

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

В.Н. Скороспешкин, М.В. Скороспешкин

АВТОМАТИЗИРОВАННЫЕ ИНФОРМАЦИОННО- УПРАВЛЯЮЩИЕ СИСТЕМЫ

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

Издательство
Томского политехнического университета
2018

УДК 681.51 (075.8)
ББК 32.965-016.5я73
С44

Скороспешкин В.Н., Скороспешкин М.В.

С44 Автоматизированные информационно-управляющие системы: учебно-методическое пособие / В.Н. Скороспешкин, М.В. Скороспешкин; Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2018. – 106 с.

В учебно-методическом пособии изложены требования к выполнению расчетно-пояснительной записки и приведены методические указания по выполнению курсового проекта по дисциплине «Автоматизированные информационно-управляющие системы». Рассмотрены состав и технические характеристики промышленных микропроцессорных контроллеров КРОСС-500 и ТРАССА-500. Описана методология разработки автоматизированных систем управления технологическими процессами с использованием SCADA технологий, рассмотрены примеры решений для основных разделов курсового проекта.

Пособие подготовлено в ОАР ИШИТР ТПУ и предназначено для студентов, обучающихся по направлению 15.03.04 «Автоматизация технологических процессов и производств»

УДК 681.51 (075.8)

ББК 32.965-016.5я73

Рецензенты

Доктор технических наук, профессор ТПУ

В.Л. Ким

Директор ООО «Сибпромкомплект»

А.Ф. Андросов

© ФГАОУ ВО НИ ТПУ, 2018

© Скороспешкин В.Н.,
Скороспешкин М.В., 2018

© Оформление. Издательство Томского
политехнического университета, 2018

ГЛОССАРИЙ

Термин	Определение
Технологический процесс (ТП)	Последовательность технологических операций, необходимых для выполнения определенного вида работ, производства сырья, продукта, энергии и т.д.
IEC 1131-3	Стандарт международной электротехнической комиссии определяющий языки программирования микропроцессорных контроллеров
Интерфейс (RS-485, RS-232, Ethernet)	Совокупность программно-аппаратных средств для обеспечения взаимодействия между микропроцессорными устройствами
OPC-сервер (англ. Ole For Process Control)	Средство взаимодействия программного обеспечения автоматизированных систем управления с техническими средствами
SCADA (англ. Supervisory Control And Data Acquisition-диспетчерское управление и сбор данных)	Под термином SCADA понимают инструментальную программу, предназначенную для разработки программного обеспечения операторских (диспетчерских) станций автоматизированных информационно-управляющих систем, работающих в режиме реального времени
ФЮРА. 425280	ФЮРА – это код организации разработчика проекта (ТПУ); 425280 – это код классификационной характеристики проектной продукции по ГОСТ 3.1201-85. Этот код означает проектирование распределенного автоматизированного управления
Функции АСУ ТП	Это совокупность действий системы, направленных на достижение частных целей управления
ISaGRAF	Программный пакет, предназначенный для программирования промышленных контроллеров

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

FBD	Язык функциональных блоков
LD	Язык релейной логики
PLC (Programmable Logic Controller)	Программируемый логический контроллер
ST	Язык структурированного текста
АВК-6	Аналоговый вычислительный комплекс
АИУС	Автоматизированная информационно-управляющая система
АРМ	Автоматизированное рабочее место
АСУ ТП	Автоматизированная система управления технологическим процессом
БЦП	Блок центрального процессора

КП	Курсовой проект
МК1	Полевой микропроцессорный контроллер
МВВ	Модули ввода/вывода аналоговых и дискретных сигналов
МВВ	Модули ввода-вывода
ПО	Программное обеспечение
ППО	Прикладное программное обеспечение
РПО	Резидентное программное обеспечение
ТД	Текстовый документ
ТПП	Технологическая программа пользователя
ФБ	Функциональный блок

ПРИМЕРНЫЕ ТЕМЫ КУРСОВЫХ ПРОЕКТОВ

1. Программное обеспечение (ПО) автоматизированной системы контроля и регулирования расхода пара.
2. ПО автоматизированной системы контроля и регулирования уровня нефти.
3. ПО автоматизированной системы контроля и регулирования давления нефти.
4. ПО автоматизированной системы контроля и регулирования температуры технической воды.
5. ПО автоматизированной системы контроля и регулирования расхода сыпучего вещества.
6. ПО автоматизированной системы контроля и регулирования давления масла магистрального насосного агрегата.
7. ПО автоматизированной системы контроля и регулирования уровня сыпучего вещества.
8. ПО автоматизированной системы контроля и регулирования расхода технической воды.

Разделы курсового проекта, входящие в задание

1. Описание промышленного микропроцессорного контроллера, определенного заданием на курсовое проектирование, и типовых структур систем автоматизации, выполненных на базе данного контроллера.
2. Описание состава и функциональных возможностей программного пакета ISaGRAF и языков программирования стандарта МЭК 1131-3.

3. Описание состава и функциональных возможностей SCADA-пакета, определенного заданием.
4. Описание параметров и процедуры настройки OPC – сервера используемого контроллера.
5. Выбор языка программирования и разработка программы для микропроцессорного контроллера, обеспечивающей выполнение функций, определенных заданием.
6. Разработка программы на базе SCADA-пакета для операторской станции, обеспечивающую визуализацию процесса контроля, регулирования и сигнализации. Минимальный состав выполняемых функций: цифровые значения регулируемого параметра, задания, ошибки регулирования, графики и тренд реального времени регулируемого параметра, задания и ошибки, изменение цвета выбранной формы при выходе параметра за установленные пределы, команды по изменению задания и параметров настройки регулятора. Информация должна быть представлена как минимум в виде двух мнемосхем с реализацией переходов между ними.
7. Настройка OPC-сервера и проверка работоспособности разработанных программ на учебном лабораторном стенде. При проверке работоспособности программ модель объекта управления набирается на аналогово-вычислительном комплексе АВК-6.

Используемые промышленные микропроцессорные контроллеры: КРОСС-500, ТРАССА-500.

Используемые SCADA-пакеты: MasterScada, TRACE MODE.

Состав функций, выполняемых контроллером: проверка достоверности введенной в контроллер информации, регулирование заданного параметра, сигнализация при выходе параметра за установленные верхний и нижний пределы, фильтрация входных сигналов, аналитическая градуировка датчиков. Сигнализация должна быть реализована путем вывода дискретных сигналов.

Языки программирования контроллеров: FBD, ST, LD.

ВВЕДЕНИЕ

В настоящее время трудно представить какой-либо технологический процесс без автоматической или автоматизированной системы управления, поскольку наличие данной системы управления обеспечивает качественный контроль и управление технологическим процессом, а также оптимизацию экономических затрат.

Современные автоматизированные системы управления технологическими процессами строятся на базе промышленных микропроцессорных контроллеров и компьютеров операторских станций

Разработка автоматизированной системы представляет собой достаточно сложную и трудоемкую работу, требующую высокую квалификацию специалистов, занимающихся проектированием и наладкой таких систем. При этом современный специалист должен иметь не только знания в области технического обеспечения АСУ, но и обладать навыками программирования промышленных микропроцессорных контроллеров и создания, с помощью различных SCADA-пакетов, программ визуализации хода технологического процесса, оперативного дистанционного управления, архивирования информации и подготовки рапортов и отчетов.

В ходе выполнения курсового проекта студент должен на базе знаний, полученных после изучения языков программирования микропроцессорных контроллеров и SCADA-пакетов, осуществить разработку программного обеспечения АСУ в соответствии с техническим заданием и проверить работоспособность программ. При этом студент должен опираться как на собственные знания, полученные в ходе изучения учебного курса «Автоматизированные информационно-управляющие системы», так и на литературные и интернет-источники.

Задача данного пособия – предоставить студентам методический материал по разработке программного обеспечения автоматизированных информационно-управляющих систем, построенных на базе отечественных микропроцессорных контроллеров (PLC) КРОСС-500 и ТРАССА-500 [1,2,3].

В этой связи данный курсовой проект предназначен для приобретения теоретических знаний в области автоматизированного управления, а именно в изучении промышленных микропроцессорных контроллеров КРОСС-500 и ТРАССА-500, программного пакета ISaGRAF [3,9], программных пакетов MasterScada [4,8] и TRACE MODE, предназначенных для оперативного управления и визуализации процесса контроля, регулирования и сигнализации. Наряду с этим, одной из важных целей данной работы является получение практических навыков составления программ контроля, регулирования и сигнализации, реализуемых промышленными микропроцессорными контроллерами, а также создания программ визуализации процесса контроля, регулирования и сигнализации в пакетах MasterScada и TRACE MODE.

1 ОБЩИЕ ТРЕБОВАНИЯ ПО КУРСОВОМУ ПРОЕКТИРОВАНИЮ ПО КУРСУ «АВТОМАТИЗИРОВАННЫЕ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЕ СИСТЕМЫ»

1.1 Цель и задачи курсового проектирования

Учебным планом специальности 220201 «Управление и информатика в технических системах» предусмотрено курсовое проектирование по дисциплине «Автоматизированные информационно-управляющие системы». Основными целями курсового проектирования являются систематизация и углубление знаний в области разработки программного обеспечения АСУ, приобретение умений выполнения инженерных проектных задач в области автоматизации технологических процессов. Для достижения поставленной цели должны найти отражение следующие элементы обучения.

- *Умение* выполнять проектные работы по созданию программного обеспечения автоматизированных информационно-управляющих систем, выбирать необходимые алгоритмы обработки информации, разрабатывать программы контроля и визуализации состояния технологического процесса, регулирования, сигнализации, блокировки и защиты на языках программирования контроллеров в соответствии с международным стандартом IEC 1131-3 с применением SCADA.
- *Знание* типовых структур АСУ ТП, протоколов микропроцессорных средств автоматизации и интерфейсов передачи данных, а также OPC технологий обмена данными между контроллерами и SCADA.
- *Приобретение* компетенций:
 - применять современные программные средства для проектирования автоматизированных информационно-управляющих систем;
 - разрабатывать технические предложения, документацию и программы;
 - осуществлять практическую работу с промышленными микропроцессорными контроллерами в процессе проверки работоспособности созданного программного обеспечения.

Задачей курсового проектирования является разработка эскизного проекта программного обеспечения автоматизированной информационно-управляющей системы. В качестве информационно-управляющей системы выбрана АСУ ТП, имеющая два уровня. Нижний уровень построен на базе промышленных микропроцессорных контроллеров, а верхний (диспетчерский) уровень на базе компьютера и SCADA. Разрабатываемое программное обеспечение включает в себя программу для микропроцессорного контроллера и программу для операторской станции, составленную с применением SCADA.

Курсовой проект выполняется на основе индивидуального задания, которое выдается преподавателем. Курсовой проект должен включать в себя

пояснительную записку (ПЗ), графическую часть, оформленную в виде альбома схем, и результаты проектирования на электронном носителе. Пояснительная записка и графическая часть проекта выполняются в соответствии со стандартом ТПУ СТО ТПУ 2.5.01-2006 «Работы выпускные квалификационные, проекты и работы курсовые. Структура и правила оформления».

Отдельные части курсового проекта должны быть представлены в логической последовательности и взаимосвязи. Работа должна иметь иллюстрации рисунками, таблицами, графиками, схемами. В тексте должны присутствовать ссылки на иллюстрации, таблицы и литературу.

1.2 Структура расчетно-пояснительной записки

Объем пояснительной записки должен составлять 30÷40 страниц печатного текста. Она должна содержать краткое описание используемых технических средств автоматизации и используемых информационных технологий и программных продуктов, а также расчеты с необходимыми пояснениями.

Структурными элементами пояснительной записки являются:

- текстовый документ (ТД);
- графический материал.

Текстовый документ должен включать структурные элементы в указанной ниже последовательности:

- титульный лист;
- задание;
- реферат;
- содержание;
- глоссарий;
- обозначения и сокращения;
- введение;
- основную часть;
- заключение;
- список использованных источников.

Титульный лист. Форма титульного листа приведена в приложении А.

Задание. Проект должен выполняться на основе индивидуального задания, которое содержит требуемые для решения поставленных задач исходные данные и соответствует ГОСТ 34.602–89 [11]. Задание оформляется по форме, приведенной в приложении Б.

Реферат. Должен быть представлен на отдельном листе. Рекомендуемый средний объем реферата 850 печатных знака. Объем реферата не должен превышать одной страницы.

Реферат должен содержать:

- сведения об объеме ТД, количестве иллюстраций, таблиц, приложений, количестве частей ТД, использованных источников, листов графического материала;

- перечень ключевых слов;
- текст реферата.

Перечень ключевых слов должен включать от 5 до 15 слов или словосочетаний из текста ТД, которые в наибольшей мере характеризуют его содержание и обеспечивают возможность информационного поиска.

Ключевые слова приводятся в именительном падеже и записываются строчными буквами в строку через запятые.

Текст реферата должен отражать оформленные в виде структурных частей:

- объект исследования или разработки;
- цель работы;
- метод или методологию проведения работы и применяемые технические средства;
- полученные результаты и их новизну;
- основные конструктивные, технологические и технико-эксплуатационные характеристики;
- степень внедрения;
- рекомендации или итоги внедрения результатов работы;
- область применения;
- прогнозные предположения о развитии объекта исследования (разработки);

Если в пояснительной записке нет сведений по какой-либо из перечисленных структурных частей реферата, то в тексте реферата она опускается, при этом последовательность изложения сохраняется. Изложение материала в реферате должно быть кратким и точным и соответствовать положениям ГОСТ 7.9. Пример составления реферата приведен в СТ ТПУ 2.5.01-2006.

Содержание. Включает введение, заголовки всех разделов, подразделов, пунктов заключение, список использованных источников и наименования приложений с указанием номеров страниц, с которых начинаются эти элементы.

Материалы, представляемые на технических носителях данных ЭВМ(CD), должны быть перечислены в содержании с указанием вида носителя, обозначения и наименования документов, имен и форматов соответствующих файлов, а также места расположения носителя в ПЗ.

Обозначения и сокращения. Если в пояснительной записке используется значительное количество (более пяти) обозначений и (или) сокращений, то оформляется структурный элемент “Обозначения и сокращения” содержащий перечень обозначений и сокращений, применяемых для данного проекта. Этот раздел должен следовать сразу за глоссарием. Запись обозначений и сокращений в этом элементе приводят в порядке их появления в тексте с необходимой расшифровкой и пояснениями. При этом:

- сокращения в виде аббревиатур приводят после термина и отделяют от него точкой с запятой;
- сокращения в виде краткой формы термина приводят после термина в скобках;

- условные обозначения приводят после термина, при этом, после условных обозначений величин приводят обозначения единиц величин, которые отделяют запятой.

В тексте документа допускается приводить без расшифровки общепринятые сокращения, установленные в национальных стандартах и правилами русской орфографии: ЭВМ, НИИ, АСУ, с. – страница, т. е. – то есть, т. д. — так далее; т. п. — тому подобное; и др. — и другие; в т. ч. — в том числе; пр. — прочие; т. к. — так как; с. — страница; г. — год; гг. — годы; мин. — минимальный; макс. — максимальный; шт. — штуки; св. — свыше; см. — смотри; включ. — включительно и др.

В тексте документа не допускается:

- применять сокращения слов, кроме установленных правилами русской орфографии, соответствующими государственными стандартами, а также в данном документе;
- сокращать обозначения единиц физических величин, если они употребляются без цифр, за исключением единиц физических величин в головках и боковиках таблиц и в расшифровках буквенных обозначений, входящих в формулы и рисунки.

В тексте следует избегать необоснованных (излишних) сокращений, которые могут затруднить пользование данным документом.

Сокращение русских слов и словосочетаний – по ГОСТ 7.12.

Перечень допускаемых сокращений, используемых в текстовой конструкторской документации, приведен в ГОСТ 2.316.

Введение. В элементе “Введение” указывают цель работы, область применения разрабатываемой проблемы, ее научное, техническое и практическое значение, экономическую целесообразность. Во введении следует:

- раскрыть актуальность вопросов темы;
- охарактеризовать проблему, к которой относится тема, изложить историю вопроса, дать оценку современного состояния теории и практики;
- привести характеристику отрасли промышленности, предприятия, технологического процесса, для которых разрабатывается проект;
- изложить задачи в области разработки проблемы, т. е. сформулировать задачи темы работы;
- перечислить методы и средства, с помощью которых будут решаться поставленные задачи;
- кратко изложить ожидаемые результаты, в том числе технико-экономическую целесообразность выполнения данной темы, либо экономическую эффективность;
- указать цель курсового проекта.

Рекомендуемый объем введения – 2–3 с.

Основная часть. Содержание основной части работы должно отвечать заданию. Наименования разделов основной части отражают выполнение задания. Содержание и объем основной части студент и руководитель формируют совместно и должно соответствовать уровню технического

предложения (эскизного проекта) в соответствии с ГОСТ 7.32. В основной части необходимо представить следующие разделы.

Промышленный микропроцессорный контроллер. В этом разделе необходимо дать описание назначения, состава, технических и функциональных характеристик необходимого промышленного микропроцессорного контроллера, определенного заданием на курсовое проектирование, и типовых структур систем автоматизации, выполненные на базе данного контроллера (объем раздела должен быть не более 8–10 с.)

Программный пакет IsaGRAF и языки программирования стандарта МЭК 1131-3. В этом разделе необходимо привести назначение и состав программного пакета IsaGRAF [9], перечислить и охарактеризовать языки программирования промышленных контроллеров, привести основные возможности пакета, последовательность составления, отладки и загрузки программы пользователя в контроллер (объем раздела должен быть не более 5–8 с.)

SCADA-пакет. В данном разделе необходимо дать пояснение понятию SCADA-пакета, привести назначение отдельных подсистем, перечислить функции, выполняемые этим инструментальным средством и указать наиболее популярные отечественные и зарубежные SCADA-пакета. А также привести описание применяемого SCADA-пакета, его особенности и порядок разработки программ с применением используемого программного пакета (объем раздела должен быть не более 5–8 с.).

Параметры настройки OPC – сервера контроллера. В данном разделе необходимо указать назначение OPC-сервера, способ настройки с описанием отдельных секций, а также привести рисунок, раскрывающий содержание файла CROSSOPC.INI(объем раздела должен быть не более 2–3 с.)

Выбор языка программирования и разработка программы для микропроцессорного контроллера.. В данном разделе необходимо обосновать выбор языка программирования, привести последовательность составления программы с иллюстрациями объявленных переменных и содержания словаря, описанием используемых функций и функциональных блоков, привести листинг программы, указать опции компилятора, вид канала связи с контроллером и его параметры. А также привести рисунок с указанием окна программы контроллера, находящегося в состоянии «работа» (объем раздела должен быть не более 6–7 с.).

Разработка программы на базе SCADA-пакета для операторской станции. В данном разделе необходимо проиллюстрировать процесс разработки программы, привести рисунки, содержащие экранные формы отображения информации и управления, описать процедуры динамизации экранных форм. При использовании пакета MasterScada привести дерево системы после добавления OPC переменных и дерево объектов. В данном разделе также должна быть описана процедура создания тренда.

Минимальный состав выполняемых функций: цифровые значения регулируемого параметра, задания, ошибки регулирования, графики и тренд реального времени регулируемого параметра, задания и ошибки, изменение цвета выбранной формы при выходе параметра за установленные пределы, команды по изменению задания и параметров настройки регулятора. Информация должна быть представлена как минимум в виде двух мнемосхем с реализацией переходов между ними. (объем раздела должен быть не более 6–8 с.)

Настройка OPC-сервера и проверка работоспособности разработанных программ. В данном разделе необходимо привести параметры настройки OPC-сервера, структурную схему лабораторного стенда, используемого для проверки работоспособности программ и методику проверки работоспособности. Работоспособность должна быть проиллюстрирована экранными формами, содержащими графики, тренды, световые и цифровые индикаторы, органы управления (объем раздела должен быть не более 5–6 с.).

Заключение должно содержать краткие выводы по результатам выполненной работы, оценку полноты решения поставленных задач, рекомендации по конкретному использованию результатов работы, ее научную и практическую значимость, а также указать полученные в результате выполнения курсового проекта знания и умения.

Список использованных источников. В список включают все источники, на которые имеются ссылки в КП. Источники в списке располагают и нумеруют в порядке их упоминания в тексте КП арабскими цифрами без точки. Сведения об источниках приводят в соответствии с требованиями ГОСТ 7.1 и ГОСТ 7.82. Примеры библиографических описаний источников приведены в СТО ТПУ 2.5.01-2006.

Альбом схем. Он должен включать в себя следующий перечень графических документов:

1. Дерево экранных форм.
2. SCADA- формы экранов мониторинга и управления операторской станции.

Графический материал должен быть органически увязан с содержанием работы, в наглядной форме иллюстрировать основные положения анализа, проектирования и использования информационных технологий [12].

Презентация проекта. Для защиты пояснительная записка должна быть представлена в формате *.pdf или .doc. Презентация КП должна быть подготовлена в виде слайдов формата MS Power Point, так как предусмотрена публичная защита проектных решений.

2. ПРОМЫШЛЕННЫЕ МИКРОПРОЦЕССОРНЫЕ КОНТРОЛЛЕРЫ КРОСС-500, ТРАССА-500

2.1 Состав, структура и АСУ ТП

АСУ ТП (автоматизированная система управления технологическим процессом) – комплекс программных и технических средств, предназначенный

для автоматизированного управления технологическим оборудованием предприятия. Основу АСУТП составляет программно-технический комплекс (ПТК), работающий в режиме реального времени выполняющий заданные функции АСУ ТП [6].

Функции АСУ ТП – это совокупность действий системы, направленных на достижение частных целей управления. Функции АСУ ТП можно подразделить на следующие.

- Управляющие функции. Результатами их выполнения являются выработка и реализация управляющих воздействий на управляемую систему.
- Информационные функции. Содержанием информационных функций является сбор, обработка и представление информации о состоянии системы оперативному персоналу или передача этой информации для последующей обработки.
- Вспомогательные функции обеспечивают решение внутрисистемных задач.

Для выполнения функций АСУ ТП необходимо взаимодействие следующих ее составных частей:

- техническое обеспечение, которое включает вычислительные и управляющие устройства, средства получения информации (датчики), средства преобразования, хранения, отображения и регистрации информации, устройства передачи сигналов и исполнительные устройства;
- программное обеспечение, состоящее из совокупности программ, необходимых для реализации функций АСУ ТП и обеспечения заданного функционирования комплекса технических средств;
- информационное обеспечение включает информацию, характеризующую состояние автоматизированного технологического комплекса;
- организационное обеспечение представляет собой совокупность описаний функциональных, технических и организационных структур, а также инструкций для оперативного персонала;
- оперативный персонал – это операторы-технологи, осуществляющие контроль за работой АСУ ТП;
- эксплуатационный персонал – это персонал, обеспечивающий эксплуатацию системы.

АСУ ТП представляет собой иерархическую систему, в которой можно выделить четыре основных уровня. На рисунке 1 представлена структура АСУ ТП с указанием отдельных уровней.



Рисунок 1. Структура АСУ ТП

Полевой уровень составляют датчики и исполнительные механизмы, которые устанавливаются на технологических объектах. Их назначение заключается в измерении параметров процесса, преобразовании их в соответствующий вид для дальнейшей передачи на более высокий уровень, а также в приеме управляющих сигналов и в выполнении соответствующих управляющих действий.

На нижнем уровне располагаются промышленные микропроцессорные контроллеры. На данном уровне выполняются информационные и управляющие функции, основными из которых являются:

- сбор информации, поступающей с полевого уровня и ее первичная обработка;
- технологическая и аварийная сигнализация
- регулирование;
- программное управление;
- аварийная защита;
- передача информации на верхний уровень.

Сетевой уровень обеспечивает обмен информацией между нижним и верхним уровнями. На данном уровне используются коммуникационный сервер, преобразователи интерфейсов, коммутаторы и т.д.

На верхнем уровне АСУ ТП размещены мощные компьютеры, выполняющие функции серверов баз данных и операторских станций и обеспечивающие анализ и хранение всей поступившей информации за любой заданный интервал времени, подготовку рапортов и отчетов о работе

технологического процесса, а также визуализацию информации и взаимодействие с оператором. Основой программного обеспечения высшего уровня являются SCADA-пакеты (Supervisory Control And Data Acquisition - диспетчерское управление и сбор данных) [4,8].

2.2 МИКРОПРОЦЕССОРНЫЙ КОНТРОЛЛЕР КРОСС-500

2.2.1 Назначение и область применения контроллера

Основное назначение контроллера – построение высокоэффективных (недорогих и надежных) систем автоматизации различных технологических объектов. Контроллер обеспечивает оптимальное соотношение производительность/стоимость одного управляющего или информационного канала, однородность аппаратуры автоматики на предприятии, уменьшает затраты на ЗИП, обучение персонала и т.п. [1,3,7,10].

Контроллер предназначен для общепромышленного применения в составе автоматизированных систем управления технологическими процессами в различных отраслях промышленности - энергетической, металлургической, химической, нефте- и газодобывающей, машиностроительной, сельскохозяйственной, пищевой и т.д. Контроллер может использоваться также в качестве автономного средства для управления объектами малой и средней сложности. Контроллер ориентирован на построение недорогих систем различной алгоритмической и информационной (число каналов) сложности:

- ✓ макро – система (до 1920 каналов);
- ✓ миди – система (64-128 каналов);
- ✓ мини – система (16-64 каналов).

