

MasterSCADA

Основы проектирования

Методическое пособие

Москва 2016

Содержание

1.	Установка MasterSCADA	8
1.1.	Выбор компьютера	8
1.2.	Выбор операционной системы	9
1.3.	Установка демонстрационной версии	9
1.4.	Установка коммерческой версии	13
1.5.	Удаление MasterSCADA	14
2.	Основные правила, действия и определения.....	15
2.1.	Выбор версии для разработки проектов	15
2.2.	Запуск MasterSCADA в режиме разработки.....	18
2.3.	Основной интерфейс программы – менеджер проектов	19
2.4.	Дерево системы	23
2.5.	Об OPC-технологии и OPC-серверах.....	28
2.6.	Дерево объектов	29
2.7.	Функциональные блоки	34
2.8.	Общая структура проектов.....	37
2.9.	Выбор порядка разработки системы: от объекта или от системы	39
2.10.	Особенности создания связей между деревьями системы и объектов	39
2.11.	Базовые документы объектов.....	42
2.12.	Запуск режима исполнения.....	47
2.13.	Операторы в режиме исполнения и их права	50
3.	Объектно-ориентированный подход.....	53
3.1.	Инкапсуляция	55
3.2.	Механизм «Шаблон-Экземпляр»	64

3.3.	Использование библиотек	69
3.3.1.	Размещение объекта в библиотеке	69
3.3.2.	Вставка объекта из библиотеки	70
3.3.3.	Синхронизация объекта с библиотекой.....	71
3.4.	Механизм «Типизация».....	76
4.	Создание проекта с использованием объектного подхода	91
4.1.	Постановка задачи	91
4.2.	Настройка дерева системы	91
4.3.	Создание объектов	92
4.4.	Создание шкал	97
4.5.	Ручной ввод аварийных границ.....	103
4.6.	Создание журнала сообщений	105
4.7.	Создание тренда	107
4.8.	Настройка архивации данных.....	110
4.9.	Создание изображения объекта.....	111
4.10.	Тиражирование объектов.....	114
4.11.	Проверка работы в режиме исполнения	118
4.12.	Тестирование и отладка проекта	119
4.13.	Создание операторов и разграничение прав доступа.....	122
4.14.	Проверка совместимости версий MasterSCADA.....	126
4.15.	Перенос проекта с одного компьютера на другой.....	126
4.16.	Дополнительные материалы	128
5.	Работа с оборудованием.....	129
5.1.	Общие принципы взаимодействия с оборудованием	129

5.2.	Использование OPC-серверов	131
5.2.1.	Стандарты OPC.....	132
5.2.2.	OPC DA	133
5.2.3.	OPC HDA.....	136
5.2.4.	OPC UA.....	139
5.3.	Получение данных по протоколу Modbus с использованием OPC-сервера ...	148
5.4.	Подключение приборов через MasterPLC	153
5.4.1.	Создание проекта с контроллером	155
5.4.2.	Установление связи с MasterSCADA	160
5.4.3.	Установление связи с модулями ввода/вывода	164
5.4.4.	Расширенные возможности MasterPLC.....	170
6.	Редактор мнемосхем MasterGraf	171
6.1.	Создание мнемосхем и окон	172
6.2.	Окно редактора	172
6.3.	Добавление элементов из дерева объектов.....	173
6.4.	Добавление элементов из палитры мнемосхемы.....	176
6.5.	Создание оконного интерфейса	187
6.5.1.	Открытие через кнопки вызова	187
6.5.2.	Открытие окон по внешним сигналам и автоматическое открытие	189
6.5.3.	Продвинутые способы открытия окон	191
6.6.	Изображение объекта	200
6.6.1.	Создание изображения объекта.....	200
6.6.2.	Создание дополнительных изображений объекта.....	203
7.	Система архивации и тренды	206

7.1.	Система архивации	206
7.1.1.	Архивация в файловый архив	208
7.1.2.	Архивация в СУБД	210
7.2.	Тренды	215
7.2.1.	Создание тренда	215
7.2.2.	Уровни тренда	218
7.2.3.	Добавление перьев.....	219
7.2.4.	Навигация по архивным данным.....	222
7.2.5.	Вертикальный и горизонтальный курсоры.....	224
7.2.6.	Отображение аварийных и предупредительных границ.....	225
7.2.7.	Закладки.....	226
7.2.8.	Операции с перьями	228
7.2.9.	Действия – возврат значений из тренда	231
7.2.10.	Расширенный функционал	234
8.	Система сообщений и журналы	236
8.1.	Категории сообщений.....	236
8.1.1.	Канал «Сообщение».....	238
8.1.2.	Канал «Строка статуса».....	238
8.1.3.	Канал «Принтер»	239
8.1.4.	Канал «Архив»	239
8.1.5.	Каналы E-mail и SMS.....	240
8.1.6.	Канал «Звук»	242
8.2.	Журнал сообщений.....	242
8.2.1.	Настройка фильтра.....	243

8.2.2.	Настройка таблицы сообщений.....	245
8.2.3.	Навигация по архиву сообщений.....	246
8.2.4.	Фильтрация сообщений по строке	246
8.2.5.	Открытие трендов и переходы из журнала	247
8.2.6.	Создание пользовательского журнала	248
8.2.7.	Расширенный функционал	251
9.	Редактор отчетов MasterReport.....	252
9.1.	Создание отчета	252
9.2.	Интерфейс редактора отчетов	253
9.3.	Добавление бэндов	254
9.4.	Создание источника данных и добавление колонок	256
9.5.	Способы обработки колонок	261
9.6.	Обработка плохих признаков качества	263
9.7.	Дополнительные материалы и шаблонные отчеты	265
10.	Сетевые проекты MasterSCADA	266
10.1.	Общий принцип создания сетевых проектов	266
10.2.	Синхронизация проекта	267
10.3.	Сетевые архитектуры	269
10.3.1.	Клиент-серверная архитектура и сервер-серверная архитектура.....	269
10.3.2.	Система с архивным сервером	271
10.3.3.	Резервированная система	272
10.3.4.	Мультиклиентная система	275
10.3.5.	Комбинация архитектур	276
10.4.	Дополнительные материалы	276

11. Заключение 277

1. Установка MasterSCADA

1.1. Выбор компьютера

Подбор характеристик компьютера во многом зависит от реализуемого проекта – чем крупнее проект, чем больше в нем опрашиваемых переменных, тем мощнее должен быть компьютер. Также существенное влияние на скорость работы оказывает стороннее ПО: антивирусы, СУБД. Далее мы приведем рекомендуемые требования к MasterSCADA.

При выборе компьютера, на котором будет установлена MasterSCADA, отдавайте предпочтение устройствам с большим объемом оперативной памяти. Объем 2048 Мб является минимальным, а для комфортной работы требуется 4096 Мб – или даже больше, если планируется крупномасштабный проект.

Процессор – тоже важная деталь системного блока, но к нему предъявляются не столь жесткие требования. Рекомендуется использовать компьютеры с многоядерными процессорами (Intel Core i3 и выше).

Перед установкой MasterSCADA убедитесь, что на жестком диске свободно более 400 Мб. Возможно, в дальнейшем потребуется больше места для обеспечения корректной работы проектов и хранения архивов, но пока этого объема хватит, чтобы установить программу и приступить к ее изучению. Выбор диска с большим быстродействием (или SSD-накопителя) также поспособствует комфорту вашей работы.

Чтобы создавать большие многофункциональные графические окна, требуется монитор с разрешением не меньше чем 1280 пикселей на 1024. Однако это «спартанский минимум». Лучше используйте современные широкоформатные мониторы Full HD мониторы – с разрешением 1920 x 1080. Если в проекте предполагается использование нескольких мониторов (или компьютеров), то желательно иметь мониторы с одинаковым разрешением: это упростит проектирование и улучшит восприятие пользовательского интерфейса.

Важно! Чтобы значительно сократить время разработки проектов, разрешения мониторов у разработчика проекта и у конечного пользователя (оператора) должны совпадать. Кроме того, если в проекте предполагается использование нескольких мониторов (или компьютеров), то желательно иметь мониторы с одинаковым

разрешением: это упростит проектирование и улучшит восприятие пользовательского интерфейса.

1.2. Выбор операционной системы

MasterSCADA поддерживает работу с операционными системами Windows XP (SP3), Windows 7, Windows 8, Windows 10.

Важно! В 64-разрядных версиях Windows MasterSCADA работает в режиме 32-разрядного приложения.

MasterSCADA также может работать на серверных версиях операционной системы: Windows Server 2003 (SP1), Windows Server 2008 (R2), Windows Server 2012. Это наиболее актуально, когда на компьютере помимо MasterSCADA будет установлена какая-либо программа, требующая именно серверного исполнения Windows, например СУБД MSSQL.

1.3. Установка демонстрационной версии

Перед установкой убедитесь, что на компьютере отсутствуют другие версии MasterSCADA. Если они обнаружатся, их необходимо удалить перед установкой новой версии.

Установка MasterSCADA ничем не отличается от большинства типовых установок программ. Запустите файл **setup.exe**¹ и следуйте указаниям программы установки. Этот процесс может занять несколько минут (зависит от производительности компьютера), он несложен и интуитивно понятен, однако некоторые моменты требуют дополнительных комментариев.

Важно! Для версии MasterSCADA 3.3 и выше требуется Microsoft .Net Framework 4.0. Многие пользователи ошибочно полагают, что если на компьютере уже установлена какая-либо версия данного приложения, более ранняя или более поздняя, то этого достаточно для MasterSCADA. Однако это не всегда так.

Если приложение **Microsoft .NET Framework** не найдется в списке программ компьютера, то инсталлятор MasterSCADA предложит установить его автоматически.

¹ Если установка осуществляется с фирменного диска, то файл **setup.exe** запускается из меню диска.

Перед тем как нажимать кнопку **Далее** в диалоговом окне, показанном на 1-1, ознакомьтесь с текстом лицензионного соглашения на демонстрационную версию MasterSCADA. В нем содержится много полезной информации: правила использования демоверсии MasterSCADA, особенности взаимодействия со службой технической поддержки компании «ИнСАТ» и др.

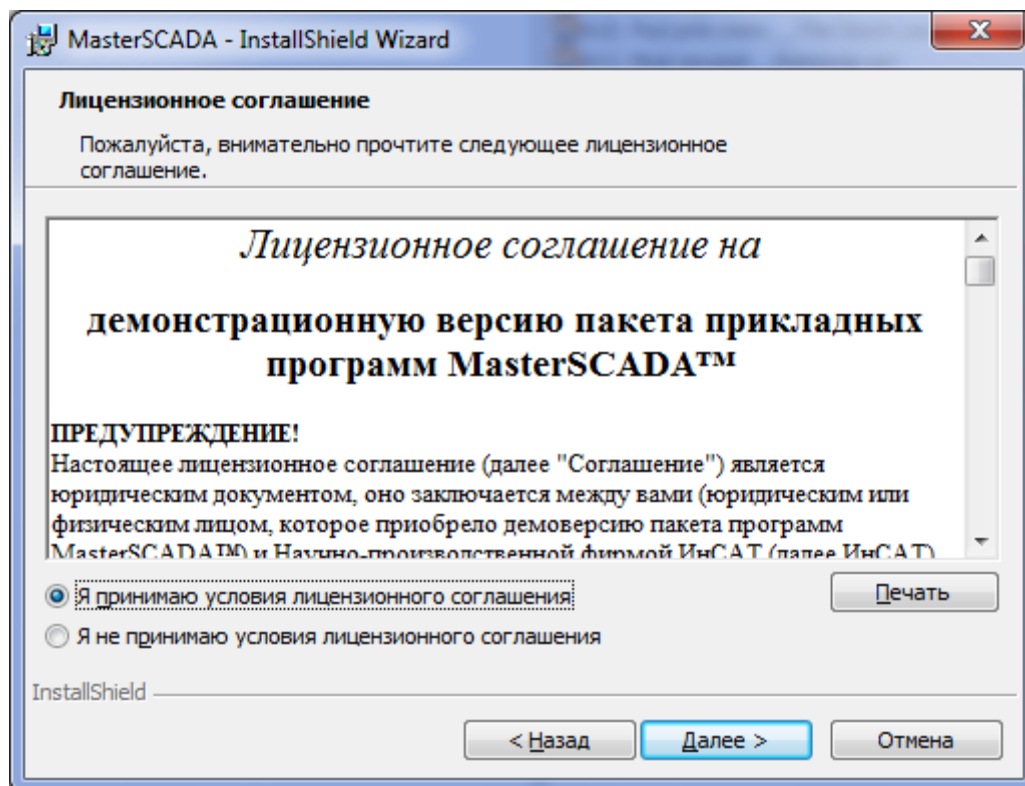


Рисунок 1-1. Окно с лицензионным соглашением

Обратите особое внимание на 1-2, где показано, как определяется папка, в которой по умолчанию будут сохраняться ваши проекты и библиотеки. В дальнейшем изменить место расположения рабочего каталога MasterSCADA **Projects** можно будет из интерфейса MasterSCADA.

Важно! Папка проектов не удаляется при удалении MasterSCADA с компьютера. Она не изменяется при следующих установках MasterSCADA. Благодаря этому вы можете спокойно переустанавливать и обновлять MasterSCADA, не беспокоясь за сохранность ваших проектов и библиотек.

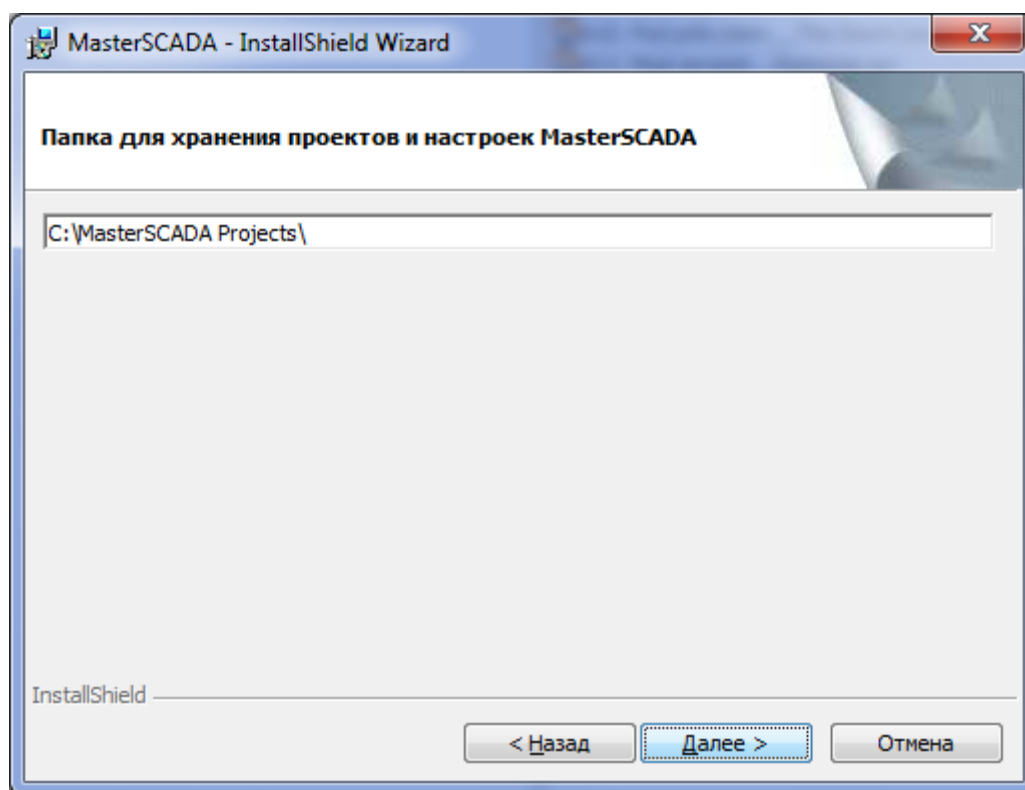


Рисунок 1-2. Выбор папки хранения проектов и настроек

Задать место установки пакета программ MasterSCADA и определить, какие компоненты необходимо установить на компьютер, можно в окне, показанном на 1-3. Основной выбор сводится к тому, ставить или нет демонстрационные программы и дополнительные библиотеки. Если вы только начинаете работать с MasterSCADA, не пренебрегайте ими, если позволяет свободное место на диске. Оставьте лучше поле выбора компонентов программы без изменений.

В процессе установки программы папка *Insat²/MasterSCADA* по умолчанию будет создана в том каталоге, куда устанавливаются все приложения на компьютере. Вы можете изменить место установки, нажав на кнопку *Изменить* в диалоговом окне.

Важно! Желательно не изменять имена указанных директорий. Служба технической поддержки может запросить какую-либо информацию о файлах, находящихся в папке

² В эту директорию по умолчанию устанавливаются все программные продукты компании «ИнСАТ», например OPC-серверы.

установки MasterSCADA, и стандартные названия значительно упростят поиск нужной информации.

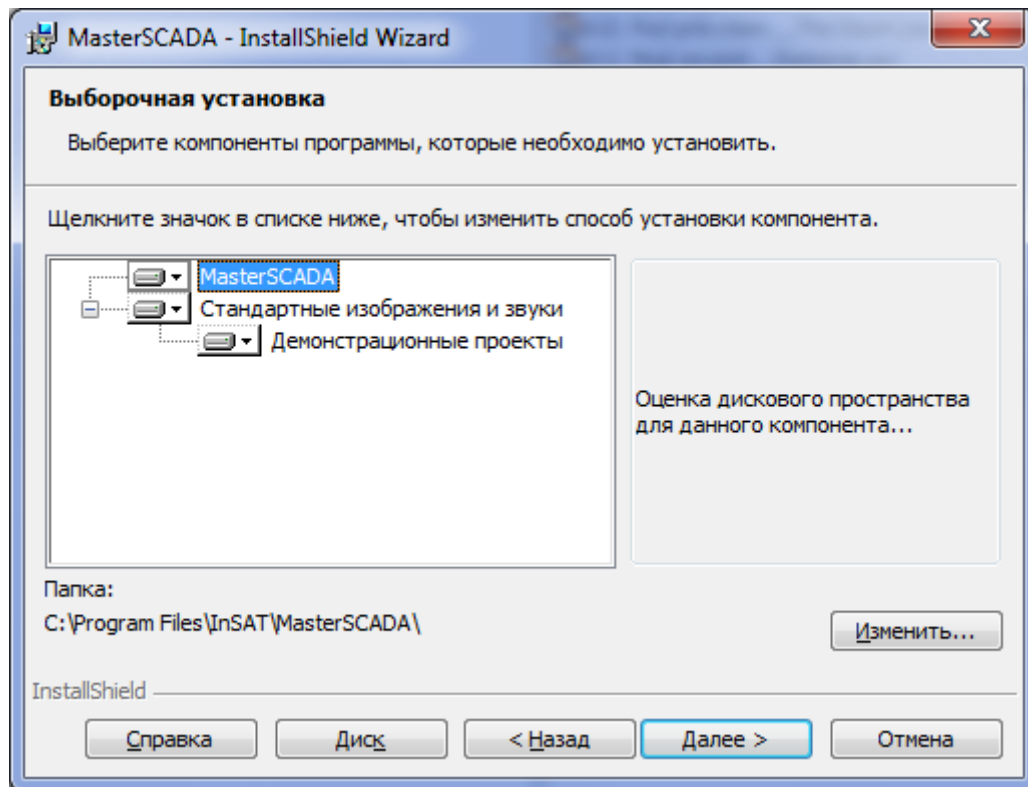


Рисунок 1-3. Место установки MasterSCADA

Завершает процесс установки MasterSCADA диалоговое окно, показанное на 1-4. После нажатия на кнопку **Готово** можно приступать к изучению MasterSCADA.

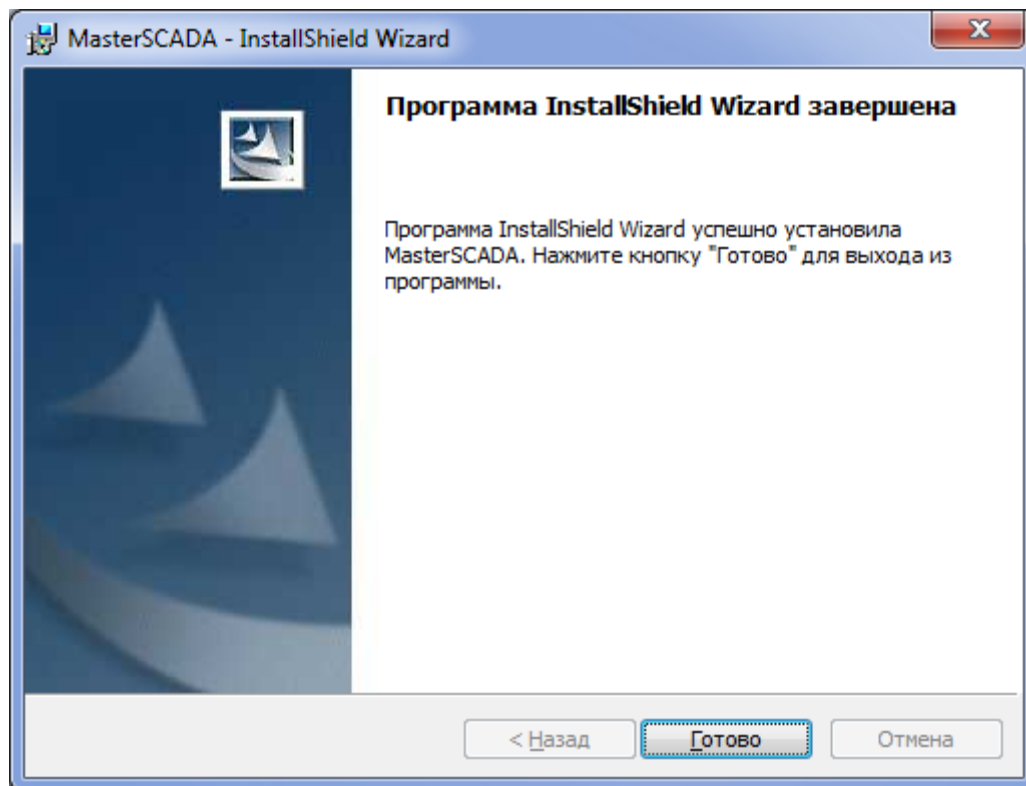


Рисунок 1-4. Завершение установки MasterSCADA

1.4. Установка коммерческой версии

Процесс установки коммерческой версии внешне не отличается от установки демонстрационной. Единственное визуальное отличие – текст лицензионного соглашения, который будет иным.

При установке лицензионной версии MasterSCADA автоматически будет установлен драйвер ключа защиты программного обеспечения.

Важно! Ключ защиты должен отсутствовать в USB-порту компьютера в момент установки MasterSCADA. Установите его в порт только по завершении инсталляции – только в этом случае драйверы будут установлены корректно. При переустановке MasterSCADA извлекать USB-ключ не нужно: если драйверы уже установлены, снова устанавливаться они не будут.

1.5. Удаление MasterSCADA

Удаление MasterSCADA происходит через «Панель управления» Windows (пункт **Установка/Удаление программ**).

Деинсталляция MasterSCADA не затрагивает программ драйвера ключа защиты и **Microsoft . NET Framework**. Они останутся на компьютере.

Важно! Папка для хранения библиотек и проектов (..\MasterSCADAProjects) не удаляется при удалении MasterSCADA.

2. Основные правила, действия и определения.

Прежде чем приступить к изучению функций и возможностей MasterSCADA, нам (разработчикам этой программы) с вами (ее будущими пользователями) необходимо научиться понимать друг друга. Очень важно научиться говорить на одном языке, в одни и те же термины вкладывать одинаковый смысл.

Неважно, какой проект вы хотите создать, используя MasterSCADA: простой лабораторный стенд или сложную АСУТП на опасном производстве. В любом случае все разработчики выполняют одни и те же действия, придерживаясь определенных правил: открывают проекты, редактируют, добавляют переменные, создают графические окна и т. д. Со временем часто повторяющиеся действия доводятся до автоматизма, поэтому очень важно, чтобы с самого начала сформировались правильные привычки.

В этой главе вы узнаете о понятиях, принятых в MasterSCADA, и о тех действиях, которые выполняют все разработчики неоднократно в течение каждого рабочего дня.

2.1. Выбор версии для разработки проектов

Проект автоматизации с помощью SCADA-системы сначала разрабатывается, а уже потом используется в реальных «боевых» условиях. Во многих SCADA-системах для этого служат отдельные программы: в одной разработчик создает проект (обычно она так и называется – «среда разработки»), в другой – запускает проект на исполнение (это «исполнительная система»). В MasterSCADA для этих целей используется единая программа, в которой это просто два отдельных режима работы. В любой версии MasterSCADA (демонстрационной, коммерческой, клиентах с функциями управления и без них, за исключением интернет-клиента MasterSCADA) можно редактировать структуру существующих проектов и создавать новые. В этом случае у пользователя возникают вопросы: какая версия дает полное представление о возможностях пакета, какую версию выбрать для создания нового проекта?

Демонстрационная версия MasterSCADA – это полноценная среда разработки. В ней можно разработать любой проект любой сложности, используя все возможные модули и опции. Ограничение заключается только в том, что проект в режиме исполнения проработает

Основные правила, действия и определения.

один час, по истечении которого будет выведено сообщение о том, что время работы в режиме исполнения закончилось. Это вовсе не означает, что вы не сможете больше использовать данный экземпляр MasterSCADA, а говорит только о том, что необходимо вновь запустить проект на исполнение и продолжить его отладку.

Время создания проекта может сильно варьироваться. На это влияют многие факторы: сложность проекта, квалификация разработчика, а зачастую и финансирование. В результате получается, что за время создания проекта MasterSCADA может измениться. Развитие MasterSCADA происходит непрерывно, но пользователь видит отдельные шаги-обновления, шага-релизы, прыжки-версии. Эти обновления отражаются в нумерации продукта. Например, номер 3.4.SP2 означает: обновление 2 релиза 4 версии 3. Обновления, так называемые «исправленные» (Service Pack) версии, выходят примерно раз в несколько месяцев. В каждом таком обновлении исправляются ошибки, а также учитываются некоторые пожелания пользователей (как правило, не требующие больших изменений в программе), добавляются небольшие усовершенствования функций и расширения библиотек. В релиз, выпускаемый 1–2 раза в год, могут включаться уже достаточно серьезные дополнения, а новая версия, выходящая раз в несколько лет (примерно раз в 3 года), всегда означает очень серьезное расширение возможностей продукта, вплоть до смены технологической платформы отдельных модулей или даже ядра.

Планируя работу над проектом, важно понимать, что переход с одной версии на другую может потребовать от вас дополнительных усилий: необходимо будет проверить, как то, что разработано под старые возможности, будет сочетаться с новыми. Мы делаем все от нас зависящее, чтобы любые проекты конвертировались в новые версии. Однако в таких ситуациях, как полная замена одного модуля на другой, может оказаться, что вместо простой прежней функциональности реализована новая, более сложная, в которой старых настроек недостаточно для формирования необходимого вам вида или поведения приложения. Пожалуйста, будьте внимательны в подобных ситуациях.

Столь подробное изложение особенностей развития MasterSCADA неслучайно. Важно, чтобы вы понимали, когда имеет смысл обновлять версию, а когда нет. Развитие программы в соответствии с графиком выхода релизов и обновлений пойдет только на пользу проекту, находящемуся в ранних стадиях разработки. При этом переход на новую версию на

последних перед внедрением этапах может потребовать определенных трудозатрат – именно тогда, когда на использование вновь открывшихся существенных возможностей уже не хватает времени. Не забудьте также учесть, что политика компании «ИнСАТ» как производителя MasterSCADA – это полноценная поддержка старых версий в течение года после появления новой. По окончании этого срока мы не откажем вам в советах, но обновлений уже ждать не придется (для их получения потребуются приобрести продление технической поддержки).

Если же до внедрения более года, а до вас уже дошла информация (следите за нашими новостями!) о планируемом выпуске новой версии – именно версии, а не релиза, – то с приобретением можно и повременить. Это не помешает вам начать разработку, потому что пользователи демонстрационных версий получают почти такую же консультационную поддержку, как и пользователи коммерческих версий. Мало того, они могут высказывать свои пожелания о развитии функциональности MasterSCADA наравне с пользователями коммерческих версий. С высокой долей вероятности эти пожелания будут удовлетворены в очередных релизах или даже обновлении. Можете смело создавать проект в демонстрационной версии. А когда убедитесь, что пакет подходит для решения вашей задачи (в этом мы не сомневаемся), приобретайте коммерческую версию, в которой ваши пожелания уже будут удовлетворены. Основное коммерческое преимущество данного подхода – не отложенное расходование средств, как многие подумают, читая эти строки, а точный, без избыточности, выбор конфигурации MasterSCADA. Такой подход поможет избежать часто встречающейся ошибки с выбором типа коммерческой версии MasterSCADA и ее опций.

Важно! MasterSCADA может сама определить по разработанному проекту, какой комплект вам нужно будет приобрести. К этому вопросу мы обязательно еще вернемся, но уже после знакомства с MasterSCADA.

Демонстрационная версия MasterSCADA – это полноценный инструмент разработчика проектов. В дальнейшем все описания в этом тексте будут ссылаться именно на нее.

Основные правила, действия и определения.

2.2. Запуск MasterSCADA в режиме разработки

Запуск MasterSCADA в режиме разработки (далее мы будем обозначать его английской аббревиатурой DT – от термина DesignTime, что в переводе и означает «режим разработки») производится так же, как запуск любой другой Windows-программы: через меню **Пуск**, с помощью ярлыка на рабочем столе или в панели быстрого запуска. Обратите внимание на то, что на рабочем столе или в панели быстрого запуска ярлык надо предварительно самостоятельно создать, перетащив туда правой клавишей мыши значок приложения MasterSCADA из меню **Пуск**.

Итак, запускаем MasterSCADA. Нажимаем главную кнопку Windows **Пуск** и выбираем **Все программы – MasterSCADA – MasterSCADA**.

В открывшемся диалоговом окне (Рисунок 2.1) необходимо указать, с каким проектом вы будете работать: новым, учебным или иным, ранее созданным.

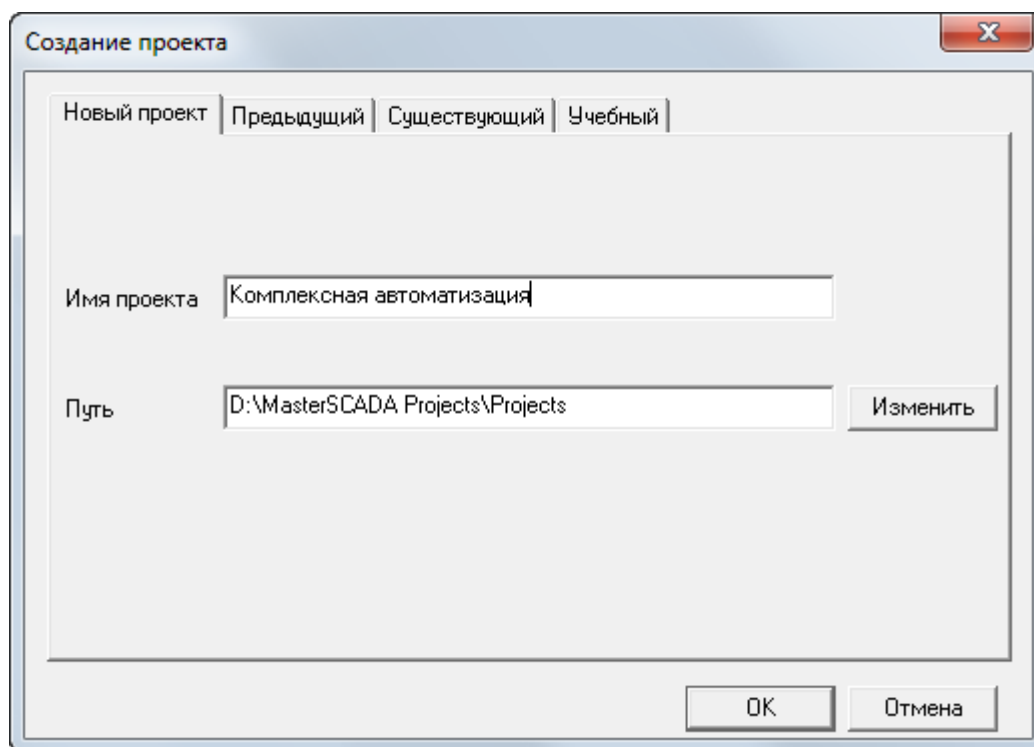


Рисунок 2.1. Окно создания или открытия проекта

Если вы создаете новый проект, то после создания и подтверждения его имени появится диалоговое окно, показанное на 2-2.

Основные правила, действия и определения.

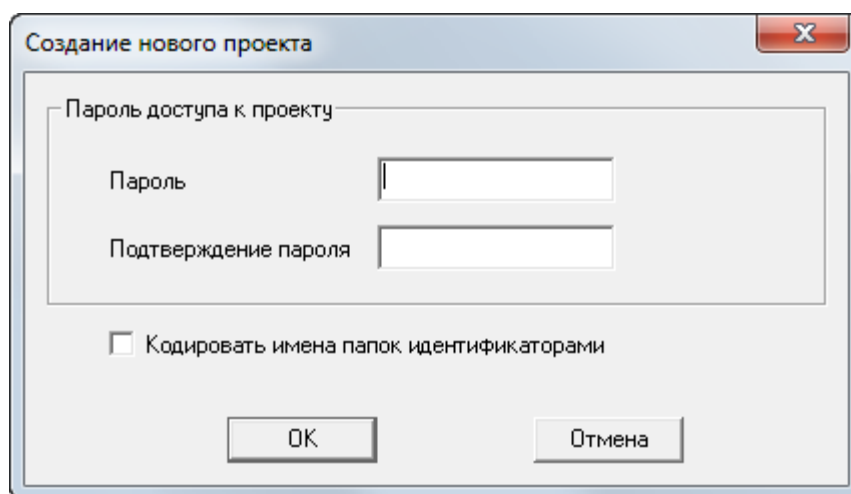


Рисунок 2-2. Окно задания пароля

Это окно позволяет с помощью пароля запретить открытие проекта для редактирования всем, кто этого пароля не знает. Если вы оставите поля пустыми, то любой пользователь сможет открыть данный проект в режиме редактирования. При его повторном открытии окно задания пароля больше появляться не будет, поэтому если вы захотите задать пароль на дальнейших этапах разработки, например после ее завершения, то для этого можно воспользоваться пунктом главного меню программы **Проект – Сохранить**. Установка пароля на разработку проекта поможет защитить вашу работу от неквалифицированных действий обслуживающего персонала, операторов, а также от происков конкурентов.

Важно! Если вы забудете комбинацию символов, вводимую в эти поля, то для восстановления данных придется переслать проект целиком в службу технической поддержки компании «ИнСАТ». Эта услуга оказывается только пользователям, у которых действует гарантийная или платная постгарантийная техническая поддержка.

2.3. Основной интерфейс программы – менеджер проектов

Итак, MasterSCADA готова к работе. После выполнения несложных действий, о которых мы только что рассказали, перед пользователем открывается основное поле его деятельности – менеджер проекта (Рисунок 2-3). В отличие от многих модульных программ MasterSCADA устроена так, что для выполнения самых разных действий и редактирования самых разных документов вам никуда не надо будет переключаться. Менеджер проектов работает по принципу единого окна в информационное пространство MasterSCADA.

Основные правила, действия и определения.

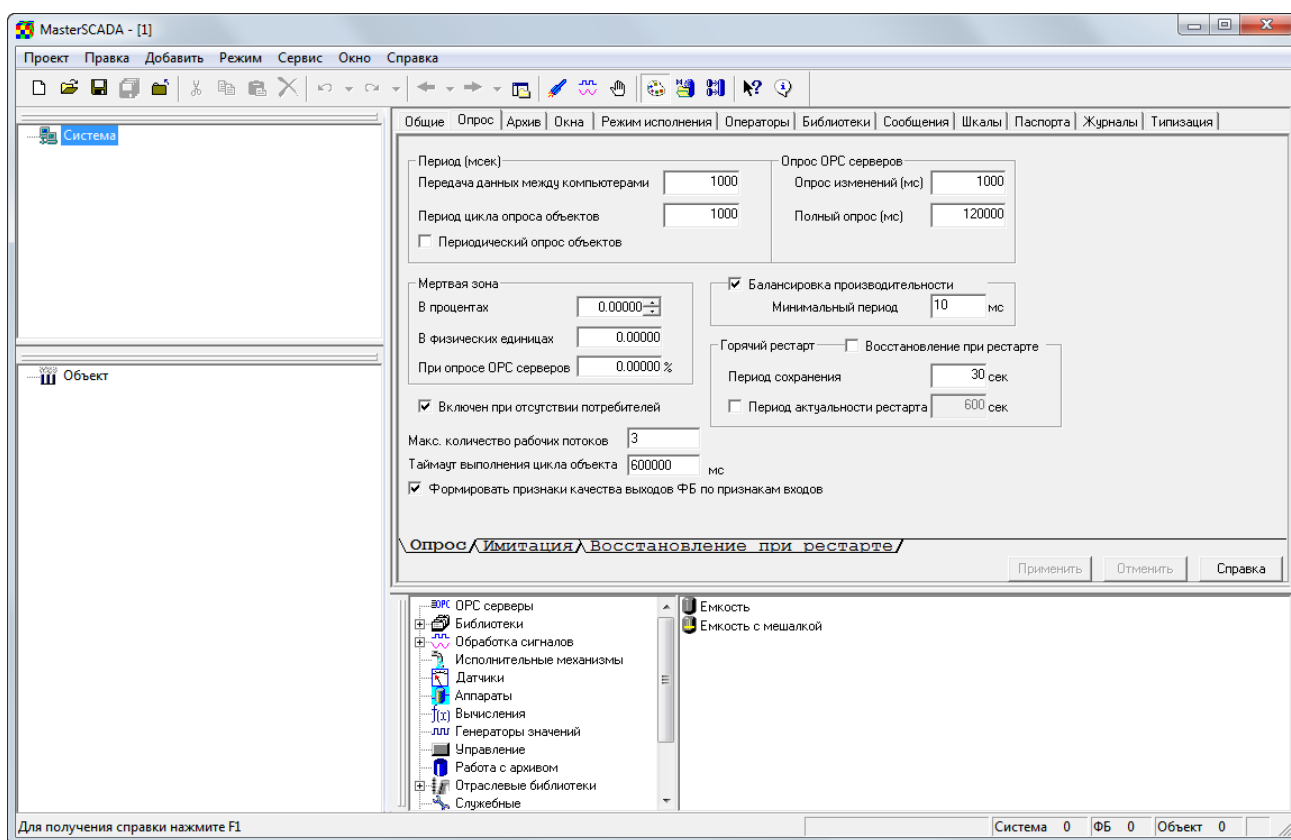


Рисунок 2-3. Менеджер проекта

Как и большинство Windows-приложений, MasterSCADA имеет главное меню программы, а также панели инструментов. Они располагаются в верхней части экрана. Наверное, эти понятия знакомы каждому, и порядок работы с иконками и пунктами меню описывать не нужно. Те детали, которые отличают MasterSCADA от других программ, мы еще обсудим в дальнейшем.

Рассматривая устройство окна программы, мы будем говорить о том расположении его частей, которое предлагается программой «по умолчанию». Но пользователь всегда может изменить его «под себя» исходя из формата своего монитора или собственных представлений об удобстве.

В левой части экрана находится главный навигатор по структуре проекта. Он состоит из двух частей: дерева системы и дерева объектов. Местоположение каждого из деревьев можно изменить в любой момент. Для этого нажмите ЛК³ на верхней кромке рамки дерева


³ Здесь и далее сокращение ЛК означает «левая кнопка мыши».

Основные правила, действия и определения.

(поле для нажатия обозначено двойной выпуклой линией) и, удерживая ее нажатой, перетащите окно.

Справа от деревьев находятся страницы свойств, содержимое которых относится к выбранному вами элементу дерева проекта. Состав страниц свойств будет изменяться в зависимости от того, какой элемент выделен.

«Нажмите правую кнопку мышки» – такую фразу неоднократно слышат пользователи, обратившиеся в службу технической поддержки по различным вопросам. Действительно, это действие любой разработчик выполняет неоднократно в течение рабочего дня. Нажав правую кнопку мыши, вы открываете контекстное меню элементов. Элементы MasterSCADA раскроют свои тайны быстрее, если знакомиться с ними по следующей схеме: добавить элемент в дерево, просмотреть его страницы свойств и контекстное меню. Как правило, этого бывает достаточно, чтобы понять назначение элемента и область его применения.

Но если это все же не помогло, то вызовите справку: функциональная клавиша **F1** на клавиатуре, либо пункт в контекстном меню, либо кнопка панели инструментов  – этой стрелкой можно щелкнуть по непонятному элементу и получить разъяснение (Рисунок 2-4).

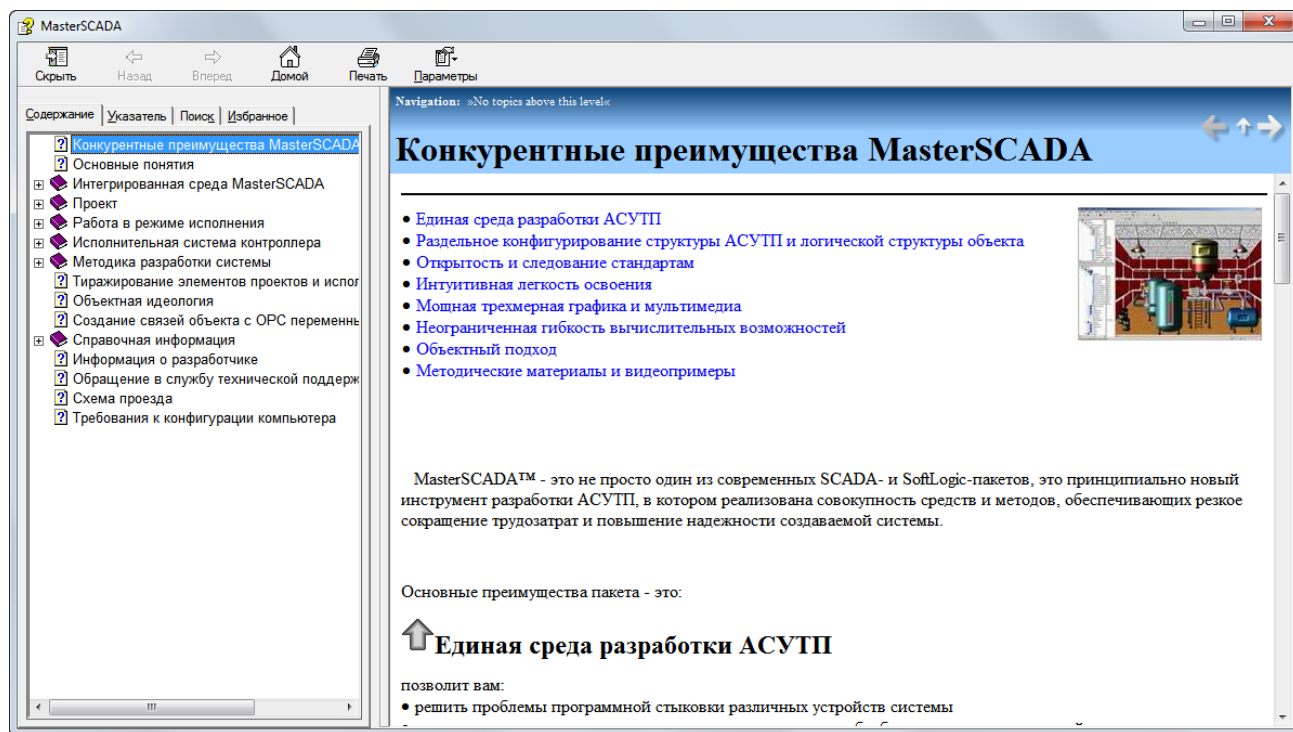


Рисунок 2-4. Окно справочной системы

Основные правила, действия и определения.

Наиболее часто контекстное меню элементов используют для формирования деревьев проекта. Пункты **Добавить** или **Вставить** подскажут, какие элементы могут быть использованы в том или ином месте. Например, если вы выделите элемент **Система** и нажмете правую кнопку мыши, то получите возможность добавить в проект **Компьютер**.

Помимо контекстного меню для формирования деревьев используют также палитру **функциональных блоков** MasterSCADA, которая находится под страницами свойств (Рисунок 2-5).

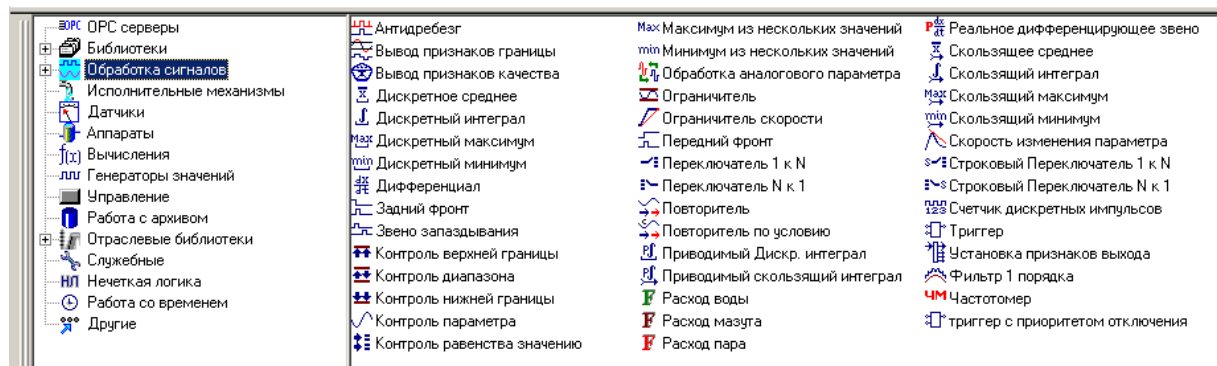



Рисунок 2-5. Палитра MasterSCADA

В ней хранятся различные ресурсы для формирования дерева проекта – как стандартные, так и создаваемые самими пользователями. Для того чтобы добавить элемент из палитры в дерево, достаточно выполнить следующие действия: щелкнуть по элементу палитры и отпустить кнопку – курсор приобретет вид стрелки с иконкой выбранного элемента (), второй раз нужно щелкнуть по тому элементу дерева, в который должен быть вставлен выбранный элемент. Если добавить элемент в то или иное место нельзя, то система сообщит об этом диалоговым окном (Рисунок 2-6).

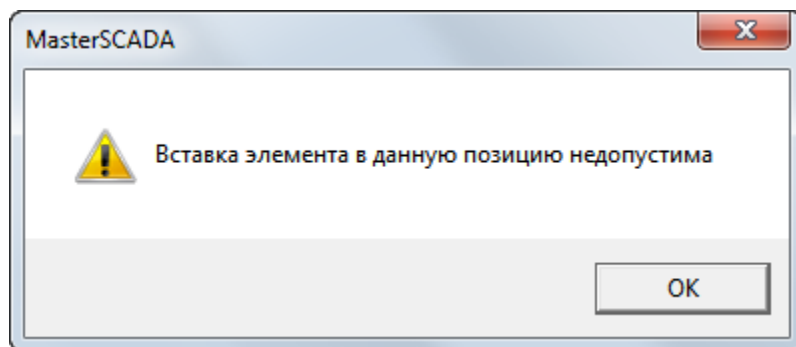


Рисунок 2-6. Сообщение о некорректной работе с палитрой MasterSCADA

Основные правила, действия и определения.

Статистическую информацию о проекте можно получить в строке статуса, она традиционно располагается в самой нижней части экрана.

Важно! Информация, находящаяся в данном поле, НЕ указывает на то, какую коммерческую версию MasterSCADA необходимо использовать для работы проекта.

2.4. Дерево системы

Изучая дерево системы готового проекта, можно определить, какую архитектуру реализовал разработчик, какое количество компьютеров будут взаимодействовать в системе, какой способ был выбран для подключения внешних устройств и модулей ввода/вывода, является ли система вертикально-интегрированной (то есть контроллеры тоже программируются в этом же проекте), какое количество внешних сигналов MasterSCADA будет обрабатывать. По сути, дерево системы – это своеобразный вид структурной схемы автоматизации.

Построение дерева системы начинается с элемента **Компьютер**. Физический компьютер даже в самом простом проекте выполняет несколько функций, например:

- к нему подключается оборудование нижнего уровня;
- он является рабочим местом оператора, технолога и т. п.;
- он передает информацию в приложения других компьютеров

С уверенностью можно сказать, что **Компьютер** – это точка пересечения реальных и виртуальных элементов создаваемой системы автоматизации. Без этого элемента не может существовать ни один проект MasterSCADA.

Компьютер может содержать дочерние элементы (Рисунок 2-7), представляющие собой понятия внешнего мира, с которым взаимодействует MasterSCADA, установленная на данном реальном компьютере: **Контроллер** (🏠), **Модуль ввода/вывода** (🔌), **ОПС-сервер** (🖨️), **MasterLink** (📡), **БД-коннектор** (🗄️). Рассмотрим эти понятия подробнее.

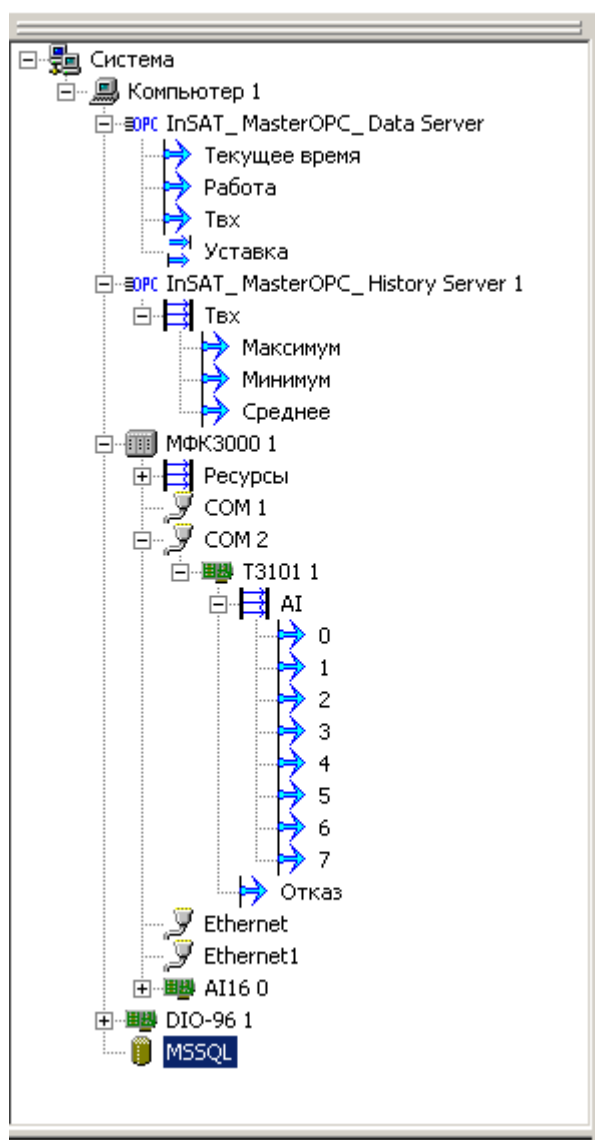


Рисунок 2-7. Дерево системы

Обращаться к пункту контекстного меню компьютера **Вставить – Контроллер** следует, если вы планируете средствами самой MasterSCADA программировать контроллер с открытой архитектурой, совместимый с исполнительной системой для контроллеров **MasterPLC** и входящий в комплекс программ MasterSCADA. В дальнейшем мы будем их обобщенно называть **MasterPLC-контроллерами**. Список таких контроллеров можно найти на сайте MasterSCADA.ru либо уточнить в службе технической поддержки компании «ИнСАТ».

Важно! Отсутствие интересующего вас типа контроллера в списке доступных для вставки в демонстрационной или, тем более, коммерческой версии MasterSCADA еще не означает отсутствия его поддержки. Возможно, она предоставляется дополнительно.

Основные правила, действия и определения.

При использовании **MasterPLC-контроллеров** они реализуют заданную в том же проекте логику контроля и управления, а MasterSCADA в режиме исполнения будет получать по различным каналам связи текущие и архивные данные с нижнего уровня напрямую (с помощью встроенных драйверов по фирменным протоколам), без участия других программных средств.

Для вставляемых в компьютер плат ввода/вывода в MasterSCADA предусмотрен элемент **Модуль ввода/вывода**. Список поддерживаемых плат определяется потребностями пользователей. Интерфейс для добавления новых плат открыт для самостоятельной разработки их драйверов. Однако в MasterSCADA такой способ подключения оборудования считается второстепенным, поэтому широкого набора драйверов не предусмотрено, а в качестве примера в демонстрационной версии представлен модуль дискретного ввода/вывода **DIO-96**.

Если контроллер запрограммирован не средствами MasterSCADA (например, с помощью фирменного ПО производителя) или в рамках другого проекта MasterSCADA, то, чтобы обеспечить его работу с MasterSCADA, потребуется OPC-сервер.

Элемент **OPC DA сервер** используется для работы с внешними физическими устройствами, модулями ввода/вывода, контроллерами, а также с различными программами по стандарту OPC DA. Этот стандарт описывает получение текущих данных. Фактически OPC является описанием стандартного программного интерфейса для драйверов различных протоколов. Эти драйверы могут быть разработаны независимыми производителями, но благодаря стандартизации совместимы с большинством современных SCADA-пакетов, включая, разумеется, и MasterSCADA, которая поддерживает OPC в полном объеме.

Если из прибора нужно получить архивные данные, то необходимо остановить свой выбор на OPC-сервере, который поддерживает стандарт OPC HDA.

Важно! Передача архивных данных через сервер OPC DA потребует большого числа переменных и очень трудоемкой обработки полученных данных в проекте (квалификация разработчика – программист).

При выборе архивного OPC-сервера в дерево системы добавляется элемент **OPC HDA сервер**. Как правило, этот элемент используется для решения задачи получения достоверных

Основные правила, действия и определения.

(в том числе с точки зрения точного указания метки времени события или измерения значений) данных в системах коммерческого и технического учета.

Важно! Некоторые OPC-серверы одновременно поддерживают стандарты DA и HDA, но в проект MasterSCADA они вставляются как два отдельных элемента.

Элемент *MasterLink* фактически является встроенным драйверным интерфейсом MasterSCADA, для которого реализованы драйверы, как для передачи мгновенных значений, так и архивов по различным каналам связи. Список поддерживаемых драйверов включает широко распространенные протоколы *Modbus*, *DCON*, собственный протокол MasterSCADA для связи с *MasterPLC-контроллерами*, принадлежащими другим проектам, а также ряд протоколов распространенных счетчиков коммерческого учета ресурсов.

Важно! Драйверный интерфейс MasterLink открыт, и вы можете написать собственные драйверы.

Несмотря на то что мы предоставляем возможность написания собственных драйверов для *MasterLink*, все же считаем, что в большинстве случаев правильнее использовать OPC-серверы, так как это более универсальный с системной точки зрения способ создания систем автоматизации.

Важно! Быстродействие при обмене данными через OPC-серверы и MasterLink практически не отличается, так как OPC также является стандартным встроенным драйверным интерфейсом MasterSCADA.

Элемент *БД-коннектор* служит для организации работы с такими базами данных, как *MSSQL*, *Oracle*, *Sybase*, *Interbase/Firebird*, *MySQL* и др. В настоящий момент, используя элемент *БД-коннектор*, можно реализовать следующие задачи:



- хранение архивов MasterSCADA во внешней базе данных⁴;
- экспорт данных;
- чтение и запись в базу данных значений переменных, используя хранимые процедуры;
- получение данных для формирования отчетов с помощью SQL-запросов.


⁴ В настоящий момент эта возможность реализована для MSSQL, Oracle, Firebird.

Основные правила, действия и определения.

Стоящая перед вами задача будет определять то, какие настройки необходимо выполнить на страницах свойств и в контекстном меню контроллеров, OPC-серверов и других элементов. В результате произведенных настроек в дереве системы появятся переменные.

Обратите внимание на маркировку переменных в дереве системы. В зависимости от иконки переменной в дереве можно понять, является ли она входом или выходом.

Обозначение  в дереве системы говорит разработчику о том, что данная величина является выходным параметром для внешнего источника сигналов (устройства или программы), а для MasterSCADA такая переменная будет считаться входом. Обозначение  сигнализирует об обратном: для внешнего устройства эта переменная будет входом, а для MasterSCADA – выходом.

Бывают переменные, которые работают и на чтение, и на запись. Такие переменные обозначены  .

Вносить изменения в дереве системы можно на любом этапе разработки проекта.

2.5. Об OPC-технологии и OPC-серверах

Понятия «OPC-сервер», «OPC-технология», «стандарт OPC» давно вошли в лексикон инженеров по автоматизации и хорошо им знакомы. OPC-сервер в широком понимании – это Windows-приложение (иногда служба), выполняющее функции драйвера для обмена информацией с внешними источниками или приемниками данных (контроллерами, модулями ввода/вывода сигналов, специализированными программами и др.) по какому-либо протоколу. Для каждого протокола – свой OPC-сервер. Благодаря тому, что OPC-серверы имеют одинаковый, определенный стандартом OPC, программный интерфейс (точнее, набор интерфейсов), с ними могут работать любые программные клиенты – например, SCADA-пакеты, также поддерживающие этот интерфейс. К сожалению, некоторые из серверов поддерживают не совсем полный комплект интерфейсов, заданных стандартом, что приводит к необходимости дополнительных настроек MasterSCADA для исключения из работы тех интерфейсов, которые не поддерживаются конкретным сервером.

Производители SCADA-пакетов, в которых имеются встроенные драйверные системы, иногда утверждают, что при опросе устройств через OPC-сервер появляется дополнительная задержка из-за того, что в работу включается лишнее приложение. По их мнению, SCADA-пакеты, которые работают с нижним уровнем через встроенные драйверы, получают данные быстрее. В их случае это действительно так, поскольку для работы с OPC-серверами они вынуждены производить преобразование OPC-интерфейсов в интерфейс внутренней драйверной системы и конвертировать пришедшие данные во внутренний формат. Для MasterSCADA это не так, поскольку технология OPC реализована внутри самой MasterSCADA. MasterSCADA не переводит данные из стандарта OPC в собственный, поэтому и дополнительной задержки практически не возникает.

Важно! Использование OPC-серверов для опроса внешних источников данных совместно с MasterSCADA не приводит к снижению быстродействия системы.

Будем различать два понятия: OPC-сервер (как отдельное приложение) и *OPC-сервер* (как элемент проекта MasterSCADA). Для того чтобы начать работать с данным элементом в

Основные правила, действия и определения.

дерева системы, необходимо установить приложение на компьютер и настроить его⁵. При обмене данными через OPC-сервер MasterSCADA воспроизводит данные в том виде, с теми отметками времени и с теми признаками достоверности, которые были получены из OPC-сервера.

Важно! Если по каким-то причинам данные не приходят, то начинать искать причину неудач нужно именно с программы OPC-сервера.

В предыдущем разделе мы говорили, что разработчик может добавлять в проект, в *Компьютер, OPC DA* или *OPC HDA серверы*. Но внимательный читатель заметит, что при первом открытии MasterSCADA пунктов *Добавить OPC-сервер* и *Вставить HDA OPC-сервер* нет. Чтобы пользователь получил такую возможность, необходимо выполнить пункт контекстного меню компьютера *Поиск OPC DA серверов* или *Поиск OPC HDA серверов* либо пункт главного меню *Сервис – Органайзер – OPC DA (HDA) серверов*.

Важно! Для использования OPC-серверов, расположенных на других компьютерах сети, необходимо на этих компьютерах запустить утилиту OPC Enum и настроить протокол Microsoft DCOM. Подробнее это описано в сопроводительной документации на MasterSCADA.

Еще раз напоминаем, что в MasterSCADA разделены элементы OPC DA и OPC HDA серверов. Даже если вы установили на компьютер одно приложение, в котором реализованы оба эти стандарта, в MasterSCADA необходимо добавлять два отдельных элемента.

2.6. Дерево объектов

Дерево объектов расскажет о технологии. В нем, как в зеркале, отражается реальный технологический объект. От мастерства разработчика зависит, будет ли это зеркало прямым или искажающим действительность, потому что в нашем субъективном мире оно показывает объект глазами разработчика проекта. Один разработчик видит объект через призму его организационной иерархии, другой опирается на последовательность технологических переделов, а третий (он, как правило, заблуждается: этот подход хуже всех) раскладывает

⁵OPC-сервера разных производителей могут настраиваться по-разному. Перед использованием лучше ознакомиться с сопроводительной документацией на сервер.

Основные правила, действия и определения.

все по формальному признаку однотипности приборов КИПиА (насосы – к насосам, датчики давления – к датчикам давления).

Объект в MasterSCADA — это основная единица разрабатываемой системы, соответствующая (мы в этом примере опираемся на первый, наиболее наглядный, подход) реальному технологическому объекту (цеху, участку, аппарату, насосу, задвижке, датчику и т. п.), управляемому разрабатываемой с помощью MasterSCADA системой. С другой стороны, это и традиционный с точки зрения программирования объект, обладающий стандартными для программных объектов качествами.

Объект может внутри себя содержать другие объекты, а также переменные и функциональные блоки (служебные библиотечные объекты, предназначенные для контроля и управления; иногда они соответствуют объектам реального мира – например, насос или задвижка, – а иногда выполняют только одну функцию контроля или управления – например, регулятор). Любой из объектов MasterSCADA имеет свойства и документы, представляющие его для оператора. Можно создавать свои окна с динамической графикой (мнемосхемы), графики изменения параметров во времени (тренды), отчеты, журналы сообщений и другие документы.

Изучение MasterSCADA и проектирование в ее среде будет гораздо проще, если вы будете четко представлять себе, какой результат хотите получить. Это позволит вам вести проектирование по самой правильной методике: от общего к частному, или, как говорят программисты, «сверху вниз». На раннем этапе разработки проекта необходимо разбить реальный технологический объект на составные части – подобъекты.

Например, мы автоматизируем на заводе цех по производству продукта X. Посмотрим, какая иерархия объектов будет в этом случае. Есть главный объект – это «Завод»; заходим на его территорию и видим наш объект автоматизации – «Цех». У этого объекта должно быть свое графическое окно – мнемосхема, куда будут вынесены все важные параметры, характеризующие ход технологического процесса и состояние оборудования. Обычно мнемосхема отображает технологическую схему объекта, на которой оборудование представлено с помощью мнемонических или «жизнеподобных» изображений, изменение цвета, положения или формы которых характеризует их состояние, а параметры представлены в виде числовых значений или «щитовых» приборов. На уровне

Основные правила, действия и определения.

цеха показываются только важнейшее оборудование и основные параметры. Излишняя детализация здесь вредна.

Для цеха важно также построить отчеты, чтобы отследить, какое количество сырья поступило и сколько продукции было изготовлено. Разумеется, мы должны видеть изменение этих параметров во времени, а для этого предназначен такой документ, как тренд. Впрочем, его возможности гораздо шире, в чем мы еще в дальнейшем убедимся. В частности, он может показывать и зависимость одного параметра от другого.

И, наконец, еще один вид необходимых нам документов – это журналы сообщений. С их помощью мы можем отслеживать события на объекте: как технологические, например начало и конец технологических циклов и операций, так и внеплановые, аварийные: выход параметра за контрольную границу, несрабатывание задвижки и т. п. Обратите внимание, что документов каждого вида может быть несколько, в зависимости от того, как вы хотите структурировать представляемую информацию. Но одна мнемосхема всегда будет главной, это лицо объекта. Ее обычно удобно использовать для расположения на ней кнопок вызова других документов.

Двигаемся дальше. На территории цеха можно увидеть несколько технологических участков (или линий), каждый из которых отвечает за свой технологический передел. Для участка мы можем создать такой же набор документов, но уже с большей детализацией. В результате мы получаем дерево объектов, показанное на 2-8.

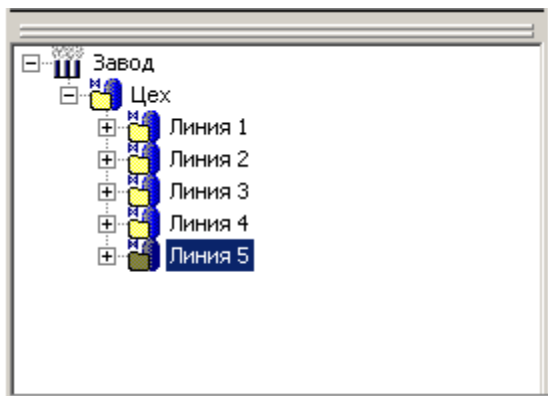


Рисунок 2-8. Дерево объектов

Итак, деление на подобъекты нужно производить до тех пор, пока соблюдаются два условия:

Основные правила, действия и определения.

- Можно выделить логически связанную группу переменных и функциональных блоков.
- Нужно создать для выделенной группы какие-либо документы, характеризующие ее как единое целое.

Важно! В настройках каждого объекта (страница свойств «Общие», поле «Компьютер») нужно определить, на каком компьютере он будет выполняться. Если данная настройка не сделана, это равносильно тому, что объекта в исполняемом проекте не существует.

Если необходимо по смыслу проекта объединить несколько переменных MasterSCADA, но при этом они не требуют никаких документов и никакой совместной обработки, то в этом случае лучше создать не дополнительный объект, а группу переменных.

Проект, как правило, на 80 процентов состоит из переменных объектов MasterSCADA (есть еще и переменные дерева системы). Казалось бы, их всего четыре типа: значение (→), расчет (↔), событие (⚡), команда (→), но, несмотря на это, они позволяют решить практически любые задачи современной автоматизации.

Расчет – используется для обработки данных с помощью математических формул, включающих, кроме действий, и функции из обширной встроенной библиотеки. Наряду с выполнением математических и логических (включая побитовые) действий над переменными (в том числе и с их архивными значениями) Расчет позволяет работать со временем, с признаками качества и нарушения границ переменных. Формула обработки в текстовом виде и список обрабатываемых переменных визуально задаются (Рисунок 2-9) на странице свойств **Формула**, а у переменной Расчет в дереве объектов в режиме исполнения показывается результат.

Основные правила, действия и определения.

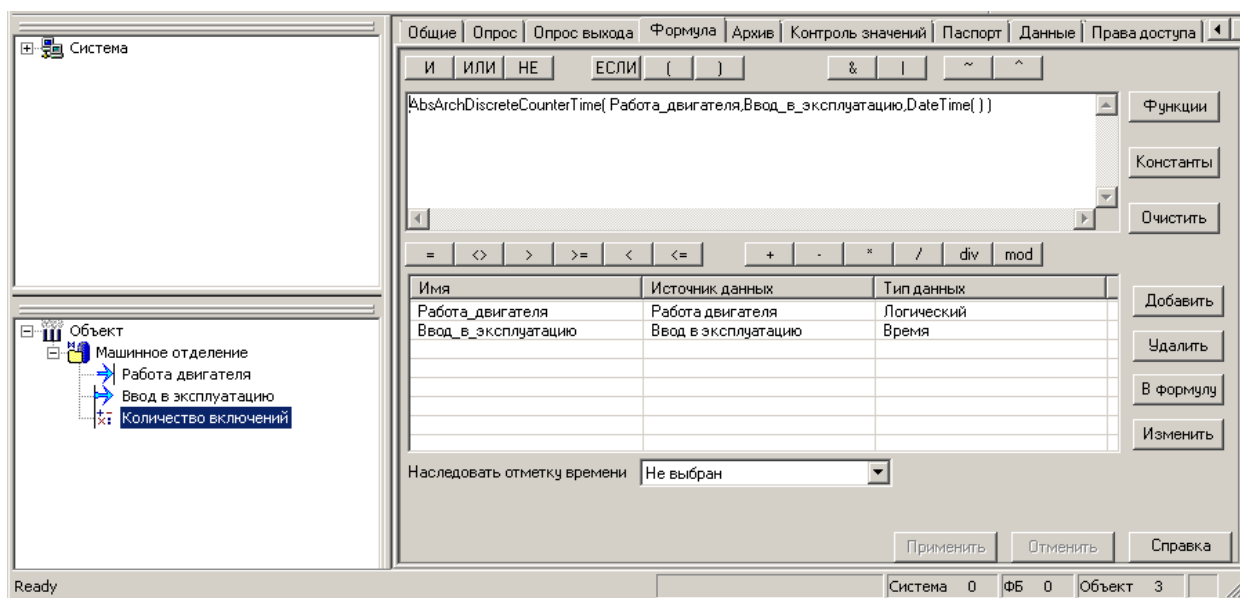


Рисунок 2-9. Страница свойств «Формула»

Значение – входная и выходная переменная MasterSCADA, то есть эта переменная является двунаправленной. Вход значения можно связать с выходами, командами, событиями, расчетами и другими значениями; выход – со входами (ФБ, расчетов, событий), а также вытаскивать на мнемосхему. Значение можно использовать в качестве повторителя, однако лучше использовать для данной задачи функциональный блок Повторитель (закладка **Обработка сигналов** палитры ФБ) или сервис внешних связей. Если перед дальнейшей передачей необходимо произвести обработку, то можно использовать модуль Расчет.

Команда – переменная, предназначенная для передачи значения, заданного оператором или присвоенного ей в результате выполнения действий (об этом мы подробно расскажем позднее), в другие переменные – например, в значения, выходы (в дереве системы) или расчеты. С помощью команды можно вводить уставки, включать или отключать оборудование, менять коэффициенты в формулах расчетов и т. п. Дискретная (0 или 1) команда позволяет на специальной странице свойств сформировать список действий, которые необходимо произвести, когда она сменит значение с 0 на 1.

Кроме того, команда может работать и в режиме с обратной связью. Данный режим может потребоваться, если уставка может меняться как из SCADA-системы, так и локально оператором. Чтобы использовать обратную связь, нужно выделить команду и перетащить в

Основные правила, действия и определения.

поле **Обратная связь** OPC переменную или выход **ФБ Повторитель**. В этом случае контроль команды будет отображать приходящее по обратной связи значение, а вводимые значения будут поступать на выход команды.

Событие отличается от расчета только тем, что результатом вычислений будет не число, а факт изменения логического значения из состояния **Ложь** (значение 0) в состояние **Истина** (значение 1). Момент перехода (как говорят в автоматике, «передний фронт сигнала») порождает необходимые действия или выдачу сообщения, как было задано разработчиком на странице свойств этой переменной. Событие имеет формулу, как расчет, страницу свойств **Действия**, как команда, но сообщение есть только у события. Именно поэтому основное назначение события – это порождение действий и сообщений при обнаружении какого-либо реального события в реальном мире.

Расчеты, команды и события – это выходы MasterSCADA. Их значения можно передавать по связям на входы – например, значениям, входам **OPC-сервера** или входам **Контроллера** в дереве системы. В последних двух случаях данные из деревя объектов будут передаваться в деревя системы и далее – на нижний уровень, к устройствам.

2.7. Функциональные блоки

Наряду с переменными MasterSCADA для создания логики работы проекта, обработки сигналов и решения других задач в деревя объектов используются функциональные блоки.

Функциональный блок⁶ имеет набор входов, выходов и встроенную логику работы, т. е. в режиме исполнения элемент обрабатывает поступившие на его входы данные по заданной программе, а результат отображается на его выходах. Некоторые функциональные блоки имеют также свое динамическое отображение в графических окнах (мнемосхемах), список генерируемых сообщений и ряд других свойств объектов – их мы будем называть визуальными функциональными блоками⁷. Для удобства разработчиков **ФБ** разделены по умолчанию на несколько категорий.

Категории **Обработка сигналов** и **Вычисления** служат для обработки аналоговых, дискретных и строковых величин. Большая часть **ФБ**, относящихся к категории **Вычисления**,

⁶ Далее – ФБ.

⁷ Далее – ВФБ.

дублируют возможности вычислений, реализованные в формулах переменных расчет и событие, поэтому используются, только когда процесс вычислений требует визуального представления в виде графической схемы функциональных блоков (об этом мы еще поговорим в дальнейшем). Однако обратите внимание на то, что формулы не поддерживают обработки строк. В этом случае соответствующие ФБ будут единственным вариантом обработки, за исключением написания программ на языках высокого уровня (об этом также далее).

Исполнительные механизмы – в этой категории нашли свое место наиболее часто встречающиеся в АСУТП исполнительные механизмы: задвижки, клапаны, насосы и т. п. Элементы представлены как в классическом мнемоническом виде, так и в псевдонатуральной графике. На первый взгляд может показаться, что данная категория немногочисленная, а перечень реального оборудования, логику работы которого должны описывать эти элементы, намного шире. Однако благодаря гибким настройкам один ВФБ MasterSCADA из категории **Исполнительные механизмы** может заменить десятки элементов. В некоторых ВФБ вид изображения выбирается из списка доступных (например, одна из разновидностей клапанов: муфтовый, фланцевый, диафрагменный и т. п.), но есть и универсальные, где выбор изображения остается целиком за пользователем, – это ВФБ **Двухпозиционный исполнительный механизм с выбором изображения**, а также ВФБ **Динамический ИМ**.

Датчики и Аппараты содержат ВФБ, которые удобно использовать при создании мнемосхем. По численности и востребованности они значительно уступают другим категориям палитры MasterSCADA. Так, **Датчики** фактически дублируют визуальные возможности обычных переменных MasterSCADA, а **Аппараты** часто сложнее представленных простейших случаев. Реальные аппараты реализуются возможностями объектов MasterSCADA и сохраняются разработчиками в их пользовательских библиотеках.

Очень часто развитие пакета зависит от высказанных пользователями пожеланий, поэтому если у вас есть необходимость в новых элементах, то, пожалуйста, обращайтесь к нам с любыми предложениями. Не забывайте, что интерфейсы создания новых ФБ открыты и вы можете сделать их самостоятельно. Во многих случаях проще создать и сохранить в библиотеке объекты проекта.

Основные правила, действия и определения.

Бывают ситуации, когда реальный сигнал необходимо заменить значением, изменяющимся по тому или иному закону. И хотя такая возможность предоставлена средствами формирования значений самих переменных (см. страницу свойств **Опрос**), тем не менее иногда нагляднее будет использовать в решении этой задачи категорию **Генераторы значений**.

ФБ группы **Работа со временем** помогут определить метку времени переменной, выдать импульс в определенный момент времени, и многое другое.

В системах, где MasterSCADA выполняет функции регулирования и управления, разработчики широко используют ФБ из категории **Управление: Циклограмма, Регулятор, Задатчик**.

Возможности записи и чтения из файлов, обращения к любому значению, сохраненному MasterSCADA, реализуются ФБ из категории **Работа с архивом**.

Для решения отраслевых задач в палитре функциональных блоков имеется категория **Отраслевые библиотеки**, которая включает в себя подразделы: **Теплоэнергетика, Вентиляция и кондиционирование, Электроэнергетика**. Безусловно, эта категория еще будет развиваться и пополняться новыми разделами.

Неограниченные возможности открывает перед разработчиком категория **Служебные**. ФБ этой категории позволяют не только определить, какой оператор работает в режиме исполнения в настоящий момент, сохранить какие-либо документы проекта или запустить внешнее приложение из интерфейса MasterSCADA, но и расширить стандартные возможности при помощи языков программирования ST (стандарт МЭК 61131-3) или C# (ФБ Скрипт).

Важно! Функциональные блоки «Скрипт» и «Программа ST» созданы для опытных пользователей с навыками программирования.

Вспомогательные ФБ и те, которые не нашли своего места по смыслу в других категориях, будут находиться в разделе **Другие**.

Важно! Одни и те же функциональные блоки могут находиться в нескольких категориях одновременно.

Основные правила, действия и определения.

Конечно, иногда встречаются задачи, которые сложно решить стандартными способами, но даже в этих непростых ситуациях мы не оставляем наших клиентов без поддержки. Если пользователь имеет достаточный опыт в программировании и время, то мы рекомендуем создавать свои собственные функциональные блоки, основываясь на документации, которая устанавливается совместно с любой версией MasterSCADA. Ее можно найти по адресу **Пуск – Все программы – MasterSCADA – Документация – Руководство по разработке ФБ и драйверов**. Если же у разработчика авральная ситуация, то мы можем и сами разработать под вашу задачу ФБ на доступных условиях⁸.

Пользователь может перераспределить элементы между категориями по своему усмотрению при помощи пункта главного меню программы **Сервис – Органайзер – ФБ**.

Важно! Прежде чем задействовать в работе ФБ, убедитесь, что поставленную задачу нельзя решить при помощи стандартных переменных MasterSCADA.

2.8. Общая структура проектов

Давайте подведем итог вышесказанному, чтобы выработать верный взгляд на разработку проектов, прежде чем увязнем в мелких деталях. Системный подход именуется такой «правильной» разработку методикой «сверху вниз».

Как мы уже говорили, проекты MasterSCADA состоят из двух разделов.

Первый из них называется **Система**, или, для наглядности, дерево системы. Он служит для описания состава, связей и настройки всех технических и системных программных средств, реализующих создаваемую нами систему автоматизации.

Второй раздел называется – **Объект**, или дерево объекта. И хотя он практически состоит из элементов всего двух видов: папок-объектов с комплектом принадлежащих им документов и настроек и переменных, – его роль в проекте является ключевой. Главное назначение дерева объекта – описание нашего представления об автоматизируемом

⁸ В любом случае, если вы столкнулись с какими-либо сложностями, не забывайте: у вас есть надежные помощники в лице службы технической поддержки, к которым вы всегда можете обратиться по электронному адресу support@insat.ru.

Основные правила, действия и определения.

объекте, логических и функциональных связях между его частями, а значит, и логики контроля и управления этим объектом.

Сможем ли мы структурировать это представление – проявить настоящий системный и одновременно объектный подход? Именно от этого в первую очередь зависит, хороший ли получится проект, будет ли он легко понятен кому-либо, кроме автора, насколько трудоемкой окажется его разработка.

Здесь важнее всего следовать принятому в программировании принципу самодостаточности элементов проекта (инкапсуляции), что применительно к предмету нашего разговора означает: количество внешних связей любой папки объекта нужно стремиться сделать минимальным. Такая минимизация чаще всего достигается при простом следовании реальной организационной или технологической структуре нашего автоматизируемого объекта.

Рассмотрим пример организационной структуры в порядке иерархической подчиненности: завод – цех – участок (линия) – агрегат – исполнительный механизм (датчик). Попытка построить проект по функциональному признаку, например сгруппировать все аналоговые параметры в одну папку, почти всегда приводит к избыточным связям между частями проекта, а следовательно, к его неэффективности. Мало того, жизнь показывает, что функциональный подход приводит еще и к огромным сложностям в развитии и модифицировании проектов. В то же время отдельная папка-объект в хорошо структурированном по объектному принципу дереве объекта автономна в такой степени, что может быть перенесена без изменений в любой другой проект, где возникнет необходимость в работе с подобным объектом. Например, однажды реализовав в качестве элемента проекта группу насосов, мы сможем переносить ее вместе со всей логикой контроля и управления, со всеми сигналами и документами в любой другой проект, где она потребуется. Единственное, что останется сделать, – это восстановить связи с источниками и приемниками реальных физических сигналов, а также организовать вызов документов этого объекта из мнемосхем более высокого уровня.

Подробнее обо всем этом – в дальнейших главах.

Основные правила, действия и определения.

2.9. Выбор порядка разработки системы: от объекта или от системы

Этот выбор не является жестким, но от него зависят удобство и скорость разработки проекта. Разработка структуры технических средств в проекте целесообразна только тогда, когда вы полностью определились с нею в жизни, когда известно, по каким каналам связи передаются данные, какие выбраны контроллеры и модули расширения, какой физический сигнал к какому входу или выходу подключен. Но, если все перечисленное находится еще в сыром виде, ничего страшного – вы можете начать разработку проекта, стартовав от структуры автоматизируемого объекта, его функциональных связей, логики его контроля и управления, то есть формируя дерево объекта.

MasterSCADA позволяет разрабатывать проект в любом порядке, в том числе и параллельно. Если это делают разные разработчики на разных компьютерах, то удобнее, чтобы «головной» проект, в который будут потом копироваться отдельно разработанные части, находился у того, кто отвечает за дерево системы. Не забудьте, что проектирование – это итерационный процесс и «стыковать» имеет смысл только полностью законченные и отлаженные части.

Отдельно разработанные части проекта требуют установления связей друг с другом. О передаче данных по связям мы сейчас и поговорим.

2.10. Особенности создания связей между деревьями системы и объектов

Связи, как и большинство других действий в MasterSCADA, настраиваются «мышью». Различают два типа связей: прямые, когда данные от переменной-выхода без какой-либо обработки сразу же поступают к переменной-входу, и косвенные – например, когда одна переменная является источником для аргументов формулы расчета или события.

Важно! Один вход напрямую можно соединить только с одним выходом, а один выход может быть связан с несколькими входами.

Итак, чтобы организовать прямую связь между переменными проекта MasterSCADA, нужно перетянуть, удерживая ЛК, одну переменную на другую. В месте, где связь может быть установлена, обозначение манипулятора примет вид «/»; отпустив ЛК, вы увидите, что к

Основные правила, действия и определения.

стандартному обозначению переменной добавится розовая вертикальная черта – это говорит о том, что связь установлена (*Рисунок 2-10*).

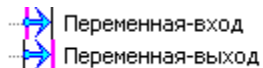


Рисунок 2-10. Обозначение прямых связей

В случае если разработчик проекта пытается установить связь, при которой возможна потеря точности (например, между вещественным двойной точности выходом дерева объекта и входом дерева системы целого типа), MasterSCADA отследит это и выдаст предупреждение.

Для того чтобы разорвать прямую связь, достаточно нажать ПК⁹ на переменной и в появившемся контекстном меню выполнить пункт **Разорвать связь**.

Если необходимо связать переменную любого из деревьев с аргументом формулы события или расчета, то в этом случае алгоритм будет таким:

- Выделите переменную событие или расчет в дереве объектов.
- Перейдите на страницу свойств **Формула**.
- Перетащите ЛК нужную переменную в таблицу страницы свойств.

После этого вы сможете использовать значение переменной в формуле. Способ создания формул рассмотрим в одной из следующих глав.

Удалив переменную из таблицы страницы свойств **Формула**, вы тем самым разорвете связь.

Чтобы оперативно определить, какие связи имеет та или иная переменная, достаточно выделить ее в дереве и перейти на страницу свойств **Список связей**. На данной странице отображаются как прямые связи, так и косвенные.

Создание связей в проекте – очень важная задача. Решая ее, самое главное – не допустить ошибок! MasterSCADA позволяет перетаскивать переменную в дерево объектов из дерева системы несколько раз. На первый взгляд это может показаться разработчику удобным, но правильно ли это? Давайте разберем ситуацию: проект готов к внедрению, но в момент пуска наладки отказывает какой-либо модуль ввода/вывода. Экстренно

⁹ Здесь и далее ПК – «правая кнопка мыши».

Основные правила, действия и определения.

производится замена, но вместо используемого модуля устанавливают сходный по свойствам прибор другого производителя. Это повлечет за собой смену конфигурации ОРС-сервера и, как следствие, изменение дерева системы. Если каждая переменная дерева системы будет связана с несколькими переменными дерева объектов, то потребуется значительно больше времени, чтобы восстановить проект. Для решения данной задачи можно использовать сервис внешних связей или ФБ Повторитель.

Важно! Связь между переменной дерева системы и переменной дерева объектов должна быть только одна! Внутри объектов ограничений по связям нет.

При копировании, вставке из палитры переменных или объектов, в случае если они имеют внешние связи, автоматически появится окно восстановления связей.

Основа построения логики контроля и управления в MasterSCADA – это передача данных по связям между переменными. Например, от аналогового входа модуля (дерево системы), к которому подключен физический датчик уровня в баке, значение передается в блок первичной обработки параметра, принадлежащий объекту **Бак** (дерево объекта), а от этого блока – в документы данного объекта (мнемосхему, тренд, журнал сообщений и т. п.).

Еще раз обратим внимание, что для успешной разработки каждого проекта очень важно как можно быстрее понять, какие связи являются устойчивыми, а какие могут меняться в процессе разработки или модификации проекта.

Вернемся к нашему примеру. Значение уровня необходимо отображать на мнемосхеме бака, а при его выходе за установленные границы формировать сообщения. Отсюда следует, что все вышеперечисленное должно принадлежать объекту **Бак**. А вот источник сигнала в виде аналогового входа модуля может меняться – сейчас мы подключились к одному входу, а при его неисправности можем «перекинуть концы» от физического датчика уровня и на соседний свободный вход. Отсюда вывод: такая связь является внешней и должна легко устанавливаться в одном-единственном месте. Если бы мы связали сигнал от аналогового входа модуля отдельно с каждой обработкой или документом, где он используется, то процедура замены стала бы трудоемкой. Но мы сделаем в рамках объекта одну-единственную переменную **Уровень**, которая и будет отвечать за связь с реальным источником сигнала, а остальные его потребители получат значение уже от нее. Тем самым замена сигнала, используемого во многих обработках или

Основные правила, действия и определения.

документах, сведется к замене единственной связи, если мы изначально правильно сгруппировали объекты и переменные.

2.11. Базовые документы объектов

Мы уже отмечали, что для каждого объекта можно создавать мнемосхемы (графические окна), тренды, журналы сообщений, отчеты (рапорты). В дальнейшем мы отдельно подробнее поговорим про каждый документ. Цель этого раздела – общий их обзор. MasterSCADA имеет как predetermined типы документов, так и возможность создавать на базе некоторых из них (мнемосхем, журналов сообщений) типизированные новые виды окон пользовательского интерфейса.

Начнем с графических окон. Поскольку в реальных проектах одному и тому же объекту необходимо для разных целей иметь несколько графических окон (мнемосхему, окно управления, окно настроек и т. п.), программа предоставляет возможность их типизации в рамках проекта. На странице свойств Окна корневого узла дерева системы (Рисунок 2-11) определяются полный список типов графических документов и их базовые настройки: размер по умолчанию (задается в пикселях), способы открытия и отображения окна в режиме исполнения.

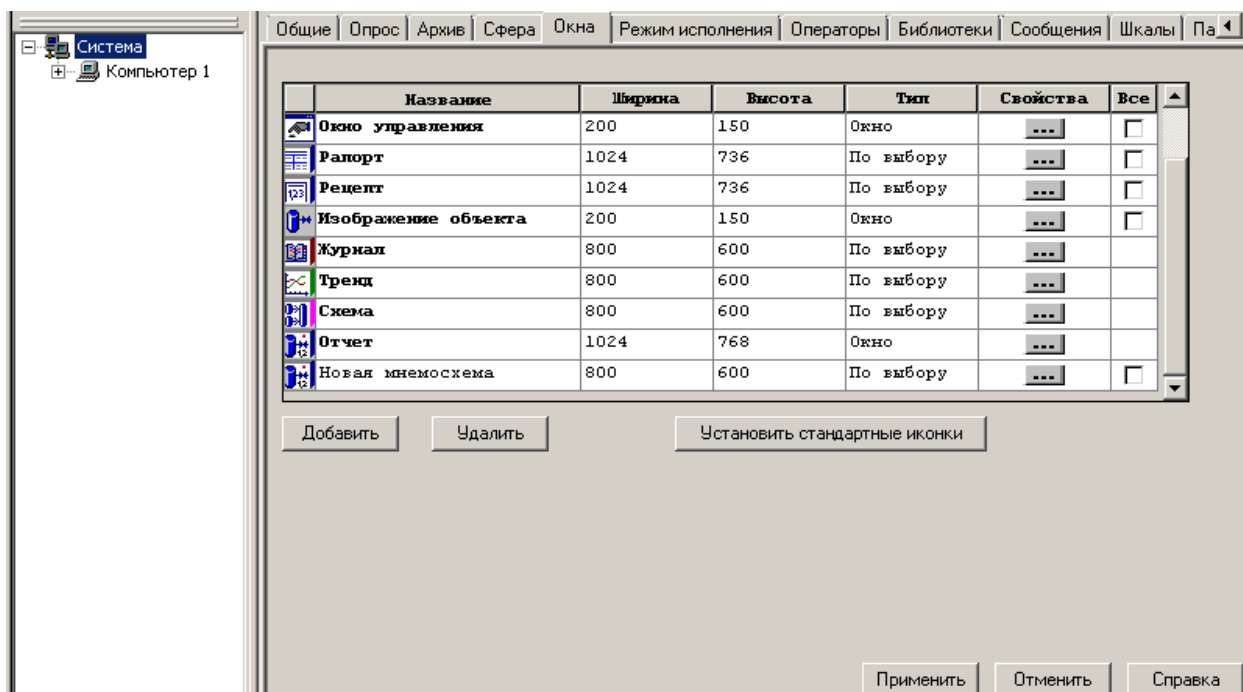


Рисунок 2-11. Страница свойств системы «Окна»

Основные правила, действия и определения.

Когда мы захотим сделать графическое окно для конкретного объекта, то на его странице свойств **Окна** мы сможем выбрать нужный (по размеру, способу вызова и отображения) тип окна. Изначально для примера задано несколько типов. Это, разумеется, **Мнемосхема**, а также **Окно объекта**, **Окно управления** и другие документы, список которых показан на . Вы всегда можете расширить этот список, вернувшись на страницу **Окна дерева системы**.

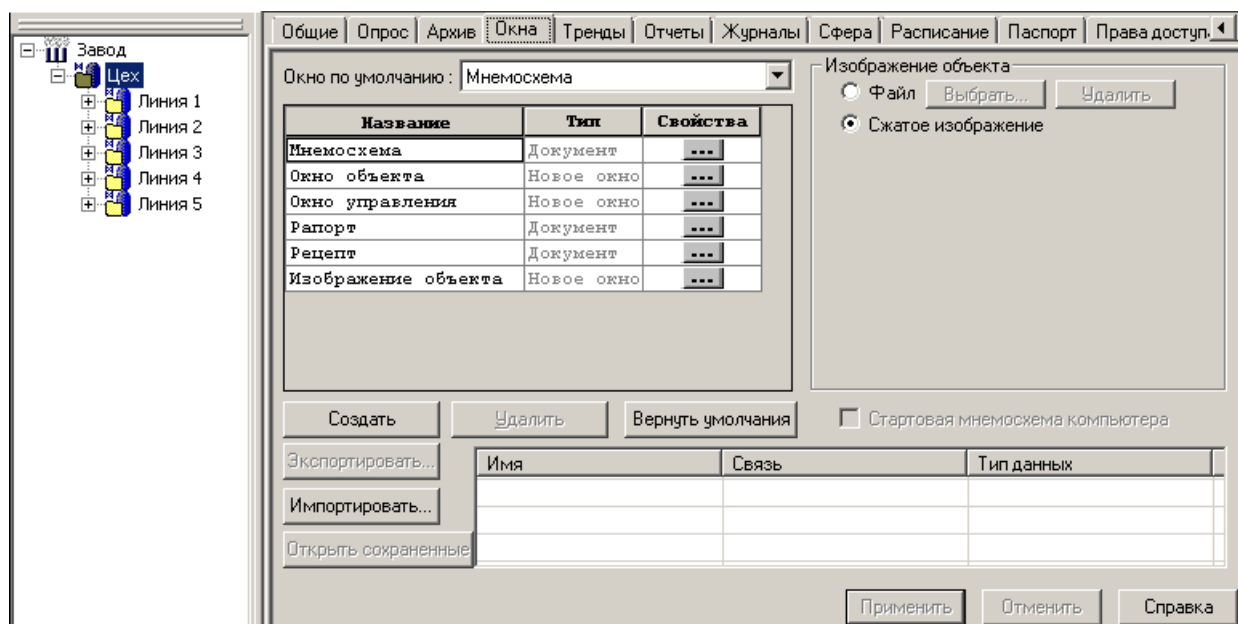


Рисунок 2-12. Страница свойств объекта «Окна»

Наиболее часто разработчик начинает с создания мнемосхемы. Итак, чтобы создать мнемосхему объекта, необходимо перейти на страницу свойств **Окна**, выделить ЛК в таблице нужный документ и нажать кнопку **Создать**. Если мнемосхема уже была создана ранее, то, чтобы ее открыть, нажмите кнопку **Редактировать**.

Интерфейс изменился (Рисунок 2-13) – по-прежнему видны два дерева, но вместо страниц свойств появилось белое поле, на котором и будет создаваться картинка мнемосхемы, над этим полем – дополнительная панель инструментов, а справа появились пять вспомогательных панелей: **Элементы**, **Входы**, **Выходы**, **Палитра**, **Свойства**.

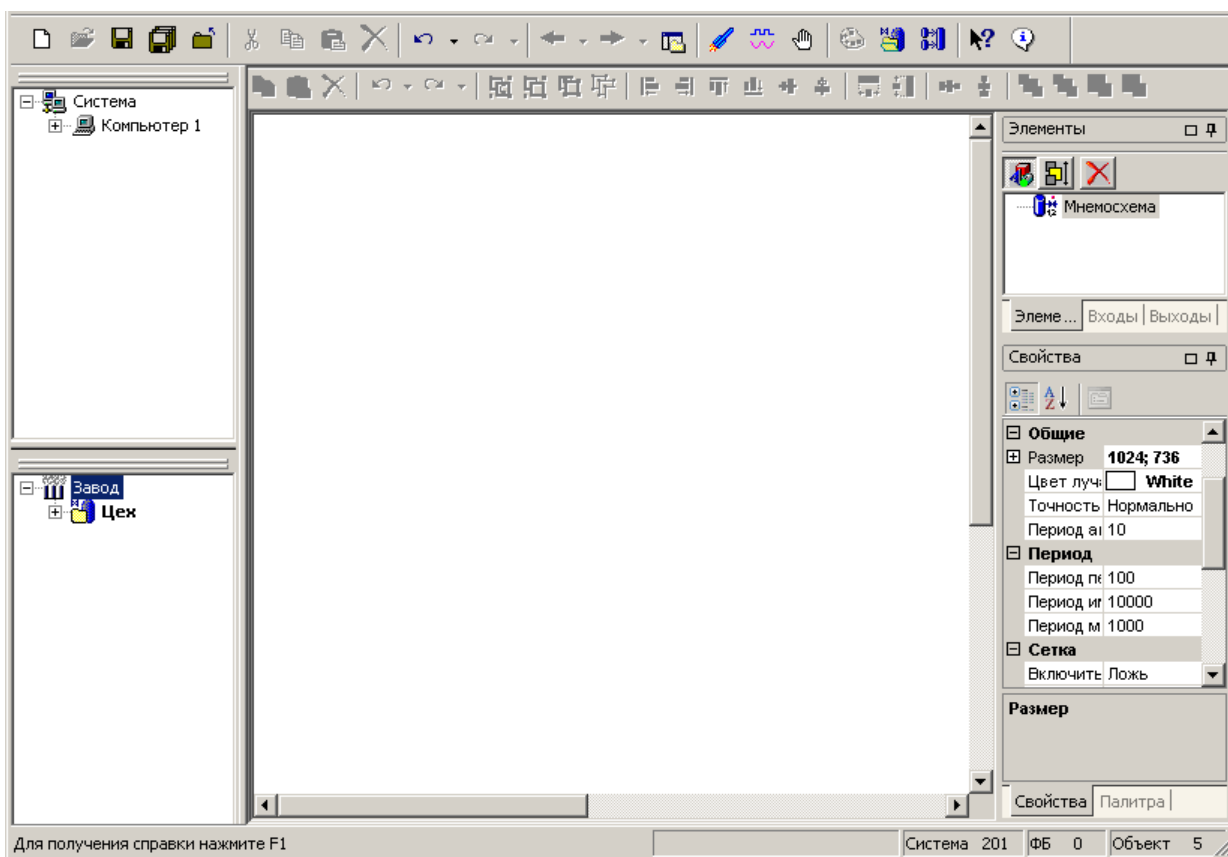


Рисунок 2-13. Интерфейс графического редактора

Особенность взаимодействия графического редактора с менеджером проектов заключается в том, что в большинстве случаев для создания динамической картин-мнемосхемы достаточно перетащить в поле редактора элементы проекта: переменные, ВФБ, объекты. Если выполнить перенос из дерева на мнемосхему при помощи ЛК, то тогда элемент примет вид по умолчанию. Если правой кнопкой мыши – то в зависимости от типа элемента MasterSCADA предложит выбор графического отображения. Никакой настройки после перетаскивания не требуется.

Чтобы создать полноценную мнемосхему с изображением объекта и сколь угодно сложными способами динамизации поведения элементов изображения в зависимости от изменения значений переменных, воспользуемся панелью графического редактора **Палитра** и средствами динамизации любых свойств элементов мнемосхемы. Подробнее это будет описано в специальном разделе.

Вернемся к вопросу типизации документов системы. Рассмотрим теперь журнал сообщений. Если посмотреть на список типов журналов на одноименной странице корневого

узла дерева системы, где уже есть полный журнал, журналы активных и неактивных сообщений, то может показаться, что их достаточно. Но фантазия заказчиков автоматизированных систем неисчерпаема, поэтому возможность создать еще несколько типов никогда не будет лишней. А отличие нового типа будет прежде всего в том, какие сообщения попадут в этот журнал, а какие пролетят мимо. Создав новый тип, в дальнейшем, как и в случае с новым типом окна, мы сможем использовать его в любом объекте. Как несложно догадаться, чтобы создать для объекта журнал сообщений нужного типа, нужно перейти на страницу свойств объекта **Журналы**, выбрать тип и нажать кнопку **Создать**. По большому счету, никаких дополнительных настроек можно больше не делать. В режиме исполнения в этот журнал будут попадать сообщения, удовлетворяющие условиям фильтрации, заданным нами для данного типа журналов.

Важно! Журналы также существуют у «Системы» и «Компьютера». В эти журналы будет попадать информация со всех объектов. Они создаются автоматически, в отличие от журналов объектов.

Рассмотрим еще один тип документов – тренды, или графики параметров.

Чтобы создать такой график, необходимо на странице свойств объекта **Тренды** нажать кнопку **Добавить** и перетащить в открывшееся поле тренда нужные переменные. Для трендов нет пользовательской типизации – только предустановленные уровни сложности. Но каждый отдельный тренд можно подогнать «под себя», выбрав **Продвинутый** уровень, на котором доступно использование всех настроек, позволяющих полностью преобразовать внешний вид и функциональность тренда.

Важно! Тренды, журналы и таблицы значений за период времени – это взаимосвязанные виды документов, которые имеют возможность синхронизированного просмотра, взаимного переключения и другого взаимодействия, основанного на доступе к сохраненным данным за прошедшее время.

Современные системы автоматизации, не говоря уже об учете ресурсов, не обходятся без такого типа документов, как отчет. Для создания отчетов можно воспользоваться встроенным редактором отчетов MasterSCADA. Создание отчета производится на странице объекта **Отчеты**. Возможности встроенного генератора отчетов MasterSCADA не

уступают ведущим специализированным продуктам этого назначения, поэтому при постановке задачи можно не задумываться о возможности ее решить – не сомневайтесь, средства для этого вполне достаточны. Создание отчетов мы рассмотрим в отдельной главе.

Важно! Тренды, отчеты, журналы в MasterSCADA – это средства просмотра архивных данных в удобном для пользователя виде. Чтобы они отображали информацию, необходимо убедиться, что установлены корректные значения параметров архивирования на страницах свойств «Архив» проекта.


Про создание различных документов можно говорить бесконечно, потому что количество создаваемых форм не ограничено, но все же в дальнейшем мы постараемся раскрыть все тайны графического редактора, модуля для создания трендов и журналов и, конечно же, редактора отчетов.


Помимо базовых документов, создаваемых встраиваемыми редакторами MasterSCADA, объекты могут иметь и другие документы, для работы с которыми используются программы других производителей, открывающиеся в интерфейсе MasterSCADA наравне с ее собственными редакторами (для этого они должны соответствовать программным спецификациям OLEDocument). Например, оператор может просмотреть паспорт оборудования в формате **Adobe Acrobat**, чертеж в **AutoCAD**, схему в **Visio**, сделать заметки в **Word**, просмотреть пошаговую инструкцию, разработанную в **PowerPoint**. Причем для этого не потребуется знать имя файла или место хранения документа – все они будут привязаны к объектам по своим типам. Для такой привязки необходимо сначала подключить внешние программы к системе, для чего воспользоваться пунктом главного меню **Сервис – Органайзер – Документов**, определить типовые названия для документов подключаемых программ – например, назвать документ **AutoCAD** «Чертеж». Далее на страницах свойств объектов **Другие** вы получите возможность создавать и подключать к объекту документы тех типов, которые вы определили, а вызывать эти документы в режиме исполнения оператор сможет с помощью всего арсенала средств вызова документов в MasterSCADA.

2.12. Запуск режима исполнения

Создание проекта – дело, конечно, интересное, но ничто не вдохновляет инженера так, как собственное творение в работающем виде. Насладиться проделанной работой можно в режиме исполнения MasterSCADA, а заодно и отладить разработанный проект, прежде чем запустить его в рабочем режиме контроля и управления технологическим процессом. Режимом исполнения называют состояние, когда проект, который вы создали, начинает работать, используя созданную структуру и настройки для интерпретирующих программных модулей MasterSCADA.


В MasterSCADA различают три типа режима исполнения: **Отладка, Имитация, Пуск**.

В процессе разработки наиболее часто используется режим **Отладка**. Он позволяет запустить программу на исполнение максимально быстро на любом этапе создания проекта. При старте не требуется идентификации оператора. Проект запускается полностью на одном компьютере – даже в случаях, когда предполагается сложная распределенная архитектура. Этим режимом можно воспользоваться только из среды разработки. Наиболее часто для запуска проекта из DT используют панель инструментов. Для запуска режима **Отладка** необходимо нажать на иконку  .

Рабочий режим исполнения проекта называется **Пуск**, для его запуска из среды разработки используйте иконку  . При запуске потребуются инициализация оператора. Этот режим необходимо запускать на всех компьютерах системы. Программа производит реальный опрос и формирование команд. Режим **Пуск** можно запустить и минуя среду разработки – например, создав ярлык на рабочем столе, из командной строки (команда **Выполнить** среды Windows) или заранее сформировав командный файл Windows. В строке запуска нужно прописать путь к файлу запуска MasterSCADA (**MasterSCADA.exe**) и путь к исполняемому файлу проекта ([имя проекта].wav). Синтаксис команды будет таким: **<путь к MasterSCADA.exe><путь к файлу проекта><параметры>**¹⁰. Если необходимо запускать проект на исполнение сразу же после загрузки компьютера, то ярлык нужно поместить в группу автозапуска (**Пуск – Все программы – Автозагрузка**).

¹⁰ Со списком возможных параметров можно ознакомиться в разделе справочной системы «Работа в режиме исполнения – Запуск проекта из командной строки».

Основные правила, действия и определения.

Режим **Имитация** (иконка ) используется в ситуациях, когда проект уже практически создан, но есть еще ряд незадействованных связей. Он запускается только из среды разработки и полностью подобен режиму **Пуск** – за исключением того, что все «оборванные» связи заменяются имитационными сигналами. Вид сигнала будет зависеть от типа переменной и выбора закона имитации (по умолчанию предлагается синусоида для аналогового сигнала и пульсатор для дискретного).

Интерфейс MasterSCADA в режиме исполнения будет зависеть от выбранного режима, а также от настроек проекта.

В режиме **Отладка** вы увидите деревья MasterSCADA и страницы свойств. Рядом с деревьями появится колонка, в которой можно отследить изменения переменной; значение переменной будет отображаться напротив названия переменной (*Рисунок 2-14*). Кликнув ЛК, вы сможете задать любое значение любой переменной. В процессе создания проекта это допустимо, т. к. позволяет, например, посмотреть вид параметра в режиме исполнения на мнемосхеме или посмотреть, как работают формулы расчетов.

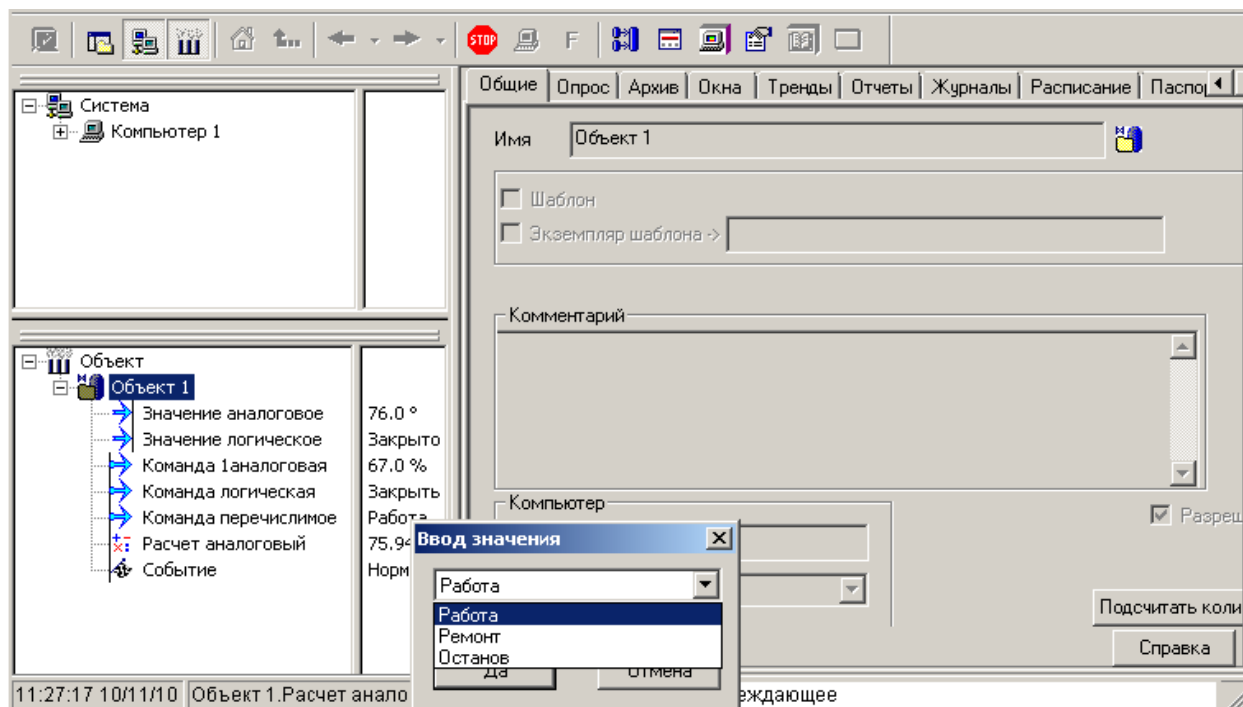


Рисунок 2-14. Менеджер проектов в режиме исполнения

Важно! В режиме исполнения колонка со значениями переменных перерисовывается не чаще чем раз в секунду. Поэтому отследить реальное значение, а

Основные правила, действия и определения.

также скорость изменения переменной в данном поле нельзя. Самую точную информацию о периоде изменения переменной, значениях, которые были при каждом изменении за последние несколько минут, и о ее признаке качества можно получить на странице свойств «Данные выхода».

Переход к любым документам объектов можно осуществлять через контекстное меню или страницы свойств.

Интерфейс после старта режимов *Пуск* и *Имитация* будет зависеть от тех настроек, которые вы сделали на странице свойств системы или компьютера *Режим исполнения* (Рисунок 2-15).

