

УТВЕРЖДАЮ

Директор ИШИТР ТПУ

«___» _____ 2023 г.

МЕЖДИСЦИПЛИНАРНЫЙ ПРОЕКТ

Часть 2

Задание и методические указания к выполнению проекта

Дисциплина	Междисциплинарный проект
Школа	Информационных технологий и робототехники (ИШИТР)
Отделение	Автоматизации и робототехники
Направление	15.03.04 Автоматизация технологических процессов и производств
ООП	Автоматизация технологических процессов и производств в нефтегазовой отрасли
Квалификация	Бакалавр
Курс / группа	4 / 8Т91
семестр	8

Зав. кафедрой -
руководитель ОАР
ИШИТР

_____ А. А. Филипас

Руководитель ООП

_____ М. В. Скороспешкин

Преподаватель

_____ В. В. Курганов

2023 г.

ВВЕДЕНИЕ

Цель настоящей дисциплины – выполнение проекта, который объединяет ряд дисциплин и, по сути, является междисциплинарным.

Настоящий курсовой проект (2 часть) является логическим продолжением курсового проекта (1 часть), который выполнялся в 7 семестре и был посвящён разработке резервированного измерительного канала на базе промышленного контроллера.

Выполнение 2-й части проекта позволит сформировать целостную картину процесса проектирования систем автоматизированного управления технологическими объектами и процессами.

Выполнение данного курсового проекта предполагает разработку видеоинтерфейса на базе программы MasterSCADA с последующей демонстрацией результатов на реальных технических средствах.

Курсовой проект объединяет ряд дисциплин, которые изучались, изучаются и будут изучаться в рамках бакалаврской подготовки по направлению 15.03.04 «Автоматизация технологических процессов и производств», а именно:

Микропроцессорная техника и системы автоматизации;

Проектирование систем управления;

Метрология, стандартизация и сертификация;

Программирование и алгоритмизация;

Программное обеспечение АСУ ТП.

1. КРАТКАЯ ИНСТРУКЦИЯ ПО ВЫБОРУ ТЕМЫ КУРСОВОГО ПРОЕКТА.

Тема и исходные данные для выполнения курсового проекта (вариант задания) выбираются в соответствии порядковым номером в зачетной ведомости.

2. СОДЕРЖАНИЕ КУРСОВОГО ПРОЕКТА

Курсовой проект должен содержать расчетно-пояснительную записку и графическую часть. В соответствии с общеинститутскими требованиями объём неправомерного заимствования результатов работы других авторов для курсовых проектов не должен превышать 15%.

2.1. Тема курсового проекта:

Интерфейс оператора на базе MasterSCADA

2.2. Для выполнения курсового проекта по дисциплине «Междисциплинарный проект» необходимо выполнить следующую работу.

1. Изучить поставленную задачу.
2. В соответствии с вариантом задания:
 - провести краткий обзор по вопросу решения поставленной задачи;
 - изучить программные средства для реализации поставленной задачи.
3. Спроектировать форму видеокadra.
4. Установить программу MasterSCADA на персональный компьютер, ноутбук.
5. Установить программу CODESYS и OPC-сервер для промышленного контроллера (ПЛК), заданного в части 1 курсового проекта.
6. На MasterSCADA разработать видеокادر в соответствии с вариантом.

7. Загрузить в ПЛК алгоритм мажоритарного выбора, разработанный в первой части курсового проекта, продемонстрировать его работу.
8. Выполнить настройку связи ПЛК со станцией оператора.
9. Продемонстрировать совместную работу ПЛК и станции оператора.
10. Оформить отчет по результатам выполнения курсового проекта.

3. ПРАВИЛА ОФОРМЛЕНИЯ РЕЗУЛЬТАТОВ КУРСОВОГО ПРОЕКТА

Объём расчётно-пояснительной записки 15-17 стр. Записка оформляется в любом текстовом редакторе и сдаётся в бумажном виде. Желательный шрифт текста Times New Roman, размер 14.

В случае отсутствия доступа к текстовому редактору, возможно оформление пояснительной записки «от руки».

Графический материал оформляется в любом доступном графическом редакторе. Соблюдение масштаба и норм и правил ЕСКД обязательно.

В случае отсутствия доступа к графическому редактору (машинной графике вообще), графическую часть проекта выполнить карандашом на чертеже формата А3 (297x420 мм) с соблюдением масштаба и норм и правил ЕСКД.

Сроки сдачи готового курсового проекта 12 ... 13 неделя семестра.

Защита курсового проекта:

Штатная защита курсового проекта проводится в течение 13 недели семестра

Для защиты курсового проекта продемонстрировать работу станции оператора с ПЛК.

Нештатная защита (защита курсового проекта по истечении установленного срока) проводится по правилам, установленным Учебным управлением ТПУ.

Варианты заданий

Вариант	Содержание видеокadra (измеряемый параметр)
1	Печь подогрева нефти (давление на выходе)
2	Компрессор (давление после второй ступени)
3	Реактор с тепловой рубашкой (температура рубашки)
4	Ректификационная колонна (уровень в кубе колонны)
5	Вертикальная ёмкость с мешалкой (количество оборотов)
6	Агрегат воздушного охлаждения (температура на выходе)
7	Регулятор расхода после себя (расход)
8	Центробежный насос (давление на нагнетании)
9	Расходомер, установленный на трубопроводе (расход)
10	Горизонтальная ёмкость с уровнемерной колонкой (уровень)
11	Дренажная ёмкость (уровень)
12	Регулятор давления до себя (давление)
13	Реактор с мешалкой (давление в реакторе)
14	Сепаратор нефти (граница раздела фаз)
15	Насос с входным фильтром (сопротивление фильтра)
16	Центробежный насос (температура подшипника)
17	Ректификационная колонна (давление на выходе)

Где скачать MasterSCADA?

Ответ: insat.ru

Продукт компании MasterSCADA 3.X.

Зарегистрироваться на сайте и скачать версию с ограничением on-lint на 1 час (размер файла приблизительно 300 Мв).

Установить на личный компьютер или ноутбук.

В ауд. 025 на компьютерах, связанных с ПЛК, версия MasterSCADA 3.X установлена.

4 РАЗРАБОТКА ИНТЕРФЕЙСА ОПЕРАТОРА

4.1 Выбор архитектуры системы

Для разработки интерфейса оператора используется SCADA – система.

Современные SCADA-системы это мощные программные пакеты, обеспечивающие комплексные решения задач, связанных с предоставлением информации оператору в самых различных видах, её регистрации, архивирования, использования. Как правило, SCADA-системы разворачиваются на персональных компьютерах. Для обмена данными с PLC, а также другими устройствами, используются стандартные интерфейсы, например, Ethernet, RS-232 и другие. Вопрос взаимодействия с внешним миром для SCADA-системы является наиважнейшим и именно поэтому разработчики таких систем всегда предлагают широкий выбор программ, так называемых драйверов, для связи с различными видами оборудования. Более того, в составе SCADA-системы всегда имеется инструмент для создания собственных драйверов на стандартных языках программирования.

Существует два основных способа обмена информацией между SCADA-системой и любым внешним устройством:

- протоколы, которые разрабатываются производителями SCADA-систем и интегрируются в программу;
- протоколы OPC (Open Platform Communications, ранее аббревиатура означала OLE for Process Control).

Первый вид протоколов, ввиду того что разрабатывается самим производителем, как правило, компактный и обеспечивает максимальное быстродействие. Недостатком является необходимость наличия драйвера к любому типу оборудования, что практически невозможно.

Второй способ связи наиболее универсальный и поддерживается практически всеми SCADA-системами. Протокол обмена OPC – технология, ориентированная на применение в задачах промышленной автоматизации, и в настоящее время является стандартом.

4.2 OPC - сервер

Структура взаимодействия SCADA-системы с устройством через OPC представлена на рисунке 4.1.

