



Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа - Инженерная школа информационных технологий и робототехники
Направление подготовки - 15.03.04 Автоматизация технологических процессов и производств
ООП Автоматизация технологических процессов и производств в нефтегазовой отрасли
Отделение школы (НОЦ) - Отделение автоматизации и робототехники

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Тема работы
<i>Система автоматического регулирования уровня на базе контроллера КРОСС</i>

УДК 621.313.13

Обучающийся

Группа	ФИО	Подпись	Дата
158Т91	Ли Яньчэн		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Скороспешкин М.В.	К.Т.Н.		

Консультант

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Спиридонова А.С.	К.Т.Н.		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШБИП	Якимова Т.Б.	К.Э.Н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ООД ШБИП	Сечин А.И.	Д.Т.Н.		

Нормоконтроль

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ОАР ИШИТР	Кучман А.В			

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП/ОПОП, должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Цавнин А.В.	К.Т.Н.		

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ООП

Код компетенции	Наименование компетенции
Универсальные компетенции	
УК(У)-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
УК(У)-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений
УК(У)-3	Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде
УК(У)-4	Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(-ых) языке(-ах)
УК(У)-6	Способен воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах
УК(У)-7	Способен поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности
УК(У)-8	Способен создавать и поддерживать безопасные условия жизнедеятельности, в том числе при возникновении чрезвычайных ситуаций Эхщпгнюждщщ№увааей
УК(У)-9	Способен проявлять предприимчивость в практической деятельности, в т.ч. в рамках разработки коммерчески перспективного продукта на основе научно-технической идеи.
Общепрофессиональные компетенции	
ОПК(У)-1	Способен использовать основные закономерности, действующие в процессе изготовления продукции требуемого качества, заданного количества при наименьших затратах общественного труда
ОПК(У)-2	№№№№й№й
ОПК(У)-3	Способен использовать современные информационные технологии, технику, прикладные программные средства при решении задач профессиональной деятельности
ОПК(У)-4	Способен участвовать в разработке обобщенных вариантов решения проблем, связанных с автоматизацией производств, выборе на основе анализа вариантов оптимального прогнозирования последствий решения
ОПК(У)-5	Способен участвовать в разработке технической документации, связанной с профессиональной деятельностью
Профессиональные компетенции	
ПК(У)-1	Способен собирать и анализировать исходные информационные данные для проектирования технологических процессов изготовления продукции, средств и систем автоматизации, контроля, технологического оснащения, диагностики, испытаний, управления процессами, жизненным циклом продукции и ее качеством; участвовать в работах по расчету и проектированию процессов изготовления продукции и указанных средств и систем с использованием современных информационных технологий, методов и средств проектирования
ПК(У)-2	Способен выбирать основные и вспомогательные материалы для изготовления изделий, способы реализации основных технологических процессов, аналитические и численные методы при разработке их математических моделей, методы стандартных испытаний по определению

Код компетенции	Наименование компетенции
	физико-механических свойств и технологических показателей материалов и готовых изделий, стандартные методы их проектирования, прогрессивные методы эксплуатации изделий
ПК(У)-3	готов применять способы рационального использования сырьевых, энергетических и других видов ресурсов, современные методы разработки малоотходных, энергосберегающих и экологически чистых технологий, средства автоматизации технологических процессов и производств
ПК(У)-4	Способен участвовать в постановке целей проекта (программы), его задач при заданных критериях, целевых функциях, ограничениях, разработке структуры его взаимосвязей, определении приоритетов решения задач с учетом правовых и нравственных аспектов профессиональной деятельности, в разработке проектов изделий с учетом технологических, конструкторских, эксплуатационных, эстетических, экономических и управленческих параметров, в разработке проектов модернизации действующих производств, создании новых, в разработке средств и систем автоматизации, контроля, диагностики, испытаний, управления процессами, жизненным циклом продукции и ее качеством в соответствии с техническими заданиями и использованием стандартных средств автоматизации расчетов и проектирования
ПК(У)-5	Способен участвовать в разработке (на основе действующих стандартов и другой нормативной документации) проектной и рабочей технической документации в области автоматизации технологических процессов и производств, их эксплуатационному обслуживанию, управлению жизненным циклом продукции и ее качеством, в мероприятиях по контролю соответствия разрабатываемых проектов и технической документации действующим стандартам, техническим условиям и другим нормативным документам
ПК(У)-6	Способен проводить диагностику состояния и динамики производственных объектов производств с использованием необходимых методов и средств анализа
ПК(У)-7	Способен участвовать в разработке проектов по автоматизации производственных и технологических процессов, технических средств и систем автоматизации, контроля, диагностики, испытаний, управления процессами, жизненным циклом продукции и ее качеством, в практическом освоении и совершенствовании данных процессов, средств и систем
ПК(У)-8	Способен выполнять работы по автоматизации технологических процессов и производств, их обеспечению средствами автоматизации и управления, готовностью использовать современные методы и средства автоматизации, контроля, диагностики, испытаний и управления процессами, жизненным циклом продукции и ее качеством
ПК(У)-9	Способен определять номенклатуру параметров продукции и технологических процессов ее изготовления, подлежащих контролю и измерению, устанавливать оптимальные нормы точности продукции, измерений и достоверности контроля, разрабатывать локальные поверочные схемы и выполнять проверку и отладку систем и средств автоматизации технологических процессов, контроля, диагностики, испытаний, управления процессами, жизненным циклом продукции и ее качеством, а также их ремонт и выбор; осваивать средства обеспечения автоматизации и управления
ПК(У)-10	Способен проводить оценку уровня брака продукции, анализировать

Код компетенции	Наименование компетенции
	причины его появления, разрабатывать мероприятия по его предупреждению и устранению, по совершенствованию продукции, технологических процессов, средств автоматизации и управления процессами, жизненным циклом продукции и ее качеством, систем экологического менеджмента предприятия, по сертификации продукции, процессов, средств автоматизации и управления
ПК(У)-11	Юждзщц?щшгшгннекекк:к:бёсмчлжапозеее,,;ук”кекк”:к +
ПК(У)-18	Способен аккумулировать научно-техническую информацию, отечественный и зарубежный опыт в области автоматизации технологических процессов и производств, автоматизированного управления жизненным циклом продукции, компьютерных систем управления ее качеством,
ПК(У)-19	Способен участвовать в работах по моделированию продукции, технологических процессов, производств, средств и систем автоматизации, контроля, диагностики, испытаний и управления процессами, жизненным циклом продукции и ее качеством с использованием современных средств автоматизированного проектирования, по разработке алгоритмического и программного обеспечения средств и систем автоматизации и управления процессами
ПК(У)-20	Способен проводить эксперименты по заданным методикам с обработкой и анализом их результатов, составлять описания выполненных исследований и подготавливать данные для разработки научных обзоров и публикаций
ПК(У)-21	Способен составлять научные отчеты по выполненному заданию и участвовать во внедрении результатов исследований и разработок в области автоматизации технологических процессов и производств, автоматизированного управления жизненным циклом продукции и ее качеством
ПК(У)-22	Способен участвовать: в разработке программ учебных дисциплин и курсов на основе изучения научной, технической и научно-методической литературы, а также собственных результатов исследований; в постановке и модернизации отдельных лабораторных работ и практикумов по дисциплинам профилей направления; способностью проводить отдельные виды аудиторных учебных занятий (лабораторные и практические), применять новые образовательные технологии, включая системы компьютерного и дистанционного обучения



Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных технологий и робототехники
Направление подготовки 15.03.04 Автоматизация технологических процессов и производств
Отделение школы (НОЦ) Отделение автоматизации и робототехники

УТВЕРЖДАЮ:
Руководитель ООП
_____ Цавнин А.В.
(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ на выполнение выпускной квалификационной работы

Обучающийся:

Группа	ФИО
158Т91	Ли Яньчэн

Тема работы:

<i>Система автоматического регулирования уровня на базе контроллера КРОСС</i>	
Утверждена приказом директора (дата, номер)	№ 33-42/с 02.02.2023

Срок сдачи обучающимся выполненной работы:	09.06.2023
--	------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

<p>Исходные данные к работе</p> <p><i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i></p>	<p>Объектом исследования является система автоматического регулирования уровня на базе контроллера КРОСС.</p>
<p>Перечень разделов пояснительной записки подлежащих исследованию, проектированию и разработке</p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<p>Программа управления реализуется в программном пакете ISaGRAF на языке FBD. Связь контроллера и SCADA-системы Master SCADA осуществляется посредством OPC-сервера. Моделирование работы системы осуществляется в программном пакете MatLab. Предусмотрены варианты автоматического или ручного управления физическим объектом.</p>