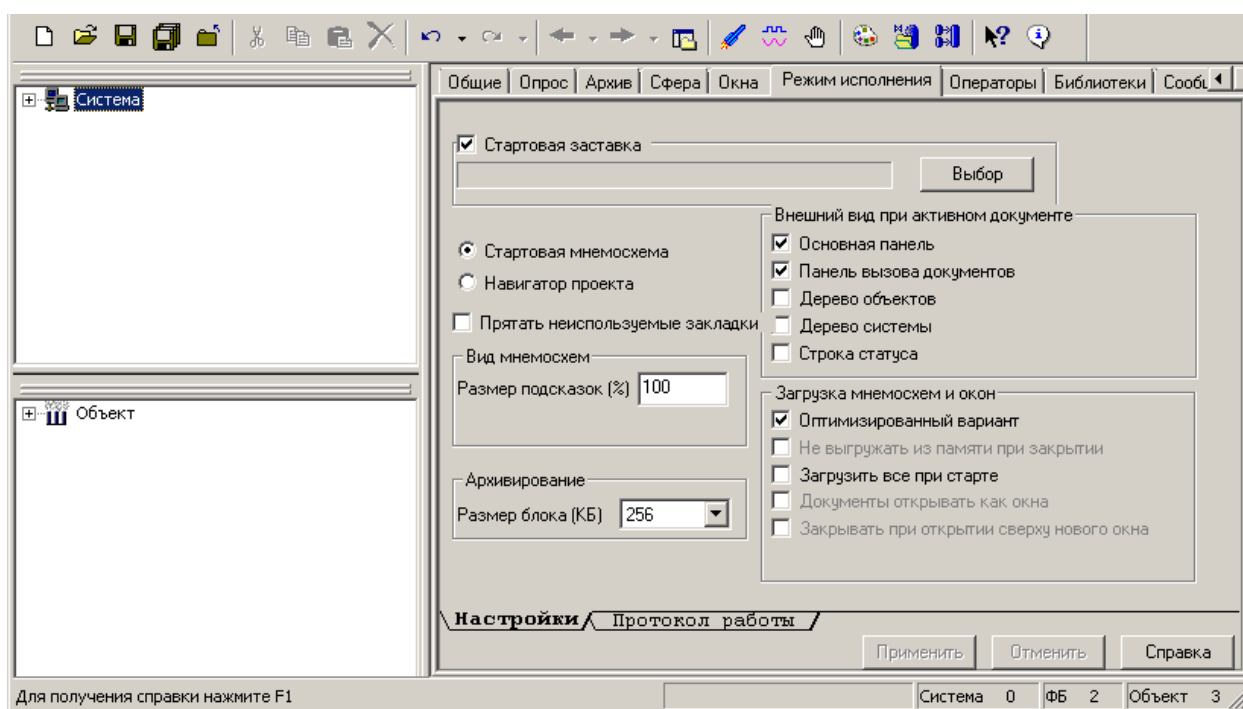


Рисунок 2-15. Настройка вида после старта режима исполнения

Как правило, для старта используют вариант с отображением стартовой мнемосхемы, которая может быть индивидуальной для каждого компьютера в системе. Эта мнемосхема загрузится сразу же после запуска системы на исполнение. Хотя возможен и такой же вид после старта, как и в режиме *Отладка*. Однако этот интерфейс может быть небезопасным, так как оператор сможет изменить через деревья значения переменных, а также увидит все внутреннее содержание проекта. Обычно такой вариант запуска используют только для пусконаладчика или инженера, обслуживающего систему.

Основные правила, действия и определения.

В любом из режимов, даже получив доступ к содержимому проекта, его можно только просматривать без изменения – нельзя изменять формулы расчетов, добавлять или удалять переменные, изменять периоды опросов объектов, OPC-серверов и т. п. Поэтому страницы свойств в режиме исполнения не активны, палитра функциональных блоков в режиме исполнения не отображается.

Стартовую мнемосхему компьютера можно создать специально на его собственной странице **Окна**, но чаще в качестве стартовой назначают одну из мнемосхем объектов, которые принадлежат данному компьютеру.

Очень важным для работы в режиме исполнения является способ навигации по проекту, то есть способ вызова его документов. Для этого есть как системные возможности, так и кнопки вызова, созданные пользователем на мнемосхемах в соответствии с собственной логикой. Но об этом мы поговорим подробнее позже.

2.13. Операторы в режиме исполнения и их права

Оператор в режиме исполнения может выполнять различные действия, предусмотренные разработчиком проекта: задавать значения, просматривать документы объектов, сохранять, формировать отчеты и многое другое. Но всегда нужно, разрабатывая систему, задавать себе вопрос: «Каждый ли оператор, работающий в режиме исполнения, должен иметь полный набор прав доступа?»

В MasterSCADA имеется гибкая система настройки прав доступа. Возможности оператора определяются его должностью. В режиме разработки на страницах свойств системы **Операторы и безопасность** (Рисунок 2-16) необходимо определить список должностей, а затем для каждой настроить права доступа.

Запрет того или иного действия зависит от состояний флагов **Лок** и **Глоб**. **Лок** («локально») определяет допустимость выполнения действия на объекте, исполняемом на компьютере оператора. **Глоб** («глобально») – применяется, если проект является сетевым, и определяет допустимость данного действия на объекте, исполняемом на другом компьютере.

Устанавливая или убирая флаг напротив того или иного действия, вы тем самым разрешаете или запрещаете это действие операторам, которые будут работать в

Основные правила, действия и определения.

настраиваемой должности. Например, сотрудник, работающий в должности **Оператор**, права **Отключить режим исполнения** может не иметь, в отличие от его коллеги – **Начальника смены**. Зачастую требуется, чтобы каждое действие оператора фиксировалось в журнале действий, в таком случае необходимо установить у действия флаг **Журнал**. Также можно установить флаг **Сообщение**, и действия будут фиксироваться в стандартный журнал сообщений – с категорией **Действие оператора**. Во избежание случайного нажатия можно установить флаг **Подтверждение** – в этом случае после действия появится специальное окно, в котором оператор должен будет подтвердить (или отменить) выполненное им действие.

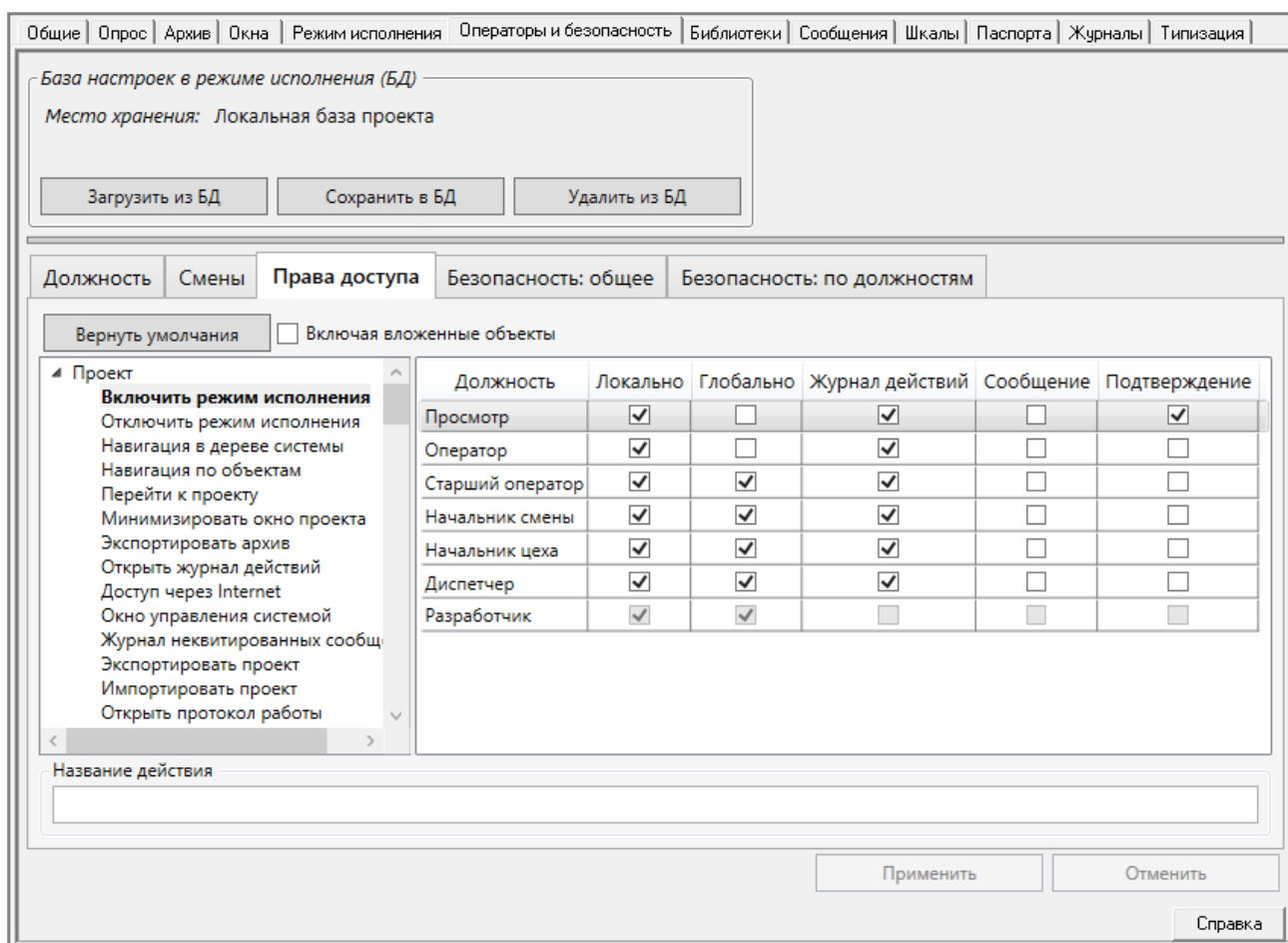


Рисунок 2-16. Настройка прав доступа операторов

После того как вы определили возможности той или иной должности, нужно ввести в проект логины и пароли каждого конкретного оператора. Они настраиваются на странице свойств **Операторы и безопасность** компьютера. Выберите должность, в которую вы хотите

Основные правила, действия и определения.

ввести сотрудника, и нажмите кнопку **Добавить**. Отредактируйте имя сотрудника – например, замените появившееся по умолчанию **Оператор1** на «Иванов Петр Сидорович». Затем, не снимая выделения с поля, где вводили имя сотрудника, нажмите кнопку **Сменить пароль**. В окне, показанном на 2-17, установите нужную комбинацию символов. Если эти поля останутся пустыми, то вход в режим исполнения **Пуск** или **Имитация** для данного оператора будет свободным, без пароля.

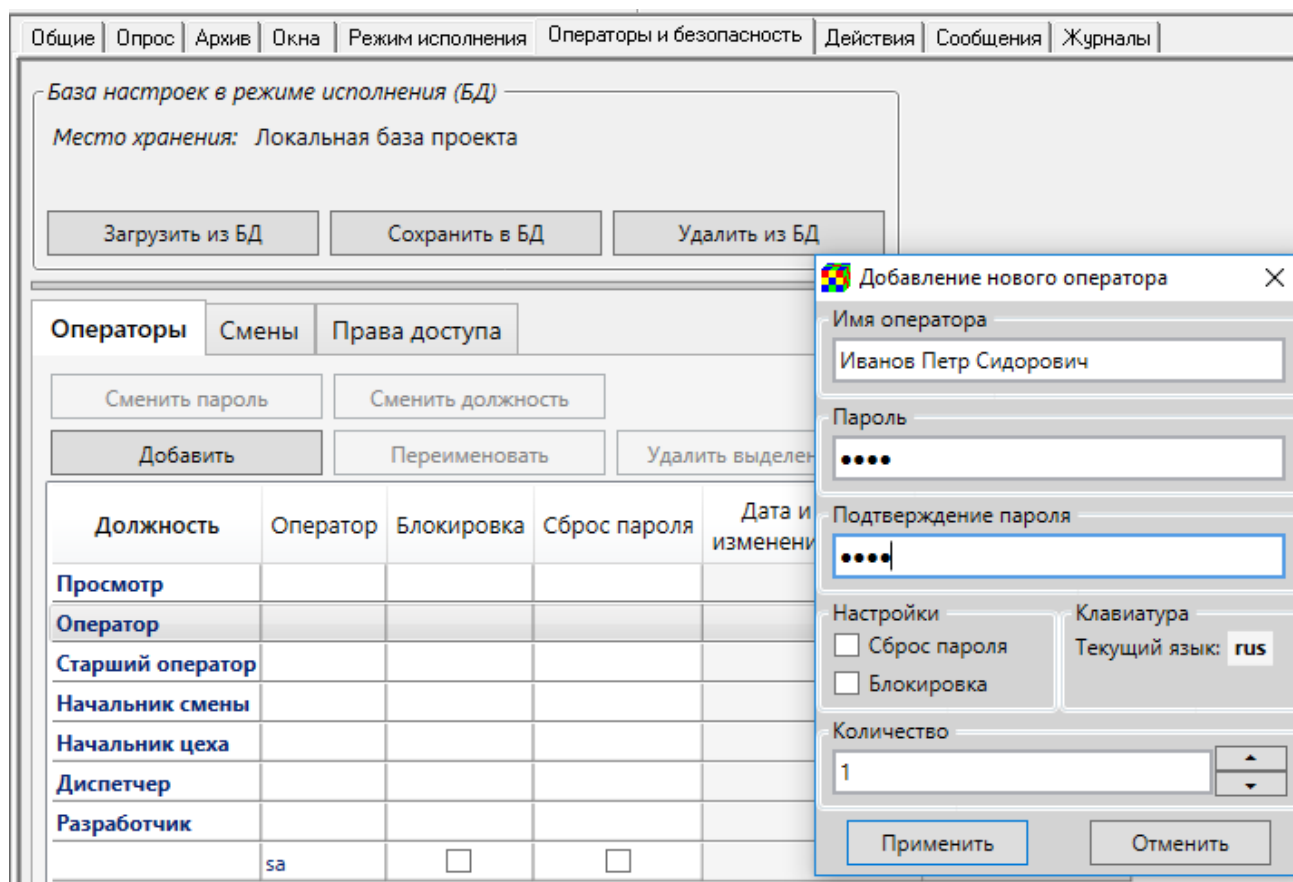


Рисунок 2-17. Назначение оператора в систему

3. Объектно-ориентированный подход

Перейдем непосредственно к методам построения проектов в MasterSCADA.

При разработке проектов MasterSCADA рекомендуется придерживаться объектно-ориентированного подхода. В чем его суть?

В «классических» SCADA-системах (без объектного подхода) компоненты проекта находятся в определенных разделах: отдельный раздел окон, отдельный раздел графиков, отдельный раздел с функциональными блоками и скриптами. Некоторые начинающие пользователи MasterSCADA, ранее работавшие с другими SCADA-пакетами, пытаются и в MasterSCADA перенести подобную архитектуру, раскладывая в отдельные объекты разные элементы проекта. Это в корне неправильно.

Объект в MasterSCADA – это эквивалент реальной технологической единицы (цеха, установки, насоса, клапана и т. д.). При этом объект содержит как все параметры этой единицы (температура, давление, состояние механизмов), так и атрибуты: мнемосхемы, окна управления, виджет для отображения на мнемосхеме верхнего уровня, графики, отчеты и т. д.

Что дает объектный подход? Первое и очевидное преимущество – упаковка всех свойств и атрибутов элемента в единый объект. То есть, например, все параметры и свойства насоса находятся в одном объекте, а не разбросаны в разных частях проекта. В программировании такой подход называется «инкапсуляцией».

Второе преимущество – тиражирование. Разработав объект (например, насоса) – создав для него все необходимые входные и выходные переменные, окна управления, графики и т. д., – можно применять его в остальных частях проекта. Аналогично поступают и с другими элементами проекта: клапанами, задвижками, агрегатами или даже целыми цехами. При этом если нужно будет внести правки в работу объекта (добавить новые переменные, изменить настройки), то внести правки в остальные не составит труда – для этих целей в MasterSCADA есть специальный механизм – **Шаблон-Экземпляр**.

Третье преимущество вытекает из второго – многократное использование разработанного объекта в различных проектах. Зачастую разработчик систем автоматизации работает в определенной отрасли, и различные проекты в целом похожи: состав

Объектно-ориентированный подход

оборудования примерно одинаковый, но может отличаться компоновка или количество устройств. В таких ситуациях очень удобно иметь наработанную базу всех узлов и агрегатов и, используя их, собирать собственный проект, как из конструктора. В MasterSCADA для этих целей предназначена библиотека – в нее помещаются элементы из одного проекта, которые затем можно использовать в других. При этом наследование сохраняется: если в библиотеке элемент будет изменен, то правки можно будет быстро внести в реализованные проекты.

Таким образом, объектный подход позволяет, создав базу из шаблонных объектов-узлов, быстро собирать из них агрегаты, из агрегатов – цех, а их цехов собирать целый завод. При этом в библиотеку для дальнейшего использования может быть помещен любой объект (как отдельный насос, так и целый цех) – это полностью зависит от типизированности вашего оборудования.

Например, модульная котельная состоит из следующих узлов (состав может варьироваться):

1. Котлы с количеством от двух до десятка.
2. Резервированная пара циркуляционных насосов внутреннего контура.
3. Теплообменники – для отопления и ГВС (может отсутствовать).
4. Насосы подачи.

Таким образом, четко видна объектная структура будущего проекта. Можно создать отдельные объекты котлов, насосов, теплообменника, а затем собирать из них будущие проекты. При этом если в одном проекте котлов всего 2 и нет теплообменника, а в другом котлов уже 4 и использованы два теплообменника, то построение проекта в обоих случаях будет быстрым – наполнить дерево объектов нужными объектами из библиотеки, расставить связи (расстановка может быть автоматизирована) и скомпоновать добавленные элементы на мнемосхеме.

Рассмотрим описанные принципы объектного подхода на конкретном примере: допустим, имеется узел перекачки жидкости, состоящий из насоса с частотным преобразователем и расходомера.

3.1. Инкапсуляция

Начнем рассмотрение объектного подхода с инкапсуляции – упаковки элементов в объект с созданием свойств и атрибутов. Итак, входными сигналами являются:

1. Состояние насоса: включен или выключен.
2. Состояние аварии насоса.
3. Выходная частота с частотного преобразователя.
4. Расход жидкости.

Атрибутами данного объекта могут быть его документы: виджет для отображения на мнемосхеме, собственная мнемосхема и/или окно управления, тренд, отчет.

Добавим в дерево системы OPC-сервер **Modbus Universal MasterOPC** и вставим все переменные и группы (конфигурация у нас уже заготовлена заранее и приложена к данному диску) – 3-1.

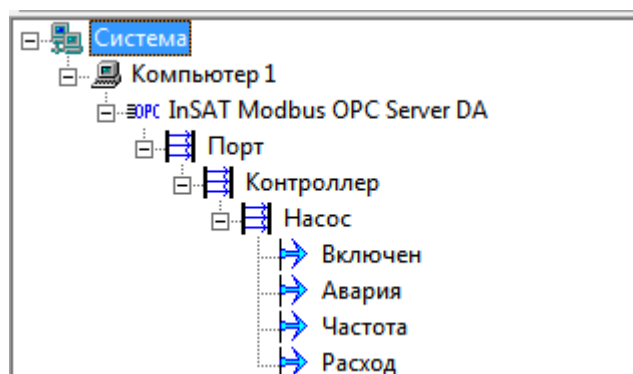


Рисунок 3-1. Дерево системы с OPC-сервером

В дереве объектов, через контекстное меню объекта, создадим объект **Узел** (Рисунок 3-2) – в нем мы будем размещать элементы узла.



Рисунок 3-2. Объект «Узел»

Итак, наш узел состоит из насоса, частотного преобразователя и расходомера. Для насоса у нас есть готовый функциональный блок. Добавим из палитры функциональных блоков раздела **Исполнительные механизмы ФБ Насос мнемонический** (Рисунок 3-3).

Объектно-ориентированный подход

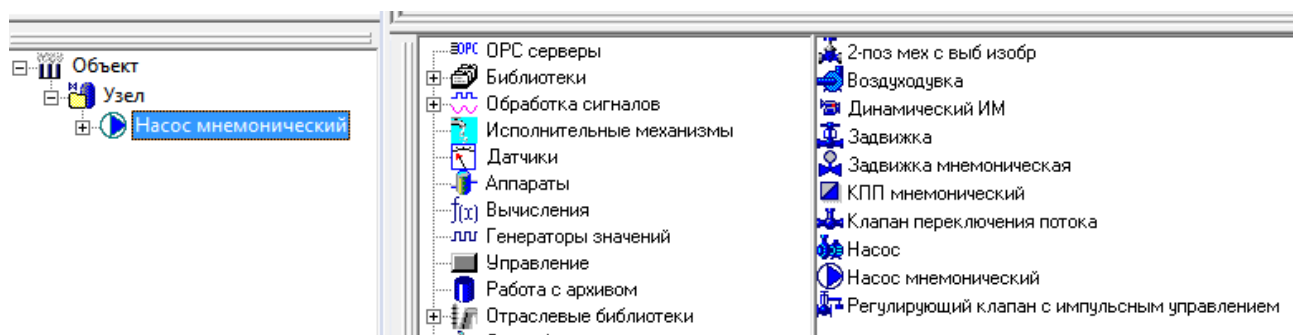


Рисунок 3-3. Добавление функционального блока «Насос»

Можно переименовать данный ФБ – назовем его просто **Насос**.

Настроим данный **Насос**. Мы будем использовать его только для отображения состояния насоса и аварии, поэтому на вкладке **Настройки** выключим **Выходы команд**, а **Входы состояния** установим в режим **Открыт** (Рисунок 3-4).

Примечание. Подробное описание всех настроек данного ФБ можно найти в справке к нему.

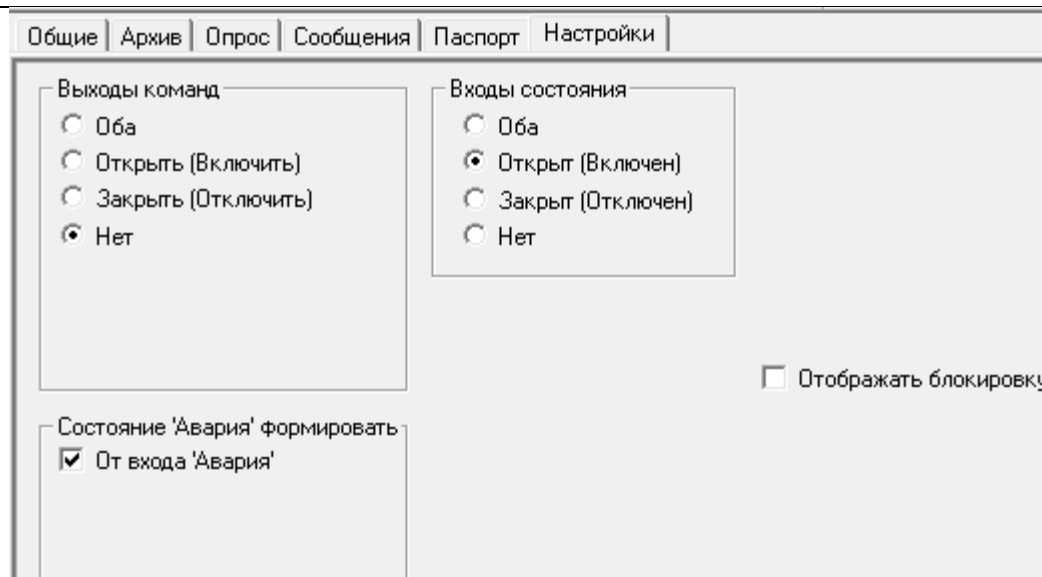


Рисунок 3-4. Настройки ФБ «Насос»

После нажатия кнопки **Применить** у ФБ останется набор входов, как на 3-5.

Объектно-ориентированный подход

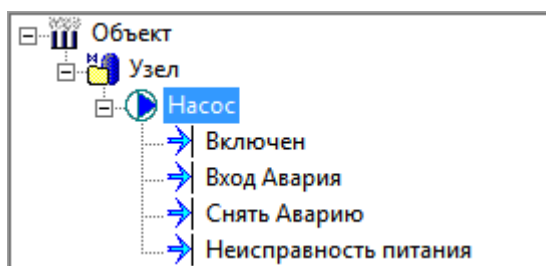


Рисунок 3-5. Входы ФБ «Насос»

Установим связи – перетащим OPC-переменные **Включен** и **Насос** на эквивалентные им входы в функциональном блоке (Рисунок 3-6). Переменные будут иметь красный индикатор – это означает, что переменная имеет связь.

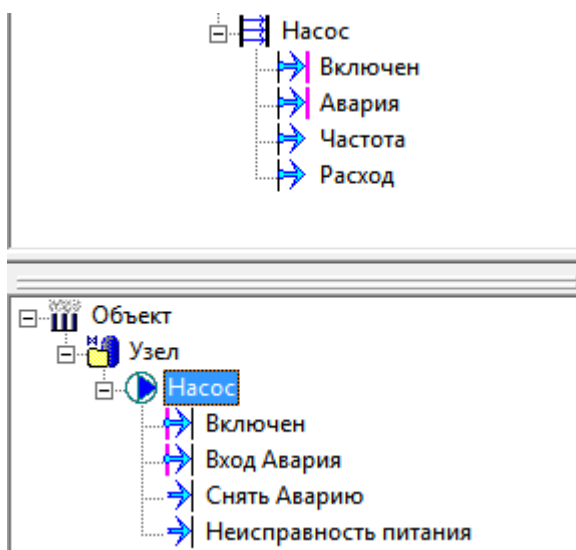


Рисунок 3-6. Установка связей

Специального функционального блока для частотного преобразователя в MasterSCADA нет. Но в данном случае он и не требуется: от преобразователя поступает только сигнал частоты, который нужно отображать на мнемосхеме. Перетащим OPC-переменную **Частота** в объект **Узел** – будет создано значение **Частота** (Рисунок 3-7).

Объектно-ориентированный подход

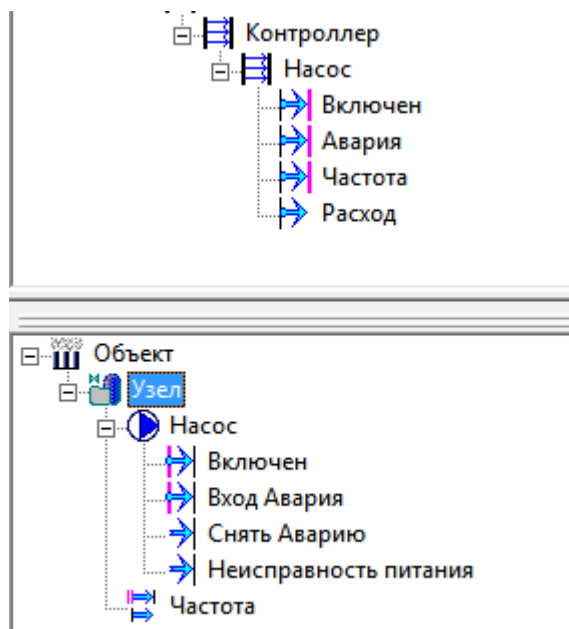


Рисунок 3-7. Создание значения частоты

Аналогично поступим с расходом (Рисунок 3-8) – для расходомера мы разместим на мнемосхеме небольшой рисунок, над которым будет отображаться текущее значение расхода.

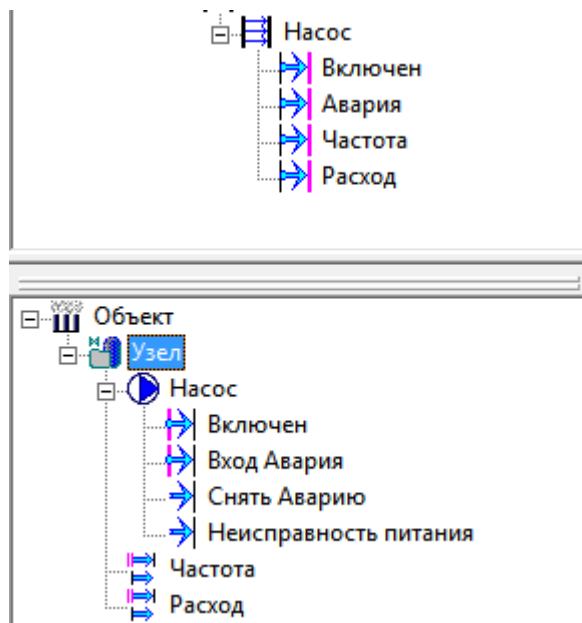


Рисунок 3-8. Создание значения расхода

Мы наполнили объект переменными. В нашем случае в объекте нет внутренних связей (связей между переменными исключительно внутри объекта), однако в большинстве

Объектно-ориентированный подход

объектов они присутствуют. Поэтому очень удобно, когда все внешние связи объекта (как входные, так и выходные) расположены в одном месте.

Для этого у объекта можно включить сервис внешних связей. При активации данного сервиса в объекте добавляются повторители, входы которых – это внешние связи, а выходы – внутренние связи. Выделим объект и на вкладке **Внешние связи** включим данный сервис (Рисунок 3-9).

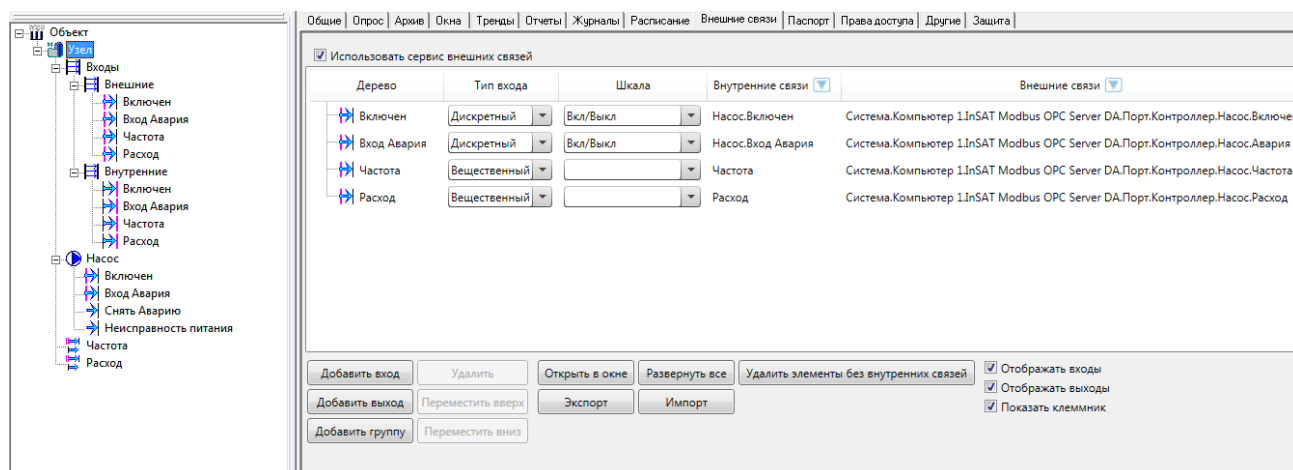


Рисунок 3-9. Сервис «Внешние связи»

Теперь все связи у нас находятся в одном месте, что облегчит работу с ними, особенно если объект будет помещен в библиотеку.

Примечание. Подробное описание данного сервиса можно найти в справке к нему.

Теперь создадим атрибуты объекта – в данном случае различные документы. Создадим у объекта мнемосхему – сделаем это на вкладке **Окна** (Рисунок 3-10).

Объектно-ориентированный подход

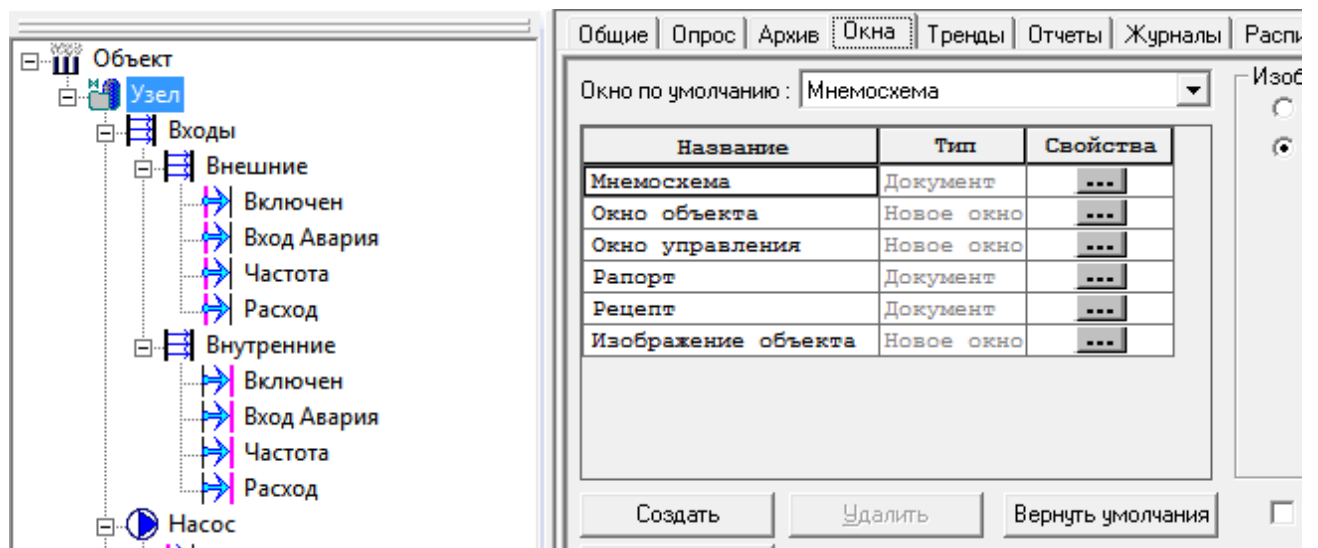


Рисунок 3-10. Создание мнемосхемы

На мнемосхему добавим наш **Насос** – вытащим его левой клавишей мыши (Рисунок 3-11).

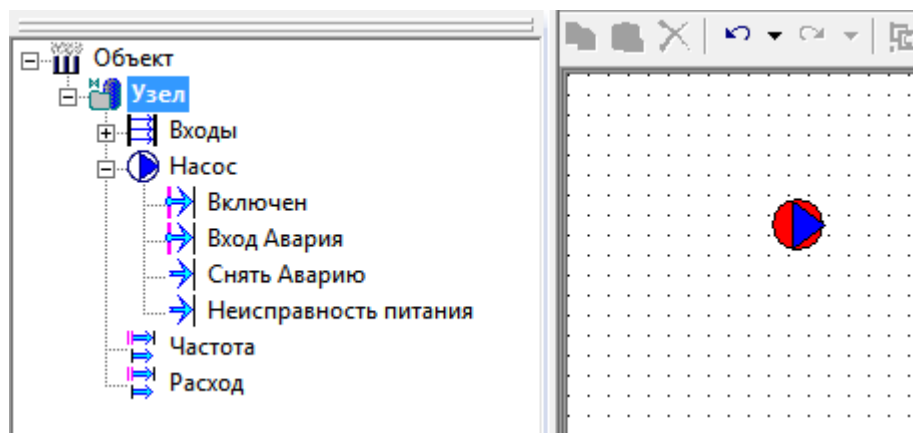


Рисунок 3-11. Вставка контроля насоса на мнемосхему

Аналогично вытащим **Частоту** и **Расход** (Рисунок 3-12).

Объектно-ориентированный подход

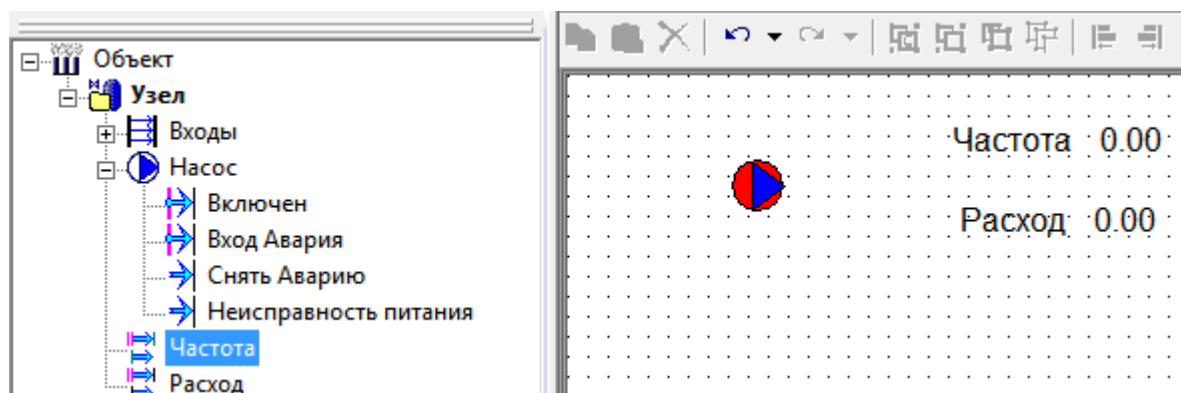


Рисунок 3-12. Создание контролов частоты и расхода

Можно разместить на мнемосхеме и другие элементы: трубы, различную графику и т. д. Ограничимся только созданием небольшого тренда с графиком расхода. Из палитры мнемосхемы, из вкладки **Окна**, добавим элемент **Тренд** и вытащим в него переменную **Расход**. Мнемосхема должна получиться такой, как на 3-13.

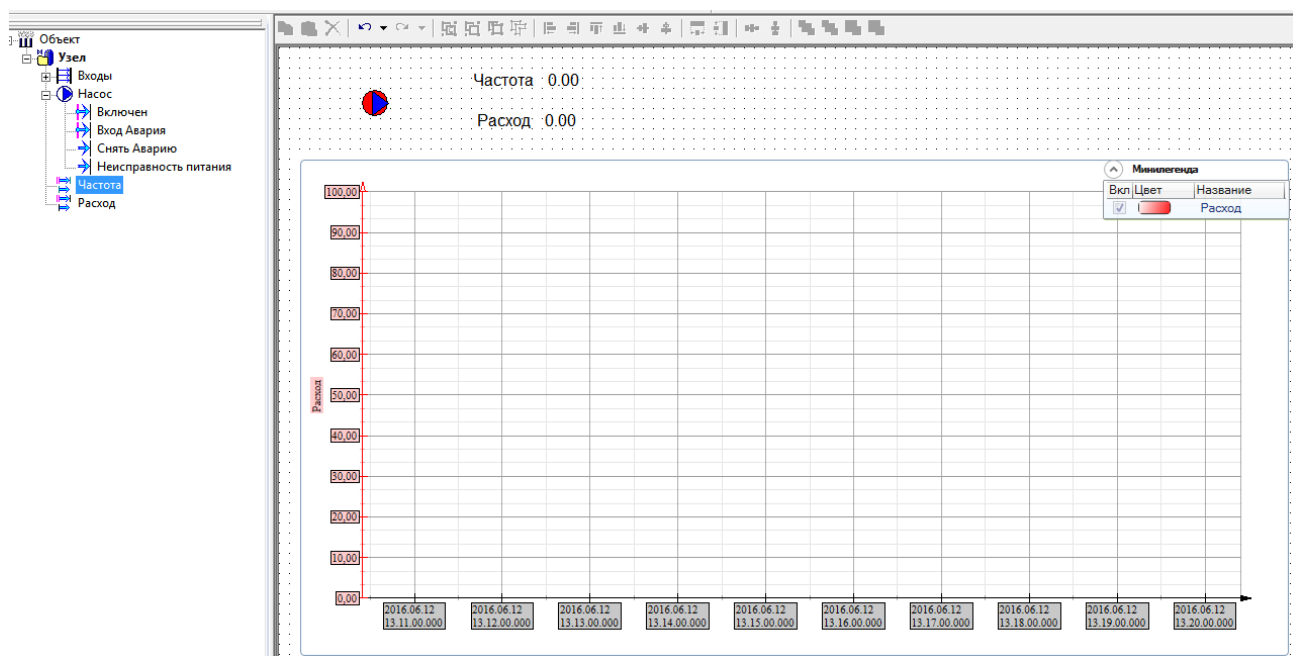


Рисунок 3-13. Готовая мнемосхема

Сохраним мнемосхему и закроем.

Примечание. Подробная работа с редактором мнемосхем будет разобрана в разделе «Редактор мнемосхем MasterGraf».

Однако наш узел нужно встраивать в мнемосхемы – то есть он должен иметь контрол с графикой, информацией об объекте и возможностью открытия окон.

Объектно-ориентированный подход

Для этих целей предназначено окно **Изображение объекта**. Изображение объекта по своей сути является виджетом – активным компонентом, позволяющим выводить различную информацию, а также открывать окна.

Создадим изображение объекта (*Рисунок 3-14*) и также разместим на нем насос, значение с текущей частотой и расходом.

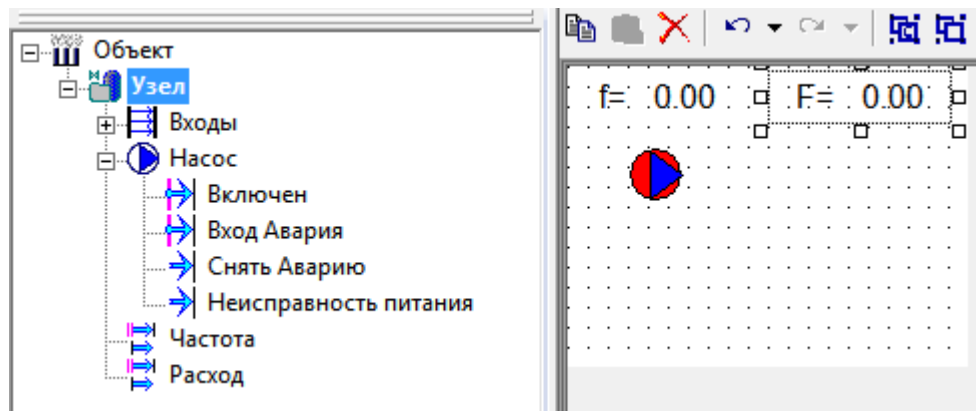


Рисунок 3-14. Создание изображения объекта

Также из палитры мнемосхемы добавим элемент **Рисунок** (раздел **Мультимедиа**), в котором укажем рисунок расходомера и нарисуем трубу (раздел **Объемные элементы**).

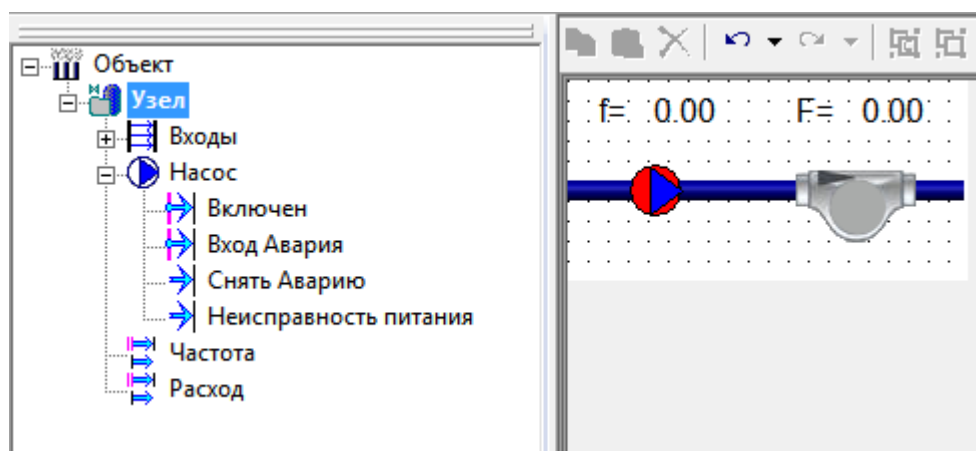


Рисунок 3-15. Создание рисунка и графики

Сохраним изображение. Теперь на другую мнемосхему можно легко добавить данный виджет – нужно просто вытащить на него объект правой кнопкой мыши (*Рисунок 3-16*).

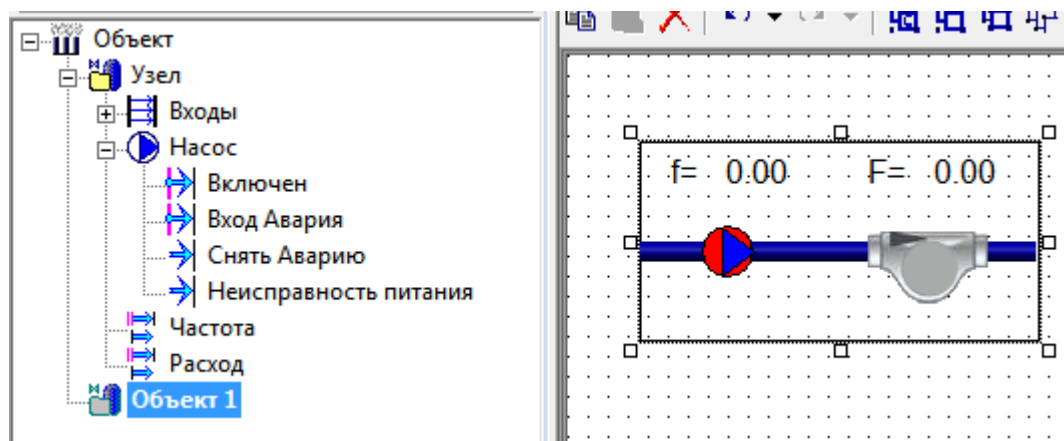


Рисунок 3-16. Вставка изображения объекта в мнемосхему другого объекта

То есть теперь объект – полностью законченный эквивалент реального узла, и его можно встраивать в проекты. Для примера сделаем две копии данного насоса (в дерево системы были добавлены еще OPC-переменные для насосов).

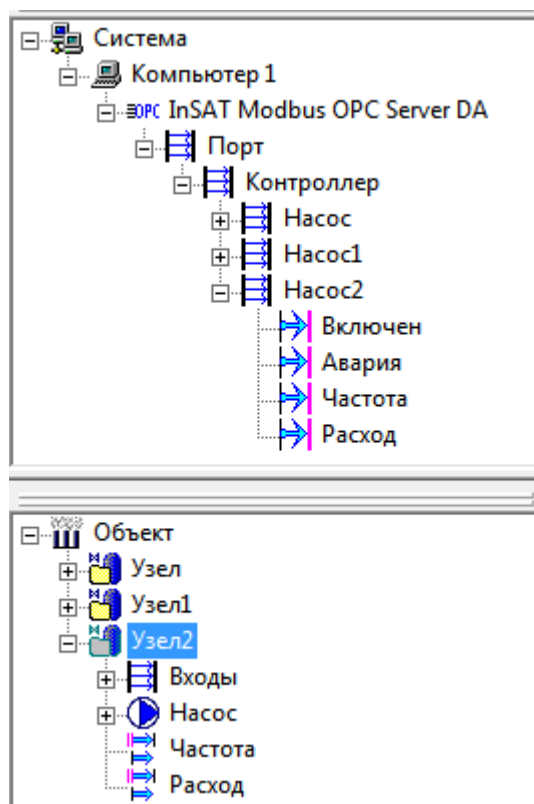


Рисунок 3-17. Дублирование объектов

3.2.Механизм «Шаблон-Экземпляр»

Подход, при котором создаются отдельные элементы, а затем из них комплектуется проект, очень удобен. Но есть в нем и недостаток – обязательно потребуется внести изменения в какой-либо элемент. Изменение может быть вызвано или некорректной работой элемента из-за какой-либо ошибки, или же необходимостью добавить какой-либо новый функционал (например, добавить переменные или сервисные окна).

В MasterSCADA есть решение этой проблемы: для этих целей предназначен механизм **Шаблон-Экземпляр**. Суть данного механизма сводится к следующему: один из объектов назначается шаблоном, после чего начинается его тиражирование – объект копируется и вставляется в нужные части проекта, каждый объект становится экземпляром-шаблоном. Затем, если требуется внести какие-либо правки, они вносятся в шаблон. После чего, используя специальную закладку шаблонного объекта, изменения вносятся в экземпляры. При этом допускается вносить не все изменения, что позволяет обеспечить уникальность каждого экземпляра. Рассмотрим данный механизм на все том же насосе.

Удалим **Узел1** и **Узел2**: мы скопировали, не включив шаблон у **Узла1**, поэтому вносить изменения мы не сможем.

Примечание. На самом деле указать, что «Узел1» и «Узел2» являются экземплярами шаблонами «Узла1», можно и после дублирования – мы покажем это позже.

У объекта **Узел** на вкладке **Общие** поставим галочку **Шаблон**.

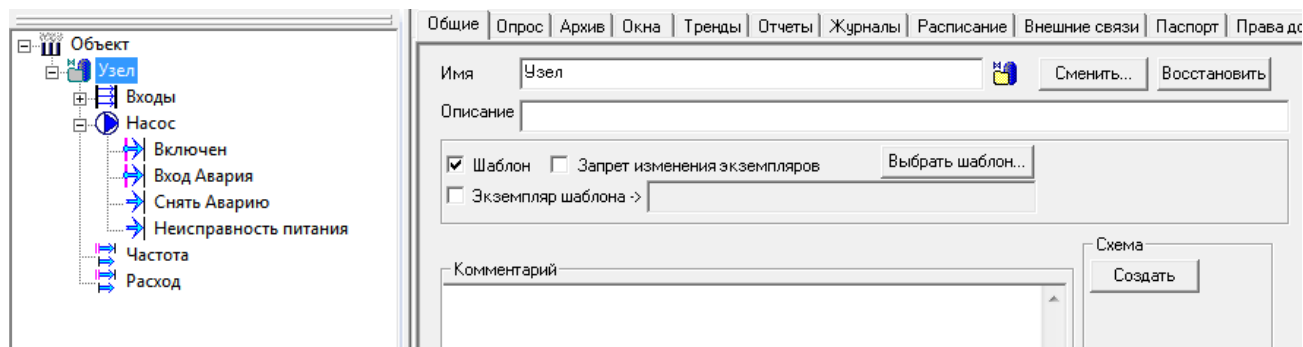


Рисунок 3-18. Включение шаблона

После нажатия кнопки **Применить** значок объекта поменяет свой внешний вид: иконка папки станет зеленой – это означает, что данный объект является шаблоном.

Объектно-ориентированный подход

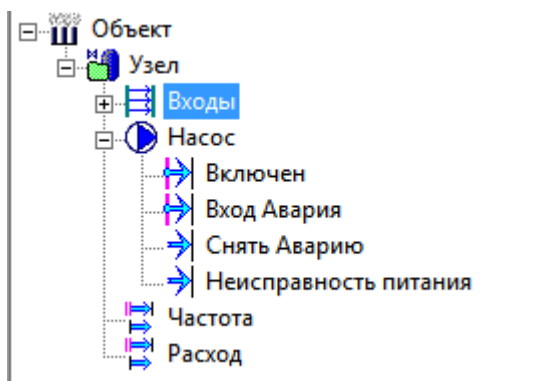


Рисунок 3-19. Индикация того, что объект является шаблоном

Теперь снова сделаем две копии данного объекта и привяжем их к переменным OPC групп **Насос1** и **Насос2**.

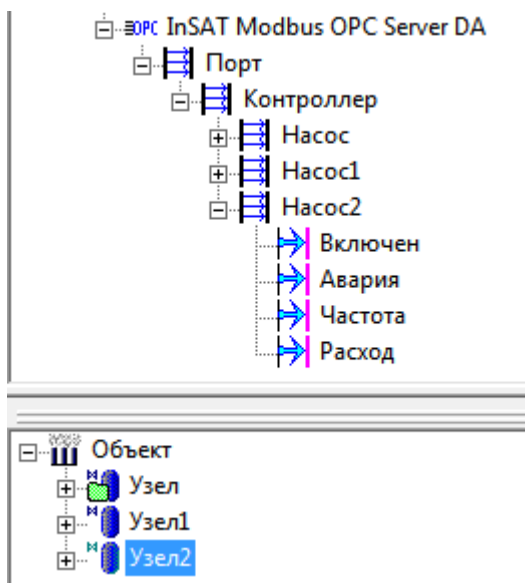


Рисунок 3-20. Экземпляры шаблона

Обратите внимание, что объекты **Узел1** и **Узел2** также поменяли свои значки – они теперь экземпляры шаблона (Рисунок 3-20).

Теперь внесем какое-нибудь изменение в шаблон – в объект **Узел**. Например, добавим на мнемосхему текстовые индикаторы состояния насоса – его состояние работы и аварии. Для этого вытащим на мнемосхему выходы из группы **Внутренние** – **Включен** и **Вход Авария** (Рисунок 3-21).

Объектно-ориентированный подход

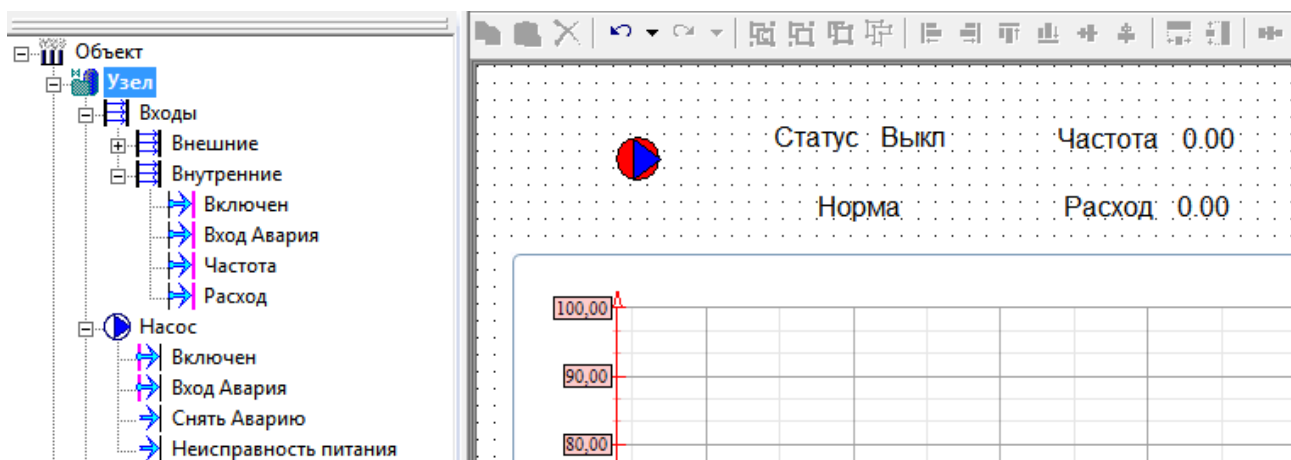


Рисунок 3-21. Внесение изменений в мнемосхему

Сохраним и закроем мнемосхему.

Теперь внесем изменения в экземпляры. Для этого у объекта есть специальная вкладка **Шаблон** – перейдем на нее.

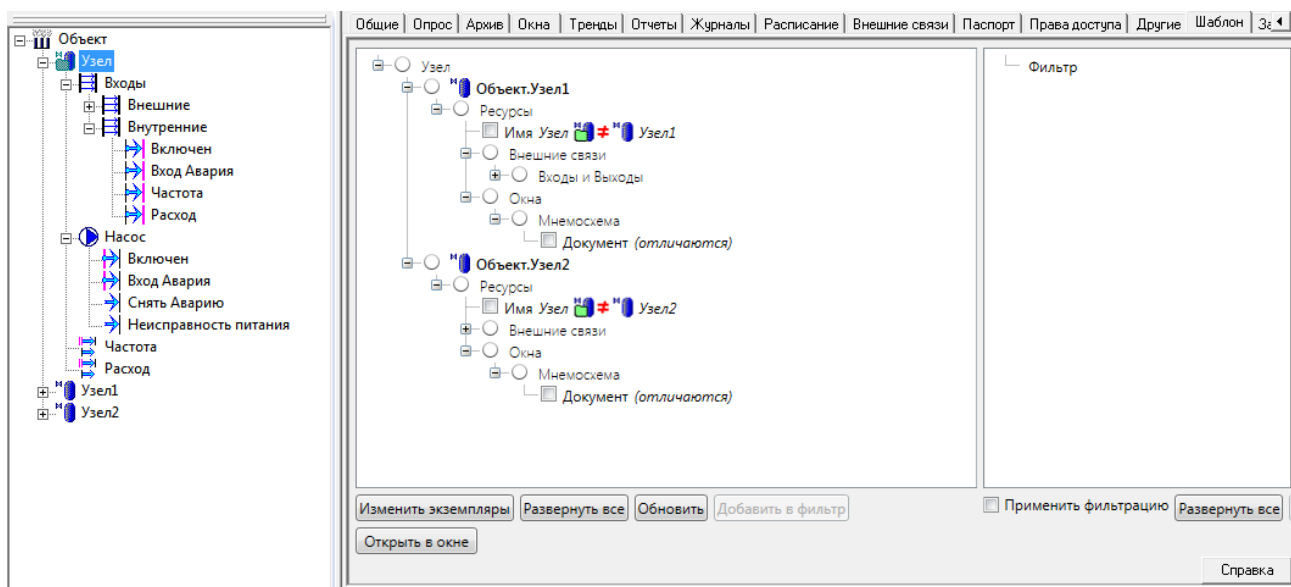


Рисунок 3-22. Дерево отличий шаблона и экземпляров

На данной вкладке будут выведены все отличия экземпляров от шаблона. В данный момент отличий три: различные имена объектов, различные внешние связи (отличаются пути к OPC-переменным) и отличие в документе в мнемосхеме (Рисунок 3-22). Имена и внешние связи нам синхронизировать не нужно, нужно внести правки только в мнемосхему. Отметим галочкой мнемосхему и нажмем кнопку **Изменить экземпляры** (Рисунок 3-23).

Объектно-ориентированный подход

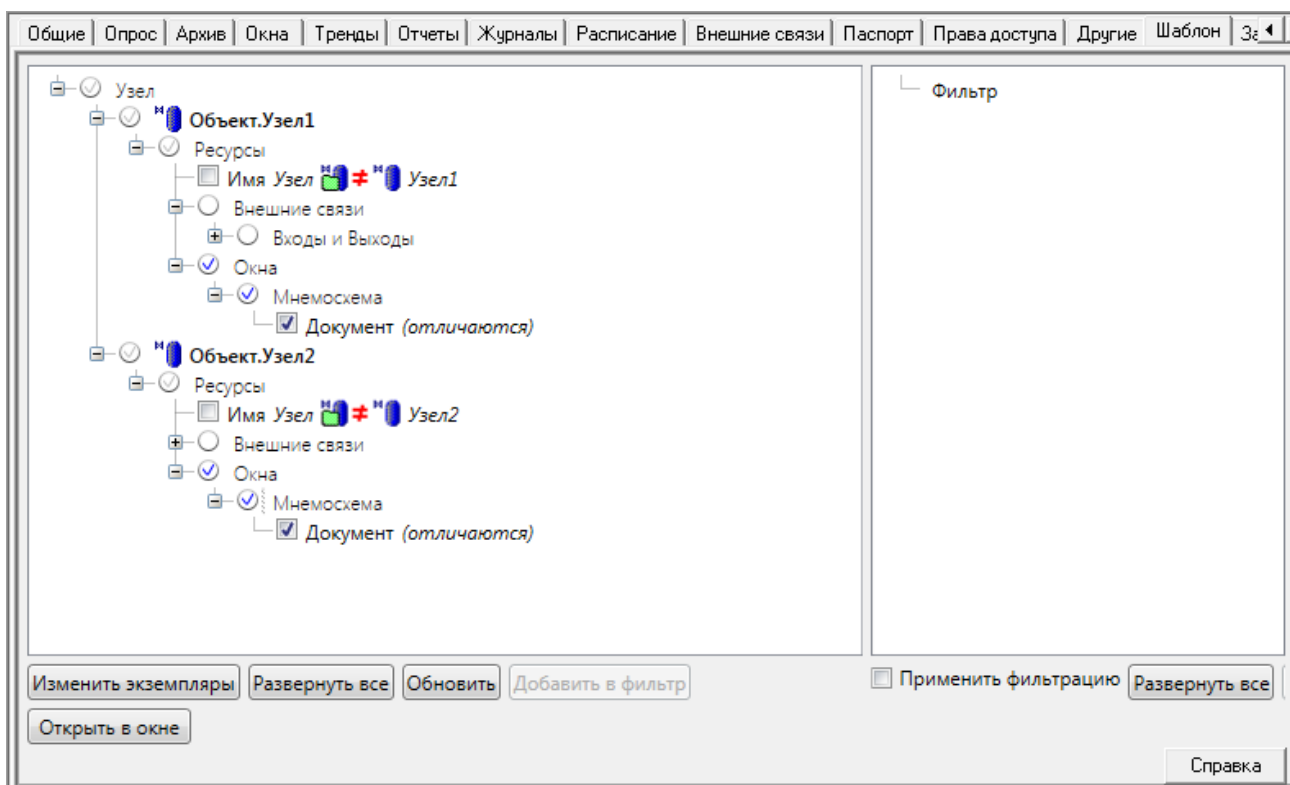


Рисунок 3-23. Внесение изменений в экземпляры

Изменения будут внесены, и отличие в мнемосхеме пропадет из дерева.

Проверим, что изменения действительно внеслись – откроем мнемосхему объекта **Узел1** (Рисунок 3-24). Изменения внесены.

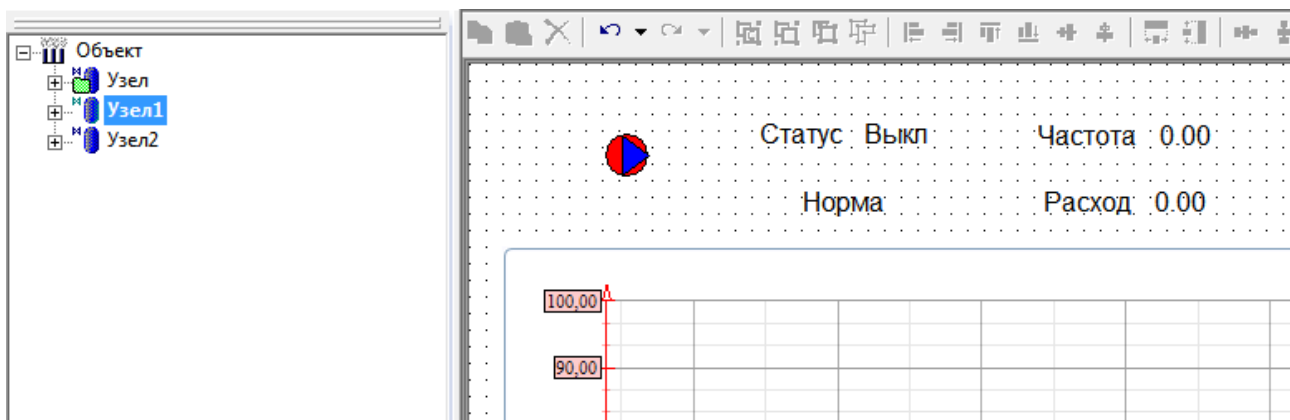


Рисунок 3-24. Проверка изменений

На вкладке **Шаблон** также можно настроить фильтрацию. Например, если вы не планируете синхронизировать имена и внешние связи (это, как правило, не требуется), то просто перетащите эти элементы в дерево фильтра (или нажмите кнопку **Добавить в**

фильтр) и поставьте галочку **Применить фильтрацию**. Элементы будут скрыты из дерева отличий (Рисунок 3-25).

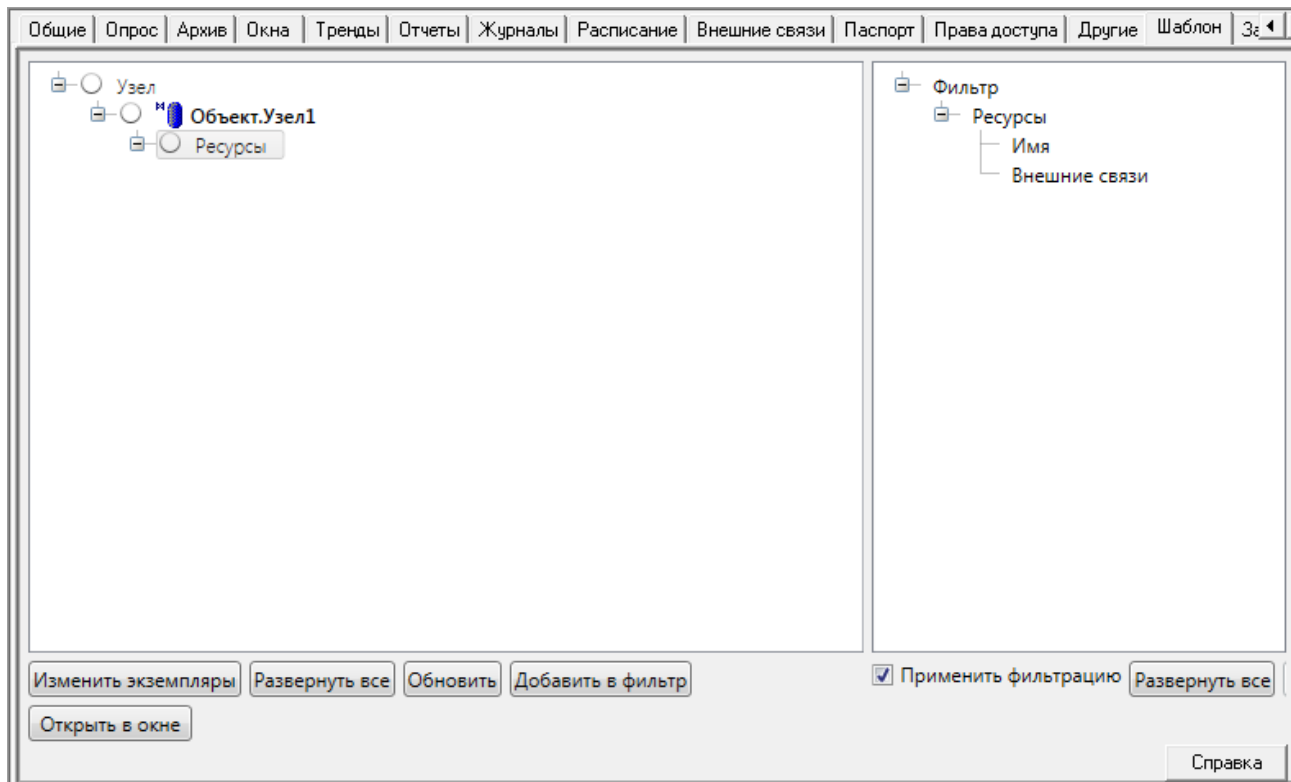


Рисунок 3-25. Фильтр дерева отличий

Как быть, если необходимо включить данный механизм уже после того, как сделано дублирование объектов? Сначала нужно также включить у нужного объекта галочку **Шаблон**. Затем выделить объект, который будет его экземпляром, и перетащить шаблонный объект в поле **Экземпляр шаблона**.

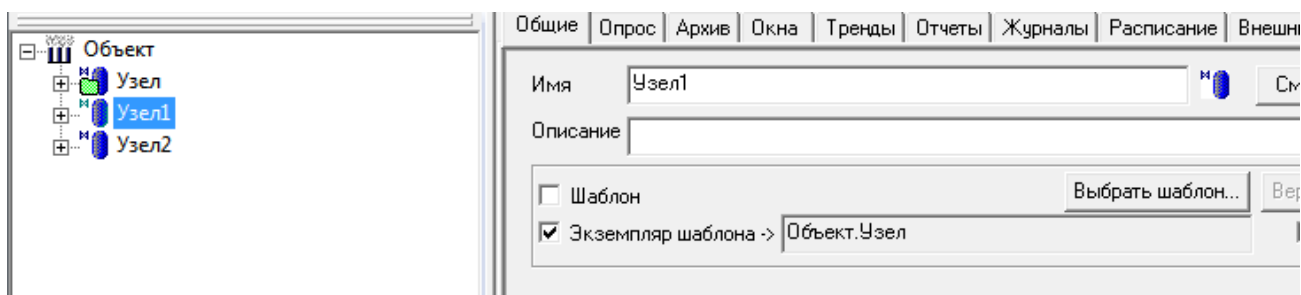


Рисунок 3-26. Привязка объекта к шаблону

Объектно-ориентированный подход

Также можно указать, что объект является экземпляром шаблона, используя кнопку **Выбрать шаблон** (подробнее этот функционал будет описан далее). Допускается вариант, когда один объект является шаблоном для одного объекта и экземпляром для другого.

3.3. Использование библиотек

Итак, вы закончили разработку вашего шаблонного объекта. Если вы планируете использовать его только в текущем проекте, то помещать его в библиотеку не обязательно – просто копируйте его. Если же вы планируете использовать данный объект во многих проектах, то тогда имеет смысл поместить его в библиотеку. При этом механизм **Шаблон-Экземпляр** будет также доступен – добавленный в проект шаблонный объект будет в проекте шаблоном, но при этом экземпляром объекта из библиотеки.

3.3.1. Размещение объекта в библиотеке

Перед тем как помещать объект в библиотеку, рекомендуется сначала создать собственный раздел библиотеки. Это делается через контекстное меню (Рисунок 3-27).

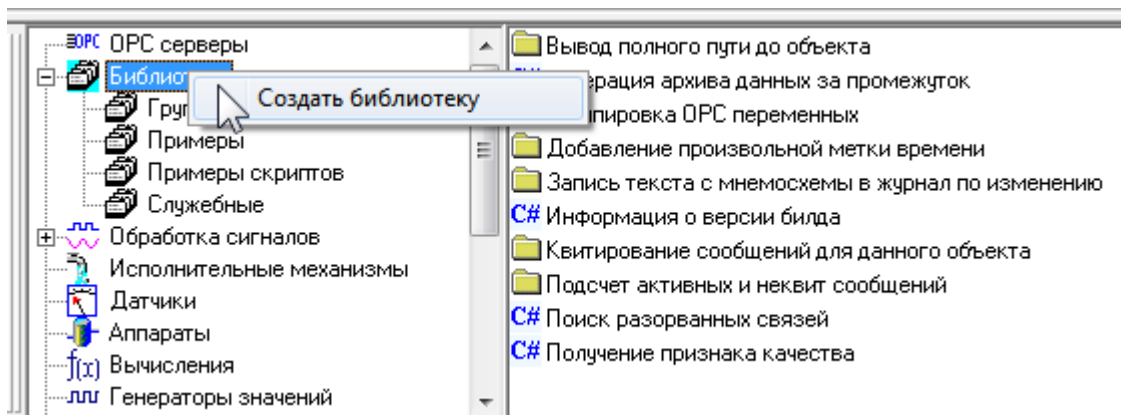


Рисунок 3-27. Создание раздела библиотеки

Создаем библиотеку, которую можно переименовать. Например, назовем ее **Моя библиотека**.

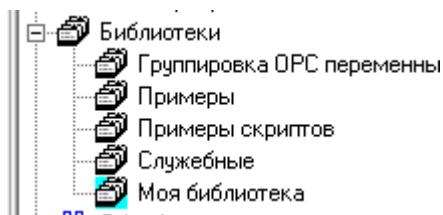


Рисунок 3-28. Переименование библиотеки

Объектно-ориентированный подход

Теперь в нее можно помещать объект. Вызовем у объекта контекстное меню и выберем пункт **Поместить в библиотеку – Моя библиотека**.

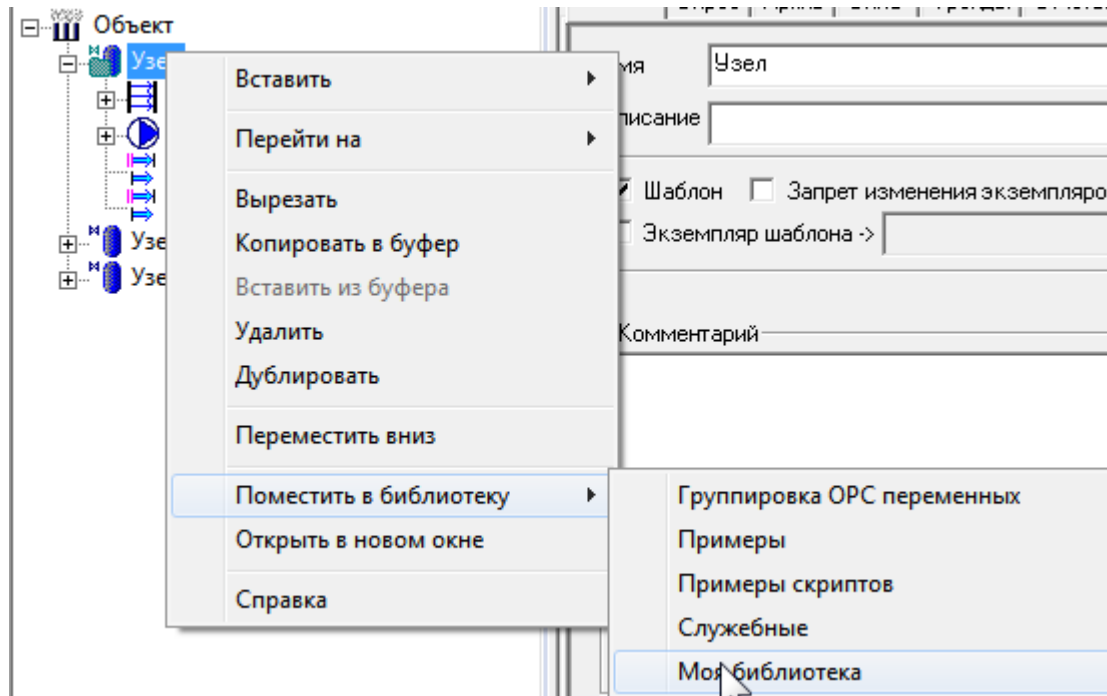


Рисунок 3-29. Вставка объекта в библиотеку

Объект будет размещен в соответствующем разделе библиотеки.

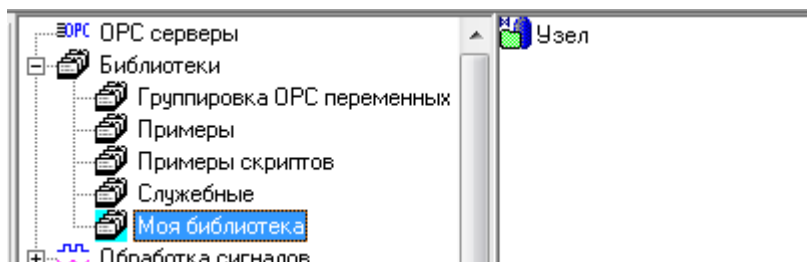


Рисунок 3-30. Объект в библиотеке

3.3.2. Вставка объекта из библиотеки

Теперь данный объект можно добавлять, как обычный функциональный блок из палитры. Его можно добавлять как в текущий проект, так и в другие.

Создадим новый проект и добавим в дерево объектов наш объект из библиотеки – аналогично тому, как мы ранее добавляли ФБ.

Объектно-ориентированный подход

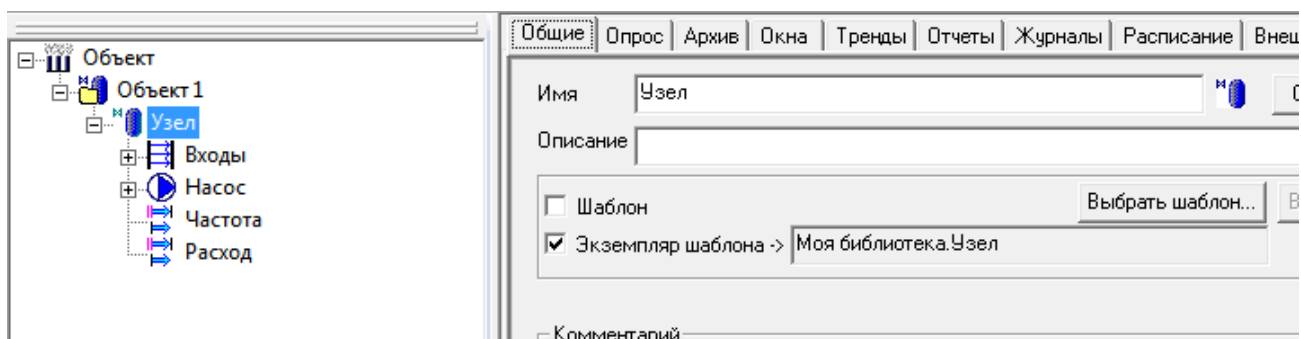
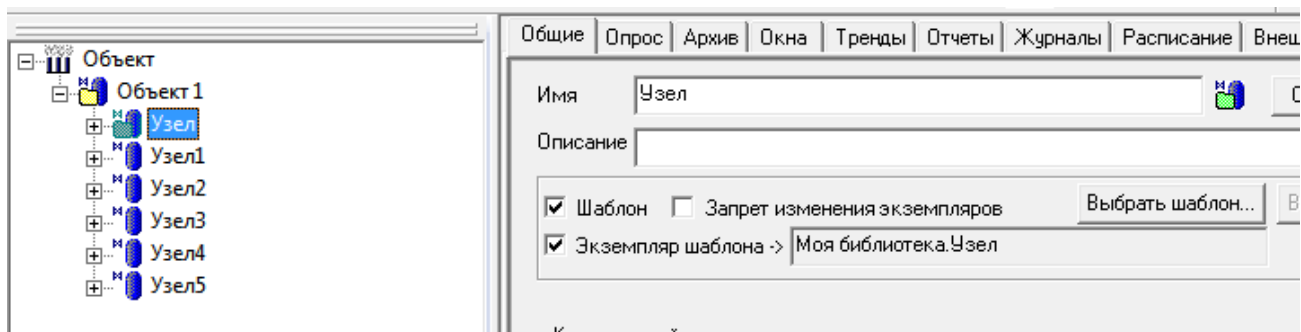


Рисунок 3-31. Вставленный из библиотеки шаблонный объект

Обратите внимание, что данный объект уже является экземпляром шаблона из библиотеки (Рисунок 3-31). Если необходимо тиражировать данный объект в проекте, то нужно установить у него галочку **Шаблон** и только после этого копировать объект.



Теперь объект является и шаблоном, и экземпляром одновременно.

3.3.3. Синхронизация объекта с библиотекой

Теперь рассмотрим ситуацию, при которой нам потребовалось внести правки в объекты в разных проектах, используя библиотеку.

Например, откроем исходный проект (из которого мы помещали объект **Узел** в библиотеку) и внесем какую-нибудь правку. Например, просто добавим переменную типа строковый с именем **Подпись** (Рисунок 3-32).

Объектно-ориентированный подход

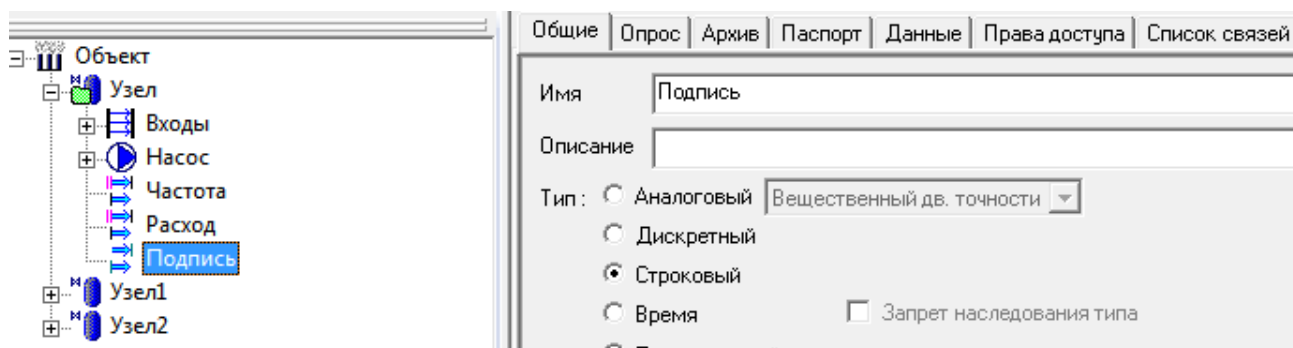


Рисунок 3-32. Создание значения в шаблонном объекте

У данного значения на вкладке **Опрос** можно задать константу и вытащить на мнемосхему. Таким способом удобно добавлять названия узлов на мнемосхему, но мы пока не будем этого делать: у объекта есть отличие, и нам этого достаточно.

Удаляем из библиотеки **Узел**.

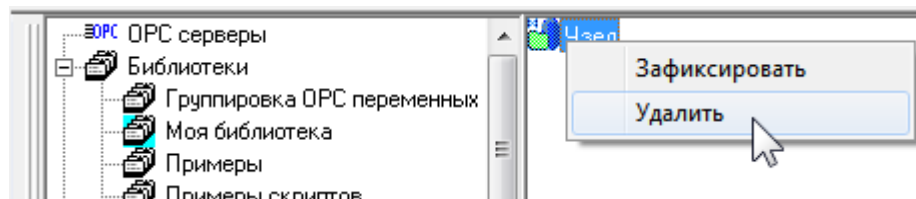


Рисунок 3-33. Удаление объекта

Теперь помещаем в библиотеку наш исправленный объект.

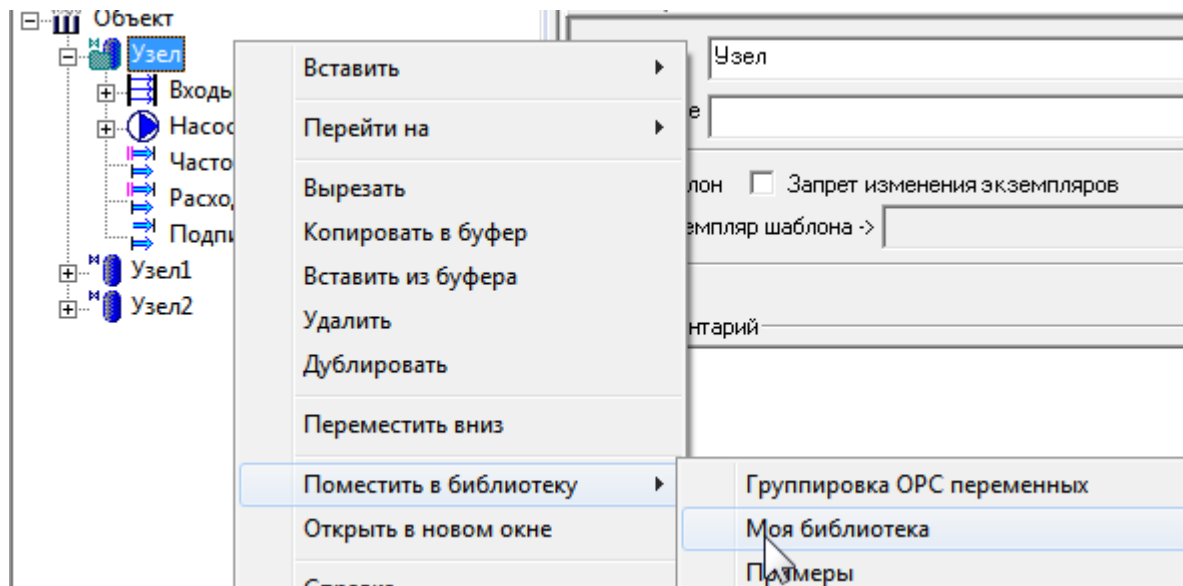


Рисунок 3-34. Размещение объекта в библиотеке

Объектно-ориентированный подход

Открываем второй проект. Выделяем объект – экземпляр библиотечного объекта и переходим на вкладку **Экземпляр**.

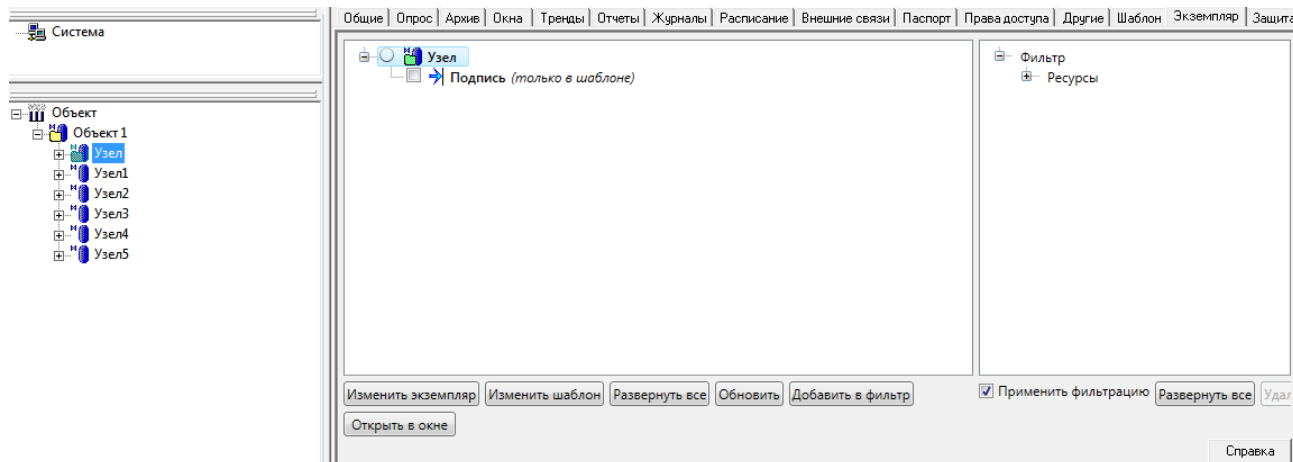


Рисунок 3-35. Отличия между объектом в проекте и библиотечным

На данной вкладке можно увидеть все отличия объектов (Рисунок 3-35). Внесем изменение – отметим добавление значения **Подпись**. Изменения внеслись в объект (Рисунок 3-36).

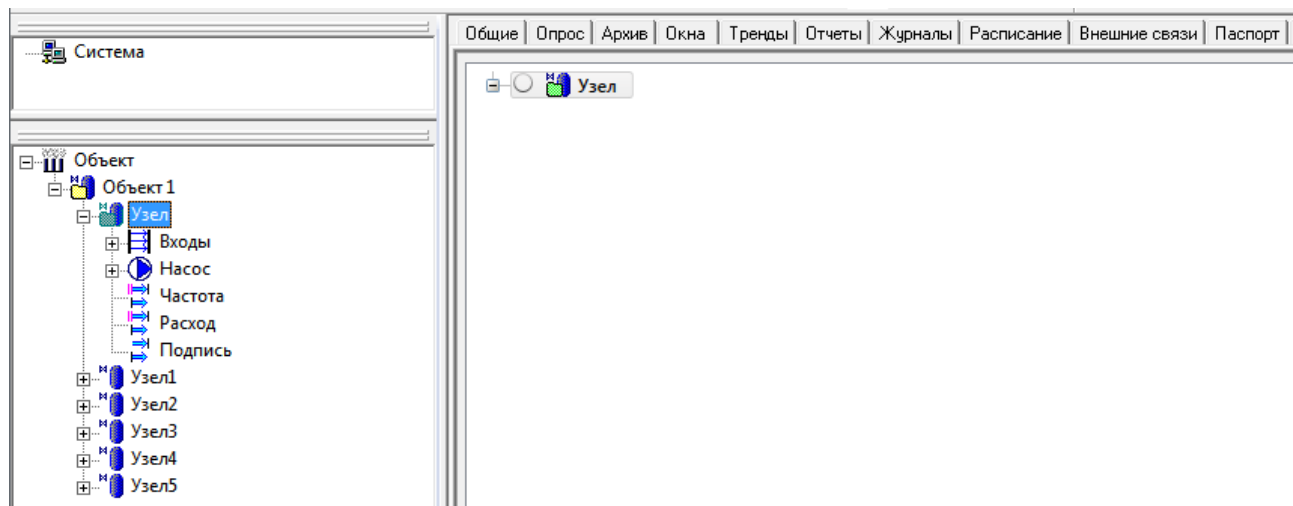


Рисунок 3-36. Объект синхронизирован

Теперь внесем исправления в экземпляры данного шаблона в этом проекте. Перейдем на вкладку **Шаблон** и также отметим внесение значения **Подпись**.

Объектно-ориентированный подход

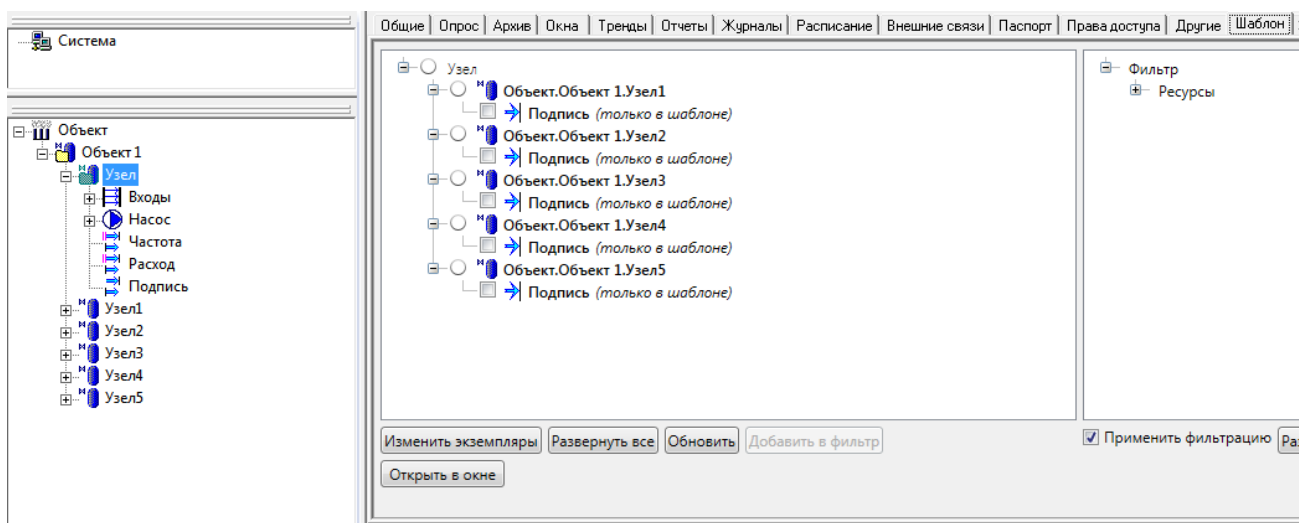


Рисунок 3-37. Отличия у экземпляров

Таким образом можно выполнять синхронизацию одинаковых объектов из разных проектов.

Можно вообще не использовать библиотеку, а использовать синхронизацию объектов из другого проекта. Для этого нужно у объекта на вкладке **Общие** нажать на кнопку **Выбрать шаблон** (Рисунок 3-38).

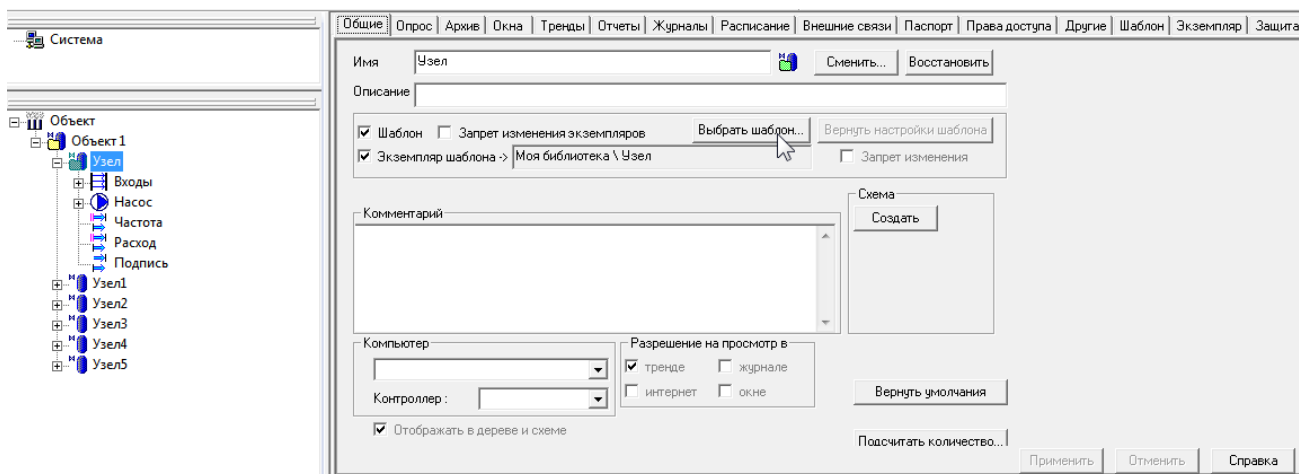


Рисунок 3-38. Выбор шаблона из другого проекта

Откроется окно выбора шаблона. Шаблон можно выбрать из текущего проекта – для этого нужно нажать на кнопку **Обновить** и выбрать нужный шаблон из сформированного дерева (Рисунок 3-39).

Объектно-ориентированный подход

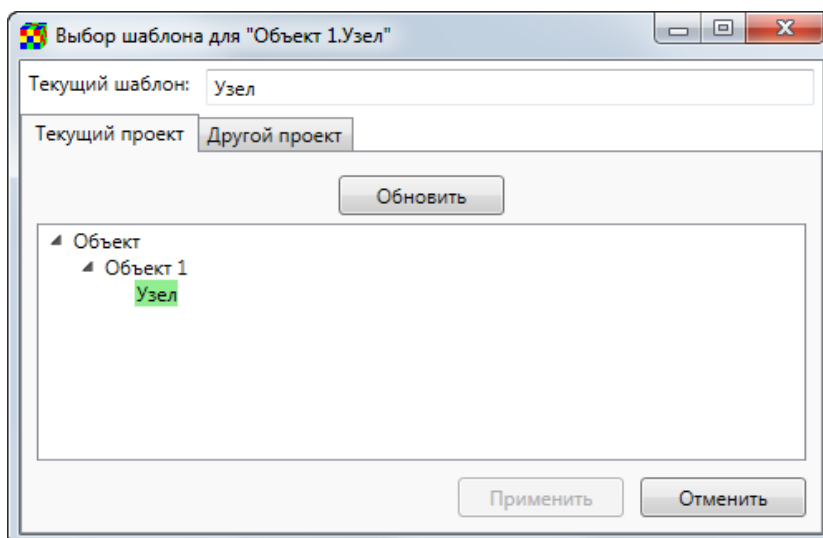


Рисунок 3-39. Дерево шаблонов текущего проекта

Или можно выбрать шаблон из другого проекта. Для этого нужно перейти на вкладку **Другой проект**, выбрать путь до нужного проекта, нажав на специальную кнопку, а затем нажать кнопку **Обновить** – будет также сформировано дерево шаблонов (Рисунок 3-40).

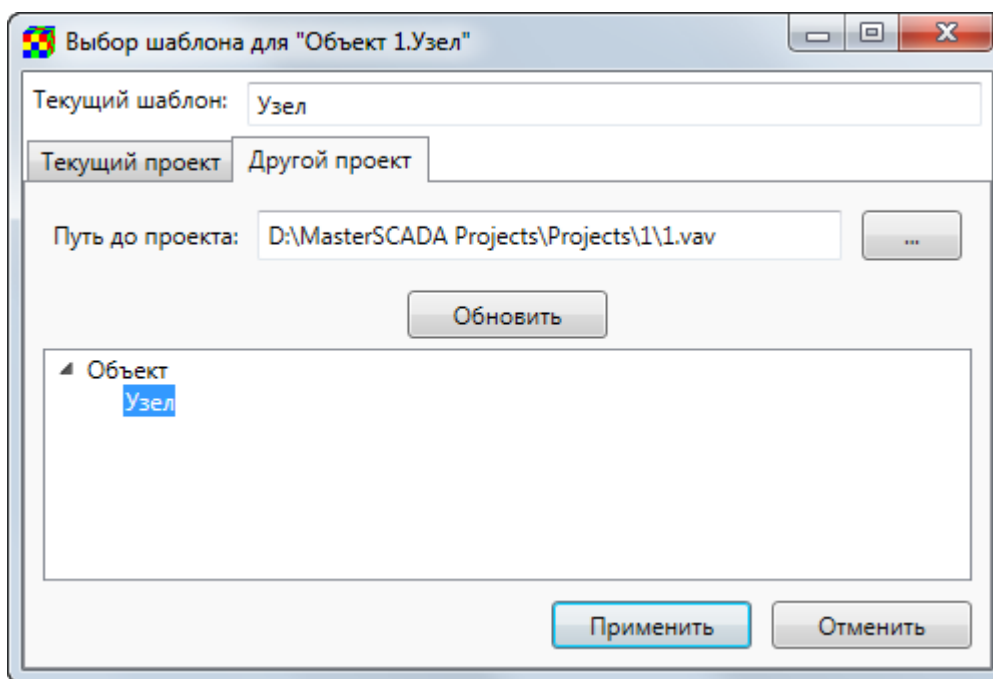


Рисунок 3-40. Дерево шаблонов другого проекта

Теперь можно выбрать нужный шаблон и нажать кнопку **Применить**.

Объектно-ориентированный подход

Таким образом вместо библиотеки можно сделать проект, содержащий все шаблонные объекты. Правки вносятся в данный проект с шаблонами, а затем переносятся в остальные проекты.

3.4. Механизм «Типизация»

Механизм **Типизация** является эволюционным продолжением механизма **Шаблон-Экземпляр**. При использовании механизма **Шаблон-Экземпляр** в дереве присутствуют шаблонный объект и набор его экземпляров (которые могут и отличаться от шаблона), в случае же использования механизма **Типизация** в дереве присутствует только объект шаблона, а его экземпляры не отображаются – они формируются динамически при запуске в режиме исполнения и исполняются в памяти. Поскольку в дереве отсутствует большое количество объектов, проект получается компактным, он быстро загружается и работает.

Механизм **Типизация** рекомендуется использовать в проектах с большим количеством полностью однотипных объектов. При этом объект может быть любым: это может быть как целый узел (например, котел, кондиционер, ИБП), так и составная часть другого узла (например, насосы, клапаны, задвижки). При этом допускается комбинация типизированных объектов с «обычными» объектами.

Рассмотрим создание проекта с использованием типизации. Например, OPC-сервер опрашивает 10 контроллеров, которые производят измерение температуры, давления и расхода. Таким образом, в OPC-сервере содержится 10 групп переменных, каждая из которых содержит по 3 тега.

Первое, что нужно сделать, – создать шаблонный типизированный объект, экземпляры которого потом будут исполняться. Создание такого объекта практически не отличается от создания объектов при обычном подходе. Единственное отличие – не нужно у него устанавливать внешние связи (с OPC-сервером и другими объектами). Поскольку экземпляры объекта явно не присутствуют в дереве, расстановка связей производится другими средствами – используя специальные закладки у переменных или специализированные контроли.

Объектно-ориентированный подход

Создадим объект, назовем его **Общий**, а в него добавим объект **Измеритель** – именно этот объект и станет типизированным. Создадим в нем три значения, каждому назначим собственную шкалу (Рисунок 3-41).

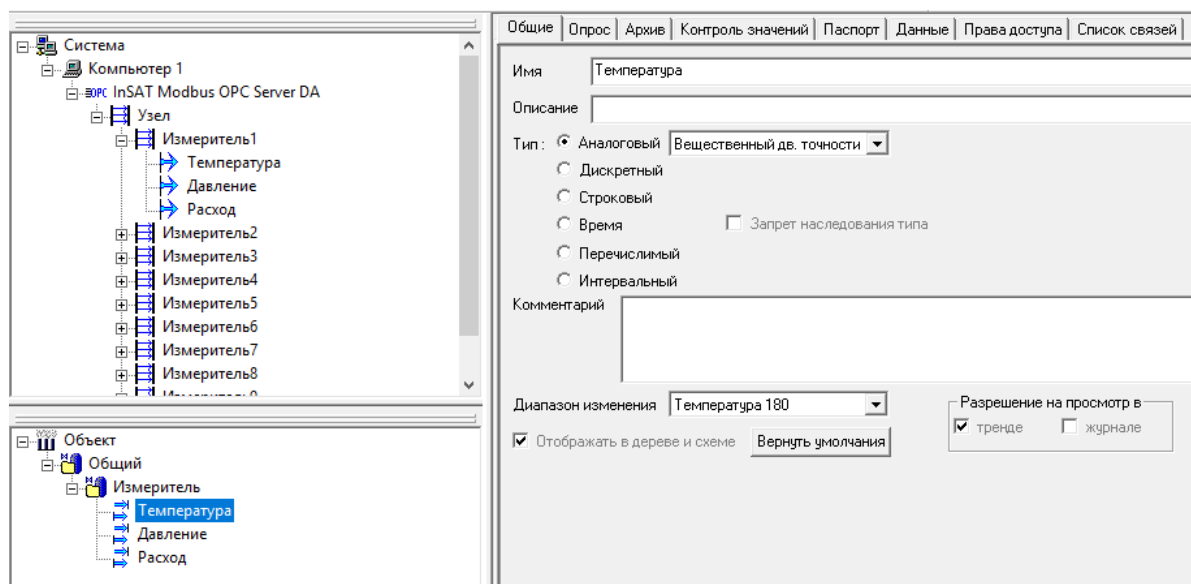


Рисунок 3-41. Создание типизированного объекта

Создадим мнемосхему объекта, разместим в нем три значения, а также тренд, в который также добавим данные значения (Рисунок 3-42). Таким образом на мнемосхеме мы сможем видеть как текущие, так и архивные данные.

Объектно-ориентированный подход

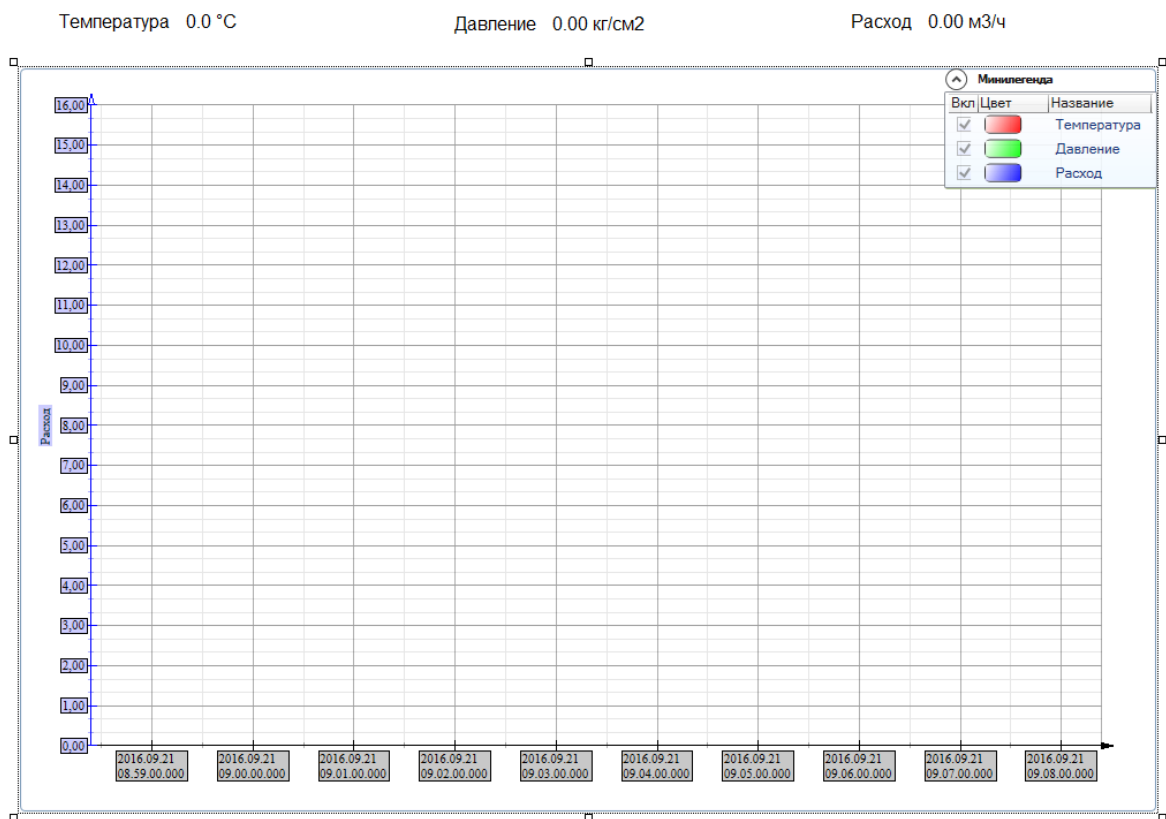


Рисунок 3-42. Мнемосхема типизированного объекта

Создадим у объекта изображение объекта. В него также добавим три наших значения – это позволит на общей мнемосхеме видеть текущие значения параметров (Рисунок 3-43).

Температура 0.0 °C

Давление 0.00 кг/см2

Расход 0.00 м3/ч

Рисунок 3-43. Изображение объекта типизированного объекта

Пока что мы с вами не сделали ничего нового – проект строится, как и обычно (разве что не установили связей). Теперь сделаем наш объект типизированным. Для этого сначала необходимо создать список экземпляров. Выделим **Система** – вкладка **Типизация** (Рисунок 3-44). Именно на этой вкладке и производится создание списков экземпляров.

Объектно-ориентированный подход

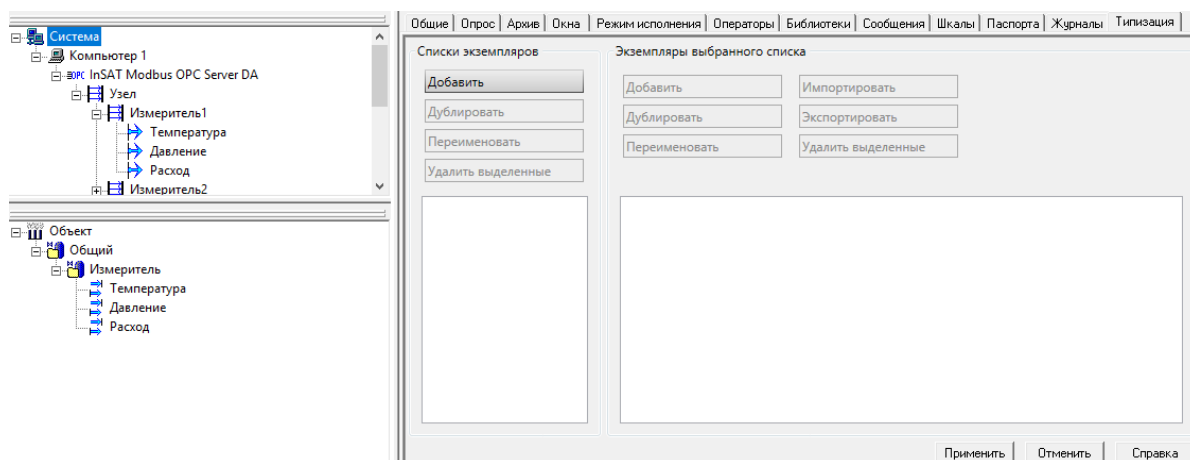


Рисунок 3-44. Вкладка «Типизация»

Создадим новый список, назовем его *Измерители*. Теперь создадим в нем экземпляры. Нажмем на кнопку **Добавить**, введем имя *Измеритель 1* и укажем количество экземпляров – **10**. Экземпляры будут созданы (Рисунок 3-45).

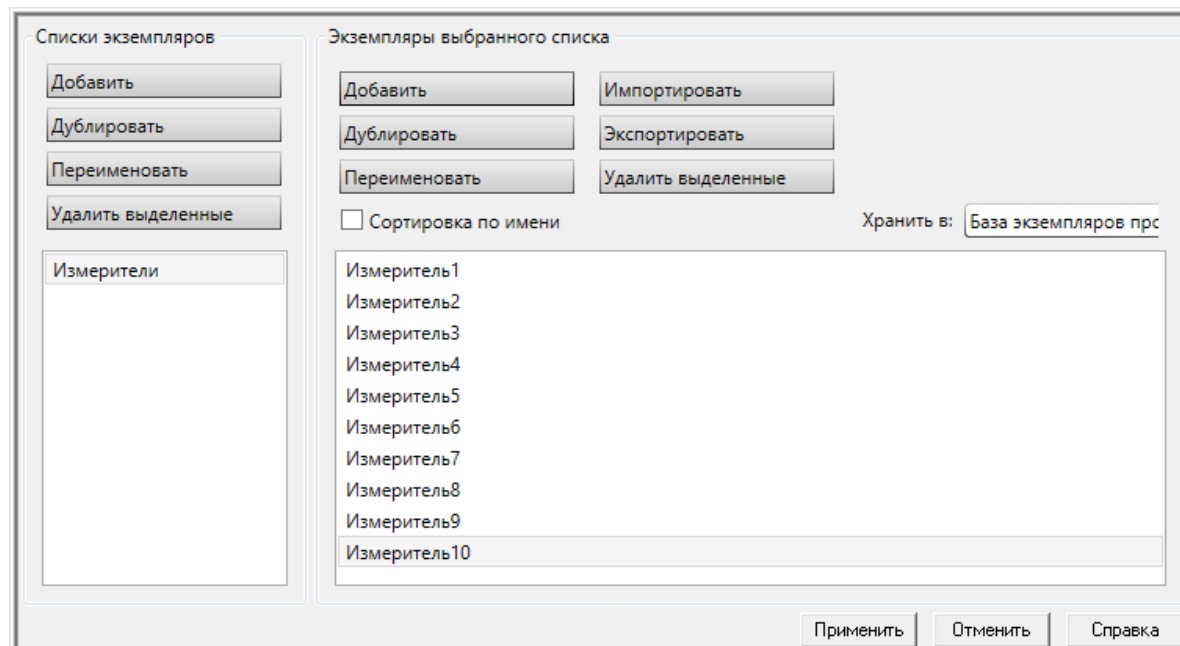


Рисунок 3-45. Создание списка типизированных объектов

Теперь выделим наш объект и перейдем на вкладку *Типизация*. В раскрывающемся списке *Список экземпляров* выберем созданный нами список (Рисунок 3-46).