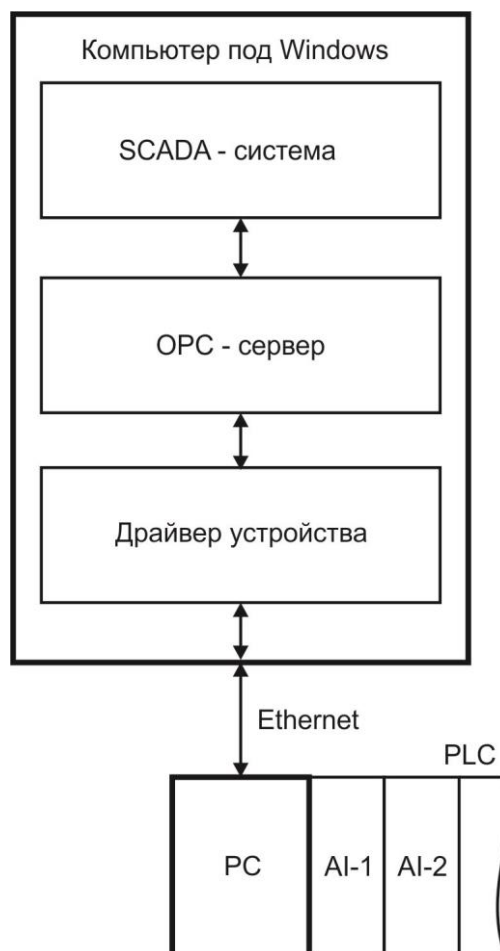


Рисунок 4.1 – Структура системы взаимодействия SCADA-системы с устройством через OPC

Стандарт OPC позволяет:

- отказаться от использования многочисленных протоколов обмена между SCADA – системой и устройствами;

- объединить различные системы контроля и управления на уровне объектов.

Главная цель стандарта OPC состоит в создании некоего универсального механизма взаимодействия между устройством и SCADA – системой.

В отличие от первого варианта, в котором драйверы связи с устройствами разрабатываются производителями SCADA – систем (не путать с драйвером устройства на рисунке 4.1, который выполняет функции подключения устройства к системе), во втором варианте разработка драйвера по стандартам OPC возлагается на производителя оборудования. Это означает, что производитель оборудования предлагает купить не только какие-либо устройства, но и OPC драйвер к ним, который позволит этим устройствам интегрироваться в любую SCADA – систему, поддерживающую OPC. Другими словами, задача разработки драйвера по известным правилам возложена на производителя устройства, избавив тем самым разработчиков SCADA – систем от рутинной работы.

На одном компьютере могут быть запущены несколько OPC-серверов, выполняющих связь с различными группами оборудования, и каждый из них может выступать в качестве OPC сервера, поставляя данные различным программам, включая SCADA – систему.

Обмен данными с OPC – сервером может происходить двумя различными способами:

1 способ – периодический, при котором клиенты запрашивают данные с определённой частотой;

2 способ – по изменению значения.

Второй способ является более предпочтительным, так как экономит время, но и реализуется сложнее, чем первый.

В таблице 4.1 представлены стандарты на разработку OPC, в каждом из которых прописано назначение определенных функций.

Таблица 4.1 - Стандарты на разработку OPC

Название	Содержание
OPC DA (Data Access)	Основной стандарт. Описывает набор функций обмена данными в реальном времени с ПЛК, РСУ, ЧМИ, ЧПУ и другими устройствами.
OPC AE (Alarms & Events)	Предоставляет функции уведомления по требованию о различных событиях: аварийные ситуации, действия оператора, информационные сообщения и другие.
OPC Batch	предоставляет функции шагового и рецептурного управления технологическим процессом
OPC DX (Data eXchange)	Предоставляет функции организации обмена данными между OPC-серверами через сеть Ethernet. Основное назначение — создание шлюзов для обмена данными между устройствами и программами разных производителей.
OPC HDA (Historical Data Access)	Предоставляет доступ к уже сохраненным (историческим) данным.
OPC Security	Определяет функции организации прав доступа клиентов к данным системы управления через OPC-сервер.
OPC XML-DA (XML-Data Access)	Предоставляет гибкий, управляемый правилами формат обмена данными через SOAP и HTTP.
OPC UA (Unified Architecture)	Последняя по времени выпуска спецификация, которая основана не на технологии Microsoft COM, что предоставляет кросс-платформенную совместимость.

4.3 SCADA – системы

В настоящее время разработчики предлагают достаточно широкий выбор SCADA – систем, отличающихся как по стоимости, так и по функциональным возможностям.

Для того чтобы привлечь пользователей использовать в проектах свои программные продукты, отечественные разработчики используют различные маркетинговые шаги, например:

- бесплатное распространение систем разработки интерфейса оператора;
- ограниченная по времени работа run-time версии;
- неограниченная работа run-time версии с ограниченным функционалом;
- приобретение SCADA – системы с заданным функционалом;
- т. д.

Рассмотрим некоторые отечественные SCADA – системы

4.3.1 MasterSCADA

Разработчиком программы MasterSCADA является компания ИнСат г. Москва. MasterSCADA полнофункциональный пакет программ, предназначенный для визуализации технологических процессов и решения всевозможных сопутствующих задач, а именно:

- редактирование мнемосхем;
- тренды технологических параметров;
- ведение внутреннего архива данных;
- ведение архива сообщений и документов;
- создание и редактирование отчетов и т.д.

Предоставляет пользователю базовый набор библиотеки функциональных блоков, а также механизмы их создания на языках FBD, ST, C#.

Опционально в рамках программы могут поставляться возможности создания сетевых проектов, взаимодействие с внешними базами данных и т.д.

Особым достоинством программы MasterSCADA является политика разработчика в области её распространения:

- бесплатная среда разработки MasterSCADA;
- бесплатная версия на 32 тега с ограниченным функционалом;
- бесплатная полнофункциональная версия с ограниченным on-line режимом (время ограничения 1 час).

4.3.2 FreeSCADA

Пакет FreeSCADA является свободно распространяемой версией программного продукта Телескоп+. В пакет включены приложения Дизайнер форм и Пульт диспетчера.

В приложении Пульт диспетчера идет наблюдение за состоянием технологического оборудования и системы управления, получение срочные сообщения о нарушениях технологического процесса.

В приложении Дизайнер форм присутствует возможность разработки панели диспетчера.

По сравнению с коммерческой версией ПО Телескоп+ Дизайнер форм имеет следующие ограничения:

- отображение только текущих данных, без архивной информации;
- не поддерживается взаимодействие с базой данных;
- не поддерживается дерево объектов;

- не поддерживается просмотр отчетов и стандартный вывод сообщений об авариях;
- работа организована на одном компьютере.

4.3.3 SoloSCADA

SCADA-система SoloScada является Web ориентированной системой. Отображение технологического процесса производится в браузере или приложении клиент. Использование HTML5, JavaScript даёт возможность отображать мнемосхемы, таблицы, графики непосредственно в браузере на любом мобильном устройстве и компьютере. Благодаря векторной графике SVG мнемосхема автоматически масштабируется под разные разрешения экранов и разные размеры экранов браузеров.

SCADA-система позволяет получать данные от нескольких OPC-серверов, по протоколу ModBus. С помощью графического редактора можно нарисовать мнемосхему из графических примитивов, создавать свою библиотеку изображений, вставлять на мнемосхему различные готовые SVG-изображения, созданные в других более продвинутых редакторах. Можно создавать кнопки перехода на другие мнемосхемы. Есть изменение атрибутов (размера, цвета, положения) графических примитивов во время исполнения для визуализации технологического процесса

Поддержка получения данных по протоколу ModBus, ModBus-RTU, ModBus-TCP или ModBus-RTU поверх TCP.

Самостоятельная реализация обмена с контроллерами через COM-порт и TCP/IP для получения с них данных с помощью скриптов.

Есть возможность добавления в проект собственных html-страниц. Графики, таблицы отчеты. Их можно создавать самому, таким образом, обеспечивается гибкость.

Имеется расширяемое API взаимодействия клиента (Web-браузера) с сервером через json-формат.

Поддержка пользовательских скриптов. Скрипты исполняются периодически, при изменении значения переменной, при запуске сервиса, при остановке сервиса. Так же с помощью скриптов можно самому сгенерировать Json-файл в ответ на запрос Web-клиента.

