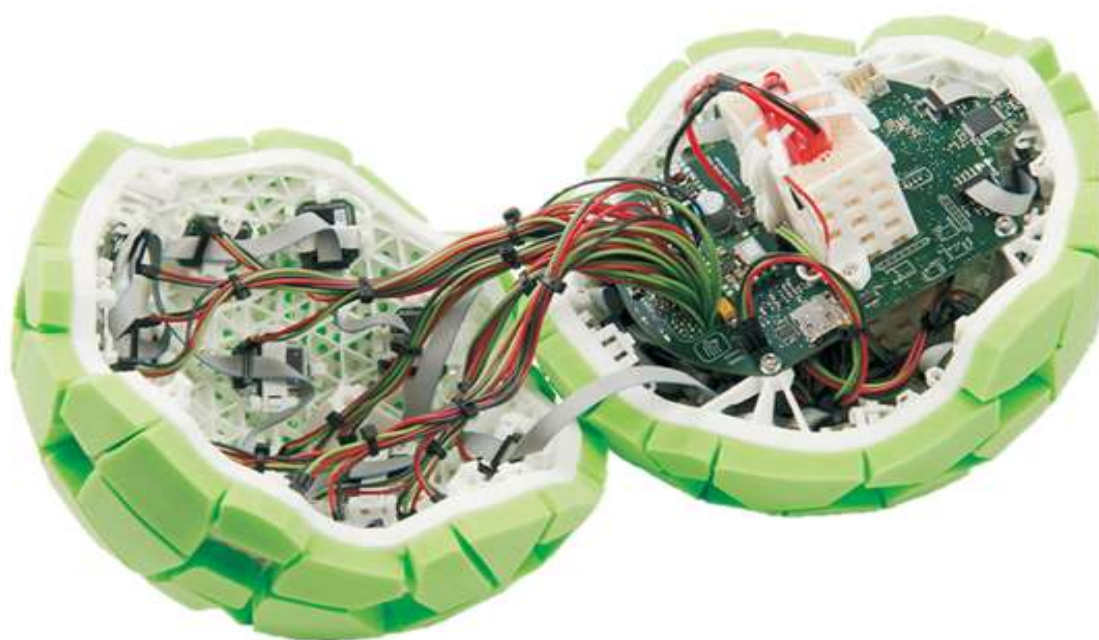


Р.Т.Газиева, Д.Б.Ядгарова  
А.М.Нигматов, Э.О.Озодов

# Master SCADA П.П.

## Учебное пособие



Ташкент 2020

УДК 519.685: 004.94

**Газиева Р.Т. , Ядгарова Д.Б., Нигматов А.М. , Озодов Э.О.**

Мастер SCADA , учебное пособие для студентов специальности  
5311000- Автоматизация и управление технологических процессов и  
производств (в водном хозяйстве)

Аннотация

Учебник подготовлен в соответствии с учебной программой дисциплины «SCADA пакеты». В книге изложены теоретические и практические вопросы, знание которых требуется для создания современного программного обеспечения для автоматизации контроля и управления технологическим процессом.

Учебное пособие рассчитано для студентов высших учебных заведений, обучающихся по направлению специальности 5311000- «Автоматизация и управление технологических процессов и производств (в водном хозяйстве)», также может использоваться и для обучения по другим аналогичным дисциплинам в высших учебных заведениях.

Аннотация

Ўқув кулланма «SCADA пакетлари» фанининг ўқув дастури асосида тайёрланган. Китобда технологик жараёнларни автоматлаштиришда замонавий амалий дастурий пакетларнинг ишлаш принциплари бўйича назарий ва амалий масалалар баён қилинган. Хусусан, ТЖАБТ таркибида «SCADA пакетлари» бўйича асосий тушунбўйича назарий ва амалий масалалар баён қилинган.

Ўқув кулланма олий ўқув юртлирининг 5311000 – “Технологик жараёнлар ва ишлаб чиқаришни автоматлаштириш ва бошқарув (сув хўжалигида)” йўналиши бўйича ўқитилаётган талабалар учун мўлжалланган, шунингдек, ушбу ўқув кулланма олий ўқув юртлирининг шунга ўхшаш фанлари бўйича таълим беришда ҳам қўлланилиши мумкин.

Annotation

The textbook was prepared in accordance with the curriculum for the discipline "SCADA packages". The book contains theoretical and practical issues, knowledge of which is required to create modern software for automation of control and process control.

The textbook made for students of higher educational institutions enrolled in the specialty 5311000- "Automation and control of technological processes and production (in water management)", and can be in usage for training in other similar disciplines in higher educational institutions.

**Рецензенты:** д. т. н., профессор Ишанходжаев Г.К.

т.ф.н., доцент

© Ташкентский институт инженеров ирригации и механизации сельского хозяйства, 2020.

## Содержание

<b>Введение</b>	4
<b>1. Основные правила, действия и определения</b>	6
1.1. Общие данные о SCADA-системах и установка среды разработки	6
1.2. Концепция и архитектура SCADA-систем	14
1.3. Интерфейс и среда разработки Master SCADA	17
1.4. Редактор мнемосхем Master SCADA.	26
<b>2. Создание проекта с использованием объектного подхода</b>	32
2.1. Методика разработки системы	32
2.2. Построение дерева системы и дерева объектов.	35
2.3. Страница свойств элементов палитры и добавление их в мнемосхему	41
2.4. Инструментальная среда порядка разработки проекта Master SCADA.	60
2.5. Переменные объектов и функциональные блоки Master SCADA	63
2.6. Объектно-ориентированный подход	71
<b>3. Разработка проектов АСУТП</b>	86
3.1. Стандарт OPC. Технология OPC в среде Master SCADA.	86
3.2. Режим отладки и запуск разработанного проекта в Master SCADA	99
<b>4. Заключение</b>	102
<b>5. Используемая литература</b>	103

## Введение

Современный этап развития АСУТП характеризуется применением индустриальных технологий создания и внедрения АСУТП на базе серийно выпускаемых промышленных контроллеров, совместимых с персональными компьютерами и мощных программно-технических комплексов (ПТК) поддержки программирования АСУТП – SCADA систем, а также развития и стандартизации сетевых технологий.

Учитывая изложенное, в настоящее время в Республике Узбекистан уделяется большое внимание подготовке в высших учебных заведениях высококвалифицированных специалистов по автоматизации технологических процессов и управления в водном хозяйстве.

В связи с этим, в настоящее время имеется потребность в подготовке и издании учебного пособия, позволяющего студентам направления 5311000-«Автоматизация и управление технологических процессов и производств (в водном хозяйстве)» в соответствии с учебными планами получить необходимые знания и навыки по применению элементов современных цифровых систем управления при создании АСУТП в сельском и водном хозяйстве.

Целью дисциплины «Цифровое управление» является обучение студентов теоретическим основам, методам и средствам создания цифровых систем управления и формирования у них навыков для самостоятельного решения теоретических и прикладных задач автоматизации технологических процессов на технологических объектах сельского и водного хозяйства.

Учитывая это, в данном учебном пособии отражены теоретические и практические вопросы, знание которых требуется для создания современной цифровых систем управления в сельском и водном хозяйстве.

В настоящее время цифровые устройства выполняются в основном с применением микроконтроллеров, сигнальных процессоров и

специализированных микросхем, таких как кодеки, модемы, аналого-цифровые или цифроаналоговые преобразователи. При этом достаточно часто возникают вопросы согласования сигналов, которыми обмениваются данные микросхемы. Иногда требуется инверсия управляющих или информационных сигналов, иногда не совсем совпадают протоколы обмена (вид сигналов записи и чтения, сигналы стробирования данных). В этом случае применение больших интегральных микросхем, таких как ПЛИС CPLD или FPGA нецелесообразно. Это либо невыгодно экономически, либо требуется большая площадь на печатной плате цифрового устройства. В то же самое время применение старых типов микросхем малой интеграции, таких как 1533 или 1564 неприемлемо либо из-за больших габаритов корпуса этих микросхем, либо несовместимости логических уровней, либо недопустимых времен распространения сигнала.

# **I. ОСНОВНЫЕ ПРАВИЛА, ДЕЙСТВИЯ И ОПРЕДЕЛЕНИЯ**

## **1.1. Общие данные о SCADA-системах и установка среды разработки**

SCADA-система – это инструментальная программа, обеспечивающая создание программного обеспечения для автоматизации контроля и управления технологическим процессом в режиме реального времени. Основная цель создаваемой с помощью SCADA программы – дать оператору, управляющему технологическим процессом, полную информацию об этом процессе и необходимые средства для воздействия на него. Основные задачи SCADA-системы: Сбор данных от датчиков и представление их оператору в удобном для него виде, включая графики изменения параметров во времени; Дистанционное управление исполнительными механизмами; Ввод заданий алгоритмам автоматического управления; Реализация алгоритмов автоматического контроля и управления (чаще эти задачи возлагаются на контроллеры, но SCADA-системы тоже способны их решать); Распознавание аварийных ситуаций и информирование оператора о состоянии процесса; Формирование отчетности о ходе процесса и выработке продукции. От надежности, быстродействия и эргономичности SCADA-системы зависит не только эффективность управления технологическим процессом, но и его безопасность.

Специалисты отдела АСУТП промышленного предприятия по изготовлению соды утверждают, что в основном используют такие компоненты, как мониторинг и управление, архивирование технологических параметров, сообщений, подсистему формирования отчетов. Мониторинг и управление, собственно, то, для чего и устанавливается система управления. Архивы параметров, сообщений и отчеты необходимы для оценки и анализа ведения технологического процесса, действий оператора и т.д. Также для них важен один из базовых инструментов SCADA – разграничение прав доступа к управлению по уровням (оператор, технолог, инженер АСУТП). В связи с тенденцией к

интеграции систем управления технологическими процессами и систем управления предприятием все чаще возникает необходимость использования SCADA в качестве источника данных для вышестоящих систем. Некоторые SCADA могут выступать и как сервер консолидации всех технологических данных, и как сервер генерации отчетов на базе этих данных. Если система управления, построена на базе ПЛК одного производителя (к примеру, Siemens SIMATIC), то обмен данными между контроллерами и SCADA происходит с помощью встроенных драйверов протоколов связи. Некоторые независимые от производителей оборудования SCADA предлагают набор драйверов ко многим (но не всем) имеющимся на рынке контроллерам и интеллектуальными приборам. Наиболее универсальный способ взаимодействия – это использование драйверов, разработанных в соответствии со стандартом OPC. Такие OPC-серверы могут быть разработаны производителями контроллеров или независимыми разработчиками, а использоваться вместе с любой SCADA-системой. Для эффективной работы с OPC- серверами SCADA должна использовать их напрямую, по технологии «OPC в ядре системы», а не через промежуточные интерфейсы. Некоторые SCADA являются вертикально-интегрированными: в их состав входят системы программирования для свободно-программируемых контроллеров. В них также используются внутренние драйверы для связи с контроллером. Такие SCADA позволяют создать ПТК с использованием оборудования разных производителей.

Системы технологической автоматизации обычно разделены на 3 уровня: нижний, средний и верхний. Выше них находится уровень управления производством в целом. Нижний уровень – это сами датчики и исполнительные механизмы Средний уровень – контроллеры. На среднем уровне происходит прием входных данных; первичная обработка данных; автоматическое формирование и выдача управляющих воздействий на исполнительные механизмы; обмен информацией с верхним уровнем. Верхний уровень – это и

есть уровень SCADA. На этом уровне происходит: сбор, обработка и хранение информации, полученной на среднем уровне; визуализация текущей и архивной информации в удобном оператору виде (мнемосхемы, графики, тренды, журналы сообщений); ввод команд оператора; формирование отчетности о результатах технологического процесса; обмен информацией с верхним уровнем. Управление предприятием производится на двух уровнях: MES (Manufacturing Execution Systems) – система управления производством продукции в реальном времени. Этот уровень служит для планирования производственных заданий для технологических процессов, построения сводных отчетов, глубокого анализа процесса (например, прогнозирование, построение энергетического и материально-го баланса и др.). Для этих целей также может быть использован инструментарий SCADA. ERP (Enterprise Resource Planning) – система автоматизированного управления административно-финансовой и административно-хозяйственной деятельностью предприятия. На этом уровне используются другие специализированные системы, например, SAP R3.

### ***Функции SCADA имеют следующие аспекты***

■ Мнемосхемы Мнемосхема – это графическое изображение (с помощью встроенного в SCADA графического редактора) технологической схемы с визуализацией значений датчиков, состояния исполнительных механизмов и др. параметров. Для визуализации используется не только отображение значений в виде цифр и надписей, но и изменение визуальных свойств отображаемых графических объектов. Например, в емкости изменяется уровень жидкости, а ее цвет изменяется в зависимости от температуры (динамизация). Исполнительные механизмы могут не просто показывать свое состояние каким-то графическим признаком (например, цветом), но и наглядно показывать свою работу – например, вращением лопастей насоса, движением ленты конвейера и т.п. (анимация).



■ Архивы Получаемые от контроллеров данные SCADA складывает в архивы. Предварительно данные могут быть обработаны (отфильтрованы, усреднены, сжаты и т.п.). Часто используется не регулярная запись, а запись по изменению с использованием порога чувствительности («мертвой зоны»). Длительность хранения настраивается в SCADA индивидуально для каждого параметра и может составлять до нескольких лет.

■ Тренды Тренд – это графическое отображение изменения параметра во времени. Тренды в SCADA- системах могут показывать изменение параметра за всю длительность его хранения в архиве. Оператору предоставляется возможность изменять масштаб, как времени, так и самого параметра. В развитых системах в тренд встроены различные инструменты анализа графика, сравнения его с уставкой или другим параметром, сглаживание или фильтрация, отметки на графике событий (например, нарушение границ) или закладок для памяти и многое другое.

■ Таблицы Зачастую технологу удобнее просматривать архивы не в графическом виде, а в виде таблиц. Обычно эти таблицы можно не только просматривать, но и экспортировать в другие системы.

■ Графики Обычно SCADA позволяют смотреть и зависимость одних параметров от других, тоже во времени. Хотя это функция и менее востребована технологами, чем тренды.

■ Гистограммы и диаграммы Другим распространенным способом представления параметров являются гистограммы (столбиковые диаграммы).

■ Сообщения – это текстовые строки, которые информируют оператора о событиях на объекте в той последовательности, в которой эти события происходят. Они всплывают на экране или отображаются в специально выделенной для этого зоне.

■ Журналы сообщений Журналы сообщений служат для отображения списков сообщений в том порядке, как они появлялись и были сохранены в архив. Как

правило, используются разные экземпляры журналов для разных зон процесса, разных категорий сообщений, разных приоритетов.

- Контроль прав доступа для того, чтобы оператор мог совершить те или иные действия, ему должны быть администратором предоставлены соответствующие права – например, право управлять исполнительным механизмом, или право изменить задание регулятору. В начале смены оператор регистрируется в системе, и она предоставляет ему выполнять только те действия, которые ему разрешены администратором.

- Журнал действий оператора Управление технологическим процессом очень ответственная задача, поэтому все действия оператора записываются для контроля в специальный журнал, который может быть проанализирован в случае нештатных ситуаций.

- Формирование отчета Удобная среда разработки отчетов позволяет легко и быстро подготовить отформатированные и насыщенные информацией отчеты.

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 Master SCADA.

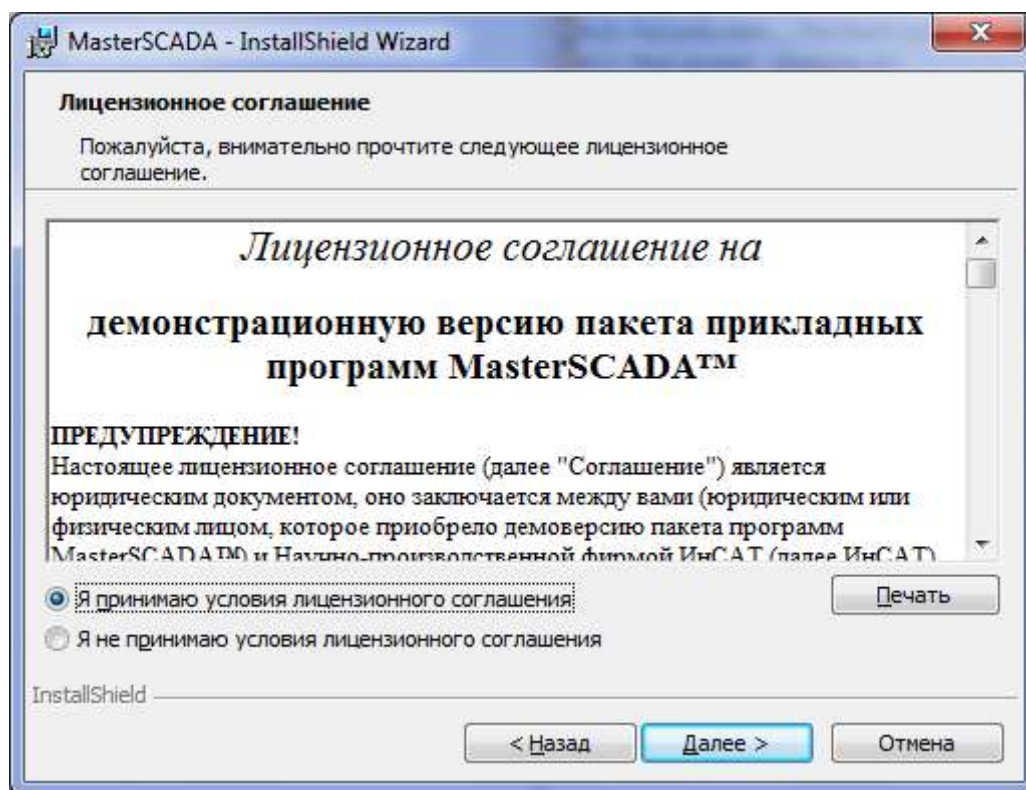
MasterSCADA поддерживает работу с операционными системами Windows XP (SP3), Windows 7, Windows 8, Windows 10.

Установка MasterSCADA ничем не отличается от большинства типовых установок программ. Запустите файл **setup.exe1** 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.

Если приложение Microsoft .NET Framework не найдется в списке программ

компьютера, то инсталлятор MasterSCADA предложит установить его автоматически.

Перед тем как нажимать кнопку далее в диалоговом окне, показанном на рисунке 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.



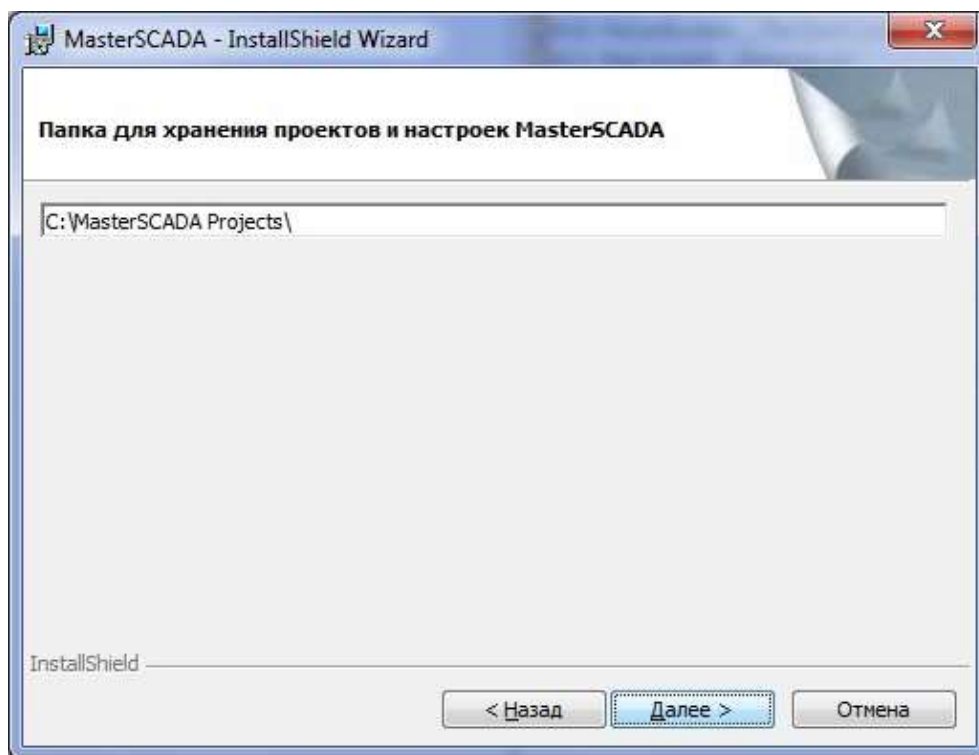
**Рисунок 1.1. Окно с лицензионным соглашением**

Обратите особое внимание на Рисунок 1.2. Он определяет место, в котором по умолчанию будут сохраняться ваши проекты и библиотеки. Окно появляется только при первой установке MasterSCADA. В дальнейшем изменить место расположения рабочего каталога MasterSCADA Projects можно будет только из интерфейса MasterSCADA.

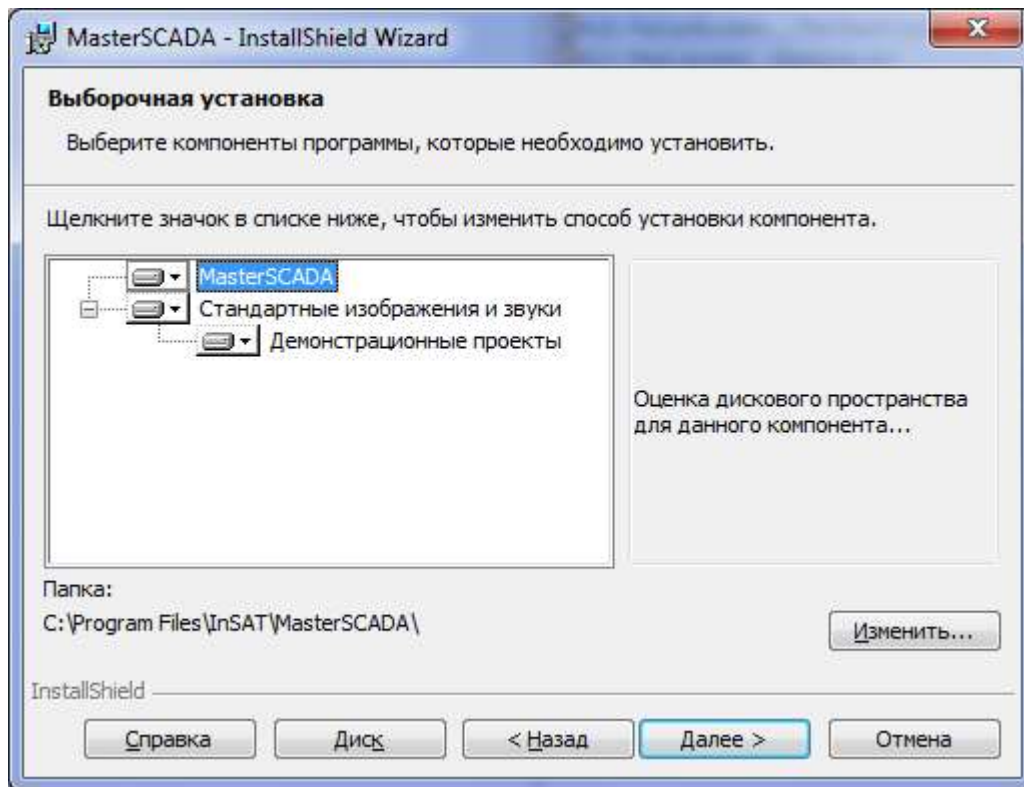
Задать место установки пакета программ MasterSCADA и определить, какие компоненты необходимо установить на компьютер можно в окне, показанном на Рисунок 1.3.

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 **Insat 2/MasterSCADA** по умолчанию будут создана в том каталоге, куда устанавливаются все приложения на компьютере. Вы можете изменить место установки, нажав на кнопку Изменить в диалоговом окне.



**Рисунок 1.2. Выбор места хранения рабочего каталога**



**Рисунок 1.3. Выбор места установки MasterSCADA**

Запуск MasterSCADA в режим разработки (далее этот режим мы будем называть английской аббревиатурой DT - от термина Design Time, что в переводе и обозначает режим разработки) производится так же, как и для 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**.

## 1.2. Концепция и архитектура SCADA-систем

Термин SCADA обычно относится к централизованным системам контроля и управления всей системой, или комплексами систем, осуществляемого с участием человека. Большинство управляющих воздействий выполняется автоматически УСО (RTU) или ПЛК (PLC). Непосредственное управление процессом обычно обеспечивается RTU или PLC, а SCADA управляет режимами работы. Например, PLC может управлять потоком охлаждающей воды внутри части производственного процесса, а SCADA система может позволить операторам изменять уставки для потока, менять маршруты движения жидкости, заполнять те или иные ёмкости, а также следить за тревожными сообщениями (алармами), такими как — потеря потока и высокая температура, которые должны быть отображены, записаны, и на которые оператор должен своевременно реагировать. Цикл управления с обратной связью проходит через RTU или PLC, в то время как SCADA система контролирует полное выполнение цикла.