Объектно-ориентированный подход

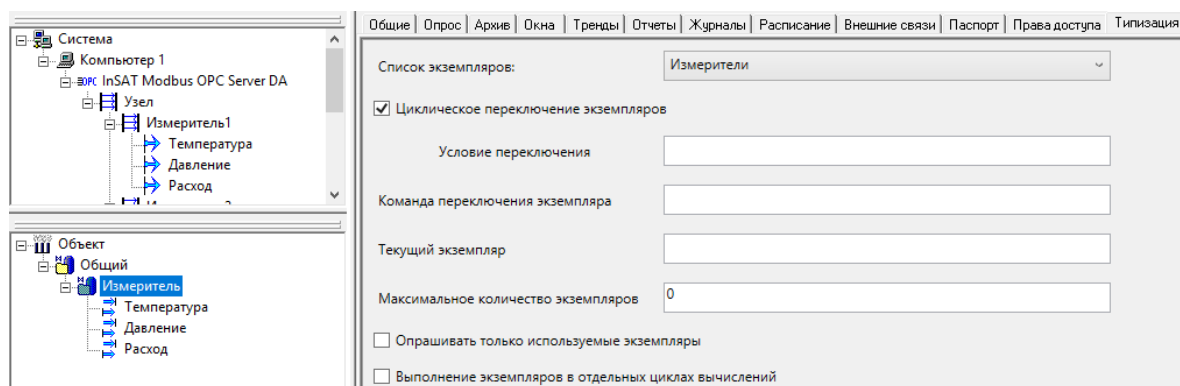


Рисунок 3-46. Назначение списка типизированному объекту

На этой вкладке можно настроить дополнительные настройки опроса – например, задать условие выполнения, указать вручную, какой объект будет опрашиваться, и т. д. Мы будем использовать самый простой опрос – циклический. В этом режиме происходит циклический опрос всех экземпляров объекта.

Теперь следующий обязательный шаг – у объекта необходимо установить собственный цикл опроса. Это делается на вкладке **Опрос** (Рисунок 3-47).

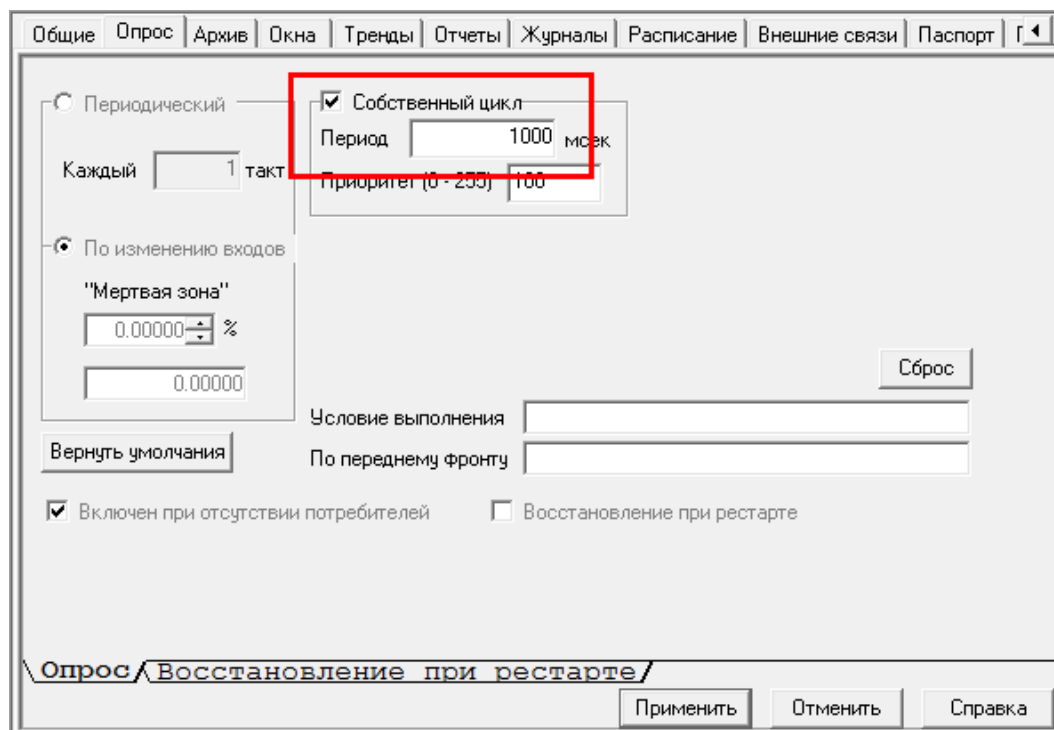


Рисунок 3-47. Включение собственного цикла опроса

Объектно-ориентированный подход

Без установки этого флага опрос типизированного объекта выполняться не будет. Но если вы забудете установить этот флаг, то SCADA предупредит вас об этом сообщением об ошибке при попытке перехода в режим исполнения (*Рисунок 3-48*).

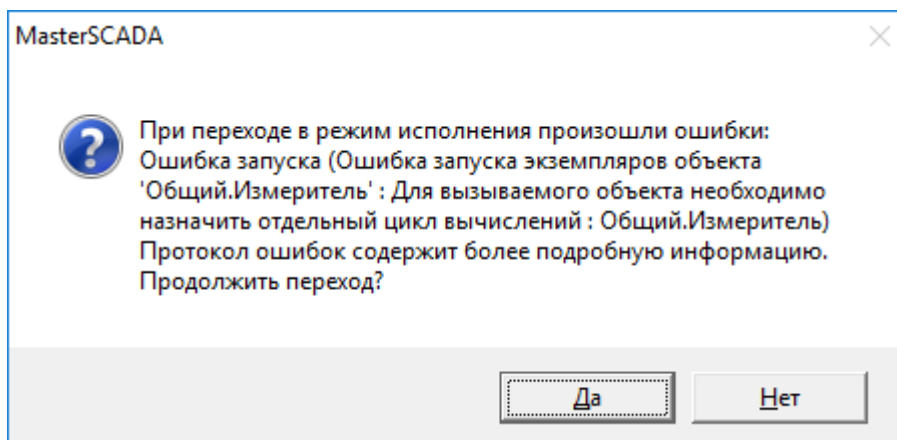


Рисунок 3-48. Предупреждение

Мы привязали объект к списку экземпляров. Как теперь задать связи переменных?

Выделим любую из переменных в типизированном объекте. Теперь у переменной также появилась вкладка **Типизация** – перейдем на нее (*Рисунок 3-49*).

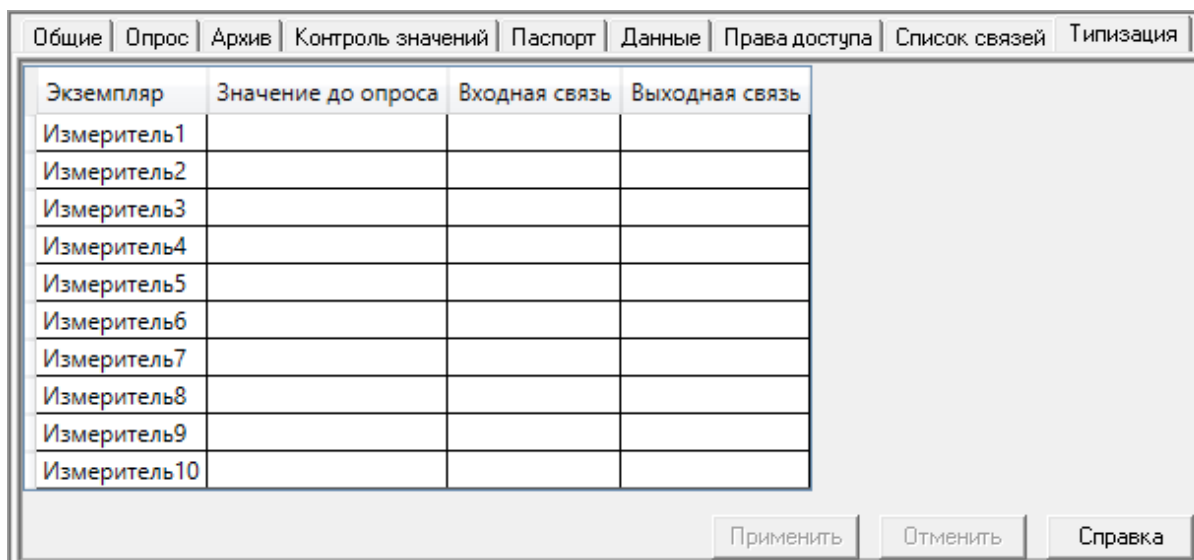


Рисунок 3-49. Вкладка создания связей типизированного объекта

Именно на этой вкладке и осуществляется привязка переменных. В строках таблицы располагаются имена экземпляров, а в столбцах – типы связей. Типов связей может быть три: входная связь; выходная связь; значение до опроса. Перетащим OPC-переменную

Температура первого измерителя с поле **Входная связь** первого экземпляра – **Измерителя 1** (Рисунок 3-50). Связь будет установлена.

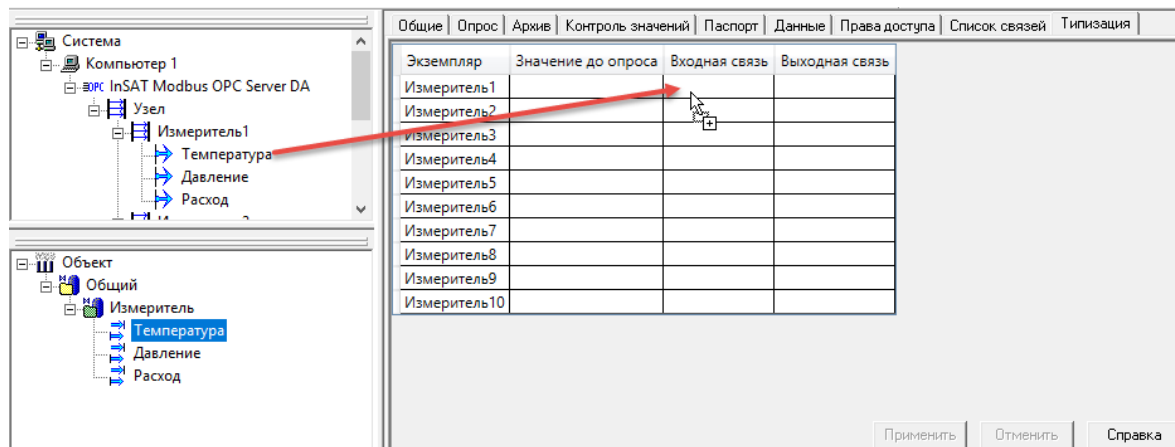


Рисунок 3-50. Создание связи типизированного объекта

Именно таким образом и производится установка связей.

Экземпляров у нас 10, переменных 3, поэтому суммарно связей будет 30. Расставить их вручную можно, но даже при столь небольшом количестве переменных это будет достаточно трудоемко – а как быть, если объектов несколько сотен?

Для этой цели нами был разработан специальный контроль, размещаемый на мнемосхеме. Контроль представляет собой таблицу, в которой представлены все связи типизированного объекта. При этом контроль имеет функции для экспорта и импорта связей в формат CSV, что позволяет произвести формирование связей с помощью сторонних средств, например Excel.

Прежде чем переходить к рассмотрению данного контроля, расставим по одной связи у всех трех переменных – тогда эти связи появятся в контроле и их будет проще править (Рисунок 3-51).

Объектно-ориентированный подход

Экземпляр	Значение до опроса	Входная связь	Выходная связь
Измеритель1		Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел.Измеритель1.Температура	
Измеритель2			
Измеритель3			
Измеритель4			
Измеритель5			
Измеритель6			
Измеритель7			
Измеритель8			
Измеритель9			
Измеритель10			

Применить Отменить Справка

Экземпляр	Значение до опроса	Входная связь	Выходная связь
Измеритель1		Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел.Измеритель1.Расход	
Измеритель2			
Измеритель3			
Измеритель4			
Измеритель5			
Измеритель6			
Измеритель7			
Измеритель8			
Измеритель9			
Измеритель10			

Применить Отменить Справка

Рисунок 3-51. Установленные связи

Контроль располагается на мнемосхеме. Рекомендуется для этой цели создать отдельный сервисный объект, у которого будет выключен опрос. Создадим объект **Сервисный** и у него создадим мнемосхему. В палитре перейдем в раздел **Окна**, выберем элемент **Таблица экземпляров** и разместим его на мнемосхеме (Рисунок 3-52).

Объектно-ориентированный подход

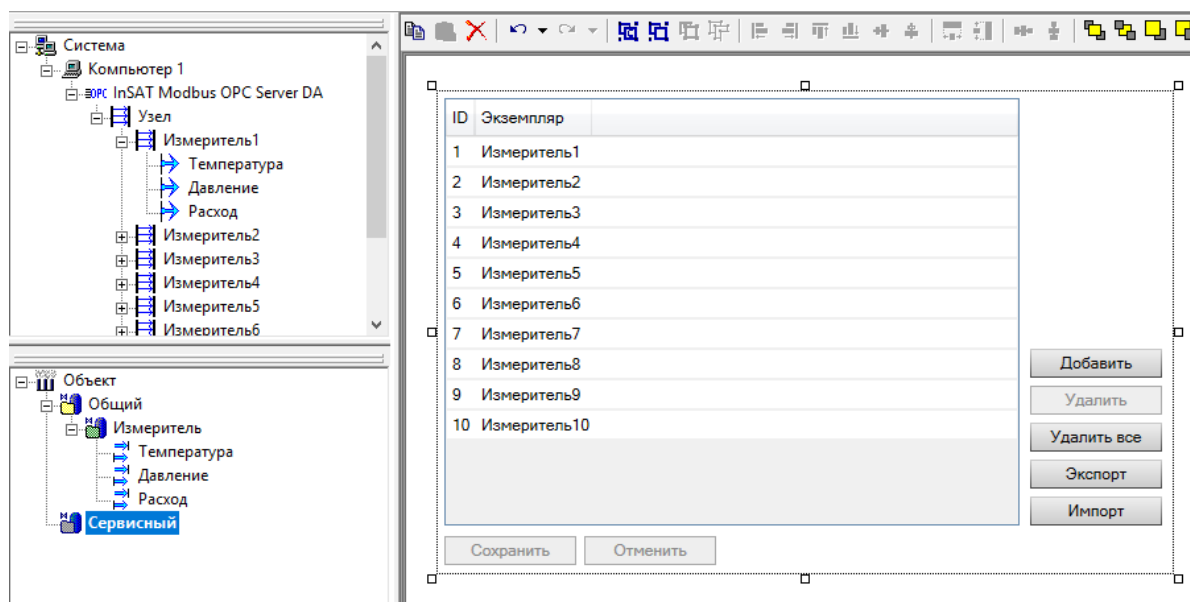


Рисунок 3-52. Контроль «Таблица экземпляров» в сервисном объекте

Поскольку список экземпляров у нас один, таблица автоматически привязалась к нашему списку. Если списков было бы несколько, необходимо было бы указать его вручную на закладке **Свойства** (Рисунок 3-53).

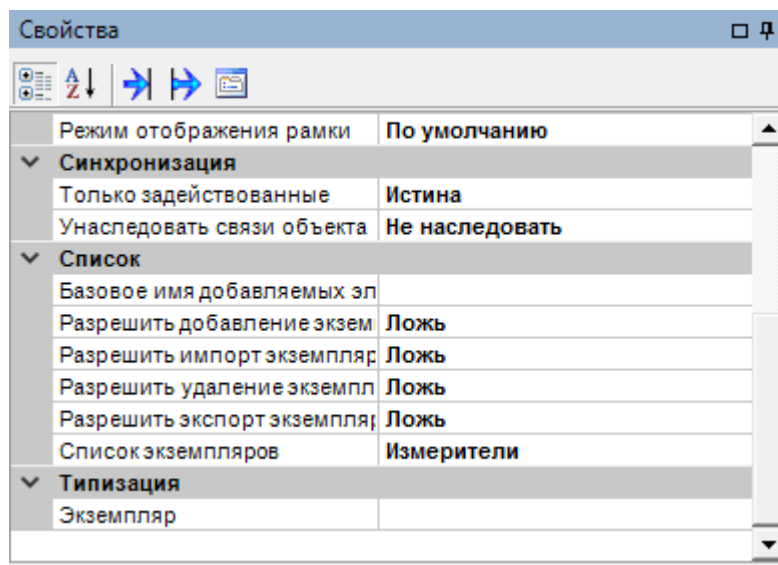


Рисунок 3-53. Свойства контроля «Таблица экземпляров»

По умолчанию синхронизация связей между контролем и списком экземпляров выключена. Поэтому для создания связей нужно перетащить переменную в заголовок таблицы, при этом будет создан столбец со связями – в него уже можно перетаскивать связи. Но, как правило, проще расставить по одной связи в дереве объектов (как мы уже сделали) и

Объектно-ориентированный подход

включить синхронизацию. Для включения синхронизации выделим *контроль*, перейдем на вкладку *Свойства* и включим настройку *Унаследовать связи объекта* (Рисунок 3-54).

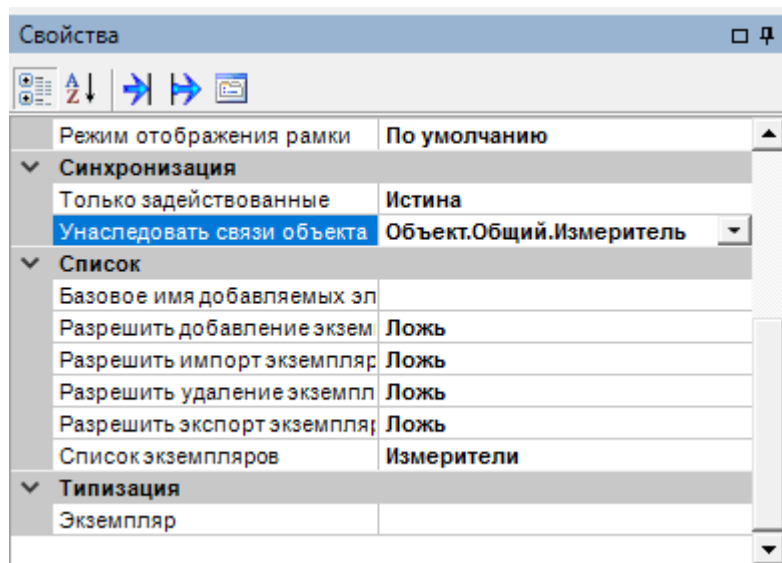


Рисунок 3-54. Включение наследования связей от объекта

В таблице создались все связи, в ней мы также видим, что у первого экземпляра связи установлены (Рисунок 3-55).

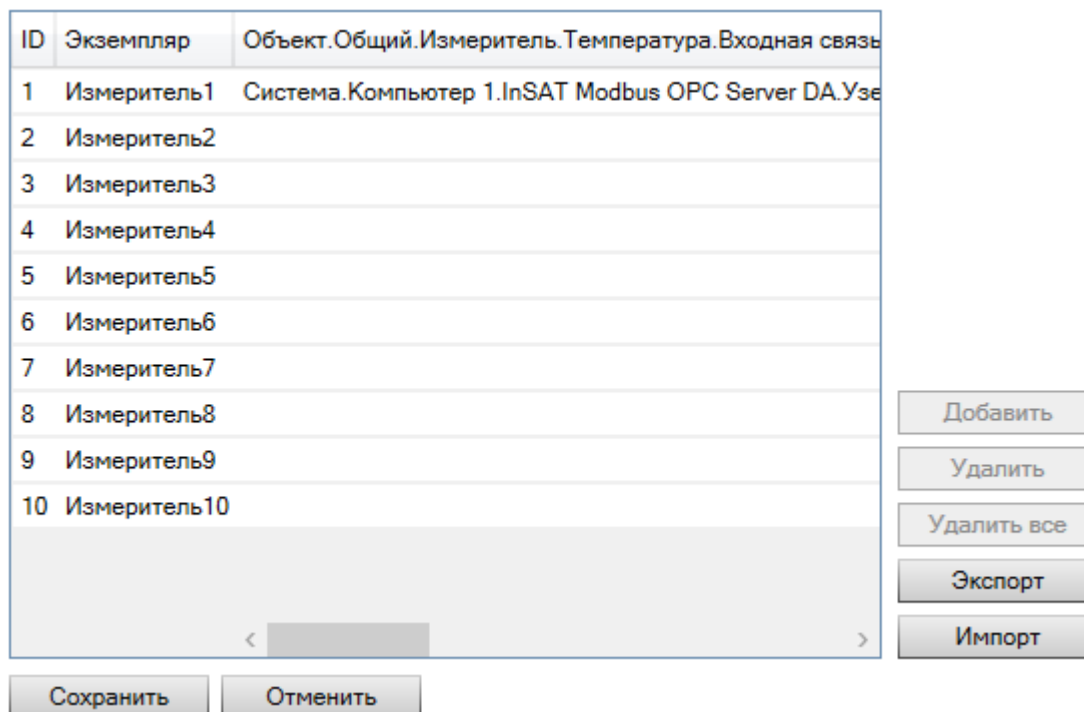


Рисунок 3-55. Контроль «Таблица экземпляров» со связями

Объектно-ориентированный подход

Теперь можно экспортировать связи, произвести правку, а затем выполнить импорт. Нажмем на кнопку **Экспортировать** и сохраним наш файл связей в любую папку.

Рассмотрим пути связей для переменной **Температура** для первого и второго экземпляров:

Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел.Измеритель1.Температура

Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел.Измеритель2.Температура

Как мы видим, путь состоит из статической части (имени OPC-сервера, имени узла и имени тега), а также из динамической (изменяемой) части – имени контроллера. Для первого экземпляра – **Измеритель1**, для второго – **Измеритель2** и т. д.

Таким образом, чтобы сформировать пути для всех экземпляров, необходимо заменить во всех путях всех переменных **Измеритель1** на **ИзмерительN**, соответствующий данному экземпляру. Для решения данной задачи нами разработан специальный макрос для Excel, который позволяет автоматизировать создание и замену связей.

Опустим принцип работы данного макроса – он подробно описан в одной из статей нашего блога, доступного по ссылке <http://blog.insat.ru/search/label/Типизация>. На этой же странице можно скачать и сам макрос.

После того как связи будут сформированы, можно импортировать сформированный файл. Перейдем в MasterSCADA, нажмем в контроле **Импортировать** и выберем наш файл. Пути добавились в ячейки контроля (Рисунок 3-56).

ID	Экземпляр	Измеритель.Температура.Входная связь	Объект.Общий.Измеритель.Давление.Входная связь
1	Измеритель1	р 1.InSAT Modbus OPC Server DA.Узел.Измеритель1.Температура	Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел
2	Измеритель2	р 1.InSAT Modbus OPC Server DA.Узел.Измеритель2.Температура	Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел
3	Измеритель3	р 1.InSAT Modbus OPC Server DA.Узел.Измеритель3.Температура	Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел
4	Измеритель4	р 1.InSAT Modbus OPC Server DA.Узел.Измеритель4.Температура	Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел
5	Измеритель5	р 1.InSAT Modbus OPC Server DA.Узел.Измеритель5.Температура	Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел
6	Измеритель6	р 1.InSAT Modbus OPC Server DA.Узел.Измеритель6.Температура	Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел
7	Измеритель7	р 1.InSAT Modbus OPC Server DA.Узел.Измеритель7.Температура	Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел
8	Измеритель8	р 1.InSAT Modbus OPC Server DA.Узел.Измеритель8.Температура	Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел
9	Измеритель9	р 1.InSAT Modbus OPC Server DA.Узел.Измеритель9.Температура	Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел
10	Измеритель10	р 1.InSAT Modbus OPC Server DA.Узел.Измеритель10.Температура	Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел

Рисунок 3-56. Таблица с импортированными связями

Проверим, что связи установились – выделим любую переменную в дереве (Рисунок 3-57).

Объектно-ориентированный подход

Экземпляр	Значение до опроса	Входная связь	Выходная связь
Измеритель1		Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел.Измеритель1.Температура	
Измеритель2		Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел.Измеритель2.Температура	
Измеритель3		Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел.Измеритель3.Температура	
Измеритель4		Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел.Измеритель4.Температура	
Измеритель5		Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел.Измеритель5.Температура	
Измеритель6		Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел.Измеритель6.Температура	
Измеритель7		Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел.Измеритель7.Температура	
Измеритель8		Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел.Измеритель8.Температура	
Измеритель9		Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел.Измеритель9.Температура	
Измеритель10		Система.Компьютер 1.InSAT Modbus OPC Server DA.Узел.Измеритель10.Температура	

Рисунок 3-57. Переменная с установленными связями

Связи установлены.

Теперь рассмотрим использование типизированных объектов в проекте. Например, разместим на мнемосхеме изображения объектов для каждого из экземпляров. Для этого создадим у объекта первого уровня мнемосхему, вытащим на нее объект правой кнопкой мыши и выберем **Изображение объекта**. **Изображение объекта** добавилось на мнемосхему (Рисунок 3-58).

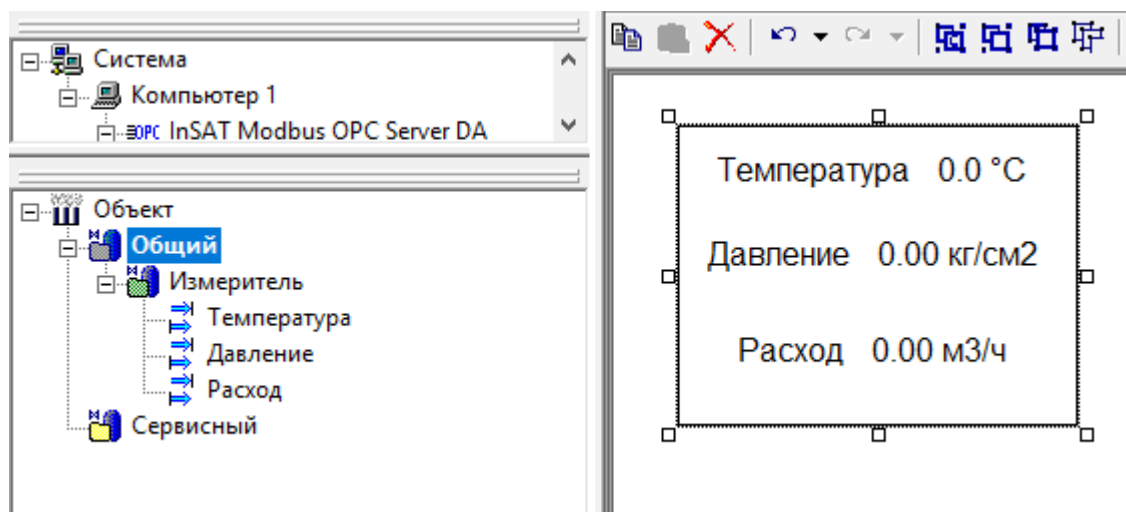


Рисунок 3-58. Создание изображения объекта на мнемосхеме

В свойствах изображения объекта укажем, к какому именно экземпляру оно принадлежит: **Измеритель 1** (Рисунок 3-59).

Объектно-ориентированный подход

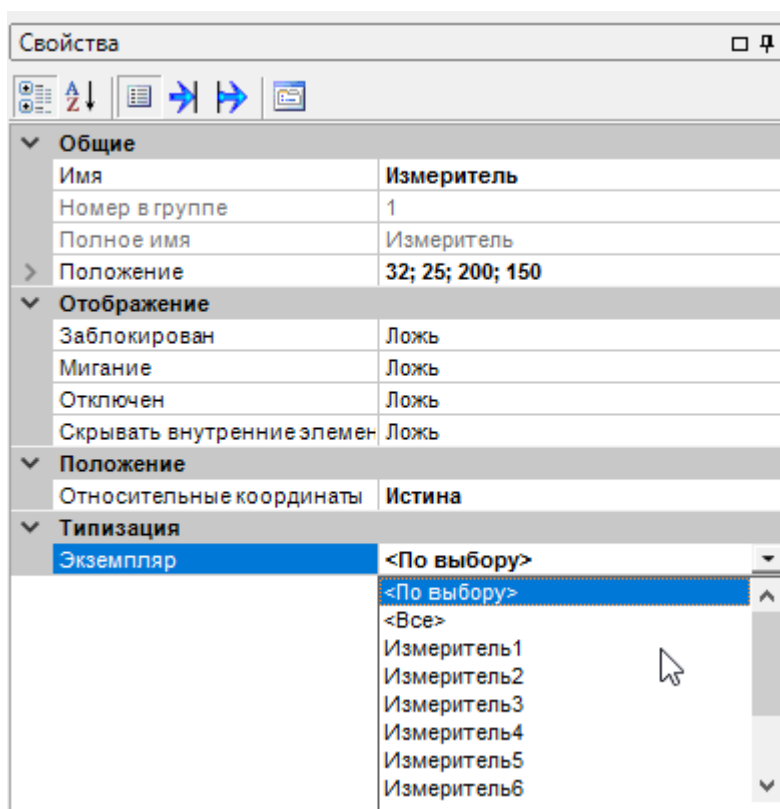


Рисунок 3-59. Привязка изображения к экземпляру

Аналогично разместим изображения для остальных измерителей (Рисунок 3-60).



Рисунок 3-60. Мнемосхема с созданными изображениями

Если экземпляров у вас много и нет необходимости для каждого создавать собственную кнопку (изображение объекта), то можно сделать общую кнопку – оставить в настройках экземпляра режим **По выбору**. В этом случае в режиме исполнения при щелчке по кнопке появится раскрывающийся список, в котором можно будет выбрать, окно какого экземпляра вы хотите открыть.

Мнемосхема готова. Проверим работу нашего проекта в режиме исполнения – запустим его и перейдем на мнемосхему объекта (Рисунок 3-61).

Объектно-ориентированный подход



Рисунок 3-61. Мнемосхема в режиме исполнения

Как мы видим, на мнемосхеме отображаются текущие значения переменных температуры, давления и расхода. Откроем мнемосхему любого измерителя (Рисунок 3-62). В ней мы видим отображение текущих параметров, а на тренде – архивных параметров.

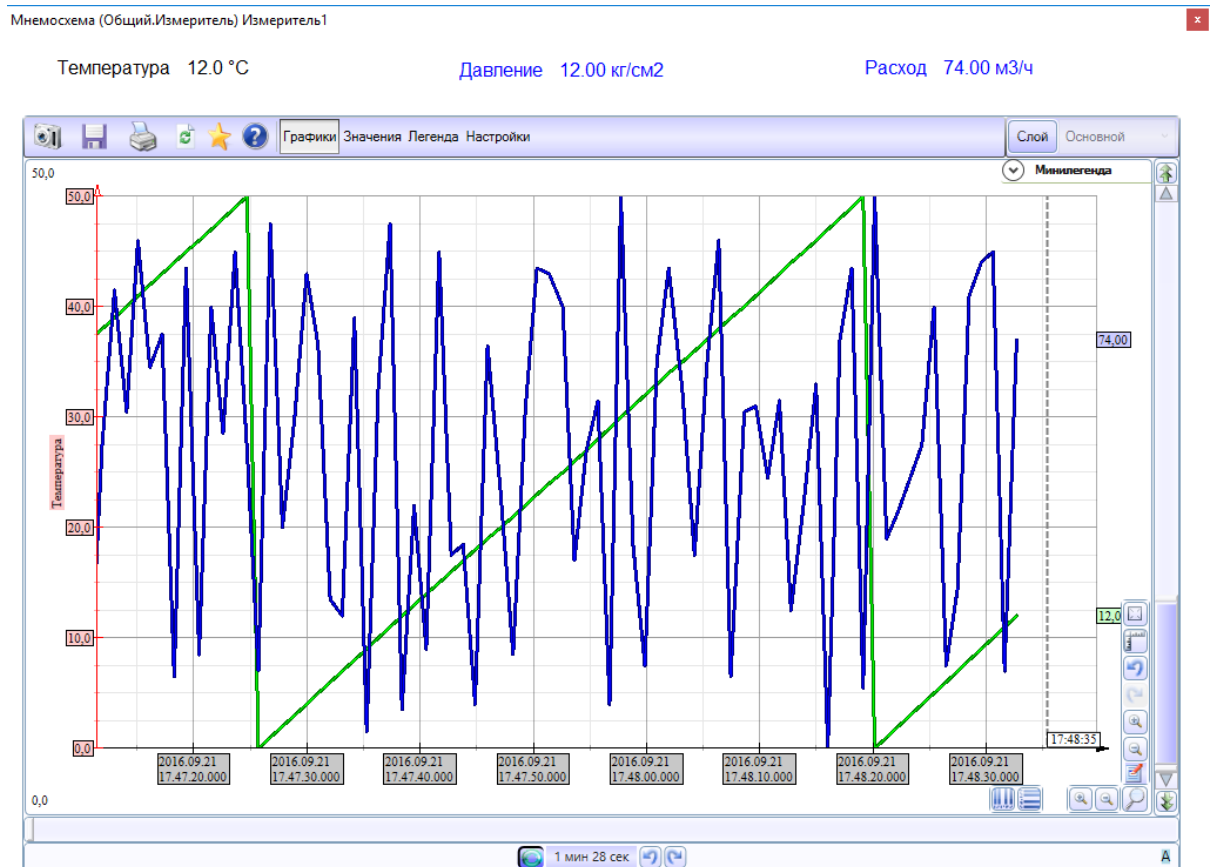


Рисунок 3-62. Мнемосхема типизированного объекта в режиме исполнения

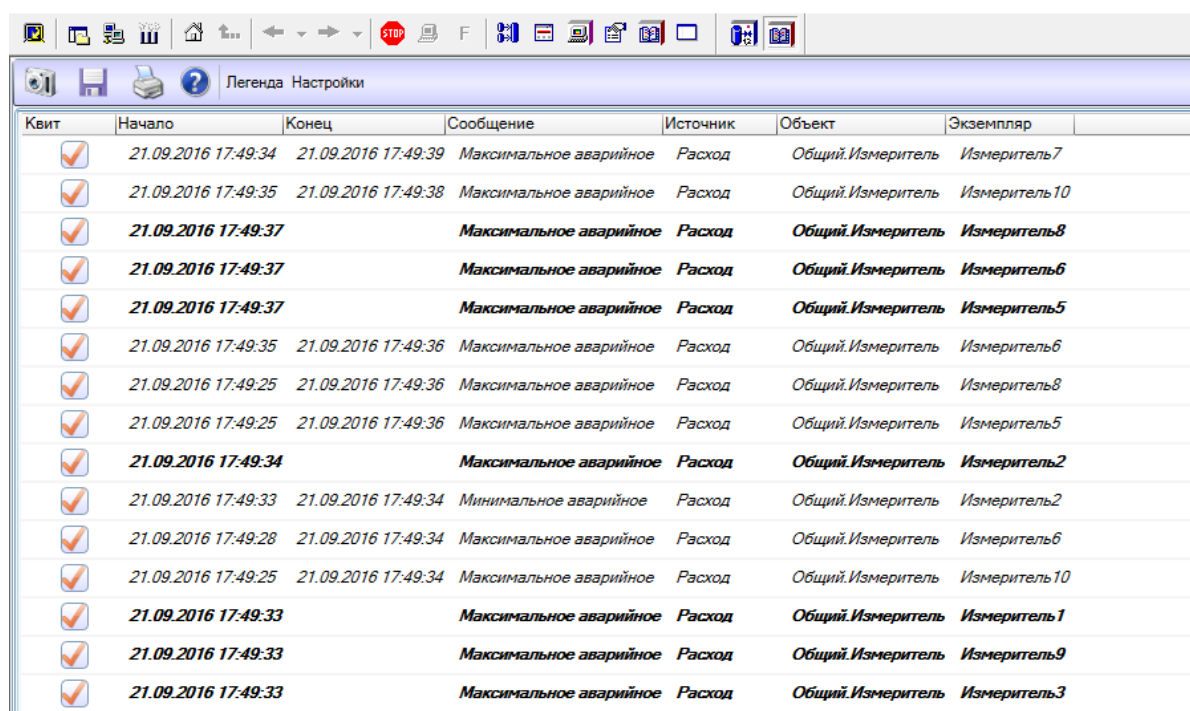
Таким образом, объекты функционируют, архивируют данные и позволяют работать с ними, как с обычными объектами. Типизированные объекты могут содержать и любые

Объектно-ориентированный подход

другие окна: журналы, тренды, отчеты. Принцип их создания ничем не отличается от построения обычных проектов.

Для удобства определения в заголовок окна помимо пути к объекту (или названию окна) выводится также имя экземпляра – чтобы иметь возможность точно идентифицировать принадлежность окна к экземпляру.

Журнал, созданный в объекте, содержащем типизированные объекты, будет выводить сообщения из всех типизированных объектов (Рисунок 3-63). Чтобы определить принадлежность сообщения к экземпляру, нужно активировать специальный столбец – **Экземпляр**.



Квит	Начало	Конец	Сообщение	Источник	Объект	Экземпляр
<input checked="" type="checkbox"/>	21.09.2016 17:49:34	21.09.2016 17:49:39	Максимальное аварийное	Расход	Общий.Измеритель	Измеритель7
<input checked="" type="checkbox"/>	21.09.2016 17:49:35	21.09.2016 17:49:38	Максимальное аварийное	Расход	Общий.Измеритель	Измеритель10
<input checked="" type="checkbox"/>	21.09.2016 17:49:37		Максимальное аварийное	Расход	Общий.Измеритель	Измеритель8
<input checked="" type="checkbox"/>	21.09.2016 17:49:37		Максимальное аварийное	Расход	Общий.Измеритель	Измеритель6
<input checked="" type="checkbox"/>	21.09.2016 17:49:37		Максимальное аварийное	Расход	Общий.Измеритель	Измеритель5
<input checked="" type="checkbox"/>	21.09.2016 17:49:35	21.09.2016 17:49:36	Максимальное аварийное	Расход	Общий.Измеритель	Измеритель6
<input checked="" type="checkbox"/>	21.09.2016 17:49:25	21.09.2016 17:49:36	Максимальное аварийное	Расход	Общий.Измеритель	Измеритель8
<input checked="" type="checkbox"/>	21.09.2016 17:49:25	21.09.2016 17:49:36	Максимальное аварийное	Расход	Общий.Измеритель	Измеритель5
<input checked="" type="checkbox"/>	21.09.2016 17:49:34		Максимальное аварийное	Расход	Общий.Измеритель	Измеритель2
<input checked="" type="checkbox"/>	21.09.2016 17:49:33	21.09.2016 17:49:34	Минимальное аварийное	Расход	Общий.Измеритель	Измеритель2
<input checked="" type="checkbox"/>	21.09.2016 17:49:28	21.09.2016 17:49:34	Максимальное аварийное	Расход	Общий.Измеритель	Измеритель6
<input checked="" type="checkbox"/>	21.09.2016 17:49:25	21.09.2016 17:49:34	Максимальное аварийное	Расход	Общий.Измеритель	Измеритель10
<input checked="" type="checkbox"/>	21.09.2016 17:49:33		Максимальное аварийное	Расход	Общий.Измеритель	Измеритель1
<input checked="" type="checkbox"/>	21.09.2016 17:49:33		Максимальное аварийное	Расход	Общий.Измеритель	Измеритель9
<input checked="" type="checkbox"/>	21.09.2016 17:49:33		Максимальное аварийное	Расход	Общий.Измеритель	Измеритель3

Рисунок 3-63. Журнал с выводом сообщений типизированных объектов

4. Создание проекта с использованием объектного подхода

Темп жизни современного инженера зачастую не позволяет прочесть учебник от корки до корки. Хочется с первых страниц получить всю необходимую информацию, чтобы начать применять полученные знания на практике. В данной главе мы попробуем пошагово разобрать этапы создания нового проекта с использованием объектно-ориентированного подхода для решения простейшей и распространенной задачи – контроля уровня и температуры жидкости в емкости. Это поможет вам «набить руку» в работе с MasterSCADA и лучше понять ее идеологию.

4.1. Постановка задачи

Итак, постановка задачи. В цехе находятся 3 бака с жидкостью, в каждом баке установлен датчик уровня жидкости, а также датчик температуры. Датчик температуры имеет диапазон от 0 до 60 градусов, при этом верхней аварийной границей является 50 градусов. Датчик уровня имеет диапазон от 0 до 300 литров, при этом четких верхней и нижней аварийных границ нет – оператор может менять их произвольно, в зависимости от обстоятельств.

Необходимо реализовать систему мониторинга, позволяющая отображать текущие и архивные данные. На стартовой мнемосхеме должна отображаться общая информация о системе: состояние уровня жидкости в баке, температура, состояние аварий. При выборе оператором нужного бака должна открыться детальная мнемосхема по конкретной емкости – с подробной информацией, доступом к архивным данным, возможностью изменения порогов.

4.2. Настройка дерева системы

Начнем построение проекта с дерева системы. Добавим в систему **Компьютер**.



Рисунок 4-1. Создание «Компьютера»

Создание проекта с использованием объектного подхода

Мы опустим создание конфигурации OPC-сервера – в разных OPC-серверах это делается по-разному. Для OPC-серверов нашей разработки есть видеопримеры, методическая документация и подробная справка. К данному примеру конфигурация OPC-сервера была сделана заранее, вы можете найти ее на диске.

В компьютер мы добавим OPC-сервер *Modbus Universal Master OPC сервер*.

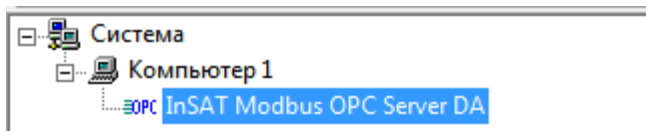


Рисунок 4-2. Вставка OPC-сервера

Примечание. Если OPC-сервер недоступен в меню «Добавить» компьютера, то нужно выполнить пункт «Поиск OPC DA серверов» и отметить его галочкой. Если и в этом окне он не будет доступен, значит, OPC-сервер не установлен или установлен некорректно.

Добавим теги, для этого выполним *Вставить – Все переменные и группы*.

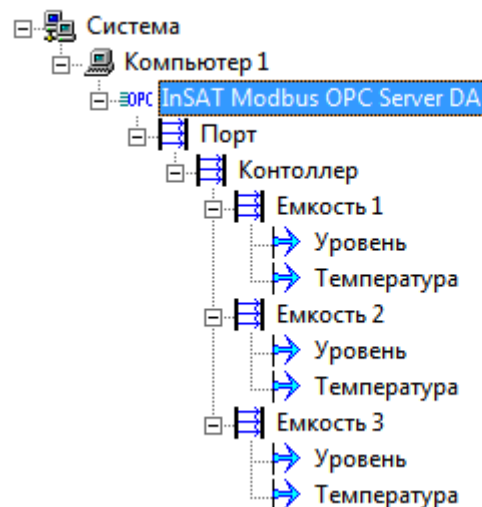


Рисунок 4-3. Группы и теги OPC-сервера

Теги добавились в сервер (Рисунок 4-3) – теперь с ними можно работать в проекте. В OPC-сервере доступно три группы – для каждой из емкости. В каждой группе находится по два тега: *Уровень* и *Температура*.

4.3. Создание объектов

Добавим в *дерево объектов* объект первого уровня, используя контекстное меню. Переименуем объект в *Цех* и назначим исполнение на *Компьютере 1*.

Создание проекта с использованием объектного подхода

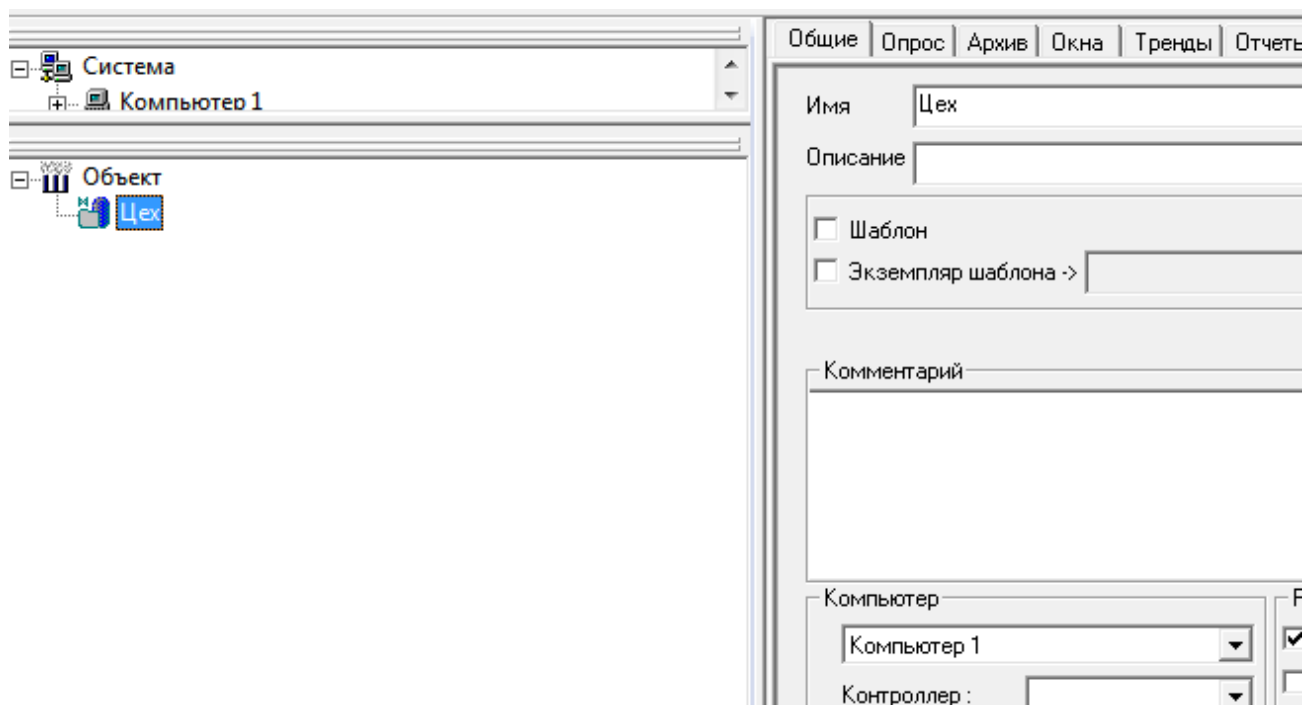


Рисунок 4-4. Создание объекта первого уровня

Добавим в объект еще один объект и назовем его **Емкость 1**.

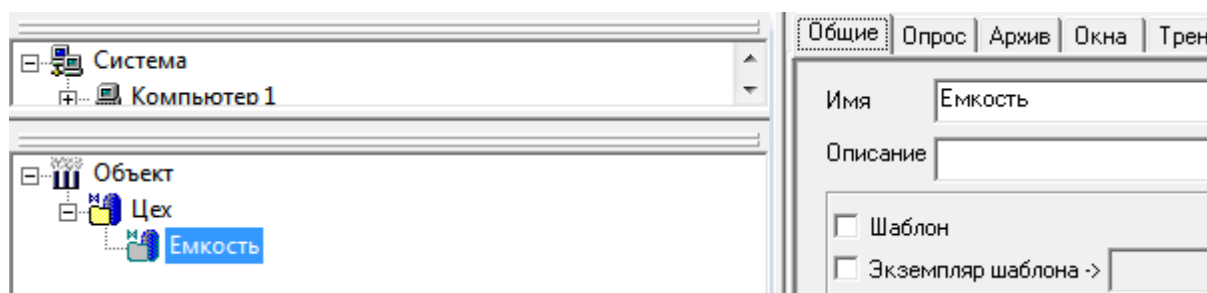


Рисунок 4-5. Создание объекта «Емкость»

Добавим в эту емкость переменные, которые позволят нам настроить систему мониторинга. Начнем непосредственно с емкости. Выберем в палитре **ФБ** раздел **Аппараты**, выберем **ФБ Емкость** и добавим его в объект.

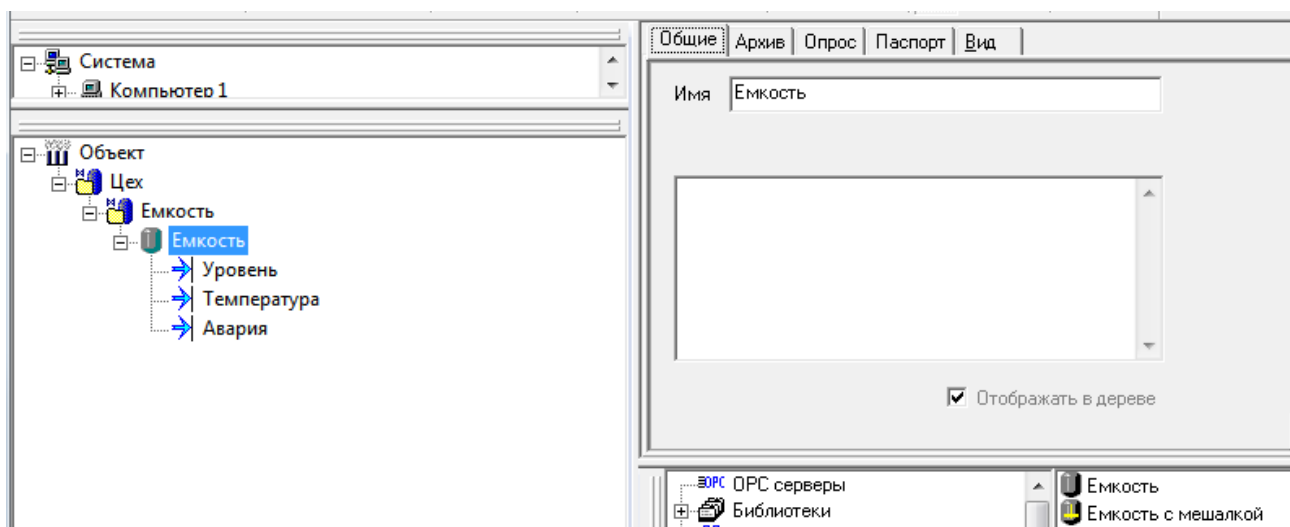


Рисунок 4-6. Добавление ФБ «Емкость»

У данного ФБ есть три входа: **Уровень**, **Температура** и **Авария**. В соответствии со значением на входе **Уровень** меняется величина заполнения жидкости в контроле бака. По умолчанию входной диапазон – от 0 до 100.



Рисунок 4-7. Изменение контроля при изменении входа «Уровень»

В соответствии со значением на входе **Температура** меняется окраска жидкости в баке. Диапазон также от 0 до 100.



Рисунок 4-8. Изменение контроля при изменении входа «Температура»

При сигнале на входе **Авария** контроль бака начинает мигать.

Создание проекта с использованием объектного подхода

Свяжем OPC-переменные из группы **Емкость 1** с соответствующими входами **ФБ Емкость**.

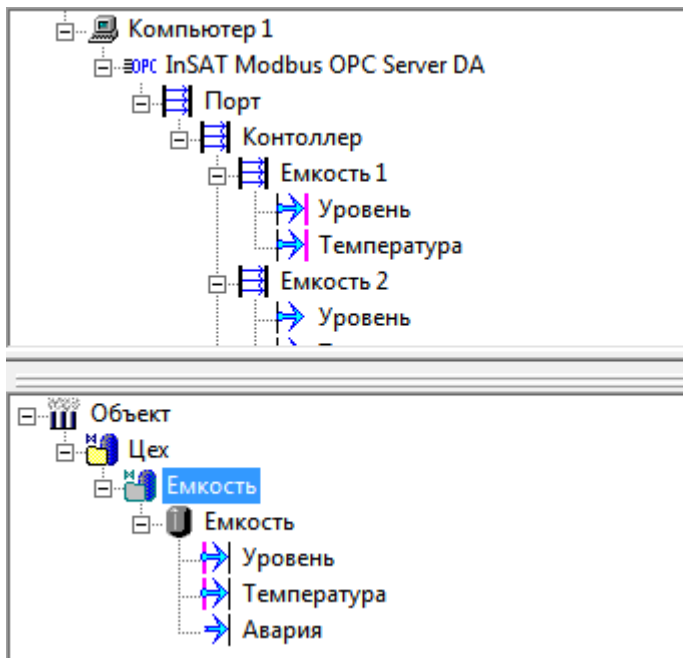


Рисунок 4-9. Установка связей с деревом системы

Включим сервис внешних связей (Рисунок 4-10) – тогда все выходные переменные объекты будут скомпонованы в одном месте. Подробнее данный механизм описан в разделе **Объектно-ориентированный подход – Инкапсуляция**.

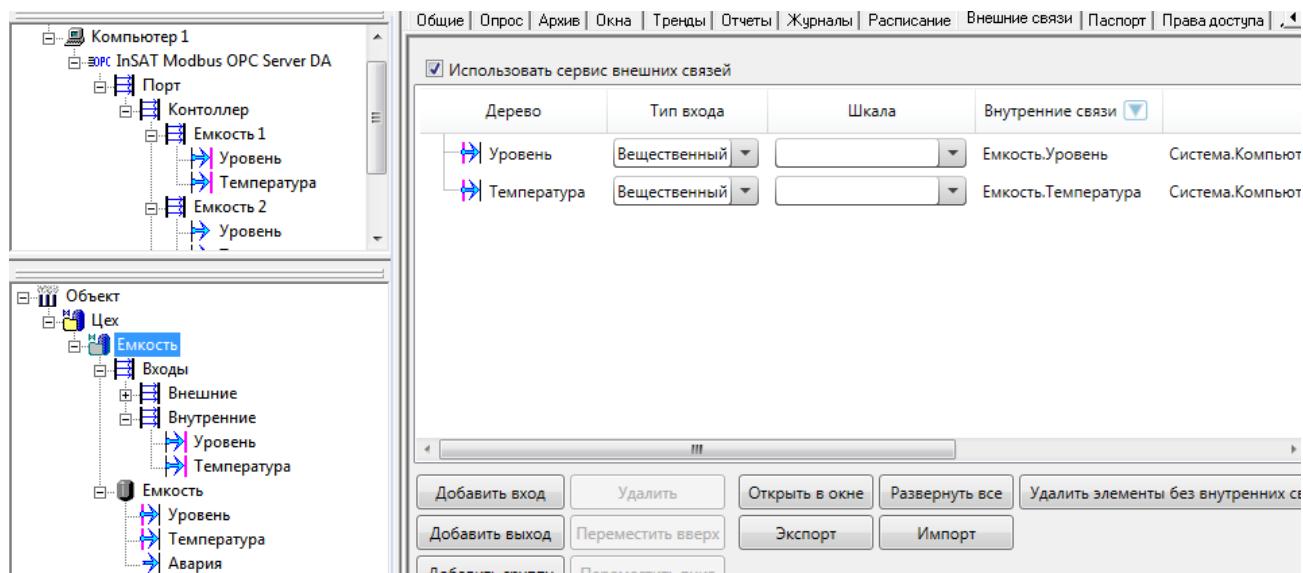


Рисунок 4-10. Включение сервиса внешних связей

Создание проекта с использованием объектного подхода

Теперь добавим контрол емкости на мнемосхему. Выделим объект Емкость и на вкладке Окна создадим мнемосхему.

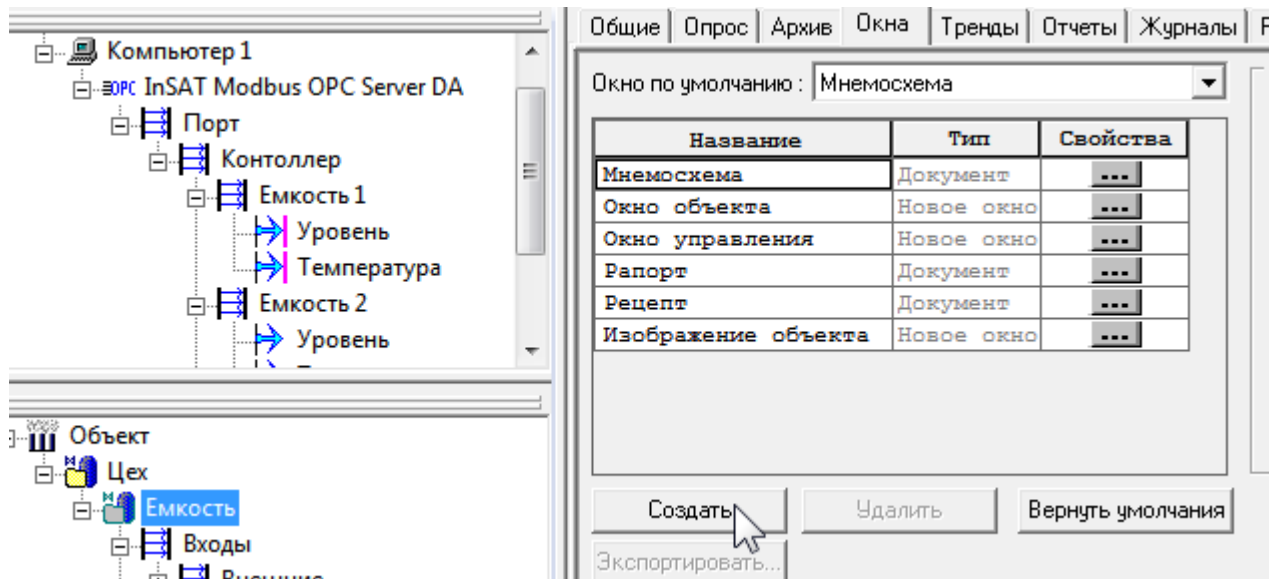


Рисунок 4-11. Создание мнемосхемы объекта «Емкость»

В окно открывшегося редактора мнемосхем перетащим ФБ Емкость – будет создан контрол.

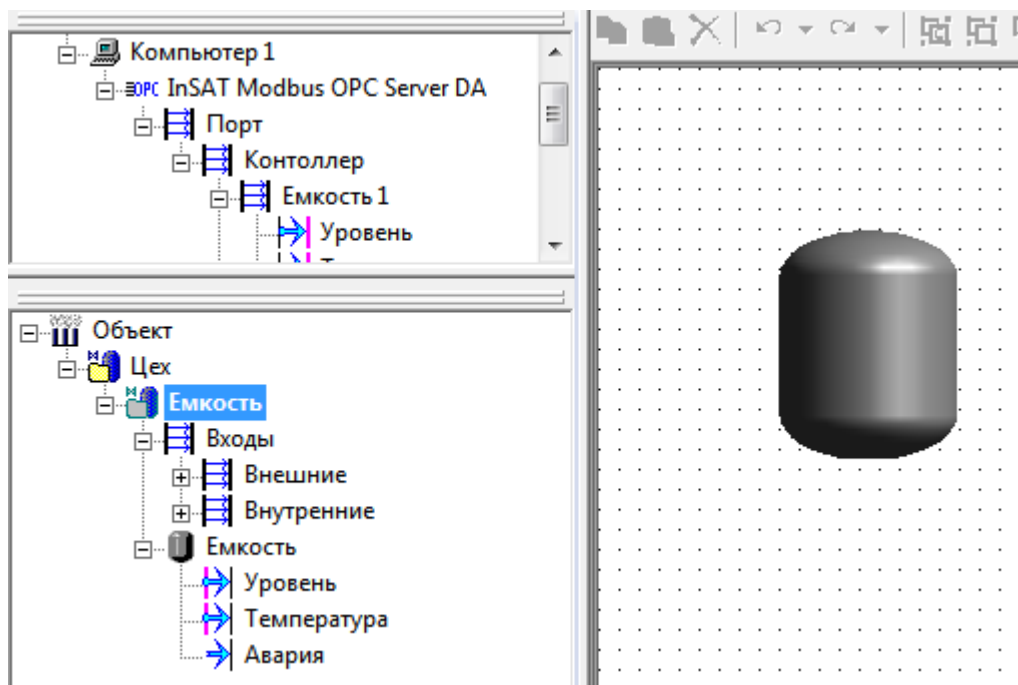



Рисунок 4-12. Создание контроля емкости

Проверим работу ФБ – запустим режим исполнения, нажав на кнопку .

Откроем нашу мнемосхему – выделим объект и нажмем на кнопку **Открыть**.

Примечание. Аналогично можно вызвать у объекта контекстное меню и выбрать пункт меню «Открыть».

Также включим видимость дерева объектов – нажмем на кнопку  в панели инструментов.

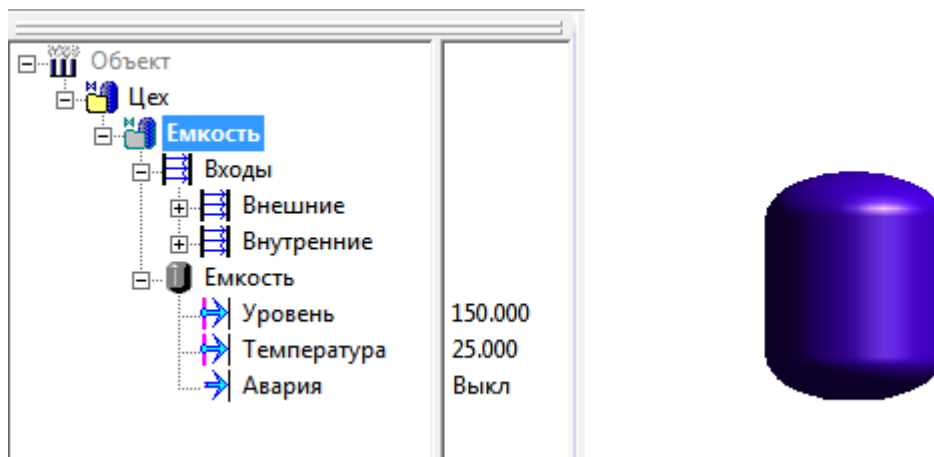


Рисунок 4-13. Состояние контроля в режиме исполнения

В настоящий момент наша емкость окрашена целиком – система показывает, что бак переполнен. Однако если открыть дерево объектов и посмотреть на поступающие значения, то мы увидим, что значение уровня всего лишь 150 литров – то есть бак заполнен только наполовину (Рисунок 4-13).

Дело в том, что по умолчанию вход данного ФБ имеет диапазон от 0 до 100, а наш реальный бак имеет диапазон от 0 до 300. Чтобы устранить данное несоответствие, необходимо использовать шкалы.

4.4. Создание шкал

Шкала – это параметр переменной, который определяет диапазон переменной (нижнюю и верхнюю границу), единицу измерения, формат отображения переменной на мнемосхеме, а также аварийные границы.

Создание шкал производится на вкладке **Системы – Шкалы – Аналоговые** (Рисунок 4-14).

Создание проекта с использованием объектного подхода

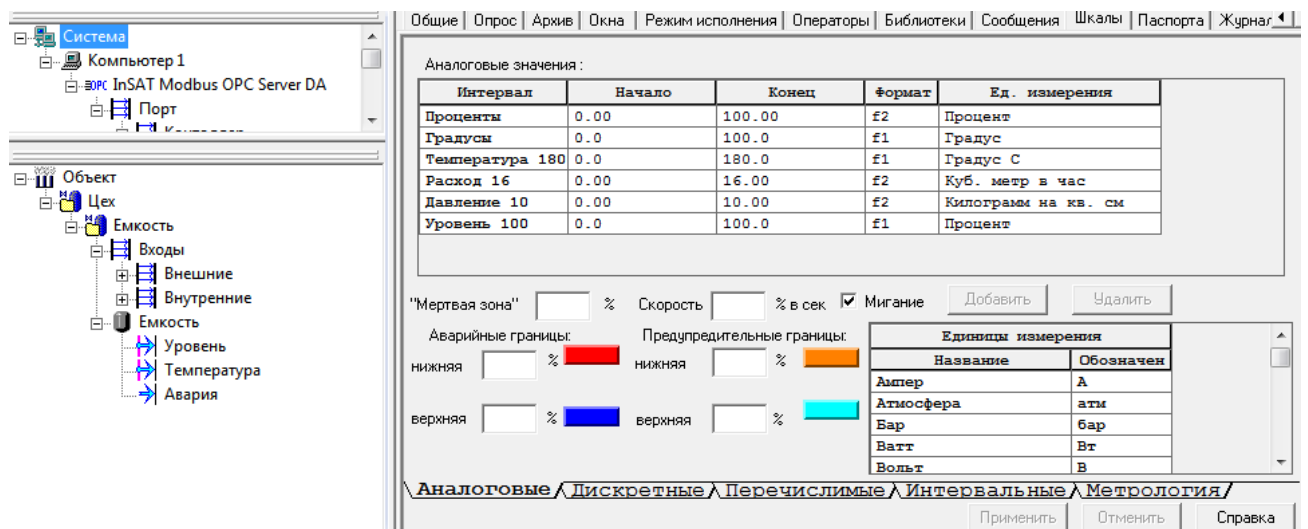


Рисунок 4-14. Вкладка шкал

Нам необходимы две шкалы: уровня и температуры. Обе эти шкалы уже присутствуют в списке, но это шкалы системные – их нельзя редактировать. Поэтому сделаем собственные. Для этого щелкнем на любой ячейке таблицы и нажмем кнопку **Добавить**.

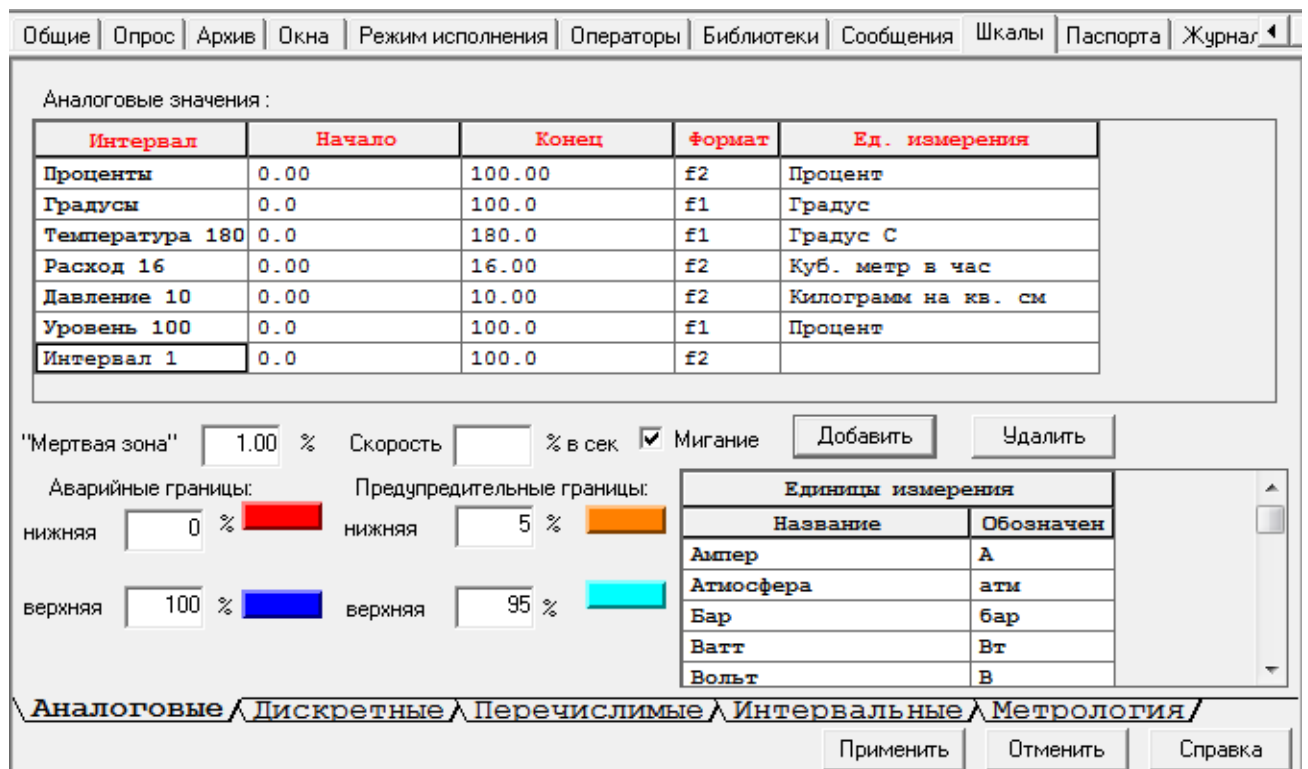


Рисунок 4-15. Добавление шкалы

Сначала сделаем шкалу температуры. Создадим новую шкалу и переименуем ее в **Температура**. Зададим верхнюю границу – 60, единицу измерения – градусы.

Создание проекта с использованием объектного подхода

Теперь зададим верхние и нижние границы – для этого в нижней части вкладки присутствуют специальные поля. По заданию необходим контроль верхней аварийной границы в 50 градусов. В настройках шкалы аварийные границы вводятся в процентах шкалы, 50 градусов по отношению к 60 будет 83 % – введем это число в поле верхней аварийной границы, остальные поля оставим пустыми.

Общие | Опрос | Архив | Окна | Режим исполнения | Операторы | Библиотеки | Сообщения | Шкалы | Паспорта | Журнал

Аналоговые значения :

Интервал	Начало	Конец	Формат	Ед. измерения
Проценты	0.00	100.00	f2	Процент
Градусы	0.0	100.0	f1	Градус
Температура 180	0.0	180.0	f1	Градус С
Расход 16	0.00	16.00	f2	Куб. метр в час
Давление 10	0.00	10.00	f2	Килограмм на кв. см
Уровень 100	0.0	100.0	f1	Процент
Температура	0.0	60	f2	Градус

"Мертвая зона" % Скорость % в сек Мигание

Аварийные границы: Предупредительные границы:

нижняя % нижняя %

верхняя % верхняя %

Единицы измерения	
Название	Обозначен
Ампер	А
Атмосфера	атм
Бар	бар
Ватт	Вт
Вольт	В

Аналоговые / Дискретные / Перечислимые / Интервальные / Метрология

Рисунок 4-16. Создание шкалы температуры

Аналогично сделаем шкалу уровня. Зададим верхнюю границу – 300, единицу измерения – литр (Рисунок 4-17).

Создание проекта с использованием объектного подхода

Общие | Опрос | Архив | Окна | Режим исполнения | Операторы | Библиотеки | Сообщения | Шкалы | Паспорта | Журнал

Аналоговые значения :

Интервал	Начало	Конец	Формат	Ед. измерения
Градусы	0.0	100.0	f1	Градус
Температура 180	0.0	180.0	f1	Градус С
Расход 16	0.00	16.00	f2	Куб. метр в час
Давление 10	0.00	10.00	f2	Килограмм на кв. см
Уровень 100	0.0	100.0	f1	Процент
Температура	0.0	60	f2	Градус
Уровень	0.0	300	f2	Литр

"Мертвая зона" % Скорость % в сек Мигание

Аварийные границы: Предупредительные границы:

нижняя % нижняя %

верхняя % верхняя %

Единицы измерения	
Название	Обозначен
Ампер	А
Атмосфера	атм
Бар	бар
Ватт	Вт
Вольт	В

Аналоговые / Дискретные / Перечислимые / Интервальные / Метрология

Рисунок 4-17. Создание шкалы уровня

Однако по заданию для шкалы уровня границы должен менять оператор индивидуально для каждого бака, поэтому задание границ для шкалы нам не подойдет. Ввод значений для ручного контроля границ мы сделаем позже, пока оставим все поля границ пустыми.

Теперь, когда шкалы назначены, их можно назначить переменным. Выделим вход Уровень у ФБ Емкость.

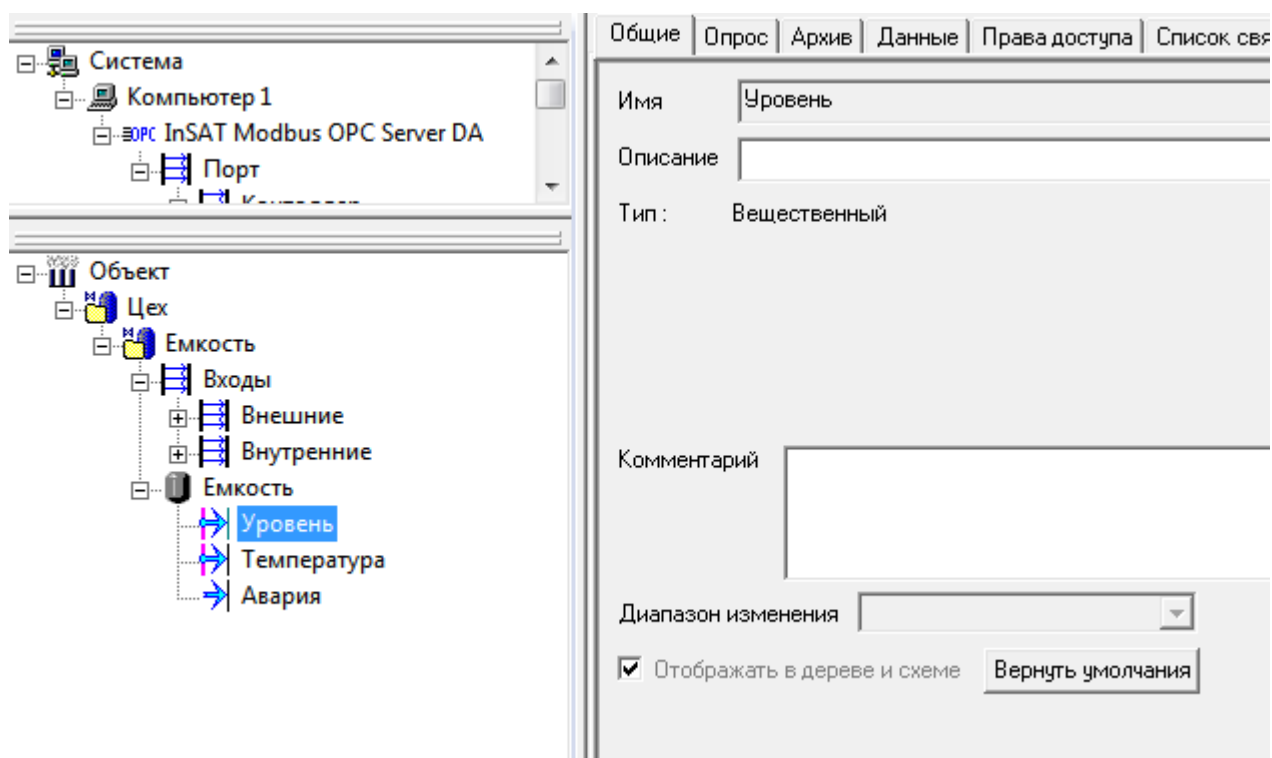


Рисунок 4-18. Вход «Уровень» ФБ «Емкость»

Для задания шкалы предназначено поле **Диапазон измерения**. Однако у данного входа оно заблокировано – дело в том, что входы наследуют настройки источника. То есть наш вход унаследует шкалу у выхода – в данном случае у выхода сервиса внешних связей. Поэтому, чтобы шкала применилась, нужно назначить шкалу выходу. Это можно сделать, выделив выход в группе **Внутренние** (Рисунок 4-19).

Создание проекта с использованием объектного подхода

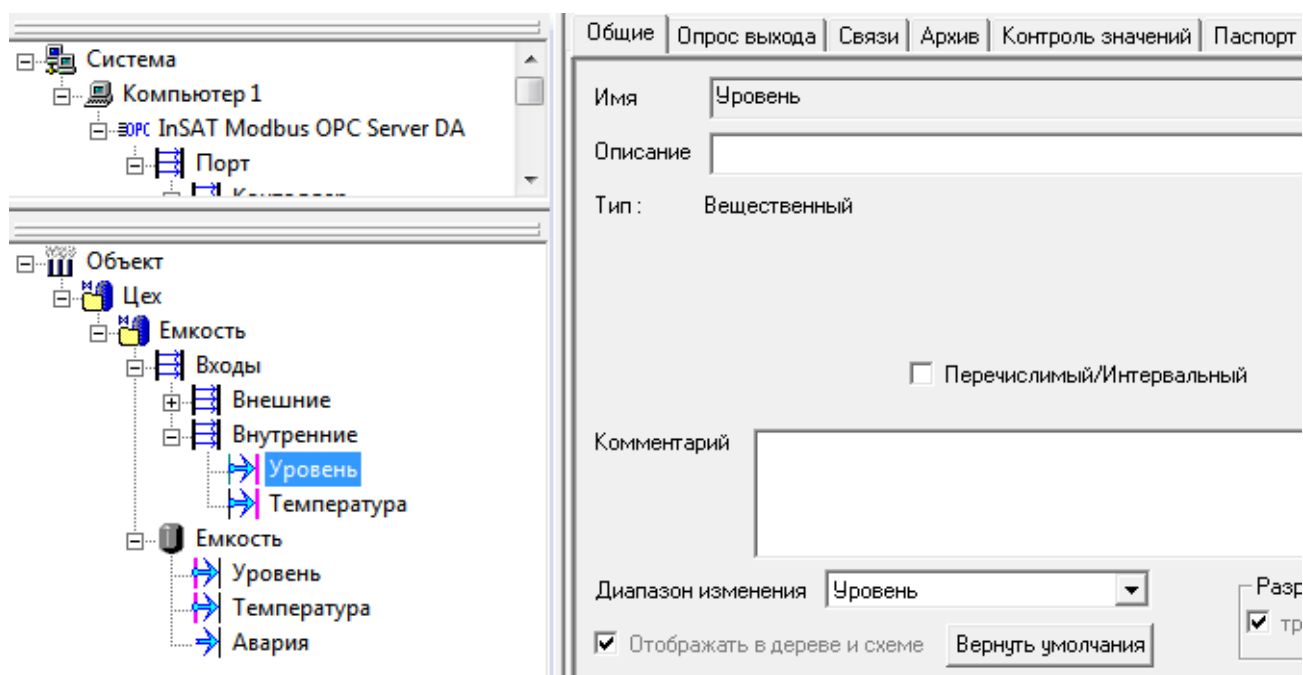


Рисунок 4-19. Назначение шкалы уровню выходу сервиса внешних связей

Аналогично назначим шкалу выходу *Температура*.

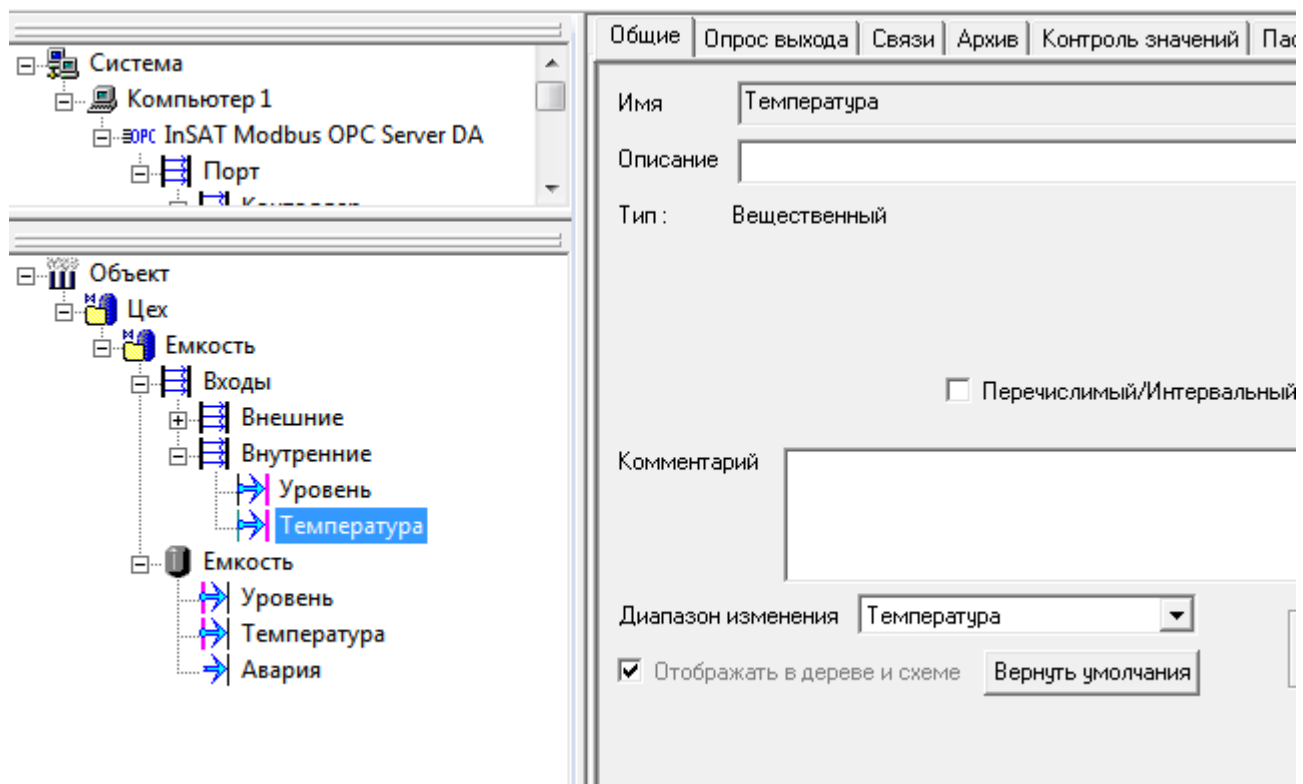


Рисунок 4-20. Назначение шкалы выходу «Температура»

Создание проекта с использованием объектного подхода

Проверим работу – запустим режим исполнения и посмотрим состояние нашей емкости.

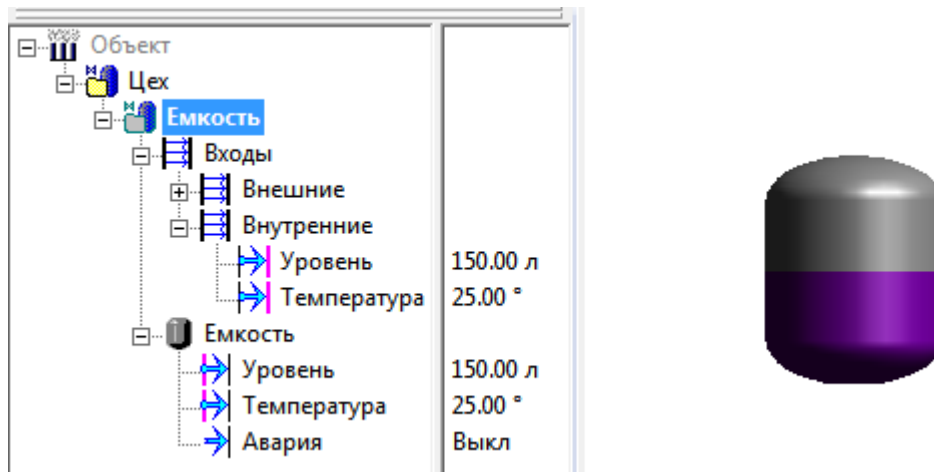


Рисунок 4-21. Вид контроля после назначения шкал

Теперь она окрашена корректно (Рисунок 4-21).

4.5. Ручной ввод аварийных границ

Теперь реализуем ввод аварийных границ для переменной уровня.

Ввод значений в MasterSCADA производится с помощью команд. Добавим в объект две команды: **ВАГ** (верхняя аварийная граница) и **НАГ** (нижняя аварийная граница).

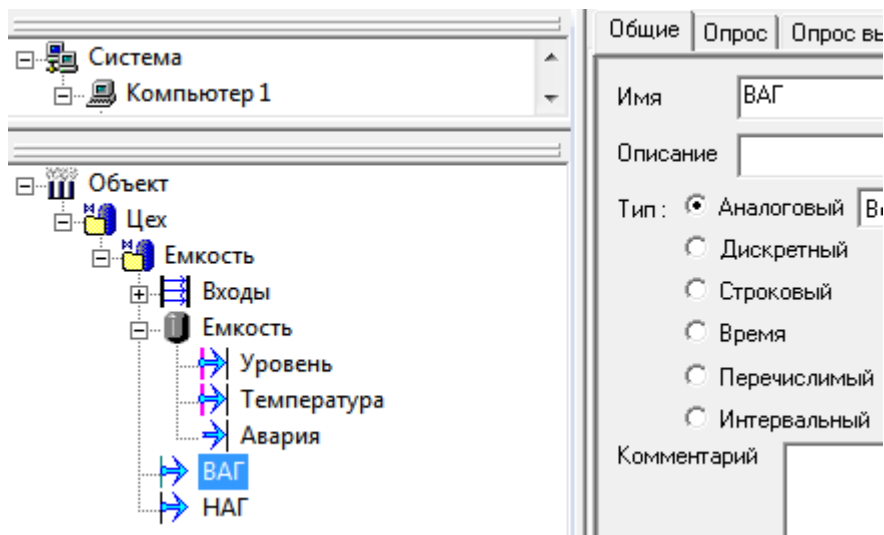


Рисунок 4-22. Создание команд ручного ввода аварийных границ

Поскольку значения уровня должны сохраняться при перезапуске, включим **Восстановление при рестарте** – это делается на вкладке **Опрос выхода**. Также желательно

включить **Значения до опроса** – чтобы команды имели бы значения до того, как оператор их в первый раз ввел.

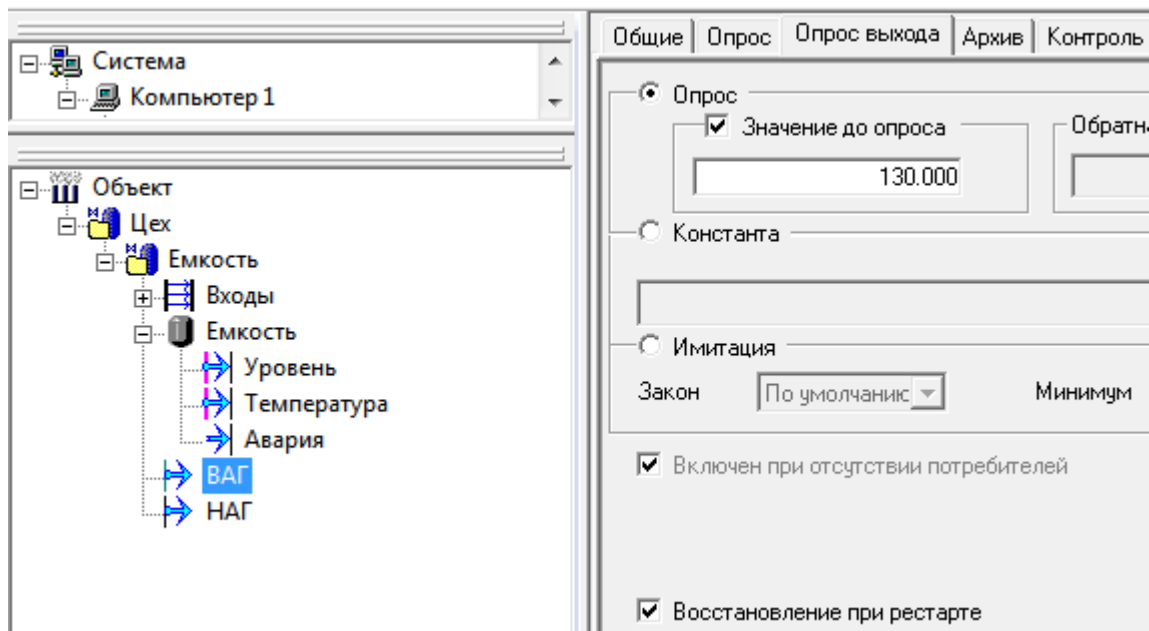


Рисунок 4-23. Задание значения до опроса и восстановления

Теперь сделаем так, чтобы данные команды стали бы аварийными границами уровня. Для этого выделим переменную **Уровень** во внешних связях: **Входы – Внутренние**. Перейдем на вкладку **Контроль значений** (Рисунок 4-24). На данной вкладке есть 4 поля со значениями аварийных границ. Когда значения введены у шкалы, то данные поля содержат константы, рассчитанные от диапазона шкалы и непосредственно аварийных границ. Поскольку мы для шкалы уровня границ не вводили, то эти поля пусты.

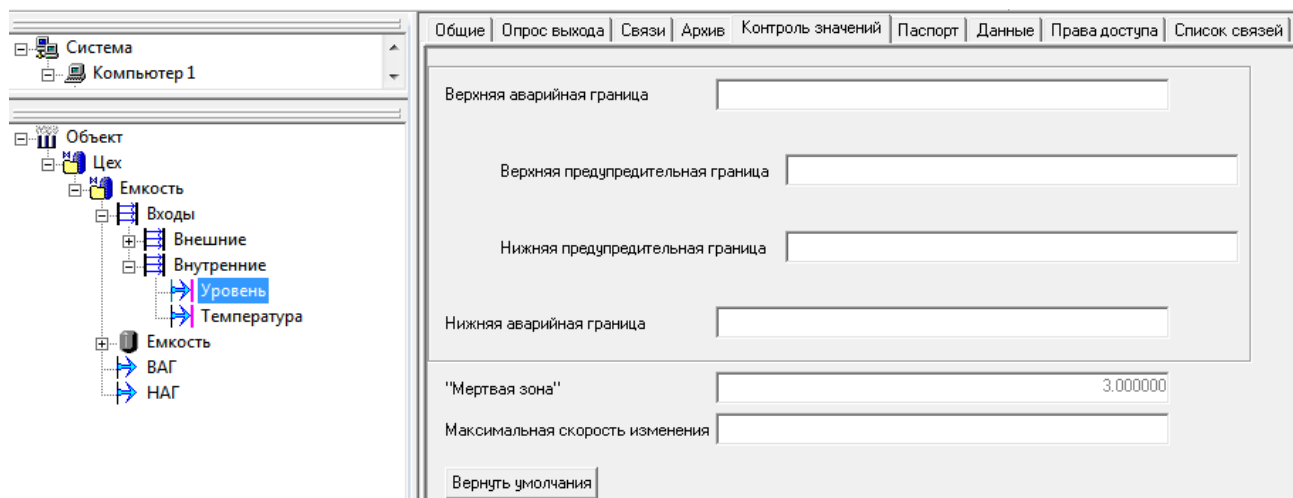


Рисунок 4-24. Вкладка «Контроль значений»

Перетащим в них наши команды ВАГ и НАГ.

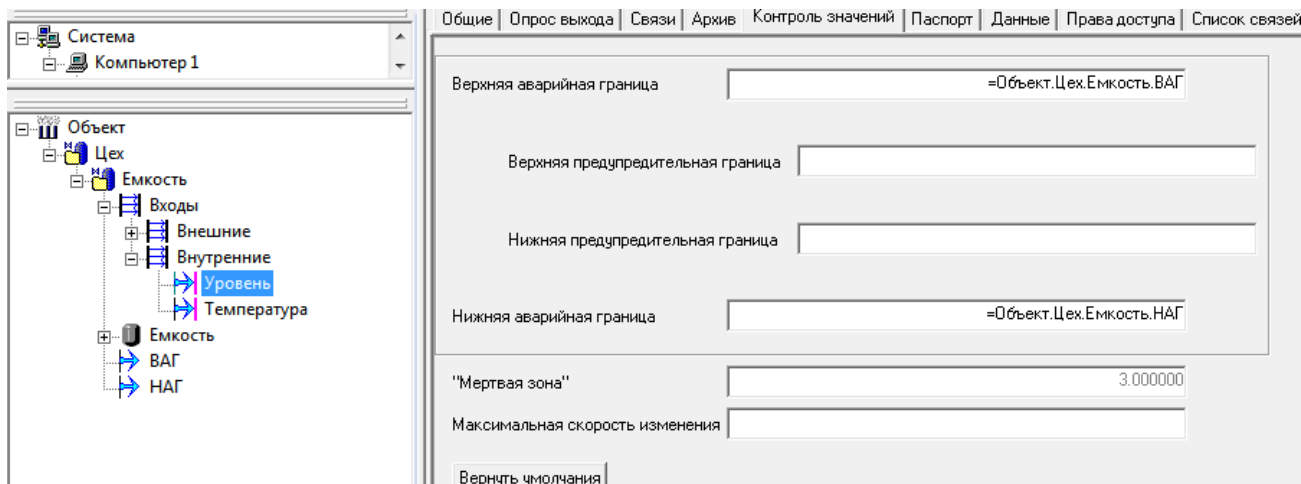


Рисунок 4-25. Привязка команд для ручного ввода границ

Теперь значения аварийных границ привязаны к значениям команд.

Далее реализуем ввод значений. Откроем мнемосхему и вытащим наши команды на мнемосхему правее емкости.

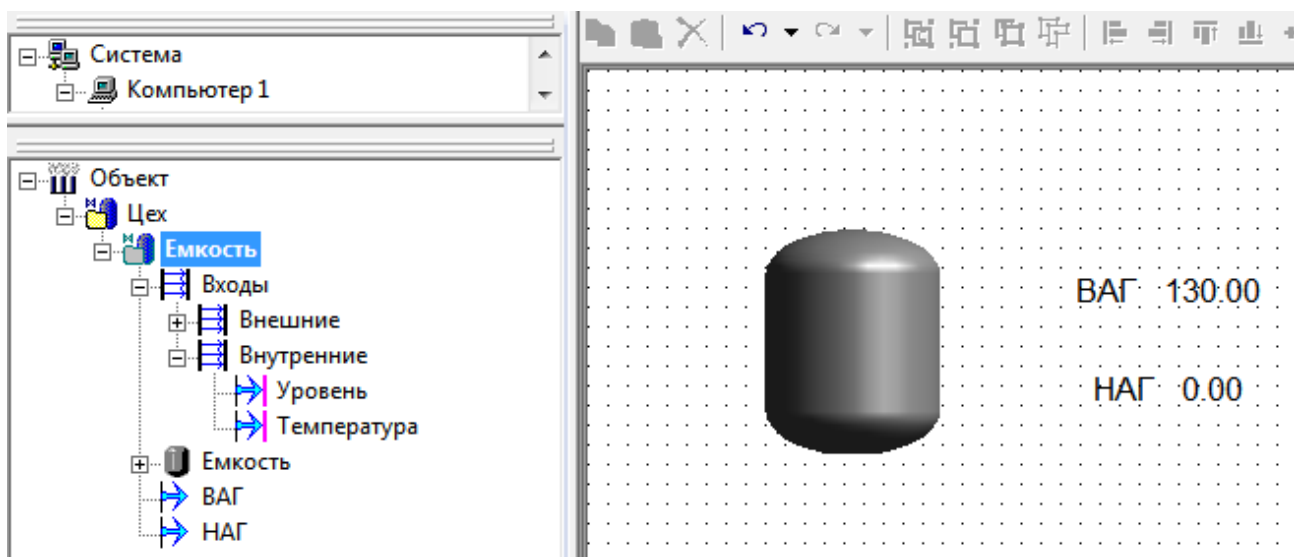


Рисунок 4-26. Создание контролов ввода границ

4.6. Создание журнала сообщений

При нарушении границ будут генерироваться сообщения. Они будут выводиться в виде всплывающих окон. Сделаем также специальный журнал сообщений, где можно будет детально просмотреть историю сообщений.

Создание проекта с использованием объектного подхода

Для этого выделим объект и перейдем на вкладку **Журналы**. Выделим **Основной журнал** и нажмем на кнопку **Создать**.

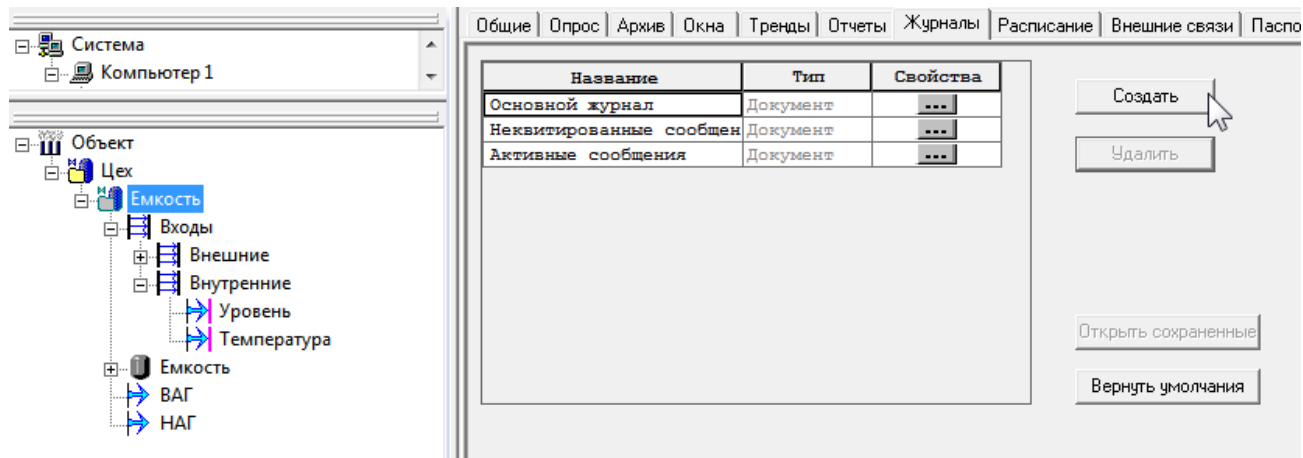


Рисунок 4-27. Создание основного журнала

Будет создан журнал (Рисунок 4-28). Как-либо настраивать его в данный момент не нужно – в него будут попадать все аварийные сообщения, которые возникают в данном объекте. Подробнее журнал сообщений описан в специальном разделе данного руководства.

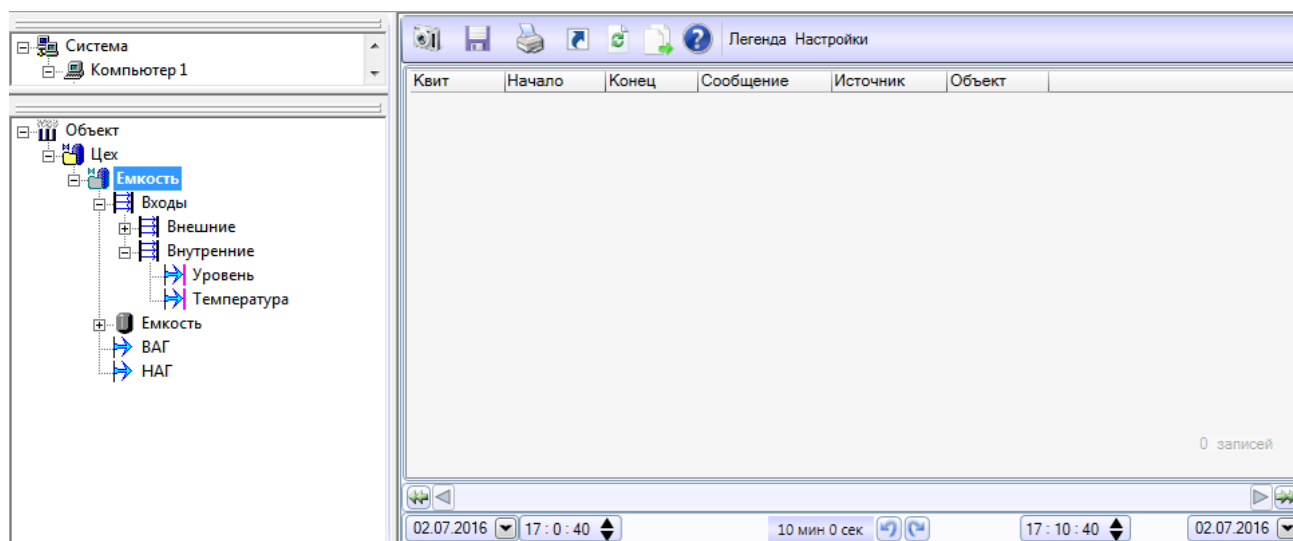


Рисунок 4-28. Журнал сообщений

Теперь сделаем так, чтобы на данный тренд можно было перейти с мнемосхемы. Для этого откроем мнемосхему, вытащим на нее объект правой кнопки мыши и выберем в списке **Журнал**.

Создание проекта с использованием объектного подхода

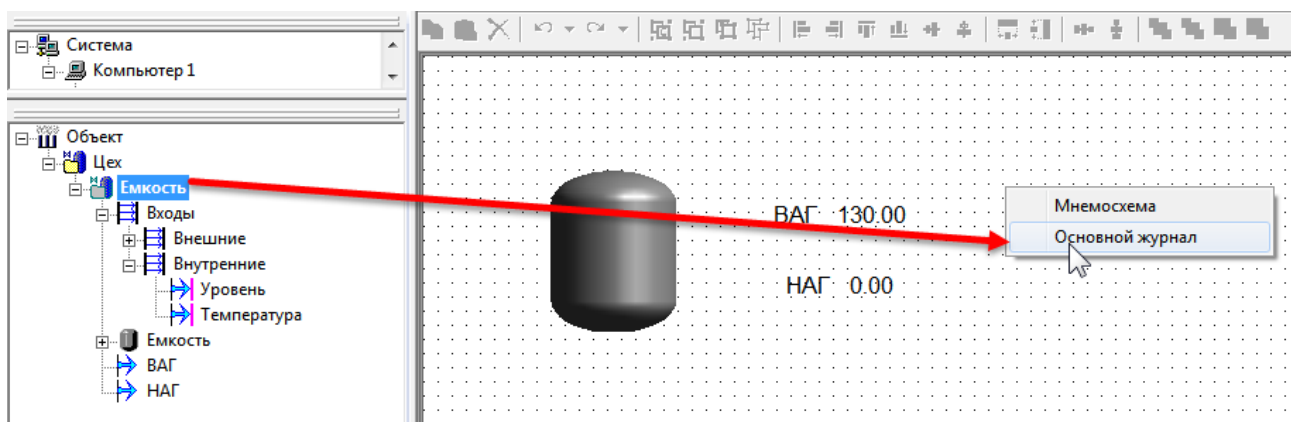


Рисунок 4-29. Создание кнопки открытия журнала

Будет создана кнопка, нажатие на которую в режиме исполнения приведет к открытию журнала.

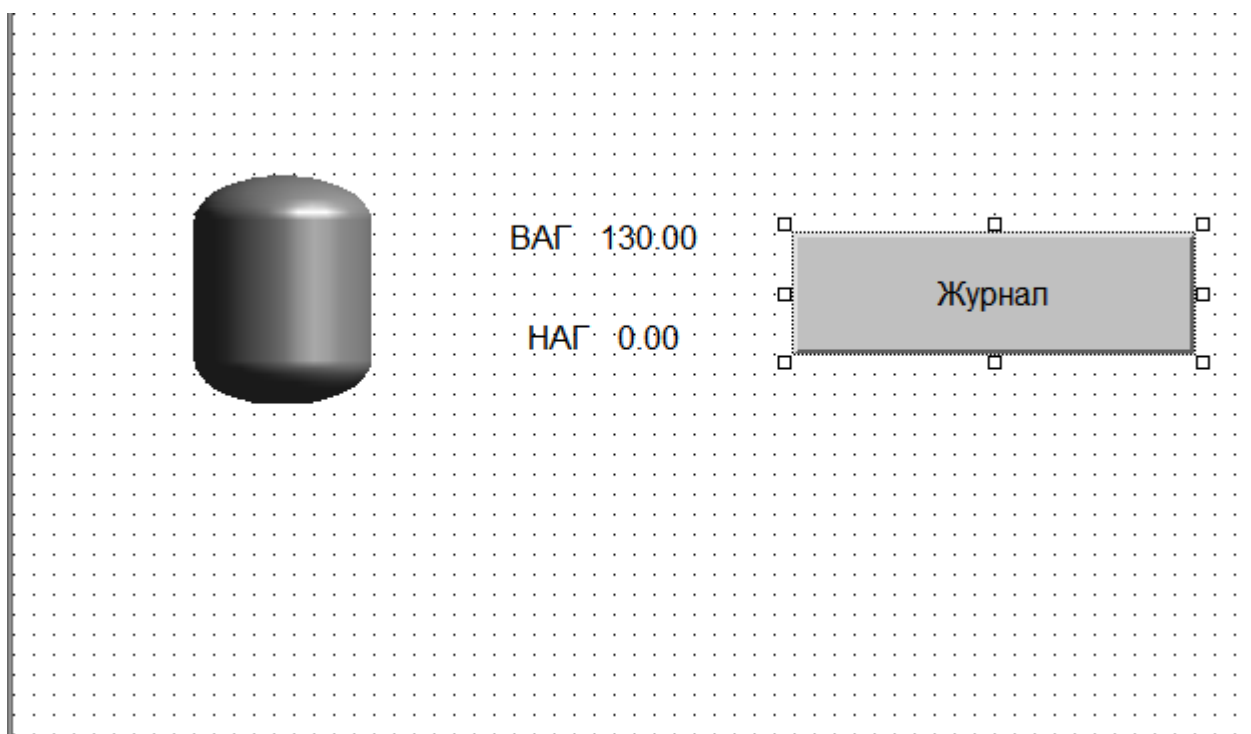


Рисунок 4-30. Кнопка открытия журнала

4.7. Создание тренда

Теперь создадим тренд – в него мы будем выводить как меняется уровень емкости.

Тренд (как, впрочем, и журнал) может быть реализован в двух вариантах: как контроль мнемосхемы или как самостоятельный документ (окно). Реализуем оба варианта – на

Создание проекта с использованием объектного подхода

мнемосхеме разместим оперативный тренд, в котором будет выводиться только уровень емкости, а в отдельном тренде разместим еще также переменную температуры.

Откроем мнемосхему, перейдем в палитру мнемосхемы – в раздел **Окна**. В списке найдем компонент **Тренд** и разместим его на мнемосхеме.

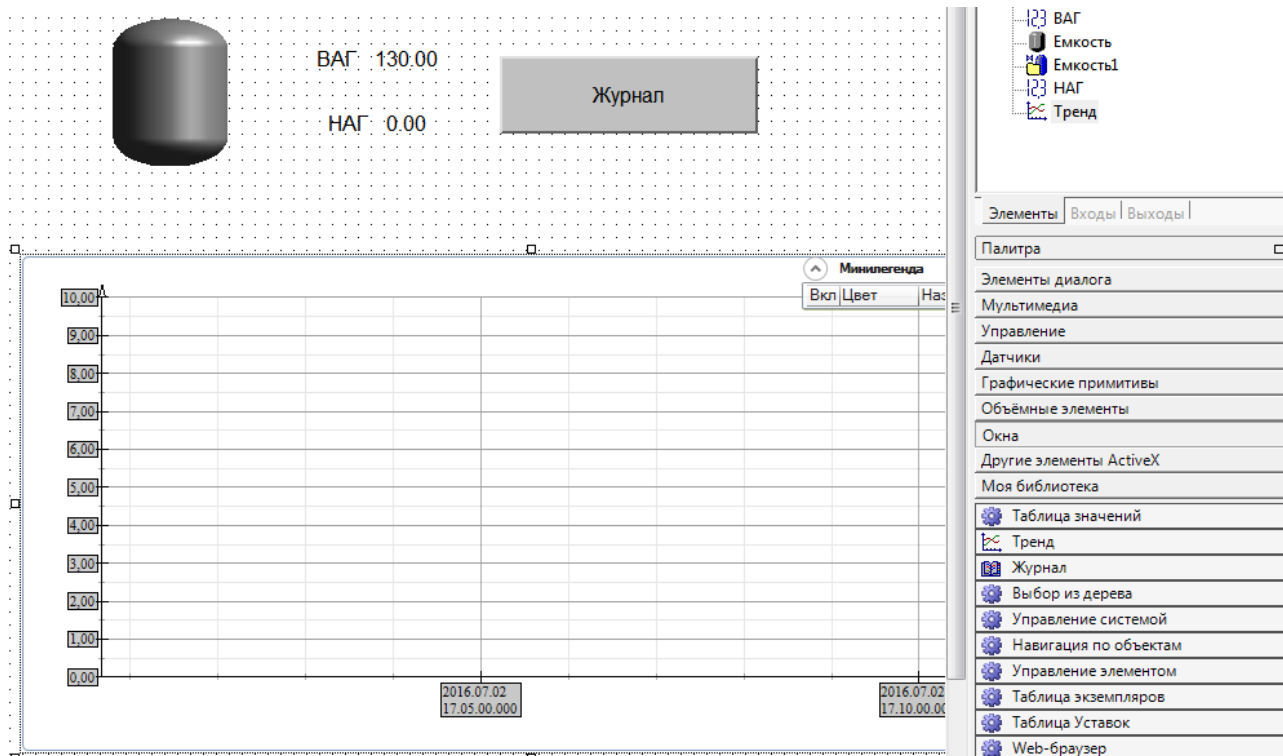


Рисунок 4-31. Создание тренда на мнемосхеме

Добавим на тренд переменную – перетащим переменную уровня из группы внешних связей (из группы **Внутренние**).