Работа с несколькими разными базами данных MySQL, PostgreSQL, Microsoft SQL Server, SQLite одновременно. Вы сами можете создавать таблицы с нужным форматом под конкретную задачу.

Также имеется архивирование в файлы. Есть заготовки для просмотра данных из файловых архивов.

Управление правами пользователей: добавление новых, редактирование прав существующих пользователей.

Работа с алармами. Разделение важности сигналов (Аварийная и предупредительная сигнализация)

Главным недостатком является ориентирование на Web-сервис.

4.3.4 OpenSCADA

Программа OpenSCADA как и предыдущие программы предназначена для сбора, архивирования, визуализации информации, выдачи управляющих воздействий, а также других родственных операций, характерных для полнофункциональной SCADA системы. Область применения программы от пол-

нофункциональных SCADA – систем до построения различных моделей технологических процессов.

Внешне, OpenSCADA состоит из трех основных компонентов - модуля визуализации, модуля конфигурирования, и модуля настройки интерфейса визуализации.

Модуль визуализации – это рабочее место оператора АСУ ТП. Рабочее место показывает данные в различных формах, актуальные данные мониторинга, позволяет просмотреть значения параметров за любой промежуток времени, воспринимает команды управления, отданные оператором.

Основная настройка работы всей OpenSCADA системы происходит в модуле конфигурирования. В нем происходит настройка подсистемы сбора данных, указываются какие данные от датчиков, через какие транспортные порты будут передаваться. Часто данные от датчиков поступают в "сыром" виде, и чтобы привести их к понятным физическим размерностям, надо сделать над такими данными некоторые математические действия. В OpenSCADA есть встроенный Java-подобный язык, который позволяет делать любые вычисления над используемыми в среде значениями. Кроме того можно указать логику слежения за значениями датчиков, и при различных отклонениях можно выдавать различные управляющие сигналы.

В модуле конфигурирования так же настраивается архивирование данных. Архивы могут храниться в разных форматах и физически, для обеспечения надежности, могут даже располагаться на другом компьютере, используя базу данных, которая поддерживает репликацию.

Недостатком среды разработки OpenSCADA является возможность разработки и режима работы только на ОС Linux.

4.4 Выбор SCADA системы

В процессе анализа существующих на отечественном рынке систем диспетчерского контроля и визуализации был проведен сравнительный анализ следующих программных пакетов:

- MasterSCADA
- FreeSCADA
- SoloSCADA
- OpenSCADA

По итогу анализа для данной работы выбрана программа MasterSCADA, разработанная компанией ИнСАТ. MasterSCADA является низкобюджетной, вертикально интегрированной и легко расширяемой программой в зависимости от сложности объекта.

Основным достоинством, как отмечалось выше, является возможность разработки проектов визуализации без финансовых затрат на приобретение продукта и ответственная политика компании Инсат области сопровождения программного продукта.

4.5 Разработка экранных форм

Экранную форму или видеокадр для реализации курсового проекта следует разрабатывать в соответствии с заданием на проектирование. Смысл разработки экранной формы заключается не только в технической реализации вывода информации на экран, но и в развитии неких творческих (дизайнерских) способностей студентов.

На рисунке 4.1 представлена мнемосхема процесса ректификации нефти, разработанная студенткой в рамках выполнения выпускной квалификационной работы.

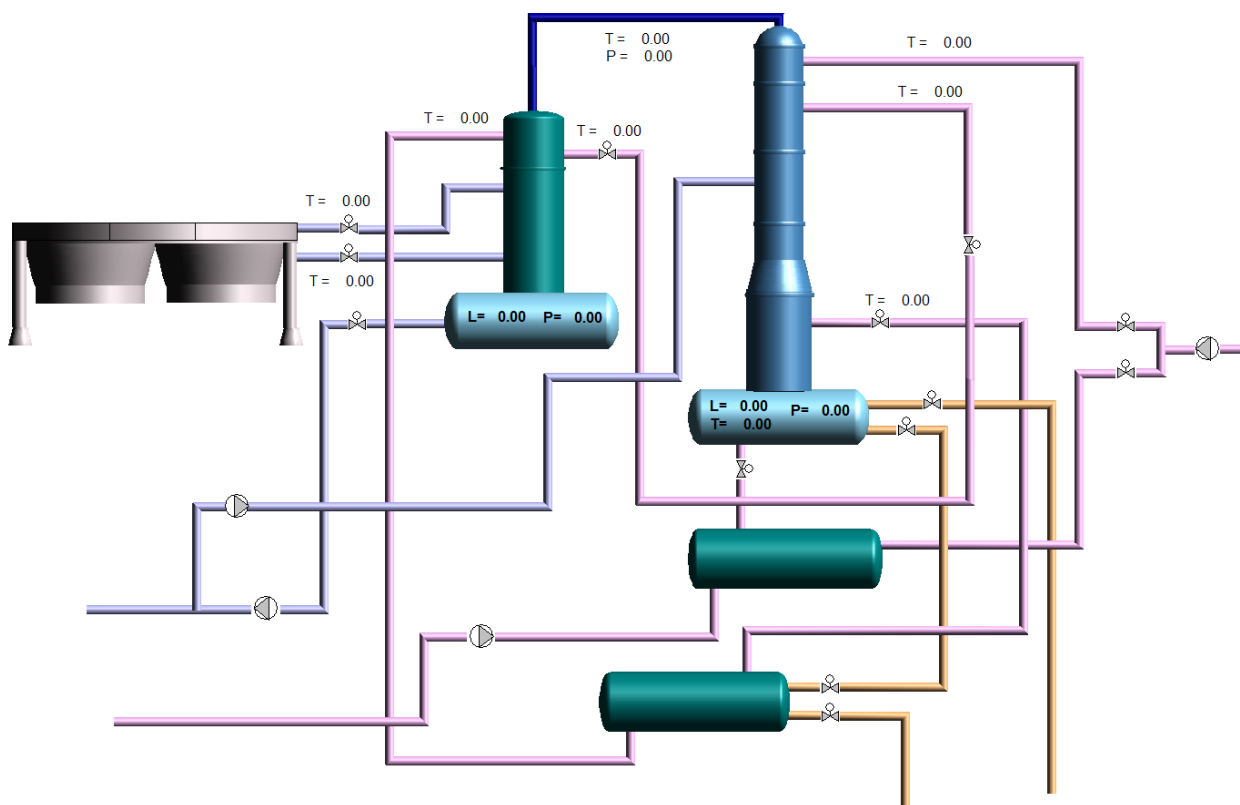


Рисунок 4.1 – Мнемосхем процесса ректификации нефти

Положительные моменты кадра

1. Трубопроводы продуктов и сырья выделены цветом – удобно отслеживать движение продуктов и сырья;
2. Колонны и ёмкости благодаря световой подсветке выглядят реалистично.
3. Соблюдены пропорции (масштаб) между элементами мнемосхемы.

Отрицательные моменты кадра

1. Объём информации, представленной на экране не должен превышать разумных пределов, особенно в случаях, когда информация должна анализироваться и по результатам анализа приниматься решения.
2. Информация, представленная на кадре, должна быть равномерно распределена по всей его площади.

3. Желательно чтобы на кадре было представлено технически или технологически законченное решение, например, компрессор, насос, колонна или их совокупность, если позволяет кадр.

4. Динамическая информация (измерения, сигнализация и т.д.) должна (желательно) отличаться от статической (надписи, единицы измерения и т.д.), например фоном, цветом и др.

5. Отсутствие единиц измерения.

Для работы в программе MasterSCADA предлагается **фрагмент учебно-методического пособия** М. В. Скороспешкина и В. Н. Скорспешкина разработанного для курса «Автоматизированные информационно управляющие системы».