Контроллер предназначен для решения следующих типовых задач автоматизации:

- сбор информации с датчиков различных типов и ее первичная обработка (фильтрация сигналов, линеаризация характеристик датчиков, проверка достоверности информации и т.д.);
- выдача управляющих воздействий на исполнительные органы различных типов;
- регулирование прямых и косвенных параметров по различным законам;
- логическое, программно-логическое управление технологическими агрегатами, автоматический пуск и останов технологического оборудования;
- математическая обработка информации по различным алгоритмам;
- регистрация и архивация технологических параметров;
- обмен данными с другими контроллерами в рамках контроллерной управляющей сети реального времени;
- аварийная, предупредительная и рабочая сигнализация, индикация значений прямых и косвенных параметров;
- обслуживание устройств верхнего уровня, прием и исполнение их команд;
- выдача значений параметров и различных сообщений на панель оператора;
- обслуживание технического персонала при наладке, программировании, ремонте, проверке технического состояния контроллера;

- самоконтроль и диагностика всех устройств контроллера в непрерывном и периодическом режимах, вывод информации о техническом состоянии контроллера обслуживающему персоналу [1].

2.2.2 Состав контроллера

Контроллер является проектно-компонуемым изделием, состав которого определяет пользователь в зависимости от решаемых задач. Компоновка контроллера осуществляется блоками, модулями и другими изделиями, входящими в его состав.

В состав контроллера входят следующие устройства:

- ✓ блок центрального процессора БЦП (предназначен для организации и управления вычислительными процессами в реальном времени; исполнения технологической программы пользователя (ТПП); организация пользовательского интерфейса через панель технолога-оператора; обмена по протоколам TCP/IP и MODBUS с другими контроллерами и верхним уровнем и т.п.);
- ✓ модули ввода – вывода (МВВ) постоянного состава (дискретные модули (DI1-16, DIO1-8/8, DO1-16), аналоговые модули (TC1-7, TR1-8, AI1-8, AIO1-8/0, AIO1-8/4, AIO1-0/4));
- ✓ модули ввода – вывода проектно-компонуемого состава (ADIO1, AIO2);
- ✓ микроконтроллер программируемый МК1 (выполняет все функции МВВ, а также различные дополнительные функции управления: исполнение ТПП, обмен с МВВ и верхним уровнем);
- ✓ терминальные блоки (для ввода сигналов объекта управления);
- ✓ соединения гибкие (для соединения ТБ и МВВ);
- ✓ блок и модули питания;
- ✓ блок переключения БПР-10 (для контактного переключения до 8 аналоговых или дискретных сигналов и может применяться в схемах резервирования);
- ✓ пульт настройки РН1(для наладки, настройки и конфигурирования модулей, а также для контроля и изменения параметров ТПП блока центрального процессора БЦП и микроконтроллеров в автономном режиме);
- ✓ панель оператора.

Внешний вид данных устройств контроллера Кросс -500 приведен на рисунке 2

[1].



Рисунок 2. Внешний вид локальных модулей контроллера КРОСС -500

2.2.3 Основные технические характеристики контроллера

Основные технические характеристики представлены в таблице 1 [1].

Таблица 1. Основные технические характеристики контроллера КРОСС-500

ВХОДНЫЕ/ВЫХОДНЫЕ СИГНАЛЫ	
Максимальное число аналоговых входов-выходов	384
Максимальное число дискретных входов-выходов	512
ВХОДНЫЕ АНАЛОГОВЫЕ СИГНАЛЫ	
Число входов на модуль постоянного состава	8
Число входов на модуль проектно-компоуемого состава	до 20
Число ячеек в модуле проектно-компоуемого состава	до 8
Число входов в ячейке	1, 2, 4
Виды сигналов: <ul style="list-style-type: none"> • унифицированные • сигналы от термопар • сигналы от термопреобразователей сопротивления 	(0-5), (0-20), \pm (0-5), \pm (0-20) мА, (0-10), \pm (0-10) В \pm (0-35), \pm (0-70) мВ (0-100), (0-200), (0-400) Ом
Погрешности:	

<ul style="list-style-type: none"> • предел допускаемой основной приведенной погрешности • предел допускаемой дополнительной приведенной погрешности при изменении температуры на 10 °С • предел допускаемой абсолютной погрешности для термопар • предел допускаемой абсолютной погрешности для термопреобразователей сопротивления • предел допускаемой абсолютной погрешности канала компенсации температуры холодного спая 	<p>± 0.1%, ± 0.2% в зависимости от типа модуля</p> <p>± 0.1%, ± 0.2% в зависимости о типа модуля</p> <p>от ± 0.5⁰С до ± 3.0⁰С в зависимости от диапазона измерения</p> <p>от ± 0.3⁰С до ± 1.2⁰С в зависимости от диапазона измерения</p> <p>± 0.5%, ± 0.6% в зависимости от типа модуля</p>
Гальваническое разделение	вид разделения (индивидуальное или групповое) зависит от типа модуля, испытательное напряжение 500В
ВЫХОДНЫЕ АНАЛОГОВЫЕ СИГНАЛЫ	
Число выходов на модуль постоянного состава	4
Число выходов на модуль проектно-компоуемого состава	до 16
Число выходов в ячейке	1, 2
Шаг изменения числа выходов	1, 2, 4
Виды сигналов: <ul style="list-style-type: none"> • унифицированные 	(0-5), (0-20), (4-20) мА
Погрешности: <ul style="list-style-type: none"> • предел допускаемой основной приведенной погрешности • предел допускаемой дополнительной приведенной погрешности при изменении температуры на 10 °С 	<p>± 0.1%, ± 0.2% в зависимости от типа модуля</p> <p>± 0.1%</p>
Гальваническое разделение	вид разделения (индивидуальное или групповое) зависит от типа модуля, испытательное напряжение 500В
ВХОДНЫЕ ДИСКРЕТНЫЕ СИГНАЛЫ	
Число входов на модуль	8, 16 (группами по 8 входов)
Шаг изменения числа входов	8
Виды сигналов: <ul style="list-style-type: none"> • сигнал логического «0» • сигнал логической «1» 	<p>(0-7) В</p> <p>(24±6) В</p>
Максимальный ток	0,01 А – на один канал по цепи 24 В
Гальваническое разделение	Между группами входов, испытательное напряжение 500В
ВЫХОДНЫЕ ДИСКРЕТНЫЕ СИГНАЛЫ	
Число выходов на модуль	8, 16 (группами по 8 входов)
Шаг изменения числа выходов	8
Напряжение коммутации транзисторного	до 40 В

выхода	
Ток коммутации	до 0,3 А на один канал, но не более 2 А – на 8 каналов одной группы
Гальваническое разделение	Между группами входов, испытательное напряжение 500В
ВХОДНЫЕ-ВЫХОДНЫЕ ДИСКРЕТНЫЕ СИГНАЛЫ ТЕРМИНАЛЬНЫХ БЛОКОВ	
Номинальное напряжение включения (коммутации)	24В, 110В, 220В
Гальваническое разделение	Имеется, испытательное напряжение 500 или 1500 В в зависимости от вида блока
ОБЩИЕ ФУНКЦИОНАЛЬНЫЕ ПАРАМЕТРЫ	
Операционная система реального времени (ОС РВ)	RTOS-32
Исполнительная система	ISaGRAF Target
Технологическое программирование (система ISaGRAF) на языках:	-последовательных функциональных схем SFC; - релейной логики LD; - структурированного текста ST; - функциональных блоков FBD; - инструкций IL;
Объем памяти процессора : • flash-память • динамическое ОЗУ • статическое ОЗУ	• 1 Мбайт • 4 Мбайт • 256 кбайт
Общая емкость памяти программ пользователя (оценочно)	до 10 инструкций входного языка ST ISaGRAF
Часы реального времени	Формат: секунды - минуты - часы - года
Минимальное время цикла (шаг его изменения)	2мс (1мс)
Время сохранения технологических программ	Без ограничения времени

Контроллер соответствует климатическому исполнению УХЛ 4.2 по ГОСТ 15150-69. Контроллер является изделием общего назначения по ГОСТ 18311, место установки контроллера - взрывопожаробезопасное помещение. Средний срок службы контроллера не менее 10 лет.

2.2.4 Программирование контроллера

Контроллеры КРОСС-500, ТРАССА-500 являются семейством программно- и системно-совместимых приборов, обеспечивающих:

- ✓ переносимость технологических программ между контроллерами семейства;
- ✓ возможность работы в одной контроллерной сети и взаимодействия по единым сетевым протоколам;
- ✓ связь с верхним уровнем с помощью единого OPC – сервера.

Эти возможности обеспечиваются использованием в контроллерах семейства популярной системы программирования ISaGRAF, соответствующей международному стандарту открытых систем IEC (МЭК) 1131-3, что обеспечивает совместимость контроллеров семейства с контроллерами других фирм, использующих ISaGRAF [3,9].

ISaGRAF представлен в виде двух частей: набора средств разработки ISaGRAF Workbench и исполняемого на контроллере ядра-интерпретатора ISaGRAF Target. Набор средств разработки выполняется на компьютере проектировщика, например, компьютере типа IBM PC, и состоит из редактора, отладчика и препроцессора, который подготавливает описанный проектировщиком алгоритм к формату, «понятному» ядру-интерпретатору. Этот набор имеет современный пользовательский интерфейс, позволяет тестировать алгоритм в режиме эмуляции и получать листинг алгоритма на языках его описания. Ядро-интерпретатор ISaGRAF Target размещается в контроллере в качестве его резидентного программного обеспечения.

После создания пользовательская программа загружается в контроллер для исполнения. Ядро-интерпретатор, как следует уже из его названия, транслирует пользовательский алгоритм во время исполнения. Это позволяет сконцентрировать машино-зависимый код и таким образом снизить накладные расходы при переходе на другой контроллер.

В целом, программное обеспечение(ПО) контроллера – это совокупность резидентного программного обеспечения (РПО), встроенного в контроллер, и внешних программных средств (ПО операторской станции).

Резидентное программное обеспечение состоит из РПО нескольких блоков.

Резидентное программное обеспечение БЦП включает в свой состав:

- ✓ операционную систему RTOS-32;
- ✓ исполнительную систему ISaGRAF Target;
- ✓ подсистему ввода-вывода для связи с внешними модулями УСО;
- ✓ коммуникационные серверы (ModBus, Telnet, FTP и т.п.), осуществляющие связь с верхним уровнем, другими контроллерами, панелью оператора и др. устройствами.

1. Резидентное программное обеспечение модулей ввода-вывода включает в себя:

- ✓ операционную систему реального времени;
- ✓ драйверы ввода-вывода аналоговых и дискретных сигналов;

- ✓ программы первичной обработки сигналов;
 - ✓ драйверы интерфейсных каналов;
 - ✓ программы контроля и диагностики аппаратуры модуля.
2. Резидентное программное обеспечение МК1 включает в себя:
- ✓ операционную систему реального времени;
 - ✓ драйверы ввода-вывода аналоговых и дискретных сигналов;
 - ✓ программы первичной обработки сигналов;
 - ✓ систему выполнения ТПП, написанных на технологических языках FBD, ST, SFC программирования системы ISaGRAF;
 - ✓ драйверы интерфейсных каналов;
 - ✓ программы контроля и диагностики аппаратуры МК1.

Ядром РПО блока БЦП (как и МК1) является исполнительная система ISaGRAF Target версии 3.3. Данная система способна исполнять код программ, созданных при помощи ISaGRAF Workbench версий 3.xx, и написанных на любых из технологических языков в соответствии со стандартом МЭК 61131-3. В качестве языков программирования МК1 используются технологические языки FBD, ST, SFC системы разработки ISaGRAF Workbench, расширенной новыми алгоритмами, в том числе из библиотеки контроллера Р-130.

Внешние (нерезидентные) программные средства контроллера, исполняемые на РС-совместимом компьютере под операционной системой Windows NT, 2000, ХТ имеют следующий состав:

- ✓ систему разработки технологических программ пользователя (ТПП) ISaGRAF Workbench;
- ✓ программный пакет «Конфигуратор» для контроля и настройки модулей контроллера;
- ✓ программные средства связи с верхним уровнем, включающие OPC сервер, библиотеки подпрограмм связи верхнего уровня с БЦП, МВВ и МК1 [3].

2.2.5 Типовые структуры АСУ ТП

Как отмечалось ранее, контроллер является проектно-компонентным устройством. Поэтому он не имеет базового состава и может функционировать в рамках как процессорных, так и микроконтроллерных структур.

Процессорная структура

В типовом случае контроллер имеет процессорную структуру, т.е. в качестве блока, исполняющего основной алгоритм и управляющего всеми остальными компонентами системы, используется блок БЦП.

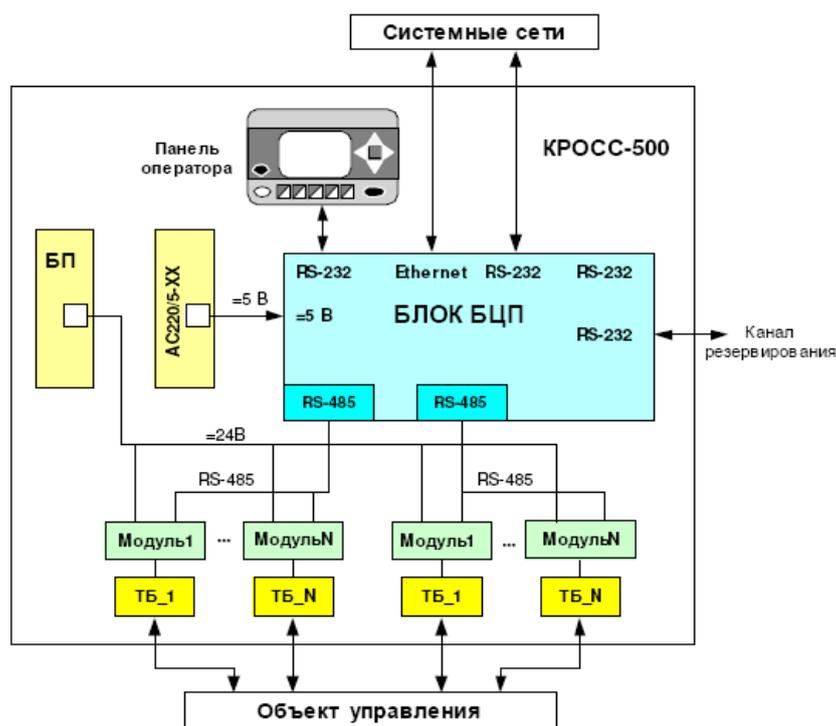
При этом в составе контроллера могут быть полевые микропроцессорные контроллеры МК1, параллельно исполняющие ТПП со своими локальными задачами и обменивающиеся данными с БЦП. Это позволяет повысить динамику и надежность системы, а также сократить информационный обмен между БЦП и МК1.

МК1, в свою очередь, также может иметь собственные модули ввода-вывода. Процессорная структура контроллера, включающая две полевые сети на основе шины RS-485, приведена на рисунке 3.

Применение резервирования повышает надежность систем автоматизации на базе контроллеров. Различные варианты резервирования аппаратных средств, наличие нескольких видов сетевых каналов дают возможность реализовать различные конфигурации контроллеров.

В контроллере КРОСС-500 предусмотрено резервирование БЦП, контроллеров, аппаратуры ввода-вывода, полевых сетей. Резервирование может быть реализовано либо встроенными системными средствами контроллера, либо проектными средствами.

Встроенные программно-аппаратные средства контроллера не требуют дополнительных мероприятий при подготовке контроллера к работе, кроме реализации схемы подключения. Проектные средства требуют управления процедурами резервирования со стороны технологической программы пользователя.



RS-485-внутренняя полевая шина контроллера;

Модуль-модули МВВ или МК1;

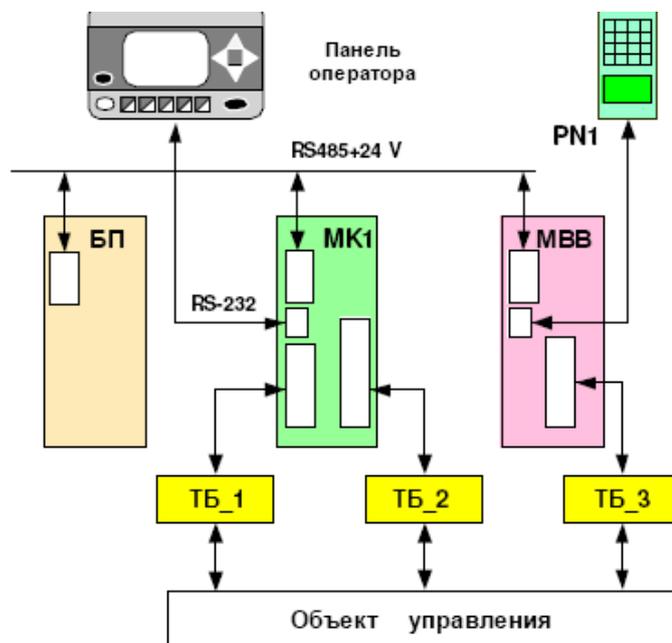
АС220/5-XX – модуль питания;

ТБ – терминальный блок

Рисунок 3. Процессорная структура контроллера АСУ ТП

Микроконтроллерная структура

Для локальных или небольших распределенных систем контроллер может не иметь в составе блока БЦП. В этом случае основную технологическую программу пользователя исполняет МК1, используя подключенные к нему МВВ. Микроконтроллерная структура приведена на рисунке 4. При необходимости, МК1 могут объединяться в сеть по интерфейсу RS-485 [1].



RS485 – внутренняя полевая шина контроллера;
 RS232 - связь с панелью оператора;
 БП – блок питания;
 МВВ – модули ввода-вывода (до 4 шт.);
 ТБ – терминальные блоки;
 РН1–пульт настройки.

Рисунок 4. Микроконтроллерная структура контроллера АСУ ТП

2.3 МИКРОПРОЦЕССОРНЫЙ КОНТРОЛЛЕР ТРАССА-500

2.3.1 Назначение и область применения контроллера

Основное назначение контроллера – построение эффективных систем автоматизации территориально рассредоточенных, протяженных или небольших объектов. Контроллер предназначен для решения задач автоматизации: сбор информации с датчиков различных типов и ее первичная обработка (фильтрация сигналов); выдача управляющих воздействий на исполнительные органы различных типов; регулирование прямых и косвенных параметров по различным законам; логическое, программно-логическое управление технологическими агрегатами, автоматический пуск и остановка технологического оборудования; математическая обработка информации по различным алгоритмам; регистрация и архивация технологических параметров; обмен данными с другими контроллерами в рамках контроллерной управляющей сети реального времени; аварийная, предупредительная и рабочая сигнализация, индикация значений прямых и косвенных параметров.

Управление объектом осуществляется с помощью ТПП, созданной пользователем на технологических языках программирования системы ISaGRAF.

Контроллер является проектно-компануемым и программируемым изделием. Его состав и ряд параметров определяются потребителем и

указываются в заказе. Контроллер имеет сертификат типа средства измерений и может выполнять учет материальных и энергетических потоков на различных участках производства.

Основные области применения контроллера – системы автоматизации технологических объектов широкого класса в различных отраслях энергетики, машиностроения, металлургической, химической, горнодобывающей промышленности и связи, в том числе теплоснабжение, водоснабжение, нефте- и газоснабжение, предприятие агропромышленного комплекса.

Контроллер ориентирован на построение недорогих систем различной алгоритмической и информационной сложности: макросистем (до 3840 каналов); миди-систем (64-128 каналов); мини-систем (16-64 каналов).

2.3.2 Состав, основные возможности и технические характеристики контроллера

Контроллер ТРАССА является проектно-компонуемым изделием, состав которого определяет пользователь в зависимости от решаемых задач. Компоновка контроллера осуществляется блоками, модулями и другими изделиями, входящими в его состав.

В состав контроллера входят следующие программные и аппаратные средства:

1. Центральный процессор БЦП
2. Модули ввода-вывода аналоговых и дискретных сигналов МВВ:
 - проектно-компонуемый модуль ввода-вывода дискретных сигналов Т-DIO;
 - проектно-компонуемый модуль ввода-вывода аналоговых и дискретных сигналов повышенного быстродействия Т-ADIO.
3. Программируемый микроконтроллер Т-МК1
4. Сетевой шлюз (СШ)
5. Модуль преобразования интерфейсов МПИ*
6. Модули питания:
 - модуль DC-24/5;
 - модуль AC220/5-15;
 - модуль DRA-60-24.
7. Блок переключения резерва БПР-10
8. Портативный пульт настройки (ПН).

Программное обеспечение контроллера позволяет пользователю:

- выполнять широкий круг алгоритмических задач: вычисление алгебраических и тригонометрических функций, статических и динамических преобразований, регулирования, программно-логического управления, защиты, учета, регистрации и архивации данных и т.п.;

- обеспечить возможность контроля, управления и тестирования каналов ввода-вывода в автономном режиме и с помощью компьютера, при этом обеспечивается возможность переноса, тиражирования программ;
- достичь снижения затрат на разработку и отладку программ пользователя за счет удобства и простоты программирования, их переноса и документирования, независимости от способов построения и работы устройств ввода-вывода.

Поддержка промышленных сетей и возможность обмена данными в реальном масштабе времени, использование технологических, процедурных языков программирования (системы ISaGRAF) и операционной системы реального времени OS-9 позволяют:

- интегрировать контроллер в единую систему АСУ ТП, содержащую контроллеры различных производителей, выполненные в стандарте открытых систем;
- масштабировать системы;
- сопрягать контроллер с различными SCADA-системами через стандартные средства межзадачного обмена;
- использовать единую технологию программирования контроллеров различных фирм и переносить технологические программы пользователя;
- подключать контроллер к глобальной информационной сети Internet благодаря наличию встроенного Web-сервера.

В контроллере применена интеллектуальная подсистема ввода-вывода.

Все модули устройств связи с объектом (УСО) контроллера имеют встроенный бортовой микропроцессор, выполняющий независимо и асинхронно по отношению к центральному процессору различные функции по обработке сигналов и диагностике оборудования.

Такой подход позволяет:

- повысить надежность контроллера за счет сокращения объема аппаратуры модулей и непрерывной самодиагностики;
- повысить живучесть контроллера за счет децентрализации и автономного выполнения различных функций;
- увеличить производительности и уменьшить время цикла контроллера за счет сокращения нагрузки на центральный процессор по объему вычислений и интенсивности обменов данными с модулями УСО;
- расширить номенклатуру модулей (модули ввода-вывода, модули контроля и управления исполнительными органами, модули микроконтроллера);
- обеспечить простоту и переносимость технологических программ, снижение затрат на их разработку и отладку за счет их независимости от способов построения и работы аппаратуры ввода-вывода аналоговых и дискретных сигналов, датчиков и исполнительных органов;
- предусмотреть возможность контроля, управления и тестирования модуля в автономном режиме с помощью переносного пульта настройки или компьютера через последовательный порт модуля с интерфейсом RS-232.

Основные технические характеристики

Центральный процессор

Центральный процессор БЦП контроллера построен на базе PC-совместимого модуля Wafer 4821 и содержит микропроцессор i486DX4-100 с встроенным сопроцессором, флэш-память, оперативную энергонезависимую и динамическую память, сторожевой таймер, таймер-календарь, каналы связи с УСО и внешними устройствами

Входы-выходы контроллера

Контроллер обеспечивает ввод-вывод сигналов следующих типов:

- входные аналоговые сигналы;
- выходные аналоговые сигналы;
- входные дискретные сигналы;
- выходные дискретные/импульсные сигналы.

Максимальное число аналоговых/дискретных входов-выходов контроллера-7648.

Каналы ввода-вывода размещаются в модулях с проектно-компонованным составом и микроконтроллерах. Электрические, точностные и динамические характеристики, а также тип гальванической развязки входных и выходных сигналов определяются типом модуля или канальной ячейки.

Контроллер обеспечивает ввод-вывод аналоговых и дискретных сигналов со следующими характеристиками.

1. Входные аналоговые сигналы:

- унифицированные сигналы тока или напряжения 0-10 В, 0-5, 0-20, 4-20 мА (0-5 мА, $R_{вх}=400$ Ом; 0-20 мА, $R_{вх}=100$ Ом; 4-20 мА, $R_{вх}=100$ Ом; 0-10 В, $R_{вх}=27$ кОм);
- сигналы термопар типа сигналов термопар ТВР(А-1, А-2, А-3), ТПР(В), ТПП(Р, S) ТХА(К), ТХК(Л, Т), ТНН(Н), ТМК(Т), ТЖК(Ж) ГОСТ Р 50342-92;
- трех- и четырехпроводные сигналы термометров сопротивлений ТСМ50, ТСМ100, ТСП50, ТСП100 ГОСТ 6651;
- частотные и число-импульсные сигналы 0-50 кГц.
- разрешающая способность АЦП — 12 или 15 разрядов;
- основная погрешность — 0.2 или 0,1 %;
- цикл обновления значения сигнала- 2 мкс, 60 мс, 120 мс на канал
- гальваническая развязка – индивидуальная для всех типов сигналов или групповая (4 канала в группе) для унифицированных сигналов тока или напряжения.

2. Выходные аналоговые унифицированные сигналы тока 0-5, 0-20, 4-20 мА:

- 0-5 мА, ($R_{н}=2$ кОм);
- 0-20 мА, ($R_{н}=0,5$ кОм);
- 4-20 мА, ($R_{н}=0,5$ кОм);
- разрешающая способность ЦАП — (12 разрядов).
- основная погрешность — 0,2 %;
- время преобразования- 20 мкс;

- гальваническая развязка – индивидуальная или групповая (2 канала в группе).
3. Входные дискретные сигналы:
- 24V DC;
 - 24V AC;
 - 110V AC;
 - 220V AC.
5. Дискретные (импульсные) выходные сигналы:
- транзистор, 24V/0.3A DC;
 - симистор, 220V/1.0A AC;
 - реле, 12-220V/6A DC/AC;
 - твердотельное реле, исполнения 24V/1A, 110V/0.19A, 220V/0.14A DC/AC.
 - гальваническая развязка — индивидуальная или групповая (8 каналов в группе);
 - защита от короткого замыкания в цепи нагрузки — имеется.