Сбор данных начинается в RTU или на уровне PLC и включает показания измерительного прибора. Далее данные собираются и форматируются таким способом, чтобы оператор диспетчерской, используя HMI, мог принять контролирующие решения — корректировать или прервать стандартное управление средствами RTU/PLC. Данные также могут быть записаны в архив для построения трендов и другой аналитической обработки накопленных данных. (Рисунок 1.4.).

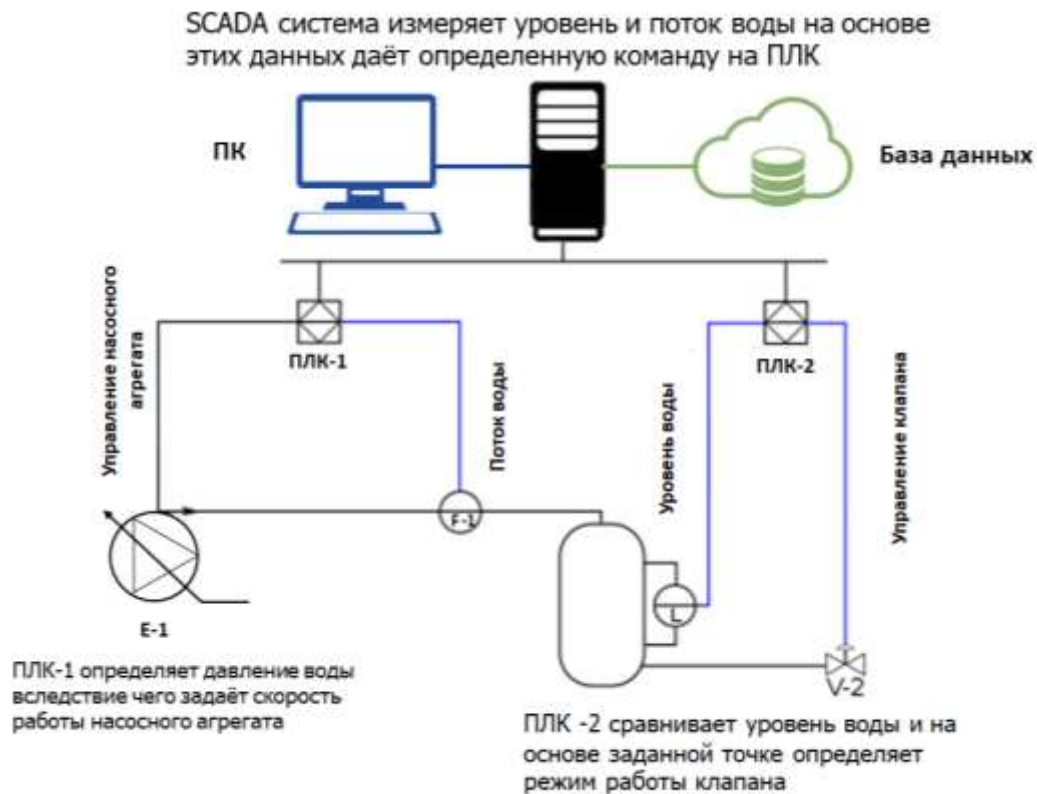
В зависимости от сложности управляемого технологического процесса, а также требований к надёжности, SCADA-системы строятся по одной из следующих архитектур:

- Автономные
- Клиент-сервер
- Распределённый

**Автономные**- при использовании данной архитектуры система состоит из одной или нескольких рабочих станций оператора, которые не "знают" друг о друге. Все функции системы выполняются на единственной (нескольких независимых) станции. Преимущества системы - простота. Недостатки системы —низкая отказоустойчивость; не обеспечивается истинность данных (исторические данные могут отличаться между разными станциями).

**Клиент-сервер** - в данном случае система выполняется на сервере, а операторы используют клиентские станции для мониторинга и управления процессом. Высокнадёжные системы строятся на базе двойного либо тройного резервирования серверов и дублирования клиентских станций оператора, дублирования сетевых подключений сервер-сервер и клиент-сервер. При данной архитектуре уже возможно разделение функций SCADA-системы между серверами. Например, сбор данных и управление ПЛК выполняется на одном сервере, архивирование данных - на втором, а взаимодействие с клиентами - на третьем.

**Распределённый**- при использовании архитектуры распределенной системы управления (PCY) вычисления осуществляются на нескольких взаимосвязанных вычислительных устройствах, часто с функцией взаимного резервирования. Распределенные SCADA-системы с взаимным резервированием отличаются повышенной надежностью.



**Рисунок 1.4. Концепция управления в SCADA системе.**

В настоящее время существуют решения, основанные на открытом исходном коде.

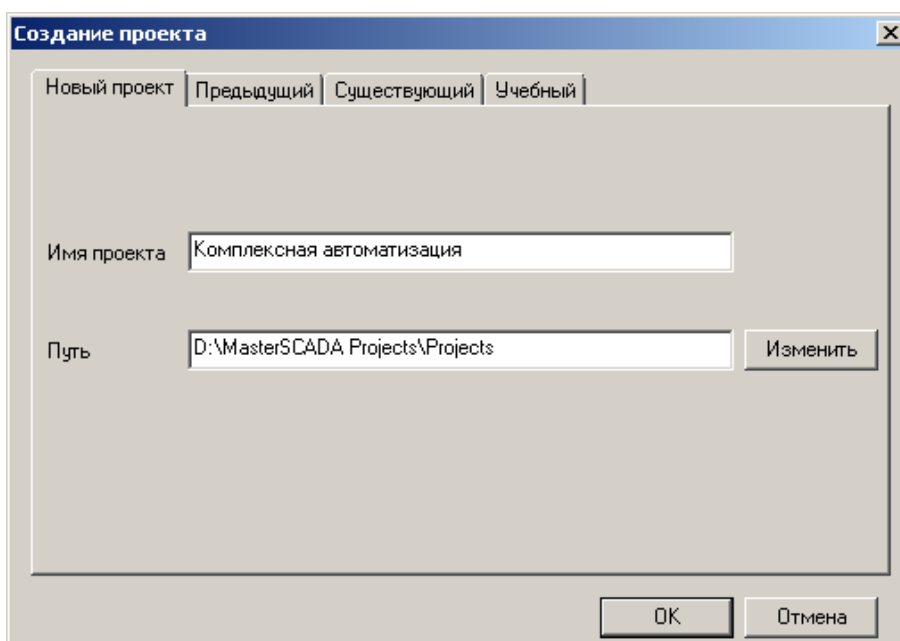
Одной из первых систем с открытым кодом, является немецкая система Lintouch. Lintouch это HMI с открытым исходным кодом, который позволяет создавать пользовательские интерфейсы для автоматизации процесса. Lintouch работает на большинстве популярных аппаратных и программных платформ, легко и просто масштабируется. Lintouch является свободным программным обеспечением и распространяется под лицензией GNU General Public License. С использованием редактора Lintouch вы можете легко создать свой собственный HMI путем разработки и тестирования графических экранов. Позже вы можете перенести созданный проект Lintouch на устройстве, где она будет выполняться в Lintouch Runtime.



### 1.3.Интерфейс и среда разработки Master SCADA

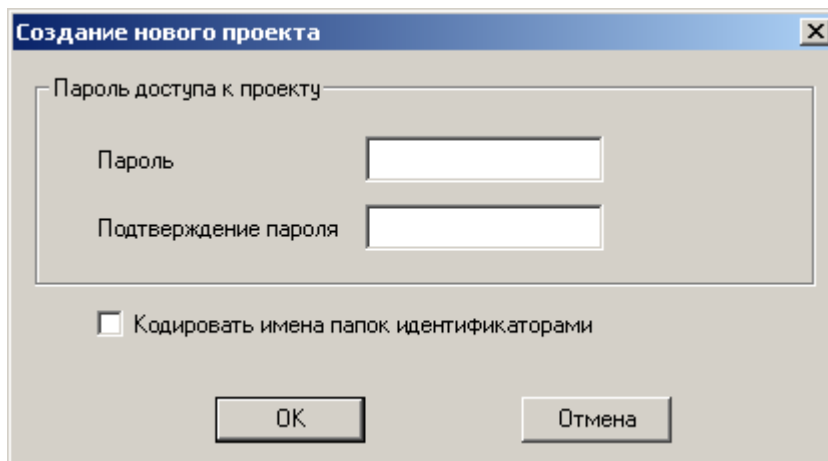
Для запуска программы Master SCADA Нажимаем главную кнопку **Windows Пуск** и выбираем **Все программы/ MasterSCADA/ MasterSCADA** после чего откроется диалоговое окно.

В открывшемся диалоговом окне (Рисунок 1.5.) необходимо указать, с каким проектом вы будете работать: новым, учебным или иным, ранее созданным.



**Рисунок 1.5. Открытие/создание проектов**

Если вы создаете новый проект, то после создания и подтверждения его имени появится диалоговое окно, показанное на Рисунок 1.5.

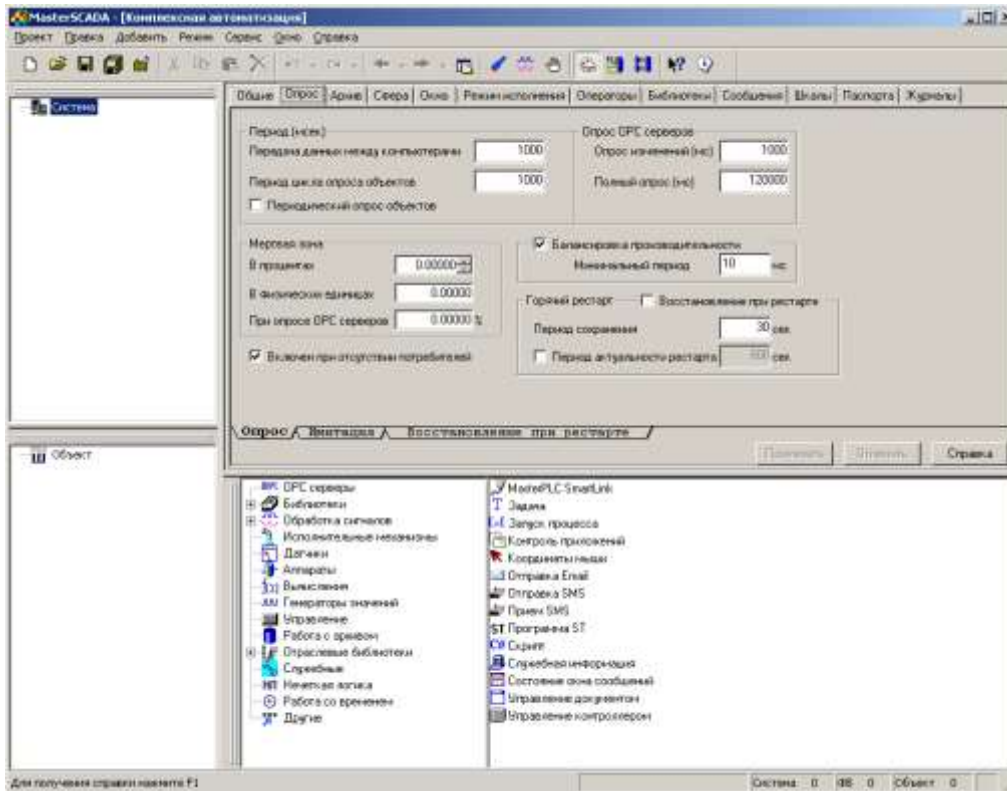


**Рисунок 1.6. Установка пароля на разработку проекта**

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 компьютера 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.

Перед пользователем открывается основное поле его деятельности – менеджер проекта (Рисунок 1.7.). 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.



**Рисунок 1.7. Менеджер проектов. Вид по умолчанию**

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

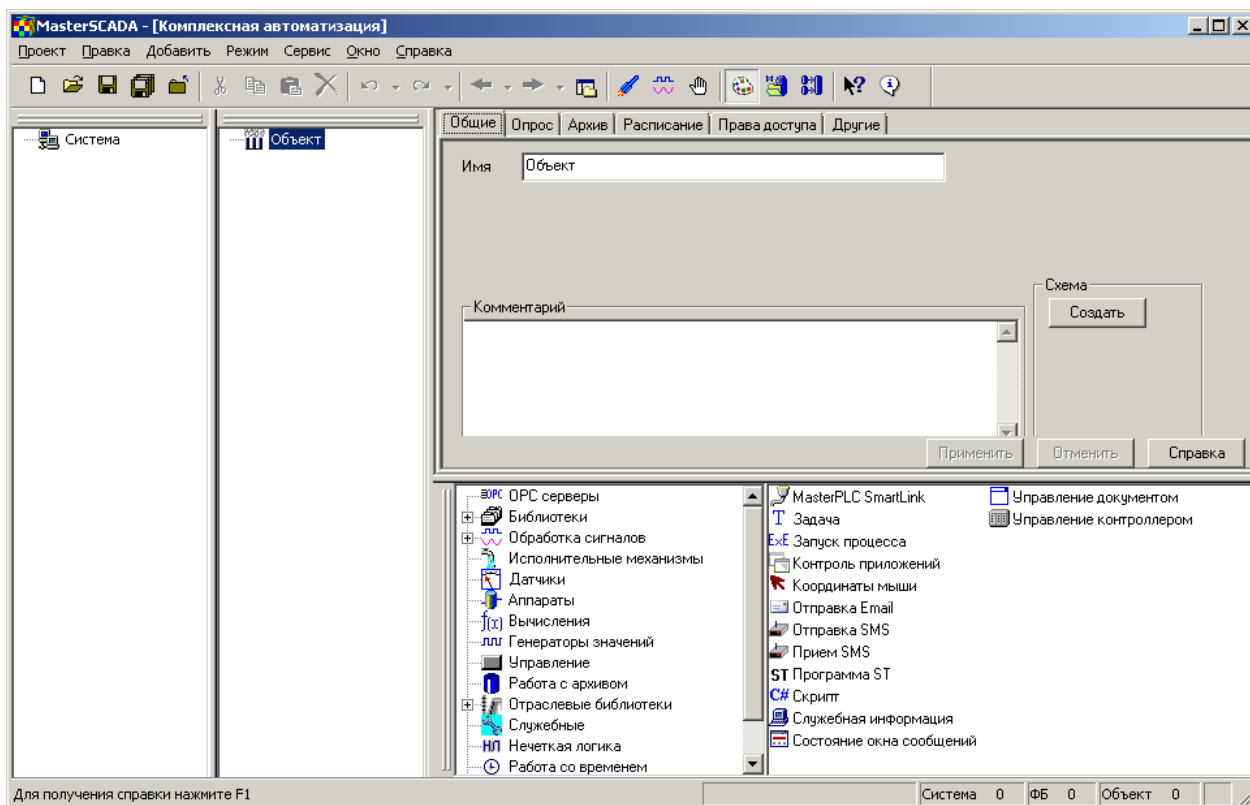
Рассматривая устройство окна программы, мы будем говорить о том расположении его частей, которое предлагается программой «по умолчанию». Но пользователь всегда может изменить их «под себя», исходя из формата своего монитора или собственных представлений об удобстве.

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 левая кнопка мыши на верхней кромке рамки дерева (поле для нажатия обозначено двойной выпуклой линией) и, удерживая ее нажатой, перетащите окно.

Последнее время стали популярны широкоформатные мониторы, и все чаще пользователи выбирают способ размещения деревьев, показанный на Рисунок 1.8.




**Рисунок 1.8. Менеджер проектов. Произвольный вид**

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: добавьте элемент в дерево, просмотрите 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 в непонятный элемент и получить разъяснение, см. Рисунок 1.9.).

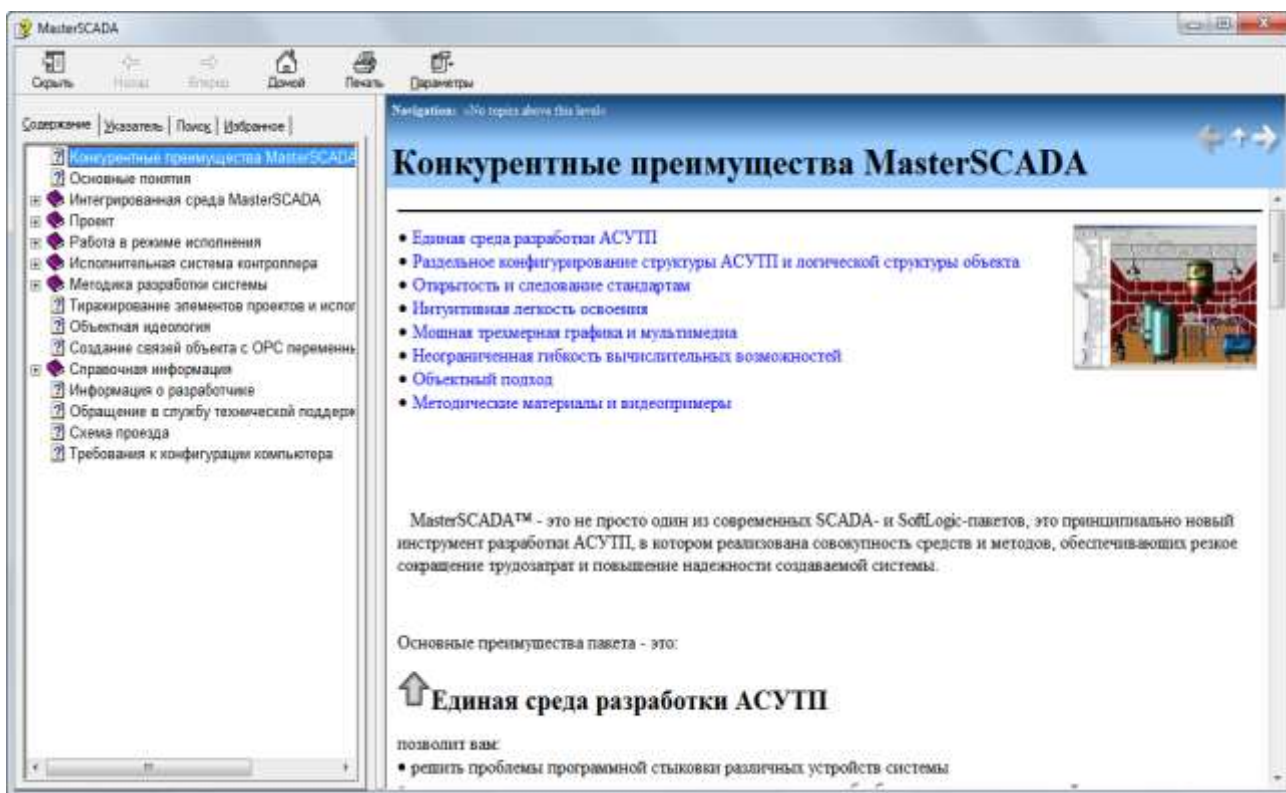
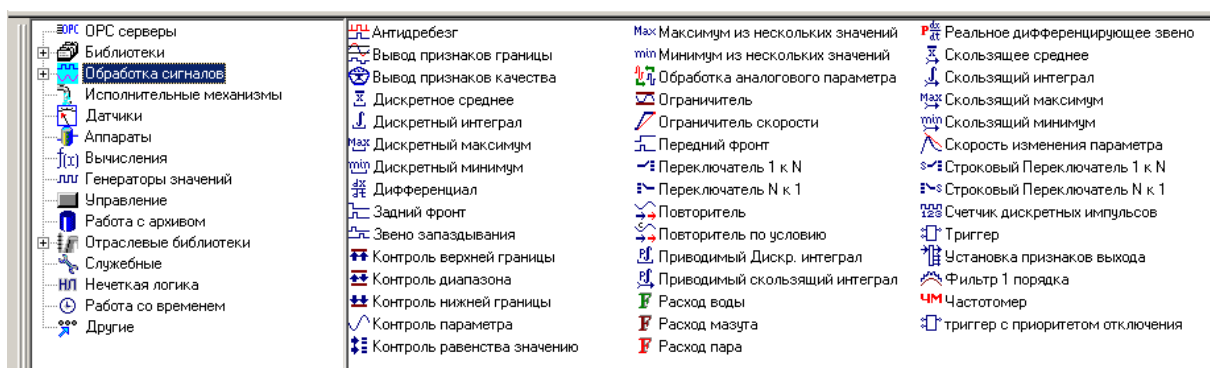



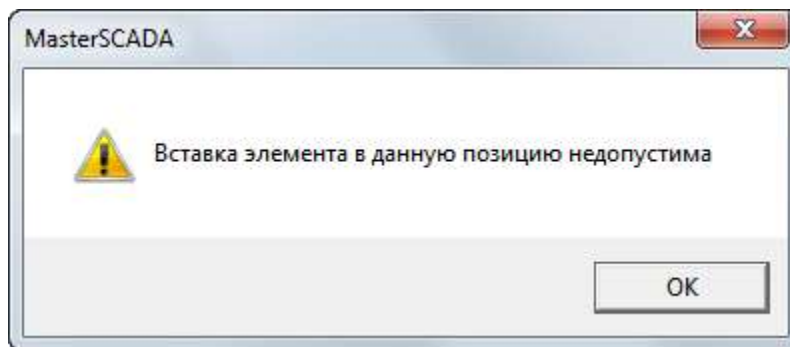
Рисунок 1.9. Окно справочной системы

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 1.10).



**Рисунок 1.10. Палитра Master SCADA**

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 1.12.).



**Рисунок 1.12. Сообщение о некорректной работе с палитрой MasterSCADA**

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.

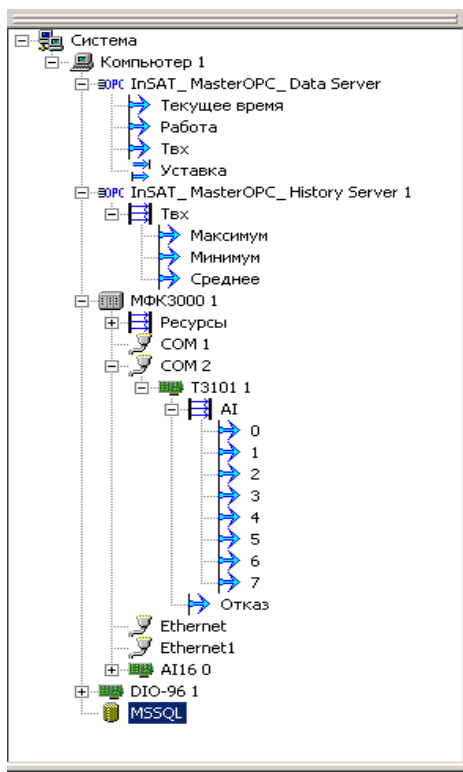
Построение дерева системы начинается с элемента Компьютер. Физический компьютер даже в самом простом проекте выполняет несколько функций таких как:

- 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 1.13.), представляющие собой понятия внешнего мира, с которым взаимодействует

MasterSCADA, установленная на данном реальном компьютере: Контроллер (🖨️), Модуль ввода/вывода(🖨️), OPC-сервер (🖨️), MasterLink (🖨️), БД-коннектор (🖨️). Рассмотрим эти понятия подробнее.



**Рисунок 1.13. 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».

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.

## 1.4. Редактор мнемосхем Master SCADA

Основным средством взаимодействия с оператором являются мнемосхемы – окна с представлением информации в графическом виде. На мнемосхемах отображаются состояние исполнительных механизмов и аппаратов, значения параметров системы, аварии и т.д.

Мнемосхемы в MasterSCADA принадлежат объекту. Число мнемосхем в проекте не ограничено.

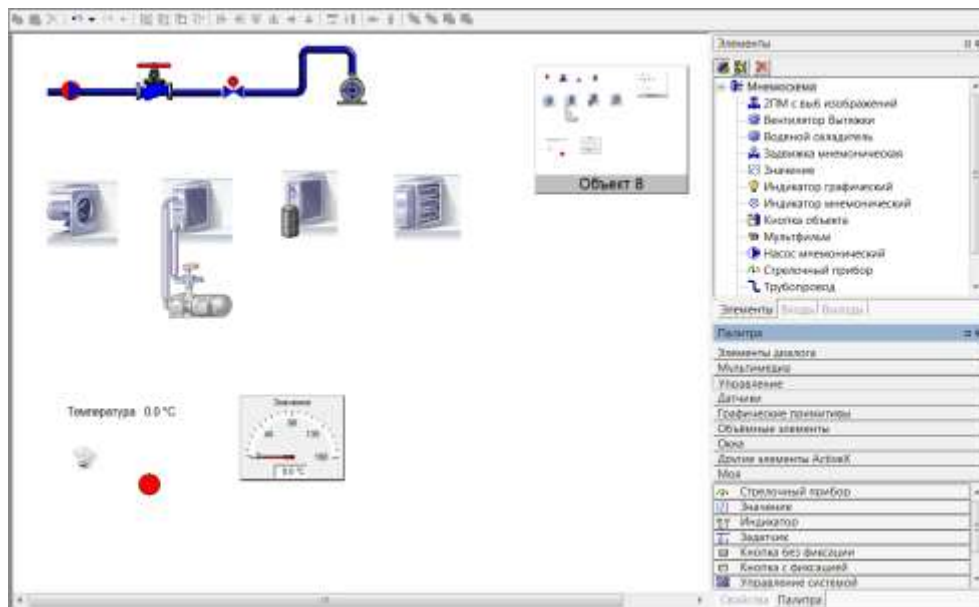
Число элементов на мнемосхеме также не ограничено. Библиотеки стандартных элементов содержат множество элементов, включая объемные элементы со встроенным индикатором заполнения, элементы для создания пользовательских диалогов, элементы, воспроизводящие полный комплект приборов щитового контроля и управления.