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

М.В. Скороспешкин, В.Н. Скороспешкин

АВТОМАТИЗИРОВАННЫЕ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЕ СИСТЕМЫ

*Рекомендовано в качестве учебно-методического пособия
Редакционно-издательским советом
Томского политехнического университета*

Издательство
Томского политехнического университета
2014

4. МЕТОДИКА СОСТАВЛЕНИЯ ПРИКЛАДНЫХ ПРОГРАММ С ПРИМЕНЕНИЕМ ПАКЕТА MASTERSCADА

4.1. Назначение, состав и функции Scada-пакетов

SCADA (сокр. от англ. *Supervisory Control And Data Acquisition* – диспетчерское управление и сбор данных) – процесс сбора информации реального времени с удаленных точек (объектов) для обработки, анализа и возможного управления удаленными объектами.

Применение SCADA-систем позволяет существенно сократить сроки разработки программного обеспечения, обеспечить высокое качество регулирования, при этом при создании программного обеспечения профессиональные программисты могут и не привлекаться.

Все современные SCADA-системы включают три основных структурных компонента, представленные на рис. 25.



Рис. 25. Основные структурные компоненты SCADA-системы

Удаленный терминал (Remote Terminal Unit – RTU) осуществляет обработку задачи (управление) в режиме реального времени. В зависимости от конкретного применения может представлять собой от примитивных датчиков, осуществляющих съем информации с объекта, до специализированных многопроцессорных отказоустойчивых вычислительных комплексов, осуществляющих обработку информации и управление в режиме жесткого реального времени. Применение устройств низкоуровневой обработки информации позволяет снизить требования к пропускной способности каналов связи с центральным диспетчерским пунктом.

Диспетчерский пункт управления (Master Terminal Unit – MTU) осуществляет обработку данных и управление высокого уровня, как правило, в режиме реального времени. Одна из его основных функций – обеспечение интерфейса между человеком-оператором и системой. В зависимости от конкретной системы диспетчерский пункт может быть реализован в виде одиночного компьютера с дополнительными устройствами подключения к каналам связи или больших вычислительных систем и/или объединенных в локальную сеть рабочих станций и серверов.

Как правило, и при построении диспетчерского пункта используются различные методы повышения надежности и безопасности работы системы.

Коммуникационная система [каналы связи – Communication System (CS)] необходима для передачи данных с удаленных точек (объектов, терминалов) на центральный интерфейс оператора-диспетчера и передачи сигналов управления на диспетчерский пункт (или удаленный объект – в зависимости от конкретного исполнения системы).

В современных диспетчерских системах выделяют следующие особенности процесса управления:

- процесс SCADA применяется в системах, в которых обязательно наличие человека (оператора, диспетчера);

- процесс SCADA был разработан для систем, в которых любое неправильное воздействие может привести к отказу (потери) объекта управления или даже катастрофическим последствиям;

- оператор несет, как правило, общую ответственность за управление системой, которая, при нормальных условиях, только изредка требует подстройки параметров для достижения оптимальной производительности;

- активное участие оператора в процессе управления происходит нечасто и в непредсказуемые моменты времени, обычно в случае наступления критических событий (отказы, нештатные ситуации и пр.);

- действия оператора в критических ситуациях могут быть жестко ограничены по времени (несколькими минутами или даже секундами).

К SCADA-системам предъявляются следующие основные требования:

- надежность системы (технологическая и функциональная);

- безопасность управления;

- точность обработки и представления данных;

- простота расширения системы.

Требования безопасности и надежности управления в SCADA включают:

- никакой единичный отказ оборудования не должен вызвать выдачу ложного выходного воздействия (команды) на объект управления;

- никакая единичная ошибка оператора не должна вызвать выдачу ложного выходного воздействия (команды) на объект управления;

- все операции по управлению должны быть интуитивно-понятными и удобными для оператора (диспетчера).

В силу тех требований, которые предъявляются к системам SCADA, спектр их функциональных возможностей определен и реализован практически во всех пакетах. Перечислим основные возможности

и средства, присущие всем системам и различающиеся только техническими особенностями реализации:

- автоматизированная разработка, дающая возможность создания программного обеспечения (ПО) системы автоматизации без реального программирования;
- средства сбора первичной информации от устройств нижнего уровня;
- средства управления и регистрации сигналов об аварийных ситуациях;
- средства хранения информации с возможностью ее пост-обработки (как правило, реализуется через интерфейсы к наиболее популярным базам данных);
- средства обработки первичной информации;
- средства визуализации представления информации в виде графиков, гистограмм и т. п.

Программные продукты класса SCADA широко представлены на мировом рынке. Это несколько десятков SCADA-систем, многие из которых нашли свое применение и в России. Наиболее популярные из них:

- RTAP/Plus – Канада.
- InTouch (Wonderware) – США.
- Citect (CI Technology) – Австралия.
- FIX (Intellution) – США.
- Genesis (Iconics Co) – США.
- Factory Link (United States Data Co) – США.
- RealFlex (BJ Software Systems) – США.
- Sitex (Jade Software) – Великобритания.
- Trace Mode (AdAstrA) – Россия.
- MasterScada (Инсат) – Россия.
- Simplicity (GE Fanuc) – США.

В данной курсовой работе используется программный пакет MasterScada, который используется для визуализации процесса работы системы регулирования и сигнализации. В связи с этим рассмотрим подробнее отличительные особенности и функциональные возможности данного пакета.

4.2. Описание программного пакета MasterScada

MasterScada – это не просто один из современных SCADA-пакетов, это принципиально новый инструмент разработки АСУ ТП, в котором реализована совокупность средств и методов, обеспечивающих резкое

сокращение трудозатрат и повышение надежности создаваемой системы, он также характеризуется оптимальным соотношением цены и качества. MasterScada является полнофункциональным SCADA-пакетом программ с расширяемой функциональностью. Пакет построен на клиент-серверной архитектуре с возможностью функционирования, как в локальных сетях, так и в Интернете. Прием и передача данных и сообщений на основе стандартов OPC встроена в ядро пакета. Максимальная поддержка всех стандартов (XML, HTML, ODBC, OLE, COM/DCOM, ActiveX и др.) и открытые описания интерфейсов и форматов данных обеспечивают все необходимые возможности для стыковки с внешними программами и системами.

Основные преимущества пакета:

- Единая среда разработки АСУ ТП.
- Раздельное конфигурирование структуры АСУ ТП и логической структуры объекта.

- Открытость и следование стандартам.
- Интуитивная легкость освоения.
- Удобство инструментария.
- Удобство методики разработки.
- Мощная трехмерная графика и мультимедиа.
- Неограниченная гибкость вычислительных возможностей.
- Объектный подход.

Функциональные возможности:

- Пользовательский интерфейс пакета. Пользовательский интерфейс MasterScada построен на идеологии «все в одном». Все модули расширения встроены в общую оболочку. Пользователь всегда работает с простым единым внешним видом программы, состоящим из древо-видного проекта, палитры библиотечных элементов и окна редактирования документов и свойств. В зависимости от типа настраиваемого свойства или редактируемого документа в окне редактирования открывается страница настройки нужного свойства, либо необходимый встроенный или внешний редактор.

- Проект состоит из двух разделов: Система и Объект. Раздел Система описывает техническую структуру реализуемой системы. Раздел Объект описывает иерархическую структуру контролируемого технологического объекта свойства и документы каждого объекта.

Окно проекта состоит из четырех основных частей (рис. 26):

1. Древа системы, в котором отображены элементы конфигурации такие, как компьютеры, OPC-серверы, и т. д.

2. Древа объектов, включающее в себя объекты, переменные, группы переменных, функциональные элементы.

3. Страницы свойств элементов, на которых производятся все необходимые настройки элементов.

4. Палитры элементов, из которой берутся библиотечные объекты, функциональные блоки и т. д.