Создание проекта с использованием объектного подхода

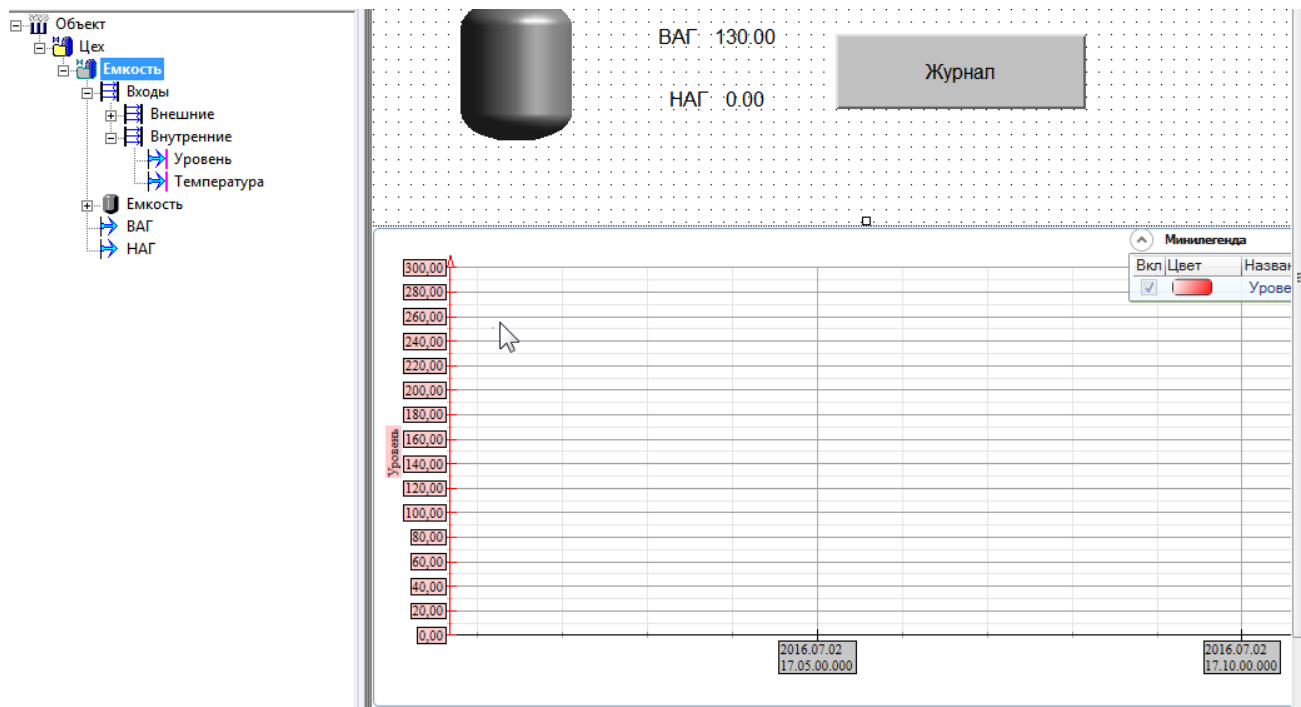


Рисунок 4-32. Добавление переменной в тренд

На тренд добавилось перо – в режиме исполнения оно будет рисовать график изменения переменной (Рисунок 4-32).

Теперь сделаем самостоятельный тренд. Выделим объект, перейдем на вкладку **Тренд** и нажмем кнопку **Добавить**.

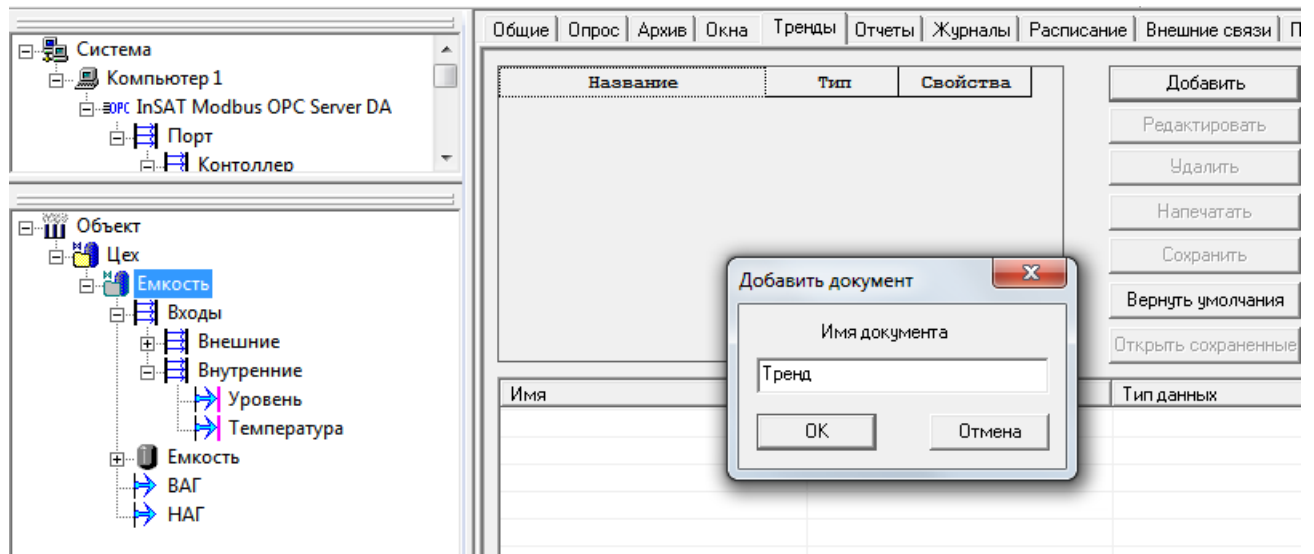


Рисунок 4-33. Создание тренда документа

Откроется документ тренда. Также вытащим на него переменные из дерева объектов – уровень и температуру. В тренд добавятся одноименные перья.

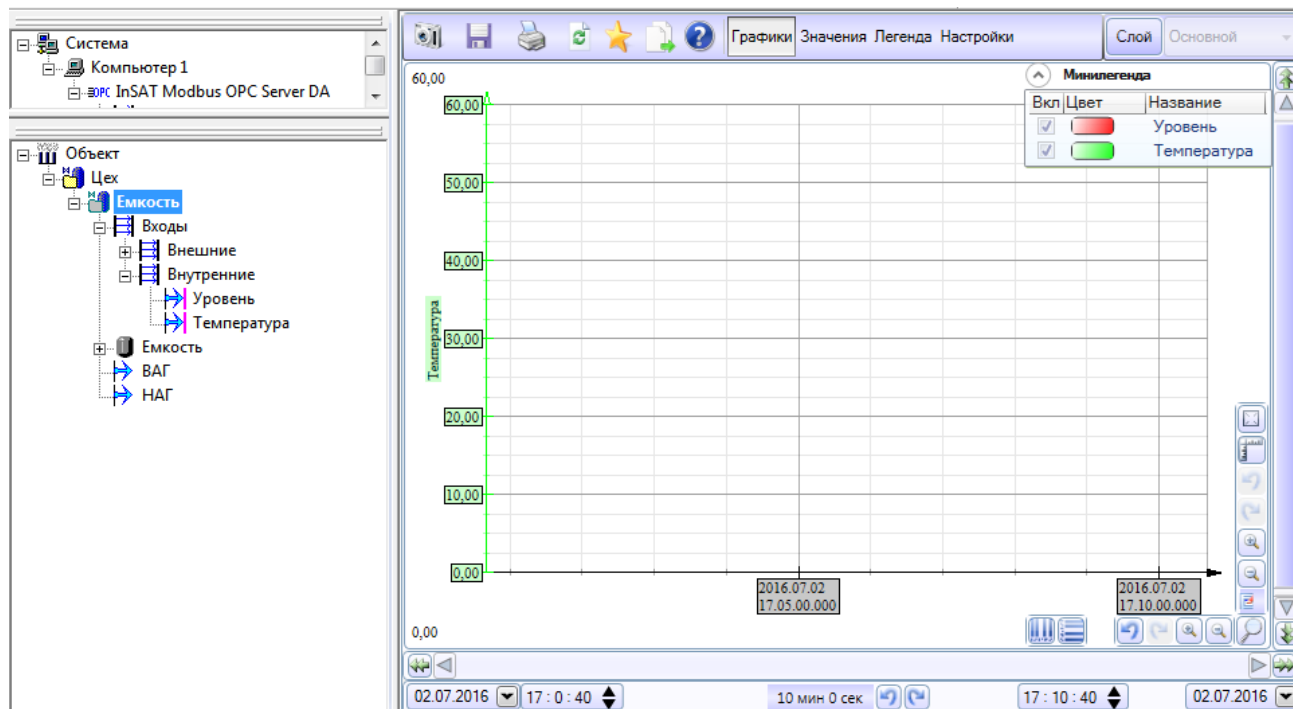


Рисунок 4-34. Добавление перьев в тренд

Создадим кнопку открытия тренда – аналогично тому, как делали это для журнала.

4.8. Настройка архивации данных

Куда будут писаться архивируемые данные и сообщения? По умолчанию архивация идет в файловый архив – специальную структуру файлов, расположенную в папке проекта. Кроме того, архивацию можно назначить в различные базы данных: **MS SQL**, **Oracle**, **Firebird**, **PostgreSQL**.

Разработчик проекта может настроить длительность и объем хранения. Настройка производится у **Компьютера**, после чего наследуется объектами, исполняемыми на этом **Компьютере**. У каждого объекта настройки архива (как и многие другие) можно переопределить.

По умолчанию длительность хранения – 30 дней, а объем ограничен 30 мегабайтами на объект. Настройки можно изменить – задать требуемую длительность и объем хранения. В настоящий момент, в связи с ростом объема жестких дисков, ограничение на объем

Создание проекта с использованием объектного подхода

хранения теряет актуальность, поэтому можно задать заведомо большое значение настройки **Объем на объект** (10 000 Мб и выше) и ограничивать только длительность хранения архива.

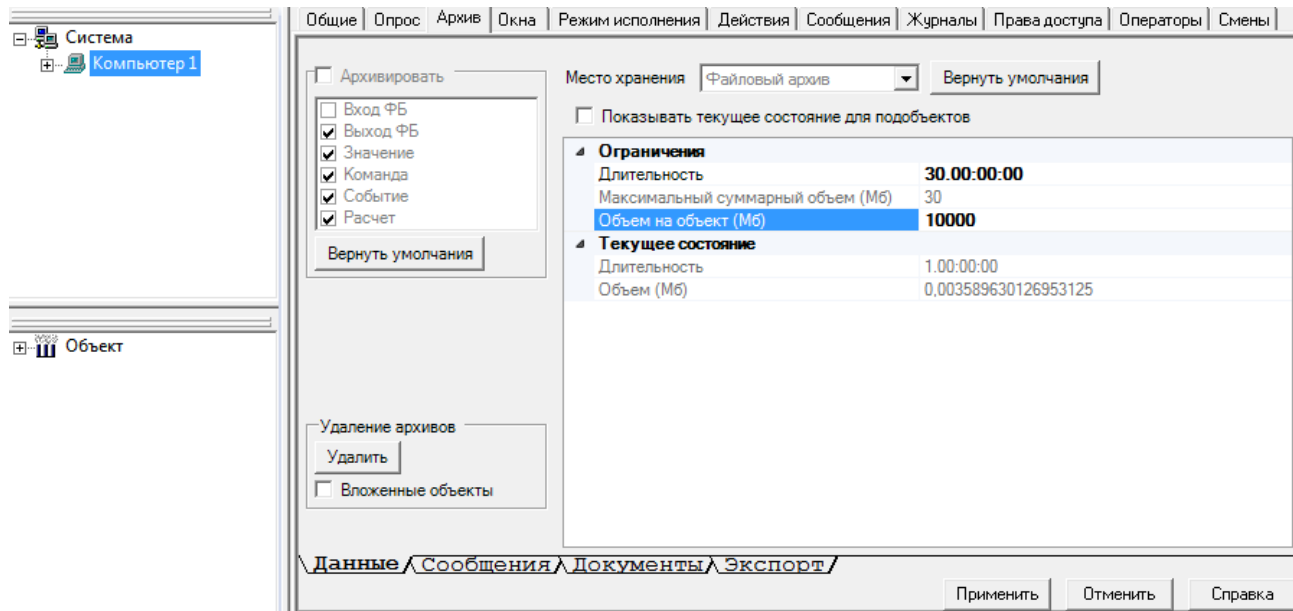


Рисунок 4-35. Настройка архивации компьютера

Подробнее система архивации описана в специальном разделе данной документации.

4.9. Создание изображения объекта

Емкостей у нас в системе будет 3. На стартовой мнемосхеме должны располагаться небольшие контролы, которые бы выводили общую информацию об объекте – состояние уровня и температуры.

Для решения данной задачи используется изображение объекта. Изображение объекта можно размещать на другой мнемосхеме как встроенный активный символ. Здесь можно провести аналогию с «живыми плитками» в Windows Phone, Windows 8 и Windows 10 – то есть элемент выполняет сразу несколько функций: отображает состояние объекта и позволяет открывать документ этого объекта.

Для создания изображения выделим объект емкости и перейдем на вкладку **Окна**. Выделим изображение объекта и нажмем кнопку **Создать**.

Создание проекта с использованием объектного подхода

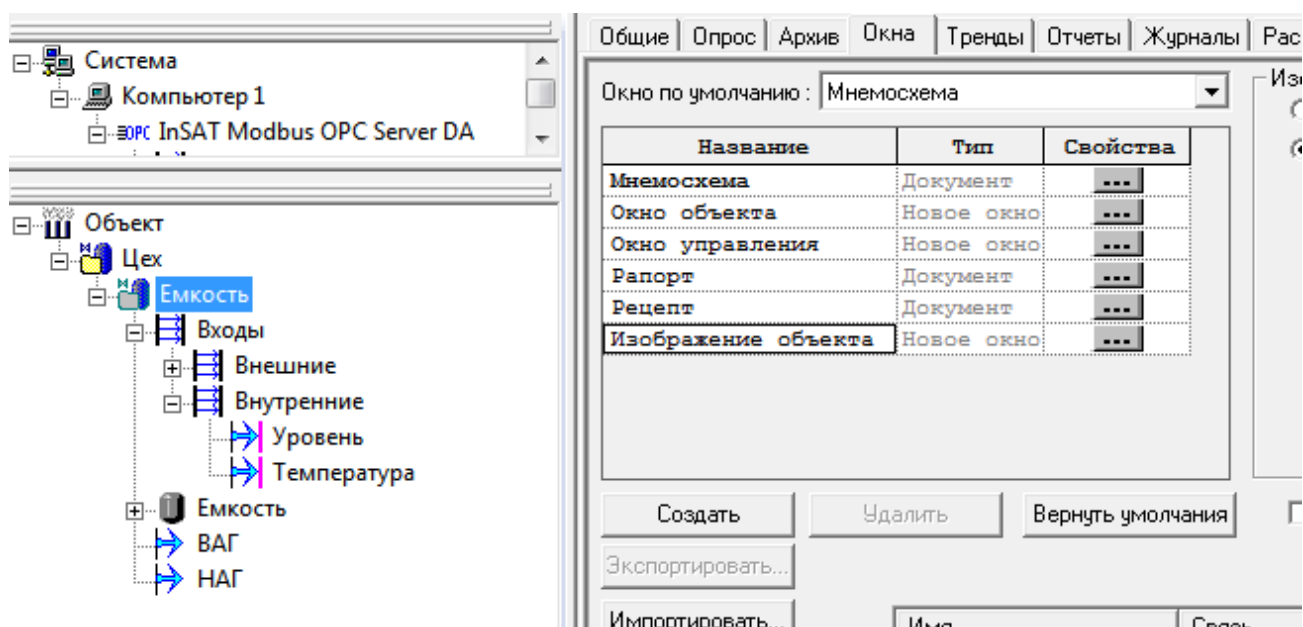


Рисунок 4-36. Создание изображения объекта

Перетащим в редактор мнемосхем **Емкость**, а также переменные уровня и температуры из группы **Входы – Внутренние**.

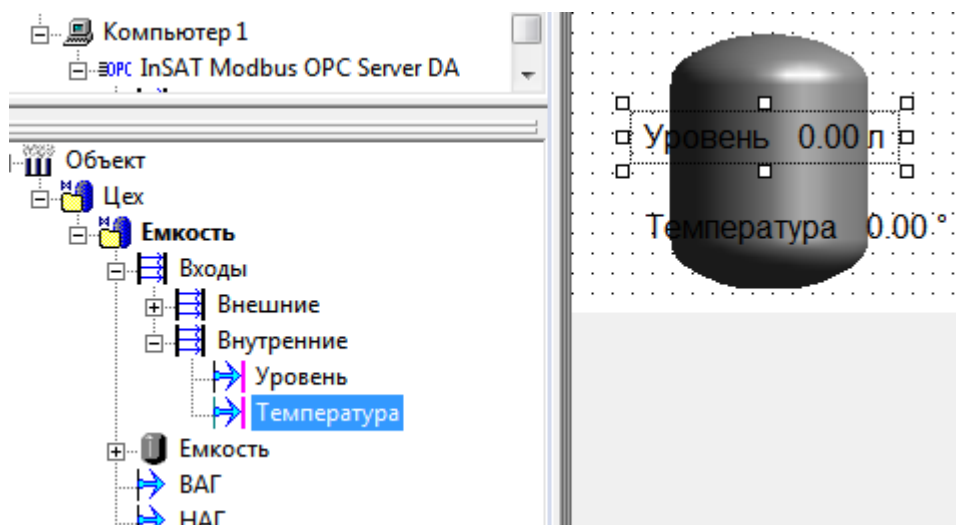


Рисунок 4-37. Создание контролов на изображении

Усечем названия текстовых полей – вызовем контекстное меню и вызовем окно свойств. На вкладке **Текст** поставим текст до значения: **L=**.

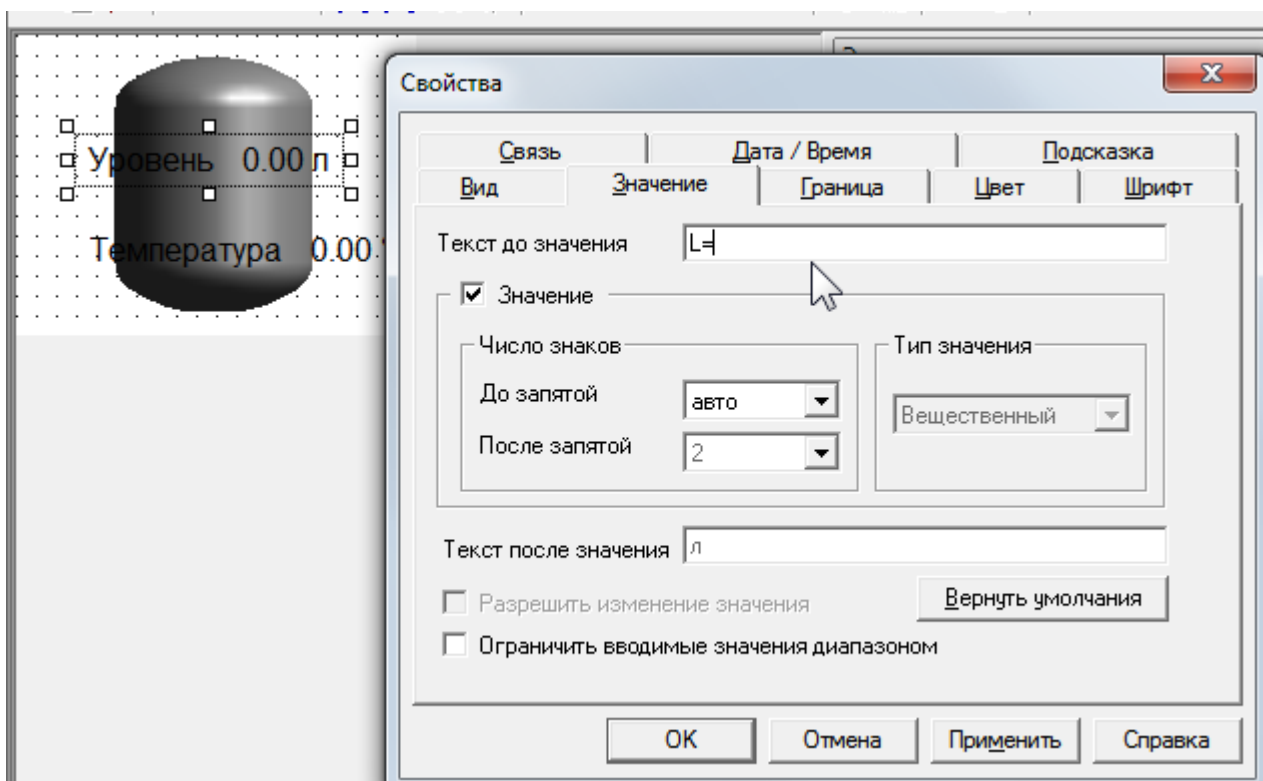


Рисунок 4-38. Создание текста до значения

Аналогично сделаем для температуры – укажем текст $T=$.

Разместим элементы правее емкости.

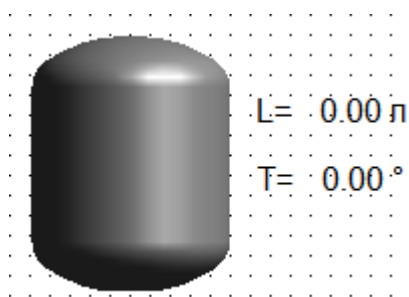


Рисунок 4-39. Расстановка элементов на изображении

Сохраним изображение объекта – теперь его можно использовать на стартовой мнемосхеме. Мы сделаем это, когда выполним тиражирование объекта.

Примечание. Подробнее изображение объекта описывается в разделе «Редактор мнемосхем MasterGraf».

4.10. Тиражирование объектов

Теперь, когда объект первой емкости полностью закончен, можно приступить к тиражированию. Но, перед тем как выполнять копирование объекта, включим галочку **Шаблон** – чтобы иметь возможность вносить правки в копии данного объекта.

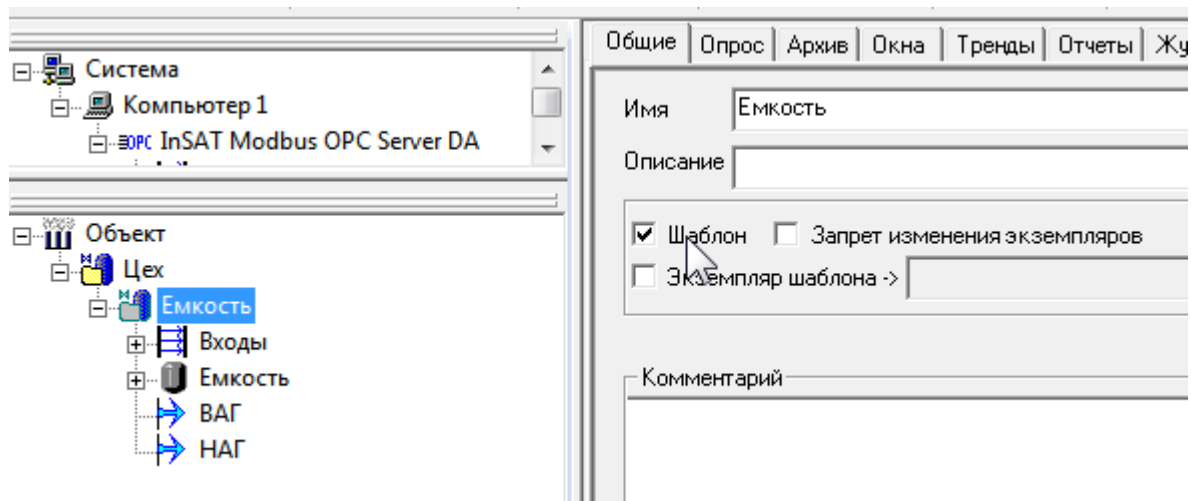


Рисунок 4-40. Установка объекта как шаблона

Теперь вызовем у объекта контекстное меню и выберем пункт **Дублировать**.

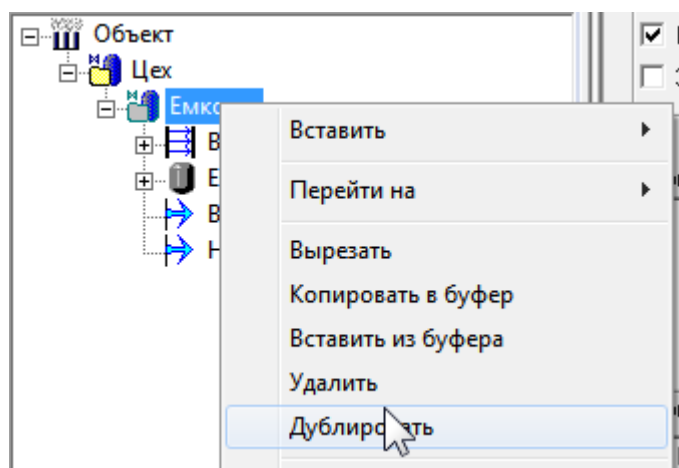


Рисунок 4-41. Дублирование объекта

Появится окно **Восстановление внешних связей**. Поскольку наш объект имеет связи с OPC-переменными, MasterSCADA хочет узнать, как мы хотим поступить с этими связями: оставить их теми же или заменить их. Нам необходимо изменить их – сделать привязку к другой группе OPC-переменных.

Создание проекта с использованием объектного подхода

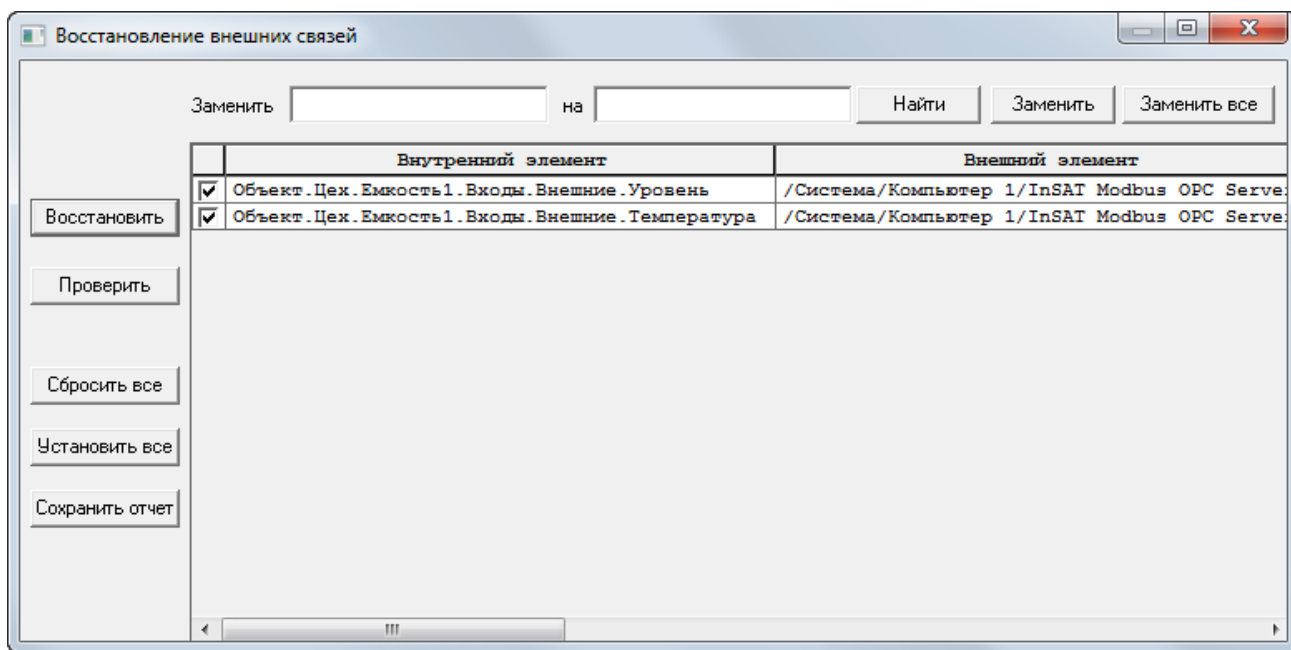


Рисунок 4-42. Окно восстановления внешних связей

Для ускорения привязки воспользуемся поиском и заменой имен – заменим **Емкость 1** на **Емкость 2**. Затем нажмем на кнопку **Проверить** – так мы проверяем, что установка связей возможна.

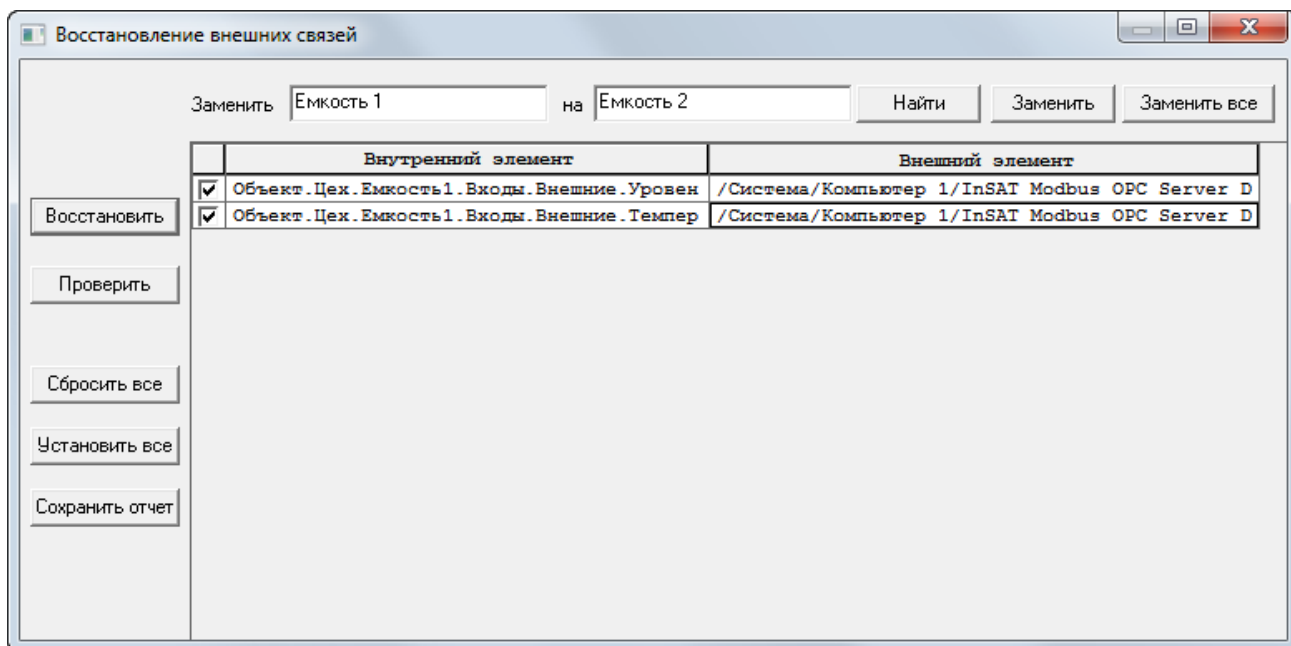


Рисунок 4-43. Замена внешних связей

Теперь нажимаем на кнопку **Восстановить** – эта команда устанавливает связи.

Создание проекта с использованием объектного подхода

Создался новый объект с установленными связями. Аналогичным образом создаем третью емкость.

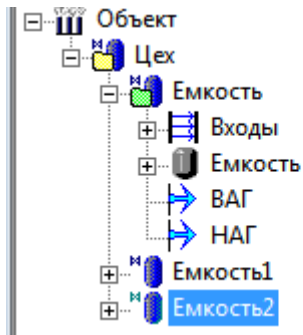


Рисунок 4-44. Создание третьей емкости

Теперь создадим у объекта первого уровня мнемосхему и добавим на нее изображения объектов наших емкостей. Для этого перетаскиваем объект правой кнопкой мыши. В появляющемся окне выбираем **Изображение объекта**.

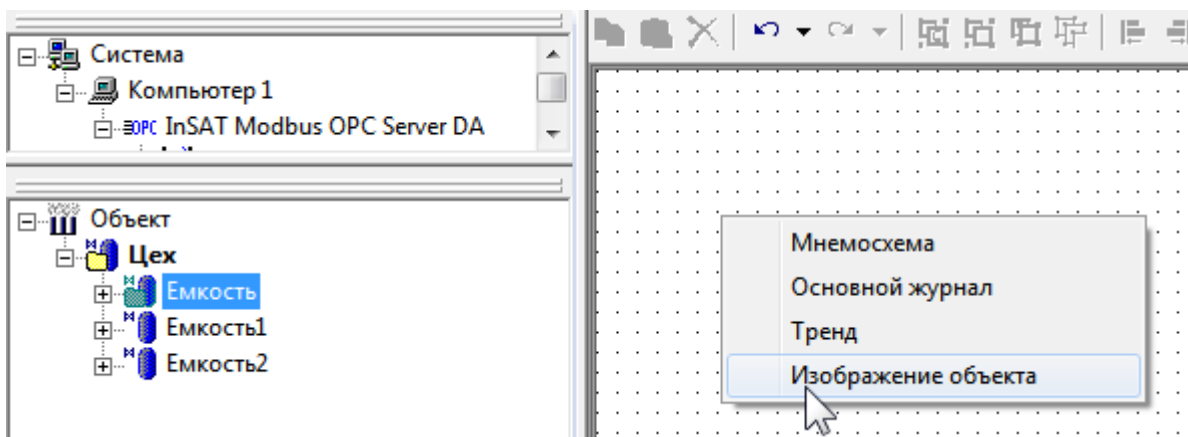


Рисунок 4-45. Добавление изображения на мнемосхему

На мнемосхему добавится наше изображение объекта (Рисунок 4-46).

Создание проекта с использованием объектного подхода

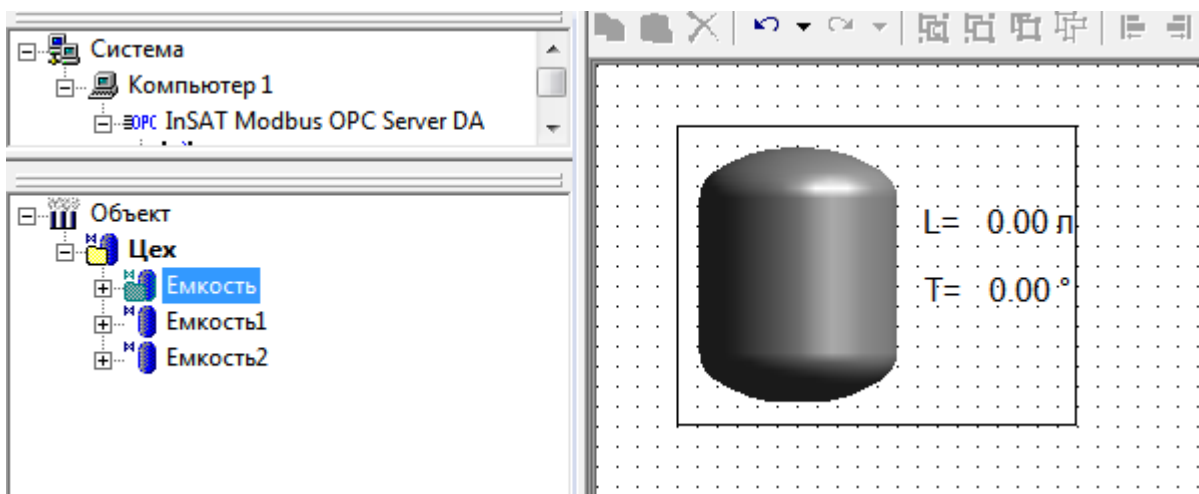


Рисунок 4-46. Мнемосхема с изображением объекта

Аналогично сделаем для остальных емкостей – мнемосхема должна получиться, как на 4-47.

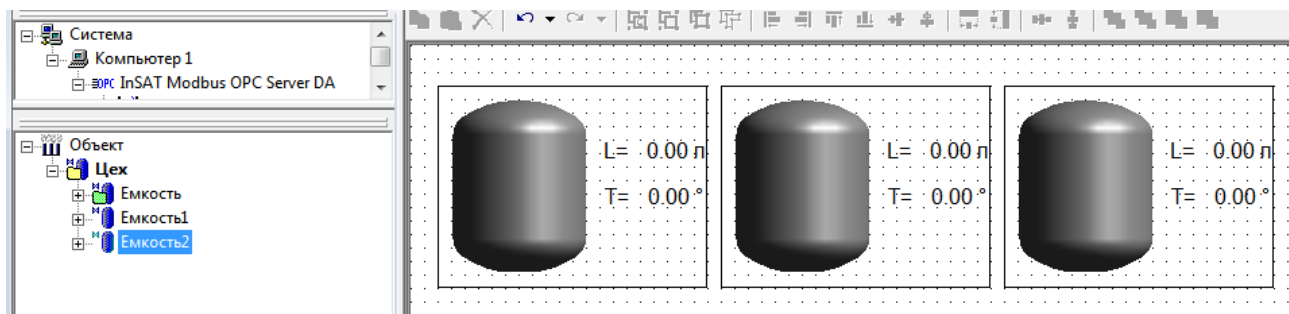


Рисунок 4-47. Мнемосхема с изображениями всех объектов

Наша мнемосхема готова.

Сделаем так, чтобы данная мнемосхема была стартовой – сразу загружалась при старте режима исполнения. Для этого на вкладке **Окна** поставим галочку **Стартовая мнемосхема компьютера**.

Создание проекта с использованием объектного подхода

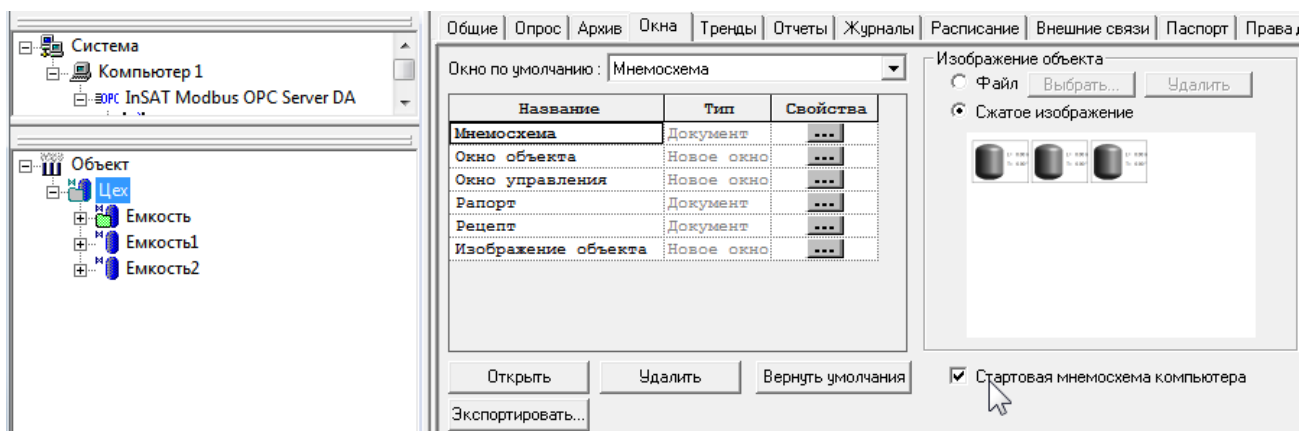


Рисунок 4-48. Установка мнемосхемы как стартовой

4.11. Проверка работы в режиме исполнения

Запустим проект в режим исполнения – загрузилась стартовая мнемосхема, в изображениях объекта мы видим состояние емкости.



Рисунок 4-49. Состояние мнемосхемы в режиме исполнения

При щелчке по емкости мы попадаем на детальную мнемосхему данной емкости (Рисунок 4-50).



ВАГ 130.00

НАГ 20.00

Журнал

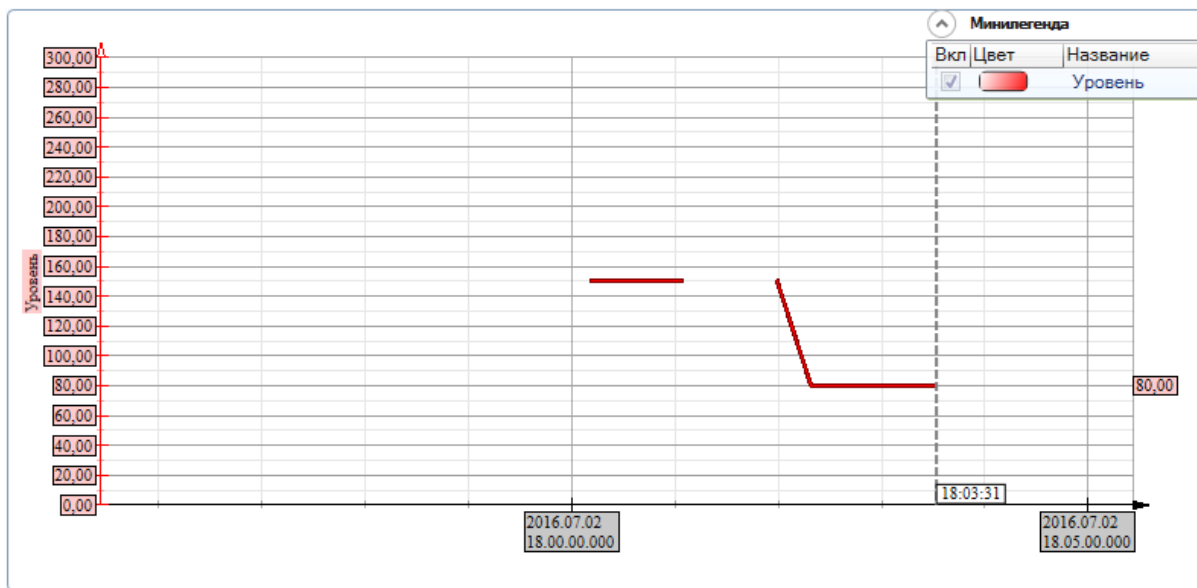




Рисунок 4-50. Детальная мнемосхема емкости

На ней мы видим состояние объекта, имеем доступ к архивным данным и можем выполнять изменение аварийных границ – все, что и требовалось по нашему заданию.

4.12. Тестирование и отладка проекта

В процессе работы часто возникает потребность посмотреть разработанный проект – увидеть дерево объектов, дерево системы, вкладки. Можно активировать видимость дерева системы и дерева объектов, оставив при этом видимость документов и окон – для этого предназначены кнопки панели инструментов . Можно полностью открыть проект, скрыв при этом документы и окна (в свойствах которых включена настройка **Закрывать при показе проекта**) – для этого предназначена кнопка **Показать проект** ().

Создание проекта с использованием объектного подхода

Вид менеджера проектов в режиме исполнения (Рисунок 4-51) отличается от режима разработки наличием дополнительного столбца справа от деревьев.

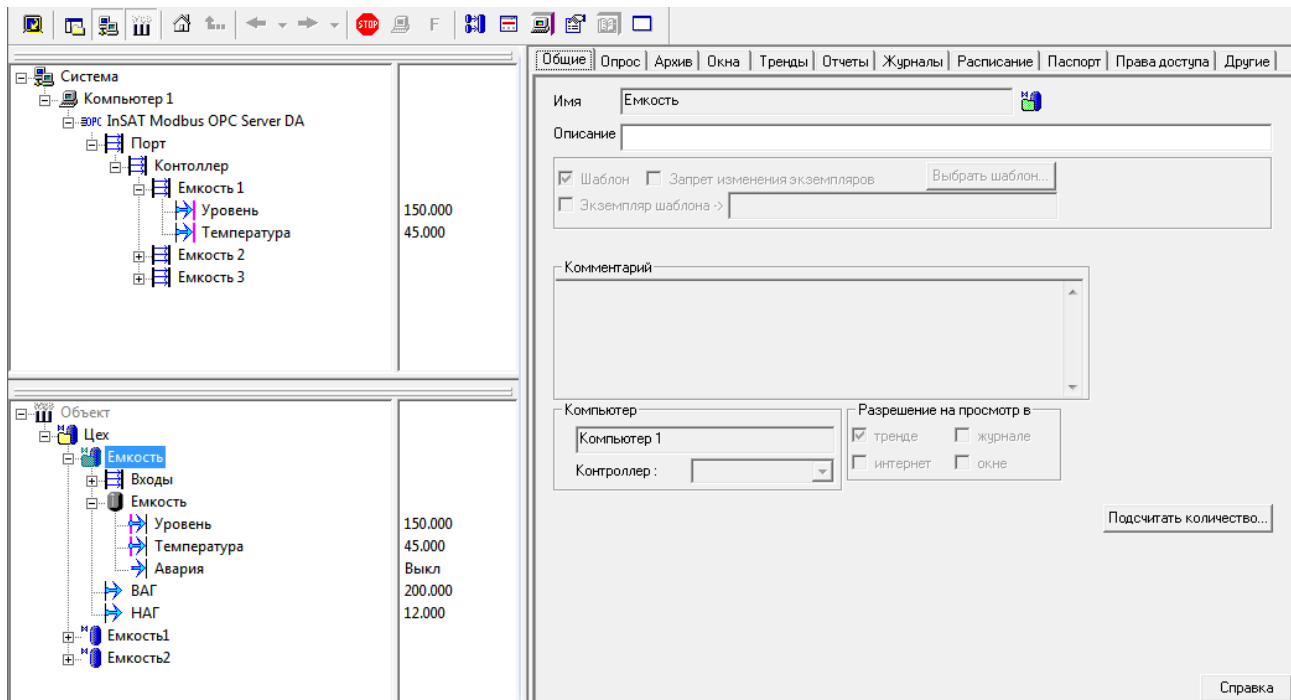


Рисунок 4-51. Менеджер проекта в режиме исполнения

В этом столбце отображаются значения переменных. В этом же поле можно произвести ввод значения – посмотреть, как будет вести себя проект при определенных значениях. Для этого достаточно щелкнуть мышью правее нужной переменной – появится окно ввода.

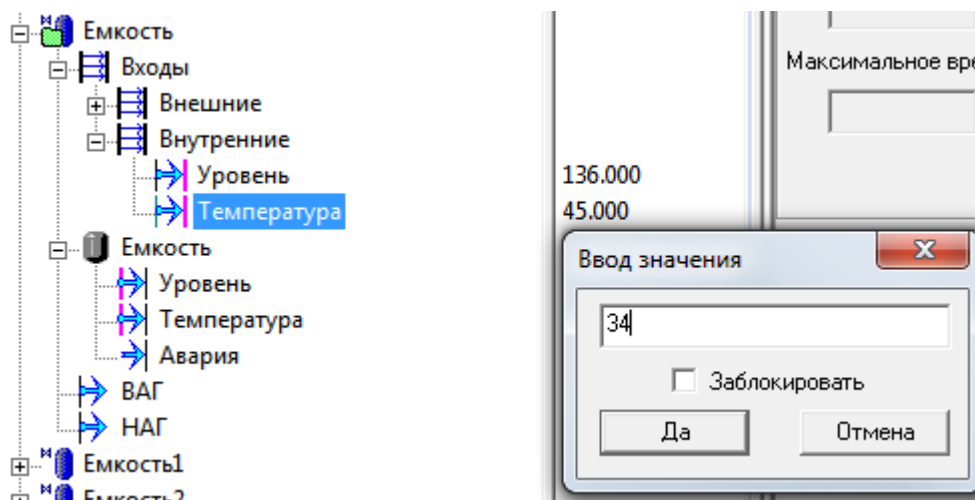


Рисунок 4-52. Ввод значения в режиме исполнения

Многие пользователи пытаются оценивать быстродействие работы системы по показаниям дерева объектов, и это неизбежно ведет к ошибочным выводам. Информация в данном столбце меняется в соответствии с собственным циклом, поэтому не позволяет судить ни о скорости изменения переменной, ни о не последнем значении. Достоверную информацию можно получить на странице свойств переменной **Данные выхода** – для этого на странице находятся три поля: **Среднее время**, **Минимальное время** и **Максимальное время**.

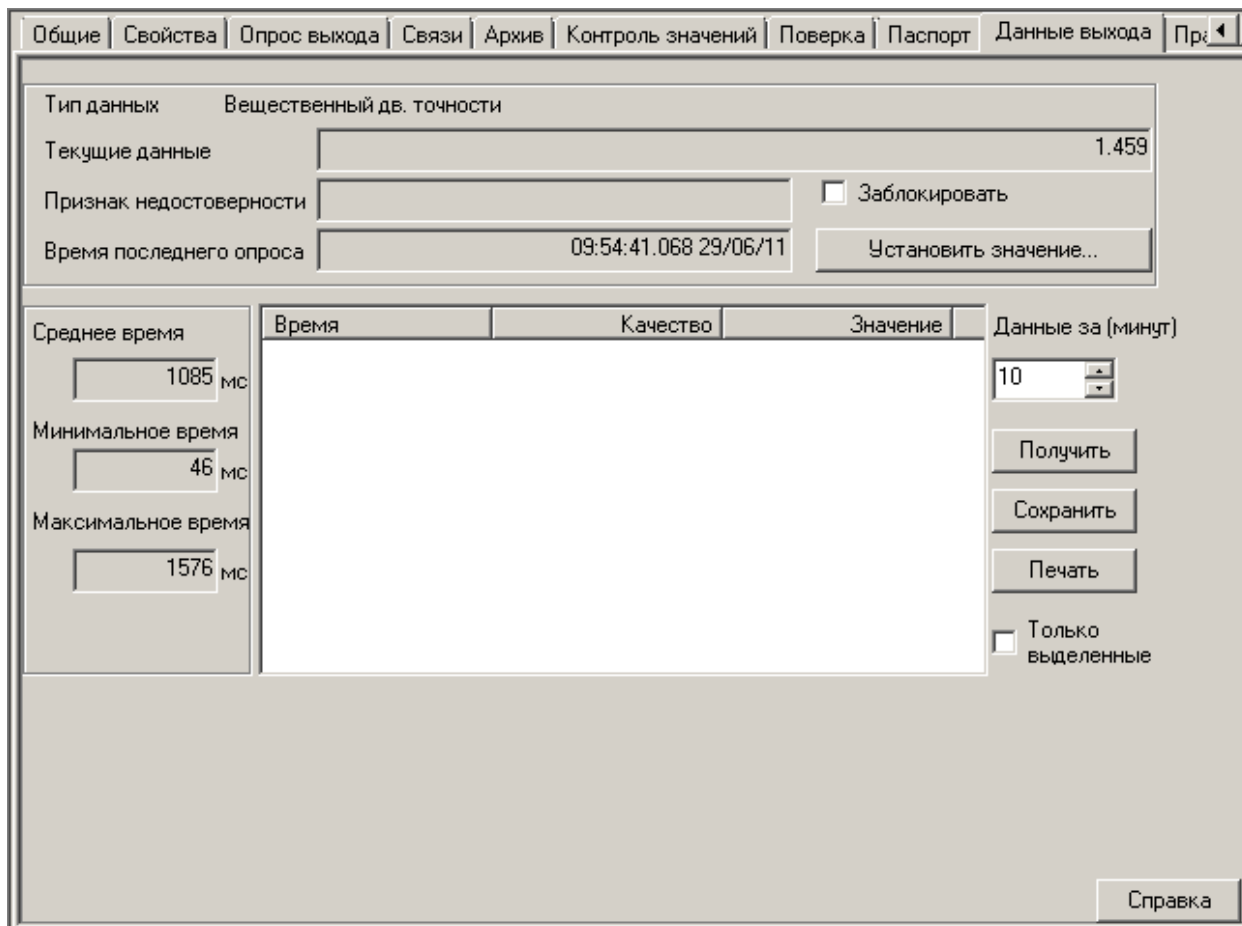


Рисунок 4-53. Страница свойств «Данные выхода» в режиме исполнения

На этапе тестирования проекта разработчик должен определить оптимальный период опроса OPC-сервера. Когда мы только приступаем к разработке проекта, мы устанавливаем желаемый период опроса, то есть тот, который прописан в техническом задании. Заранее сказать, будет ли этот период опроса допустим на реальном объекте, нельзя. Свою роль могут сыграть быстродействие оборудования нижнего уровня, пропускная способность

каналов связи, объем опрашиваемых данных, быстродействие компьютера и другие факторы.

Нужно понимать, что MasterSCADA в большинстве случаев не влияет на скорость, с которой OPC-сервер опрашивает приборы. Это зависит от реализации и настроек самого OPC-сервера, а также от характеристик контроллера и линии связи. Наиболее важно определить, как OPC-сервер взаимодействует с оборудованием. Именно этот период опроса является отправной точкой в настройке MasterSCADA. Самый оптимальный вариант – чтобы период, с которым OPC-сервер опрашивает устройства, совпадал с периодом опроса, с которым MasterSCADA опрашивает сервер. Если период опроса оборудования будет выше, чем период опроса сервера клиентом, то тогда часть данных будет пропадать (за исключением ситуации, когда сервер поддерживает стандарт HDA и передает в MasterSCADA данные в виде архива).

Также стоит обратить внимание на поле **Признак достоверности**. Если данное поле пустое, как показано **на Ошибка! Источник ссылки не найден.**, то это означает, что от OPC-сервера приходят данные без каких-либо признаков достоверности, передаваемых стандартными признаками качества. В дереве недостоверные переменные отображены серым шрифтом. Если мы видим, что имеется признак достоверности, важно понять причину его появления.

MasterSCADA выставляет признак качества OPC-переменным лишь в том случае, если связь с OPC-сервером не обнаружена. Во всех остальных случаях признак качества переменных приходит непосредственно от OPC-сервера и проблему необходимо искать именно на его уровне (причиной может быть обрыв связи с источником данных, нарушение допустимого диапазона значений и т. п. – для каждого типа OPC-сервера описание алгоритма формирования признаков качества может отличаться). Сигналом о том, что связь с OPC-сервером не обнаружена, служит красная звездочка рядом с обозначением OPC-сервера в дереве системы. Отсутствие маркера говорит о наличии подключения.

4.13. Создание операторов и разграничение прав доступа.

По умолчанию в проекте присутствует оператор **sa** (System Administrator), он имеет категорию **Разработчик** – высший уровень доступа к параметрам. Как правило, операторам

Создание проекта с использованием объектного подхода

нежелательно давать такие полномочия, поэтому по окончании разработки и отладки проекта можно произвести настройку прав доступа для каждой категории операторов.

MasterSCADA имеет гибкие средства разграничения прав доступа. Первое с чего нужно начать – это создать пользователя с нужной должностью. Для этого нужно выделить **Компьютер** и перейти на вкладку **Операторы и безопасность**. Выделим нужную нам должность, например, **Оператор**, и нажмем кнопку **Добавить**. Создастся новый оператор (*Рисунок 4-54*) – можно изменить ему имя и задать пароль.

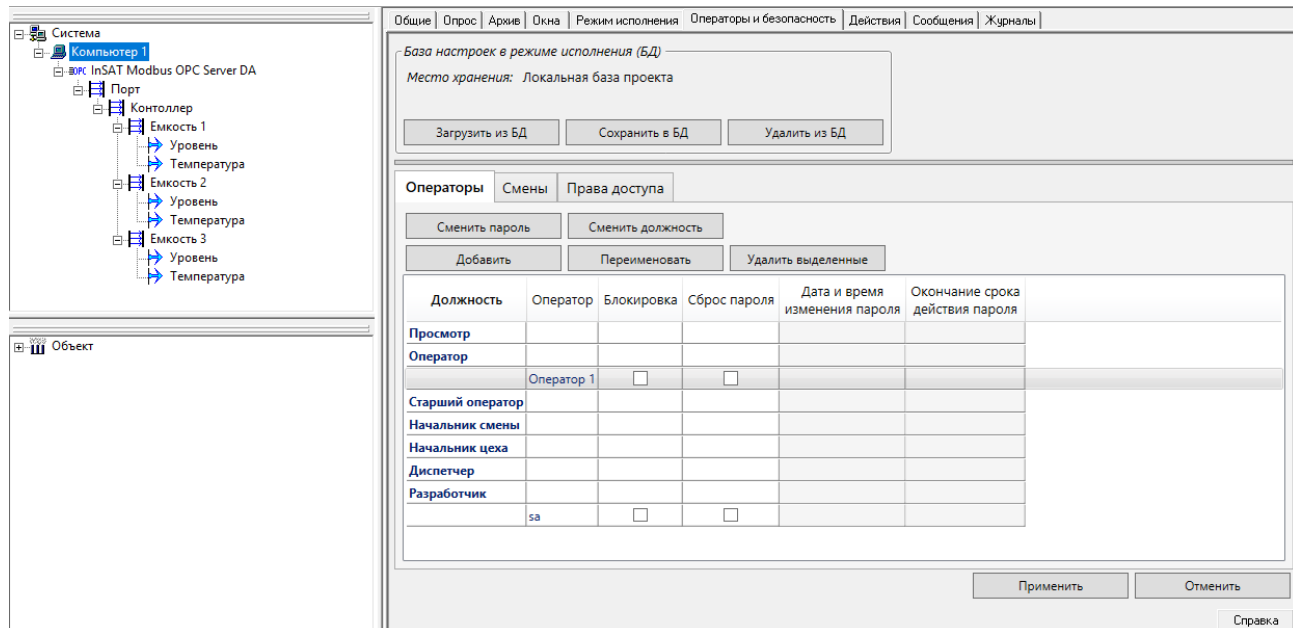


Рисунок 4-54. Добавление оператора

Теперь назначим его в смену: в какое время оператор может входить в систему. Это делается на вкладке **Смены**.

Создание проекта с использованием объектного подхода

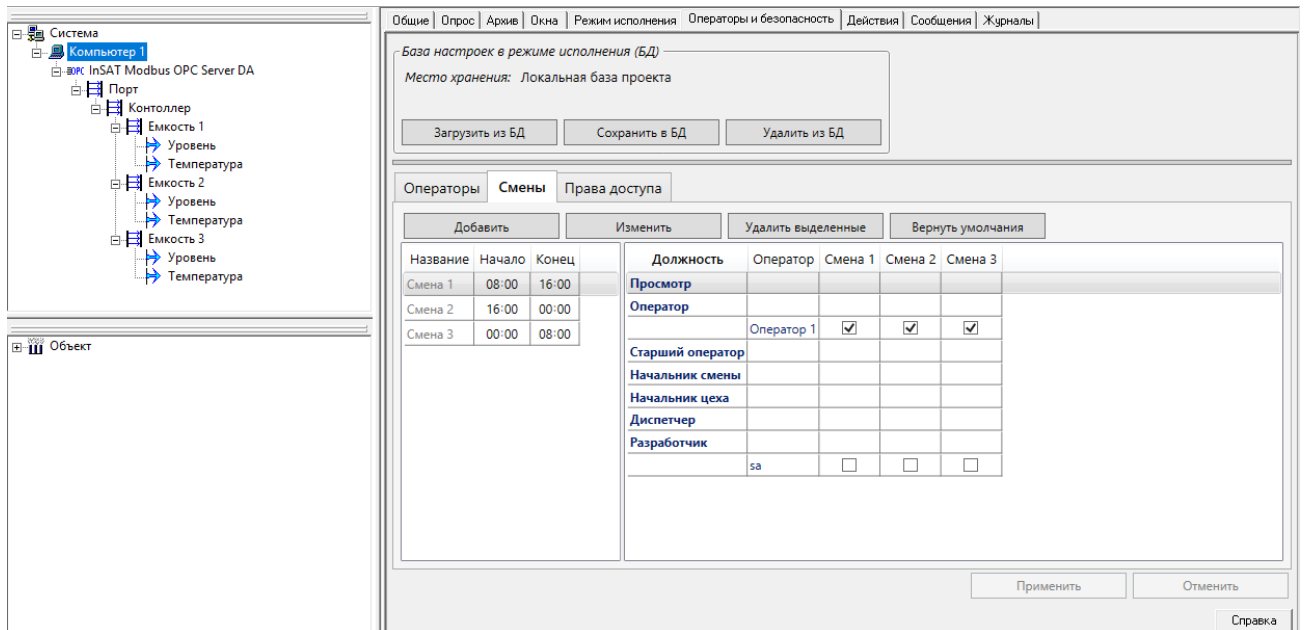


Рисунок 4-55. Назначение оператора в смену

Примечание. При необходимости можно отключить сменность у должности. Это делается на вкладке «Системы» – «Операторы и безопасность» – «Должности».

Теперь для данной должности можно установить права доступа: разрешить или запретить те или иные действия. Настройка прав доступа наследуется в MasterSCADA сверху вниз – от системы к компьютерам, к объектам и к переменным. При этом на любом уровне права могут перенастроены (переопределены).

Выделим элемент **Система**, перейдем на вкладку **Операторы и безопасность** – **Права доступа**. Выберем в списке должностей **Операторы**.

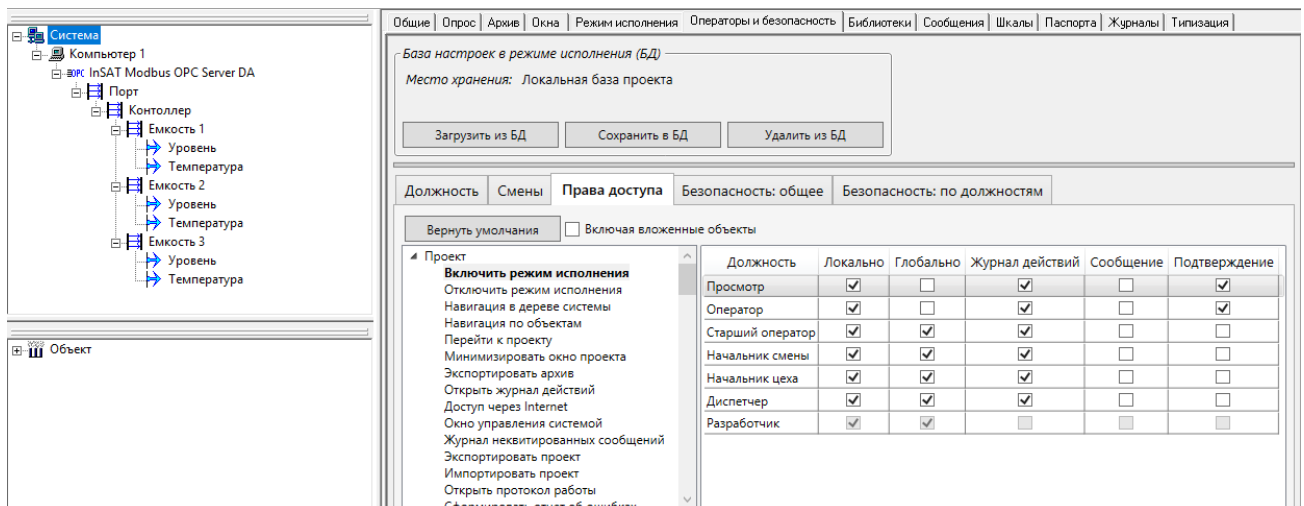


Рисунок 4-56. Настройка прав доступа

Создание проекта с использованием объектного подхода

На данной вкладке присутствуют все доступные действия в системе. Каждое из действий можно отключить оператору, сняв галочку **Лок.**; можно установить подтверждение действия, установив галочку **Подтв.**; а также включить запись действия в **журнал**, установив галочку **Журн.** Галочка **Глоб.** аналогична **Лок.**, но предназначена для сетевых проектов – в случае, когда оператор пытается выполнить действие на объекте, исполняемом не на его компьютере.

Используя данную вкладку, можно гибко настроить права каждого оператора. Например, можно запретить оператору ввод **уставок** – для этого нужно в разделе **Переменная** снять флаг **Установить значение**.

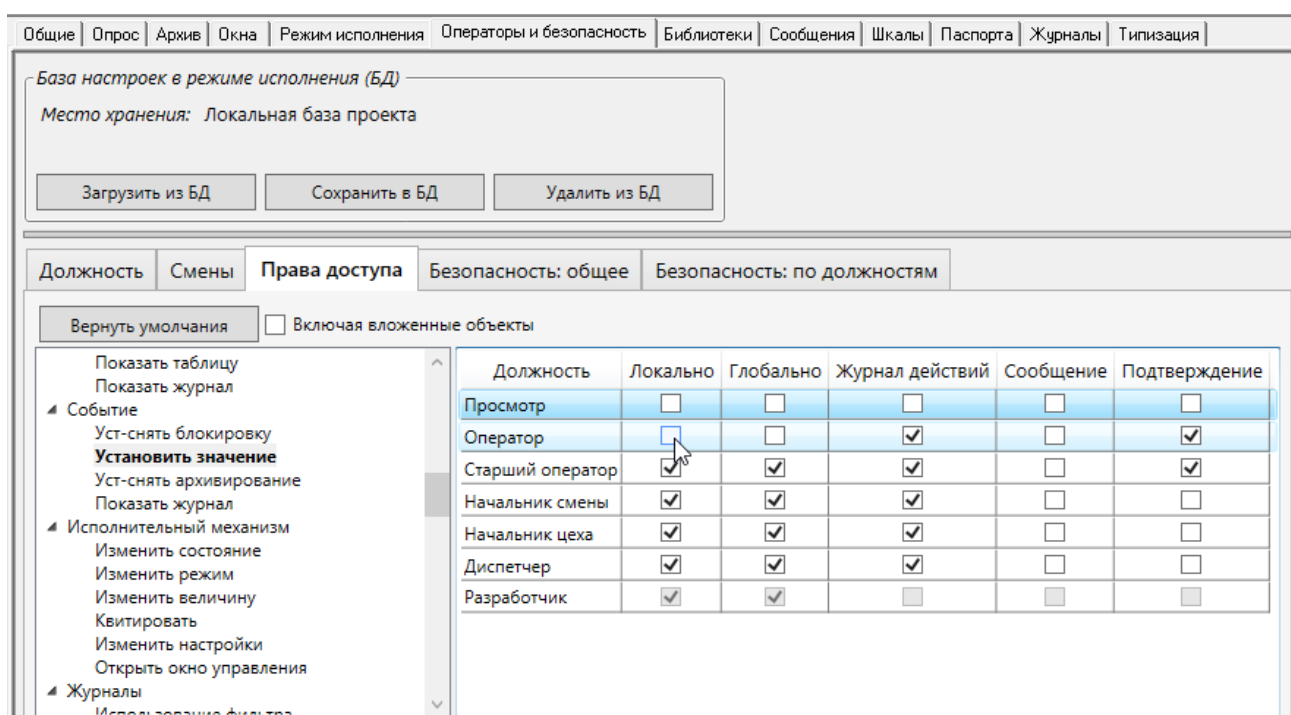


Рисунок 4-57. Отключение оператору действия «Установить значение»

Теперь оператор не сможет изменить аварийные границы – ему будет выведено сообщение о недостаточности прав.

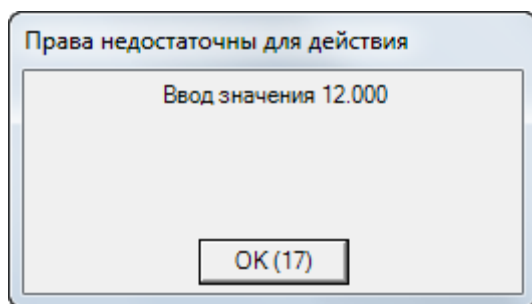


Рисунок 4-58. Окно предупреждения

Проект готов. Его можно переносить на рабочий компьютер оператора.

4.14. Проверка совместимости версий MasterSCADA

Прежде чем переносить проект с компьютера разработчика на компьютер оператора, необходимо убедиться в том, что на новом компьютере установлена совместимая с проектом версия MasterSCADA.

Демонстрационная версия, которую, как правило, используют разработчики, включает в себя все возможные модули, в коммерческой версии к базовому комплекту добавлены только оплаченные опции. Если проект уже создан, то состав необходимой коммерческой версии можно определить с помощью встроенного инструментария. Перейдите на страницу свойств компьютера **Общие** и нажмите на кнопку **Сформировать опции для заказа**. Появившееся диалоговое окно покажет требуемый для данного проекта состав MasterSCADA.

В обсуждаемом нами учебном примере для обеспечения бесперебойной работы проекта можно воспользоваться бесплатной версией **MSRT32**, которую вы можете скачать с сайта. Перед переносом проекта на другой компьютер обратите внимание на номер версии (номер сборки) MasterSCADA. Его можно увидеть в окне, выполнив пункт главного меню **Справка – О программе**. Нужная информация отобразится в поле **Версия**. Номера версий либо должны совпадать, либо номер версии на новом компьютере должен быть выше, чем номер версии на компьютере разработчика.

4.15. Перенос проекта с одного компьютера на другой

Проект MasterSCADA – это директория, в которой хранится сам файл проекта ([название проекта].vav), а также набор дополнительных файлов и папок. Состав проекта зависит от конфигурации проекта. Важно помнить, что при переносе проекта необходимо не

нарушить файловую структуру. При этом на новом месте запуска проекта нам обычно не нужны архивы, накопившиеся на старом месте и расположенные в папках своих объектов проекта. Для корректного переноса проекта надо выполнить пункт главного меню программы **Проект – Экспортировать**. В диалоговом окне укажите место для экспорта. Проект сохранится по указанному адресу. После этого сохраненную папку можно копировать на другой компьютер. При экспорте формируется проект со всеми необходимыми ресурсами и без архивов. Это позволяет копировать (или импортировать) поверх рабочего проекта, не беспокоясь о сохранности накопленных архивных данных.

Перед запуском проекта необходимо убедиться в наличии используемого в нем OPC-сервера с необходимыми компонентами (при отсутствии – установить его), а также скопировать на другой компьютер конфигурацию OPC-сервера, сформированную для работы с данным проектом.

После установки OPC-сервера нужно запустить конфигурацию и установить флаг, который позволит использовать ее по умолчанию.

Дополнительные компоненты: ActiveX-элементы, собственные нестандартные ФБ и другие элементы – мы в данном проекте не использовали, соответственно, на этом процесс переноса проекта на другой компьютер завершен.

Перед тем как запустить проект на исполнение, откройте все документы в режиме разработки и убедитесь, что их внешний вид не претерпел изменений.

Если вы отредактировали проект на собственном компьютере и хотите внести правки в работающий проект, нужно на компьютере, на котором вносилась правка, выполнить **Проект – Экспортировать**. Экспортированный проект не содержит архивов, и его можно переносить на другой компьютер.

Внести проект на новый компьютер можно как в режиме исполнения, так и в режиме разработки. В режиме разработки это можно сделать двумя способами:

1. **Проект – Импортировать**: выбираете **Папка с импортируемым проектом** и папку, в которую вы экспортировали проект.
2. Просто скопировать новый проект поверх старого с заменой. Архивы при этом затронуты не будут.

Создание проекта с использованием объектного подхода

В режиме исполнения нажмите **Alt+F10 – Показать проект**. Выделите **Компьютер** – закладка **Режим исполнения – Управление проектом**.

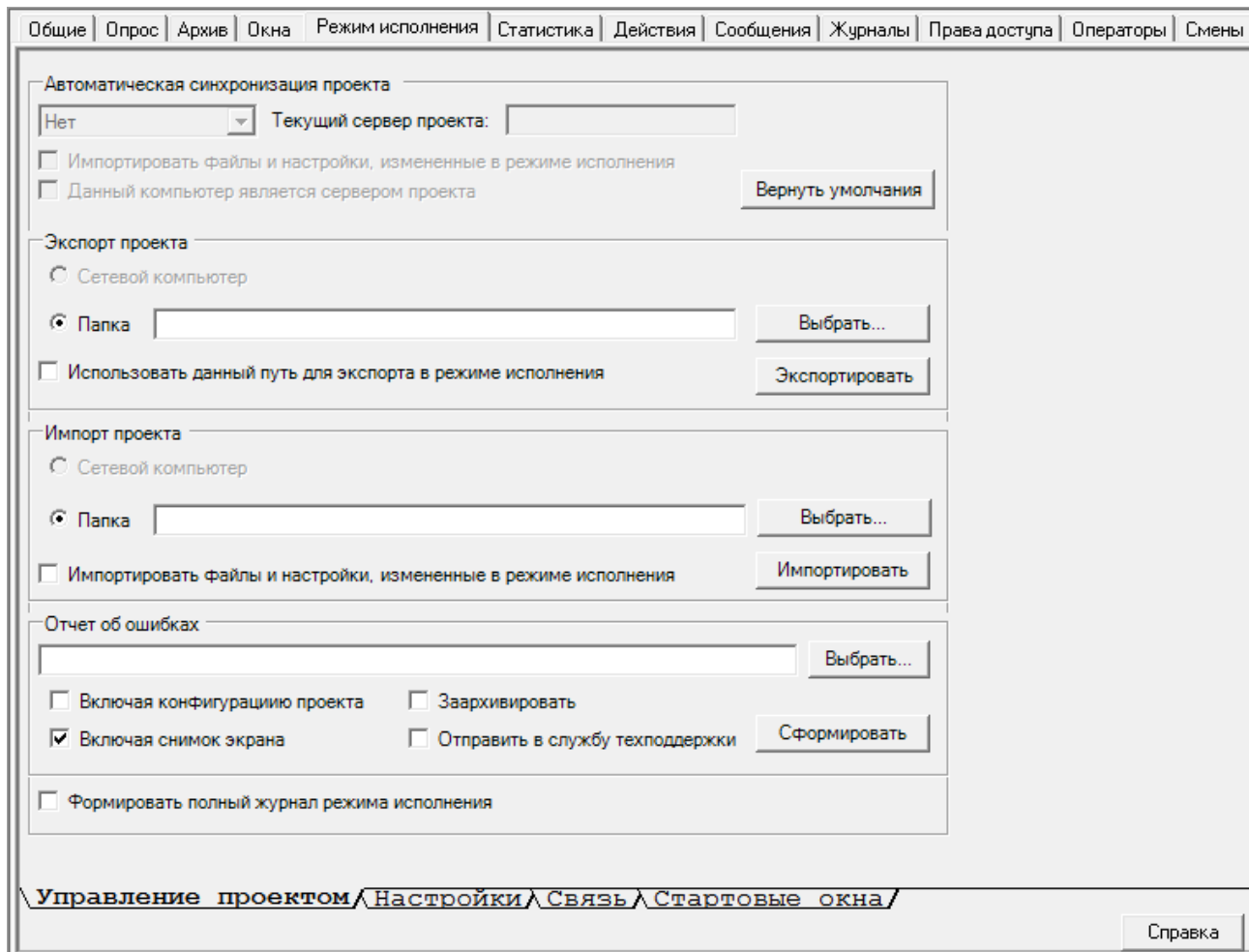


Рисунок 4-59. Вкладка управления проектом

Выберите импорт из папки, а также папку, в которую вы экспортировали проект, и нажмите **Импортировать**. Режим исполнения временно остановится, затем автоматически запустится уже с обновленным проектом.

4.16. Дополнительные материалы

Мы рассмотрели общие принципы создания проектов. На прилагаемом диске содержится цикл видеопримеров, в которых наглядно рассматривается создание проекта. Рекомендуем вам ознакомиться с ними.

5. Работа с оборудованием

Рассмотрим основные способы работы MasterSCADA с оборудованием. Обратим внимание на основные инструменты, предоставляемые разработчику.

5.1. Общие принципы взаимодействия с оборудованием

SCADA-система является промежуточным звеном между оператором и системой управления. Поэтому, помимо взаимодействия с оператором через мнемосхемы, тренды и журналы, SCADA-система должна вести обмен информацией с нижним уровнем: контроллерами, модулями, приборами. Обмен может осуществляться по стандартным (**Modbus**, **DCON**) или по закрытым протоколам (собственные разработки фирм – производителей контроллеров).

Существует два способа получения данных SCADA-системой от нижнего уровня: драйверы и OPC-серверы.

Драйверы: в этом случае SCADA-система получает данные через собственный уникальный драйвер (Рисунок 5-1).



Рисунок 5-1. Функциональная схема связи через драйвер

Для каждого протокола необходимо написать (разработчику SCADA или прибора) специальный драйвер. При этом драйвер будет работать только с данной SCADA-системой.

Поэтому разработчику системы автоматизации нужно убедиться, что SCADA поддерживает выбранные им приборы.

В случае использования драйверов могут также возникнуть проблемы, если необходимо иметь данные от нижнего уровня в нескольких программах (например, в двух SCADA-системах на разных компьютерах).

В итоге такой подход усложняет работу разработчиков систем автоматизации, так как им приходится подстраиваться и под разработчиков SCADA-систем, и под разработчиков приборов автоматизации.

В MasterSCADA имеется два способа подключения драйверов: напрямую (используется только для встроенных в компьютер плат ввода/вывода) и через модуль подключения **MasterLink**. **MasterLink** является специальной версией контроллерной исполнительской системы **MasterPLC**, входящей в состав MasterSCADA. Благодаря этому один и тот же драйвер может быть использован для подключения оборудования и к верхнему уровню системы (компьютерам), и к нижнему уровню (контроллерам).

Кроме того, для обмена небольшим количеством данных по простым последовательным протоколам в MasterSCADA имеется функциональный блок **SmartLink**, позволяющий описать форматы кадров запросов и ответов вообще без программирования.

Наиболее часто MasterSCADA взаимодействует с приборами нижнего уровня посредством OPC-сервера. Термин «OPC» обозначает семейство стандартов, описывающих технологии взаимодействия между собой разработанных для области промышленной автоматизации программных средств различных производителей. Как правило, речь идет о стандартизации опроса технических средств нижнего уровня (контроллеров, интеллектуальных приборов и т. п.), то есть об унифицированных драйверах, с помощью которых программные средства верхнего уровня систем автоматизации (прежде всего SCADA-систем) могут опрашивать оборудование. Основное преимущество OPC – это унификация работы с источниками данных. Благодаря стандарту любой OPC-сервер может работать с любой SCADA-системой. Что в итоге позволяет подключить любой прибор к любой SCADA – достаточно установить OPC-сервер для данного протокола. При этом написать OPC-сервер может фирма – производитель приборов, может разработчик SCADA-систем (при этом OPC-сервер будет также работать и со SCADA других фирм), а может и сторонняя фирма.

5.2. Использование OPC-серверов

В случае использования OPC-серверов схема получения данных выглядит так. OPC-сервер при помощи встроенного драйвера получает данные с нижнего уровня, анализирует их, устанавливает признак качества и метку времени. Передает данные по интерфейсу OPC-клиенту. OPC-клиентом является SCADA-система (или другая программа, например MES-система) – 5-2.



Рисунок 5-2. Функциональная связь через OPC-сервер

Стандарт обмена данными OPC базируется на схеме «клиент-сервер». Это позволяет подключать множество клиентов к одному серверу. И наоборот, данный стандарт позволяет использовать с помощью одного клиента несколько OPC-серверов (Рисунок 5-3).

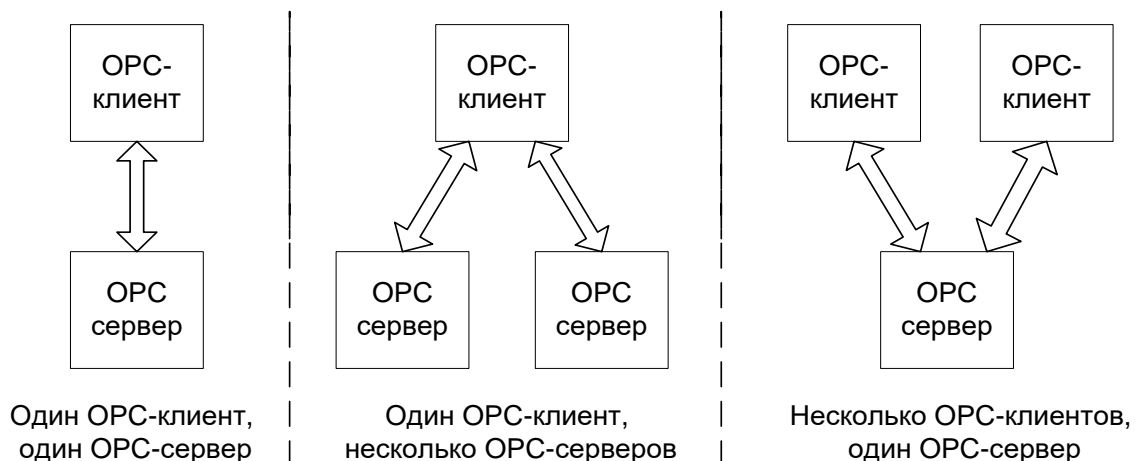


Рисунок 5-3. Клиент-серверная архитектура OPC

Примечание. OPC-сервер может получать данные не только от приборов по промышленным протоколам. Через интерфейс OPC можно передавать практически

любые данные: реализовать обмен между программами (например, между SCADA и MES), получать данные из базы данных, выполняя SQL-запрос, опрашивать ключи реестра и т. д.

5.2.1. Стандарты OPC

OPC – набор спецификаций стандартов. Каждый стандарт описывает набор функций определенного назначения (более подробную информацию вы можете найти на официальном сайте организации OPC Foundation www.opcfoundation.org). Различают следующие стандарты:

- OPC DA (Data Access) — основной и наиболее востребованный стандарт. Описывает набор функций обмена для получения текущих данных с приборов, ПЛК и других устройств.
- OPC HDA (Historical Data Access) — в то время как OPC Data Access предоставляет доступ к текущим данным, OPC Historical Data Access предоставляет доступ к уже сохраненным данным (архивам).
- OPC AE (Alarms&Events) — предоставляет функции уведомления по требованию о различных событиях (аварийные ситуации, действия оператора, информационные сообщения и другие).
- OPC Batch — предоставляет функции шагового и рецептурного управления технологическим процессом.
- OPC DX (Data eXchange) — предоставляет функции организации обмена данными между OPC-серверами через сеть Ethernet. Основное назначение — создание шлюзов для обмена данными между устройствами и программами разных производителей.
- OPC Security — определяет функции организации прав доступа клиентов к данным системы управления через OPC-сервер.
- OPC XML-DA (XML-Data Access) — предоставляет гибкий, управляемый правилами формат обмена данными через SOAP и HTTP.

- OPC UA (Unified Architecture) — последняя по времени выпуска спецификация, которая основана не на технологии Microsoft DCOM, что предоставляет кроссплатформенную совместимость.

На данный момент наиболее распространены стандарты OPC DA (версия 2.05) и OPC HDA (версия 1.20). Оба стандарта поддерживаются в MasterSCADA – как в качестве клиента, так и в качестве сервера. Как серверный функционал, так и клиентский OPC HDA являются опциональными и поддерживаются дополнительными модулями.

Версия стандарта OPC DA 3.0 не поддерживается, так как она содержит функционально избыточные расширения и по этой причине, а также из-за неполной совместимости с версией 2 не получила распространения в промышленности.

Стандарт OPC AE используется редко, так как не предназначен для работы с оборудованием и в основном рассчитан на передачу подробной информации о событиях из одной программы верхнего уровня в другую. В MasterSCADA состав и структура сообщений соответствуют этому стандарту, однако клиент и сервер не были реализованы из-за отсутствия спроса на них. При необходимости они могут быть добавлены в состав системы.

В связи с тем, что перечисленные распространенные стандарты основаны на технологии Microsoft DCOM, реально эффективно работающей только под Windows, был разработан кроссплатформенный стандарт нового поколения – OPC UA, базирующийся на XML. Данный стандарт полностью снимает все проблемы «классического» OPC: отсутствие кроссплатформенности, сложность межкомпьютерного обмена, недостаточная безопасность, невозможность работы через Интернет. Кроме того, новый стандарт, учитывая кроссплатформенность, открывает перспективы использования прямо в контроллерах! Теперь, для того чтобы получить данные с ПЛК, не нужно ставить отдельные OPC-серверы, настраивать обмен и протоколы – достаточно подключить кабель к ПЛК, задать в SCADA-системе IP и порт, и все теги и данные будут считаны. В MasterSCADA 3.7 функционал OPC UA клиента реализован, и мы обязательно рассмотрим его в следующих главах.

5.2.2. OPC DA

Это самая распространенная версия стандарта OPC. OPC DA предназначен для взаимодействия с нижним уровнем систем в реальном времени.

OPC-сервер представляет собой отдельную программу, которая ведет обмен с источниками или приемниками данных по заданному для них протоколу. При получении данных драйвер анализирует полученное значение на достоверность по созданному разработчиком OPC-сервера алгоритму. Для того чтобы передать OPC-клиенту результат анализа, в стандарте предусмотрено специальное поле – признак качества. Признак качества представляет собой число из набора стандартных значений. Производитель OPC-сервера может поддерживать не все значения признака качества, а только некоторые из них.

Полученный признак качества пользователь SCADA-системы может использовать для реализации алгоритмов обработки ошибок – например, для выдачи сообщений оператору.

В MasterSCADA есть средства автоматической обработки признаков качества для формирования сообщений, средства извлечения признаков качества для пользовательской обработки (специальный ФБ **Признаки качества** и специальная функция для формул в расчетах и событиях), средства формирования признаков качества (для целей оформления результатов пользовательского анализа информации о достоверности переменных), а также правила передачи признаков качества в цепочке обработки данных. Перечень признаков качества и причины их возникновения описаны в таблице:

Признак	Код	Значение	Описание
<i>Opс_quality_bad</i>	0	Значение недостоверно	Значение недостоверно (признак качества – плохой)
<i>Opс_quality_config_error</i>	4	Ошибка конфигурации	Алгоритм имеет проблемы с конфигурацией – например, не все связи или настройки заданы
<i>Opс_quality_not_connected</i>	8	Нет соединения	Алгоритм имеет проблемы со связью – например, у входа алгоритма оборвана связь с выходом алгоритма – источника данных
<i>Opс_quality_device_failure</i>	12	Ошибка	Предписанные алгоритмом

		устройства	операции нельзя выполнить (например, деление на нуль или переполнение порядка)
<i>Opс_quality_sensor_failure</i>	16	Ошибка датчика	Хотя бы одна входная переменная не удовлетворяет условиям, накладываемым на входные переменные алгоритмом, или хотя бы одна входная переменная пришла с любым из плохих признаков качества
<i>Opс_quality_last_known</i>	20	Известно последнее значение	Сбой связи. Последнее опрошенное значение доступно. Время опроса имеется в отметке времени, сопровождающей значение
<i>Opс_quality_comm_failure</i>	24	Нет связи	Связи с источником данных для какого-либо из входов нет. Последнее опрошенное значение не существует. Или OPC-сервер еще не запущен
<i>Opс_quality_out_of_service</i>	28	Не обслуживается	При существующих настройках алгоритма данный выход не формируется
<i>Opс_quality_uncertain</i>	64	Значение не определено	Значение не определено
<i>Opс_quality_good</i>	192	Норма	Признак качества – норма

К полученному значению помимо признака качества OPC-сервер добавляет еще одно поле – метку времени. Метка времени либо может быть получена по сети от контроллера, либо OPC-сервер при получении данных устанавливает в качестве метки текущее время.

Привязанная метка времени позволяет при представлении или обработке данных уменьшить расхождение используемого или отображаемого времени с реальным временем измерения. Кроме того, если OPC-клиентов несколько, то они одни и те же данные получают с одинаковой меткой времени.

5.2.3. OPC HDA

В отличие от стандарта OPC DA, передающего текущие данные, стандарт OPC HDA предназначен для передачи архивных данных. OPC HDA применяется в тех случаях, когда опрос устройства производится реже, чем меняются его данные (например, ежесуточный опрос счетчика электроэнергии для чтения почасовых профилей мощности), или в случаях, когда соединение с удаленным устройством нестабильное, но потеря данных не допускается.

Для получения данных по OPC HDA:

- устройство должно иметь возможность вести архив в памяти;
- протокол обмена должен поддерживать передачу архивных данных;
- к устройству должен поставляться OPC HDA сервер с реализацией чтения архива.

Как правило, OPC HDA сервер поставляется к различным приборам учета: счетчикам электроэнергии, тепловычислителям, расходомерам и т. д. OPC HDA сервер может быть разработкой как компании – производителя устройства, так и других фирм. Наша компания имеет OPC-серверы (как с поддержкой DA, так и HDA) для различных устройств: счетчиков [Меркурий](#), [Энергомера](#), [НЗИФ](#), [Милур](#) и ряда других устройств.

Как быть, если устройство имеет возможность вести архив, протокол имеет возможность передавать архивные значения, но производитель устройства OPC HDA (или даже OPC DA сервера) не предоставляет? В этом случае можно попробовать разработать собственный OPC-сервер – для этого у нашей компании также имеется полный перечень инструментов. Если протокол несложен и количество опрашиваемых параметров невелико, можно реализовать обмен, используя [Modbus Universal MasterOPC сервер](#) или плагин [User Script](#) Multi-Protocol MasterOPC сервера. Если же протокол сложен, то можно реализовать его на языке C++, используя плагин [User Protocol](#) того же Multi-Protocol MasterOPC сервера. Инструкции по разработке протоколов есть на странице продуктов.

Принцип работы со всеми OPC HDA серверами в MasterSCADA одинаков. Первое, что нужно сделать, – произвести поиск доступных OPC HDA серверов и отметить нужный галочкой.

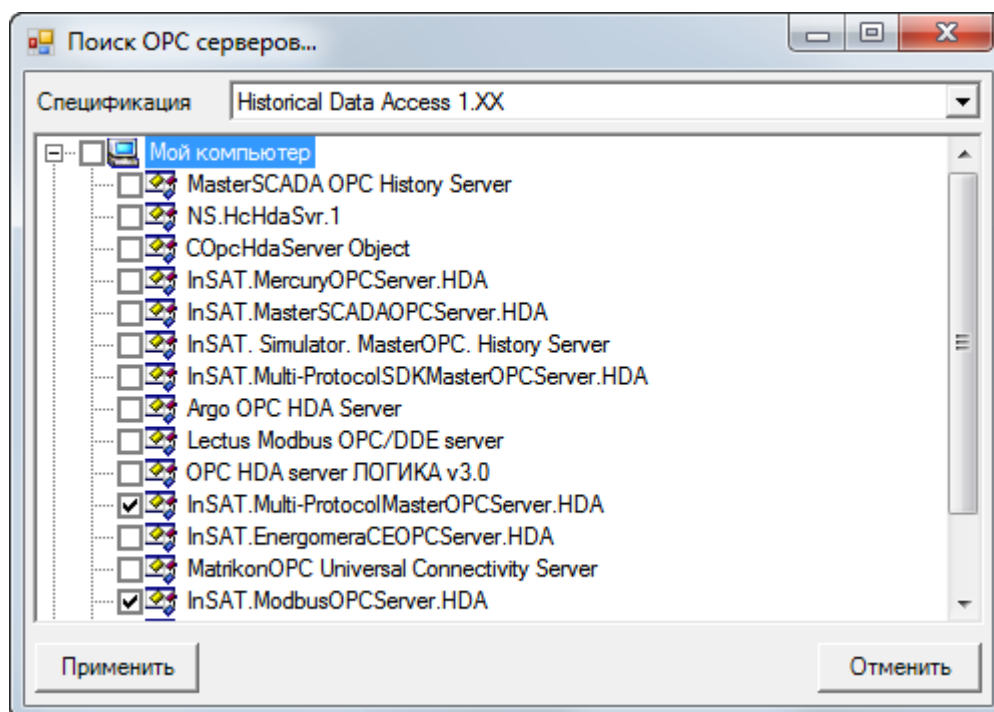


Рисунок 5-4. Окно поиска OPC HDA серверов

Затем добавить OPC HDA сервер в компьютер.

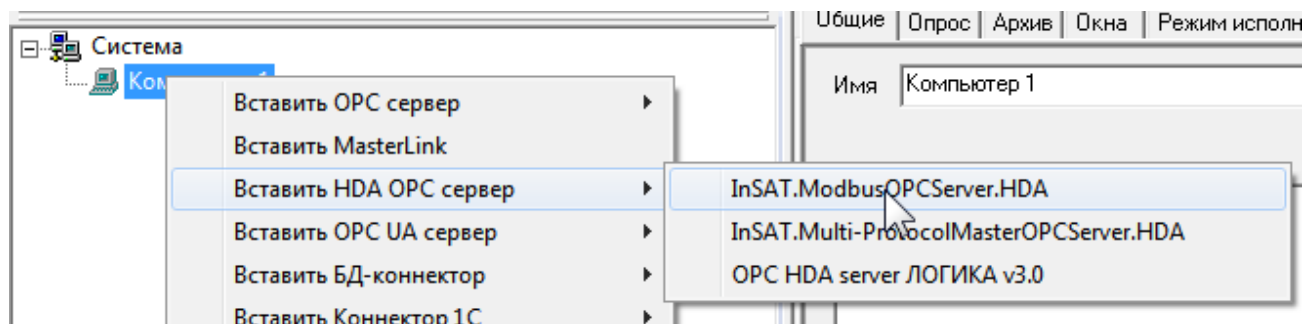


Рисунок 5-5. Вставка OPC HDA сервера

Затем нужно добавить в OPC HDA теги и настроить алгоритм получения архива. Оба этих действия производятся на вкладке **Настройки**. После перехода на данную вкладку происходит загрузка доступных HDA-тегов (Рисунок 5-6).

Работа с оборудованием

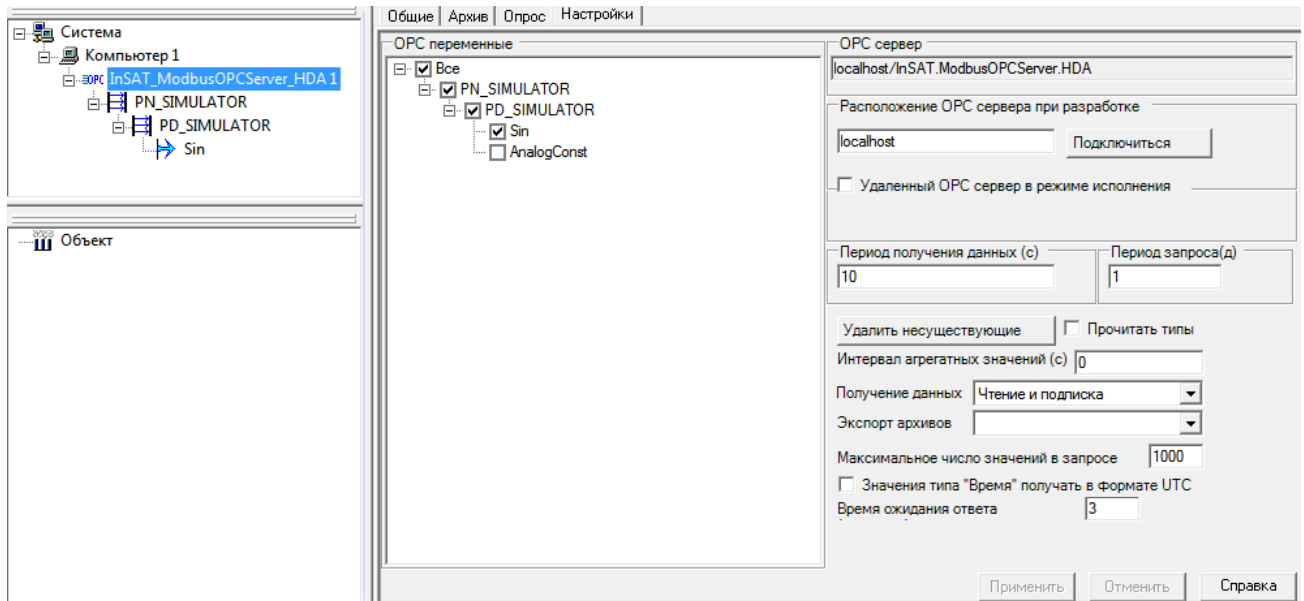


Рисунок 5-6. Настройка OPC HDA сервера

Отметьте необходимые теги галочками. После нажатия кнопки **Применить** они добавятся в дерево системы.

Выделим OPC HDA тег и перейдем на вкладку **Архив**. Обратите внимание, что у тега уже включена периодическая архивация с шагом в 0 секунд. Не нужно переопределять эти настройки: при таких настройках происходит получение всех архивных данных из OPC HDA сервера.

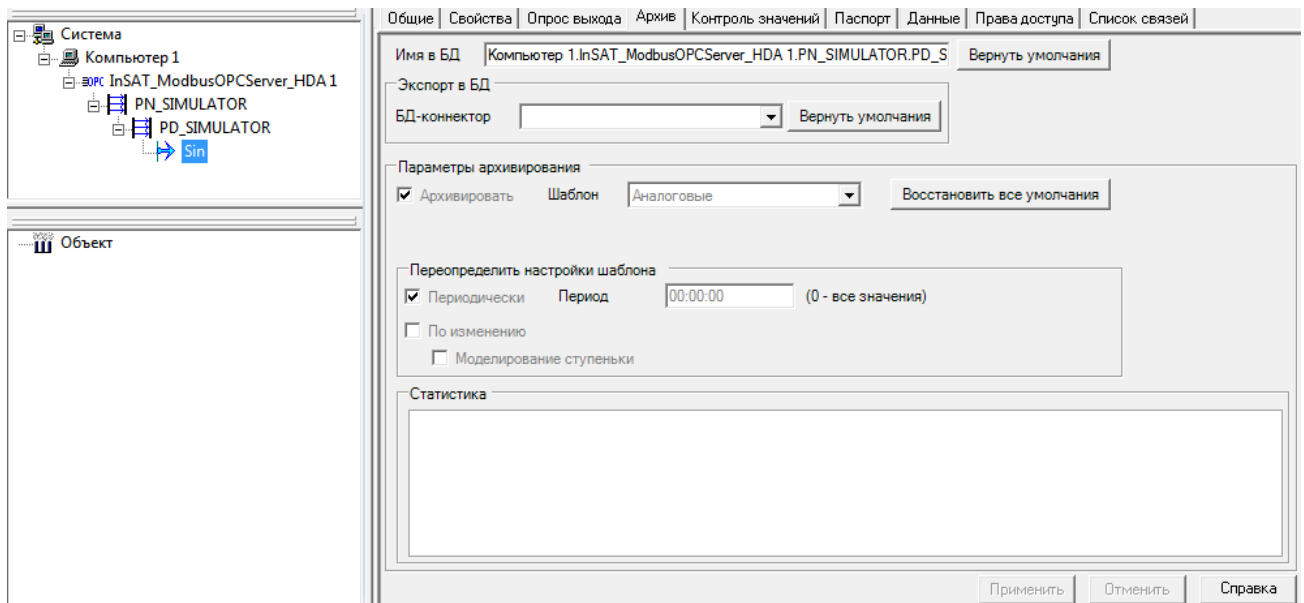


Рисунок 5-7. Архивация OPC HDA тега по умолчанию

Снова вернемся на вкладку **Настройки**. Основными настройками являются: **Получение данных**, **Период запроса (д)**, **Период получения данных (с)**. Настройка **Получение данных** определяет метод получения данных. Как и в OPC DA, в OPC HDA есть два метода получения данных: **Подписка** и **Чтение** (полный опрос). Данная настройка определяет, каким методом будут получены данные из OPC HDA сервера. Наиболее оптимальным является метод **Чтение и Подписка**, в этом режиме SCADA с помощью метода **Чтение** получает данные из сервера за количество дней, определенных в настройке **Период запроса (д)**, а затем подписывается на получение новых данных с периодом, заданным в настройке **Период получения данных (с)**.

К сожалению, не все OPC HDA серверы поддерживают оба метода получения данных, поэтому если данные от сервера не поступают, то стоит попробовать перебрать разные варианты настройки **Получение данных**. OPC-серверы нашей разработки поддерживают оба метода, поэтому для них лучше оставить стандартный способ получения данных. Параметр **Период запроса (д)** по умолчанию равен 1 дню. Вам необходимо установить нужную вам глубину получения данных при старте, например 30 дней.

Примечание. Необходимо иметь в виду, что данная настройка не всегда определяет, что из прибора будут считаны данные именно на эту глубину. В наших OPC HDA серверах есть специальная настройка, которая определяет глубину считывания данных из прибора (причем она не обязательно может измеряться в днях, а может, например, измеряться в количестве запрашиваемых архивных записей). Эти данные попадают в специальный кеш HDA-данных, из которых и будет забирать архивы SCADA, – то есть настройка «Период запроса (д)» определяет глубину считывания из этого кеша. Нужно иметь это в виду и не забывать задавать корректную глубину считывания при старте в самом OPC-сервере.

5.2.4. OPC UA

OPC UA – это новый стандарт технологии OPC. Новый стандарт полностью отвязан от технологии DCOM и является кроссплатформенным. Кроме того, он является унифицированным: через единый интерфейс могут передаваться текущие данные, архивные данные, а также сообщения.

Несмотря на сложность самого стандарта, настройка со стороны пользователя предельно проста. Все, что требуется, – это сообщить OPC UA клиенту IP-адрес и порт компьютера (или контроллера), на котором установлен OPC UA сервер, а при использовании шифрования – логин и пароль. При этом есть возможность выполнить автоматический поиск OPC UA серверов на определенном компьютере, но, как правило, этого делать не требуется: задать нужный порт вручную будет даже быстрее.

Рассмотрим подключение OPC UA на примере Multi-Protocol MasterOPC сервера – OPC DA, HDA, UA сервера нашей разработки (<http://insat.ru/products/?category=2350>). В качестве примера возьмем конфигурацию с имитатором значений. В устройство добавлены 10 тегов, содержащих скрипт имитации случайного сигнала.

Текущая конфигурация : имитатор.mpp

Объекты

- Server
 - SCRIPT
 - script
 - Tag1
 - Tag2
 - Tag3
 - Tag4
 - Tag5
 - Tag6
 - Tag7
 - Tag8
 - Tag9
 - Tag10

Протокол <<SCRIPT>> Устройство <<script>>

Общие настройки	
Комментарий	Название устройства
Включено в работу	true
Период опроса	1000
Размерность периода опроса	мс
Начальная фаза	0
Размерность фазы	мс
Старт после запуска	true
Разрешение отладочных сообщений	true
Скрипт	
Выполнение скрипта	false

Рисунок 5-8. Дерево OPC-сервера с имитационными тегами

Пока что данные можно получать только по DA- и HDA-интерфейсу. Сделаем так, чтобы данные можно было получать и по UA. Для этого нужно выделить корневой элемент сервера и включить подключение OPC UA сервера.

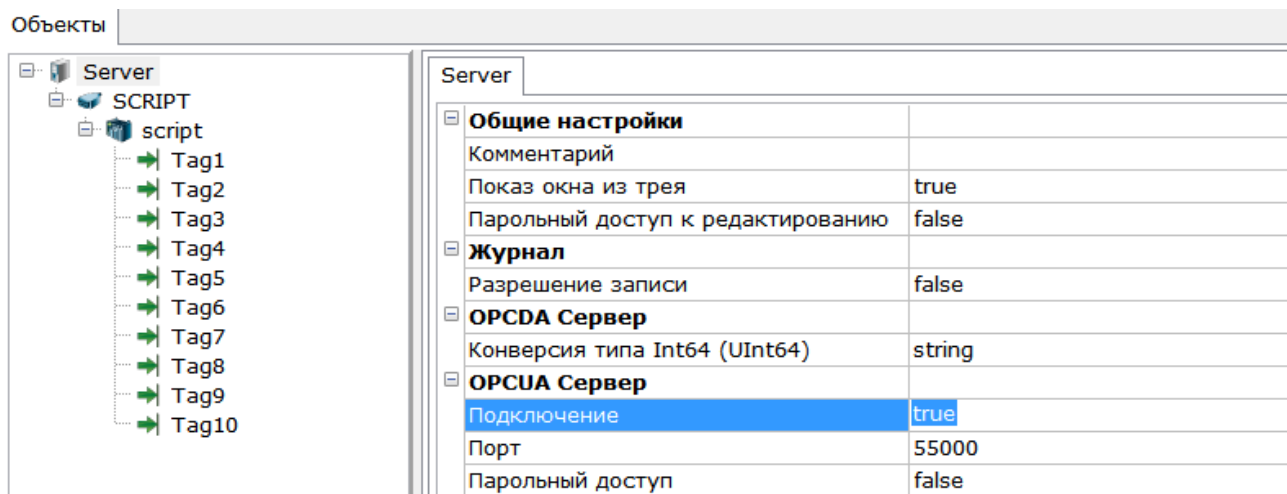


Рисунок 5-9. Активация режима OPC UA сервера

По умолчанию используется порт 55000, при необходимости его можно изменить. Также можно включить парольный доступ к OPC-серверу.

Запустим OPC-сервер в режим исполнения.

Внимание! В отличие от технологии OPC DA, в OPC UA нет возможности автоматического запуска сервера при подключении к нему клиента. Сервер нужно запустить как приложение или как службу заранее.

В режиме исполнения у OPC-сервера происходит изменение значений по случайному закону (Рисунок 5-10).

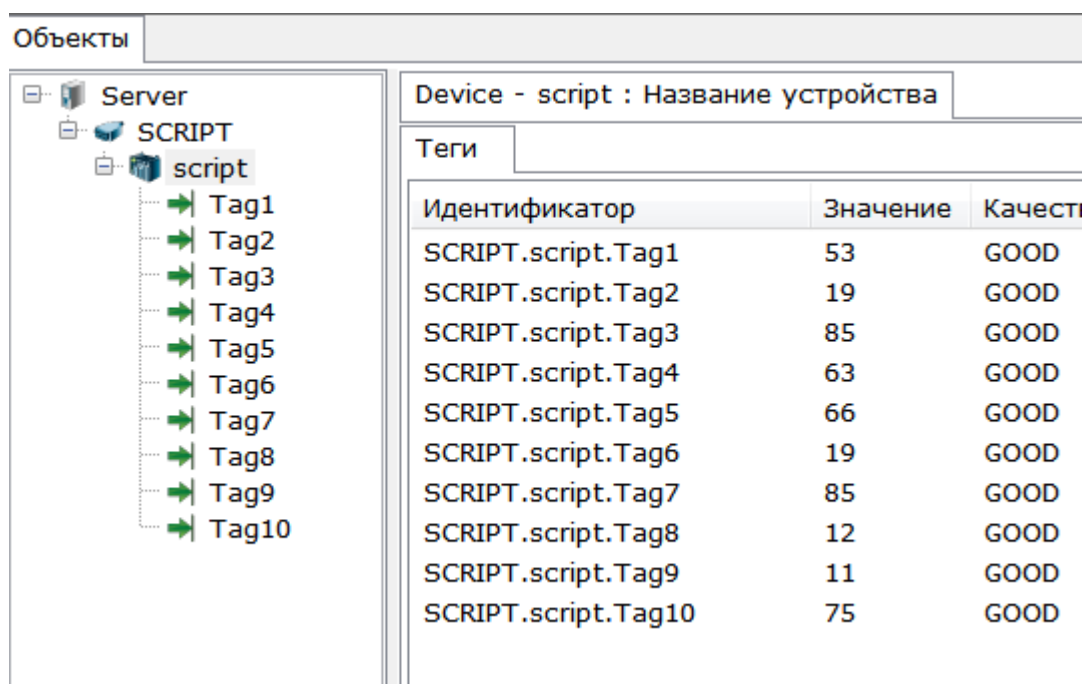


Рисунок 5-10. Значения тегов в режиме исполнения

Подключимся к нашему серверу из MasterSCADA. Добавим в систему компьютер, а в него OPC UA сервер.

