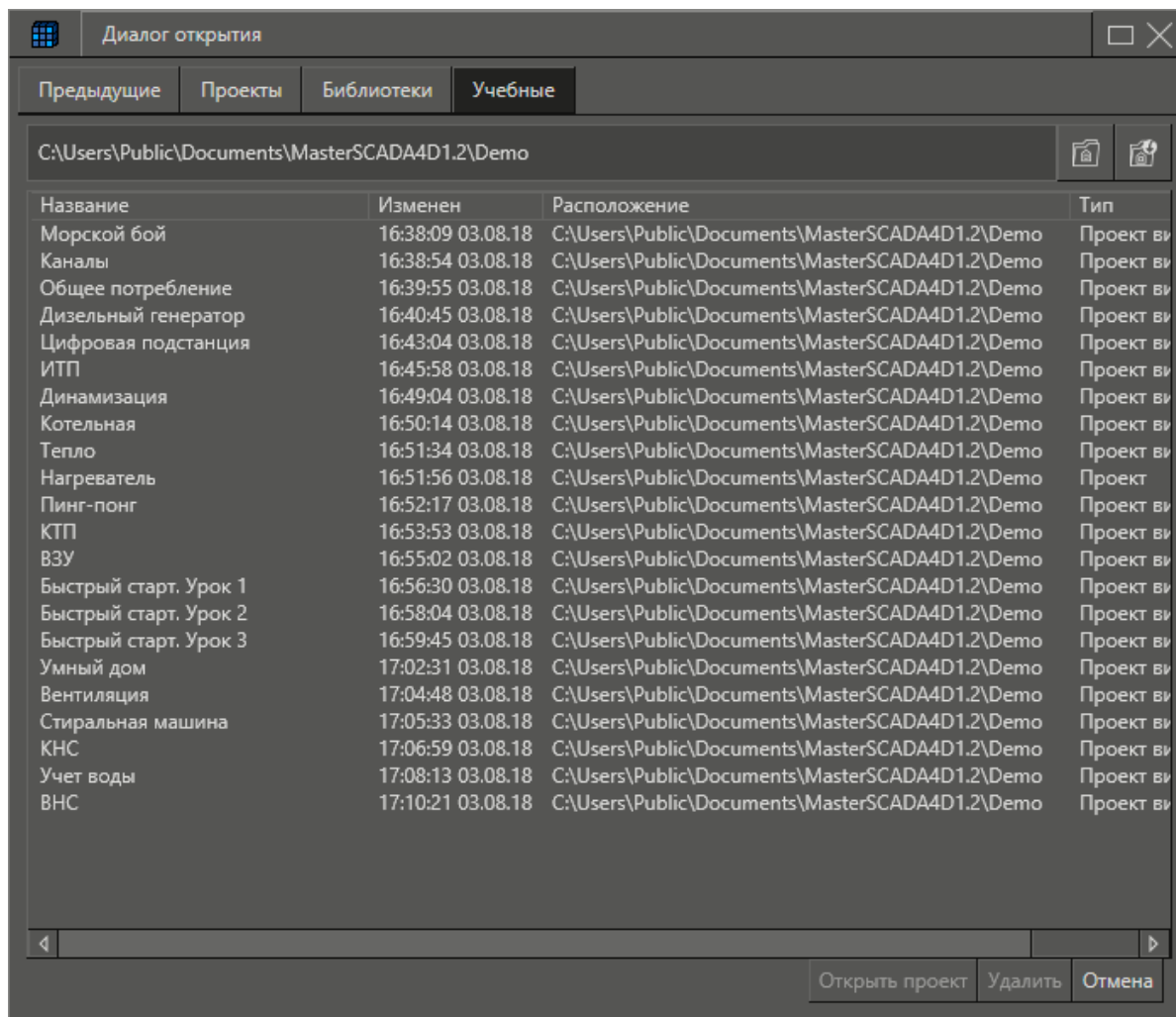
По умолчанию, проекты хранятся в папке C:\Users\Public\Documents\MasterSCADA4D[номер версии]\Projects

### 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.

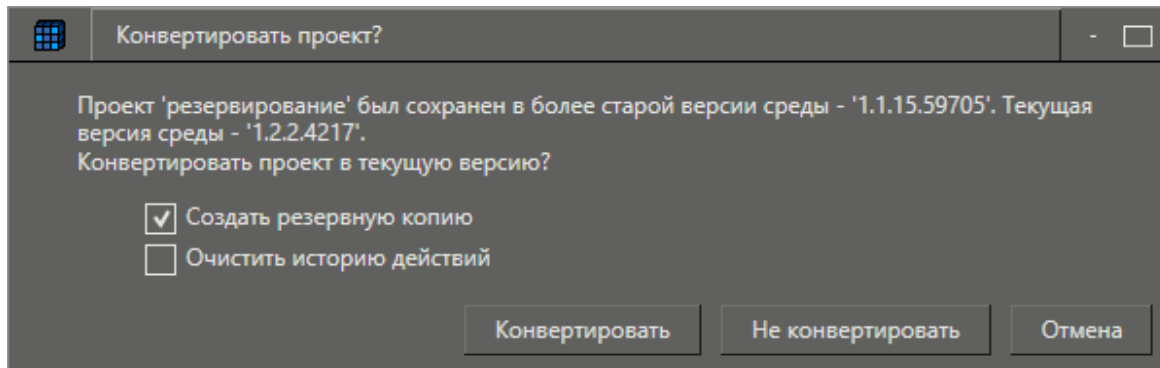
Описание рабочих элементов окна

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.

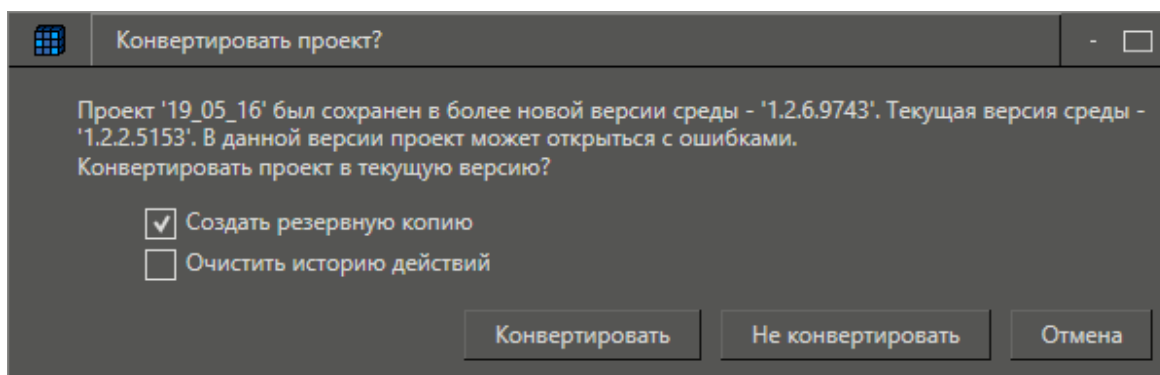
Столбцы таблицы	
Название	Указано имя проекта.
Изменен	Показано время последнего сохранения проекта.
Расположение	Указано место хранения проекта на диске.
Тип	Задан тип проекта. В текущей версии среды разработки все проекты имеют тип Проект визуализации, независимо от настроек, сделанных внутри проекта.
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"> <li>• если выбранный проект открыт в некотором окне редактора проекта, то это окно активируется;</li> <li>• если выбранный проект не открыт ни в одном из окон редактора проекта, то проект открывается в новом окне.</li> </ul> <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.
Закрыть 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. При этом, если в текущем окне был открыт другой проект, то он закрывается. Если проект редактировался, и после этого не был сохранен, то на экране появится диалог, с помощью которого можно сохранить или отменить выполненные изменения).</p>

### 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:



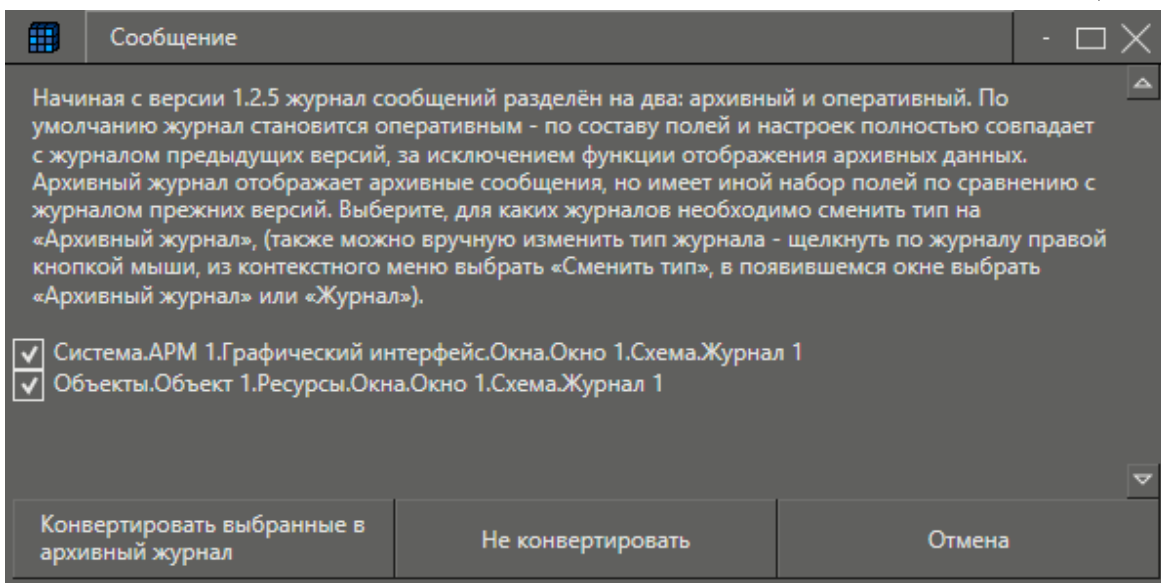
Описание диалога:

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.
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

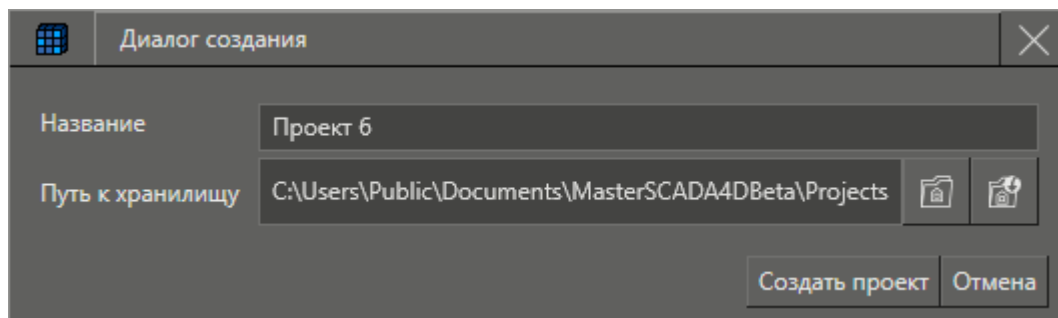
В версии MasterSCADA 4D 1.2.5 появилось два журнала для работы с сообщениями в режиме исполнения: архивный и текущий (оперативный). В более ранних версиях для работы с сообщениями был только один элемент, который работал со всеми сообщениями. Если проект был сделан в более ранних версиях, то при его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



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

#### 5.11.1.1.1. Диалог создания проекта

Вид диалогового окна



#### Поле Название

Поле редактирования имени проекта. При создании для проекта выполняется автоматическая генерация имени <Проект> N.

#### Поле Путь к хранилищу

Задаёт место хранения проекта.

По умолчанию, проект хранится в папке – C:\Users\Public\Documents\MasterSCADA4D<версия>\Projects.

#### Кнопка Отмена

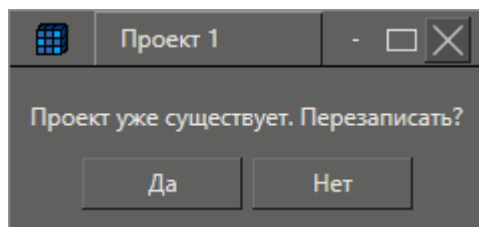
Для отмены создания проекта необходимо нажать на кнопку Отмена. Диалог при этом закрывается.

#### Кнопка Создать проект

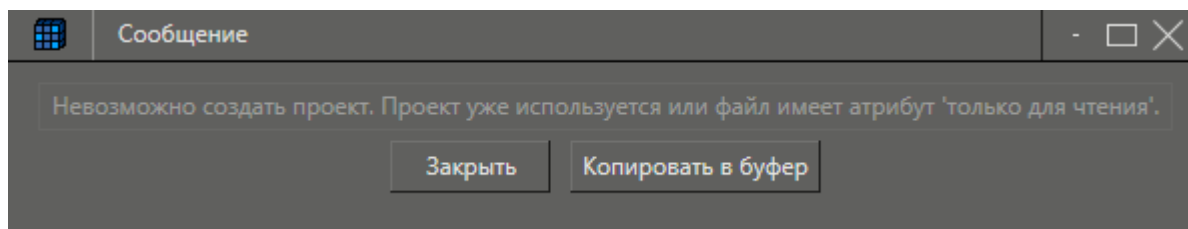
Для создания проекта с заданными параметрами необходимо нажать на кнопку Создать проект. В случае успешного создания проекта в хранилище создается папка проекта с заданным именем.

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:



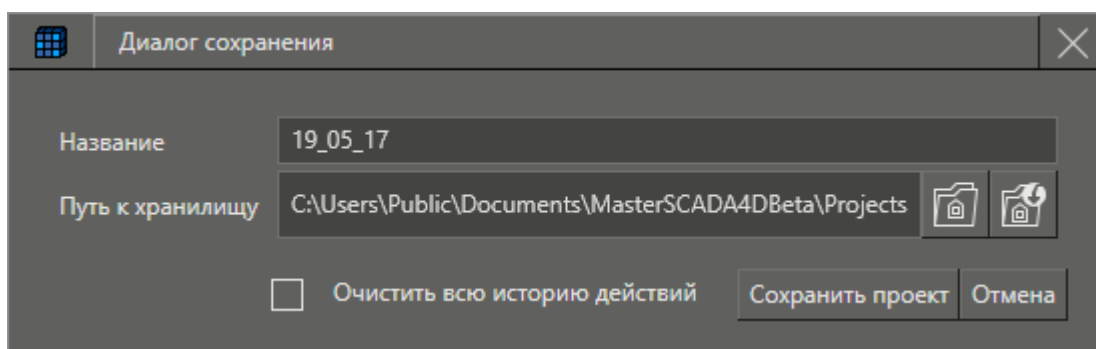
Созданный/перезаписанный проект открывается в новом окне редактора проекта, поэтому создание/перезапись проекта не приводит к закрытию проектов, открытых ранее.



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

### 5.11.1.2. Диалог сохранения проекта

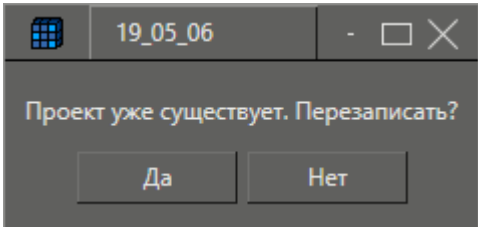
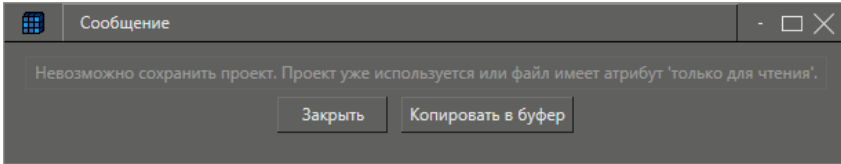
Если выполняется действие Сохранить как, то появляется диалоговое окно.

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"> <li>• проекта с заданным именем не существует в указанной папке – проект сохраняется;</li> <li>• проект с заданным именем существует в указанной папке и открыт в текущем окне редактора – проект сохраняется;</li> <li>• 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):</li> </ul>  <ul style="list-style-type: none"> <li>• проект с заданным именем существует в указанной папке и открыт в другом окне редактора – появляется сообщение о невозможности сохранения проекта:</li> </ul> 
Отмена	Закрывает диалог, отменяя сохранение проекта.

### 5.11.1.3. 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.

#### **5.11.1.4. 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.

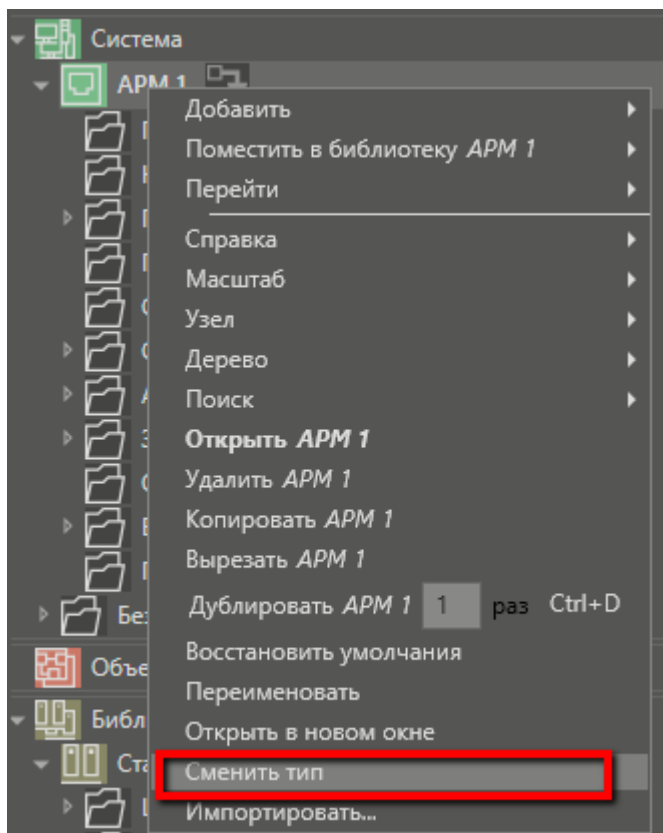
В этом случае подход принципиально ничем не будет отличаться от метода, описанного в Примере 1. Разница лишь в том, что общей частью для обоих участников будет не только дерево системы, но и параметры дерева объектов. Библиотеки разработчиков будут содержать только окна и только программы.

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.

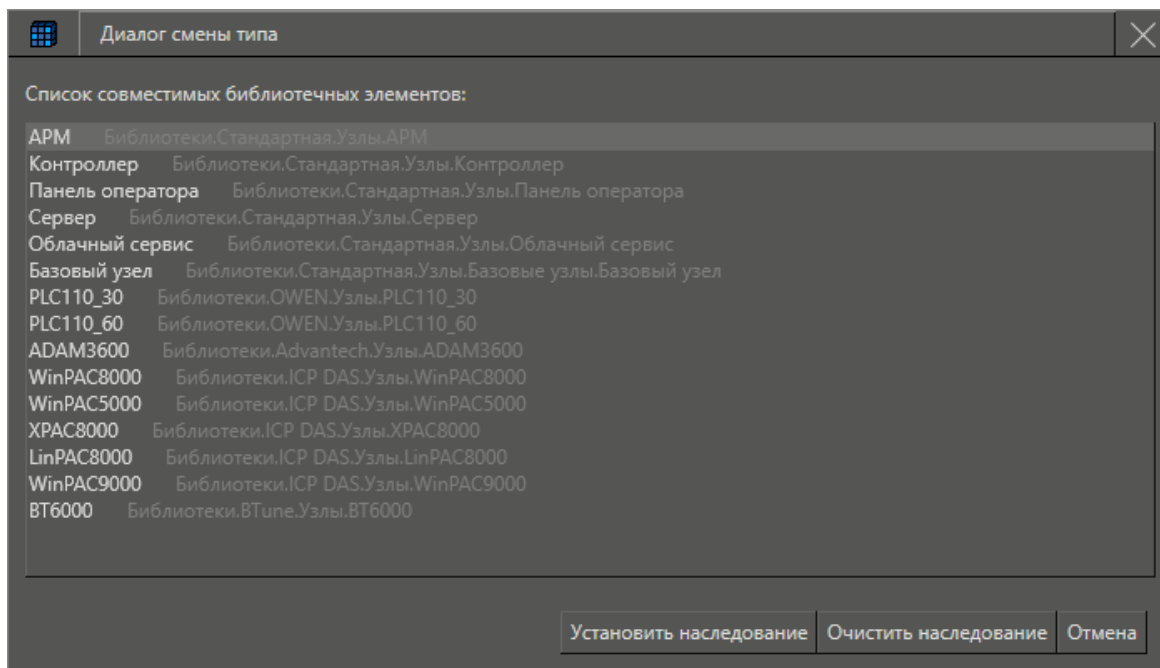
## 5.11.2. Операции с элементами

### 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:



### 5.11.2.2. 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 подобными именами со старшим индексом  $M$  и не существует свободного индекса  $k < M$ ;
- $N=k_{\min}$ , если группа содержит элементы с подобными 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.

Переименование элементов

Для переименования элемента предназначены также следующие инструменты/

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

- панель Свойства .

### 5.11.2.3. 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.

### 5.11.2.4. Автоматическая прокрутка

При переносе элемента за пределы видимой части дерева или редактора (например, редактора FBD или редактора НМІ) выполняется автоматическая прокрутка содержимого дерева/редактора.

### 5.11.2.5. 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.

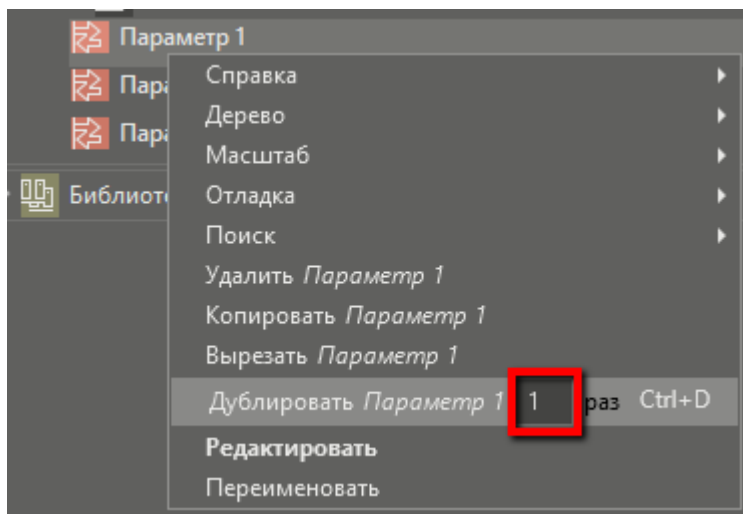
По этой команде в объекте создается соответственно параметр или папка параметров, связанных с исходными параметрами. У создаваемого параметра свойство Доступ имеет противоположное значение относительно исходного параметра.

### **5.11.2.6. 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:

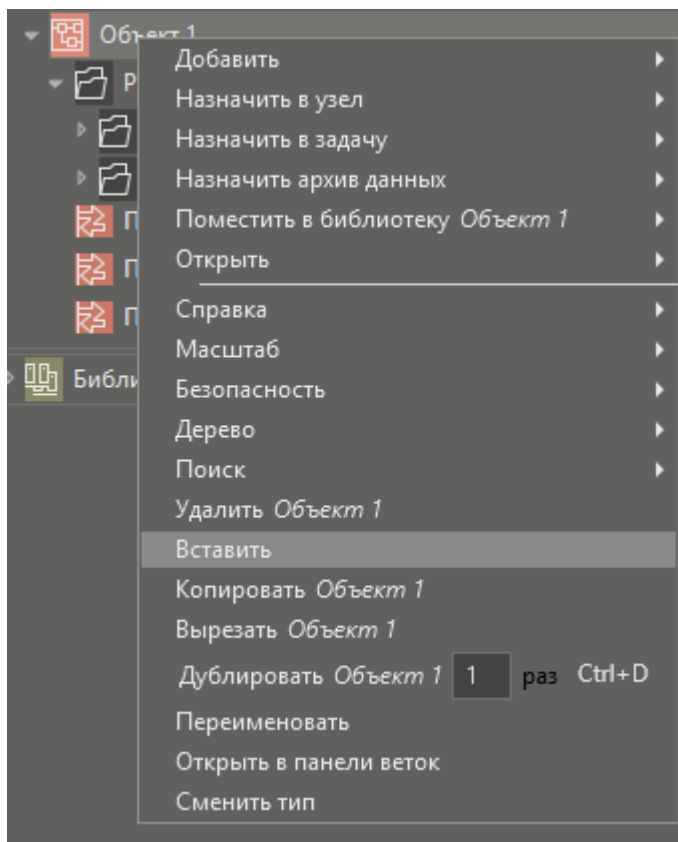
- комбинация клавиш клавиатуры CTRL+C;
- пункт контекстного меню Копировать.

### 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.

### 5.11.2.7. Перемещение элементов в дереве

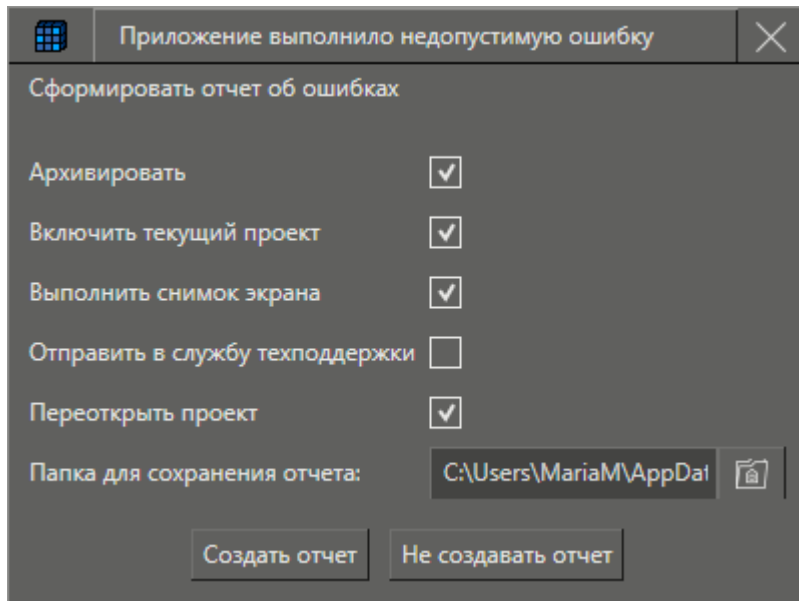
Для перемещения элемента (выделенной группы элементов) в дереве могут быть использованы следующие инструменты (см. также Перетаскивание с удержанием ПК):

- Команда контекстного меню Дерево.Вверх по дереву (SHIFT+ ) и команда Дерево.Вниз по дереву (SHIFT+ ) 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.

## 5.12. Служебные диалоговые окна

Приложение выполнило недопустимую ошибку

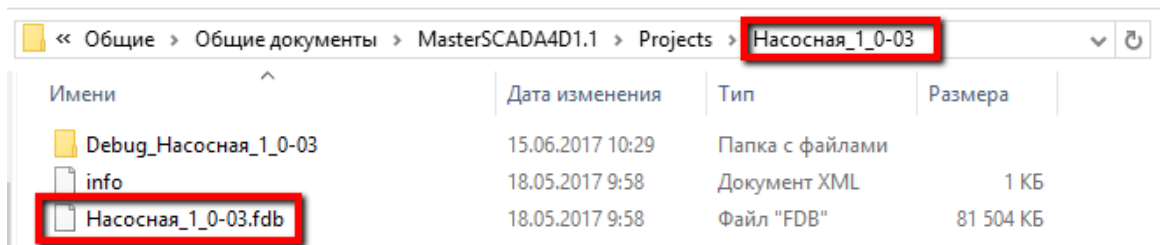
В случае возникновения непредвиденных ошибок, в среде разработки может появиться такое окно:



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 MasterSCADA 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. Типы и экземпляры MasterSCADA 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.

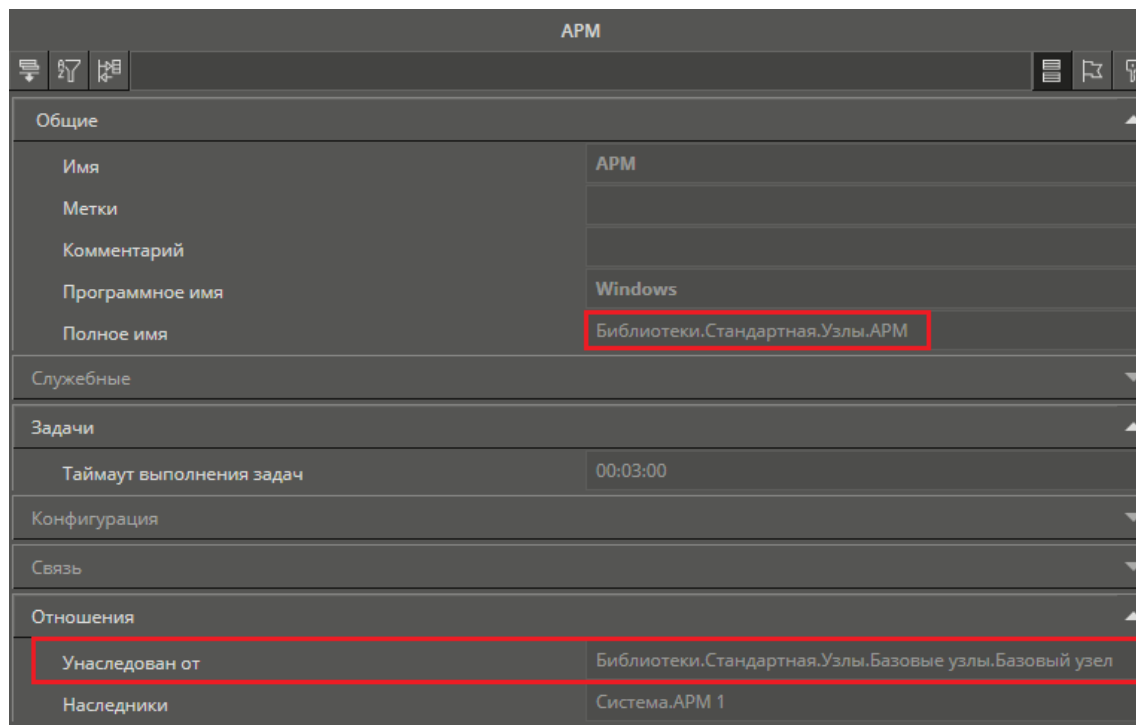
Отношение	Элементы	Описание
Имеет тип	Параметры	При назначении параметру библиотечного типа данных, в дерево автоматически добавляется необходимое количество полей структуры, в зависимости от библиотечного элемента. При изменении библиотечного типа данных, автоматически изменится вид и настройки параметра 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 самостоятельными элементами, а ссылками на элементы типа.
--	--	--

Например, если добавить в дерево системы узел АРМ, то у него будет указано:

АРМ 1	
Общие ▲	
Имя	АРМ 1
Метки	
Комментарий	
Программное имя	АРМ 1
Полное имя	Система.АРМ 1
Служебные ▼	
Задачи ▲	
Таймаут выполнения зад	00:03:00
Конфигурация ▼	
Связь ▼	
Отношения ▲	
Унаследован от	Библиотеки.Стандартная.Узл

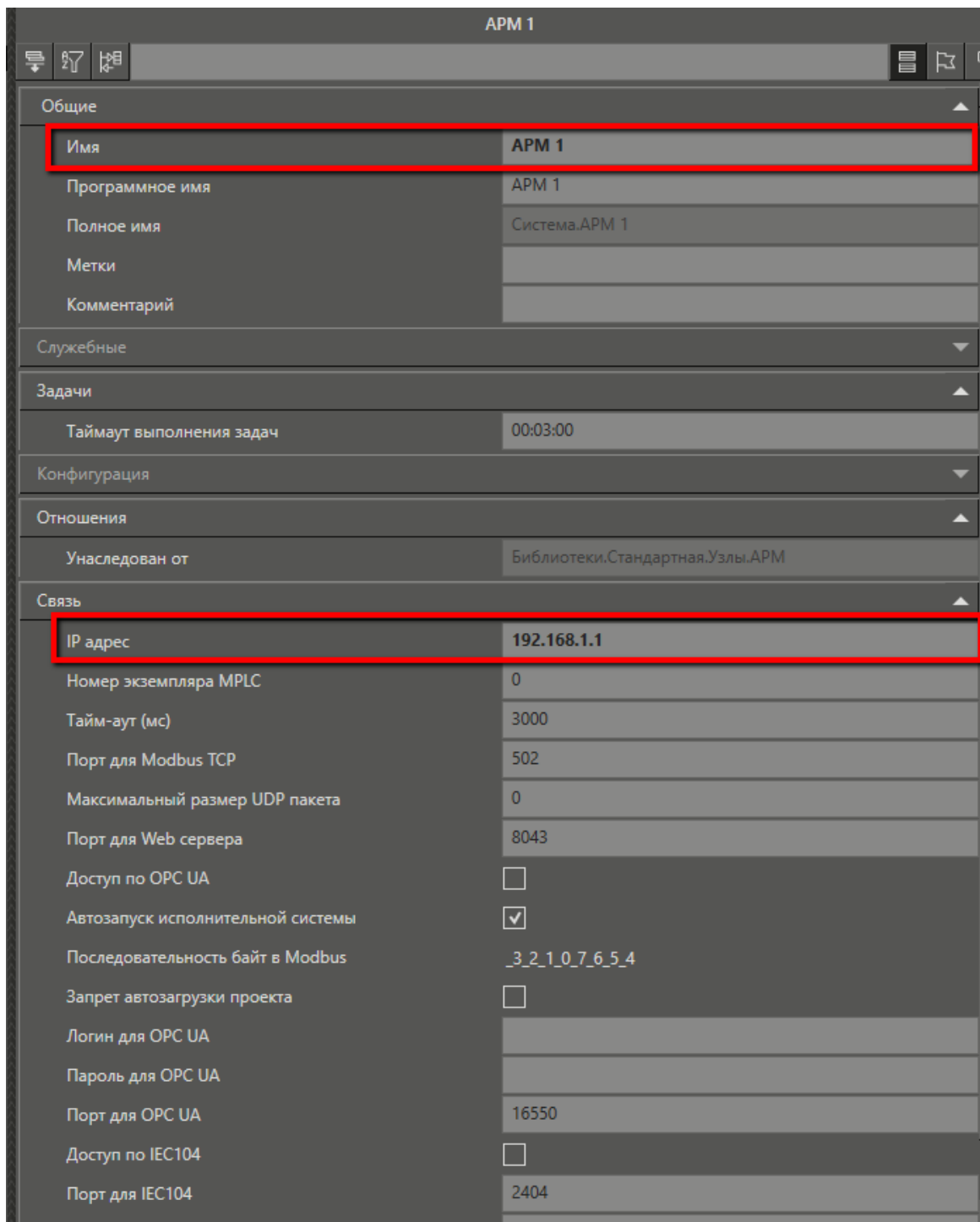
Для расширения базового функционала типы могут наследоваться друг от друга. Рассмотрим узел АРМ в дереве библиотек. Несмотря на то, что он сам является типом (и мы можем добавить его в проект), в его настройках видно, что он унаследован от другого типа. От Базового узла он отличается именем и рядом предопределенных 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.

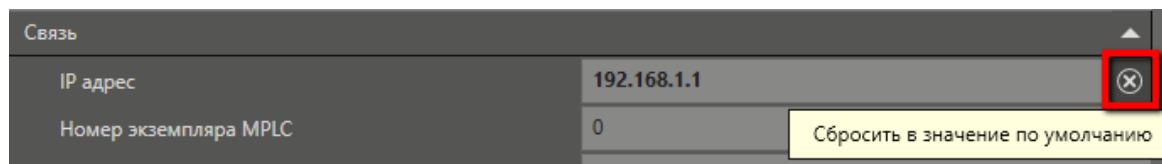
При анализе элементов необходимо обращать внимание на панель свойств. В ней свойства, которые не менялись разработчиком (т.е. которые совпадают со свойствами типа), отображаются обычным шрифтом, а измененные – жирным.



Это означает, что если настройку IP-адреса изменить в типе, то она не поменяется в проекте автоматически.

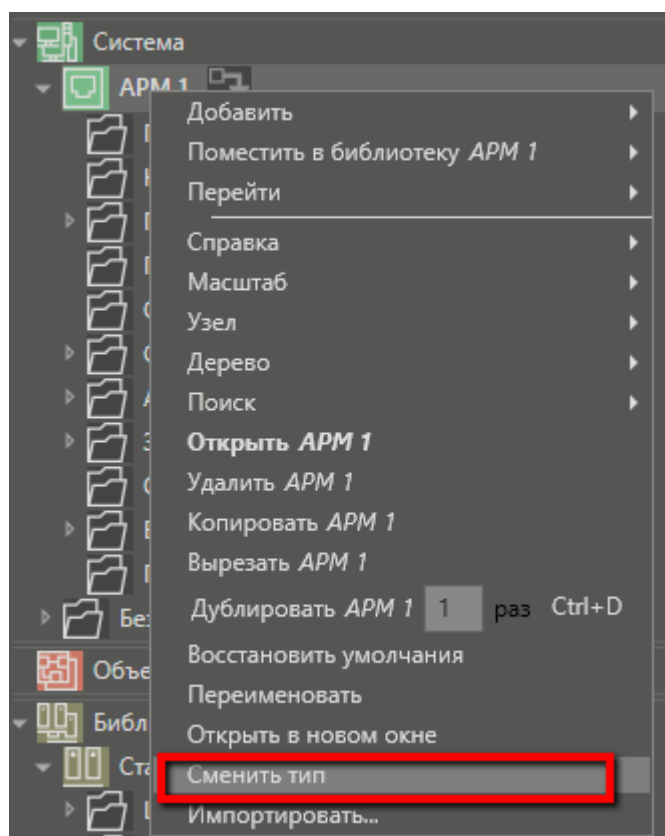
Настройки и составные части экземпляра можно синхронизировать с типом несколькими способами. Во-первых, если выполнить пункт контекстного меню элемента Восстановить умолчания. В этом случае все дочерние группы и свойства элемента

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:



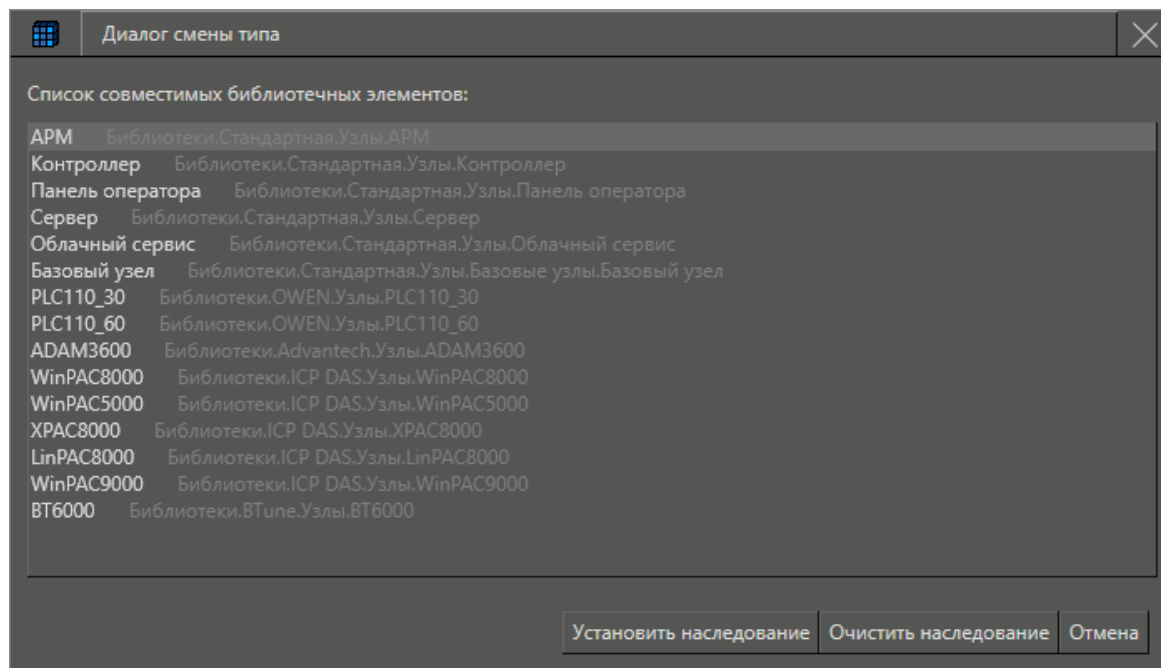
Это приведет только к синхронизации текущего свойства.

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:





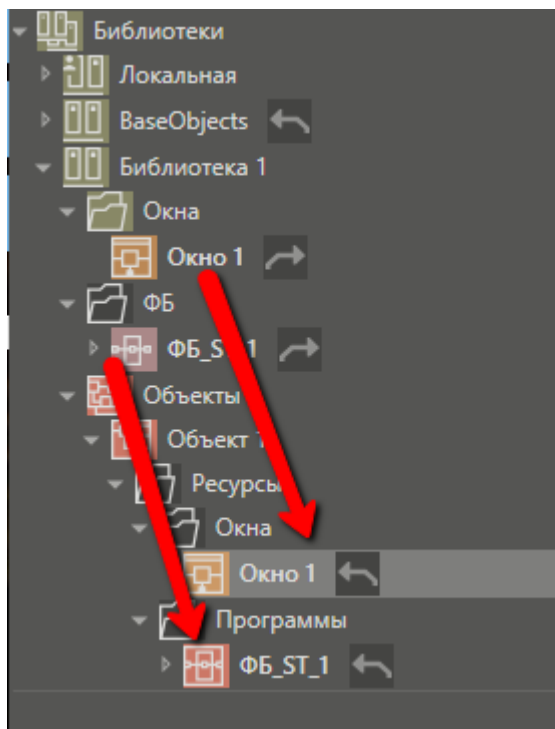
## 6.2. Методы разработки типов элементов

Чаще всего разработчики проектов создают свои типы объектов, каналов или тегов. Существует два подхода к разработке новых типов элементов:

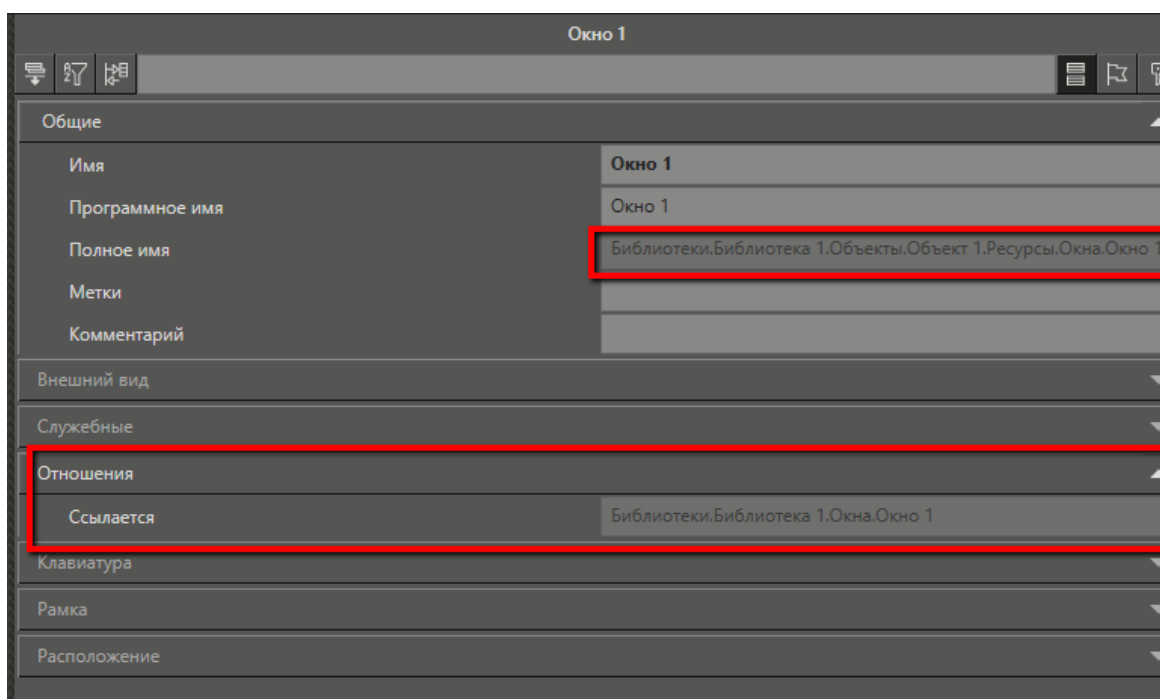
- от типа к экземпляру;
- от экземпляра к типу.

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).

### 6.3. 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.

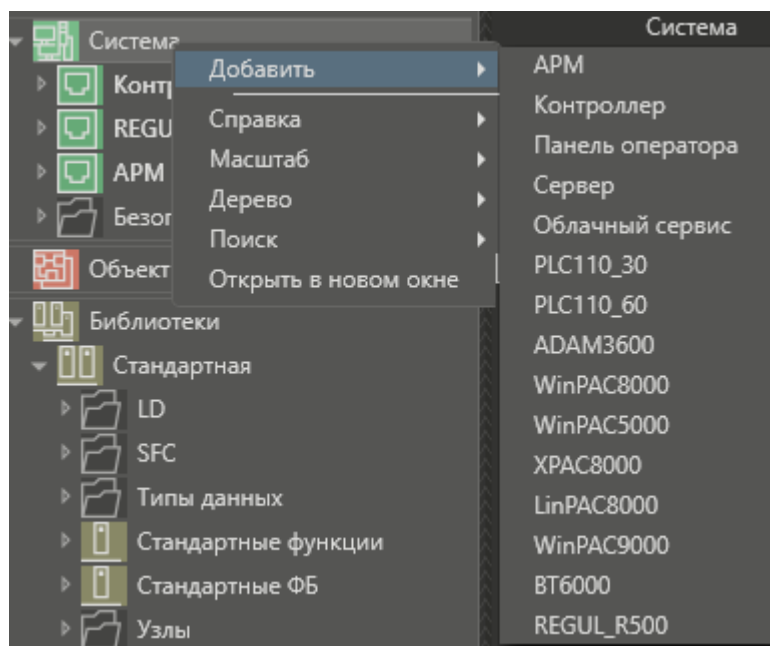
В дереве системы также существует возможность создавать и программы, и окна, но в этом случае разрабатываемый проект строго привязан к используемому оборудованию, и при смене оборудования будет затруднена адаптация проекта. Такой способ разработки называется линейным. Мы рекомендуем использовать объектный подход, когда в дереве системы описывается только физическая часть проекта, а вся обработка (программы, окна, настроенные параметры) находятся в дереве объектов.

#### 6.3.1. 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:

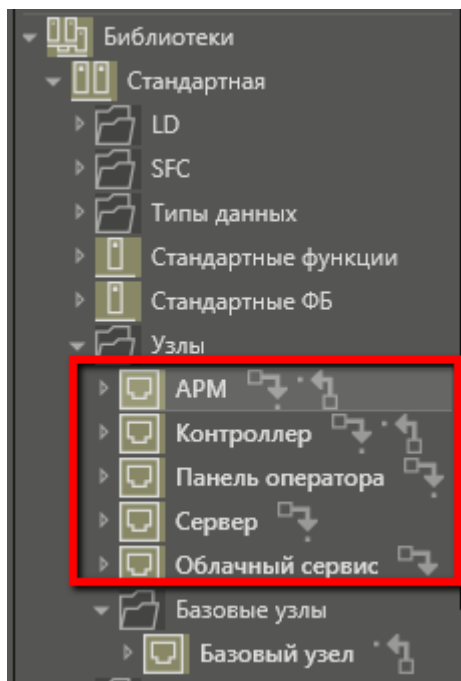


Разделяют два типа узлов:

- 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 крейте контроллера могут размещаться устройства ввода-вывода, то его конфигурация по умолчанию содержит корневую группу Встроенные модули

### 6.3.1.1. Панель свойств Узла

Панель свойств узла имеет вид:

APM 1	
Общие	
Задачи	
Конфигурация	
Отношения	
Связь	
Автозапуск клиента визуализации	<input checked="" type="checkbox"/>
IP адрес	127.0.0.1
Доступ по OPC UA	<input type="checkbox"/>
Порт для OPC UA	16550
Максимальный размер UDP пакета	0
Автозапуск исполнительной системы	<input checked="" type="checkbox"/>
Последовательность байт в Modbus	_3_2_1_0_7_6_5_4
Номер экземпляра MPLC	0
Тайм-аут (мс)	3000
Порт для Modbus TCP	502
Запрет автозагрузки проекта	<input type="checkbox"/>
Логин для OPC UA	
Пароль для OPC UA	
Порт для Web сервера	8043
Запрет входящих подключений	<input type="checkbox"/>
Максимальное число подключений Modbus	20
Время неактивности подключения Modbus	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 checked="" type="checkbox"/>
Порт для IEC61850	102
В IEC61850 параметры SYSTEM_*_VALUE пре	<input checked="" type="checkbox"/>
Использовать TLS для IEC61850	<input type="checkbox"/>
Ограничение длины имени узла в IEC61850	10

Описание основных свойств Узла:

Свойство	Назначение
Категория Задачи	
Таймаут выполнения задач	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.
Категория Конфигурация	
Наличие визуализации	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.
Параметры запуска RT	Задаются ключи запуска исполнительной системы.
Категория Связь	
Автозапуск клиента визуализации	Определяет необходимость запуска клиента визуализации при загрузке конфигурации. Если флаг установлен, то клиент визуализации запустится автоматически.
IP-адрес	Задается IP-адрес устройства, на котором установлена среда исполнения, и на которое должен загрузиться разрабатываемый проект.
Номер экземпляра MLC	Если на одном устройстве запущено несколько исполнительных систем, то задается номер экземпляра исполнительной системы, в которую должен загрузиться разрабатываемый проект. <b>Важно!</b> 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.



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	Если клиент визуализации запускается из среды разработки, то данный порт будет использоваться в адресной строке. 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
Доступ по OPC UA	Активирует для среды исполнения возможность работать как OPC UA сервер.
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, или произойдет отключение от среды исполнения, то среда исполнения также завершит свой процесс. Смотрите также: Автоматическая загрузка проекта.
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 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	<p>Определяет возможность работать MasterSCADA 4D в роли IEC61850 Slave.</p> <p>Ограничения:</p> <ul style="list-style-type: none"> <li>• При превышении размера ответа сервера 65099 байт, сервер отвечает сообщением Confirmed-ErrorPDU.</li> <li>• Ограничение на длину полного имени переменной 64 байт. Полная длина имени переменной /LD/LN/DO..... не может превышать 64 байта</li> </ul>
Порт для IEC61850	Задается порт, который будет использоваться для обмена данными с клиентами, если MasterSCADA 4D работает в роли IEC61850 Slave.
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/clt.</p> <p>Имя файла ключа клиента должно быть MasterSCADA4Dclt_key.pem. Имя файла сертификата клиента должно быть MasterSCADA4Dclt.cer.</p> <p>Имя файла ключа сервера должно быть MasterSCADA4Dsrv_key.pem. Имя файла сертификата сервера должно быть MasterSCADA4Dsrv.cer.</p> <p>Сертификаты CA должны храниться в папке IEC61850/clt/root для клиента и IEC61850/srv/root для сервера. Сертификаты CA могут иметь любое имя и расширение .cer.</p>

<p>В IEC61850 параметры SYSTEM_*_VALUE представлять как t, q, stVal</p>	<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>
<p>Ограничение длины имени узла в IEC61850</p>	<p>Определяется максимальная длина идентификатора тега протокола IEC61850 Slave. Идентификатор формируется на основании полного имени параметра. Рассмотрим пример. Пусть полное имя параметра будет: Система.ARM_DISPETCHERA.Параметры.Temperatura_max. Т.к. по стандарту IEC61850 имена должны быть заданы только латинскими буквами, то полное имя будет автоматически преобразовано: SystemRoot.ARM_DISPETCHERA.Parameters.Temperatura_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. <b>Важно!</b> 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>

### 6.3.1.1.1. 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 mpls.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	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 при передачи данных в режиме Slave через механизм Внешние каналы.
/bZ	Z - это скорость, на которой будет работать com-порт.
/HDA Archive-Save-Period:30	Используется в случае, когда MasterSCADA 4D передает архивные данные в M-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.
/HDA ArchiveSizeLimit:60000	Используется в случае, когда MasterSCADA 4D передает архивные данные в M-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.
/HDA Archive-Store:1	Используется в случае, когда MasterSCADA 4D передает архивные данные в M-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 внутри контроллера>	Указывает по какому пути будет находиться файл с информацией о работе исполнительной системы. По умолчанию лог-файл пишется в папку 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.
/t	Включает расширенную диагностику работы среды исполнения. При использовании данного ключа, в файл с информацией о работе исполнительной системы будет попадать лог обмена среды исполнения с различными устройствами, работающими по заданным в проекте протоколам.

/nowdt	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).
/iATE0	Инициализация модемов ОБЕН ПМ01 и IRZ TU32. Если используются модемы этих типов, то данный ключ необходим для их корректной работы. Пример, строки Параметры запуска RT /iATE0 /m3. В этом случае будет использоваться скорость, установленная производителем по умолчанию, 9600.
/export	Используется для экспорта в файл данных горячего рестарта для контроллеров с энергонезависимой памятью. <b>Важно!</b> 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.
/delhr	Удаляет файл горячего рестарта. <b>Важно!</b> 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.
/imit	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 устройство не подключено. Аналоговые значения будут изменяться в диапазоне от 0 до 100, а дискретные менять свое значение с TRUE на False.

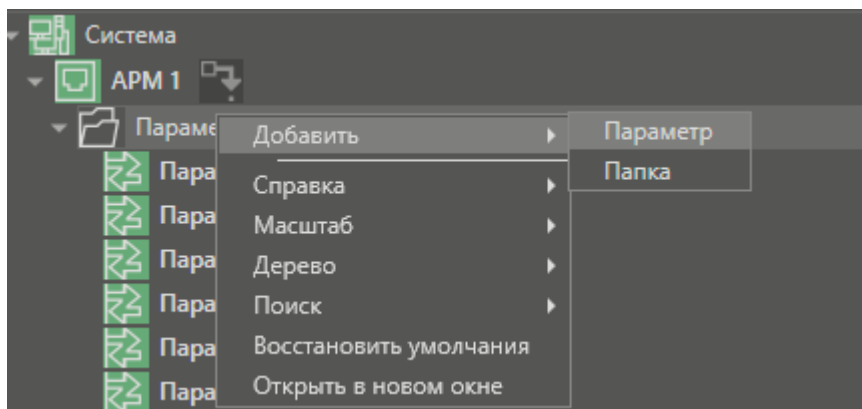
**Важно!** Отключение режима watchdog должно быть осмысленным. Рекомендуется отключать его только в том случае, когда при его наличии работа невозможна.

Данные ключи можно также использовать при настройке среды исполнения, предназначенной для работы на ОС Windows, при помощи приложения MasterSCADA 4D Monitor

### 6.3.1.2. 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.

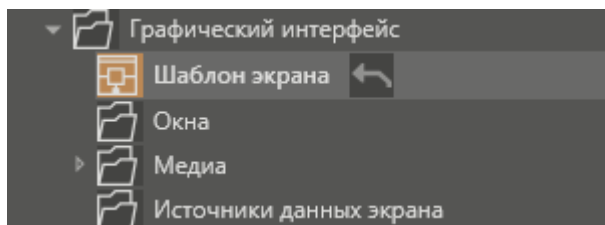
Если параметры созданы в дереве системы, то они являются глобальными. Такие параметры можно использовать в программе ST без дополнительного определения, достаточно написать имя переменной в тексте программы.

**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.

### 6.3.1.3. 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

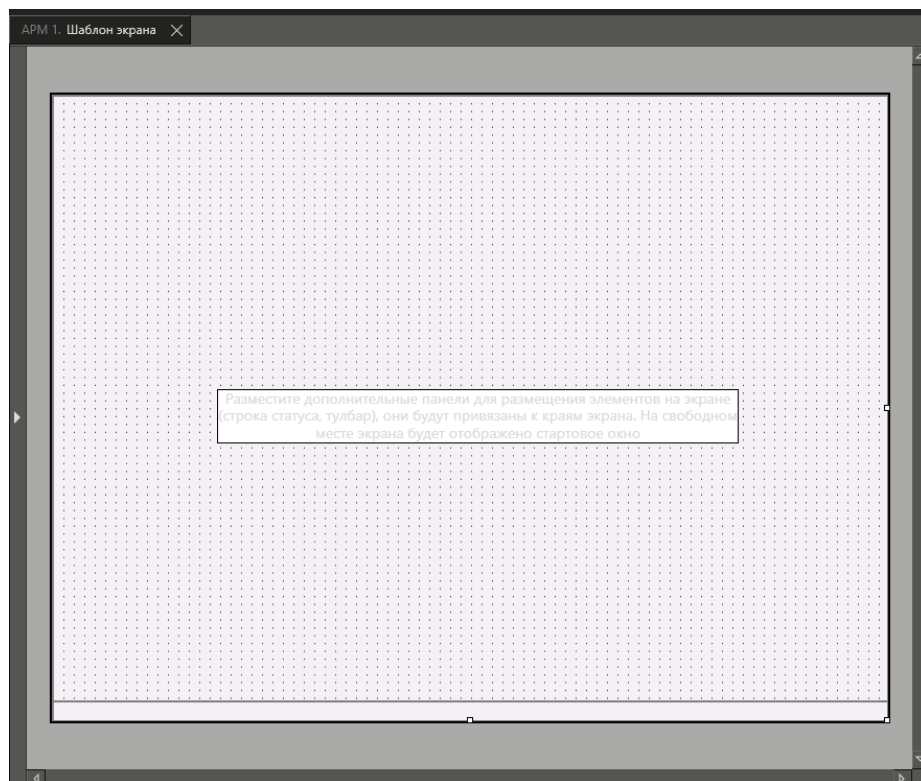
### 6.3.1.3.1. Шаблон экрана

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:



По сути Шаблон экрана является элементом Панель-Док

#### 6.3.1.3.1.1. Панель свойств Шаблона экрана

Вид панели:



Шаблон экрана

Общие

Имя: Шаблон экрана

Метки:

Комментарий:

Внешний вид:

Рамка:

Расположение:

Служебные:

Заголовок окна

Шрифт заголовка окон: Arial

Размер шрифта заголовка: 12

Экран

Период обновления: 100

Период мигания: 1000

DPI экрана: 72

Полноэкранный режим:

Язык исполнения: ru


Запрет масштабирования:

Уровень предупреждений: Все

Высота заголовка всплывающих окон: 20

Панель Свойства для элемента Шаблон экрана содержит, наряду с другими, свойства, общие для всех мнемосхем узла. Наиболее часто изменяемые свойства Шаблона экрана:

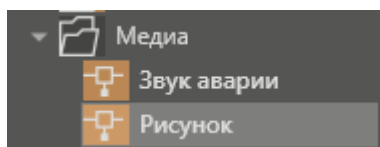
Название	Назначение
Группа Общие	Описание доступно в разделе справочной системы.

Группа Внешний вид	Настройки, заданные на странице свойств, распространяются только на оболочку экрана. На окна, открывающиеся в контейнерах шаблона экрана эти свойства не распространяются. Смотрите также Группа Внешний вид
Группа Заголовок 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. Смотрите также: Подробное описание настроек группы Расположение.
Группа Служебные	Смотрите также: Описание группы настроек Служебные.
Группа Экран	
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.
Высота заголовка всплывающих окон	Данная настройка определяет высоту заголовков всплывающих окон, окон управления, окон ввода уставок и т.п., созданных пользователем.
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</p>

	<p>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 пор, пока оператор вручную не изменит масштаб страницы специальными командами или колесом мыши. Это может привести к скрытию элементов за границами видимости.</p>
Период мигания	<p>Если у элементов свойство Мигание установлено в состояние True, то элемент будет мигать с указанным периодом. Данная настройка едина для всех элементов. Установить разный период мигания для различных элементов нельзя.</p>
Период обновления	<p>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.</p>
Полноэкранный режим	<p>Если флаг установлен, то окно клиента, открытое из среды разработки, откроется на весь экран. Данная настройка не влияет на другие способы открытия клиента. Смотрите также: Работа с клиентом визуализации</p>
Язык исполнения	<p>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</p>

### 6.3.1.3.2. Медиа

Эта группа входит в состав групп Графический интерфейс , Шаблон экрана и Окно . В группе Медиа создаются элементы Медиа (см. Элемент Медиа).



#### 6.3.1.3.2.1. 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 НМІ .

Медиа-ресурс, добавленный в узле, будет доступен для всех окон проекта.

### 6.3.1.3.3. Источники данных экрана

Эта группу содержат группы Графический интерфейс и библиотека НМІ .

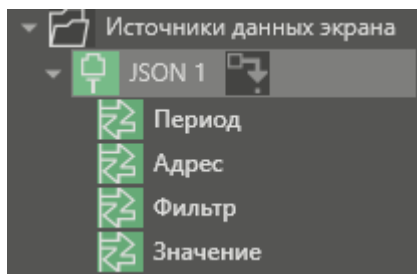
В этой группе создаются элементы JSON (см. Элемент JSON )

#### 6.3.1.3.3.1. Элемент JSON

Элемент JSON используется для получения данных от сторонних программ в клиенте, без использования WEB-сервера MasterSCADA 4D.

Элементы JSON создаются в группе Источники данных экрана . Данный элемент предусмотрен для обеспечения взаимодействия со службой Windows, реализующей соответствующий функционал.

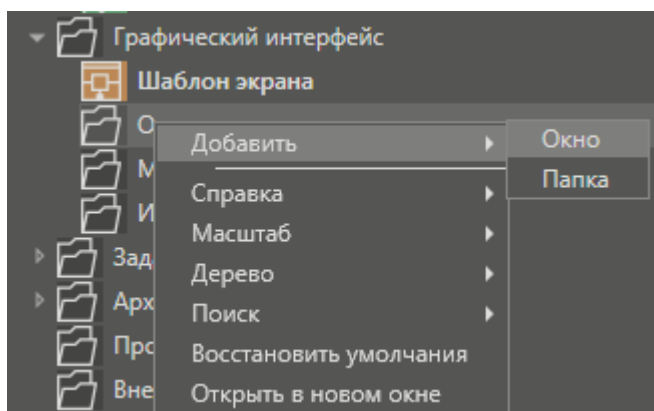
JSON (англ. JavaScript Object Notation) – текстовый формат обмена данными, основанный на JavaScript и обычно используемый именно с этим языком.



Для получения дополнительной информации обращайтесь в техническую поддержку

#### 6.3.1.3.4. Окна

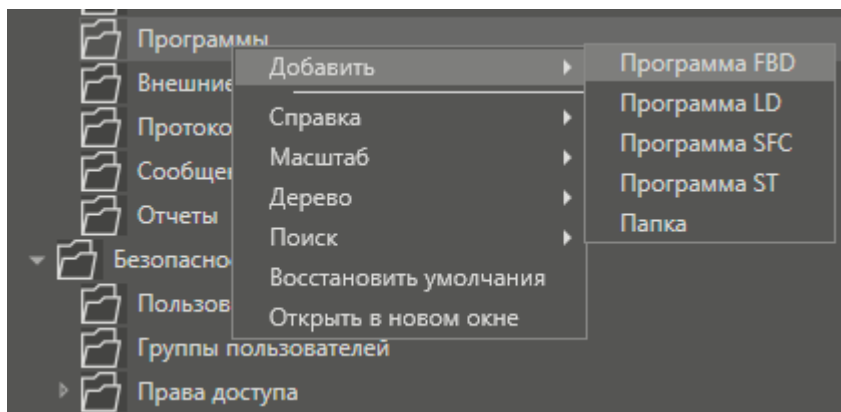
В данной группе создаются Окна.



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.

#### 6.3.1.4. Программы



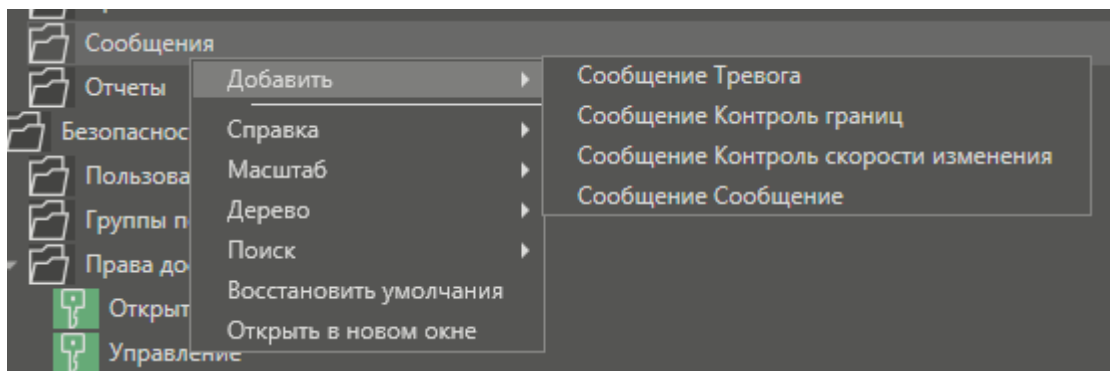
В этой группе создаются программы. Для написания программы можно использовать любой из поддерживаемых языков. Как правило, программы создаются непосредственно в дереве системы в тех случаях, когда нельзя по смыслу отнести ту или иную программу к конкретному объекту. Например, когда в программе производится 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.

### 6.3.1.5. 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.

### 6.3.1.6. Службы

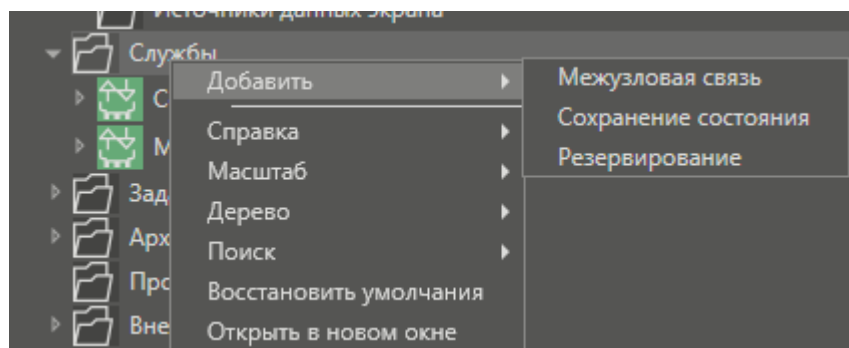
Данная группа узла содержит службы, обеспечивающие связи между различными исполнительными системами, включенными в один проект, а также сохранение данных, которые в последствии используются для горячего рестарта.

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 разработки. 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.

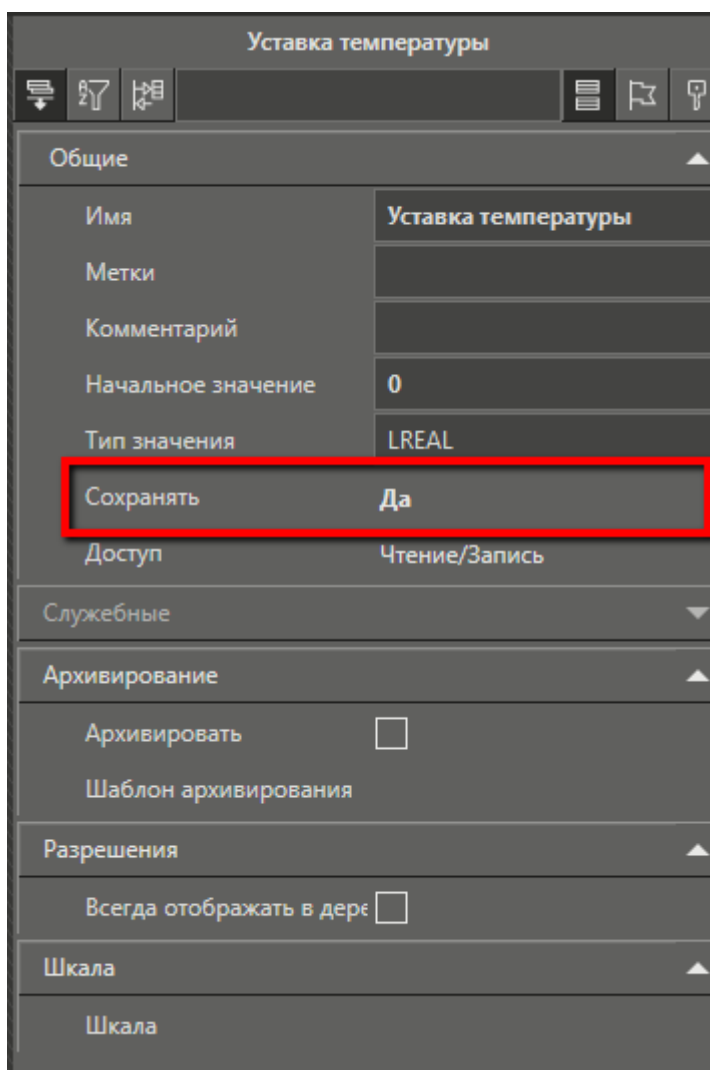
Для организации резервирования узлов необходимо добавить соответствующую службу:



- Резервирование - используется при резервировании узлов.

### 6.3.1.6.1. 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.



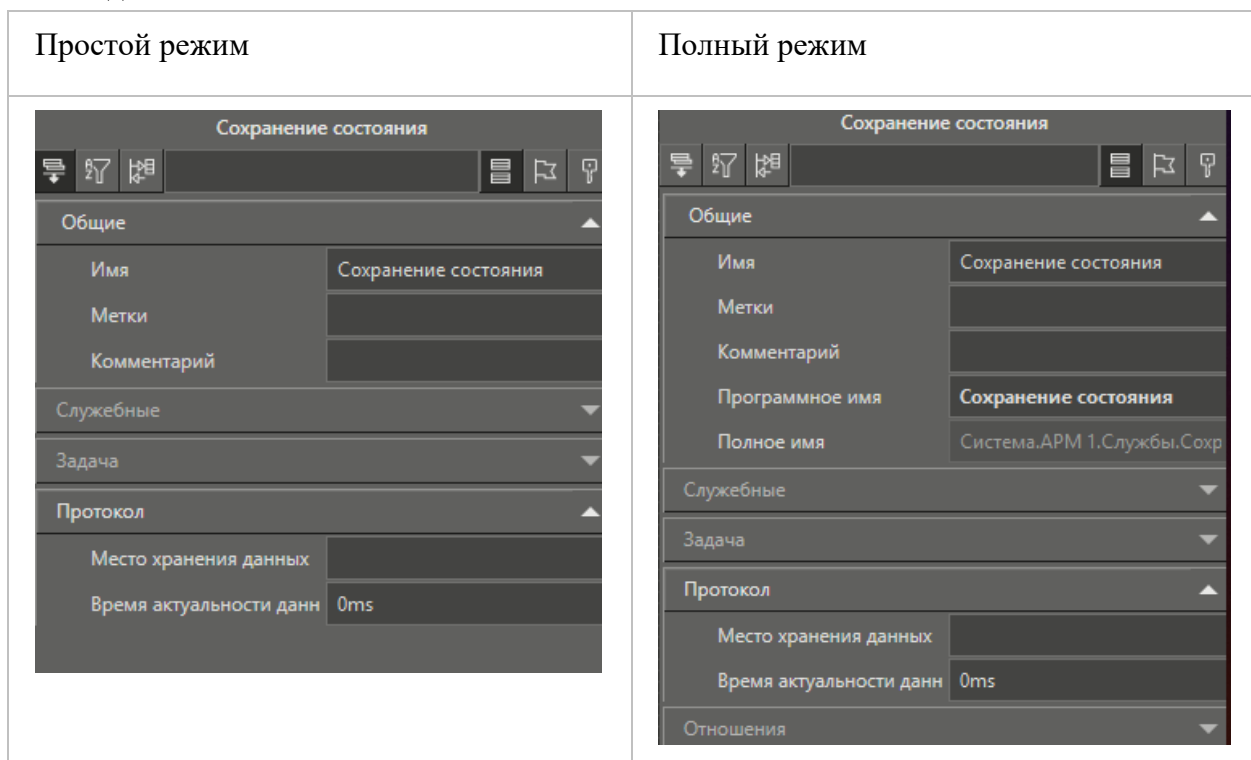
Уставка температуры	
Общие	
Имя	Уставка температуры
Метки	
Комментарий	
Начальное значение	0
Тип значения	LREAL
<b>Сохранять</b>	<b>Да</b>
Доступ	Чтение/Запись
Службные	
Архивирование	
Архивировать	<input type="checkbox"/>
Шаблон архивирования	
Разрешения	
Всегда отображать в дереве	<input type="checkbox"/>
Шкала	
Шкала	

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 session.bin. Если поле пустое, то файл сохранится в рабочую папку исполнительной системы:</p> <p>для ОС Linux - mpls/session.bin.</p> <p>для ОС Windows - в папке Server, расположение которой задается в настройках MasterSCADA 4D Monitor (по умолчанию - c:\ProgramData\InSAT\MasterSCADA4DRTBeta\Server\session.bin)</p> <p>В поддерживаемых контроллерах с энергонезависимой памятью (например, PLC110, M903) данные горячего рестарта хранятся в такой памяти.</p>

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.
---------------------------	------	--

### 6.3.1.7. 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.

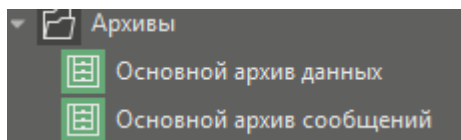
Сроки подключения и особенности работы можно узнать в технической поддержке.

База данных SQLITE 3 входит в состав MasterSCADA 4D. Другие базы данных необходимо установить самостоятельно.

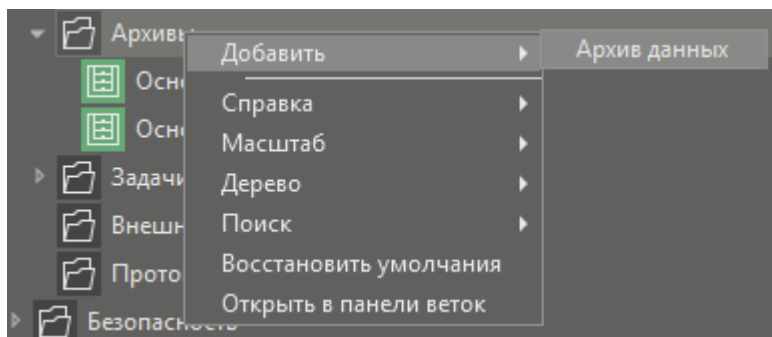
Различают два типа архивов: архив данных и архив сообщений.

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.



### 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.

## 6.3.1.7.1. 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 ):

Название	Тип	Назначение
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;
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. Если данные не менялись дольше, чем указано в поле Максимальный период записи, то также произойдет запись значения в архив.</p> <p>Если флаг снят, то будет использоваться периодическая запись данных в архив с периодом, указанным в поле Период задачи записи архива.</p>
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 чистка архива, задается в настройке Период чистки архива
Максимальный объем архива (Mb)	LREAL	<p>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.</p>

Максимальный период записи (Tmax)	TIME	Значения пишутся в архив с данным периодом, даже если они не изменялись. При нулевом значении, указанном в данном поле, настройка не влияет на работу БД
Интервал между попытками подключения	TIME	Интервал, с которым осуществляется повторное подключение в случае неудачного подключения к БД
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"> <li>• Чтение (если указано &gt; 3 потоков, то чтение может выполняться одновременно из нескольких потоков)</li> <li>• Запись</li> <li>• Удаление старых записей</li> </ul>
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 происходит прореживание данных.
Мертвая зона (Dead-band)	LREAL	Значение не записывается в архив, если оно изменяется на величину меньшую, чем задано данным

		параметром. Данная настройка влияет только на вещественные параметры. Используется только если установлен флаг Запись по изменению
Минимальный период записи (Tmin)	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.
Параметры подключения	STRING	
Пароль	STRING	Пароль доступа к базе данных.
Период задачи записи архива	TIME	Период записи в архив при периодической записи.
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;
Пользователь	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.
Сервер	STRING	При использовании удаленной базы данных, указывается адрес компьютера, на котором находится база данных. (IP-адрес, либо MAC-адрес)
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.
Число удаляемых элементов в одной операции	DINT	Количество элементов (данных), которые удаляются из архива за один раз;

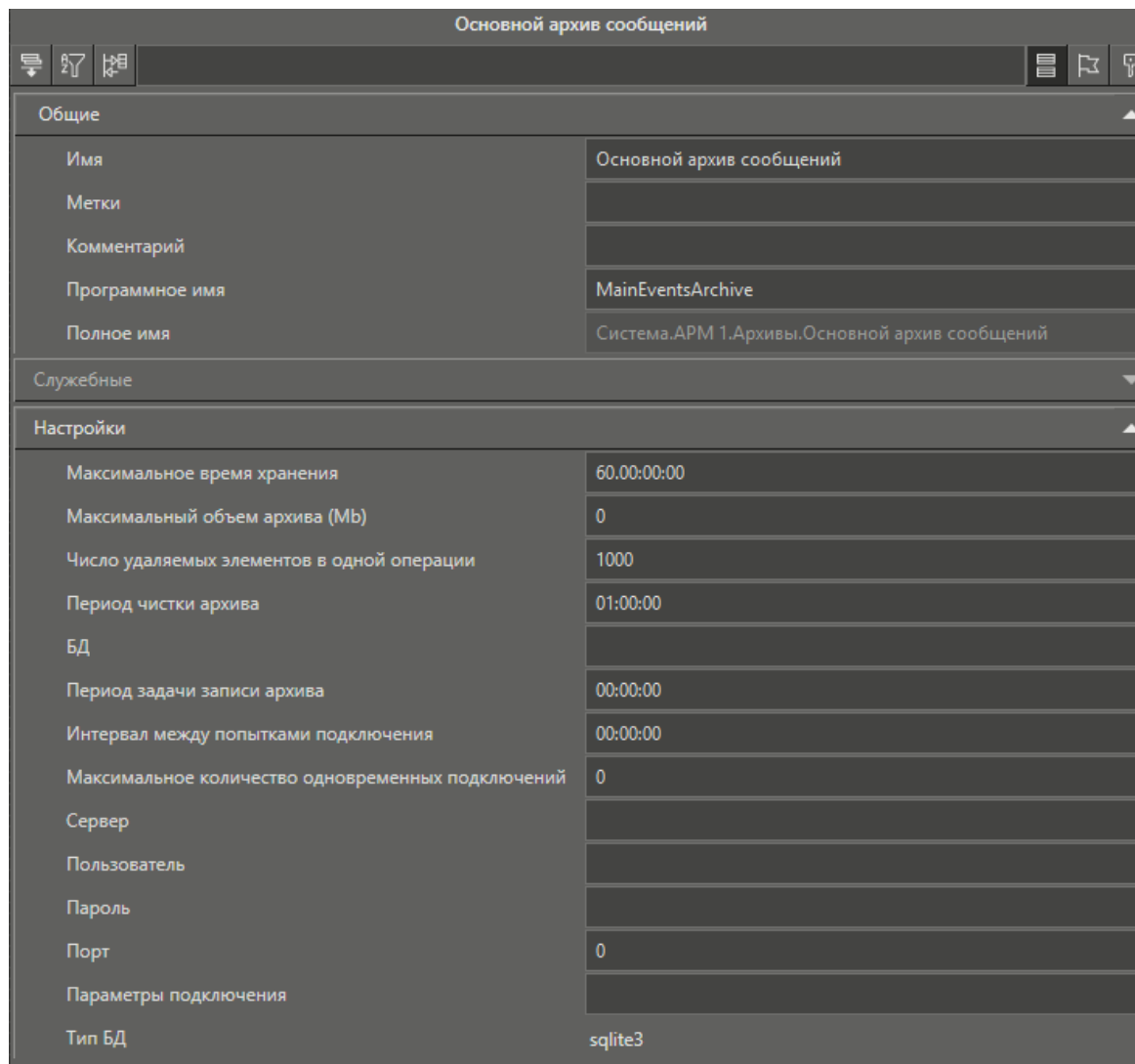
Пусть DV – разница текущего значения и последнего записанного, DT – разница текущего времени и времени последней записи. Тогда условие записи в архив выглядит примерно так:

$$(Ch \ \&\& \ DV \geq \text{Deadband} \ \&\& \ (Tmin==0 \ || \ DT > Tmin)) \ || \ (Tmax \neq 0 \ \&\& \ Tmax < DT)$$

### 6.3.1.7.2. 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



## 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 запускается очистка архива, задается в настройке Период очистки 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.
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;
Период задачи записи архива	TIME	Период записи в архив при периодической записи. При периодической записи в архив записываются только изменившиеся сообщения, с заданным периодом.

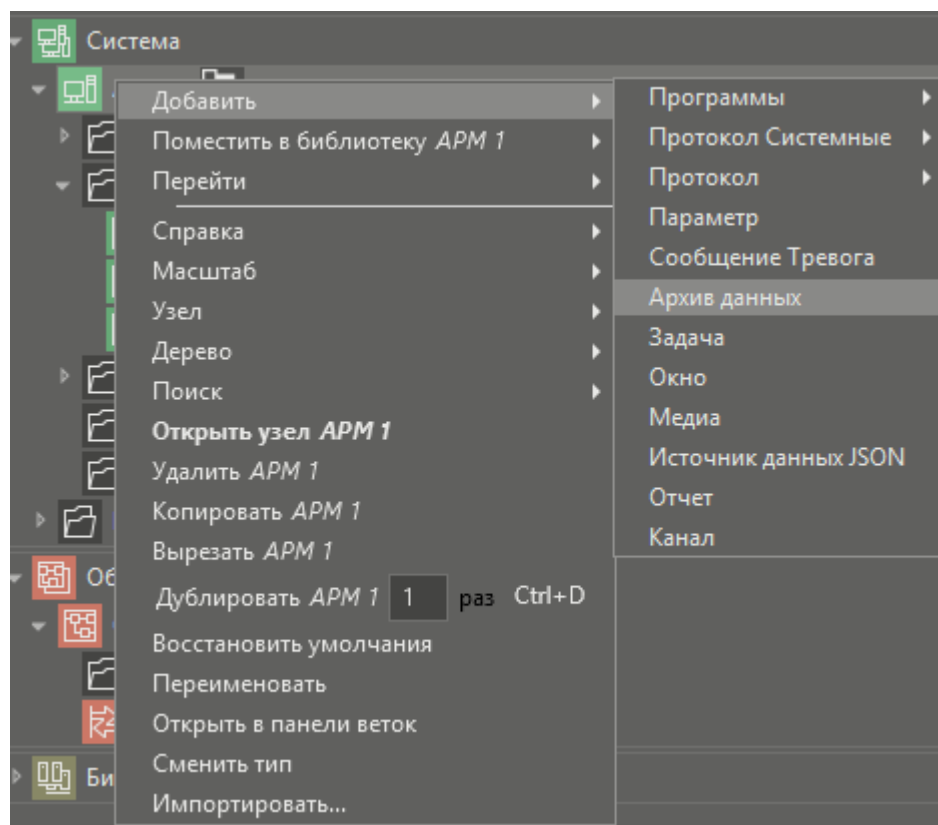
Интервал между попытками подключения	TIME	Интервал, с которым осуществляется повторное подключение в случае неудачного подключения к БД
Максимальное количество одновременных подключений	DINT	<p>Определяет количество подключений к БД, которые создает среда исполнения. Если установлено значение 0, то используется 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 свои подключением:</p> <ul style="list-style-type: none"> <li>• Чтение (если указано &gt; 3 потоков, то чтение может выполняться одновременно из нескольких потоков)</li> <li>• Запись</li> <li>• Удаление старых записей</li> </ul>
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	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.
--------	--	---

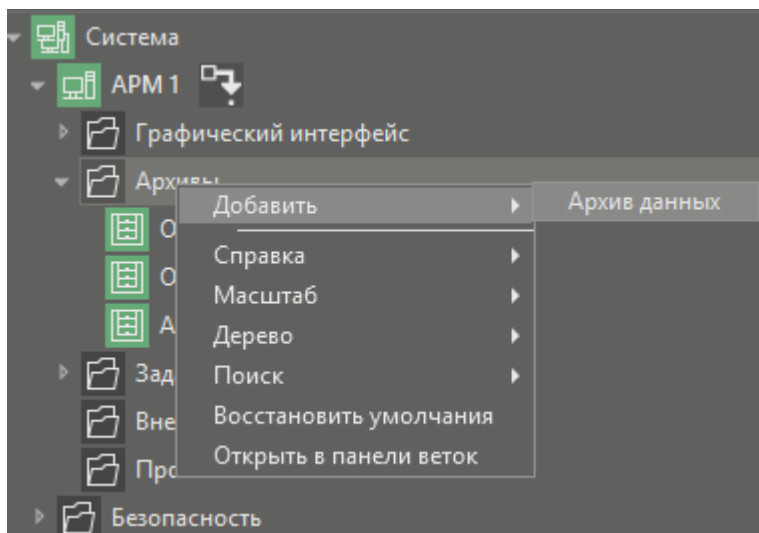
### 6.3.1.7.3. Элемент Архив данных

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:

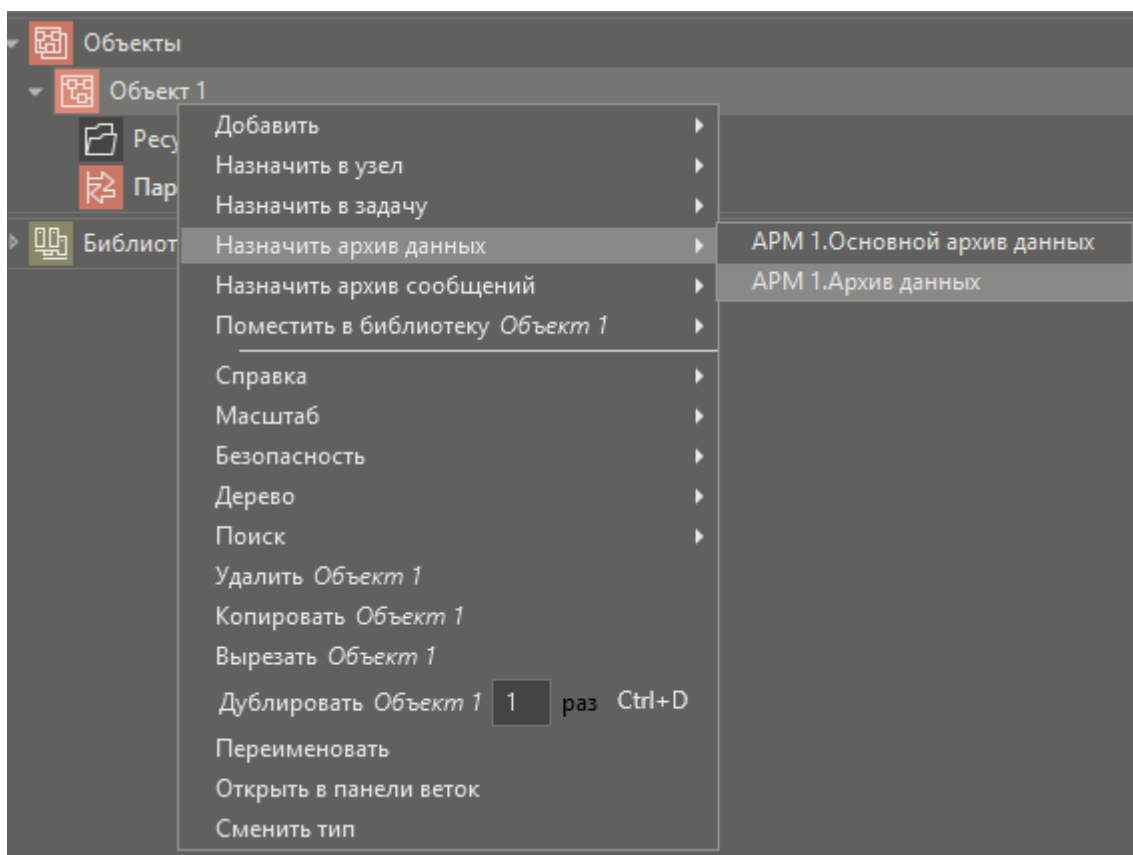


Либо при помощи контекстного меню группы узла Архивы:



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:



#### 6.3.1.7.4. 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.

#### 6.3.1.7.4.1. SQLITE

База данных SQLITE3 входит в состав MasterSCADA 4D. Эта база данных используется для хранения архивов данных и сообщений по умолчанию. При работе с демо-версией среды исполнения, которая входит в состав среды разработки, файл с архивом данных можно найти по адресу: <профиль пользователя>\AppData\Roaming\InSAT\<продукт>\Debug\_<имя проекта>\<имя узла>\PLC\data.db.

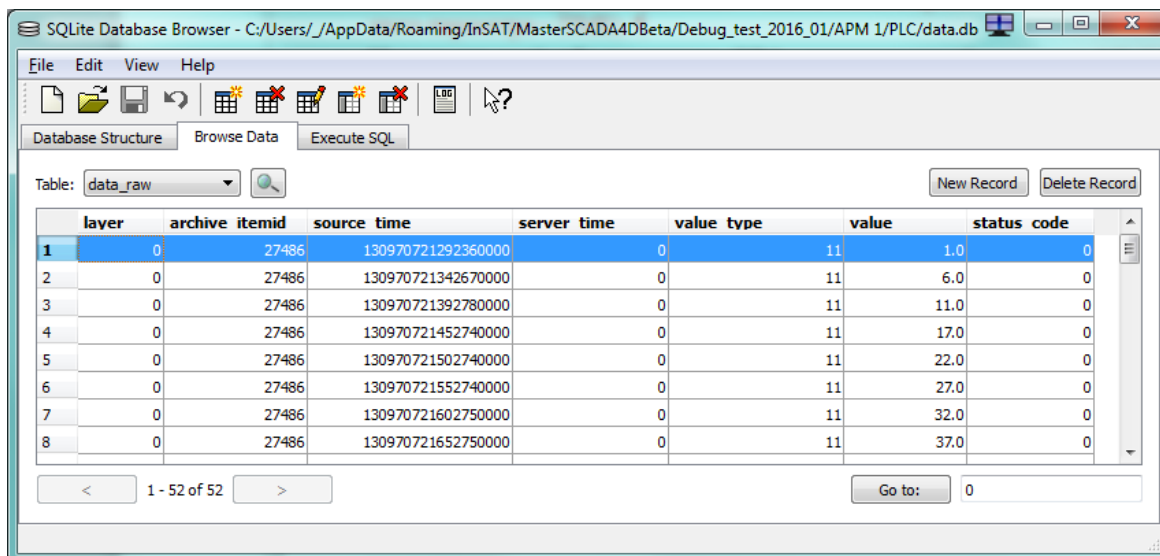
**Важно!** База данных может находиться только на том же устройстве, на котором установлена среда исполнения.

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:





SQLite Database Browser - C:/Users/\_/AppData/Roaming/InSAT/MasterSCADA4DBeta/Debug\_test\_2016\_01/APM 1/PLC/data.db

File Edit View Help

Database Structure Browse Data Execute SQL

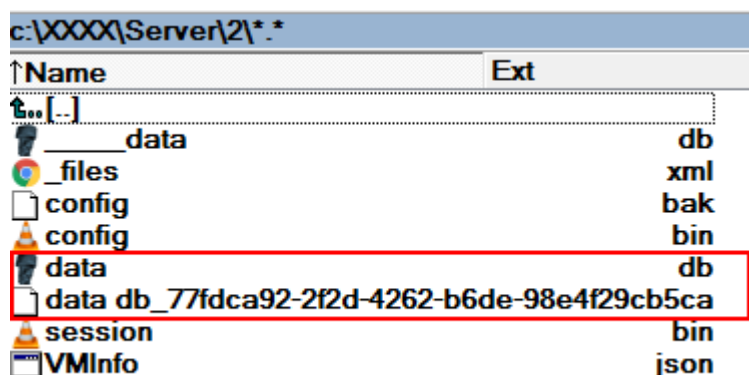
Table: data\_raw

layer	archive itemid	source time	server time	value type	value	status code
1	0	27486	130970721292360000	0	11	1.0
2	0	27486	130970721342670000	0	11	6.0
3	0	27486	130970721392780000	0	11	11.0
4	0	27486	130970721452740000	0	11	17.0
5	0	27486	130970721502740000	0	11	22.0
6	0	27486	130970721552740000	0	11	27.0
7	0	27486	130970721602750000	0	11	32.0
8	0	27486	130970721652750000	0	11	37.0

1 - 52 of 52

Go to: 0

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:



c:\XXX\Server\2\\*.\*

Name	Ext
data	db
_files	xml
config	bak
config	bin
data	db
data db_77fdca92-2f2d-4262-b6de-98e4f29cb5ca	
session	bin
VMInfo	json

### 6.3.1.7.4.2. 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:

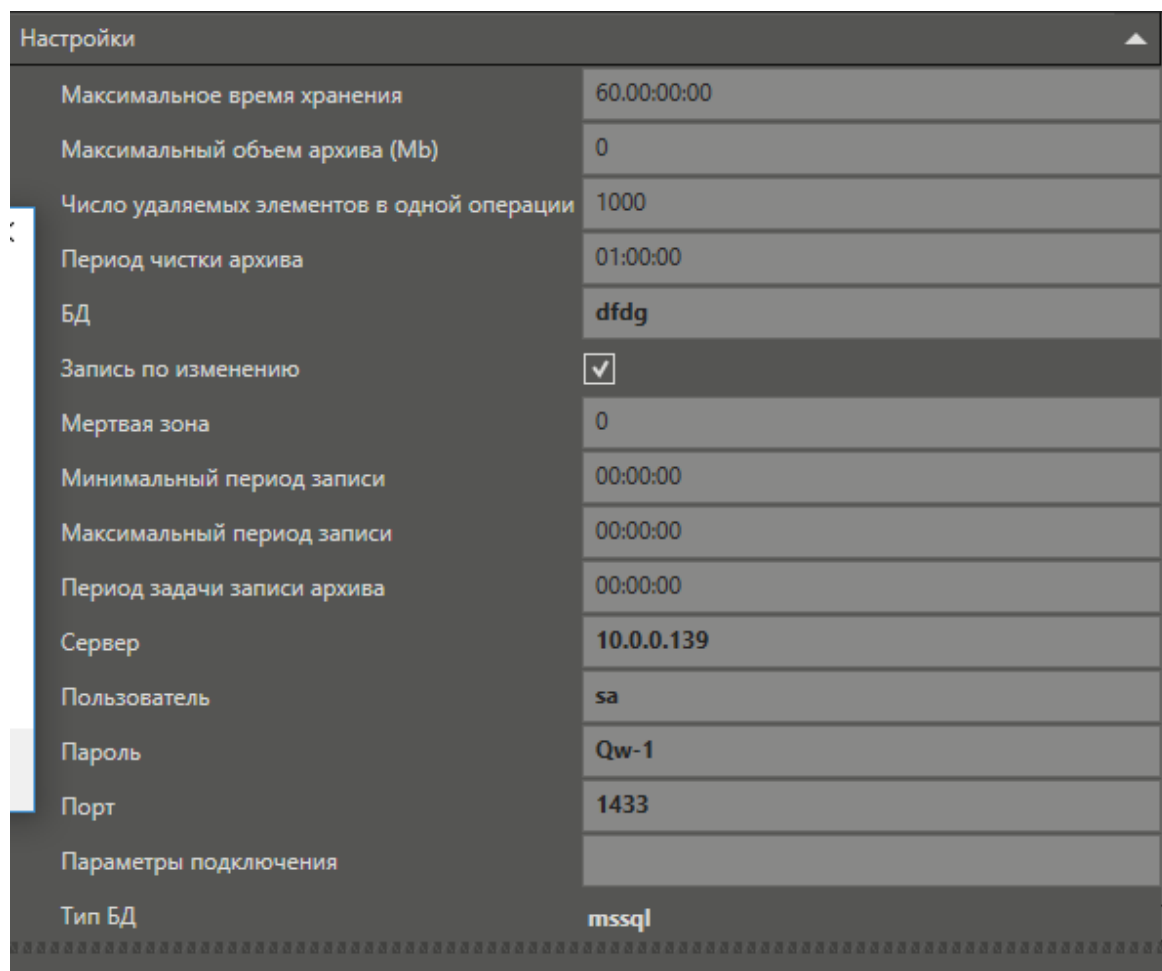
Свойство	Рекомендация
БД	Имя ранее созданной базы данных PostgreSQL.
Сервер	Указывается IP-адрес компьютера, на котором установлена база данных.
Пользователь	Имя пользователя, который добавлен в настройках самой базы данных. Соответствует LoginRole на сервер Postgre.
Пароль	Пароль пользователя, который 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 PostgreSQL используется порт 5432.
Тип БД	Выбирается база данных PostgreSQL.

При первом подключении к базе данных будут созданы все необходимые таблицы и другие необходимые для корректной работы элементы.

### 6.3.1.7.4.3. 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.
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.

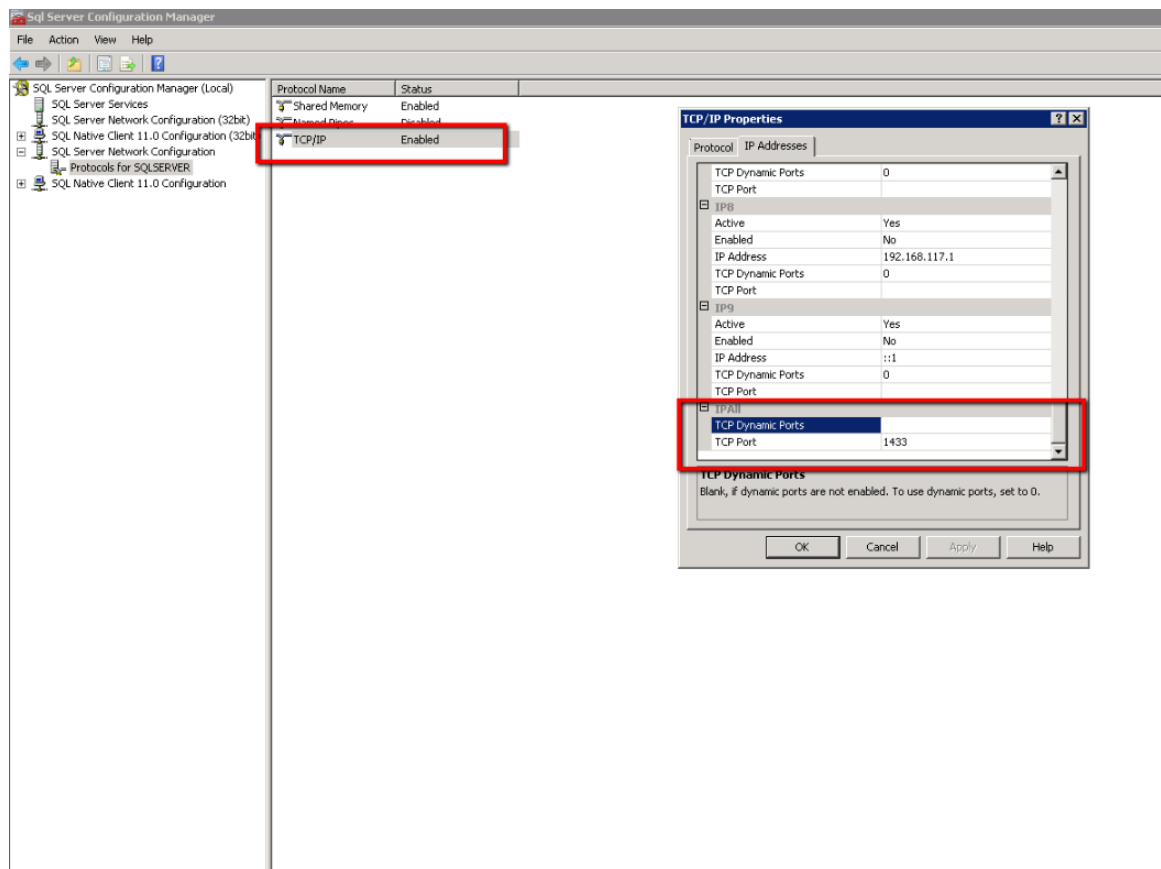
Настройки, которые необходимо сделать непосредственно в базе данных описаны в разделе Настройки базы данных MSSQL

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:



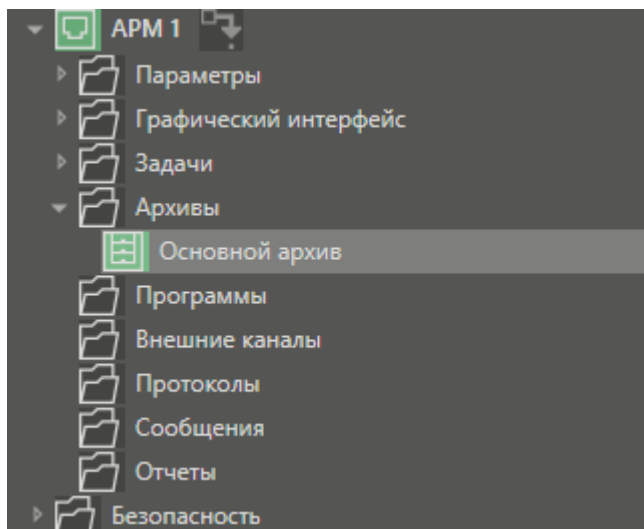
### 6.3.1.7.5. Конфигурирование архива данных

Архивы данных в MasterSCADA 4D хранятся в базах данных.

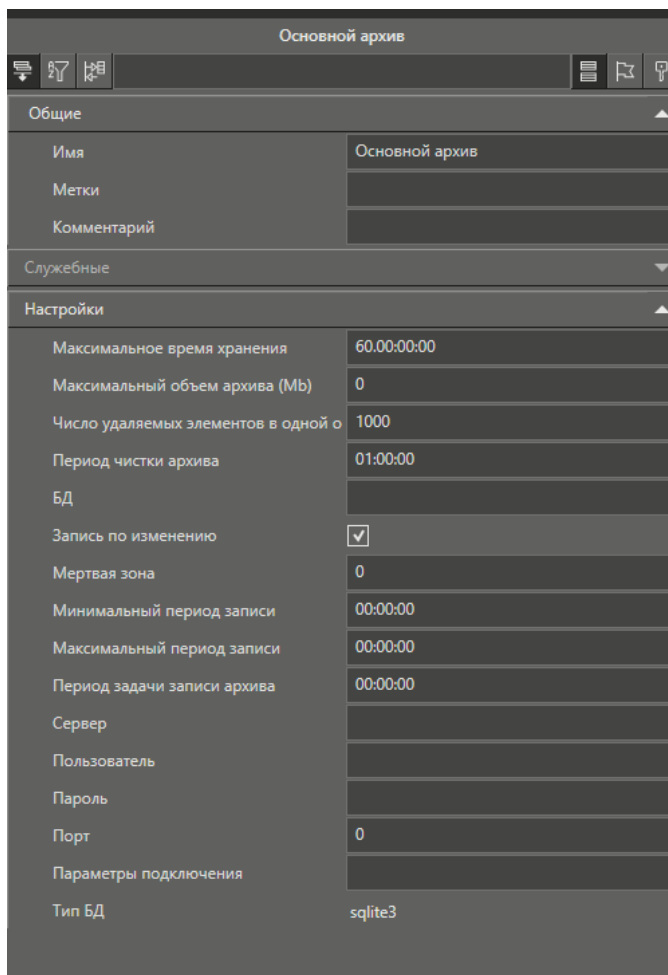
Прежде чем приступать к разработке проекта, в котором потребуется архивировать данные, необходимо убедиться в том, что устройство, на котором устанавливается среда исполнения MasterSCADA 4D, имеет технические средства для хранения архивов.

В текущей версии могут архивироваться параметры объектов, узлов и каналов, 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:



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 SQ LITE, необходимо выбрать соответствующий пункт.

### 6.3.1.7.5.1. 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.

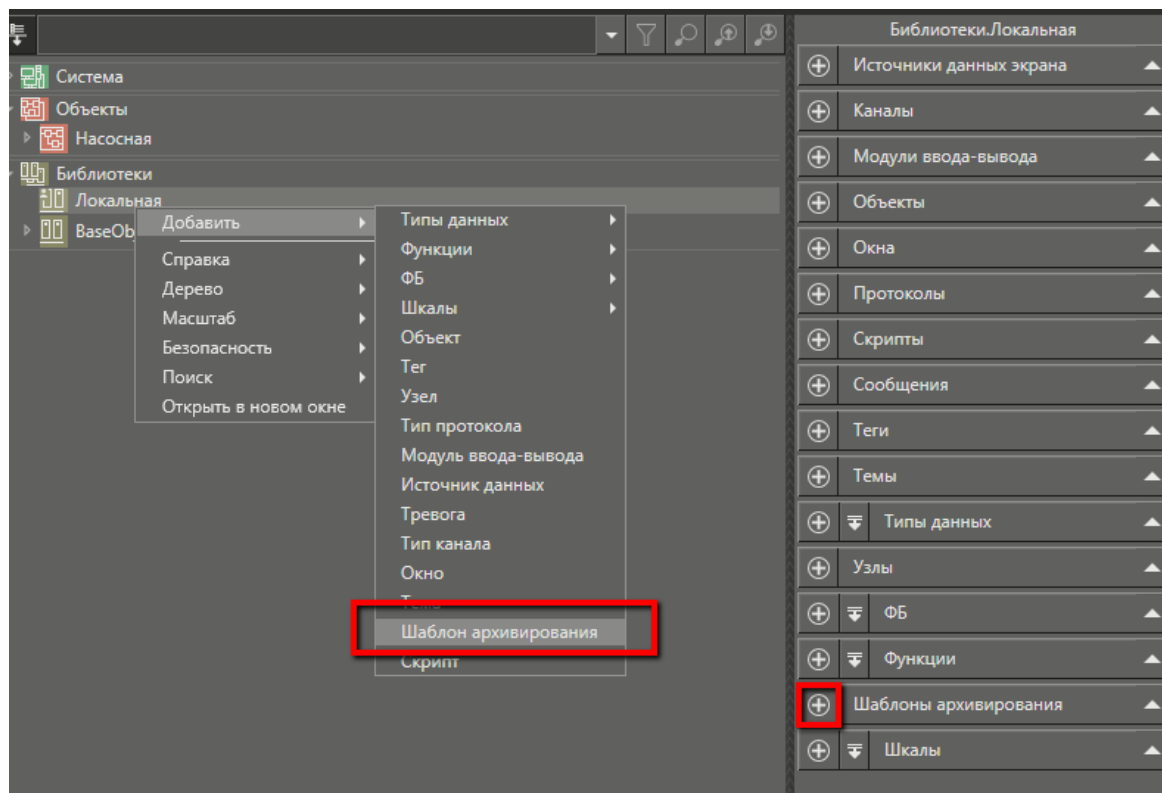
### 6.3.1.7.5.2. Шаблоны архивирования

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.

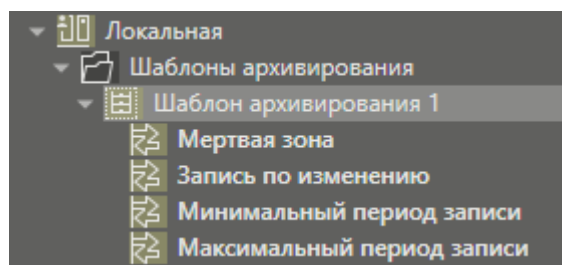
#### Создание шаблона архивирования

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

Метки:

Комментарий:

Служебные

Параметры архивирования

Мертвая зона: 0

Запись по изменению:

Минимальный период записи: 0ms

Максимальный период записи: 0ms

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.
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.

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.

### 6.3.1.7.5.3. 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.

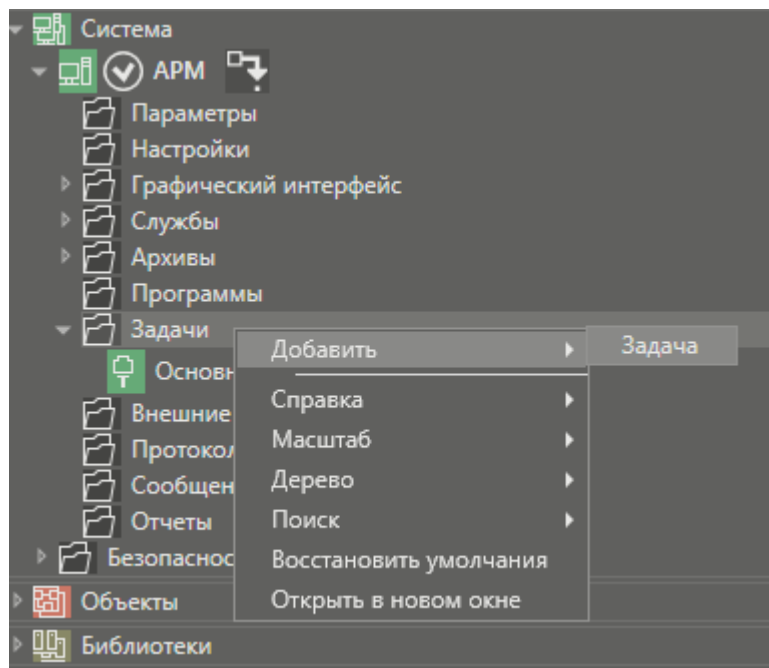
#### Передача архивных данных между параметрами одного узла

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 братья 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.

### 6.3.1.8. Задачи

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 программы ST, 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.

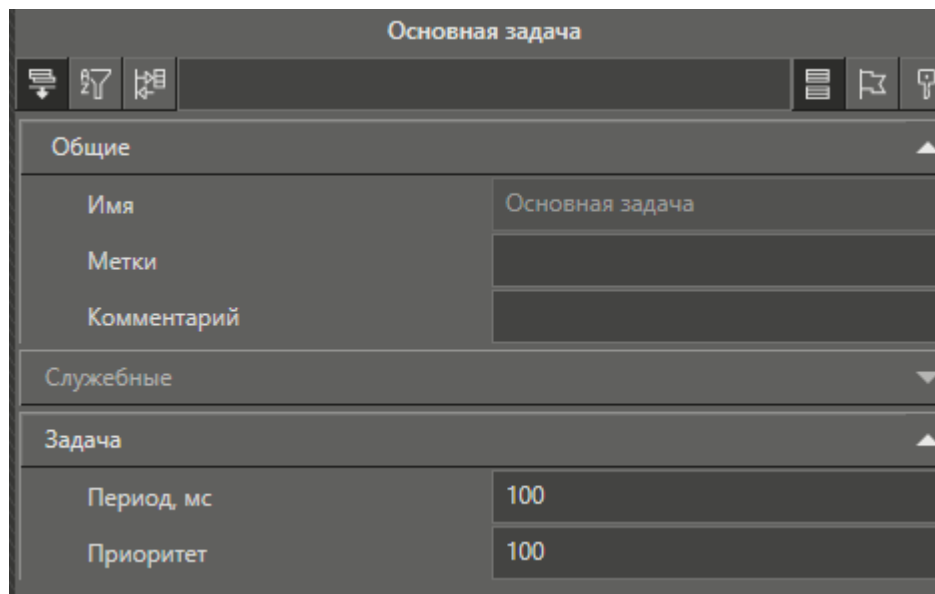


По умолчанию, данная группа содержит один элемент Основная задача. В основную задачу попадают все программы, которые находятся непосредственно в дереве системы в узле, а также объекты, в контекстном меню которых выбран один из пунктов меню:

- 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.

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]

### 6.3.1.9. 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.

### **6.3.1.10. 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).

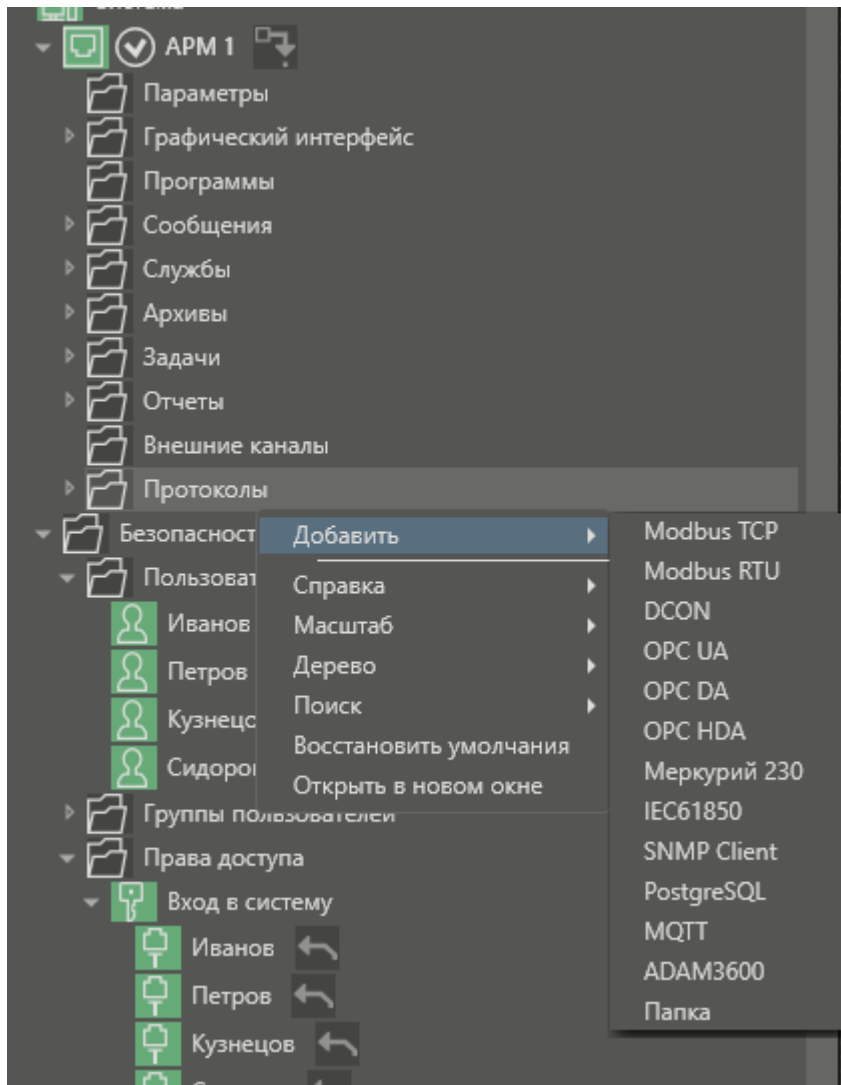
Т.е. в данном случае MasterSCADA 4D выступает в роли Slave.

### **6.3.1.11. 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.

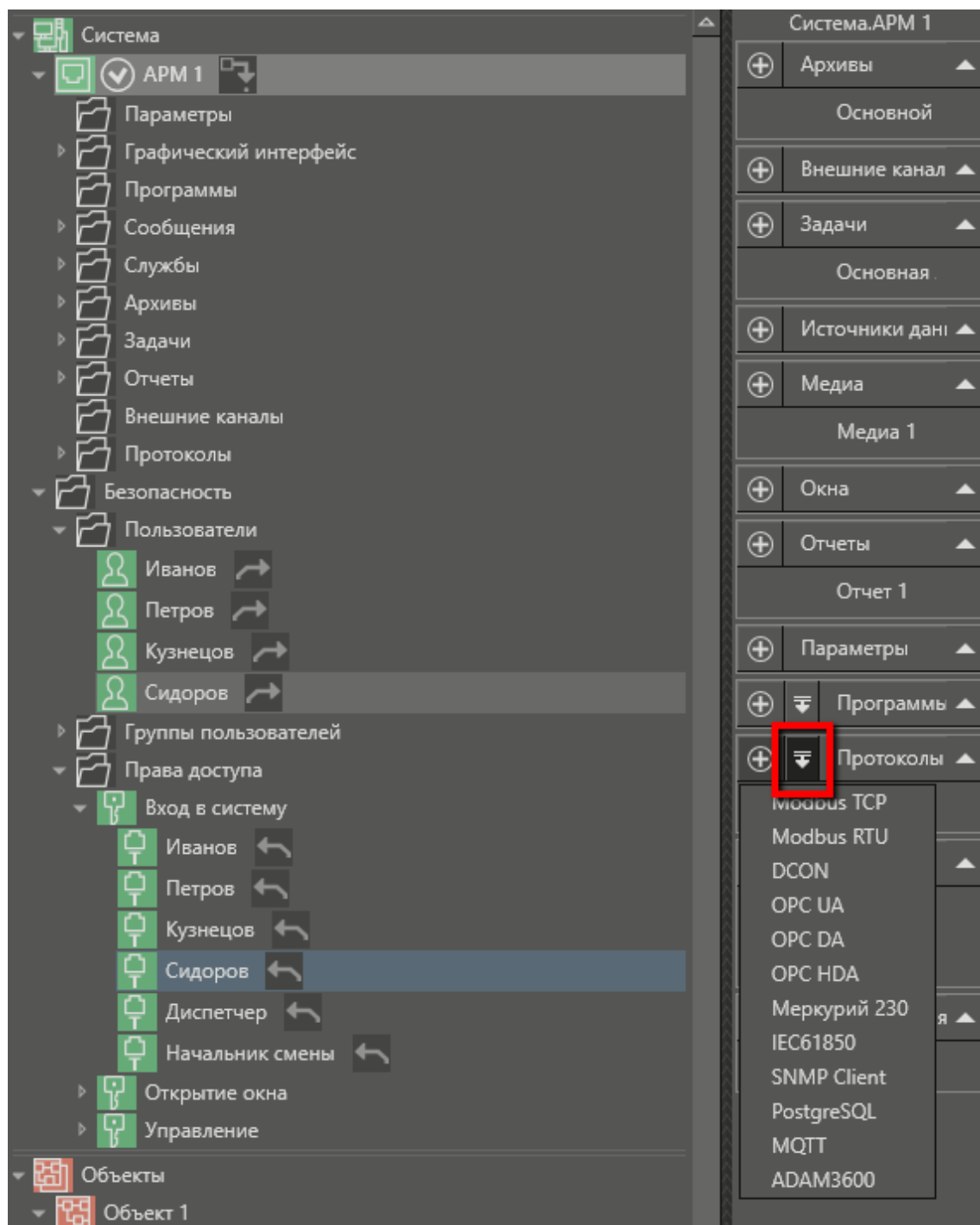
Чтобы добавить протокол, можно воспользоваться одним из двух способов:

- Выбрать требуемый протокол в контекстном меню:



- Добавить требуемый протокол, используя контекстную панель:

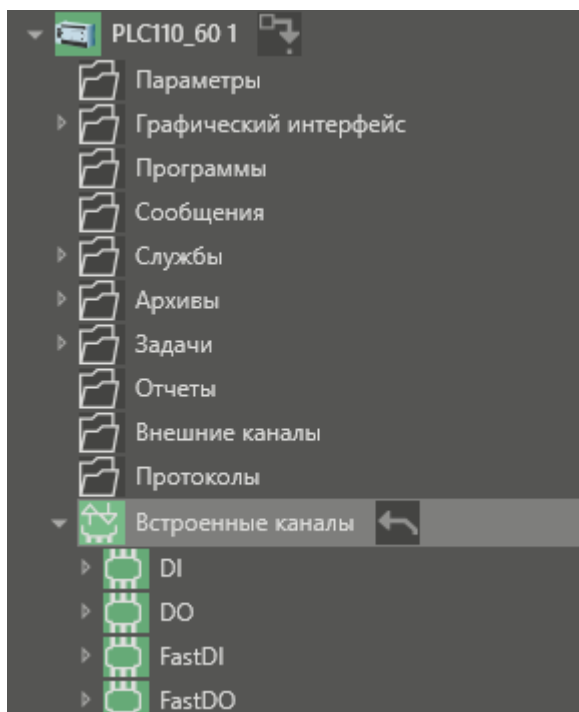




В списке возможных для добавления протоколов будут присутствовать те элементы, которые предназначены для работы с другими устройствами и ПО в режиме Master.

### 6.3.1.12. Встроенные каналы

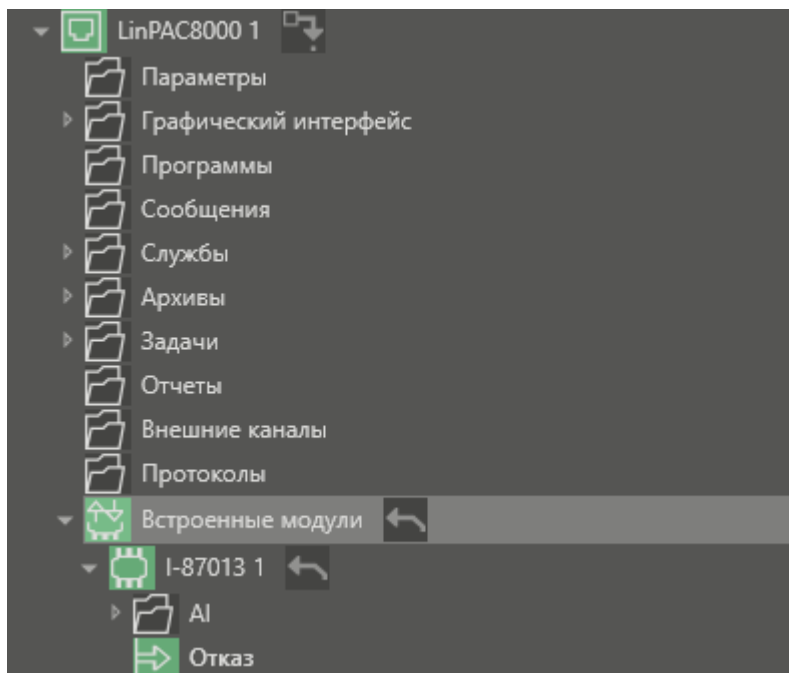
Группа Встроенные каналы присутствует только у определенных узлов.



В этой группе содержатся каналы для получения данных из встроенных физических средств ввода-вывода контроллера, панели оператора.


### 6.3.1.13. Встроенные модули

Группа Встроенные модули присутствует только у определенных узлов, если в кресте 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.

#### 6.3.1.14. 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.

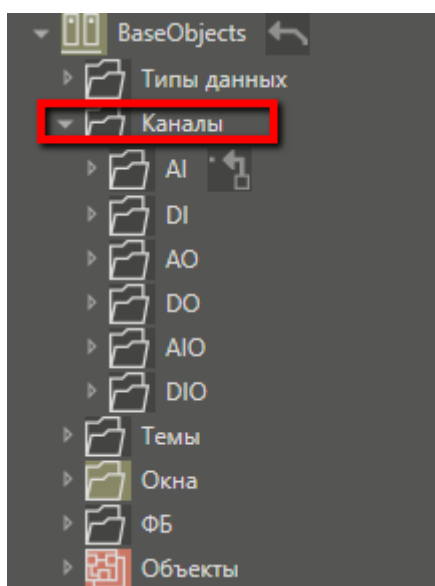
Настройки у каналов, принадлежащих различным протоколам, различны. В зависимости от типа используемого протокола, разработчику проекта будет дана возможность добавить канал с требуемыми настройками.

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.

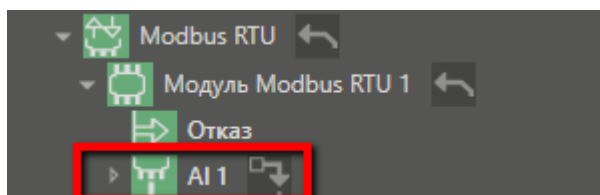
Значение свойства	Параметры канала
Input	Вход
Output	Выход
InOut	Вход, Выход

В библиотеке MasterSCADA 4D представлен ряд каналов predetermined типов, наиболее часто встречающихся в проектах.

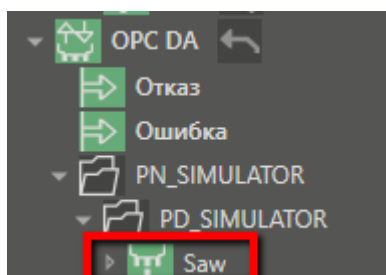


Примеры добавленных каналов в дерево Системы:

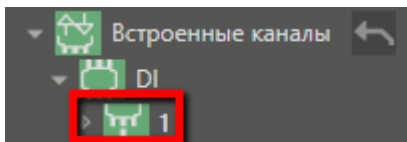
- Канал в модуле Modbus RTU:



- Канал в OPC DA-сервере:



- Канал группы Встроенные каналы:

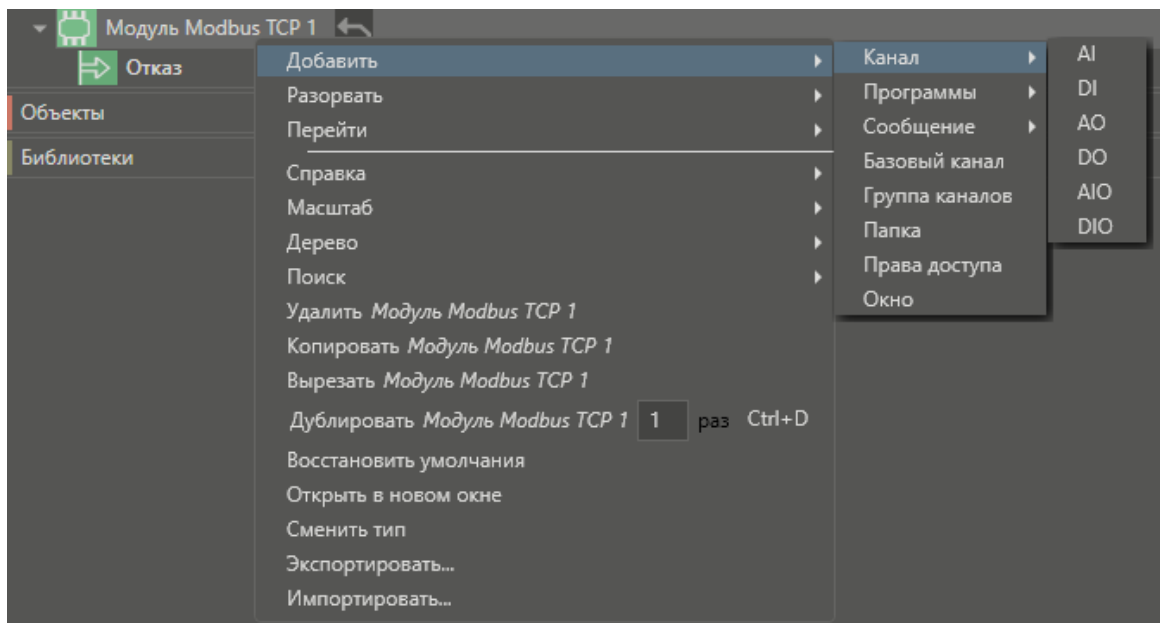


## Способы добавления каналов в проект

### 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:

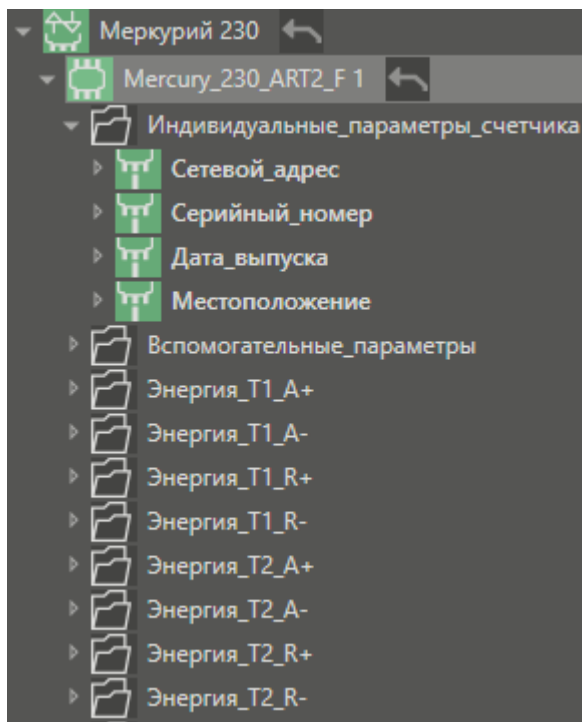


### Диалог добавления

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, посвященным работе с OPC-серверами).

### Автоматическое добавление

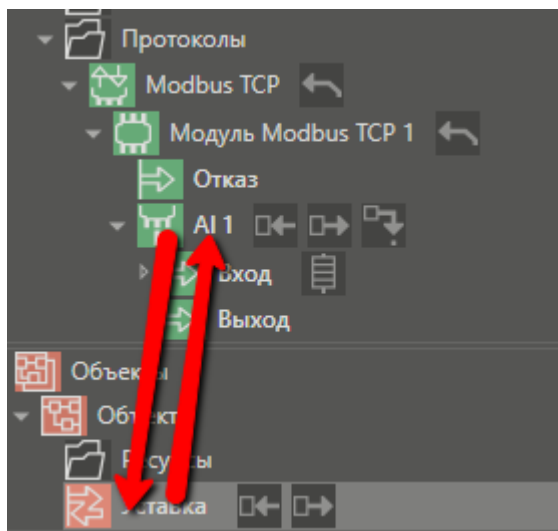
Если в проект добавляется библиотечный модуль ввода-вывода, соответствующий реальному оборудованию, которое поддерживается в MasterSCADA 4D, то каналы добавятся автоматически после добавления модуля в проект :



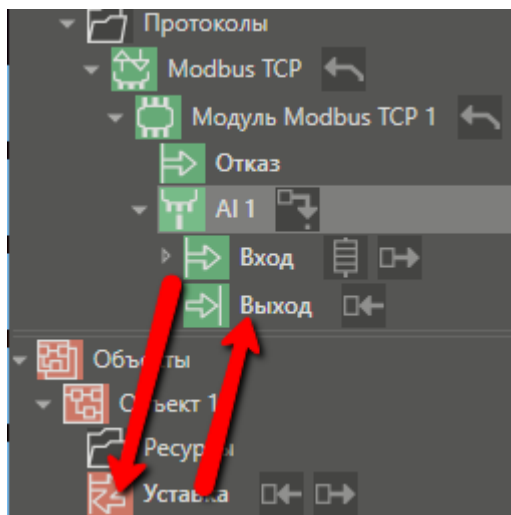
При добавлении предопределенного узла, имеющего встроенные физические входы-выходы, каналы также автоматически добавятся в группе Встроенные каналы.

#### 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).

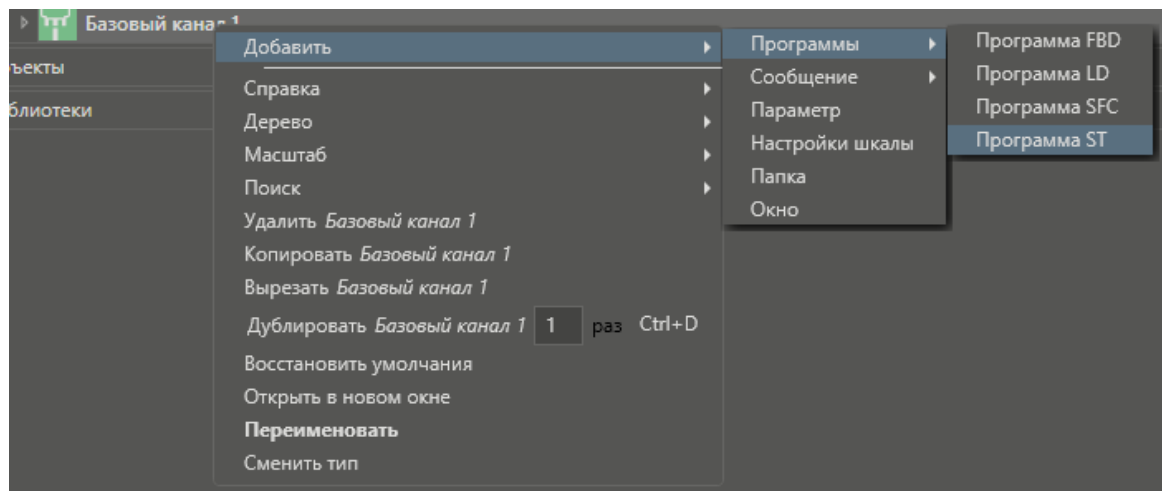


Это будет равнозначно тому, как если бы связь была с параметрами канала:



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:



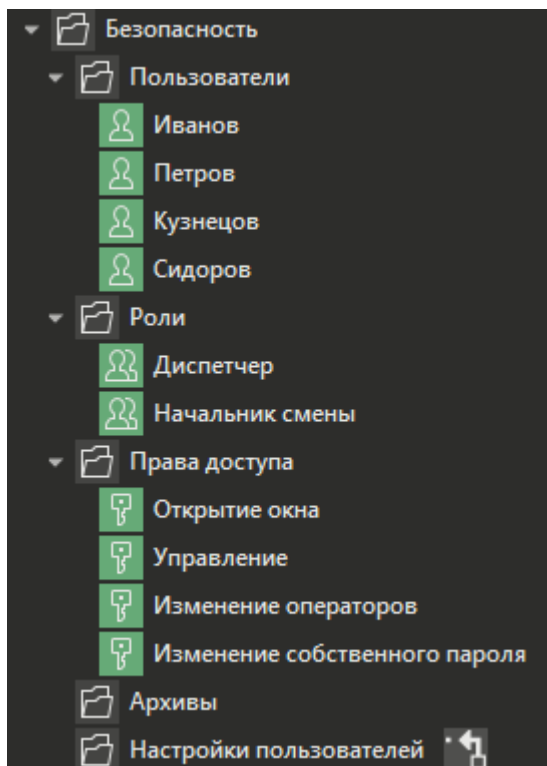
В этом случае у канала появится группа Ресурсы, такая же как и у Объектов и Тегов. Непосредственно в канал можно также добавить сообщения, окна и другие элементы, характерные скорее для объектов. В этом случае, канал будет отличаться от объекта только возможностью получать данные от устройств и других 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.

### 6.3.2.Безопасность

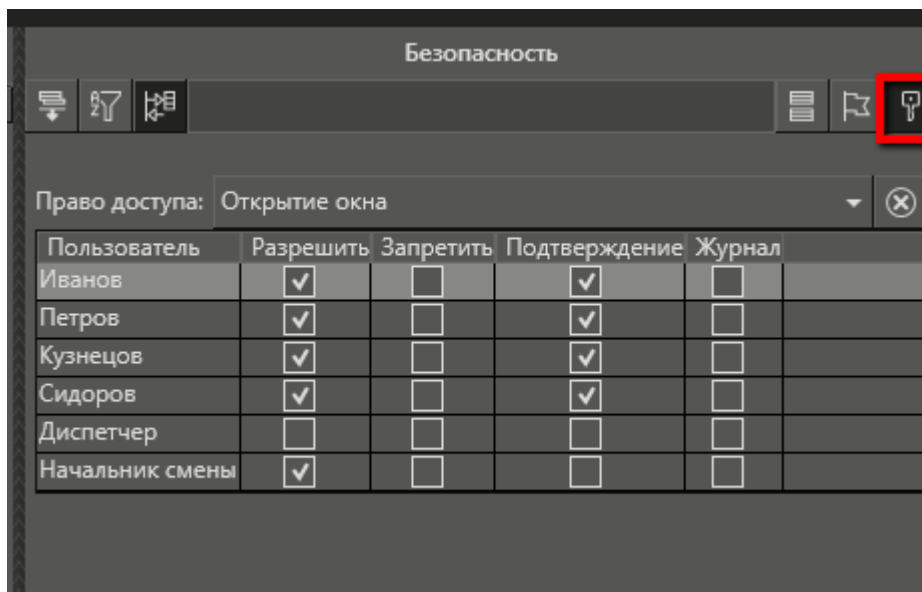
В настройках данной группы задаются параметры проверки целостности программного обеспечения и проекта, а также в данной группе содержатся элементы для настройки прав доступа операторов (пользователей) в режиме исполнения.

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



Список прав доступа:

Название	Назначение
----------	------------

Открытие окна	Позволяет пользователю открывать окна в режиме исполнения в клиенте визуализации. Если у пользователя нет прав на открытие окна, то появится соответствующее сообщение.
Управление	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
Изменение операторов	Позволяет изменять параметры пользователей в режиме исполнения
Изменение собственного пароля	Позволяет изменить собственный пароль пользователя, который авторизован в текущий момент в клиенте визуализации

### 6.3.2.1. Панель свойств Безопасность

Панель свойств элемента Безопасность имеет вид:

**Безопасность**

Общие

Имя: Безопасность

Полное имя: Система.Безопасность

Метки:

Комментарий:

Служебные

Настройки

Контроль целостности проекта

Блокировать запуск при неуспешной проверке проекта

Период проверки целостности проекта: 00:00:00

Контроль целостности ПО

Блокировать запуск при неуспешной проверке ПО

Период проверки целостности ПО: 00:00:00

Системная настройка: 0

Адрес сервера Active Directory:

Название	Описание
Контроль целостности проекта	Активирует контроль целостности проекта. Если флаг установлен, то в конфигурацию проекта включается зашифрованный файл <code>cfg_files.dat</code> (шифрование AES 256 фиксированным ключем) со списком MD5 контрольных сумм всех загружаемых из среды разработки файлов конфигурации узла (включая папки <code>cfg</code> и <code>htdocs</code> )
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 <code>cfg_files.dat</code> , 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 файлов выдается как системное сообщение в журнале.
Период проверки целостности проекта	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.
Контроль целостности ПО	Включает контроль целостности программных файлов режима исполнения (включая <code>mplc</code> , <code>*.so</code> , папки <code>nginx</code> , <code>nodejs</code> ). Создается два файла со списком контрольных сумм:  Для контроля папок <code>nginx</code> , <code>nodejs</code> . Они не обновляются из среды разработки. Этот файл создается при установке дополнительной системы.  Для контроля <code>*.so</code> файлов, которые могут обновляться из среды разработки. Этот файл создается при сборке вместе с <code>*.so</code> файлами и загружается вместе с ними.
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	<p>Указывается имя домена (host) или IP-адрес сервера, где настроен Active Directory. Если свойство задано, то разрешается при авторизации в 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 predetermined 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>

Примеры ошибок, выдаваемых в журнал:

- 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

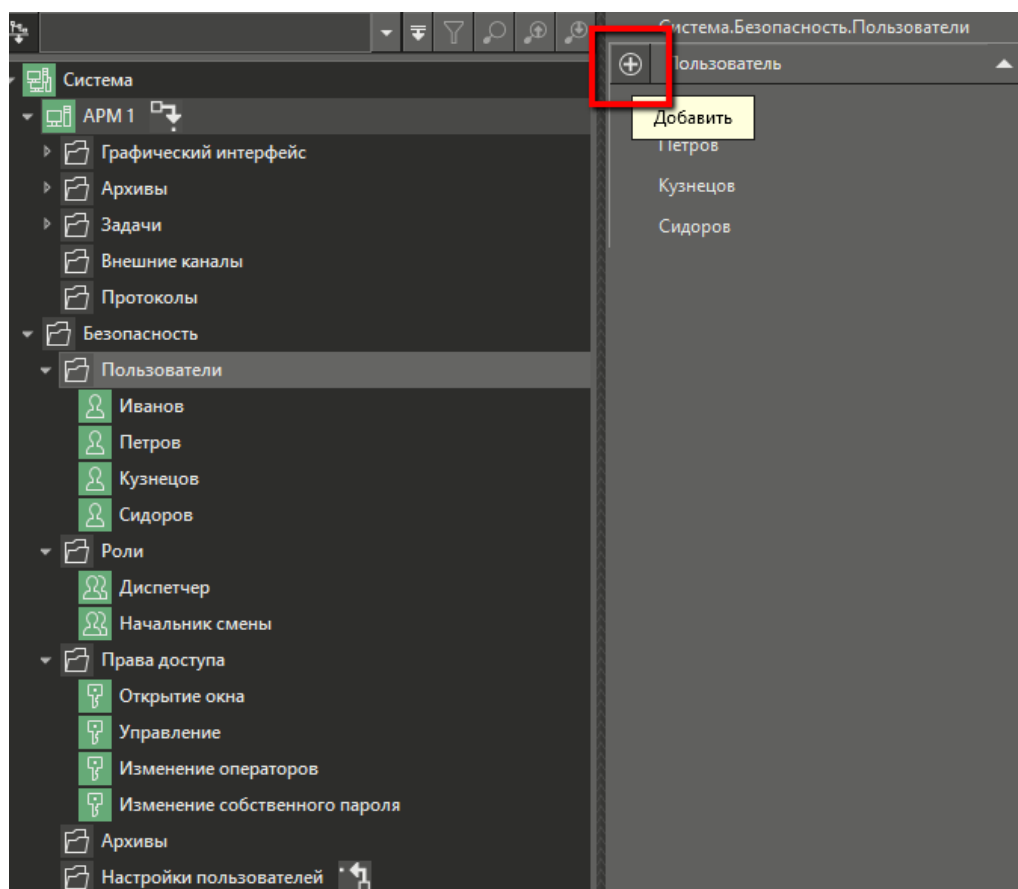
## 5 - Ошибка дешифрации файла

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.

### 6.3.2.2. Пользователи

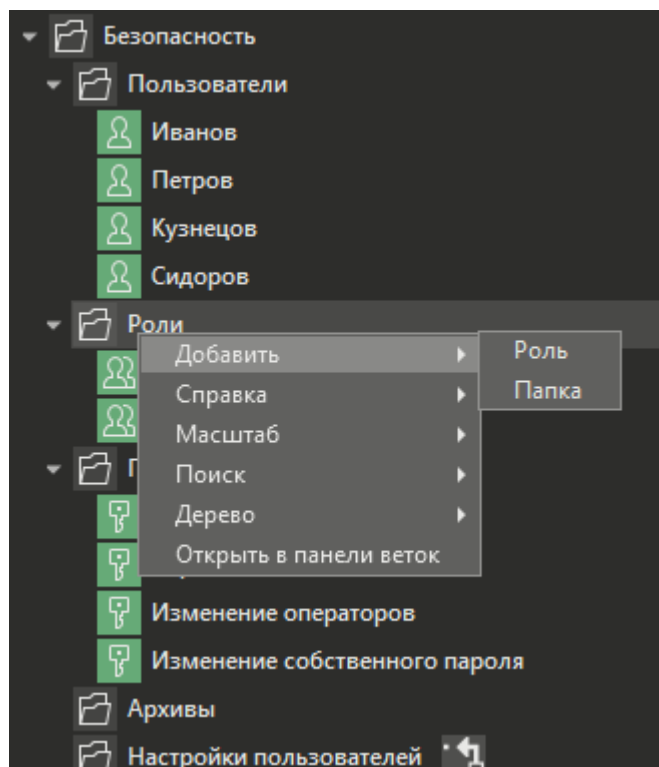
Данная группа предназначена для создания/удаления пользователей проекта, которые будут допущены к управлению и просмотру в режиме исполнения. Для того, чтобы добавить пользователя, необходимо выполнить пункт контекстного меню группы Добавить.Пользователь, либо воспользоваться контекстной панелью.



**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.

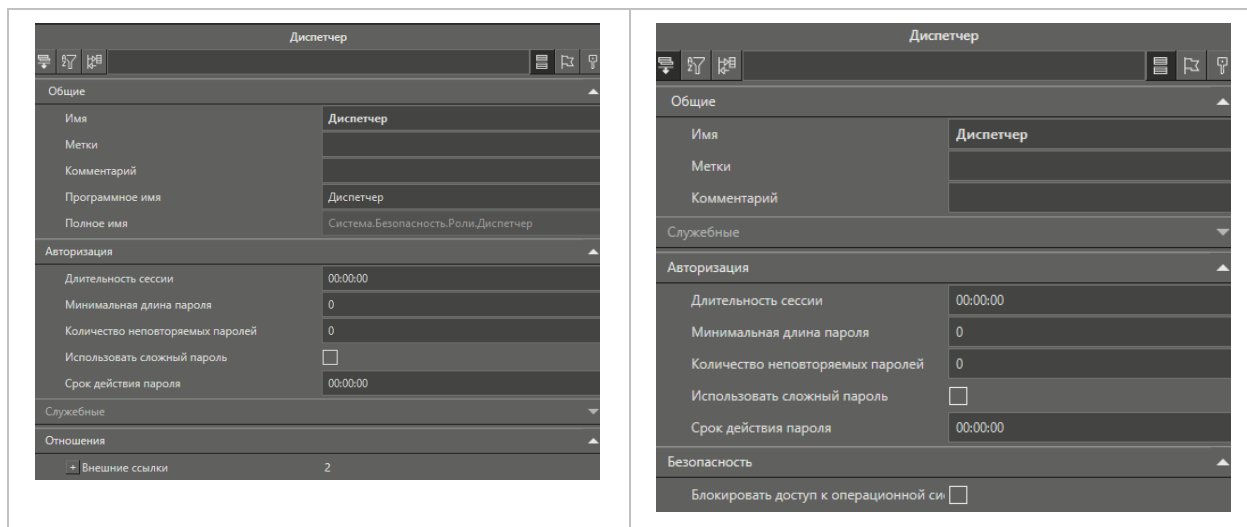
### 6.3.2.3. Роли

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.



Панель свойств Роли имеет вид::

Полный вид	Упрощенный вид (по умолчанию)
------------	-------------------------------



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.
Количество неповторяемых паролей	Определяется, какое количество раз нужно указать пароль отличный от предыдущих при настройке пароля в режиме исполнения. Пустые пароли не учитываются. 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.
Использовать сложный пароль	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.
Срок действия пароля	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

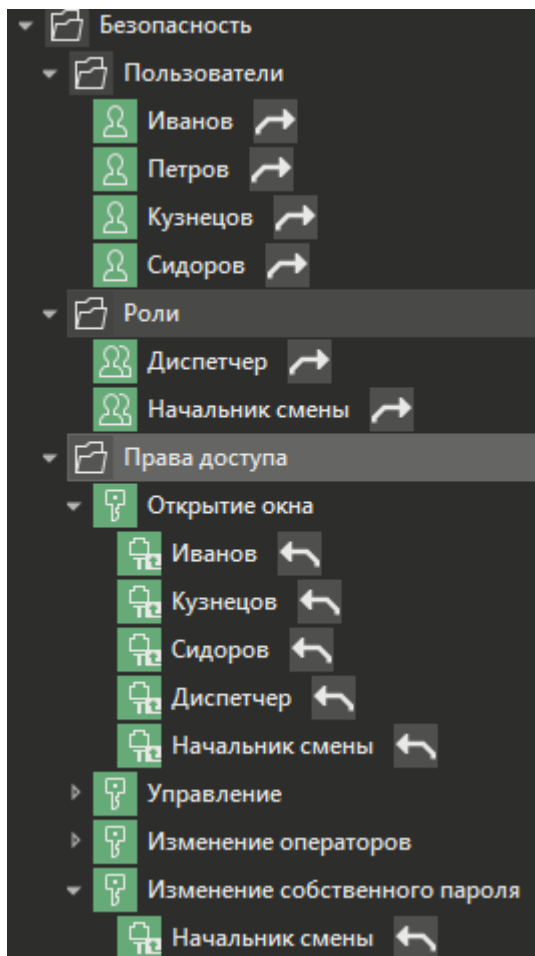
### 6.3.2.4. Права доступа

Данная группа содержит подгруппы-глобальных прав: Открытие окна, Управление, Изменение операторов, Изменение собственного пароля

Название	Назначение
Открытие окна	Позволяет пользователю открывать окна в режиме исполнения в клиенте визуализации. Если у пользователя нет прав на открытие окна, то появится соответствующее сообщение.
Управление	Позволяет нажимать на кнопки, изменять значения, при помощи различных графических элементов, например, Инкремент, Текстовый ввод и др.
Изменение операторов	Позволяет изменять параметры пользователей в режиме исполнения
Изменение собственного пароля	Позволяет изменить собственный пароль пользователя, который авторизован в текущий момент в клиенте визуализации

Данная группа может формироваться вручную. Для того, чтобы разрешить оператору то или иное действие, это действие необходимо перетащить мышью на нужное свойство. Но, как правило, эта группа формируется автоматически. При проставлении флагов на панели свойств Безопасность, в данную группу добавляются ссылки на тех или иных пользователей проекта.

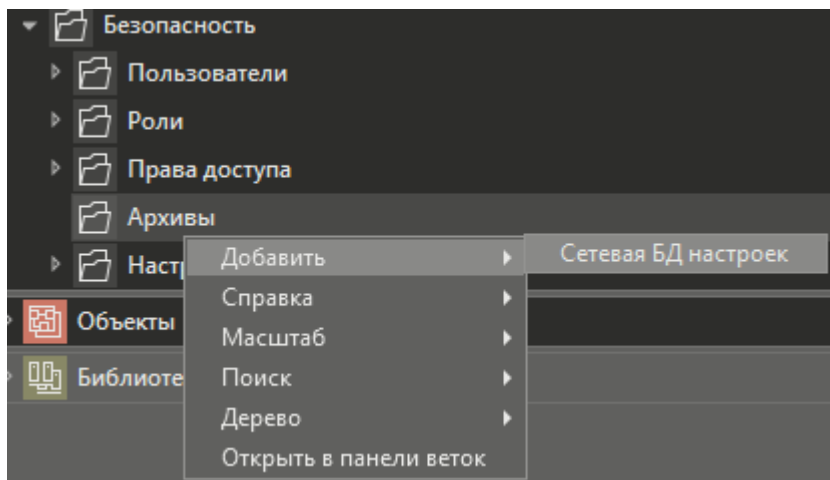




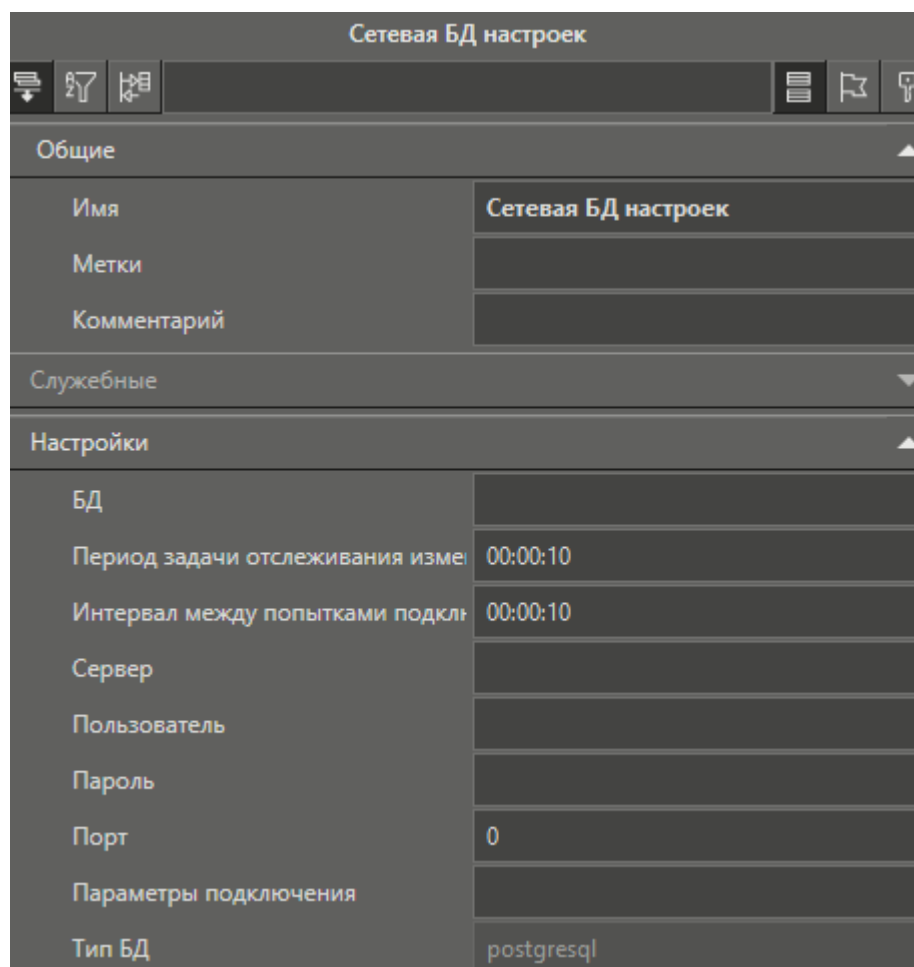
### 6.3.2.5. Архивы

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.)

Для добавления элемента в группу 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 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 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.

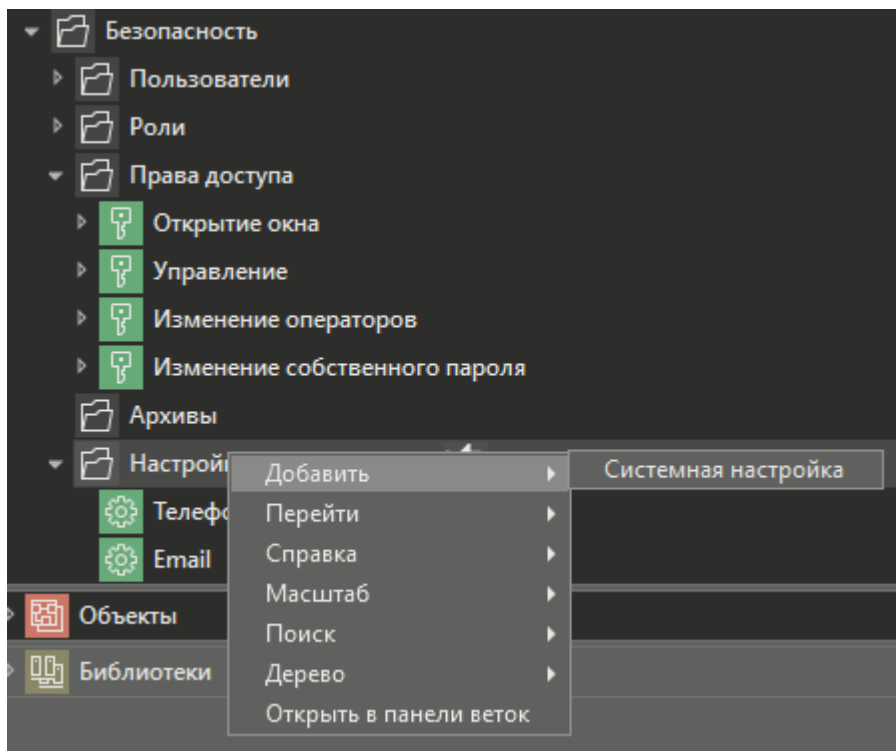
Как в локальном файле, так и во внешней базе данных хранится метка времени, которая генерируется при записи новых данных о пользователях.

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 работает.

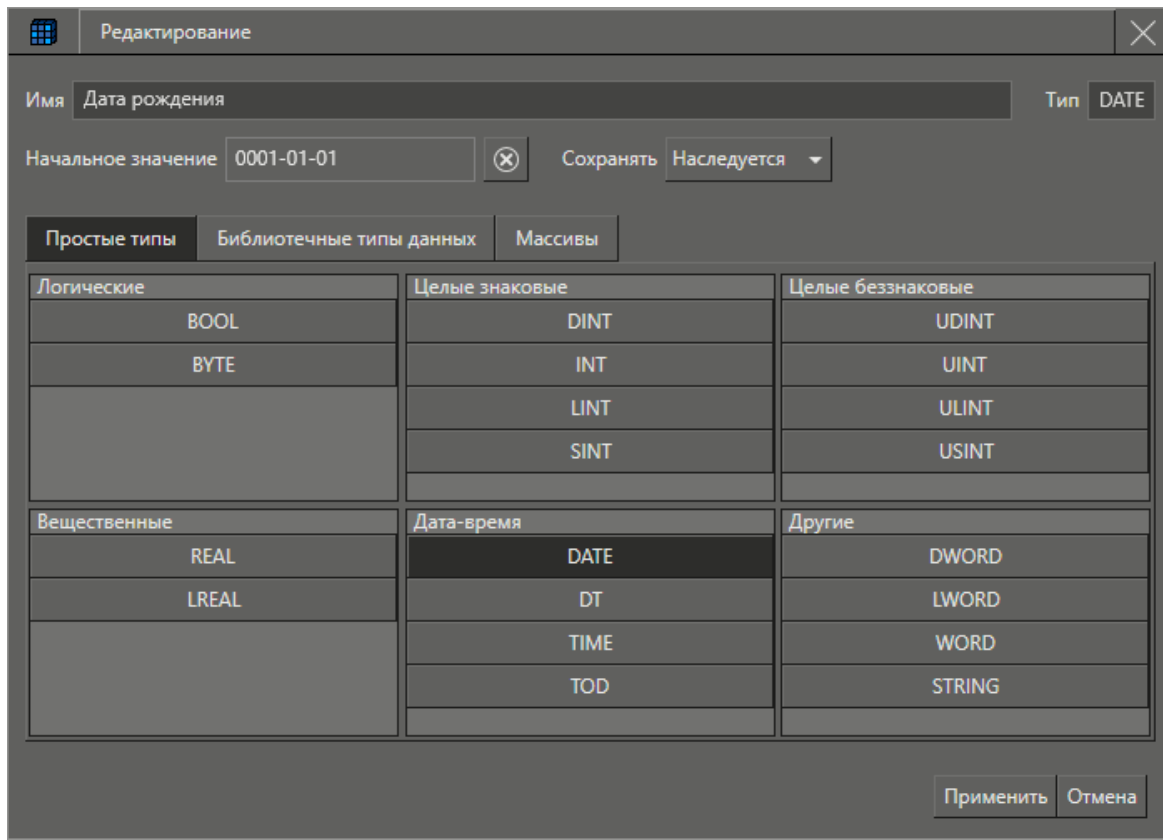
### 6.3.2.6. Настройки пользователей

Группа содержит список настроек, которые можно дополнительно задавать у пользователей.

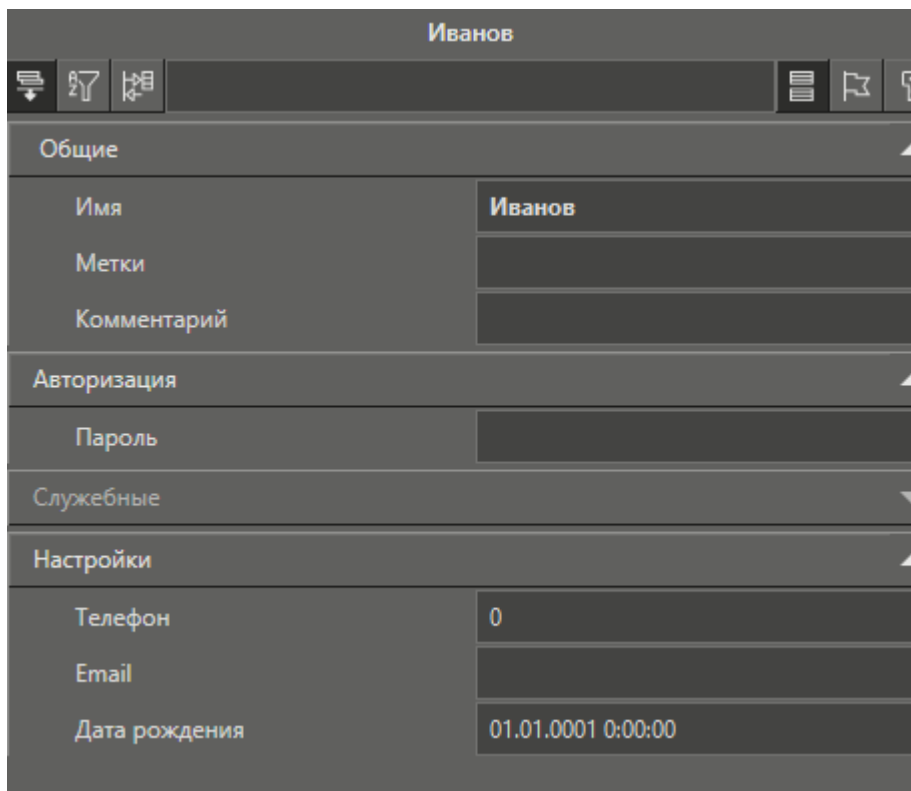
Для добавления нового элемента необходимо выполнить пункт контекстного меню Добавить. Системная настройка:



В окне указать тип и имя новой настройки:



После чего в панели свойств пользователя появятся указанные настройки:



В режиме исполнения получить доступ к данным настройкам можно при помощи функциональных блоков UsersGetSettings и UsersSetSettings

### 6.3.2.7. 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.

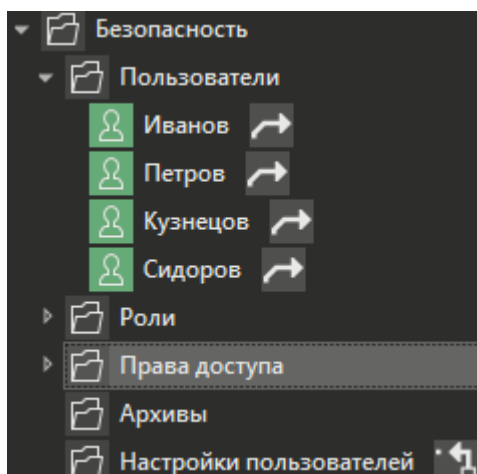
Настраивать права доступа для пользователей можно как индивидуально, так и для группы пользователей.

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.



Панель свойств пользователя имеет вид:

Иванов

Общие

Имя: Иванов

Метки

Комментарий

Служебные

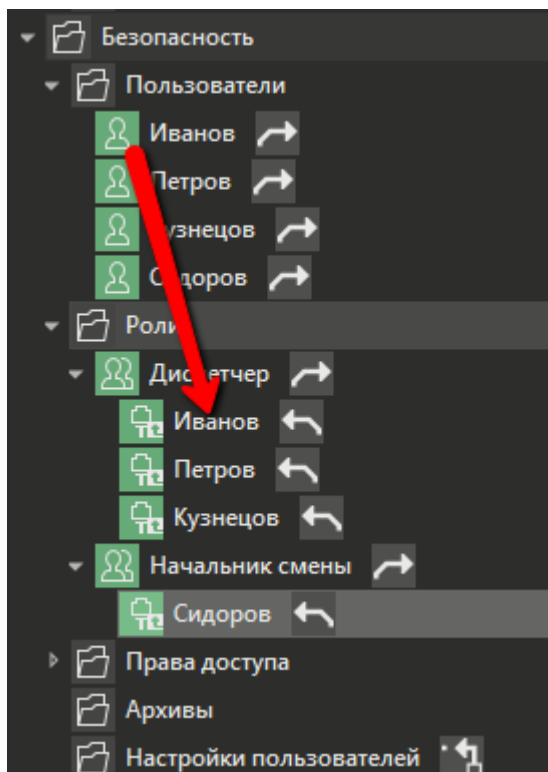
Пользователь

Пароль: d8845

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.

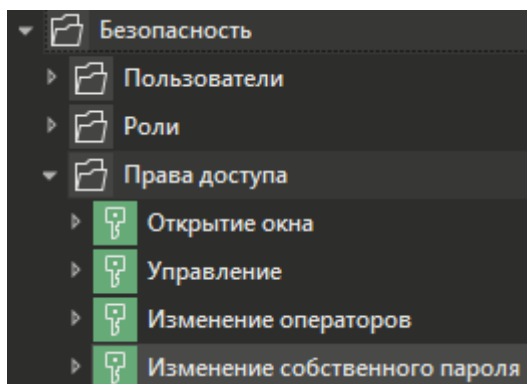
Создание ролей пользователей.

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.




### Настройка глобальных прав доступа

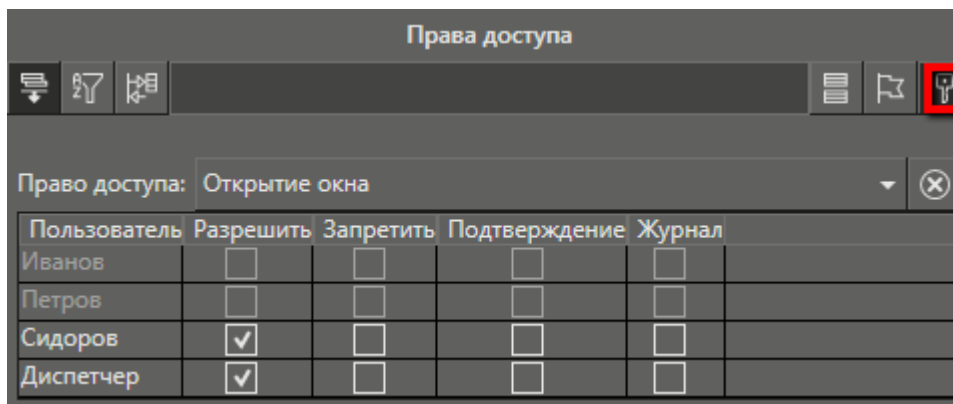
Права доступа можно разделить на два типа: Управление и Открытие окон. Именно эти категории прав доступа находятся в группе Безопасность. Права доступа.



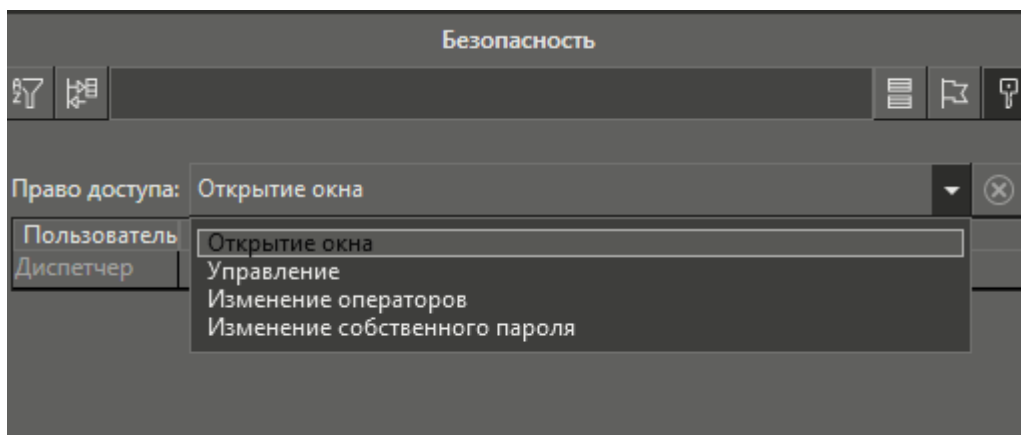
Права, заданные на уровне дерева системы, будут считаться умолчаниями для всех элементов проекта.

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 для всего проекта, независимо от других настроек. 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 действие, появится диалоговое окно:

### Подтвердите действие

Управление




Если нажать на кнопку Отмена, то действие не выполнится.

Для фиксации действий операторов 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.

**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

Если действие на уровне дерева системы не определено, либо разрешено для всех элементов данного типа, но для конкретного элемента его требуется запретить, то тогда у объектов, окон, параметров, кнопок и др элементов, с которыми работает оператор, в панели свойств при нажатии на кнопку  откроется также таблица с правами доступа, где также можно выполнить настройку прав доступа.

Если, например, Объект, содержит дочерние элементы, то настройки, сделанные у этого объекта, будут пронаследованы для всех его дочерних элементов.

### Старт проекта с настроенными правами доступа.

При старте клиента визуализации производится проверка прав доступа. Если у элемента или у его родительского элемента где-либо установлен флаг Запретить, то действие будет запрещено независимо от других настроек.

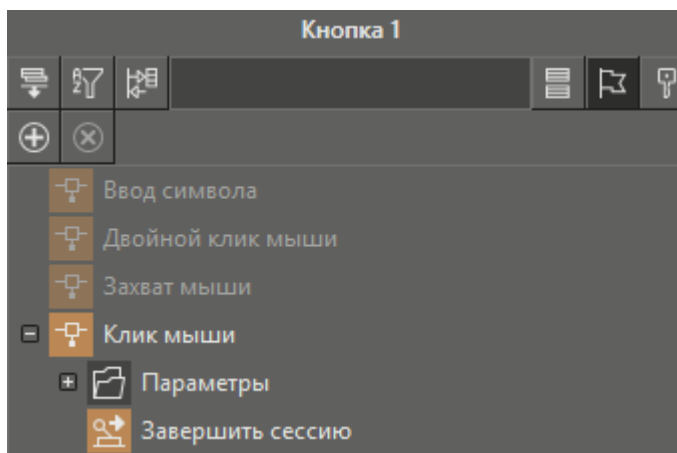
В случае, если оператор принадлежит разным ролям с разными правами доступа, то тогда все настройки будут суммироваться. Т.е. если в какой-либо группе действие запрещено, то пользователь выполнить его не сможет.

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:



Смотрите также:

Получение параметров текущего пользователя

Задание начального окна для пользователя

Журнал действий операторов

Пример настройки прав доступа

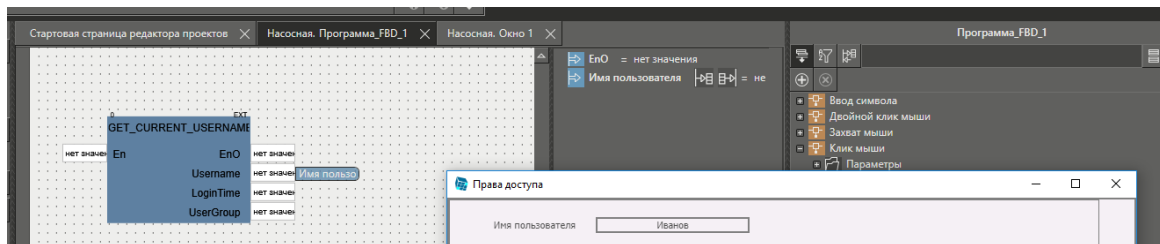
Настройка пользователя по умолчанию

### 6.3.2.7.1. 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.

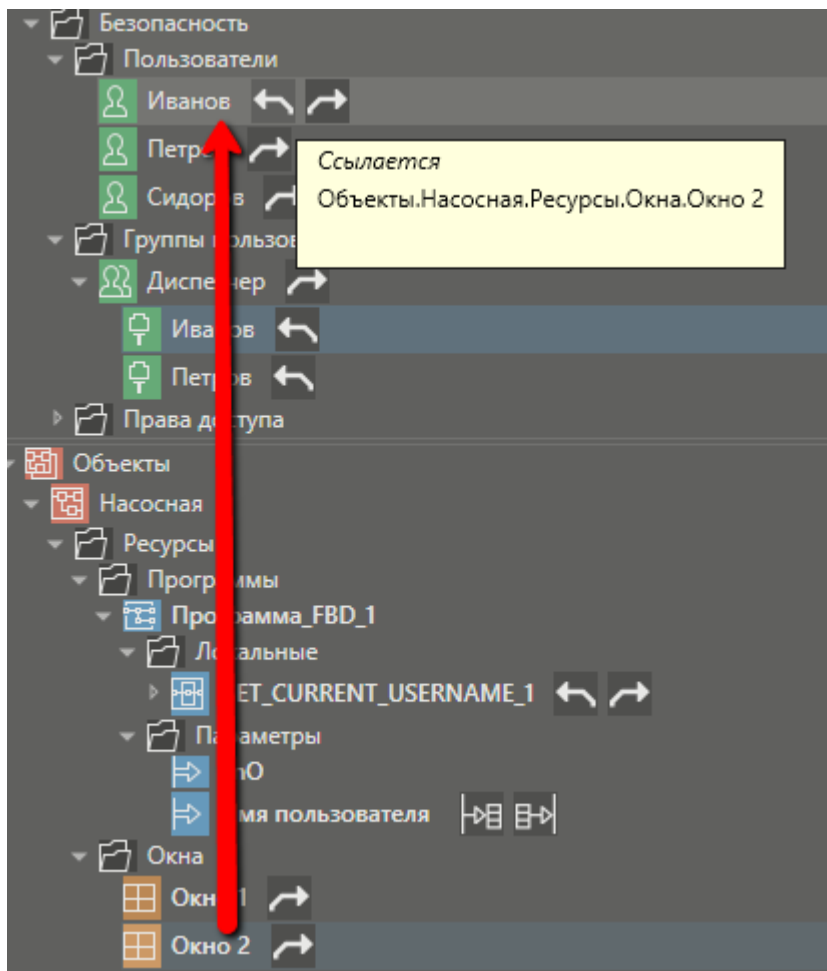
Параметры функционального блока нужно разместить в окне клиента.



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

### 6.3.2.7.2. Задание начального окна 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:



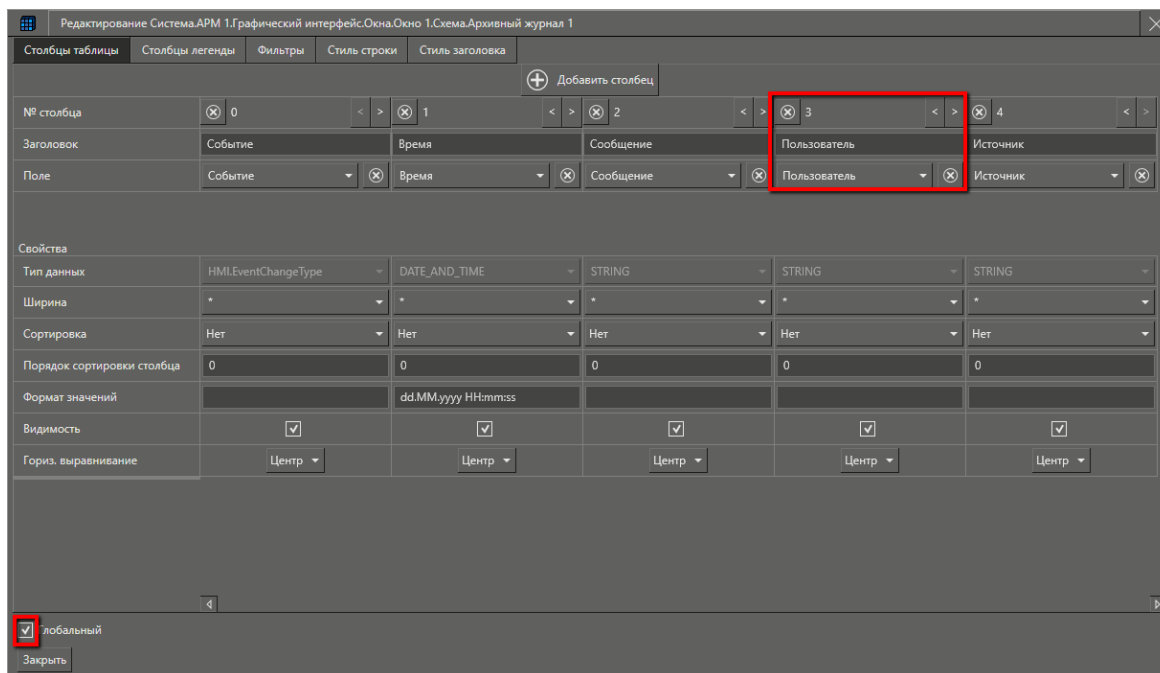
Если начальное окно для пользователя не задано, то при авторизации этого пользователя откроется окно, установленное как стартовое, стартовая мнемосхема.

### 6.3.2.7.3. 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.

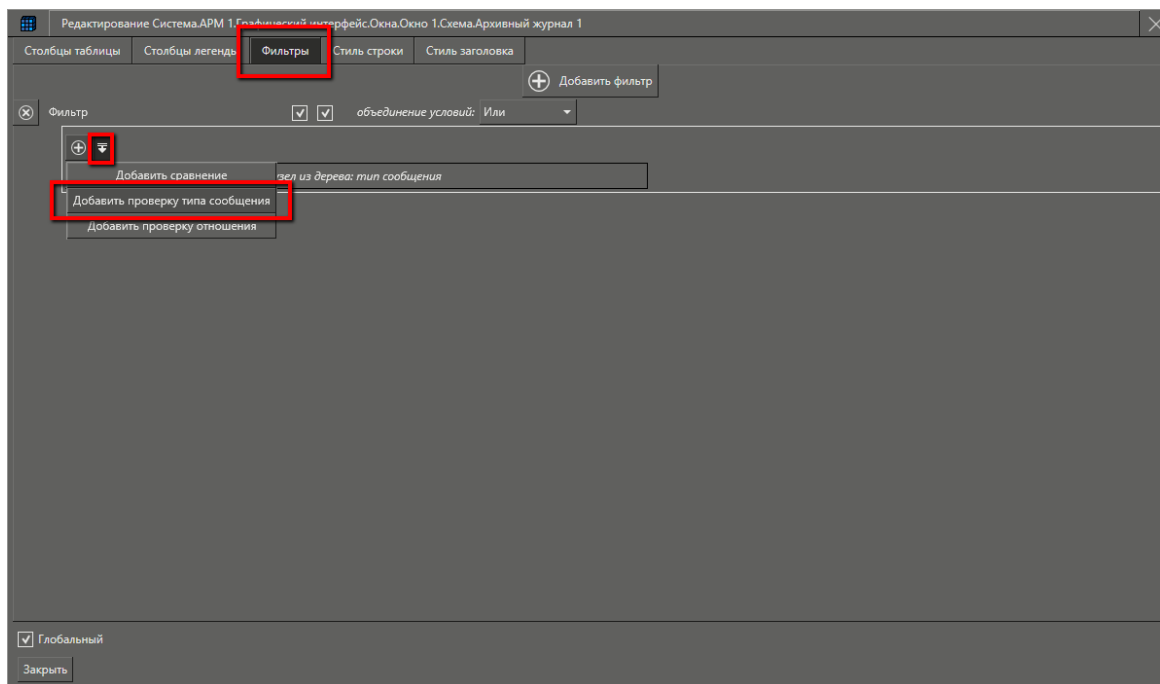
Разместим в окне элемент Архивный журнал. В диалоговом окне настройки журнала добавим столбец, который будет отображать какой Пользователь совершил, то или иное действие:



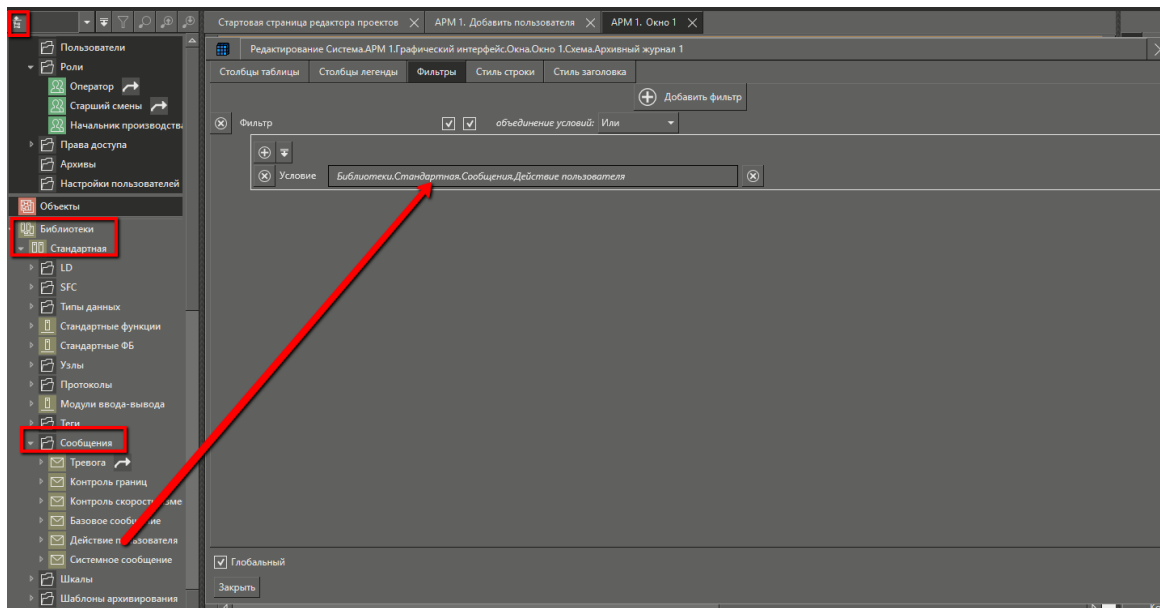
Если установить флаг Глобальный, то в журнал будут попадать все сообщения узла, удовлетворяющие фильтрам.

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

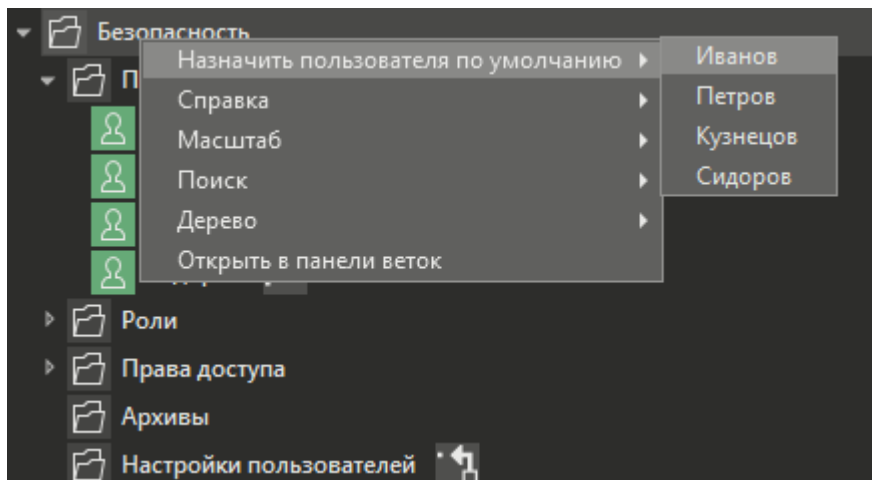


После старта среды исполнения в клиенте визуализации в таком журнале будут отображаться все действия операторов, которые отмечены флагами в настройках прав доступа, а также информация о начале и окончании сессии пользователя.

#### 6.3.2.7.4. Настройка пользователя по умолчанию

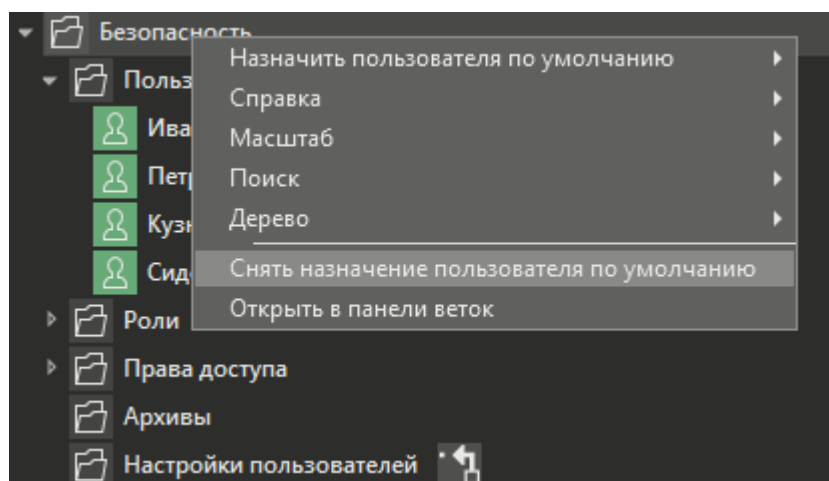
Для автоматического подключения клиента визуализации к исполнительной системе, минуя диалоговое окно авторизации, имеется возможность настроить пользователя по умолчанию.

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:



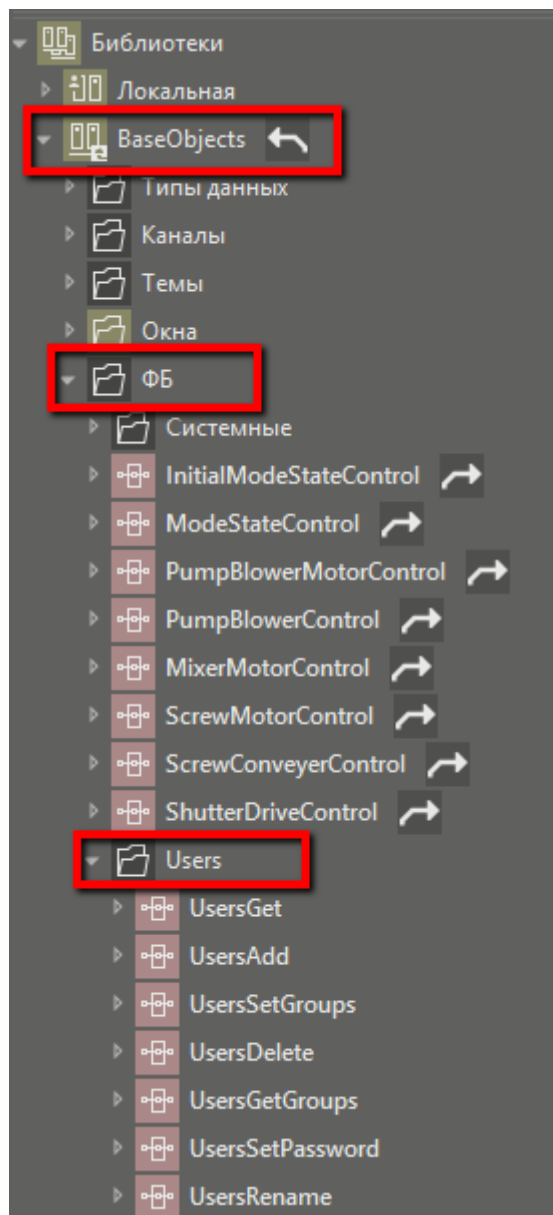
### 6.3.2.8. Работа с пользователями в режиме исполнения

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

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 исполняется 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

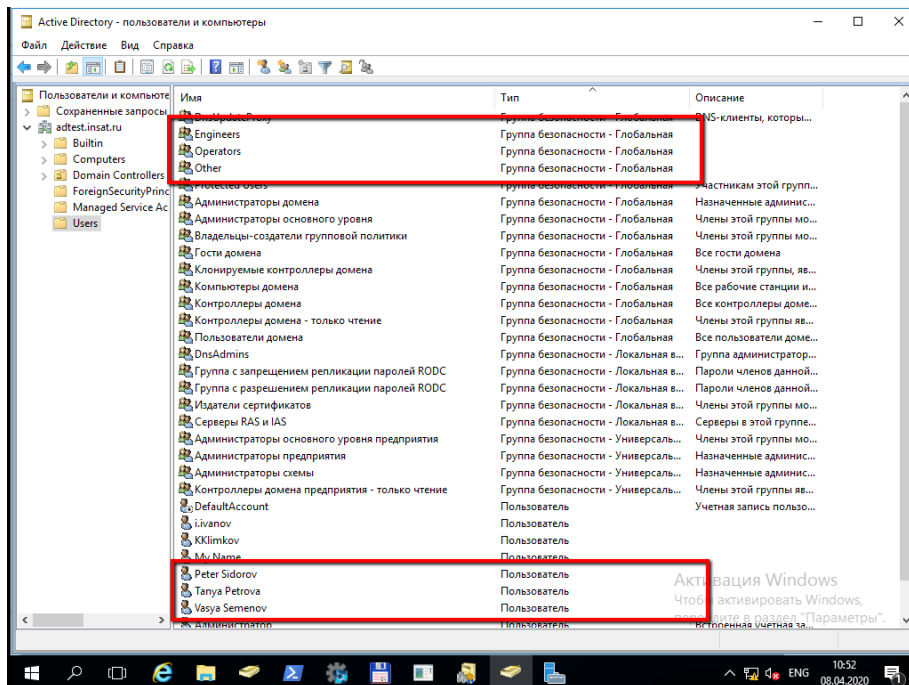
### 6.3.2.9. Active Directory

Active Directory (AD) — это технология управления компьютерами и устройствами в сети. 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>.

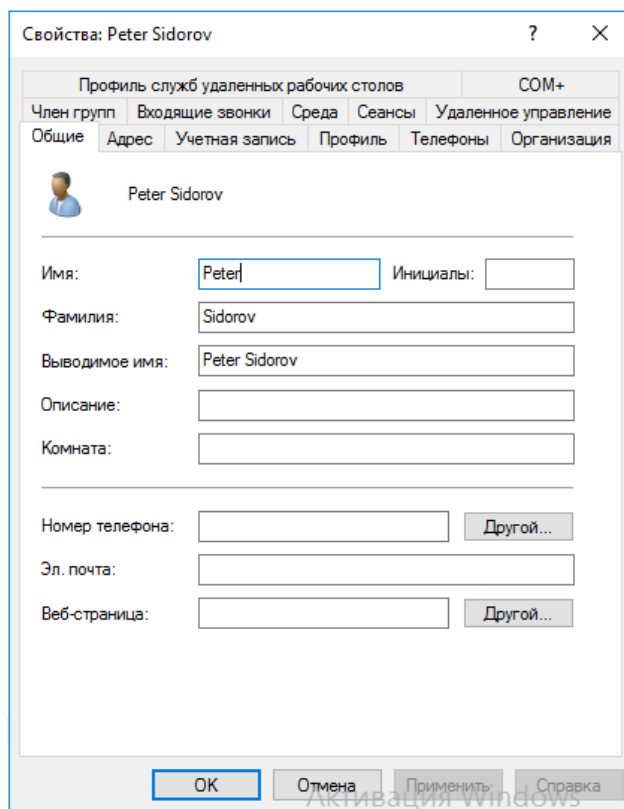
Пример настроек Active Directory Windows

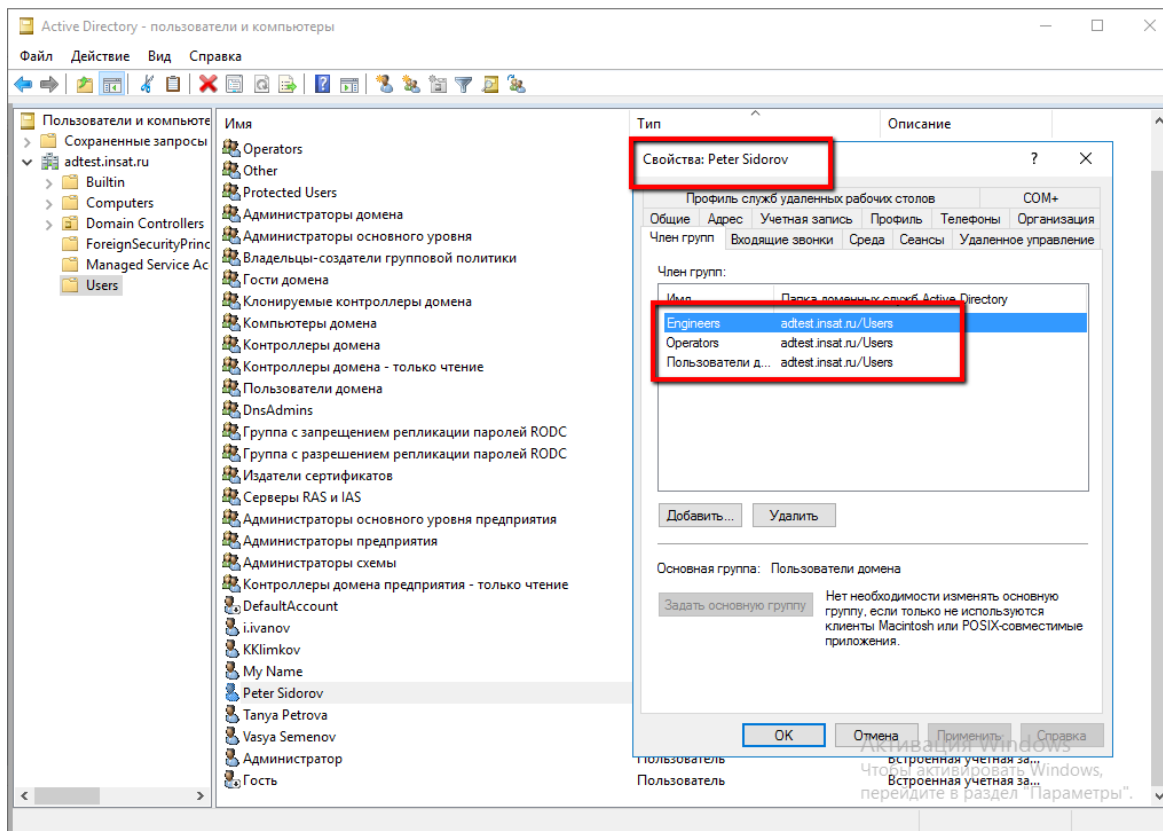
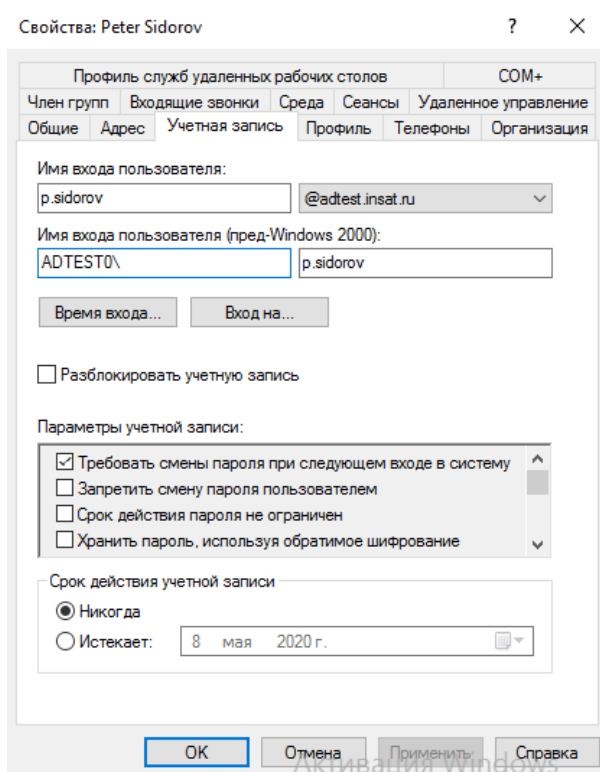
На приведенном ниже рисунке изображены созданные администратором три группы безопасности и три пользователя, назначенные в эти группы:



Каждый пользователь имеет свои собственные настройки.

Пользователь Peter Sidorov назначен в группы Engineers и Operators. Имя входа в операционную систему p.sidorov:






Пользователь Tanya Petrova назначен в группу Operators. Имя входа в систему t.petrova

Свойства: Tanya Petrova

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:  Другой...

Эл. почта:

Веб-страница:  Другой...

OK Отмена Применить Справка

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:  
 @adtest.insat.ru

Имя входа пользователя (пред-Windows 2000):

Время входа... Вход на...

Разблокировать учетную запись

Параметры учетной записи:

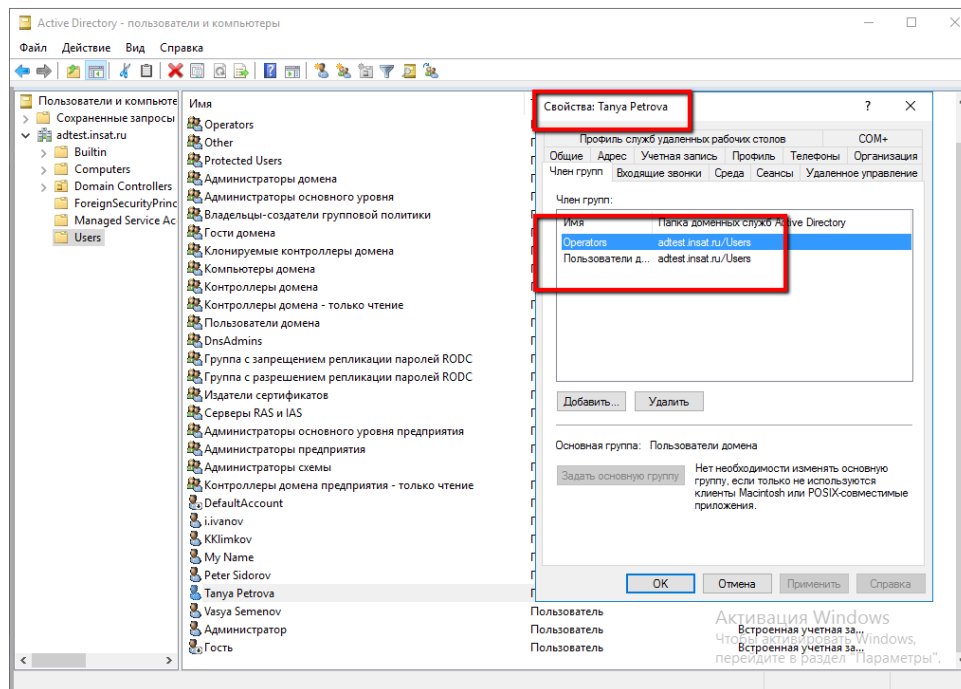
- Требовать смены пароля при следующем входе в систему
- Запретить смену пароля пользователем
- Срок действия пароля не ограничен
- Хранить пароль, используя обратимое шифрование

Срок действия учетной записи

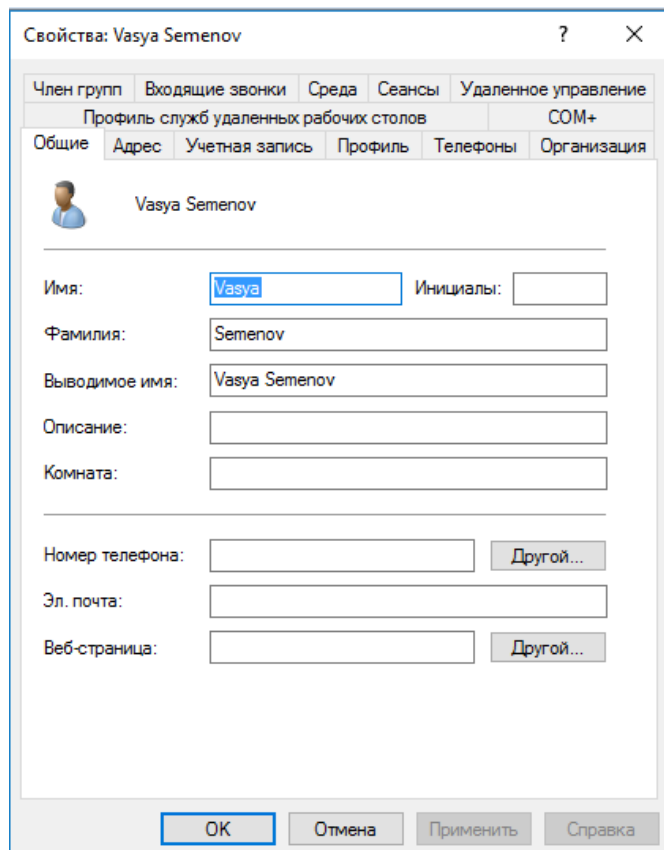
Никогда

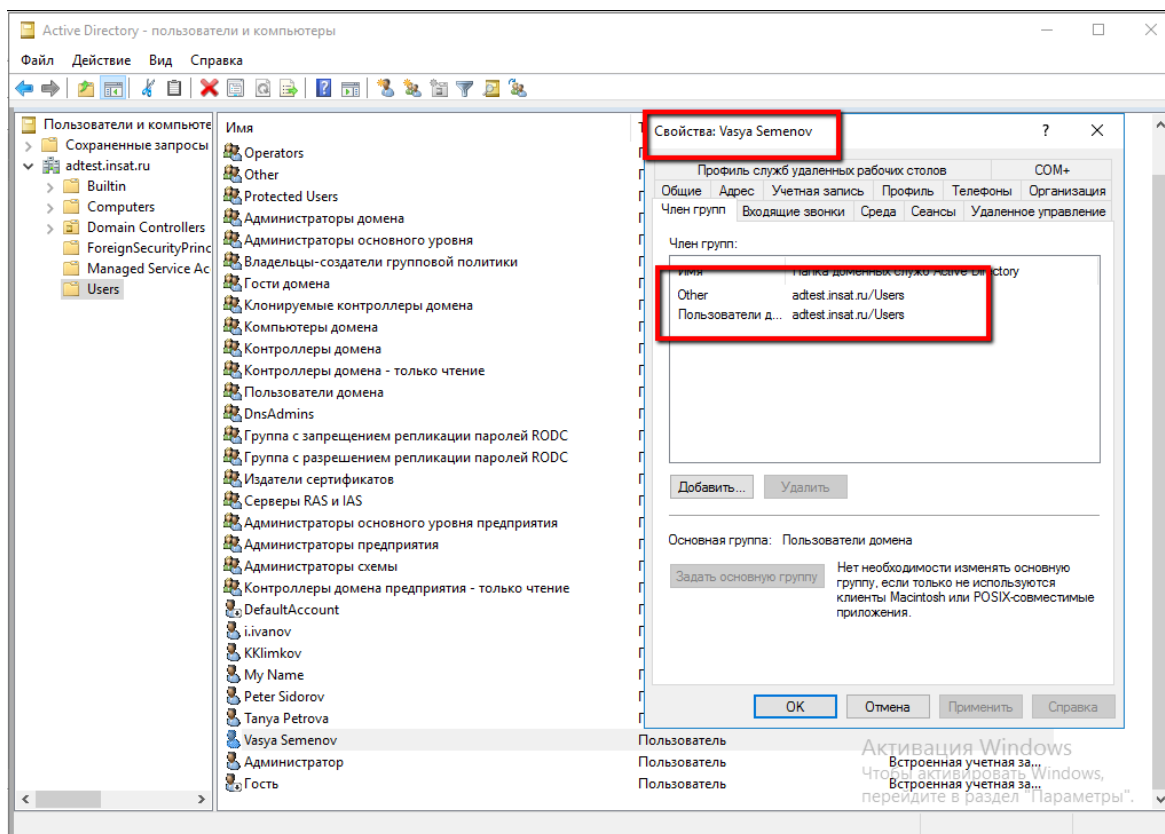
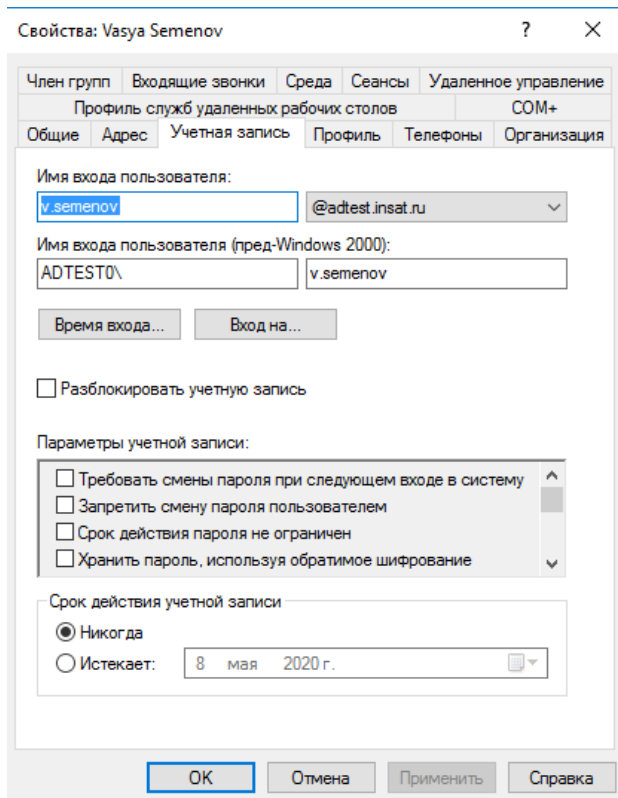
Истекает:

OK Отмена Применить Справка



Пользователь Vasya Semenov назначен в группу Other. Имя входа в систему v.semenov:



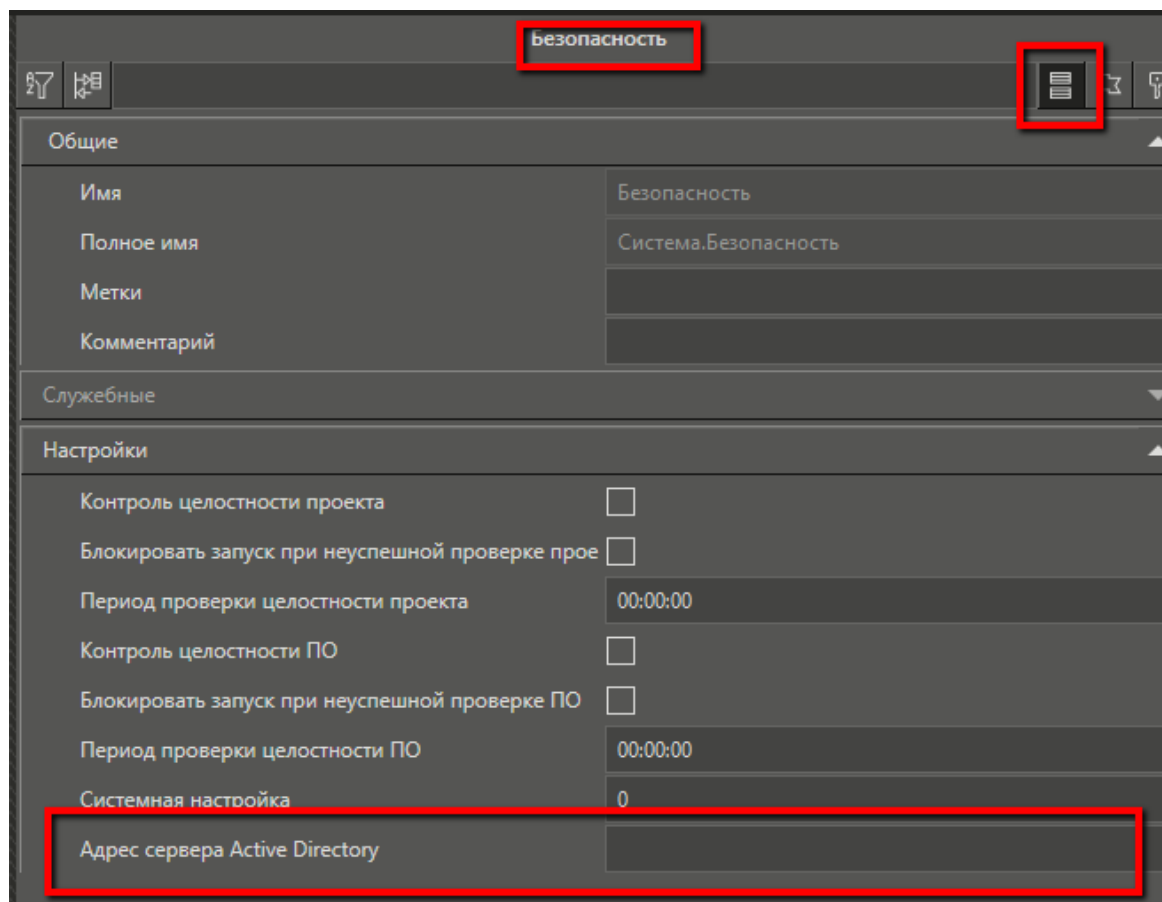




## Взаимодействие MasterSCADA 4D и ActiveDirectory

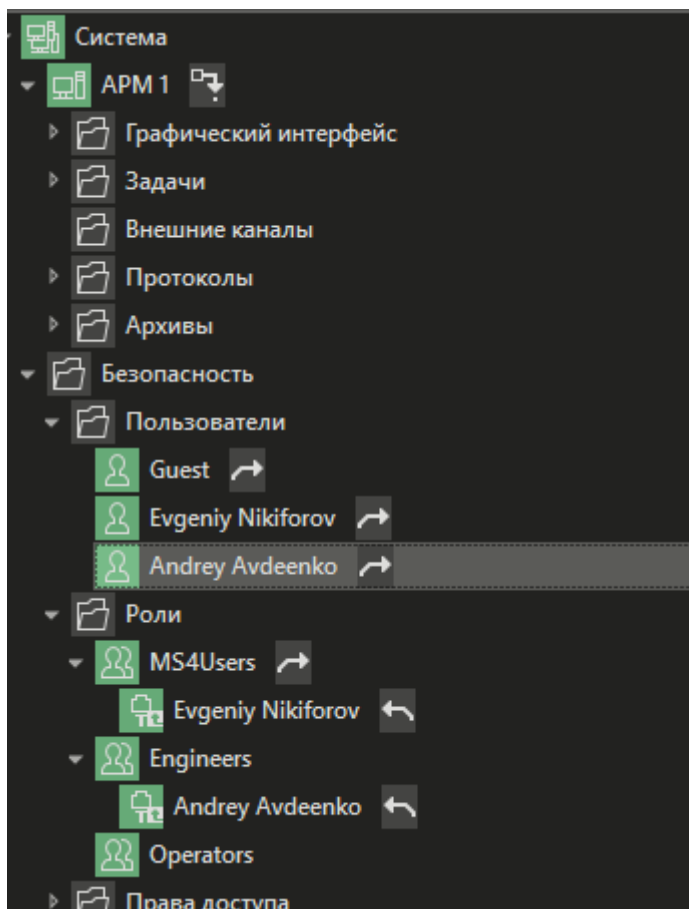
MasterSCADA 4D позволяет объединить встроенную систему Безопасности и группы Active Directory.

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:

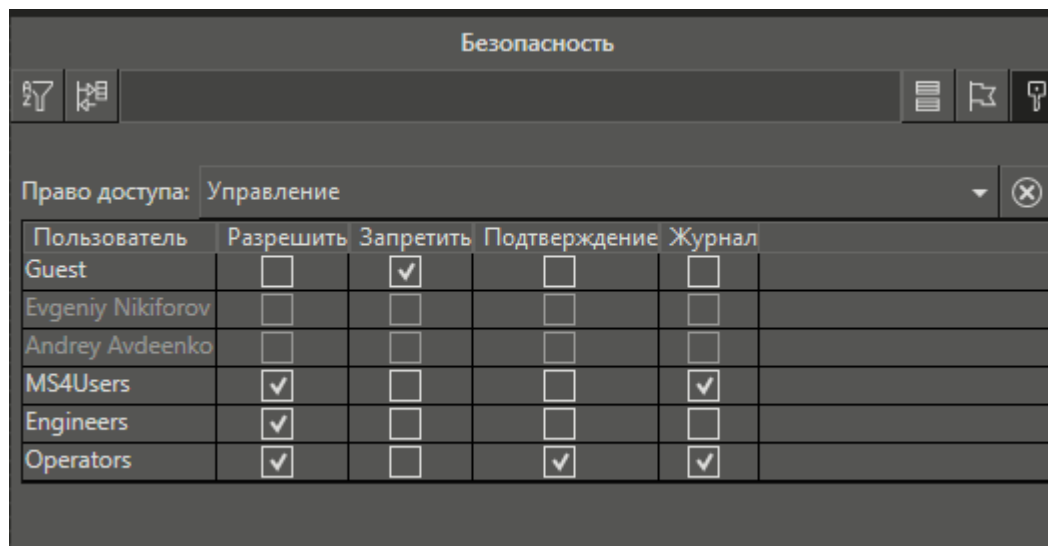


В созданные роли можно также добавить и пользователей, которые созданы в MasterSCADA 4D.

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"/>	

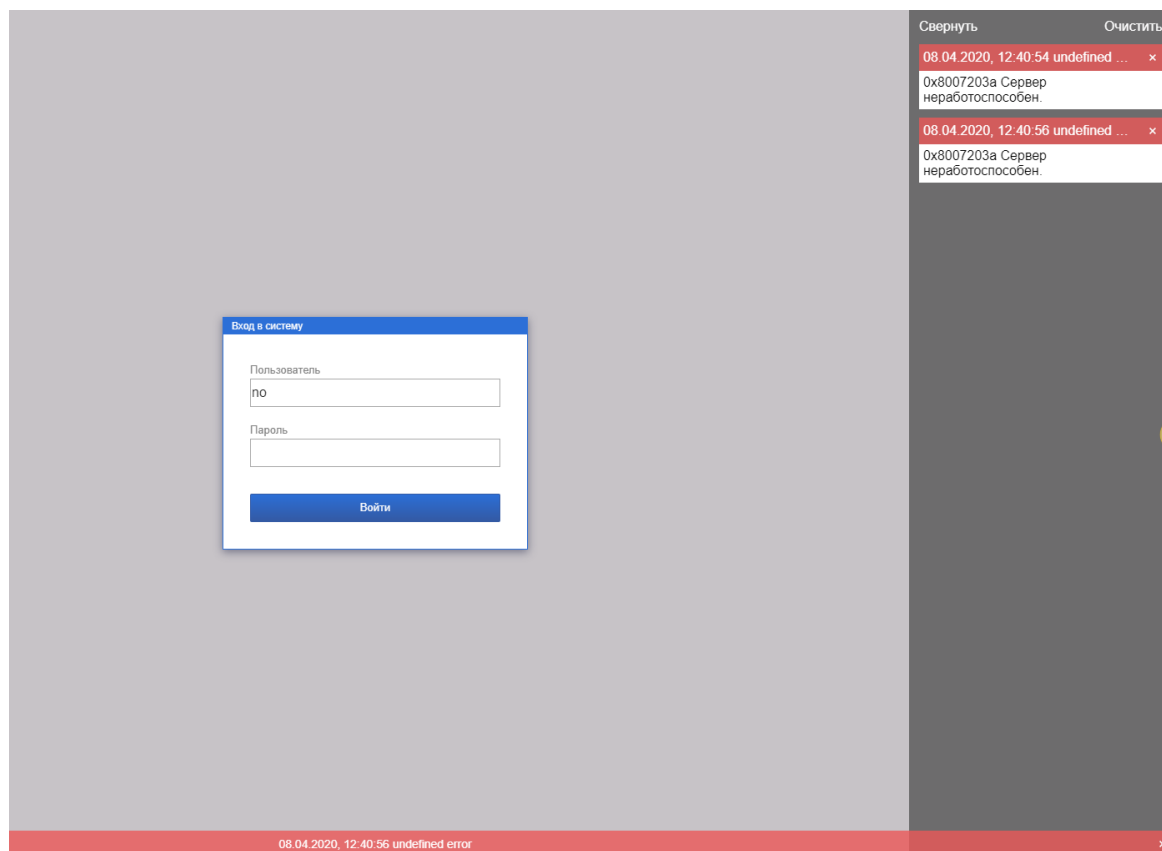
А также права доступа Управление:



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

### 6.3.3. 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)\InSAT\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

### 6.3.3.1. Стандартные протоколы

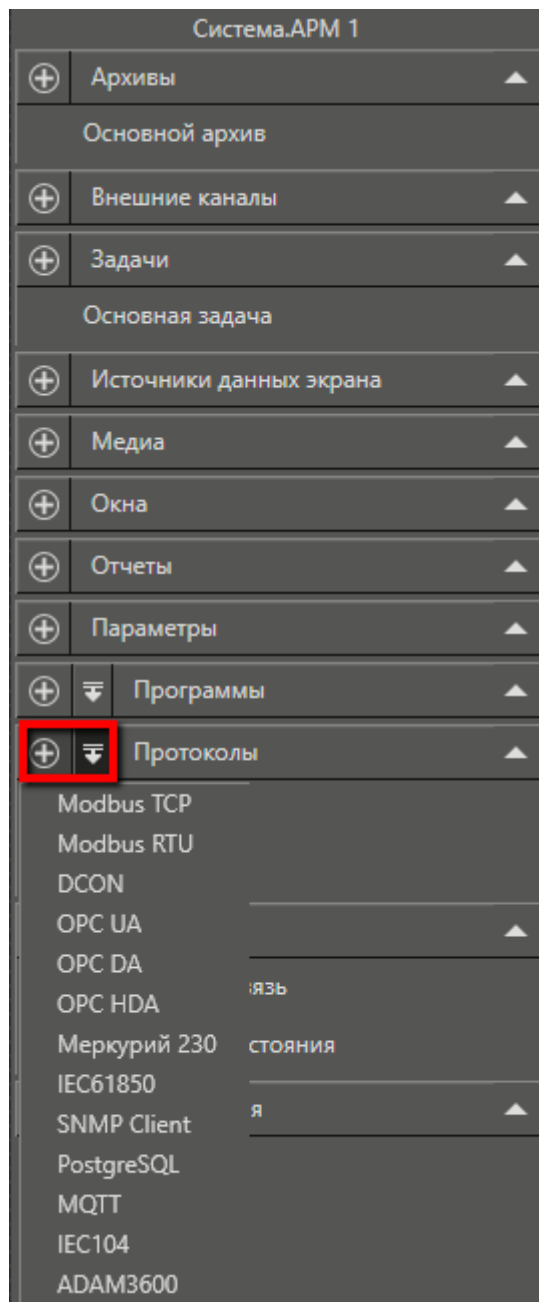
В базовый комплект MasterSCADA 4D входят следующие протоколы:

- 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.



### 6.3.3.1.1. Modbus

Modbus TCP

Свойства протокола Modbus TCP

Модуль Modbus TCP

Modbus RTU

Свойства протокола Modbus RTU

Модуль Modbus RTU

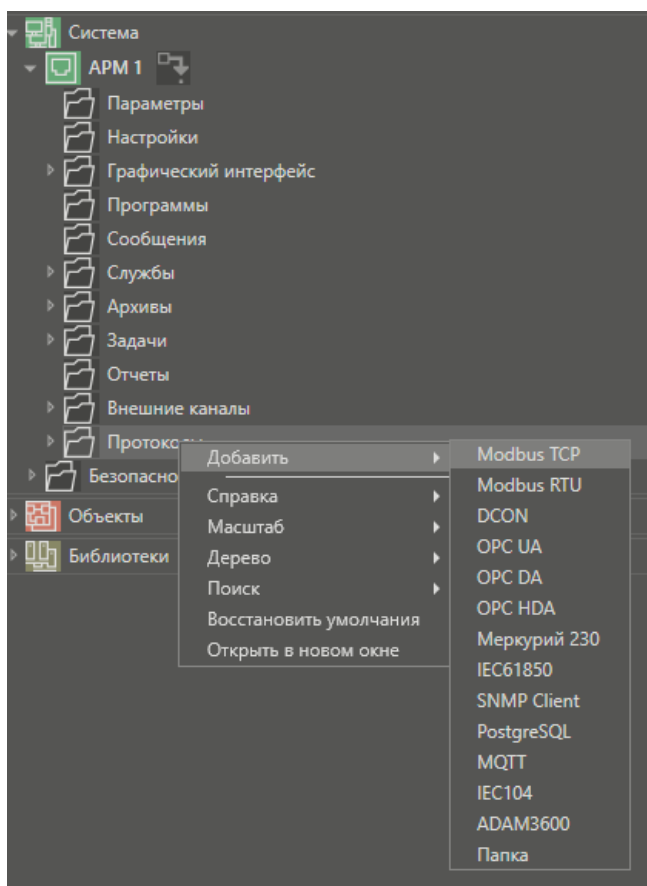
Свойства модулей Modbus TCP и Modbus RTU

Свойства каналов Modbus

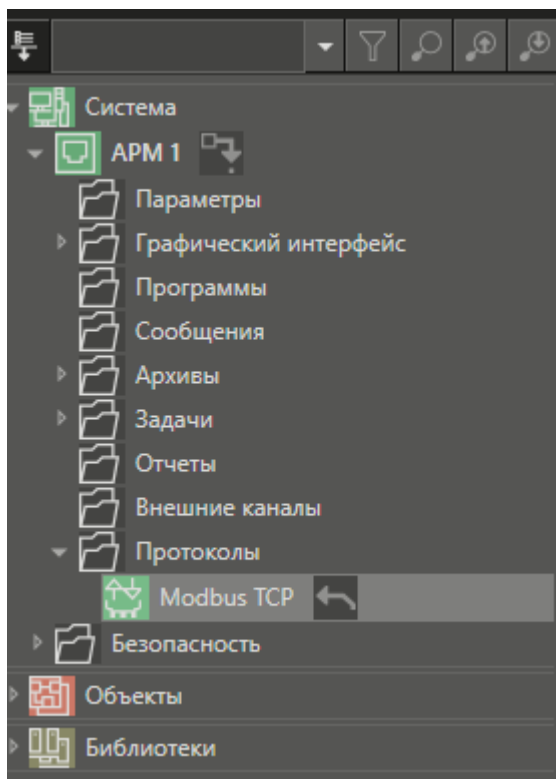
### 6.3.3.1.1.1. Modbus TCP

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.

Смотрите также:

Настройки модуля ввода-вывода ModbusTCP

Настройки каналов Modbus

Modbus Slave

## 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"/>

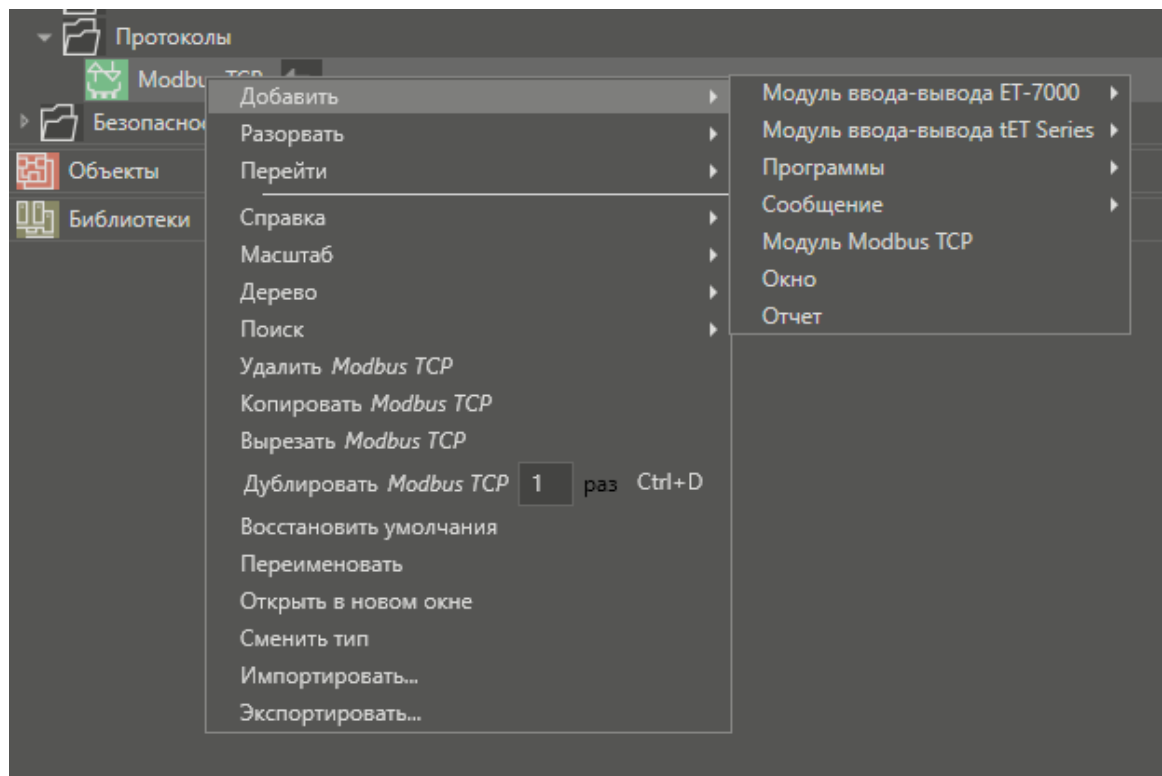
Описание:

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 зафиксируется ошибка данного запроса для конкретного модуля.

<p>Количество повторов при неудачном опросе</p>	<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>
<p>Modbus поверх TCP</p>	<p>Свойство предназначено для работы с конвертерами Ethernet-COM (например, Муха 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. 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

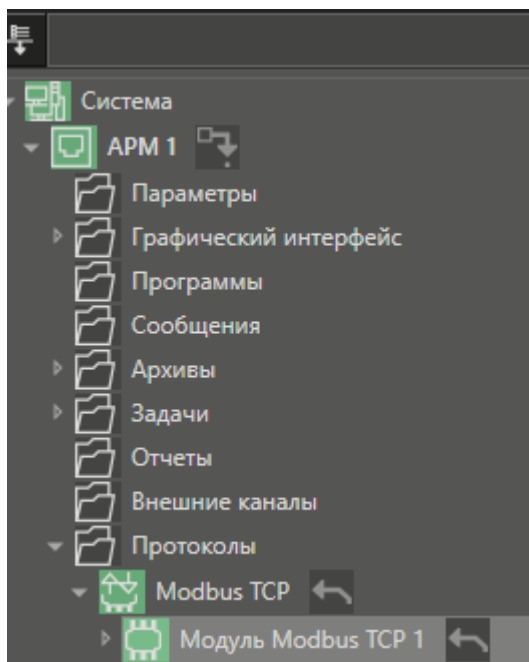
Модуль ввода-вывода добавляется в дерево системы в соответствующий протокол при помощи контекстного меню протокола:



В проект можно добавить как predetermined модули ввода-вывода (т.е. ранее созданные в библиотеке компанией "ИнСАТ" 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.

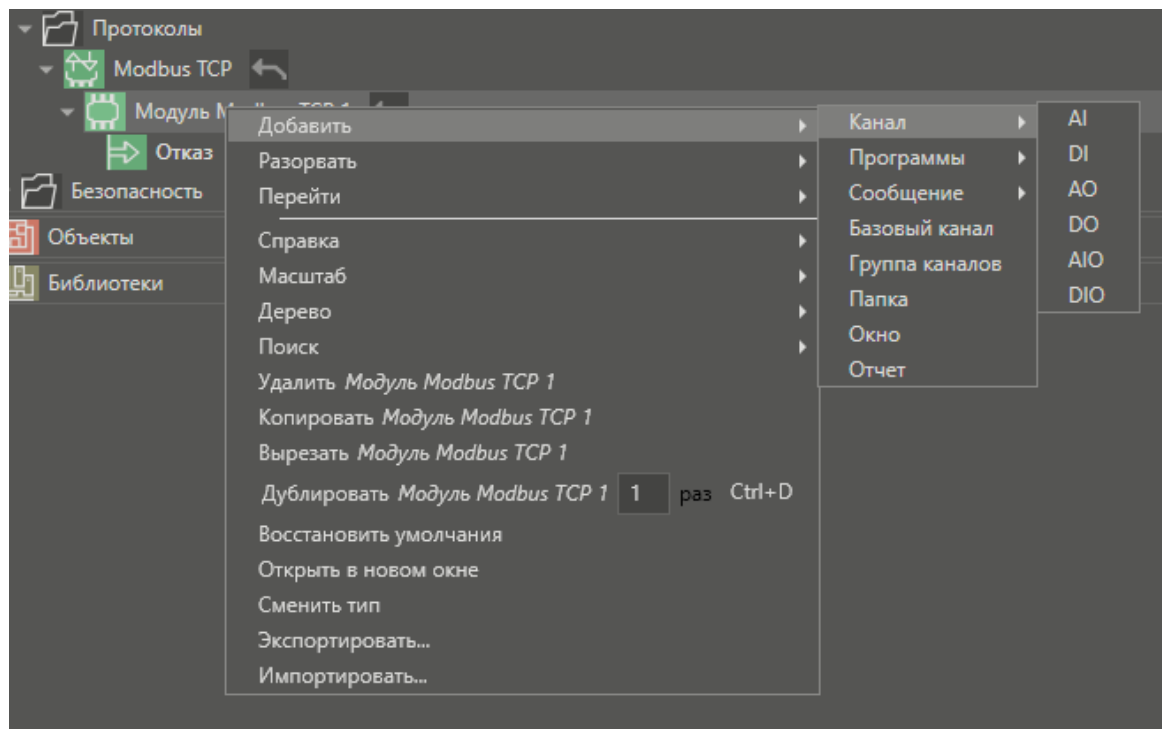
Если добавить элемент Модуль Modbus TCP, каналы необходимо будет добавить вручную.



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 объекта, тег, канал.

Смотрите также:

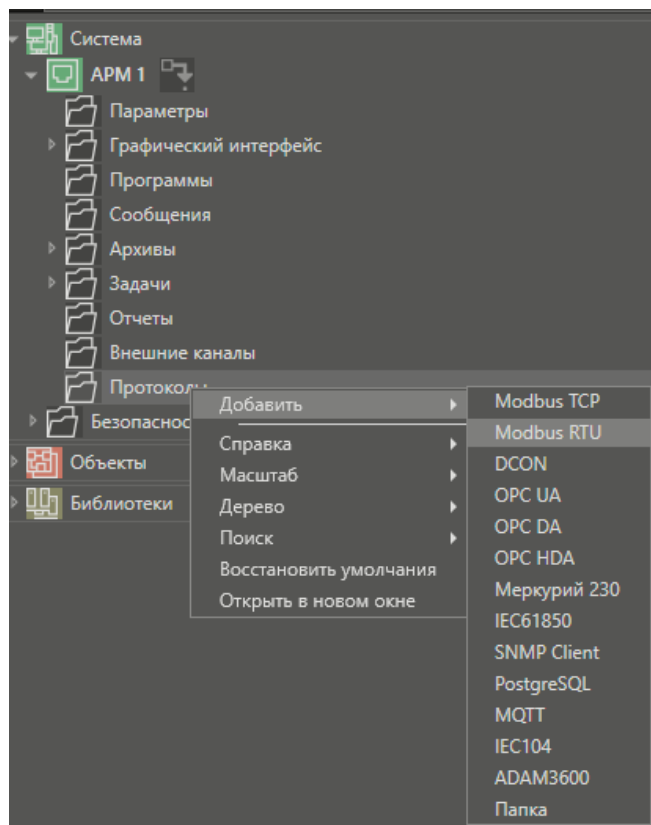
Панель свойств Модуля Modbus TCP

Рекомендации по созданию новых типов протоколов, модулей

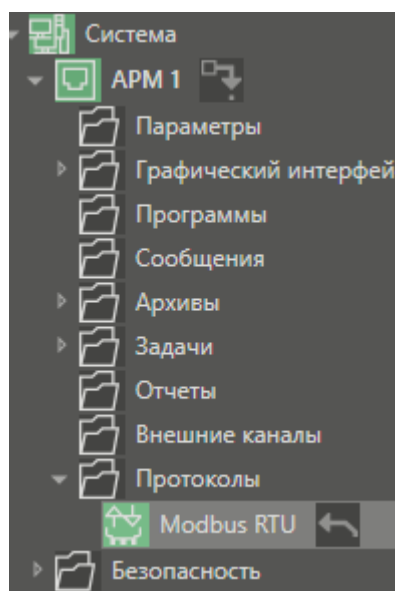
### 6.3.3.1.1.2. Modbus RTU

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.

Смотрите также:

Настройки модуля ввода-вывода Modbus RTU

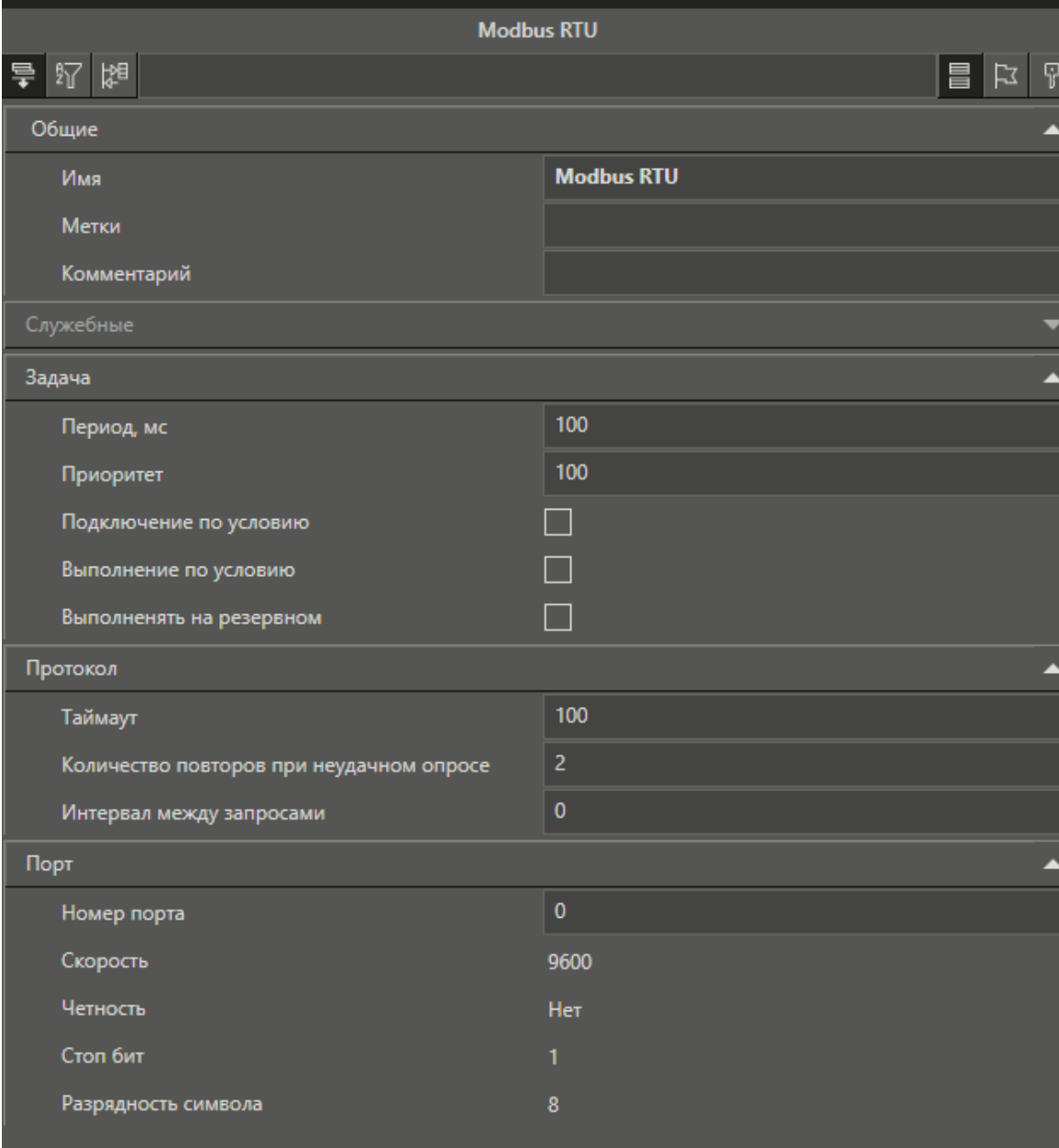
Настройки каналов Modbus

Modbus Slave

## СВОЙСТВА ПРОТОКОЛА MODBUS RTU

Настройка элемента производится в панели свойств.

Вид панели свойств:



Modbus RTU	
<b>Общие</b>	
Имя	Modbus RTU
Метки	
Комментарий	
<b>Служебные</b>	
<b>Задача</b>	
Период, мс	100
Приоритет	100
Подключение по условию	<input type="checkbox"/>
Выполнение по условию	<input type="checkbox"/>
Выполнять на резервном	<input type="checkbox"/>
<b>Протокол</b>	
Таймаут	100
Количество повторов при неудачном опросе	2
Интервал между запросами	0
<b>Порт</b>	
Номер порта	0
Скорость	9600
Четность	Нет
Стоп бит	1
Разрядность символа	8

Описание:

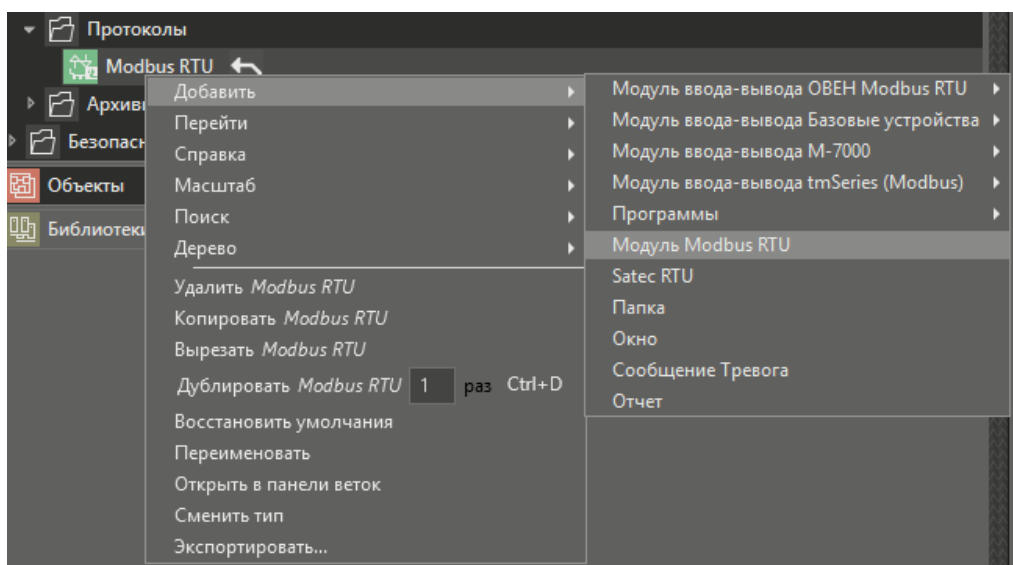
Название	Рекомендации
Категория Общие	Как правило, для данного элемента не настраивается. Подробное описание данной категории приведено в соответствующем разделе.
Категория Задача	Задается период, с которым будут опрашиваться модули ввода-вывода. Определяется способ работы протокола при резервировании узла. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи у протоколов.
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 течении заданного интервала времени, то попытка опроса будет считаться неудачной, и зафиксироваться 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 формируется 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	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.
Номер порта	Номер COM-порта, к которому подключаются модули ввода-вывода.
Скорость	Скорость, на которой будет производиться опрос модулей ввода-вывода (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 – 1;
Разрядность символа	Задается число информационных битов (7 или 8), значение по умолчанию – 8;

## МОДУЛЬ MODBUS RTU

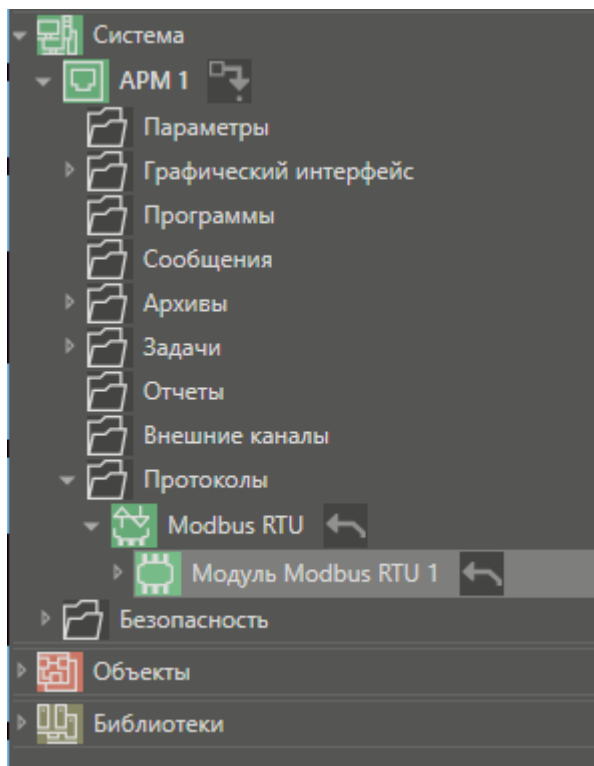
Модуль ввода-вывода добавляется в дерево системы в соответствующий протокол, при помощи контекстного меню протокола:



В проект можно добавить как predefined модули ввода-вывода (т.е. ранее созданные в библиотеке компанией "ИнСАТ" или разработчиком проекта), которые работают по данному протоколу, так и произвольный модуль.

При добавлении predefined модуля в проект, автоматически добавятся и каналы. Количество каналов будет соответствовать возможностям устройства, которое описывает predefined модуль.

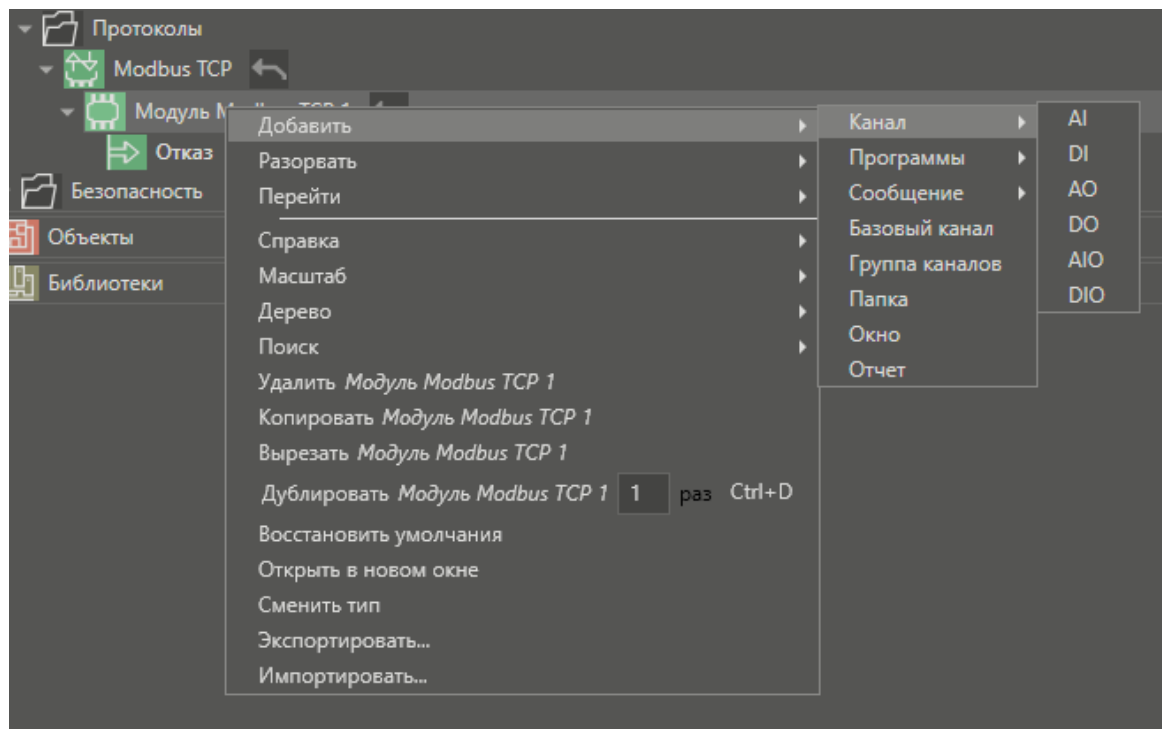
Если добавить элемент Модуль Modbus RTU, то каналы необходимо будет добавить вручную.



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 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 объекта, тег, канал.

Смотрите также:

Панель свойств Модуля Modbus RTU

Рекомендации по созданию новых типов протоколов, модулей

### 6.3.3.1.1.3. 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

Масштабирование

Настройки

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
Адрес	1

Настройки каналов

Адрес ячейки	0
Тип значения	Целый
Тип ячейки	Регистр хранения

## Описание:

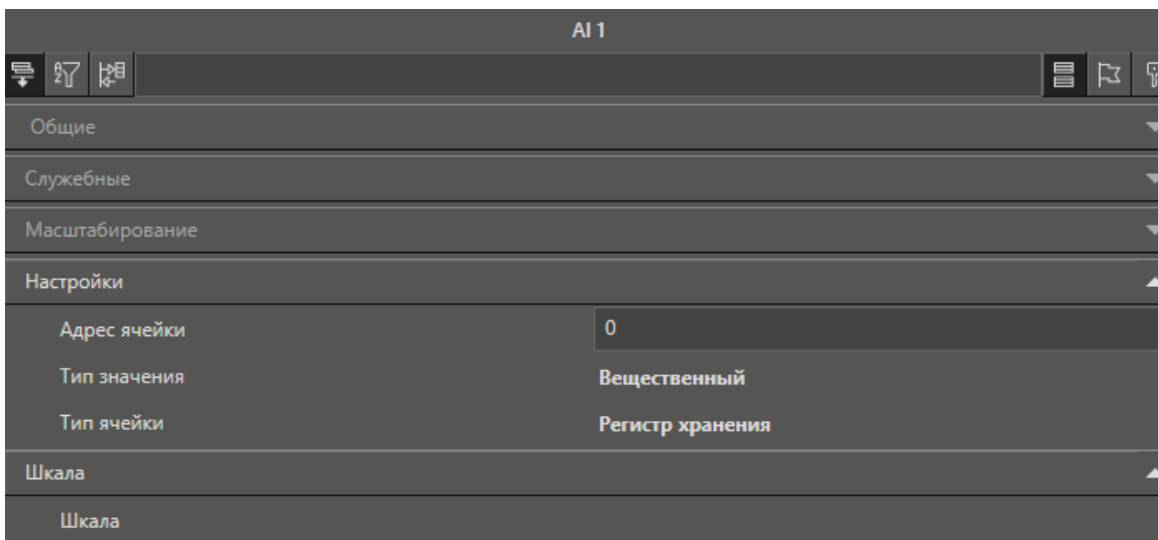
Название	Описание
Категория Общие	Как правило, для данного элемента не настраивается. Подробное описание данной категории приведено в соответствующем разделе.
Категория Задача	Задается способ опроса устройства. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи.
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.
IP-адрес	Задается IP-адрес модуля (значение IP-адреса определяется документацией на подключаемое устройство).

ТСР-порт	Задается ТСР-порт модуля ввода-вывода (номер порта определяется документацией на подключаемое устройство)
Адрес устройства (Адрес)	Задается адрес устройства (значение адреса определяется документацией на подключаемое устройство).
Таймаут	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.
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.
-------------------	--

### 6.3.3.1.1.4. 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.



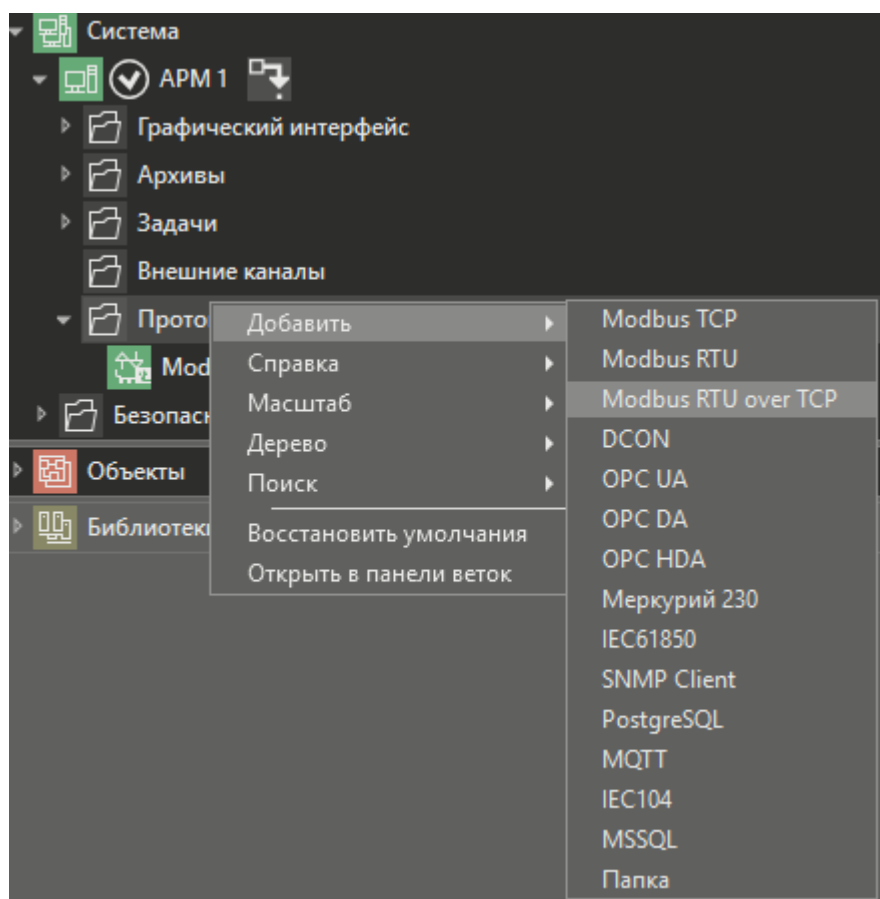
Для чтения всегда используются групповые запросы. Максимальное количество записей в запросе определяется настройкой модуля Максимальный размер массива

Если настройки сделаны верно, то при подключении к устройству и загрузке в него конфигурации должна появиться связь с модулем.

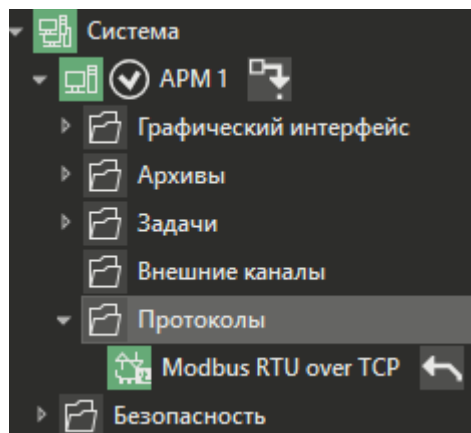
### 6.3.3.1.1.5. Modbus RTU over TCP

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.

Смотрите также:

Настройки модуля ввода-вывода Modbus RTU

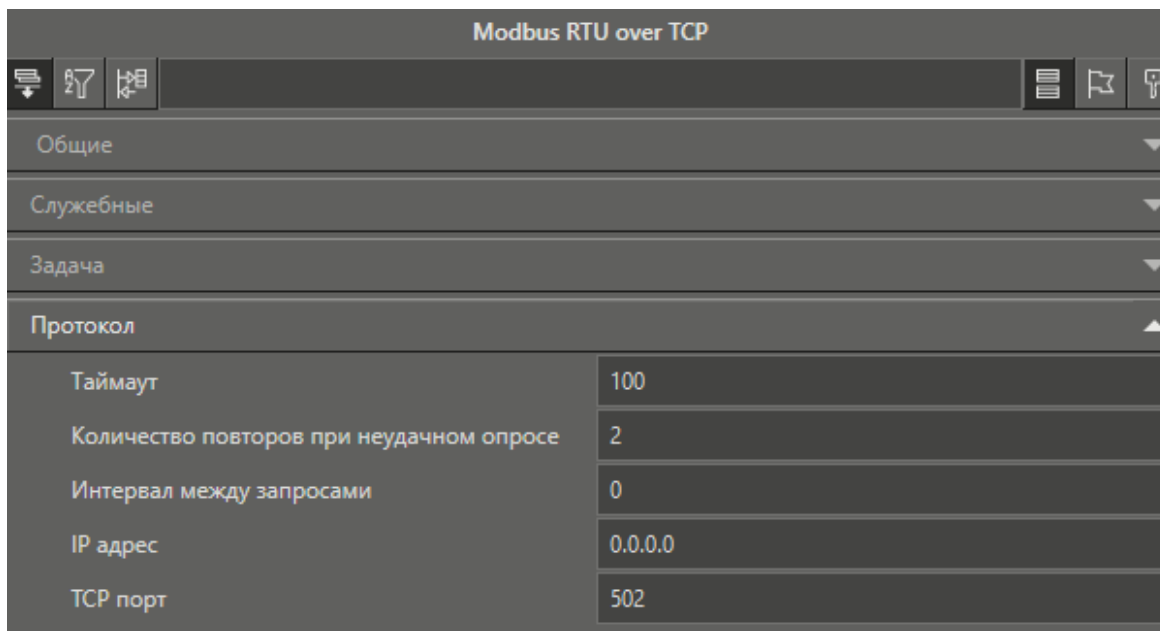
Настройки каналов Modbus

Modbus Slave

## СВОЙСТВА ПРОТОКОЛА MODBUS RTU OVER TCP

Настройка элемента производится в панели свойств.