Внешние интерфейсы

Контроллер имеет следующие внешние интерфейсы:

1. Системный канал Ethernet 10 BaseT для подключения к верхнему уровню и обмена данными между контроллерами. Канал имеет следующие характеристики:
 - режимы работы канала-master/slave;
 - физический интерфейс-RJ45;
 - скорость передачи данных-10МБод;
 - линия связи-витая пара.
2. Системный канал с протоколами Modbus или альтернативными протоколами для подключения к верхнему уровню. Канал имеет следующие характеристики:
 - режимы работы канала- master/slave;
 - физический интерфейс-RS-485;
 - скорость передачи данных-стандартный ряд скоростей СОМ-порта;
 - линия связи-витая пара.
3. Канал для резервирования процессоров и контроллеров. Канал имеет следующие характеристики:
 - режимы работы канала- master/slave;
 - физический интерфейс-RS-232;
 - скорость передачи данных- до 1 МБод;
 - линия связи-девятипроводный кабель.
4. Канал для подключения инженерной станции. Канал имеет следующие характеристики:
 - режимы работы канала-slave;
 - физический интерфейс-RS-232;
 - скорость передачи данных-стандартный ряд скоростей СОМ-порта.
 - линия связи-трехпроводный кабель.

5. Канал для подключения пульта технолога-оператора. Канал имеет следующие характеристики:
 - режимы работы канала-master;
 - физический интерфейс-RS-232;
 - скорость передачи данных-стандартный ряд скоростей СОМ-порта.
 - линия связи-трехпроводный кабель.
6. Каналы для подключения полевых приборов.
 - число каналов-до четырех;
 - режимы работы канала-master;
 - физический интерфейс-RS-485;
 - линия связи-витая пара.

2.3.3 Модули контроллера

Микроконтроллер Т-МК1 предназначен для использования в процессорных и беспроцессорных структурах контроллеров. В последнем случае он может использоваться самостоятельно или совместно с модулями МВВ и служить приборной базой для замены существующих систем на базе релейной автоматики и аналоговых приборов АКЭСР.

Т-МК1 построен как проектно-компонованное по заказу потребителя изделие на базе подсистемы ввода-вывода модуля Т-АДИО и имеет до 32 каналов ввода-вывода аналоговых и дискретных сигналов.

В качестве языков программирования МК1 используются пять типов технологических языков системы разработки IsaGRAF Workbench в соответствии со стандартом МЭК 1131-3, расширенной новыми алгоритмами, в том числе из библиотеки контроллера Р-130. Программирование Т-МК1 осуществляется при помощи компьютера. В перспективе планируется возможность программирования Т-МК1 на фирменном языке FST («функционально-ориентированный структурированный текст») с помощью обычного текстового редактора. Особенности программирования Т-МК1 приведены в руководстве по программированию.

Микроконтроллер выполняет все перечисленные выше функции модулей ввода-вывода, а также различные дополнительные функции управления, заложенные в технологической программе пользователя, в том числе:

- нормализация сигналов;
- логические функции, в том числе релейная логика;
- функции автономной защиты по значениям дискретных и аналоговых входных сигналов;
- управление позиционерами;
- функции порогового элемента;
- функции двух- и трехпозиционного регулятора, ПИД-регулятора;
- управление движением и диагностика исполнительного механизма;

- другие алгебраические, логические и временные функции.

В качестве языков программирования Т-МК1 используются пять типов технологических языков системы разработки IsaGRAF Workbench в соответствии со стандартом МЭК 1131-3, расширенной новыми алгоритмами, в том числе из библиотеки контроллера Р-130. Программирование Т-МК1 осуществляется при помощи компьютера.

Блок Т-ADIO1, блок программируемого микроконтроллера Т-МК1 имеют проектно-компонованный состав, до 8 аналоговых, частотных и дискретных ячеек по заказу, параметры которых приведены в таблицах 2, 3.

Таблица 2 – Аналоговые и частотные ячейки блоков Т-ADIO1, Т-МК1

Ячейка	Вид и количество каналов в ячейке	Входной/ выходной сигнал	Входное сопротивление (сопротивление нагрузки)	Быстродействие, не более
AI1	1 канал ввода	Аналоговый сигнал: - постоянный ток: (4-20) мА; ±(0-5), ±(0-20) мА; - напряжение ± 10 В	(100±3) Ом (100±3) Ом не менее 32 кОм	60 мс
AI2	1 канал ввода	Аналоговый сигнал: - постоянный ток: (0-5) мА; (0-20), (4-20) мА; - напряжение (0-10) В	(400±10) Ом (100±3) Ом не менее 32 кОм	2 мкс
AI3	4 канала ввода	Аналоговый сигнал: - постоянный ток: (4-20) мА; ±(0-5), ±(0-20) мА	(100±3) Ом (100±3) Ом	120 мс
AO1	1 канал вывода	Аналоговый сигнал: - постоянный ток: (0-5) мА (0-20), (4-20) мА	не более 4 кОм не более 1 кОм	20 мкс
AO2	2 канала вывода			
FI1	2 канала ввода	Частотный сигнал (0-50) кГц; амплитуда 5, 12, 24 В	не менее кОм	-
TC1	1 канал ввода	Напряжение постоянного тока: ±(0-35), ±(0-70), ±(0-140), ±(0-280), ±(0-560), ±(0-1120), ±(0-2240) мВ. Напряжение постоянного тока ± (0-70) мВ от термопар	не менее 100 кОм	60 мс
TR1	1 канал ввода. Ввод трехпроводный	Сопротивление (0-50), (0-100), (0-200), (0-400) Ом. Сопротивление (0-400) Ом от термопреобразователей сопротивления.	-	60 мс
TR2	1 канал ввода. Ввод четырехпроводный			
TR3	2 канала ввода. Ввод четырехпроводный	Сопротивление (0-50), (0-100), (0-200), (0-400) Ом. Сопротивление (0-400) Ом от термопреобразователей сопротивления. Измерение разности температур.	-	120 мс

Таблица 3- Дискретные ячейки блоков Т-ADIO1, Т- МК1

Ячейка	Вид и количество каналов в ячейке	Входной/ выходной сигнал	Входное сопротивление, кОм, не менее	Гальваническое разделение, В, не менее
DI2	4 канала ввода	Дискретный сигнал: напряжение постоянного тока (0-7) В – логический «0», (24±6) В – логическая «1»	2	500
DO2	4 канала вывода	Дискретный сигнал: бесконтактный ключ, коммутируемое постоянное напряжение 24 В, ток до 2 А		500

Блок Т-DIO1 имеет проектно-компоновочный состав, до 8 дискретных ячеек по заказу, параметры которых приведены в таблицах 4-6.

Таблица 4 - Дискретные ячейки входных сигналов блока Т-DIO1

Параметр	DI1	DI3/220	DI3/110	DI3/24
Количество каналов в ячейке	2	1	1	1
Входной ток канала, мА	=7	~5	~5	~5
Максимальное напряжение включения, В, не более	30	250	180	30
Минимальное напряжение включения, В, не менее	18	120	60	15
Максимальное напряжение выключения, В, не более	7	100	50	12
Гальваническое разделение, В, не менее	1500	1500	1500	1500

Таблица 5 - Дискретная ячейка DO3 блока T-DIO1

Параметр	DO3
Количество каналов в ячейке	1
Коммутируемый ток реле, А: *	
- при напряжении ~250 В и активной нагрузке;	6
- при напряжении =30 В и активной нагрузке	6
Минимальное напряжение коммутации, В *	5
Минимальный ток коммутации, мА *	10
Число циклов коммутации при напряжении 30 В постоянного тока 6 А, не менее *	5 ⁵ 5*10
Число циклов коммутации при напряжении 250 В переменного тока 6 А, не менее *	4 ⁴ 3*10
Число циклов механической коммутации (без нагрузки), не менее *	7 ⁷ 3*10
Гальваническое разделение, В, не менее	1500
* Технические данные реле	

Таблица 6 - Дискретные ячейки выходных сигналов блока T-DIO1

Параметр	DO1	DO4	DO5/220	DO5/110	DO5/24
Количество каналов в ячейке	2	1	1	1	1
Номинальное напряжение коммутации, В	=(18-30)	~220	~220	~110	~24
Максимальное напряжение коммутации, В, не более	=40	~250	~400	~250	~60
Максимальный ток коммутации, А, не более (один канал)	0,3	1	0,12	0,17	1
Гальваническое разделение, В, не менее	1500	1500	1500	1500	1500

Пределы допускаемой основной погрешности (γ_0) и дополнительной погрешности (γ_d) контроллеров, вызванной изменением температуры в рабочем диапазоне от плюс 5 до плюс 50 °С, указаны в таблице 7.

Таблица 7-Пределы погрешности

Ячейка Т-ADIO1, Т-МК1	Сигналы *		Предел допускаемой основной приведенной погрешности γ_0 , %	Пределы допускаемой дополнительной погрешности при изменении температуры на 10 °С, %
	на входе	на выходе		
ТС1	Напряжение постоянного тока: $\pm(0-35)$, $\pm(0-70)$, $\pm(0-140)$, $\pm(0-280)$, $\pm(0-560)$, $\pm(0-1120)$, $\pm(0-2240)$ мВ	$(0-100,0)$ %; 15 бит	$\pm 0,1$	γ_0
	Напряжение $\pm(0-70)$ мВ от термопар.			
TR1, TR2, TR3	Сопротивление $(0-50)$, $(0-100)$, $(0-200)$, $(0-400)$ Ом	$(0-100,0)$ %; 15 бит	$\pm 0,1$	γ_0
	Сопротивление $(0-400)$ Ом от термопреобразователей сопротивления			
AI1	Напряжение постоянного тока $\pm(0-10)$ В. Постоянный ток $\pm(0-5)$, $\pm(0-20)$ мА, $(4-20)$ мА	$(0-100,0)$ %; 15 бит	$\pm 0,1$	γ_0
AI2	Напряжение постоянного тока $(0-10)$ В. Постоянный ток $(0-5)$, $(0-20)$, $(4-20)$ мА.	$(0-100,0)$ %; 12 бит	$\pm 0,1$	γ_0
AI3	Постоянный ток $\pm(0-5)$, $\pm(0-20)$ мА, $(4-20)$ мА.	$(0-100,0)$ %; 15 бит	$\pm 0,1$	γ_0
FI1	Частота $(0-50)$ кГц; амплитуда 5, 12, 24 В.	$(0-100,0)$ %; 15 бит	$\pm 0,1$	γ_0
AO1, AO2	$(0-100,0)$ %; 12 бит	Постоянный ток: $(0-5)$, $(0-20)$, $(4-20)$ мА	$\pm 0,1$	γ_0

2.3.4 Надежность контроллера

Контроллер ТРАССА-500 обеспечивает высокую надежность управления технологическим процессом благодаря принятым мерам обеспечения собственной высокой надежности и живучести.

Базовый уровень надежности обеспечивается следующими средствами:

- исполнение элементной базы ведущих зарубежных фирм;
- резкое снижение числа межблочных контактных соединений и длины линий связи за счет использования последовательных полевых шин (2-4 сигнальных провода);
- использование высоконадежного программного обеспечения, имеющего сотни тысяч инсталляций (исполнительная подсистема ISaGRAF, сетевое ПО, операционная система RTOS-32);
- непрерывная диагностика блоков, наличие сторожевых таймеров во всех блоках;
- защита дискретных выходов блоков от короткого замыкания;
- SMD-монтаж, автоматизированная сборка и контроль блоков.

Базовый уровень живучести обеспечивается следующими средствами:

- Возможность автономного дублирования на блоках контроля и управления особо ответственных функций центрального процессора (защиты, блокировки, регулирование и т.д.)
- Возможность «горячей» замены блоков.

2.3.6 Типовые структуры АСУ ТП, разработанной на базе контроллера ТРАССА-500

Контроллер является проектно-компонуемым устройством. Поэтому он не имеет базового состава и может функционировать в рамках как процессорных, так и микроконтроллерных структур.

Процессорная структура

В типовом случае контроллер имеет процессорную структуру (рисунок 5), то есть в качестве блока, исполняющего основной алгоритм и управляющего всеми остальными компонентами системы, используется блок БЦП. При этом в составе контроллера могут быть блоки Т-МК1, параллельно исполняющие ТПП со своими локальными задачами и обменивающиеся данными с БЦП. Блоки Т-МК1 могут иметь собственные блоки ввода-вывода.

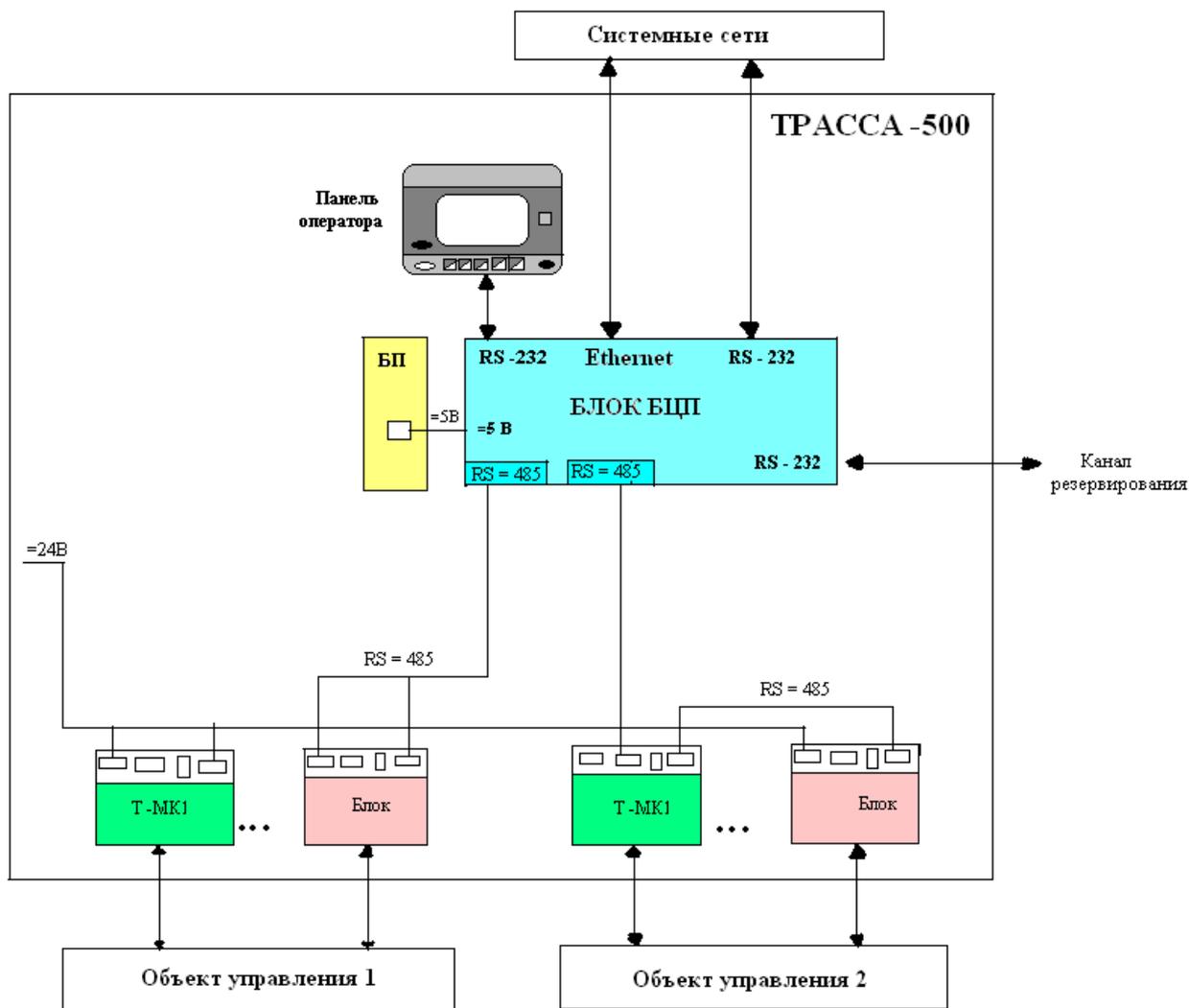


Рисунок 5. Процессорная структура контроллера АСУ ТП

- RS-485 - внутренняя полевая шина контроллера;
- Т-МК1 - микроконтроллер;
- БП – блок питания;
- Блок- блок ввода/вывода аналоговых и дискретных сигналов;
- RS-232 - связь с панелью оператора;
- БЦП – блок центрального процессора.

В качестве операторской станции используются SCADA- пакеты, которые выполняют функции сбора информации реального времени с удаленных точек (объектов) для обработки, анализа и возможного управления удаленными объектами.

Блок центрального процессора БЦП управляет работой контроллера, имеет резидентное программное обеспечение (РПО), включающее операционную систему реального времени RTOS-32 и исполнительную систему ISaGRAF Target. Которые предназначены для загрузки и выполнения технологической программы пользователя (ТПП).

OPC сервер осуществляет соединение и обмен данными между ISaGRAF-приложением контроллера ТРАССА и SCADA-системами через Ethernet или последовательный порт (RS-232).

В качестве панели оператора использует персональный компьютер, который позволяет создать и запустить программу контроллера через RS 232.

Модули ввода-вывода (МВВ) или микроконтроллер в зависимости от вида сигналов подразделяются на 2 группы:

- модули ввода-вывода аналоговых сигналов с групповой или индивидуальной гальванической развязкой между каналами;
- модули ввода-вывода дискретных сигналов с групповой гальванической развязкой.

Аналоговые сигналы используются для представления непрерывно изменяющихся физических величин. Дискретный сигнал: информационный сигнал, который представляется в виде отдельных отсчетов взятых по времени. Используется для сигнализации состояния объекта.

Модули соединяются с БЦП через RS-485. RS-485 (внутренняя полевая шина контроллера) - стандарт передачи данных по двухпроводному полудуплексному многоточечному последовательному каналу связи.

Терминальные блоки предназначены для подсоединения внешних цепей Т-МК1 через клеммные колодки, а также для преобразования и гальванического разделения дискретных сигналов.

Микроконтроллерная структура

Для локальных или небольших распределенных систем контроллер может не иметь в составе блока БЦП. В этом случае основную ТПП исполняет блок Т-МК1, используя подключенные к нему блоки ввода – вывода через последовательный порт 1, который должен быть настроен как Modbus Master. Последовательные порты 1,2 могут быть настроены на связь со средствами верхнего уровня (БЦП, OPC сервер, панель оператора и тд.) Микроконтроллерная структура приведена на рисунке 6. При

необходимости, блоки Т-МК1 могут объединяться в микроконтроллерную сеть по интерфейсу RS-485.

МК1, в свою очередь, также может иметь собственные модули ввода-вывода. Процессорная структура контроллера, включающая две полевые сети на основе шины RS-485. Как для БЦП, программу для микропроцессора можно составить с помощью ISaGRAF.

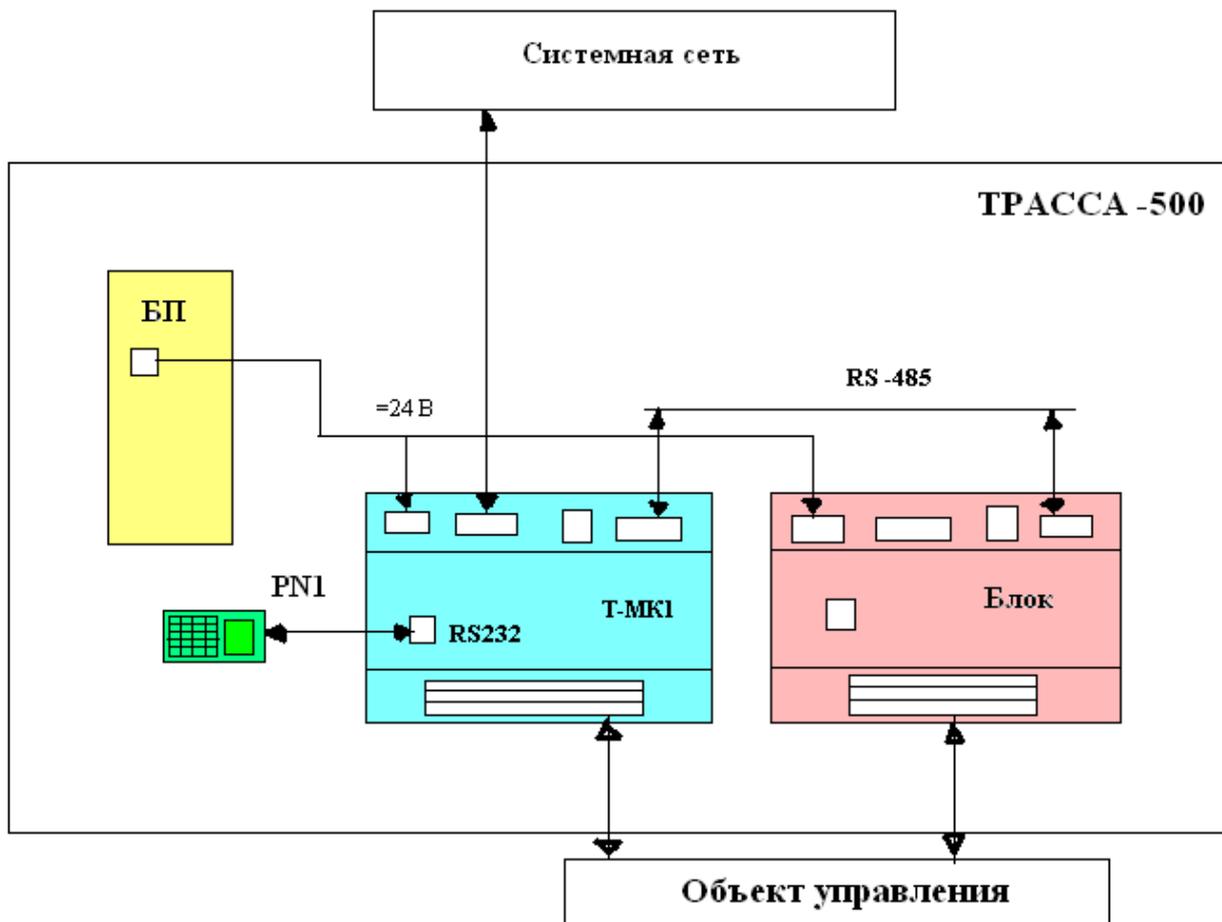


Рисунок 6. Микроконтроллерная структура АСУ ТП

- RS-485 - внутренняя полевая шина контроллера;
- RS-232 - связь с панелью оператора;
- РН1 – пульт настройки;
- БП - блок питания.

3 МЕТОДИКА РАЗРАБОТКИ ПРИКЛАДНЫХ ПРОГРАММ НА БАЗЕ ПАКЕТА ISAGRAF

В данной курсовом проекте для создания технологической программы пользователя для контроллеров КРОСС-500 и ТРАССА-500 необходимо использовать пакет SaGRAF.

3.1 Краткое описание программного пакета

ISaGRAF – это технология программирования контроллеров, которая позволяет создавать локальные или распределенные АСУТП. Основа технологии - среда разработки приложений (ISaGRAF Workbench) и адаптируемая под различные аппаратно-программные платформы исполнительная система (ISaGRAF Target).

Инструментальная система разработки приложений обеспечивает проект всеми возможностями языков стандарта IEC 1131-3. Устойчивость к сбоям, способность обрабатывать большое количество точек ввода/вывода позволяют с успехом применять ISaGRAF как в небольших встраиваемых приложениях, так и в распределенных проектах автоматизации.

ISaGRAF предоставляет пользователю удобный графический интерфейс для разработки и отладки приложений. Сгруппировав основные возможности описываемой интегрированной системы, то их можно представить в следующем виде:

- ✓ поддержка всех пяти языков программирования стандарта МЭК 61131-3;
- ✓ возможность смешивания программ/процедур, написанных на разных языках, а также вставка кодовых последовательностей из одного языка в код, написанных в другом языке;
- ✓ мощный отладчик, позволяющий во время работы прикладной задачи просматривать состояние переменных, код программ и многое другое;
- ✓ поддержка основных функции протокола MODBUS (RTU, SLAVE);
- ✓ открытость системы для доступа извне к внутренним структурам данных прикладной ISaGRAF – задачи, возможность разработки драйверов на модули ввода/вывода самим пользователем и портации ISaGRAF – ядра под любую аппаратно-программную платформу;
- ✓ наличие библиотеки драйверов для работы с устройствами ввода/вывода большого количества фирм - производителей (Kontron, Motorola MVME - 162, ABB, Computer Boards, Keitley Metrabyte, WEIDMULLER, Industrial Computer Source и др.).

Система разработки представляет собой набор Windows-приложений, интегрированных в единую инструментальную среду и работающих под ОС Win 95/98/NT. Основу системы исполнения составляет набор программных модулей (для каждой целевой системы свой), выполняющих самостоятельные задачи под управлением ядра ISaGRAF.

Ядро ISaGRAF реализует поддержку стандартных языков программирования, типового набора функций и функциональных блоков и драйверов ввода/вывода. Задача связи обеспечивает поддержку процедуры загрузки пользовательского ISaGRAF-приложения со стороны программируемого контроллера, а также доступ к рабочим переменным этого приложения со стороны отладчика системы разработки ISaGRAF. Взаимодействие систем разработки и исполнения осуществляется по протоколу MODBUS, что дает возможность доступа к данным контроллера не только отладчику ISaGRAF, но и любой системе визуализации и управления данными (SCADA). Драйверы устройств сопряжения с объектом организуют прозрачный доступ к аппаратуре ввода/вывода. Функции пользователя реализуют процедуры и алгоритмы функций, не представленные в стандартном варианте поставки системы ISaGRAF. Системные функции предназначены для описания специфики конкретной операционной системы, реализованной на данном типе контроллеров. Структура программного пакета ISaGRAF представлена на рисунке 7 [9].