Имеется встроенный редактор для создания мультфильмов (с регулируемой прозрачностью изображения) с различными законами трансформации исходных графических файлов (покадровый показ, прокрутка в любом направлении, изменение резкости или размера и т.п.).

Поддерживаются основные мультимедийные форматы: avi, gif, jpg, png, gif, tiff, bmp.

Объемные трубопроводы произвольной конфигурации создаются в несколько щелчков мыши.

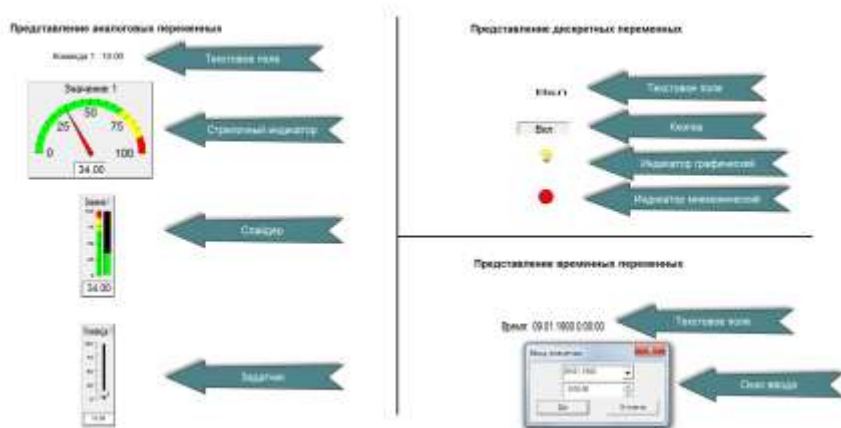
Основной способ создания мнемосхем - перетаскивание элементов из дерева объектов: как включенных в проект из библиотек визуальных функциональных блоков



**Рисунок 1.14. Визуальные блоки рабочей среды**

Переменные, которые, уже обладающих всей необходимой функциональностью (динамизированное изображение, окно управления и т.п.), так и созданных пользователем объектов со своими изображениями и окнами управления.

Например, для переменной может быть выбран способ отображения: в виде текстового значения или в виде одного из типовых приборов щитового монтажа.

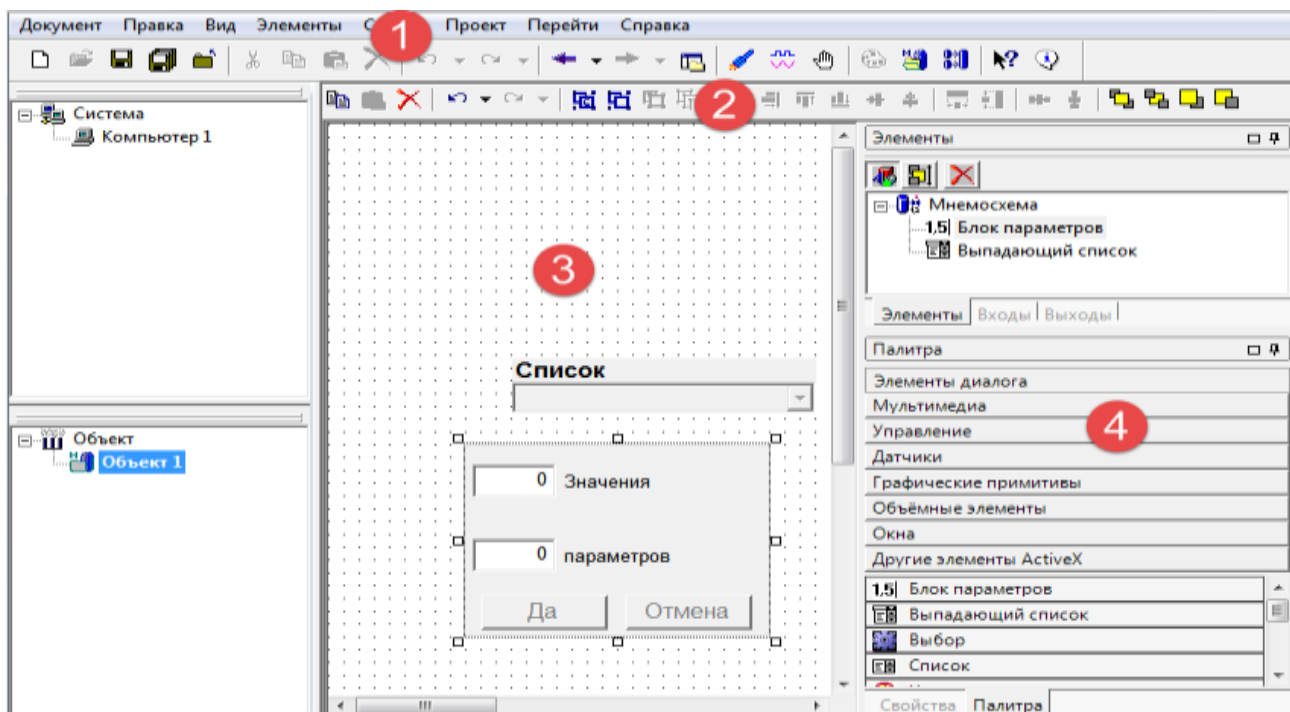


**Рисунок 1.15. Типовые приборы визуализации**

Объект может быть представлен на мнемосхеме в виде кнопки с текстом, либо уменьшенным изображением его мнемосхемы, также существует возможность создания изображений объекта – который можно иначе назвать «виджетом», т.е. элементом способным не только открывать мнемосхемы объектом, но и отображать актуальную информацию о состоянии объекта (например – наличие аварий, значение критических параметров, состояние главных исполнительных механизмов).

Также в качестве элемента мнемосхемы может быть использован любой элемент управления ActiveX, с возможностью динамизации любого его свойства, а также одного из стандартных свойств (положения, размера, отрисовки, мигания и т.п.). Связь с деревом проекта в этом случае осуществляется через переменные динамизации.

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



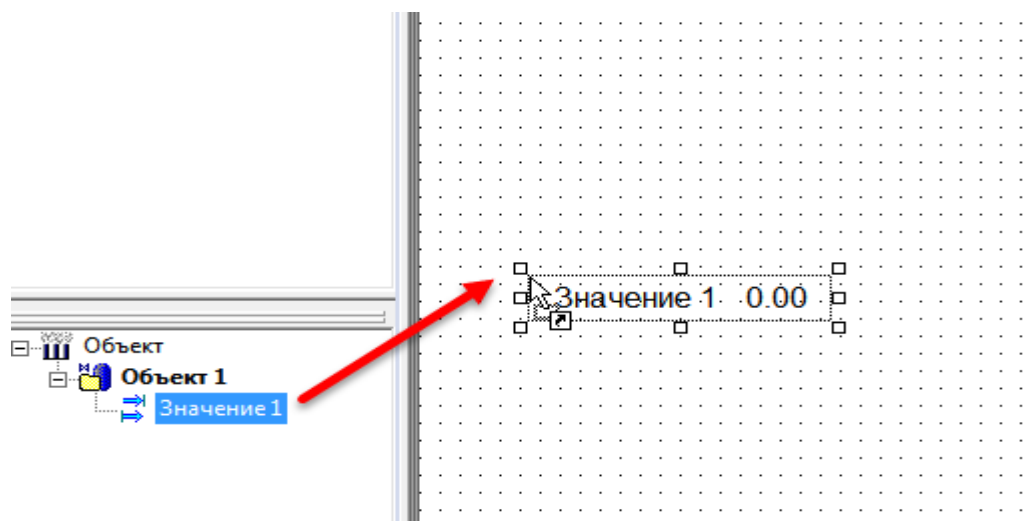
**Рисунок 1.16. Внешний вид редактора**

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.

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

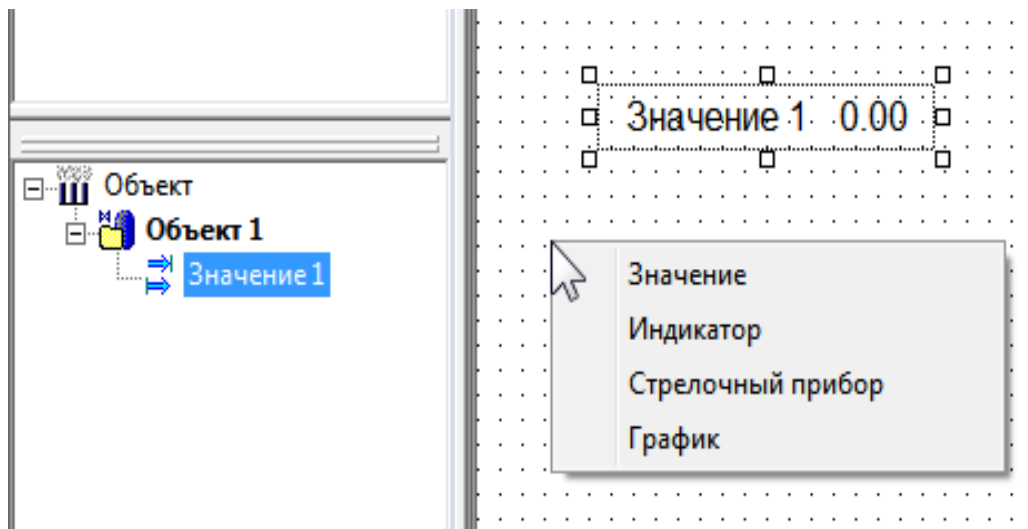
имеют контролы (или даже несколько контролов) для отображения на мнемосхеме. Чтобы добавить контроль элемента на мнемосхему, достаточно перетащить элемент мышью в поле мнемосхемы. Для примера создадим в дереве значение и вытащим его на мнемосхему левой клавишей мыши.



**Рисунок 1.17. Добавление переменных на мнемосхему**

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

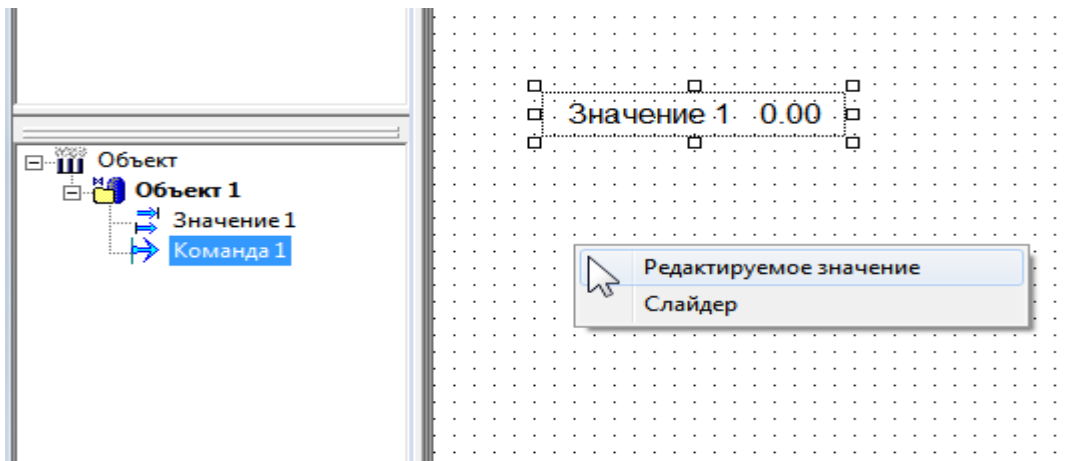
Снова вытащим элемент, но теперь правой кнопкой мыши (Рисунок 1.18.). Поскольку элемент Значение 1 имеет несколько контролов, появилось всплывающее окно, в котором мы можем выбрать, какой именно контрол хотим добавить.



**Рисунок 1.19. Добавление переменных с выбором контрола**

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.

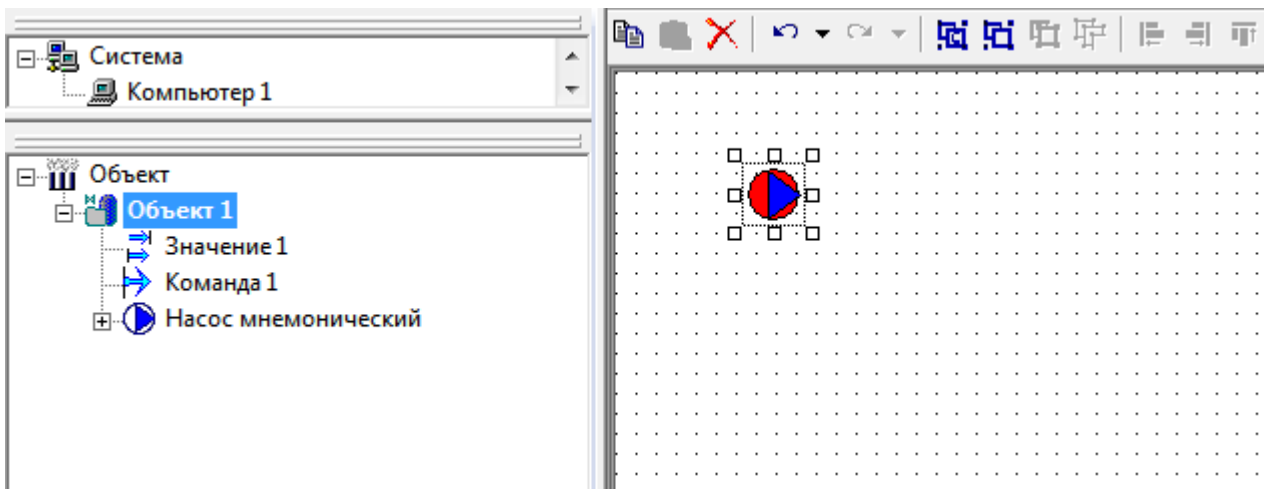


**Рисунок 1.20. Варианты контроля команды**

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 1.21. Контроль 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.

## **II. СОЗДАНИЕ ПРОЕКТА С ИСПОЛЬЗОВАНИЕМ ОБЪЕКТНОГО ПОДХОДА**

### **2.1. Методика разработки системы**

Master SCADA во многом новаторская программа. Впервые в этом классе продуктов инструментальная система SCADA-пакета предоставила пользователю объектно-ориентированный метод разработки проектов. Метод, который благодаря очевидности и наглядности существенно облегчает освоение системы и обеспечивает простое тиражирование любых проектных решений. Понятие объектно-ориентированной методики в первую очередь ориентировано на специалистов в области технологических процессов и их автоматизации в пакете реализованы две объектных иерархии: технологического объекта (например, цех/аппарат/исполнительный механизм/датчик) и системы автоматизации (компьютер/контроллер/модуль ввода-вывода/вход-выход). Но и программист может увидеть некоторые знакомые для него признаки объектной методологии: наследование свойств, возможность ограничения области видимости параметров объектов и т.п. Практика показала, что такая методика абсолютно понятна новичкам, впервые имеющим дело со SCADA-системами, и профессионалам, знакомыми с несколькими программными пакетами этого класса.

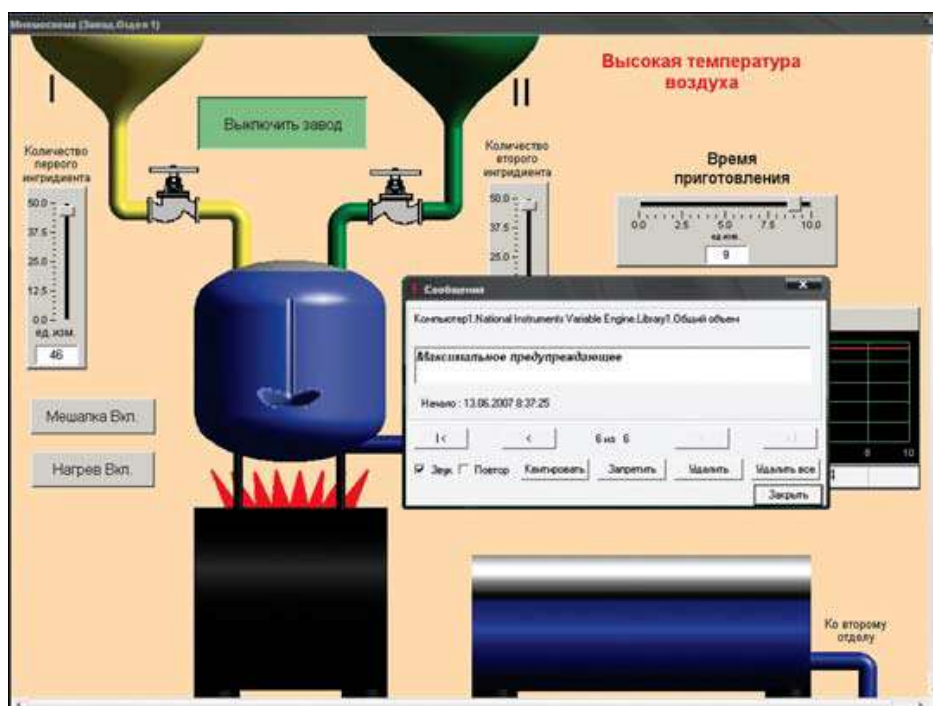




**Рисунок 2.1. структура иерархия в Master SCADA**

Master SCADA вертикально-интегрированная система, она позволяет не только реализовать АРМы операторов-технологов, но и запрограммировать управляющие контроллеры, причем распределение и перераспределение программ между компьютерами и контроллерами также возможно на любой стадии разработки проекта. Это связано с тем, что объект в объектной иерархии Master SCADA принадлежит конкретному компьютеру и, возможно, связанному с ним контроллеру. Поэтому, если у объекта (например, технологической установки или аппарата) имеется какая-либо логика контроля и управления, то она при наличии связанного с объектом контроллера будет автоматически загружена в него сразу после старта системы в режиме исполнения. Интересно, что при использовании predefined библиотечных объектов, таких,

например, как исполнительные механизмы, они ведут себя так же, как и пользовательские объекты: загружают в контроллер свою контрольно-управляющую часть, в то время как информационно-визуальная (динамический символ, окно управления, сообщения) остается работать на операторской станции.



**Рисунок 2.2. Мнемосхема технологического процесса**

С точки зрения назначения компьютерных узлов распределенной системы можно выделить серверы ввода-вывода, операторские станции, клиентские станции, Интернет-клиенты и серверы баз данных. В данном случае, вероятно, в пояснениях нуждается только разделение на серверы ввода-вывода и операторские станции. И серверы ввода-вывода и операторские станции обмениваются данными с нижним уровнем системы (контроллерами, модулями удаленного ввода-вывода, счетчиками коммерческого учета, интеллектуальными датчиками и т.п.), но серверы не имеют визуальности. На них устанавливается

один и тот же комплект MasterSCADA, а функциональная разница определяется разработчиком проекта. Клиентские станции получают информацию от серверов и операторских станций. Для поддержки Интернет-клиентов на них дополнительно должен быть установлен Интернет-сервер. Сами Интернет-клиенты ни в чем, кроме браузера, не нуждаются. Для связи с базами данных в MasterSCADA предусмотрены так называемые БД-коннекторы модули связи с SQL-совместимыми СУБД. На сегодня есть поддержка для MS SQL Server, Oracle, Sybase, Interbase (Firebird), Linter. При необходимости этот список может быть расширен. В следующей версии в состав MasterSCADA будет включен новый тип исполнительной системы выделенный архивный сервер.

## **2.2 Построение дерева системы и дерева объектов**

Дерево системы описывает структуру создаваемой системы (один или несколько компьютеров), а также содержит источники данных – OPC серверы, контроллеры, связи с базами данных.

Дерево объектов описывает иерархическую структуру проекта – в данном дереве создается программа, реализующая обработку получаемых из дерева системы данных, их преобразование, архивирование и представление оператору в нужном виде.

Рекомендуется строить иерархический проект: от крупных элементов – к мелким (например – Завод – Цех – Печь – Клапан – Состояние). Это позволяет использовать объектный подход к построению проекта дающий разработчику такие инструменты как тиражирование, наследование, шаблоны. При этом каждый объект может иметь свои настройки и документы (мнемосхемы, окна, тренды, отчеты), собственные параметры архивирования и набор сообщений. Использование объектного подхода к разработке позволяет быстро создавать крупные проекты, с удобной для оператора системой навигации, а также быстро корректировать их.

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 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.

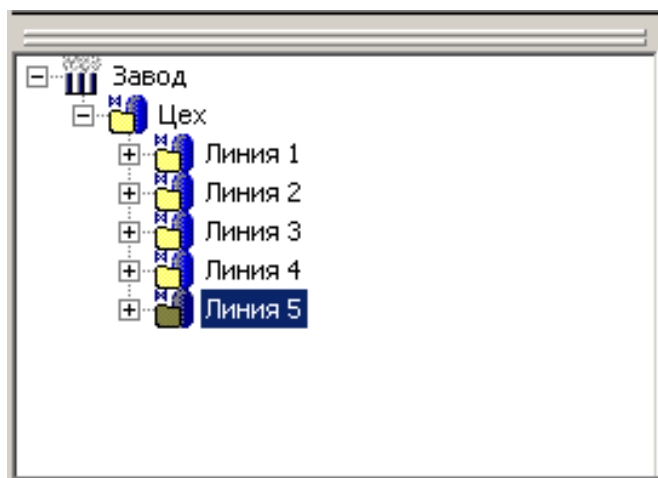
Например, мы автоматизируем на заводе цех по производству продукта X. Посмотрим, какая иерархия объектов будет в этом случае. Есть главный объект – это «Завод»; заходим на его территорию и видим наш объект автоматизации – «Цех». У этого объекта должно быть свое графическое окно – мнемосхема, куда будут 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 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 рисунке 2.3.



**Рисунок 2.3. Дерево объектов**

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

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

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 2.4.) на странице свойств Формула, а у переменной Расчет в дереве объектов в режиме исполнения показывается результат.

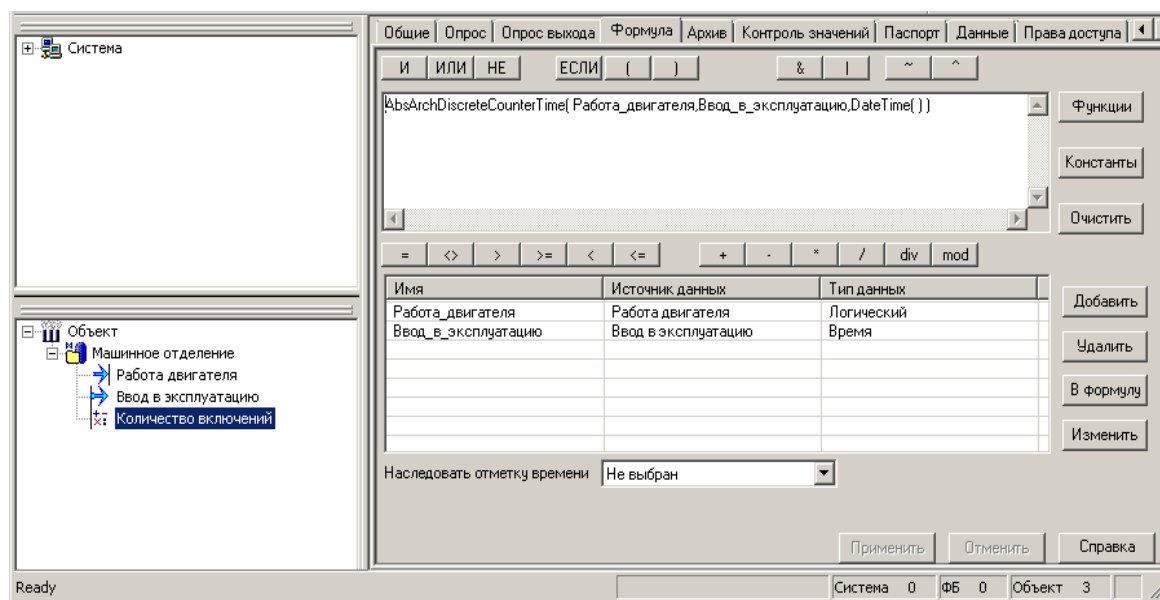


Рисунок 2.4. Страница свойств «Формула»

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 поле Обратная связь OPC переменную или выход ФБ Повторитель. В этом случае контрол 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 дерева объектов будут передаваться в дерево системы и далее – на нижний уровень, к устройствам.

### 2.3. Страница свойств элементов палитры и добавление их в мнемосхему

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.