<p>Перечень графического материала (с точным указанием обязательных чертежей)</p>	<p>Структурная схема контроллера Структурная схема лабораторного комплекса Схема внешних соединений Алгоритм работы управляющей программы Программа управления гидравлическим объектом на языке FBD Основная мнемосхема проекта Окно «Контроль и управление» Окно настроек ПИД-регулятора</p>
--	---

Консультанты по разделам выпускной квалификационной работы

(с указанием разделов)

Раздел	Консультант
<p>Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</p>	<p>Якимова Т.Б., Кандидат экономических наук, Доцент ОСГН ШБИП</p>
<p>Социальная ответственность</p>	<p>Сечин А.И., Доктор технических наук, Профессор ООД ШБИП</p>

Названия разделов, которые должны быть написаны на иностранном языке:

--

<p>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</p>	<p>02.02.2023 г.</p>
---	----------------------

Задание выдал руководитель / консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
<p>Доцент ОАР ИШИТР</p>	<p>Скороспешкин М.В.</p>	<p>к.т.н. доцент</p>		

Задание принял к исполнению обучающийся:

Группа	ФИО	Подпись	Дата
<p>158Т91</p>	<p>Ли Яньчэн</p>		

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных технологий и робототехники
 Направление подготовки - 15.03.04 Автоматизация технологических процессов и производств
 Уровень образования Бакалавриат
 Отделение школы (НОЦ) Отделение автоматизации и робототехники
 Период выполнения Весенний семестр 2022 /2023 учебного года

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Обучающийся:

Группа	ФИО
158Т91	Ли Яньчэн

Тема работы:

<i>Система автоматического регулирования уровня на базе контроллера КРОСС</i>

Срок сдачи обучающимся выполненной работы:

--

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
27.05.2023 г.	<i>Основная часть ВКР</i>	60
30.05.2023 г.	<i>Раздел «Социальная ответственность»</i>	20
30.05.2023 г.	<i>Раздел «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»</i>	20

СОСТАВИЛ:

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Скороспешкин М.В.	к.т.н.		

Консультант

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Спиридонова А.С.	к.т.н.		

СОГЛАСОВАНО:

Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Цавнин А.В.	к.т.н.		

Обучающийся

Группа	ФИО	Подпись	Дата
158Т91	Ли Яньчэн		

РЕФЕРАТ

Выпускная квалификационная работа 167 страниц, 32 рисунков, 17 таблиц, 16 источников, 2 приложения.

Ключевые слова: программное обеспечение, методическое обеспечение, лабораторный стенд, промышленный микропроцессорный контроллер КРОСС, системы программирования ISaGRAF, язык программирования FBD, функциональные блоки, OPC сервер, ETHERNET, SCADA-система MasterSCADA, мнемосхема.

Объектом разработки является програмно-методическое обеспечение лабораторного стенда системы автоматического регулирования уровня жидкости.

Цель работы – реализация программного и методического обеспечения лабораторного стенда. Программы регулирования разработаны в системе программирования ISaGRAF, программы визуализации разработаны в SCADA-системе MasterSCADA.

Реализовано автоматическое и ручное управление уровнем в емкости лабораторного стенда.

Основные конструктивные, технологические и технико-эксплуатационные характеристики: установка, состоящая из трёх ёмкостей, регулирующего клапана мембранного типа с электроприводом, насоса, одного аналогового и трёх дискретных датчиков измерения уровня, стенд с контроллером и персональный компьютер – операторская станция. В качестве операторской станции в системе используется персональный с установленной SCADA-системой MasterSCADA и системой программирования ISaGRAF.

Степень внедрения: Система реализована и проверена ее работоспособность.

Область применения: учебный процесс отделения автоматизации и робототехники ИШИТР.

Содержание

ВВЕДЕНИЕ	10
1 Назначение, состав и технические характеристики контроллера КРОСС	12
1.1 Назначение, состав и архитектура контроллера	12
1.2 Технические характеристики контроллера КРОСС	13
1.2.1 Условия эксплуатации	13
1.2.2 Общие функциональные характеристики контроллера КРОСС ...	14
1.2.3 Параметры модулей УСО и входы-выходы контроллера	16
1.2.4 Электрическое питание контроллера КРОСС	19
1.2.6 Показатели надежности контроллера КРОСС	20
1.3 Устройство контроллера КРОСС	20
1.3.1 Описание аппаратуры контроллера КРОСС	20
1.3.2 Описание конструкции контроллера КРОСС	22
2 СТРУКТУРНАЯ СХЕМА ЛАБОРАТОРНОГО КОМПЛЕКСА, СХЕМА ВНЕШНИХ СОЕДИНЕНИЙ	23
2.1 Структурная схема комплекса	23
2.2 Конфигурация контроллера КРОСС. Схема внешних соединений	26
3 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЛАБОРАТОРНОГО КОМПЛЕКСА	29
3.1 Основные сведения об операционной системе OS-9	30
3.2 Краткие сведения о системе ISaGRAF	31
3.3 Конфигуратор	32
3.4 OPC сервер	34
3.5 Языки программирования	37
3.6 Разработка программы для САР уровня на основе языка программирования FBD	39
3.6.1 Программа управления гидравлическим объектом	39
3.6.2 Функциональные блоки используемые при создании программы	41
3.6.3 Переменные использованные в программе	46
3.6.4 Алгоритм работы управляющей программы	47
3.6.5 Организация ввода/ вывода аналоговых и дискретных сигналов .	49
3.6.6 Программа регулирования уровня в емкости E2 входящая в состав программы управления гидравлическим объектом	50
3.6.7 Программа управления насосом НЗ входящая в состав программы управления гидравлическим объектом	51
3.6.8 Программа световой сигнализации входящая в состав программы управления гидравлическим объектом	52
3.7 Разработка программы для САР уровня на основе языка программирования ST	53
3.7.1 Программа управления гидравлическим объектом	54
3.7.2 Организация ввода/ вывода аналоговых и дискретных сигналов .	56
3.7.2 Программа регулирования уровня в емкости E2 входящая в состав программы управления гидравлическим объектом	57
3.7.3 Программа управления насосом НЗ входящая в состав программы управления гидравлическим объектом	58

3.7.4 Программа организации световой сигнализации входящая в состав программы управления гидравлическим объектом	59
3.8 Расчёт параметров настройки регулятора	60
4 ПРИМЕНЕНИЕ SCADA-ПАКЕТА MasterSCADA ДЛЯ ВИЗУАЛИЗАЦИИ РАБОТЫ ГИДРАВЛИЧЕСКОГО ОБЪЕКТА	63
4.1 Назначение SCADA- систем	63
4.2 SCADA – пакет MasterSCADA 2.2	63
4.2.1 Основные преимущества пакета	63
4.2.2 Пользовательский интерфейс MasterScada	65
4.2.3 Библиотеки	66
4.2.4 Мнемосхемы	66
4.2.5 Тренды	67
4.3 Программа визуализации процесса управления гидравлическим объектом в пакете MasterSCADA	67
5 МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ	71
5.1 Методические указания по выполнению лабораторной работы «Создание программ управления работой установки «Гидравлический объект» в системе ISaGRAF на основе языков программирования FBD и ST»	71
5.2 Методические указания по выполнению лабораторной работы «Создание программы визуализации процесса управления установкой «Гидравлический объект» в пакете MasterSCADA»	73
6 КОНСТРУКТОРСКАЯ ЧАСТЬ	74
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение ...	76
1 планирование научно-исследовательских работ	77
2 Бюджет научно-технического исследования	80
3 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	84
4 Вывод	87
СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ	89
1 Правовые и организационные вопросы обеспечения безопасности	92
2 Производственная безопасность	93
2.1 Анализ опасных и вредных производственных факторов	94
2.2 Анализ показателей шума и вибрации	95
2.3 Анализ показателей микроклимата	96
2.4 Анализ освещенности рабочей зоны	96
2.5 Анализ электробезопасности	97
2.6 Анализ пожарной безопасности	97
3 Экологическая безопасность	98
4 Безопасность в чрезвычайных ситуациях	99
5 Заключение по разделу социальная ответственность	100
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	101
ПРИЛОЖЕНИЕ А. МЕТОДИКА ISAGRAG	
Приложение Б. Методика по MasterSCADA	

ВВЕДЕНИЕ

Важной задачей в современном технологическом развитии общества является исследование и разработка различных систем автоматического управления для контроля оборудования и технических процессов, используя все известные методы программирования и симуляционной реализации.

Опыт показывает, что внедрение систем регулирования позволяет повысить эффективность управления технологическим процессом.

В связи с текущим экономическим и технологическим развитием общества и автоматизации, системы автоматического управления и соответствующее оборудование автоматического управления должны быть усовершенствованы, в дополнение к развитию микропроцессорной техники.

Реализация этих задач невозможна без подготовки высококвалифицированных специалистов в области регулирования технологических процессов, способных разрабатывать, внедрять и применять в условиях реального производства такие устройства и системы.