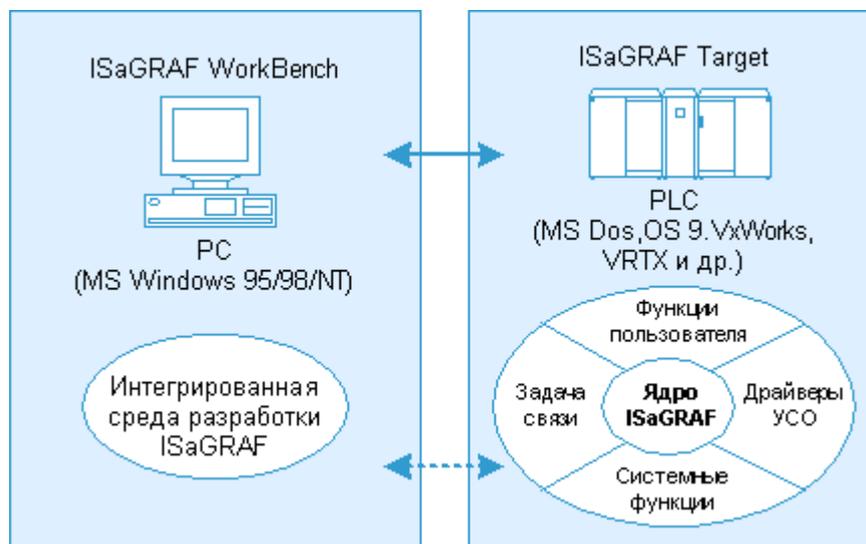


Рисунок 7. Структура программного пакета ISaGRAF

В ISaGRAF заложена методология структурного программирования, позволяющая пользователю представить автоматизируемый процесс в наиболее легкой и понятной форме. Стандартом IEC 1131-3 определяется пять языков: три графических (SFC, FBD, LD) и два текстовых (ST, IL). Все языки

программирования интегрированы в единую инструментальную среду и работают с едиными объектами данных.

Языки программирования пакета ISaGRAF

Для программирования контроллеров могут применяться следующие языки.

1. SFC - графический язык последовательных функциональных схем (Sequential Function Chart). Язык SFC предназначен для использования на этапе проектирования ПО и позволяет описать "скелет" программы - логику ее работы на уровне последовательных шагов и условных переходов.
2. LD - графический язык диаграмм релейной логики (Ladder Diagram). Язык LD применяется для описания логических выражений различного уровня сложности, программа на котором представляет собой аналог релейной схемы. Данный язык должен был облегчить переход инженеров из области релейной автоматики на ПЛК.
3. FBD - графический язык функциональных блочных диаграмм (Function Block Diagram), так же, как и LD, использующий аналогию с электрической (электронной) схемой. Программа на языке FBD представляет собой совокупность функциональных блоков, входа и выхода которых соединены линиями связи. Практика показывает, что FBD является наиболее распространенным языком стандарта IEC. Графическая форма представления алгоритма, простота в использовании, повторное использование функциональных диаграмм и библиотеки функциональных блоков делают язык FBD незаменимым при разработке программного обеспечения ПЛК.
4. IL - язык инструкций (Instruction List). Это язык низкого уровня класса ассемблера и применяется для программирования эффективных, оптимизированных процедур. Ввиду своей ненаглядности, IL практически не используется для программирования комплексных алгоритмов автоматизированного управления, но часто применяется для кодирования отдельных функциональных блоков, из которых впоследствии складываются схемы FBD или SFC.
5. ST - язык структурированного текста (Structured Text). Это язык высокого уровня, по мнемонике похож на Pascal и применяется для разработки процедур обработки данных. С помощью ST можно легко реализовывать арифметические и логические операции (в том числе, побитовые), безусловные и условные переходы, циклические вычисления; возможно использование как библиотечных, так и пользовательских функций. Язык также интерпретирует более 16 типов данных.

Язык ST может быть освоен технологом за короткий срок, однако текстовая форма представления программ служит сдерживающим фактором при разработке сложных систем, так как не дает наглядного представления ни о структуре программы, ни о происходящих в ней процессах.

Рисунок 8 демонстрирует место каждого из представленных в ISaGRAF языков на различных этапах разработки прикладного программного обеспечения (ППО).

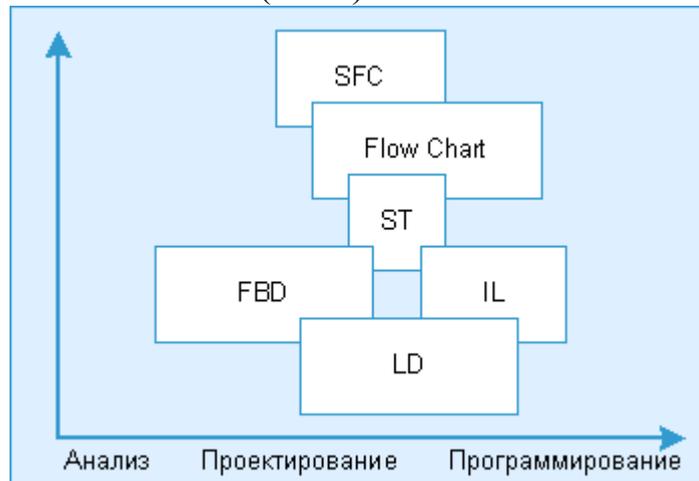


Рисунок 8. Языки программного пакета ISaGRAF на различных этапах разработки ППО

3.2 Методика составления программ регулирования и сигнализации и их отладка

Для создания программы, реализующей регулирование по ПИД - закону и сигнализацию необходимо запустить приложение ISaGRAF. После запуска приложения открывается окно, представленное на рисунке 9.

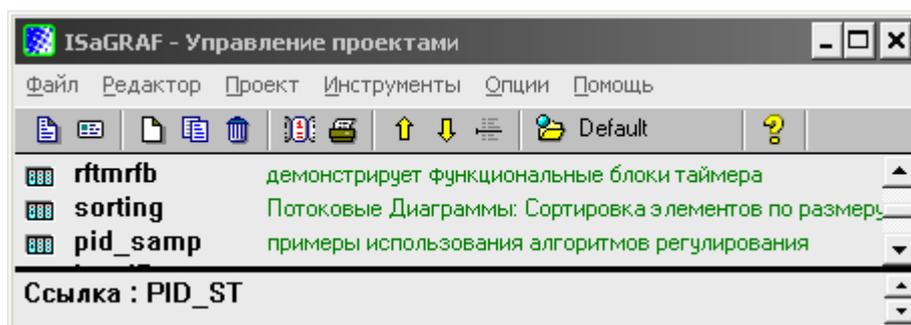


Рисунок 9. Окно приложения ПРОЕКТЫ

Для создания своего проекта следует воспользоваться меню ФАЙЛ/НОВЫЙ или используя сочетания клавиш CTRL+N. В результате, в появившемся окне необходимо ввести имя своего проекта, например PID_ST, данный проект сохранится в следующей директории:

C:\isawin\APL\PID_ST. После этого ваш проект появится в списке ПРОЕКТОВ.

Если вы реализуете на языке программирования FBD, это ваш проект будет включать одну программу, а если на языке ST - две. Поскольку реализовать всю программу на языке ST не возможно, т.к. обязателен функциональный блок для чтения значений на аналоговых входах и установки значений на аналоговых выходах модулей контроллера, поэтому проект PID_ST содержит в себе две программы – IN_OUT, выполненная на языке FBD, и PIDREG, выполненная на языке ST. Программу IN_OUT будет использоваться для ввода-вывода аналоговых сигналов с АВК-6, а в программе PIDREG описывается регулирование и сигнализации.

Нажав дважды ЛК на проекте, появится окно ПРОГРАММЫ или командой ФАЙЛ→ОТКРЫТЬ активируем менеджер программ (рис. 10), в котором создаются программы. Используя меню ФАЙЛ/НОВЫЙ или кнопкой , откройте диалоговое окно редактирования имени и языка программирования создаваемой программы. Наберите в нем имя программы, выберите язык программирования, при желании можно набрать комментарий к программе (рис. 11).

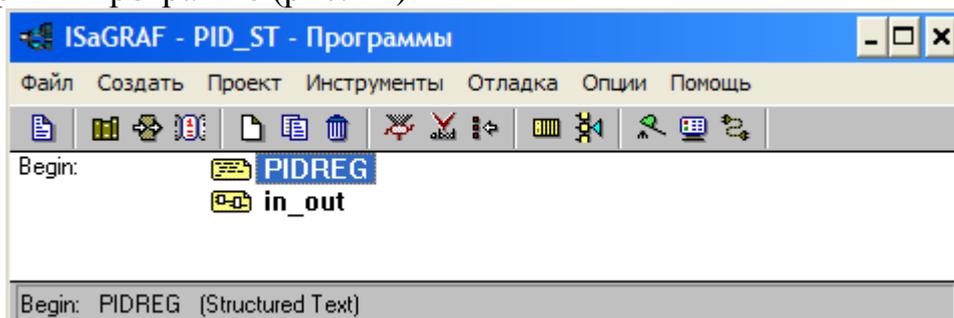


Рисунок 10. Менеджер программ

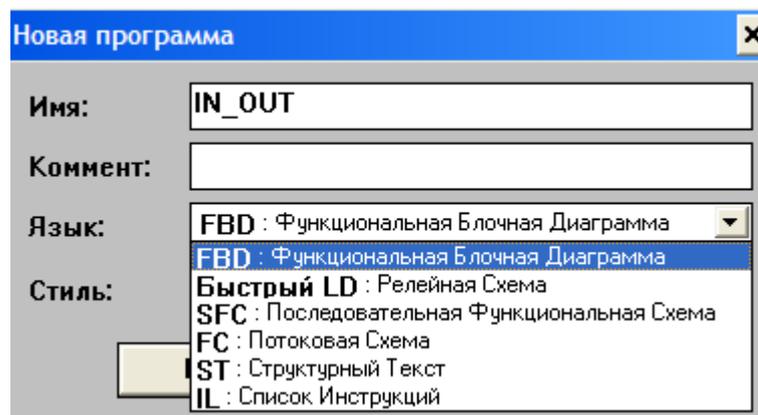


Рисунок 11. Окно создания нового проекта

Теперь необходимо открыть программу IN_OUT, нажав дважды ЛК по ее имени. Появится рабочая область, где и будем создавать нашу

программу. Прежде чем выберем какие – либо блоки, необходимо объявить в СЛОВАРЕ все переменные, которые пригодятся нам в процессе программирования. Вызов словаря осуществляется при помощи меню ФАЙЛ/СЛОВАРЬ или кнопкой  в менеджере программ, внешний вид словаря продемонстрирован на рисунке 12. Переменные объявляются в соответствии с типом данных. Вот данные типы данных:

- ✓ Булевские – логическая величина;
- ✓ Целые/Действительные – целая или действительная непрерывная величина;
- ✓ Таймеры – временная величина;
- ✓ Сообщения – строка символов;
- ✓ FB экземпляры – экземпляры функциональных блоков, используемые при программировании на языках ST и IL;
- ✓ Макроопределения – синонимы, используемые для замены текстовых строк.

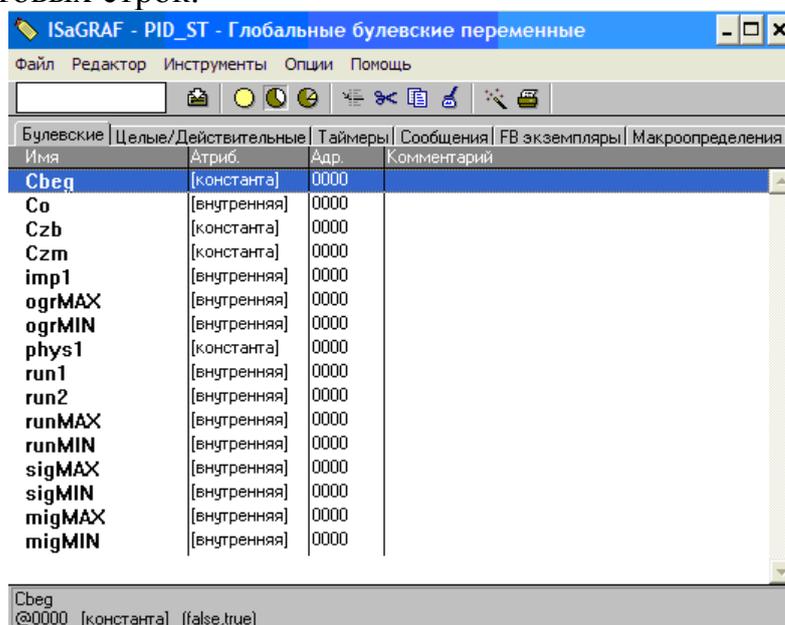


Рисунок 12. Словарь переменных

Для объявления переменной в словаре необходимо кликнуть ЛК по первой свободной строчке или же командой РЕДАКТОР→НОВЫЙ активируйте диалог редактирования булевской переменной, наберите имя переменной, атрибут и сохранить (рис. 13). Объявление целых/вещественных переменных (рис. 14) включает в себя описание имени переменной, атрибута переменной, формата, начального значения (в случаи константы) и комментария. Объявление FB экземпляров (рис. 15) сводится к выбору ФБ, вызов которого будет осуществляться в программе, написанной на языке структурированного текста ST.

Булевская переменная

Имя: Сетевой адрес:

Коммент:

Атрибуты

- Внутренняя
- Вход
- Выход
- Константа

Значения

False:

True:

нач. знач. true

Хранить

Сохранить

Отказ

Следующ

Предыдущ

Расширенный

Рисунок 13. Объявление булевской переменной

Целая/Вещественная переменная

Имя: Сетевой адрес:

Коммент:

Единицы:

Преобраз:

Атрибуты

- Внутренняя
- Вход
- Выход
- Константа

Формат

- Целая
- Вещест

Нач. знач:

Хранить

Сохранить

Отказ

Следующ

Предыдущ

Расширенный

Рисунок 14. Объявление целых/вещественных переменных

Функциональный блок

Имя: Тип:

Коммент:

Сохранить

Отказ

Следующ

Предыдущ

Рисунок 15. Объявление ФВ экземпляра

Далее следует вызвать функциональный блок (ФБ) ввода-вывода аналоговых сигналов (сгаіо) для этого можно использовать выпадающее меню панели управления либо использовать кнопку ВСТАВИТЬ ФБ и двойным нажатием по данному блоку, выбрать необходимый ФБ (рис. 16).

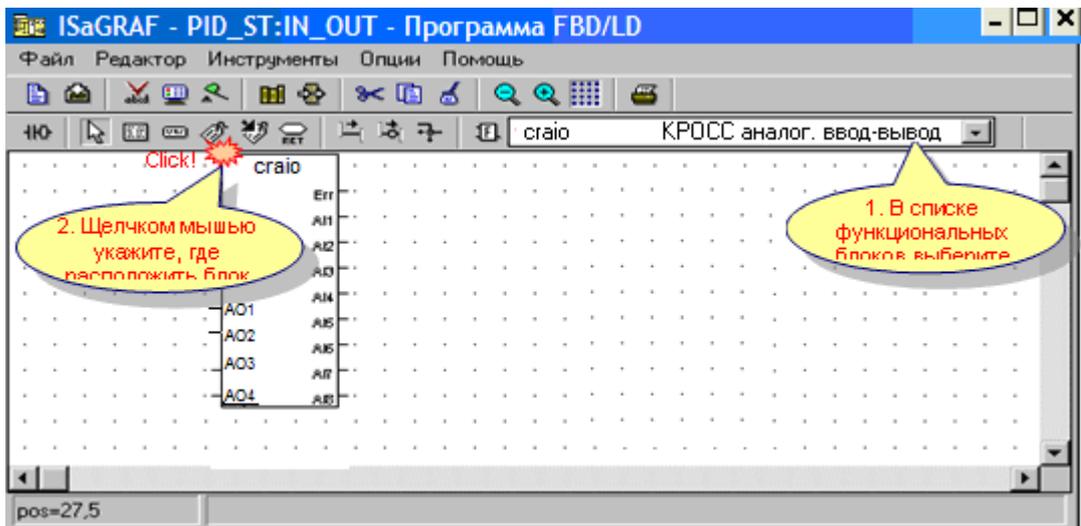


Рисунок 16. Вставка функционального блока модуля ввода-вывода craio

Теперь осуществляем вставку необходимых переменных, для этого нажмем на кнопку на панели управления **ВСТАВИТЬ ПЕРЕМЕННУЮ**, щелчком мышь в области редактирования текста программы редактора укажите место, где вы хотите его разместить. В появившемся диалоговом окне выбора переменной наберите название и кнопкой **Принять** вставьте ее в программу (рис. 17). Таким же образом необходимо вставить все переменные, используемые в данной программе.

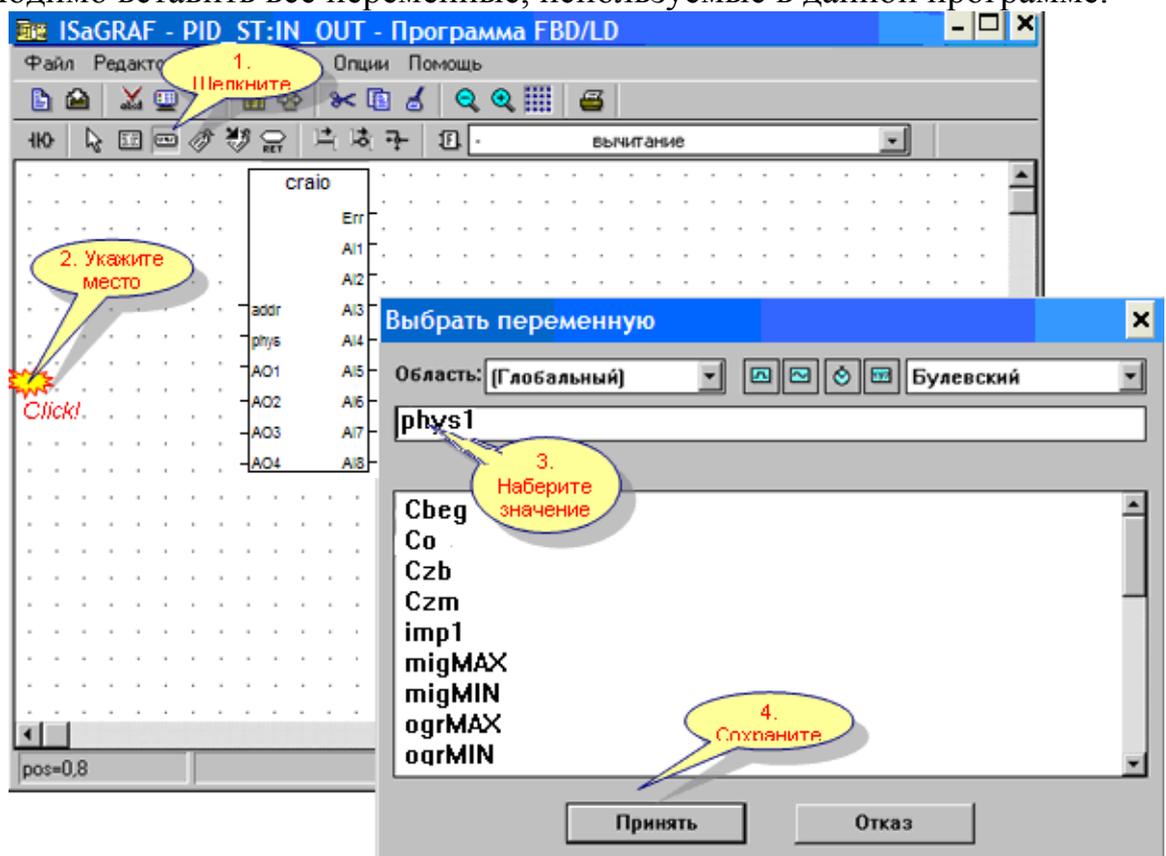


Рисунок 17. Вставка переменных

Для соединения переменной с требуемым входом функционального блока необходимо использовать кнопку НАЧЕРТИТЬ ЛИНИЮ СВЯЗИ в панели инструментов редактора FBD, щелкнуть на нужной переменной и, не отпуская кнопку мыши, провести соединяющую линию к требуемому входу функционального блока (рис. 18). Осуществляем соединение всех переменных с ФБ таким образом. Все входы и выходы ФБ должны быть соединены, в противном случае программа не будет работать. Готовая программа должна выглядеть так же как на рисунке 19.

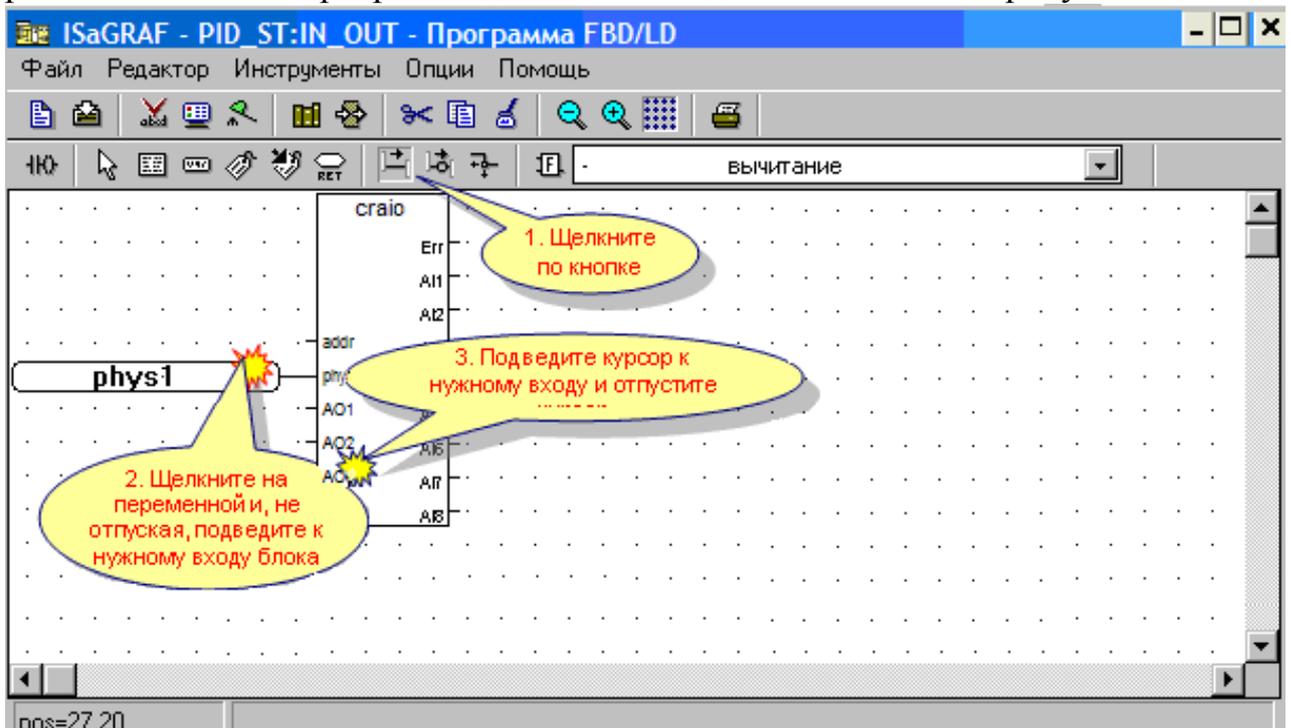


Рисунок 18. Соединение переменных с ФБ

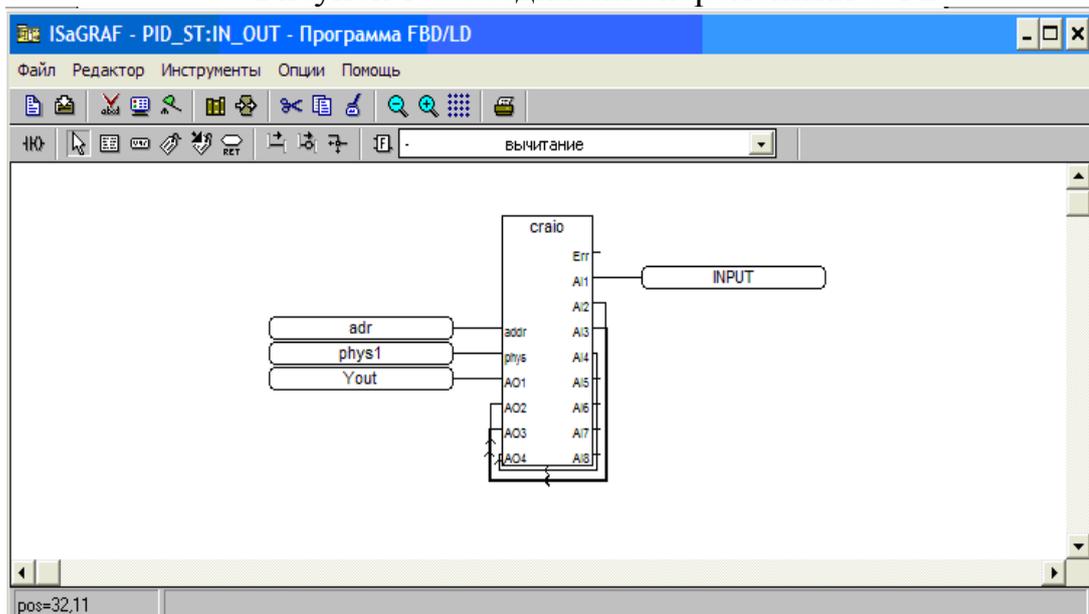


Рисунок 19. Окно программы IN_OUT

При программировании на языке FBD таким же образом вставляем все функциональные блоки, необходимые для реализации ПИД-регулятора и сигнализации и осуществляем их конфигурацию, как описывалось выше.

Затем сохраняем программу и закрываем. При программировании на языке структурированного текста ST следует создать программу PIDREG, где описывается непосредственно уравнение ПИД-регулятора и сигнализация.

Рассмотрим особенности программирования на языке ST. Программа, составленная на языке ST – это список ST операторов. Каждый оператор заканчивается точкой с запятой. Имена используемых в исходном коде (идентификаторы переменных, константы, ключевые слова) разделены неактивными разделителями (пробелами, символами окончания строки и табуляции) или активными разделителями, которые имеют определенное значение. Например, разделитель «>» означает сравнение «больше чем». В текст могут быть введены комментарии. Комментарий должен начинаться с «(» и заканчивается «)».