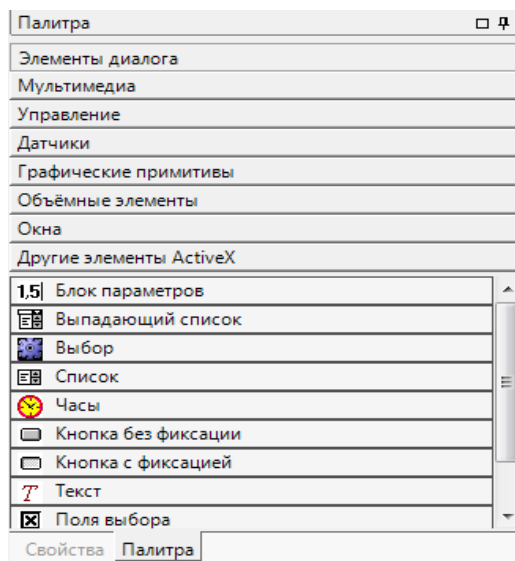
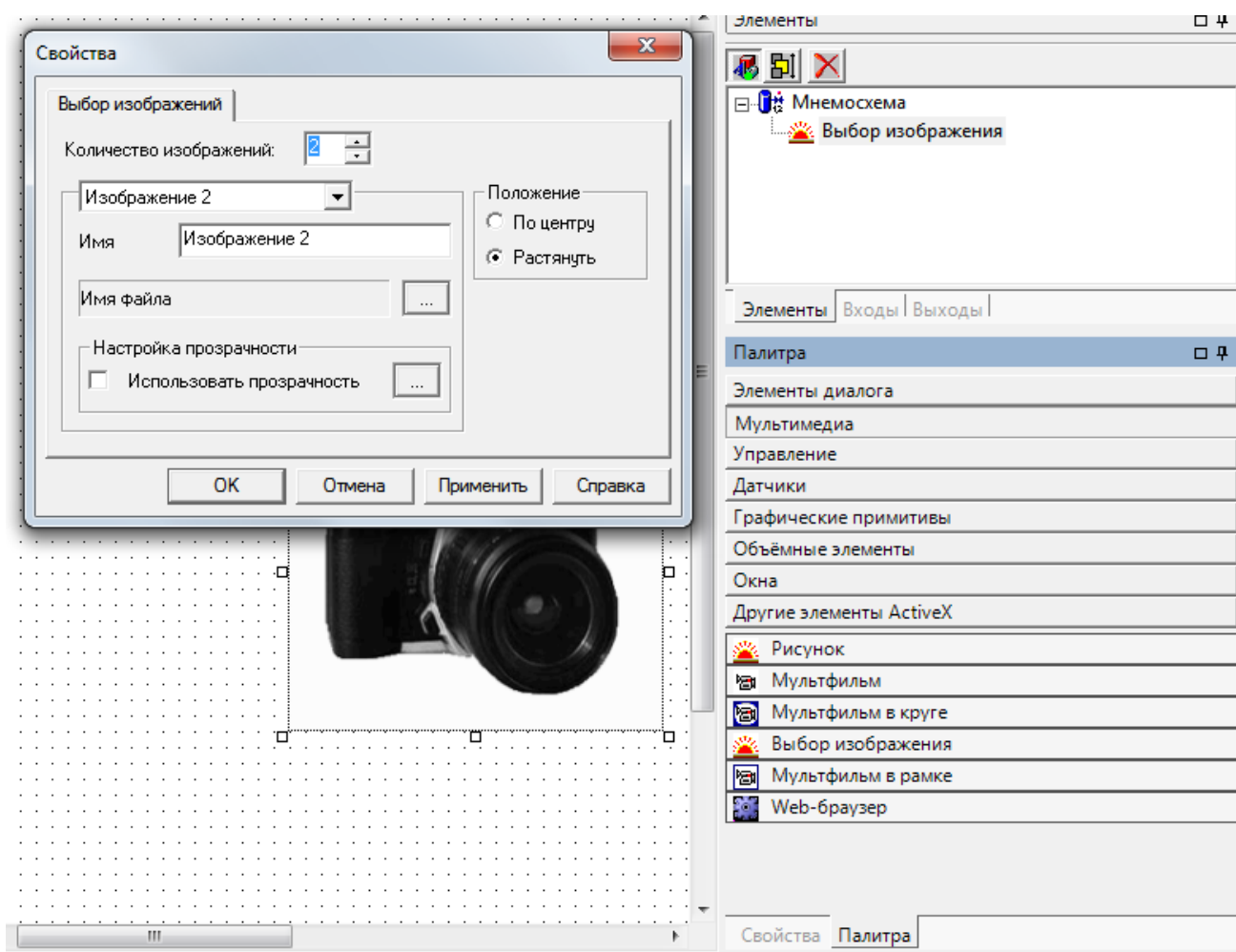


Рисунок 2.5. Палитра мнемосхемы

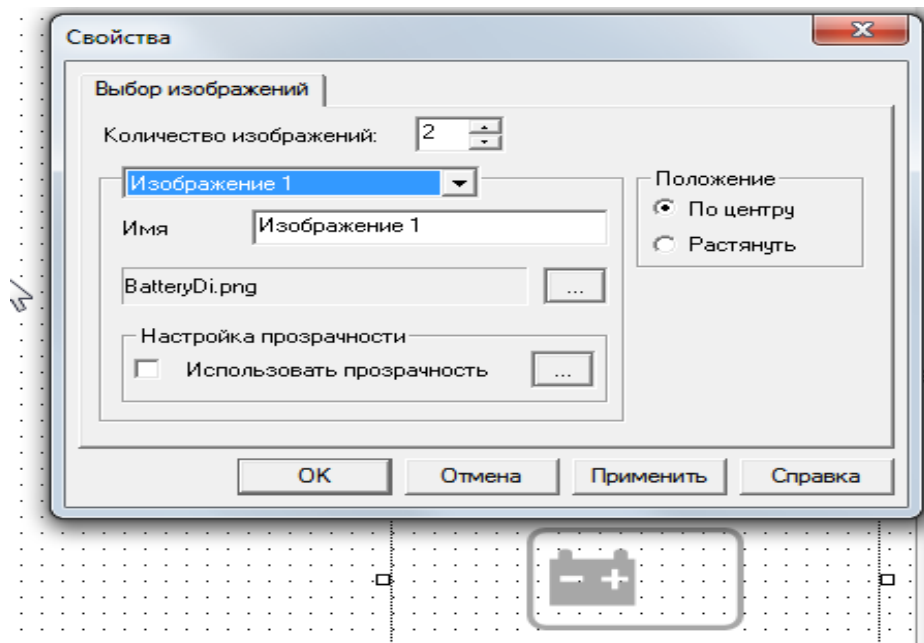
Для этого выберем соответствующий раздел. Для примера добавим элемент Выбор изображения из раздела Мультимедиа. Данный контрол предназначен для отображения рисунков на мнемосхеме, при этом номер отображаемого в текущий момент рисунка можно быстро менять. Так с помощью данного контрола можно отображать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

Для задания количества рисунков и конкретных файлов откроем окно свойств через контекстное меню контрола (Рисунок 2.6.).



**Рисунок 2.6. Окно настройки «Выбор изображения»**

Зададим два рисунка и выберем нужные. После нажатия кнопки Применить первый в списке рисунок будет отображаться в контроле (Рисунок 2.7.).

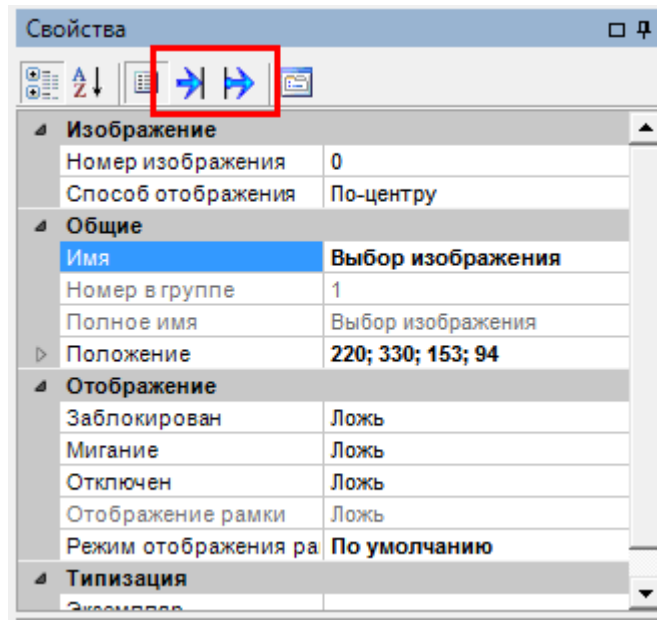


**Рисунок 2.7. Назначение рисунков**

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 контрола и элементами дерева объектов. Список доступных входов и выходов динамизации элемента можно увидеть, нажав специальные кнопки в верхней части его панели свойств (Рисунок 2.8.).

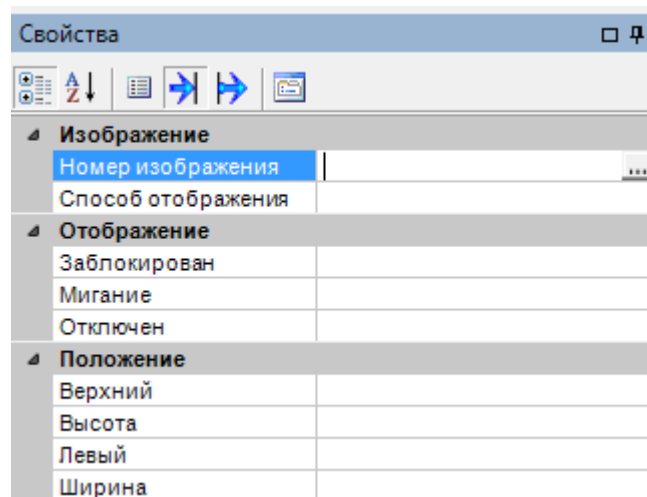
Рассмотрим управление номером рисунка в контроле Выбор изображения через вход динамизации. Пусть в нашем случае, если событие в дереве имеет значение Выкл, то номер рисунка должен быть 0 (нумерация в контроле идет с нуля), а если Вкл – то 1.

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



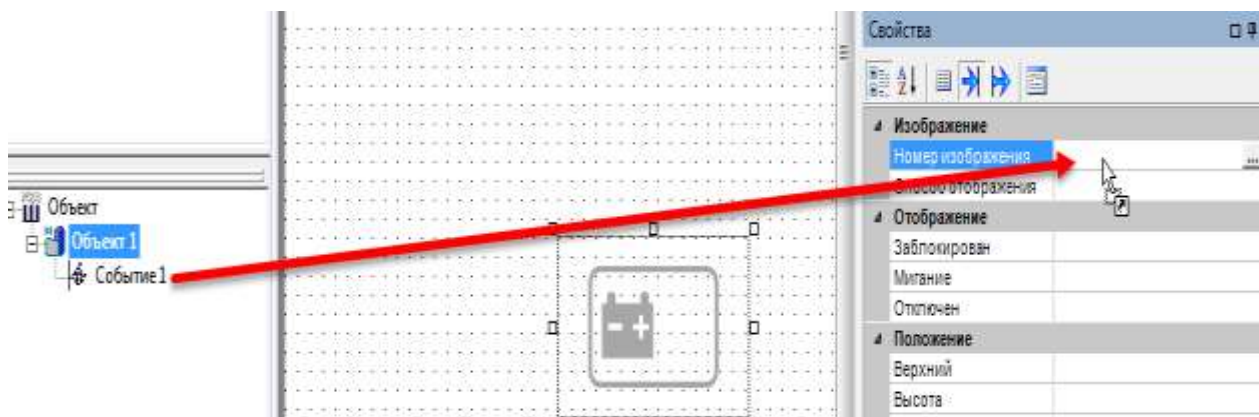
**Рисунок 2.8. Кнопки входов и выходов динамизации**

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



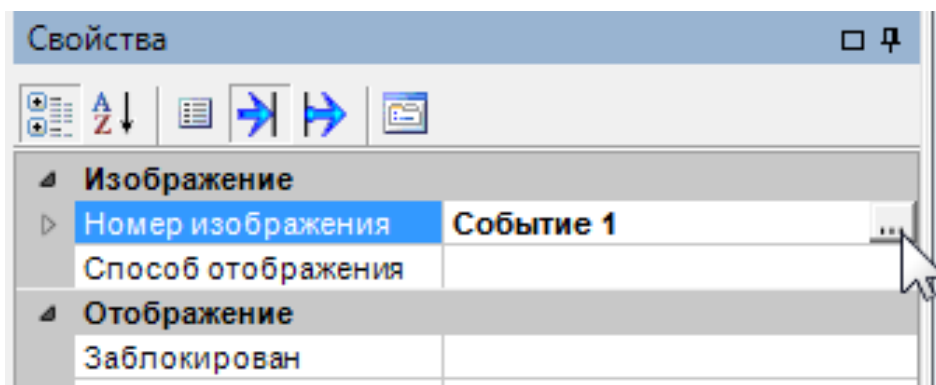
**Рисунок 2.9. Вход динамизации для управления номером изображения**

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



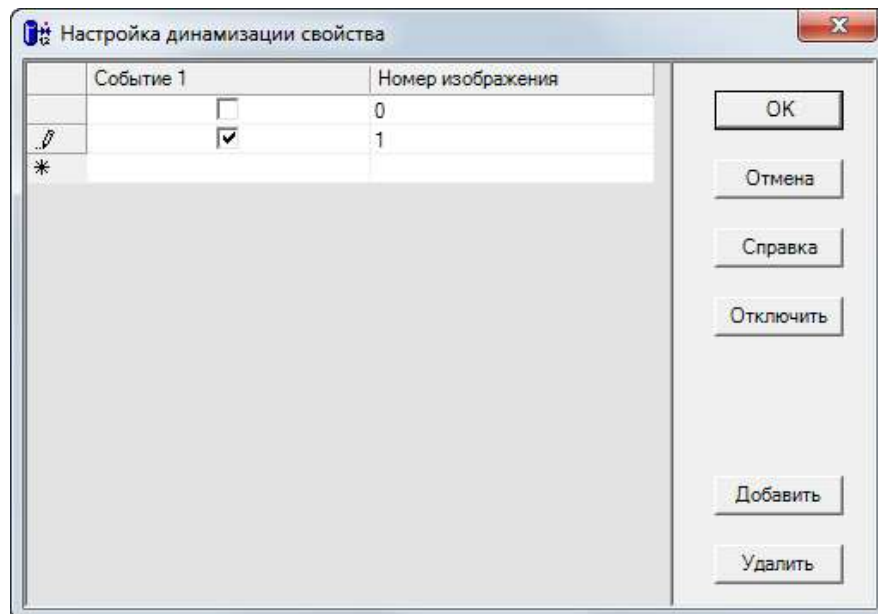
**Рисунок 2.10. Привязка входа динамизации к событию**

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



**Рисунок 2.11. Открытие окна настройки динамизации**

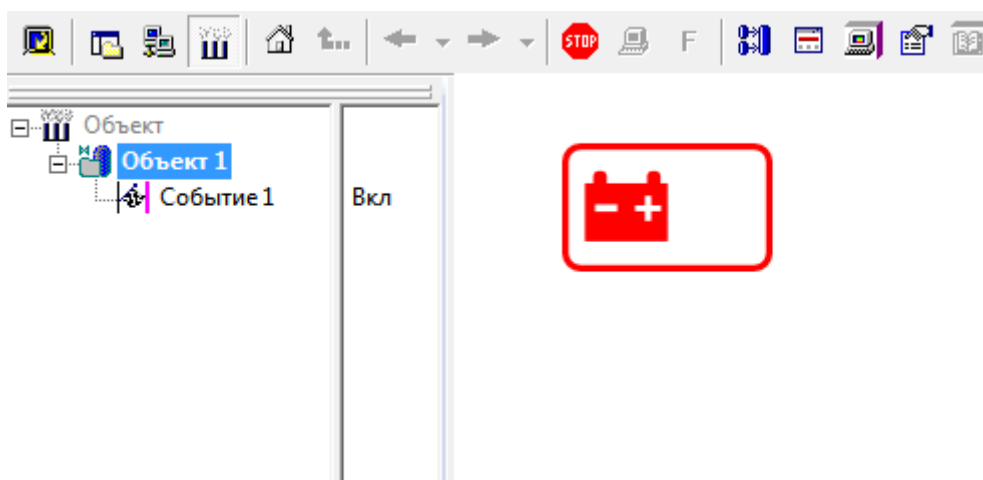
В окне динамизации свойства можно задать соответствие входным значениям из дерева объектов и значениям контрола (Рисунок 2.12.). В нашем случае, если входное значение – Выкл, то ему соответствует значение 0, если Вкл– то 1.



**Рисунок 2-12. Окно настройки динамизации**

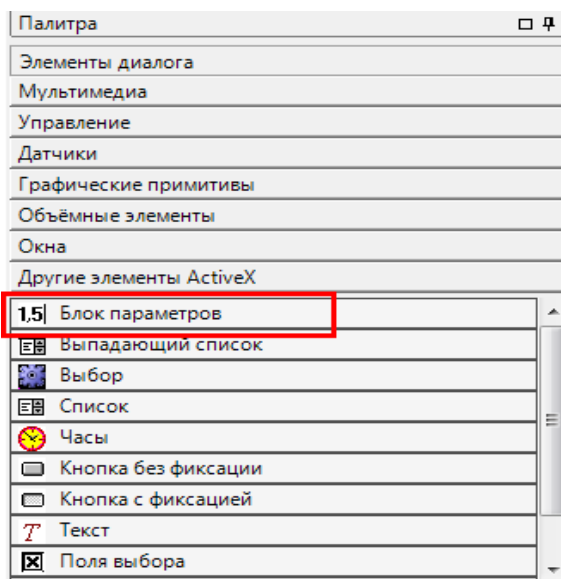
Настройка может быть и более сложной – это полностью зависит от ваших значений и выбранного вами контрола.

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



**Рисунок 2.13. Изменение рисунка в режиме исполнения**

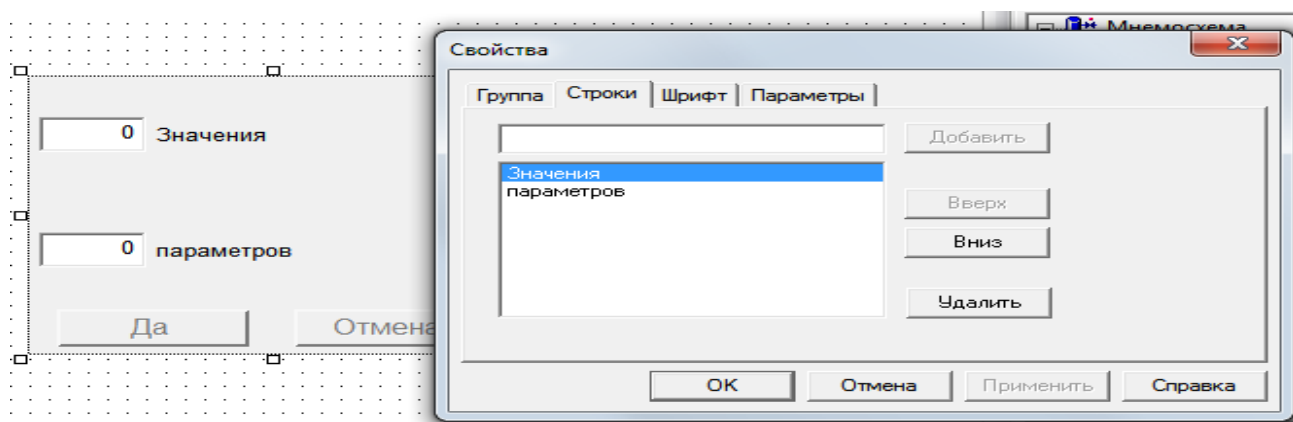
Рассмотрим еще один контрол, в котором будут использоваться как входы, так и выходы деноминации. Для примера рассмотрим элемент Блок параметров из раздела Элементы диалога.



**Рисунок 2.14. Добавление «Блока параметров»**

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

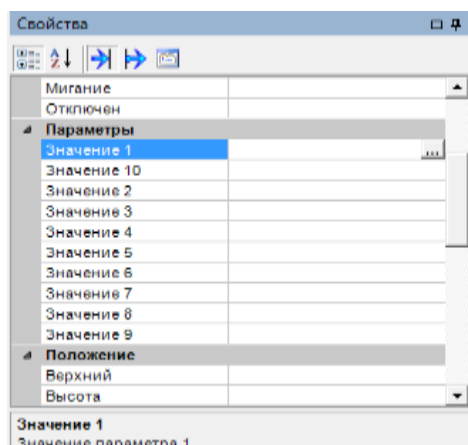
Добавление полей производится на закладке Строки в окне свойств, которое можно вызвать через контекстное меню.



**Рисунок 2.15. Настройка контрола «Блок параметров»**

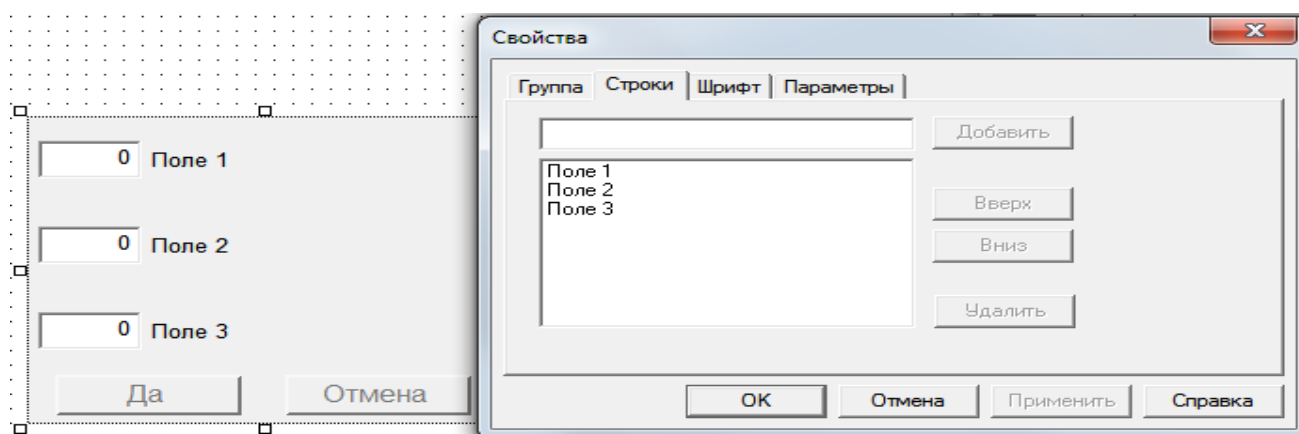
Рассмотрим процедуру связывания данного контрола с деревом объекта.

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



**Рисунок 2.16. Основные входы динамизации «Блока параметров»**

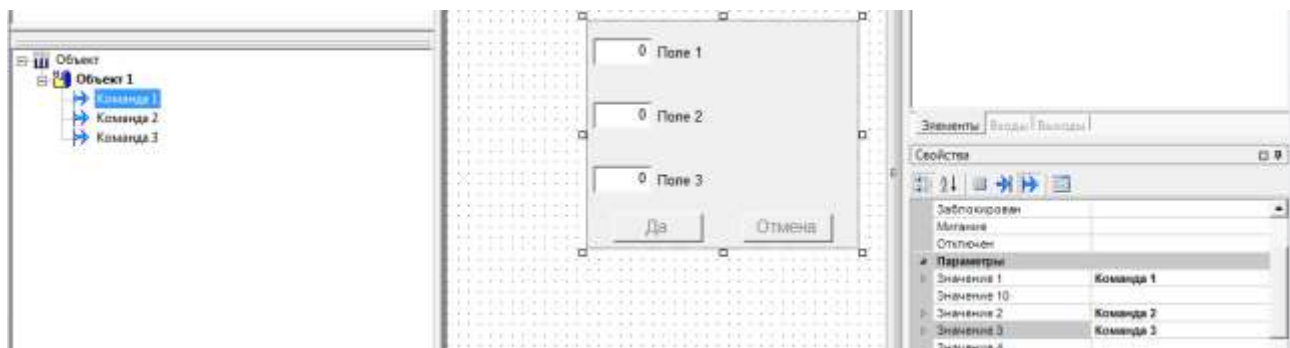
Наиболее удобный вариант связывания данного контрола – привязать вход и выход динамизации Значение N к одной и той же команде. Прделаем данные действия. Настроим наш контрол на ввод трех значений – добавим три поля ввода в контрол.



**Рисунок 2.17. Настройка контрола «Блок параметров»**

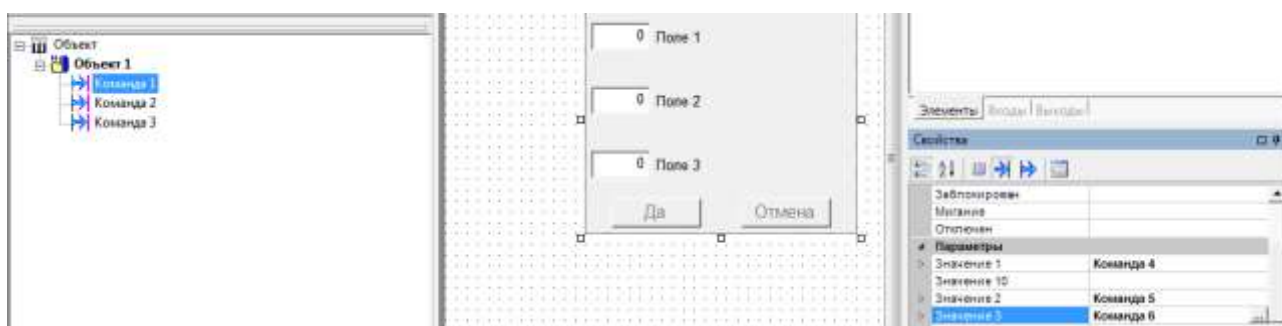
Добавим в дерево объектов три команды типа вещественный. Перетащим каждую команду в соответствующий выход динамизации Значение.