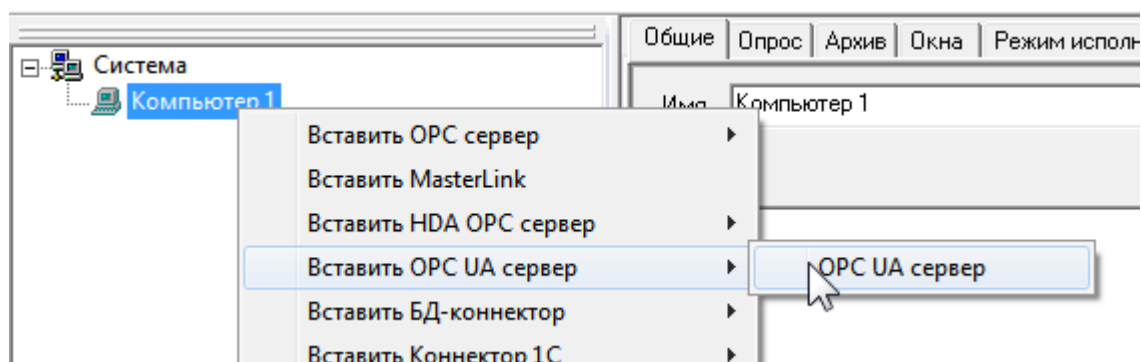


Рисунок 5-11. Добавление OPC UA сервера в MasterSCADA

Перейдем на вкладку **Настройки** OPC-сервера.

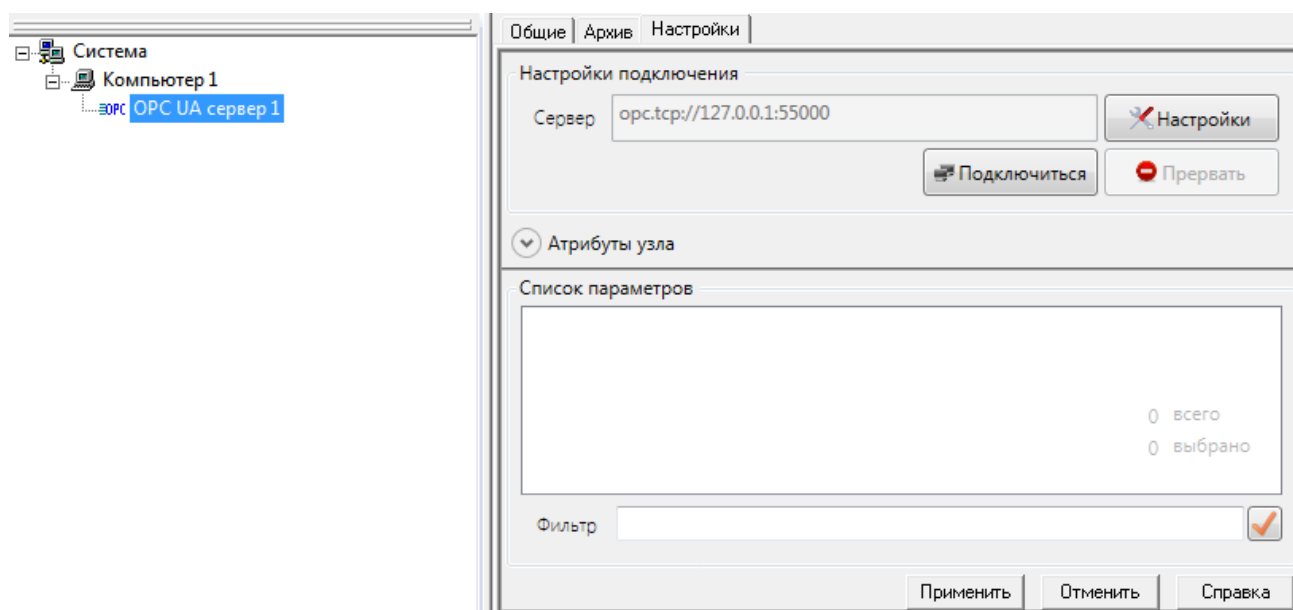


Рисунок 5-12. Настройка подключения к OPC UA серверу

На данной вкладке осуществляется настройка параметров подключения к серверу, выбор нужных тегов, а также просмотр текущих значений. Для задания подключения к серверу используется строка **Сервер** – сейчас в ней задана строка:

Оpc.tcp://127.0.0.1:55000

Данная строка означает, что будет произведено подключение к серверу по протоколу **opc.tcp** по IP-адресу **127.0.0.1** (*localhost* – то есть текущий компьютер) по порту **55000**. Эта строка соответствует стандартным параметрам Multi-Protocol MasterOPC, и в нашем случае ее можно менять. Если же используется другой сервер, то нужно нажать на кнопку **Настройки**. В появившемся окне на вкладке **Настройки подключения** можно исправить строку подключения, а также добавить резервный OPC-сервер.

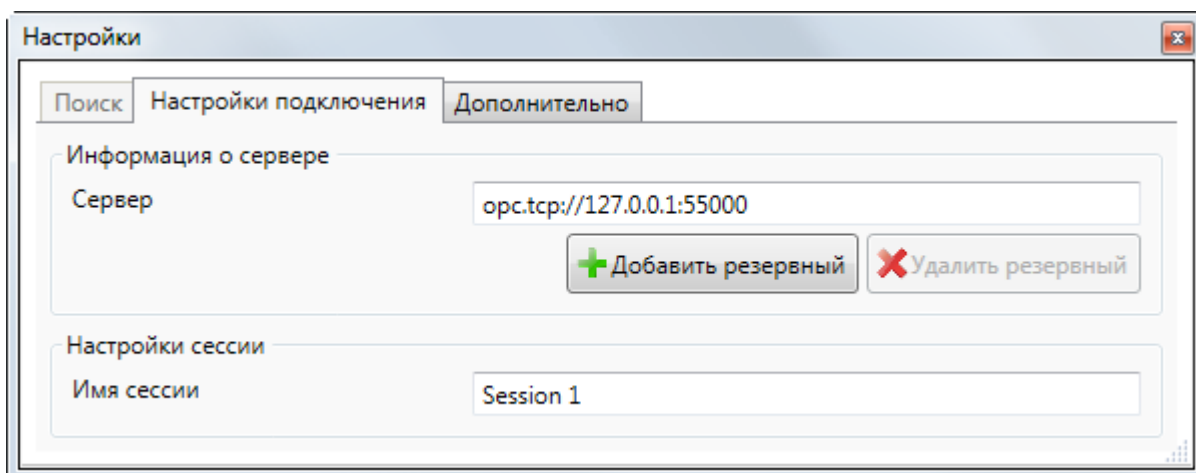


Рисунок 5-13. Параметры подключения

На вкладке **Дополнительно** (Рисунок 5-14) задаются различные параметры, отвечающие за оптимизацию чтения дерева узлов из сервера (количество возвращаемых ссылок, атрибутов), параметры получения данных (период чтения, период записи), параметры работы с резервированным сервером. Также можно включить чтение архивов – в этом случае из переменных с включенным атрибутом **Historian** (исторические, архивные) OPC UA клиент будет читать архивы и записывать в архив тегов.

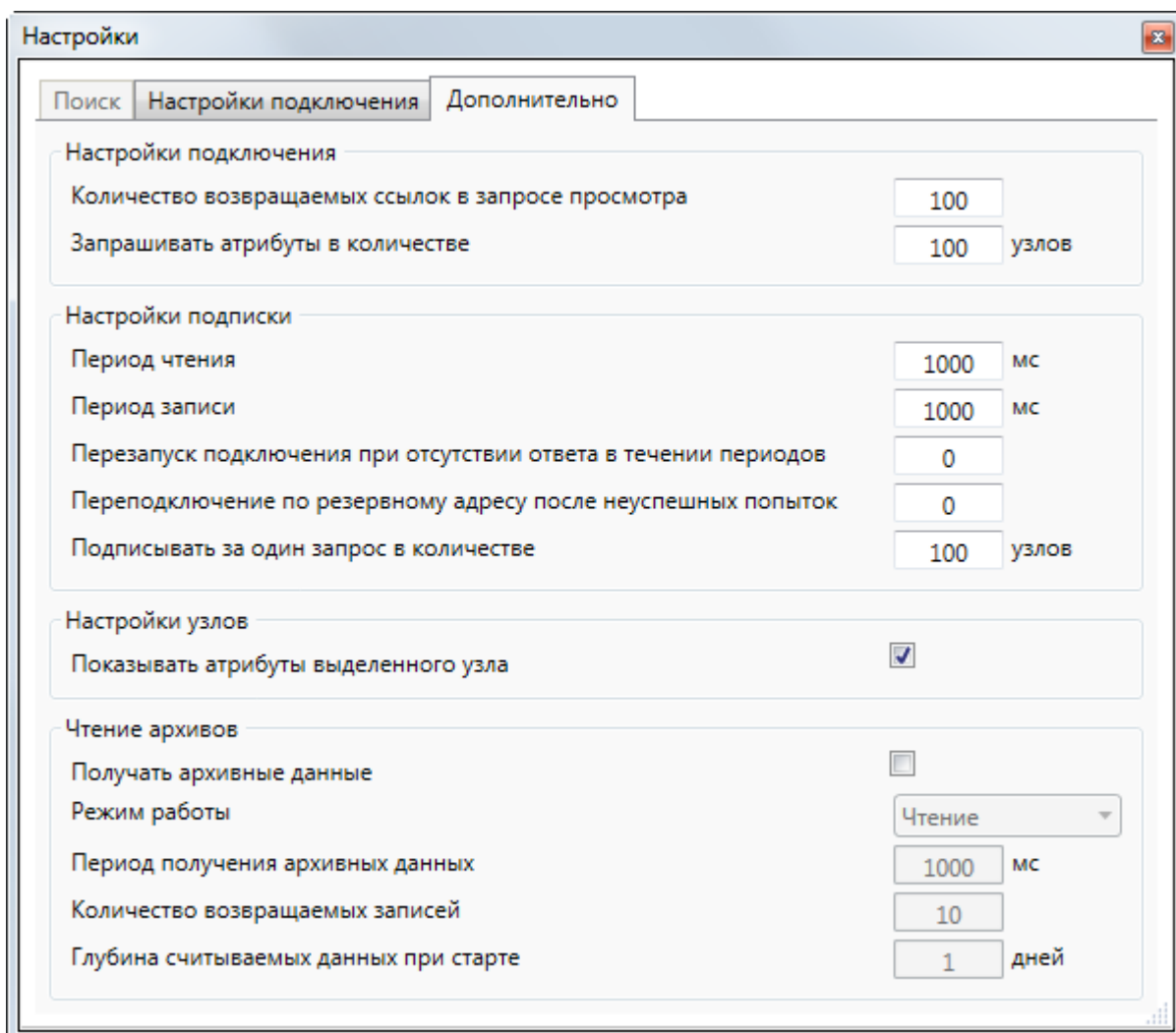


Рисунок 5-14. Дополнительные настройки подключения

Нам здесь ничего менять не требуется, поэтому просто закроем окно.

Нажмем на кнопку **Подключиться** – произойдет попытка подключения к OPC UA серверу. Если попытка будет успешной, произойдет загрузка всех узлов OPC UA сервера, из которых будет сформировано дерево.

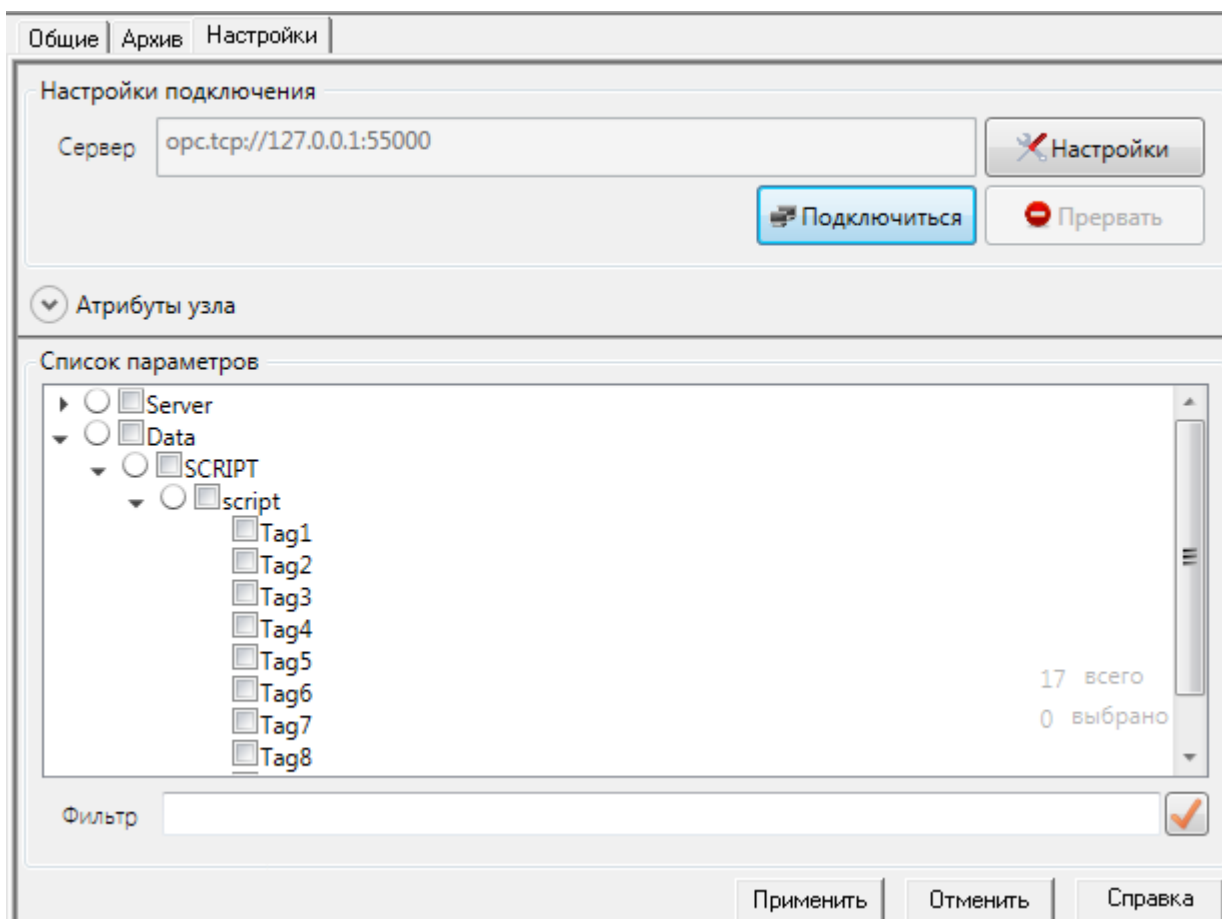


Рисунок 5-15. Получение списка OPC UA тегов

Каждый из тегов можно выделить отдельно и, активировав вкладку **Атрибуты узла**, посмотреть параметры этого тега (Рисунок 5-16).

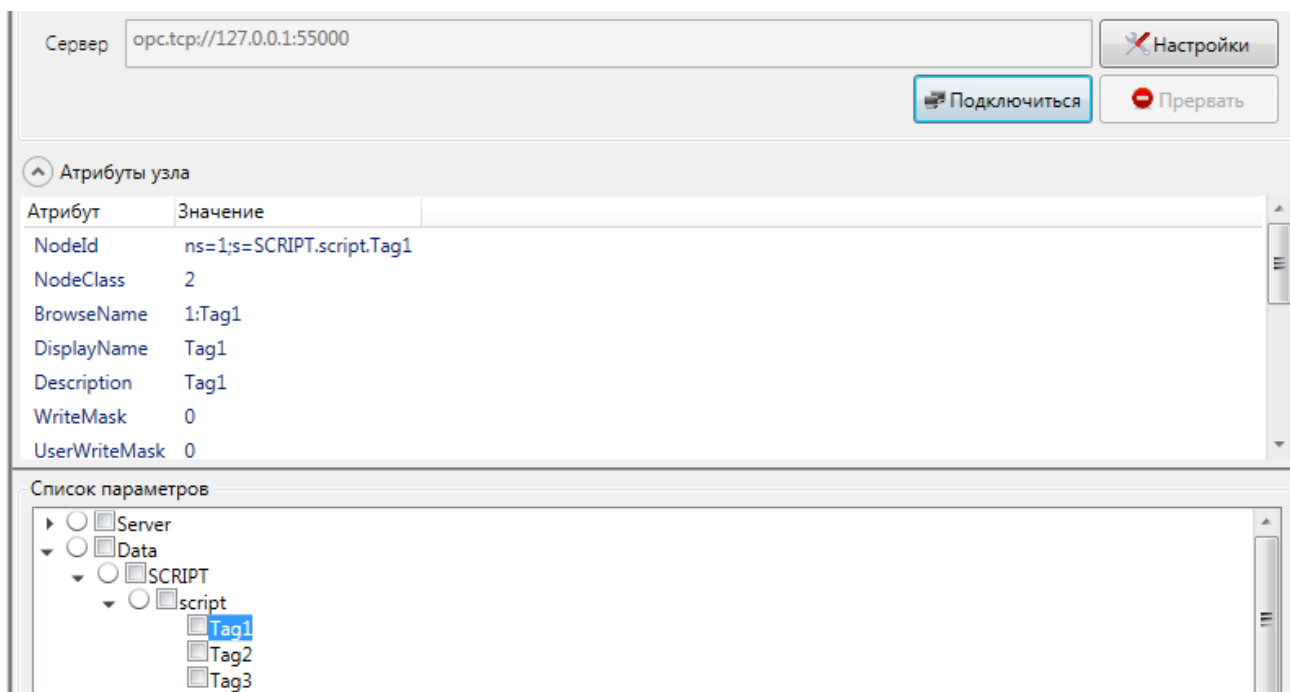


Рисунок 5-16. Атрибуты тега

Нужные теги или узлы отметим галочками и нажмем кнопку **Применить** – они будут добавлены в дерево системы.

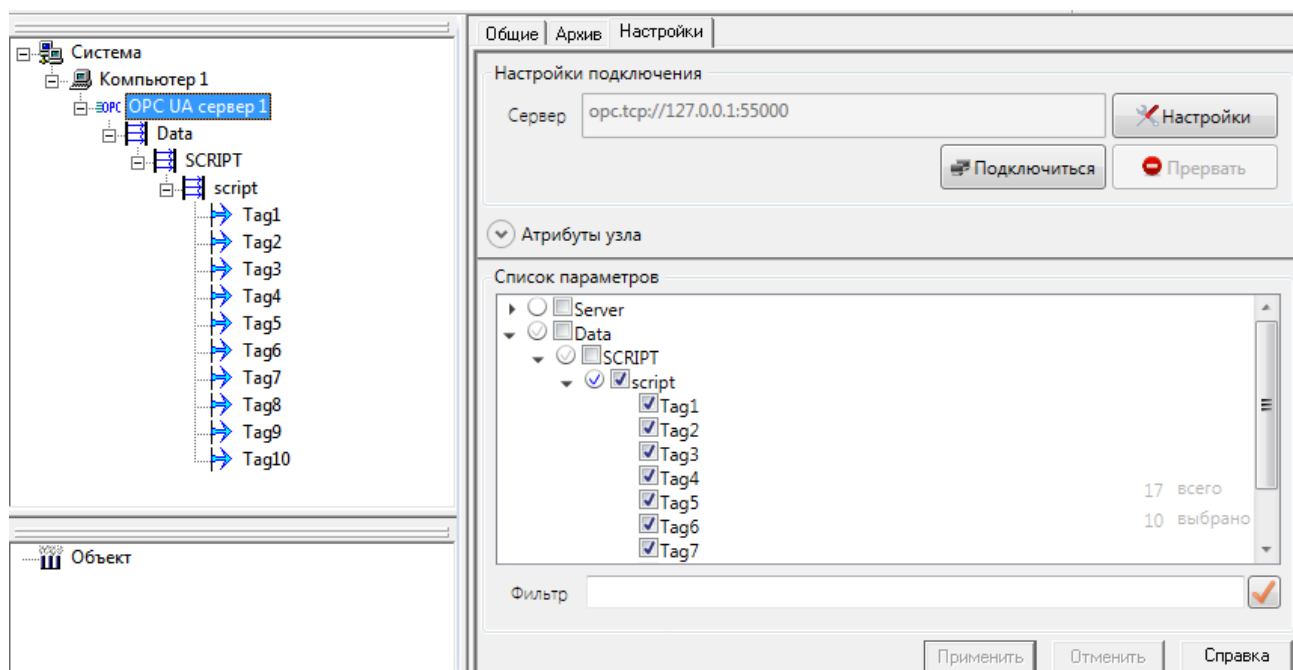
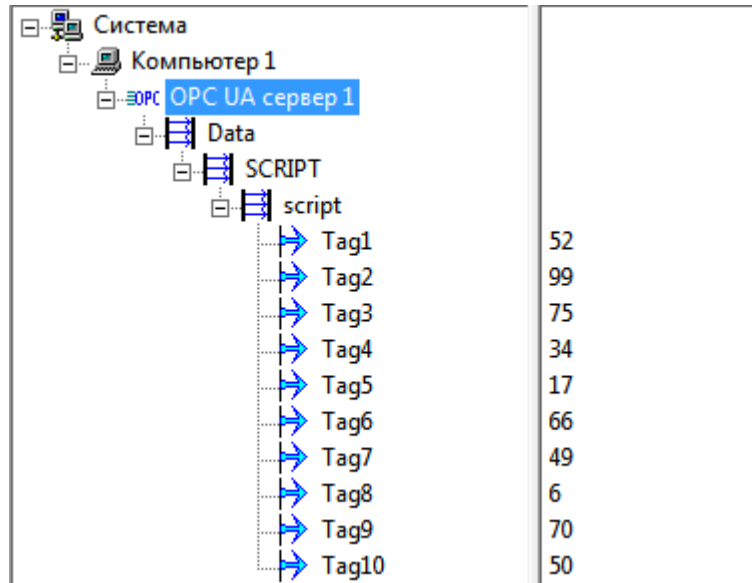


Рисунок 5-17. Выбор тегов для вставки в дерево системы

Теперь с данными тегами можно работать, как с обычными переменными в MasterSCADA: связывать их с другими переменными, вытаскивать на мнемосхемы, тренды, отчеты.

Запустим режим исполнения – значения из сервера поступили в систему.



The screenshot shows a tree view of the system configuration. Under 'Система' (System), there is 'Компьютер 1' (Computer 1), which contains 'ОПС OPC UA сервер 1' (OPC UA server 1). Under the server, there is a 'Data' folder containing a 'SCRIPT' folder, which in turn contains a 'script' folder. To the right of the tree view, a list of tags is displayed with their corresponding values:

Tag1	52
Tag2	99
Tag3	75
Tag4	34
Tag5	17
Tag6	66
Tag7	49
Tag8	6
Tag9	70
Tag10	50

Рисунок 5-18. Значения OPC UA тегов в режиме исполнения

5.3. Получение данных по протоколу Modbus с использованием OPC-сервера

Наиболее распространенным промышленным протоколом является **Modbus**. Несмотря на свой «солидный» возраст (первая редакция вышла еще в 1970-х), **Modbus** за счет открытости стандарта, простоты реализации и скорости работы продолжает активно применяться до сих пор.

Для взаимодействия с устройствами **Modbus** мы предлагаем вам использовать OPC-сервер нашей разработки – **Modbus Universal MasterOPC сервер**. Данный OP-сервер поддерживает стандарты **OPC DA 2.05** и **OPC HDA 1.2**. В отличие от многих других Modbus OPC-серверов **Modbus Universal** поддерживает все функции протокола **Modbus** (в том числе из перечня функций расширенного стандарта) за счет наличия встроенного сценарного языка разработки. Также можно, используя встроенный язык разработки, выполнять произвольный запрос (как по COM-портам, так и TCP-соединениям), обращаться к другим тегам и устройствам, работать с файлами и базами данных.

Работа с оборудованием

Создадим проект с реальным прибором, работающим по протоколу **Modbus RTU**, например с прибором **ОВЕН МДВВ** (модуль дискретного ввода/вывода).

Запустим **Modbus Universal MasterOPC Server**.

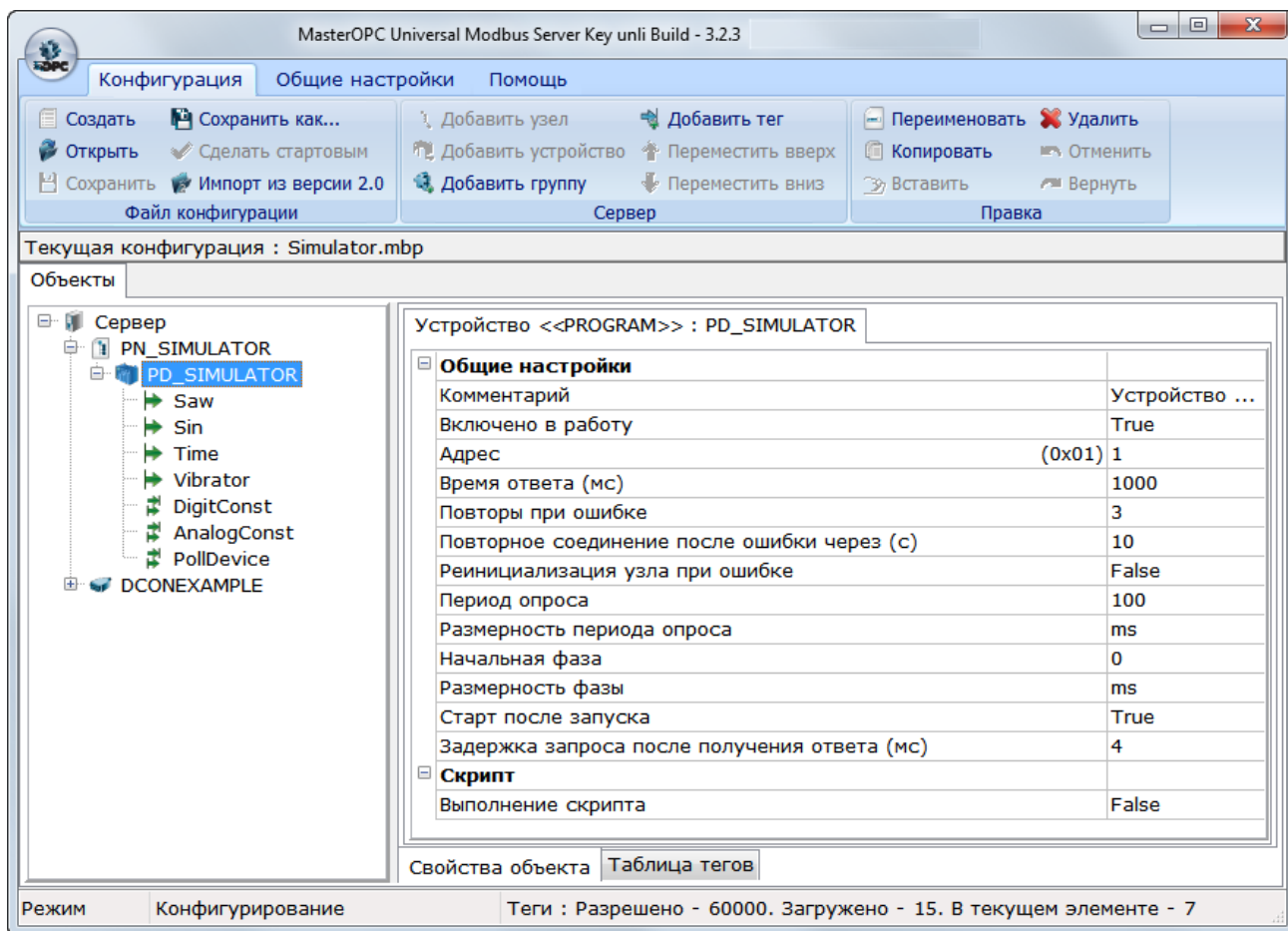


Рисунок 5-19. Интерфейс Modbus Universal MasterOPC сервера

Добавим в конфигурацию узел и зададим параметры связи с устройствами: **номер СОМ-порта, скорость, четность, стоповые биты**.

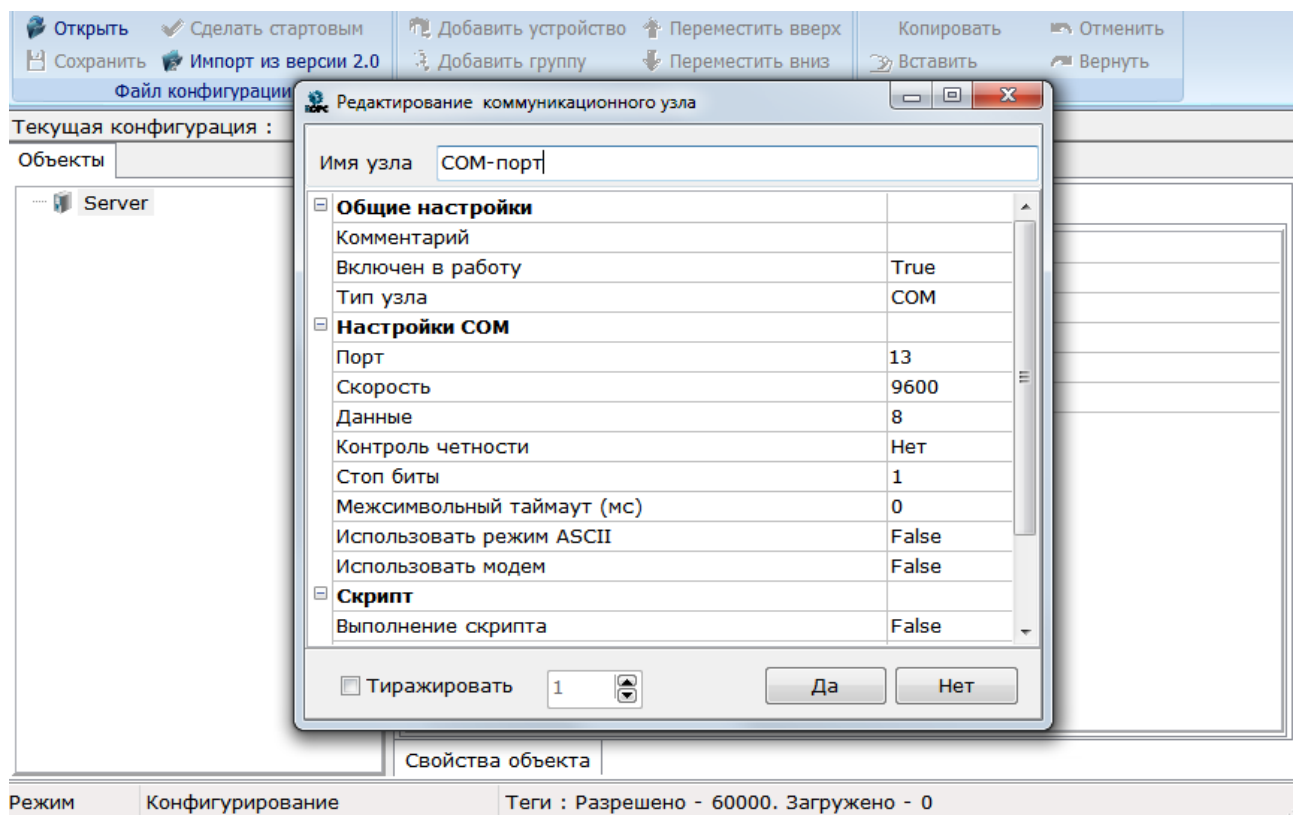


Рисунок 5-20. Добавление и настройка COM-порта

В узел, также через контекстное меню, добавим устройство и укажем адрес устройства – в нашем случае он равен 16.

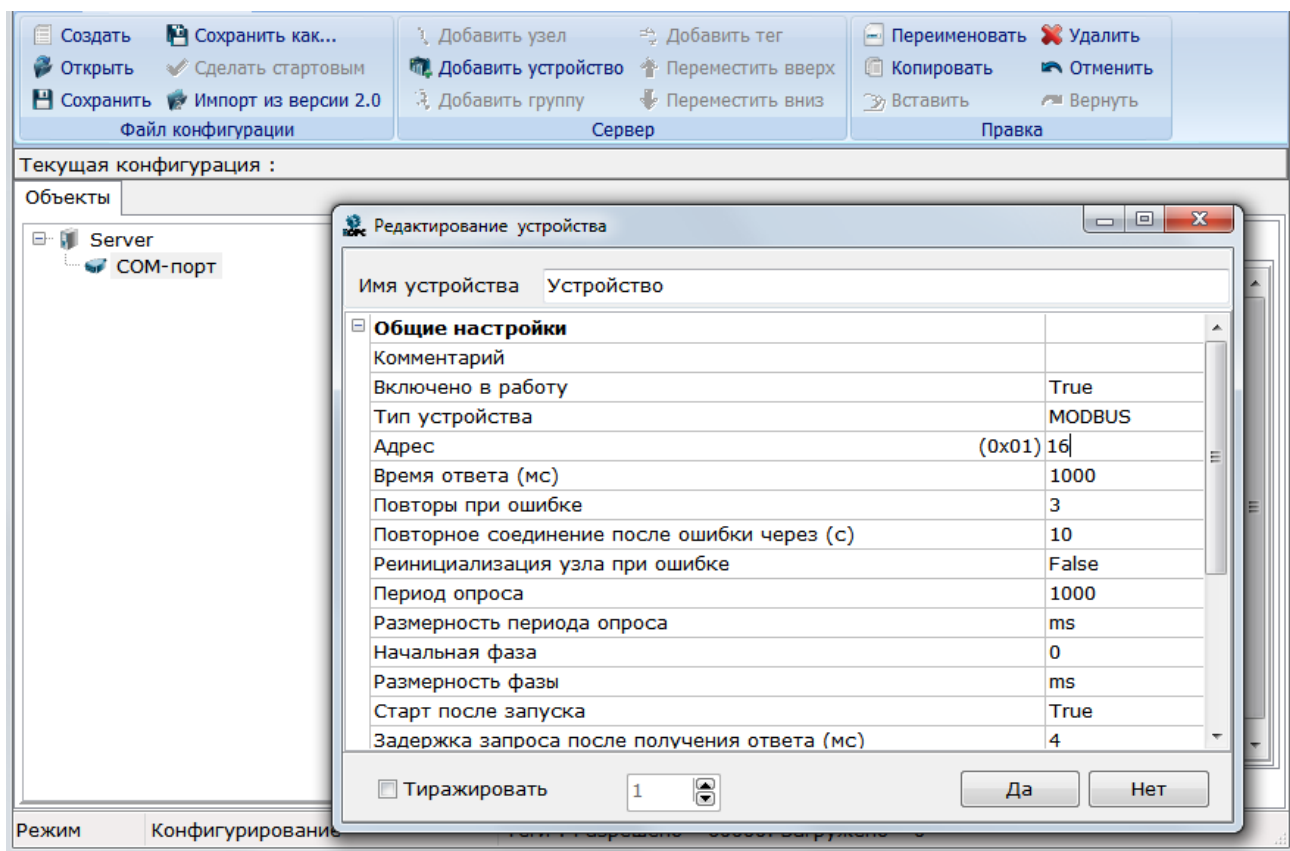


Рисунок 5-21. Добавление устройства и задание адреса

Теперь в устройство необходимо добавить переменные и указать им номера регистров в приборе. Для примера будем опрашивать вход и выход прибора. **МДВВ** имеет 12 входов и 8 выходов, входы могут опрашиваться по маске или в режиме счетчика. Входы в режиме счетчика имеют номера регистров от 64 до 75 в десятичной системе исчисления. Из этих регистров можно производить считывание накопленного значения, а также записывать значение – для сброса. Таким образом, регистры используются для чтения и записи.

Добавим первый вход в наш OPC. Для этого вызовем контекстное меню устройства и добавим тег. В появившемся окне зададим регион тега – в нашем случае это будет **Holding Registers** (регистр хранения, чтение производится функцией 0x03, запись – 0x06 или 0x10). В поле **Адрес** укажем номер регистра – **64** и зададим имя – **Вход1**.

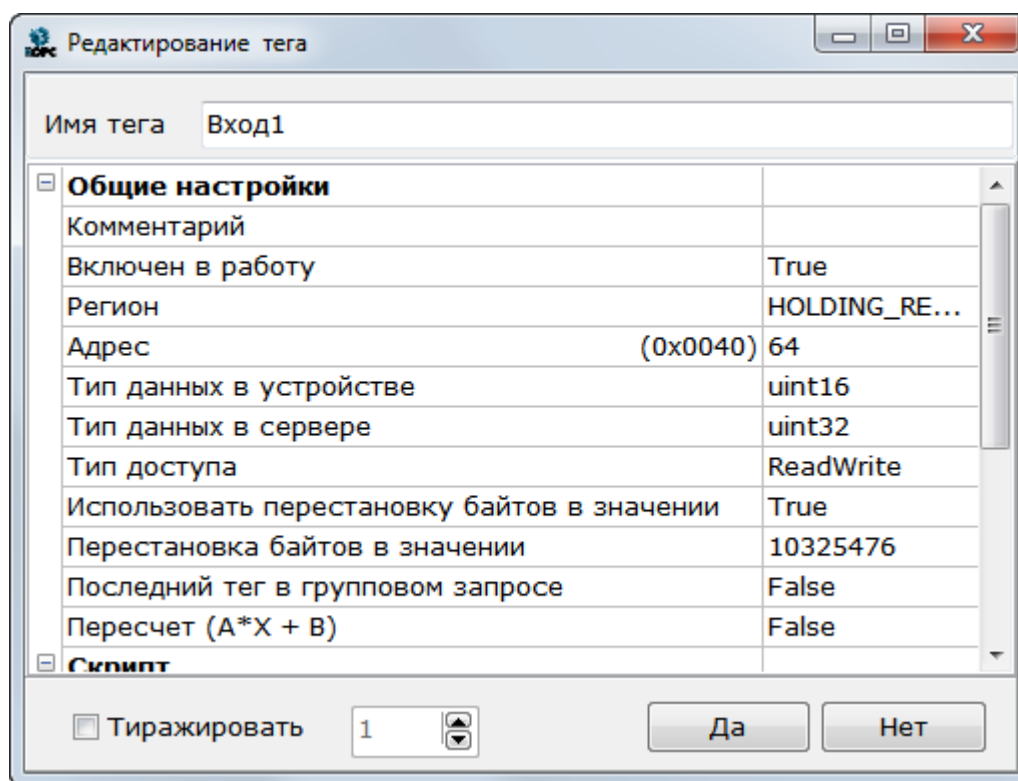


Рисунок 5-22. Окно добавления тега

Сохраним конфигурацию и сделаем ее конфигурацией по умолчанию.

Запустим MasterSCADA, произведем поиск OPC-устройств, добавим OPC и все OPC-переменные (рис. 6-23).

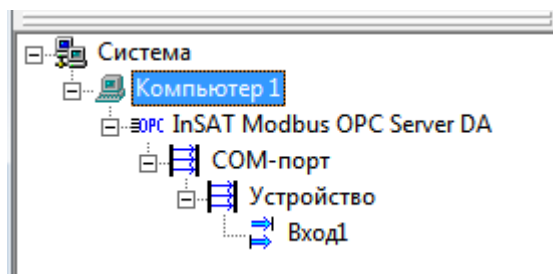


Рисунок 5-23. Вставка тегов в дерево системы

Вход1 добавился в дерево системы. Обратите внимание, что переменная имеет значок двунаправленной, то есть можно осуществлять как чтение переменной, так и запись в нее (рис. 6-24).

Добавим в дерево объекта одно **Значение** и команду. Установим связь между переменной OPC и **Значением**. Также установим связь между OPC-переменной и командой **Запись**.

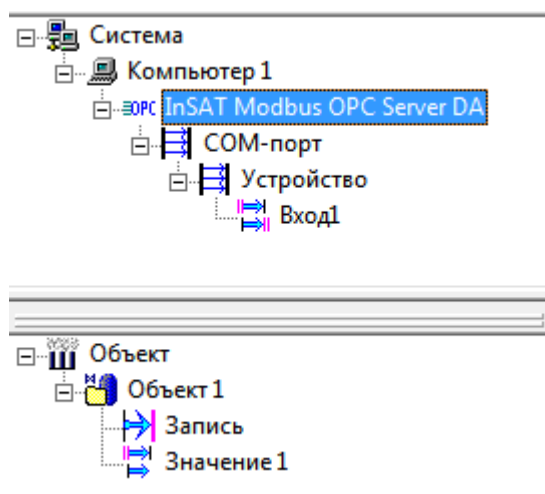


Рисунок 5-24. Установленные связи

Значок OPC-переменной изменился – он отображает, что вход и выход переменной имеют связи. Аналогичные значки появились у команды и значения.

Запустив режим исполнения, можно убедиться, что значения из прибора поступают, а при записи значения 0 происходит сброс накопленных значений.

Примечание. На странице загрузки OPC-сервера можно скачать различные видеопримеры по работе с «Modbus Universal MasterOPC сервером»: создание конфигураций для Modbus RTU и Modbus TCP устройств, написание скриптов, обзор различных вспомогательных функций. Также есть текстовая документация по работе с GSM-модемами, инструкции по подключению контроллеров разных производителей (ОВЕН, Siemens, ABB, Delta и т. д.).

5.4. Подключение приборов через MasterPLC

Кроме создания проекта SCADA-системы, MasterSCADA позволяет программировать контроллеры с открытой архитектурой (SoftPLC). В этом случае проект SCADA и ПЛК создается в единой среде программирования как единый проект автоматизации. Рассмотрим преимущества такого подхода к разработке.

Например, нам необходимо создать проект автоматизации, состоящий из ПЛК и SCADA-системы. Сначала рассмотрим вариант создания проекта с контроллерами закрытой архитектуры – они программируются в собственных средах разработки. В этом случае создание проекта будет состоять из следующих этапов:

Работа с оборудованием

- 1) Создание программы для контроллера и его отладка в среде программирования контроллера.
- 2) Создание карты адресов параметров (например, **Modbus**) в контроллере для обмена с верхним уровнем.
- 3) Настройка OPC-сервера – прописывание карты адресов для обмена с нижним уровнем.
- 4) Создание проекта SCADA-системы и его отладка.
- 5) Установление связей между проектом SCADA и переменными OPC-сервера.
- 6) Отладка всего проекта.

В процессе создания придется переключаться между тремя различными программами (среда программирования ПЛК, конфигуратор OPC-сервера, SCADA-система) и вручную создавать карту адресов обмена (в которых нередко можно запутаться).

В последнее время в промышленности широко используются так называемые распределенные системы управления, когда управление осуществляют несколько небольших ПЛК, расположенных непосредственно возле объекта, – это упрощает монтаж и сокращает количество подводимых проводов. Рассмотрим создание такого проекта – с несколькими ПЛК с закрытой архитектурой. В этом случае для каждого ПЛК нужно будет написать программу, создать свою карту адресов, настроить их в OPC и добавить в проект SCADA-системы. Проект еще больше усложнится, если контроллеры должны вести обмен не только с верхним уровнем, но и друг с другом (без участия компьютера).

В случае использования контроллеров с открытой архитектурой и **MasterPLC** (среда исполнения проекта в контроллере) проект SCADA и ПЛК создается в единой среде – в MasterSCADA. Чтобы создать проект, необходимо добавить в дерево системы контроллер и назначить ему объект с созданной программой. Создание адресов, реализация обмена с верхним уровнем и межконтроллерного обмена производятся средой разработки автоматически. Таким образом создание и отладка проекта значительно упрощаются.

Контроллеры с открытой архитектурой работают под управлением какой-либо встраиваемой операционной системы. Для использования **MasterPLC** поддерживаются DOS,

MiniOS 7, Linux, WinCE. На данный момент поддерживаются контроллеры фирм Advantec, ICP DAS, Моха, «ОВЕН», «Текон» (список контроллеров будет пополняться).

5.4.1. Создание проекта с контроллером

Рассмотрим вариант создания проекта с использованием контроллеров.

Как и в случае с обычным проектом, добавим в проект компьютер. Контроллер добавляется через контекстное меню компьютера. В отсутствие реального контроллера можно использовать эмулятор – Windows-контроллер. Добавим его.

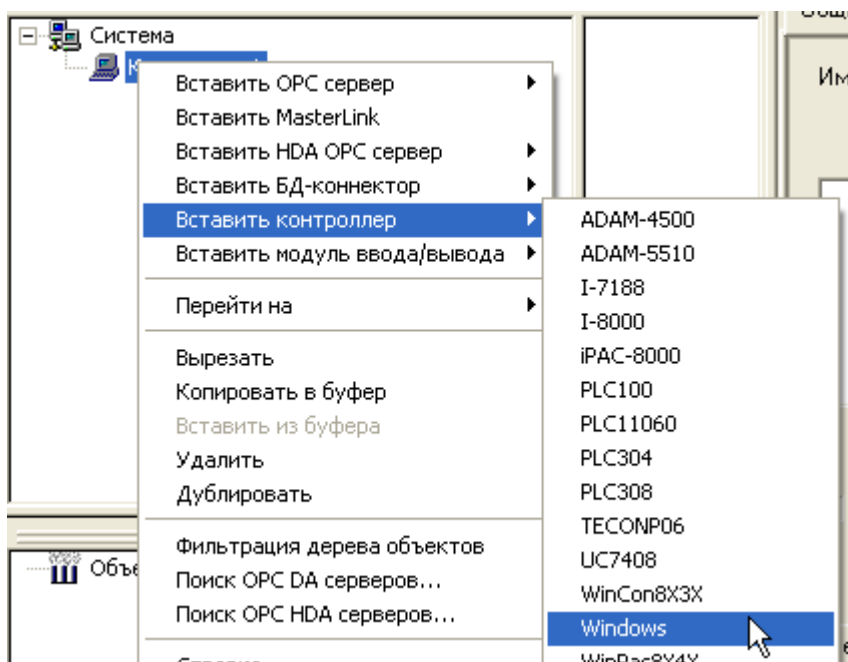


Рисунок 5-25. Добавление Windows-контроллера

Контроллер появился в дереве системы.

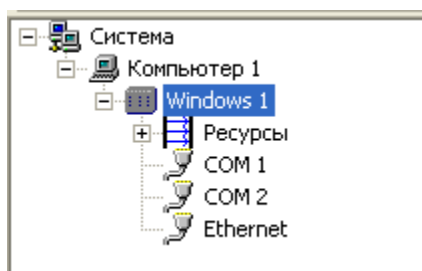


Рисунок 5-26. Контроллер в дереве системы

Теперь создадим проект для контроллера. Сначала назначим выполнение главного объекта на **Компьютер 1**, добавим в него объект и назовем его **Программа контроллера**.

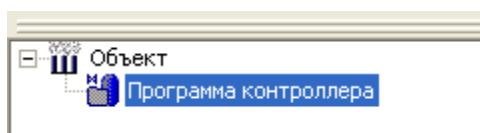


Рисунок 5-27. Создание объекта контроллера

Теперь создадим небольшой проект – генератор и счетчик импульсов с возможностью остановки. Для этого сначала добавим в объект первый ФБ – *Пульсатор* из закладки *Генераторы значений* палитры ФБ. Назовем его *Генератор*.

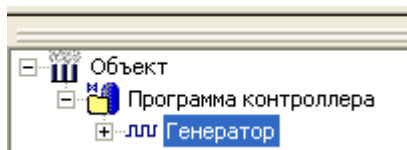


Рисунок 5-28. Добавление ФБ в объект

Для запуска мы будем использовать ФБ *Триггер* из закладки *Обработка сигналов*. Импульс на счетчик будет проходить, когда есть сигнал на выходе генератора и триггер включен, поэтому необходимо добавить элемент *Логическое И*. Теперь добавим элемент ФБ *Счетчик дискретных импульсов* из закладки *Обработка сигналов*.

Настроим ФБ *Счетчик*. Необходимо установить на входе *Максимальное значение* какое-нибудь значение, например **1000**.

Добавим также две команды дискретного типа: *Пуск* и *Стоп*. Дерево проекта будет выглядеть следующим образом:

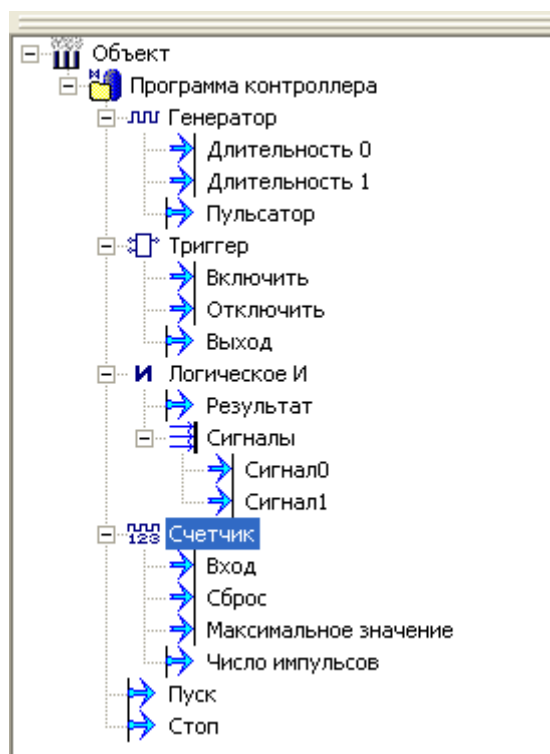


Рисунок 5-29. Дерево объектов с добавленными элементами

Теперь расставим связи:

Команда **Пуск** – вход **Включить** ФБ Триггер.

Команда **Стоп** – выход **Отключить** ФБ Триггер.

Выход **Выход** ФБ Триггер – вход **Сигнал0** ФБ Логическое И.

Выход **Пульсатор** ФБ Генератор – вход **Сигнал1** ФБ Логическое И.

Выход **Результат** ФБ Логическое И – вход **Вход** ФБ Счетчик.

Пока что создание проекта ничем не отличалось от создания проекта для верхнего уровня – SCADA-системы. Теперь, чтобы созданный нами проект работал в ПЛК, нужно назначить ему исполнение в контроллере. Для этого выделим объект **Программа контроллера** и на закладке **Общие** укажем исполнение на контроллере **Windows 1** (Рисунок 5-30).

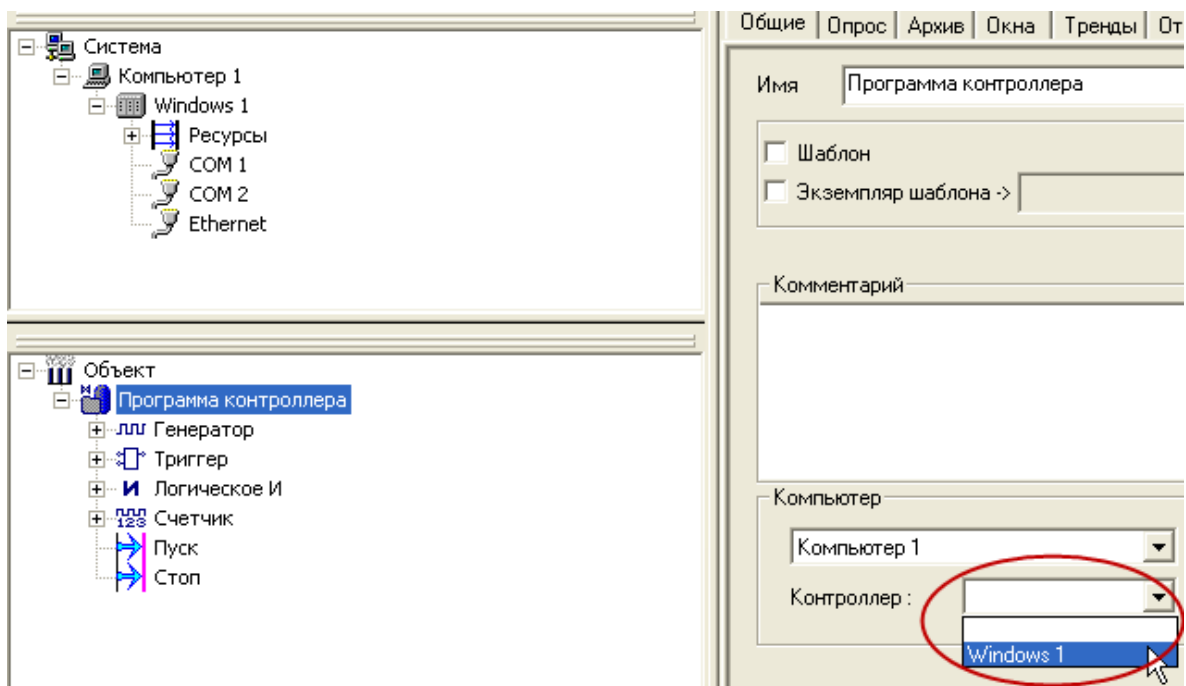


Рисунок 5-30. Назначение объекта к исполнению на контроллере

Теперь проект будет исполняться на указанном контроллере.

Запустим проект в режиме исполнения. После запуска появится всплывающее сообщение:

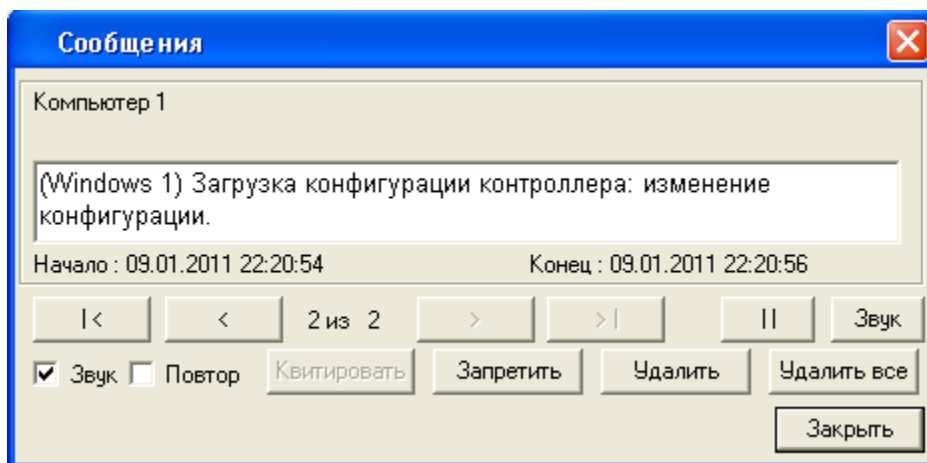


Рисунок 5-31. Окно с сообщением о загрузке конфигурации

Сообщение означает, что конфигурация загружена и программа находится в режиме исполнения. Таким образом, при старте система автоматически проверяет конфигурацию в контроллере и, если она не совпадает с имеющейся в проекте, загружает новую.

Конфигурацию можно также загрузить вручную. Для этого в режиме исполнения нужно выделить в дереве системы нужный контроллер и перейти на закладку Управление. Нажатие на кнопку **Загрузить** загружает конфигурацию.

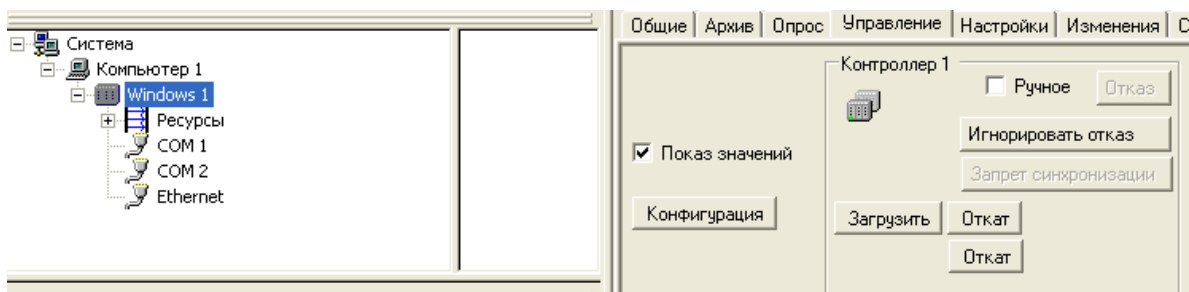


Рисунок 5-32. Вкладка «Управление» контроллера

При использовании настоящего контроллера необходимо помимо создания проекта также провести настройку соединения, связать переменные проекта с входами и выходами контроллера, добавить встроенные и удаленные модули ввода/вывода и при необходимости провести дополнительную настройку: задать архивирование в контроллере, «горячий рестарт», резервирование.

Создадим проект с реальным контроллером. Удалим контроллер **Windows** и добавим контроллер **WinCon**.

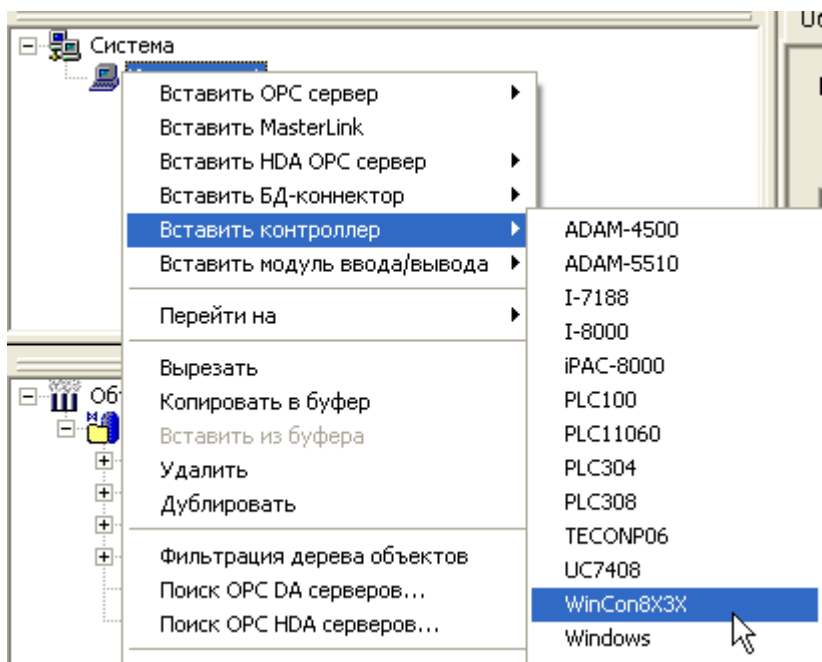


Рисунок 5-33. Добавление контроллера WinCON

Это модульный контроллер с тремя или более слотами расширения. Контроллер имеет процессор Intel Strong ARM 206МГц, 32Мб Flash, 64Мб SRAM, 1xRS232, 1xRS485, Ethernet, установлена операционная система Win CE.NET.

Контроллер появился в дереве системы.

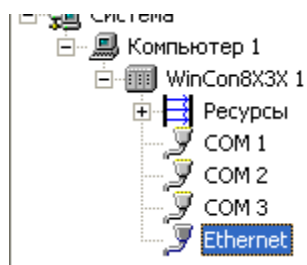


Рисунок 5-34. Контроллер WinCon в дереве системы

5.4.2. Установление связи с MasterSCADA

С контроллером можно установить связь по следующим интерфейсам: **Ethernet**, **RS-232**, **RS-485**, **GSM**.

Ethernet. Этот вид связи с верхним уровнем используется чаще всего – данный интерфейс поддерживает наибольшую скорость обмена и поддерживает топологию «звезда», кроме того, допускается использовать имеющиеся линии связи ЛВС (кабели, концентраторы).

Для того чтобы настроить работу контроллера с верхним уровнем по **Ethernet**, нужно выделить порт в дереве системы и перейти на закладку **Настройки**.

На данной закладке нужно установить галочку **Связь с верхним уровнем** и указать IP в поле **IP-адрес контроллера 1** (Рисунок 5-35). IP-адрес компьютера нужно указывать, только если он резервируется.

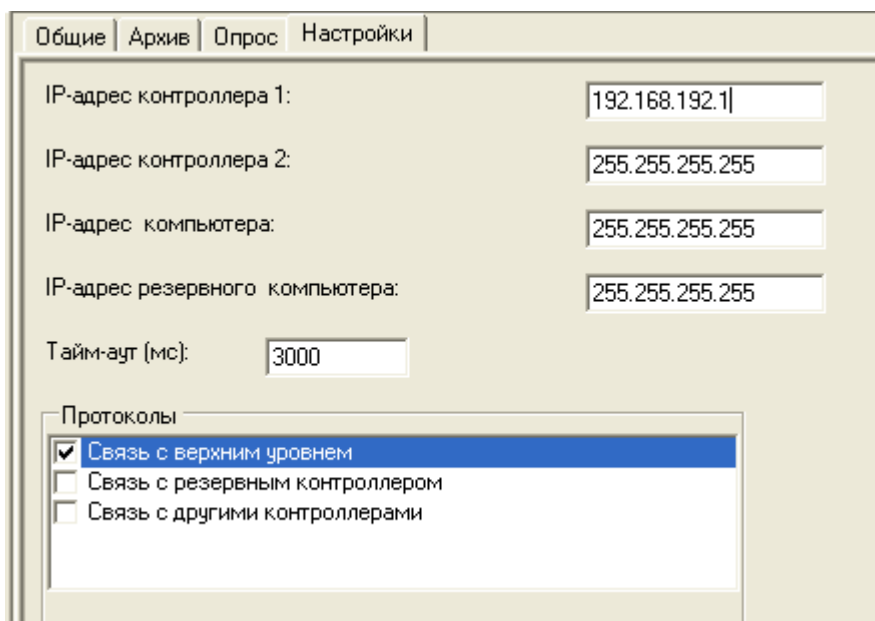


Рисунок 5-35. Настройка связи с верхним уровнем по Ethernet

RS-485. Для того чтобы установить связь по **RS-485**, необходимо выделить в дереве системы COM-порт контроллера, через который планируется установить связь, и перейти на закладку **Настройки**. На закладке **Настройки** в списке **Протокол** нужно выбрать **Связь с верхним уровнем** (Рисунок 5-36).

Далее необходимо задать скорость обмена по интерфейсу, адрес контроллера в сети (если по **RS-485** объединяются несколько контроллеров), номер COM-порта компьютера, к которому подключен преобразователь интерфейса RS-232/RS-485 (USB/RS-485).

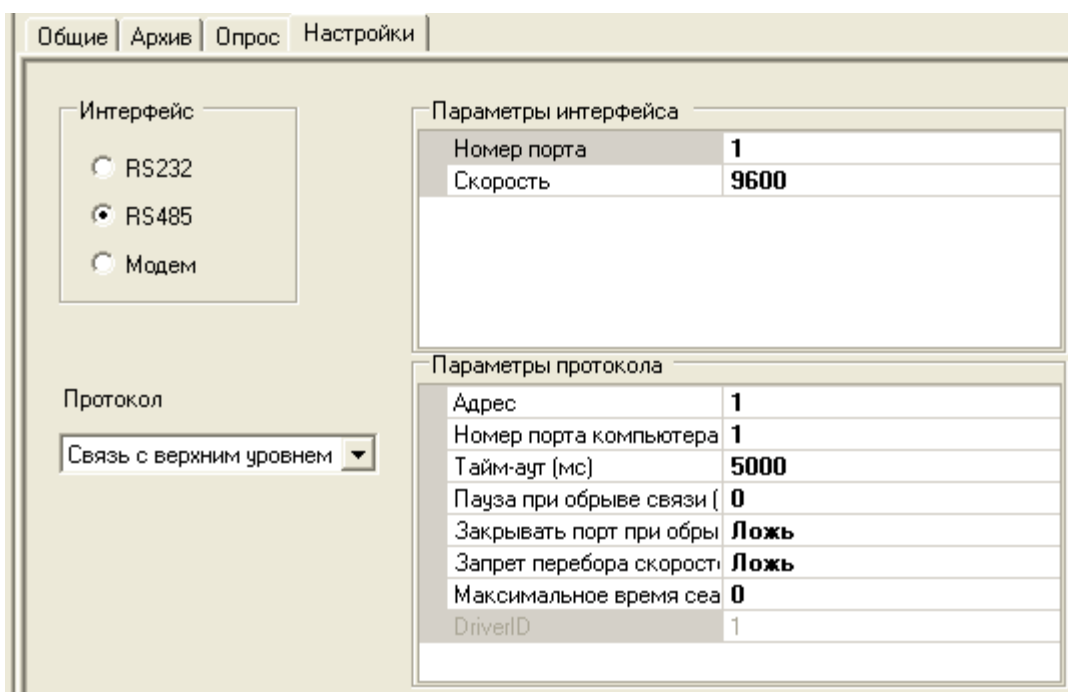


Рисунок 5-36. Настройка связи с верхним уровнем по RS-485

RS-232. Подключение по данному интерфейсу аналогично подключению по **RS-485**. Но, поскольку **RS-232** – это интерфейс типа «точка-точка», контроллер можно подсоединить только один, а значит, значение поля **Адрес** не используется.

GSM. Этот режим используется только при наличии опций в контроллере (**M-PLC-GSM**) и MasterSCADA (**MSRT-MPLC-GSM**). Используется голосовой канал (CSD). В этом режиме компьютер и контроллер должны иметь модемы. Компьютер ведет опрос контроллера – с заданным интервалом делает дозвон и получает данные от контроллера, после чего разрывает соединение. Контроллер может также произвести инициативный дозвон (по событию от программы) и передать данные.

Чтобы создать соединение с верхним уровнем через **GSM**, сначала необходимо добавить модем в компьютер. Модем добавляется на закладке компьютера **Опрос** – нижняя закладка **Модемы** (Рисунок 5-37). На этой закладке необходимо добавить модем и произвести его настройку: задать номер порта компьютера, к которому подключен модем; скорость соединения; номер телефона, соответствующий SIM-карте этого модема; строку инициализации модема (по умолчанию задана для модемов Siemens).

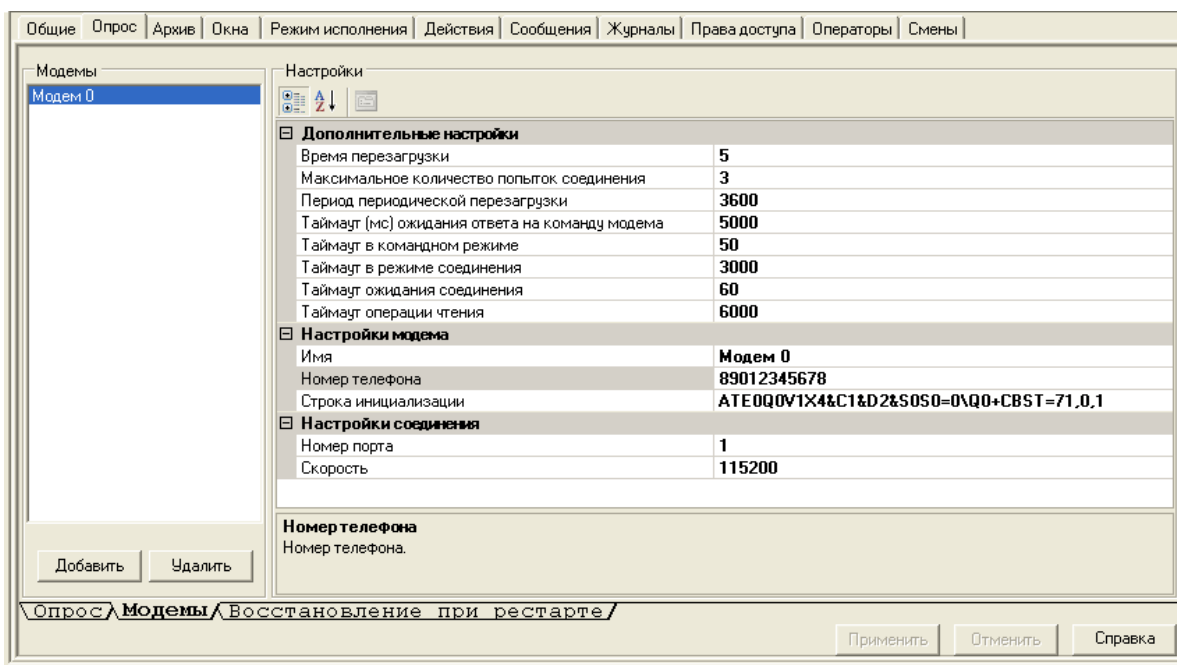


Рисунок 5-37. Добавление модема в компьютер

Настроим модем в контроллере. Для этого на закладке контроллера **Настройки** в разделе **Протоколы** нужно включить использование GSM-модема.

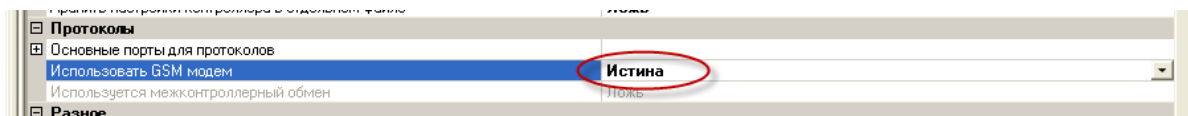


Рисунок 5-38. Включение использования модема в контроллере

У контроллера в ресурсах появилась группа **Модем**.

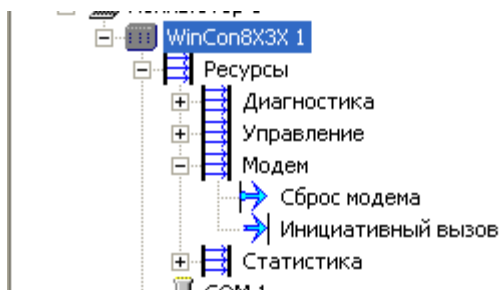


Рисунок 5-39. Ресурсы модема в контроллере

Группа имеет выход **Сброс модема**, при помощи которого можно определить, что модем находится в состоянии перезапуска, а также вход **Инициативный вызов** – для выполнения инициативного дозвона.

Теперь нужно выделить COM-порт, к которому будет подключен модем контроллера. На закладке **Настройки** нужно выбрать интерфейс **Модем**, протокол – **Связь с верхним уровнем**.

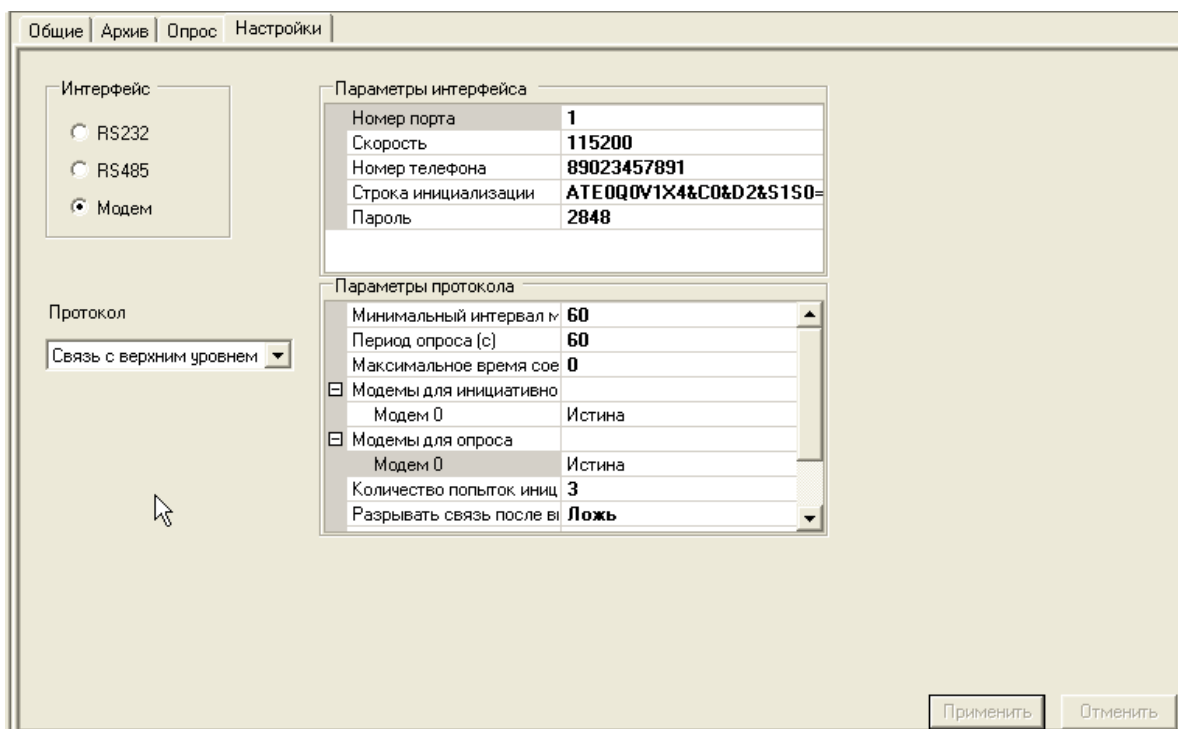


Рисунок 5-40. Настройка связи с верхним уровнем в контроллере через модем

Далее нужно задать параметры соединения: скорость, номер телефона данного модема и строку инициализации.

В параметрах протокола нужно задать **Период опроса** (с данной периодичностью будет устанавливаться связь с контроллером), а также выбрать, какой модем компьютера будет производить опрос контроллера (поле **Модем для опроса**) и на какой модем будет звонить контроллер в случае инициативного вызова (поле **Модем для инициативного вызова**). Допускается использовать один и тот же модем для опроса и инициативного вызова, хотя для надежного дозвона рекомендуется использовать отдельные модемы.

Канал GSM также можно использовать для загрузки конфигурации в контроллер.

5.4.3. Установление связи с модулями ввода/вывода

Контроллер взаимодействует с объектом при помощи датчиков и исполнительных механизмов, подключаемых к входам и выходам. Моноблочные контроллеры имеют

определенный набор встроенных входов и выходов – в MasterSCADA они появляются в дереве системы автоматически при добавлении контроллера. Кроме того, современные контроллеры поддерживают увеличение количества входов и выходов при помощи модулей ввода/вывода.

Модульные контроллеры имеют внутреннюю шину расширения, при помощи которой наращивается количество модулей. При этом модули используются уникальные, собственной разработки фирмы – производителя контроллера.

К моноблочным и модульным контроллерам могут подключаться модули по имеющимся у контроллера сетевым или последовательным интерфейсам (**RS-485** или **RS-232**), с использованием универсальных протоколов: **Modbus**, **DCON** или других. В этом случае контроллер и модули расширения могут быть разработками различных фирм. Кроме того, по этим интерфейсам могут подключаться интеллектуальные приборы и исполнительные механизмы: счетчики, измерители, регуляторы и т. д.

MasterPLC поддерживает расширение и по внутренней шине, и по стандартным протоколам. Поддерживаются открытые протоколы: **Modbus (RTU и TCP)** и **DCON**, а также ряд проприетарных (частнофирменных) протоколов для взаимодействия с конкретными моделями приборов: счетчиками электроэнергии «Меркурий-230», счетчиками тепла, электроэнергии и газа «Логика», счетчиками импульсов «Пульсар» и др.

Добавление модуля на шину расширения. На внутреннюю шину расширения устройство добавляется, только если контроллер является модульным.

Добавим модуль в контроллер WinCon. Модуль добавляется через контекстное меню.

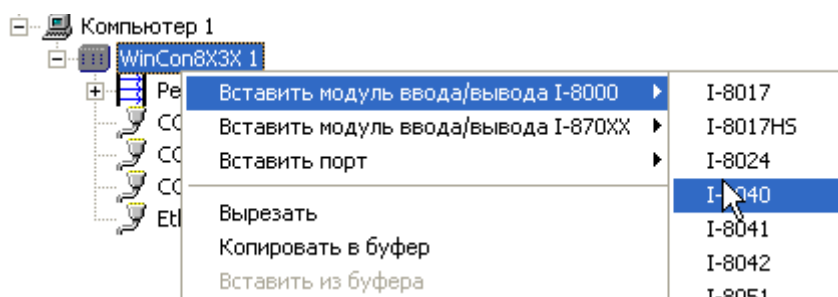


Рисунок 5-41. Добавление встраиваемых модулей ввода/вывода

После этого модуль появится в дереве системы.

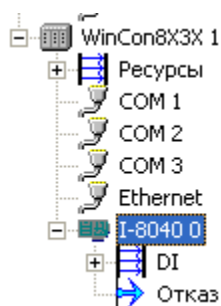


Рисунок 5-42. Модуль в дереве системы

На выходе **Отказ** формируется значение **Вкл**, в случае если с модулем не удается установить связь.

Настройка модуля сводится к заданию его номера на шине. Это делается на закладке **Настройка**.

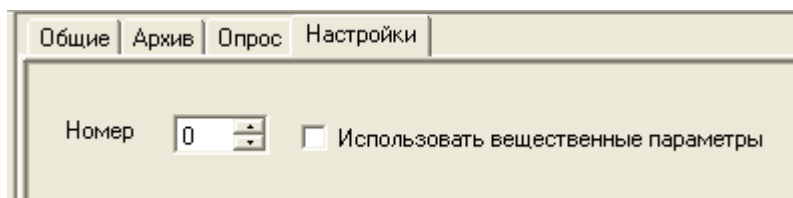


Рисунок 5-43. Настройка модуля

Добавление модуля на интерфейс RS-485 (RS-232). В этом случае добавление модуля происходит через добавление на COM-порт контроллера в дереве системы. Добавим модуль, работающий по интерфейсу **RS-485** с протоколом **Modbus RTU**. Выделим порт **COM2** и перейдем на закладку **Настройка**. Как и в случае настройки взаимодействия с верхним уровнем, нужно задать параметры обмена по интерфейсу: скорость, четность, стоп-биты. Далее в раскрывающемся списке **Протокол** выбирается протокол обмена – **Modbus RTU**.

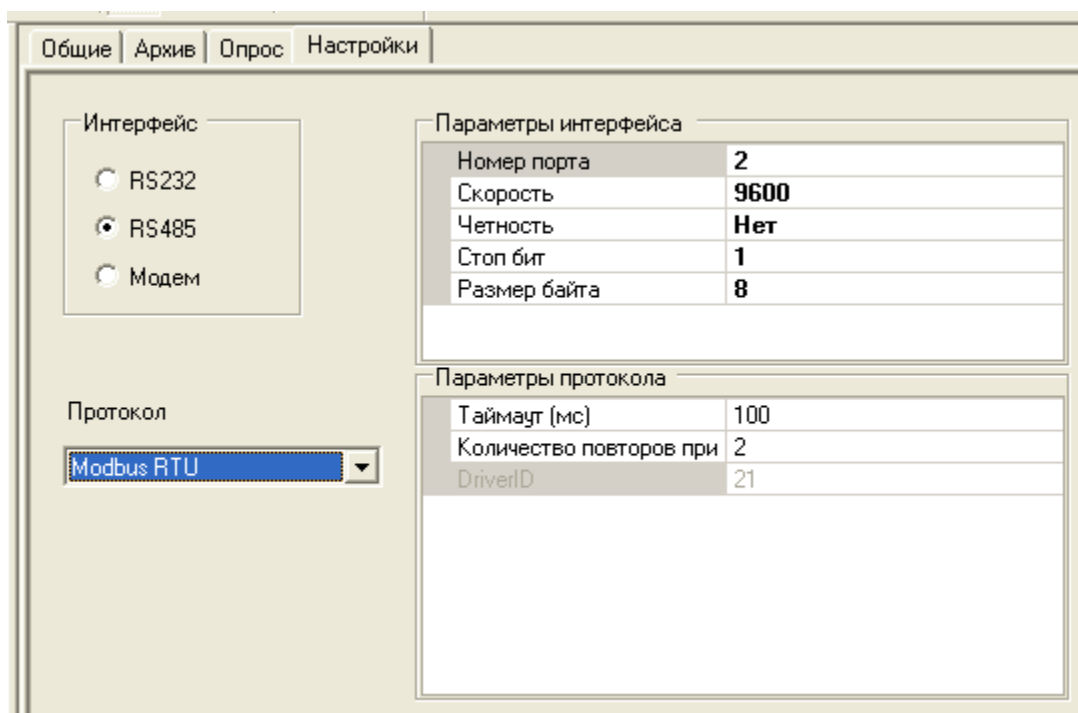


Рисунок 5-44. Настройка COM-порта для обмена по Modbus RTU

Также настраиваются **Таймаут** (время ожидания ответа) и **Количество повторов запроса**. Настройка обмена завершена, теперь можно добавлять модули ввода/вывода.

Модуль (устройство) добавляется через контекстное меню COM-порта (в данном случае **COM2**).

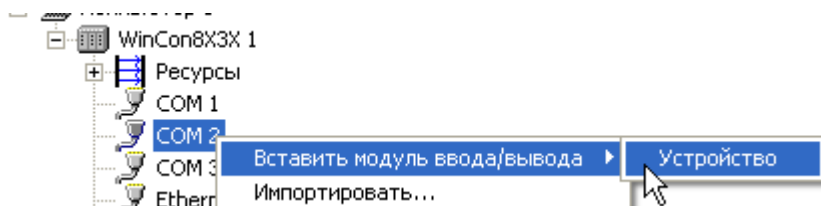


Рисунок 5-45. Добавление устройства

На выходе **Отказ** формируется значение **Вкл**, в случае если с модулем не удастся установить связь.

У устройства на закладке **Настройки** можно задать адрес, который имеет устройство, таймаут, включить групповые запросы.

Общие Архив Опрос Свойства	
Таймаут (мс)	0
Адрес устройства	1
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	True

Рисунок 5-46. Настройки устройства

Теперь в устройство можно добавить нужные регистры для чтения и записи. Это делается через контекстное меню устройства.

О предназначении (функции **Modbus**) каждого регистра можно прочитать в справке.

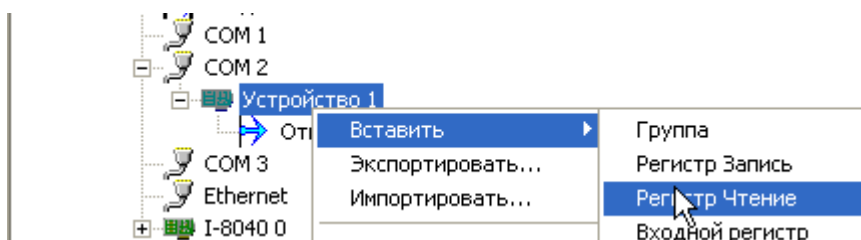


Рисунок 5-47. Добавление регистров

Настройка адреса и типа значения производится на закладке **Свойства** регистра.

Общие Свойства Опрос выхода Архив Контроль значений Паспорт Данные Права доступа Список связей	
Адрес ячейки	0
Тип значения	Целый
Тип ячейки	РегистрХранения

Рисунок 5-48. Настройка регистра

Имена устройств, регистров можно изменять.

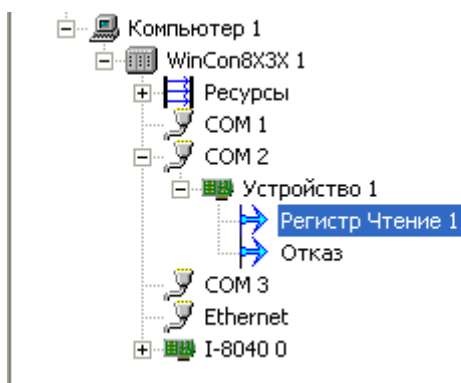


Рисунок 5-49. Регистр в дереве системы

Добавление устройств, работающих на других протоколах, осуществляется аналогичным образом. Подробнее об этом можно прочитать в справочной системе.

Добавление модуля на интерфейс TCP/IP (протокол Modbus TCP).

В контроллеры, имеющие интерфейс *Ethernet* (а также *MasterLink* и *Windows-контроллер*), можно добавлять устройства, работающие по протоколу *Modbus TCP*.

Для этого нужно вызвать контекстное меню контроллера, выбрать пункт **Вставить порт – Виртуальный порт**.

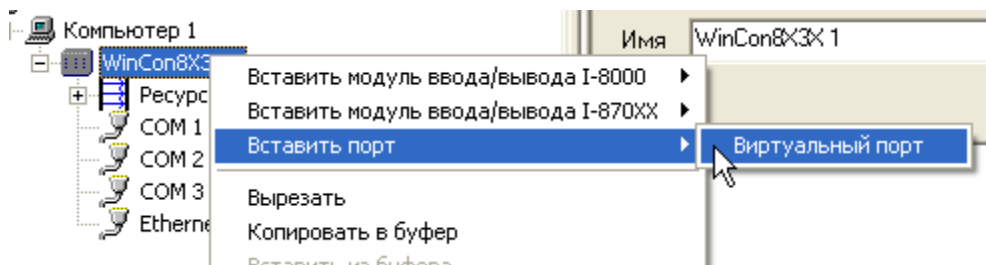


Рисунок 5-50. Добавление виртуального порта

Виртуальный порт добавится в контроллер. На закладке **Настройки** выберите протокол – *Modbus TCP*.

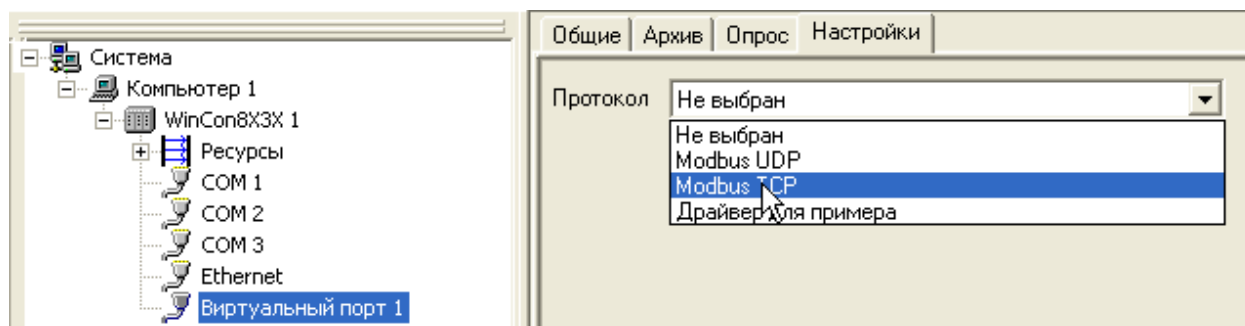


Рисунок 5-51. Настройка порта на протокол Modbus TCP

Теперь в данный порт можно добавлять устройства.

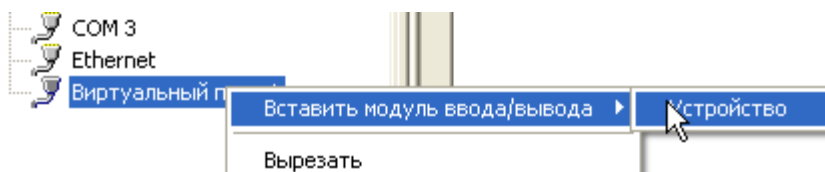


Рисунок 5-52. Вставка Modbus TCP устройства

Настройка устройства, в том числе его IP-адреса, осуществляется на закладке **Свойства**. Добавление и настройка регистров осуществляются аналогично тому, как это делается для интерфейса *RS-485*.

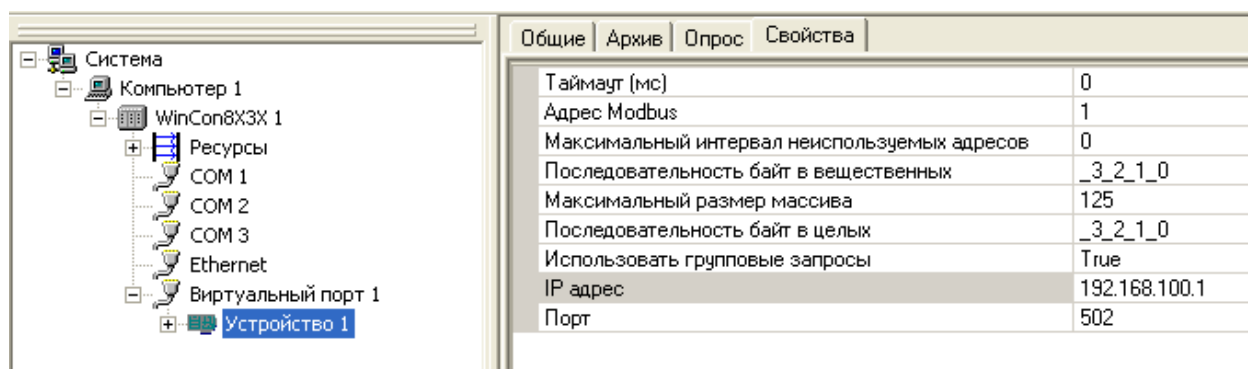


Рисунок 5-53. Настройки Modbus TCP устройства

5.4.4. Расширенные возможности MasterPLC

Контроллер с исполнительной системой **MasterPLC** также имеет ряд дополнительных возможностей.

Архивирование в контроллере – по умолчанию контроллер передает в MasterSCADA текущие данные. Но при необходимости можно включить архивацию переменных прямо в контроллере. При этом данные будут писаться в оперативную или flash-память, а SCADA при каждом цикле опроса будет забирать их в собственный архив. Данная возможность позволяет архивировать быстро изменяющиеся значения или избегать потери данных при работе с неустойчивыми каналами связи (радио, GSM, Wi-Fi и т. д.).

Горячий рестарт – позволяет сохранять состояние переменных в памяти и при старте загружать их, что позволяет продолжить работу программы с того же места. При этом есть возможность задания актуальности «горячего рестарта»: если после выключения прошло слишком много времени, то значения будут сброшены.

Межконтроллерный обмен – если в системе используется несколько контроллеров и данные одного контроллера требуются другому, то есть возможность включить прямой межконтроллерный обмен данными, без участия SCADA как буфера.

Резервирование контроллеров – для создания высоконадежных систем есть возможность использовать резервирование контроллеров.

Данный функционал системы **MasterPLC** подробно описан в справочной системе – в разделе «Контроллеры».

6. Редактор мнемосхем MasterGraf

Основным способом представления информации оператору является графический интерфейс – мнемосхемы и окна. Мнемосхемы предоставляют оператору, управляющему технологическим процессом, графический интерфейс для наглядного динамического отображения хода технологического процесса и выдачи управляющих воздействий с помощью виртуальных органов управления.

Мнемосхемы создаются из элементов, которые поставляются вместе с программой, приобретаются отдельно или создаются самим пользователем. Это – библиотеки графических примитивов, объемных фигур, элементов создания диалогов, индикаторов значений переменных и графиков, а также элементы технологического оборудования (насосы, задвижки и т.п.). Наряду с уже имеющимися элементами можно без какого-либо программирования использовать любые элементы управления (ActiveX-контролы), приобретенные отдельно или загруженные из Интернета. Пользователь может самостоятельно создавать новые элементы, группируя и настраивая уже существующие, либо написав новый элемент на любом языке программирования.

MasterGraph позволяет сделать динамическим любое свойство любого элемента, то есть сопоставить его поведение с изменением технологического параметра. Для создания сложных видов отображения технологической динамики имеется встроенный редактор мультфильмов.

Мнемосхема может быть намного больше экрана, а сложность взаимосвязей различных мнемосхем и открывающихся поверх них окон практически ничем не ограничена. Способы изображения динамики объекта, выбор способов управления практически тоже зависят только от фантазии разработчика мнемосхемы. При этом крайне важно эргономически грамотное проектирование пользовательского интерфейса – старайтесь избегать нагромождения графики, не применять анимацию без надобности, правильно подбирать цвета.

6.1. Создание мнемосхем и окон

Мнемосхемы и окна создаются на уровне объекта, на закладке **Окна**. По умолчанию доступны следующие типы окон: **Мнемосхема**, **Окно управления**, **Окно объекта**, **Рапорт**, **Рецепт**, **Изображение объекта**. Все окна, кроме **Изображение объекта**, предоставляют одинаковый функционал и отличаются только способами открытиями и размерами. Также можно создавать собственные окна – на закладке **Окна** элемента **Система**.

Изображение объекта предоставляет дополнительную возможность помимо возможности открыть его как отдельное окно: если добавить изображение объекта на мнемосхему (вытащив правой кнопкой мыши объект), то будет добавлен символ. Символ будет содержать все элементы, расположенные на изображении объекта, и кнопку вызова – то есть будет способен отображать состояние элементов, а также позволит выполнять переход к другому окну. Подробнее про изображение объекта будет рассказано в отдельной главе.

6.2. Окно редактора

Окно программы (независимо от типа создаваемого окна) в режиме разработки состоит из следующих частей:

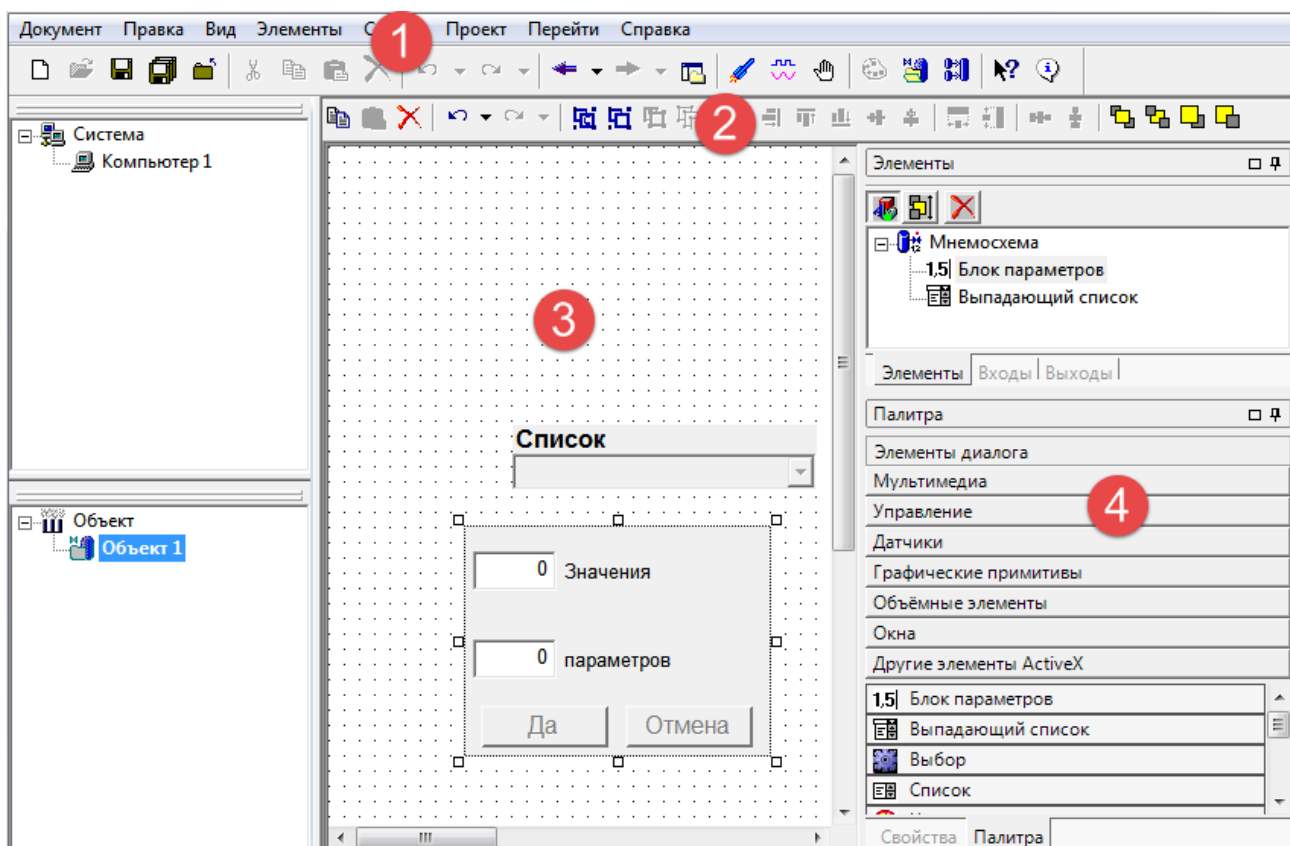


Рисунок 6-1. Внешний вид редактора

1. Меню.
2. Панель инструментов.
3. Рабочая область.
4. Панели настройки: палитра, элементы, входы, выходы, свойства.

Создание (рисование) и редактирование мнемосхемы производятся в рабочей области главного окна.

6.3. Добавление элементов из дерева объектов

Элементы могут добавляться на мнемосхему двумя способами: перетаскиванием элементов (команд, расчетов, событий, ФБ) на мнемосхему, а также добавлением элементов из палитры мнемосхемы, при необходимости их можно связать с элементами дерева объектов через входы и выходы динамизации.

Рассмотрим добавление элементов из дерева объектов. Элементы дерева объектов имеют контролы (или даже несколько контролов) для отображения на мнемосхеме. Чтобы

добавить контроль элемента на мнемосхему, достаточно перетащить элемент мышью в поле мнемосхемы. Для примера создадим в дереве значение и вытащим его на мнемосхему левой клавишей мыши.

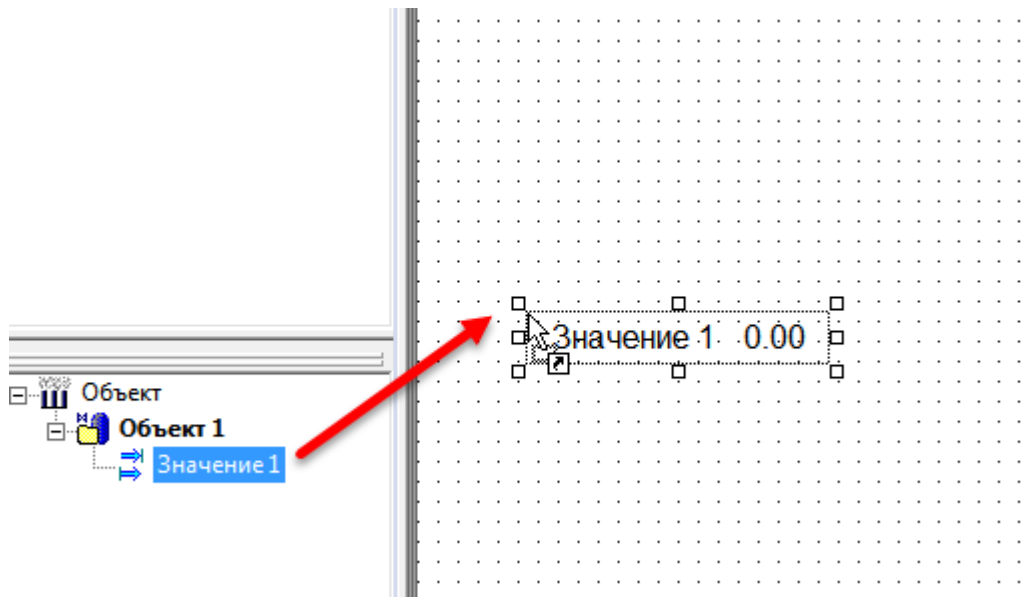


Рисунок 6-2. Добавление переменных на мнемосхему

На мнемосхеме создан элемент **Значение 1**, в режиме исполнения в нем будет отображаться текущее значение элемента.

Снова вытащим элемент, но теперь правой кнопкой мыши (Рисунок 6-3). Поскольку элемент **Значение 1** имеет несколько контролей, появилось всплывающее окно, в котором мы можем выбрать, какой именно контроль хотим добавить.

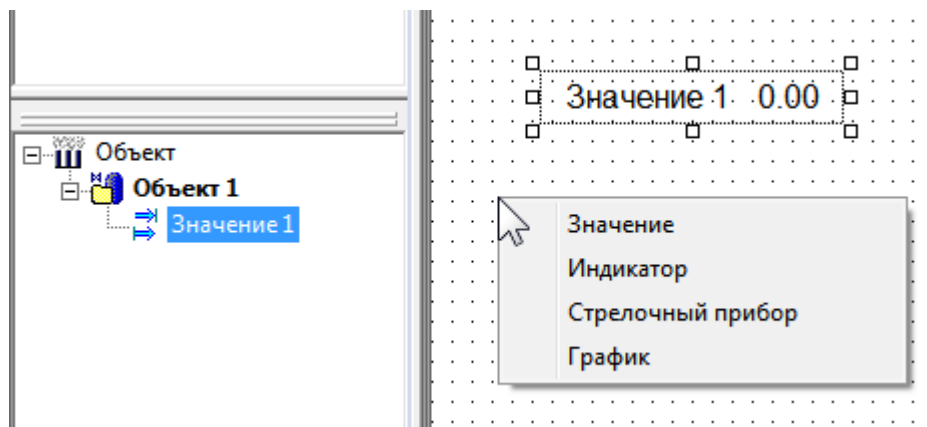


Рисунок 6-3. Добавление переменных с выбором контроля

Добавьте самостоятельно все контролы типа значение и посмотрите их поведение их в режиме исполнения.

Элементы расчет, событие и команда также имеют свой набор контролов. При этом контролы команды поддерживают ввод значений оператором. В зависимости от типа данных команды контрол может принимать вид кнопки, редактируемого значения или слайдера.

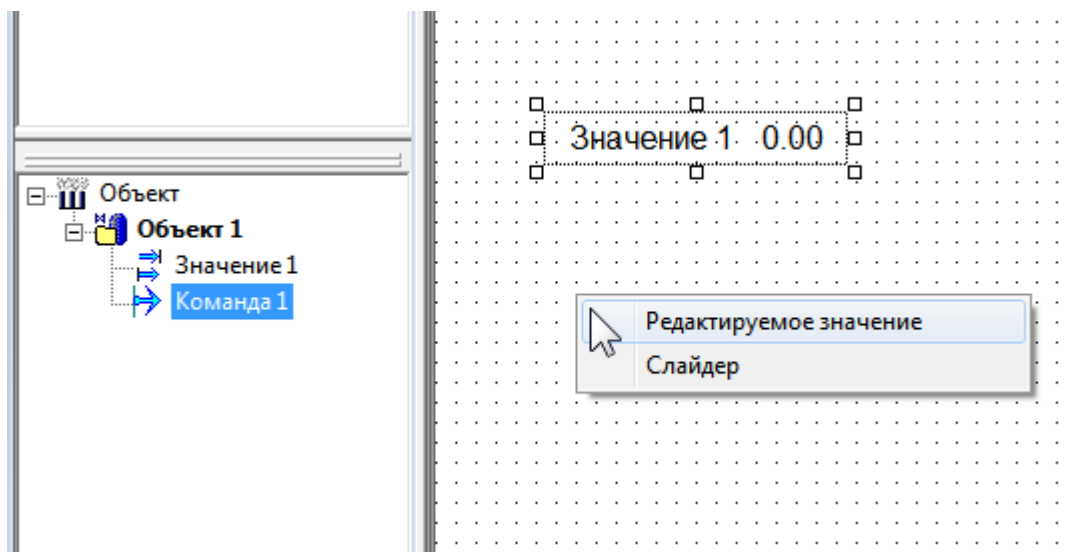


Рисунок 6-4. Варианты контрола команды

Теперь рассмотрим добавление на мнемосхему функциональных блоков. В MasterSCADA блоки делятся на функциональные блоки (ФБ) и визуальные функциональные блоки (ВФБ). Их отличие в том, что ВФБ могут иметь отображение на мнемосхеме.

Для примера добавим в дерево объектов ВФБ Насос мнемонический из раздела палитры Исполнительные механизмы, а затем добавим его на мнемосхему.

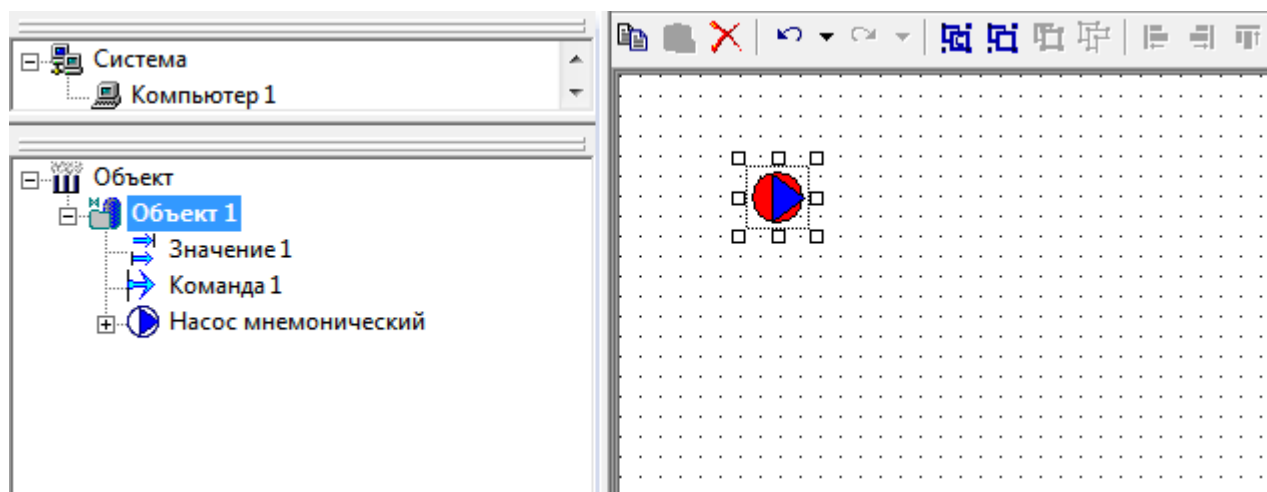


Рисунок 6-5. Контрол ВФБ «Насос мнемонический»

В режиме исполнения контрол будет менять свое состояние в зависимости от внешних факторов (состояний входов, действий пользователя и т. д.). Вы можете самостоятельно изучить поведение данного ВФБ при различных настройках и внешних сигналах. Также вы всегда можете вызвать справку к функциональному блоку через его контекстное меню.

Визуальные функциональные блоки также находятся в разделах палитр **Аппараты**, **Управление** и **Отраслевые библиотеки**.

6.4. Добавление элементов из палитры мнемосхемы

Теперь рассмотрим добавление элементов из палитры мнемосхемы. Для этого сначала необходимо перейти на специальную вкладку в редакторе мнемосхемы, выбрать нужный элемент из любого раздела нажатием мыши, а затем разместить его в поле мнемосхемы.

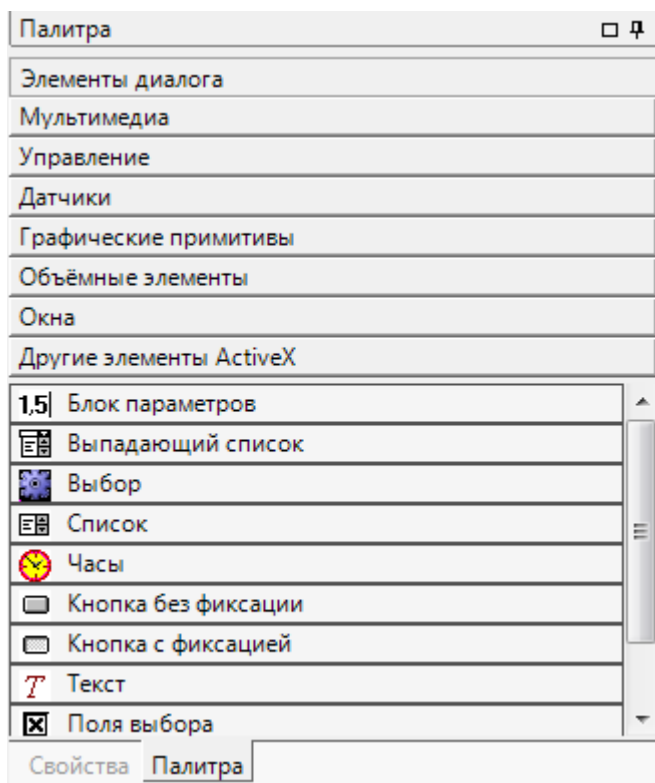


Рисунок 6-6. Палитра мнемосхемы

Для этого выберем соответствующий раздел. Для примера добавим элемент **Выбор изображения** из раздела **Мультимедиа**. Данный контроль предназначен для отображения рисунков на мнемосхеме, при этом номер отображаемого в текущий момент рисунка можно быстро менять. Так с помощью данного контроля можно отображать состояние исполнительных механизмов, активности тех или значений и т. д.

Для задания количества рисунков и конкретных файлов откроем окно свойств через контекстное меню контроля (Рисунок 6-7).