В экспериментах используется все - от написания базовых программ до имитационного моделирования - для исследования систем автоматического управления технологическими процессами для промышленных контроллеров, при этом проверяется точность автоматического управления на промышленном оборудовании с поддержкой стандарта IEC 1131-3, и завершается работа по обучению персонала. Вместе с тем появляется возможность освоения программного обеспечения верхнего уровня структуры АСУ ТП, т.е. изучение различных SCADA-пакетов и последующей отладкой на данных лабораторных стендах.

Данный проект, выполняемый в Томском политехническом университете в области автоматизации процессов и производств, посвящен разработке программного и методического обеспечения для эксплуатации систем автоматического управления, с целью приобретения практических навыков программирования системы ISaGRAF на базе промышленного микропроцессорного контроллера КРОСС, а также навыков моделирования,

создания моделей технологического процесса, построения кривых переходных процессов.

Цель - приобретение навыков работы в программном комплексе Master SCADA и ISaGRAF. Данная работа предназначена для студентов Томского политехнического университета обучающихся по направлению автоматизации процессов и производств.

1 Назначение, состав и технические характеристики контроллера КРОСС

1.1 Назначение, состав и архитектура контроллера

Данный контроллер выпускается в основном для промышленного производства и предназначен для использования в условиях современного развития высокоиндустриального общества, где в различных отраслях промышленности используются автоматизированные системы управления технологическими процессами (АСУ ТП), например - нефтепереработка, энергетические системы, переработка минерального сырья, аэрокосмическая промышленность и т.д.

Контроллеры представляют собой небольшие средства автоматического управления, в основном используемые для управления менее сложными объектами.

В рамках реализации автоматизированных систем управления технологическими процессами контроллер в основном разрабатывается для управления опасными ситуациями, такими как взрывы и пожары, включая нефтепереработку, системы выработки электроэнергии, переработку минерального сырья, как определено в ПБ 09-170-97. Контроллер является изделием автоматического управления в соответствии с ГОСТ 18311-80.

Конструктивный состав этого контроллера и соответствующие данные приведены в таблице 1.

Определенный контроллер состоит из устройств, причем их количество и тип определяются заказом клиента.

Модули ввода / вывода (модули УСО) выпускаются группами до 8 штуки и подключаются к центральному блоку ЦБ1 через модуль ИСК1 посредством разъема SPI, каждый модуль имеет два порта интерфейса SPI.

Элементы контроллера можно получить отдельно.

Таблица 1 - Состав контроллера

Наименование устройства	Кол. шт.	Ограничени я	Примечание
Блок центральный ЦБ1 в составе: Модуль процессора SM2-CPU-1,5	1 1		
Панель SM2-CPU-FP Субмодуль ETHERNET SM2-ETH	0-1 0-1	Сумма равна 1	Если в заказе нет субмодуля Интерфейс Ethernet
Базовый монтажный блок SMART2- BASE Панель-заглушка SMDUMMY-FP Модуль ИСК1 Модуль питания KP-DC24V1	1 0-1 1-2 0-1		
Блок питания LOK4601- 2R/P-ONE/	0-4		Если задано напряжение питания 220 В
Модуль питания DC-24/5	0-4		Устанавливаются не более одного на соединитель SPI
Модули AI1-8, AIO1-0/4, AIO1-8/4, AIO1-8/0, TC1-7, TR1-8, DI1-16, DIO1-8/8, DO1-16	0-32		
Блок T1-AI Соединение гибкое C1-AI	0-32	Сумма не более 32	Количество равно количеству модулей AI1 -8
Блок T1-AIO Соединение гибкое C1-AIO	0-32		Количество равно сумме модулей AIO1-0/4, AIO1-8/4, AIO1-8/0
Блок T1-TC Соединение гибкое C1- TC	0-32		Количество равно количеству модулей TC 1-7
Блок T1-TR Соединение гибкое C1- TR	0-32		Количество равно количеству модулей TR1-8
Блок T1-D Соединение гибкое C1-D	0-32		Количество равно сумме модулей DI1-16, DIO1-8/8, DO1-16
Комплект запасных частей согласно ведомости ЯЛБИ.421457.018 ЗИ	1 компл.		
Комплект эксплуатационных документов согласно ведомости ЯЛБИ.421457.018ВЭ	1 компл.		
Ведомость эксплуатационных документов	1 экз.		

1.2 Технические характеристики контроллера КРОСС

1.2.1 Условия эксплуатации

Контроллер соответствует климатическому допустимому диапазону УХЛ 4.2 по ГОСТ 15150-69.

Эксплуатационные характеристики:

- диапазон рабочих температур от плюс 5°С до плюс 50°С;
- влажность до 95 процентов без конденсата;
- не требует принудительной вентиляции в диапазоне рабочих температур;
- гарантия - 18 месяцев со дня ввода в эксплуатацию, сопровождение-10 лет.

1.2.2 Общие функциональные характеристики контроллера КРОСС

В целом, контроллер может использоваться для выполнения следующих функций:

- измерение сигналов от датчиков и фильтрация входных сигналов в реальном времени, а также оптимизация линейаризации отфильтрованных выходных сигналов, преобразование различных полученных сигналов в цифровые сигналы для;
- использовать полученные аналоговые сигналы для получения дискретных сигналов;
- запуск, автоматическое управление и линейная настройка аналоговых и вычислительных программ;
- фильтрация и генерация управляющих сигналов и вывод цифровых сигналов в виде аналоговых и дискретных сигналов;
- хранение записей экспериментов и программ моделирования и хранение технических переменных в статическом энергонезависимом ОЗУ;
- обмен информацией между программным обеспечением;
- для обмена информацией с другими контроллерами или компьютерами через Ethernet;
- последовательный интерфейс RS 232 для связи с компьютером;
- интерфейсное подключение контроллера к различным SCADA системам;

- воспроизведение информации на экране компьютера, соединенного с контроллером через интерфейс RS232 или через сеть Ethernet;
- возможность осуществлять самодиагностику модуля и контроллера в целом.

Функциональные характеристики контроллера КРОСС приведены в таблице 2.

Таблица 2 - Общие функциональные характеристики контроллера КРОСС

Наименование параметра		Характеристика
Операционная система реального времени (ОС РВ)		OS-9 фирмы MICROWARE
Система программирования в среде ОС РВ		Ultra C
Технологическое программирование (система ISaGRAF, UltraC/C++) на языках:		<ul style="list-style-type: none"> - последовательных функциональных схем SFC; - релейной логики LD; - структурированного текста ST; - функциональных блоков FBD; - инструкций IL;
Объем памяти процессора	<ul style="list-style-type: none"> - flash-память - динамическое ОЗУ - статическое ОЗУ 	1 Мбайт, размер сектора 128 кбайт 1,5 Мбайт 256 кбайт (пользовательский RAM- диск)
Общая емкость памяти программ пользователя (оценочно)		до 10 инструкций входного языка ST ISaGRAF
Часы реального времени		Формат: секунды - минуты - часы - года
Время цикла		более или равно 0,01 с
Время сохранения технологических программ (статическое ОЗУ) при отключении питания		не менее 2 суток при температуре плюс 25 °С, типовое значение 5 суток
Время выполнения 10 ³ логических инструкций, мс		около 8

Параметры интерфейсных сигналов приведены в таблице 3.

Таблица 3 - Параметры интерфейсных сигналов

Интерфейс	Характеристика
RS232:	два последовательных порта RS232:
- главный порт	поддерживает интерфейс типа «полный модем», имеет на ЦБ1 разъем «RS232» типа RJ45;
- вспомогательный порт	поддерживает линии RxD и TxD без аппаратной поддержки «рукопожатия»; имеет на ЦБ1 зажимы «под винт»;
Ethernet:	имеет некритичные ко времени доставки сетевые коммуникации, передача информации пакетами на большие расстояния; имеет на ЦБ1 порт (разъем) «ETHERNET» (субмодуль ETHERNET SM2-ETH);
- протокол	TCP/IP;
- скорость обмена	10 Мбод (IEEE 802.3);
- тип разъема «Ethernet»	RJ45 для 2 витых пар кабеля 5 UTP/STP (10BASE-T);
- гальваническая изоляция	1 кВ постоянного тока
SPI:	выполняет обмен между ЦБ1 и модулями УСО; один порт на модулях ввода/вывода, два порта на ИСК1;
- скорость обмена	максимальная частота 1 МГц;
- тип разъема «SPI»	IDC, 10 контактов
- гальваническая изоляция	отсутствует

1.2.3 Параметры модулей УСО и входы-выходы контроллера

УСО - Устойство связи с объектом.

Модули аналогового и дискретного ввода / вывода (модули УСО) и их параметры показаны в таблице 4.