**Рисунок 2.18. Связывание выходов динамизации и команд**

Теперь перетащим эти же команды на входы динамизации Значение.



**Рисунок 2.19. 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 контуру палитры мнемосхемы есть подробное описание в справочной системе. Также на прилагаемом диске находится отдельная документация по динамизации графических контролеров в MasterSCADA.

Трехмерные объекты имеют как общие, так и специфические для конкретного типа объекта свойства. К общим свойствам 3D объектов можно получить доступ, открыв диалог Свойства и выбрав одну из закладок: Материал, Заливка, Окончания и Объекты. Аналогичные группы свойств есть и на палитре настройки.

Помимо общих свойств, у каждого типа объекта есть дополнительные, присущие только ему настройки (специфические свойства). Настройка этих свойств производится на закладке **Параметры** диалога **Свойства**.

Свойство Заливка служит для изменения параметров заливки материала (уровня жидкости в условном сосуде, образуемом внешним контуром объекта). Заливка определяется двумя настройками: цветом и процентом заливки (отношение высоты цвета заливки к общей высоте объекта).

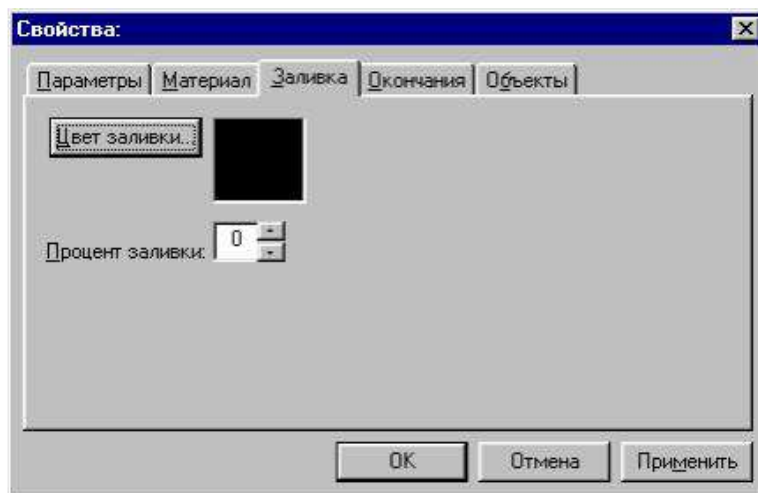


Рисунок 2.20 параметры свойств

Изменение цвета заливки производится в стандартном для **Windows** диалоговом окне Цвет, открываемое двойным щелчком на области цвета или нажатием на кнопку Цвет заливки. В этом диалоге можно выбрать цвет из основной палитры или создать новый цвет.

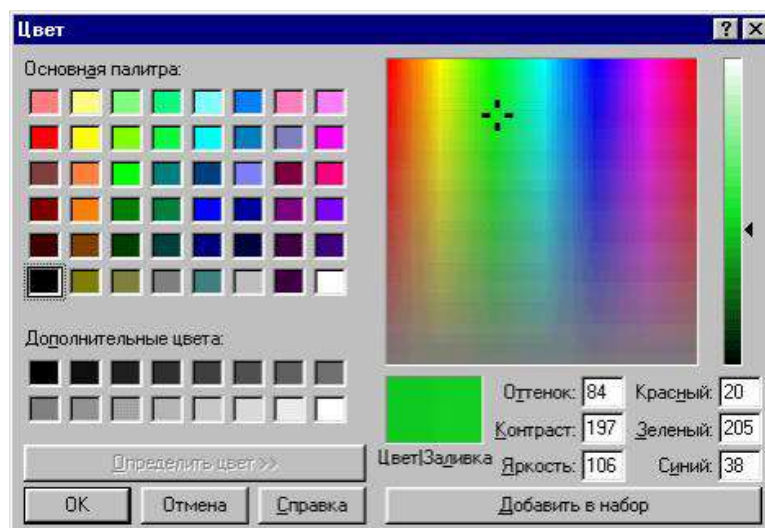


Рисунок 2.21 параметры палитры

Создать новый цвет можно щелкнув в большой правый квадрат. В месте щелчка появится перекрестие и область **Цвет/Заливка** окрасится в выбранный цвет. Оттенок цвета можно изменить при помощи ползунка, расположенного справа от большого квадрата. Определить цвет можно также вводом в поля Оттенок, Контраст, Яркость чисел от 0 до 240 и в поля Красный,

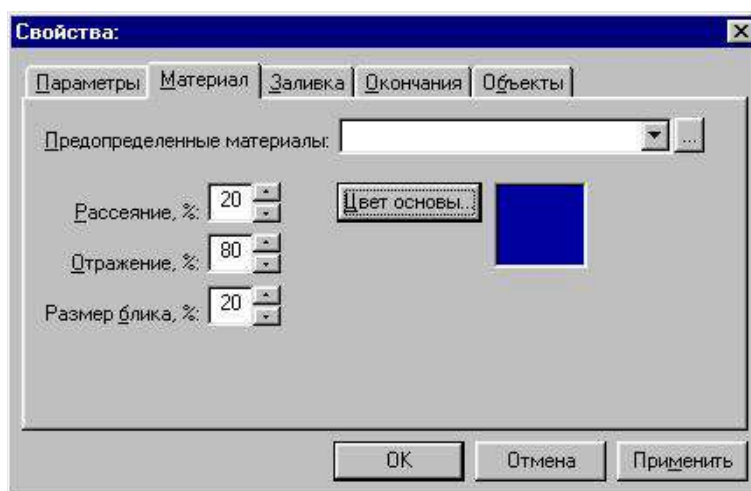
Зеленый, Синий чисел от **0** до **255** . Кнопка **Добавить** в набор добавляет созданный цвет в дополнительные цвета основной палитры.

Процент заливки изначально равен 0, т.е. при первоначальной отрисовке заливки нет. Как правило, степень заливки задается динамически в режиме исполнения в соответствии со значением параметра. Для создания заливки введите значение от 1 до 100 в поле **Процент заливки** и нажмите кнопку **ОК**.

На закладке **Материал** производится настройка свойств цветового покрытия 3D элементов. Материал характеризуется следующими четырьмя параметрами:

- Цветом поверхности;
- Степенью рассеяния (коэффициент отражения рассеянного белого света) - чем больше рассеяние, тем ярче становится элемент;
- Степенью отражения (коэффициент отражения направленного света, направление и цвет источника задаются в свойствах мнемосхемы) - при полном отсутствии отражения элемент имеет ровную закраску;
- Размером блика (размер светового пятна на поверхности элемента).

Эти значения задаются в процентах. Цвет и расположение блика зависят от общих настроек мнемосхемы (цвета и направления луча освещения).



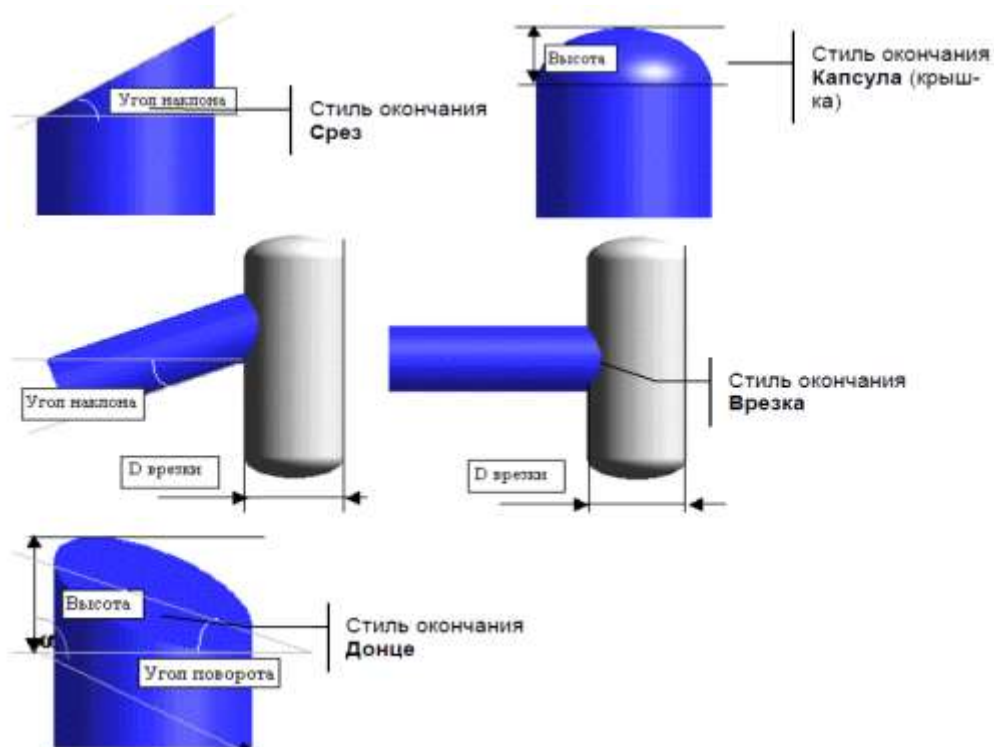
**Рисунок 2.22 Материалы для текстуры**

Закладка **Окончания** позволяет изменять начальные и конечные окончания объекта. Для 3D объектов можно использовать 4 стиля окончаний (стили

присущи не всем объектам: например, у сегмента поворота нет донца, а у конуса – среза), показанные ниже на примере цилиндра.

Параметры стилей окончаний могут принимать значения из определенного диапазона:

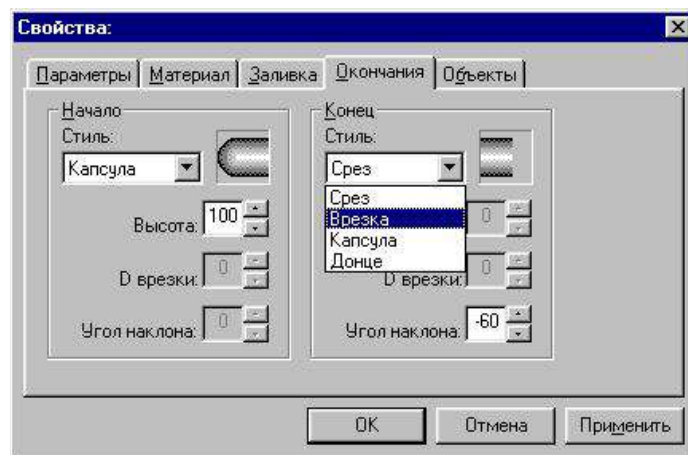
Параметры окончаний	Диапазон значений
Высота	Для капсулы: от 1 (эквивалентно срезу) до 100 (полусфера) Для донца: от -100 (наклон на нас) до 100 (наклон в обратную сторону).
D (диаметр) врезки	Измеряется в пикселях: от 1 до 999.
Угол наклона	От - 60 до 60



**Рисунок 2.23** Параметры изменение 3D объектов в Master SCADA

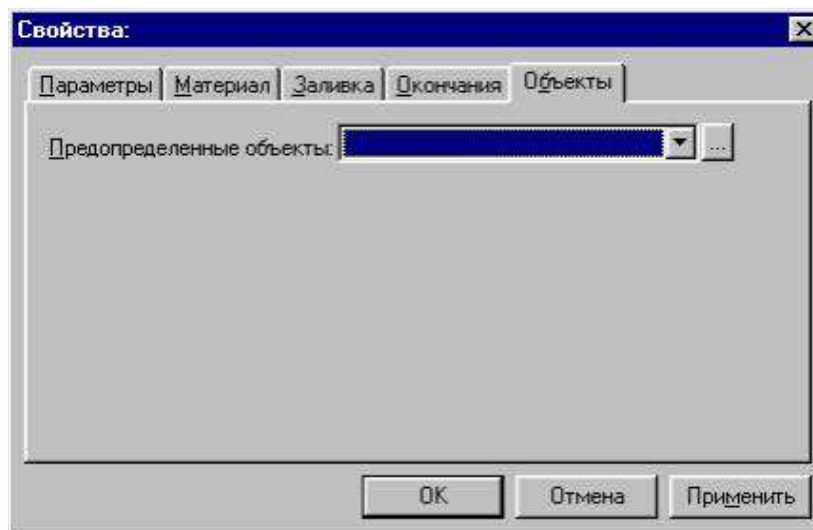
Хотя наиболее удобным и быстрым способом задания этих параметров является визуальное проектирование, для более точной настройки используйте диалог Свойства.

В диалоге Свойства выберите закладку Окончания. На ней расположены две группы полей настройки окончаний объекта Начало и Конец. Стил окончания выбирается в выпадающем списке Стил, а настройки задаются в полях Высота, D врезки и Угол наклона.



**Рисунок 2.24 Панель изменение свойств 3D объектов в Master SCADA.**

Все настройки объекта (его материал, длину, высоту, заливку, окончания и др.) можно сохранять в файле коллекции объектов, а в будущем использовать для создания нового объекта с теми же параметрами. Для совершения операций сохранения и чтения файлов коллекций объектов используется закладка Объекты. Сохранение и чтение коллекций объектов (предопределенных объектов) производится аналогично предопределенным материалам, описание которых указано в следующем разделе.



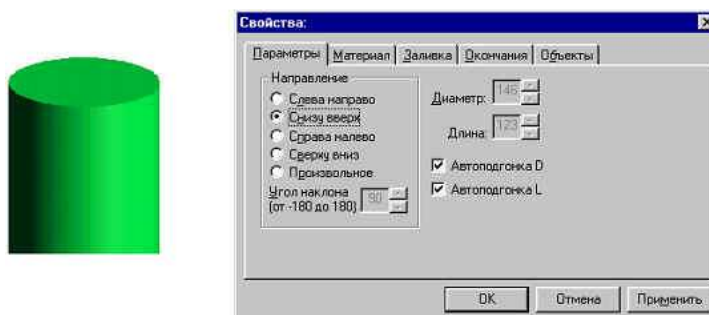
**Рисунок 2.25 Панель изменение свойств объектов в Master SCADA.**

Объект Цилиндр используется для создания несложных объектов, таких как цистерны, боч-ки, танки, ректификационные колонны, отрезки трубопроводов и т.д.



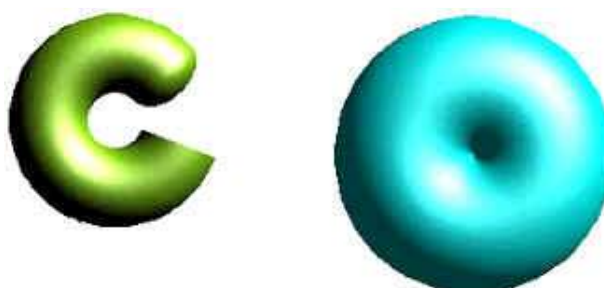
**Рисунок 2.26 Примеры объёмных объектов в Master SCADA .**

На рисунке ниже показан вид Цилиндра при значениях настроек, заданных на закладке Параметры:



**Рисунок 2.27. вид Цилиндра при значениях настроек, заданных на закладке Параметры**

Объект Сегмент поворота используется для углов поворота трубопроводов, втулок, колец и т.д.



**Рисунок 2.28 Сегмент поворота объёмных объектов в Master SCADA .**

На рисунке ниже показан вид Сегмента поворота при значениях настроек, заданных на закладке Параметры:





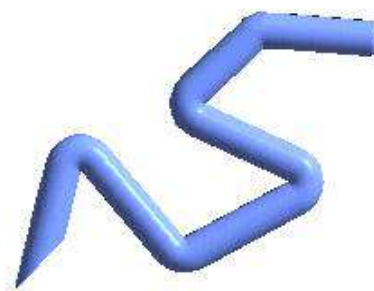
**Рисунок 2.29** окно параметров 3D объекта .

Поля настройки закладки *Параметры Сегмента поворота*:

Свойство	Описание
Диаметр	Диаметр <i>Сегмента поворота</i> .
Начальный угол	Угол между горизонтальной плоскостью и начальным окончанием (-180°; 180°).
Конечный угол	Угол между горизонтальной плоскостью и конечным окончанием (-180°; 180°).
Радиус поворота	Радиус дуги <i>Сегмента</i> .
Автоподгонка	Автоматическая подгонка радиуса поворота для точного вписывания элемента в размер рамки.

Для данного объекта не определяется угол наклона стилей окончания Срез и Врезка, отсутствует стиль Донце.

Объект Трубопровод позволяет легко создавать трубопроводы различной протяженности, диаметра и конфигурации.



**Рисунок 2.30** 3D трубопровод

На рисунке ниже показан вид сегмента поворота при значениях настроек, заданных на за-кладке Параметры:



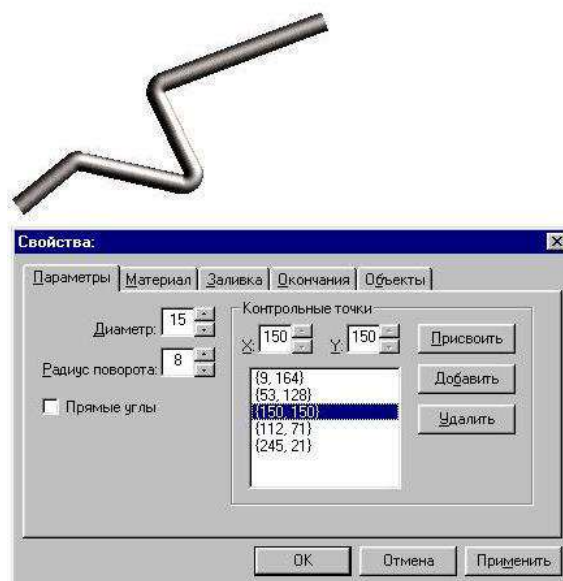


Рисунок 2.31 Панель параметров изменение 3D трубопровод

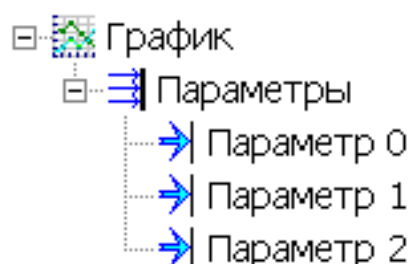
Свойство	Описание
Диаметр	Диаметр трубы <i>Трубопровода</i> .
Радиус поворота	Аналогично сегменту поворота.
Прямые углы	После установки этого признака все дополнительные отрезки <i>Трубопровода</i> будут прокладываться под прямым углом к предыдущим отрезкам.
Контрольные точки X	Координаты (X; Y) точек построения <i>Трубопровода</i> .
Контрольные точки Y	

Количество контрольных точек трубопровода неограниченно: их можно добавлять, удалять, править. Для этих целей используются кнопки: **Добавить** - добавление новой контрольной точки; **Удалить** - удаление выделенной контрольной точки; **Присвоить** - присвоение значений в полях ввода X и Y координатам выбранной в списке контрольной точки. Но гораздо удобнее добавлять, удалять, править контрольные точки в режиме визуального проектирования, как описано в тексте ниже.

Активные точки бордового цвета показывают контрольные точки. Перемещая эти контрольные точки (нажать левую клавишу мыши и, не отпуская, передвинуть курсор мыши на нужное место) оператор может изменять вид трубы (корректировать углы сгиба).

Чтобы добавить новую контрольную точку выполните двойной щелчок мышью на мнемосхеме (новая точка добавляется в конец или начало трубы, в зависимости от того куда щелкнули мышью). Двойной щелчок на самом Трубопроводе добавляет новую контрольную точку в том месте куда щелкнули (между уже существующими точками трубы).

Визуальный функциональный блок График относится к категории Датчики и имеет настраиваемое количество входов, соответствующих параметрам графика. Вид функционального блока в дереве объекта представлен на рисунке ниже.



**Рисунок 2.32 Вид функционального блока в дереве объекта**

График предназначен для просмотра значений переменных в мнемосхеме. Существует три основных способа создания элемента управления График в мнемосхеме:

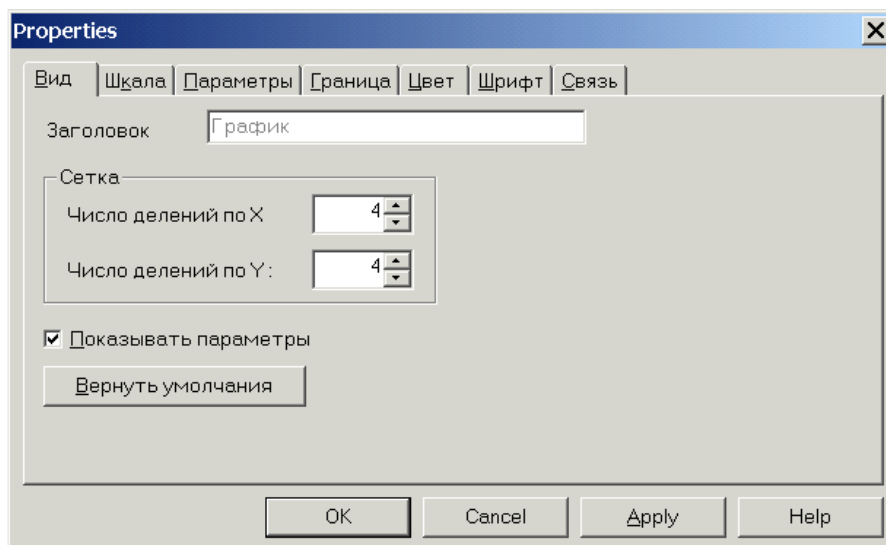
- Вставить ВФБ График из палитры функциональных блоков в объект, привязать интересующие переменные к входам Графика, и перетащить его в мнемосхему;
- Для просмотра значений аналоговых переменных в виде графика (Значение, Расчет, ОРС переменная для чтения, Выход ФБ), можно просто перетащить их в мнемосхему правой кнопкой мыши, и в открывающемся контекстном меню выбрать график;
- Создать элемент управления График непосредственно в мнемосхеме. Затем, завести необходимое количество входов мнемосхемы, связанных с переменными MasterSCADA, и динамизировать свойства графика Значение от ЭТИХ ВХОДОВ.

График реализует функции тренда реального времени (зависимость параметра от времени). Может иметь от 1 до 8 параметров. Диапазон шкалы, единица измерения и формат значения каждого параметра наследуются от настроек переменной, привязанной к параметру Графика в дереве объекта. Число параметров, отображаемых на Графике, задается в проекте на странице Свойства. По умолчанию это всегда один параметр. ВФБ



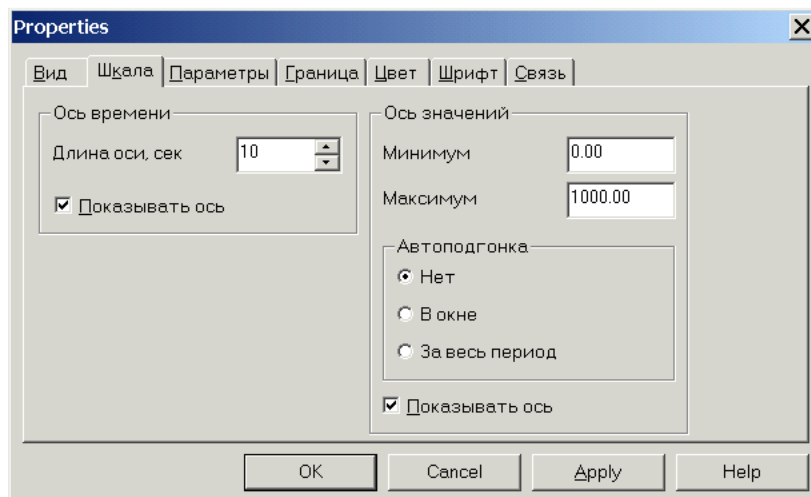
**Рисунок 2.33 Вид графика в дереве объекта**

ВФБ График имеет возможность отображать недостоверность данных входов. На закладке Вид настраивается внешний вид ВФБ График



**Рисунок 2.34 Вид графика в дереве объекта**

При отсутствии связи у выхода ВФБ или работы графического редактора MasterGraph вне среды пакета программ MasterSCADA наследование настроек свойств переменной не работает. В этом случае все свойства графика настраиваются непосредственно на страницах свойств. После установления в дереве связи выхода ВФБ для импорта всех необходимых настроек следует выполнить команду меню Сервис/Обновить связи.



**Рисунок 2.35 Вид параметров шкалы.**

## **2.4. Инструментальная среда порядка разработки проекта Master SCADA.**

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.

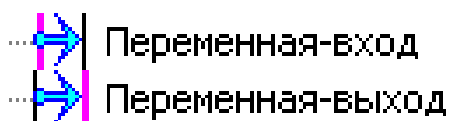
MasterSCADA позволяет разрабатывать проект в любом порядке, в том числе и параллельно. Если это делают разные разработчики на разных

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.