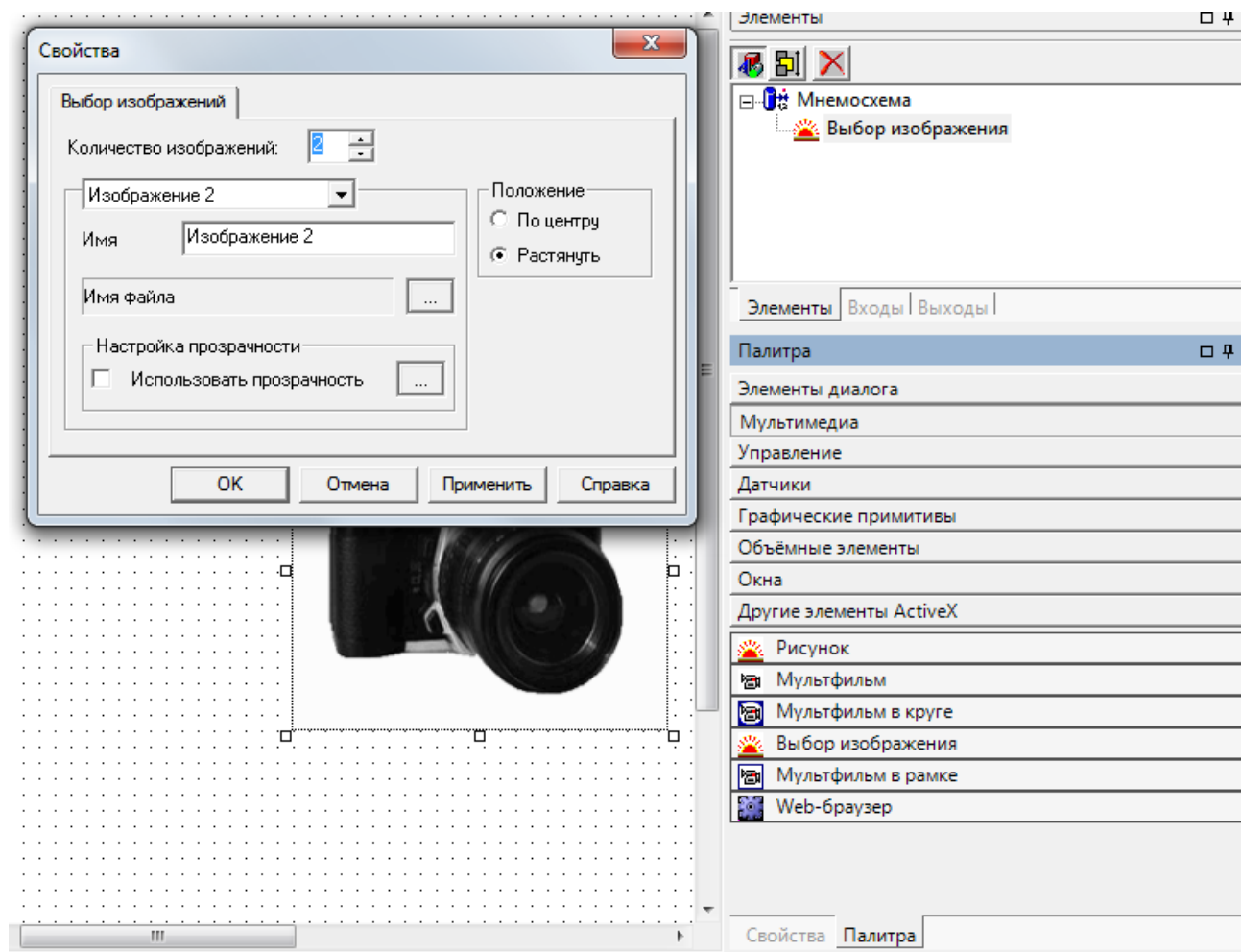


Рисунок 6-7. Окно настройки «Выбор изображения»

Зададим два рисунка и выберем нужные. После нажатия кнопки **Применить** первый в списке рисунок будет отображаться в контроле (Рисунок 6-8).

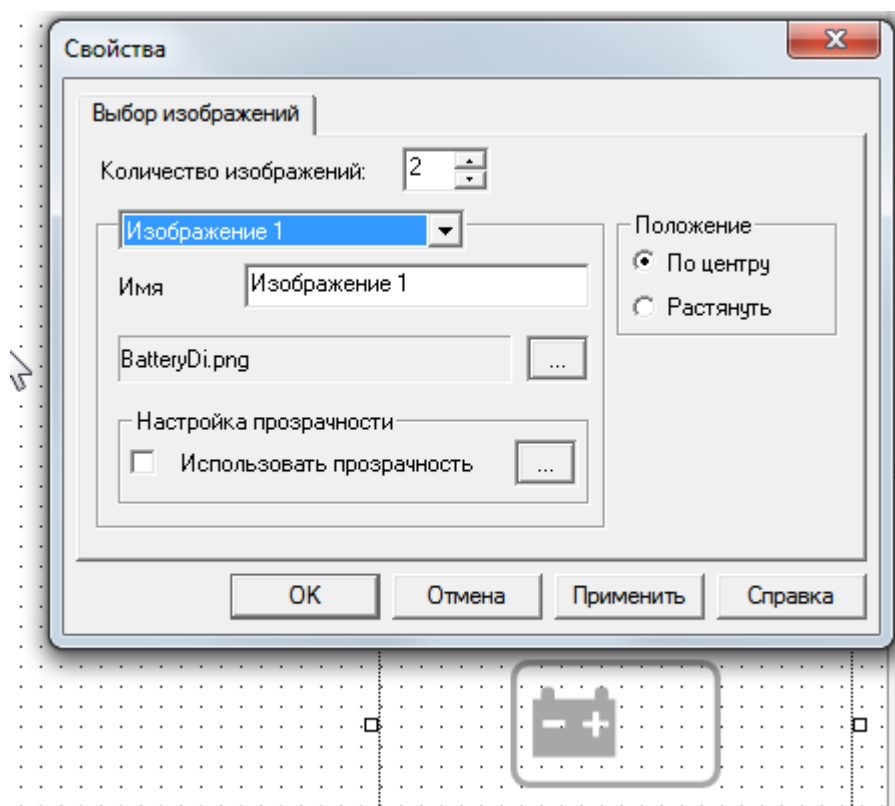


Рисунок 6-8. Назначение рисунков

Как теперь управлять данными рисунками – сменять их в зависимости от внешних состояний? Для этого предусмотрены входы и выходы динамизации – они предназначены для установления связи между свойствами контроля и элементами дерева объектов. Список доступных входов и выходов динамизации элемента можно увидеть, нажав специальные кнопки в верхней части его панели свойств (Рисунок 6-9).

Рассмотрим управление номером рисунка в контроле **Выбор изображения** через вход динамизации. Пусть в нашем случае, если событие в дереве имеет значение **Выкл**, то номер рисунка должен быть **0** (нумерация в контроле идет с нуля), а если **Вкл** – то **1**.

Выделим наш контроль. В верхней части панели свойств контроля есть кнопки входов и выходов динамизации, представленные в виде синих стрелок.

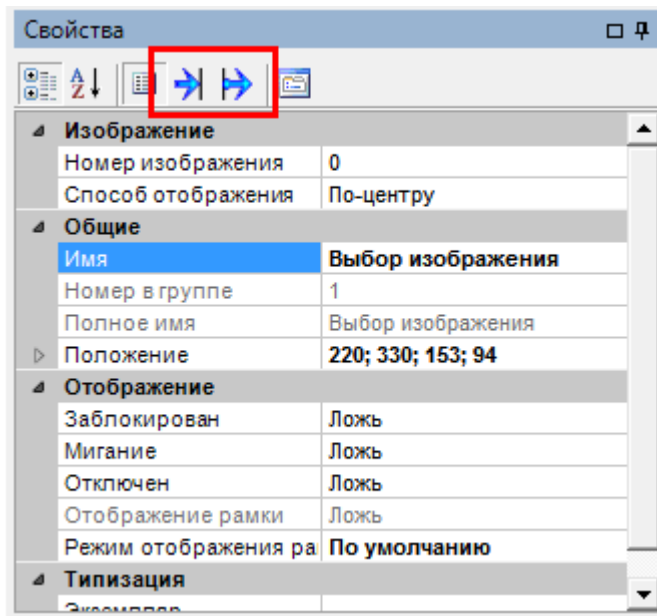


Рисунок 6-9. Кнопки входов и выходов динамизации

Нажмем на кнопку входов динамизации. В таблице будут представлены все свойства данного контроля с доступными для динамизации свойствами. Нас интересует свойство **Номер изображения**.

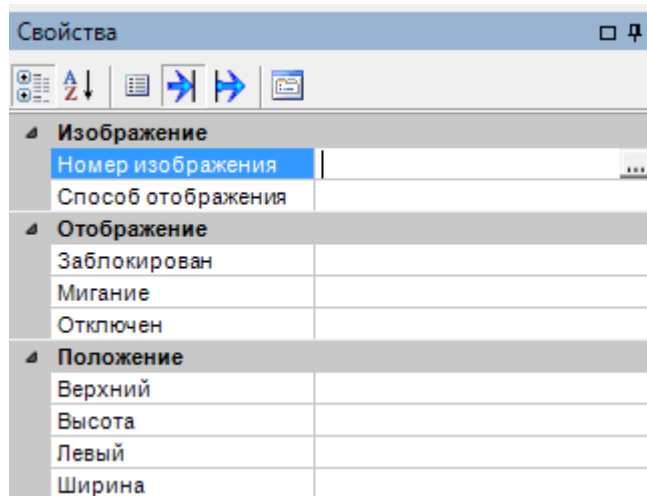


Рисунок 6-10. Вход динамизации для управления номером изображения

Свяжем данное свойство с событием. Для этого перетащим левой кнопкой мыши событие в поле **Свойства**.

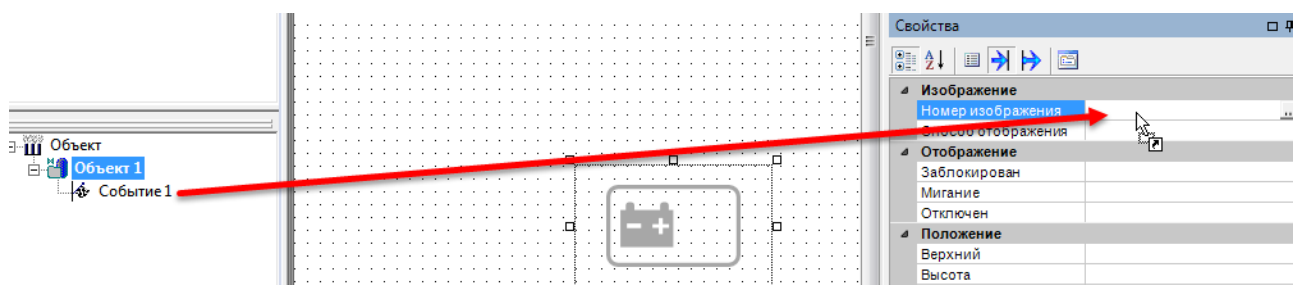


Рисунок 6-11. Привязка входа динамизации к событию

Связь установлена. Теперь необходимо произвести настройку динамизации свойства – нажмем на кнопку с тремя точками, которая появилась в поле.

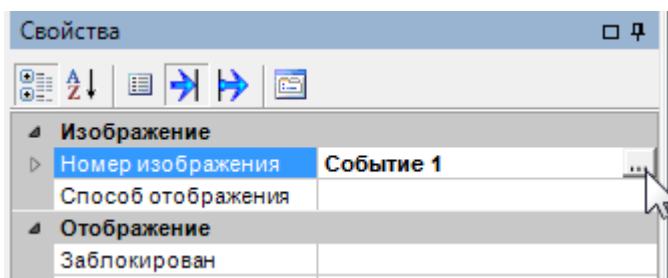


Рисунок 6-12. Открытие окна настройки динамизации

В окне динамизации свойства можно задать соответствие входным значениям из дерева объектов и значениям контроля (Рисунок 6-13). В нашем случае, если входное значение – **Выкл**, то ему соответствует значение **0**, если **Вкл** – то **1**.

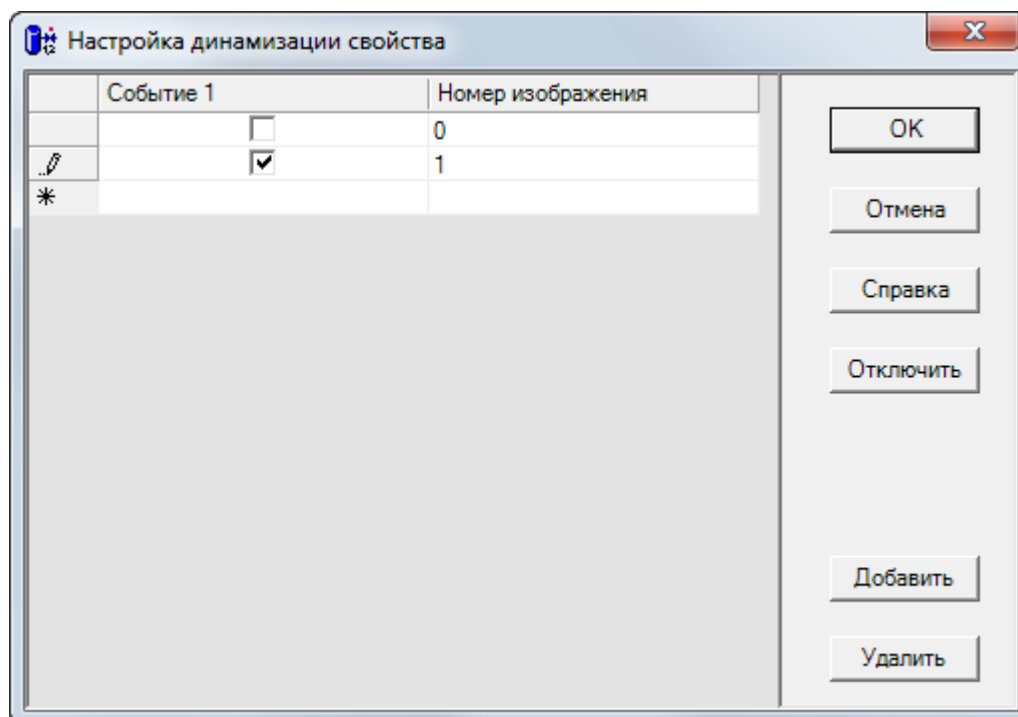


Рисунок 6-13. Окно настройки динамизации

Настройка может быть и более сложной – это полностью зависит от ваших значений и выбранного вами контроля.

Запустим режим исполнения. Теперь при изменении значения события меняется и рисунок на мнемосхеме.

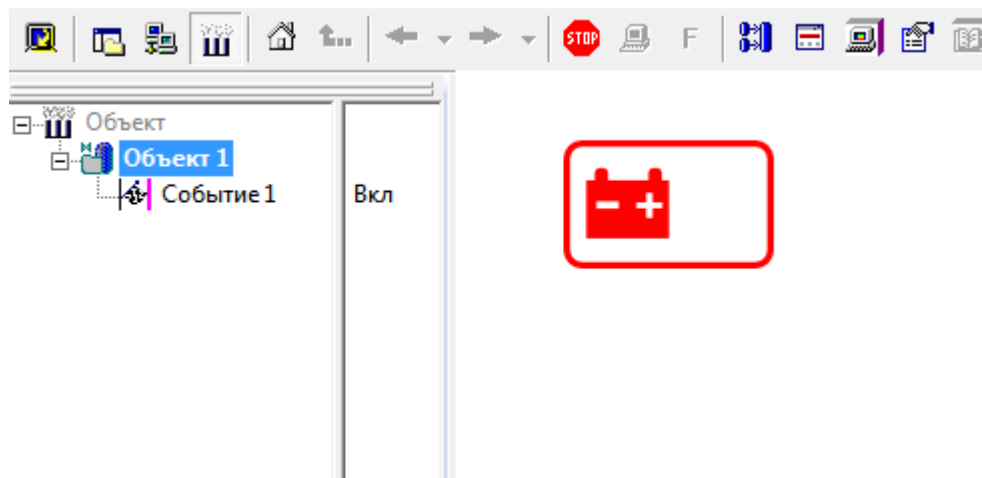


Рисунок 6-14. Изменение рисунка в режиме исполнения

Рассмотрим еще один контроль, в котором будут использоваться как входы, так и выходы динамизации. Для примера рассмотрим элемент **Блок параметров** из раздела **Элементы диалога**.

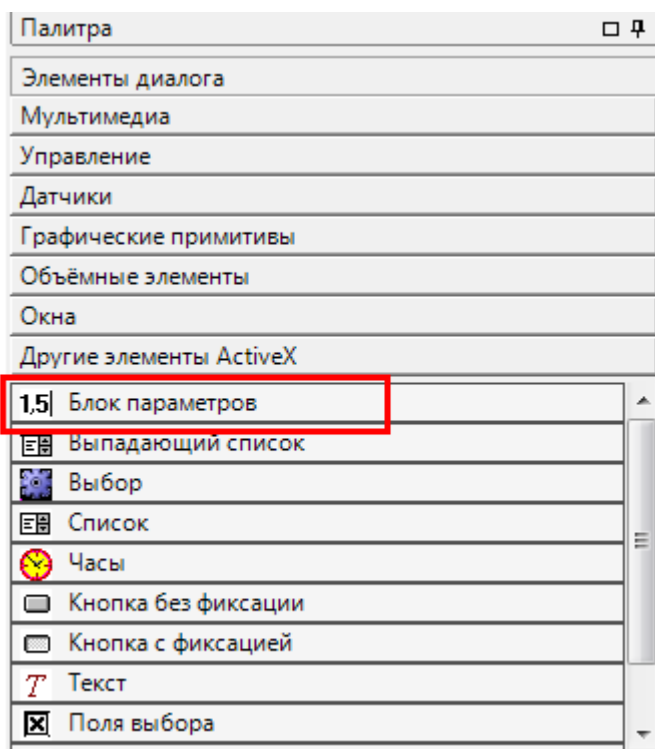


Рисунок 6-15. Добавление «Блока параметров»

Данный контроль предназначен для ввода значений в текстовые поля, максимальное количество полей в одном контроле – до 10.

Добавление полей производится на закладке **Строки** в окне свойств, которое можно вызвать через контекстное меню.

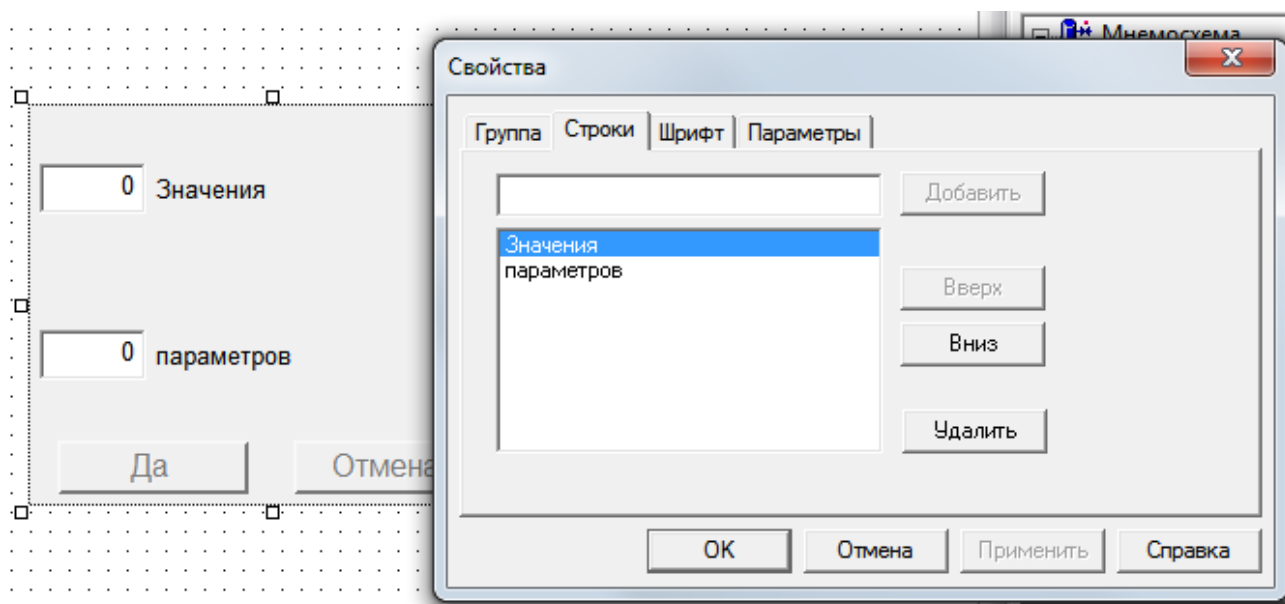


Рисунок 6-16. Настройка контроля «Блок параметров»

Рассмотрим процедуру связывания данного контроля с деревом объекта.

Данный контроль имеет набор входов и выходов динамизации. Значения, вводимые оператором, поступают на выходы динамизации контроля **Значение 1 – Значение 10**. На входы динамизации с аналогичными названиями можно подать значение, тогда это значение отобразится в контроле.

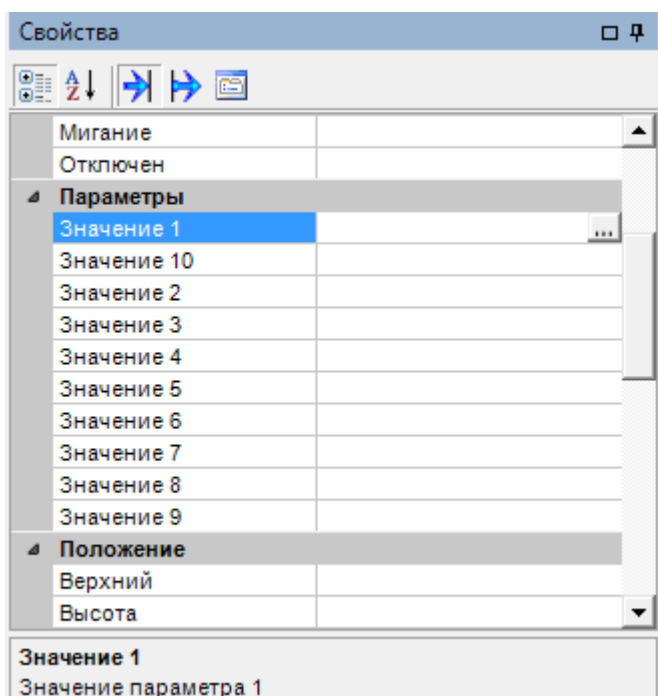


Рисунок 6-17. Основные входы динамизации «Блока параметров»

Наиболее удобный вариант связывания данного контрола – привязать вход и выход динамизации Значение N к одной и той же команде. Прделаем данные действия.

Настроим наш контрол на ввод трех значений – добавим три поля ввода в контрол.

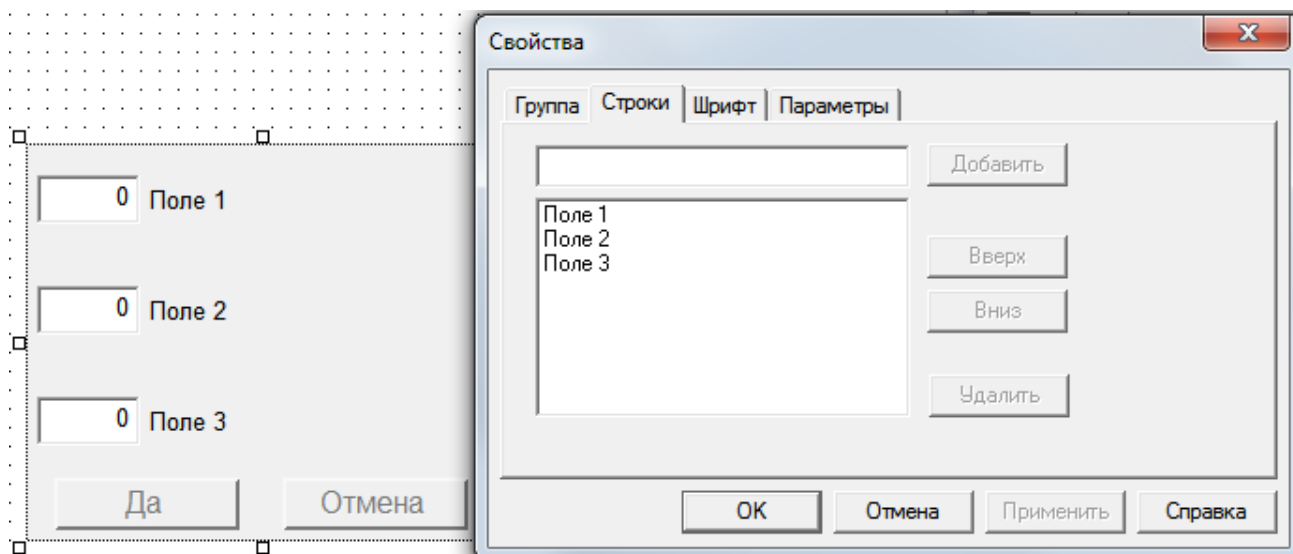


Рисунок 6-18. Настройка контрола «Блок параметров»

Добавим в дерево объектов три команды типа вещественный. Перетащим каждую команду в соответствующий выход динамизации Значение.

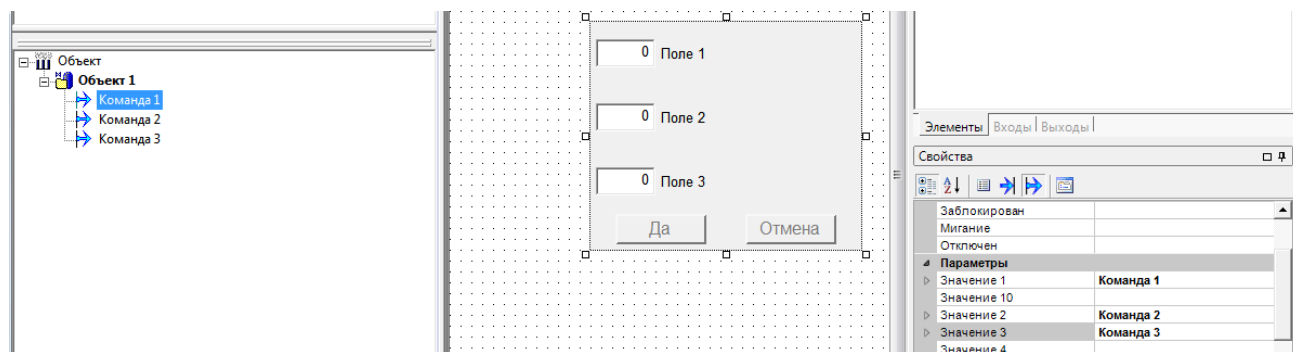


Рисунок 6-19. Связывание выходов динамизации и команд

Теперь перетащим эти же команды на входы динамизации Значение.

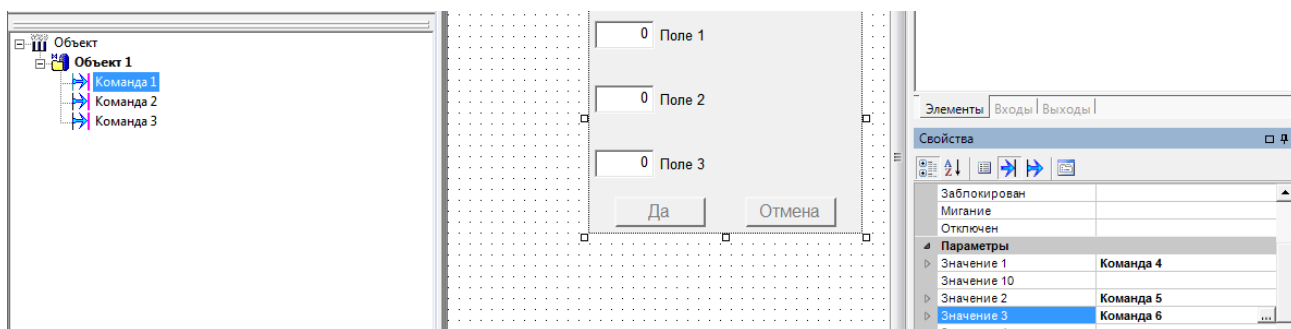


Рисунок 6-20. Связывание входов динамизации и команд

Теперь вводимые оператором значения будут поступать в команду, и затем их можно передать к другим элементам дерева объектов или дерева системы.

Если в вашей команде используется обратная связь, то есть значение как поступает в OPC-переменную, так и возвращается обратно (например, если ввод уставки может производиться как из SCADA-системы, так и локально с панели управления прибора), то в этом случае необходимо у команды на закладке **Опрос** снять галочку **Не передавать данные обратной связи**. Если этого не сделать, то значения, поступающие по обратной связи, в контроле **Блок параметров** отображаться не будут.

Остальные контролы из раздела палитры **Элементы диалога** настраиваются аналогично – с использованием команды.

Контролы остальных разделов предназначены для решения других задач. В разделе **Графические примитивы** находятся контролы для рисования прямоугольников, кругов, эллипсов, линий и т. д.

В разделе **Мультимедиа** находятся контролы для работы с мультимедийными файлами (с рисунками, мультфильмами). При этом контролы **Мультфильм в рамке** и **Мультфильм в круге** имеют встроенный редактор, позволяющий создавать анимацию из набора рисунков или анимировать статичный рисунок по определенному закону. В разделе **Объемные элементы** находятся контролы для создания 3D-изображений трубопроводов, эллипсов, цилиндров и конусов. Элементы перечисленных групп предназначены для оформления мнемосхем и внесения в них динамизации.

К каждому контролу палитры мнемосхемы есть подробное описание в справочной системе. Также на прилагаемом диске находится отдельная документация по динамизации графических контролов в MasterSCADA.

6.5. Создание оконного интерфейса

В большинстве случаев одного окна в интерфейсе недостаточно, поэтому необходим многооконный интерфейс, а иногда даже «умный интерфейс» – чтобы нужные окна открывались самостоятельно (например, при возникновении аварии открывалась мнемосхема с проблемным участком).

В MasterSCADA существует три вида окон:

Документ – в данном режиме мнемосхема занимает всю рабочую область MasterSCADA (полноэкранный режим). В один момент времени может быть открыт только один **Документ**.

Окно – представляет собой мнемосхему в виде плавающего окна (заголовок и рамку при необходимости можно выключать). Если из окна открыть еще одно **Окно**, то старое **Окно** закрывается.

Новое окно – также представляет собой мнемосхему в виде плавающего окна. Отличие от **Окна** в том, что при открытии нового окна старое не закрывается.

6.5.1. Открытие через кнопки вызова

Для открытия окон и переходов между документами используются кнопки вызова. Создавать их очень просто – нужно просто вытащить нужный объект на мнемосхему. Если вытаскивать объект правой кнопкой мыши, то появится раскрывающийся список со всеми документами данного объекта – из этого списка можно выбрать, какой документ будет открывать создаваемая кнопка (Рисунок 6-21).

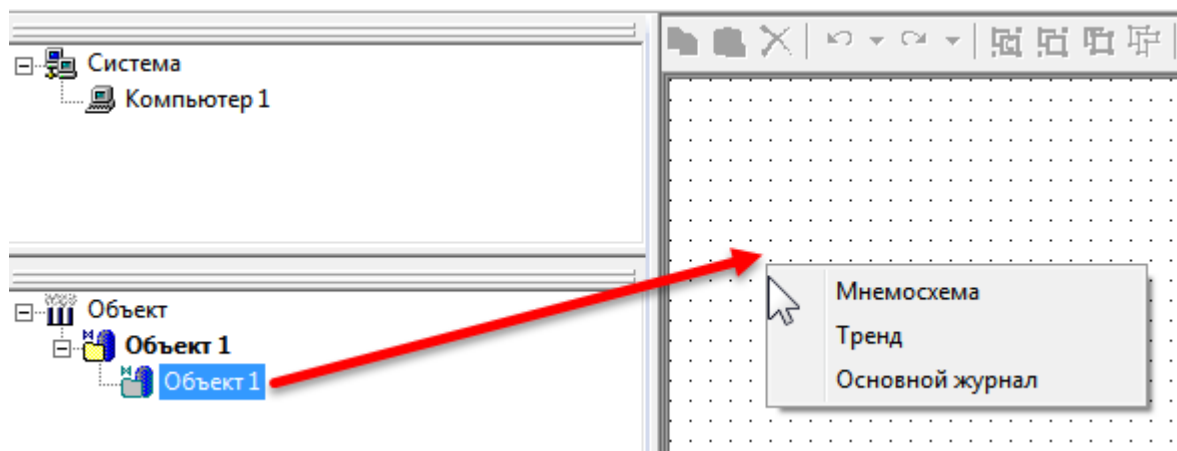


Рисунок 6-21. Создание кнопки вызова окна

Если происходит вставка окна или МНЕМОСХЕМЫ, то по умолчанию кнопка имеет вид миниатюры – небольшого скриншота вставляемого окна.

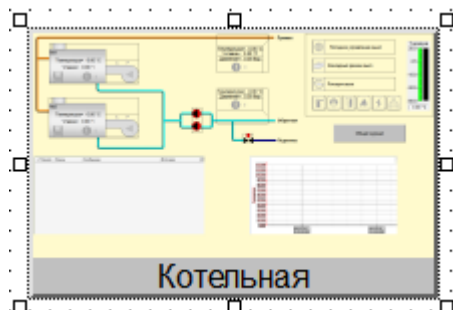


Рисунок 6-22. Миниатюра открываемого окна в кнопке

В окне свойств можно исправить параметры кнопки (к какому документу она привязана), задать подпись, вид кнопки, а также переопределить параметры открытия окна.

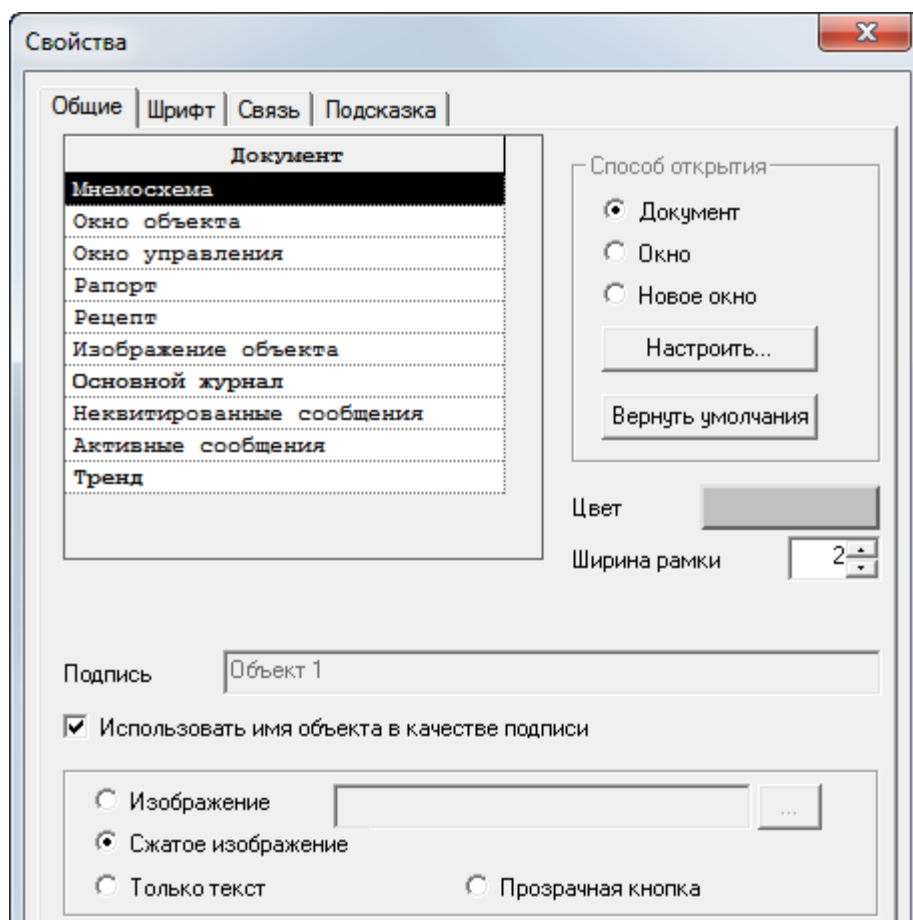


Рисунок 6-23. Окно настройки кнопки вызова

При необходимости кнопку можно сделать прозрачной (например, положить поверх какого-нибудь исполнительного механизма), тогда при щелчке по ней оператор будет открывать окно.

6.5.2. Открытие окон по внешним сигналам и автоматическое открытие

Как быть, если необходимо открывать мнемосхемы не просто кнопкой, а по различным внешним (или внутренним) сигналам – например, при возникновении аварии? Для решения этой задачи можно использовать модуль **Событие** и его закладку **Действия**, а также ФБ Управление документом.

Например, рассмотрим следующую задачу. Необходимо открывать окно, если в объекте возникло сообщение с категорией **Авария**, но при этом автоматический переход в подобных ситуациях должен быть разрешен оператором.

Создадим в объекте команду типа дискретный и назовем ее **Разрешить открытие**. Если команда включена, то будет разрешено автоматическое открытие мнемосхемы при возникновении аварии.

Создадим два значения и назначим им шкалы **Температура** и **Давление** – этими переменными мы будем генерировать возникновение аварии.

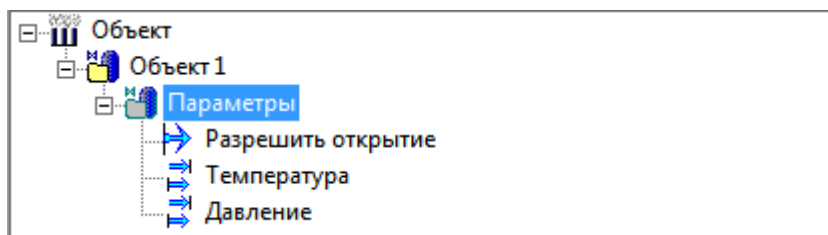


Рисунок 6-24. Переменные в дереве объектов

Для того чтобы отследить наличие аварийных сообщений, мы будем использовать ФБ Счетчик событий из раздела **Службные** палитры ФБ. Добавим этот ФБ в объект и на закладке **Настройки** оставим подсчет только сообщений категории **Авария**. Теперь при возникновении сообщений на выходе ФБ Есть сообщения появится сигнал **Истина**.

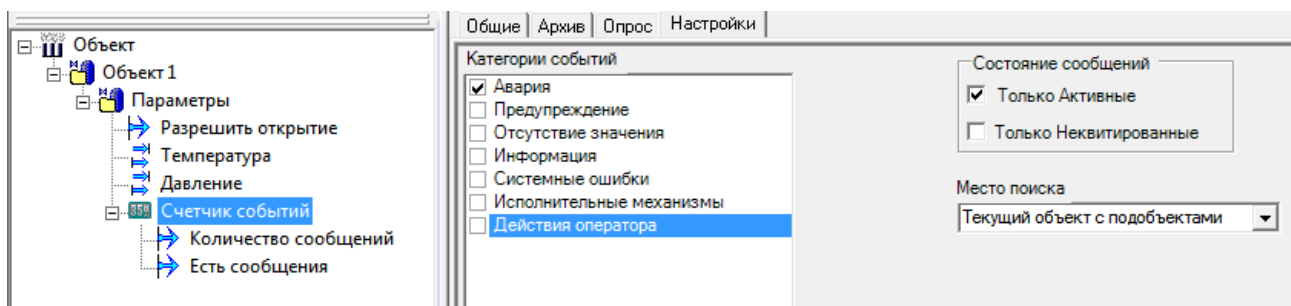


Рисунок 6-25. ФБ «Счетчик событий» и его настройки

Открытие окна мы будем производить с помощью модуля **Событие** – добавим его в объект через контекстное меню объекта. Напишем формулу события, при котором будет производиться открытие: когда включена команда **Разрешить открытие** и есть сообщения у **ФБ Счетчик событий**.

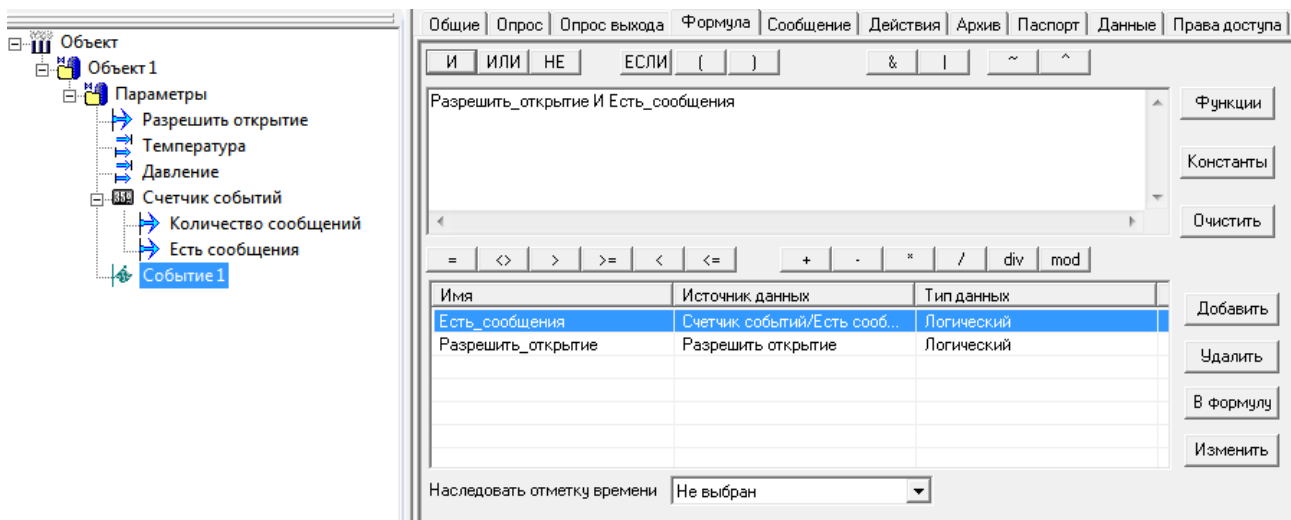


Рисунок 6-26. Формула события

Теперь настроим открытие мнемосхемы. На закладке **Действия** добавим действие типа **Открыть** и выберем документ, который мы хотим открыть. На этом настройка завершена.

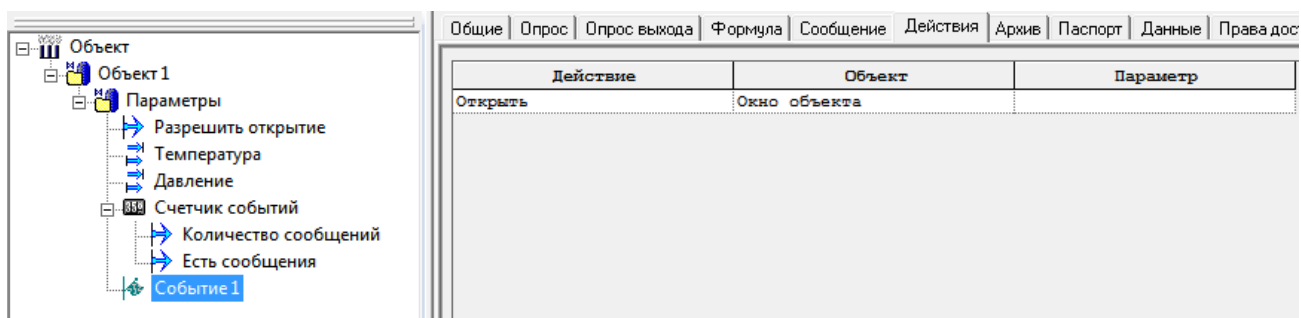


Рисунок 6-27. Действие для открытия окна объекта

Теперь в режиме исполнения при возникновении аварий происходит открытие окна.

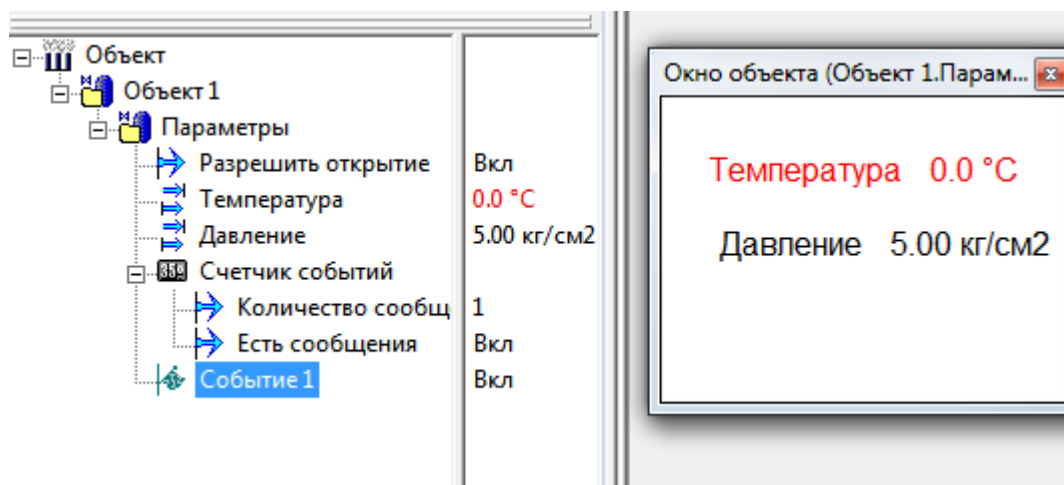


Рисунок 6-28. Автоматическое открытие окна

В зависимости от формулы события открытие окна можно выполнить по различным условиям: от сигнала контроллера, по времени и т. д.

Аналогичным образом можно использовать ФБ **Управление документом**. Помимо открытия и закрытия он может сохранять документы в различные форматы и выводить их на печать. Также с его помощью можно отслеживать состояние документа и открывать его в определенной позиции. Подробное описание данного ФБ можно найти к справке к этому блоку.

6.5.3. Продвинутое способы открытия окон

Существует и ряд других, специализированных, способов открытия. К таким способам относятся открытие из журнала и через контроль **Навигация по объектам**.

Рассмотрим открытие окна через контроль **Навигация по объектам**. Данный контроль находится в разделе **Окна** палитры мнемосхемы. Контроль загружает список вложенных

объектов, имеющих окна, или другие документы, разрешенные к открытию из контроля. В режиме исполнения оператор может вызвать контекстное меню и открыть документ объекта. Контроль имеет поиск по имени объекта.

В качестве примера использования рассмотрим навигацию по дереву объекта диспетчеризации газопоршневой установки (ГПУ). У объекта первого уровня (уровень установки) создаем отдельное окно – **Окно управления**.

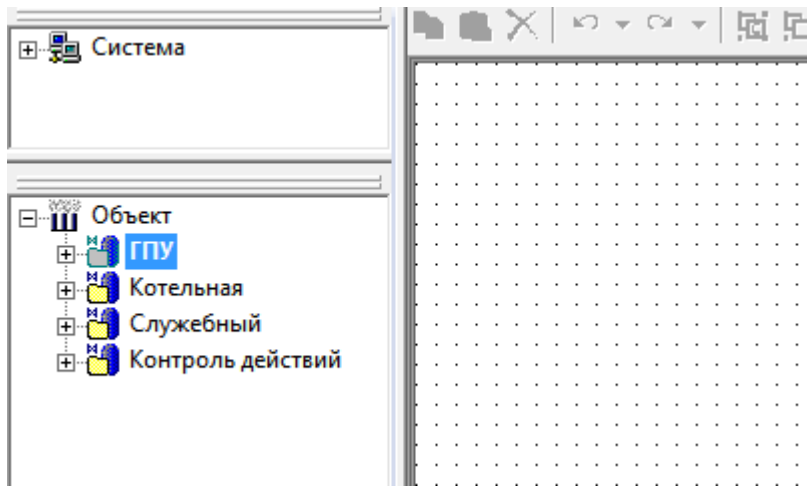


Рисунок 6-29. Окно объекта ГПУ

Располагаем в нем контроль **Навигация по дереву**.

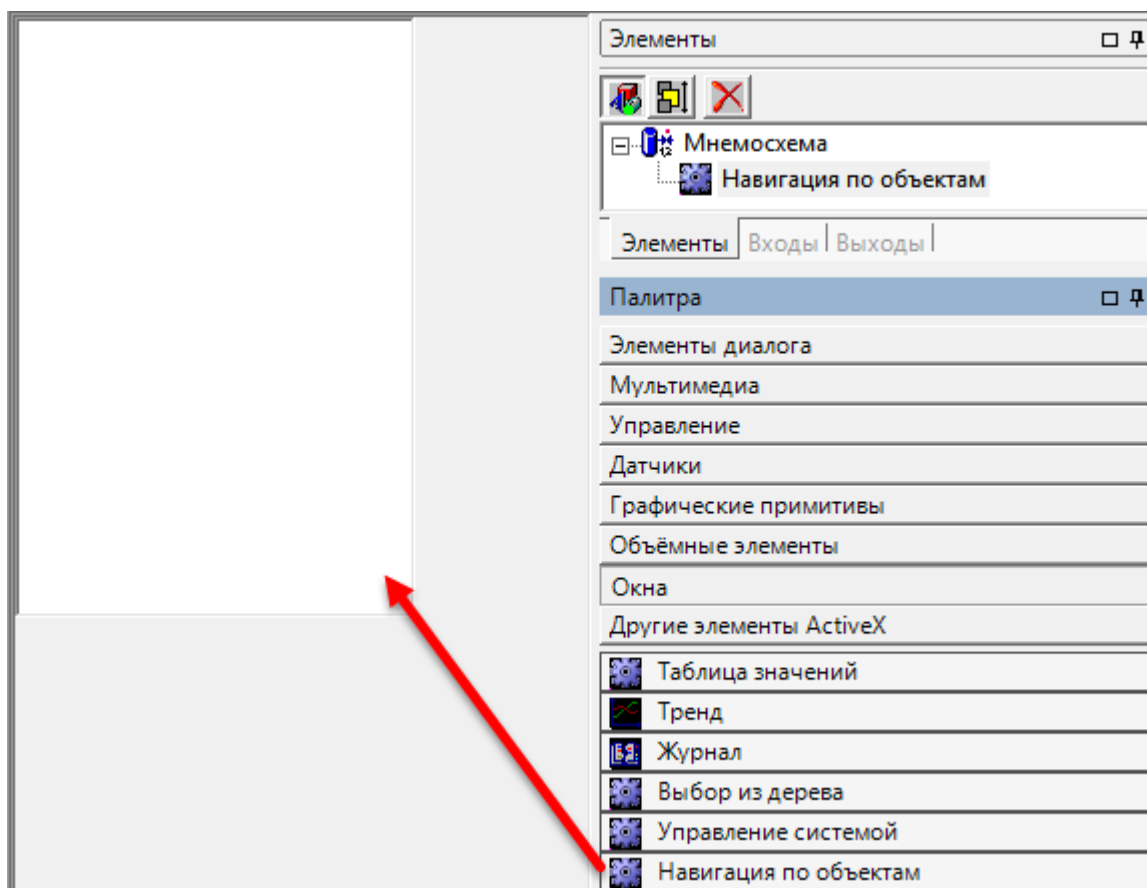


Рисунок 6-30. Добавление контрола «Навигация по дереву»

Теперь настроим, какие объекты мы хотим видеть в дереве. Для этого у объекта на закладке **Общие** установим галочку **Видимость в – Окне** (Рисунок 6-31). У всех дочерних объектов также будет включена данная настройка. Если у каких-то объектов видимость в контроле не требуется, то данную галочку у них можно снять.

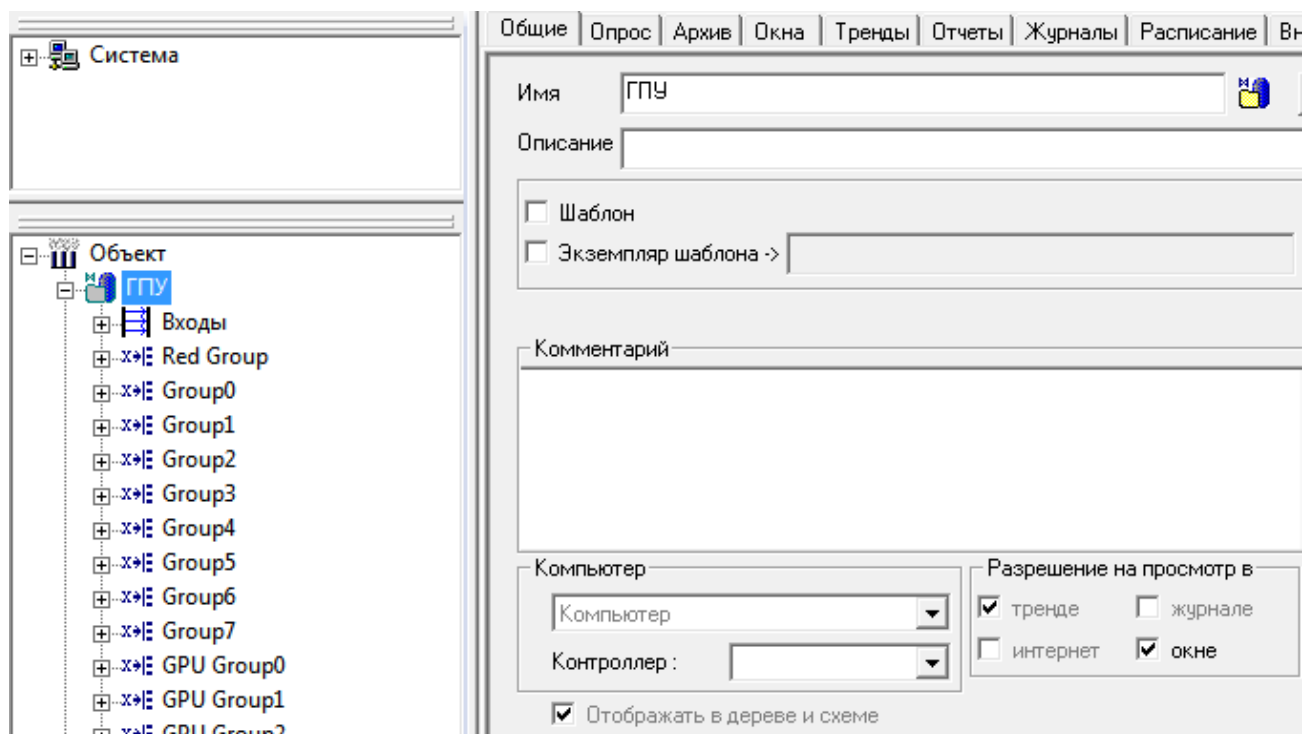


Рисунок 6-31. Настройка видимости объекта в окне

Запустим режим исполнения и откроем окно управления. В нем мы видим все объекты, где есть окна и разрешена видимость в окне. Вызвав контекстное меню, можно открыть какое-либо окно данного объекта.

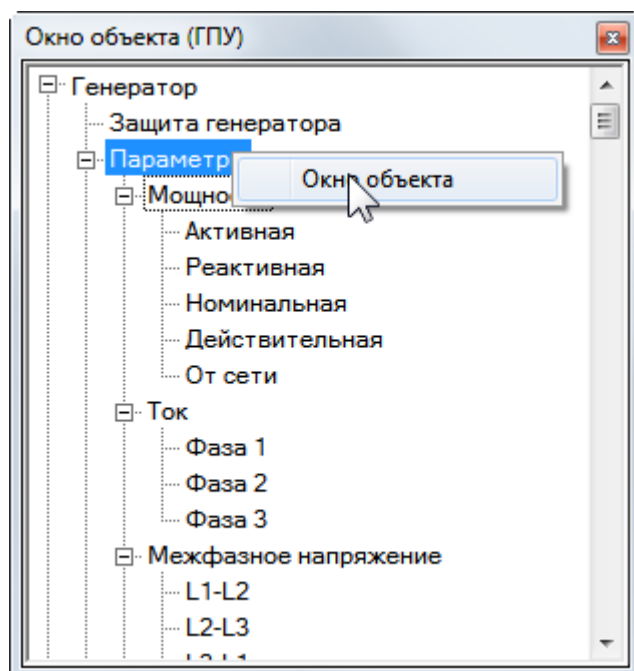


Рисунок 6-32. Открытие окна из дерева

При необходимости можно добавить функцию поиска. Для этого предназначен вход динамизации **Поиск по имени**, на который нужно подать строку (например, от команды типа строковый). В зависимости от настройки **Способ отображения** результата найденные компоненты будут просто выделены (если включен режим **Отображать все**), или все элементы, не соответствующие фильтру, будут скрыты, а результаты отображены в плоском или иерархическом виде. Результат поиска при выключенной настройке **Отображать все**, желтом цвете выделения найденных элементов и иерархическом способе отображения показан на 6-33.

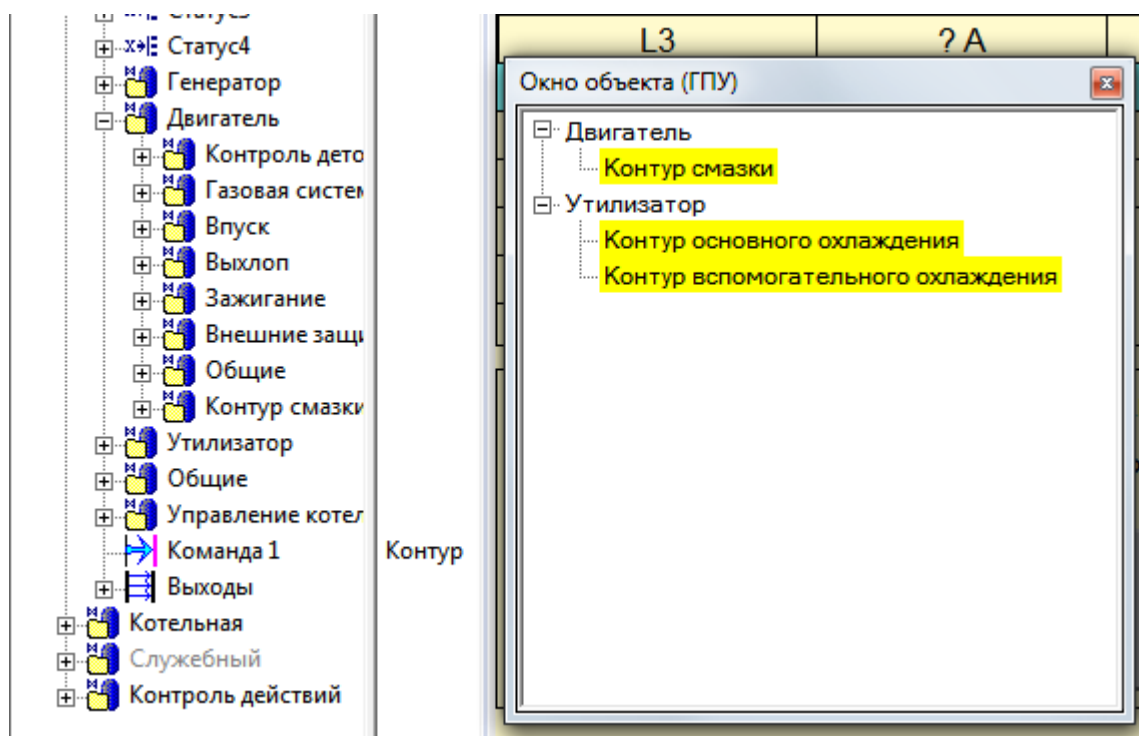


Рисунок 6-33. Результат поиска по дереву

Подробное описание контроля также есть в справочной системе.

Кроме того, можно открывать определенные окна по сообщению из журнала сообщений. Например, можно разместить на стартовой мнемосхеме контроль журнала с выводом аварийных сообщений проекта. При возникновении аварии оператор сможет выделить нужное сообщение и перейти к определенному окну (например, на мнемосхему клапана, из-за которого возникло аварийное сообщение). Рассмотрим создание такой конфигурации.

Добавим в корневой объект еще один объект (назовем его **Общий**), а в него – еще один объект (назовем его **Объект с насосом**); добавим **ФБ Насос мнемонический**. Создадим у объекта **Окно объекта** и вытащим на него **контрол** насоса.

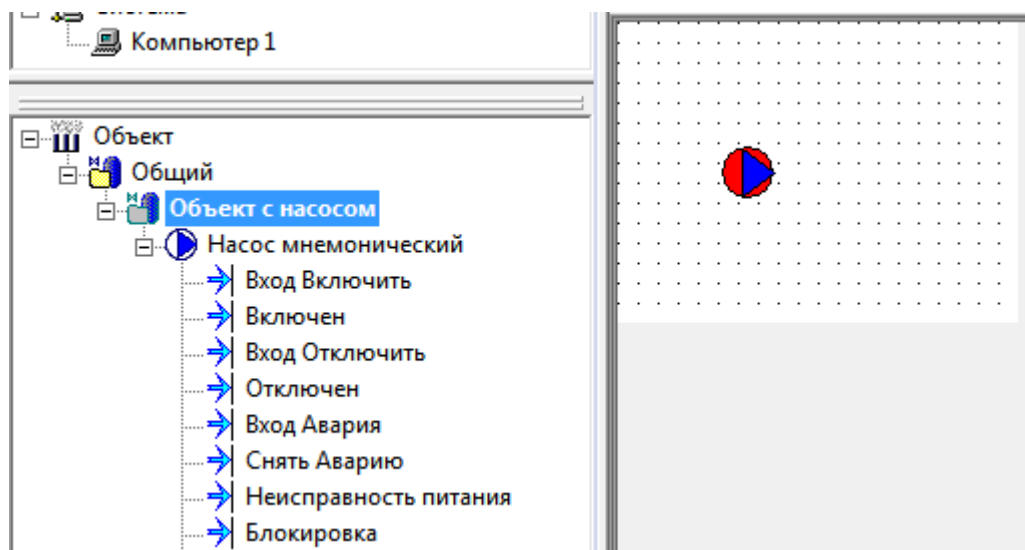


Рисунок 6-34. Создание окна объекта с контролем насоса

Закроем мнемосхему, откроем вкладку объекта **Окна** и откроем окно свойств. В данном окне нужно включить настройку **Открывать из журнала** (Рисунок 6-35), в этом случае окно будет доступно в списке открываемых окон в журнале.

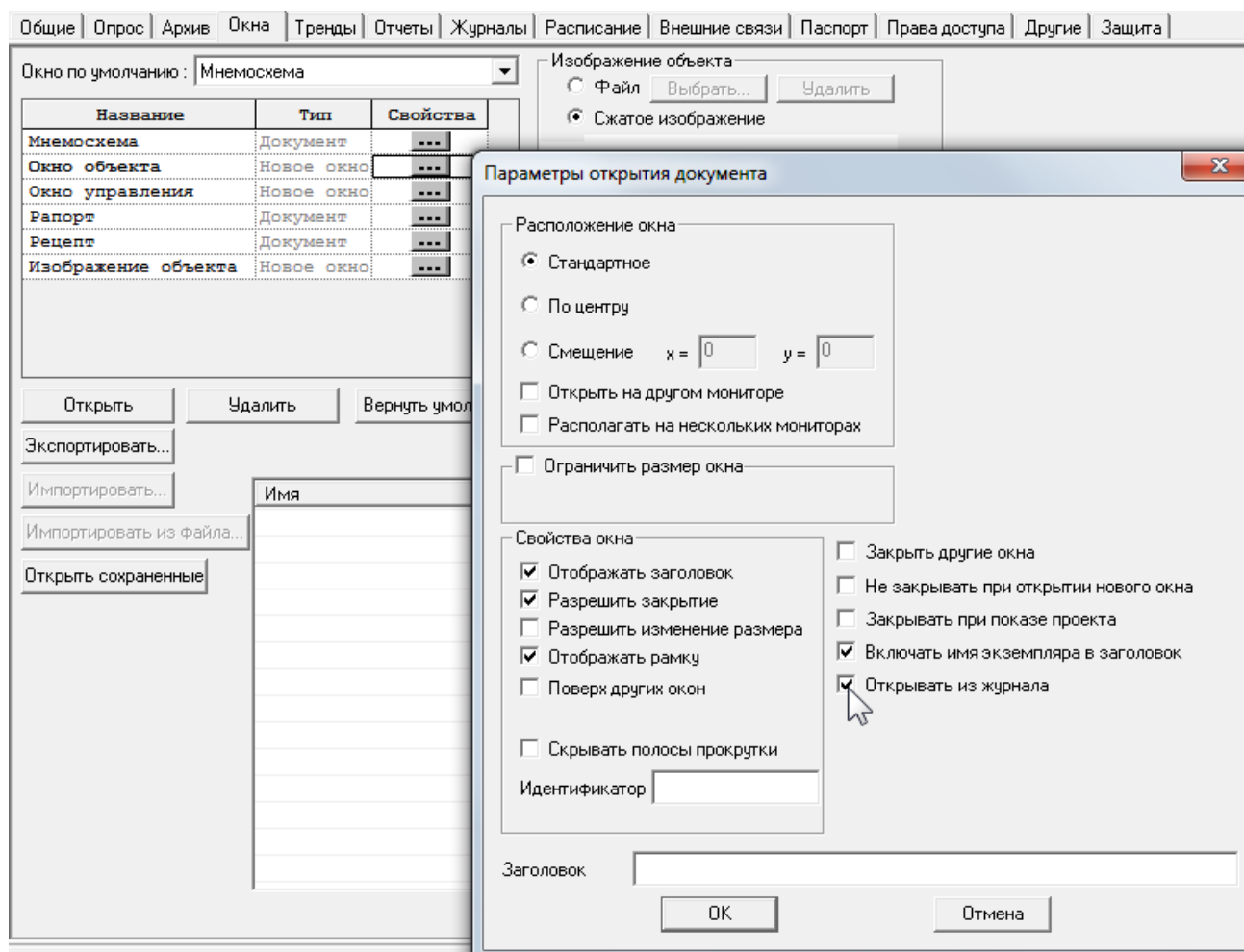


Рисунок 6-35. Включение разрешения открытия окна из журнала

У объекта первого уровня создадим мнемосхему и разместим журнал – добавим его из раздела **Окна** палитры мнемосхемы (Рисунок 6-36).

Редактор мнемосхем MasterGraf

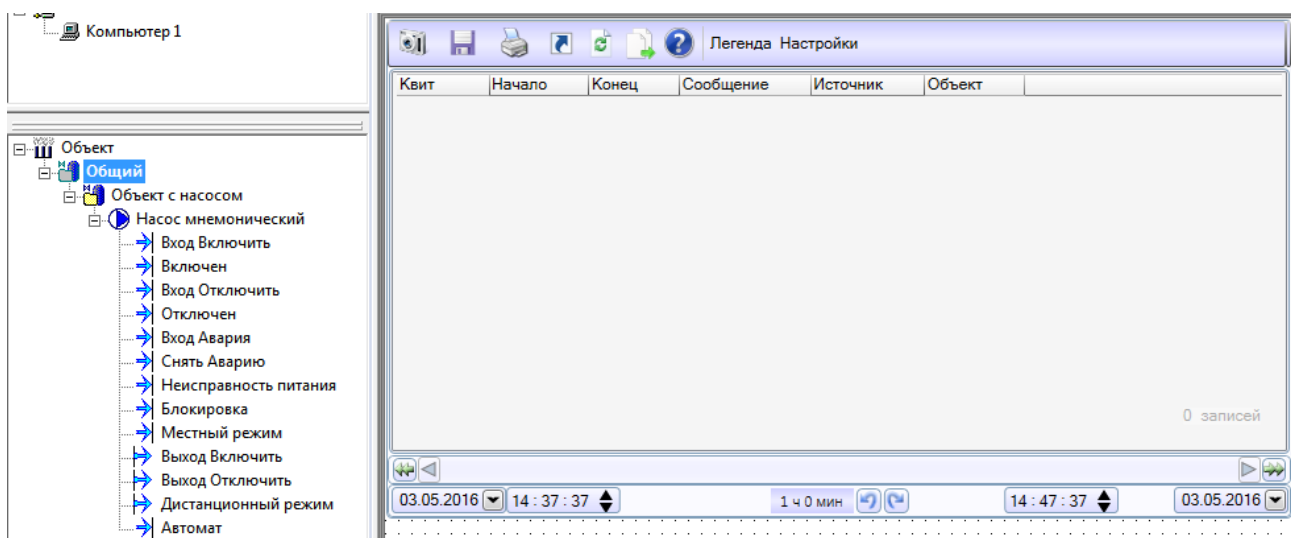


Рисунок 6-36. Создание журнала на мнемосхеме

Запустим режим исполнения, откроем мнемосхему объекта с журналом и сформируем аварию у насоса – подадим сигнал на вход аварии. В журнале появится аварийное сообщение.

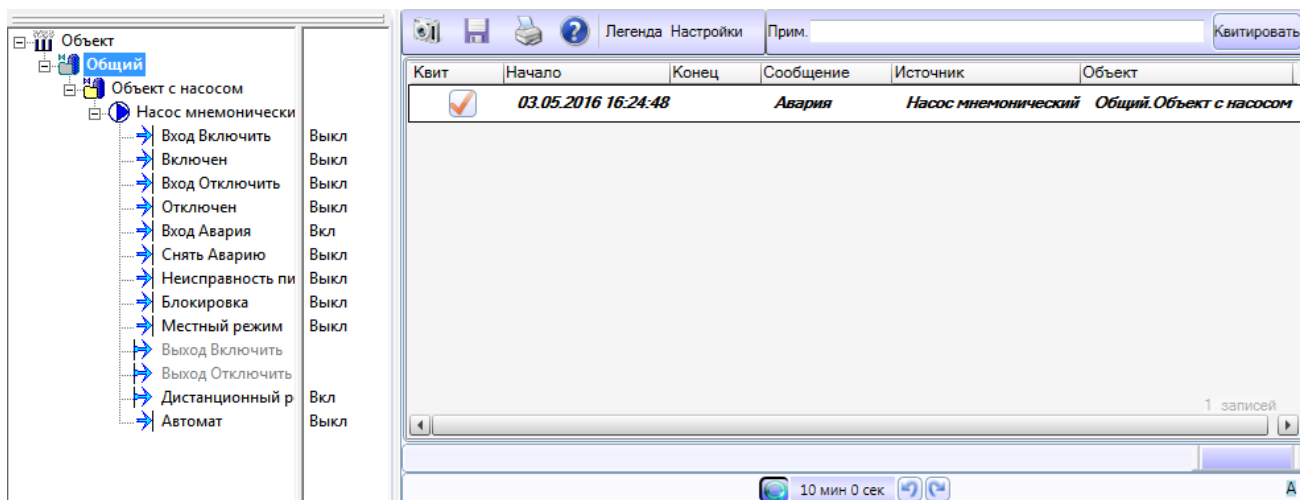


Рисунок 6-37. Аварийное сообщение в журнале

Вызовем у него контекстное меню **Открыть окно** и выберем наше окно.

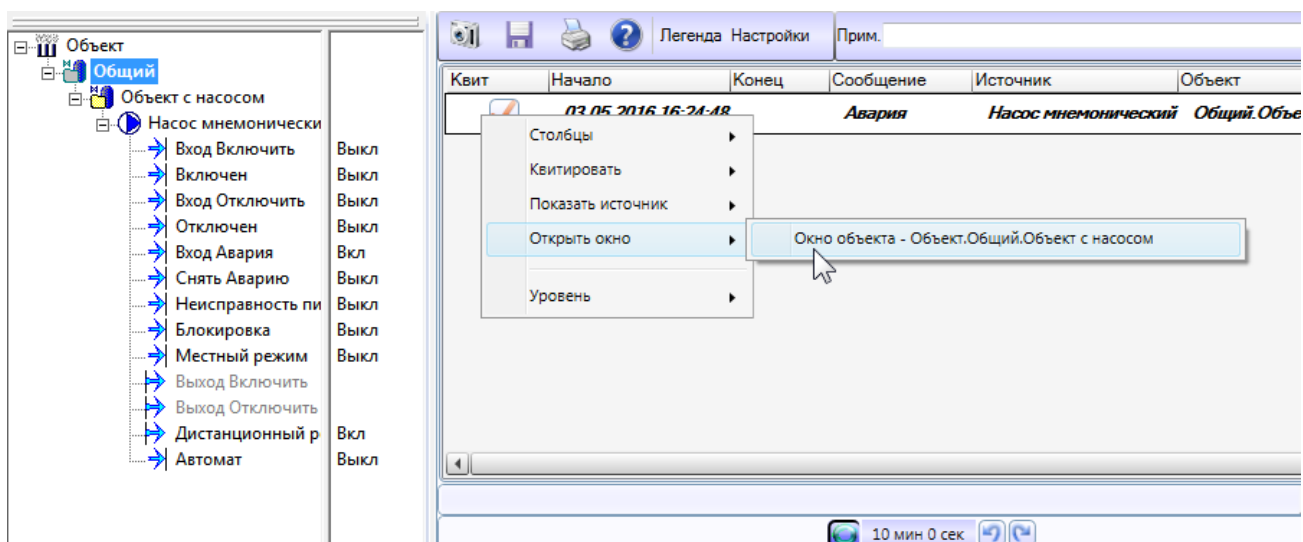


Рисунок 6-38. Вызов меню перехода к окну

Откроется окно с насосом.

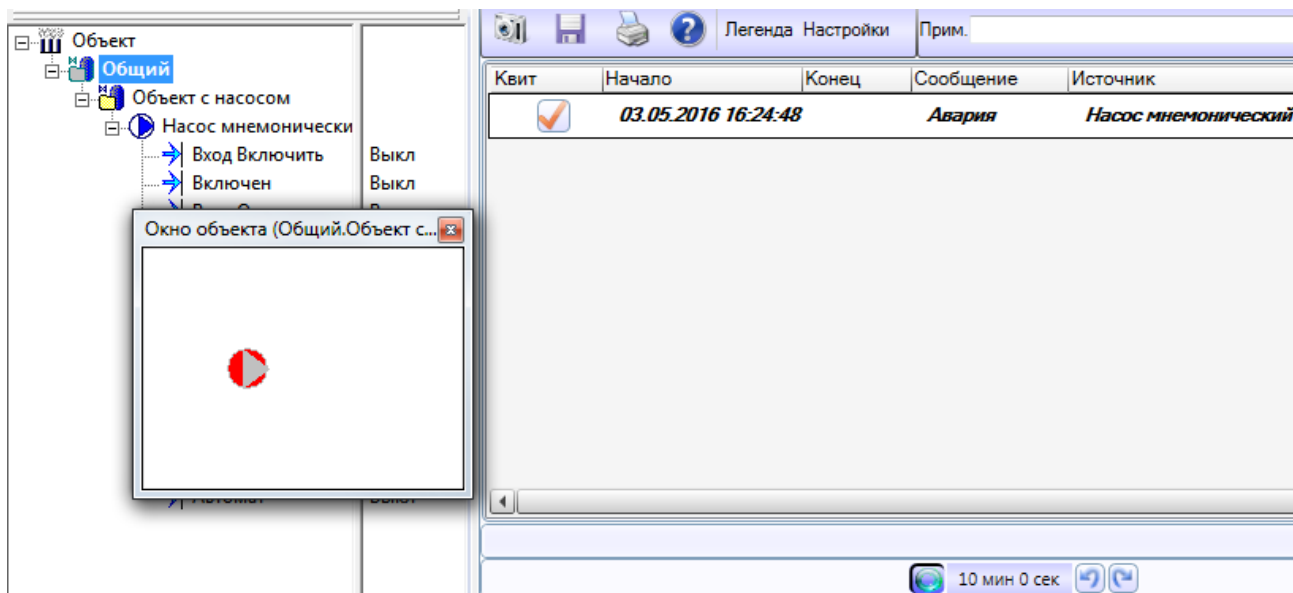


Рисунок 6-39. Открытие окна из журнала

Если у объекта будет создано несколько окон, у которых разрешено открытие из журнала, то все они будут присутствовать в списке. В список выводятся все окна всех объектов пути, по которому расположен элемент, вызвавший сообщение. То есть можно открыть окно объекта, в котором находится насос, или общую мнемосхему объекта установки, или мнемосхему всего участка, и т.д.

6.6. Изображение объекта

В отдельном разделе разберем тип окна **Изображение объекта**. Изображение объекта можно размещать на другой мнемосхеме как встроенный активный символ. Здесь можно провести аналогию с «живыми плитками» в Windows Phone, Windows 8 и Windows 10 – то есть элемент выполняет сразу несколько функций: отображает состояние объекта и позволяет открывать документ этого объекта.

6.6.1. Создание изображения объекта

Рассмотрим создание изображения объекта. Для примера сделаем изображение объекта, которое будет отображать состояние объекта с емкостью – ее уровень, температуру, наличие аварийного состояния.

Создадим изображение объекта на закладке объекта **Окна** и разместим на ней компоненты **Емкость**, **Значения** уровня и температуры, **Индикатор аварии**.

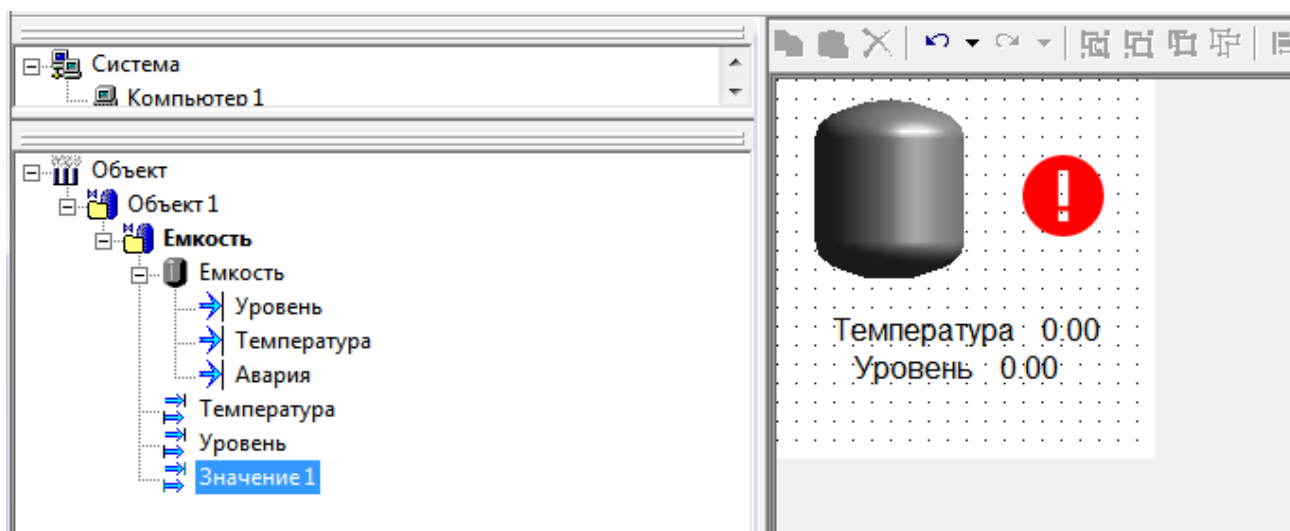


Рисунок 6-40. Изображение объекта с набором контролов

Теперь сохраним изображение объекта, закроем его и создадим мнемосхему в **Объекте 1**.

Примечание. Сохранение изображения объекта перед «вытаскиванием» на другую мнемосхему обязательно! Иначе будет вставлено изображение объекта с последнего сохранения.

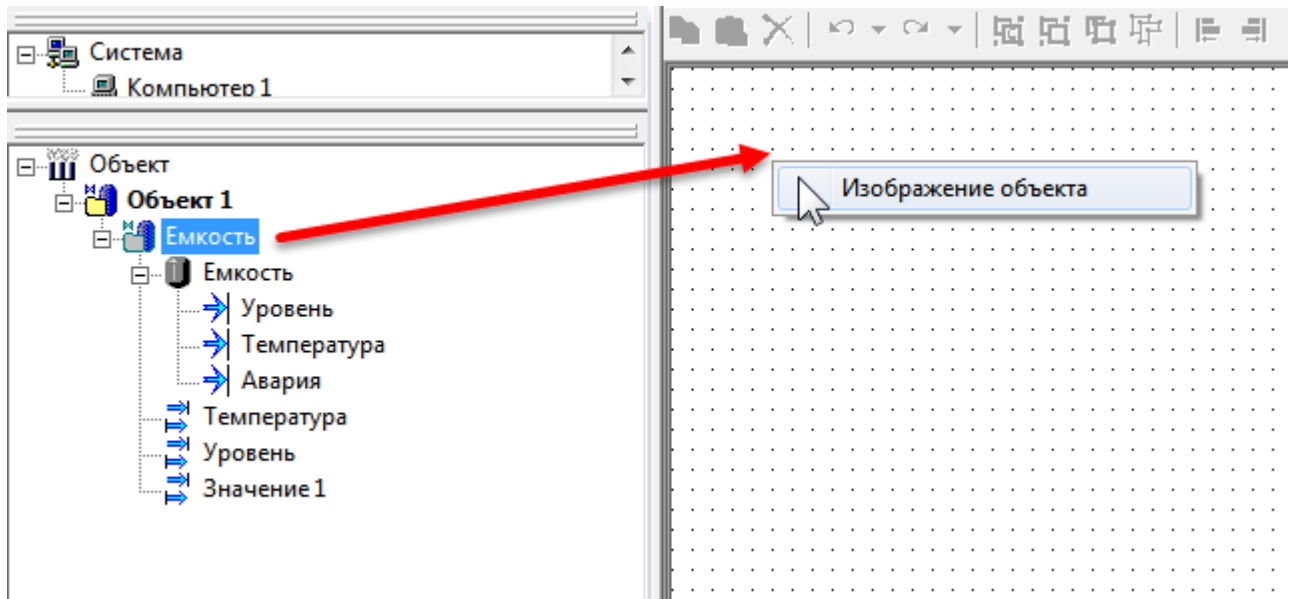


Рисунок 6-41. Вставка изображения на мнемосхему

Вытащим на мнемосхему объект **Емкость** правой клавишей мыши. В появившемся списке выберем **Изображение объекта**.

На мнемосхему добавились элементы, которые мы добавили на изображение объекта.

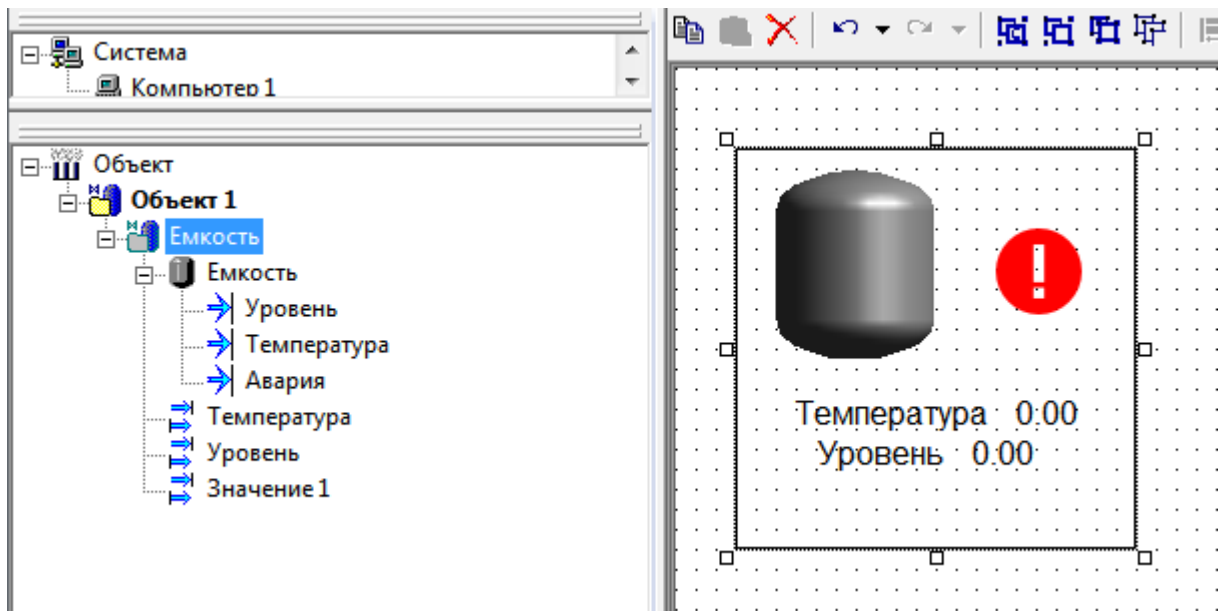


Рисунок 6-42. Мнемосхема с изображением объектов

Рассмотрим, из чего состоит наше изображение объекта.

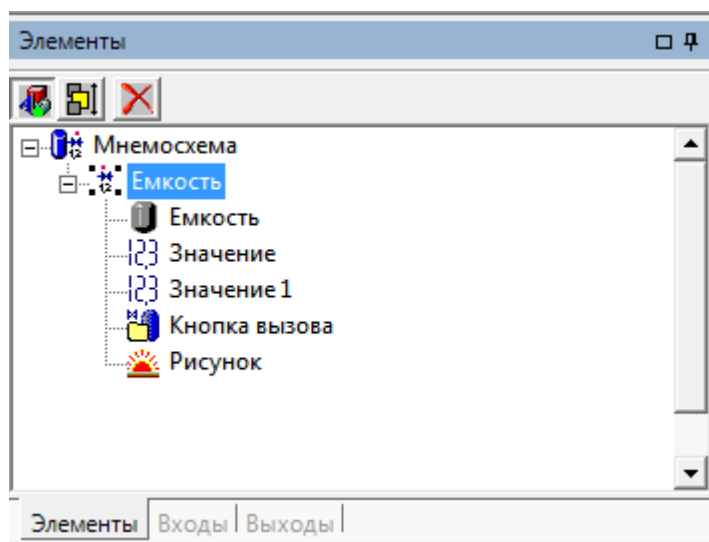


Рисунок 6-43. Элементы изображения объекта

Оно представляет собой символ, содержащий все элементы, добавленные на изображение, а также еще один элемент – кнопка вызова (Рисунок 6-43). При этом у кнопки вызова включен режим прозрачности, поэтому в режиме исполнения она не будет видна. Теперь, в режиме исполнения, оператор при щелчке по элементам изображения объекта сможет выполнить переход к другой мнемосхеме этого объекта (если она создана), а элементы на изображении будут отображать состояние переменных в данный момент времени (Рисунок 6-44).

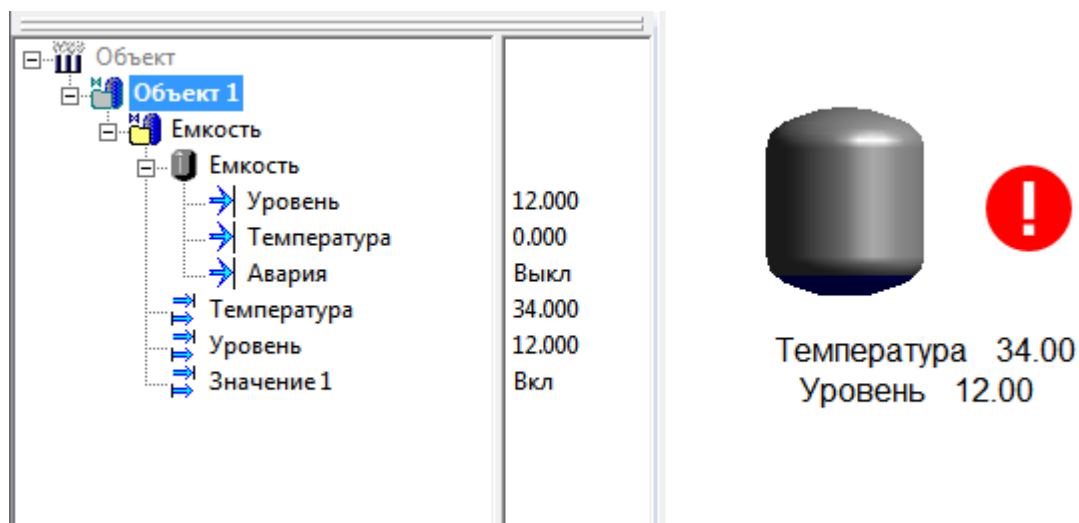


Рисунок 6-44. Изображение объекта в режиме исполнения

6.6.2. Создание дополнительных изображений объекта

Иногда возникает ситуация, когда один объект должен иметь несколько изображений – например, необходимо иметь разные отображения на разных мнемосхемах. Сделать собственные изображения несложно.

Для этого надо выделить элемент **Система** – вкладка **Окна** и добавить новое окно. Для примера назовем его **Изображение объекта 2**.



Рисунок 6-45. Добавление нового окна изображения объекта

Теперь нажмем на кнопку свойств – откроется окно параметров открытия документа. В нем установим галочку **Вставлять как изображение объекта**.

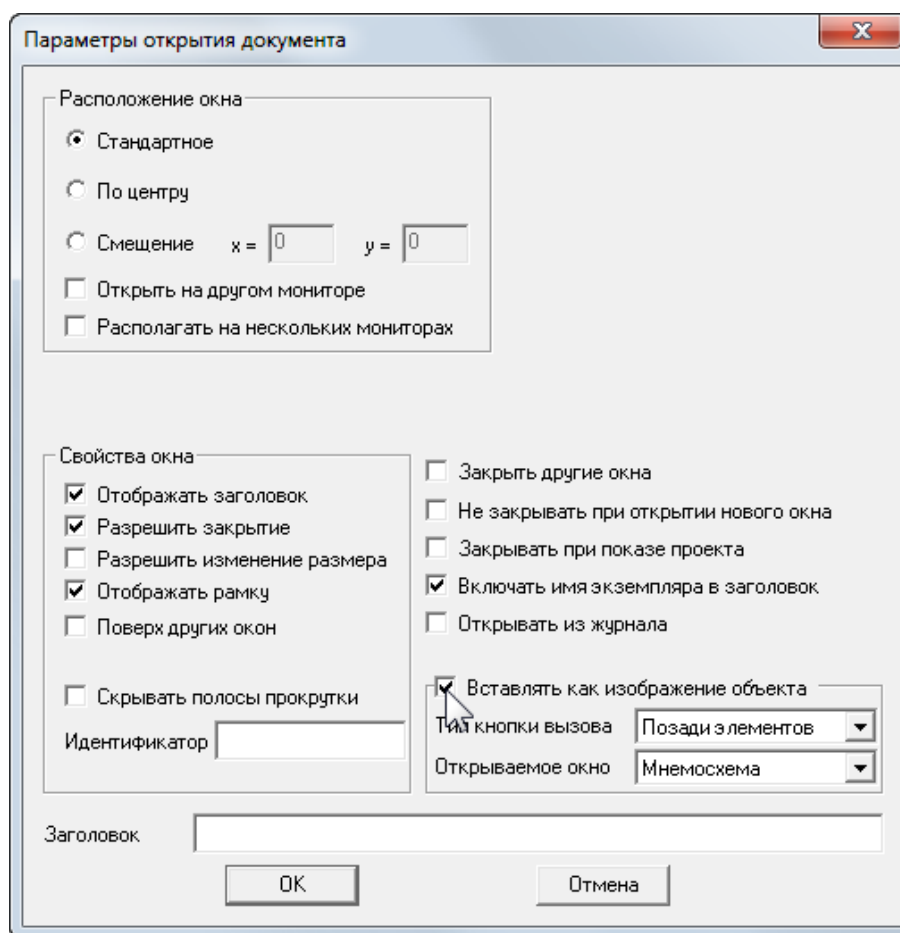


Рисунок 6-46. Параметры открытия окна

Теперь данное окно появится у всех объектов проекта – после создания его можно будет вытаскивать на мнемосхему как изображение объекта.

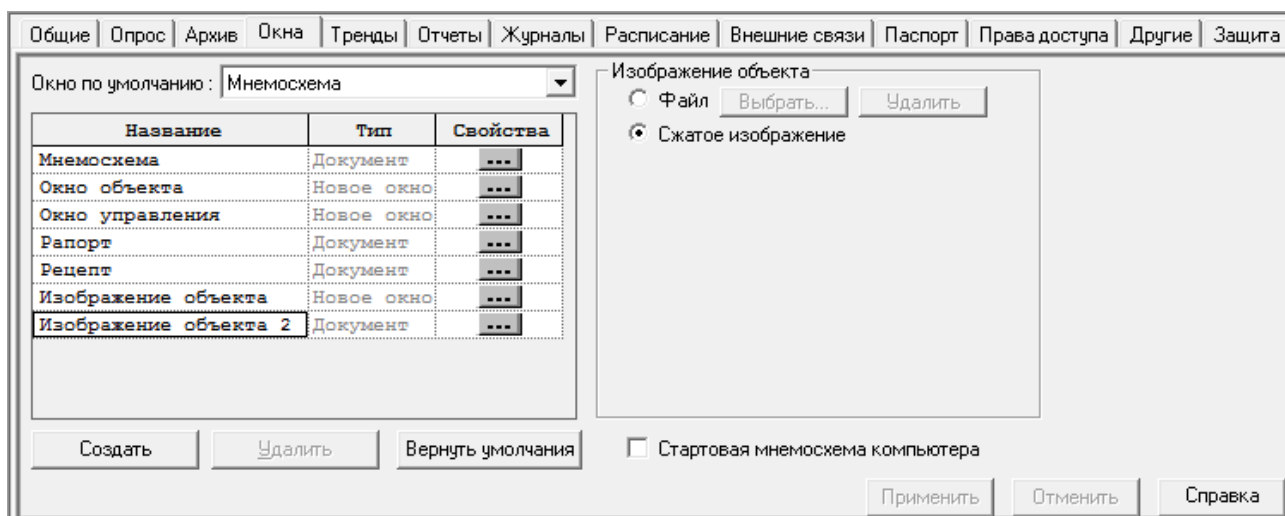


Рисунок 6-47. Созданное изображение у объекта

Обратите внимание на настройку **Тип кнопки вызова**. Данная настройка определяет положение кнопки вызова в изображении объекта. Возможно три варианта:

Позади объектов – кнопка вызова будет находиться ниже всех элементов изображения в Z-иерархии. Это приводит к тому, что элементы выше кнопки становятся доступными для нажатия. То есть если на изображении находится контроль насоса, то щелчок по нему откроет его окно управления.

Поверх объектов – кнопка вызова будет находиться выше всех элементов. Таким образом, остальные элементы для нажатия будут недоступны.

Не добавлять – кнопка вызова будет добавлена, но у нее будет включена настройка **Заблокирован**, то есть нажатие на нее приводит к открытию окна не будет.

Настройка **Открываемое окно** определяет, какое окно будет открывать кнопка вызова данного изображения объекта. По умолчанию она открывает мнемосхему. Можно изменить этот параметр и установить любое другое окно.

7. Система архивации и тренды

Помимо отображения текущих данных на мнемосхемах и в окнах также требуется архивация параметров с последующим отображением их на трендах. MasterSCADA поддерживает архивацию в собственный файловый архив или во внешние СУБД; для отображения в поставку входит модуль трендов с очень широким функционалом.

Рассмотрим систему архивации.

7.1. Система архивации

По умолчанию переменные в MasterSCADA не архивируются. Чтобы началось ведение архивов переменной, необходимо перетащить переменную на тренд или установить галочку **Архивировать** на вкладке **Архив**.

The screenshot displays the 'Archive' configuration window for a variable. The 'Name in DB' field contains 'Объект 1.Значение 1'. The 'Export to DB' section has an empty 'DB-connector' dropdown. The 'Parameters of archiving' section includes an unchecked 'Archive' checkbox, a 'Шаблон' dropdown set to 'Аналоговые', and a 'Восстановить все умолчания' button. The 'Redefine template settings' section has 'Периодически' unchecked, 'По изменению' checked, and 'Моделирование ступеньки' unchecked. The 'Мертвая зона' field is set to 0. The 'Statistics' section is empty. The bottom of the window features 'Применить', 'Отменить', and 'Справка' buttons.

Рисунок 7-1. Вкладка «Архив» переменной

На этой же вкладке настраиваются параметры архивации и метод архивации. В MasterSCADA есть два метода архивации: **По изменению** и **Периодически**.

По изменению – при данном способе архивации новое значение записывается, только если его величина отклонилась от предыдущего значения более чем на величину настройки **Мертвая зона**. Например, если значение было равно 100, а мертвая зона равна 1, то колебания значений в диапазоне от 99 до 101 в архив записываться не будут, и только если значение выйдет за пределы мертвой зоны, оно будет записано в архив, после этого мертвая зона будет контролироваться относительно нового значения. Рекомендуется задавать настройку мертвой зоны в пределах 0,5 % от шкалы – так как именно такой класс точности обеспечивают большинство промышленных приборов, а архивировать значения ниже величины погрешности, как правило, не имеет смысла.

Периодически – в этом случае значения записываются в архив с определенным шагом времени, заданным в поле **Период**. При этом необходимо иметь в виду, что запись нового значения в архив происходит только в том случае, если метка времени переменной изменилась более чем на величину периода. То есть если используется опрос OPC-сервера по изменению и значение в OPC-сервере не меняется, то и метка его меняться не будет, соответственно, и записи значения в архив не будет производиться: это позволяет избежать дублирования данных в архиве.

Мы рекомендуем использовать архивацию по изменению – это наиболее оптимальный механизм. Полное описание методов архивации и настроек данной закладки приведено в справочной системе.

У каждой переменной можно задать собственные параметры архивации. Также можно сделать глобальные настройки – с использованием шаблонов архивации. Настройка шаблонов архивации производится на закладке **Система – Архив – Шаблоны** (Рисунок 7-2).

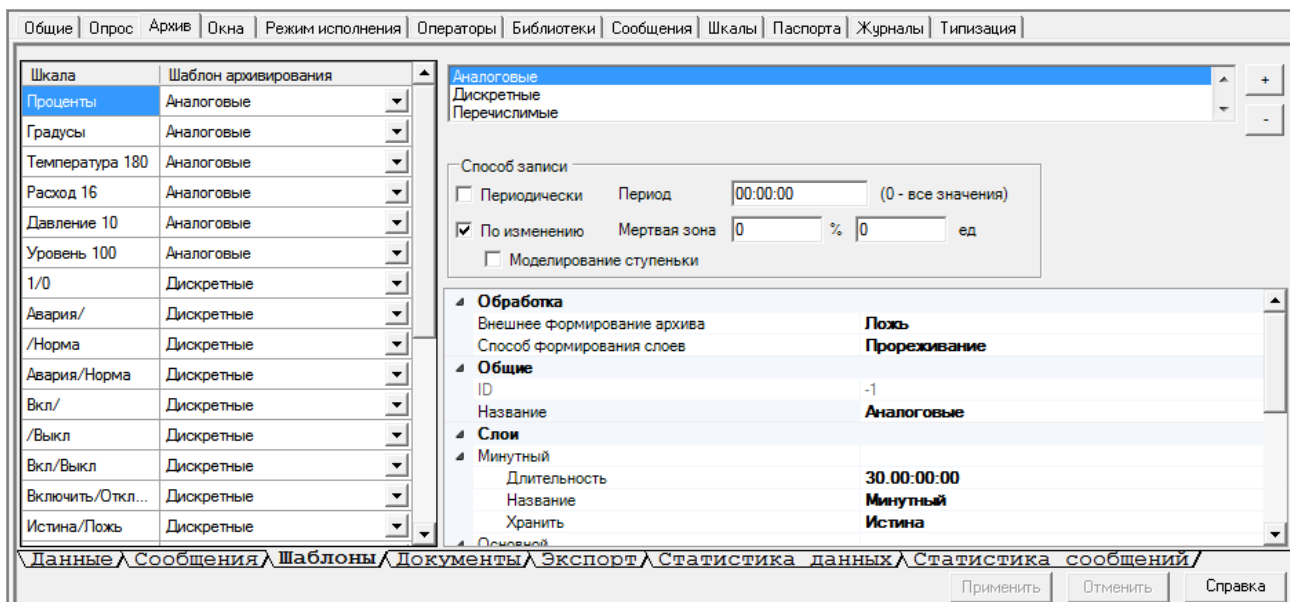


Рисунок 7-2. Вкладка «Архив – Шаблоны» элемента «Система»

На данной закладке каждой аналоговой и дискретной шкале можно задать шаблон архивирования – как системный, так и собственный. Кроме того, у каждого шаблона можно переопределить настройки: задать метод архивации и параметры (**Мертвая зона**, **Период**, **Моделирование ступеньки**). Таким образом для разных шкал можно задать разные параметры архивации (например, для давлений включить моделирование ступеньки и сделать меньшую мертвую зону).

Произведенные на данной закладке настройки будут применены ко всем переменным с данным шаблоном архивирования.

7.1.1. Архивация в файловый архив

Архивация в файловый архив – это способ архивации по умолчанию. В этом режиме данные пишутся в кольцевой буфер из файлов, который хранится в папке проекта. Папка проекта MasterSCADA имеет аналогичную структуру проекта. В папке объекта, в которой находится архивируемая переменная, создается папка **_Data**. В этой папке для каждой переменной этого объекта создается стек файлов с данными. Каждый файл имеет размер до 256 Кб. Когда длительность или объем архива превышают заданные пользователем, самый старый из файлов удаляется.

Настройки длительности и объема хранения задаются у **Системы** и наследуются сверху вниз (от **Системы** к **Компьютеру**, от **Компьютера** к **Объектам**), при этом на каждом уровне параметры могут быть переопределены. Рассмотрим настройки архивации **Компьютера** (Рисунок 7-3).

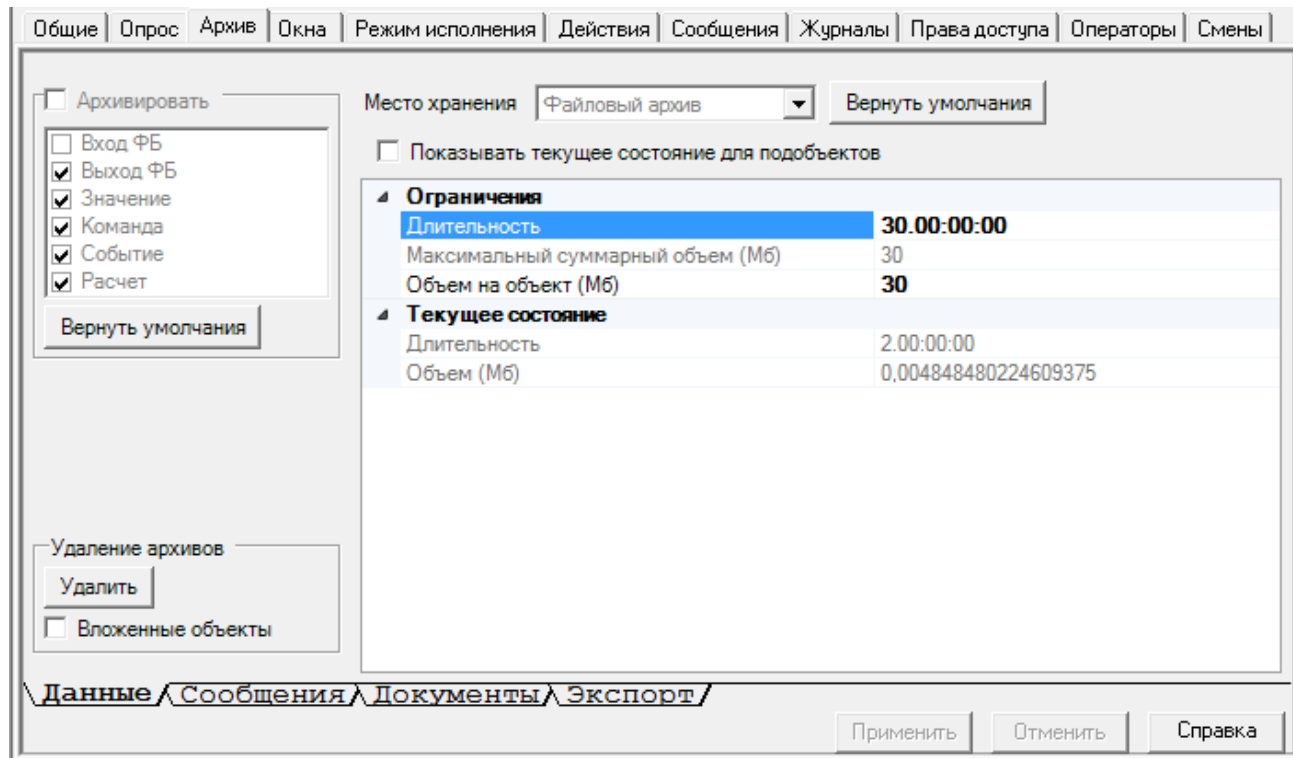


Рисунок 7-3. Настройки архивации в файловый архив

Основных настроек две: **Длительность** и **Объем на объект**. Настройка **Длительность** определяет, сколько времени хранить данные, – по достижении этого предела самый старый из файлов стека переменной будет удален (когда размер текущего превысит 256 Кб).

Объем на объект определяет, сколько мегабайтов выделено на каждый из объектов с архивными данными. При этом данная настройка независима для каждого из объекта, в котором есть архивируемые переменные, – можно настроить разные параметры хранения данных для каждого объекта (но обычно этого не требуется).

В настоящий момент жесткие диски имеют достаточно большие объемы, поэтому, как правило, ограничение на объем архивации не используют, полагаясь только на

длительность. В таком случае можно задать **Объем на объект** 5000–10000 Мб и задать нужную длительность хранения данных.

Также следует помнить, что старые данные удаляются не сразу, когда будет превышен срок хранения, а только когда текущий блок данных (то есть текущий файл, в который пишутся данные) превысит размер 256 Кб.

Удалить файловый архив можно с помощью кнопки «Удалить» на вкладке «Архив». При этом можно удалить архивы только текущего объекта или всех вложенных. При экспорте проекта («Проект» – «Экспортировать») формируется проект без архивов – его можно импортировать или копировать поверх старого, тогда архивы старого проекта сохраняются.

7.1.2. Архивация в СУБД

Помимо архивации в файловый архив также есть возможность архивации в различные базы данных: **MS SQL, Oracle, Firebird, PostgreSQL**.

Архивация в СУБД дает ряд преимуществ, основное из которых – значительное ускорение получения данных и, как следствие, ускорение построения трендов. Основой всех преимуществ архивации в СУБД являются слои. Слой – это отдельный, прореженный слой данных. В файловом архиве слой всего один – основной, в него попадают все архивные данные, в СУБД также есть набор (настраиваемый пользователем) прореженных слоев. По умолчанию в системе три слоя: **минутный, часовой, суточный**. Настройка слоев производится на закладке **Системы – Архив – Данные** – кнопка **Слои**.

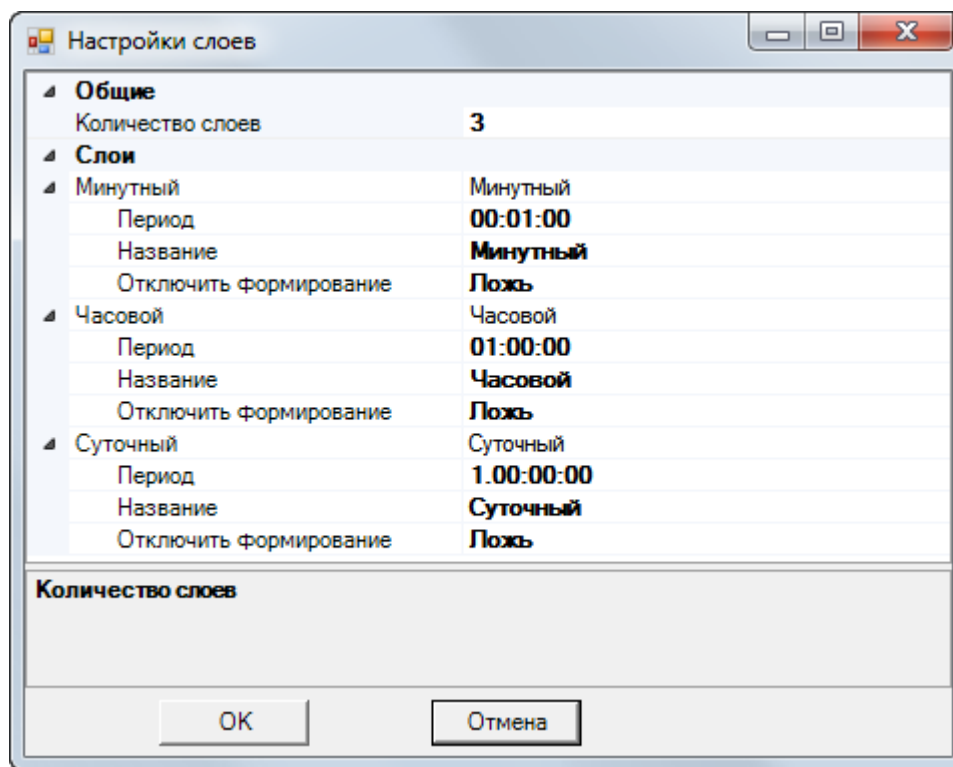


Рисунок 7-4. Окно настройки слоев

Рассмотрим работу слоев на примере минутного слоя.