Вид панели свойств:



Описание основных свойств:

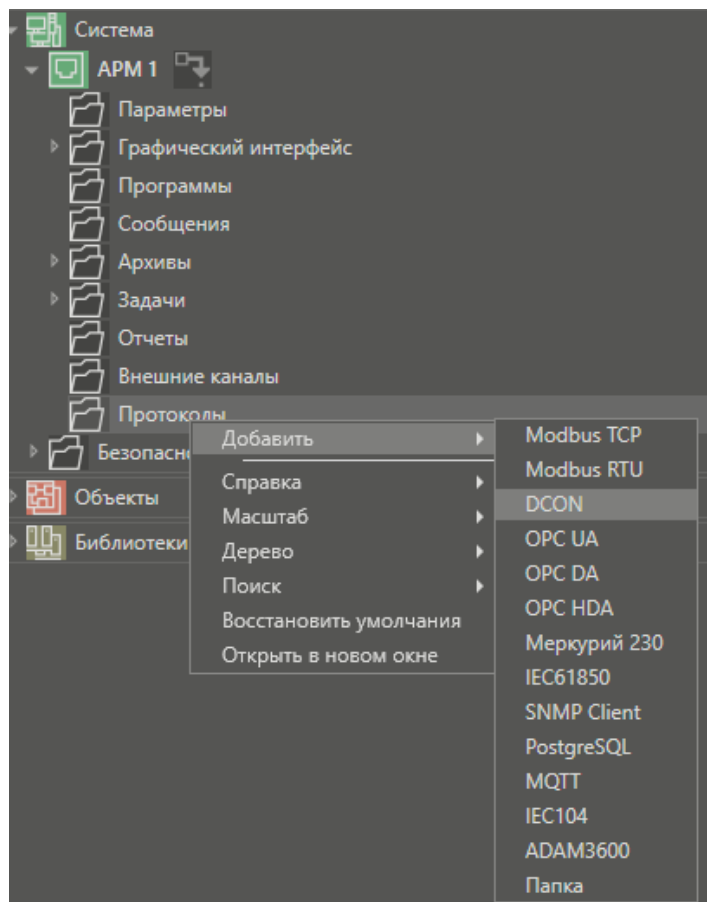
Название	Рекомендации
Категория Общие	Как правило, для данного элемента не настраивается. Подробное описание данной категории приведено в соответствующем разделе
Категория Задача	Задается период, с которым будут опрашиваться модули ввода-вывода. Определяется способ работы протокола при резервировании 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 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</p>

	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.
Интервал между запросами	Определяет интервал между запросами. Если этот интервал не задан, то MasterSCADA 4D рассчитывает 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)

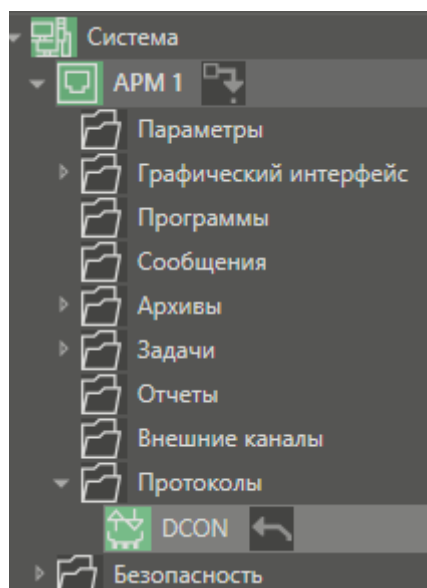
### 6.3.3.1.2. DCON

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.

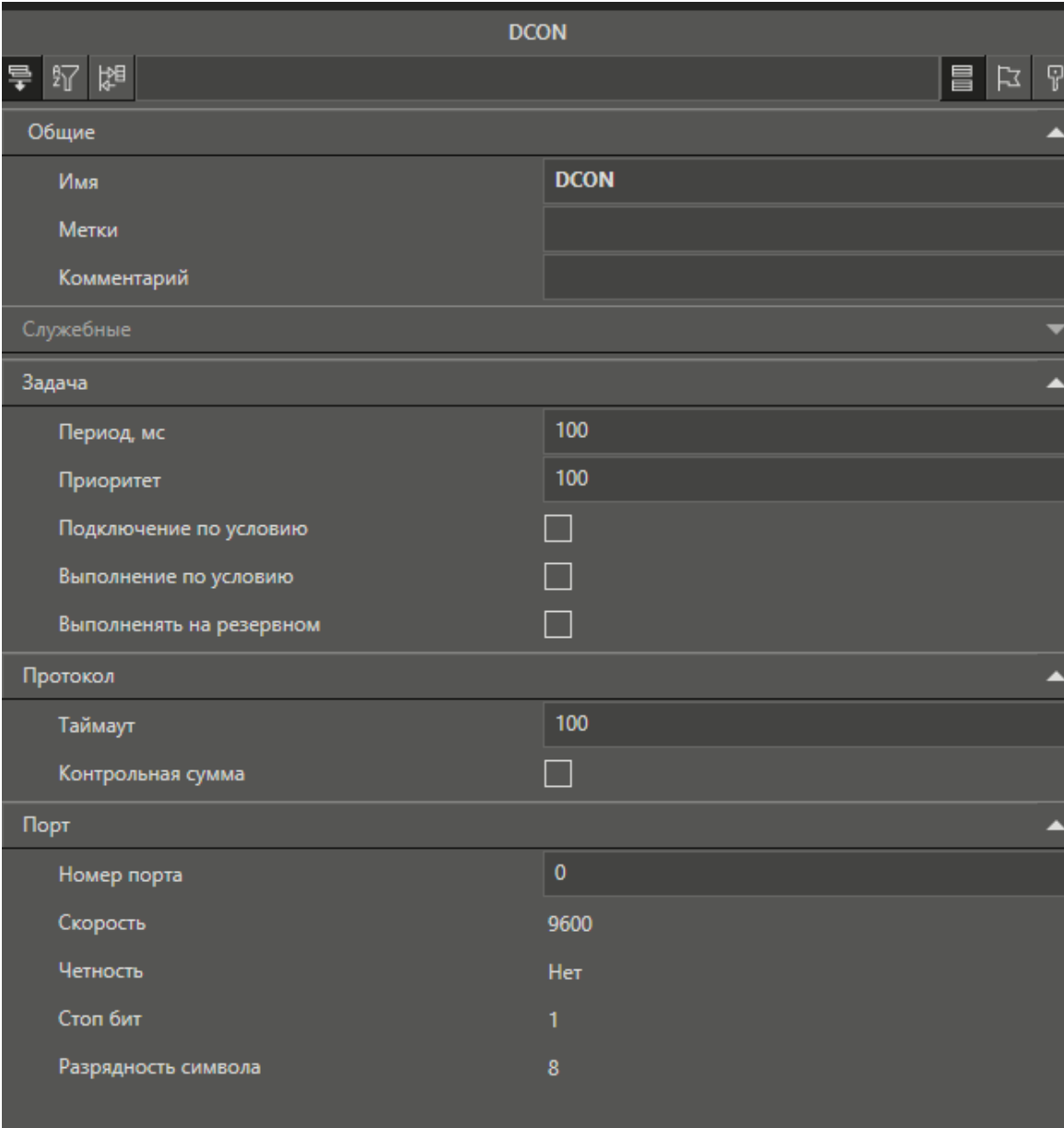
Смотрите также:

Настройки модуля ввода-вывода DCON

### 6.3.3.1.2.1. Свойства протокола DCON

Настройка элемента производится в панели свойств.

Вид панели свойств:



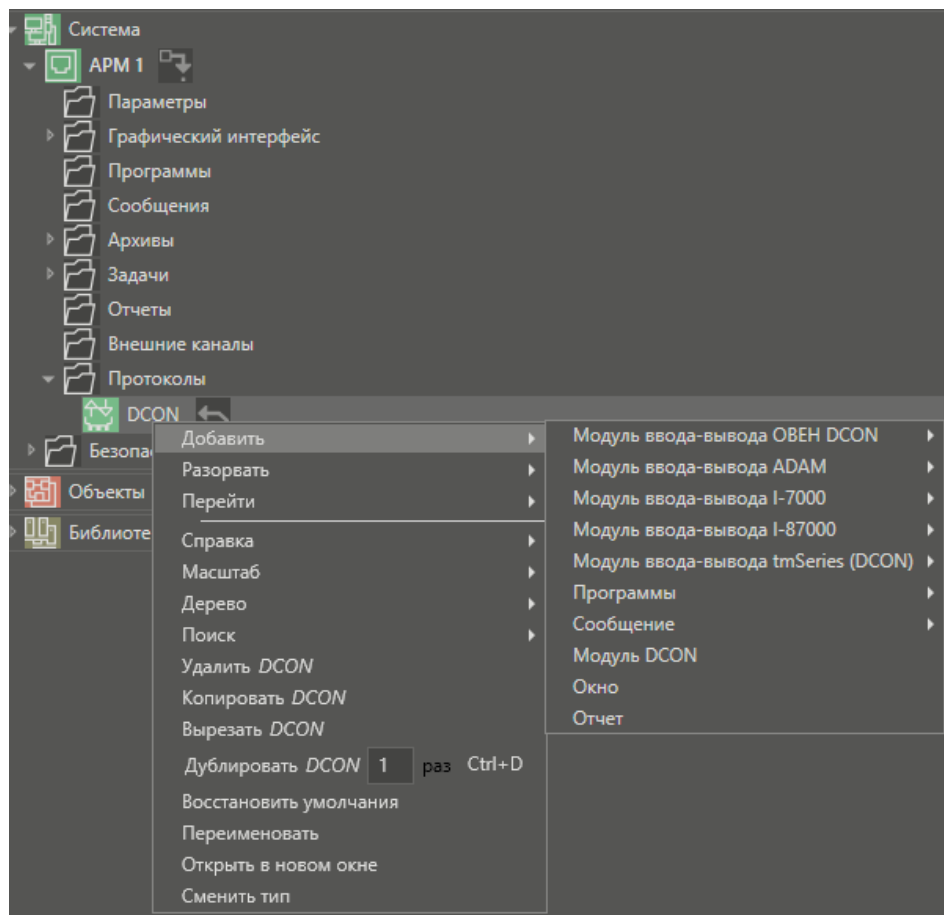
DCON	
<b>Общие</b>	
Имя	DCON
Метки	
Комментарий	
<b>Служебные</b>	
<b>Задача</b>	
Период, мс	100
Приоритет	100
Подключение по условию	<input type="checkbox"/>
Выполнение по условию	<input type="checkbox"/>
Выполнять на резервном	<input type="checkbox"/>
<b>Протокол</b>	
Таймаут	100
Контрольная сумма	<input type="checkbox"/>
<b>Порт</b>	
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
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;

### 6.3.3.1.2.2. Модуль DCON

Модуль ввода-вывода добавляется в дерево системы в соответствующий протокол, при помощи контекстного меню протокола:

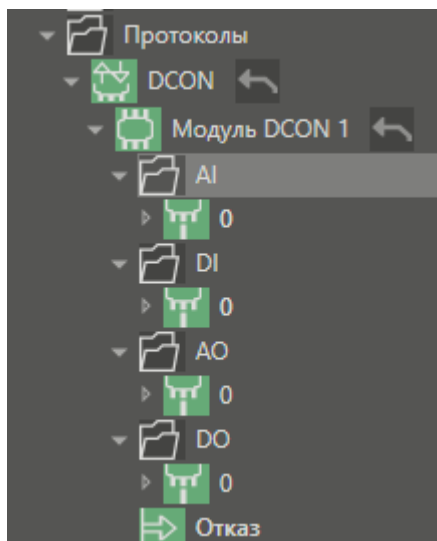


В проект можно добавить как predetermined модули ввода-вывода (т.е. ранее созданные в библиотеке компанией "ИнСАТ" или разработчиком проекта), работающие по данному протоколу, так и произвольный модуль.

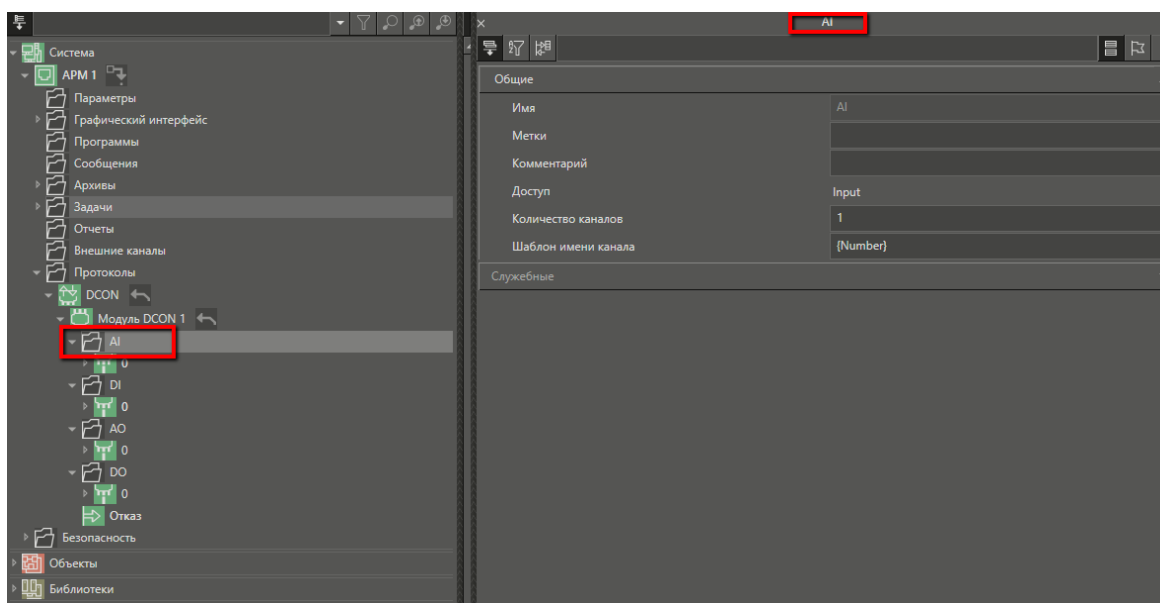
При добавлении predetermined модуля в проект, автоматически добавятся и каналы. Количество каналов будет соответствовать возможностям устройства, которое описывает predetermined модуль.

Если добавить элемент Модуль DCON, то по умолчанию добавится модуль, у которого будет создано 4 группы для добавления каналов predetermined 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 объект, тег, канал.

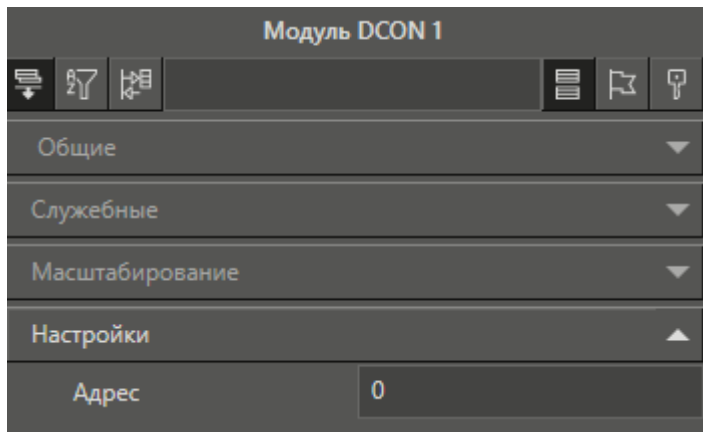
Смотрите также:

Панель свойств Модуля DCON

Рекомендации по созданию новых типов протоколов, модулей

## СВОЙСТВА МОДУЛЯ DCON

Вид панели свойств:



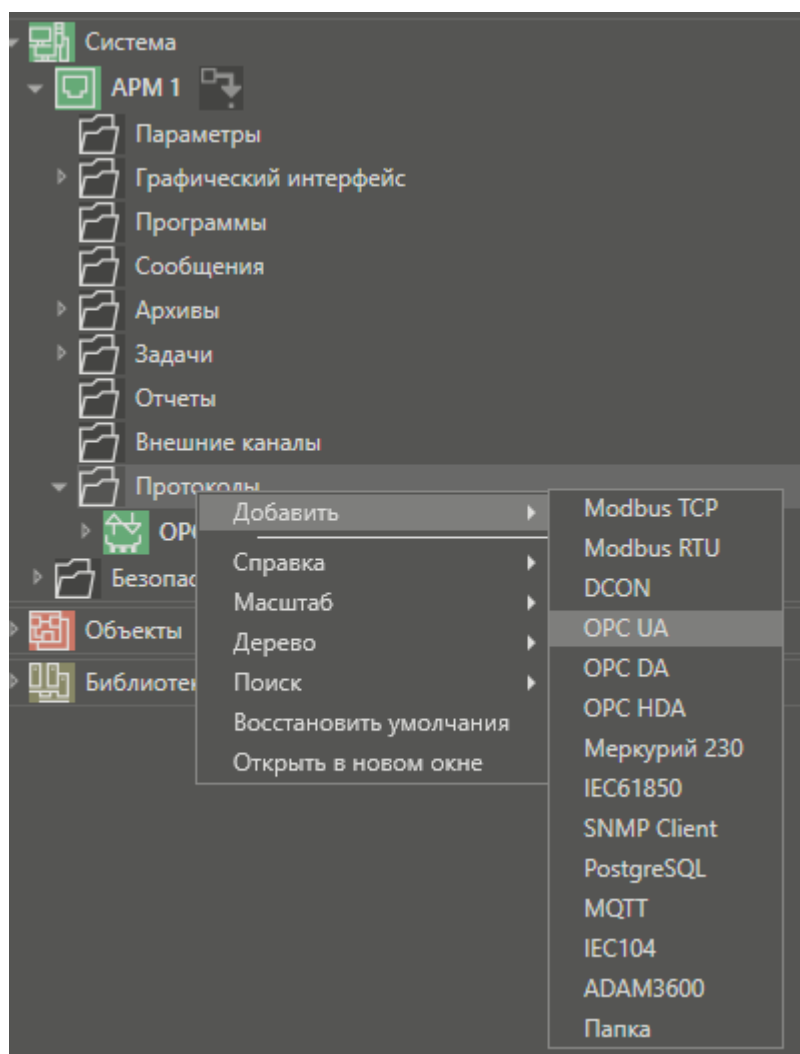
Описание:

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 определяются документацией на подключаемое устройство).

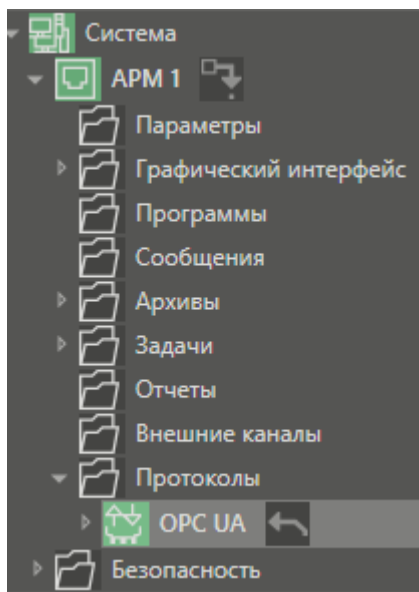
### 6.3.3.1.3. OPC UA

MasterSCADA 4D поддерживает платформонезависимый стандарт OPC UA.

Для того чтобы MasterSCADA 4D выступала в роли клиента, необходимо в группу узла Протоколы добавить соответствующий протокол:

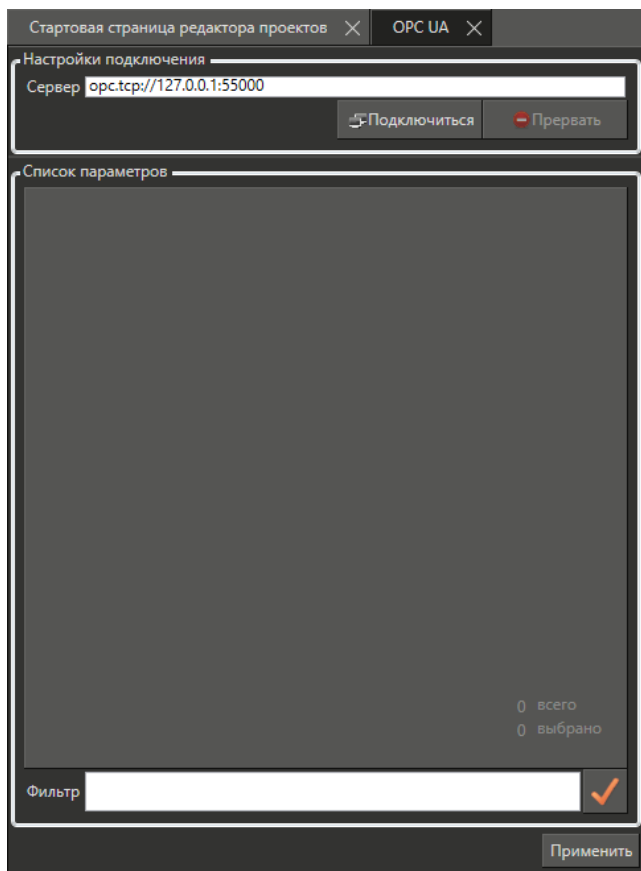


Получим результат в упрощенном дереве:



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 определенных типов

После подключения к серверу необходимо настроить панель свойств OPC UA сервера.

### **6.3.3.1.3.1. Свойства OPC UA**

Вид панели свойств:

OPC UA

Общие

Имя	OPC UA
Метки	
Комментарий	
Программное имя	OPC UA
Полное имя	Система.APM 1.Протоколы.OPC UA

Служебные

Отношения

Задача

Настройки каналов

Протокол

URI	opc.tcp://127.0.0.1:55000
Таймаут подписки	10s
Период записи	100ms
Имя пользователя	
Пароль	
Получать признак качества и метку вре	<input checked="" type="checkbox"/>
Политика безопасности	Нет
Режим безопасности сообщений	Нет
Чтение архивов	<input type="checkbox"/>
Режим чтения архивов	Чтение
Глубина считываемых данных при стар	1d
Количество возвращаемых архивных заг	1000
Период получения архивных данных	1s

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.
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.
Имя пользователя	Задается имя пользователя в случае неанонимного подключения. Если данное поле пустое, то MasterSCADA 4D считает, что соединение анонимное.
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	В случае если используется шифрование (Свойство Режим безопасности сообщений=SingAndEncript), в данном поле выбирается тип шифрования.

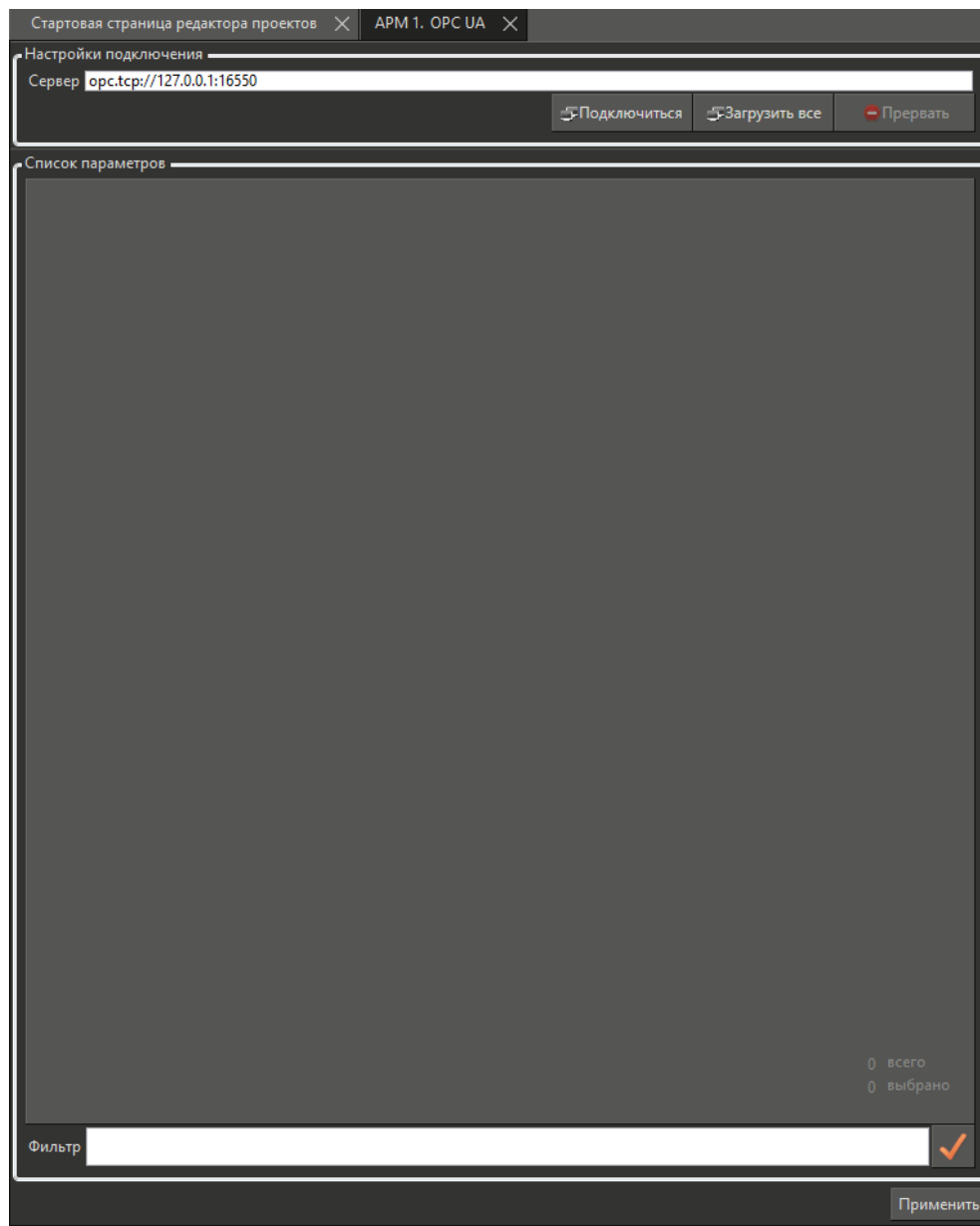
Режим безопасности сообщений	<p>Выбирается режим работы безопасности.</p> <p>Нет - данные передаются в открытом виде</p> <p>Подпись - данные подписываются</p> <p>Подпись и шифрование - данные подписываются и шифруются.</p> <p>Тип шифрования определяется настройкой Политика безопасности</p>
Чтение архивов	<p>Если флаг установлен, то MasterSCADA 4D будет получать архивные данные от сервера.</p>
Режим чтения архивов	<p>Настройка определяет метод получения архива переменных:</p> <ul style="list-style-type: none"> <li>• 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 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</li> <li>• Подписка - при данном методе данные получают 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</li> <li>• Чтение и подписка - при этом режиме архивные данные, ранее накопленные в приборе, будут получены методом Чтение, а дальнейшая работа будет производиться методом Подписка</li> </ul>
Период получения архивных данных	<p>Период, с которым OPC-клиент обращается к OPC-серверу для получения архивных данных в режиме работы Чтение.</p>



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.
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

### 6.3.3.1.3.2. Диалог подключения к серверу OPC UA

Данный диалог открывается во вкладке документов и редакторов по команде Открыть контекстного меню протокола OPC UA.

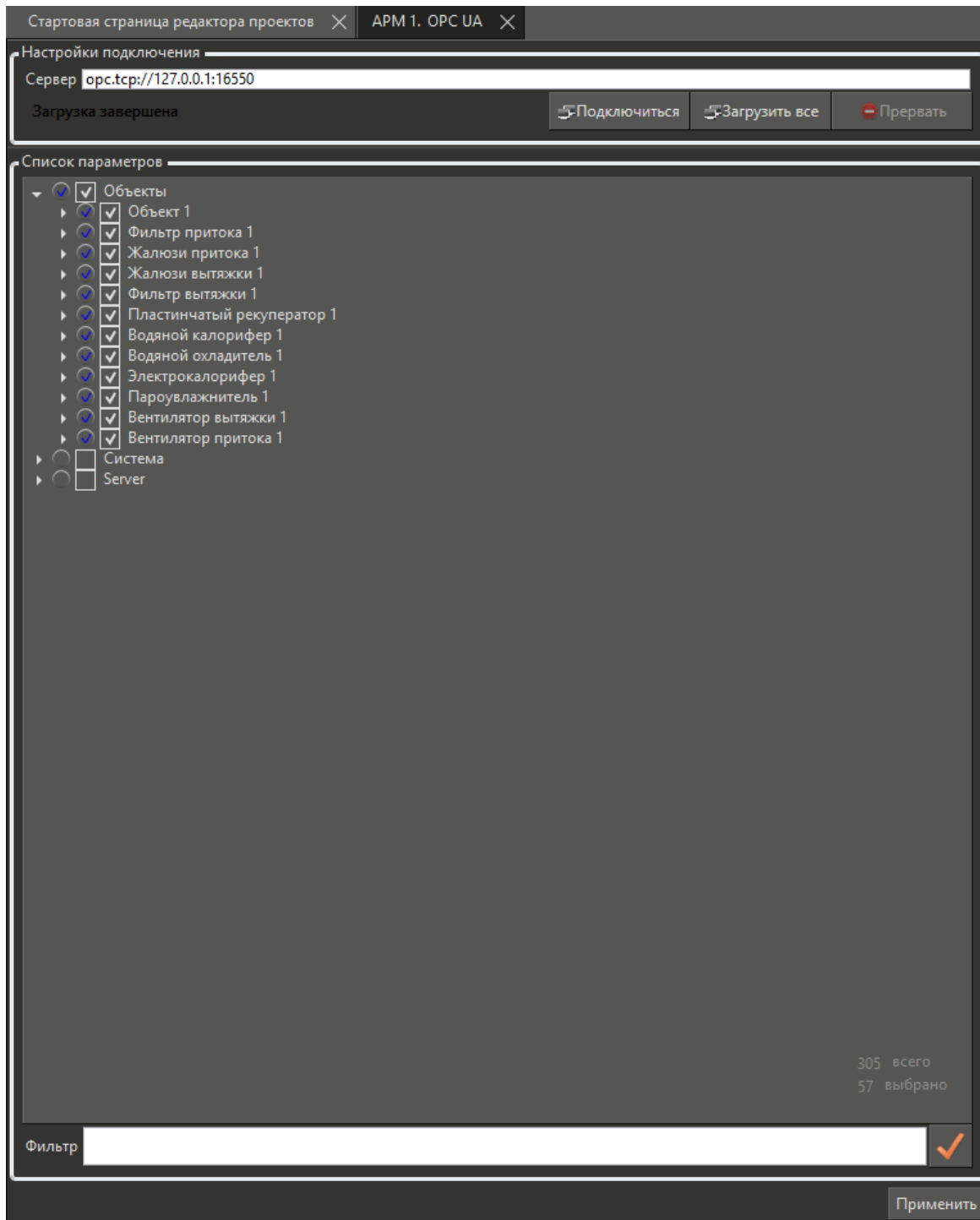


Группа Настройки подключения:


Название	Описание
Сервер	Задается IP-адрес и TCP/IP порт сервера, в виде: <code>opc.tcp://[IP-адрес][номер порта]</code> . Пример записи показан в значении по умолчанию: <code>opc.tcp://127.0.0.1:55000</code> .
Подключит ься	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



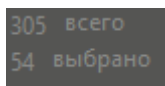

	<p>этом идет проверка, существуют ли уже такие группы, каналы в проекте. Если существуют, то у них устанавливается флаг, сигнализирующий о том, что они добавлены в проект. При установке круглой галочки сначала загружаются все дочерние группы этого элемента, а затем все загруженные элементы выбираются для вставки.</p>
Загрузить все	<p>Выполняет подключения к ранее запущенному в режим исполнения серверу. В группе диалога Список параметров видны все возможные для добавления группы и каналы.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"><p>Важно! Если сервер имеет большую конфигурацию, то этот процесс может занять длительное время.</p></div>
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.

	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

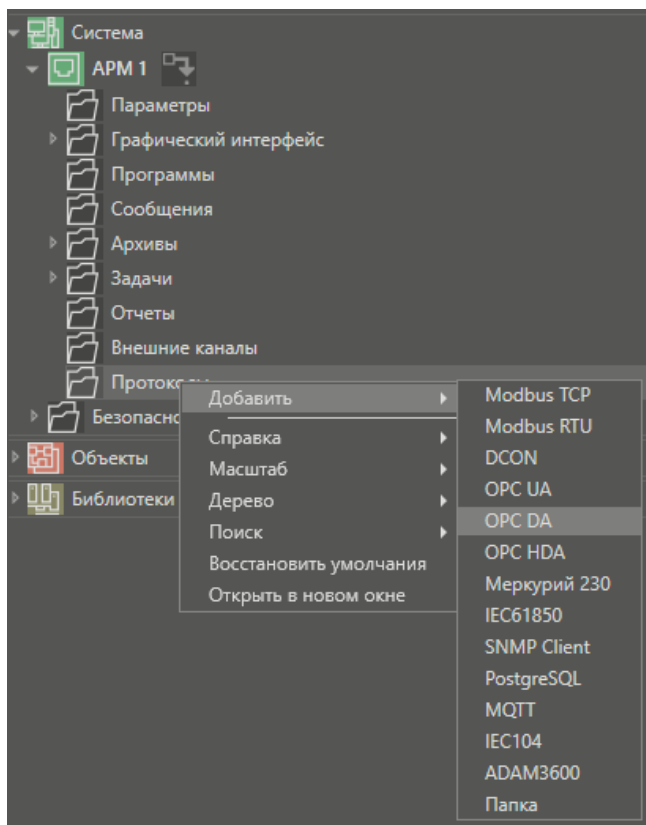
Важно! Изменения вступят в силу только после нажатия на кнопку Применить

#### 6.3.3.1.4. OPC DA

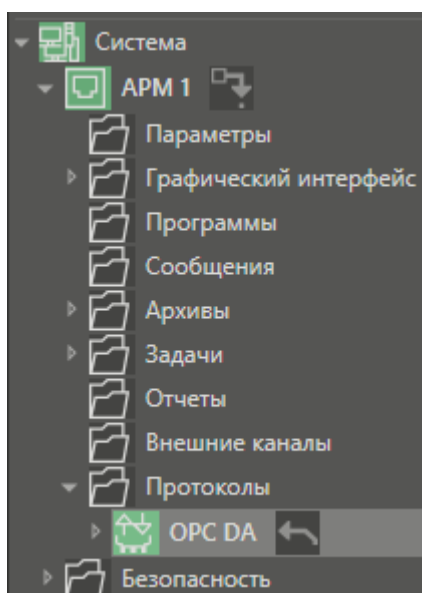
MasterSCADA 4D поддерживает стандарт OPC DA. Данный стандарт описывает получение текущих данных.

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:

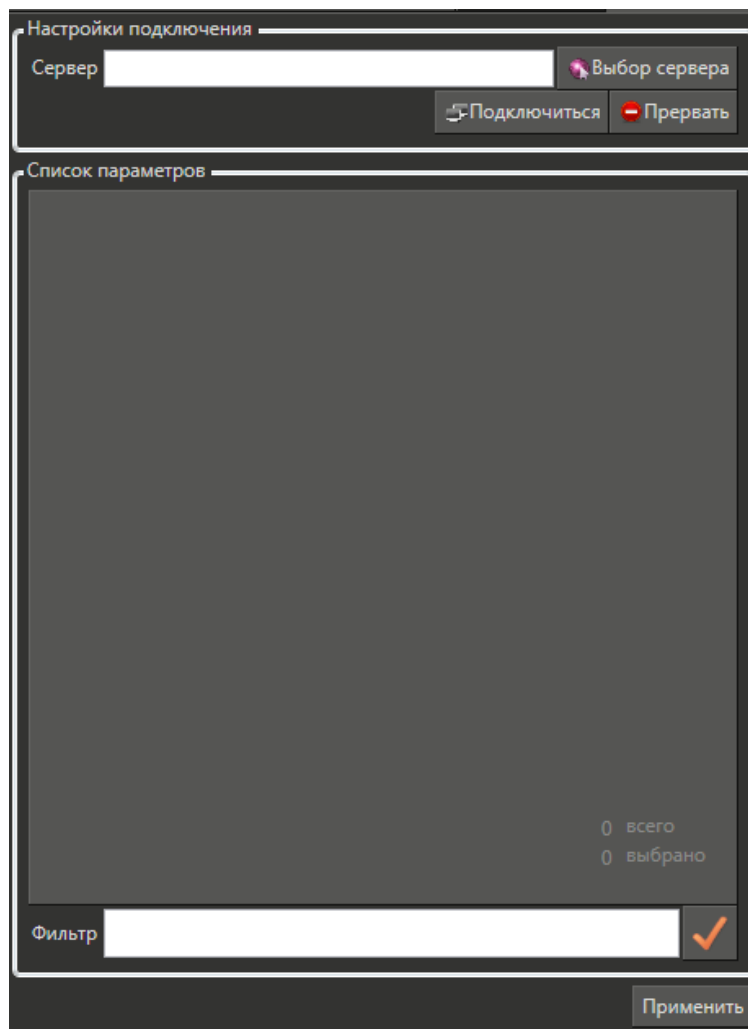


Получим результат в упрощенном дереве:



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 предопределенных 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.

### 6.3.3.1.4.1. 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.



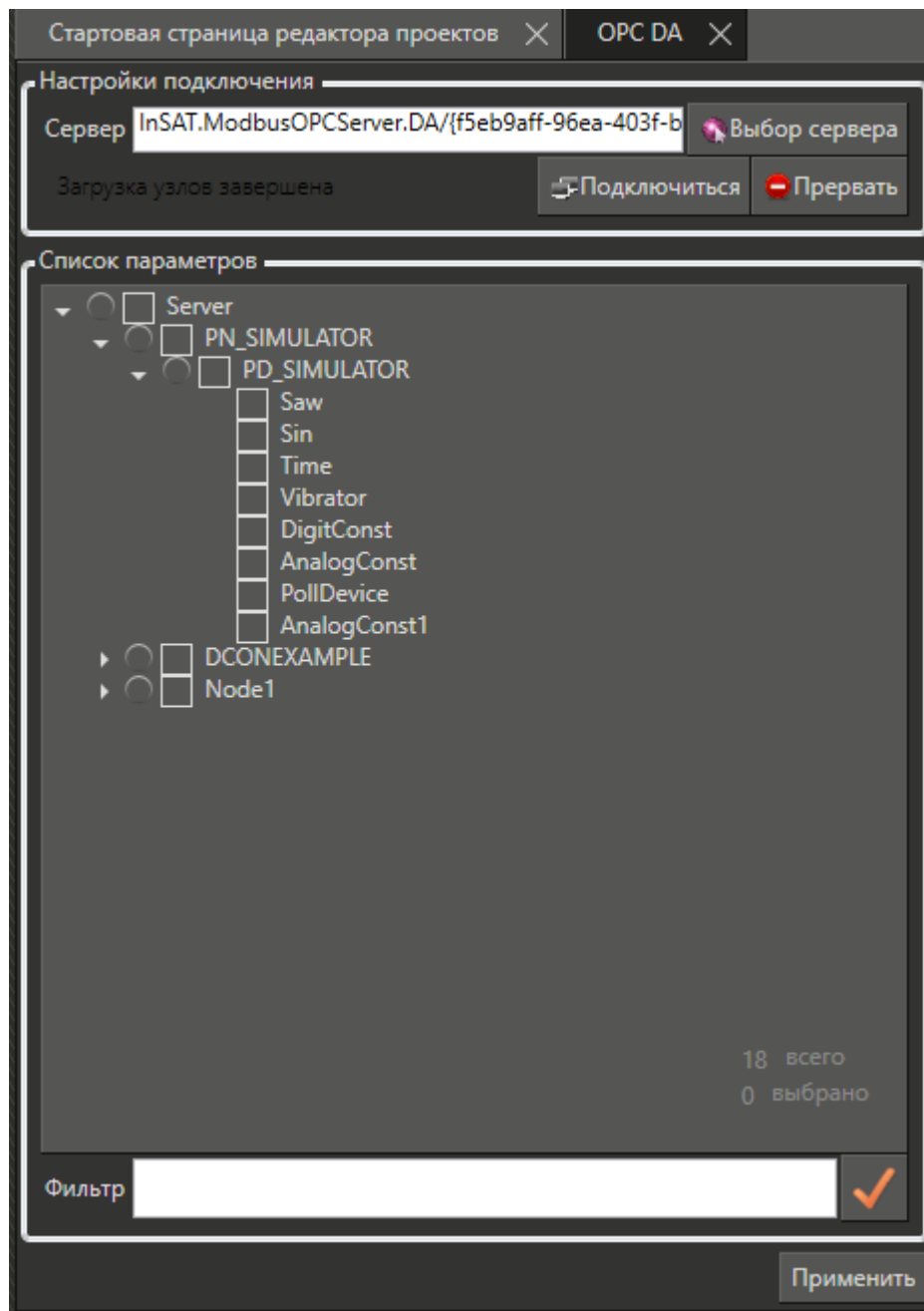
	значения по Подписке, но и все значения в режиме Чтение, с периодом указанным в данном поле. Тип значения свойства -Time.
Период записи	Запись в OPC DA сервер происходит по изменению. Данное поле определяет период, в течении которого происходит накопление данных на запись, после чего происходит их запись 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 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.
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. Рекомендуется как самый эффективный способ получения данных . Если в категории Задача установлен флаг в поле Выполнять по условию, то в протоколе появится параметр 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. Инструкция по настройке поставляется совместно с OPC-серверами.

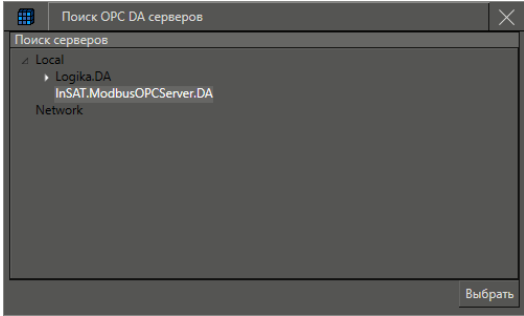

#### 6.3.3.1.4.2. Диалог подключения к серверу OPC DA

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.

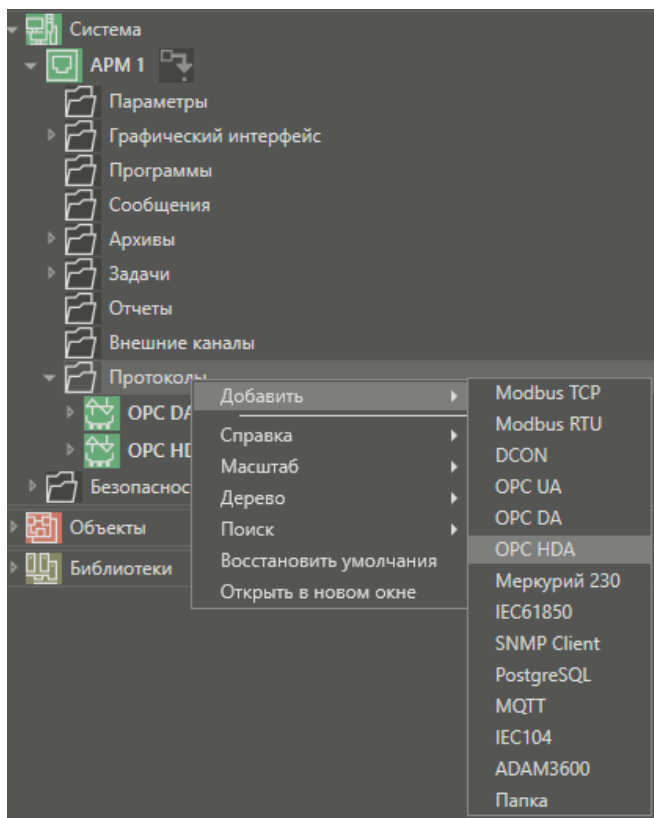
Выбор сервера	<p>Открывает диалог выбора сервера. В нем отображаются все OPC-серверы, установленные на компьютере. Вид окна:</p>  <p>В нем нужно выбрать нужный OPC-сервер и нажать кнопку Выбрать</p>
Прервать	Прерывает подключение к серверу
Список параметров	<p>Отображается конфигурация сервера OPC DA, 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

### 6.3.3.1.5. OPC HDA

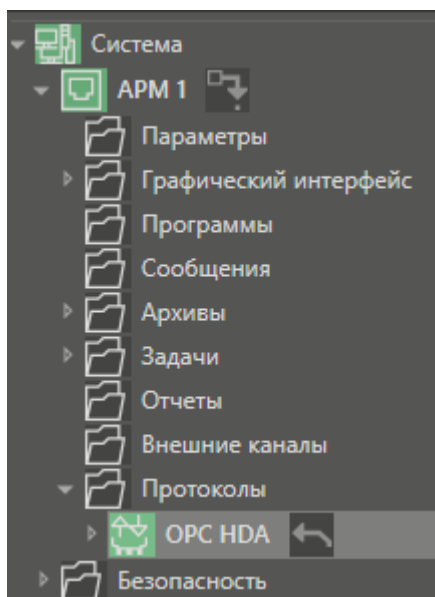
MasterSCADA 4D поддерживает стандарт OPC HDA. Данный протокол описывает получение архивных данных.

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:

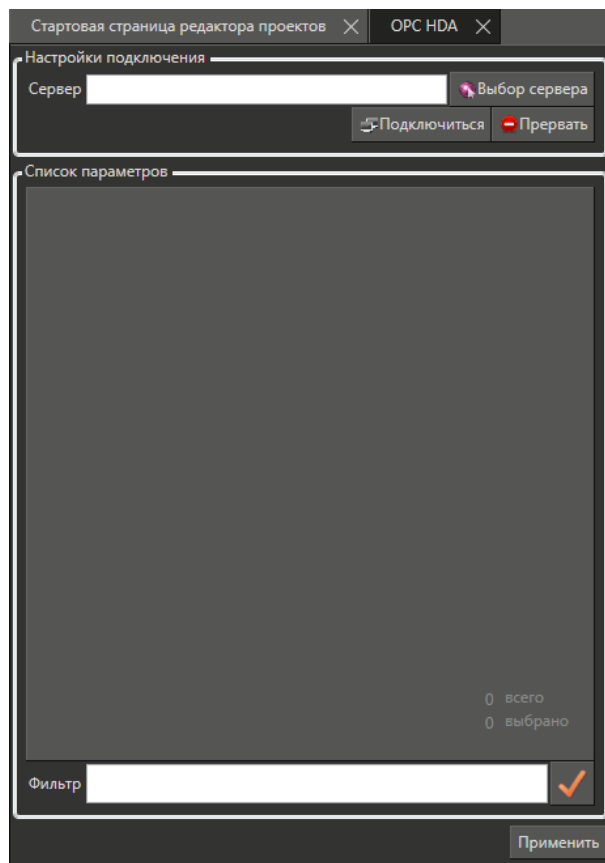


Получим результат в упрощенном дереве:



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.

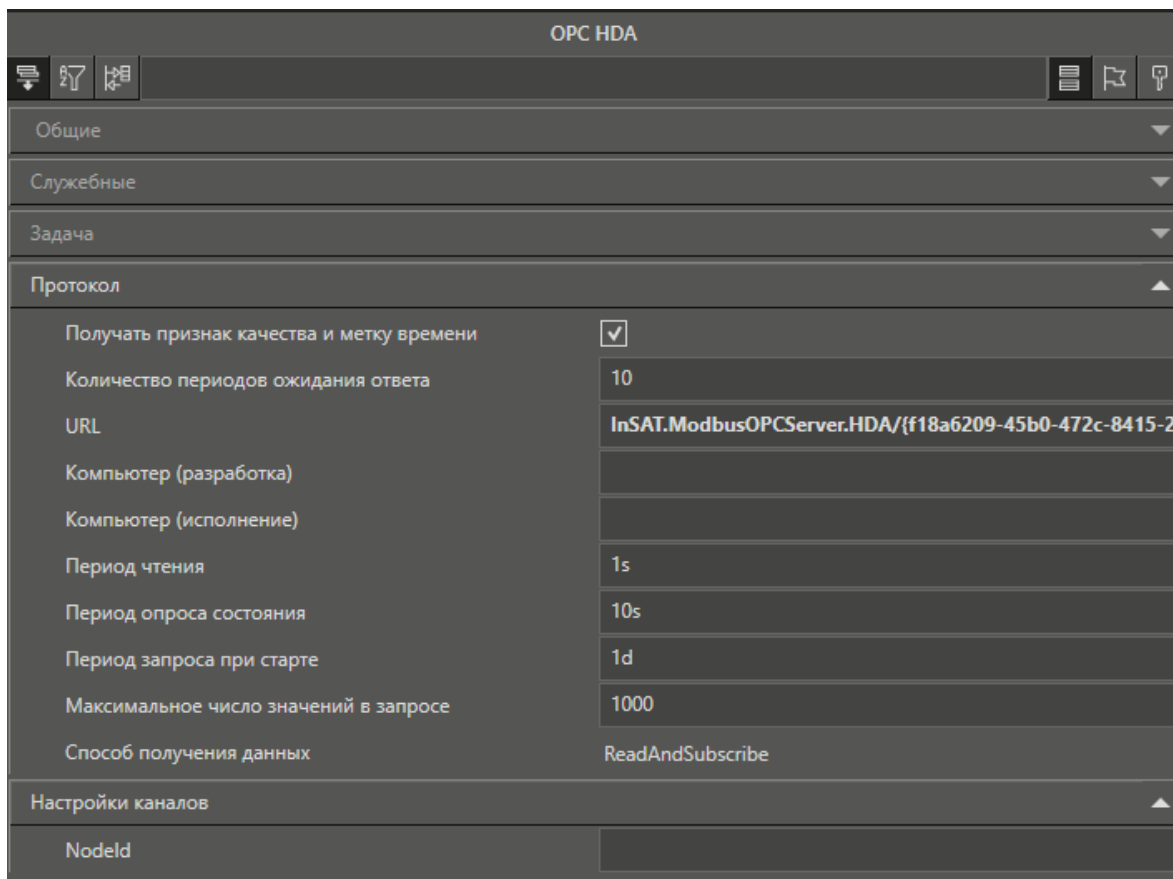
Затем следует нажать кнопку **Подключиться**.

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 предопределенных типов.

После этого необходимо настроить панель свойств OPC HDA сервера.

### 6.3.3.1.5.1. Свойства OPC HDA

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
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

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.
Способ получения данных	OPC HDA клиент может работать в трех режимах: Подписка (Subscribe) - клиент при старте подписывается получать новые значения, желаемый период получения данных указывается в поле Период категории Задача.

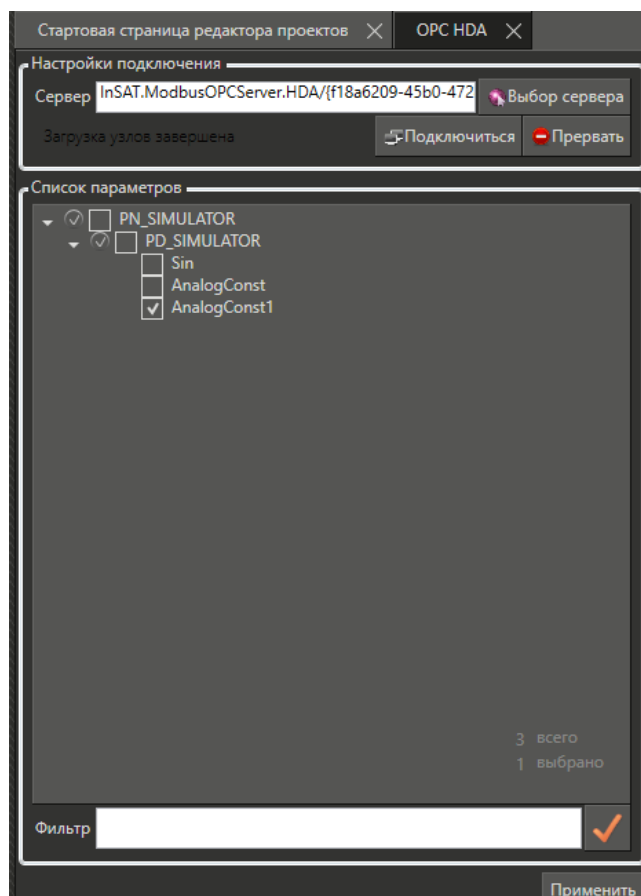


	<p>Чтение и подписка (ReadAndSubscribe) - клиент при старте читает, все значения, которых у него нет. Когда чтение закончено он подписывается на новые значения.</p> <p>Чтение (Read)- клиент с периодом заданным в свойстве Период чтения читает значения из сервера.</p>
NodeId	<p>NodeId - это идентификатор 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. Инструкция по настройке поставляется совместно с OPC-серверами.

### 6.3.3.1.5.2. 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	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.
Выбор сервера	Открывает диалог выбора сервера. В нем отображаются все OPC-серверы, установленные на компьютере. Вид окна: <div data-bbox="487 646 1023 966" data-label="Image"> </div> <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.

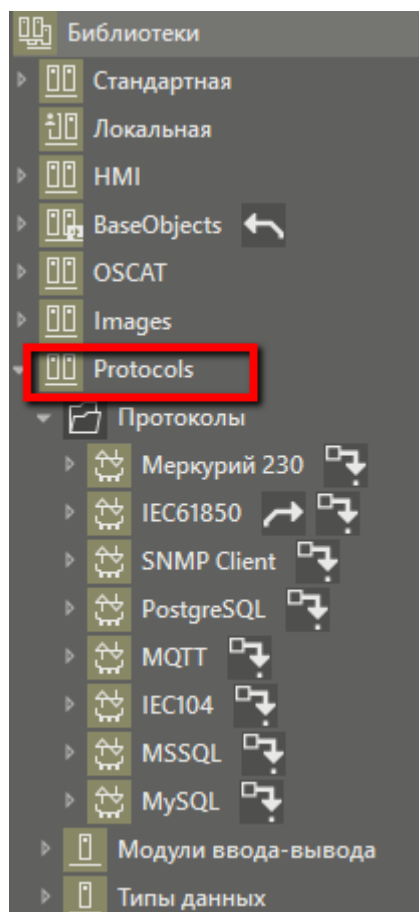
### 6.3.3.2. Дополнительные протоколы

Кроме стандартных протоколов, входящих в состав среды исполнения, в нее можно добавить и другие протоколы.

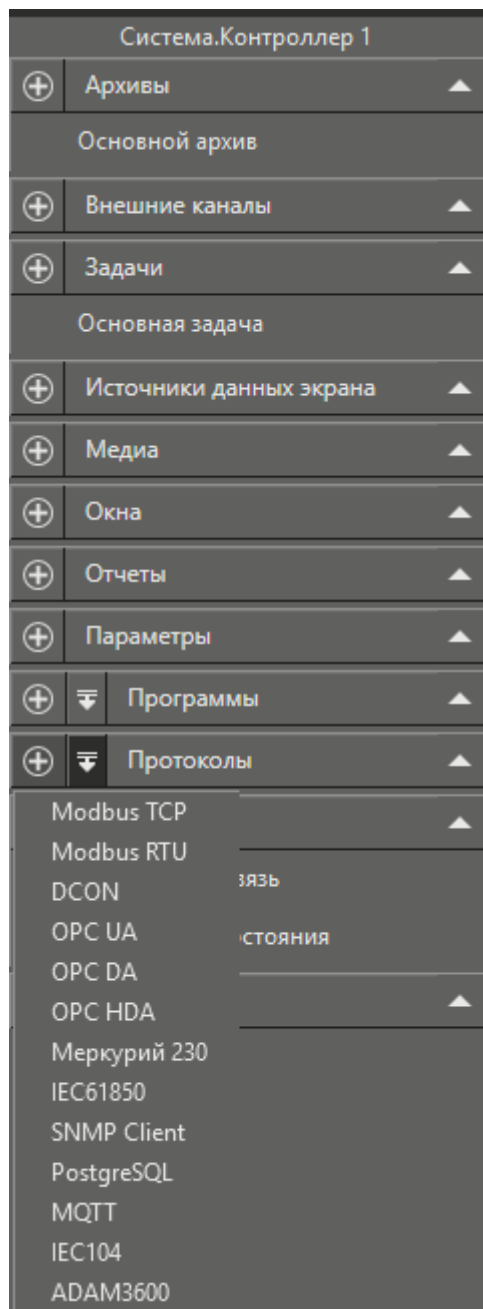
Важно! По умолчанию, дополнительные протоколы не входят в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте [scada@insat.ru](mailto:scada@insat.ru)

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:

- Меркурий 230
- IEC61850
- SNMP Client
- PostGreSQL
- MQTT

- IEC104

Можно подключить следующие библиотеки, содержащие протоколы:

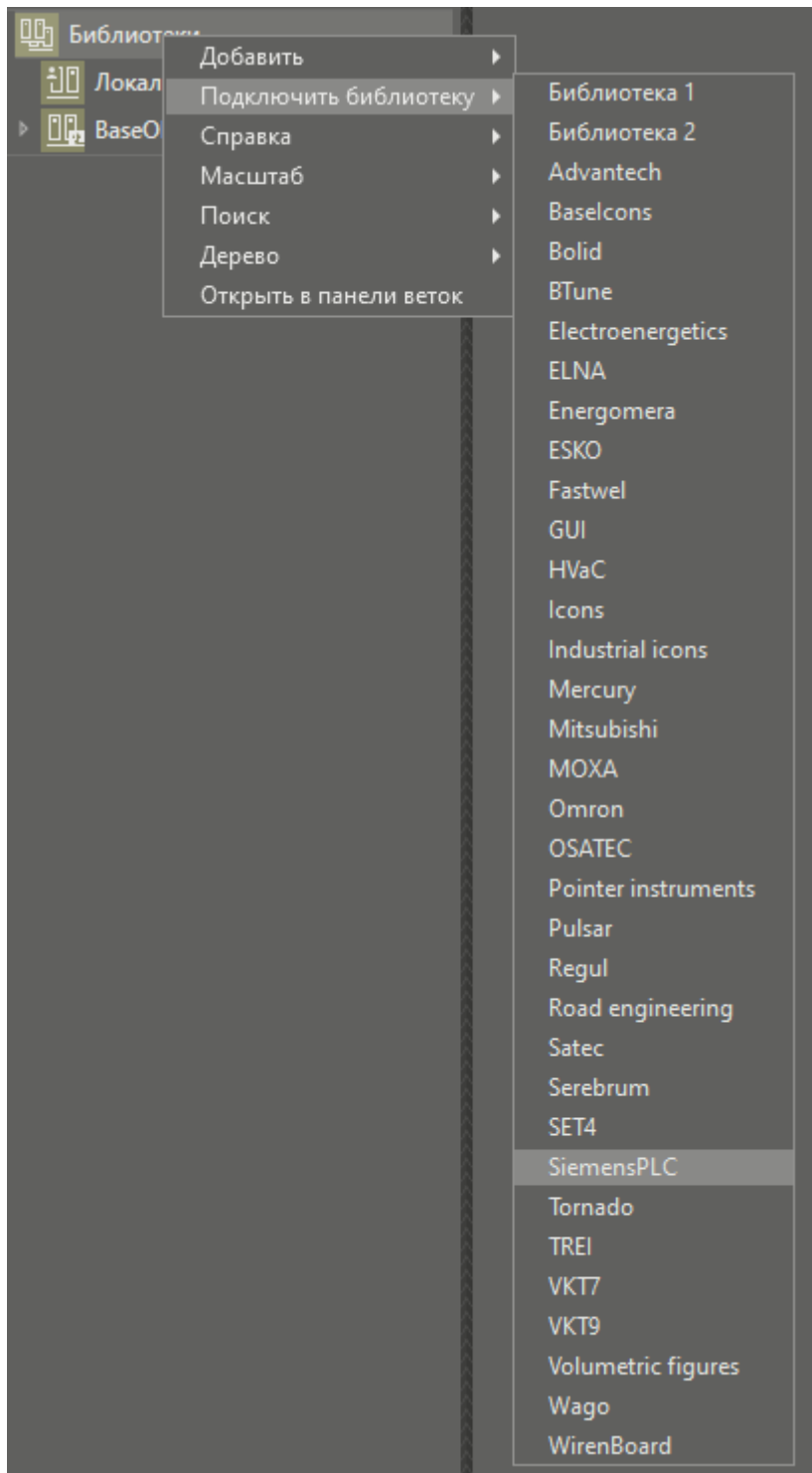
- VKT7
- SET4
- VKT9
- Mercury

### 6.3.3.2.1. SiemensPLC

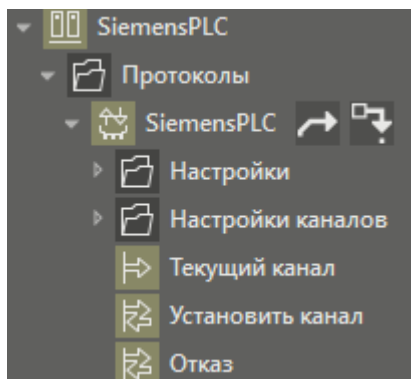
MasterSCADA 4D RT поддерживает работу по протоколу Profinet, который в среде разработки называется SiemensPLC и предназначен для опроса контроллеров фирмы Siemens серии S7 – S7-200, S7-300, S7-400, S7-1200, S7-1500. В качестве интерфейса SiemensPLC использует сети, базирующиеся на TCP/IP - Ethernet, Wi-Fi.

Важно! По умолчанию протокол не входит в состав исполнительной системы. Стоимость и условия лицензирования этого протокола необходимо уточнять по электронной почте [scada@insat.ru](mailto:scada@insat.ru)

Для получения возможности использования протокола в проекте необходимо подключить соответствующую библиотеку:



В полном дереве библиотека SiemensPLC имеет вид:

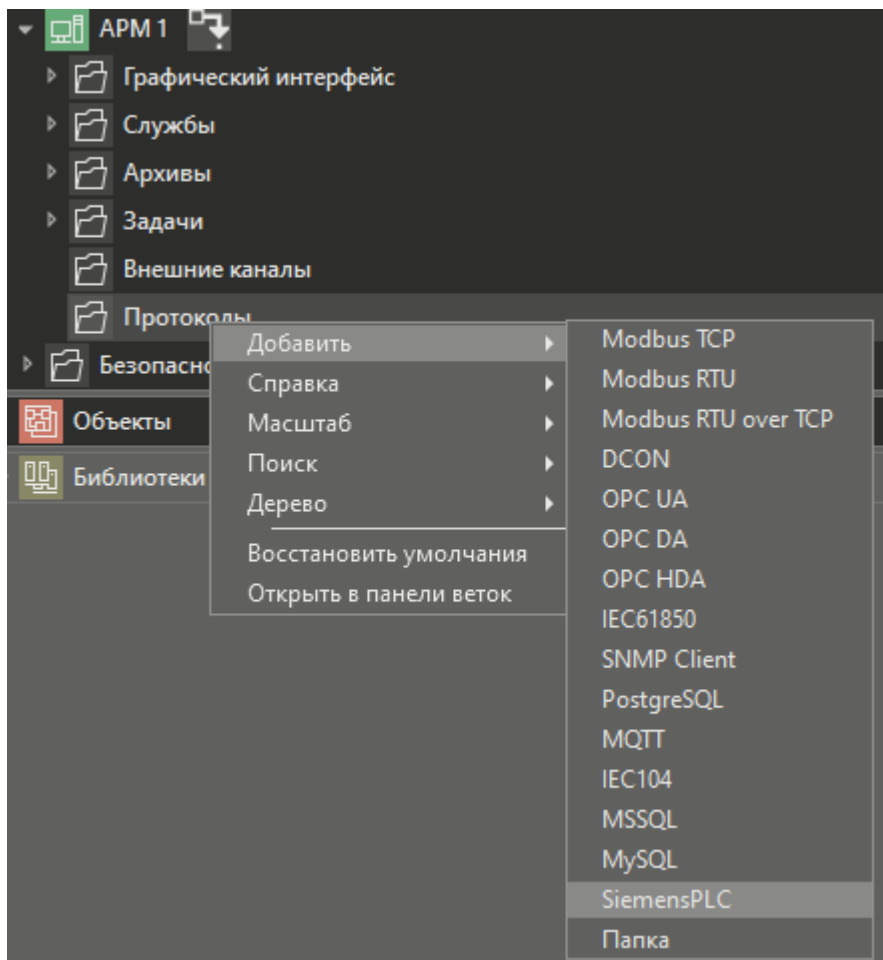


### Назначение параметров

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:

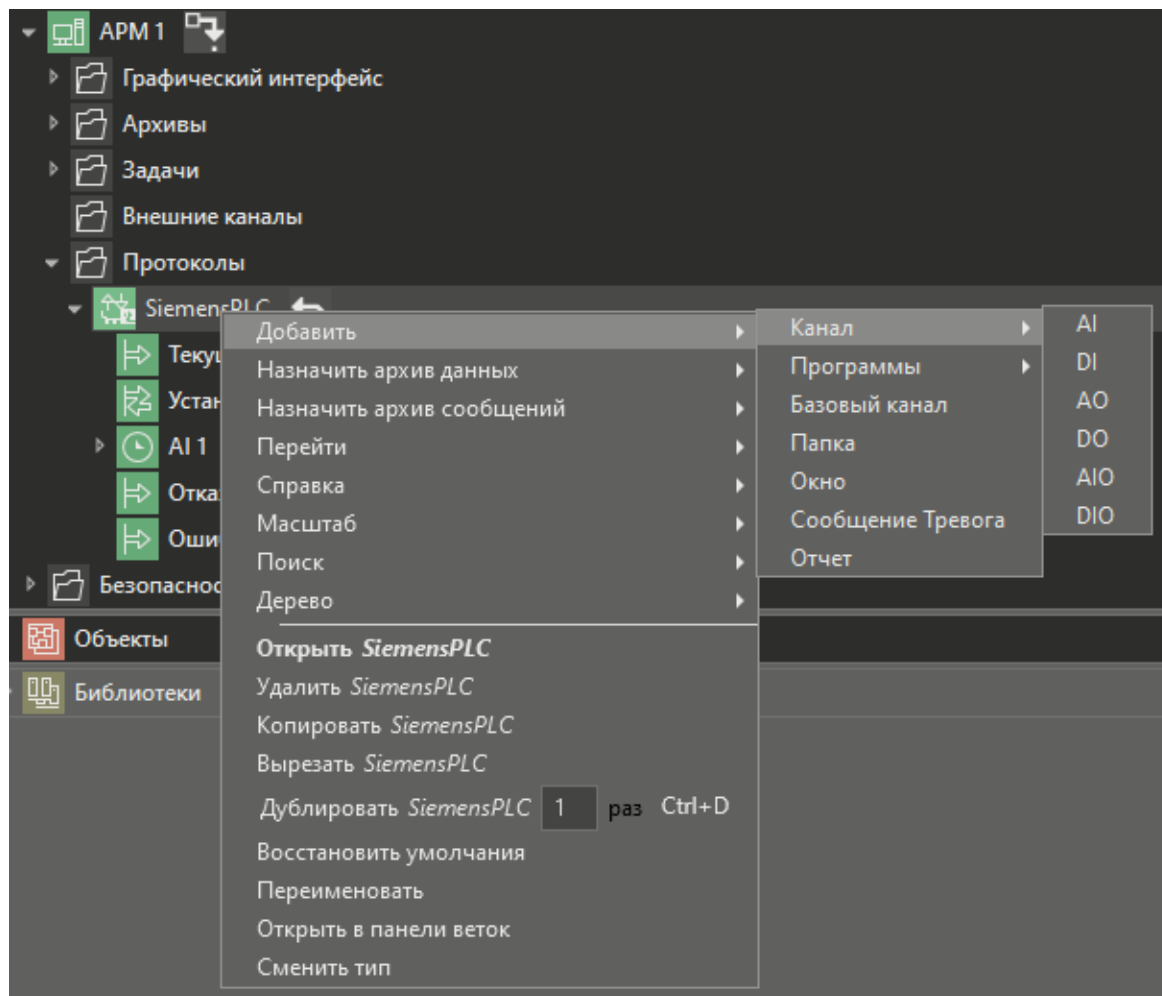


Затем необходимо настроить панель свойств протокола.

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.

### 6.3.3.2.1.1. Свойства SiemensPLC

Вид панели свойств:

SiemensPLC

🔍
🔍
☰
🚩
🔑

Общие ▲

Имя	SiemensPLC
Полное имя	Система.APM 1.Протоколы.SiemensPLC
Метки	
Комментарий	

Служебные ▼

Отношения ▼

Задача ▼

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	

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

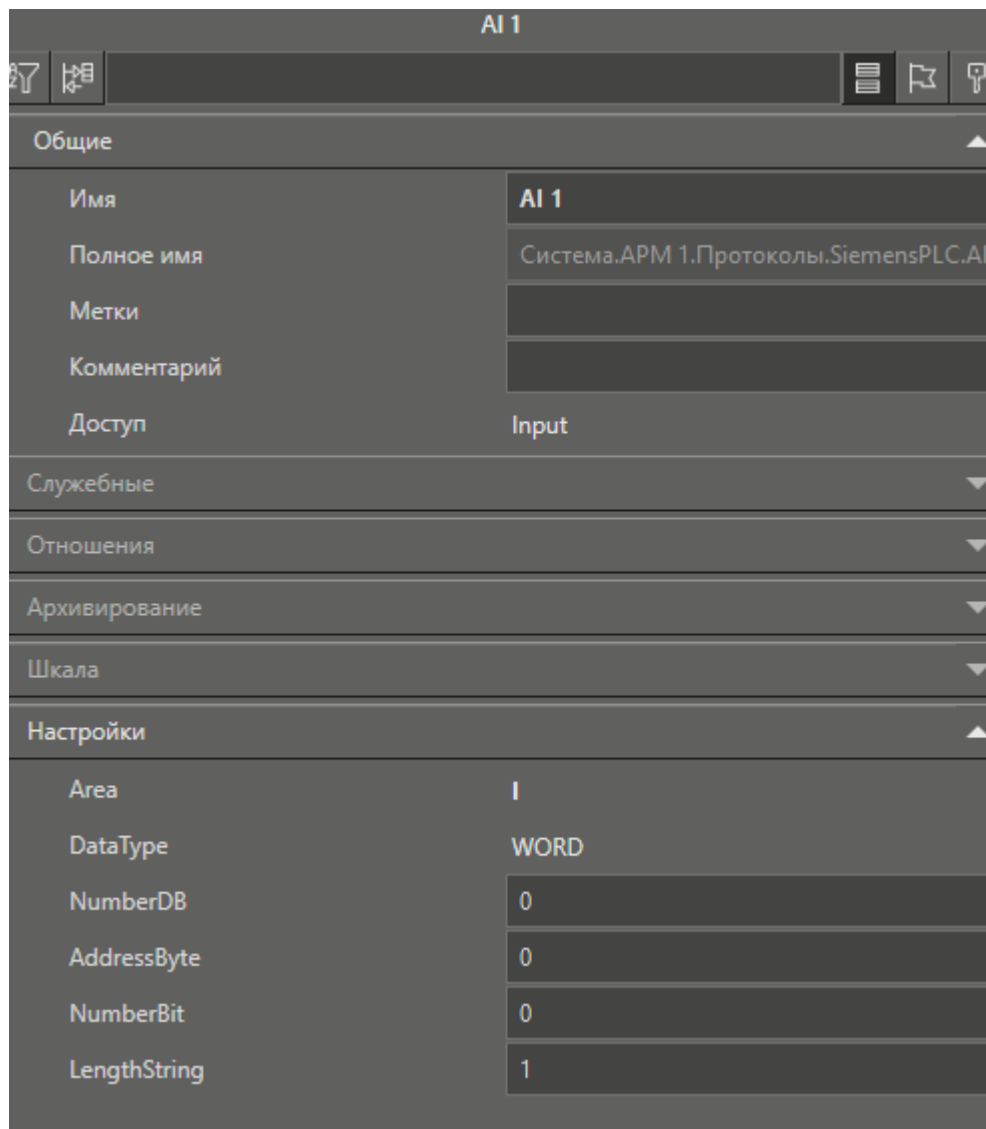
	приведено в разделе, описывающем настройку категории Задачи у протоколов.
Категория Настройки каналов	Задаются значения по умолчанию для создаваемых каналов. В настройке не нуждаются, т.к. индивидуальные настройки каналов формируются автоматически.
IP-адрес	Основной IP-адрес устройства, которое отправляет и получает данные от MasterSCADA 4D.  <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>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.</p> </div>
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"> <li>• вручную, для этого нужно указать номер канала в параметре протокола Установить канал</li> <li>• 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</li> </ul>

	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.

### 6.3.3.2.1.2. 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"> <li>• I – входы.</li> <li>• Q – выходы.</li> <li>• M – ячейки памяти.</li> <li>• DB – блоки данных (DataBlock). При этом у DataBlock должна быть выключена настройка Optimized block access.</li> </ul>

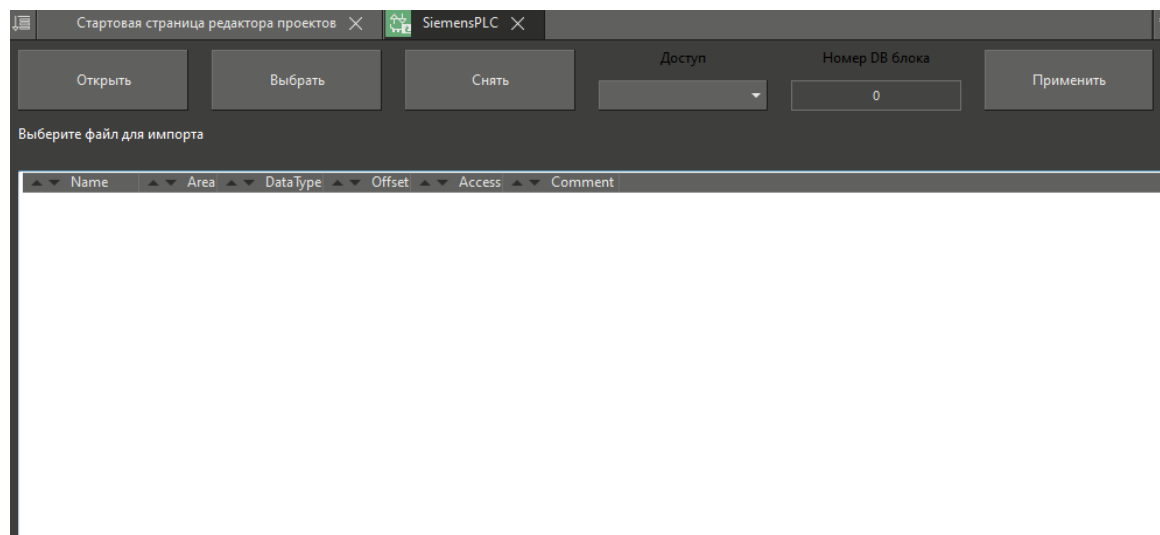
	<ul style="list-style-type: none"> <li>• Т – таймеры (обратный отсчет оставшегося времени, в миллисекундах).</li> <li>• С – счетчики (накопленное значение).</li> </ul>
DataType	Задается тип данных опрашиваемой переменной в контроллере. Значение подбирается в зависимости от установленного типа данных 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

### 6.3.3.2.1.3. 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.

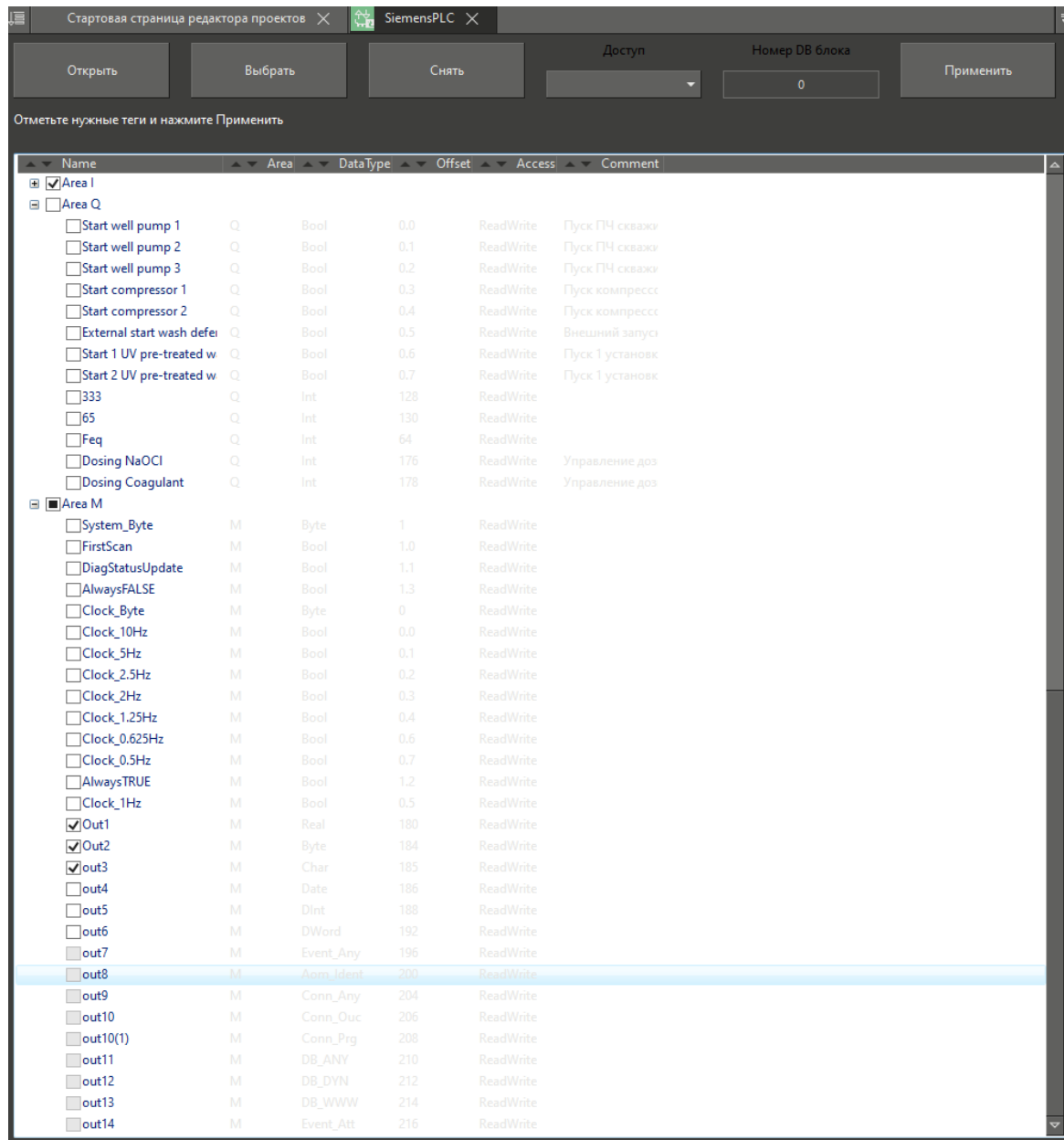
Вид вкладки по умолчанию:



Описание элементов управления:

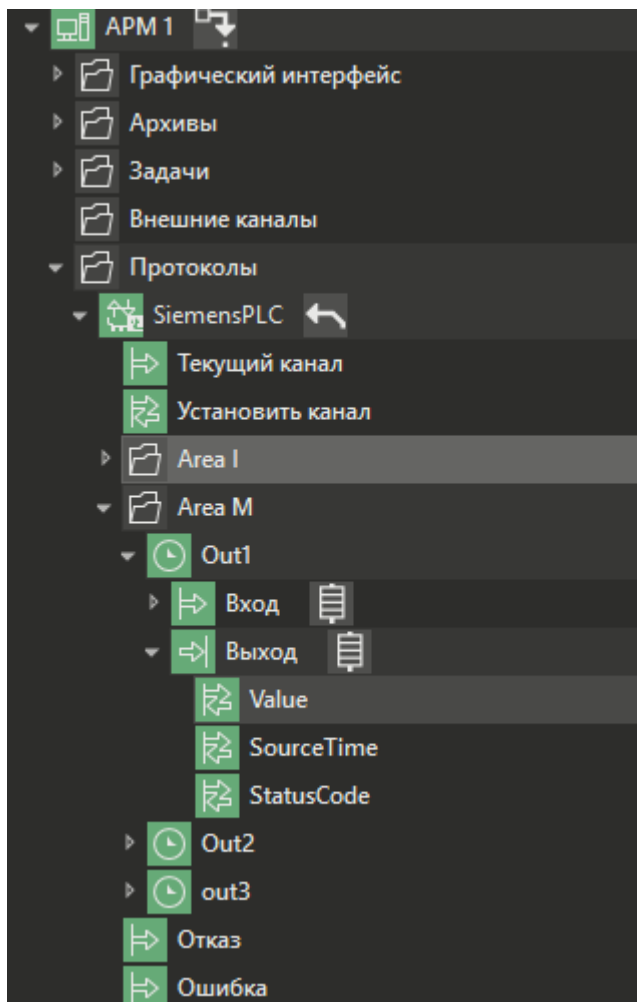
Элемент	Описание
Открыть	Открывает диалоговое окно Windows, в котором можно выбрать файл, из которого нужно выполнить импорт данных. Позволяет открыть файлы Siemens Export File (*.db, *.xlsx). Для открытия файла типа *.xlsx, на компьютере должен быть установлен MS Office. После того как файл будет открыт 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	
Номер DB блока	
Применить	После нажатия на кнопку Применить отмеченные флагами каналы будут добавлены 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

### 6.3.3.2.2. Тепловычислители Теплоком

MasterSCADA 4D поддерживает работу с тепловычислителями (теплосчетчиками) компании Теплоком - ВКТ-7 (VKТ7) и ВКТ-9 (VKТ9).

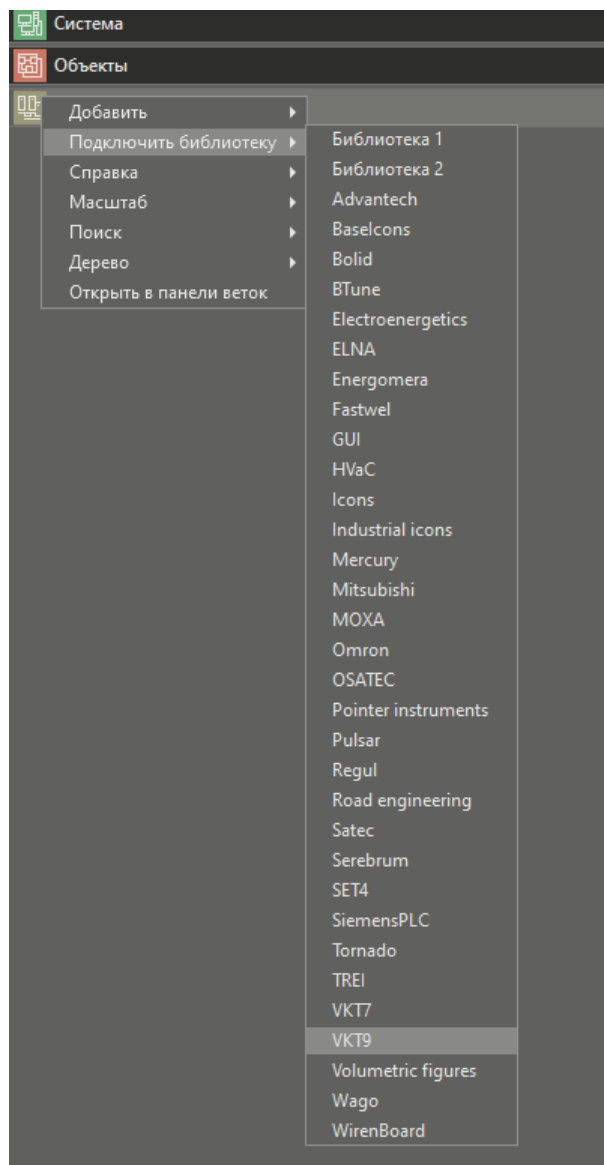
Работа с этими устройствами возможна через последовательные интерфейсы (RS-232, RS-485), а при использовании конвертеров Ethernet-COM (Моха NPort 5150 и т.п.) также и по сетям TCP/IP (Ethernet, Wi-Fi).

#### 6.3.3.2.2.1. VKТ9

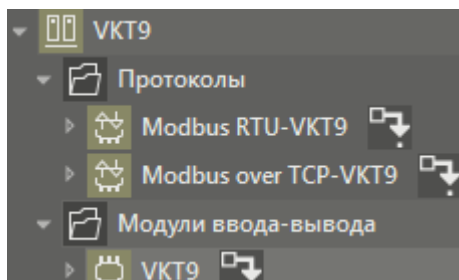
MasterSCADA 4D RT работает с теплосчетчиками VKТ9 компании Теплоком по платформонезависимым протоколам, которые поддерживаются исполнительными системами, предназначенными для различных операционных систем.

Важно! По умолчанию протокол не входит в состав исполнительной системы. Стоимость и условия лицензирования этого протокола необходимо уточнять по электронной почте [scada@insat.ru](mailto:scada@insat.ru)

Для получения возможности использования теплосчетчиков VKT9 в проекте необходимо подключить соответствующую библиотеку:

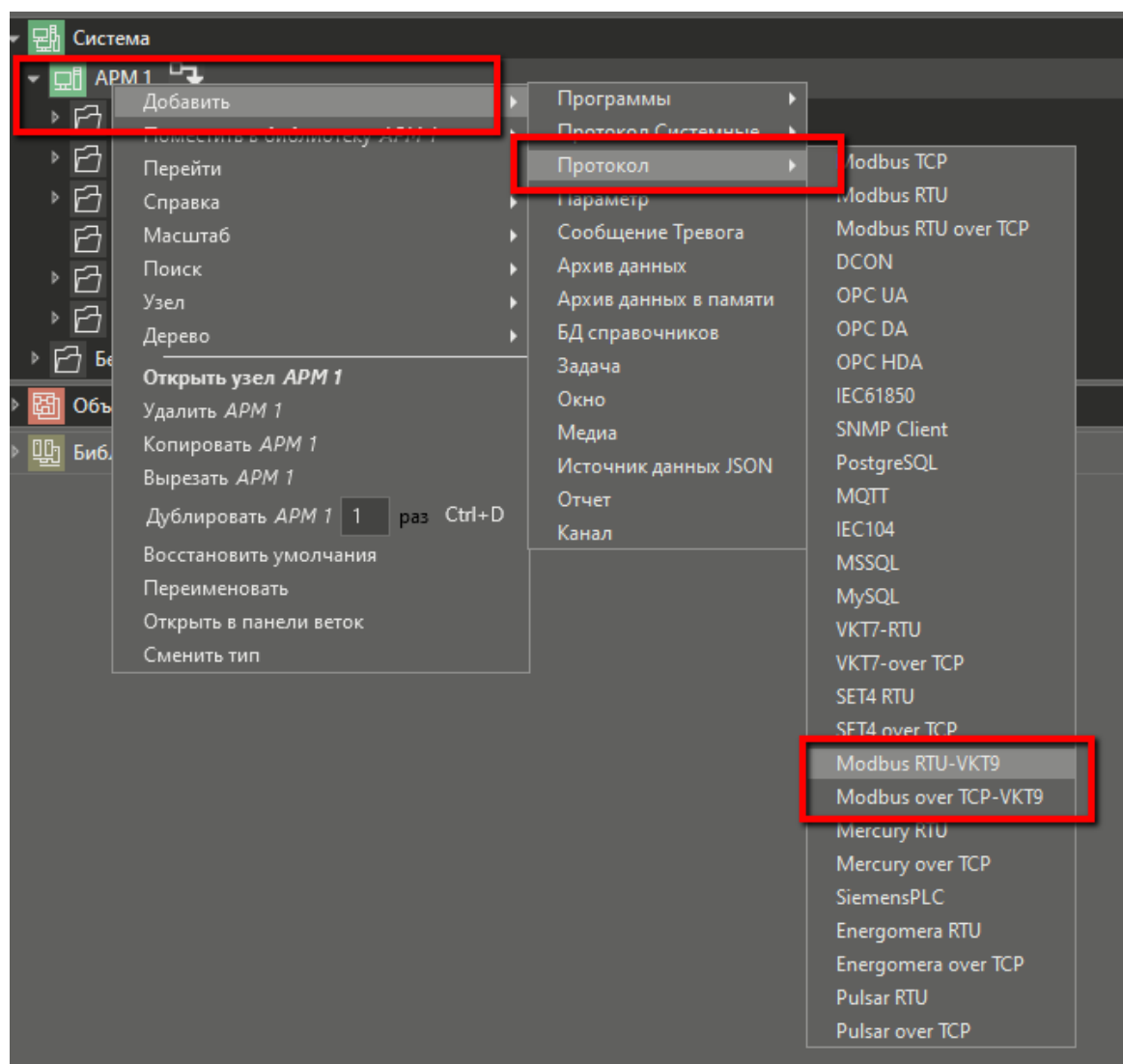


В полном дереве библиотека VKT9 имеет вид:



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.

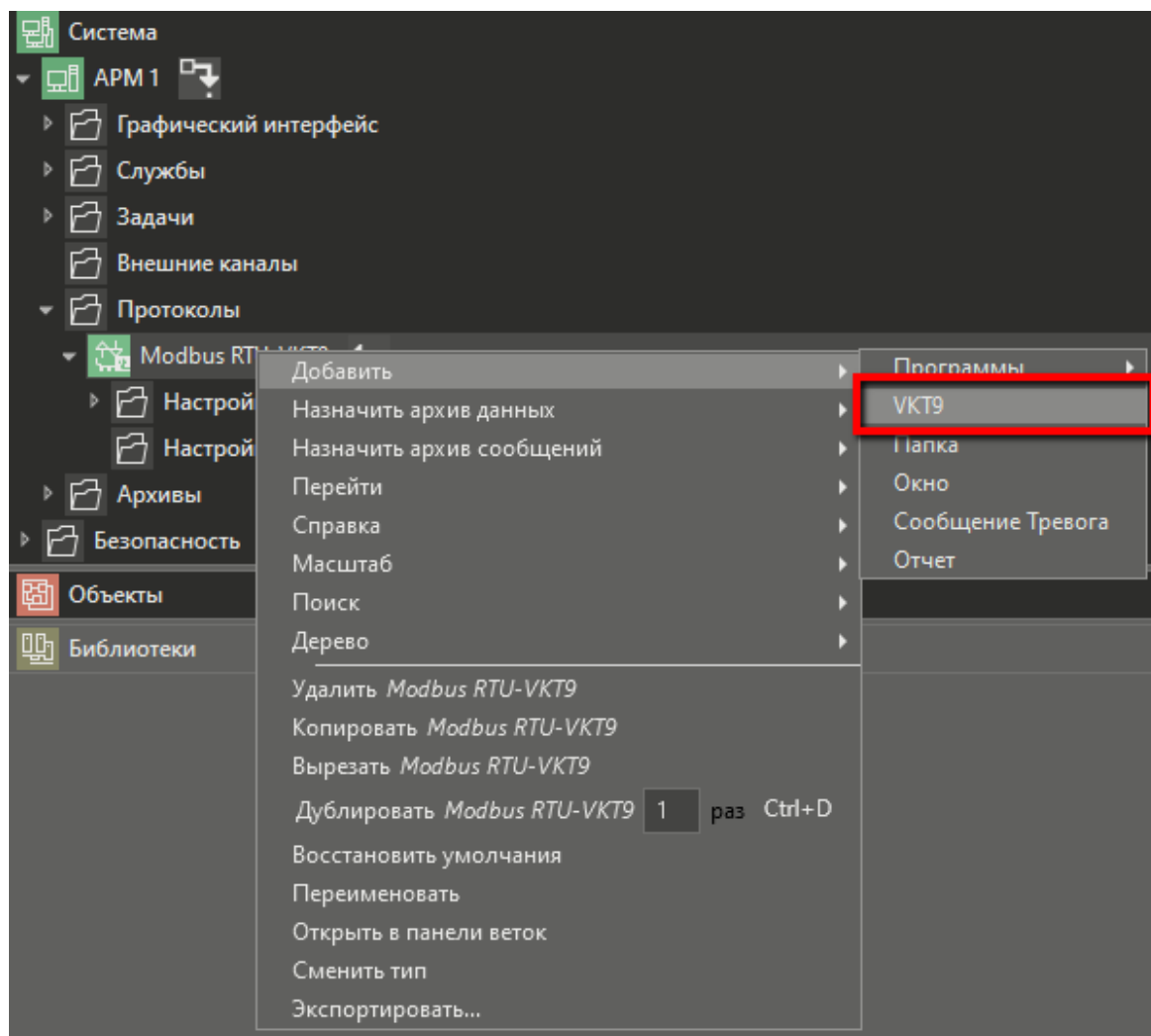
Рекомендуется добавлять элементы в проект при помощи контекстного меню:



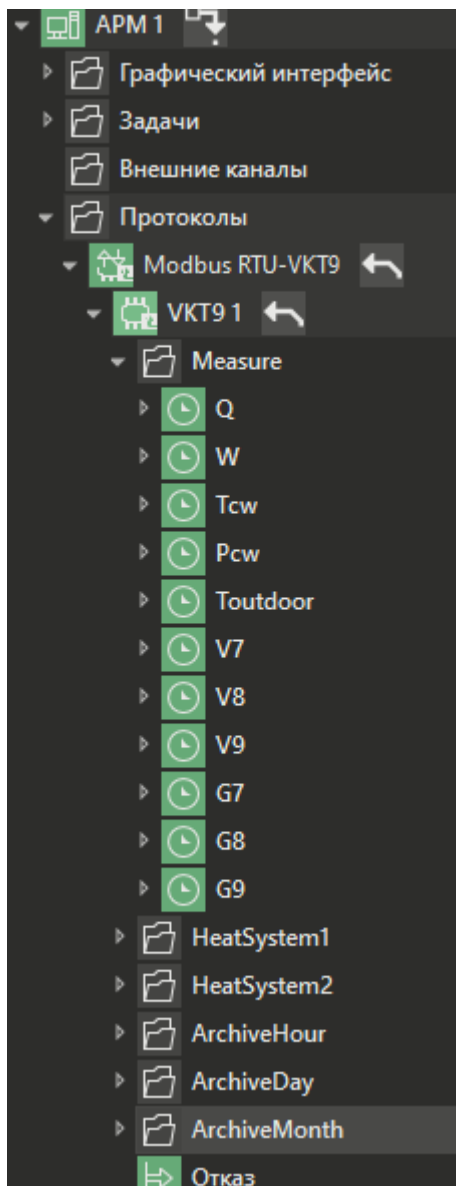
Тип добавляемого протокола зависит от реального способа подключения теплосчетчика:

- 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 тепло-счетчику 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. .

Для корректной работы добавленного элемента необходимо настроить его панель свойств.

## СВОЙСТВА VKT9

Панель свойств элемента VKT9 имеет вид:

The screenshot shows a configuration window for an element named 'VKT9 1'. The window has a title bar with the element name and several icons. Below the title bar, there are three main sections: 'Общие' (General), 'Служебные' (Service), and 'Настройки' (Settings). The 'Общие' section contains three rows: 'Имя' (Name) with the value 'VKT9 1', 'Метки' (Tags), and 'Комментарий' (Comment). The 'Служебные' section contains three rows: 'Задача' (Task), 'Масштабирование' (Scaling), and 'Настройки' (Settings). The 'Настройки' section contains four rows: 'Адрес' (Address) with the value '0', 'Максимальный разрыв адресов' (Maximum address gap) with the value '5', 'Отслеживать качество измерений' (Track measurement quality) with a checked checkbox, and 'Количество считываемых записей архива за цикл' (Number of archive records per cycle) with the value '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>

	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 VKT9

Элемент, соответствующий реальному тепловычислителю VKT9, имеет выход Отказ, а также группы каналов, согласно его максимальной конфигурации:

Название группы	Назначение
Measure	Общие измеренные значения
Heat-System1	Параметры теплосистемы 1
Heat-System2	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
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
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
Tcw	Температура холодной воды	HS1_P2	Давление канала 2
Pcw	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
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
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		

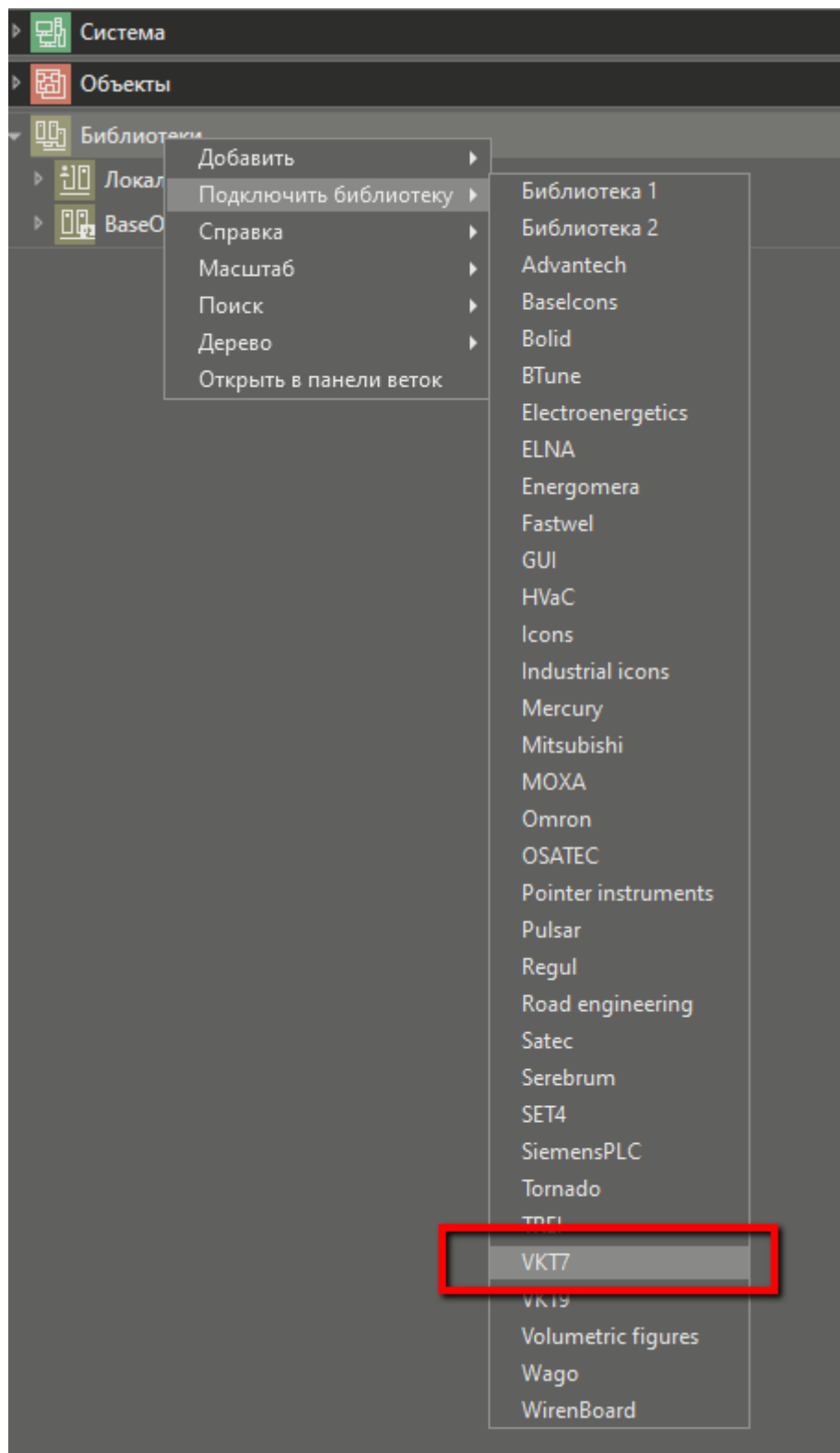
Каналы содержат типовые параметры.

#### 6.3.3.2.2.2. VKT7

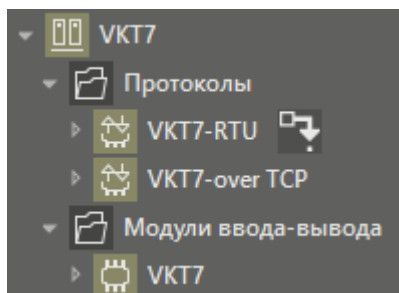
MasterSCADA 4D RT может работать с теплосчетчиками VKT7 компании Теплоком по платформонезависимому протоколу, который может использоваться в исполнительных системах, предназначенных для различных операционных систем.

Важно! По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте [scada@insat.ru](mailto:scada@insat.ru)

Чтобы была возможность использовать теплосчетчики VKT7 в проекте необходимо подключить соответствующую библиотеку:

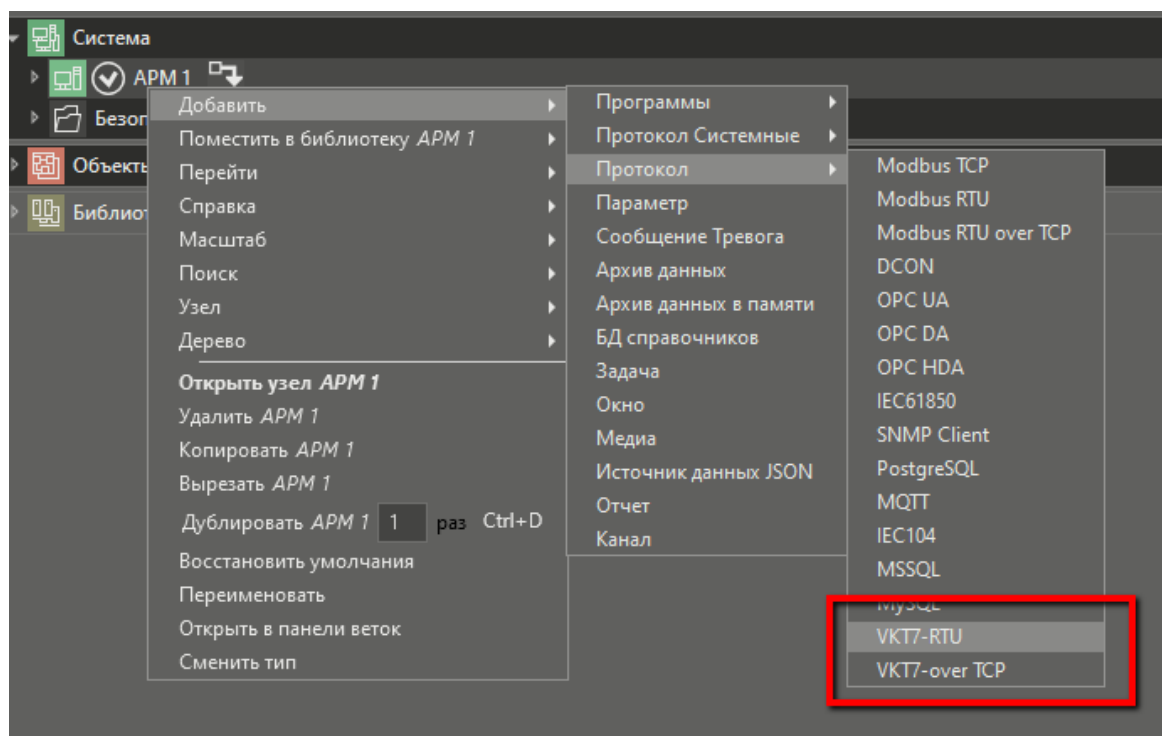


В полном дереве библиотека VKT7 имеет вид:



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.

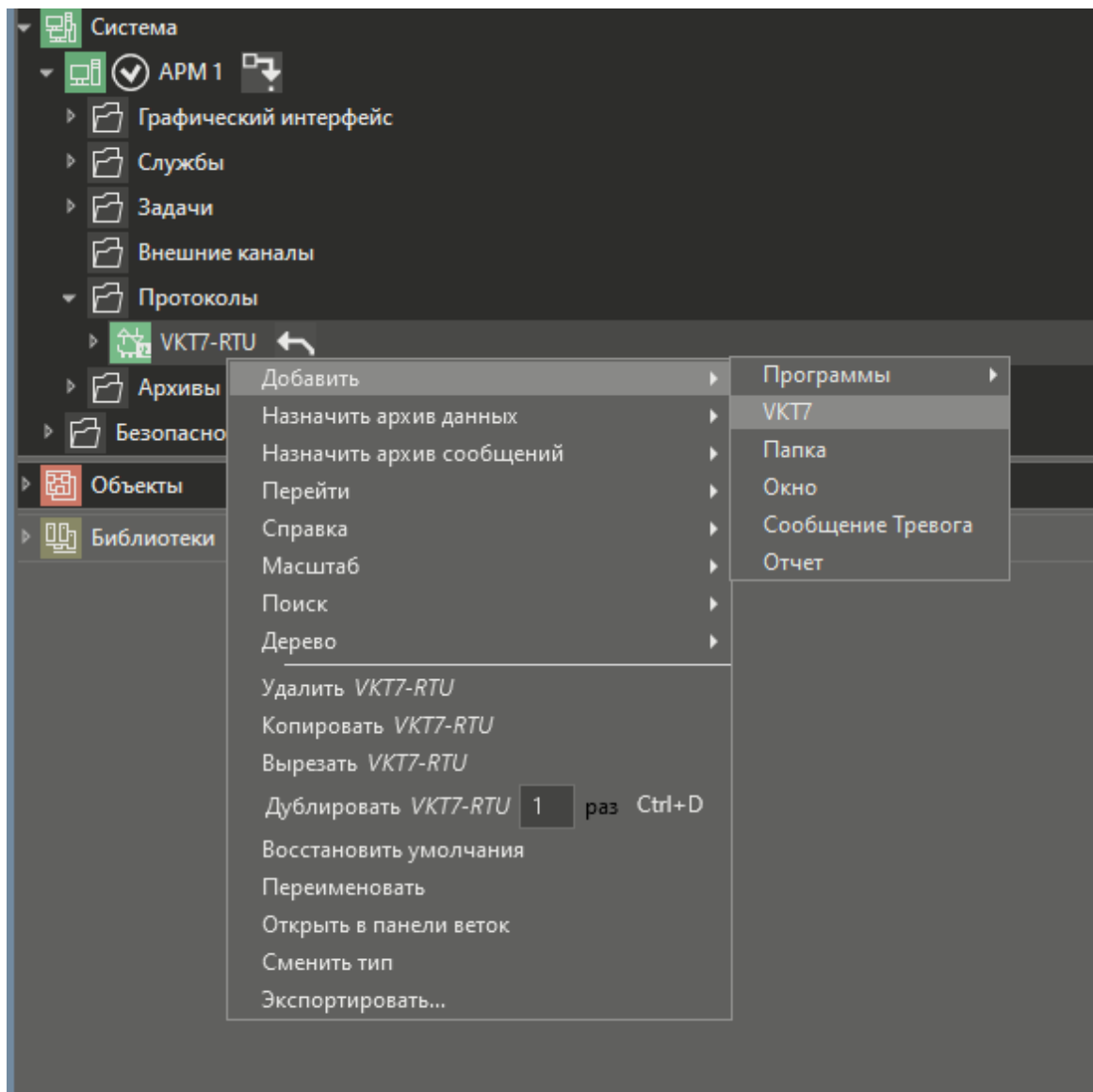
Рекомендуется добавлять элементы в проект при помощи контекстного меню:



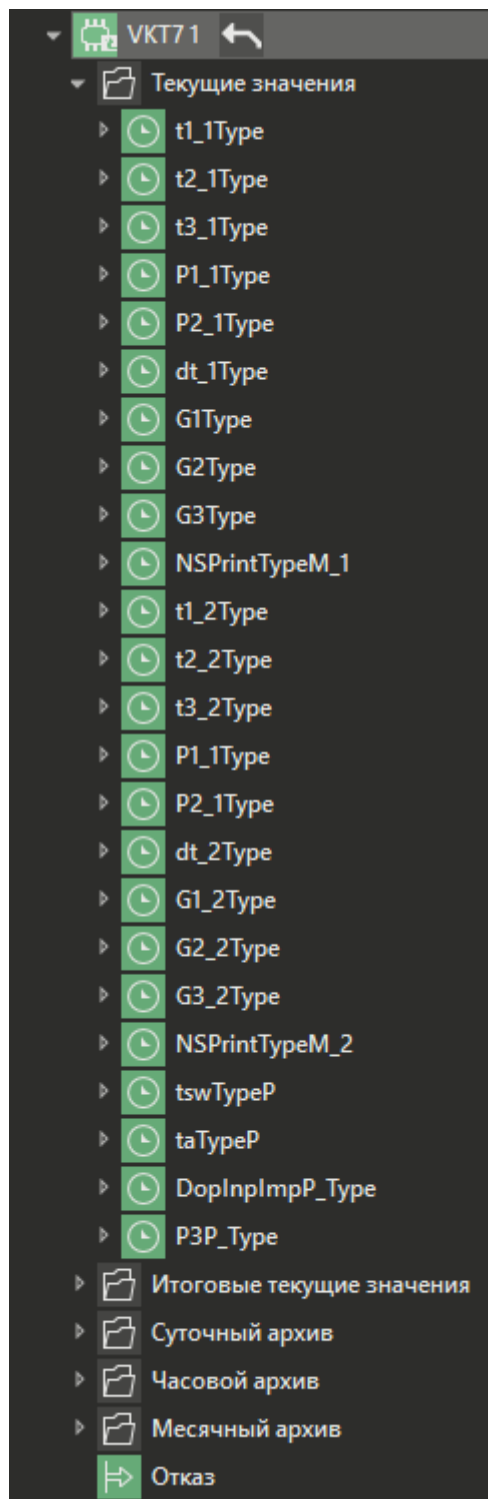
Тип добавляемого протокола зависит от реального способа подключения теплосчетчика:

- 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:

VKT7 1	
Общие	
Имя	VKT7 1
Метки	
Комментарий	
Служебные	
Задача	
Масштабирование	
Настройки	
Адрес	0
Глубина чтения часового архива	20
Глубина чтения суточного архива	20
Глубина чтения месячного архива	5
Число записей архива за раз	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	Задает 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.
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> <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 ВКТ7

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 ВКТ7, 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

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	DopIn-rImpP_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
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_2Type	Время нормальной работы Тв2
Qnt_1Type	Время нормальной работы Тв1	Qnt_2Type	Время отсутствия счета Тв2
Qnt_1Type	Время отсутствия счета Тв1	DopImpP_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	DopImpP_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
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_2TypeReNIP	Время нормальной работы Тв2
tswTypeP	Температура холодной воды	Qnt_2TypeReP	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.

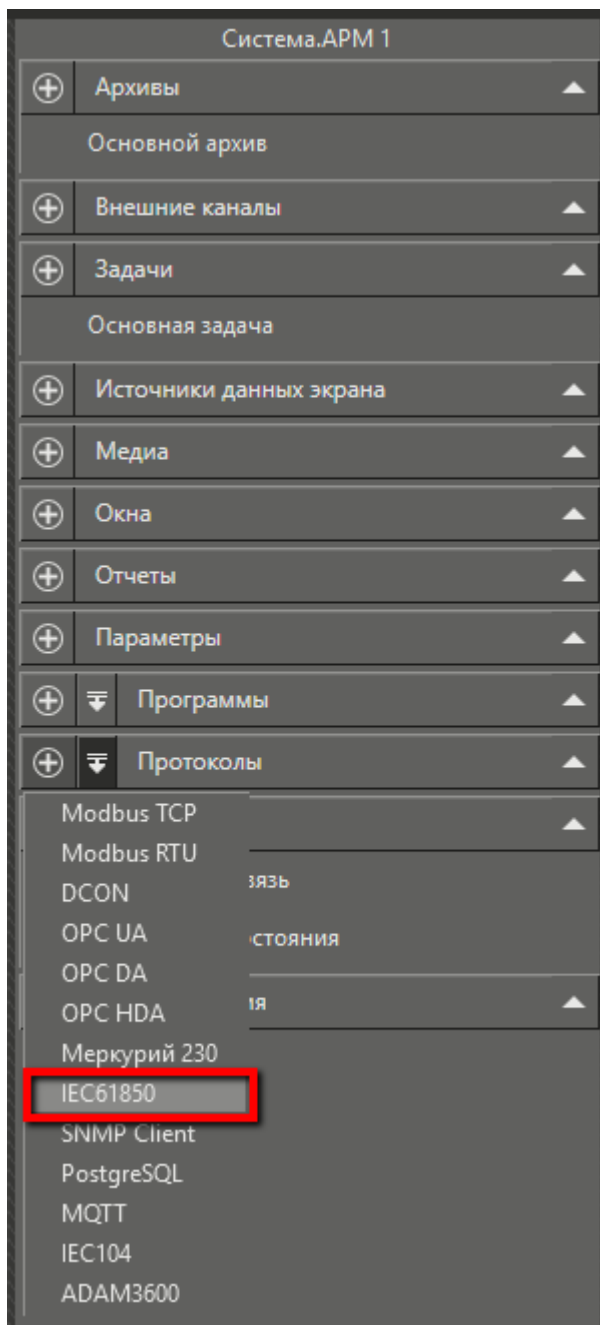
OpсUa\_BadNoCommunication - формируется, если исполнительная система запрашивает значение сигнала, а устройство сигнализирует об отсутствии сигнала среди активных тегов для опроса. Как правило это означает, что канал есть в конфигурации, но переменная отсутствуют в приборе.

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).

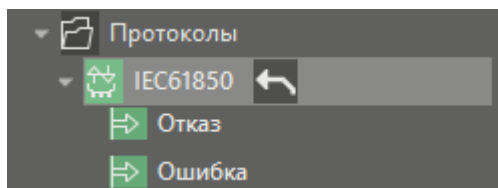
### 6.3.3.2.3. IEC61850

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:



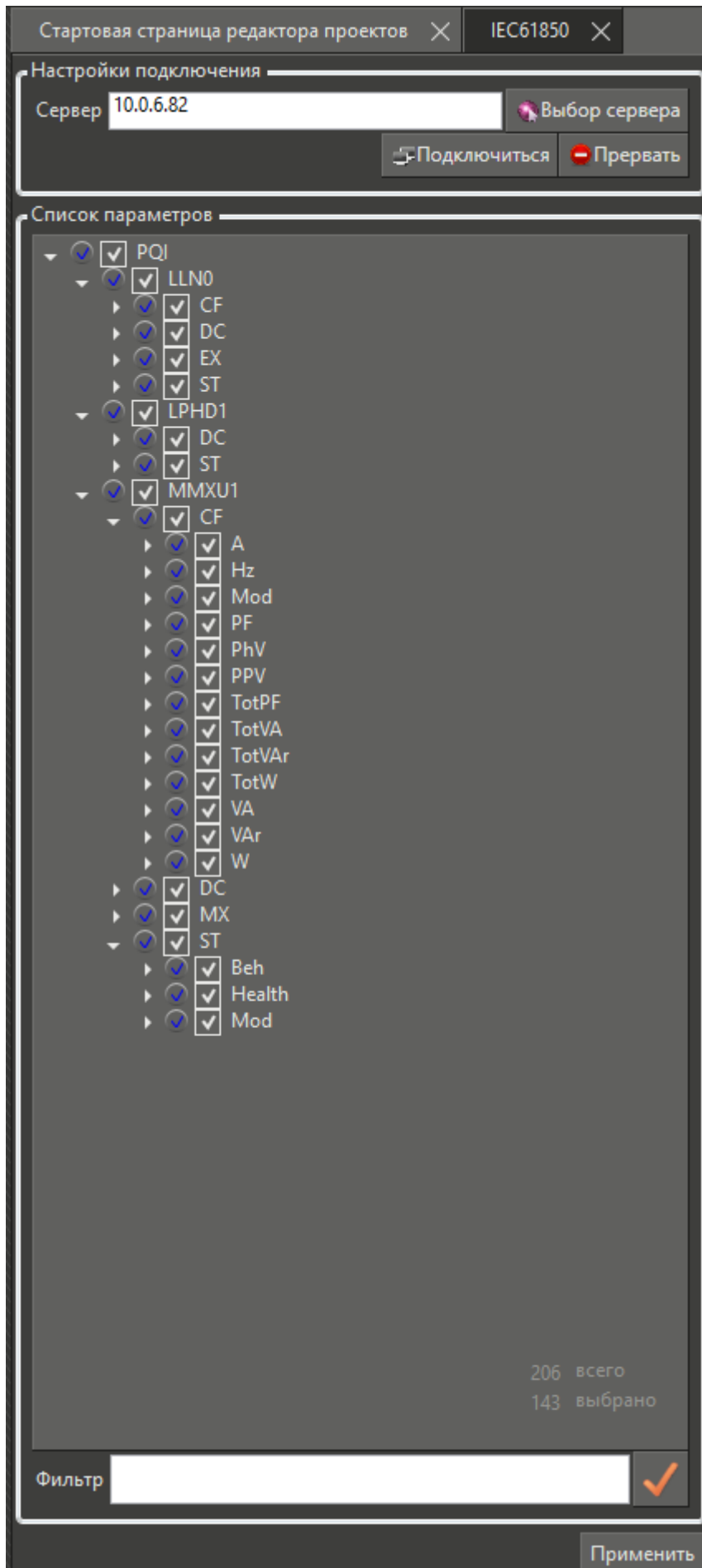
Затем, для того чтобы добавить каналы, необходимо выполнить подключение к серверу.



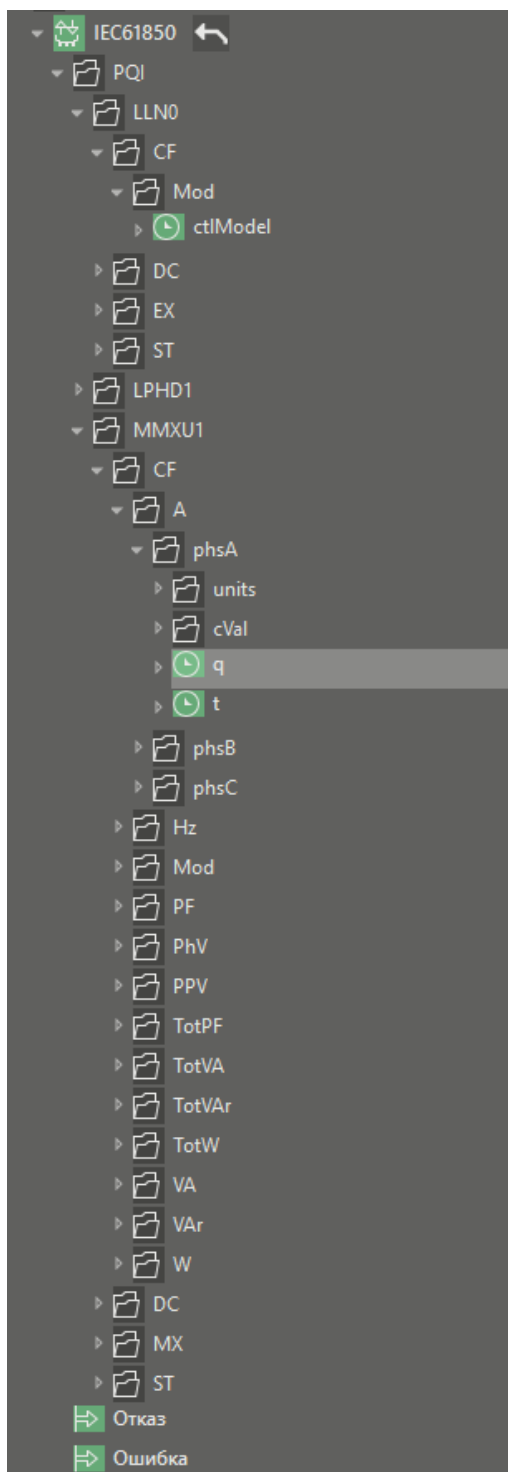
Для этого нужно дважды нажать левую кнопку мыши на элементе в дереве. При этом откроется вкладка:



В строке Сервер необходимо указать IP-адрес устройства, на котором установлен IEC61850-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.

Смотрите также:

Каналы предопределенных типов

### 6.3.3.2.3.1. Свойства IEC61850

Вид панели свойств:

IEC61850	
<b>Общие</b>	
Имя	IEC61850
Программное имя	IEC61850
Полное имя	Система.APM 1.Протоколы.IEC61850
Метки	
Комментарий	
<b>Служебные</b>	
<b>Отношения</b>	
<b>Задача</b>	
<b>Настройки каналов</b>	
NodeID	
FC	
<b>Протокол</b>	
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 способ работы протокола при резервировании узла. Подробное

	описание категории приведено в разделе, описывающем настройку категории Задачи у протоколов.
Настройки каналов	Задаются значения по умолчанию для создаваемых каналов. В настройке не нуждаются, т.к. индивидуальные настройки каналов формируются автоматически.
IP-адрес	IP-адрес сервера, который отправляет и получает данные от MasterSCADA 4D.
TCP-порт	Номер TCP-порта сервера, по которому осуществляется подключение.
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. <b>Важно!!!</b> 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	Определяет количество параметров, опрашиваемых в одном запросе.
Каталог для записи файлов	Задается каталог на локальном носителе, в который будет производиться запись файлов, полученных от сервера IEC 61850.
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.

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_CALL

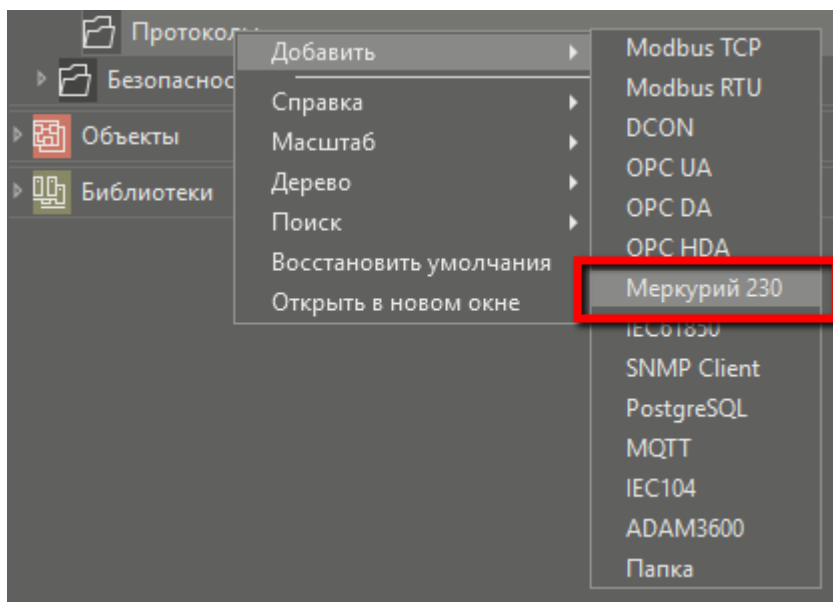
QUALITY_DETAIL_BAD_REFERENCE	16	Потеря калибровки.	SENSOR_CALL
QUALITY_DETAIL_OSCILLATORY	32	Дребезг контакта (частое включение-выключение дискретного сигнала).	UNCERTAIN
QUALITY_DETAIL_FAILURE	64	Внутренняя или внешняя неисправность.	UNCERTAIN
QUALITY_DETAIL_OLD_DATA	128	Данные не обновлялись в течении долгого времени.	UNCERTAIN
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

#### 6.3.3.2.4. Меркурий 230

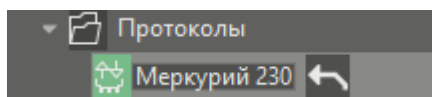
Протокол обмена со счетчиками электроэнергии Меркурий-230 по последовательному интерфейсу. При обмене узел играет роль ведущего. Поддерживается считывание архивов из счетчика.

Важно! По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте [scada@insat.ru](mailto:scada@insat.ru)

Добавить протокол в дерево системы можно при помощи контекстного меню или контекстной панели узла:



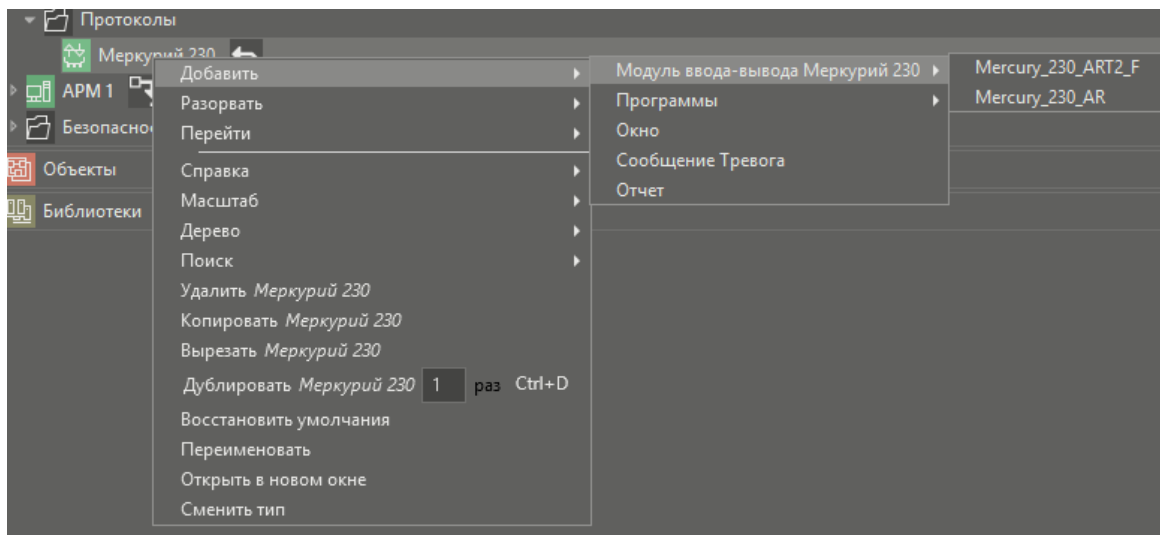
В результате получим:



Модули ввода-вывода подключаются к устройству, в котором работает исполнительная система MasterSCADA 4D по COM-порту. Следовательно, в панели свойств протокола необходимо задать соответствующие настройки.

Затем при помощи контекстного меню протокола можно добавить модули ввода-вывода, которые уже содержат все необходимые каналы:





Поддерживаются следующие модули ввода-вывода:

- Mercury\_230\_ART2\_F
- Mercury\_230\_AR

После добавления в дерево проекта модулей ввода-вывода, настраивается панель свойств модулей.

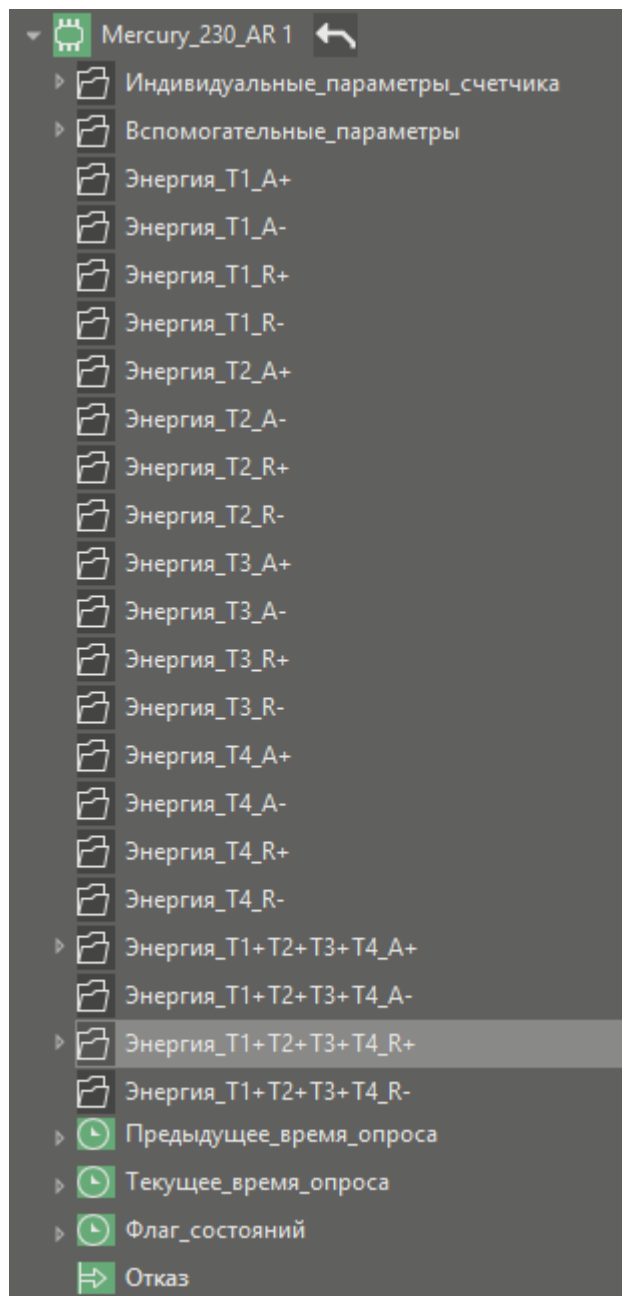
Смотрите также:

Каналы предопределенных типов

#### 6.3.3.2.4.1. Mercury\_230\_AR

Модуль ввода-вывода добавляется в протокол Меркурий 230 при помощи контекстного меню.

Вид в дереве:



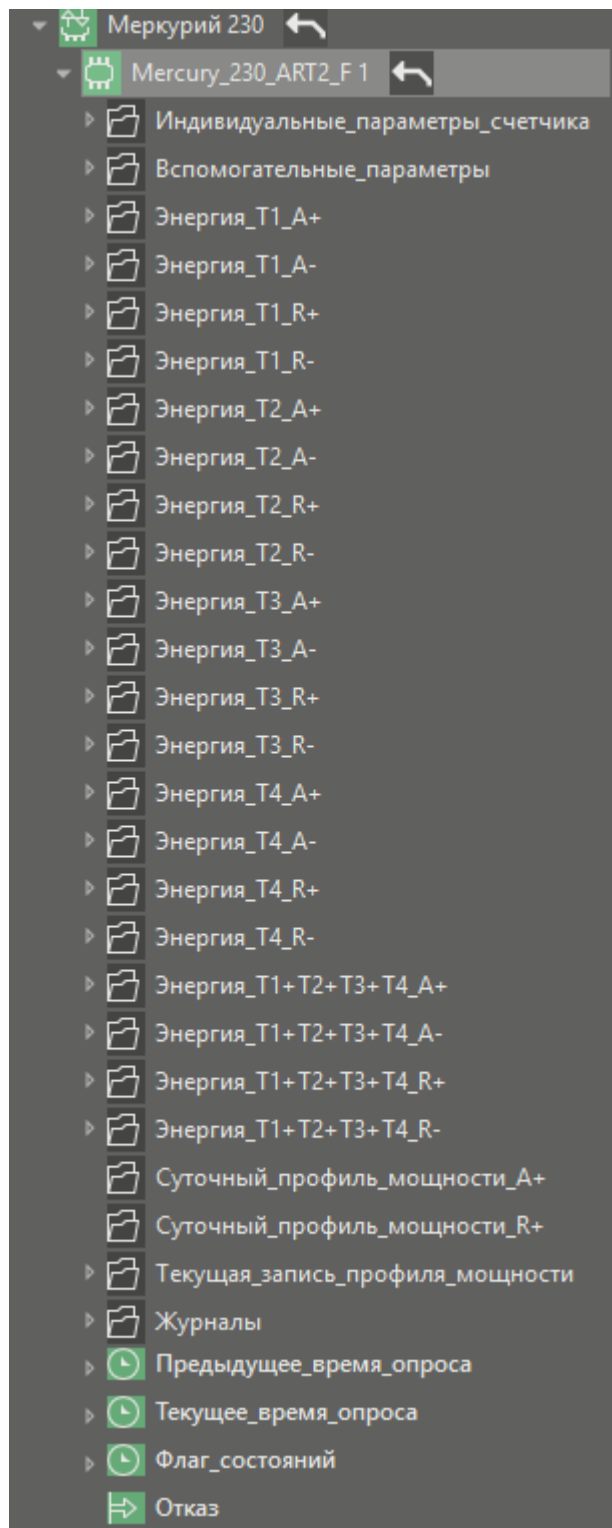
Экземпляр модуля ввода-вывода содержит все необходимые группы и каналы, соответствующие данному типу модуля.

Настройки модуля задаются в его панели свойств.

#### 6.3.3.2.4.2. Mercury\_230\_ART2\_F

Модуль ввода-вывода добавляется в протокол Меркурий 230 при помощи контекстного меню.

Вид в дереве:



Экземпляр модуля ввода-вывода содержит все необходимые группы и каналы, соответствующие данному типу модуля.

Настройки модуля задаются в его панели свойств.

### 6.3.3.2.4.3. Свойства модулей Меркурий 230

Вид панели свойств:

The screenshot shows a software interface for configuring a 'Mercury\_230\_AR 1' module. The interface is dark-themed and features a top toolbar with icons for navigation and settings. Below the toolbar, the configuration is organized into expandable sections:

- Общие (General):**
  - Имя (Name): Mercury\_230\_AR 1
  - Метки (Tags):
  - Комментарий (Comment): Контроллер
- Служебные (Service):** (Currently collapsed)
- Настройки (Settings):**
  - Таймаут (мс) (Timeout (ms)): 1000
  - Адрес (Address): 0
  - Смещение времени (Time Offset): 3
  - Пароль уровня доступа 1 (Access Level 1 Password): 111111
  - Пароль уровня доступа 2 (Access Level 2 Password): 222222
- Масштабирование (Scaling):** (Currently collapsed)

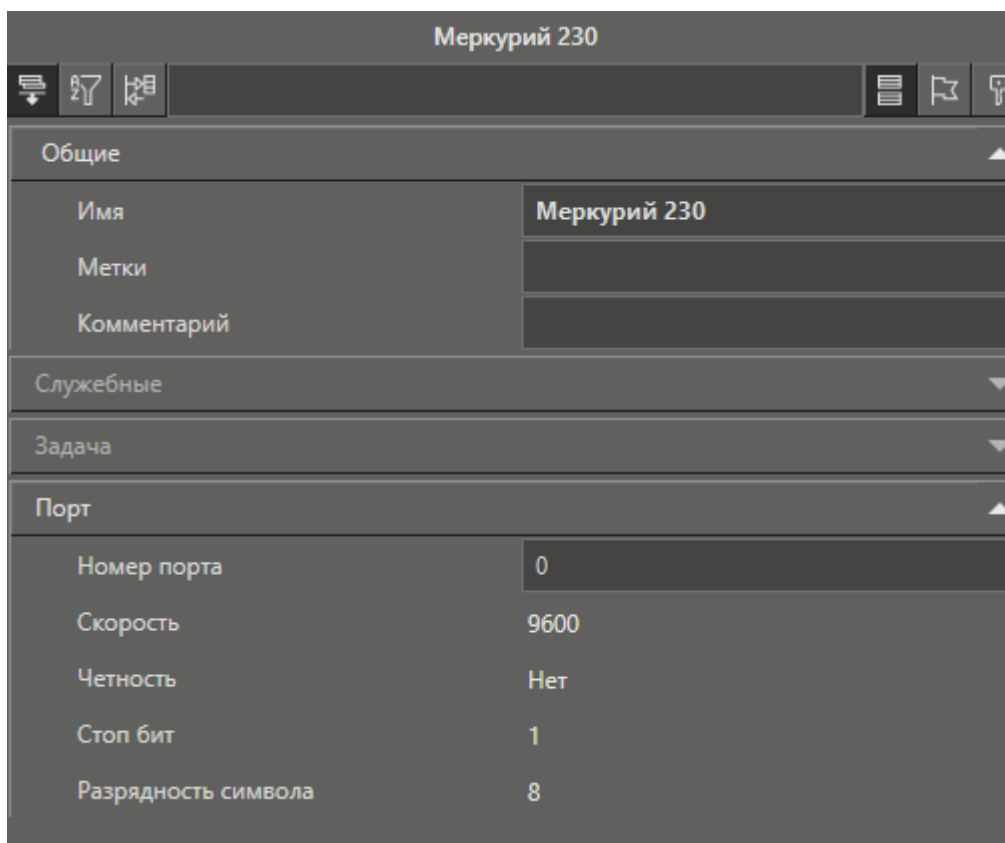
Описание:

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 (значение адреса определяется документацией на подключаемое устройство).
Смещение времени	
Пароль уровня доступа 1	
Пароль уровня доступа 2	
Таймаут	Время ожидания отклика от устройства.

#### 6.3.3.2.4.4. Свойства протокола Меркурий 230

Вид панели свойств:



Описание:

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 будут опрашиваться модули ввода-вывода. Определяется способ работы протокола при резервировании узла. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи у протоколов.
Номер порта	Номер COM-порта, к которому подключаются модули ввода-вывода.
Скорость	Скорость, на которой будет производиться опрос модулей ввода-вывода (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 – 1.
Разрядность символа	Задается число информационных битов (7 или 8), значение по умолчанию – 8.

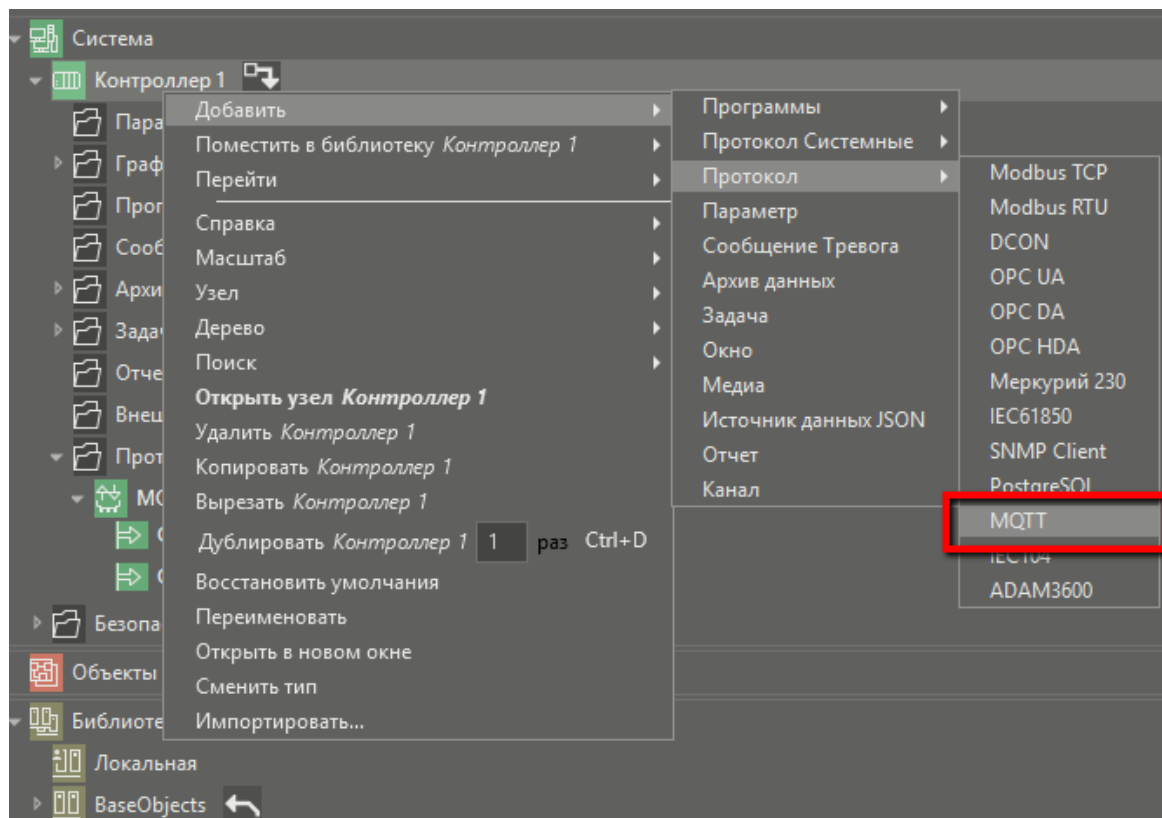
### 6.3.3.2.5. MQTT

Протокол MQTT – Message Queuing Telemetry Transport – протокол для передачи последовательности сообщений с телеметрическими данными, то есть информации от датчиков температуры, влажности, освещенности и др.

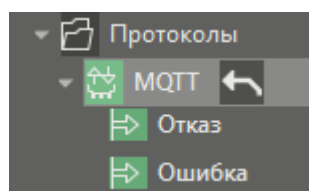
Это платформонезависимый протокол, который может использоваться в исполнительных системах, предназначенных для различных операционных систем.

**Важно!** По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте [scada@insat.ru](mailto:scada@insat.ru)

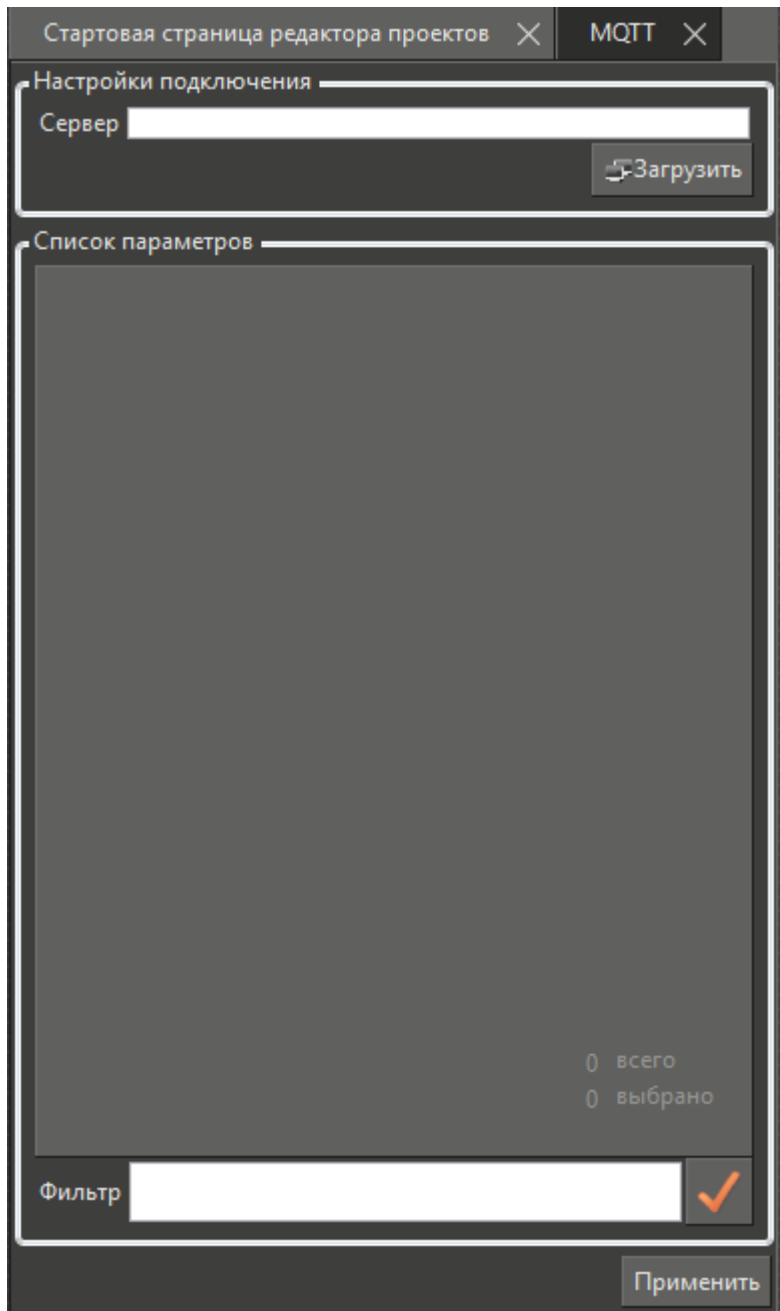
Для добавления протокола в проект необходимо выполнить пункт контекстного меню узла **Добавить.Протокол.MQTT**:



В результате получим:

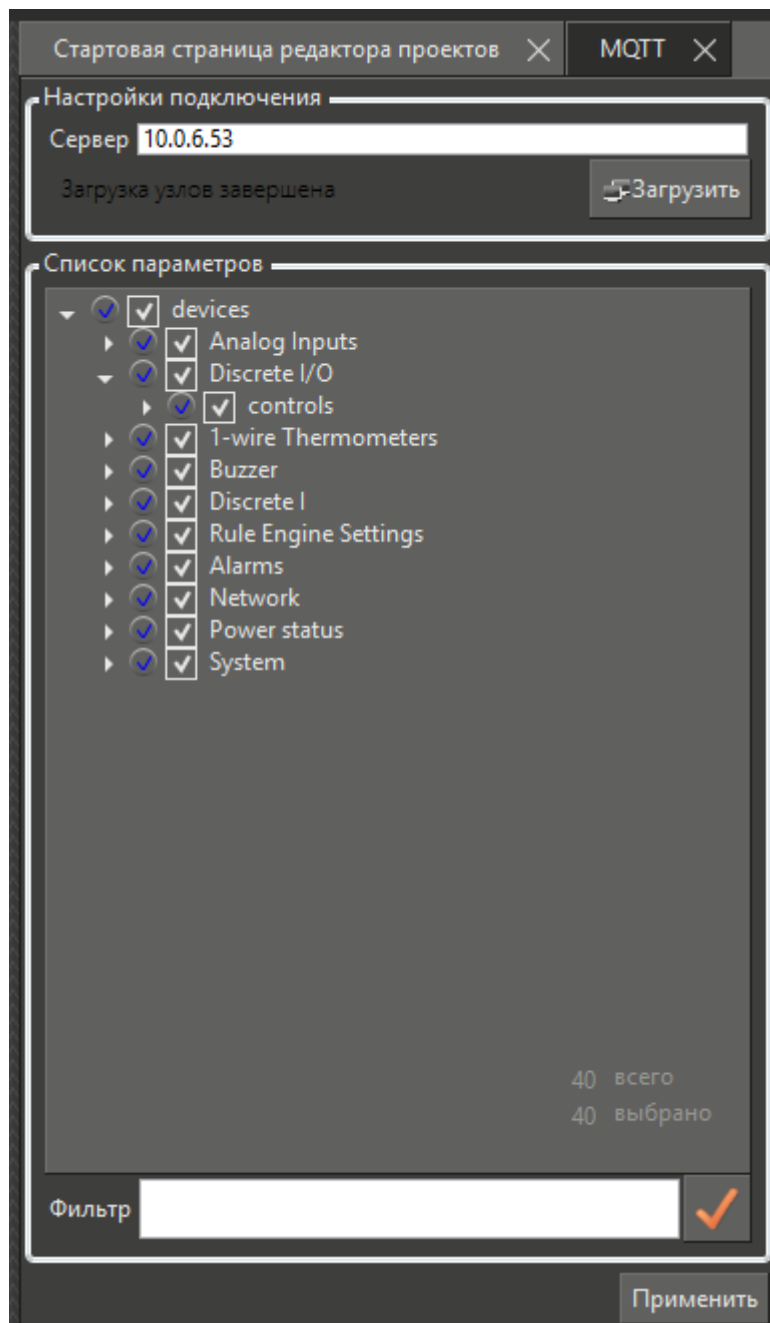


Панель добавления тегов в протокол открывается, если выполнить двойное нажатие левой кнопкой мыши на новый элемент:



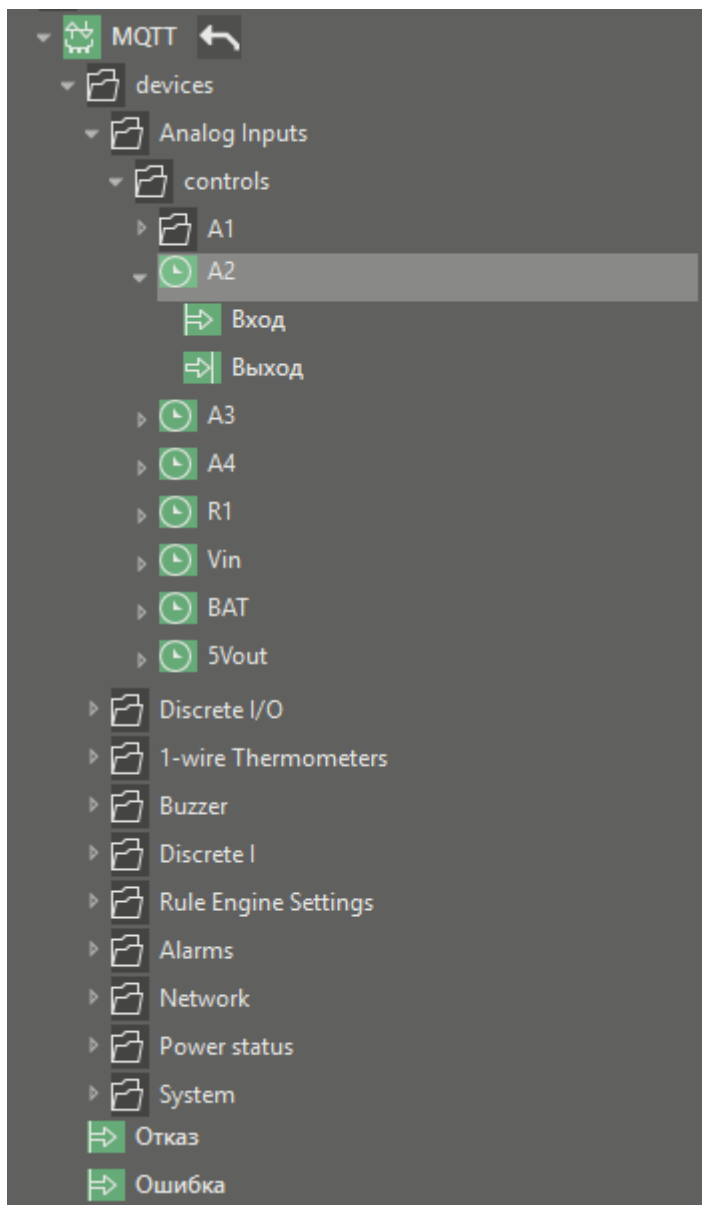
В строке Сервер необходимо указать IP-адрес устройства, на котором установлен MQTT-сервер, а затем нажать кнопку Загрузить. В результате отобразится список доступных параметров, в котором флагами нужно отметить необходимые:





Далее необходимо нажать кнопку Применить.

В результате получим:



Дополнительные настройки протокола задаются в панели свойств.

Смотрите также:

Каналы predetermined типов

### 6.3.3.2.5.1. 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

Торіс:

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 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 при- знак 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 /on.

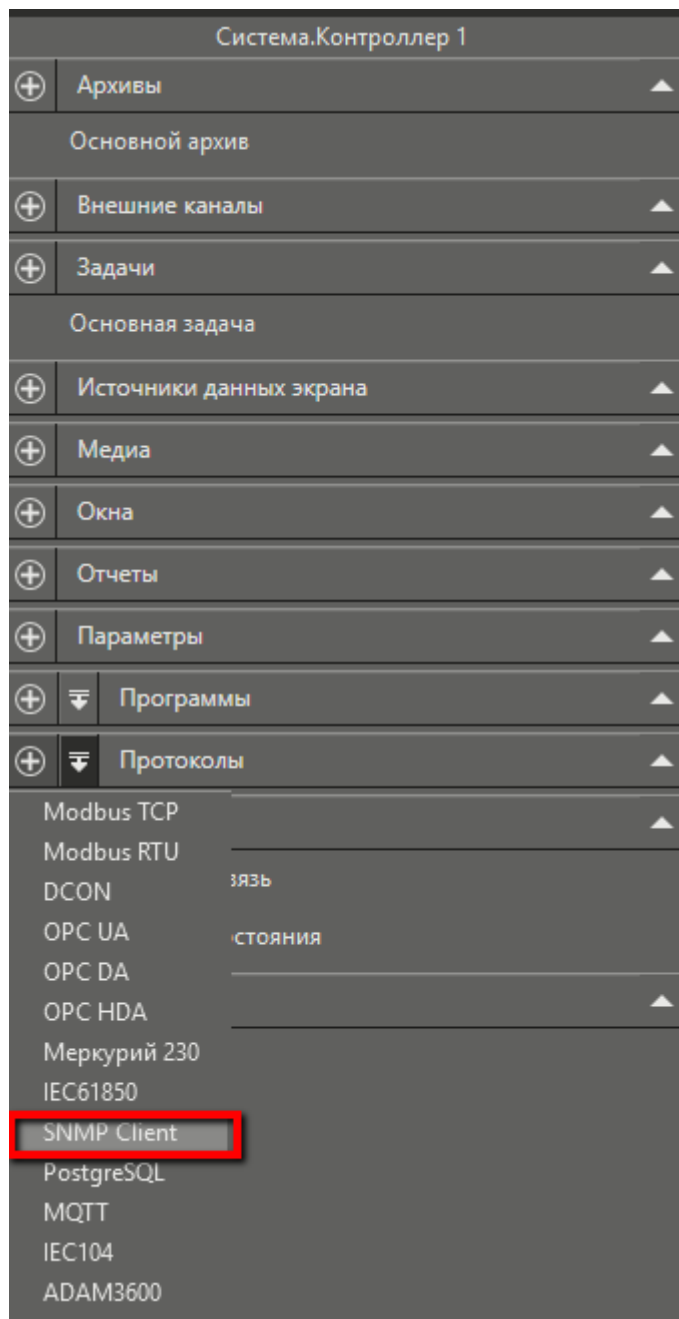
### 6.3.3.2.6. SNMP Client

Simple Network Management Protocol — простой протокол сетевого управления.

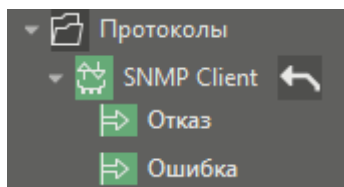
MasterSCADA 4D RT может получать/отправлять данные от устройств и других приложений по протоколу SNMP (версии 1 и 2). Это платформонезависимый протокол, который может использоваться в исполнительных системах, предназначенных для различных операционных систем.

Важно! По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте [scada@insat.ru](mailto:scada@insat.ru)

Если необходимо получать данные по протоколу SNMP, то в узел, при помощи контекстного меню или контекстной панели, добавляется протокол SNMP Client:

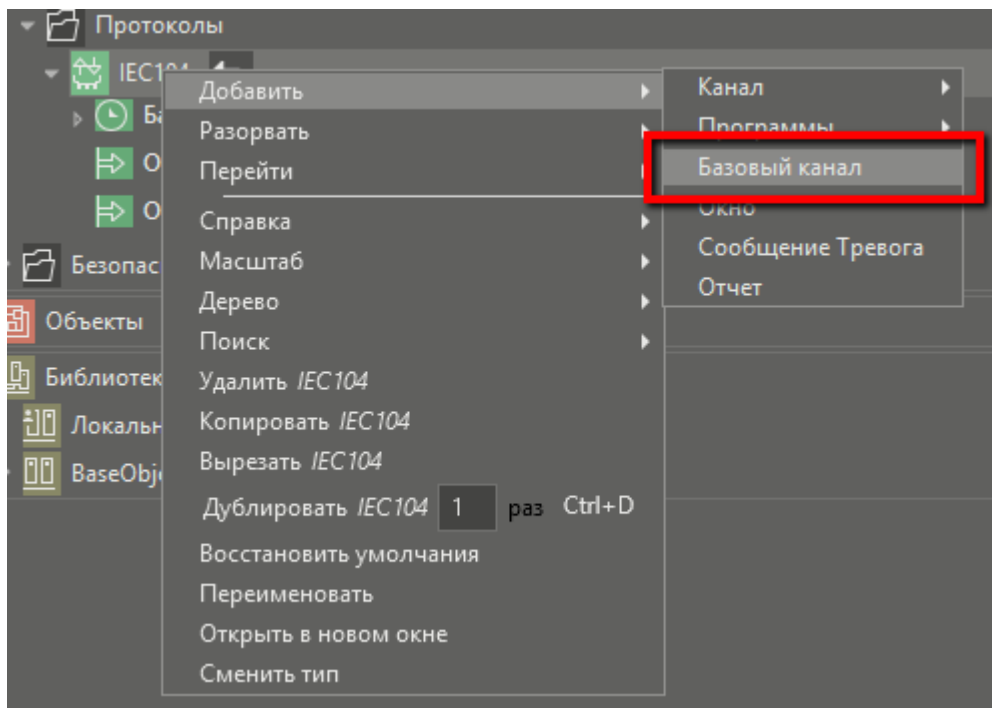


В результате получим:



Затем необходимо настроить панель свойств.

При помощи контекстного меню протокола можно добавить каналы, например, универсальный Базовый канал:



Затем настраивается панель свойств канала.

Смотрите также:

Каналы predetermined типов

### 6.3.3.2.6.1. Свойства SNMP Client

Вид панели свойств протокола SNMP Client:

SNMP Client

☰
☰

**Общие**

Имя	SNMP Client
Программное имя	SNMP Client
Полное имя	Система.APM 1.Протоколы.SNMP Client
Метки	
Комментарий	

Служебные

Отношения

Задача

**Настройки каналов**

oid	
-----	--

**Протокол**

IP адрес	127.0.0.1
Объединение	public
Таймаут ответа	1s
Версия протокола	
Количество попыток опроса	5
Имя безопасности	
Уровень безопасности	noAuthNoPriv
Алгоритм аутентификации	
Пароль аутентификации	
Алгоритм шифрования	
Пароль шифрования	
Имя контекста	

Название	Рекомендации	Версия
----------	--------------	--------

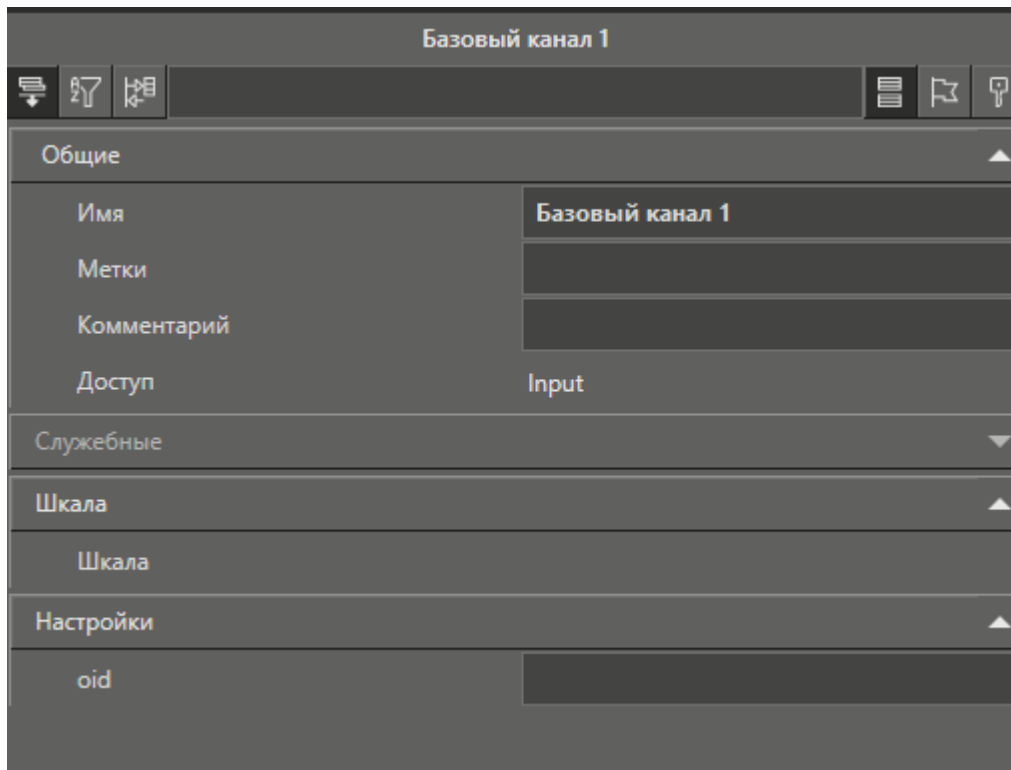


Категория Общие	Как правило, для данного элемента эта категория не настраивается. Подробное описание данной категории приведено в соответствующем разделе.	Все
Категория Задача	Задается период, с которым MasterSCADA 4D будет стараться получать значения из сервера. Определяется способ работы протокола при резервировании узла. Подробное описание категории приведено в разделе, описывающем настройку категории Задачи у протоколов.	Все
Настройки каналов	Задаются значения по умолчанию для создаваемых каналов. В дальнейшем эти настройки у каналов можно будет изменить.	Все
IP-адрес	IP-адрес сервера, который отправляет и получает данные от MasterSCADA 4D.	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 исчерпания заданного числа попыток, то выдается отказ протокола.	Все
Имя безопасности	Задается имя пользователя, используемое при подключении к устройству.	v3
Уровень безопасности	Задается уровень безопасности работы с устройством. Существует три варианта: <ul style="list-style-type: none"> <li>oNoAuthNoPriv - без аутентификации и без шифрования. Используется только имя пользователя.</li> </ul>	v3

	<ul style="list-style-type: none"> <li>• oauthNoPriv - с аутентификацией, но без шифрования. Для аутентификации нужен пароль, но обмен данными не шифруется.</li> <li>• oauthPriv - с аутентификацией и с шифрованием.</li> </ul>	
Алгоритм аутентификации	<p>Задается алгоритм хэширования пароля аутентификации. Варианты:</p> <ul style="list-style-type: none"> <li>• MD5</li> <li>• SHA</li> </ul>	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"> <li>• DES</li> <li>• AES128</li> </ul>	v3
Пароль шифрования	Устанавливается пароль шифрования передаваемых данных.	v3
Имя контекста	Задается набор доступной информации управления по объекту SNMP.	v3

### 6.3.3.2.6.2. 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

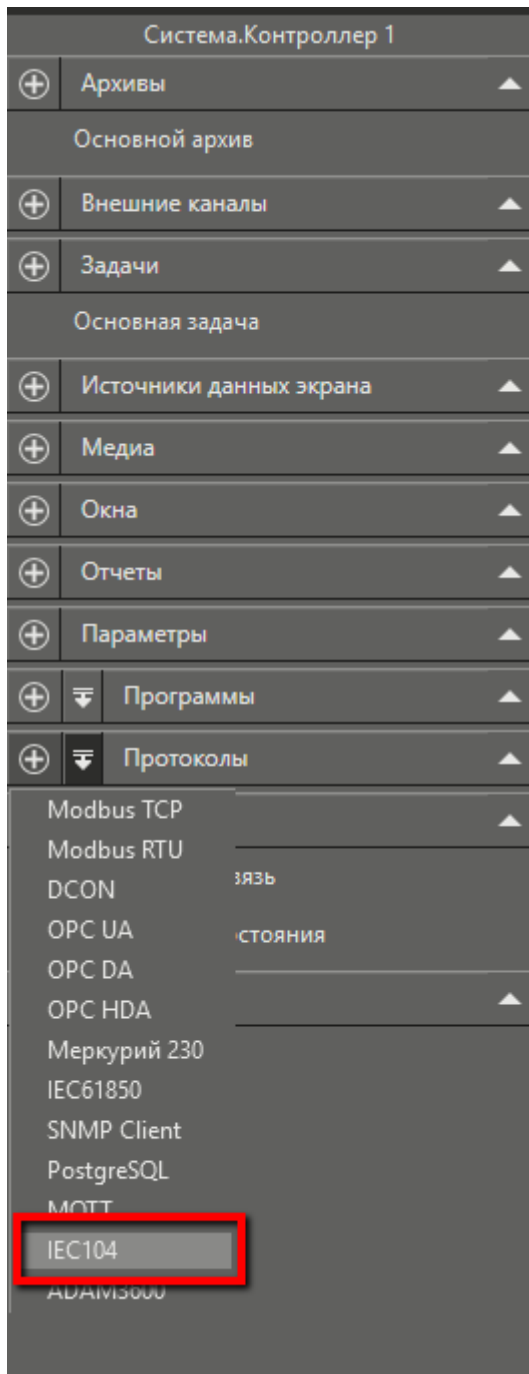
**Важно!** В текущей версии автоматической загрузки каналов из mib-файлов не предусмотрено. Идентификаторы (адреса) задаются вручную.

### 6.3.3.2.7. IEC104

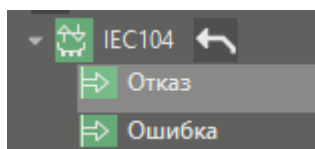
MasterSCADA 4D RT может получать/отправлять данные от устройств и других приложений по протоколу IEC 60870-5-104 (МЭК 60870-5-104). Это платформонезависимый протокол, который может использоваться в исполнительных системах, предназначенных для различных операционных систем.

**Важно!** По умолчанию этот протокол не входит в состав среды исполнения. Стоимость и условия лицензирования необходимо уточнять по электронной почте [scada@insat.ru](mailto:scada@insat.ru)

Если необходимо получать данные по протоколу IEC 60870-5-104, то в узел, при помощи контекстного меню или контекстной панели, добавляется протокол IEC104:

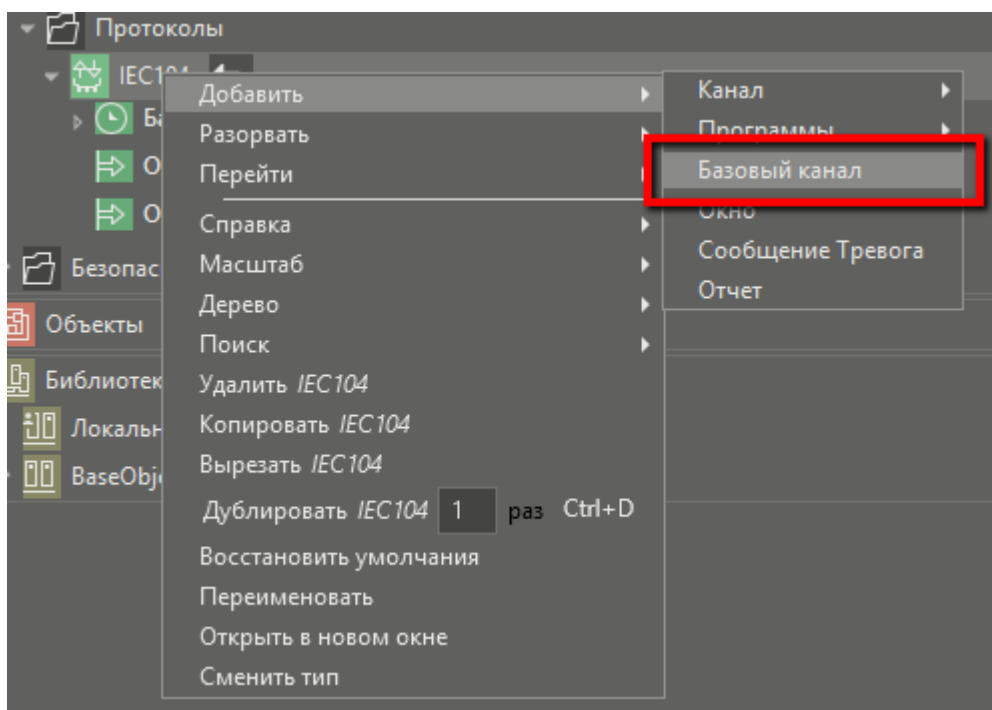


В результате получим:



Затем необходимо настроить панель свойств.

При помощи контекстного меню протокола можно добавить каналы, например, универсальный Базовый канал:



Затем настраивается панель свойств канала.

### 6.3.3.2.7.1. Свойства IEC104

Вид панели свойств протокола IEC104:

IEC104

Общие

Имя: IEC104

Метки:

Комментарий:

Служебные

Задача

Период, мс: 100

Приоритет: 100

Подключение по условию:

Выполнение по условию:

Выполнять на резервном:

Настройки каналов

IOA address: 0

CA address: -1

Command type: Короткое число с плавающей запятой

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 опроса: 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.
Настройки каналов	Задаются значения по умолчанию для создаваемых каналов. В дальнейшем эти настройки у каналов можно будет изменить.
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 СА	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. 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.
---	--

APDU - Application Protocol Data Unit (Протокольный Блок Данных Прикладного уровня).

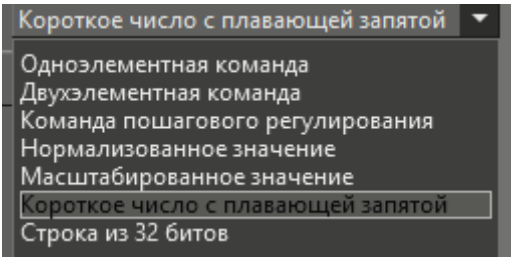
ASDU - Application Service Data Unit (Блок Данных Прикладного уровня).

### 6.3.3.2.7.2. Свойства канала IEC104

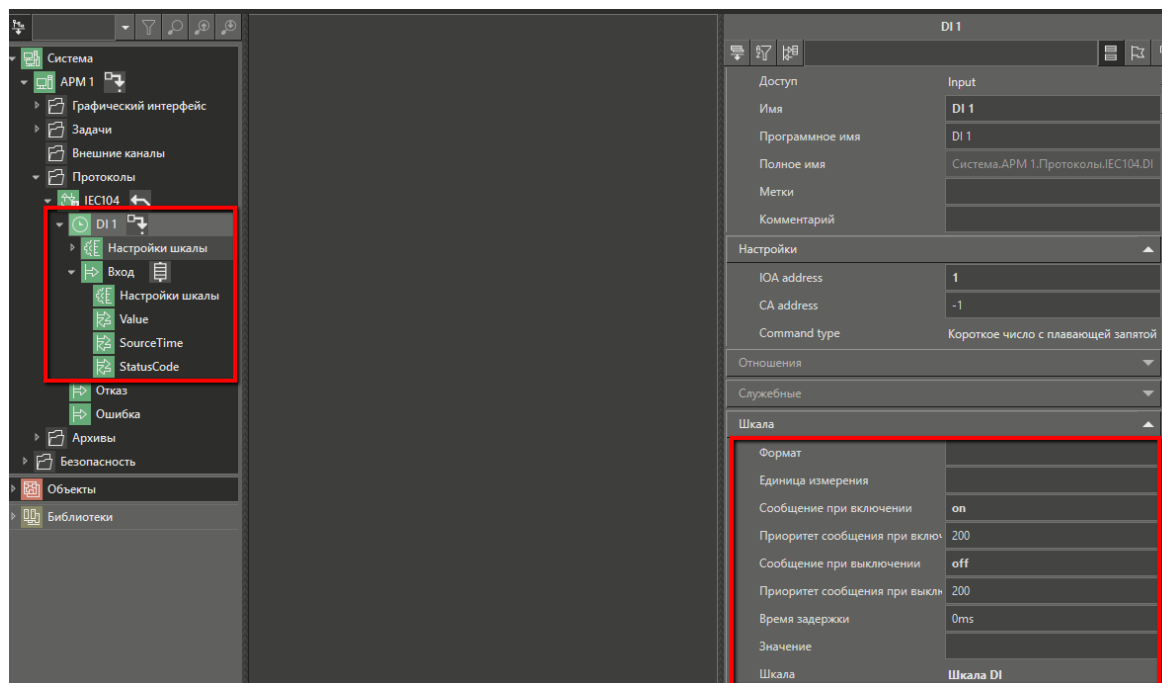
Вид панели свойств:

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.

	<p>Подробное описание данной категории приведено в соответствующем разделе</p>
Шкала	<p>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.</p>
IOA address	<p>Information object address - адрес переменной.</p>
CA address	<p>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.</p>
Command type	<p>Тип команды, настраивается для выходных каналов. Согласно требованиям протокола поддерживаются следующие типы команд:</p>  <p>манд:</p>

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.



### 6.3.3.2.8. Протоколы для работы с базами данных

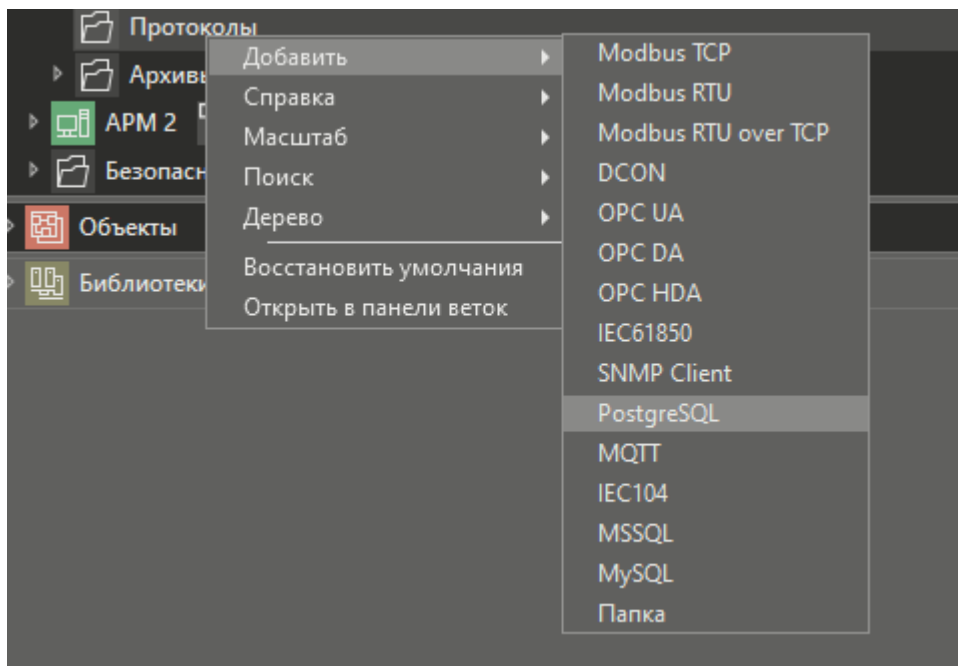
В данном разделе описаны дополнительные протоколы, которые необходимы для чтения и записи значений параметров в произвольные базы данных.

Данные протоколы не связаны с функцией архивирования.

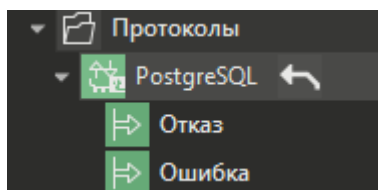
#### 6.3.3.2.8.1. Протокол PostgreSQL

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:

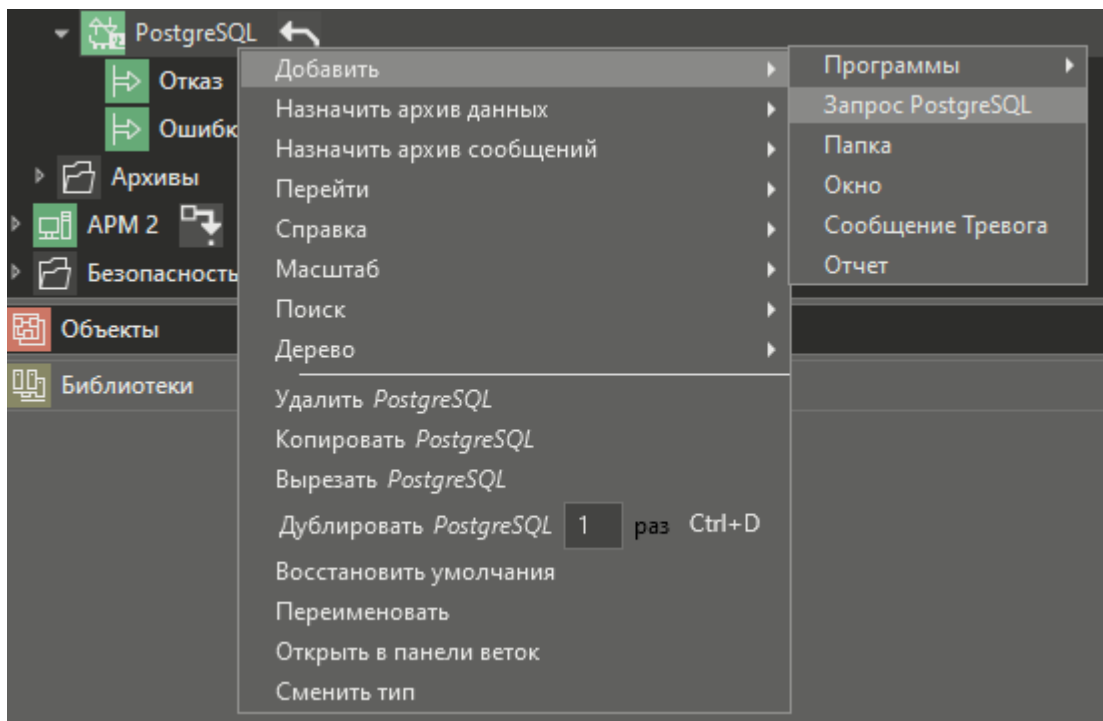


Получим результат:



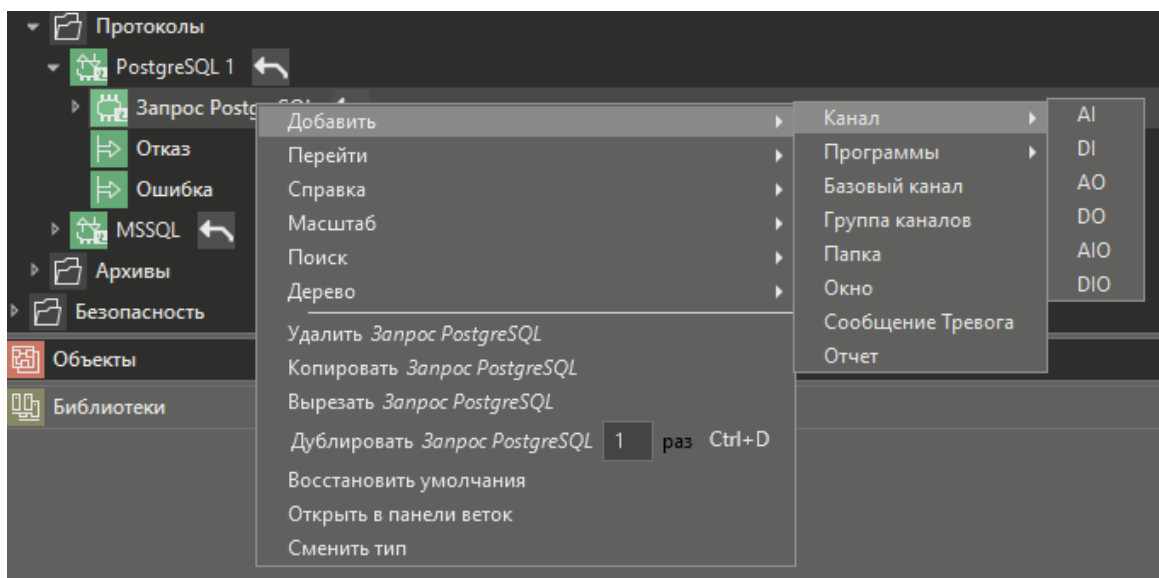
Далее необходимо настроить панель свойств протокола.

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.

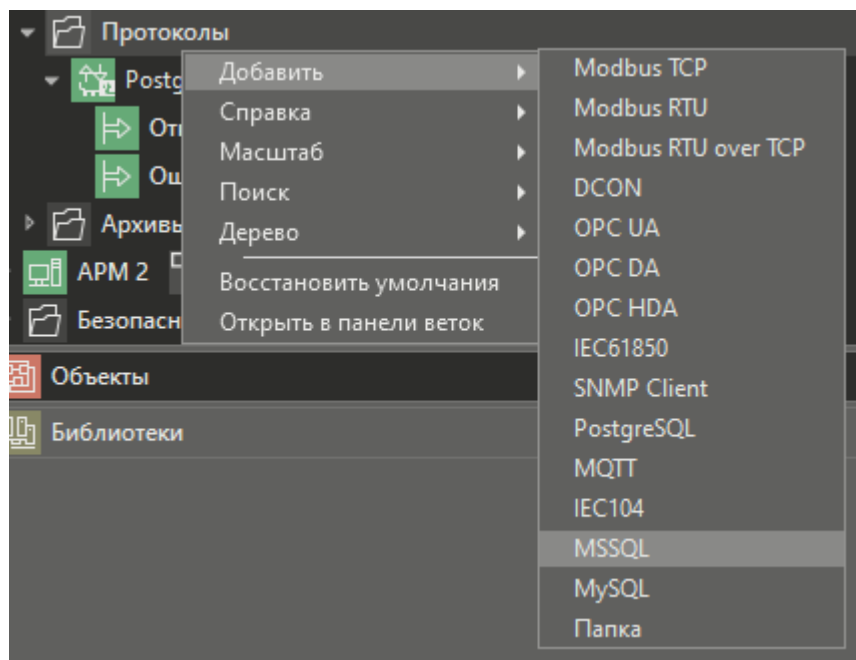
**Важно!** Имена каналов должны совпадать с именами переменные присутствуют в базе данных, к которой идет обращение

Затем необходимо задать настройки в панели свойств Запрос PostgreSQL.

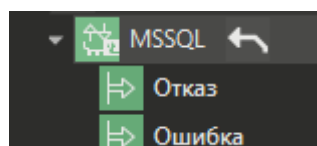
### 6.3.3.2.8.2. Протокол MSSQL

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:

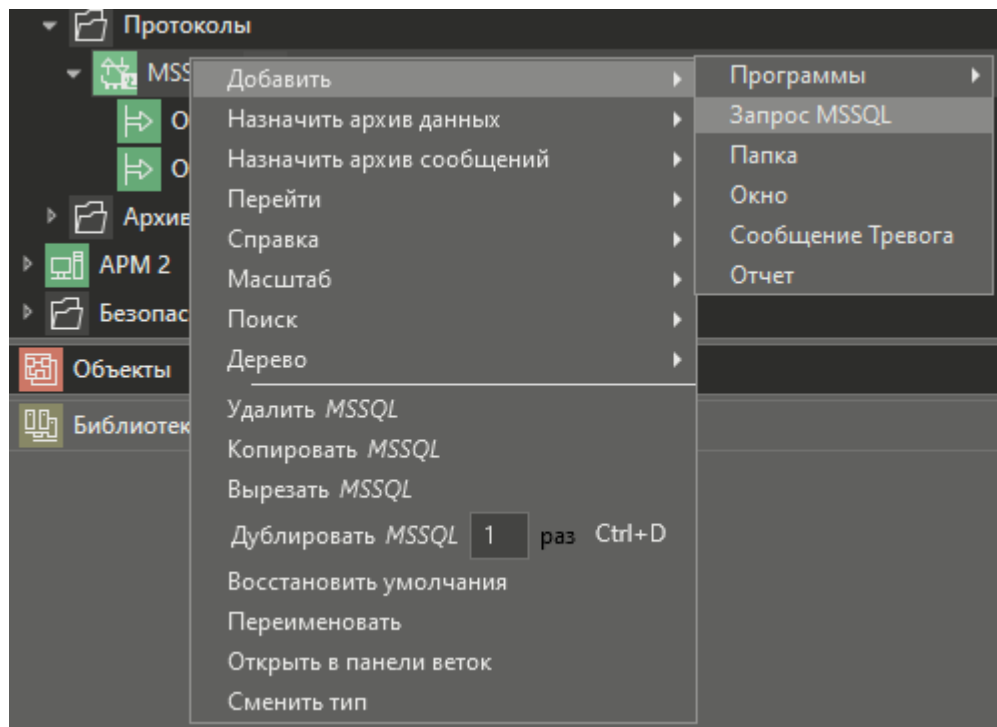


Получим результат:



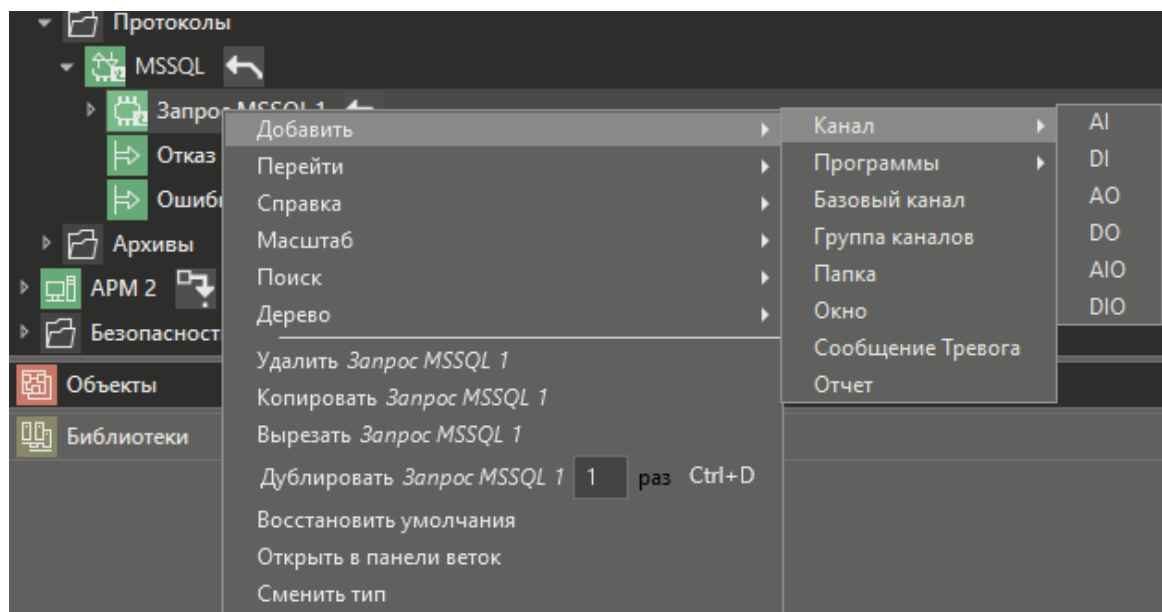
Далее необходимо настроить панель свойств протокола.

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

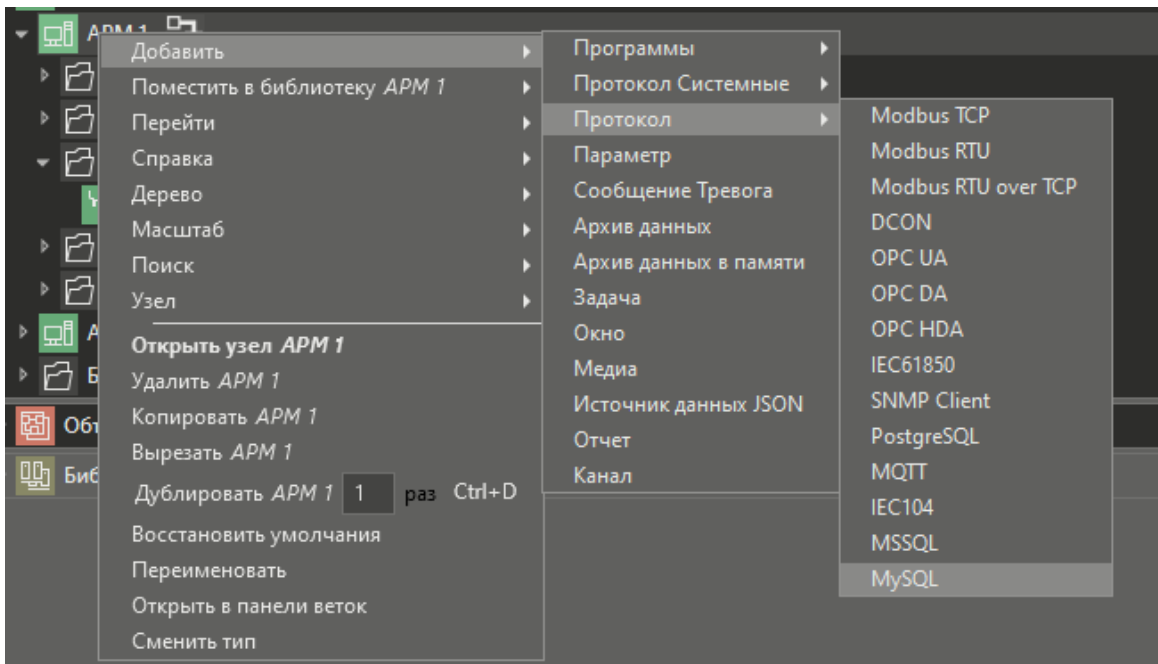
Затем необходимо задать настройки в панели свойств Запрос MSSQL.

Настройки, которые необходимо сделать непосредственно в базе данных описаны в разделе Настройки базы данных MSSQL

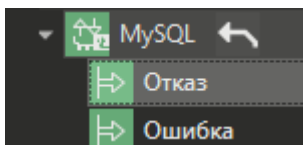
### 6.3.3.2.8.3. Протокол MySQL

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:



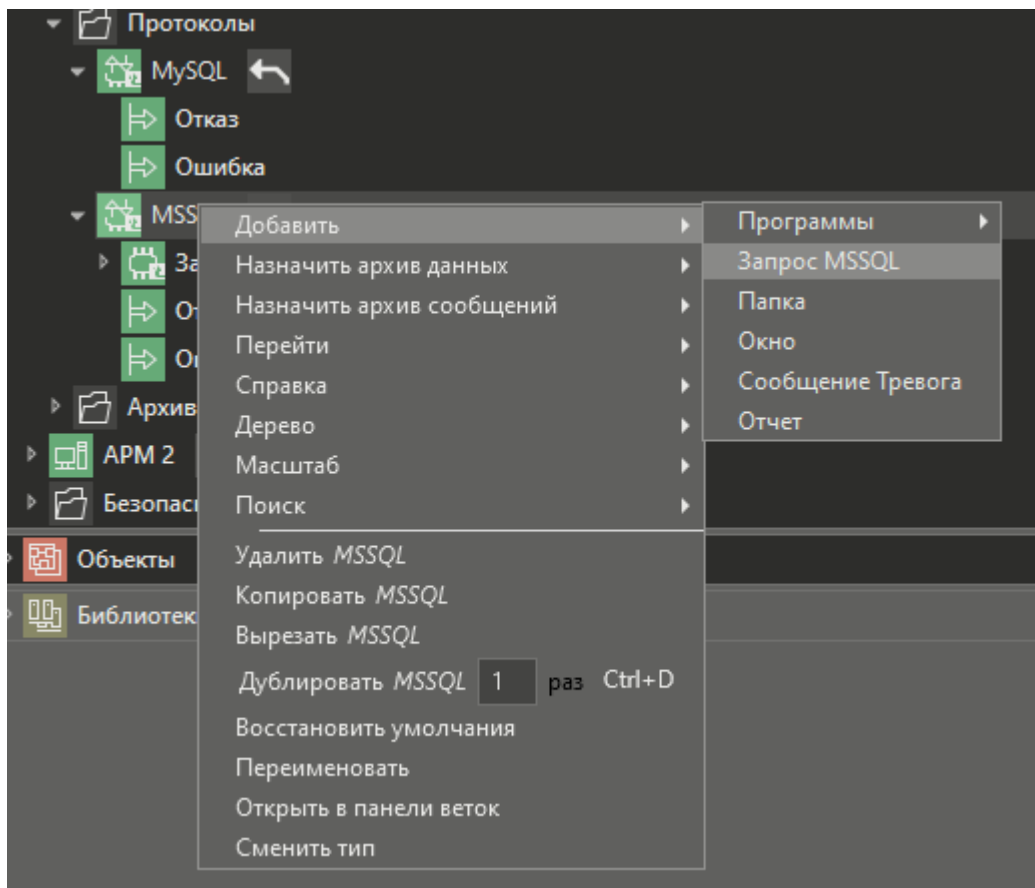
Получим результат:



Далее необходимо настроить панель свойств протокола.

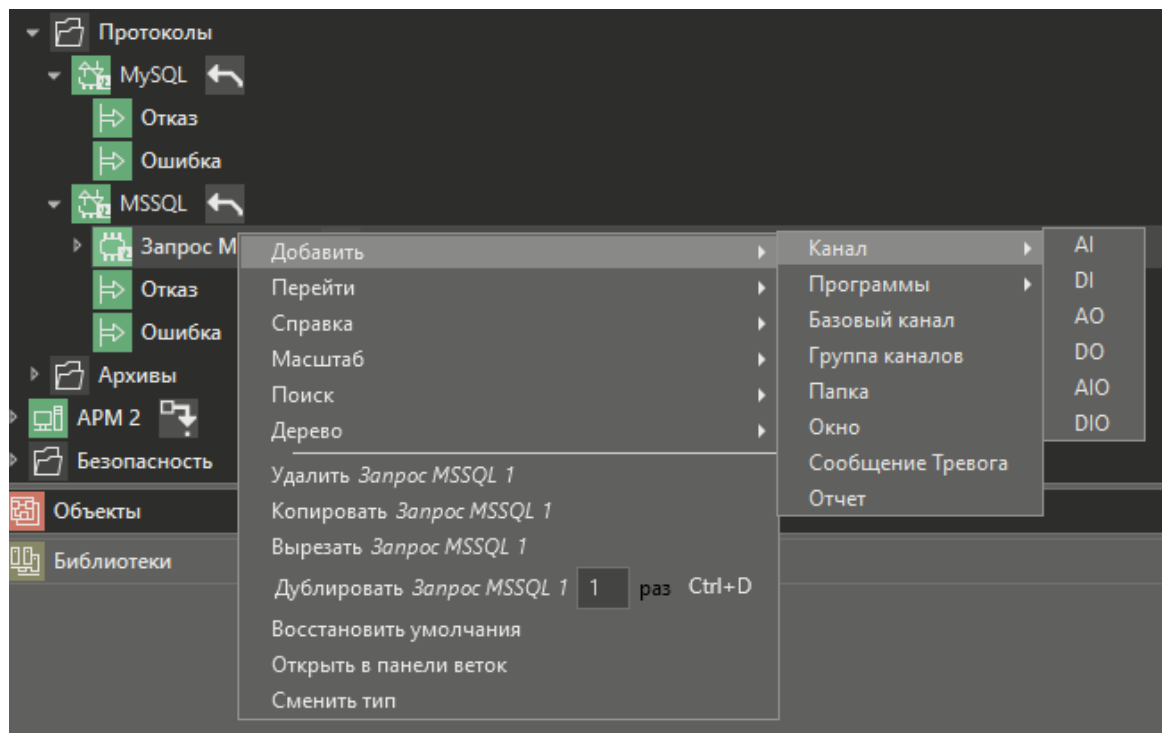
Затем при помощи контекстного меню протокола можно добавить Запрос MySQL:





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.

#### 6.3.3.2.8.4. 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	<p>Задается период, с которым будет выполняться подключение к базе данных. Если используется свойство Подключение по условию, то 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.

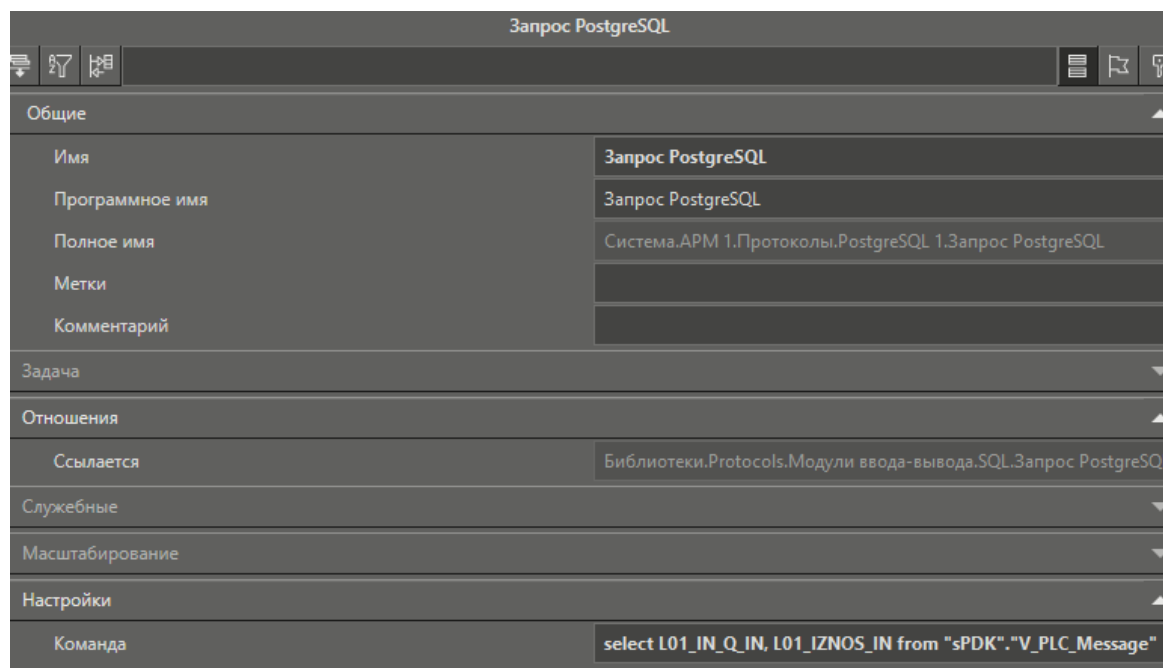
Пользователь	Имя пользователя для подключения к базе данных. Настраивается при конфигурировании базы данных.
Пароль	Пароль для подключения к базе данных. Настраивается при конфигурировании базы данных.
База	Имя базы данных, к которой необходимо обращаться.
Порт	Номер TCP порта, через который идет обмен 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

### 6.3.3.2.8.5. 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 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 "TABLE-NAME"(DATA) VALUES(@Channelname). <b>Важно!</b> 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}`

{SCH} Данные из каналов напрямую вставляются в тело запроса в момент отправки на сервер.

Плюсы:

- Можно динамизировать запросы, т.е. писать запросы вида: `select {SCH1} from {SCH2} where {SCH3} = {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

### 6.3.3.3. MasterSCADA 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

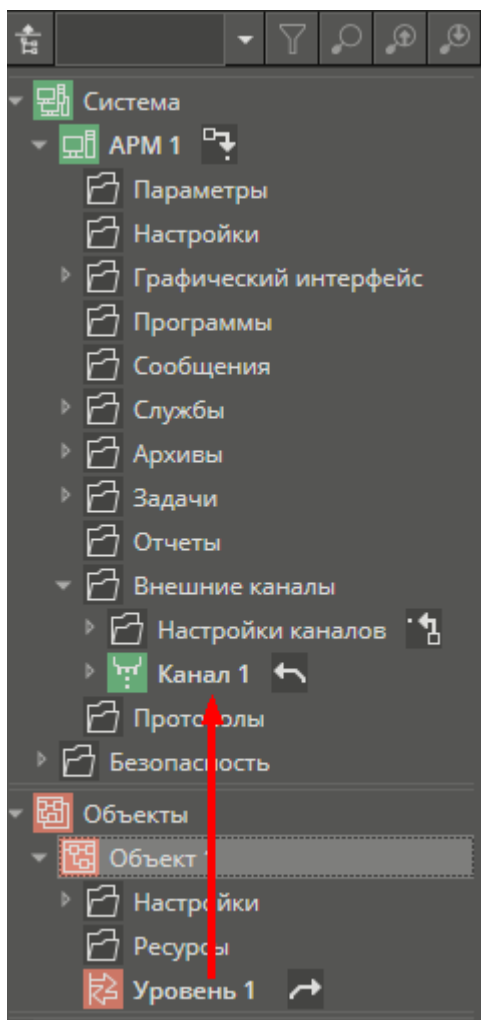
#### 6.3.3.3.1. Внешние каналы

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:

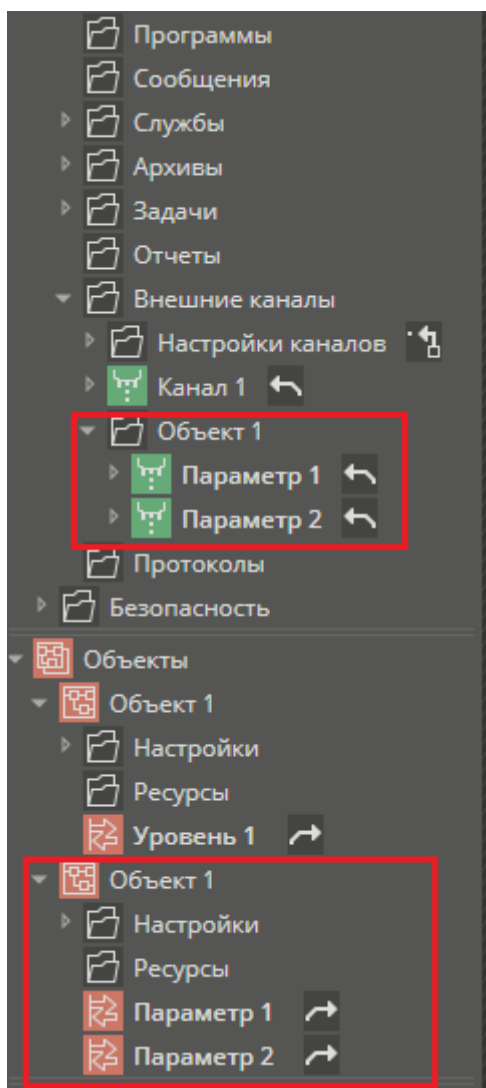


#### Автоматическое создание внешних каналов

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"> <li>• Чтение - через OPC/Modbus Slave/ IEC104 Slave можно только читать.</li> <li>• Чтение/Запись - через OPC/Modbus Slave/ IEC104 Slave можно читать и писать.</li> <li>• Запись - через OPC/Modbus Slave/IEC104 Slave можно только писать.</li> </ul>
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 автоматически при формировании карты переменных Modbus Slave.</p>

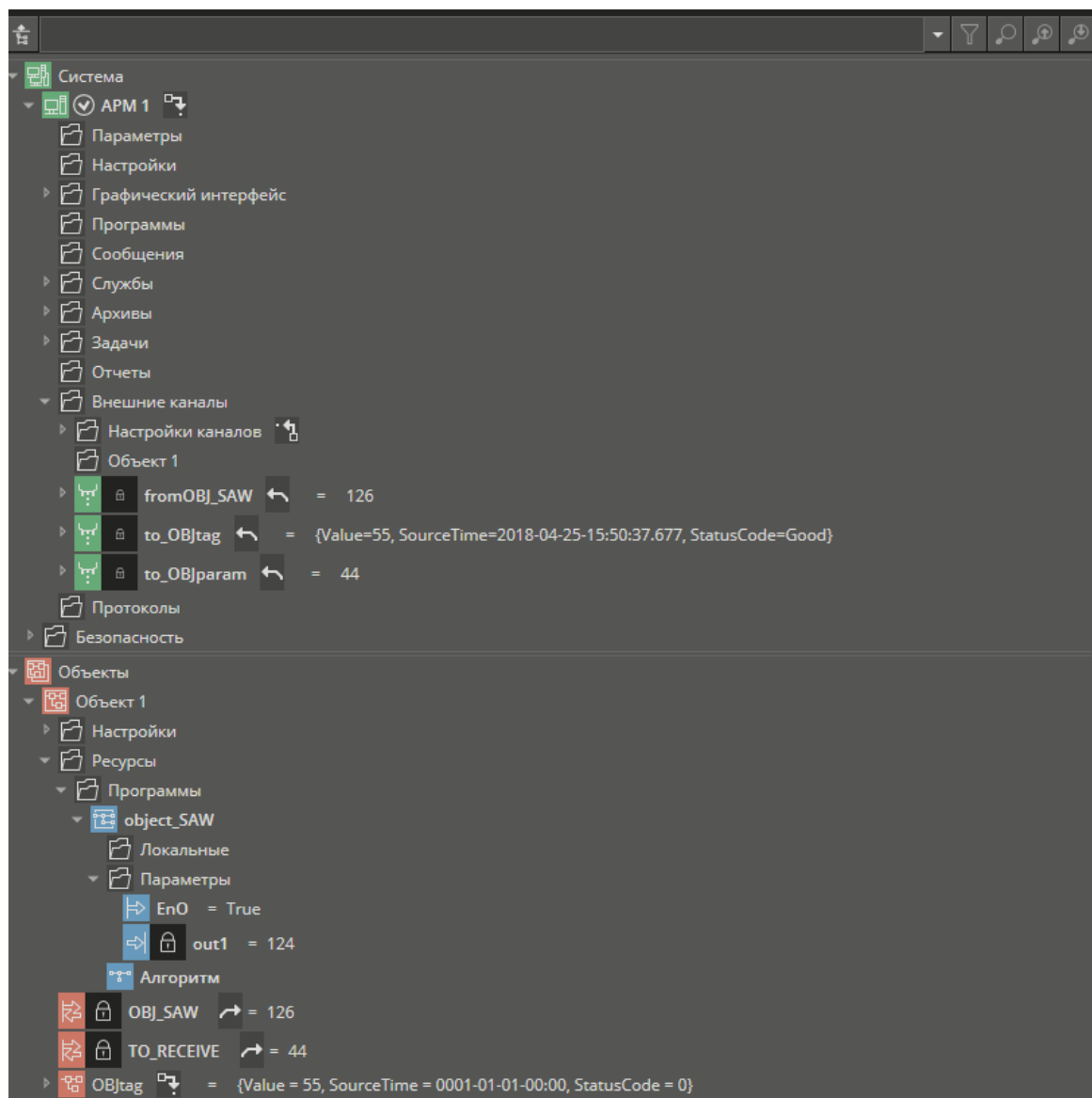
Адрес IEC104	Задается адрес канала для доступа по IEC104 Slave.
--------------	--

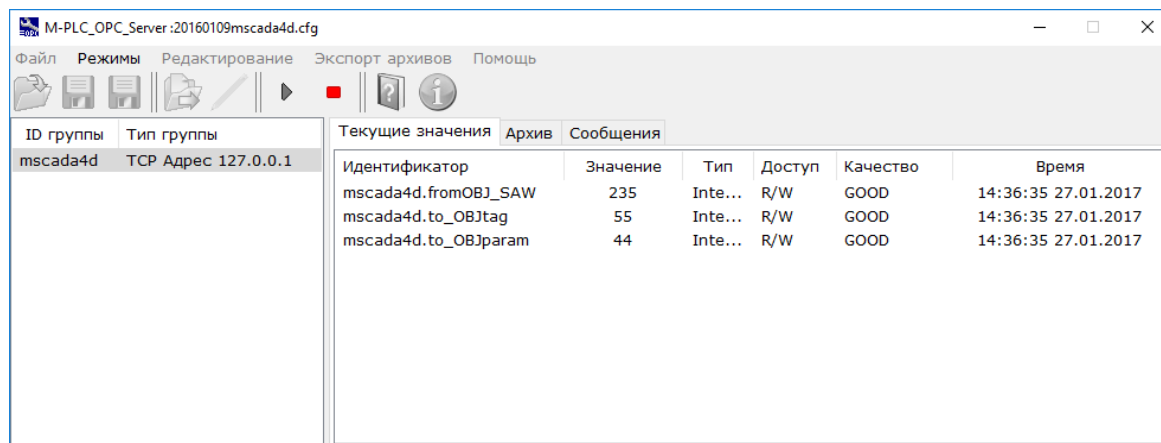
Смотрите также:

- M-PLC-OPC-server
- Modbus Slave
- IEC104 SLave

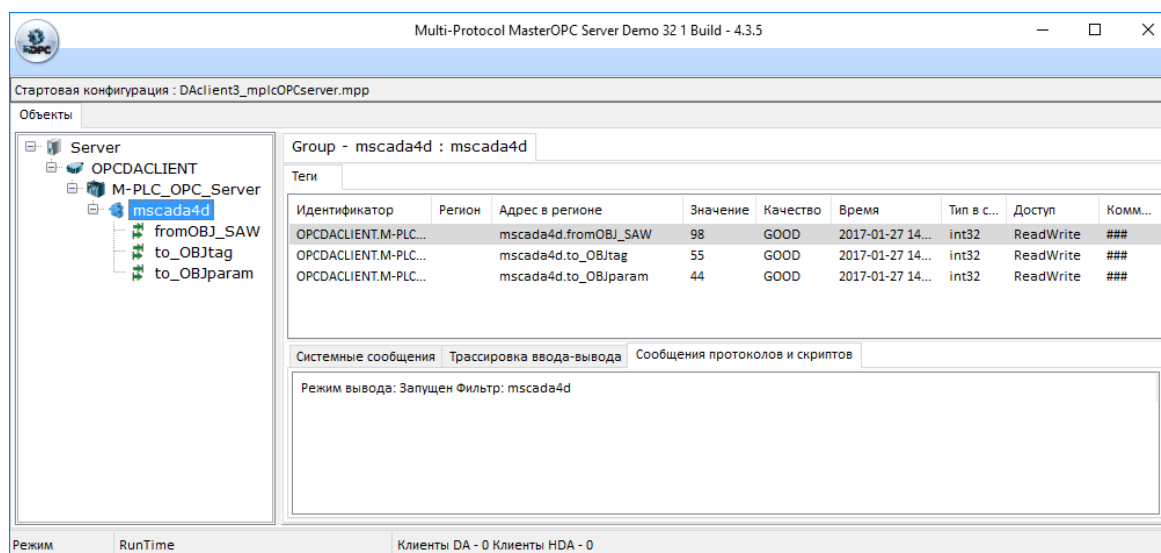
### 6.3.3.3.1.1. MasterSCADA 4D как сервер OPC DA

Каналы, созданные в группе Внешние каналы, доступны для сервера M-PLC OPC Server.



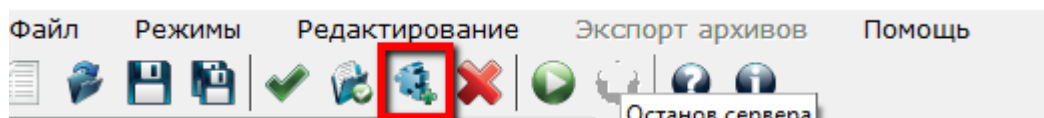


В паре с M-PLC OPC Server, MasterSCADA 4D является сервером OPC DA, и переменные проекта MasterSCADA 4D доступны клиенту OPC DA:

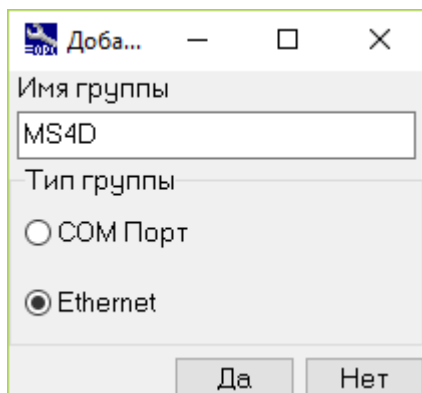


### Конфигурирование M-PLC OPC Server

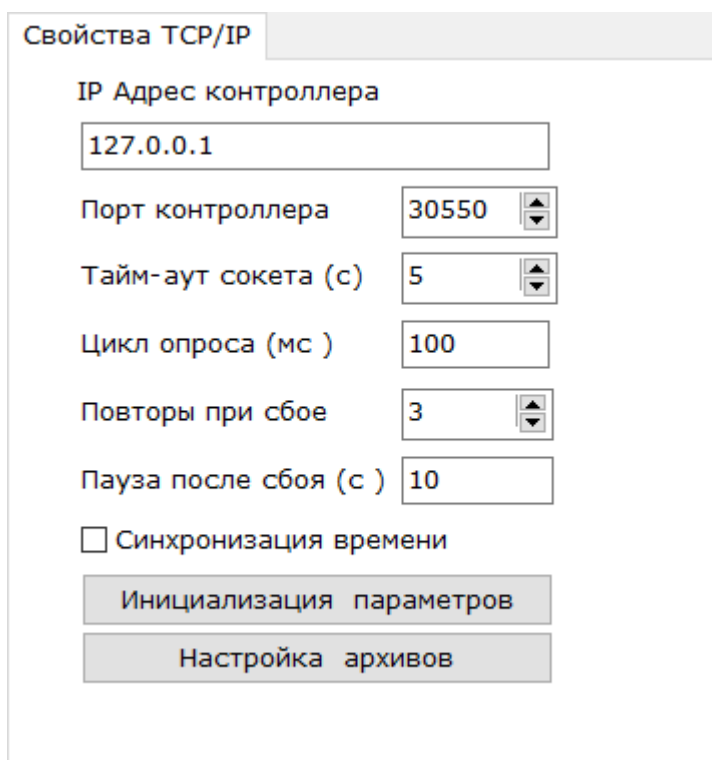
Для конфигурирования сервера необходимо нажать иконку в панели инструментов Добавить группу:



Затем указать имя группы, а также выбрать канал, по которому сервер будет подключаться к среде исполнения. Как правило это Ethernet:

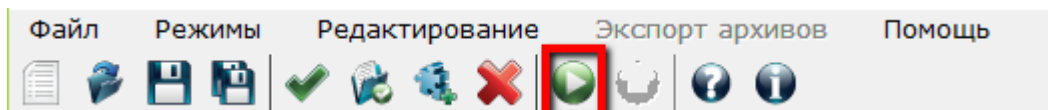


Далее требуется задать настройки подключения:



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.

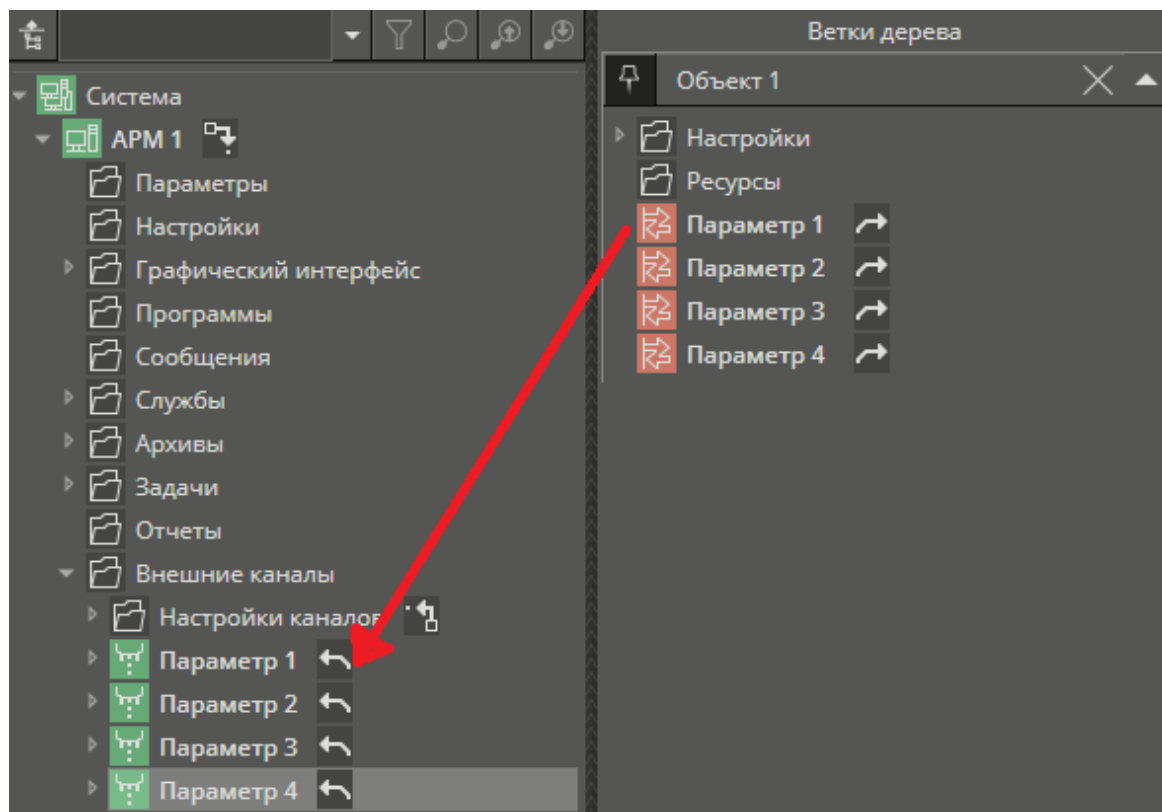
Для запуска сервера в режим исполнения нужно нажать иконку Старт сервера:



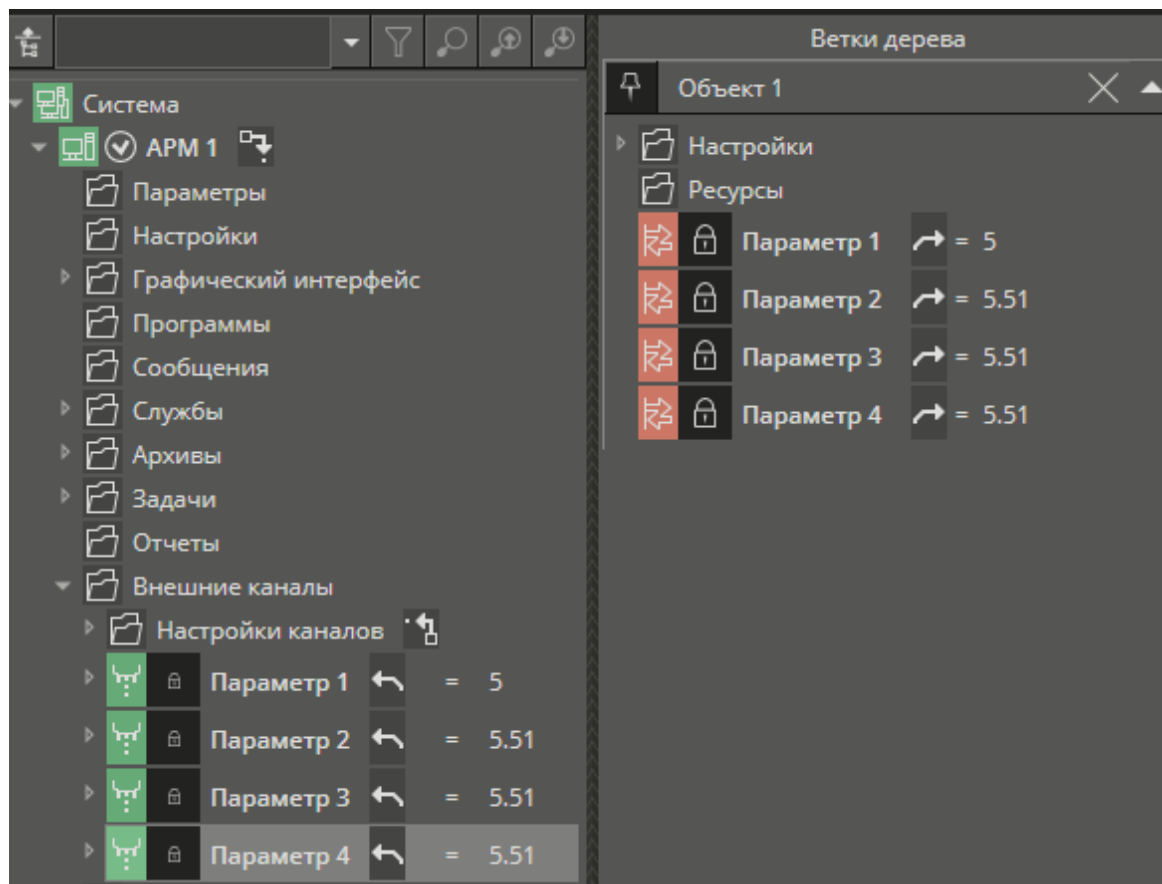
### Запись параметра объекта в архив MasterSCADA 3

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:

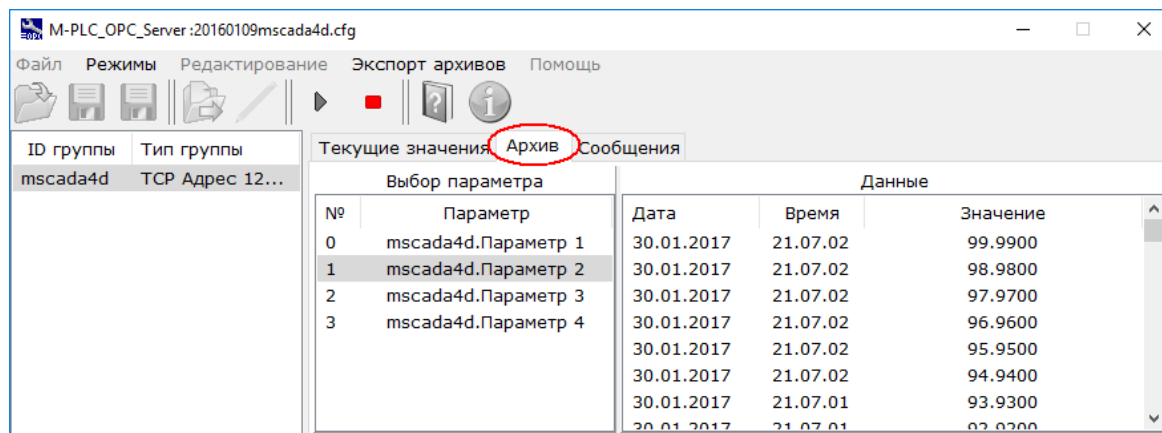


- подключить сервер M-PLC OPC Server, инициализировать параметры, и в настройке архивов для архива 0 задать Такт=1:

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\InSAT\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.

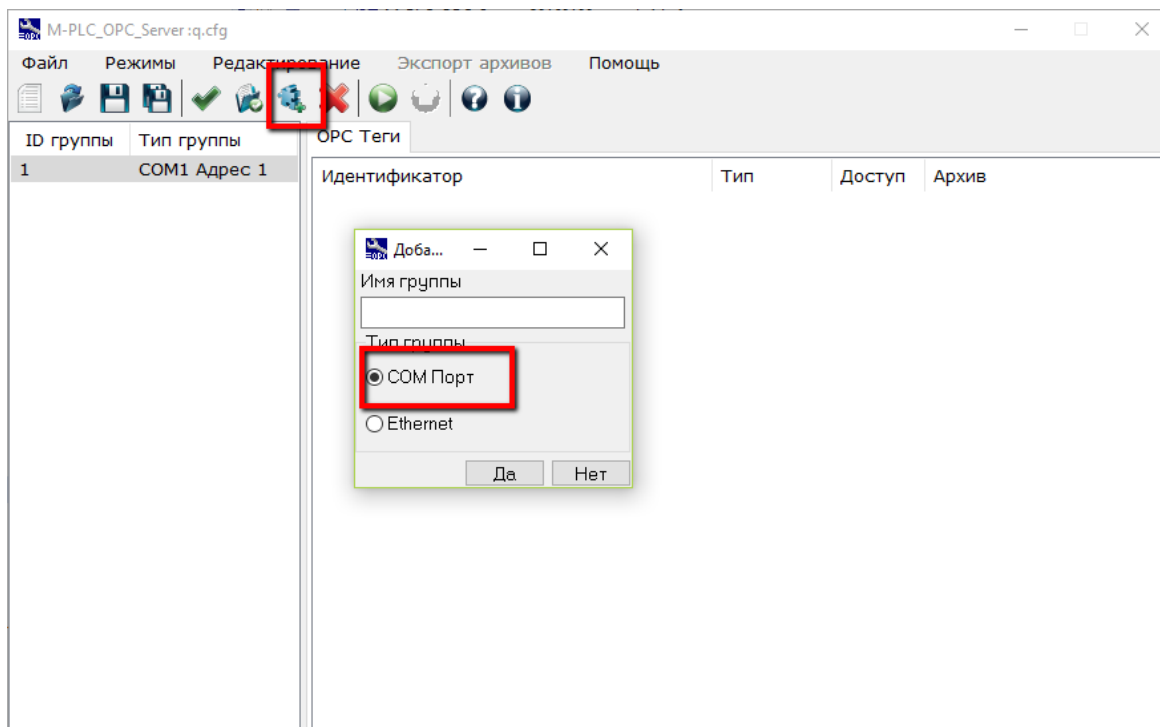
При этом используется следующая строка инициализации модема, подключенного к устройству, на котором работает среда исполнения:  
ATE0Q0V1&C0&D2&S1S0=0S7=60+CBST=71,0,1

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.

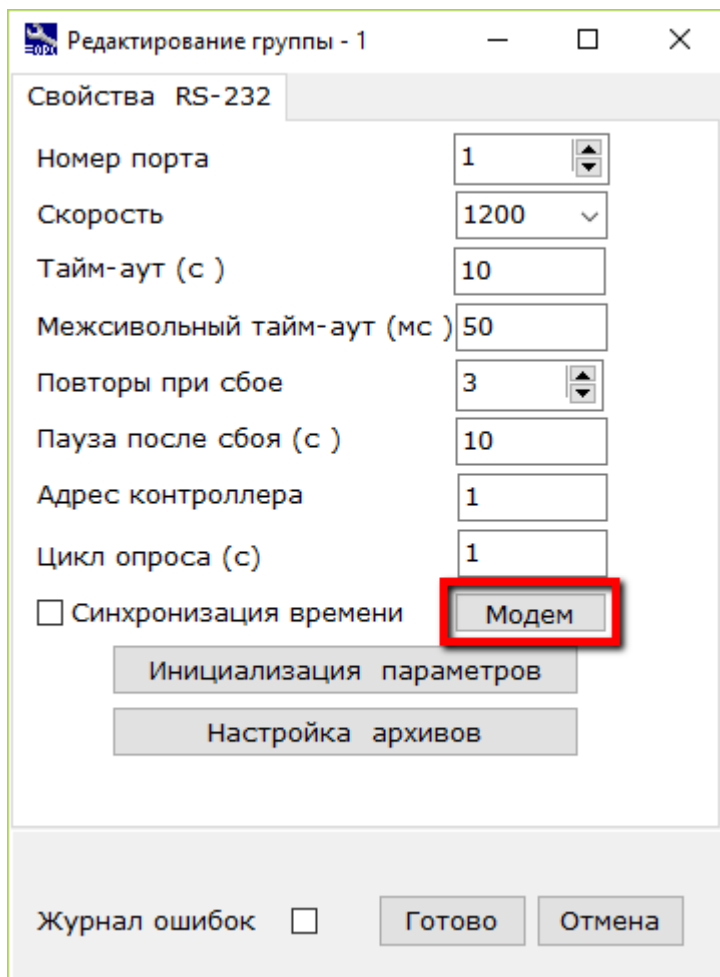
В настройках M-PLC OPC Server необходимо сделать следующее.

Добавить группу, в настройках которой выбрать COM Порт.

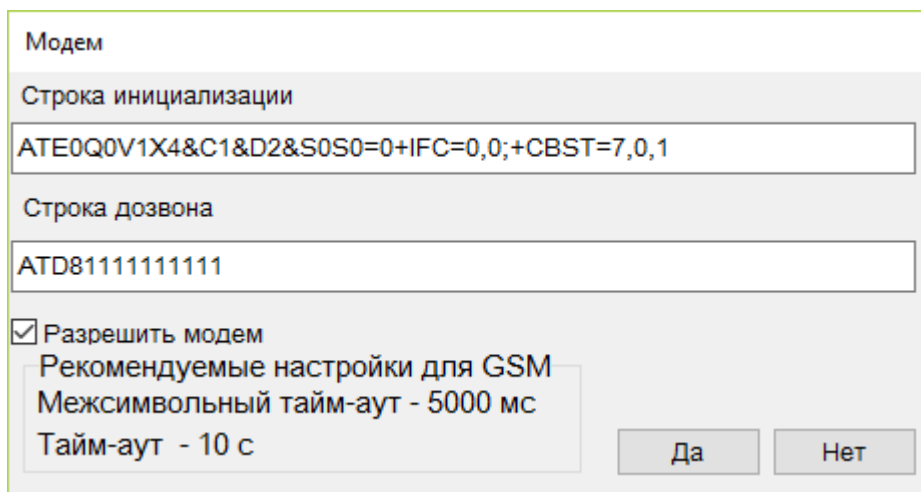


В настройках COM-порта, к которому физически подключен gsm-модем, задать его настройки. Значения выбираются исходя из возможностей подключаемого оборудования. Для настройки специальных параметров модема следует нажать кнопку Модем:





В открывшемся диалоговом окне необходимо установить флаг Разрешить модем, указать телефон, на который требуется дозваниваться, а также строку инициализации.



По окончании настройки следует провести инициализацию параметров.

### 6.3.3.3.1.2. MasterSCADA 4D в роли Modbus Slave

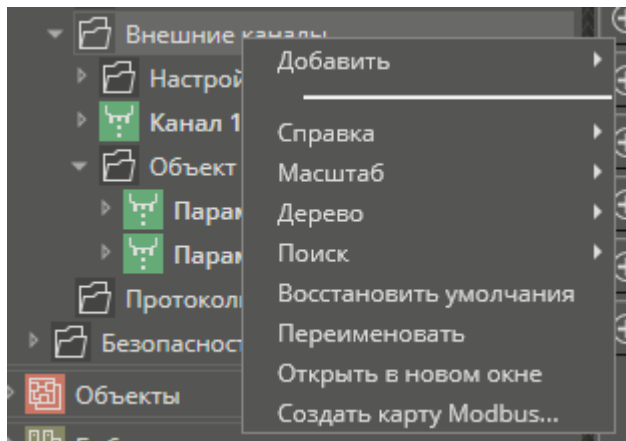
Внешние каналы, наряду с доступом через стандарт OPC, могут быть также доступны другим системам по протоколам Modbus TCP, Modbus RTU. В этом случае исполнительная система MasterSCADA 4D выступает в роли Slave, а другое приложение - в роли Master. В качестве Master'a может выступать и другое устройство, 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	
<b>Общие</b>	
Доступ	Чтение/Запись
Имя	Канал 1
Метки	
Комментарий	
<b>Служебные</b>	
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
<b>Настройки</b>	
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(he	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.

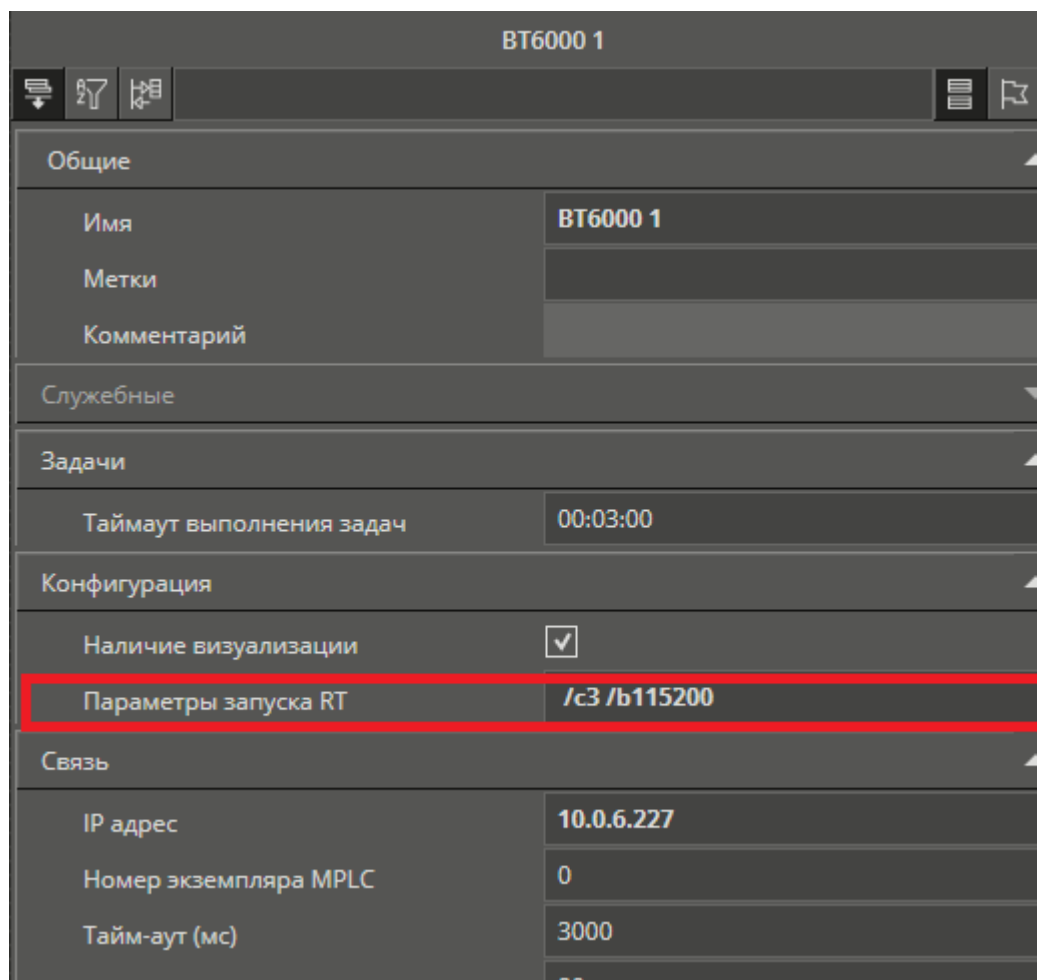
Никаких дополнительных настроек в среде разработки не требуется.

В клиенте достаточно указать IP-адрес узла, в котором будет работать проект Master-SCADA 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:



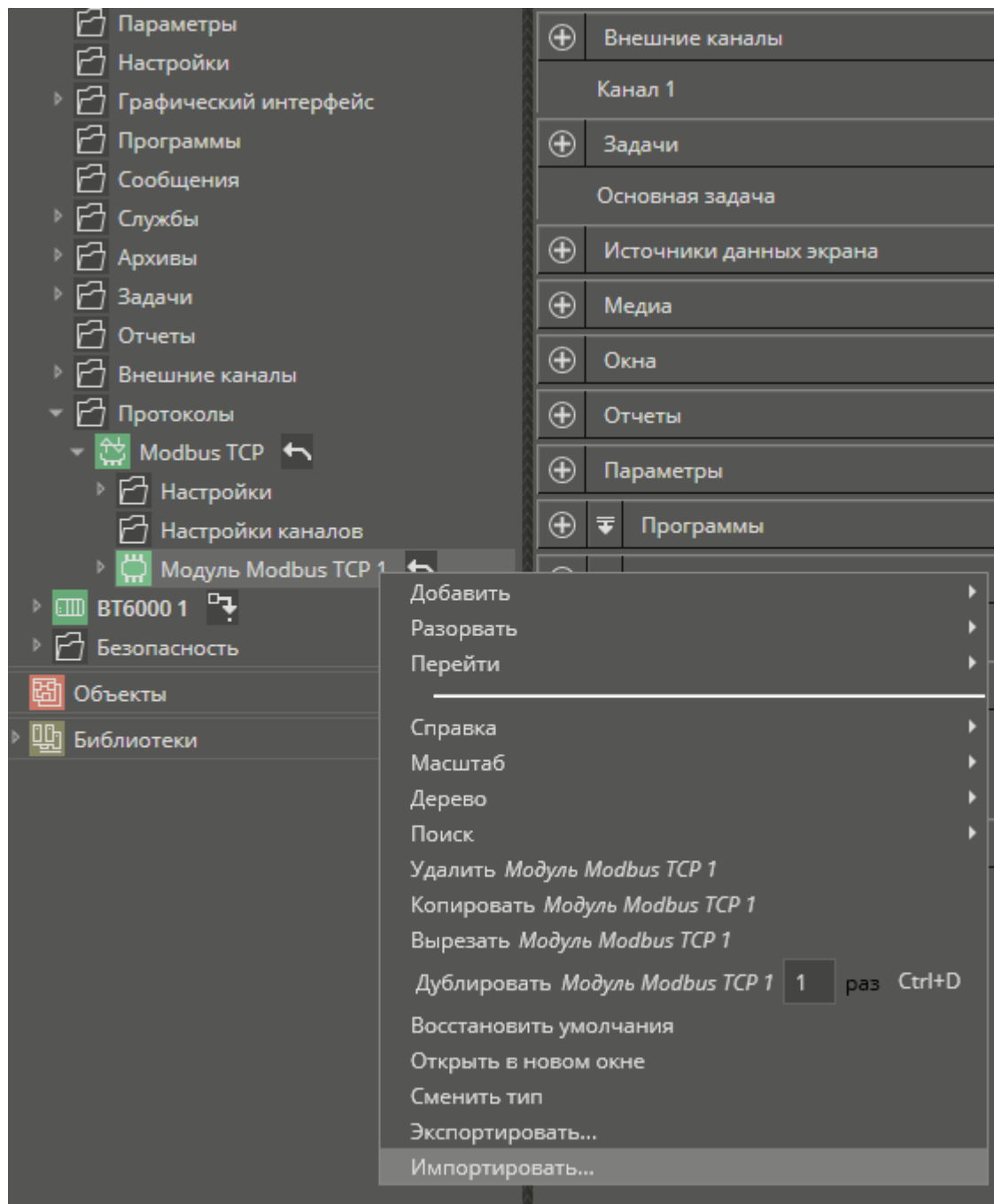
Настройка имеет вид /cN /bS, где N - номер порта, а S - скорость.

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, служащем поставщиком данных.

### 6.3.3.3.1.3. 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:

Настройка	Значение
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 Modbus	20
Время неактивности подключения Modbus	10s
<b>Доступ по IEC104</b>	<input checked="" 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	<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	
<b>Общие</b>	
Доступ	Чтение/Запись
Имя	Канал 1
Метки	
Комментарий	
<b>Служебные</b>	
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
<b>Настройки</b>	
Тип значения	Вещественный
Адрес	0
Адрес IEC104	0

Подключение по IEC104 SLave

Вышеуказанных настроек достаточно для подключения клиентов по IEC 104.

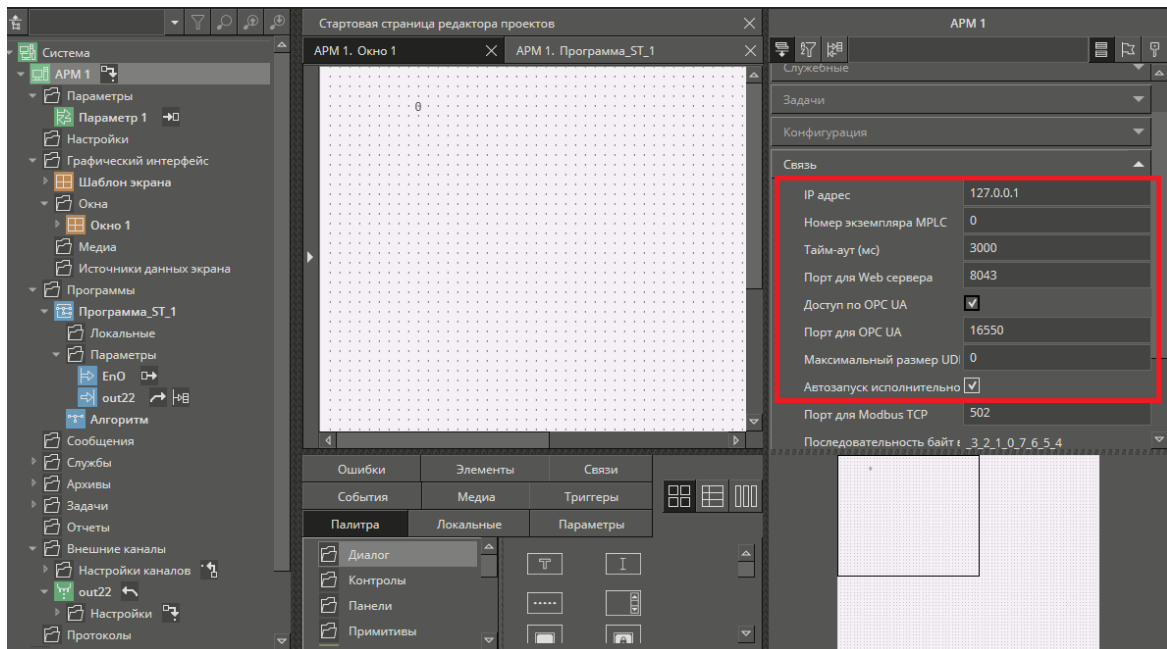
Никаких дополнительных настроек в среде разработки не требуется.



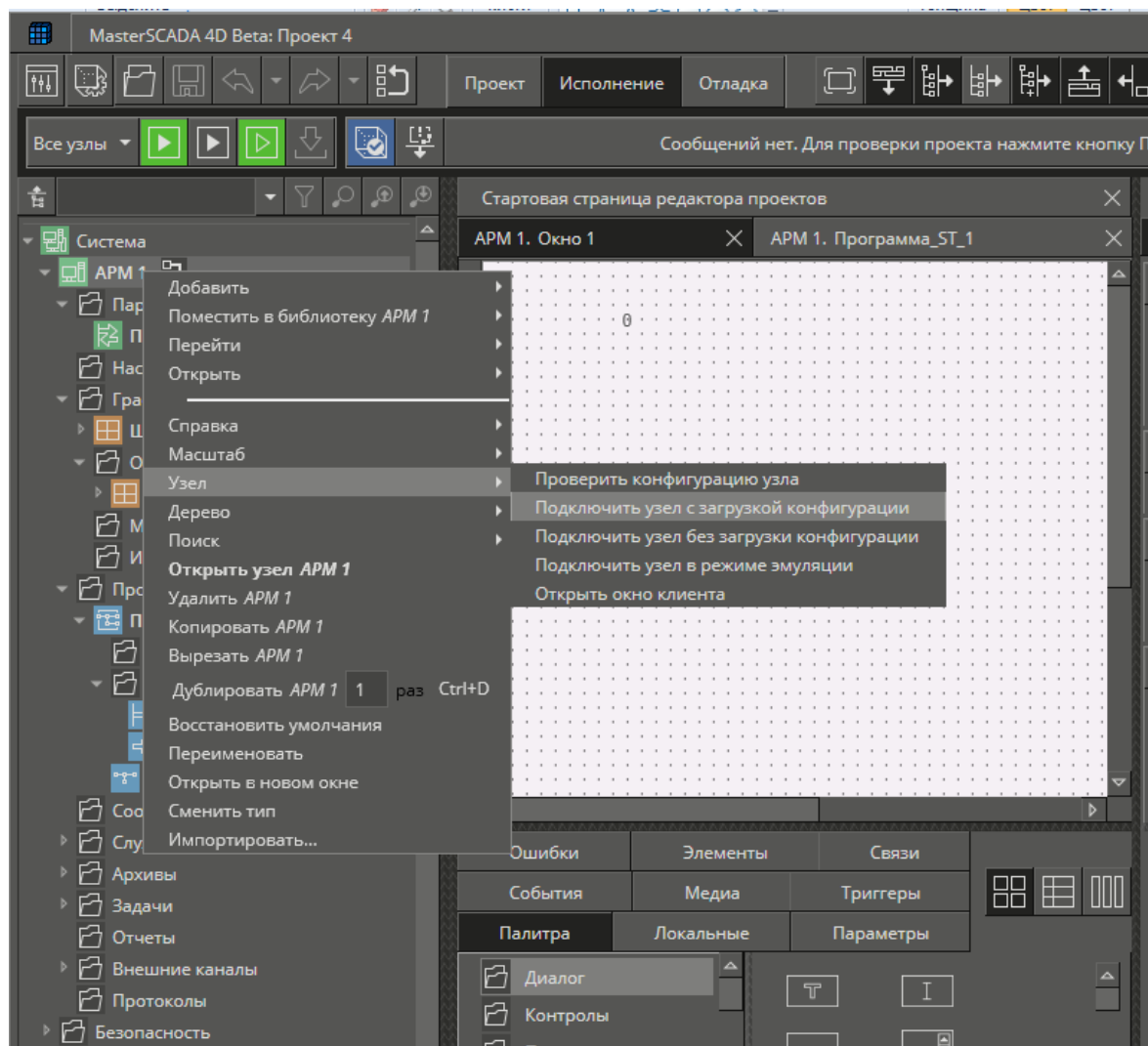
В клиенте достаточно указать IP-адрес узла, в котором будет работать проект MasterSCADA 4D, указать номер порта TCP/IP ( 2404 по умолчанию), адрес ASDU (по умолчанию) и создать карту переменных, согласно настройкам сделанным в настройках элементов группы Внешние каналы.

### 6.3.3.3.2. MasterSCADA 4D как сервер OPC UA

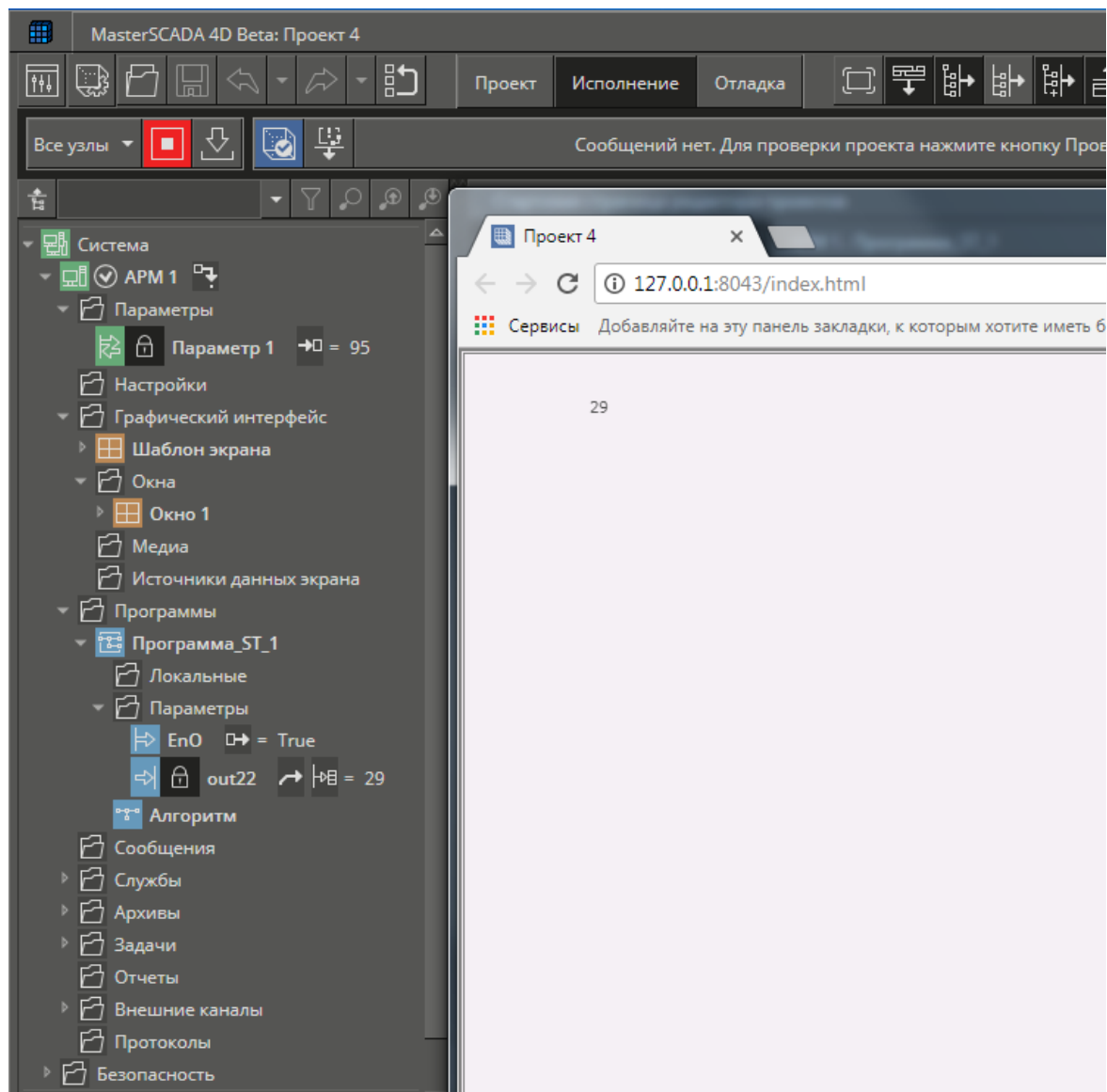
Предположим, что в проекте создан узел с параметрами связи, как показано на рисунке ниже:



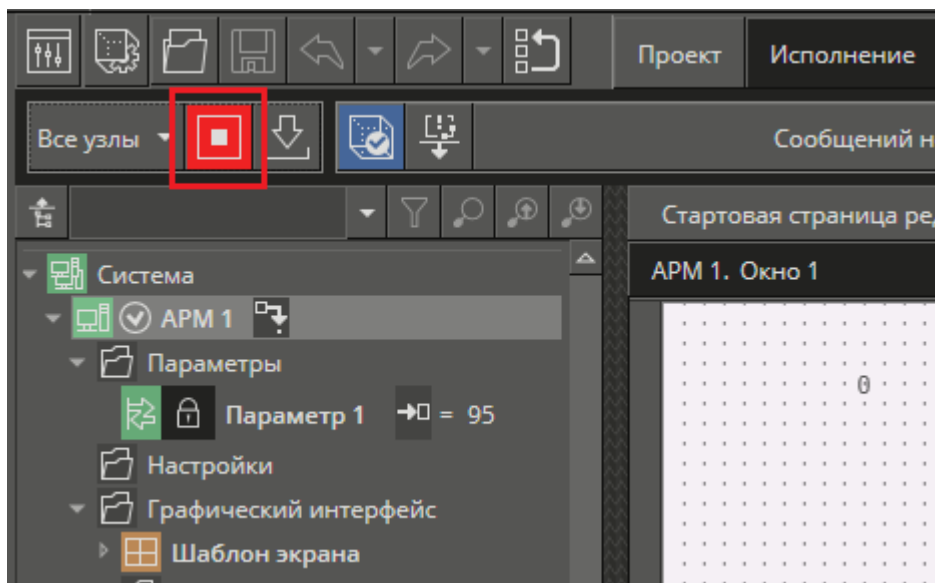
Запустим среду исполнения и загрузим в нее текущий проект. Для этого можно в контекстном меню узла выбрать пункт Узел.Подключить с загрузкой конфигурации



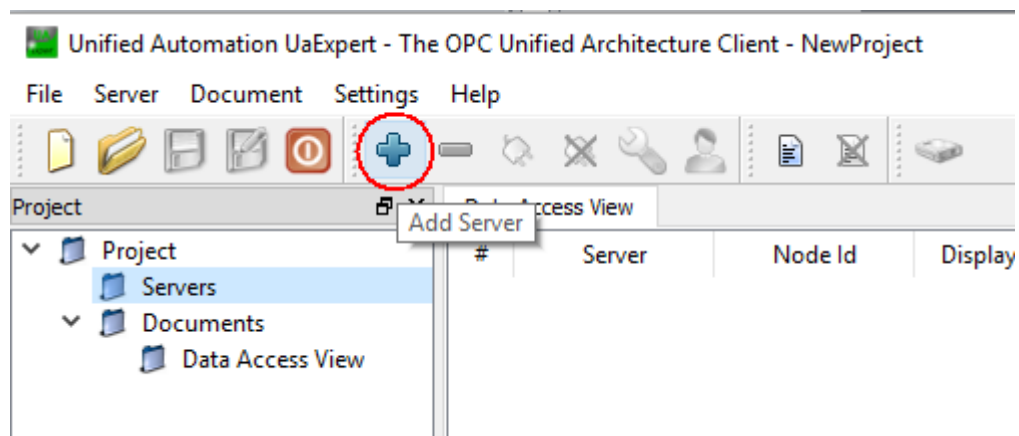
По окончании загрузки конфигурации узла среда подключается к исполнительной системе (при заданных параметрах, кроме того, графический клиент откроется автоматически):



Отключаем среду от исполнительной системы:



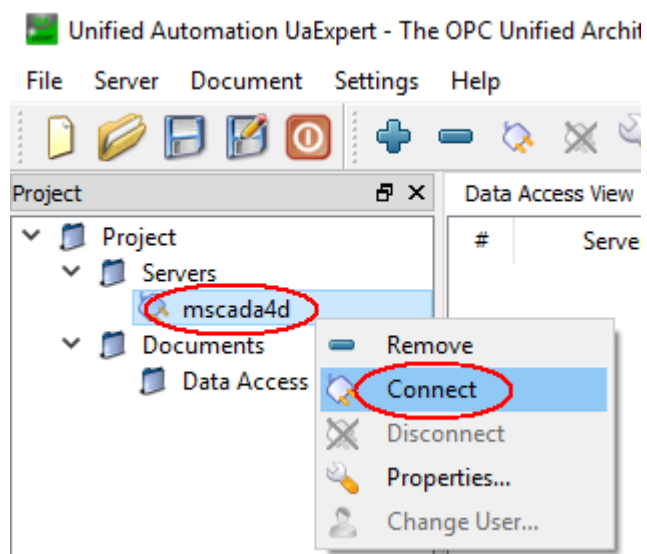
Запускаем клиент OPC UA (в примере используется UaExpert) и нажимаем кнопку добавления сервера:



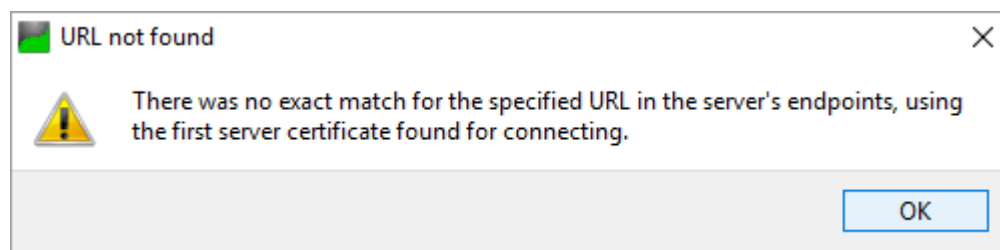
На вкладке Advanced открывшегося диалога конфигурируем подключение к исполнительной системе MasterSCADA 4D как к серверу OPC UA:

The image shows a screenshot of the "Add Server" dialog box in the MasterSCADA4D software. The dialog box has a title bar with a question mark and a close button. The "Configuration Name" field is set to "mscada4d". Below this, there are two tabs: "Discovery" and "Advanced". The "Advanced" tab is selected. The "Server Information" section contains the "Endpoint Url" field, which is set to "opc.tcp://127.0.0.1:16550". The "Security Settings" section has "Security Policy" and "Message Security Mode" both set to "None". The "Authentication Settings" section has "Anonymous" selected, with "Username" and "Password" fields below it. There is also a "Certificate" field and a "Private Key" field, both with "..." buttons next to them. The "Session Settings" section has a "Session Name" field. At the bottom left, there is a checkbox for "Connect Automatically" which is unchecked. At the bottom right, there are "OK" and "Cancel" buttons.