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.

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 левую кнопку мыши, вы увидите, что к стандартному обозначению переменной добавится розовая вертикальная черта – это говорит о том, что связь установлена (Рисунок 2.36.).



**Рисунок 2.36. 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» 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 потребуются значительно больше времени, чтобы восстановить проект. Для решения данной задачи можно использовать сервис внешних связей или ФБ Повторитель. 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 получают значение уже от нее. Тем самым замена сигнала, используемого во многих обработках или

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

## **2.5. Переменные объектов и функциональные блоки Master SCADA**

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

Функциональный блок 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 блоками. 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, 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.

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 С# (ФБ Скрипт).

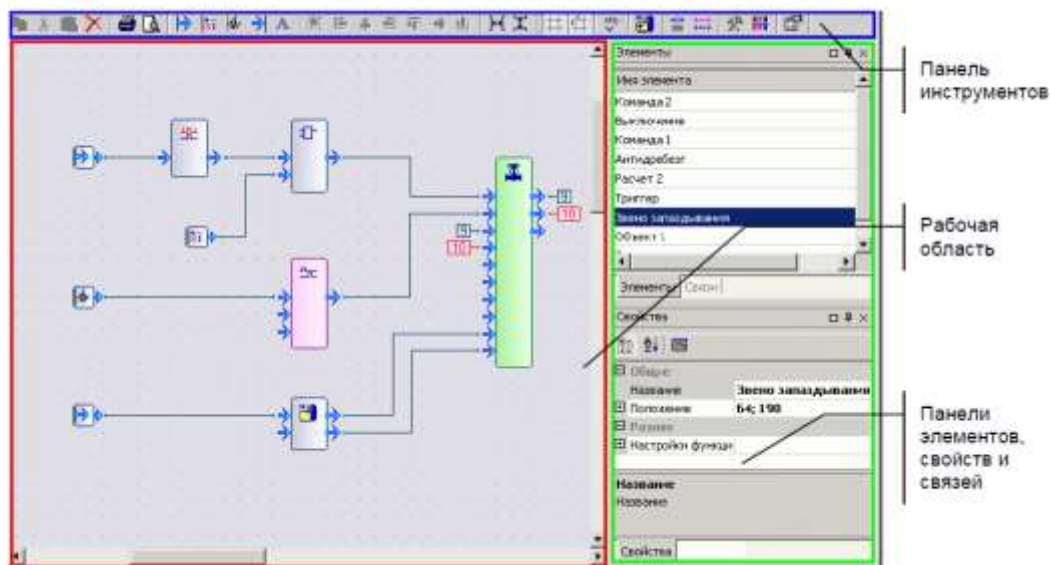
Конечно, иногда встречаются задачи, которые сложно решить стандартными способами, но даже в этих непростых ситуациях мы не оставляем наших клиентов без поддержки. Если пользователь имеет достаточный опыт в программировании и время, то мы рекомендуем создавать свои собственные функциональные блоки, основываясь 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 условиях 8.

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

#### Редактор схем функциональных блоков

Редактор предназначен для создания схем функциональных блоков. Схема функциональных блоков описывает порядок выполнения блоков и связи по передачи данных от выхода одно-го блока к входам других блоков.

Схема функциональных блоков представляет собой графическое отображение набора функциональных блоков, принадлежащих объекту. Каждому объекту может принадлежать не более одной схемы.




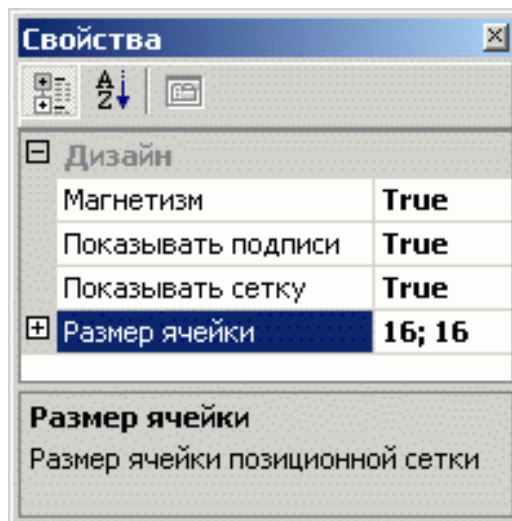
**Рисунок 2.37. Интерфейс окна изменение блок схем**

Рабочая область предназначена для отображения и редактирования схемы функциональных блоков. На схеме отображаются все переменные, функциональные блоки и объекты, входящие в состав текущего объекта.

Элементы схемы функциональных блоков:

- Функциональные блоки;
- Визуальные функциональные блоки;
- Объекты из дерева объектов Проекта, подчиненные объекту, которому принадлежит схема;
- Группы переменных;


Свойства схемы отображаются на панели Свойства. Существует два режима настройки: настройка основных свойств и настройка цветов схемы. Для редактирования основных свойств, необходимо щелкнуть мышью по иконке  панели инструментов, либо, щелкнув мышью по пустому месту схемы, перейти на панель Свойства

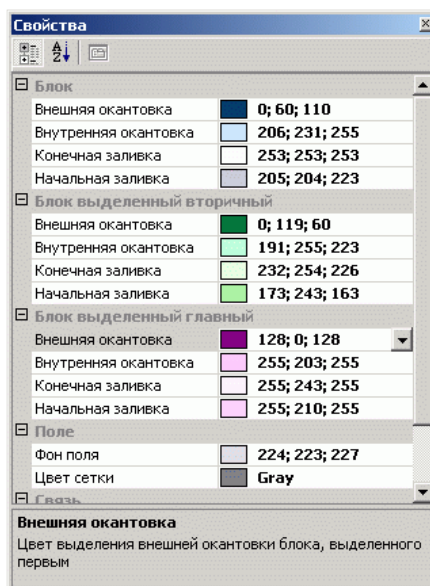


**Рисунок 2.38. Панель блока свойств**

Свойства схемы:


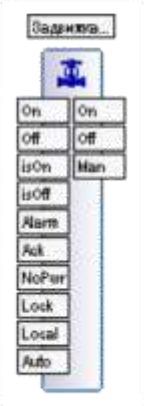
- Магнетизм - выравнивает элементы по узлам сетки;
- Показывать подписи - показывает подписи имен элементов схемы, названия входов и выходов функциональных блоков;
- Показывать сетку - включает отображение сетки на схеме;
- Размер ячейки - устанавливает высоту и ширину ячейки сетки.

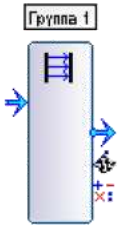




Для открытия панели Свойства в режиме настройки цветов схемы, необходимо щелкнуть мышью по иконке .



**Рисунок 2.39. Панель цветокоррекции свойств объекта.**

## Отображение элементов на схеме

Наименование	Внешний вид	Описание
Объекты		Изображения дочерних объектов. Переменные Проекта, входящие в состав дочернего объекта, отображаются в виде входов и выходов объекта (Входы: Значения; Выходы: Команды, Расчеты, События).
Функциональные блоки		В режиме отображения подписей функциональные блоки вместо изображения входов/выходов выводят их короткие имена.

Наименование	Внешний вид	Описание
Группы переменных		
Команда		присутствует только в MSRT (исполнительная система MasterSCADA)
Расчёт		
Событие		присутствует только в MSRT (исполнительная система MasterSCADA)
Значение		присутствует только в MSRT (исполнительная система MasterSCADA)

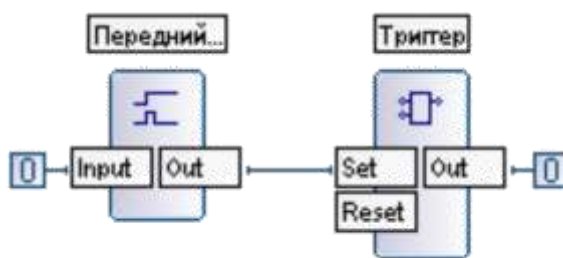
## Отображение связей

Существует три типа связи: связь с элементом дочернего объекта, внутренняя связь и внешняя.

Внутренняя связь осуществляется между элементами, которые находятся внутри текуще-го объекта на первом уровне вложенности. На схеме отображаются одним из следующих способов:

- Линиями, если выход находится левее входа;

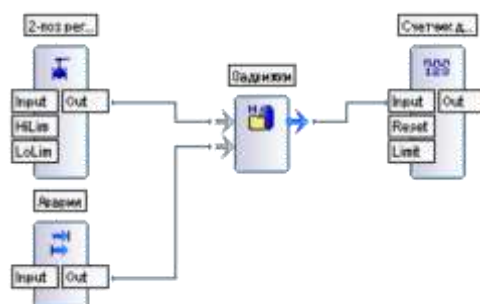
- Цифрами (маркерами), если выход находится правее входа.



**Рисунок 2.39. Функциональная блок схема.**

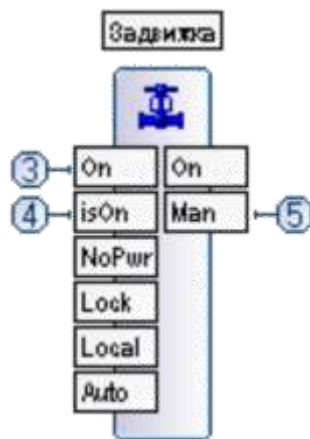
Связь с элементом дочернего объекта осуществляется между элементом текущего объекта и элементом дочернего объекта, и выглядит на схеме аналогично внутренней связи. При этом, дочерний объект отображается в виде блока, имеющего входы и выходы, которые представляют входы и выходы вложенных в него элементов. Входы/выходы дочернего объекта могут отображаться одним из двух способов:

- Серым цветом - если связь установлена со входом/выходом функционального блока, входящего в состав дочернего объекта (или любого другого объекта, принадлежащего дочернему);
- Синим цветом - если связь установлена с любой из переменных, принадлежащих непосредственно дочернему объекту.



**Рисунок 2.40. Функциональная блок схема с подключением объекта управление.**

Внешняя связь осуществляется между элементом текущего объекта и элементом, не входящим ни в данный объект, ни в любой из дочерних. В этом случае связь отображается цифрой, помещенной в круг.



**Рисунок 2.40. Функциональный блок исполнительного механизма.**

## 2.6. 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 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):

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



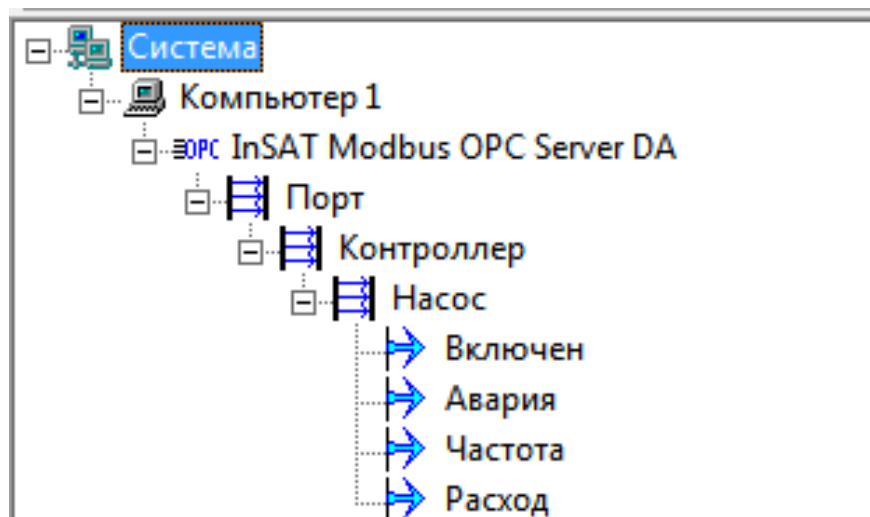
Таким образом, четко видна объектная структура будущего проекта. Можно создать отдельные объекты котлов, насосов, теплообменника, а затем собирать из них будущие проекты. При этом если в одном проекте котлов всего 2 и нет теплообменника, а в другом котлов уже 4 и использованы два теплообменника, то построение 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.

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

- Состояние насоса: включен или выключен.
- Состояние аварии насоса.
- Выходная частота с частотного преобразователя.
- 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)



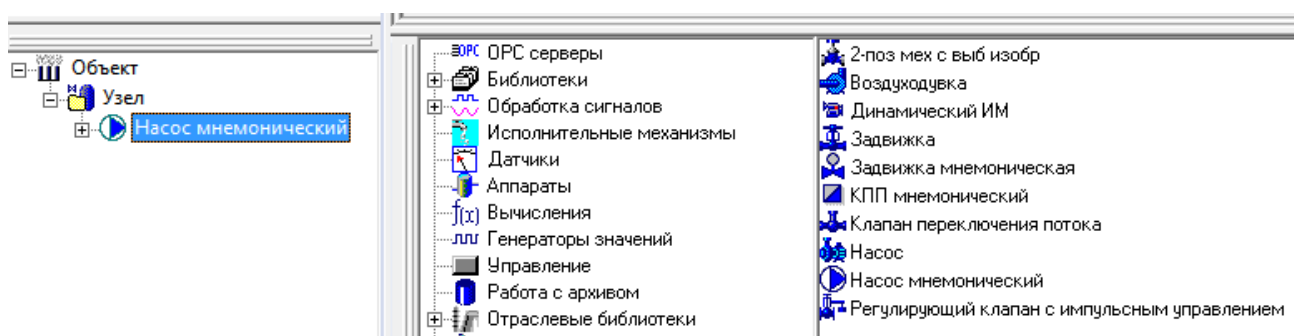
**Рисунок 2.41. Дерево системы с OPC-сервером**

В дереве объектов, через контекстное меню объекта, создадим объект Узел (Рисунок 2.42.) – в нем мы будем размещать элементы узла.



**Рисунок 2.42. 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 2.43.).

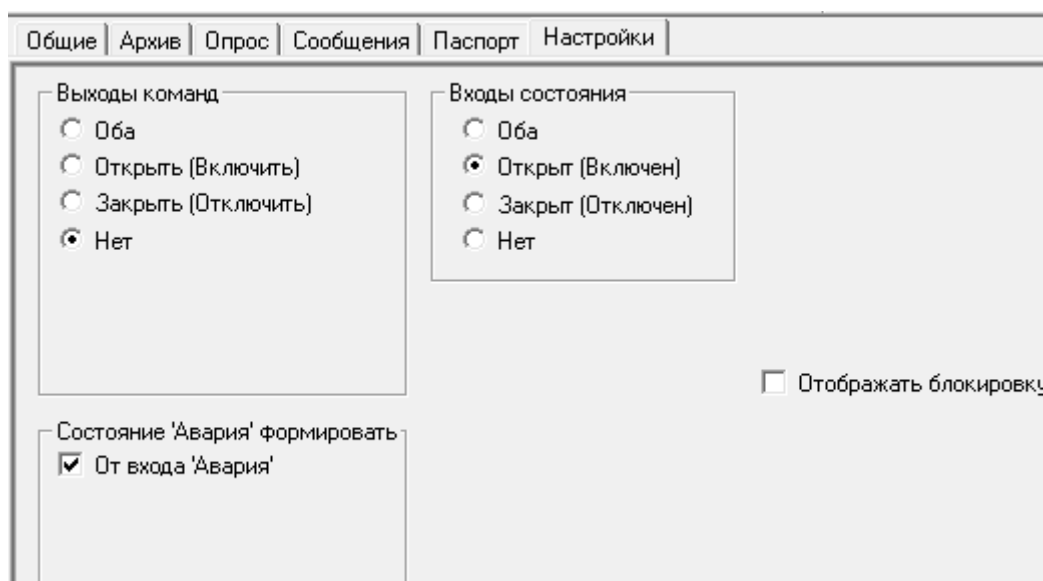


**Рисунок 2.43. Исполнительные механизмы ФБ Насос мнемонический**

Можно переименовать данный ФБ – назовем его просто Насос.

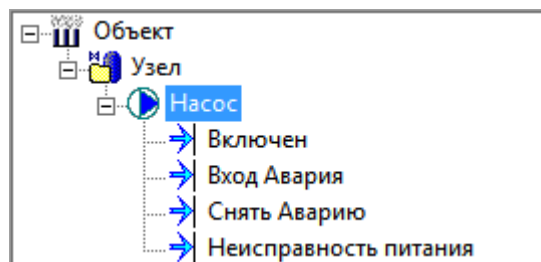
Настроим данный Насос. Мы будем использовать его только для отображения состояния насоса и аварии, поэтому на вкладке Настройки

выключим Выходы команд, а Входы состояния установим в режим Открыт (Рисунок 2.44.).



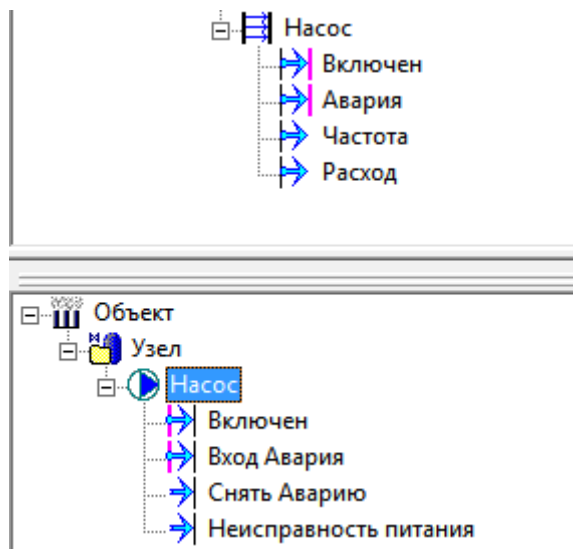
**Рисунок 2.44. Настройки ФБ «Насос»**

После нажатия кнопки Применить у ФБ останется набор входов, как на Рисунке 2.45



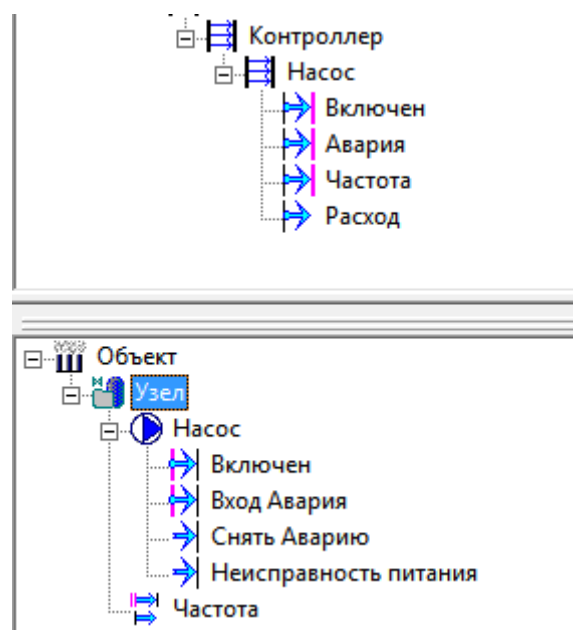
**Рисунок 2.45. Входы ФБ «Насос»**

Установим связи – перетащим ОРС-переменные Включен и Насос на эквивалентные им входы в функциональном блоке (Рисунок 2.46.). Переменные будут иметь красный индикатор – это означает, что переменная имеет связь.



**Рисунок 2.46. Установка связей**

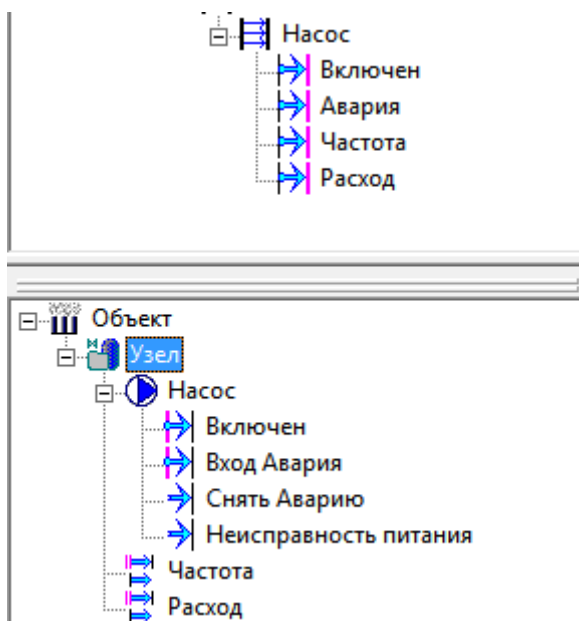
Специального функционального блока для частотного преобразователя в MasterSCADA нет. Но в данном случае он и не требуется: от преобразователя поступает только сигнал частоты, который нужно отображать на мнемосхеме. Перетащим OPC- переменную Частота в объект Узел – будет создано значение Частота (Рисунок 2.47).



**Рисунок 2.47. Создание значения частоты**

Аналогично поступим с расходом (Рисунок 2.48) – для расходомера мы разместим на

мнемосхеме небольшой рисунок, над которым будет отображаться текущее значение расхода.



**Рисунок 2.48. Создание значения расхода**

Мы наполнили объект переменными. В нашем случае в объекте нет внутренних связей (связей между переменными исключительно внутри объекта), однако в большинстве 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 2.49).

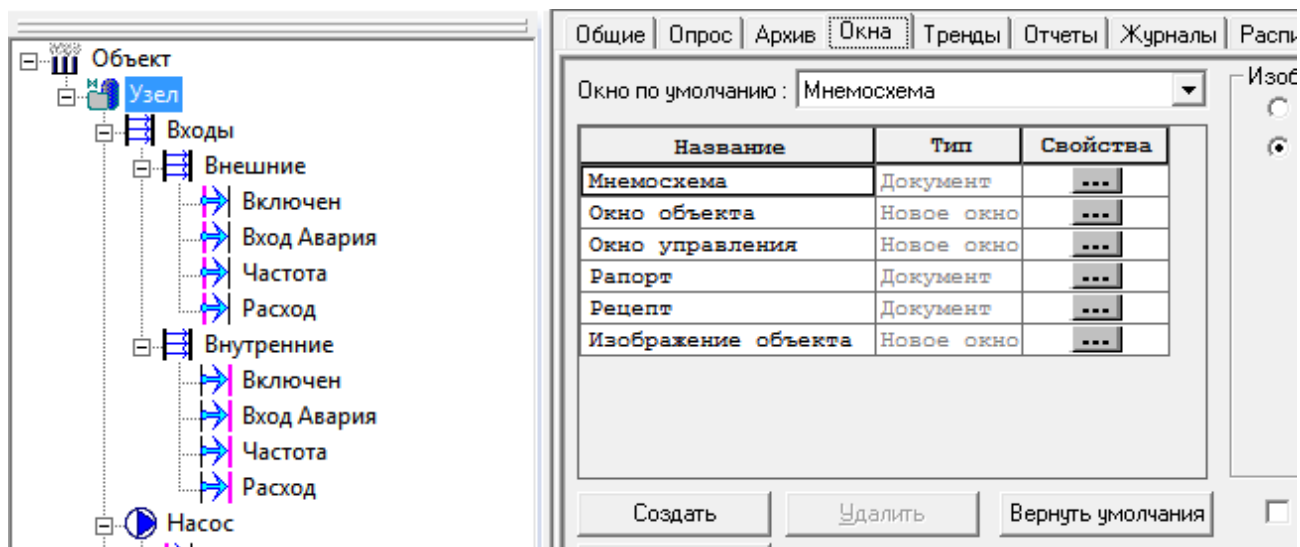


**Рисунок 2.49. Сервис «Внешние связи»**

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

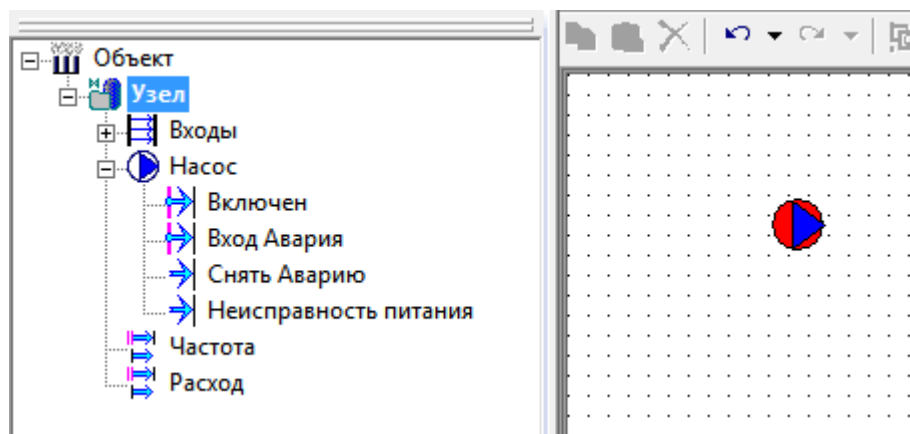
Теперь создадим атрибуты объекта – в данном случае различные документы.

Создадим у объекта мнемосхему – сделаем это на вкладке Окна (Рисунок 2.50).