В режиме исполнения MasterSCADA каждую минуту, используя данные основного слоя, формирует 4 значения: максимальное, минимальное, первое и последнее. Эти значения и записываются в минутный слой. Аналогично происходит с часовым и суточным слоями.

Какие преимущества дают слои? Главное преимущество – ускорение построения трендов. Представим себе, что нам необходимо построить тренд переменной за месяц. В случае использования файлового архива система начнет извлекать из архива все данные за месяц и передавать их в тренд, в итоге может оказаться, что считано несколько десятков тысяч значений, при этом на один пиксель монитора приходится несколько значений. Загрузка такого объема, разумеется, потребует много времени, особенно при работе по сети (в сетевом проекте). В случае же использования слоев тренд может автоматически выбрать предпочтительный слой и брать данные не из основного слоя, а из прореженного. Это многократно снижает объем считываемых данных, не ухудшая внешнего вида построенного графика.

Второе преимущество слоев – возможность настроить отдельный срок хранения данных независимо для каждого слоя каждого шаблона переменной. Это позволяет гибко настроить параметры архивации для различных типов переменных. Настройка длительности хранения производится на закладке **Система – Архив – Шаблоны**.

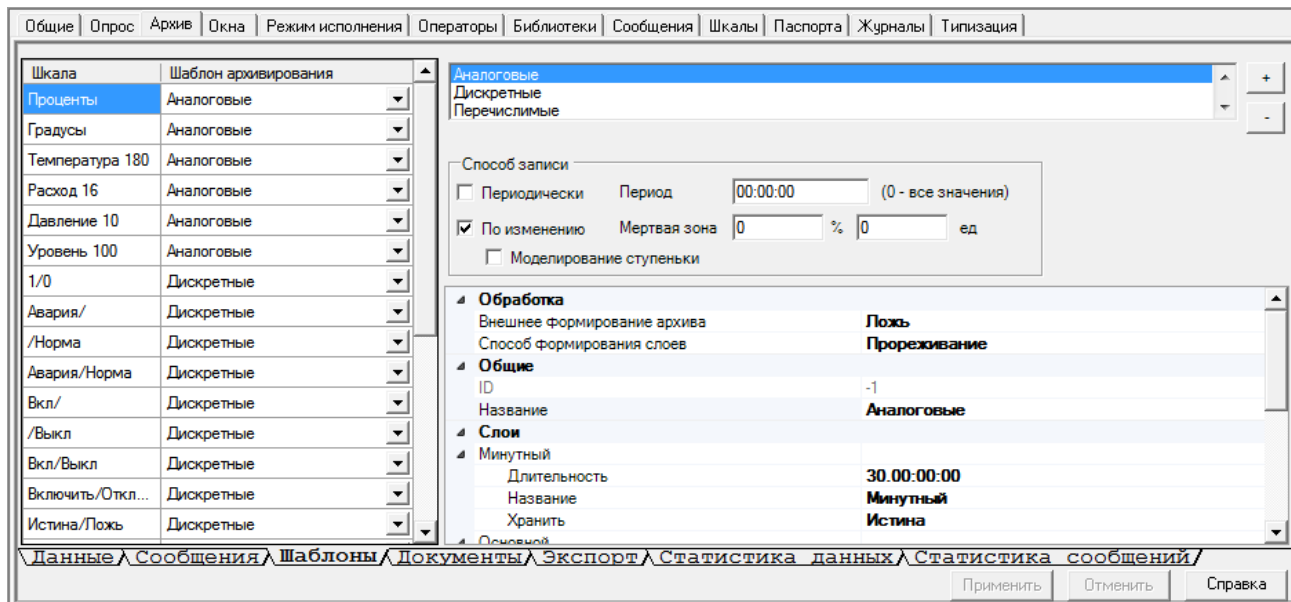


Рисунок 7-5. Настройки шаблонов архивирования

Теперь рассмотрим непосредственно настройку СУБД для архивации данных. Первое, что нужно сделать, – добавить в компьютер БД-коннектор нужной базы данных. В качестве примера мы будем использовать **MS SQL**. Добавим в компьютер эту БД (Рисунок 7-6).

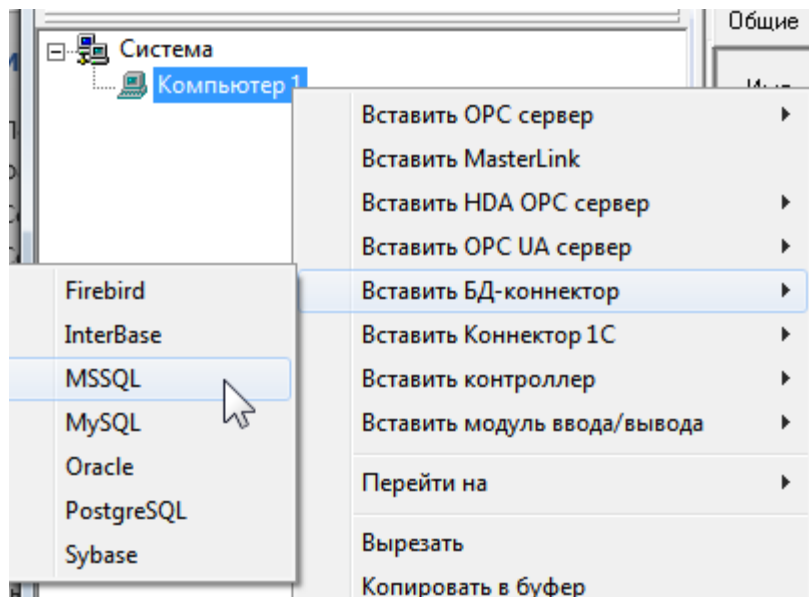


Рисунок 7-6. Добавление БД-коннектора

Перейдем на вкладку **Параметры** добавленного БД-коннектора.

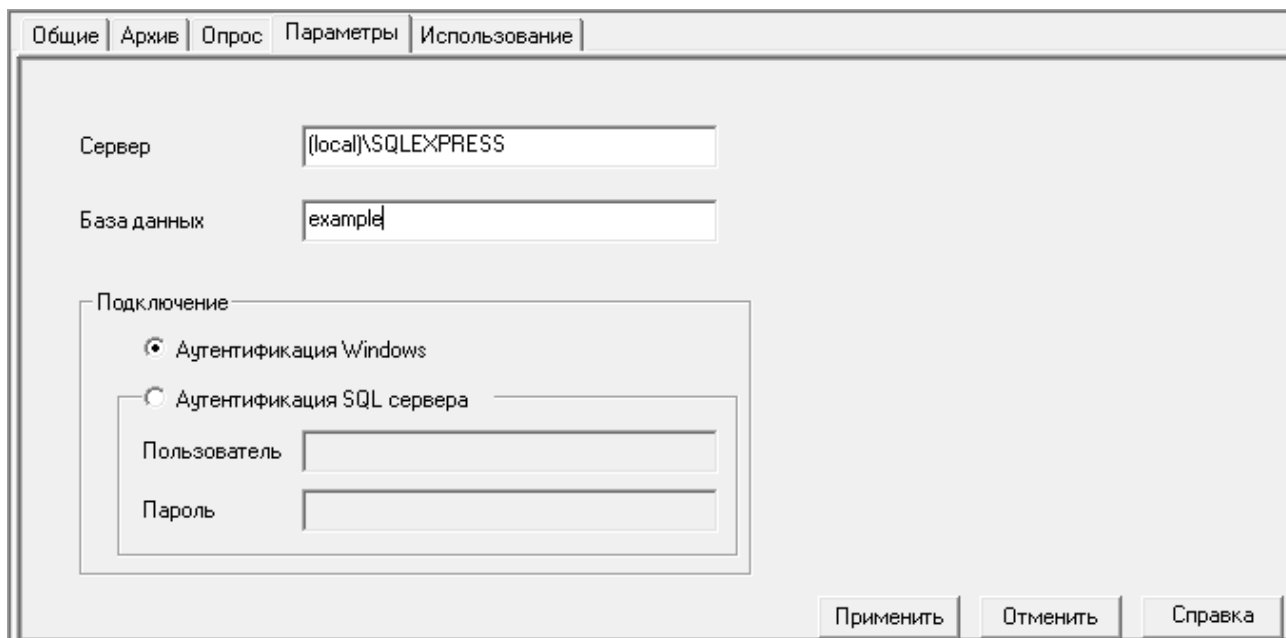


Рисунок 7-7. Настройки соединения с базой

В поле **Сервер** прописывается путь к серверу базы данных. Если сервер установлен на этом же компьютере, то можно указать **(local)\ИмяЭкземпляраСервера**. Если же сервер установлен на удаленной машине, то указывается IP-адрес или имя компьютера, а также имя экземпляра сервера. В поле **База данных** указывается имя базы, в которую будет идти архивация. Для **MS SQL** базу данных необходимо создать вручную – это делается через **Management Studio** (среда администрирования **MS SQL**), также может потребоваться настройка прав доступа. Хранимые процедуры и таблицы MasterSCADA создаст сама.

Также указывается способ подключения: аутентификация Windows или же аутентификация самого сервера по логину и паролю (определяется настройками СУБД).

После того как параметры связи с базой заданы, нужно перейти на вкладку **Использование**.

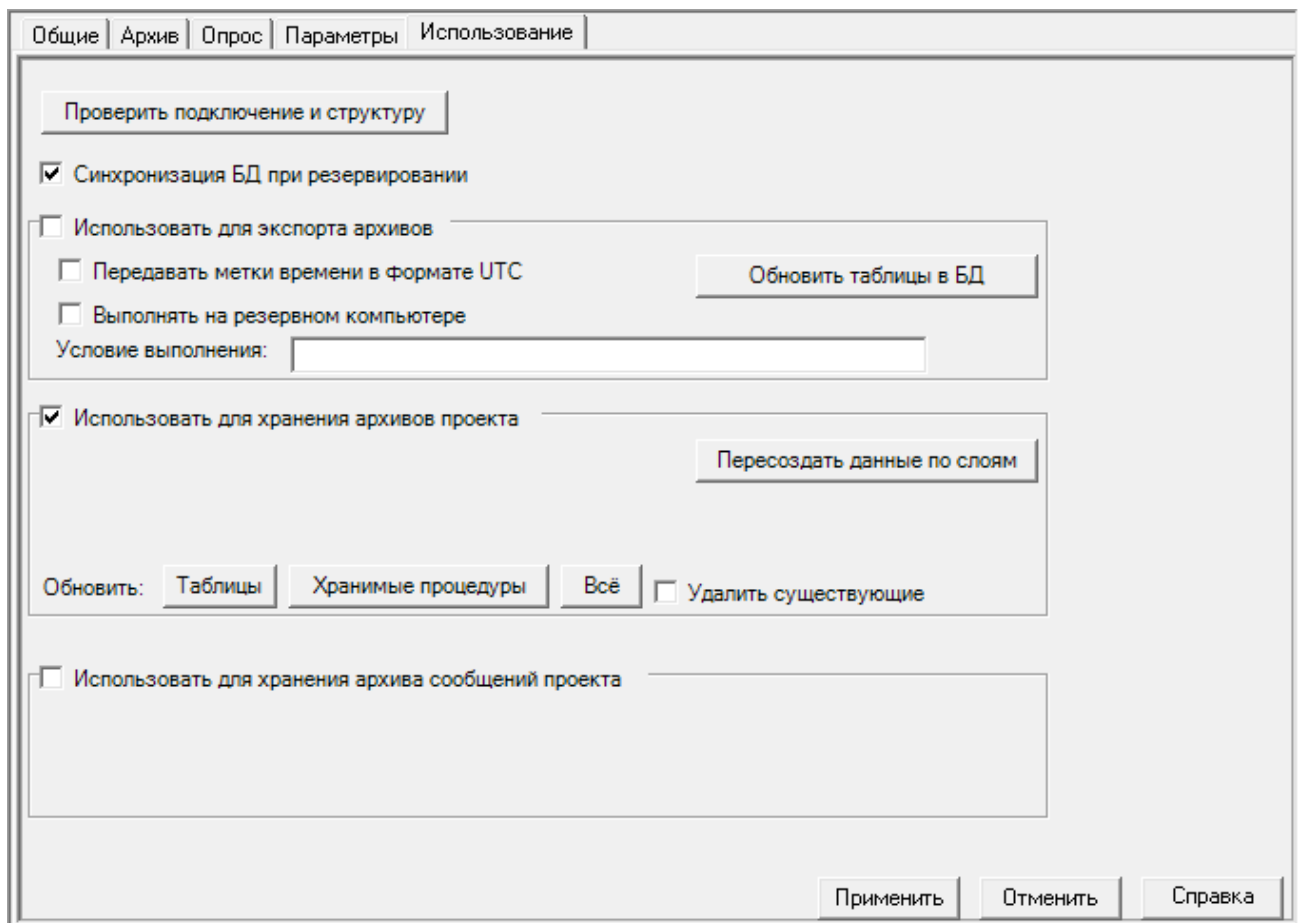


Рисунок 7-8. Выбор режима использования БД-коннектора

На данной вкладке указывается, как будет использоваться база данных. Возможно три режима: экспорт данных, архивация данных, архивация сообщений. Можно использовать одну базу данных во всех режимах сразу. Мы будем использовать базу для хранения архивов – поэтому поставим соответствующую галочку и нажмем кнопку **Обновить – Все**. При нажатии на эту кнопку в базе создаются хранимые процедуры и таблицы.

Пока что мы еще только указали, что данный БД-коннектор можно использовать для хранения архивов. Теперь необходимо указать, что архивация будет идти сюда. Для этого нужно выделить **Компьютер**, перейти на вкладку **Архив** и назначить место хранения – наш БД-коннектор **MS SQL 1** (Рисунок 7-9).

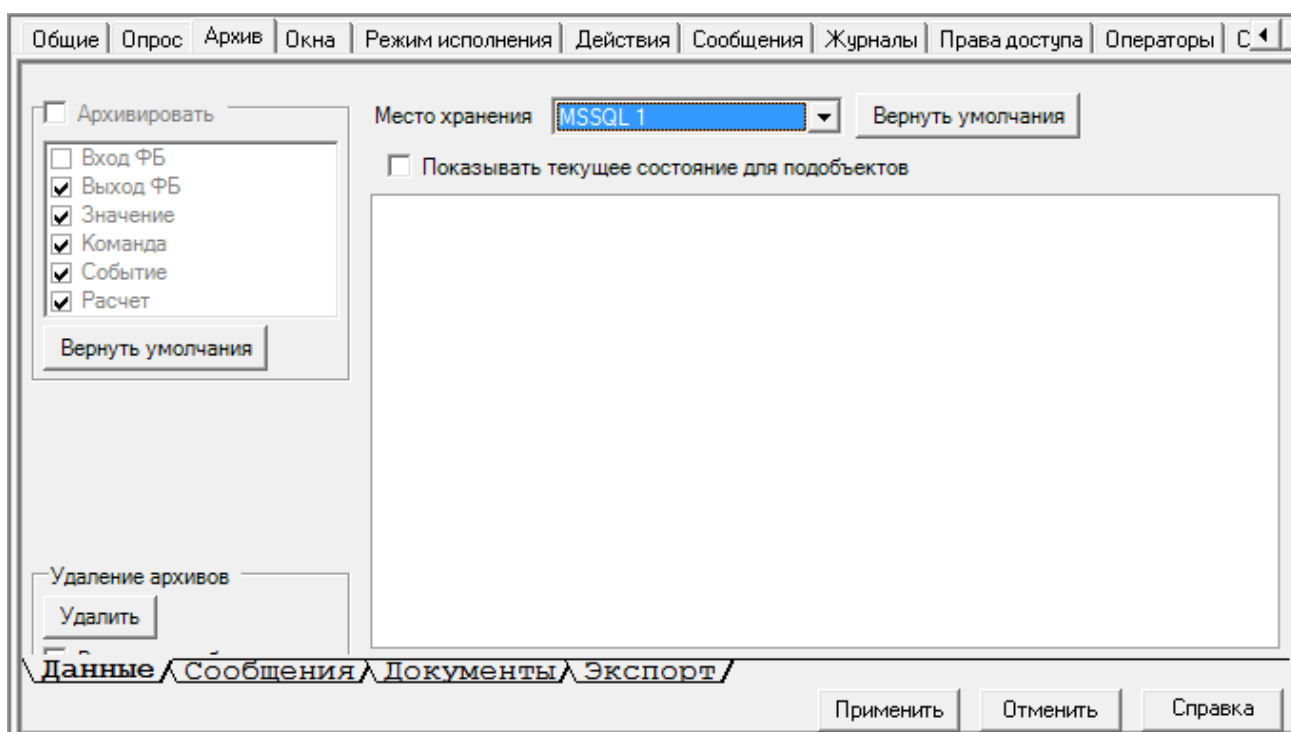


Рисунок 7-9. Назначение архивации в БД-коннектор

Теперь архивация данных будет идти в базу данных **MS SQL**. При необходимости можно добавить несколько БД-коннекторов и назначить архивацию разных объектов в разные СУБД.

Подробнее все режимы работы с базами данных описаны в специальной документации на прилагаемом диске.

7.2. Тренды

Теперь, сохранив данные, их можно читать и обрабатывать. Для работы с архивными данными можно использовать специальные функции модуля **Расчет**, функциональные блоки из раздела работы с архивом, в редакторе отчетов; также есть возможность обрабатывать архивы в скрипте. Но, конечно, основным способом получения архивных данных являются тренды. Модуль трендов позволяет отображать архивные данные в виде графика или таблицы.

7.2.1. Создание тренда

Тренды, как и другие документы, создаются у объектов – для них выделена отдельная вкладка.

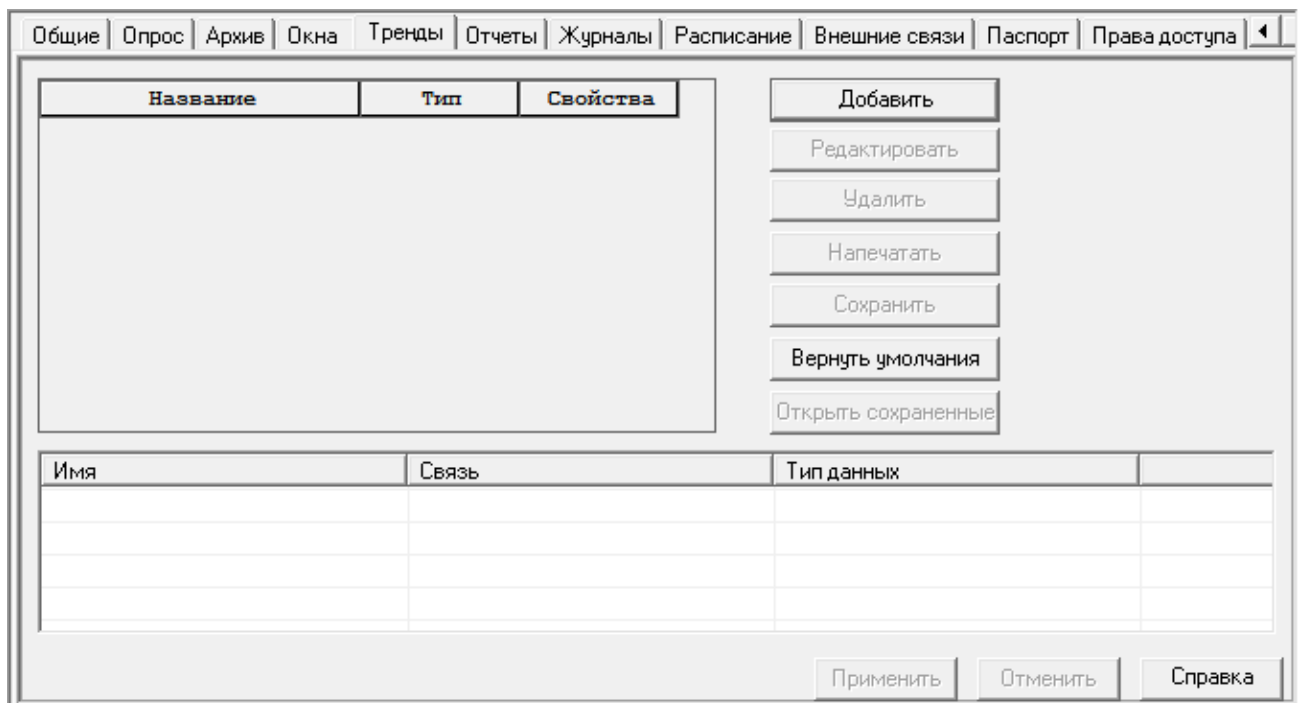


Рисунок 7-10. Вкладка для создания трендов

Трендов у каждого объекта может быть несколько. Для создания нового нажмите кнопку **Добавить** и в появившемся окне введите название тренда. Загрузится интерфейс модуля трендов (Рисунок 7-11).

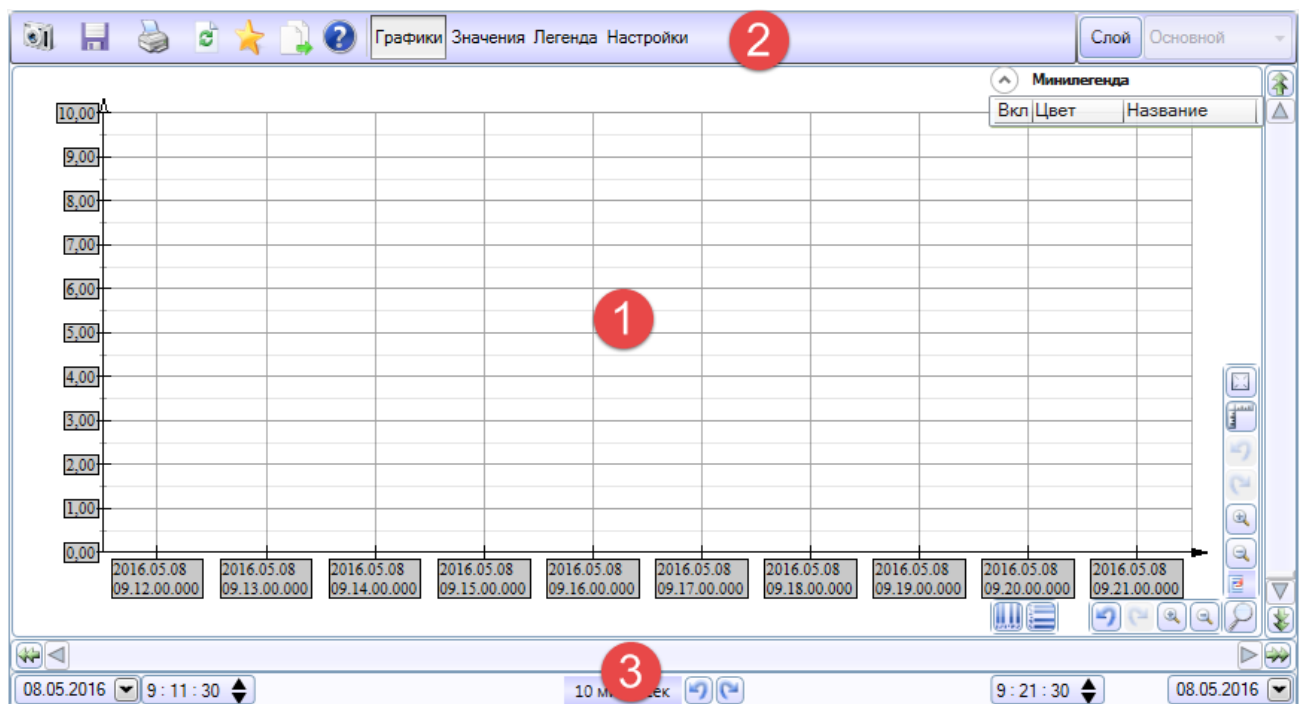


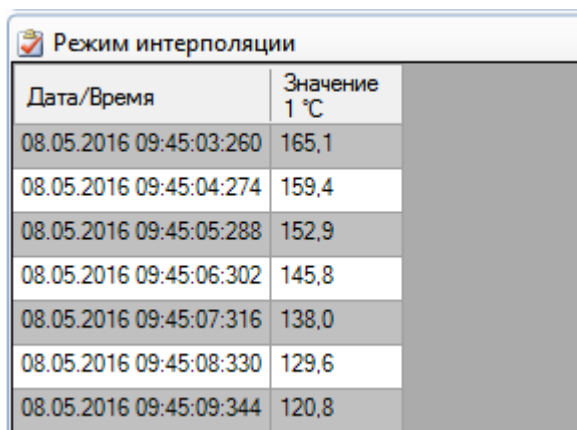
Рисунок 7-11. Интерфейс модуля трендов

Интерфейс состоит из следующих элементов:

1. **Графическая область** – в ней происходит построение графика. В левой части области расположена ось значений, в нижней – ось времени. Расположение осей может быть изменено.
2. **Тулбар** – панель инструментов. В тулбаре расположены кнопки экспорта в графические и текстовые форматы, печати, выбора перьев, вызова справки. А также кнопки для открытия дополнительных панелей: панели значений, легенды, настроек.
3. **Панель даты/времени и навигации** – в данной панели расположены инструменты для навигации по тренду. Можно указать диапазон развертки, используя скроллер, перемещаться по архиву, и т. д.

Также из тулбара могут быть активированы панели:

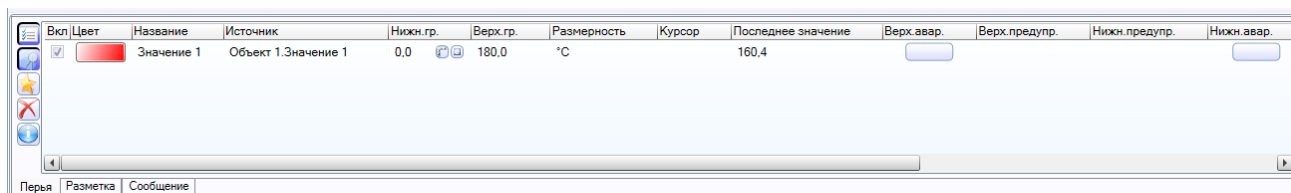
Значения – представление данных не в графическом, а в табличном виде.



Дата/Время	Значение 1 °C
08.05.2016 09:45:03:260	165,1
08.05.2016 09:45:04:274	159,4
08.05.2016 09:45:05:288	152,9
08.05.2016 09:45:06:302	145,8
08.05.2016 09:45:07:316	138,0
08.05.2016 09:45:08:330	129,6
08.05.2016 09:45:09:344	120,8

Рисунок 7-12. Панель значений

Легенда – панель для настройки добавленных перьев. В данной панели можно задать цвет пера, границы, включить аварийные и предупредительные границы.





Вкл.	Цвет	Название	Источник	Нижн. гр.	Верх. гр.	Размерность	Курсор	Последнее значение	Верх. авар.	Верх. предупр.	Нижн. предупр.	Нижн. авар.
<input checked="" type="checkbox"/>		Значение 1	Объект 1.Значение 1	0,0	 180,0	°C		160,4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Рисунок 7-13. Панель легенды

Настройки – панель для задания настроек. При этом осуществляются как общие настройки тренда, так и настройки его отдельных элементов (перьев, осей, бордюра и т. д.).

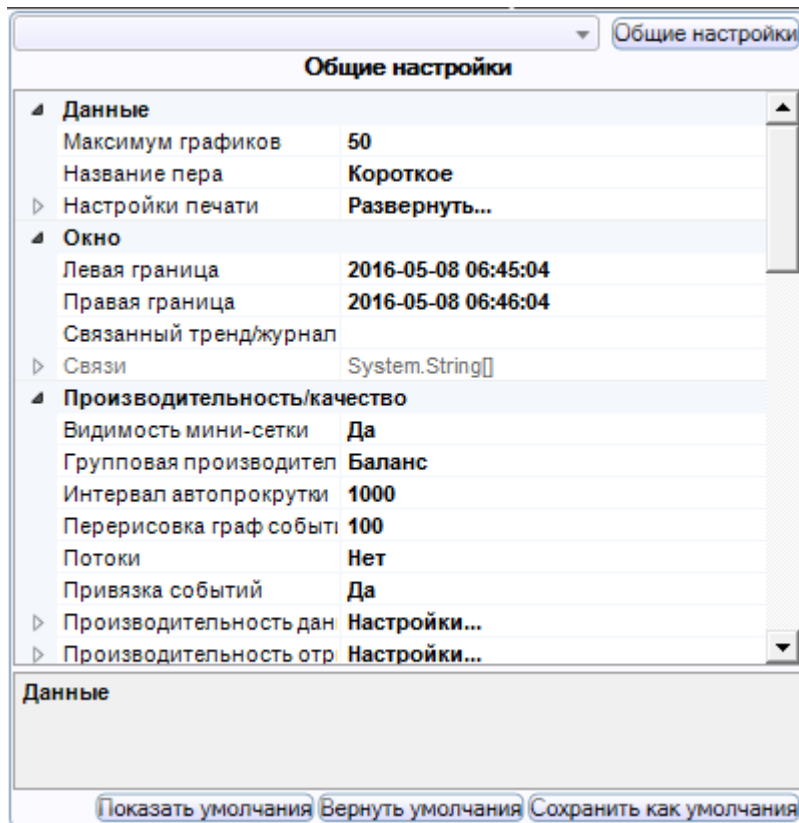


Рисунок 7-14. Панель настроек

Примечание. Прежде чем рассматривать функции и настройки тренда, отметим, что все настройки тренда можно одинаково производить как в режиме разработки, так и в режиме исполнения.

7.2.2. Уровни тренда

Модуль трендов имеет очень широкий функционал и большое количество настроек. При этом многие функции оператору могут не потребоваться, и их было бы корректнее скрыть. Для реализации этой задачи применяются **Уровни**. **Уровни** – это шаблоны настроек тренда, которые отвечают за отображение тех или иных настроек и функций.

Для изменения уровня нужно вызвать контекстное меню графической области и выбрать раздел **Уровень** (Рисунок 7-15).

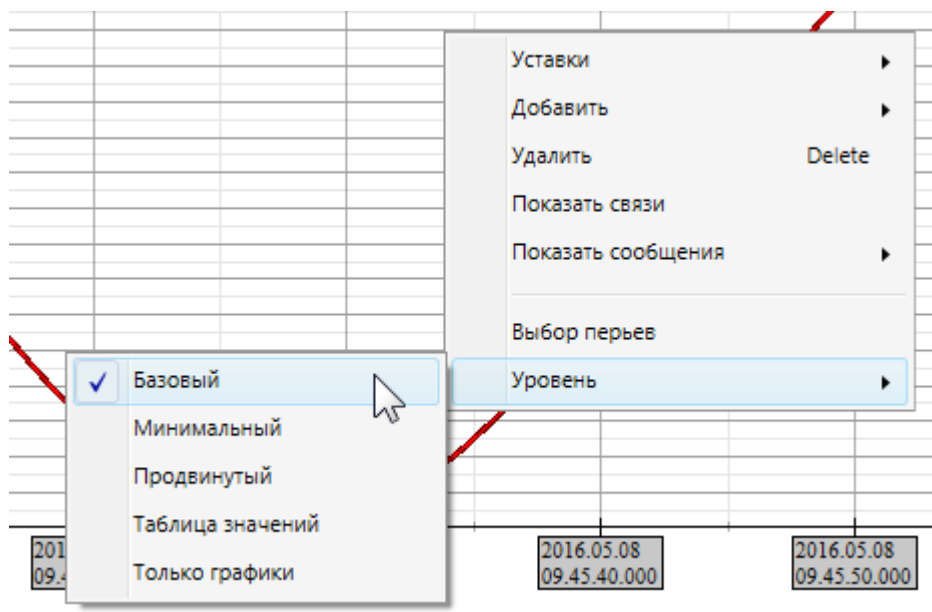


Рисунок 7-15. Изменение уровня тренда

Уровень с наименьшим количеством настроек – **Только графики**; уровень наивысшего доступа – **Продвинутый**. Произведите переключение уровней и посмотрите, как будет меняться внешний вид тренда.

Помимо стандартных уровней можно сделать собственные – для этих целей предназначен редактор уровней. Создание уровня происходит за счет усечения функций шаблонного уровня.

Рекомендуем подобрать уровень (или сделать собственный), подходящий по функциональности для оператора, и запретить переключение на другие уровни через права доступа.

7.2.3. Добавление перьев

Рассмотрим процесс добавления перьев на тренд. Данную операцию можно выполнить двумя способами.

Первый способ – просто перетащить нужную переменную из дерева объектов в поле графической области.

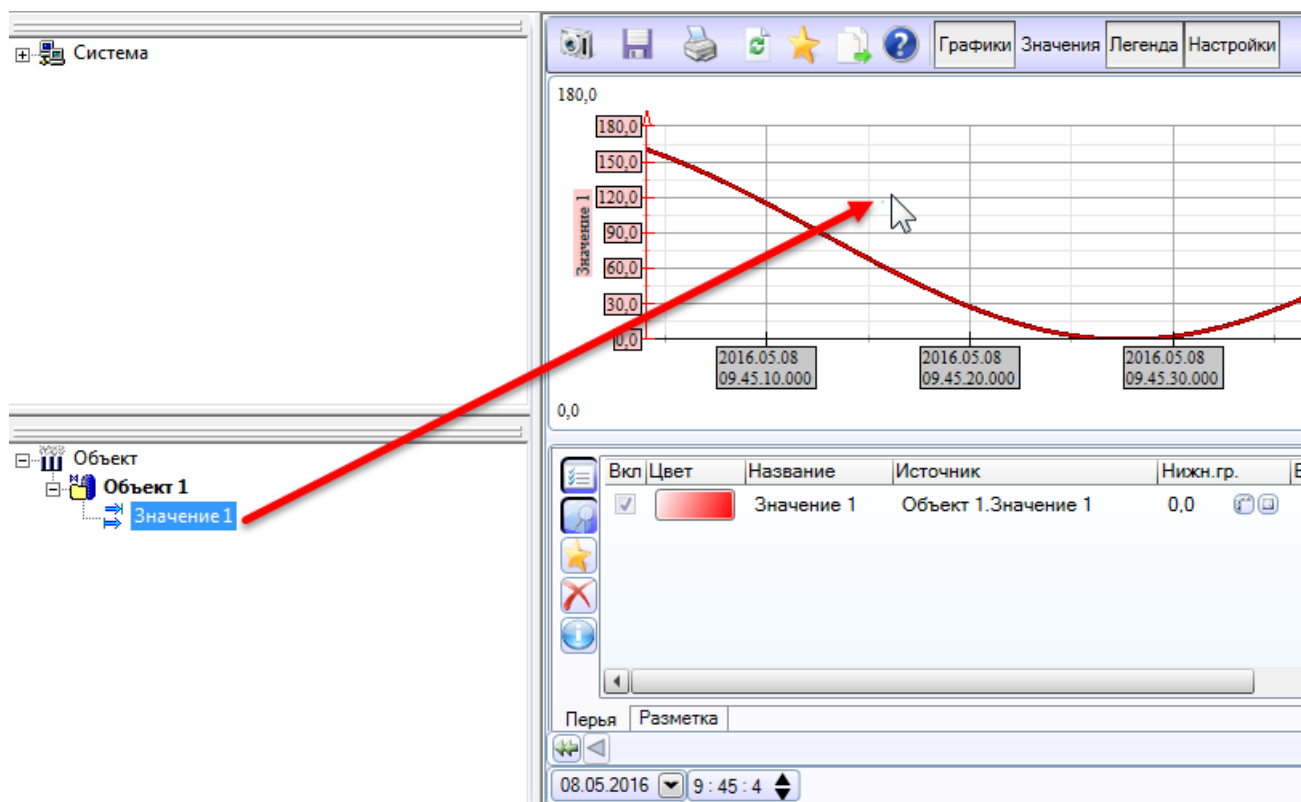


Рисунок 7-16. Добавления пера в тренд перетаскиванием переменной

При этом если у переменной не была включена архивация, то она будет включена автоматически. А если архивация уже включена и переменная имеет архив, то архивные данные сразу же отобразятся на тренде. Данный способ подходит, если нужно перетащить небольшое количество перьев на тренд.

Если же на одном тренде будет присутствовать очень много перьев, то можно воспользоваться вторым способом – использовать окно **Выбор переменных**. Данное окно вызывается при нажатии на кнопку со звездочкой (в тулбаре или в панели легенды).

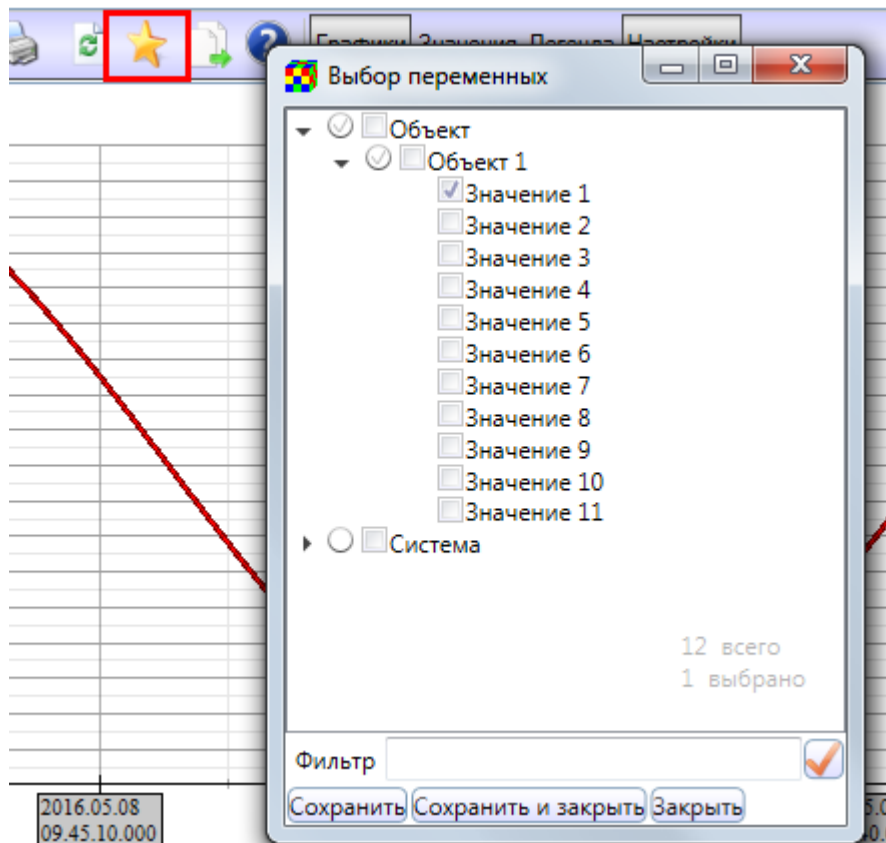


Рисунок 7-17. Добавление пера выбором из окна

В данном окне вам нужно отметить галочками перья, которые необходимо добавить на *тренд*, и нажать кнопку **Сохранить**. При этом применить фильтр по именам.

Кроме того, видимость переменной (а также объектов) в данном окне определяется флагом **Разрешение на просмотр в тренде** на закладке **Общие** переменной.

Рисунок 7-18. Настройка видимости пера из тренда

Данное окно можно вызывать в том числе и в режиме исполнения. То есть, правильно настроив отображение определенных переменных в окне, можно предоставить оператору возможность самостоятельно выбирать отображаемые на экране перья (для этих целей можно сделать отдельный тренд, а в остальных запретить добавление новых перьев через права доступа).

7.2.4. Навигация по архивным данным

По умолчанию в режиме исполнения тренд работает в режиме автопрокрутки – постоянно смещается, тем самым отображая на графике актуальные данные (за установленный интервал развертки).

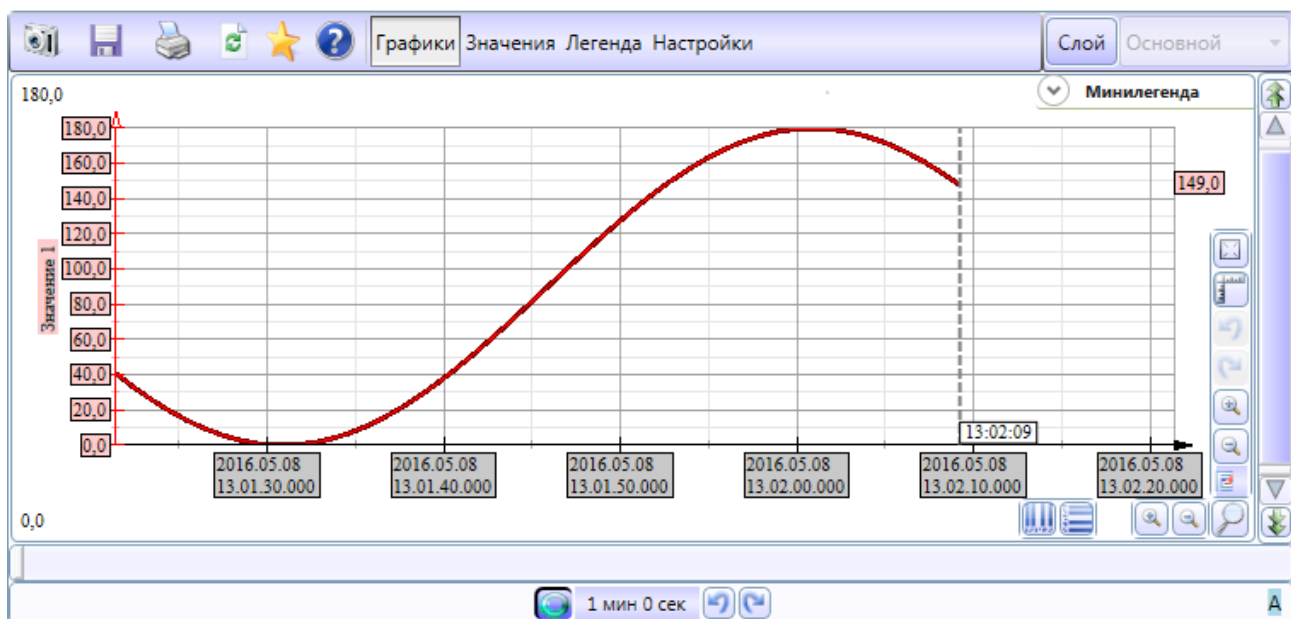


Рисунок 7-19. Тренд в режиме автопрокрутки

Для смены интервала развертки нужно нажать на текущий диапазон и выбрать нужное значение в списке.

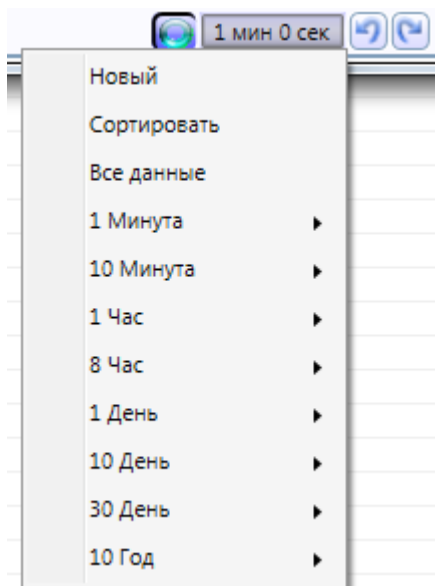


Рисунок 7-20. Смена диапазона развертки

Если необходимо посмотреть не текущие, а архивные данные, то нужно отключить автопрокрутку. При этом активируются инструменты для навигации по данным: скроллер, кнопки смещения, контролы для ручного задания начала и конца отображаемых данных.

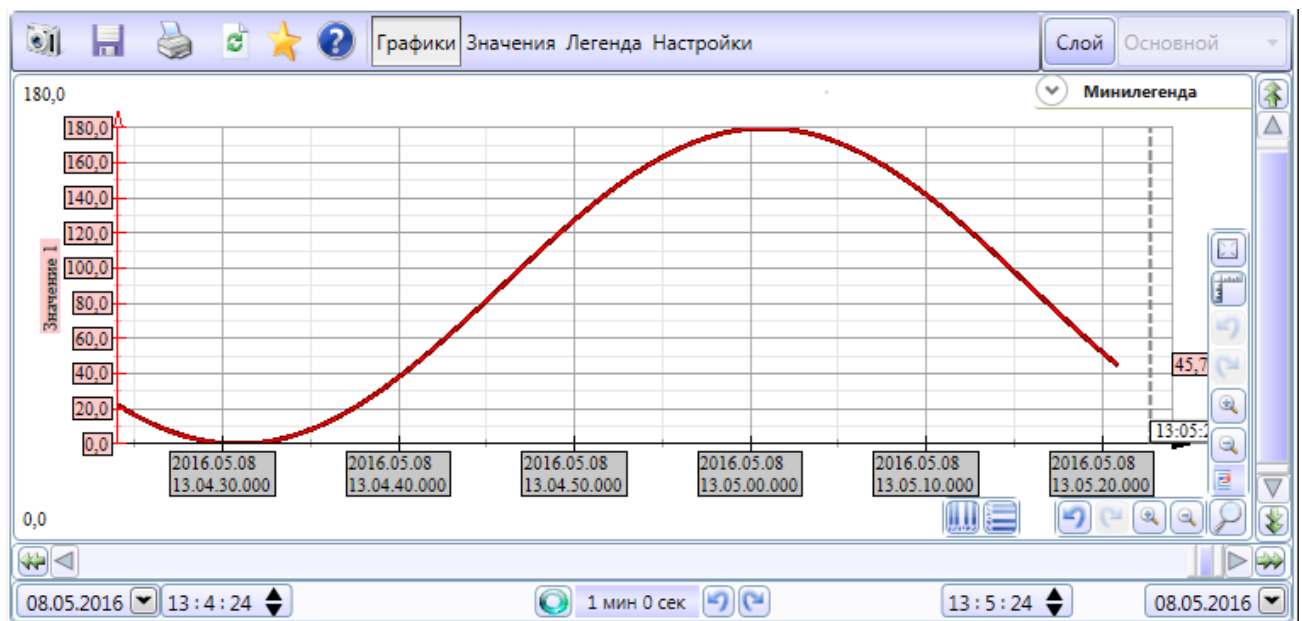


Рисунок 7-21. Тренд в режиме просмотра архивных данных

Попробуйте осуществить прокрутку тренда разными способами.

7.2.5. Вертикальный и горизонтальный курсоры

В процессе работы операторам и технологам часто требуется получить точные значения в определенный момент времени – или, наоборот, получить время пересечения переменной определенной границы.

Для решения данной задачи в трендах есть вертикальный (Рисунок 7-22) и горизонтальный курсоры (Рисунок 7-23).

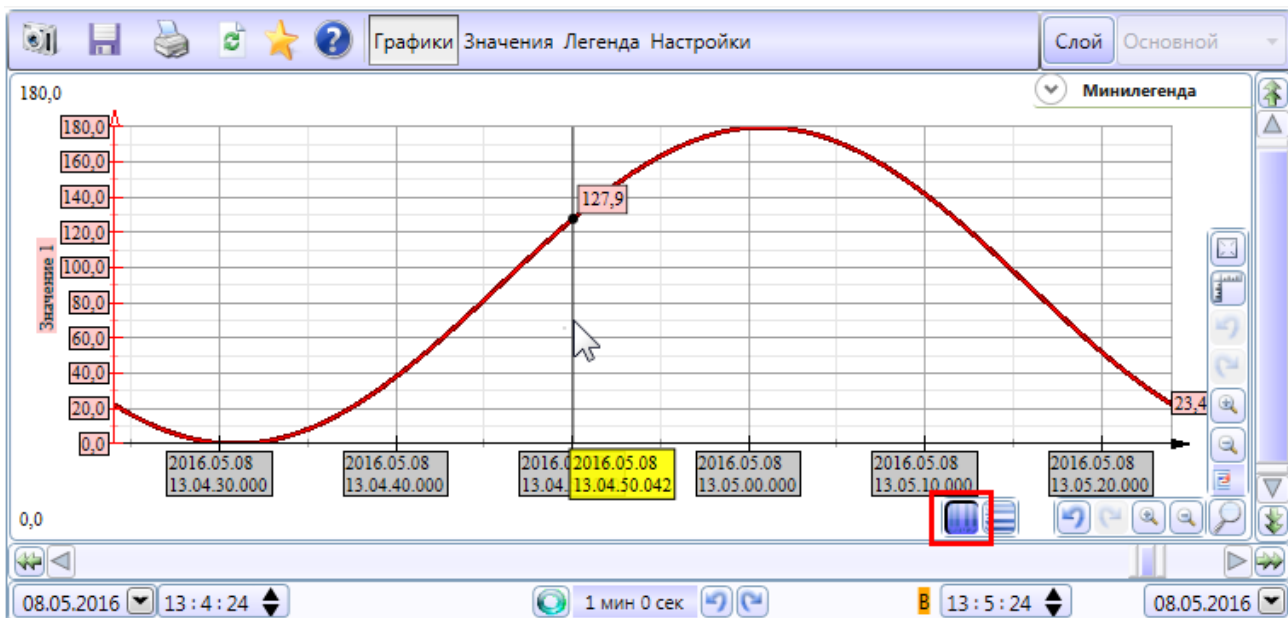


Рисунок 7-22. Режим вертикального курсора

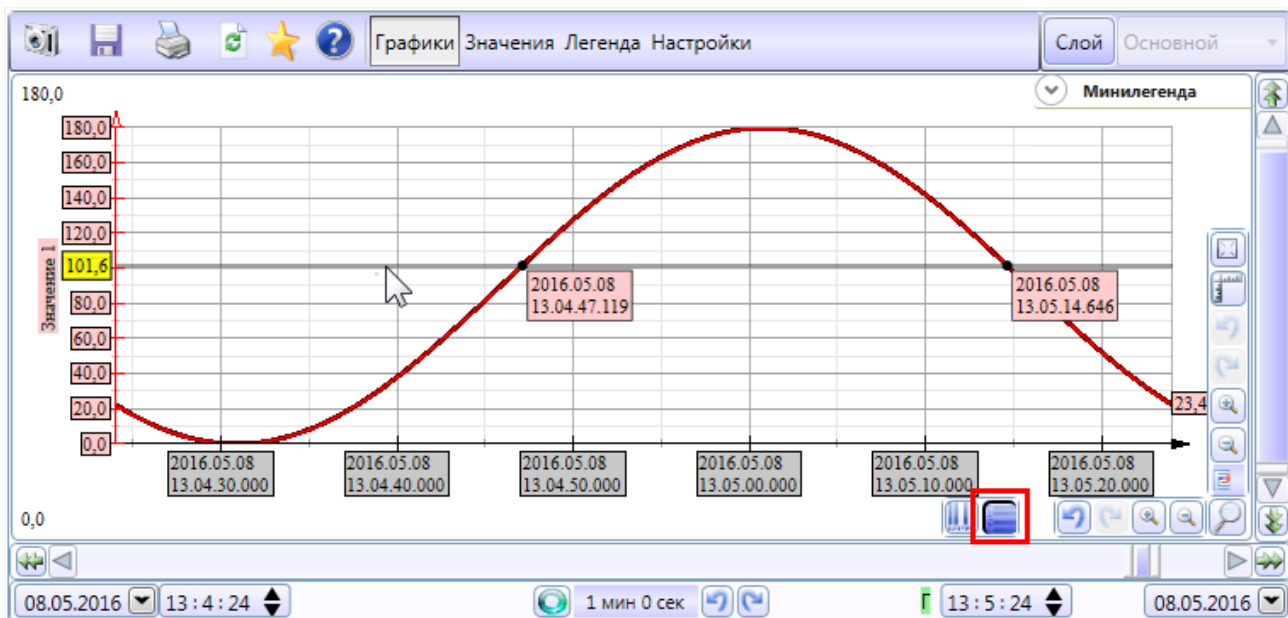


Рисунок 7-23. Режим горизонтального курсора

7.2.6. Отображение аварийных и предупредительных границ

Если переменная имеет шкалу, то у нее могут быть заданы верхние и нижние аварийные и предупредительные границы. Данные границы также можно отобразить на тренде. Для этого нужно активировать панель легенды и у нужного пера нажать на кнопки в столбцах **Верхнее аварийное**, **Нижнее аварийное**, **Верхнее предупредительное** или **Нижнее предупредительное** (Рисунок 7-24). На тренд будет добавлена горизонтальная линия,

область пересечения переменной данной линии будет залита цветом для лучшего восприятия.

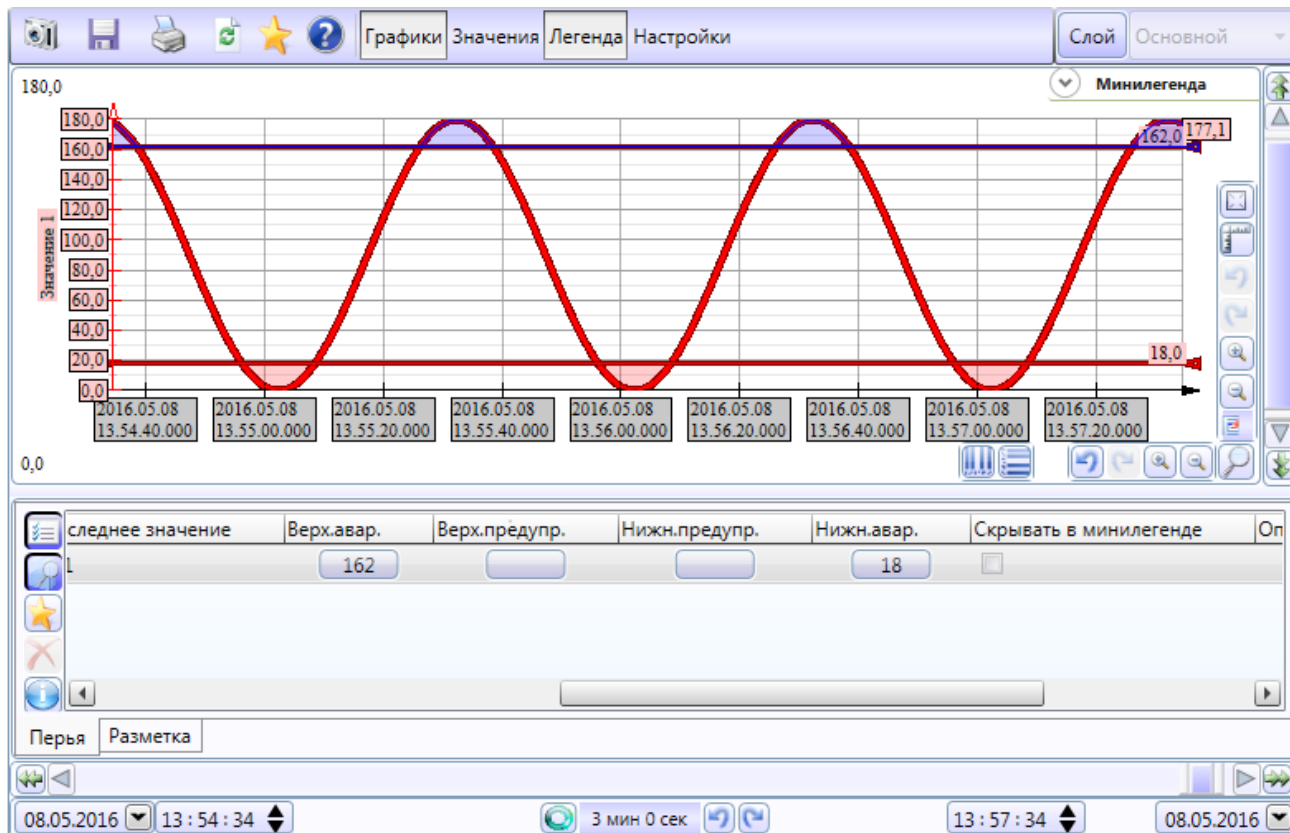


Рисунок 7-24. Режим отображения аварийных границ

Удаление линий аварийных и предупредительных границ производится только при открытой панели настроек повторным нажатием на кнопку в панели легенды или через контекстное меню линии.

7.2.7. Закладки

Также в процессе работы зачастую требуется отметить определенный момент времени, чтобы потом можно было быстро вернуться к нему (например, при разборе аварийной ситуации). Для этого в *тренд* можно добавить закладку. Данная операция выполняется через контекстное меню.

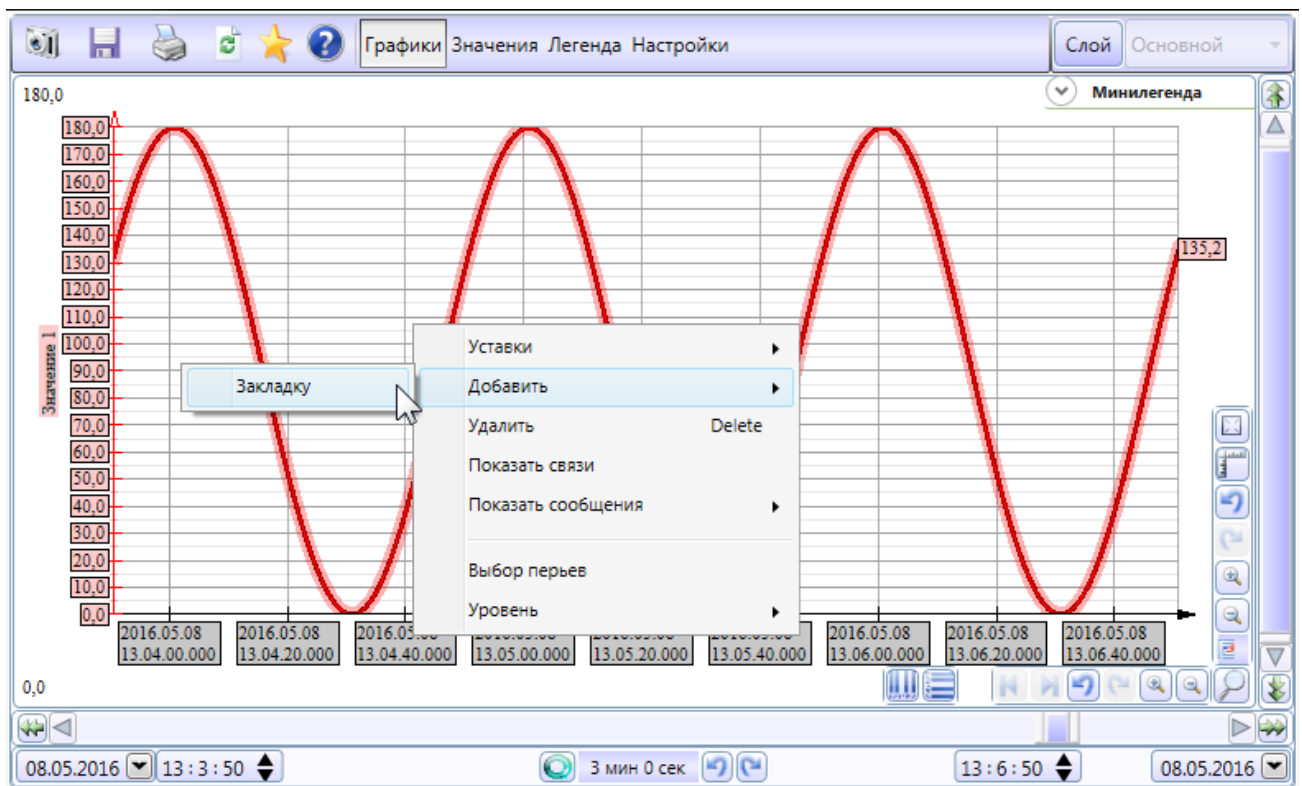


Рисунок 7-25. Добавление закладки

В этом месте будет добавлена закладка. В нижней части *тренда* появятся кнопки для навигации по закладкам – теперь в любой момент можно вернуться к закладке.

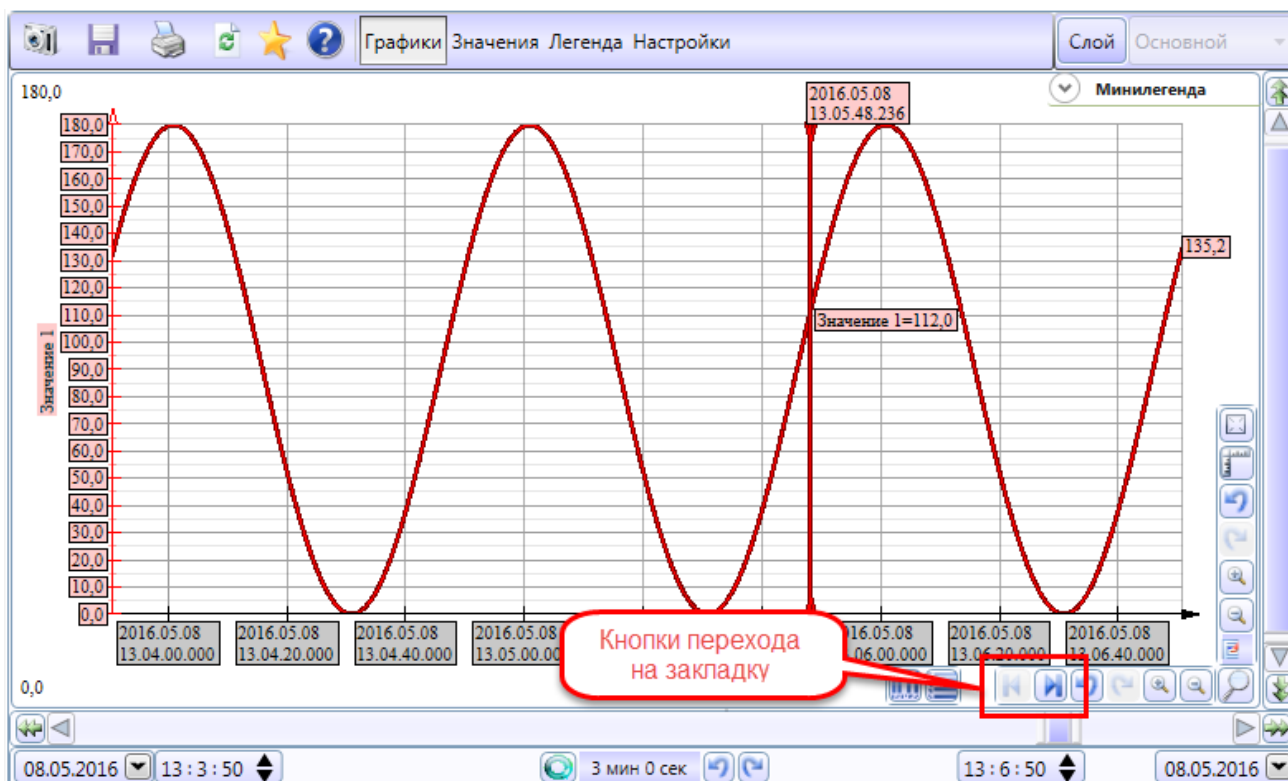


Рисунок 7-26. Кнопка для перехода к закладке

Удаляется закладка также через контекстное меню.

7.2.8. Операции с перьями

В тренде можно также выполнять различные операции с перьями: вычислять сумму, разность между значениями, выводить зависимость и корреляцию нескольких перьев. Все эти действия производятся после добавления нужной операции.

К примеру, выведем график разности между двумя перьями. Изначально у нас должен быть тренд, в котором уже добавлены два пера. В продвинутом уровне через контекстное меню добавим операцию **Разность графиков** (Рисунок 7-27).

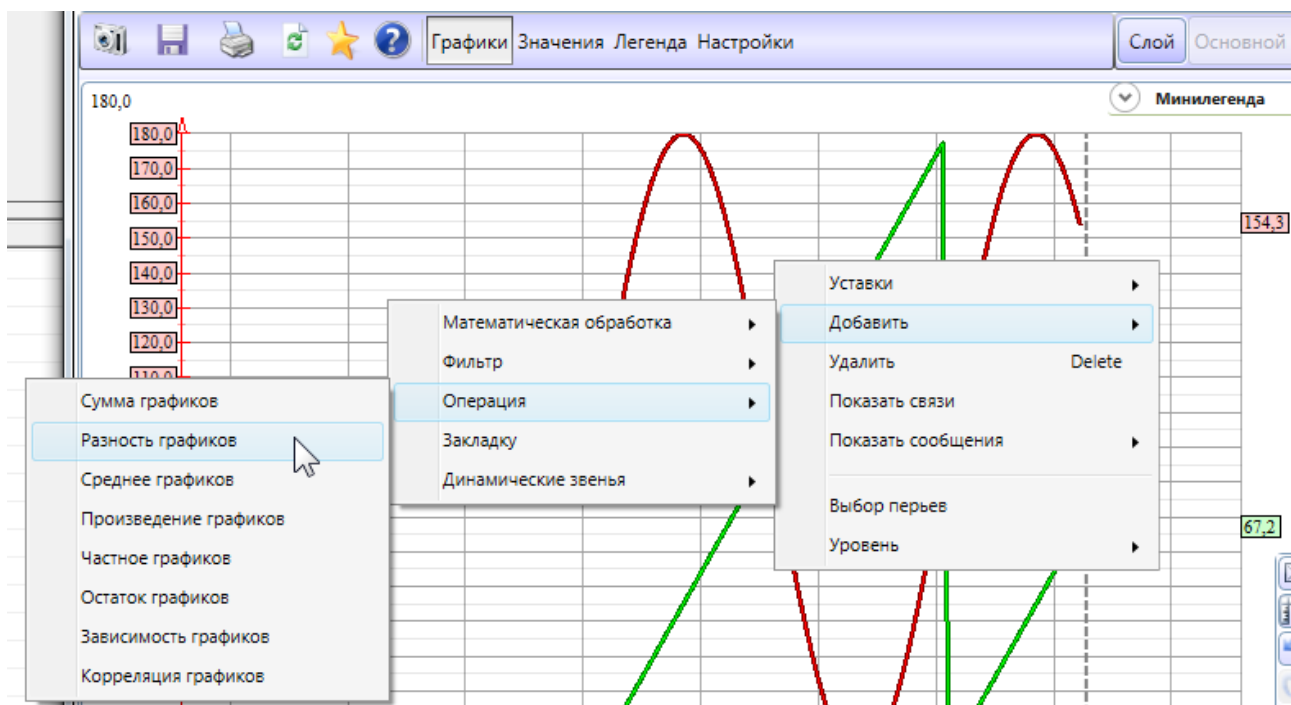


Рисунок 7-27. Добавление операции «Разность графиков»

Если активировать панель легенды, можно увидеть, что добавился новый график.

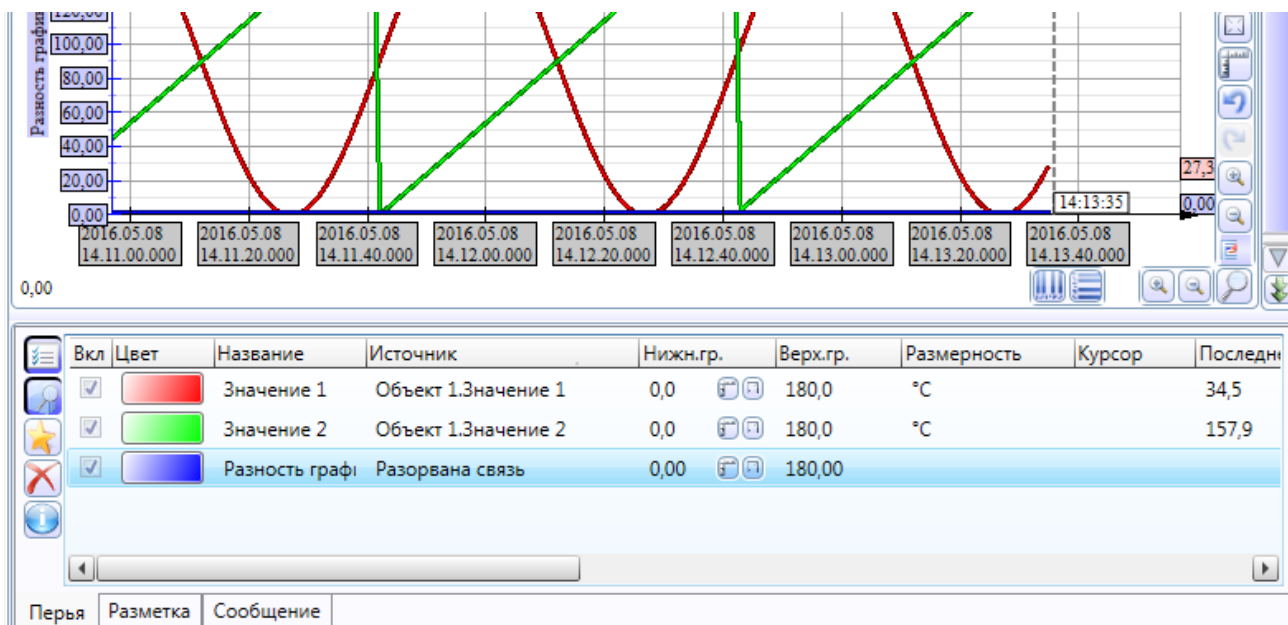


Рисунок 7-28. Панель легенды с графиком «Разность»

Выделим его и в панели настроек зададим **Аргумент X** и **Аргумент Y**.

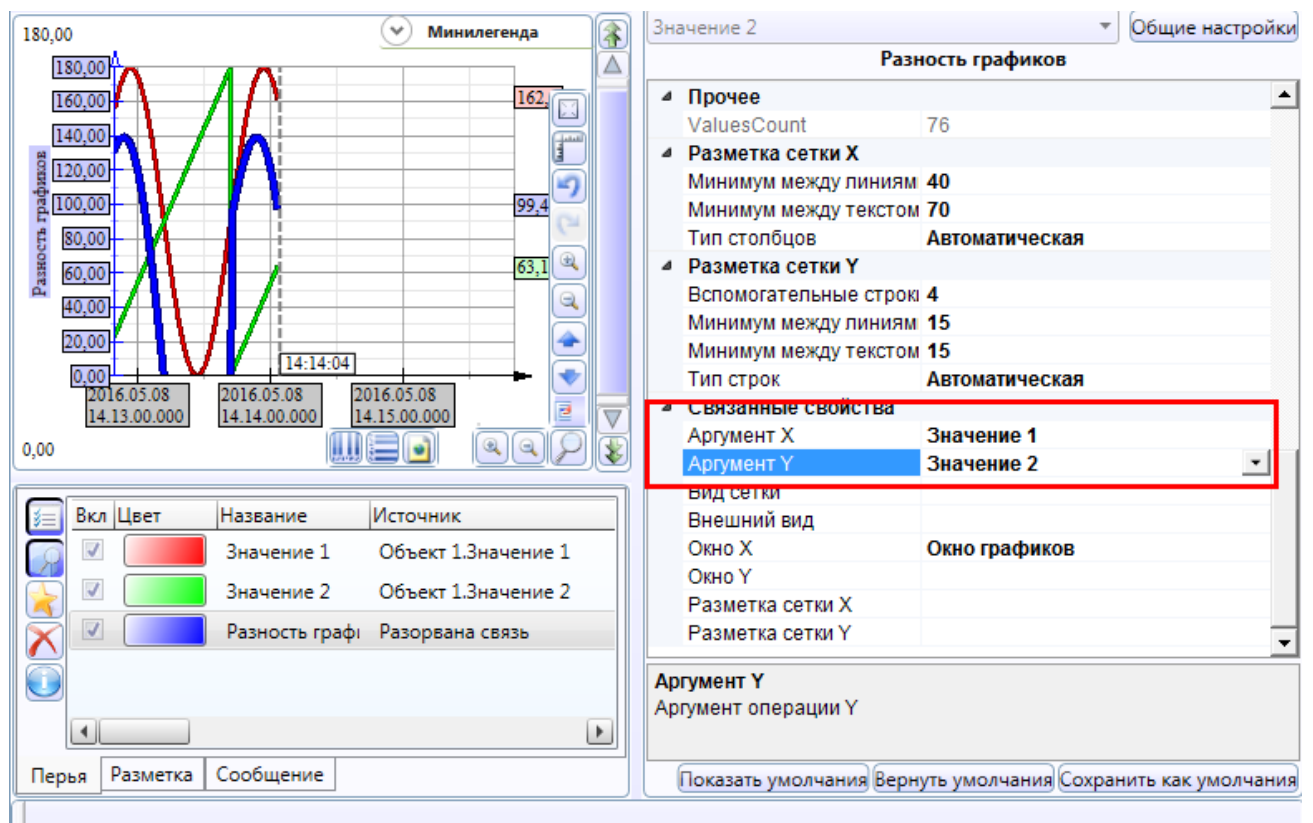


Рисунок 7-29. Настройка аргументов графика

Скроем в мини-легенде два основных пера и оставим только перо с разностью.

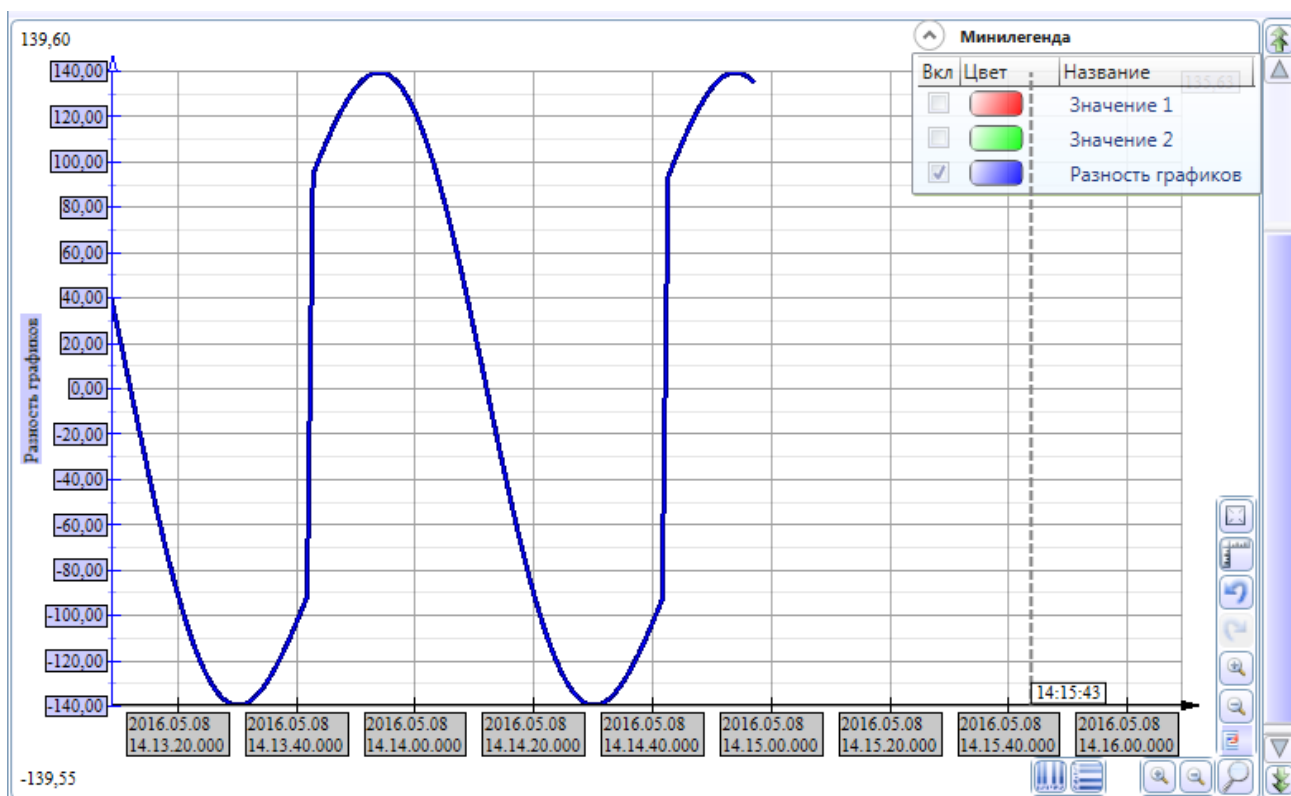


Рисунок 7-30. Выключение остальных перьев в мини-легенде

Аналогичным образом можно строить графики на основе других операций.

7.2.9. Действия – возврат значений из тренда

Иногда нужно не только отображать значения на *тренде*, но и иметь возможность вернуть из *тренда* какие-либо значения. Например, если оператору надо построить *отчет* на основе данных *тренда*, ему необходимо на *тренде* отметить начало и конец – за этот интервал и должен быть сформирован *отчет*.

Сначала подготовим в *дереве объектов* переменные, в которые мы будем записывать результат действия. Создадим отдельный объект и добавим в него три команды:

Значение – тип *вещественный дв. точности*, в нее будет записано значение переменной в указанный момент времени.

Время – тип *время*, в нее будет записана метка времени значения.

Импульс – тип *дискретный*, в нее будет сформирован дискретный импульс в момент выбора действия. По этому импульсу можно будет понять, что оператор выполнил действие, и обработать его.

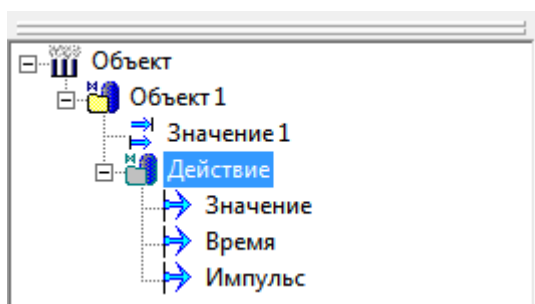


Рисунок 7-31. Объект «Действия» с набором команд

Откроем *тренд*, выделим перо, для которого мы хотим создать действие, и активируем панель настроек. Найдем раздел *Действия*, настройку *Кол-во действий*, зададим в ней 1. Создастся *Действие 1*.

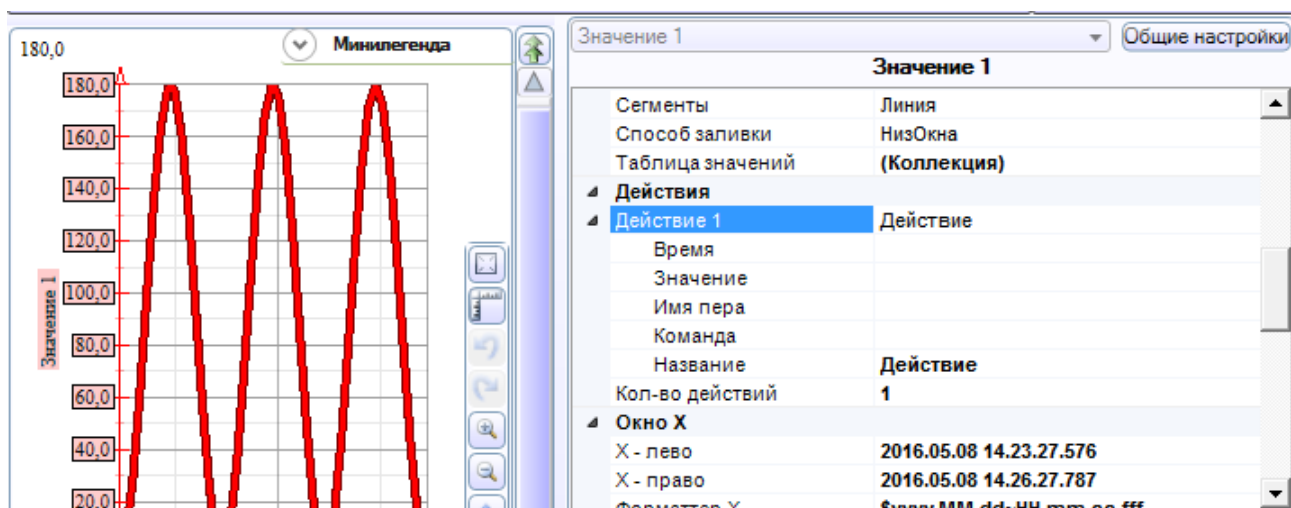


Рисунок 7-32. Создание нового действия

У действия есть 4 поля: *Время*, *Значение*, *Имя пера* и *Команда*. Перетащим в них наши команды.

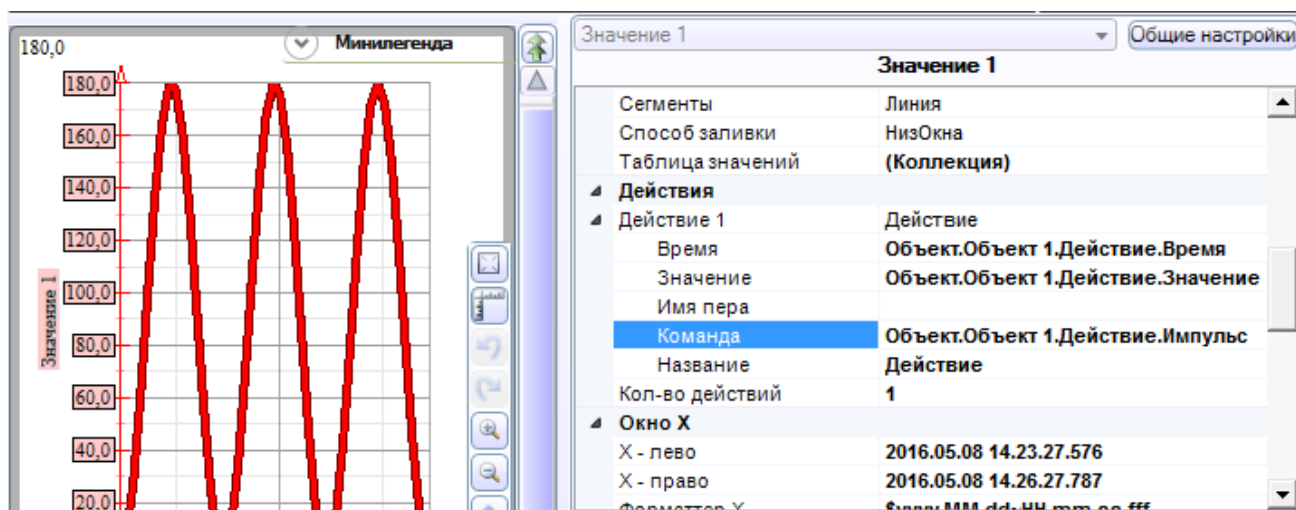


Рисунок 7-33. Привязка действия к командам из объекта

Теперь запустим режим исполнения, выделим перо. Для удобства можно также активировать вертикальный курсор, чтобы точно видеть значение величины и ее время. Вызовем контекстное меню пера и выберем в меню созданное нами действие.

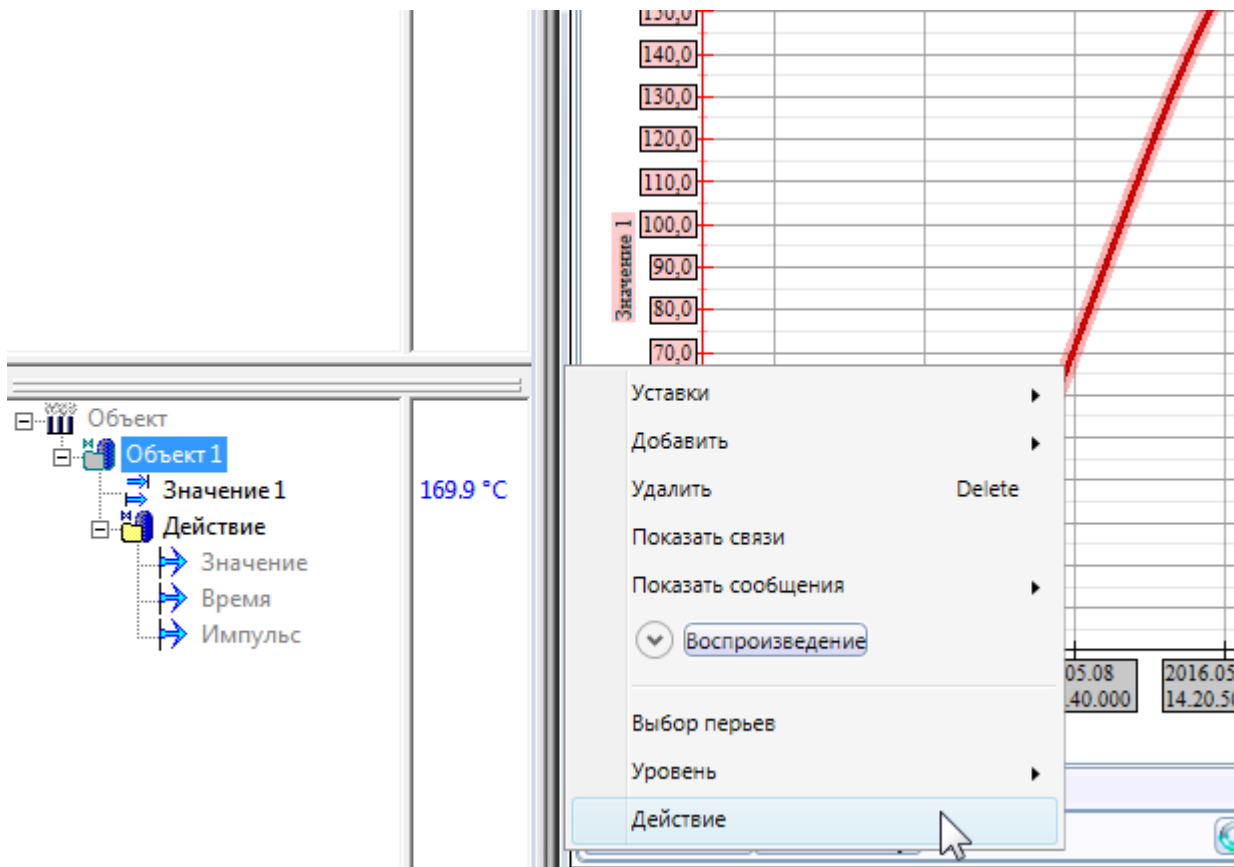


Рисунок 7-34. Вызов действия

Значения пера в указанный момент времени записались в команды в дереве объектов.

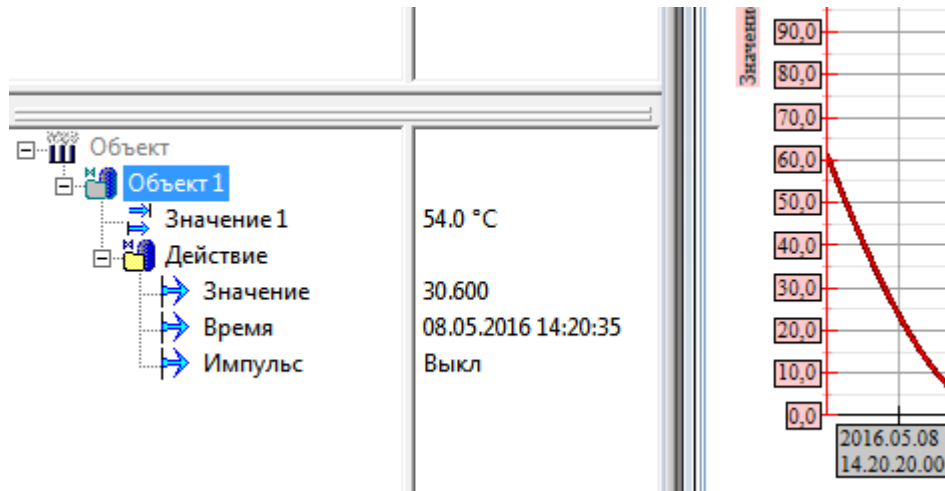


Рисунок 7-35. Команды с присвоенными значениями

Теперь с ними можно работать, как с обычными переменными.

Аналогичным образом можно создать любое количество действий и обрабатывать получаемые от них значения.

7.2.10. Расширенный функционал

Рассмотренные функции – лишь малая часть модуля трендов. В нем также есть следующие возможности:

- Добавлять несколько осей с различными настройками и положением.
- Сохранять данные в графические форматы и CSV.
- Комбинированный режим тренда и журнала – отображение сообщений на тренде.
- Воспроизведение графиков – просмотр графика «в движении».
- Связывание трендов – несколько разных трендов связываются по времени, т. е. изменение настроек интервала, положения автоматически приводит к изменению связанного с ними тренда.
- Редактор уровней сложности – отдельный редактор, в котором можно создать собственный уровень.

Система архивации и тренды

Все перечисленные функции подробно разобраны в видеопримерах, которые содержатся на прилагаемом диске.

8. Система сообщений и журналы

Технологические процессы необходимо сопровождать сообщениями, чтобы оператор мог быстрее оценить ситуацию и правильно на нее среагировать. При этом сообщения могут быть различных типов: просто информационные (включение насоса, клапана), предупреждающие и аварийные. Могут быть различными и способы предоставления сообщений оператору: всплывающие окна, звуковые сигналы, журналы с представлением информации в табличном виде, уведомления по SMS и электронной почте. MasterSCADA поддерживает разбиение сообщений на определенные категории, а также различные способы вывода этих сообщений.

8.1. Категории сообщений

Технологический процесс может генерировать сообщения различных типов – это может быть как общая информация, например о переключении исполнительных механизмов, так и аварийные сообщения. Для улучшения восприятия оператором сообщения можно разделить на несколько категорий, которые затем настроить (настроить способы вывода сообщений, задать разные цвета шрифта и фона, назначить различные звуковые файлы). Настройка категорий сообщений производится на вкладке **Система – Сообщения – Категории**.

Категория	Приоритет
Авария	Высокий
Предупреждение	Средний
Отсутствие значения	Средний
Информация	Обычный
Системные ошибки	Средний
Исполнительные механизмы	Низкий
Действия оператора	Низкий

Канал
<input type="checkbox"/> Сообщение
<input type="checkbox"/> Строка статуса
<input checked="" type="checkbox"/> Журнал
<input type="checkbox"/> Принтер
<input checked="" type="checkbox"/> Архив
<input type="checkbox"/> E-mail
<input type="checkbox"/> SMS
<input type="checkbox"/> Звук

Текст: [] Фон: [] Пример текста сообщения

Добавить категорию | Удалить категорию | Звук... [] Повтор звука

Включить расширенные настройки текста

Рисунок 8-1. Вкладка категорий сообщений

На данной вкладке сразу присутствует набор системных категорий. Системные категории нельзя удалять, но можно переименовывать или создавать собственные. Каждой категории можно задать цвет текста и фона. При этом есть упрощенная настройка и расширенная. В расширенных настройках можно задать цвет текста, фона; тип шрифта, мигания для каждого из состояний сообщений:

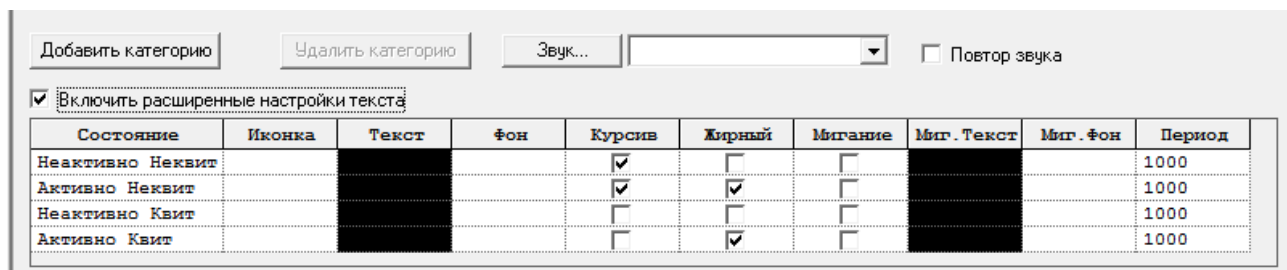


Рисунок 8-2. Расширенные настройки категорий

В разделе **Канал** задается способ вывода данной категории сообщений.

Каждый канал можно настроить – для этого существует отдельная вкладка (Рисунок 8-3).

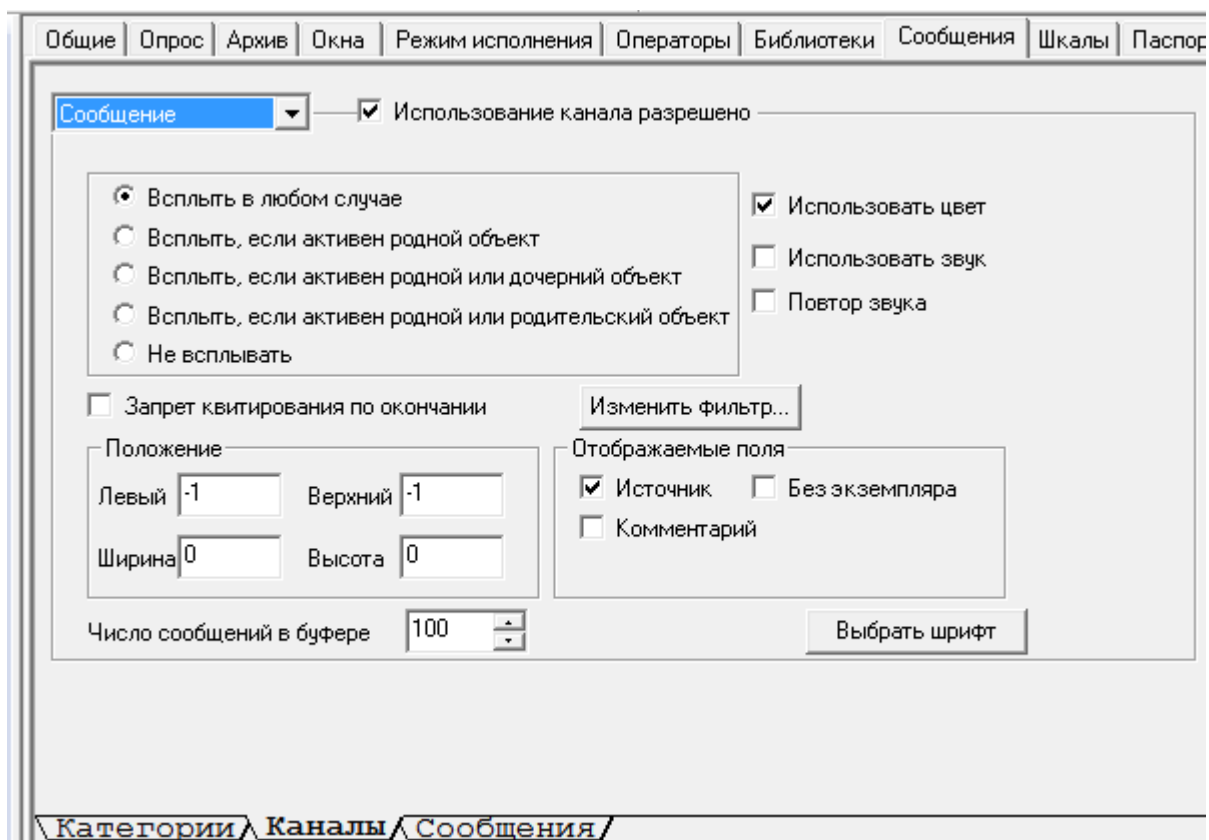


Рисунок 8-3. Вкладка настройки каналов

Рассмотрим существующие каналы сообщений.

8.1.1. Канал «Сообщение»

Данный канал представляет всплывающее окно, которое возникает, когда источник сообщения переходит в состояние **Активно**.

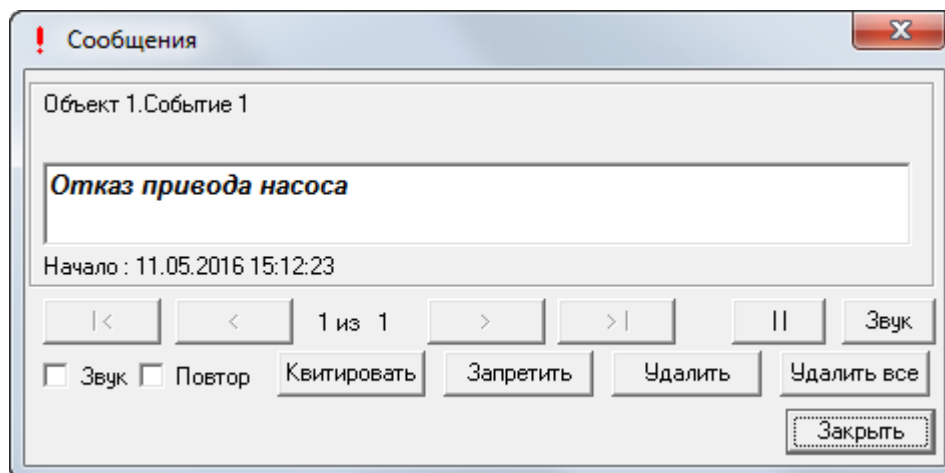


Рисунок 8-4. Внешний вид окна сообщений

При этом выводится текст самого сообщения, а также источник, сформировавший сообщение. Также можно включить комментарий к этому источнику. В окне можно осуществлять навигацию по последним сообщениям и выполнять квитирование.

В настройках канала можно задать положение окна, настройки шрифта, а также фильтрацию по источникам (например, не выводить в окно сообщения из определенных объектов). Подробное описание всех настроек данного канала есть в справке.

8.1.2. Канал «Строка статуса»

Строка статуса выводит сообщение в специальную строку в нижней части окна MasterSCADA.

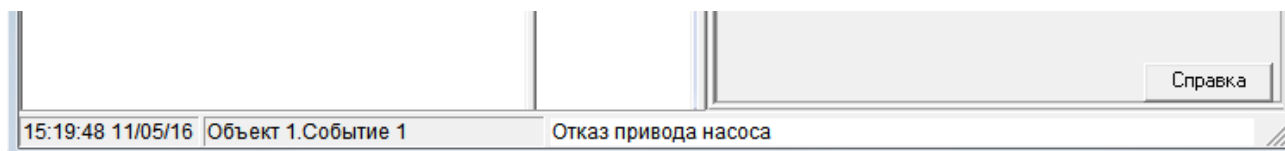


Рисунок 8-5. Канал «Строка статуса»

Поскольку обычно среда разработки скрыта от оператора, канал используется редко.

8.1.3. Канал «Принтер»

Данный канал позволяет выводить сообщения на печать. Все генерируемые сообщения (с включенным каналом) попадают в буфер и, когда формируется список на страницу, выводятся на печать. Также возможен вариант построчной печати каждого сообщения на матричном принтере.

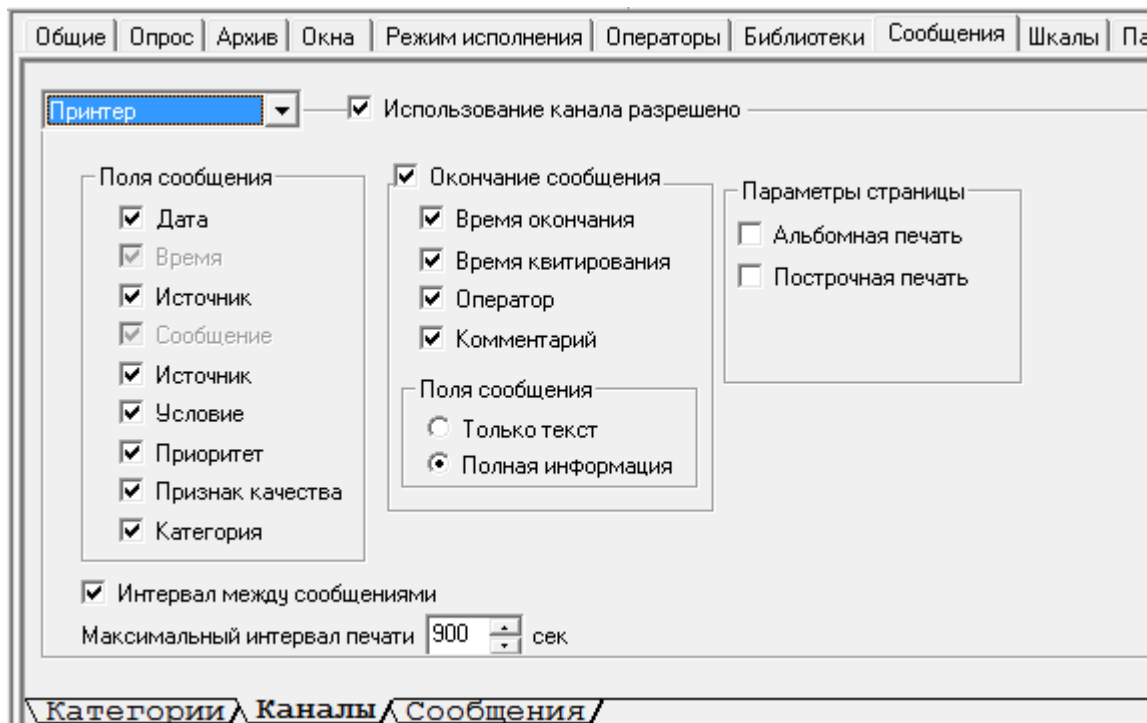


Рисунок 8-6. Настройки канала «Принтер»

8.1.4. Канал «Архив»

Сообщения, у которых включен данный канал, архивируются в базе. Данный канал обязательно должен быть включен, если планируется использовать журнал сообщений. Выключать данный канал не рекомендуется.

Как и в случае с данными, архив сообщений может писаться в файловый архив или в СУБД – это определяется настройкой **Место хранения**. Настройки полностью идентичны настройкам архивации данных.

Общие	Опрос	Архив	Окна	Режим исполнения	Действия	Сообщения	Журналы	Права доступа	Операторы	Смена				
Место хранения	Файловый архив		Вернуть умолчания											
<input type="checkbox"/> Показывать текущее состояние для подобъектов														
▲ Ограничения														
Длительность						30.00:00:00								
Максимальный суммарный объем (Мб)						3								
Объем на объект (Мб)						3								
▲ Текущее состояние														
Длительность						2.00:00:00								
Объем (Мб)						0,001678466796875								
<table border="1"> <tr> <td>Данные</td> <td>Сообщения</td> <td>Документы</td> <td>Экспорт</td> </tr> </table>											Данные	Сообщения	Документы	Экспорт
Данные	Сообщения	Документы	Экспорт											
						Применить		Отменить		Справка				

Рисунок 8-7. Вкладка настроек архивации сообщений

8.1.5. Каналы E-mail и SMS

Данные каналы являются опциональными и входят в Pro-версию MasterSCADA. Каналы позволяют производить удаленное оповещение по электронной почте и SMS-сообщением.

Чтобы настроить канал SMS, нужно указать номер COM-порта, к которому подключен модем, с которого будет производиться рассылка, телефон получателя (или получателей) и текст сообщения:

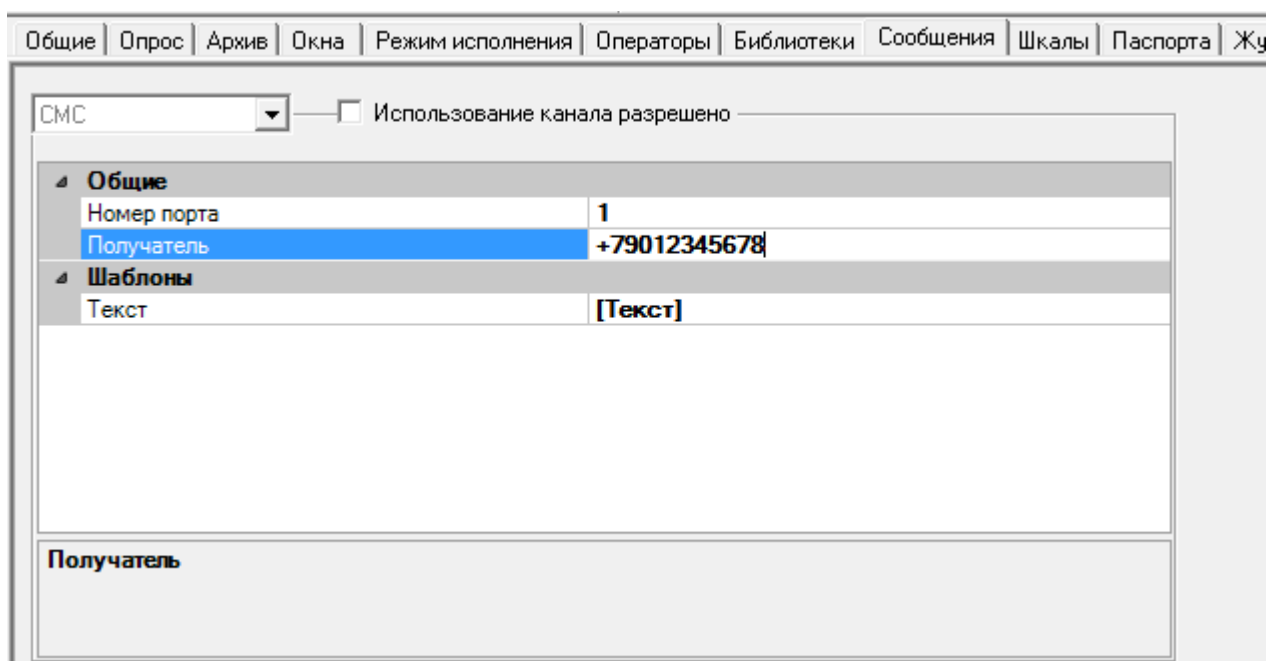


Рисунок 8-8. Настройки канала SMS

Для отправки сообщения по E-mail указываются **Имя учетной записи**, **Пароль**, **Сервер SMTP**, **Получатели** и **Текст** сообщения.

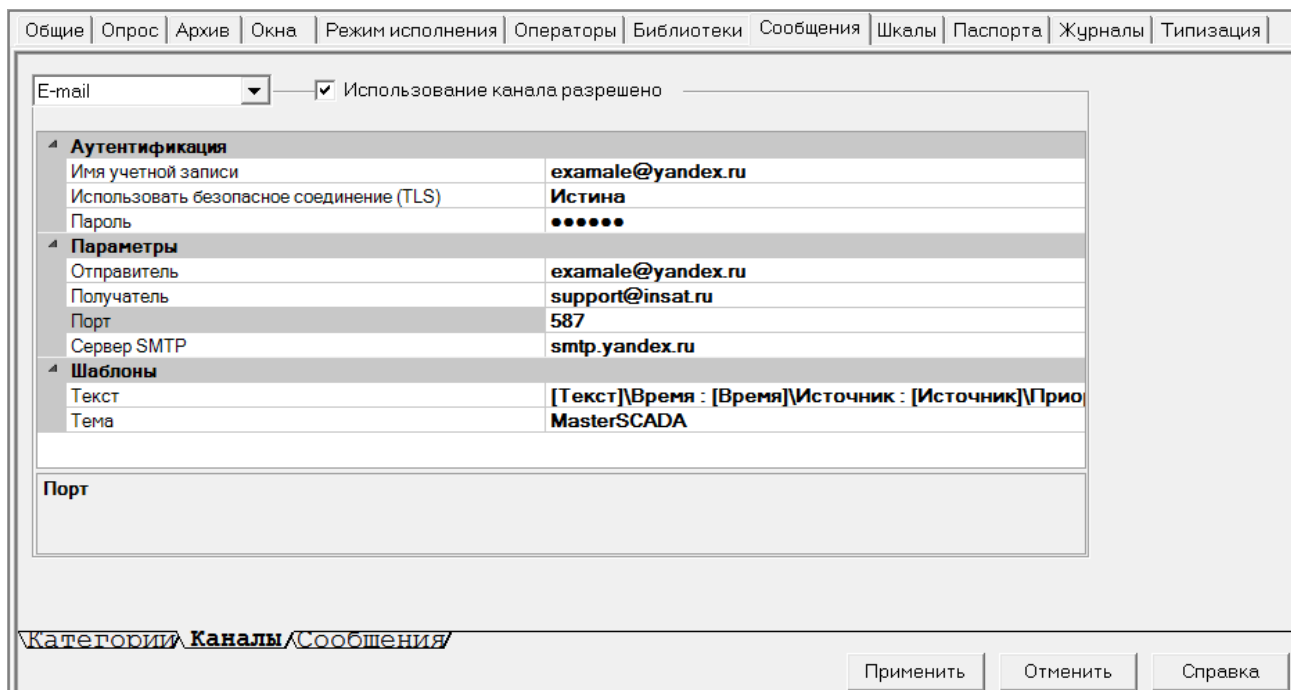


Рисунок 8-9. Настройки канала E-mail

У обоих каналов текст сообщения может состоять из статической и динамической части (то есть поясняющего текста) и значений, сформированных в режиме исполнения (текст

сообщения, канал, время, текущий оператор и т. д.). Текст сообщения формируется с помощью маски. Подробное описание настроек к обоим каналам есть в справочной системе.