Таблица 4- Параметры модулей УСО

Модуль УСО	Вид и количество каналов на модуль		Входной/ выходной сигнал	Входное сопротивление (сопротивление нагрузки)	Максимальная потребляемая мощность, Вт, не более по цепи	
					5В	24В
ТС1-7	Входы (1-7)		Термопары по таблице 5	не менее 100	1,25	-
	Вход 8		Термопреобразователь сопротивления ТСМ50'	-		
TR1-8	Входы (1-8)		Термопреобразователи сопротивления по таблице 4	-	1,45	-
АП -8	Входы (1-8)		Аналоговый сигнал: - постоянный ток (0-5) мА, (0-20) мА, (4-20) мА; - напряжение (0-10) В	(400±10) Ом	1,2	-
АЮ1-8/0	Входы (1-8)			(100±3) Ом	0,5	-
АЮ1-8/4	Входы (1-8)			не менее 10 кОм	0,6	*
	Выходы (1-4)		Аналоговый сигнал: - постоянный ток (0-5) мА (0-20) мА, (4-20) мА;	не более 2 кОм не более 0,5 кОм		1,05 2,5
АЮ1-0/4	Выходы (1-4)				0,12	
ДП -16	Входы	(1-16)	Дискретный сигнал: - напряжение постоянного тока	не менее 2кОм	0,12	*
ДИО1-8/8	Входы (1-8)		(0-7) В - логический "0", (24±6) В - логическая "1 "		0,4	
	Выходы (1-8)		Дискретный сигнал: - бесконтактный ключ напряжение до 40В ток до 2А			**
ДО1-16	Выходы (1-16)				0,7	**
* Максимальный ток 0,01 А на один канал						
** Максимальный ток не более: 0,3 А на один канал, 2 А на 8 каналов.						

Контроллер включает измерительный канал (АЦП) для конвертации аналоговых сигналов в цифровой код и измерительный канал (ЦАП) для конвертации цифрового кода в постоянный ток.

Характеристики измерения данного контроллера показаны в таблице 5.

Таблица 5 – Метрологические характеристики контроллера

Модуль	Сигналы		Пределы допускаемой основной приведенной (γ ₀ , %) или абсолютной (Δ ₀ , °С)	Пределы допускаемой дополнительной погрешности при изменении температуры на 10 °С
	на входе ***	на выходе ***		
ТС1-7	(0-65) мВ	(0-100,0)%	± 0,2%	0,5γ ₀
	ТПП(К), (0-20,620) мВ	(0-1730,0) °С	± 3,5 °С *	0,5Δ ₀
	ТПП(8), (0-1 8,285) мВ	(0-1730,0) °С	± 3,5 °С *	
	ТПР(В), (0,431-13,361) мВ	(300,0-1780,0) °С	± 3,0 °С **	
	ТЖК(Ж), (0-64,948) мВ	(0-1120,0) °С	± 2,2 °С *	
	ТМК(Т), (0-20,255) мВ	(0-390,0) °С	± 0,8 °С *	
	ТХК _H (Е), (0-64,922) мВ	(0-850,0) °С	± 1,7 °С *	
	ТХА(К), (0-53,795) мВ	(0-1 340,0) °С	± 2,7 °С *	
	ТНН(Н), (0-46,425) мВ	(0-1270,0) °С	± 2,5 °С *	
	ТВР(А1), (0-33,252) мВ	(0-2450,0) °С	± 4,9 °С *	
	ТВР(А2), (0-26,773) мВ	(0-1760,0) °С	± 3,5 °С *	
	ТВР(А3), (0-26,31 6) мВ	(0-1760,0) °С	± 3,5 °С *	
	ТХК(Л), (0-64,771) мВ	(0-780,0) °С	± 1,6 °С *	
	ТСМ50', (50-92,800) Ом	(0-200,0) °С	± 0,4 °С	1,0Δ ₀
ТР1-8	ТСП150, (50-99,937) Ом	(0-266,0) °С	± 0,5 °С	1,0Δ ₀
	ТСП150', (50-99,996) Ом	(0-262,0) °С	± 0,5 °С	
	ТСМ50, (50-92,61 5) Ом	(0-200,0) °С	± 0,4 °С	
	ТСМ50', (50-92,800) Ом	(0-200,0) °С	± 0,4 °С	
	ТСП100, (100-199,876) Ом	(0-266,0) °С	± 0,5 °С	
	ТСП100', (100-199,992) Ом	(0-262,0) °С	± 0,5 °С	
	ТСМ100, (100-185,230) Ом	(0-200,0) °С	± 0,4 °С	
	ТСМ100', (100-185,600) Ом	(0-200,0) °С	± 0,4 °С	
АП1 -8	Напряжение постоянного тока (0-10) В	13 бит	± 0,2%	0,5γ ₀
	Постоянный ток (0-5), (0-20), (4-20) мА	13 бит	± 0,2%	
АЮ1-8/4, АЮ1-8/0	Напряжение постоянного тока (0-10) В	13 бит	± 0,2%	0,5γ ₀
	Постоянный ток (0-5), (0-20), (4-20) мА	13 бит	± 0,2%	
АЮ1-8/4, АЮ1-0/4	12 бит	Постоянный ток (0-5) мА	± 0,2%	0,5γ ₀
	12 бит	Постоянный ток (0-20), (4-20) мА	± 0,2%	
* Погрешность приведена с учетом компенсации температуры холодных спаев.				
** Погрешность приведена без учета компенсации температуры холодных спаев.				
*** Нормирующие значения сигнала равны конечному значению диапазона.				

Параметры аналоговых и частотных сигналов ячеек модулей АЮ1, АЮ2, МК1 приведены в таблицах 4, 5.

1.2.4 Электрическое питание контроллера КРОСС

Источник питания для контроллера зависит от заказа и имеет следующие варианты:

Однофазное питание переменного тока с номинальным напряжением 220 В, с допустимым диапазоном перепада от минус 33 до плюс 22 В.

Максимальная мощность для работы контроллера, в пределах:

- 45 ВА от сети переменного тока напряжением 220 В;
- 40 Вт от внешнего нестабилизированного источника постоянного тока.

Данные о мощности, потребляемой модулем УСО, приведены в таблице 4.

1.2.5 Параметры изоляции и гальваническое разделение цепей

Схемы контроллера и блока питания LOK 4601-2R/P-ONE, а также изоляция между ними выдерживают одну минуту действительного синусоидального испытательного напряжения переменного тока частотой 50 Гц со среднеквадратическим значением:

- 500 В для цепей с рабочим напряжением до 42 В;
- 1500В для цепей с рабочим напряжением от 60 до 250 В.

Цепи модулей AI1-8, TR1-8, TC1-7 имеют гальваническое разделение:

- между входами;
- между цифровой шиной SPI (контактами разъема «SPI») и каждым входом.

Цепи модулей AIO1-8/4, AIO1-8/0, AIO1-0/4, DIO1-8/8, DI1-16; DO1-16 имеют гальваническое разделение:

- между входами и выходами;
- между цифровой шиной SPI и входами/выходами (между контактами разъемов «SPI» и «INOUT»).

1.2.6 Показатели надежности контроллера КРОСС

С учетом требований к техническому обслуживанию, изложенных в инструкции по эксплуатации, среднее время между отказами должно составлять не менее:

- канала аналоговых модулей AI1 -8, AIO1 -8/4, AIO1-0/4, AIO1-8/0, TC1-7, TR1-8 70000 ч;

- канала дискретных модулей DIO1-8/8, DI1-16, DO1-16 100000ч.

Среднее время восстановления 2 часа или менее.

Средний срок использования контроллера составляет не менее 10 лет.

1.3 Устройство контроллера КРОСС

1.3.1 Описание аппаратуры контроллера КРОСС

На рисунке 1 показана принципиальная схема контроллера с обозначенными на рисунке соединениями между модулями. Модуль SM2-CPU-1.5 включает в себя процессор и сетевой контроллер, модуль УСО также имеет процессор. Обмен данными между процессорами осуществляется по шине SPI через модуль ИСК 1.

Обычно к каждой шине SPI может быть подключено восемь модулей УСО. В соответствии с их количеством, здесь могут присутствовать один или два модуля ИСК 1. Модули УСО соединяются с внешним подключением либо через клеммную колодку с клеммником, либо через разъем "INOUT".

1.3.2 Описание конструкции контроллера КРОСС

Контроллер состоит из центрального блока СВ 1, источника питания и модуля питания, модуля ODR, разъема SPI и клеммной колодки.

СВ 1 представляет собой фиксированную конструкцию с тремя гнездами SMART2-BASE, каждое из которых подключено к винтовой клеммной колодке SM2-SCR7. Питание от модуля питания вместе с выбранным пользователем разъемом обеспечивает необходимую мощность для нормальной работы внешнего управления.