Основные типы операторов языка ST:

- ✓ Оператор присвоения (variable:=expression).
- ✓ Вызов подпрограммы или функции.
- ✓ Вызов функционального блока.
- ✓ Операторы выбора (IF, THEN, ELSE, CASE).
- ✓ Итеративные операторы (FOR, WHILE, REPEAT).
- ✓ Управляющие операторы (RETURN, EXIT).
- ✓ Специальные операторы для связи с такими языками как SFC

Выражения ST объединяют операторы и операнды (переменные или константы). Для одного выражения (объединяющего операнды с одним ST оператором), тип операндов должен быть одним и тем же. Это одно выражение имеет тот же тип, что и его операторы и может быть использовано в более сложном выражении.

Стандартные ST вызовы функций могут быть использованы для каждого из следующих объектов: подпрограммы, библиотечные функции и функциональные блоки, «C» функции и функциональные блоки, функции преобразование типов [3].

Реализация любой задачи на языке структурированного текста осуществляется при помощи вызова функциональных блоков, в нашем случае для реализации уравнения ПИД – регулятора.

Для описания необходимых блоков на языке ST необходимо изначально в словаре описать переменные этого блока, а именно входные и выходные переменные, а также ввести переменную, которая будет вызывать соответствующий блок на вкладке FB – экземпляры с именем отличным от имени функционального блока (differ – dif). Для описания переменных каждого блока потребуется использовать информацию о

типе входных и выходных переменных, для этого необходимо вызвать блок в программе FBD, двойным щелчком открыть окошко данного блока, выбрав информацию о данном блоке, перейти на замечания и ознакомиться со свойствами каждой переменной, как показано на рисунке 20.

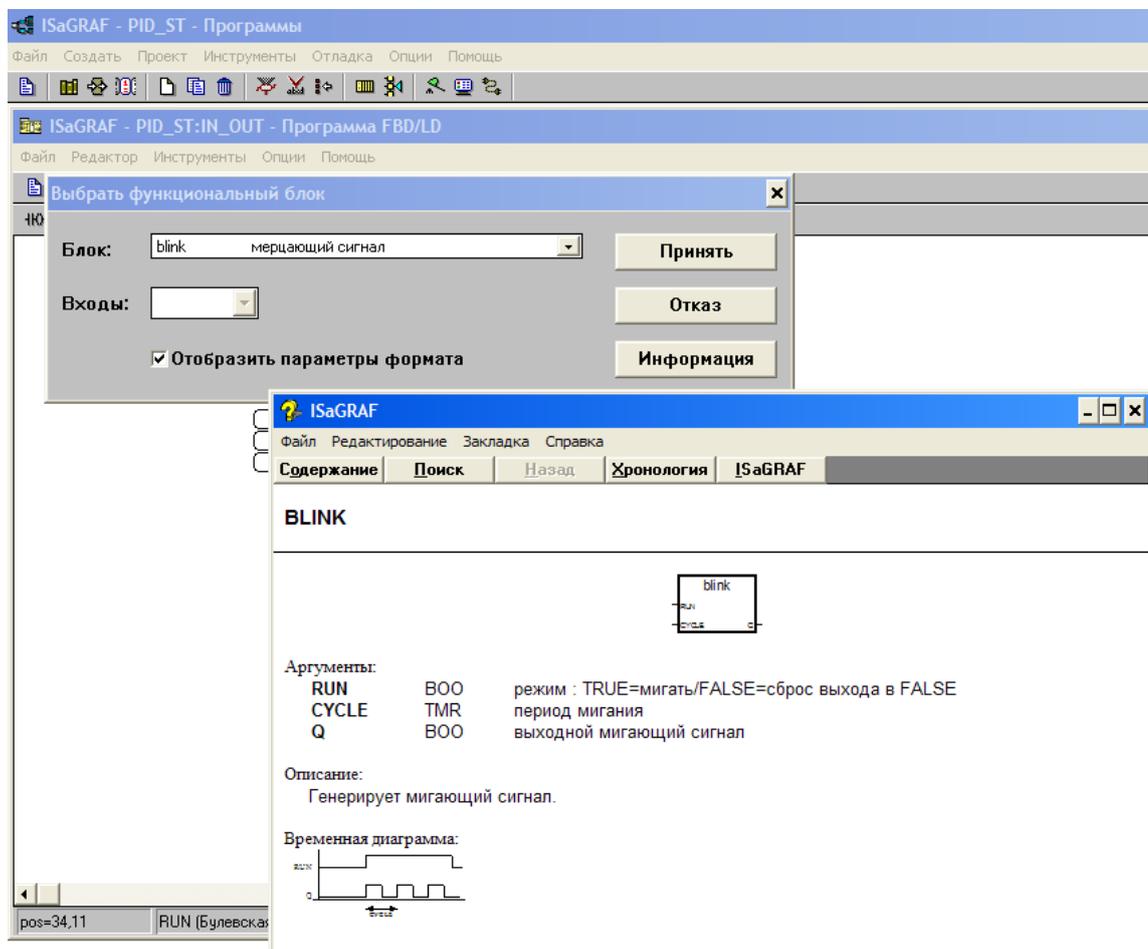


Рисунок 20 – Доступ к информации о функциональном блоке

Также необходимо помнить, что имя выходной переменной должно отличаться от имени выходной переменной существующего функционального блока, поскольку впоследствии осуществляется привязка к блоку посредством этой переменной.

После описания переменных вызываем блок следующим образом:

- ✓ в программе ST вводим новое имя функционального блока и его новые переменные в скобках.
- ✓ присваиваем значение новой выходной переменной к функциональному блоку, который описан выше с выходной переменной реального функционального блока через точку.

Также при программировании на языке ST при реализации сигнализации используется цикл условия. Структура описания цикла условия следующая: IF (условие) THEN (действие, на которое накладывается условие) ELSE (обратное действие) END_IF (конец цикла).

После завершения процедуры программирования данных программ необходимо проверить программы на ошибки. Для этого существует команда ФАЙЛ→ПРОВЕРИТЬ или кнопка . При этом открывается информационное окно состояния хода проверки программы и при отсутствии ошибок появляется соответствующее сообщение – рисунок 21.

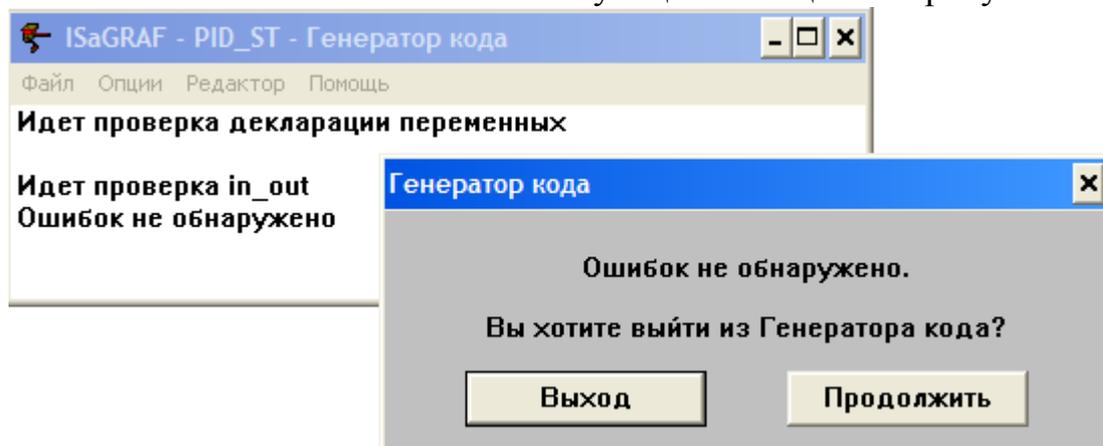


Рисунок 21 – Окно состояния проверки программы

Следующий этап – это установка связи программы с контроллером. Для этого необходимо зайти в меню ОПЦИИ/ОПЦИИ КОМПИЛЯТОРА (рис.22) и установить метку напротив TIC code for Intel для контроллера Кросс – 500. Затем открыть окно ISaGRAF-ПРОГРАММЫ, зайти в меню ОТЛАДКА/УСТАНОВЛЕНИЕ СВЯЗЕЙ или кнопкой  настройте параметры связи ПК с блоком БЦП, как приведено на рисунке 23 Установить коммуникационный порт ETHERNET, а в УСТАНОВКЕ прописать адрес 192.168.0.11 (адрес контроллера) и номер порта 1100.

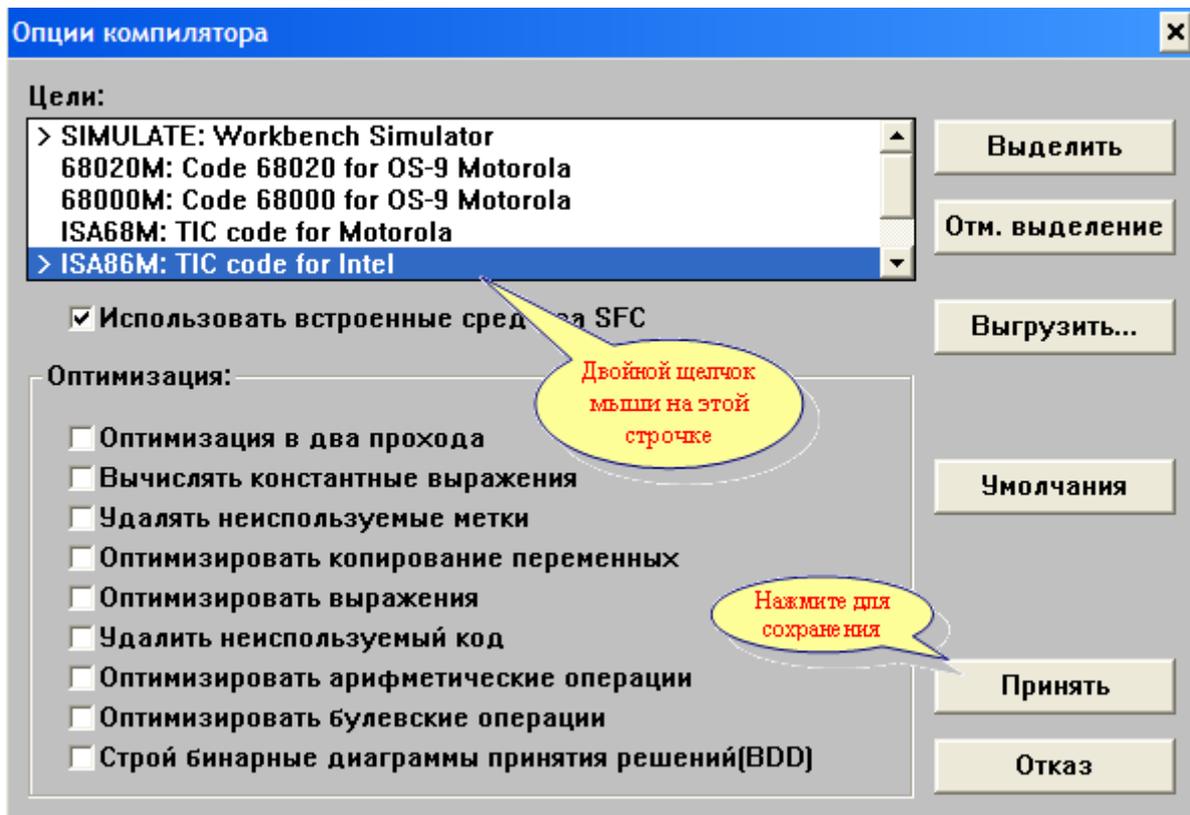


Рисунок 22 – Опции компилятора

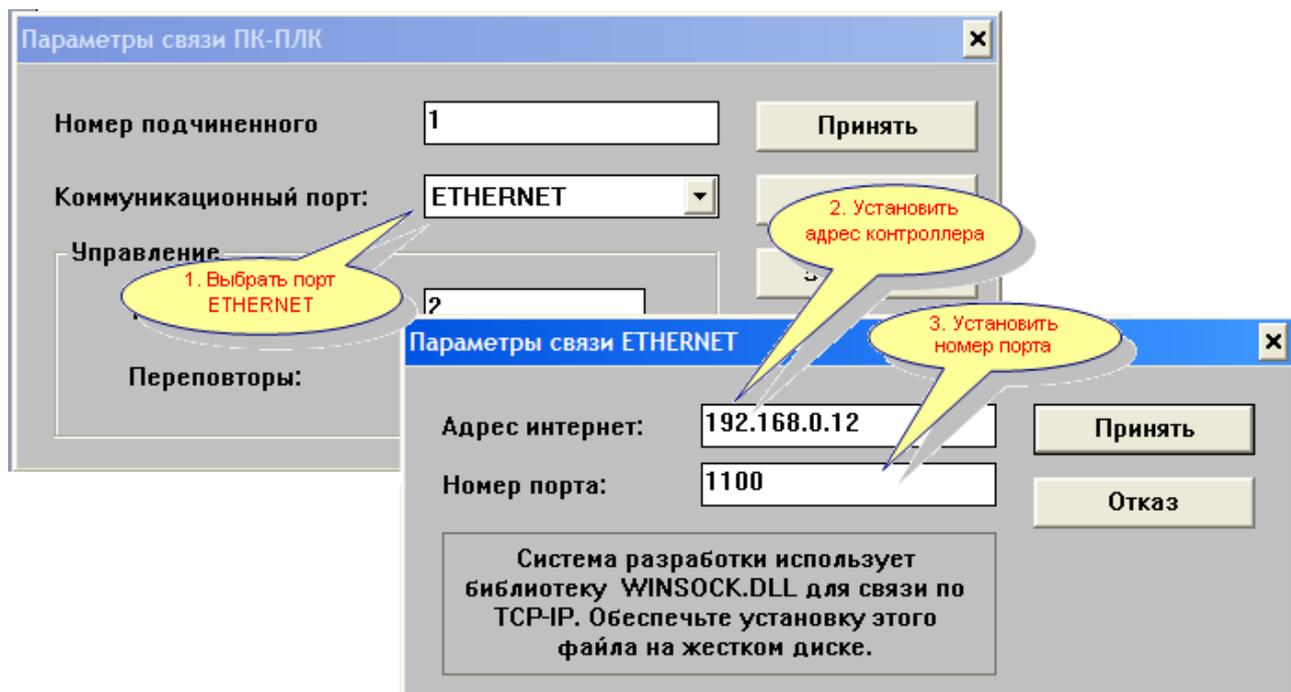


Рисунок 23 – Настройка параметров связи ПК с БЦП

Теперь необходимо создать код приложения для загрузки программы в контроллер. Для этого следует открыть окно ISaGRAF-ПРОГРАММЫ и зайти в меню СОЗДАТЬ/СОЗДАТЬ КОД ПРИЛОЖЕНИЯ или кнопкой  на панели управления. При наличии ошибок исправьте их, и проверьте программу заново.

Теперь программа регулирования готова к загрузке в контроллер. Необходимо запустить отладчик с помощью ФАЙЛ/ОТЛАДКА или кнопкой  на панели управления, при этом на экране появится окно отладчика. При наличии связи между ПК и БЦП отладчик будет иметь вид как на рисунке 24. Если в БЦП уже загружено какое-то приложение, перед загрузкой нового необходимо командой ФАЙЛ→ОСТАНОВИТЬ ПРИЛОЖЕНИЕ или кнопкой  остановить его. Загрузка готового приложения осуществляется командой ФАЙЛ→ЗАГРУЗИТЬ или . [7]

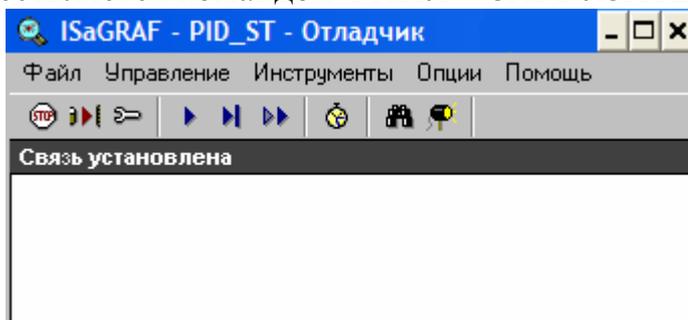


Рисунок 24 – Отладчик ISaGRAF

4 МЕТОДИКА СОСТАВЛЕНИЯ ПРИКЛАДНЫХ ПРОГРАММ С ПРИМЕНЕНИЕМ ПАКЕТА MasterScada

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 рисунке 25.



Рисунок 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 (далее 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 Master Scada

MasterScada - это не просто один из современных SCADA-пакетов, это принципиально новый инструмент разработки АСУТП, в котором реализована совокупность средств и методов, обеспечивающих резкое сокращение трудозатрат и повышение надежности создаваемой системы, а 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:

1. Единая среда разработки АСУТП.
2. Раздельное конфигурирование структуры АСУТП и логической структуры объекта.
3. Открытость и следование стандартам.
4. Интуитивная легкость освоения:
 - ✓ Удобство инструментария.
 - ✓ Удобство методики разработки.
 - ✓ Мощная трехмерная графика и мультимедиа.
 - ✓ Неограниченная гибкость вычислительных возможностей.
5. 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 древовидного 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
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;

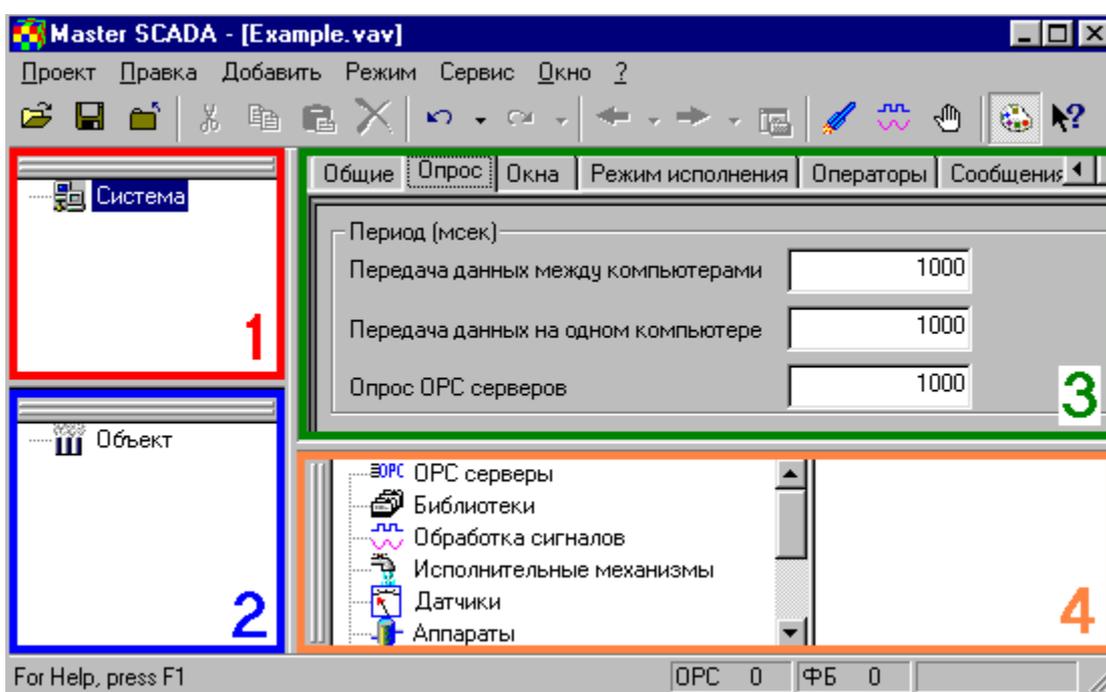


Рисунок 26 - Окно проекта

3. страницы свойств элементов, на которых производятся все необходимые настройки элементов;
4. палитра элементов, из которой берутся библиотечные объекты, функциональные блоки и т. д.

Размеры дерева системы, дерева объектов и палитры можно изменять 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.
- ✓ OPC-переменные. Используются для связи с переменными контроллера. Наследуют настройки, заданные при конфигурировании переменной в OPC-сервере. В зависимости от заданного в OPC-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 арифметические 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).
- Обработка данных .
 - Библиотеки.
 - Сценарии.
 - Мнемосхемы.
 - Тренды.
 - Сообщения.
 - Рапорты.
 - Архивы.
 - Обмен данными по сети.
 - Права доступа и контроль действий оператора.
 - Режимы работы.

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.
- Возможности отладки.
- Средства повышения надежности работы.

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 ActiveX 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.

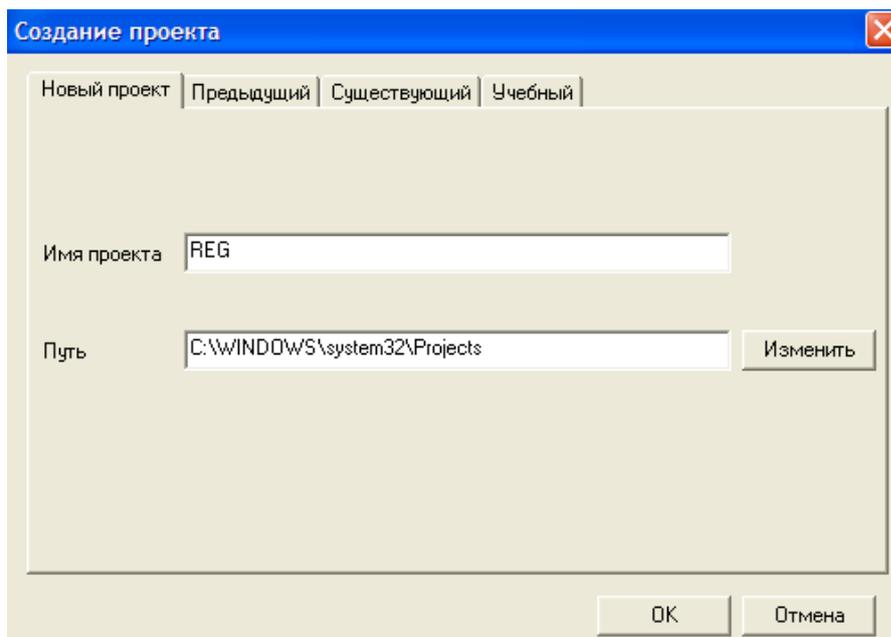


Рисунок 27 – Окно создания проекта

В итоге появится окно проекта (рис. 28).

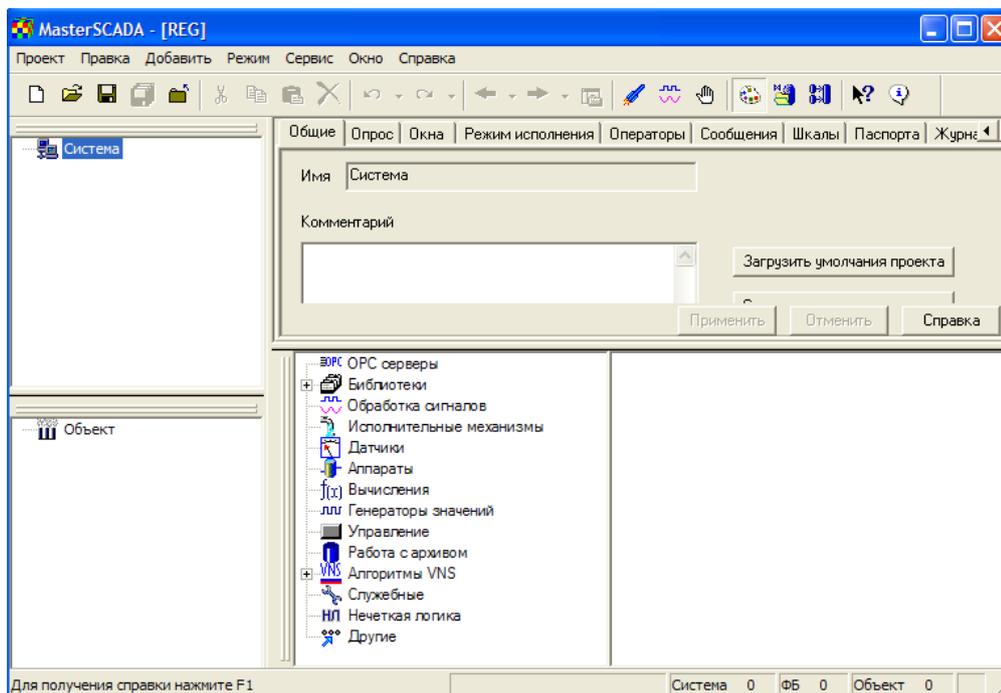


Рисунок 28 – Окно проекта

Теперь создадим соединение между контроллером и средой ISaGRAF. Для этого выделите объект «Система» в дереве системы и

щелкните по ней правой клавишей (ПК), выберете ВСТАВИТЬ/КОМПЬЮТЕР. В странице свойств элемента в поле «Имя» введите имя используемого компьютера (например titan2), не забыв нажать на кнопку ПРИМЕНИТЬ – для сохранения.

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).

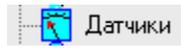
В появившемся окне «Свойства: выбор переменных» поставьте галочку, выделяя все переменные. В результате все переменные ISaGRAF, используемые при написании технологической программы пользователя будут доступны в MasterScada.

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 (ПК), выберете ВСТАВИТЬ/ОБЪЕКТ. В странице свойств ОБЪЕКТА 1 выберете компьютер «titan2», здесь же можно поменять имя объекта. Далее в палитре инструментов выберете

 , щелкните левой клавишей мышки (ЛК) по объекту  и перетащите его в «Объект 1». В итоге дерево объектов будет выглядеть следующим образом (рис. 29):

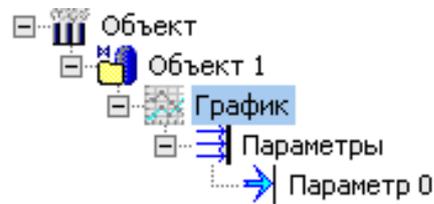
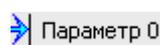


Рисунок 29 – Дерево объектов

На странице свойств элемента «График» войдите на закладку «Настройки» и укажите ЧИСЛО ПАРАМЕТРОВ. Нажмите кнопку ПРИМЕНИТЬ. Затем из дерева системы выберете переменную t1_XIN и перетащите ее в . 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, выберете ПЕРЕЙТИ НА/МНЕМОСХЕМА.