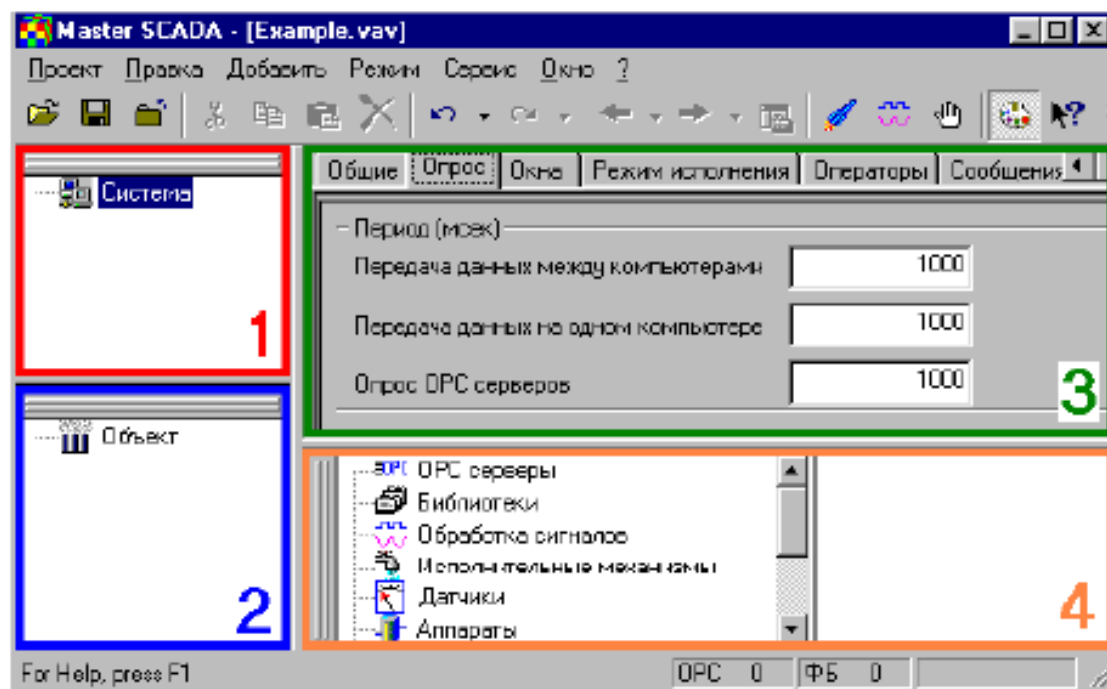


Рис. 26. Окно проекта

Размеры дерева системы, дерева объектов и палитры можно менять и они могут располагаться в любом месте экрана. Страницы свойств занимают все оставшееся пространство.

Проект состоит из ограниченного набора многофункциональных элементов.

Набор элементов дерева Система следующий:

- Система (корневой элемент). Используется для общих настроек проекта (периоды опроса, типы мнемосхем и документов объектов, шкалы приборов, категории сообщений, настройки журналов, права доступа операторов и т. п.).
 - Компьютер. Используется для коррекции тех настроек проекта, которые для данного компьютера отличаются от общих, а также стартовой мнемосхемы, списка операторов, имеющих доступ к этому рабочему месту, и т. п.
 - OPC-сервер. Используется для настройки связи с контроллерами.
 - Группы OPC-переменных. Формируются на основании информации о группировании переменных, полученной из OPC-сервера. Можно создавать дополнительные группы. Используются для группового задания настроек опроса переменных.

- **ОРС-переменные.** Используются для связи с переменными контроллера. Наследуют настройки, заданные при конфигурировании переменной в ОРС-сервере. В зависимости от заданного в ОРС-сервере разрешения на чтение-запись подразделяются на входы, выходы и входы-выходы.

Набор элементов дерева **Объект**

- **Объект (корневой элемент).** Используется для задания общих настроек, наследуемых другими объектами (периоды обработки данных и т. п.).

- **Объект (элемент иерархии).** Используется для задания перечня и содержимого относящихся к нему документов (мнемосхем, трендов, журналов сообщений, рапортов, архивов, расписаний действий и т. п.). Подчиненные объекты наследуют настройки родительского объекта. Объект всегда позиционирован на одном из компьютеров системы. Тем самым задается, что эта рабочая станция используется для обработки данных объекта и хранения его первичного архива, а также определяет перечень операторов, имеющих к нему доступ.

- **Функциональный блок.** Библиотечный объект, предназначенный для обработки данных. Имеет функцию, входы, выходы, параметры настройки, сообщениями.

- **Визуальный функциональный блок.** Функциональный блок, имеющий визуальное (в виде динамического элемента мнемосхемы) представление. Визуальный функциональный блок можно путем перетаскивания вставлять в документы объекта.

- **Группа переменных.** Используется для задания общих настроек, наследуемых входящими в нее переменными и группами переменных (периоды обработки данных).

- **Значение.** Переменная для отображения измеренного значения. Как и все другие виды переменных имеет шкалу, единицу измерения, встроенный контроль границ и скорости изменения с формированием сообщений и изменением цвета отображения. При перетаскивании в мнемосхемы и окна может быть вставлена, как в виде числового значения, так и в виде щитового прибора выбранного типа.

- **Команда.** Переменная для передачи введенного значения от органа управления мнемосхемы или поля ввода иных документов. При перетаскивании в мнемосхемы и окна может быть вставлена, как в виде изменяемого числового значения, так и в виде щитового органа управления выбранного типа.

- **Расчет.** Переменная, значение которой формируется в результате вычисления заданной пользователем формулы (содержащей арифмети-

ческие и логические выражения любой сложности с включением библиотечных функций, в том числе для работы с архивами).

- Событие. Отличается от расчета логическим результатом вычисления, а также возможностью формирования сообщения и выполнения заданного перечня действий в момент перехода значения из 0 в 1 (из отключенного во включенное состояние).

Существует три возможных варианта работы в режиме исполнения.

- Рабочий режим. Это основной режим исполнения. В рабочем режиме должен быть осуществлен переход к нему на всех компьютерах системы. Программа производит реальное управление технологическим процессом.

- Режим отладки. Этот режим предназначен для отладки проекта на одном компьютере. Независимо от того, сколько компьютеров находится в дереве системы, все объекты, функциональные блоки, OPC-серверы создаются на текущем компьютере и все действия производятся на нем.

- Режим имитации. В этом режиме на все входы, не имеющие связей, вместо констант будет подаваться имитация в соответствии с настройками системы.

Пакет программ MasterSCADA имеет ряд возможностей, направленных на повышение надежности работы (восстановление проекта после сбоя, горячий рестарт, копирование архива, журналирование внутренних событий). В режиме разработки используется безопасная вставка Active-X элементов сторонних разработчиков (перехват наиболее типичных ошибок). Список этих возможностей постоянно расширяется [8, 9].

4.3. Методика разработки программ визуализации процессов контроля, регулирования и сигнализации

Программа визуализации регулирования REG использует переменные программы PID_ST, т. е. переменные SCADA-системы ссылаются на переменные ISaGRAF. При этом связь переменных осуществляется через OPC-сервер, предназначенный для сопряжения ISaGRAF с Master Scada. Поэтому перед началом создания программы визуализации необходимо настроить и запустить OPC-сервер.

4.3.1. Создание мнемосхемы

Запустите программу MasterScada (ПУСК/все программы/MasterScada/MasterScada). В результате появится окно **Создание проекта** (рис. 27), в котором введите имя своего проекта. Затем введите пароль доступа к вашему

проекту. Если ничего не ввести, то при новом запуске ваш проект не будет требовать пароль.