8.1.6. Канал «Звук»

Данный канал предназначен для воспроизведения звуковых файлов при возникновении сообщений. Задание файла, который будет непосредственно воспроизводиться у сообщения, задается на уровне категории. В настройках же канала можно задать, проигрывать ли звук постоянно, пока оператор его не квитирует, и распределение воспроизведения звука по приоритетам сообщений.

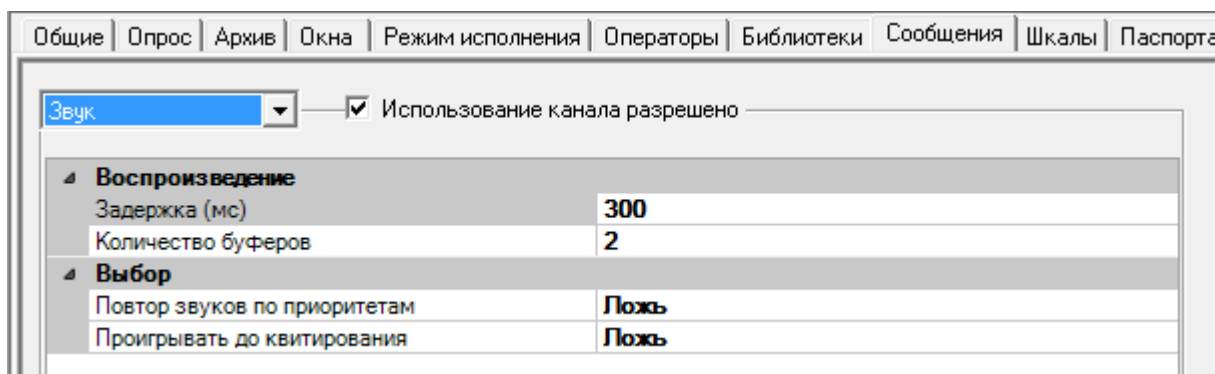


Рисунок 8-10. Настройки канала «Звук»

8.2. Журнал сообщений

Несмотря на обилие различных каналов сообщений, главным все же является канал журнал сообщений. Журналы сообщений создаются у объектов на закладке **Журналы**.

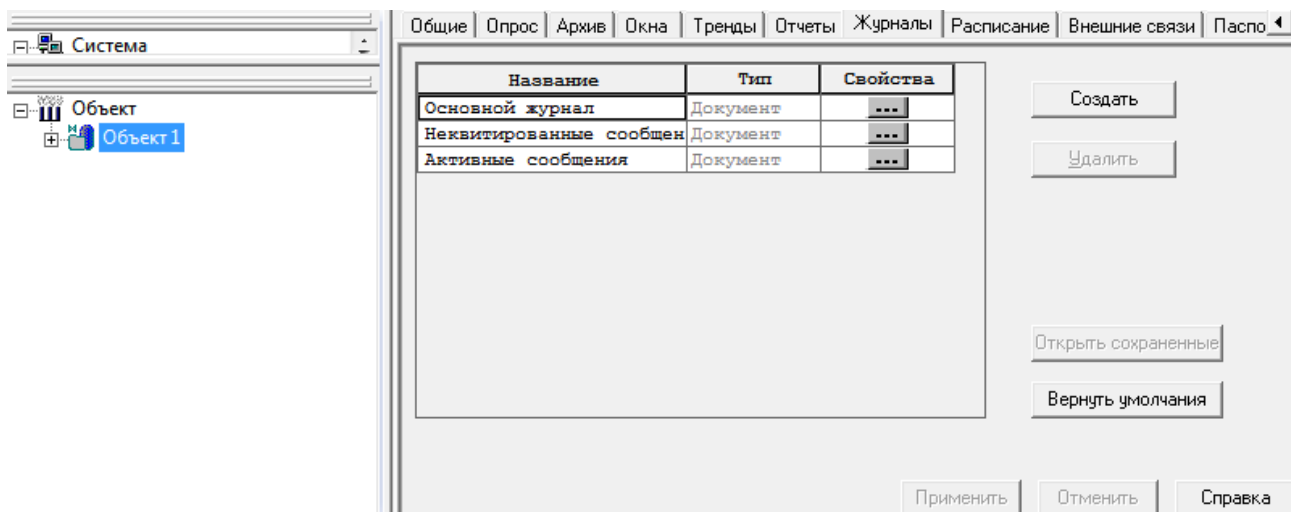


Рисунок 8-11. Вкладка с различными журналами сообщений

По умолчанию на закладке присутствует три журнала: **Основной журнал**, **Неквитированные сообщения**, **Активные сообщения**. Все три журнала одинаковы и отличаются лишь настройкой фильтра. Для примера создадим основной журнал.

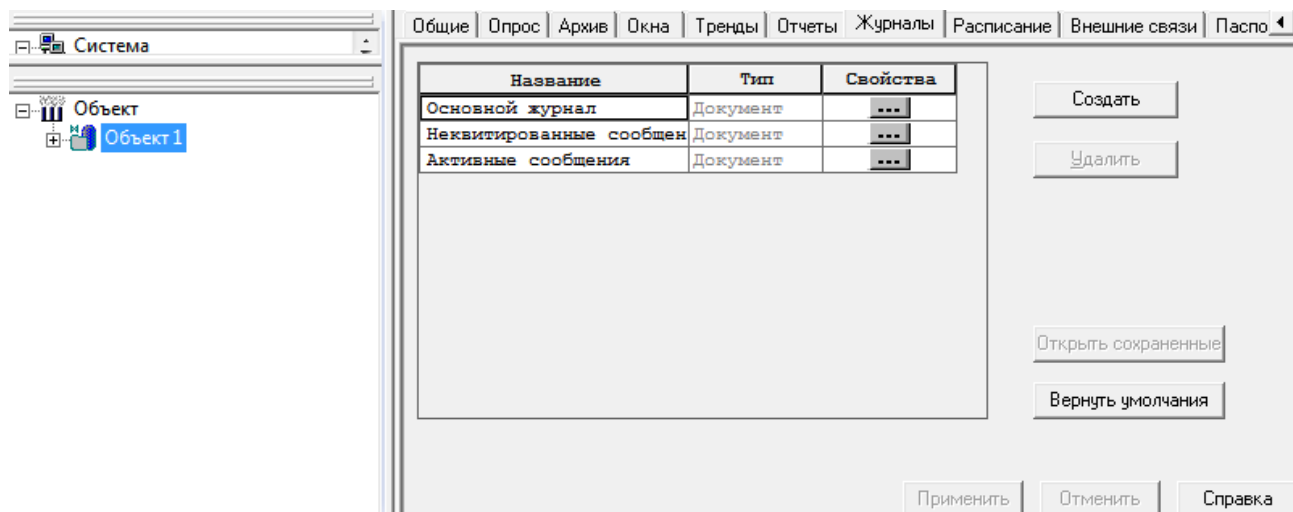


Рисунок 8-12. Создание основного журнала

Модули трендов и журнала сообщений реализованы по одной технологии – их интерфейс и принципы настроек практически идентичны. Отличие только в том, что у журнала вместо графической области находится таблица для вывода сообщений.

8.2.1. Настройка фильтра

В тренде мы вручную выбирали, какие переменные хотим видеть на графике. В журнале для настройки вывода сообщений используется фильтр, который находится на панели **Настройка**.

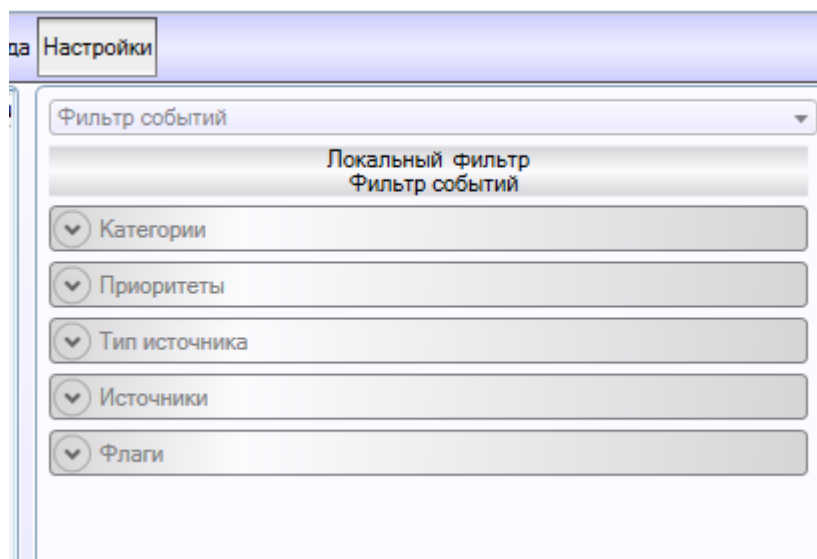


Рисунок 8-13. Настройка фильтра журнала

У фильтра событий есть 5 разделов настроек:

Категории – в настройке задается, какие категории сообщений будут отображаться в журнале. По умолчанию отображаются все категории.

Приоритеты – можно отфильтровать сообщения не по категориям, а по приоритетам. Например, выводить только сообщения с высоким приоритетом.

Тип источника – позволяет отфильтровать сообщения по типу источника, выводить сообщения о нарушении признака качества, сообщения от ФБ, сообщения от модуля **События**, системные сообщения.

Источники – в данной настройке указывается, сообщения от каких именно переменных будут попадать в журнал. По умолчанию в журнал попадают события из текущего объекта и всех его подобъектов. Эта настройка используется в большинстве случаев, но можно указать вывод сообщений от конкретных переменных.

Флаги – данной настройкой указывается статус сообщений, которые попадают в журнал: **Активные** – в журнал будут попадать только активные сообщения; **Неквитированные** – в журнал будут попадать только неквитированные сообщения; **Активные или Неквитированные** – в журнал будут попадать активные или неквитированные сообщения. Именно этой настройкой отличаются журналы Только активные или Только неквитированные на вкладке **Журналы** у объектов.

8.2.2. Настройка таблицы сообщений

В таблицу сообщений выводятся сообщения, удовлетворяющие настройкам фильтра. Каждое сообщение имеет атрибуты: **начало активности**, **конец активности**, **время квитирования**, **текст сообщения**, **источник** и т. д. Все атрибуты сообщения выводятся в отдельные столбцы таблицы. При этом состав и чередование столбцов настраиваются. Для включения/отключения видимости отдельного столбца нужно вызвать контекстное меню заголовка и выбрать/снять галочку у нужного атрибута сообщений.

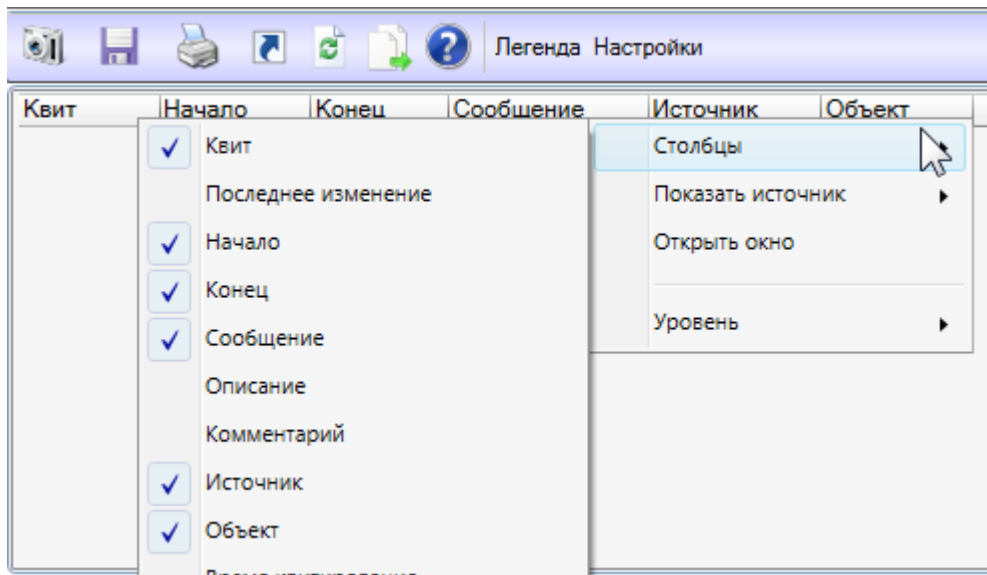


Рисунок 8-14. Настройка таблицы сообщений

Для изменения порядка столбцов нужно просто перетащить столбец и разместить его в нужном месте (Рисунок 8-15).

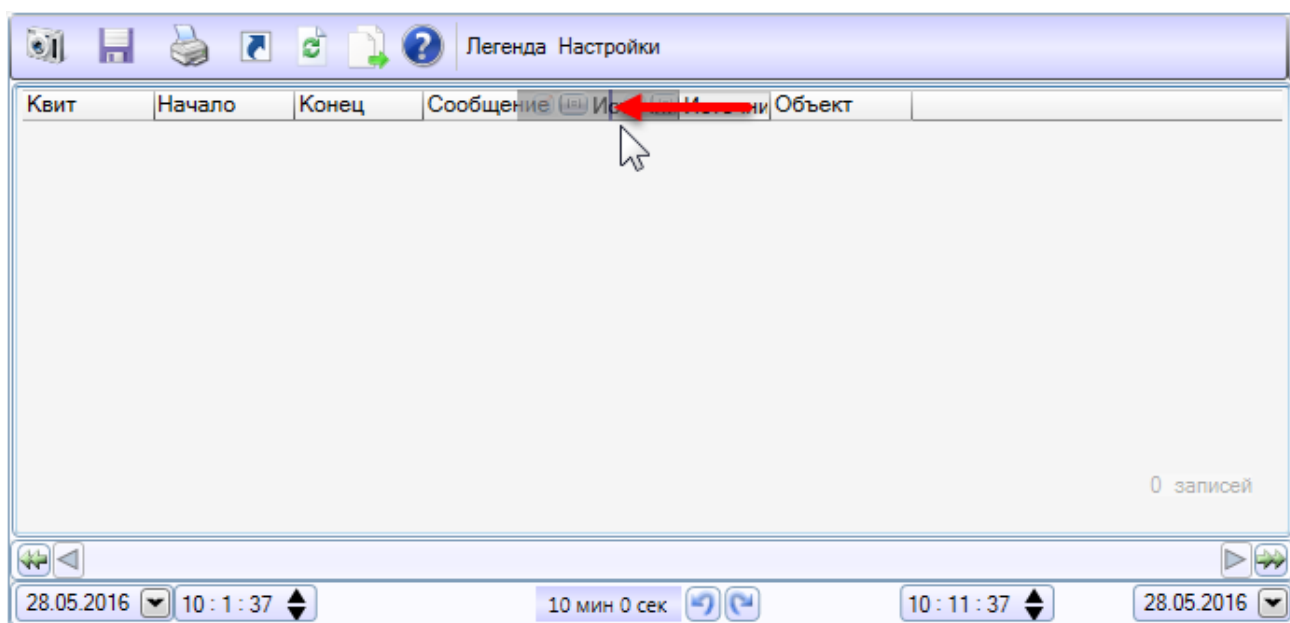


Рисунок 8-15. Изменение чередования столбцов

Для изменения размера столбца нужно привести мышь на границу столбцов и перетащить ее.

8.2.3. Навигация по архиву сообщений

Навигация по архиву сообщений в журнале происходит аналогично навигации по данным в тренде. В нижней части журнала находится панель навигации, в которой можно задать диапазон развертки (диапазон времени, за который будут выведены сообщения), переместить положение скроллера на нужный диапазон. Сообщения за данный временной диапазон и будут выведены в таблицу. Если сообщений много и на страницу они не помещаются, то в правой части появляется полоса прокрутки.

8.2.4. Фильтрация сообщений по строке

Выведенные в таблицу сообщения можно дополнительно отфильтровать по наличию определенной строки – например, вывести только сообщения со словом «Максимальное» (*Максимальное аварийное, Максимальное предупреждающее*). Для этого нужно привести мышь на заголовок столбца – появится кнопка **Фильтровать столбец**.

Начало	Конец	Сообщение	Источник
05.2016 09:35:42	30.05.2016 09:35:46	Максимальное предупрежда	Команда 1
05.2016 09:35:30	30.05.2016 09:35:42	Максимальное аварийное	Команда 1
05.2016 09:35:27	30.05.2016 09:35:33	Минимальное предупрежда	Команда 2
05.2016 09:35:27	30.05.2016 09:35:30	Максимальное предупрежда	Команда 1

Рисунок 8-16. Вызов окна фильтрации сообщений

Нажатие на эту кнопку приведет к появлению в столбце дополнительного поля – в это поле и нужно ввести текст для фильтра.

Сообщение	Источник
173	
Максимальное	

Рисунок 8-17. Поле для ввода фильтра

Сообщения будут отфильтрованы.

Квит	Начало	Конец	Сообщение	Источник	Объект
<input checked="" type="checkbox"/>	30.05.2016 09:36:15	30.05.2016 09:36:21	Максимальное аварийное	Команда 2	Объект 1
<input checked="" type="checkbox"/>	30.05.2016 09:36:09	30.05.2016 09:36:15	Максимальное предупрежда	Команда 2	Объект 1
<input checked="" type="checkbox"/>	30.05.2016 09:35:42	30.05.2016 09:35:46	Максимальное предупрежда	Команда 1	Объект 1
<input checked="" type="checkbox"/>	30.05.2016 09:35:30	30.05.2016 09:35:42	Максимальное аварийное	Команда 1	Объект 1
<input checked="" type="checkbox"/>	30.05.2016 09:35:27	30.05.2016 09:35:30	Максимальное предупрежда	Команда 1	Объект 1
<input checked="" type="checkbox"/>	30.05.2016 09:35:01	30.05.2016 09:35:04	Максимальное предупрежда	Команда 1	Объект 1
<input checked="" type="checkbox"/>	30.05.2016 09:34:49	30.05.2016 09:35:01	Максимальное аварийное	Команда 1	Объект 1
<input checked="" type="checkbox"/>	30.05.2016 09:34:46	30.05.2016 09:34:49	Максимальное предупрежда	Команда 1	Объект 1

Рисунок 8-18. Результат фильтрации

8.2.5. Открытие трендов и переходы из журнала

Источник, вызвавший сообщение, можно посмотреть на тренде – например, посмотреть, как менялось значение температуры в момент возникновения сообщения.

Для этого у нужного сообщения нужно вызвать контекстное меню **Показать источник**. Будет открыто окно тренда, в котором будет отображен график переменной.

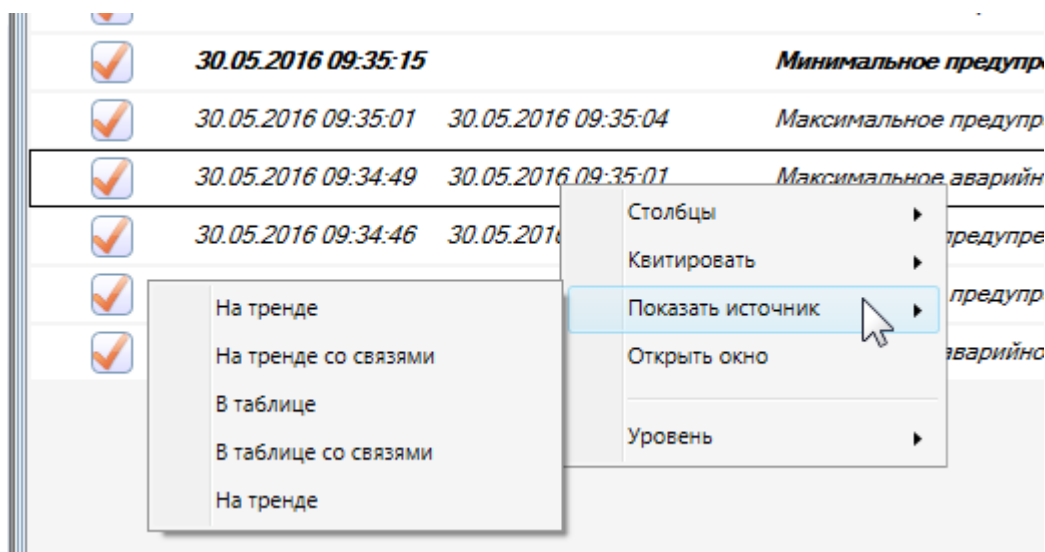


Рисунок 8-19. Вызов тренда из журнала

Также из журнала можно выполнить переход на определенное окно – например, при возникновении аварийного сообщения от клапана перейти на мнемосхему, где этот клапан находится. Такой переход также выполняется через контекстное меню сообщения.

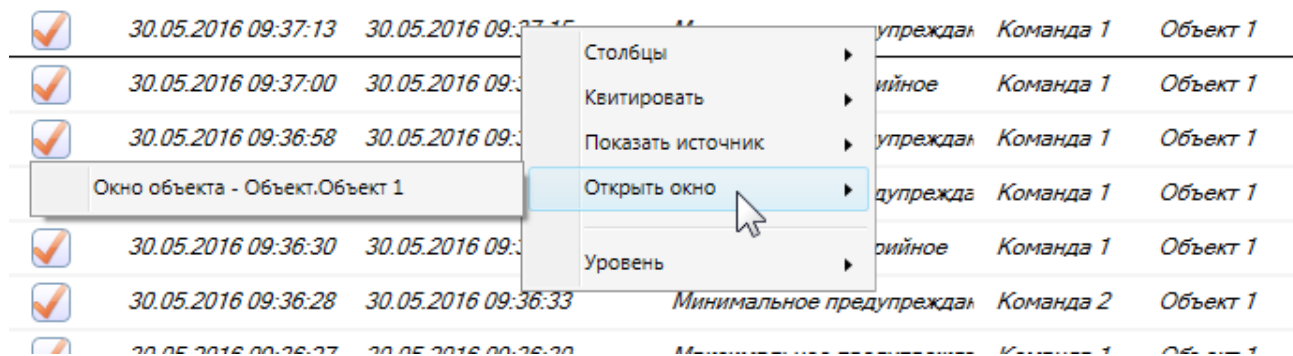


Рисунок 8-20. Меню перехода к окну из журнала

Подробно данный режим описан в разделе 6.5.3 «Продвинутые способы открытия окон».

8.2.6. Создание пользовательского журнала

Как было сказано ранее, по умолчанию у объектов присутствует три журнала: **Основной**, **Активные**, **Неквитированные**. При необходимости можно сделать собственный журнал, который будет присутствовать у всех объектов.

Для этого нужно выделить элемент **Система**, вкладка **Журналы**. При нажатии на кнопку **Добавить** создается новый журнал.

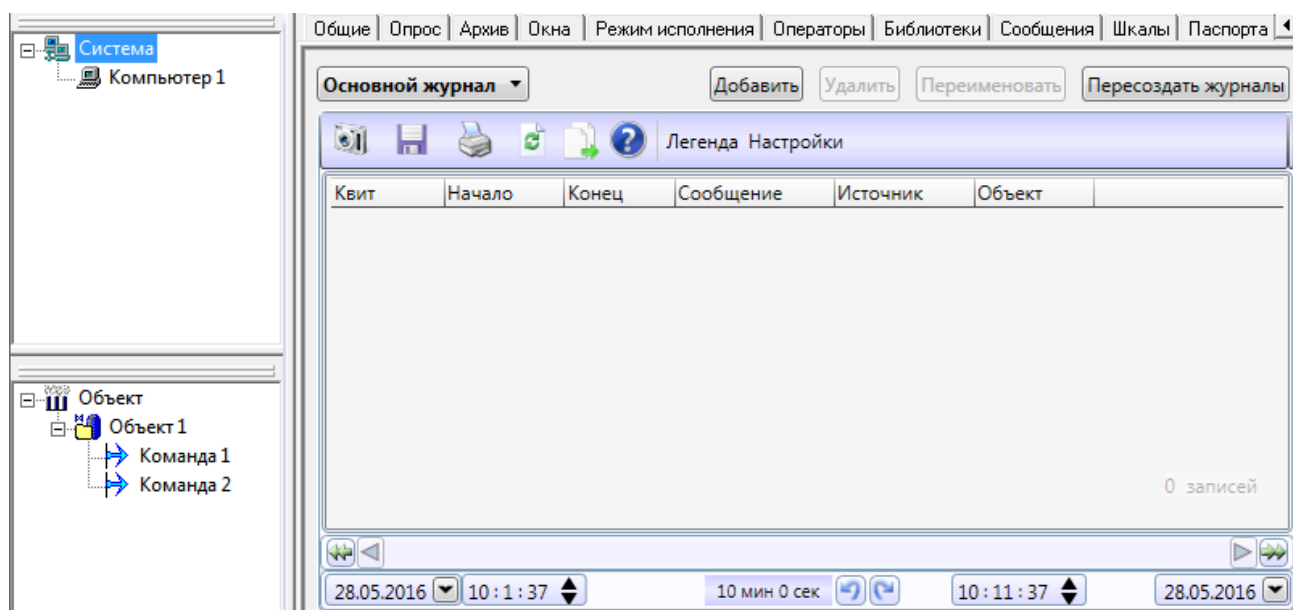


Рисунок 8-21. Вкладка «Журналы» элемента «Система»

Вы можете настроить фильтр этого журнала согласно вашей задаче. Например, сделаем так, чтобы журнал выводил только активные аварии. У добавленного журнала активируем панель настроек, включим вывод категории **Аварии** и в разделе **Флаги** установим **Только активные**.

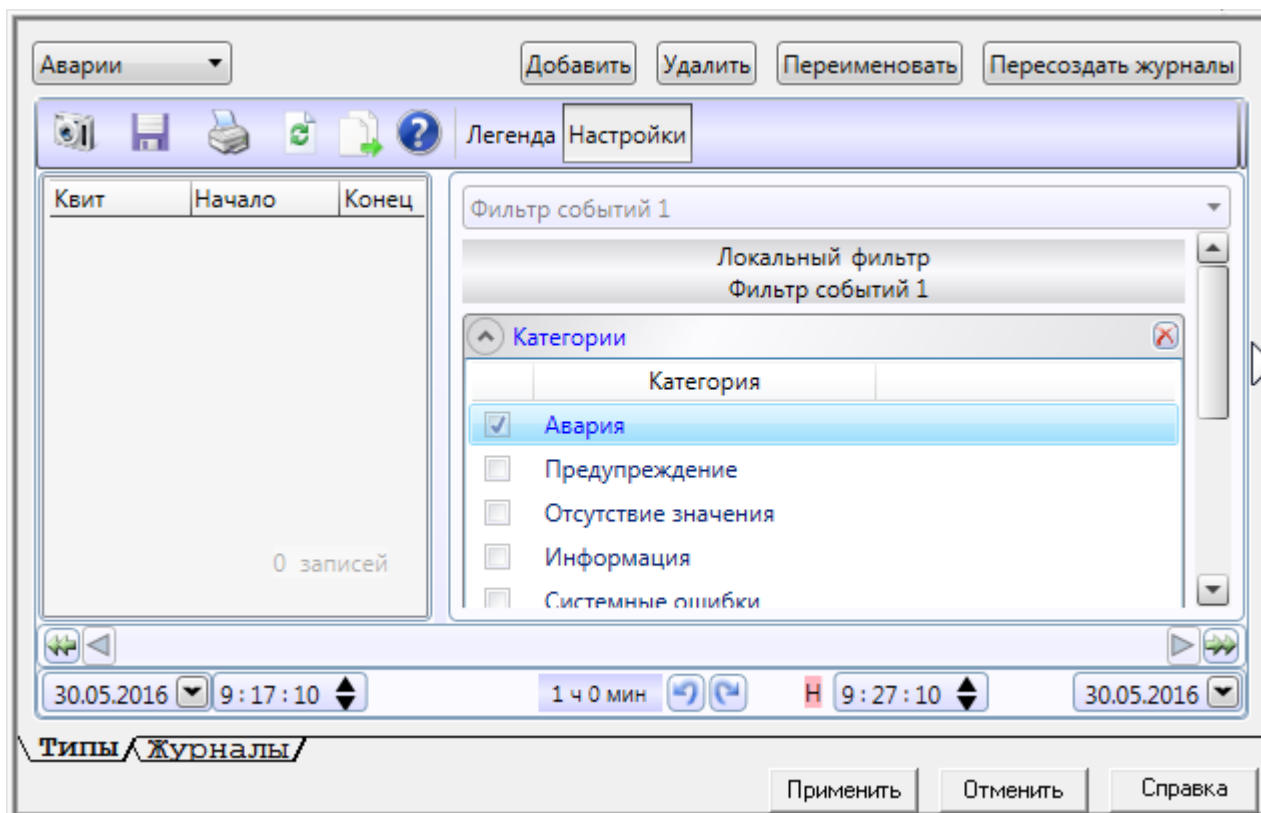


Рисунок 8-22. Настройка собственного журнала

Теперь данный журнал будет доступен для создания у всех объектов проекта.

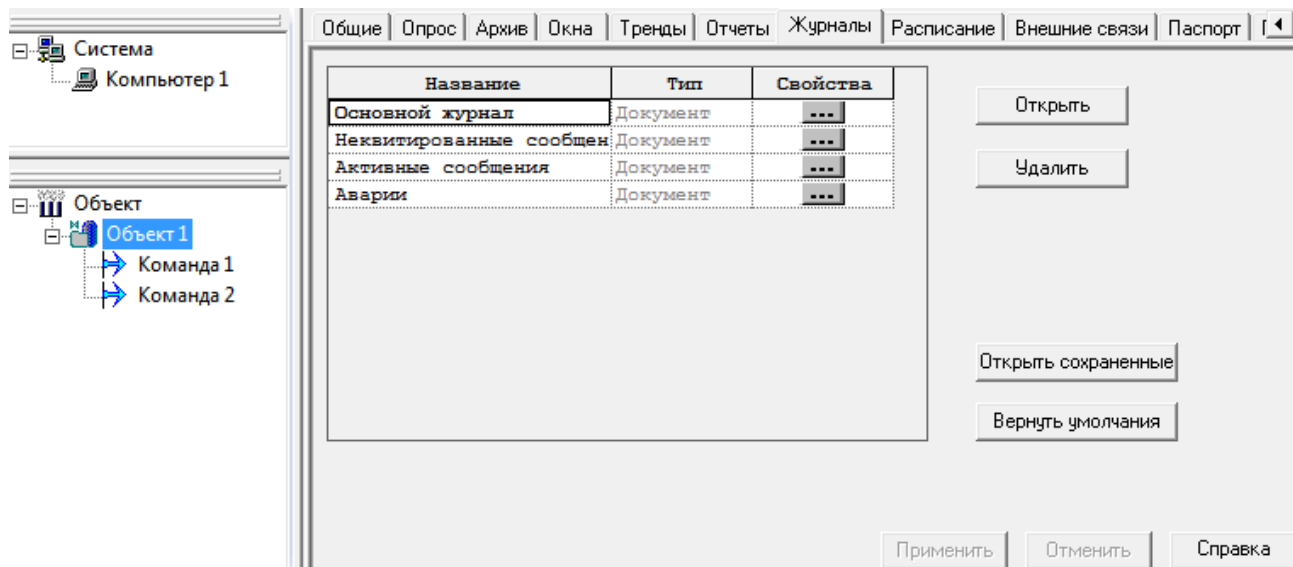


Рисунок 8-23. Вкладка «Журналы» после создания собственного журнала

8.2.7. Расширенный функционал

Рассмотренные функции – это лишь малая часть модуля журналов сообщений. В нем также есть возможности:

- Выполнять сортировку (в том числе множественную) по всем столбцам.
- Сохранять данные в графические форматы и HTML.
- Связывание журналов и трендов – несколько разных журналов связываются по времени, то есть изменение настроек интервала, положения автоматически приводит к изменению связанного с ними журнала. При этом можно связывать между собой тренды и журналы.
- Редактор уровней сложности – отдельный редактор, в котором можно создать собственный уровень.

Все перечисленные функции подробно разобраны в наших видеопримерах, которые содержатся на прилагаемом диске.

9. Редактор отчетов MasterReport

Зачастую помимо простого просмотра архивов требуется более глубокая работа с архивными данными – обработка данных и формирование отчета специальной формы.

Для решения данной задачи в комплект MasterSCADA входит встроенный редактор отчетов MasterReport. С его помощью можно формировать отчеты любой формы и сложности.

9.1.Создание отчета

Как и другие документы, отчет создается на уровне объекта, на одноименной вкладке. Для его создания нужно нажать на кнопку **Добавить отчет** и ввести название отчета в диалоговом окне. Открыть уже созданный отчет можно нажатием кнопки **Редактировать**.

У созданного отчета также можно задать параметры окна и свойства (размер, тип открытия и т. д.).

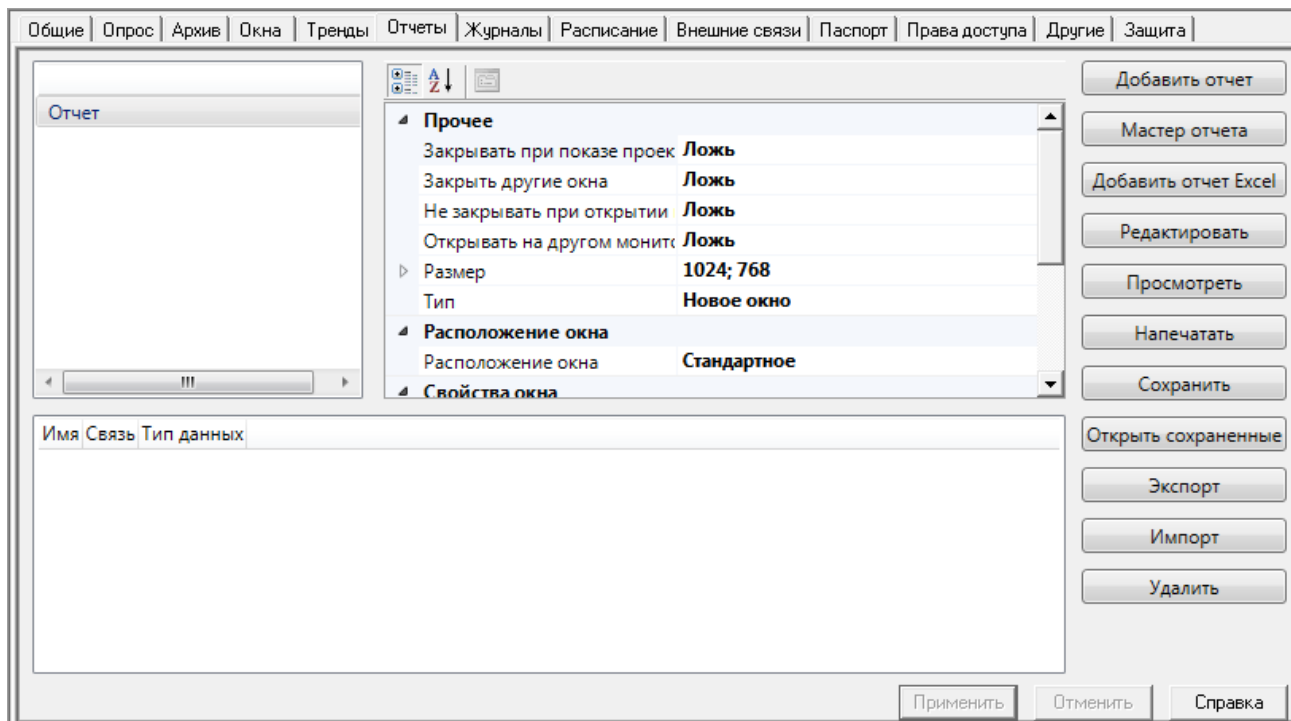


Рисунок 9-1. Вкладка для создания отчетов

9.2. Интерфейс редактора отчетов

После создания отчета загружается среда разработки. Интерфейс разделен на следующие секции.

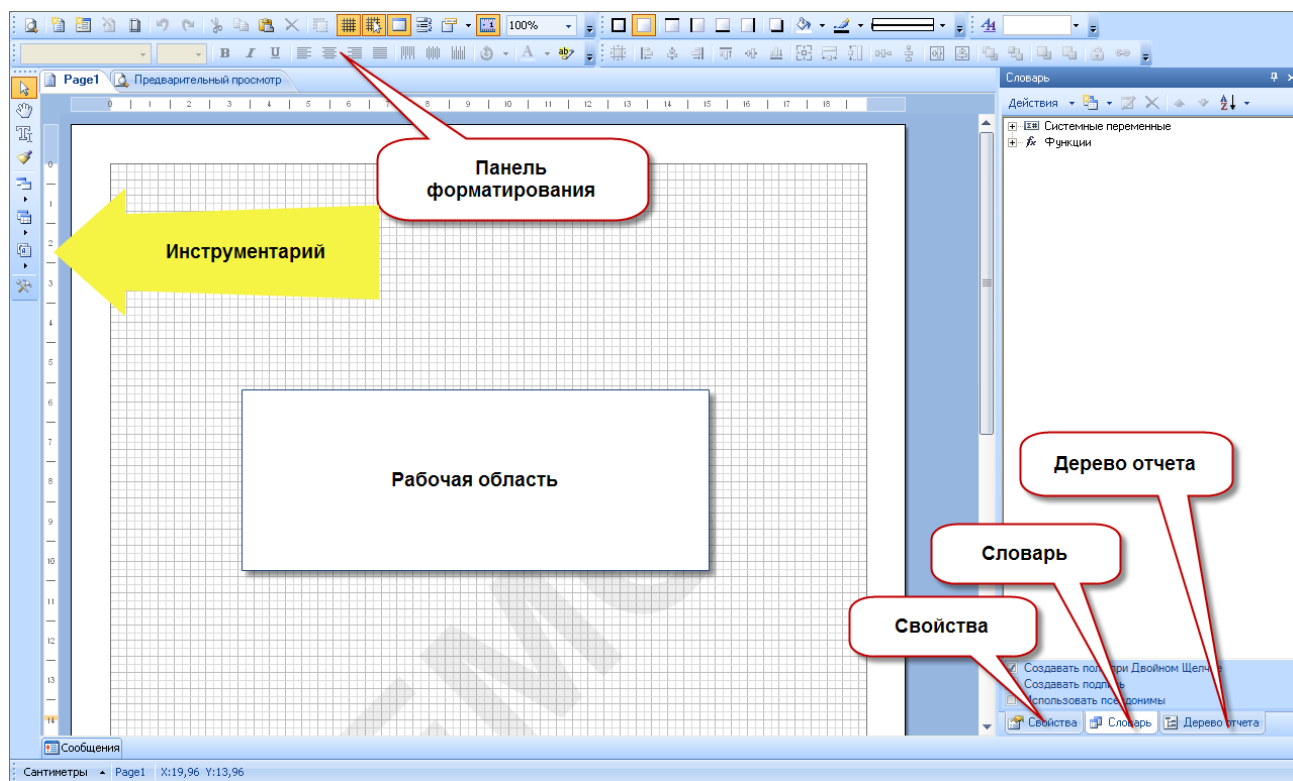


Рисунок 9-2. Интерфейс редактора отчетов

В левой части редактора отчета находится панель **Инструментарий**, в которой расположены основные инструменты: кнопки добавления бэндов, кросс-бэндов, компонентов. Инструментарий можно настраивать – выносить на верхний уровень те или иные кнопки.

В правой части редактора отчетов находится окно с вкладками **Свойства**, **Словарь**, **Дерево отчета**. На вкладке **Свойства** отображаются параметры бэндов, переменных, источников данных. На вкладке **Словарь** находятся источники данных, пользовательские переменные, системные переменные. На вкладке **Дерево отчета** можно просмотреть и изменить структуру отчета.

9.3. Добавление бэндов

Основой логики отчетов является бэнд. Бэнд (или секция) – это специальный компонент, располагаемый на странице. Бэнд невидим после построения отчета, и основная его задача – определение логики отображения расположенных на нем элементов.

Добавление бэндов производится после нажатия на кнопку **Секции**. Появляется окно, в котором можно выбрать нужный бэнд.

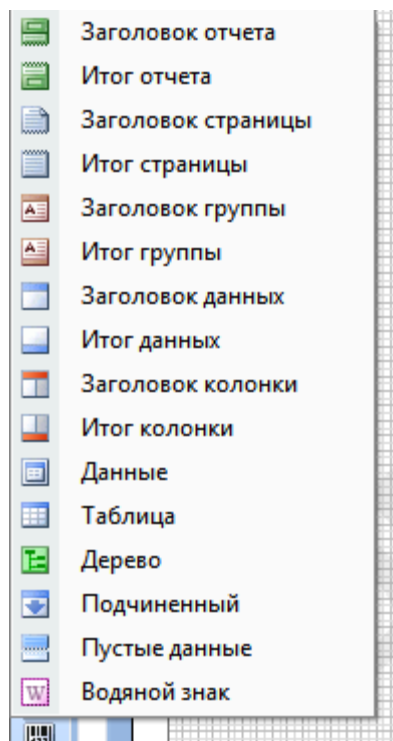


Рисунок 9-3. Меню с бэндами

Каждый бэнд выводится на страницу в определенный момент времени. Например, **Заголовок отчета** – один раз в самом начале отчета, **Заголовок страницы** – в верхней части каждой страницы, **Данные** – выводят последовательно набор строк данных из источника, и т. д. Подробнее каждый бэнд описан в справочной системе редактора отчетов.

Для примера добавим на страницу бэнд **Заголовок отчета** – выберем его в списке и разместим на странице.

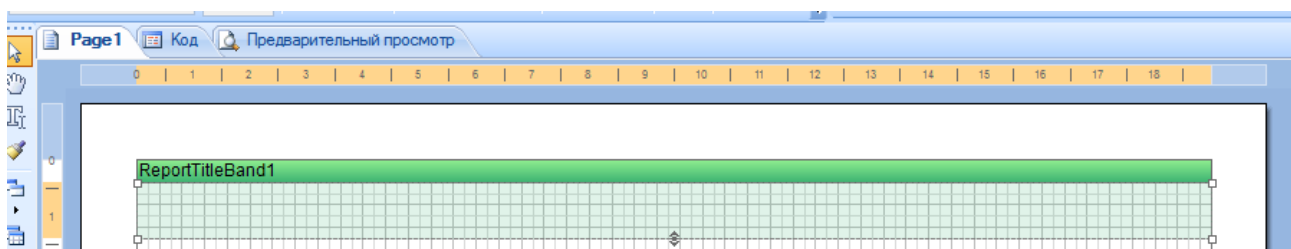


Рисунок 9-4. Бэнд «Заголовок отчета» на странице

Теперь на этом бэнде можно размещать элементы. Например, добавим подпись, которая и будет выводиться в начале отчета. Выберем компонент **Текст** и разместим его на бэнде (или просто щелкнув на бэнде, или «нарисовав» наш компонент).

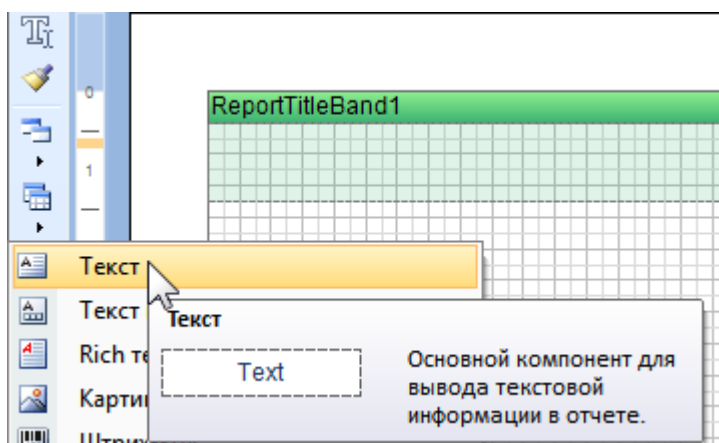


Рисунок 9-5. Добавление элемента «Текст»

Сразу после добавления появится окно редактирования текста – в нем можно написать комментарий. Напишем текст «Наш первый отчет» и нажмем **ОК**.

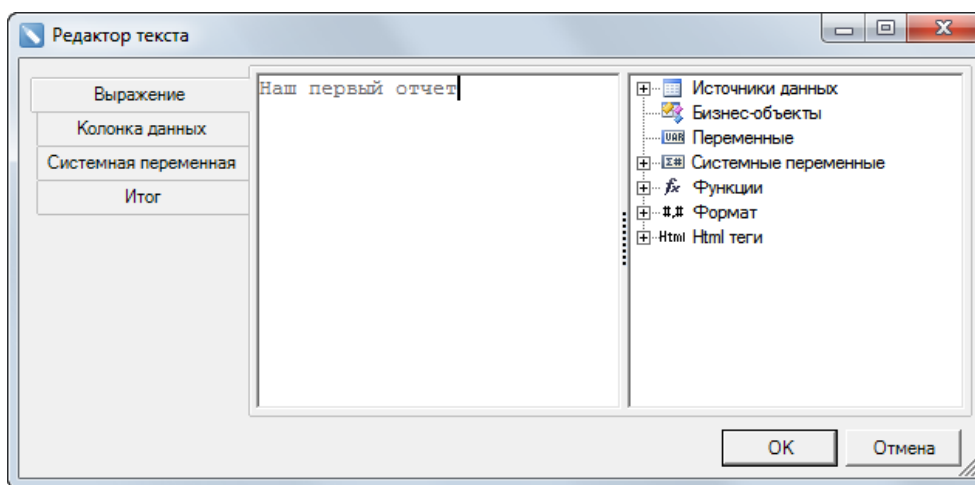


Рисунок 9-6. Редактор текста

Текст добавился в поле. Сделаем шрифт покрупнее и разместим текст по центру, используя стандартные кнопки на панели форматирования.

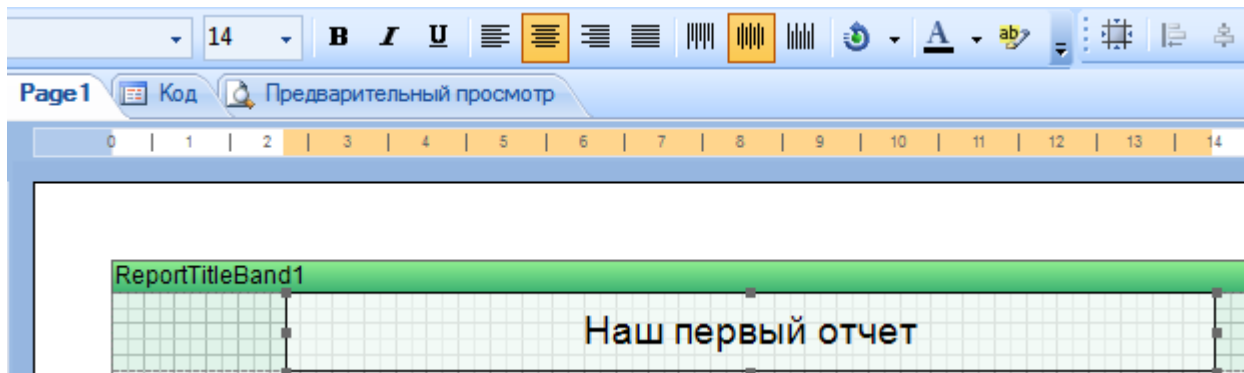


Рисунок 9-7. Бэнд с текстом

Запустим построение отчета. Для этого нажмите на вкладку **Предварительный просмотр**. Будет построен отчет из одной страницы, содержащей текст «Наш первый отчет» в самом верху.

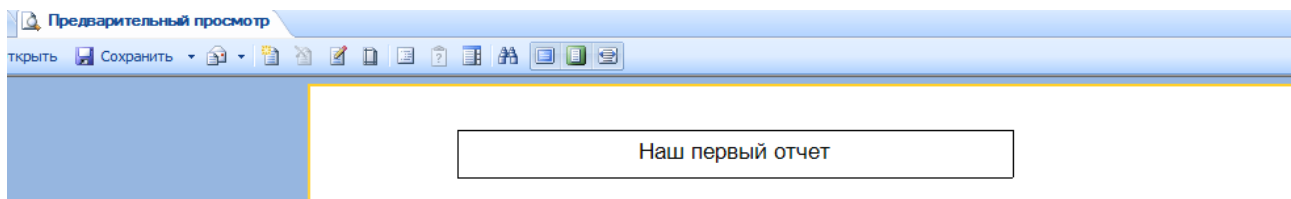


Рисунок 9-8. Вкладка предварительного просмотра

Теперь рассмотрим добавление данных из дерева объектов в отчет.

9.4. Создание источника данных и добавление колонок

Вытащить переменные из дерева объектов прямо на страницу отчета нельзя – необходимо сначала добавить их в словарь, а уже затем, из словаря, добавлять на страницу. При этом в редакторе отчетов есть возможность обращаться к архиву переменной или к его текущему значению.

Для того чтобы обратиться к архиву переменной, нужно добавить переменные из дерева объектов в источник данных **Архив MasterSCADA**. На основе архива добавленных в источник переменных по различным алгоритмам будут сформированы строки данных, которые и можно вывести на страницу.

Создадим данный источник. Для этого вызовем на закладке **Словарь** контекстное меню **Добавить источник данных – Архив MasterSCADA** (Рисунок 9-9).

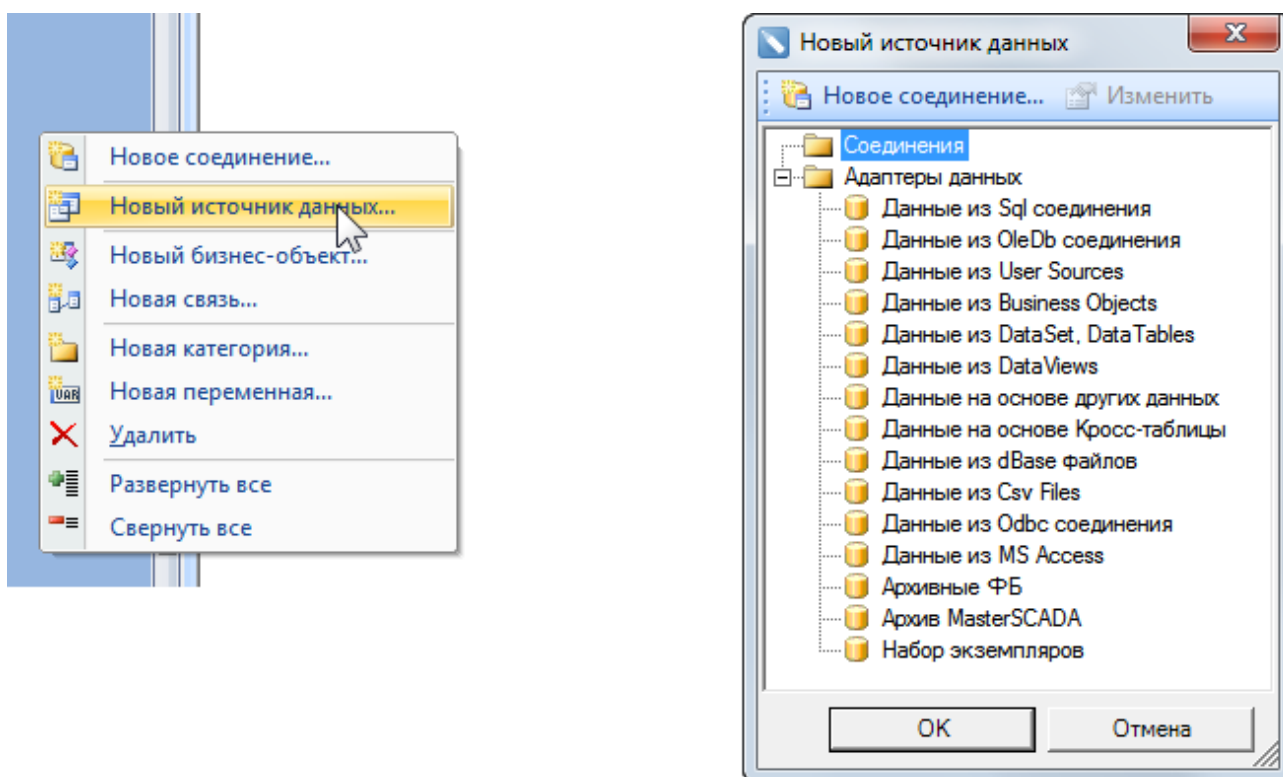


Рисунок 9-9. Создание источника данных

В словаре будет создан источник данных. Сейчас он содержит одну колонку – **Время** и три параметра в отдельной группе: **Начало**, **Конец** и **Интервал**. Колонка **Время** будет отображать время выводимой строки. Параметры **Начало** и **Конец** ограничивают диапазон выборки из архива. Параметр **Интервал** при периодическом способе формирования отчета задает временной шаг, с которым выводятся строки в отчет.

Теперь в источник данных можно перетащить переменные. В дереве объектов у нас уже добавлены **Команда1–Команда3**, которые содержат архив, а также команды **Начало** и **Конец**, определяющие интервал, за который нужно сформировать отчет.

Перетащим данные переменные в источник данных. Переменные можно перетаскивать по одной или, выделив их, перетащить группой.

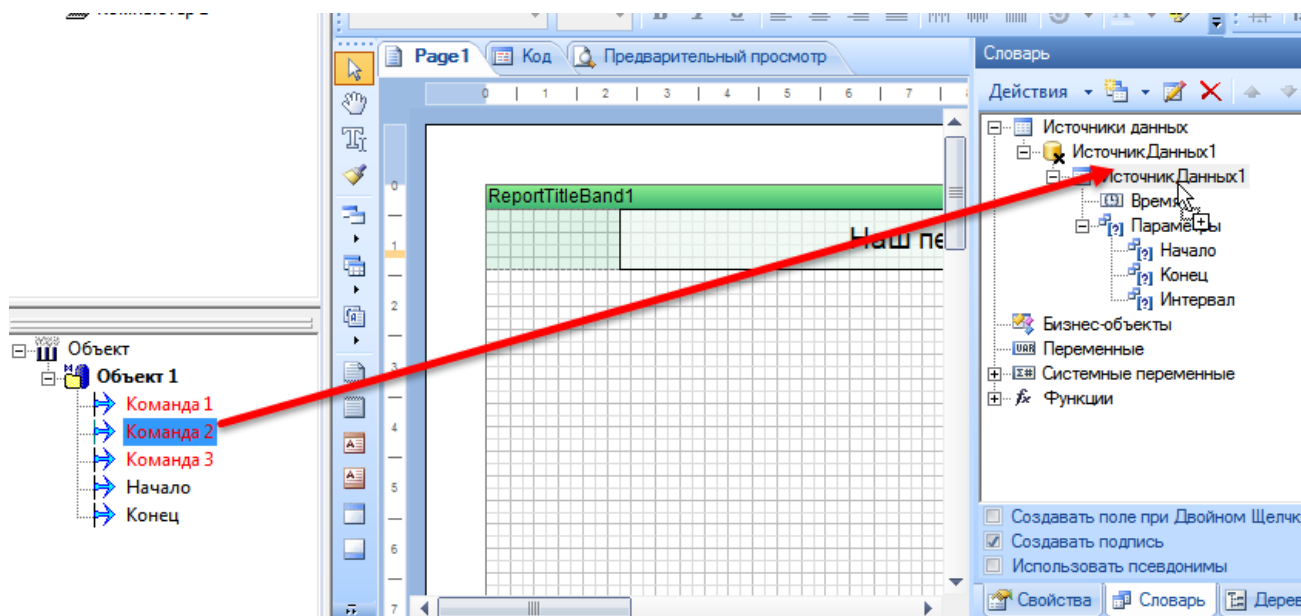


Рисунок 9-10. Добавление переменных в источник

Теперь свяжем команды *Начало* и *Конец* с одноименными параметрами.

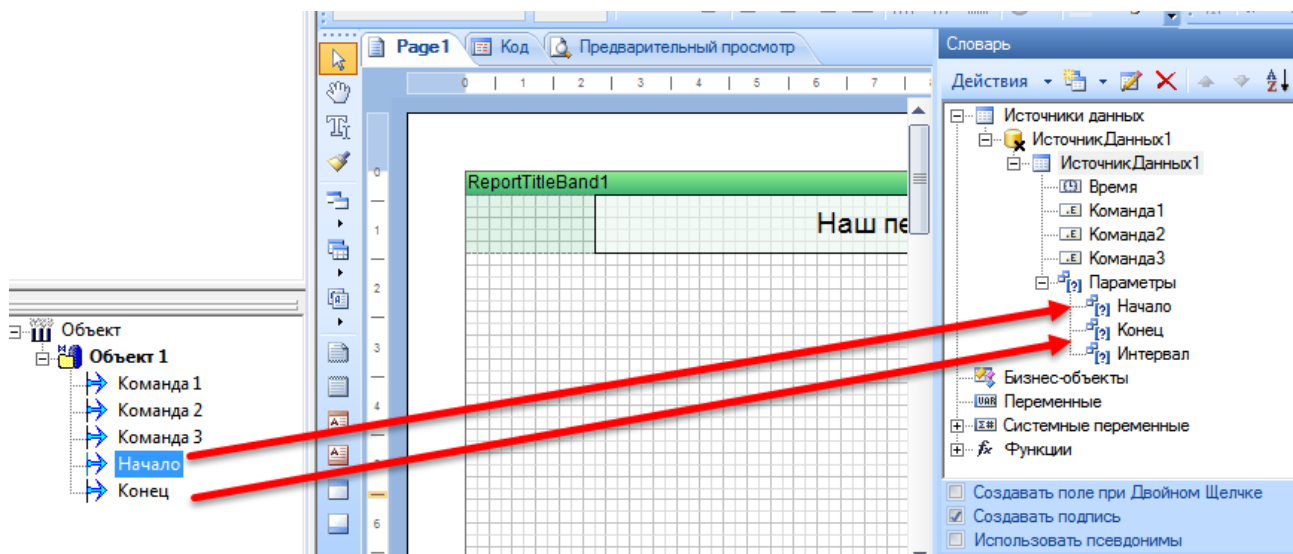


Рисунок 9-11. Связывание переменных «Начало» и «Конец»

Теперь, когда источник данных полностью настроен, его можно добавить на страницу. Для этого можно просто вытащить источник данных на страницу – это приведет к созданию бэнда Данные, привязанного к источнику.

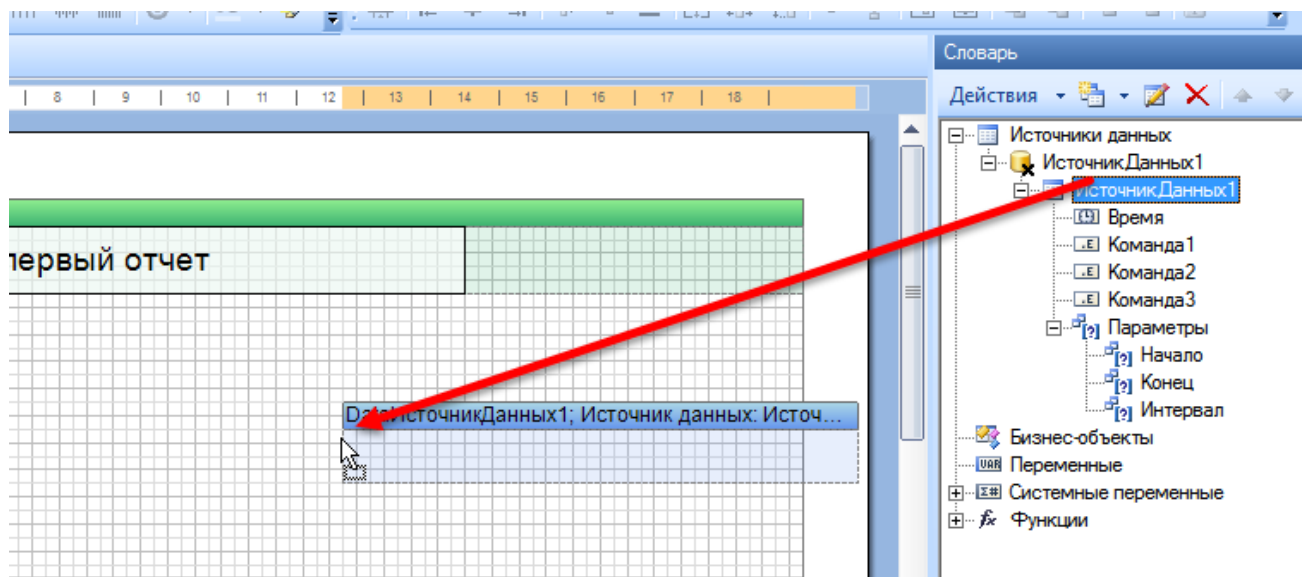


Рисунок 9-12. Создание бэнда «Данные»

После вытаскивания источника на страницу появляется окно, в котором предлагается выбрать, какие колонки требуется добавить и в каком порядке; также можно отметить создание бэндов **Заголовок данных** и **Итог данных** (эти бэнды будут, соответственно, выведены перед бэндом **Данные** и под ним).

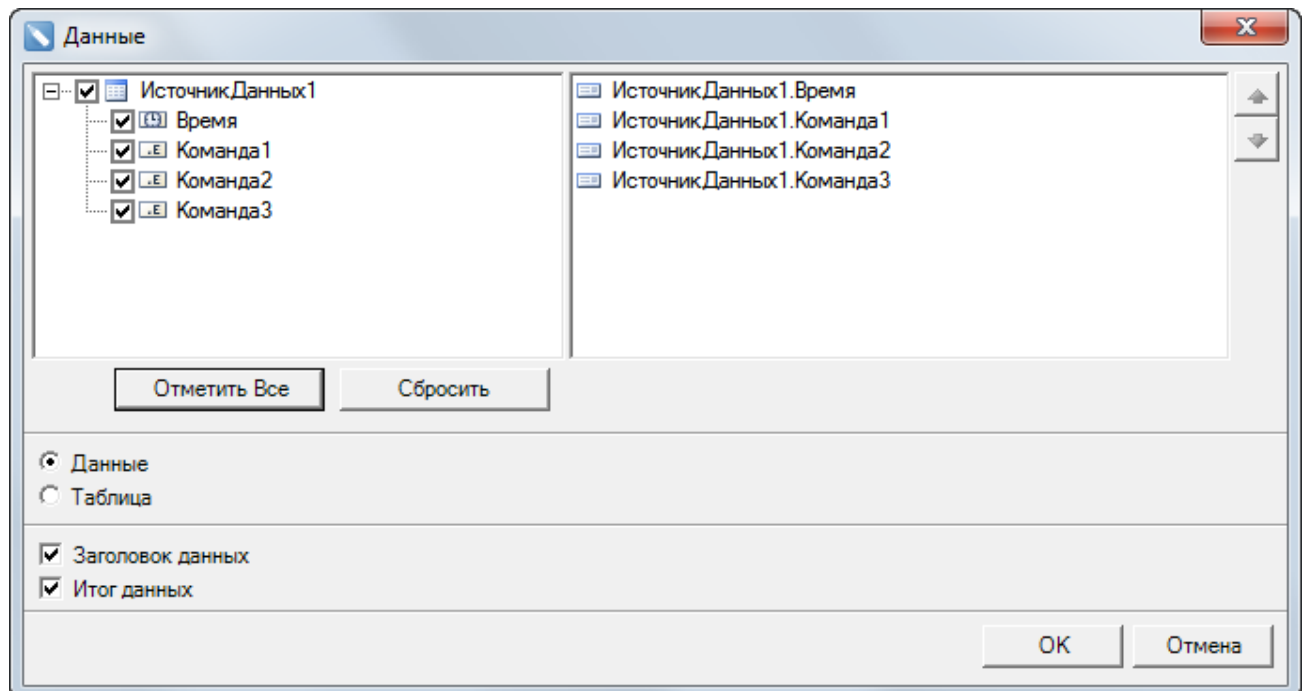


Рисунок 9-13. Настройка бэнда «Данные»

Запустим построение отчета, нажав на кнопку предварительного просмотра. Будет сформировано несколько страниц со строками данных, содержащими время и значения переменных.

The screenshot shows a window titled 'Предварительный просмотр' (Preview) with a toolbar containing icons for 'Открыть' (Open), 'Сохранить' (Save), and other standard functions. The main content area displays a report titled 'Наш первый отчет' (Our first report) containing a table with the following data:

Время	Команда1	Команда2	Команда3
04.06.2016 13:51:06	1,09333333333333	53,4321069774727	0,125125888851588
04.06.2016 13:51:07	2,78333333333333	58,6995963449409	56,358531449324
04.06.2016 13:51:08	4,47333333333333	63,869086152933	19,3304239020966
04.06.2016 13:51:09	6,16333333333333	68,8823429267205	80,8740501113926
04.06.2016 13:51:10	7,85333333333333	73,6828931316828	58,500930814639

Рисунок 9-14. Предварительный просмотр отчета

Строки в источнике данных могут формироваться тремя способами: **По изменению**, **Периодический** и **Итоговый**. Выбор режима производится на закладке **Свойства** источника данных.

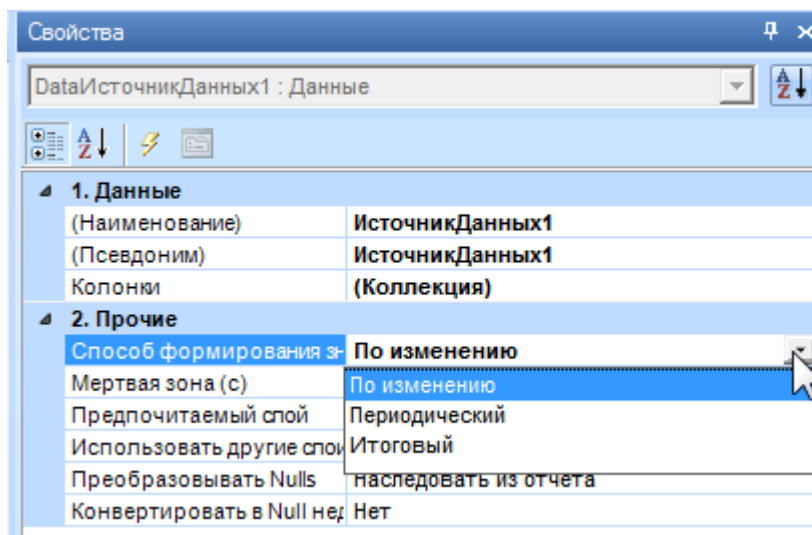


Рисунок 9-15. Выбор способа формирования

В режиме **По изменению** новая строка в источнике создается при любом изменении любой переменной источника (кроме тех, у которых выключена настройка **Основной**).

1. Данные	
(Наименование)	Команда1
Связь	Объект.Объект 1.Команда 1
Форматировать значение	Нет
Тип	double (Nullable)
2. Прочие	
Основной	True
3. Обработка данных	
Атрибут данных	Значение
Обработка	Последнее значение
Граница интервала	Включать
4. Обработка плохих признаков	
Обработка признака Stop	Нет; Good
Обработка признака Bad	Нет; Uncertain
Обработка признака Unc	Нет; Uncertain

Рисунок 9-16. Настройки переменной в режиме «По изменению»

В режиме *Периодический* строки формируются с определенным временным шагом (1 секунда, 5 секунд, 10 минут и т. д.). Временной шаг вывода определяется значением параметра *Интервал*.

1. Данные	
(Наименование)	Интервал
(Функция)	False
Тип	double (Nullable)
Значение	<input type="text" value="1"/>

Рисунок 9-17. Настройки переменной в режиме «Периодический»

Режим *Итоговый* является разновидностью периодического отчета – при данном способе формируется одна строка за весь интервал времени.

9.5.Способы обработки колонок

При периодическом способе формирования на одну строчку выводимых в отчет данных может приходиться несколько архивных значений переменной за выводимый интервал времени. Какое значение вывести в строку? Для решения этой проблемы существует настройка *Способ обработки* у колонки источника.

Выберем способ формирования отчетов – *Периодический* и зададим *Интервал* – 5 секунд. Выделим колонку и перейдем на вкладку *Свойства*. Раскроем список обработок.

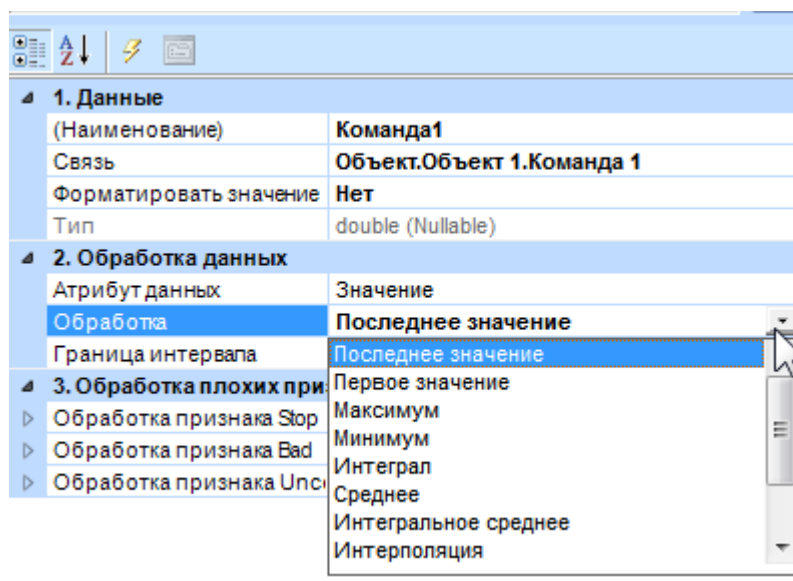


Рисунок 9-18. Выбор способа обработки

Последнее значение – при данном способе выводится последнее значение на интервале.

Первое значение – первое значение на интервале.

Максимум – максимальное значение за интервал времени.

Минимум – минимальное значение за интервал времени.

Разница – разница между первым и последним значением на интервале. Данный способ обработки применяется, когда нужно вычислить расход за интервал по архиву нарастающего итога (например, расход электроэнергии за час по его накопленному нарастающему значению).

Сумма – сумма первого и последнего значения на интервал времени.

Интеграл – вычисляет интеграл, то есть площадь фигуры, ограниченной сверху архивными значениями. Данный способ обработки применяется, если необходимо найти расход по его действующему значению (например, расход газа за час по действующему значению расходомера).

Интегральное среднее – вычисляет интеграл, а затем делит результат на временной промежуток.

Среднее – среднее арифметическое, то есть сумма всех значений делится на их количество.

Количество значений – количество значений на интервале.

Подробнее каждый способ обработки описан в справке.

Используя различные способы обработки, можно точно и гибко подобрать обработку архива данных. При этом допускается добавлять одну и ту же переменную в источник данных и задавать им различные способы обработки.

Также у колонки есть настройка **Атрибут данных**, которая позволяет выводить в колонку не только значение переменной, но и метку времени и признак качества.

Вывод метки времени может быть необходим, например, если требуется максимальное значение за интервал времени и время максимума. В этом случае одна и та же переменная дерева объектов добавляется в отчет дважды, у обеих ставится **Способ обработки – Максимум**, но у одной будет атрибут **Значение**, а у другой – **Время**.

Вывод признака качества позволяет выводить в отчет нарушения связи, отказы датчиков и т. д. При этом можно настроить выделение ячеек цветом в зависимости от признака качества. Кроме того, для обработки недостоверных значений есть специальная группа настроек.

9.6. Обработка плохих признаков качества

Как быть, если на интервале времени присутствуют недостоверные значения? Например, мы выводим максимальное значение, и на интервале максимальное значение имеет недостоверный признак качества. Какое значение вывести в таком случае: максимальное или максимальное достоверное? Для решения данной проблемы у каждой колонки есть специальная группа настроек – **Обработка плохих признаков**.

▶ 1. Данные	
▶ 2. Обработка данных	
▲ 3. Обработка плохих признаков	
▲ Обработка признака Stop	Нет; Good
Игнорировать значения	Нет
Настройка признака результата	Good
▲ Обработка признака Bad	Нет; Uncertain
Игнорировать значения	Нет
Настройка признака результата	Uncertain
▲ Обработка признака Uncertain	Нет; Uncertain
Игнорировать значения	Нет
Настройка признака результата	Uncertain

Рисунок 9-19. Настройки для обработки плохих признаков качества

Группа содержит три вложенных группы для каждой категории недостоверных признаков: **Stop** (признак останова режима исполнения), **Uncertain** (не определенные признаки качества) и **Bad** (ошибки, отказы устройств и датчиков и т. д.). Если на интервале обработки будет обнаружено значение, влияющее на результат, с данным признаком качества (**Stop**, **Bad**, **Uncertain** – в зависимости от группы настроек), то в результате колонка будет иметь признак, определенный данной настройкой.

Алгоритм определения признака результата различается в зависимости от способа обработки параметра.

Если используются способы обработки **Первое значение**, **Последнее значение**, **Максимум**, **Минимум**, то признак качества колонки выставляется на основании результата. Например, если способ обработки – **Последнее значение** и последнее значение на интервале имело признак, относящийся к **Bad** (**Ошибка датчика**, **Ошибка устройства** и т. д.), то признак результата определится настройкой группы настроек **Bad**. Если на интервале есть значение с признаком **Bad**, но оно не является последним, то на результат оно не влияет.

Если используются способы обработки **Среднее**, **Интеграл**, **Интегральное среднее**, то признак качества колонки выставляется на основании всех значений на интервале – поскольку все значения влияют на результат. Например, если способ обработки **Среднее** и на интервале было обнаружено значение с признаком качества **Bad**, то признак результата определится настройкой группы настроек **Bad**.

Каждая группа содержит настройки **Игнорировать значения** и **Настройка признака результата**. Настройка **Игнорировать значения** определяет, игнорировать ли недостоверные значения данной категории. По умолчанию значения не игнорируются, то есть если используется обработка **Максимум**, то будет выведено максимальное из достоверных значений. Настройка признака результата определяет, какой признак получит колонка в результате: **Good** (норма), **Bad** (плохой), **Uncertain** (неопределенный) или **Наследуется** (наследуется признак результата обработки). То есть если в разделе **Обработка признаков Bad** и настройка признака результата будет задана **Uncertain**, то при обнаружении на интервале значения с признаком качества **Bad** результату колонки будет установлен признак **Uncertain**.

Обработки *Интеграл* и *Интегральное среднее* имеют большее количество настроек, позволяющее задать, как поступать с недостоверными областями при расчете интеграла (исключать их, считать по последнему достоверному и т. д.). Подробнее все комбинации настроек с описанием результата приведены в справочной системе.

9.7.Дополнительные материалы и шаблонные отчеты

На этом завершим обзор редактора отчетов **MasterReport**. На прилагаемом диске вы можете найти большое количество дополнительных материалов – это текстовая документация с различными примерами создания отчетов и видеопримеры с описанием интерфейса и нескольких несложных отчетов.

Также наша компания выпускает шаблонные отчеты для коммерческого и технического учета электроэнергии, а также систем теплоучета. Шаблонные отчеты также находятся на прилагаемом диске.

10. Сетевые проекты MasterSCADA

Зачастую в проекте автоматизации требуется использовать несколько компьютеров, которые могут выполнять различные задачи: работать как дополнительный сервер ввода (например, отдельный сервер для каждого цеха), как резервный (для создания высоконадежных систем) или просто как клиентское место – для отображения данных и управления процессом оператором. При этом сетевые конфигурации могут быть самыми разными, и все их несложно реализовать в MasterSCADA.

Далее будут рассмотрены различные варианты сетевых архитектур и способы их построения в MasterSCADA.

10.1. Общий принцип создания сетевых проектов

Сетевой проект в MasterSCADA создается в три шага:

1. В дерево системы добавляется еще один компьютер. Таким образом, должно получиться как минимум 2 компьютера.

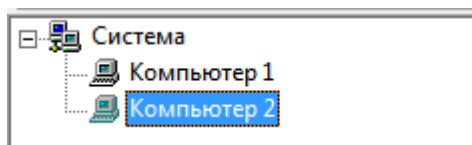


Рисунок 10-1. Создание второго компьютера

2. У компьютеров задаются IP-адреса, соответствующие реальным IP-адресам компьютеров (по IP-адресу MasterSCADA определяет роль данного компьютера в проекте). Это делается у компьютера на закладке **Система – Режим исполнения – Связь**.

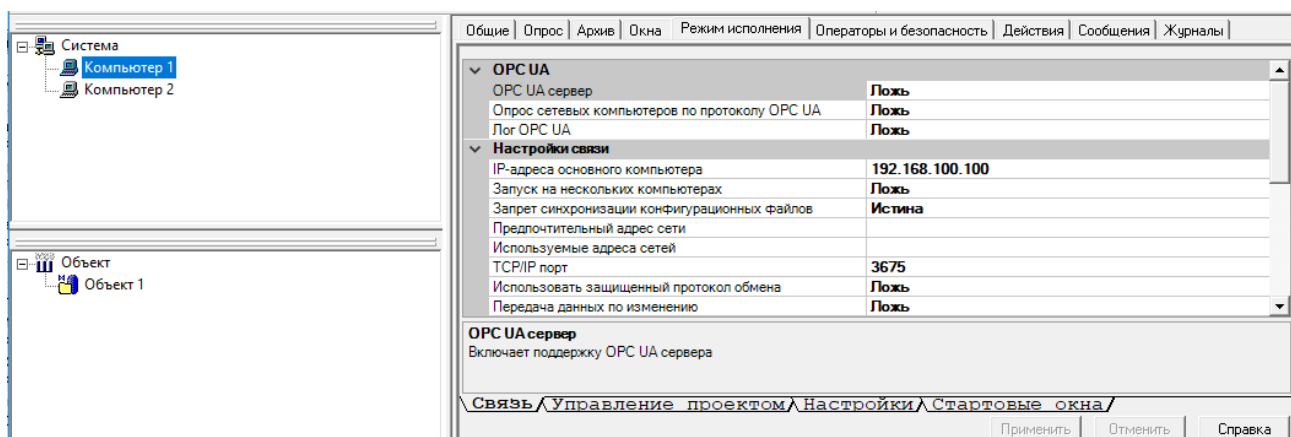


Рисунок 10-2. Настройки связи «Компьютера 1»

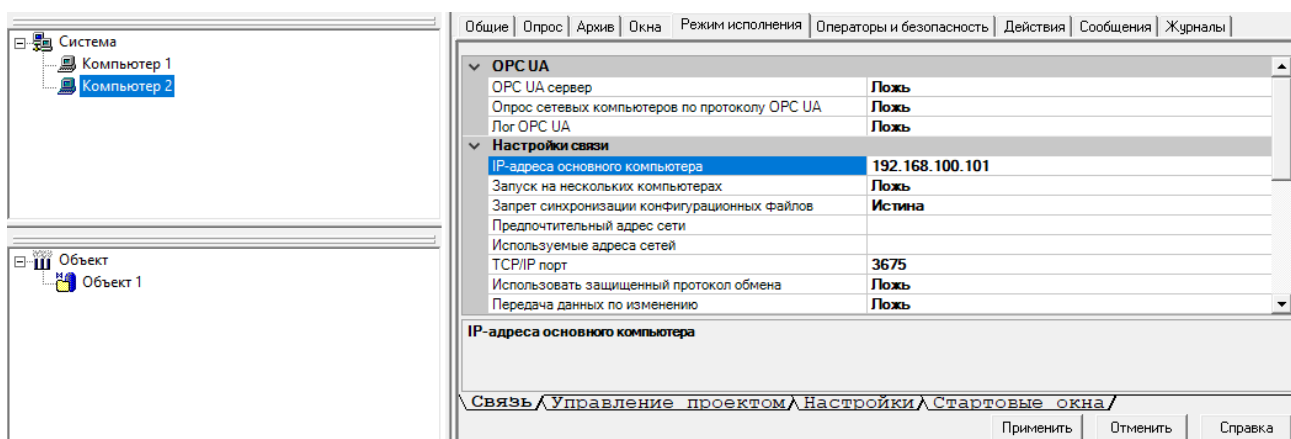


Рисунок 10-3. Настройки связи «Компьютера 2»

3. Проект переносится на все компьютеры. Для этого выполняются экспорт проекта и его копирование на другие компьютеры.

На этом настройка сетевого проекта завершена. При запуске в режиме исполнения система определит роль каждого компьютера и начнет исполнять назначенные каждому компьютеру объекты.

10.2. Синхронизация проекта

Чтобы не выполнять ручное копирование проекта на каждый компьютер сетевого проекта (а их могут быть десятки), есть возможность настройки автоматической синхронизации. Ее суть сводится к следующему: один из компьютеров назначается сервером проекта, то есть проект на данном компьютере будет считаться эталоном. Затем по команде

разработчика (при запуске в режиме исполнения или просто при обнаружении изменений проекта) начинается обновление проекта на остальных компьютерах системы.

Чтобы настроить автоматическую синхронизацию, выделите элемент **Система** и перейдите на вкладку **Режим исполнения – Управление проектом**. Выберите нужный вам вариант синхронизации.

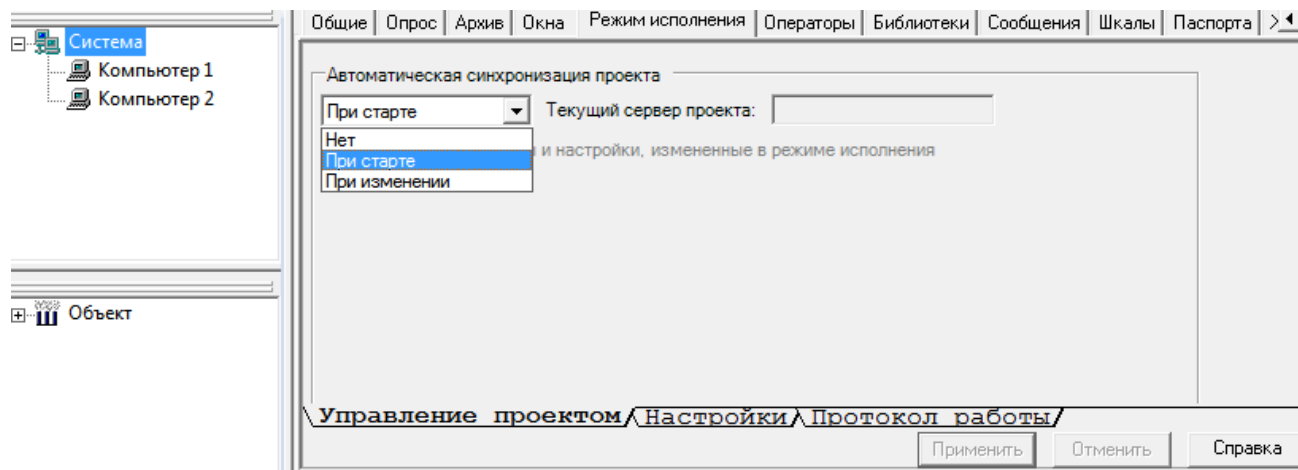


Рисунок 10-4. Выбор режима синхронизации

Теперь выберем сервер проекта. Например, в нашем случае это будет **Компьютер 1**. Выделим его в дереве системы и также на вкладке **Режим исполнения – Управление проектом**, установим галочку **Данный компьютер является сервером проекта**.

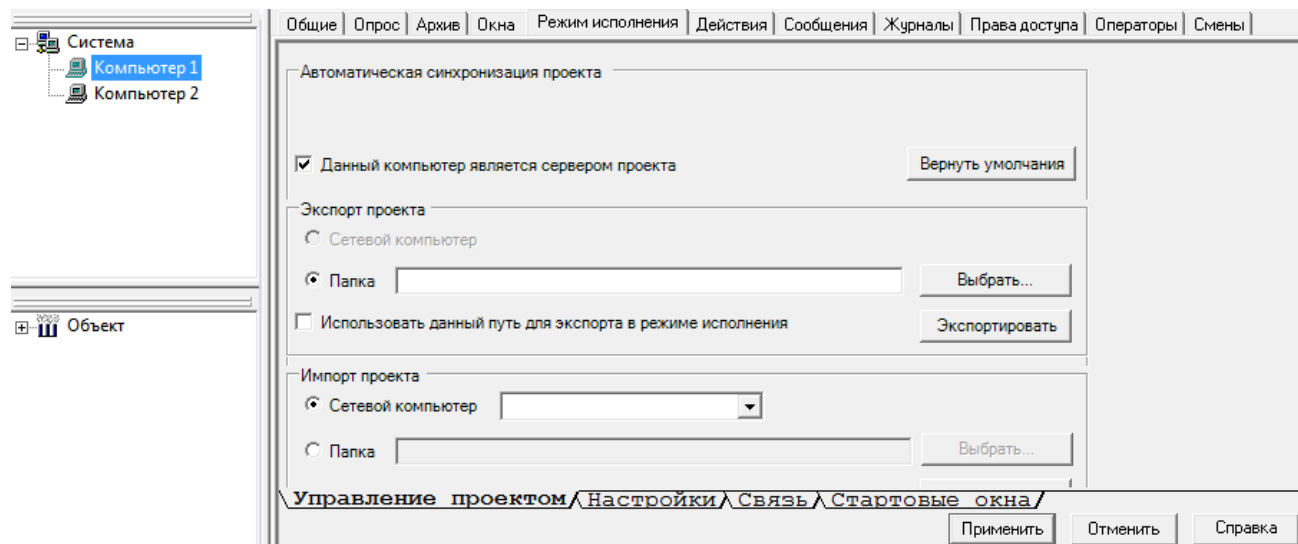


Рисунок 10-5. Назначение сервера проекта

Если выделить **Компьютер 2**, то в настройках импорта этого компьютера будет установлен **Компьютер 1**.

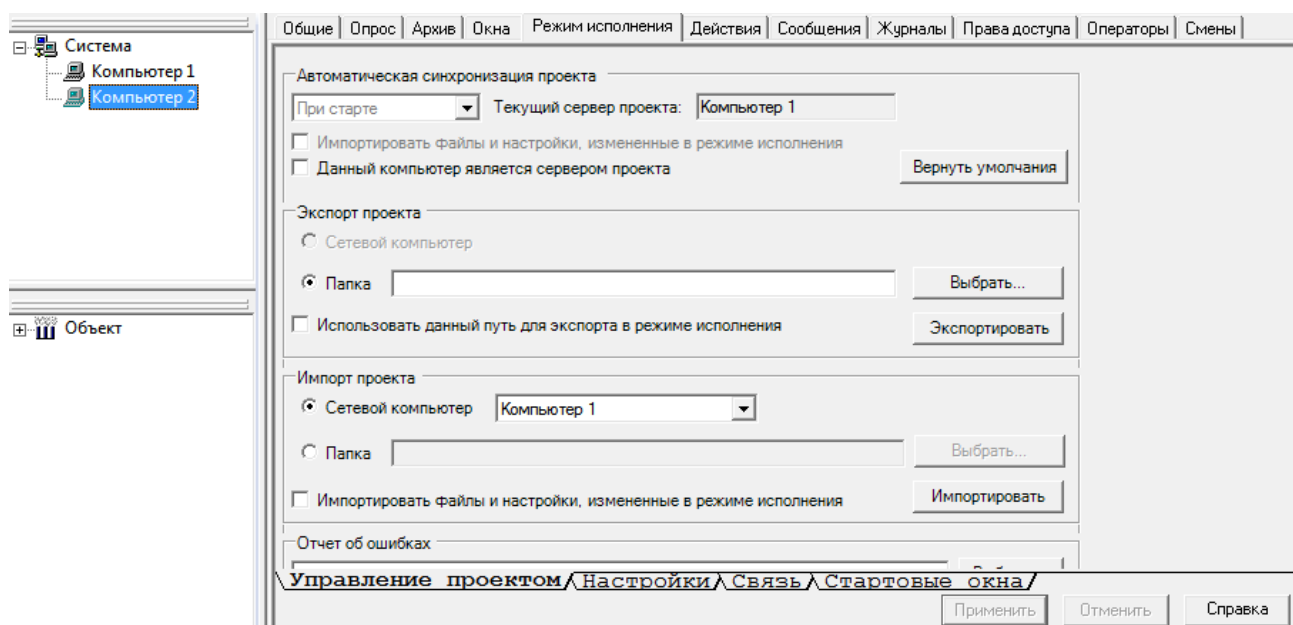


Рисунок 10-6. Выбор сервера проекта для импорта

Если на **Компьютере 1** запущен режим исполнения, то импорт проекта можно запустить прямо сейчас, нажав на кнопку **Импортировать**. Аналогично можно сделать и в режиме исполнения: данная вкладка в нем также доступна.

10.3. Сетевые архитектуры

Теперь, зная общие принципы создания сетевых проектов, можно рассмотреть варианты сетевых архитектур.

Все варианты архитектур мы будем рассматривать на примере из главы «Создание простейшего проекта».

10.3.1. Клиент-серверная архитектура и сервер-серверная архитектура

В клиент-серверной архитектуре один из компьютеров ведет опрос оборудования (через OPC-сервер или по встроенному протоколу в случае контроллера с **MasterPLC**), а второй предоставляет доступ к данным (мнемосхемам, трендам, журналам).

Типичный вариант подобной системы – компьютер оператора и компьютер мастера (*Рисунок 10-7*).

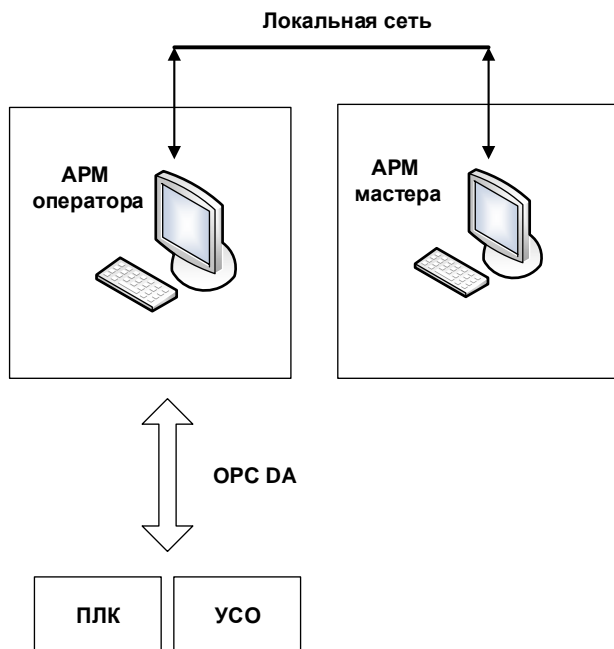


Рисунок 10-7. Клиент-серверная архитектура с одним сервером и одним клиентом

Чтобы создать такой проект, нужно добавить в систему еще один компьютер. Для удобства можно задать имена, более приближенные к реальным.

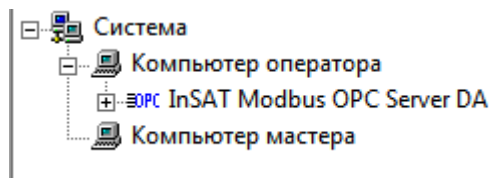


Рисунок 10-8. Клиент-серверная архитектура в дереве системы

В компьютер оператора добавлен OPC-сервер, который будет вести опрос оборудования. На этом же компьютере будет исполняться объект (при разработке проекта мы назначили исполнение на нем). С компьютера мастера можно зайти на мнемосхему, посмотреть тренд, задать верхнюю и нижнюю границу. При этом разрешения действий, выполняемых по сети (на другом компьютере), настраиваются в правах доступа отдельно: разрешение выполнения действия на объекте чужого компьютера определяется галочкой **Глоб.**

Сервер-серверная архитектура отличается от клиент-серверной только тем, что OPC-серверы находятся в обоих компьютерах.

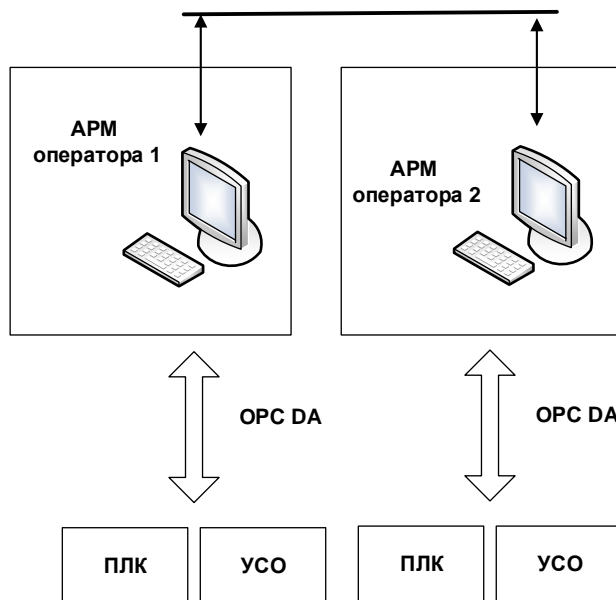


Рисунок 10-9. Клиент-серверная архитектура с двумя серверами

При этом могут быть и вариации такого проекта: к паре серверов может быть добавлен клиент (или несколько клиентов). Независимо от этого архитектура сетевой части строится одинаково – добавляется нужное количество компьютеров (с OPC-серверами или без них) и задаются их IP-адреса.

10.3.2. Система с архивным сервером

В разделе, посвященном системе архивации, в том числе описывалась архивация в базу данных. Такой способ архивации за счет использования слоев дает существенный прирост скорости, что особенно заметно как раз в сетевых проектах.

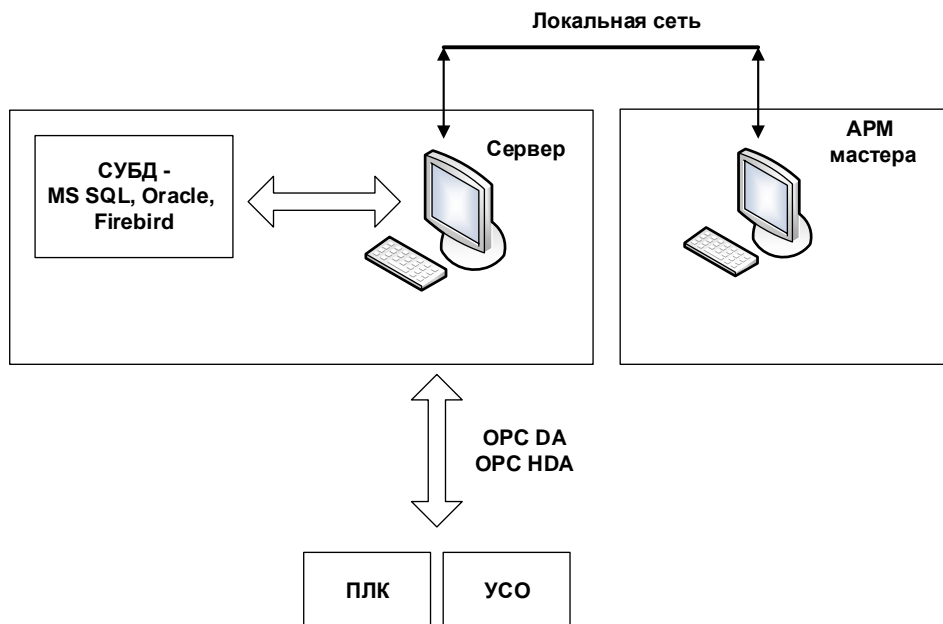


Рисунок 10-10. Структурная схема системы с архивным сервером

Чтобы создать проект такой сетевой архитектуры, необходимо просто добавить в сервер БД-коннектор и настроить его на архивацию данных (или сообщений), как описывалось ранее. После этого достаточно открыть использование портов для связи с СУБД, и сетевой компьютер сможет получать данные из базы данных.

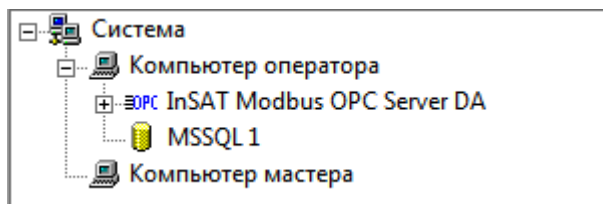


Рисунок 10-11. Система с архивным сервером в дереве системы

Архивных серверов в проекте также может быть несколько.

10.3.3. Резервированная система

Для создания высоконадежных систем используется резервирование. В подобных системах один из компьютеров является основным, а второй – резервным (Рисунок 10-12). Основной компьютер ведет опрос OPC-сервера, с него производится управление. Резервный находится в режиме ожидания, но постоянно синхронизируется с основным. В случае обнаружения отказа (выключение компьютера или режима исполнения, отказ сети, отказ

ОПС-сервера или контроллера, опрашиваемого данным компьютером) происходит переключение управления на резервный компьютер.

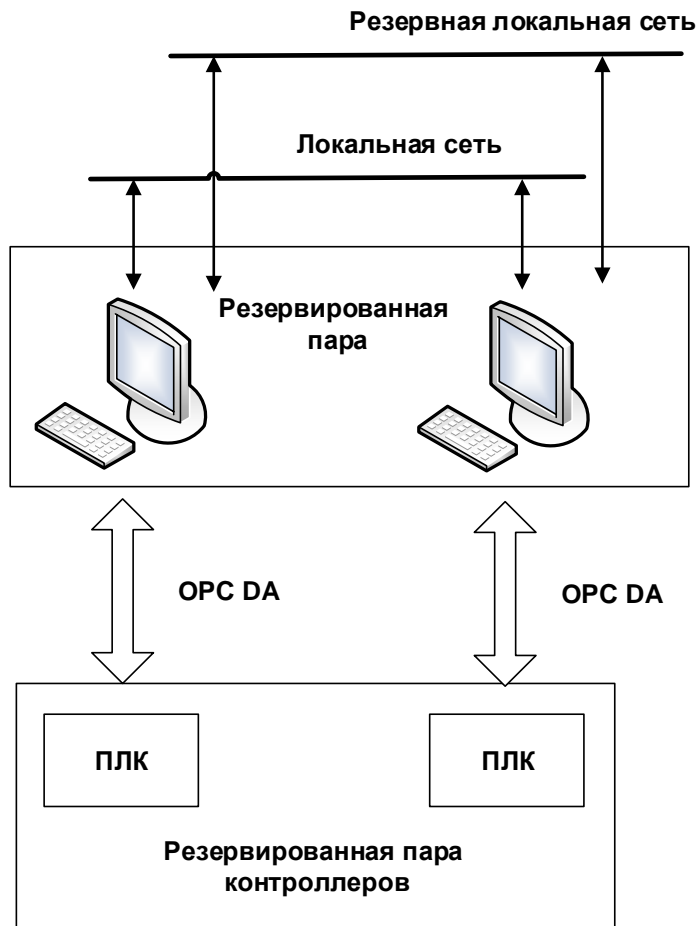


Рисунок 10-12. Схема резервированной системы

В проекте MasterSCADA резервированная пара компьютеров выглядит как один, но у него на вкладке **Режим исполнения** – **Связь** должна быть включена настройка **Использовать резервирование**.

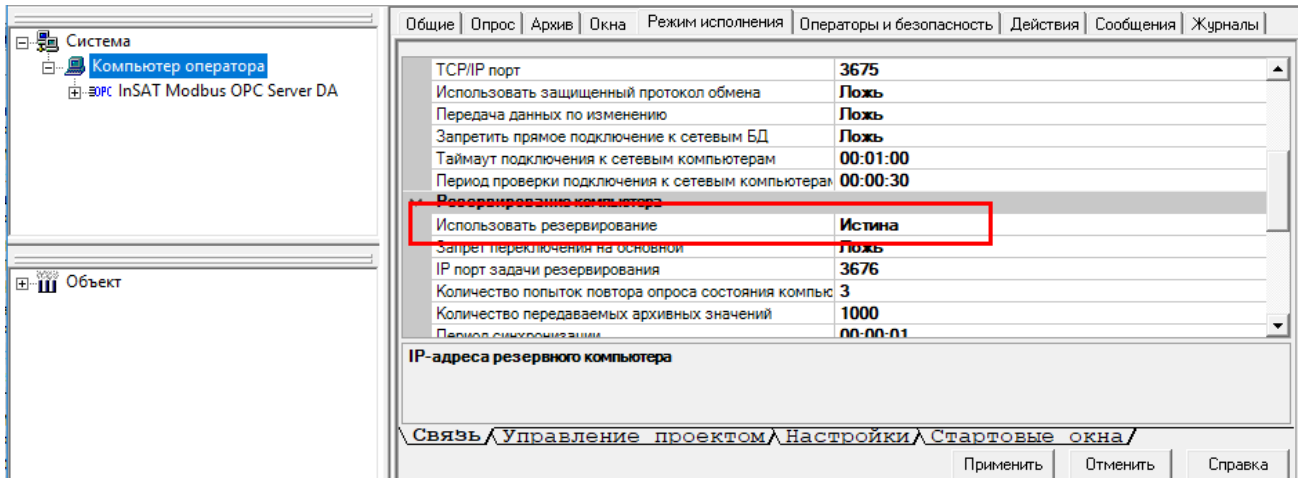


Рисунок 10-13. Резервированный компьютер в дереве системы

Затем у компьютера на этой же вкладке пропишите IP-адреса основного и резервного компьютеров.

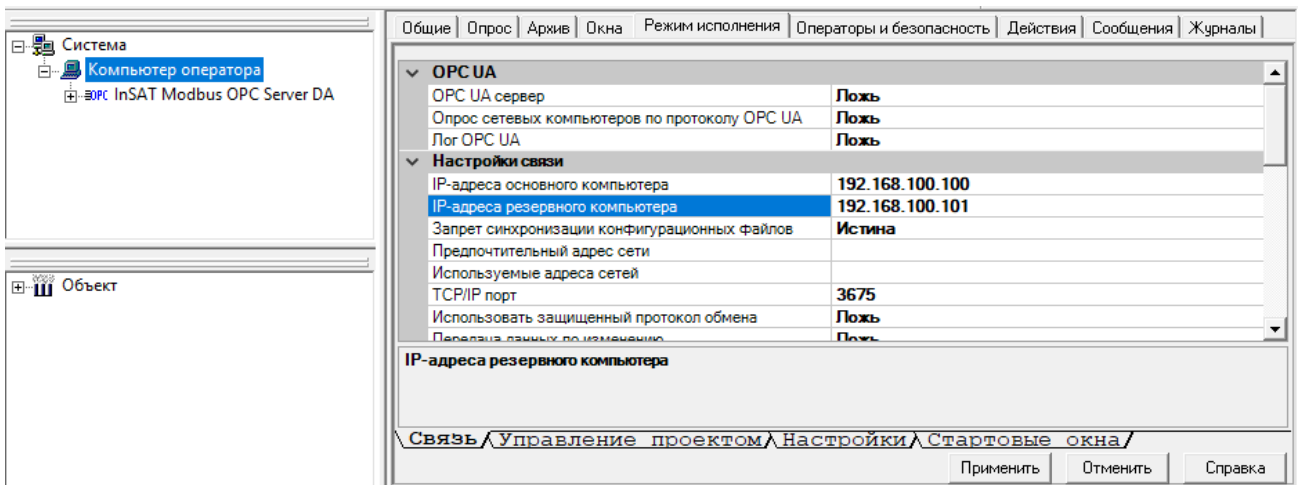


Рисунок 10-14. Настройки связи резервированного компьютера

В резервированный компьютер может быть добавлена и СУБД.

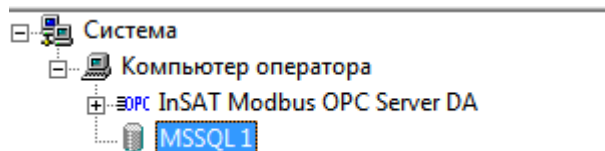


Рисунок 10-15. Резервированный сервер с архивацией в СУБД

При этом архив в СУБД может вестись одновременно на обоих компьютерах параллельно (с полной их синхронизацией) или на каком-то одном.

10.3.4. Мультиклиентная система

Как быть, если в проекте будет использоваться несколько сотен рабочих мест (АРМ)? Добавлять каждый компьютер в систему и прописывать IP-адрес займет очень много времени. Для подобных проектов есть возможность построить мультиклиентную систему – в такой системе под одним логическим компьютером (компьютером в дереве системы) будет работать несколько физических.

Архитектура такой системы в дереве системы выглядит следующим образом.

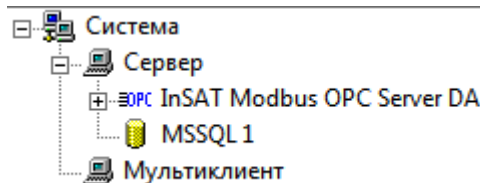


Рисунок 10-16. Мультиклиентная система в дереве системы

Сервер – это MAS (архивный сервер), как правило, резервированный (хотя это и не обязательно). Мультиклиент – обычный компьютер, но в настройках связи у него должна быть включена опция **Запуск на нескольких компьютерах**.

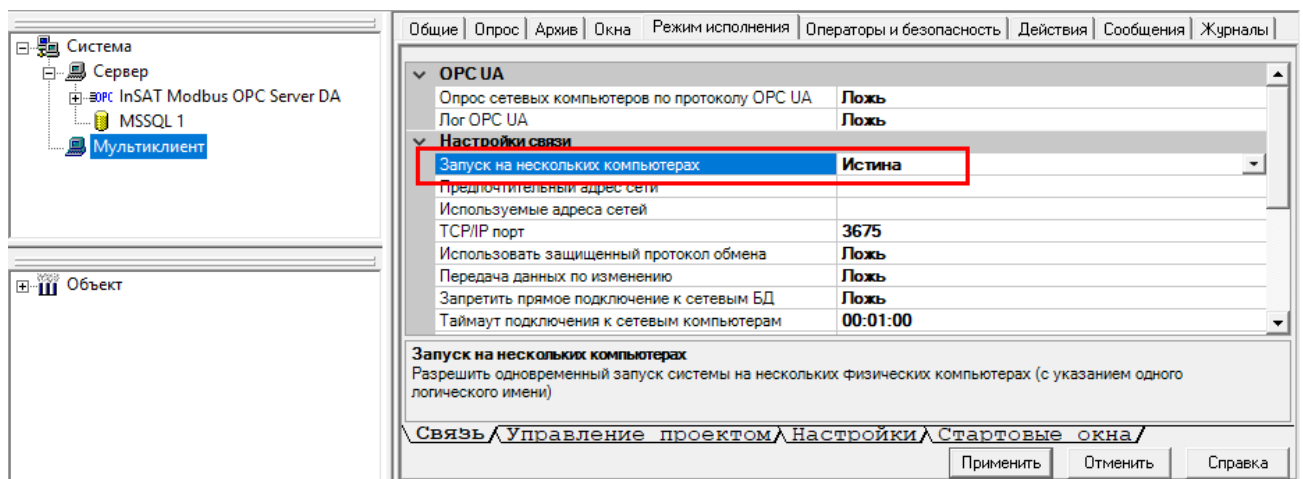


Рисунок 10-17. Настройки мультиклиентного компьютера

При включении этой опции настройка IP-адресов у компьютера скрывается: они ему больше не нужны. Теперь при старте клиент подключается к серверу для обмена данными (навигации по документам и т. д.).

В такой системе есть возможность автоматизированного разворачивания. В дистрибутив архивного сервера входит специальный облегченный инсталлятор для клиентских машин. Для его установки не требуются права администратора, а после его инсталляции есть возможность настройки автоматического обновления не только проекта, но и самой исполнительной системы.

10.3.5. Комбинация архитектур

Мы описали лишь основные виды архитектур. Разумеется, вы можете комбинировать их в зависимости от вашей задачи: допустим вариант с резервированной парой серверов ввода (для опроса), парой резервированных архивных серверов (для ведения архива) и множеством клиентских рабочих мест – в MasterSCADA вы можете создать любую архитектуру.

10.4. Дополнительные материалы

На прилагаемом диске находится специальная документация с подробным описанием настройки сетевых проектов различных архитектур.

Большинство настроек, относящихся к сетевым проектам, находятся на закладке **Режим исполнения – Связь**, подробное описание каждой из них вы можете найти, вызвав справку к этой вкладке.

11. Заключение

Вы прочитали методическое пособие. Это лишь самый первый шаг к освоению MasterSCADA. Вы приоткрыли дверку в большой мир, в котором есть свои потайные уголки и фирменные секреты. Дальнейший шаг – системное освоение документации и учебных примеров, а еще лучше – прохождение учебных курсов непосредственно у разработчика программы. Не забудьте, что на сайте www.insat.ru регулярно появляются новые методические статьи и видеопримеры, облегчающие освоение программы.