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

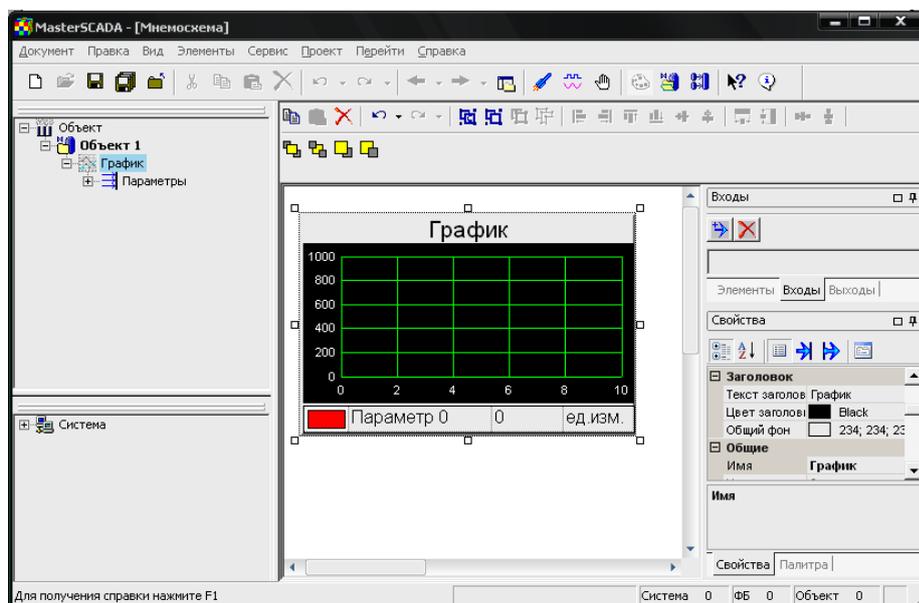


Рисунок 30 - График

В правом нижнем углу редактора мнемосхем отображаются свойства элементов мнемосхемы, которые позволяют изменять, 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».

4.3.3 Создание команд

В дереве системы выделите «Объект 1», щелкните ПК и выберете ВСТАВИТЬ/КОМАНДА. Команды позволяют изменять значения переменных в режиме работы. Перетащите необходимую переменную, например t1_КР (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. 3.3.8.3 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, иллюстрация на рисунке 31.

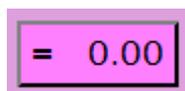


Рисунок 31 – Значение переменной

4.3.5 Создание кнопок

Для того чтобы создать кнопку, необходимо перейти на палитру инструментов, выбрать УПРАВЛЕНИЕ/КНОПКА и перетащите 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 кнопочку.

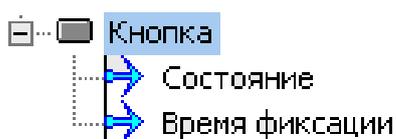


Рисунок 32. – Кнопка

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).

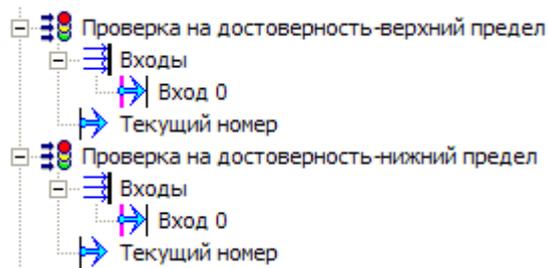


Рисунок 33 – Индикаторы состояния

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 раскрывающемся 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, длина шкалы,

тип значения, выбор цветовой сигнализации стрелочного прибора, цветное решение прибора и многое другое (рис. 34).

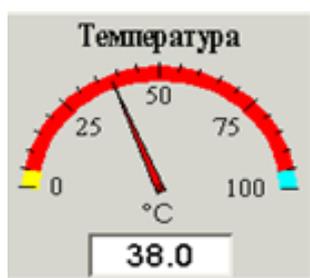


Рисунок 34 – Стрелочный прибор

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).



Рисунок 35 – Индикатор значения

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 данный элемент управления на мнемосхеме: перетащим дискретную переменную (значение переменной из пункта 3.3.4) в мнемосхему ПК мыши, и в раскрывающемся контекстном меню выберем «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 рисунке 36 и 37.



Рисунок 36 – Индикатор графический

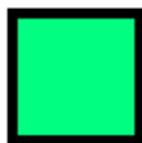


Рисунок 37 – Индикатор мнемонический

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 рисунке 38.

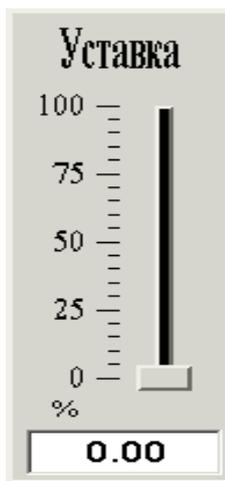


Рисунок 38 – Задатчик

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» (рис. 39).

На закладке «Общие» страницы свойств элементов изменяем название таблицы, добавляем комментарии, изменяем число входов и выходов таблицы на закладке "Параметры". Таблицу перетаскивается 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 мышью и изменить оформление таблицы (рис. 39).

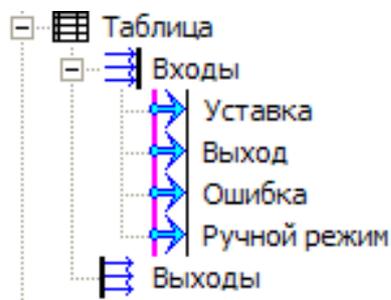


Рисунок 39 – Таблица

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 её будет заключаться в наличии кнопок с переходами на другие мнемосхему, 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 мышью, 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 кнопочках. Кроме того, необходимо изменить курсор с «Исходного размера» на «Растягивать» для корректировки размера кнопочек. Именно так создаются кнопки переходов на всех мнемосхемах.

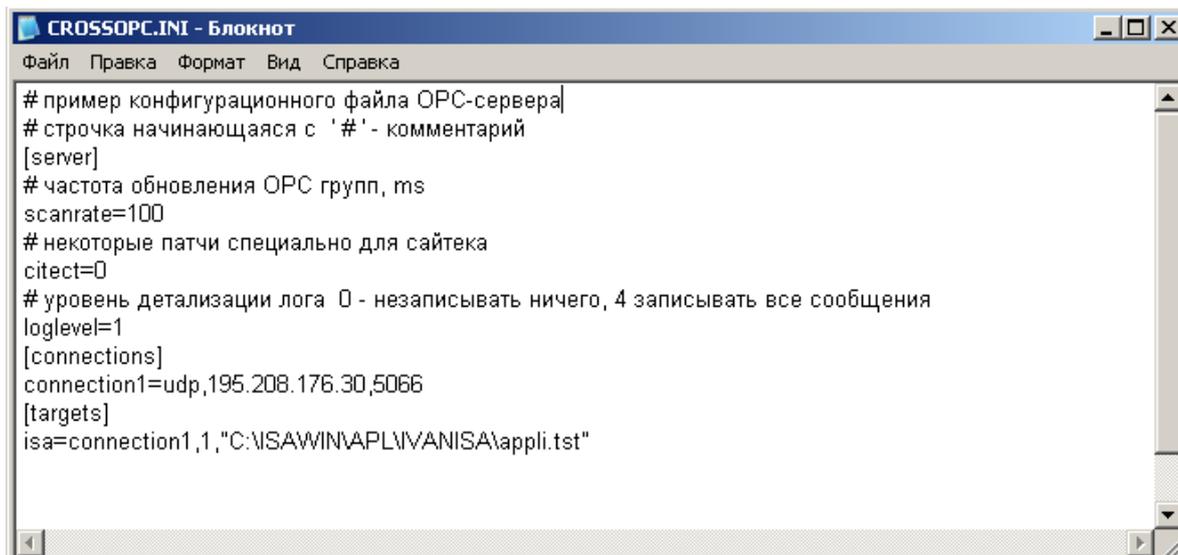
Также на главной мнемосхеме должна присутствовать кнопочка, осуществляющая закрытие мнемосхем для этого в данном объекте создается, при помощи ПК мыши ВСТАВИТЬ, событие и команда, позволяющие осуществить закрытие мнемосхемы.

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».

5 НАЗНАЧЕНИЕ OPC – СЕРВЕРА И ЕГО НАСТРОЙКА

OPC – сервер для контроллеров предназначен для обмена данными между технологической программой пользователя контроллера КРОСС-500 и SCADA-системами через интерфейс, определяемый спецификацией OPC Data Access 2.0.

Настройка OPC производится путем редактирования файла `crossopc.ini`, расположенного в каталоге установки OPC – сервера. Это текстовый файл в стиле стандартных ini-файлов Windows, расположенный в следующей директории: `C:\Program files\ OPC Server\CROSSOPC`. Пример конфигурационного файла представлен на рисунке 40.



```
CROSSOPC.INI - Блокнот
Файл  Правка  Формат  Вид  Справка

# пример конфигурационного файла OPC-сервера
# строка начинающаяся с '#' - комментарий
[server]
# частота обновления OPC групп, ms
scanrate=100
# некоторые патчи специально для сайтека
citect=0
# уровень детализации лога 0 - не записывать ничего, 4 записывать все сообщения
loglevel=1
[connections]
connection1=udp,195.208.176.30,5066
[target]
isa=connection1,1,"C:\ISAWIN\APL\IVANISA\appli.tst"
```

Рисунок 40 – Пример конфигурационного файла OPC - сервера

Каждая строка содержит либо комментарий, начинающийся с символов «#» или «;» (точка с запятой), либо заголовок секции, либо набор параметров секции. Имена параметров к регистру нечувствительны. Описание секций:

1. Секция [SERVER] может содержать следующие параметры:

- ✓ LOGFILE = <file >. Указывает имя файла для записи протокола работы сервера. Имя файла необходимо указывать в кавычках.
- ✓ LOGLEVEL = <level> Число от 0 до 3, показывающее, какие сообщения будут выводиться в лог. Допустимы следующие значения.
 - 0 – не выводить сообщения совсем,
 - 1 – только ошибки,
 - 2 – ошибки и предупреждения,
 - 3 – ошибки, предупреждения и диагностические сообщения.
- ✓ SCANRATE = <rate> Целое число, показывающее период опроса контроллера в миллисекундах.

- ✓ CИТЕСТ= <0/1> Если используется SCADA-система Citect, то этот параметр необходимо установить в 1. Этот ключ заставляет OPC сервер принудительно устанавливать для данных, передаваемых в Citect OPC_QUALITY_GOOD, потому что OPC-драйвер Citect работает следующим образом: при поступлении хотя бы одного элемента с OPC_QUALITY_BAD остальные элементы тоже рассматриваются как содержащие недостоверные данные.

2. Секция [CONNECTIONS] содержит определения логических подключений. Формат:

- ✓ <connection_name>=SERIAL,<port>,<speed> для подключения через последовательный порт
- ✓ <connection_name>=UDP,<ip_address>,<ip_port> для подключения через TCP/IP

Параметры:

- < connection_name > - имя подключения;
- < port > - имя последовательного порта, например COM4;
- < speed > - скорость порта в бодах;
- < ip_address > - ip – адрес контроллера или dns имя;
- < ip_port > - ip – порт (по умолчанию 5066).

3. Секция [TARGETS] содержит определения задач (targets). Под задачей подразумевается исполнительная система ISaGRAF, выполняемая на контроллере. Формат:

- ✓ <target_name>=<connection_name>,<logical_num>,"<имя конфигурационного файла>"

Параметры:

- <target_name> - имя целевой задачи;

- <connection_name> - имя подключения, определенное в секции [connections];
- <logical_num > - логический номер задачи, обычно 1;
- <имя конфигурационного файла> - путь к файлу, который содержит определения переменных целевой задачи ISaGRAF.

В данном случае, путь к программе PID_ST описывается как t1=connection, 1, “C:\isawin\APL\PID_ST\ appli.tst”.

Необходимо запустить OPC сервер C:\Program Files\КРОСС\OPC\crossopc.exe для привязки OPC – переменных.

6 ПРИМЕР РАЗРАБОТКИ ПРОГРАММЫ РЕГУЛИРОВАНИЯ И СИГНАЛИЗАЦИИ

В данном примере в качестве языка программирования используется технологический язык ST. Существует несколько вариантов по структуре программного обеспечения. Первый вариант предполагает разработку программы на языке ST, обеспечивающей ввод и вывод сигналов, а также регулирование и сигнализацию. Пример такой программы приведен в приложениях 3, 4. Второй вариант предполагает разработку двух программ, одна из которых обеспечивает ввод и вывод сигналов, а вторая – регулирование и сигнализацию. Причем, допускается разработка программы ввода- вывода сигналов на языке FBD. в данном примере созданный проект PID_ST содержит в себе две программы: IN_OUT, составленной на языке FBD, и PIDREG, составленной на языке ST. Программу IN_OUT используется для ввода-вывода аналоговых сигналов с АВК-6, а в программе PIDREG обеспечивает регулирование и сигнализацию.

Сначала представляем таблицу 8 переменных, которые были объявлены в словаре, с полным описанием: тип данных, атрибут, формат,

начальное значение и комментарий, определяющие роль той или иной переменной в нашей программе.

Таблица 8 – Описание переменных

Имя переменной	Тип данных	Атрибуты/ Тип	Формат	Начальное значение	Комментарии
phys1	Булевские	константа		false	признак необходимости преобразования ФБ craio
Co		внутренняя		false	обнуление - вход ФБ dif
Czb		константа		false	сигнал запрета «Больше» - вход ФБ integr
Czm		константа		false	сигнал запрета «Меньше» - вход ФБ integr
Cbeg		константа		false	команда установки начальных условий - вход ФБ integr
migMAX		внутренняя		false	выход ФБ blink для достоверности данных – верхней предел
migMIN		внутренняя		false	выход ФБ blink для достоверности данных – нижней предел
ogrMAX		внутренняя		false	режим работы ФБ blink для достоверности данных – верхней предел
ogrMIN		внутренняя		false	режим работы ФБ blink для достоверности данных – нижний предел
runMAX		внутренняя		false	режим работы ФБ blink для мигания формы сигнализации по верхнему пределу
runMIN	внутренняя		false	режим работы ФБ blink для мигания формы сигнализации	

					по нижнему пределу
sigMAX		внутренняя		false	выход ФБ blink для мигания формы сигнализации по верхнему пределу
sigMIN		внутренняя		false	выход ФБ blink для мигания формы сигнализации по нижнему пределу
adr	Целые - действительные	константа	целая	230	адрес модуля ФБ craio
Xin		внутренняя	веществ .	0	выход
Yout		внутренняя	веществ .	0	управляющее воздействие
e		внутренняя	веществ .	0	ошибка регулирования
Xzdn		внутренняя	веществ .	20	задание
Td		внутренняя	веществ .	0	Постоянная времени дифференцирования - вход ФБ dif
km		константа	веществ .	1	коэффициент усиления - вход ФБ dif
Xmax		константа	веществ .	5000	уровень ограничения по максимуму - вход ФБ integr
Xmin		константа	веществ .	-5000	уровень ограничения по минимуму - вход ФБ integr
Xbeg		константа	веществ .	0	значение начальных условий - вход ФБ integr
Xpor		константа	веществ .	1000	установка порогового элемента - вход ФБ integr
Ti		внутренняя	веществ .	0,05	постоянная времени интегрирования - вход ФБ integr
Ydif		внутренняя	веществ .	0	выход ФБ dif
Yint		внутренняя	веществ .	0	выход ФБ integr

Кр		внутренняя	веществ .	0,5	постоянная времени пропорциональности
INPUT		внутренняя	веществ .	0	вход ФБ cr aio
imp_real		внутренняя	веществ .	0	выход ФБ real
INPUTmax		внутренняя	веществ .	50	верхний предел сигнализации
INPUTmin		внутренняя	веществ .	0	нижний предел сигнализации
X1max		внутренняя	веществ .	100	верхний предел достоверности данных
X1min		внутренняя	веществ .	0	нижний предел достоверности данных
tm3	Таймеры	внутренняя		t#3s	период мигания ФБ blink для сигнализации верхн. предела
tm4		внутренняя		t#3s	период мигания ФБ blink для сигнализации нижнего предела
tm5		внутренняя		t#3s	период мигания ФБ blink при недостоверности данных – верхней предел
tm6		внутренняя		t#3s	период мигания blink при недостоверности данных – нижний предел
differ integrir ogranMAX ogranMIN signalMAX signalMIN	ФВ экземп ляры	dif integr blink blink blink			Вызов ФБ для реализации уравнения ПИД – регулятора, осуществления сигнализации и проверки на достоверность данных.
signalMIN		blink			

Программа IN_OUT, используемая для ввода-вывода аналоговых сигналов с АВК-6, представлена на рисунке 41.

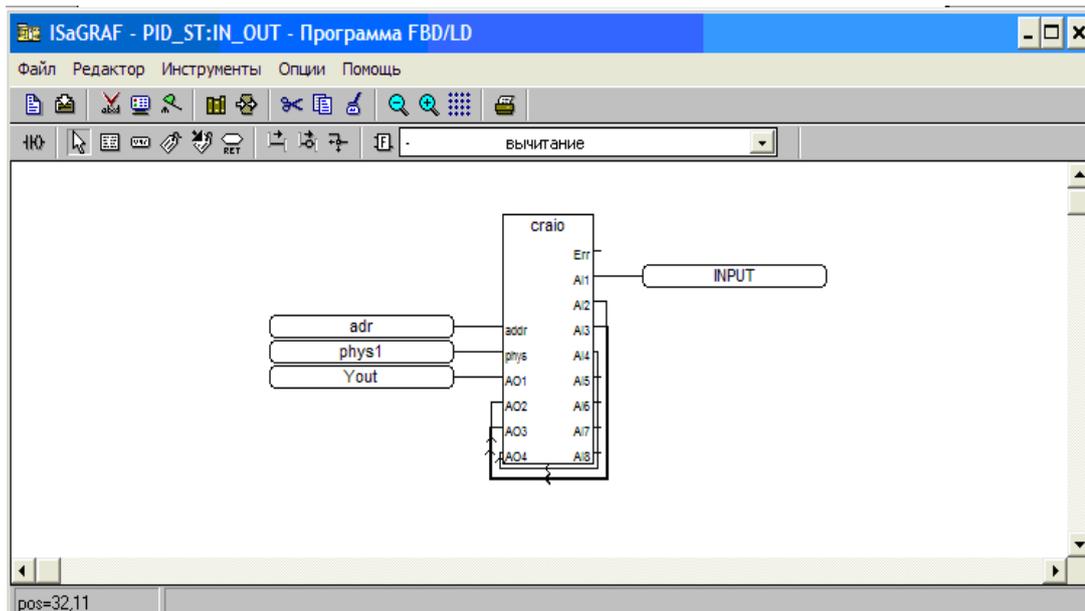


Рисунок 41 – Окно программы IN_OUT

Программа PIDREG описывает программа регулирования и сигнализации. В данной программе согласно варианту необходимо описать уравнение ПИД - регулятора, осуществить сигнализацию по верхнему и нижнему предельным значениям, мигание формы при выходе параметра за установленные пределы, а также осуществить проверку на достоверность данных. Реализуется это при помощи вызова функциональных блоков и использовании цикла для введения ограничений по максимуму и минимуму.

Осуществляем вызов следующих функциональных блоков:

- ✓ blink – для осуществления мигания формы при выходе параметра за установленные пределы;
- ✓ dif – для ввода дифференциальной составляющей ПИД – регулятора;
- ✓ integr – для ввода интегральной составляющей ПИД – регулятора.

Следуя методики программирования, описанной выше, получаем следующую программу:

*(*Описание уравнения ПИД - регулятора*)*

```

Xin:=INPUT; - присвоение текущего значение новой переменной;
(*Ошибка регулирования определяется как разность между
заданным и текущим
значениями сигнала *)
e:= Xzdn - Xin;
(*Вызов ФБ dif для ввода дифференциальной составляющей ПИД –
регулятора*)
differ(e, km, Td, Co);
Ydif:=differ.Y;
(*Вызов ФБ integr для ввода интегральной составляющей ПИД –
регулятора *)
integrir(e, Ti, Cbeg, Xbeg, Xmax, Xmin, Xpor, Czb, Czm);
Yint:=integrir.Y;
(*Уравнение ПИД-регулятора*)
Yout:=Kp*(e+Yint+Ydif);

(*Ограничение по максимуму*)
if (Xin>INPUTmax) then – условие ограничения по максимуму;
Xin:=INPUTmax; - действие, на которое накладывается условие;
runMAX:=true; - привязка к переменной, которая будет
использоваться в MasterScada;
else runMAX:=false; - обратное действие;
end_if; - конец цикла;
(*Ограничение по минимуму*)
if (Xin<INPUTmin) then – условие ограничения по минимуму;
Xin:=INPUTmin; - действие, на которое накладывается условие;
runMIN:=true; ; - привязка к переменной, которая будет
использоваться в MasterScada;
else runMIN:=false; - обратное действие;
end_if; - конец цикла;

(*Вызов ФБ blink для реализации мигания формы при выходе
параметра за верхний предел*)
signalMAX(runMAX,tm3);
sigMAX:=signalMAX.Q;
(*Вызов ФБ blink для реализации мигания формы при выходе
параметра за нижний предел*)
signalMIN(runMIN,tm4);
sigMIN:=signalMIN.Q;

(*Введение проверки на достоверность данных*)
(*Ограничение по максимуму*)
if (INPUT>X1max) then

```

```

INPUT:=X1max;
ogrMAX:=true;
else ogrMAX:=false;
end_if;
(*Ограничение по минимуму*)
if (INPUT<X1min) then
INPUT:=X1min;
ogrMIN:=true;
else ogrMIN:=false;
end_if;

```

```

(*Вызов ФБ blink для реализации мигания формы при
недостоверности данных –
при выходе за верхний предел*)
ogranMAX(ogrMAX,tm5);
migMAX:=ogranMAX.Q;
(*Вызов ФБ blink для реализации мигания формы при
недостоверности данных –
при выходе за нижний предел*)
ogranMIN(ogrMIN,tm6);
migMIN:=ogranMIN.Q.

```

Программа PIDREG, общий вид которой представлен на рисунке 42, позволяет вести регулирование аналоговым сигналом, снятым с АВК-6, по ПИД-закону, ограничивает входной сигнал на максимум и минимум, осуществляет мигание формы при выходе параметра за установленные пределы, а также осуществляет проверку на достоверность данных.

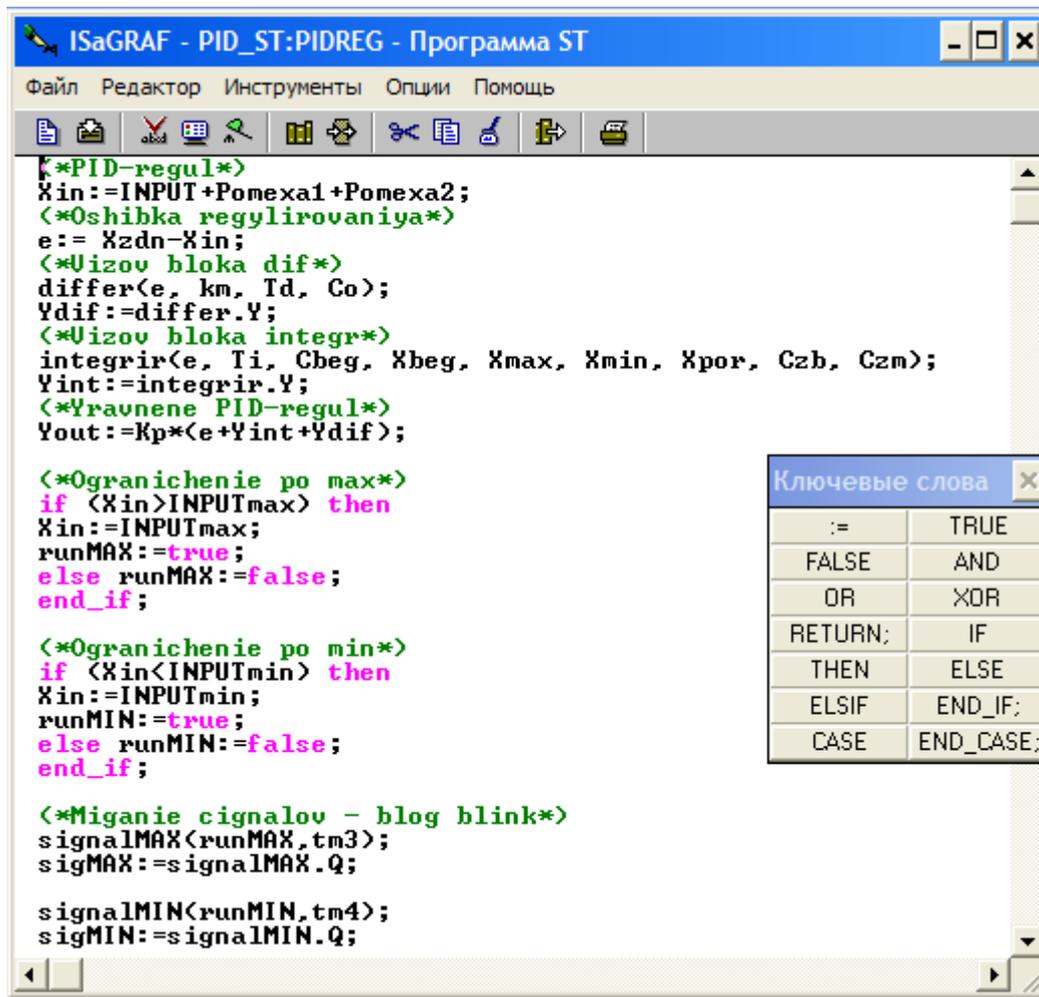


Рисунок 42 – Общий вид программа PIDREG

7 ПРИМЕР РАЗРАБОТКИ ПРОГРАММЫ ВИЗУАЛИЗАЦИИ ПРОЦЕССОВ КОНТРОЛЯ, РЕГУЛИРОВАНИЯ И СИГНАЛИЗАЦИИ

Программа визуализации процессов контроля, регулирования и сигнализации выполнена в пакете MasterScada.