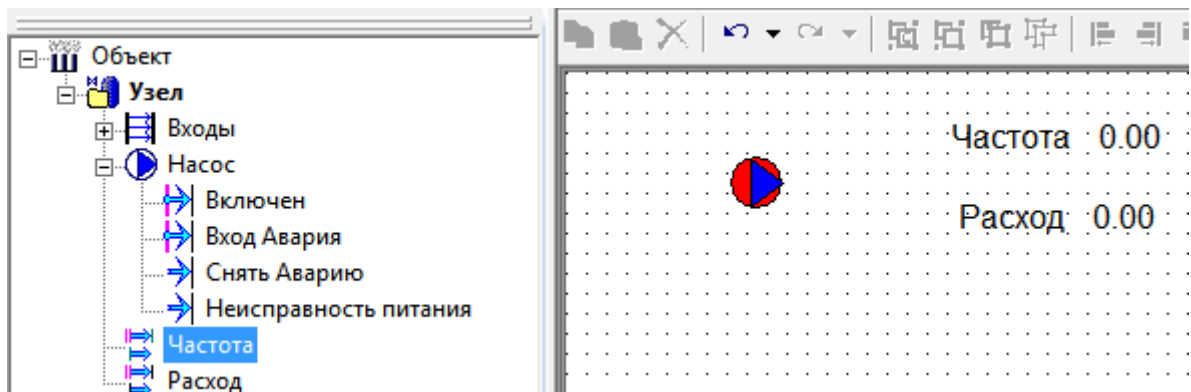
**Рисунок 2.50. Создание мнемосхемы**

На мнемосхему добавим наш Насос – вытащим его левой клавишей мыши (Рисунок 2.51).



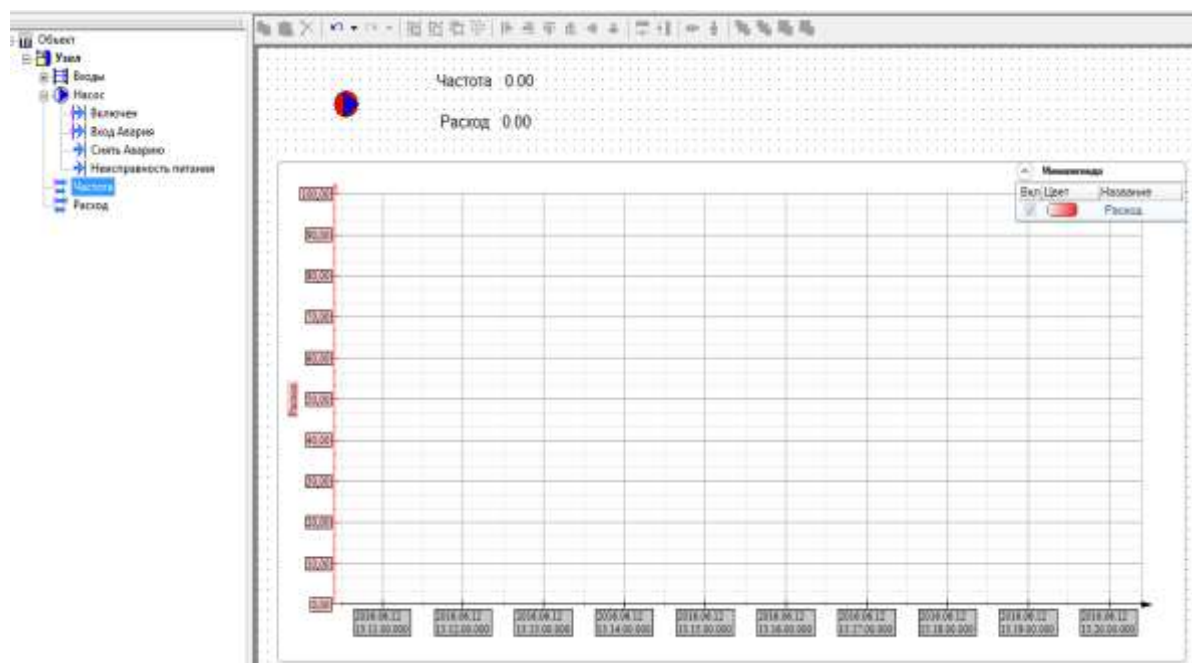
**Рисунок 2.51. Вставка контроля насоса на мнемосхему**

Аналогично вытащим Частоту и Расход (Рисунок 2.52).



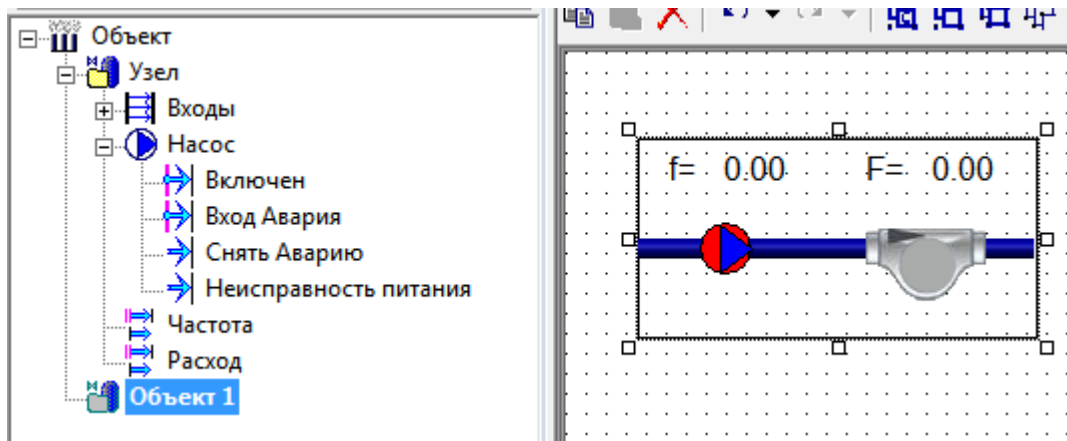
**Рисунок 2.52. Создание контролов частоты и расхода**

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 2.53.



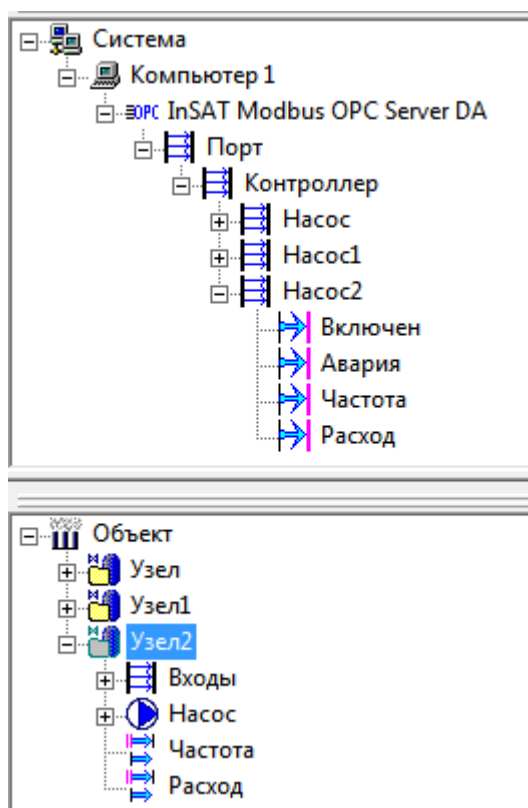
**Рисунок 2.53. Готовая мнемосхема**

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



**Рисунок 2.54. 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 2.55. Дублирование объектов**

Механизм «Шаблон-Экземпляр»

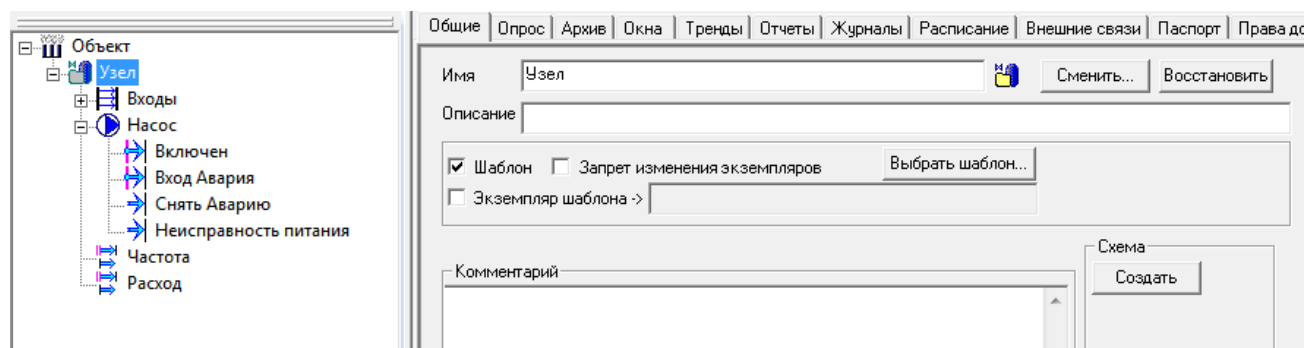


Подход, при котором создаются отдельные элементы, а затем из них комплектуется проект, очень удобен. Но есть в нем и недостаток – обязательно потребуется внести изменения в какой-либо элемент. Изменение может быть вызвано или некорректной работой элемента из-за какой-либо ошибки, или же необходимостью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

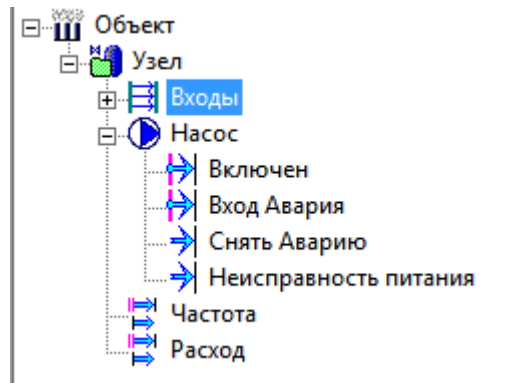
Удалим Узел1 и Узел2: мы скопировали, не включив шаблон у Узла1, поэтому вносить изменения мы не сможем.

У объекта Узел на вкладке Общие поставим галочку Шаблон.



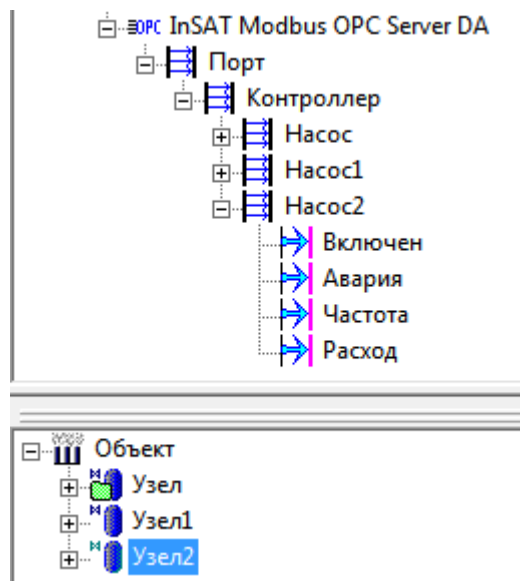
**Рисунок 2.56. Включение шаблона**

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



**Рисунок 2.57. Индикация того, что объект является шаблоном**

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



**Рисунок 2.58. Экземпляры шаблона**

Обратите внимание, что объекты Узел1 и Узел2 также поменяли свои значки – они теперь экземпляры шаблона (Рисунок 2.58).

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 2.59).

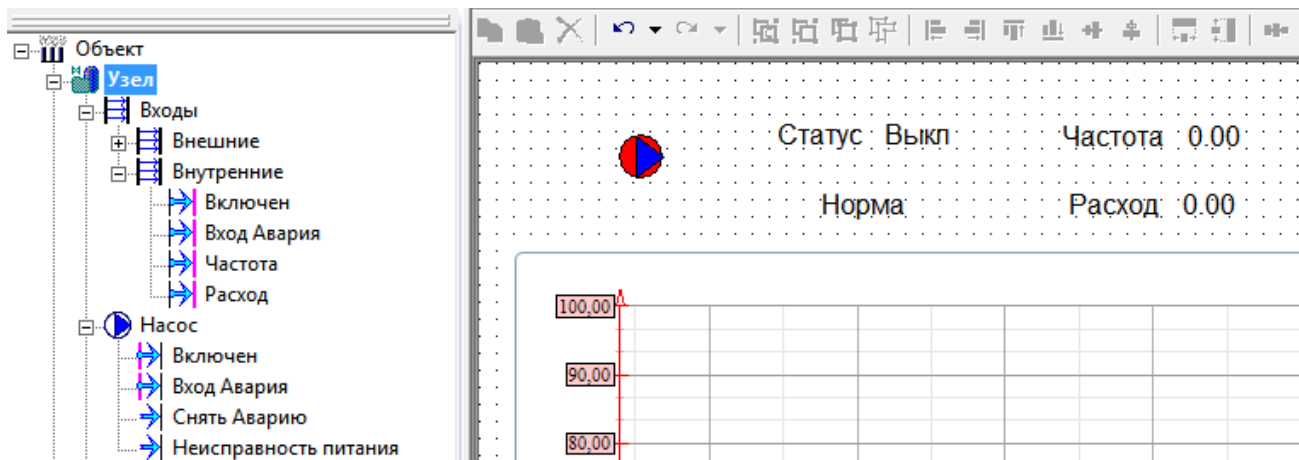


Рисунок 2.59. Внесение изменений в мнемосхему

Сохраним и закроем мнемосхему.

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

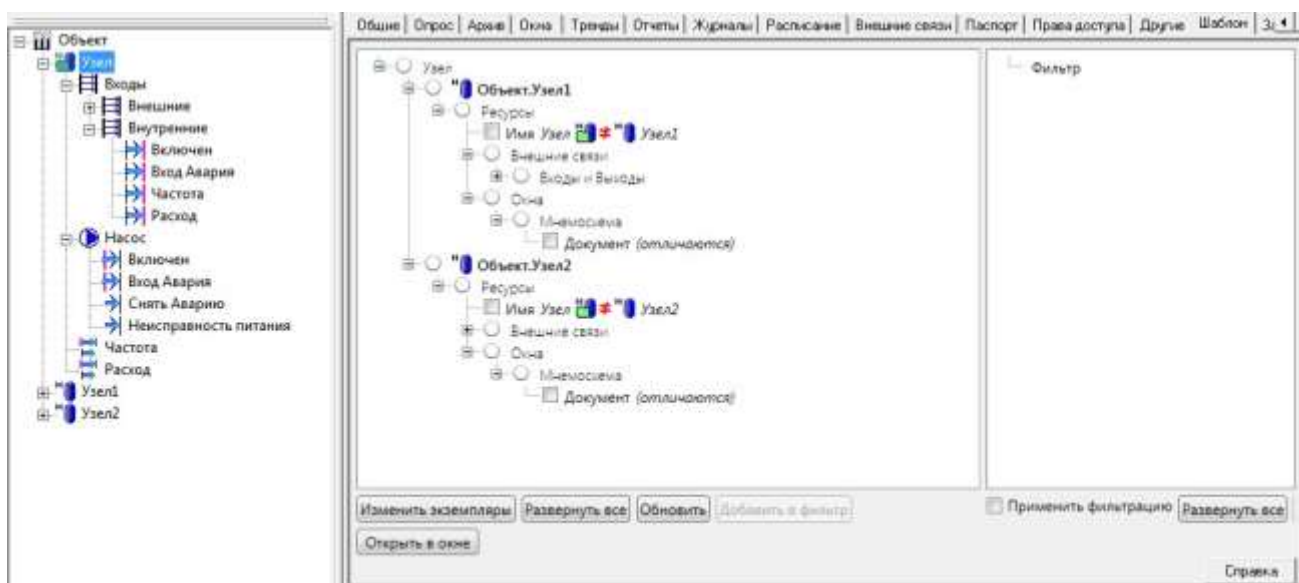
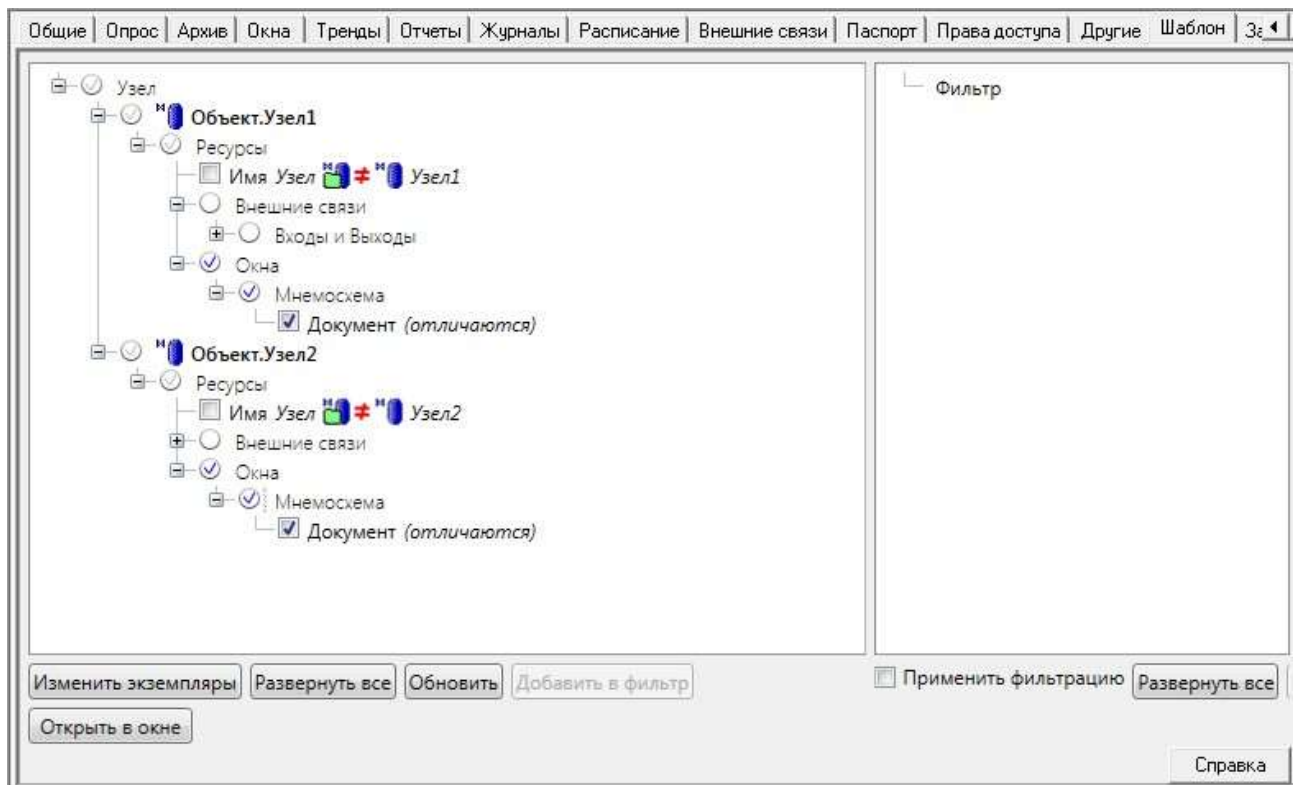


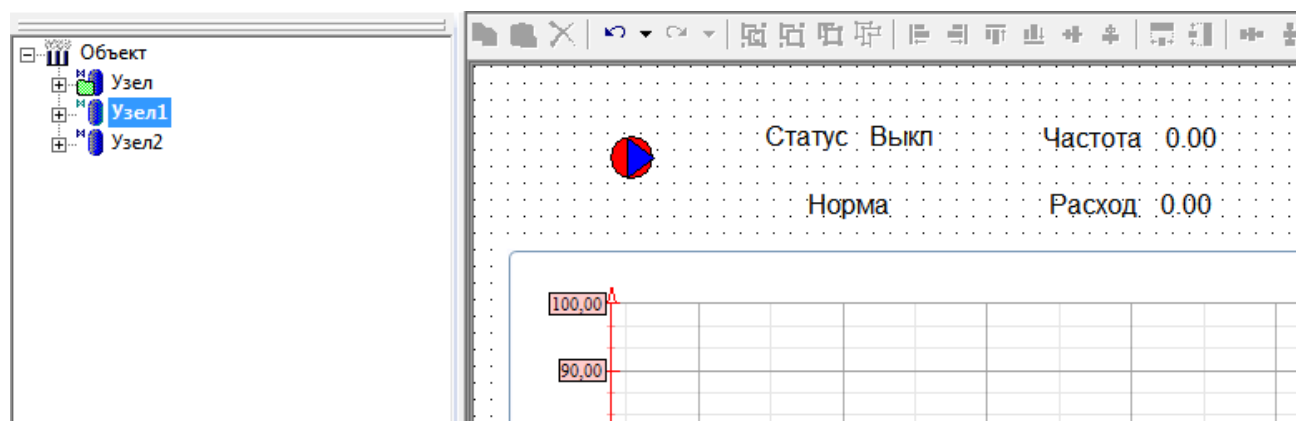
Рисунок 2.60. Дерево отличий шаблона и экземпляров

На данной вкладке будут выведены все отличия экземпляров от шаблона. В данный момент отличий три: различные имена объектов, различные внешние связи (отличаются пути к ОРС-переменным) и отличие в документе в мнемосхеме (Рисунок 2.60). Имена и внешние связи нам синхронизировать не нужно, нужно внести правки только в мнемосхему. Отметим галочкой мнемосхему и нажмем кнопку Изменить экземпляры (Рисунок 2.61).



**Рисунок 2.61. Внесение изменений в экземпляры**

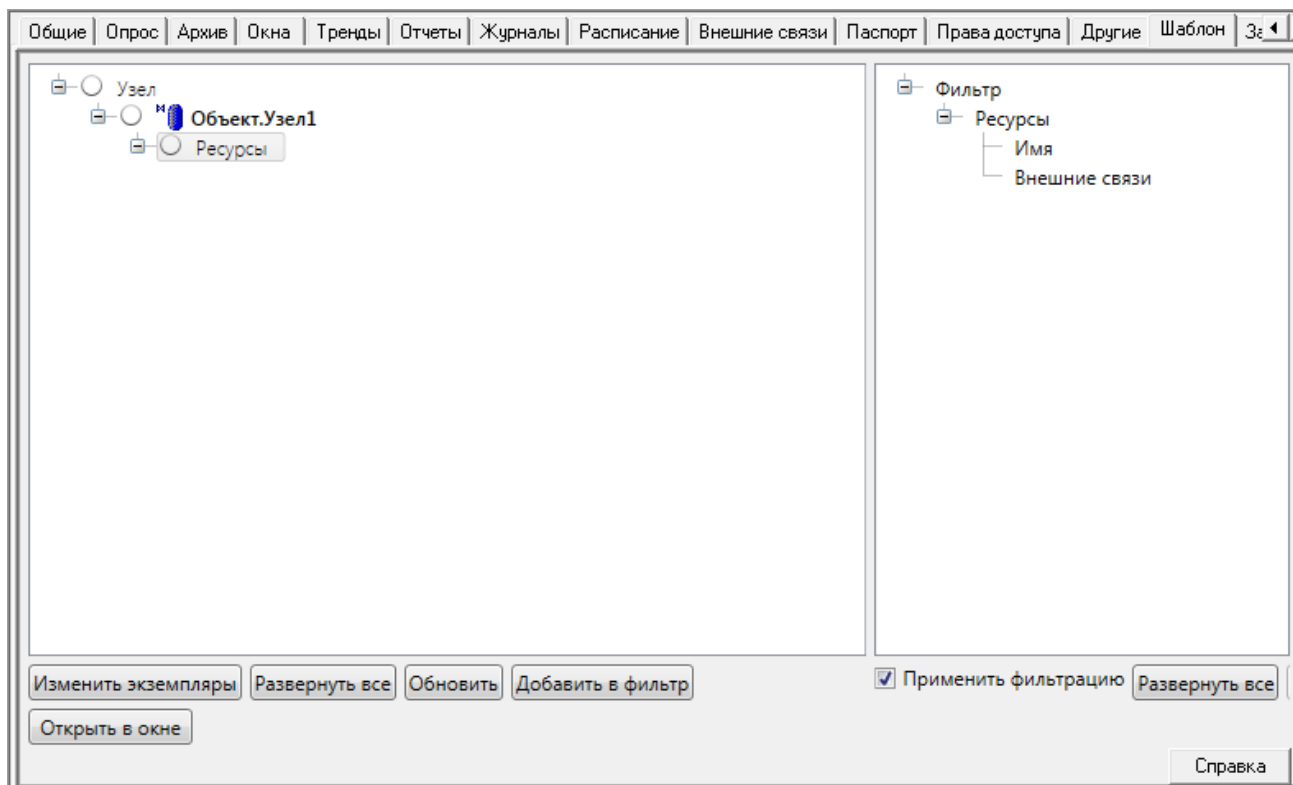
Изменения будут внесены, и отличие в мнемосхеме пропадет из дерева. Проверим, что изменения действительно внеслись – откроем мнемосхему объекта Узел1 (Рисунок 2.62). Изменения внесены.