Функция модуля ЦП заключается в управлении контроллером, обмене данными с внешними устройствами, такими как ISaGraf, и обеспечении связи пользователя с контроллером через системы OPC и SCADA.

Основная функция модуля питания KP-DC24V1 заключается в обеспечении питания модулей CP1 и ODR.

Источник питания рассчитан на напряжение 220 В и преобразует переменное напряжение в 24 В постоянного тока.

Блоки терминальные T1-AI, T1-AIO, T1-D, T1-TC, T1-TR позволяют подключать к контроллеру кабели различных сечений, приходящие от объекта управления - от датчиков и исполнительных органов.

Соединители гибкие C1-AI, C1-AIO, C1-D, C1-TC, C1-TR соединяют модули УСО и терминальные блоки.

1.3.3 Организации питания

Контроллер имеет необходимое стандартное напряжение, обычно 220 В. Дополнительный источник питания не требуется, так как контроллер поставляется с собственным внутренним модулем питания.

Если потребляемая контроллером мощность ниже $10\text{В} \cdot \text{А}$, то подойдет схема блока питания, показанная на рисунке 2.

В этом случае блок питания подает 24 В на модуль питания DC-24/5. В отличие от обычного контроллера, в контроллер добавляется модуль KP-DC24V1.

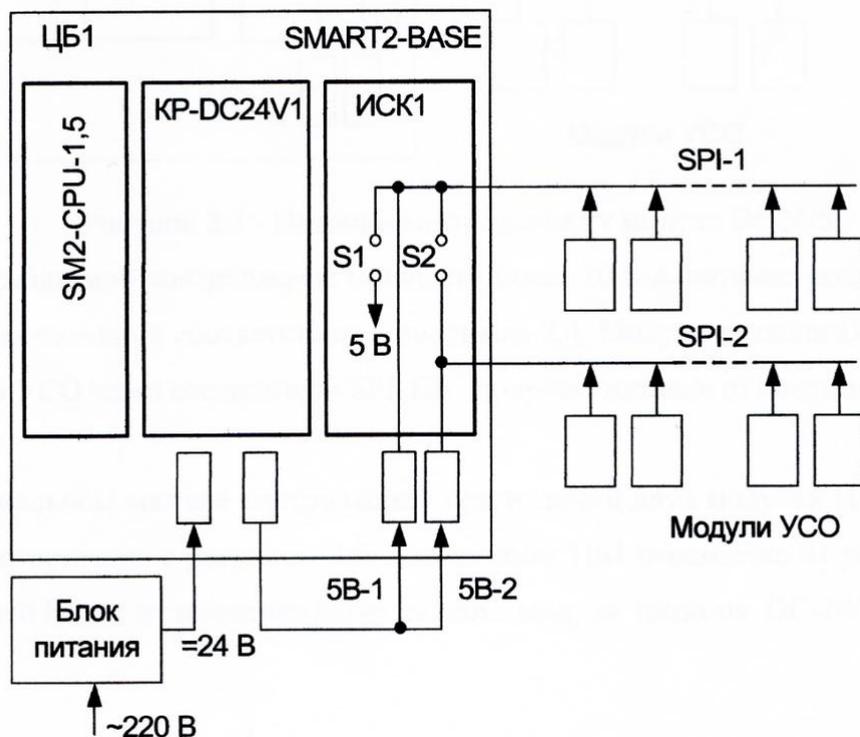


Рисунок 2 – Питание контроллера от модуля KP-DC 24 V1

2 СТРУКТУРНАЯ СХЕМА ЛАБОРАТОРНОГО КОМПЛЕКСА, СХЕМА ВНЕШНИХ СОЕДИНЕНИЙ

2.1 Структурная схема комплекса

В рассматриваемую в данной работе систему автоматического регулирования уровня жидкости входят:

- установка, состоящая из трёх ёмкостей, регулирующего клапана мембранного типа с электроприводом, насоса, одного аналогового и трёх дискретных датчиков измерения уровня (далее по тексту – «Гидравлический объект»);

- стенд с контроллером;

- персональный компьютер – операторская станция.

Блок-схема гидравлической системы управления показана на рисунке 3.

Гидравлический объект включает в себя три водяных бака Е1, Е2 и Е3, датчики и средства воздействия на процесс техники.

Водяной бак Е1 - это напорный бак. Первоначально он заполняется водой через верхнее отверстие, а затем перекачивается в резервуар Е3 в ходе

последующего процесса. Резервуар E1 также имеет переливную трубу и выходной патрубок, к которому подключен исполнительный механизм (ИМ) или пневматический односедельный блок с мембранным приводом. В ручном режиме ИМ управляется с помощью дистанционной панели управления, в автоматическом режиме исполнительный механизм управляется контроллером.

Двигатель исполнительного механизма вращает мембранный вентиль посредством червячной передачи. Мембранный вентиль исполняет роль регулирующего органа. Исполнительный механизм оснащён концевыми выключателями, которые обеспечивают выключение электродвигателя при достижении крайних положений вентиля. Также на валу ИМ расположен указатель положения реостатного типа.

Датчик уровня воды расположен в резервуаре для воды E2. Когда уровень воды в баке меняется, датчик уровня улавливает сигнал и передает сигнал с изменением напряжения. В этот момент преобразователь конвертирует это напряжение в сигнал в виде тока (0-20) мА, который поступает на аналоговый вход контроллера.

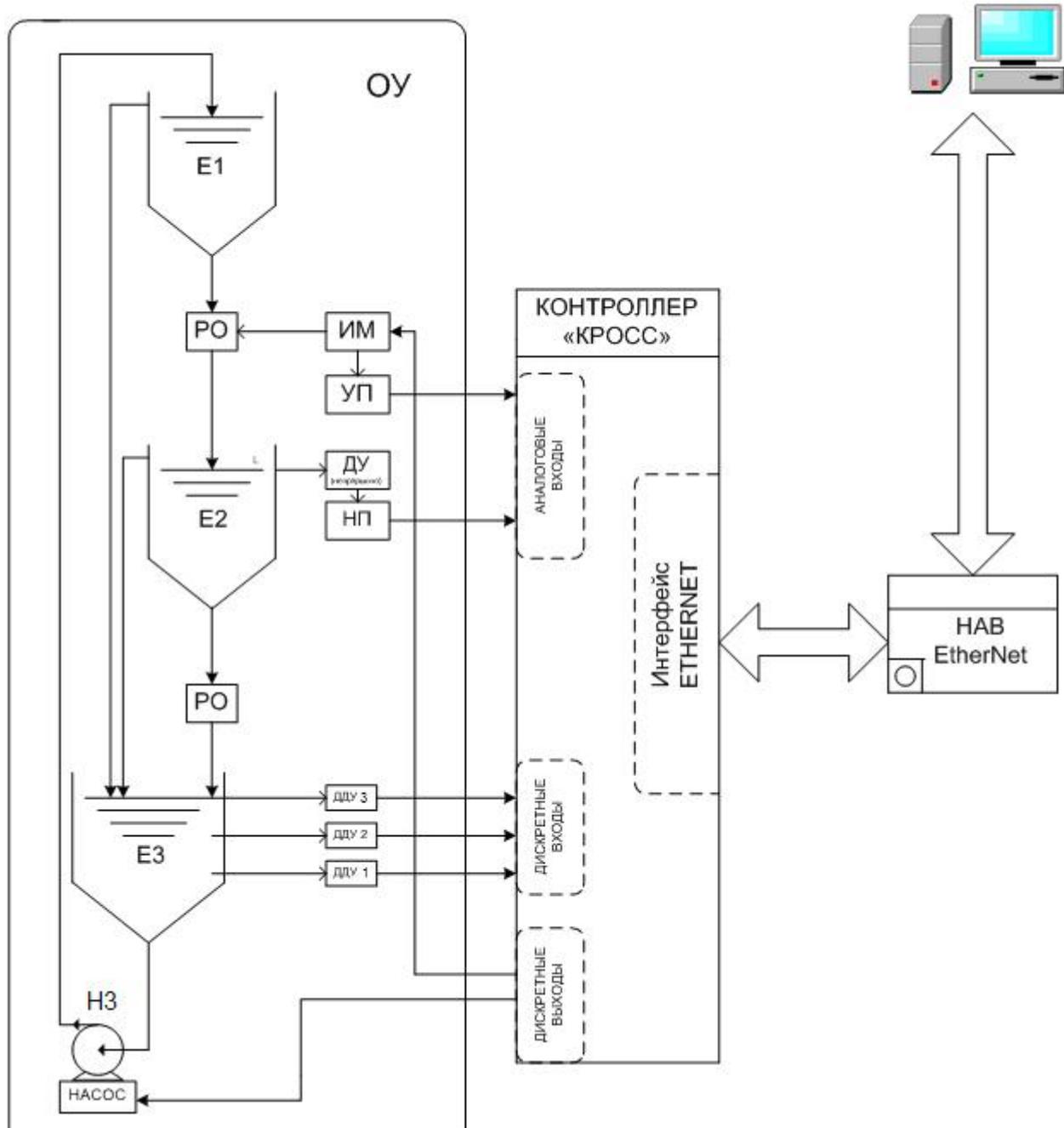


Рисунок 3 – Структурная схема лабораторного комплекса

Между резервуаром E2 и резервуаром E3 находится мембранный клапан, который можно открывать и закрывать для управления водой из E2 в E3, чтобы стабилизировать уровень воды в резервуаре E2.