Программа визуализации создана при помощи различных функциональных элементов данного пакета, среди которых имеется возможность создания трендов, кнопок, цветовых индикаторов, команд, графических элементов, представлена на рисунке 43:

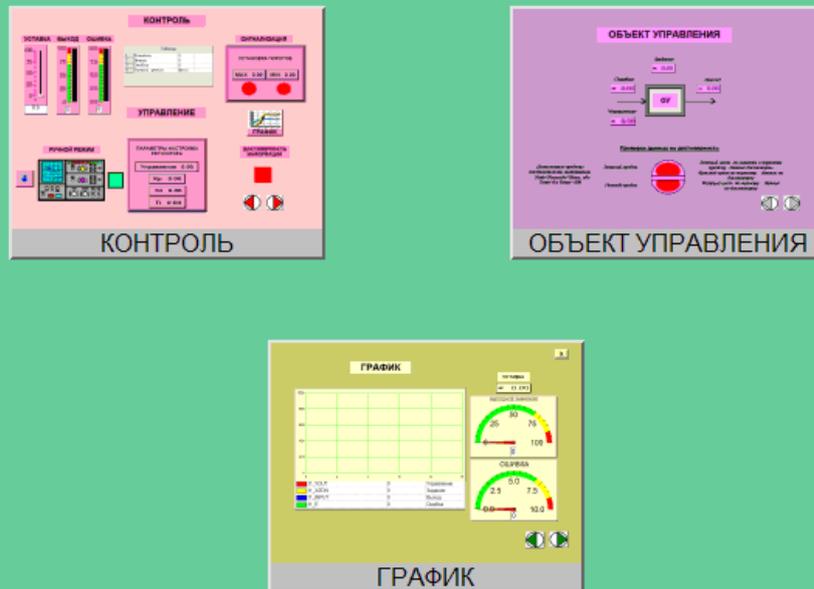
ПИД - РЕГУЛЯТОР

Рисунок 43 – Внешний вид главной мнемосхемы программы визуализации

Данная программа содержит несколько мнемосхем, первая из которых мнемосхема КОНТРОЛЬ, представлена на рисунке 44 и обеспечивает контроль и управление процессом регулирования и сигнализации. Дерево объекта, представленное на рисунке 45, позволяет подробнее ознакомиться с разработанной мнемосхемой.

На данной мнемосхеме возможно изменение задания (уставки) в виде задатчика, установка верхнего и нижнего порогов сигнализации (команды max и min). На данной мнемосхеме осуществляется выбор ручного режима, при котором возможно вручную подбирать параметры настройки, при этом, изменяя управление, которое также отражается на мнемосхеме в цифровой форме. Реализован ручной режим при помощи анимационного регулятора (мультфильм), кнопки, запускающей ручной

режим и индикатора мнемонического, сигнализирующего о запуске ручного режима.

Также мнемосхема дает информацию о выходном значении и о значении ошибки, как в цифровом виде в таблице, так и в виде индикаторов значения. Присутствует на мнемосхеме сигнализация о нарушении достоверности данных, которая реализована таким же образом, как и сигнализация по верхнему и нижнему пределу, при срабатывании мигающей цветовой сигнализации необходимо сразу же проверить выходное значение, а также пороги, проверяющие данные на достоверность. Также в случае необходимости эта мнемосхема снабжена переходом на мнемосхему ГРАФИК, по соответствующей кнопке, где можно проанализировать все величины в графической форме.

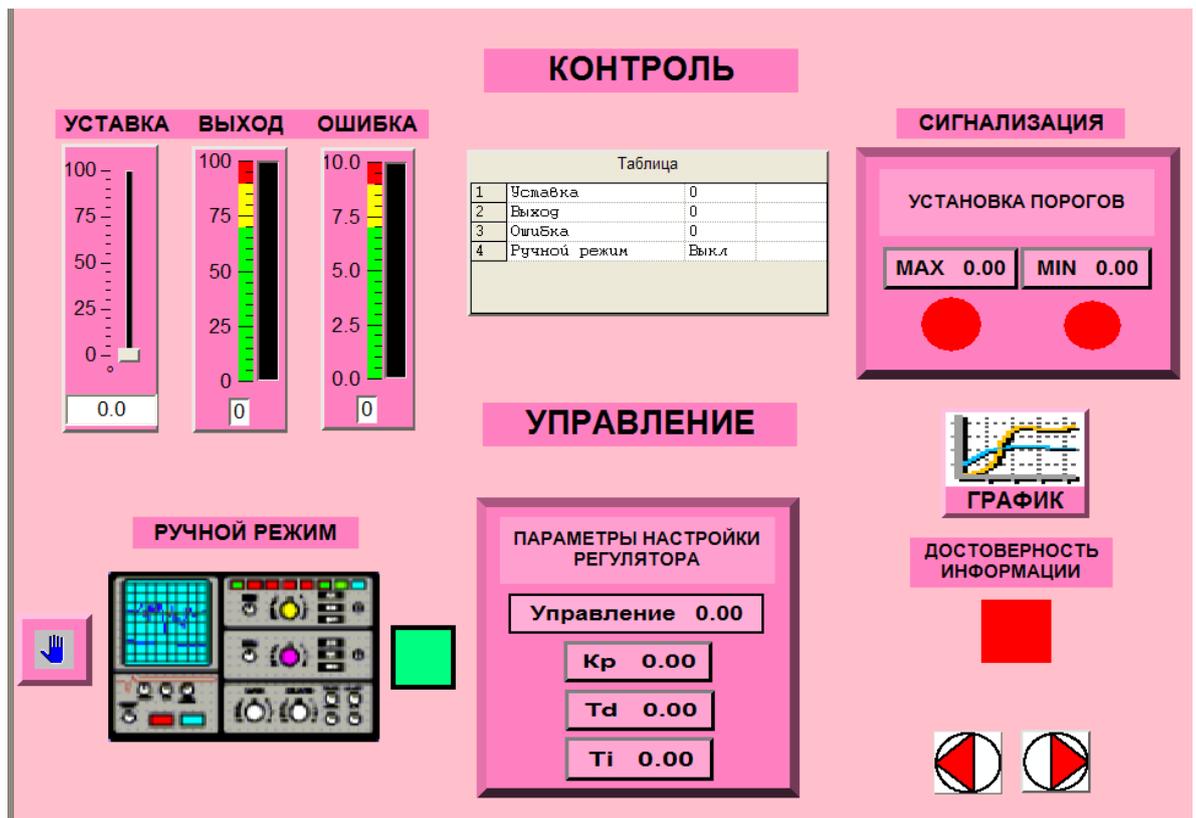


Рисунок 44 – Внешний вид мнемосхемы КОНТРОЛЬ

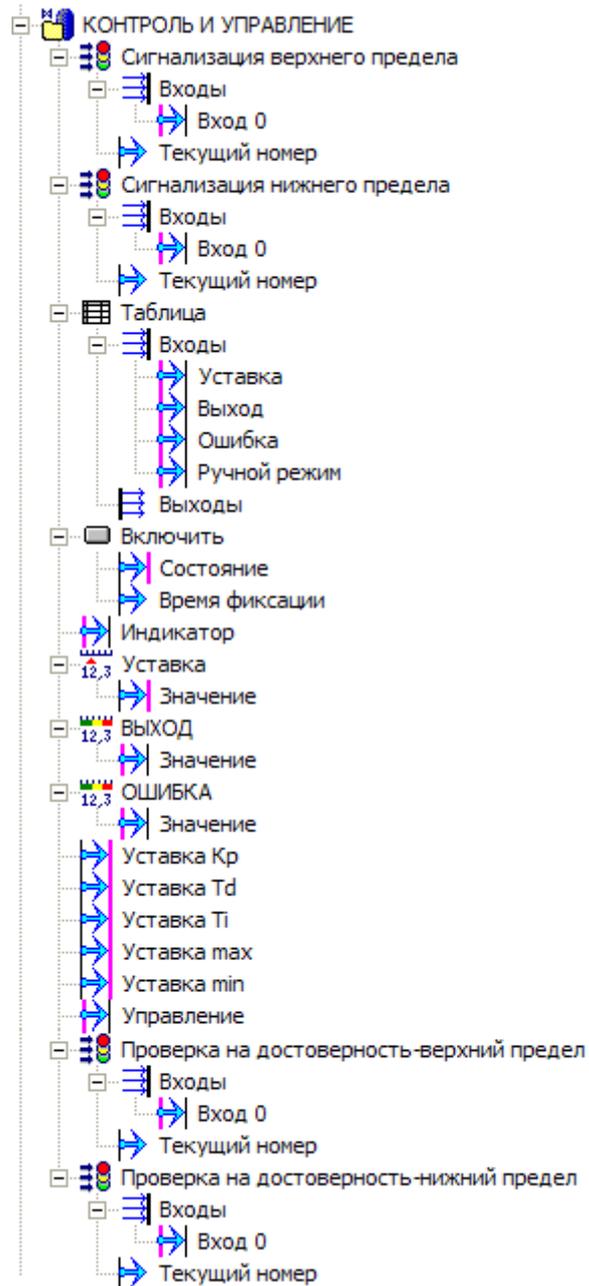


Рисунок 45 – Дерево объекта «КОНТРОЛЬ И УПРАВЛЕНИЕ»

Следующая мнемосхема носит название - ОБЪЕКТ УПРАВЛЕНИЯ, она представлена на рисунке 46, дерево объекта данной мнемосхемы представлено на рисунке 47.

Поскольку конкретного объекта управления задано не было, визуализация заключалась в создании графической интерпретации объекта управления (прямоугольник с кантом) с входными и выходными значения, а также значением ошибки и управляющего воздействия

(значения непосредственно с дерева системы). Также на данной мнемосхеме приведена информация о первичной обработке данных, а именно проверка данных на достоверность, при нарушении которой срабатывает мигающая цветовая сигнализация. Приведена полная информация о порогах, проверяющие данные на достоверность, а также о цветовой сигнализации. Реализация вышеперечисленных элементов визуализации сопровождается созданием разнообразных графических примитив (статического текста, эллипсов, линий).



Рисунок 46 – Внешний вид мнемосхемы ОБЪЕКТ УПРАВЛЕНИЯ

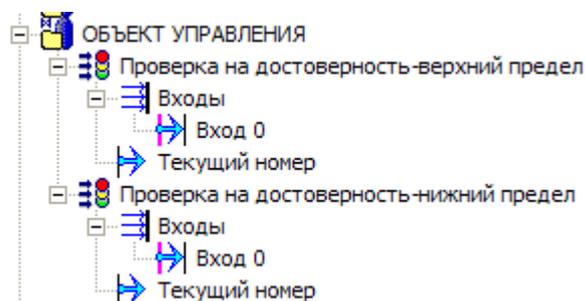


Рисунок 47 – Дерево объекта «ОБЪЕКТ УПРАВЛЕНИЯ»

Последняя мнемосхема – ГРАФИК проиллюстрирована на рисунке 48. Дерево объекта данной мнемосхемы представлено на рисунке 49.

Данная мнемосхема представляет собой график с четырьмя выходными величинами: управление, задание, выход и ошибка. Также здесь можно увидеть уставку в цифровой форме (значение переменной), а также выходное значение и ошибку в виде стрелочных приборов.

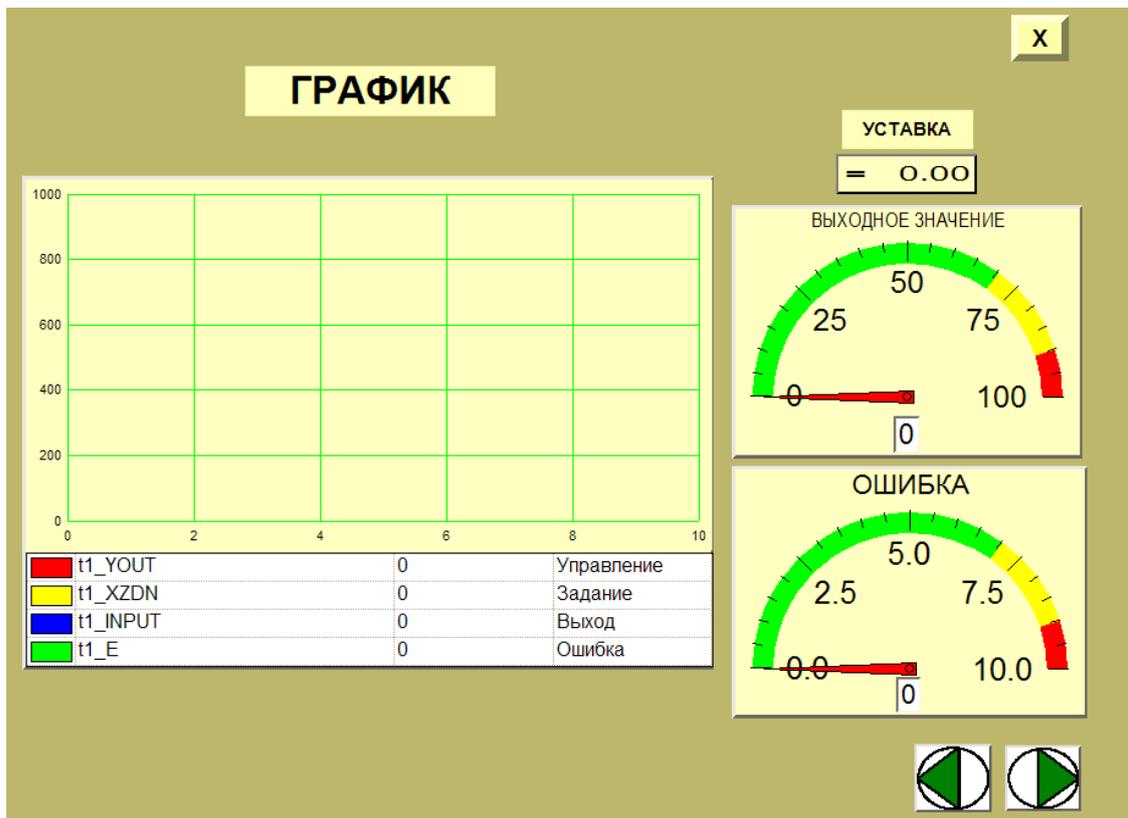


Рисунок 47 – Внешний вид мнемосхемы ГРАФИК

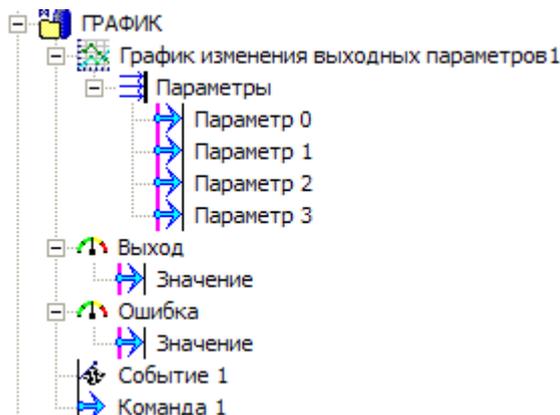


Рисунок 48 – Дерево объекта «ГРАФИК»

Переход с главной мнемосхемы осуществляется по кнопочкам с названием мнемосхем, а переходы между мнемосхемами осуществляются посредством кнопочек ДАЛЕЕ и НАЗАД. Лишь при переходе с мнемосхемы КОНТРОЛЬ на ГРАФИК, для просмотра графика, необходимо закрыть мнемосхему, нажав на крестик вверху мнемосхемы, так как переход по кнопочках ДАЛЕЕ и НАЗАД осуществляться не будет. Закрытие мнемосхем возможно при помощи кнопочки ЗАКРЫТЬ МНЕМОСХЕМУ на главной мнемосхеме.

8 ПРОВЕРКА РАБОТОСПОСОБНОСТИ РАЗРАБОТАННЫХ ПРОГРАММ

После того как все программы созданы, необходимо осуществить проверку на работоспособность разработанных программ. Проверка проводилась на лабораторном стенде, структурная схема которого представлена на рисунке 49.

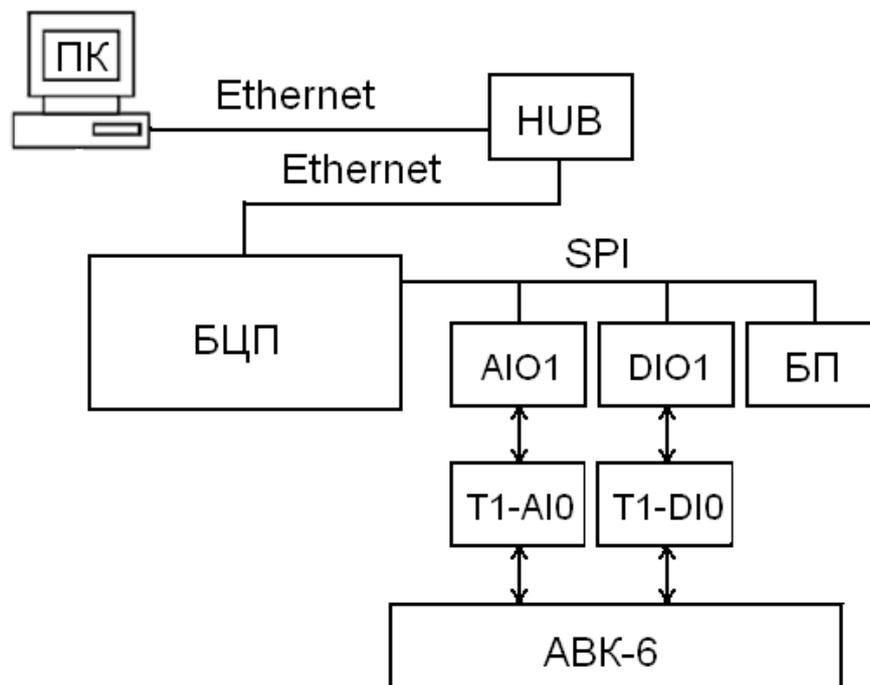


Рисунок 49 – Структурная схема лабораторного стенда

На рисунке введены следующие обозначения:

БЦП – блок центрального процессора контроллера Кросс – 500.

SPI – внутрисистемный интерфейс.

AIO1 – модуль аналогового ввода/вывода контроллера Кросс – 500.

DIO1 – модуль дискретного ввода/вывода контроллера Кросс – 500.

T1AIO, T1DIO – терминальные блоки модулей ввода/вывода аналоговых и дискретных сигналов контроллера Кросс – 500.

БП – блок питания.

АВК – 6 – аналого-вычислительный комплекс.

ПК – персональный компьютер.

HUB – концентратор сети Ethernet.

Методика проверки на лабораторном стенде заключается в реализации системы регулирования, выполненной на базе контроллера Кросс – 500, модели объекта управления, набранной на аналого-вычислительном комплексе АВК – 6 с применением разработанной программы визуализации. Объектом является последовательное

соединение двух апериодических звеньев. Звенья реализованы на базе интеграторов, охваченных обратной связью. Коэффициенты обратной связи равны $K_{oc} = -1$, коэффициенты усиления: $K = 1$.

Проверка осуществляется в два этапа. На первом этапе проверяется работоспособность программа контроллера, а на втором этапе осуществляется комплексная проверка, в которой участвуют программы контроллера и программы визуализации.

Последовательность проверки:

1. Первый шаг – загрузка программы регулирования и сигнализации, созданную в программном пакете ISaGRAF в контроллер. Для этого как уже описывалось выше необходимо открыть пакет и запустить отладчик с помощью ФАЙЛ/ОТЛАДКА или кнопкой  на панели управления, при этом на экране появится окно отладчика. Если в БЦП уже загружено какое-то приложение, перед загрузкой нового необходимо командой ФАЙЛ→ОСТАНОВИТЬ ПРИЛОЖЕНИЕ или кнопкой  остановить его. Загрузка готового приложения осуществляется командой ФАЙЛ→ЗАГРУЗИТЬ или . Внешний вид загруженной программы в отладчике представлен на рисунке 50.

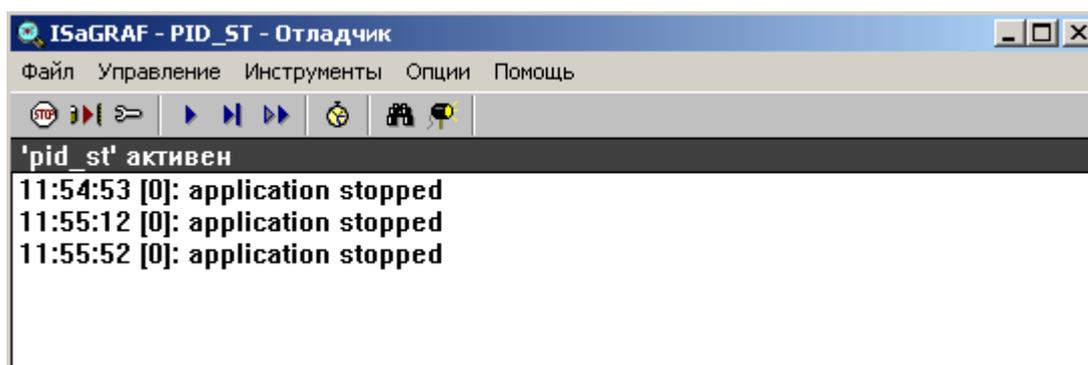


Рисунок 50 – Внешний вид загруженной программы PID_ST в отладчике

Первоначально проверяется соответствие установленного в программе номера модуля ввода/вывода аналоговых сигналов (значение adr) номеру модуля заданного преподавателем.

Устанавливается значение задания в диапазоне от 20 до 60, устанавливаются параметры регулятора. Диапазон изменения коэффициента пропорциональности: 0.45 – 0.55, диапазон постоянной времени интегрирования: 0.045 – 0.0555, постоянная времени дифференцирования равна нулю. Если значение выхода объекта близко к величине задания, то программа является работоспособной. Если задание не обрабатывается, то ошибка может заключаться в самой программе, либо в неправильной настройке регулятора.

При проверки работоспособности можно обратиться к словарю. Если программа работоспособна, то значения переменных, соответствующих заданию и выхода объекта управления в словаре программы будет близки друг другу. Например, как показано на рисунке 5,1 значение $Xzdn$ равно значению $INPUT$.

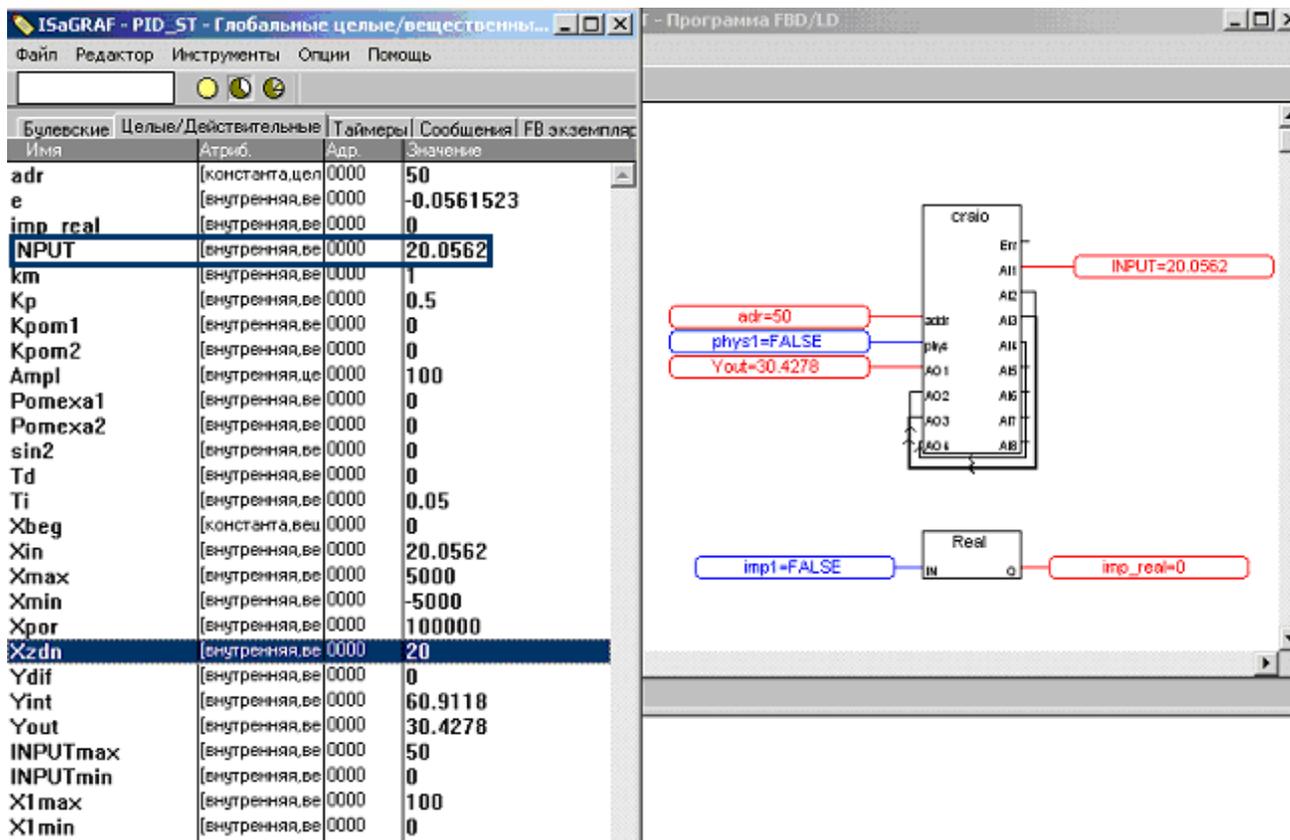


Рисунок 51 – Внешний вид работающей программы регулирования PID_ST

- Второй шаг – проверка настройки и запуск OPC сервера, который находится по адресу C:\Program Files\KPOCC\OPC\crossopc.exe и запускается для привязки OPC – переменных. Посредством OPC осуществляется взаимосвязь технологической программы пользователя контроллера Кросс-500, на базе которого осуществляется регулирование и программного пакета MasterScada, на базе которого осуществляется программа визуализации процессов регулирования и сигнализации. Свидетельством запущенного OPC – сервера является иконка «OPC» в системном трее (рисунок 52).