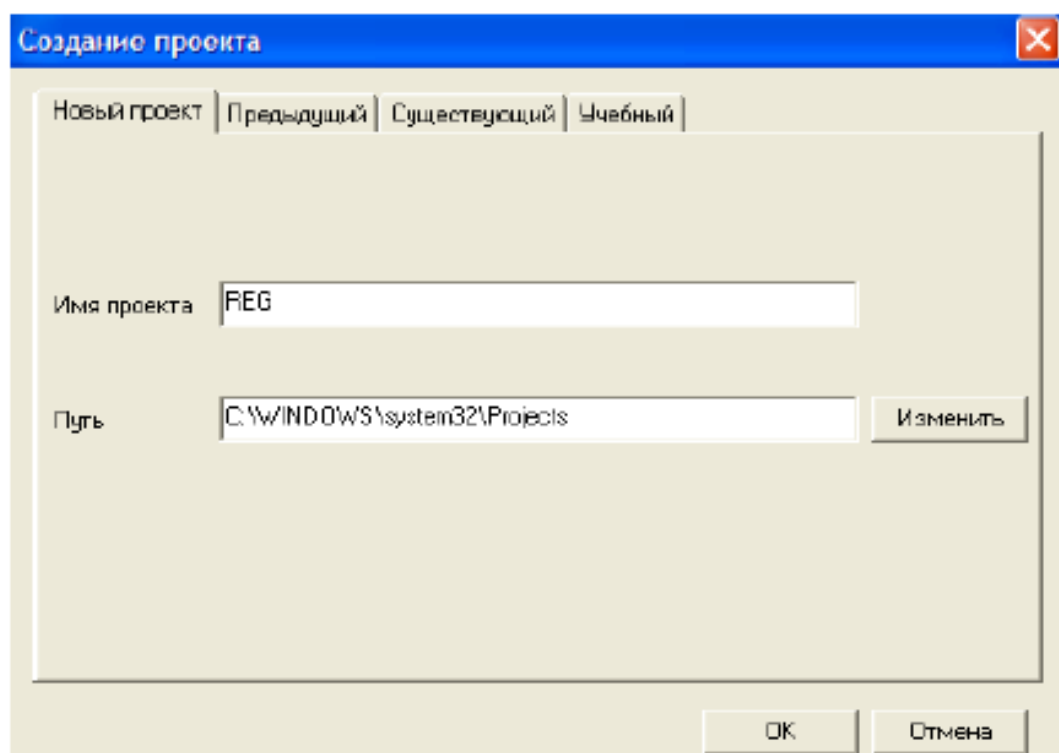


Рис. 27. Окно создания проекта

В итоге появится окно проекта (рис. 28).

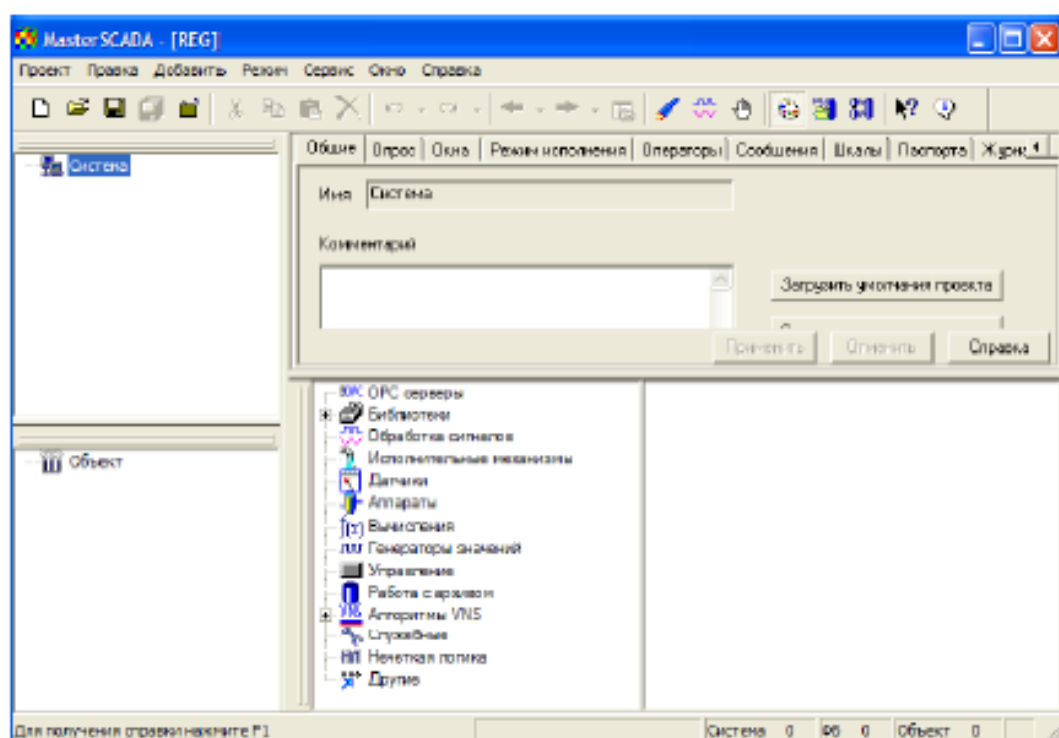


Рис. 28. Окно проекта

Теперь создадим соединение между контроллером и средой ISaGRAF. Для этого выделите объект «Система» в дереве системы и щелкните по ней правой клавишей мыши, выберите **ВСТАВИТЬ/КОМПЬЮТЕР**. В странице свойств элемента в поле **Имя** введите имя используемого компьютера (например **titan2**), не забыв нажать на кнопку **ПРИМЕНИТЬ** для сохранения результата.

Теперь нажав ПК на **titan2**, добавьте OPC-сервер (**ВСТАВИТЬ/OPC-СЕРВЕР**).



Доступ к данным OPC-серверов осуществляется через OPC-переменные в MasterScada. Поэтому необходимо добавить OPC-переменные (**ВСТАВИТЬ/OPC ПЕРЕМЕННЫЕ**).

В появившемся окне **Свойства: выбор переменных** поставьте галочку, выделяя все переменные. В результате все переменные ISaGRAF, используемые при написании технологической программы пользователя будут доступны в MasterScada.

Число мнемосхем в проекте не ограничено, но число мнемосхем объекта ограничено – объект может иметь по одной мнемосхеме каждого разрешенного в проекте типа. Именно поэтому для создания нескольких мнемосхем необходимо создавать несколько объектов, у каждого из которых будет своя мнемосхема.

Основной способ создания мнемосхем – перетаскивание из дерева проекта объектов, визуальных функциональных блоков и переменных, уже обладающих всей необходимой функциональностью (изображение, динамизации, окна управления и т. п.).

4.3.2. Создание графика

Для того чтобы создать графическую часть проекта, необходимо работать с деревом объекта. Выделите **Объект** и щелкните по нему правой клавишей мыши, выберите **ВСТАВИТЬ/ОБЪЕКТ**. В странице свойств **ОБЪЕКТА 1** выберите компьютер **titan2**, здесь же можно поменять имя объекта. Далее в палитре инструментов выберите  **Датчики**, щелкните левой клавишей мышки по объекту  **График** и перетащите его в **Объект 1**. В итоге дерево объектов будет выглядеть следующим образом (рис. 29).

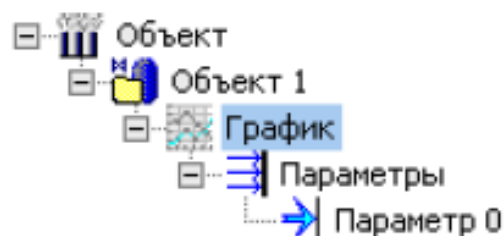


Рис. 29. Дерево объектов

На странице свойств элемента «График» войдите на закладку **Настройки** и укажите **ЧИСЛО ПАРАМЕТРОВ**. Нажмите кнопку **ПРИМЕНИТЬ**. Затем из дерева системы выберите переменную **t1_XIN** и перетащите ее

в **Параметр 0**. В результате, у этого параметра появится розовая черта, это и означает, что переменные связаны. Таким же образом свяжите остальные переменные с параметрами графика.

Для перехода на мнемосхему необходимо выделить **Объект 1** и на странице свойств элемента перейти на закладку **Окна**, где по умолчанию должна быть выбрана мнемосхема, нажать на кнопку **РЕДАКТИРОВАТЬ**. Впоследствии переход на мнемосхему будет осуществляться таким образом: выделите **Объект 1** и щелкните по нему ПК, выберите **ПЕРЕЙТИ НА/МНЕМОСХЕМА**.

Выделите объект **График** и перетащите его на мнемосхему (рис. 30).

В правом нижнем углу редактора мнемосхем отображаются свойства элементов мнемосхемы, которые позволяют менять, например, цвет, размер, положение элементов и т. д. Установите у графика максимальное и минимальное значения по осям. Ввести ряд изменений также можно посредством окна **СВОЙСТВ** элемента, которое появится, если выделить элемент, нажать ПК и выбрать **СВОЙСТВА**.