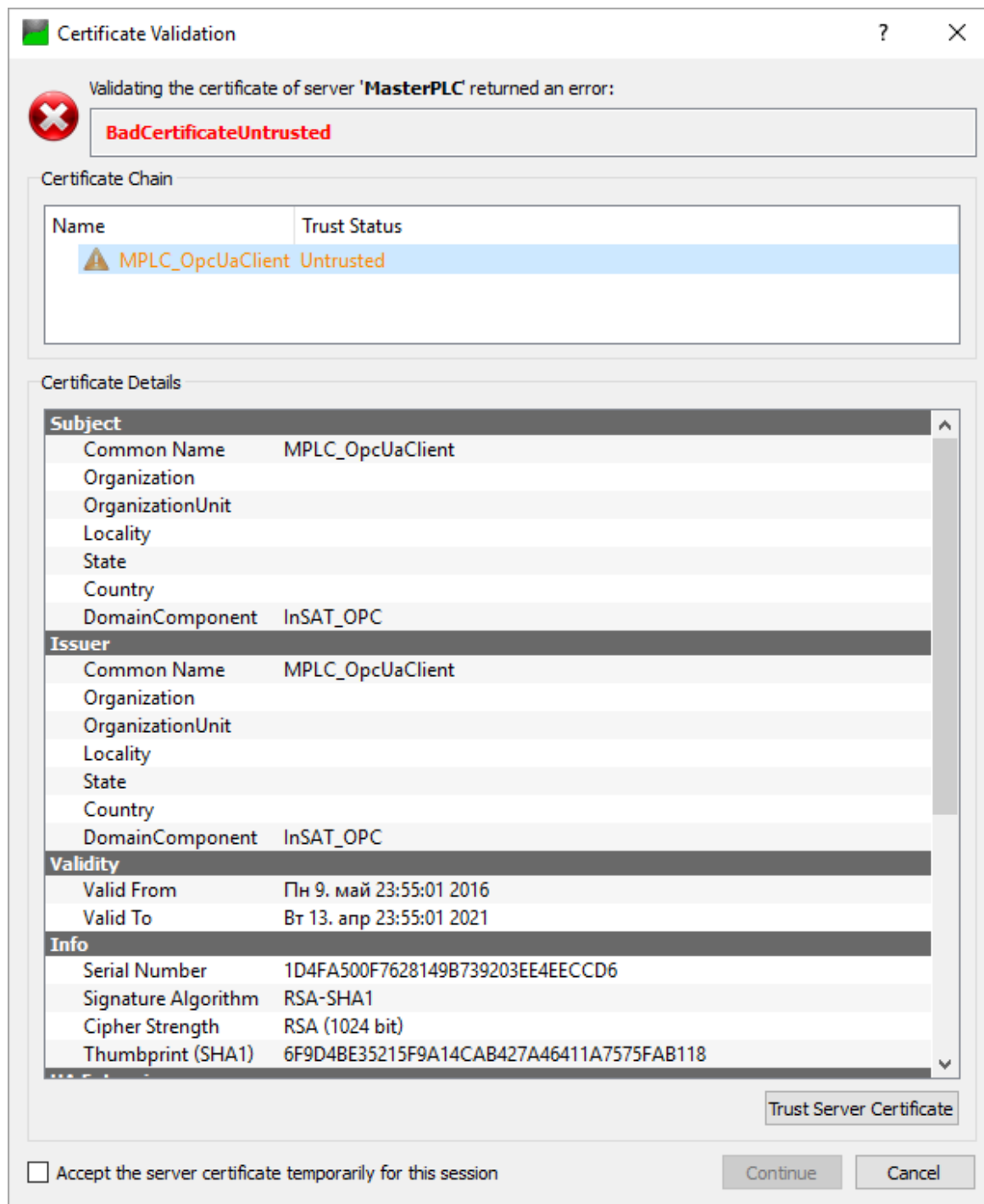
Сервер добавляется в дерево UaExpert. Для подключения к этому серверу выполняем команду Connect из его контекстного меню:



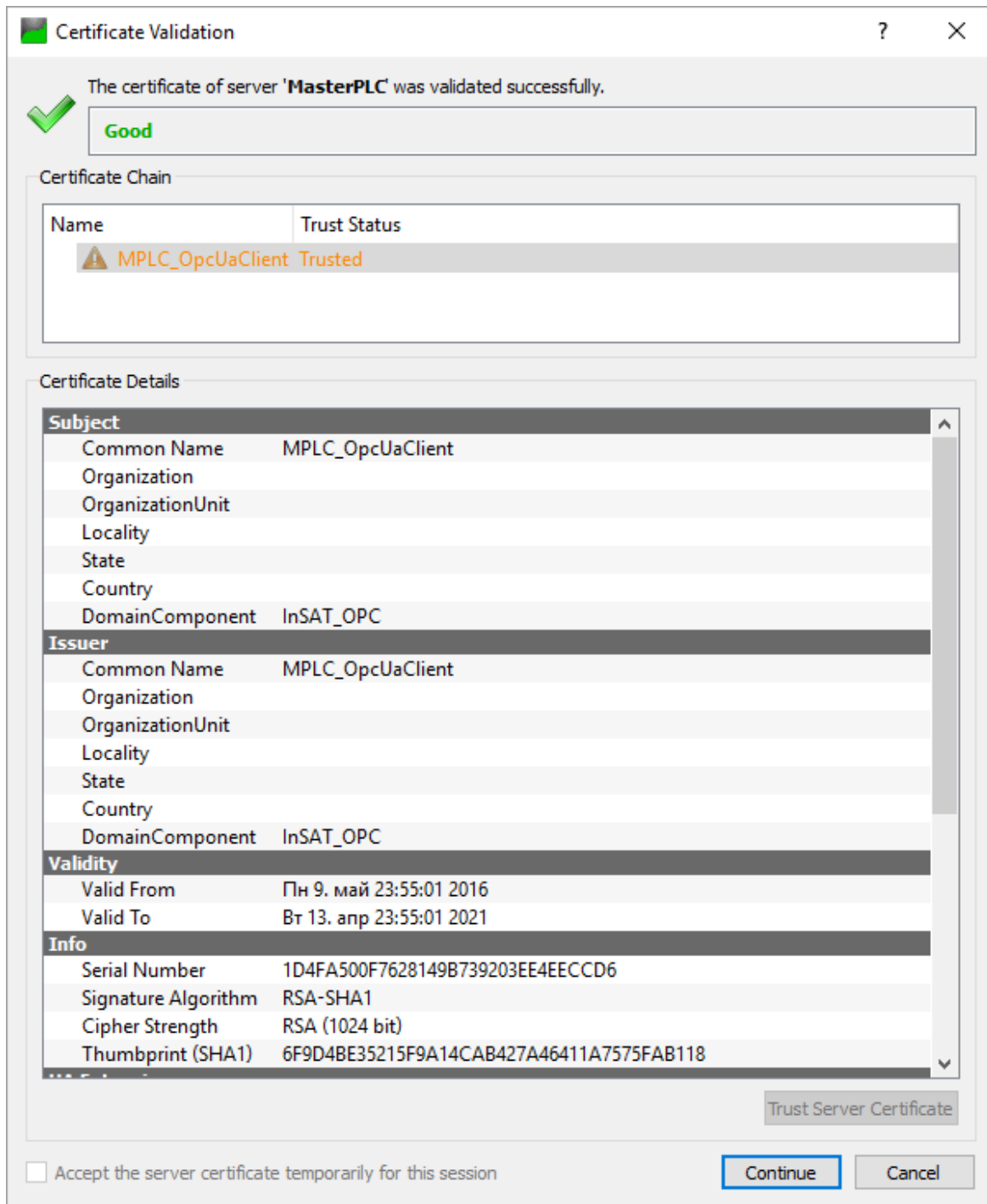
Открывается следующее сообщение:



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:

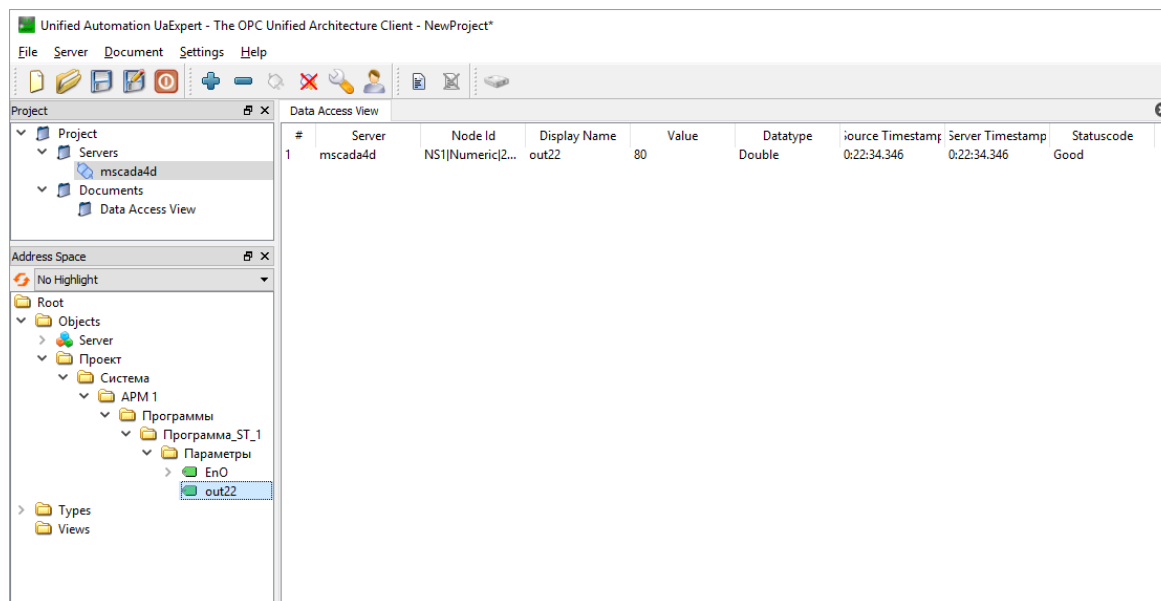


Для исправления ситуации нажимаем кнопку Trust Server Certificate:



Нажимаем Continue, после чего выполняется подключение к серверу:

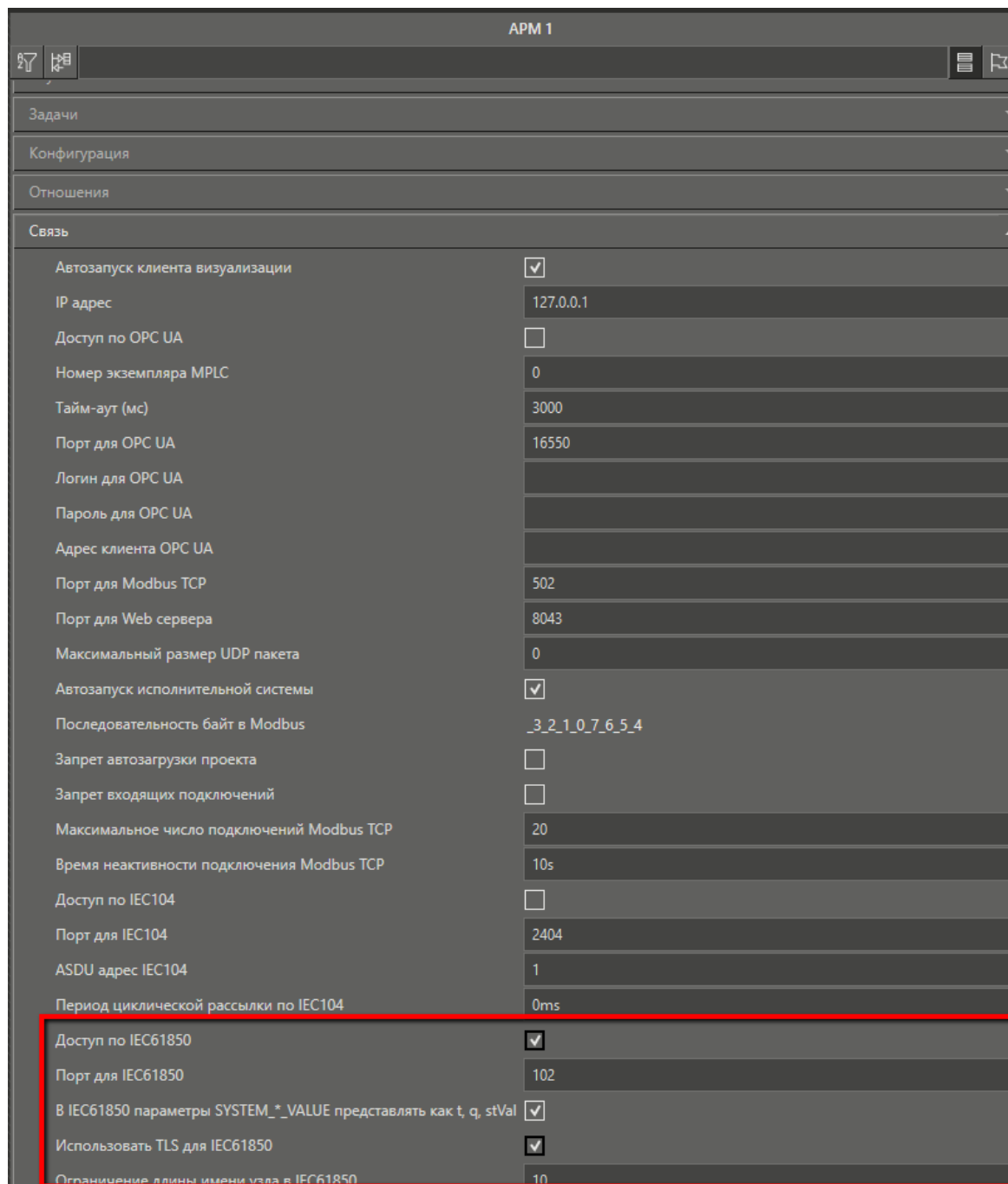




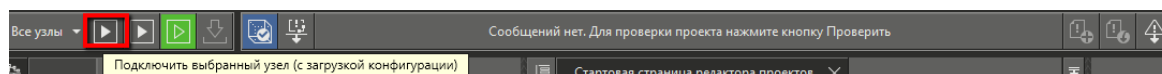
### 6.3.3.3. IEC61850 Slave

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:



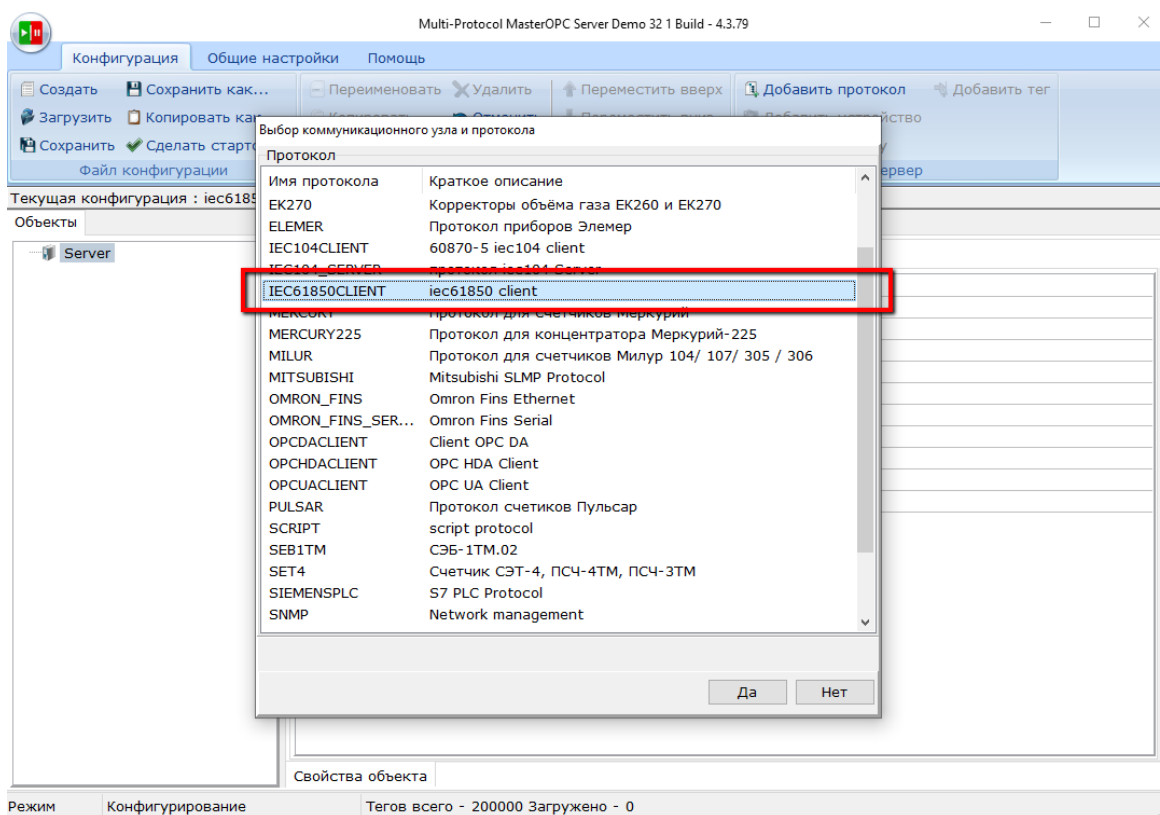
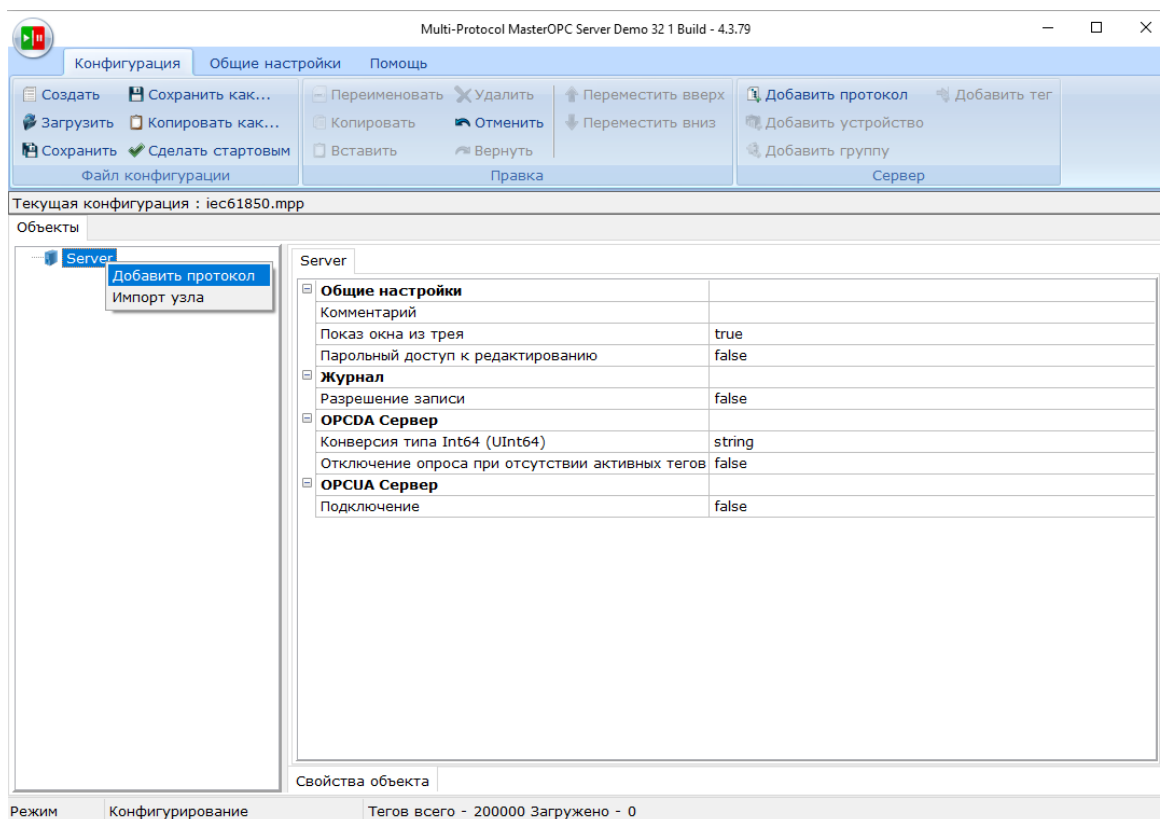
Запустим среду исполнения и загрузим в нее текущий проект:



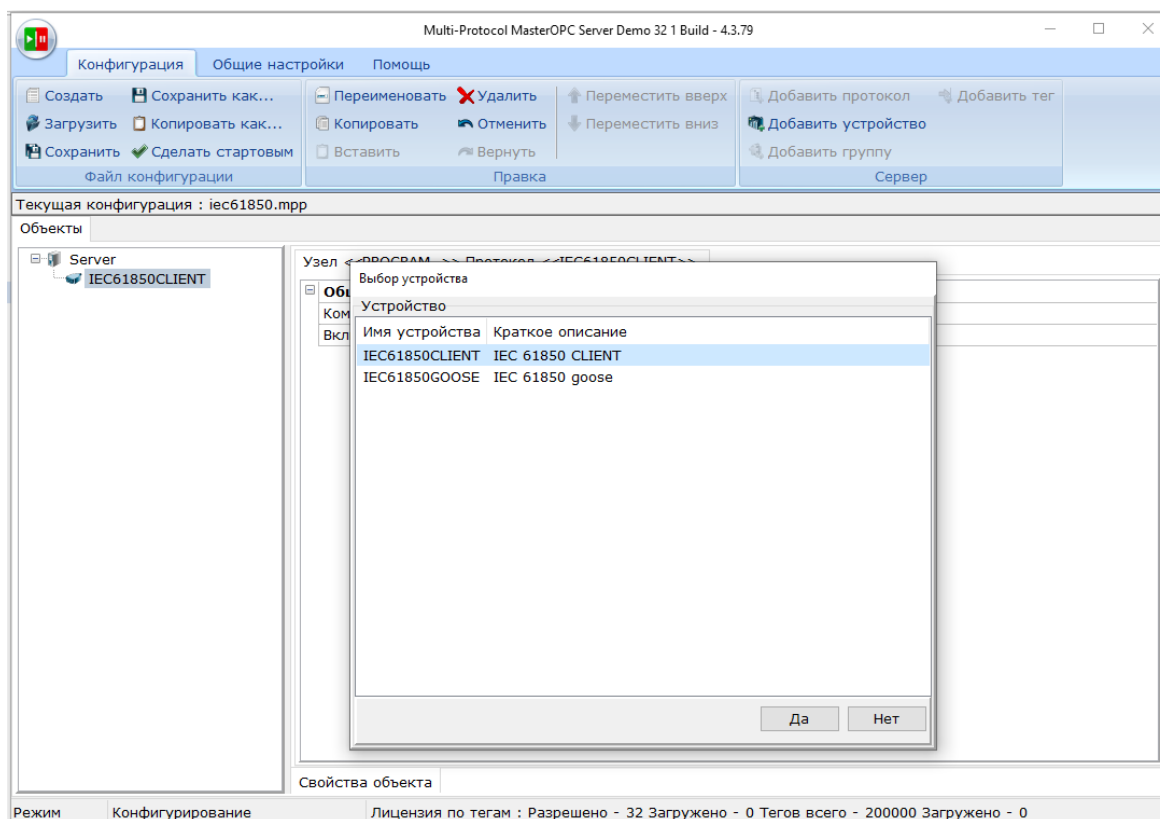
После этого параметры, добавленные в проект, будут доступны клиентам по протоколу IEC61850.

Рассмотрим пример подключения по протоколу IEC61850 приложением Multi-Protocol MasterOPC Server.

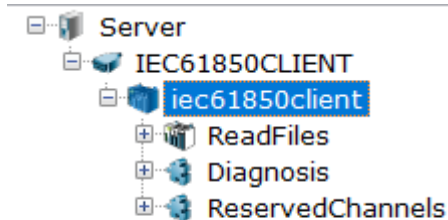
Добавим протокол IEC61850:



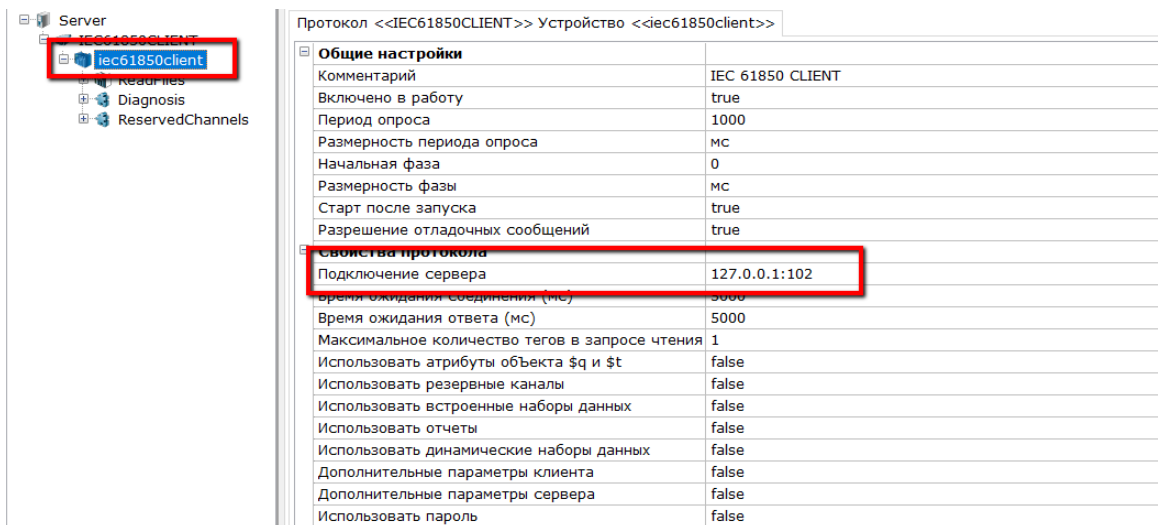
Затем при помощи контекстного меню элемента IEC61850 добавляем устройство:



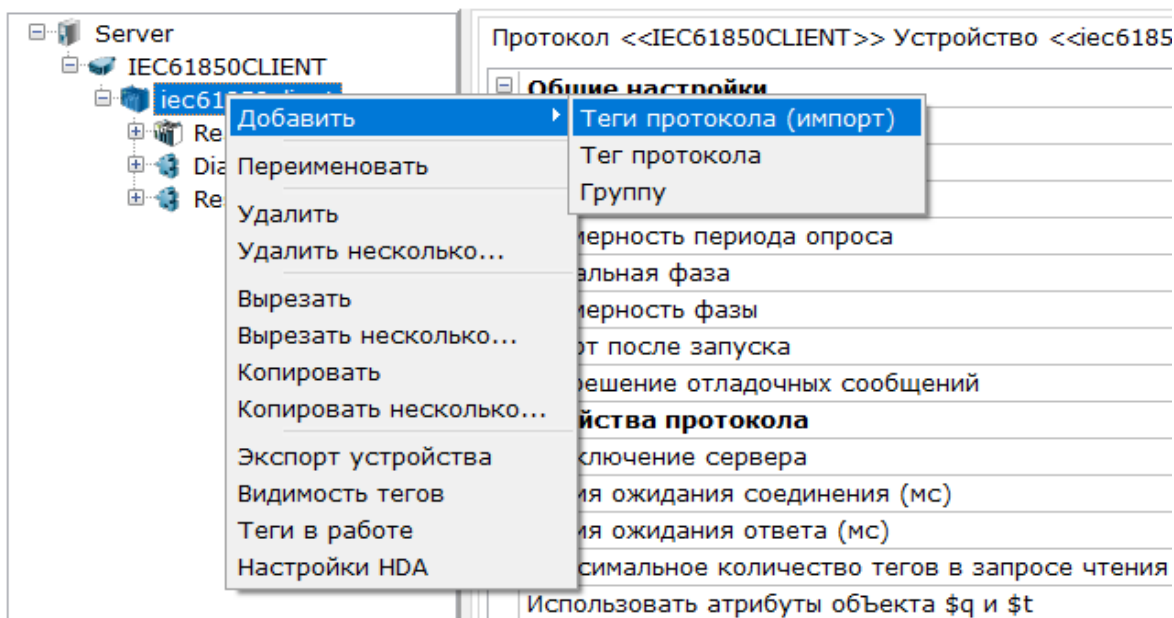
Получим результат:



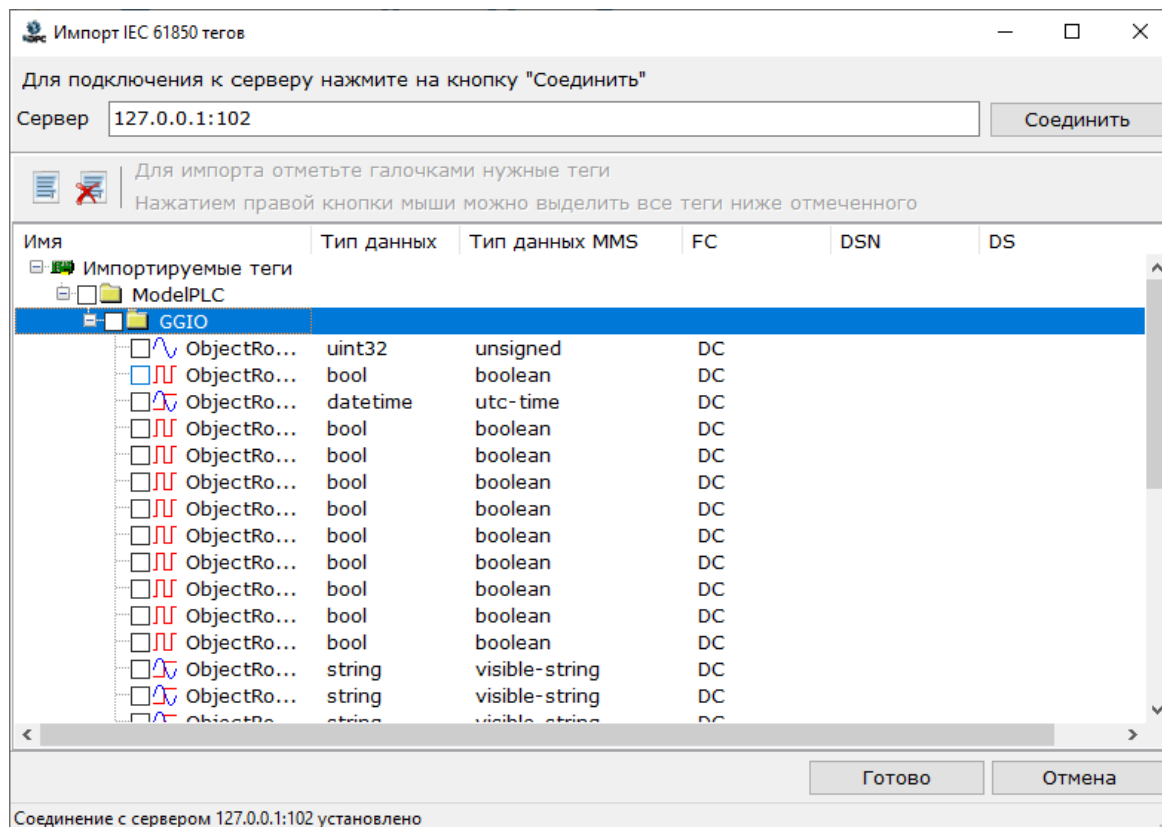
В настройках устройства необходимо указать IP-адрес сервера и номер порта:



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:

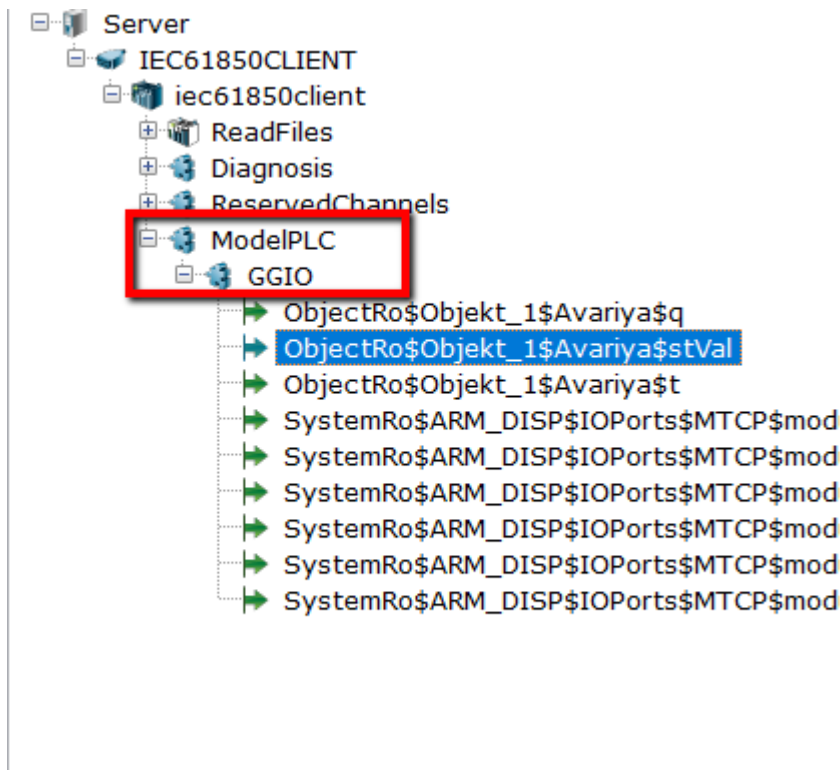


Если установлен флаг В IEC61850 параметры SYSTEM\_\*\_VALUE представлять как t, q, stVal в настройках узла MasterSCADA 4D, то параметры типа SYSTEM\_\*\_VALUE будут представлены в списке тремя параметрами, например:

ObjectRoot\$Objekt_1\$Temperatur\$q	uint32	unsigned	
ObjectRoot\$Objekt_1\$Temperatur\$stVal	float	float	
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 приступить к опросу. Для этого необходимо запустить приложение Multi-Protocol MasterOPC Server

#### 6.3.3.4. Типовые каналы обмена

По умолчанию библиотека MasterSCADA 4D содержит следующие каналы:

Название	Тип данных параметров	Доступ
AI	SYS- TEM_LREAL_PARAM	Чтение
DI	SYSTEM_BOOL_PARAM	Чтение
AO	SYS- TEM_LREAL_PARAM	Запись
DO	SYSTEM_BOOL_PARAM	Запись
AIO	SYS- TEM_LREAL_PARAM	Чтение и запись

DIO	SYSTEM_BOOL_PARAM	Чтение и запись
-----	-------------------	-----------------

Эти структуры содержат параметр StatusCode (статус), начальное значение которого установлено в `OpcUa_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).

### 6.3.3.5. 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.

Смотрите также:

Категория Задача

### 6.3.3.6. Новые типы протоколов и модулей

Раздел находится в разработке

### 6.3.3.7. Рекомендации по настройке 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 "ИнСАТ", которые могут преобразовывать стандарт OPC DA/HDA в UA.

## 6.4. 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.

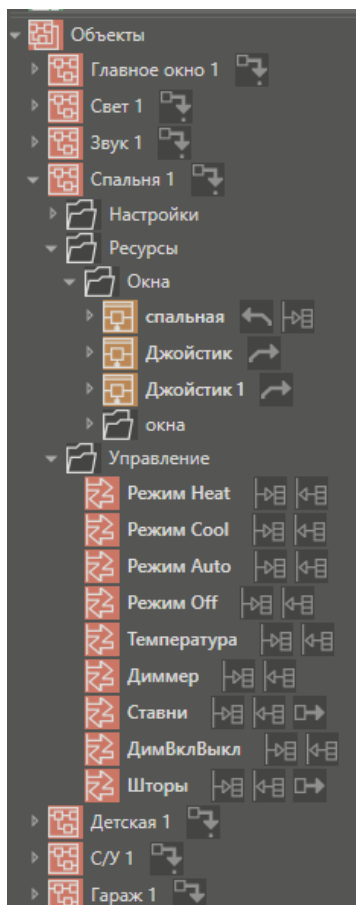
Первым в данном дереве создается Объект.

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).

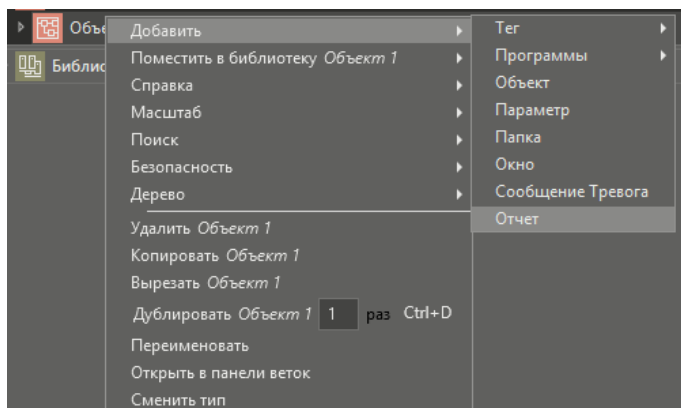
В случае, если проект состоит из нескольких типовых объектов, например, в цеху есть три однотипные линии, которые включают в себя набор одинаковых конвейеров, то целесообразно сначала создать в библиотеке типовой объект Конвейер. Этот объект следует включить в другой типовой библиотечный объект Линия, который уже можно добавить в дерево объектов. За счет тиражирования ускорится разработка, а за счет использования наследования упростится дальнейшее внесение правок.

### **6.4.1.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 груп-

пах 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.

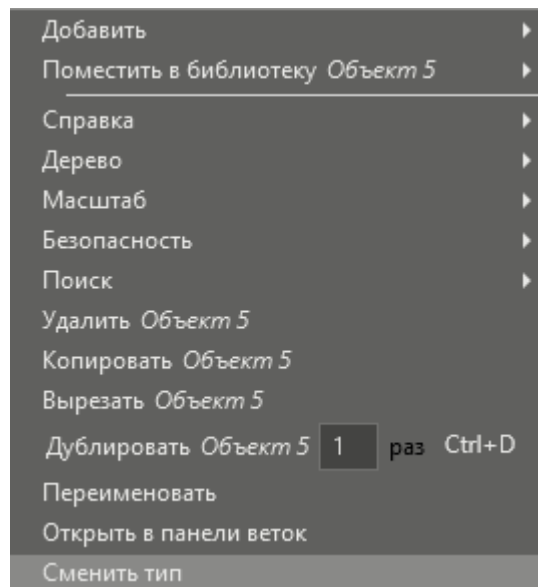
### Использование библиотечных объектов в дереве объектов

Если объект используется в проекте неоднократно, то необходимо воспользоваться механизмом типизирования элементов проекта.

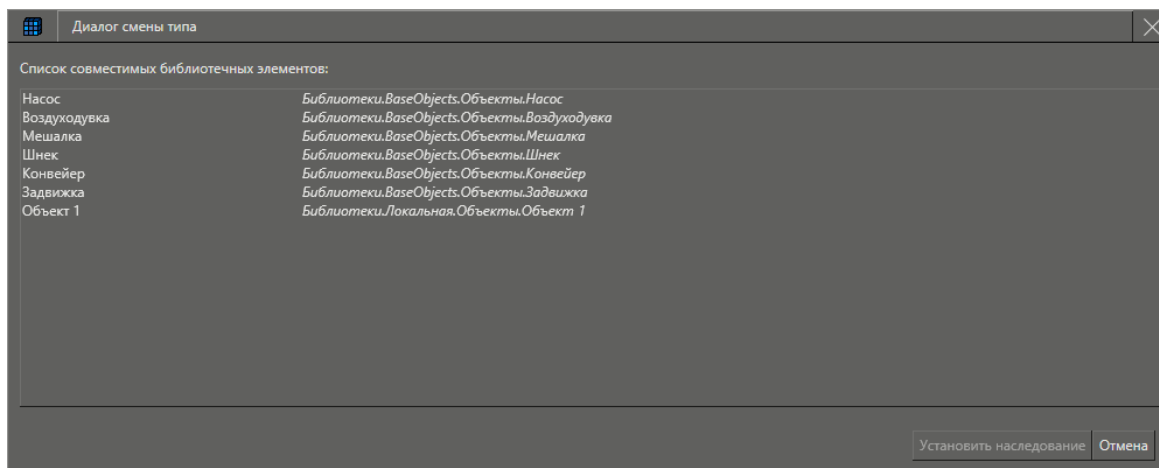
Типизация обеспечивает многократное использование одного и того же объекта как шаблона в рамках одного проекта, так и при разработке других систем. При изменении типа объекта в библиотеке, наследование обеспечивает изменение всех объектов – экземпляров этого типа – в проекте. Это значительно упрощает 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.

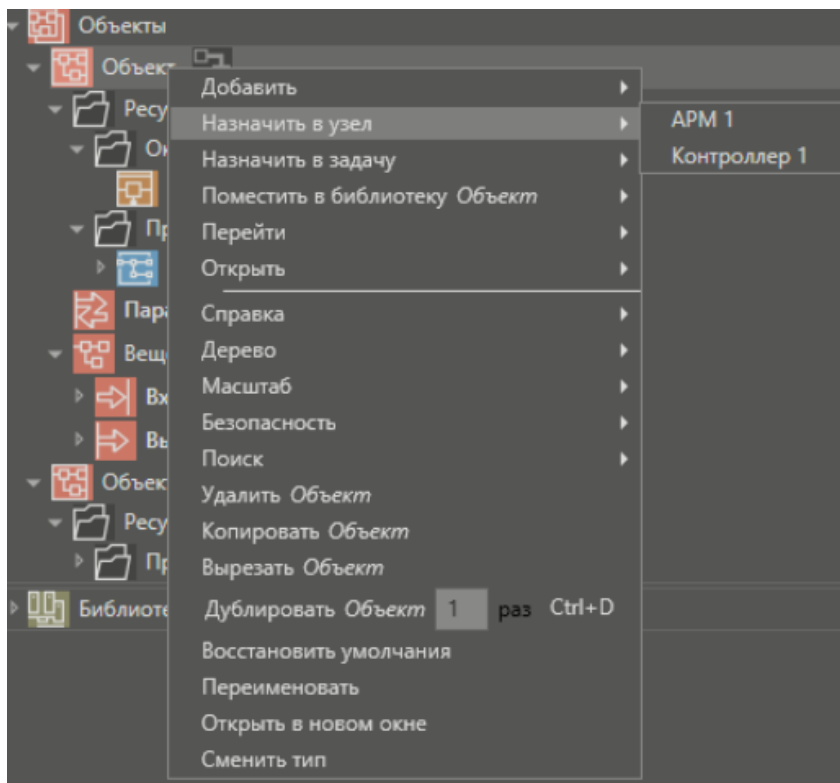
**Важно!** В версии 1.2.4 при перетаскивании объекта из библиотеки левой кнопкой мыши добавляется в проект объект с типом отношения Унаследован от.

## 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

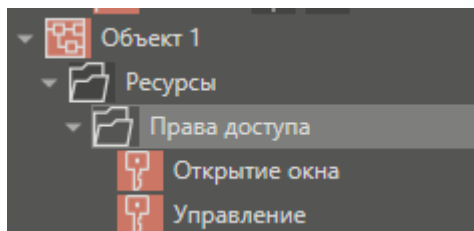
Для того чтобы задать место исполнение объекта, предусмотрены пункты контекстного меню Назначить в узел и Удалить из узла , а для распределения объектов по задачам – команды Назначить в задачу и Удалить из задачи.



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



#### 6.4.1.1. Панель свойств Объекта

Панель свойств Объекта имеет вид:

Объект 1

Общие

Имя	Объект 1
Программное имя	Объект 1
Полное имя	Объекты.Объект 1
Метки	
Комментарий	
Место исполнения	В задаче экрана
Отключить исполнение	<input type="checkbox"/>

Разрешения

Всегда отображать в дереве	<input type="checkbox"/>
----------------------------	--------------------------

Службные

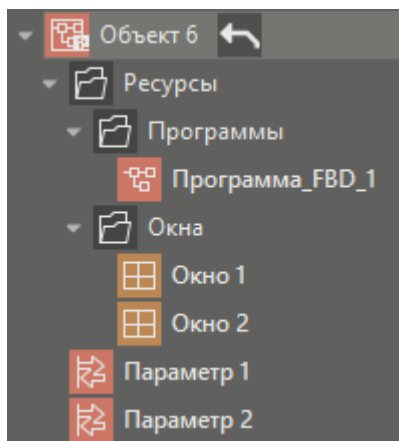
Категория	Рекомендации
Общие	В данной категории у объекта, как правило, редактируются свойства Имя, Место исполнения.
Разрешения	Данную категорию имеет смысл настраивать только у вложенных объектов, которые необходимо отображать в дереве, несмотря на то, что у них содержимое родительского объекта скрыто от других разработчиков проекта.

## 6.4.2. 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.

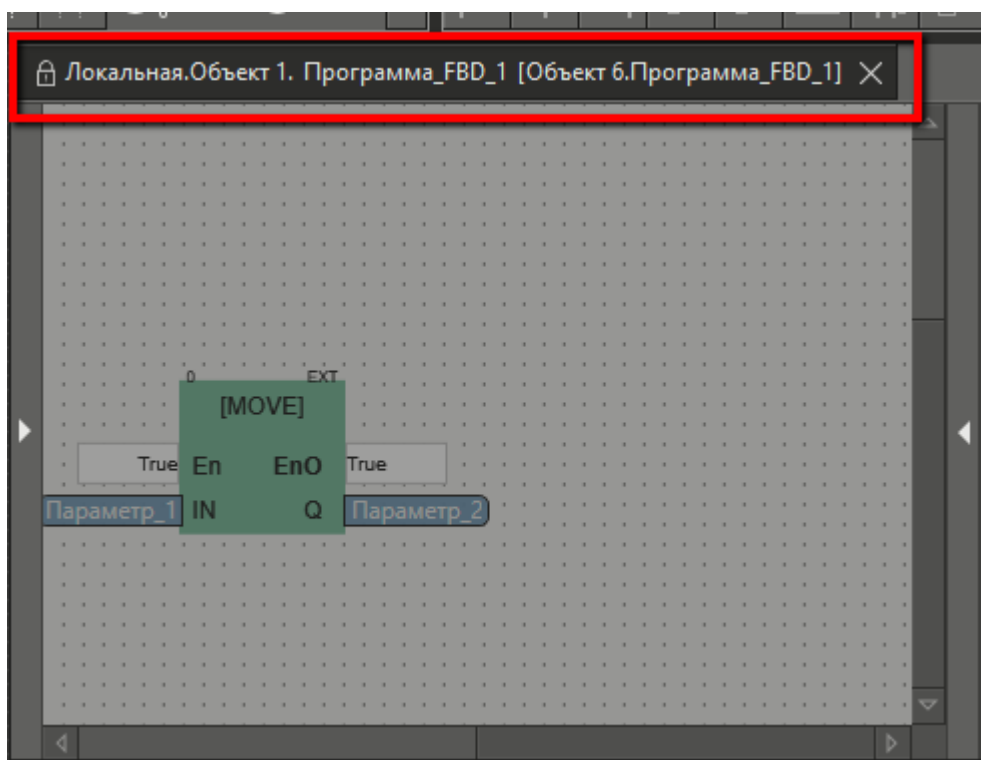
Вид экземпляра объекта в дереве:



### 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 элемент не сохраняется.

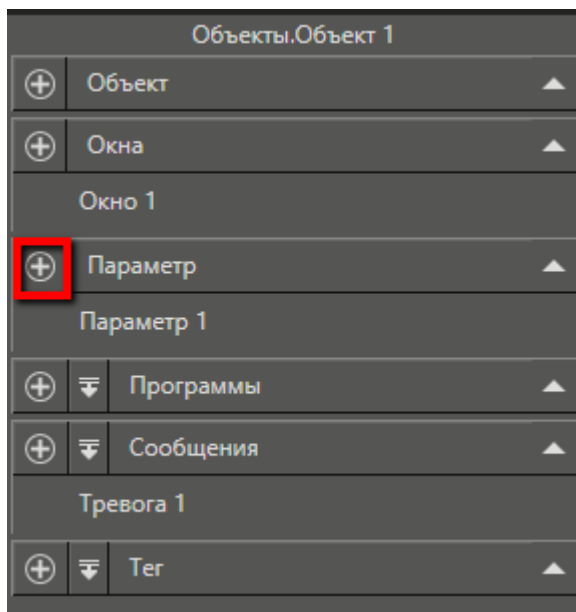
Значение Наследуется в типе объекта в библиотеке или в типе ФБ означает, что элемент не сохраняется.

### **6.4.3.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.

#### 6.4.4.Папка

Назначение этой группы – дополнительное структурирование проекта. Папка не имеет специальных настроек.

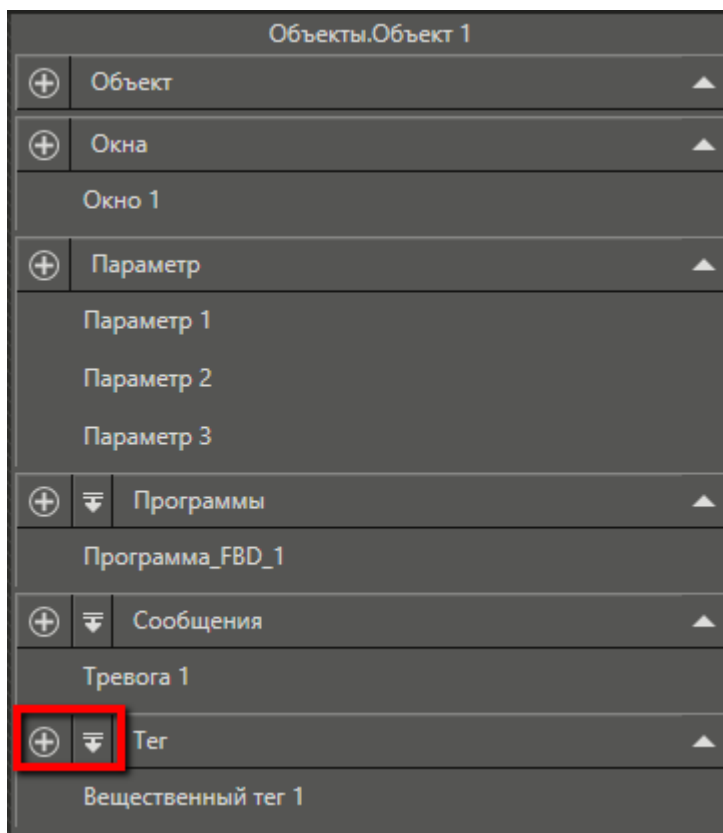
#### 6.4.5.Тег

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:

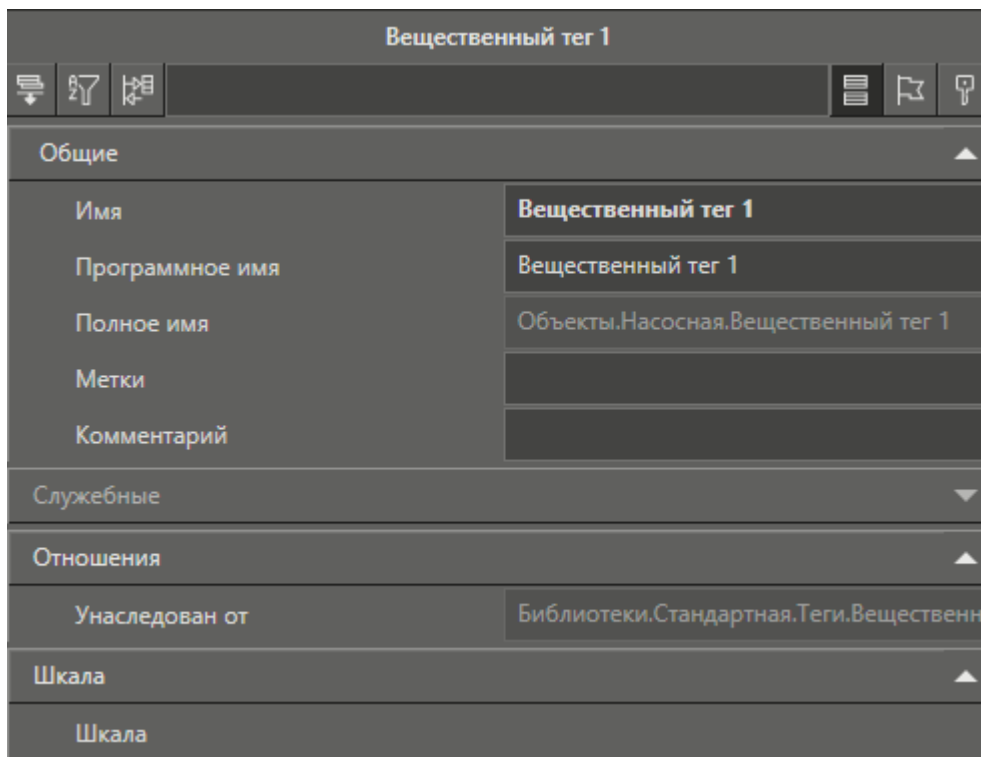


В тег, находящийся в дереве объектов, можно добавить окно, тревогу или программу при помощи контекстного меню пункта Добавить. Это имеет смысл в двух случаях:

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

#### 6.4.5.1. Панель свойств Тега

Вид панели свойств:



#### 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 Master-SCADA 4D
Шкала	Можно выбрать шкалу для тега, для автоматического формирования сообщений при нарушении границ.

#### 6.4.5.2. 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.

### 6.4.5.2.1. Вещественный тег

Тег , в котором тип данных параметров Вход и Выход – SYSTEM\_LREAL\_PARAM. Данные со входа автоматически передаются на выход

### 6.4.5.2.2. Целый тег

Тег , в котором в котором тип данных параметров Вход и Выход – SYSTEM\_DINT\_PARAM. Данные со входа автоматически передаются на выход.

### 6.4.5.2.3. Логический тег

Тег, в котором в котором тип данных параметров Вход и Выход – SYSTEM\_BOOL\_PARAM. Данные со входа автоматически передаются на выход

## 6.4.5.3. Новый тип тега

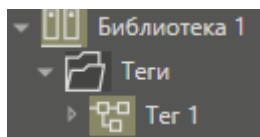
Новый тип тега создается в том случае, если нельзя решить задачу, используя Параметр, например, когда при получении данных на вход тега, их необходимо обработать в программе, отобразить нестандартным образом в окне, передать результат работы программы на выход, и выдать какое-либо сообщение.

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

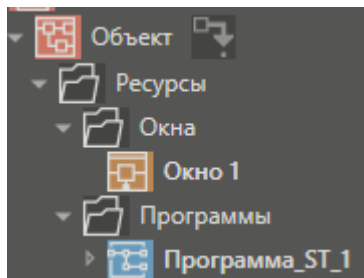
Созданные типы тегов будут находиться в разделе пользовательской библиотеки Теги:



## 6.4.6.Ресурсы

Группа ресурсы может быть у таких элементов проекта как Объект, Тег, Канал.

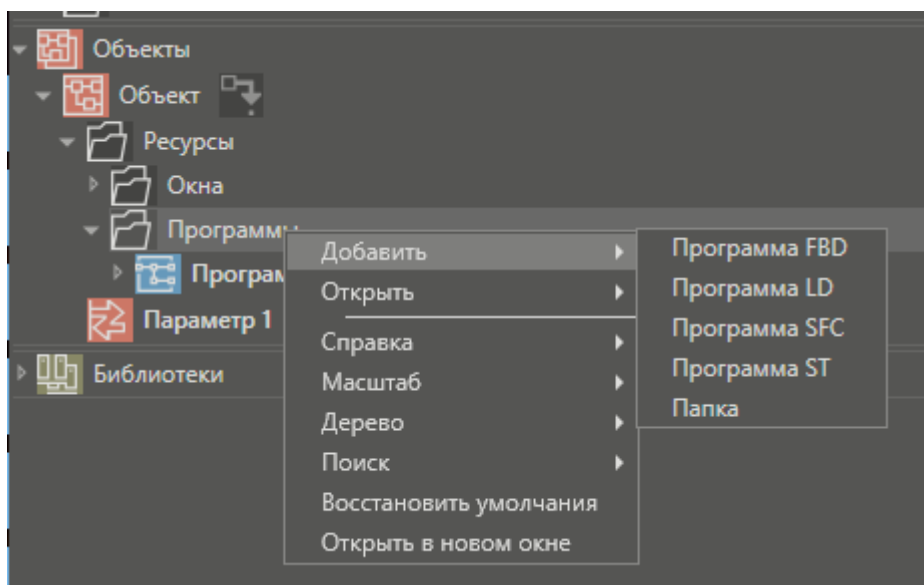
По умолчанию группа Ресурсы не содержит дочерних элементов. После выполнения пункта контекстного меню или контекстной панели Добавить-Программы, Добавить-Окна или Добавить-Сообщения, в ней появятся соответствующие группы: группа Окна, группа Программы, группа Сообщения



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).

### 6.4.6.1. 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.

### 6.4.6.2. 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 планируются использовать в других проектах. В противном случае в дереве объектов следует использовать типизированные окна.

### **6.4.6.3. Сообщения**

В данной группе находятся сообщения объектов.

В группу можно добавлять экземпляры стандартного сообщения Тревога, а также экземпляры пользовательских сообщений.

### **6.4.6.4. Отчеты**

MasterSCADA 4D поддерживает создание отчетов как на основе текущих данных, так и на основе архивных данных.

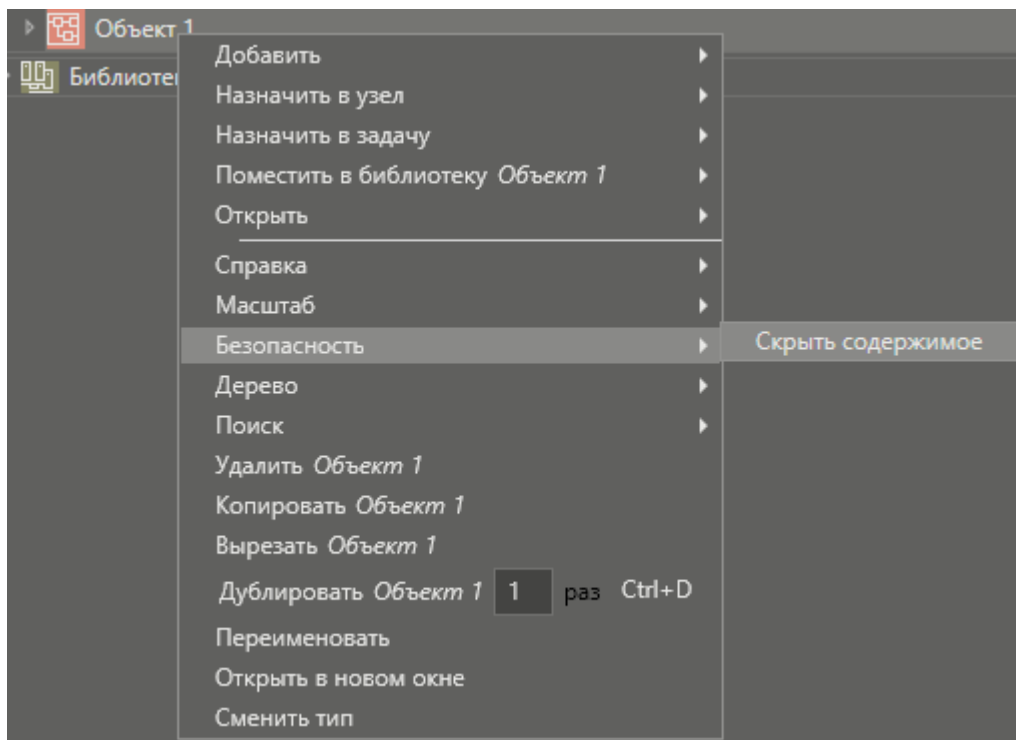
В этой группе создаются отчеты, которые в дальнейшем редактируются в дизайнера отчетов.

После создания отчета, дизайнер отчетов открывается автоматически. Для повторного открытия необходимо дважды нажать на созданный элемент, или воспользоваться пунктом контекстного меню Редактировать.

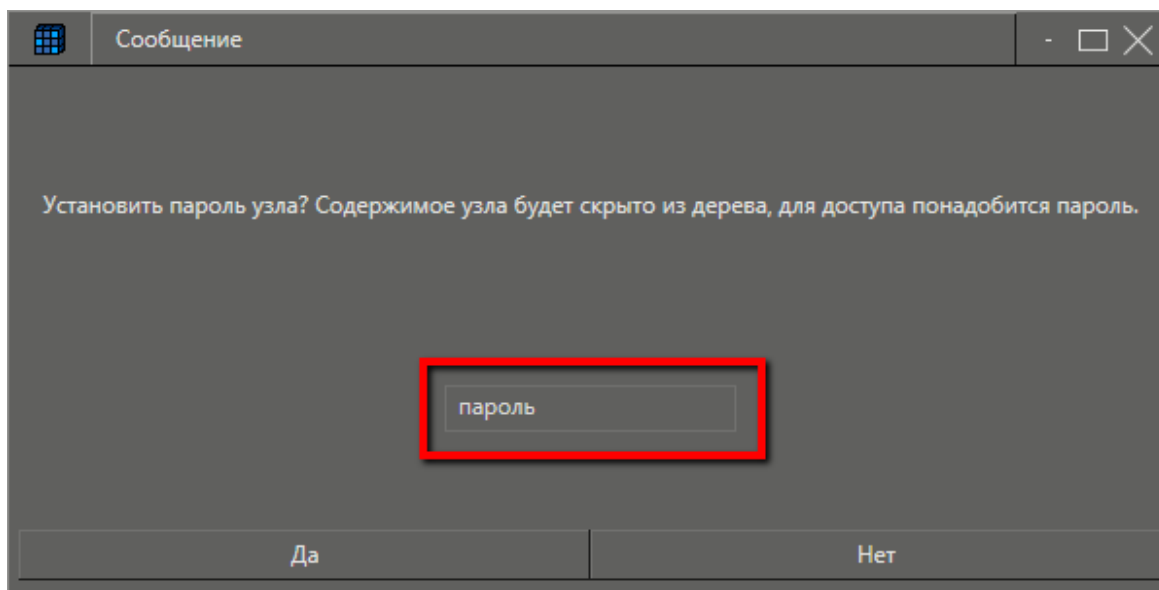
Группа в узле не отображается до тех пор пока не будет добавлен хотя бы один отчет. Добавить отчет в узел в этом случае можно при помощи контекстного меню или контекстной панели.

### **6.4.7. Установка пароля на содержимое объектов**

В контекстном меню объекта предусмотрена команда Безопасность.Скрыть содержимое:

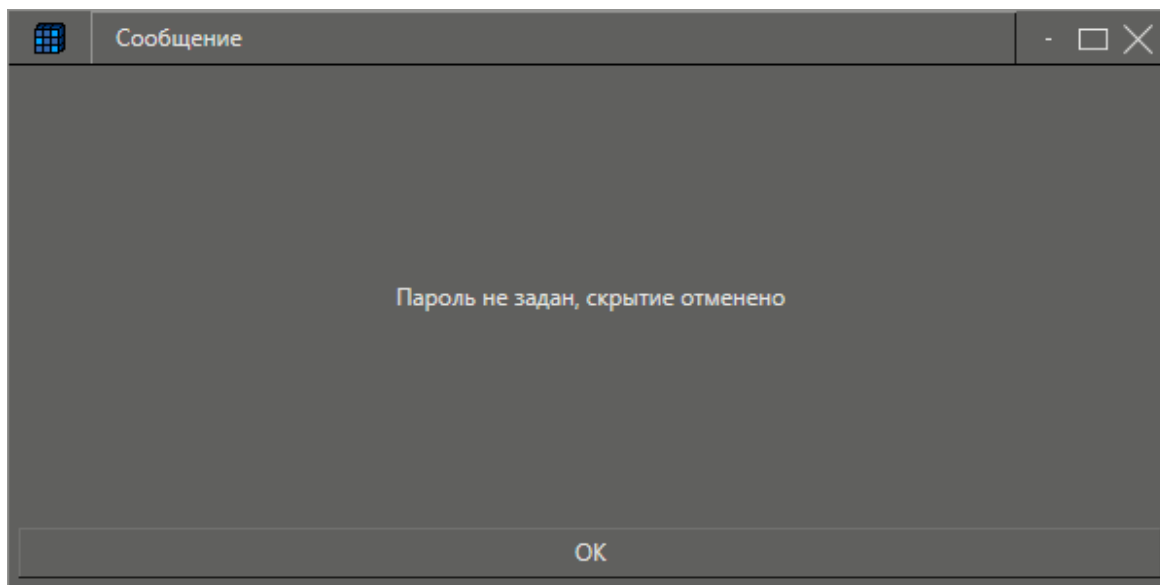


По этой команде открывается диалоговое окно, в котором можно задать пароль:

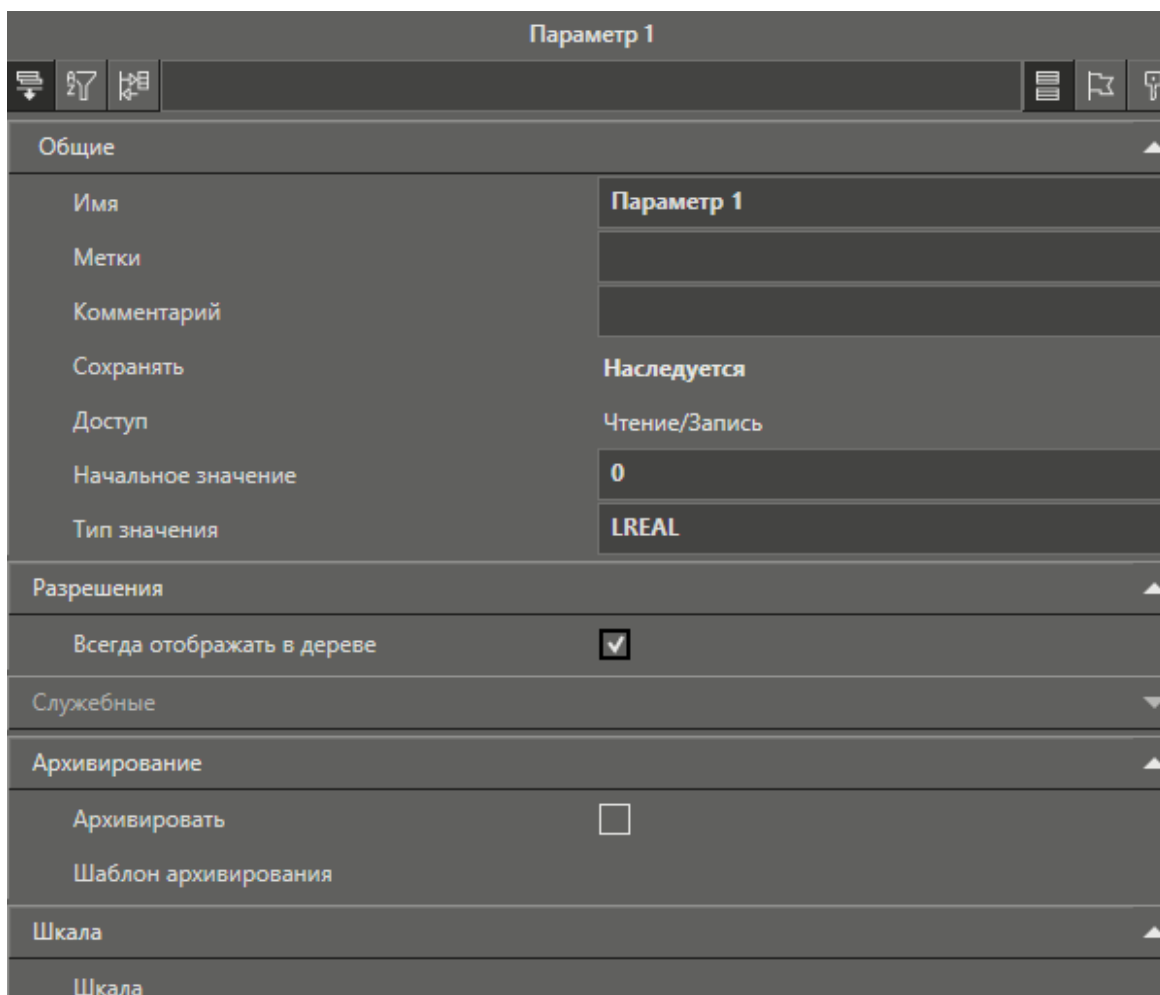


Если нажать кнопку Нет, то диалоговое окно закрывается.

Если нажать кнопку Да, не задав пароль, то окно закрывается, и открывается сообщение об отмене скрывания содержимого объекта:

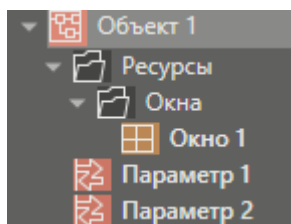


Если задать пароль и нажать кнопку Да, то содержимое объекта скрывается, и в дереве остается видимым только собственно элемент Объект, а также элементы проекта, у которых в панели свойств установлен флаг Всегда отображать в дереве:

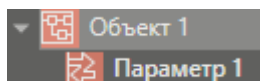




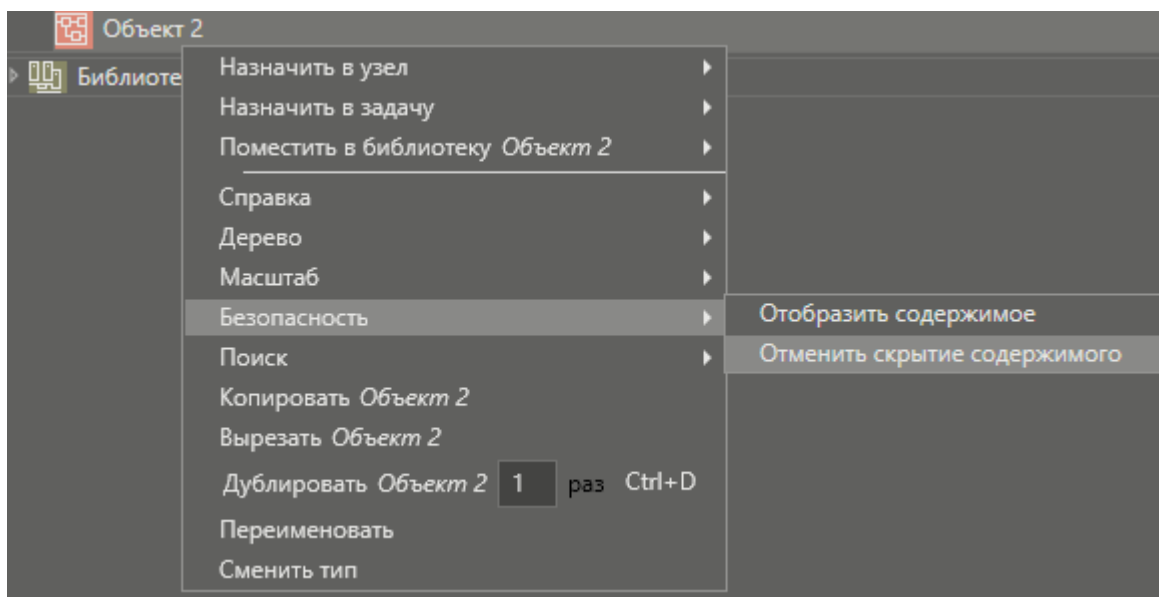
Вид объекта в дереве до скрытия содержимого:



Вид после:



Чтобы отменить данную операцию, следует в контекстном меню выбрать пункт Отменить скрытие содержимого:




Если требуется, чтобы содержимое объекта отобразилось только на время текущей сессии работы с проектом, то необходимо выполнить пункт контекстного меню Отобразить содержимое.

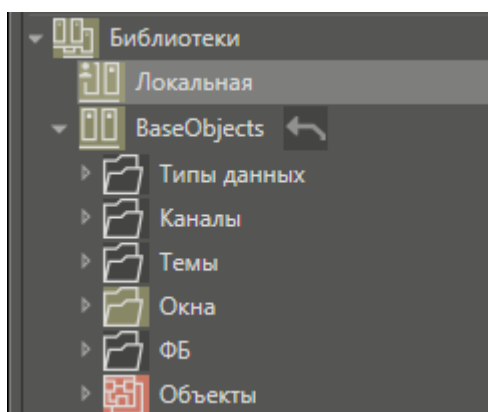
При выполнении любого из этих пунктов откроется диалоговое окно, в котором следует ввести ранее установленный пароль.

## 6.5. Дерево библиотек


В дереве библиотек находятся библиотеки проекта и выполняются операции с ними.

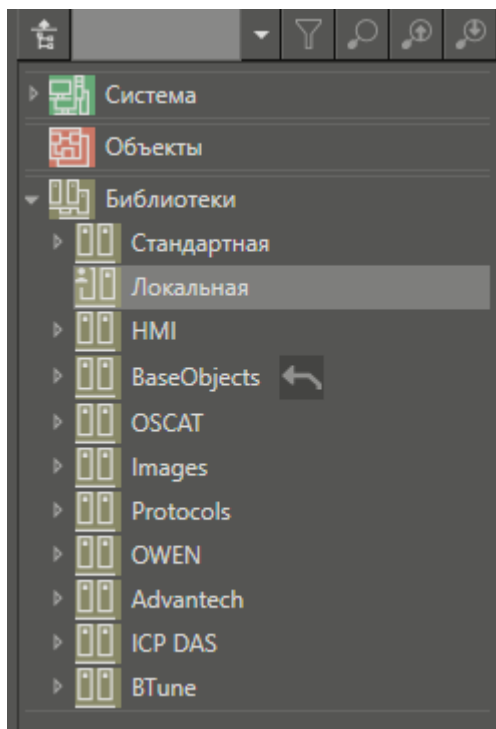
Важно! Элементы проекта, которые добавляются в дерево системы или в дерево объектов чаще всего являются экземплярами какого-либо библиотечного элемента: узла, протокола, программы, функции, канала, параметра и др.

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.

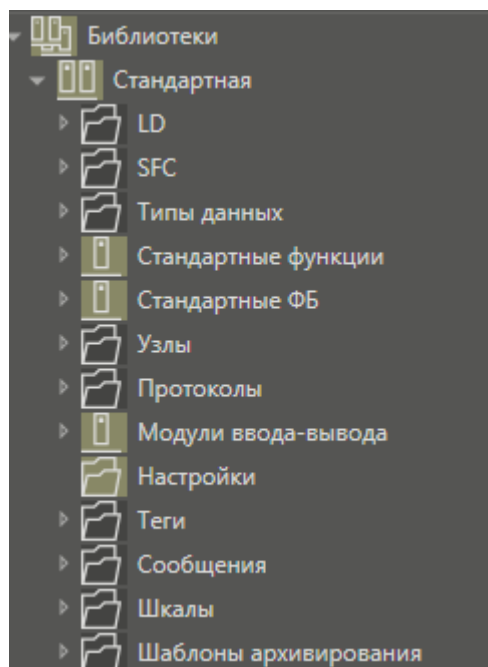
Поставляемые библиотеки содержат ряд элементов, необходимых для разработки проекта (например, для конфигурирования обмена с соответствующим оборудованием) или существенно упрощающих разработку. Поставляемые библиотеки хранятся в папке <папка установки MasterSCADA 4D>\Configs\MasterPLC\Config\Library\.

**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.

### 6.5.1. 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.
Шаблоны архивирования	Содержит основные типы шаблонов архивирования. На основании этих типов можно создать свои типы шаблонов архивирования.

Смотрите также:

Типы и экземпляры в MasterSCADA 4D

### 6.5.1.1. Типы данных

Группа Типы данных библиотеки Стандартная содержит встроенные производные типы данных (структуры):

- SYSTEM\_LREAL\_PARAM,
- SYSTEM\_BOOL\_PARAM,
- SYSTEM\_DINT\_PARAM,
- SYSTEM\_STRING\_PARAM,
- SYSTEM\_COUNTER\_PARAM

Эти структуры содержат 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 ).

#### **6.5.1.1.1. SYSTEM\_LREAL\_PARAM**

Структура, содержащая следующие переменные:

- Value – значение, тип данных LREAL;
- SourceTime – метка времени, тип данных DT;
- StatusCode – статус (качество), тип данных UDINT.

#### **6.5.1.1.2. SYSTEM\_BOOL\_PARAM**

Структура, содержащая следующие переменные:

- Value – значение, тип данных BOOL;
- SourceTime – метка времени, тип данных DT;
- StatusCode – статус (качество), тип данных UDINT.

#### **6.5.1.1.3. SYSTEM\_DINT\_PARAM**

Структура, содержащая следующие переменные:

- Value – значение, тип данных DINT;
- SourceTime – метка времени, тип данных DT;
- StatusCode – статус (качество), тип данных UDINT.

#### **6.5.1.1.4. SYSTEM\_STRING\_PARAM**

Структура, содержащая следующие переменные:

- Value – значение, тип данных STRING;
- SourceTime – метка времени, тип данных DT;
- StatusCode – статус (качество), тип данных UDINT.

#### **6.5.1.1.5. SYSTEM\_COUNTER\_PARAM**

Структура, содержащая следующие переменные:

- Value – значение, тип данных DINT;
- BoolValue – значение, тип данных BOOL

- SourceTime – метка времени, тип данных DT;
- StatusCode – статус (качество), тип данных UDINT.

### 6.5.1.1.6. StatusCode

Входит в структуру следующих типов данных:

- SYSTEM\_LREAL\_PARAM,
- SYSTEM\_BOOL\_PARAM,
- SYSTEM\_DINT\_PARAM,
- SYSTEM\_STRING\_PARAM,
- SYSTEM\_COUNTER\_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_Bad-NotConnected	Алгоритм имеет проблемы со связью, например, у входа алгоритма оборвана связь с переменной алгоритма-источника данных.
2150694912	OpcUa_Bad-NoCommunication	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_Bad-SensorFailure	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.
------------	--------------------------------	---

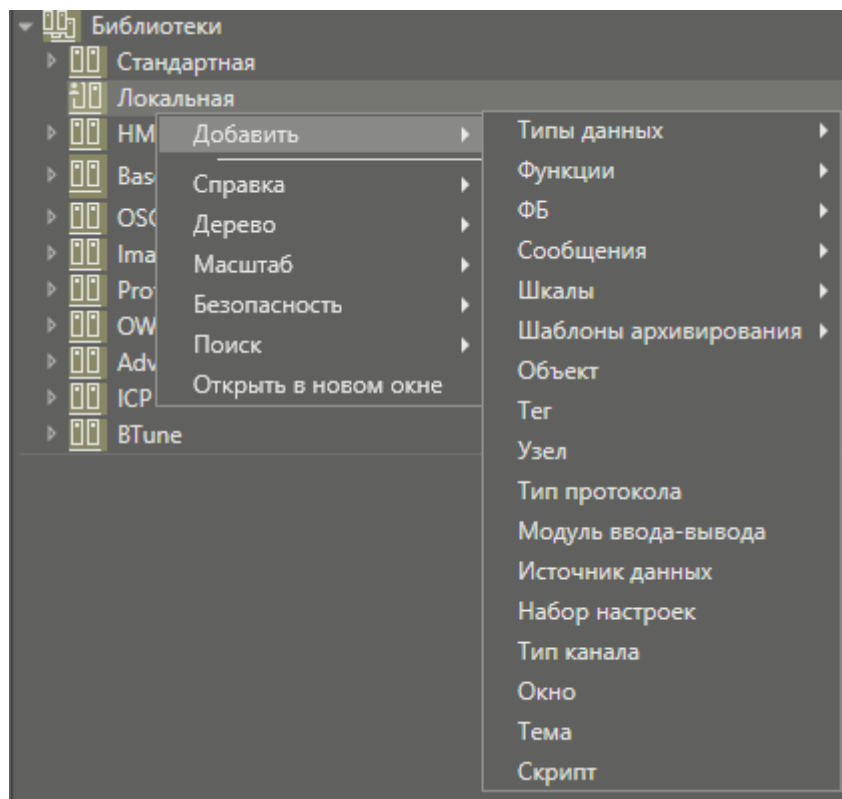
При получении данных с использованием стандарта OPC DA признаки качества преобразуются в признаки качества стандарта OPC UA согласно таблице ниже.

OPC DA	OPC UA
OPC_QUALITY_GOOD	OpCua_Good
OPC_QUALITY_LOCAL_OVERRIDE	OpCua_GoodLocalOverride
OPC_QUALITY_UNCERTAIN	OpCua_Uncertain
OPC_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.

## 6.5.2. Пользовательская библиотека

Пользовательская библиотека – это библиотека, создаваемая пользователем, и которая может содержать произвольные элементы:



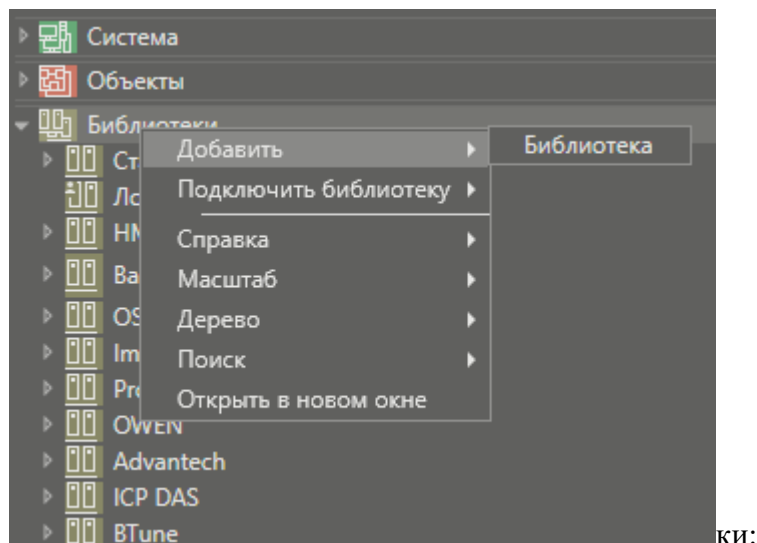
При создании проекта в нем автоматически создается пустая пользовательская библиотека Локальная.

По своей сути пользовательская библиотека - это проект определенной формы, который хранится в папке: C:\Users\Public\Documents\MasterSCADA4D\Libraries

### 6.5.2.1. Создание/подключение библиотеки

Создание библиотеки

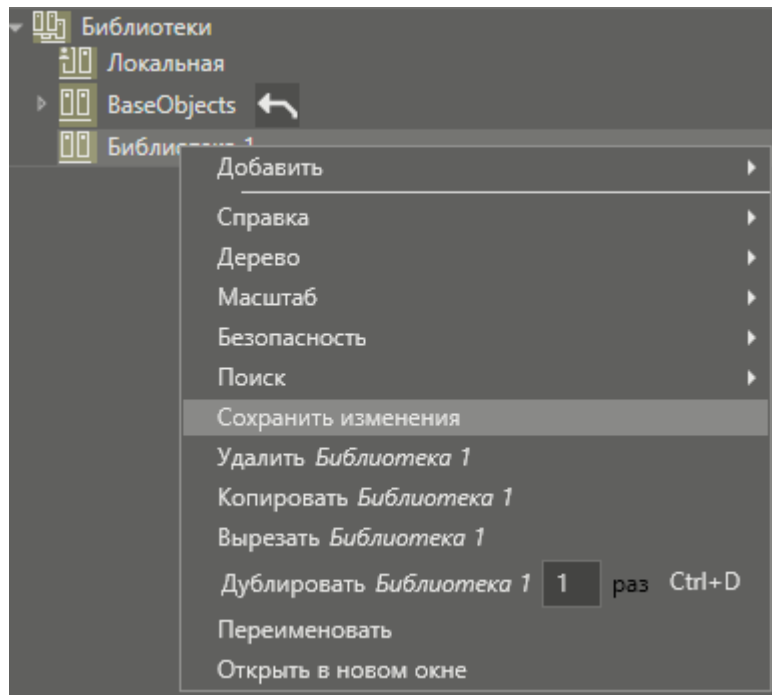
Создать пользовательскую библиотеку можно при помощи пункта Добавить контекстного меню Библиотеки:



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:



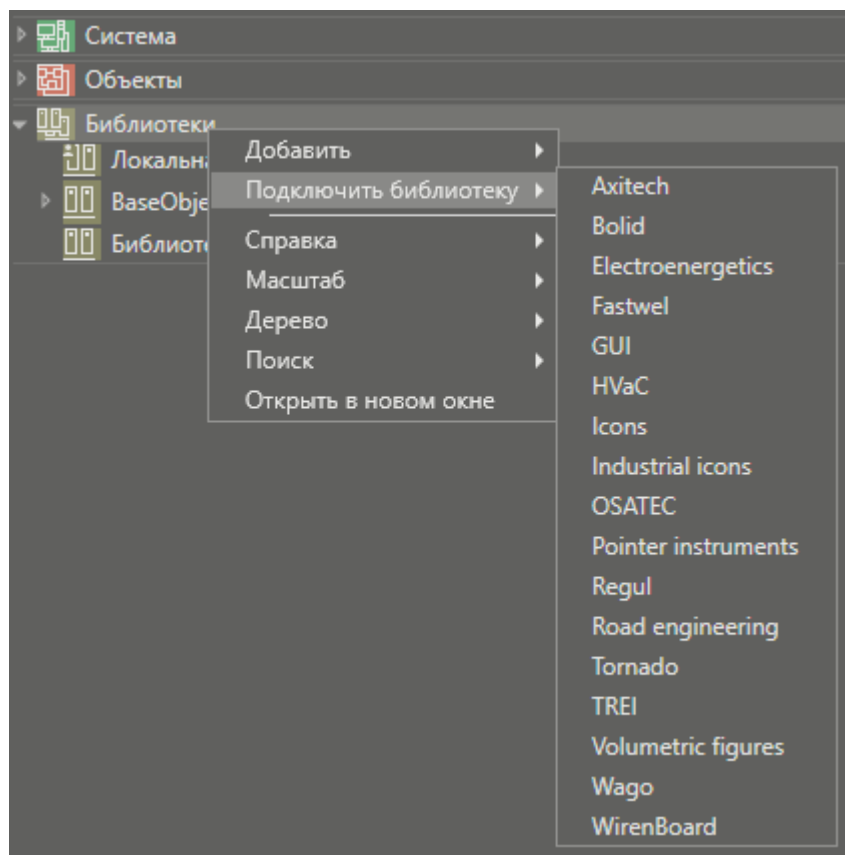
Пользовательская библиотека сохраняется вместе с зависимостями. Например, если в пользовательской библиотеке использованы функциональные блоки других библиотек, то после сохранения такой библиотеки и ее подключения к другому проекту, ранее установленные связи будут работать.

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 "ИнСАТ", 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.

### 6.5.2.2. 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.

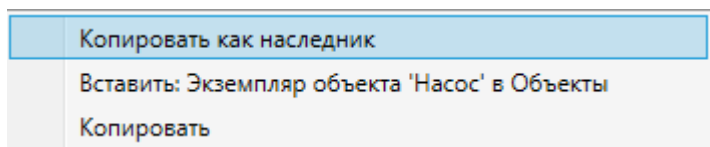
Экземпляр объекта - это элемент, который появляется путем вставки объекта из библиотеки. Имеет отношения с типом Ссылается. Экземпляр содержит в себе только элементы, необходимые для связи с другими элементами проекта. Это позволяет значительно уменьшить размер создаваемого проекта и увеличить скорость 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

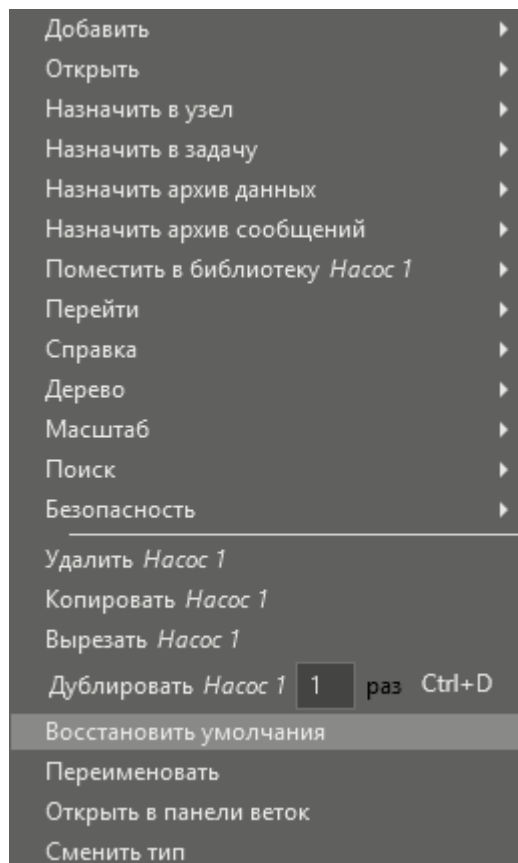
Если перетаскивать объект из библиотеки левой кнопкой мыши, то по умолчанию добавится наследник

#### 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.

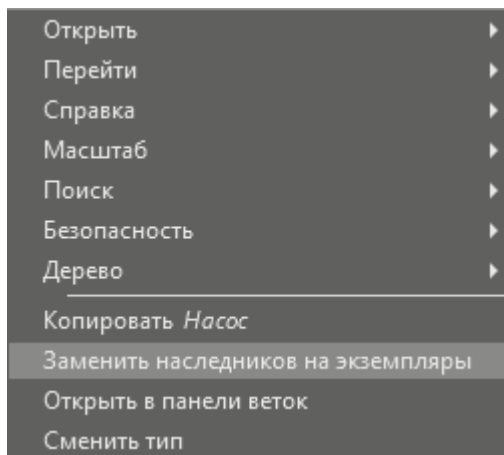
Если необходимо отменить изменения, сделанные в наследнике, то в контекстном меню наследника необходимо выполнить пункт Восстановить умолчания:



В этом случае наследник опять станет точной копией типа.

#### 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**:

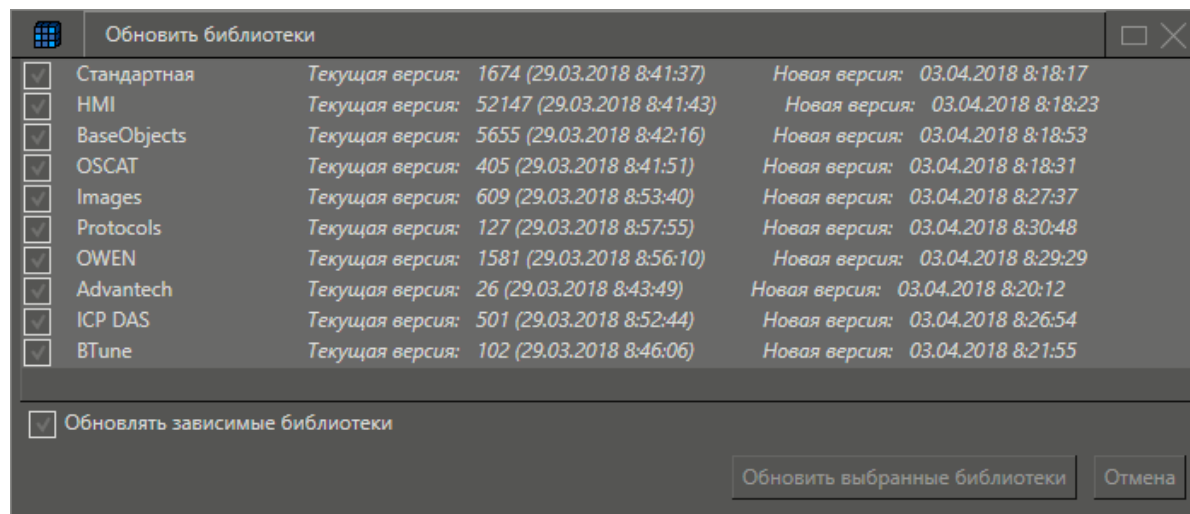


### 6.5.3. Диалог Обновить библиотеки

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.

## 6.5.4. 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 предопределенные 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.

### 6.5.4.1. 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

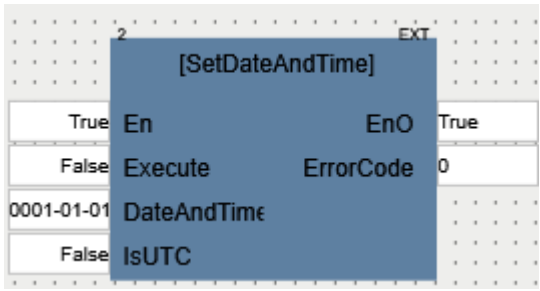
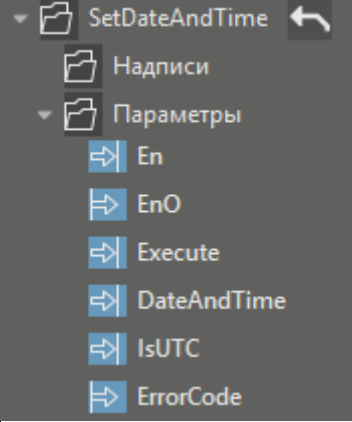
PlaySound

SEND\_SMS

ReportFB  
 GET\_CURRENT\_USERNAME  
 SysProcessCreate  
 SysProcessCreateWithResult  
 SysProcessCreateAsync  
 SysProcessCreateWithResultAsync  
 COMPort  
 COMPortByte  
 COMPortByteSync  
 WriteArchiveData

### 6.5.4.1.1. 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.
Date-AndTime	DT	Вход ФБ. Поступают устанавливаемые системная дата и время.

IsUTC	BOOL	Вход ФБ. Указывается формат времени на входе Date-AndTime (TRUE – UTC, False – локальное время).
Error-Code	DINT	Выход ФБ. Выдается информация об ошибке.

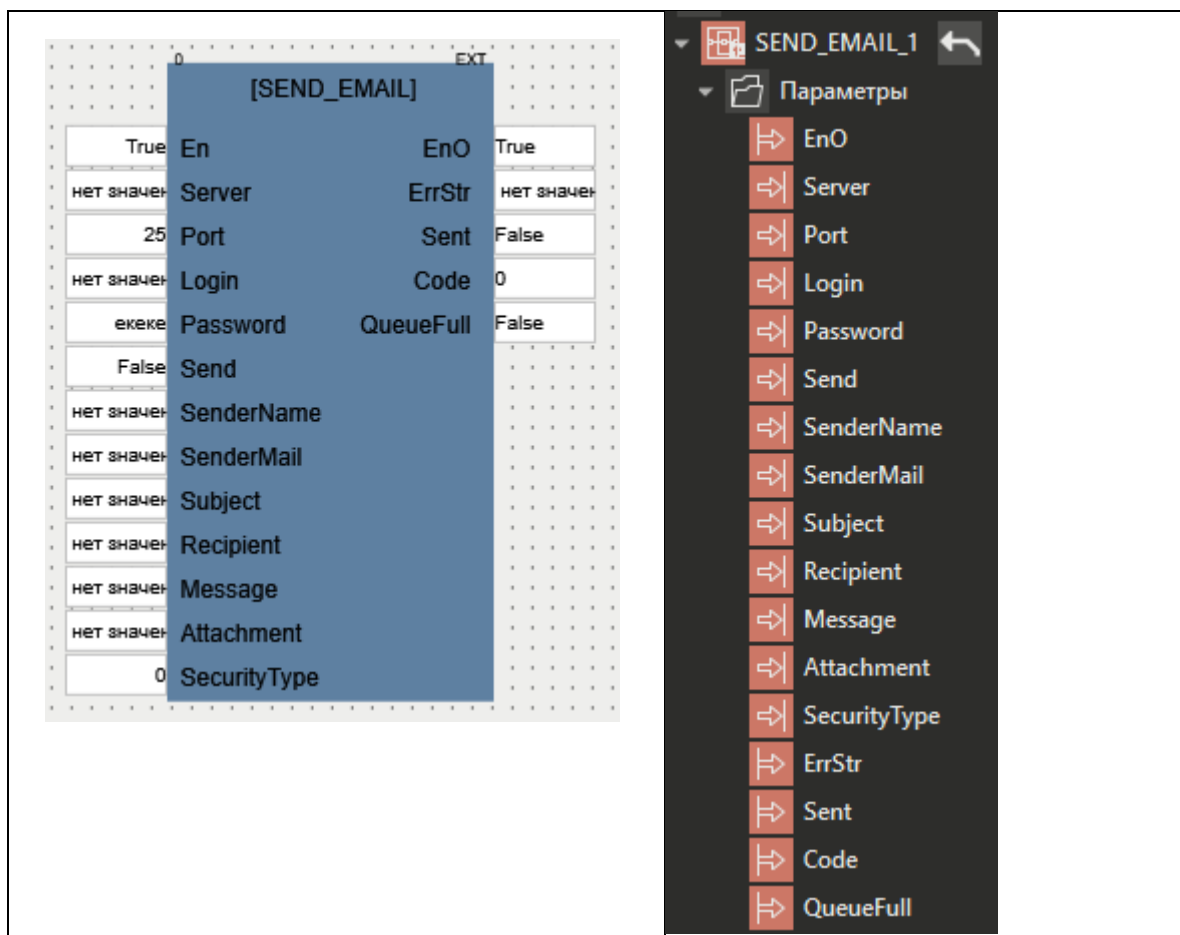
Важно! Если среда разработки запускает демоверсию среды исполнения, входящую в ее состав, автоматически, то для корректной работы данного функционального блока среда разработки должна быть запущена под правами администратора.

#### 6.5.4.1.2. 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).
Sender-Name	STRING	Имя отправителя.
Sender-Mail	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.
SecurityType	UINT	Задается тип шифрования SMTP: <ul style="list-style-type: none"> <li>• 0 - NO_SECURITY</li> <li>• 1 - USE_TLS</li> <li>• 2 - USE_SSL</li> </ul>
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.
Queue-Full	BOOL	Показывает, когда очередь на отправку сообщений переполнена.

Примеры текстов ошибок:

- Unable to initialise winsock2
- Wrong version of the winsock2
- Function send() failed
- Function recv() failed
- Function connect failed
- Unable to determine remote server
- Invalid winsock2 socket
- Function hostname() failed
- Function ioctlsocket() failed
- Improper IPv4 address
- Undefined message header
- Undefined mail sender
- Undefined message subject
- Undefined at least one recipient
- Undefined recipient mail
- Undefined user login
- Undefined user password
- Invalid user login or password
- Server returned a bad digest MD5 response
- Unable to determine server name for digest MD5 response
- Server returned error after sending MAIL FROM
- Server returned error after sending EHLO
- Server returned error after sending AUTH PLAIN
- Server returned error after sending AUTH LOGIN
- Server returned error after sending AUTH CRAM-MD5
- Server returned error after sending AUTH DIGEST-MD5
- Server returned error after sending MD5 DIGEST
- Server returned error after sending DATA
- Server returned error after sending QUIT
- Server returned error after sending RCPT TO
- Error in message body

- Server has closed the connection
- Server is not ready
- Server not responding
- Attachment file does not exist
- Message is too big
- Bad login or password
- Undefined xyz SMTP response
- Lack of memory
- time() error
- RecvBuf is empty
- SendBuf is empty
- Specified line number is out of message size
- Server returned error after sending STARTTLS
- SSL problem
- Failed to send data block
- The STARTTLS command is not supported by the server
- AUTH LOGIN is not supported by the server
- Undefined error id

Примеры кодов:

- 0 CSMTP\_NO\_ERROR
- 100 WSA\_STARTUP
- 101 WSA\_VER
- 102 WSA\_SEND
- 103 WSA\_RECV
- 104 WSA\_CONNECT
- 105 WSA\_GETHOSTBY\_NAME\_ADDR
- 106 WSA\_INVALID\_SOCKET
- 107 WSA\_HOSTNAME
- 108 WSA\_IOCTL\_SOCKET
- 109 WSA\_SELECT
- 110 BAD\_IPV4\_ADDR

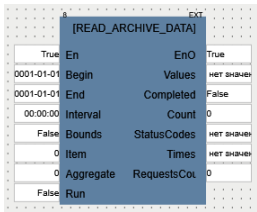
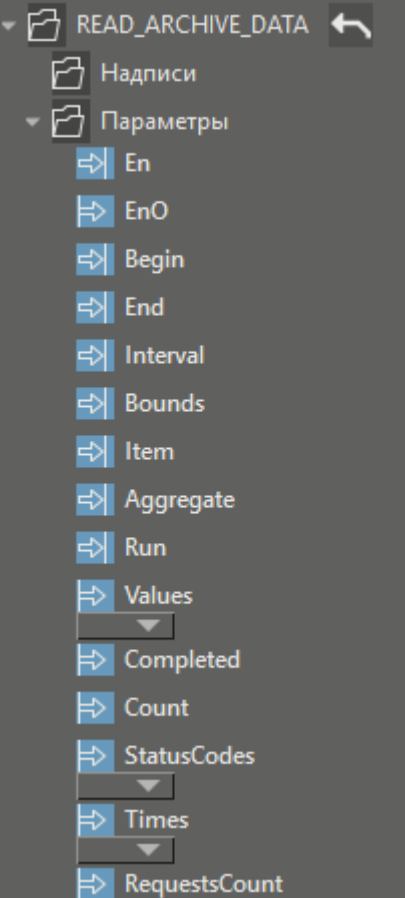
- 200 UNDEF\_MSG\_HEADER
- 201 UNDEF\_MAIL\_FROM
- 202 UNDEF\_SUBJECT
- 203 UNDEF\_RECIPIENTS
- 204 UNDEF\_LOGIN
- 205 UNDEF\_PASSWORD
- 206 BAD\_LOGIN\_PASSWORD
- 207 BAD\_DIGEST\_RESPONSE
- 208 BAD\_SERVER\_NAME
- 209 UNDEF\_RECIPIENT\_MAIL
- 300 COMMAND\_MAIL\_FROM
- 301 COMMAND\_EHLO
- 302 COMMAND\_AUTH\_PLAIN
- 303 COMMAND\_AUTH\_LOGIN
- 304 COMMAND\_AUTH\_CRAMMD5
- 305 COMMAND\_AUTH\_DIGESTMD5
- 306 COMMAND\_DIGESTMD5
- 307 COMMAND\_DATA
- 308 COMMAND\_QUIT
- 309 COMMAND\_RCPT\_TO
- 310 MSG\_BODY\_ERROR
- 400 CONNECTION\_CLOSED
- 401 SERVER\_NOT\_READY
- 402 SERVER\_NOT\_RESPONDING
- 403 SELECT\_TIMEOUT
- 404 FILE\_NOT\_EXIST
- 405 MSG\_TOO\_BIG
- 406 BAD\_LOGIN\_PASS
- 407 UNDEF\_XYZ\_RESPONSE
- 408 LACK\_OF\_MEMORY
- 409 TIME\_ERROR



- 410 RECVBUF\_IS\_EMPTY
- 411 SENDBUF\_IS\_EMPTY
- 412 OUT\_OF\_MSG\_RANGE
- 413 COMMAND\_EHLO\_STARTTLS
- 414 SSL\_PROBLEM
- 415 COMMAND\_DATABLOCK
- 416 STARTTLS\_NOT\_SUPPORTED
- 417 LOGIN\_NOT\_SUPPORTED

### 6.5.4.1.3. 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
<b>Входы</b>		
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.
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.
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.
<b>Выходы</b>		
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.
Status-Codes	ARRAY [*] OF DINT	Массив признаков качества.

Times	ARRAY [*] OF DATE_A ND_TIM E	Массив меток времени.
Request- sCount	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:

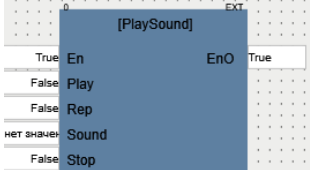
```

READ_ARCHIVE_DATA_1();
if (READ_ARCHIVE_DATA_1.Completed) then
MyRes := READ_ARCHIVE_DATA_1.Values
end_if

```

#### 6.5.4.1.4. 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
	<b>PlaySound_1</b> (Play:=Включить_звук, Rep:=TRUE, Sound:="mp3.MP3"); //(где: Включить_звук - это логический параметр программы ST, связанный с параметром, который отвечает за запуск воспроизведения).

**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)/
Stop	BOOL	Вход ФБ. Останавливает воспроизведение/

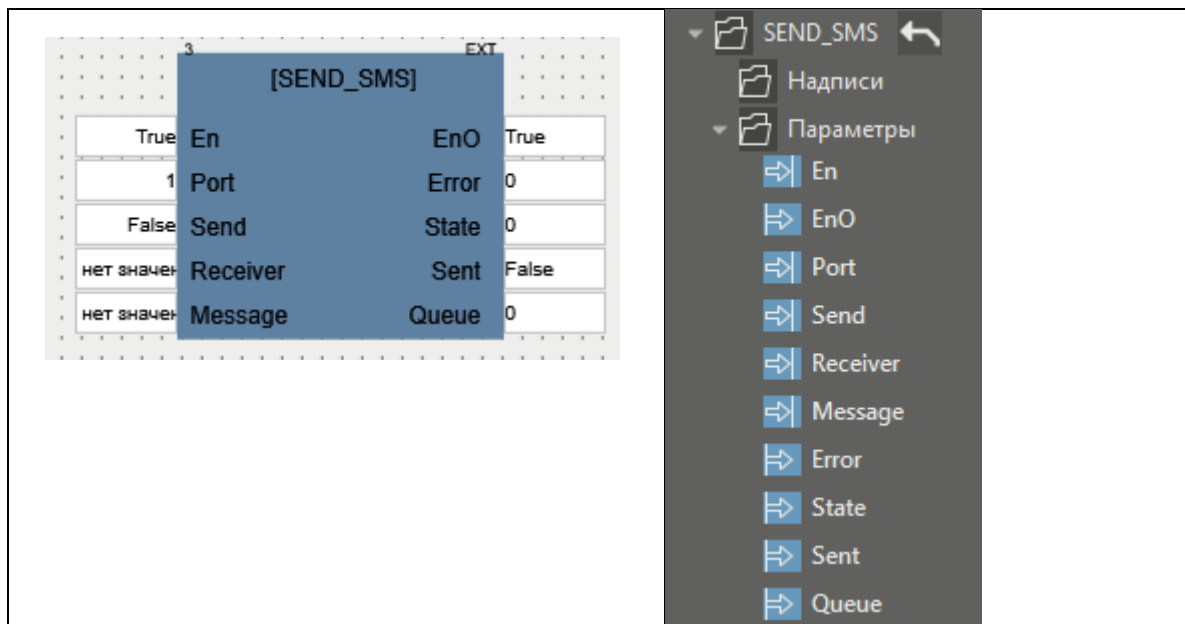
Важно! Расширение файла необходимо указывать заглавными (прописными) буквами.

Важно! Перед использованием аудиофайла его необходимо добавить в Элемент Медиа.

#### 6.5.4.1.5. 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
---------------------	--------------



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.

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.

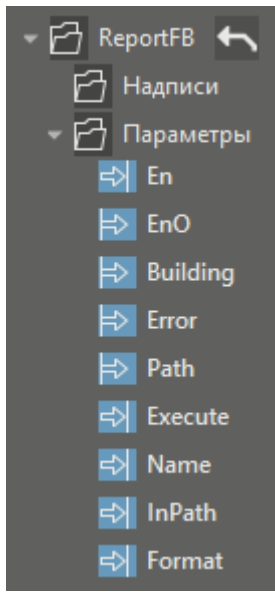
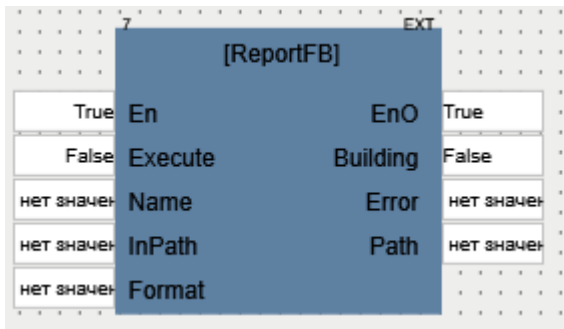
Коды состояний:

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

### 6.5.4.1.6. 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: 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. <b>Важно!</b> Вы можете указать произвольную папку на диске, в случае если собираетесь для просмотра отчетов использовать внешние по отношению к продуктам компании ИнСАТ 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 \htdocs проекта, загруженного в среду исполнения.</p> <p>Если этот параметр не задан, то по умолчанию используется папка &lt;папка узла&gt;\htdocs\ (например,</p>



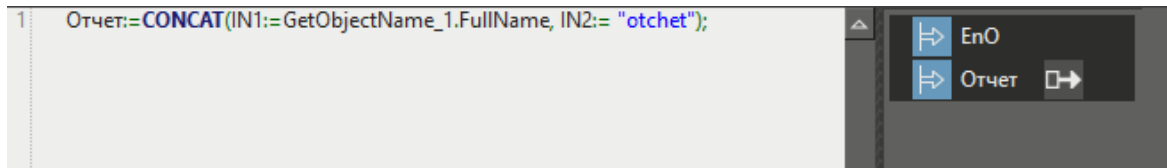
		если узел запускается локально из среды разработки, то используется папка <имя пользователя>\AppData\Roaming\InSAT\MasterSCADA4D<версия продукта>\Debug_<имя проекта>\<имя узла>\PLC\htdocs, а если узел запускается из папки запуска проекта по умолчанию среды исполнения для Windows, то используется папка c:\Users\<имя пользователя>\AppData\Roaming\InSAT\MasterSCADA4DRT<версия продукта>\Server\htdocs);
Format	STRING	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.
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	Указывает путь к созданному файлу отчета. Этот путь можно использовать при создании проекта для открытия отчета в контейнере окна, копирования, печати и др. 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: <транслитерация имени отчета в проекте>_YYYY_mm_DD_HH_MM_SS.<значение входа Format>.

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.

### 6.5.4.1.7. GET\_CURRENT\_USERNAME

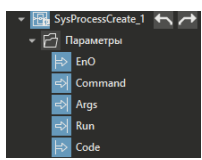
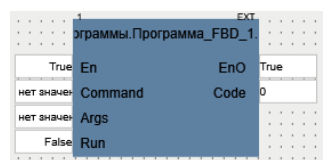
Функциональный блок GET\_CURRENT\_USERNAME выдает информацию о пользователе, который авторизовался в клиенте, подключенном к среде исполнения. У сервера могут быть одновременно несколько клиентов, и на каждом клиенте авторизован свой пользователь, поэтому функциональный блок должен работать в задаче 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, исполняется 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

#### 6.5.4.1.8. 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
		<b>SysProcessCreate_1</b> (Command:="chrome.exe", Args:="http://127.0.0.1:8043",Run:=Запустить); //(где: Запустить - это логический параметр, связанный с кнопкой без фиксации в окне).

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

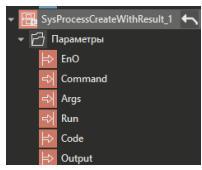
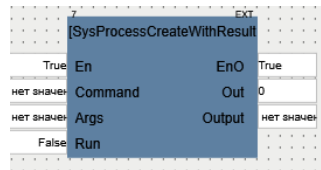
Для запуска GUI-приложений (например, Firefox) требуется указывать имя пользователя ОС, который работает в текущий момент. Для этого необходимо использовать команду su.

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.)

#### 6.5.4.1.9. 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(Com- mand:="chrome.exe", Args:="http://127.0.0.1:8043",Run:=Запу- стить); //(где: Запустить - это логиче- ский 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 iexplorer.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 исполнятся 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.

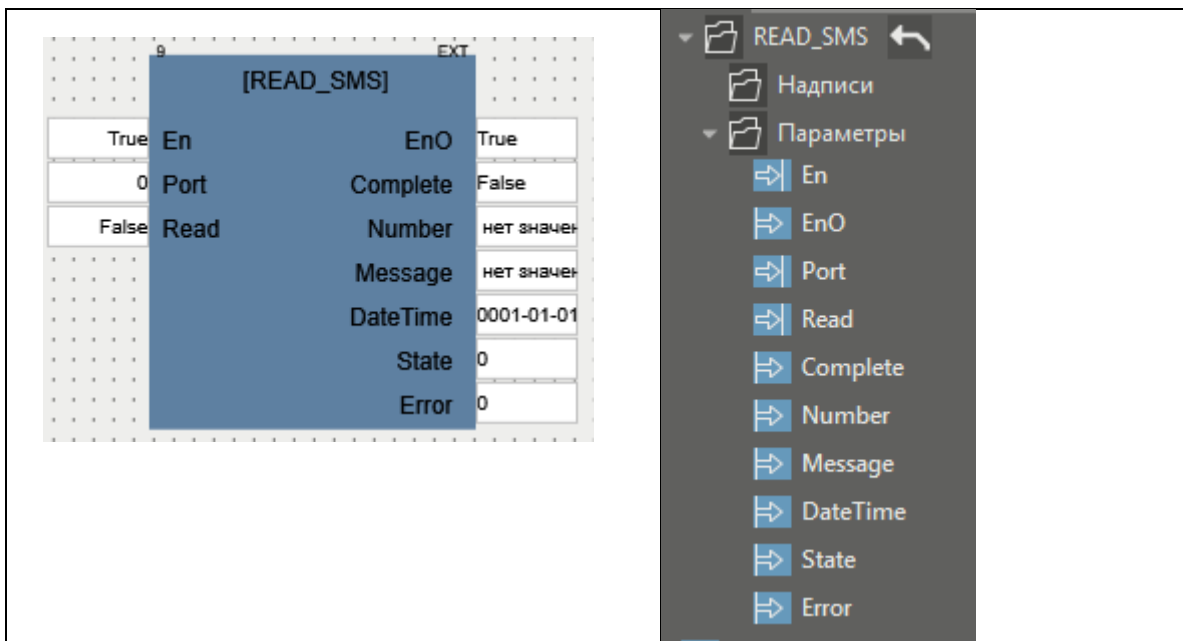
В этом случае на вход Command необходимо подать su - [имя пользователя] [имя приложения], либо на вход Command подать su - [имя пользователя], а на вход Args - [имя приложения].

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).

#### 6.5.4.1.10. 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.

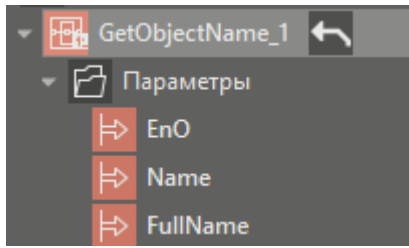
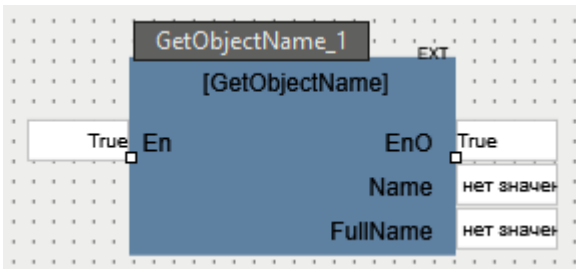
Входы и выходы ФБ:

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	Текущее состояние ФБ: <ul style="list-style-type: none"> <li>• 0 - Ожидание инициализации</li> <li>• 1 - Ожидание команды</li> <li>• 2 - Начало чтения</li> <li>• 33 - Чтения 1 SMS</li> <li>• 34 - Удаление прочитанного SMS</li> <li>• 16- Инициализация отправки с модема</li> <li>• 17 - Начало отправки SMS</li> <li>• 18 - Отправка SMS</li> </ul>
Error	DINT	Номер ошибки: <ul style="list-style-type: none"> <li>• 0 - Нет ошибок</li> <li>• 1 - Нет подключения к модему</li> <li>• 5 - Ошибка разбора PDU</li> <li>• 6 - Превышен лимит ожидания от модема</li> </ul>

### 6.5.4.1.11. GetObjectName

Функциональный блок GetObjectName выдает информацию об объекте (или экземпляре объекта), в котором он сам находится.

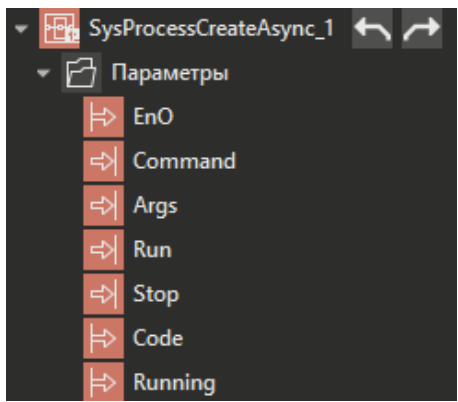
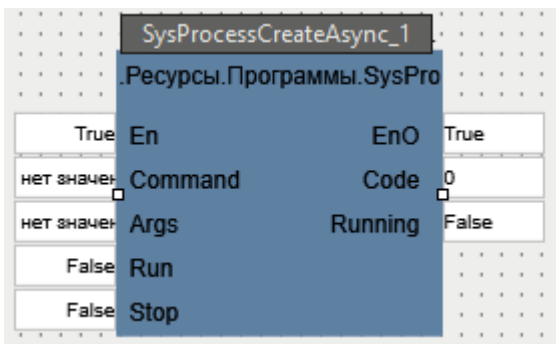
Вид в дереве		Вид в редакторе FBD	
			
Название	Тип	Назначение	
Name	STRING	Выход ФБ. Отображает Имя объекта. Например, Насосная.	



FullName	STRING	Выход ФБ. Отображается Полное имя объекта. Например, Объекты.Завод.Инженерные сооружения.Насосная.
----------	--------	--

### 6.5.4.1.12. 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 исполнительской системы. Например: ping -t, telnet и др.

Вид в дереве	Вид в редакторе FBD
	

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 iexplorer.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

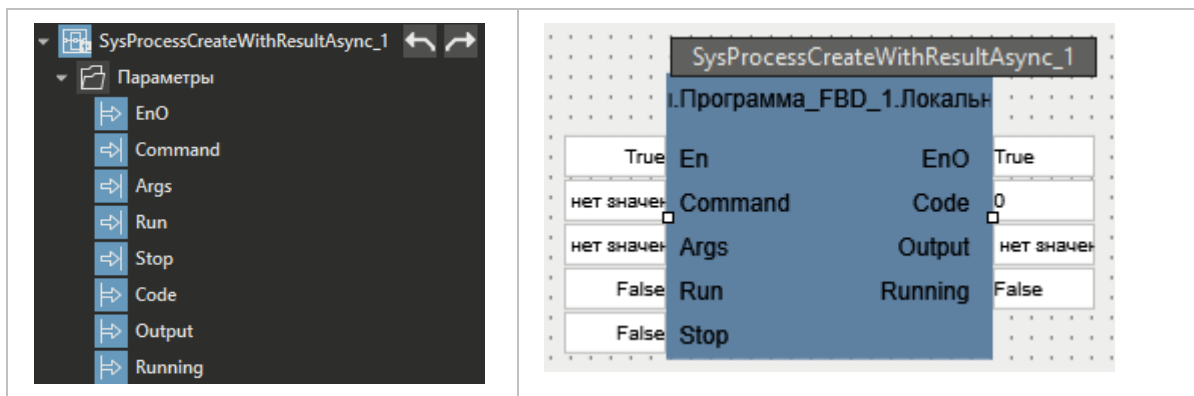
Для запуска GUI-приложений (например, Firefox) требуется указывать имя пользователя ОС, который работает в текущий момент. Для этого необходимо использовать команду `su`.

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]`.

#### 6.5.4.1.13. 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
<b>Входы</b>		
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.
Stop	BOOL	Останавливает выполнение процесса.
<b>Выходы</b>		
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`.

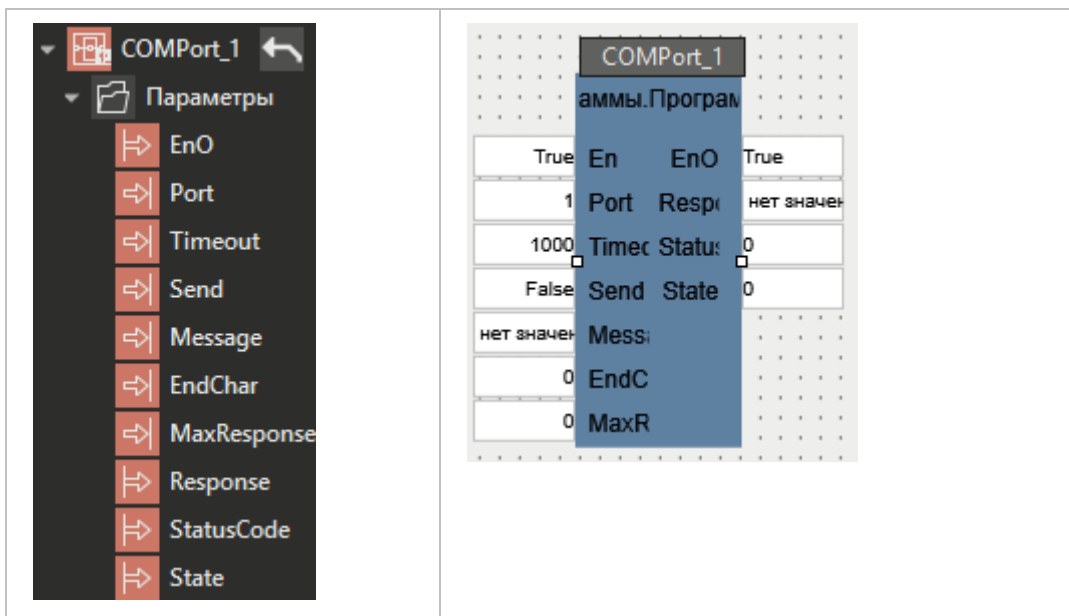
В этом случае на вход `Command` необходимо подать `su - [имя пользователя] [имя приложения]`, либо на вход `Command` подать `su - [имя пользователя]`, а на вход `Args` - `[имя приложения]`.

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).

#### 6.5.4.1.14. 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.

Вид в дереве	Вид в редакторе FBD
--------------	---------------------



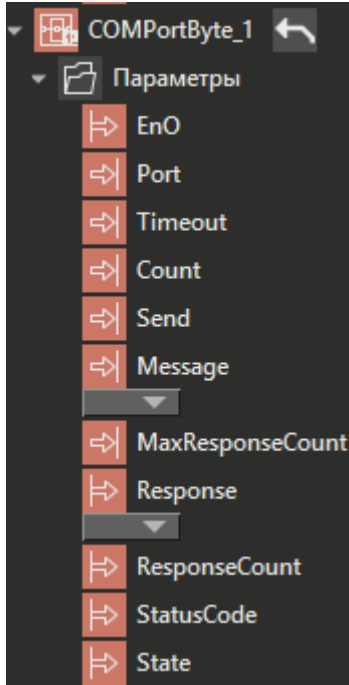
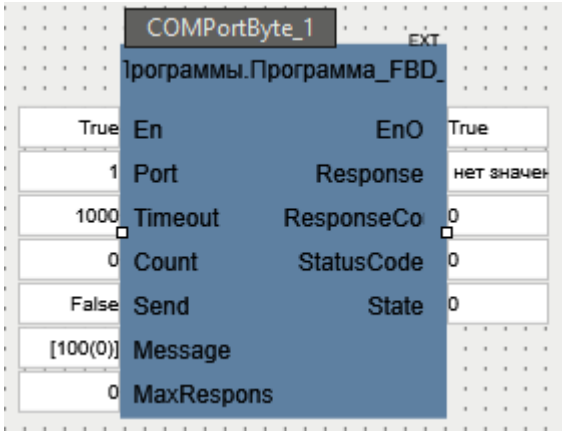
## Входы и выходы ФБ:

Название	Тип	Назначение
Входы		
Port	DINT	Задается номер COM-порта устройства.
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.
Выходы		
Response	STRING	Выдается ответ

Status-Code	DINT	Выдается код ответа: <ul style="list-style-type: none"> <li>• 5 — ответ получен по истечению Timeout;</li> <li>• 0 — ответ получен по достижению завершающего символа (или по достижению MaxResponse символов в ответе)</li> </ul>
State	DINT	Показывает состояние ФБ. <ul style="list-style-type: none"> <li>• 0 — ожидание регистрации порта;</li> <li>• 1 — ожидание отправки запроса;</li> <li>• 2 — отправка запроса.</li> </ul>

### 6.5.4.1.15. 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.

Вид в дереве	Вид в редакторе FBD
	

Входы и выходы ФБ:

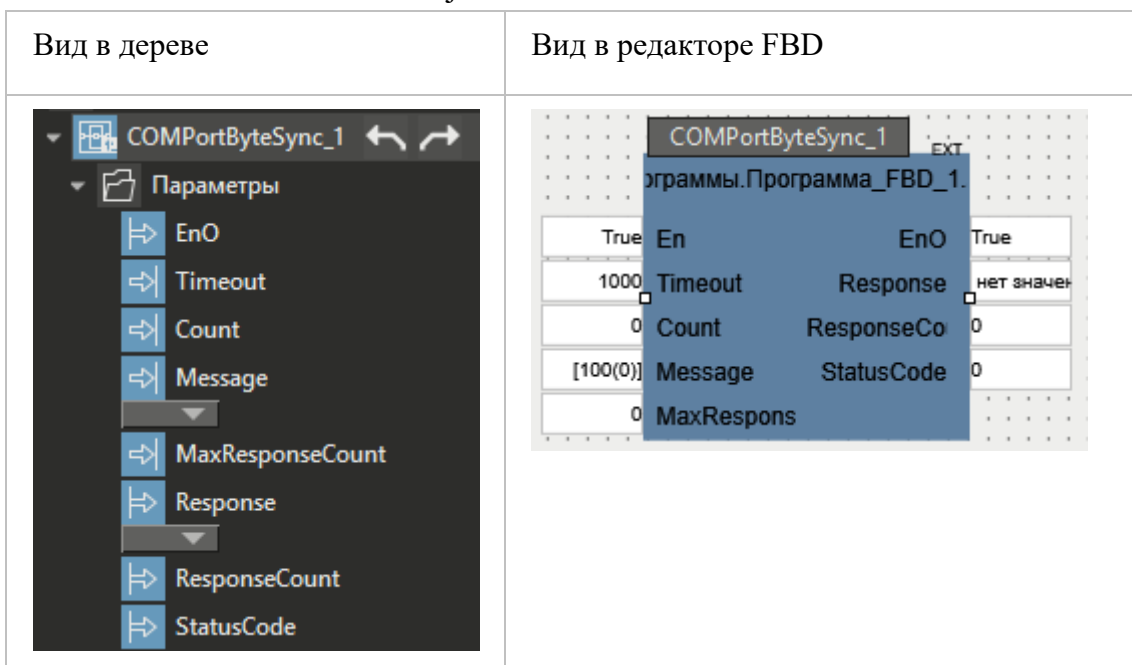
Название	Тип	Назначение
Входы		
Port	DINT	Задается номер COM-порта устройства.
Timeout	DINT	Задается время ожидания. По умолчанию установлено значение 1000. Если данный параметр не требуется, то необходимо установить значение 0.
Count	INT	Задается количество байт, которое необходимо передать.
Send	BOOL	Отправляет запрос по переднему фронту изменения значения на этом входе.
Message	ARRAY [1..100] of BYTE	Задается запрос.
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.
Status-Code	DINT	Выдается код ответа: <ul style="list-style-type: none"> <li>• 5 — ответ получен по истечению Timeout;</li> <li>• 0 — ответ получен по достижению количества символов в ответе, заданном в MaxResponseCount.</li> </ul>
State	DINT	Показывает состояние ФБ. <ul style="list-style-type: none"> <li>• 0 — ожидание регистрации порта;</li> </ul>

- 1 — ожидание отправки запроса;
- 2 — отправка запроса.

### 6.5.4.1.16. 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 от отправку сообщения и ожидание получения ответа.

Находится в библиотеке BaseObjects.



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.
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.
Status-Code	DINT	Выдается код ответа: <ul style="list-style-type: none"> <li>• 5 — ответ получен по истечению Timeout;</li> <li>• 0 — ответ получен по достижению количества символов в ответе, заданном в MaxResponseCount.</li> </ul>

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.

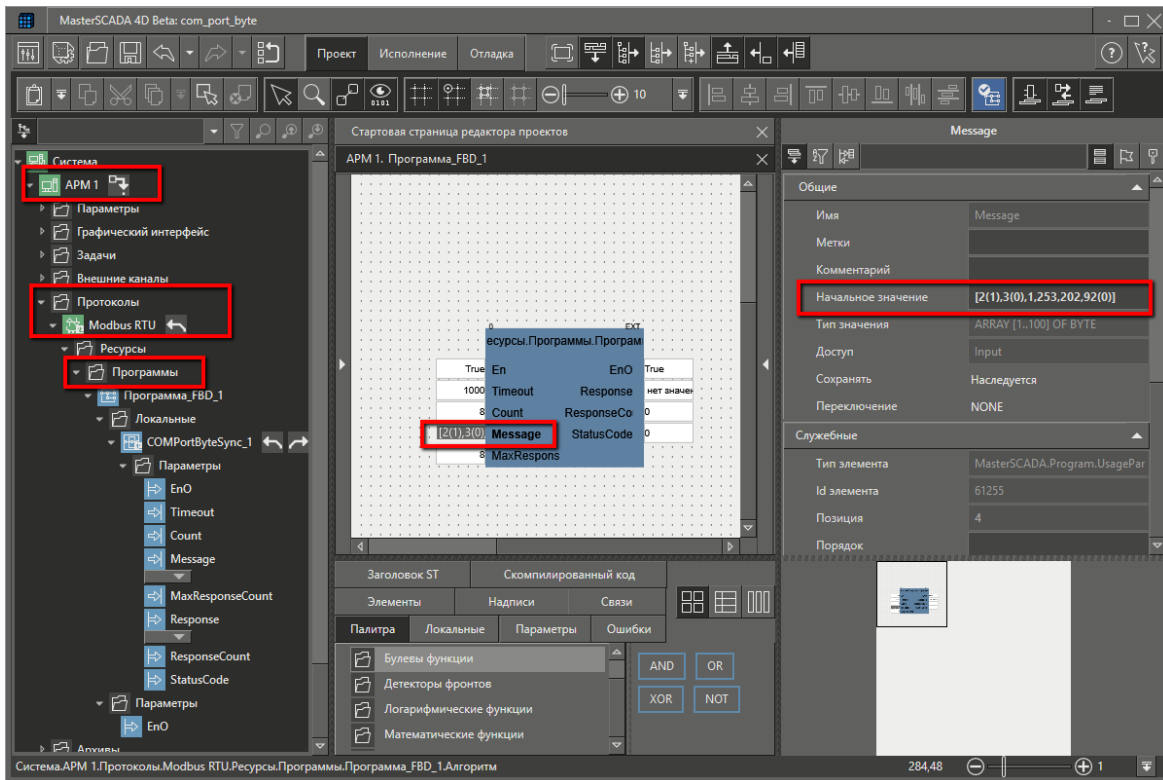
У данного ФБ отсутствует вход SEND, имеющийся у ФБ COMPortByte, т.к. ФБ COMPortByteSync выполняется синхронно, т. е. отслеживания выполнения не требуется. В случае если отправку производить не нужно, то стандартному входу En следует присвоить значение False.

#### Порядок работы

1. Добавить в узел какой-либо протокол, например, Modbus RTU.
2. В настройках протокола установить номер COM-порта.
3. Добавить в протокол в какую-либо Программу.

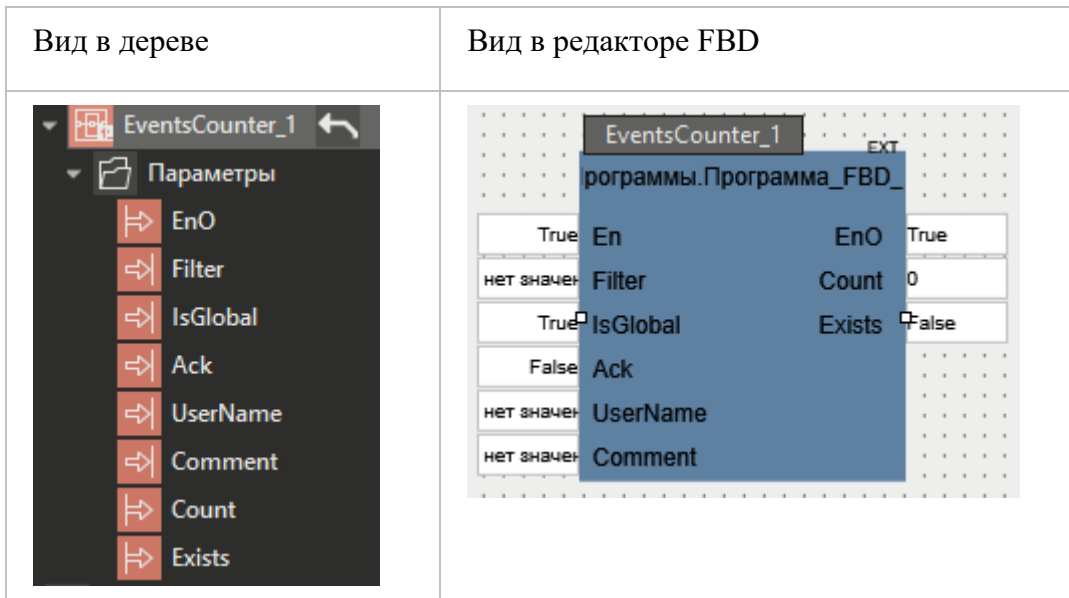
4. Добавить в программу нужные ФБ.

5. Установить связи с параметрами проекта или установить константы на входах ФБ.



### 6.5.4.1.17. EventsCounter

Функциональный блок EventsCounter служит для подсчета количества сообщений, удовлетворяющих заданному фильтру. Позволяет также квитировать сообщения.

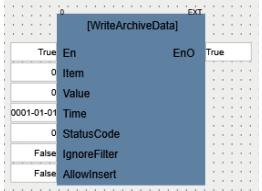
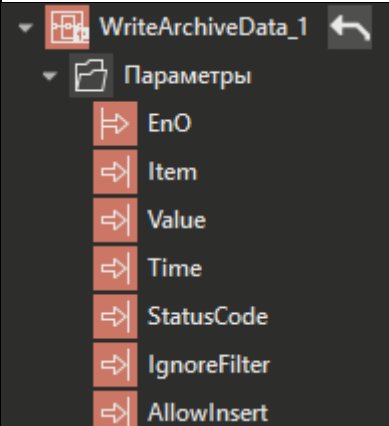


## Входы и выходы ФБ:

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 Окно настройки журнала. Столбцы таблицы.
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.

**6.5.4.1.18. 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
<b>Входы</b>		
Item	REF_TO	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. <b>Важно!</b> Если ФБ находится внутри программы, например, FBD или ST, то в программу необходимо добавить вход типа REF_TO, связать его с архивируемым параметром, а затем уже этот вход программы связать со входом ФБ Item
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.

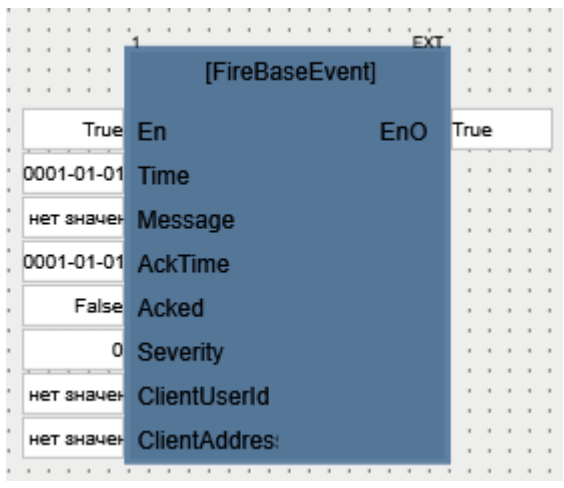
IgnoreFilter	BOOL	Позволяет игнорировать настройки фильтра архивирования (то есть писать даже в случае если значение не изменилось).
AllowInsert	BOOL	Позволяет записать в архив значение даже если его метка времени не является последней из всех меток по данному параметру. В этом случае, при записи сбрасывается 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.

#### 6.5.4.1.19. 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

Вид в дереве	Вид в редакторе FBD
	

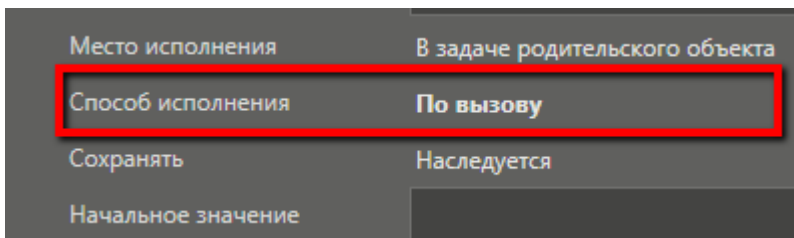
Входы и выходы ФБ:

Название	Тип	Назначение
Входы		
Time	DT	Задается время возникновения сообщения.
Message	STRING	Задается текст сообщения.
AckTime	DT	Задается время квитирования.
Acked	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.

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 добавления пользователя не произойдет, то выход примет значение False.

#### 6.5.4.2. BaseObjects.ФБ.Users

Данная группа содержит функциональные блоки, предназначенные для работы с пользователями проекта в режиме исполнения.

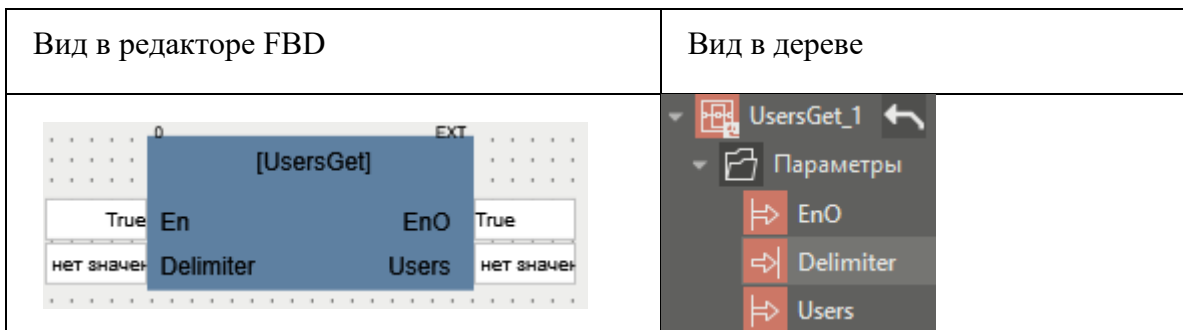
Список функциональных блоков:

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
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

#### 6.5.4.2.1. UsersGet

Функциональный блок UsersGet служит для получения списка пользователей, добавленных в режиме исполнения.

Важно! Данный ФБ не выдает информацию о пользователях, добавленных в режиме разработки

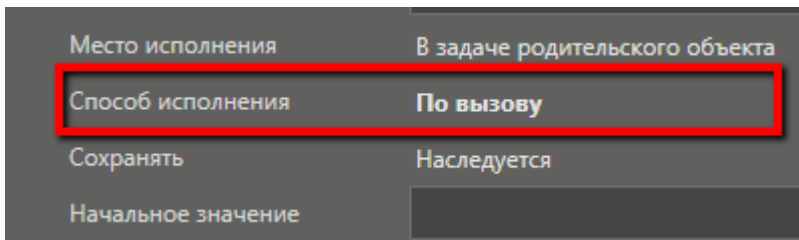


Входы и выходы ФБ:

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 исполняется данный ФБ

Рекомендации по работе с ФБ

В панели свойств ФБ в категории Общие рекомендуется установить свойство Способ исполнения в состояние По вызову:





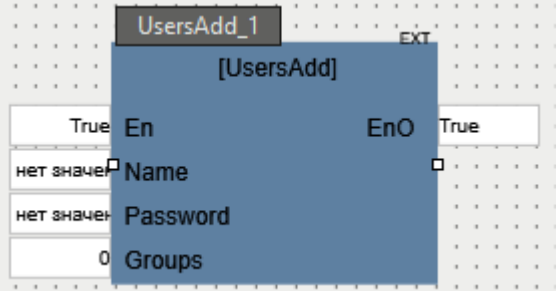
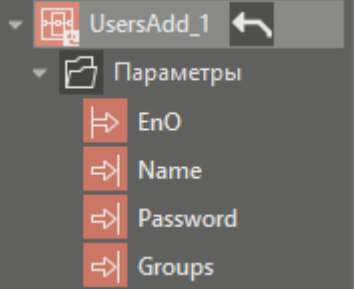
Как правило, для получения списка пользователей, в окне создают Кнопку. Затем в событие Клик мыши этой кнопки добавляют действие Вызвать программу (для этого функциональный блок следует перетащить на вход действия Программа).

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.

### 6.5.4.2.2. 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

Вид в редакторе FBD	Вид в дереве
	

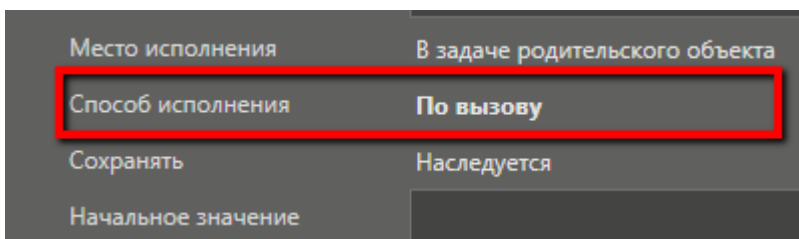
Входы и выходы ФБ:

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.

	<p>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.</p>
--	--

### 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 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.

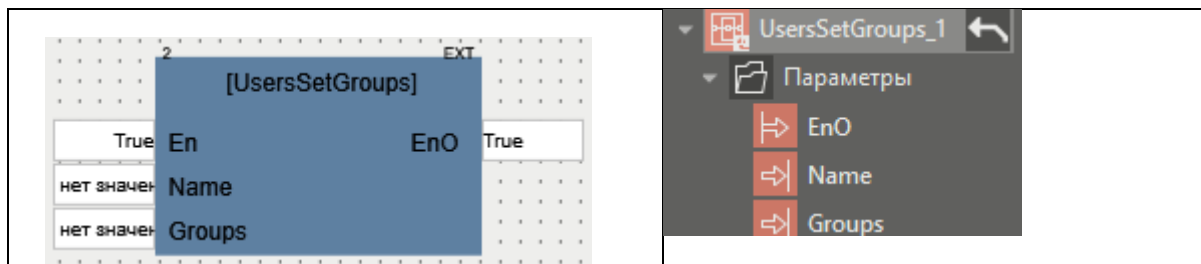
### 6.5.4.2.3. UsersSetGroups

Функциональный блок UsersSetGroups служит для назначения/изменения списка ролей пользователя, ранее созданного 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

Вид в редакторе FBD	Вид в дереве
---------------------	--------------

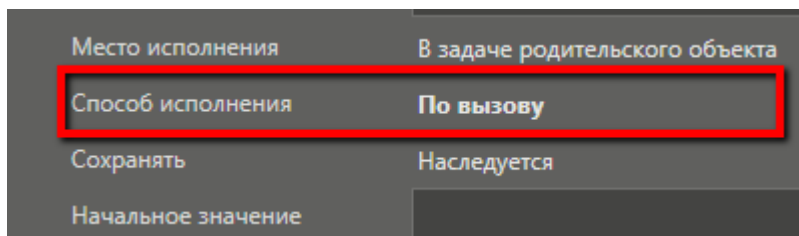


#### Входы и выходы ФБ:

Название	Тип	Назначение
Name	STRING	Вход ФБ. Задается имя пользователя, роль которого требуется изменить. Список добавленных пользователей можно получить, используя ФБ UsersGet.
Groups	STRING	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 UsersGet-Groups.

#### 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.


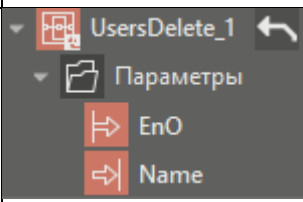
О том, была ли работа ФБ успешной, можно судить по выходу ФБ EnO. Если добавления не произойдет, то выход примет значение False

#### 6.5.4.2.4. 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

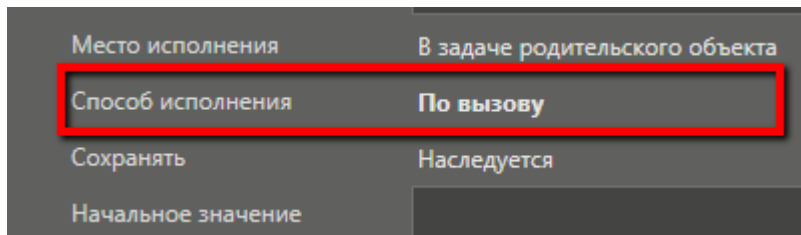
Вид в редакторе FBD	Вид в дереве
	

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:



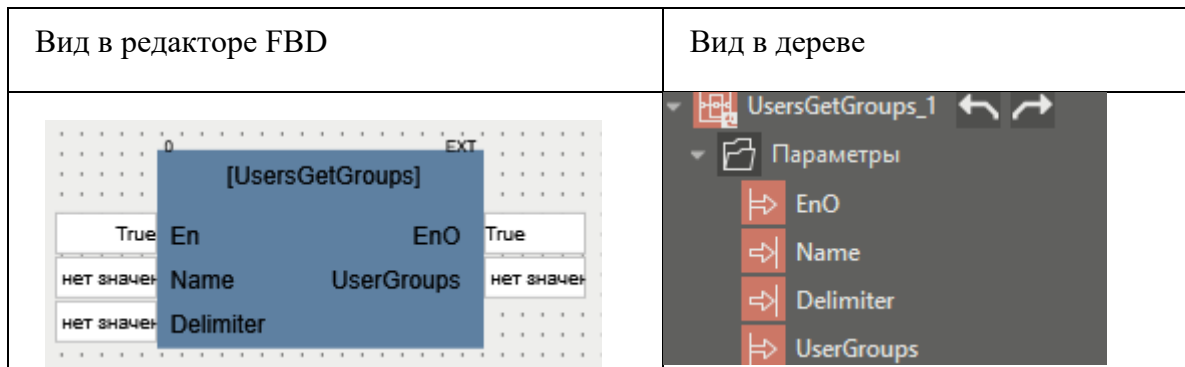
Как правило, удаление пользователя происходит после нажатия на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания функционального блока на вход действия Программа.

О том, была ли работа ФБ успешной, можно судить по выходу ФБ EnO. Если удаления не произойдет, то этот выход примет значение False.

### 6.5.4.2.5. UsersGetGroups

Функциональный блок UsersGetGroups служит для получения списка ролей, в которые добавлен определенный пользователь.

**Важно!** Данный ФБ не выдает информацию о пользователях, добавленных в режиме разработки.



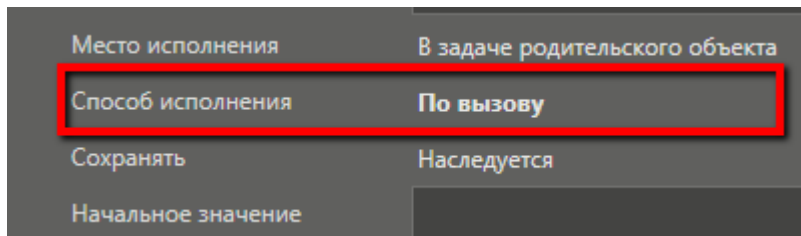
Входы и выходы ФБ:

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.

### 6.5.4.2.6. UsersSetPassword

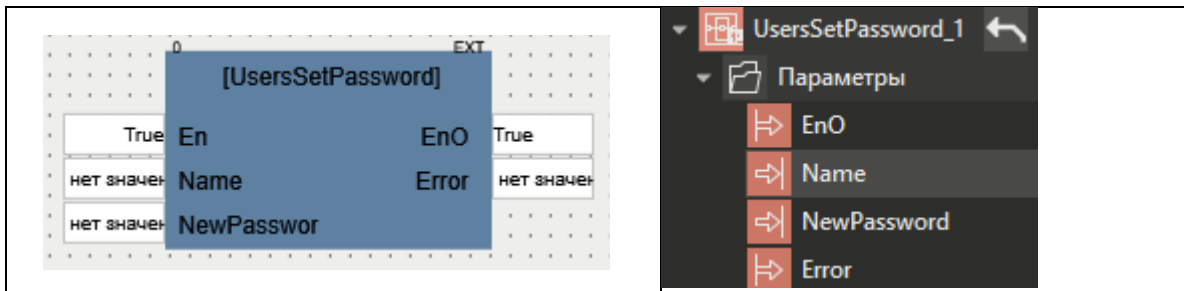
Функциональный блок UsersSetPassword служит для назначения/изменения пароля пользователя, ранее добавленного в режиме исполнения.

**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

Вид в редакторе FBD

Вид в дереве

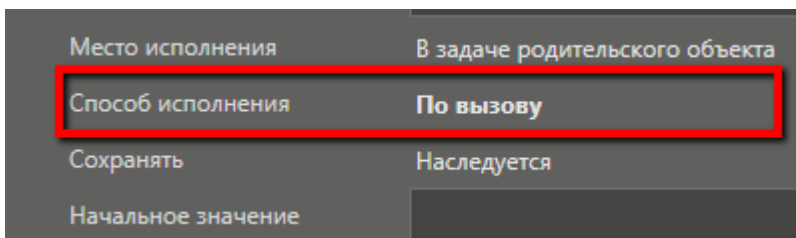


Входы и выходы ФБ:

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.

### 6.5.4.2.7. 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 операторов

Вид в редакторе FBD	Вид в дереве

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 добавляют действие Вызвать программу путем перетаскивания функционального блока на вход действия Программа.

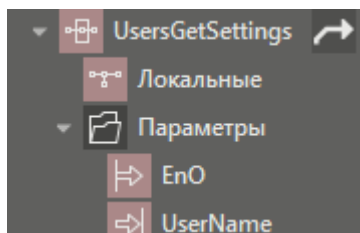
Для того, чтобы не ошибиться с написанием имени пользователя, можно вход Name соединить со свойством Выбор графического элемента Выпадающий список, у которого список строк формируется с использованием ФБ UsersGet.

О том, была ли работа ФБ успешной, можно судить по выходу ФБ EnO. если переименования не произойдет, то выход примет значение False

### 6.5.4.2.8. 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

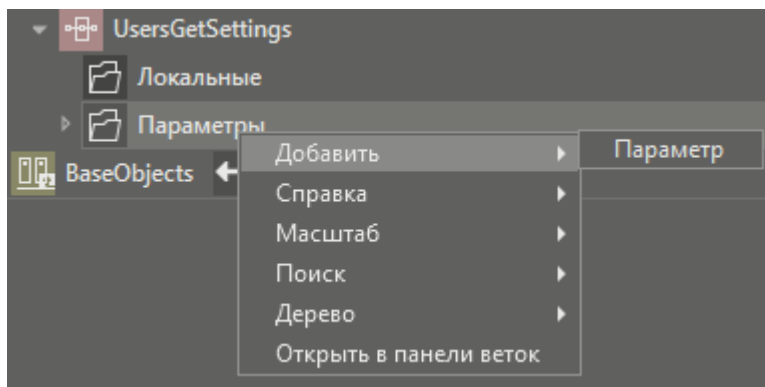


Входы и выходы ФБ:

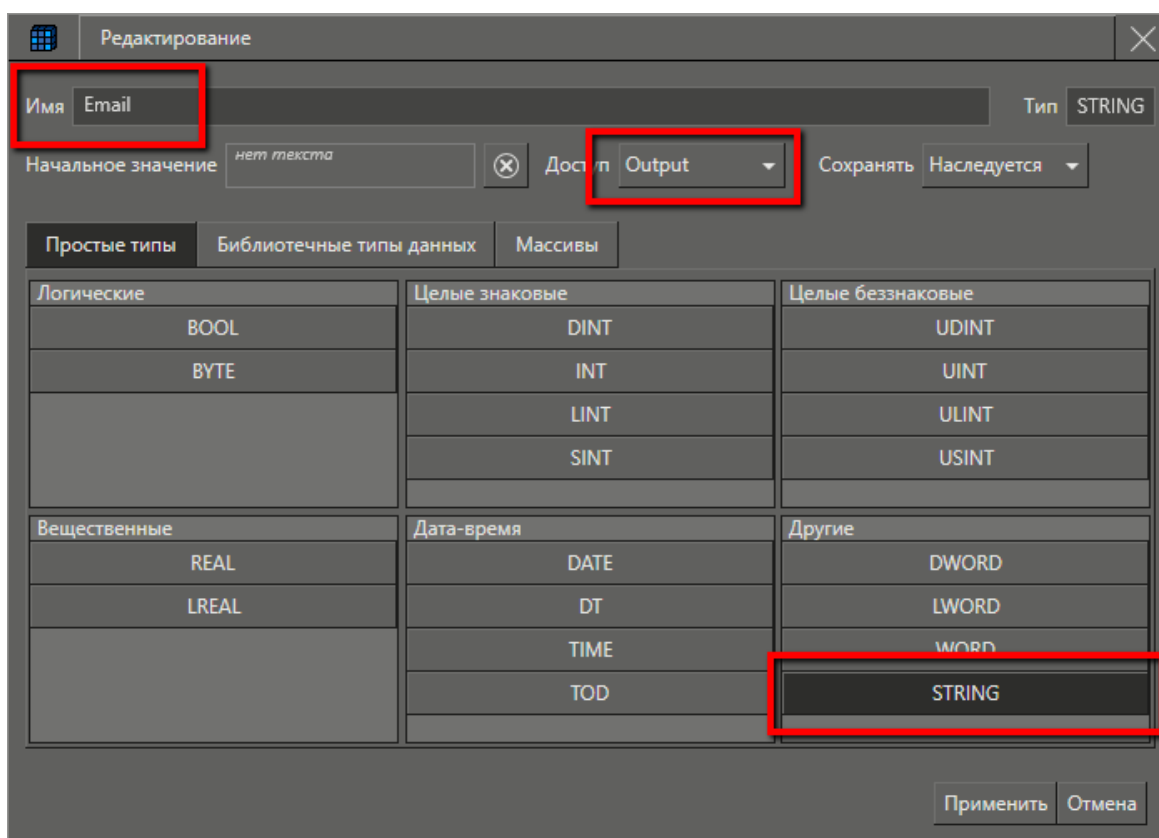
Название	Тип	Назначение
UserName	STRING	Задается имя пользователя, по которому требуется получить значения дополнительных настроек

Рекомендации по работе с ФБ

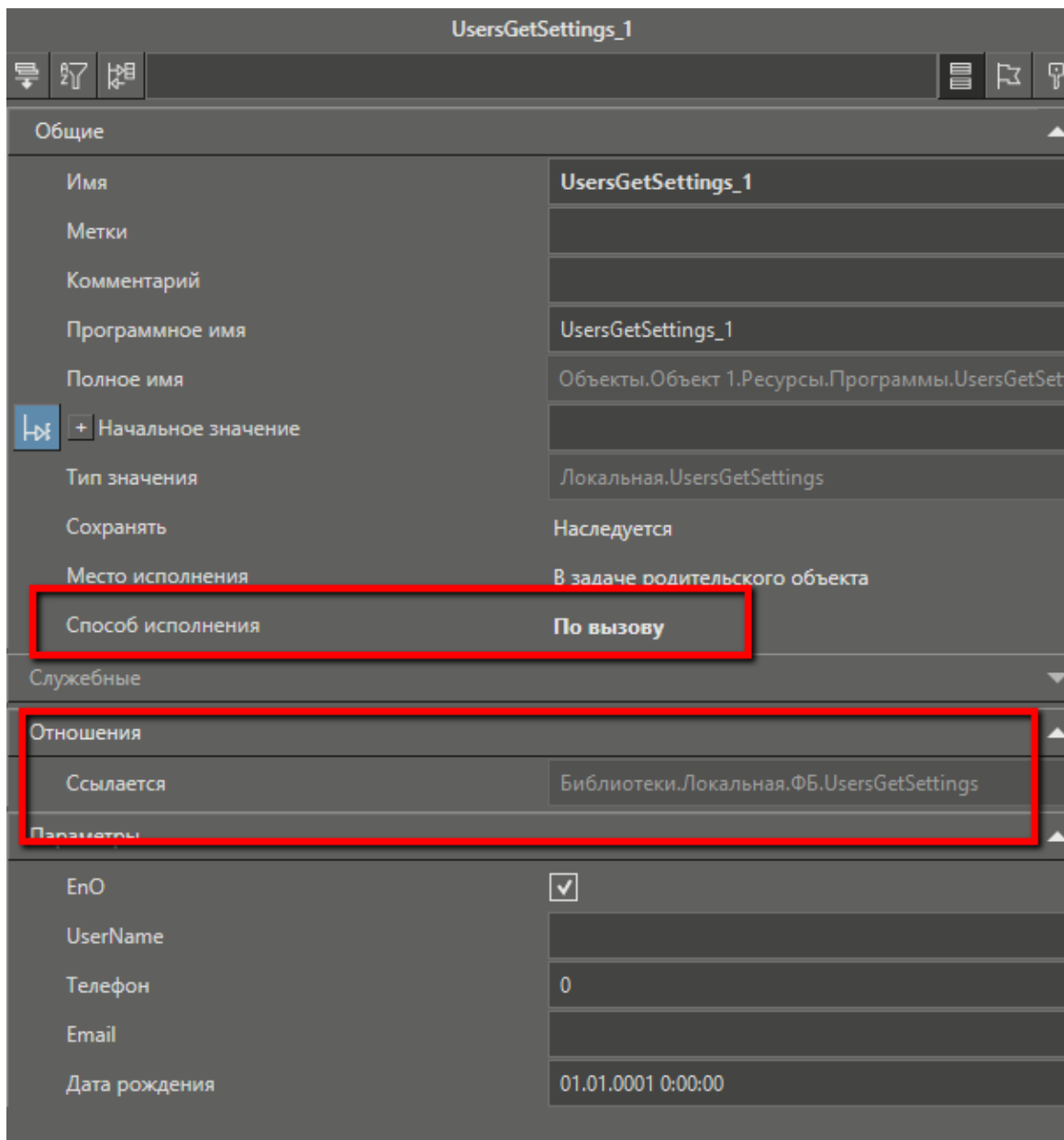
Список дополнительных настроек пользователя зависит от требований проекта, чтобы создать экземпляр функционального блока, который может работать в конкретном проекте необходимо добавить в пользовательскую библиотеку экземпляр функционального блока UsersGetSettings. Затем добавить в него нужное количество 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:



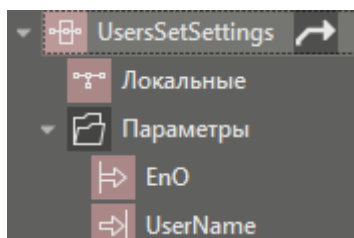
Как правило, для получения значений дополнительных настроек пользователя в окне создают Кнопку, затем в ее событие Клик мыши добавляют действие Вызвать программу (для этого функциональный блок следует перетащить на вход действия Программа).

#### 6.5.4.2.9. 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 операторов

Вид в дереве библиотек

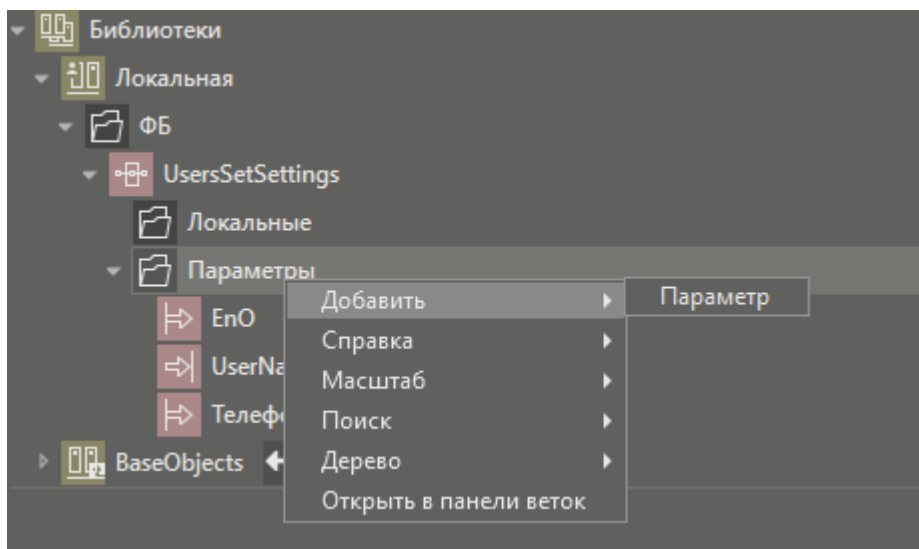


Входы и выходы ФБ:

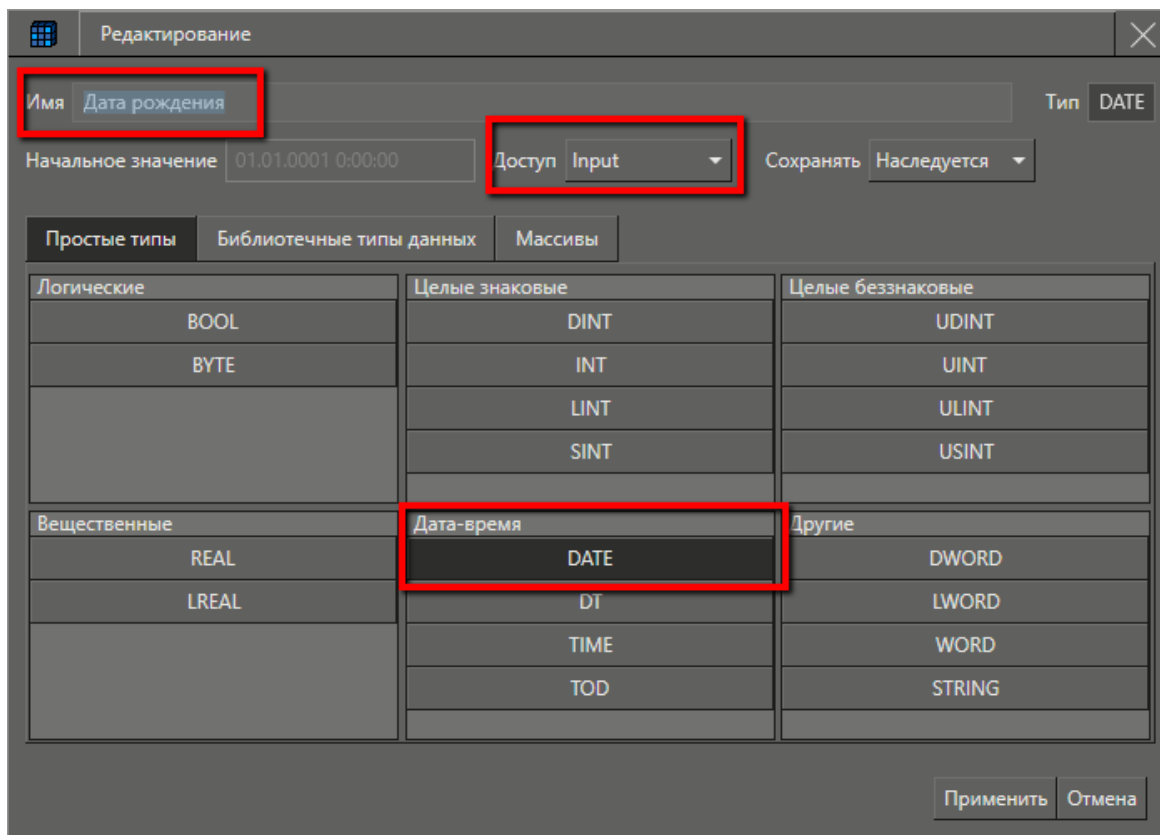
Название	Тип	Назначение
UserName	STRING	Задается имя пользователя, значения дополнительных настроек которого нужно изменить

Рекомендации по работе с ФБ

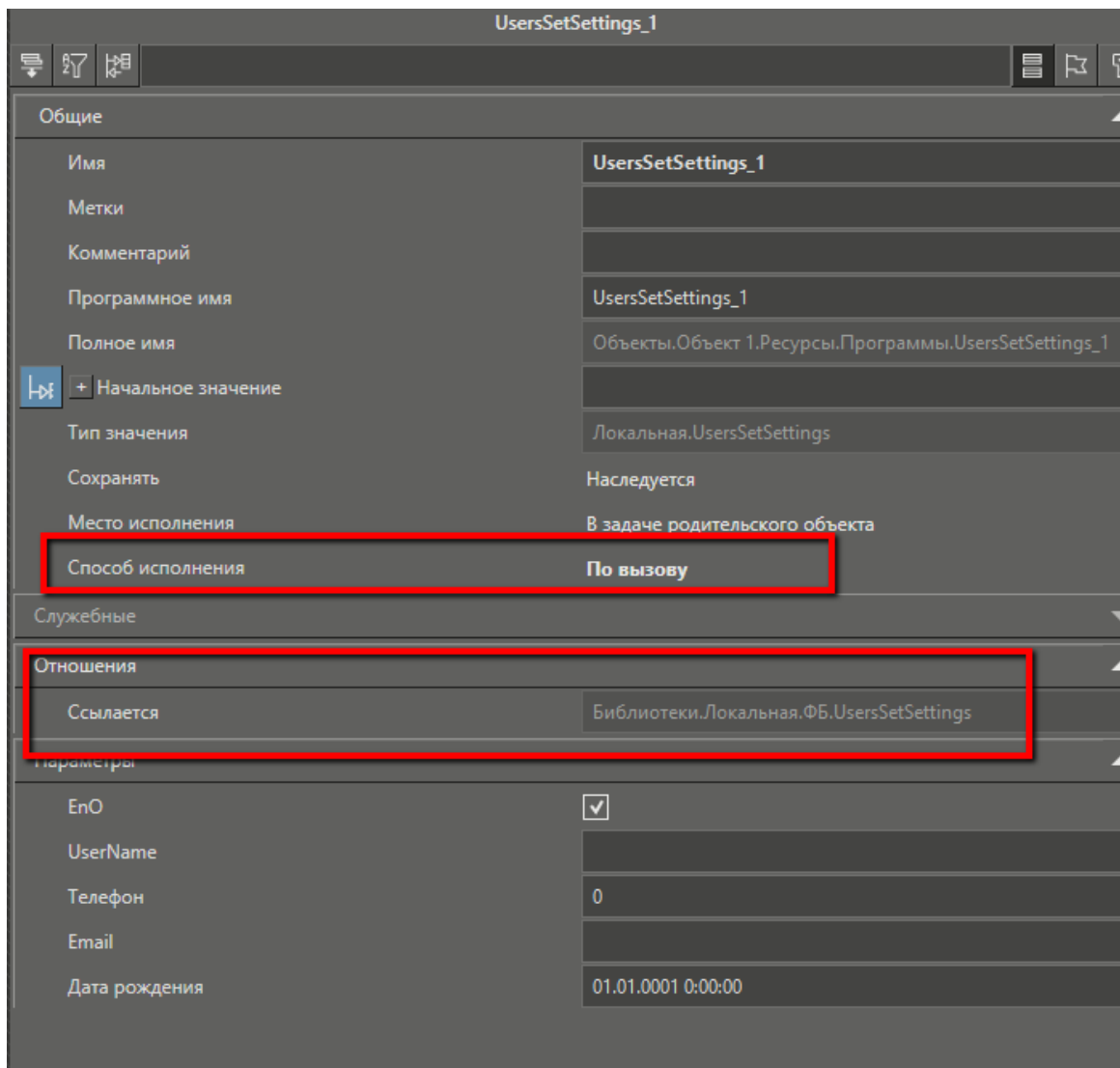
Список дополнительных настроек пользователя зависит от требований проекта, чтобы создать экземпляр функционального блока, который может работать в конкретном проекте необходимо добавить в пользовательскую библиотеку экземпляр функционального блока UsersSetSettings. Затем добавить в него нужное количество 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:

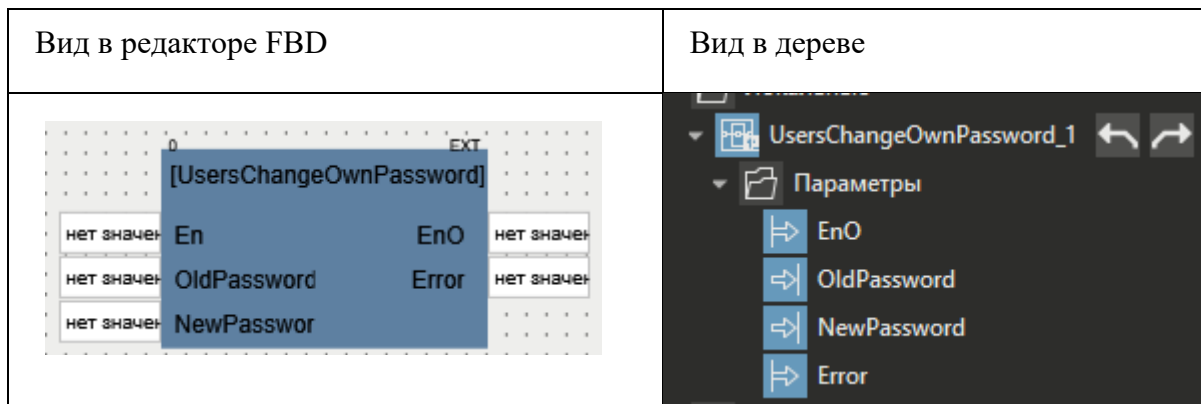


Как правило, для изменения значений дополнительных настроек пользователя в окне создают Кнопку, затем в ее событие Клик мыши добавляют действие Вызвать программу (для этого функциональный блок следует перетащить на вход действия Программа).

#### 6.5.4.2.10. UsersChangeOwnPassword

Функциональный блок UsersChangeOwnPassword служит изменения собственного пароля текущего пользователя, который был добавлен в режиме исполнения.

Важно! Данный ФБ не работает с пользователями, добавленными в режиме разработки.



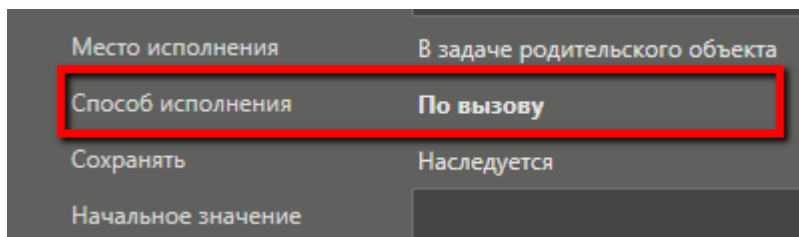
#### Входы и выходы ФБ:

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:

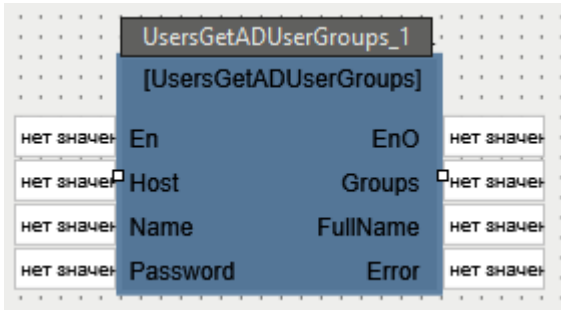
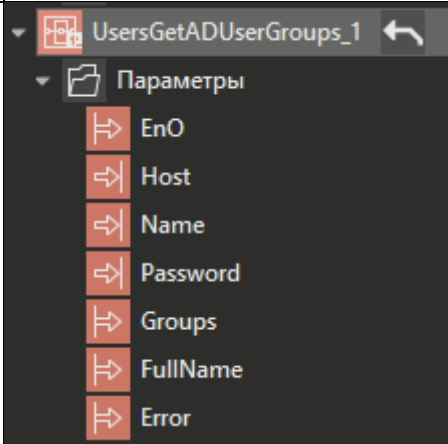


Как правило, изменение пароля происходит в результате нажатия на Кнопку. Для этого в ее событие Клик мыши добавляют действие Вызвать программу путем перетаскивания функционального блока на вход действия Программа.

Входы функционального блока можно связать с параметрами Текст элементов Пароль

### 6.5.4.2.11. UsersGetADUserGroups

Функциональный блок UsersGetADUserGroups служит для получения списка групп, в которые добавлен пользователь операционной системы (Active Directory).

Вид в редакторе FBD	Вид в дереве
	

Входы и выходы ФБ:

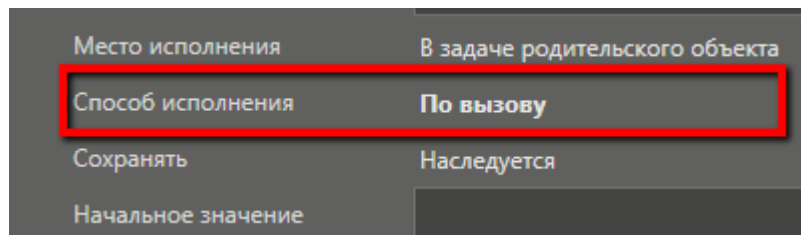
Название	Тип	Назначение
Host	STRING	Вход ФБ. Указывается имя домена (host) или IP-адрес сервера, где настроен Active Directory, например, adtest.insat.ru или 10.0.6.90
Name	STRING	Вход ФБ. Задается имя (логин) пользователя AD, о котором необходимо получить информацию.
Password	STRING	Вход ФБ. Задается 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).

## 6.6. 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

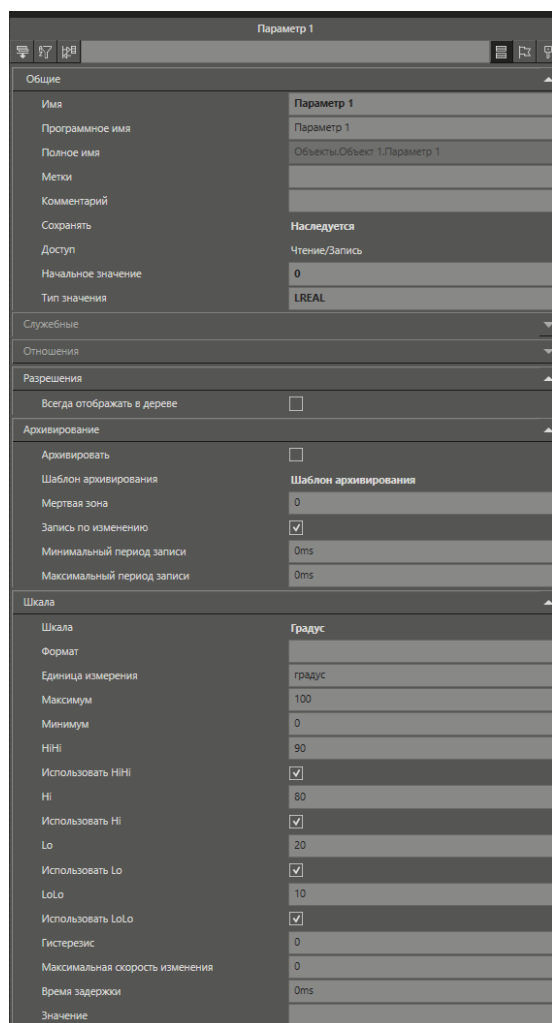
**Важно!** В дерево системы и в дерево объектов могут добавляться только экземпляры библиотечных программ (функциональных блоков) и экземпляры тревог, тип отношений которых Ссылается. Поэтому параметры могут добавляться только в типы этих элементов в дереве библиотек.

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.

### 6.6.1. Панель свойств Параметров

Панель свойств параметров имеет вид:



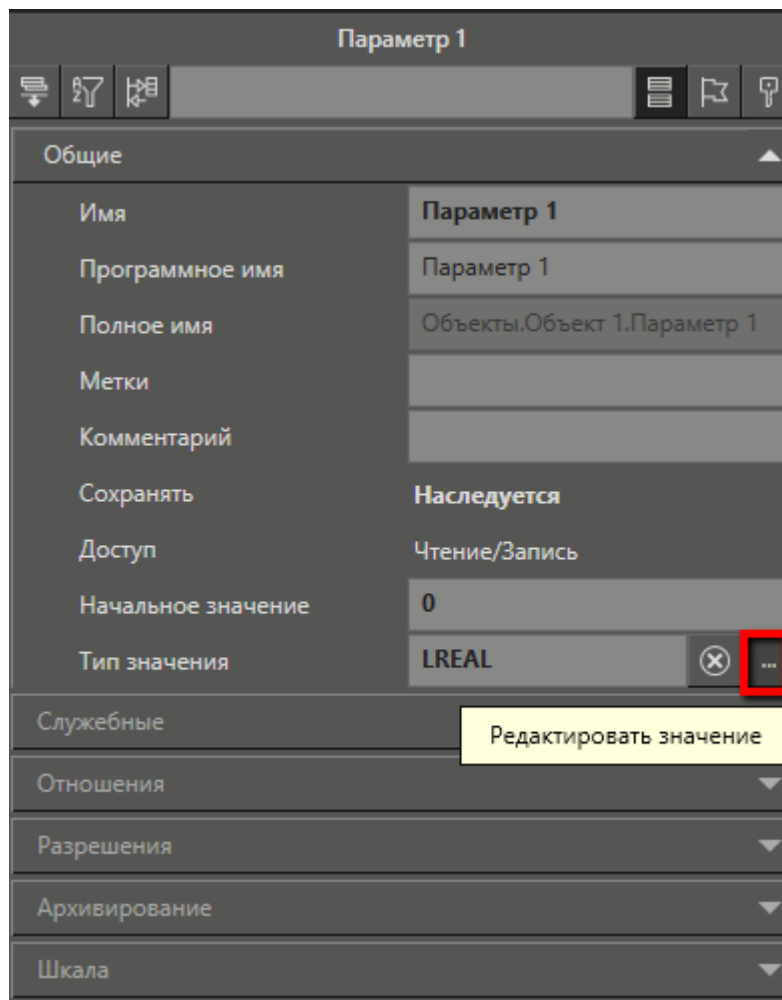
Содержит следующие категории:

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	Если содержимое объекта скрыто в проекте, но этот флаг установлен, то параметр останется доступен в дереве.
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.

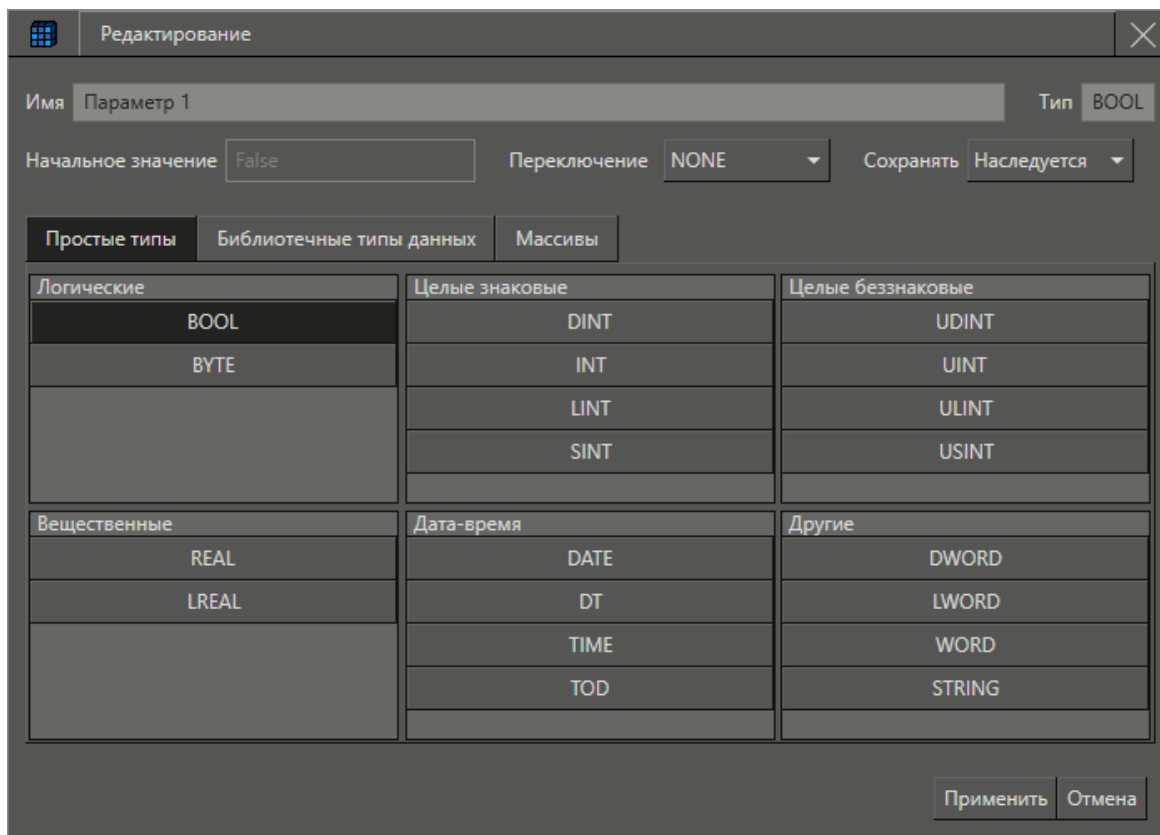
### 6.6.2.Окно настройки параметра

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	В текстовом поле задается имя параметра. <b>Важно!</b> Имена не могут начинаться с цифр.
Тип	Отображается выбранный тип параметра.
Начальное значение	Задается начальное значение параметра, которое будет присвоено ему при старте исполнительной системы.
Доступ	Задается для параметров программ. Определяет доступ к элементу: <ul style="list-style-type: none"> <li>• input - означает, что параметр принимает значение от внешнего источника для использования его при работе программы;</li> <li>• output - означает, что значение параметра получилось в результате работы программы;</li> <li>• InOut - означает, что значение параметра берется от внешнего источника перед началом работы программы, затем</li> </ul>

	<p>меняется в программе, и может быть отправлено внешнему источнику. Для таких параметров нельзя устанавливать свойство Сохранять в состояние Да.</p> <p>Для локального Параметра программы:</p> <ul style="list-style-type: none"> <li>• Local - значение параметра между вызовами программы будет сохраняться;</li> <li>• Temp - после срабатывания программы значение параметра не сохраняется.</li> </ul>
Переключение	В данной версии не поддерживается.
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"> <li>• Наследуется - свойство определяется у родительского элемента;</li> <li>• Да - значение параметра сохраняется, настройки задаются в службе Сохранение состояния;</li> <li>• Нет - Значение параметра не сохраняется;</li> <li>• Константа - значение параметра не может быть изменено в режиме исполнения;</li> <li>• Смотрите также: Модификаторы операторов определения переменных.</li> </ul> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Важно! Свойство Сохранять=Да нельзя устанавливать для параметров программ, у которых свойств Доступ=InOut.</p> </div>
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, 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.
--------------------	---

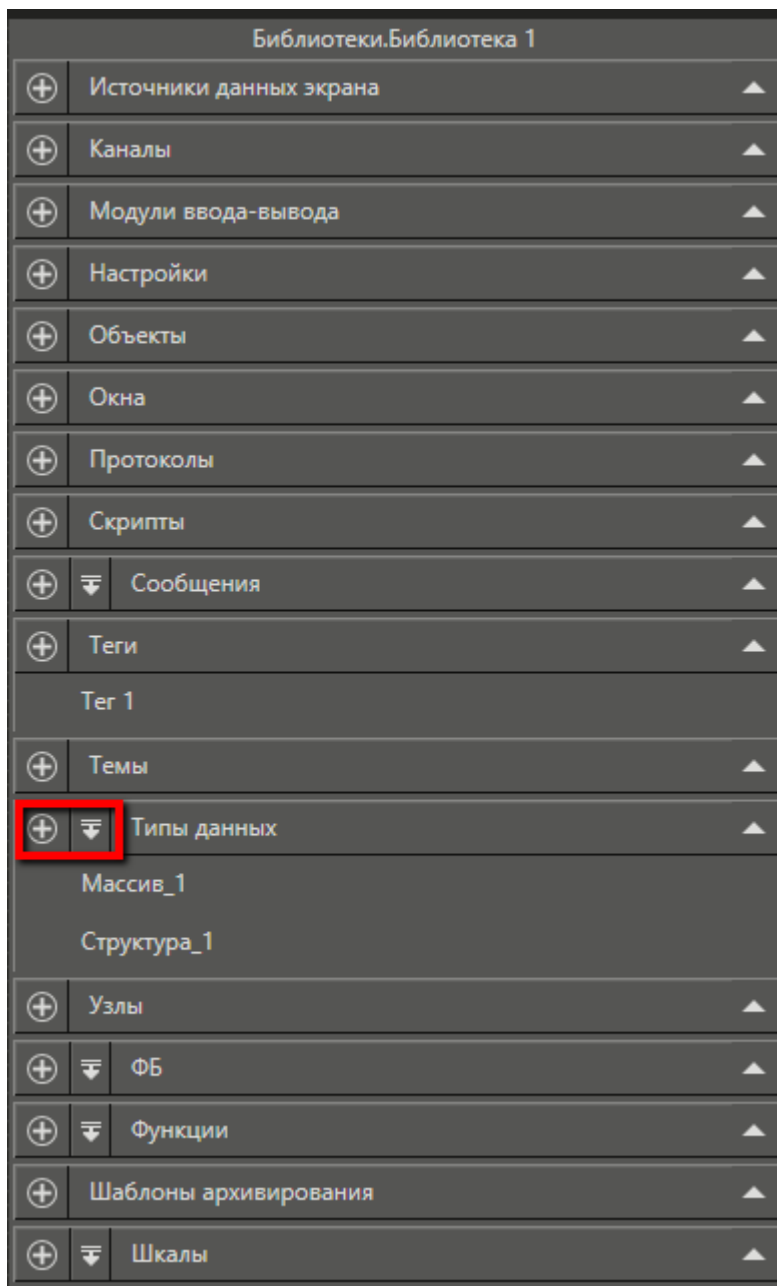
### 6.6.3.Создание нового типа данных

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).

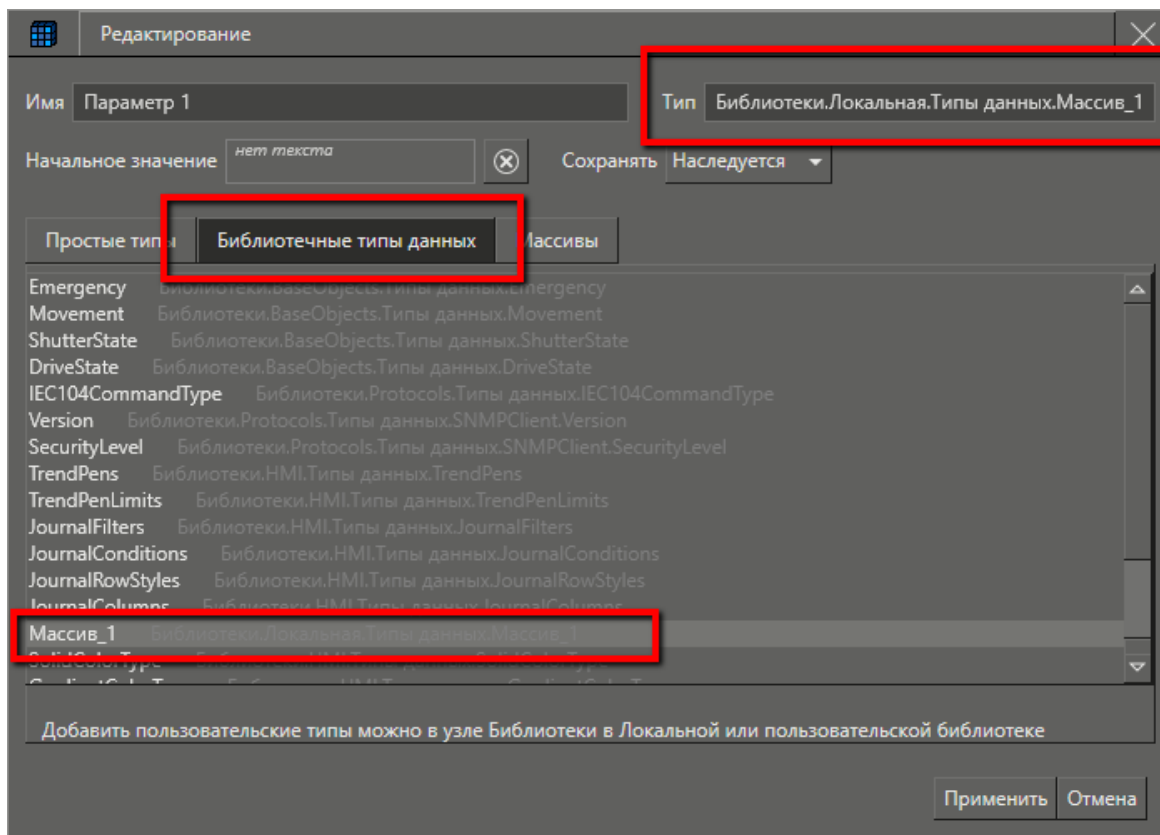
Новый тип данных добавляется при помощи контекстного меню библиотеки, либо ее контекстной панели:



Настройка нового типа данных зависит от его вида.

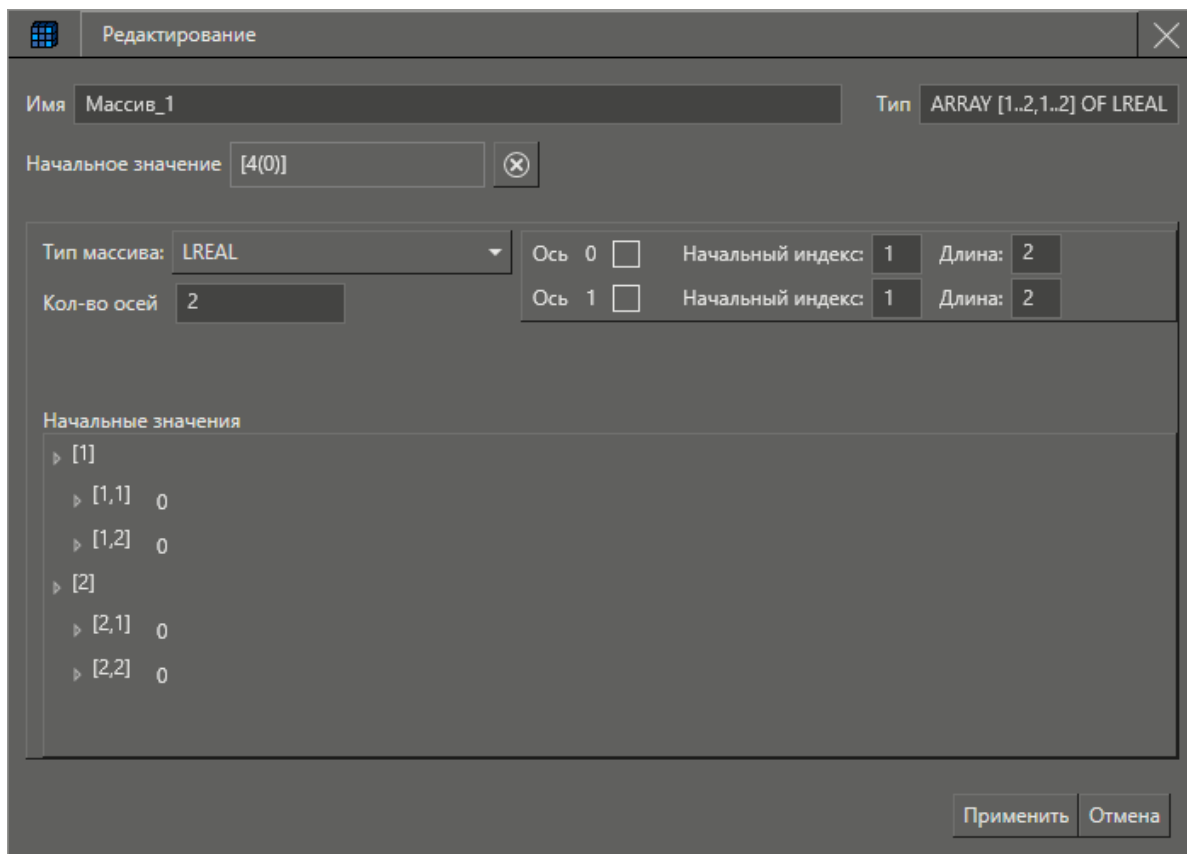
После добавления нового типа данных, его можно назначить параметру проекта в окне настройки, во вкладке Библиотечные типы данных:





### 6.6.3.1. 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:

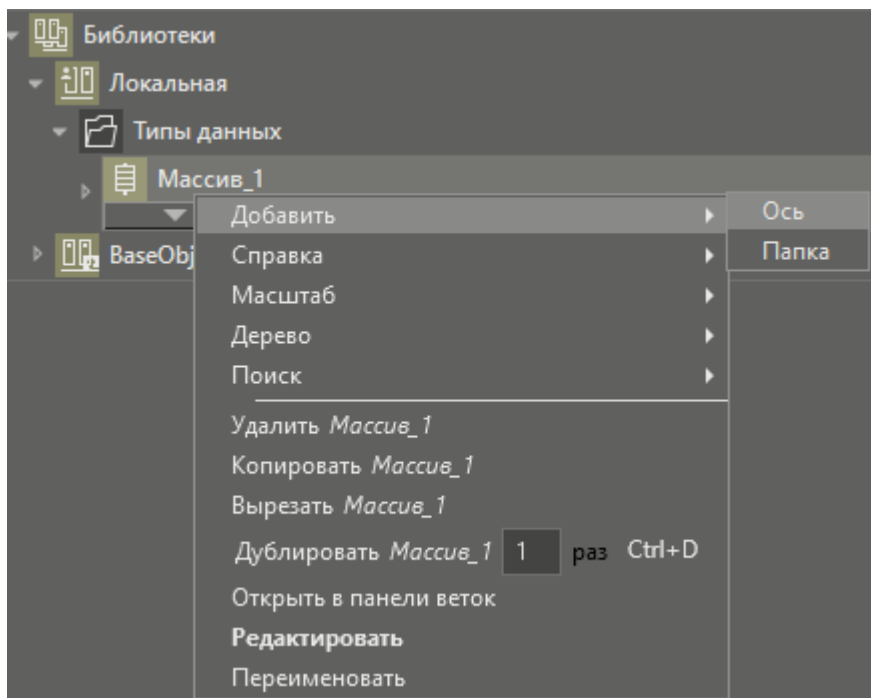
Элемент	Описание
Имя	Задается имя массива
Тип	Формируется тип массива. Данное поле может быть отредактировано вручную (изменения вступят в силу после нажатия на кнопку Применить), либо оно сформируется автоматически, в зависимости от настроек сделанных ниже.
Начальное значение	Позволяет установить значения по умолчанию для всех элементов массива. Формат записи: $[A(B)]$ , где: <p style="margin-left: 40px;">A - обозначает скольким элементам массива, начиная с первого, нужно присвоить значение по умолчанию.</p>

	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.
Настройки оси 	<p>Поле служит для настройки оси:</p> <ul style="list-style-type: none"> <li>• 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> <li>• Начальный индекс - определяет начальный индекс по оси;</li> <li>• Длина - определяет длину оси.</li> </ul>
Группа Начальные значения	<p>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> 

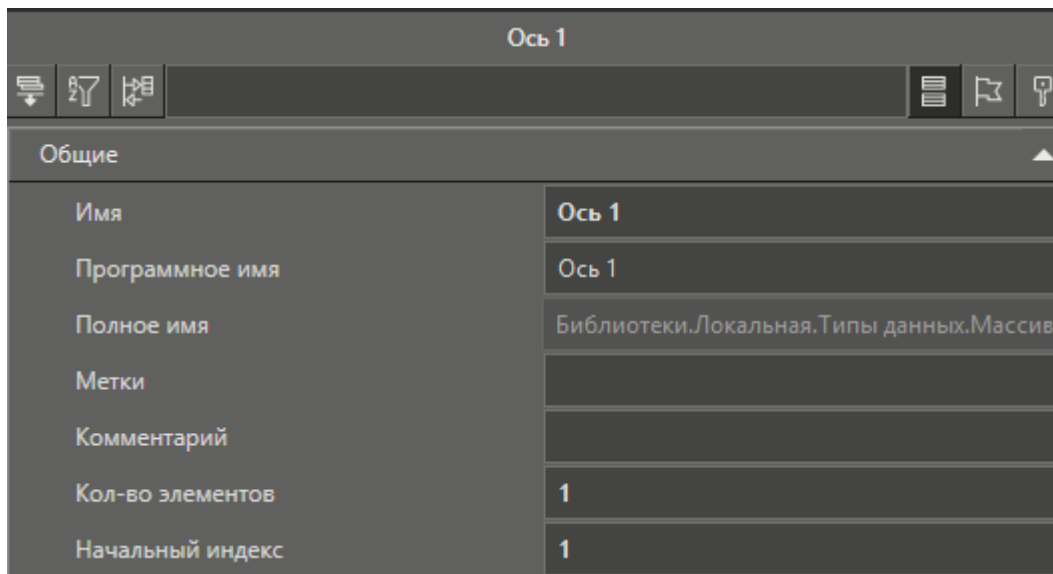
**Важно!** Для дальнейшей работы с параметрами, которым назначен тип данных Массив с динамическим размером осей, следует использовать специальные функции.

Добавление осей через контекстное меню

Ось можно добавить в массив, используя пункт Добавить-Ось контекстного меню:



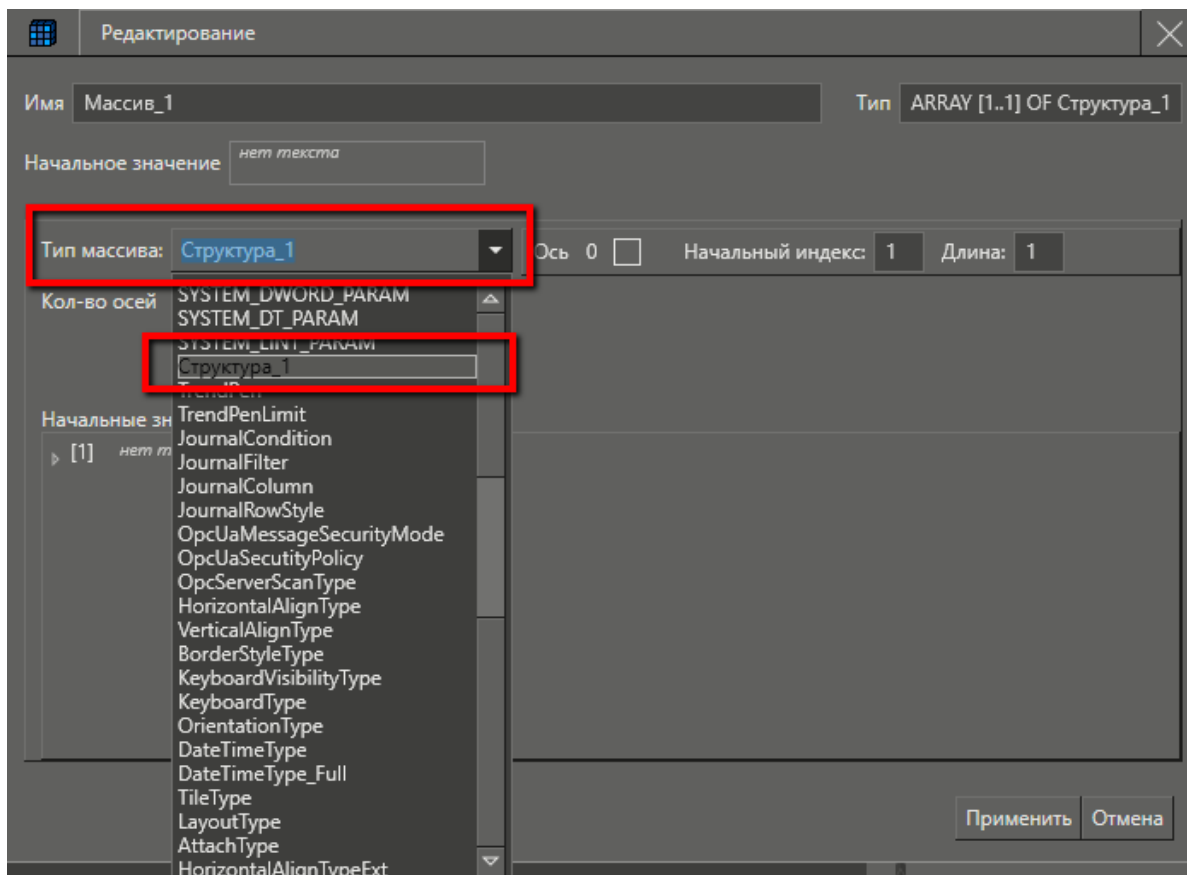
Затем, в панели свойств оси необходимо задать количество элементов и начальный индекс :



### 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:

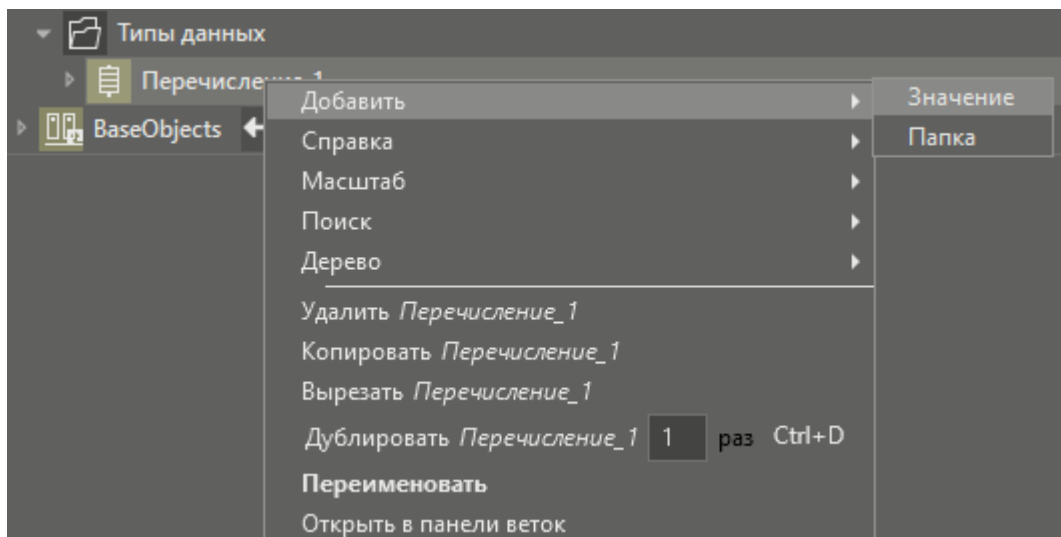


### 6.6.3.2. Новый тип данных Перечисление

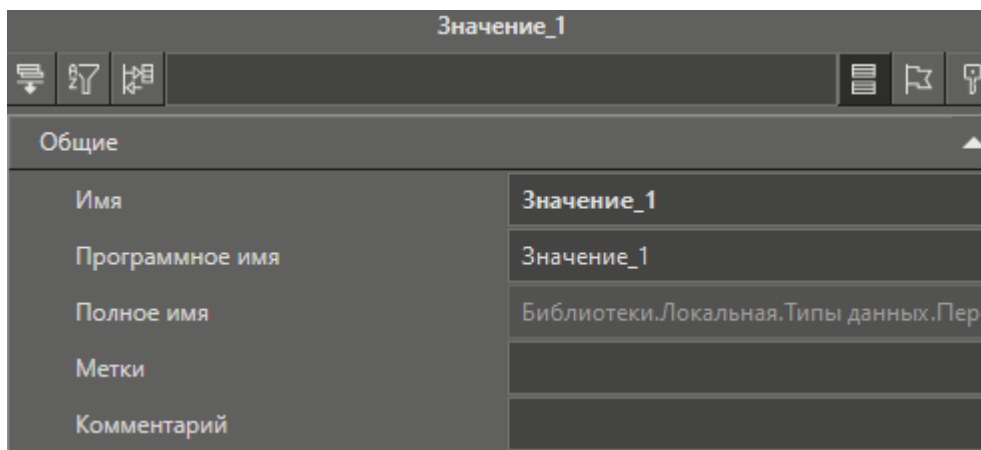
Перечисление используется в случае, когда параметр может принимать только predetermined values.

After adding a new data type to the library, it is recommended to change its name in the properties panel.

Then, using the context menu, you should add the required number of values:



Затем, в панели свойств задать его имя:



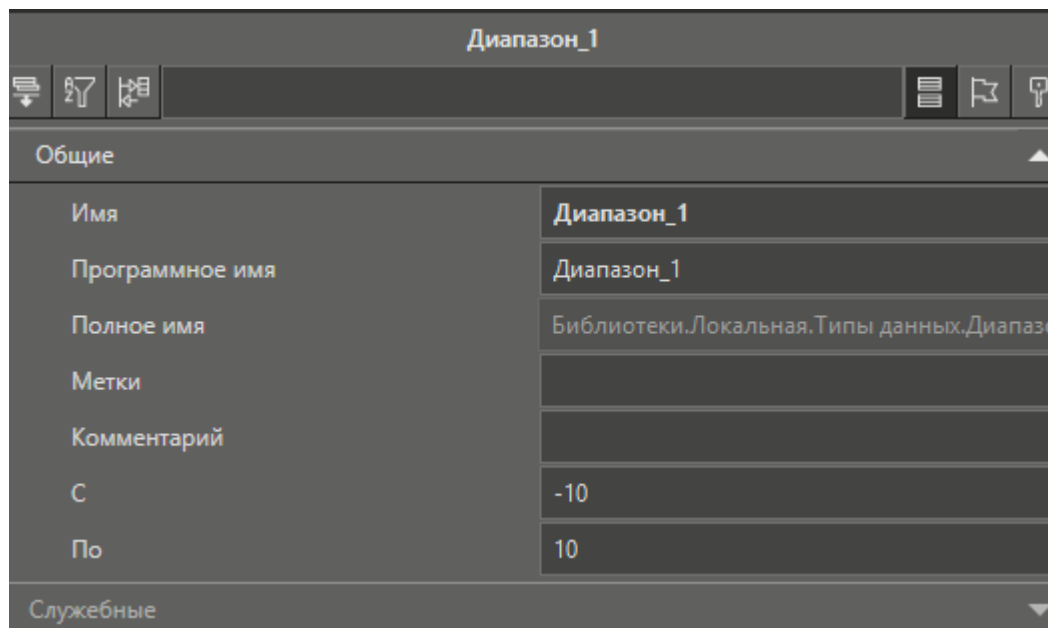
Параметр, которому назначен тип Перечисление сможет принимать значения, которые заданы в поле Имя его дочерних элементов.

**Важно!** Имя значения Перечисления не может содержать пробелы.

Удалить созданное значение перечисления можно, используя контекстное меню либо кнопку клавиатуры Del.

### 6.6.3.3. Новый тип данных Диапазон

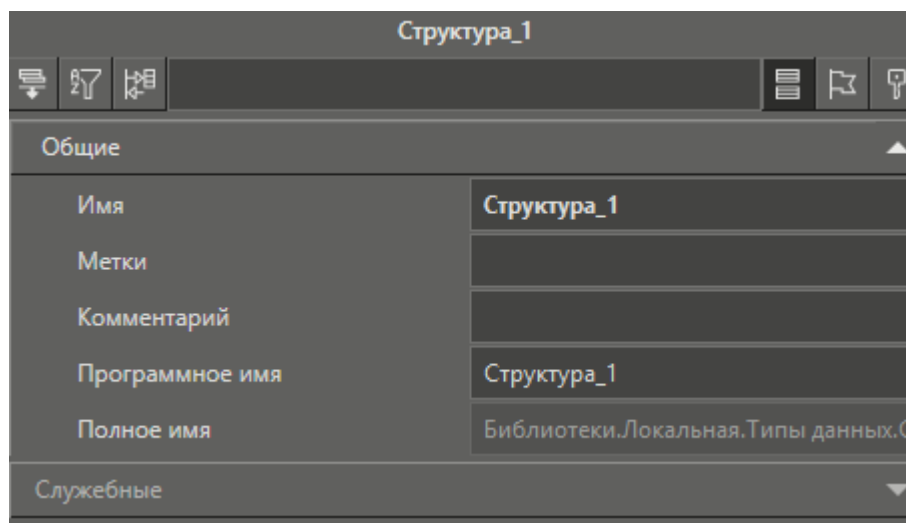
После добавления этого типа данных, необходимо в панели свойств задать его имя и значения диапазона (свойства С и По)



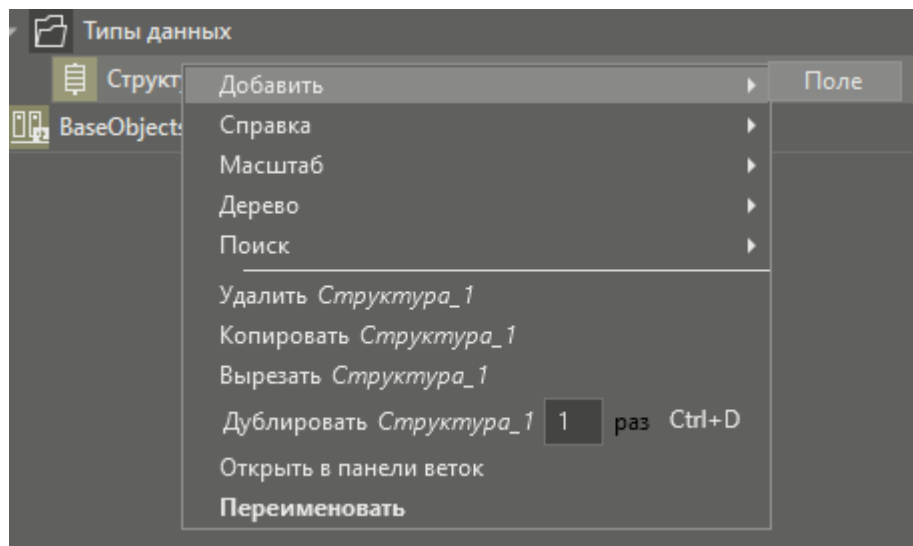
Важно! Значения свойств С и По носят информативный характер. Сообщения о нарушениях границ диапазона выдаваться не будут .

### 6.6.3.4. Новый тип данных Структура

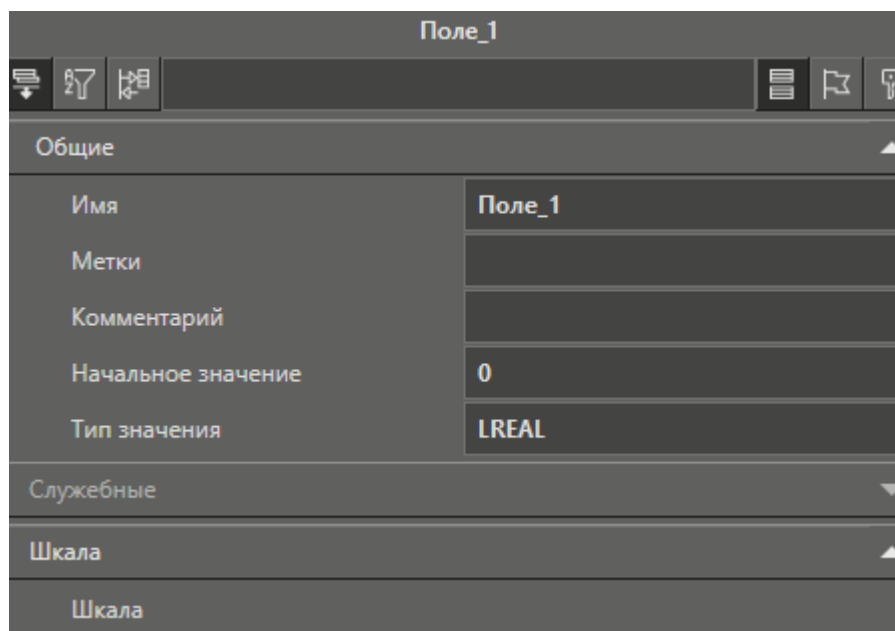
После добавления Структуры в библиотеку желательно изменить ее имя по умолчанию, в дереве либо в панели свойств:



Затем, при помощи контекстного меню добавить требуемое количество Полей :



Автоматически откроется окно Настройки параметров, в котором можно установить все необходимые свойства. 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.



Важно! Имя Поля структуры не может содержать пробелы!

## 6.7. 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.

### 6.7.1.Шкала AI

Шкала предназначена для аналоговых параметров дерева объектов, дерева системы. 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) назначена шкала, то в его панели свойств появится расширенный список настроек:

Шкала AI

☰
🔍
🔗

☰
🚩
🔑

Общие ▲

Имя	Шкала AI
Метки	
Комментарий	

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

Описание 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.
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 $A=HiHi$ -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.
Hi	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 А, где $A=Hi$ -Гистерезис. Приоритет сообщения, появляющегося в результате срабатывания границы, равен 500.
Использовать Hi	BOOL	Если флаг установлен в состояние TRUE, то граница Hi будет использоваться в работе.
Lo	Соответствует типу параметра	Если значение параметра опускается ниже заданной величины, то выдается соответствующее сообщение. Сообщение будет считаться активным до тех пор пока параметр не примет значение, превышающее величину А, где: $A=Lo$ +Гистерезис. Приоритет сообщения, появляющегося в результате срабатывания границы, равен 500.
Использовать Lo	BOOL	Если флаг установлен в состояние TRUE, то граница Lo будет использоваться в работе.
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 $A=LoLo$ +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 A, где: $A = \text{Значение границы} + \text{Гистерезис}$ , для нижних границ, и $A = \text{Значение границы} - \text{Гистерезис}$ , 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.

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.

### 6.7.2.Шкала DI

Шкала предназначена для дискретных параметров дерева объектов, дерева системы. 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) назначена шкала, то в его панели свойств появится расширенный список настроек:

Шкала	
Шкала	Шкала DI
Формат	
Единица измерения	
Сообщение при включении	
Приоритет сообщения при вкл	200
Сообщение при выключении	
Приоритет сообщения при выкл	200
Время задержки	0ms

Название	Тип	Описание
Формат	STRING	Не используется
Единица измерения	STRING	Не используется
Сообщение при включении	STRING	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	STRING	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.
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.

### 6.7.3.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:

Шкала		Шкала
Шкала		
Формат		
Единица измерения		
Название	Тип	Описание
Формат	STRING	Влияет на графическое отображение элемента, например, на количество знаков после запятой, которые будут отображаются 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.
--	--	---

## 6.7.4.Динамическая шкала

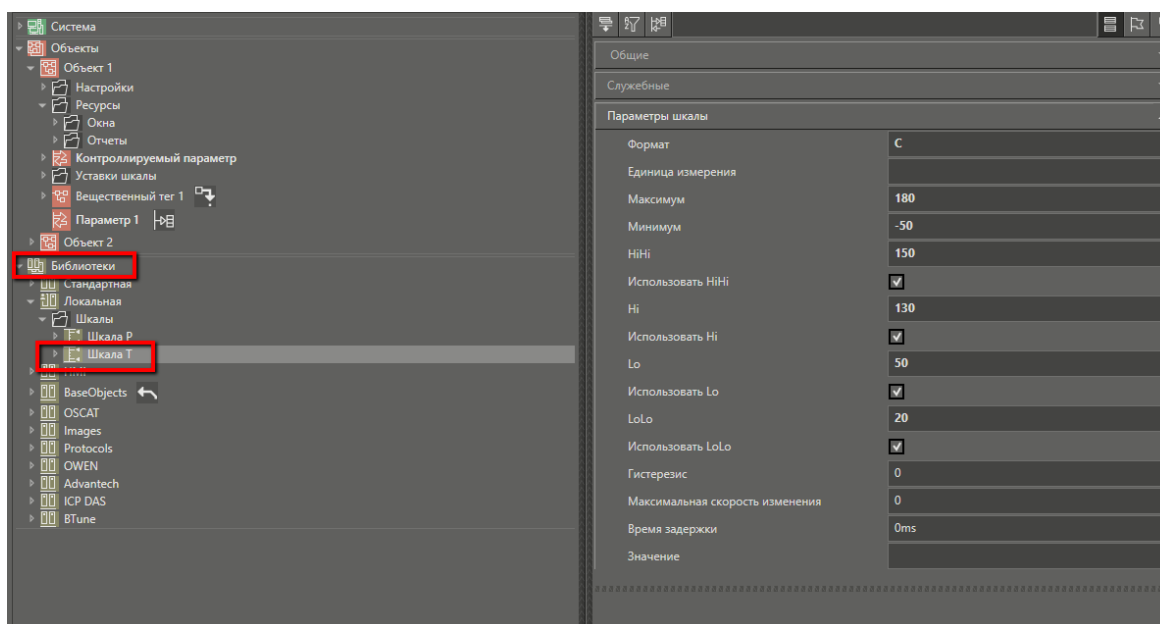
Если необходимо, чтобы значения границ шкал менялись в режиме исполнения, то необходимо установить связь между каким либо параметром и свойством, отвечающим за значения границ.

## 6.7.5.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:



## 6.8. Свойства

### 6.8.1.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"> <li>• Чтение - означает, что допускаются только исходящие связи к внешним источникам;</li> <li>• Запись - означает, что допускаются только входящие связи от внешних источников;</li> <li>• Чтение/Запись - означает, что допускаются и входящие, и исходящие связи с внешними источниками.</li> </ul> <p>Для элемента Канал возможные значения:</p> <ul style="list-style-type: none"> <li>• input - означает, что в канале будет только один параметр, свойство Доступ которого - Чтение . Используется в случае когда канал получает данные от устройства;</li> <li>• output - означает, что в канале будет только один параметр, свойство Доступ которого - Запись. Используется в случае когда канал записывается данные в устройство;</li> <li>• InOut - означает, что в канале будет два параметра: один используется для чтения из устройства, а второй для записи.</li> </ul> <p>Для Параметра программы:</p> <ul style="list-style-type: none"> <li>• input - означает, что параметр принимает значение от внешнего источника для использования его при работе программы;</li> </ul>	Параметр, Канал, Переменная программы

	<ul style="list-style-type: none"> <li>• output - означает, что значение параметра получается в результате работы программы;</li> <li>• 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.</li> </ul> <p>Для локального Параметра программы:</p> <ul style="list-style-type: none"> <li>• Local - значение параметра между вызовами программы будет сохраняться;</li> <li>• Temp - после срабатывания программы значение параметра не сохраняется.</li> </ul>	
Имя	Название элемента, которое дается ему разработчиком проекта. Для каждого языка, поддерживаемого в среде разработки, можно задать свое название элемента.	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"> <li>• В задаче родительского объекта – элемент будет исполняться в задаче объекта-родителя. Устанавливается по умолчанию;</li> <li>• В основной задаче узла – если родительский объект будет исполняться в задаче пользователя, то при установке этого значения текущий элемент будет исполняться в основной задаче того узла, в который он назначен.</li> </ul>	Программы, ФБ, Объекты

	<p>Элемент будет исполняться в клиенте, в браузере, если в выпадающем списке выбрано :</p> <ul style="list-style-type: none"> <li>• В задаче экрана – элемент будет исполняться в приложении клиента (браузера). Период исполнения задается в элементе 'Шаблон экрана'.</li> </ul> <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	Разработчик проекта может создать свои обозначения элементов, записать их в данном поле, и в дальнейшем использовать эти обозначения при поиске элементов.	У всех элементов
Начальное значение	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.	Параметры
Отключить исполнение	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. Например, когда проект уже сделан, но на объекте еще не подключена часть оборудования.	Объект

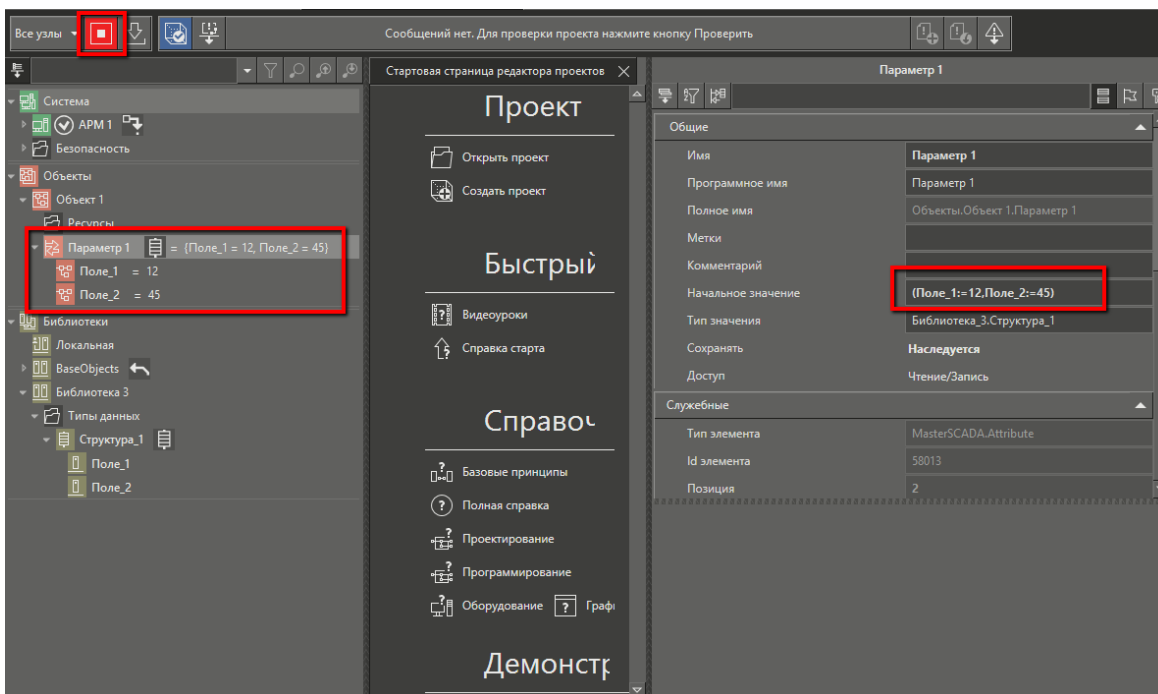
Полное имя	Показывает путь к элементу в дереве, например, Система.APM 1.Протоколы.Modbus TCP.Модуль Modbus TCP 1.AI. Носит информативный характер. Не редактируется.	У всех элементов
Программное имя	Используется для идентификации элемента независимо от текущего языка. Вручную Программное имя можно менять только если разработчик проекта осознает смысл этого действия! Менять вручную это свойство не рекомендуется.	У всех элементов
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"> <li>• Наследуется - свойство определяется у родительского элемента;</li> <li>• Да - значение сохраняется, настройки задаются в службе Сохранение состояния;</li> <li>• Нет - значение не сохраняется;</li> <li>• Константа - значение не может быть изменено в режиме исполнения.</li> </ul> <p>Смотрите также:</p> <ul style="list-style-type: none"> <li>• Модификаторы операторов определения переменных</li> <li>• Работа с файлом горячего рестарта.</li> <li>• Служба Сохранение состояния</li> </ul> <p><b>Важно!</b> Свойство Сохранять=Да нельзя устанавливать для параметров программ, у которых свойств Доступ=InOut</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"> <li>• По вызову - исполняется по условию;</li> </ul>	Программы, ФБ

	<ul style="list-style-type: none"> <li>• Периодически – выполняется периодически (период определяется в задаче узла).</li> </ul>	
Тип значения	Определяет тип данных. Смотрите также Редактор переменной.	Параметры

## Примеры настроек

### Свойство Начальное значение для полей структуры

Свойство Начальное значение для полей структуры можно установить в панели свойств параметра 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:



### Свойство Сохранять для экземпляров ФБ

Если экземпляр функционального блока используется в пользовательской библиотечной программе.

## 6.8.2. Категория Задача

Категория задача присутствует у Протоколов и Служб, модулей ввода-вывода, запросов к базам данных

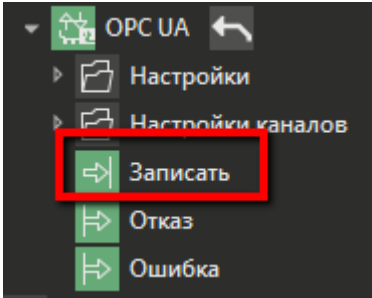
Категория Задача панели свойств у протоколов и служб может содержать все либо часть из приведенных ниже свойств:

Задача	
Период, мс	100
Приоритет	100
Подключение по условию	<input type="checkbox"/>
Выполнение по условию	<input type="checkbox"/>
Выполнять на резервном	<input type="checkbox"/>
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

Имеет следующие настройки:

Название	Описание	Служба и Протокол
Период, (мс)	Задается период, с которым будет производиться опрос модулей ввода-вывода. Данный период должен выбираться с учетом возможностей устройств, работающих по данному протоколу.	Все
Приоритет	Задаёт 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 наиболее важных частей проекта.	Все

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.	Все протоколы
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.	Все протоколы кроме DCON и Modbus
Выполнять на резервном	Если данный флаг установлен, то будет выполняться опрос устройств, подключенных по настраиваемому протоколу, узлом, находящимся в состоянии резервного.	Все протоколы
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 бу-</p>	Протоколы, кроме DCON и Modbus

	<p>дет происходить только по переднему фронту изменения значения этого параметра:</p> 	
<p>Формировать отказ при отказе всех модулей</p>	<p>Определяет способ формирования выхода Отказ протокола. Если значение свойства равно TRUE, то выход Отказ примет значение TRUE в случае когда отказали все модули. Это может быть вызвано, например, отказом шины, тогда имеет смысл переключение на резервный сервер.</p>	<p>Все протоколы</p>
<p>Формировать отказ узла</p>	<p>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.</p>	<p>Все протоколы</p>
<p>Задержка записи при старте опроса (циклов)</p>	<p>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.</p>	<p>Все протоколы</p>



Категория Задача панели свойств у модулей ввода-вывода и запросов к базам данных:

Задача	
Подключение по условию	<input type="checkbox"/>
Опрос по условию	<input type="checkbox"/>
Способ записи	По умолчанию

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> <ul style="list-style-type: none"> <li>По изменению - запись происходит при изменении значения выхода канала</li> <li>По изменению времени - запись происходит при изменении значения или метки времени выхода канала</li> <li>Периодический - запись происходит на каждом цикле</li> <li>По условию - появляется параметр Записать (тип BOOL). Запись всех выходных каналов устройства будет происходить только по переднему фронту изменения значения этого параметра.</li> <li>По умолчанию - используется настройка Способ записи протокола</li> </ul>

### 6.8.3. 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:

Масштабирование	В
Включить масштабирование	<input checked="" type="checkbox"/>
SourceMin	
SourceMax	
TargetMin	
TargetMax	

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	В
Включить масштабирование	<input checked="" type="checkbox"/>
SourceMin	0
SourceMax	65535
TargetMin	4
TargetMax	20

## 6.8.4. 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

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
Шаблон имени последовательно го 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>	Объект

	Если у библиотечного объекта скрыто содержимое, то данное свойство не отображается и изменить его нельзя.	
Образ	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.	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 строка вида: proj_Kanal;help\\proj.chm::/Kanal.htm&gt;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 proj.chm, раздел Kanal.htm.</p>	Все элементы

### 6.8.5. Категория свойств Разрешения

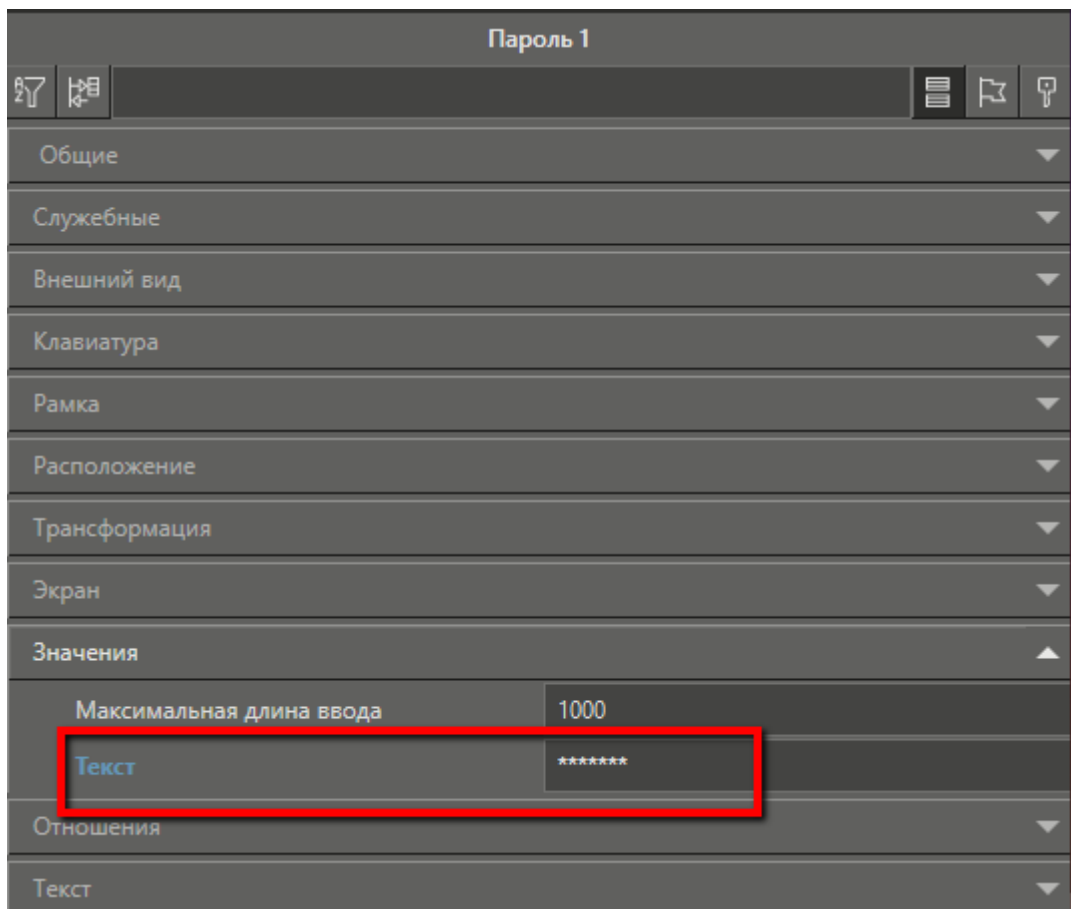
В данном разделе представлены описания настроек, которые могут встречаться у различных элементов в группе Разрешения панели Свойств.

Разрешения		
<div style="display: flex; justify-content: space-between; align-items: center;"> <span>Всегда отображать в дереве</span> <input type="checkbox"/> </div> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>Скрывать значение</span> <input type="checkbox"/> </div>		
Название	Описание	Элементы
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

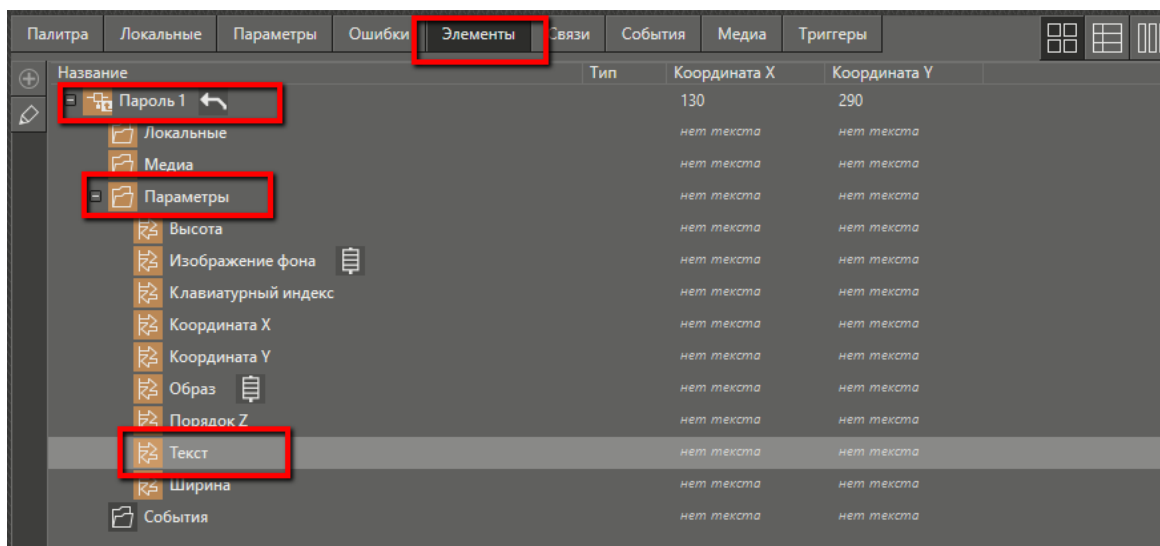
#### Пример использования свойства Скрывать значение

Рассмотрим пример как установленный флаг Скрывать значение повлиял на отображение Начального значения в панели свойств.

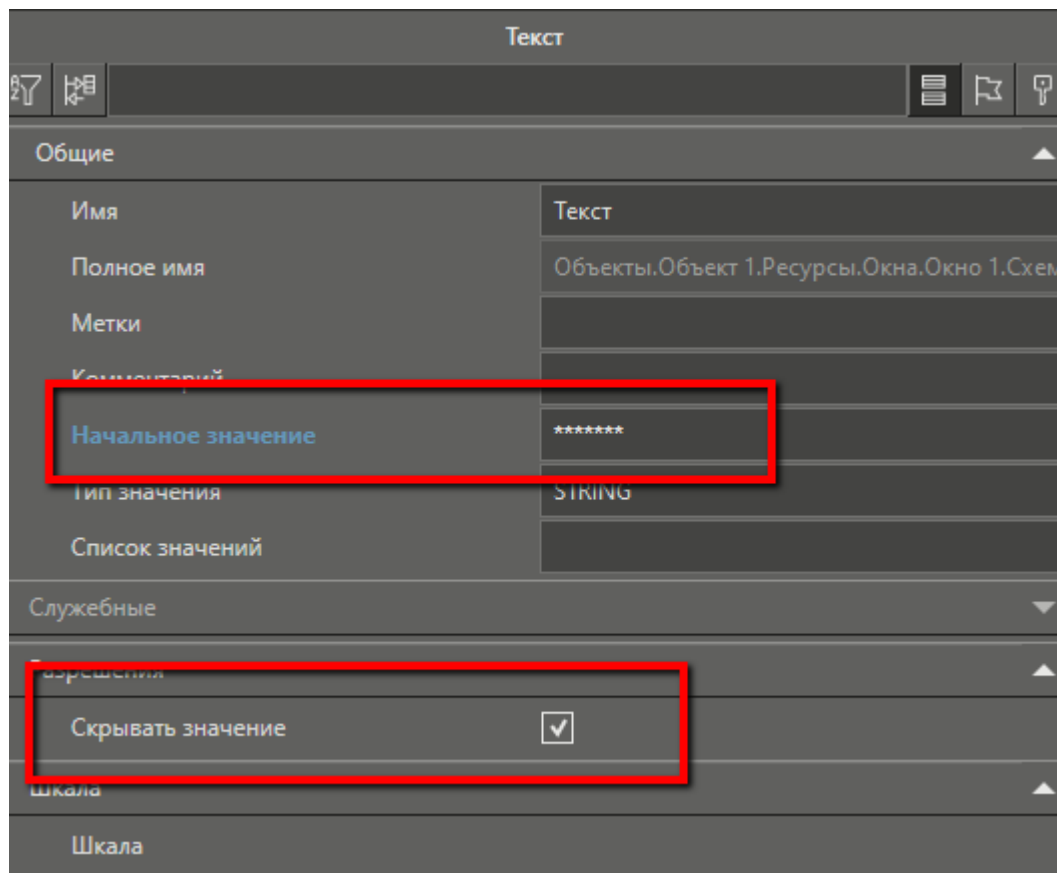
Добавим в окно элемент Пароль. Обратим внимание, что после ввода значение в свойство Текст оно заменяется звездочками:



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:



## 6.9. 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.

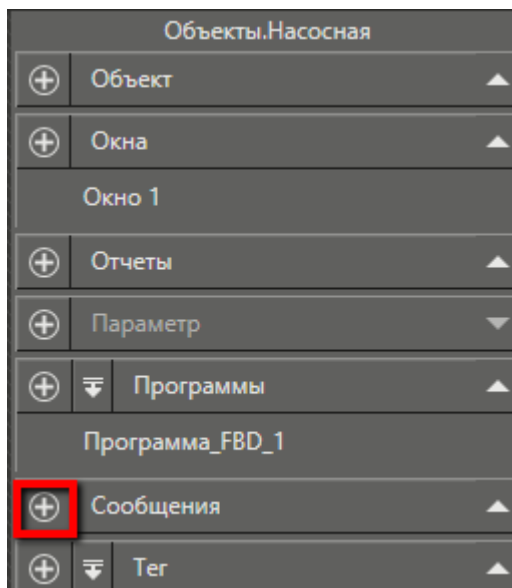
Сообщения можно разделить на условные и безусловные.

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:



## 6.9.1. Типы стандартных сообщений

В MasterSCADA 4D по умолчанию предусмотрены следующие типы сообщений:

- Тревога
- Контроль границ
- Контроль скорости изменения
- Базовое сообщение

Смотрите также:

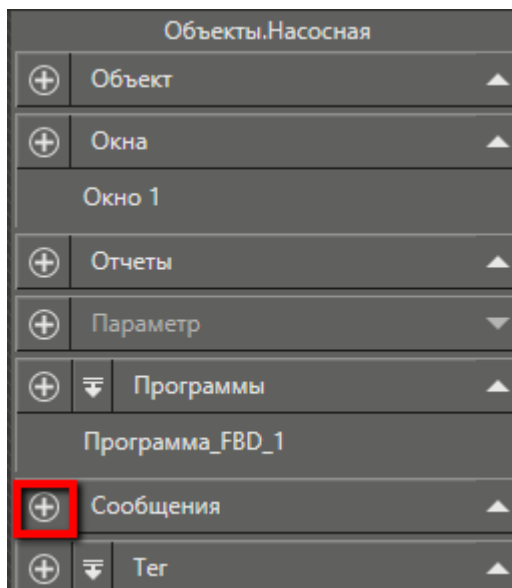
Работа с сообщениями

Журнал - средство отображения сообщений

### 6.9.1.1. Тревога

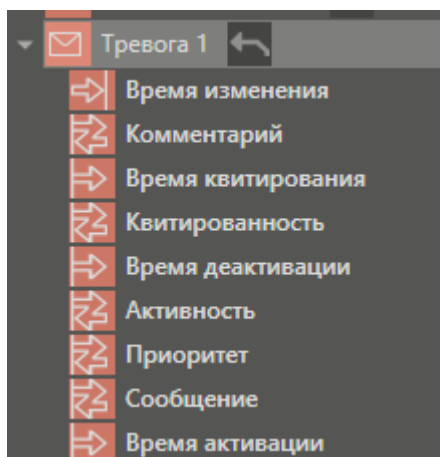
Тип сообщения Тревога служит для формирования условных сообщений.

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
<b>Активность</b>	<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	Тип	Назначение

Приоритет	DINT	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.
Сообщения	STRING	Задается текст сообщения. Как правило, значение задается в виде константы. При необходимости можно установить связь параметра со строковой переменной (например, выходом программы), в которой будет формироваться разный текст сообщения.
Время изменения	DT	Показывается время последнего изменения Тревоги. Совпадает с одним из трех параметров: временем активации, временем квитирования, временем деактивации.
Комментарий	STRING	Может содержать произвольный текст. Как правило, заполняется оператором при квитировании сообщения.
Время квитирования	DT	Выдается момент времени, в который оператор квитировал сообщение, т.е. когда свойство Квитированность изменилось с False на TRUE.
Квитированность	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.
Время деактивации	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.

Активнос ть	BOOL	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 активаци и	DT	Отображается время, когда Активность изменила состояние с False на TRUE.

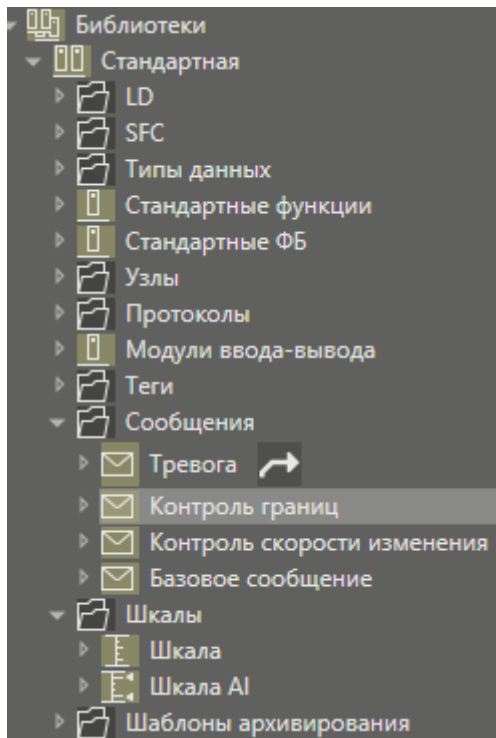
Тип сообщения Тревога хранится в библиотеке Стандартная, в группе Сообщения.

Разработчик проекта на основе данного типа может создать свои типы тревог, например, добавив свои собственные параметры, которые будут архивироваться и выводиться в журнал.

### 6.9.1.2. 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.

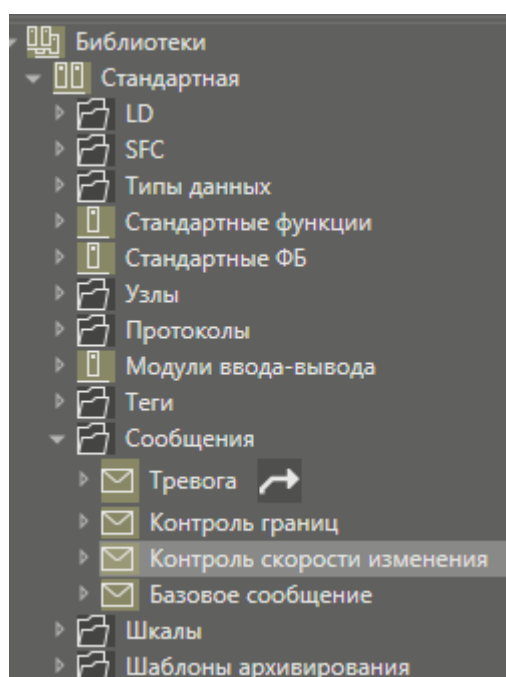
Сообщения будут автоматически попадать в журнал сообщений в соответствии с настройками журнала.

Время активации, время деактивации и другие параметры состояния будут зависеть от настроек шкалы параметра и его значения в тот или иной момент времени.

### 6.9.1.3. 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.

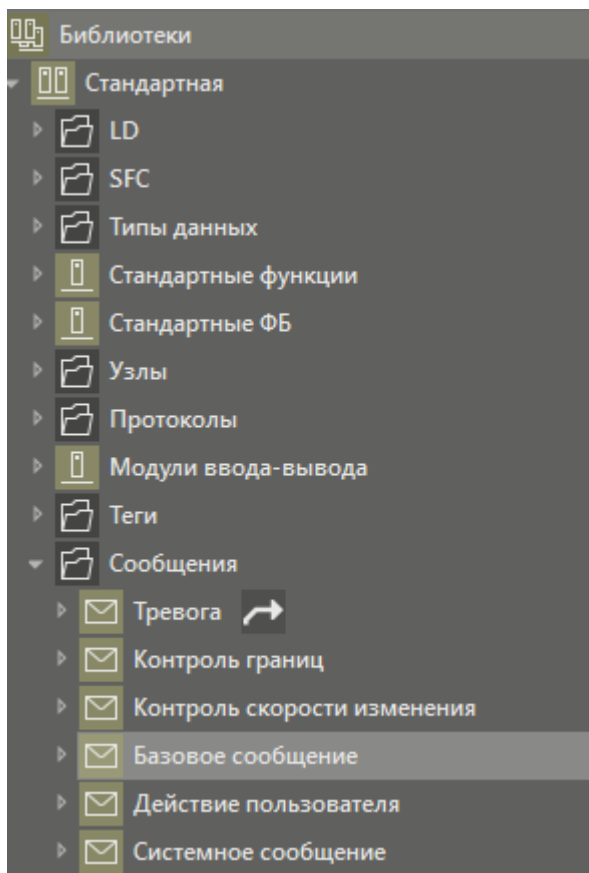
Сообщения будут автоматически попадать в журнал сообщений в соответствии с настройками журнала.

Время активации, время деактивации и другие параметры состояния будут зависеть от настроек шкалы параметра и его значения в тот или иной момент времени.

### 6.9.1.4. Базовое сообщение

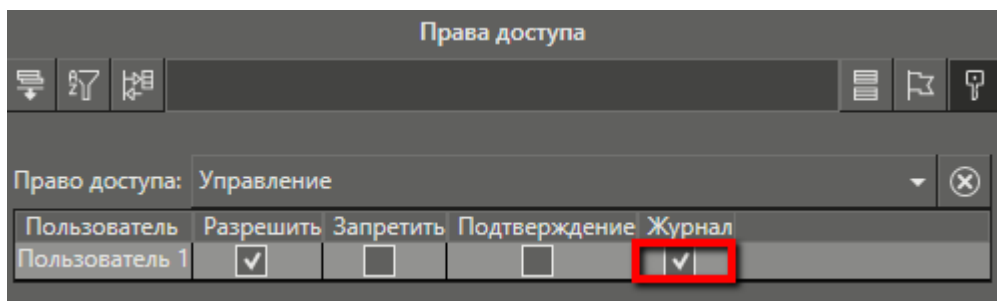
Базовое сообщение используется для формирования безусловных сообщений. Безусловное сообщение создается в MasterSCADA 4D при помощи ФБ FireBaseEvent.

Данный тип сообщений находится в библиотеке Стандартная:



### 6.9.1.5. Действие пользователей

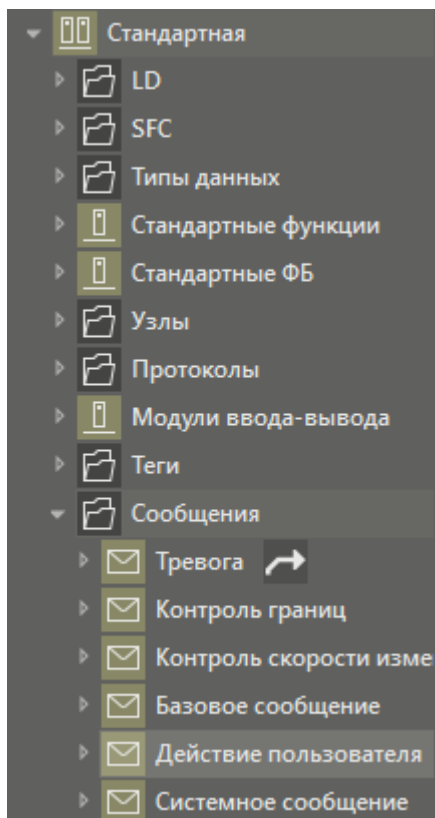
Этот тип сообщений используется для формирования сообщений по действиям пользователей. Если в настройках прав доступа для конкретного действия установлен флаг Журнал, то когда пользователь произведет данное действие, сообщение о нем появится в системе автоматически:



Сообщения, сигнализирующие о начале и окончании сессии пользователей формируются автоматически.

Данный тип сообщений находится в библиотеке Стандартная:





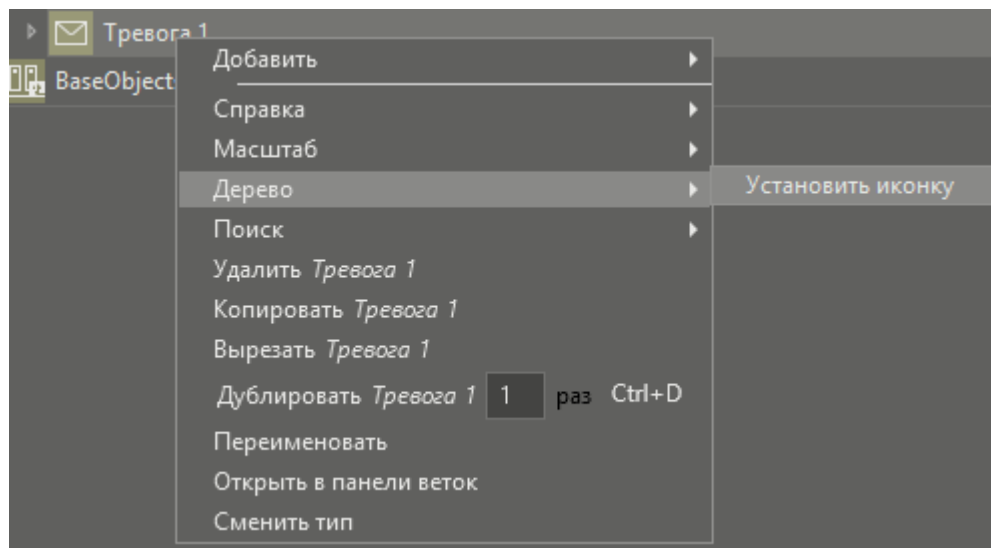
Дополнительных настроек не требуется.