Рисунок 52 – OPC – сервер в работе

3. Последний этап – загрузка программы визуализации процессов контроля, регулирования и сигнализации, созданной в пакете MasterScada. Для этого как уже описывалось выше необходимо открыть программный пакет, загрузить свой проект и запустить его, нажав на соответствующую кнопочку на панели управления . Появится окно, изображенное на рисунке 53. В поле «Имя» введите «sa» и нажмите ОК.

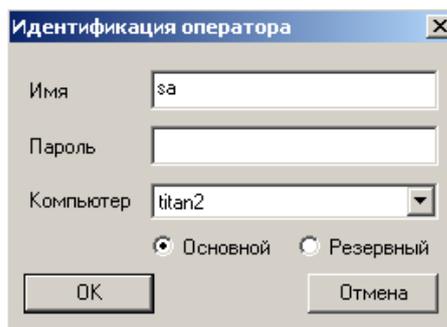


Рисунок 53 – Окно идентификации оператора

Изменяя с помощью кнопок мнемосхемы значение уставки, параметров настройки регулятора, параметров сигнализации, а, также изменяя пороги, проверяющие данные на достоверность убедитесь в работоспособности используемой программы визуализации. В данной программе визуализация осуществляется с помощью столбиковых индикаторов, а параметры регулятора отображаются с помощью цифровых значений и изображение мнемосхемы КОНТРОЛЬ имеет вид, представленный на рисунке 54.

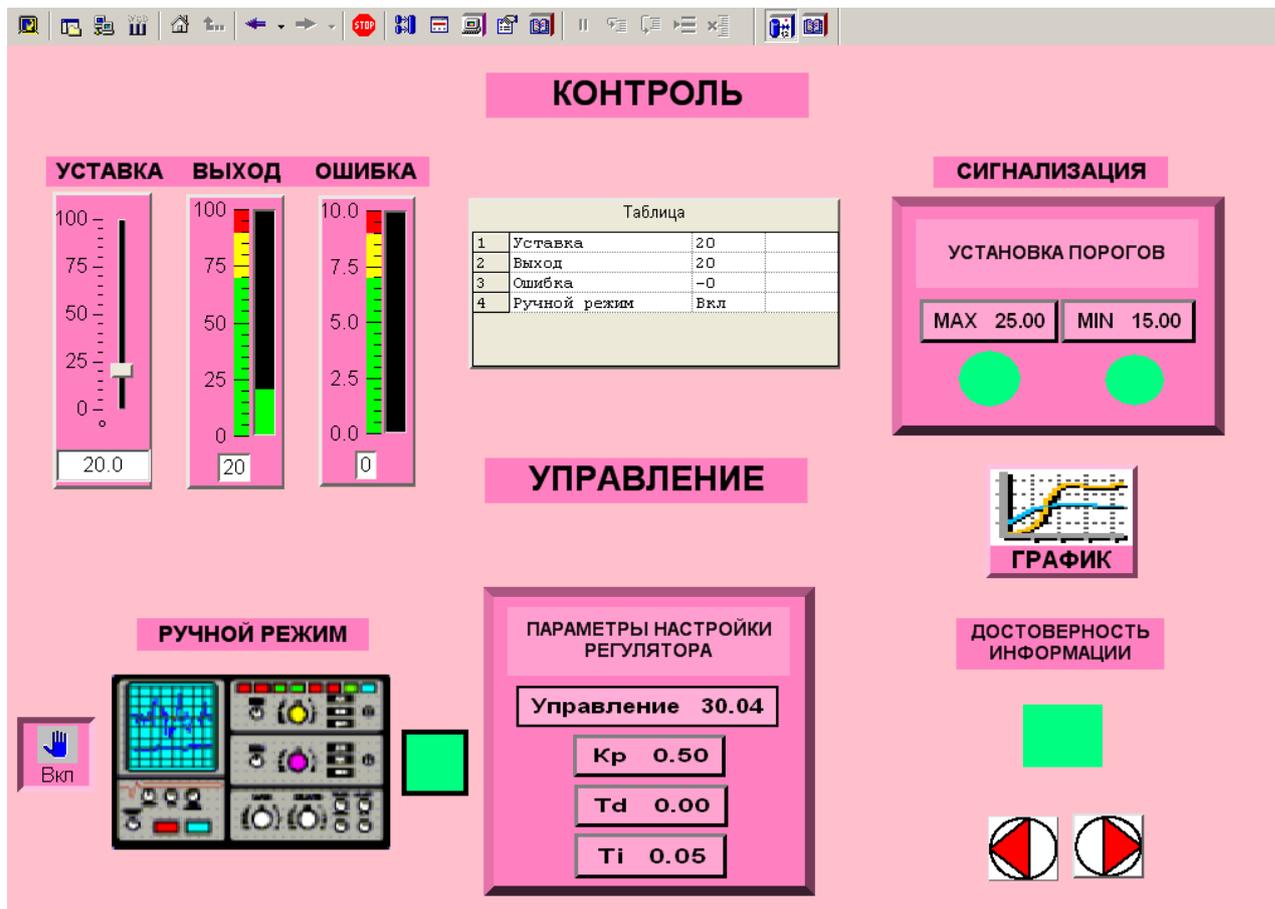


Рисунок 54 – Внешний вид работающей программы визуализации
REG

Проверка переходов от одной мнемосхемы к другой осуществлялась следующим образом: нажимая на кнопку ДАЛЕЕ, программа переходит на следующую мнемосхему ОБЪЕКТ УПРАВЛЕНИЯ, которая имеет вид, представленный на рисунке 55.



Рисунок 55 – Мнемосхема ОБЪЕКТ УПРАВЛЕНИЯ

Нажимая на кнопку НАЗАД, программа переходит на предыдущую мнемосхему, которая является главной, вид которой представлен на рисунке 56.

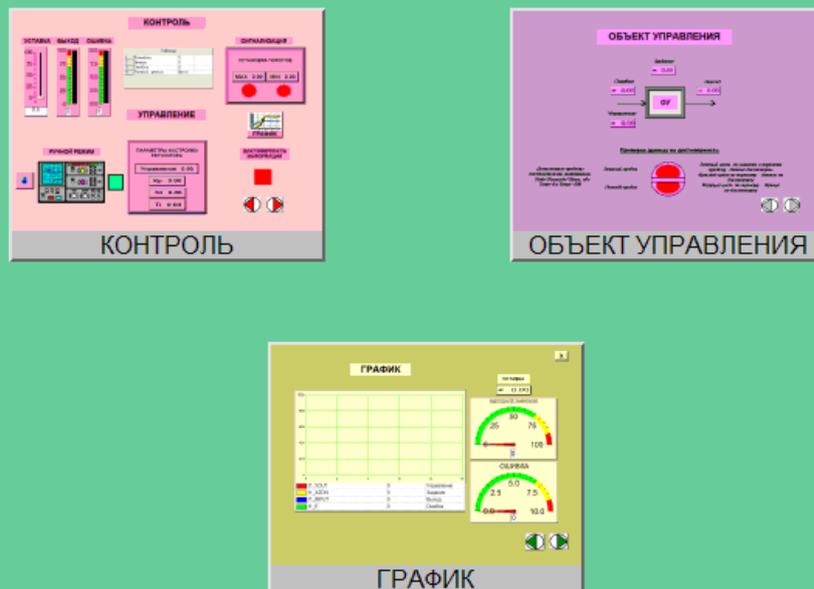
ПИД - РЕГУЛЯТОР

Рисунок 56 – Главная мнемосхема

Если информация не достоверна или не соответствует порогам сигнализации, установленным оператором, то мнемосхема выглядит согласно рисунку 57. Индикаторы, отвечающие за сигнализацию и недостоверность данных, осуществляют цветовую мигающую сигнализацию.

Исходя из выше описанного, можно сделать вывод, что все программы проверены на работоспособность как отдельно, так и в комплексе. Можно сделать вывод, созданные программы полностью верны и работоспособны.

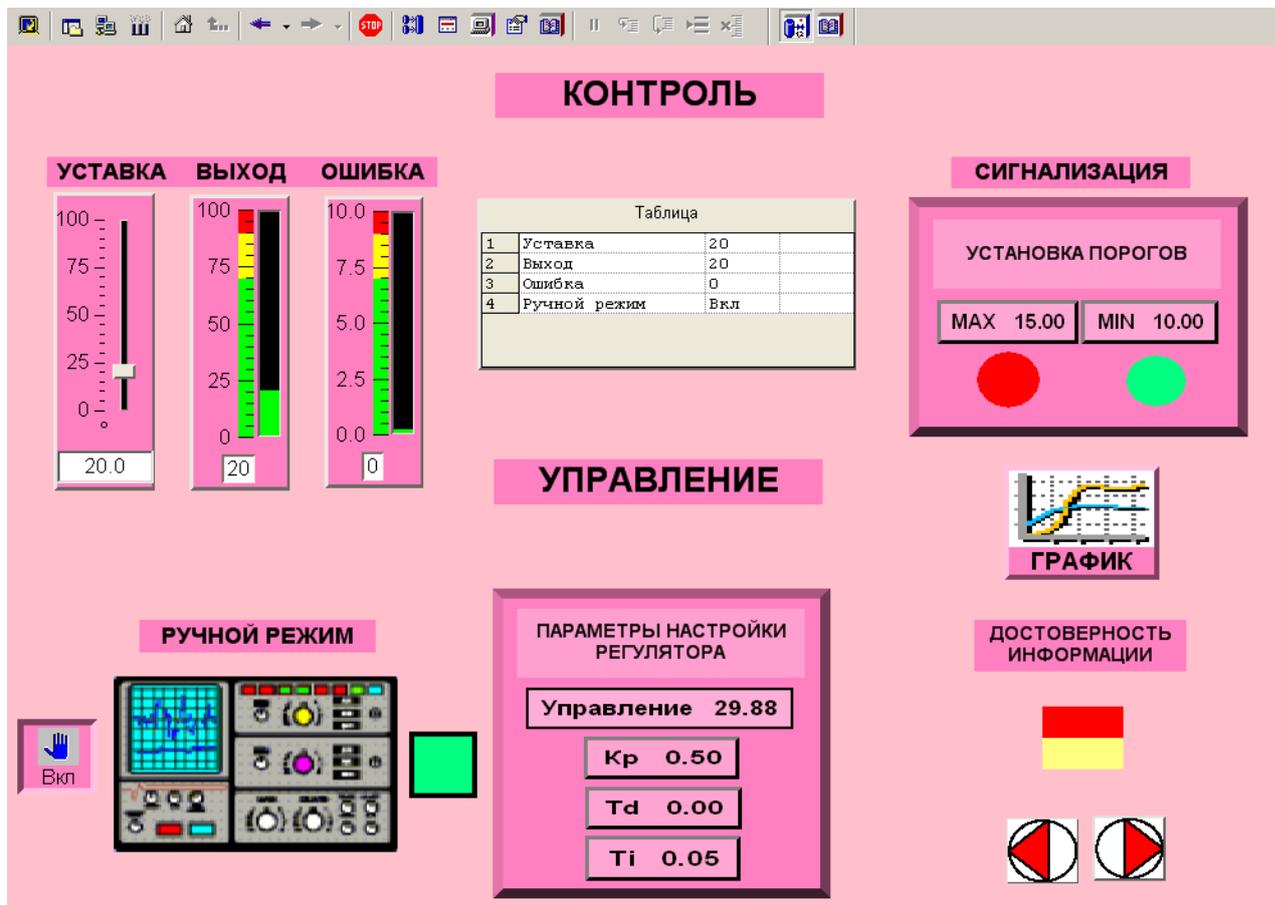


Рисунок 57– Внешний вид аварийной работы программы визуализации REG

Выполнение данного курсового проекта направлено на изучение промышленных микропроцессорных контроллеров КРОСС-500 и ТРАССА-500, типовых структур систем автоматизации, построенных на базе данных контроллеров. Изучение программного пакета ISaGRAF, ознакомление с методикой программирования на языках программирования промышленных контроллеров, определенных международным стандартом IEC 1131-3, а также реализацию собственной программы контроля, регулирования и сигнализации на языках структурированного текста и функциональных блоков. Данная работа также позволяет освоить OPC технологию построения АСУТП, изучить отечественный SCADA-пакет MasterScada, характеризующийся

развитыми функциональными возможностями и удобствами реализации функции динамизации.

Особенностью данного курсового проекта является то, что при его выполнении студенты получают не только теоретические знания, но и практические навыки создания программного обеспечения АСУ ТП и проверки его работоспособности на лабораторном стенде в режиме реального времени.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Контроллер для распределенных открытых систем КРОСС-500. Руководство по эксплуатации ЯЛБИ.421457.018 РЭ.
2. Контроллер для распределенных открытых систем ТРАССА-500. Руководство по эксплуатации ЯЛБИ.421458.018 РЭ.
3. Контроллеры КРОСС - 500, ТРАССА - 500, Р -130ISA. Руководство по эксплуатации. Часть 1. Система технологического программирования ISaGRAF. ЯЛБИ.420146.001 РЭ.
4. Документация на пакет MasterScada. – М.: InSAT, 2007. –382 с.
5. Громаков Е.И. Проектирование автоматизированных систем: учебно-методическое пособие.- Томск: Изд-во Томского политехнического университета, 2010.–168 с.
6. Технические средства систем автоматики и управления: учебное пособие / В.Н. Скороспешкин, М.В. Скороспешкин; Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2013. – 219 с.
7. Освоение КРОСС-500. Руководство по сборке и программированию контроллера КРОСС-500 - ОАО «ЗЭиМ», г. Чебоксары, 2006.-28 с
8. SCADA <http://www.masterscada.ru>
9. Казанцев А.А. Isagraf - <http://prodcs.ru>

10. Микропроцессорные контроллеры производства ОАО «АБС Автоматизация» <http://www.absholdings.biz/absnew/news/im/04-05-08/kabanenko.pdf>
11. ГОСТ 34.602–89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы / Комплекс стандартов и руководящих документов на автоматизированные системы. Информационная технология. – М.: 1991. С. 3–15.
12. ГОСТ 34.601–90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы, стадии создания / Комплекс стандартов и руководящих документов на автоматизированные системы. Информационная технология. – М., 1991.– С. 45.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Государственное образовательное учреждение высшего профессионального образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Отделение Автоматизации и Робототехники
Направление 15.03.04 «Автоматизация технологических процессов и производств»

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ КОНТРОЛЯ, РЕГУЛИРОВАНИЯ И
ВИЗУАЛИЗАЦИИ АСУ ТП, РЕАЛИЗОВАННОЙ НА БАЗЕ КОНТРОЛЛЕРА КРОСС-
500

**Курсовой проект по дисциплине
«Автоматизированные информационно-управляющие системы»**

КП ФЮРА. 424280. 001 ПЗ.01

Студент группы _____
(Подпись, дата) (И.О. Фамилия)

Руководитель работы _____
(Должность) (Подпись, дата) (И.О. Фамилия)

Томск 20** г.

Приложение Б

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Государственное образовательное учреждение высшего профессионального образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Отделение Автоматизации и
Робототехники

УТВЕРЖДАЮ:
Руководитель
ОАР _____ (Ф.И.О.)
(подпись, дата)

ЗАДАНИЕ

на выполнение курсового проекта

Студенту _____

1 Тема курсового проекта

_____ (утверждена _____ от _____ № _____)

2 Срок сдачи студентом готовой работы _____

3 Исходные данные к работе _____

4 Содержание текстового документа (перечень подлежащих
разработке вопросов)

5 Перечень графического материала (с точным указанием обязательных чертежей)

5.1

5.2

5.3

5.4

5.5

5.6

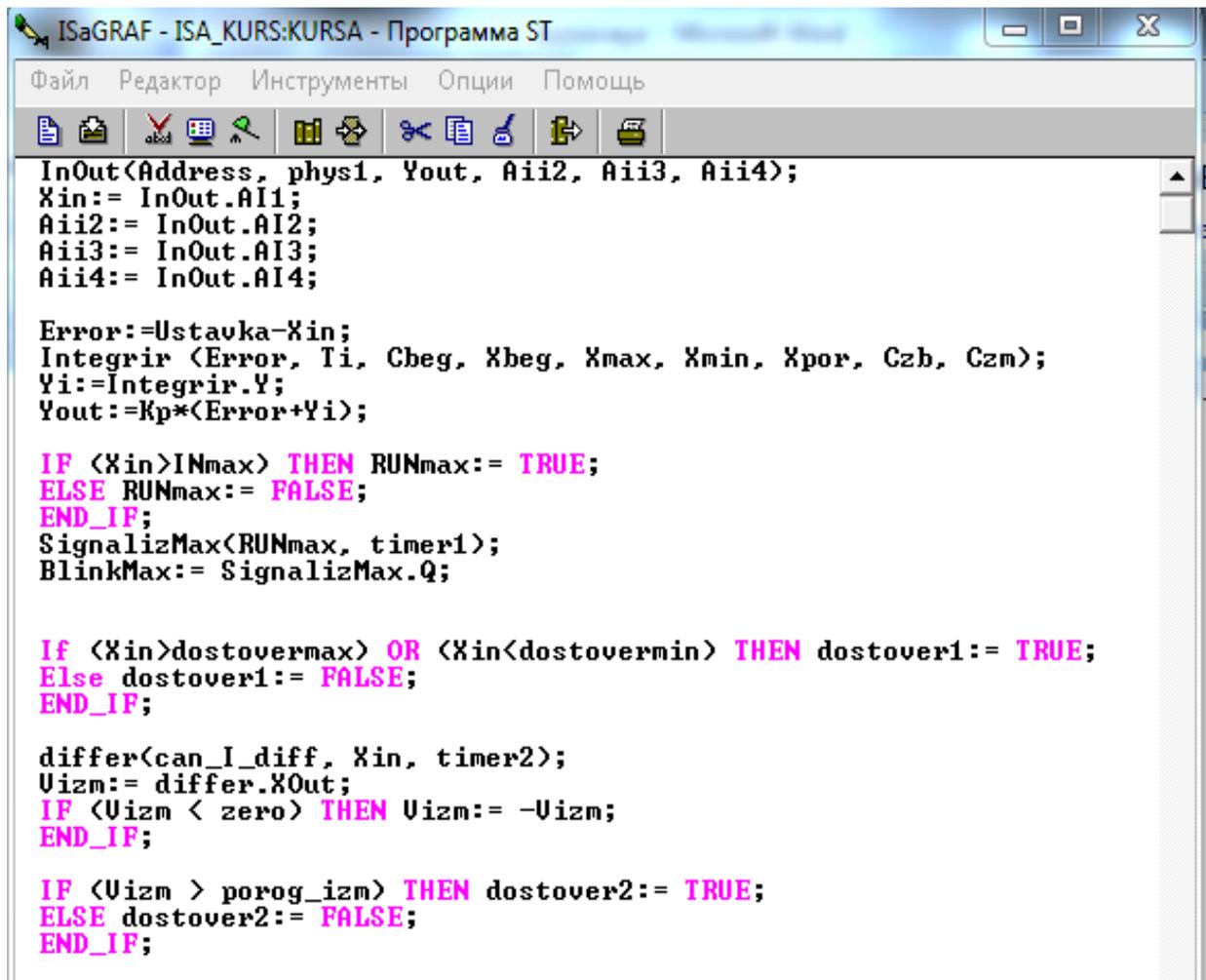
6 Дата выдачи задания на выполнение курсового проекта

Руководитель _____ (И.О.Ф.)
(подпись, дата)

Задание принял к исполнению

_____ (И.О.Ф.)
(подпись, дата)

Программа на языке ST



The screenshot shows a window titled "ISaGRAF - ISA_KURSA:KURSA - Программа ST". The menu bar includes "Файл", "Редактор", "Инструменты", "Опции", and "Помощь". The toolbar contains icons for file operations, editing, and execution. The main text area contains the following ST code:

```
InOut(Address, phys1, Yout, Aii2, Aii3, Aii4);
Xin:= InOut.AI1;
Aii2:= InOut.AI2;
Aii3:= InOut.AI3;
Aii4:= InOut.AI4;

Error:=Ustavka-Xin;
Integrir (Error, Ti, Cbeg, Xbeg, Xmax, Xmin, Xpor, Czb, Czm);
Yi:=Integrir.Y;
Yout:=Kp*(Error+Yi);

IF (Xin>INmax) THEN RUNmax:= TRUE;
ELSE RUNmax:= FALSE;
END_IF;
SignalizMax(RUNmax, timer1);
BlinkMax:= SignalizMax.Q;

If (Xin>dostovermax) OR (Xin<dostovermin) THEN dostover1:= TRUE;
Else dostover1:= FALSE;
END_IF;

differ(can_I_diff, Xin, timer2);
Uizm:= differ.XOut;
IF (Uizm < zero) THEN Uizm:= -Uizm;
END_IF;

IF (Uizm > porog_izm) THEN dostover2:= TRUE;
ELSE dostover2:= FALSE;
END_IF;
```

Типы переменных

Булевские	Целые/Действительные	Таймеры	Сообщения	FB экземпляры	Макроопределения
Имя	Атриб.	Адр.	Комментарий		
BlinkMax	[внутренняя]	0000	гориг-не гориг		
phys1	[константа]	0000	сигнал в процентах от диапазона		
Cbeg	[константа]	0000	команда установки начальных условий		
Czb	[константа]	0000	Сигнал запрета в направлении "Больше"		
Czm	[константа]	0000	Сигнал запрета в направлении "Меньше"		
RUNmax	[внутренняя]	0000	превышено или нет макс значение		
dostover2	[внутренняя]	0000	достоверность по скорости изменения величины		
can_l_diff	[константа]	0000	разрешение дифференцирования		
dostover1	[внутренняя]	0000	Достоверность по порогу		

Булевские	Целые/Действительные	Таймеры	Сообщения	FB экземпляры	Макроопределения
Имя	Атриб.	Адр.	Комментарий		
zero	[константа,вещественная]	0000	ноль		
porog_izm	[константа,вещественная]	0000	порог изменения		
Xin	[внутренняя,вещественная]	0000	Вход от АВК		
Yout	[внутренняя,вещественная]	0000	Выход возд на АВК		
Address	[константа,целая]	0000	Адрес контроллера		
Error	[внутренняя,вещественная]	0000	Ошибка регулирования		
Ustavka	[внутренняя,вещественная]	0000	Значение уставки		
Ti	[внутренняя,вещественная]	0000	постоянная интегрирования		
Xbeg	[константа,вещественная]	0000	значение начальных условий		
Xmax	[константа,вещественная]	0000	Уровень ограничения по максимуму		
Xmin	[константа,вещественная]	0000	Уровень ограничения по минимуму		
Xpor	[константа,вещественная]	0000	порог		
Vizm	[внутренняя,вещественная]	0000	Скорость изменения параметра		
Yi	[внутренняя,вещественная]	0000	выход интегратора		
Kp	[внутренняя,вещественная]	0000	коэф пропорциональности		
INmax	[внутренняя,вещественная]	0000	Максимальное значение		
dostovermax	[внутренняя,вещественная]	0000	Макс знач вход сигнала с АВК		
dostovermin	[внутренняя,вещественная]	0000	Мин знач вход сигнала с АВК		
Aii2	[внутренняя,вещественная]	0000	первый параметр		
Aii3	[внутренняя,вещественная]	0000	2й параметр		
Aii4	[внутренняя,вещественная]	0000	3й параметр		

Булевские	Целые/Действительные	Таймеры	Сообщения	FB экземпляры	Макроопределения
Имя	Атриб.	Адр.	Комментарий		
timer2	[внутренняя]	0000			
timer1	[внутренняя]	0000			

Булевские	Целые/Действительные	Таймеры	Сообщения	FB экземпляры	Макроопределения
Имя	Значение			Комментарий	
SignalizMax	blink			сигнализация при выходе за пределы	
differ	derivate			производная	
dostovern1	blink			сигнал за порогом	
dostovern2	blink			высокая скорость изменения сигнала	
InOut	craio			Ввод-вывод анал сигн с модуля УСО КРОСС	
Integrir	integr			Блок интегрирования	

Учебное издание

СКОРОСПЕШКИН Владимир Николаевич
СКОРОСПЕШКИН Максим Владимирович

АВТОМАТИЗИРОВАННЫЕ ИНФОРМАЦИОННО- УПРАВЛЯЮЩИЕ СИСТЕМЫ

Учебно-методическое пособие

Подписано к печати xx.xx.xxxx. Формат 60x84/16. Бумага «Снегурочка».

Печать XEROX. Усл.печ.л. __, __. Уч.-изд.л. __, __.

Заказ . Тираж __ экз.



Национальный исследовательский Томский политехнический университет
Система менеджмента качества
Издательства Томского политехнического университета сертифицирована
NATIONAL QUALITY ASSURANCE по стандарту BS EN ISO 9001:2008



ИЗДАТЕЛЬСТВО  **ТПУ**. 634050, г. Томск, пр. Ленина, 30
Тел./факс: 8(3822)56-35-35, www.tpu.ru