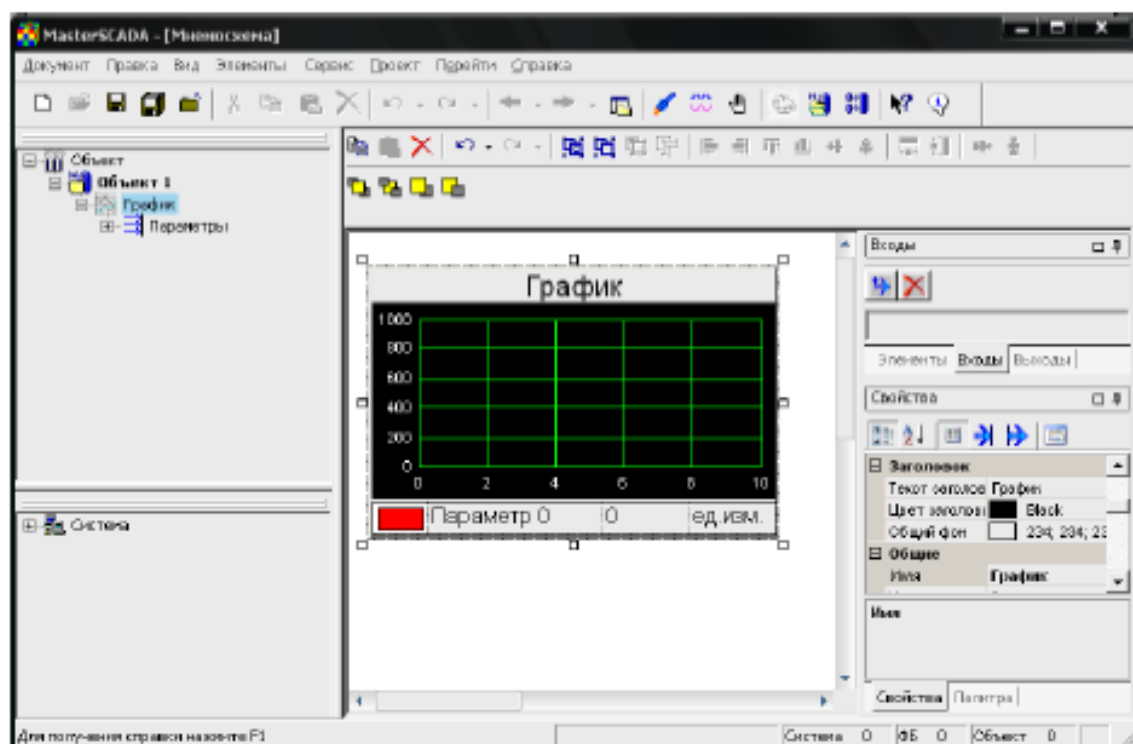




Рис. 30. График

4.3.3. Создание команд

В дереве системы выделите **Объект 1**, щелкните ПК и выберите **ВСТАВИТЬ/КОМАНДА**. Команды позволяют изменять значения переменных в режиме работы. Перетащите необходимую переменную, например **t1_KP**

(коэффициент пропорциональности ПИД-регулятора), в  Команда 1. Аналогичным образом создайте команды изменения остальных параметров. Перетащите команды из дерева объектов на мнемосхему. Имя команды можно изменить, используя закладку **Общие** на странице свойств элемента.

4.3.4. Вставка значений

В дереве системы выделите **Объект 1**, щелкните ПК и выберите **ВСТАВИТЬ/ЗНАЧЕНИЕ**. Значение используется для визуализации данных. Имя значения можно изменить, используя закладку **Общие** на странице свойств элемента. Создадим значение с именем **Индикатор**. Теперь перетащим необходимую переменную в  Индикатор, как описывается в следующем пункте. Затем перетащим значения из дерева объектов на мнемосхему ПК мыши, выбрав из контекстного меню вид индикатора, ниже в п. 4.3.8 описано более подробно их реализация. В свойствах можно изменить вид данного индикатора.

Также значения можно помещать непосредственно на мнемосхему из дерева системы, тогда это будет цифровое представление данной переменной (рис. 31).

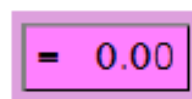


Рис. 31. Значение переменной

4.3.5. Создание кнопок

Для того чтобы создать кнопку, необходимо перейти на палитру инструментов, выбрать **УПРАВЛЕНИЕ/КНОПКА** и перетащить ее в **Объект 1** (рис. 32). Для начала необходимо изменить название данной кнопки на **Включить** (закладка **Общие** страницы свойств элемента). На странице свойств кнопки укажите **Дискретные значения – Вкл/**. Перетащите кнопку на мнемосхему. Откройте окно ее **СВОЙСТВ** ПК мыши, поставьте галочку напротив **Кнопка с фиксацией**. По желанию можете изменить размер, шрифт или цвет, а также можно добавить изображение на кнопку.

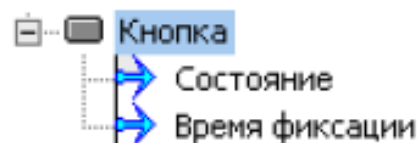


Рис. 32. Кнопка

4.3.6. Создания элементов мультимедиа

В данной работе необходимо создание элемента мультимедиа (мультфильм), а именно прообраз регулятора, который начинает работать при включении ручного режима. Для этого вставим **МУЛЬТФИЛЬМ**

В ПРЯМОУГОЛЬНИКЕ из палитры на мнемосхеме на вкладке **Мультимедиа**. Теперь динамизируем этот элемент. Для этого выделим элемент и перейдем на панель **Входы**, затем выделим **Вход** и снова перейдем на панель **Свойства**, где появились общие свойства. Сначала выберем источник – **Объект**, и затем перетащим переменную **Состояние** кнопки **Включить** из дерева объектов в поле **Объект**. Переменная добавляется в список входов или выходов мнемосхемы, если ее не было в нем до этого.

4.3.7. Организация сигнализации

Для организации сигнализации по ограничению входного сигнала на максимальное и минимальное значения, необходимо добавить в **Объект 1** из палитры инструментов **ДАТЧИКИ/ИНДИКАТОР СОСТОЯНИЯ**. Затем в странице свойств этого объекта на закладке **Настройки** укажите число параметров – 1. На закладке **Общие** измените имя индикатора, на закладке **Цвета** определите цвета индикатора, которые соответствовали бы начальному состоянию (по умолчанию – например, зеленый) и состоянию, при котором входной сигнал превышал бы заданное максимальное значение (красный). Вход индикатора свяжите с переменной **t1_SIGMAX**, которая отвечает за мигающую сигнализацию по верхнему пределу. Аналогично создайте индикатор нарушения ограничения на минимальное значение входного сигнала (**t1_SIGMIN**). Перетащите индикаторы на мнемосхему.

Аналогичным образом формируется проверка данных на достоверность (переменные **t1_MIGMAX** и **t1_MIGMIN**, рис. 33).

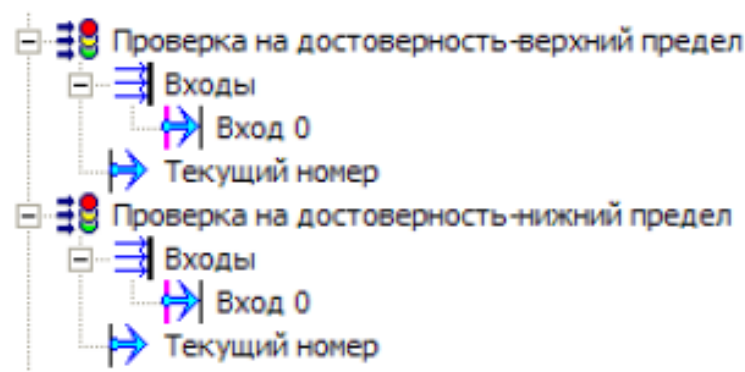


Рис. 33. Индикаторы состояния

4.3.8. Создание дополнительных элементов управления

Для качественного контроля программой визуализации используем следующие элементы управления: стрелочный прибор, индикатор значения, индикатор мнемонический, графический, а также задатчик.