Резервуар E3 оснащен тремя датчиками уровня, соответствующие датчики уровня посылают сигналы, когда уровень воды в резервуаре E3 находится на разных уровнях. Между резервуаром E3 и резервуаром E1 также

установлен насос НЗ, который может управляться автоматически или вручную, или по сигналам контроллера, для управления уровнем воды в резервуаре ЕЗ.

Функции управления и регулирования в системе выполняет промышленный микропроцессорный контроллер «КРОСС».

В качестве операторской станции в системе используется компьютер типа IBM PC на базе процессора CELERON 2800, с установленной SCADA системой MasterSCADA и системой программирования ISaGRAF.

2.2 Конфигурация контроллера КРОСС. Схема внешних соединений

Контроллер представляет собой набор функционально законченных модулей.

Питание контроллера осуществляется напряжением 24 В постоянного тока от источника питания.

Все входы / выходы контроллера выведены на клеммные соединители (терминальные блоки). Все сигнальные цепи от объекта управления и от операторской станции подключаются через эти соединители.

Схема состоит из следующих устройств (таблица 6).

Таблица 6 – Перечень устройств

Наименование блока	Назначение блока
ЦБ-1	Центральный блок
ЛОК4601	Блок питания контроллера
ДИО1-8/8	Модуль УСО дискретных сигналов контроллера
АИО1-8/4	Модуль УСО аналоговых сигналов контроллера
Т1-ДИО	Терминальный блок ввода/вывода дискретных сигналов
Т1-АИО	Терминальный блок ввода/вывода аналоговых сигналов
БР-1	Блок реле
БС-1	Блок сигнализации

Схема внешних соединений лабораторного стенда приведена на рисунке 4.

Лабораторный стенд состоит из устройств:

- промышленный микропроцессорный контроллер «КРОСС»;
- гидравлический объект.

Питание электрических цепей стенда осуществляется от сети переменного тока 220 В 50 Гц. Подсоединение сигнальных цепей каждого устройства осуществляется через соответствующий кабель и клеммные соединители (терминальные блоки).

Соответствие входов / выходов блоков ввода / вывода используемым сигналам приведено в таблице 7.

Таблица 7 – Используемые блоки и сигналы

Блоки ввода-вывода	Терминальные блоки	Вход №	Выход №	Обозначение	Описание сигнала (переменной)
АЮ1-8/4	Т1-АЮ	4		a10+, a21-	Положение ИМ
		5		a11+, a22-	Уровень в Е2
DЮ1-8/8	Т1-DЮ	2		a7+, a18-	Уровень L3 в Е3
		3		a8+, a19-	Уровень L2 в Е3
		4		a9+, a20-	Уровень L1 в Е3
			2	a1+, a12	Пуск/ Стоп НЗ
			3	a2+, a13	Сигнал Меньше
			4	a3+, a14	Сигнал Больше
			5	a4+, a15	Пред сигн МАХ
			6	a5+, a16	Пред сигн MIN
			7	a6+, a17	Звуковой сигн.

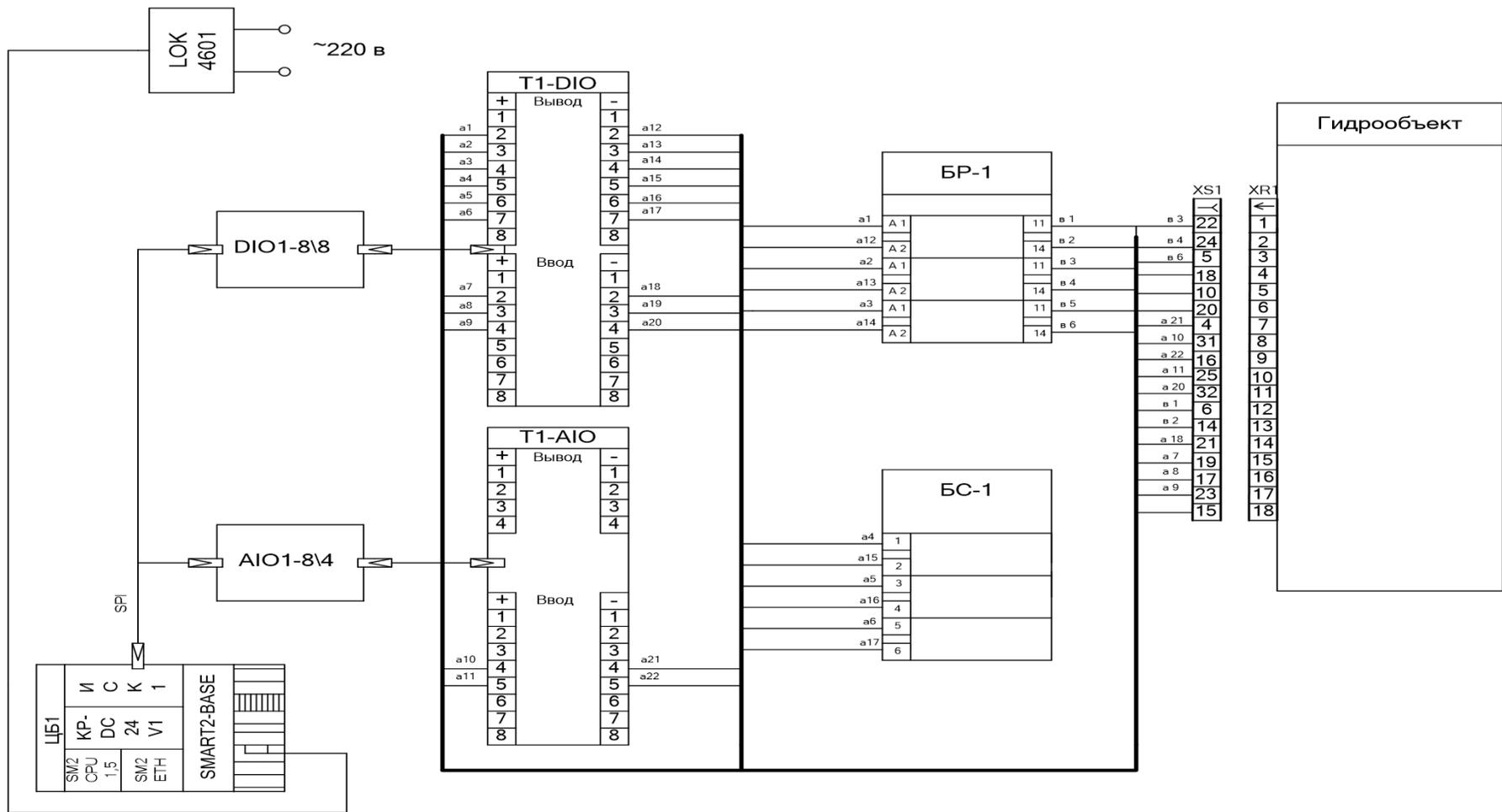


Рисунок 4 – Схема внешних соединений

Карта соединений контроллера с гидравлическим объектом приведена в таблице 8.

Таблица 8 - Карта соединений контроллера с гидравлическим объектом

Гидравлический объект		Контроллер	
Контакт	Цепь	Контакт	Цепь
XR 1: 4	Резерв	XS 1:18	Резерв
XR 1: 5	Резерв	XS 1:10	Резерв
XR 1: 6	Резерв	XS 1:20	Резерв
XR 1: 7	УП + U	XS 1: 4	Аналоговый вход 4+
XR 1: 8	УП – U	XS 1:31	Аналоговый вход 4-
XR 1: 9	ДУ +	XS 1:16	Аналоговый вход 5+
XR 1: 10	ДУ -	XS 1:25	Аналоговый вход 5-
XR 1: 11	ДУ_Общ	XS 1:32	Резерв
XR 1: 12	Насос	XS 1:6	Резерв
XR 1: 13	Насос общ.	XS 1:14	Резерв
XR 1: 14	ДДУ_Общ	XS 1:21	Дискретный вход 2-
XR 1: 15	ДДУ1	XS 1:19	Дискретный вход 2+
XR 1: 16	ДДУ2	XS 1: 17	Дискретный вход 3+
XR 1: 17	ДДУ3	XS 1: 23	Дискретный вход 4+
XR 1: 18	Корпус	XS 1: 15	

3 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЛАБОРАТОРНОГО КОМПЛЕКСА

Программное обеспечение лабораторного комплекса «Система автоматического регулирования уровня на базе контроллера КРОСС» состоит из:

- программное обеспечение нижнего уровня;
- программное обеспечение верхнего уровня.