Сообщения будут автоматически попадать в журнал сообщений, в соответствии с настройками журнала.

## 6.9.2. 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 predetermined 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**.



### 6.9.3. 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:

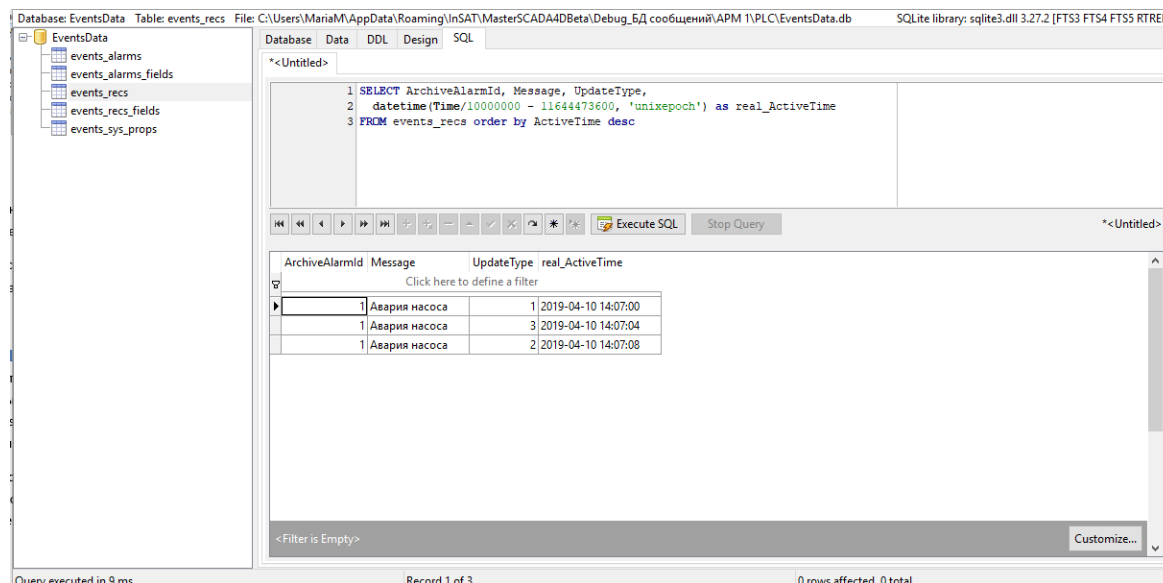
- SQLITE;
- POSTGRESQL;
- MS SQL.

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\InSAT\<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\InSAT\MasterSCADA4DRT1.2\Server\EventsData.db

Архив сообщений на других ОС хранится внутри папки mpls.

Работа с архивными сообщениями

Для просмотра архивов сообщений в MasterSCADA 4D предусмотрен графический элемент – Архивный журнал.

## 6.10. Настройка сетевых проектов

Поддерживаемые архитектуры:

- Клиент-серверная

- Сервер-серверная
- Резервирование узлов

### 6.10.1. 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.

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:

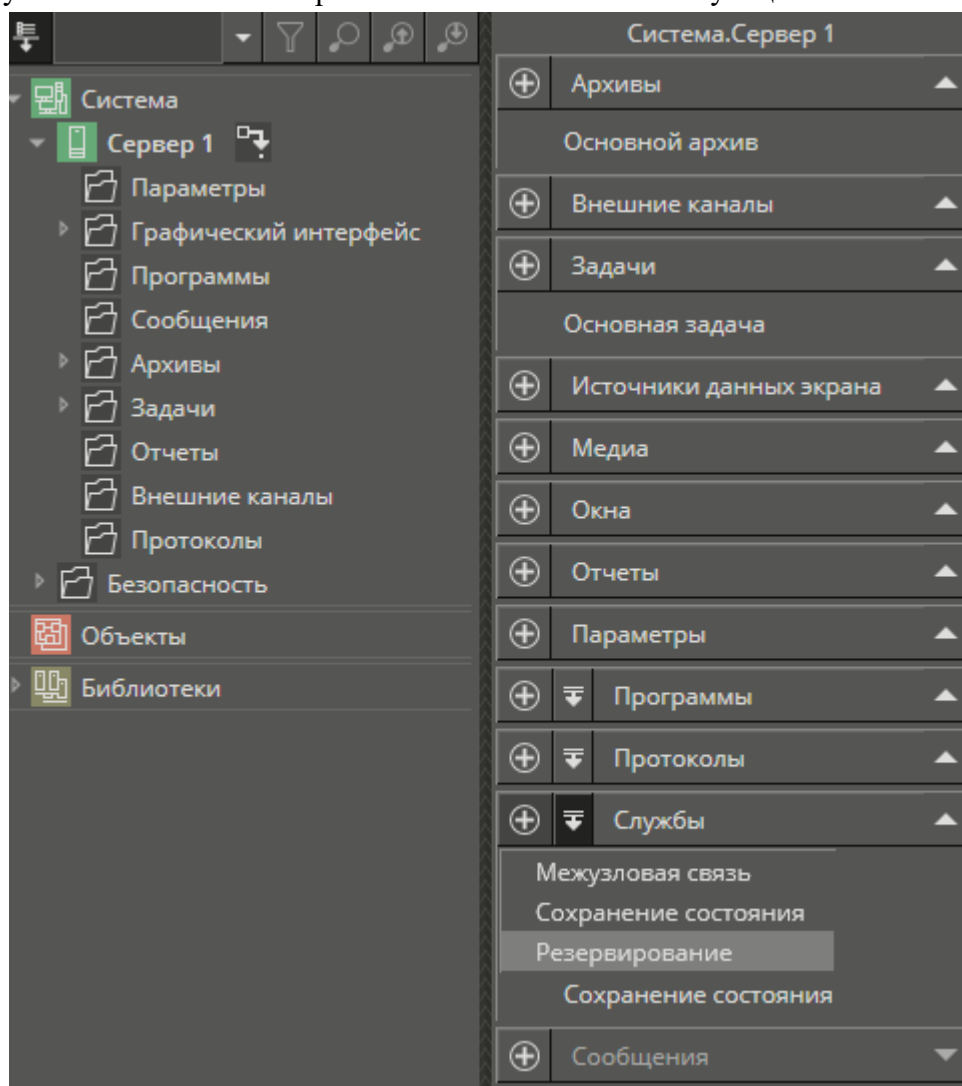
1. если текущий узел Slave и при этом другой узел не отвечает или тоже находится в режиме Slave, то осуществляется переключение текущего узла в MASTER;

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.

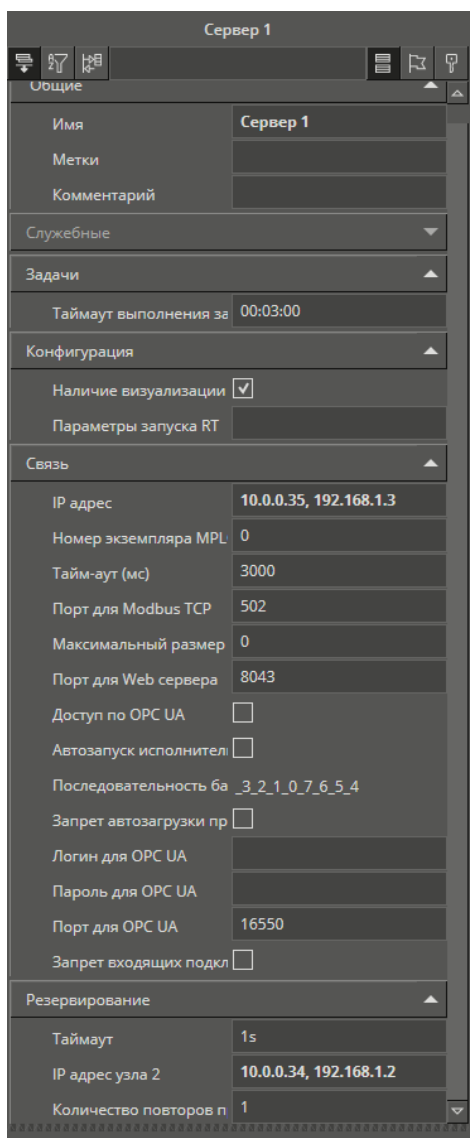
Оба узла равноправны, однако на узле 2 имеется пауза при старте 10 сек, для того чтобы при одновременном включении узел 1 загрузился раньше и стал MASTER. Но в дальнейшем переключение идет по общим правилам.

#### 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.



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.

3. Расширенный доступ к настройкам, отвечающим за резервирование, можно получить в панели настройки. Для этого в дереве системы следует выделить службу **Резервирование**.

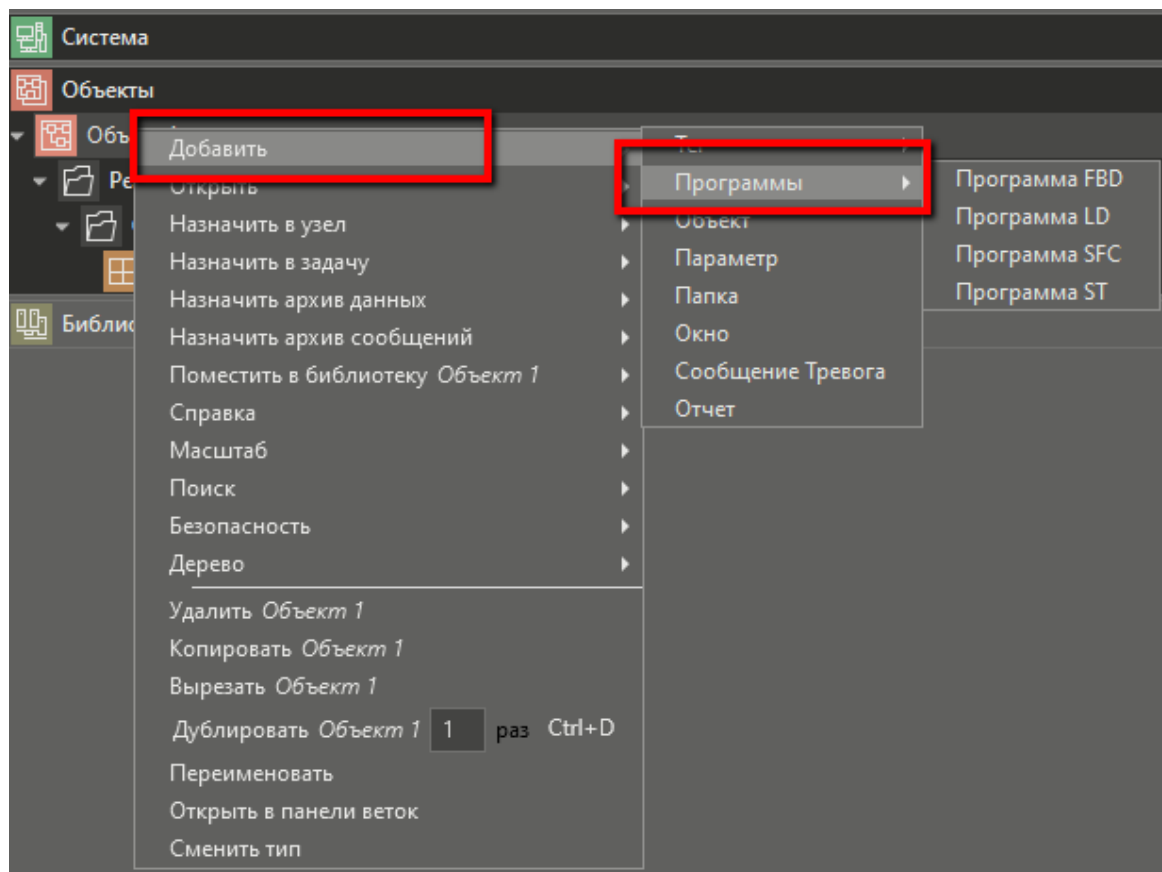
Свойство	Назначение
Период	Период, с которым будут передаваться данные от основного узла резервному.

**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.

## 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**

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 перетащить папку с параметрами, то имена созданных параметров в программе будут 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.

## 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, 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.

### 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.

### ANY

Позволяет связывать входы и выходы функций и ФБ с параметрами имеющими произвольные типы данных. Если вход функции имеет тип ANY, то при установки связи входа с параметром типа BOOL значение функции будет рассчитываться исходя из реального значения входа, т.е. BOOL.

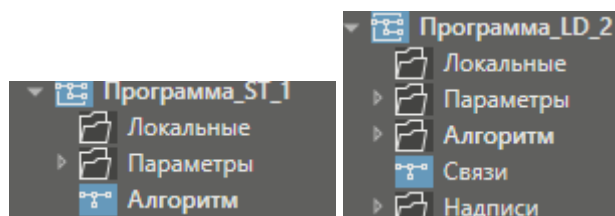
В некоторых случаях используются родовые типы данных, когда необходимо уточнение, какой диапазон типов данных можно использовать. Родовые типы данных идентифицируются использованием префикса ANY.

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.

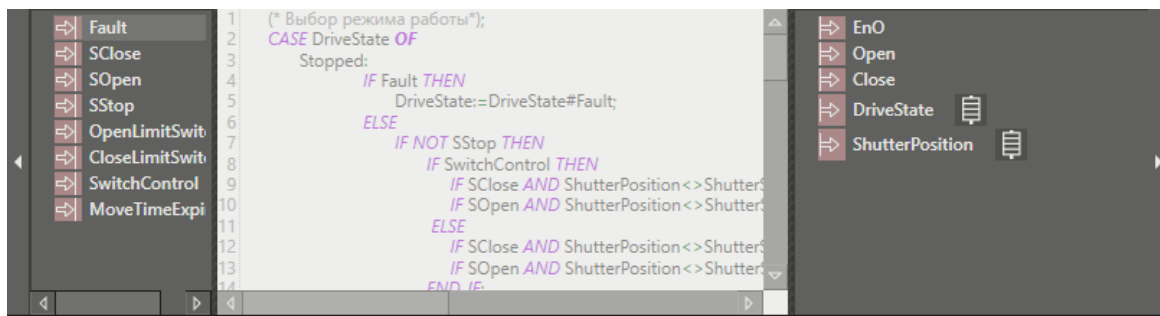
Группы программ:

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, FB D, 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 (см. Инструмент описания SFC-условия). Условие может быть задано на любом языке, кроме SFC.

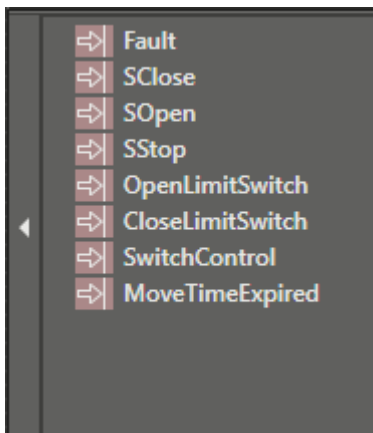
## 7.2.2. Клеммники

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):



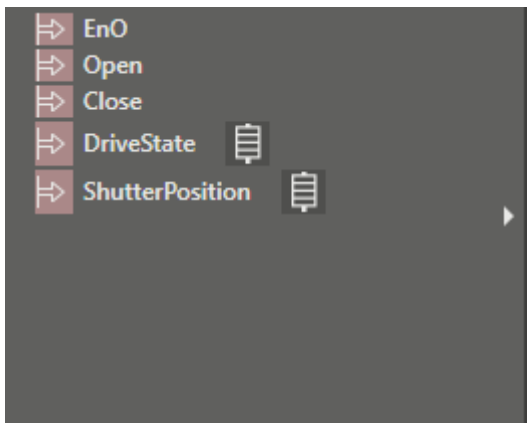
Видимость клеммника определяет кнопкой слева от списка параметров

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:

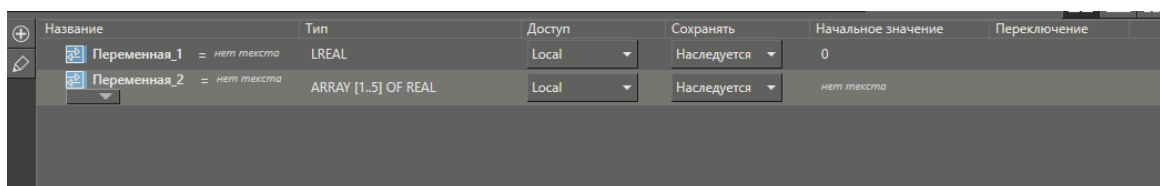
- Легенда. Вкладка 'Локальные';
- Легенда. Вкладка 'Параметры';
- Легенда. Вкладка 'Заголовки ST';
- Легенда. Вкладка 'Ошибки';
- Легенда. Вкладка 'Скомпилированный код'.

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.

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

На этой вкладке отображается заголовок ST-кода, полученного в результате трансляции программы (заголовок содержит операторы определения переменных):


```

1  VAR_OUTPUT
2  EnO: BOOL := True;
3  END_VAR
4
5  VAR_INPUT
6  Параметр_1: LREAL := 0;
7  END_VAR
8
9

```

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.

Поскольку переменные могут быть добавлены/удалены как на данной вкладке, так и в дереве, то при нажатии на кнопку  происходит синхронизация объявленных параметров.

#### 7.2.3.3.1. Легенда. Вкладка Ошибки

На вкладке отображаются ошибки, которые были обнаружены при компиляции программы.

№	Сообщение	Детали	Источник	Тип	Описание
0	Ожидается определение переменной	Положение: 3;1	Код: END_VAR Программа_ST_1	Ошибка	

В отличие от таблицы в окне ошибок, строки таблицы на вкладке не имеют контекстного меню.

#### 7.2.3.4. Легенда. Вкладка Скомпилированный

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.

### 7.2.3.5. Легенда. Вкладка Элементы

На этой вкладке отображаются элементы программ, написанных в редакторах LD, SFC, FBD.

Название	Тип	Декларация	Координата X	Координата Y
COS	COS	COS	177	577
NOT	NOT	NOT	170	280
R_TRIG	R_TRIG	R_TRIG_1	195	408
RS	RS	RS_1	381	540

### 7.2.3.6. 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	

### 7.2.3.7. Легенда. Вкладка Палитра

Вкладка Палитра легенды содержит элементы, характерные для того или иного редактора.

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).

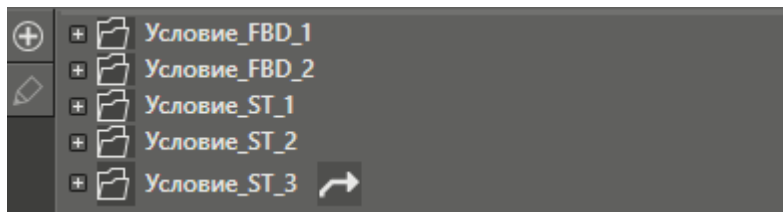
### 7.2.3.8. Легенда. Вкладка Действия

На этой вкладке отображаются уже созданные действия в редакторе SFC, их можно отредактировать, а также создать новые.



### 7.2.3.9. Легенда. Вкладка Условия

На этой вкладке отображаются уже созданные условия в редакторе SFC, их можно отредактировать, а также создать новые.



### 7.2.3.10. 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	Остановка	100	800
	Надпись 3	Выбор условий	60	560

## 7.2.4. Редактирование графических языков

### 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

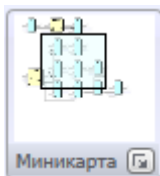
Помимо средств прокрутки, которые используются во всех окнах редактора проекта (см. Прокрутка), при редактировании мнемосхем, FBD-, LD- и SFC-диаграмм могут использоваться дополнительные средства прокрутки:

- 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').

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.

## 7.3. 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 ST представляет собой текстовый редактор, для масштабирования в редакторе используются стандартные операции (см. Масштабирование в редакторе проекта).

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. Вызов функций и функциональных блоков

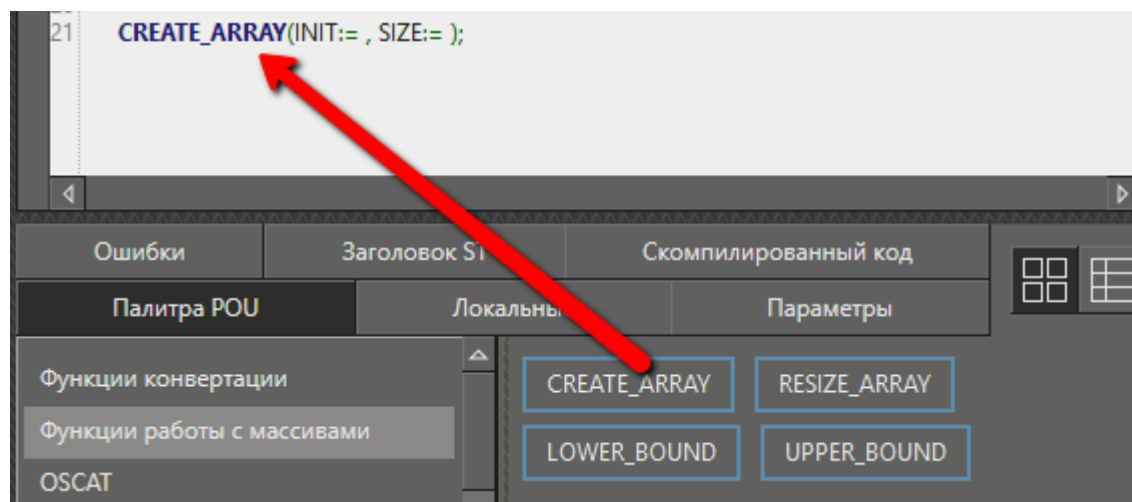
Вызов функций

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.

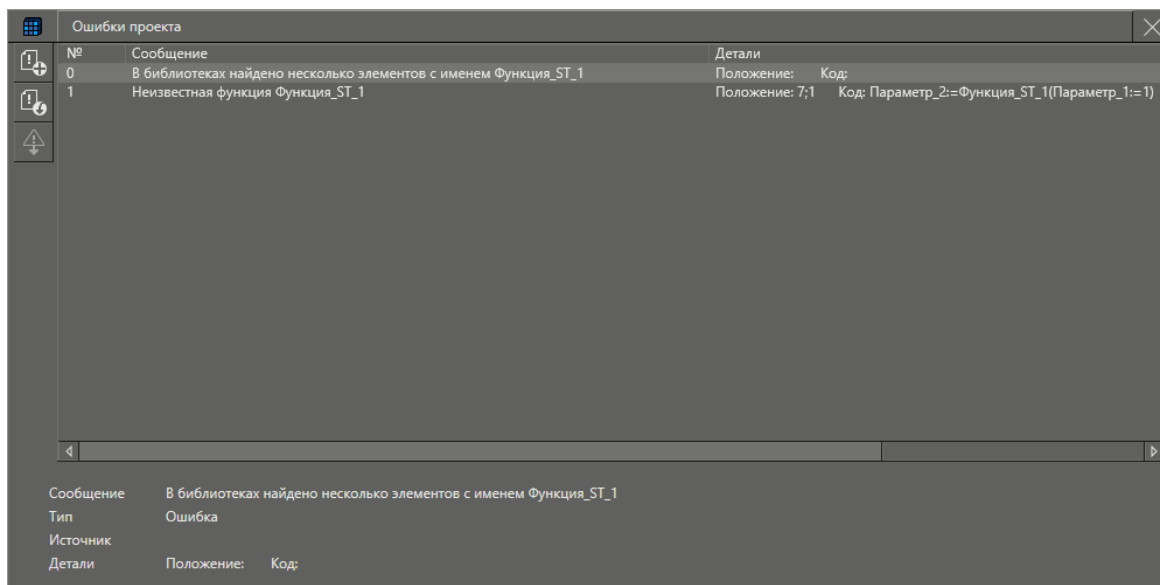
Функция возвращает значение того типа данных, которое задано для ее выхода Q.

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`;

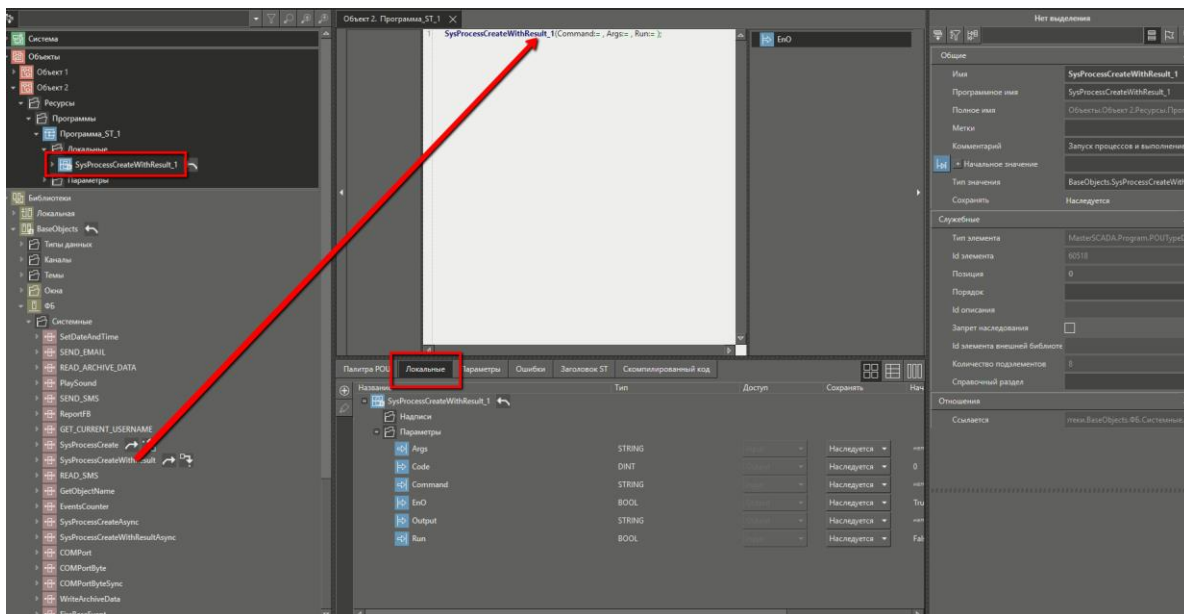
`mREAL2:=myADD(x:=(3+mREAL), y:=3)` - вход `x` пользовательской функции `myADD` рассчитывается исходя из значения параметра программы `mREAL`, значение входа `y` не меняется, после вычисления значения функции оно присваивается параметру программы `mREAL2`;

`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`

Вызов ФБ


В отличие от функций в программах `ST` вызываются не типы, а экземпляры.

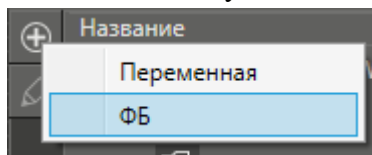
Для того чтобы создать экземпляр ФБ в программе нужно перетащить тип ФБ из Палитры в поле программы. Также можно перетащить нужный функциональный блок из дерева библиотек:



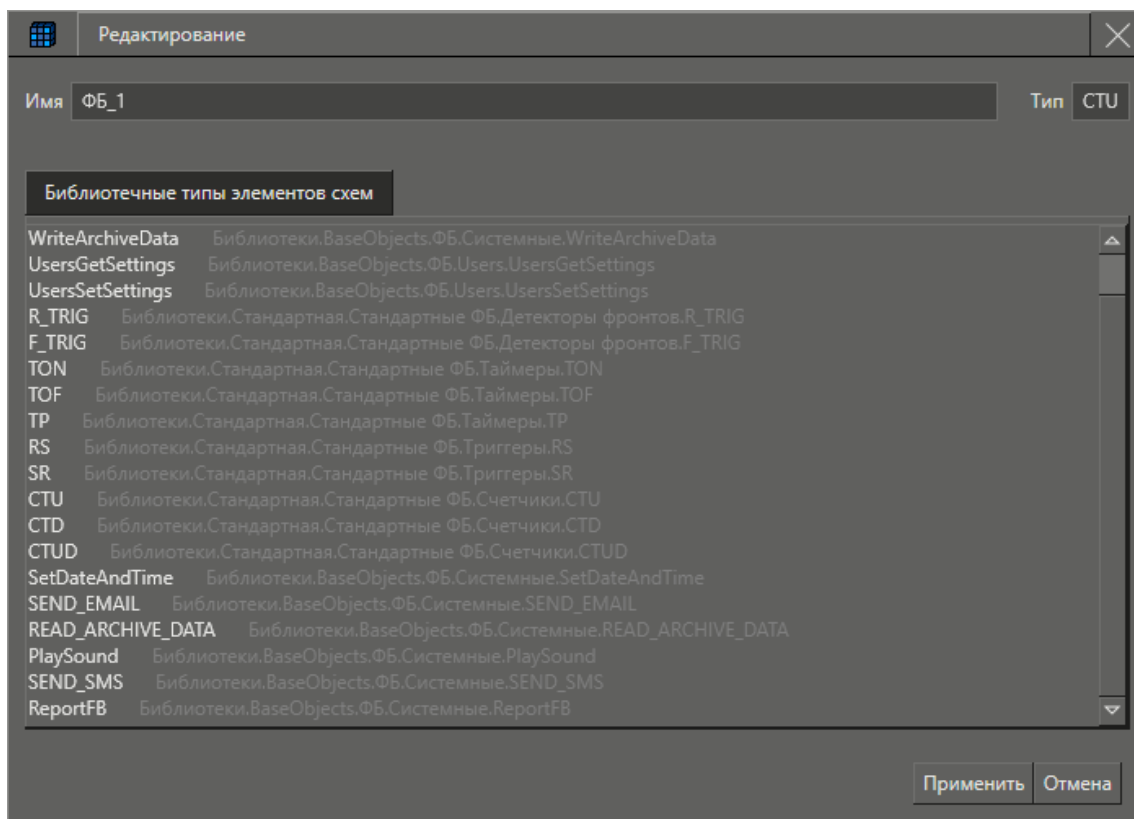
Функциональный блок добавится в текст программы со всеми входами. Затем эти входы можно задать, например, указать константу, либо перетащить в нужное место локальную переменную программы или параметр программы из клеммника входа. При этом видно, что в группе Локальные в дереве и в одноименной панели 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`.

## 7.3.2. 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:

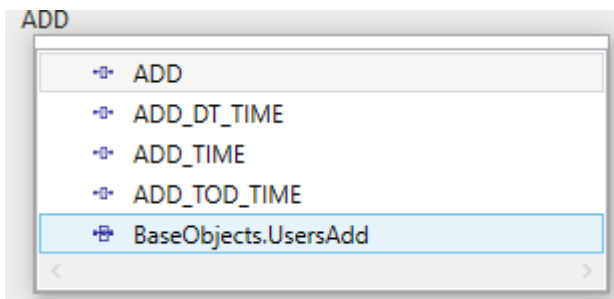
```
USINT_TO_BCD_BYTE(25) = 2#0010_0101
```

```
WORD_BCD_TO_UINT (2#0011_0110_1001) = 369
```

### 7.3.3.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`.



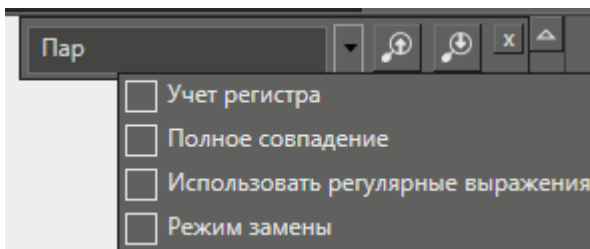
Чтобы из подсказчика вставить в `ST`-код переменную/функцию, нужно выбрать ее в списке при помощи мыши или клавиш курсора и кнопки `Enter`.

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.

### 7.3.4. Поиск и замена в тексте ST

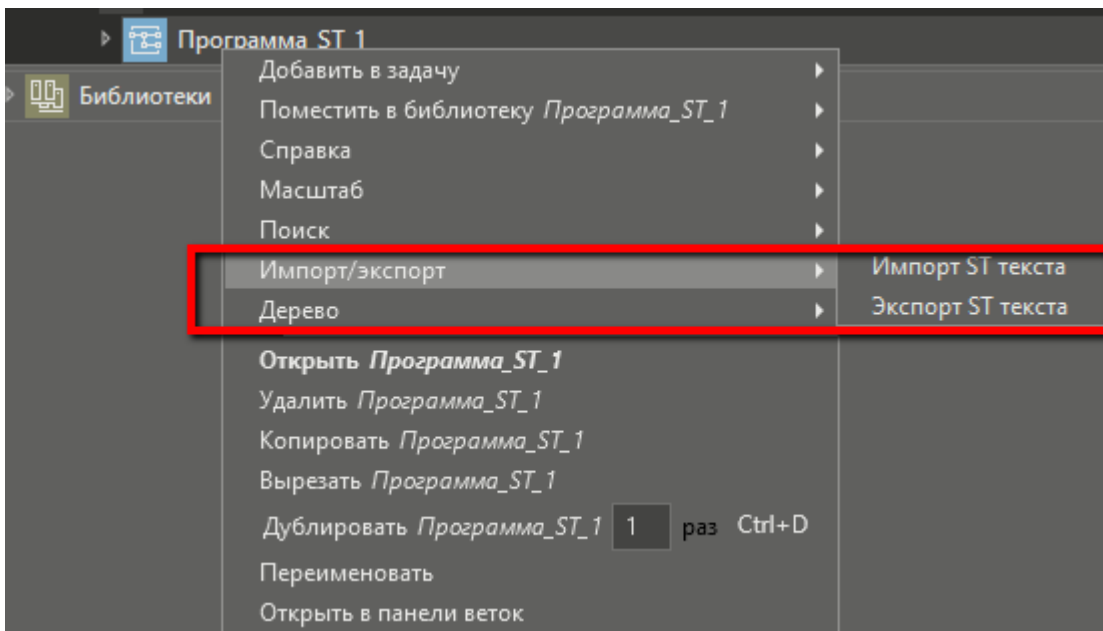
Чтобы открыть диалог поиска необходимо нажать комбинацию клавиш CTRL+F. Диалог появится в верхнем левом углу.



### 7.3.5. Экспорт и импорт программ ST

Допустим импорт и экспорт программ, функциональных блоков и функций, которые созданы на языке Master ST.

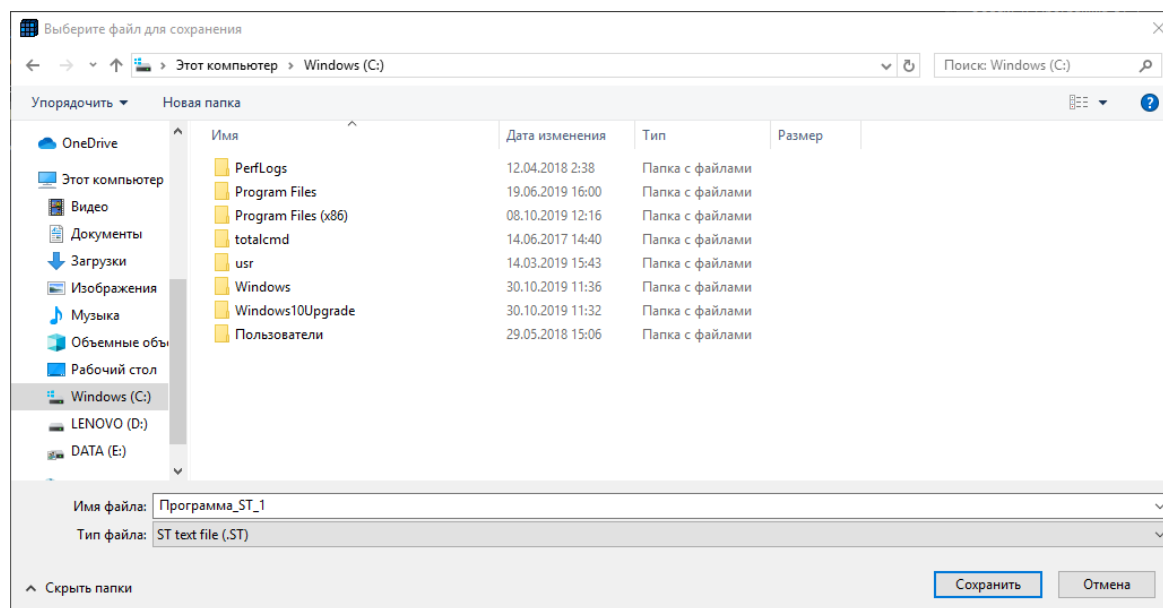
В контекстном меню этих элементов есть соответствующие пункты:



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 создается 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).

## 7.3.6. 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.

### 7.3.6.1. Операторы Master ST

#### 7.3.6.1.1. 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;
```

#### 7.3.6.1.2. 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}  
elsif {выражение2} then {последовательность предложений2}  
...  
elsif {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;  
elsif x>10 and x<20 then y:=15;  
elsif x=20 then y:=20;  
else y:=0;  
end_if;
```

### 7.3.6.1.3. 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;

```

### 7.3.6.1.4. 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;
```

### 7.3.6.1.5. 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;
```

### 7.3.6.1.6. 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;
```

### 7.3.6.1.7. 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;
```

### 7.3.6.1.8. 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 дампы 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' ).

### 7.3.6.1.9. 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;
```

### 7.3.6.2. 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 исключаящее ИЛИ	XOR	
Побитовое ИЛИ	OR	
Присвоение	:=	Низший

### 7.3.6.3. Особенности вычислений

#### 7.3.6.3.1. 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  $b - a = 16\#FFFFFFFF - a + b + 1$ .

В частности, это свойство используется для корректного вычисления длины временного интервала по отсчетам, полученным с помощью функции TIME, таймер ПЛК.

#### 7.3.6.3.2. 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;
```

### 7.3.6.4. Константы Master ST

#### 7.3.6.4.1. 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.

### 7.3.6.4.2. 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-мм-дд` (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).

### 7.3.6.4.3. 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

N	Пример	Описание
N		
1	Строки 1-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

	'\$0A'	Строка длины 1, содержащая символ LF
	'\$\$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\$00C B"	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

N	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.

### 7.3.6.5. 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.

### 7.3.6.5.1. Идентификаторы

Идентификаторы 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.

### 7.3.6.5.2. Разделители

В качестве разделителей в Master ST используются следующие лексемы:

(\* \*) + - # . e E ' \$ : := () [] , ; .. % => \*\* \* / < > <= >= = <> &

### 7.3.6.5.3. 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

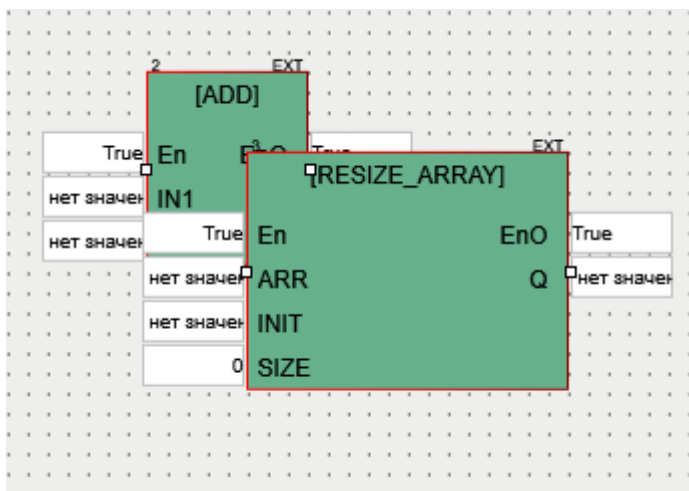
На языке FBD могут быть разработаны программы, функции и функциональные блоки.

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);
- задать переменные (см. Операции с переменными);
- 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
-----------	--

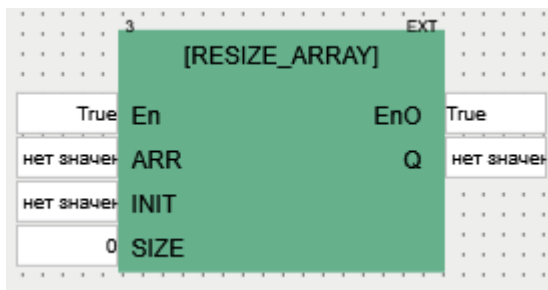
Название	Назначение
Координата X	Определяет положение верхнего левого угла блока в рх. Если установлено значение 0, то это значит блок занимает крайнее левое положение. Положение FBD-блока можно также изменять с помощью мыши
Координата Y	Определяет положение верхнего левого угла блока в рх. Если установлено 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.

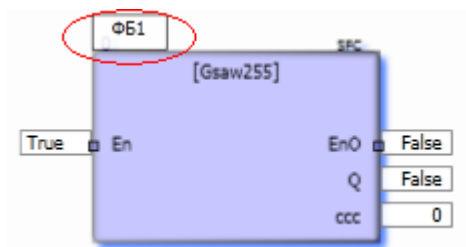
Буквы над блоком справа указывают язык, на котором написан вызываемый блок (например, ST, FBD, LD, SFC). Для встроенных элементов (библиотечных) справа над блоком выводится EXT (как на рисунке).

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, перетащив одно на другое левой кнопкой мыши, в этом случае связь отобразится и на схеме.
Группа Разметка	
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

### 7.4.3. 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.

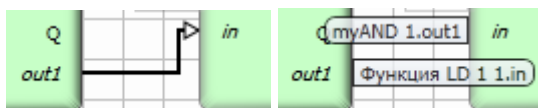
Слой может находиться в активном и неактивном состоянии. Элементы неактивных слоев не реагируют на действия мыши.

Для управления видимостью и активностью слоев предусмотрены инструменты, которые содержит панель Слои.

### 7.4.4. Операции со связями FBD-диаграммы

#### 7.4.4.1. Отображение связей

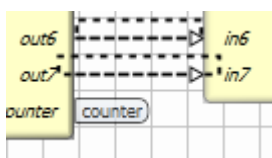
Каждая межблочная связь на FBD-диаграмме может быть независимо отображена как линия со стрелкой (стрелка указывает направление передачи значения) или как ссылка:



Связь с аргументом программного компонента всегда отображается как ссылка.

Для переключения отображения в контекстном меню связи предусмотрена соответствующая команда ('Отобразить ссылкой', если связь отображается как линия, или 'Отобразить линией', если связь отображается как ссылка). Для переключения отображения связи может также использоваться флаг Разъединить. Этот флаг содержит столбец 'Разъединить' редактора связей и Легенда. Вкладка 'Связи', а также окно 'Свойства'.

Связь отображается пунктирной линией, если редактору не удалось проложить ее корректно:



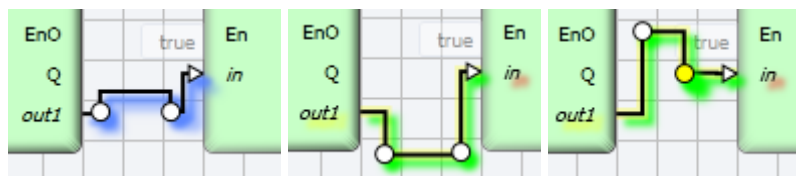
Для исправления подобной ситуации нужно использовать маршрутные точки.

#### 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:

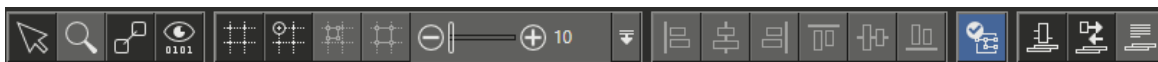
- двойной щелчок некоторой точки связи – установка маршрутной точки в данную точку связи;
- двойной щелчок маршрутной точки – удаление этой маршрутной точки.

’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)
	Управляет видимостью слоя связей
	Управляет видимостью слоя надписей

### 7.4.5.2. Панель Режимы





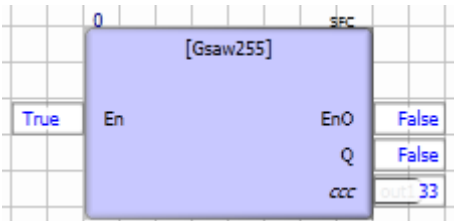
Панель определяет поведение схемы при работе с мышью



Доступны следующие режимы:






Режим	Описание
Выделение рамкой	<p>Чтобы перейти в этот режим, нужно щелкнуть кнопку </p> <p>(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 </p> <p>(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>



	Для перехода в данный режим на короткое время нужно нажать и удерживать левую кнопку SHIFT.
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



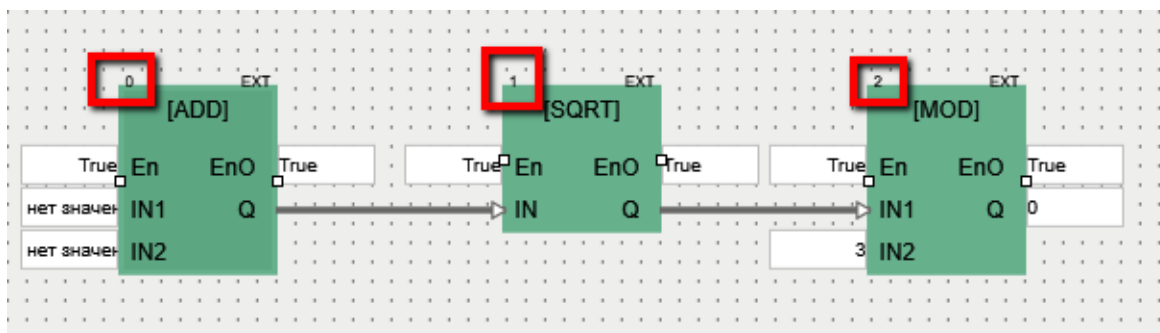
Инструменты этой панели доступны при выделении не менее двух элементов схемы.

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

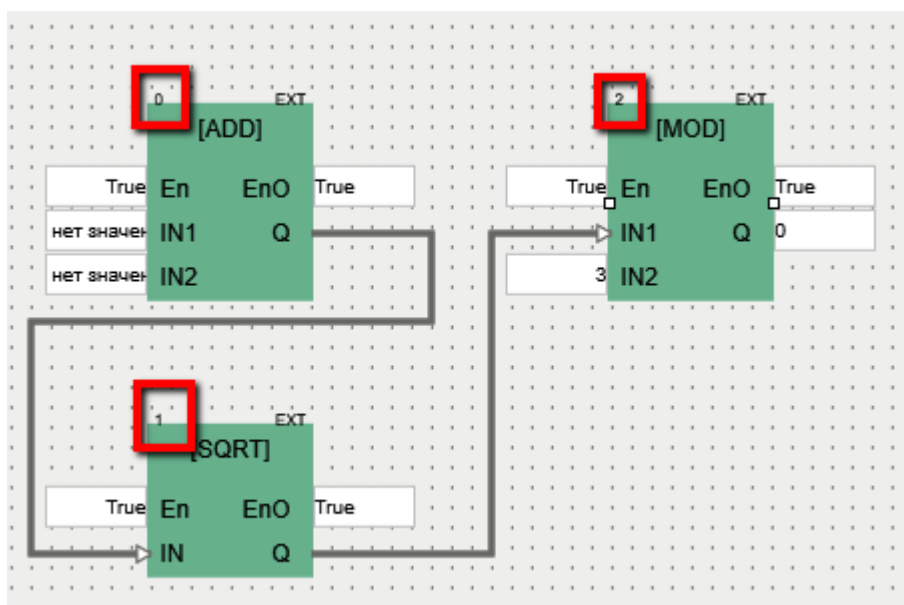
### 7.4.5.5. Порядок исполнения FBD-блоков

Порядок исполнения FBD-блоков зависит от их положения на схеме. 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)

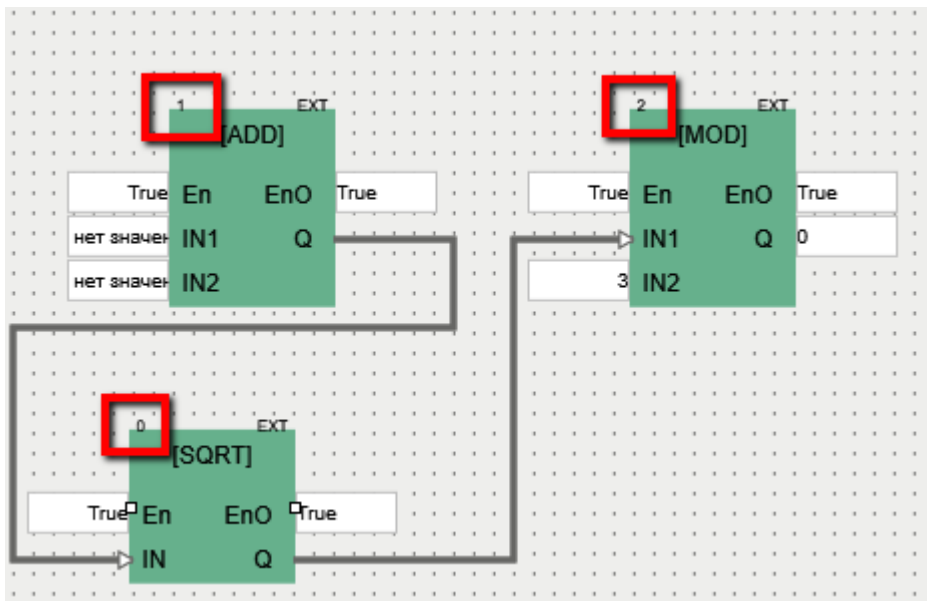
Порядковый номер исполнения блока отображается над блоком слева:



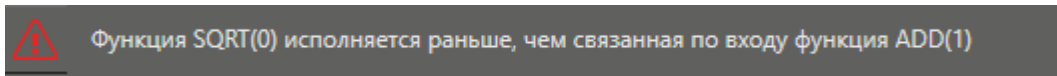
На приведенном выше рисунке видно, что сначала выполняется блок ADD, затем SQRT и далее MOD. Такой же порядок исполнения будет и в таком случае:



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:

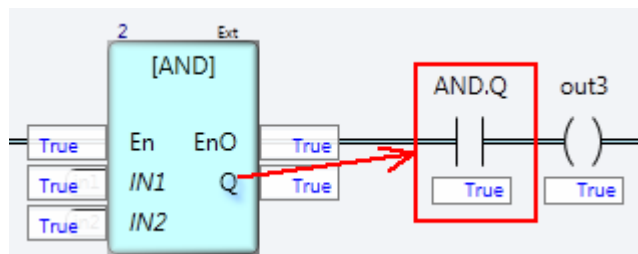
- меню LD-редактора;
- инструменты и операции масштабирования (см. Масштабирование в редакторе проекта и Масштабирование диаграмм);
- общие команды редактирования;
- прокрутка диаграмм;

- 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;

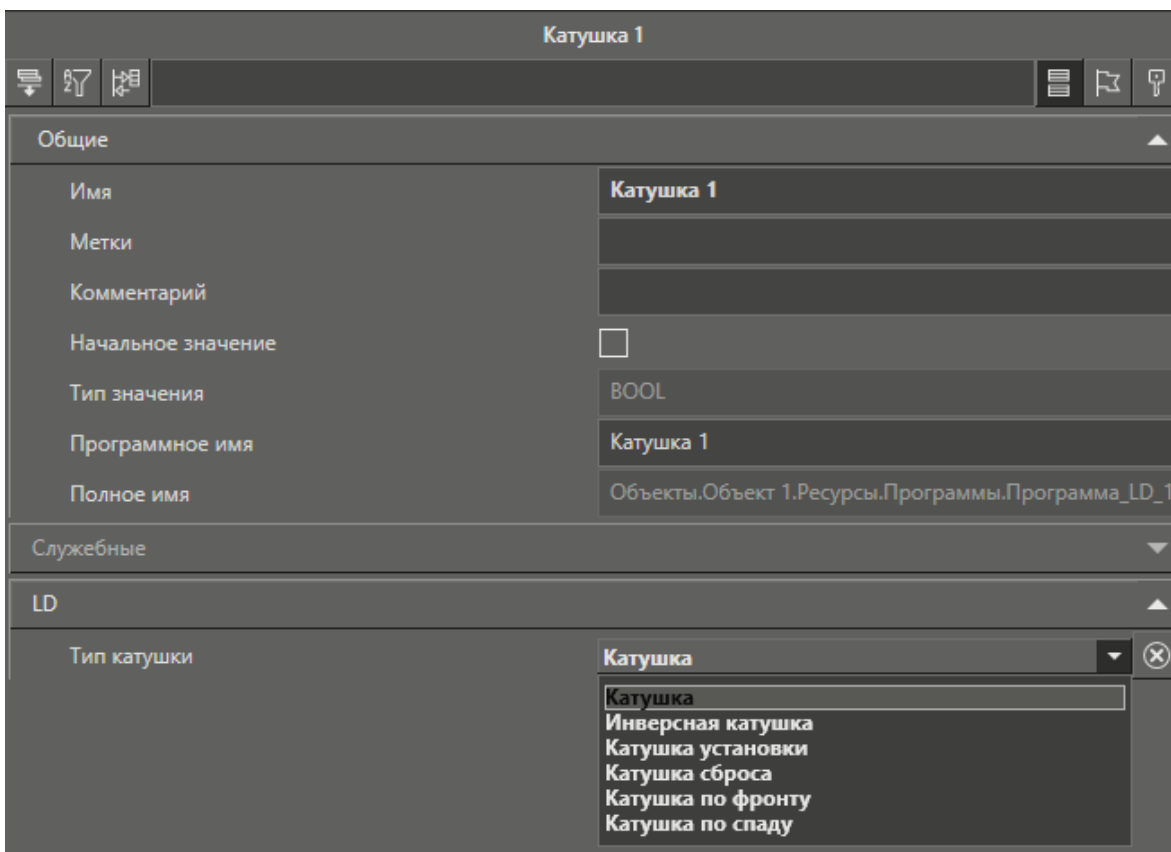


### 7.5.2.2. Перемещение элемента LD-алгоритма

Для перемещения элемента алгоритма после его размещения на диаграмме используется перетаскивание.

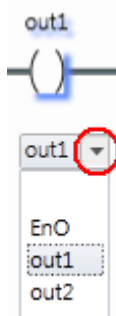
### 7.5.2.3. Изменение типа контакта/катушки после размещения


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:

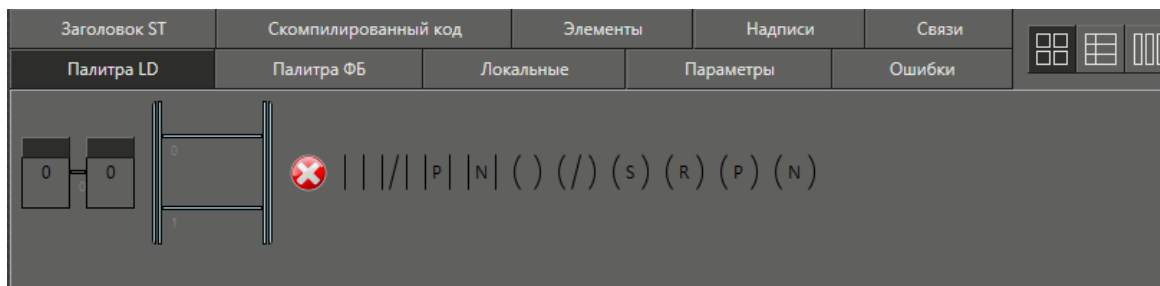


Вид	Описание
	Панель Режимы . Переключает режимы работы в редакторе
	Запускает проверку текущей программы.
	Панель Слои.

### 7.5.4. Палитра LD

Легенда LD-редактора содержит вкладку Палитра LD, которая содержит все элементы библиотеки Стандартная, категория LD.



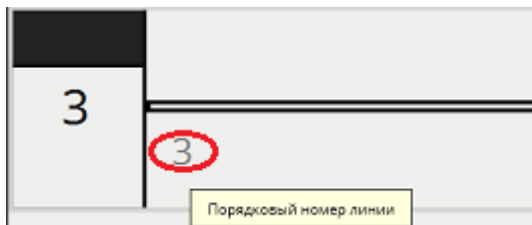


### 7.5.4.1. Силовая линия

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):

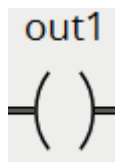


Значение левой шины всегда равно 1 (аналог положительной шины питания).

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):



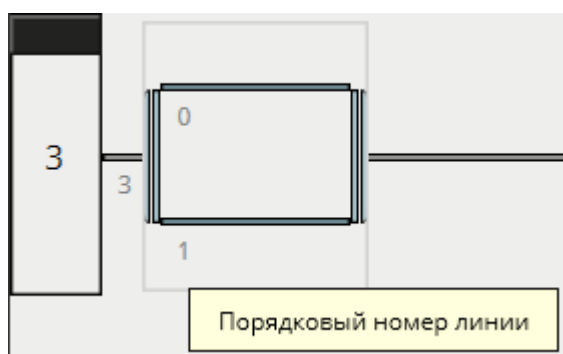
Отрезок слева обозначает вход блока, отрезок справа – выход. Все встроенные LD-функции имеют один вход (in) и один выход (out).

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



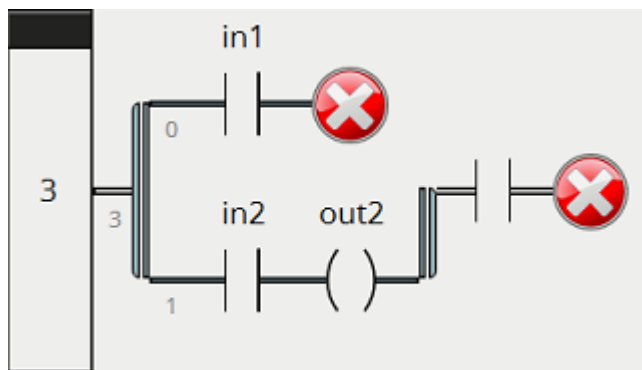
Ветвление содержит две параллельные вспомогательные силовые линии, заключенные между левой и правой вспомогательными шинами (см. силовая линия). Вспомогательная силовая линия имеет номер <номер основной силовой линии>.<номер вспомогательной силовой линии> (2.2 на рисунке выше).

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):

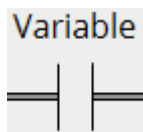


### 7.5.4.5. Другие элементы LD

- 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.

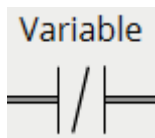
Контакт  $||$



Если Variable  $\triangleleft$  0, то out = in.

Если Variable = 0, то out = 0.

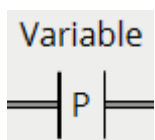
Замкнутый контакт  $|/|$



Если Variable = 0, то out = in.

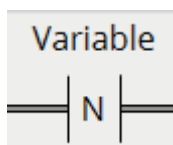
Если Variable  $\neq$  0, то out = 0.

Контакт по фронту |P|



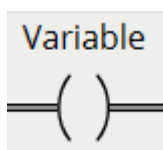
Если in = 1, а Variable меняет свое значение с 0 на любое ненулевое, то на один (следующий) такт out = 1. Во всех остальных случаях out = 0.

Контакт по спаду |N|



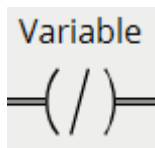
Если in = 1, а Variable меняет свое значение с любого ненулевого на 0, то на один (следующий) такт out = 1. Во всех остальных случаях out = 0.

Катушка ( )



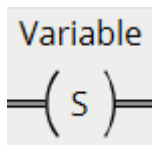
Variable = out = in.

Инверсная катушка (/)



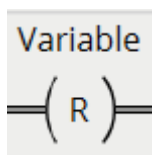
out = in, Variable = NOT in.

Катушка установки (S)



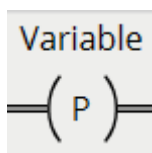
Переменная Variable принимает значение 1 при подаче 1 на вход и в дальнейшем может быть сброшена только с помощью катушки сброса. Во всех случаях  $out = 1$ .

Катушка сброса (R)



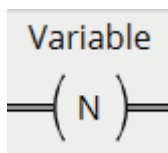
Переменная Variable принимает значение 0 при подаче 1 на вход и в дальнейшем может быть установлена только с помощью катушки установки. Во всех случаях  $out = 1$ .

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').

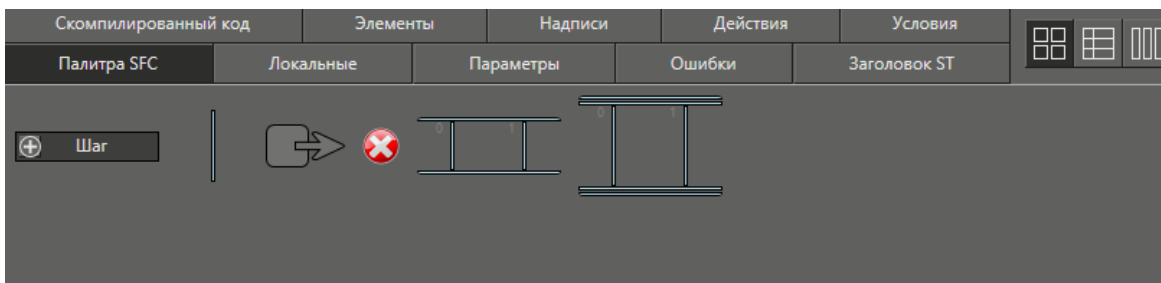
### 7.6.3. Панель инструментов SFC-редактора

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.



#### 7.6.4.1.1. Библиотека SFC

Содержит элементы SFC-алгоритма:

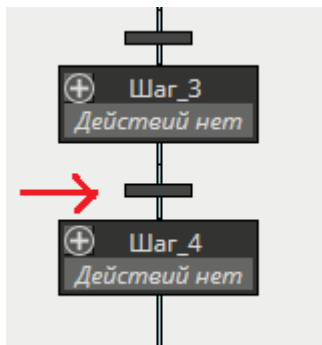
- SFC-ветка;
- SFC-шаг;

- SFC-переход;
- SFC-терминатор;
- SFC-ветвление выбора;
- параллельное SFC-ветвление.


#### 7.6.4.1.1. SFC-ветка

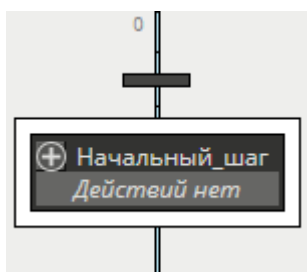
SFC-ветка – это часть SFC-алгоритма с непосредственными связями между элементами (связи образуются автоматически при размещении элементов алгоритма на ветке).

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.




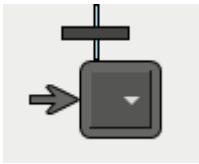
### 7.6.4.1.3. SFC-условие

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).

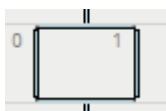
### 7.6.4.1.5. SFC-терминатор



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

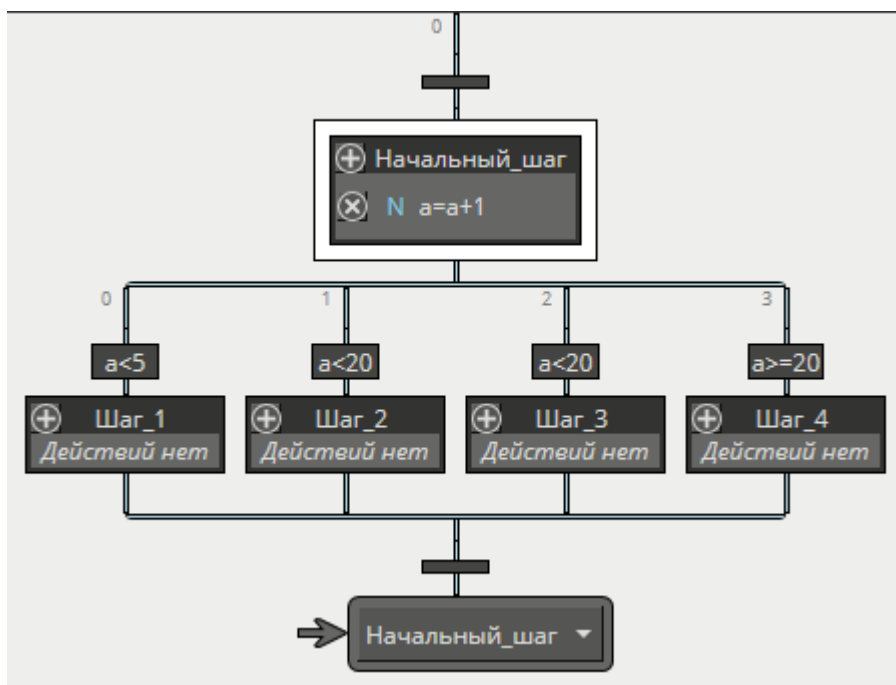
При размещении данный элемент имеет две пустые ветки (вертикальные линии):



С помощью команды 'Добавить.Ветка' контекстного меню горизонтальной линией элемента число веток может быть увеличено:



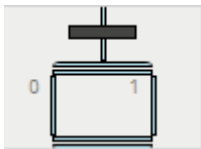
Ветки элемента анализируются слева направо, и выполняется первая (и единственная) ветка, для которой заданное условие истинно. Например, на следующей диаграмме переход к шагу 3 не произойдет никогда:



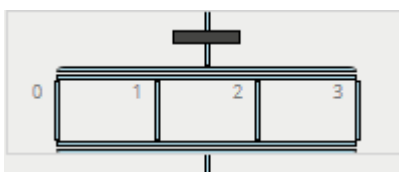
У SFC-ветвления выбора отсутствует предшествующее SFC-условие.

### 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-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.

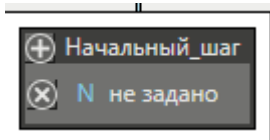
### Способ 2

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  ;

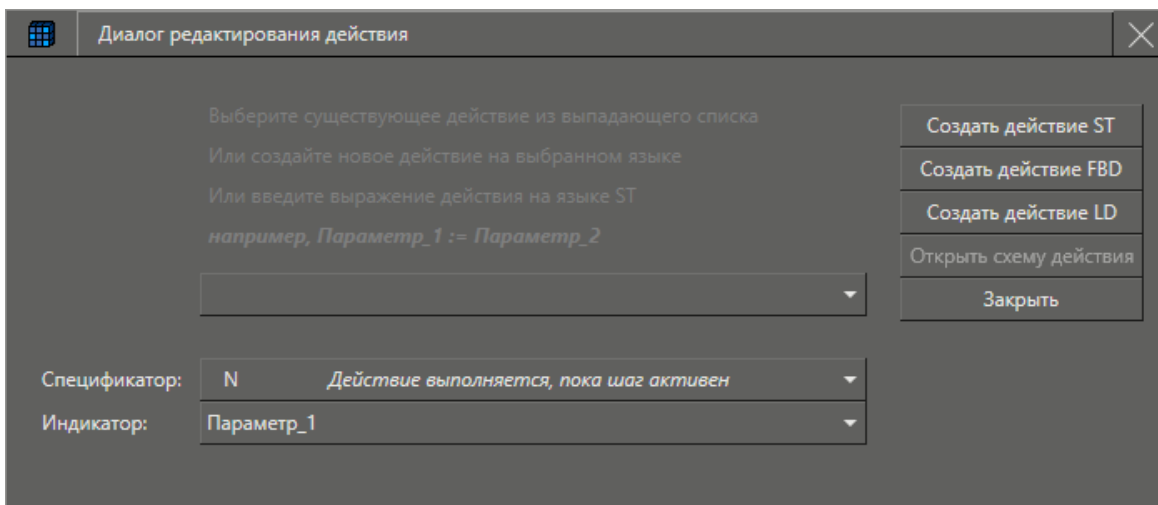
- выполнить двойной клик в строке действия поле описания ( **не задано** ) – откроется диалог редактирования SFC-действия – и сконфигурировать созданное действие в диалоге. (диалог открывается также по команде Редактировать контекстного меню Действия)

Инструмент удаления SFC-действия

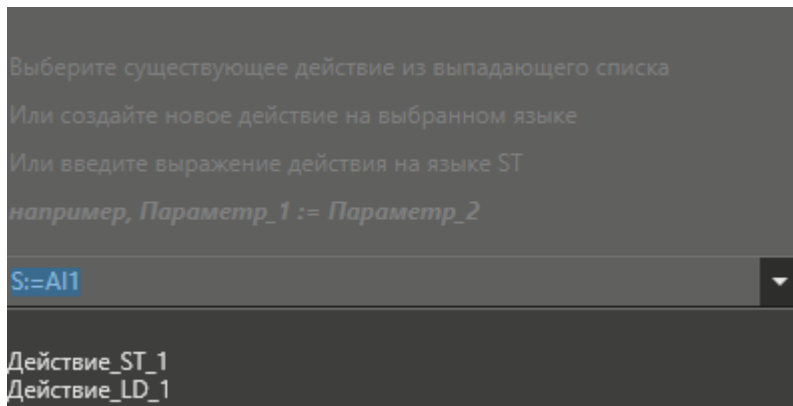
-  – удалить SFC-действие. Удаление SFC-действия помещается в стек UnDo и может быть отменено.

### 7.6.5.1. Диалог редактирования SFC-действия

Диалог имеет вид:



Инструмент описания SFC-действия

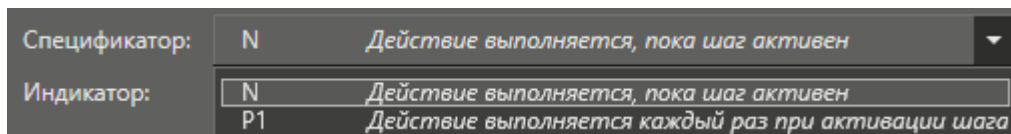


Выпадающий список действий, которые содержит корневая группа 'Действия' программного компонента, позволяет использовать действия, созданные предварительно.

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

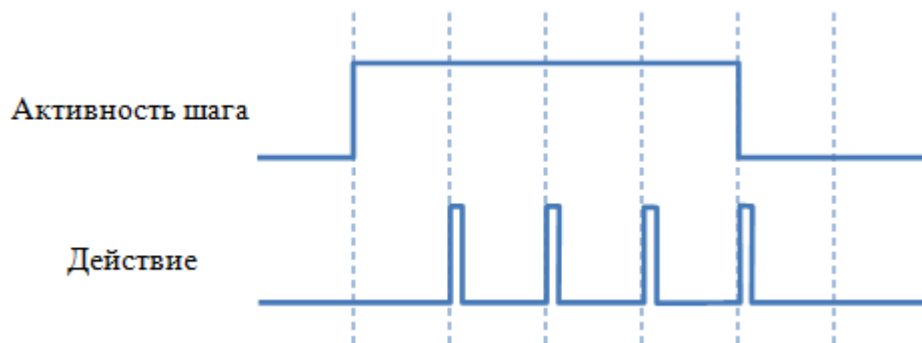


Выпадающий список, в котором выбирается спецификатор действия (action qualifier по IEC 61131-3).

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.

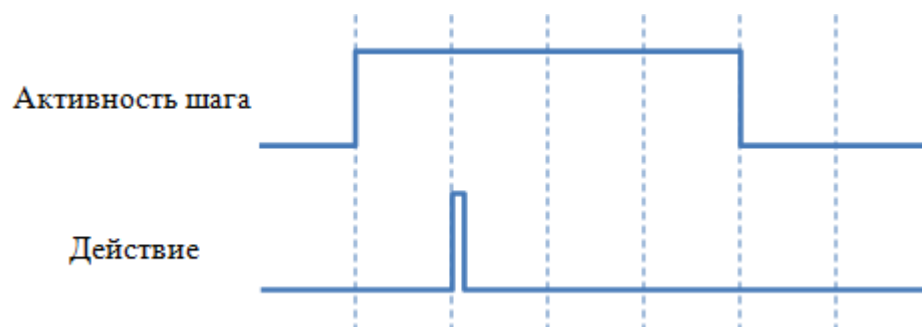


Работает в соответствии с определением стандарта, но со сдвигом на один цикл.

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



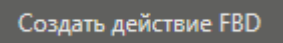
Выпадающий список, в котором выбирается индикатор действия (переменная).

Значение индикатора равно 1 (TRUE), если действие выполняется, или 0 (FALSE) в противном случае.

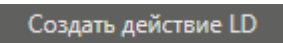
Кнопка создания действия ST

- 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

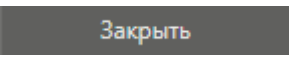

-  – создать SFC-действие на языке LD в корневой группе Действия программного компонента и использовать это действие.

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.

## 7.6.6.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

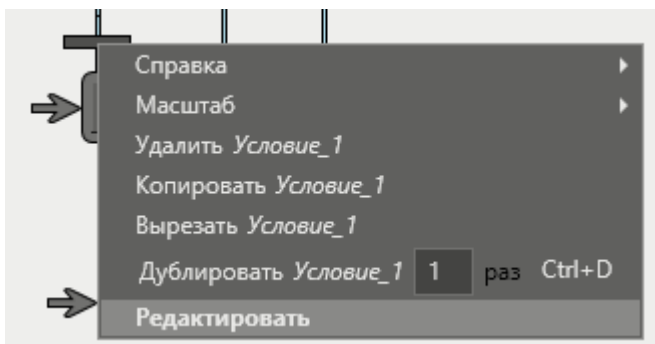
Для задания SFC-условия можно создать условие предварительно в корневой группе Условия программного компонента (см. Структура программы в дереве) и перетащить его на ветку или условие в диаграмме (в первом случае условие будет добавлено, во втором – заменено).

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).

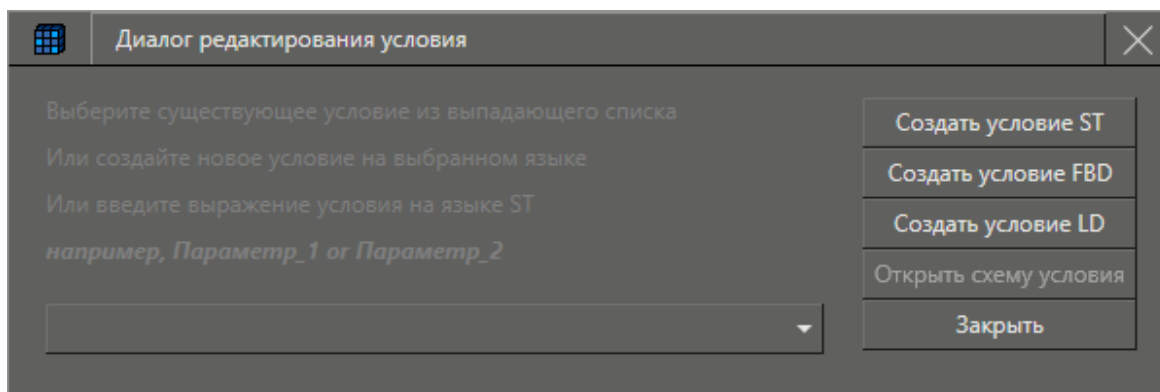
### Способ 3

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:

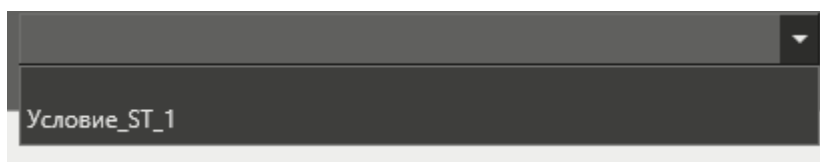


#### 7.6.6.1. Диалог редактирования SFC-условия

Диалог имеет вид



Инструмент описания SFC-условия



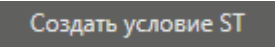
Выпадающий список SFC-условий, которые содержит корневая группа Условия программного компонента, позволяет использовать условия, созданные предварительно.

В группе Алгоритм.<ветка>.Шаг\_<n>.Условия отображается ссылка на использованное условие.

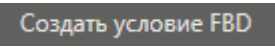
Окно элемента управления списком представляет собой поле редактирования, в котором условие (логическое выражение) может быть задано вручную на языке ST.



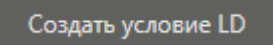
Кнопка создания условия ST

-  – 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

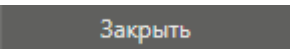

-  – создать SFC-условие на языке LD в корневой группе Условия программного компонента и использовать это условие.

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.

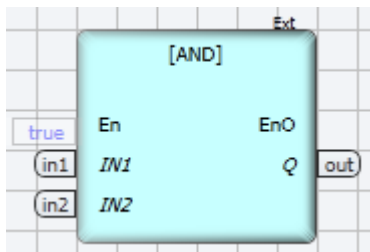
## 7.7. Функции и ФБ библиотеки Стандартная

### 7.7.1. Библиотека Стандартные функции

#### 7.7.1.1. Булевы функции

- AND, побитовое И
- OR, побитовое ИЛИ
- XOR, побитовое исключающее ИЛИ
- NOT, побитовое НЕ

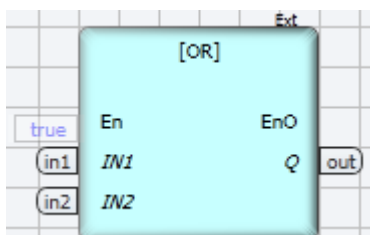
AND, побитовое И



$Q := IN1 \& IN2$

Например,  $6 \& 10 = 2$ .

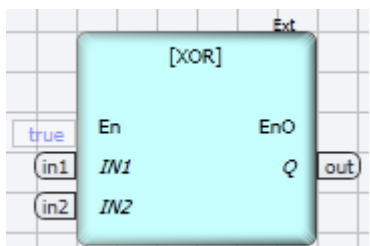
OR, побитовое ИЛИ



$Q := IN1 \text{ or } IN2$

Например,  $6 \text{ or } 3 = 7$ .

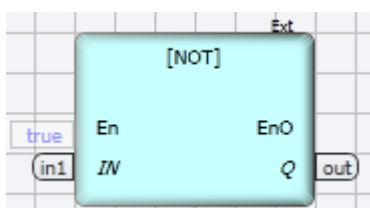
XOR, побитовое исключающее ИЛИ



$Q := IN1 \text{ xor } IN2$

Например,  $6 \text{ xor } 3 = 5$ .

NOT, побитовое НЕ



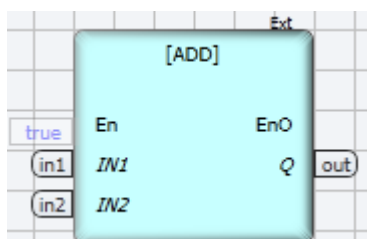
$Q := \text{not } IN$

Например (см. рис.), если тип данных in1 – UINT и in1=0, то out=16#FFFF (тип данных out – UINT, UDINT или ULINT).

### 7.7.1.2. Математические функции

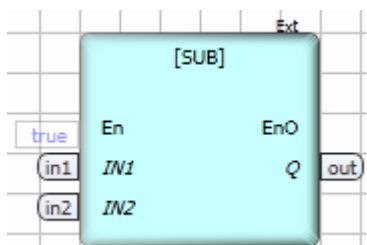
- ADD, сложение
- SUB, вычитание
- MUL, умножение
- DIV, деление
- MOD, остаток от деления
- EXPT, возведение в степень
- MOVE, передача значения
- ABS, абсолютная величина
- TRUNC, целая часть к нулю
- SQRT, квадратный корень

ADD, сложение



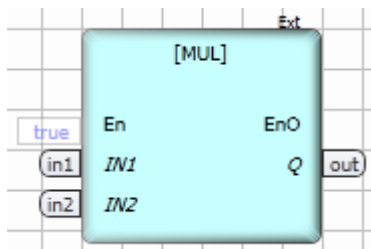
$Q := IN1 + IN2$

SUB, вычитание



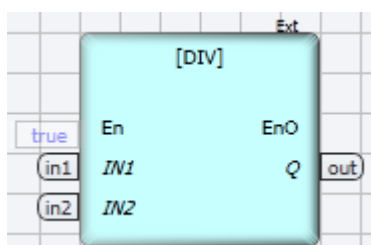
$Q := IN1 - IN2$

MUL, умножение



$Q := IN1 * IN2$

DIV, деление

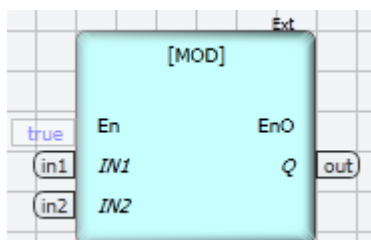


$Q := IN1 / IN2$

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

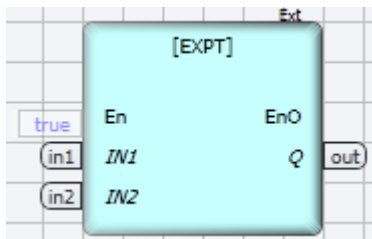


Функция возвращает остаток от деления целых чисел (IN1 на IN2):

$Q := \text{mod}(IN1, IN2)$

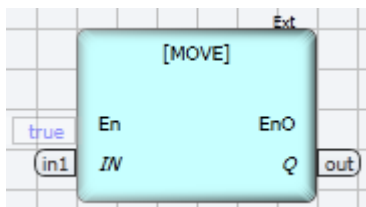
Например,  $\text{mod}(8, 3) = 2$ .

EXPT, возведение в степень



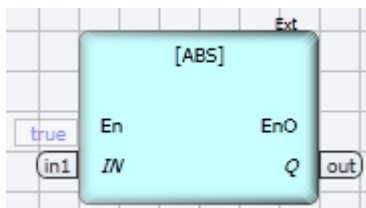
$Q := IN1 ** IN2$

MOVE, передача значения



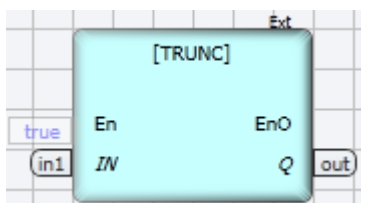
$Q := IN$

ABS, абсолютная величина



$Q := \text{abs}(IN)$

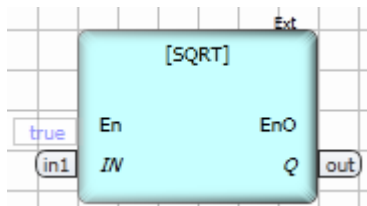
TRUNC, целая часть к нулю



$Q := \text{trunc}(IN)$

Например,  $\text{trunc}(5.9)=\text{trunc}(5.1)=5$ ;  $\text{trunc}(-5.9)=\text{trunc}(-5.1)=-5$ .

SQRT, квадратный корень

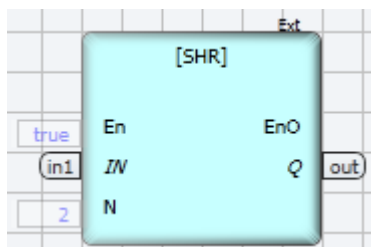


$Q := \text{sqrt}(IN)$

### 7.7.1.3. Функции сдвига

- SHR, сдвиг вправо
- SHL, сдвиг влево
- ROR, циклический сдвиг вправо
- ROL, циклический сдвиг влево

SHR, сдвиг вправо

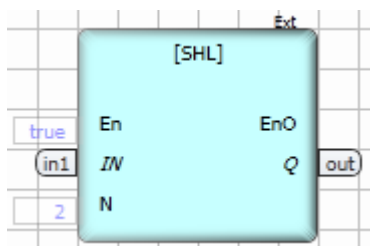


Сдвиг IN вправо на N разрядов, слева число дополняется нулевыми разрядами:

$Q := \text{shr}(IN, N)$

Например,  $\text{shr}(8, 2) = 2$ .

SHL, сдвиг влево

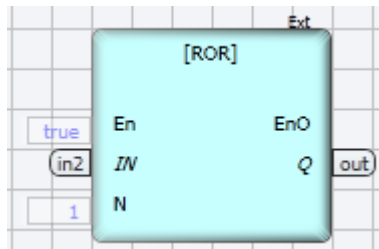


Сдвиг IN влево на N разрядов, справа число дополняется нулевыми разрядами:

$Q := \text{shl}(IN, N)$

Например,  $\text{shl}(2, 2) = 8$ .

ROR, циклический сдвиг вправо

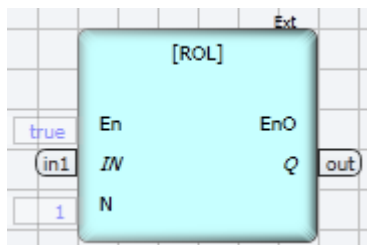


Сдвиг IN вправо на N разрядов. Слева число дополняется разрядами, которые при сдвиге "выбывают" справа:

$Q := \text{ror}(IN, N)$

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:

$Q := \text{rol}(IN, N)$

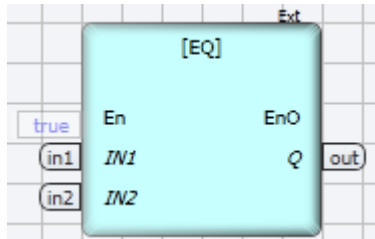
Например (см. рис.), если тип данных in1 – UINT и in1=16#8000, то out=1.

#### 7.7.1.4. Функции сравнения

- EQ, равенство
- NE, неравенство
- LT, меньше
- LE, меньше или равно
- GT, больше

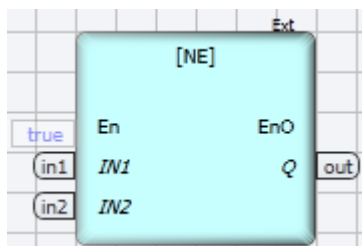
- GE, больше или равно

EQ, равенство



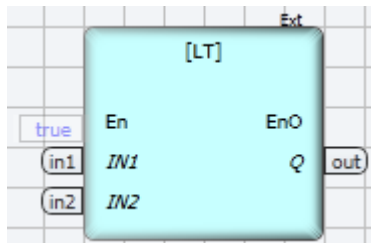
Если  $in1=in2$ , то  $out=1$ , в противном случае  $out=0$ .

NE, неравенство



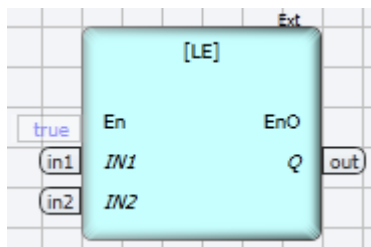
Если  $in1 \neq in2$ , то  $out=1$ , в противном случае  $out=0$ .

LT, меньше



Если  $in1 < in2$ , то  $out=1$ , в противном случае  $out=0$ .

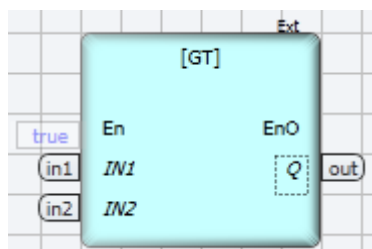
LE, меньше или равно



Если  $in1 \leq in2$ , то  $out=1$ , в противном случае  $out=0$ .

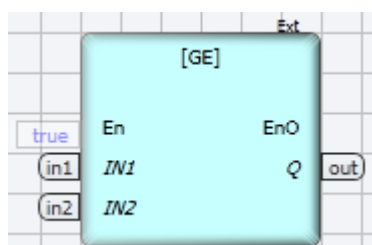


GT, больше



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  $out=0$ .

### 7.7.1.5. 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.

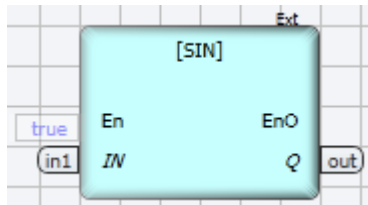
Обратные тригонометрические функции возвращают главное значение соответствующих функций в радианах:

$$-\frac{\pi}{2} \leq \arcsin x \leq \frac{\pi}{2}$$

$$0 \leq \arccos x \leq \pi$$

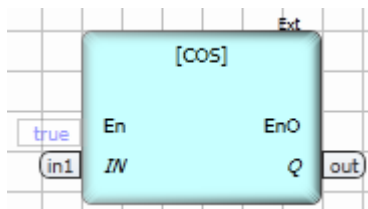
$$-\frac{\pi}{2} \leq \operatorname{arctg}x \leq \frac{\pi}{2}$$

SIN, синус



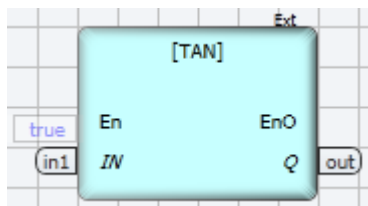
$Q := \sin(IN)$

COS, косинус



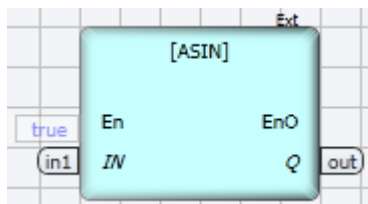
$Q := \cos(IN)$

TAN, тангенс



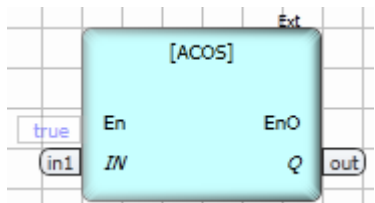
$Q := \operatorname{tg}(IN)$

ASIN, арксинус



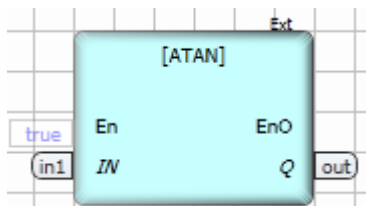
$Q := \operatorname{asin}(IN)$

ACOS, арккосинус



$Q := \text{acos}(IN)$

ATAN, арктангенс

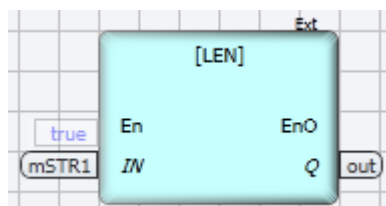


$Q := \text{atan}(IN)$

### 7.7.1.6. Функции работы со строками

- LEN, длина строки
- LEFT, первые символы
- RIGHT, последние символы
- MID, подстрока
- INSERT, вставка строки
- DELETE, удаление подстроки
- REPLACE, замена подстроки
- FIND, поиск подстроки
- CONCAT, конкатенация строк

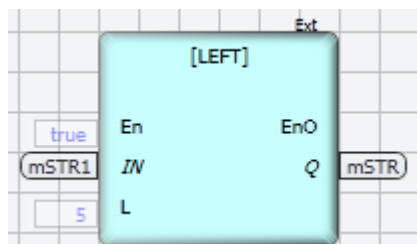
LEN, длина строки



Функция возвращает число символов в строке:

$$Q := \text{len}(\text{IN})$$

LEFT, первые символы

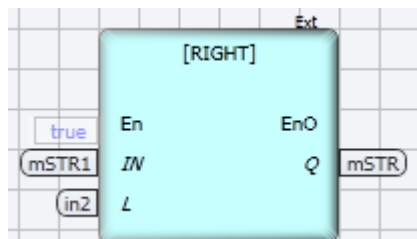


Функция возвращает L первых символов строки IN:

$$Q := \text{left}(\text{IN}, L)$$

Например (см. рис.), если mSTR1="InSAT Company", то mSTR="InSAT".

RIGHT, последние символы

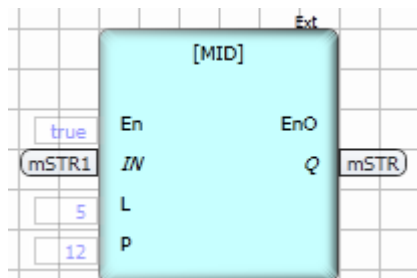


Функция возвращает L последних символов строки IN:

$$Q := \text{right}(\text{IN}, L)$$

Например (см. рис.), если mSTR1="Product of InSAT" и in2=5, то mSTR="InSAT".

MID, подстрока

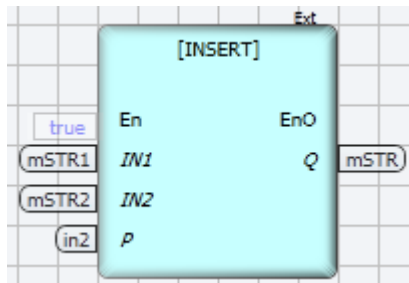


Функция возвращает L символов из строки IN, начиная с P-го символа:

$$Q := \text{mid}(\text{IN}, L, P)$$

Например (см. рис.), если  $mSTR1="Product\ of\ InSAT\ Company"$ , то  $mSTR="InSAT"$ .

INSERT, вставка строки

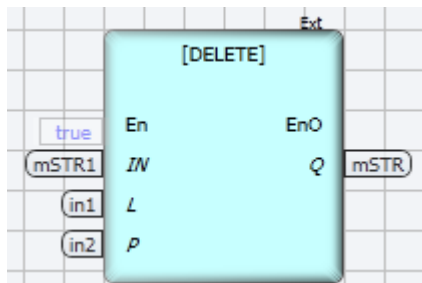


Вставка строки IN2 в строку IN1 после P-го символа:

$Q := insert(IN1, IN2, P)$

Например (см. рис.), если  $mSTR1="Product\ InSAT"$ ,  $mSTR2="of\ "$  и  $in2=8$ , то  $mSTR="Product\ of\ InSAT"$ .

DELETE, удаление подстроки

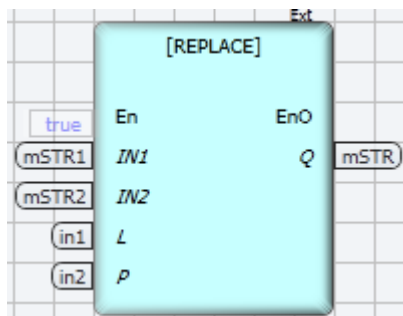


Удаление L символов из строки IN, начиная с P-го символа:

$Q := delete(IN, L, P)$

Например (см. рис.), если  $mSTR1="ABCDEF"$ ,  $in1=4$  и  $in2=2$ , то  $mSTR="AF"$ .

REPLACE, замена подстроки

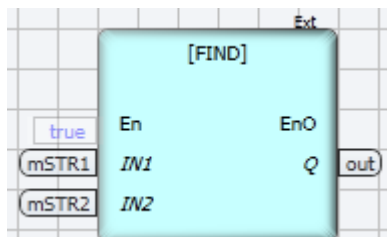


Замена в строке IN1 L символов, начиная с P-го символа, строкой IN2:

$Q := \text{replace}(\text{IN1}, \text{IN2}, L, P)$

Например (см. рис.), если  $\text{mSTR1} = \text{"Product of ИнCAT Company"}$ ,  $\text{mSTR2} = \text{"InSAT"}$ ,  $\text{in1} = 5$  и  $\text{in2} = 12$ , то  $\text{mSTR} = \text{"Product of InSAT Company"}$ .

FIND, поиск подстроки

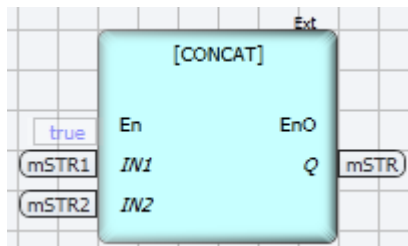


Функция возвращает номер символа, начиная с которого в строке IN1 располагается первое вхождение подстроки IN2:

$Q := \text{find}(\text{IN1}, \text{IN2})$

Например (см. рис.), если  $\text{mSTR1} = \text{"Product of InSAT Company"}$ ,  $\text{mSTR2} = \text{"InSAT"}$ , то  $\text{out} = 12$ .

CONCAT, конкатенация строк

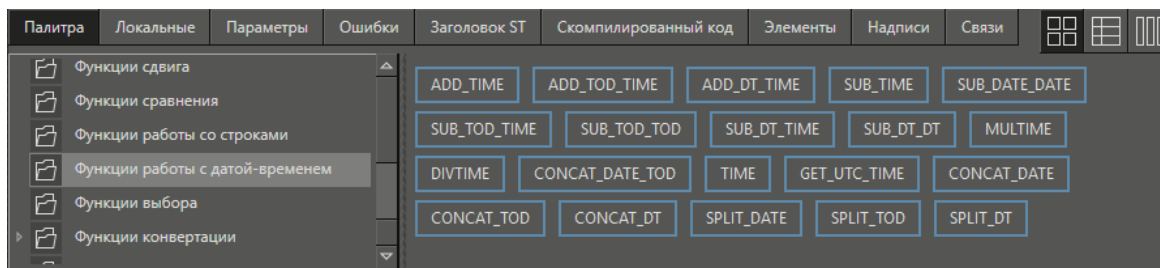


$Q := \text{concat}(\text{IN1}, \text{IN2})$

Например (см. рис.), если  $\text{mSTR1} = \text{"InSAT"}$ ,  $\text{mSTR2} = \text{" Company"}$ , то  $\text{mSTR} = \text{"InSAT Company"}$ .

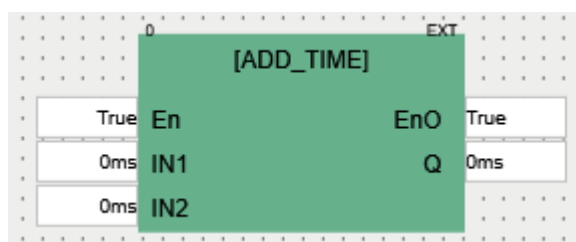
### 7.7.1.7. Функции работы с датой-временем

Вид категории в палитре языков программирования:



## ADD\_TIME

Сложение переменных типа TIME



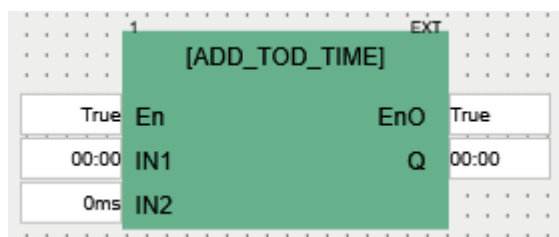
Входы/Выходы	Тип	Описание
IN1	TIME	Вход функции, 1 слагаемое.
IN2	TIME	Вход функции, 2 слагаемое
Q	TIME	Выход функции, сумма времен поданных на входы функции

Пример в ST

```
Q := add_time(IN1, IN2)
```

## ADD\_TOD\_TIME

Сложение переменных типа TOD и TIME



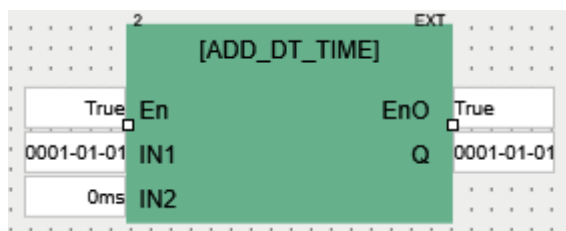
Входы/Выходы	Тип	Описание
IN1	TOD	Вход функции, 1 слагаемое.
IN2	TIME	Вход функции, 2 слагаемое
Q	TOD	Выход функции, сумма времен поданных на входы функции

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

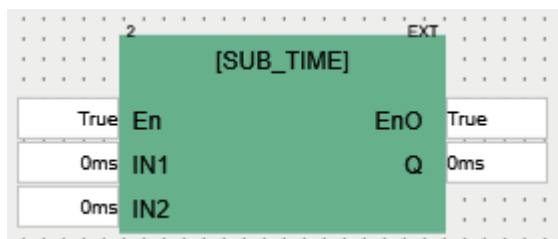
Пример в ST

```
Q := add_dt_time(IN1, IN2)
```

SUB\_TIME

Вычитание переменных типа TIME





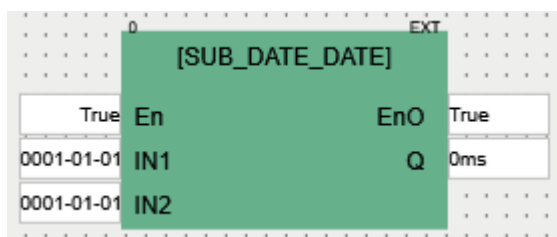
Входы/Выходы	Тип	Описание
IN1	TIME	Вход функции, уменьшаемое
IN2	TIME	Вход функции, вычитаемое
Q	TIME	Выход функции, разность времен поданных на входы функции

Пример в ST

```
Q := sub_time(IN1, IN2)
```

### SUB\_DATE\_DATE

Вычитание переменных типа DATE



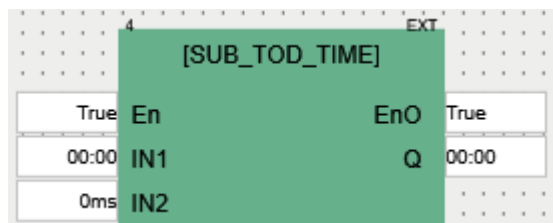
Входы/Выходы	Тип	Описание
IN1	DATE	Вход функции, уменьшаемое
IN2	DATE	Вход функции, вычитаемое
Q	TIME	Выход функции, разность времен поданных на входы функции

Пример в ST

Q := sub\_date\_date(IN1, IN2)

### SUB\_TOD\_TIME

Вычитание переменных типа TOD и TIME



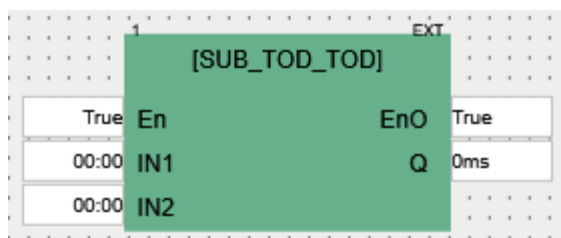
Входы/Выходы	Тип	Описание
IN1	TOD	Вход функции, уменьшаемое
IN2	TIME	Вход функции, вычитаемое
Q	TOD	Выход функции, разность времен поданных на входы функции

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

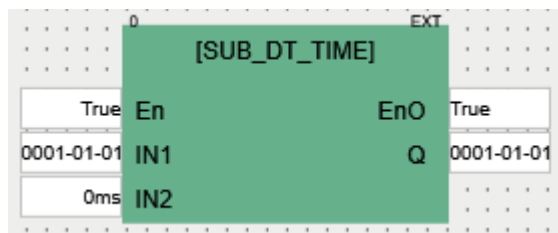
Q	TIME	Выход функции, разность времен поданных на входы функции
---	------	--

Пример в ST

```
Q := sub_tod_tod(IN1, IN2)
```

### SUB\_DT\_TIME

Вычитание переменных типа DT и TIME



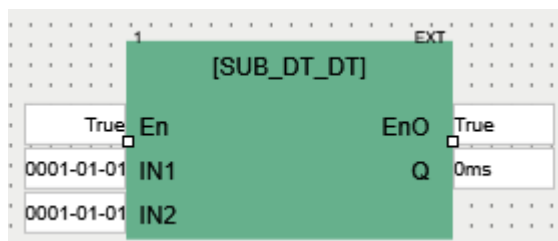
Входы/Выходы	Тип	Описание
IN1	DT	Вход функции, уменьшаемое
IN2	TIME	Вход функции, вычитаемое
Q	DT	Выход функции, разность времен поданных на входы функции

Пример в ST

```
Q := sub_dt_time(IN1, IN2)
```

### SUB\_DT\_DT

Вычитание переменных типа DT



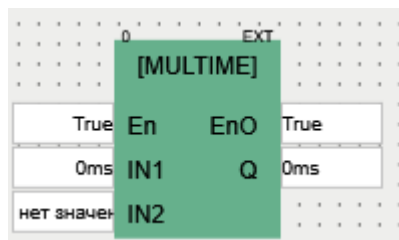
Входы/Выходы	Тип	Описание
IN1	DT	Вход функции, уменьшаемое
IN2	DT	Вход функции, вычитаемое
Q	TIME	Выход функции, разность времен поданных на входы функции

Пример в ST

```
Q := sub_dt_dt(IN1, IN2)
```

### MULTIME

Умножение времени на число



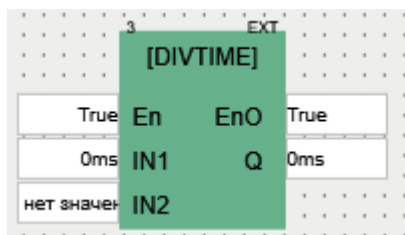
Входы/Выходы	Тип	Описание
IN1	TIME	Вход функции, множимое
IN2	ANY_NUM	Вход функции, множитель
Q	TIME	Выход функции, произведение

Пример в ST

```
Q := multime(IN1, IN2)
```

### DIVTIME

Деление времени на число



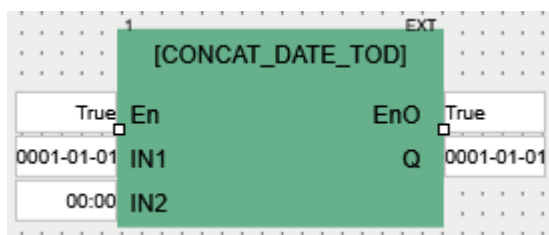
Входы/Выходы	Тип	Описание
IN1	TIME	Вход функции, делимое
IN2	ANY_NUM	Вход функции, делитель
Q	TIME	Выход функции, частное

Пример в ST

```
Q := divtime(IN1, IN2)
```

CONCAT\_DATE\_TOD

Добавление времени к дате



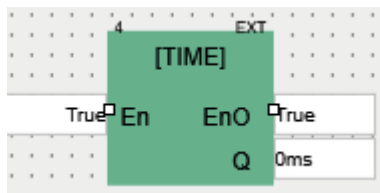
Входы/Выходы	Тип	Описание
IN1	DATE	Вход функции, дата
IN2	TIME	Вход функции, время
Q	DT	Выход функции, результат

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  $(2^{32} - 1) \text{ мс} = \text{t\#49d17h02m47s295ms}$ , 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  $\text{deltaT}$ :

```
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;
```

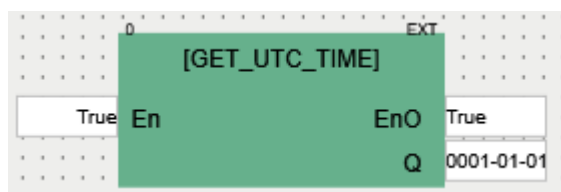
Нестандартный таймер

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 time() в тип данных BYTE.

## GET\_UTC\_TIME

Выдает дату и время UTC заданную в устройстве



Тип данных Q должен быть DATE\_AND\_TIME.

Пример в ST

```
Q := get_utc_time()
```

Смотрите также:

CONCAT\_DATE

CONCAT\_TOD

CONCAT\_DT

SPLIT\_DATE

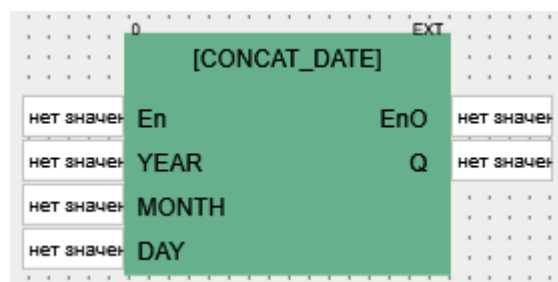
SPLIT\_TOD

SPLIT\_DT

### 7.7.1.7.1. CONCAT\_DATE

Соединение отдельных значений в формат DATE. Функция находится в категории Функции работы с датой-временем.

Вид функции в редакторе FBD:

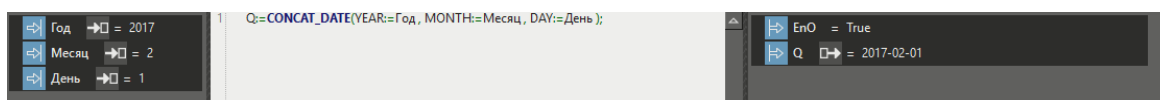


Входы/Выходы	Тип	Описание
--------------	-----	----------

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

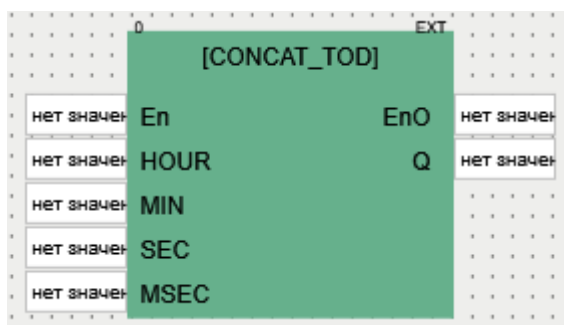
Пример в ST



### 7.7.1.7.2. 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:

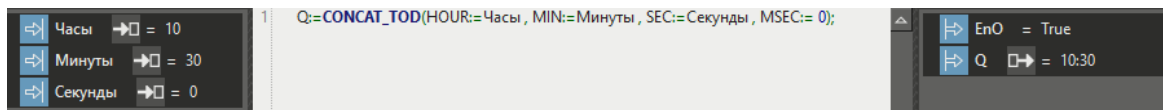


Входы/Выходы	Тип	Описание
HOUR	USINT	Вход функции, отвечающий за часы
MIN	USINT	Вход функции, отвечающий за минуты
SEC	USINT	Вход функции, отвечающий за секунды



MSEC	USINT	Вход функции, отвечающий за миллисекунды
Q	TOD	Выход функции, результат

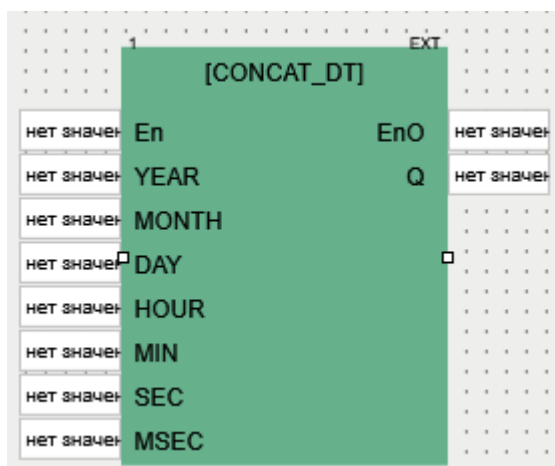
Пример в ST



### 7.7.1.7.3. CONCAT\_DT

Соединение отдельных значений формат DT. Функция находится в категории Функции работы с датой-временем.

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

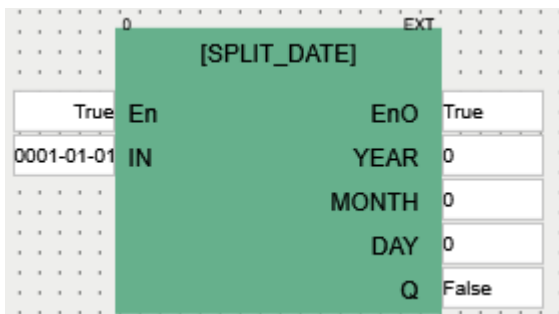
Пример в ST

```
Q:=CONCAT_DT(YEAR:=Год , MONTH:=Месяц , DAY:=День , HOUR:=Часы ,
MIN:=Минуты , SEC:=Секунды , MSEC:= 0);
```

#### 7.7.1.7.4. SPLIT\_DATE

Разделение значения DATE на отдельные составляющие. Функция находится в категории Функции работы с датой-временем.

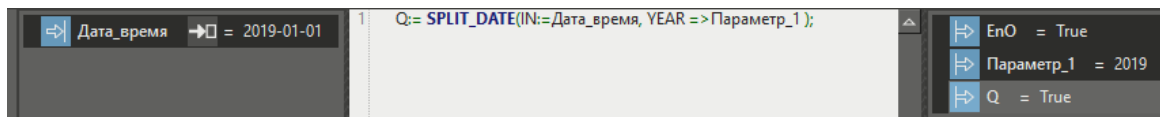
Вид функции в редакторе FBD:



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
---	------	-------------------------

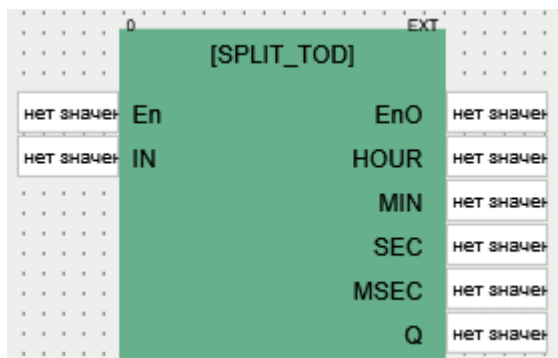
Пример в ST



### 7.7.1.7.5. SPLIT\_TOD

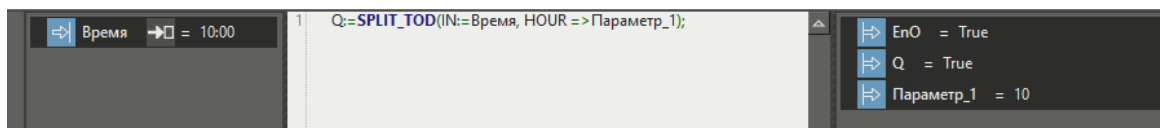
Разделение значения TOD на отдельные составляющие. Функция находится в категории Функции работы с датой-временем.

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

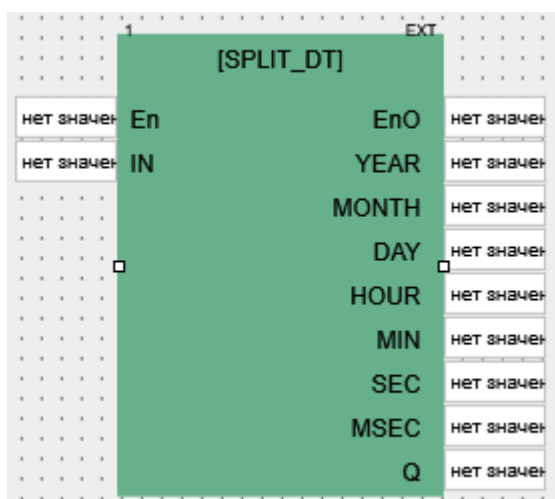
Пример в ST



### 7.7.1.7.6. SPLIT\_DT

Разделение значения DT 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.
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

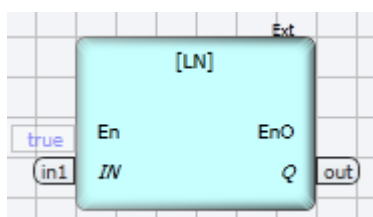
Пример в ST