Стрелочный прибор

Стрелочный прибор предназначен для отображения значения аналоговой переменной. Для осуществления привязки стрелочного прибора к источнику данных, позволяющих отслеживать значение некоторой переменной в мнемосхеме, перетащим интересующую аналоговую переменную (за исключением команд) в мнемосхему ПК мыши, и в раскрываемся контекстном меню выберем **СТРЕЛОЧНЫЙ ПРИБОР**. Существует другой способ привязки, воспользовавшись палитрой элементов графического редактора. Но при работе графического редактора MasterGraph в составе пакета программ MasterScada, первый способ является наиболее предпочтительным.

Также возможно изменить свойства стрелочного прибора, вызвав контекстное меню ПК мыши. Здесь представлен стандартный набор свойств: вид, шкала, значение, аварийные зоны, граница, цвет, шрифт, связь, подсказка. Есть возможность изменения заголовка стрелочного прибора, изменение единицы измерения, выбор штриховки, длины шкалы, типа значения, выбор цветовой сигнализации стрелочного прибора, цветового решения прибора и многого другого (рис. 34).

Индикатор значения

Индикатор значения предназначен для отображения аналогового значения переменной в виде вертикального или горизонтального столбика. Процедура привязки индикатора к источнику данных осуществляется аналогичным способом, описанным выше, только выбирается соответственно **ИНДИКАТОР**. Обладает теми же свойствами (рис. 35).

Индикатор мнемонический и графический

Индикаторы предназначены для индикации состояния дискретных переменных. Индикаторы не представлены ни в палитре функциональных блоков MasterScada, ни в палитре элементов графического редактора MasterGraph. Единственный способ, с помощью которого можно создать



Рис. 34. Стрелочный прибор

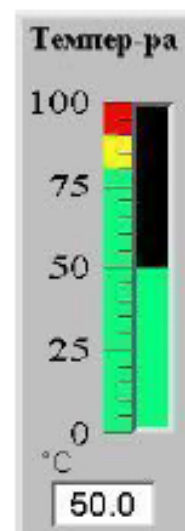


Рис. 35. Индикатор значения

данный элемент управления на мнемосхеме: перетащить дискретную переменную (значение переменной из п. 4.3.4) в мнемосхему ПК мыши, и в раскрывающемся контекстном меню выбрать **Индикатор мнемонический** или **графический**, для того чтобы наблюдать состояние этой переменной.

Существует возможность задания цвета отображения данного индикатора в зависимости от состояния связанной дискретной переменной. Эти и другие настройки внешнего вида произведем на странице свойств **Вид**, непосредственно на мнемосхеме.

Изображение индикаторов представлено на рис. 36 и 37.



Рис. 36. Индикатор графический



Рис. 37. Индикатор мнемонический

Задатчик

Основное назначение задатчика – быть источником данных для аналоговой команды в дереве объектов. Кроме того, задатчик может быть использован и для изменения свойств других элементов мнемосхем. В поле задатчика может отображаться градуированная шкала, поле цифрового значения переменной, единица измерения.

Привязка задатчика и выхода команды схожа с привязкой индикатора или стрелочного прибора к источнику данных, описанная выше. Разница заключается лишь в том, что теперь необходимо использовать не аналоговую переменную, а команду. Также существует два варианта:

1. Перетащить нужную аналоговую команду в мнемосхему правой кнопкой мыши, и в раскрывающемся контекстном меню выбрать **Задатчик**.

2. Создать **Задатчик** в мнемосхеме, воспользовавшись палитрой элементов графического редактора. Динамизировать свойство задатчика **Значение по выходу**, связанному с Командой.

Свойства задатчика полностью схожи со свойствами вышеописанных элементов управления. Внешний вид задатчика представлен на рис. 38.

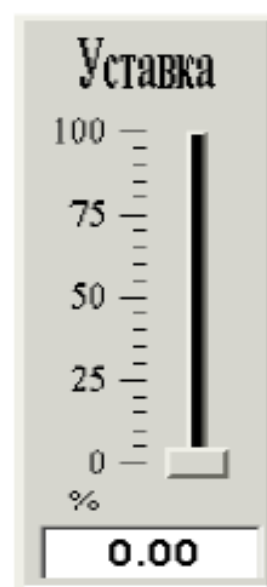


Рис. 38. Задатчик

4.3.9. Создание таблицы

Таблица предназначена для отображения значений переменных в виде таблицы параметров. Для создания таблицы необходимо перейти на палитру инструментов, выбрать **УПРАВЛЕНИЕ/ТАБЛИЦА**, перенести в **Объект 1**.

На закладке **Общие** страницы свойств элементов меняем название таблицы, добавляем комментарии, меняем число входов и выходов таблицы на закладке **Параметры**. Таблицу перетаскиваем на мнемосхему, единицы измерения, формат значения и аварийные зоны наследуются от настроек переменных привязанных ко входам таблицы в дереве объекта. Свойства таблицы можно вызвать ПК мыши и изменить оформление таблицы (рис. 39).

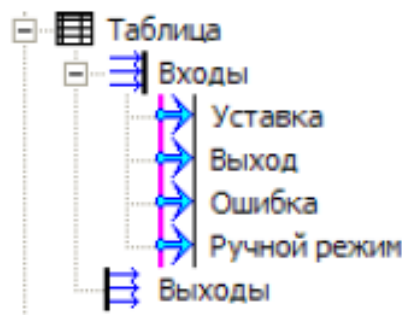


Рис. 39. Таблица

4.3.10. Создание графических примитивов

Для того чтобы создать графические примитивы на мнемосхеме, будь-то статический текст или графические элементы (различные прямоугольники, эллипсы, линии), необходимо открыть вкладку **Палитра**, расположенную в правом нижнем углу редактора мнемосхем, затем выбрать **ГРАФИЧЕСКИЕ ПРИМИТИВЫ/ТЕКСТ** или **ПРЯМОУГОЛЬНИК** и щелкнуть по свободному месту на мнемосхеме. В **СВОЙСТВАХ** объекта можно изменить надпись, шрифт текста, цвет, заливку, вид и многое другое.

4.3.11. Создание главной мнемосхемы, переходов между мнемосхемами и осуществление закрытия мнемосхемы

После создания всех мнемосхем создаем главную мнемосхему, на которой будут представлены все мнемосхемы. Особенность ее будет заключаться в наличии кнопок с переходами на другие мнемосхемы, для этого необходимо просто перейти на мнемосхему и перенести туда все объекты. Они сразу определяются как кнопочки со сжатым изображением мнемосхем этих объектов. Вид кнопочки можно изменить, нажав ПК мыши, в свойствах можно добавить надпись, изменить цвет,

шрифт, ширину рамки, а главное есть возможность изменить вид кнопки на изображение **Дальше**, **Назад**. Для этого необходимо перенести курсор со **Сжатого** изображения на просто **Изображение** и из контекстного меню выбрать директорию нахождения изображения, которое должно стать лицевым на кнопках. Кроме того, необходимо изменить курсор с **Исходного размера** на **Растягивать** для корректировки размера кнопочек. Именно так создаются кнопки переходов на всех мнемосхемах.

Также на главной мнемосхеме должна присутствовать кнопочка, осуществляющая закрытие мнемосхем для этого в данном объекте создается, при помощи ПК мыши **ВСТАВИТЬ**, событие и команда, позволяющие осуществить закрытие мнемосхемы.

Произведя все выше описанные процедуры, получаем программу визуализации процесса контроля, регулирования и сигнализации. Но программа будет готова к запуску лишь, после того как, вернувшись на свойства **Объекта 1**, зайдём на вкладку **Окна** и поставим галочку напротив **Стартовая мнемосхема**.

ЛИТЕРАТУРА

1. MasterSCADA [Электронный ресурс] // URL: <https://ru.wikipedia.org/wiki/MasterSCADA> (дата обращения: 05.04.2018).
2. OpenSCADAWiki: Doc/ОписаниеПрограммы [Электронный ресурс] // URL: <http://wiki.oscada.org/Doc/OpisanieProgrammy/#1>