Программное обеспечение нижнего уровня решает задачи программирования контроллера, загрузки и ввода в эксплуатацию управляющей программы контроллера.

Программное обеспечение нижнего уровня представлено операционной системой OS-9, системой программирования ISaGRAF, сервисной программой конфигуратор, программой управления ГО.

3.1 Основные сведения об операционной системе OS-9

Операционная система OS-9 - это широко используемая в настоящее время высокопроизводительная система с масштабируемой модульной архитектурой, которая поддерживает различные модели операционных систем и имеет усовершенствованный многоуровневый ввод-вывод, включая последовательные коммуникации, дисковые устройства, Ethernet, TCP I/P, ISDN, сетевую файловую систему 8MB и NFS.

OS-9 может работать в минимальной конфигурации, исполняясь непосредственно из ПЗУ и используя менее 100 кбайт памяти.

Конфигурация OS-9 в контроллере КРОСС включает ядро OS-9/68K, версия V3.0.3, и набор ISP, поддерживающий протоколы TCP IP INTERNET, включая сокет, FTP, WEB и TELNET серверы.

OS-9 обеспечивает:

- Объединение учетных записей, поддерживающих несколько пользователей, без использования At Ease;
- Keychain, функция, позволяющая пользователям хранить пароли и зашифрованные текстовые данные в защищенной связке ключей;
- переработанная панель управления звуком с поддержкой USB-аудио;
- Улучшенное управление шрифтами с помощью FontSync;
- Удаленный доступ к Personal Server 3.5, включая поддержку клиентов TCP/IP через протокол Point-to-Point Protocol (PPP);
- Обновленная версия AppleScript для поддержки TCP/IP;
- совместное использование личных файлов через TCP/IP;
- совместное использование принтеров USB - панель управления, позволяющая предоставлять общий доступ к определенным принтерам USB через сети TCP/IP;

Программное обеспечение имеет рамочную структуру, которая может быть полностью адаптирована к стандартным и нестандартным протоколам, и на этой основе программируется на ряде распространенных языков. С позволяет использовать его для решения широкого круга прикладных задач.

Интегрированная операционная система делает его сравнимым с настольным компьютером общего назначения. Поэтому серию CROSS можно рассматривать как "два в одном" (компьютер общего назначения плюс промышленный контроллер).

Однако операционная система OS-9 может быть сложной для освоения типичными промышленными пользователями, которые привыкли программировать обычные программируемые логические контроллеры (ПЛК). Чтобы удовлетворить потребности таких пользователей, в базовое программное обеспечение контроллера было включено так называемое "ядро ISaGRAF" (ISaGRAF - это наиболее известная и успешная реализация стандарта МЭК 1131-3 от CJ International).

3.2 Краткие сведения о системе ISaGRAF

ISaGRA - это программная технология промышленной автоматизации для создания автономных или распределенных систем управления. Она сочетает в себе механизм управления (микропрограмму) и интуитивно понятную среду разработки приложений (верстак). Технология ISaGRAF разработана с учетом масштабируемости и подходит для разработки как миниатюрных контроллеров, так и крупных распределенных систем автоматизации.

Когда пользователь создает новую программу, она загружается в контроллер для выполнения. Интерпретатор работает по принципу, что интерпретация и представление алгоритма пользователя происходит во время выполнения программы. В этом случае код централизован, и нет необходимости переключаться на другой контроллер, что снижает энергопотребление.

Программы ISaGRAF доступны на шести языках стандарта IEC 1131-3: четырех графических (SFC, FC, FBD, LD) и двух текстовых (ST и IL). Технология встроенного программного обеспечения ISaGRAF доступна в виде исходного кода на языке C как часть инструментария встроенного программного обеспечения ISaGRAF. Код сформирован в хорошо

документированные, переносимые и настраиваемые файлы. Следуя подробным пошаговым инструкциям в документации к инструментарию, вы можете настроить свой собственный механизм управления, реализовав различные интерфейсы ОС и аппаратного обеспечения, коммуникационные уровни и свои основные знания в функциях и функциональных блоках языка С, а также самостоятельные задачи.

Благодаря этому контроллер сочетает в себе операционную систему реального времени OS-9 плюс традиции ПЛК, что приведет к появлению других функций:

- Экспериментатор может быть квалифицированным в программировании контроллера и может игнорировать новые знания OS-9;
- специалист, кто знаком с OS-9 и может получить удовольствие от программирования на С;
- специалист-универсалы могут использовать все преимущества обоих вариантов и получить фантастическую гибкость.

Эти контроллеры разрабатываются программно при поддержке очень продвинутого и интеллектуального программного обеспечения: на сайте

3.3 Конфигуратор

Конфигуратор (рисунок 5) предназначен для просмотра входных и выходных значений сигналов, изменения адресов модулей УСО, изменения некоторых параметров входов и выходов.

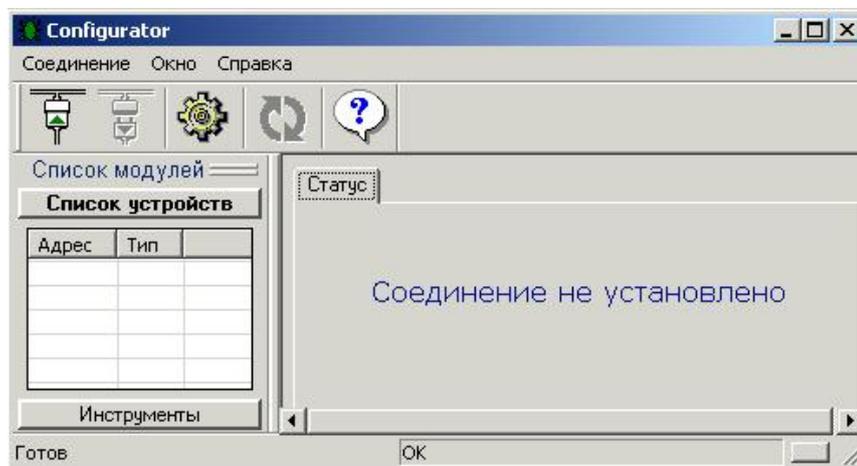


Рисунок 5 – Окно конфигуратора

При работе с конфигуратором первым делом устанавливают параметры связи ПК с контроллером. Для этого после загрузки «Конфигуратор» нажмите **F10** или нажмите иконку на панели, чтобы войти в меню **Соединение/Параметры** (рис. 6).

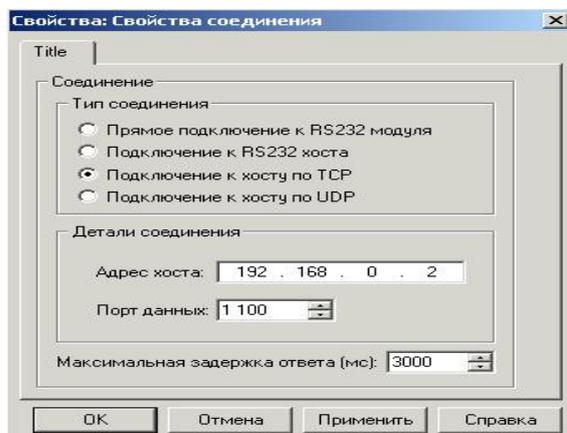


Рисунок 6 - Окно настроек подключения контроллера

После этого устанавливаем связь КОНФИГУРАТОРА с контроллером, посредством выбора меню **Соединение/Установить**, или клавишей **F2**, или нажав иконку установления связи на панели управления . Конфигуратор выдаст список модулей контроллера. Окно конфигуратора будет выглядеть, как показано на рисунке 7.

Список модулей в нашем случае содержит: центральный процессор контроллера(CROSS CPU), дискретный(DIO1-8/8) и аналоговый(AIO1-8/4) модули ввода/вывода, а также модуль термопреобразователей(TC1-7). Можно выбрать и получить доступ к меню устройств, как показано на рисунке 8, где можно посмотреть входы и выходы модулей, а также некоторые их параметры.

После работы с КОНФИГУРАТОРОМ нужно разорвать соединение и закрыть его окно, выбрав меню **Соединение/Разорвать**, или нажав клавишу **F3**, или нажав кнопку на панели .