**Рисунок 2.62. Проверка изменений**

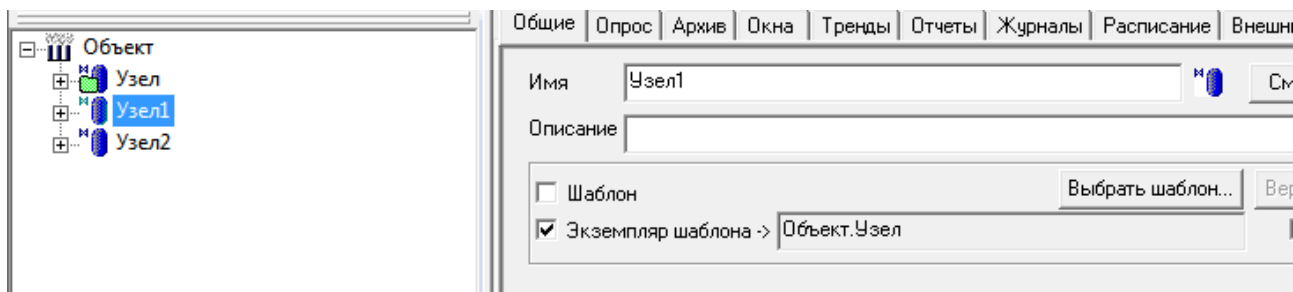
На вкладке Шаблон также можно настроить фильтрацию. Например, если вы не планируете синхронизировать имена и внешние связи (это, как

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 2.63).



**Рисунок 2.63. 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 2.64. Привязка объекта к шаблону**

### **3. Разработка проектов АСУТП**

#### **3.1. Стандарт OPC. Технология OPC в среде Master SCADA.**

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.

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.

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.

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

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 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.

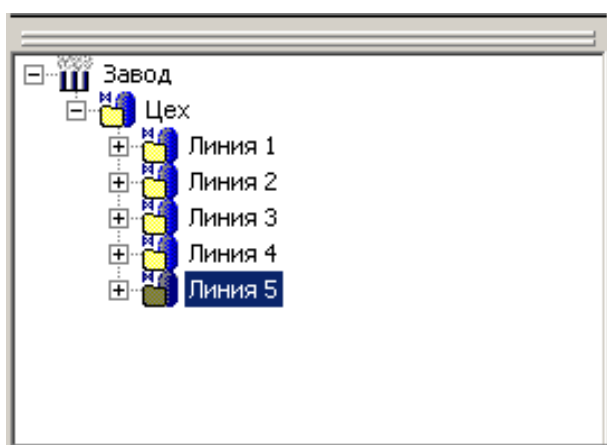
Например, мы автоматизируем на заводе цех по производству продукта X. Посмотрим, какая иерархия объектов будет в этом случае. Есть главный объект – это «Завод»; заходим на его территорию и видим наш объект автоматизации – «Цех». У этого объекта должно быть свое графическое окно – мнемосхема, куда будут 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 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 Рисунке 3.1.



**Рисунок 3.1. Дерево объектов**

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

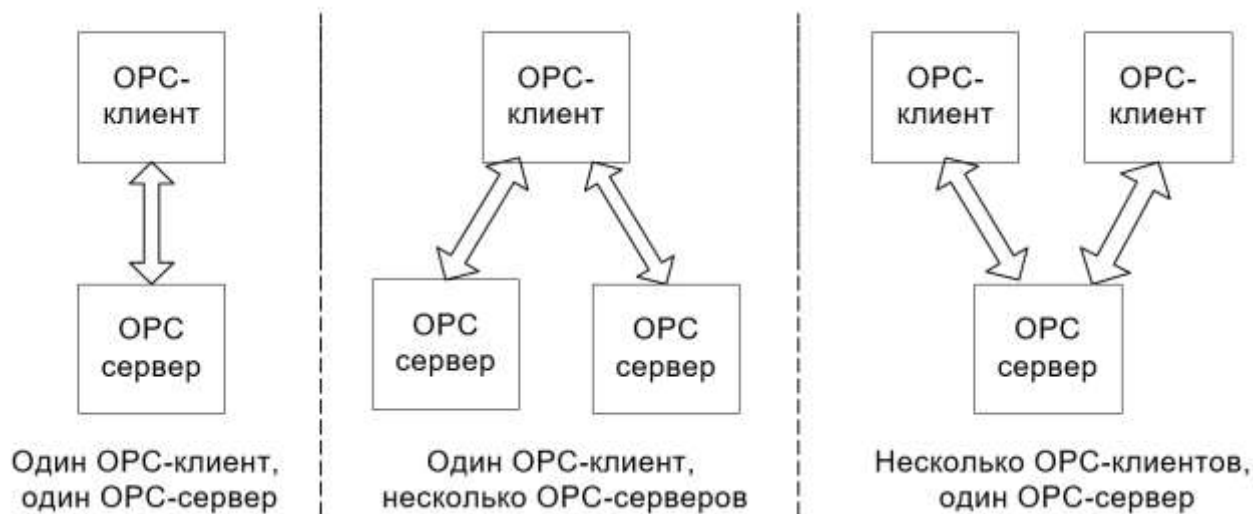
- Можно выделить логически связанную группу переменных и функциональных блоков.
- Нужно создать для выделенной группы какие-либо документы, характеризующие ее как единое целое.

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).



**Рисунок 3.2. 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 3.3).



**Рисунок 3.4. Клиент-серверная архитектура ОПС**

ОПС – 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](http://www.opcfoundation.org)). Различают следующие стандарты:

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

Объектная идеология Master SCADA- Инструментальная среда MasterSCADA предлагает объектный подход к разработке проекта. Объект в MasterSCADA - это основная единица разрабатываемой системы, соответствующая 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 имеет набор свойств и документов, которые жестко связаны с ним.

Свойства объекта – это, например, период опроса и способ обработки сигналов от его датчиков. Документы объекта – его изображение, описание, чертеж, перечень сообщений и т.п. В MasterSCADA нет просто тренда, рапорта или мнемосхемы: каждый документ в разрабатываемой системе всегда относится к какому-либо объекту, являясь его свойством. По умолчанию все настройки наследуются от родительского объекта.

Каждый объект имеет множество настроек. Такое обилие могло бы потребовать от разработчика системы выполнения огромного количества действий. Но так как для разных объектов их список в основном одинаков, то все настройки можно сделать только один раз, все подчиненные объекты автоматически воспримут настройки родительского элемента, то есть унаследуют их. Исключение будут составлять только те настройки и только у тех элементов, которые разработчик изменил сам. Схема наследования свойств представлена на следующем рисунке.



**Рисунок 3.5. Системная архитектура OPC**

Допускается многократное использование одного и того же объекта со всеми созданными для него документами, в том числе при разработке различных систем. При копировании объекта или сохранении его в библиотеке все его настройки и документы, настройки документов и внутренние связи будут сохранены.

Внешние связи с источниками данных будут восстановлены при наличии источников с такими именами, внешние связи с приемниками данных будут восстановлены, если эти приемники данных свободны, остальные будут показаны в общем списке. Благодаря этому управление и контроль типовым технологическим объектом (насосом, задвижкой, реактором, фильтром и т.п.) создаются один раз для всех проектов. Это позволяет создавать объекты для одной системы параллельно независимыми разработчиками.

Связывать переменные Объекта с OPC переменными можно двумя способами:

- Если в Объекте уже есть переменная, которую вы хотите связать с OPC переменной, то перетащите мышью одну к другой (также, как делается связь между двумя переменными объекта);
- Если необходимо создать сразу несколько связей, то не обязательно предварительно вставлять переменную объекта. Возможно выделить несколько переменных в дереве Системы и перетащить их в дерево объектов. При этом в Объекте автоматически будут созданы переменные с названиями,

максимально соответствующими названиям OPC переменных (без конфликтов имен), у которых уже будет настроена связь.

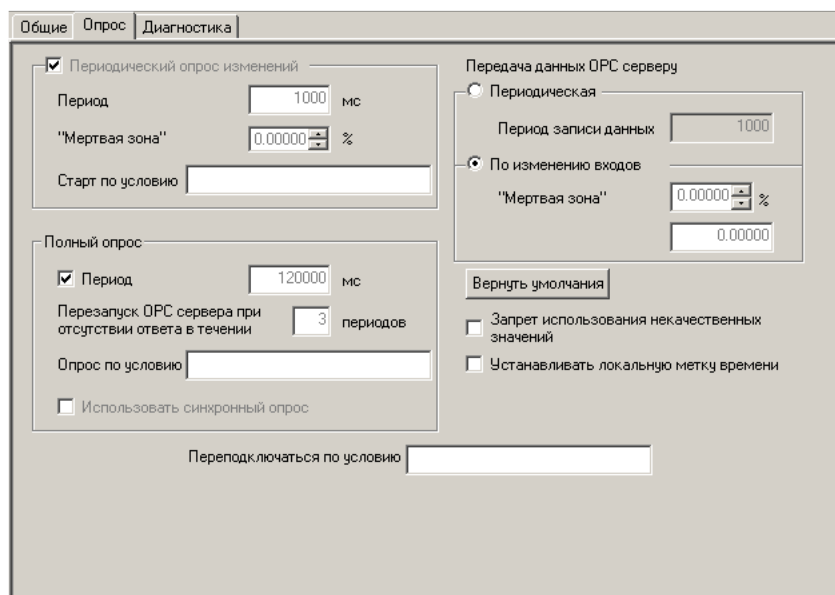
Устранение неполадок:

- ***OPC сервер не виден из клиента.*** Необходимо проверить, что *opc enum* правильно настроен и работает в данный момент. Если проблему не удалось устранить, то необходимо повторить установку *opc enum*, используя пакет *OPC Foundation - OPC Core Components 2.00 Redistributable*;
- ***OPC сервер виден, но невозможно вставить его переменные.*** Проверить загружаемую по умолчанию конфигурацию OPC сервера. Необходимо её настроить спомощью средств OPC сервера. Проверить, что данный сервер зарегистрирован в операционной системе и его *dcom* компонент настроен в соответствии с инструкцией. Для OPC сервера ИнСАТ необходимо запустить утилиты *unregister*, а затем *register*. Это приведёт к повторной регистрации OPC сервера;
- ***OPC сервер виден, но не запускается.*** Если клиент выдаёт сообщение: *Не возможно запустить OPC сервер на удалённой машине*, то необходимо повторить регистрацию OPC сервера и регистрацию этой библиотеки *opcproxu.dll*, которая расположена в папке *%system32%* операционной системы *Windows*;
- ***OPC сервер виден и переменные вставлены, но в режиме работа значения переменных отсутствуют.*** Проверить, что в настройках *dcom типа подключения*, установлено *нет*. Проверить связь с прибором, возможна проблема с драйве-ром устройства. Смотрим отладочную информацию по OPC серверу (есть или нет ответа от прибора).

Страница свойств Опрос сервера OPC: MasterSCADA принимает значения переменных из сервера OPC при их изменении:

- MasterSCADA получает значения из сервера OPC не чаще, чем величина периода (задается в группе Периодический опрос изменений);

- Сигнал принимается, если он изменился больше, чем на величину мертвой зоны (за-дается в группе Передача данных OPC серверу);
- Период (задается в группе Полный опрос), с которым MasterSCADA принудительно запрашивает значения, даже если они не изменились.

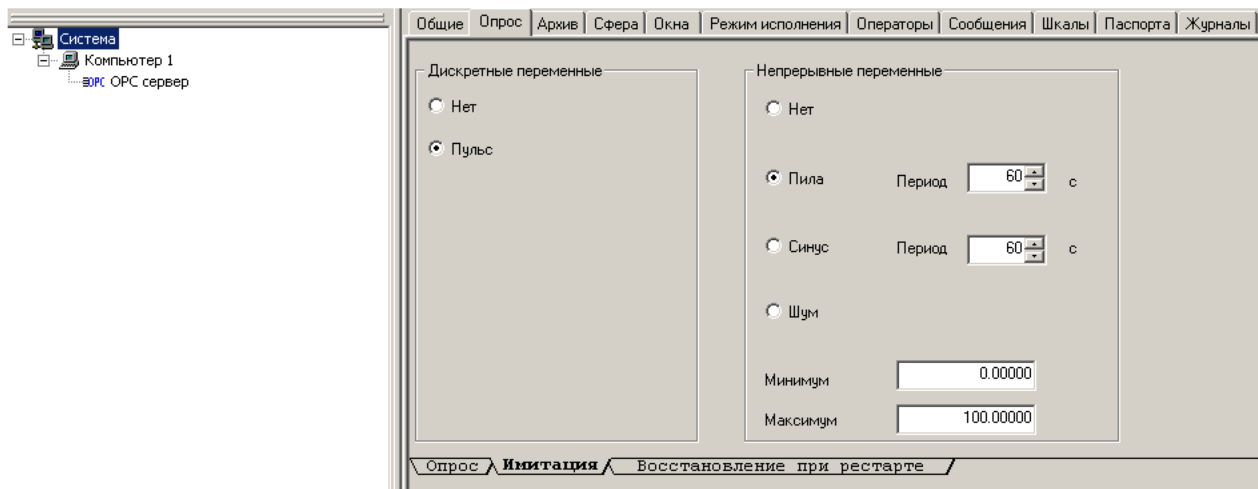


**Рисунок 3.5. Общие параметры OPC**

В режиме имитации на Входы без связей подаются имитационные сигналы. Здесь есть возможность настроить законы, по которым эти сигналы будут изменяться. Обратите внимание, что дискретные команды будут сменять друг друга не с заданным периодом, а с периодом опроса.

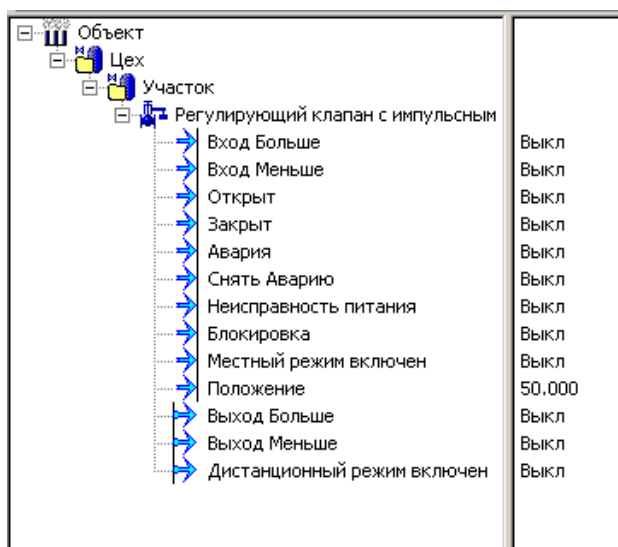
Значения Входов и Выходов видны прямо в дереве объектов. Можно увидеть значение любой переменной без каких-либо специальных действий как показано на рисунке ниже.



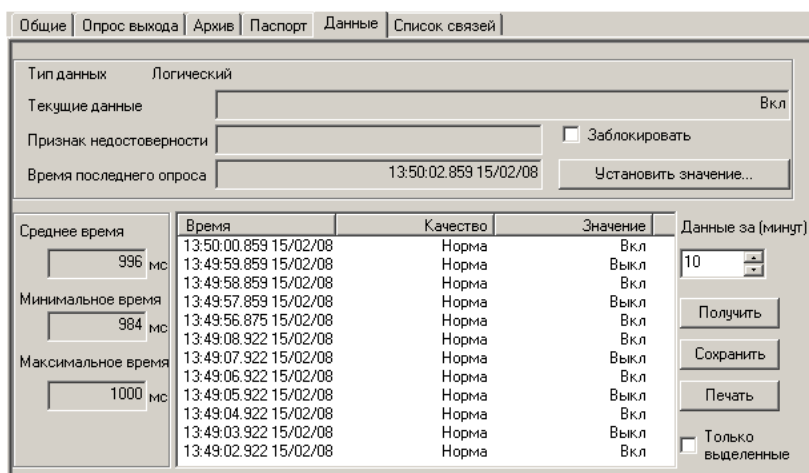


**Рисунок 3.5. Опрос OPC**

Чтобы управлять значением, следует выделить *Вход* или *Выход*, открыть закладку *Данные*.



**Рисунок 3.6. Структура системы управления**



### Рисунок 3.7. Опрос периода OPC

В поле с данными можно просмотреть архив переменной с метками времени;

- Если необходимо, чтобы значение не изменялось, то следует поставить флаг в поле *Заблокировать*;
- В поле *Текущие данные* отображаются текущее значение;
- Чтобы установить свое значение, следует нажать на кнопку *Установить значение*.

Чтобы выяснить оптимальный период опроса, необходимо:

- Для OPC сервера на закладке *Опрос* выставить период меньше ожидаемого;

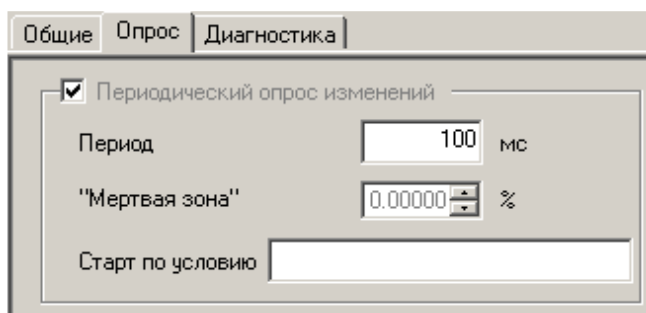


Рисунок 3.8. Панель времени опрос

В режиме исполнения для переменной открыть закладку *Данные* и проследить среднее время. Это время и нужно будет выставить в качестве периода опроса.

Время последнего опроса		14:07:17.234 15/02/08	
Среднее время	Время	Качество	
998 мс	14:07:09.234 15/02/08	Норма	
	14:07:08.234 15/02/08	Норма	
	14:07:07.234 15/02/08	Норма	
Минимальное время	14:07:06.234 15/02/08	Норма	
984 мс	14:07:05.234 15/02/08	Норма	
	14:07:04.250 15/02/08	Норма	
Максимальное время	14:01:30.000 15/02/08	Норма	
1000 мс	14:01:29.000 15/02/08	Норма	
	14:01:28.000 15/02/08	Норма	
	14:01:27.000 15/02/08	Норма	
	14:01:26.000 15/02/08	Норма	
	14:01:25.000 15/02/08	Норма	

**Рисунок 3.9. Панель времени опрос**

### 3.2. Режим отладки и запуск разработанного проекта в Master SCADA.

Прежде чем использовать функциональные блоки из палитры MasterSCADA, убедитесь в том, что они реализованы в исполнительной системе контроллера. В контроллере не реализованы блоки, работающие со строками, с временем, с буферами данных, с архивом, с файлами, с признаками качества и нарушения границ.

Для каждого из поддерживаемых типов контроллеров используется отдельная исполнительная система (системное программное обеспечение).

Исполнительная система включает в себя следующие файлы:

- *autoexec.bat* - пакетный файл для загрузки исполнительной системы;
- *I<тип\_контроллера>.exe* - исполняемый файл системы. Например для контроллера 7188E: *I7188E.exe*;
- *7188hw.ini* - конфигурационный файл для программы связи с контроллером.

Данный файл содержит следующие строки:

S3 V115200 P0 D8 S1

F

Xautoexec.bat Xi8000.exe

w25

-----

C-номер порта

B - скорость обмена

P-четность

D-Размер байта

S-стоп бит

*Xautoexec.bat Xi8000.exe* -устанавливает файлы для автозагрузки

w25 - устанавливает разрешение дисплея в колонках (по умолчанию 25 - макс 50)

Список функциональных блоков, реализованных в контроллере:

Категория	Наименование ФБ
Исполнительные механизмы	Насос мнемонический
	2-поз. мех. в выб. изобр.
	Задвижка
	Воздуходувка
	Насос
	Задвижка мнемоническая
	Динамический ИМ
	КПП мнемонический
	Клапан переключения потока
	Регулирующий клапан с импульсным управлением
ФБ Протокол	
Управление	2-поз. регулятор
	Циклограмма Не формируется выход <i>Имя шага</i>
	ПИД-регулятор
Алгоритмы VNS	Деление с доп. выходами
	Счетчик дискретных импульсов
	$X:=Y \text{ if } A=B$
	Контроль равенства по значению
	Контроль диапазона
	Звено запаздывания Необходимо указывать максимальное число хранимых значений
	Максимум из нескольких знач
	Минимум из нескольких знач
	Ограничитель
	Счетчик пробега

*7188xw.exe*. Файл не загружается в контроллер, служит для настройки параметров связи с контроллером (№ порта, скорость передачи данных и т.п.). Перед началом работы с новым контроллером необходимо загрузить комплект перечисленных файлов в контроллер. Этот комплект находится в папке *Program Files\InSAT\MasterSCADA\Config\MasterPLC* во вложенной папке с названием контроллера. В эту же папку необходимо загрузить программу связи с контроллером *7188xw.exe* (<http://www.icpdas.com/download/7188xw7188x.htm>).

Для загрузки исполнительной системы в контроллер нужно проделать следующие операции:

- Контакт *INIT* контроллера необходимо соединить с контактом *GND* (для *I-7000*) или с контактом *INIT COM* (для *I-8000*);
- Включите питание контроллера;
- Запустить программу связи с контроллером;
- В появившемся окне необходимо набрать команду *del*, которая предложит удалить уже существующие файлы в контроллере;
- После удаления файлов в контроллере, нажмите *ALT+F9*. После этого начнется загрузка исполнительной системы в контроллер;
- Выключите питание контроллера;
- Контакт *INIT* контроллера разъедините с контактом *GND* (для *I-7000*) или с контактом *INIT COM* (для *I-8000*);
- Для выставления адресов контроллеров необходимо в файле *autoexec.bat*, загружаемом в контроллер, прописать строку вида;
- *I7188XA.exe /an*, где *n* - адрес данного контроллера.

Системное программное обеспечение представляет собой один исполняемый файл. Этот файл выполняется под управлением встроенной операционной системы микроконтроллера (*DOS, MiniOS7* и др.). При запуске исполнительная система загружает конфигурацию из конфигурационного файла. На основе этой конфигурации ядро исполнительной системы за-

пускает на выполнение пользовательские и системные задачи, реализующие заложенную в конфигурацию при создании функциональность. Конфигурационный файл генерируется ин-струментальной средой разработки. Архитектура исполнительной системы представлена на рисунке ниже.



Все задачи в исполнительной системе контроллера работают под управлением многозадачного ядра реального времени. Задачи выполняются параллельно, в режиме приоритетной многозадачности. Все задачи пользователя имеют одинаковый приоритет. Если объекты, для которых создавались схемы ФБ, имеют одинаковый период выполнения, то все схемы ФБ выполняются в одной задаче в порядке их расположения в проекте. Если у какого-либо объекта период выполнения отличается, то его схема ФБ выполняется в отдельной задаче.

#### 4. Заключение

Данное пособие рассматривает базовые вопросы создания проектов для слежение технологических процессов в реальном времени. Вы можете использовать данные примеры и в дальнейшем модифицировать их для определённого типа технологических процессов. Рекомендуется ознакомиться с стандартными типами библиотек Master SCADA и добавить их в библиотеку вашей рабочей среды

## 5. Использованная литература

1. ИнСАТ “MasterSCADA Руководство пользователя Версия 3.X”, Москва 2017 г., 574 стр.
2. *Потоцкий И.В., “Практикум по Master SCADA ” НТУУ “КПИИ”, г.Киев 2017 6 стр.*
3. А. Третьяков, А. Н. Пчелинцев и др. Интегрированные системы проектирования и управления: SCADA-системы : учебное пособие – Тамбов, 2015. – 160 с.
4. Boyer, Stuart A.. *SCADA Supervisory Control and Data Acquisition*. USA: ISA - International Society of Automation. 179 стр.
5. Jeff Hieb (2008). *Security Hardened Remote Terminal Units for SCADA Networks*. University of Louisville.
6. R. Fan, L. Cheded and O. Toker, "Internet-based SCADA: a new approach using Java and XML," in *Computing & Control Engineering Journal*, vol. 16, no. 5, pp. 22-26, Oct.-Nov. 2005.
7. Qiu B, Gooi HB. Web-based scada display systems (wsds) for access via internet. *Power Systems, IEEE Transactions on* 2000;15(2):681–686.
8. . M. Lynch, “An Internet Based SCADA System”, BSc Project Report, University of Southern Queensland, Queensland, Oct. 2005
9. Robert Lemos (26 July 2006). "SCADA system makers pushed toward security". *SecurityFocus*. Retrieved 9 May 2007.
10. J. Russel. "A Brief History of SCADA/EMS (2015)". Archived from [the original](#) on 11 August 2015.

**Газиева Раъно Тешабаевна , Ядгарова Дилноза Бахтияровна,  
Нигматов Азиз Махкамович , Озодов Эъзоз Ойбекович**

## **МАСТЕР SCADA**

Учебное пособие

для студентов специальности

5311000-Автоматизация и управление технологических процессов и  
производств (в водном хозяйстве)



Bosishga ruxsat etildi \_\_\_\_\_ 20 y.

Qog`oz o`lchami 60x84 1/16

Hajmi \_\_\_\_\_ bosma taboq \_\_\_\_\_ nusxa

Buyurtma №

TIQXMMI bosmaxonasida chop etildi

Toshkent - 700000, Qori-Niyoziy ko`chasi, 39-uy.