```
Q:=SPLIT_DT(IN:=Время, HOUR =>Параметр_1, MONTH=>Параметр_2);
```

### 7.7.1.8. Логарифмические функции

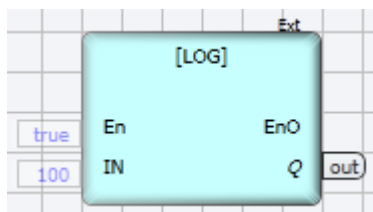
- LN, натуральный логарифм
- LOG, десятичный логарифм
- EXP, экспонента

LN, натуральный логарифм



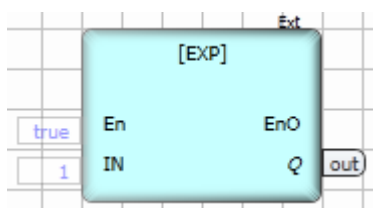
$Q := \ln(IN)$

LOG, десятичный логарифм



$Q := \log(IN)$

EXP, экспонента

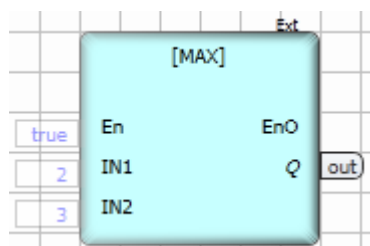


$Q := e ** IN$

### 7.7.1.9. Функции выбора

- MAX, максимальное из двух
- MIN, минимальное из двух
- SEL, выбор из двух
- MUX, мультиплексор
- LIMIT, ограничение

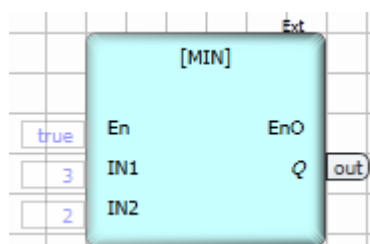
MAX, максимальное из двух



Функция возвращает наибольшее из двух значений:

$Q := \max(\text{IN1}, \text{IN2})$

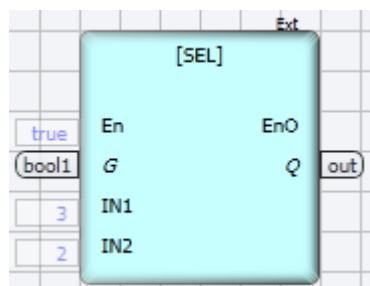
MIN, минимальное из двух



Функция возвращает наименьшее из двух значений:

$Q := \min(\text{IN1}, \text{IN2})$

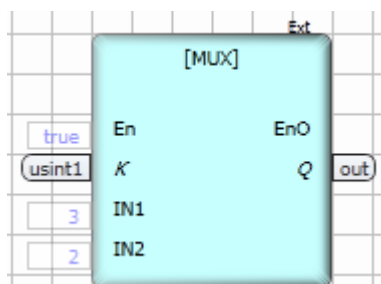
SEL, выбор из двух



$Q := \text{sel}(G, \text{IN1}, \text{IN2})$

Если  $G=\text{FALSE}$ ,  $Q=\text{IN1}$ , если  $G=\text{TRUE}$ ,  $Q=\text{IN2}$

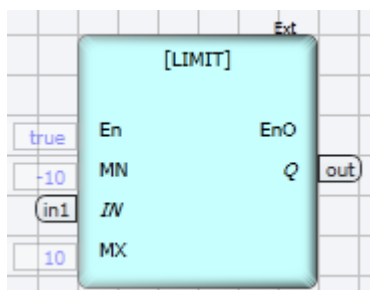
MUX, мультиплексор



$Q := \text{mux}(K, \text{IN1}, \text{IN2})$

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  $\text{MN} \leq \text{IN} \leq \text{MX}$ ,  $Q := \text{IN}$ .

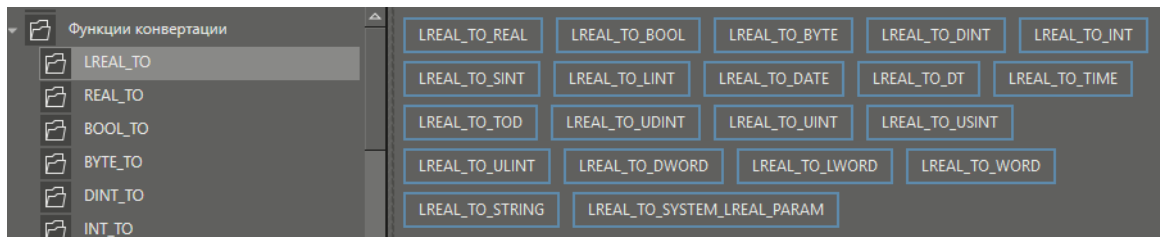
Если  $\text{IN} < \text{MN}$ ,  $Q := \text{MN}$ .

Если  $\text{IN} > \text{MX}$ ,  $Q := \text{MX}$ .

### 7.7.1.10. Функции конвертации

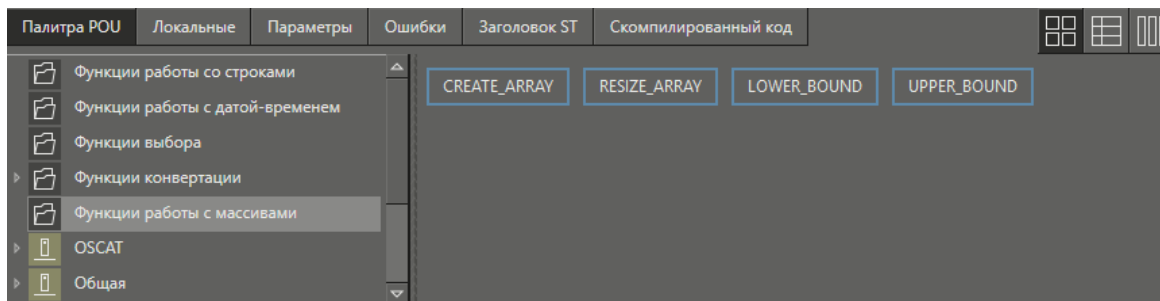
Данная категория содержит внутри себя различные группы функций предназначенных для конвертации из одних типов данных в другие.

Группа LREAL\_TO:



### 7.7.1.11. Функции работы с массивами

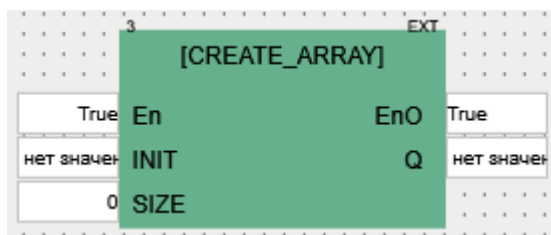
Вид категории в палитре языков программирования:



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



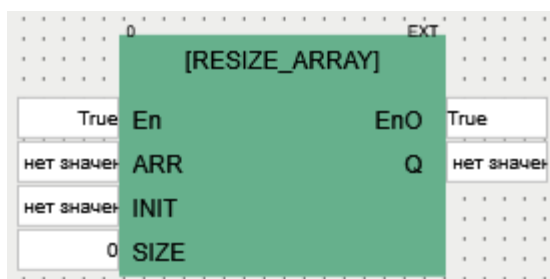
## Пример в ST



Выход программы Параметр\_1 имеет тип Массив с динамическим размером осей

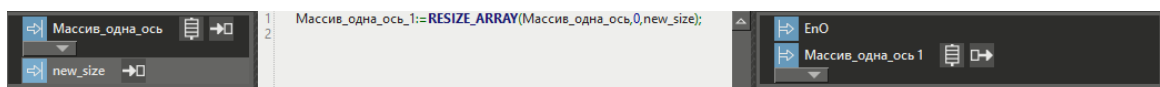
## Функция RESIZE\_ARRAY

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.
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

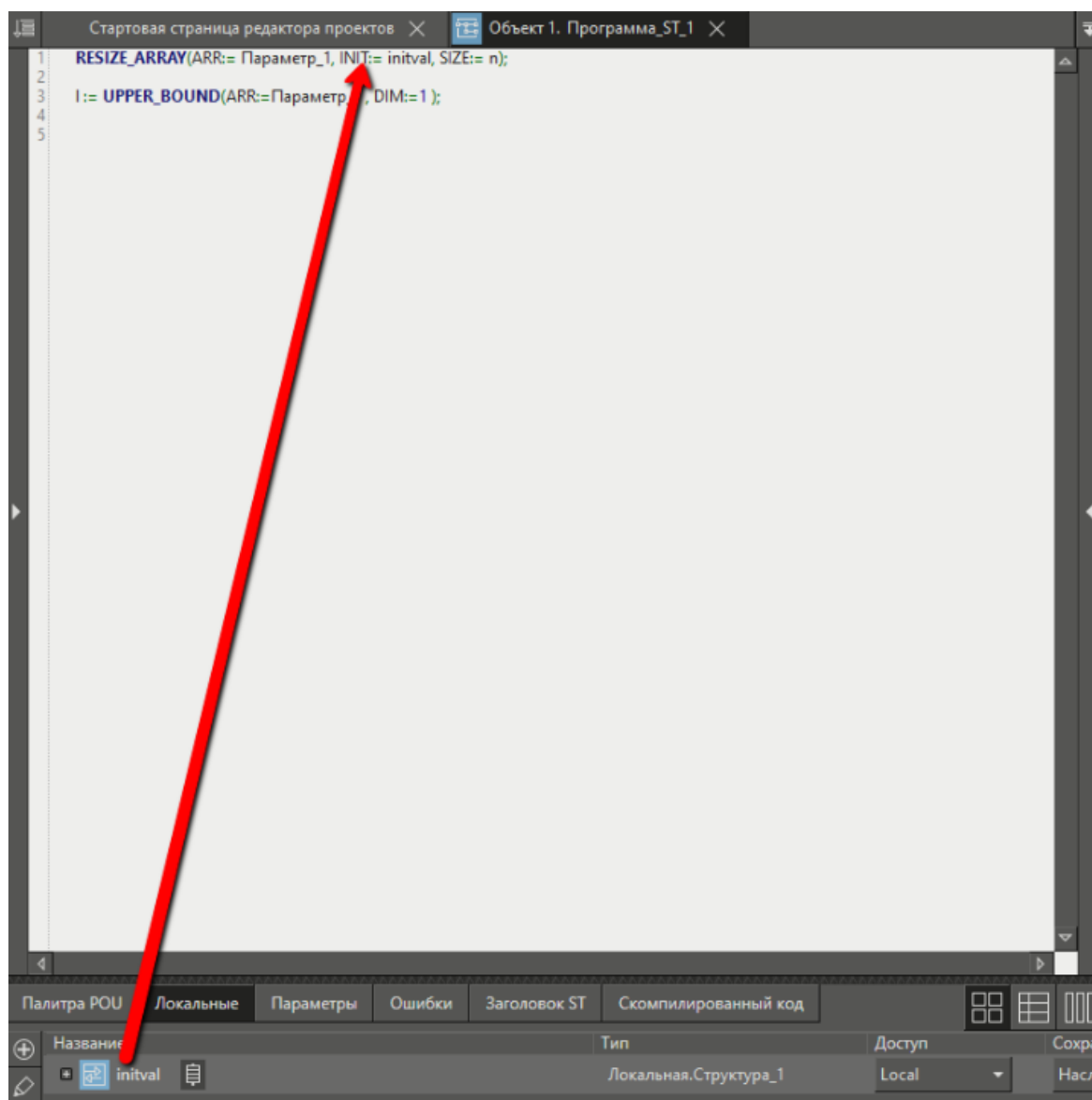
## Пример в ST для массива чисел



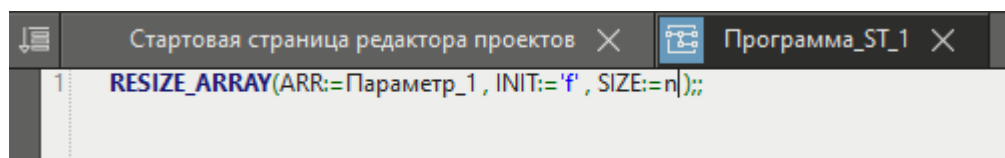
Вход и выход программы связаны с параметром, который имеет тип Массив, размер которого нужно изменить. Новые значения массива будут равны 0, а размер массива будет зависеть от значения входа программы new\_size.

## Пример в ST для массива структур

Сперва нужно на вкладке Локальные создать переменную, с типом данных Структура. Затем эту переменную нужно перетащить в программу:



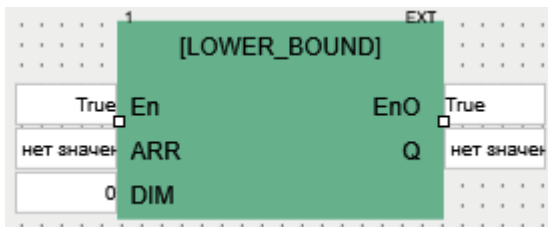
Пример в ST для строк



Порядок работы такой-же как и для числовых одномерных массивов.

Функция LOWER\_BOUND

Функция получает нижнюю границу массива (минимальный индекс).



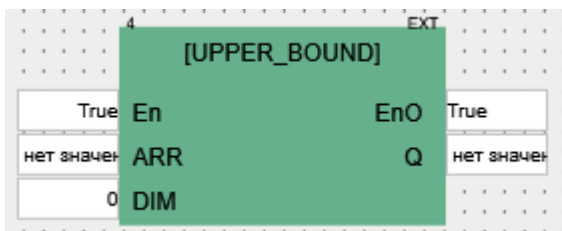
Входы/Выходы	Тип	Описание
ARR	ANY	Вход функции, который необходимо связать с массивом, границу которого нужно проверить.
DIM	ANY_INT	Вход функции, где задается номер измерения (оси), индекс которого необходимо получить. В случае работы с двумерным массивом если установить 1, то получится минимальный 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

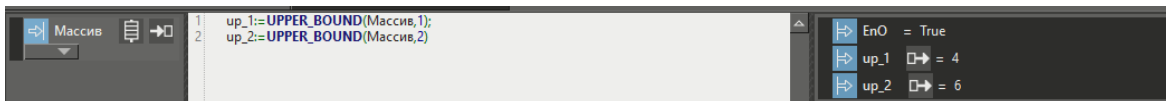
Функция получает верхнюю границу массива (максимальный индекс).



Входы/Выходы	Тип	Описание
ARR	ANY	Вход функции, который необходимо связать с массивом, границу которого нужно проверить.

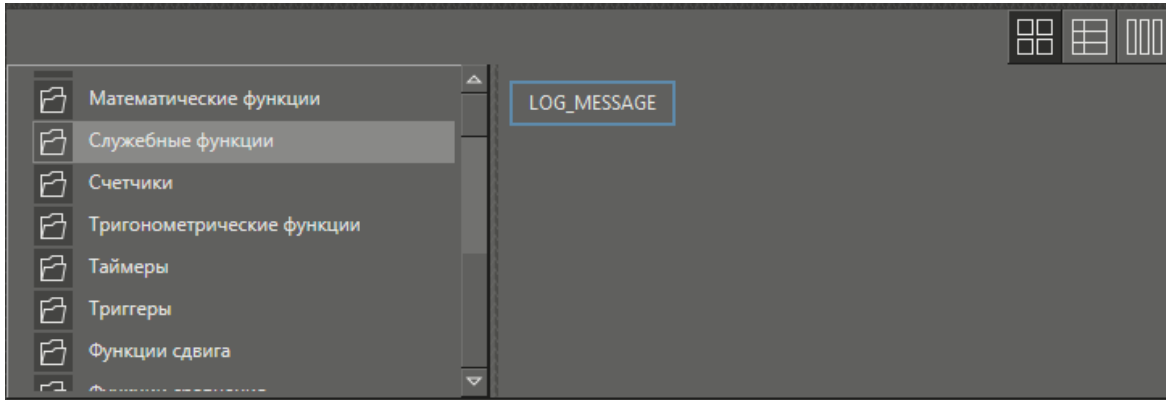
DIM	ANY_INT	Вход функции, где задается номер измерения (оси), индекс которого необходимо получить. В случае работы с двумерным массивом если установить 1, то получится максимальный индекс по вертикали, а если 2, то по горизонтали.
Q	ANY	Выдается верхний индекс массива.

Пример в ST



### 7.7.1.12. Служебные функции

Вид категории в палитре языков программирования:



Функция LOG\_MESSAGE

Функция используется для формирования сообщений, которые сохраняются в лог исполнительной системы



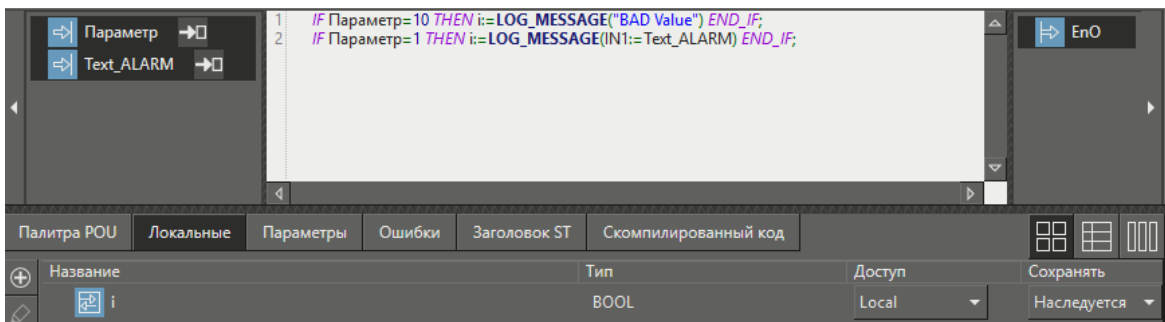
Специальные входы и выходы.

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.

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

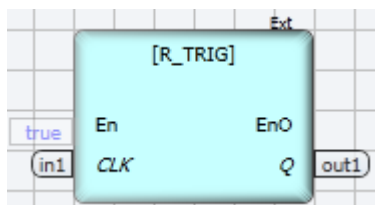
Локальная переменная i принимает значение выхода функции Q, после того как сообщение будет сформировано.

## 7.7.2. Библиотека Стандартные ФБ

### 7.7.2.1. ФБ 'Детекторы фронтов'

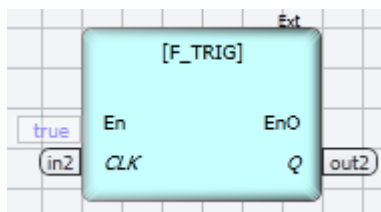
- R\_TRIG, импульс по переднему фронту
- F\_TRIG, импульс по заднему фронту

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

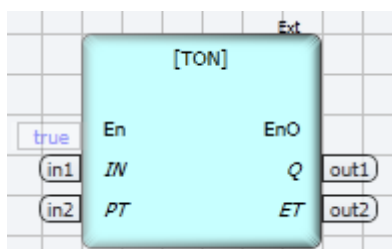


$Q=1$  в течение 1 такта при изменении значения CLK с любого ненулевого на 0.

### 7.7.2.2. ФБ 'Таймеры'

- TON, задержка включения
- TOF, задержка выключения
- TP, импульс произвольной длительности

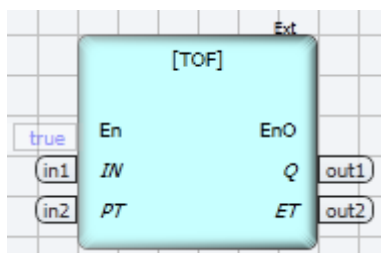
TON, задержка включения



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.

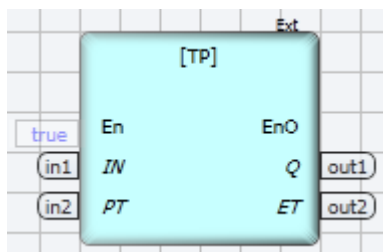
TOF, задержка выключения



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

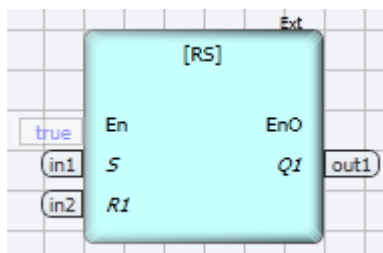


При изменении значения IN с 0 на любое ненулевое  $Q=1$  на время PT. ET индицирует время, прошедшее с начала формирования импульса.

### 7.7.2.3. ФБ 'Триггеры'

- RS, триггер с приоритетом по сбросу
- SR, триггер с приоритетом по установке

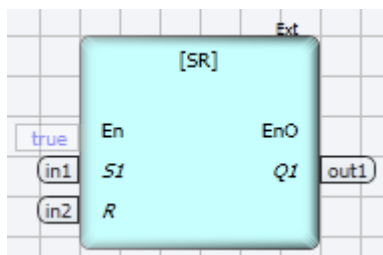
RS, триггер с приоритетом по сбросу



Если  $R1 \neq 0$ ,  $Q1=0$ .

$Q1=1$ , если  $S \neq 0$  и  $R1=0$ .

SR, триггер с приоритетом по установке



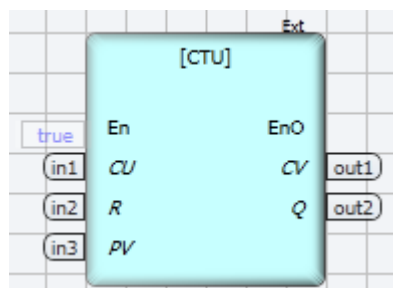
Если  $S1=0$  и  $R \neq 0$ ,  $Q1=0$ .

$Q1=1$ , если  $S1 \neq 0$ .

### 7.7.2.4. ФБ 'Счетчики'

- СТУ, счетчик прямой
- СТД, счетчик обратный
- СТУД, счетчик комбинированный

СТУ, счетчик прямой



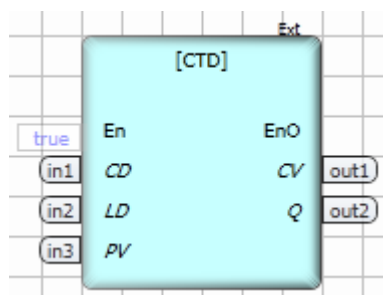
Входы:

- CU (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.

СТД, 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);

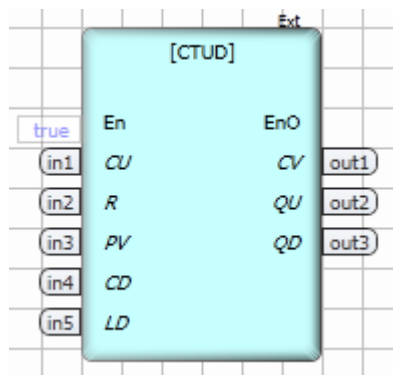


- PV (INT) – верхний предел, т.е. начальное значение декремента.

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.

## 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:

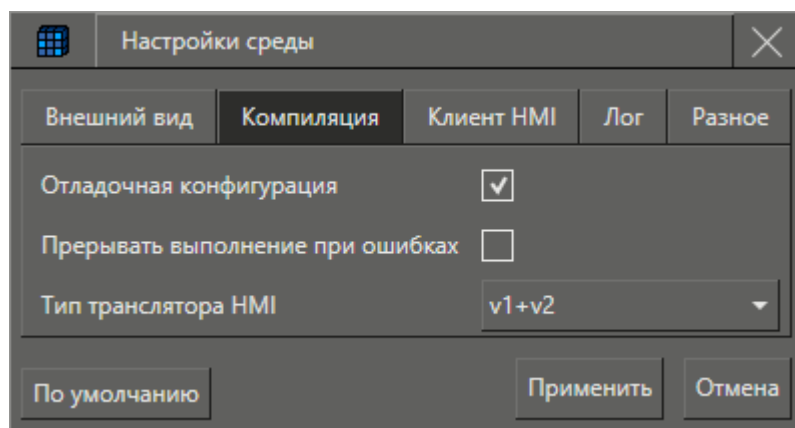
- В программе должны использоваться все возможные новые мировые тренды и технологии;
- Должен обеспечиваться максимально возможный функционал программы даже в самой узкоспециализированной ОС.

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.

Отличие версий описано в разделе Создание окон для графического интерфейса операторов.

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 крестику 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 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, всплыва- ющая подсказка в Тренде	Стандартная встроенная.	Собственная (html5)
Методы панели инстру- ментов в Тренде	Доступны только в па- нели инструментов	Можно вызывать отдельно
Стрелочный прибор без образа	Не работает	Работает
Все всплывающие окна приведены к одному типу	Сообщение о потере связи с сервером, сооб- щение из действия По- казать сообщение и всплывающие окна имеют разный вид	Есть
Модальность и непо- движность всплываю- щих окон	Нет	Есть
Восстановление связи с исполнительной систе- мой	Только с перезагрузкой страницы	Без перезагрузки
Права доступа	Принадлежит контролю	Принадлежит связи или действию
Автоматический логин на всех экземплярах	Нет	Есть

встроенного браузера по первому входу		
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

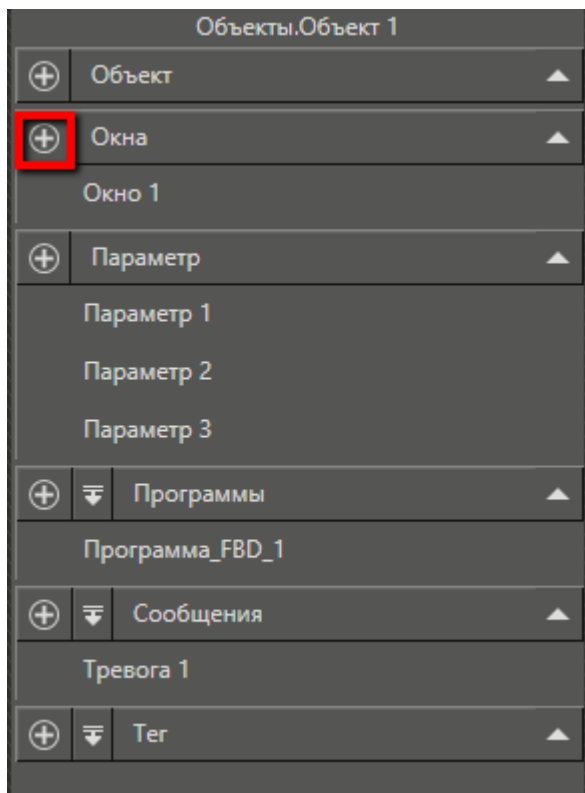
Окно – это элемент проекта. Оно может использоваться как самостоятельно, так и в качестве составной части другого окна.

Окна создаются в группах Ресурсы. Окна следующих элементов:

- объект;
- канал;
- тег.


Кроме того, они могут быть созданы в группе Окна у узлов.

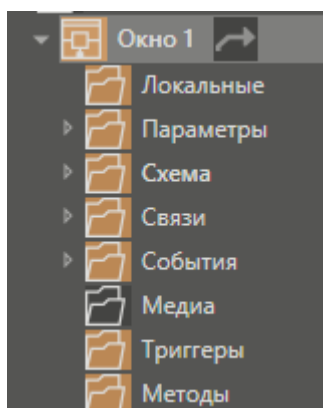
Уникальные окна, т.е. встречающиеся в проекте один раз, добавляются в контекстном меню родительского элемента при помощи команды Добавить.Окно или через контекстную панель:



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.

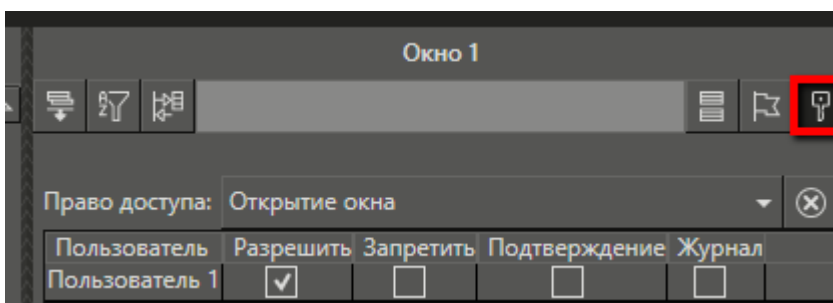
Если при разработке проекта используется упрощенное дерево, то окно представляется в виде простого элемента:  Окно 1. А в случае, если используется полное дерево, окно имеет вид:



Основная работа с окном осуществляется в редакторе HMI, 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 HMI.

После добавления окна в проект, редактор HMI 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:



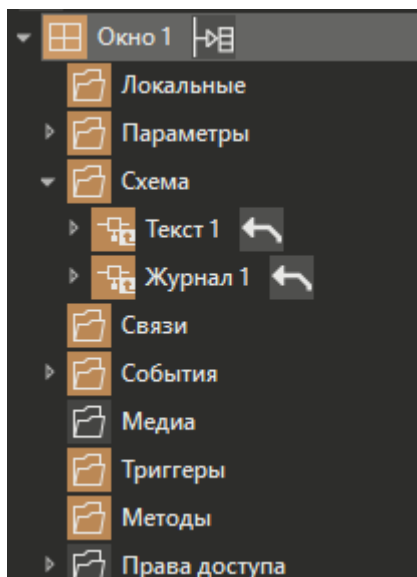
В этом случае появится дополнительная группа Права доступа в структуре окна в дереве.

### 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:

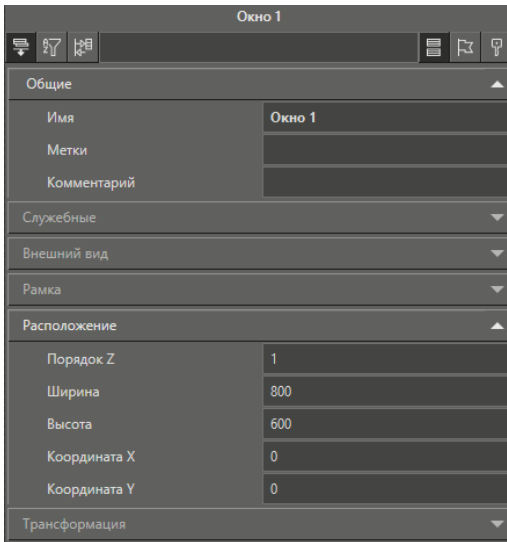
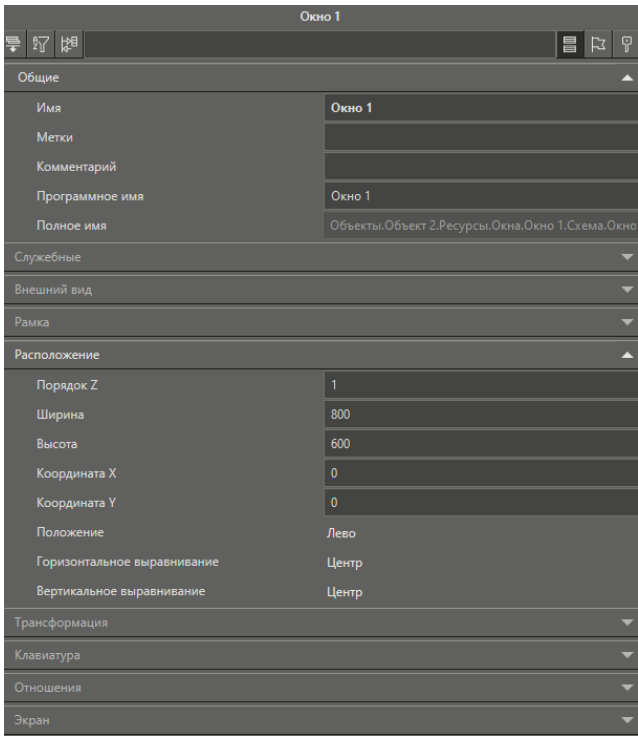
Название	Назначение
Локальные	Группа содержит локальные параметры окна, используемые для обеспечения внутренней логики работы окна. Соответствует вкладке легенды Локальные.
Параметры	Содержит параметры окна. Соответствует вкладке легенды Параметры.
Схема	Содержит элементы, которые находятся в окне. Каждый 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
	

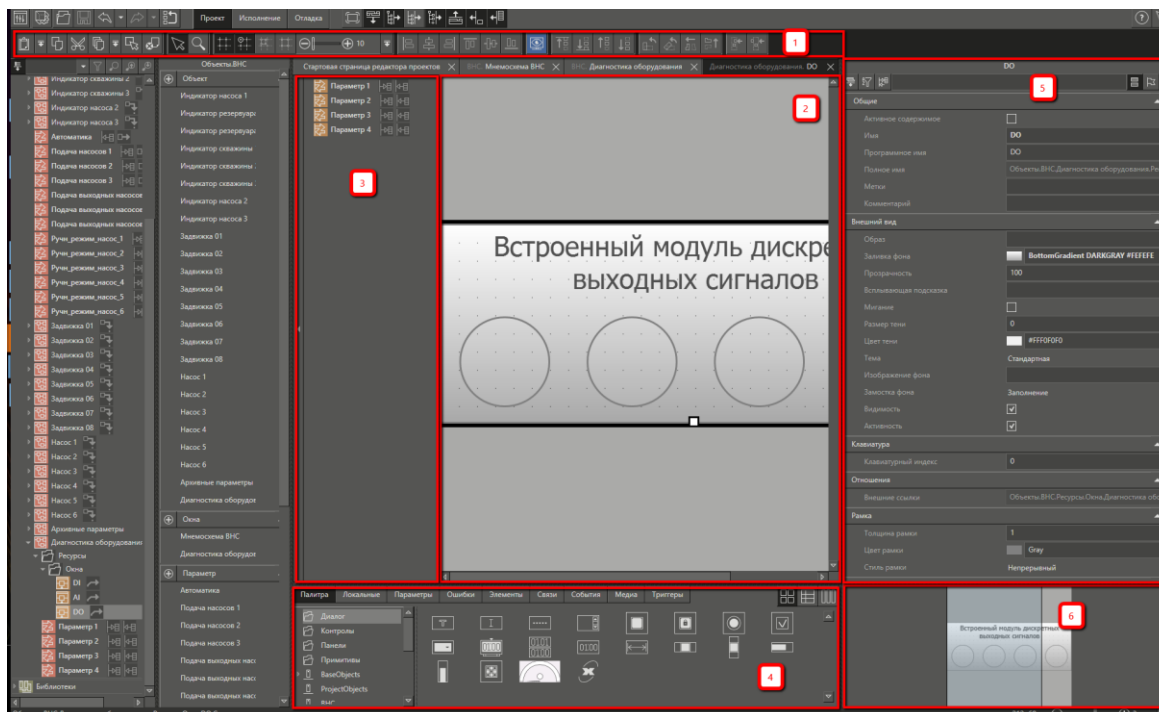
Название	Рекомендации
Категория Общие	Как правило, в этой категории настраивается только свойство Имя. Подробное описание данной категории приведено в соответствующем разделе.
Категория Внешний вид	В этой категории чаще всего настраивается свойство Заливка фона. Для увеличения скорости загрузки окон реко-

	мендуется 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 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
	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

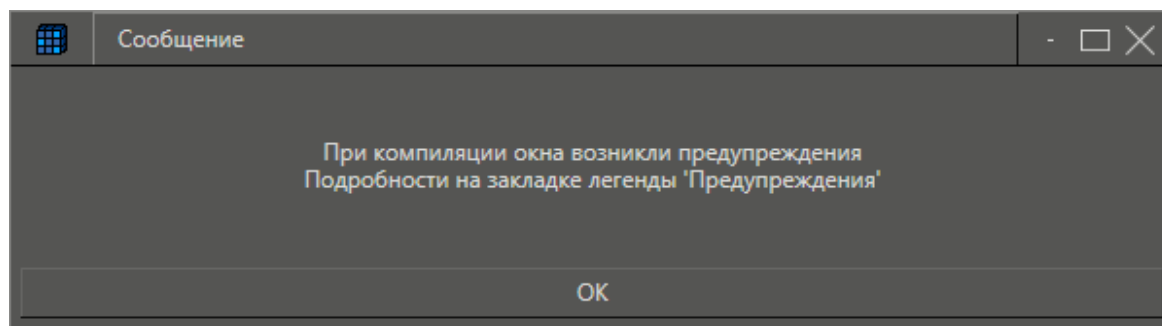
При нажатии на кнопку Просмотр панели инструментов редактора НМІ появляется возможность без запуска среды исполнения предварительно оценить внешний вид, который окно (мнемосхема) будет иметь в клиенте визуализации.



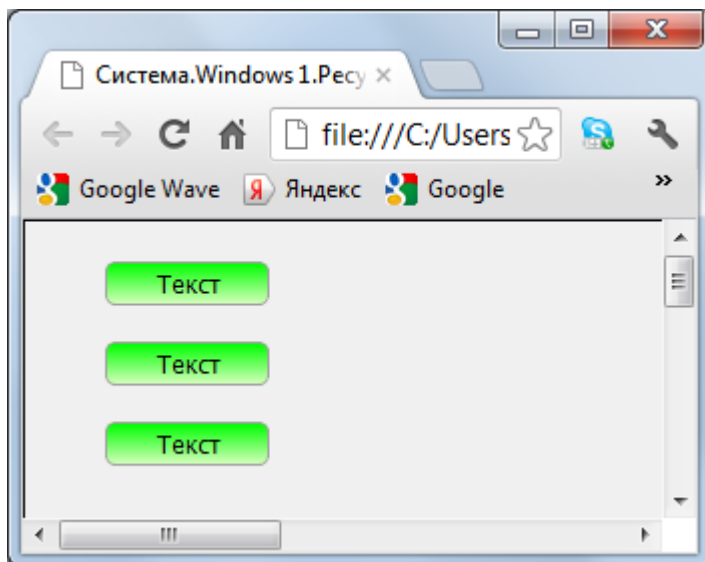
По команде  выполняются следующие действия:

- 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 вверх	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.

	Переместить на шаг вниз	Перемещает выделенные элементы на шаг вниз по оси Z.
--	-------------------------	--

### 8.2.1.4. Панель Поворот

В редакторе HMI панель имеет вид:

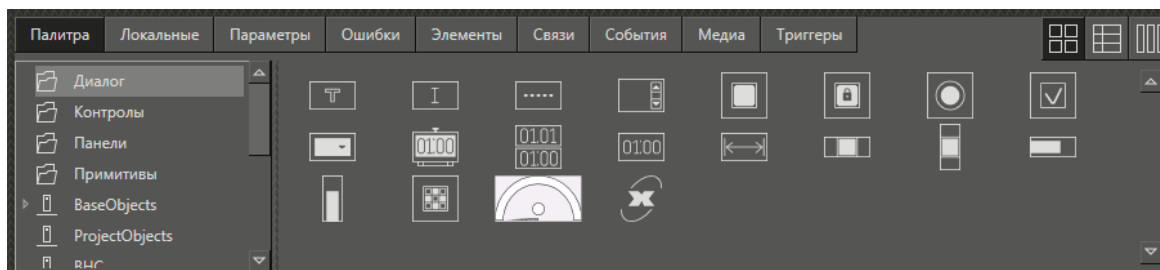


Панель содержит следующие кнопки:

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.

### 8.2.2. Легенда в редакторе HMI

Если в окне документов и редакторов активен редактор HMI, то легенда имеет вид:



Легенда содержит следующие вкладки:

- Легенда HMI. Вкладка Палитра;
- Легенда HMI. Вкладка Локальные;
- Легенда HMI. Вкладка Параметры;

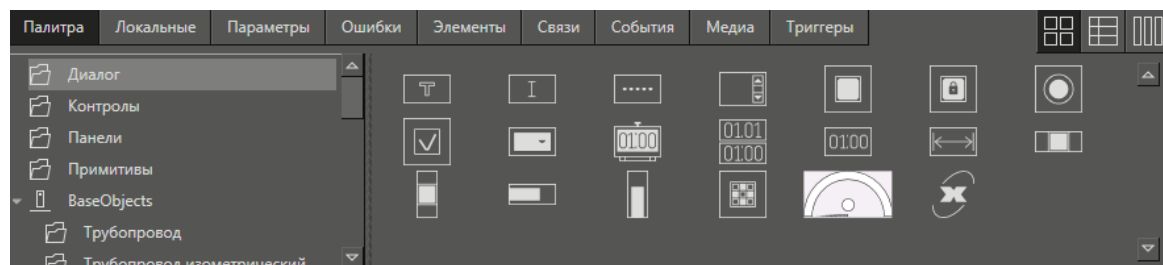
- Легенда HMI. Вкладка Ошибки
- Легенда HMI. Вкладка Элементы;
- Легенда HMI. Вкладка Связи;
- Легенда HMI. Вкладка События;
- Легенда HMI. Вкладка Медиа;
- Легенда HMI. Вкладка Триггеры;

Видимостью легенды управляет переключатель Легенда (см. Панель управления видимостью окон ).

### 8.2.2.1. Легенда HMI. Вкладка Палитра

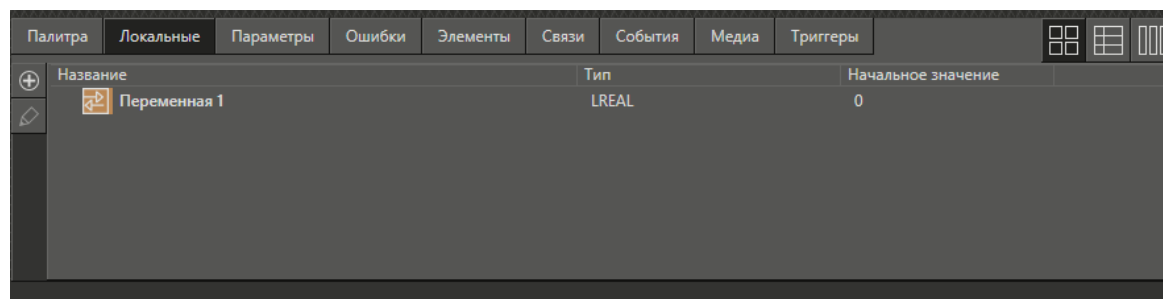
На этой вкладке отображаются встроенные графические элементы, находящиеся в библиотеке (например, HMI , BaseObjects, Images и др.), и которые можно разместить в рабочей области редактора HMI.

В левой части вкладки отображаются категории графических элементов, а в правой - элементы, которые могут быть перетащены в окно.



### 8.2.2.2. Легенда HMI. Вкладка Локальные

На этой вкладке отображается редактор локальных переменных окна:





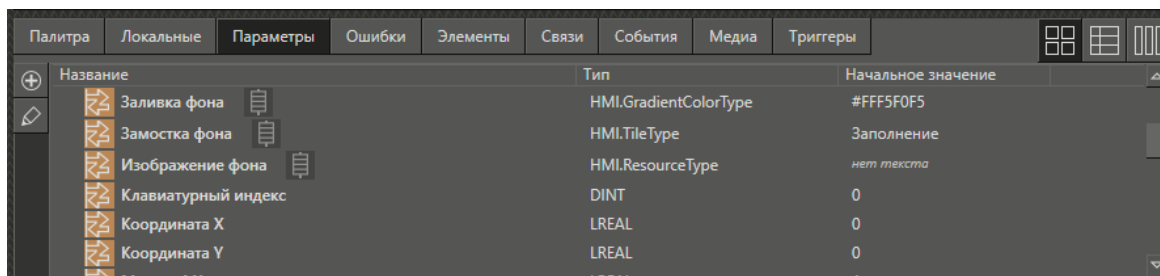
Редактор выполнен так же, как редактор локальных переменных программы (см. Легенда. Вкладка Локальные), но содержит только столбцы Название, Тип и Начальное значение.

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.

### 8.2.2.3. Легенда HMI. Вкладка Параметры

На этой вкладке отображается редактор Параметров окна:



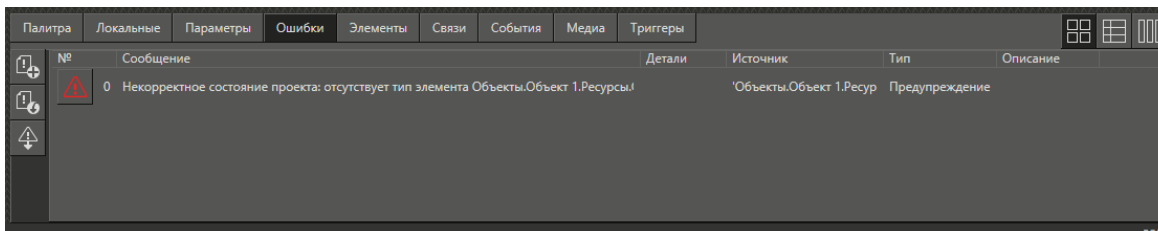
Т.е. в данной вкладке перечислены все стандартные параметры окна, значения которых могут быть изменены в режиме исполнения, а также параметры, созданные пользователем в окне клеммников

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.

### 8.2.2.4. Легенда HMI. Вкладка Ошибки

На этой вкладке отображаются Ошибки проекта. Вкладка содержит Ошибки только текущей мнемосхемы (см. Предварительный просмотр мнемосхемы и Команда проверки проекта):

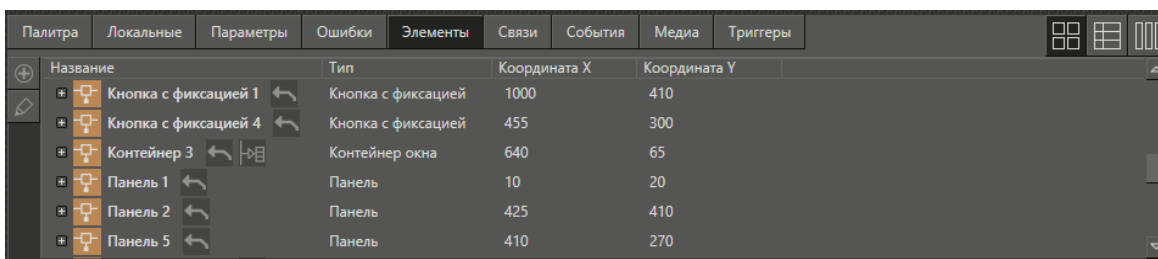


В отличие от таблицы, в окне ошибок строки на вкладке ошибок не имеют контекстного меню.

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.

### 8.2.2.5. Легенда НМІ. Вкладка Элементы

На этой вкладке отображается список графических элементов окна.

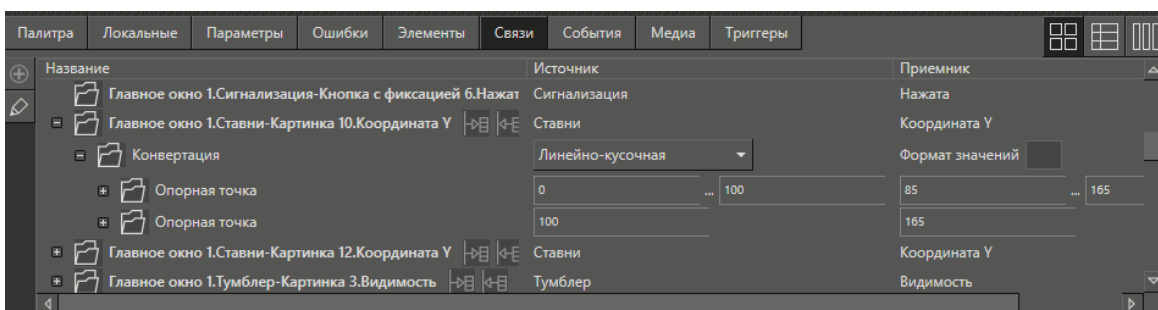


Элементы в списке можно отсортировать, нажав левой кнопкой мыши на заголовок какого-либо столбца.

Элемент, выделенный в панели, выделяется в рабочей области редактора НМІ, а в панели свойств отображается свойство выделенного элемента.

### 8.2.2.6. Легенда НМІ. Вкладка Связи

На этой вкладке отображаются все Связи элементов окна, используемые для динамизации. Кроме того, на ней можно отредактировать существующие Связи.

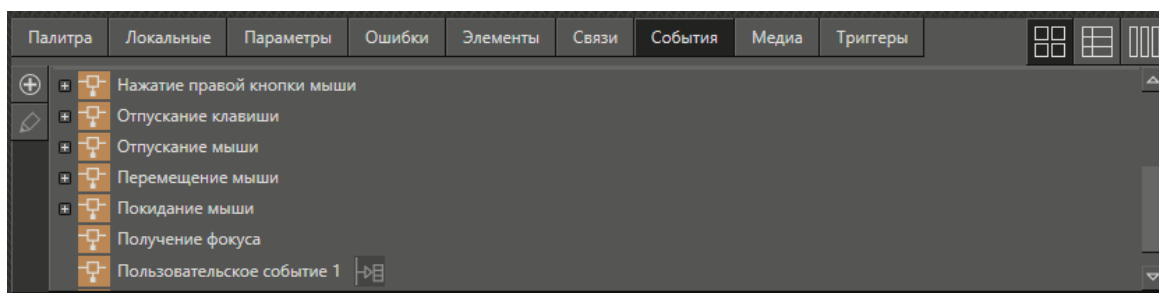


Связь редактируется таким же образом как и в панели свойств: сначала определяется тип конвертации, а затем указываются опорные точки, для которых задается зависимость между значением источника и приемника.

Если дважды кликнуть левой кнопкой мыши по названию какого-либо свойства или названию его дочернего элемента Конвертация, то откроется соответствующий этому элементу Конвертер значений.

### 8.2.2.7. Легенда HMI. Вкладка События

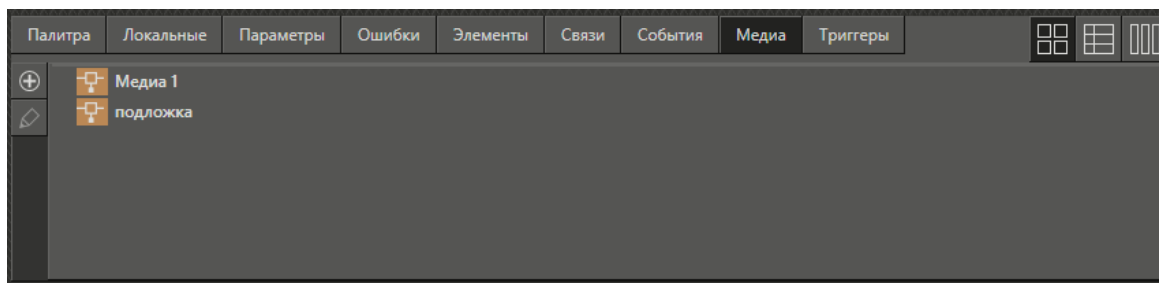
На этой вкладке отображается редактор группы событий окна:



На данной вкладке перечислены стандартные события окна. Кроме того, на ней имеется возможность создать пользовательские события. Пользовательские события окон необходимы при создании библиотечных окон.

### 8.2.2.8. Легенда HMI. Вкладка Медиа

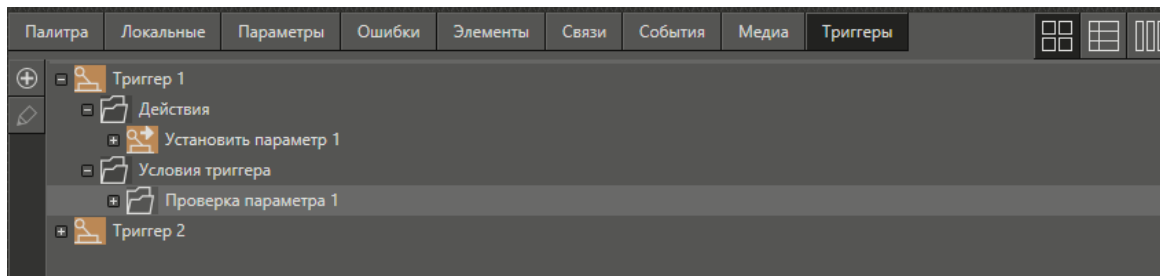
На этой вкладке отображаются медиа, которые уже используются в окне, например, в элементе Картинка. Кроме того, вкладка дает возможность добавлять медиа-файлы, которые еще только планируются задействовать:



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.

### 8.2.2.9. Легенда HMI. Вкладка Триггеры

На этой вкладке отображается редактор группы триггеров окна



### 8.2.3.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.

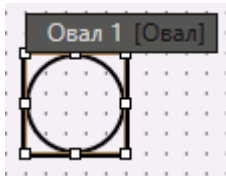
Список элементов дерева, имеющих графическое отображение:

- 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.

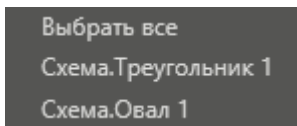
Если элемент выделен в окне, то отображается ограничивающий его прямоугольник (толщина линии – 1 px):



Кроме того, 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:

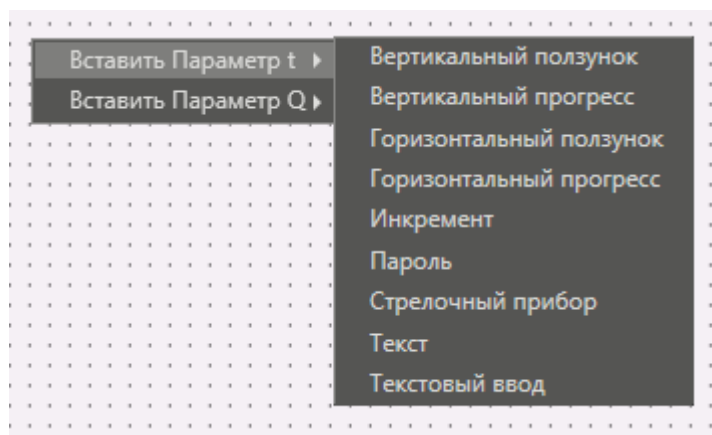


### 8.2.3.1. Размещение Тега, Канала, Объекта в окне

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 $\leftrightarrow$ 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

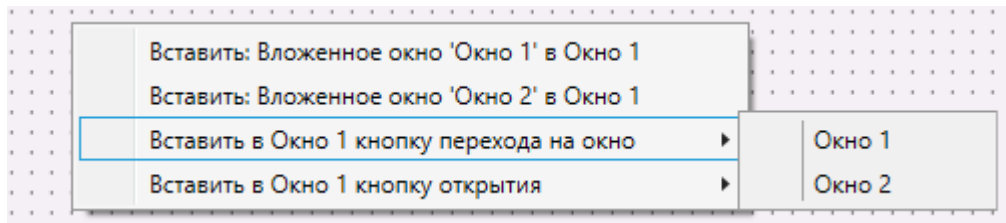
При перетаскивании левой кнопкой мыши в окно тега/канала/объекта, не содержащего собственных окон, для первого параметра такого тега/канала/объекта создается соответствующий графический элемент, в зависимости от типа данных параметра (см. Перетаскивание параметра в окно левой кнопкой мыши).

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 тегах/каналах/объектах 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 .

### 8.2.3.2. 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.

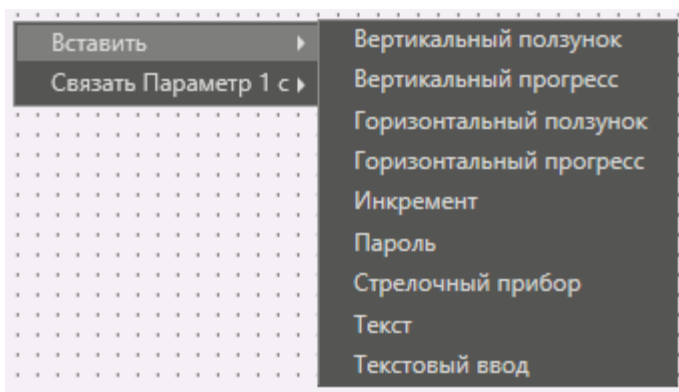
### 8.2.3.2.1. Перетаскивание параметра в окно правой кнопкой мыши

При перетаскивании параметра в окно правой кнопкой мыши открываются меню Вставить и Связать:

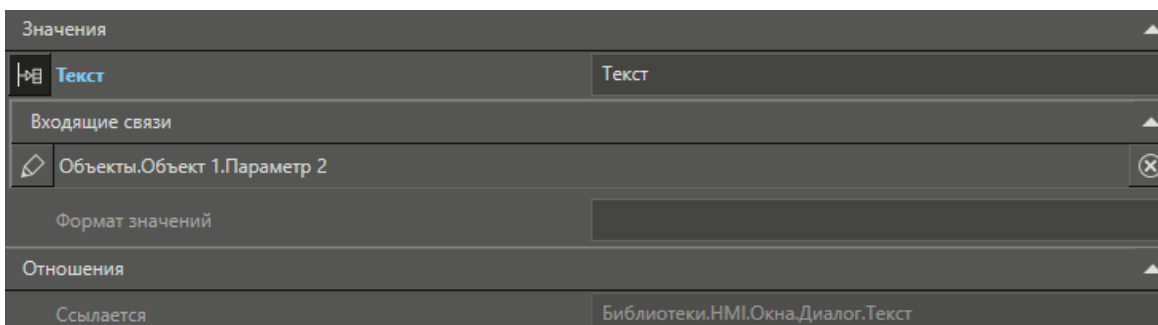
Меню Вставить

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.

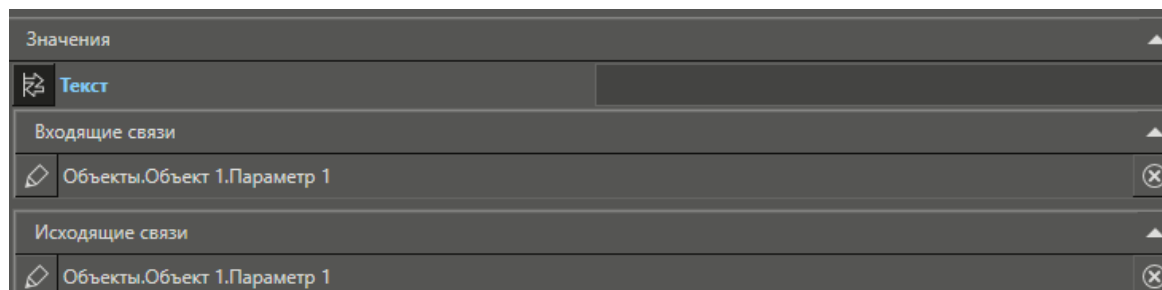
Набор элементов, доступных для вставки, зависит от типа данных перетаскиваемого параметра.



Если добавляется элемент, который допускает только получение данных в окне, то образуется входящая связь между свойством и параметром дерева. Например, для элемента Текст:



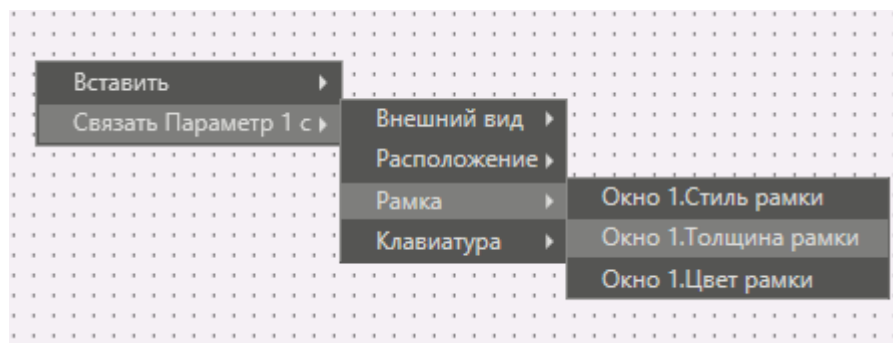
Для элементов типа Кнопка, Текстовый ввод, которые могут не только отображать полученные значения из дерева, но и изменять их, автоматически создаются две связи: входящая и исходящая:



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.

### 8.2.3.2.1.1. 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.

## 8.2.4.НМІ. Программирование в окнах

Для того, чтобы программировать в редакторе НМІ, 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

### 8.2.4.1. 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.

### 8.2.4.1.1. 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

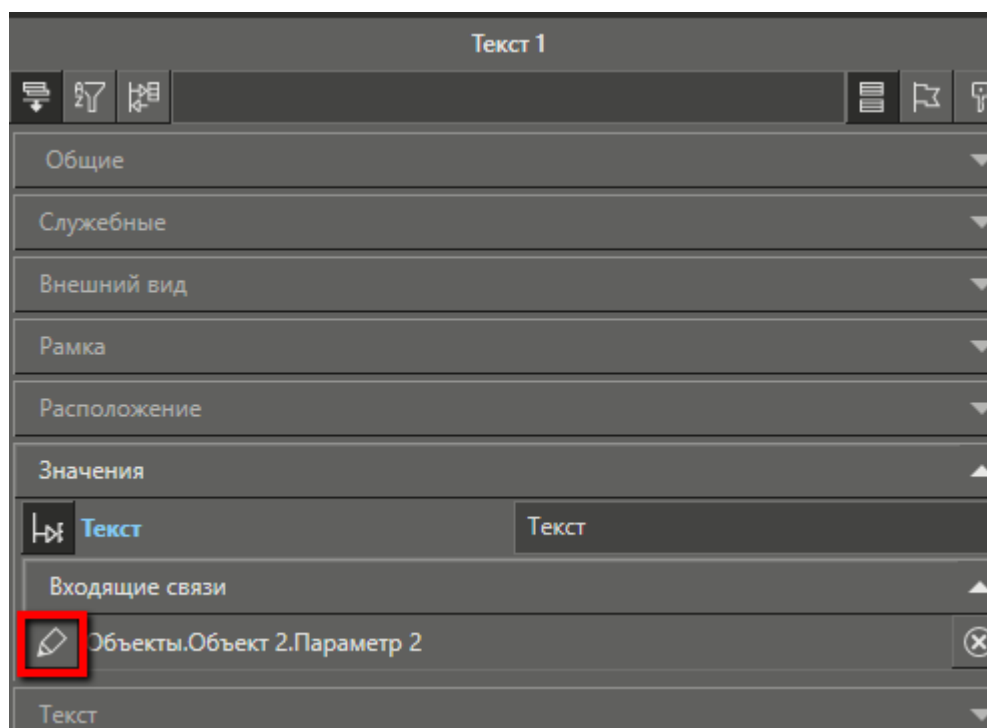
MasterSCADA 4D имеет следующие типы конвертации:

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 <math>X=X_i</math> (<math>X</math> – текущее значение источника), то <math>Y=Y_i</math> (<math>Y</math> – текущее значение приемника).</p> <p>Если <math>X</math> лежит между опорными точками, то <math>Y</math> имеет Значение по умолчанию, заданное в конвертере значений значений.</p>	Такой тип конвертации удобно использовать для работы с параметрами и свойствами, которые имеют тип данных Перечисление
Линейно-кусочная	<p>Если <math>X \leq X_{\min}</math>, то <math>Y</math> имеет Значение по умолчанию, заданное в конвертере значений</p> <p>Если <math>X \geq X_{\max}</math>, то <math>Y=Y_{\max}</math>.</p> <p>Если <math>X_i &lt; X &lt; X_{i+1}</math>, то <math>Y</math> 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 <math>X \leq X_{\min}</math>, то <math>Y</math> имеет Значение по умолчанию, заданное в конвертере значений.</p> <p>Если <math>X \geq X_{\max}</math>, то <math>Y=Y_{\max}</math>.</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

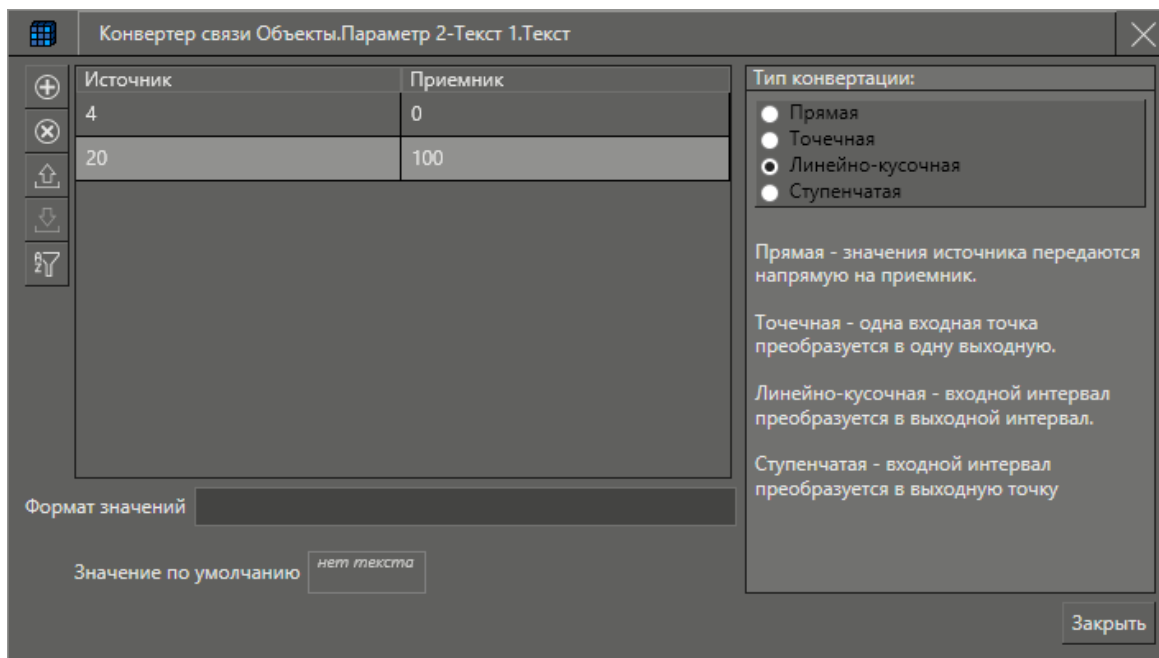
	Если значение источника лежит между опорными точками, приемник сохраняет свое значение.	меньше 70, и красный цвет, когда значение равно или больше 70.
--	---	--

### 8.2.4.1.1.1. 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.
	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 расчёта значения при конвертации.

Данное диалоговое окно используется также при работе с журналом для настройки зависимости стиля строки от какого-либо свойства сообщения.

### 8.2.4.1.1.2. 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 НН:мм:сс.fff (если формат не задан явно). Данный формат используется для даты и времени при всех неявных преобразованиях.

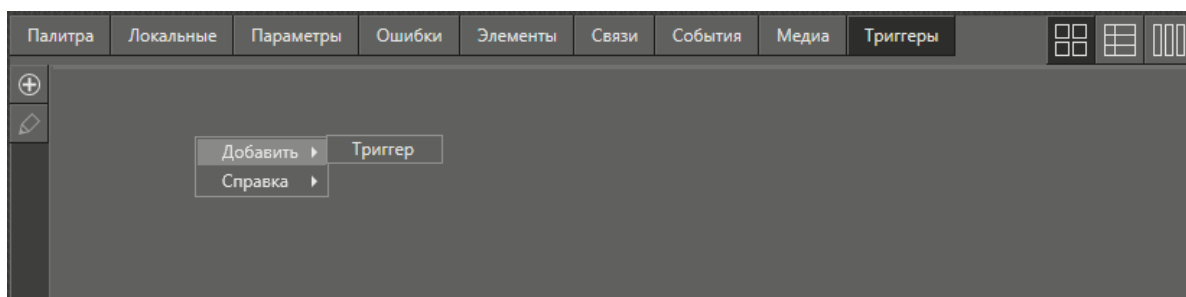
### 8.2.4.2. НМІ. Действия по условию

Динамизация свойства окна или графического элемента может быть сконфигурирована с помощью Триггеров, которые создаются и настраиваются в легенде редактора НМІ на вкладке Триггеры.

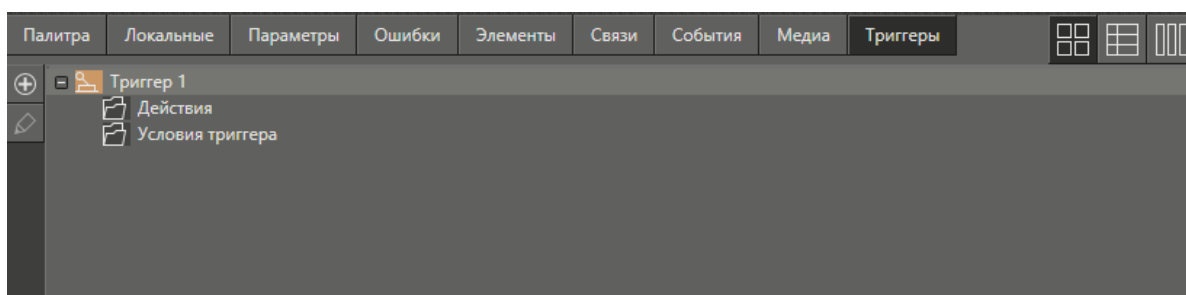
Триггер используется в тех случаях, когда требуется проверить какое-либо условие, и при его удовлетворении выполнить действия.

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 

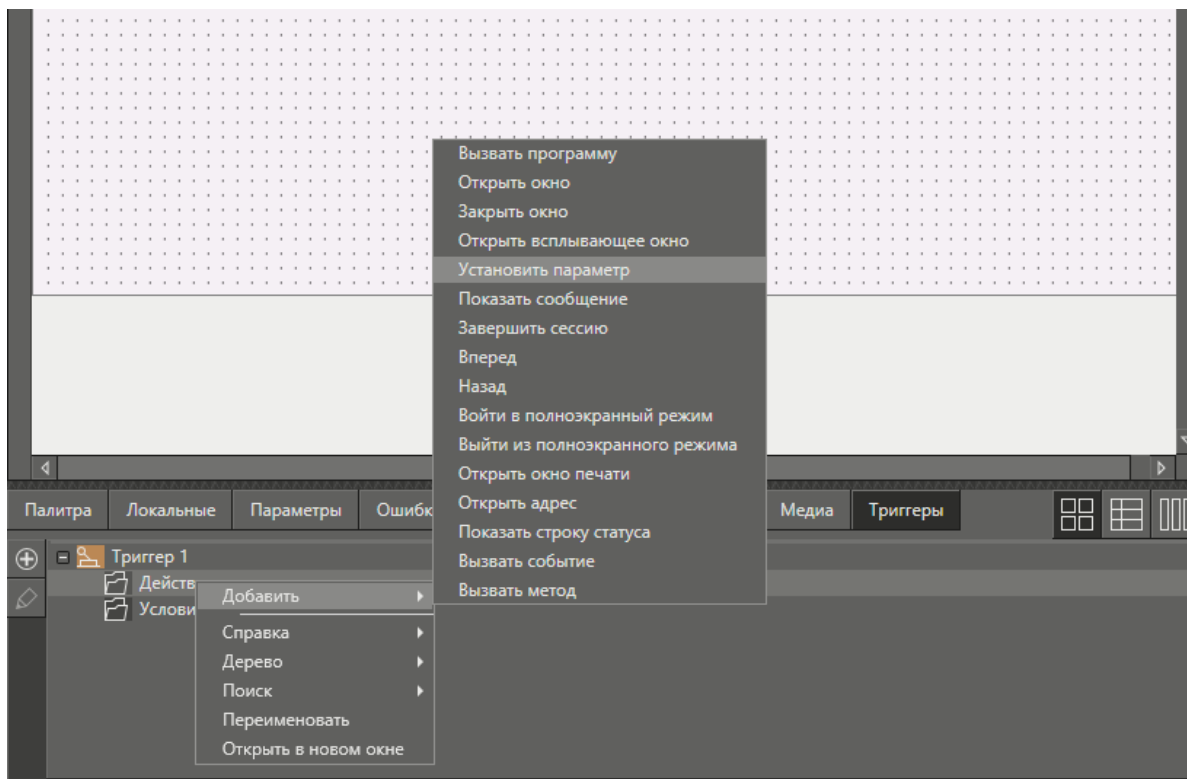


В результате получим:

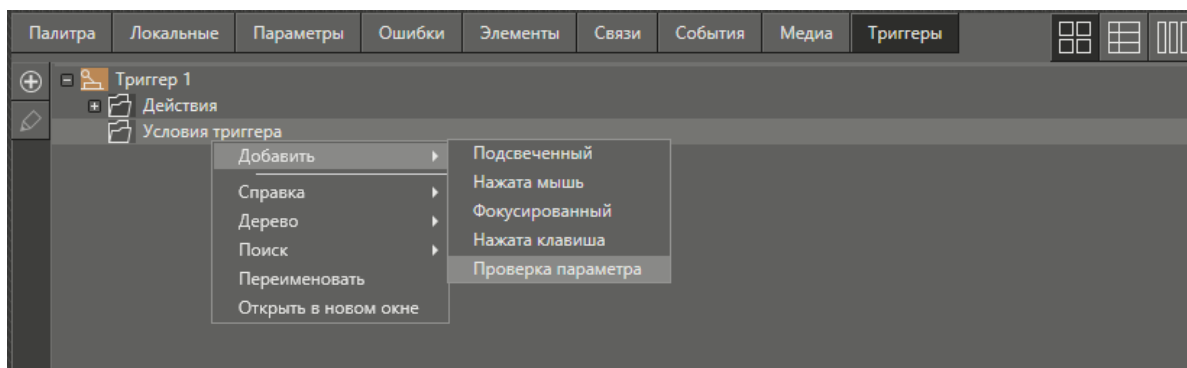


В группу Действия добавляются действия, которые необходимо выполнить, когда условие Триггера будет удовлетворено.





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

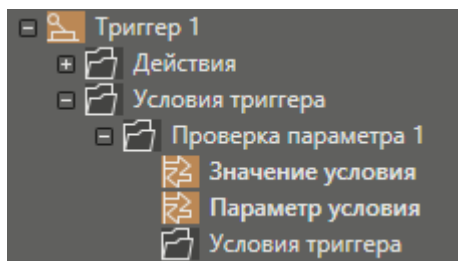
К таким условиям относятся:

- 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

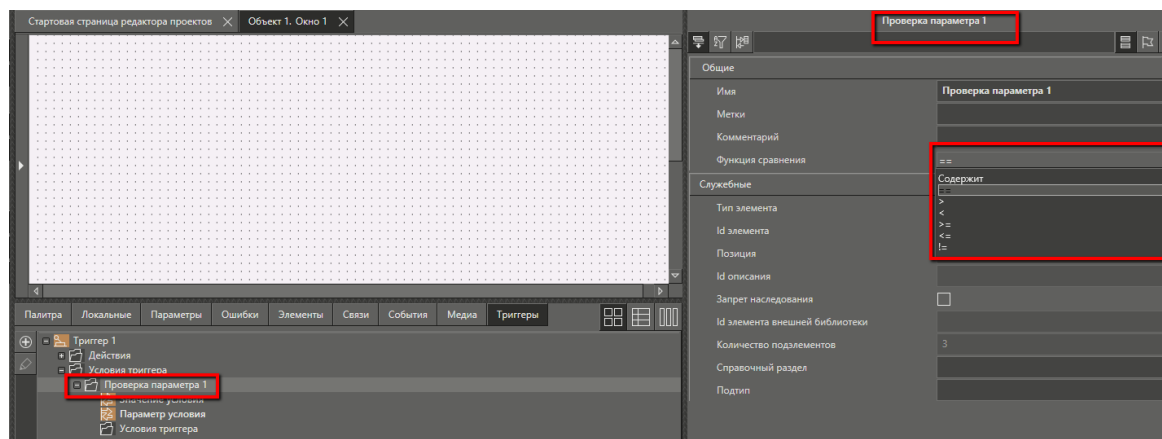
Данное условие конфигурируется с помощью следующих параметров:



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:



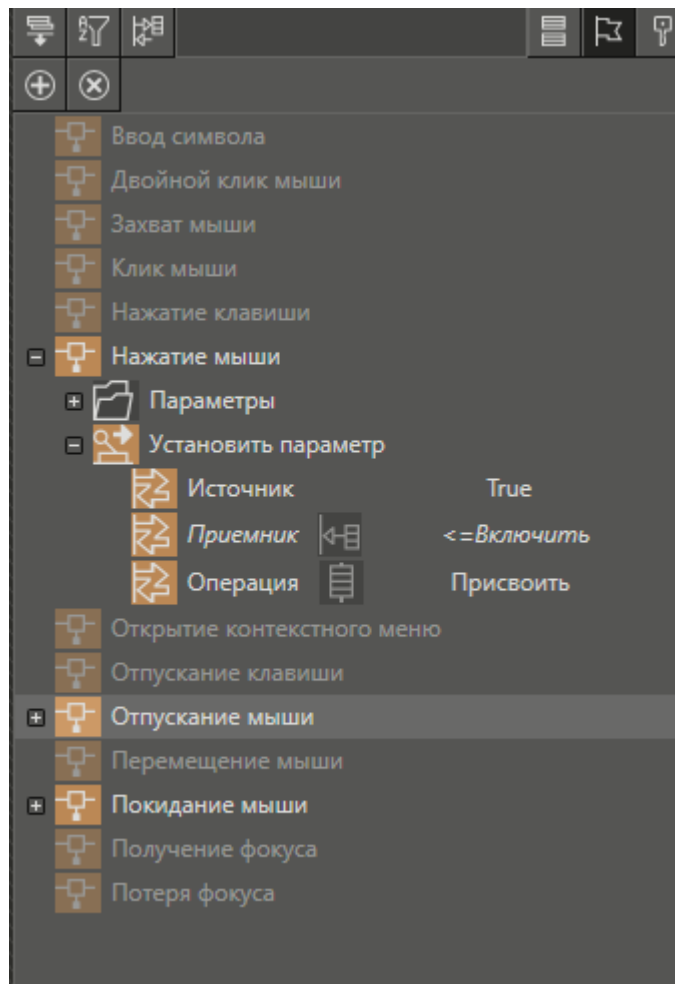
## 8.2.4.3. НМІ. Действия по событию

Для мнемосхемы или графического элемента могут быть заданы predetermined действия, автоматически выполняемые при возникновении указанных predetermined событий (см. НМІ. События и НМІ. Действия).

### 8.2.4.3.1. НМІ. События

Предопределенные 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:

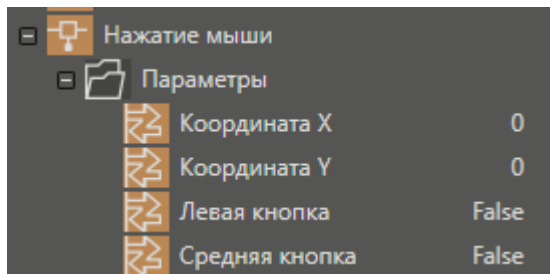


В редких случаях для выполнения операций с событиями может быть использована группа Окно. Группа События мнемосхемы или графического элемента. Кроме того, для выполнения операций с событиями окна предназначена вкладка Легенда НМІ. Вкладка События.

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:



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"> <li>• Символ - выводится значение введенного символа;</li> <li>• Код клавиши-числовые коды нажатой клавиши, используются для работы со служебными клавишами.</li> </ul> <p>Как правило, используется для вызова программы, в которой анализируется нажатая клавиша на клавиатуре (т.е. параметр события связывается с параметром программы).</p> <p>Если в клиенте визуализации клавиша была нажата и не отпущена, то введенный символ повторяется периодически пока клавиша остается нажатой.</p>
Двойной клик мыши	<p>Нажатие и отпускание левой кнопки мыши, выполненные дважды с интервалом не более 0.5 с при нахождении мыши в пределах мнемосхемы или графического элемента.</p>
Захват мыши	<p>Наведение мыши на мнемосхему или графический элемент.</p>

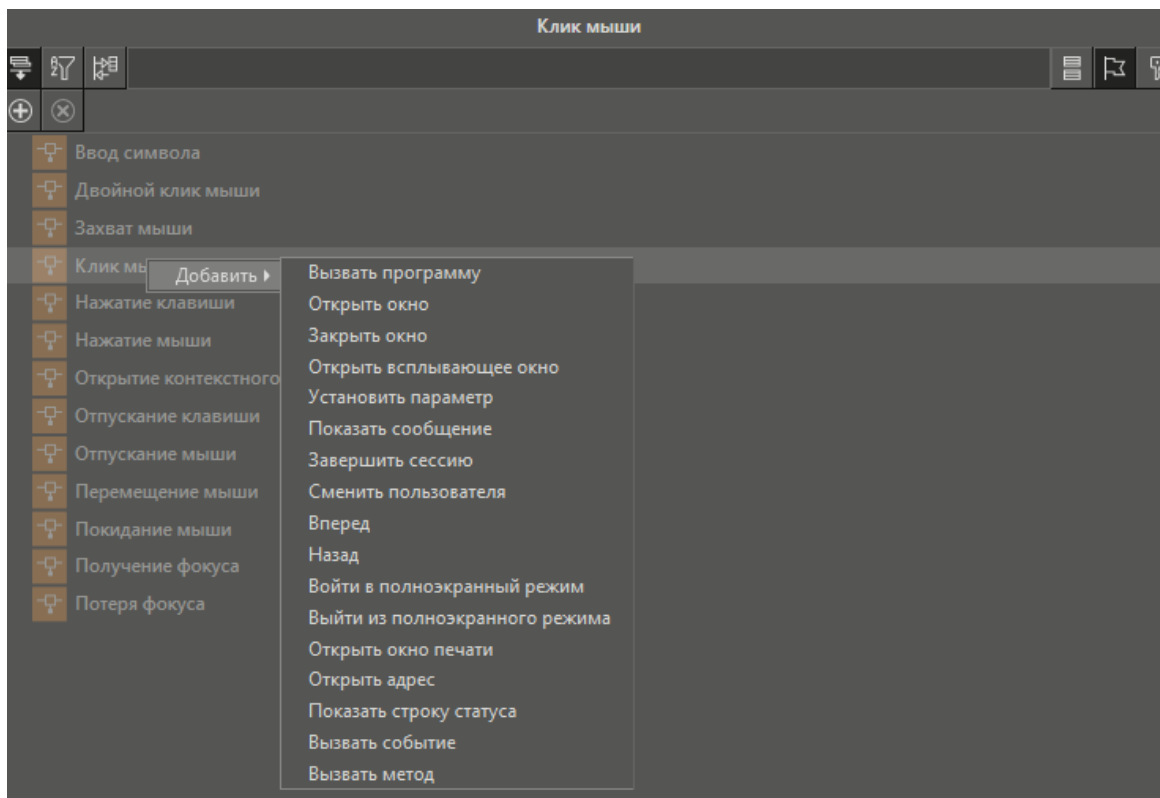
Клик мыши	Нажатие и отпускание левой кнопки мыши при нахождении мыши в пределах мнемосхемы или графического элемента.
Нажатие клавиши	<p>Нажатие клавиши клавиатуры при фокусе на мнемосхеме или графическом элементе. Событием является нажатие не только клавиши ввода символа, но и, например, клавиши CTRL. При нажатии клавиши клавиатуры мнемосхема или графический элемент теряет фокус.</p> <p>Группа Параметры содержит:</p> <ul style="list-style-type: none"> <li>• Символ - выводится значение введенного символа;</li> <li>• Код клавиши-числовые коды нажатой клавиши, используются для работы со служебными клавишами.</li> </ul>
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	<p>Отпускание клавиши клавиатуры при фокусе на мнемосхеме или графическом элементе. Событием является отпускание не только клавиши ввода символа, но и, например, клавиши CTRL.</p> <p>Подгруппа Параметры содержит те же параметры, что и в событии Событие Ввод символа.</p>
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.

### 8.2.4.3.2. НМІ. Действия

Для каждого события (см. НМІ. События ) может быть задано одно или более предопределенных действий.

Для задания действий используется команда Добавить, контекстного меню события, а также команда удаления выделенных элементов:

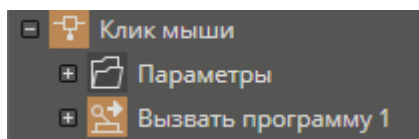


- Действие Вызвать Программу
- Действие Открыть окно

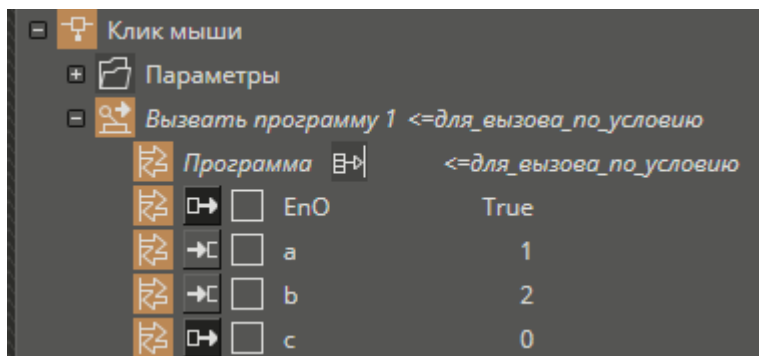
- Действие Закрыть окно
- Действие Открыть всплывающее окно
- Действие Установить параметр
- Действие Показать сообщение
- Действие Завершить сессию
- Действие Сменить пользователя
- Действие Вперед
- Действие Назад
- Действие Войти в полноэкранный режим
- Действие Выйти из полноэкранного режима
- Действие Открыть окно печати
- Действие Открыть адрес
- Действие Показать строку статуса
- Действие Вызвать событие
- Действие Вызвать метод

#### 8.2.4.3.2.1. 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):

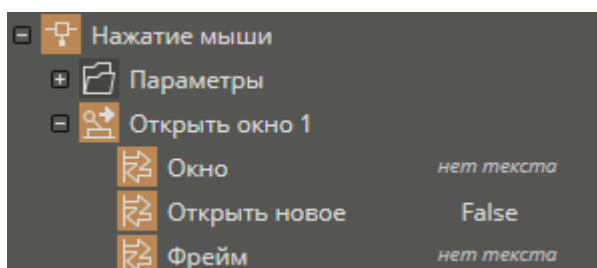


Действие заключается в однократном вызове программы (в т.ч. допускается вызов программы объекта с параметрами свойство Место исполнения = В задаче экрана и свойство Способ исполнения = По вызову).

По команде Добавить контекстного меню действия Вызвать Программу в него может быть добавлен аргумент.

#### 8.2.4.3.2.2. Действие Открыть окно

При задании этого действия группа события принимает следующий вид (на рисунке показано событие Нажатие мыши в окне свойств; см. также НМІ. Действия):



Действие заключается в открытии окна, привязанного к параметру Окно. Для конфигурирования действия используются дополнительные параметры:

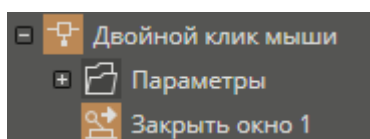
- Открыть новое – если параметр имеет значение TRUE, то привязанное к параметру окно открывается на новой вкладке браузера, а если False, то на той же. С помощью данной опции может быть открыто не более одной новой вкладки браузера;
- Фрейм – если к этому параметру привязан графический элемент Контейнер 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).

#### 8.2.4.3.2.3. Действие Закрыть окно



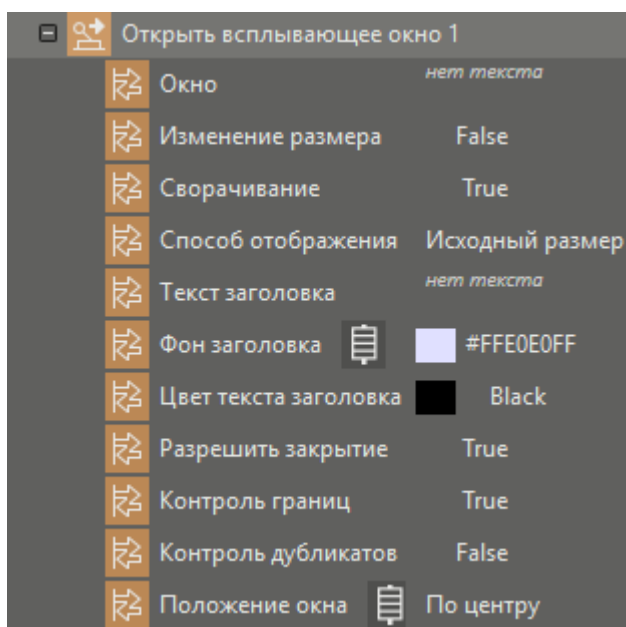
При задании этого действия группа события принимает следующий вид (на рисунке показано событие Двойной клик мыши, см. также НМІ. Действия):



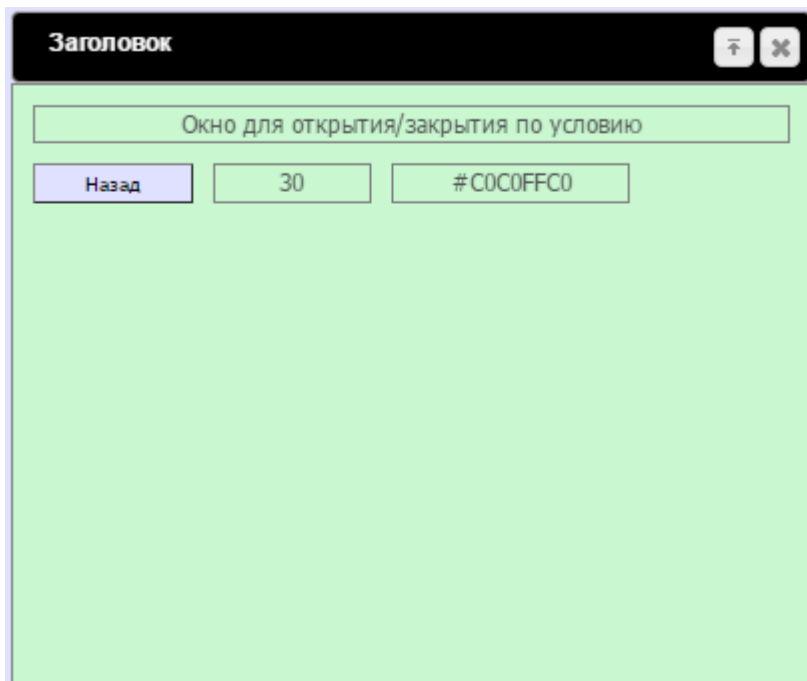
Действие заключается в закрытии текущего всплывающего окна.

#### 8.2.4.3.2.4. 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: :



Важно! В НМІ 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:

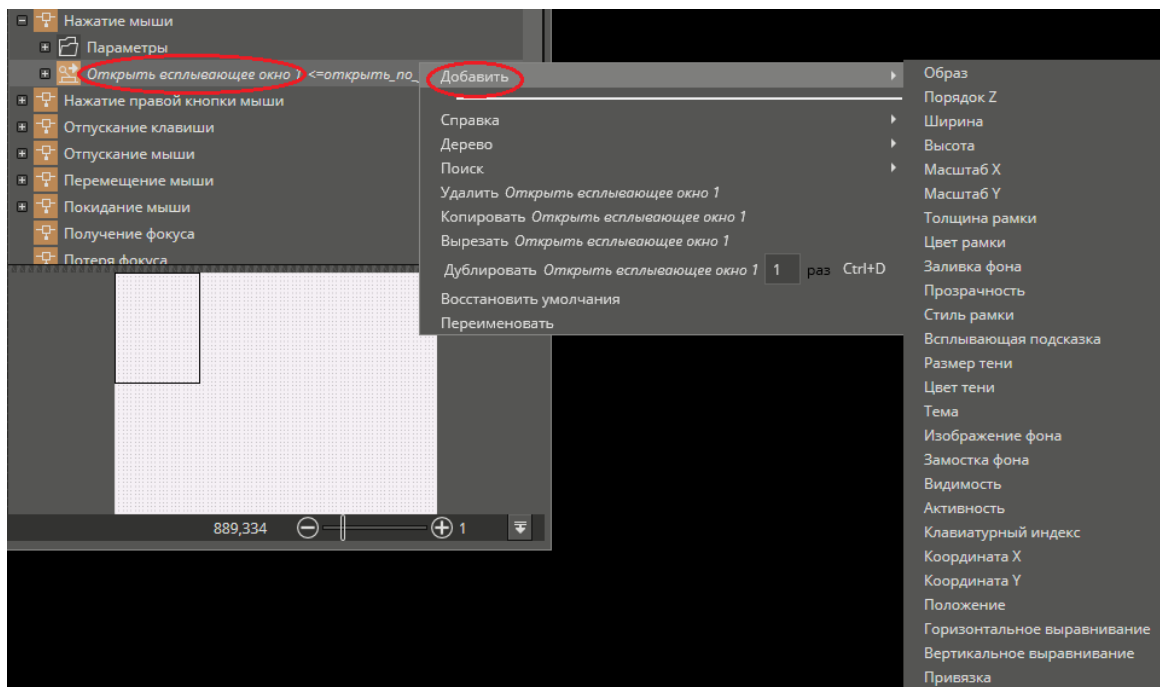
Параметр	Назначение
Окно	Устанавливает связь с окном, которое требуется открыть. Для установления связи необходимо перетащить окно из дерева проекта на свойство.
Изменение размера	Определяет возможность изменения размера окна в клиенте визуализации. Если параметр имеет значение TRUE, то разрешено изменение размеров всплывающего

	окна (растягивание рамки разрешено при условии, что тип ширины/высоты в настройках открываемого окна - относительный).
Сворачивание	Определяет возможность сворачивания окна в клиенте визуализации. Если параметр имеет значение False, то в заголовке будет отсутствовать инструмент  .
Способ отображения	Определяет способ подгонки окна просмотра и отображаемого окна.
Текст заголовка	Определяет текст заголовка всплывающего окна. Если текст заголовка не задан или не установлена связь с параметром, то используется название объекта/узла, которому принадлежит окно.
Фон заголовка	Определяет цвет фона заголовка. При нажатии на кнопку  , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.
Цвет текста заголовка	Определяет цвет текста заголовка. При нажатии на кнопку  , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.
Разрешить закрытие	Определяет возможность закрытия окна в клиенте визуализации.
Контроль границ	Определяет возможность перемещения/открытия окна за границами видимости клиента визуализации.
Контроль дубликатов	Определяет необходимость закрытия ранее открытых копий и экземпляров данного окна.

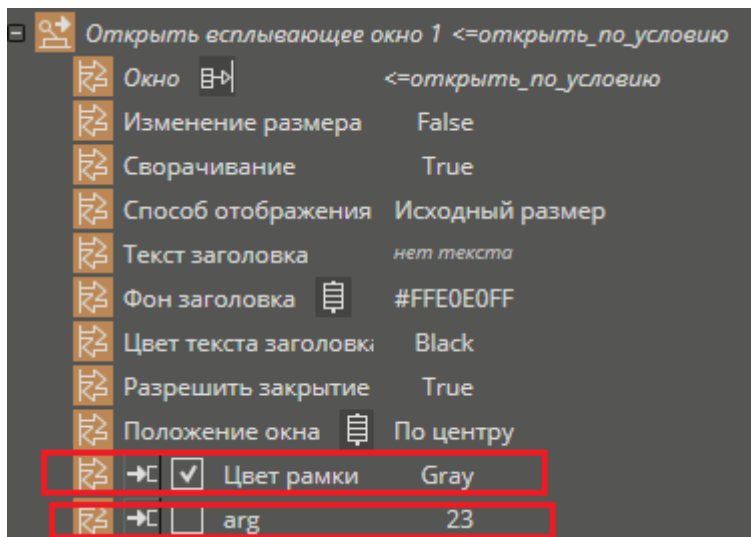
	<p>Если задано значение TRUE и при этом открывается окно объекта, то будут закрыты все ранее открытые копии этого окна.</p> <p>Если задано значение TRUE и при этом открывается окно экземпляра объекта, то будут закрыты все ранее открытые окна любых экземпляров этого объекта.</p>
<p>Положение окна</p>	<p>Определяет место открытия окна. Возможные варианты:</p> <ul style="list-style-type: none"> <li>• По центру экрана;</li> <li>• По месту указателя мыши (по умолчанию);</li> <li>• По заданным координатам;</li> <li>• С заданным смещением относительно элемента;</li> <li>• С заданным смещением относительно окна элемента.</li> </ul> <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>

#### 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:

- C 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

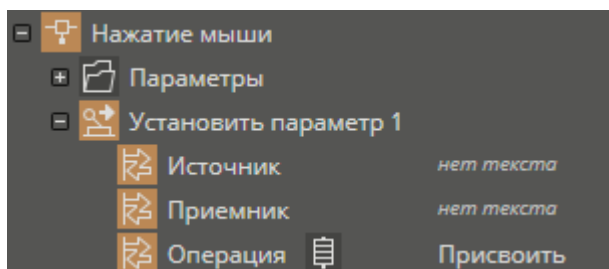
с постоянным значением), то вместо поля отображается имя привязанного параметра с указанием направления передачи значения:

 counter – из параметра в аргумент;

 Переменная\_1 – из аргумента в параметр.

### 8.2.4.3.2.5. 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):

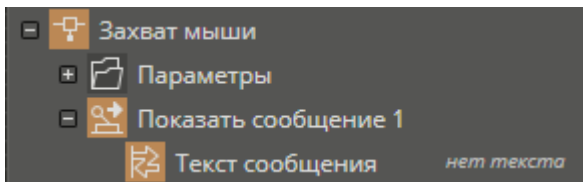


Действие заключается в выполнении математического действия, заданного параметром Операция:

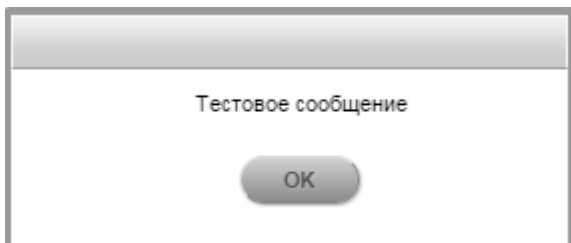
- если Операция=Присвоить, то Приемник := Источник;
- при других операциях Приемник := Приемник <операция> Источник (например, при Операция=Разделить Приемник := Приемник/Источник).

### 8.2.4.3.2.6. 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):

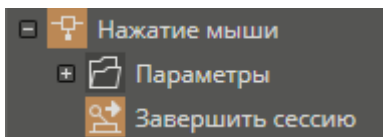


Ниже показан вид сообщения при возникновении события в реальном времени (чтобы закрыть сообщение, нужно нажать на кнопку ОК):



### 8.2.4.3.2.7. 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:

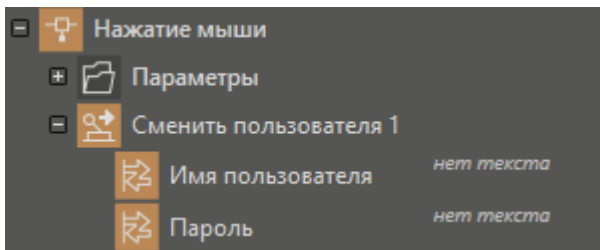
Логин  
Пользователь 1

Пароль

Вход

### 8.2.4.3.2.8. Действие Сменить пользователя

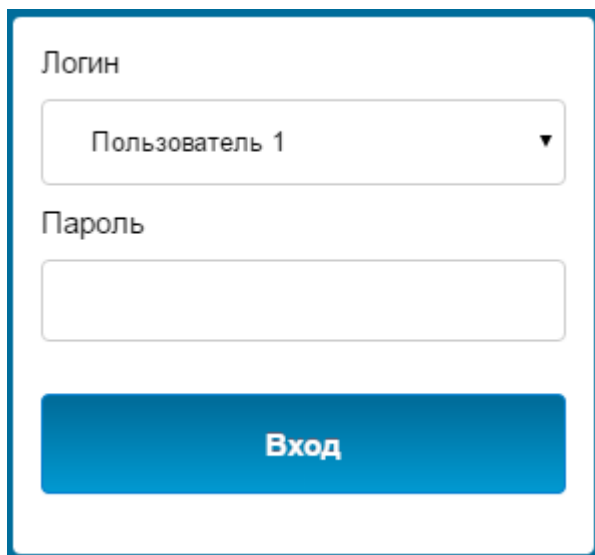
При задании этого действия группа события принимает следующий вид (на рисунке показано событие Нажатие мыши, см. также НМІ. Действия):



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 бу-

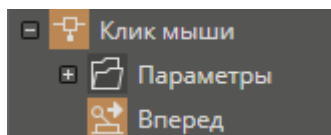
дет. Если параметры действия примут значения, которые не соответствуют существующим пользователям, то откроется окно авторизации, при этом сессия предыдущего пользователя будет продолжена до тех пор, пока не пройдет авторизация нового.



Важно! Действие Сменить пользователя работает только в НМІ v2.

#### 8.2.4.3.2.9. Действие Вперед

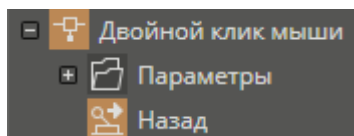
При задании этого действия группа события принимает следующий вид (на рисунке показано событие Клик мыши, см. также НМІ. Действия):




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).

#### 8.2.4.3.2.10. Действие Назад

При задании этого действия группа события принимает следующий вид (на рисунке показано событие Двойной клик мыши, см. также НМІ. Действия):

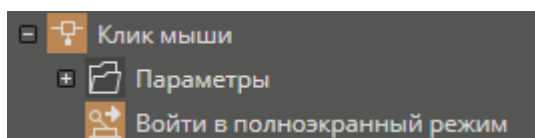




Если действие задано, то при работе в реальном времени при возникновении события выполняется команда  Назад браузера (т.е. если в браузере существует история переходов по страницам, то данная команда – это переход на предыдущую страницу согласно истории переходов).

#### 8.2.4.3.2.11. Действие Войти в полноэкранный режим

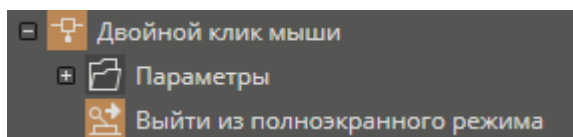
При задании этого действия группа события принимает следующий вид (на рисунке показано событие Клик мыши, см. также НМІ. Действия):



Если действие задано, то при работе в реальном времени при возникновении события применяется полноэкранный режим отображения.

#### 8.2.4.3.2.12. Действие Выйти из полноэкранного режима

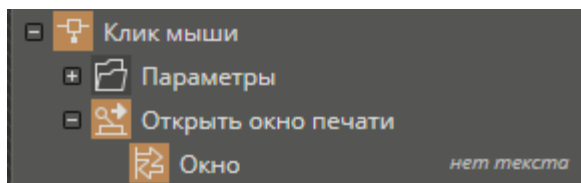
При задании этого действия группа события принимает следующий вид (на рисунке показано событие Двойной клик мыши, см. также НМІ. Действия):



Если действие задано, то при работе в реальном времени при возникновении события применяется обычный (не полноэкранный) режим отображения.

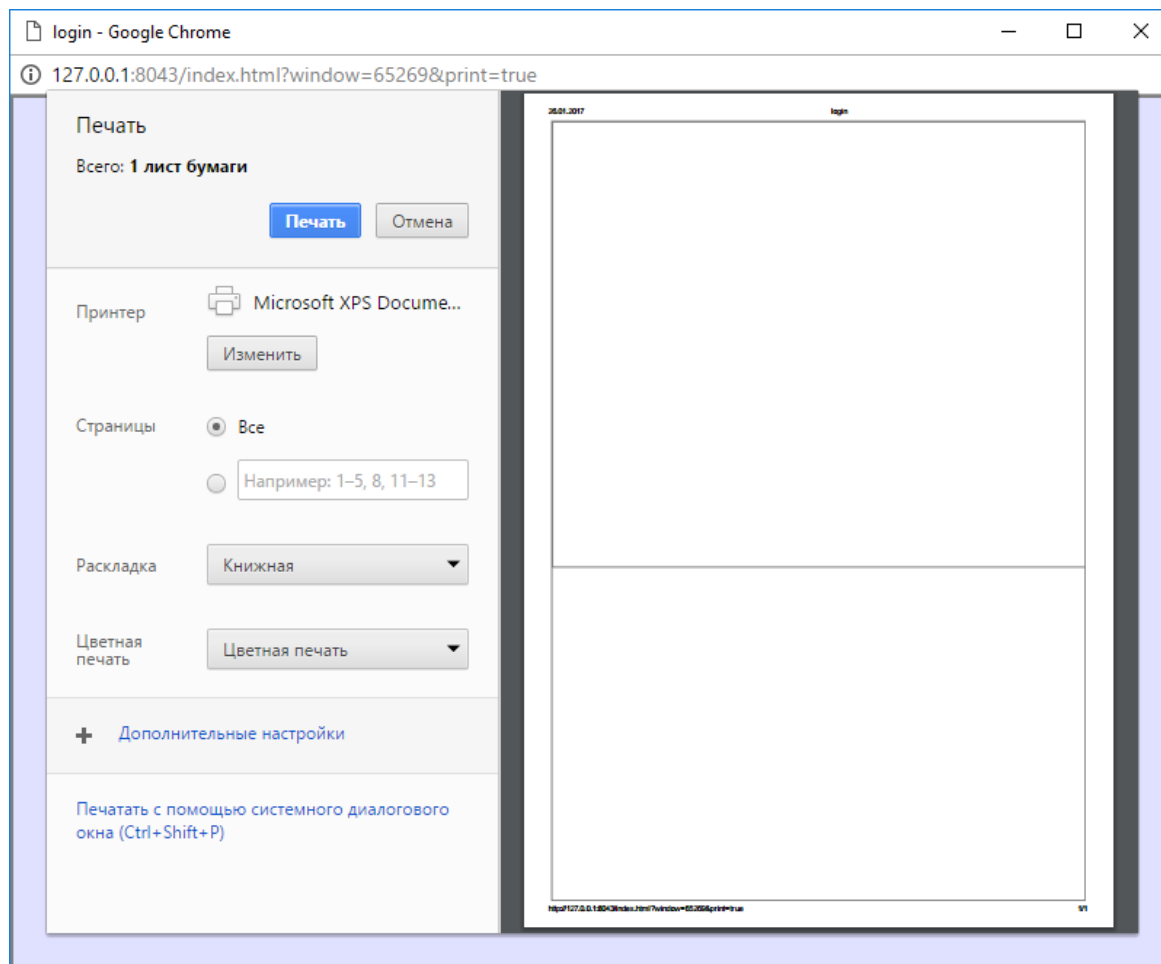
#### 8.2.4.3.2.13. Действие Открыть окно печати

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:



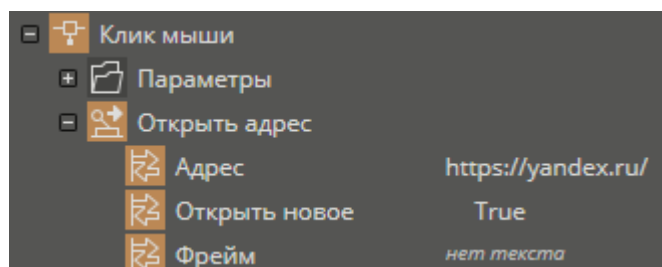
### Работа в разных версиях НМІ

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.

#### 8.2.4.3.2.14. 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):

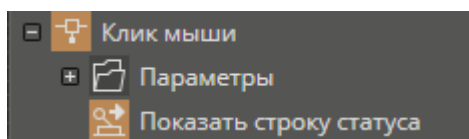


Данное действие – это действие Открыть окно, совмещенное с отображением а нем ресурса с адресом URL, заданным параметром Адрес.

Данное действие используется также для того, чтобы открывать отчеты, сгенерированные генератором отчетов MasterSCADA 4D.

#### 8.2.4.3.2.15. Действие Показать строку статуса

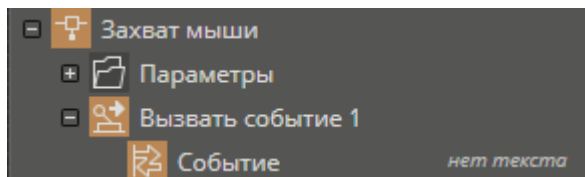
При задании этого действия группа события принимает следующий вид (на рисунке показано событие Клик мыши , см. также НМІ. Действия):



Если действие задано, то при работе в реальном времени при возникновении события отображается строка статуса (о строке статуса см. Отображение мнемосхем в браузере ).

#### 8.2.4.3.2.16. Действие Вызвать событие

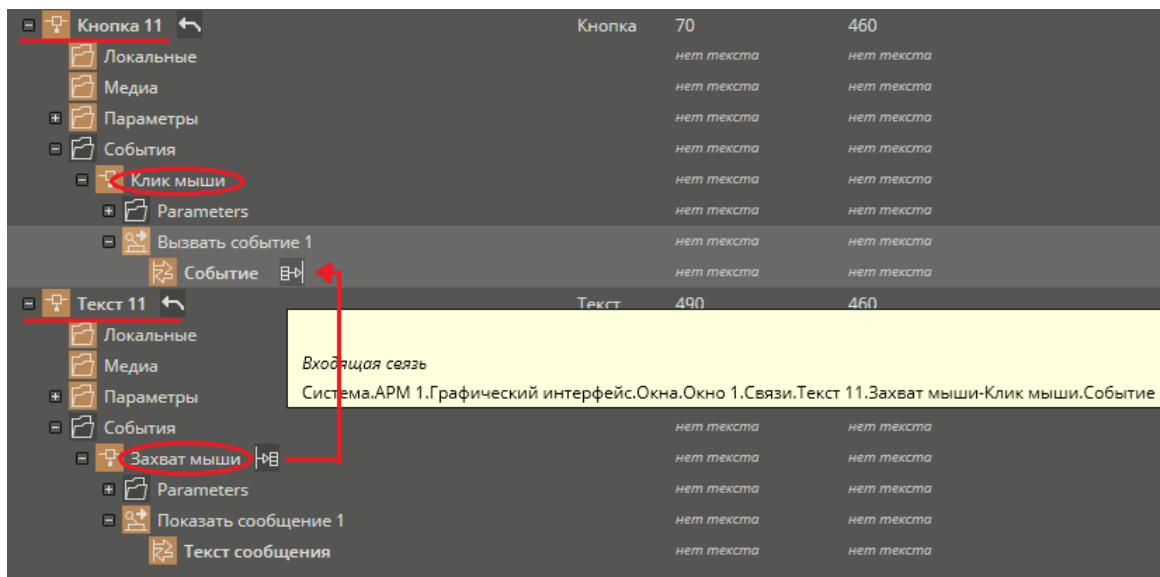
При задании этого действия группа события принимает следующий вид (на рисунке показано событие Захват мыши , см. также НМІ. Действия):



К параметру Событие привязывается некоторое событие.

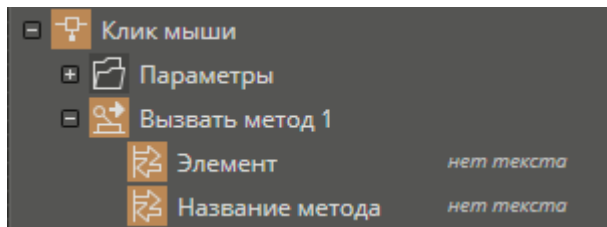
#### Пример

В примере ниже при событии Клик мыши кнопки Кнопка 11 наступает событие Захват мыши элемента Текст 11 (и в результате выполняется действие Показать сообщение).

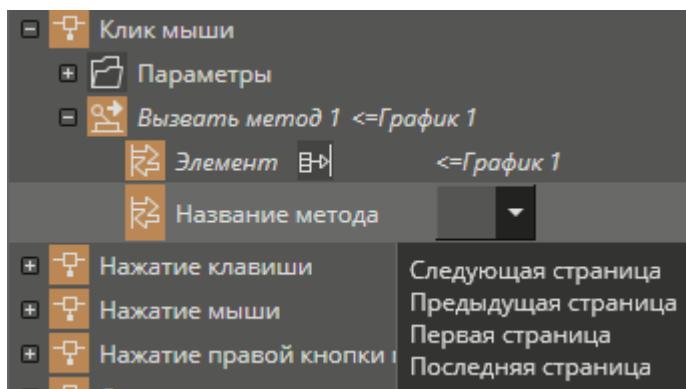


### 8.2.4.3.2.17. Действие Вызвать метод

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:

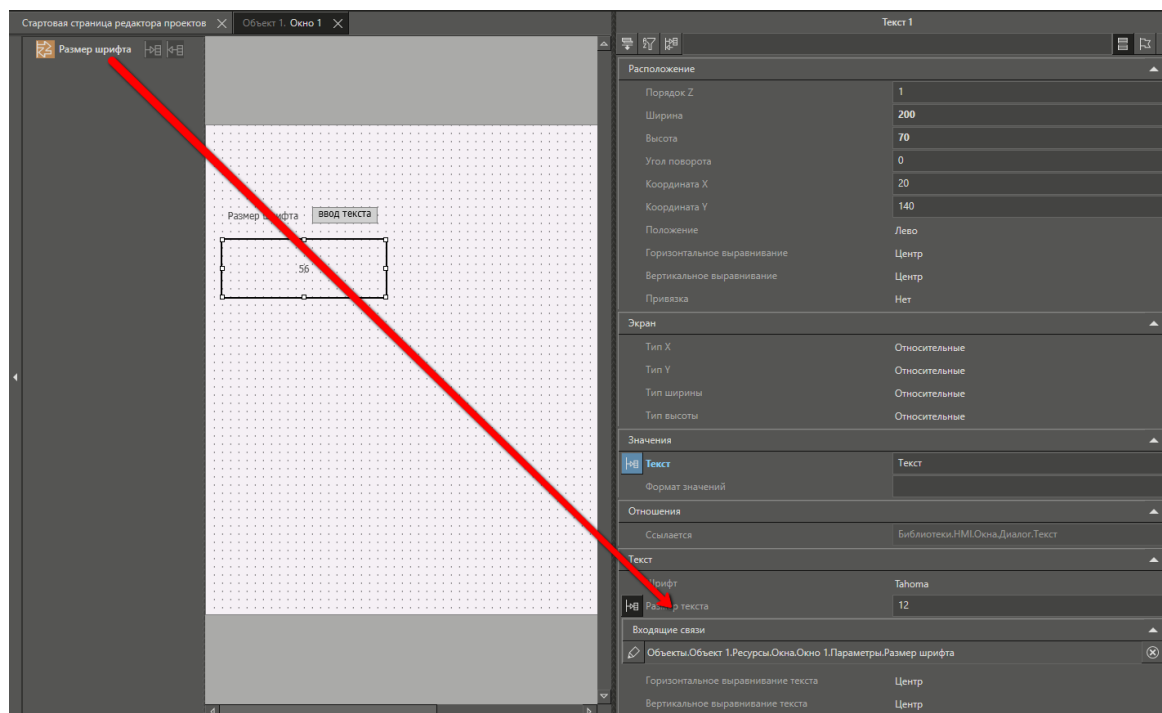


По команде Добавить контекстного меню действия Вызвать метод в действие можно добавить аргумент.

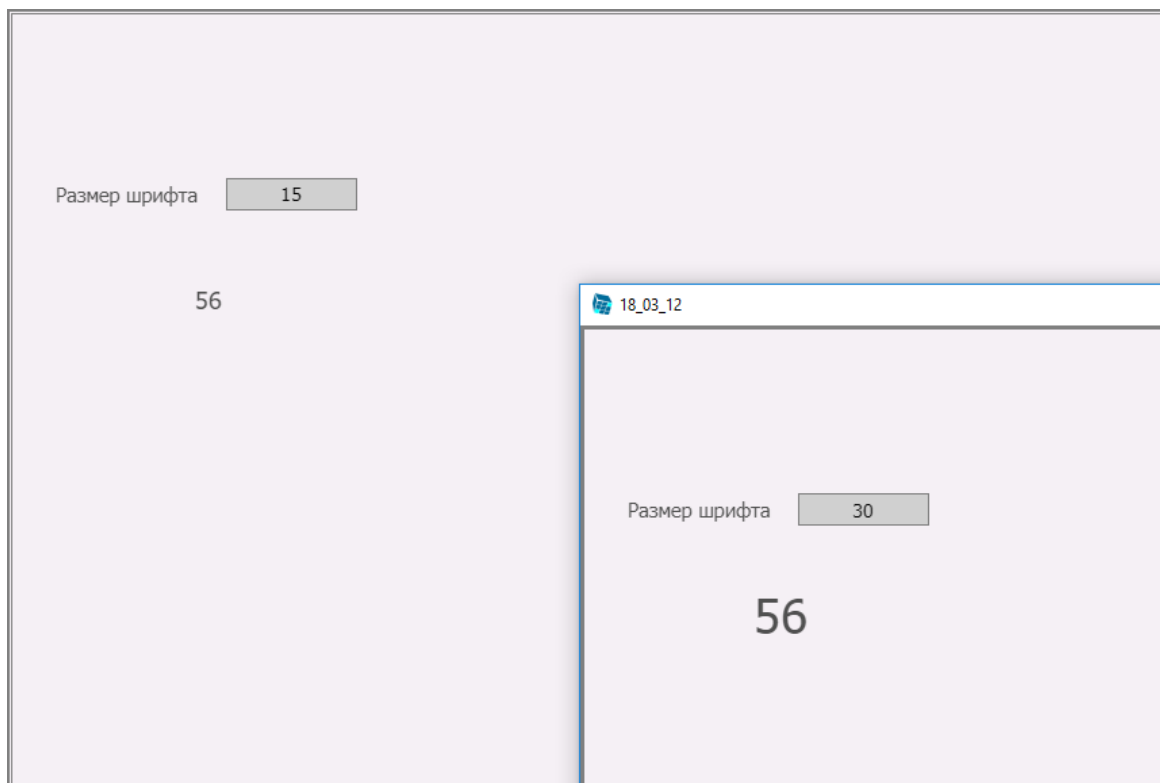
## 8.2.5.Клеммники в редакторе НМІ

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.



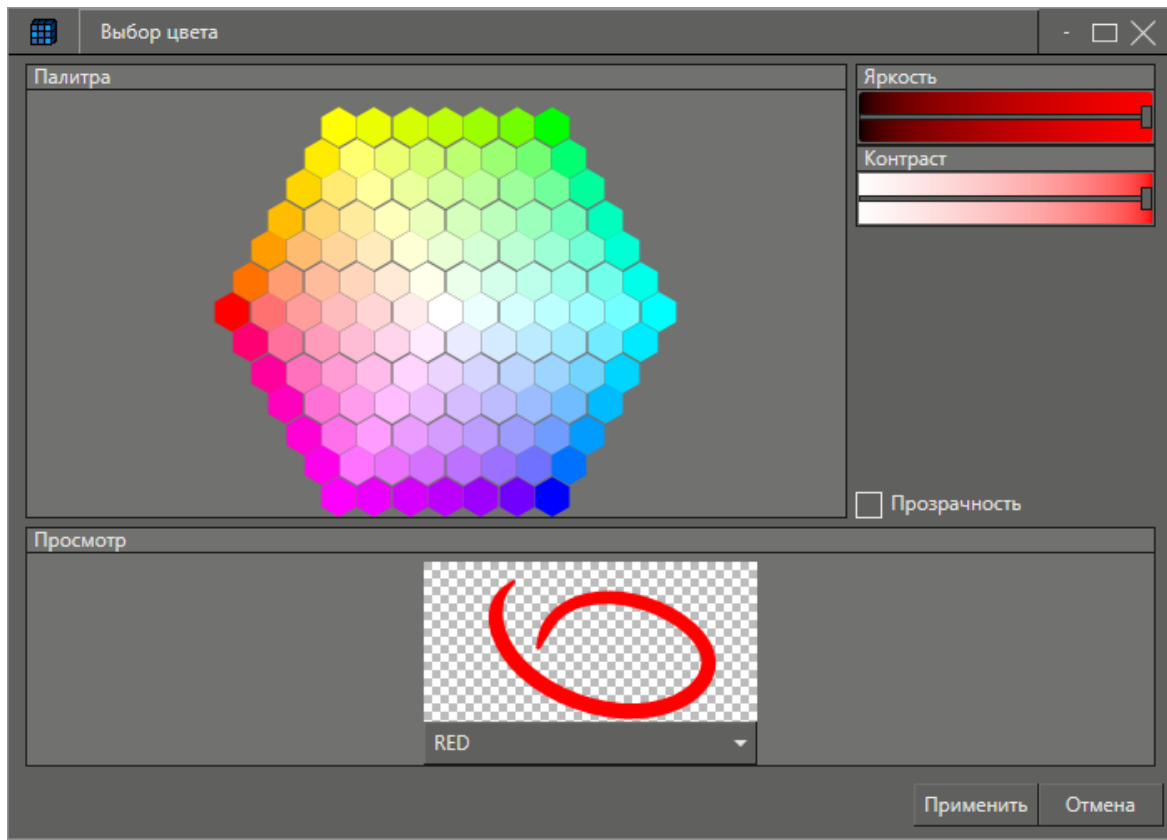
Кроме того, Клеммники окна используются при создании библиотечных окон. Когда в библиотеке создается окно, то определяются параметры, которые необходимо использовать для динамизации. Затем, когда экземпляр окна добавляется в объект, настраивается связь между клеммником окна и параметром объекта.

### 8.2.6. Диалоговое окно Выбор цвета

В зависимости от настраиваемого свойства может открыться либо простой диалог выбора цвета, либо диалог выбора цвета с выбором градиента.

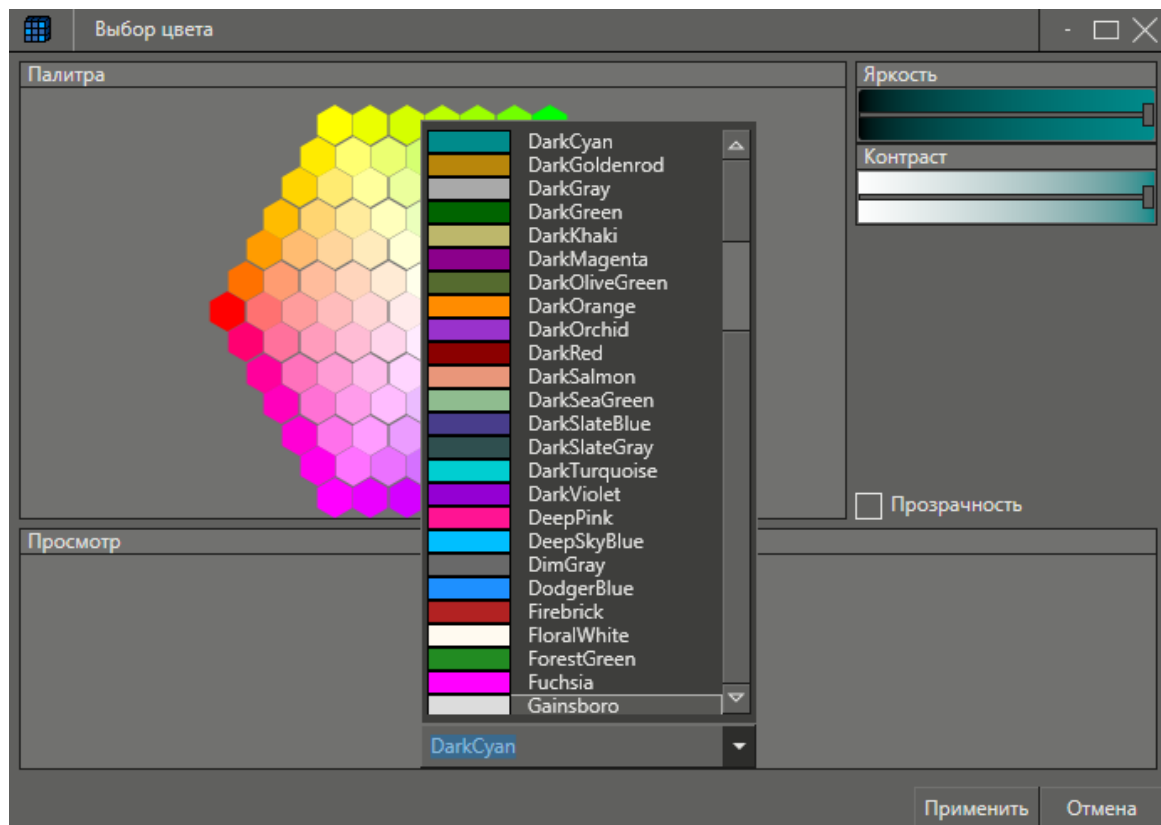
Простой диалог

Простой диалог выбора цвета открывается для настройки цветов линий и текстов, и позволяет выбрать необходимый цвет, например, рамки элемента:



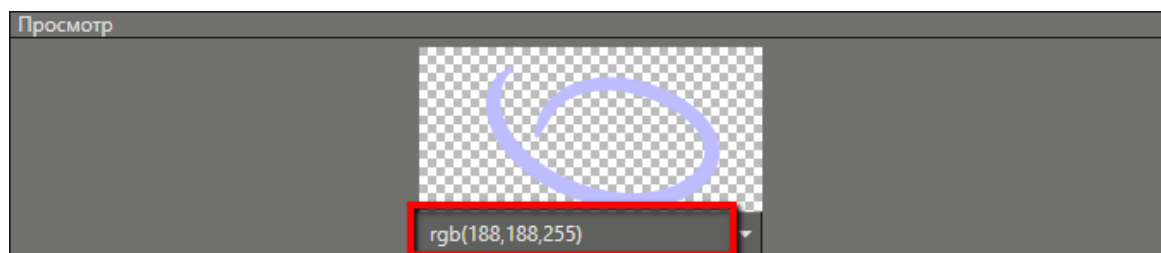
### Порядок работы

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:

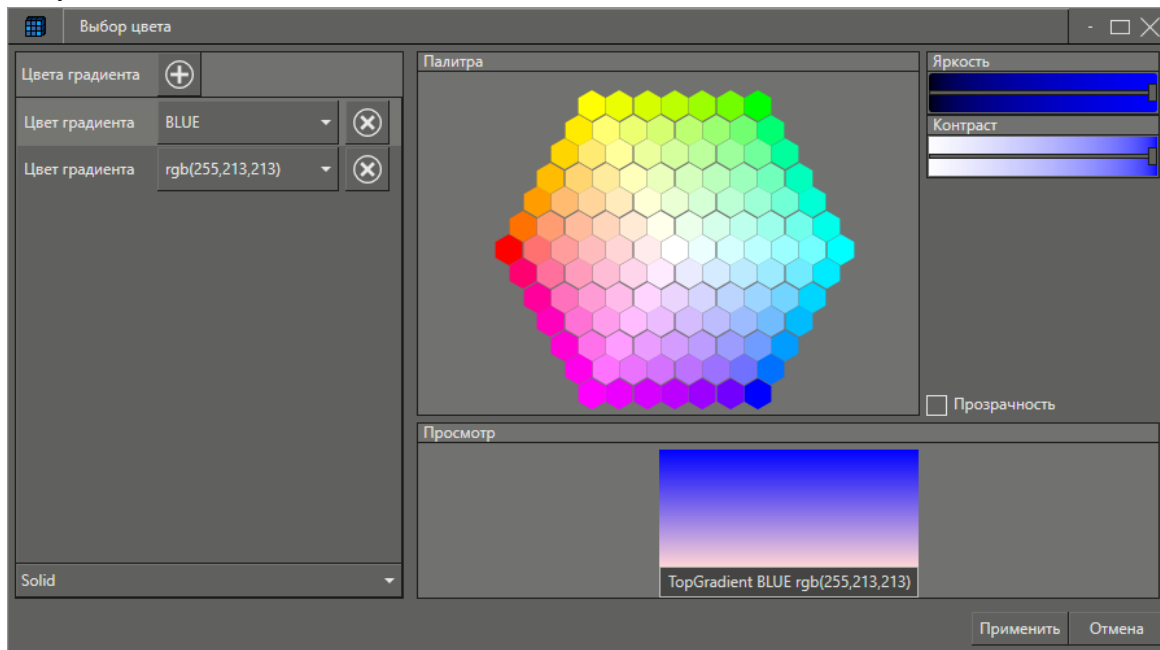


В том случае, если требуется задать цвет элемента, указанный в техническом задании в формате HEX, необходимо воспользоваться таблицами соответствия цветов, которые можно найти в сети Интернет.

Диалог выбора цвета с настройкой градиента

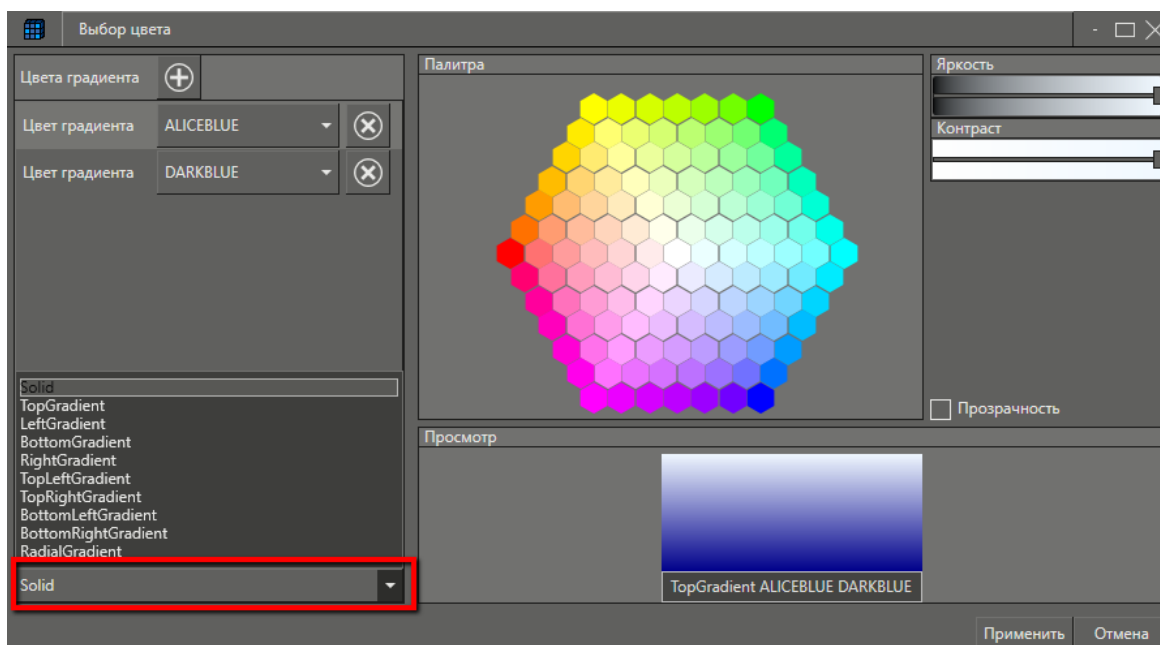


Диалог выбора цвета с настройкой градиента, позволяет задать, например, заливку фона в случае, когда необходимо обеспечить плавный переход от одного цвета к другому.



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.

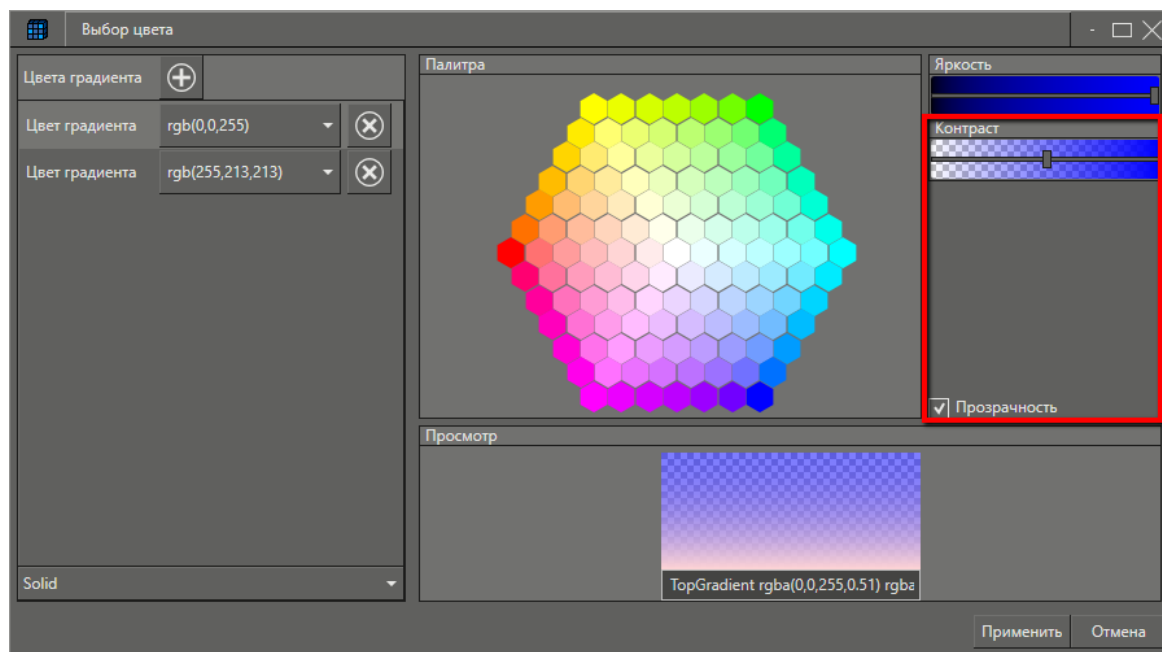
Если требуется реализовать градиент, то в левой части добавляется необходимое количество цветов градиента, а в выпадающем списке задается способ перехода от одного цвета к другому:



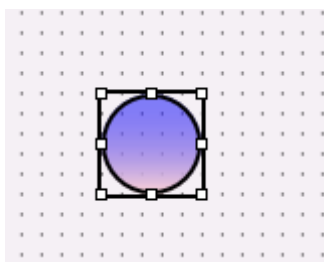
### Настройка прозрачности

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:

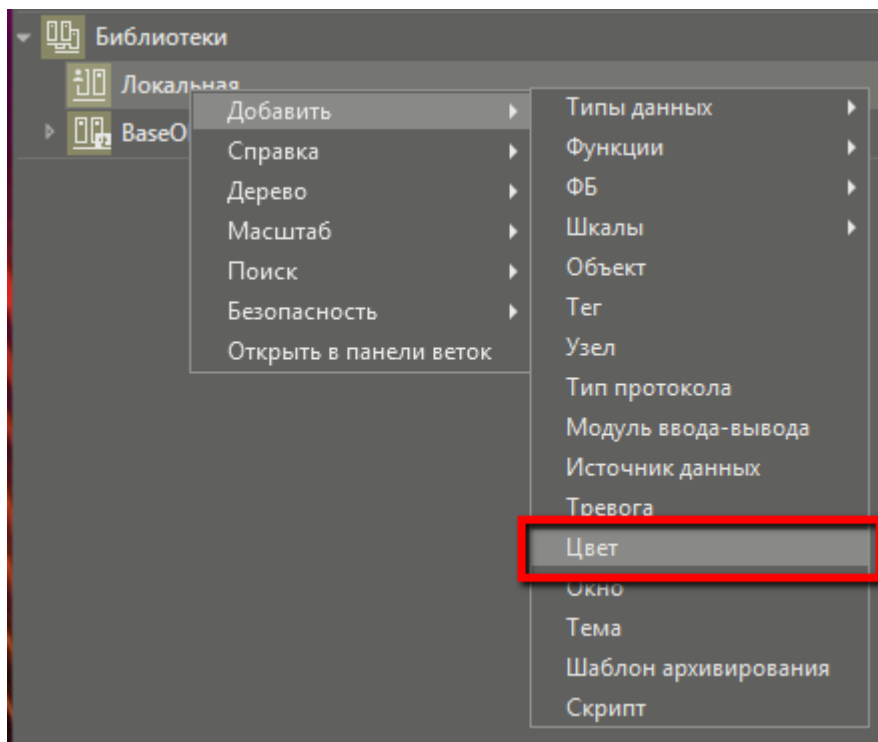


Результат:

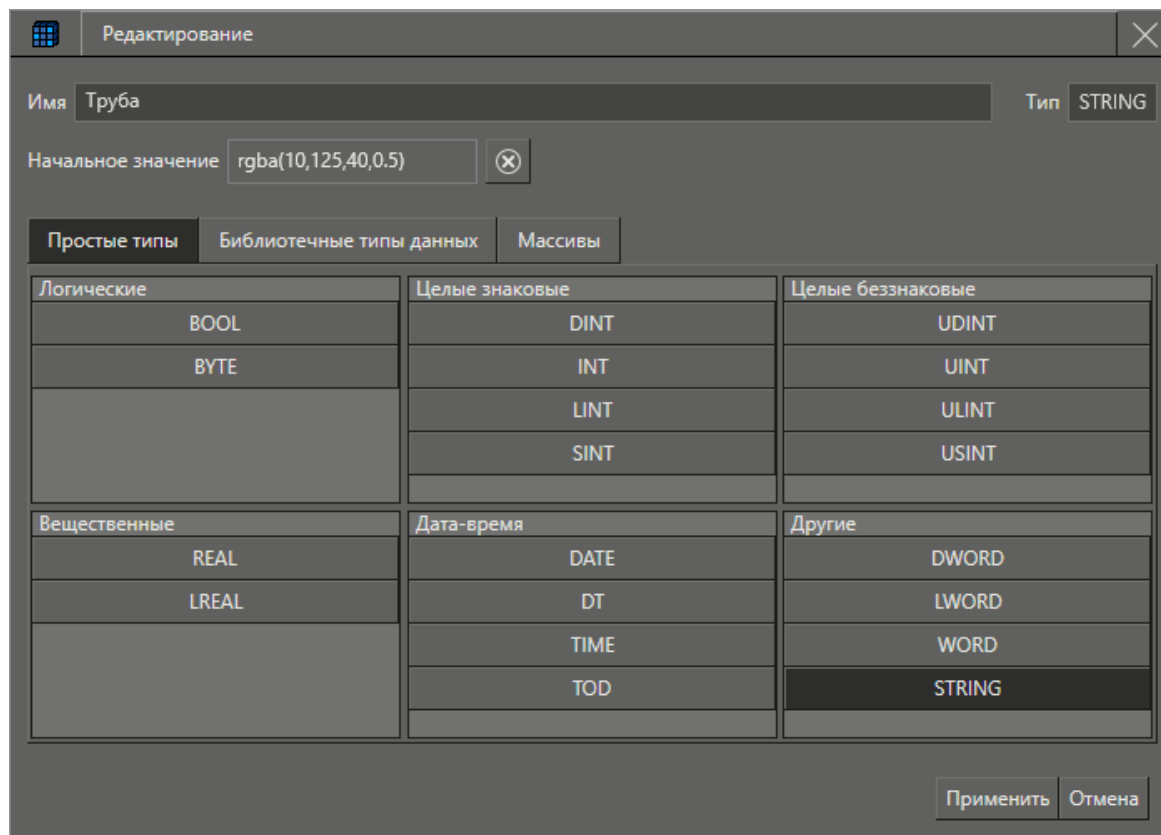


### 8.2.7. Создание цвета пользователя

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:



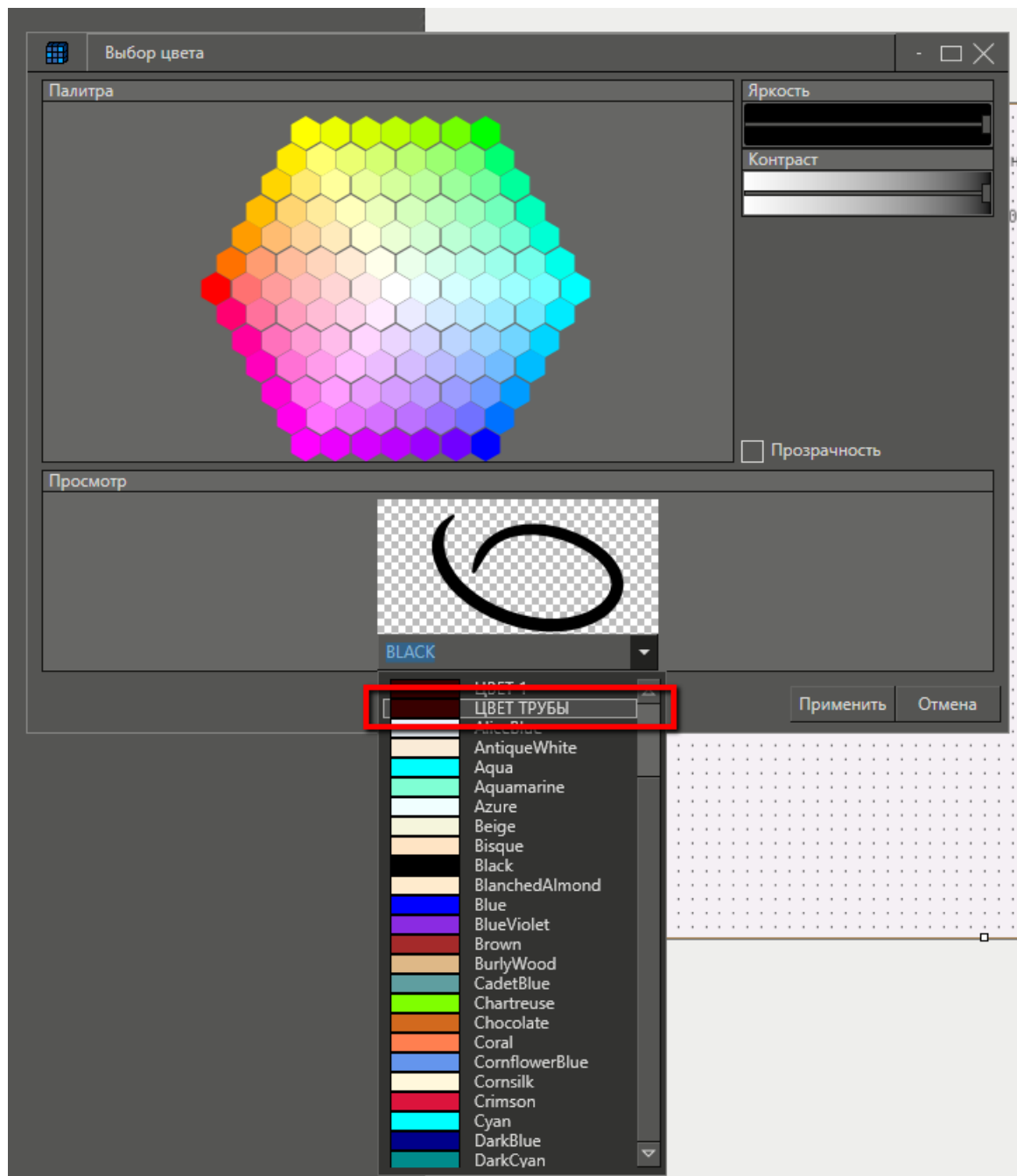
RGBA расшифровывается как Red Green Blue Alpha. Alpha - это значение прозрачности, при 1 цвет полностью не прозрачный, при 0 - абсолютно прозрачный.

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:



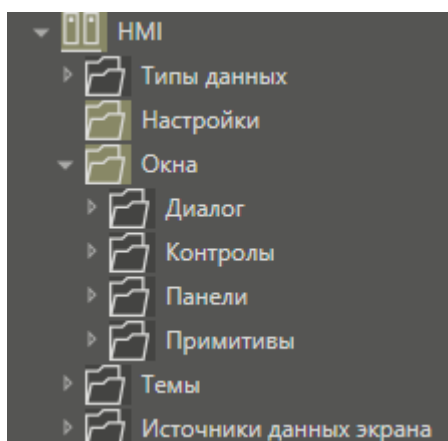
Либо выбрать в окне Выбор цвета:



Важно! Недопустимо перетаскивать элемент из библиотеки в поле свойства.

### 8.3. Библиотека НМІ

Библиотека HMI – это библиотека шаблонов графических и вспомогательных элементов, необходимых для работы в редакторе HMI.

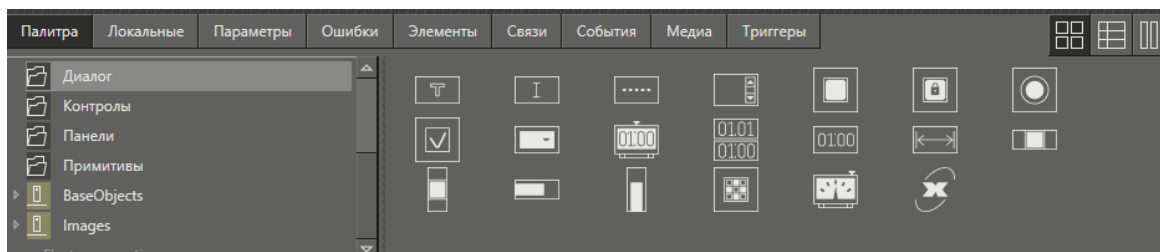


Элементы, содержащиеся в данной библиотеке доступны на вкладке Палитра в левом редактора HMI, поэтому данная библиотека не отображается в упрощенном дереве.

### 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:

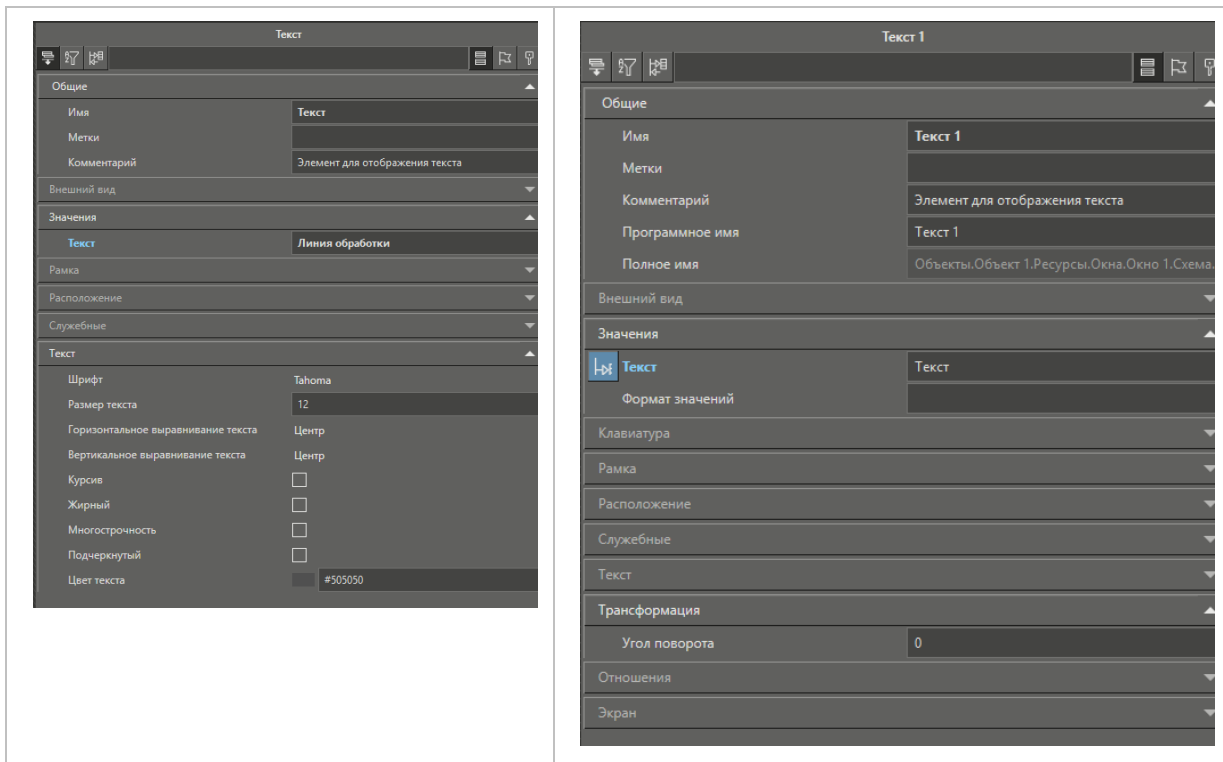


Вид элемента после добавления в рабочую область редактора НМІ:




Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
---------------	--------------



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



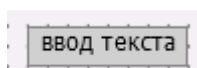
Данный элемент предназначен для ввода в клиенте визуализации значений параметров простых типов при помощи реальной или виртуальной клавиатуры. Чаще всего элементы Текстовый ввод в окне создаются путем перетаскивания параметра в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.

В палитре редактора HMI элемент Текстовый ввод находится в категории Диалог.

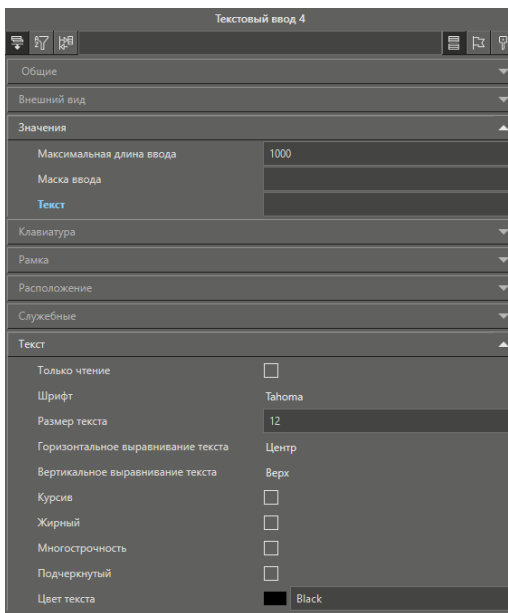
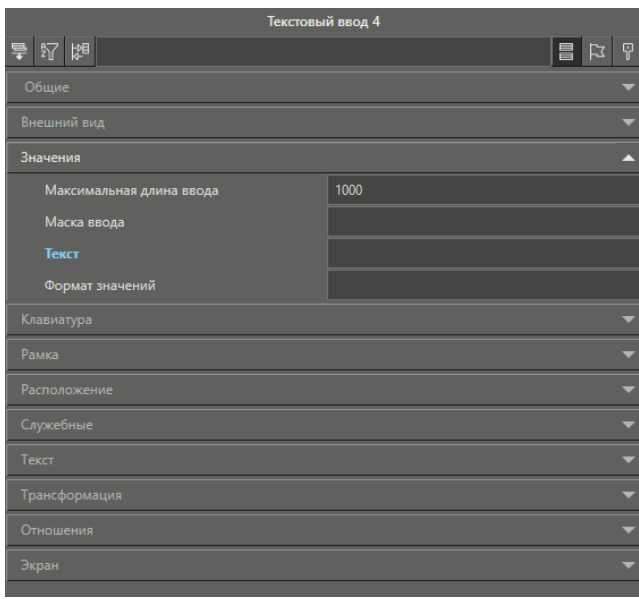
Вид элемента в палитре:



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 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 был перетасчен 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.
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.

### 8.3.1.3. Пароль

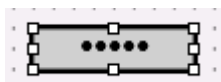
Данный элемент предназначен для ввода пароля в клиенте визуализации при помощи реальной или виртуальной клавиатуры. В отличие от других элементов, позволяющих вводить значения параметров в поле ввода, здесь вводимые данные маскируются точками. Чаще всего элементы Пароль в окне создаются путем перетаскивания 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.

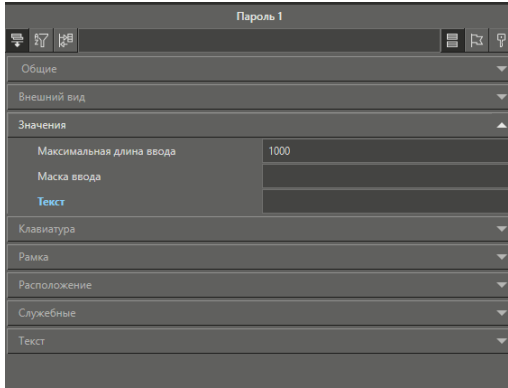
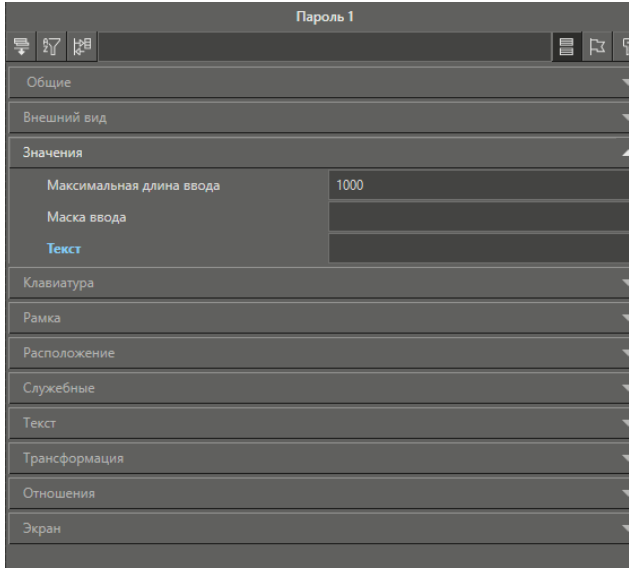
В палитре элемент имеет вид:



Вид элемента после добавления в рабочую область редактора НМІ:



Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
	

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.
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 HMI, 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.

Описание других свойств элемента смотрите в разделе HMI.Свойства

Внешний вид элемента не отличается в HMI v1 и в HMI v2, и порядок работы с элементом одинаков.

### 8.3.1.4. Инкремент

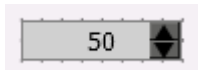
Данный элемент предназначен для установки в клиенте визуализации числовых значений параметров простых типов при помощи реальной или виртуальной клавиатур, а также при помощи нажатия левой кнопкой мыши на стрелки вверх и вниз или на символы + и -. Чаще всего элементы Инкремент в окне создаются путем перетаскивания аналогового параметра в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.

В палитре редактора HMI элемент Инкремент находится в категории Диалог.

В палитре элемент имеет вид:

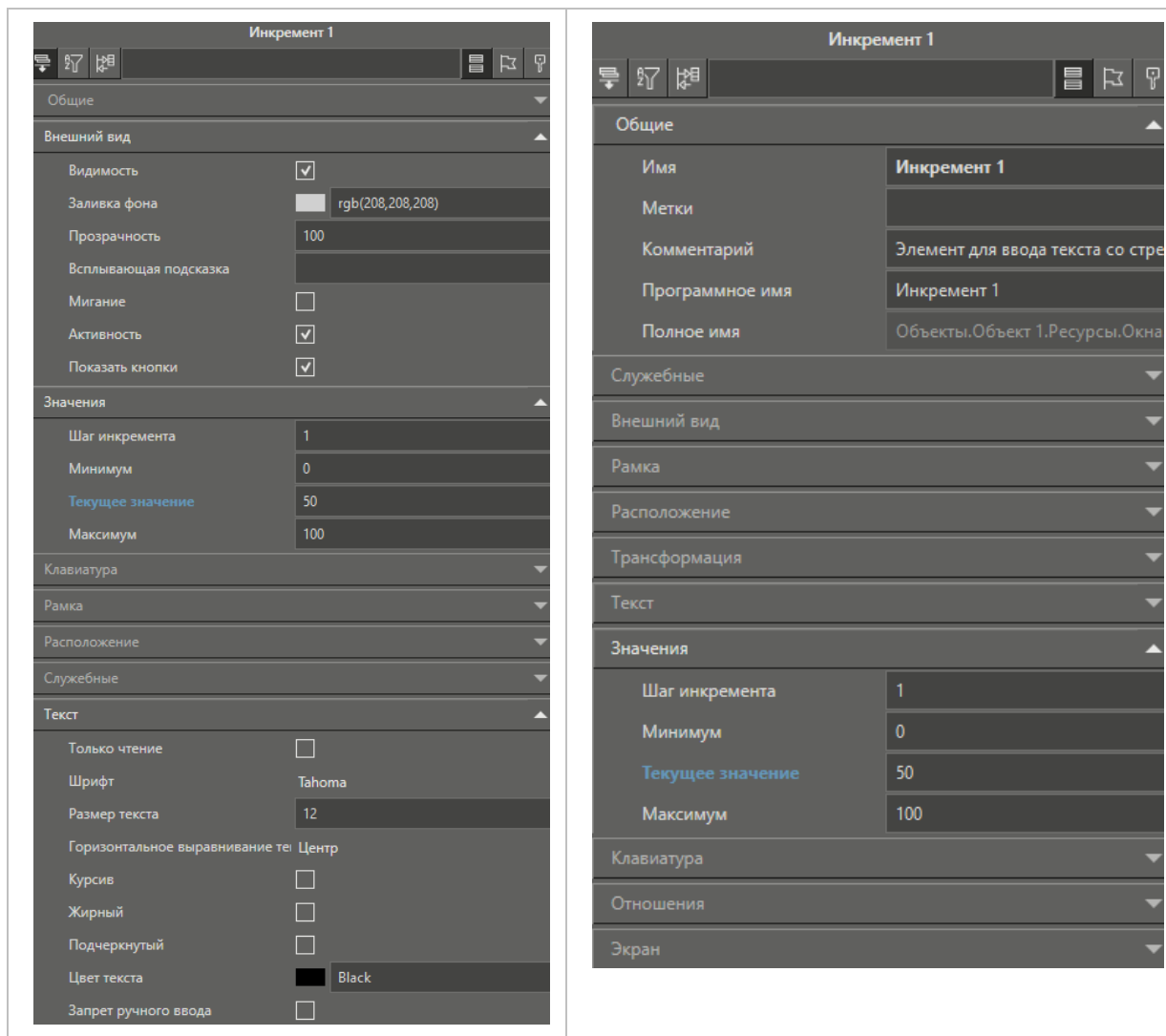


Вид элемента после добавления в рабочую область редактора HMI:



Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
---------------	--------------



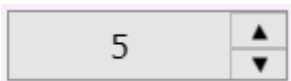
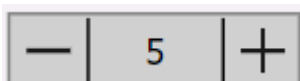
### Описание основных свойств графического элемента Инкремент:

Название	Описание
<b>Категория Внешний вид</b>	
Показать кнопки	Определяет будут ли отображаться кнопки с символами + и - в клиенте визуализации. Если кнопки не отображаются, то ввод значений может осуществляться только при помощи клавиатуры (только для HMI v2).
<b>Категория Значения</b>	
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.
Категория Текст	
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
	

### 8.3.1.5. Кнопка

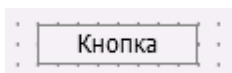
Данный элемент предназначен для выполнения сконфигурированных действий (см. Действия по условию и Действия по событию) либо для изменения значения дискретного параметра, связанного со свойством Нажата. Чаще всего элементы Кнопка создаются в окне путем перетаскивания дискретного параметра, объекта или 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.

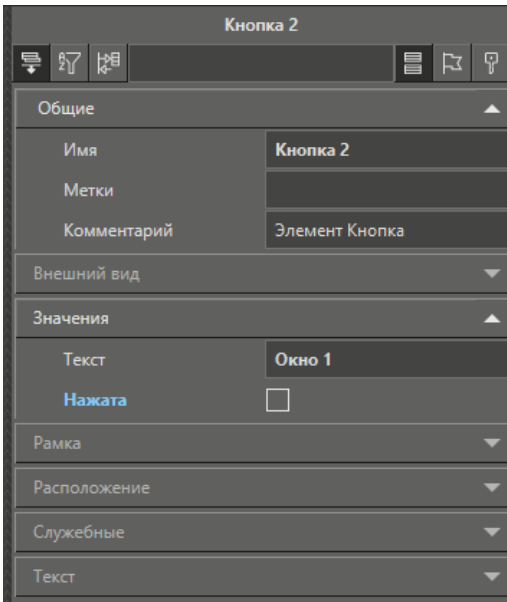
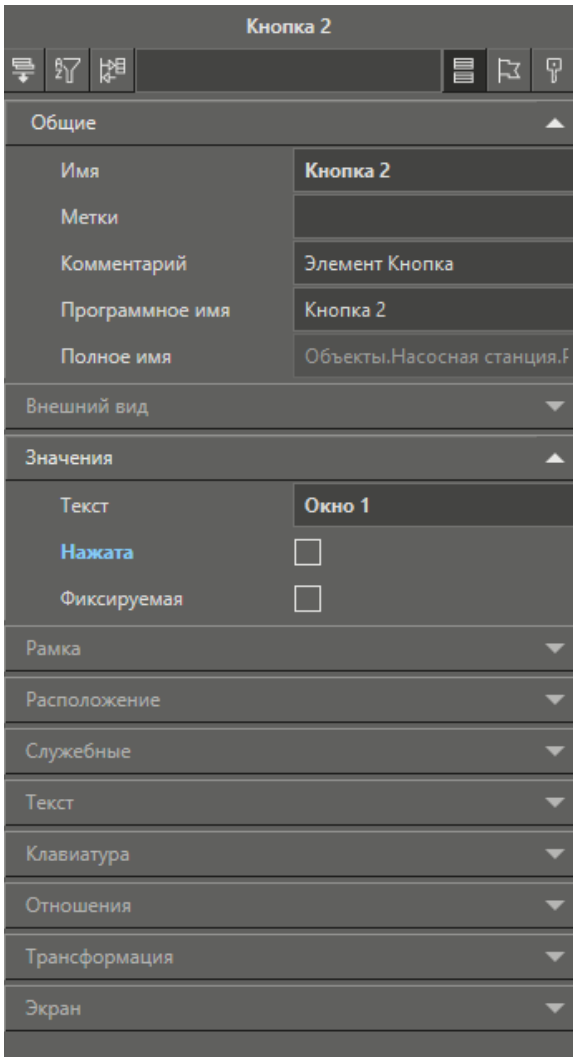
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:




Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
	

Описание основных свойств графического элемента Кнопка:

Название	Описание
----------	----------

Категория Значения	
Текст	Задается текст надписи на кнопке.
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 НМІ элемент Кнопка с фиксацией находится в категории Диалог.

Вид элемента в палитре:

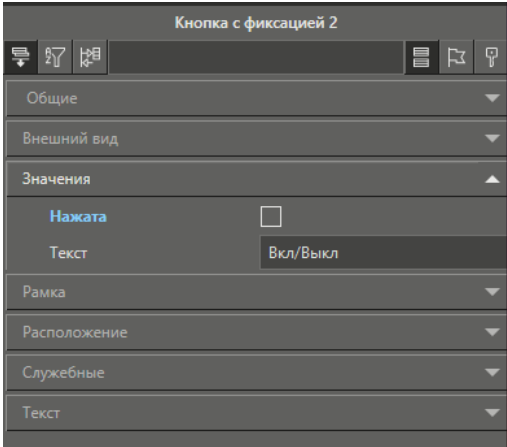
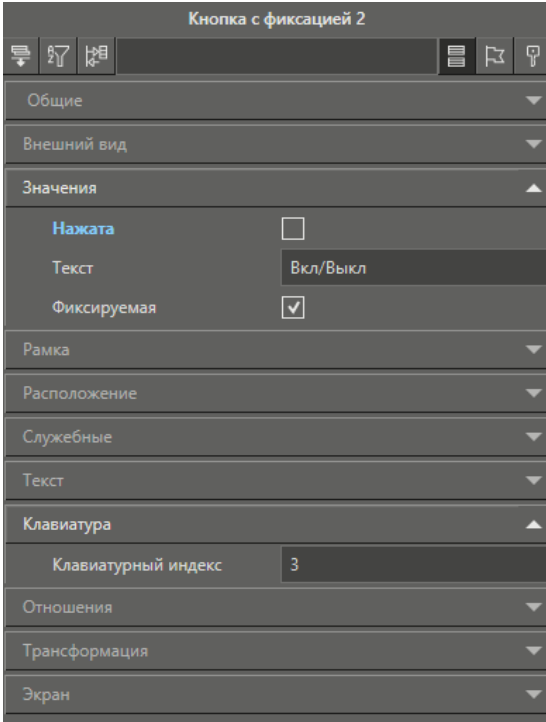


Вид элемента после добавления в рабочую область редактора НМІ:






Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
	

Описание основных свойств графического элемента Кнопка с фиксацией:

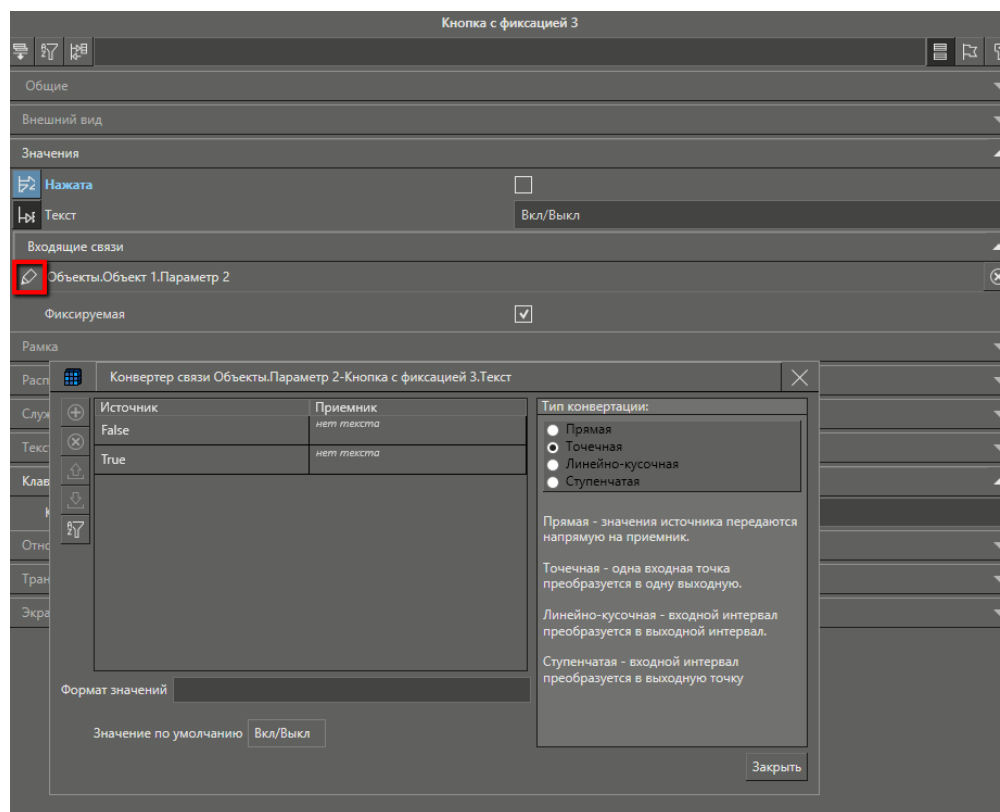
Название	Описание
Категория Значения	
Текст	Задается текст надписи на кнопке с фиксацией. Рекомендации по настройке читайте ниже в этом разделе.
Нажата	Основное свойство кнопки с фиксацией, неразрывно связанное с её положением. Может иметь входящую и исходящую связи. Если элемент был добавлен в окно 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.

Фиксируемая	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.
-------------	--

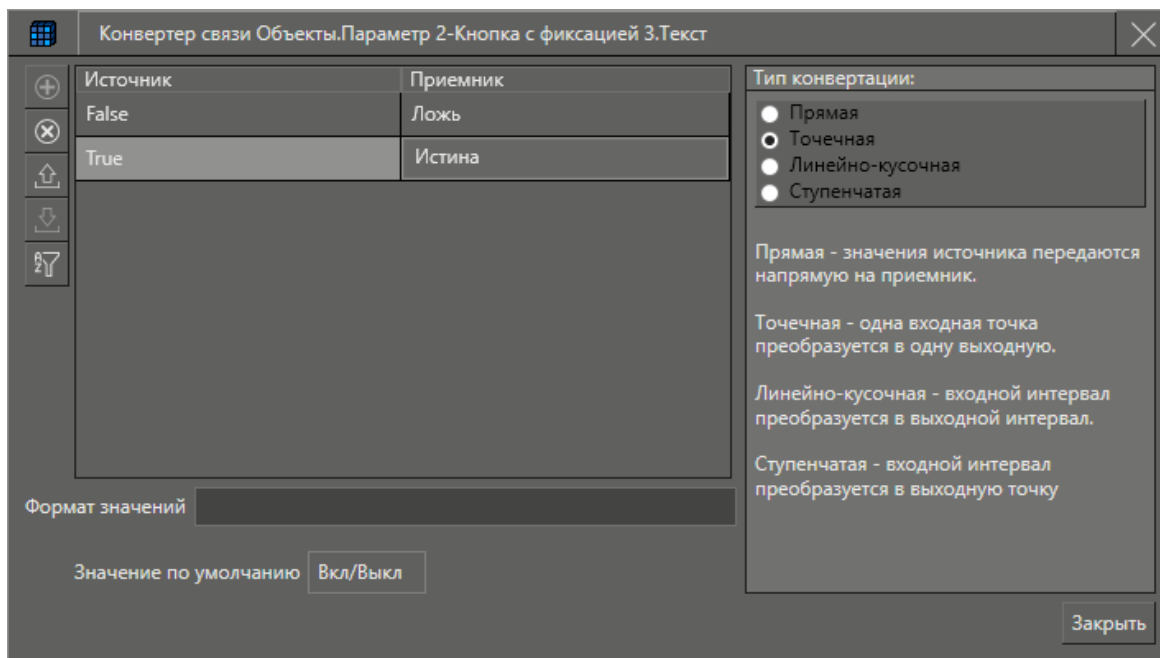
Описание других свойств элемента смотрите в разделе HMI.Свойства

### Настройка свойства Текст

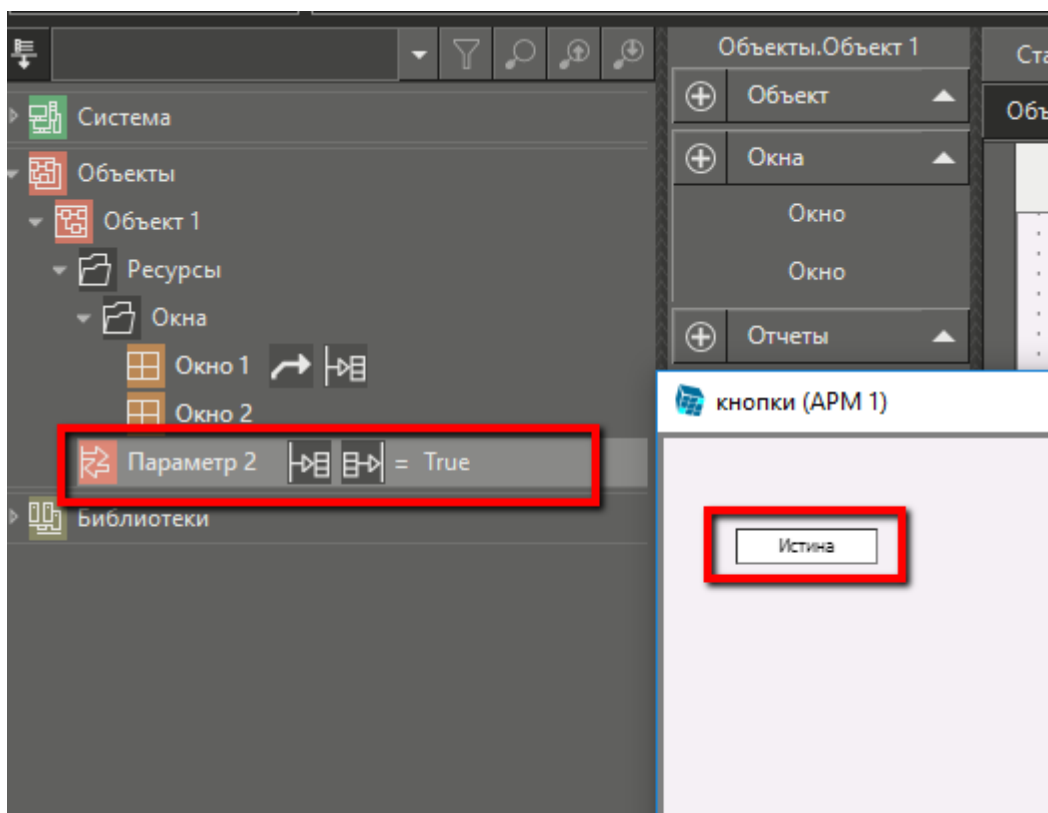
Для того чтобы текст на кнопке отображал состояние параметра, необходимо перетащить параметр на свойство текст и открыть конвертер образовавшейся связи (для точечной передаточной характеристики этой связи поля значений источника будут уже заполнены как TRUE и False):



Затем следует заполнить соответствующие поля приемника и нажать кнопку Заккрыть:



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 HMI элемент Радиокнопка находится в категории Диалог.

Вид элемента в палитре:

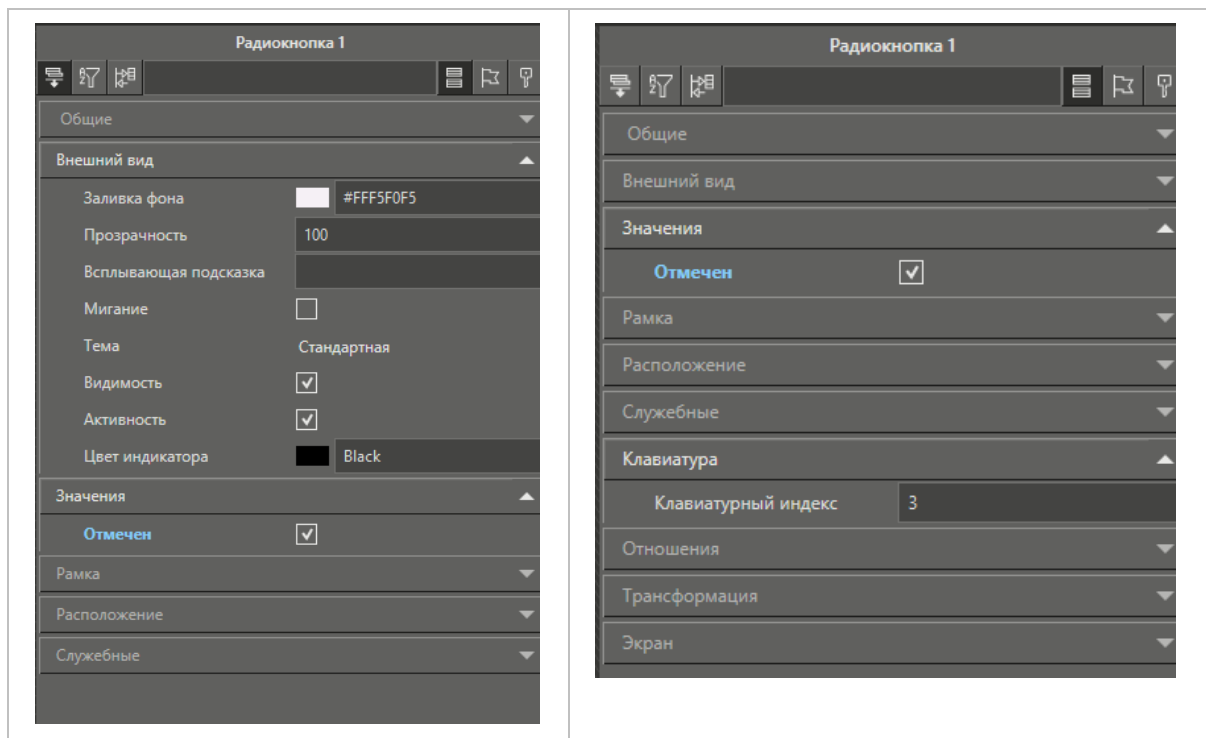


Вид элемента после добавления в рабочую область редактора HMI:



Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
---------------	--------------



#### Описание основных свойств графического элемента Радиокнопка:

Название	Описание
Категория Значения	
Отмечен	<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 перетащен 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> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p><b>Важно!</b> Допускается чтобы только у одной радиокнопки в окне или панели это свойство имело значение TRUE по умолчанию.</p> </div>

Описание других свойств элемента смотрите в разделе HMI.Свойства

Внешний вид элемента не отличается в HMI v1 и в HMI 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 HMI элемент Флаг находится в категории Диалог.

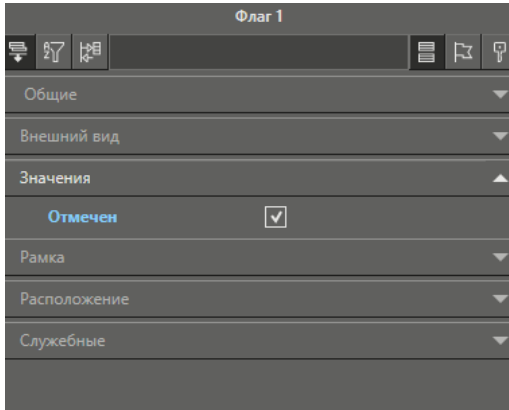
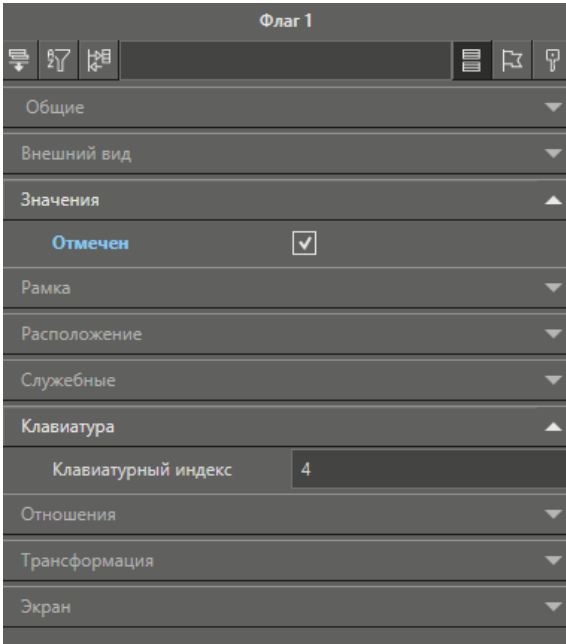
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:



Вид панели свойств элемента при нажатой и отжатой кнопке :

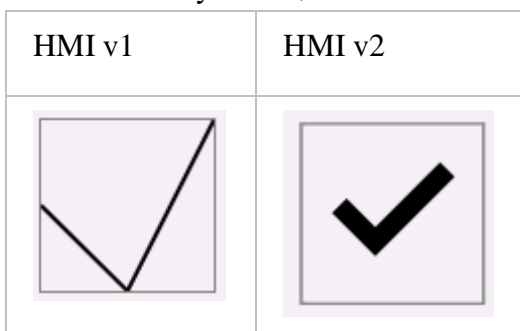
Простой режим	Полный режим
	

Описание основных свойств графического элемента 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 HMI, 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.

Описание других свойств элемента смотрите в разделе HMI.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 HMI элемент Выпадающий список находится в категории Диалог.

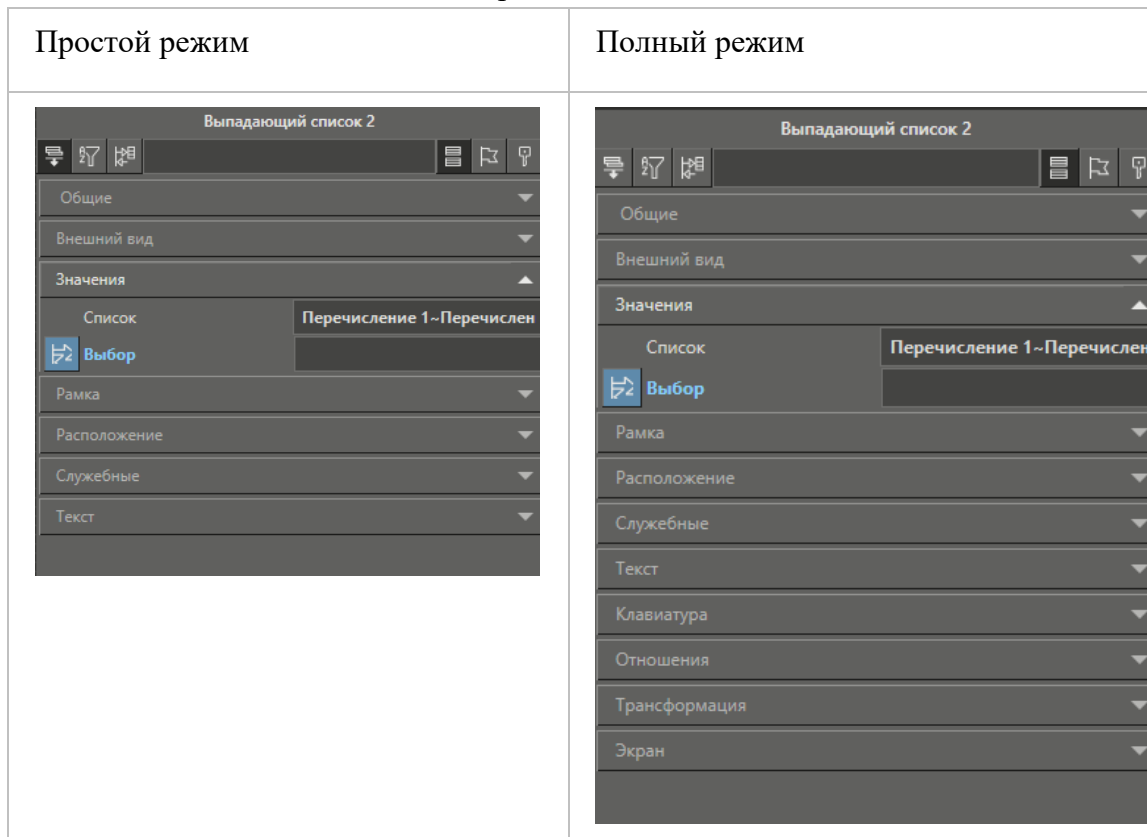
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:



Вид панели свойств элемента при нажатой и отжатой кнопке :



Описание основных свойств графического элемента 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 <code>~</code> . При установке входящей связи список строк может быть изменен в результате работы среды 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 <code>STRING</code> 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.
--	--

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.

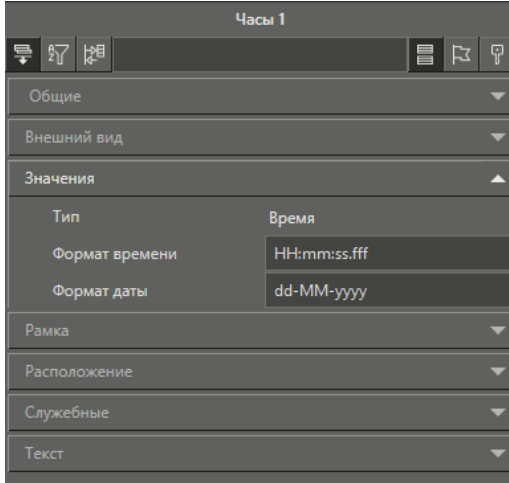
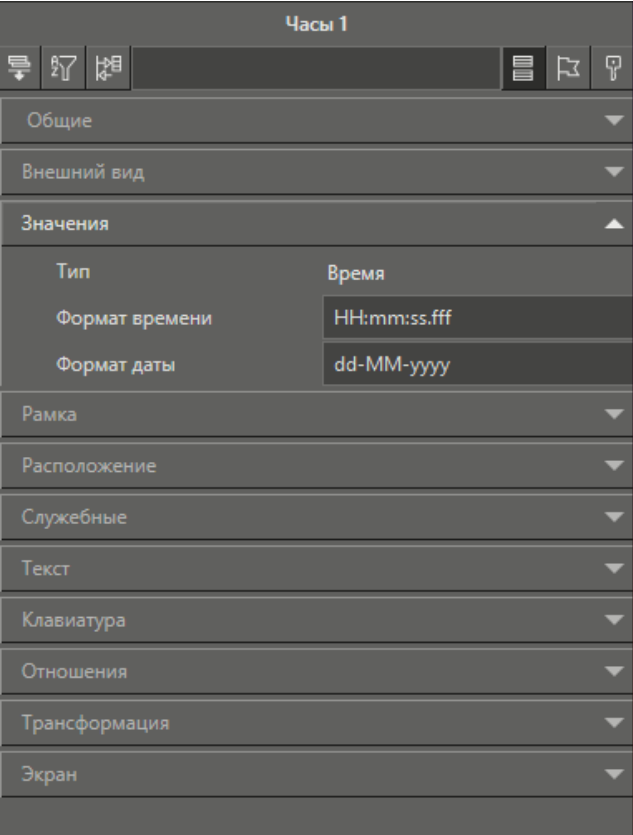
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора НМІ:



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"> <li>• дата;</li> <li>• дата и время;</li> <li>• время и дата;</li> <li>• время.</li> </ul>
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

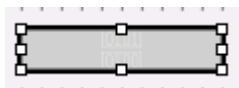
Данный элемент предназначен для отображения и/или установки значений параметров типа DT. Чаще всего элементы ДатаВремя в окне создаются путем перетаскивания параметра типа DT в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.

В палитре редактора НМІ элемент ДатаВремя находится в категории Диалог.

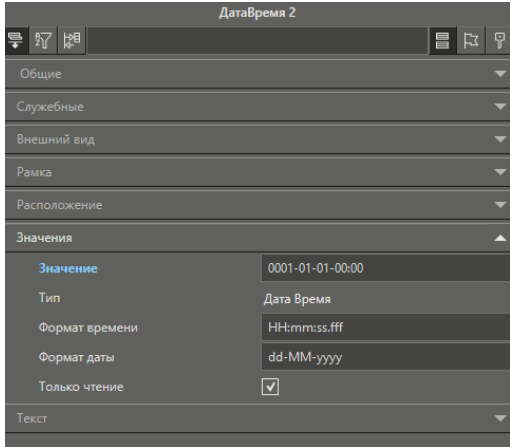
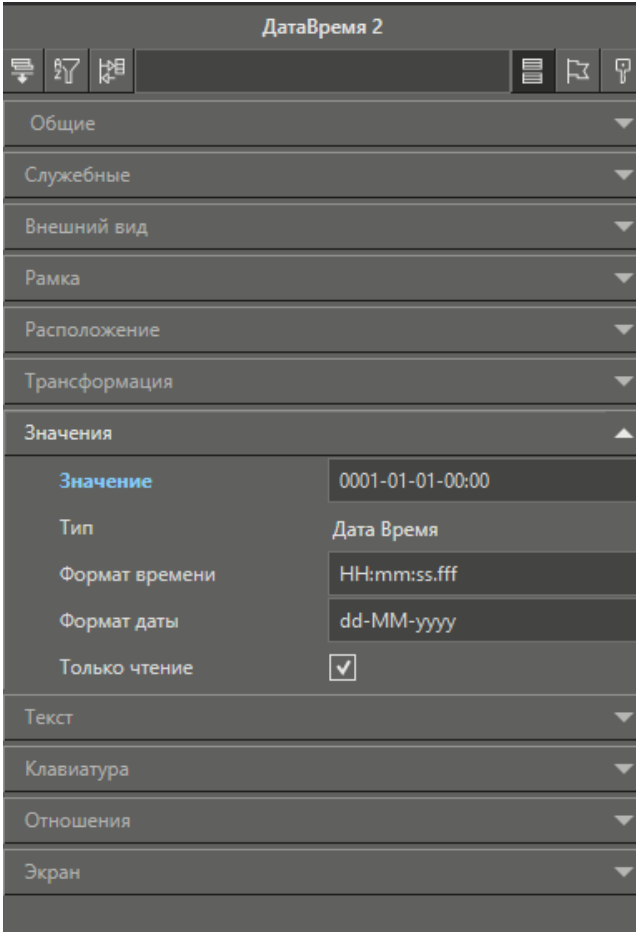
Вид элемента в палитре:



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	<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 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</p>

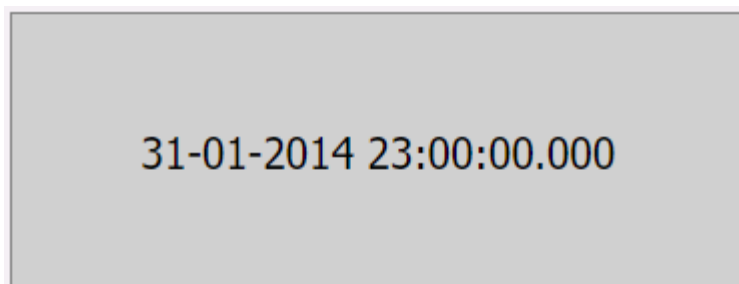
	свойство будет содержать дополнительные строки, в которых будет находиться информация о связях.
Тип	<p>Задаёт, что именно и в каком порядке необходимо отображать. Выпадающий список содержит следующие варианты:</p> <ul style="list-style-type: none"> <li>• дата;</li> <li>• дата и время;</li> <li>• время и дата;</li> <li>• время.</li> </ul>
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:



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:

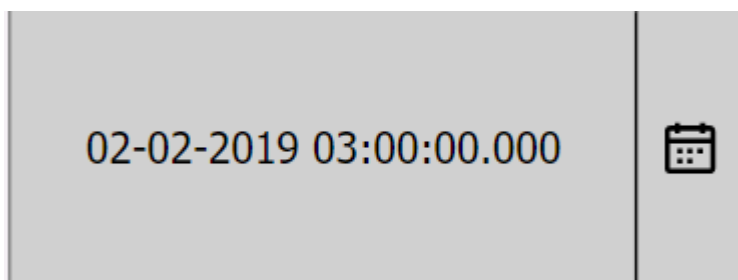
- Навести мышь на отдельный элемент даты или времени и нажать левой кнопкой мыши на нужное значение;
- Навести мышь на отдельный элемент даты или времени и вращением колеса мыши перемещать его до нужного значения;
- Навести мышь на отдельный элемент даты или времени и перемещать его левой кнопкой мыши 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:

15:54:31.892			
13	52	37	881
14	53	38	882
15	54	39	883
16	55	40	884
17	56	41	885
		ОТМЕНА	УСТАНОВИТЬ

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.	Миллисек.
▲	▲	▲	▲
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

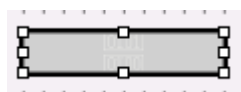
Данный элемент предназначен для отображения и/или установки значений параметров типа TOD. Чаще всего элементы Время в окне создаются путем перетаскивания параметра типа TOD в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.

В палитре редактора НМІ элемент Время находится в категории Диалог.

Вид элемента в палитре:

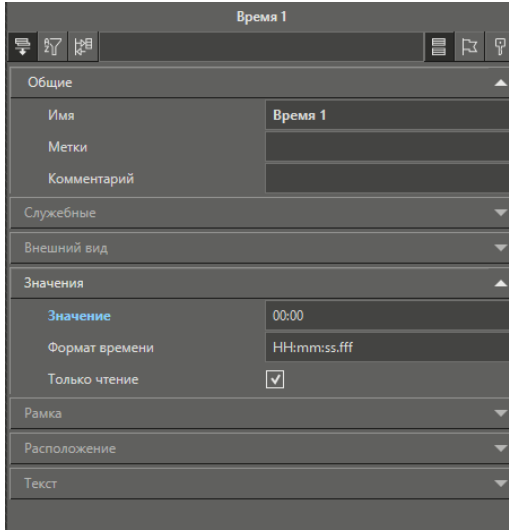
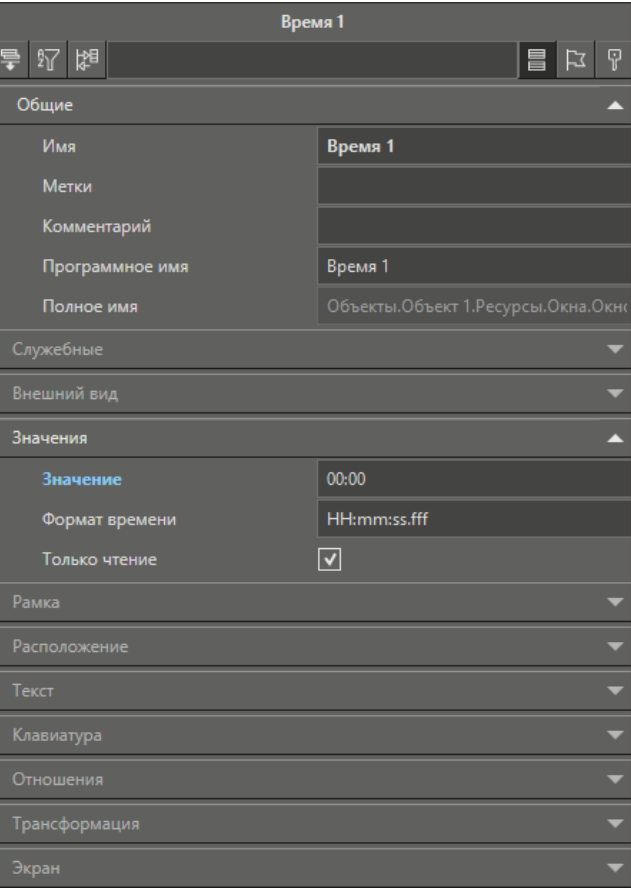


Вид элемента после добавления в рабочую область редактора НМІ:



Вид панели свойств элемента при нажатой и отжатой кнопке :



Простой режим	Полный режим
	

Описание основных свойств графического элемента **Время**:

Название	Описание
Категория Значения	
Значение	<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 HMI, то связи необходимо установить вручную. Если параметр типа TOD был перетащен в окно правой кнопкой мыши, и в контекстном меню был выбран пункт <b>Время</b>, то данное поле заполнится автоматически. После установки связи данное свойство будет содержать дополнительные строки, в которых будет находиться информация о связях.</p>

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:

00.00:00:00

Для того чтобы отобразить инструменты задания времени, необходимо нажать левой кнопкой мыши на элемент:

15:54:31.892

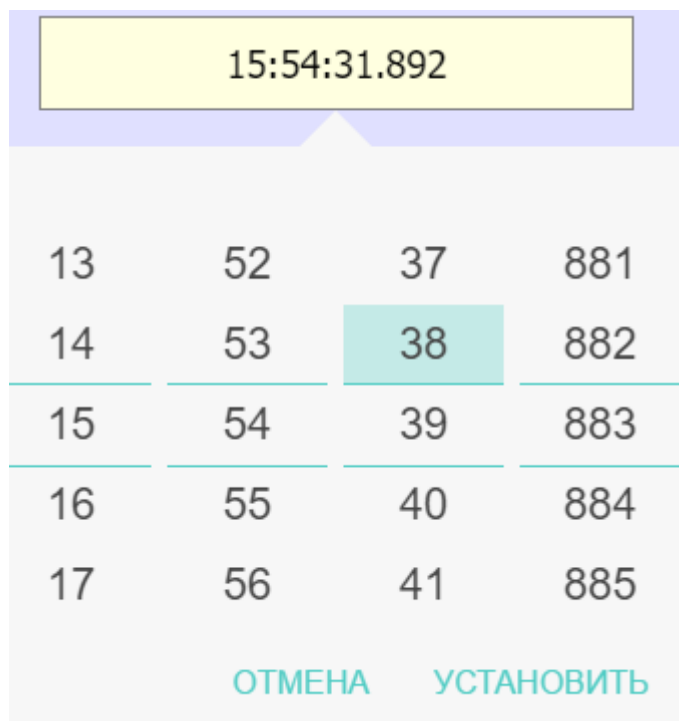
13	52	29	890
14	53	30	891
15	54	31	892
16	55	32	893
17	56	33	894

ОТМЕНА
УСТАНОВИТЬ

Для установки отдельных элементов времени с помощью показанных выше инструментов можно использовать различные методы:

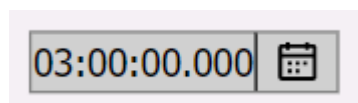
- Навести мышь на отдельный элемент даты или времени и нажать левой кнопкой мыши на нужное значение;
- Навести мышь на отдельный элемент даты или времени и вращением колеса мыши перемещать его до нужного значения;
- Навести мышь на отдельный элемент даты или времени и перемещать его левой кнопкой мыши 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:



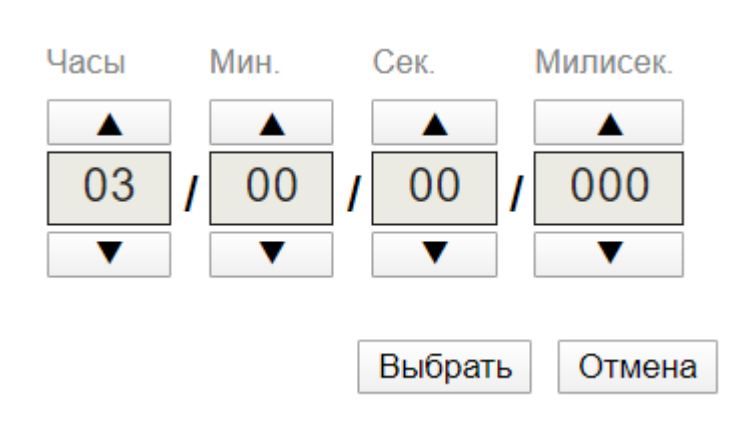
Тип транслятора НМІ v2

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

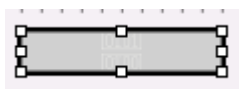
Данный элемент предназначен для отображения и/или установки значений типа TIME. Чаще всего элементы Интервал в окне создаются путем перетаскивания параметра типа TIME в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.

В палитре редактора НМІ элемент Интервал находится в категории Диалог.

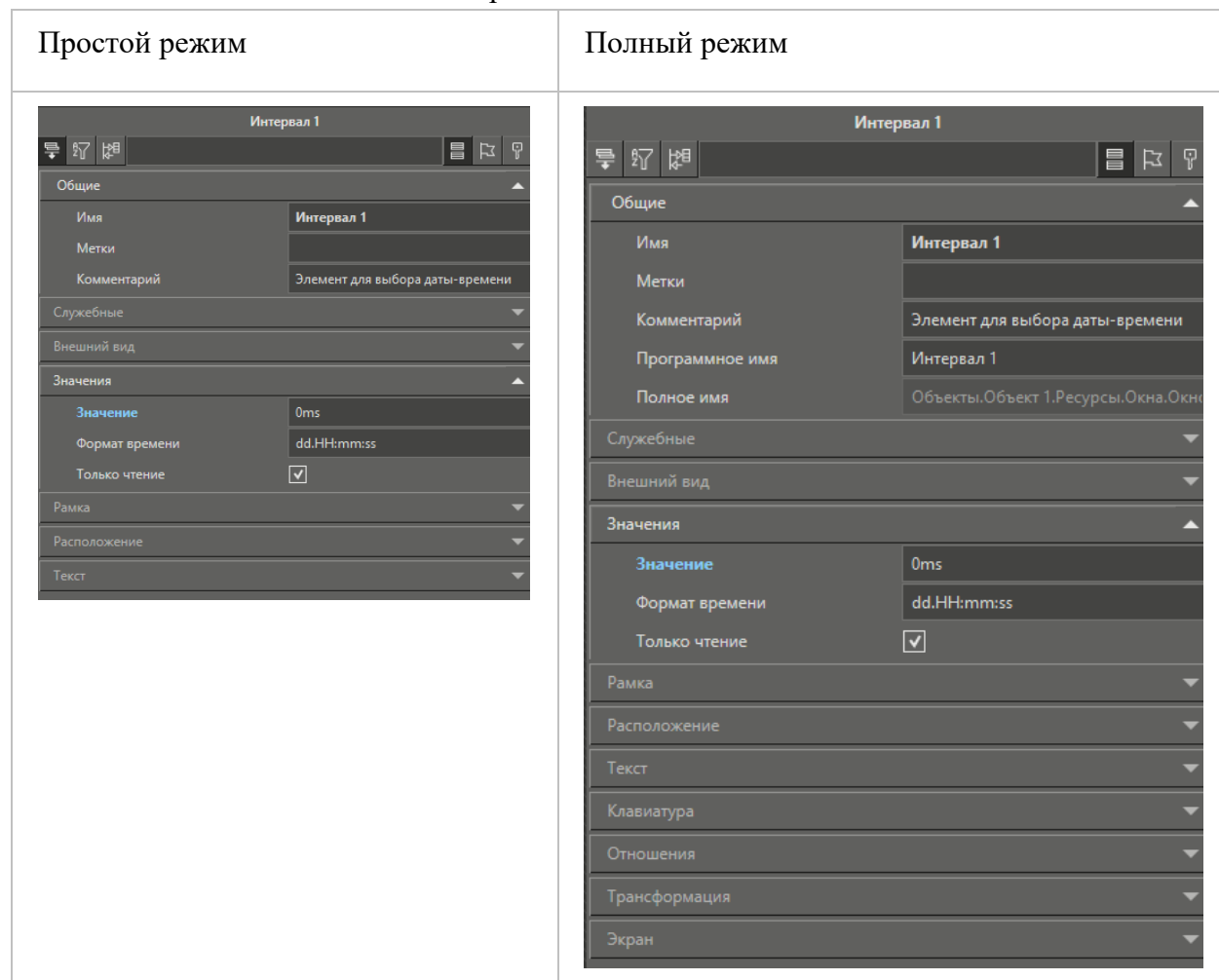
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора НМІ:



Вид панели свойств элемента при нажатой и отжатой кнопке :



Описание основных свойств графического элемента Интервал:

Название	Описание
Категория Значения	
Значение	<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</p>

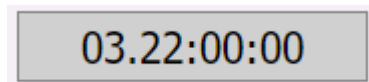
	свойство будет содержать дополнительные строки, в которых будет находиться информация о связях.
Формат времени	Определяет формат времени. По умолчанию свойство принимает значение HH:mm:ss.fff, где: HH – это часы, mm – минуты, ss – секунды, fff – миллисекунды. Например, если формат времени будет иметь 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:



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	

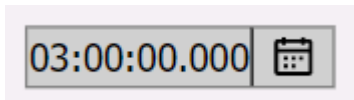
Для установки отдельных элементов интервала с помощью показанных выше инструментов можно использовать различные методы:

- 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.

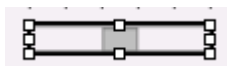
В палитре редактора НМІ элемент Горизонтальный ползунок находится в категории Диалог.


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	

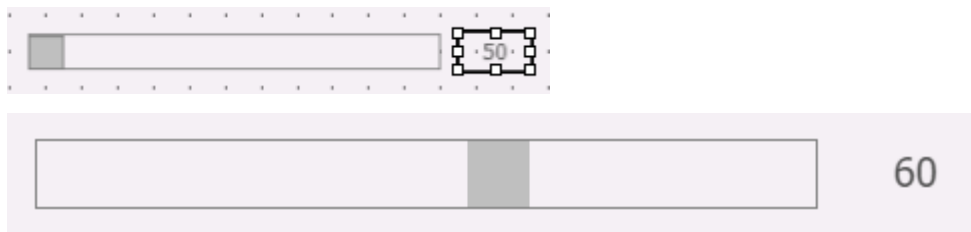
Цвет заливки	Определяет цвет закрашенной области ползунка. При наведении мыши на свойство появляется кнопка  , при нажатии на которую открывается диалоговое окно Выбор цвета.
Ориентация	Имеет два значения: Горизонтально и Вертикально. Элементы Горизонтальный ползунок и Вертикальный ползунок отличаются 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).
Радиус скругления ползунка	Задает 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 НМІ.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

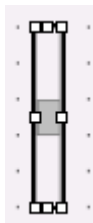
Графический элемент Вертикальный ползунок аналогичен графическому элементу Горизонтальный ползунок, у которого свойство Ориентация имеет значение Вертикально. Чаще всего элементы Вертикальный ползунок в окне создаются путем перетаскивания аналогового параметра в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.

В палитре редактора НМІ 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

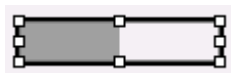
Данный элемент предназначен для отображения текущего значения аналогового параметра в виде показаний линейного индикатора. Чаще всего элементы Горизонтальный прогресс в окне создаются путем перетаскивания аналогового параметра в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.

В палитре редактора HMI элемент Горизонтальный прогресс находится в категории Диалог.

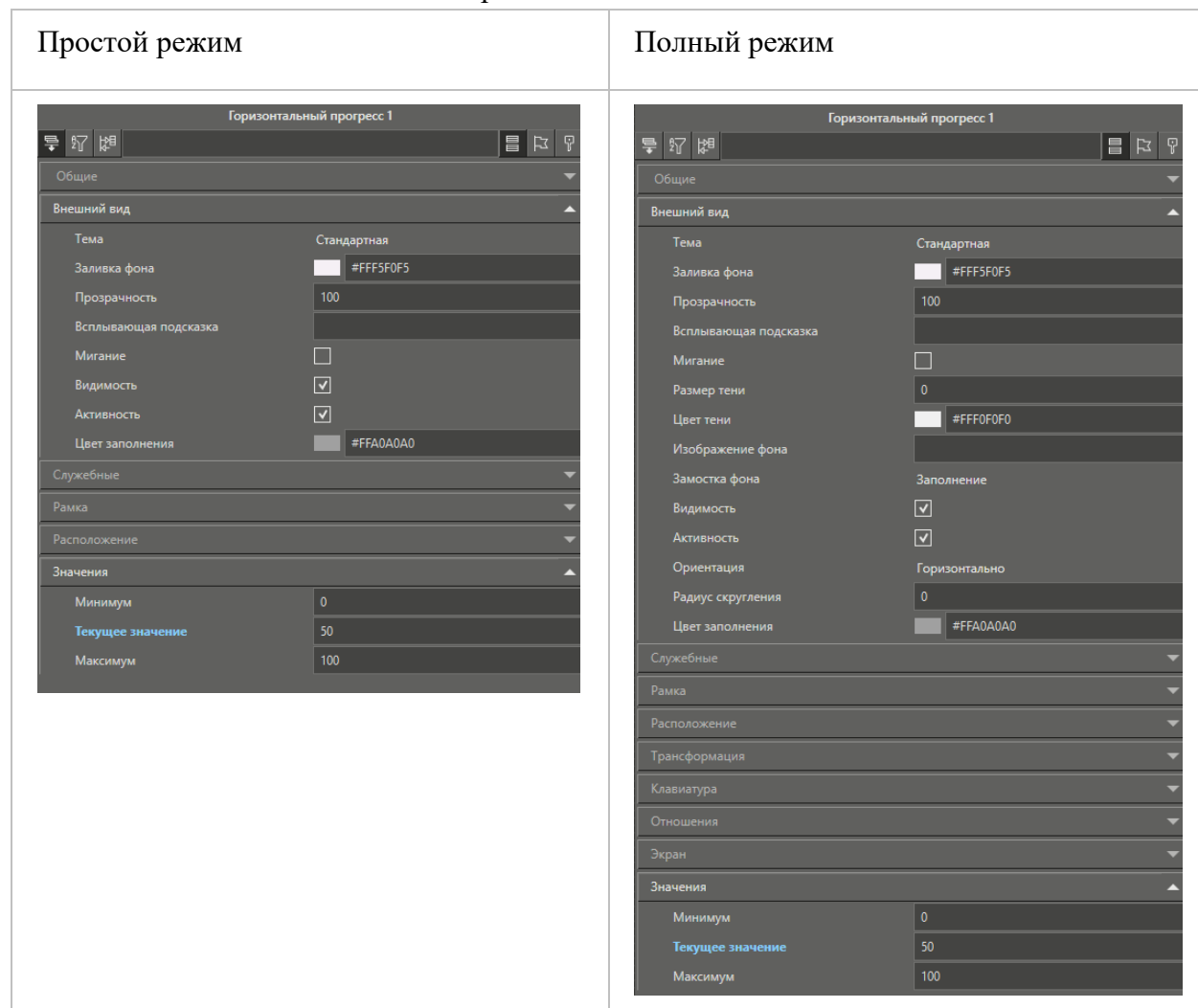
Вид элемента в палитре:



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	Имеет два значения: Горизонтально и Вертикально. Элементы Горизонтальный прогресс и Вертикальный прогресс отличаются только значением этого свойства, заданным для них по умолчанию. Цвет столбика элемента задается при помощи свойства цвет заполнения.

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

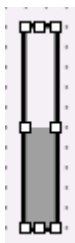
Графический элемент Вертикальный прогресс аналогичен графическому элементу горизонтальный прогресс, у которого свойство ориентация имеет значение вертикально. Чаще всего элементы Вертикальный прогресс в окне создаются путем перетаскивания аналогового параметра в окно правой кнопкой мыши. В этом случае 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 НМІ:



### 8.3.1.18. Кнопка выбора цвета

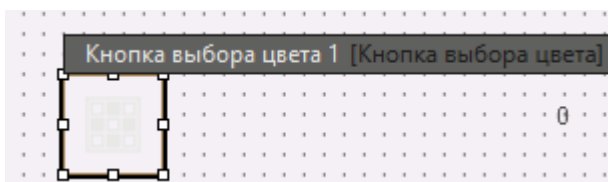
Данный элемент предназначен для выбора цвета в клиенте визуализации. Если параметр проекта будет иметь входящую связь со свойством Значение кнопки и исходящую связь с каким-либо свойством графического элемента, отвечающего за цвет, то появится возможность менять цвет элементов в клиенте визуализации, например, цвет пера графика.

В палитре редактора НМІ элемент Кнопка выбора цвета находится в категории Диалог.

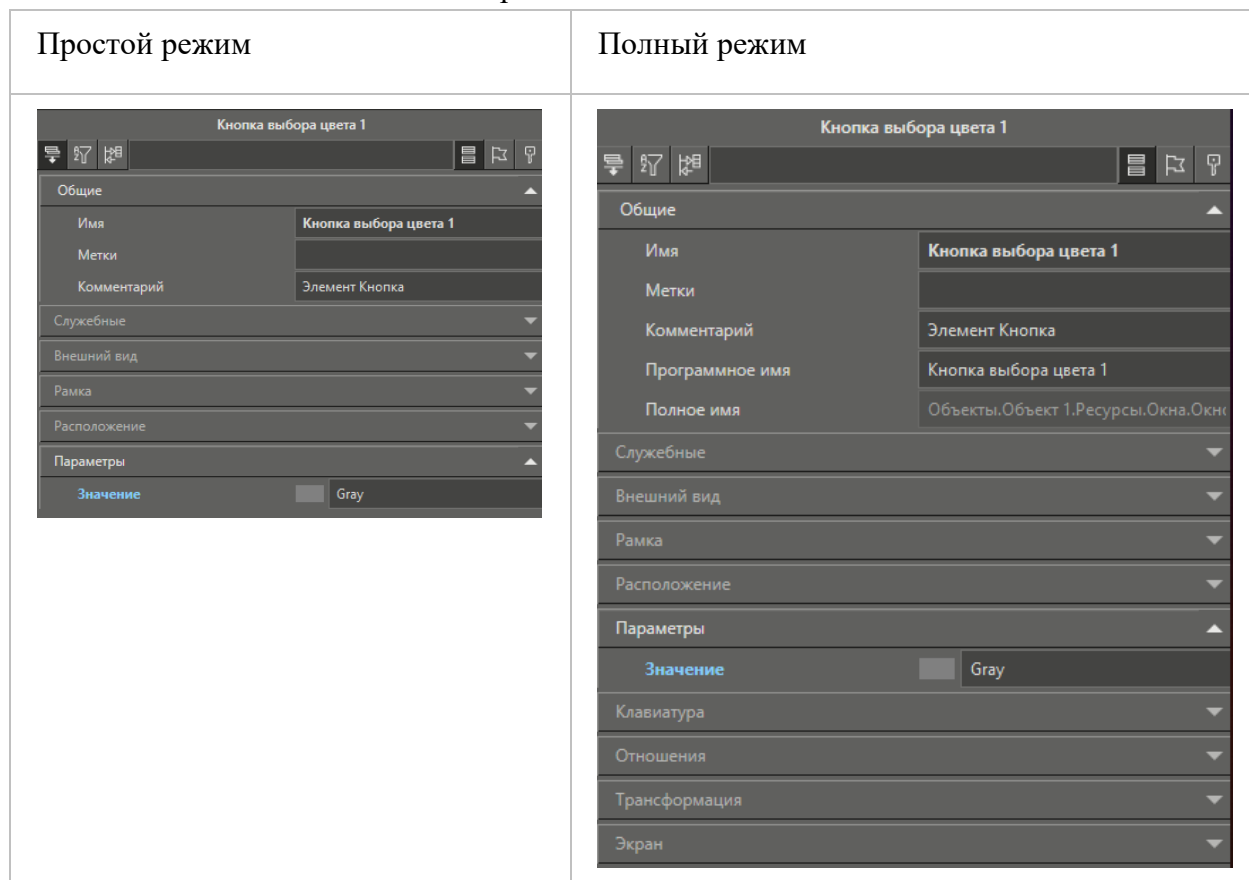
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора НМІ:



Вид панели свойств элемента при нажатой и отжатой кнопке :



Описание основных свойств графического элемента Кнопка выбора цвета:

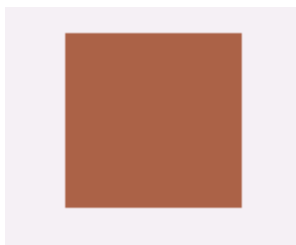
Название	Описание
Категория Параметры	
Значения	Основное свойство элемента, которое необходимо связать с параметром типа STRING. В этом свойстве будет указан цвет (RGB-значение цвета), выбранный в клиенте визуализации. Может иметь входящую и исходящую связи. Если кнопка связана с параметром окна, не имеющим связи с параметрами объектов и узлов, то допускается выбрать в разных клиентах разное цветовое оформление.

Описание других свойств элемента смотрите в разделе НМІ.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 разным.

Окно выбора цвета в HMI v1

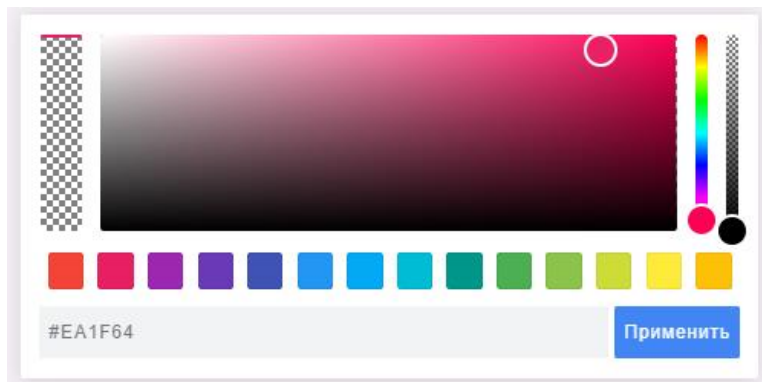
В HMI v1 окно имеет вид:



Изменения вступят в силу незамедлительно после нажатия левой кнопкой мыши на выбранный цвет.

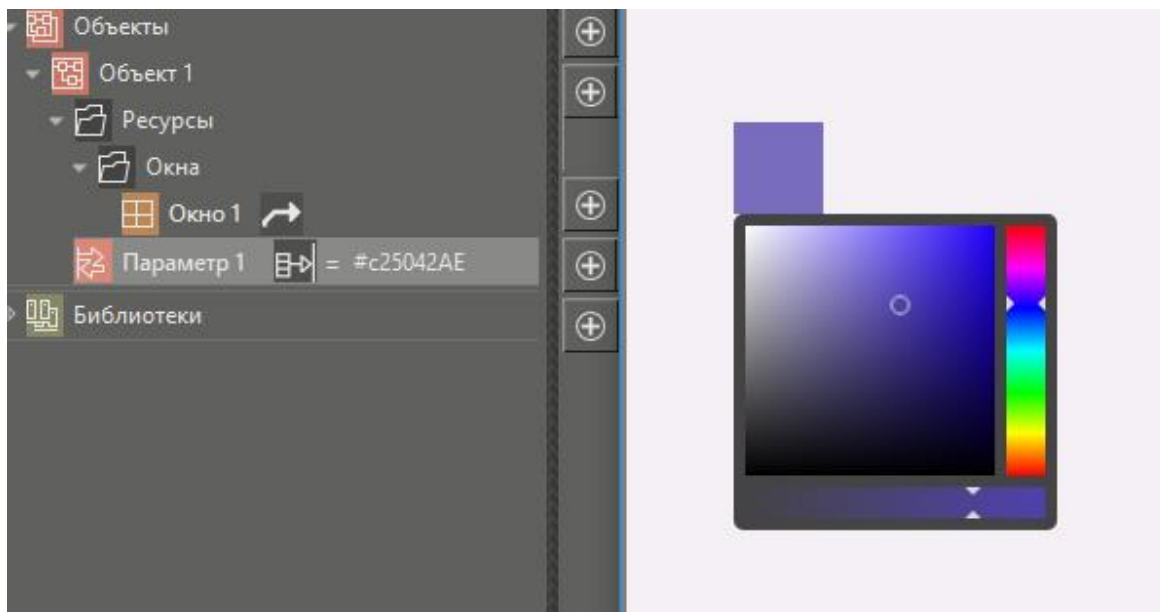
Окно выбора цвета в HMI v2

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

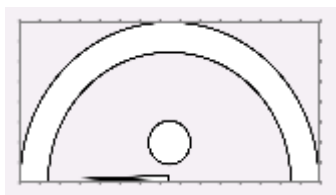
Данный элемент предназначен для отображения значения аналогового параметра в виде показаний стрелочного индикатора. Чаще всего элементы Стрелочный прибор в окне создаются путем перетаскивания аналогового параметра в окно правой кнопкой мыши. В этом случае все необходимые связи будут установлены автоматически.


В палитре редактора НМІ элемент Стрелочный прибор находится в категории Диалог.

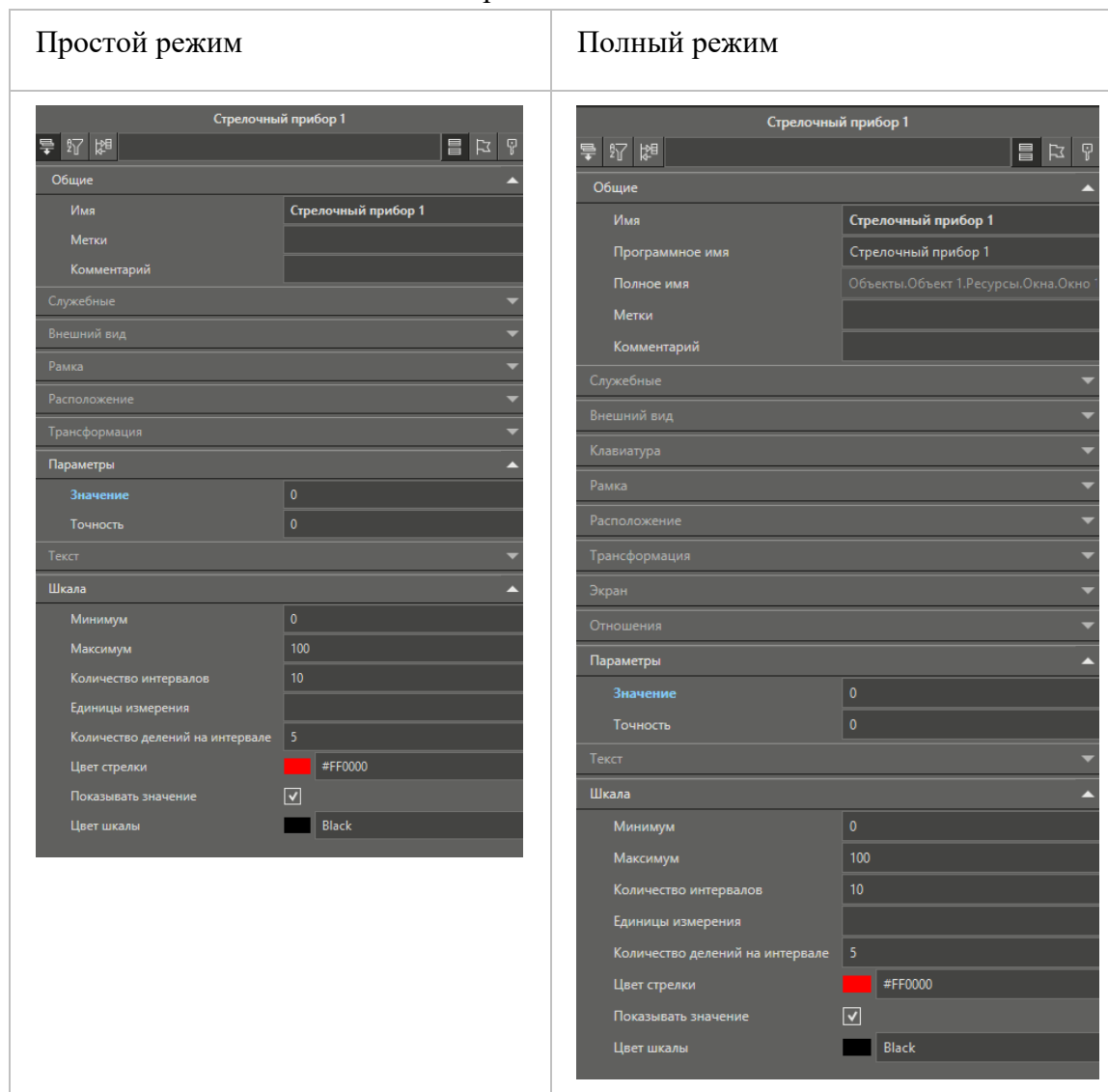
Вид элемента в палитре:



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:

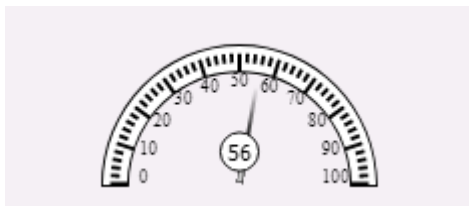
Название	Описание
Категория Шкала	
Максимум	Максимальное значение, которое может показывать элемент.
Минимум	Минимальное значение, которое может показывать элемент.
Количество интервалов	Это свойство имеет тип INT и задает количество больших делений шкалы.

Единицы измерения	Это свойство имеет тип <code>STRING</code> 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 <code>INT</code> 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 <code>BOOL</code> . Если оно имеет значение <code>TRUE</code> , 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.
<b>Категория Параметры</b>	
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 ручную. 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 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

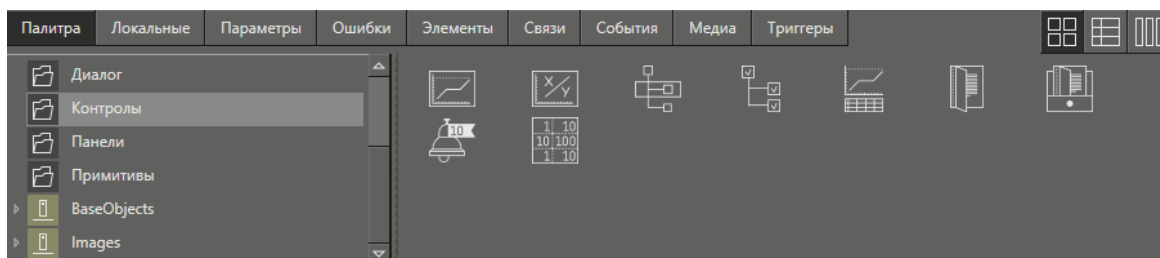
Вид элемента клиенте визуализации:



### 8.3.2. 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

### 8.3.2.1. Тренд

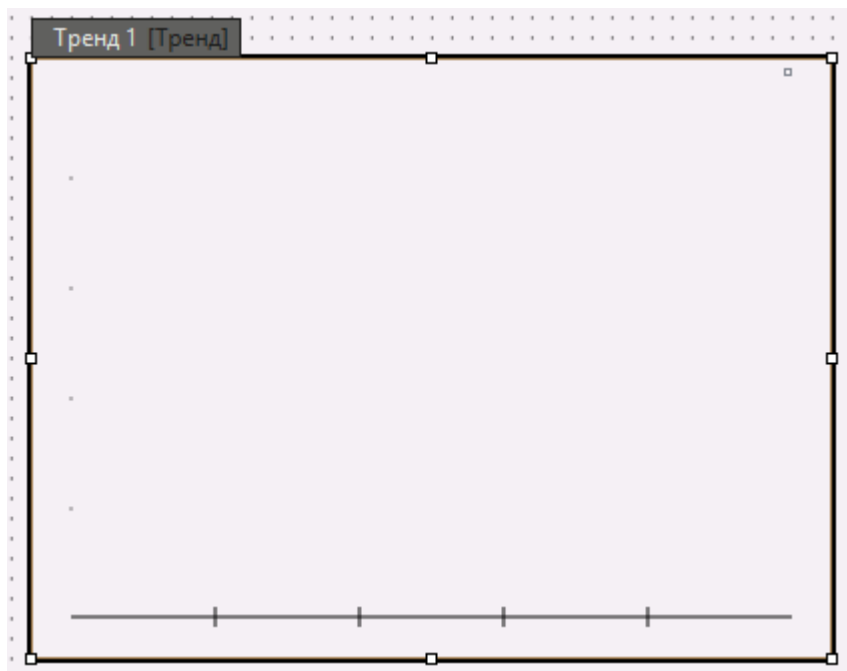
Графический элемент Тренд предназначен для отображения изменения значения параметра во времени. Данный элемент может работать как с архивными, так и с текущими значениями.

В палитре редактора HMI элемент Тренд находится в категории Контролы.

Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:



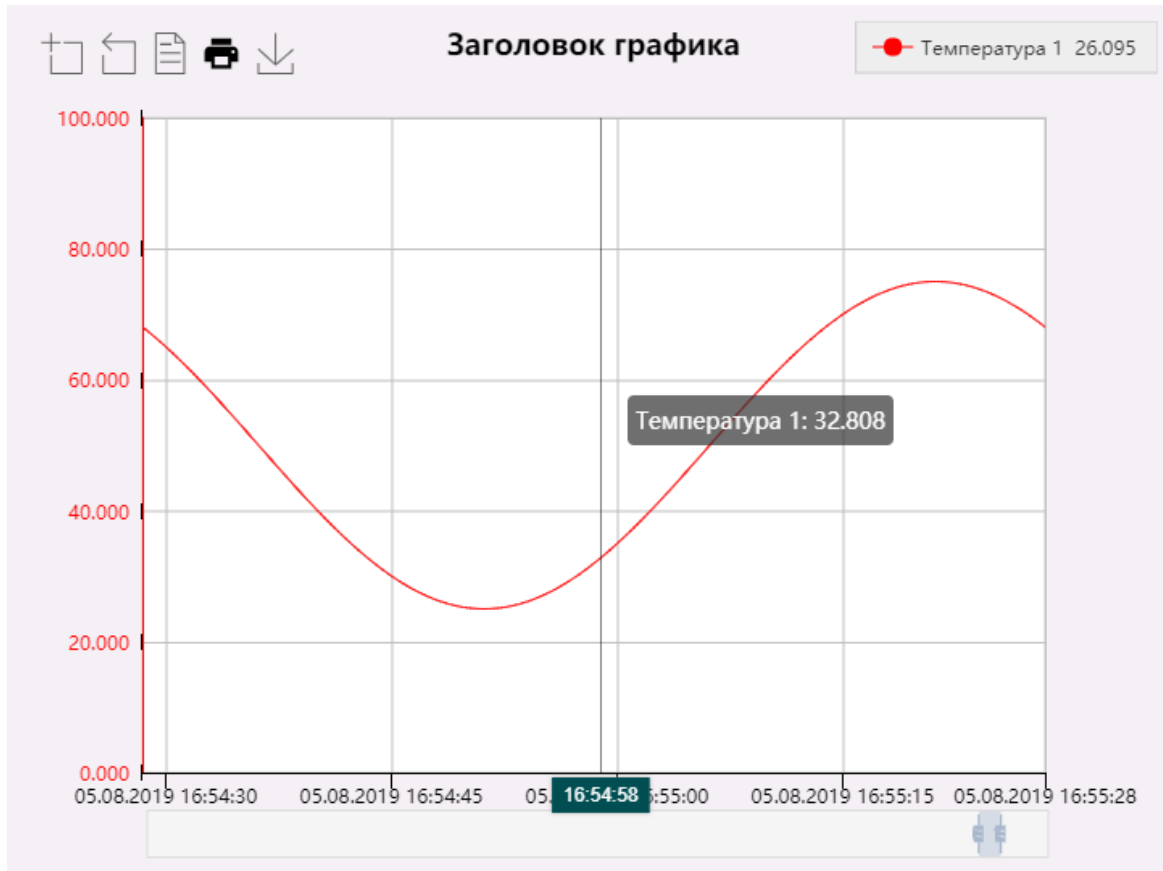
Для того чтобы добавить параметры, которые должны отображаться на тренде, необходимо перетащить параметр из дерева в поле тренда. При этом добавится перо. Кроме того, изменится внешний вид элемента: появится минилегенда в верхнем левом углу элемента. Одновременно с этим в панели свойств добавится группа 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 HMI.

Пример вида тренда, если выбран тип транслятора HMI v1:



Тренд в клиенте визуализации имеет следующие элементы:

- Поле тренда - в котором отображается изменение параметра во времени, поле тренда может содержать сетку.
- Ось X- ось времени.
- Ось Y -ось значений, если на тренд добавлено несколько параметров, то отображается несколько осей Y.
- Курсор - вертикальная черта, появляющаяся при нажатии левой кнопкой мыши на поле тренда, рядом с которой отображается время и значение параметра, соответствующие пересечению этой чертой оси времени и линии тренда.

- Минилегенда - отображаются имена перьев, их обозначение и текущее значение , находится в верхней части тренда.
- Линейка прокрутки - предназначена для изменения отображаемого интервала, находится в нижней части тренда.

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

### 8.3.2.2. Свойства Тренда




Вид панели свойств элемента при нажатой и отжатой кнопке :




Простой режим	Полный режим


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 задает начало выборки из архива.
Конец архива	Используется, если на тренде отображается архивируемый параметр. Данное свойство имеет тип DT и задает конец выборки из архива.
Категория тренд	
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.
Отображать легенду	Данное свойство имеет тип BOOL и может принимать значение TRUE (отображать легенду тренда) или False (не отображать).
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 измене-

	ния интервала можно использовать текстовый ввод, который связан с параметром, передающим данные в свойство тренда Интервал.
Заливка тренда	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.
Категория Курсор выделенной позиции	
Видимость курсора	Это свойство имеет тип BOOL и задает видимость курсора выбранной позиции на тренде.
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.yuuu 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
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.
Цвет оси Y	Определяет цвет 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

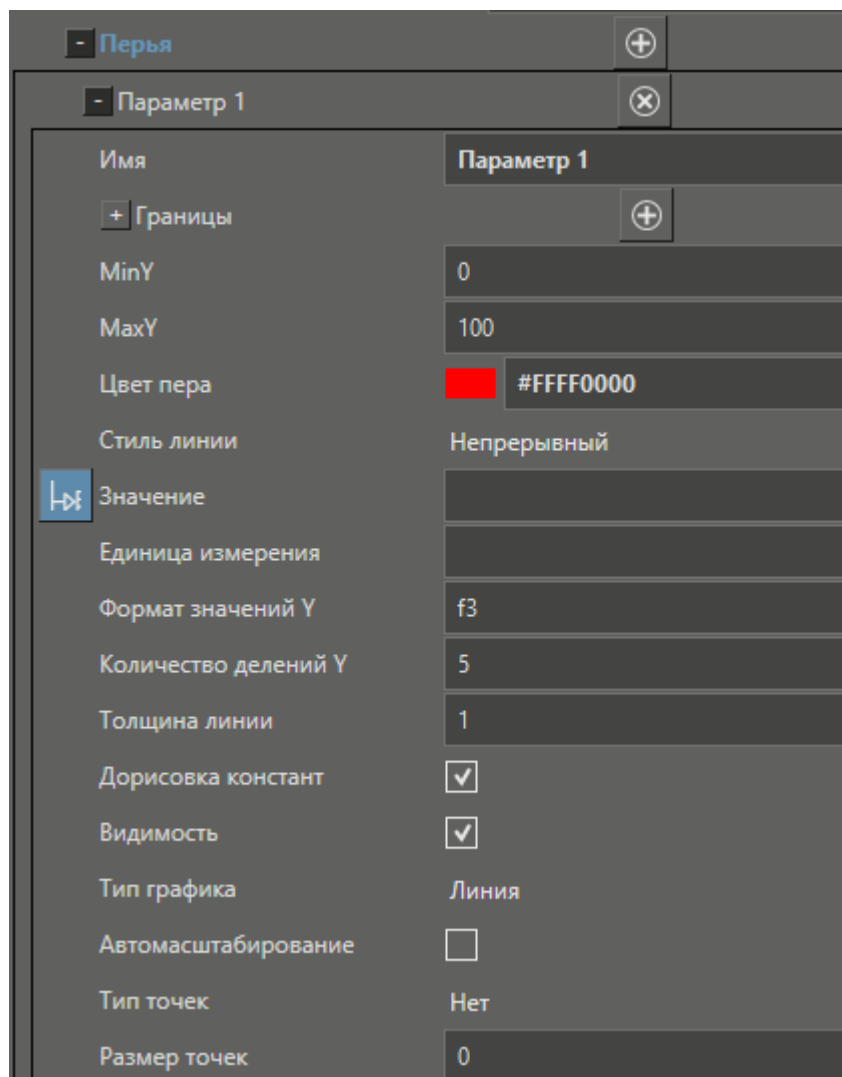
	<p>чем четырех значений. Для этого вычисляется интервал данных по формуле = Интервал времени видимой области графика / (Ширина графика (в px) * Шаг прореживания). Выборка из архива делается с тем условием, что за каждый такой интервал данных из него придет не более четырех значений (первое, последнее, минимальное, максимальное). По умолчанию Шаг прореживания равен 1.</p>
Интервал Листания	<p>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.</p>
Категория Сетка	
Цвет сетки	<p>Определяет цвет линий сетки. При наведении мыши на свойство появляется кнопка: , после нажатия на которую открывается диалоговое окно Выбор цвета.</p>


### Свойства группы Перья


Количество настроек соответствует типу данных TrendPen, который входит в состав библиотеки НМІ. Типы данных.

Набор перьев представляет собой библиотечный тип TrendPens. Эту информацию можно использовать для создания трендов, на которых набор перьев будет меняться 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.

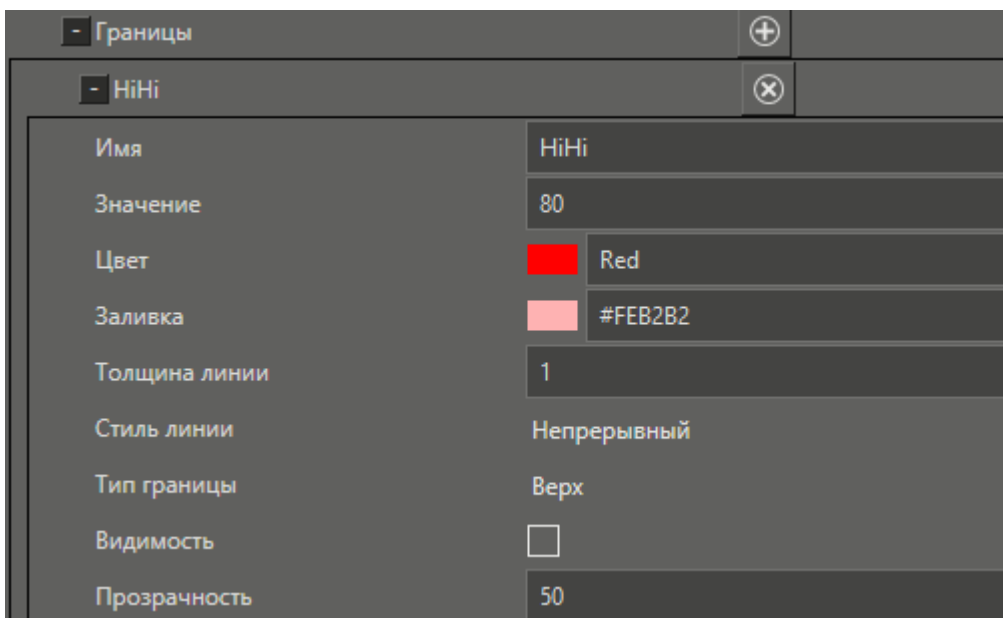
MinY	Минимальное значение оси Y.
Автомасштабирование	Автоматически определяет значения MaxY и MaxX таким образом, чтобы тренд занимал всю площадь поля тренда.
Видимость	Определяет видимость пера в тренде. Данное свойство может быть задинамизировано, т.е. зависеть от других элементов управления, созданных разработчиком проекта, либо от какой-то программы.
Группа Границы	В данной группе настраивается вид тренда в случае если связанный параметр выйдет за указанные пределы.
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 Легенде.
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 свойств 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 тренда	Определяет способ рисования тренда между точками тренда. Возможные варианты: <ul style="list-style-type: none"> <li>• Линия - между двумя точками будет рисоваться прямая линия, т.е. считается, что значение параметра менялось плавно.</li> </ul>

	<ul style="list-style-type: none"> <li>• 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.</li> </ul> <p>Разработчик проекта должен выбрать 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 перетаскать в поле тренда несколько параметров, то у связанных с ними перьев автоматически выберутся разные цвета.
Тип точек	<p>Определяет тип точек, нанесенных на тренд. Точка рисуется в местах излома тренда:</p> <ul style="list-style-type: none"> <li>• Нет</li> <li>• Квадрат</li> <li>• Круг</li> </ul>

	<ul style="list-style-type: none"> <li>• Треугольник</li> <li>• Ромб</li> <li>• Крест</li> </ul>
Размер точек	Определяет размер точек.

## Группа Границы

Вид:



Если у связанного с пером параметра назначена шкала, то настройки будут выставлены автоматически.

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.



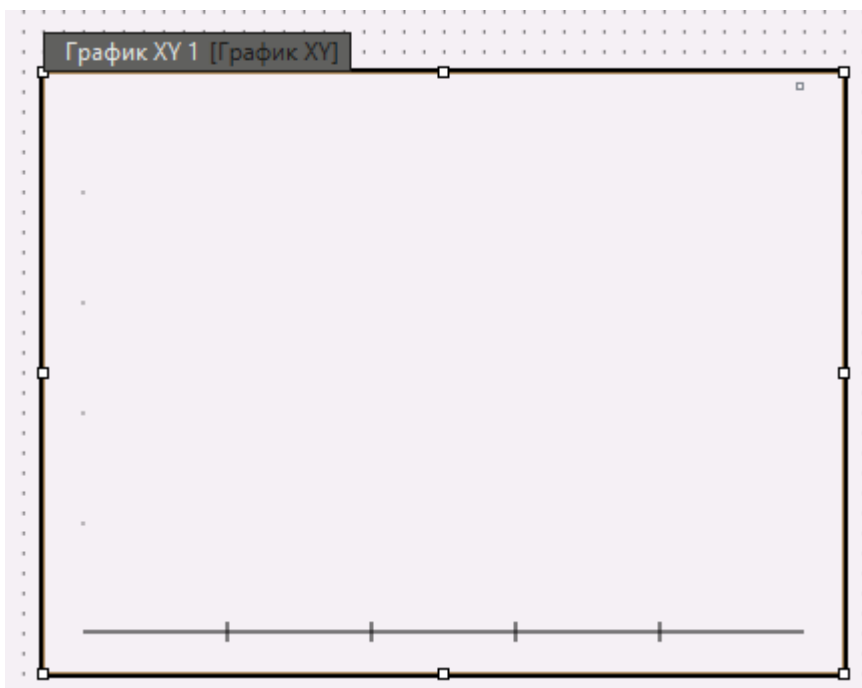
Важно! Элемент поддерживается только в HMI v2

В палитре редактора HMI элемент График XY находится в категории Контролы.

Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:



Для того, чтобы добавить параметр с типом данных массив на График XY, 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:

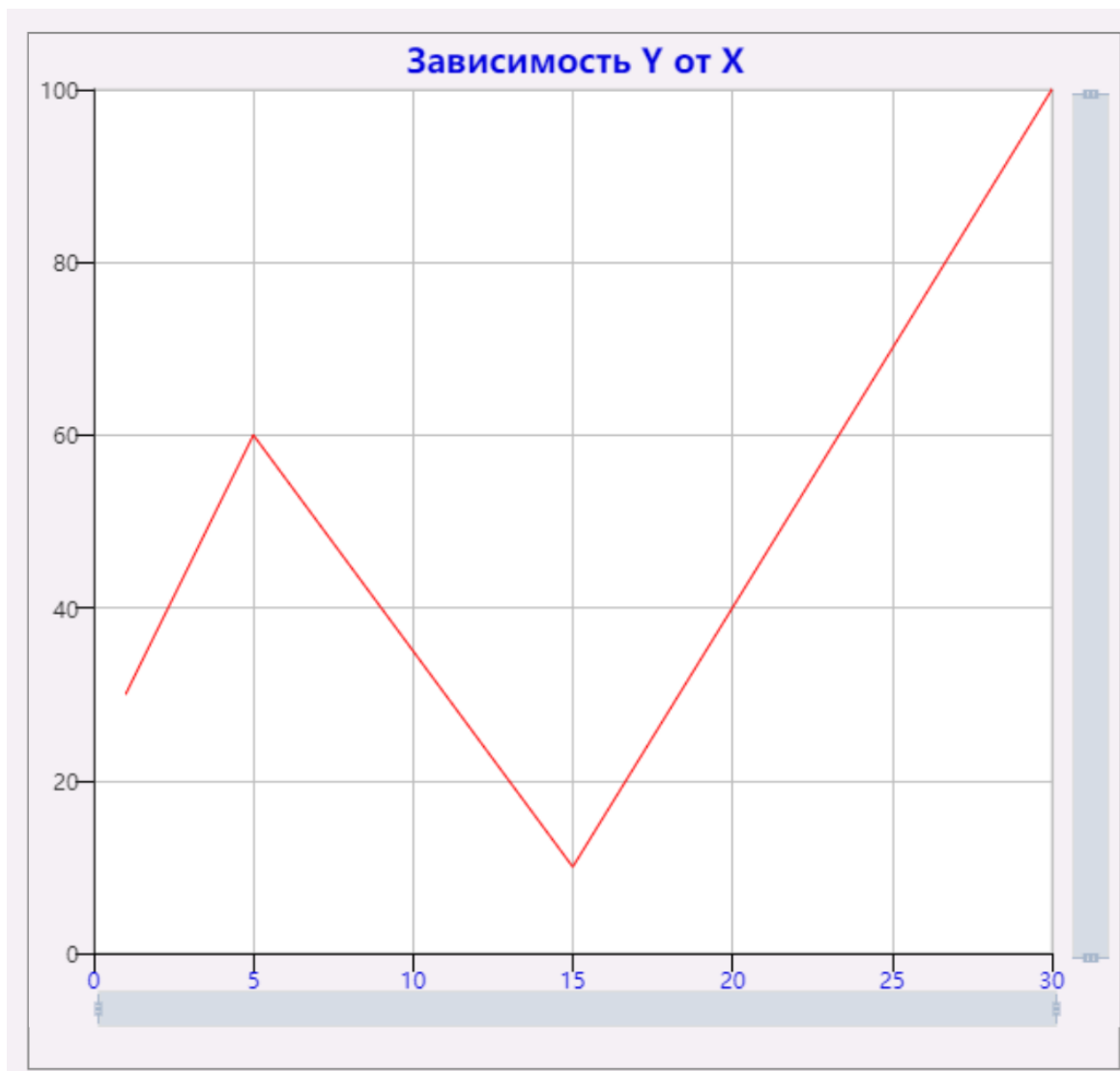


График в клиенте визуализации имеет следующие элементы:

- 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

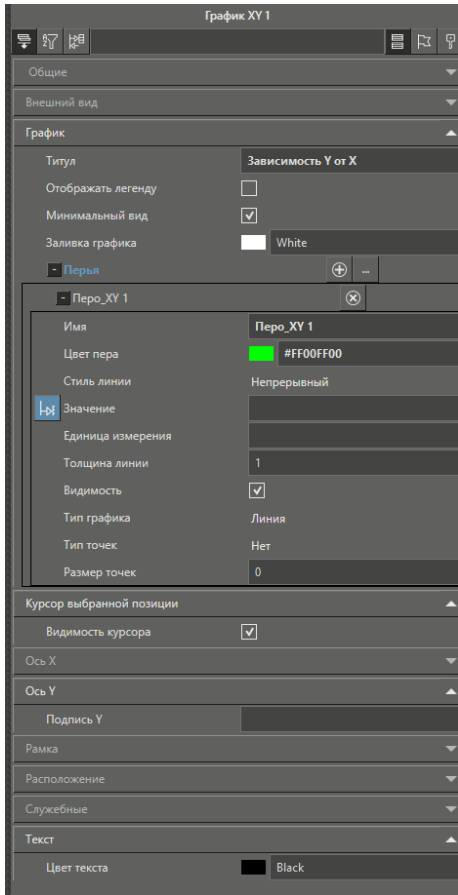
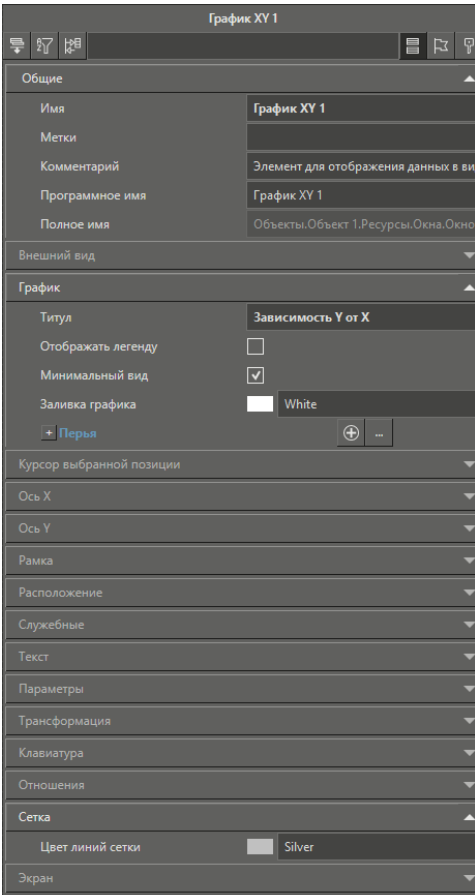
- Линейки прокрутки - инструменты для изменения отображаемого интервала. Находятся справа от поля графика и под ним.

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



### 8.3.2.3.1. Свойства Графика XY




Вид панели свойств элемента при нажатой и отжатой кнопке :


Простой режим	Полный режим
	

Описание основных свойств графического элемента График XY:

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.
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	Задаёт формат значений оси X графика.
Количество делений X	Задаётся 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.
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 значений (первое, последнее, минимальное, максимальное). По умолчанию Шаг прореживания равен 1.
Категория Сетка	

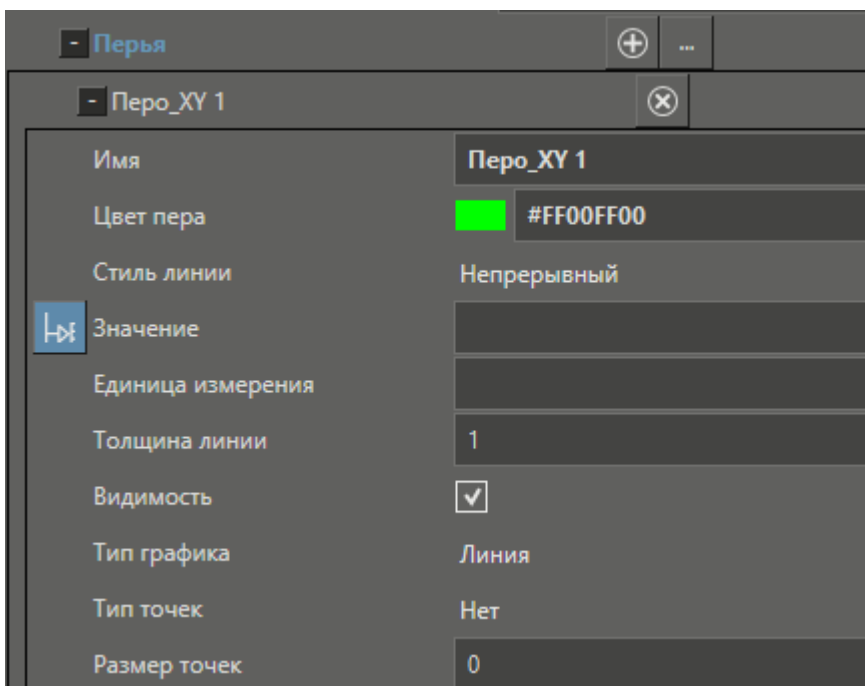
Цвет сетки	<p>Определяет цвет линий сетки. При нажатии на кнопку  , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора цвета.</p>
------------	---


### 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:



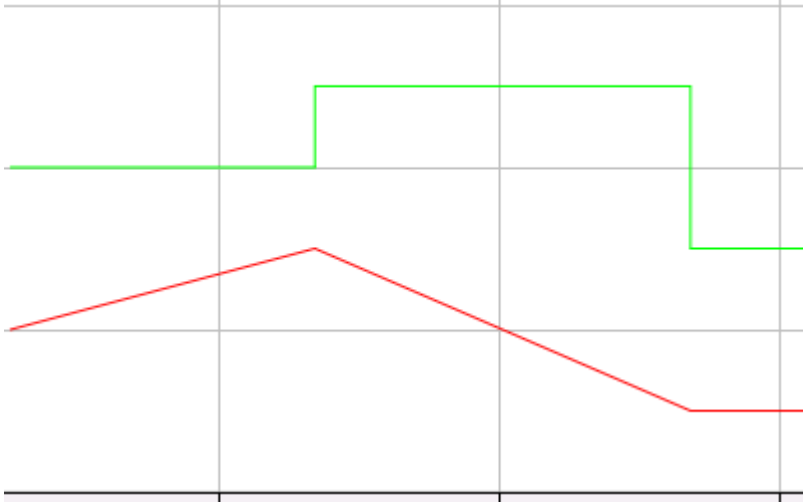

Новая группа добавится автоматически, после того как разработчик перетащит новый параметр в поле графика. Кроме того, группа настроек может быть добавлена вручную с использованием кнопки  .

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"> <li>• Линия - между двумя точками будет рисоваться прямая линия, т.е. будет считаться, что значение параметра менялось плавно.</li> <li>• 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.</li> </ul> <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.
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"> <li>• Нет</li> <li>• Квадрат</li> <li>• Круг</li> <li>• Треугольник</li> <li>• Ромб</li> <li>• Крест</li> </ul>
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.

#### 8.3.2.4. 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, квитировать 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 квитировании 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.

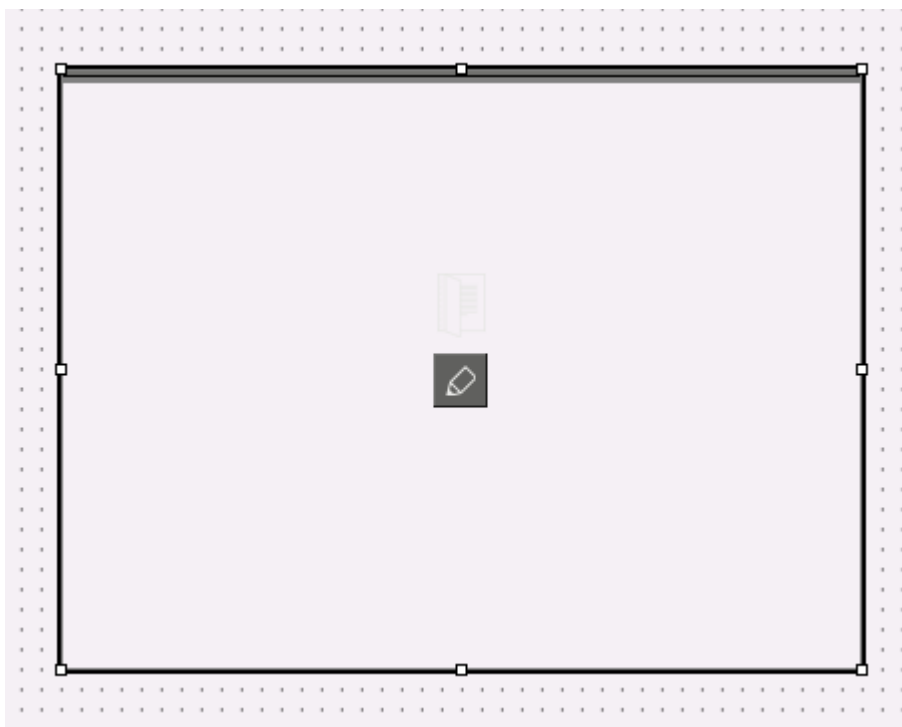
Вид элемента Журнал в палитре:




Вид элемента Архивный журнал в палитре:



Вид элементов после добавления в рабочую область редактора НМІ:



Для того чтобы начать настройку журналов, необходимо нажать кнопку: . При этом откроется диалоговое окно, в котором настраивается содержимое журнала, а также его внешний вид:

Редактирование Объекты.Объект 1.Ресурсы.Окно.Окно 1.Схема.Архивный журнал 1

Столбцы таблицы    Столбцы легенды    Фильтры    Стиль строки    Стиль заголовка

Добавить столбец

№ столбца	0	1	2	3	4
Заголовок			Событие	Время	Сообщение
Поле	Квит	Иконка	Событие	Время	Сообщение
Свойства					
Тип данных	STRING	STRING	HMLEventChangeType	DATE_AND_TIME	STRING
Ширина	45	20	*	*	*
Сортировка	Нет	Нет	Нет	Нет	Нет
Порядок сортировки столбца	0	0	0	0	0
Формат значений				dd.MM.yyyy HH:mm:ss	
Видимость	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Гориз. выравнивание	Центр	Центр	Центр	Центр	Центр

Глобальный

Заккрыть

Подробное описание диалогового окна смотрите в разделе Настройка журнала.

В зависимости от настроек, произведенных в данном окне, в клиенте визуализации журнал будет иметь вид:

Квитировать ...	Время	Сообщение	Комментарий
	29.12.2018 16:33:11	Кнопка с фиксацией 2 : Управление (Клик мыши) Оператор	
Квит	29.12.2018 16:33:10	Плановый ремонт	
	29.12.2018 16:33:10	Кнопка с фиксацией 2 : Управление (Клик мыши) Оператор	
	29.12.2018 16:33:04	Кнопка с фиксацией 1 : Управление (Клик мыши) Оператор	
Квит	29.12.2018 16:32:03	Останов оборудования	
	29.12.2018 16:33:03	Кнопка с фиксацией 1 : Управление (Клик мыши) Оператор	
	29.12.2018 16:32:59	Кнопка с фиксацией 1 : Управление (Клик мыши) Оператор	
	29.12.2018 16:32:53	Кнопка с фиксацией 2 : Управление (Клик мыши) Оператор	
Квит	29.12.2018 16:32:50	Останов оборудования	
	29.12.2018 16:32:50	Кнопка с фиксацией 1 : Управление (Клик мыши) Оператор	
Квит	29.12.2018 16:32:48	Плановый ремонт	
	29.12.2018 16:32:48	Кнопка с фиксацией 2 : Управление (Клик мыши) Оператор	
	29.12.2018 16:19:34	Текстовый ввод 1 : Управление (Клик мыши) Оператор	

Время 29.12.2018 16:33:11

Сообщение Кнопка с фиксацией 2 : Управление (Клик мыши) Оператор

Приоритет 100

Активность false

Время снятия

Квитированность true

Время квитирования

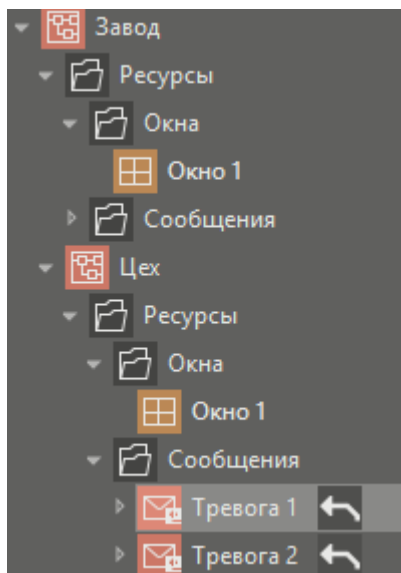
Комментарий

Пропущено: 0 Отображаемый интервал: 16:15:49 29.12.2018 - 16:45:49 29.12.2018

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.

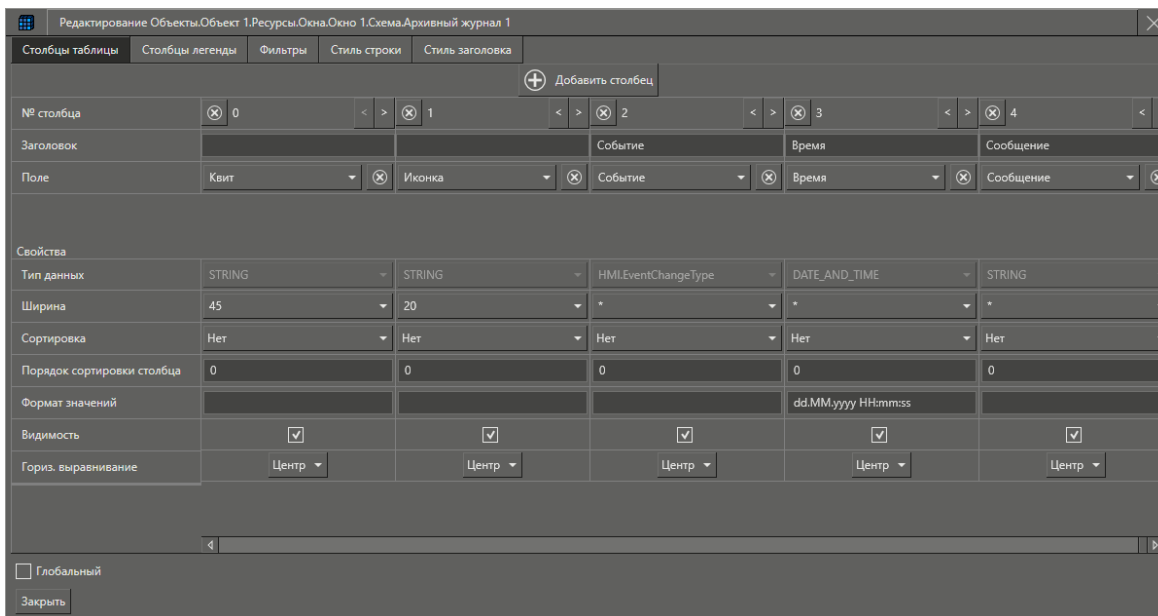
В случае, если флаг Глобальный отмечен, то журнал может отображать сообщения узла, на котором выполняется объект, а также всех других объектов, исполняемых на этом же узле.

В дальнейшем можно сделать выборку из сообщений, отображаемых в журнале, и ввести дополнительные фильтры.

### 8.3.2.4.1. 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 Закрывать окно настройки журнала будет закрыто.

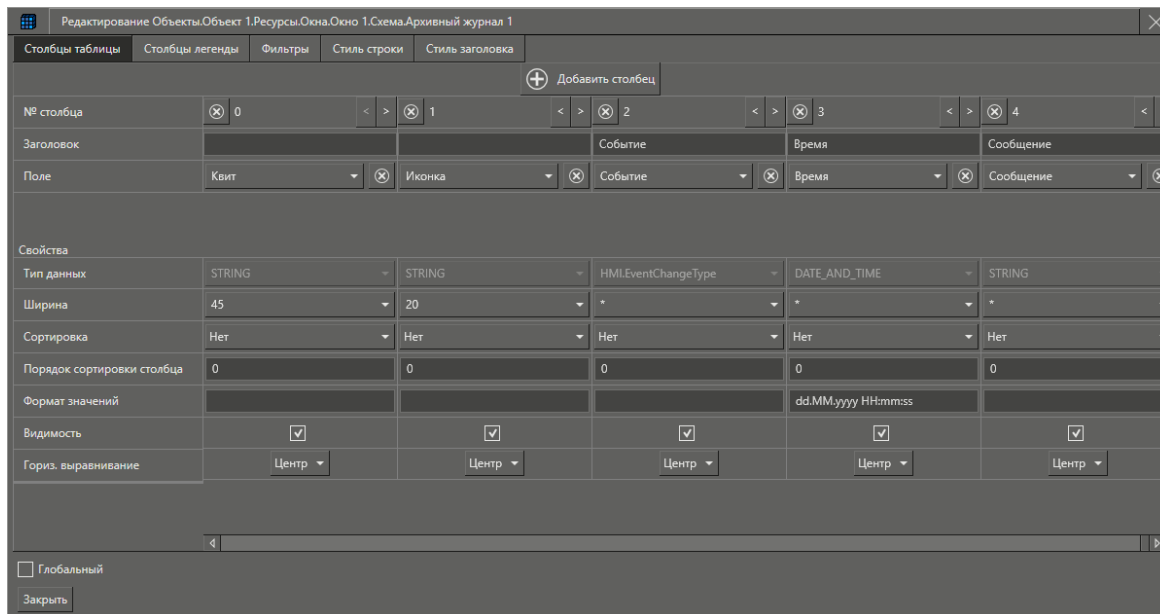
В панели свойств журнала, как правило, настраивается период обновления данных. При необходимости, некоторые свойства можно использовать для создания собственных инструментов управления журналом, например, для настройки интервала отображаемых сообщений и т.п.

#### **8.3.2.4.1.1. 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	Определяет порядок столбцов. В клиенте визуализации порядок столбцов будет таким же, как и в данной таблице. Чтобы переместить столбец влево или вправо следует воспользоваться соответствующими стрелками -   . Нажатие на кнопку  приведет к удалению столбца.
Заголовок	Определяет текст заголовка столбца журнала. Существующий текст заголовка можно редактировать после двойного нажатия на него левой кнопкой мыши :  Квитир

Поле	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.
Свойства	
Тип данных	Показывает тип данных свойства Поле. Определяется автоматически после настройки строки Поле. Носит информативный характер и не может быть изменен при помощи окна настройки журнала.
Ширина	<p>Определяет ширину столбца журнала. Возможны следующие варианты значения свойства:</p> <ul style="list-style-type: none"><li>• 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.</li><li>• 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.</li><li>• 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 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 для просмотра их содержимого.</li></ul>



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"> <li>• Нет - сортировка в столбце не используется</li> <li>• По возрастанию - сортировка в столбце используется, первыми в таблице располагаются сообщения, у которых поле имеет меньшее значение.</li> <li>• По убыванию - сортировка в столбце используется, первыми в таблице располагаются сообщения, у которых свойство имеет большее значение</li> </ul> <p><b>Важно!</b> В НМІ v2 сортировка в столбцах поддерживается только в элементе Журнал.</p>
Порядок сортировки столбцов	<p>Определяет приоритет сортировки столбцов в случае, когда необходима сортировка по нескольким столбцам (множественная сортировка). В текущей версии не поддерживается.</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, отображались дата и время с точностью до миллисекунды: dd.ММ.уууу НН: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.ММ.уууу НН: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"> <li>• Центр</li> <li>• Лево</li> <li>• Право</li> </ul>

Возможные значения свойства Поле по умолчанию.

Значение	Описание	Тип сообщений
Квит	<p>Этот столбец дает возможность пользователю квити́ровать сообщение, а также оценить, какие сообщения требуют квити́рования.</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 квити́рования (только для НМІ v2).</p>	Условные сообщения
Иконка	<p>Отображает графическое изображение сообщения. Для того чтобы назначить иконку Тревоге, необходимо в контекстном меню библиотечного элемента выбрать пункт Дерево.Установить иконку. В этом случае у всех экземпляров, добавленных в дерево объектов, будет отображаться соответствующая иконка.</p>	Условные сообщения
Полное имя объекта	<p>Отображает путь к родительскому объекту источника сообщения в проекте. Например, Объекты. Завод. Цех 1.Линия</p>	Все
Объект (Object)	<p>Отображает имя родительского объекта источника сообщения. Например, Линия</p>	Все
Источник	<p>Отображает имя источника сообщения. Если сообщение появилось в результате нарушения границ, то отобразится имя параметра (тега, канала), для которого назначена шкала</p>	Все
Состояние	<p>Отображает, в каком состоянии находится сообщение.</p> <p>Возможны следующие варианты:</p> <ul style="list-style-type: none"> <li>0 - не активно, не квити́ровано;</li> <li>1 - не активно, квити́ровано;</li> <li>2 - активно, не квити́ровано;</li> <li>3 - активно, квити́ровано.</li> </ul>	Все

	<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>	
Время активации (ActiveTime)	Отображает время появления сообщения.	Все
Пользователь	Отображает имя пользователя, который квитировал 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 квитировал 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"> <li>1. Появление</li> <li>2. Исчезновение</li> <li>3. Квитирование</li> </ol>	Все

Сообщение (Message)	Отображает текст сообщения. У пользовательских тревог в журнал попадает значение свойства Сообщение.	Все
Приоритет (Severity)	Отображает приоритет сообщения. У пользовательских тревог значение приоритета настраивается. Предопределенные сообщения имеют следующие приоритеты: 100 - сообщение о действиях пользователя 750 - сообщения о нарушении аварийных границ 500 - сообщения о нарушении предупредительных границ	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 столбца будет TRUE, а если сообщение не было квитировано, то False.  Для сообщений о действиях пользователей это свойство всегда принимает значение TRUE	Все
Время квитирования	Отображает момент времени, в который сообщение было квитировано.	Условные сообщения

(AckedTime)		
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 скорости изменения
Время (ChangeTime)	Отображает момент времени, в который произошло изменение состояния сообщения.	Все

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.

### 8.3.2.4.1.2. 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	⊗ Время	⊗ Сообщение	⊗ Приоритет	⊗ Время активации	⊗ Время снятия	⊗ Время квитиров	⊗ Пользователь	⊗ Комментарий	
Поле	Время ▾	Сообщение ▾	Приоритет ▾	Время акти ▾	Время деак ▾	Время квит ▾	Пользовате ▾	Комментар ▾	
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.

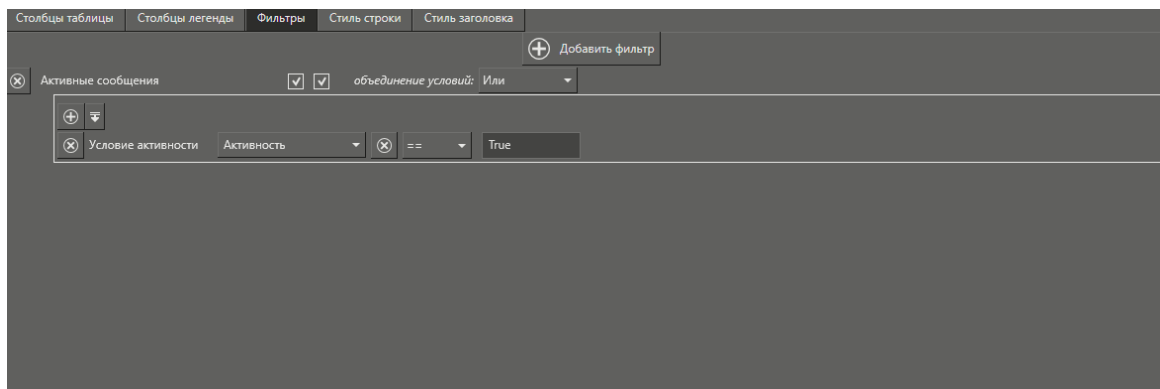
### 8.3.2.4.1.3. Окно настройки журнала. Фильтры

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 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.
	Фильтр срабатывает когда сообщения удовлетворяют условиям. Данное поле определяет работу фильтра том в случае, если условий несколько. Возможные варианты:





	<p>Или - фильтр сработает при выполнении хотя бы одного условия.</p> <p>И - фильтр сработает 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.</p> <p>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. Возможные варианты:</p> <p>Содержит - условие работает, если выбранная характеристика сообщения содержит в себе символы, введенные в соседнее поле;</p> <p>== - условие работает, если выбранная характеристика сообщения равна значению, введенному в соседнее поле;</p> <p>&gt; - условие работает, если выбранная характеристика сообщения больше значения, введенного в соседнее поле;</p> <p>&lt;- условие работает, если выбранная характеристика сообщения меньше значения, введенного в соседнее поле;</p> <p>&gt;= - условие работает, если выбранная характеристика сообщения больше или равна значению, введенному в соседнее поле;</p> <p>&lt;= - условие работает, если выбранная характеристика сообщения меньше или равна значению, введенному в соседнее поле;</p> <p>!= - условие работает, если выбранная характеристика сообщения не равна значению, введенному в соседнее поле.</p>
	<p>Поле, в которое вводится значение, которое сравнивается с выбранной характеристикой сообщения.</p>

### Работа с условием Проверка типа сообщения

После добавления условия появится строка:



В поле строки необходимо перетащить нужный тип сообщения из библиотеки.

### Работа с условием Проверка отношения

После добавления условия появится строка:



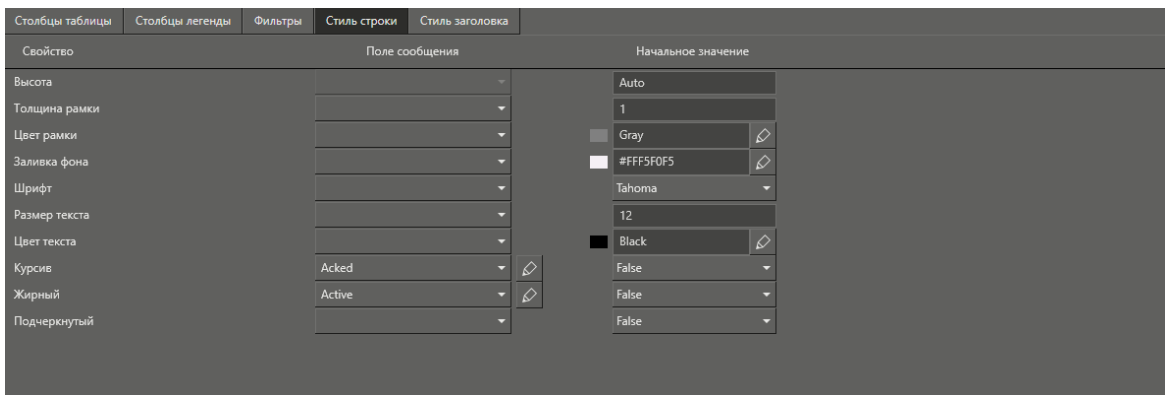
В поле строки необходимо перетащить объект, тег, канал, узел, т.е. тот элемент, сообщения которого необходимо отображать при использовании этого фильтра.

### 8.3.2.4.1.4. Окно настройки журнала. Стилль строки

Настройки, описанные в данном разделе, применимы к элементам Журнал и Архивный журнал.

Вкладка Стилль 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 квитирированные сообщения отображаются курсивом.

Вкладка имеет вид :



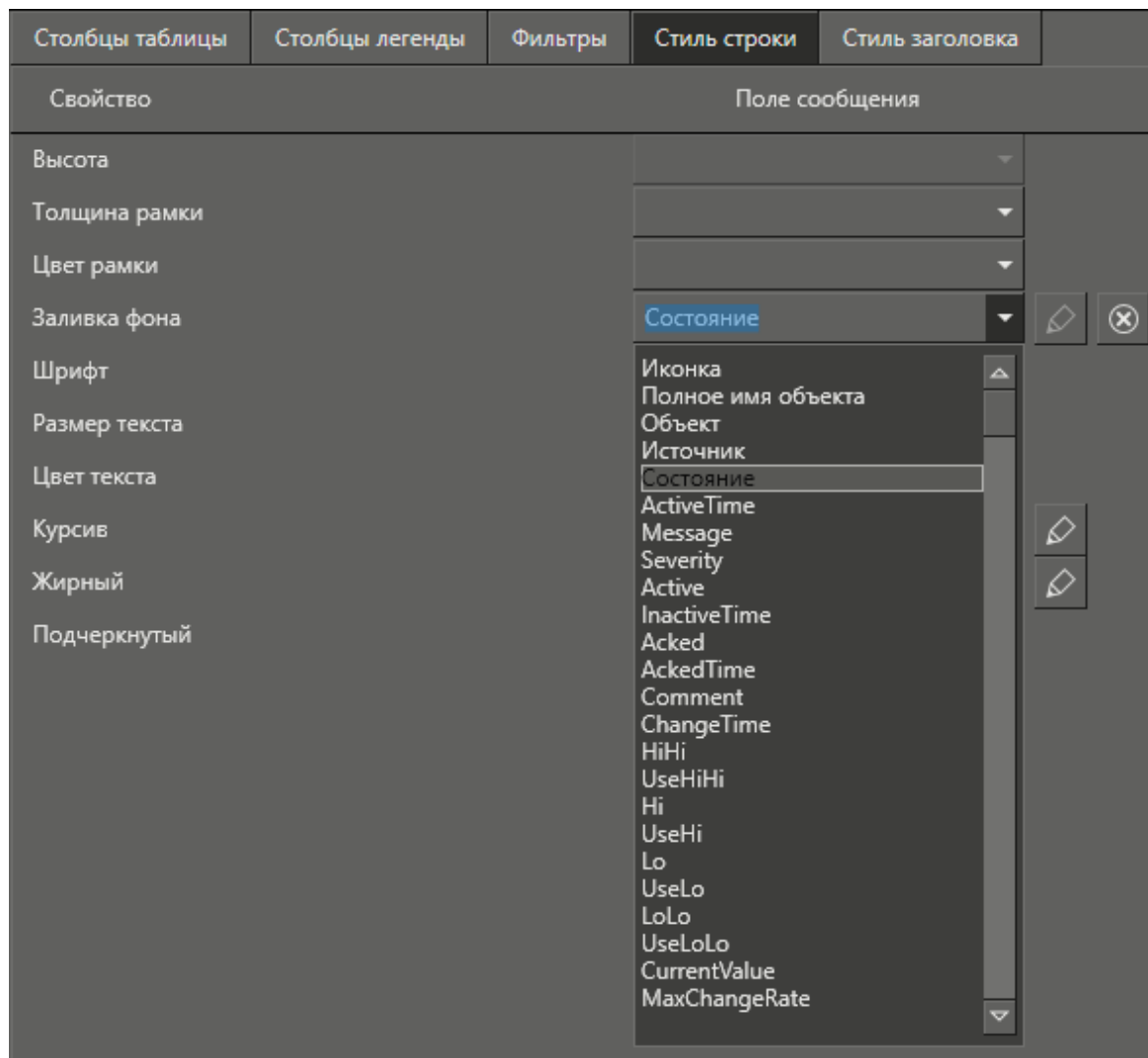
На вкладке находится таблица, состоящая из трех столбцов:


Название	Назначение
Свойство	Название свойства строки.
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 стилия строки

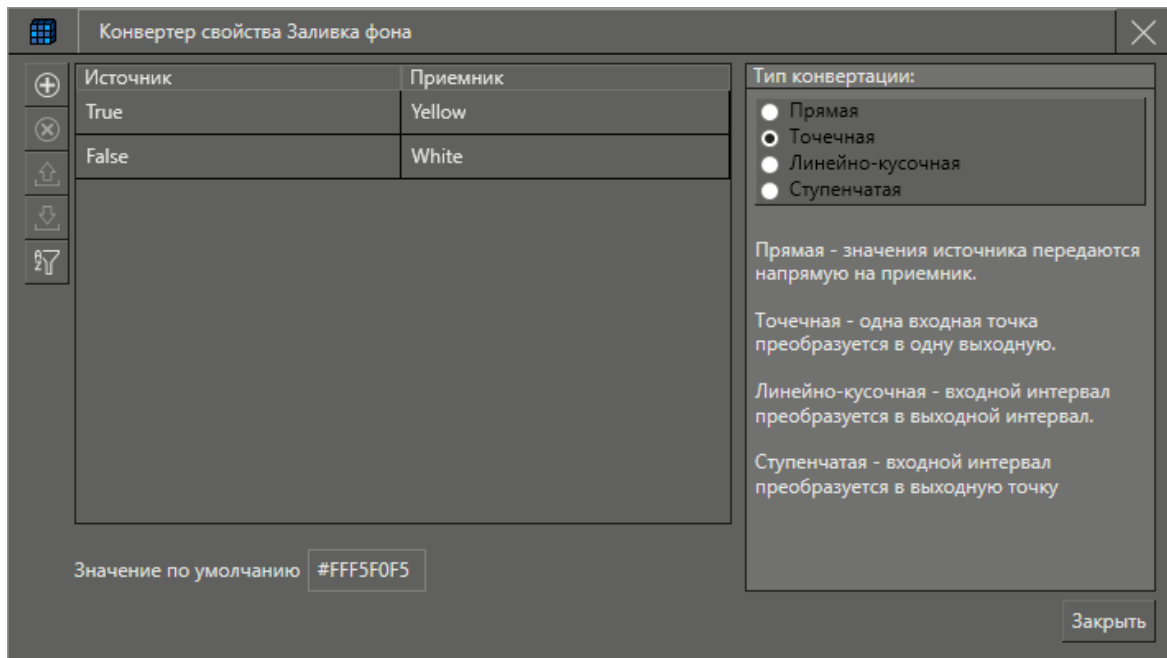
Допустим, что необходимо активные сообщения отображать на желтом, а сообщения, уже потерявшие свою актуальность, - на белом.

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:

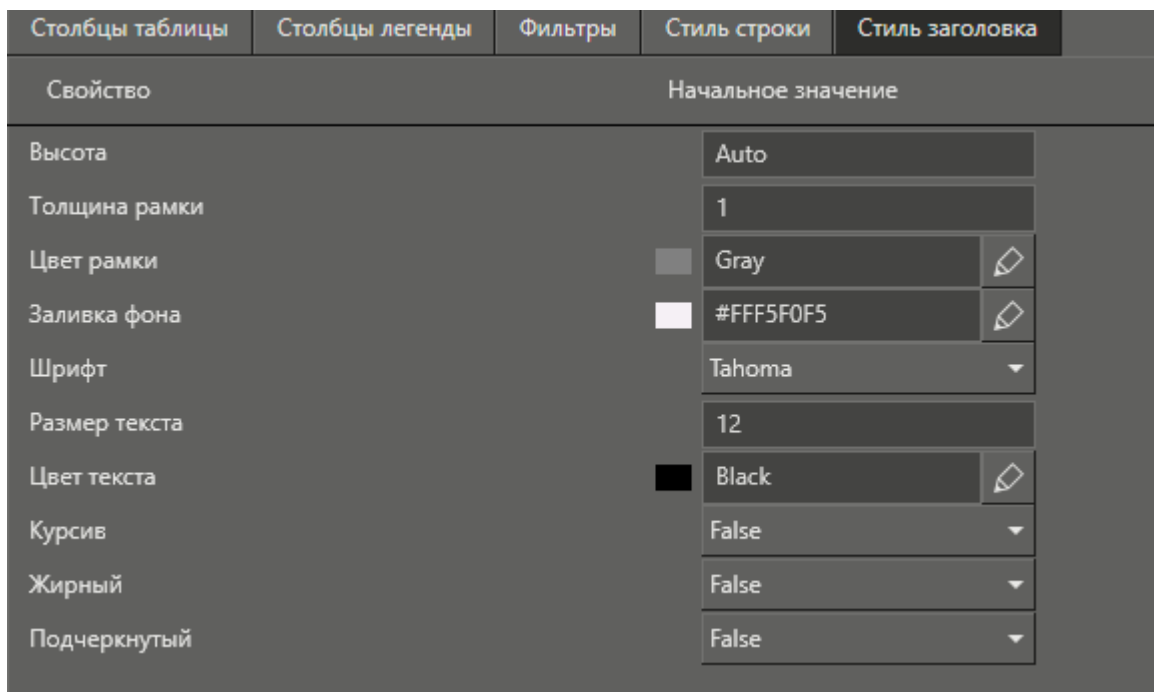


### 8.3.2.4.1.5. Окно настройки журнала. Стиль заголовка

Настройки, описанные в данном разделе, применимы к элементам Журнал и Архивный журнал.


Вкладка **Стиль заголовка** определяет внешний вид заголовка таблицы сообщений.

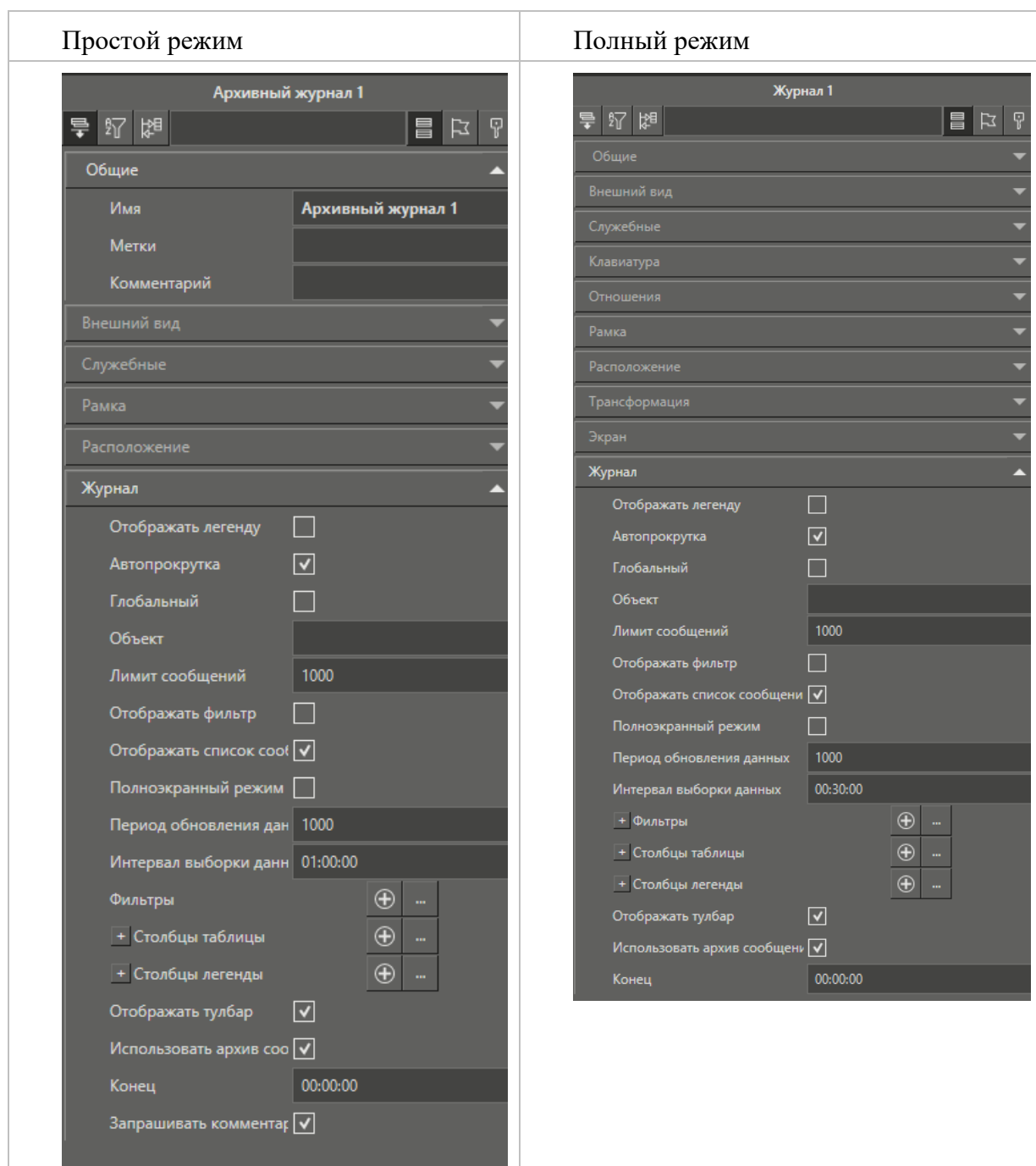
Вкладка имеет вид:



В столбце **Начальное значение** задаются значения параметров строки.




### 8.3.2.4.1.6. Свойства Журнала

Вид панели свойств элементов Журнал и Архивный журнал при нажатой и отжатой кнопке :









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.
Глобальный	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 .
Объект	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.
Лимит сообщений	Задаёт максимальное число сообщений, извлекаемое из БД (только для НМІ v2).
Отображать фильтр	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.
Отображать список сообщений	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.
Полноэкранный режим	Определяет, будет ли отображаться элемент журнал в полноэкранный режим после открытия окна, содержащего журнал. В клиенте визуализации имеется возможность включать/выключать полноэкранный режим, 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.
Период обновления данных	Задаёт период автоматического обновления данных при включенной автопрокрутке.

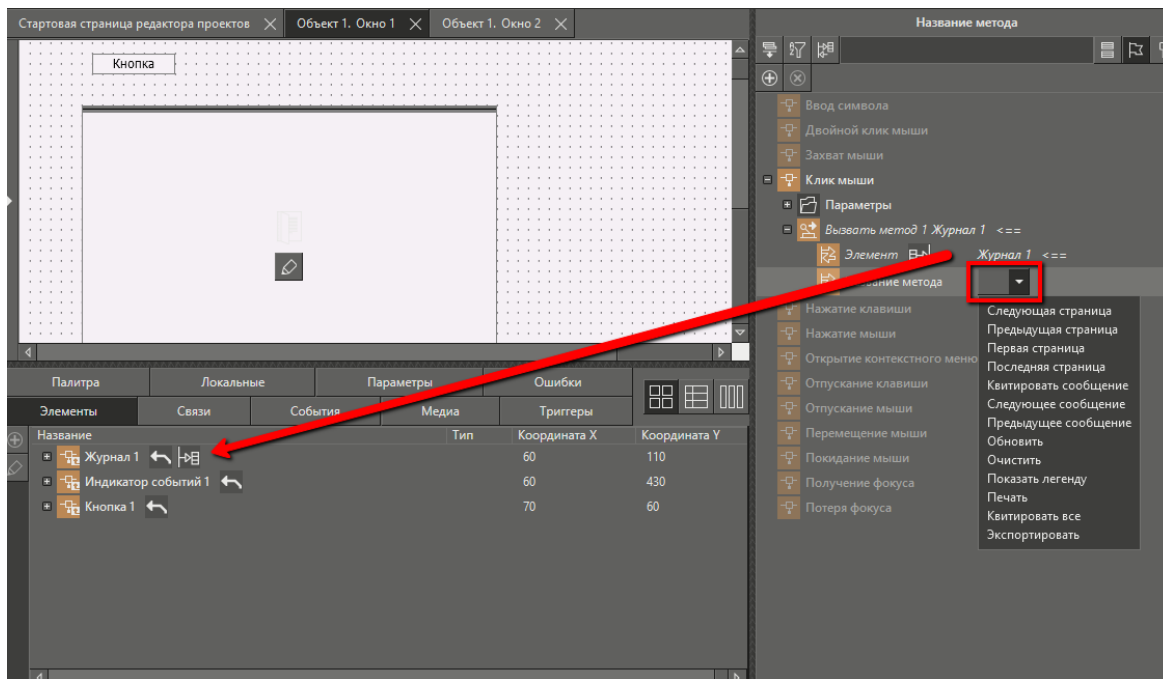
Интервал выборки данных	Задаёт интервал, который будет отображаться на одной странице журнала. Свойство может быть задинамизировано, т.е. зависеть от других элементов управления, созданных разработчиком проекта, либо от какой-то программы.
Фильтры	Группа настроек, отображающих ранее созданные фильтры 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  .
Столбцы таблицы	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  .
Столбцы легенды	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  .
Отображать тулбар	Определяет, будет ли отображаться панель инструментов журнала. Если флаг снят, и панель инструментов не 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 HMI v2).

### 8.3.2.4.1.7. 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.
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.
Квитируют все	Квитирует все видимые сообщения на странице.
Экспортировать	Производит экспорт загруженных сообщений журнала в файл.

#### 8.3.2.4.2. 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 HMI v1

Журнал в клиенте визуализации при HMI v2

### 8.3.2.4.2.1. Журнал в клиенте визуализации при HMI v1

Вид журнала в клиенте визуализации:

Квит	Событие	Время	Сообщение	Источник
	Появление	20.05.2019 14:01:25	Уровень воды в норме	Тревога 4
	Появление	20.05.2019 14:01:25	Кнопка с фиксацией 4 : Управление (Кли...	Объект 1
	Исчезновение	20.05.2019 14:01:24	Уровень воды в норме	Тревога 4
	Появление	20.05.2019 14:01:24	Кнопка с фиксацией 4 : Управление (Кли...	Объект 1
Квит	Появление	20.05.2019 14:01:16	Уровень воды в норме	Тревога 4
	Появление	20.05.2019 14:01:16	Кнопка с фиксацией 4 : Управление (Кли...	Объект 1
	Исчезновение	20.05.2019 14:01:11	Выполнена проверка всех систем. Норма	Тревога 3
	Появление	20.05.2019 14:01:11	Кнопка с фиксацией 3 : Управление (Кли...	Объект 1
Квит	Появление	20.05.2019 14:01:03	Выполнена проверка всех систем. Норма	Тревога 3
	Появление	20.05.2019 14:01:03	Кнопка с фиксацией 3 : Управление (Кли...	Объект 1
	Исчезновение	20.05.2019 13:55:30	Авария насоса	Тревога 1
	Появление	20.05.2019 13:55:30	Кнопка с фиксацией 1 : Управление (Кли...	Объект 1
Квит	Появление	20.05.2019 13:55:29	Авария насоса	Тревога 1
	Появление	20.05.2019 13:55:29	Кнопка с фиксацией 1 : Управление (Кли...	Объект 1

Время 20.05.2019 14:01:25

Сообщение Уровень воды в норме

Приоритет 5

Время активации 20.05.2019 14:01:25

Время снятия

Время квитирования

Пользователь

Комментарий












Пропущено: 10 Отображаемый интервал: 13:01:41 20.05.2019 - 14:01:41 20.05.2019







Где:


- 1 - Панель инструментов
- 2 - Таблица сообщений
- 3 - Легенда сообщений
- 4 - Строка статуса
- 5 - Линейка прокрутки

Панель инструментов

Панель содержит следующие инструменты:

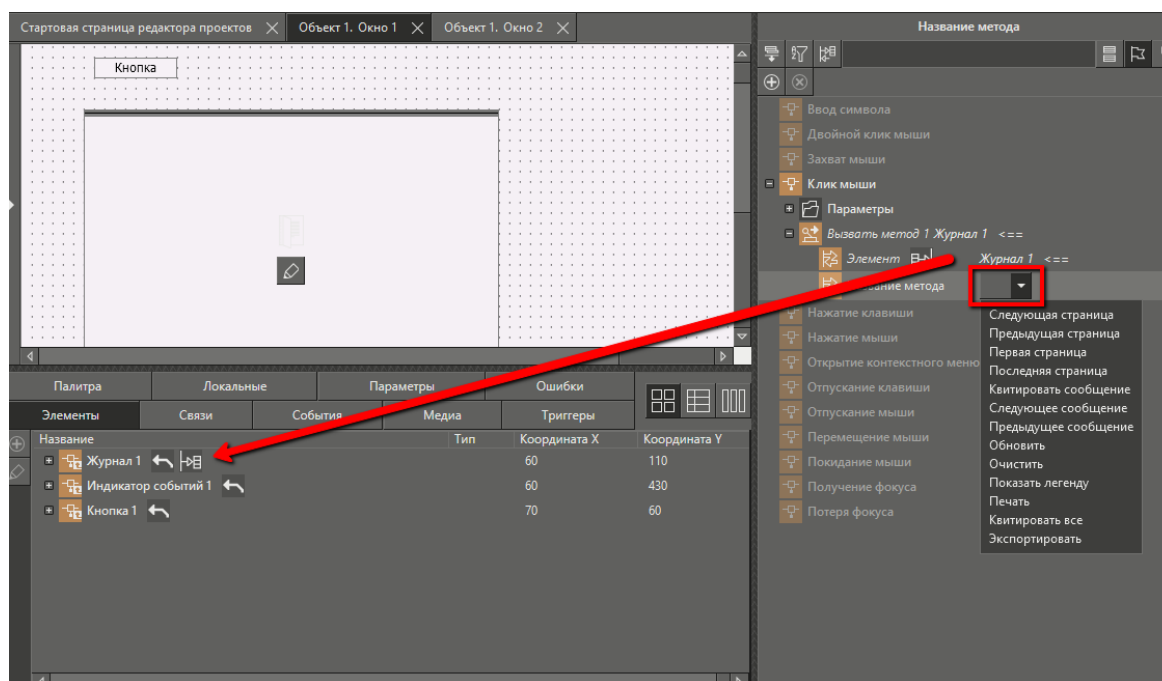
Элемент	Название	Назначение
	Квитировать все	Квитирует все сообщения на странице.
	Экспортировать	Открывает диалог сохранения текстового файла, в который экспортируются сообщения на текущей странице (имя по умолчанию – journal.txt).
	Печать	Открывает диалог печати. Будут выведены на печать все загруженные 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.

 / 	<p>Остановить/з апустить автопрокрутк у</p>	<p>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 появилось в журнале за то время, пока автопрокрутка была остановлена.</p>
 / 	<p>Скрыть грид</p>	<p>Скрывает и отображает таблицу сообщений в журнале.</p>
 / 	<p>Скрыть/показ ать 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="662 1171 1019 1528" style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;"> <input checked="" type="checkbox"/> Квитированные сообщения  <input type="checkbox"/> Тип сообщения  <input type="checkbox"/> Сообщения АРМ2 </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>Полноэкранный режим</p>	<p>Включает полноэкранный режим, когда размер журнала станет равным размеру окна, в котором он находится.</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:

Квитировать	Время	Сообщение	Объект
	30.01.2019 14:19:21	Кнопка с фиксацией 1 : Управл...	Объект 1
Квит	<i>30.01.2019 14:19:21</i>	<i>Останов оборудования</i>	<i>Объект 1</i>
	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	<i>30.01.2019 14:19:20</i>	<i>Останов оборудования</i>	<i>Объект 1</i>
	30.01.2019 14:19:20	Кнопка с фиксацией 1 : Управл...	Объект 1
	<b>30.01.2019 14:19:19</b>	<b>Плановый ремонт</b>	<b>Объект 1</b>
	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	<i>30.01.2019 14:19:17</i>	<i>Плановый ремонт</i>	<i>Объект 1</i>
	30.01.2019 14:19:17	Кнопка с фиксацией 2 : Управл...	Объект 1
Квит	<i>30.01.2019 14:19:01</i>	<i>Минимальное аварийное</i>	
Квит	<i>30.01.2019 14:16:50</i>	<i>Плановый ремонт</i>	<i>Объект 1</i>
	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
	<b>30.01.2019 14:16:49</b>	<b>Останов оборудования</b>	<b>Объект 1</b>
	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 Стиль 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 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.

	<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.</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.

Вид:

Время	04.02.2019 15:19:45
Сообщение	Плановый ремонт
Приоритет	1
Активность	true
Время снятия	
Квнтированность	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
---

## 8.3.2.4.2.2. Журнал в клиенте визуализации при НМІ v2

Вид журнала в клиенте визуализации:



Событие	Время	Сообщение	Пользователь
Квитирование	20.02.2020 15:40:11	Останов оборудования	Оператор
<b>Появление</b>	<b>20.02.2020 15:40:03</b>	<b>Останов оборудования</b>	<b>Оператор</b>
Появление	20.02.2020 15:40:03	Управление: Кнопка с фиксац...	Оператор
Появление	20.02.2020 15:40:01	Управление: Текстовый ввод 1	Оператор
квит	Появление	20.02.2020 15:39:56	Плановый ремонт
Появление	20.02.2020 15:39:56	Управление: Кнопка с фиксац...	Оператор

Время	20.02.2020 15:40:11
Сообщение	Останов оборудования
Приоритет	0
Время активации	20.02.2020 15:40:03
Время снятия	0
Время квитирования	20.02.2020 15:40:11
Пользователь	Оператор

Пропущено: 1 Отображаемый интервал: 20.02.2020 15:41:39 - 20.02.2020 15:25:09











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. 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 – journal.txt).
	Печать	Открывает диалог печати. Будут выведены на печать все загруженные в журнал сообщения.
	К первой странице	Открывает первую страницу журнала. Если автопрокрутка была включена, то при нажатии на эту кнопку она выключится.
	Последняя страница	Открывает последнюю страницу журнала. Если автопрокрутка была выключена, то при нажатии на эту кнопку она включится.
 / 	Скрыть/показать легенду	Включает и выключает отображение легенды.
 / 	Остановить/запустить автопрокрутку	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 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,
 / 	Скрыть/показать фильтр	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.

		<ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Пользователь</li> <li><input checked="" type="checkbox"/> Объект</li> <li><input checked="" type="checkbox"/> Только АРМ диспетчера</li> </ul> <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>
--	--	--

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:

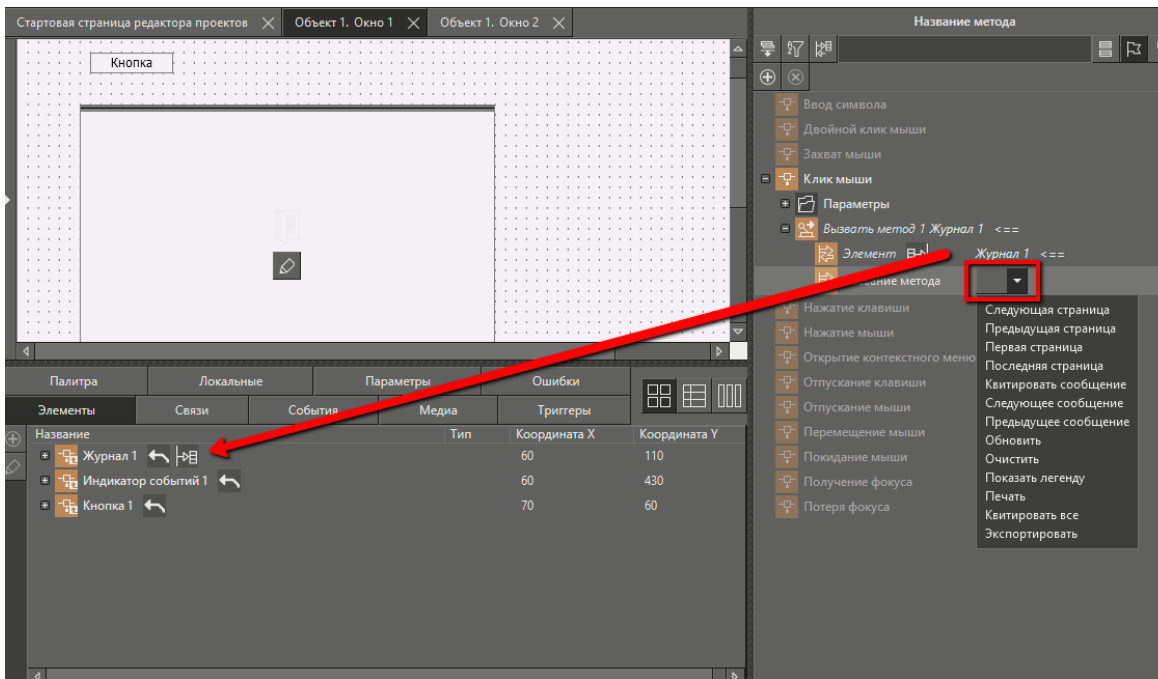


Таблица сообщений

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 15:41:38	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
		<b>Появление</b>	<b>20.02.2020 15:40:03</b>	<b>Останов оборудов...</b>	<b>Оператор</b>
		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
		<b>Появление</b>	<b>20.02.2020 15:25:09</b>	<b>Минимальное ава...</b>	<b>Оператор</b>


Пропущено: 0 Отображаемый интервал: 20.02.2020 16:13:54 - 20.02.2020 15:25:09

Инструменты таблицы:

Элемент	Описание
	<p>Заголовок таблицы. Вид заголовка зависит от настроек, заданных на вкладке Стилль заголовка окна настройки журнала. Изменить ширину столбца можно, передвигая левой кнопкой мыши разделительную линию между названиями столбцов.</p> <p>При нажатии на кнопку  рядом с заголовком столбца, откроется окно настройки фильтра, где можно указать, какие элементы необходимо отображать в таблице.</p>
	<p>Строка таблицы. В каждой строке отображается информация о том или ином сообщении. Количество информации в строке таблицы зависит от настроек, заданных на вкладке</p>

	<p>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 квитируемые сообщения - курсивом.</p>
	<p>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.</p>
	<p>Кнопка квитирувания сообщения. Для квитируемых сообщений, а также для сообщений, не требующих квитирувания, не отображается.</p> <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 квитирувания будет заполнен столбец Пользователь (в случае если он используется в таблице журнала).</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

### 8.3.2.5. 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 НМІ элемент Индикатор событий находится в категории Контролы.

Вид элемента в палитре:

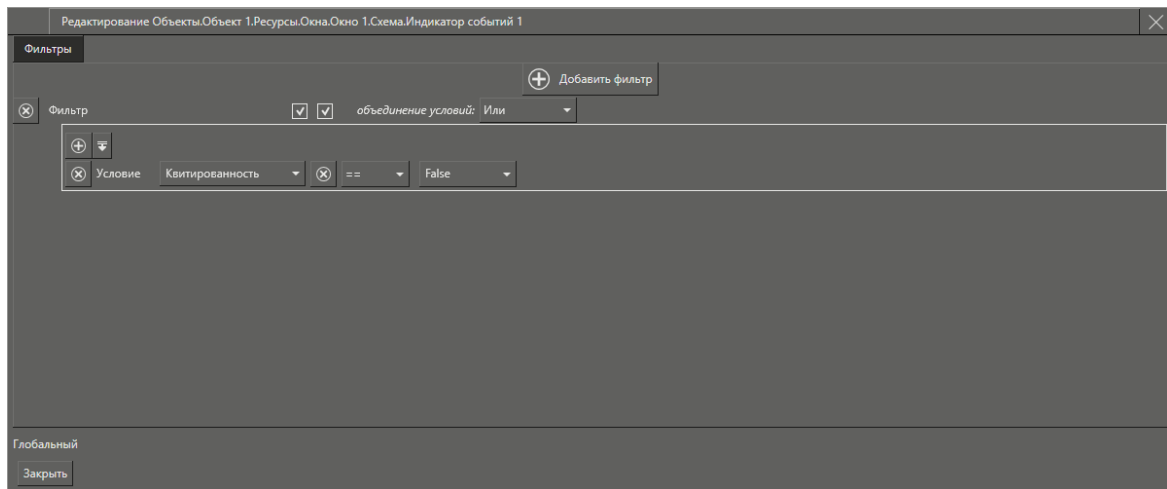


Вид элемента после добавления в рабочую область редактора НМІ:




При наведении мыши на элемент становится активной кнопка , при нажатии на которую открывается окно настройки индикатора.

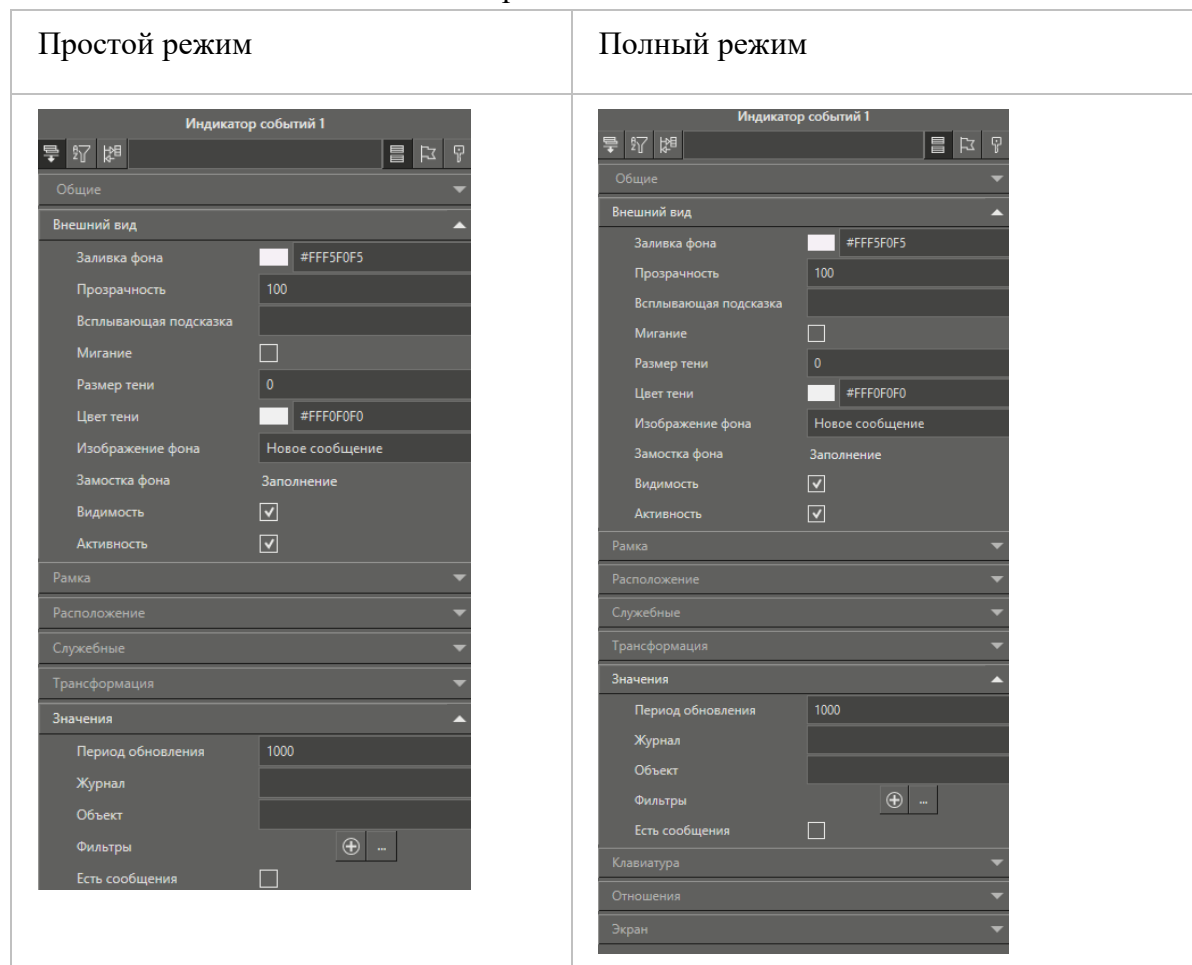
Окно настройки индикатора




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 :




## Описание основных свойств графического элемента Индикатор событий:

Название	Ссылка на описание
Категория Внешний вид	
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"> <li>• Заполнение - размер фонового изображения будет изменен по границам элемента (используется по умолчанию);</li> <li>• Мозаика - размер фонового изображения будет исходным, и повторен столько раз, сколько необходимо чтобы замостить всю площадь элемента;</li> <li>• Центр - размер фонового изображения будет исходным, изображение будет центрировано;</li> <li>• Нет - фоновое изображение использоваться не будет.</li> </ul>
Категория Значения	
Период обновления	Период с которым происходит проверка новых сообщений на сервере.
Журнал	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.
Объект	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 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
	

### 8.3.2.6. 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.

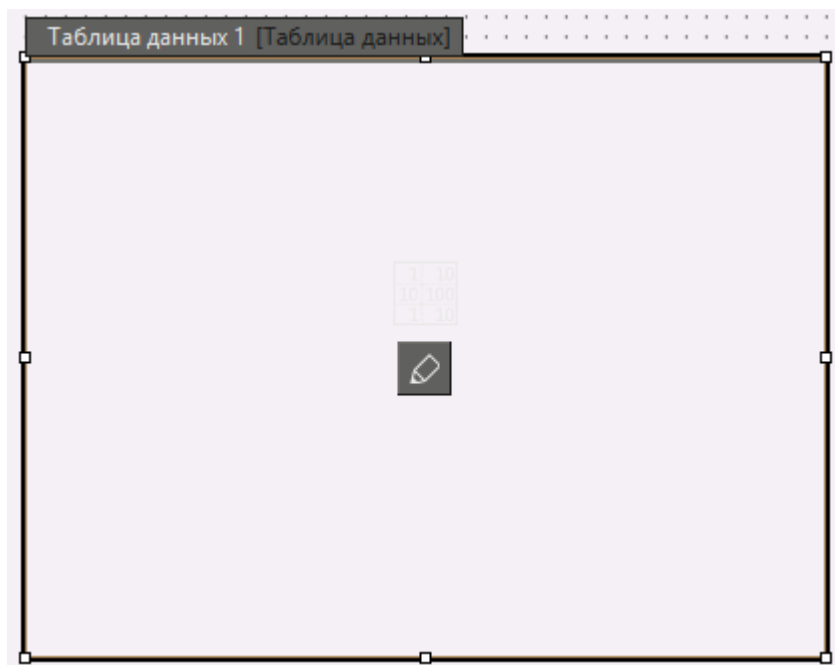
В палитре редактора НМІ элемент Таблица данных находится в категории Контролы

**Важно!** Этот элемент рекомендуется использовать в НМІ 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.

Редактор таблицы данных

Чтобы открыть редактор, необходимо нажать на кнопку  после размещения элемента Таблица данных в окне:

Редактирование Объекты.Объект 1.Ресурсы.Окна.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_задвижки
Поле	Расход	Давление	Температура	Состояние_задвижки

Свойства

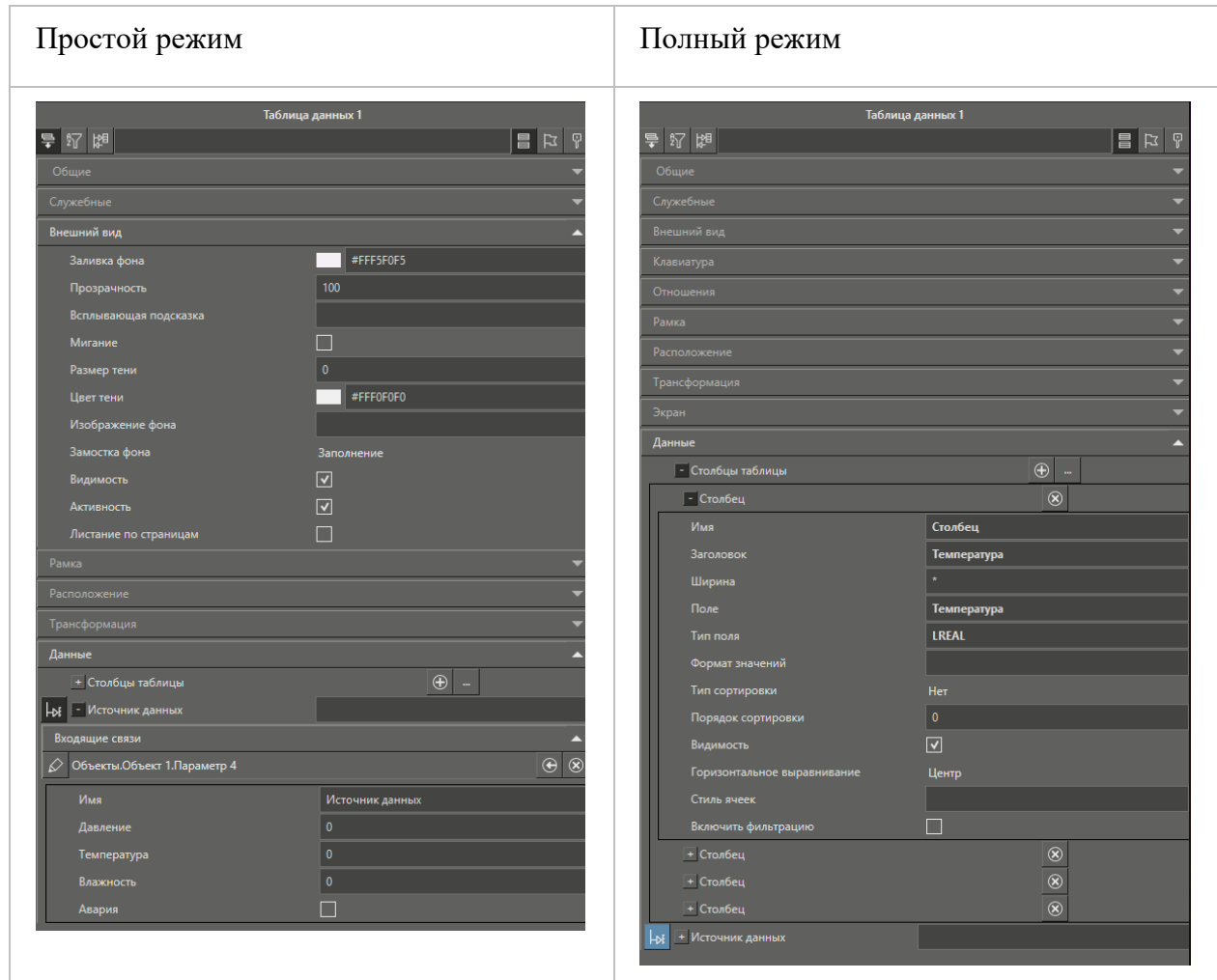
Свойство	0	1	2	3
Тип данных	LREAL	LREAL	LREAL	BOOL
Ширина	*	*	*	*
Сортировка	Нет	Нет	Нет	Нет
Порядок сортировки столбца	0	0	0	0
Формат значений				
Видимость	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
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"/>

Закреть

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
<b>Категория Внешний вид</b>	
Листание по страницам	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: <div style="text-align: center; border: 1px solid gray; padding: 5px; margin: 10px 0;"> <span>Предыдущая</span> <span style="border: 1px solid gray; padding: 2px 10px;">1</span> <span>2</span> <span>3</span> <span>Следующая</span> </div> Если свойство имеет значение False, то таблице не будет разделяться на страницы, но появится линейка прокрутки.

Категория Данные	
Столбцы таблицы	При нажатии на кнопку  в таблицу добавится столбец . В панели свойств добавится группа свойств Элемент массива. При нажатии на кнопку  откроется Редактор таблицы данных.
Столбец	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.  Столбцы таблицы рекомендуется настраивать с использованием Редактора таблицы данных.
Источник данных	Основное свойство элемента. Его необходимо связать с параметром типа массив структур. После этого появится возможность настраивать столбцы таблицы.

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)

Таблица данных в клиенте визуализации

Пример отображения Таблицы данных в клиенте визуализации:

Качество	Значение	Метка времени
192	7	09-02-2017 21:00:28
192	2	09-02-2017 21:00:28
192	8	09-02-2017 21:00:28

Выделенная строка в таблице подсвечивается. Таблица служит только для отображения данных. В случае, если необходимо не только отображать данные, но и редактировать их, то следует использовать элемент Стековая панель.

### 8.3.2.6.1. 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	Определяет порядок столбцов. В клиенте визуализации порядок столбцов будет таким же, как и в данной таблице.

	<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 пустой, то порядок столбцов распределится случайным образом (не рекомендуется).</p>
Свойства	
Тип данных	<p>Показывает тип данных поля структуры. Определяется автоматически после настройки строки Поле. Носит информативный характер и не может быть изменен при помощи редактора таблицы данных.</p>
Ширина	<p>Определяет ширину столбца таблицы. Возможны следующие варианты значения свойства:</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• 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.</li> <li>• 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</li> </ul>

	<p>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.</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"> <li>• Нет - сортировка в столбце не используется.</li> <li>• По возрастанию - сортировка в столбце используется, первыми в таблице располагаются строки, у которых поле имеет меньшее значение.</li> <li>• По убыванию - сортировка в столбце используется, первыми в таблице располагаются строки, у которых поле имеет большее значение.</li> </ul>
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><b>Важно!</b> Множественная сортировка поддерживается только в НМІ v2</p>
Формат значений	<p>Определяет формат отображения выводимого значения. Описание этого свойства смотрите в разделе Свойство Формат значений.</p>

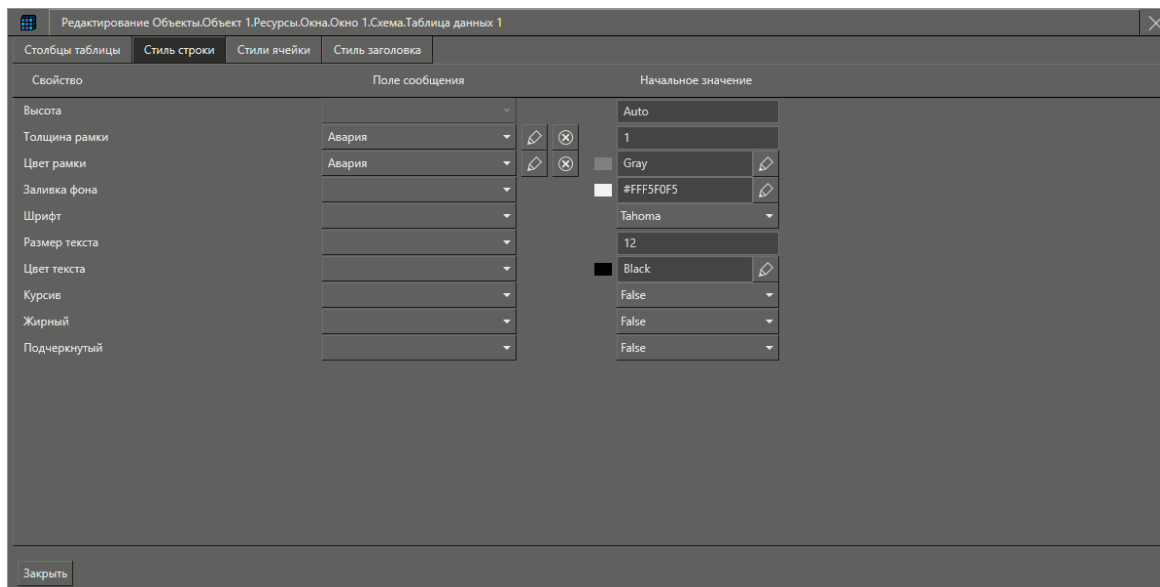


Видимость	Флаг, определяющий видимость столбца. Если флаг снят, то для соответствующего столбца место в таблице не выделяется. Данное свойство имеет смысл динамизировать, используя панель свойств элемента.
Гориз. выравнивание	Определяет выравнивание заголовка и самого столбца.
Стиль ячеек столбца	Позволяет 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.

### 8.3.2.6.2. 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:



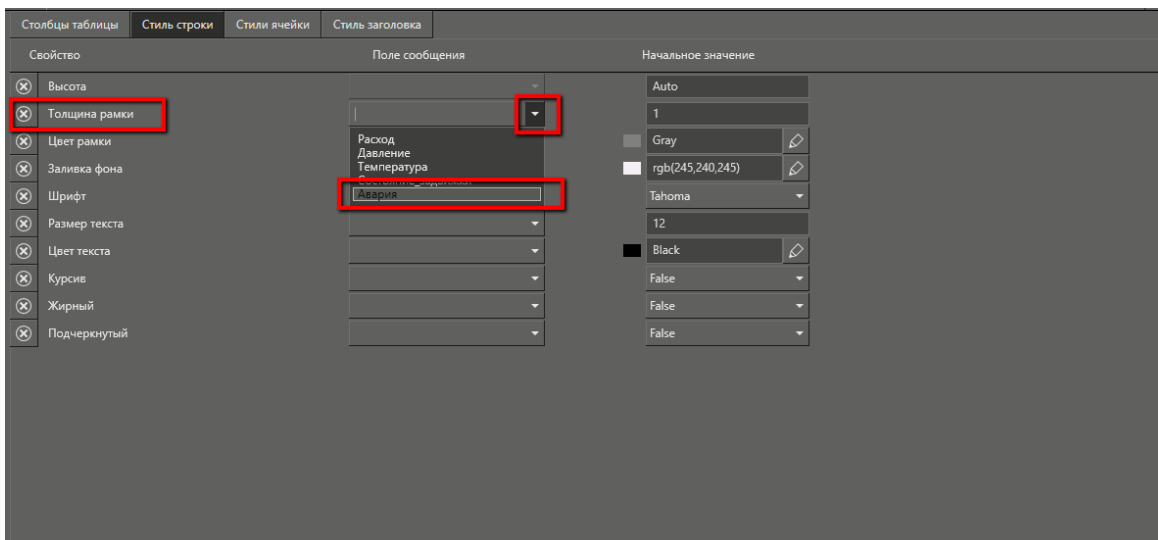
На вкладке находится таблица, состоящая из трех столбцов:

Название	Назначение
Свойство	Название свойства строки.
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.

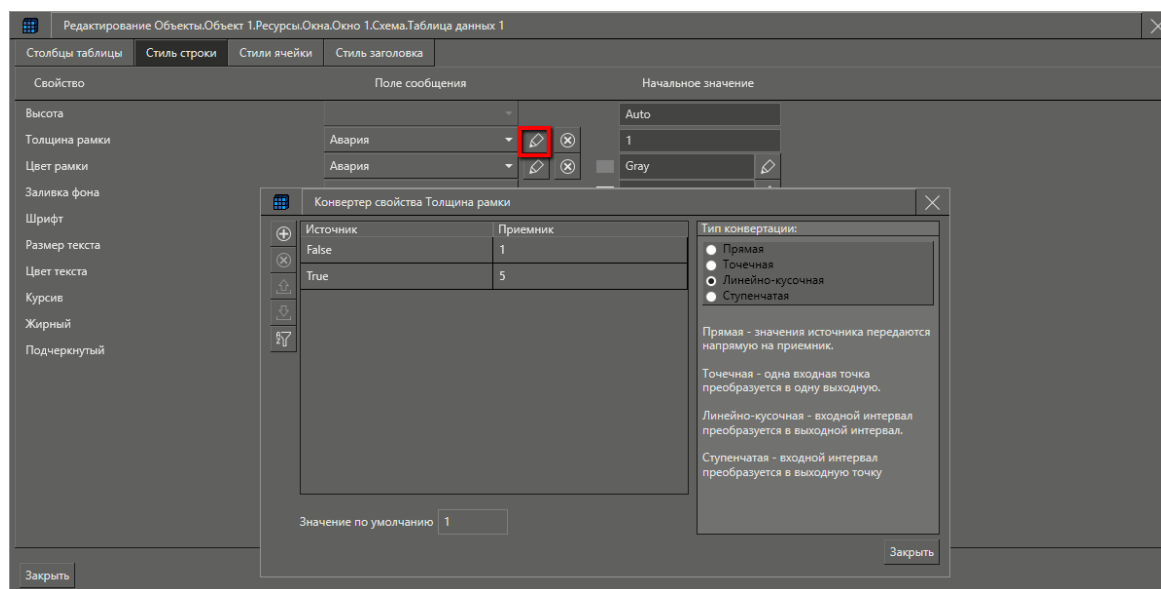
В этом случае требуется настроить свойства Толщина рамки и Цвет рамки. В выпадающем списке Поле сообщения следует выбрать поле Авария:



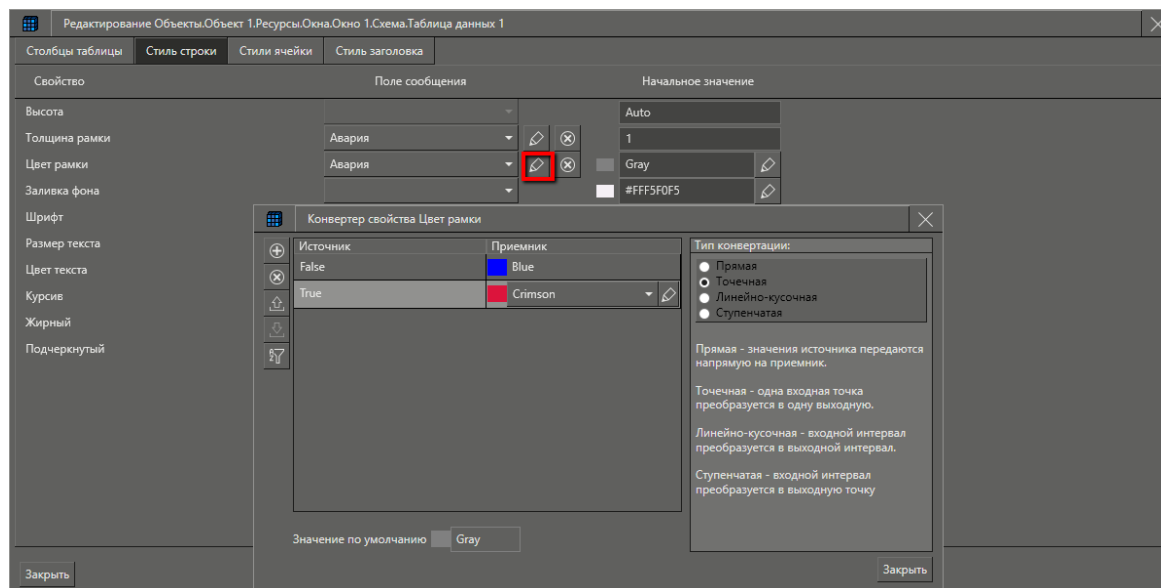
**Важно!** Варианты в выпадающем списке в столбце Поле сообщения зависят от созданного библиотечного элемента Массив структур

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:

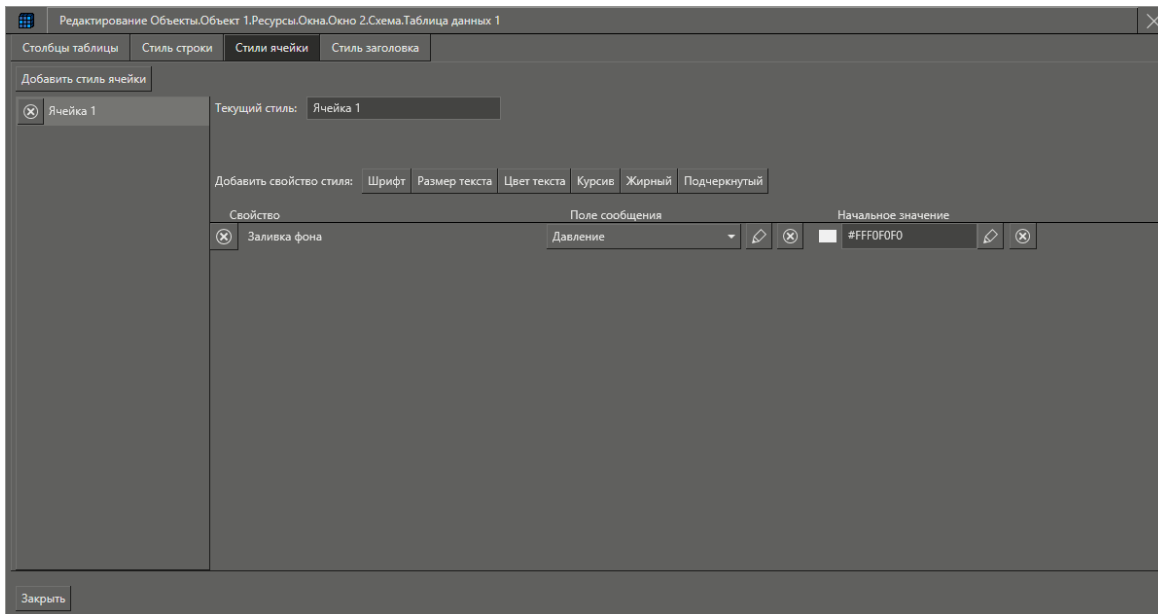



### 8.3.2.6.3. Редактор таблицы данных. Стили ячеек

Вкладка Стили ячеек 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 <b>Добавить свойство стиля</b> , в таблице появится соответствующая строка, отвечающая за то или иное свойство. Нажатие на кнопку  , находящуюся слева от свойства, приведет к его удалению.

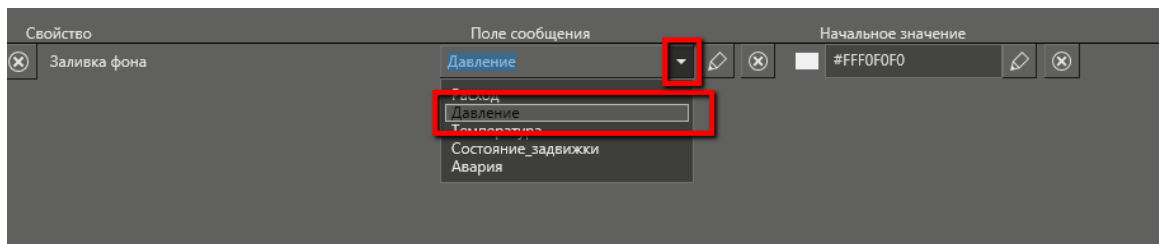
Описание таблицы настроек:

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:

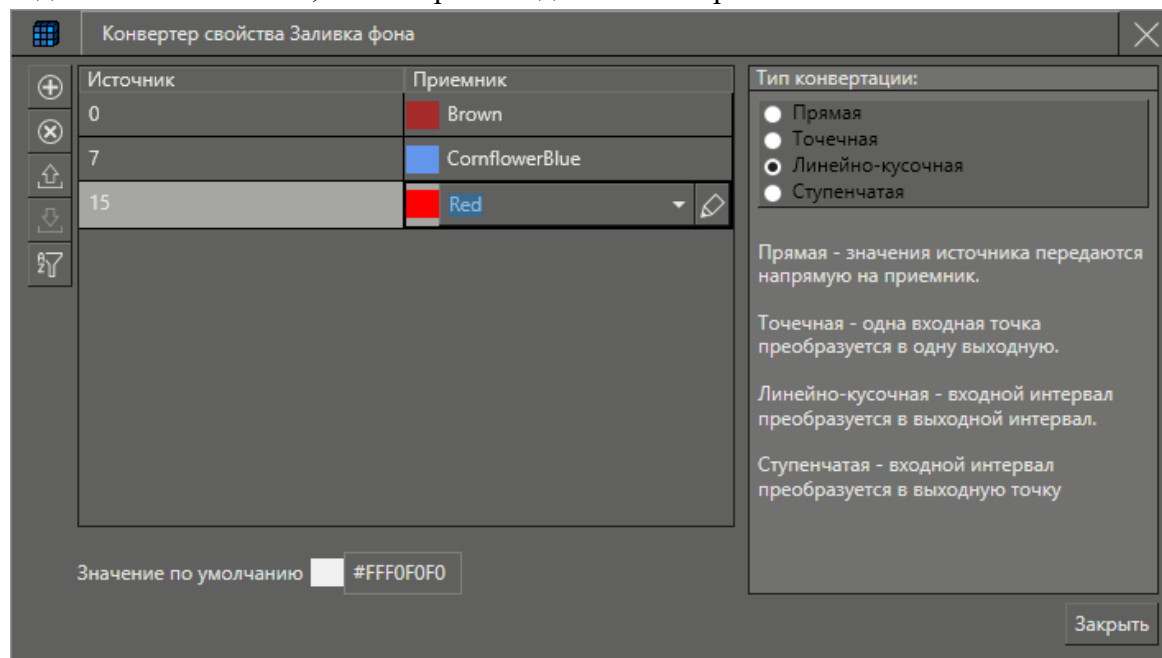


**Важно!** Варианты в выпадающем списке в столбце Поле сообщения зависят от созданного библиотечного элемента Массив структур.

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:

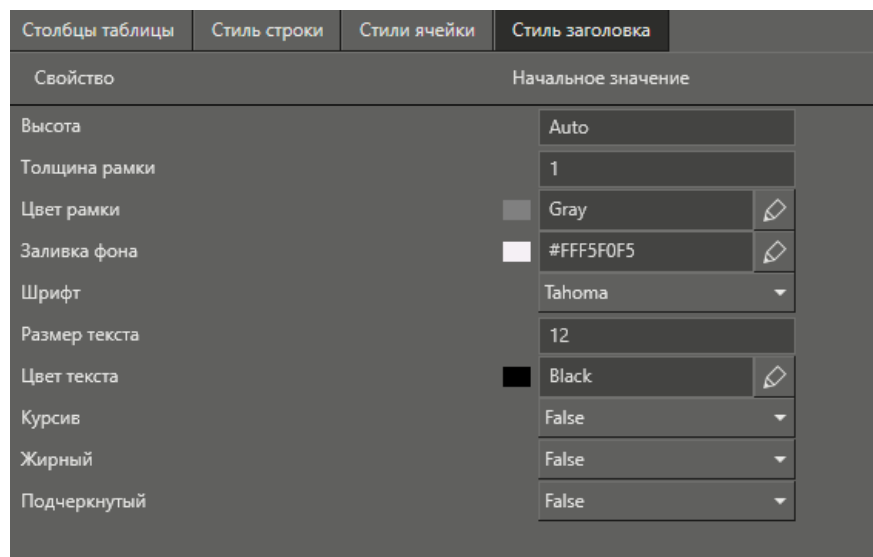


В приведенном примере выбран тип конвертации Линейно-кусочная, а это означает, что в диапазоне значений от 7 до 15 заливка фона ячейки будет голубой, но если значение опустится ниже 7, то заливка фона ячейки станет коричневой, а если повысится выше 15, то красной.

#### 8.3.2.6.4. Редактор таблицы данных. Стиль заголовка

Вкладка Стиль заголовка определяет внешний вид заголовка таблицы данных.

Вкладка имеет вид:

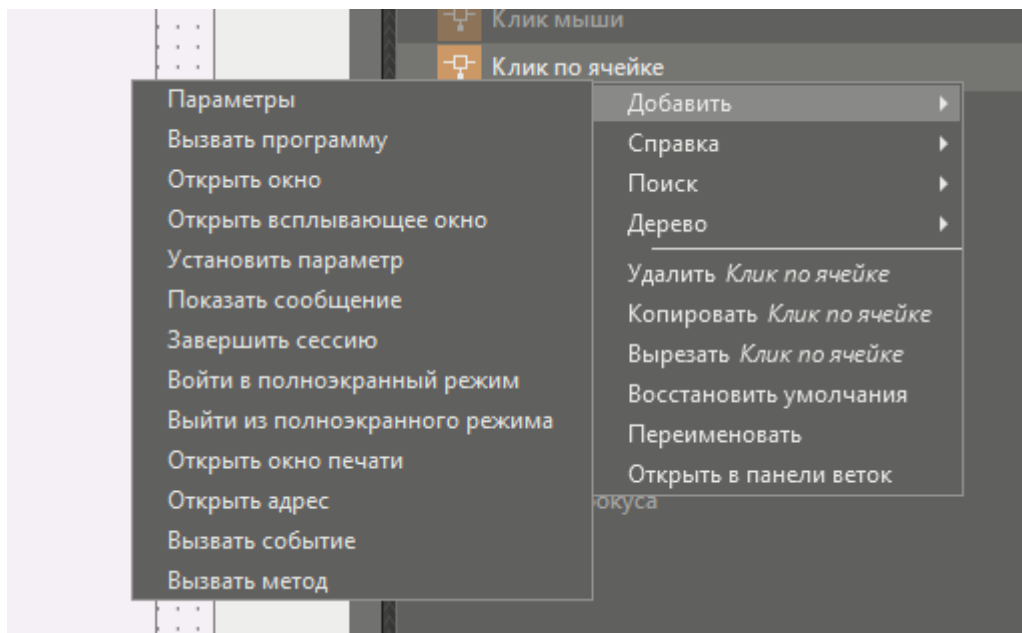


Важно! Стиль заголовка определяется только в диалоговом окне. В панели свойств стиль заголовка не настраивается.

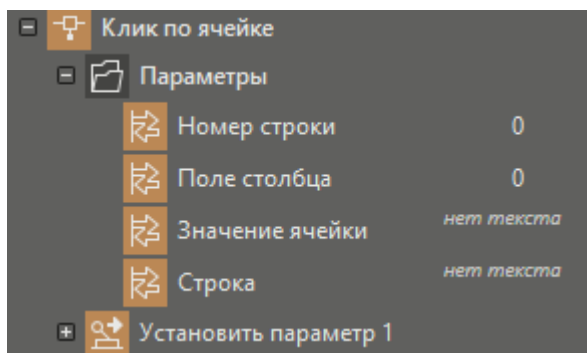
### 8.3.2.6.5. Событие Клик по ячейке

Элемент Таблица данных имеет 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

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

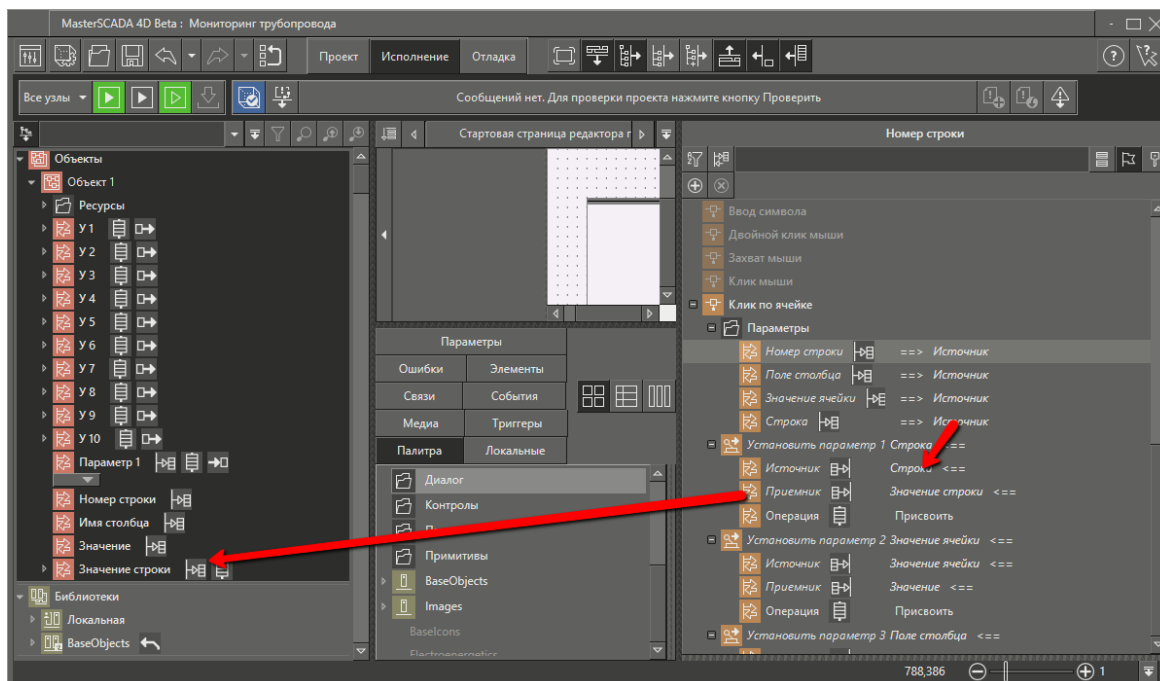
В проекте создаем четыре параметра:

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:

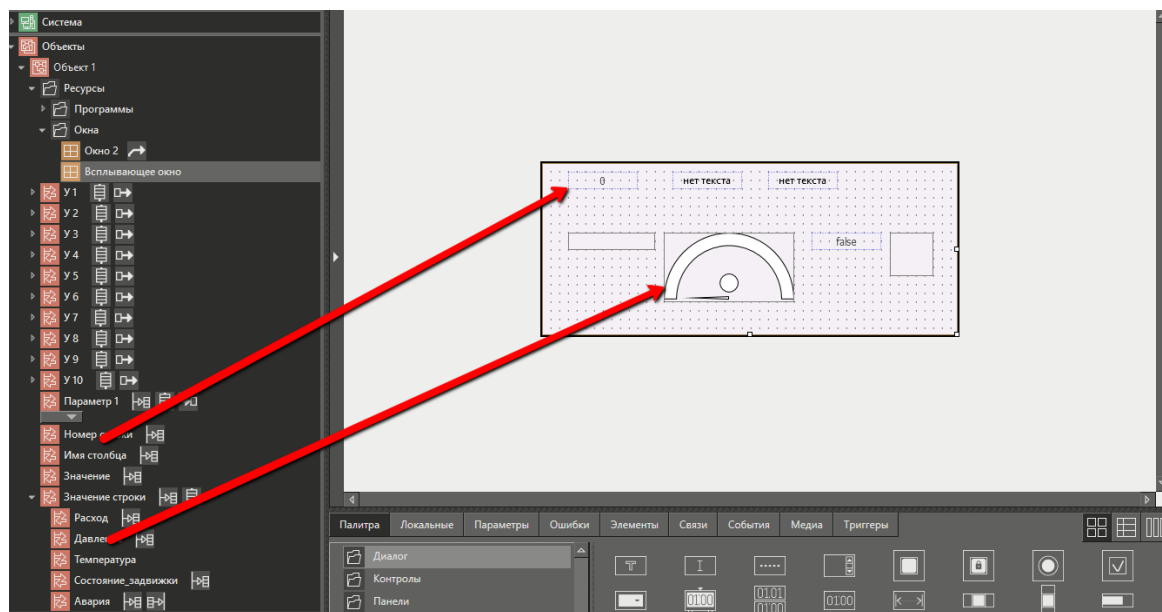




### Создание всплывающего окна

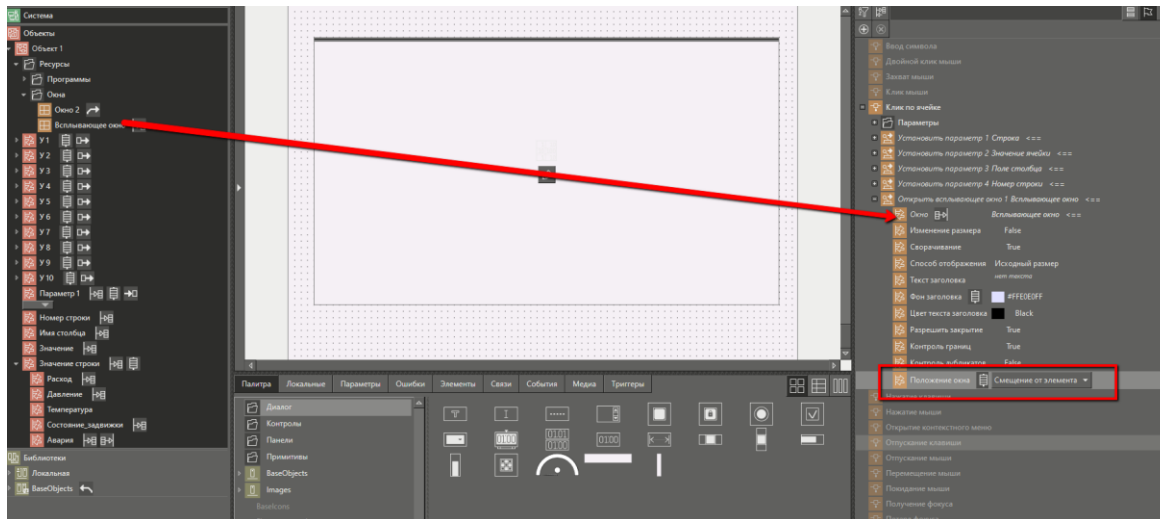
Создаем окно, которое должно открываться при нажатии на ячейку. Пусть оно называется Всплывающее окно.

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

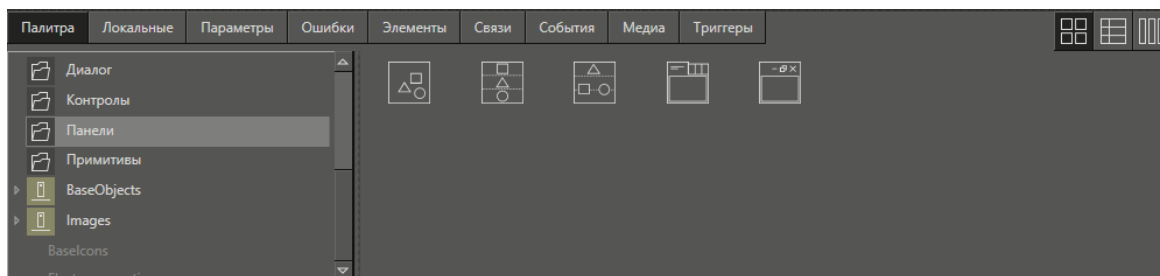
Для элемента Таблица данных добавляем действие Открыть всплывающее окно в событие Клик ячейки. Параметр действия Окно нужно связать с созданным окном. Если установить значение параметра Положение окна Смещение от элемента, то окно будет открываться рядом с ячейкой, по которой был выполнен клик:



## 8.3.3. 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

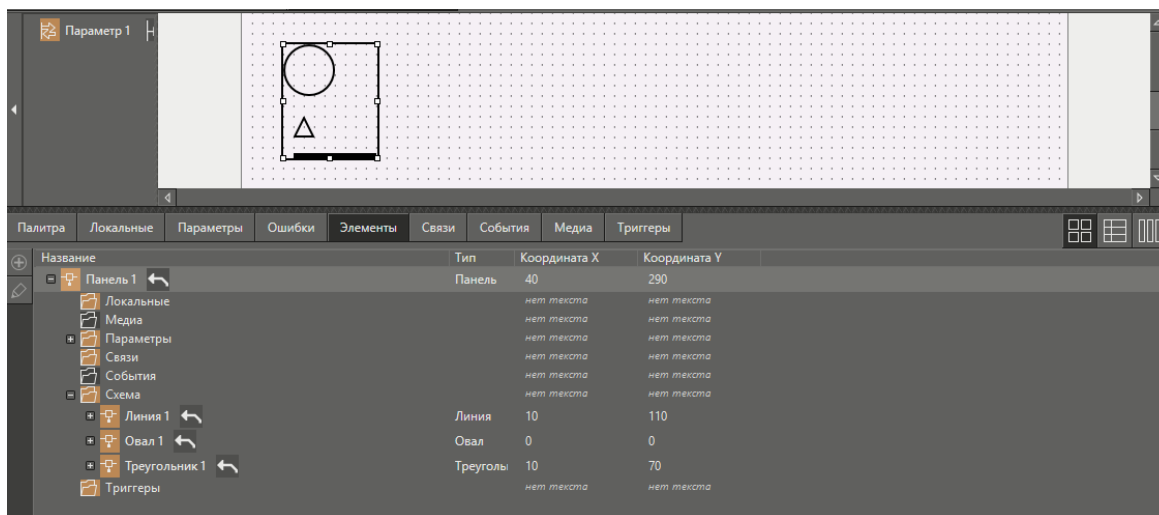
### 8.3.3.1. Общие инструменты панелей

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 НМІ 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.

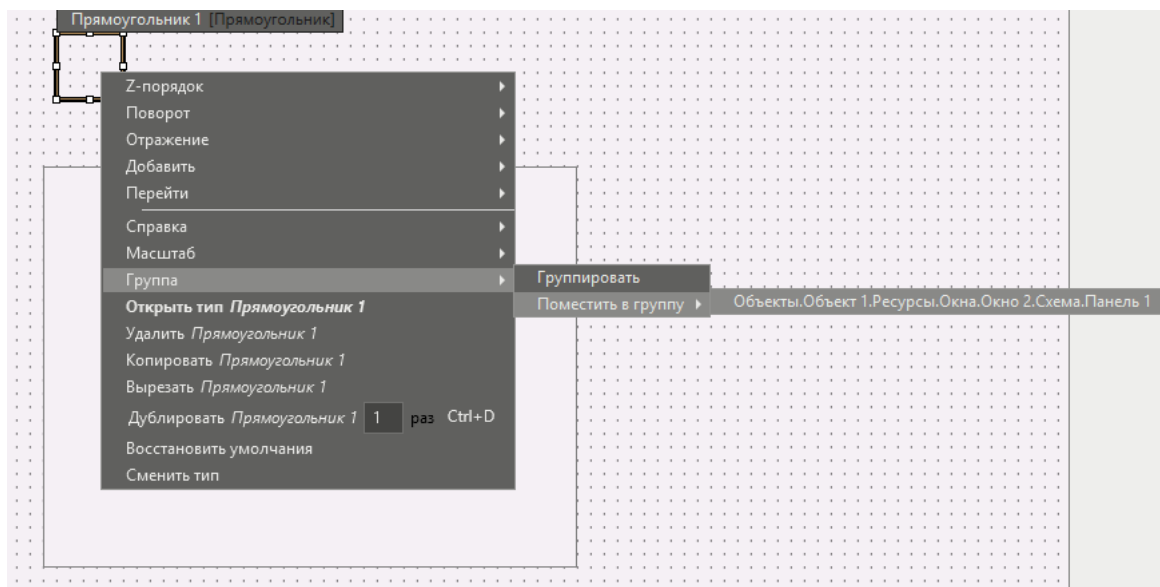
**Важно!** Изменение размеров панели не приводит к изменению размеров элементов размещенных в панели. Если необходимо, чтобы размеры элементов изменялись пропорционально изменяемой области, то элементы следует разместить в отдельном окне, а затем это окно перетащить в рабочую область редактора НМІ.

В палитре редактора НМІ элемент **Панель** находится в категории **Панели**.

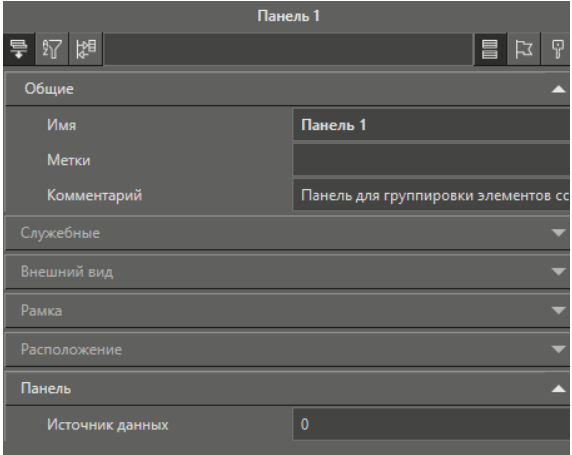
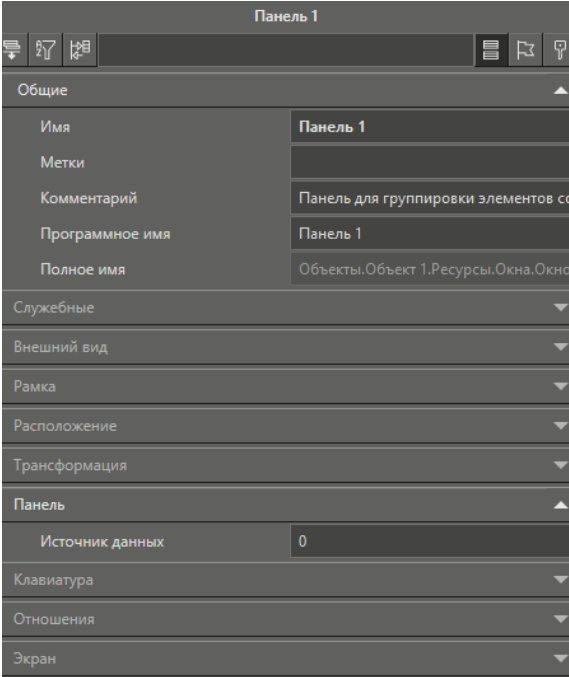
Вид элемента в палитре:



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]:



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:

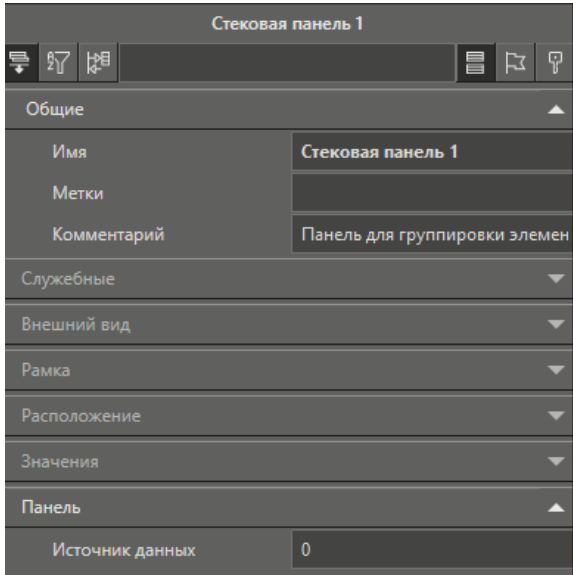
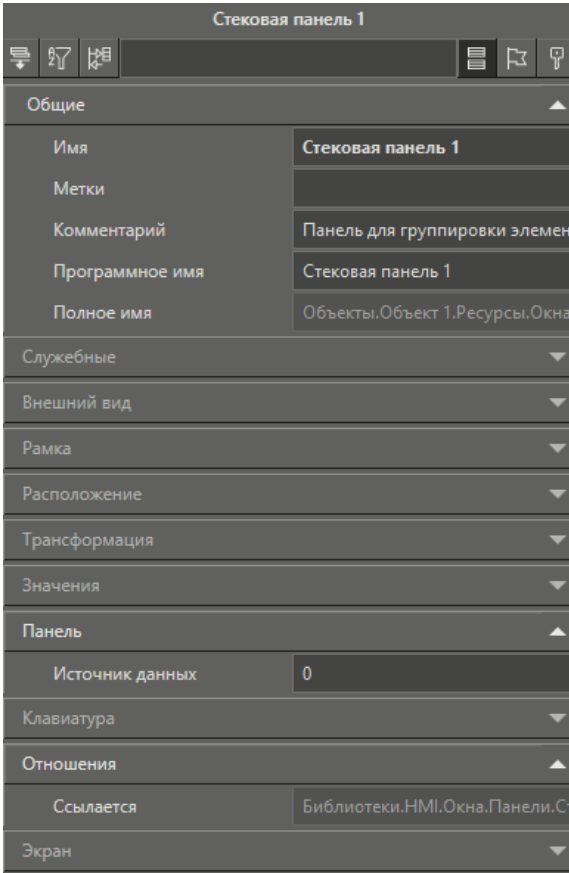


Вид элемента после добавления в рабочую область редактора НМІ:



Работа с элементом начинается с настройки его свойства Источник данных.

Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
	

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"> <li>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 размножаются, количество строк будет совпадать с количеством элементов массива.</li> </ul>

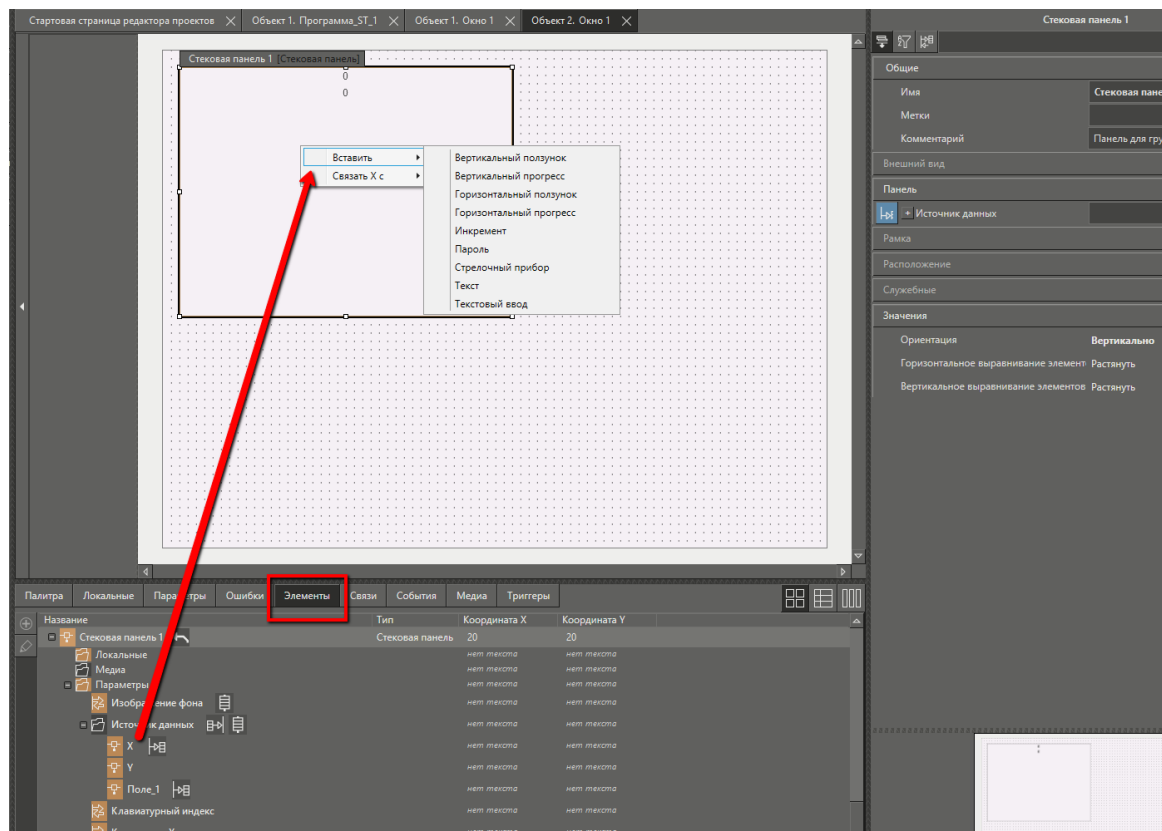
	<ul style="list-style-type: none"> <li>Вертикально - размещаемые на панели элементы последовательно (по порядку размещения) располагаются на панели вертикально сверху вниз, и весь набор размещенных элементов прикрепляется к верхней границе панели, образуя столбец. В режиме исполнения элементы размножаются, количество строк будет совпадать с количеством элементов массива.</li> </ul>
Горизонтальное выравнивание	Определяет поведение итоговой таблицы элементов в панели в режиме исполнения, если размер итоговой таблицы не будет совпадать с размерами панели.
Вертикальное выравнивание	
Категория Панель	
Источник данных	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

Описание других свойств элемента смотрите в разделе НМІ.Свойства

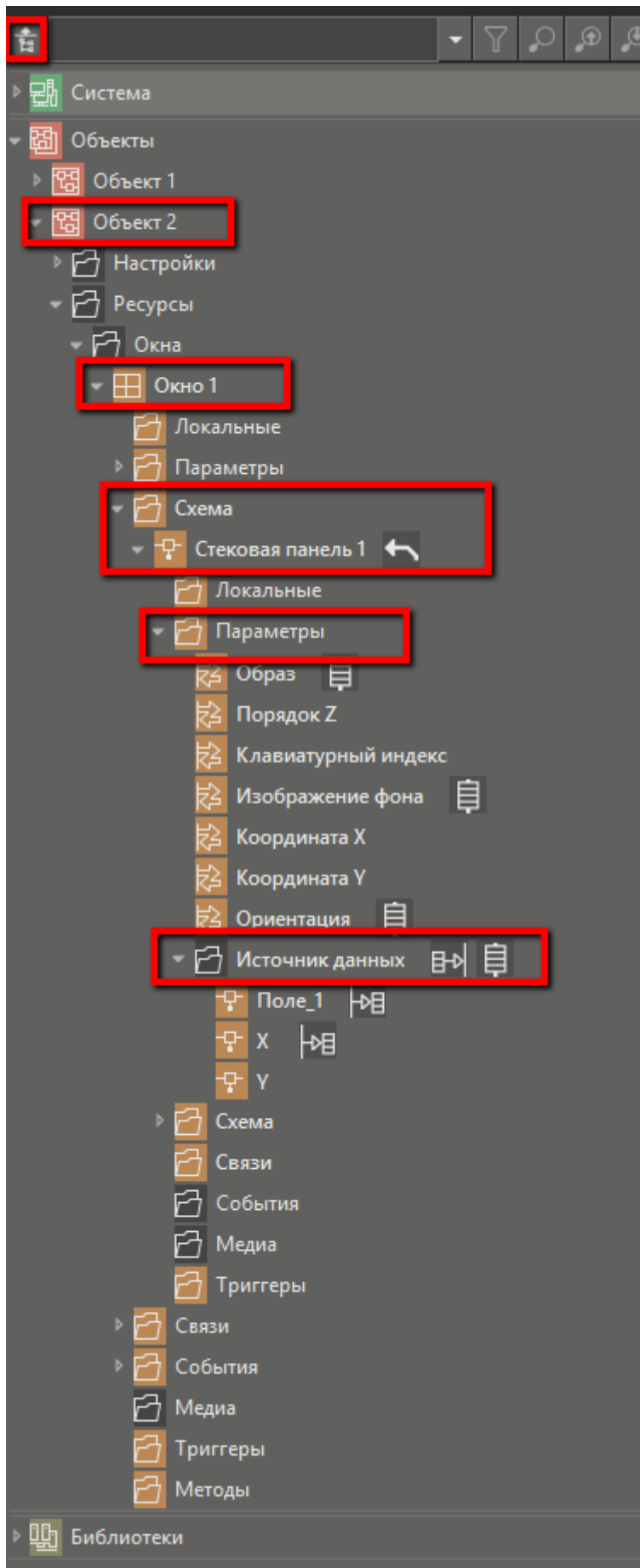
#### Порядок работы с элементом Стековая панель

1. Элемент добавляется в рабочую область редактора НМІ. 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 перетасканы 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 НМИ 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.

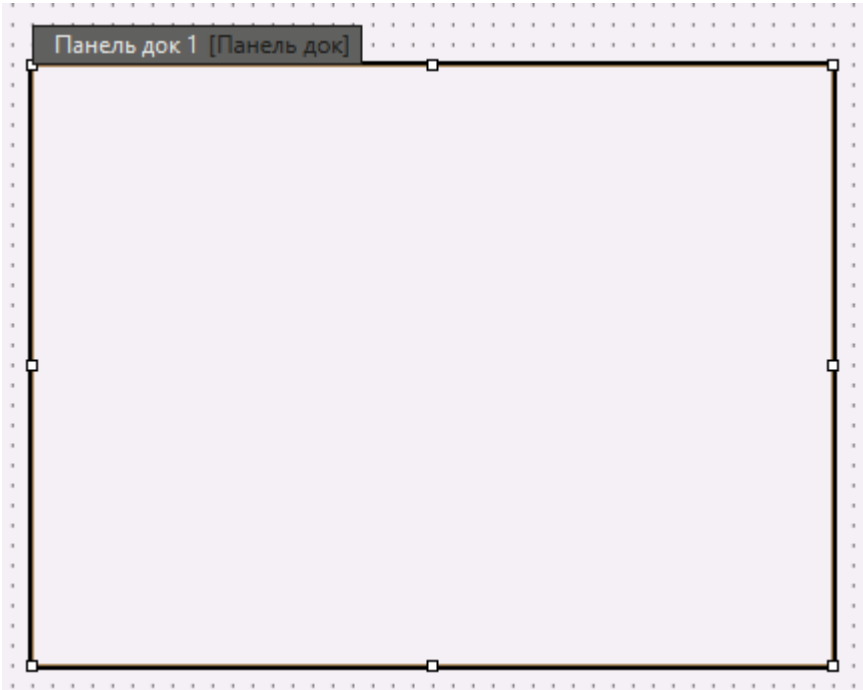
Важно! Данный элемент поддерживается только в НМІ v1.

В палитре редактора НМІ элемент Панель док находится в категории Панели.

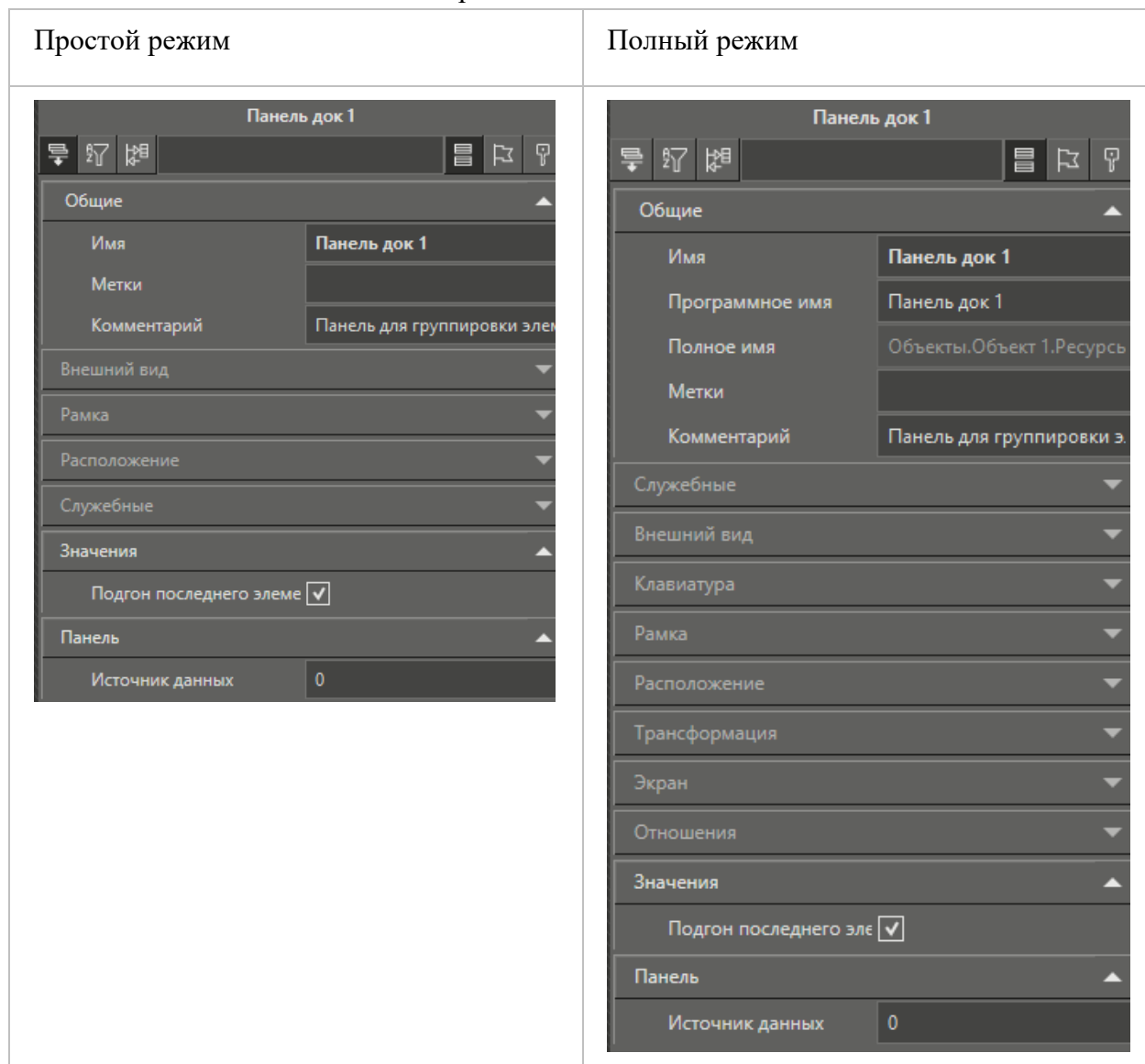
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора НМІ:



Вид панели свойств элемента при нажатой и отжатой кнопке :



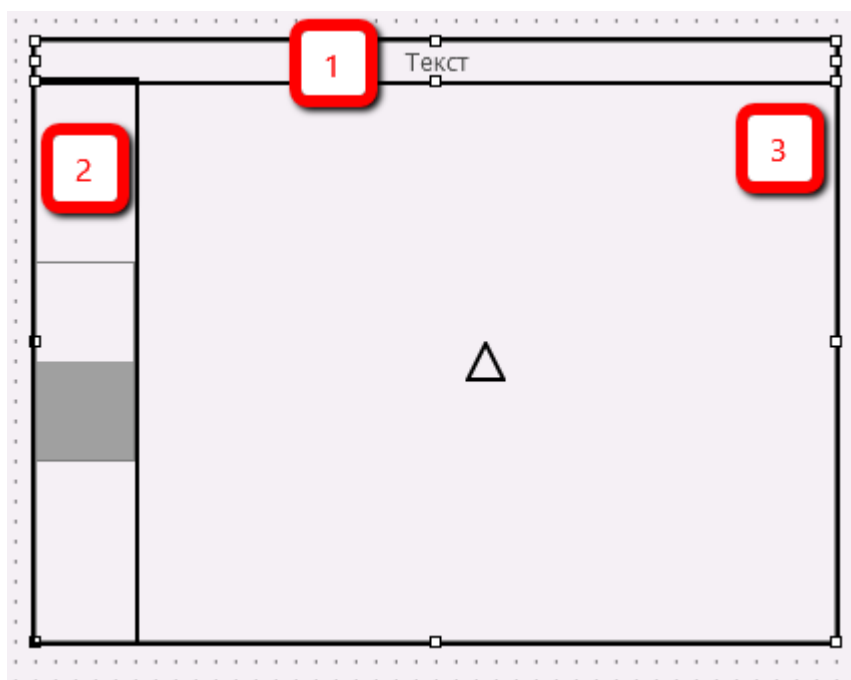
Описание основных свойств графического элемента Панель:

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	

Источник данных	Не используется.
-----------------	------------------

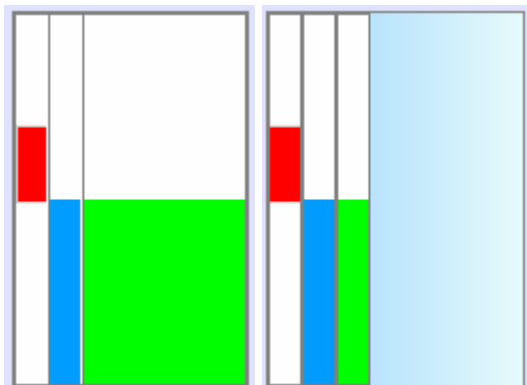
### Правила размещения элементов в панели

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

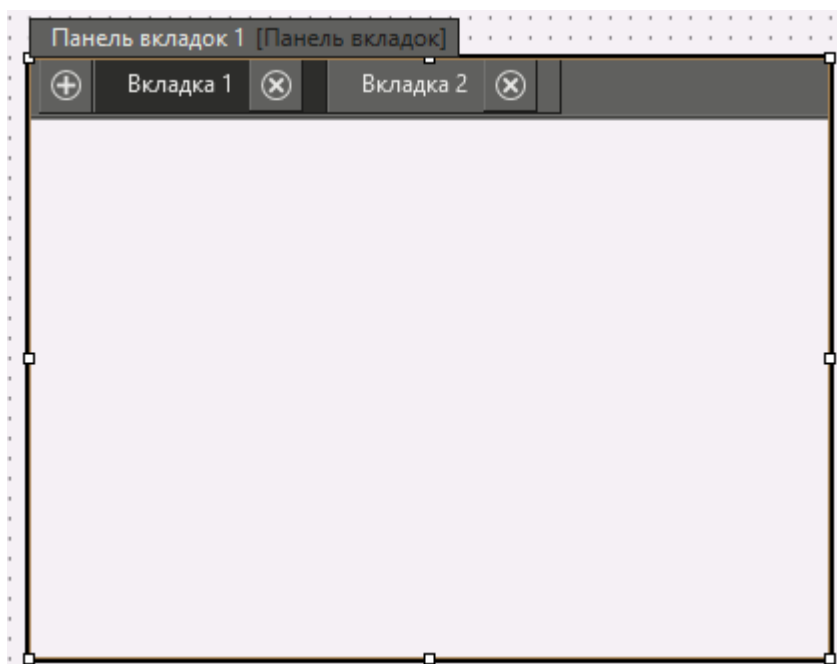
Данный элемент представляет собой набор вкладок и предназначен для объединения в группы других элементов.

В палитре редактора HMI элемент Панель вкладок находится в категории Панели.

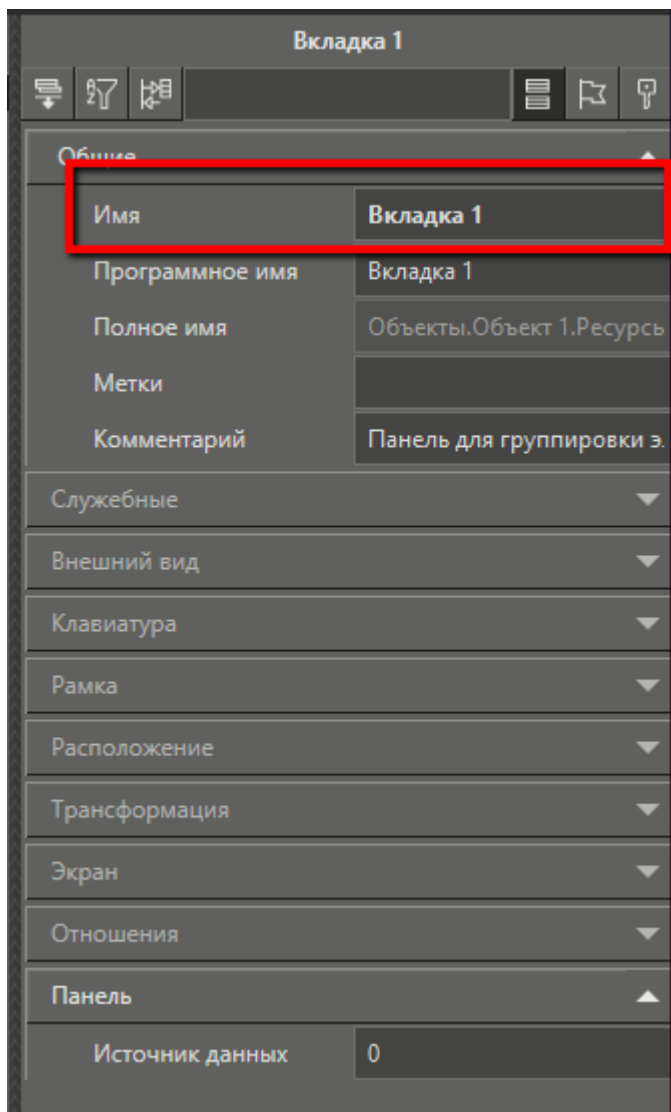
Вид элемента в палитре:





Вид элемента после добавления в рабочую область редактора HMI:



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:

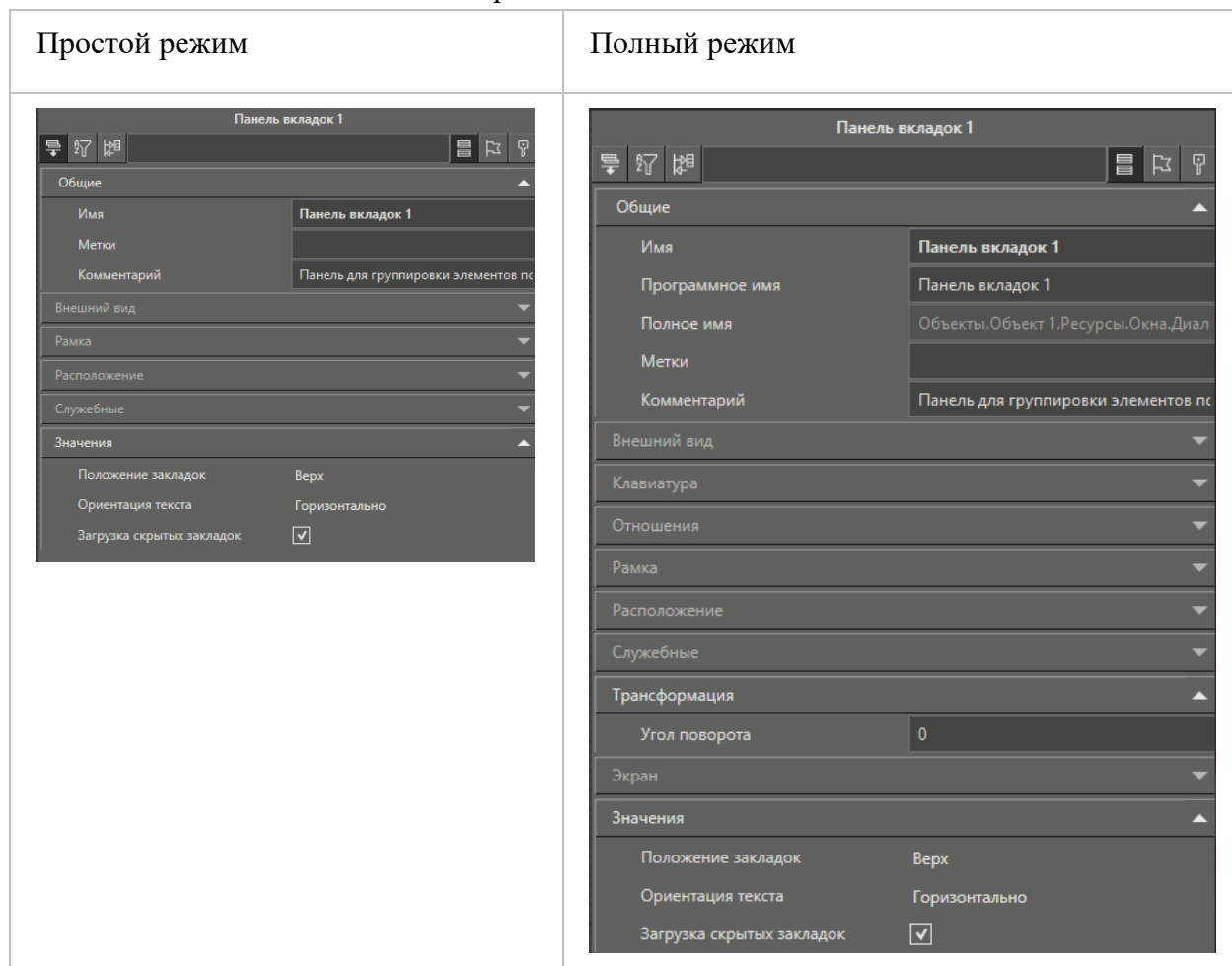


Для добавления вкладки следует нажать на кнопку , а для удаления ранее созданной вкладки - на кнопку .

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 :



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

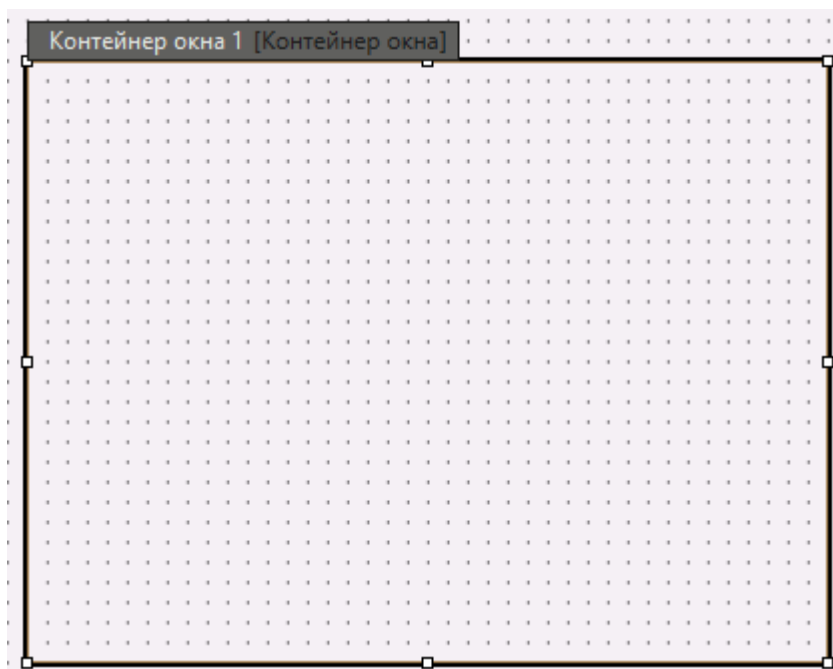
Данный элемент предназначен для отображения содержимого различных окон или страниц сайтов.

В палитре редактора HMI элемент Контейнер окна находится в категории Панели.

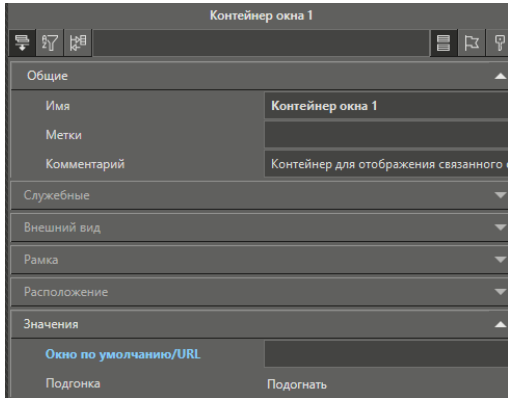
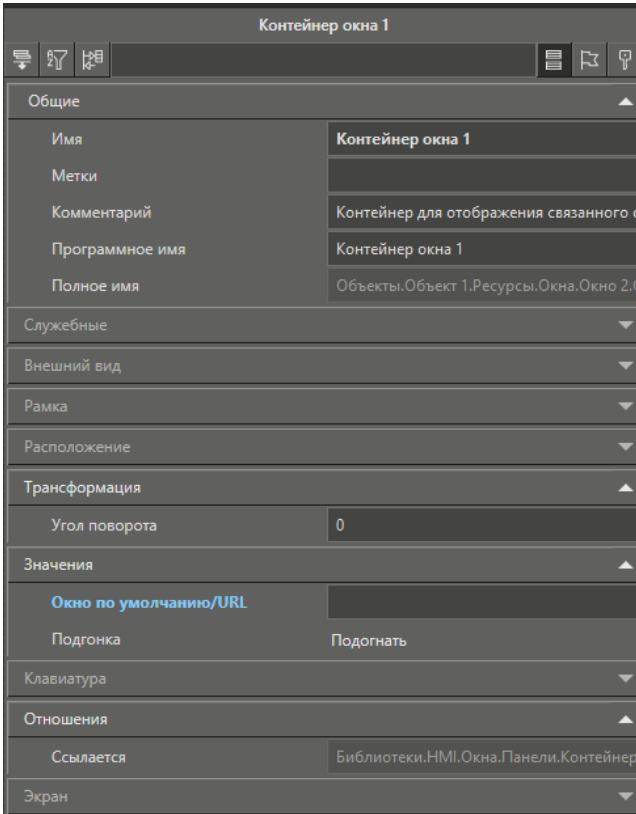
Вид элемента в палитре:



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
Категория Значения	
Окно по умолчанию/ URL	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).
Подгонка	Определяет, каким образом будет отображаться содержимое окна или страницы сайта в случае, если их размер отличается от размера контейнера. .  Возможны следующие значения:

	<ul style="list-style-type: none"><li>• Исходный размер - не используется;</li><li>• Подогнать - отображаемое содержимое масштабируется под размеры контейнера;</li><li>• Обрезать - в случае превышения размеров контейнера, содержимое при отображении будет совмещено с левым верхним углом контейнера и обрезано снизу и справа;</li><li>• Прокрутить - если содержимое превышает размеры контейнера, то появляются линейки прокрутки.</li></ul>
--	--

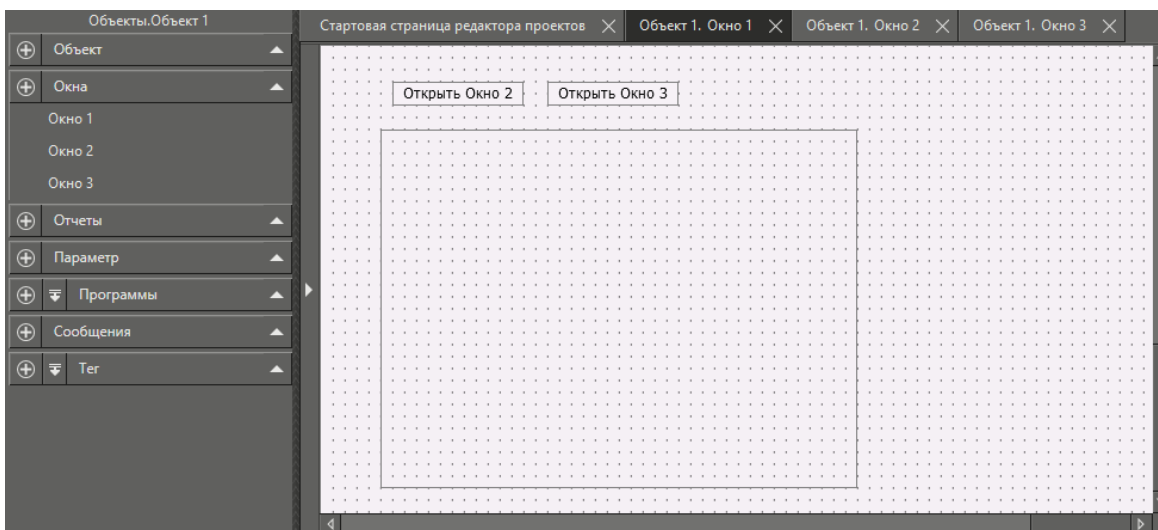
Описание других свойств элемента смотрите в разделе НМІ.Свойства

## Рекомендации

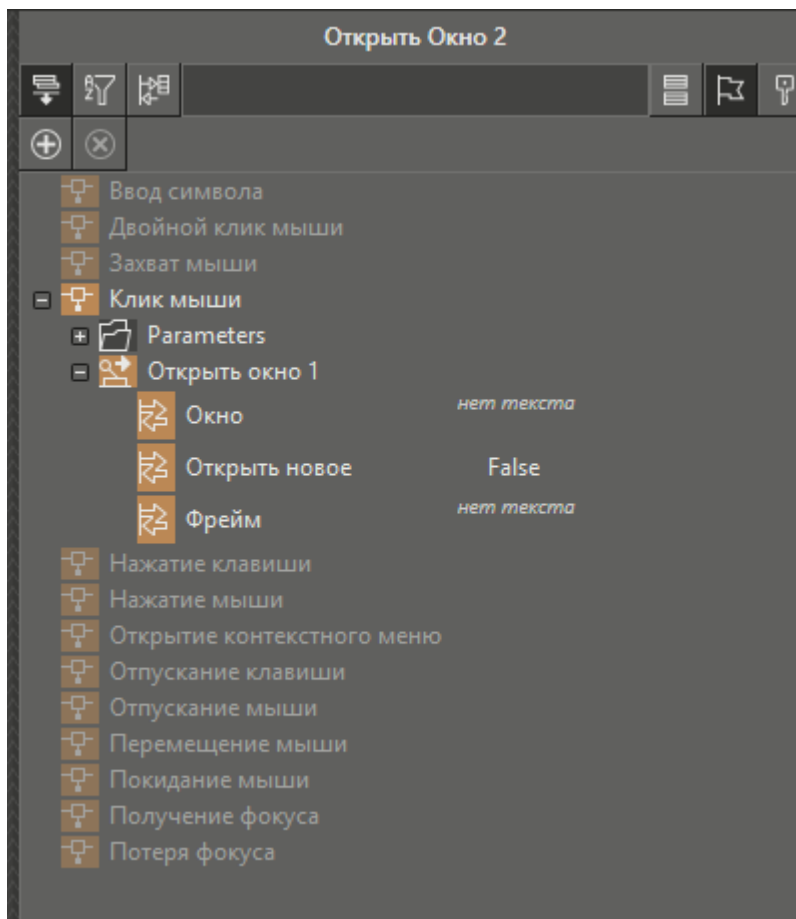
### Отображение различных окон в одном контейнере

Рассмотрим пример, когда требуется при нажатии на кнопки открывать разные окна в одном контейнере.

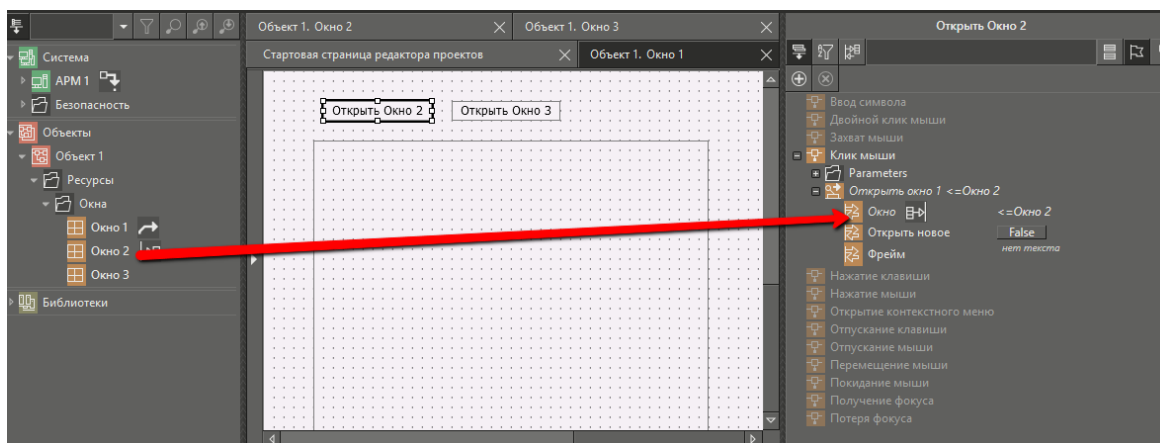
Допустим имеется три окна: Окно 1 - основное стартовое окно, которое содержит элемент Контейнер окна и две кнопки. Окно 2 и Окно 3 должны открываться при нажатии на эти кнопки.



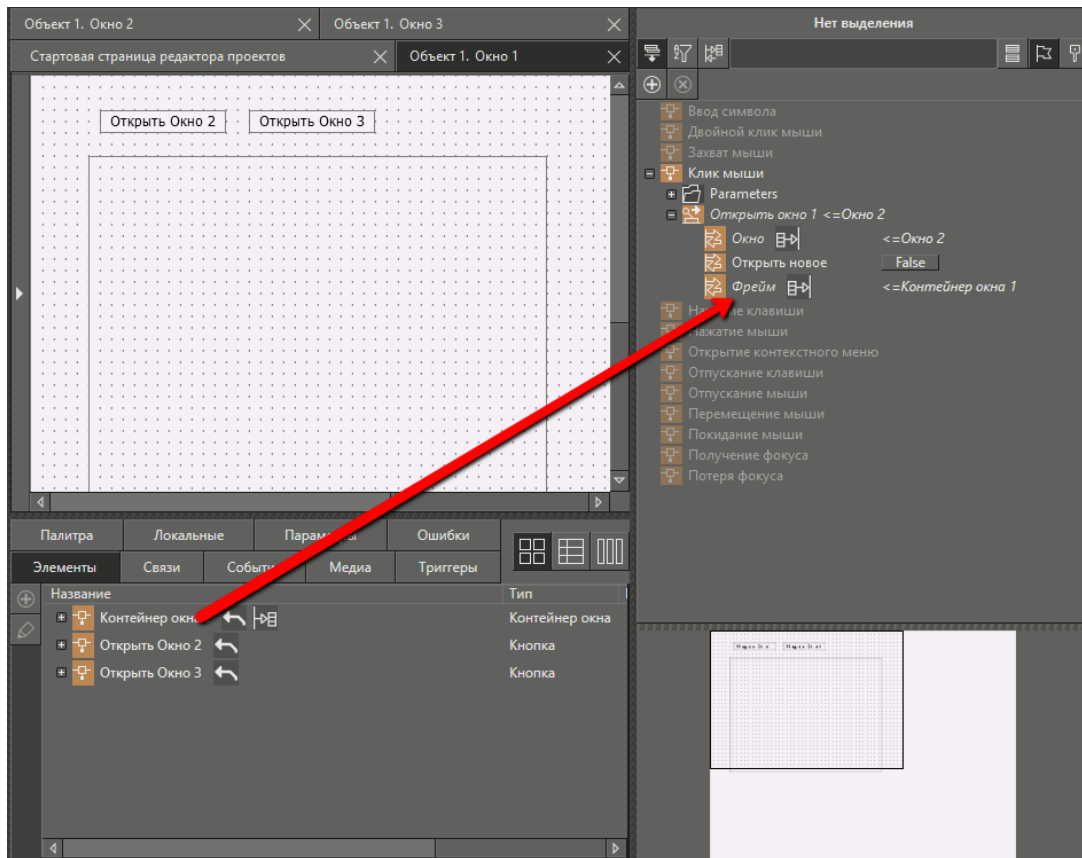
Выделим кнопку Открыть окно 2. Перейдем в ее панель свойств, переключимся в режим работы с событиями. И добавим в событие Клик мыши действие Открыть окно:



Затем перетащим левой кнопкой мыши Окно 2 из дерева в параметр действия Окно:



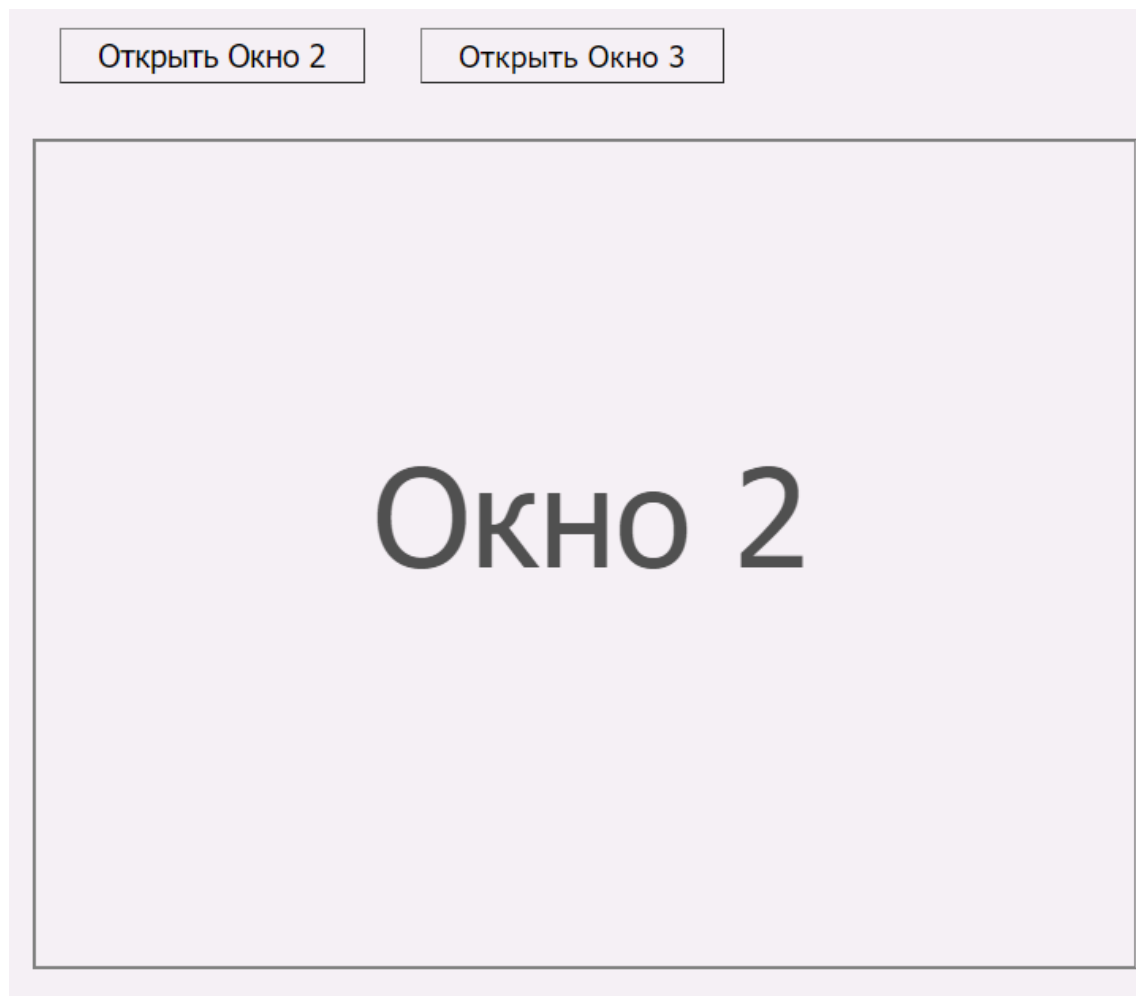
Откроем вкладку Элементы в легенде редактора НМІ. Перетащим элемент Контейнер окна на параметр Фрейм действия:



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

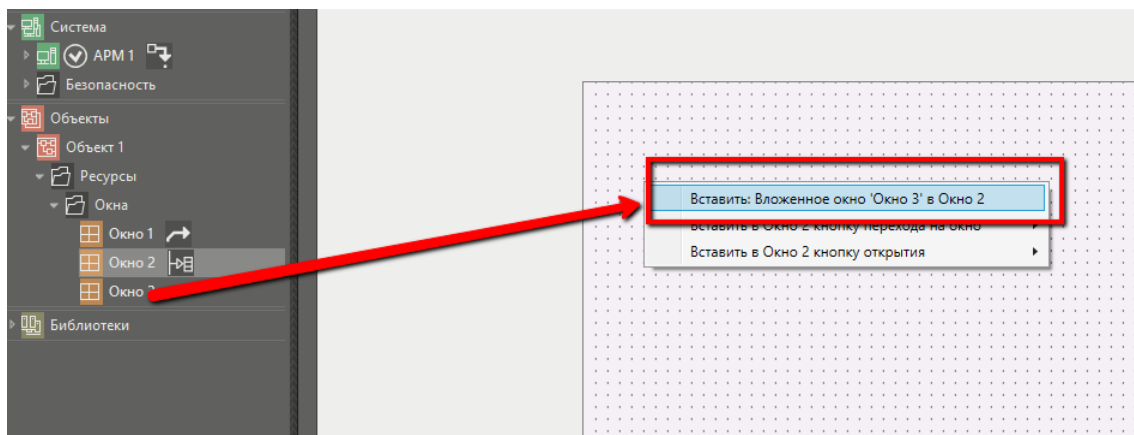


#### Отображение различных сайтов в одном контейнере

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

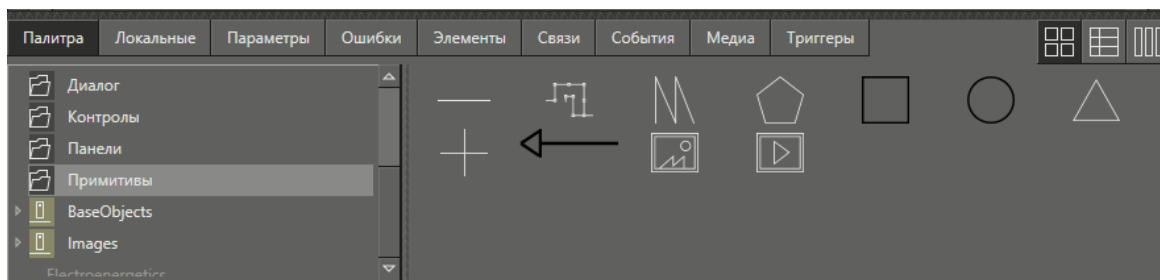
Если необходимо, чтобы в одном окне всегда отображалось одно и то же другое окно, то можно использовать контейнер окна, перетащив требуемое окно в свойство Окно по умолчанию/URL. Однако можно просто перетащить левой кнопкой мыши окно, требующее отображения, в нужное место другого окна. И в появившемся списке выбрать:



### 8.3.4. Категория Прimitives

Категория Прimitives палитры редактора HMI 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 HMI 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

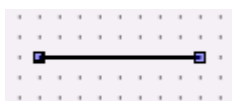
Данный элемент предназначен для рисования прямых линий.


В палитре редактора HMI элемент Линия находится в категории Прimitives.

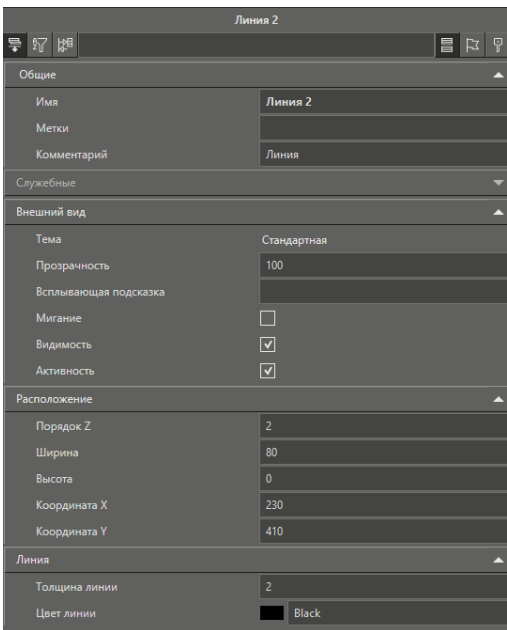
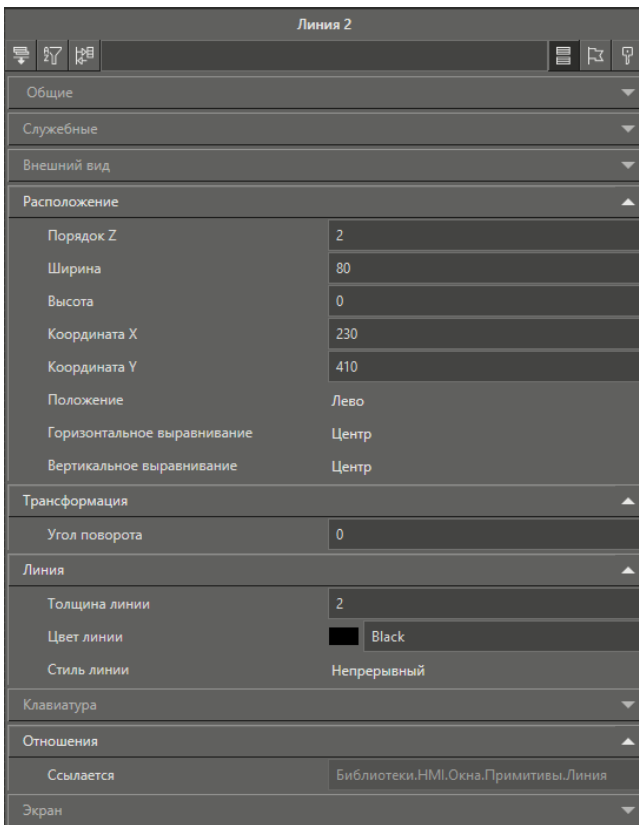
Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:



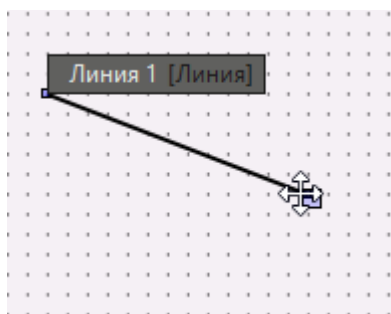
Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
	

Описание свойств элемента смотрите в разделе HMI.Свойства

Рекомендации

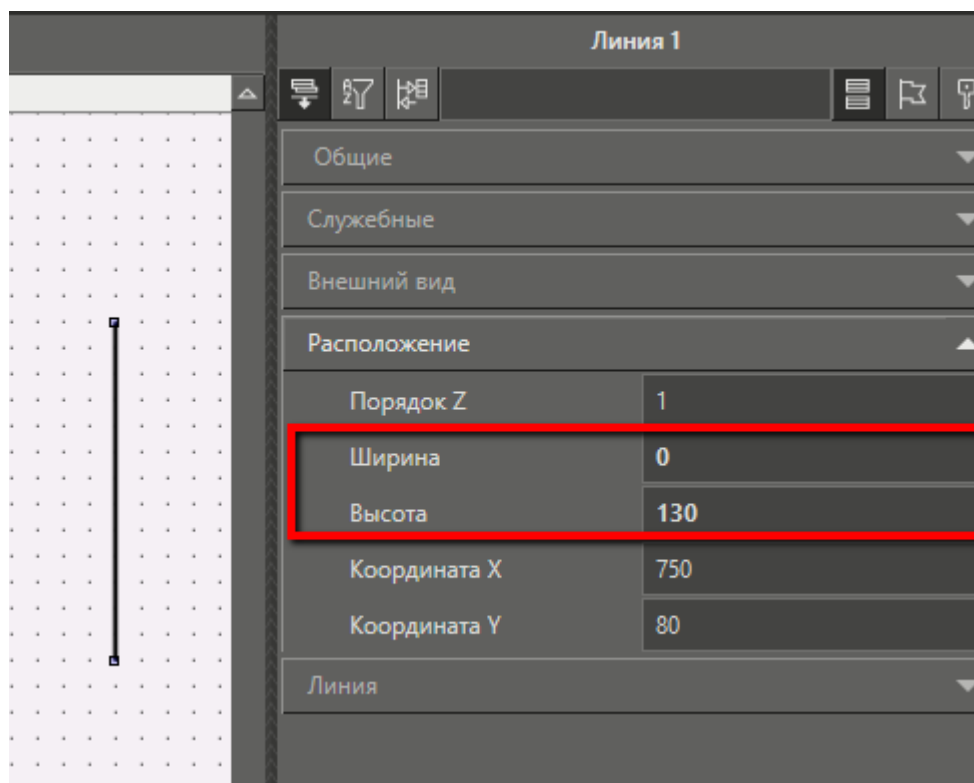
- Для того чтобы изменить угол наклона можно потянуть начало или конец линии левой кнопкой мыши. При этом указатель мыши будет изменен:



- 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

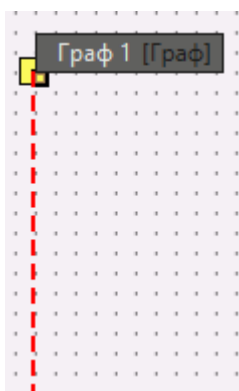
Данный элемент предназначен для рисования ломаных линий с ответвлениями, представляет собой SVG-фигуру.

В палитре редактора НМІ элемент Граф находится в категории Прimitives.

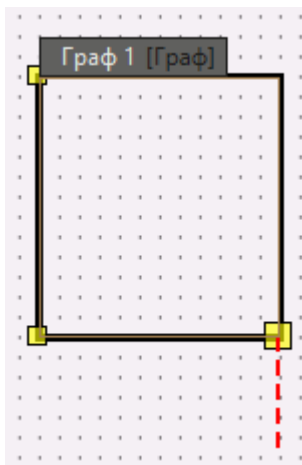
Вид элемента в палитре:



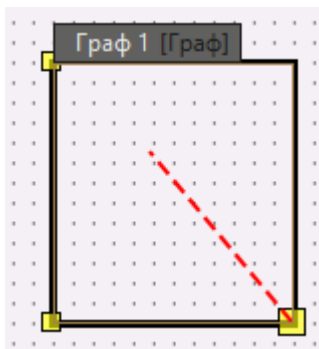
После добавления элемента в рабочую область редактора НМІ граф будет находиться в режиме редактирования, в котором необходимо задать его конфигурацию. Сначала появится первая точка, из которой будет выходить пунктирная линия, направление которой будет следовать за движением мыши:



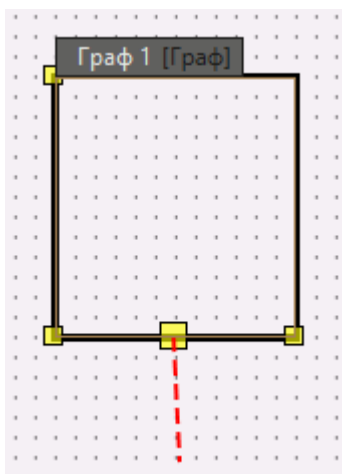
Если нажать левой кнопкой мыши на какое-либо место рабочей области, то будет создана вторая точка. Если переместить мышь и опять нажать левую кнопку, то будет создана третья точка:



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).

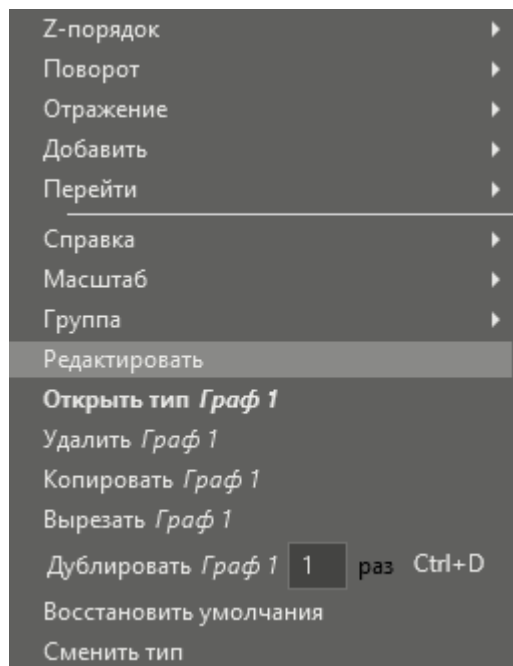


Для удаления существующей точки, нужно дважды нажать на неё левой кнопкой мыши, после чего точка удалится, а граф перерисуеться.

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.

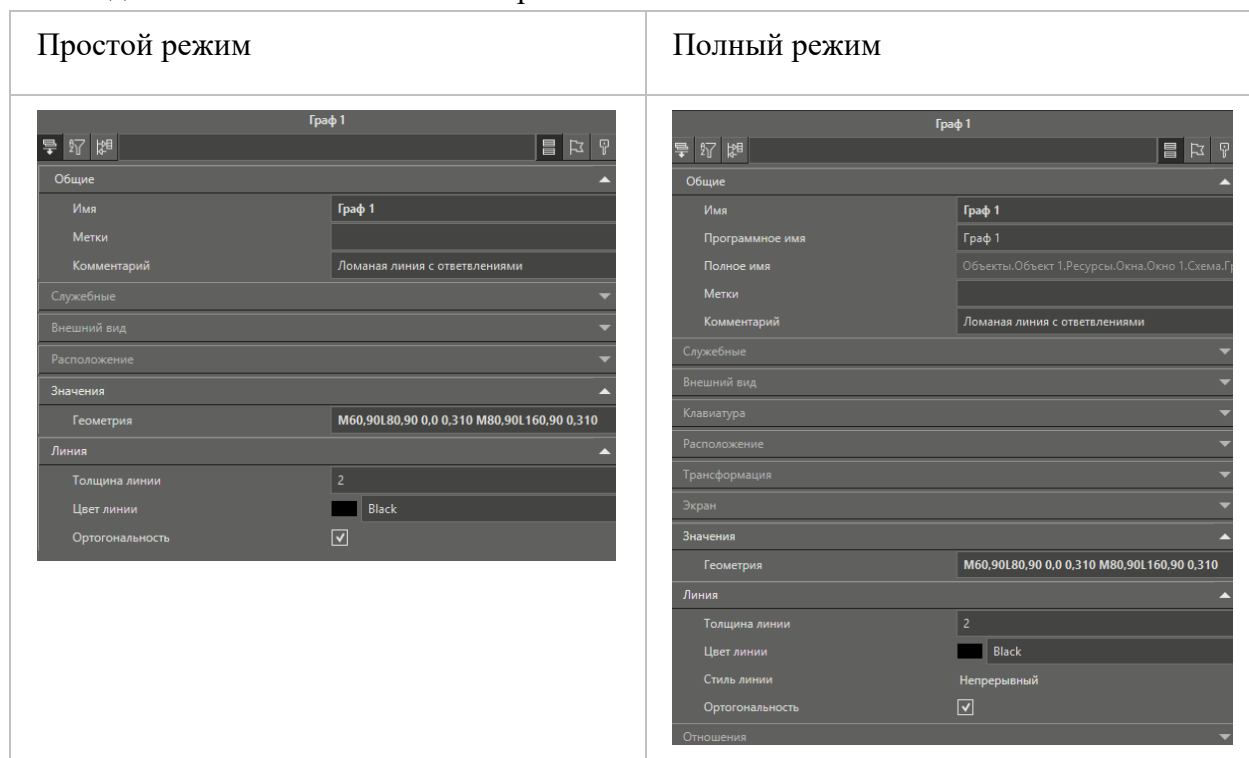
Для того чтобы войти в режим редактирования ранее нарисованного графа, необходимо нажать правую кнопку мыши, и в контекстном меню выбрать пункт Редактировать:



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

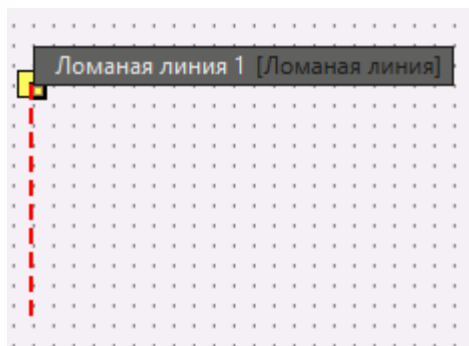
Данный элемент предназначен для рисования ломаных линий, представляет собой SVG-фигуру.

В палитре редактора НМІ элемент Ломаная линия находится в категории Прimitives.

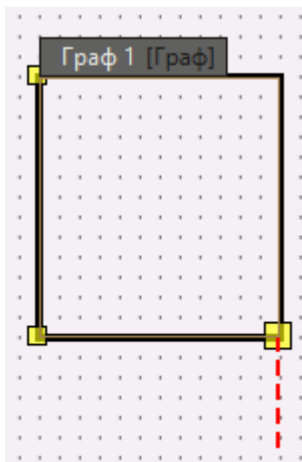
Вид элемента в палитре:



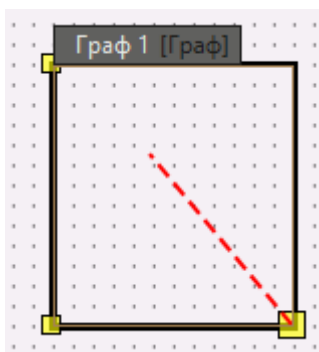
После добавления элемента в рабочую область редактора НМІ ломаная линия будет находиться в режиме редактирования, в котором необходимо задать её конфигурацию. Сначала появится первая точка и из нее будет выходить пунктирная линия, направление которой будет следовать за движением мыши:



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:

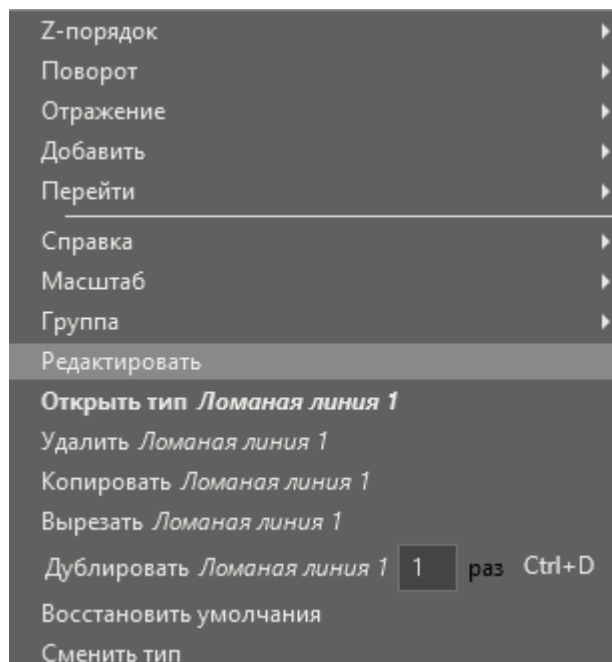


По умолчанию, флаг в свойстве Ортогональность отмечен, поэтому линия будет рисоваться под прямым углом. Это означает, что новая точка будет создаваться не точно в месте нажатия левой кнопки мыши, а максимально близко к нему, так чтобы получился прямой угол. В случае снятого флага Ортогональность новая 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:



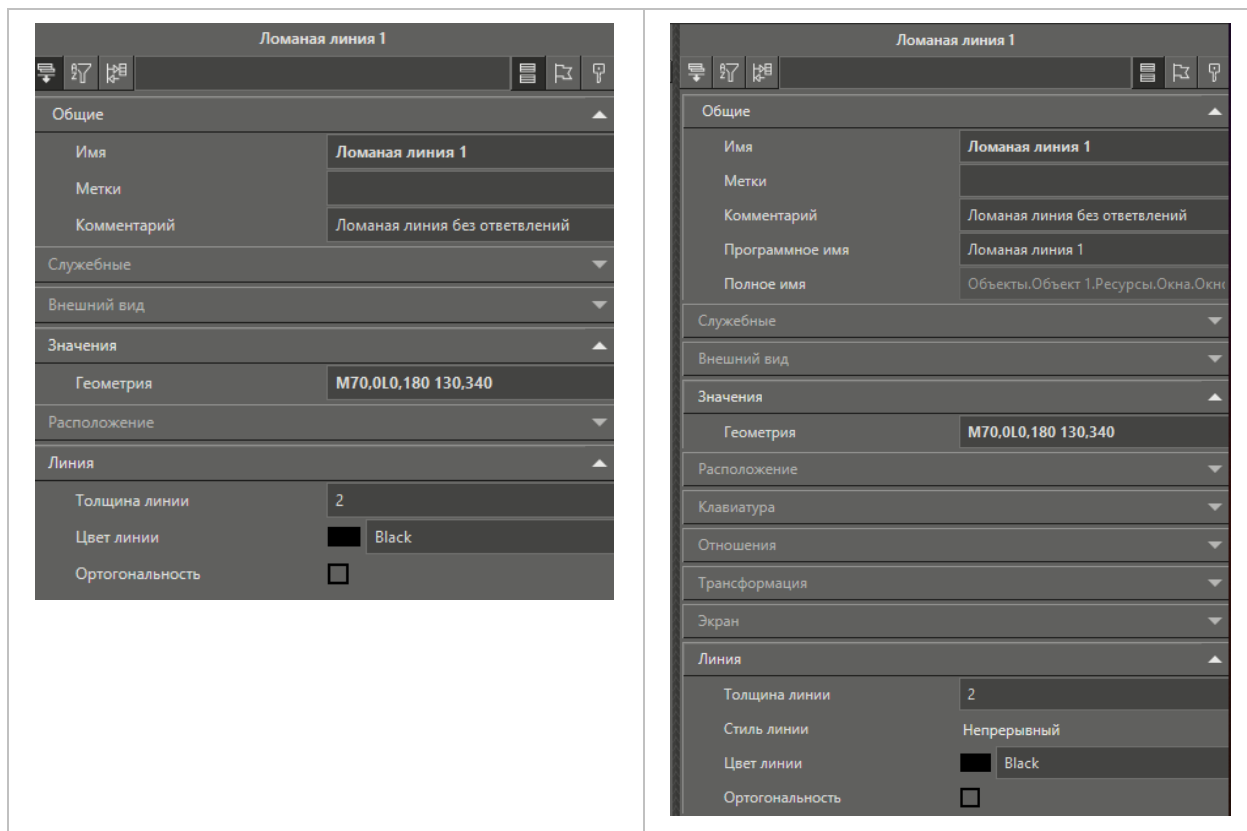
Для того чтобы добавить новую точку, необходимо дважды нажать на линию левой кнопкой мыши. В месте нажатия кнопки будет создан дополнительный излом линии (точка). Чтобы удалить точку, нужно дважды нажать на неё левой кнопкой мыши, после чего точка удалится, а Ломаная линия перерисуеться.

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	
Ортогональн ость	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

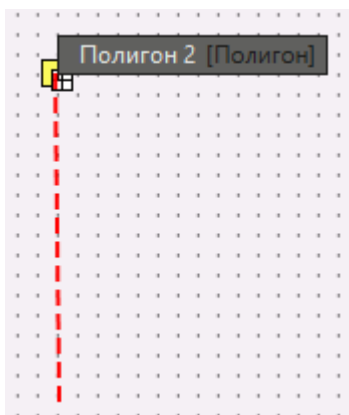
Данный элемент предназначен для рисования произвольных геометрических фигур, представляет собой SVG-фигуру.

В палитре редактора НМІ элемент Полигон находится в категории Прimitives.

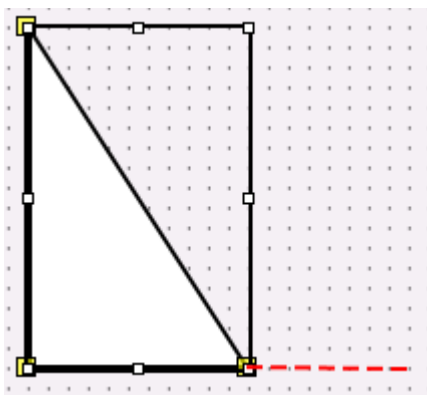
Вид элемента в палитре:



После добавления элемента в рабочую область редактора HMI 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:



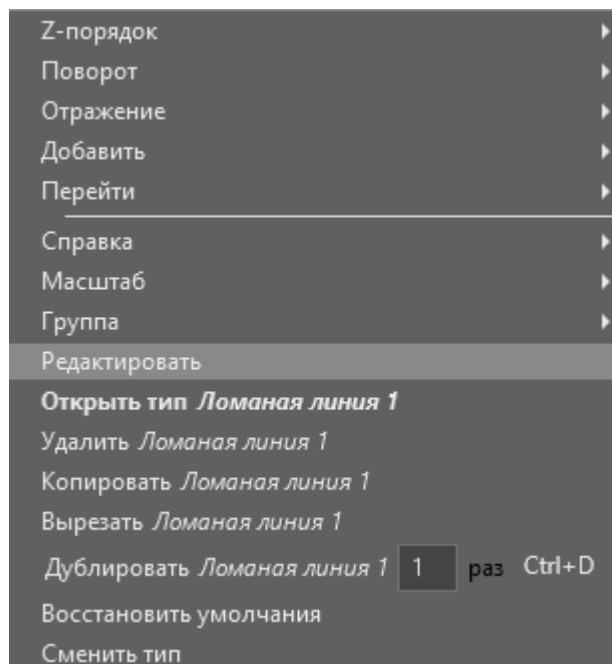
Если нажать левой кнопкой мыши на какое-либо место рабочей области, то будет создана вторая точка. Если переместить мышь и опять нажать левую кнопку, то будет создана третья точка:



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.

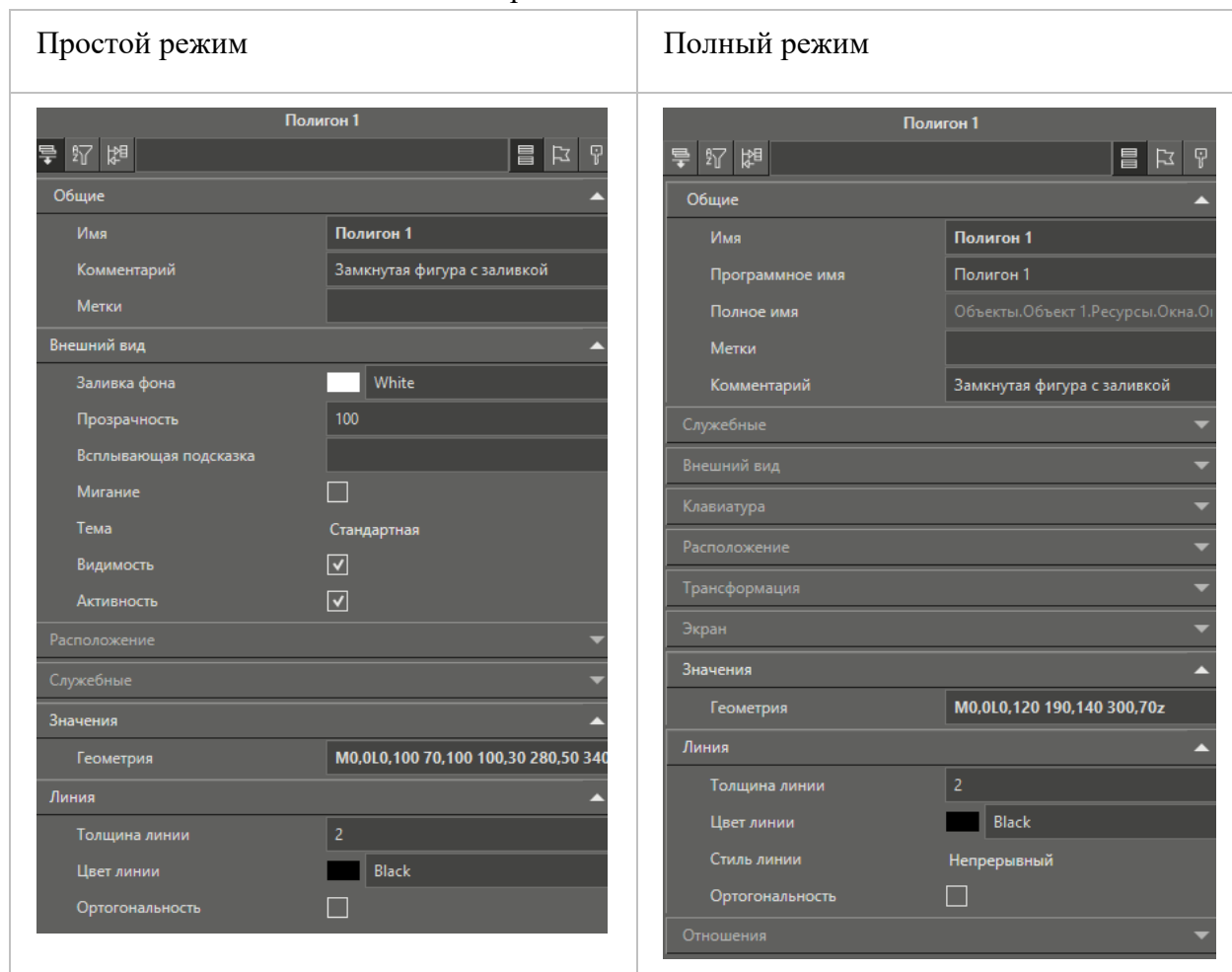
Для того чтобы войти в режим редактирования ранее нарисованной фигуры, необходимо нажать правую кнопку мыши и в контекстном меню выбрать пункт Редактировать:



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


### 8.3.4.5. Овал, Прямоугольник, Треугольник

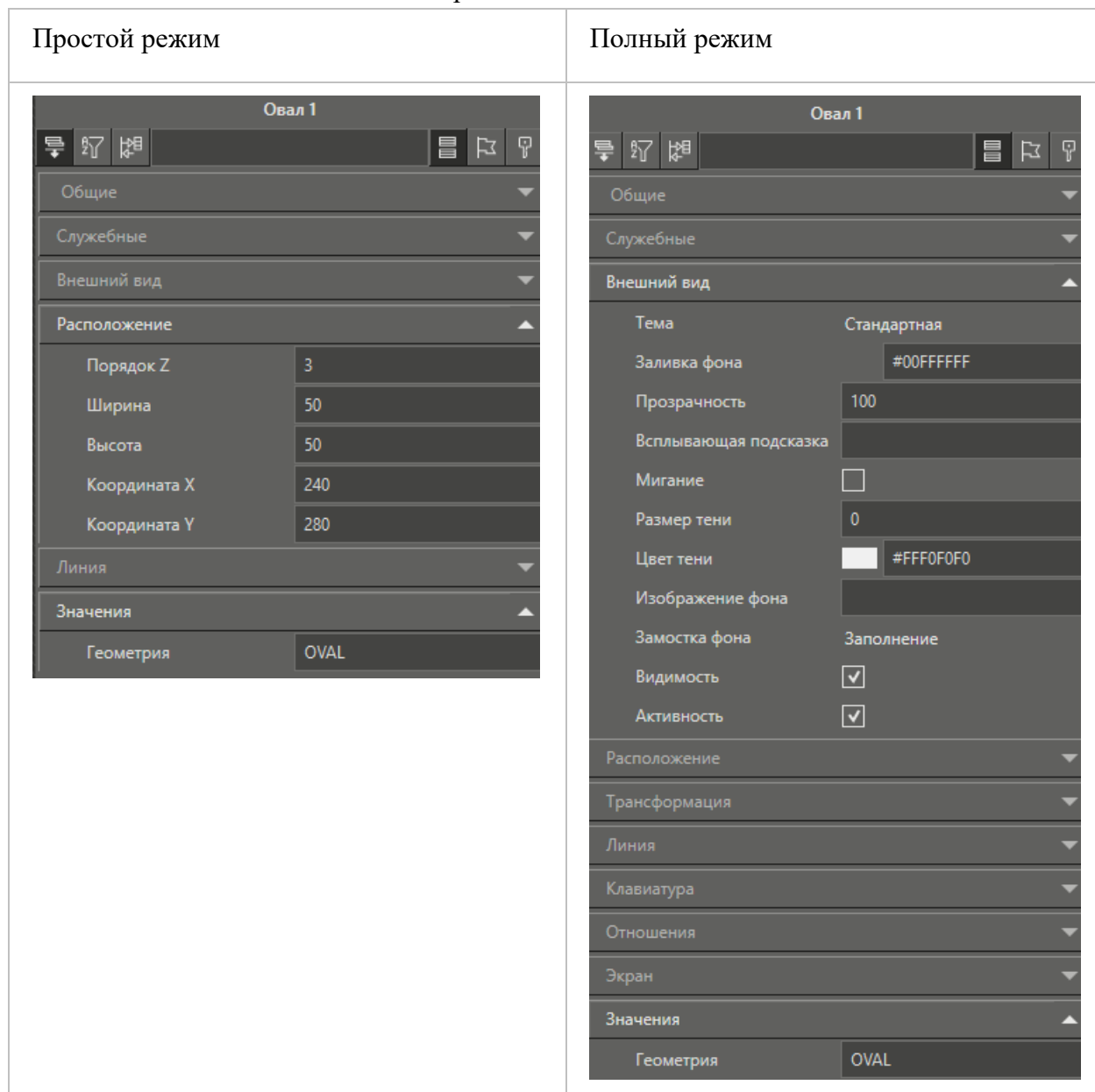
Данные элементы представляют собой SVG-фигуры, в настройках которых по умолчанию задана их геометрическая форма (тип фигуры). Чаще всего используются в окнах в виде индикаторов (т.е. меняют свой внешний вид в зависимости от значений параметров).

В палитре редактора НМІ 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 :



Описание основных свойств графических элементов Овал, Прямоугольник, Треугольник:

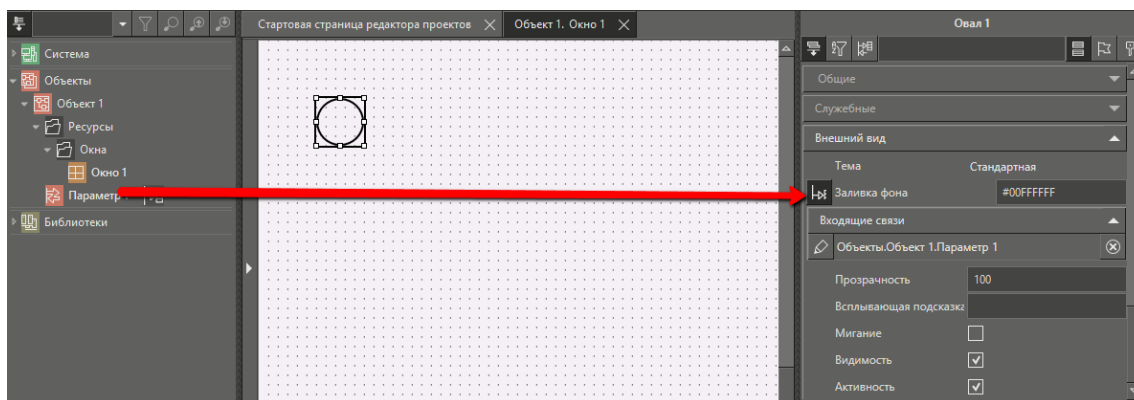
Название	Ссылка на описание
Категория	Значение
Геометрия	Задается тип фигуры. Настройку по умолчанию можно изменить, при этом соответственно изменится и тип фигуры.

Описание других свойств элементов смотрите в разделе НМІ.Свойства

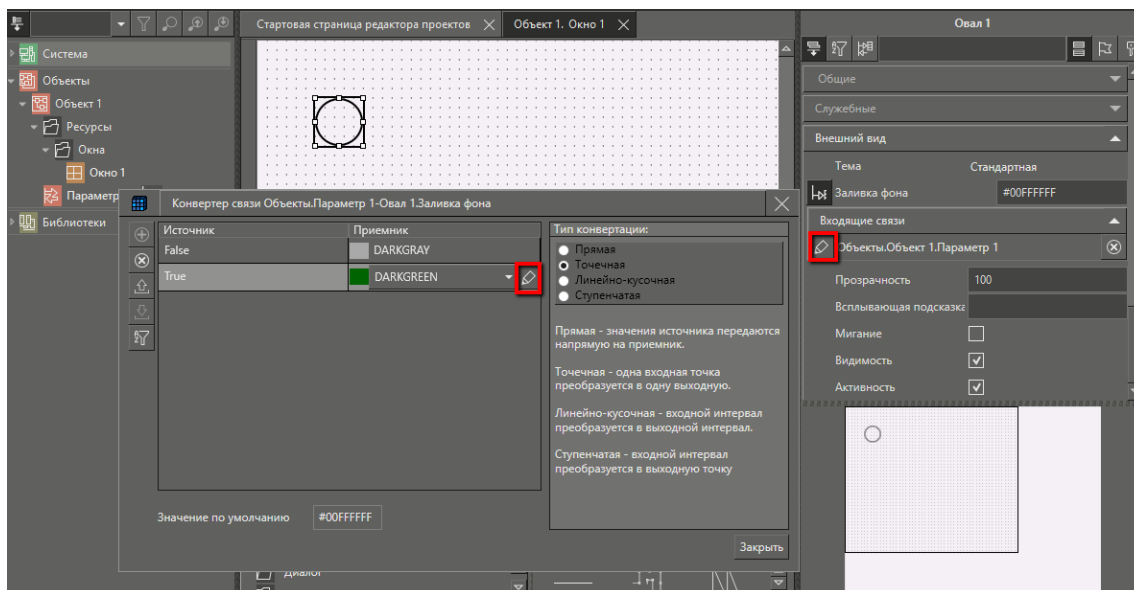
## Рекомендации

Рассмотрим основные принципы работы с элементами на примере Овала

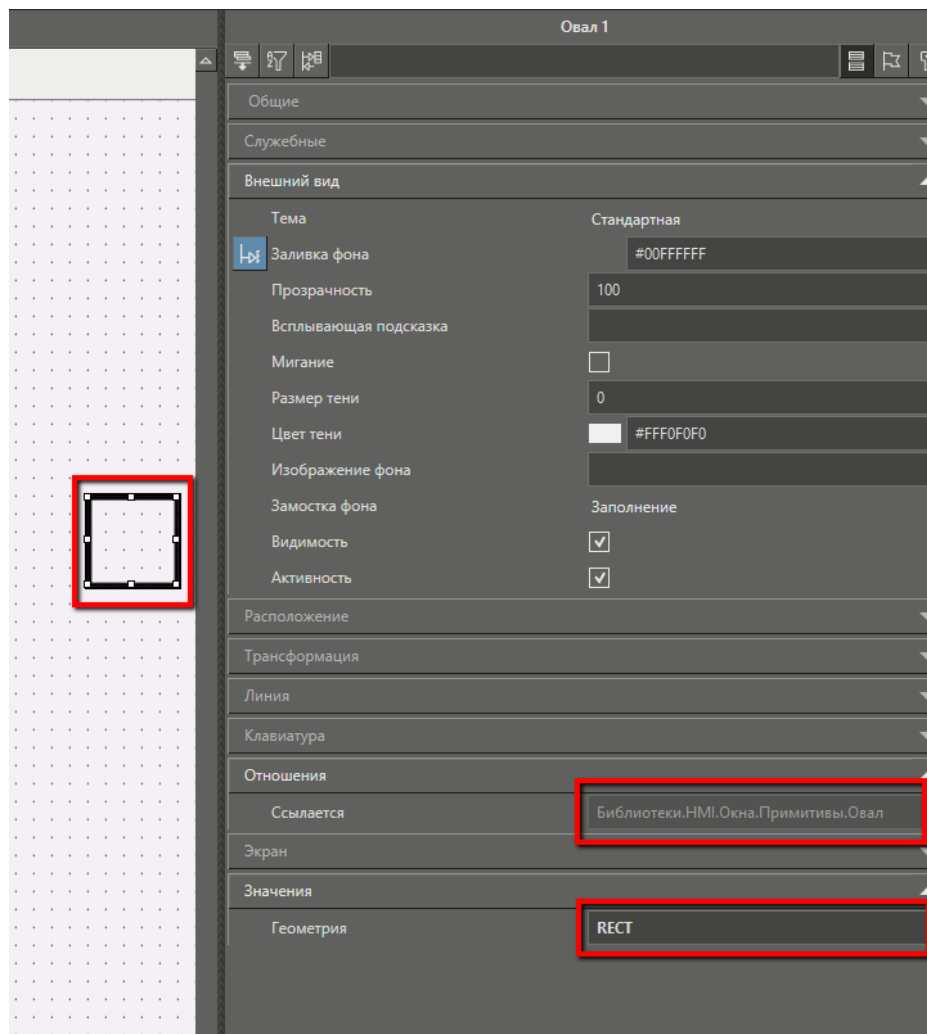
- Данный элемент часто используется для индикации состояния параметра типа BOOL. Например, если параметр имеет значение TRUE, то индикатор должен быть зеленым, а если False - серым. В этом случае параметр проекта необходимо перетащить в свойство Заливка фона.



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

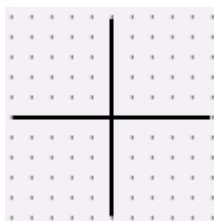
Данный элемент представляет собой SVG-фигуру, в настройках которой, по умолчанию, задано пересечение двух линий под прямым углом: M50,0L50,100M0,50L100,50.

В палитре редактора НМІ элемент Крест находится в категории Примитивы.

Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора НМІ:





Свойства элемента Крест идентичны свойствам элемента Овал.

### 8.3.4.7. Стрелка

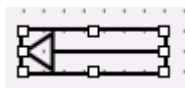
Данный элемент представляет собой SVG-фигуру, в настройках которой по умолчанию задан рисунок стрелки, указывающей влево: M0,0.5 L1,0 L1,1 L 0, 0.5 M1, 0.5 L5, 0.5.

В палитре редактора HMI элемент Стрелка находится в категории Прimitives.

Вид элемента в палитре:



Вид элемента после добавления в рабочую область редактора HMI:



Свойства элемента Стрелка идентичны свойствам элемента Овал.

Смотрите также:

Динамизация

Программирование в окнах

Создание пользовательской библиотеки примитивов

Создание произвольных фигур

### 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 Прimitives.

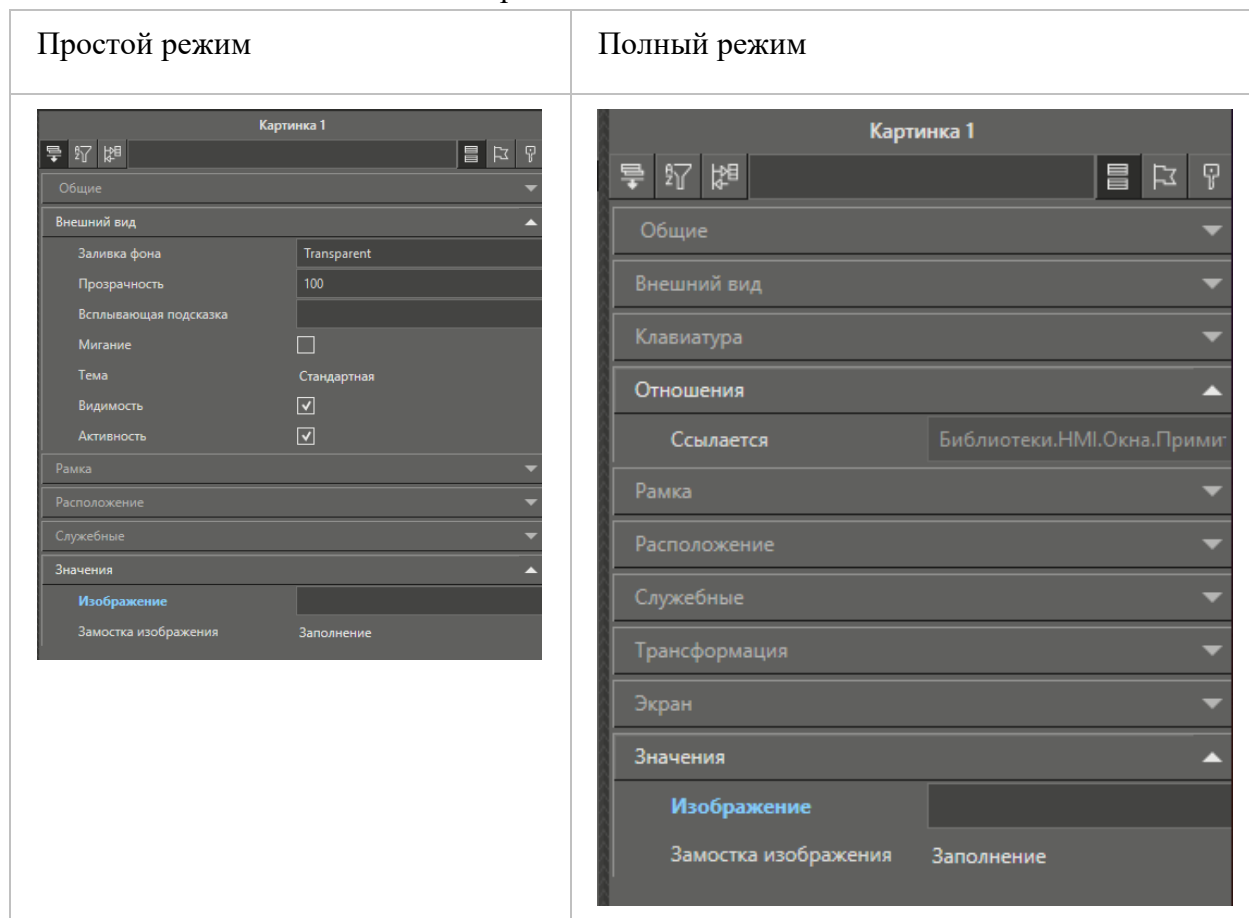
Вид элемента в палитре:




Вид элемента после добавления в рабочую область редактора HMI:

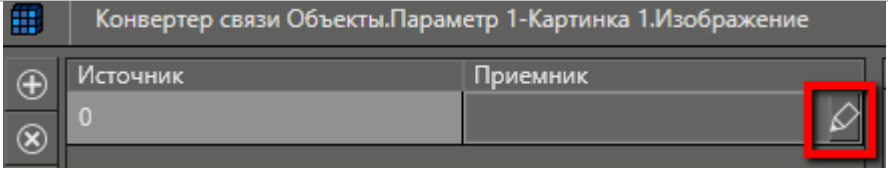


Вид панели свойств элемента при нажатой и отжатой кнопке :



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>

	
Замостка изображения	<p>Задает тип использования изображения элемента. Используется совместно со свойством Изображение. Может принимать значения:</p> <ul style="list-style-type: none"> <li>• Заполнение - размер изображения будет изменен по границам элемента;</li> <li>• Мозаика - размер изображения будет исходным, и повторен столько раз, сколько необходимо чтобы замостить всю площадь элемента;</li> <li>• Центр - размер изображения будет исходным, изображение будет центрировано;</li> <li>• Нет - изображение использоваться не будет.</li> </ul>

Описание других свойств элемента смотрите в разделе НМІ.Свойства.

### 8.3.4.9. Графический элемент Видео

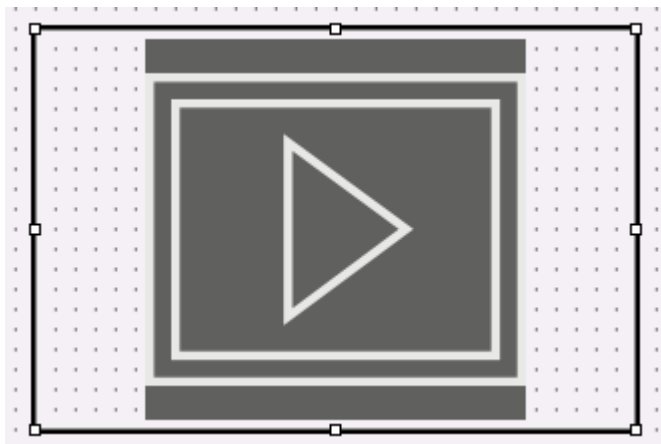
Данный элемент предназначен для проигрывания медиа-ресурса (видео из файла). Доступный формат файла: WebM.

В палитре редактора НМІ элемент Видео находится в категории Примитивы.

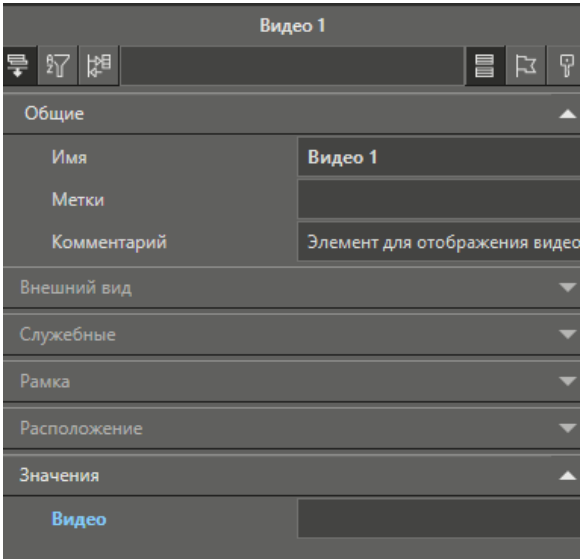
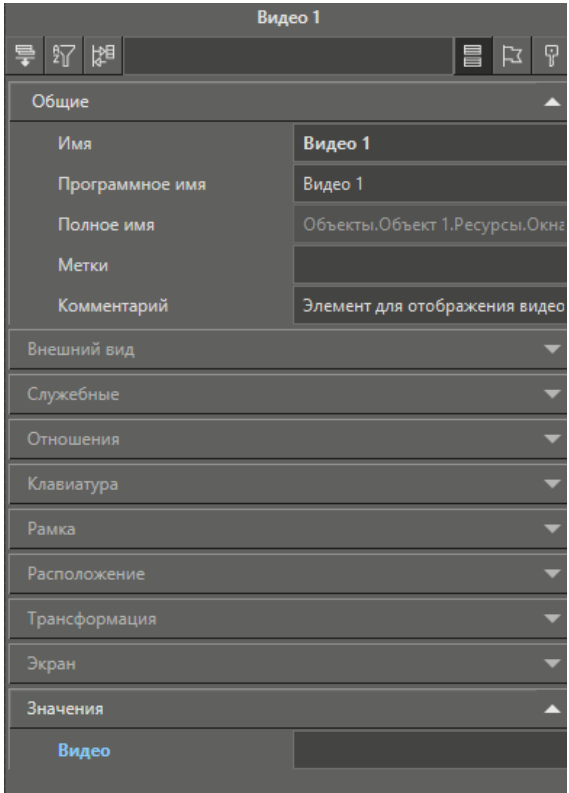
Вид элемента в палитре:




Вид элемента после добавления в рабочую область редактора НМІ:

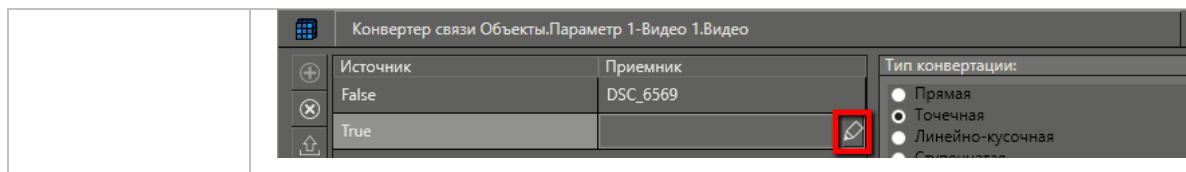


Вид панели свойств элемента при нажатой и отжатой кнопке :

Простой режим	Полный режим
	

Описание основных свойств графического элемента Видео:

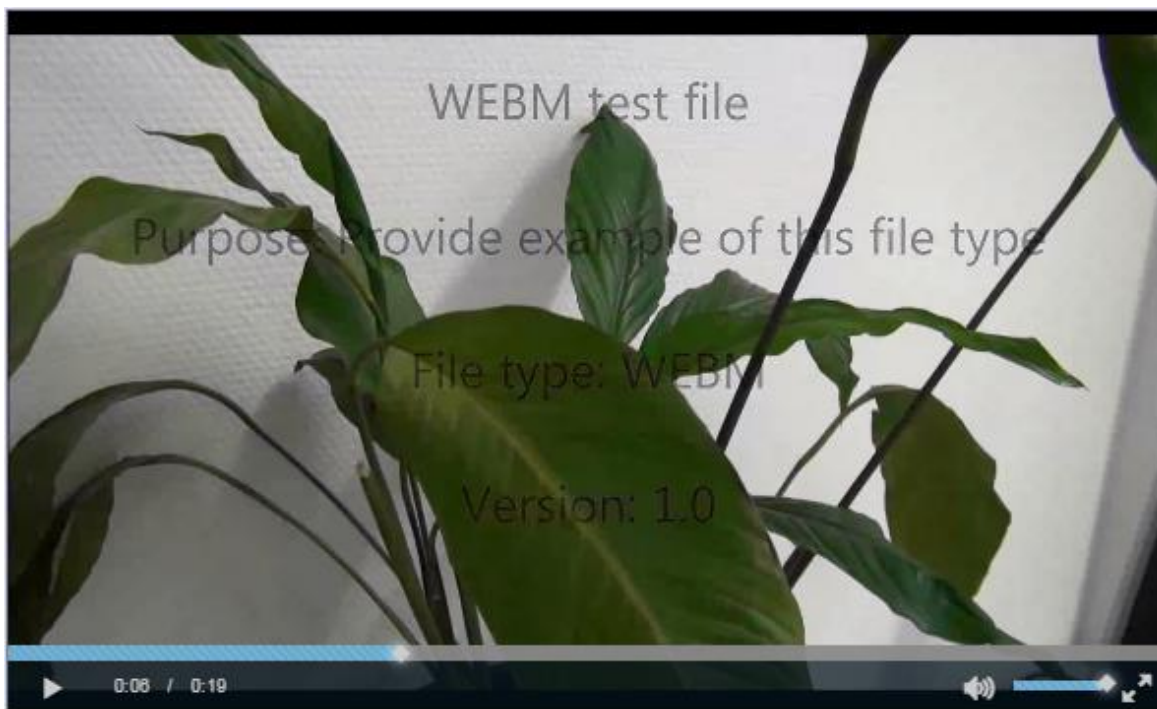
Название	Ссылка на описание
Категория Значения	
Видео	<p>Определяет видео, которое будет проигрываться в клиенте визуализации. При нажатии на кнопку , которая появляется с правой стороны при наведении мыши на свойство, открывается диалоговое окно выбора медиа-ресурса. В качестве видео может быть использован произвольный видео-файл.</p> <p>Если требуется, чтобы в клиенте визуализации проигрывались разные видео, 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>



Описание других свойств элемента смотрите в разделе НМІ.Свойства.

Работа в режиме исполнения

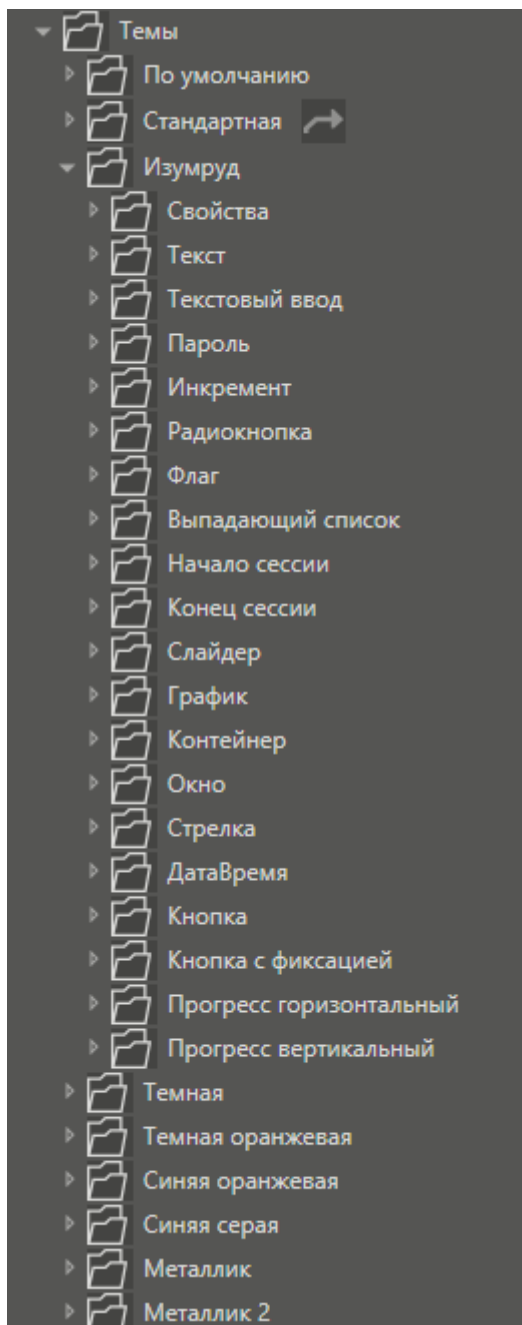
Вид элемента в браузере:



В режиме исполнения можно остановить и перемотать видео, управлять звуком и включить полноэкранный режим.

### 8.3.5. Темы библиотеки НМІ

Группа Темы библиотеки НМІ содержит несколько тем. Каждая Тема содержит те же графические элементы, что и группа Окна библиотеки НМІ.



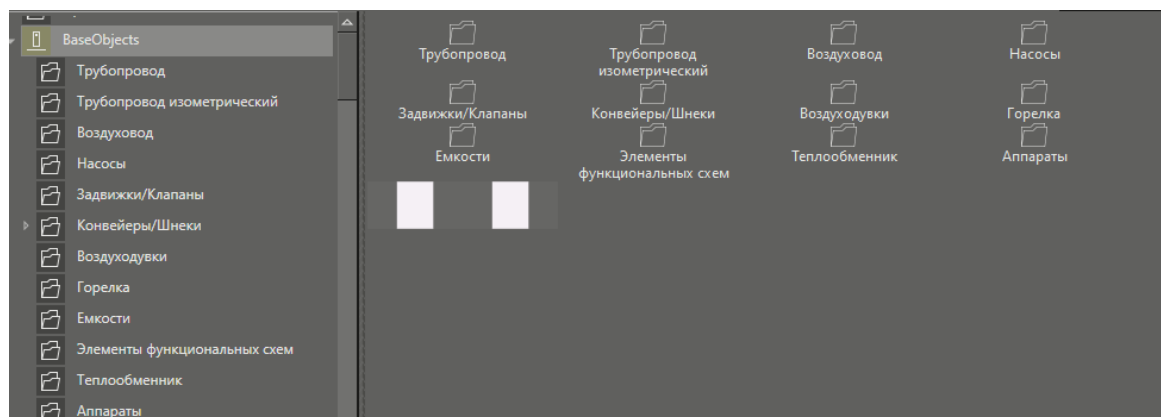
Изменение свойства графического элемента в применяемой теме (в библиотеке) автоматически изменяет данное свойство на мнемосхеме, если это свойство не редактировалось на мнемосхеме вручную.

### 8.3.6. Категория BaseObjects

Категория BaseObjects палитры редактора НМІ 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 НМІ эта категория имеет вид:



После добавления элемента из палитры могут быть установлены входящие/исходящие связи между свойствами элементов и параметрами проекта.

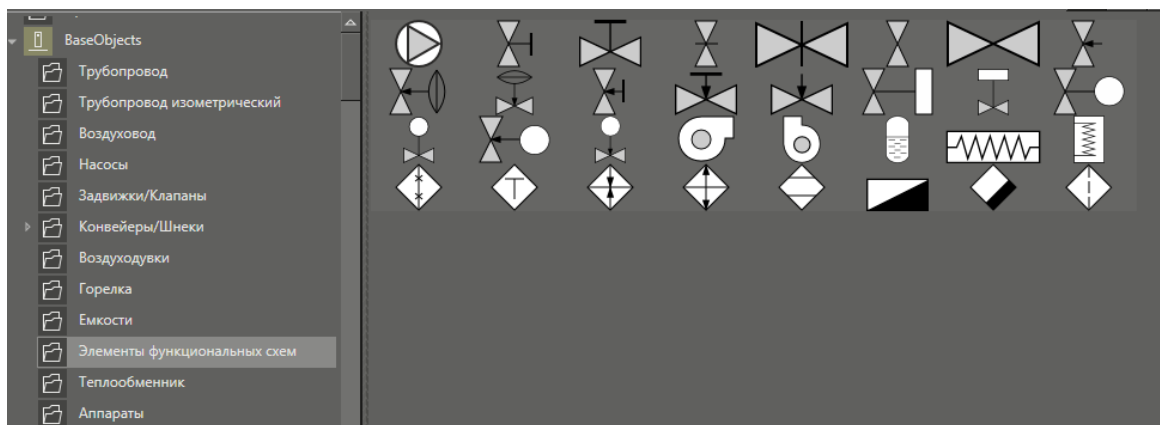
Категория BaseObjects имеет сложную структуру, т.к. содержит внутри себя не только отдельные элементы, но и вложенные категории:

- 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 HMI содержит элементы, внешний вид которых соответствует условным графическим изображениям технологического оборудования на функциональных схемах автоматизации.

В палитре редактора HMI эта категория имеет вид:

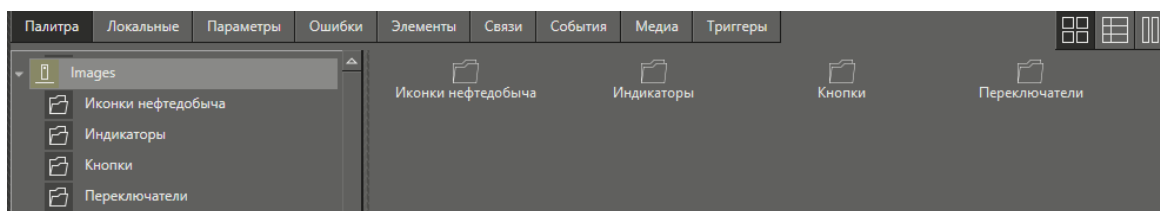


После добавления элемента из палитры могут быть установлены входящие/исходящие связи между свойствами элементов и параметрами проекта.

### 8.3.7. Категория Images

Категория Images палитры редактора HMI 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 HMI 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. HMI. Свойства

### 8.4.1.Свойство Формат значений

Данное свойство имеет тип STRING и задает формат отображения значений при использовании конвертации:

```

Формат значений, применяемый к результату конвертации
примеры:
f2 - 10,03
{f2} м3 - 10,03 м3
давление {f3} кПа - давление 10,034 кПа
e2 - 1,0034E1
{e2} м3 - 1,0034E1 м3
{dd.MM.yyyy HH:mm:ss.fff} - 12.01.2014 23:59:59.143

```

### 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.

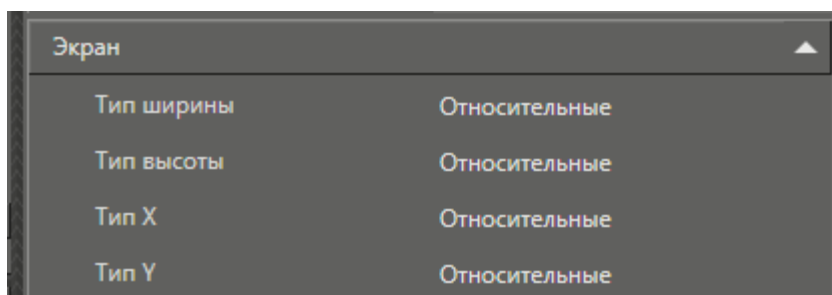
В случае, если между необязательными символами оказываются не введенные, при обработке введенного значения пробелы будут удаляться.

Задача	Значения свойства	Вид поля ввода	Ввод в режиме исполнения	Результат
	009.90	____.____	__0.0__	0.0

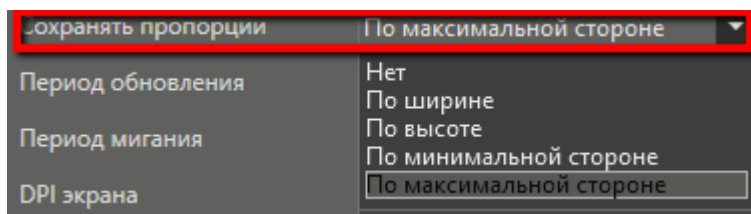
Допустим, требуется ограничить ввод чисел диапазоном от 0.0 до 999.99.			_10.1_	10.1
			_10.99	10.99
			1_1.1_	11.1
			1__.1_	т.к. не хватает обязательной цифры перед точкой, то введенное значение будет сброшено
Пусть можно вводить любое число от 0.001 до 99999.999	00000.00 0	____. __	1___.__1	1.1
			____.____	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 обязательной цифры после символа /
			0d23/23	0d23/23
			0k2_/1_	0k2/1

### 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.
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 У 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.

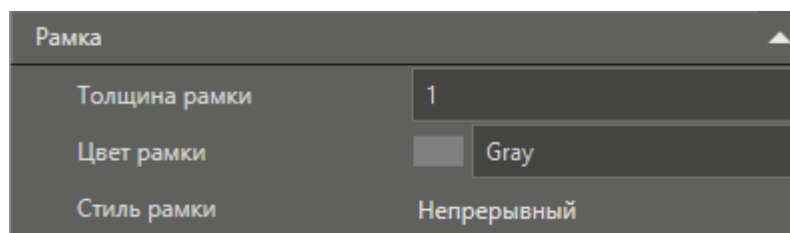
4	Абсолютные	Относительные	Относительные	Не используется
---	------------	---------------	---------------	-----------------

Важно! Данная настройка не влияет на всплывающие окна. Размер, местоположение и возможности масштабирования зависят от настроек, заданных в действии, которое приводит к открытию окна.

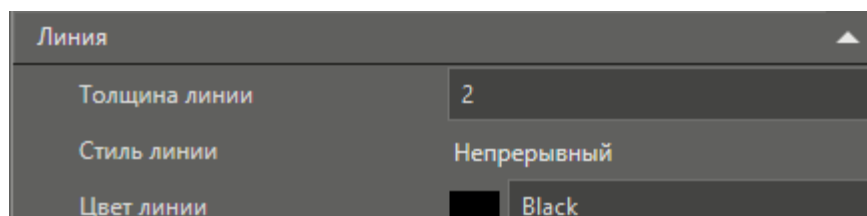
#### 8.4.4. Категория свойств Рамка/Линия


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.Solid-ColorType (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.

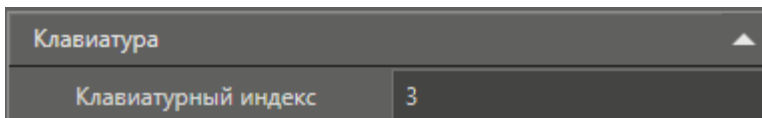
Стиль рамки/линии	HMI. BorderStyleType (Перечисление)	Определяет стиль рамки. Выпадающий список содержит следующие возможные значения: Непрерывный, Пунктир, Точка, Нет.
-------------------	-------------------------------------	--

### 8.4.5. Категория свойств Клавиатура

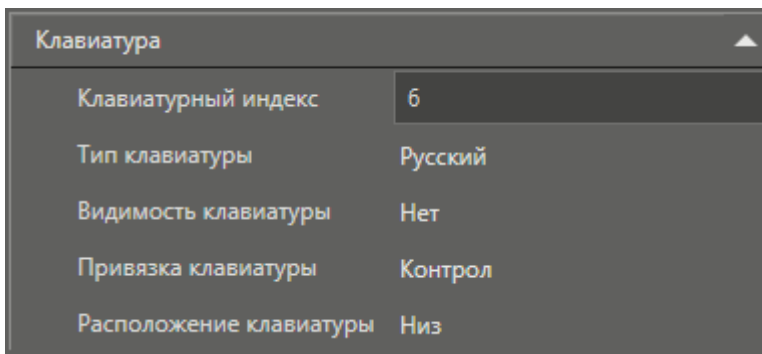
Категория свойств Клавиатура присутствует у всех элементов, однако, как правило, настраивается только для тех элементов, которые предназначены для ввода произвольных данных в клиенте визуализации, и когда подключить физическую клавиатуру нет возможности.

**Важно!** Виртуальная клавиатура поддерживается только в HMI v1.

Вид категории для элементов, у которых не предусмотрен ввод значений:



Вид категории для элементов, у которых предусмотрен ввод значений:



Свойство	Тип	Описание
Клавиатурный индекс	INT	Определяет очередь перевода фокуса на элемент при последовательных нажатиях клавиши TAB. При первом нажатии TAB фокус переводится на элемент с наименьшим клавиатурным индексом. При размещении элементов, им автоматически присваивается значение данного свойства, которое затем можно изменить вручную.

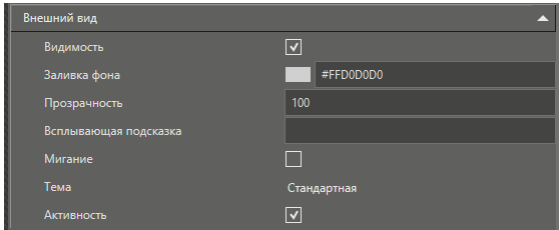
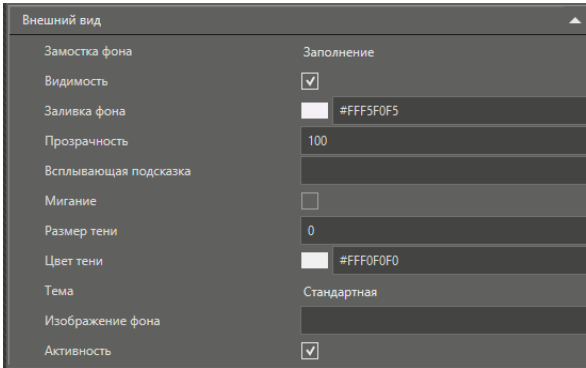
Тип клавиатуры	HMI. KeyboardType	Определяет тип раскладки виртуальной клавиатуры. Возможные варианты: английский, русский, цифры.
Видимость клавиатуры	HMI. KeyboardVisibilityType	<p>Включает виртуальную клавиатуру. Из выпадающего списка можно выбрать один из трех вариантов использования:</p> <ul style="list-style-type: none"> <li>• Нет - клавиатура не будет использоваться (задано по умолчанию);</li> <li>• Всегда - клавиатура будет отображаться всегда, когда открыто окно с элементом, для которого задаются текущие настройки;</li> <li>• При фокусе - виртуальная клавиатура будет отображаться после выделения элемента в окне.</li> </ul> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Важно! Доступно только для HMI v1.</p> </div>
Привязка клавиатуры	HMI. AttachType	<p>Определяет местоположение виртуальной клавиатуры. Из выпадающего списка можно выбрать один из двух вариантов:</p> <ul style="list-style-type: none"> <li>• Контрол - клавиатура будет находиться рядом с элементом;</li> <li>• Родитель - клавиатура будет находиться рядом с окном, в котором находится элемент.</li> </ul>
Расположение клавиатуры	HMI. LayoutType	<p>Определяет с какой стороны от элемента или его родителя будет находиться виртуальная клавиатура. Из выпадающего списка можно выбрать следующие значения: Верх, Низ, Лево, Право.</p>

### 8.4.6. 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 элемента, при нажатой и отжатой кнопке  :

Простой режим	Полный режим
	

Описание:

Название	Описание
Замостка фона	<p>Задает 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"> <li>• Затопление - размер фонового изображения будет изменен по границам элемента;</li> <li>• Мозаика - размер фонового изображения будет исходным, и повторен столько раз, сколько необходимо чтобы замостить всю площадь элемента;</li> <li>• Центр - размер фонового изображения будет исходным, изображение будет центрировано;</li> <li>• Нет - фоновое изображение использоваться не будет.</li> </ul>
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>Задает степень прозрачности элемента. Значение изменяется в пределах от 0 до 100, где 0 - элемент полностью прозрачный, 100 - элемент полностью не прозрачный. Если элемент полностью прозрачный, то динамизация и Действия не исполняются.</p>



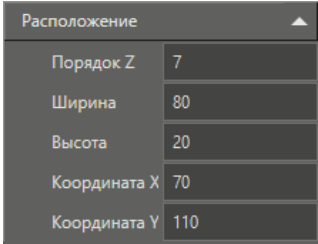
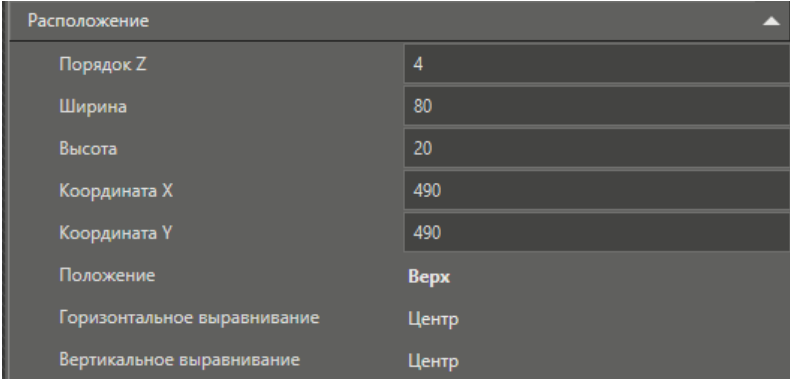
Всплывающая подсказка	Данное свойство имеет тип STRING и задает текст, отображаемый во всплывающей подсказке элемента.
Мигание	Данное свойство имеет тип BOOL и может принимать значение TRUE (использовать мигание) или False (не использовать мигание). Период мигания задает свойство Период мигания в панели свойств Шаблона экрана.
Видимость	Данное свойство имеет тип BOOL и может принимать значение TRUE (элемент видим) или False (элемент невидим).
Активность	<p>Данное свойство имеет тип BOOL. Если свойство находится в состоянии TRUE, то свойства элементов в режиме исполнения принимают и отправляют значения (работает динамизация), а также выполняются Действия. Если свойство находится в состоянии False, то работает только входящая связь, а Действия и ввод значений исполняться не будут. Если свойство имеет значение TRUE, то элемент отображается серым цветом. На рисунке слева находится элемент, свойство которого принимает значение TRUE, а справа - False:</p> <div data-bbox="521 1058 891 1129" style="text-align: center;">  </div> <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.
Изображение фона	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.

Радиус скругления	Задаёт радиус скругления углов рамки элемента (в px).
-------------------	---

### 8.4.7. 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:

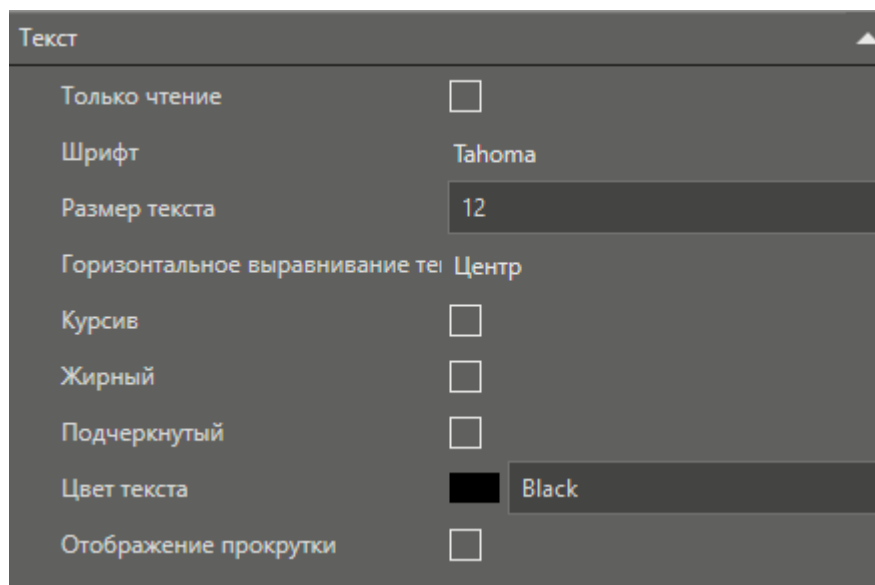
Название	Тип	Описание
Порядок Z	INT	Задаёт положение элемента по оси Z (0 – самое нижнее положение).
Ширина	REAL	Задаёт ширину элемента (в px) в редакторе НМІ. У окна, открывающегося в клиенте визуализации, значение может быть изменено в зависимости от настроек, заданных в категории Экран.
Высота	REAL	Задаёт высоту элемента (в px) в редакторе НМІ. У окна, открывающегося в клиенте визуализа-

		ции, 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 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		

### 8.4.8. 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 <b>BOOL</b> . Если свойство Только чтение имеет значение <b>TRUE</b> , то в элементе блокируется ввод значений.
Шрифт	Данное свойство имеет тип <b>STRING</b> и задает семейство шрифта, используемое в элементе.
Размер текста	Это свойство имеет тип <b>BYTE</b> 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 <b>BOOL</b> и может принимать значение <b>TRUE</b> (использовать курсив) или <b>False</b> (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).
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 HMI 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.

### 8.4.9. 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

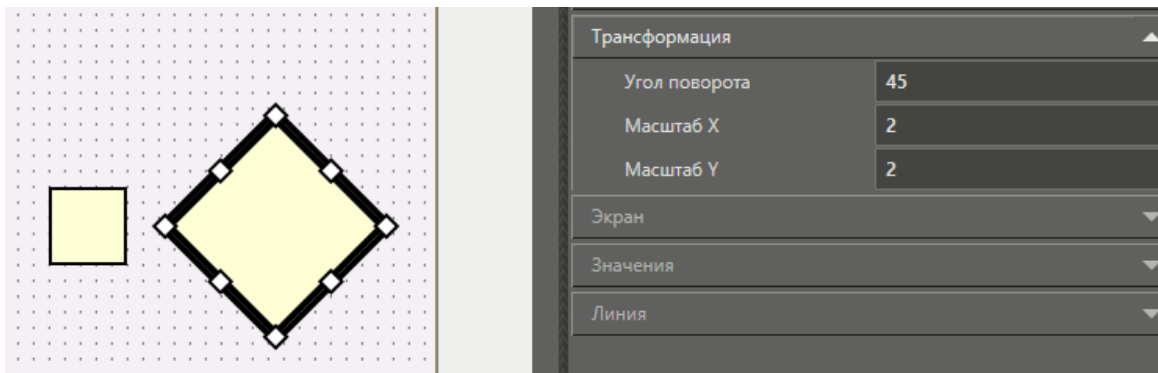
Описание:

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.
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.

## 8.5. 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.
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:

$M X_0, Y_0$  [<line1>]  $X_1, Y_1$  .. [<lineN>]  $X_N, Y_N$  [<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:

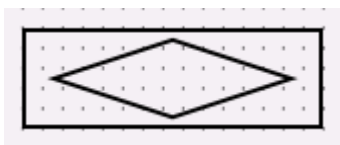
- не указан – в рх;
- Z – от 0 до 1, 1 соответствует 160 рх.

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.

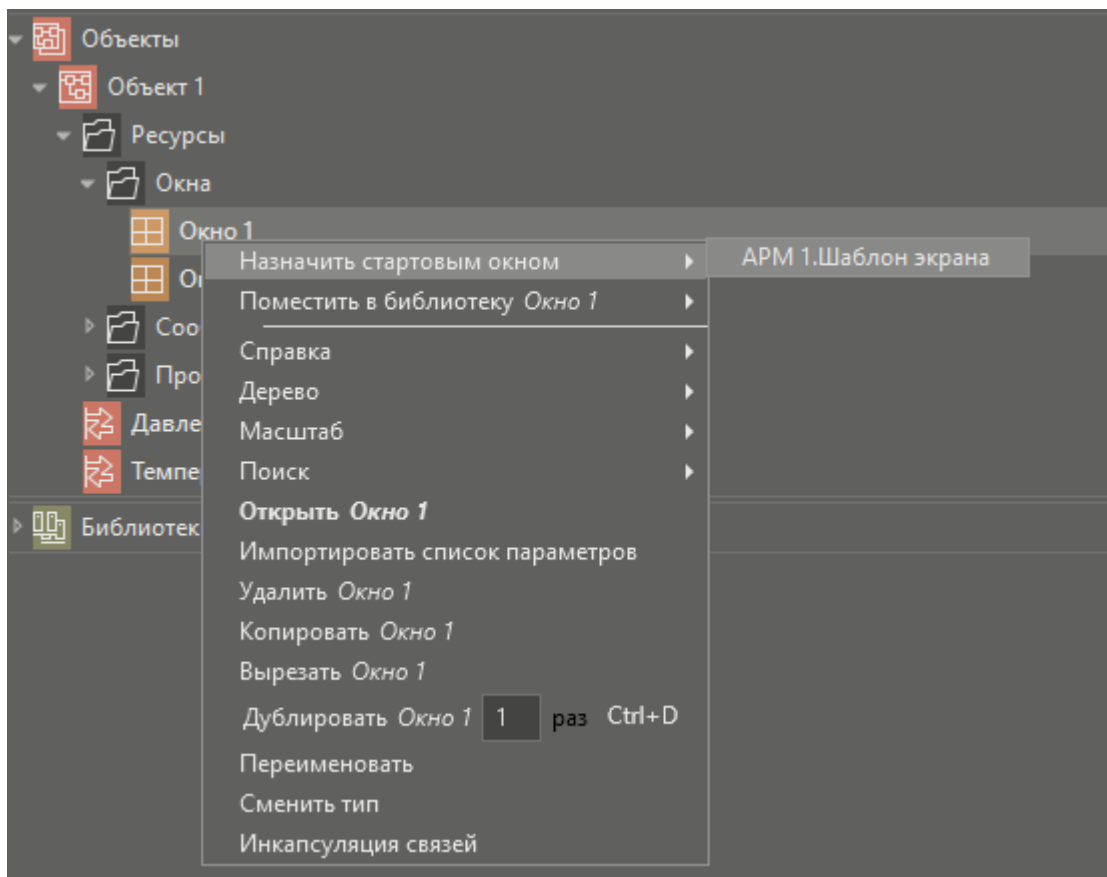
В строке, содержащей зарезервированное слово, нельзя использовать никакие другие описания фигур, в т.ч. другие зарезервированные слова.

Заливка фигур при наложении вычисляется методом XOR.

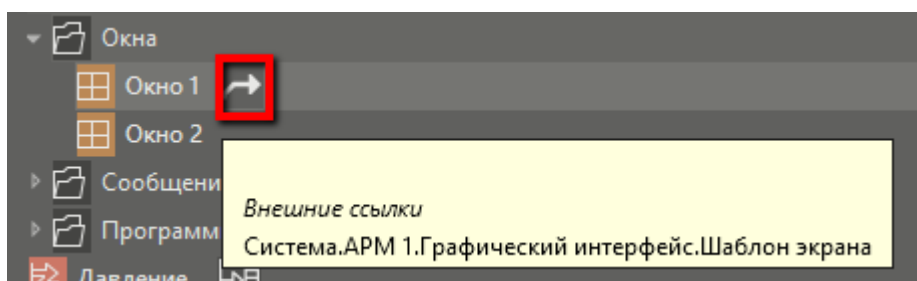
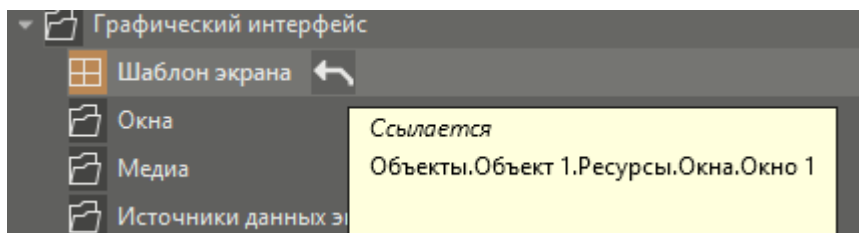
## 8.5.2.Стартовое окно

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.

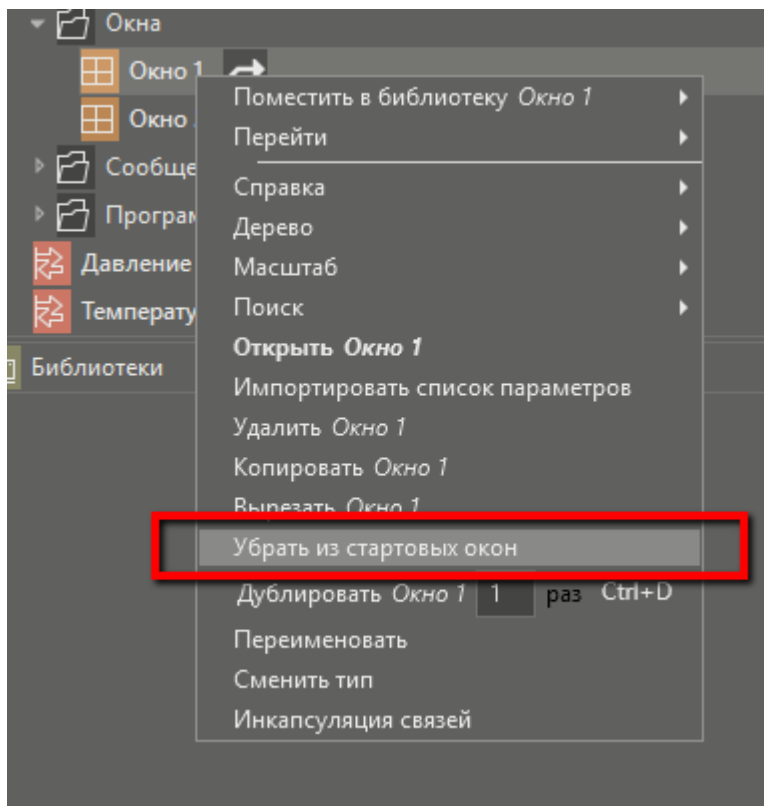




При этом образуется ссылка у Шаблона экрана на окно и у окна - на шаблон экрана:



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

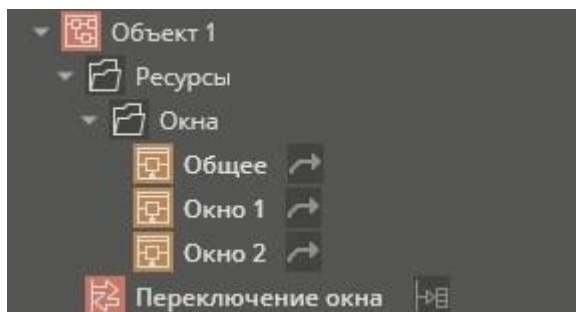
Скорость загрузки окон в режиме исполнения зависит от количества элементов, которые располагаются в окне: чем больше элементов, тем медленнее происходит загрузка окна.

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 контролируемую 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.

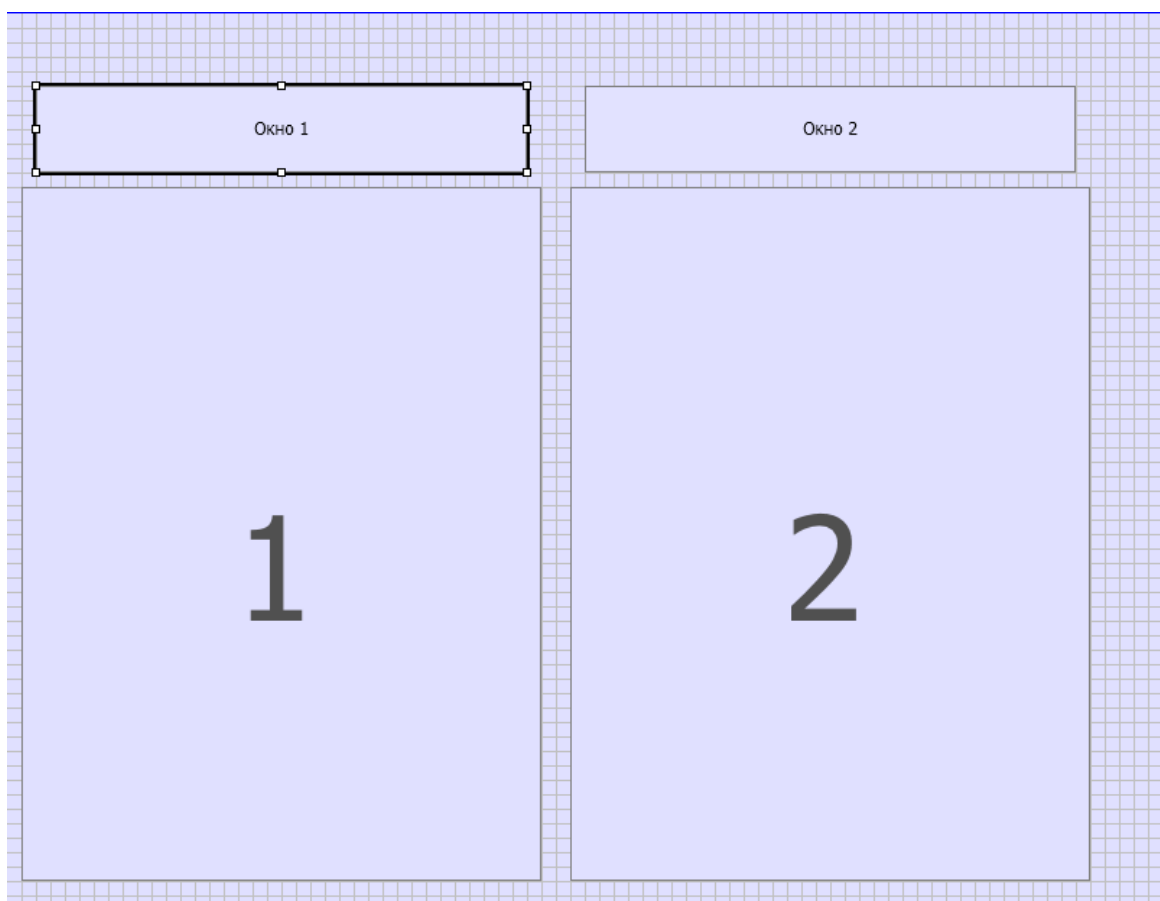
С этой целью можно использовать следующий прием.

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.



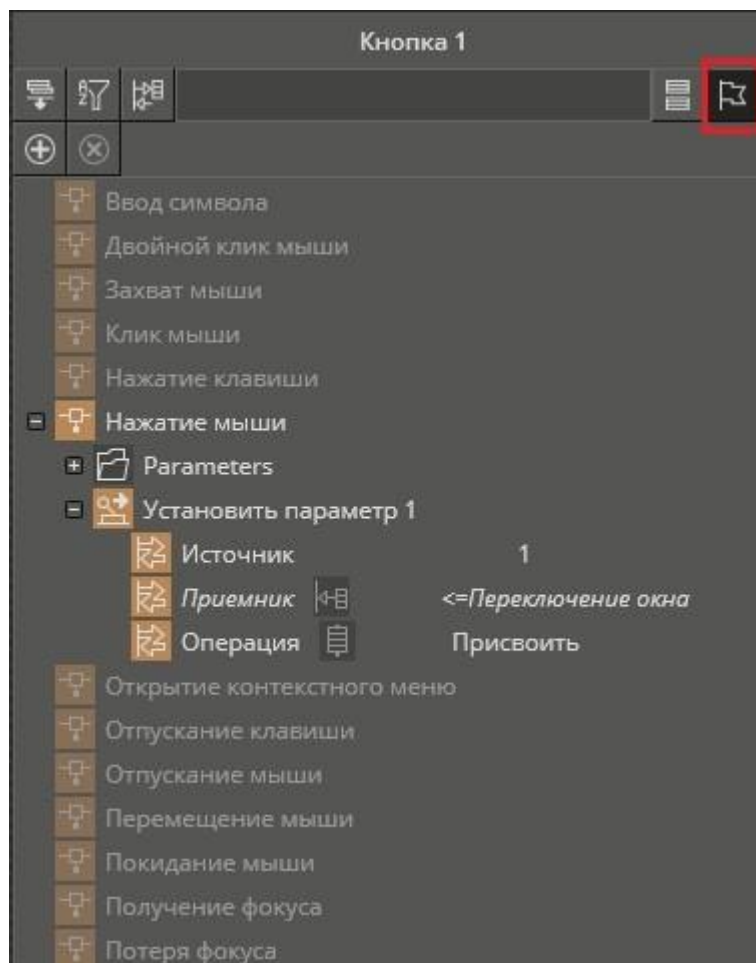
В общее окно перетащим левой кнопкой мыши элементы проекта Окно 1 и Окно 2.

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:

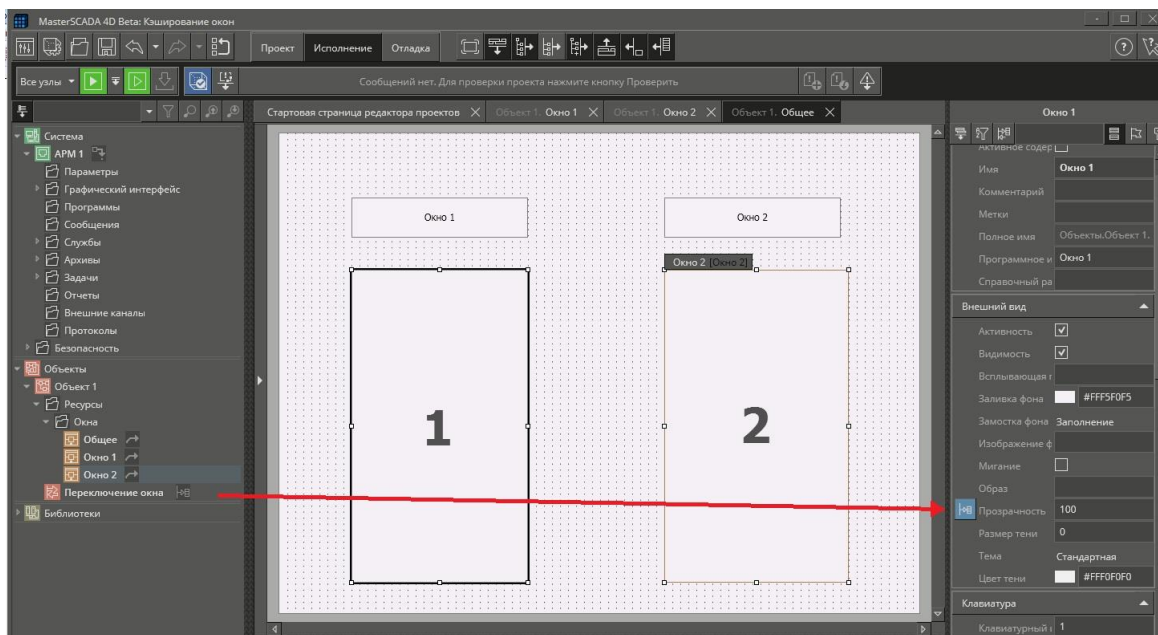


Необходимо, чтобы при нажатии на кнопку Окно 1 параметр Переключение окна принимал значение 1, на кнопку Окно 2 принимал значение 2. Для этого у каждой кнопки следует настроить событие, которое будет выполняться при нажатии мыши.

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.

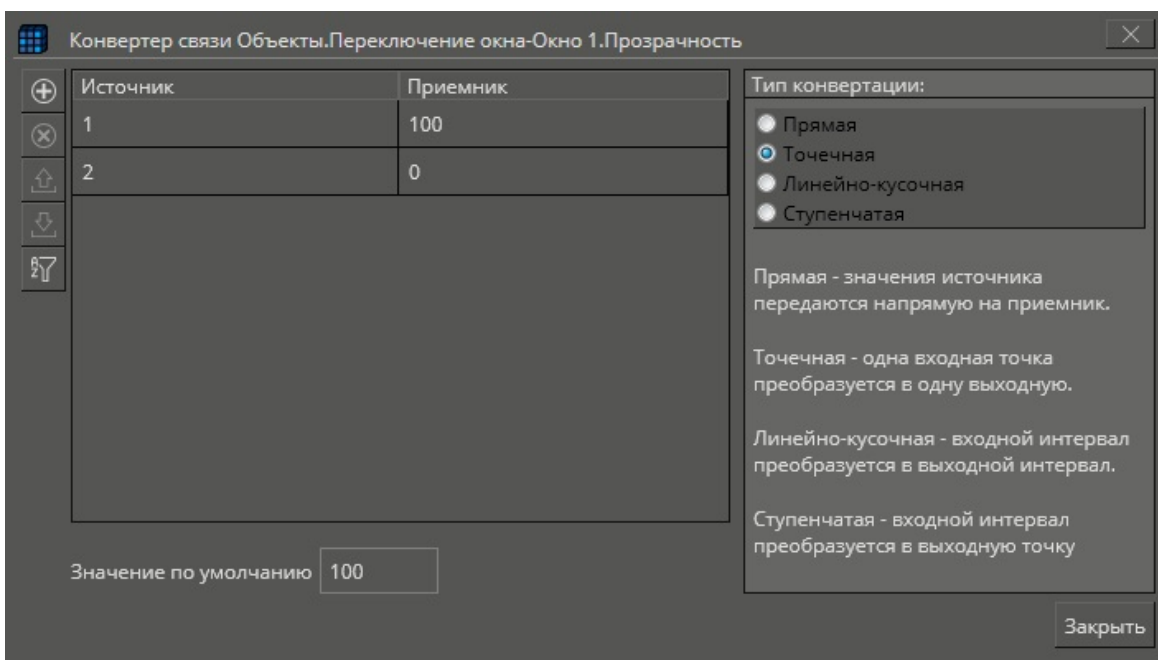


Далее, необходимо установить связь между параметром дерева и свойством Прозрачность окна. Для этого необходимо перетащить Переключение окна в поле нужного свойства.

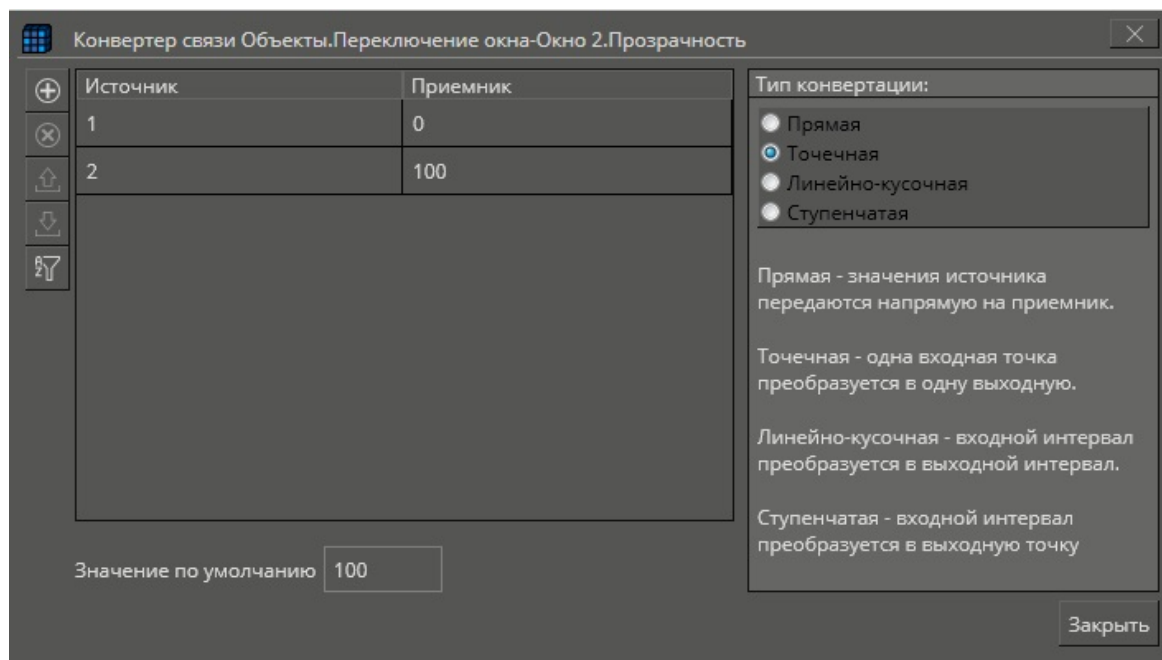


Открыв окно конвертера связи, необходимо настроить точечную конвертацию. Значение свойства 100 обозначает, что окно будет видно пользователю, а 0 - что окно будет скрыто.

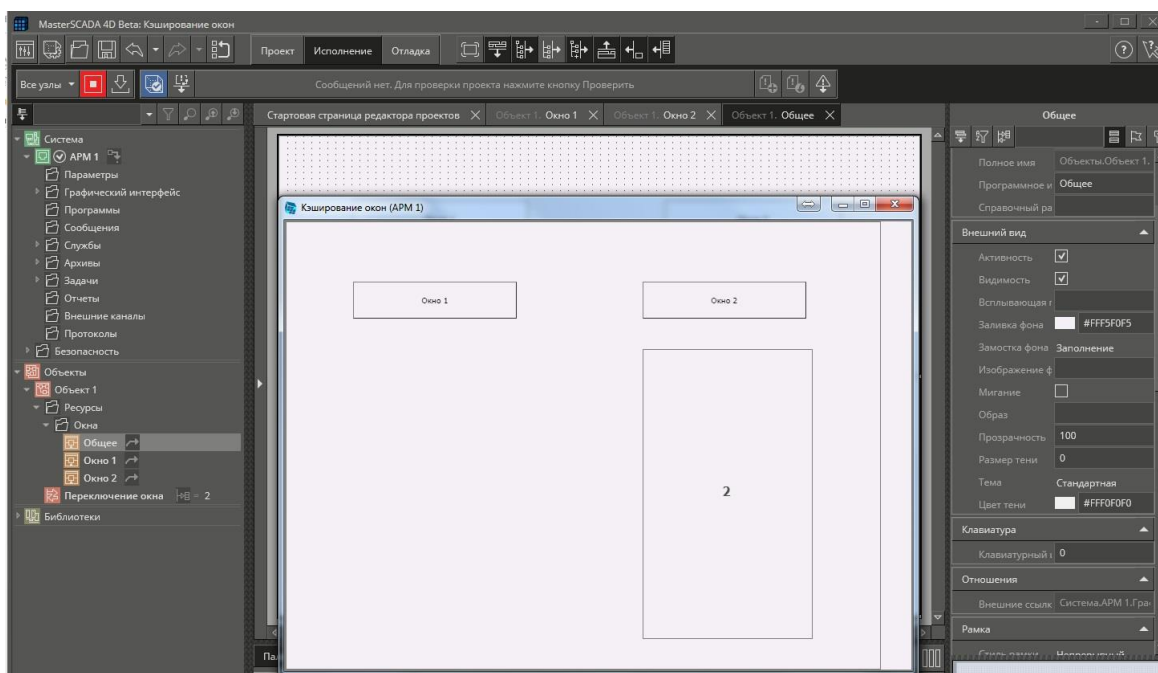
Настройки конвертации для Окна 1:



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.

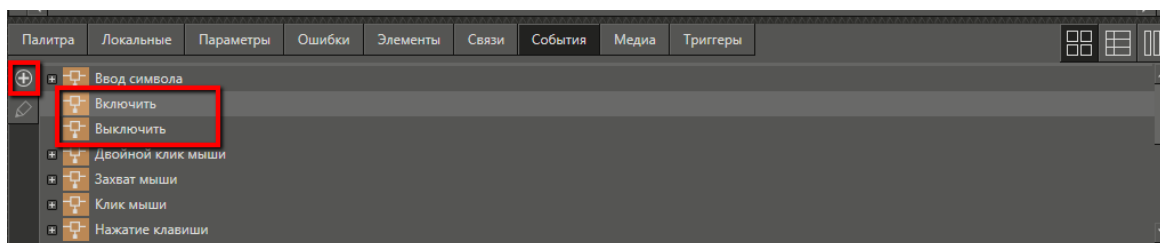
Однако, при таком подходе разработчик проектов должен учитывать ряд особенностей.

### Работа с событиями

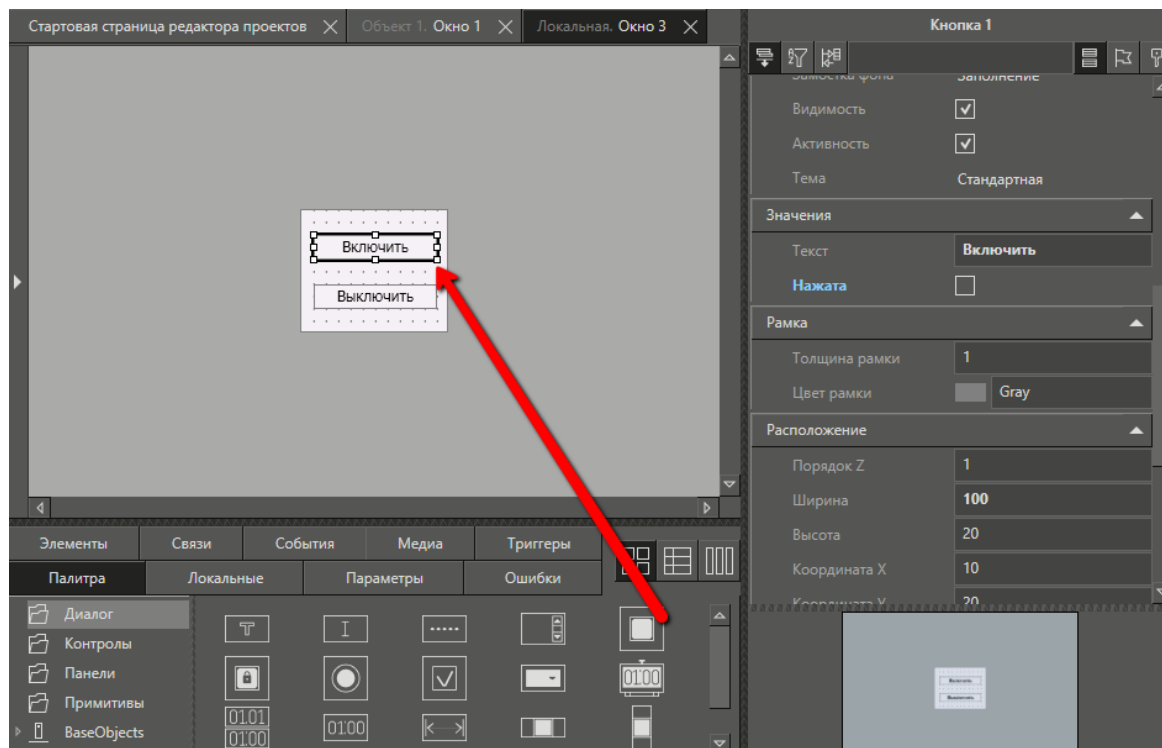
**Важно!** Разработчик проекта не имеет доступа к событиям отдельных элементов экземпляра окна, и может работать только с событиями окна.

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.

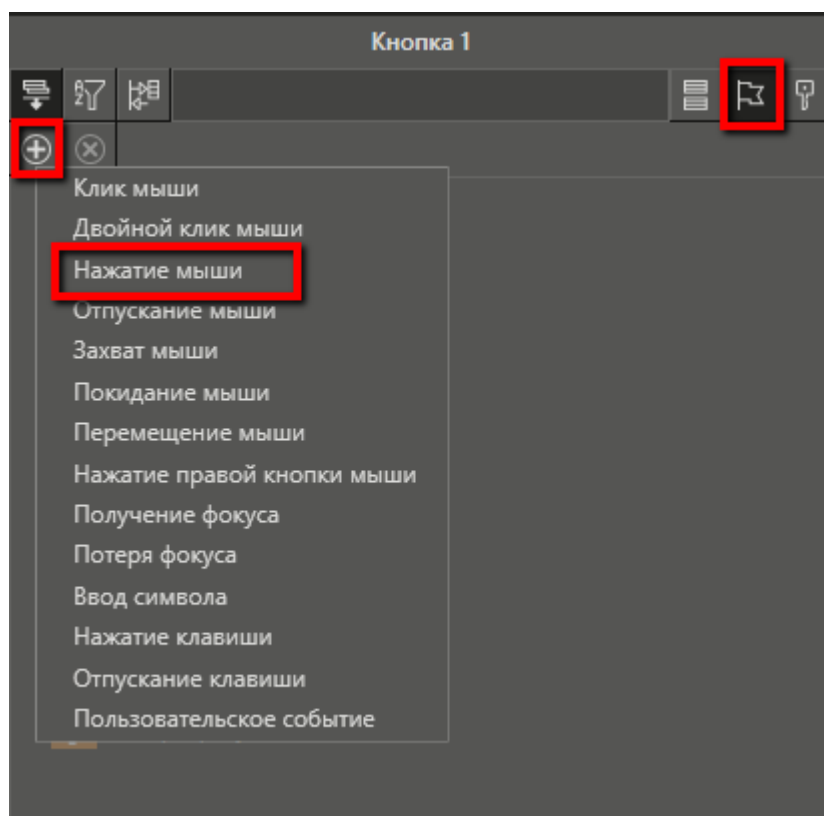
В легенде редактора НМІ на вкладке События создаем два события, назовем их Включить и Выключить:



Добавляем в него элементы управления, например, две кнопки.

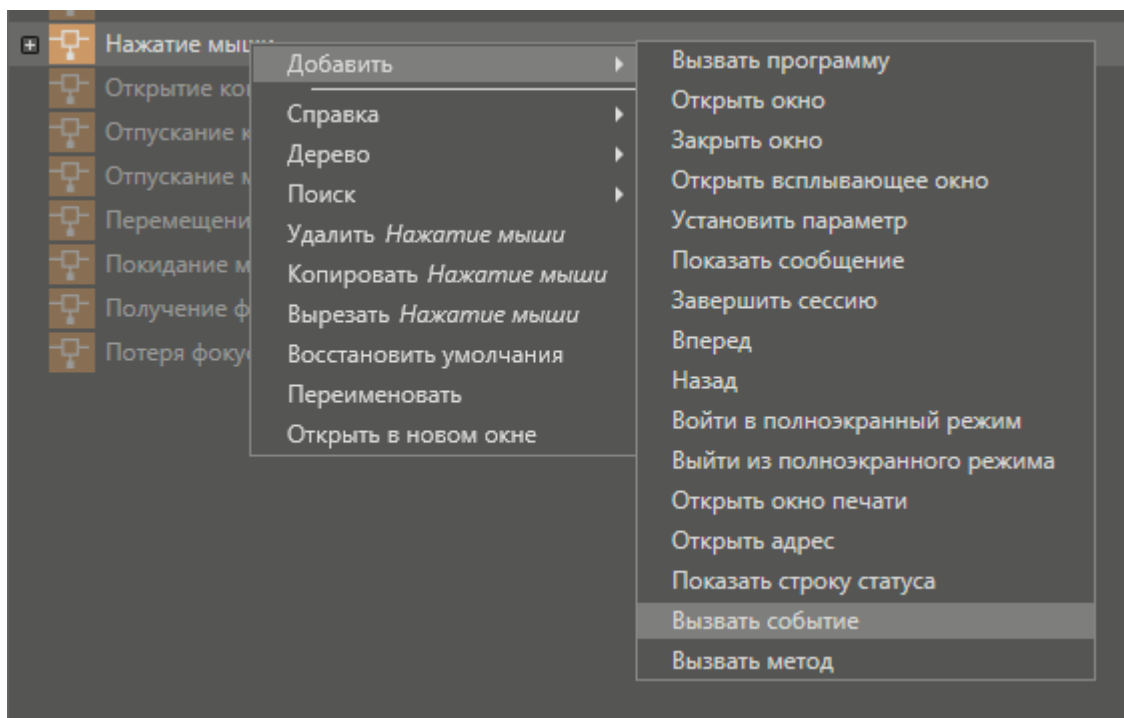


Затем необходимо сделать вызов событий окна из событий кнопок. Для этого выделим кнопку, и в её панели свойств переключимся в режим отображения, позволяющий работать с событиями. Добавим нужное событие, например, Нажатие мыши:

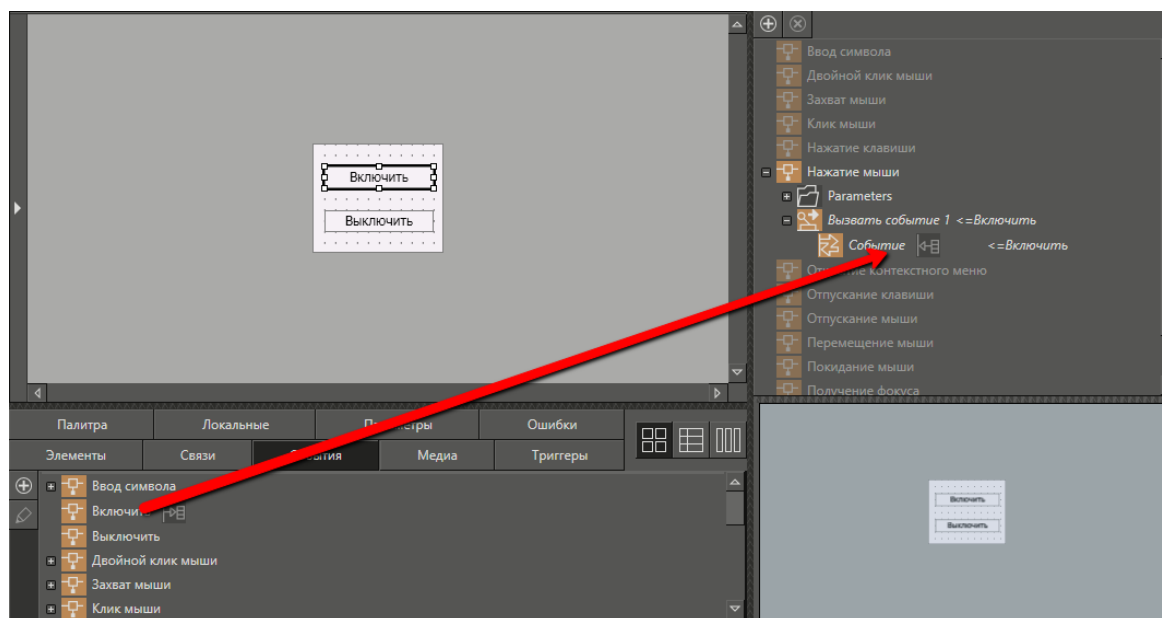




В это событие необходимо добавить действие Вызвать событие:



Затем перетащить пользовательское событие окна в параметры созданного действия:

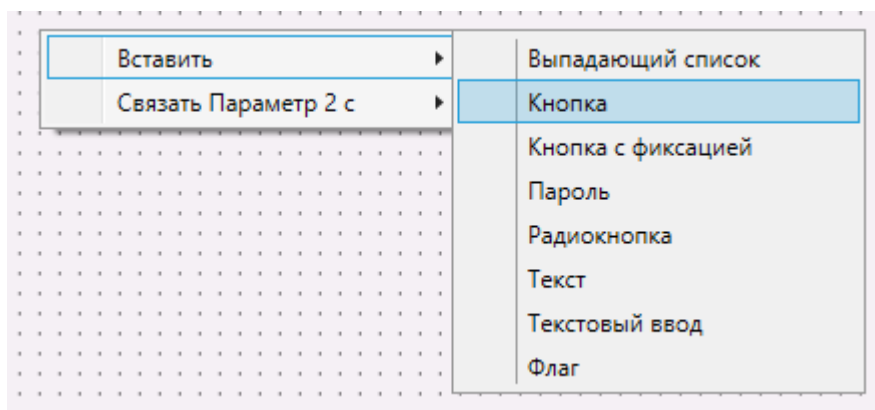


После этого, при помещении экземпляра окна в другое окно, можно будет определять действие, которое будет происходить при вызове того или иного события окна.

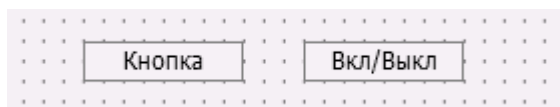
### 8.5.5. Особенности работы с элементами Кнопка и Кнопка с фиксацией

Изменение параметра типа BOOL

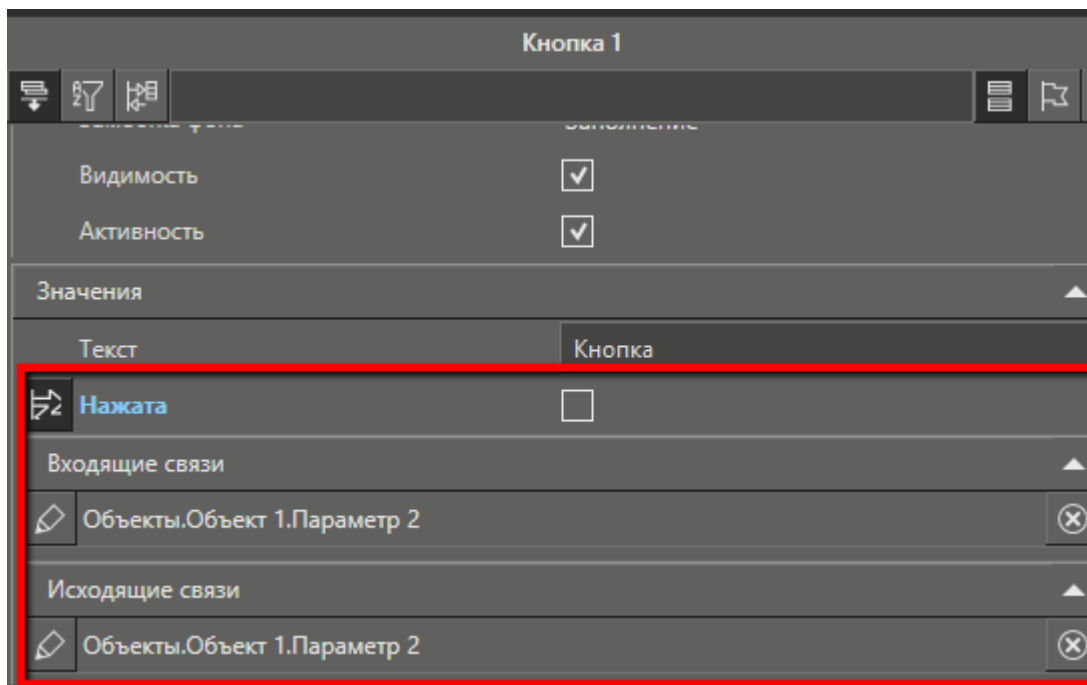
Если элементы Кнопка и Кнопка с фиксацией используются для изменения значения параметра типа BOOL, то необходимо перетащить этот параметр в нужное место рабочей области редактора НМІ правой кнопкой мыши, и в выпадающем списке выбрать требуемый элемент:



После этого в окне появится элемент Кнопка или Кнопка с фиксацией:



Кроме того, будет автоматически установлена двунаправленная связь между параметром проекта и свойством Нажата:

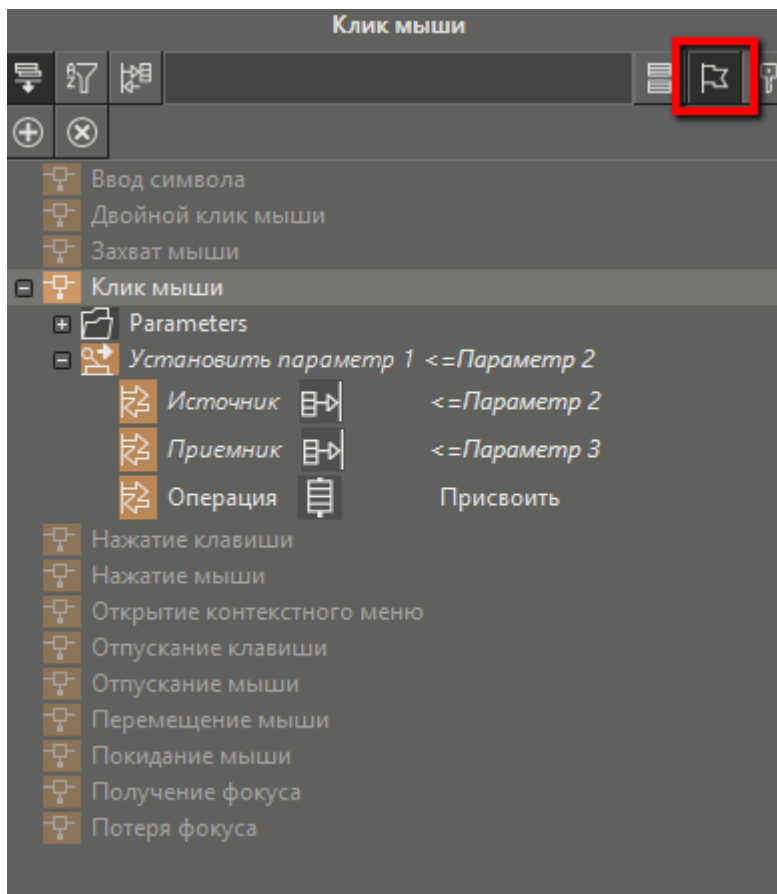


Если в клиенте визуализации в режиме исполнения нажать на Кнопку левой кнопкой мыши, то связанный параметр будет принимать значение TRUE до тех пор, пока Кнопка не будет отпущена.

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

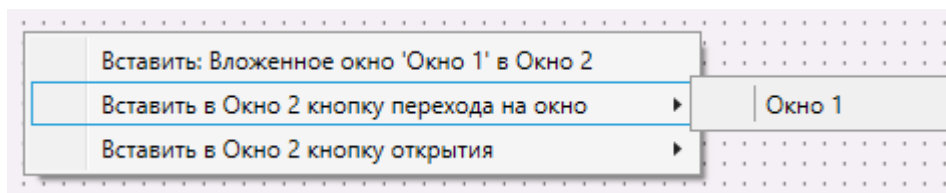
Если при нажатии на кнопку необходимо выполнить какое-либо действие, например, значение одного параметра присвоить значению другого, либо переключить окно, либо открыть всплывающее окно, то в окно необходимо добавить элемент Кнопка. Затем, переключившись в панели свойств в режим работы с событиями, в нужное событие добавить требуемое действие:



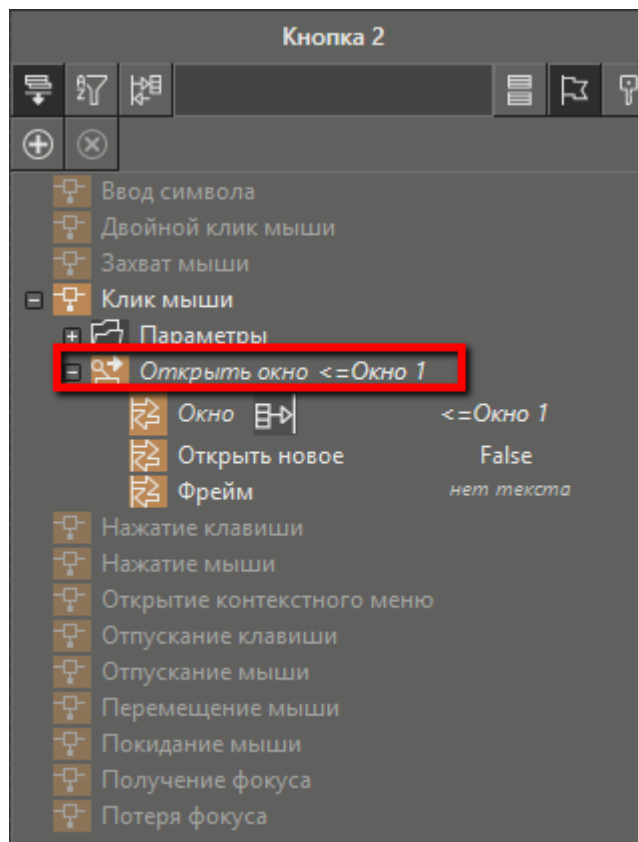
### Переключение окон

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 НМІ, и в появившемся меню выбрать пункт Вставить в [имя редактируемого окна] кнопку перехода на окно - [Имя окна, на которое нужно выполнить переход]:



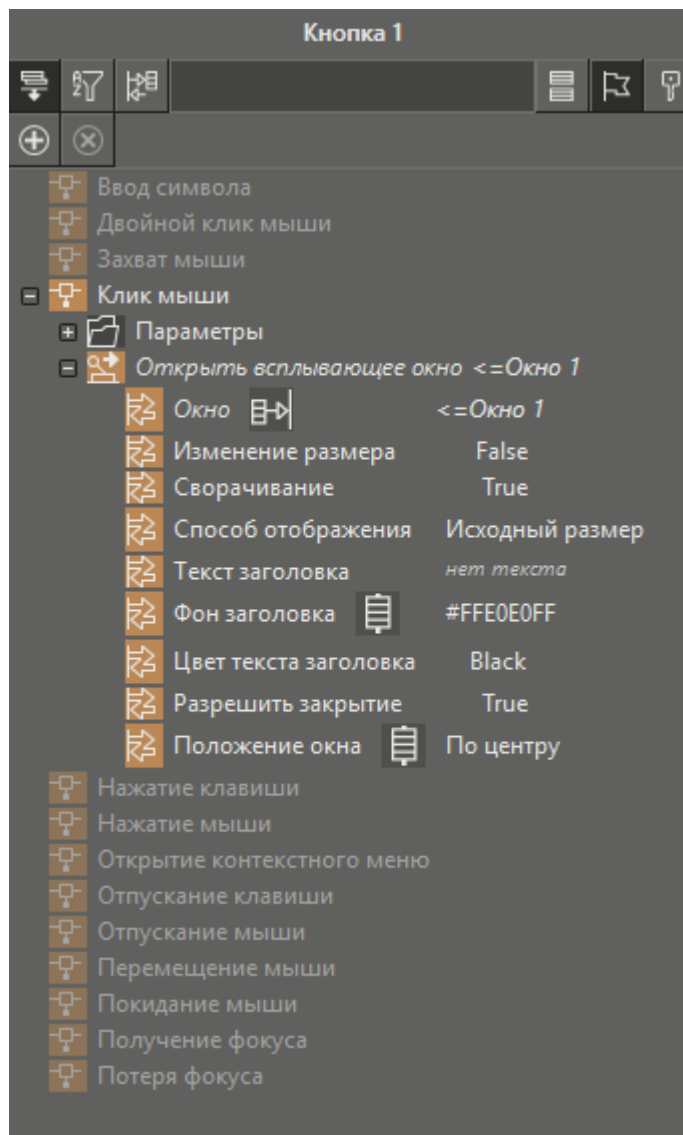
В этом случае кнопка с настроенным действием добавится автоматически:



### Открытие всплывающего окна

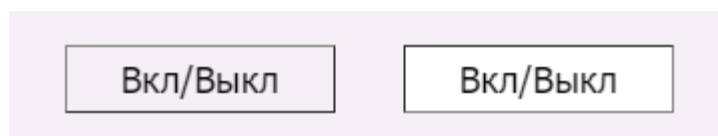
Самый быстрый способ создать кнопку для открытия всплывающего окна - это перетащить правой кнопкой мыши объект, содержащий окно, либо непосредственно само окно в рабочую область редактора HMI, и в появившемся меню выбрать пункт Вставить в [имя редактируемого окна] кнопку открытия - [Имя окна, на которое нужно выполнить переход].

В этом случае кнопка с настроенным действием добавится автоматически:



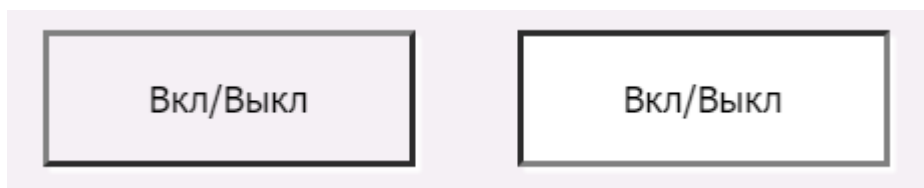
Как сделать нажатие кнопки более заметным

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

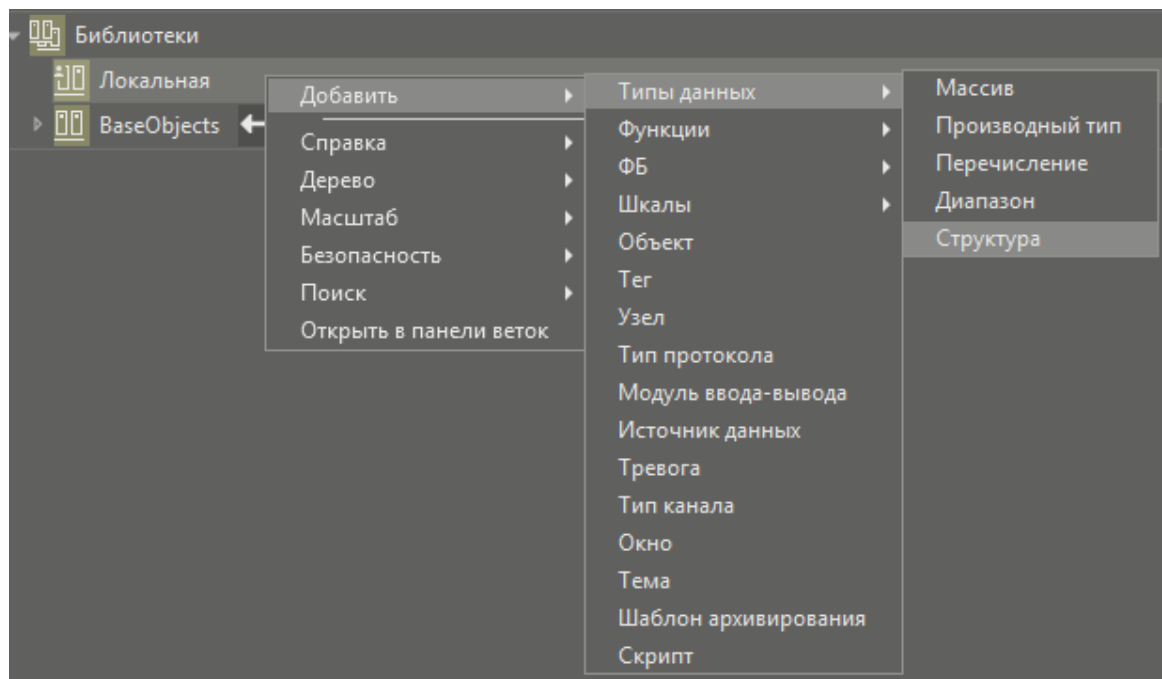
### Постановка задачи

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.

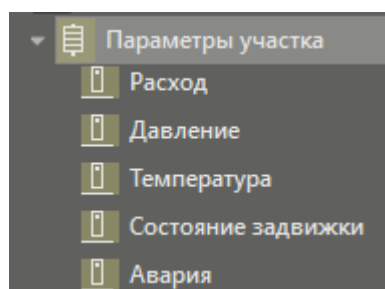
### Формирование библиотечных элементов

#### 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:

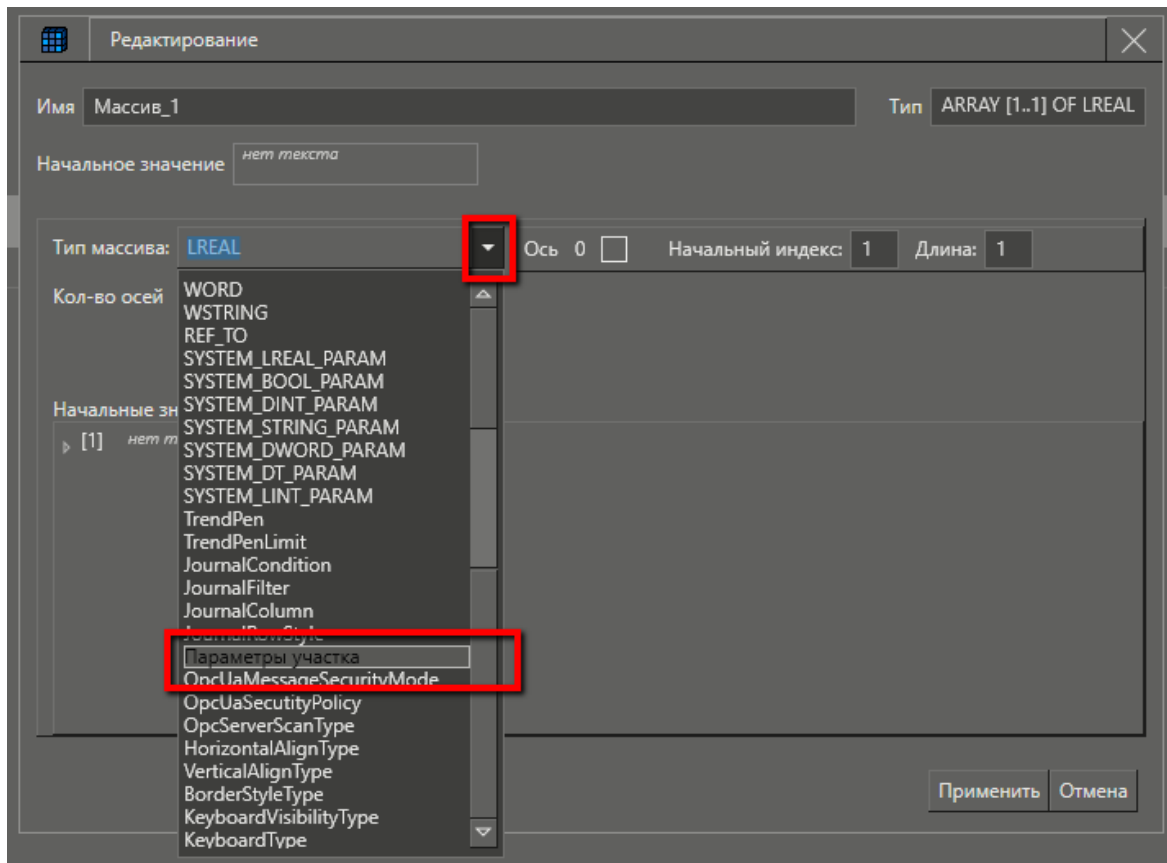


Поле	Тип
Расход	LREAL
Давление	LREAL
Температура	LREAL
Состояние_задвижки	BOOL
Авария	BOOL

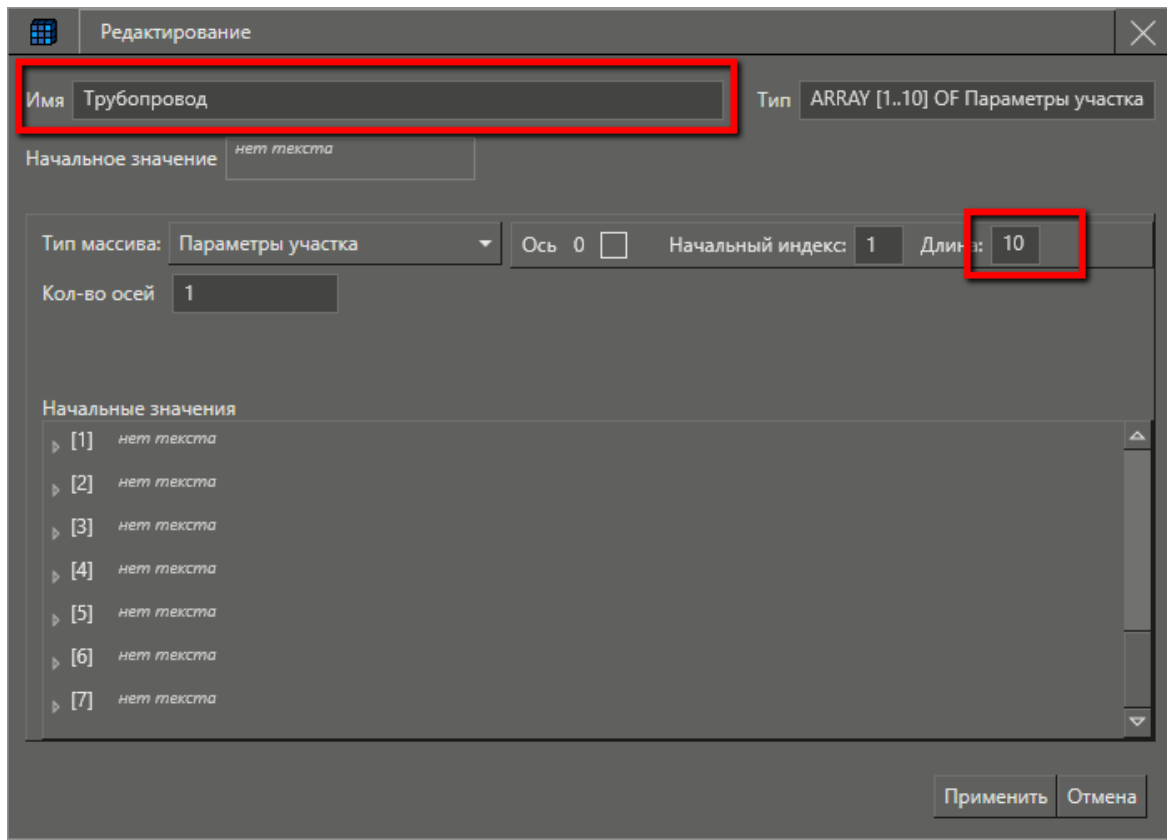
Создание массива структур

На основе ранее созданной структуры в библиотеке создадим массив:



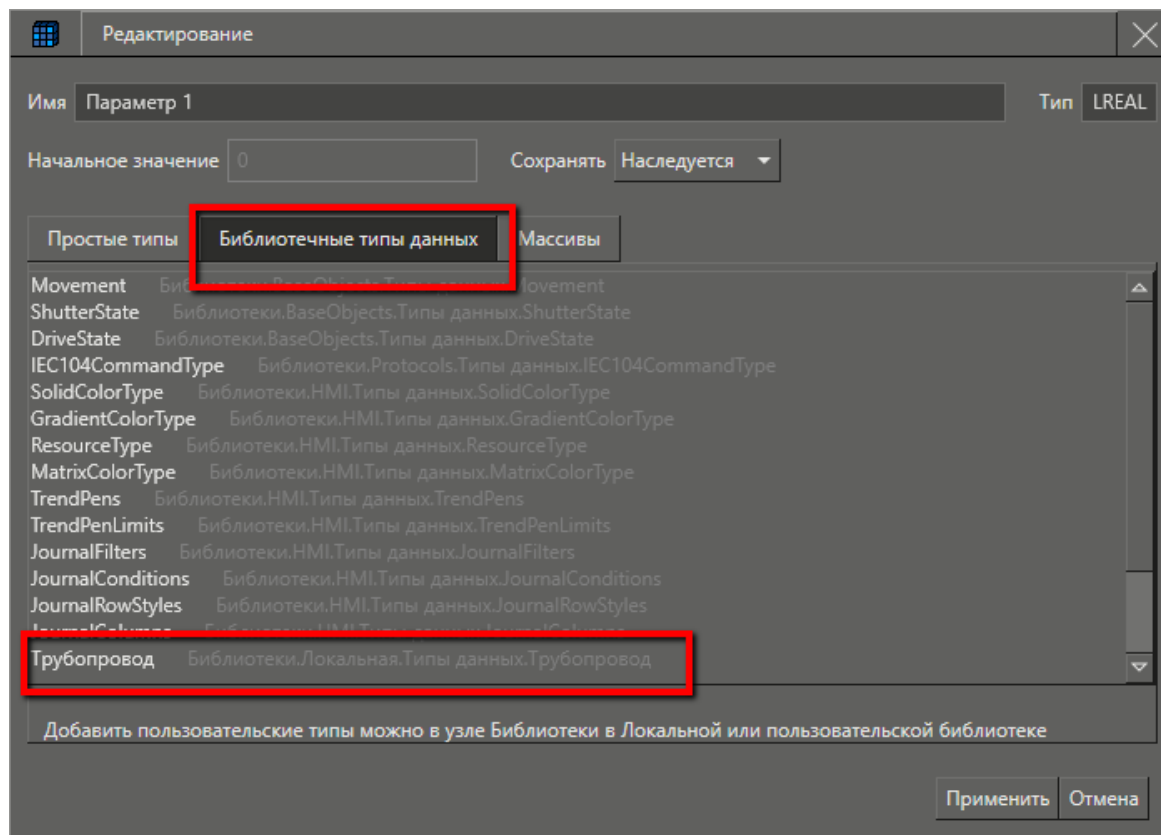


Установим длину массива 10 и изменим имя массива:



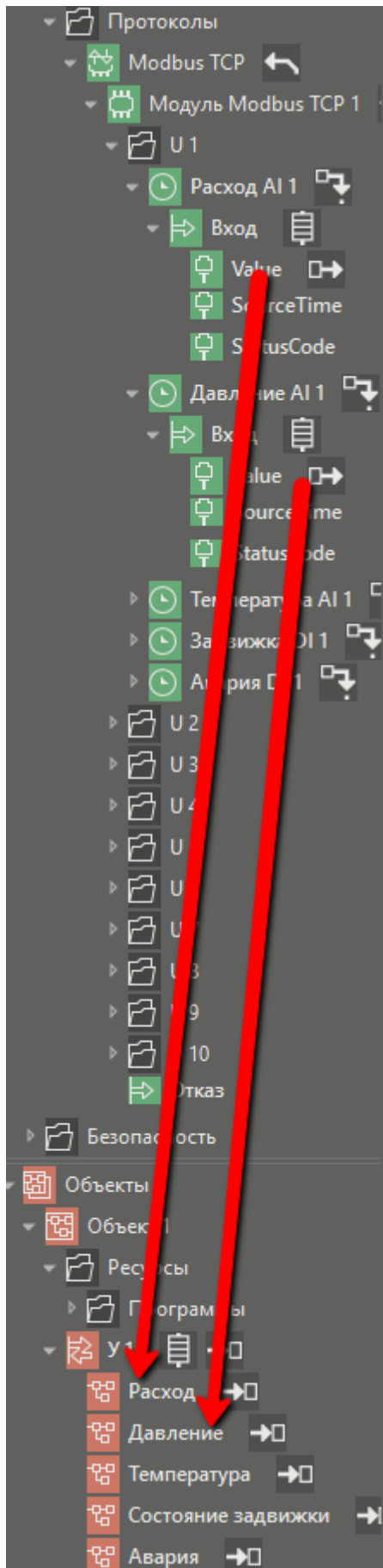
### Создание дерева объектов

Добавим в дерево один параметр типа Параметры участка, назовем его У 1 и один Параметр 1 типа Трубопровод:

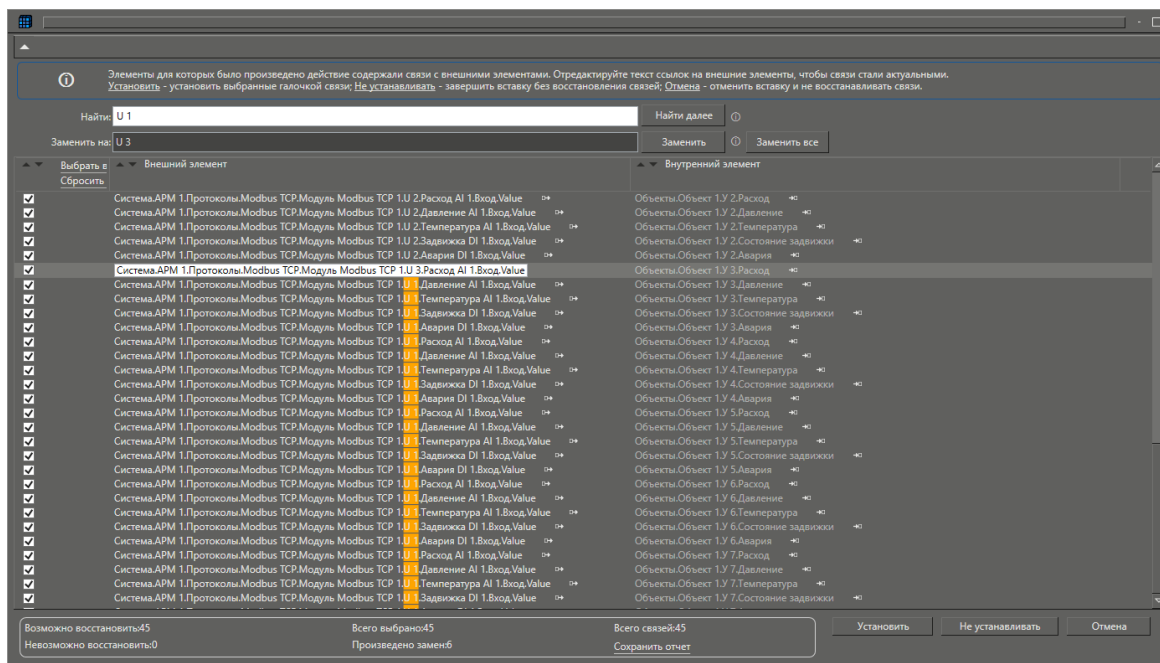


Теперь необходимо обеспечить присвоение данных из дерева система в объект. Допустим, что в примере данные для проекта поступают по протоколу Modbus TCP. Не будем здесь подробно описывать формирование дерева системы, а описание работы с протоколами изложено в разделе Получение и передача данных.

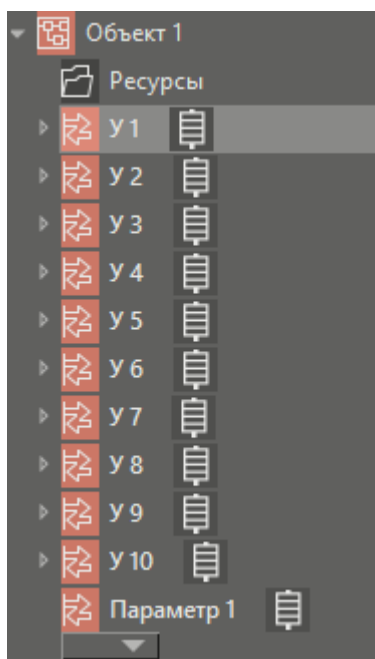
Параметры каналов свяжем с полями структуры параметра У 1:



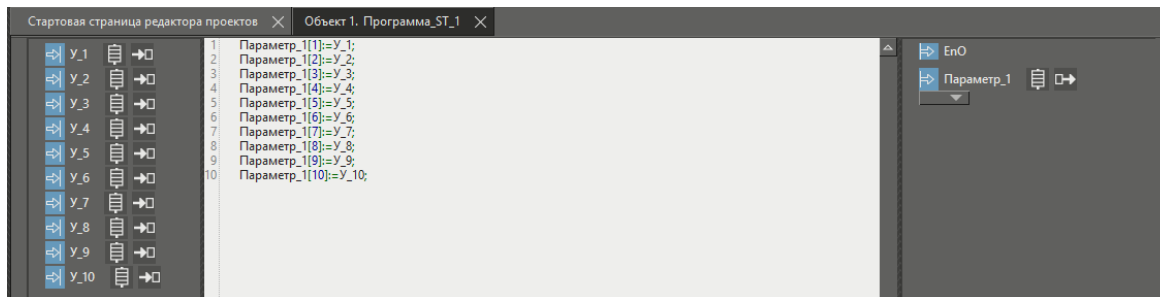
Затем продублируем параметр У 1. Требуется сделать 9 копий. В появившемся диалоговом окне, пользуясь его инструментами, необходимо отредактировать путь к параметрам каналов в дереве системе:



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:

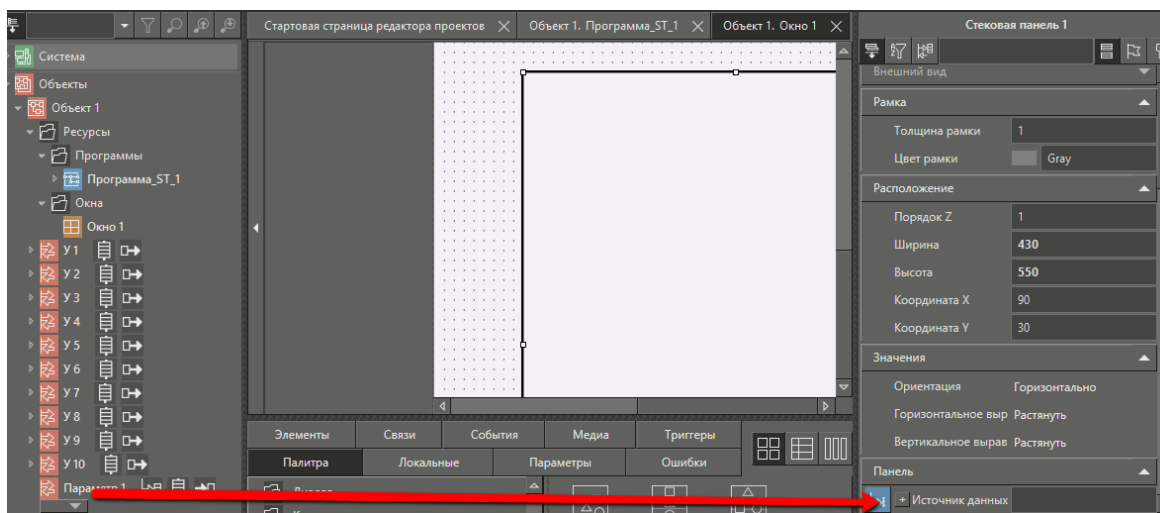


Создание окна

Добавим в объект Окно.

В окно вставляем элемент Стековая панель из категории Панели в палитре редактора НМИ.

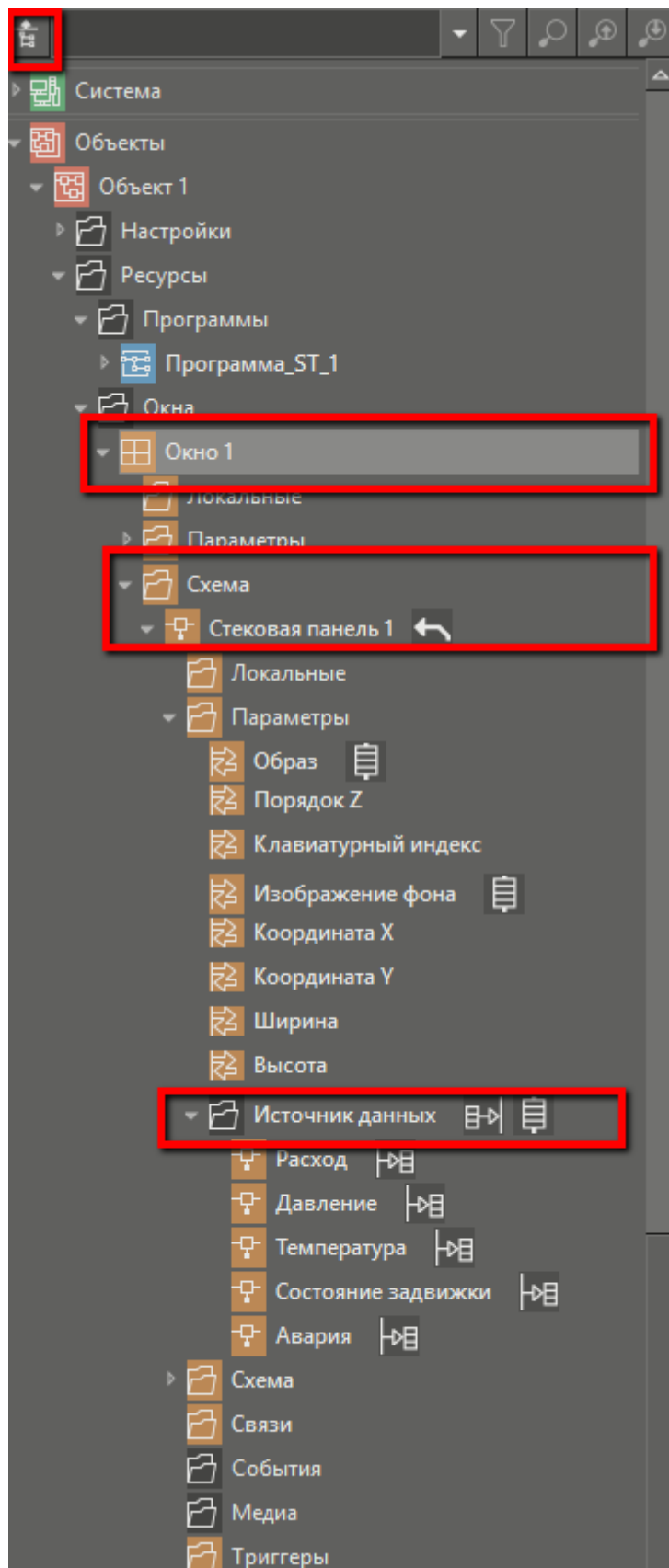
Настраиваем связь между Параметр 1 и свойством панели Источник данных:



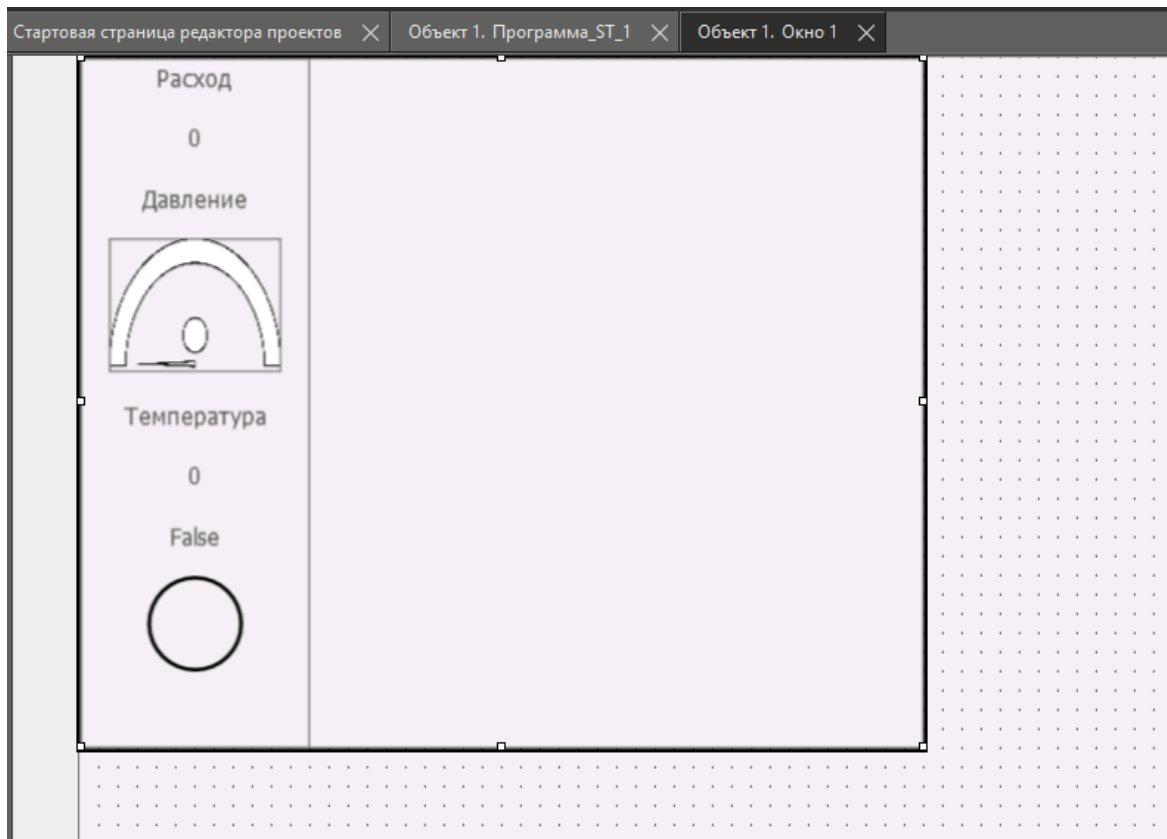
Затем в стековую панель добавляем элемент Панель.

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:





### 8.5.7.Создание пользовательской библиотеки примитивов

Если необходимо модифицировать элементы из палитры редактора HMI, например, категории Примитивы для дальнейшего использования во многих окнах, то необходимо создать в пользовательской библиотеке элемент Окно, в которое добавить элемент и задать ему необходимые настройки.

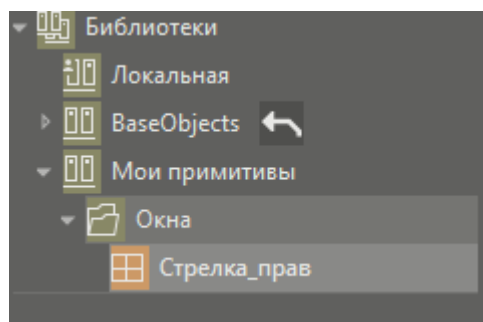
Рассмотрим порядок работы на примере элемента Стрелка. В палитре есть горизонтальная стрелка, указывающая влево. Перечислим порядок операций для создания стрелки указывающей направо.

#### Шаг 1

Создаем в пользовательскую библиотеку, назовем ее Мои примитивы.

#### Шаг 2

В новой библиотеке создаем Окно, назовем его Стрелка\_прав:

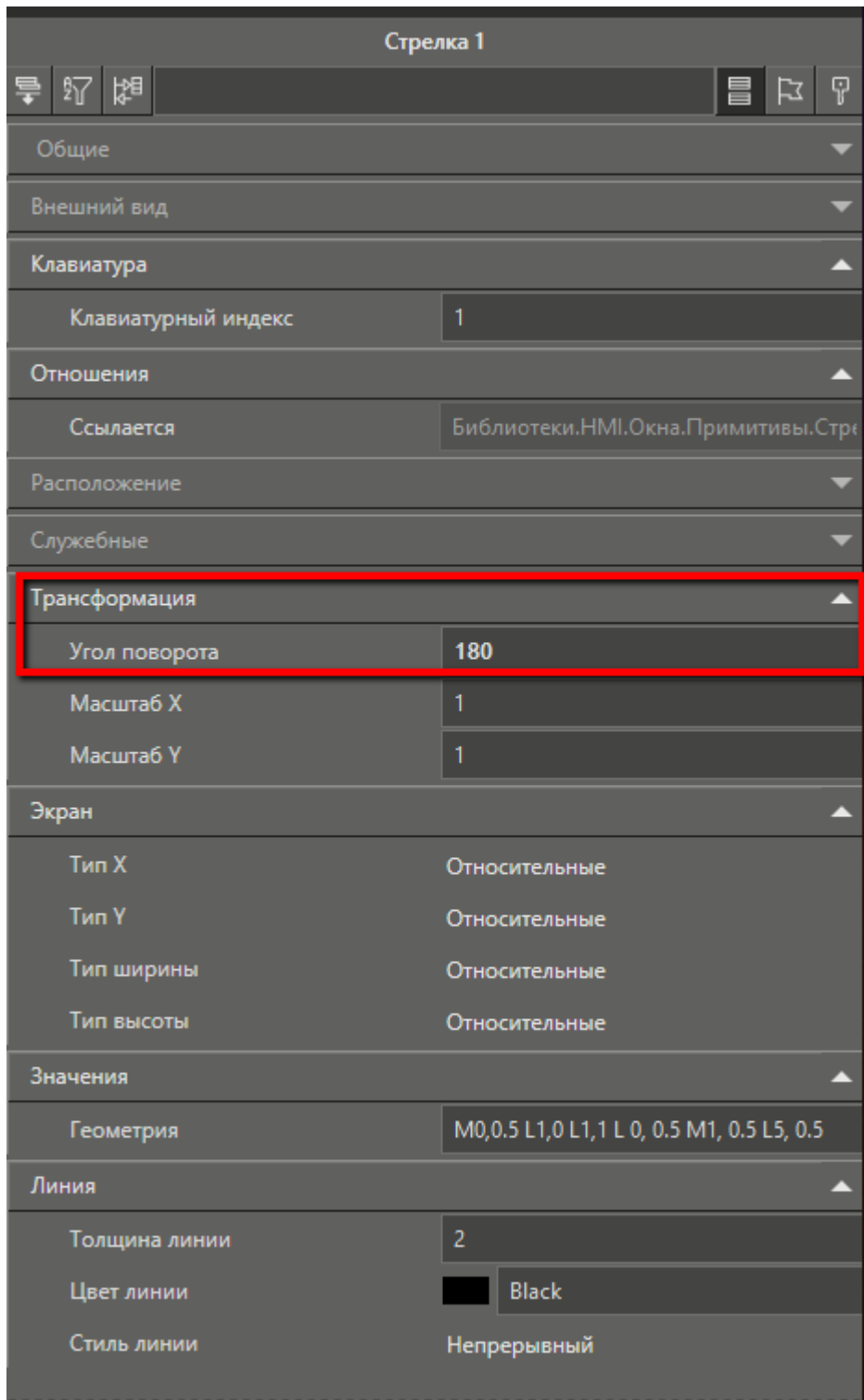


#### Шаг 3

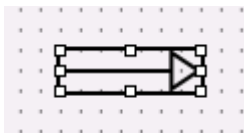
Добавляем в окно элемент Стрелка.

#### Шаг 4

Определяем необходимый внешний вид элемента. В панели свойств элемента задаем значение свойства Угол поворота, равное 180:



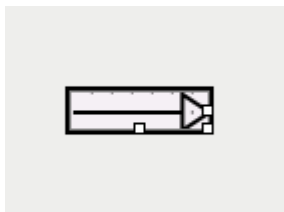
Получим результат:



Разместим элемент в верхнем левом углу.

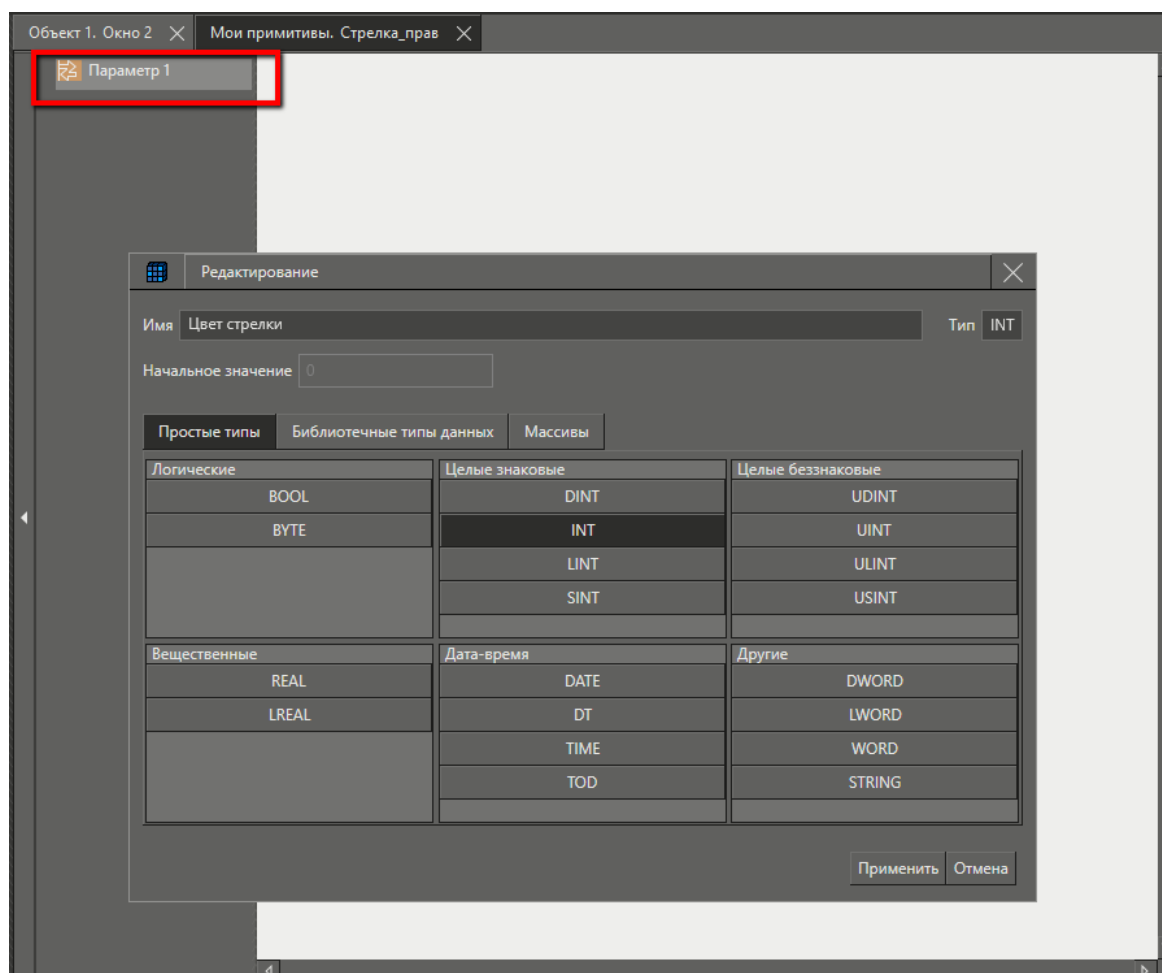
### Шаг 5

Изменим размер окна так, чтобы он совпадал с размером стрелки:

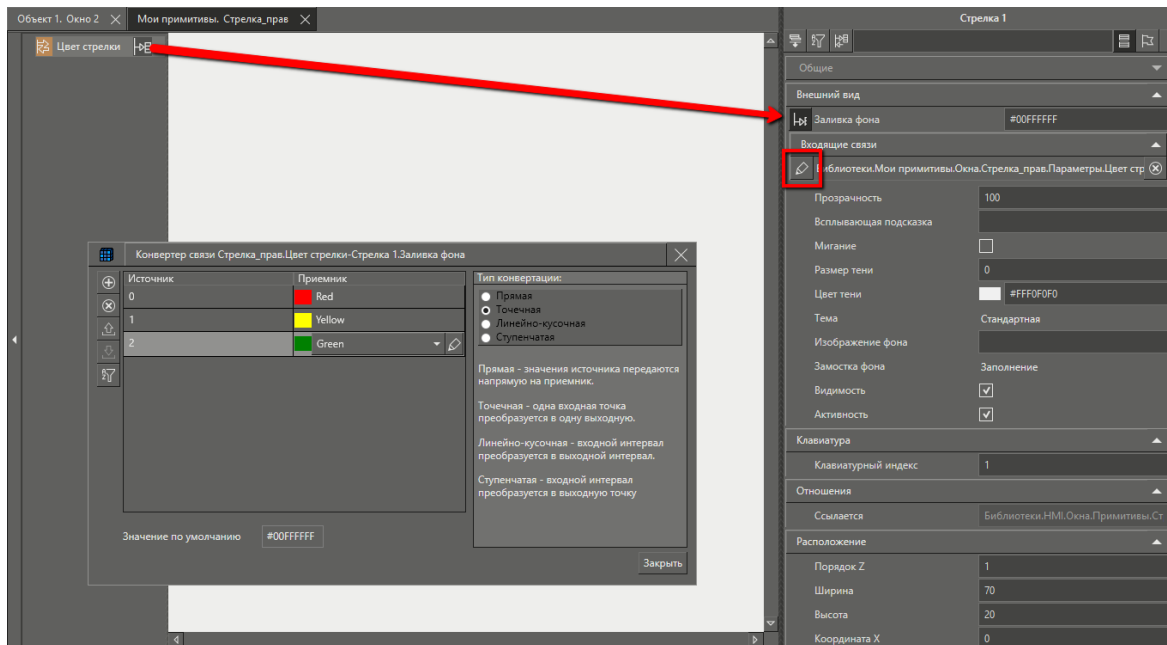


### Шаг 6

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:

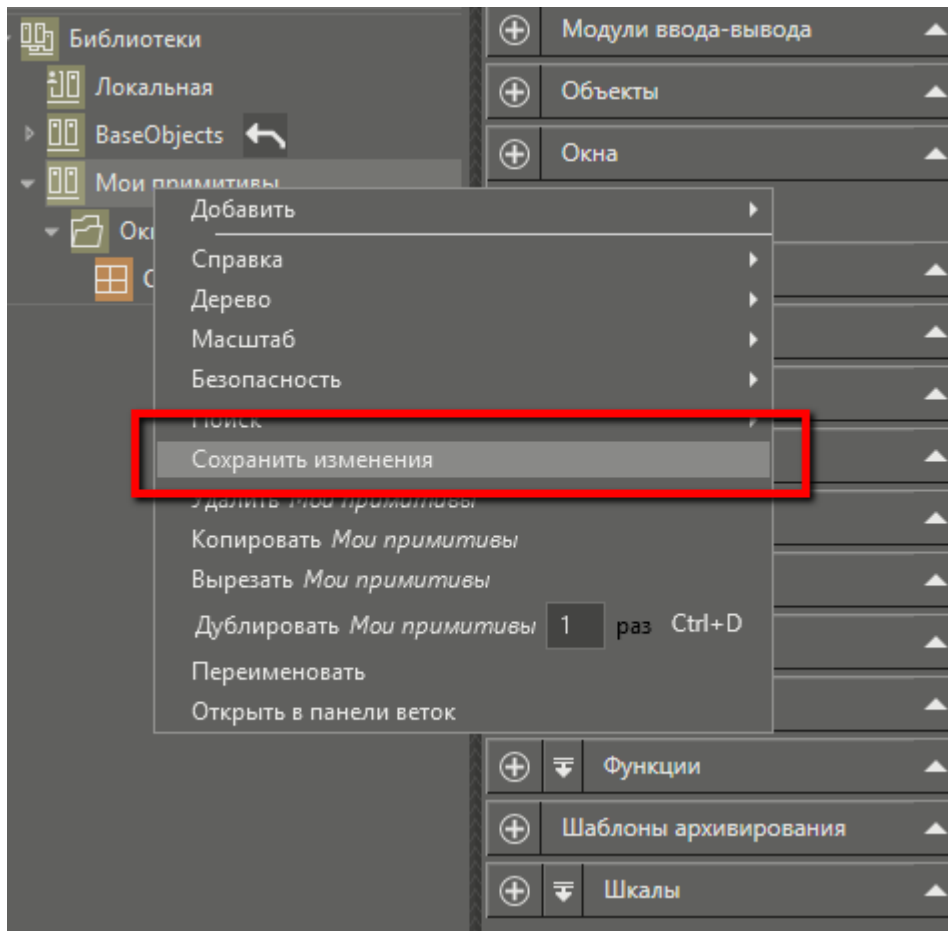


Затем установим связь между свойством стрелки Заливка фона и параметром, и настроим конвертер значений:

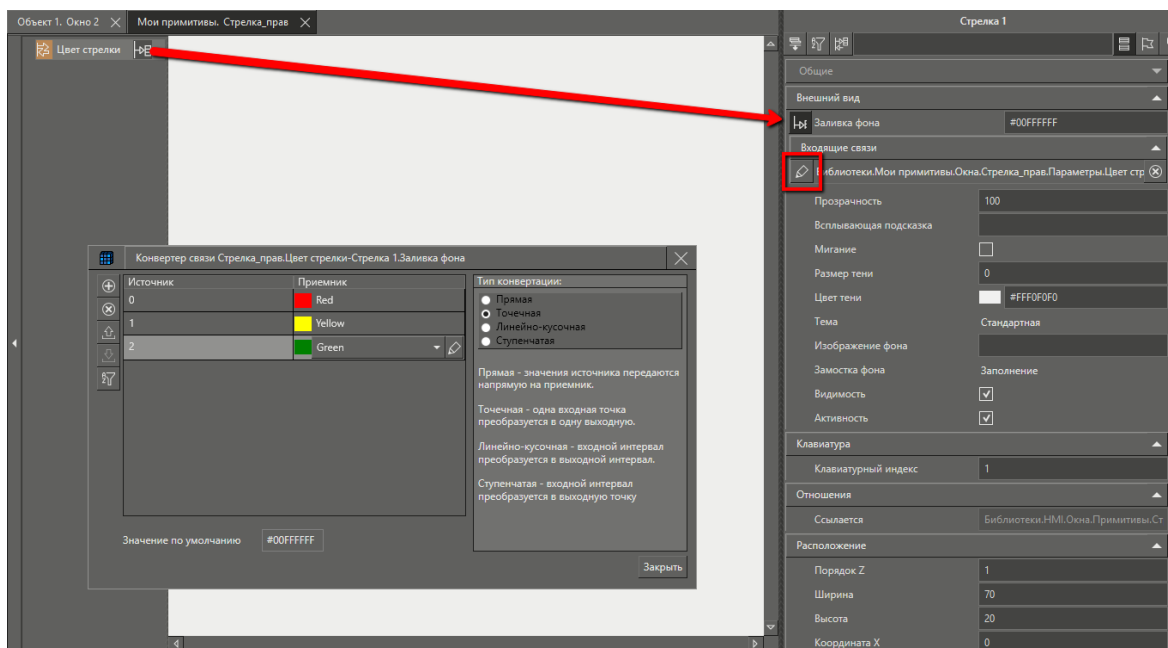


## Шаг 7

Сохраняем библиотеку:



После этого можно добавлять элементы из пользовательской библиотеки в окна:

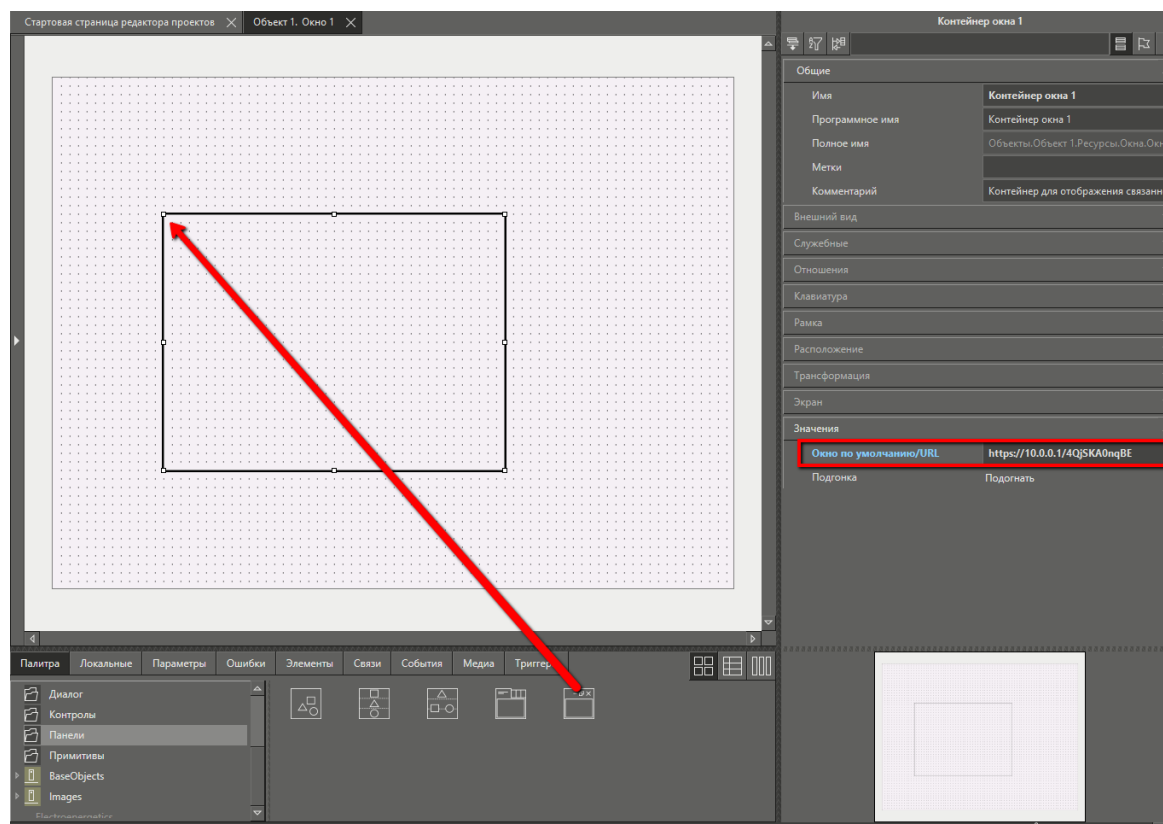


Параметр, добавленный в библиотечное окно, отображается как свойство элемента. Если установить связь между этим свойством и каким-либо параметром проекта, то в режиме исполнения стрелка будет менять свой цвет.

### 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.



## 9. Работа в режиме исполнения

### 9.1. 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:

- mpls – основной процесс исполнительных систем;
- nginx – процесс веб-сервера;
- node\_ms4d – процесс генератора отчетов;
- mpls\_service – вспомогательный процесс, контролирующий работу основного. Если используется отдельный инсталлятор среды исполнения для Windows, то вместо mpls\_service при старте компьютера запускается Windows-служба MS4DService.WinService.exe; при вводе логина пользователя запускается MS4DMonitor.exe для управления работой сервиса.

#### 9.1.1. Рабочая папка среды исполнения

Рабочая папка независимо установленной среды исполнения для ОС Windows носит название Server, а в других ОС – mpls4.

Местонахождение рабочей папки может быть задано в настройках сервиса. По умолчанию, она находится в:

- для ОС Windows: *C:\ProgramData\InSAT\MasterSCADA4DRT[номер версии]/Server*. Эта папка является общей для всех пользователей, так как сервис запускается при старте компьютера;
- для других ОС – */opt/mps4*.



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	
CItCertificateStore	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"> <li>• config.bin – конфигурация;</li> <li>• VMInfo.json – дерево проекта с Id - элементов и типами значений;</li> <li>• _files.xml – список хеш-сумм загруженных файлов;</li> </ul>

	<ul style="list-style-type: none"> <li>• подпапка templates – содержит шаблоны отчетов</li> </ul>
ClCertificateStore	Сертификат UA-клиента
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 IEC61850. Ключи и сертификаты сервера хранятся в папке IEC61850/srv. Ключи и сертификаты клиента хранятся в папке IEC61850/clt.</p> <p>Имя файла ключа клиента должно быть MasterSCADA4Dclt_key.pem. Имя файла сертификата клиента должно быть MasterSCADA4Dclt.cer.</p> <p>Имя файла ключа сервера должно быть MasterSCADA4Dsrv_key.pem. Имя файла сертификата сервера должно быть MasterSCADA4Dsrv.cer.</p> <p>Сертификаты CA должны храниться в папке IEC61850/clt/root для клиента и IEC61850/srv/root для сервера. Сертификаты CA могут иметь любое имя и расширение .cer.</p>
log\0	<p>Содержит лог-файлы исполнительной системы. Состав папки:</p> <p>Подпапка ErrorDumps содержит дампы, которые формируются автоматически при возникновении ошибки.</p> <p>Содержит лог-файлы типа log_18_01_07.txt</p>
logs	Хранятся лог-файлы клиента визуализации
Temp	Содержит временные файлы nginx;
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.
--	--

### 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.

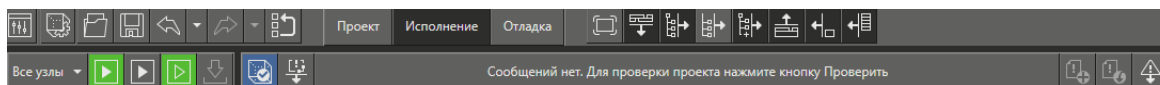
Рассмотрим структуру рабочей папки на примере исполнительной системы для ОС Windows. К существующим файлам и папкам для нулевого экземпляра в рабочую папку будут добавлены следующие подпапки и файлы:

Папка	Назначение
Server\N, где N номер экземпляра	Содержит папку с конфигурацией проекта cfg и файлы, создаваемые в режиме исполнения для узла 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 log_18_01_07.txt, соответствующие работе узла, загруженного в экземпляр N, а также подпапки ErrorDumps и logs

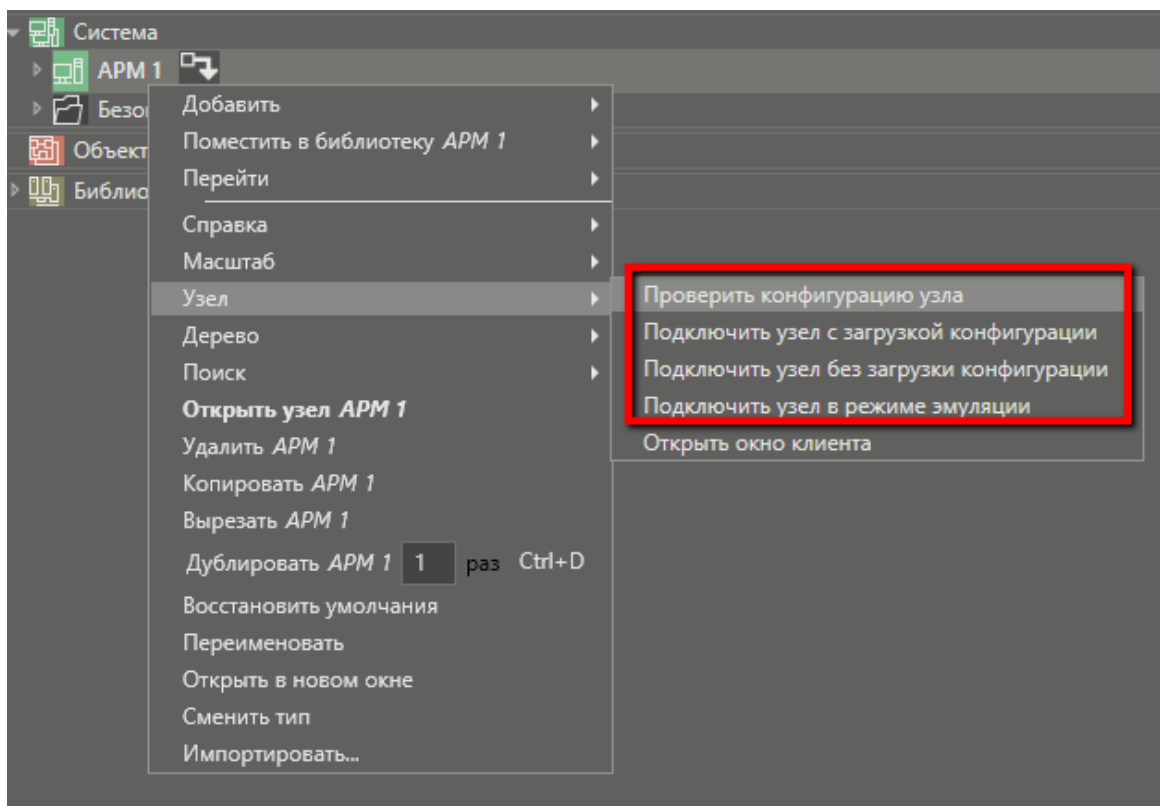
## 9.1.2. Подготовка проекта к запуску в среде исполнения

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 Среда разработки: Подключить, Эмуляция или Проверить.



После выполнения команд контекстного меню узла (см. рис. ниже) произойдет компиляция той части проекта, которая относится к данному узлу:

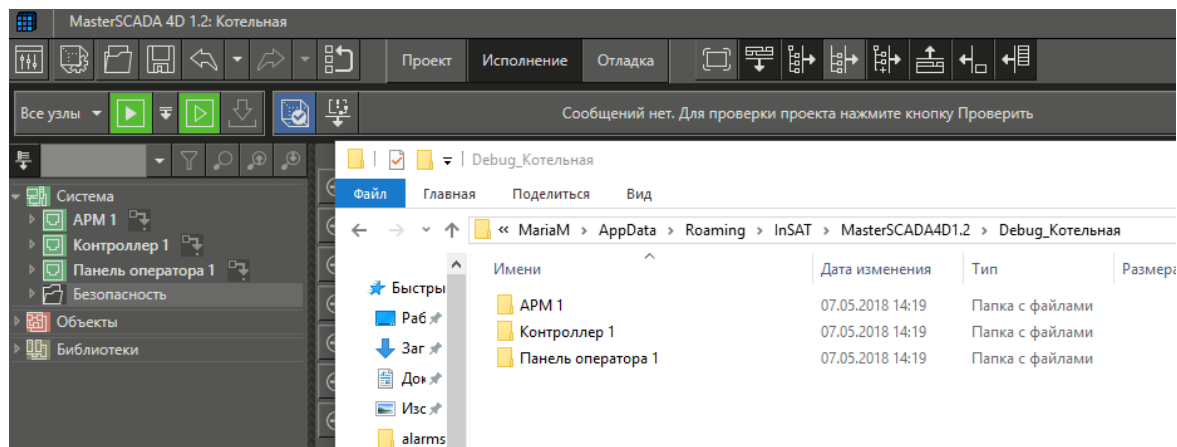


Если узел имеет межузловые связи, то будут также скомпилированы части проекта, относящиеся к связанным узлам.

Место хранения скомпилированных проектов: *c:\Users\[имя пользователя]\AppData\Roaming\InSAT\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.



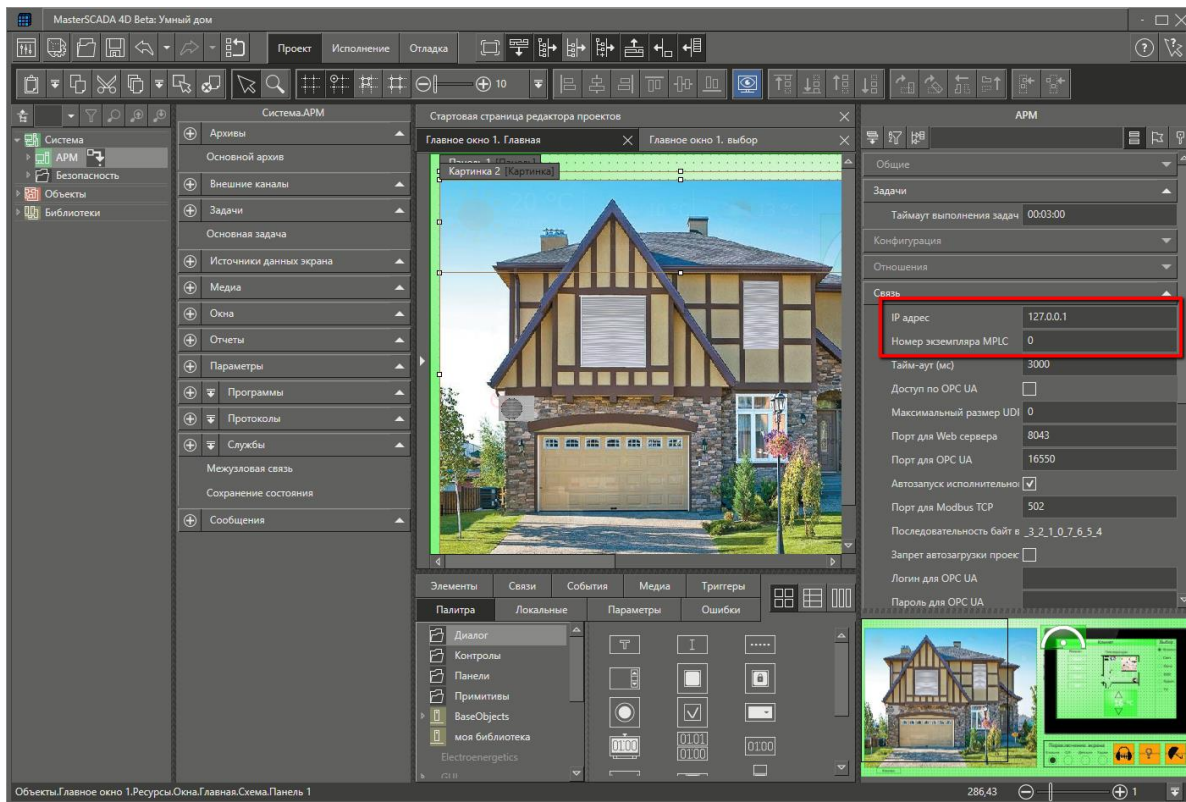
Важно! Скомпилированный проект из папки Debug\_[имя проекта] невозможно открыть в среде разработки.

### 9.1.3. Автоматическая загрузка проекта

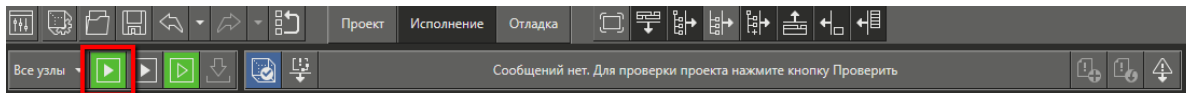
Рассмотрим два варианта работы: в первом случае исполнительная система и среда разработки работают на одном компьютере; во втором – на разных устройствах.

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).

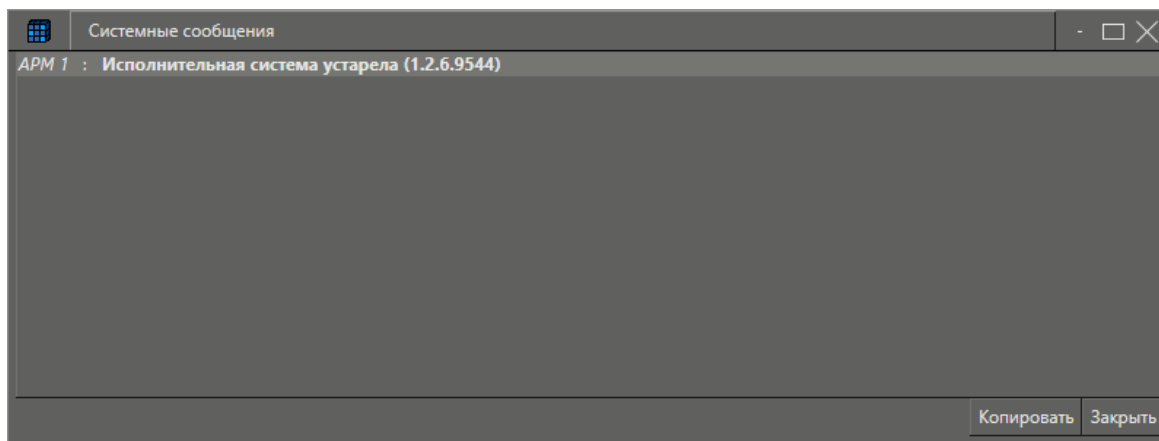
Среда разработки и среда исполнения работают на разных устройствах

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)\InSAT\MasterSCADA 4D 1.2\bin\Config\MasterPLC\PLC`. Если среду исполнения не активировать, то она будет работать с ограничениями демоверсии.

Инсталлятор демоверсию среды исполнения для Windows можно скачать с сайта компании "ИнСАТ". Коммерческую версию для ОС Windows необходимо заказать по электронному адресу [scada@insat.ru](mailto:scada@insat.ru)

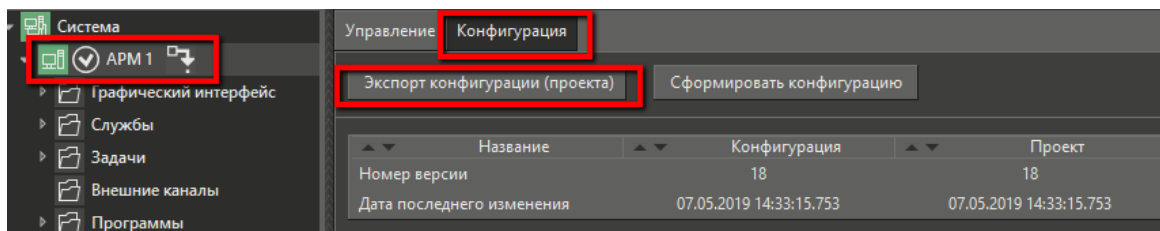
## 9.1.4. Ручная загрузка проекта в среду исполнения

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.

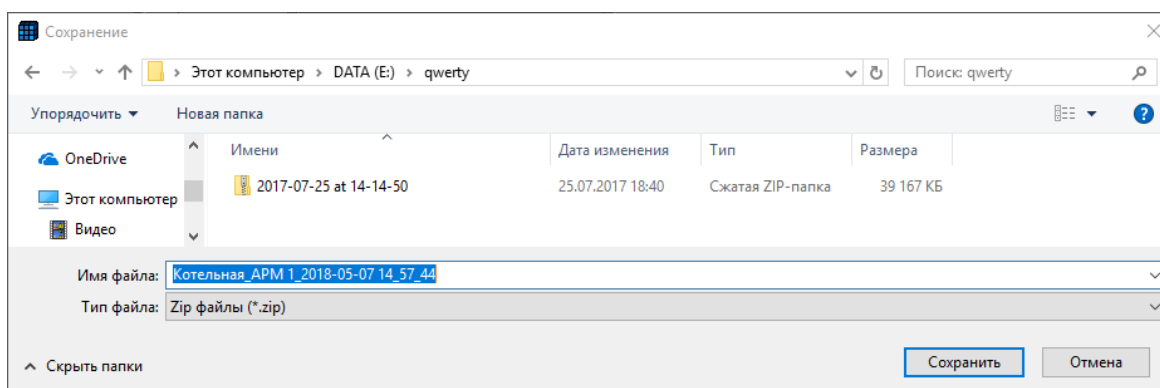
Среда исполнения автоматически запускает последний загруженный в нее проект. Под проектом понимается папка, которая создается в результате компиляции проекта. Для каждого узла создается своя папка.

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:



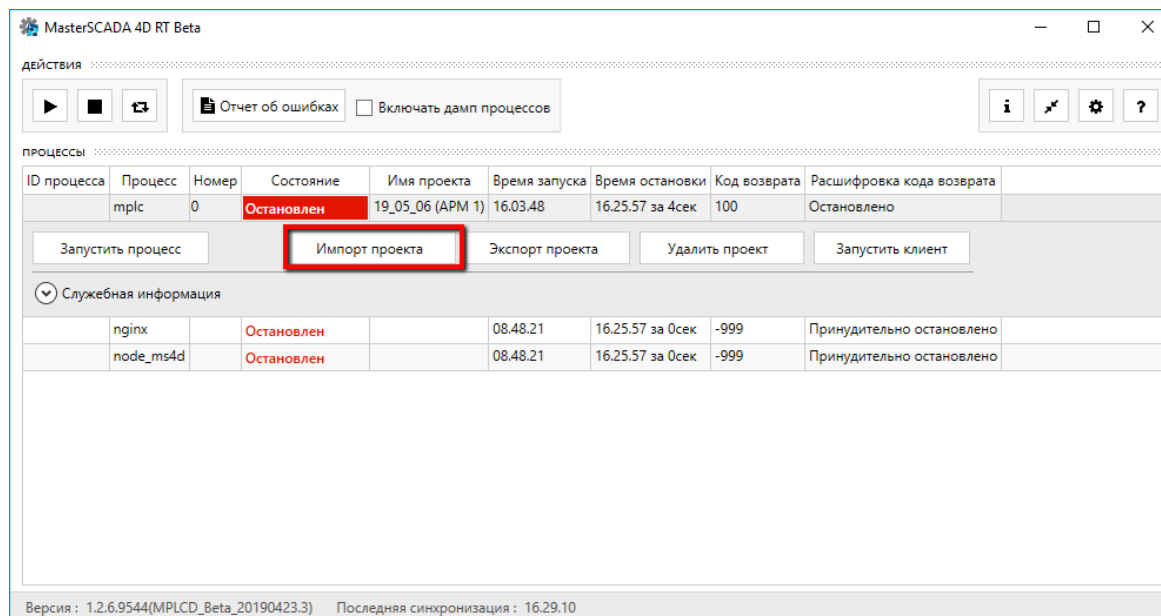
После нажатия на кнопку Сохранить создается архив, имеющий, по умолчанию, имя : [название проекта]\_[имя узла]\_[дата и время экспорта].

После этого содержимое архива необходимо разместить в рабочей папке среды исполнения.

## ОС 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 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\InSAT\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\InSAT\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.

### 9.1.5. Восстановление работы узла

Работу проекта в среде исполнения обеспечивают три процесса:

- `mplc` – основной процесс исполнительской 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.

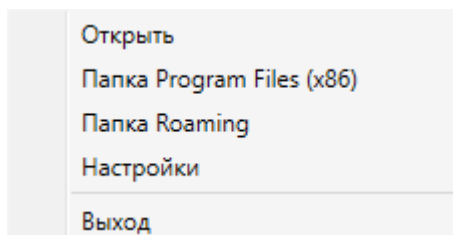
После восстановления связи с узлом, клиент визуализации переключится к среде исполнения автоматически.


Управлять работой процессов среды исполнения можно при помощи MasterSCADA 4D Monitor.

### 9.1.6. MasterSCADA 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.

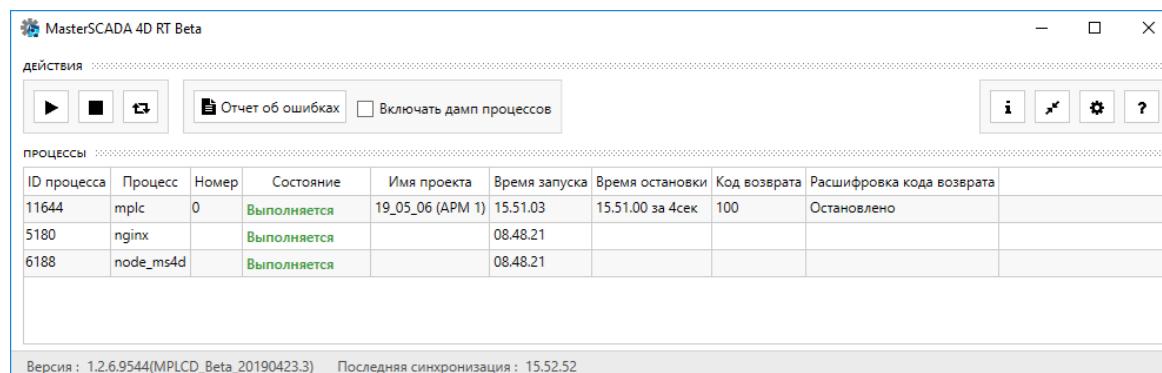
После установки и запуска среды исполнения, в области уведомлений панели задач появляется значок  (этот значок присутствует всегда, когда запущена среда исполнения, за исключением случая, когда пользователь вышел из приложения MasterSCADA 4D Monitor в текущей сессии работы среды исполнения). При нажатии правой кнопки мыши откроется 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. C:\Program Files (x86)\InSAT\MasterSCADA 4D RT
Папка Roaming	Открывает папку, в которой хранится вспомогательная информация C:\Users\[имя пользователя]\AppData\Roaming\InSAT\MasterSCADA4DRT
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 c:\Program Files (x86)\InSAT\MasterSCADA 4D RT\MS4DMonitor.exe

## 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.
	Запускает формирование отчета об ошибках среды исполнения. 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.
Флаг Включать дамп процесса	Если флаг установлен, то сформируется дамп процессов среды исполнения. Этот флаг необходимо устанавливать в случаях, если отчет формируется из-за некорректной работы с другими приложениями и заметного ухудшения быстродействия среды исполнения.
Таблица Процессы	Содержит информацию о запущенных процессах, относящихся к работе среды исполнения: <ul style="list-style-type: none"> <li>• ID процесса – отображает ID процесса в диспетчере задач;</li> <li>• Процесс – имя процесса;</li> </ul>

	<ul style="list-style-type: none"> <li>• 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;</li> <li>• Состояние – состояние процесса: Выполняется, Остановлен и т.п.;</li> <li>• Имя проекта - показывает имя работающего проекта;</li> <li>• Время запуска, Время остановки – указывается время работы процесса;</li> <li>• Код возврата – указывает номер ошибки, если процесс запустить не удалось или в процессе работы возникли проблемы;</li> <li>• Расшифровка кода возврата – текстовое описание ошибки.</li> </ul> <p>При нажатии на строку процесса, открываются дополнительные строки, позволяющие выполнять действия с процессом, например, остановить его. Для процесса mrlc есть возможность осуществлять работу с проектом, например, выполнить загрузку проекта вручную, либо получить информацию о загруженном проекте, и необходимых для него опциях. Смотрите также раздел Работа с процессом mrlc</p>
	Позволяет скопировать в буфер обмена дополнительную информацию о выделенном процессе.
	Сворачивает дополнительные строки, появляющиеся после нажатия на строку процесса в таблице.
	Открывает окно настройки среды исполнения.
	Открывает справочную систему.
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
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

### 9.1.6.1. Окно настройки среды исполнения

Если в приложении MasterSCADA 4D Monitor нажать кнопку , то откроется окно управления средой исполнения.

Вид окна:

Настройки MasterSCADA 4D RT 1\_2

**ИСПОЛНИТЕЛЬНАЯ СИСТЕМА**

Автозапуск исполнительной системы:

Количество экземпляров mplc:

Рабочая папка:  
 ... ->

Параметры запуска mplc:

**КЛИЕНТ**

Автозапуск клиента:

Адрес сервера:

Полноэкранный режим для клиента:

Параметры запуска клиента:

**ВНЕШНИЙ ВИД**

Элемент	Описание
<b>Исполнительная система</b>	
Автозапуск исполнительной системы	Если флаг установлен, то среда исполнения запускается автоматически при старте ОС.
Количество экземпляров mplc	Если требуется запустить несколько исполнительных систем на одном компьютере, то в данном поле необходимо указать их количество.



Рабочая папка	Указывается место хранения рабочей папки Server. <b>Важно!</b> Путь к рабочей папке среды исполнения не может содержать в себе буквы русского алфавита.
Параметры запуска mp1c	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: 127.0.0.1:8043/index.html?user=sa&amp;password=123, 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 НМІ - v1+v2, то откроется клиент визуализации выполненный в версии НМІ v2.</p>
Полноэкранный режим для клиента	Если флаг установлен, то клиент запустится во весь экран. В полноэкранном режиме клавиша F11 не работает
Параметры запуска клиента	<p>Могут быть задаваны следующие ключи:</p> <ul style="list-style-type: none"> <li>-d – если ключ установлен, то в процессе работы по кнопке F12 отрываются средства разработчика;</li> <li>-t "text" – устанавливает заголовок окна. Заголовком окна будет текст, указанный в кавычках. Если текст не содержит пробелы, то кавычки допускается не ставить;</li> <li>--enable-logging – обеспечивает вывод некоторых ошибок в консоль.</li> </ul> <p>Ключи для настройки положения и размера окна клиента:</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;

-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;

-w [число] - устанавливает ширину окна (в пикселях);

-h [число] - устанавливает высоту окна (в пикселях);

Если данные параметры не заданы, то окно открывается в верхнем левом углу с наибольшими шириной и высотой. Для переключения в полноэкранный режим можно воспользоваться клавишей F11.

-f - включает полноэкранный режим. В полноэкранный режиме клавиша F11 не работает;

Ключи, которые будут работать только в случае если выбран тип транслятора NMI v2:

-sr -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. Если путь к файлу не содержит пробелы, то кавычки допускается не ставить;

-c - открывает конфигурационное окно, позволяющее установить папку для сохранения файлов по умолчанию, принтер по умолчанию;

-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 авто-

	ризации 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.
Запустить клиент	По нажатию на кнопку откроется клиент визуализации, разработанный компанией "ИнСАТ".
Внешний вид	
Сменить тему	Меняет цветовое решение приложения. В данной версии не поддерживается.
Сохранить и перезапустить процессы	Останавливает ранее запущенные процессы и запускает их вновь. Количество запускаемых процессов с именем mpls соответствуют значению, введенному в поле Количество экземпляров mpls.
Сохранить	Сохраняет введенные настройки.
Отмена	Закрывает окно настроек, введенные изменения не сохраняются.

### 9.1.6.2. Работа с процессом mpls

Процесс mpls – основной процесс исполнительной системы.

Если в приложении MasterSCADA 4D Monitor в таблице процессов выделить mpls, то откроются дополнительные строки, содержащие элементы управления:

ID процесса	Процесс	Номер	Состояние	Имя проекта	Время запуска	Время остановки	Код возврата	Расшифровка кода возврата
1684	mpls	0	Выполняется	19_05_06 (АРМ 1)	16.59.06	16.25.57 за 4сек	100	Остановлено



Остановить процесс
Импорт проекта
Экспорт проекта
Удалить проект
Запустить клиент

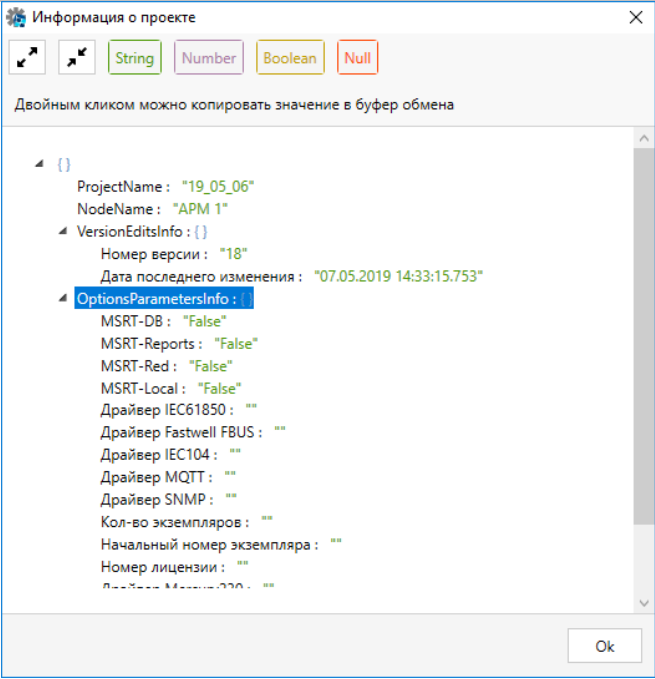
⬆ Службная информация

Информация о проекте
Информация об опциях
О процессе: Вирт\_память = 5488640; Поток = 1 шт;

Описание элементов управления:

Элемент	Описание

Остановить процесс/Запустить процесс	Кнопка останавливает и запускает процесс mpls.
Импорт проекта	Кнопка позволяет осуществлять импорт скомпилированного проекта для запуска в среде исполнения. Подробное описание в разделе Ручная загрузка проекта в среду исполнения
Экспорт проекта	Кнопка позволяет экспортировать загруженный проект в какое-либо место, указанное пользователем. После нажатия на кнопку откроется стандартное диалоговое окно Windows , с помощью которого можно выбрать место экспорта проекта. <b>Важно!</b> При экспорте проекта архивы и файл данных для горячего рестарта не экспортируются.
Удалить проект	Кнопка позволяет удалять загруженный проект.
Запустить клиент	Кнопка запускает клиент визуализации.
Переключатель  / 	Скрывает/раскрывает доступ к группе Службная информация.
Информация о проекте	Кнопка открывает окно, в котором отображается дополнительная информация о проекте. В группе OptionsParametrInfo представлены опции, необходимые для корректной работы проекта.

	
Информация об опциях	Кнопка открывает окно, в котором отображается информация об установленной версии среды исполнения, номере ключа, а также доступных дополнительных опциях
О процессе	В строке отображается объем, занимаемый процессом в виртуальной памяти, а также количество запущенных потоков.

### 9.1.7. Запуск исполнительной системы

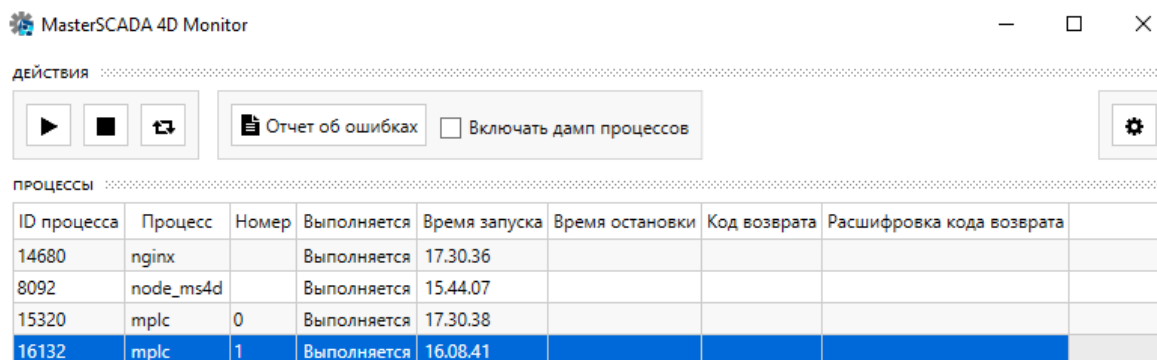
Если исполнительная система установлена как независимое приложение, то при старте операционной системы, по умолчанию, среда исполнения запускается автоматически.

Среду исполнения можно запустить также при помощи приложения MasterSCADA 4D Monitor, либо при помощи ярлычков в меню Пуск – MasterSCADA 4D RT.

### 9.1.8. Запуск нескольких узлов на одном компьютере

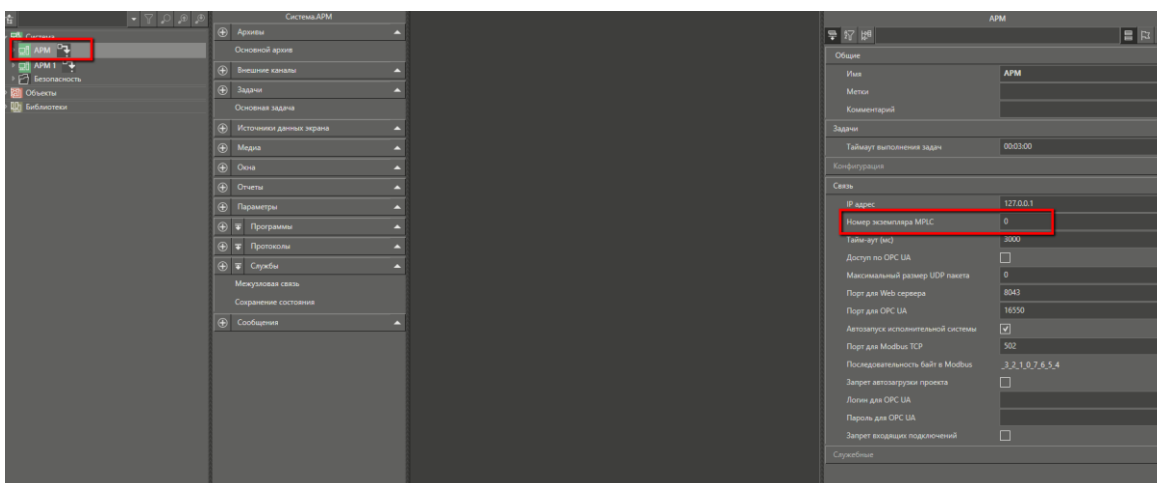
Для запуска нескольких узлов на одном компьютере, работающем под ОС Windows, можно воспользоваться приложением MasterSCADA 4D Monitor, в настройках которого предварительно устанавливается количество экземпляров исполнительной системы, которые требуется запустить одновременно. При этом в дереве системы также должно быть создано необходимое количество узлов. Затем в настройках узлов в свойстве Номер экземпляра mpIc следует установить номера процессов, которые указаны в приложении MasterSCADA 4D Monitor.

Вид MasterSCADA 4D Monitor:

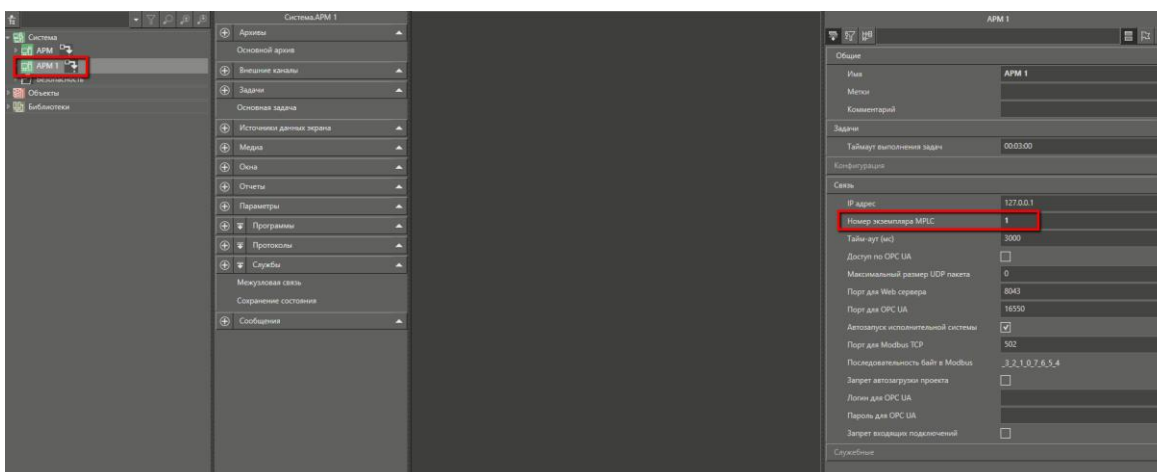


ID процесса	Процесс	Номер	Выполняется	Время запуска	Время остановки	Код возврата	Расшифровка кода возврата
14680	nginx		Выполняется	17.30.36			
8092	node_ms4d		Выполняется	15.44.07			
15320	mplc	0	Выполняется	17.30.38			
16132	mplc	1	Выполняется	16.08.41			

Настройки для АРМ, который должен запускаться в нулевом процессе:



Настройки АРМ 1, который должен запускаться в первом экземпляре:

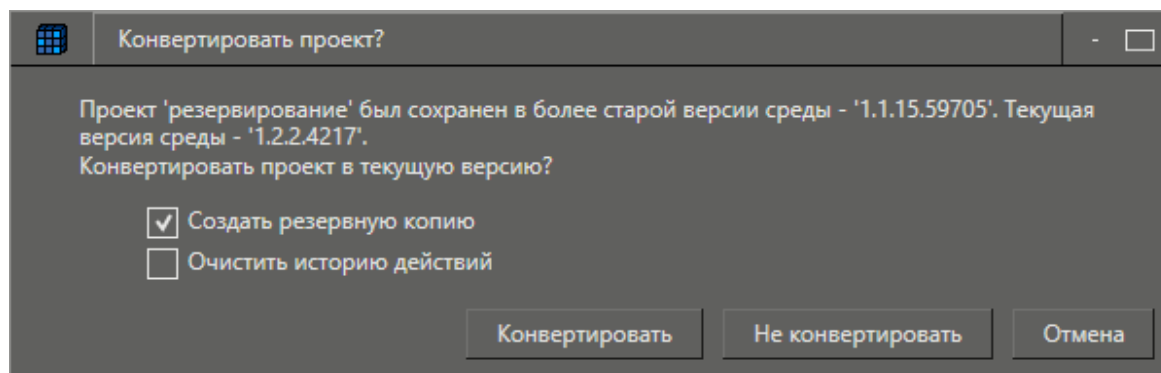


Загрузить проект в среды исполнения можно как автоматически, так и вручную.

## 9.1.9.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.

### 9.1.10. 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 демо-версии, 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 Меркури230		
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 mpIc - откроется дополнительная информация.
3. Раскройте строки Служебная информация
4. Нажмите кнопку Информация о проекте - откроется окно, в котором в группе OptionsParametrInfo отображаются опции, необходимые для корректной работы проекта.

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 [scada@insat.ru](mailto:scada@insat.ru)

### 9.1.11. Коды ошибок среды исполнения

В случае возникновения неисправности, исполнительная системы выдает код ошибки. В исполнительной системе для ОС Windows код ошибки отображается в приложении MasterCADA 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	Описание
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	
-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	

## 9.1.12. 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.

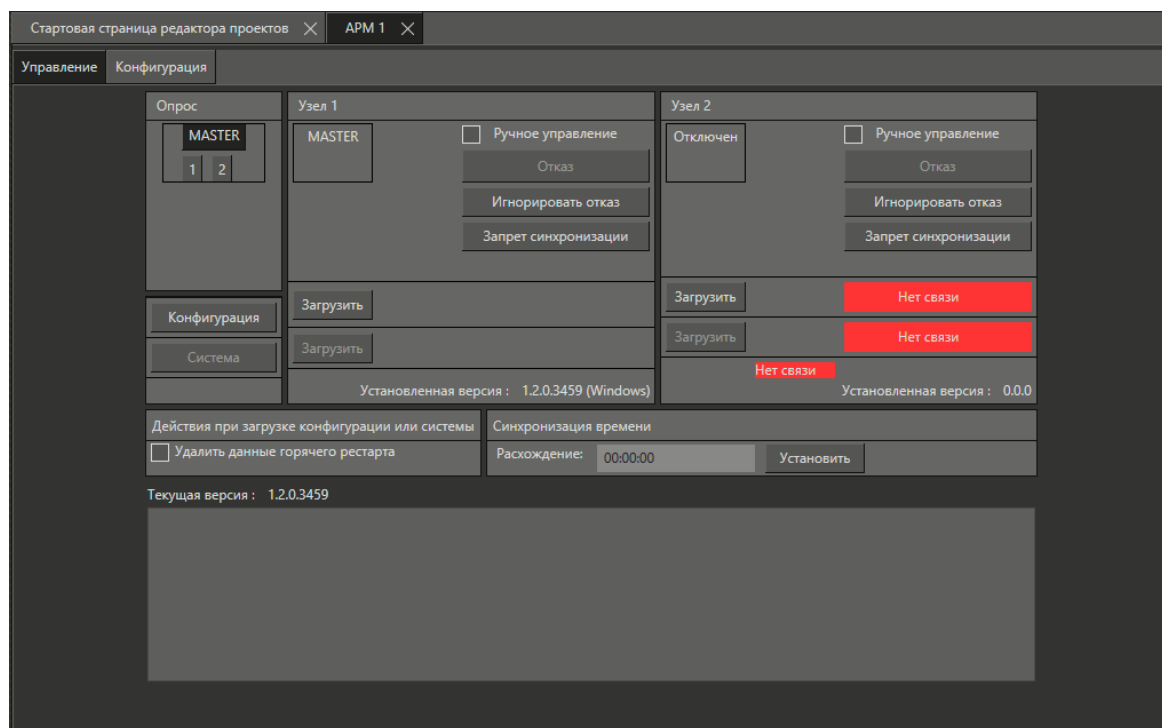
Панель содержит две вкладки:

- Управление
- Конфигурация

### 9.1.12.1. 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:

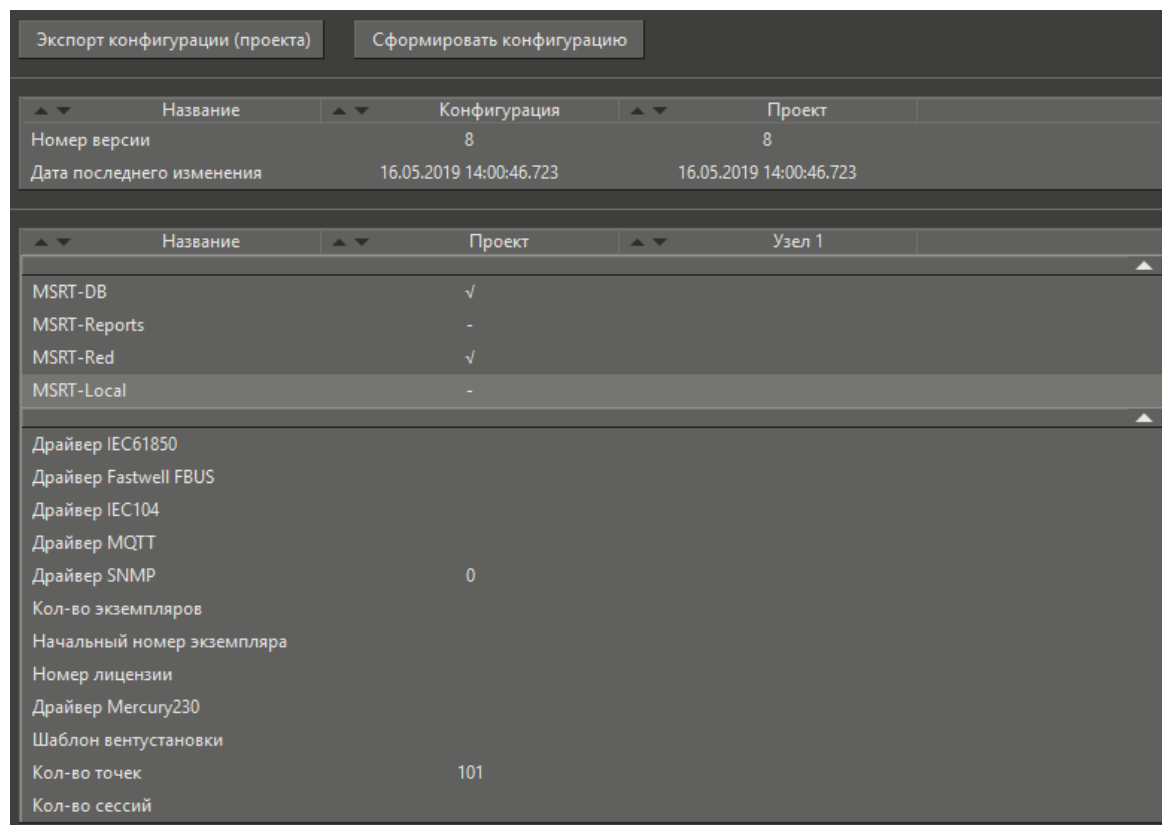
Название	Описание
Группа Опрос	Определяет, какой из двух резервируемых узлов опрашивается средой разработки, значения которого отображаются в интерфейсе среды разработки.
Кнопка Конфигурация	После нажатия кнопки происходит последовательная загрузка конфигурации проекта в оба узла по очереди. Сначала 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"
Кнопка Система	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".
Группа Узел 1 (Узел 2)	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

	<p>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. 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.</p>
<p>Поле Синхронизация времени</p>	<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>
<p>Поле Действия при загрузке конфигурации или системы</p>	<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>
<p>Нижняя часть панели управления</p>	<p>Если загрузка невозможна из-за отсутствия каких-либо модулей, не включенных в состав исполнительной системы, в данном поле появится соответствующее сообщение.</p>

### 9.1.12.2. Вкладка Конфигурация

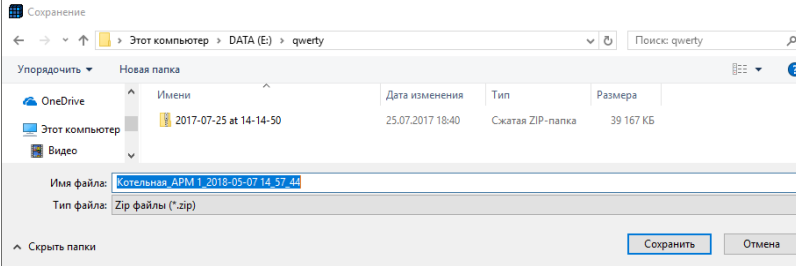
Вкладка панели узла Конфигурация служит для подготовки проекта для загрузки в среду исполнения, расположенную на другом устройстве, а также для выбора опций, необходимых для корректной работы проекта.

Вид вкладки:



Описание:

Элемент	Назначение
Экспорт конфигурации (проекта)	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>После нажатия на кнопку Сохранить, создается архив со следующим именем по умолчанию: [название проекта]_[имя узла]_[дата и время экспорта].</p>
Сформировать конфигурацию	<p>Кнопка позволяет сформировать конфигурацию (т.е. скомпилировать проект для работы в среде исполнения). Проект сохранится в папке <code>c:\Users\[имя пользователя]\AppData\Roaming\InSAT\MasterSCADA4DBeta\Debug_[имя проекта]\[имя узла]</code></p>
Таблица с информацией о проекте	<p>Служит для получения информации о загруженном проекте в среду исполнения, и о разрабатываемом проекте в среде разработки.</p> <p>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.</p>
Таблица с информацией об используемых опциях	<p>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.</p>

**Важно!** Информация об опциях и загруженном проекте в среду исполнения доступна только при подключении среды разработки к среде исполнения.

### 9.1.13. Работа с файлом данных для горячего рестарта

Как правило, данные для горячего рестарта хранятся в файле `session.bin`. Основное место расположения этого файла зависит от настроек проекта, а также от типа контроллера, на котором работает исполнительная система. Исключения 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 `mplc` с опцией `/export`.

После этого в рабочей папке исполнительной системы `/opt/mplc4` 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/mplc4/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/mplc4/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 `mplc` с опцией `/delhr`.

## 9.2. Работа с клиентом визуализации

В качестве клиента визуализации может выступать браузер, поддерживающий HTML5, а также специальное приложение для ОС Windows или Android, поставляемое компанией "ИнСАТ".



Важно! Браузеры Internet Explorer и Microsoft Edge не поддерживаются

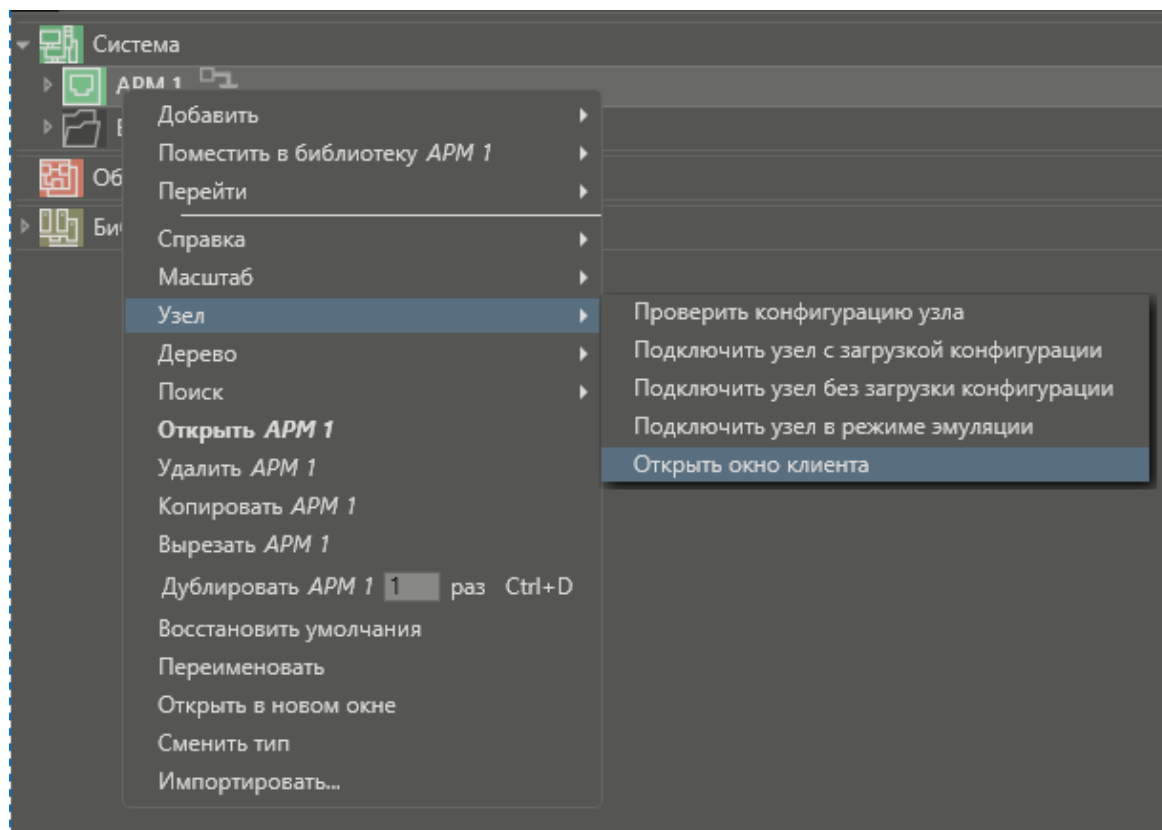
При подключении клиента к среде исполнения загрузится окно, назначенное стартовым при разработке проекта.

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).

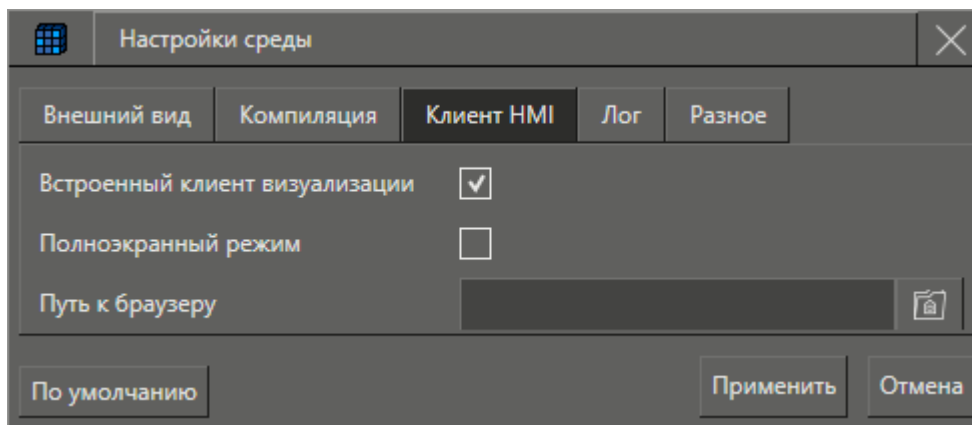
Важно! При подсчете клиентов, подключенных к среде исполнения, их происхождение не имеет значения: это могут быть как специальные приложения компании "ИнСАТ", так и браузеры сторонних производителей .

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 "ИнСАТ" – 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 НМІ v1 и v2:

- v1 - произойдет обновление страницы;
- v2 - переподключение 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 Среда исполнения, включенная в комплект Среда разработки, то файл находится по адресу: c:\Program Files (x86)\InSAT\MasterSCADA 4D 1.1\bin\Config\MasterPLC\WIN32\nginx\conf. В исполнительной системе для ОС Linux, по умолчанию, файл находится в папке: /opt/mplc4/nginx/conf.

### 9.2.1.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:

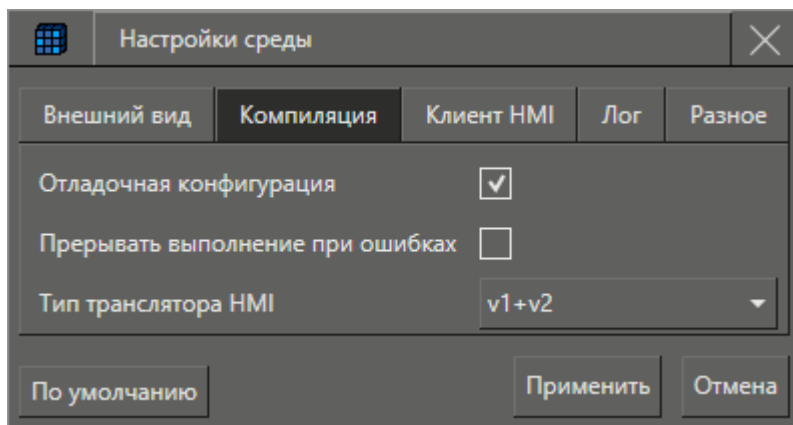
- В программе должны использоваться все возможные новые мировые тренды и технологии;
- Должен обеспечиваться максимально возможный функционал программы даже в самой узкоспециализированной ОС.

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.

Отличие версий описано в разделе Создание окон для графического интерфейса операторов.

## 9.2.2.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:




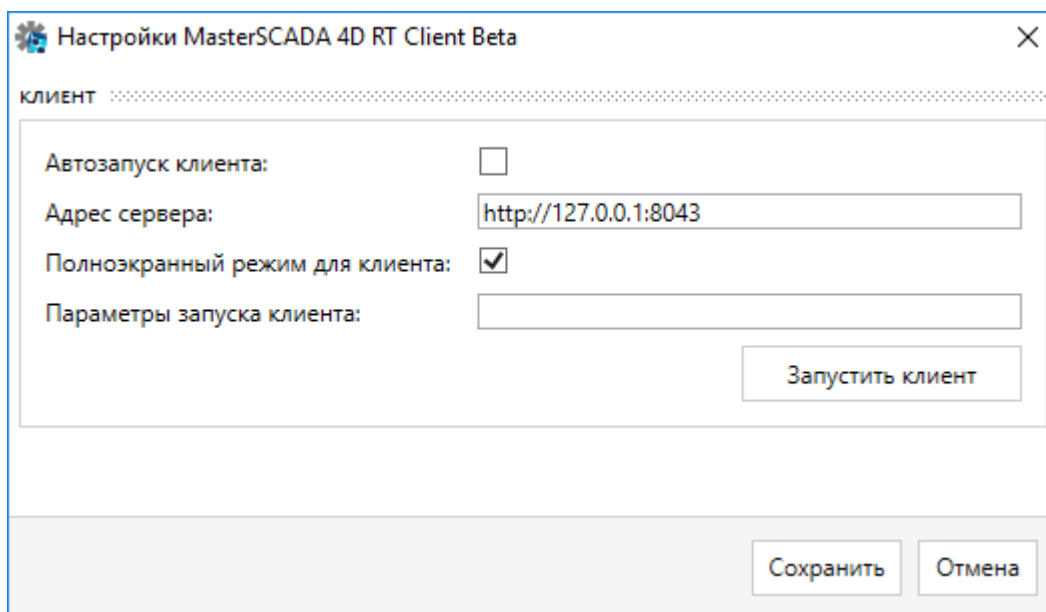
- Message Level – уровень конфликта (ошибка/предупреждение);
- Text – описание конфликта.

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.).

### 9.2.3. MasterSCADA 4D Client Monitor

Если MasterSCADA 4D Client установлен на компьютере не в составе исполнительной системы, а как независимое приложение, разработанное компанией "ИнСАТ", 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	<p>Задается путь к веб-серверу: <i>[IP]:[порт TCP/IP] / [номер экземпляра] /</i></p> <p>Если требуется обеспечить запуск клиента без процедуры авторизации, то следует ввести в адресную строку: 127.0.0.1:8043/index.html?user=sa&amp;password=123, 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</p>

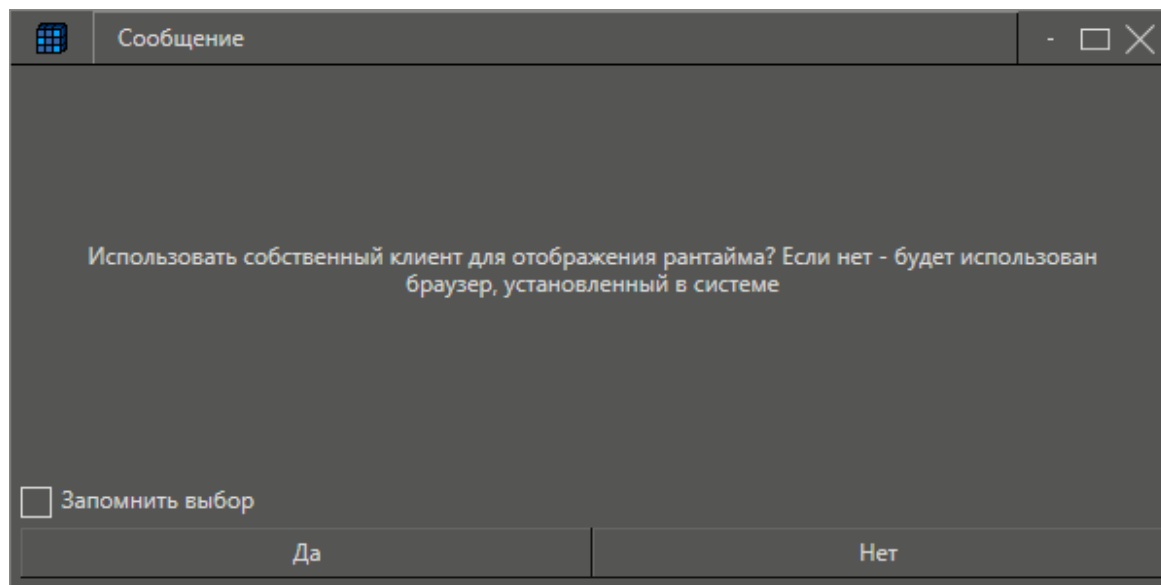
	откроется клиент визуализации выполненный в версии HMI v2.
Полноэкранный режим для клиента	Если этот флаг установлен, то MasterSCADA 4D Client запускается во весь экран.
Параметры запуска клиента	<p>Могут быть заданы следующие ключи:</p> <p>-d – если ключ установлен, то в процессе работы по кнопке F12 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 края 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 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>-ср -р "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>-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 на двух мониторах.</p>
Запустить клиент	По нажатию на кнопку откроется MasterSCADA 4D Client, разработанный компанией "ИнСАТ".
Сохранить	Сохраняет введенные настройки.
Отмена	Закрывает окно настроек, введенные изменения не сохраняются.

#### 9.2.4. Особенности первой автоматической загрузки проекта

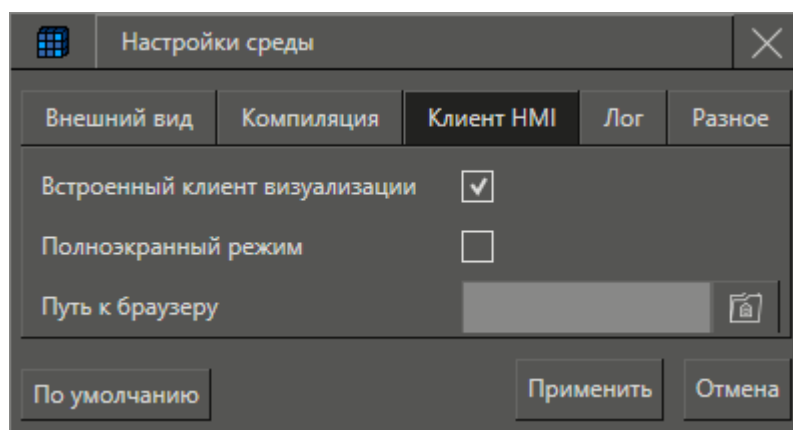
Если узел запускается в первый раз после установки продукта (т.е. до того как выполнено конфигурирование редактора проекта, и, соответственно, пока еще отсутствует файл %профиль пользователя%\AppData\Roaming\InSAT\MasterSCADA4D<версия>\MasterPLCUserSettings.xml), то открывается следующий диалог:





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:



### 9.2.5.Окно авторизации пользователей

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:

Логин

Пароль

**Вход**

Для того чтобы начать сессию, необходимо выбрать пользователя в списке **Логин** (этот список содержит пользователей, созданных в группе Безопасность. Пользователи), указать пароль выбранного пользователя, и нажать кнопку **Вход**. Если будет указан неверный пароль, то в диалоге отобразится соответствующее сообщение.

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 "ИнСАТ", требуется выполнить необходимые настройки в MasterSCADA 4D Monitor, или в MasterSCADA 4D Client Monitor (если используется клиент, установленный независимо от среды исполнения)

## 9.2.6. 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.

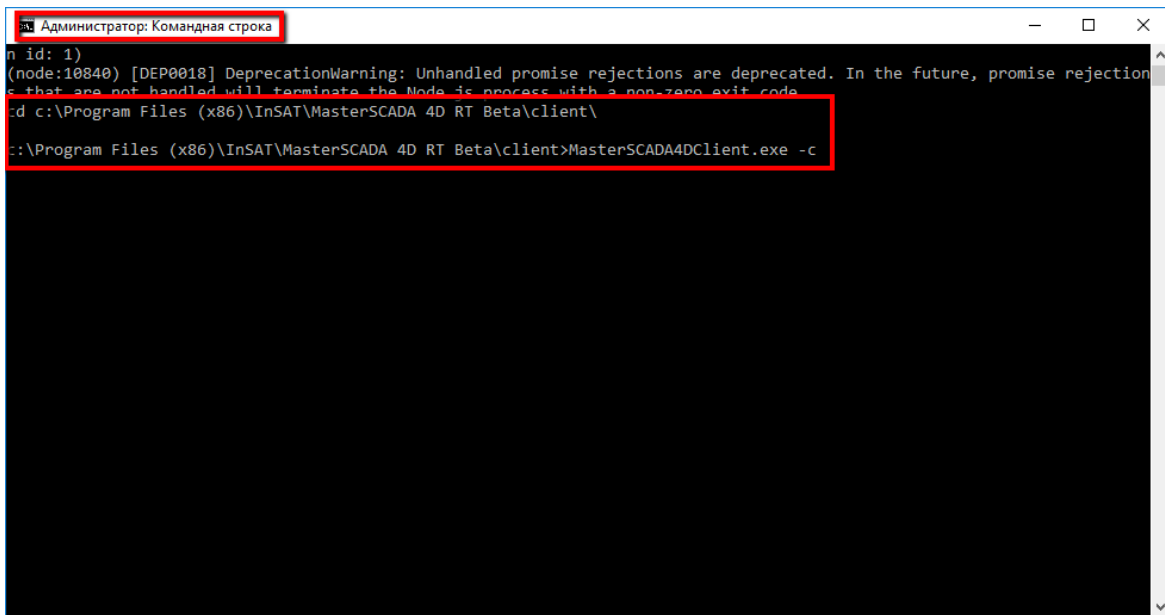
В поле Параметры запуска клиента необходимо указать следующие ключи:

- sr -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 в состав среды исполнения, то файл находится в папке: c:\Program Files (x86)\InSAT\MasterSCADA 4D RT[номер версии]\client\

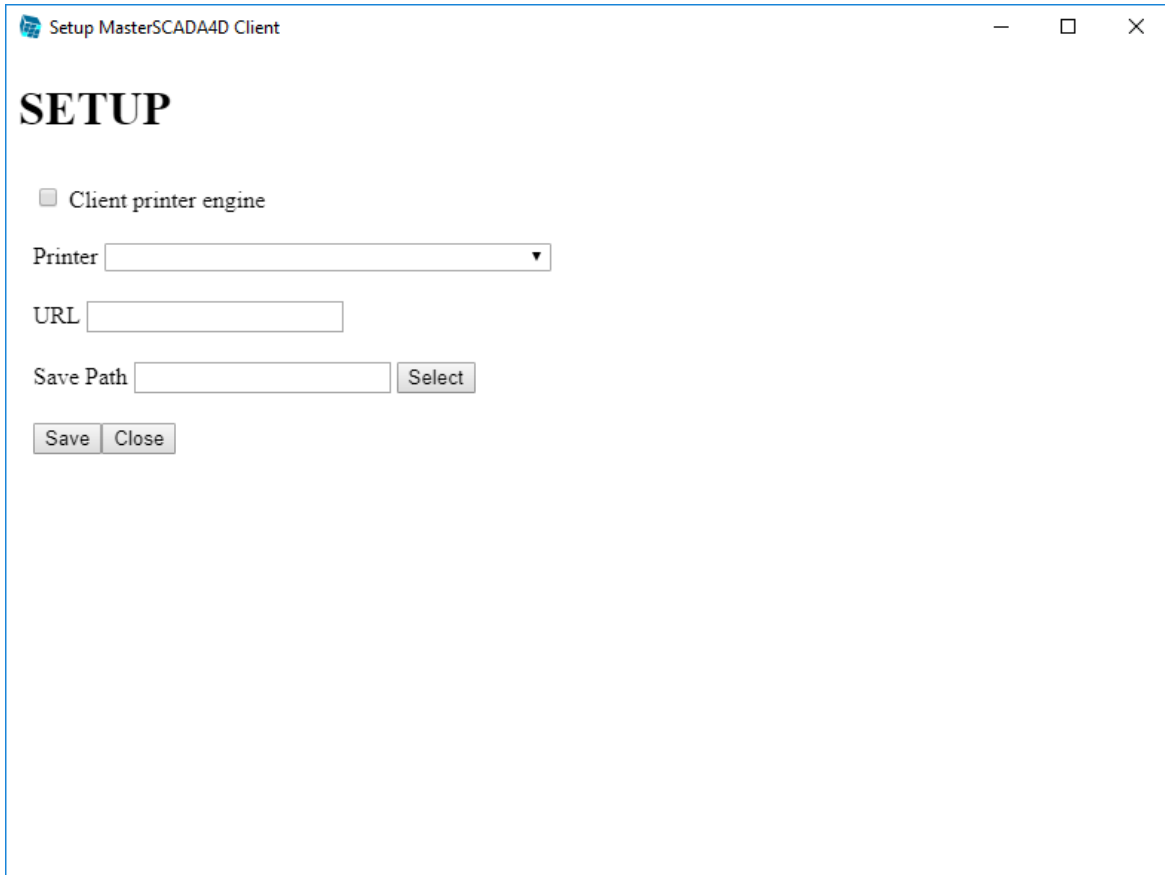
Для получения возможности редактирования этого файла необходимо запустить приложение с ключем -s. Например, можно воспользоваться стандартным приложением Windows Командная строка:



```
Администратор: Командная строка
n id: 1)
(node:10840) [DEP0018] DeprecationWarning: Unhandled promise rejections are deprecated. In the future, promise rejections
that are not handled will terminate the Node.js process with a non-zero exit code
c:\Program Files (x86)\InSAT\MasterSCADA 4D RT Beta\client\
:\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.

## 9.2.7. Горячие клавиши в клиенте визуализации

В клиенте визуализации MasterSCADA 4D Client доступны следующие горячие клавиши:

Клавиши	Назначение	Версия HMI
ALT+ курсор влево	Аналог стандартному действию в браузерах Назад. Открывает предыдущее окно.	

ALT+ курсор вправо	Аналог стандартному действию в браузерах Вперед. Открывает следующее окно.	
F11	Включает/отключает полноэкранный режим. Если изначально клиент был запущен в дан- ном режиме, то отключить его при помощи этой клавиши невозможно.	Все
F12	Открывает инструменты разработчика. Кла- виша активна только в том случае, если в па- раметрах запуска клиента установлен ключ - d. Ключ можно установить приложениях Mas- terSCADA 4D Monitor, MasterSCADA 4D Cli- ent Monitor (если клиент запускается незави- симо от среды исполнения)	Все

Использование горячих клавиш в других клиентах визуализации (браузерах) опреде-  
ляется их возможностями.

## 9.2.8.Многомониторный режим

Клиент визуализации MasterSCADA 4D Client поддерживает многомониторный ре-  
жим работы.

**Важно!** Функция поддерживается только в версии HMI 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 Mas-  
terSCADA 4D Client, но не больше, чем количество физически подключенных монито-  
ров. 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 ра-  
боты 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.

## 9.3. Отладка проекта

Необходимые средства и режимы отладки обеспечивают инструменты, которые содержатся в панели Устройства и вкладке инструментов Отладка.

### 9.3.1. Режимы отладки

В MasterSCADA 4D предусмотрены следующие виды отладки проекта:

- непрерывная отладка;
- отладка с шагом 1 цикл;

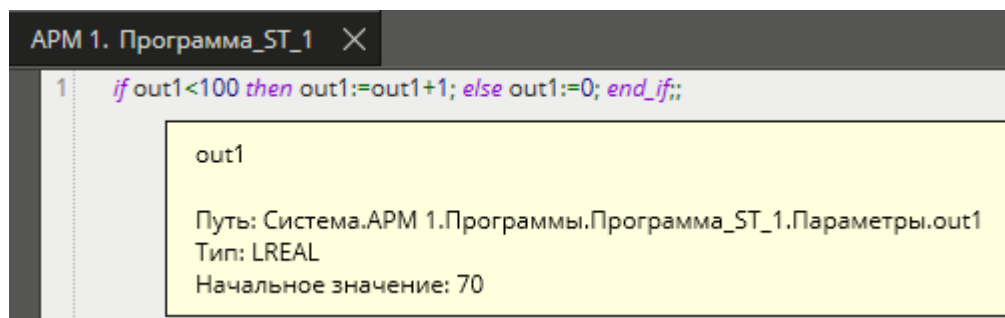
Для индикации/изменения значений параметров в режиме отладки в редакторе проекта используются:

- Дерево (Окно структуры проекта, Клеммники, Легенда и т.д.);
- Наблюдатель.

Чтобы выбрать режим отладки, нужно воспользоваться вкладкой инструментов Отладка.

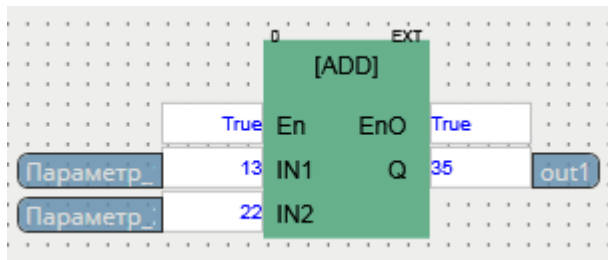
Значения параметров также индицируются и могут быть изменены в редакторах программ:

- в редакторе ST – во всплывающей подсказке параметра (только индикация):

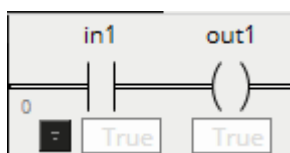


- в FBD- и LD-редакторе:

- 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):

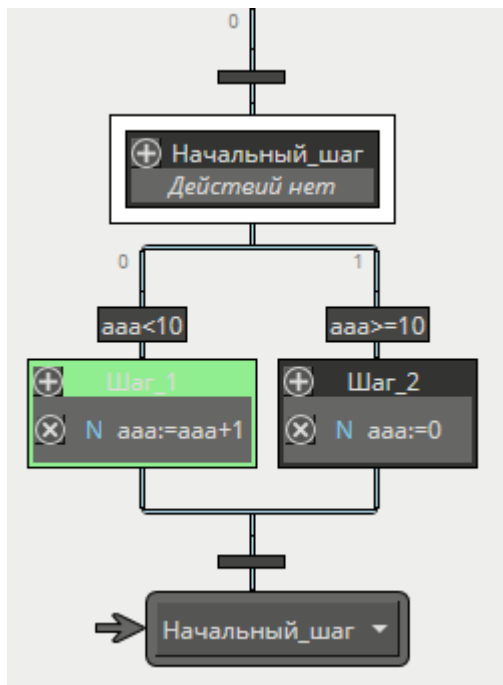


- значение связанного параметра LD-блока – в поле редактирования под блоком:




В режиме отладки в редакторах программ используется цветовая индикация конструкций языка:

- на SFC-диаграмме зеленым цветом подсвечиваются разрешенные переходы и активный шаг:



### 9.3.1.1. 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).

### 9.3.2.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.

### 9.3.3.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\InSAT\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:

MasterSCADA 4D RT Beta

ДЕЙСТВИЯ: [все] [все] [все] [Отчет об ошибках] [Включать дампы процессов] [i] [x] [⚙️] [?]

ПРОЦЕССЫ:

ID процесса	Процесс	Номер	Состояние	Имя проекта	Время запуска	Время остановки	Код возврата	Расшифровка кода возврата
	mplc	0	Остановлен	19_06_25 (APM 1)	19.45.34	15.42.35 за 3сек	100	Остановлено
	mplc	1	Остановлен	19_06_25 (APM 2)	19.45.34	15.42.39 за 4сек	100	Остановлено
	mplc	2	Остановлен		19.45.34	15.42.44 за 4сек	100	Остановлено
	nginx		Остановлен		19.45.34	15.42.44 за 0сек	-999	Принудительно остановлено
	node_ms4d		Остановлен		19.45.34	15.42.44 за 0сек	-999	Принудительно остановлено

Настройки MasterSCADA 4D RT Beta

ИСПОЛНИТЕЛЬНАЯ СИСТЕМА

Автозапуск исполнительной системы при запуске ОС:

Количество экземпляров mplc:

Рабочая папка:  ... ->

предпросмотр: C:\ProgramData\InSAT\MasterSCADA4DRTBeta

Параметры запуска mplc:

КЛИЕНТ

Автозапуск клиента при запуске ОС:

Адрес сервера:

Полноэкранный режим для клиента:

Параметры запуска клиента:

Запустить клиент

ВНЕШНИЙ ВИД

Сменить тему

Сохранить и перезапустить процессы | Сохранить | Отмена

Версия: 1.2.6.10014(MPLCD\_Beta\_20190527.5) Последняя синхронизация: 15.16.27

Далее необходимо открыть папку Server/Log, в которой может содержаться несколько вложенных папок, в зависимости от того, сколько экземпляров запускалось на данном компьютере:

Windows (C:) > ProgramData > InSAT > MasterSCADA4DRTBeta > Server > log

Имя	Дата изменения	Тип	Размер
0	12.07.2019 19:45	Папка с файлами	
1	12.07.2019 19:45	Папка с файлами	
2	12.07.2019 19:45	Папка с файлами	

**Важно!** Диагностическая информация о периодах работы среды исполнения выдается в конце файла.

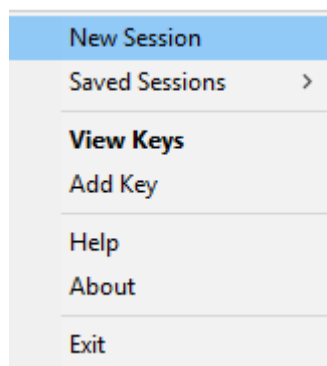
Любая среда исполнения

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..

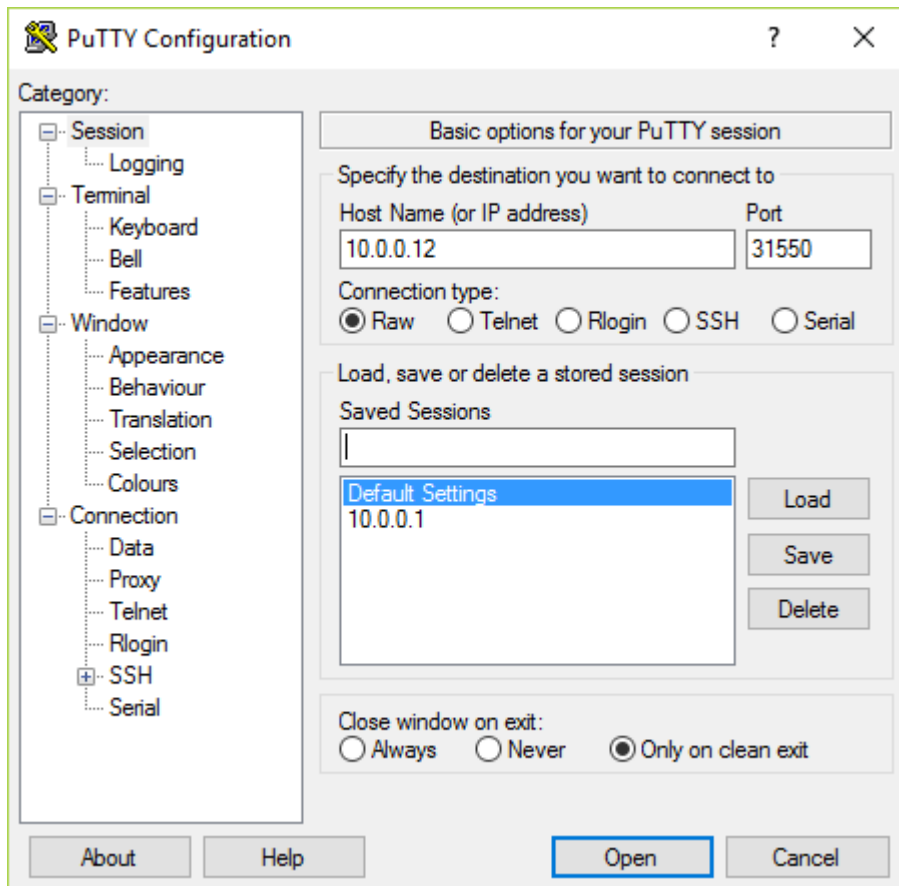
Рассмотрим запись лога с отладочной информацией при помощи свободно распространяемой программы PuTTY – <http://www.putty.org>

Запустить программу можно с помощью ярлыка  Pageant.

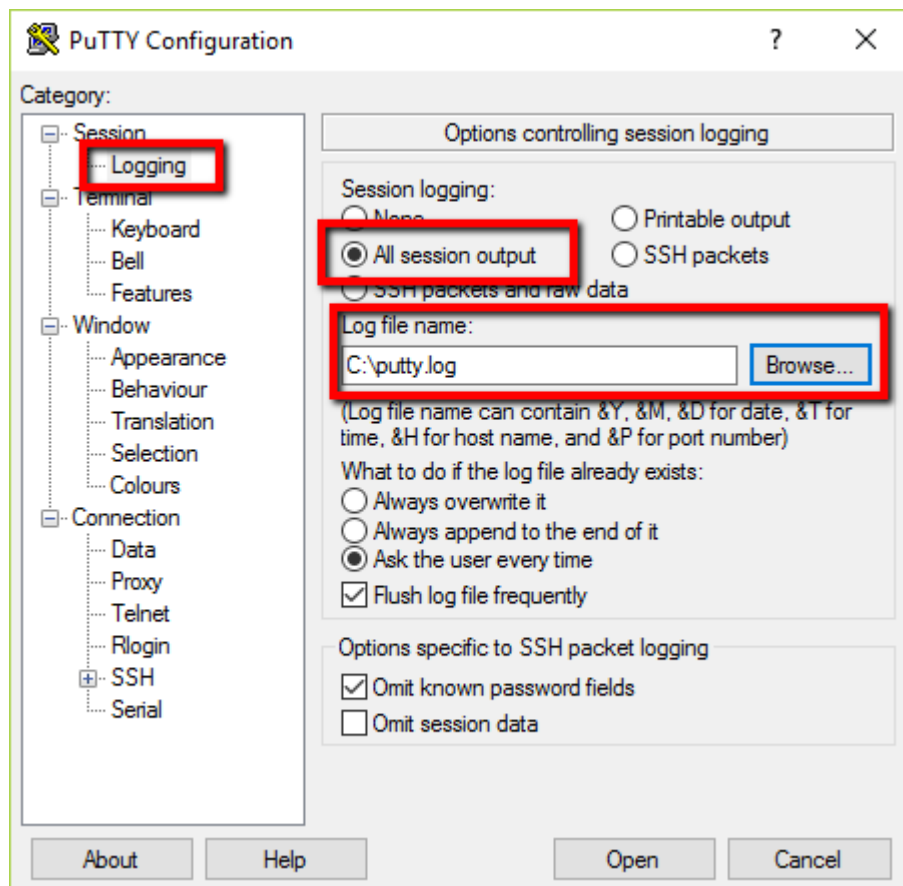
По умолчанию, программа минимизируется при запуске в панели задач. Для выполнения настройки программы необходимо нажать правой кнопкой мыши на её ярлык в панели задач, и в контекстном меню выбрать пункт New Session:



*При этом откроется окно настроек, в котором следует задать IP-адрес узла с установленной средой исполнения, порт и тип соединения.*

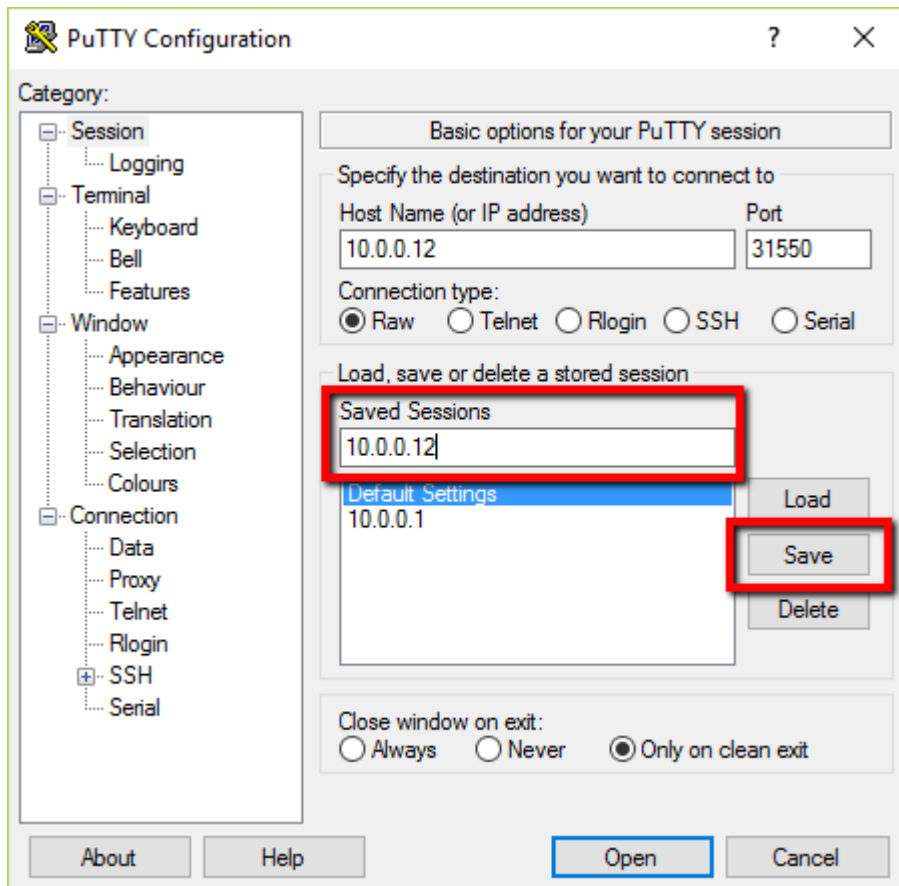


*После этого требуется определить место хранения лог-файлов:*



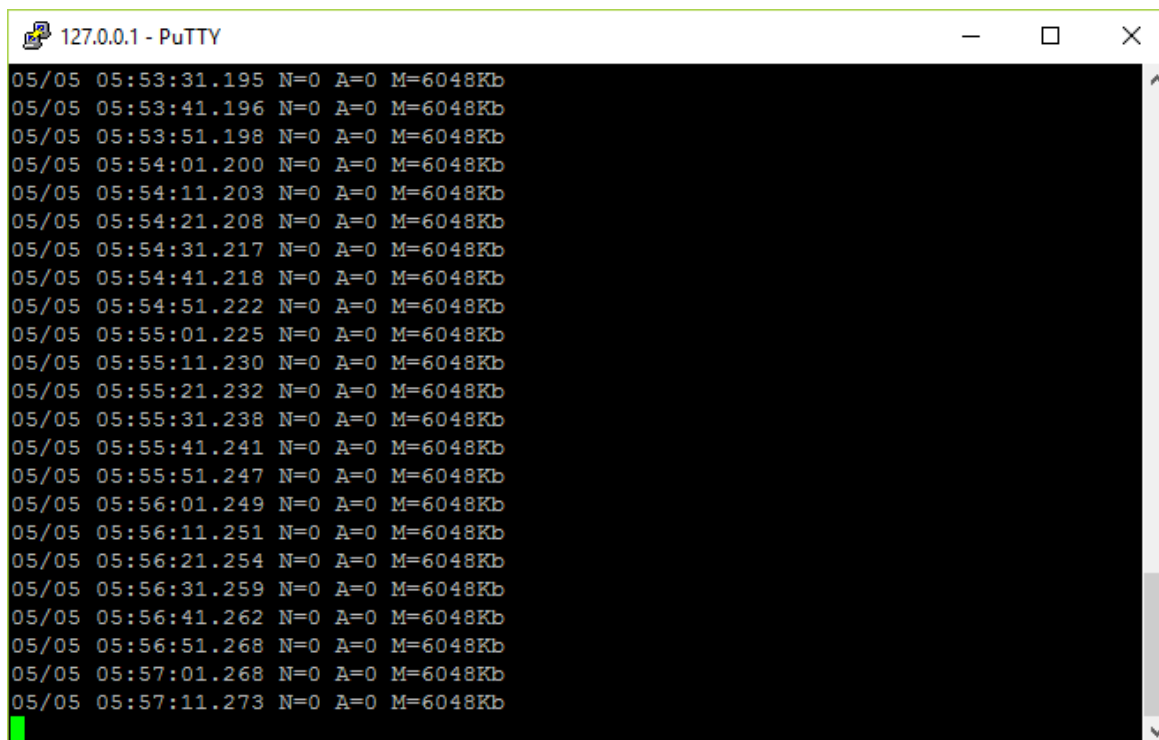
*Место, в которое планируется сохранить лог-файл, должно быть доступно для записи.*

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:



```
127.0.0.1 - PuTTY
05/05 05:53:31.195 N=0 A=0 M=6048Kb
05/05 05:53:41.196 N=0 A=0 M=6048Kb
05/05 05:53:51.198 N=0 A=0 M=6048Kb
05/05 05:54:01.200 N=0 A=0 M=6048Kb
05/05 05:54:11.203 N=0 A=0 M=6048Kb
05/05 05:54:21.208 N=0 A=0 M=6048Kb
05/05 05:54:31.217 N=0 A=0 M=6048Kb
05/05 05:54:41.218 N=0 A=0 M=6048Kb
05/05 05:54:51.222 N=0 A=0 M=6048Kb
05/05 05:55:01.225 N=0 A=0 M=6048Kb
05/05 05:55:11.230 N=0 A=0 M=6048Kb
05/05 05:55:21.232 N=0 A=0 M=6048Kb
05/05 05:55:31.238 N=0 A=0 M=6048Kb
05/05 05:55:41.241 N=0 A=0 M=6048Kb
05/05 05:55:51.247 N=0 A=0 M=6048Kb
05/05 05:56:01.249 N=0 A=0 M=6048Kb
05/05 05:56:11.251 N=0 A=0 M=6048Kb
05/05 05:56:21.254 N=0 A=0 M=6048Kb
05/05 05:56:31.259 N=0 A=0 M=6048Kb
05/05 05:56:41.262 N=0 A=0 M=6048Kb
05/05 05:56:51.268 N=0 A=0 M=6048Kb
05/05 05:57:01.268 N=0 A=0 M=6048Kb
05/05 05:57:11.273 N=0 A=0 M=6048Kb
```

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@insat.ru](mailto:support.ms4d@insat.ru) для анализа.

Пример отладочных сообщений

Периоды выполнения задач

$N=3725$   $A=0$   $U0=28311(4,4,13)(0,0,6)$   $S0=4038(30,e0)$   $HR=1(0)$   $M=6784Kb$ , где:

$U0$  – статистика задачи объектов

$28311(4,4,13)(0,0,6)$  – кол-во циклов (среднее,мин,макс цикла)(среднее,мин,макс реального выполнения)

$S0$  – статистика задачи системы кол-во циклов (среднее, e кол-во ошибок)

$M$  – используемый объем памяти.

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 Task=0 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 00=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) D(Tm:float, Ct:int, Av:int)
```

Где:

R - Чтение

W - Запись

D - Удаление

Tm - Число секунд, затраченных на обработку R/W/D

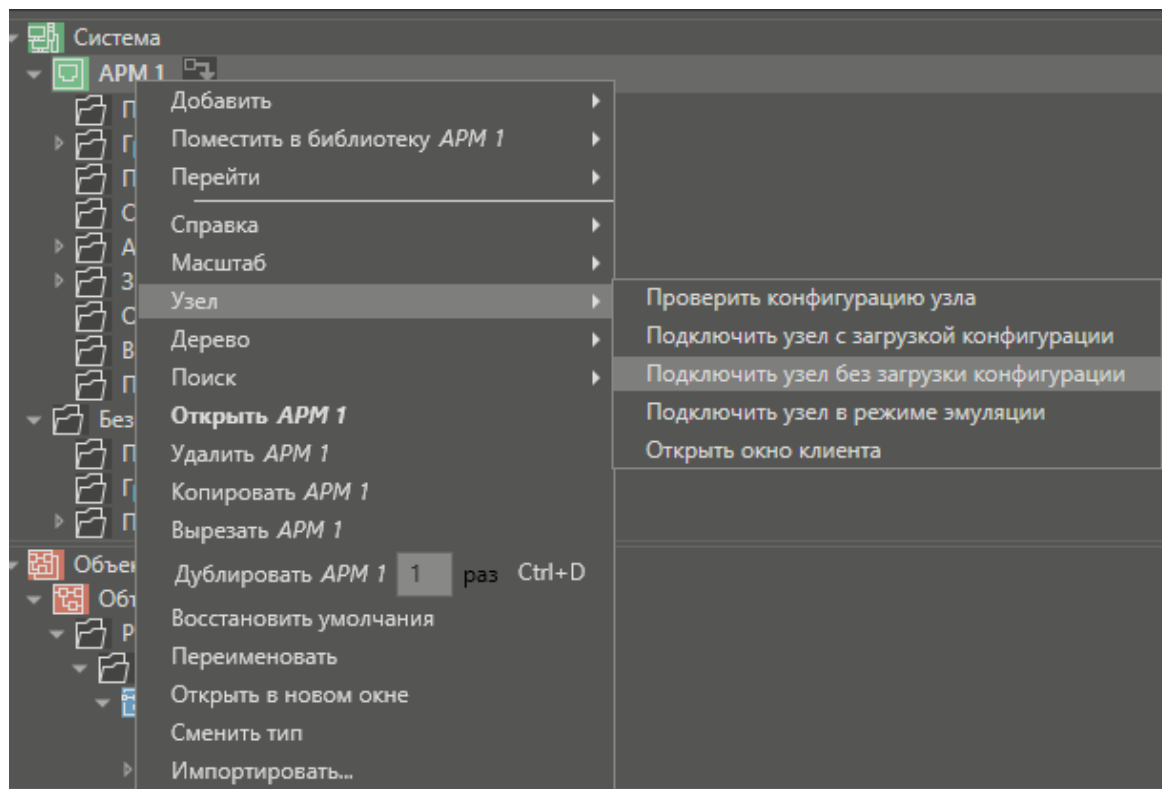
Ct - Количество записей обработанных за последнюю минуту R/W/D

Av - Скорость обработки записей в секунду R/W/D

### 9.3.4. 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:



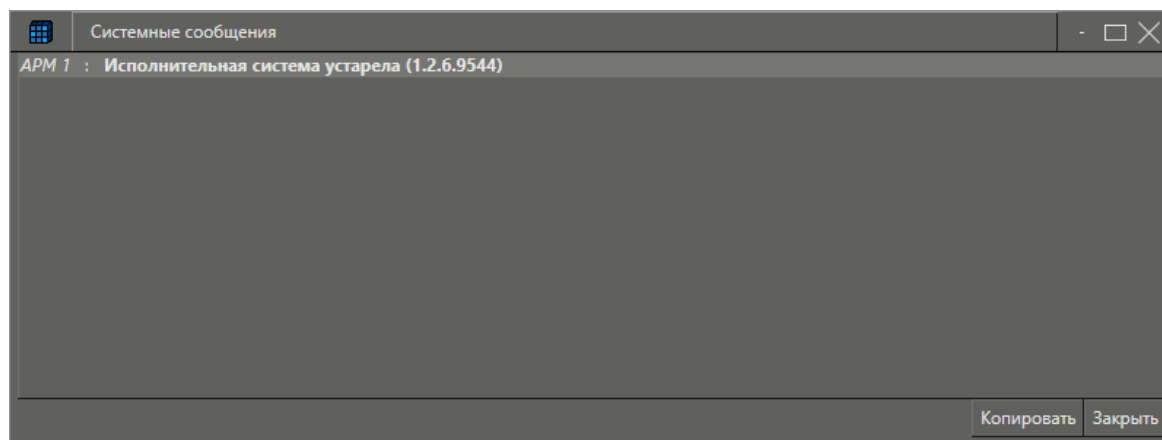


В результате этого, в интерфейсе среды разработки отобразятся текущие значения параметров, которые допускается принудительно изменить.

Отключение среды разработки от исполняемого проекта, выполненное при помощи контекстного меню или вкладки Исполнение, не приводит к остановке исполнения проекта.

#### 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 определенных 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 [scada@insat.ru](mailto:scada@insat.ru).

Среда исполнения для различных контроллеров и операционных систем поставляется в комплекте со средой разработки и находится в папке `c:\Program Files (x86)\InSAT\MasterSCADA 4D 1.2\bin\Config\MasterPLC\PLC`.

Инсталлятор демоверсии среды исполнения для Windows можно скачать с сайта компании "ИнСАТ". Коммерческую версию для ОС Windows необходимо заказать по электронному адресу [scada@insat.ru](mailto:scada@insat.ru).

### 9.3.5. 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. 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.

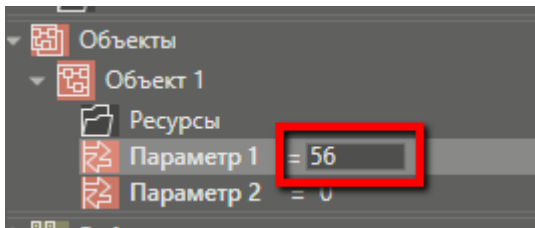
Если проект содержит окна, то будет открыто несколько окон клиентов по числу узлов.

**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.


### 9.3.6. Имитация значений параметров

Установка произвольного значения параметра

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.




### Блокировка параметра

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

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.

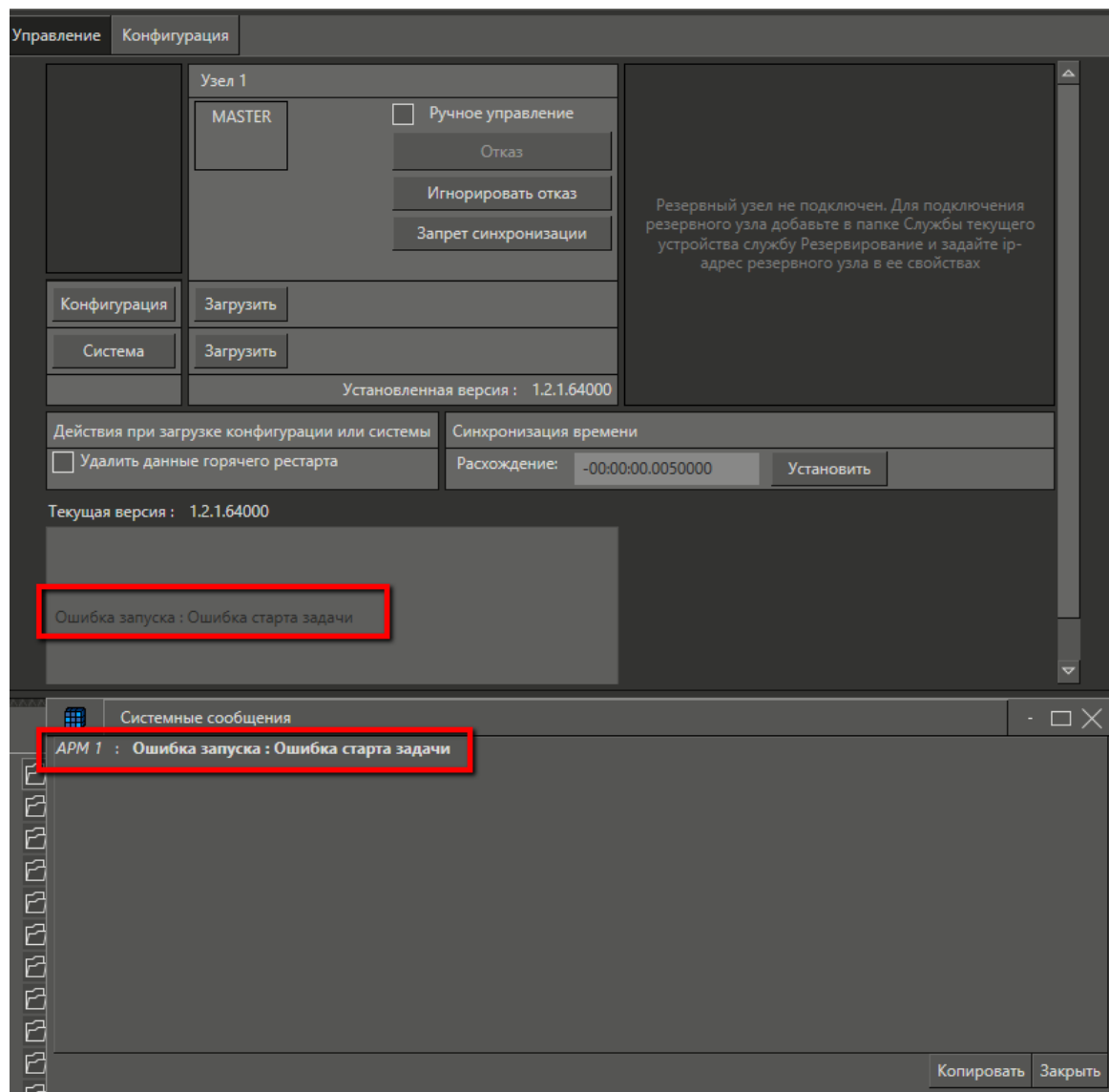
### Автоматическая имитация значений каналов

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.

## 9.3.7. Сообщения об ошибках

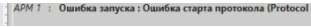
В среде разработки, подключенной к среде исполнения могут появиться сообщения, сигнализирующие о возникновении каких-либо проблем в работе.

Сообщения отображаются в окне системных сообщений, которое открывается автоматически, а также в панели узла, во вкладке Управление.



Сообщение	Причина	Комментарии
Превышено разрешенное число экземпляров	На одном устройстве запущено большее, чем позволяет лицензия, число экземпляров среды исполнения.	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 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.
Не загружена конфигурация	Среда разработки подключена к среде исполнения, в которую не загружена конфигурация (проект)	Выполните загрузку конфигурации доступным способом: <ul style="list-style-type: none"> <li>• Автоматическая загрузка проекта</li> <li>• Ручная загрузка проекта в среду исполнения</li> </ul>
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.
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	

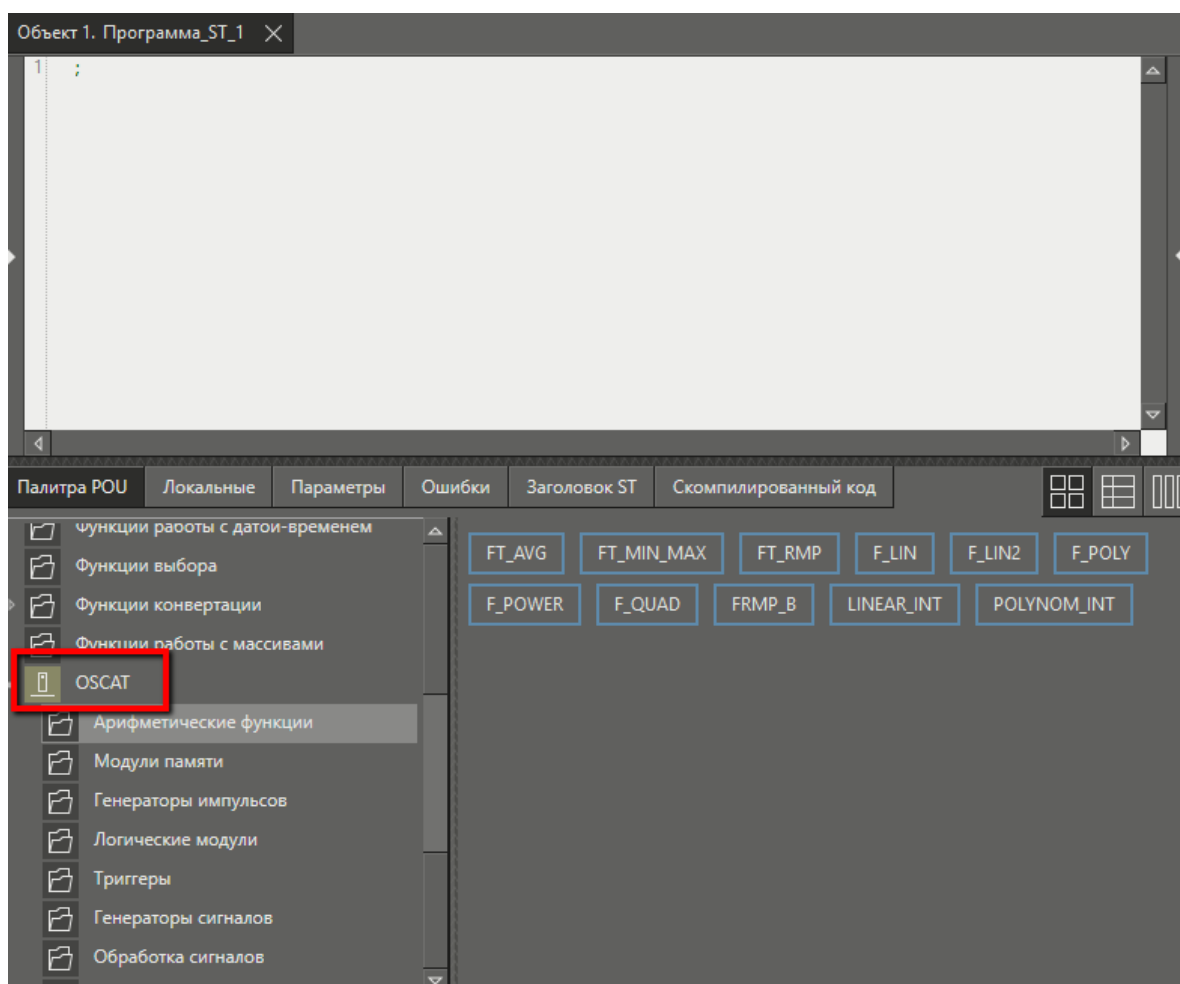
<p>Ошибка старта задачи</p>	<p>Возникает, если в проекте используется ФБ, неподдерживаемый в данной платформе, а также в случае использования отчетов в исполнительных системах, предназначенных для работы на контроллерах.</p> <p>В тексте ошибки выдается имя элемента проекта, который стал причиной проблемы.</p>	<p>Удалите из проекта проблемный элемент. В случае возникновения сложностей при выборе альтернативного решения обратитесь в службу технической поддержки по электронной почте <a href="mailto:support.ms4d@insat.ru">support.ms4d@insat.ru</a></p>
<p>Ошибка старта протокола</p>	<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>	<p>Удалите протокол из проекта.</p>
<p>Превышено ограничение для [название протокола/драйвера]: Доступное - X, Требуемое - Y</p>	<p>В проекте задействовано больше каналов, чем позволяет используемая лицензия (<math>X &lt; Y</math>). При этом загрузки конфигурации и подключения к узлу происходить не будет.</p>	<p>Задействованным каналом считается тот, который имеет связи или архивируется. Если все каналы необходимы для функционирования проекта, то необходимо увеличить количество точек данного протокола.</p>

## 10. Библиотека OSCAT

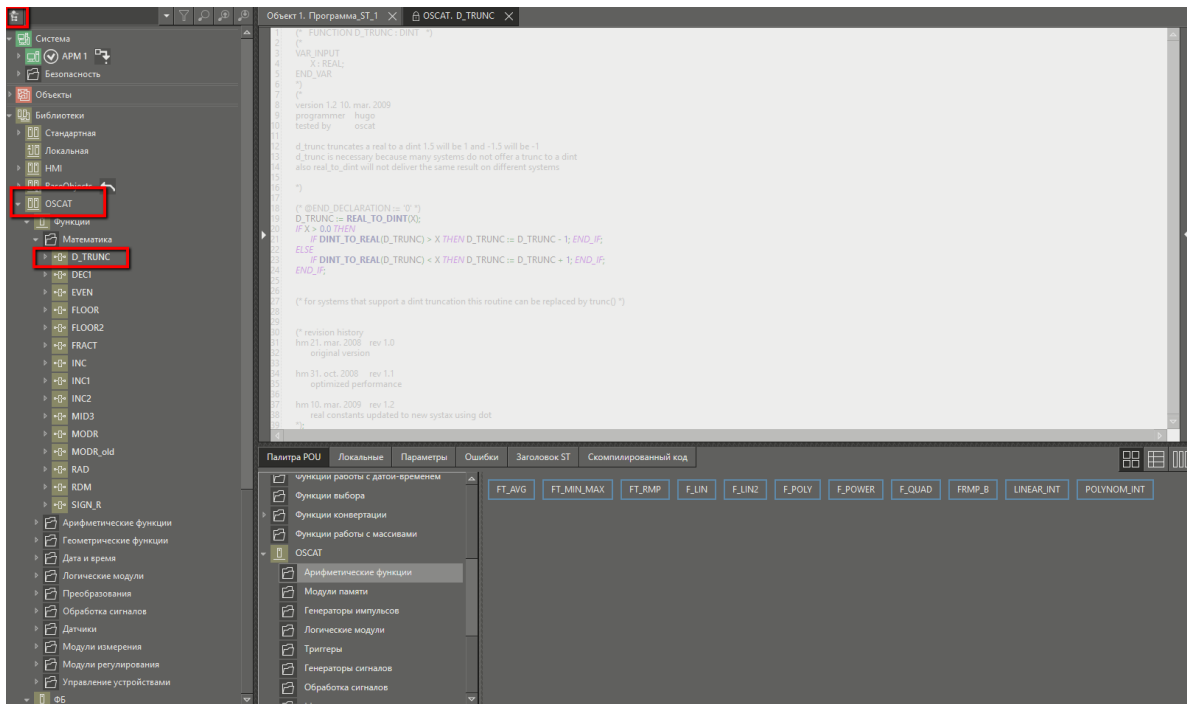
В состав MasterSCADA 4D включена библиотека компании OSCAT (Open Source Community for Automation Technology; библиотека и ее описание доступны для свободной загрузки на сайте <http://www.oscat.de> ). Библиотека содержит функции с открытым исходным кодом.

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).



## 10.1. OSCAT.Функции

OSCAT.Функции.Математика

D\_TRUNC (OSCAT)

DEC1 (OSCAT)

EVEN (OSCAT)

FLOOR (OSCAT)

FLOOR2 (OSCAT)

FRACT (OSCAT)

INC (OSCAT)

INC1 (OSCAT)

INC2 (OSCAT)

MID3 (OSCAT)

MODR (OSCAT)

MODR\_old (OSCAT)

RAD (OSCAT)

RDM (OSCAT)

SIGN\_R (OSCAT)



## OSCAT.Функции.Арифметические функции

F\_LIN (OSCAT)

F\_LIN2 (OSCAT)

F\_POLY (OSCAT)

F\_POWER (OSCAT)

F\_QUAD (OSCAT)

FRMP\_B (OSCAT)

LINEAR\_INT (OSCAT)

POLYNOM\_INT (OSCAT)

## OSCAT.Функции.Геометрические функции

CIRCLE\_A (OSCAT)

CIRCLE\_C (OSCAT)

CIRCLE\_SEG (OSCAT)

CONE\_V (OSCAT)

ELLIPSE\_A (OSCAT)

ELLIPSE\_C (OSCAT)

SPHERE\_V (OSCAT)

TRIANGLE\_A (OSCAT)

## OSCAT.Функции.Дата и время

DAY\_OF\_WEEK (OSCAT)

DAY\_OF\_YEAR (OSCAT)

LEAP\_OF\_DATE (OSCAT)

MONTH\_OF\_DATE (OSCAT)

YEAR\_OF\_DATE (OSCAT)

## OSCAT.Функции.Логические модули

BCDC\_TO\_INT (OSCAT)

BIT\_COUNT (OSCAT)

BIT\_LOAD\_B (OSCAT)

BIT\_LOAD\_B2 (OSCAT)  
BIT\_LOAD\_DW (OSCAT)  
BIT\_LOAD\_DW2 (OSCAT)  
BIT\_LOAD\_W (OSCAT)  
BIT\_LOAD\_W2 (OSCAT)  
BIT\_OF\_DWORD (OSCAT)  
BIT\_TOGGLE\_B (OSCAT)  
BIT\_TOGGLE\_DW (OSCAT)  
BIT\_TOGGLE\_W (OSCAT)  
BYTE\_OF\_BIT (OSCAT)  
BYTE\_OF\_DWORD (OSCAT)  
BYTE\_TO\_GRAY (OSCAT)  
CHECK\_PARITY (OSCAT)  
DWORD\_OF\_BYTE (OSCAT)  
DWORD\_OF\_WORD (OSCAT)  
GRAY\_TO\_BYTE (OSCAT)  
INT\_TO\_BCDC (OSCAT)  
MUX\_2 (OSCAT)  
MUX\_4 (OSCAT)  
PARITY (OSCAT)  
REFLECT (OSCAT)  
REVERSE (OSCAT)  
SHL1 (OSCAT)  
SHR1 (OSCAT)  
SWAP\_BYTE (OSCAT)  
SWAP\_BYTE2 (OSCAT)  
WORD\_OF\_BYTE (OSCAT)  
WORD\_OF\_DWORD (OSCAT)

OSCAT.Функции.Преобразования

BFT\_TO\_MS (OSCAT)

C\_TO\_F (OSCAT)  
C\_TO\_K (OSCAT)  
F\_TO\_C (OSCAT)  
F\_TO\_OM (OSCAT)  
F\_TO\_PT (OSCAT)  
GEO\_TO\_DEG (OSCAT)  
K\_TO\_C (OSCAT)  
KMH\_TO\_MS (OSCAT)  
MS\_TO\_BFT (OSCAT)  
MS\_TO\_KMH (OSCAT)  
OM\_TO\_F (OSCAT)  
PT\_TO\_F (OSCAT)

OSCAT.Функции.Обработка сигналов

AIN (OSCAT)  
AOUT (OSCAT)  
AOUT1 (OSCAT)  
BYTE\_TO\_RANGE (OSCAT)  
MIX (OSCAT)  
MUX\_R2 (OSCAT)  
MUX\_R4 (OSCAT)  
OFFSET (OSCAT)  
OFFSET2 (OSCAT)  
OVERRIDE (OSCAT)  
RANGE\_TO\_BYTE (OSCAT)  
RANGE\_TO\_WORD (OSCAT)  
SCALE (OSCAT)  
SCALE\_B (OSCAT)  
SCALE\_B2 (OSCAT)  
SCALE\_B4 (OSCAT)  
SCALE\_B8 (OSCAT)

SCALE\_D (OSCAT)  
SCALE\_R (OSCAT)  
SCALE\_X2 (OSCAT)  
SCALE\_X4 (OSCAT)  
SCALE\_X8 (OSCAT)  
STAIR (OSCAT)  
WORD\_TO\_RANGE (OSCAT)

OSCAT.Функции.Датчики

MULTI\_IN (OSCAT)  
RES\_NI (OSCAT)  
RES\_NTC (OSCAT)  
RES\_PT (OSCAT)  
RES\_SI (OSCAT)  
SENSOR\_INT (OSCAT)  
TEMP\_NI (OSCAT)  
TEMP\_NTC (OSCAT)  
TEMP\_PT (OSCAT)  
TEMP\_SI (OSCAT)

OSCAT.Функции.Модули измерения

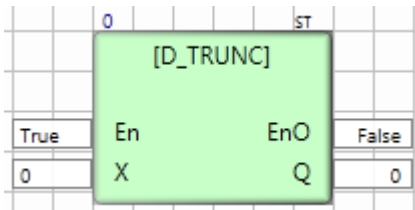
T\_PLC\_MS (OSCAT)  
T\_PLC\_US (OSCAT)

OSCAT.Функции.Модули регулирования

BAND\_B (OSCAT)  
CTRL\_IN (OSCAT)  
DEAD\_BAND (OSCAT)  
DEAD\_ZONE (OSCAT)

OSCAT.Функции.Управление устройствами

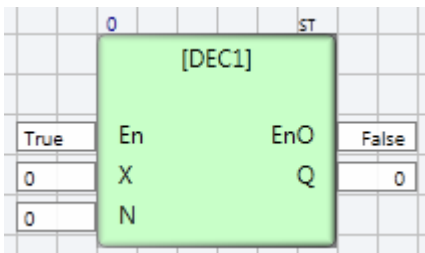
## MANUAL (OSCAT)

**10.1.1. OSCAT.Функции.Математика****10.1.1.1. D\_TRUNC (OSCAT)**

Тип данных X – REAL, Q – DINT.

Аналог TRUNC, целая часть к нулю.

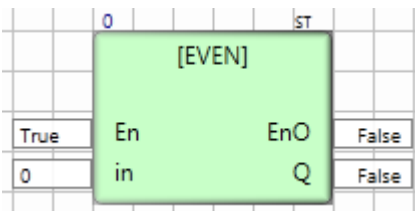
Например,  $d\_trunc(5.9)=d\_trunc(5.1)=5$ ;  $d\_trunc(-5.9)=d\_trunc(-5.1)=-5$ .

**10.1.1.2. DEC1 (OSCAT)**

Тип данных X, N и Q – INT.

Если  $X=0$ , то  $Q=N-1$ , иначе  $Q=X-1$ .

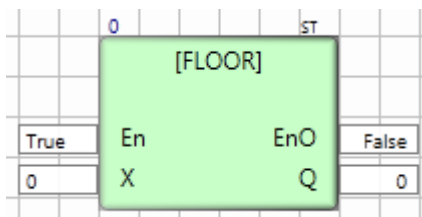
При связи  $Q \rightarrow X$  и  $N > 0$  функция работает как генератор убывающего пилообразного сигнала ( $N-1, N-2 \dots 0, N-1 \dots$ ).

**10.1.1.3. EVEN (OSCAT)**

Тип данных IN – DINT, Q – BOOL.

Если IN – четное (в т.ч. 0),  $Q=TRUE$ , иначе  $Q=FALSE$ .

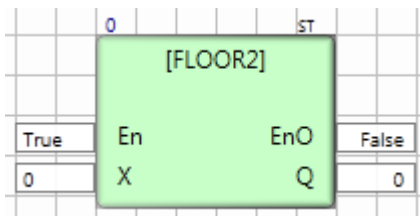
### 10.1.1.4. FLOOR (OSCAT)



Тип данных X – REAL, Q – INT.

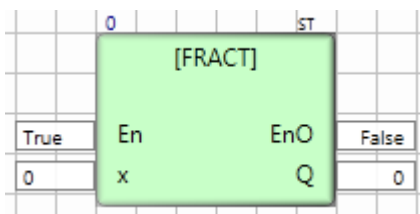
Функция возвращает наибольшее целое число, меньшее или равное X:  
 $\text{floor}(5.1)=\text{floor}(5.9)=5$ ,  $\text{floor}(-5.1)=\text{floor}(-5.9)=-6$ .

### 10.1.1.5. FLOOR2 (OSCAT)



Аналог FLOOR (OSCAT) , но тип данных Q – DINT.

### 10.1.1.6. FRACT (OSCAT)

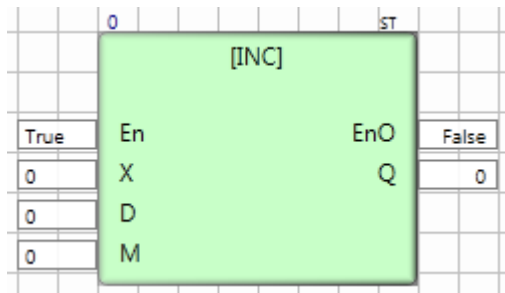


Тип данных X – REAL, Q – REAL.

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).

### 10.1.1.7. INC (OSCAT)



Тип данных X, D, M и Q – INT.

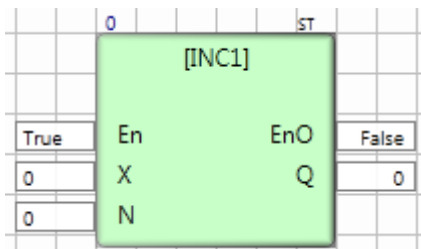
Функция используется при обращении к элементам массива, она увеличивает X на D и предотвращает выход индекса (X+D) за M:

- 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:  $INC(3, 2, 5) = 5$ ,  $INC(4, 2, 5) = 0$ .

При любых неотрицательных X, D и M значение Q лежит в диапазоне [0, M].

### 10.1.1.8. INC1 (OSCAT)

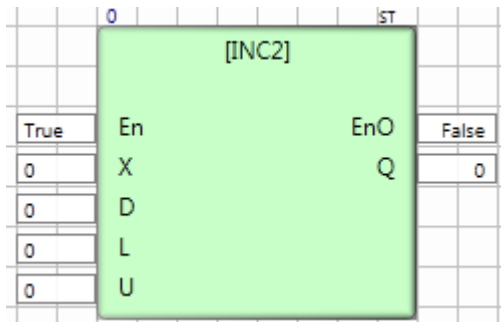


Тип данных X, N и Q – INT.

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)).

### 10.1.1.9. INC2 (OSCAT)



Тип данных X, D, L, U и Q – INT.

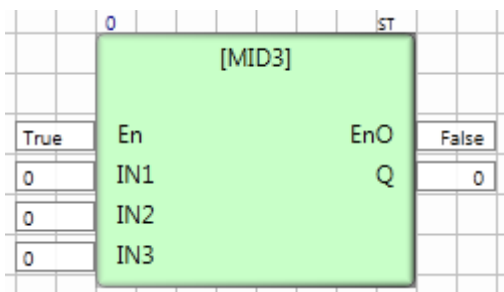
Функция используется при обращении к элементам массива, она увеличивает X на D и предотвращает выход индекса (X+D) за U:

- 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].

### 10.1.1.10. MID3 (OSCAT)

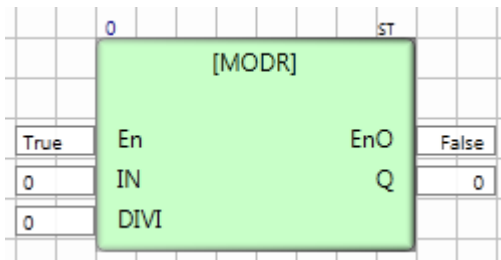


Тип данных IN1, IN2, IN3 и Q – REAL.

Функция возвращает значение входа, расположенное на числовой оси между двумя другими значениями входов. Например,  $mid(-3, 9, 2) = 2$ .

### 10.1.1.11. MODR (OSCAT)





Тип данных IN, DIVI и Q – REAL.

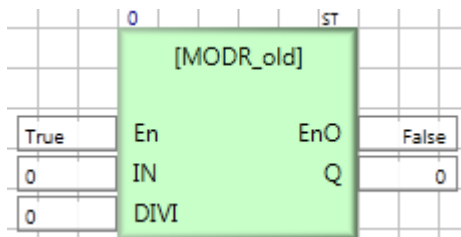
Функция работает по алгоритму, который используется в ФБ-генераторах для сдвига генерируемого сигнала по фазе (см., например, GEN\_SIN (OSCAT)):

$$Q := IN - DIVI * FLOOR2(IN/DIVI)$$

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).

#### 10.1.1.12. MODR\_old (OSCAT)



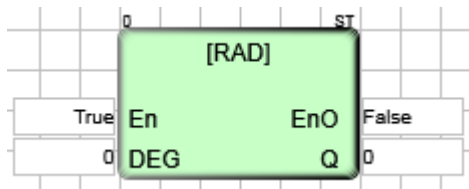
Тип данных IN, DIVI и Q – REAL.

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).

#### 10.1.1.13. RAD (OSCAT)



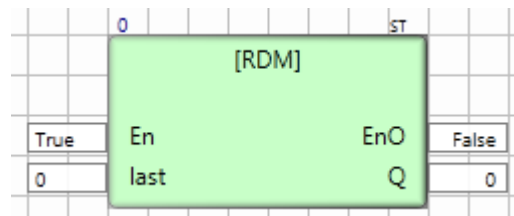
Тип данных входа DEG и выхода Q – REAL.

Функция переводит угловые градусы ( $00 \leq \text{DEG} < 3600$ ) в радианы (Q).

Если  $\text{DEG} \geq 3600$ , перед конвертацией из DEG необходимое число раз вычитается 3600 – до тех пор, пока конвертируемое значение не окажется в диапазоне  $[00, 3600)$ . Таким образом,  $\text{RAD}(3600) = \text{RAD}(00) = 0$ ,  $\text{RAD}(9000) = \text{RAD}(1800) = \pi$ .

Используемые функции: MODR (OSCAT).

#### 10.1.1.14. RDM (OSCAT)



Тип данных LAST и Q – REAL.

Функция возвращает псевдослучайное число в интервале (0, 1).

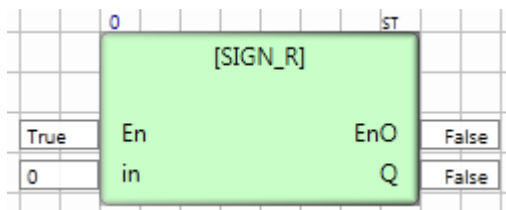
Если в программе требуется иметь несколько независимых генераторов псевдослучайных чисел, функцию RDM нужно вызывать с различными значениями входа LAST ( $0 \leq \text{LAST} < 1$ ):

```
r116:=RDM(last:=0);
```

```
r117:=RDM(last:=0.5);
```

Используемые функции: FRACT (OSCAT), BIT\_COUNT (OSCAT), T\_PLC\_MS (OSCAT).

#### 10.1.1.15. SIGN\_R (OSCAT)

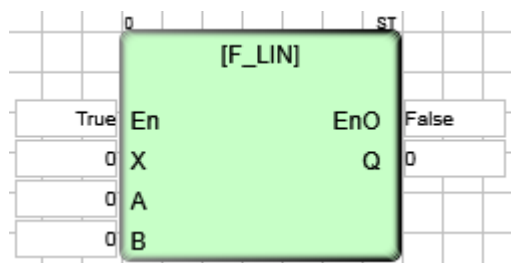


Тип данных IN – REAL, Q – BOOL.

Если  $IN < 0$ ,  $Q = TRUE$ , иначе  $Q = FALSE$ .

## 10.1.2. OSCAT.Функции.Арифметические функции

### 10.1.2.1. F\_LIN (OSCAT)

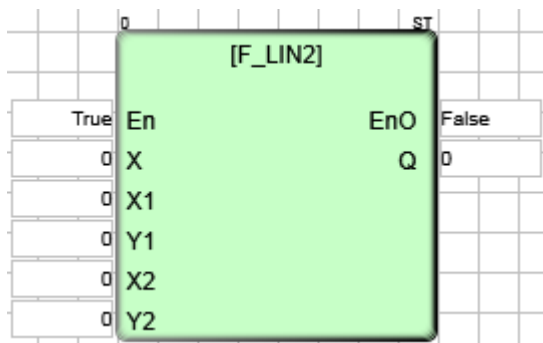


Тип данных входов X, A и B и выхода Q – REAL.

Функция F\_LIN выполняет стандартное линейное преобразование:

$$Q := A * X + B$$

### 10.1.2.2. F\_LIN2 (OSCAT)

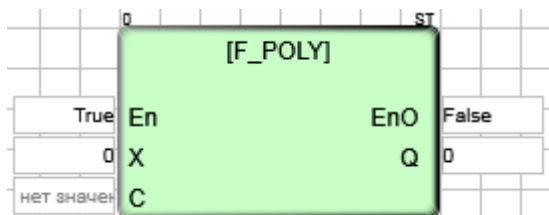


Тип данных входов и выхода – REAL.

Функция возвращает ординату точки прямой по абсциссе (X); прямая определяется двумя точками – (X1, Y1) и (X2, Y2):

$$F\_LIN2 := (Y2 - Y1) / (X2 - X1) * (X - X1) + Y1$$

### 10.1.2.3. F\_POLY (OSCAT)



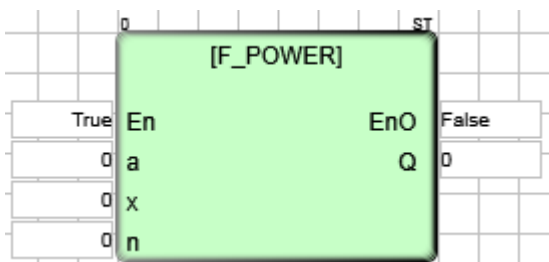
Тип данных входа X и выхода Q – REAL.

Вход C – массив 8 переменных REAL (индексы – 0 .. 7).

Функция возвращает значение полинома 7-й степени от X:

$$Q := C[7] \cdot x^7 + C[6] \cdot x^6 + C[5] \cdot x^5 + C[4] \cdot x^4 + C[3] \cdot x^3 + C[2] \cdot x^2 + C[1] \cdot x + C[0]$$

### 10.1.2.4. F\_POWER (OSCAT)

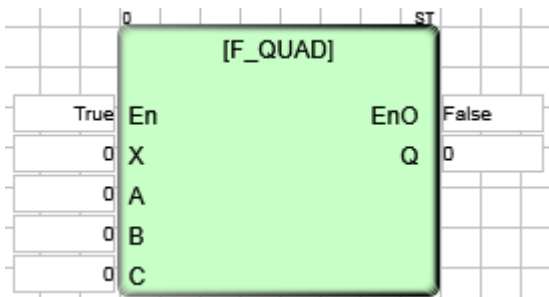


Тип данных входов и выхода – REAL.

$$Q := A \cdot X^N$$

Используемые функции: EXPT, возведение в степень.

### 10.1.2.5. F\_QUAD (OSCAT)

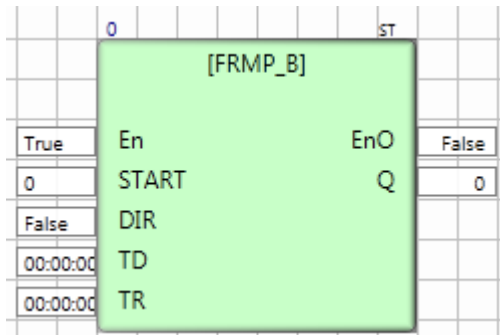


Тип данных входов и выхода – REAL.

Функция возвращает значение полинома 2-й степени от X:

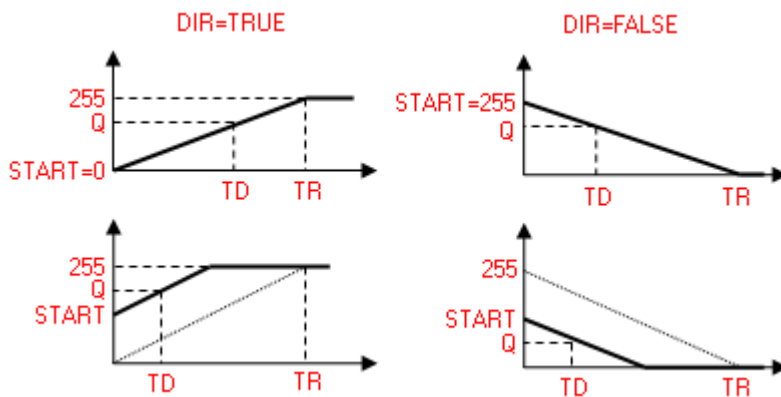
$$Q := A \cdot X^2 + B \cdot X + C$$

### 10.1.2.6. FRMP\_B (OSCAT)

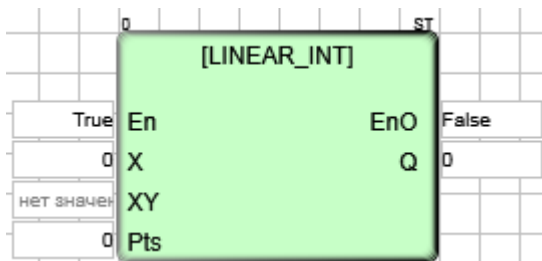


Тип данных START и Q – BYTE, DIR – BOOL, TD и TR – TIME.

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.



### 10.1.2.7. 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  $3 \leq PTS \leq 20$ ).

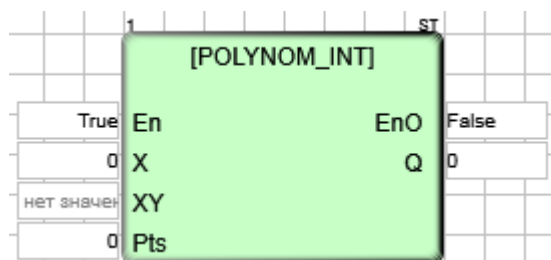
С помощью  $XY$  и  $PTS$  отрезок  $(XY[1,0], XY[PTS,0])$  оси  $X$  разбивается на  $(PTS-1)$  интервалов (при максимально возможном  $PTS$  число интервалов равно 19). В каждом интервале  $LINEAR\_INT$  работает так же, как  $F\_LIN2$  ( $OSCAT$ ), т.е. вычисляет линейно-интерполированное значение ординаты по абсциссе ( $X$ ).

При  $X < XY[1,0]$  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  $X > XY[PTS,0]$  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]	

### 10.1.2.8. 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).

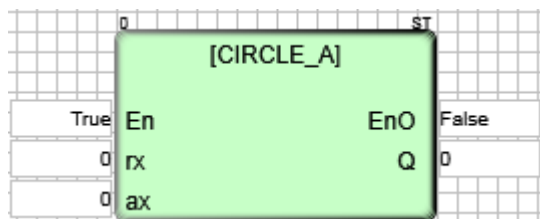
Например, при PTS=4 по следующим точкам XY строится передаточная функция  $Q:=X^3$  и POLYNOM\_INT(2)=8:

Point	[i,0]	[i,1]
[1]	-2	-8
[2]	-1	-1
[3]	0	0
[4]	1	1
[5]		

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.

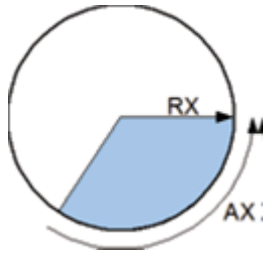
### 10.1.3. OSCAT.Функции.Геометрические функции

#### 10.1.3.1. CIRCLE\_A (OSCAT)



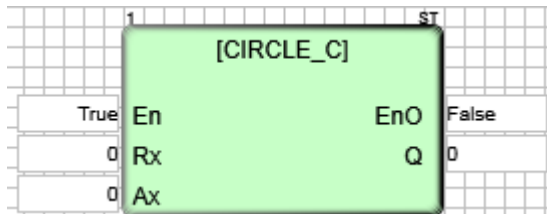
Тип данных входов и выхода – REAL.

Функция вычисляет площадь сектора круга радиуса RX:



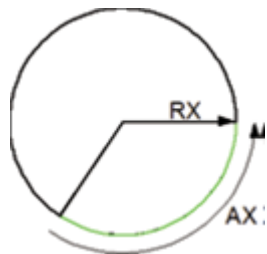
AX задает угол сектора (в градусах).

### 10.1.3.2. CIRCLE\_C (OSCAT)



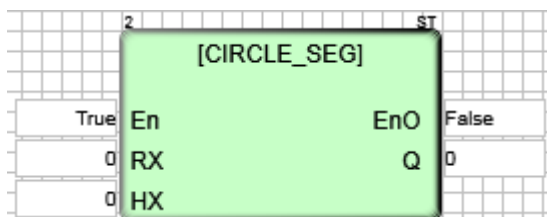
Тип данных входов и выхода – REAL.

Функция вычисляет длину дуги окружности радиуса RX:



AX задает угол дуги (в градусах).

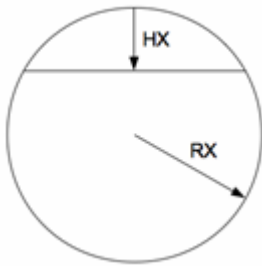
### 10.1.3.3. CIRCLE\_SEG (OSCAT)



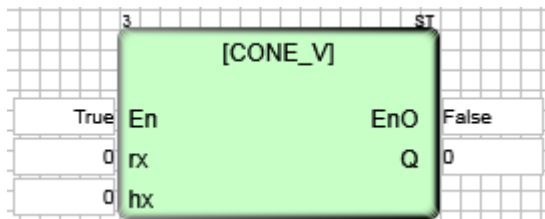
Тип данных входов и выхода – REAL.

Функция вычисляет площадь сегмента круга. RX – радиус круга, HX – высота сегмента:



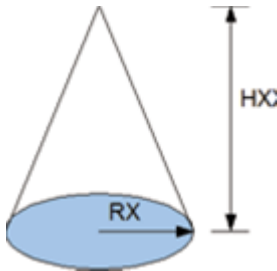


#### 10.1.3.4. CONE\_V (OSCAT)

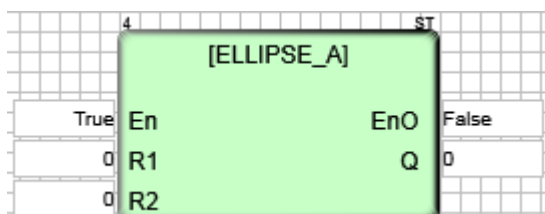


Тип данных входов и выхода – REAL.

Функция вычисляет объем конуса. RX – радиус основания, HX – высота конуса:

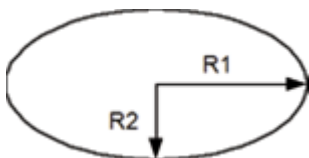


#### 10.1.3.5. ELLIPSE\_A (OSCAT)

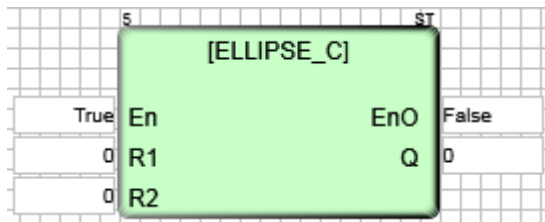


Тип данных входов и выхода – REAL.

Функция вычисляет площадь эллипса по большой (R1) и малой (R2) полуосям:

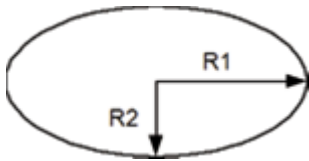


### 10.1.3.6. ELLIPSE\_C (OSCAT)

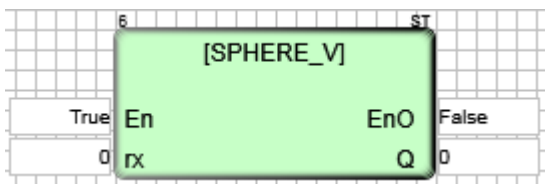


Тип данных входов и выхода – REAL.

Функция вычисляет длину эллипса по большой (R1) и малой (R2) полуосям:



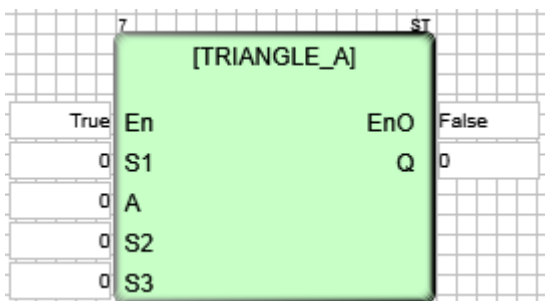
### 10.1.3.7. SPHERE\_V (OSCAT)



Тип данных входа и выхода – REAL.

Функция вычисляет объем шара радиуса RX.

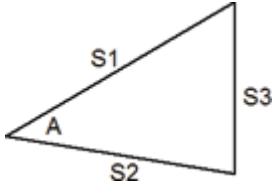
### 10.1.3.8. TRIANGLE\_A (OSCAT)



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):

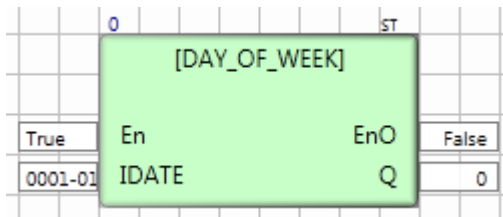
- ▶ если  $A=0$  – по трем сторонам;
- ▶ если  $A \neq 0$  – по двум сторонам  $S1$  и  $S2$  и углу  $A$  между ними.



Используемые функции: RAD (OSCAT).

## 10.1.4. OSCAT. Функции. Дата и время

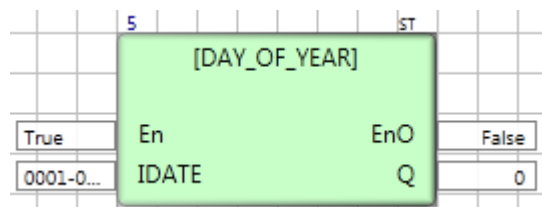
### 10.1.4.1. DAY\_OF\_WEEK (OSCAT)



Тип данных IDATE – DATE, Q – INT.

Функция вычисляет день недели по дате IDATE в соответствии с ISO8601 (1 – понедельник .. 7 – воскресенье). Например,  $DAY\_OF\_WEEK(D\#2007-1-8) = 1$ .

### 10.1.4.2. DAY\_OF\_YEAR (OSCAT)

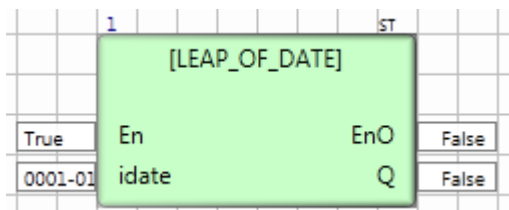


Тип данных IDATE – DATE, Q – INT.

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.

### 10.1.4.3. LEAP\_OF\_DATE (OSCAT)



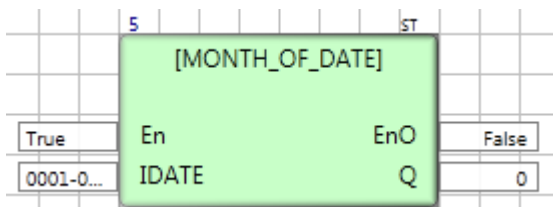
Тип данных IDATE – DATE, Q – BOOL.

Если год, заданный IDATE, – високосный, то Q=TRUE, иначе Q=FALSE.

Функция работает корректно в диапазоне годов [1970, 2099]. Для 2100 года индицируется Q=TRUE, хотя этот год високосным не является. Это не скорректировано, поскольку, согласно IEC 61131-3, даты простираются только до 2106 года.

Пример: LEAP\_OF\_DATE(D#2004-01-12) = TRUE.

#### 10.1.4.4. MONTH\_OF\_DATE (OSCAT)

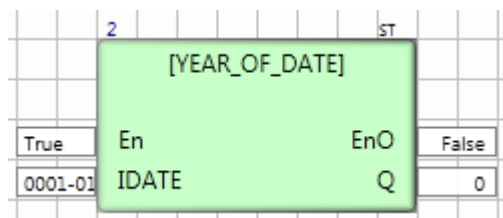


Тип данных входа IDATE – DATE, выхода Q – INT.

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.

#### 10.1.4.5. YEAR\_OF\_DATE (OSCAT)



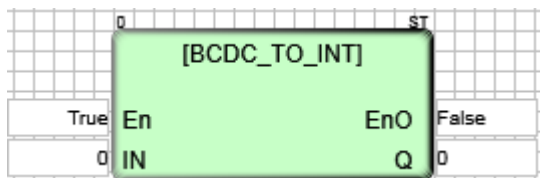
Тип данных IDATE – DATE, Q – INT.

Функция возвращает год по дате IDATE (1970 .. 2099).

Пример: YEAR\_OF\_DATE(D#2007-12-31) = 2007.

## 10.1.5. OSCAT.Функции.Логические модули

### 10.1.5.1. BCDC\_TO\_INT (OSCAT)

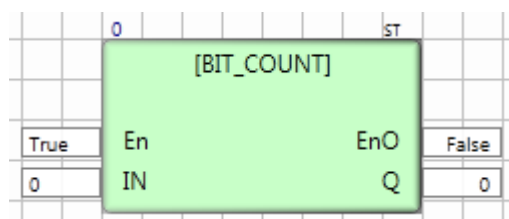


Тип данных входа IN – BYTE, на вход подается значение в формате BCD (не более 2 BCD-разрядов).

Тип данных выхода Q – INT.

BCDC\_TO\_INT обратна функции INT\_TO\_BCDC (OSCAT), т.е. преобразует формат значения из BCD в INT. Например,  $BCDC\_TO\_INT(231)=147$  ( $231=2\#1110\_0111$ ).

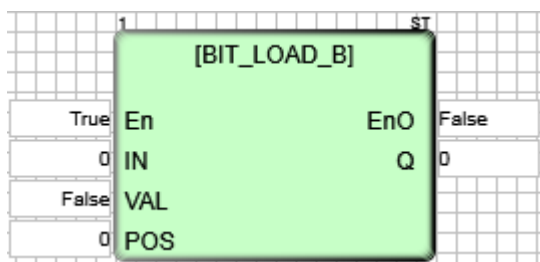
### 10.1.5.2. BIT\_COUNT (OSCAT)



Тип данных IN – DWORD, WORD или BYTE, Q – INT.

Функция возвращает число установленных бит IN. Например,  $bit\_count(6)=2$ .

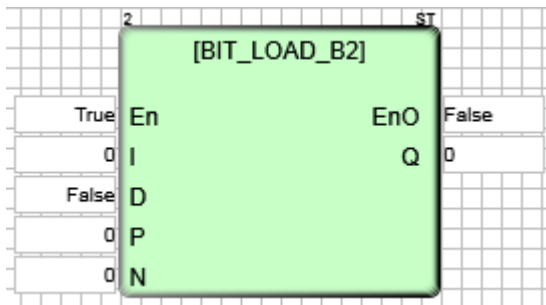
### 10.1.5.3. BIT\_LOAD\_B (OSCAT)



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,  $bit\_load\_b(0,true,3)=8$ .

### 10.1.5.4. 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).

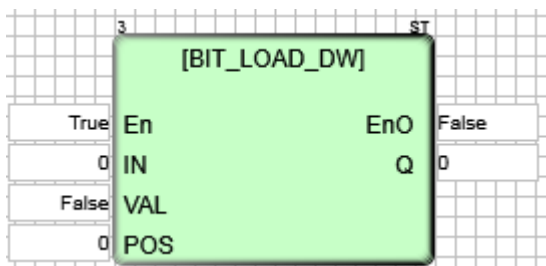
Если комбинация P и N задает номер бита больше 7, из этого номера нужное количество раз вычитается 8.

Примеры:

BIT\_LOAD\_B2(2#1111\_0000, TRUE, 1, 2) = 2#1111\_0110

BIT\_LOAD\_B2(2#1111\_1111, FALSE, 7, 2) = 2#0111\_1110

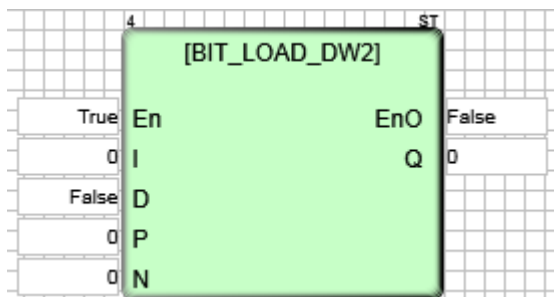
### 10.1.5.5. BIT\_LOAD\_DW (OSCAT)



Тип данных входа IN – DWORD, VAL – BOOL, POS – INT, выхода Q – DWORD.

BIT\_LOAD\_DW – аналог BIT\_LOAD\_B (OSCAT) для значений DWORD.

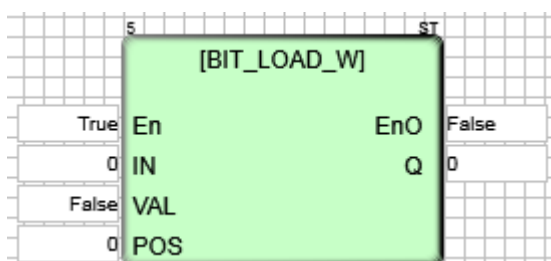
### 10.1.5.6. BIT\_LOAD\_DW2 (OSCAT)



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.

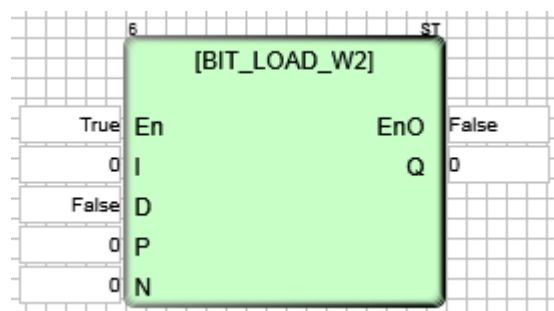
### 10.1.5.7. BIT\_LOAD\_W (OSCAT)



Тип данных входа IN – WORD, VAL – BOOL, POS – INT, выхода Q – WORD.

BIT\_LOAD\_W – аналог BIT\_LOAD\_B (OSCAT) для значений WORD.

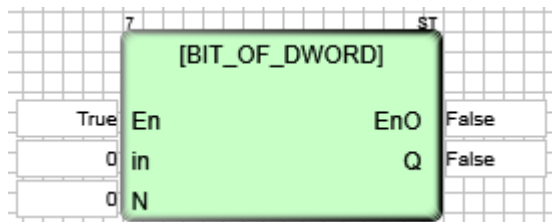
### 10.1.5.8. BIT\_LOAD\_W2 (OSCAT)



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.

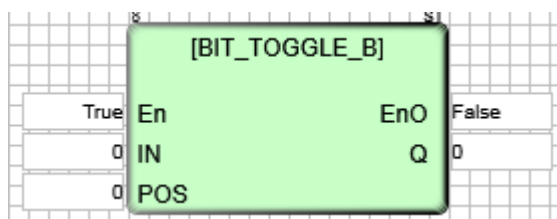
### 10.1.5.9. BIT\_OF\_DWORD (OSCAT)



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.

### 10.1.5.10. BIT\_TOGGLE\_B (OSCAT)



Тип данных входа IN – BYTE, POS – INT, выхода Q – BYTE.

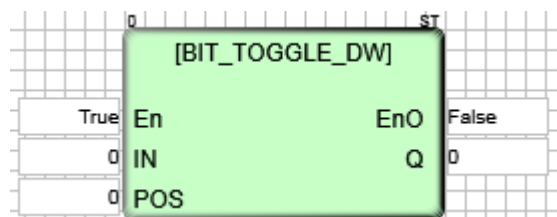
BIT\_TOGGLE\_B инвертирует бит POS в IN.

Примеры:

$\text{BIT\_TOGGLE\_W}(2\#0000\_1111, 2) = 2\#0000\_1011$

$\text{BIT\_TOGGLE\_W}(2\#0000\_1111, 7) = 2\#1000\_1111$

### 10.1.5.11. BIT\_TOGGLE\_DW (OSCAT)

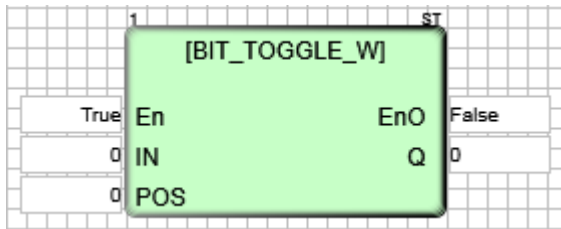


Тип данных входа IN – DWORD, POS – INT, выхода Q – DWORD.

BIT\_TOGGLE\_DW – аналог BIT\_TOGGLE\_B (OSCAT) для значений DWORD.

### 10.1.5.12. BIT\_TOGGLE\_W (OSCAT)

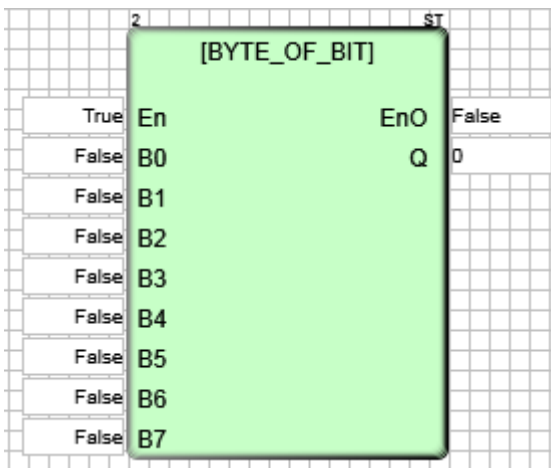




Тип данных входа IN – WORD, POS – INT, выхода Q – WORD.

BIT\_TOGGLE\_W – аналог BIT\_TOGGLE\_B (OSCAT) для значений WORD.

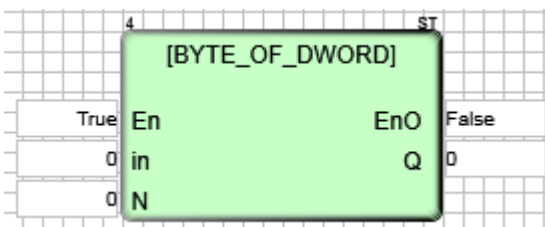
### 10.1.5.13. BYTE\_OF\_BIT (OSCAT)



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.

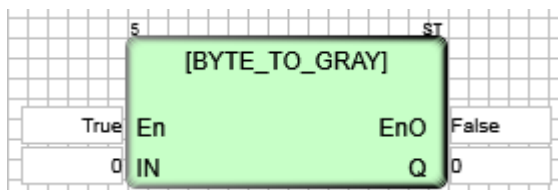
### 10.1.5.14. BYTE\_OF\_DWORD (OSCAT)



Тип данных IN – DWORD, выхода Q – BYTE.

Функция возвращает значение байта N значения IN (N может иметь значение 0, 1, 2 или 3).

### 10.1.5.15. BYTE\_TO\_GRAY (OSCAT)

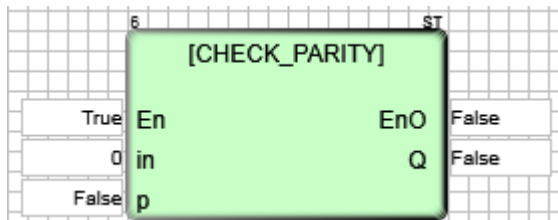


Тип данных входа IN и выхода Q – BYTE.

Функция конвертирует IN в код Грея:

$Q := IN \text{ XOR } \text{SHR}(IN, 1);$

### 10.1.5.16. CHECK\_PARITY (OSCAT)



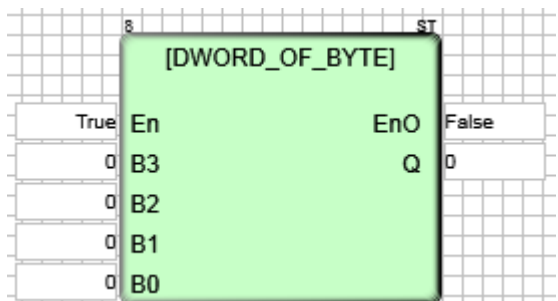
Тип данных входа IN – DWORD, P – BOOL, выхода Q – BOOL.

Функция анализирует биты IN и бит P как единую группу.

Если число установленных битов в группе – нечетное,  $Q := \text{FALSE}$ .

Если число установленных битов в группе – четное,  $Q := \text{TRUE}$ .

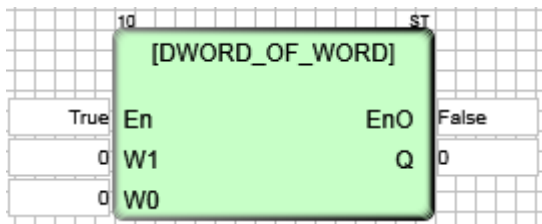
### 10.1.5.17. DWORD\_OF\_BYTE (OSCAT)



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.

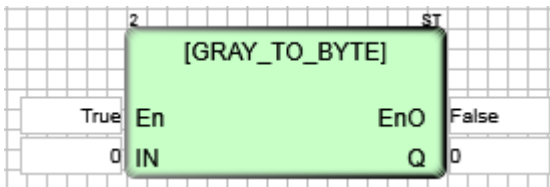
### 10.1.5.18. DWORD\_OF\_WORD (OSCAT)



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.

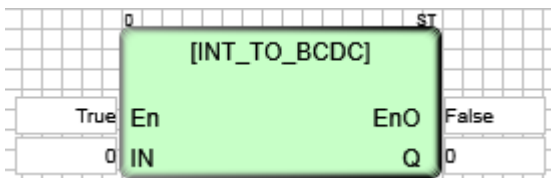
### 10.1.5.19. GRAY\_TO\_BYTE (OSCAT)



Тип данных входа IN и выхода Q – BYTE.

GRAY\_TO\_BYTE обратна функции BYTE\_TO\_GRAY (OSCAT), т.е. декодирует код Грея.

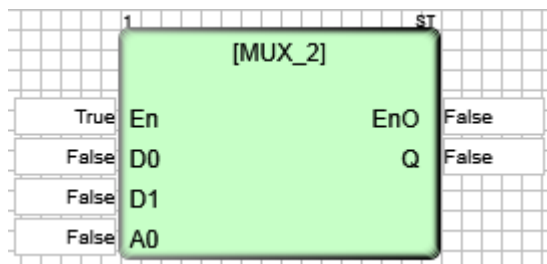
### 10.1.5.20. INT\_TO\_BCDC (OSCAT)



Тип данных входа IN – INT, выхода Q – BYTE.

INT\_TO\_BCDC конвертирует значение IN в формат BCD (для обратного преобразования используется BCDC\_TO\_INT (OSCAT)).

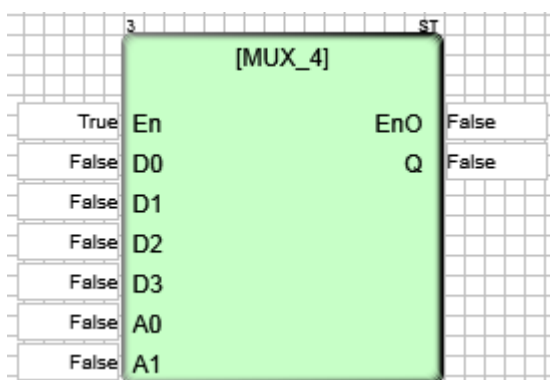
### 10.1.5.21. MUX\_2 (OSCAT)



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).

### 10.1.5.22. MUX\_4 (OSCAT)

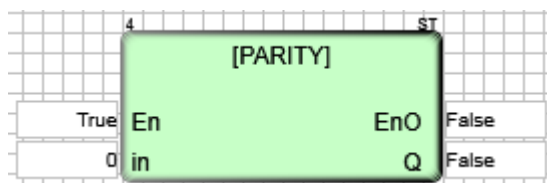


Тип данных входов и выхода – BOOL.

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.

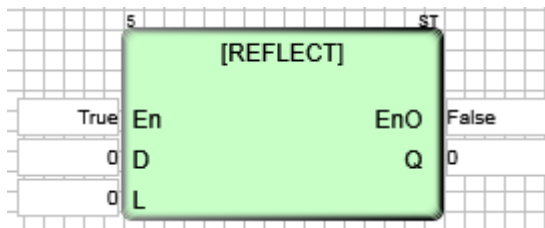
### 10.1.5.23. PARITY (OSCAT)



Тип данных IN – DWORD, выхода Q – BOOL.

Q=TRUE, если число установленных бит в IN – нечетное (проверка на нечетность).

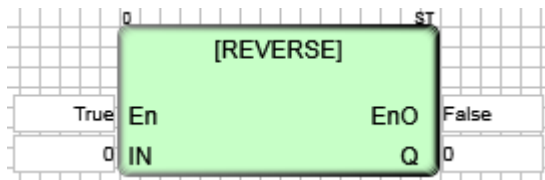
### 10.1.5.24. REFLECT (OSCAT)



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.

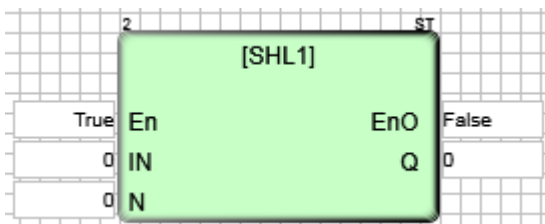
### 10.1.5.25. REVERSE (OSCAT)



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.

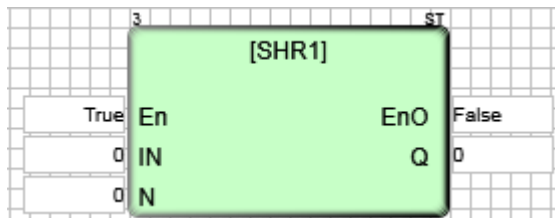
### 10.1.5.26. SHL1 (OSCAT)



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  $shl\_1(1, 3)=15$ .

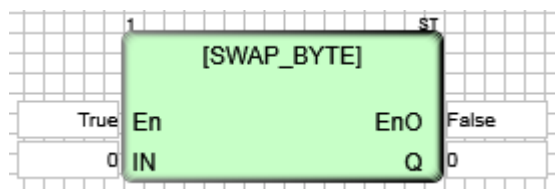
### 10.1.5.27. SHR1 (OSCAT)



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  $\text{shr}(16\#\text{FFFFFFFF}, 31)=1$ , то  $\text{shr}_1(16\#\text{FFFFFFFF}, k)=16\#\text{FFFFFFFF}$  при любом k.

### 10.1.5.28. SWAP\_BYTE (OSCAT)

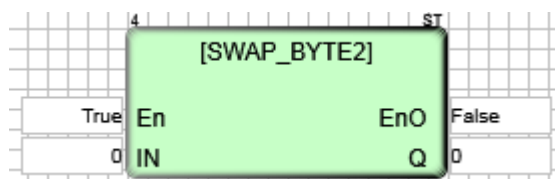


Тип данных входа IN и выхода Q – WORD.

Функция выполняет перестановку байтов в IN.

Например,  $\text{SWAP\_BYTE}(16\#33df) = 16\#df33$ .

### 10.1.5.29. SWAP\_BYTE2 (OSCAT)

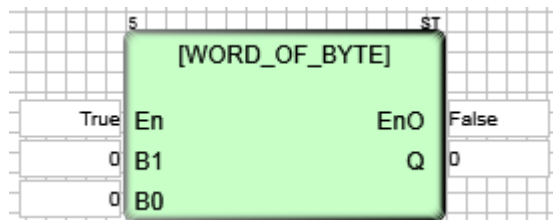


Тип данных входа IN и выхода Q – DWORD.

Функция зеркально изменяет порядок байтов в IN.

Например,  $\text{SWAP\_BYTE2}(16\#33df1122) = 16\#2211df33$ .

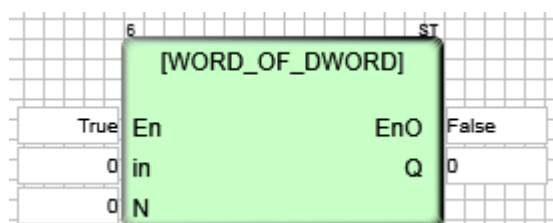
### 10.1.5.30. WORD\_OF\_BYTE (OSCAT)



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.

### 10.1.5.31. WORD\_OF\_DWORD (OSCAT)



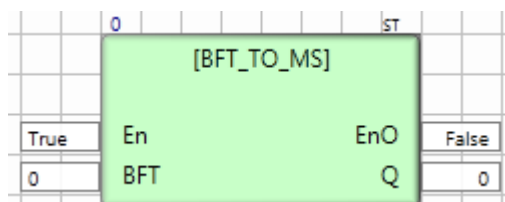
Тип данных входа IN – DWORD, N – BYTE, выхода Q – WORD.

Функция возвращает значение слова N значения IN (N = 0 или 1).

Например, WORD\_OF\_DWORD(65537, N)=1 при любом N.

## 10.1.6. OSCAT.Функции.Преобразования

### 10.1.6.1. BFT\_TO\_MS (OSCAT)

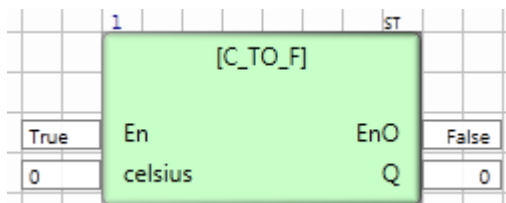


Тип данных BFT – INT, Q – REAL.

Функция переводит скорость ветра по бофортской шкале в м/с по формуле

$$\text{BFT\_TO\_MS} = 0.836 * \text{BFT}^{3/2} \text{ [m/s]}$$

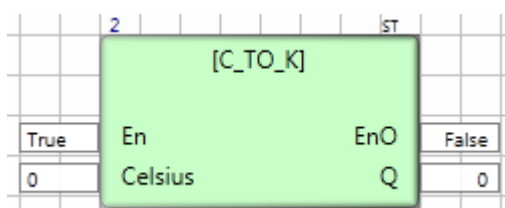
### 10.1.6.2. C\_TO\_F (OSCAT)



Тип данных CELSIUS и Q – REAL.

Функция переводит температуру по Цельсию в температуру по Фаренгейту.

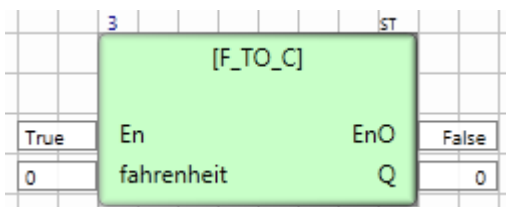
### 10.1.6.3. C\_TO\_K (OSCAT)



Тип данных CELSIUS и Q – REAL.

Функция переводит температуру по Цельсию в температуру по Кельвину.

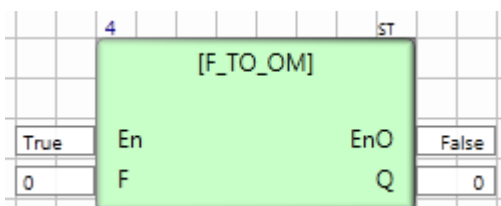
### 10.1.6.4. F\_TO\_C (OSCAT)



Тип данных FAHRENHEIT и Q – REAL.

Функция переводит температуру по Фаренгейту в температуру по Цельсию.

### 10.1.6.5. F\_TO\_OM (OSCAT)



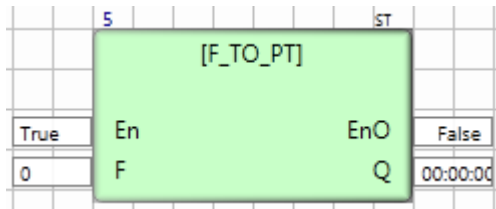
Тип данных F и Q – REAL.



Функция вычисляет циклическую частоту по заданной частоте:

$$Q := 2\pi F$$

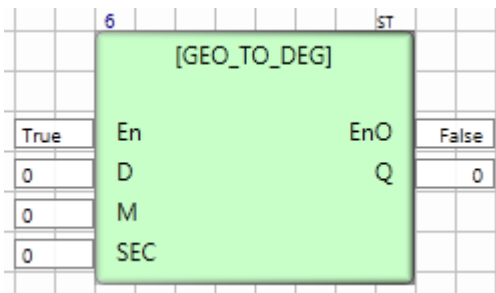
### 10.1.6.6. F\_TO\_PT (OSCAT)



Тип данных F – REAL, Q – TIME.

Функция вычисляет период по заданной частоте в герцах.

### 10.1.6.7. 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.

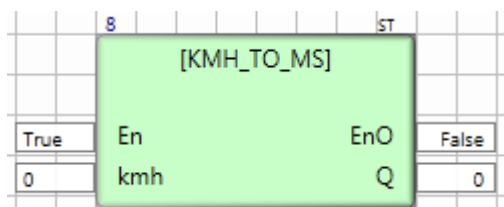
### 10.1.6.8. K\_TO\_C (OSCAT)



Тип данных KELVIN и Q – REAL.

Функция переводит температуру по Кельвину в температуру по Цельсию.

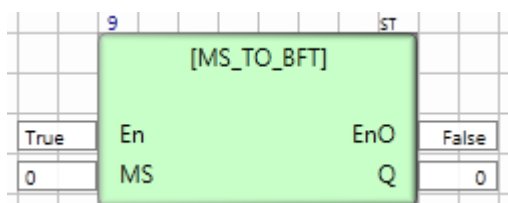
### 10.1.6.9. КМН\_TO\_MS (OSCAT)



Тип данных КМН и Q – REAL.

Функция переводит скорость в км/ч в м/с.

### 10.1.6.10. MS\_TO\_BFT (OSCAT)

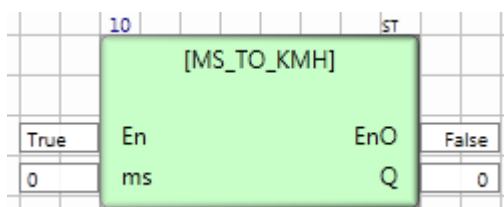


Тип данных MS – REAL, Q – INT.

Функция переводит скорость ветра в м/с в бофорову шкалу по формуле

$$MS\_TO\_BFT = (MS * 1.196172)**(2/3)$$

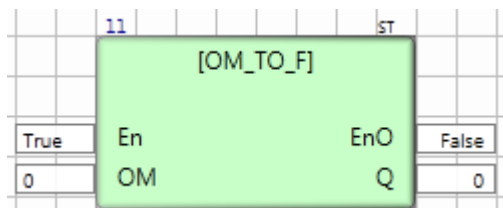
### 10.1.6.11. MS\_TO\_KMH (OSCAT)



Тип данных MS и Q – REAL.

Функция переводит скорость в м/с в км/ч.

### 10.1.6.12. OM\_TO\_F (OSCAT)

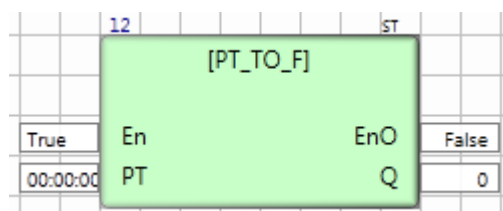


Тип данных OM и Q – REAL.

Функция вычисляет частоту по заданной циклической частоте:

$$Q := \frac{OM}{2\pi}$$

### 10.1.6.13. PT\_TO\_F (OSCAT)

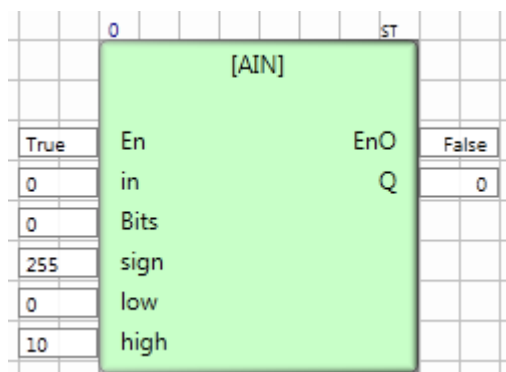


Тип данных PT – TIME. Q – REAL.

Функция вычисляет частоту в герцах по заданному периоду PT:

## 10.1.7. OSCAT.Функции.Обработка сигналов

### 10.1.7.1. 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.

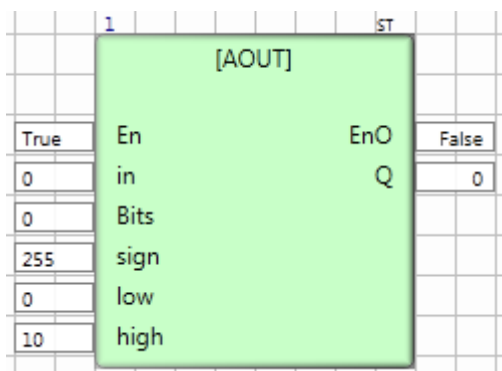
Конфигурационные параметры:

- 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.

### 10.1.7.2. 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 AOUT – модель АЦП, конфигурируется для преобразования входа IN таким образом, чтобы на выходе AOUT 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:

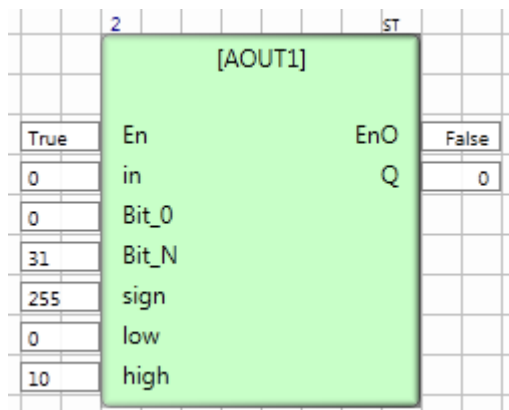
- BITS задает разрядность ЦАП. Для 12-разрядного ЦАП должно быть задано BITS=12;
- SIGN задает, требуется ли вставка знакового бита, и где этот бит должен быть расположен в Q. SIGN=255 (значение по умолчанию) означает, что знаковый бит не требуется, SIGN=15 означает, что знаковый бит будет вставлен в 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).

### 10.1.7.3. 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).

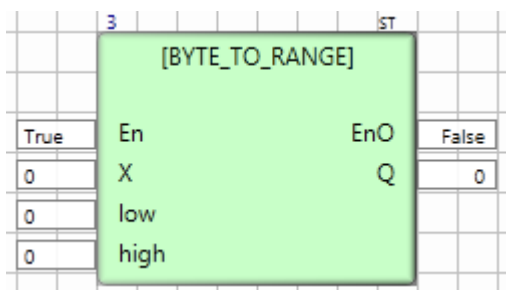
Выходы функции:

- 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).

#### 10.1.7.4. BYTE\_TO\_RANGE (OSCAT)

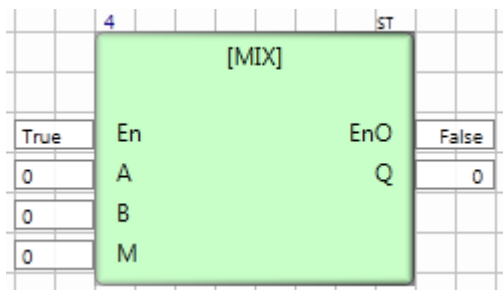


Тип данных X – BYTE, LOW, HIGH и Q – REAL.

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).

#### 10.1.7.5. MIX (OSCAT)

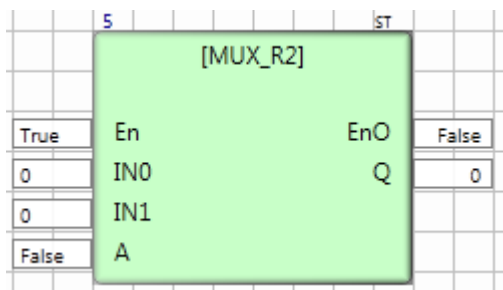


Тип данных A, B, M и Q – REAL.

Функция представляет собой аналоговый микшер:

$$Q := (1 - M) * A + M * B;$$

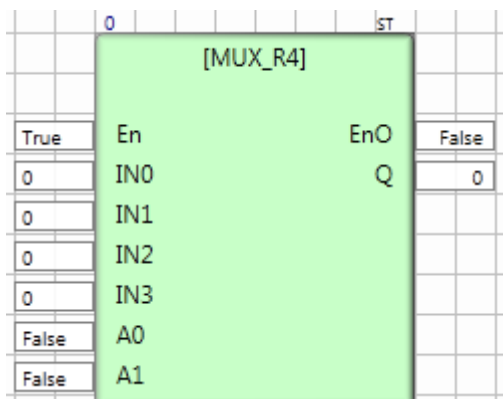
### 10.1.7.6. MUX\_R2 (OSCAT)



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.

### 10.1.7.7. MUX\_R4 (OSCAT)



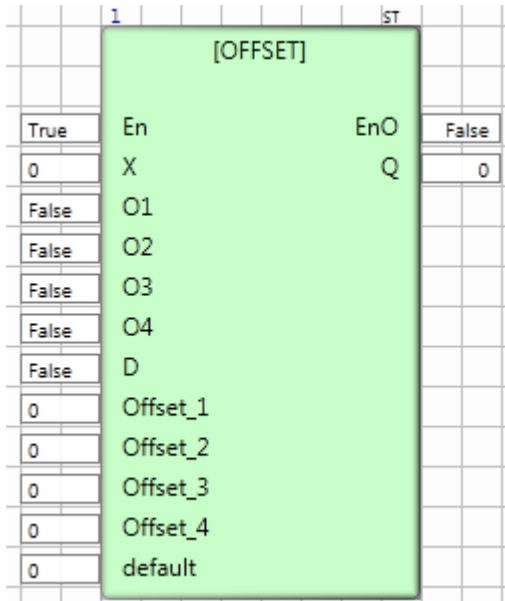
Тип данных IN0, IN1, IN2, IN3 и Q – REAL, A0 и A1 – BOOL.



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.

### 10.1.7.8. OFFSET (OSCAT)

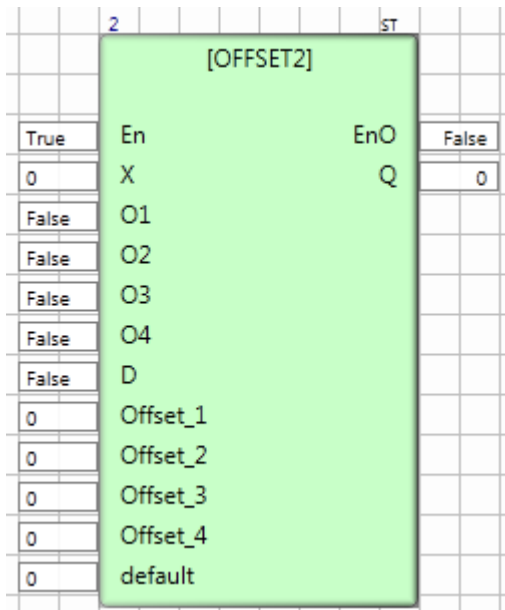


Тип данных X, OFFSET\_1, OFFSET\_2, OFFSET\_3, OFFSET\_4, DEFAULT и Q – REAL, O1, O2, O3, O4 и D – BOOL.

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.

### 10.1.7.9. OFFSET2 (OSCAT)

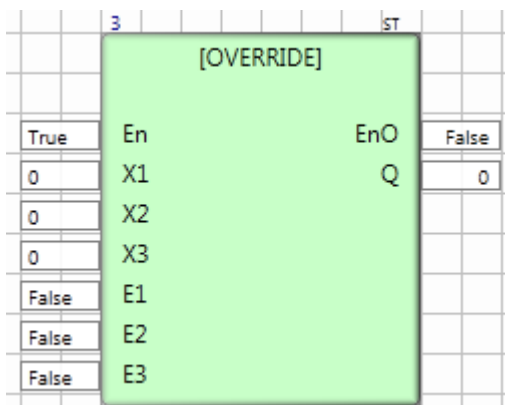


Тип данных X, OFFSET\_1, OFFSET\_2, OFFSET\_3, OFFSET\_4, DEFAULT и Q – REAL, O1, O2, O3, O4 и D – BOOL.

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.

### 10.1.7.10. OVERRIDE (OSCAT)

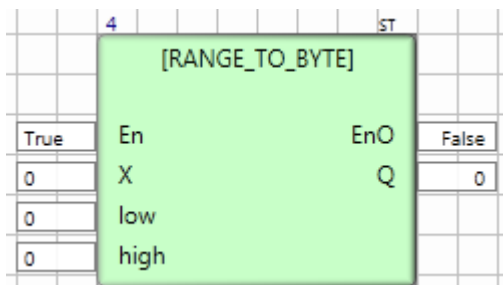


Тип данных X1, X2, X3 и Q – REAL, E1, E2 и E3 – BOOL.

Входы E1..E3 разрешают (TRUE) или запрещают (FALSE) обработку входов X1..X3 (соответственно).

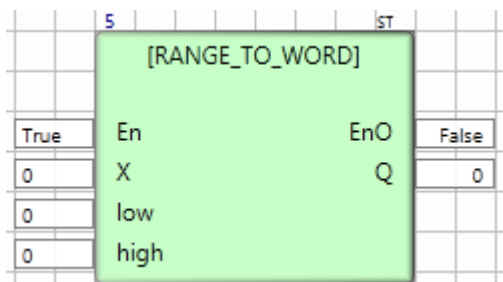
На выход Q передается значение входа X с максимальным абсолютным значением среди обрабатываемых входов X.

### 10.1.7.11. RANGE\_TO\_BYTE (OSCAT)



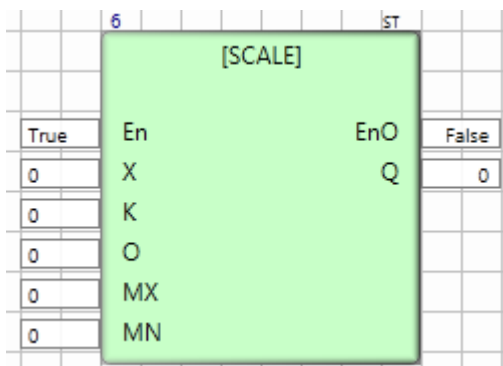
Эта функция обратна функции BYTE\_TO\_RANGE (OSCAT).

### 10.1.7.12. RANGE\_TO\_WORD (OSCAT)



Эта функция обратна функции WORD\_TO\_RANGE (OSCAT).

### 10.1.7.13. SCALE (OSCAT)

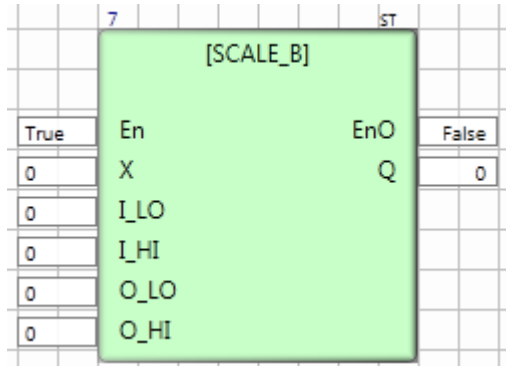


Тип данных входов и выхода – REAL.

SCALE умножает X на K и прибавляет смещение O. Результат ограничивается значениями MN и MX и присваивается выходу Q:

$$\text{SCALE} = \text{LIMIT}(\text{MN}, \text{X} * \text{K} + \text{O}, \text{MX})$$

### 10.1.7.14. SCALE\_B (OSCAT)

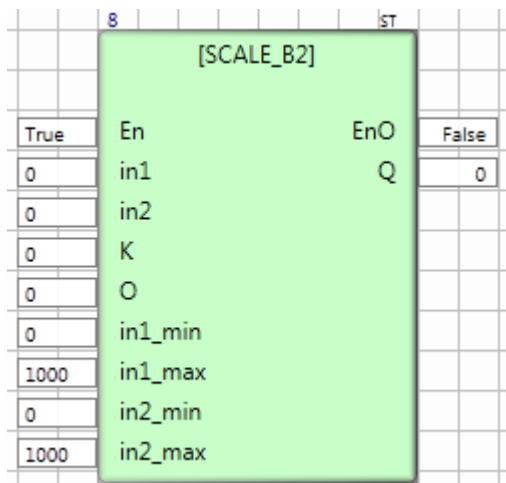


Тип данных X, I\_LO и I\_HI – BYTE, O\_LO, O\_HI и Q – REAL.

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.

### 10.1.7.15. SCALE\_B2 (OSCAT)



Входы функции:

- IN1 (тип данных BYTE) – первый вход;
- IN2 (тип данных BYTE) – второй вход;
- K (тип данных REAL) – множитель;

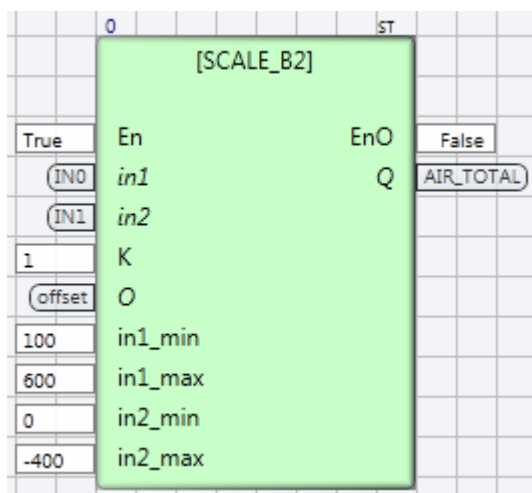
- O (тип данных REAL) – смещение;
- IN1\_MIN (тип данных REAL) – нижняя граница диапазона, в который масштабируется IN1;
- IN1\_MAX (тип данных REAL) – верхняя граница диапазона, в который масштабируется IN1;
- IN2\_MIN (тип данных REAL) – нижняя граница диапазона, в который масштабируется IN2;
- IN2\_MAX (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



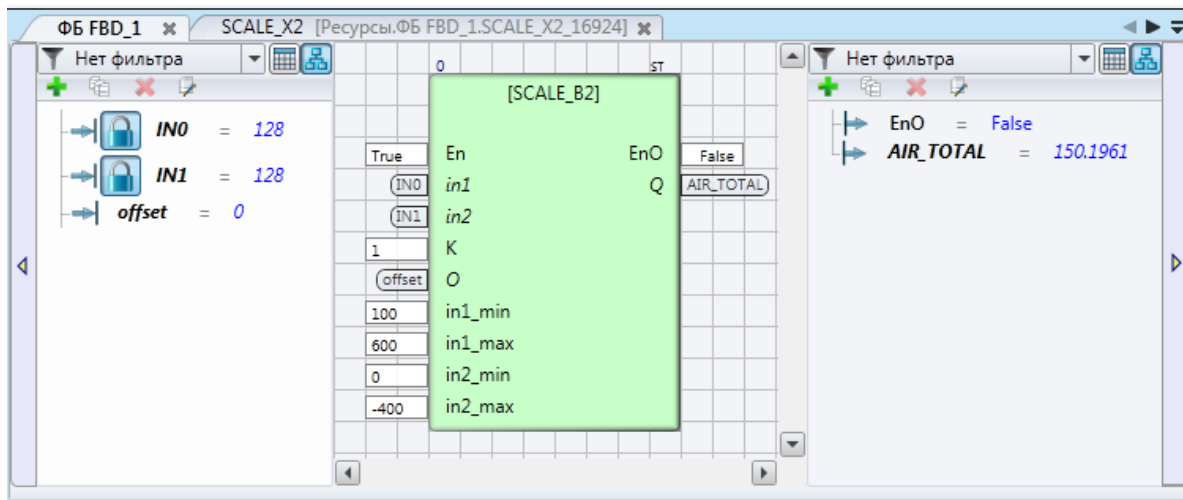
IN0 – воздушный клапан, который регулирует приток воздуха в диапазоне от 100м<sup>3</sup>/ч до 600м<sup>3</sup>/ч при изменении IN0 в диапазоне [0, 255].

IN1 – вытяжка, от 0м<sup>3</sup>/ч до 400м<sup>3</sup>/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)).

### 10.1.7.16. 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.



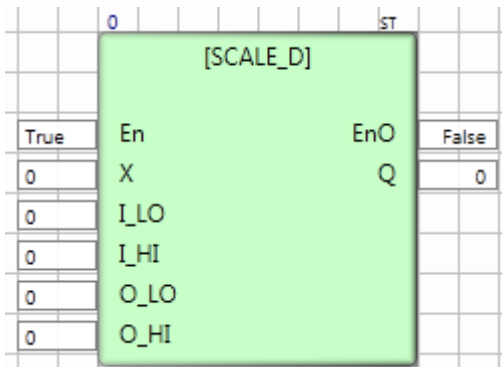
### 10.1.7.17. SCALE\_B8 (OSCAT)

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:

$$Q := (in1*(IN1\_MAX-IN1\_MIN)/255 + IN1\_MIN + in2*(IN2\_MAX-IN2\_MIN)/255 + IN2\_MIN + in3*(IN3\_MAX-IN3\_MIN)/255 + IN3\_MIN + in4*(IN4\_MAX-IN4\_MIN)/255 + IN4\_MIN + in5*(IN5\_MAX-IN5\_MIN)/255 + IN5\_MIN + in6*(IN6\_MAX-IN6\_MIN)/255 + IN6\_MIN + in7*(IN7\_MAX-IN7\_MIN)/255 + IN7\_MIN + in8*(IN8\_MAX-IN8\_MIN)/255 + IN8\_MIN)*K + O$$

### 10.1.7.18. SCALE\_D (OSCAT)



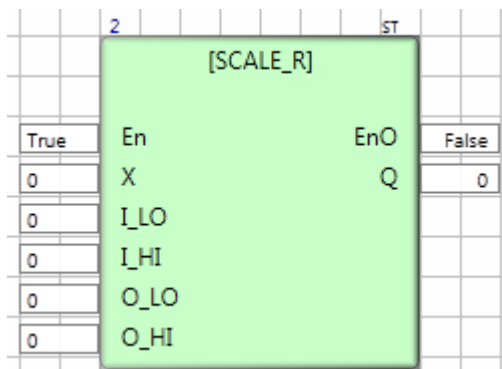
Тип данных X, I\_LO и I\_HI – DWORD, O\_LO, O\_HI и Q – REAL.

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.

### 10.1.7.19. 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  $I\_LO < I\_HI$ .

### 10.1.7.20. SCALE\_X2 (OSCAT)

1		ST	
[SCALE_X2]			
True	En	EnO	False
False	IN1	Q	0
False	IN2		
0	K		
0	O		
0	IN1_MIN		
1000	IN1_MAX		
0	IN2_MIN		
1000	IN2_MAX		

Тип данных IN1 и IN2 – BOOL, остальных входов и выхода – REAL.

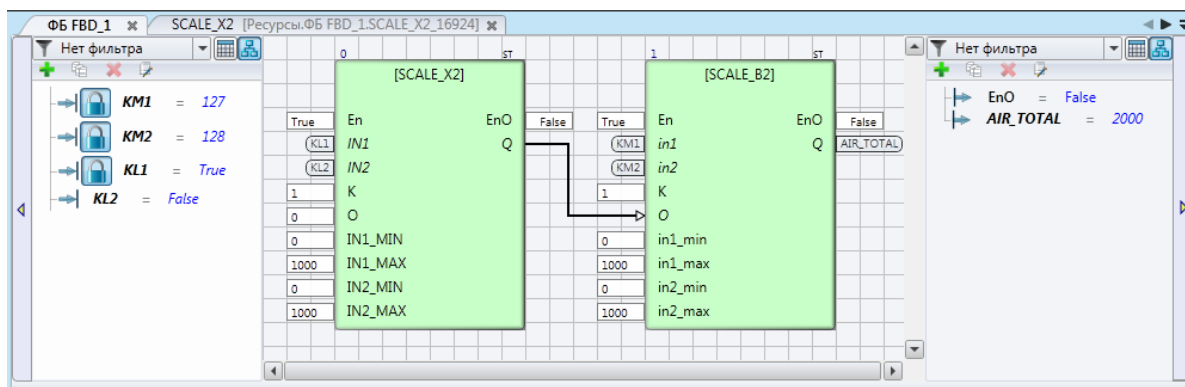
SCALE\_X2 использует две функции SEL, выбор из двух, суммирует 2 выбранные значения, умножает сумму на множитель K и прибавляет смещение O.

Первое значение для суммирования выбирается следующим образом: если IN1=FALSE, выбирается IN1\_MIN, если IN1=TRUE, выбирается IN1\_MAX.

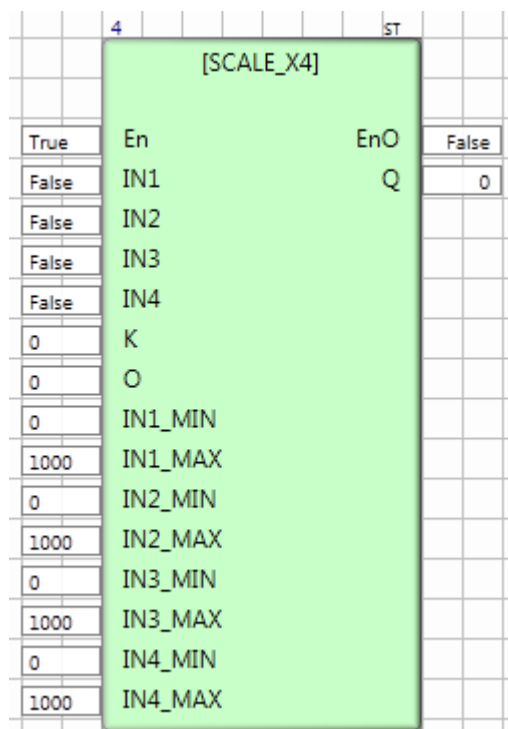
Второе значение для суммирования выбирается аналогично: если IN2=FALSE, выбирается IN2\_MIN, если IN2=TRUE, выбирается IN2\_MAX.

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:



### 10.1.7.21. SCALE\_X4 (OSCAT)



То же, что и SCALE\_X2 (OSCAT), но суммирует 4 значения (в SCALE\_X4 используется 4 функции SEL).

### 10.1.7.22. SCALE\_X8 (OSCAT)

5		ST	
[SCALE_X8]			
True	En	EnO	False
False	in1	Q	0
False	in2		
False	in3		
False	in4		
False	in5		
False	in6		
False	in7		
False	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\_X2 (OSCAT), но суммирует 8 значений (в SCALE\_X8 используется 8 функций SEL).

### 10.1.7.23. STAIR (OSCAT)

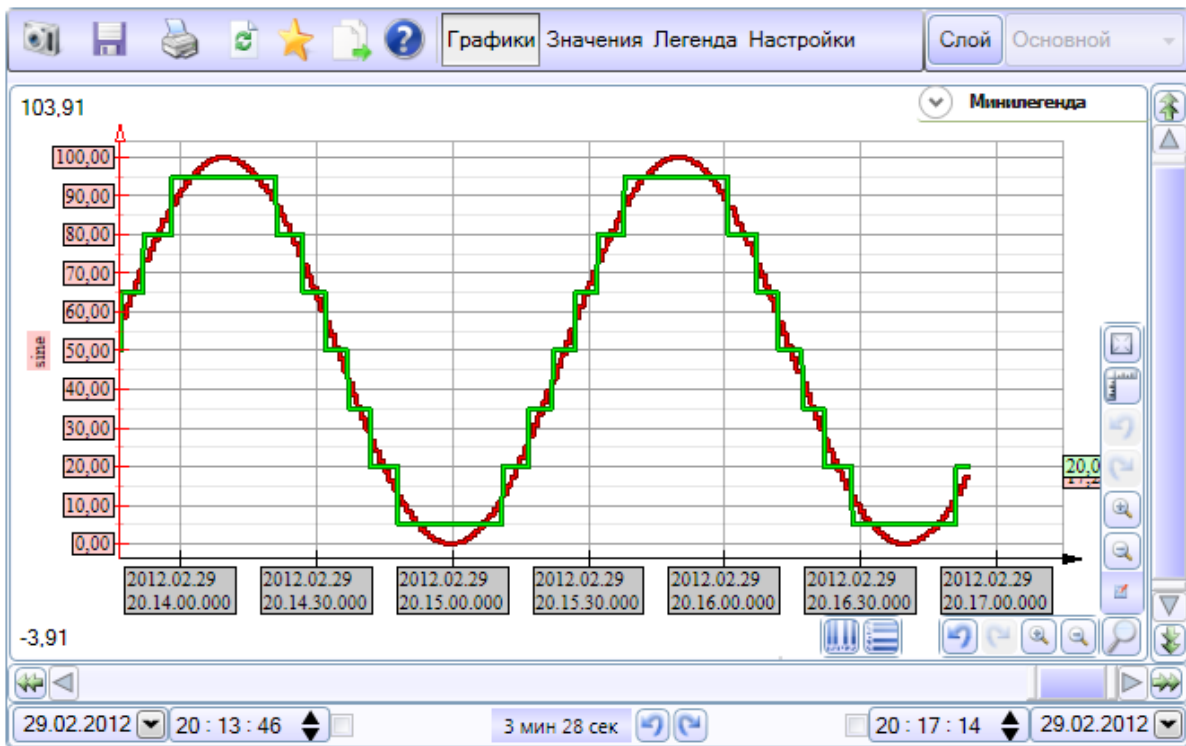
3		ST	
[STAIR]			
True	En	EnO	False
0	X	Q	0
0	D		

Тип данных входов и выхода функции – REAL.

Если  $D \leq 0$ ,  $Q := X$ .

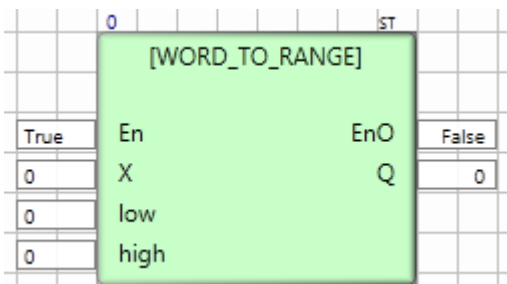
Если  $D > 0$ , функция преобразует гладкий сигнал, поданный на вход  $X$ , в ступенчатый с величиной шага  $D$  по оси значений (выход  $Q$  принимает только значения, кратные  $D$ ).

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).

### 10.1.7.24. WORD\_TO\_RANGE (OSCAT)

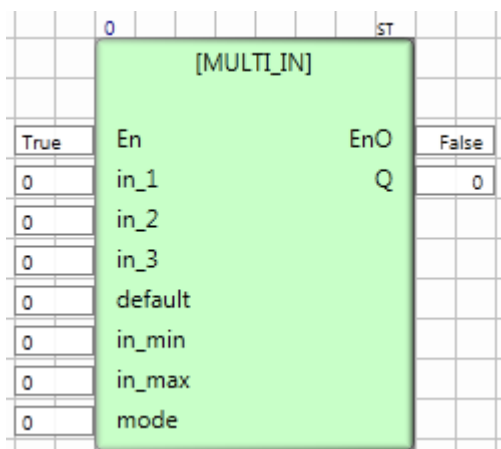


Тип данных X – WORD, LOW, HIGH и Q – REAL.

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).

## 10.1.8. OSCAT.Функции.Датчики

### 10.1.8.1. 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;

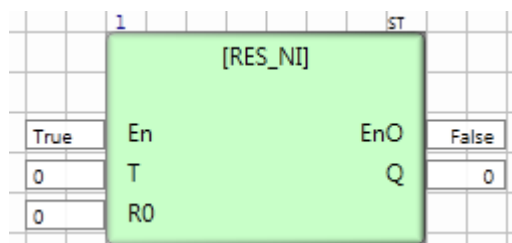
- $>7$  – MULTI\_IN возвращает 0.

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 IN3 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.

### 10.1.8.2. 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		

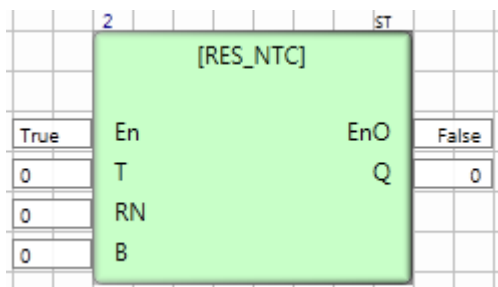
Функция вычисляет сопротивление никелевого терморезистивного датчика в диапазоне температур [-60, +180]0C по следующей формуле:



$$\text{RES\_NI} := R_0 + A \cdot T + B \cdot T^2 + C \cdot T^4$$

где  $A = 0.5485$ ,  $B = 0.665e-3$  и  $C = 2.805e-9$ .

### 10.1.8.3. RES\_NTC (OSCAT)



Тип данных входов и выхода – REAL.

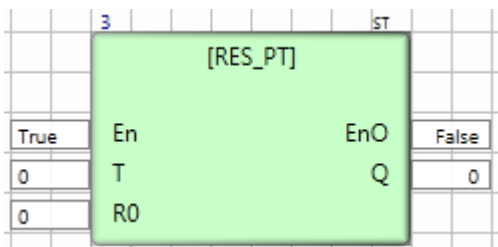
RES\_NTC вычисляет сопротивление терморезистивного NTC-датчика в зависимости от значений входов T (температура в 0C) и RN (сопротивление при 250C). Вход B – константа, значение которой должно быть взято из описания датчика.

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.

### 10.1.8.4. 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	35,53	-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]°C.

При  $T > 00\text{C}$  используется формула

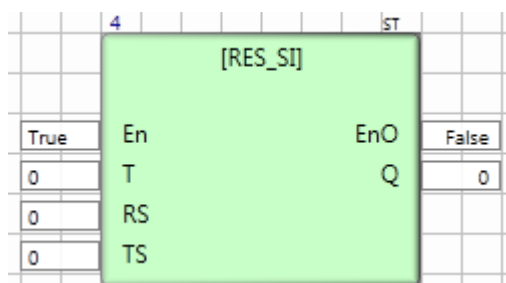
$$\text{RES\_PT} = R0 * (1 + A * T + B * T^2)$$

При  $T < 00\text{C}$  используется формула

$$\text{RES\_PT} = R0 * (1 + A * T + B * T^2) + C * (T - 100) * T^3$$

где  $A = 3.90802\text{e-}3$ ,  $B = -5.80195\text{e-}7$ ,  $C = -427350\text{e-}12$ .

### 10.1.8.5. 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).

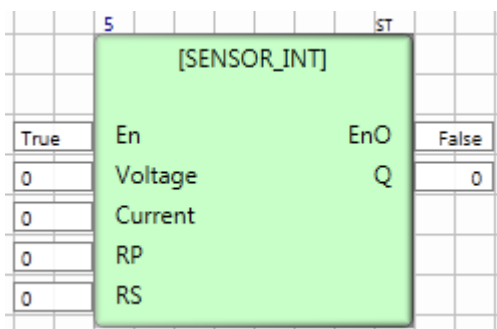
В отличие от функций RES\_NI (OSCAT), RES\_NTC (OSCAT) и RES\_PT (OSCAT), для которых реперное значение сопротивления задается при фиксированной температуре, сопротивление RS может быть задано при произвольной температуре TS (например, при 250C для KTY-10).

Для вычислений используется следующая формула:

$$RES\_SI = RS + A*(T - TS) + B*(T - TS)^2$$

где  $A = 7.64e-3$ ,  $B = 1.66e-5$

### 10.1.8.6. 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.

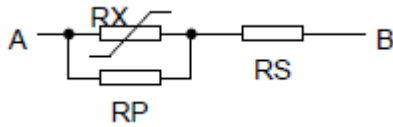
Выход функции:

- 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

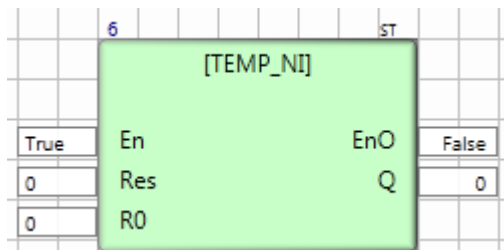
току, – это не только сопротивление датчика, оно включает также два паразитных сопротивления RS и RP:



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.

### 10.1.8.7. TEMP\_NI (OSCAT)



Входы функции:

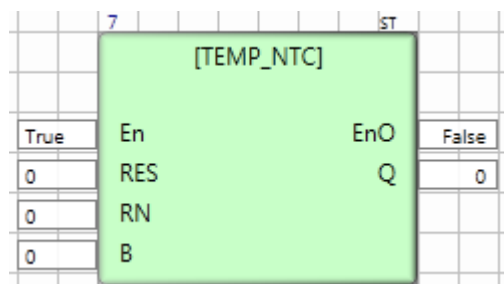
- RES (тип данных REAL) – измеренное сопротивление Ni-термистора, Ом;
- R0 (тип данных REAL) – сопротивление Ni-термистора при 00С.

Выход функции:

- Q (тип данных REAL) – температура в 0С, соответствующая RES.

Данная функция обратна функции RES\_NI (OSCAT).

### 10.1.8.8. 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.

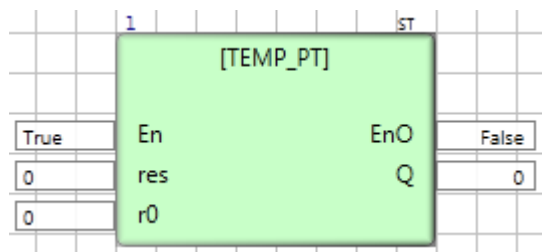
Выход функции:

- Q (тип данных REAL) – температура в 0С, соответствующая RES.

Данная функция обратна функции RES\_NTC (OSCAT):

$$Q := B * 298.15 / (B + \ln(RES/RN) * 298.15) - 273.15$$

### 10.1.8.9. TEMP\_PT (OSCAT)



Входы функции:

- RES (тип данных REAL) – измеренное сопротивление Pt-термистора, Ом;
- R0 (тип данных REAL) – сопротивление Pt-термистора при 00С.

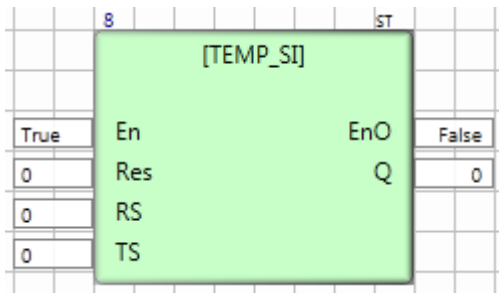
Выход функции:

- Q (тип данных REAL) – температура в 0С, соответствующая RES.

Данная функция обратна функции RES\_PT (OSCAT).

Важно! TEMP\_PT переписана без использования указателей.

### 10.1.8.10. 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.

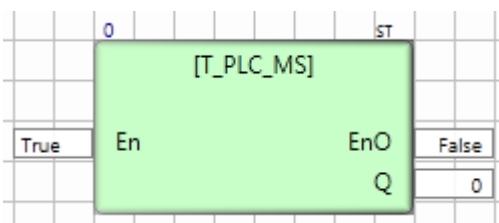
Выход функции:

- Q (тип данных REAL) – температура в 0С, соответствующая RES.

Данная функция обратна функции RES\_SI (OSCAT).

## 10.1.9. OSCAT.Функции.Модули измерения

### 10.1.9.1. 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;

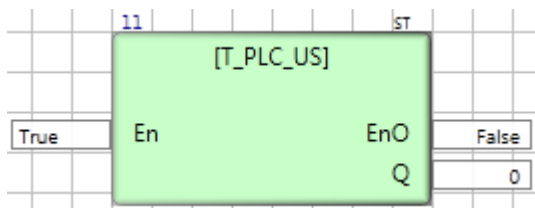
```

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.

### 10.1.9.2. 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();
```

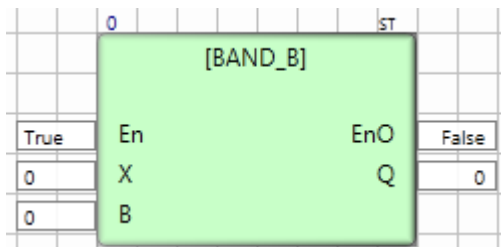
```
T_PLC_US := TIME_TO_DWORD(Tx)*1000;
```

**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.

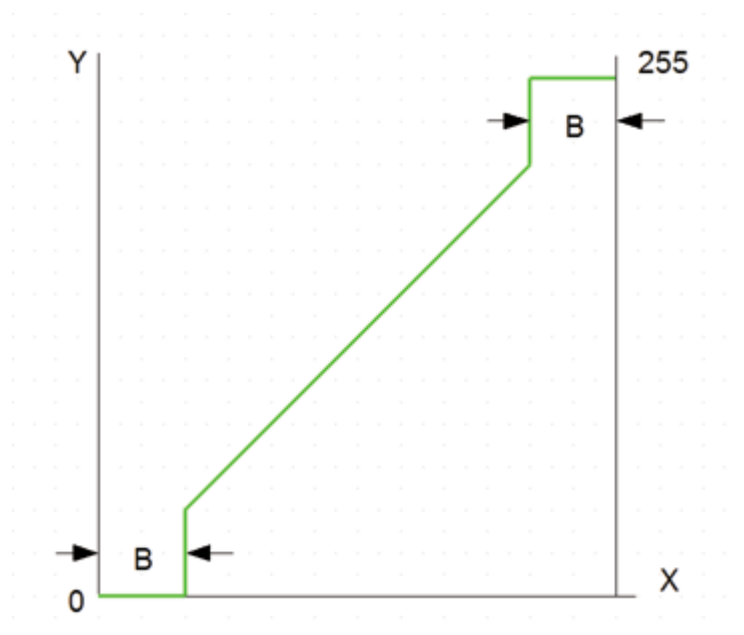
**10.1.10. OSCAT.Функции.Модули регулирования****10.1.10.1. BAND\_B (OSCAT)**

Тип данных входов и выхода – BYTE.

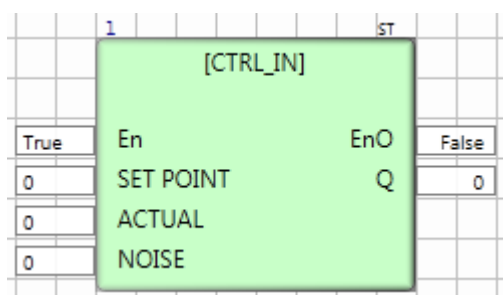
Выход Q равен значению входа X везде, кроме следующих областей:

- [0, B) – в этой области значений X Q=0;
- (255-B, 255] – в этой области значений X Q=255.





### 10.1.10.2. 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.

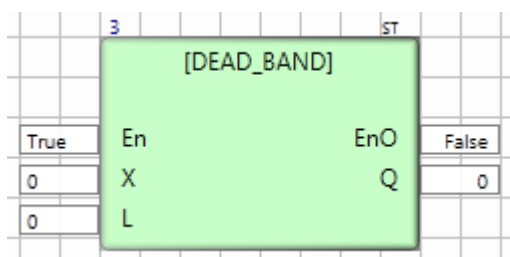
CTRL\_IN вычисляет отклонение процесса от уставки (SET\_POINT - ACTUAL). Если разница меньше NOISE, выход остается равным 0.

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).

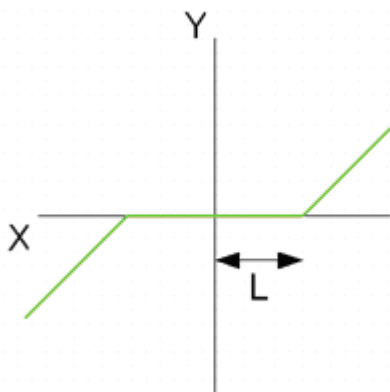
### 10.1.10.3. DEAD\_BAND (OSCAT)



Тип данных входов и выхода – REAL.

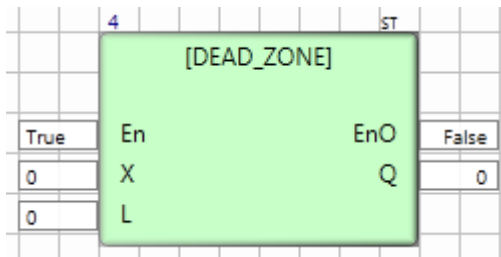
DEAD\_BAND – безударная линейная передаточная функция с зоной нечувствительности шириной  $2L$  ( $L > 0$ ):

- 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.

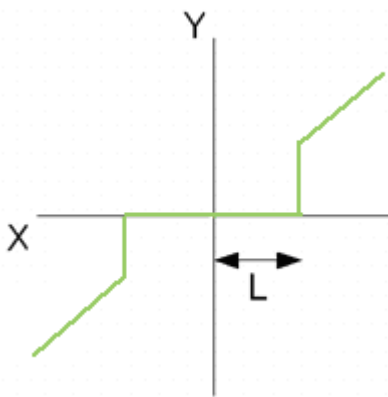
### 10.1.10.4. DEAD\_ZONE (OSCAT)



Тип данных входов и выхода – REAL.

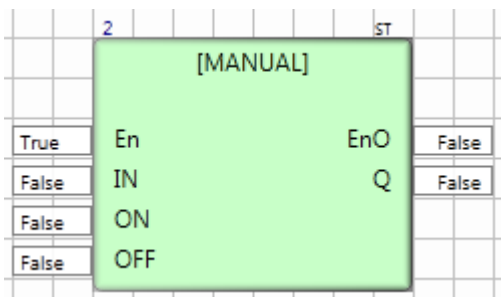
DEAD\_ZONE – ударная линейная передаточная функция с зоной нечувствительности шириной  $2L$  ( $L > 0$ ):

- если  $X < -L$  или  $X > L$ ,  $Q := X$ ;
- если  $-L \leq X \leq L$ ,  $Q = 0$ .



### 10.1.11. OSCAT.Функции.Управление устройствами

#### 10.1.11.1. MANUAL (OSCAT)

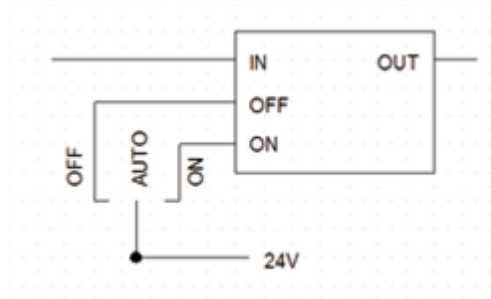


Тип данных входов и выхода функции – BOOL.

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:



## 10.2. OSCAT. Функциональные блоки

OSCAT.ФБ.Арифметические функции

FT\_AVG (OSCAT)

FT\_MIN\_MAX (OSCAT)

FT\_RMP (OSCAT)

OSCAT.ФБ.Модули памяти

FIFO\_16 (OSCAT)

FIFO\_32 (OSCAT)

STACK\_16 (OSCAT)

STACK\_32 (OSCAT)

OSCAT.ФБ.Генераторы импульсов

A\_TRIG (OSCAT)

B\_TRIG (OSCAT)

CLICK\_CNT (OSCAT)

CLICK\_DEC (OSCAT)

CLK\_DIV (OSCAT)

CLK\_N (OSCAT)  
CLK\_PRG (OSCAT)  
CLK\_PULSE (OSCAT)  
CYCLE\_4 (OSCAT)  
D\_TRIG (OSCAT)  
GEN\_BIT (OSCAT)  
GEN\_SQ (OSCAT)  
SCHEDULER (OSCAT)  
SCHEDULER\_2 (OSCAT)  
SEQUENCE\_4 (OSCAT)  
SEQUENCE\_64 (OSCAT)  
SEQUENCE\_8 (OSCAT)  
TMAX (OSCAT)  
TMIN (OSCAT)  
TOF\_1 (OSCAT)  
TONOF (OSCAT)  
TP\_1 (OSCAT)  
TP\_1D (OSCAT)  
TP\_X (OSCAT)

OSCAT.ФБ.Логические модули

BYTE\_TO\_BITS (OSCAT)  
DEC\_2 (OSCAT)  
DEC\_4 (OSCAT)  
DEC\_8 (OSCAT)  
MATRIX (OSCAT)

OSCAT.ФБ.Триггеры

COUNT\_BR (OSCAT)  
COUNT\_BR\_plcd  
COUNT\_DR (OSCAT)

COUNT\_DR\_plcd

FF\_D2E (OSCAT)

FF\_D4E (OSCAT)

FF\_DRE (OSCAT)

FF\_JKE (OSCAT)

FF\_RSE (OSCAT)

LTCH (OSCAT)

LATCH4 (OSCAT)

SELECT\_8 (OSCAT)

SHR\_4E (OSCAT)

SHR\_4UDE (OSCAT)

SHR\_8PLE (OSCAT)

SHR\_8UDE (OSCAT)

STORE\_8 (OSCAT)

TOGGLE (OSCAT)

OSCAT.ФБ.Генераторы сигналов

\_RMP\_B (OSCAT)

\_RMP\_NEXT (OSCAT)

\_RMP\_W (OSCAT)

GEN\_PULSE (OSCAT)

GEN\_PW2 (OSCAT)

GEN\_RDM (OSCAT)

GEN\_RDT (OSCAT)

GEN\_RMP (OSCAT)

GEN\_SIN (OSCAT)

GEN\_SQR (OSCAT)

PWM\_DC (OSCAT)

PWM\_PW (OSCAT)

RMP\_B (OSCAT)

RMP\_SOFT (OSCAT)

RMP\_W (OSCAT)

OSCAT.ФБ.Обработка сигналов

AIN1 (OSCAT)

DELAY (OSCAT)

DELAY\_4 (OSCAT)

FADE (OSCAT)

FILTER\_DW (OSCAT)

FILTER\_I (OSCAT)

FILTER\_MAV\_DW (OSCAT)

FILTER\_MAV\_W (OSCAT)

FILTER\_W (OSCAT)

FILTER\_WAV (OSCAT)

SEL2\_OF\_3 (OSCAT)

SEL2\_OF\_3B (OSCAT)

SH (OSCAT)

SH\_1 (OSCAT)

SH\_2 (OSCAT)

SH\_T (OSCAT)

STAIR2 (OSCAT)

TREND (OSCAT)

TREND\_DW (OSCAT)

OSCAT.ФБ.Модули измерения

ALARM\_2 (OSCAT)

BAR\_GRAPH (OSCAT)

CALIBRATE (OSCAT)

CALIBRATE\_mplc

CYCLE\_TIME (OSCAT)

DT\_SIMU (OSCAT)

FLOW\_METER (OSCAT)

M\_D (OSCAT)

M\_T (OSCAT)

M\_TX (OSCAT)

ONTIME (OSCAT)

TC\_MS (OSCAT)

TC\_S (OSCAT)

TC\_US (OSCAT)

METER\_STAT (OSCAT)

OSCAT.ФБ.Преобразования

ASTRO (OSCAT)

ENERGY (OSCAT)

LENGTH (OSCAT)

PRESSURE (OSCAT)

SPEED (OSCAT)

TEMPERATURE (OSCAT)

OSCAT.ФБ.Модули регулирования

CONTROL\_SET1 (OSCAT)

CONTROL\_SET2 (OSCAT)

CTRL\_OUT (OSCAT)

CTRL\_PI (OSCAT)

CTRL\_PID (OSCAT)

CTRL\_PWM (OSCAT)

DEAD\_BAND\_A (OSCAT)

DEAD\_ZONE2 (OSCAT)

FT\_DERIV (OSCAT)

FT\_IMP (OSCAT)

FT\_INT (OSCAT)

FT\_PD (OSCAT)



FT\_PDT1 (OSCAT)  
FT\_PI (OSCAT)  
FT\_PID (OSCAT)  
FT\_PIDW (OSCAT)  
FT\_PIDWL (OSCAT)  
FT\_PIW (OSCAT)  
FT\_PIWL (OSCAT)  
FT\_PT1 (OSCAT)  
FT\_PT2 (OSCAT)  
FT\_TN16 (OSCAT)  
FT\_TN64 (OSCAT)  
FT\_TN8 (OSCAT)  
HYST (OSCAT)  
HYST\_1 (OSCAT)  
HYST\_2 (OSCAT)  
HYST\_3 (OSCAT)  
INTEGRATE (OSCAT)

OSCAT.ФБ.Управление устройствами

DRIVER\_1 (OSCAT)  
DRIVER\_4 (OSCAT)  
DRIVER\_4C (OSCAT)  
FLOW\_CONTROL (OSCAT)  
FT\_PROFILE (OSCAT)  
INC\_DEC (OSCAT)  
INTERLOCK\_4 (OSCAT)  
INTERLOCK (OSCAT)  
MANUAL\_1 (OSCAT)  
MANUAL\_2 (OSCAT)  
MANUAL\_4 (OSCAT)  
PARSET (OSCAT)

PARSET2 (OSCAT)

SIGNAL (OSCAT)

SIGNAL\_4 (OSCAT)

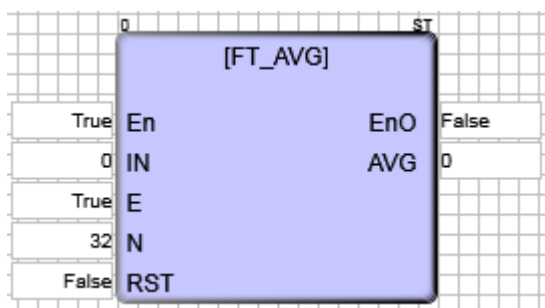
SRAMP (OSCAT)

TUNE (OSCAT)

TUNE2 (OSCAT)

## 10.2.1. OSCAT.ФБ.Арифметические функции

### 10.2.1.1. FT\_AVG (OSCAT)



Входы ФБ:

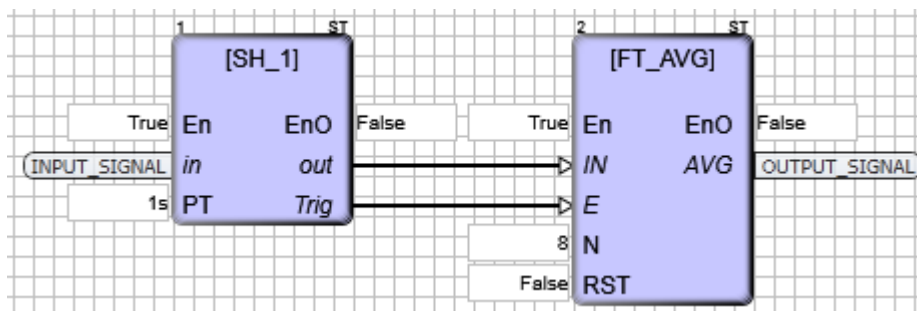
- IN (REAL) – входной сигнал;
- E (BOOL) – разрешение (TRUE) или запрет (FALSE) вычислений;
- N (UINT) – число значений в буфере;
- RST (BOOL) – сброс.

Выходы ФБ:

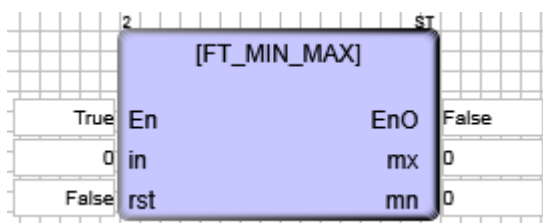
- AVG (REAL) – выходной сигнал.

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:



### 10.2.1.2. FT\_MIN\_MAX (OSCAT)



Входы ФБ:

- IN (REAL) – входной сигнал;
- RST (BOOL) – сброс.

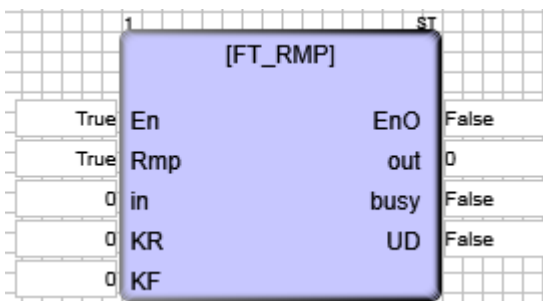
Выходы ФБ:

- MX (REAL) – максимальное значение входного сигнала;
- MN (REAL) – минимальное значение входного сигнала.

ФБ вычисляет максимальное и минимальное значения входного сигнала с момента RST=FALSE.

При RST=TRUE MX=MN=IN.

### 10.2.1.3. 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:

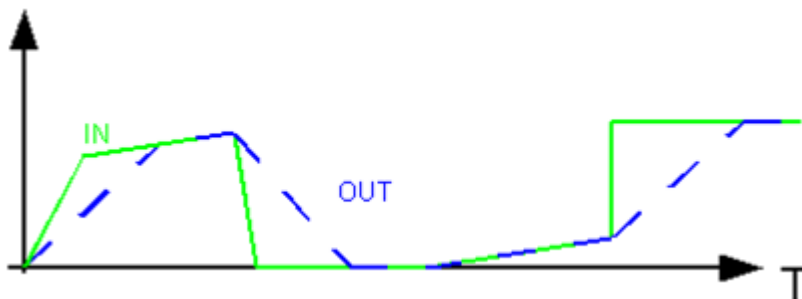
- OUT (REAL) – выходной сигнал;
- BUSY (BOOL) – индикатор процесса сглаживания;
- UD (BOOL) – индикатор нарастания/убывания при сглаживании.

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.

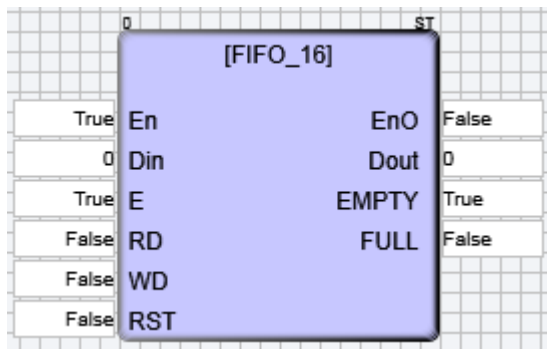
Строго говоря, сглаживание работает в том случае, если  $T_{цикла} < |IN_n - OUT_{n-1}| / K$  (K равно KR при нарастании и KF при убывании), в противном случае OUT:=IN. Для упрощения понимания можно использовать более мягкое условие начала сглаживания: скорость возрастания (убывания) IN должна быть больше KR (KF), где под скоростью возрастания/убывания IN понимается отношение изменения IN за 1 цикл к длительности цикла.

Пример работы ФБ:



## 10.2.2. OSCAT.ФБ.Модули памяти

### 10.2.2.1. 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.

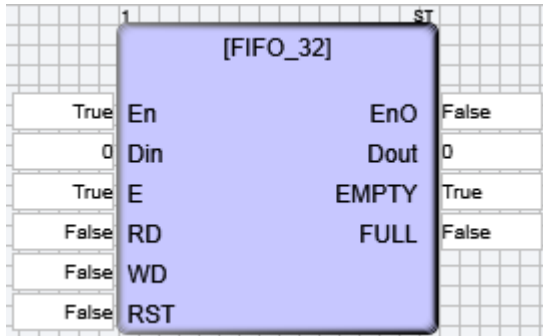
Если данные помещены во все свободные позиции стека, FULL=TRUE, и помещение новых значений в стек невозможно. EMPTY=TRUE индицирует, что стек пуст (все позиции свободны, и невозможно извлечение значений из стека).

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).

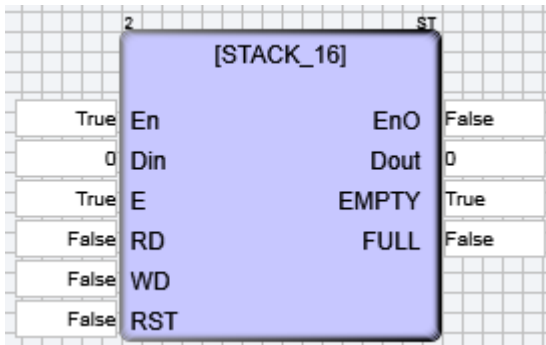
### 10.2.2.2. FIFO\_32 (OSCAT)



**FIFO\_32** – это аналог FIFO\_16 (OSCAT), но стек содержит 32 позиции.

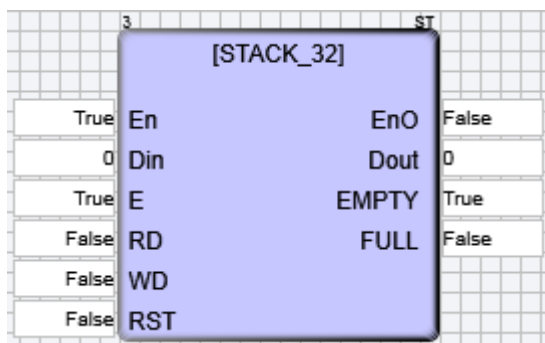
Используемые функции: INC1 (OSCAT).

### 10.2.2.3. 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).

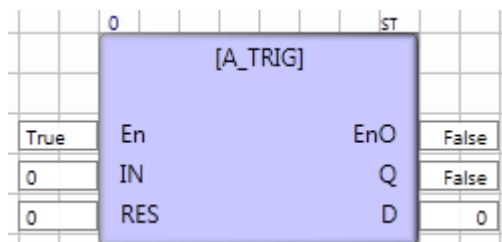
### 10.2.2.4. STACK\_32 (OSCAT)



**STACK\_32** – это аналог **STACK\_16 (OSCAT)**, но стек содержит 32 позиции.

### 10.2.3. OSCAT.ФБ.Генераторы импульсов

#### 10.2.3.1. A\_TRIG (OSCAT)



Входы ФБ:

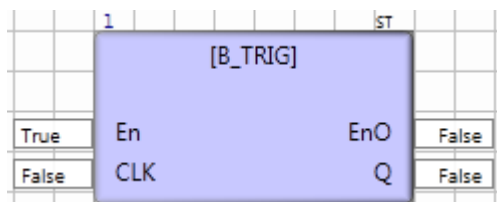
- IN (тип данных REAL) – входной сигнал;
- RES (тип данных REAL) – порог.

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**.

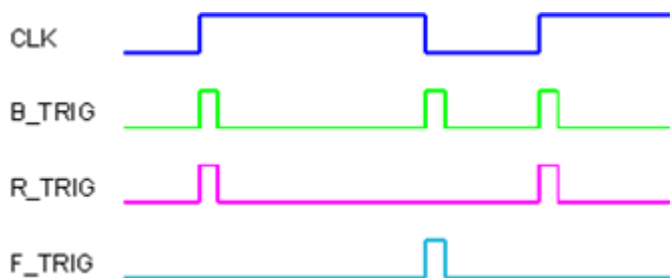
#### 10.2.3.2. B\_TRIG (OSCAT)



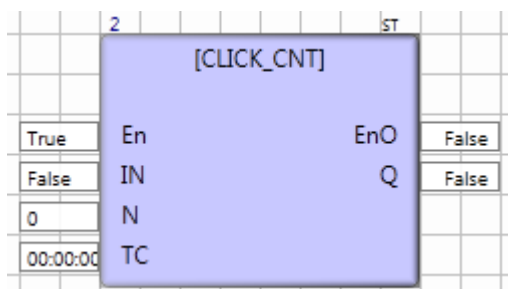
Тип данных входа и выхода – **BOOL**.

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:



### 10.2.3.3. 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.

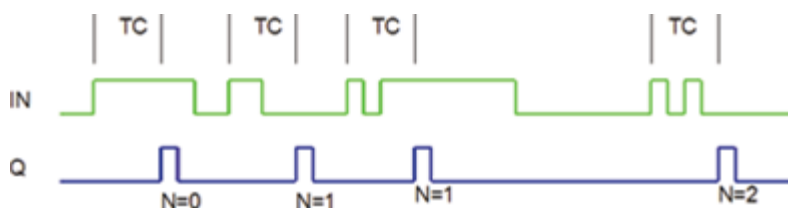
ФБ содержит внутренний таймер, который запускается по первому переднему фронту в сигнале IN и останавливается по истечении времени TC, затем по первому переднему фронту в IN вновь запускается на время TC и т.д.

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.

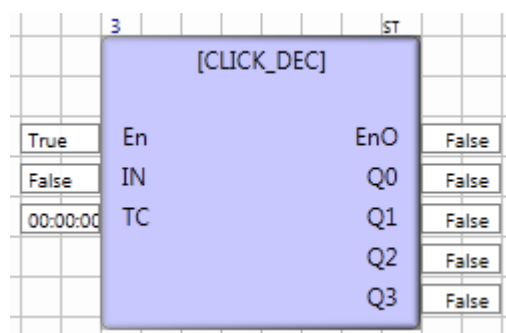


При  $N=0$  CLICK\_CNT генерирует на выходе Q одиночный импульс длиной 1 цикл, если в течение одного периода работы таймера во входном сигнале прошел только один передний фронт (который, собственно, и запустил таймер).

Следующая диаграмма иллюстрирует алгоритм ФБ:



#### 10.2.3.4. 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 TC;
- Q1 (тип данных BOOL) – выходной сигнал в случае 1 импульса в IN в течение TC;
- Q2 (тип данных BOOL) – выходной сигнал в случае 2 импульсов в IN в течение TC;
- Q3 (тип данных BOOL) – выходной сигнал в случае 3 импульсов в IN в течение TC.

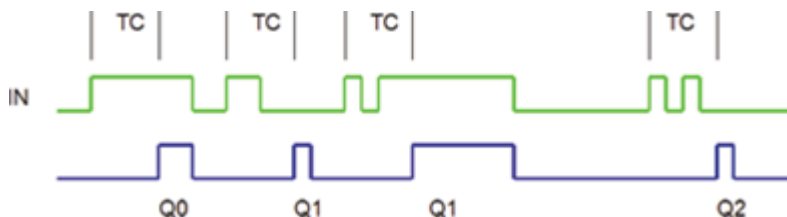
ФБ содержит внутренний таймер, который запускается по первому переднему фронту в сигнале IN и останавливается по истечении времени TC, затем по первому переднему фронту в IN вновь запускается на время TC и т.д.

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.

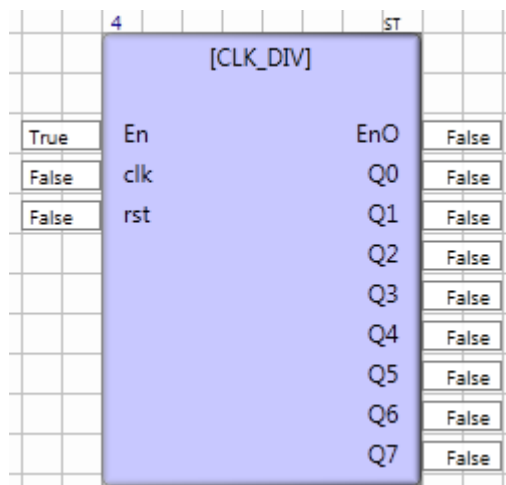
Если в течение одного периода работы таймера во входном сигнале IN прошло  $(n+1)$  передних и  $n$  задних фронтов (т.е.  $n=1,2,3$  полных импульсов и передний фронт), выход  $Q<n>$  принимает значение 1 и остается равным 1 до тех пор, пока IN не примет значение FALSE.

Следующая диаграмма иллюстрирует алгоритм ФБ:



CLICK\_DEC может использоваться для детектирования повторяющегося нажатия клавиш.

### 10.2.3.5. 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:

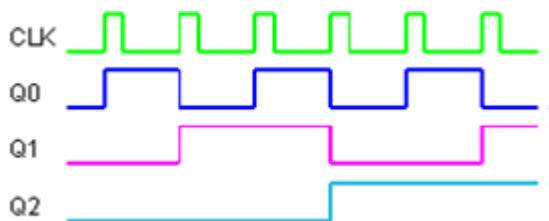
- Q0 (тип данных BOOL) – меандр с частотой F/2;

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.

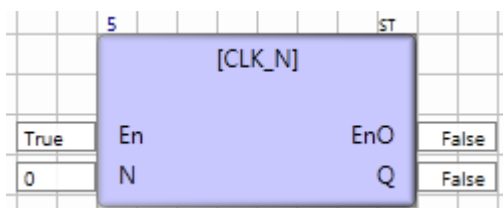
- Q1 (тип данных BOOL) – меандр с частотой F/4;
- Q2 (тип данных BOOL) – меандр с частотой F/8;
- Q3 (тип данных BOOL) – меандр с частотой F/16;
- Q4 (тип данных BOOL) – меандр с частотой F/32;
- Q5 (тип данных BOOL) – меандр с частотой F/64;
- Q6 (тип данных BOOL) – меандр с частотой F/128;
- Q7 (тип данных BOOL) – меандр с частотой F/256.

ФБ используется как делитель частоты.

Временная диаграмма ФБ показана на рисунке:



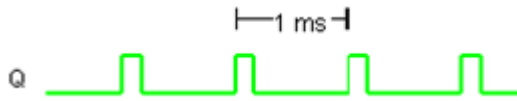
### 10.2.3.6. CLK\_N (OSCAT)



Тип данных входа N – INT, выхода Q – BOOL.

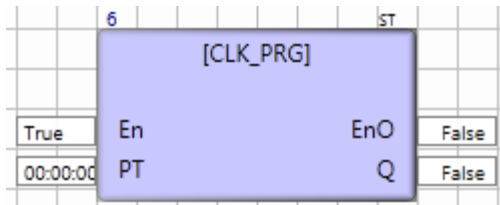
С помощью SPS-таймера ПЛК (разрешение этого таймера – 1 мс) ФБ генерирует импульс шириной ровно 1 цикл каждые  $2N$  миллисекунд (т.е. при  $N=0$  период следования импульсов равен 1мс, при  $N=1$  – 2мс, при  $N=2$  – 4мс и т.д.).

Следующая диаграмма иллюстрирует работу ФБ при  $N=0$ :



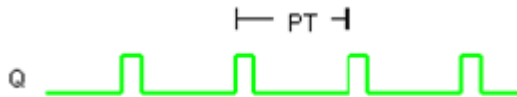
Используемые функции: T\_PLC\_MS (OSCAT).

### 10.2.3.7. CLK\_PRG (OSCAT)



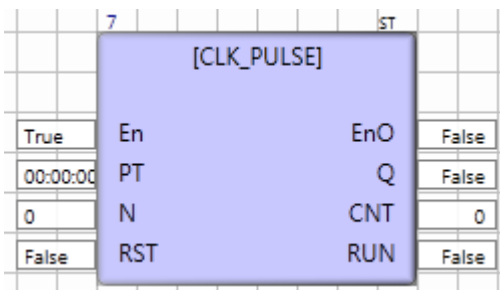
Тип данных входа PT – TIME, выхода Q – BOOL.

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).

### 10.2.3.8. CLK\_PULSE (OSCAT)



Входы ФБ:

- PT (тип данных TIME) – период следования генерируемых импульсов;

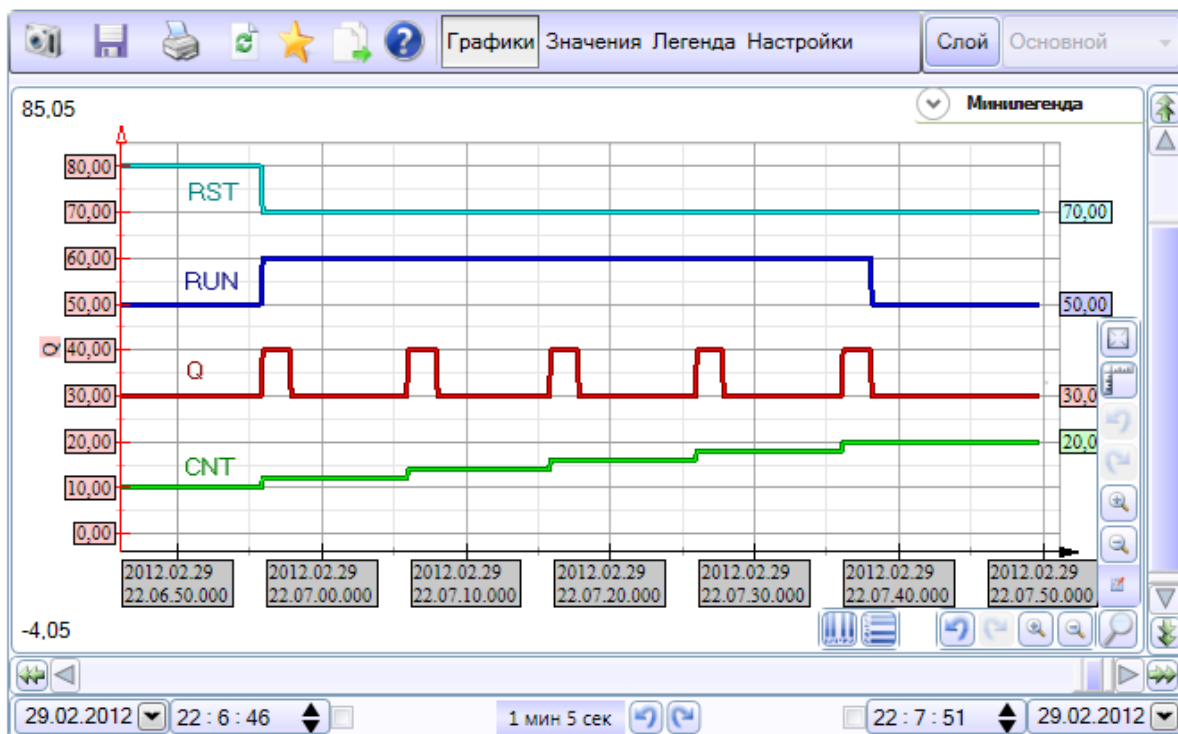
- N (тип данных INT) – число генерируемых импульсов (1-32767). Если N=0, генерируется бесконечная последовательность импульсов;
- RST (тип данных BOOL) – сброс.

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).

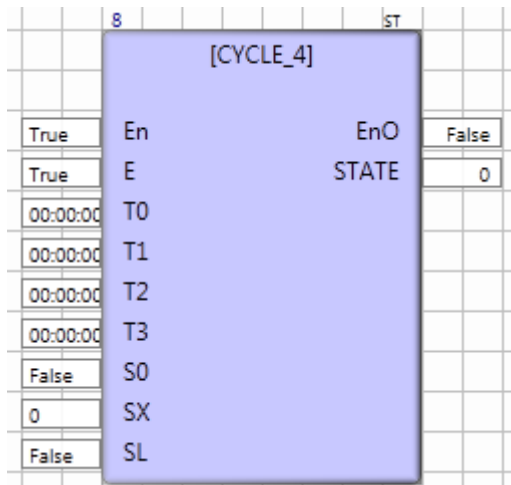
CLK\_PULSE генерирует на выходе Q заданное число импульсов шириной 1 цикл с заданным периодом следования PT. Генерация начинается при RST=FALSE (первый импульс генерируется в том же цикле, в котором RST принимает значение FALSE). Для отмены генерации нужно присвоить RST значение TRUE (по этой команде 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).

### 10.2.3.9. 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).

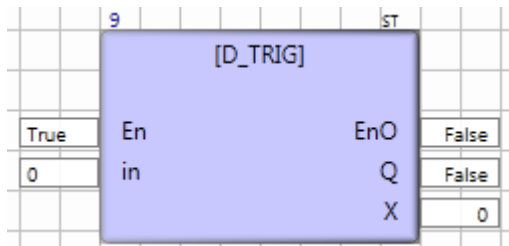
Если SL=FALSE и SO=FALSE, ФБ проходит все состояния от начального до 3, и на этом переходы прекращаются.

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).

### 10.2.3.10. D\_TRIG (OSCAT)



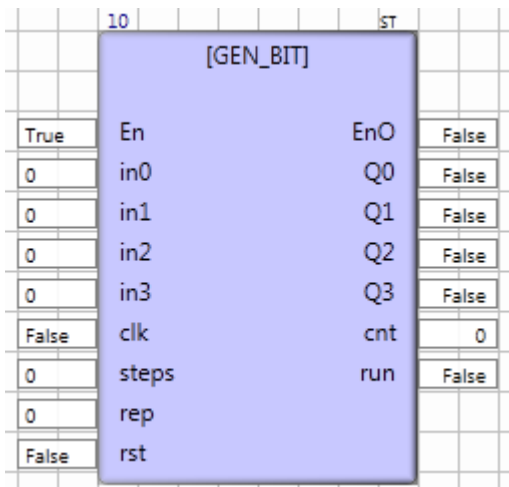
Тип данных входа IN – DWORD, выхода Q – BOOL, выхода X – DWORD.

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.

### 10.2.3.11. 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.

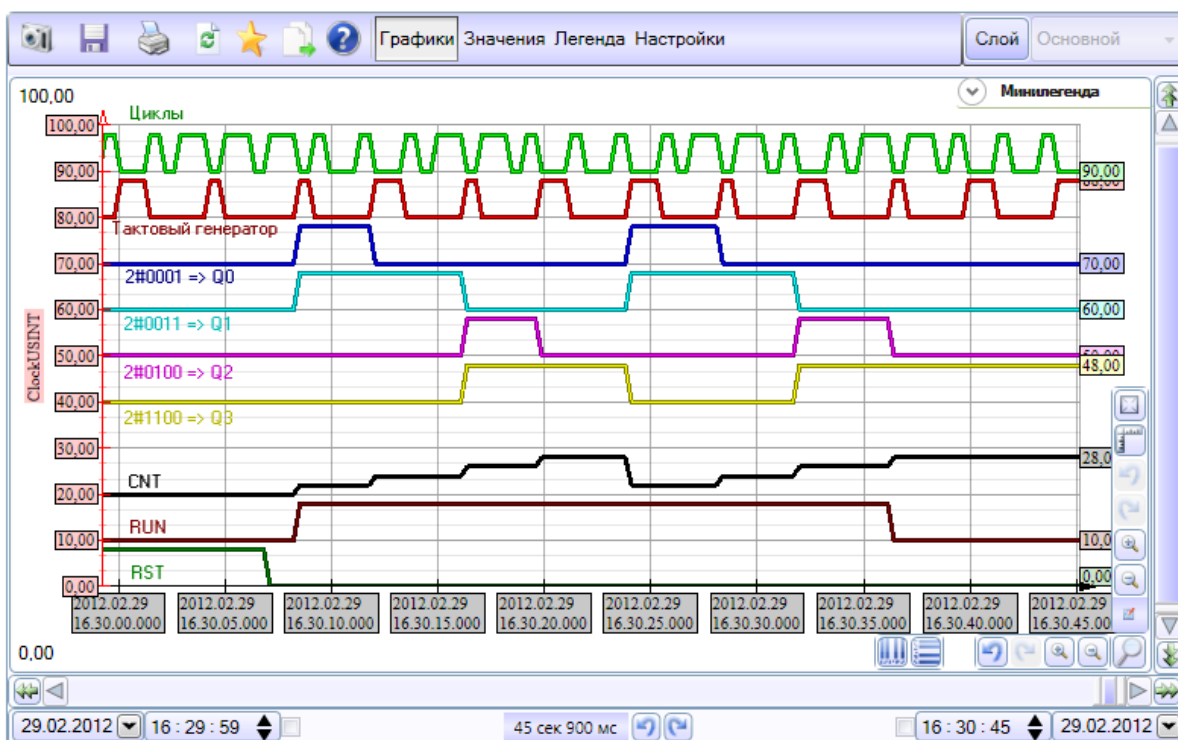
Вход REP определяет число повторений образцов в генерируемых последовательностях. Если REP=0, образцы повторяются бесконечно.



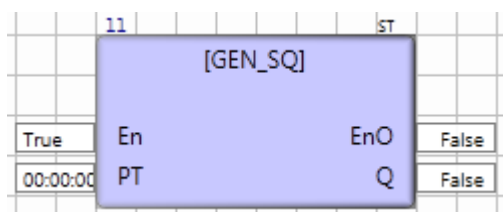
После окончания генерации, как и после каждого переданного бита, выходы не обнуляются и сохраняют последние принятые значения. Это, в частности, приводит к тому, что по единичным битам, идущим в образце подряд, генерируется один импульс соответствующей ширины.

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):

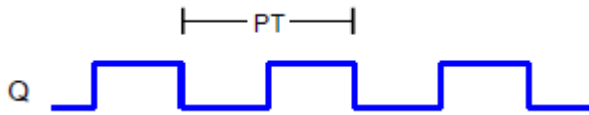


### 10.2.3.12. GEN\_SQ (OSCAT)



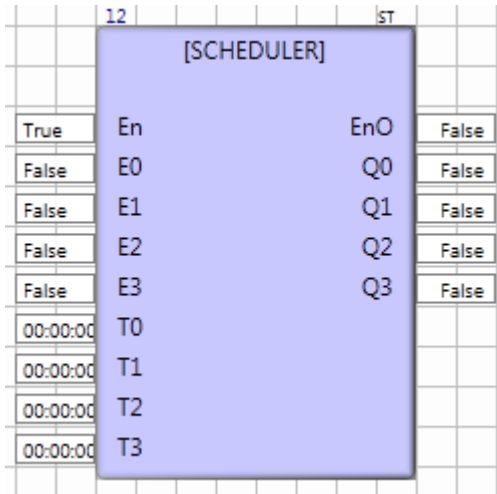
Тип данных входа PT – TIME, выхода Q – BOOL.

GEN\_SQ генерирует на выходе Q меандр с заданным периодом PT:



Используемые функции: T\_PLC\_MS (OSCAT).

### 10.2.3.13. SCHEDULER (OSCAT)

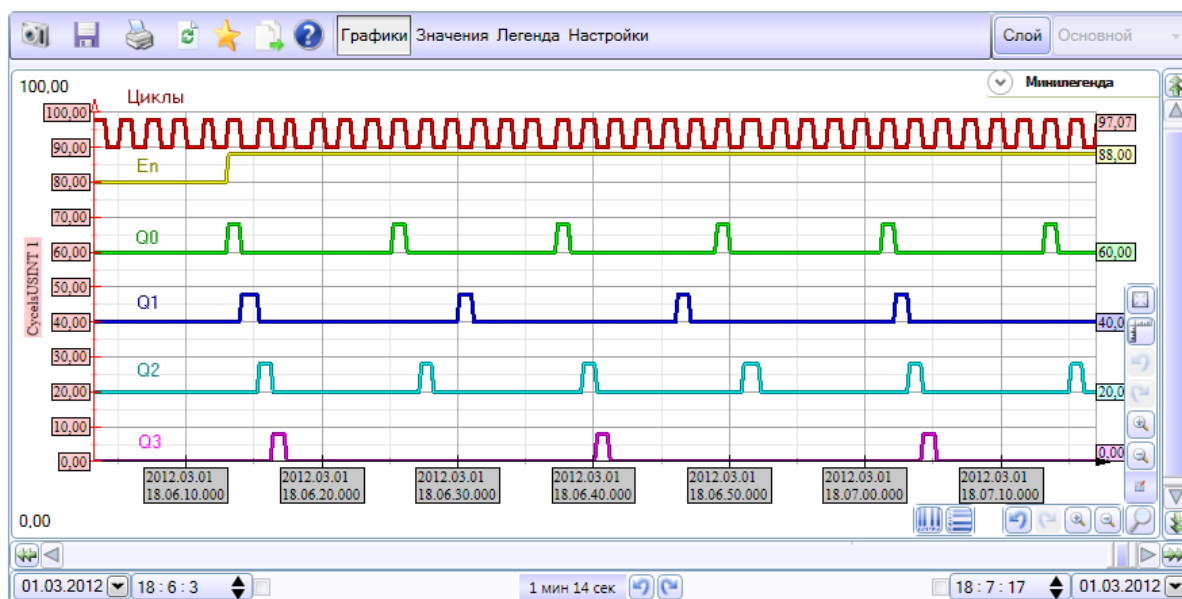


Тип данных входов E0 .. E3 и соответствующих им выходов Q0 .. Q3 – BOOL, тип данных входов T0 .. T3 – TIME.

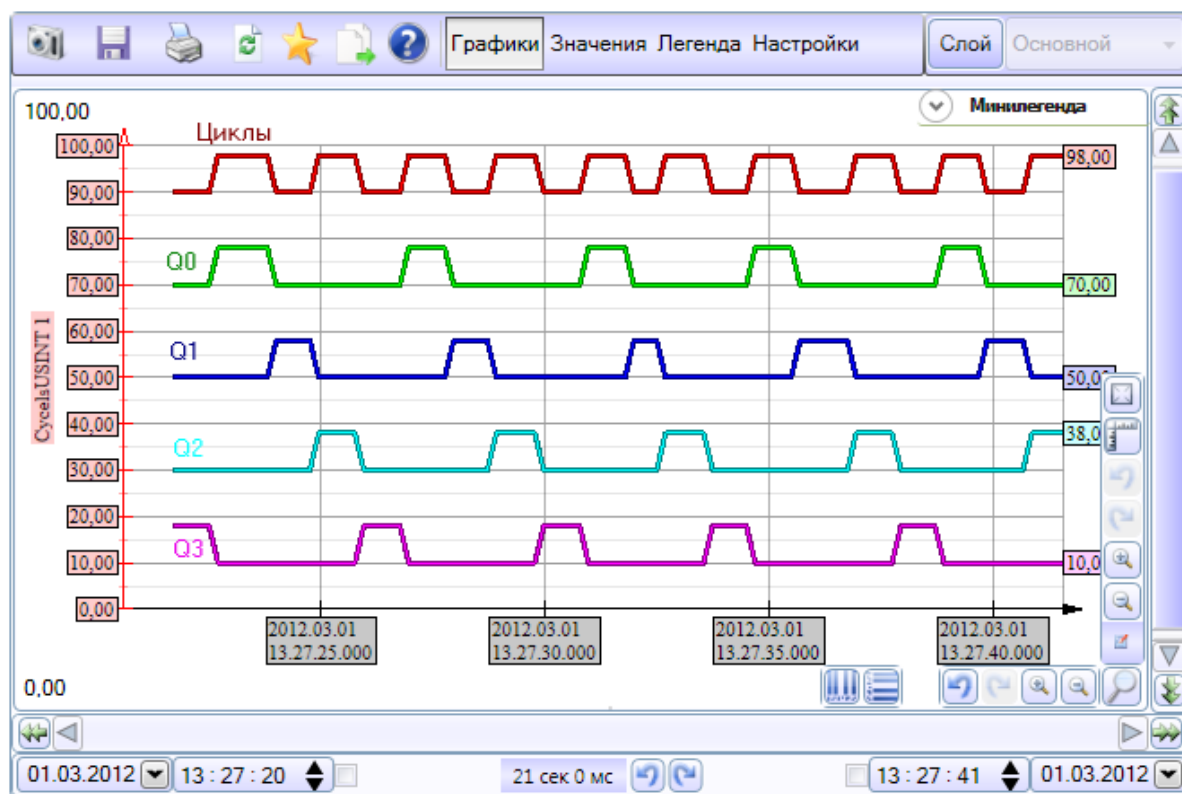
ФБ SCHEDULER, предназначенный для вызова программ по расписанию, работает по следующему алгоритму:

- 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 En=1) при E0=E1=E2=E3=1, T0=10с, T1=15с, T2=10с и T3=20с (длительность цикла – 1с):

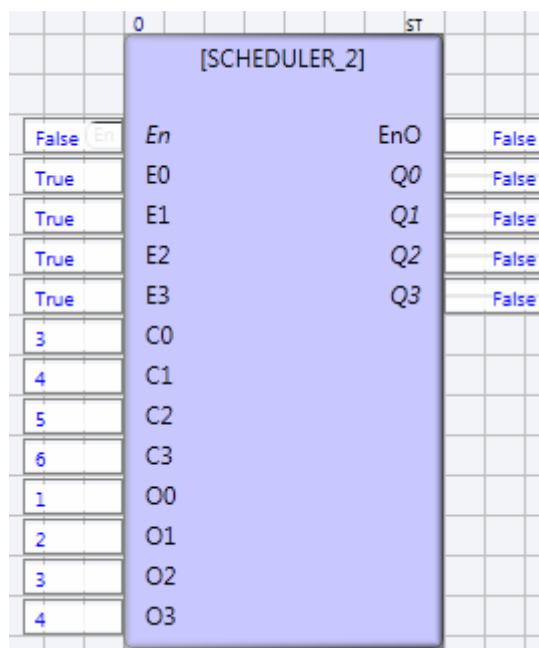


Условие равенства единице только одного выхода в течение цикла может приводить к увеличению  $T_{<n>}$ . Например, при длительности цикла 1с и всех  $T_{<n>} = 0$  реальный период составит 4с:



Используемые функции: T\_PLC\_MS (OSCAT).

### 10.2.3.14. SCHEDULER\_2 (OSCAT)

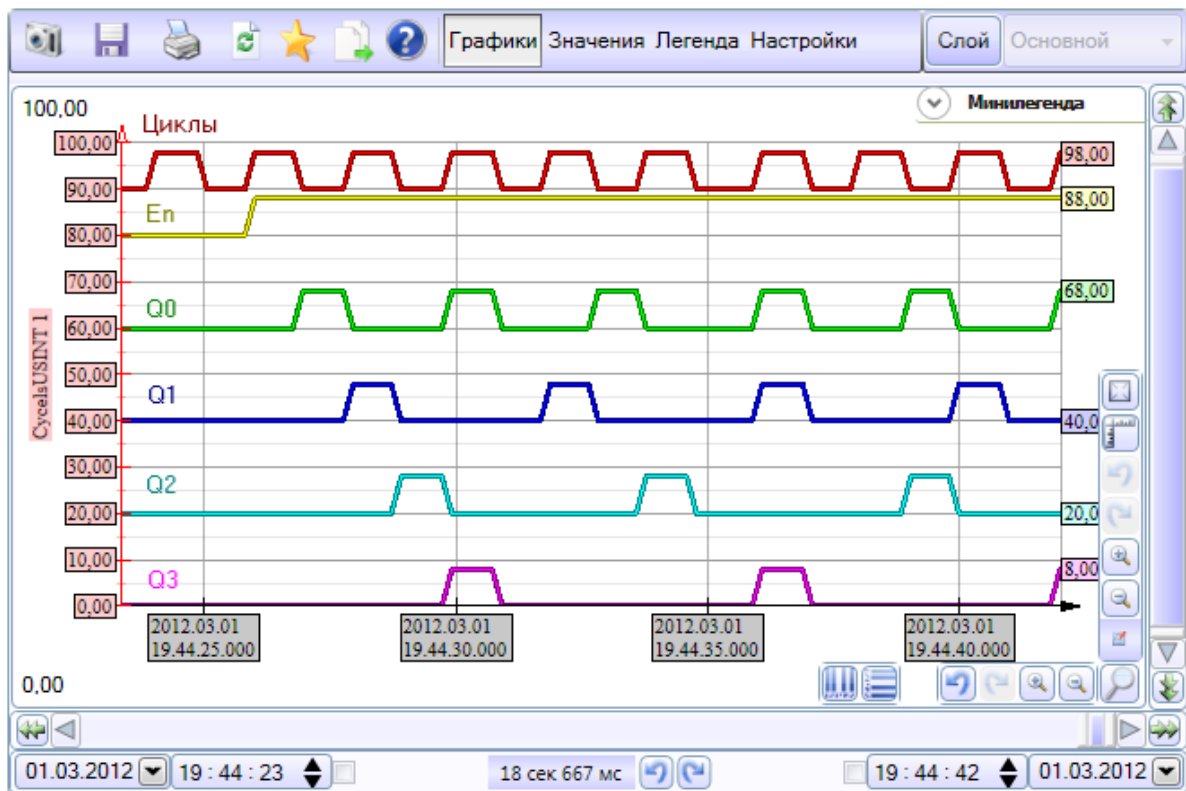


Тип данных входов E0 .. E3 и соответствующих им выходов Q0 .. Q3 – BOOL, тип данных входов C0 .. C3 и O0 .. O3 – UINT.

SCHEDULER\_2 используется для вызова программ на определенных циклах. При E<n> = 1 этот ФБ активирует 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 En=TRUE на выходах формируются следующие сигналы:



### 10.2.3.15. SEQUENCE\_4 (OSCAT)

0	[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:

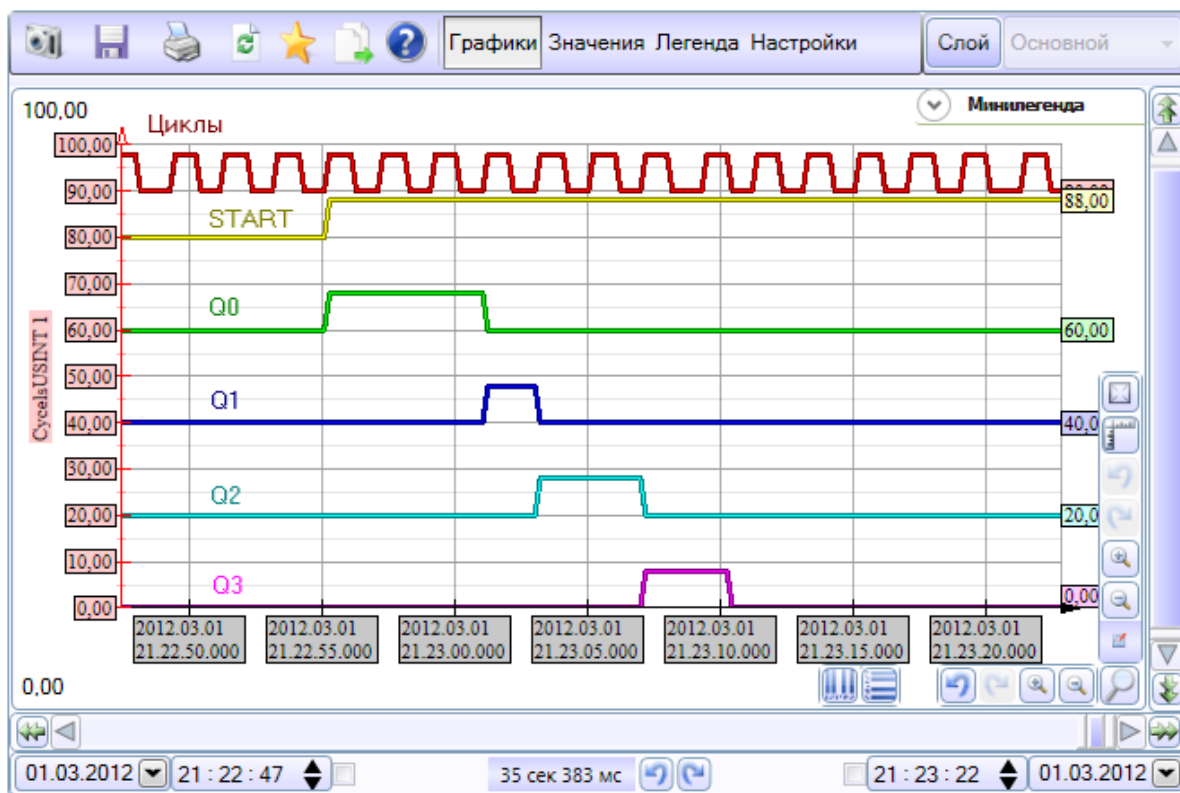
- если  $STOP\_ON\_ERROR=TRUE$ , алгоритм ФБ завершает работу ( $RUN=FALSE$ ), выход  $STEPP$  индицирует номер входа с ошибкой, а  $STATUS=STEPP+1$ ;
- если  $STOP\_ON\_ERROR=FALSE$ , алгоритм ФБ продолжает работу (так же, как если бы 1 на соответствующем входе детектирована в самом конце интервала ожидания, но без генерации импульса на  $Q<n>$ ).

Выход  $STATUS$  индицирует 110, если алгоритм остановлен, и 111, если алгоритм запущен.

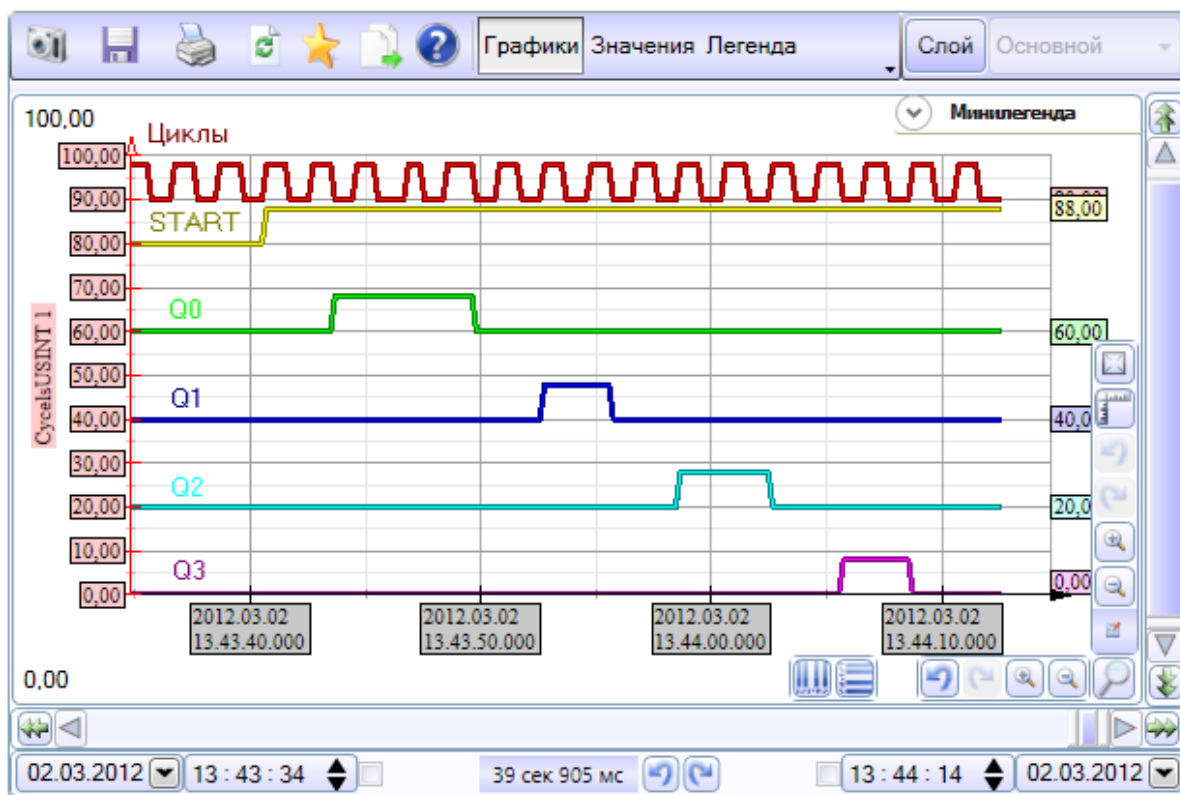
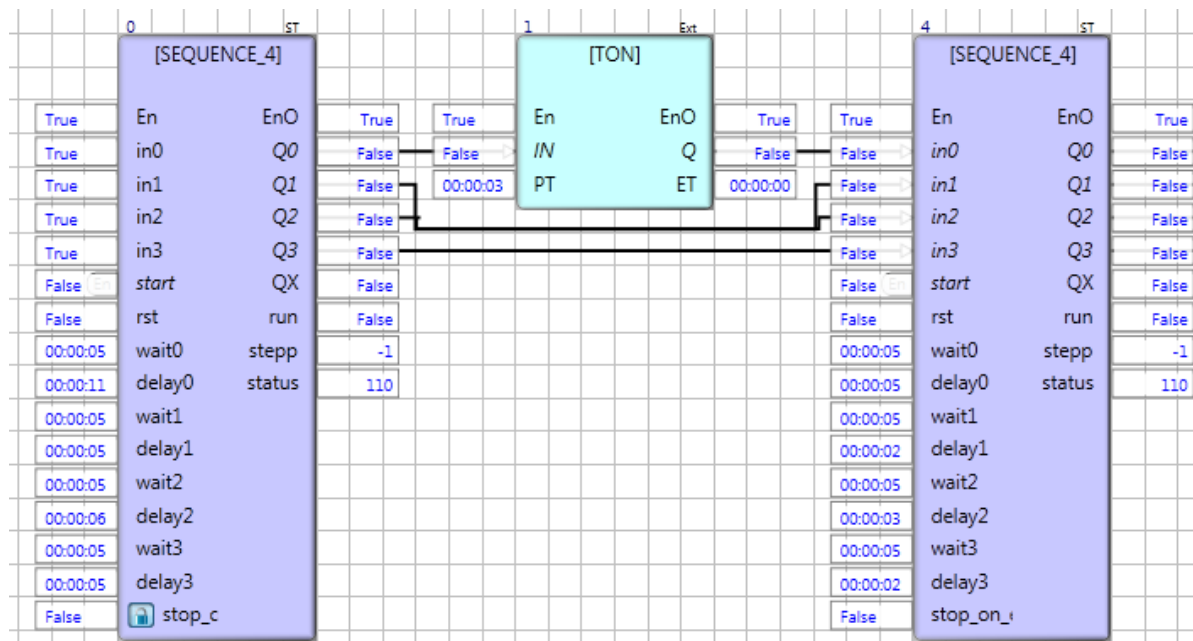
После последнего шага, на котором генерируется импульс на  $Q3$ , ФБ возвращается в исходное состояние.

Используемые функции:  $T\_PLC\_MS$  (OSCAT).

При конфигурации ФБ, показанной на рисунке выше, после  $START=TRUE$  на выходах формируются следующие сигналы (длительность цикла – 1с):

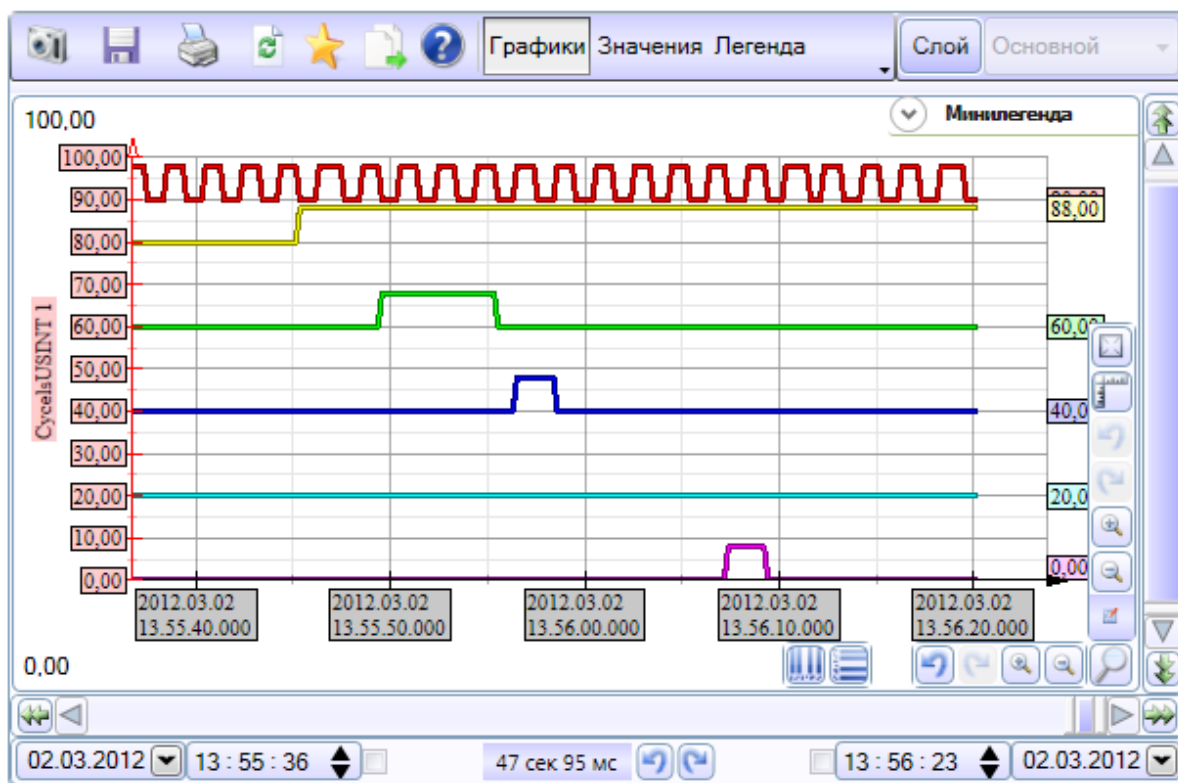


В следующей программе левый ФБ  $SEQUENCE\_4$  и ФБ  $TON$ , задержка включения имитируют задержку в 3с появления 1 на каждом входе правого ФБ  $SEQUENCE\_4$  относительно заднего фронта импульса на предыдущем выходе:

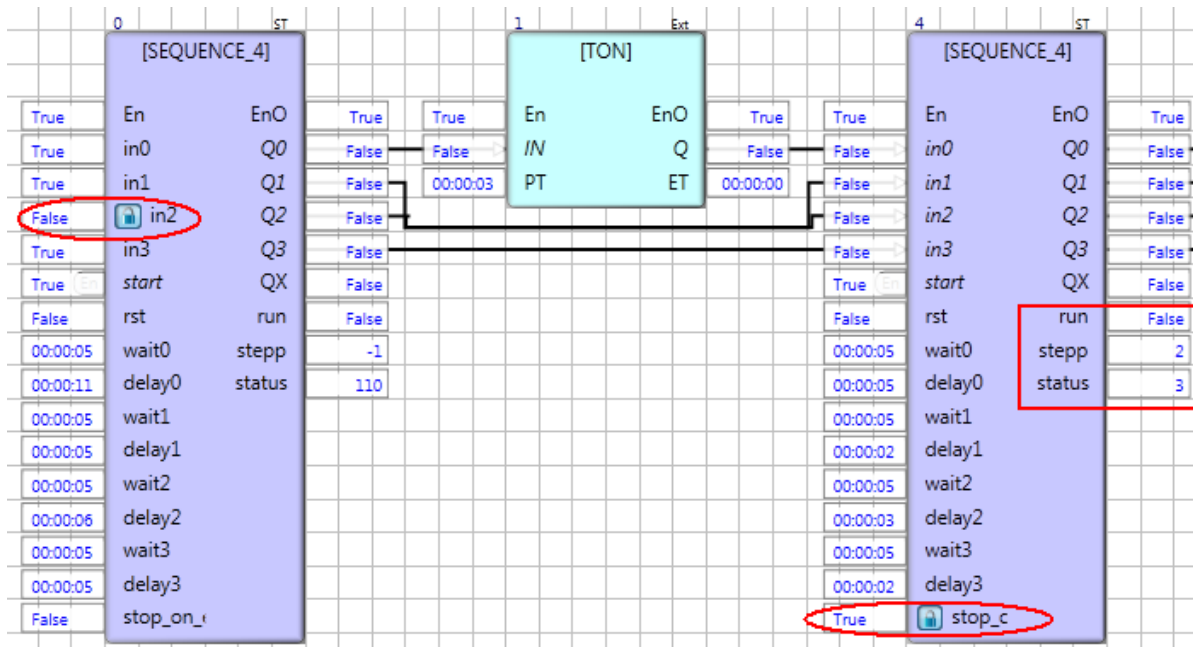
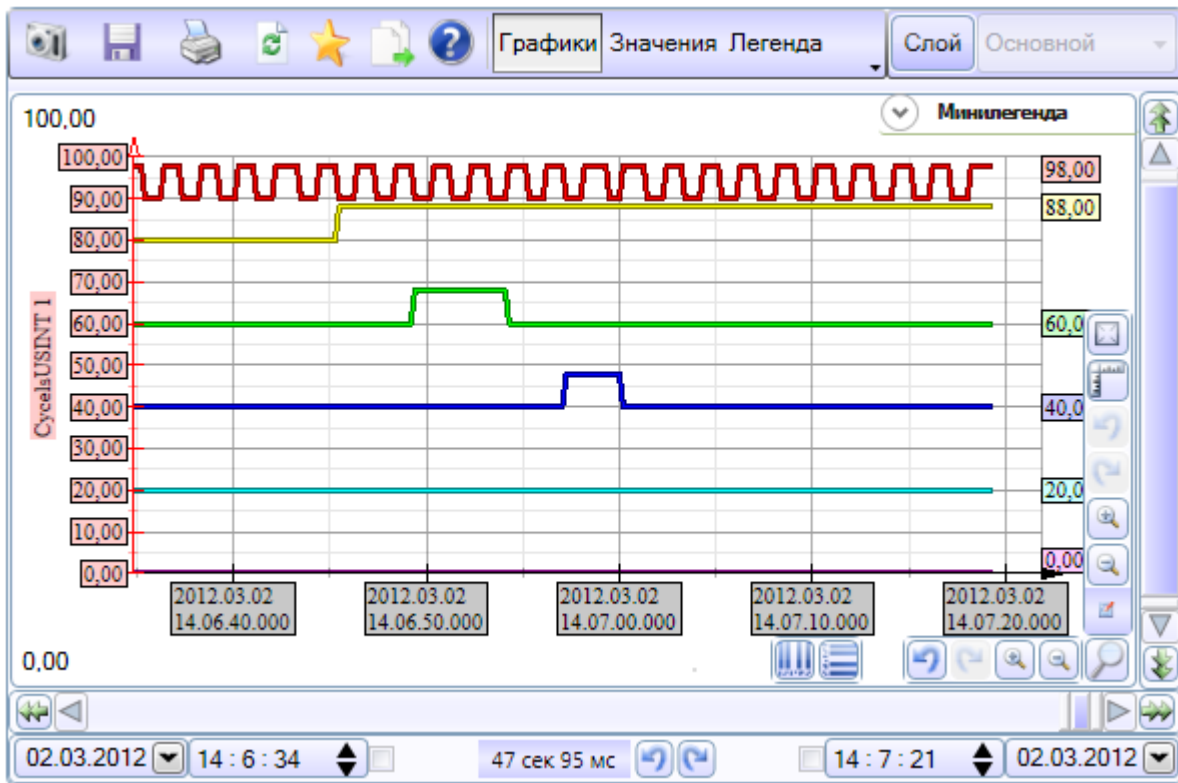


Следующая диаграмма демонстрирует игнорирование отсутствия 1 на IN2 (IN2 левого ФБ SEQUENCE\_4 – FALSE, STOP\_ON\_ERROR обоих ФБ – FALSE):

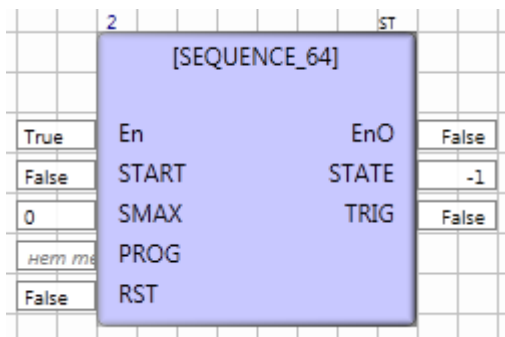




Следующая диаграмма демонстрирует останов генерации при отсутствии 1 на IN2 (IN2 левого ФБ SEQUENCE\_4 – FALSE, STOP\_ON\_ERROR правого ФБ SEQUENCE\_4 – TRUE):



### 10.2.3.16. 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.

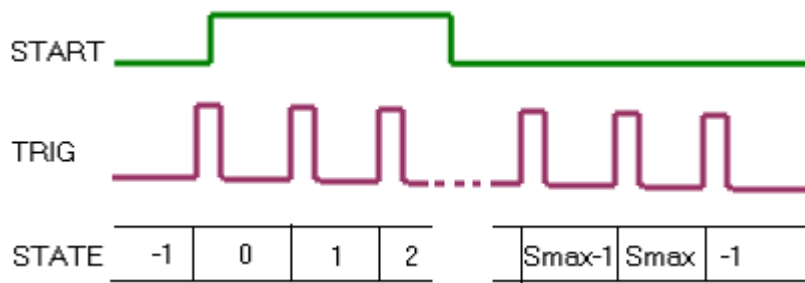
SEQUENCE\_64 предназначен для последовательного прохождения состояний 1..N (NMAX=64). В последнем состоянии STATE = -1 ("спящее" состояние ФБ, т.е. алгоритм остановлен, но готов к старту).

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:

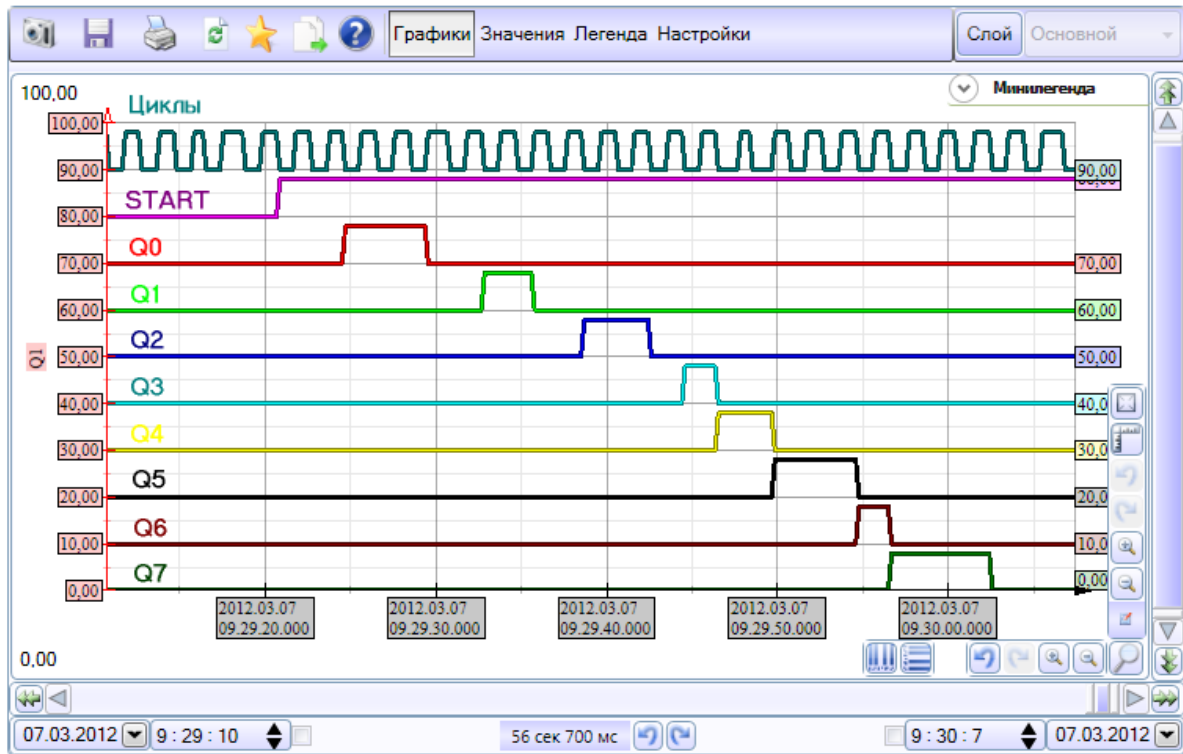


Используемые функции: T\_PLC\_MS (OSCAT), INC2 (OSCAT).

### 10.2.3.17. 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			

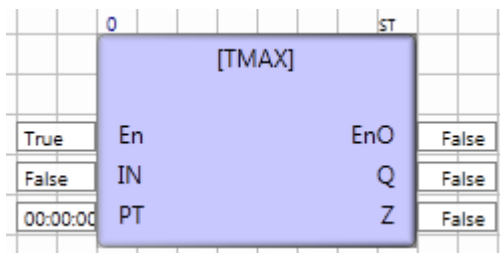
Данный ФБ представляет собой 8-канальный аналог ФБ SEQUENCE\_4 (OSCAT):



Используемые функции: T\_PLC\_MS (OSCAT).

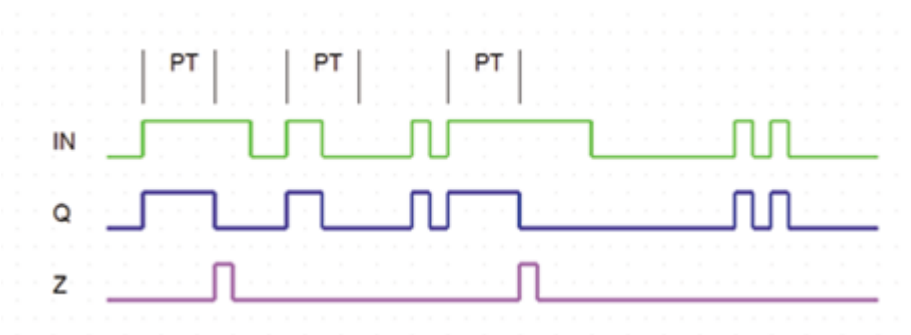
SEQUENCE\_8 используется в ФБ **Legionella** (OSCAT).

### 10.2.3.18. TMAX (OSCAT)



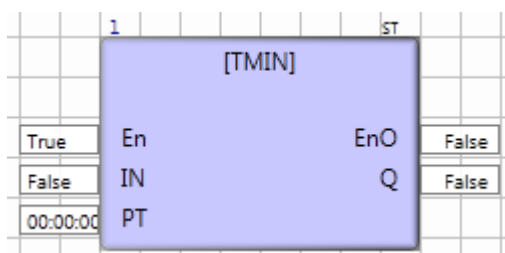
Тип данных входа PT – TIME, входа IN и выходов Q и Z – BOOL.

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).

### 10.2.3.19. TMIN (OSCAT)

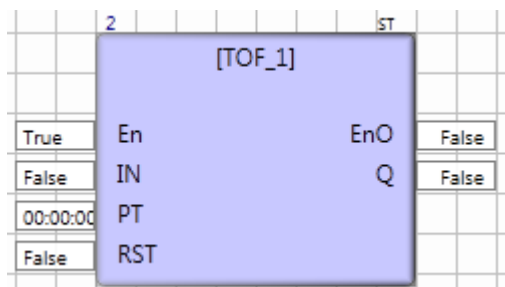


Тип данных входа PT – TIME, входов IN и выхода Q – BOOL.

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.

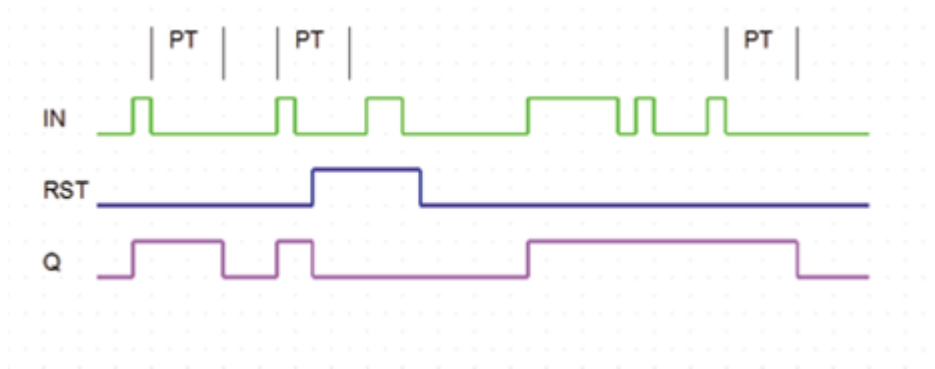


### 10.2.3.20. TOF\_1 (OSCAT)



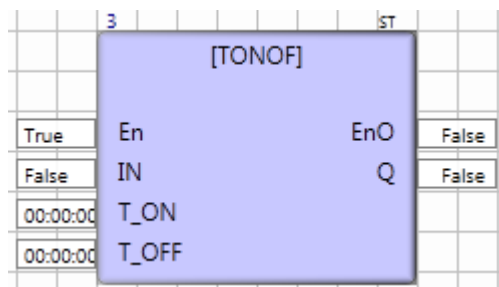
Тип данных входа PT – TIME, входов IN и RST и выхода Q – BOOL.

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).

### 10.2.3.21. TONOF (OSCAT)



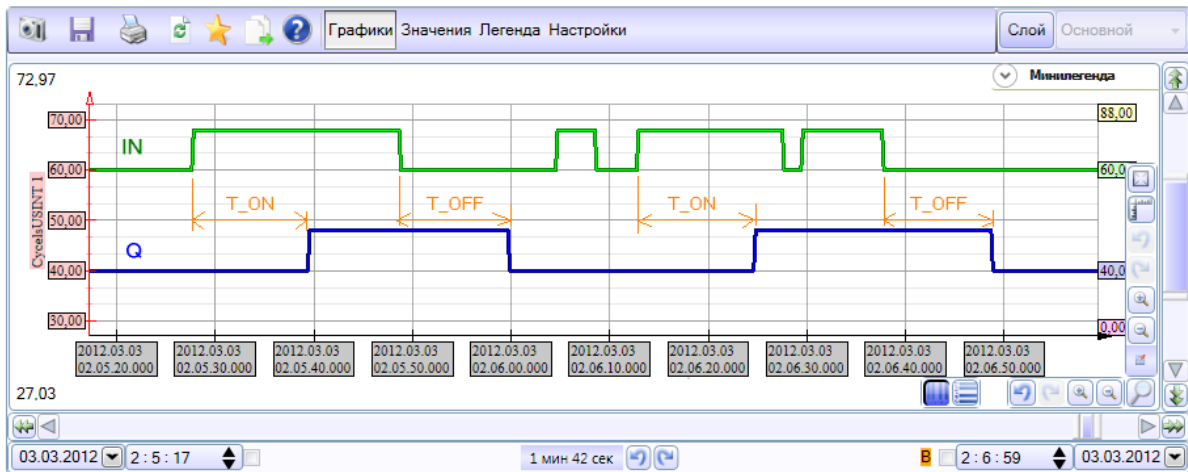
Тип данных входов T\_ON и T\_OFF – TIME, входа IN и выхода Q – BOOL.

TONOF задерживает включение (передний фронт на IN) на T\_ON и выключение (задний фронт) – на T\_OFF.

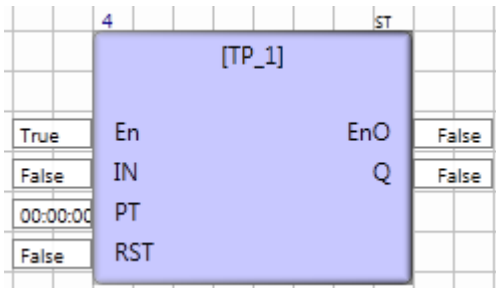
Для корректной работы ФБ ширина импульса включения должна быть больше T\_ON, а после заднего фронта выключения не должно быть импульсов.

На рисунке ниже T\_ON=T\_OFF=10с:





### 10.2.3.22. TP\_1 (OSCAT)

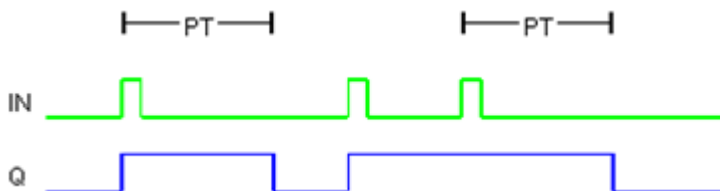


Тип данных входа PT – TIME, входов IN и RST и выхода Q – BOOL.

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).

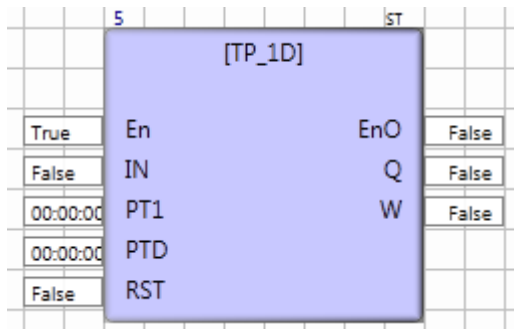
Для останова алгоритма нужно присвоить TRUE входу RST (выход Q при этом обнуляется).

Временная диаграмма ФБ:



Используемые функции: T\_PLC\_MS (OSCAT).

### 10.2.3.23. TP\_1D (OSCAT)



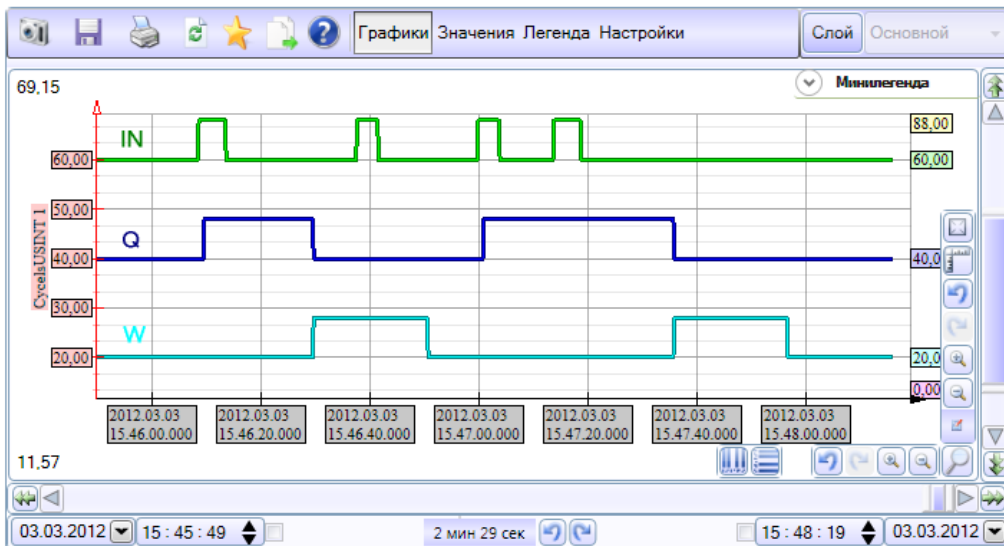
Тип данных входов IN и RST и выходов Q и W – BOOL, входов PT1 и PTD – TIME.

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.

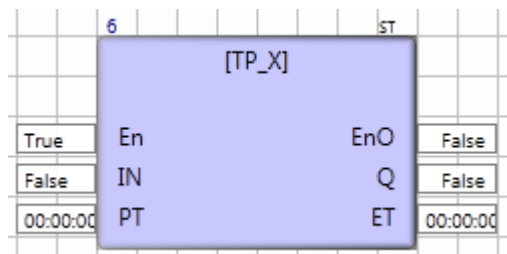
Важно! Данный ФБ модифицирован, в нем отсутствует автоматический сброс IN и RST.

Временная диаграмма ФБ:



Используемые функции: T\_PLC\_MS (OSCAT).

### 10.2.3.24. TP\_X (OSCAT)

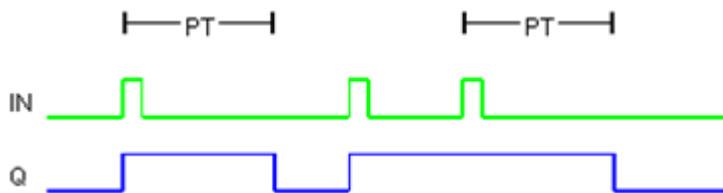


Тип данных входа IN и выхода Q – BOOL, входа PT и выхода ET – TIME.

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.

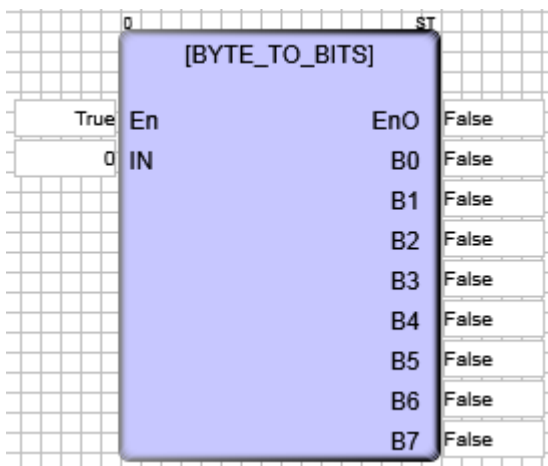
Временная диаграмма ФБ:



Используемые функции: T\_PLC\_MS (OSCAT).

## 10.2.4. OSCAT.ФБ.Логические модули

### 10.2.4.1. BYTE\_TO\_BITS (OSCAT)

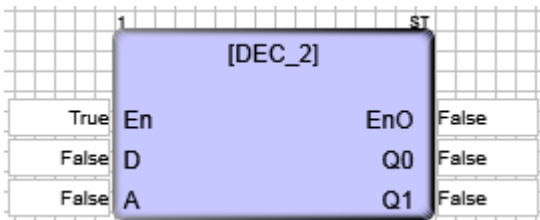


Тип данных входа IN – BYTE, выходов B0..B7 – BOOL.

Выходы B0..B7 индицируют значения битов 0..7 значения IN.

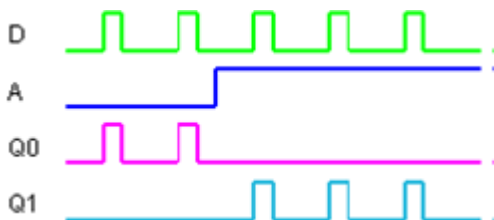
К входу IN может быть привязана переменная DWORD, WORD или BYTE (во всех случаях распаковываются биты 0..7).

### 10.2.4.2. DEC\_2 (OSCAT)

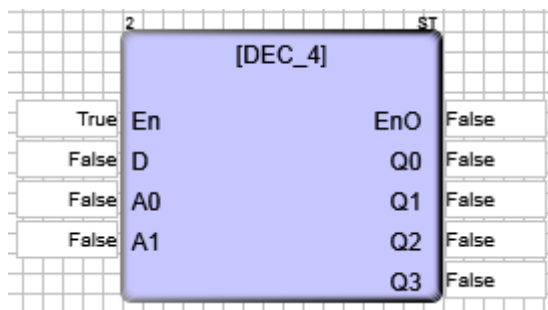


Тип данных входов и выходов – BOOL.

**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).



### 10.2.4.3. DEC\_4 (OSCAT)

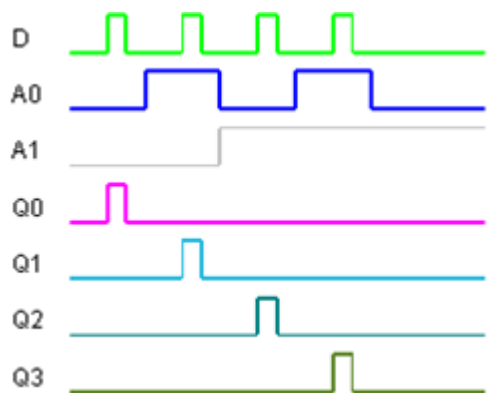


Тип данных входов и выходов – BOOL.

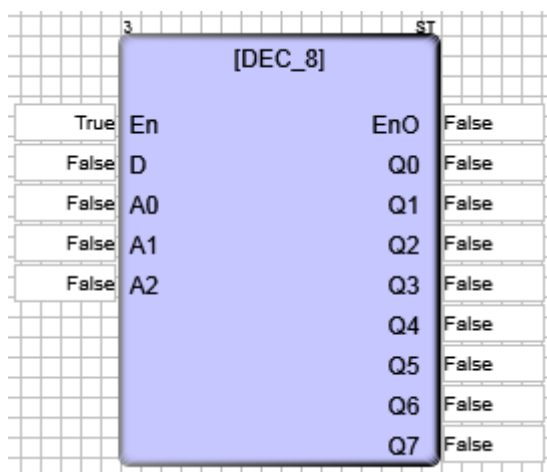
**DEC\_4** – это 4-позиционный переключатель:

- 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.



### 10.2.4.4. DEC\_8 (OSCAT)

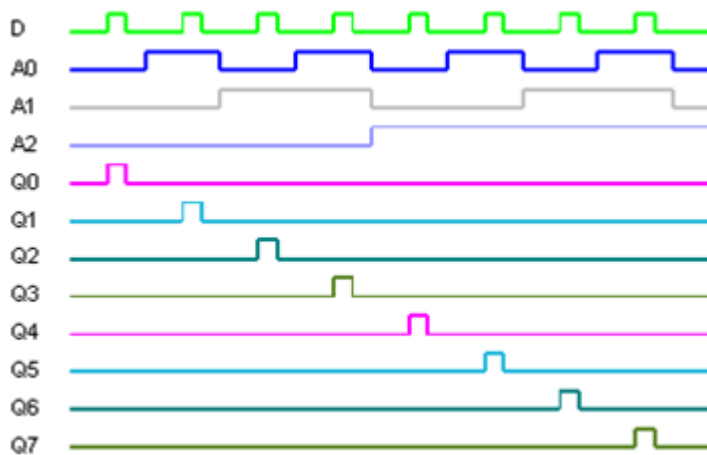


Тип данных входов и выходов – BOOL.

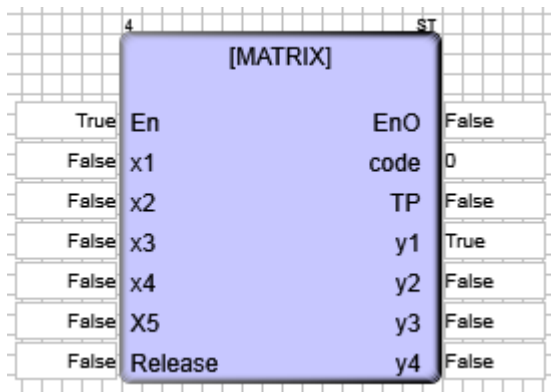
**DEC\_8** – это 8-позиционный переключатель:

- 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.



### 10.2.4.5. MATRIX (OSCAT)



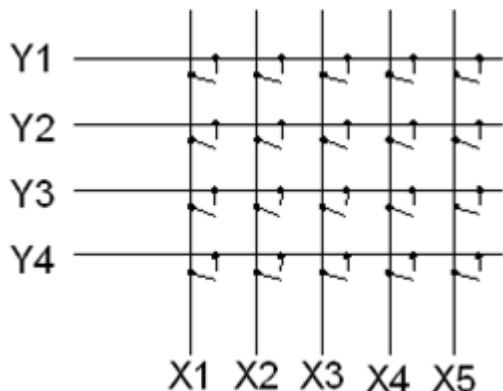
Входы ФБ:

- 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.

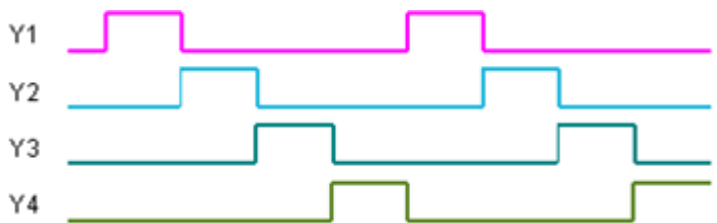
Выходы ФБ:

- 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 (\*\*).

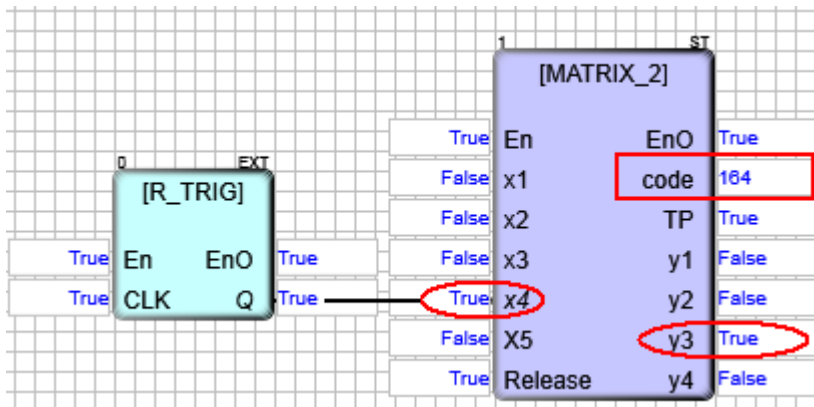


(\*\*) Номер строки **row** принимает значение от 0 до 4. Для строки X1 **row**=0, для X2 **row**=1 и т.д.

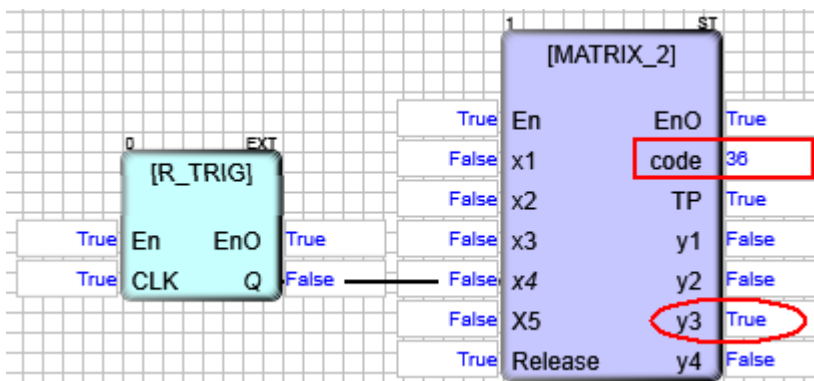
Важно! Текущий код контроллера не анализирует удержание клавиши, т.е. длина импульса от нажатия клавиши не должна превышать длительность цикла, в противном случае будут "считываться" соседние по строке клавиши. При нажатии двух клавиш одной линии корректно считывается клавиша с наименьшим номером строки.

### Пример

Пусть на некотором цикле N при опросе линии 2 (line=2, Y3=1) детектируется X4=1. В этом случае CODE принимает значение 164:

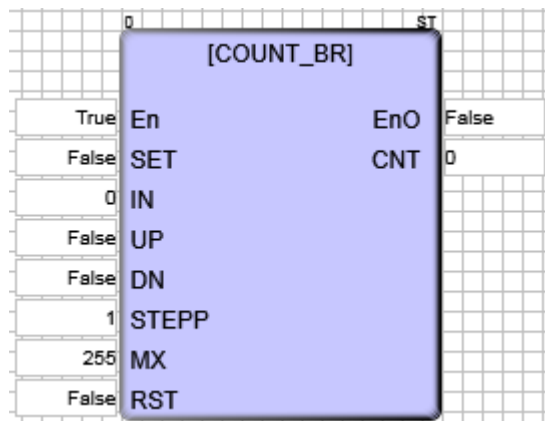


На циклах N+1, N+2 и N+3 CODE=0, а на цикле N+4 CODE=36, поскольку RELEASE=TRUE:



## 10.2.5. OSCAT.ФБ.Триггеры

### 10.2.5.1. 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.

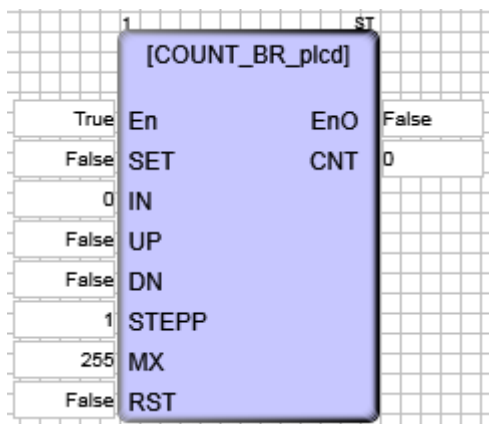
Если RST=TRUE, CNT=0.

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 ...

### 10.2.5.2. 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.

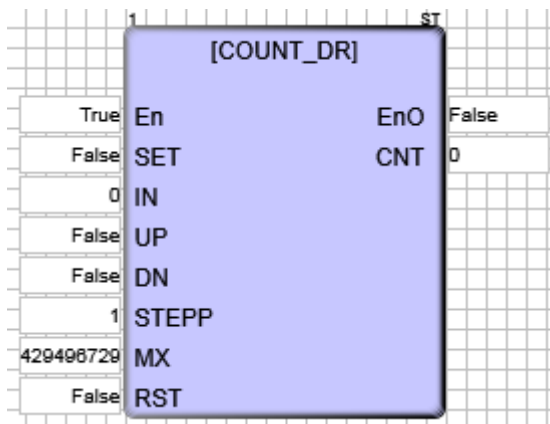
Если SET=TRUE, CNT:=IN (если  $IN \leq MX$ ) или CNT:=MX (если  $IN > MX$ ), и после SET:=FALSE счетчик начинает считать с установленного значения CNT.

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 ...

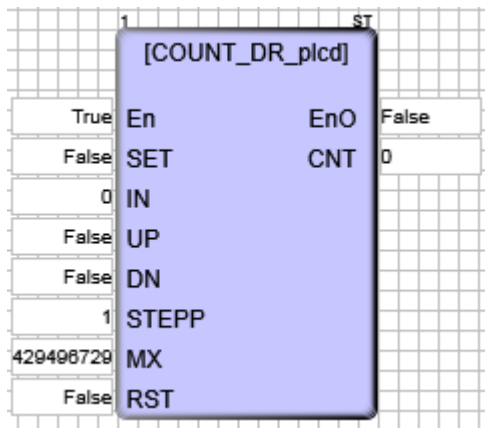
### 10.2.5.3. COUNT\_DR (OSCAT)



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.

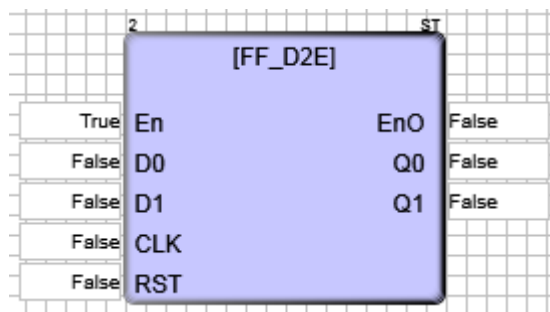
### 10.2.5.4. COUNT\_DR\_plcd



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.

### 10.2.5.5. FF\_D2E (OSCAT)

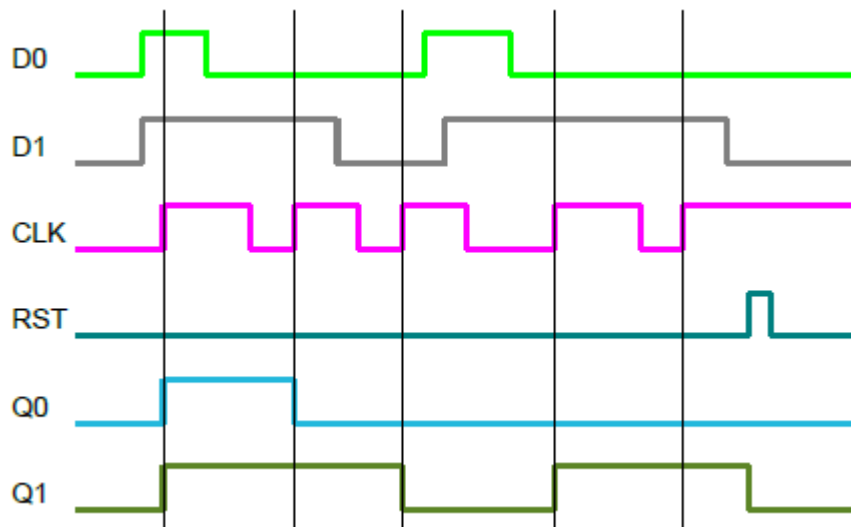


Тип данных входов и выходов – BOOL.

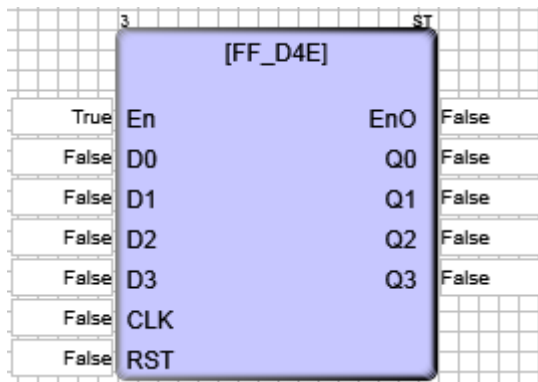
FF\_D2E – это D-триггер.

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.



### 10.2.5.6. FF\_D4E (OSCAT)



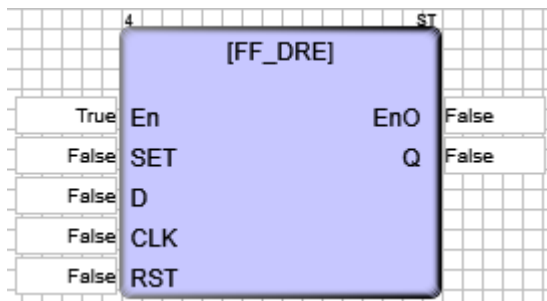
Тип данных входов и выходов – BOOL.

**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.

### 10.2.5.7. FF\_DRE (OSCAT)



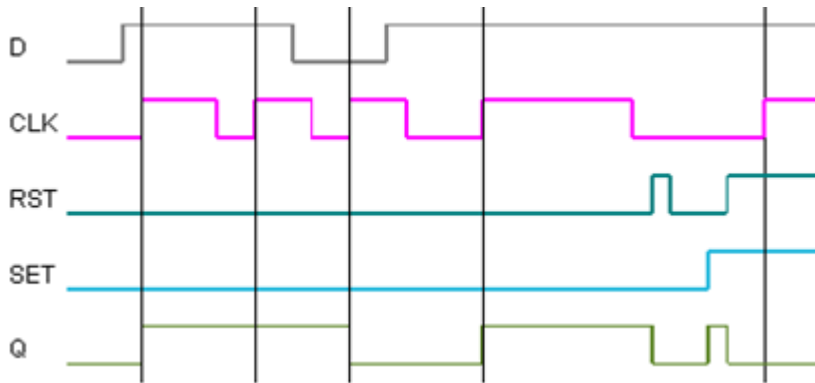
Тип данных входов и выхода – BOOL.

**FF\_DRE** – это D-триггер.

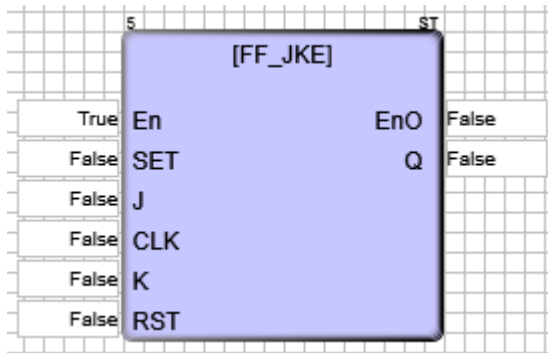
При RST=SET=FALSE и детектировании переднего фронта на входе CLK значение D записывается в Q.

При RST=FALSE и SET=TRUE Q:=TRUE (принудительная установка выхода).

При RST=TRUE Q:=FALSE (принудительный сброс выхода).



### 10.2.5.8. FF\_JKE (OSCAT)



Тип данных входов и выхода – BOOL.

**FF\_JKE** – это JK-триггер.

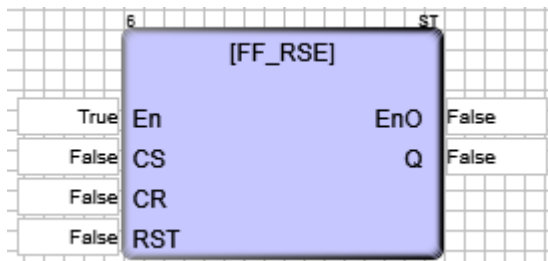
При  $RST=SET=FALSE$ ,  $J \neq K$  и детектировании переднего фронта на входе CLK  $Q:=J$ .

При  $RST=SET=FALSE$ ,  $J=K$  и детектировании переднего фронта на входе CLK  $Q:=Q$  xor  $K$ .

При  $RST=FALSE$  и  $SET=TRUE$   $Q:=TRUE$  (принудительная установка выхода).

При  $RST=TRUE$   $Q:=FALSE$  (принудительный сброс выхода).

### 10.2.5.9. FF\_RSE (OSCAT)



Тип данных входов и выхода – BOOL.

**FF\_RSE** – это RS-триггер.

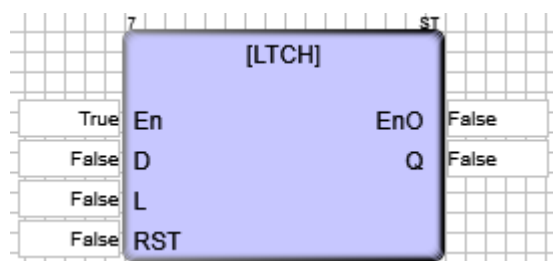
При детектировании переднего фронта на входе CS Q:=TRUE.

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.

### 10.2.5.10. LTCH (OSCAT)

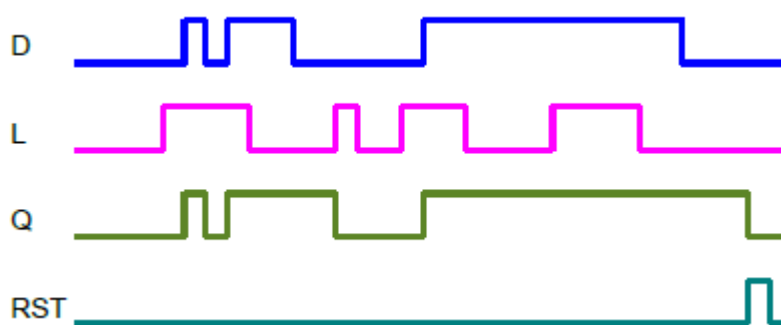


Тип данных входов и выхода – BOOL.

**LTCH** – т.н. прозрачная защелка.

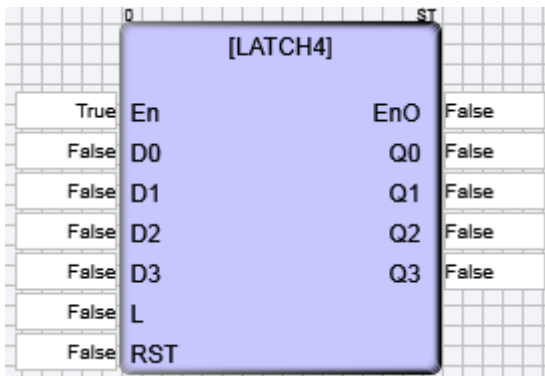
До тех пор, пока L=TRUE, значение D передается в Q. При L=FALSE передача блокируется, а Q сохраняет свое значение.

Если RST=TRUE, Q:=FALSE (принудительный сброс выхода).





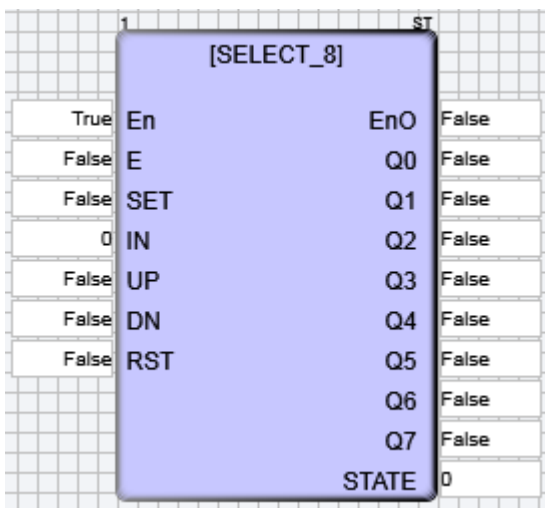
### 10.2.5.11. LATCH4 (OSCAT)



Тип данных входов и выхода – BOOL.

LATCH4 представляет собой 4 ФБ LTCH (OSCAT) (D0 записывается в Q0 и т.д.).

### 10.2.5.12. 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:

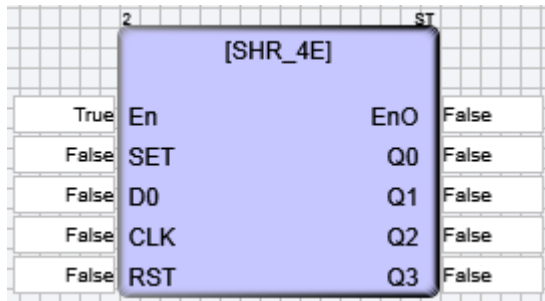
- если RST=SET=FALSE, передний фронт на входе UP устанавливает следующий по номеру выход (STATE:=STATE+1, Q0 после Q7), а передний фронт на входе DN – предыдущий (STATE:=STATE-1, Q7 после Q0); увеличение STATE имеет более высокий приоритет;
- если RST=FALSE, а SET=TRUE, STATE:=IN (установка заданного 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).

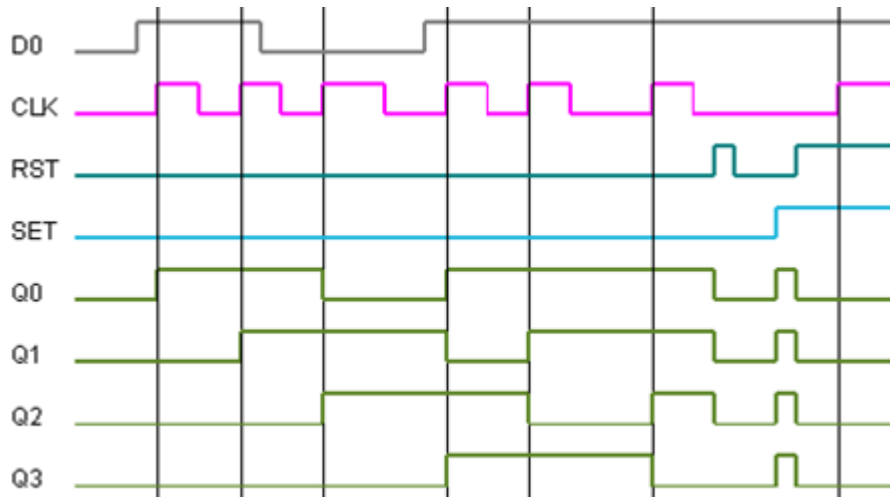
### 10.2.5.13. SHR\_4E (OSCAT)



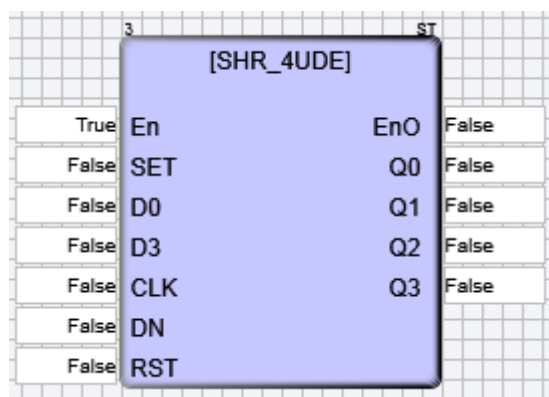
Тип данных входов и выходов – BOOL.

**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.



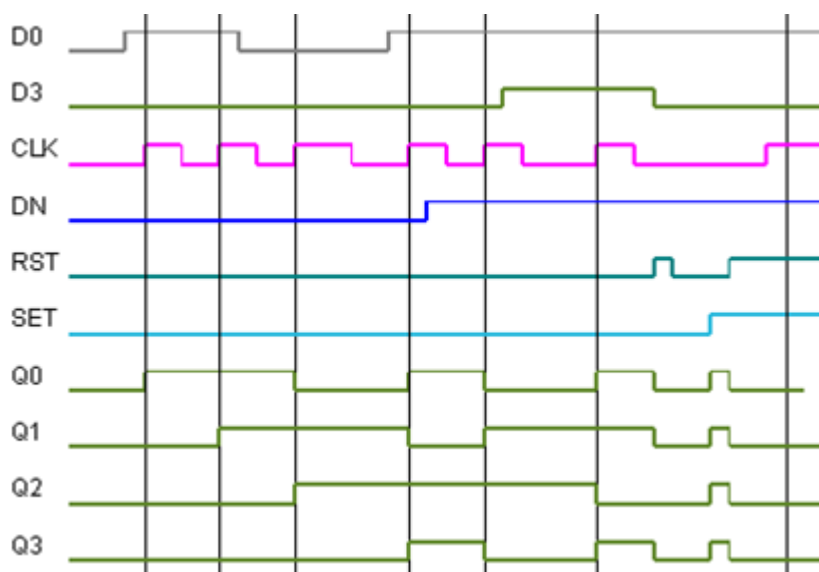
### 10.2.5.14. SHR\_4UDE (OSCAT)



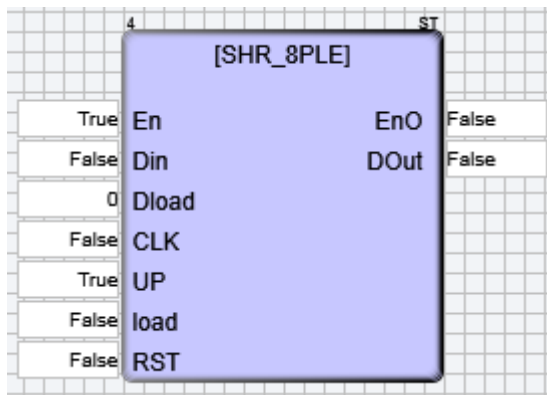
Тип данных входов и выходов – BOOL.

**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.



### 10.2.5.15. 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.

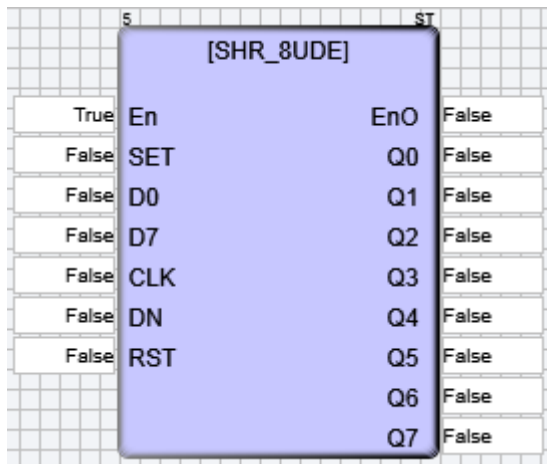
При RST=LOAD=FALSE:

- 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).

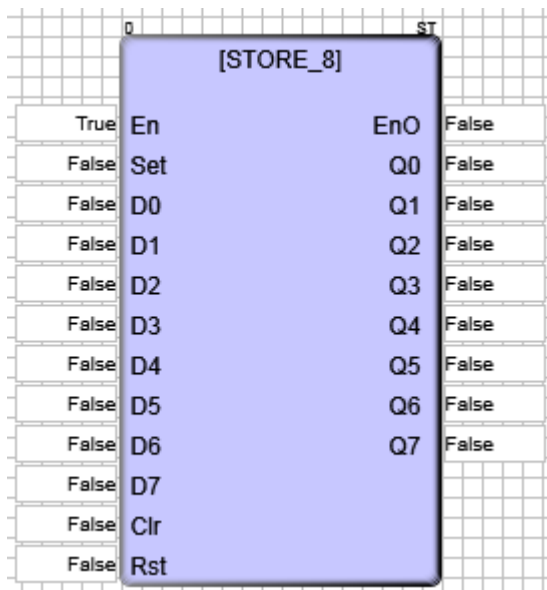
### 10.2.5.16. SHR\_8UDE (OSCAT)



Тип данных входов и выходов – BOOL.

**SHR\_8UDE** – это 8-разрядный аналог SHR\_4UDE (OSCAT).

### 10.2.5.17. STORE\_8 (OSCAT)



Тип данных входов и выходов – BOOL.

**STORE\_8** – это элемент памяти восьми событий.

При RST=SET=FALSE:

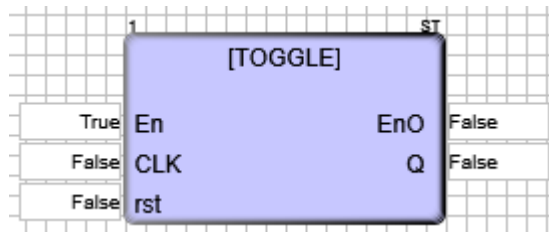
- значение TRUE входов D0..D7 записывается соответственно в выходы Q0..Q7 (значение FALSE входа не передается на соответствующий выход);

- 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).

### 10.2.5.18. TOGGLE (OSCAT)



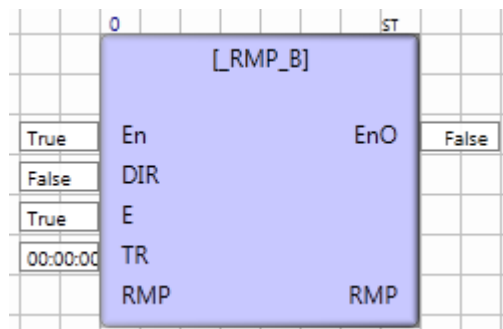
Тип данных входов и выхода – BOOL.

При  $RST=FALSE$  передний фронт на входе CLK инвертирует значение выхода Q.

При  $RST=TRUE$   $Q:=FALSE$  (сброс).

## 10.2.6. OSCAT.ФБ.Генераторы сигналов

### 10.2.6.1. \_RMP\_B (OSCAT)



Тип данных входов DIR и E – BOOL, TR – TIME, входа-выхода RMP – BYTE.

\_RMP\_B – генератор пилообразного сигнала в диапазоне 0..255.

Вход DIR задает направление пила ( $DIR=TRUE/FALSE$  – возрастающая/убывающая пила), 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\_B (OSCAT).

Примеры:

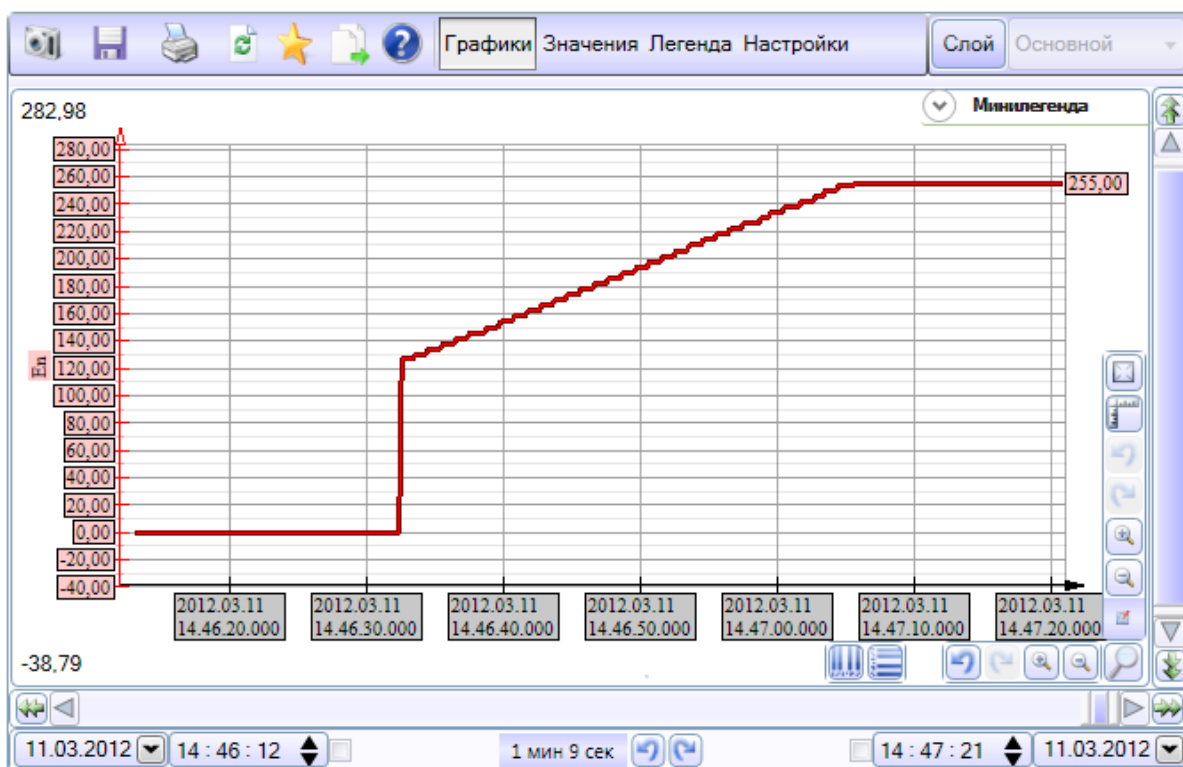
Пусть ST-программа с выходом BYTE\_out типа BYTE и локальной переменной \_RMP\_B\_1 типа \_RMP\_B имеет следующий код:

```
var mBYTE1:BYTE:=127; end_var;
```

```
_RMP_B_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 пилой (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;
```

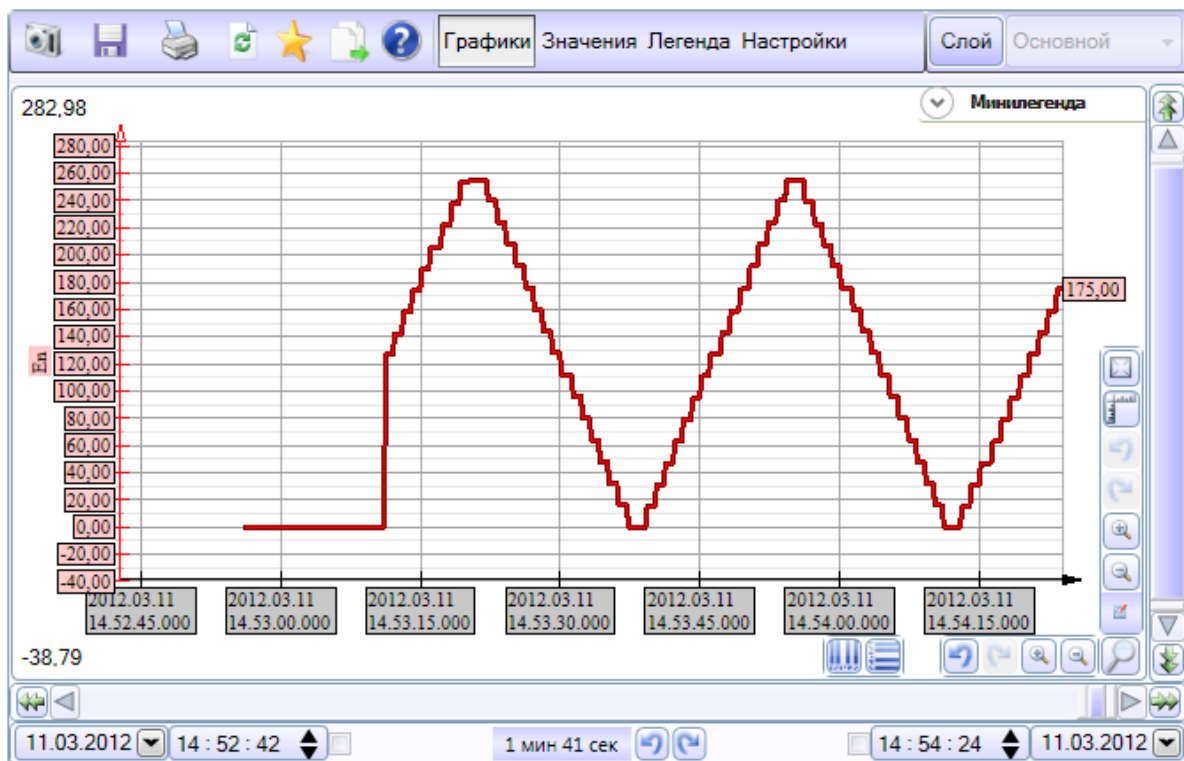
```
elsif GGG=FALSE and mBYTE1=0 then
```

```
GGG:=TRUE;
```

```
end_if;
```

```
_RMP_B_1(DIR:=GGG, E:=true, TR:=t#16s, RMP:=mBYTE1);
```

```
BYTE_out := mBYTE1;
```





### 10.2.6.2. `_RMP_NEXT` (OSCAT)

1		ST	
[RMP_NEXT]			
True	En	EnO	False
True	E	DIR	False
0	IN	UP	False
00:00:00	TR	DN	False
00:00:00	TF		
00:00:00	TL		
	OUT	OUT	

Тип данных входов TR, TF и TL – TIME, E – BOOL, IN – BYTE, выходов DIR, UP и DN – BOOL, входа-выхода OUT – BYTE.

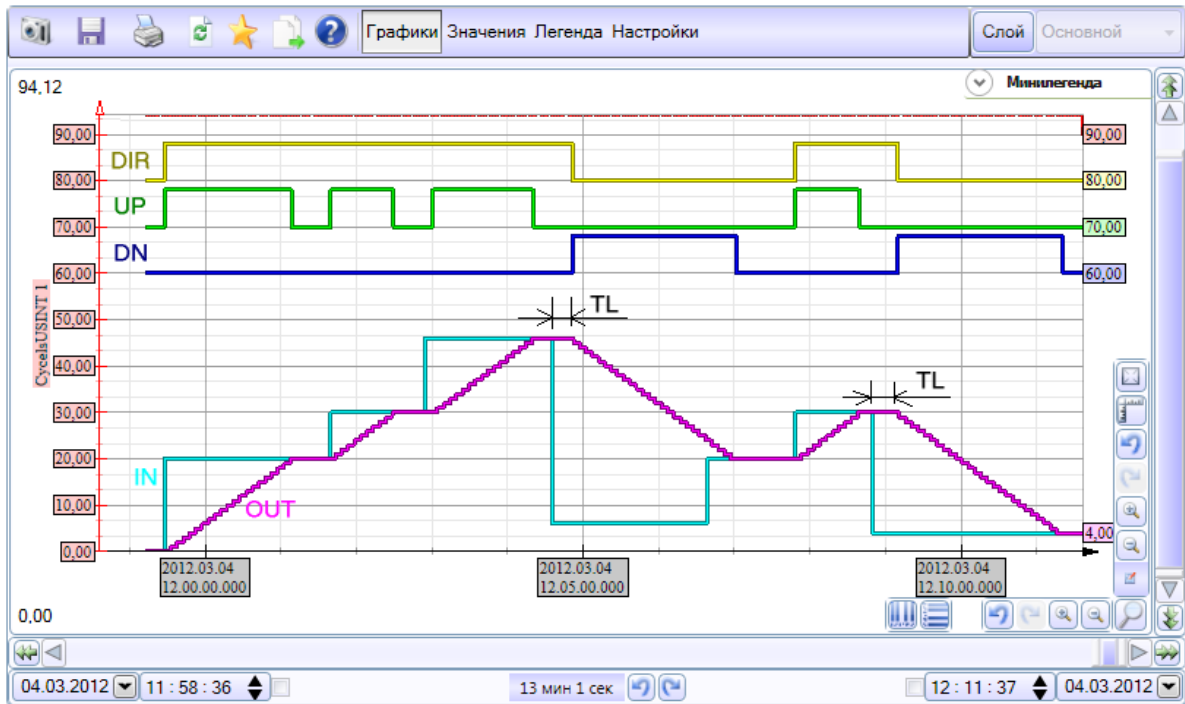
Вход E разрешает (TRUE) или приостанавливает (FALSE) исполнение алгоритма ФБ. При E=FALSE выходы ФБ сохраняют свои значения.

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.

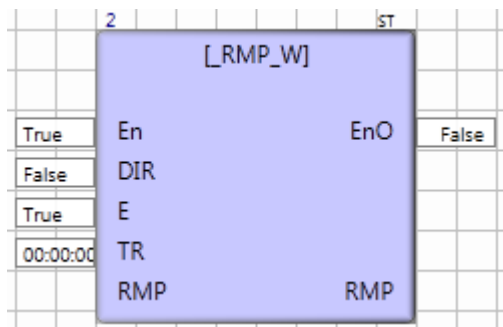
В течение возрастания/убывания OUT UP:=TRUE/FALSE и DN:=FALSE/TRUE.

При изменении направления с возрастания на убывание процедура задерживается на время TL.



Используемые функции: RMP\_B (OSCAT), TREND\_DW (OSCAT).

### 10.2.6.3. RMP\_W (OSCAT)

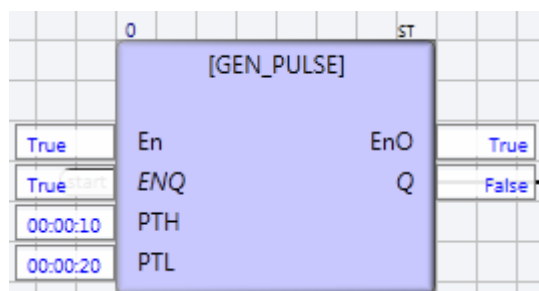


Тип данных входов DIR и E – BOOL, TR – TIME, входа-выхода RMP – WORD.

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).

### 10.2.6.4. 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.

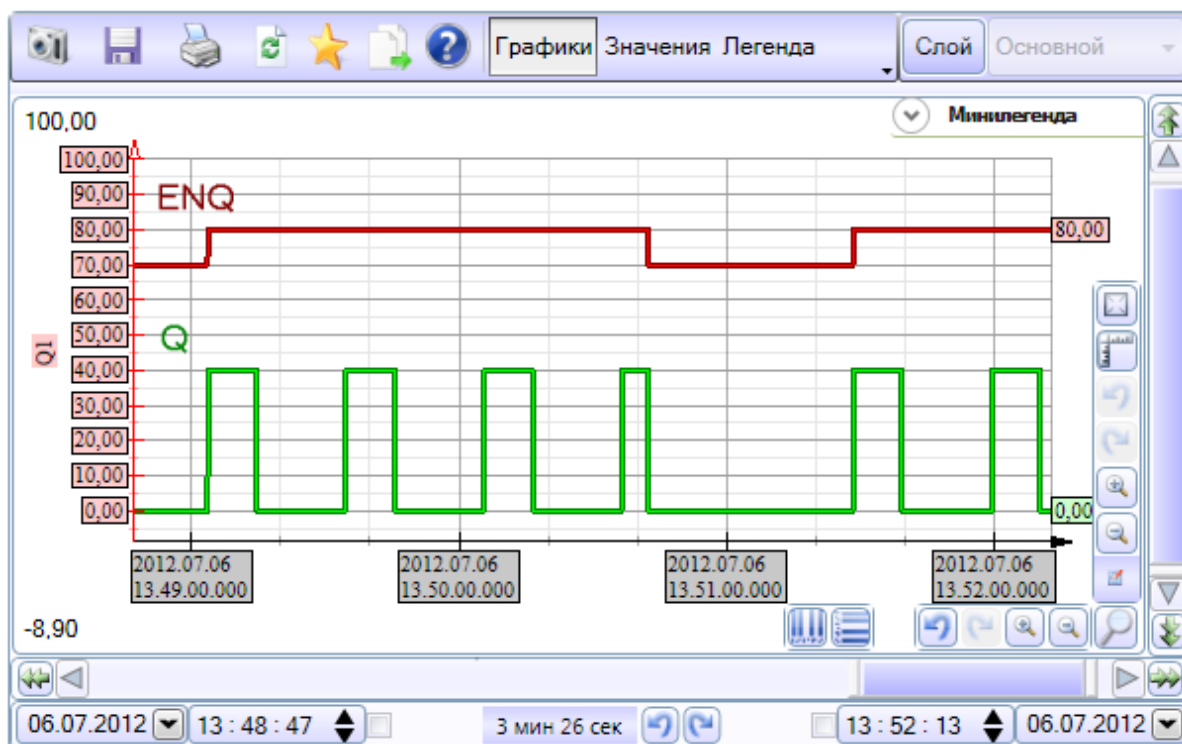
Команда запуска генератора – ENQ:=TRUE, при запуске вырабатывается первый импульс. Генерация импульсов продолжается до тех пор, пока ENQ:=TRUE.

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:



### 10.2.6.5. GEN\_PW2 (OSCAT)

0	[GEN_PW2]		1st
True	En	EnO	False
False	ENQ	Q	False
00:00:00	TH1	TL	00:00:00
00:00:00	TL1	TL	00:00:00
00:00:00	TH2		
00:00:00	TL2		
False	TS		

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;

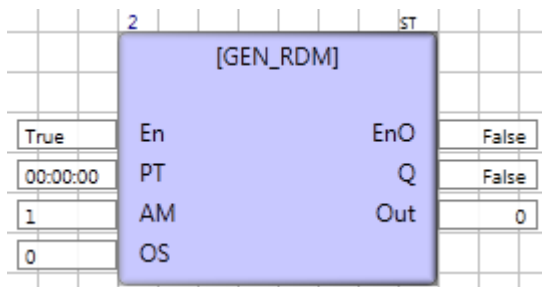
- TL2 – пауза между импульсами в сигнале 2;

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).

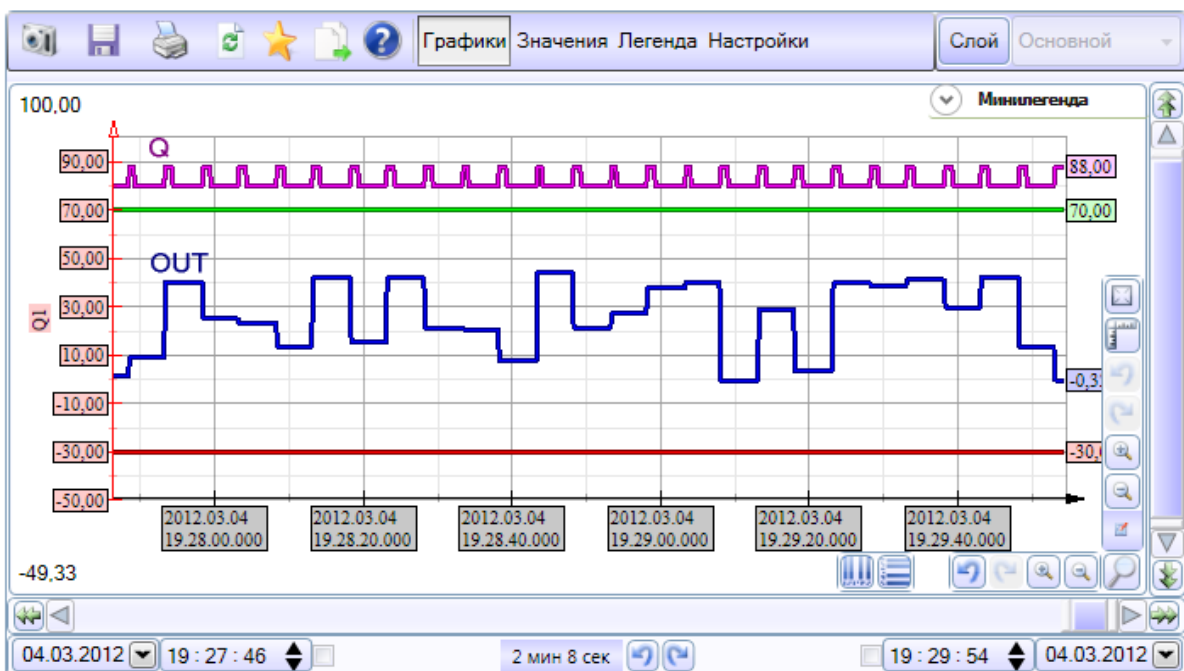
### 10.2.6.6. GEN\_RDM (OSCAT)



Тип данных входа PT – TIME, входов AM и OS и выхода OUT – REAL, выхода Q – BOOL.

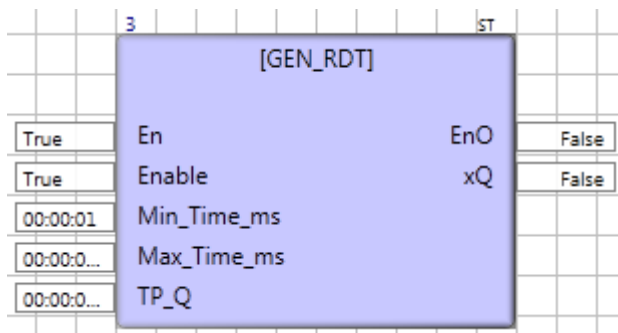
GEN\_RDM генерирует на выходе OUT случайное число в интервале  $(-0.5 \cdot AM + OS, 0.5 \cdot AM + OS)$  с частотой 1 раз за время PT. При изменении OUT Q=1 в течение 1 цикла.

Временная диаграмма ФБ при AM=50 и OS=20:



Используемые функции: T\_PLC\_MS (OSCAT), RDM (OSCAT).

### 10.2.6.7. GEN\_RDT (OSCAT)

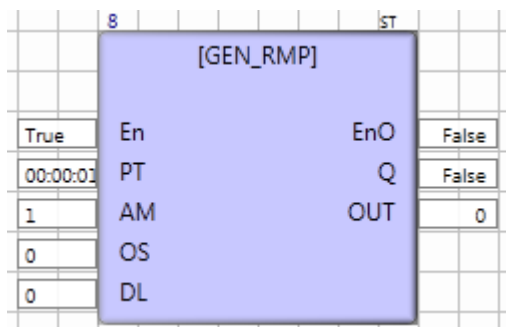


Тип данных входов MIN\_TIME\_MS, MAX\_TIME\_MS и TP\_Q – TIME, входа ENABLE и выхода XQ – BOOL.

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.

### 10.2.6.8. GEN\_RMP (OSCAT)



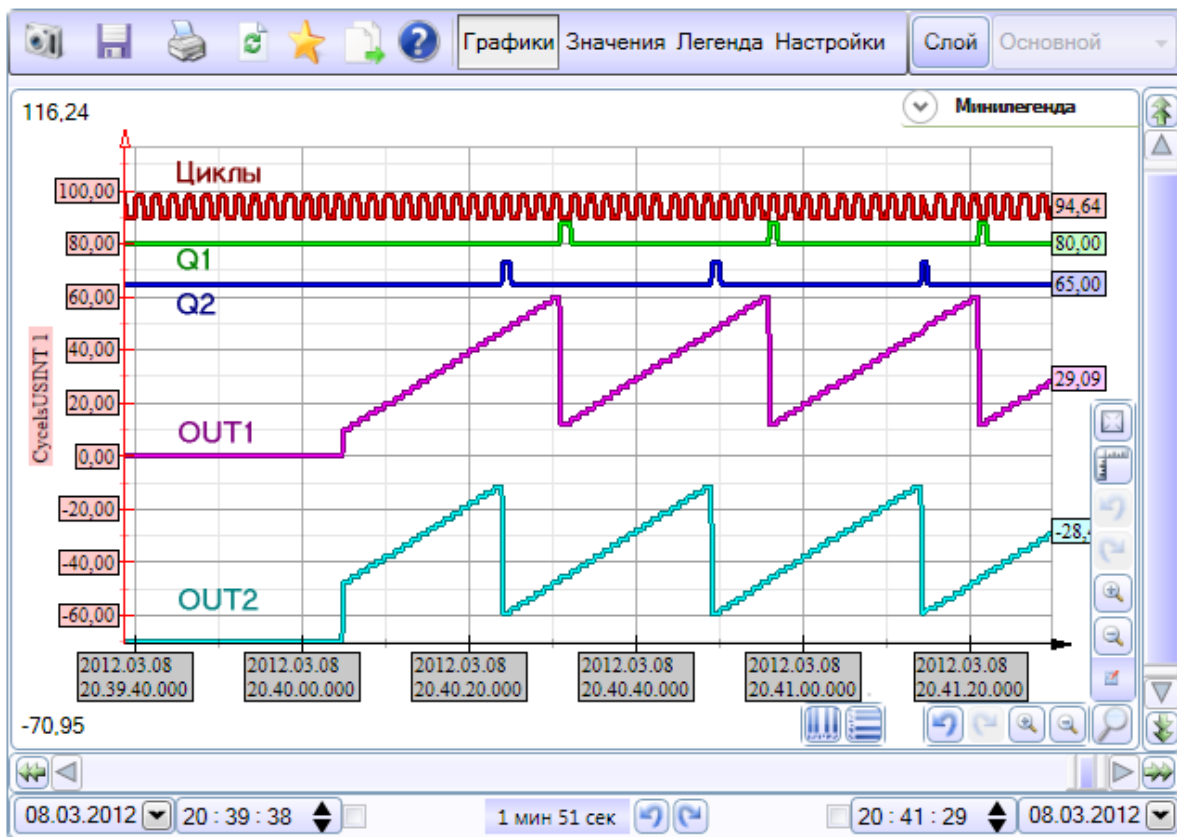
Тип данных входа PT – TIME, выхода Q – BOOL, остальных входов и выхода OUT – REAL.

GEN\_RMP генерирует на выходе OUT пилообразный сигнал в диапазоне [OS, AM+OS] с периодом PT. Число шагов в периоде – PT/<время цикла>, шаг по оси значений – AM\*<время цикла>/PT. По завершении каждого периода Q=1 на 1 цикл.

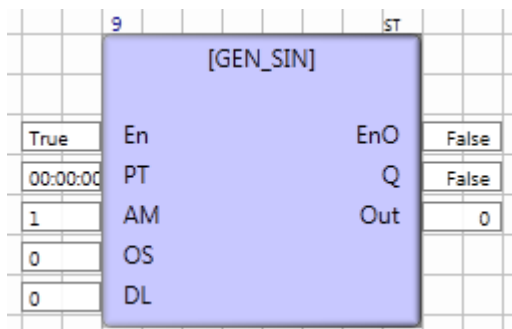
Вход DL задает смещение сигнала по фазе. Значение этого входа должно находиться в диапазоне [-1, 1], что соответствует фазовому сдвигу на величину от  $(-2\pi)$  (отставание на период (PT)) до  $2\pi$  (опережение на период (PT)).

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):



### 10.2.6.9. GEN\_SIN (OSCAT)



Тип данных входа PT – TIME, выхода Q – BOOL, остальных входов и выхода OUT – REAL.

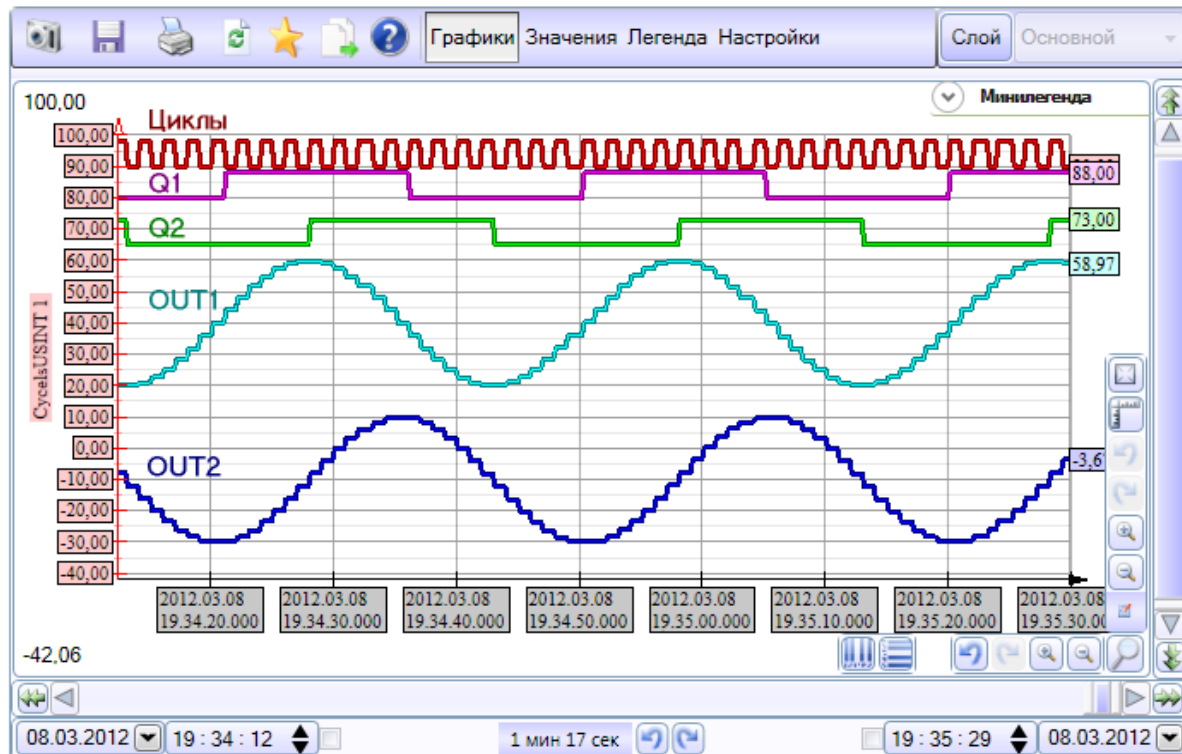
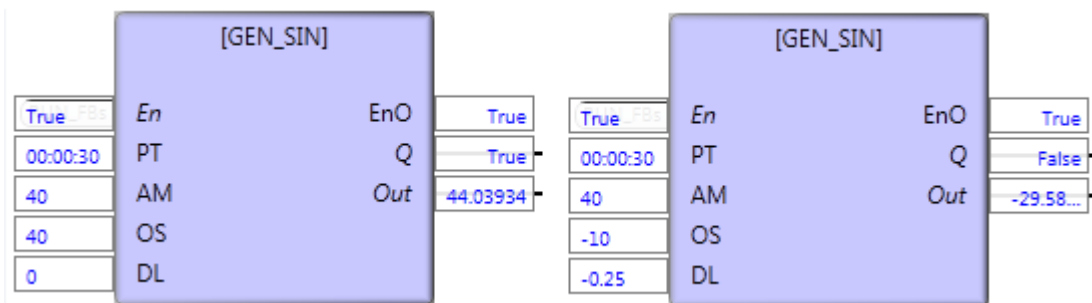
GEN\_SIN генерирует на выходе OUT синусоидальный сигнал в диапазоне  $[-0.5*AM + OS, 0.5*AM+OS]$  с периодом PT.

Вход DL задает смещение сигнала по фазе. Значение этого входа должно находиться в диапазоне  $[-1, 1]$ , что соответствует фазовому сдвигу на величину от  $(-\frac{\pi}{2})$  (отставание на период (PT)) до  $(\frac{\pi}{2})$  (опережение на период (PT)).

$Q:=TRUE$  в первом полупериоде генерируемого сигнала (фаза принадлежит диапазону  $[\pm 2\pi n, \pi \pm 2\pi n]$ ) и  $Q:=FALSE$  во втором полупериоде (фаза принадлежит диапазону  $[\pi \pm 2\pi n, 2\pi \pm 2\pi n]$ ).

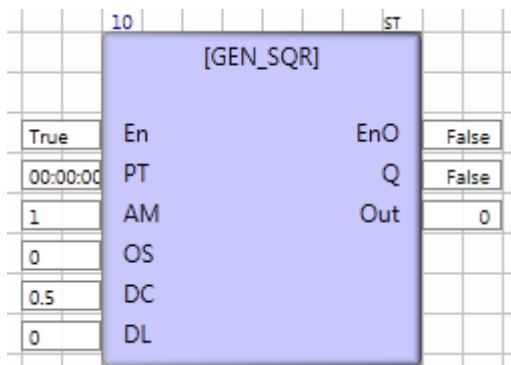
Используемые функции: T\_PLC\_MS (OSCAT), MODR (OSCAT), SIGN\_R (OSCAT).

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)):





### 10.2.6.10. GEN\_SQR (OSCAT)



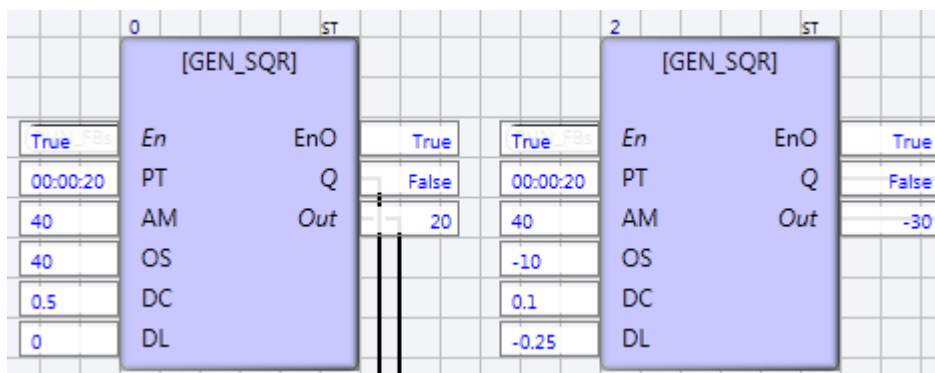
Тип данных входа PT – TIME, выхода Q – BOOL, остальных входов и выхода OUT – REAL.

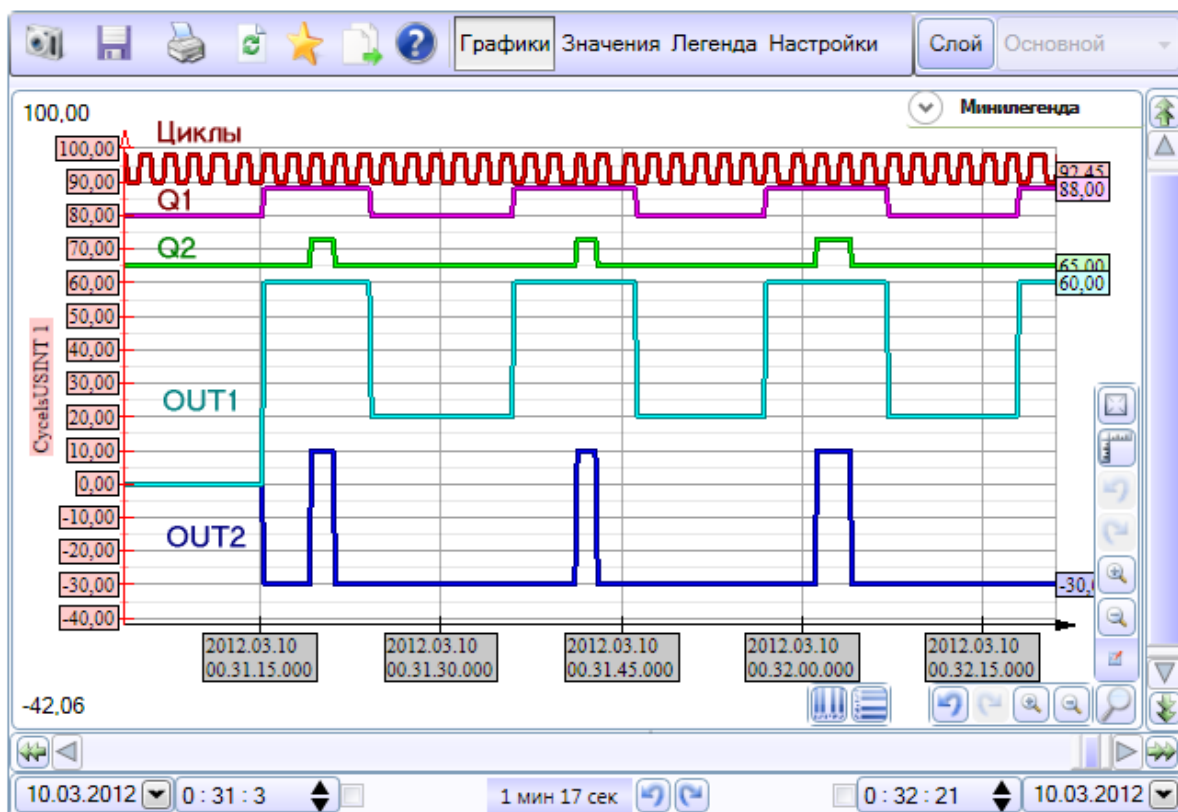
GEN\_SQR генерирует на выходе OUT прямоугольный сигнал в диапазоне  $[-0.5*AM + OS, 0.5*AM+OS]$ , а на выходе Q – аналогичный цифровой сигнал с периодом следования импульсов PT.

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.

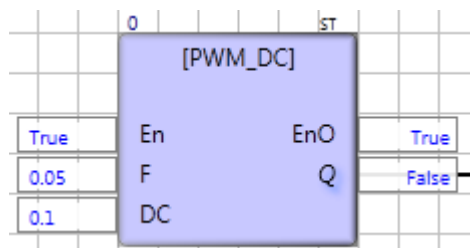
Дробная часть значения входа DL задает смещение сигнала по фазе (-1 соответствует фазовому сдвигу на -3600 (отставание на период PT), +1 – фазовому сдвигу на +3600 (опережение на период PT)).

Ниже на рисунках показаны конфигурации двух ФБ GEN\_SQR и их временные диаграммы (OUT2 отстает по фазе от OUT1 на четверть периода):





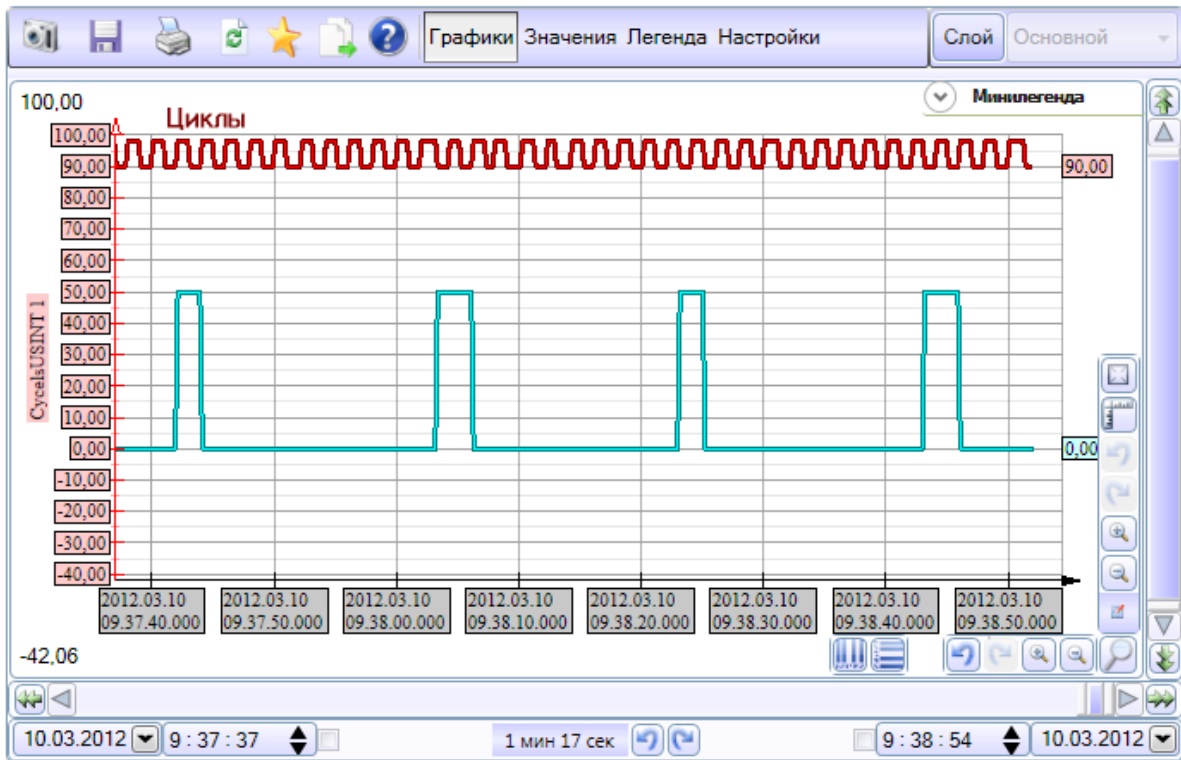
### 10.2.6.11. PWM\_DC (OSCAT)



Тип данных входов – REAL, выхода – BOOL.

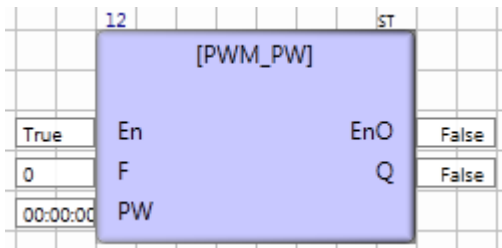
PWM\_DC генерирует прямоугольный сигнал с заданной частотой F, Гц и длительностью импульса ( $DC/F$ ), с. Значение DC должно лежать в диапазоне  $[0, 1]$ .

Например, при показанной выше конфигурации генерируются импульсы шириной 2с и периодом 20с:



Используемые функции: CLK\_PRG (OSCAT), TP\_X (OSCAT).

### 10.2.6.12. PWM\_PW (OSCAT)

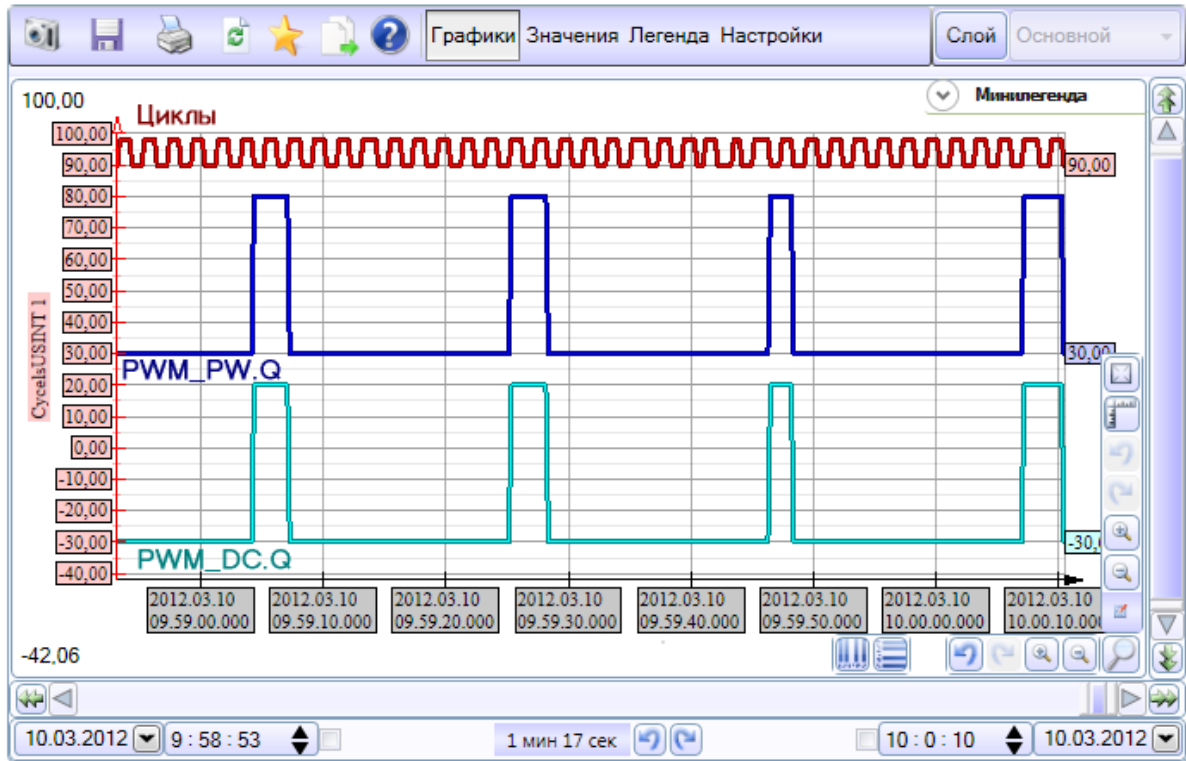
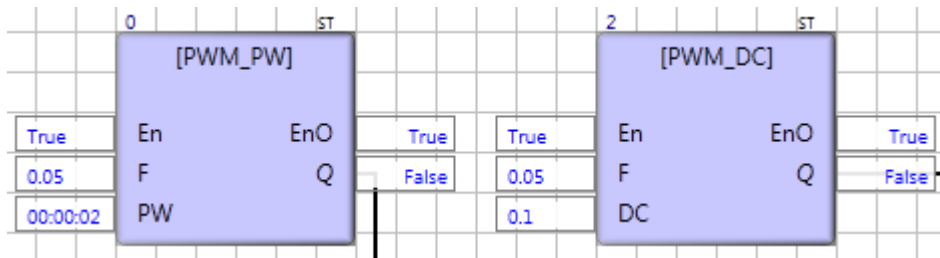


Тип данных входа F – REAL, входа PW – TIME, выхода Q – BOOL

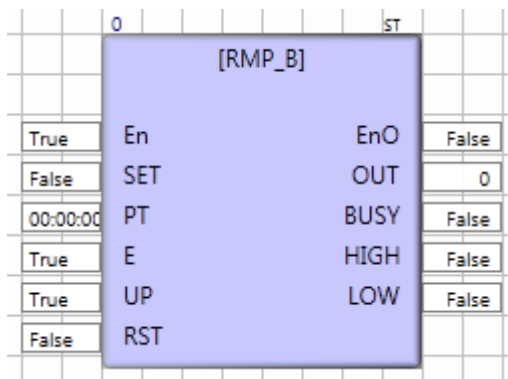
**PWM\_PW** – это аналог PWM\_DC (OSCAT), генерирует прямоугольный сигнал с заданной частотой F, Гц и длительностью импульса PW.

Используемые функции: CLK\_PRG (OSCAT), TP\_X (OSCAT).

Ниже для сравнения показаны оба ФБ, генерирующие одинаковые сигналы (ширина импульса – 2с, период – 20с):



### 10.2.6.13. 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 пилла), 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 пила (при E=TRUE) SET и RST только маскируют (скрывают) генерируемое значение OUT.

Пока OUT:=0, LOW:=TRUE, в остальных случаях LOW:=FALSE.

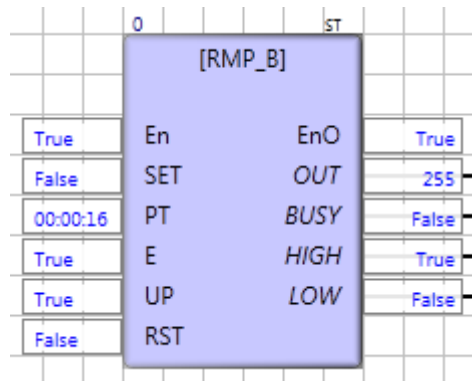
Пока OUT:=255, HIGH:=TRUE, в остальных случаях HIGH:=FALSE.

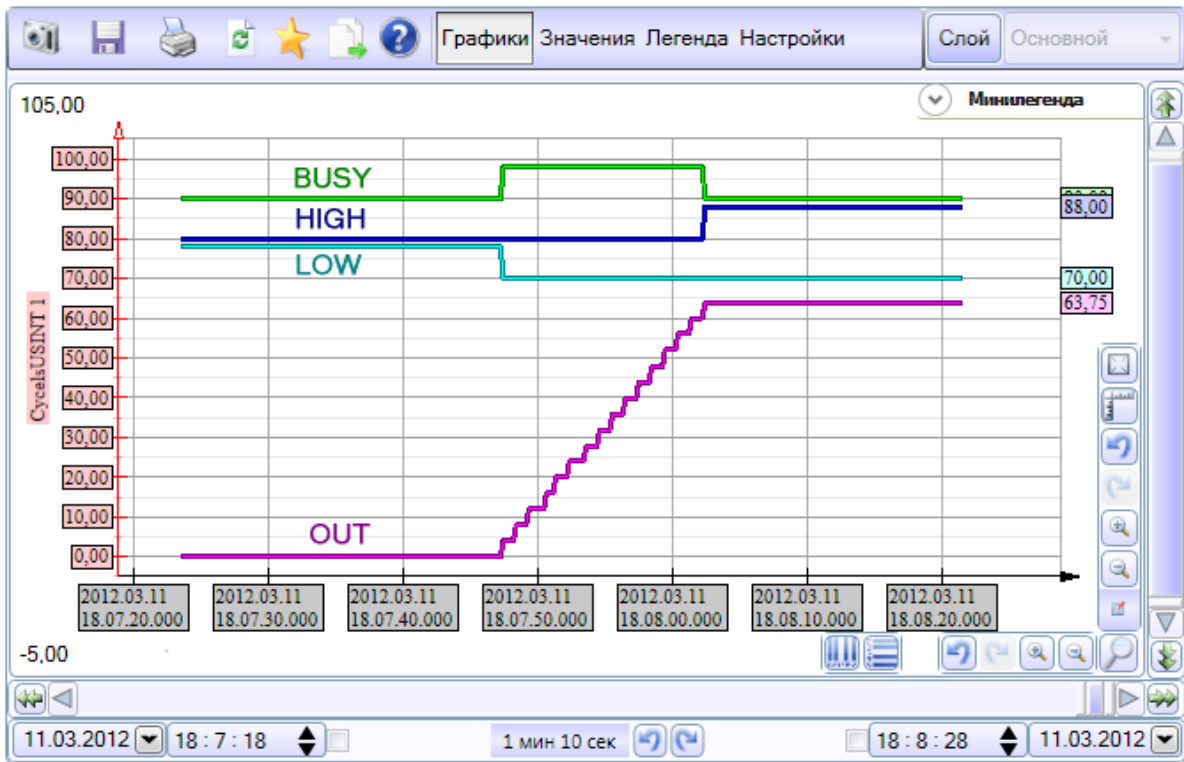
BUSY:=TRUE при генерации пила, т.е. если E:=TRUE и  $0 < \text{OUT} < 255$ .

Число шагов в периоде –  $\text{PT} / \langle \text{время цикла} \rangle$ , шаг по оси значений –  $255 * \langle \text{время цикла} \rangle / \text{PT}$ .

#### Пример 1

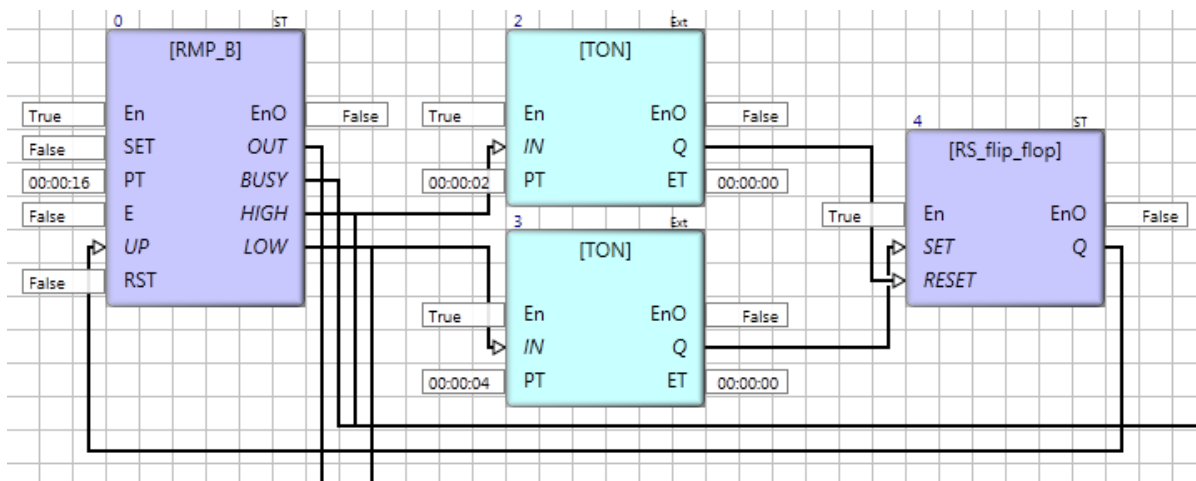
Следующая программа при неизменной конфигурации RMP\_B генерирует один период возрастающей пила:

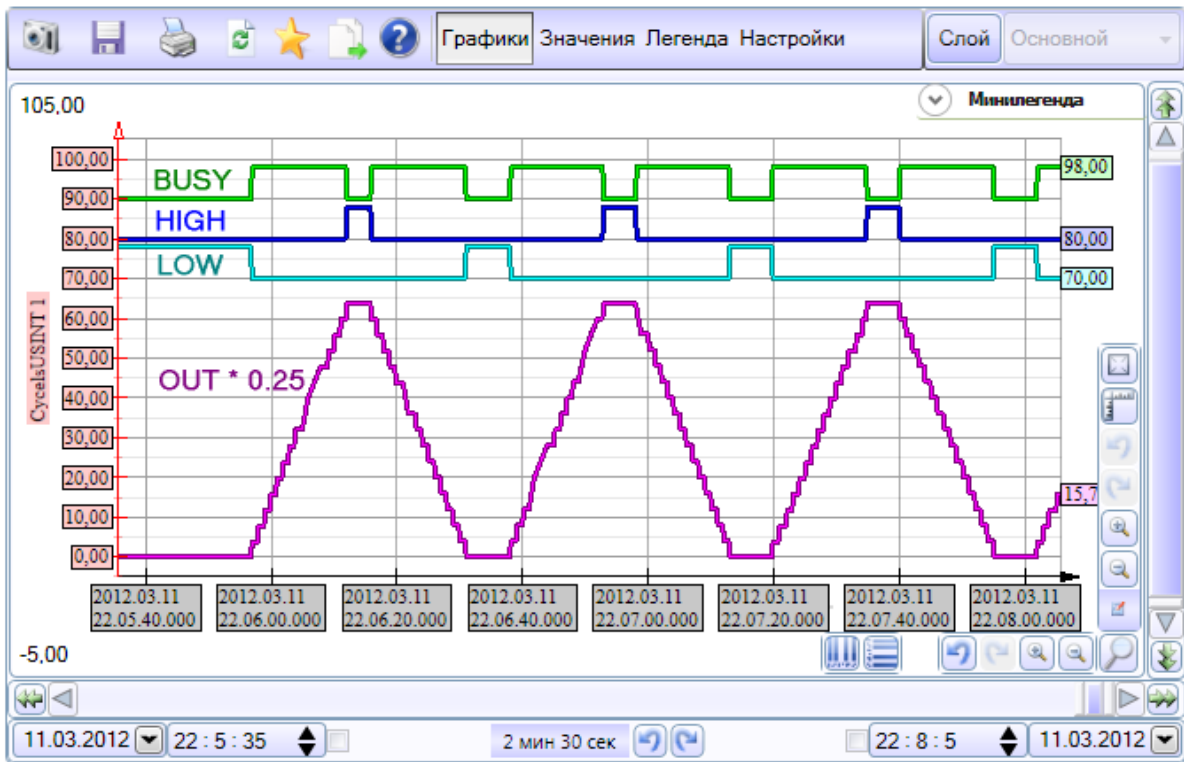




Пример 2

Следующая программа после RMP\_B.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):



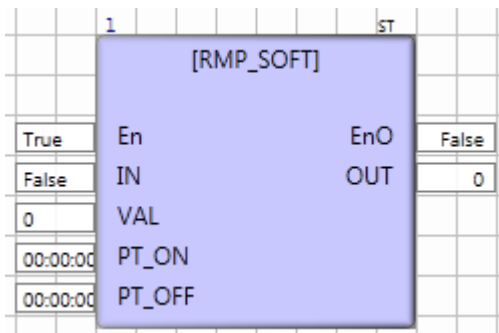


Код пользовательского ФБ RS\_flip\_flop:

```
if SET and NOT RESET then Q:=TRUE; end_if;
```

```
if RESET then Q:=FALSE; end_if;
```

#### 10.2.6.14. 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:

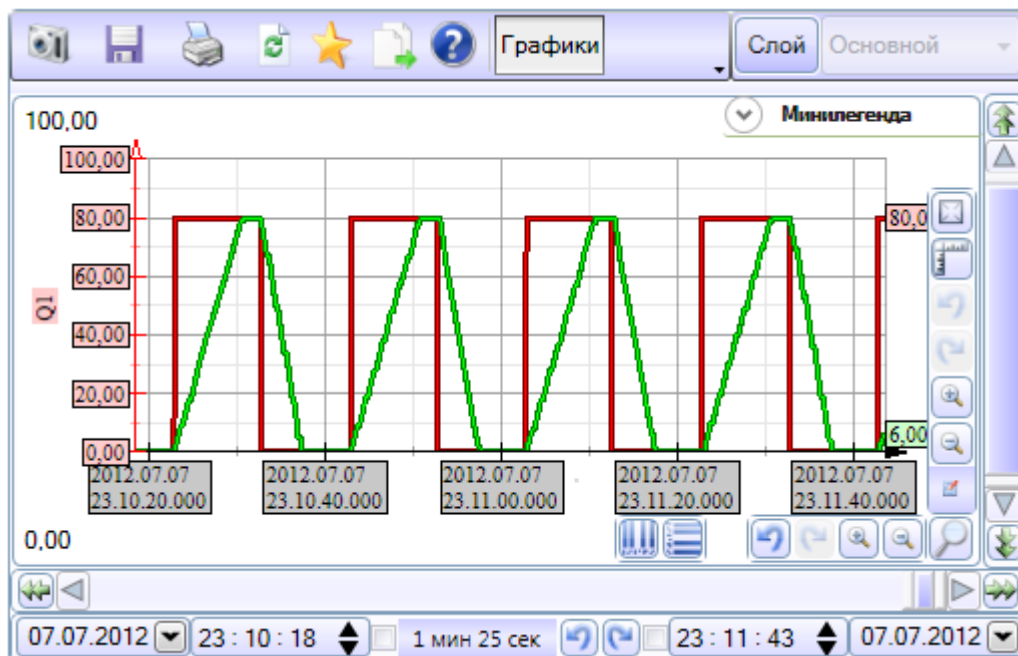
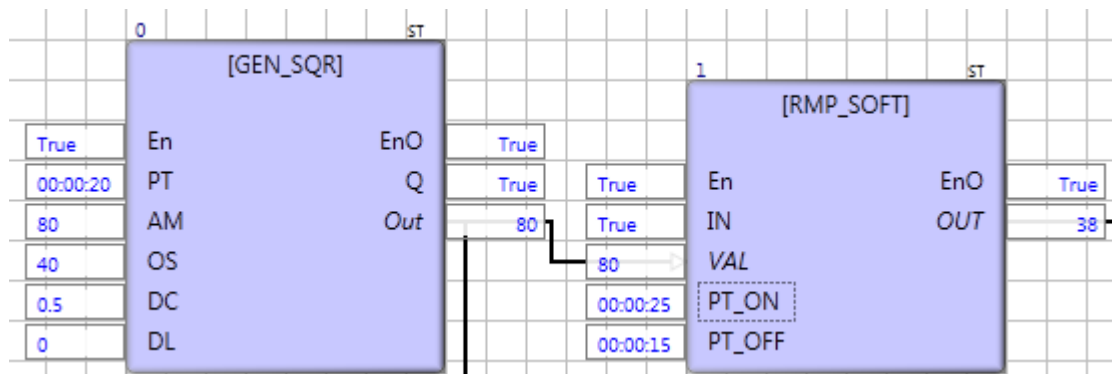
- сглаживание переднего фронта: PT\_ON задает время нарастания сигнала от 0 до 255; если  $VAL < 255$ , время нарастания сигнала на выходе OUT будет соответственно меньше;
- сглаживание заднего фронта: PT\_OFF задает время убывания сигнала от 255 до 0; если  $VAL < 255$ , время убывания сигнала на выходе OUT будет соответственно меньше.

При  $IN=FALSE$   $OUT=0$ .

Используемые функции: `_RMP_V` (OSCAT).

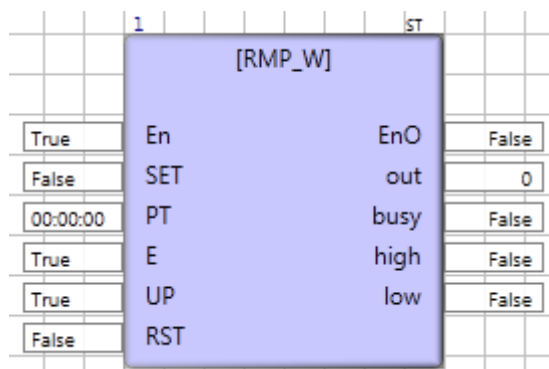
Пример

В качестве генератора прямоугольного сигнала в данном примере используется ФБ GEN\_SQR (OSCAT).





### 10.2.6.15. RMP\_W (OSCAT)



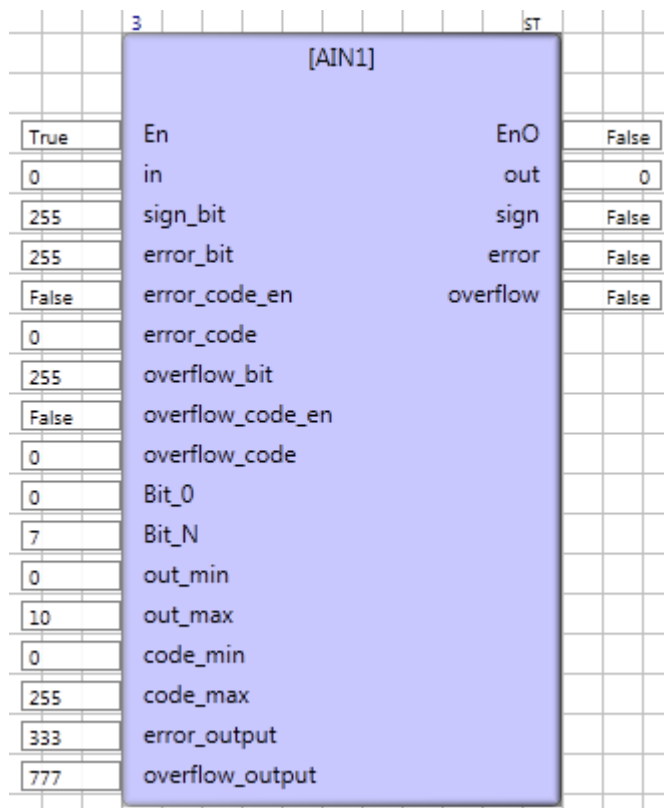
**RMP\_W** – это 16-битовый аналог RMP\_B (OSCAT), т.е. данный ФБ генерирует пилообразный сигнал в диапазоне 0..65535.

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.

## 10.2.7. OSCAT.ФБ.Обработка сигналов

### 10.2.7.1. 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

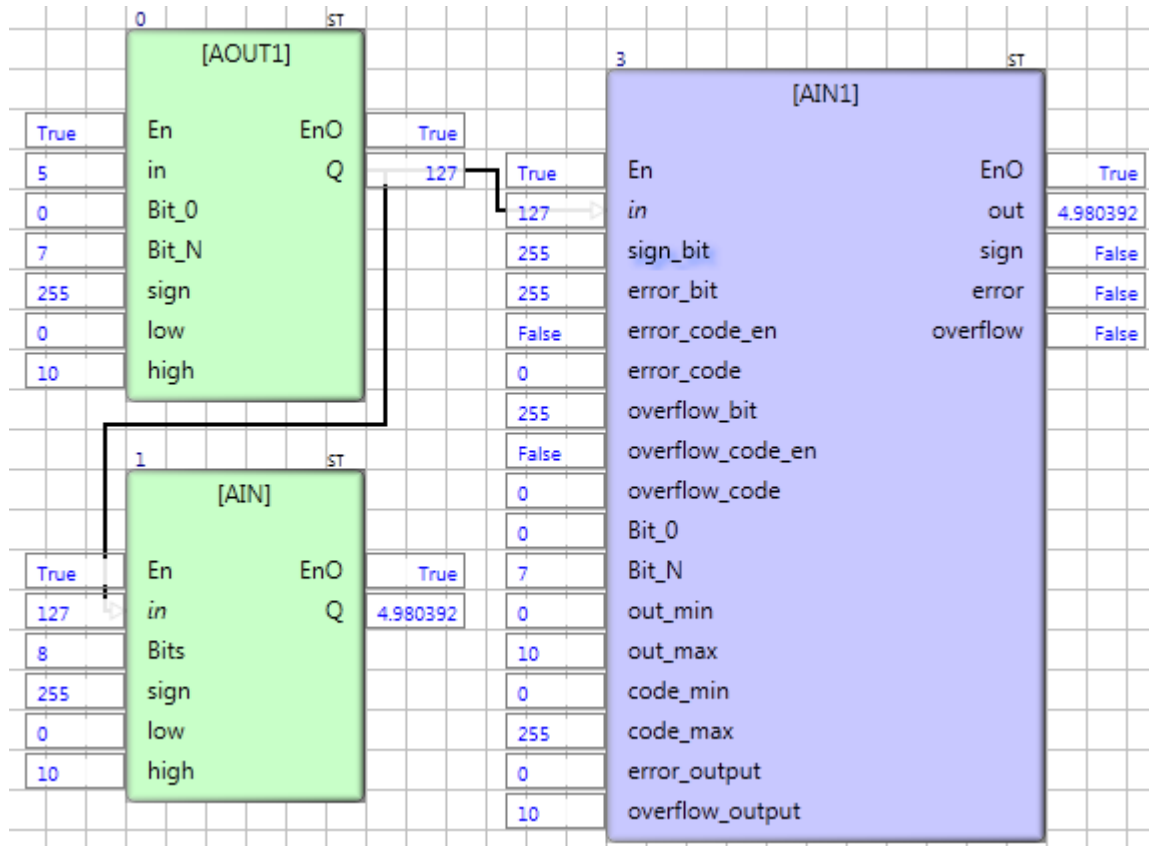
Поскольку АЦП могут сигнализировать о наличии ошибки с помощью передачи флага ошибки в некотором бите или с помощью передачи некоторого фиксированного значения (вне рабочего диапазона), ФБ AIN1 поддерживает оба эти механизма как для общей ошибки АЦП, так и для ошибки переполнения АЦП. По умолчанию, ошибки 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

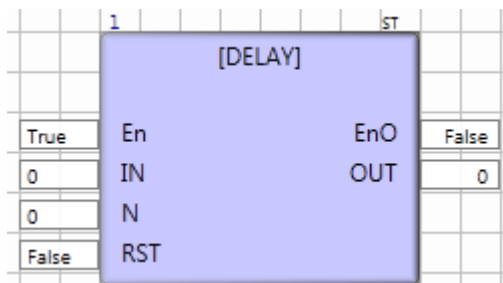
- Для обработки общей ошибки по флагу INT-вход ERROR\_BIT должен быть равен номеру бита, в котором АЦП передает флаг ошибки. Кроме того, нужно задать необходимое значение REAL-входу ERROR\_OUTPUT. Если значение IN содержит установленный флаг ошибки, BOOL-выход ERROR принимает значение TRUE, а выход OUT – 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.



### 10.2.7.2. DELAY (OSCAT)



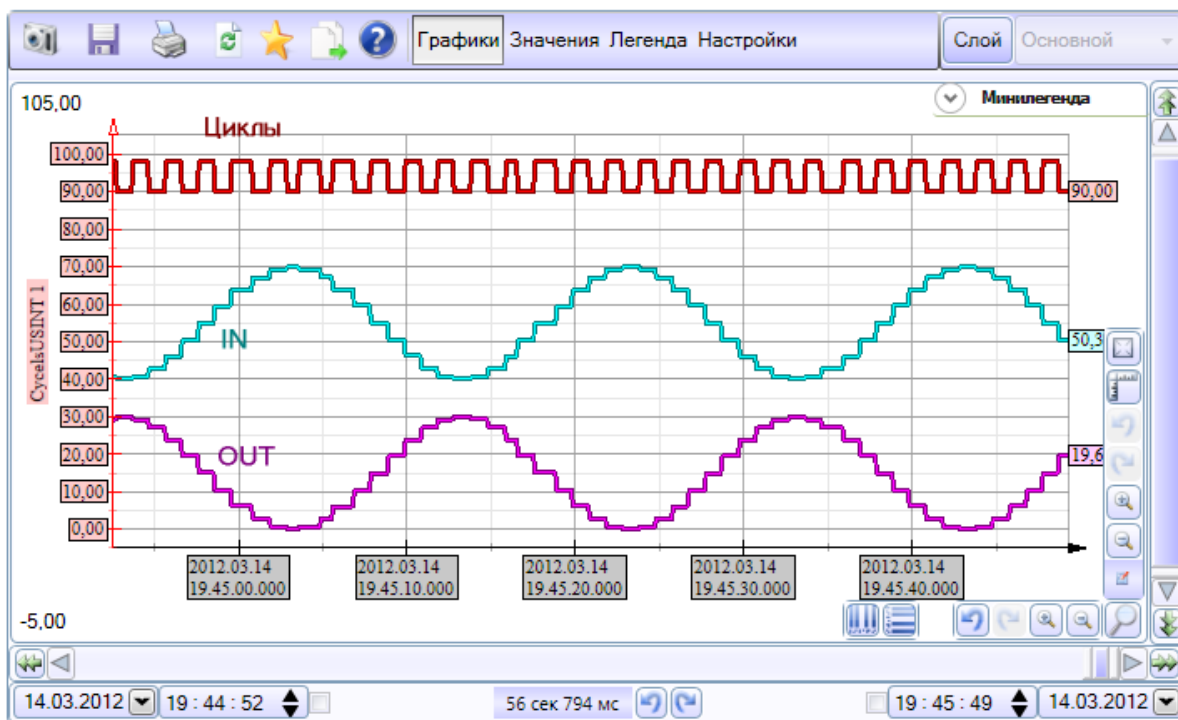
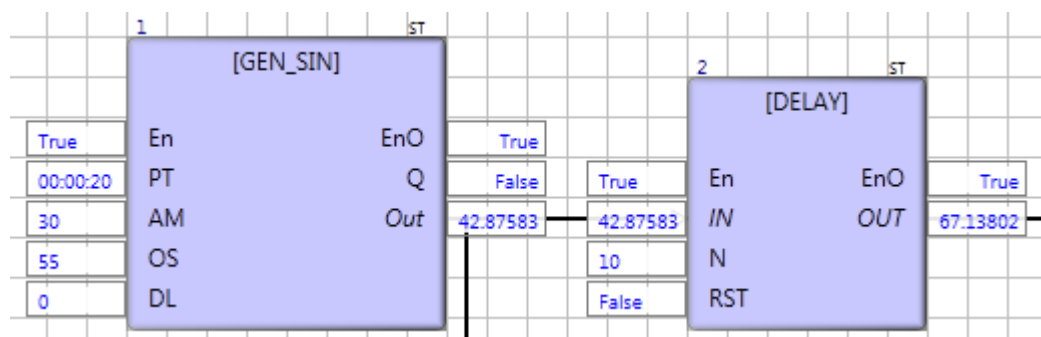
ФБ DELAY задерживает сигнал, поданный на REAL-вход IN, на N циклов (тип данных N – INT). Задержанный сигнал передается на REAL-выход OUT.

Для сброса буфера ФБ входу RST нужно присвоить значение TRUE (если RST=TRUE, OUT=IN).

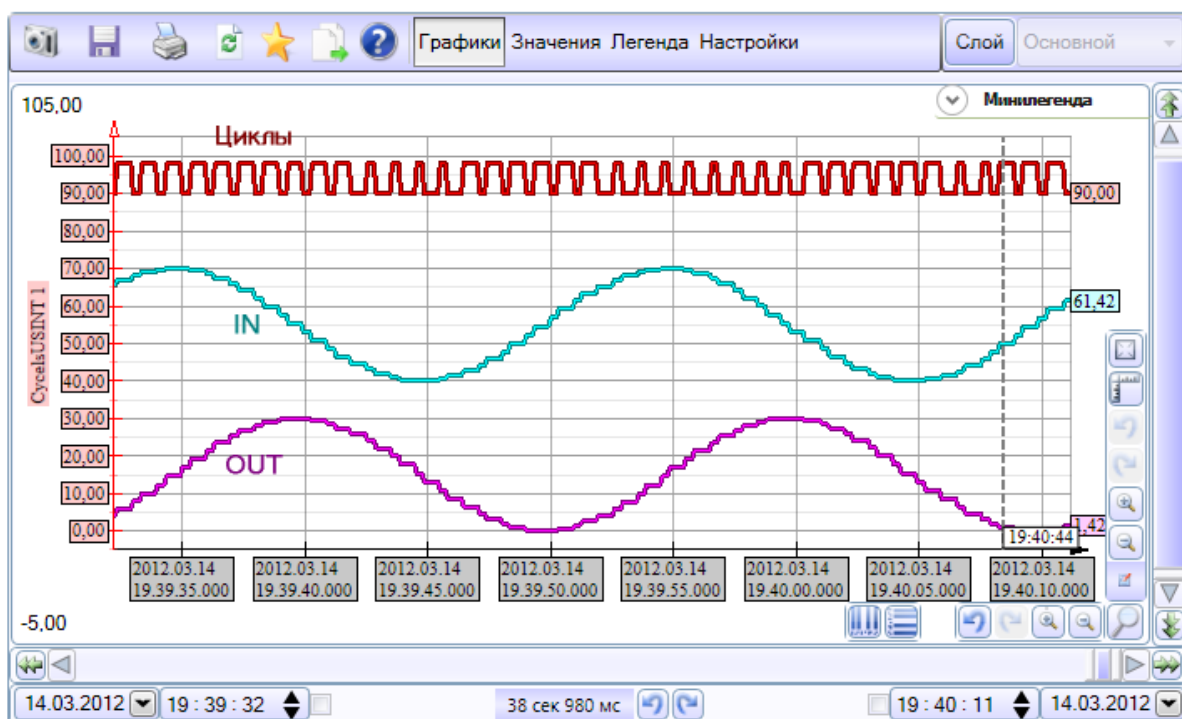
Используемые функции: INC1 (OSCAT).

## Пример

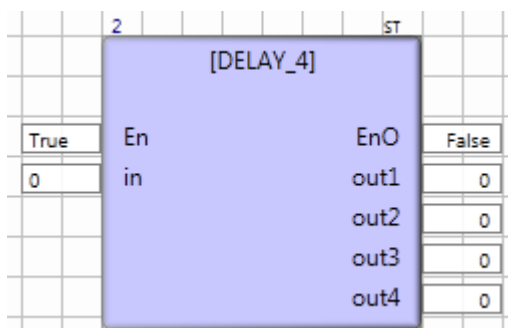
В данном примере длительность цикла – 1с, задержка N=10 циклов, поэтому синусоида с периодом 20с задерживается на половину периода ( $10 \cdot 1 = 10\text{с}$ ). Для генерации синусоиды используется ФБ GEN\_SIN (OSCAT).



В той же программе при времени цикла 0.5с синусоида задерживается на четверть периода ( $10 \cdot 0.5 = 5\text{с}$ ):



### 10.2.7.3. DELAY\_4 (OSCAT)

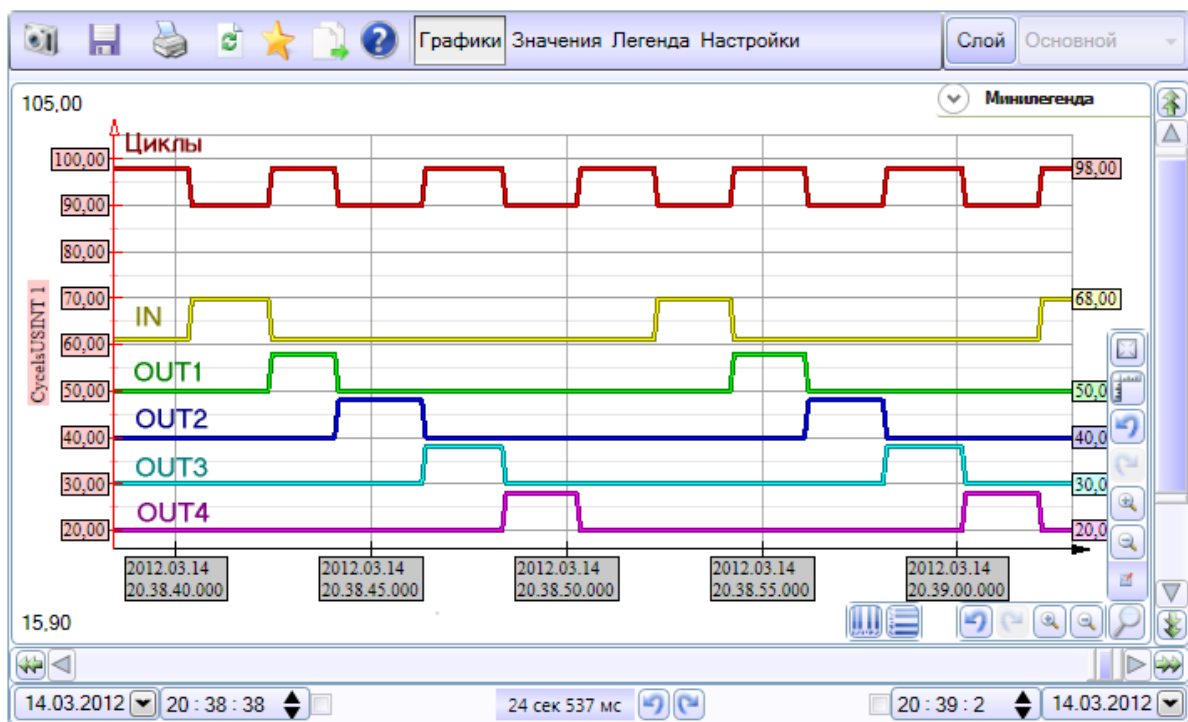
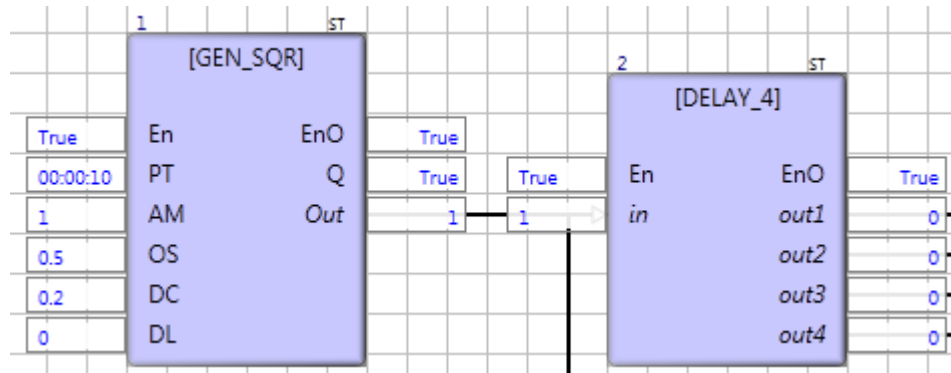


DELAY\_4 передает сигнал, поданный на REAL-вход IN, на REAL-выходы с фиксированными задержками:

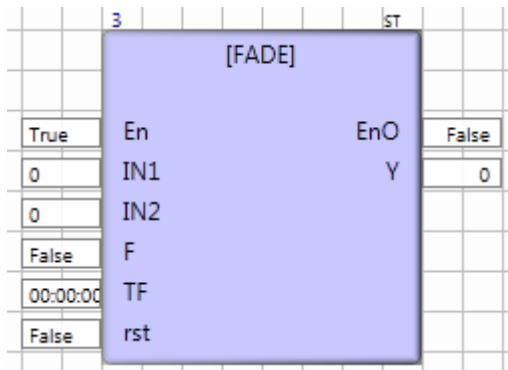
- на OUT1 – с задержкой 1 цикл;
- на OUT2 – с задержкой 2 цикла;
- на OUT3 – с задержкой 3 цикла;
- на OUT4 – с задержкой 4 цикла.

Пример

В данном примере временная диаграмма ФБ DELAY\_4 получена при длительности цикла 2с. В качестве генератора прямоугольного сигнала использован ФБ GEN\_SQR (OSCAT).



### 10.2.7.4. 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<sub>M</sub> на IN<sub>N</sub> (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).

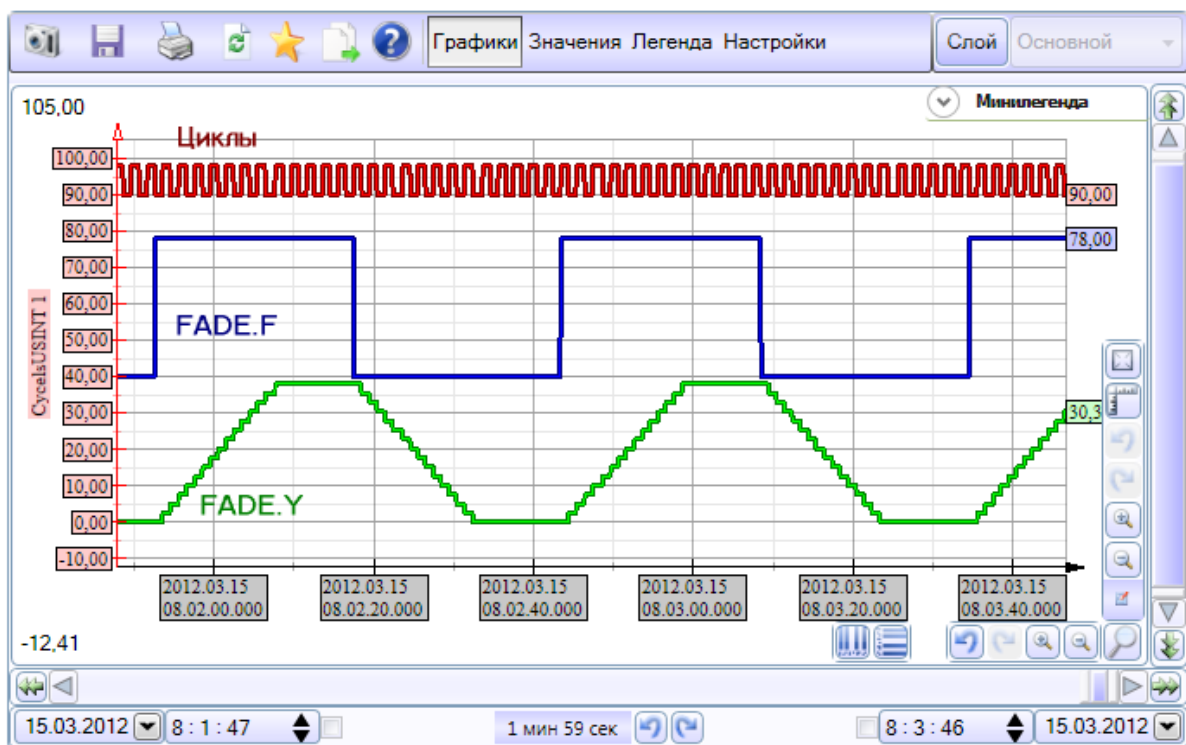
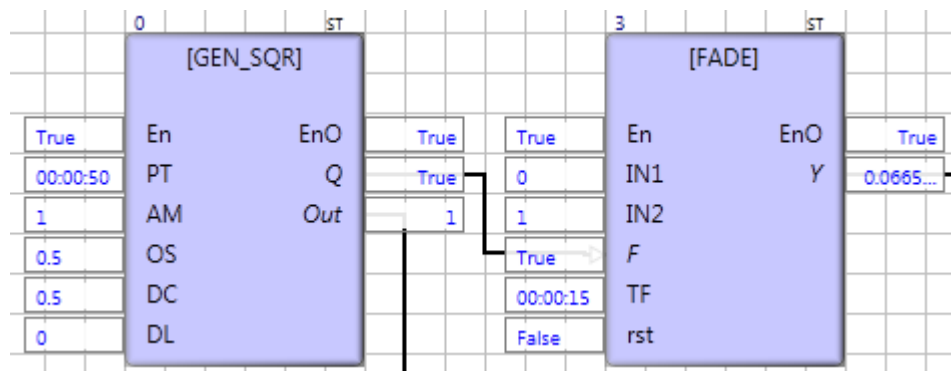


Т.к. выход вычисляется динамически, ФБ FADE может быть также использован для переключения динамических сигналов. В принципе, переход может быть разбит на 65535 шагов, однако длительность шага не может быть меньше, чем время цикла, поэтому, например, при времени цикла 10мс и TF=1с получится только  $1с/10мс = 100$  шагов.

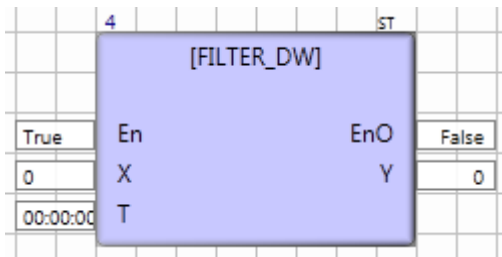
Используемые функции: RMP\_W (OSCAT).

Пример

В данном примере для переключения входов (для изменения значения F) используется ФБ GEN\_SQR (OSCAT). Время цикла – 1с, поэтому переход выполняется за  $15с/1с=15$  шагов.



### 10.2.7.5. FILTER\_DW (OSCAT)



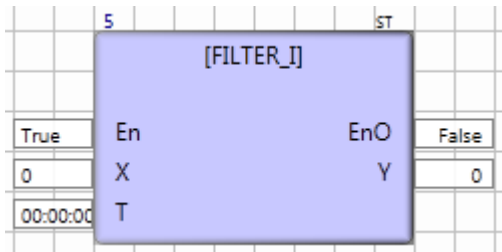
Тип данных входа T – TIME, входа X и выхода Y – DWORD.

**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).

### 10.2.7.6. FILTER\_I (OSCAT)



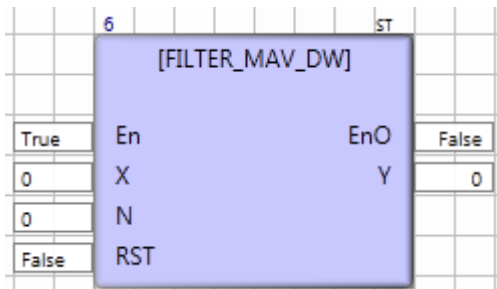
Тип данных входа T – TIME, входа X и выхода Y – INT.

**FILTER\_I** – 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).

### 10.2.7.7. 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.

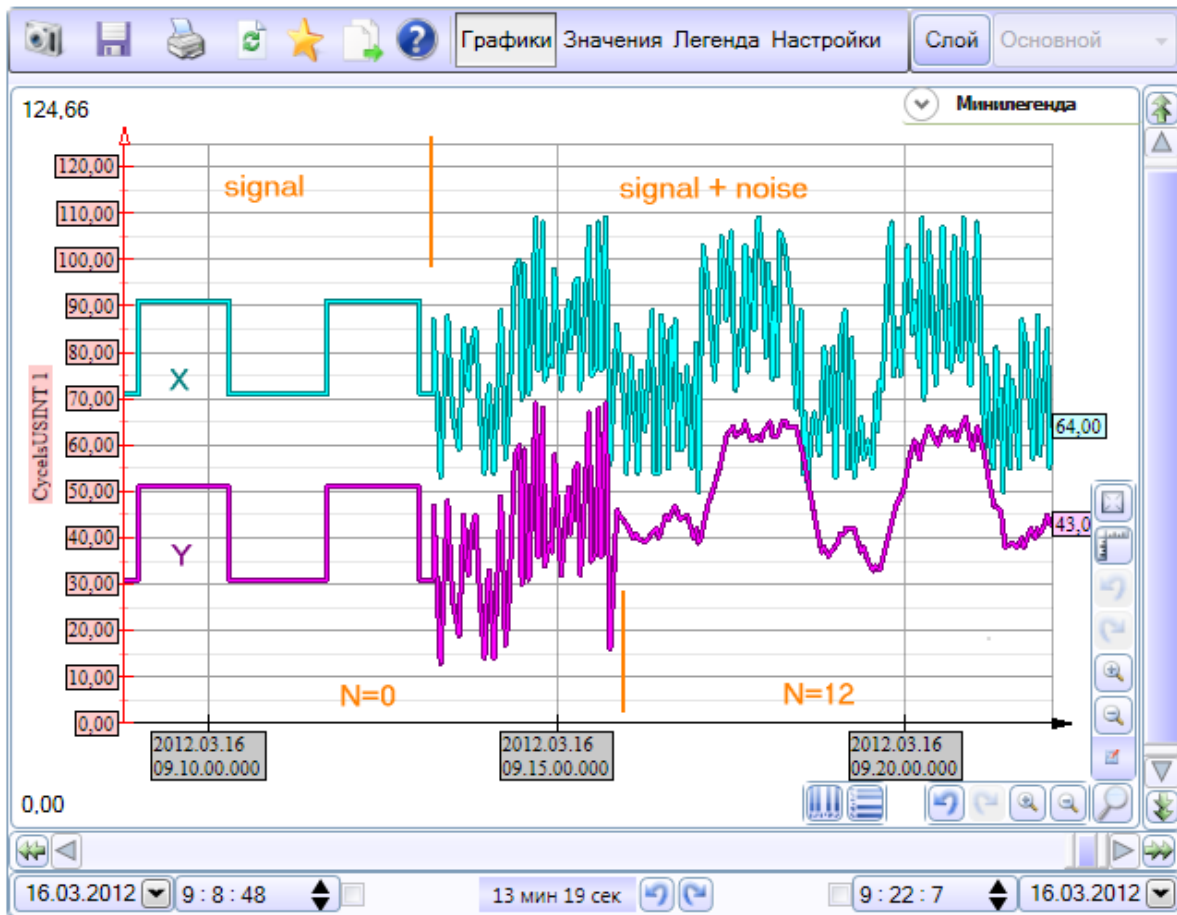
Если  $N=0$  или 1,  $Y:=X$ , в противном случае ФБ вычисляет скользящее среднее (метод сглаживания колебаний данных):

$$Y := (X_0 + X_1 + \dots + X_{N-1}) / N$$

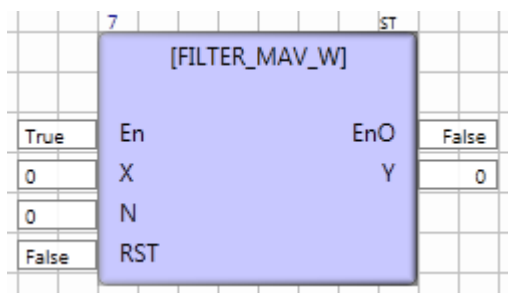
В формуле  $X_0$  – это значение входа X в текущем цикле,  $X_1$  – значение входа X в предыдущем цикле и т.д. Число значений, используемых в усреднении, определяет вход N ( $N \leq 32$ ).

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:

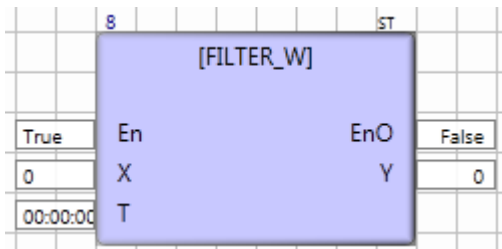


### 10.2.7.8. FILTER\_MAV\_W (OSCAT)



**FILTER\_MAV\_W** – это аналог ФБ **FILTER\_MAV\_DW (OSCAT)** вычисления скользящего среднего для данных **WORD** (тип данных **X** и **Y** – **WORD**).

### 10.2.7.9. FILTER\_W (OSCAT)



Входы ФБ:

- X (WORD) – входной сигнал;
- T (TIME) – постоянная времени фильтра.

Выходы ФБ:

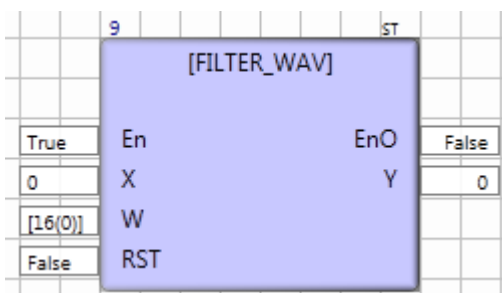
- Y (WORD) – отфильтрованный сигнал.

**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).

### 10.2.7.10. 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.

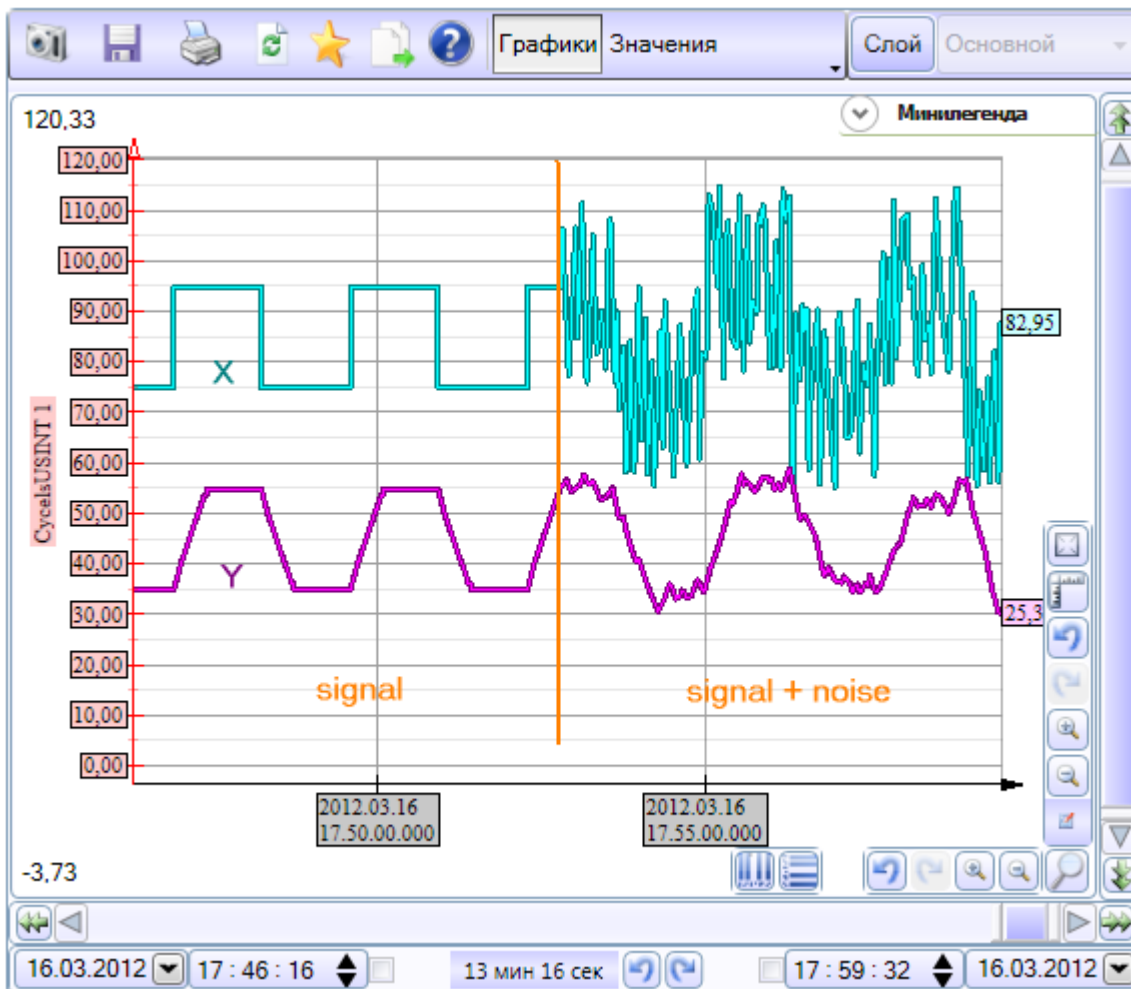
FILTER\_WAV вычисляет скользящую сумму с весом:

$$Y := X_0 * W_0 + X_1 * W_1 + .. + X_{15} * W_{15}$$

В формуле X0 – это значение входа X в текущем цикле, X1 – значение входа X в предыдущем цикле и т.д.

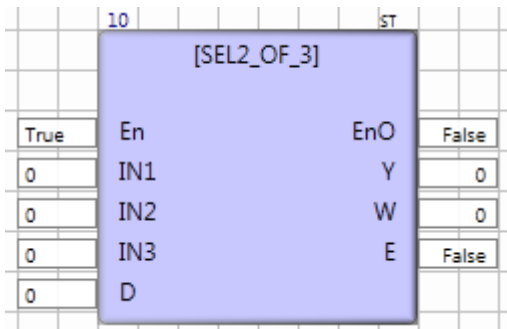
Применение FILTER\_WAV имеет смысл, если набор весовых коэффициентов определен с помощью методов, соответствующих задаче.

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).

### 10.2.7.11. 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.

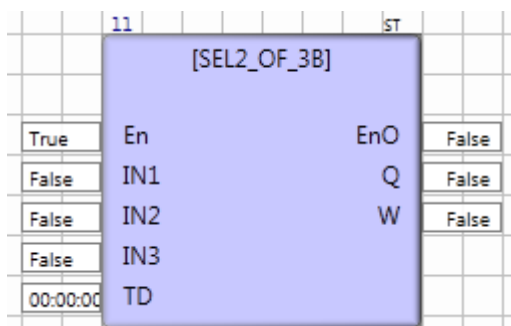
SEL2\_OF\_3 сравнивает абсолютные значения разностей (IN1-IN2), (IN1-IN3) и (IN2-IN3) с заданным отклонением D. Если, по меньшей мере, две разности меньше или равны D, выход Y равен среднему трех входов.

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.

### 10.2.7.12. SEL2\_OF\_3B (OSCAT)



Входы ФБ:

- IN1 (BOOL) – вход 1;
- IN2 (BOOL) – вход 2;
- IN2 (BOOL) – вход 3;
- TD (TIME) – задержка для выхода W.

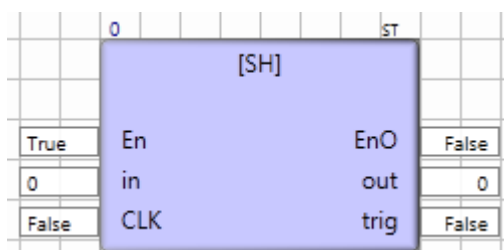
Выходы ФБ:

- 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.

### 10.2.7.13. SH (OSCAT)



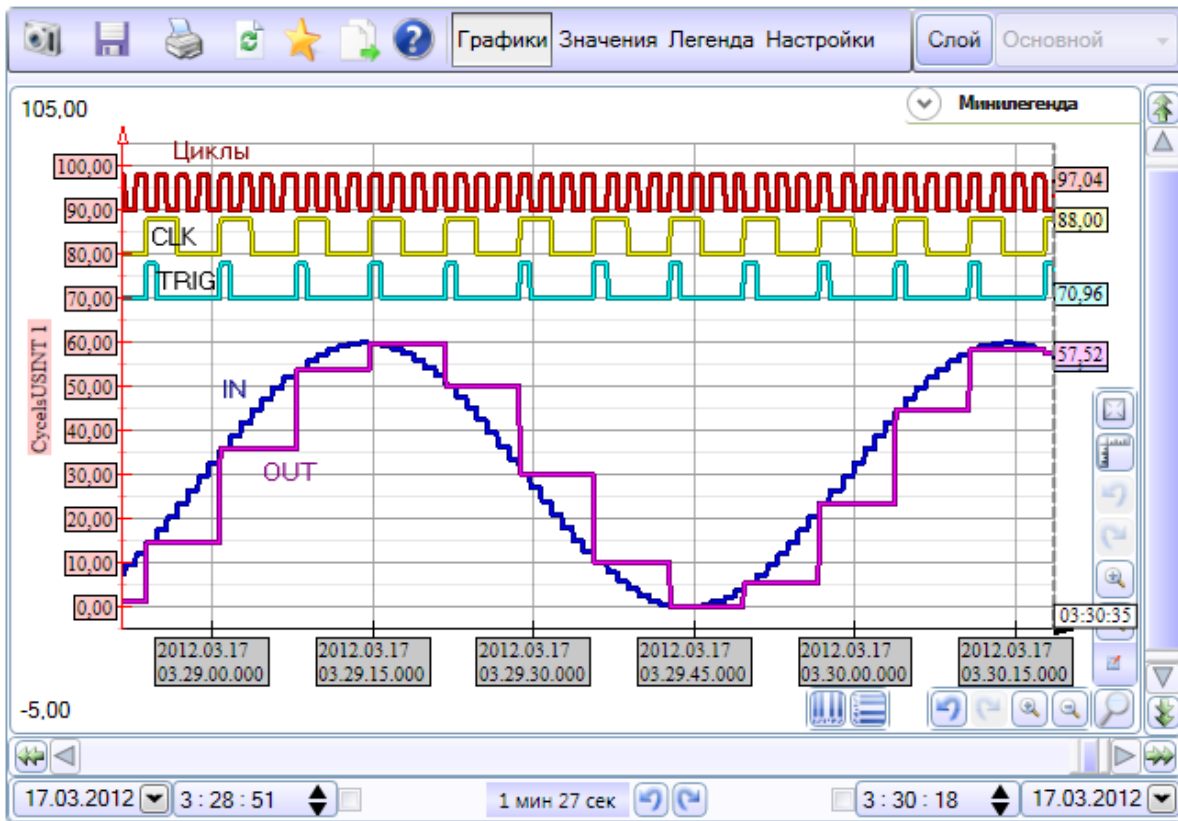
Тип данных входа IN и выхода OUT – REAL, входа CLK и TRIG – BOOL.



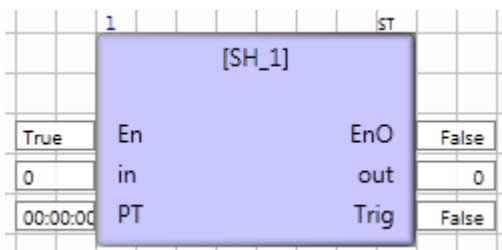
SH предназначен для выборки значений из сигнала, подаваемого на вход IN. По переднему фронту импульса на входе CLK значение IN записывается в выход OUT и сохраняется на выходе до следующего переднего фронта на CLK.

При каждой записи в OUT TRIG=TRUE в течение 1 цикла.

Следующая временная диаграмма иллюстрирует алгоритм ФБ:



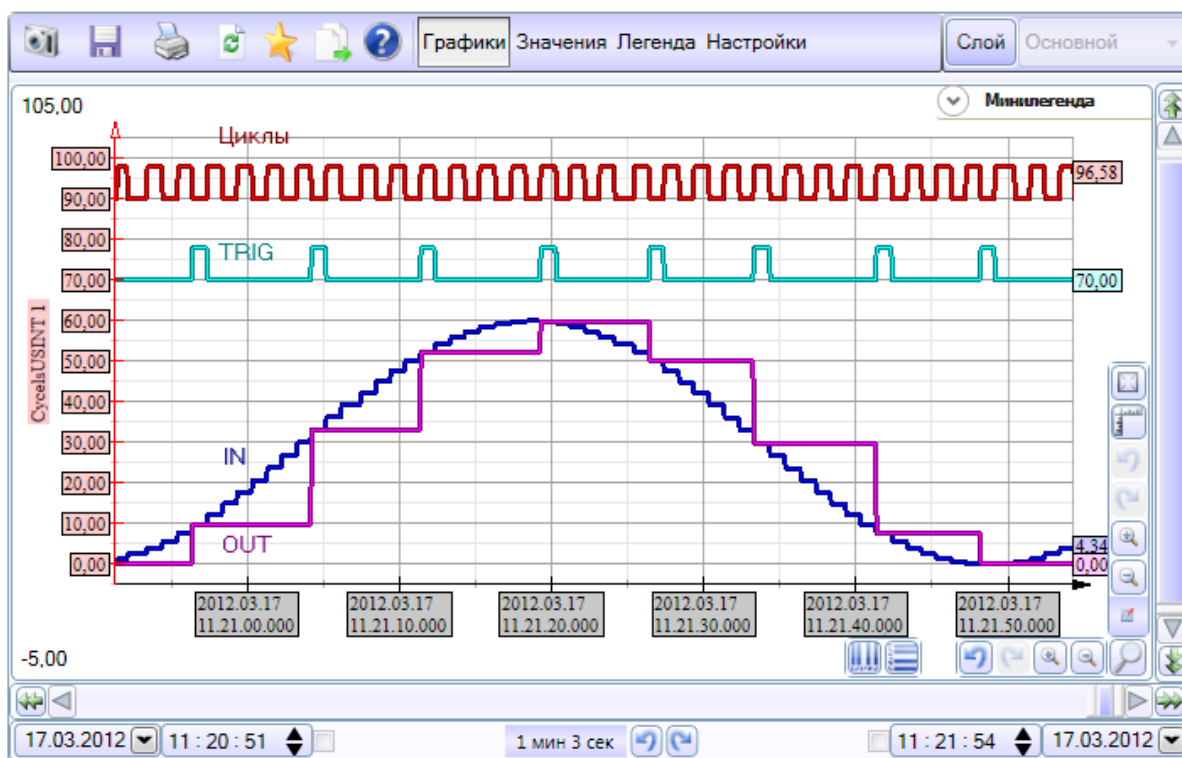
### 10.2.7.14. SH\_1 (OSCAT)



Тип данных входа IN и выхода OUT – REAL, входа PT – TIME, выхода TRIG – BOOL.

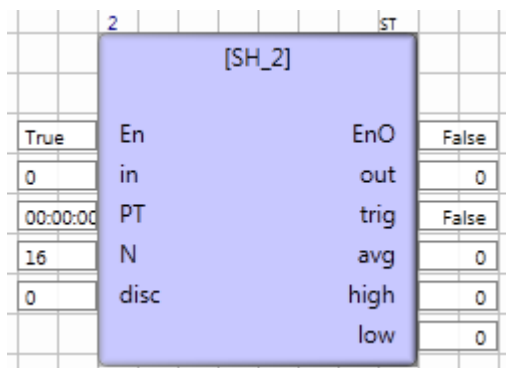
SH\_1 предназначен для периодической выборки значений из сигнала, подаваемого на вход IN. Период задается значением входа PT. Один раз за период значение IN записывается в выход OUT и сохраняется на выходе до следующей выборки. При каждой записи в OUT TRIG=TRUE в течение 1 цикла.

Следующая временная диаграмма иллюстрирует алгоритм ФБ:



Используемые функции: T\_PLC\_MS (OSCAT).

### 10.2.7.15. 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.

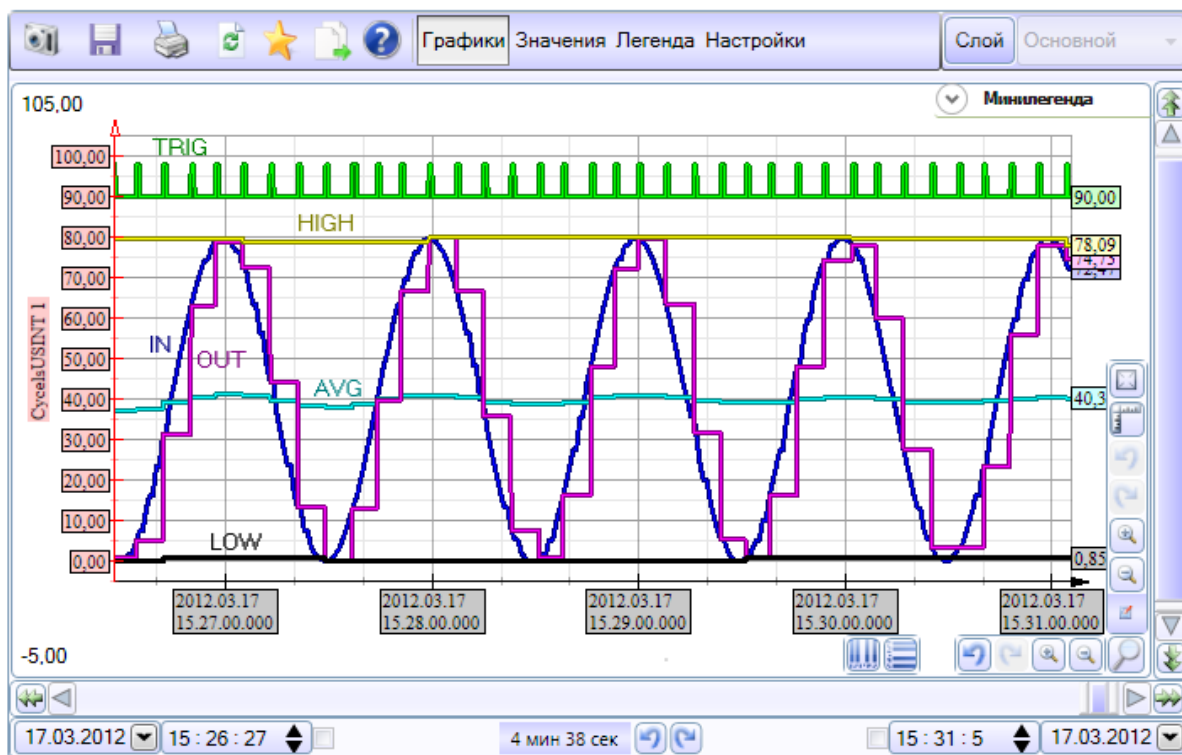
Вход N задает число последних выбранных значений (не более 16), по которым вычисляются среднее, максимальное и минимальное значения. Алгоритм вычислений зависит от входа DISC:

- 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.

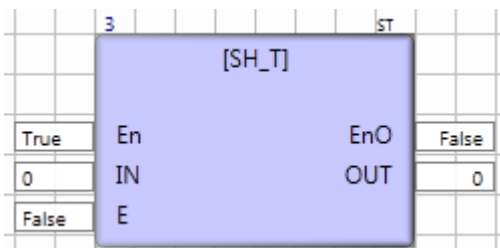
Среднее, максимальное и минимальное значения записываются соответственно в выходы AVG, HIGH и LOW.

Следующая временная диаграмма иллюстрирует работу ФБ при DISC=0:



Используемые функции: T\_PLC\_MS (OSCAT).

### 10.2.7.16. SH\_T (OSCAT)

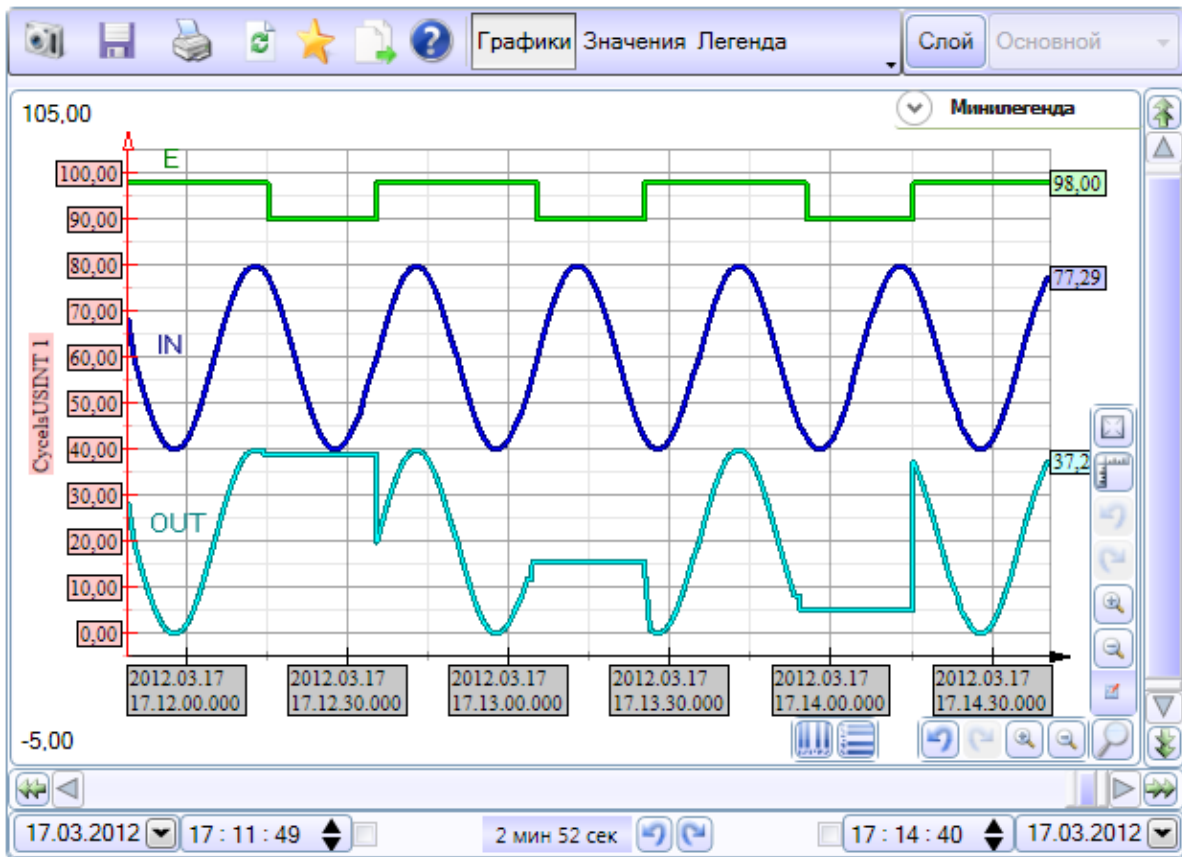


Тип данных входа IN и выхода OUT – REAL, входа E – BOOL.

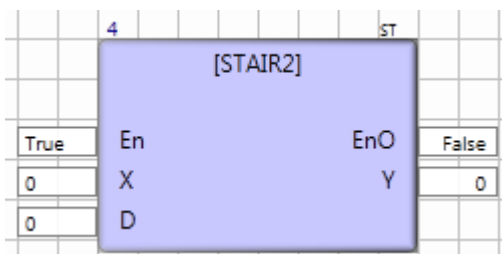
При E=TRUE SH\_T передает значения IN на выход OUT.

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:



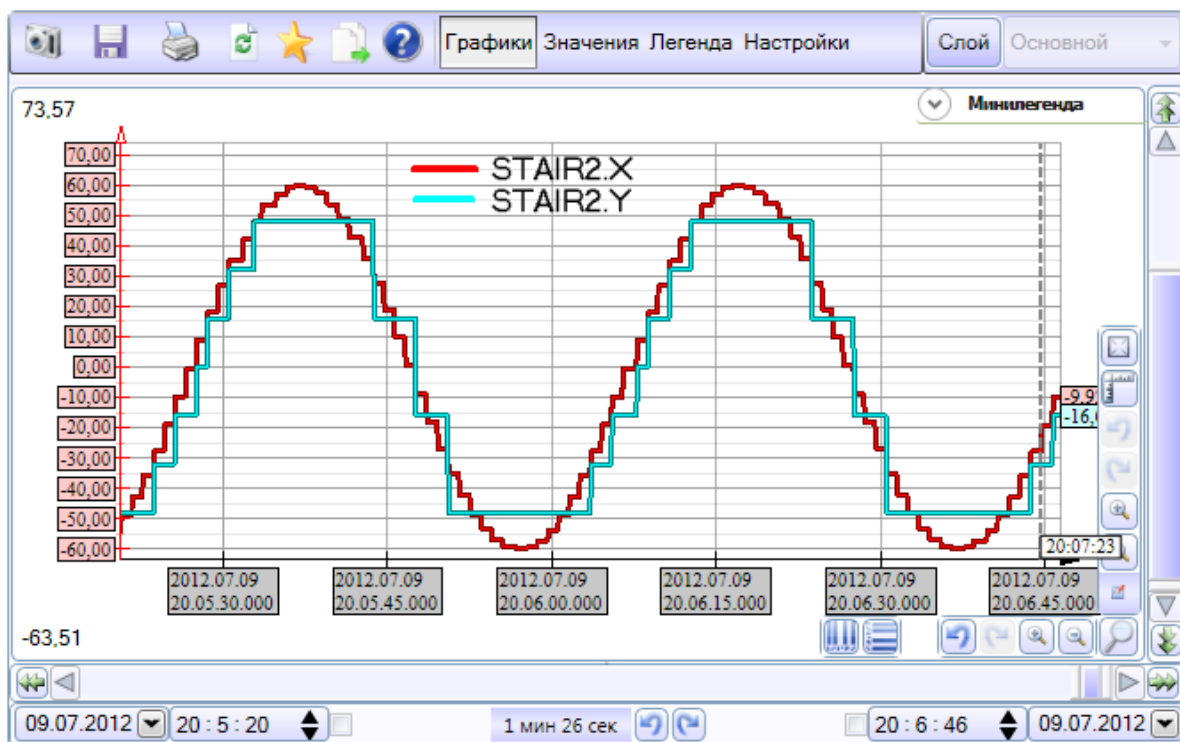
### 10.2.7.17. STAIR2 (OSCAT)



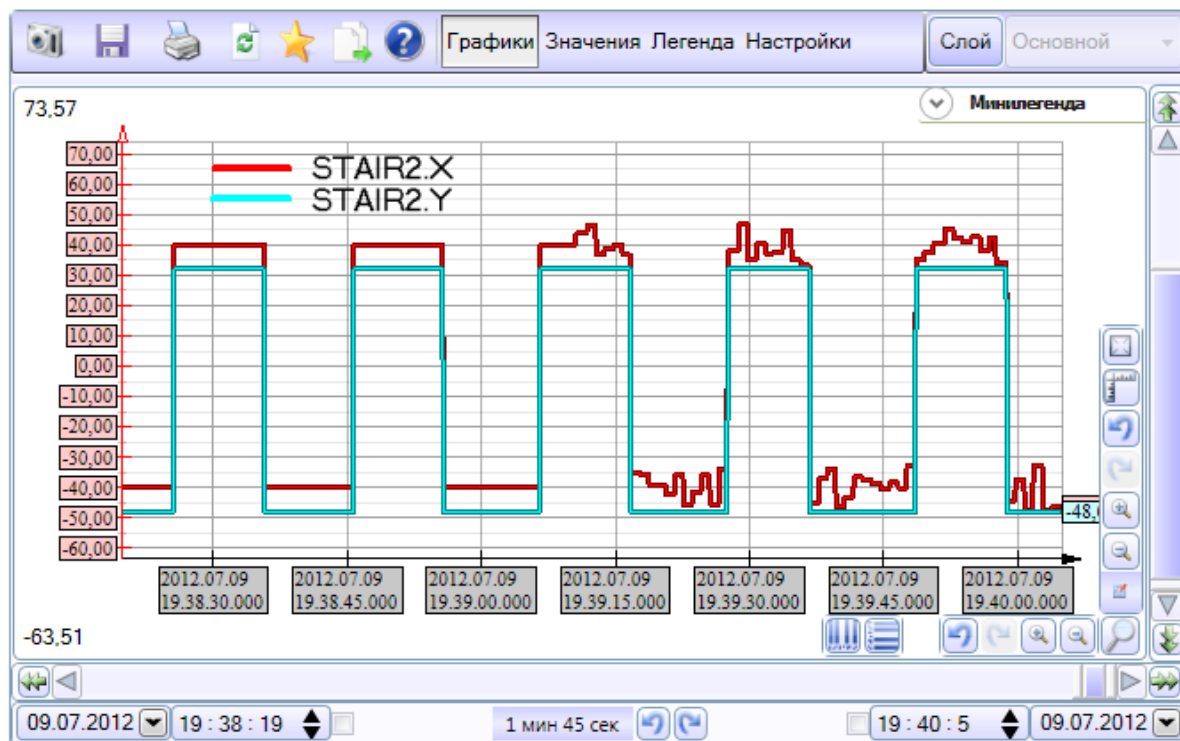
Тип данных входов и выхода функции – REAL.

Если  $D \leq 0$ ,  $Y := X$ .

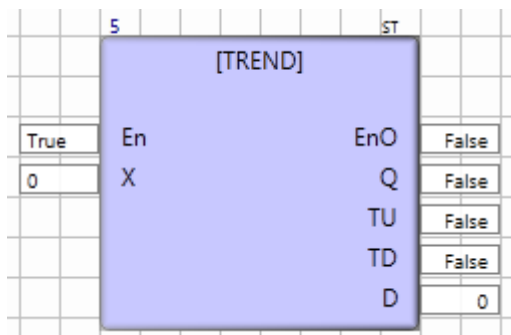
Если  $D > 0$ , ФБ преобразует гладкий сигнал, поданный на вход X, в ступенчатый (выход Q принимает только значения  $n \cdot D$ , где  $n$  – любое целое число), при этом передаточная характеристика ФБ имеет гистерезис D. Следующий график иллюстрирует работу ФБ STAIR2 на синусоидальном сигнале при  $D=16$  (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:



### 10.2.7.18. TREND (OSCAT)

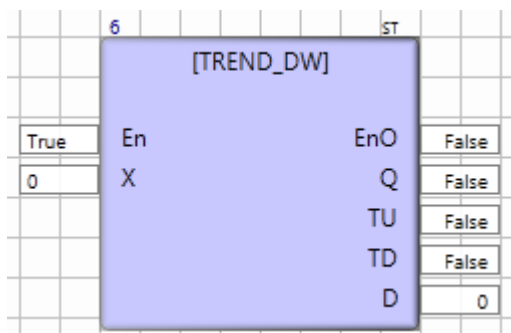


Тип данных входа X и выхода D – REAL, выходов Q, TU и TD – BOOL.

TREND сравнивает значения входа X в текущем и предыдущем циклах. По результатам равенства 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  $Q := \text{FALSE}$ .

### 10.2.7.19. TREND\_DW (OSCAT)



Тип данных входа X и выхода D – DWORD, выходов Q, TU и TD – BOOL.

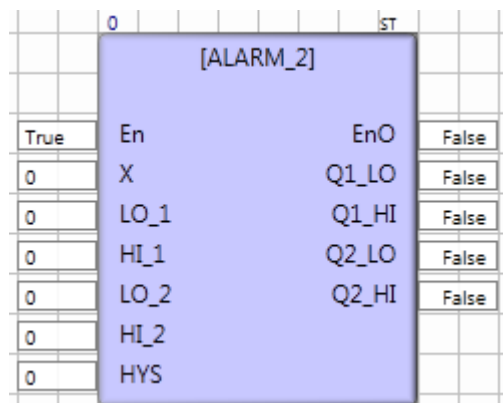
TREND\_DW сравнивает значения входа X в текущем и предыдущем циклах. По результатам равенства выходы ФБ принимают следующие значения:

- $D := \text{abs}(X_{\text{текущее}} - X_{\text{предыдущее}})$ ;
- 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  $X_{\text{текущее}} = X_{\text{предыдущее}}$ ,  $TU=FALSE$ ,  $TD=FALSE$ , а  $Q$  сохраняет предыдущее значение.

## 10.2.8. OSCAT.ФБ.Модули измерения

### 10.2.8.1. ALARM\_2 (OSCAT)

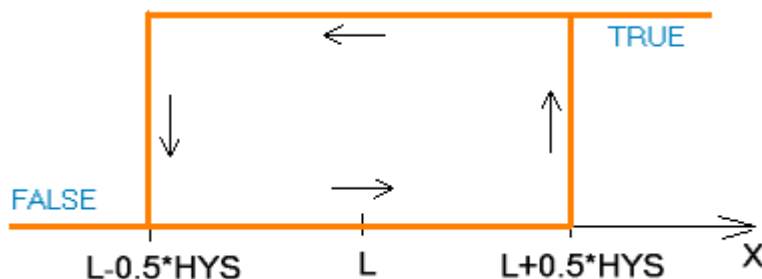


Тип данных входов – REAL, выходов – BOOL.

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  $(L+0.5*HYS)$  (здесь  $L$  – граница, соответствующая выходу);
- если выход имеет значение TRUE, то он примет значение FALSE только после того, как  $X$  станет меньше величины  $(L-0.5*HYS)$ .

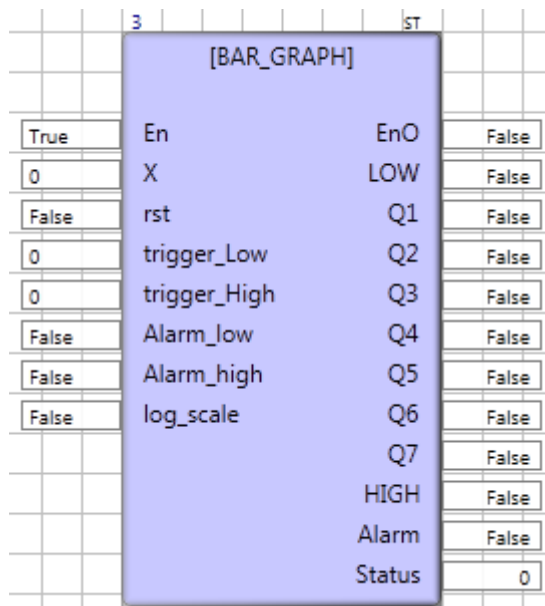
Т.е. значение  $X$  переходит границу по гистерезису:



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).



### 10.2.8.2. 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  $X < \text{TRIGGER\_LOW}$ ;
- 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).

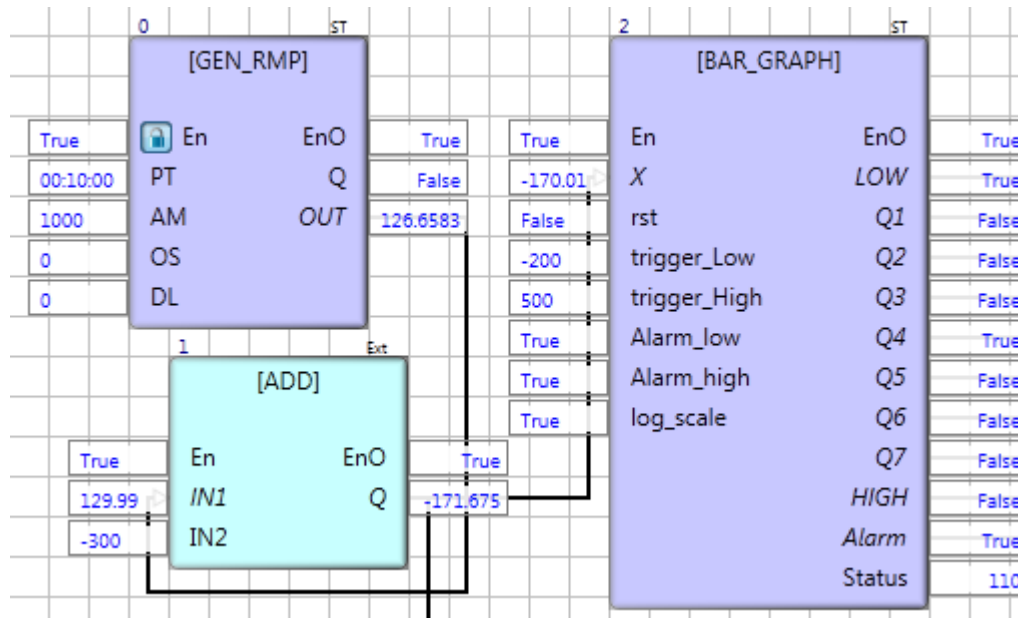
Диапазон [TRIGGER\_LOW, TRIGGER\_HIGH] разбит на 7 поддиапазонов по линейному (LOG\_SCALE=FALSE) или логарифмическому (LOG\_SCALE=TRUE) закону. Нахождение X в поддиапазоне индицирует соответствующий выход Q1..Q7 (одновременно только один из выходов Q1..Q7 может иметь значение TRUE). Выходы Q1..Q7 игнорируют команду сброса (RST=TRUE).

Выход STATUS индицирует ESR-совместимый статус:

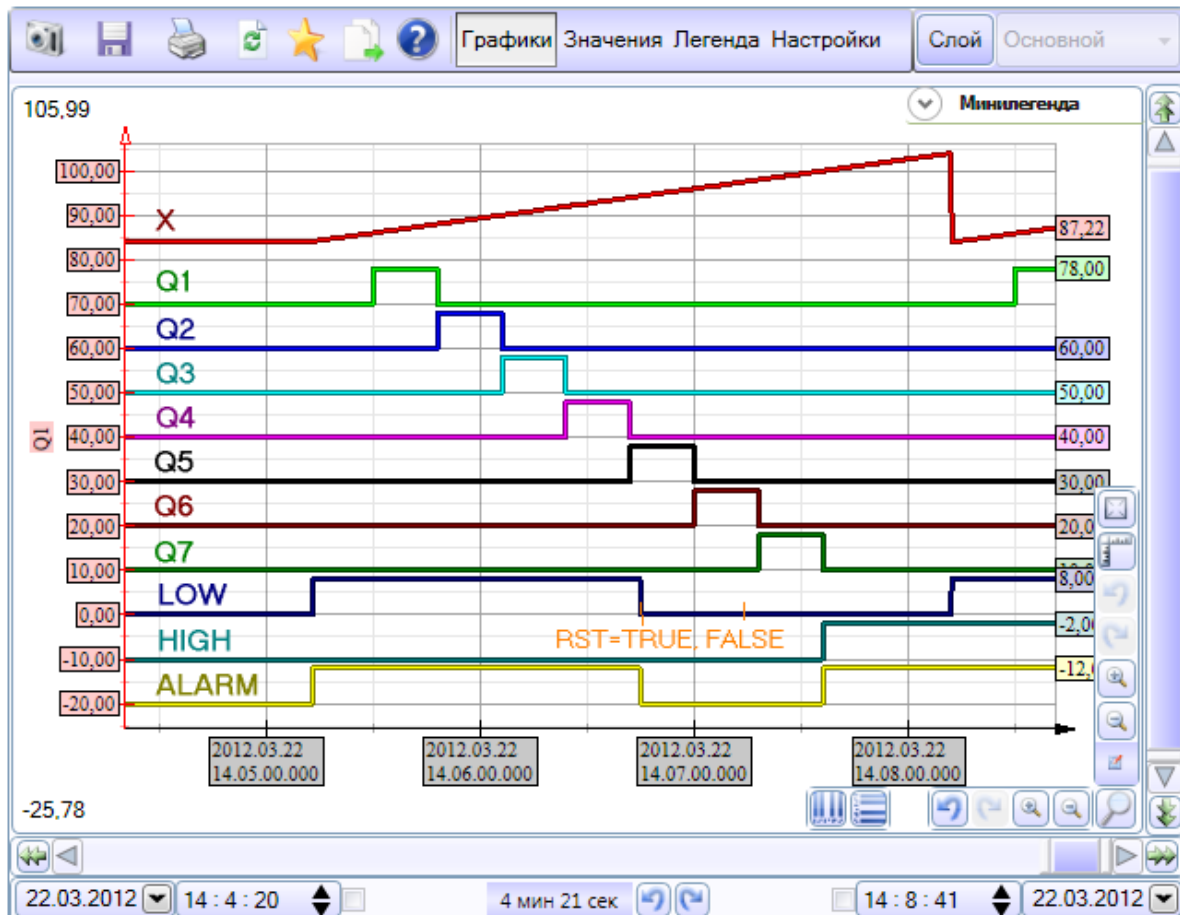
СТАТУС	
110	$\text{TRIGGER\_LOW} < X < \text{TRIGGER\_HIGH}$
111	$X < \text{TRIGGER\_LOW}$ (LOW=TRUE)
112	$X > \text{TRIGGER\_HIGH}$ (HIGH = TRUE)
1	$X < \text{TRIGGER\_LOW}$ (LOW=TRUE) и ALARM_LOW=TRUE
2	$X \geq \text{TRIGGER\_HIGH}$ (HIGH = TRUE) и ALARM_HIGH=TRUE

### Примеры

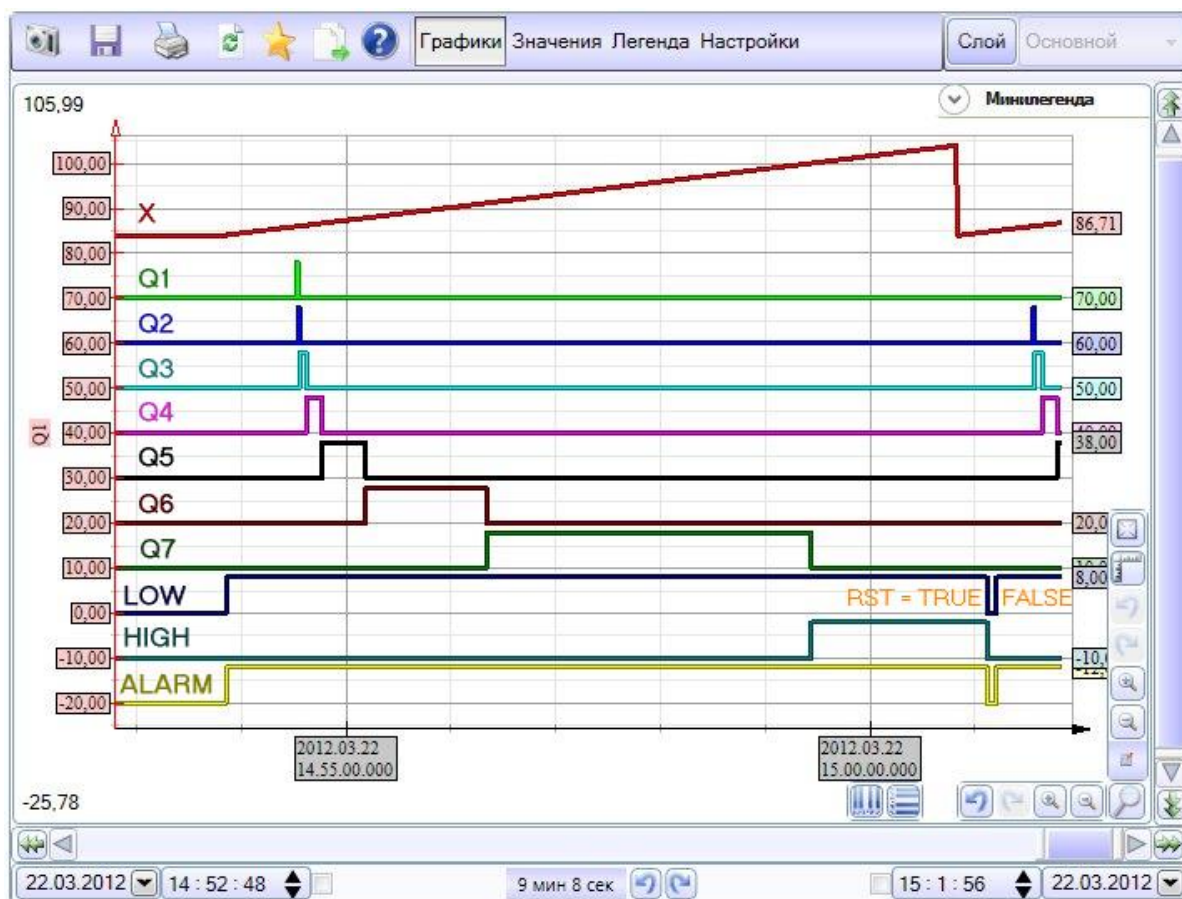
Приведенные ниже временные диаграммы ФБ BAR\_GRAPH получены с помощью следующей программы (для генерации пилообразного сигнала используется ФБ GEN\_RMP (OSCAT)):



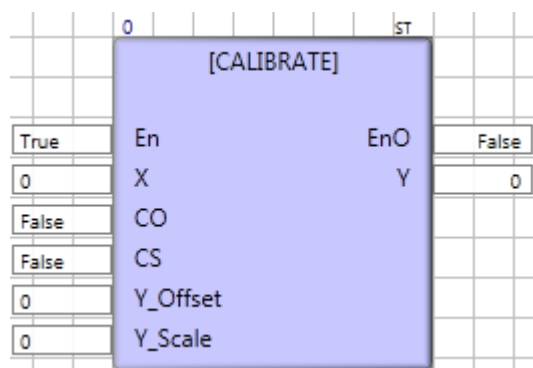
Временная диаграмма при линейном разбиении диапазона [TRIGGER\_LOW, TRIGGER\_HIGH] (LOG\_SCALE=FALSE):



Временная диаграмма при логарифмическом разбиении диапазона [TRIGGER\_LOW, TRIGGER\_HIGH] (LOG\_SCALE=TRUE):



### 10.2.8.3. 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:

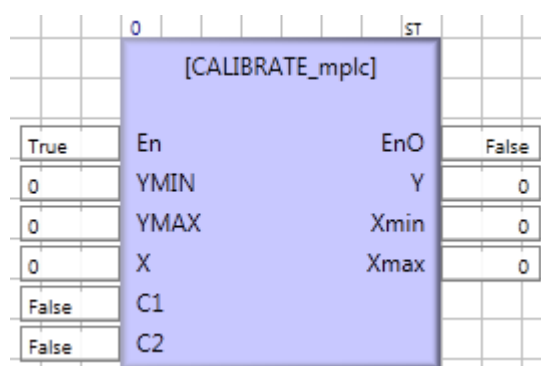
- задать Y\_OFFSET=0 и Y\_SCALE=70;

- 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  $Y\_OFFSET=0$ .*

#### 10.2.8.4. 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.

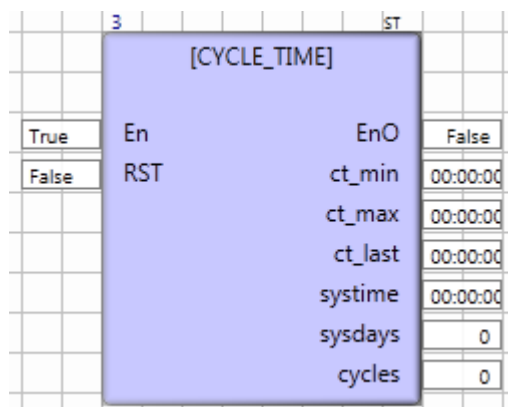
Для калибровки ФБ необходимо выполнить следующую последовательность действий, после которой ФБ автоматически переходит в рабочий режим:

- 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  $X_{max} = X_{min}$ .

### 10.2.8.5. 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.

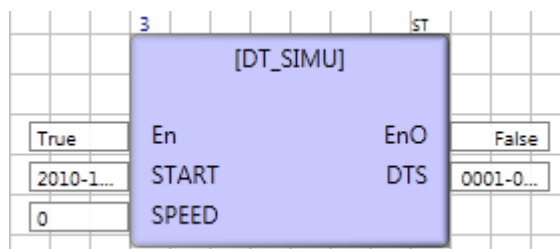
CYCLE\_TIME отслеживает реальную длительность цикла ПЛК и индицирует информацию, связанную с этим параметром, а также информацию об общем времени работы ПЛК. Число циклов с начала работы также измеряется.

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).

### 10.2.8.6. 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.

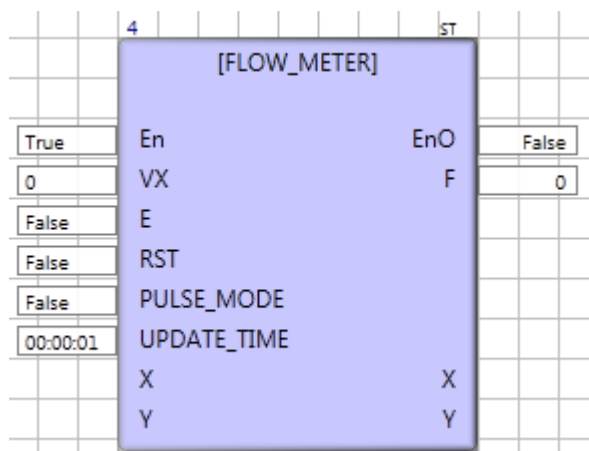
Вход START задает начальное показание симулируемых часов.



ФБ может быть использован при отладке в качестве симулятора RTC, а также для настройки скорости RTC при тестировании.

Используемые функции: T\_PLC\_MS (OSCAT).

### 10.2.8.7. 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.

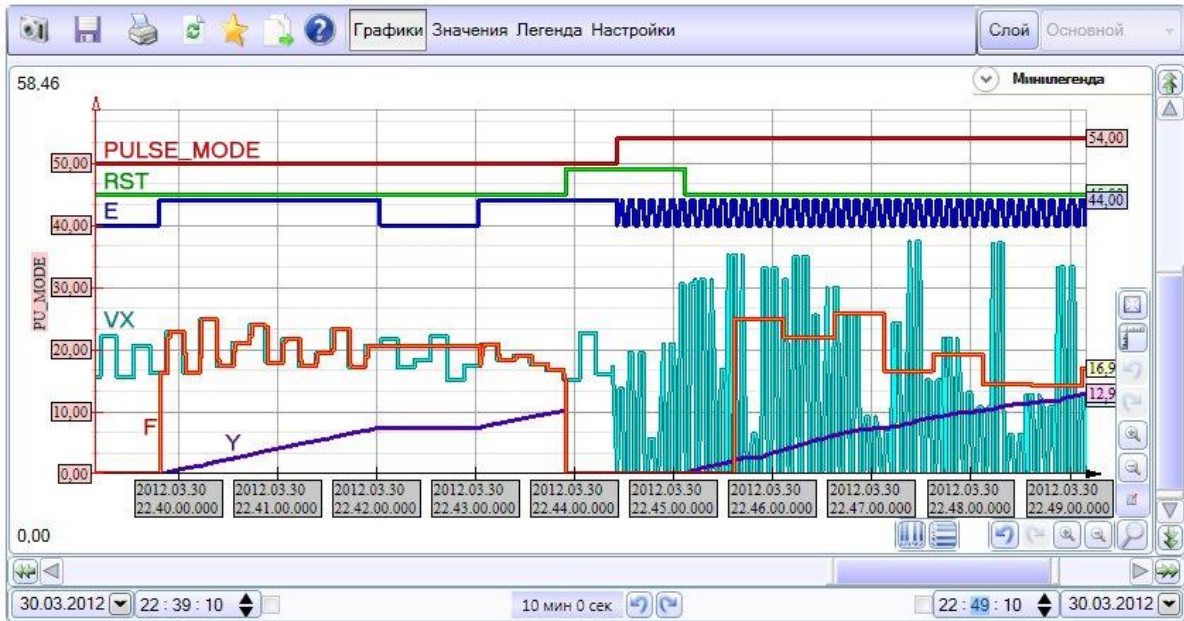
По команде E=FALSE интегрирование приостанавливается, а X и Y сохраняют свои значения, и при последующем возобновлении интегрирования (по команде E=TRUE) сохраненные значения X и Y определяют начальное значение интеграла. При E=FALSE выход F также сохраняет свое значение.

Режим 2 (PULSE\_MODE=TRUE)

Во втором режиме (PULSE\_MODE=TRUE) по переднему фронту на входе E к текущему значению расхода прибавляется значение VX.

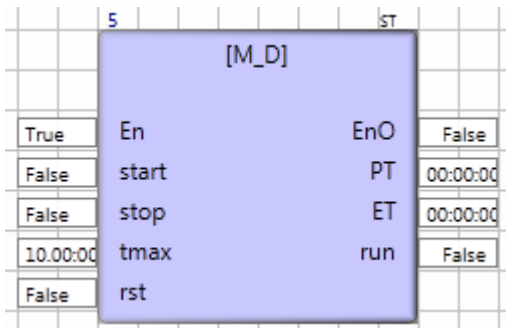
Пример

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.

### 10.2.8.8. M\_D (OSCAT)



Тип данных входов START, STOP и RST и выхода RUN – BOOL, входа TMAX и выходов PT и ET – TIME.

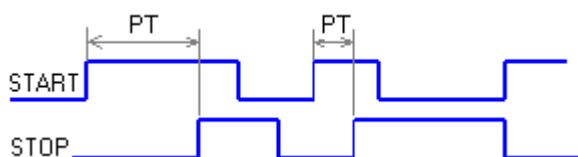
M\_D измеряет интервал времени между передним фронтом на входе START и передним фронтом на входе STOP. По окончании измерения результат выводится на выход PT. В течение измерения выход ET индицирует время, прошедшее с момента детектирования переднего фронта на входе START.

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.

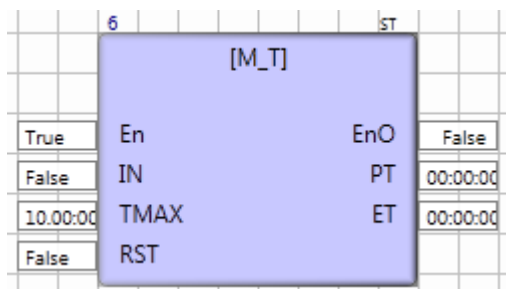
По команде RST=TRUE измерение прерывается и выходы обнуляются.

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).

### 10.2.8.9. M\_T (OSCAT)



Тип данных входов IN и RST – BOOL, входа TMAX и выходов PT и ET – TIME.

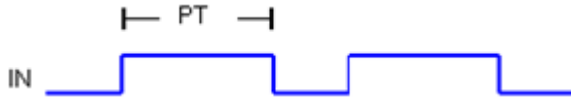
M\_T измеряет длительность импульса на входе IN (интервал времени между передним и задним фронтами). По окончании измерения результат выводится на выход PT. В течение измерения выход ET индицирует время, прошедшее с момента детектирования переднего фронта на входе IN.

Если IN=TRUE при вызове ФБ, это не рассматривается как передний фронт.

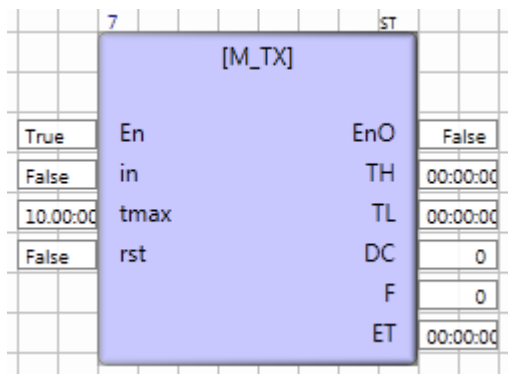
По окончании измерения выходы PT и ET сохраняют свои значения до начала нового измерения.

По команде RST=TRUE измерение прерывается и выходы обнуляются.

Измерение также прерывается и выходы обнуляются, если ЕТ достигает значения ТМАХ (ТМАХ задает максимально возможную длительность импульса). По умолчанию ТМАХ=Т#10d.



### 10.2.8.10. M\_TX (OSCAT)



Тип данных входов IN и RST – BOOL, входа ТМАХ и выходов TH, TL и ЕТ – 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 ЕТ индицирует время, прошедшее с начала текущего измерения (ЕТ сбрасывается при достижении величины (TH+TL)).

Вход ТМАХ задает максимальный измеряемый интервал (TH+TL). Значение ТМАХ по умолчанию – t#10d. Если ЕТ достигает ТМАХ, выполняется перезапуск ФБ с обнулением выходов.

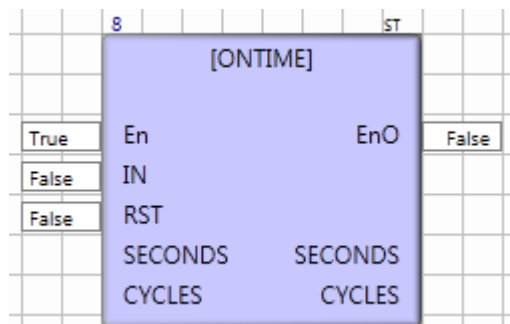
По измеренным величинам вычисляются следующие параметры сигнала:

- 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).

### 10.2.8.11. ONTIME (OSCAT)



Тип данных входов IN и RST – BOOL, входов-выходов SECONDS и CYCLES – UDINT.

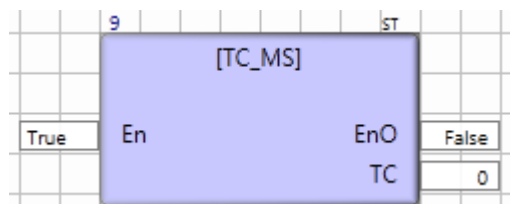
ONTIME суммирует время, в течение которого IN=TRUE. Текущее значение суммы (от 1с до 136 лет) выводится как число секунд на вход-выход SECONDS. Отсчет начинается с момента старта или с момента последнего RST=FALSE (при RST=TRUE входы-выходы равны 0).

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).

### 10.2.8.12. TC\_MS (OSCAT)

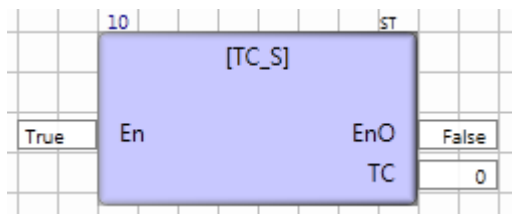


Тип данных выхода TC – DWORD.

ФБ возвращает длительность последнего цикла как целое число миллисекунд (точность – 1 миллисекунда).

Используемые функции: T\_PLC\_MS (OSCAT).

### 10.2.8.13. TC\_S (OSCAT)

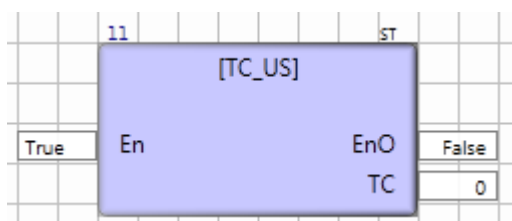


Тип данных выхода TC – REAL.

ФБ возвращает длительность последнего цикла как float-число секунд (точность – 1 миллисекунда).

Используемые функции: T\_PLC\_MS (OSCAT).

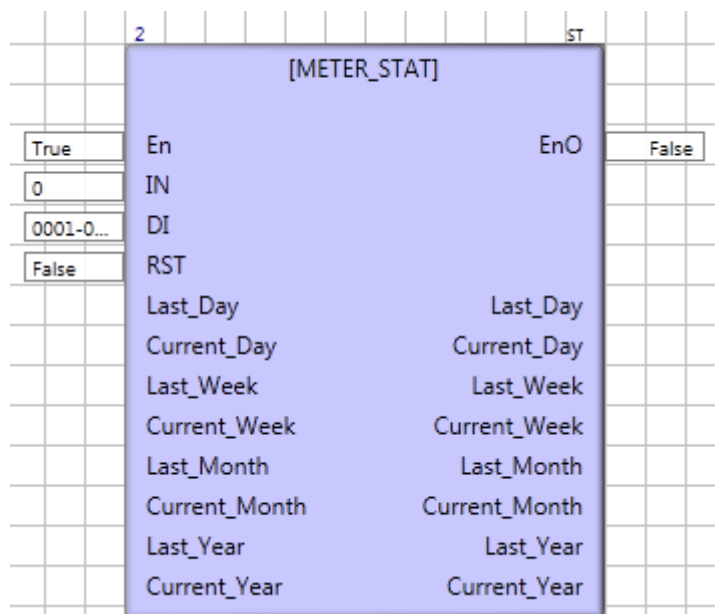
### 10.2.8.14. TC\_US (OSCAT)



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.

### 10.2.8.15. 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.

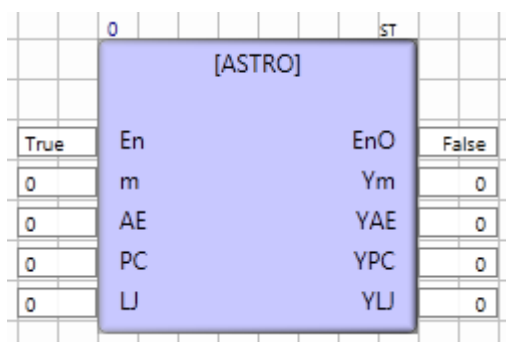
Чтобы перезапустить ФБ, нужно выполнить последовательность команд RST:=TRUE, RST:=FALSE.

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).

## 10.2.9. OSCAT.ФБ.Преобразования

### 10.2.9.1. 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.

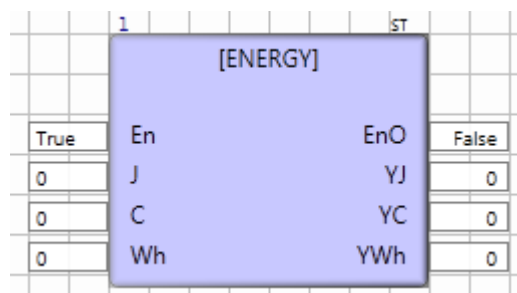
ASTRO конвертирует значения входов в а.е., суммирует полученные величины и выводит сумму, конвертированную соответствующим образом, на выходы.

$$1 \text{ а.е.} = 149,597870 * 10^9 \text{ м}$$

$$1 \text{ световой год} = 9,460530 * 10^{15} \text{ м} = 63240 \text{ а.е.} = 0,30659 \text{ парсека}$$

$$1 \text{ парсек} = 206265 \text{ а.е.} = 30,857 * 10^{15} \text{ м} = 3,261622 \text{ световых года}$$

### 10.2.9.2. ENERGY (OSCAT)



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.

$$1 \text{ Дж} = 1 \text{ Вт} \cdot \text{с} = 1 \text{ н} \cdot \text{м} \text{ (ньютон} \cdot \text{метр)}$$

$$1 \text{ кал} = 4,1868 \text{ Дж} = 1,163 * 10^{-3} \text{ Вт} \cdot \text{ч}$$

$$1 \text{ Вт} \cdot \text{ч} = 3,6 * 10^3 \text{ Дж} = 860 \text{ кал}$$

### 10.2.9.3. LENGTH (OSCAT)

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 саженях.

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

#### 10.2.9.4. PRESSURE (OSCAT)

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

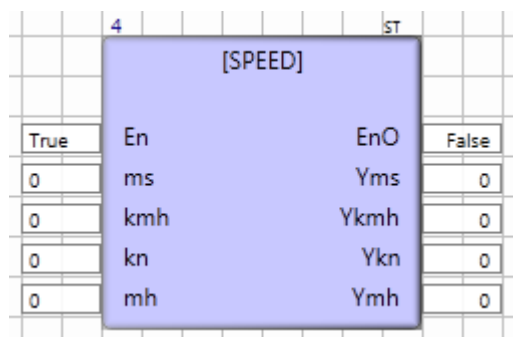
1 тех.атм = 1 кПа/см<sup>2</sup> = 0.980665 бар

1 физ.атм = 1.01325 бар

1 Па = 1 н/м<sup>2</sup>

1 бар = 105 Па

### 10.2.9.5. SPEED (OSCAT)



Входы ФБ:

- MS (REAL) – скорость в метрах в секунду (м/с);
- КМН (REAL) – скорость в километрах в час (км/ч);

- KN (REAL) – скорость в узлах (1 узел=1 мор.миля/ч);
- MH (REAL) – скорость в милях в час.

Выходы ФБ:

- YMS (REAL) – скорость в метрах в секунду (м/с);
- YKMH (REAL) – скорость в километрах в час (км/ч);
- YKN (REAL) – скорость в узлах (1 узел=1 мор.миля/ч);
- YMH (REAL) – скорость в милях в час.

ФБ конвертирует значения входов в м/с, суммирует полученные величины и выводит сумму, конвертированную соответствующим образом, на выходы.

$$1 \text{ м/с} = 3.6 \text{ км/ч}$$

$$1 \text{ узел} = 0.5144 \text{ м/с}$$

$$1 \text{ миля/ч} = 0.44704 \text{ м/с}$$

### 10.2.9.6. TEMPERATURE (OSCAT)

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.

$$T K = T 0C - 273.15$$

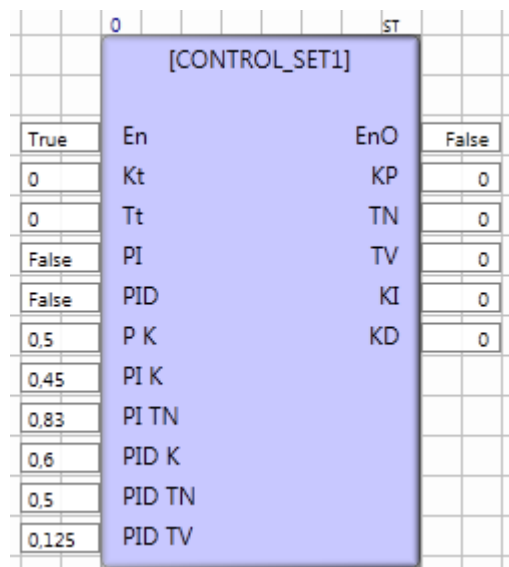
$$T 0F = 1.8 * T 0C + 32$$

$$T Re = 0.8 * T 0C$$

$$T Ra = 1.8 * T K$$

## 10.2.10. OSCAT.ФБ.Модули регулирования

### 10.2.10.1. CONTROL\_SET1 (OSCAT)



Входы ФБ:

- KT (REAL) – критический KP;
- TT (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 КР регулятора линейно увеличивается от 0 до тех пор, пока при некотором критическом КР=КТ на выходе регулятора не возникают вибрации с постоянной амплитудой и периодом ТТ. По КТ и ТТ 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.

### 10.2.10.2. CONTROL\_SET2 (OSCAT)

[CONTROL_SET2]			
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:

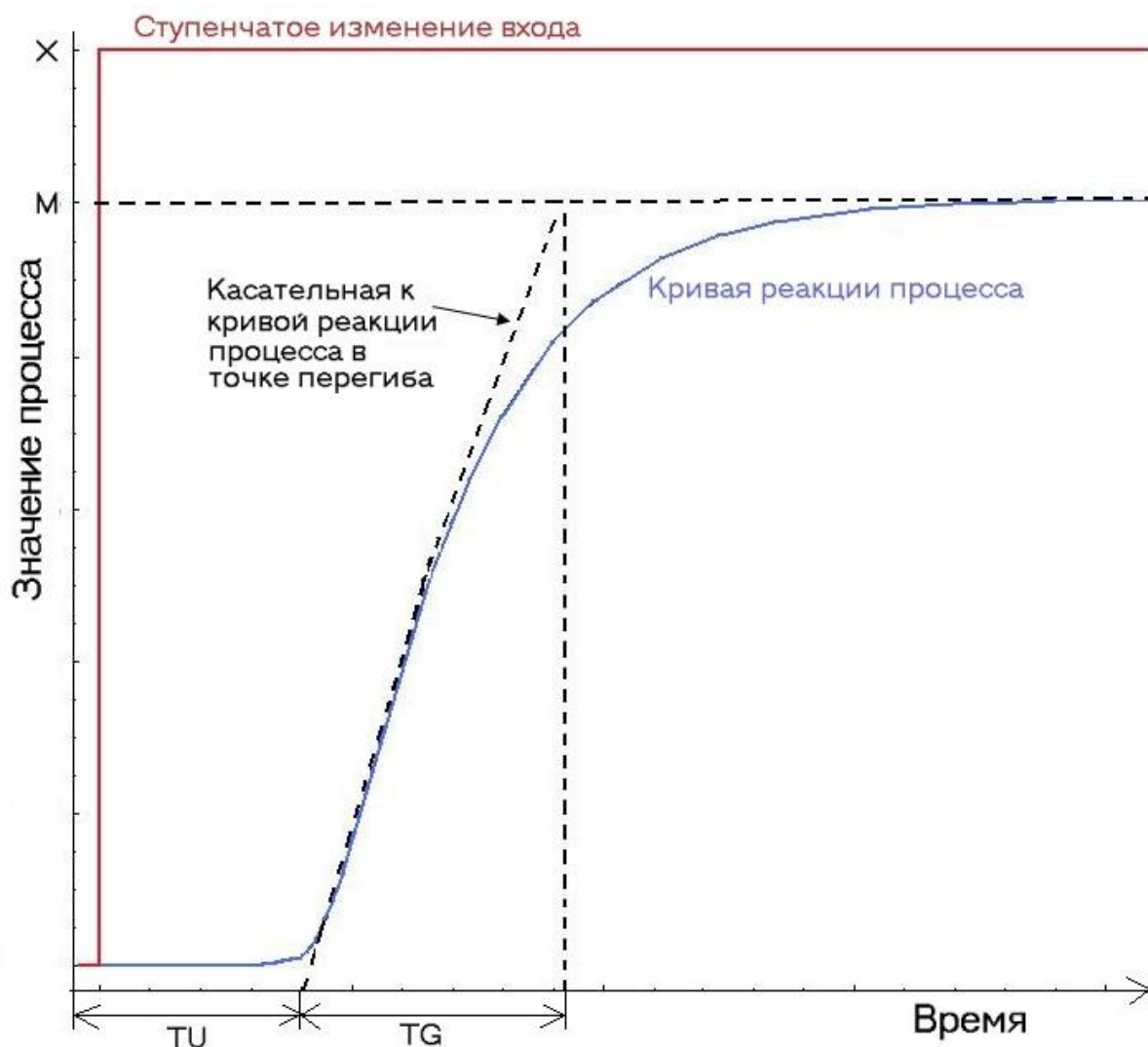
- KS (REAL), TU (REAL) и TG (REAL) – параметры, определяемые по кривой реакции процесса (см. рис. ниже);
- PI (BOOL) и PID (BOOL) – устанавливаются в соответствии с типом регулятора, для которого определяются параметры:
- П – PI=PID=0;
- ПИ – PI=1, PID=0;
- ПИД – PI=0, PID=1;
- P\_K (REAL) – константа (1 по умолчанию) для вычисления КР П-регулятора;
- PI\_K (REAL) – константа (0.9 по умолчанию) для вычисления КР 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 КР 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:

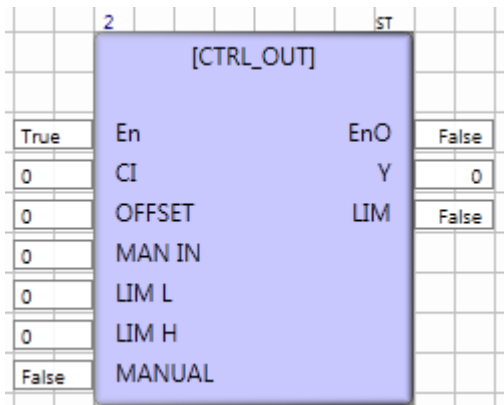
Тип регулятора	KP	TN	TV
П	$1 * TG / TU / KS$		

ПИ	$0.9 \cdot TG/TU/KS$	$3.33 \cdot TU$	
ПИД	$1.2 \cdot TG/TU/KS$	$2 \cdot TU$	$0.5 \cdot TU$

Если  $TN > 0$ , то  $KI = KP/TN$ , в противном случае  $KI = 0$ .

$KD$  вычисляется по формуле  $KD = KP \cdot TV$ .

### 10.2.10.3. 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.

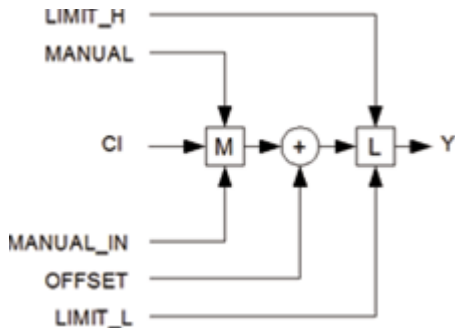
Выходы ФБ:

- $Y$  (REAL) – выходной (управляющий) сигнал;
- $LIM$  (BOOL) – этот выход принимает значение TRUE, если выходной сигнал достигает предела.

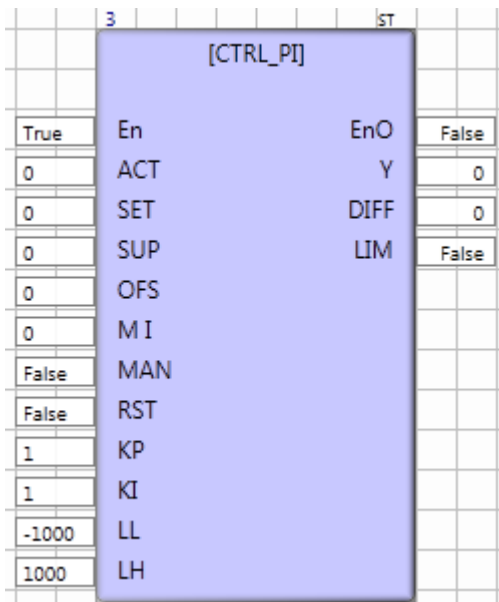
CTRL\_OUT прибавляет к значению входа CI значение OFFSET и присваивает результат выводу Y, если MANUAL = FALSE. Если MANUAL=TRUE, выводу Y присваивается значение (MAN\_IN+OFFSET).

Y всегда ограничивается значениями LIM\_L и LIM\_H. Если Y достигает одной из границ, выход LIM принимает значение TRUE.

Функциональная схема CTRL\_OUT:



#### 10.2.10.4. CTRL\_PI (OSCAT)



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<=(SET-ACT)<=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.

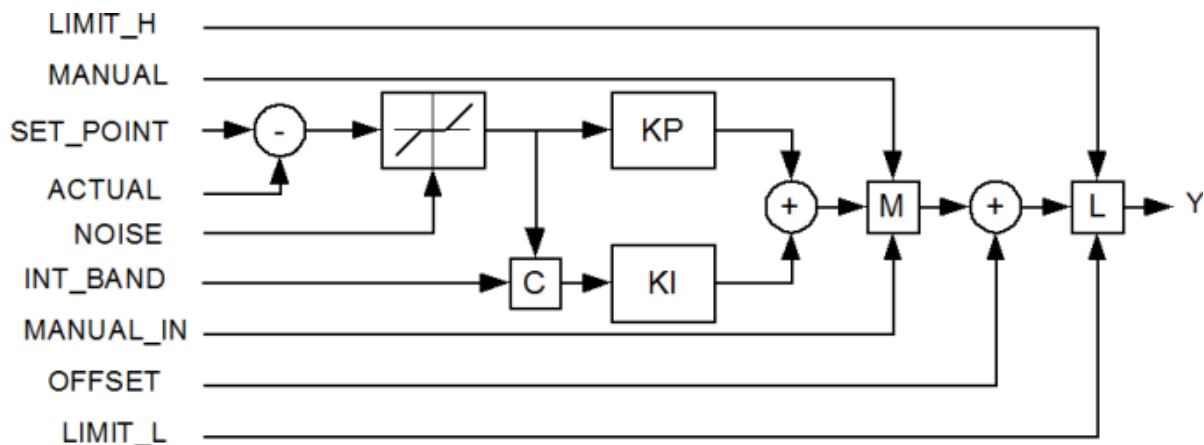
Значение выхода  $Y$  всегда ограничивается значениями  $LL$  и  $LH$ :  $LL \leq Y \leq LH$ .

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:



### 10.2.10.5. CTRL\_PID (OSCAT)

[CTRL_PID]			
True	En	EnO	False
0	ACT	Y	0
0	SET	DIFF	0
0	SUP	LIM	False
0	OFS		
0	MI		
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<=(SET-ACT)<=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**.

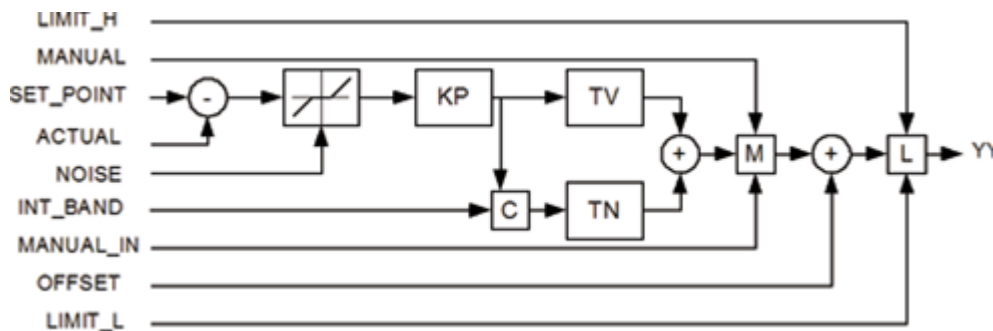
Значение выхода Y всегда ограничивается значениями LL и LH:  $LL \leq Y \leq LH$ .

Если  $TN=0$ , интегрирование приостанавливается (значение интеграла сохраняется). Для сброса значения интеграла нужно подать импульс на вход RST.

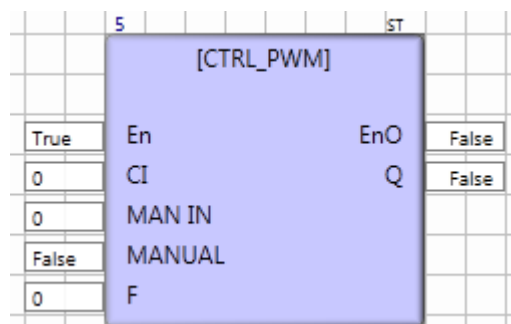
По умолчанию входы ФБ имеют следующие значения:

$KP=TN=TV=1$ ,  $LIMIT\_L=-1000$ ,  $LIMIT\_H=+1000$

Блок-схема ФБ CTRL\_PID:



### 10.2.10.6. 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.

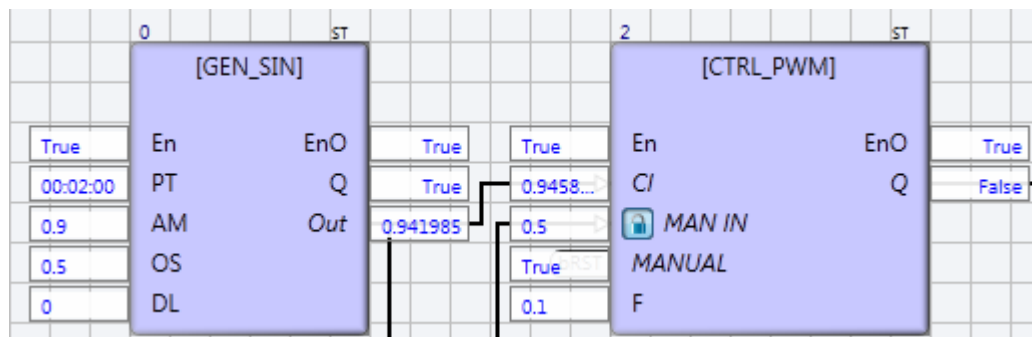
CTRL\_PWM генерирует на выходе Q широтно-модулированный импульсный сигнал с частотой следования импульсов F. Для генерации используется ФБ PWM\_DC (OSCAT).

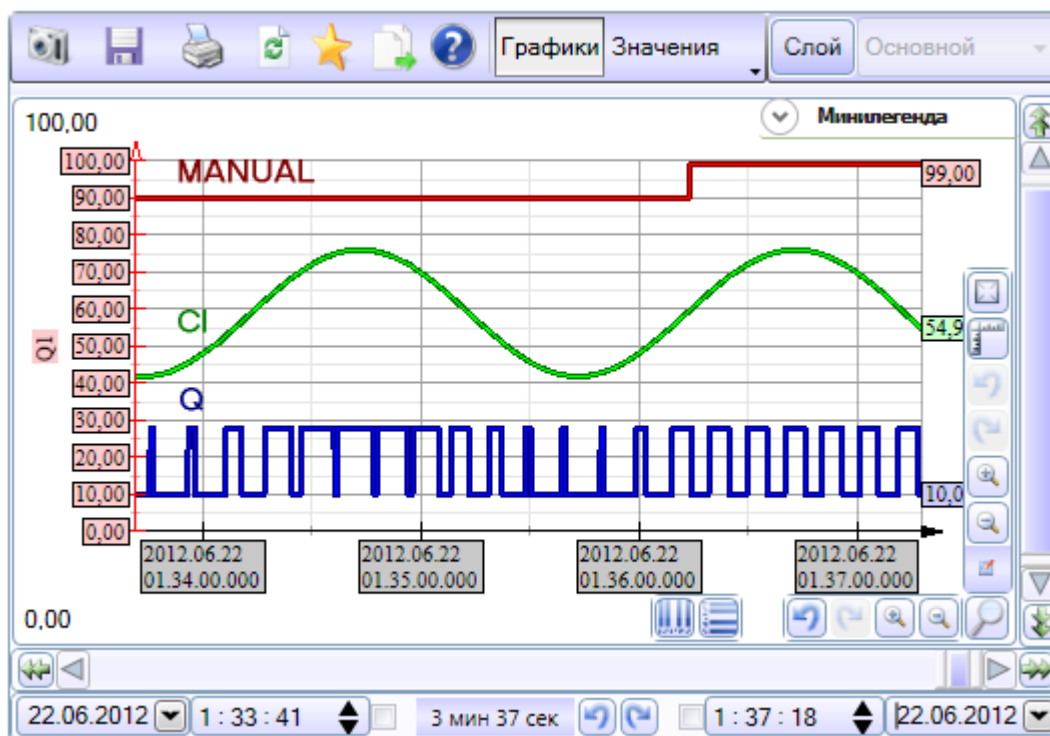
Если MANUAL=FALSE, в качестве модулирующего выбирается сигнал CI, и ширина импульсов в этом случае равна CI/F.

Если MANUAL=TRUE, в качестве модулирующего выбирается сигнал MAN\_IN, и ширина импульсов в этом случае равна MAN\_IN/F.

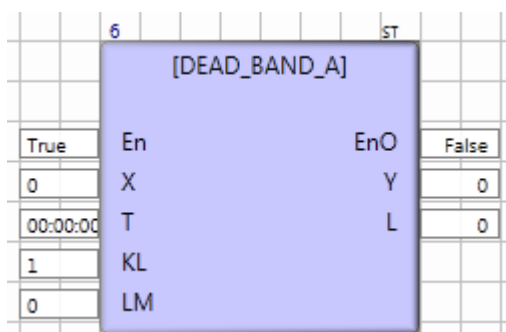
Любой из входов CI и MAN\_IN может быть использован для ручного задания ширины импульсов.

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):





### 10.2.10.7. 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 ( $L$ ).

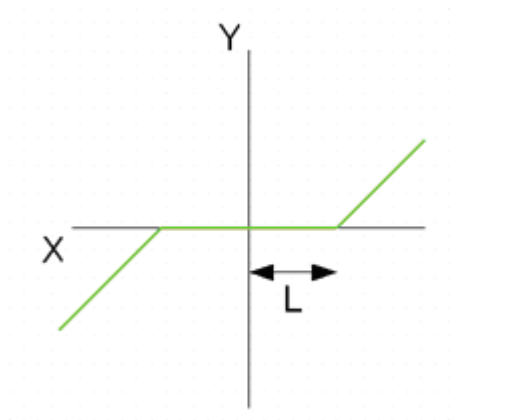
Функция смещает положительную часть кривой на  $-L$ , а отрицательную – на  $+L$ . DEAD\_BAND\_A используется для фильтрации шумов во входном сигнале – например, для предотвращения постоянного переключения контроллера при малых изменениях сигнала, что приводит к перегрузкам и преждевременному старению исполнительного механизма.

Величина  $L$  вычисляется из амплитуды высокочастотной составляющей входного сигнала  $X$ :

$$L = \langle \text{амплитуда ВЧ} \rangle * KL$$

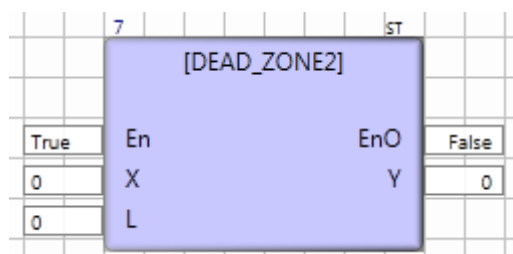
Для выделения ВЧ-составляющей используется фильтр с постоянной времени  $T$ . Для изменения чувствительности алгоритма используется множитель  $KL$ . Значение  $KL$  по умолчанию – 1, допустимые значения  $KL$  – от 1 до 5.

Поскольку алгоритм должен оставаться работоспособным даже в экстремальных условиях, величина  $L$  ограничивается значением входа  $LM$ .



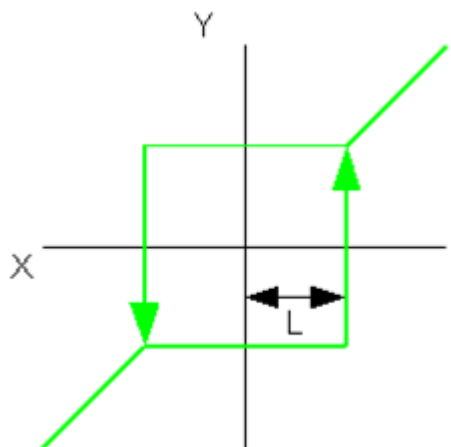
Используемые функции: FT\_PT1 (OSCAT).

### 10.2.10.8. DEAD\_ZONE2 (OSCAT)

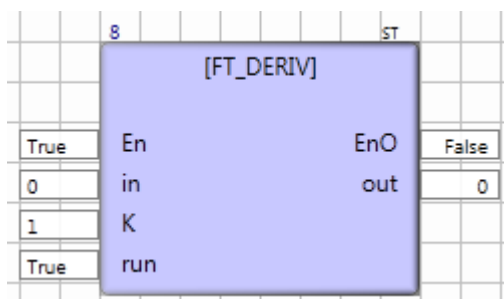


Тип данных выходов X, L и выхода Y – REAL.

**DEAD\_ZONE2** – ударная линейная передаточная функция с зоной нечувствительности и гистерезисом:



### 10.2.10.9. 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:

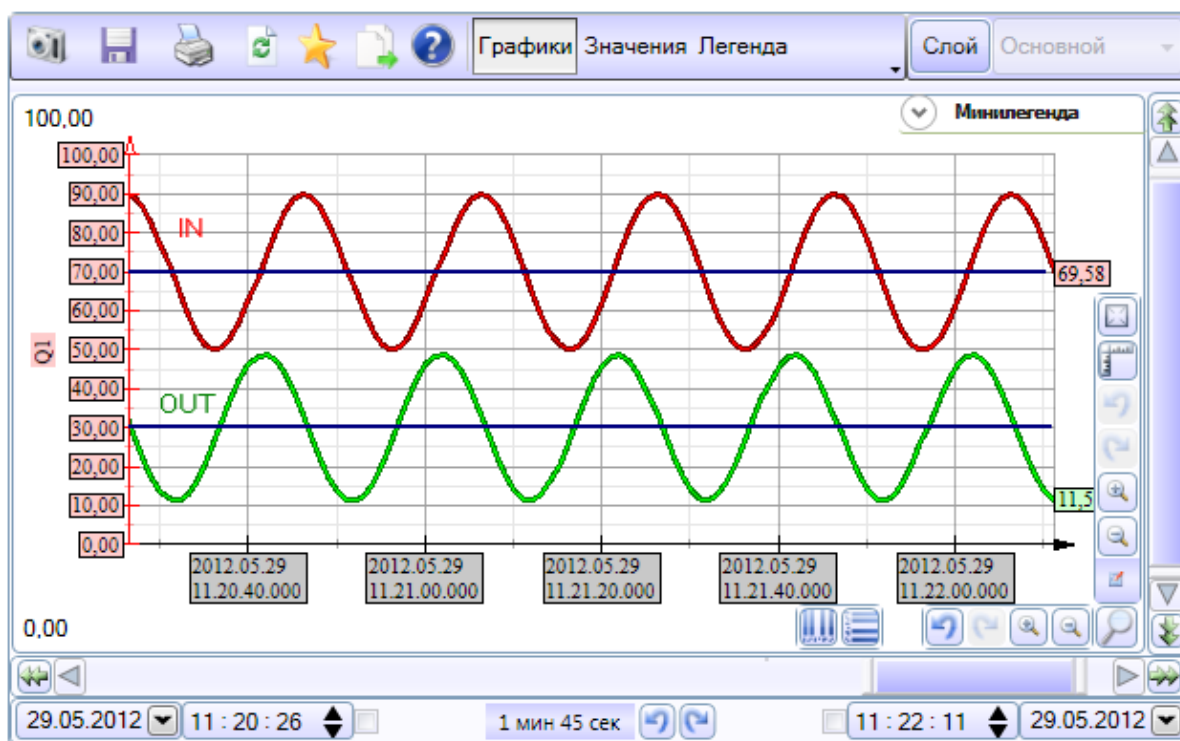
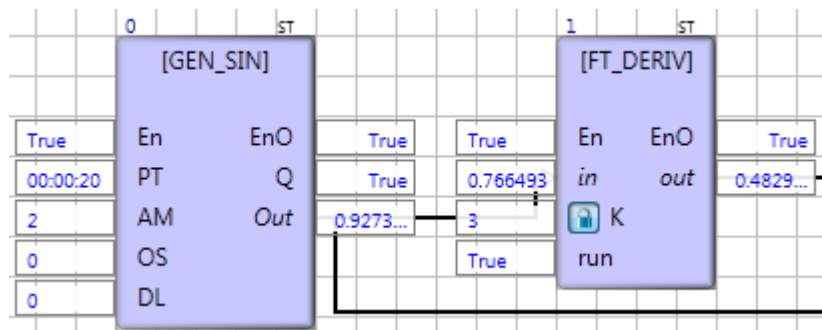
$$\text{out}(t) := K * \text{IN}'(t)$$

При K=1 размерность выхода равна [ $\langle$ размерность IN $\rangle$ /с]

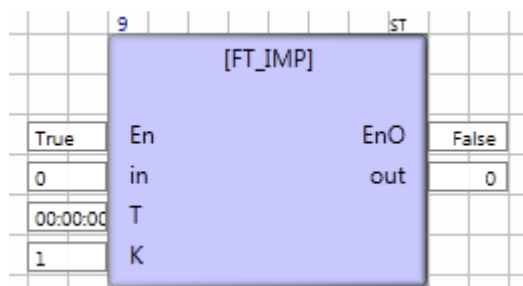
Если RUN=FALSE, OUT:=0.

Используемые функции: T\_PLC\_MS (OSCAT).

Пример



### 10.2.10.10. FT\_IMP (OSCAT)



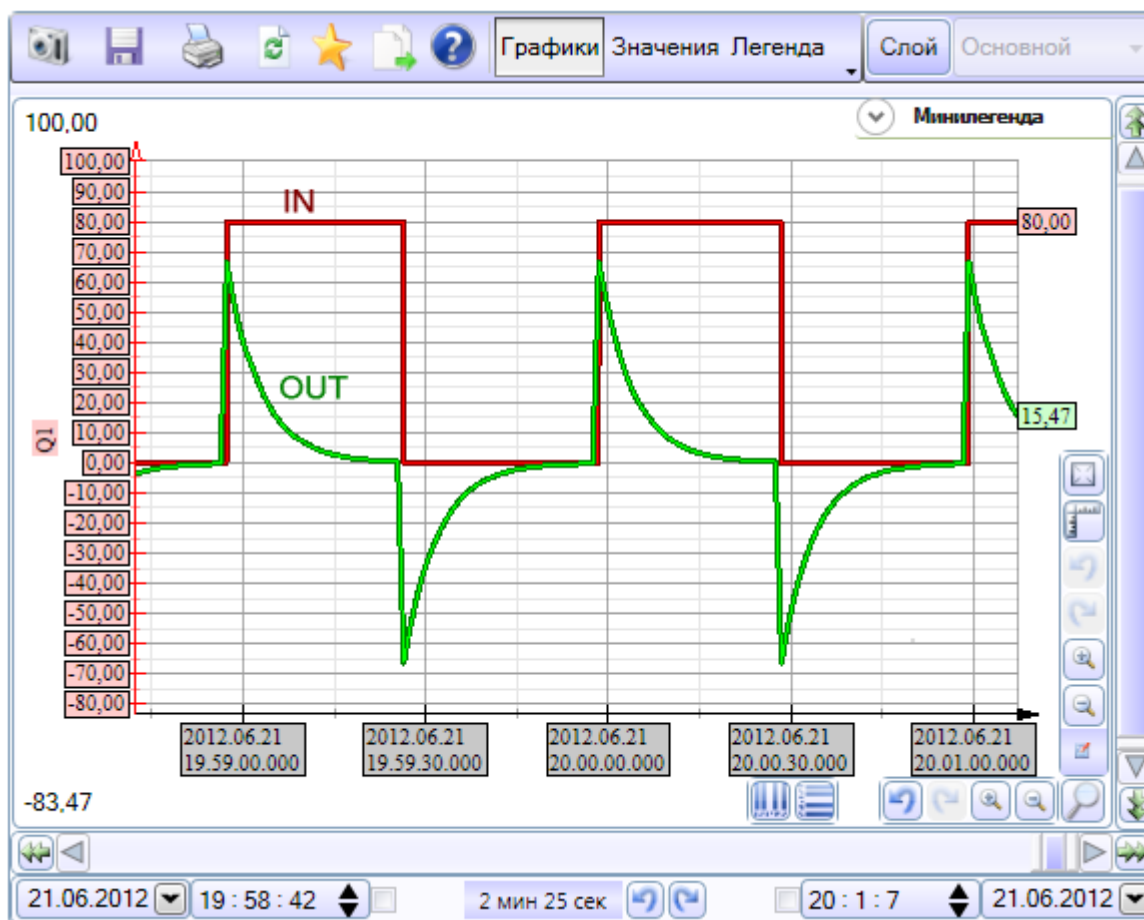
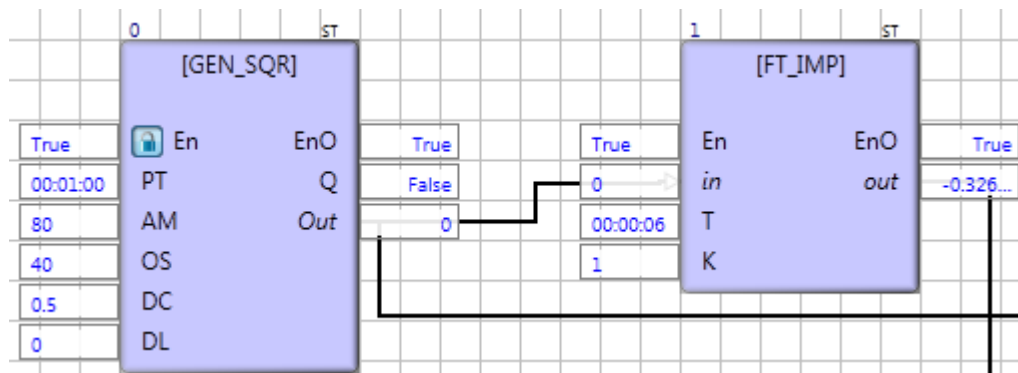
Тип данных входов IN, K и выхода OUT – REAL, выхода OUT – TIME.

Импульсный (высокочастотный) фильтр FT\_IMP работает по следующему алгоритму:

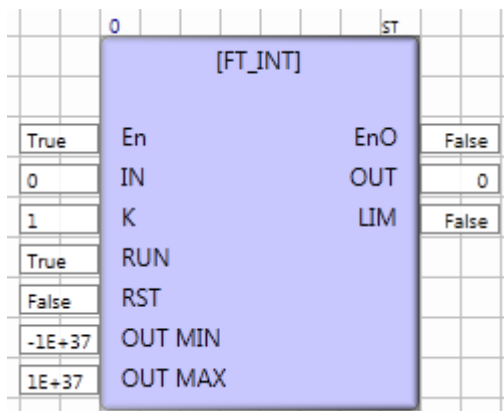
- входной сигнал IN фильтруется с помощью ФБ FT\_PT1 (OSCAT);

- 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)):



### 10.2.10.11. 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.

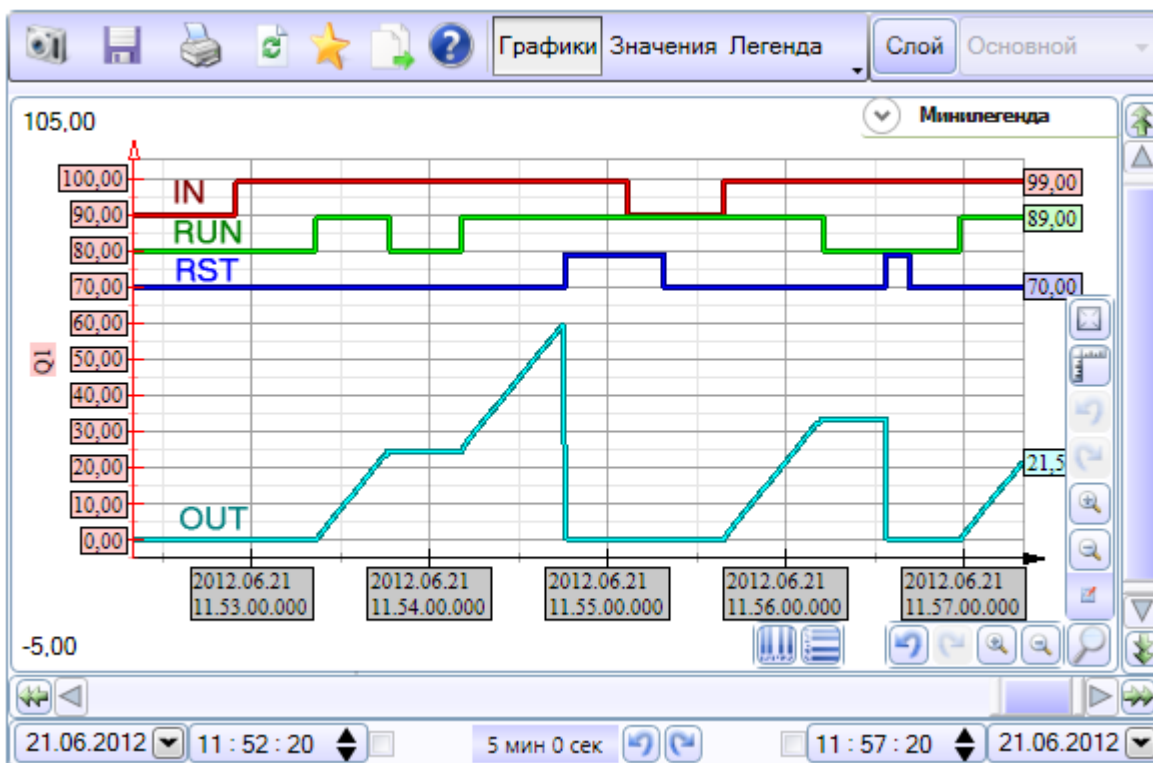
FT\_INT интегрирует входной сигнал (IN) с помощью ФБ INTEGRATE (OSCAT).

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  $RUN:=TRUE$ .

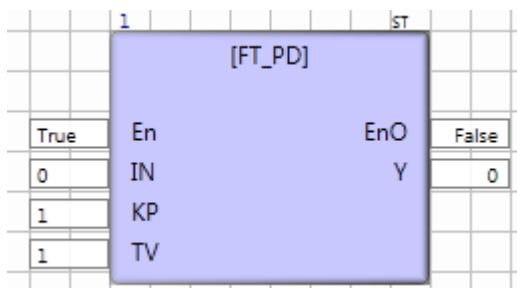


Следующая временная диаграмма ФБ FT\_INT демонстрирует управление интегрированием с помощью RUN и RST:



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.

### 10.2.10.12. 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.

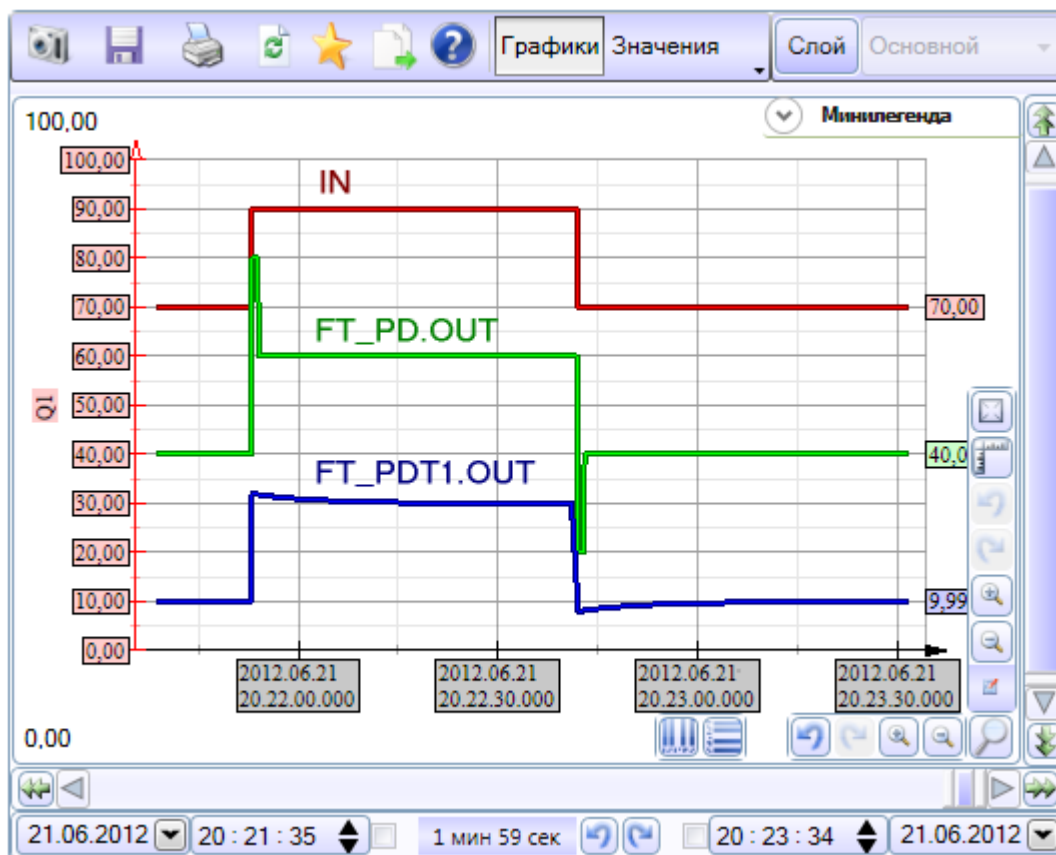
Пропорционально-дифференциальный регулятор (ПД-регулятор) FT\_PD работает по следующей формуле:

$$Y := KP * (IN + FT\_DERIV(IN, TV))$$

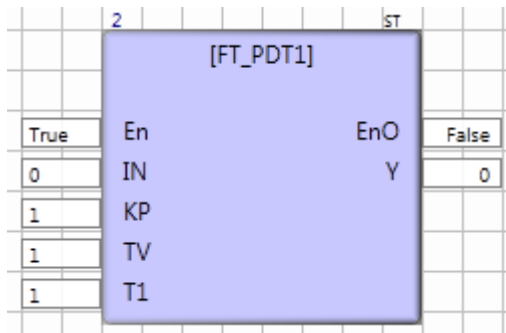
Т.е. коэффициент пропорциональной части равен KP, а коэффициент при производной – KP\*TV.

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:



### 10.2.10.13. 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:

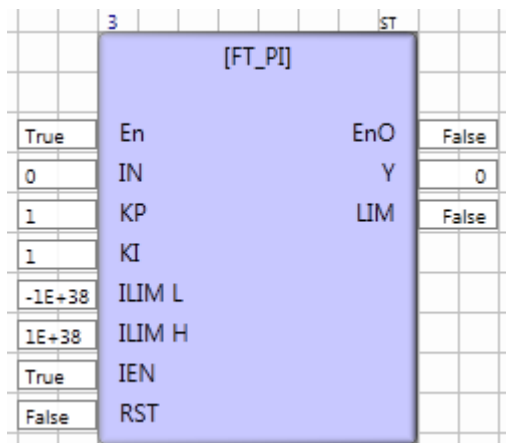
$$Y := KP * (IN + FT\_PT1(FT\_DERIV(IN,TV), T1))$$

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).

### 10.2.10.14. 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.

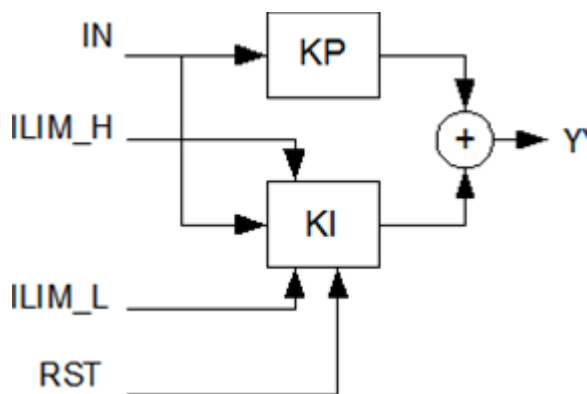
Выходы ФБ:

- Y (REAL) – выходной сигнал;
- LIM (BOOL) – TRUE, если значение интегратора достигает границы.

Пропорционально-интегральный регулятор (ПИИ-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).

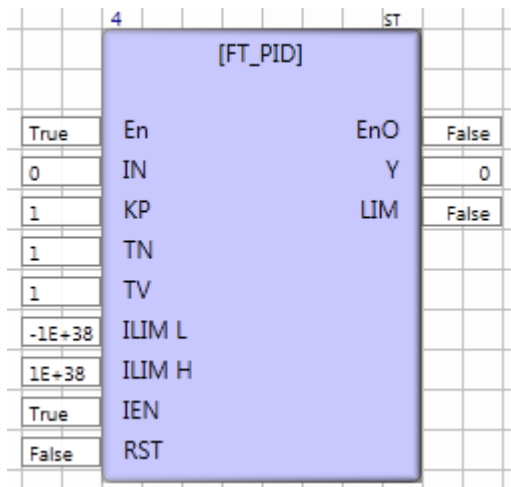
Для реинициализации интегратора нужно последовательно выполнить команды RST:=TRUE и RST:=FALSE – после этого начальное значение интеграла равно 0, а интегрирование начнется при IEN:=TRUE.

На вход IN ФБ FT\_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=1; KI=1; ILIM\_L=-1E38; ILIM\_H=+1E38; IEN=TRUE

### 10.2.10.15. 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);
- ILIM\_L (REAL) – нижний предел интегратора;
- ILIM\_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)):

$$Y := KP * [IN + FT\_DERIV(IN, TV) + FT\_INT(IN, 1/TN, IEN, RST, ILIM\_L, ILIM\_H)]$$

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.

### 10.2.10.16. FT\_PIDW (OSCAT)

5		ST	
[FT_PIDW]			
True	En	EnO	False
0	IN	Y	0
1	KP	LIM	False
1	TN		
1	TV		
-1E+38	LIM L		
1E+38	LIM H		
False	RST		

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 ( $KD$ ) –  $KP \cdot TV$ , 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  $TN=0$ .

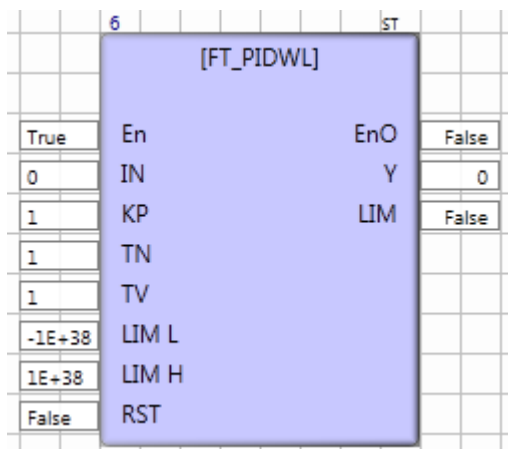
Значение выхода  $Y$  ограничивается значениями  $LIM\_L$  и  $LIM\_H$ :  $LIM\_L \leq Y \leq LIM\_H$ .

На вход  $IN$  ФБ  $FT\_PIDW$  подается выходной сигнал функции  $CTRL\_IN$  ( $OSCAT$ ), вычисляющей девиацию процесса.

По умолчанию входы ФБ имеют следующие значения:

$KP=TN=TV=1$ ,  $LIM\_L=-1E38$ ;  $LIM\_H=+1E38$

### 10.2.10.17. 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.

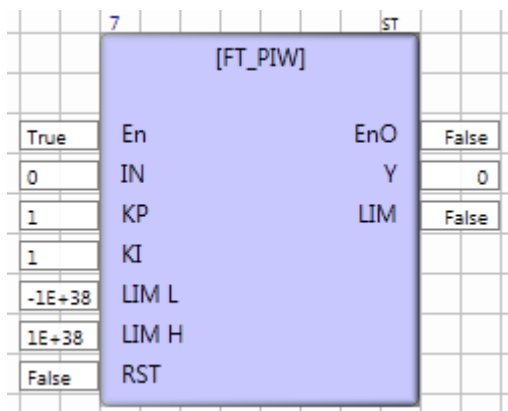
Значение выхода Y ограничивается значениями LIM\_L и LIM\_H:  $LIM\_L \leq Y \leq LIM\_H$ .

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

### 10.2.10.18. 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.

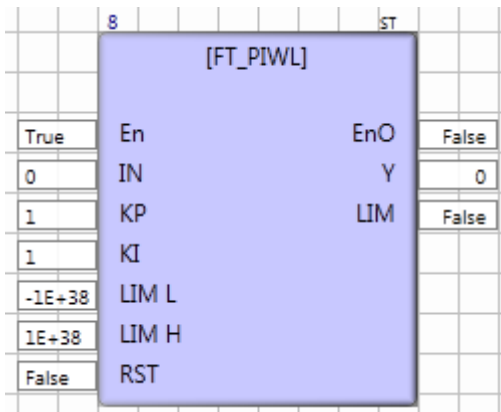
На вход IN ФБ FT\_PI подается выходной сигнал функции CTRL\_IN (OSCAT), вычисляющей девиацию процесса.

Значение выхода Y ограничивается значениями LIM\_L и LIM\_H:  $LIM\_L \leq Y \leq LIM\_H$ .

По умолчанию входы ФБ имеют следующие значения:

$$KP=1; KI=1; LIM\_L=-1E38; LIM\_H=+1E38$$

### 10.2.10.19. 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).

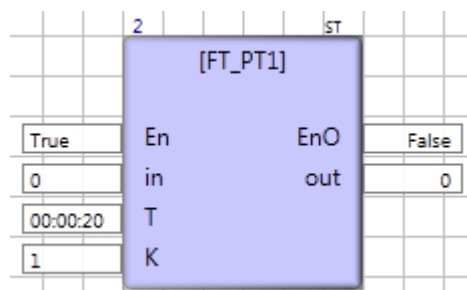
Значение выхода  $Y$  ограничивается значениями  $LIM\_L$  и  $LIM\_H$ :  $LIM\_L \leq Y \leq LIM\_H$ . Если  $Y$  достигает границы, значение интегральной части  $I$  ограничивается соответственно значениями  $LIM\_H - KP * IN$  и  $LIM\_L - KP * IN$ .

На вход  $IN$  ФБ  $FT\_PIWL$  подается выходной сигнал функции  $CTRL\_IN$  ( $OSCAT$ ), вычисляющей девиацию процесса.

По умолчанию входы ФБ имеют следующие значения:

$KP=1$ ;  $KI=1$ ;  $LIM\_L=-1E38$ ;  $LIM\_H=+1E38$

### 10.2.10.20. FT\_PT1 (OSCAT)



Входы ФБ:

- $IN$  (REAL) – входной сигнал;
- $T$  (TIME) – постоянная времени;
- $K$  (REAL) – коэффициент усиления.

Выходы ФБ:

- $OUT$  (REAL) – выходной сигнал.

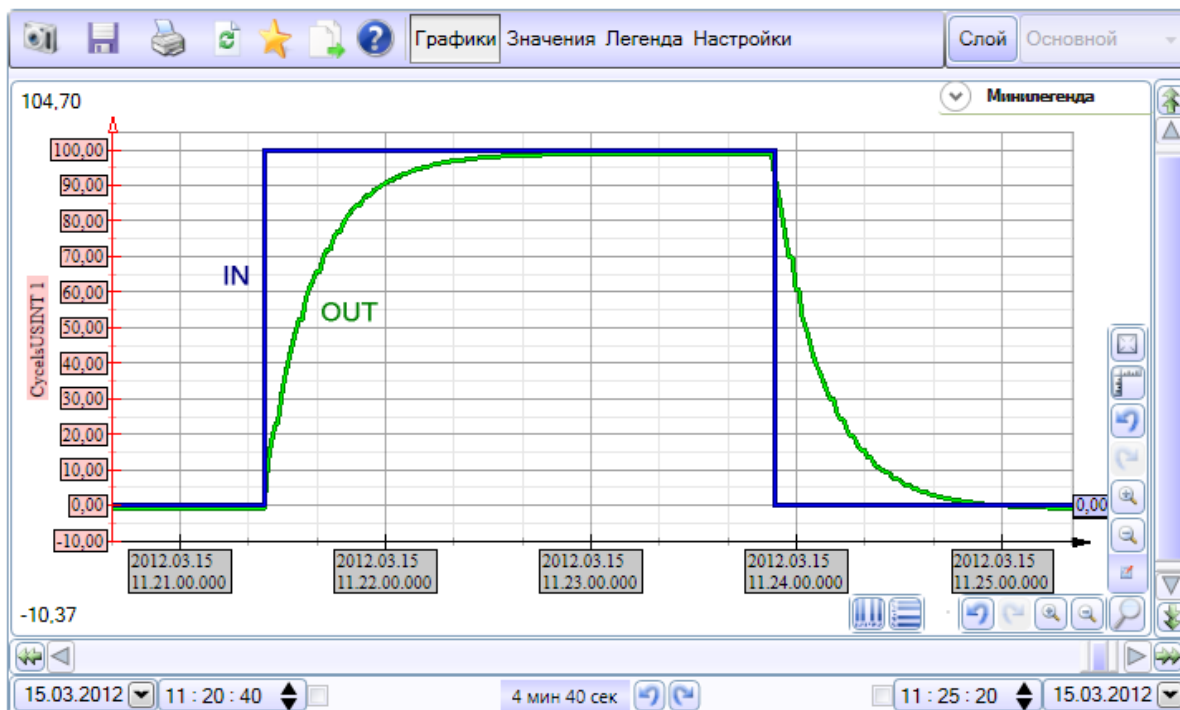
**FT\_PT1** – это низкочастотный фильтр 1-го порядка.

$FT\_PT1$  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).

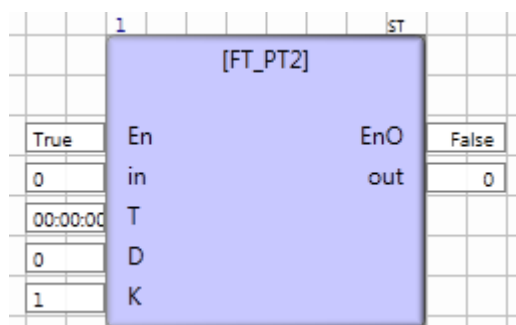
При старте ФБ  $OUT := K * IN$ .

Ниже показаны отклики ФБ FT\_PT1 на изменения IN при  $T = 15c$  (время цикла – 1с):



Используемые функции: T\_PLC\_MS (OSCAT).

### 10.2.10.21. 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.

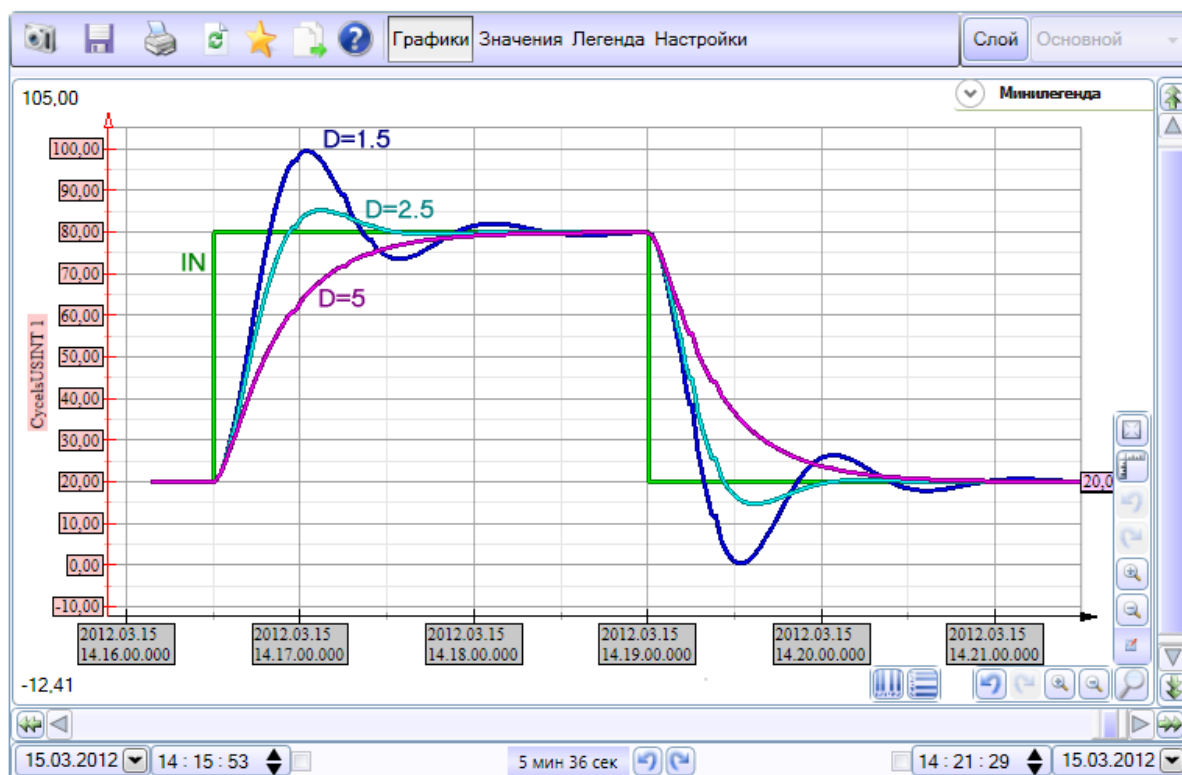
**FT\_PT2** – это низкочастотный фильтр 2-го порядка.

Если  $T=t\#0s$ ,  $OUT = K * IN$ .

Связь между переменными ФБ выражается следующим дифференциальным уравнением:

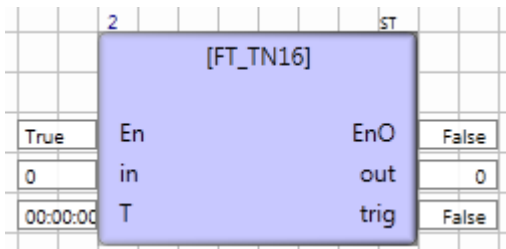
$$T^2 * OUT''(T) + 2 * D * T * OUT'(T) + OUT(T) = K * in(T)$$

Ниже показаны отклики ФБ FT\_PT2 на изменения IN при  $T=10c$  и  $D = 1.5 / 2.5 / 5$  (время цикла – 1c):



Используемые функции: INTEGRATE (OSCAT)₂

### 10.2.10.22. FT\_TN16 (OSCAT)



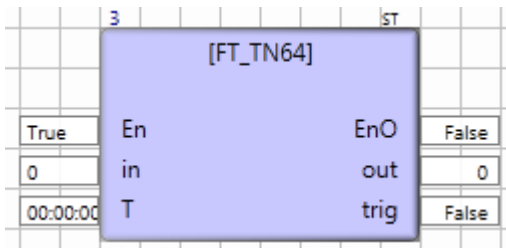
Тип данных входа IN и выхода OUT – REAL, входа T – TIME, выхода TRIG – BOOL.

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).

### 10.2.10.23. FT\_TN64 (OSCAT)



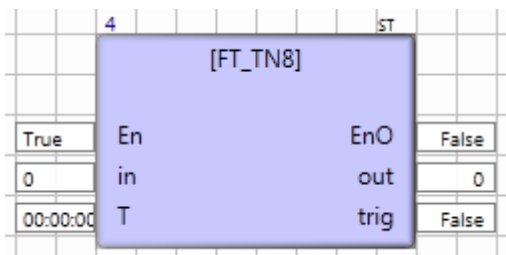
Тип данных входа IN и выхода OUT – REAL, входа T – TIME, выхода TRIG – BOOL.

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).

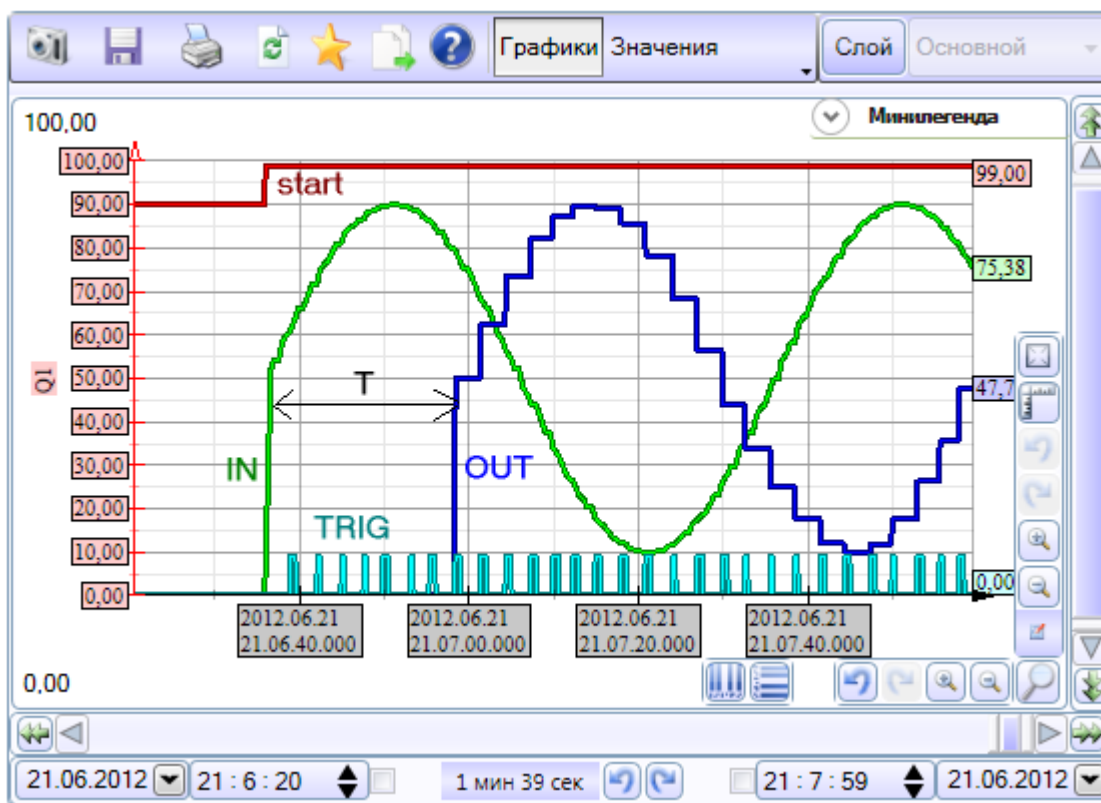
Используемые функции: T\_PLC\_MS (OSCAT).

### 10.2.10.24. 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:

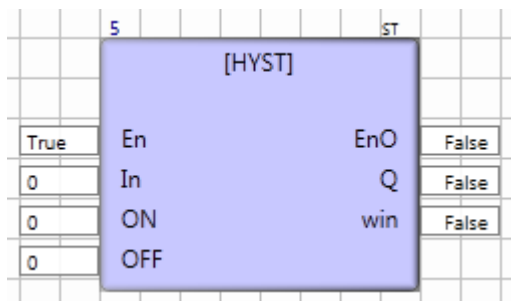


Аналогами FT\_TN8 являются ФБ FT\_TN16 (OSCAT) и FT\_TN64 (OSCAT).

Используемые функции: T\_PLC\_MS (OSCAT).

### 10.2.10.25. HYST (OSCAT)

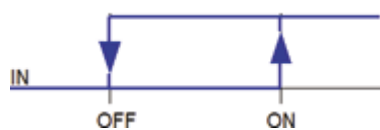




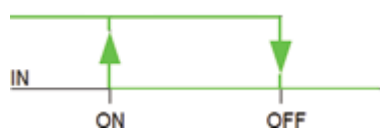
Тип данных входов IN, ON и OFF – REAL, выходов Q и WIN – BOOL.

HYST реализует гистерезис, зависящий от соотношения между значениями входов ON и OFF:

- $ON \geq OFF$ :
- если  $Q=FALSE$ , то Q становится TRUE при  $IN > ON$ ; если  $Q=TRUE$ , то Q становится FALSE при  $IN < OFF$ :



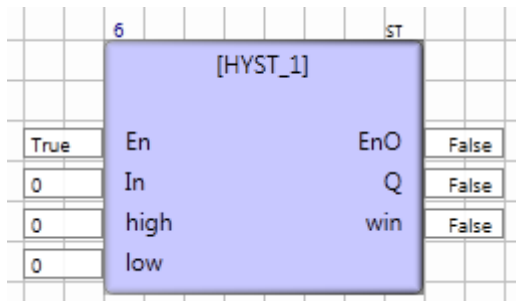
- $ON < OFF$ :
- если  $Q=FALSE$ , то Q становится TRUE при  $IN < ON$ ; если  $Q=TRUE$ , то Q становится FALSE при  $IN > OFF$ :



В обоих случаях  $WIN:=TRUE$ , если  $ON \leq IN \leq OFF$ ; если IN лежит вне диапазона [ON, OFF],  $WIN:=FALSE$ :

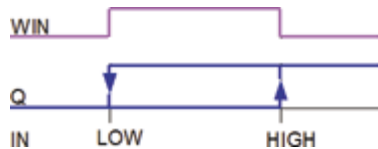


### 10.2.10.26. HYST\_1 (OSCAT)

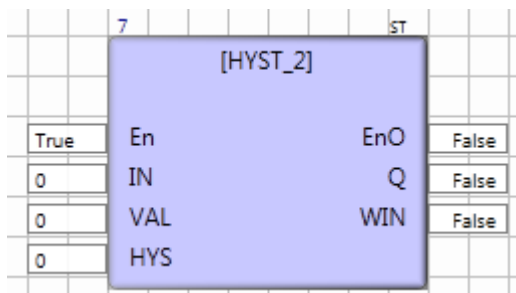


Тип данных входов IN, HIGH и LOW – REAL, выходов Q и WIN – BOOL.

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  $LOW \leq IN \leq HIGH$ , то  $WIN := TRUE$ , иначе  $WIN := FALSE$ .



### 10.2.10.27. HYST\_2 (OSCAT)



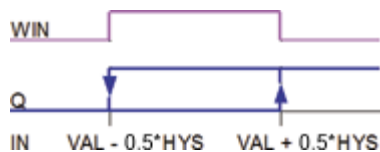
Тип данных входов IN, VAL и HYS – REAL, выходов Q и WIN – BOOL.

HYST\_2 реализует гистерезис.

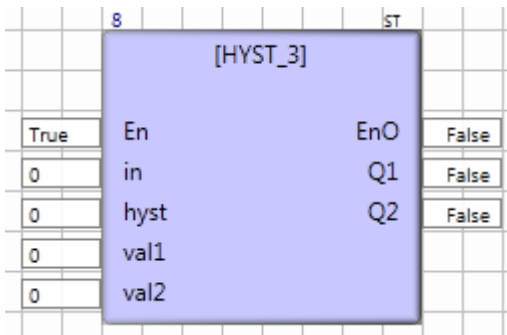
Если  $Q := FALSE$ , то Q становится TRUE при  $IN > VAL + 0.5 * HYS$ .

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  $WIN := FALSE$ .



### 10.2.10.28. HYST\_3 (OSCAT)

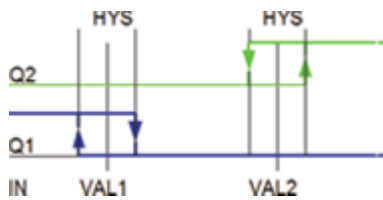


Тип данных входов IN, HYST, VAL1 и VAL2 – REAL, выходов Q1 и Q2 – BOOL.

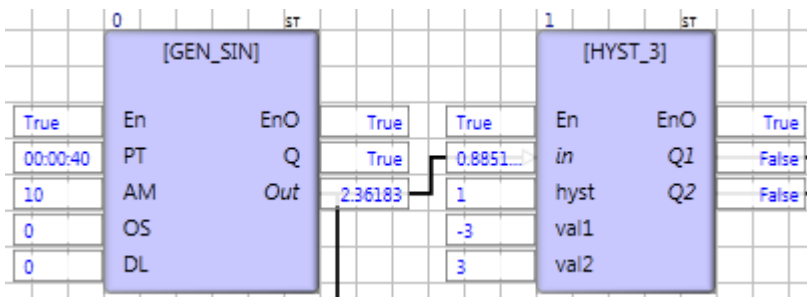
HYST\_3 – это 3-точечный контроллер. 3-точечный контроллер содержит две функции гистерезиса.

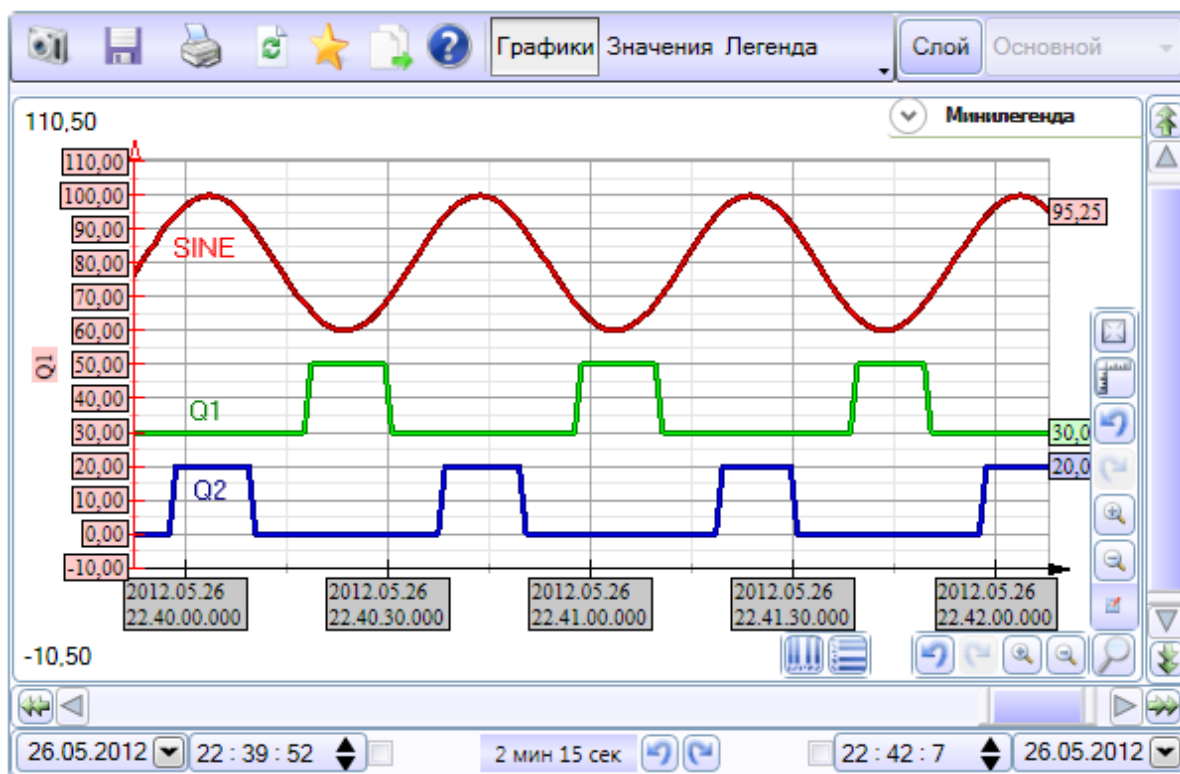
Если Q1=FALSE, то Q1 становится TRUE при  $IN < (VAL1 - HYST/2)$ . Если Q1=TRUE, то Q1 становится FALSE при  $IN > (VAL1 + HYST/2)$ .

Если Q2=FALSE, то Q2 становится TRUE при  $IN > (VAL2 + HYST/2)$ . Если Q2=TRUE, то Q2 становится FALSE при  $IN < (VAL2 - HYST/2)$ .

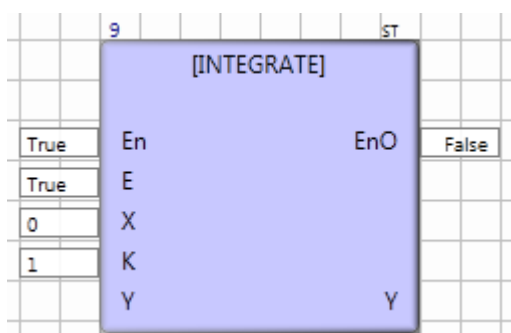


3-точечный контроллер используется, например, для управления приводом клапана. Следующий пример демонстрирует сигналы выходов 3-точечного контроллера:





### 10.2.10.29. 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):

$$Y_n := Y_{n-1} + K \cdot \frac{X_n + X_{n-1}}{2 \cdot 1000} \cdot (t_n - t_{n-1})$$

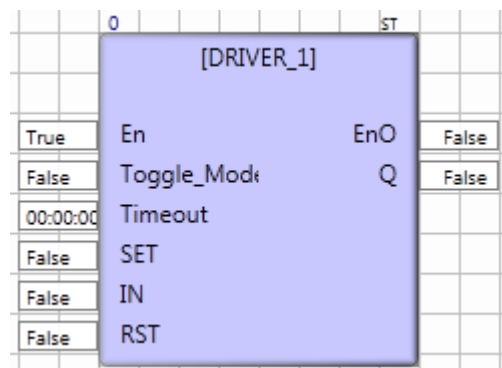
Временные метки определяются с помощью функции T\_PLC\_MS (OSCAT).

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.

## 10.2.11. OSCAT.ФБ.Управление устройствами

### 10.2.11.1. DRIVER\_1 (OSCAT)



Тип данных входа TIMEOUT – TIME, остальных входов и выхода Q – BOOL.

Алгоритм ФБ при SET=RST=FALSE:

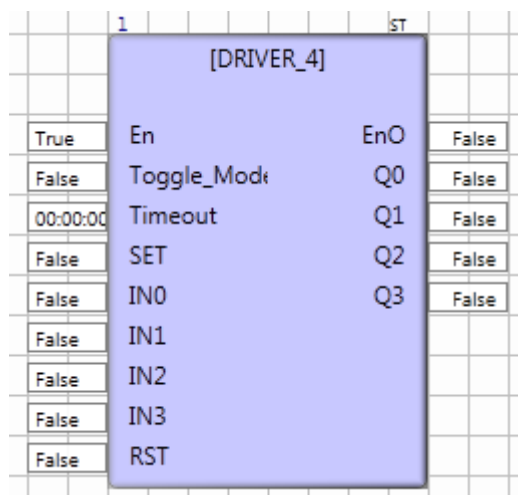
- TOGGLE\_MODE=FALSE:
- если IN=TRUE, Q=TRUE (если TIMEOUT>t#0s, Q=TRUE в течение TIMEOUT);
- если IN=FALSE, Q=FALSE;
- TOGGLE\_MODE=TRUE:

- 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).

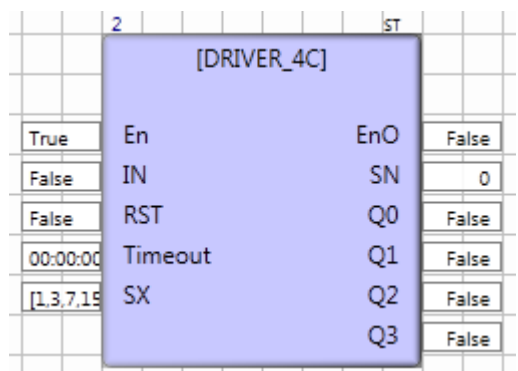
Если RST=TRUE, Q=FALSE (принудительный сброс Q).

### 10.2.11.2. DRIVER\_4 (OSCAT)



Данный ФБ представляет собой совокупность четырех ФБ DRIVER\_1 (OSCAT).

### 10.2.11.3. 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 состо-

яния 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.

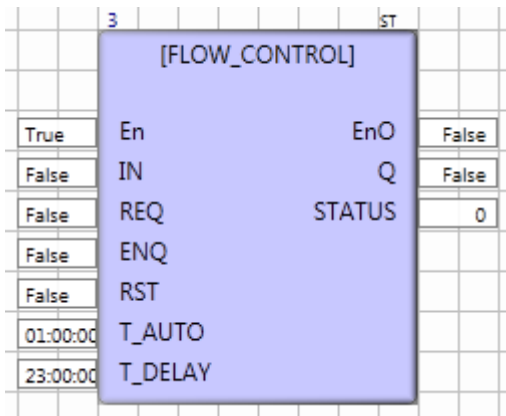
Пример

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.

### 10.2.11.4. 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).



### 10.2.11.5. FT\_PROFILE (OSCAT)

[FT_Profile]			
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 начинает 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

$M * (TIME\_<N> - \langle \text{время начала обработки участка } N \rangle)$

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:

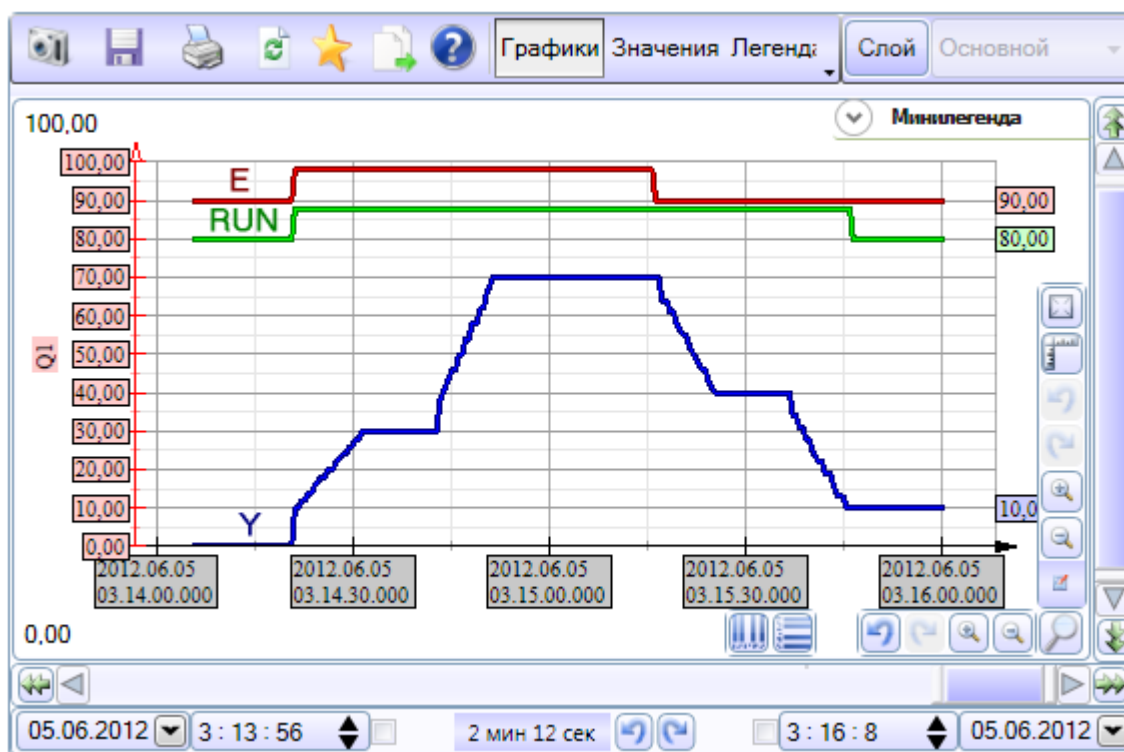
$$Y = K * [(VALUE\_<N> - VALUE\_<предыдущее>) * <время с начала участка N> / <длительность участка N>] + O;$$

По окончании отработки всех участков  $RUN=FALSE$ ,  $Y=VALUE\_13$ , а  $ET$  индицирует время, затраченное на отработку.

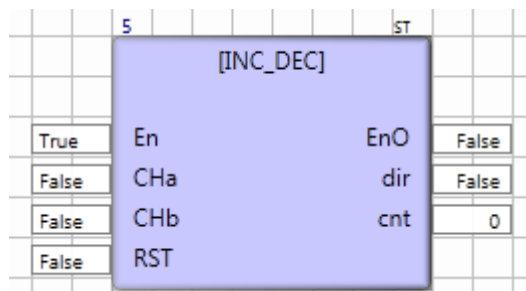
Используемые функции:  $T\_PLC\_MS$  (OSCAT).

Пример

Следующая временная диаграмма демонстрирует профиль при конфигурации ФБ, показанной на рисунке выше:



### 10.2.11.6. 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.

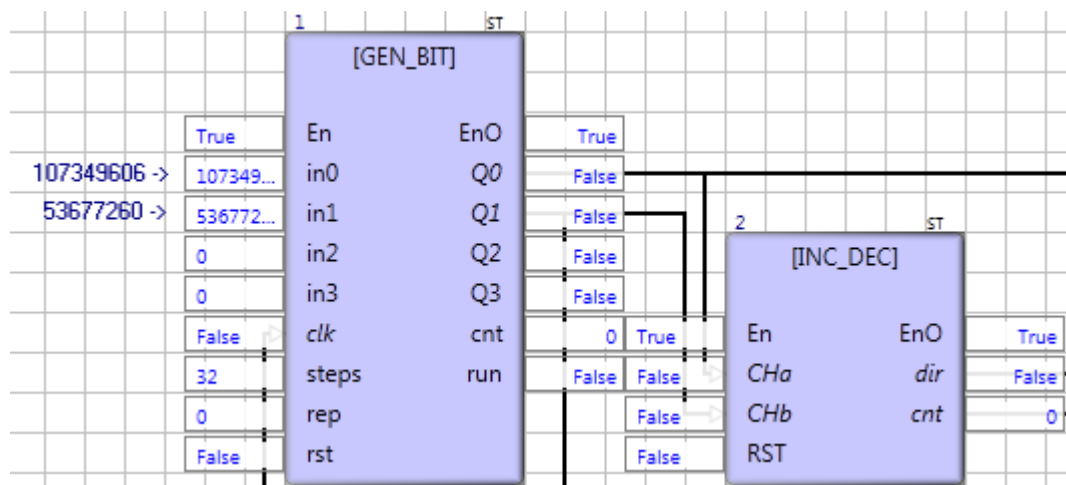
INC\_DEC – это декодер сигналов кодера вращения.

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).

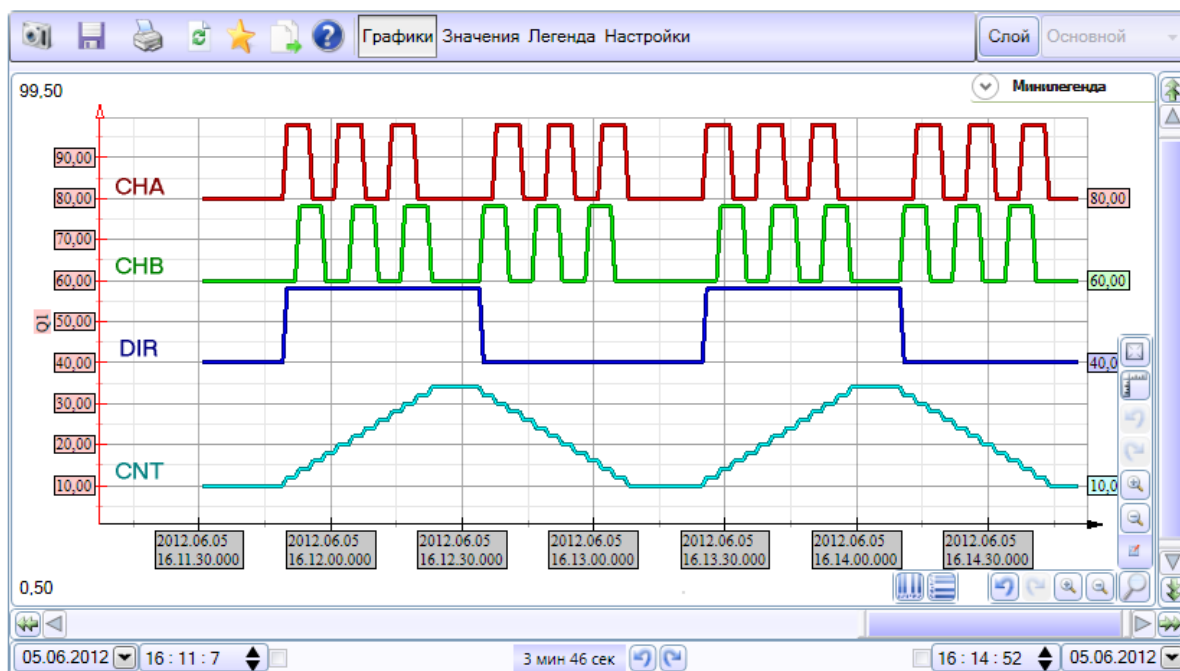
Команда RST=TRUE сбрасывает счетчик. Счетчик нарастающий, если DIR=TRUE, и убывающий, если DIR=FALSE.

Пример

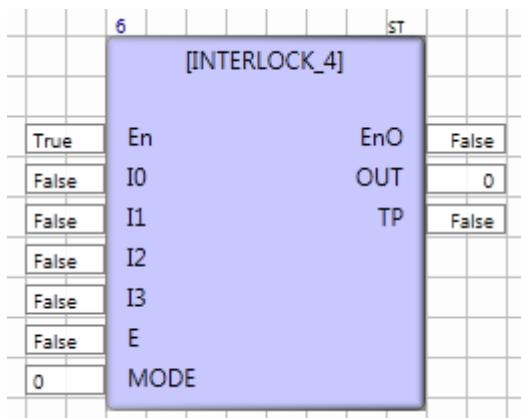
В данном примере в качестве имитатора кодера вращения используется ФБ GEN\_BIT (OSCAT); ФБ имитирует 3 шага поворота по часовой стрелке и 3 шага в обратном направлении ( $107349606 = 2\#0000\_0110\_0110\_0110\_0000\_0110\_0110\_0110$ ,  $53677260 = 2\#0000\_0011\_0011\_0011\_0000\_1100\_1100\_1100$ ):



Временная диаграмма программы показывает разбиение 3 шагов поворота на 12 и детектирование направления вращения:



### 10.2.11.7. 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.

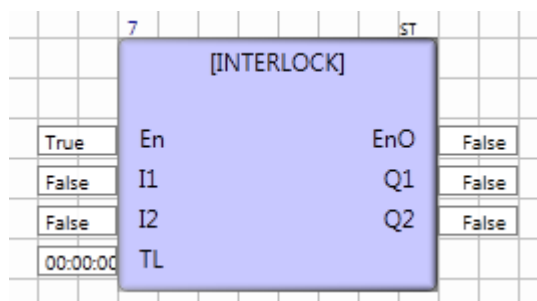
Если E=FALSE, выходы ФБ обнуляются.

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.

### 10.2.11.8. 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:

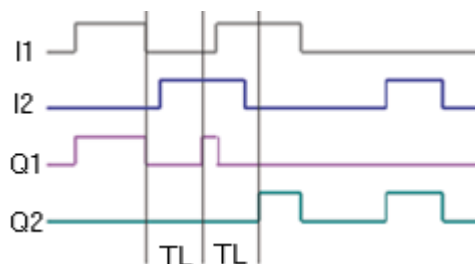
- выход Q<i> может принять значение TRUE только в том случае, если значение соответствующего входа I<i> равно TRUE, а с момента времени, когда другой выход принял значение FALSE, прошло не менее TL.

Как следствие, оба выхода не могут быть равны TRUE одновременно.

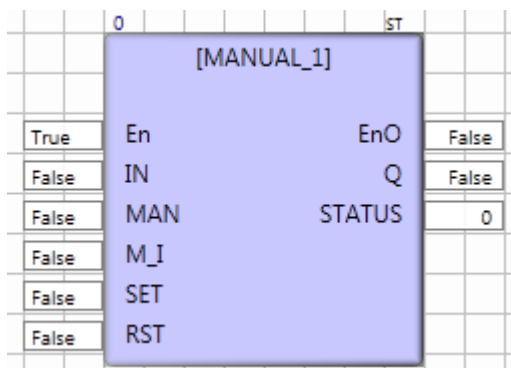
I1	I2	Q1	Q2
0	0	0	0

0	1	0	1
1	0	1	0
1	1	0	0

Следующая временная диаграмма поясняет алгоритм ФБ:



### 10.2.11.9. 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.

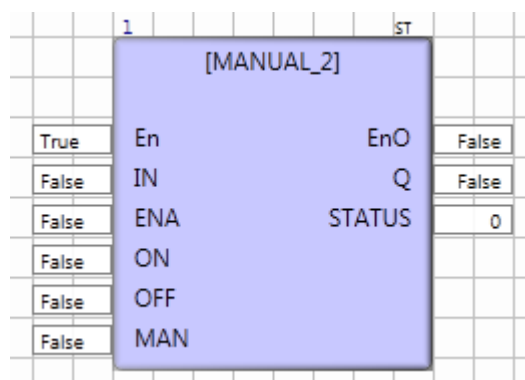
Выходы ФБ:

- 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).

### 10.2.11.10. 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.

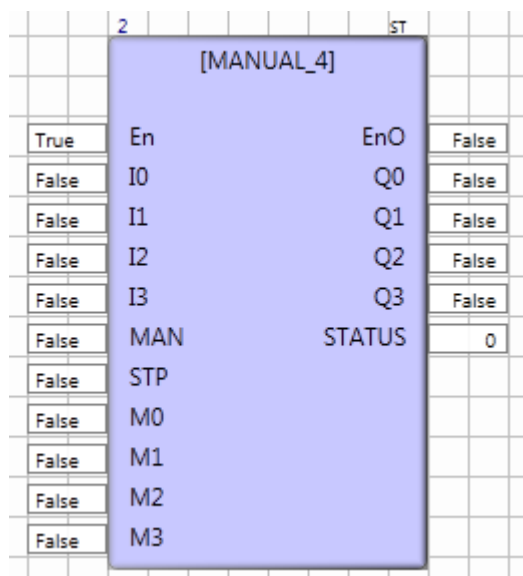
MANUAL\_2 позволяет заменить цифровой сигнал в ручном режиме:

- 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).

### 10.2.11.11. MANUAL\_4 (OSCAT)



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).

### 10.2.11.12. PARSET (OSCAT)

[PARSET]			
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).

### 10.2.11.13. 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:

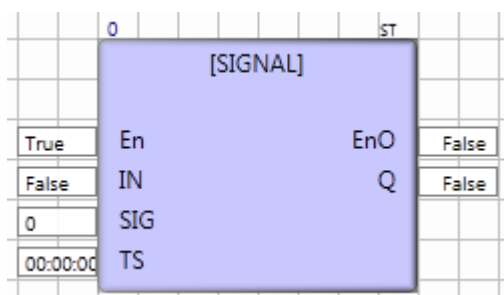
X	P1	P2	P3	P4
$\text{abs}(X) < L1$	X01	X02	X03	X04
$L1 \leq \text{abs}(X) < L2$	X11	X12	X13	X14
$L2 \leq \text{abs}(X) < L3$	X21	X22	X23	X24
$\text{abs}(X) \geq L3$	X31	X32	X33	X34

Изменение значений входов  $X_{i < i > < k >}$  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).

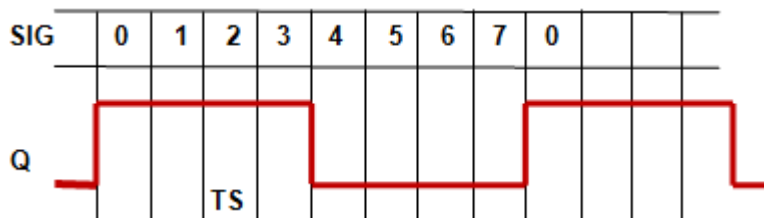
### 10.2.11.14. SIGNAL (OSCAT)



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 –  $1/(8*TS)$ ):

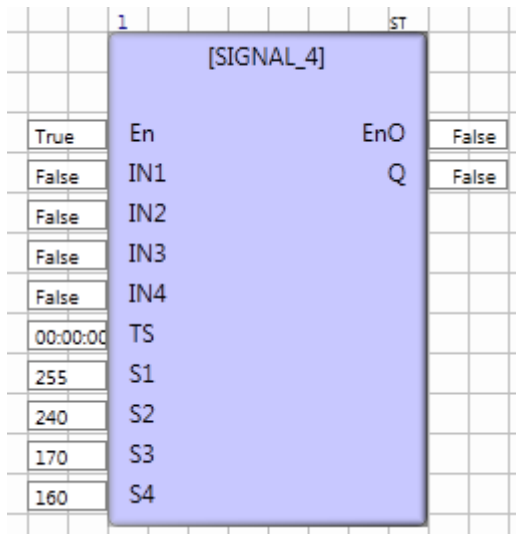


При старте генерации выходного сигнала первым передается случайный бит образца, далее передаются старшие биты, если  $TS > t\#0s$ , или младшие в противном случае.

Типичное применение ФБ – генерация сигнала для сирен или сигнальных ламп.

Используемые функции: T\_PLC\_MS (OSCAT).

### 10.2.11.15. 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.

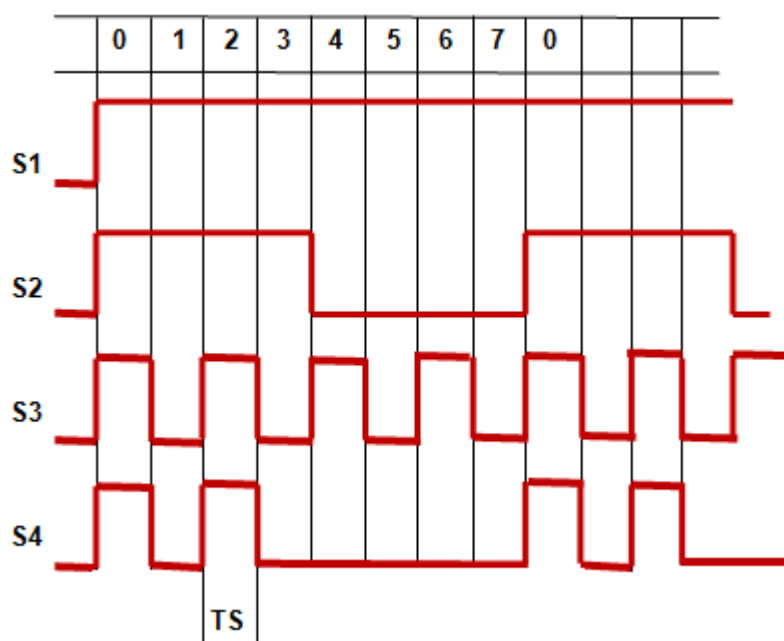
В ФБ заданы по умолчанию следующие образцы (могут быть изменены, в том числе в реальном времени):

S1 = 2#1111\_1111

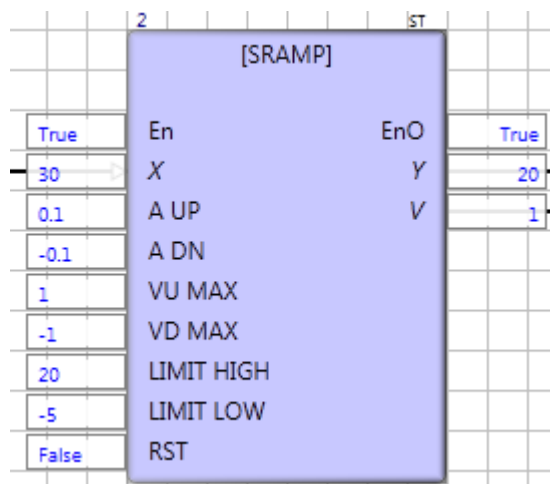
S2 = 2#1111\_0000

S3 = 2#1010\_1010

S4 = 2#1010\_0000



### 10.2.11.16. 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:

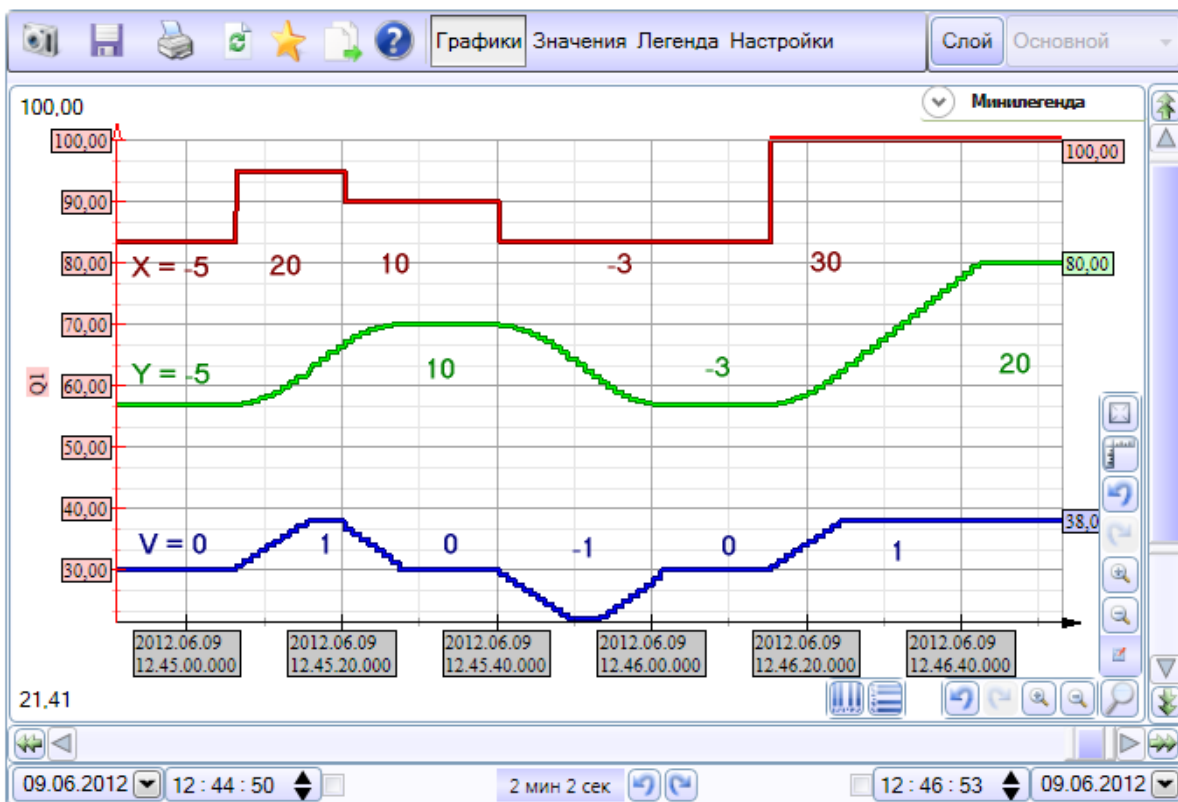
- Y (REAL) – выходной сигнал;
- V (REAL) – текущая скорость выходного сигнала.

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).



### 10.2.11.17. 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).

### 10.2.11.18. TUNE2 (OSCAT)

4		ST	
[TUNE2]			
True	En	EnO	False
False	SET	Y	0
False	SU		
False	SD		
False	FU		
False	FD		
False	RST		
0.1	SS		
5	FS		
0	Limit_L		
100	LIMIT_H		
0	RST_val		
100	SET_val		
00:00:00	TR		
2	S1		
10	S2		

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).

Адрес: 123298, г. Москва, ул. Маршала Бирюзова, д. 1, корп. 3

Телефон: +7 (495) 989-22-49

Факс: +7 (499) 943-02-14

Е-mail:

Отдел продаж программного обеспечения - [scada@insat.ru](mailto:scada@insat.ru)

Служба технической поддержки пользователей - [support.ms4d@insat.ru](mailto:support.ms4d@insat.ru)

Отдел продаж оборудования – [sales@insat.ru](mailto:sales@insat.ru)

Наши сайты:

[www.insat.ru](http://www.insat.ru)

[www.masterscada.ru](http://www.masterscada.ru)

[www.masteropc.ru](http://www.masteropc.ru)