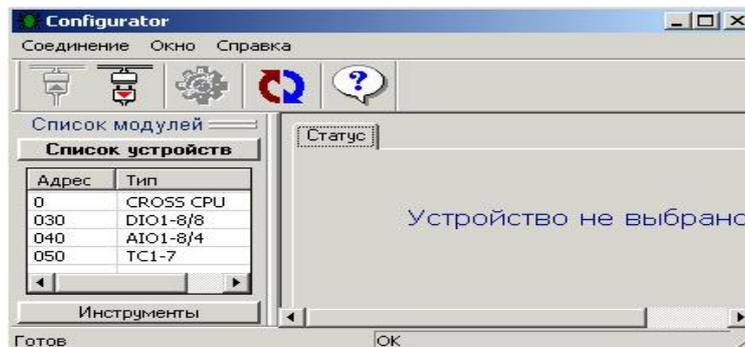


Рисунок 7 - Окно конфигуратора после установления связи с контроллером

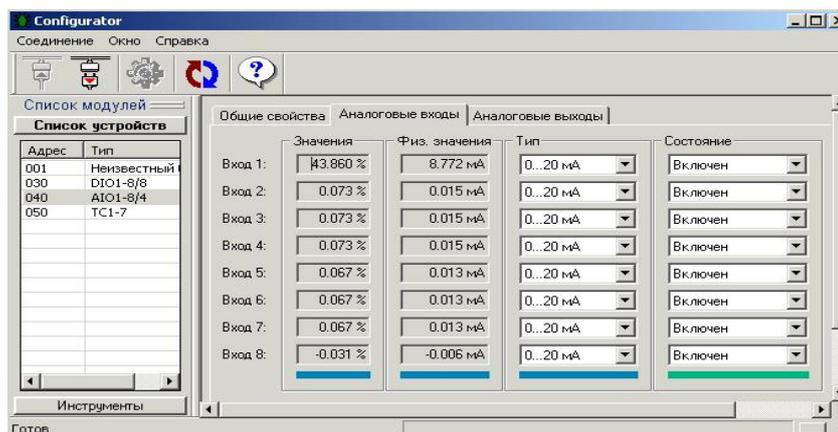


Рисунок 8 - Меню аналогового модуля ввода/вывода AIO1-8/4

Программное обеспечение верхнего уровня отвечает за обмен информацией между контроллером и операторской станцией, предоставление единого адресного пространства среди всех приложений, оперативное управление и архивирование, также содержит средства разработки графических мнемосхем технологических процессов. Программное обеспечение верхнего уровня представлено SCADA пакетом «MasterSCADA»(см.п. 5) и OPC сервером .

3.4 OPC сервер

Технология OPC относится к созданию единой спецификации доступа к данным между промышленным управляющим оборудованием и управляющим программным обеспечением с целью установления стандарта интерфейса для связи между приложениями промышленных систем управления. Она обеспечивает стандартный механизм доступа к данным в области

промышленного управления, эффективно разделяя аппаратное и прикладное программное обеспечение. Это набор независимых от производителя стандартных интерфейсов и протоколов для обмена данными между программным обеспечением, в основном предназначенных для обмена данными между системами управления технологическими процессами и их источниками данных, обеспечивая прозрачный доступ к данным между приложениями.

Формат файла crossopc.ini (рис. 9).

```
# пример конфигурационного файла OPC-сервера
# строка начинающаяся с '#' - комментарий

[server]
# частота обновления OPC групп, ms
scanrate=100

# OPC драйвер для SCADA Citect 5.21 имеет свойство отображать как BAD все теги на
# странице, если хотя бы у одного из них качество BAD
# при ключе citect=1 OPC сервер для все тегов принудительно возвращает
# OPC_QUALITY_GOOD
# состояние связи можно проверить с помощью тега state
citect=0

# уровень детализации лога 0 - не записывать ничего, 4 записывать все сообщения
loglevel=1

# здесь описываются сетевые каналы
[connections]
# пример соединения по протоколу TCP/IP
# указан протокол udp, IP адрес и порт
connection1=udp,192.168.0.2,1100
#connection2=serial,COM2,9600
connection1=udp,192.168.0.2,5066
#connection3=serial,COM1,19200

# здесь описываются контроллеры
[targets]
#соединение, номер MODBUS, путь к файлу с описанием переменных
#t1=connection1,1,"C:\\isawin\\APL\\P_REG2\\appli.tst"
#t1=connection1,1,"c:\\isawin\\APL\\DIPLOM\\appli.tst"
#t1=connection1,1,"C:\\isawin\\APL\\gidrob\\appli.tst"
#t7=connection1,1,"c:\\isawin\\apl\\slava\\appli.tst"
#t8=connection1,1,"C:\\isawin\\APL\\DIPLIOM\\appli.tst"
#controller2=connection2,1,"appli.tst"
#t1=connection1,1,"c:\\isawin\\APL\\OCHRIOV\\appli.tst"
t1=connection1,1,"c:\\isawin\\APL\\liyanch\\appli.tst"
#t1=connection1,1,"C:\\isawin\\APL\\liyanch\\appli.tst"
#t1=connection1,1,"c:\\isawin\\APL\\BKPLYC\\appli.tst"

# В секции [Backup] указываются резервируемые контроллеры
#[backup]
#controller3 = controller1 & controller2
```

Рисунок 9 - Блокнот CROSS OPC

OPC может взаимодействовать с любым источником данных без необходимости установки каких-либо драйверов устройств, при условии, что источник данных должен быть устройством с поддержкой OPC, в то время как

проприетарное программное обеспечение требует установки дополнительных драйверов устройств.

OPC поддерживает доступ к данным в режиме реального времени, регистрацию исторических данных, включение данных о тревогах и событиях и т.д. Он состоит из двух частей: клиента OPC и сервера OPC, который представляет собой программное приложение или стандартный драйвер, предназначенный для доступа к данным реального времени и обеспечения других функций от различных производителей, таких как обработка событий, ведение журнала и т.д. Он выступает в качестве клиента OPC и локального сервера OPC. Он действует как транслятор между OPC-клиентом и локально взаимодействующим источником данных. Он также имеет возможность "чтения" и "записи" в источники данных.

OPC обеспечивает взаимодействие основного клиента с данными установки, предоставляя различные спецификации (например, аварийные сигналы, обработка событий, исторические данные и т.д.). Используя одну и ту же спецификацию OPC, сервер OPC может взаимодействовать с клиентами разных производителей.

Клиент OPC также является программным приложением, предназначенным для связи с сервером OPC. Он фактически является приемником данных, который преобразует коммуникационные запросы приложения в запросы OPC и отправляет их на сервер OPC. Когда данные прочитаны, клиент преобразует их обратно в родной коммуникационный формат приложения. Это программные модули, встроенные в приложение, такие как HMI, исторические базы данных и т.д., чтобы они могли запрашивать и направлять программное обеспечение OPC-сервера. Они также способны взаимодействовать с различными OPC-серверами.

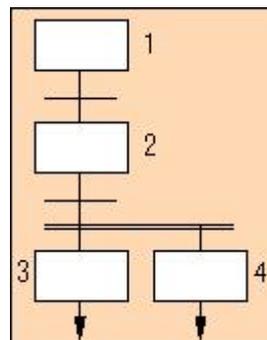
Как показано выше, после завершения настройки OPC его можно нормально использовать для передачи данных для достижения коммуникационного обмена.

3.5 Языки программирования

1. Язык управляющих последовательностей (SFC).
2. Язык функциональных блоков (FBD).
3. Язык релейной логики (LD).
4. Структурированный текст (ST).
5. Язык инструкций (IL).

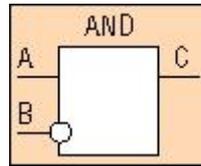
1. SFC.

При использовании завершенной функции в качестве основной линии, преимущество заключается в том, что процесс работы четко организован, что облегчает понимание и обдумывание процесса работы программы; для больших программ дизайн может быть разделен, используется более гибкая структура программы, что экономит время разработки программы и время отладки, а поскольку сканируются только активные шаги, время выполнения программы может быть сокращено.



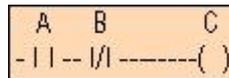
2. FBD.

Используя функциональный блок в качестве единицы проектирования, можно начать с функции управления. Преимущество заключается в том, что это облегчает анализ и понимание решения по управлению, а функциональный блок интуитивно понятен, прост для восприятия и обладает хорошей работоспособностью. При работе со сложными системами управления они все еще могут быть четко описаны графически. Недостатком является то, что каждый функциональный блок занимает место в памяти программы и увеличивает цикл выполнения программы.



3. LD.

В отличие от электрических операционных схем, их преимущества в том, что они интуитивно понятны и просты для восприятия и изучения электротехническим персоналом. Недостатком является то, что при работе с программированием сложных систем управления часто описательный характер программы недостаточно ясен. Лестничная диаграмма в отечественной промышленной автоматизации является наиболее используемым языком программирования ПЛК.



4. ST.

Преимущества в том, что он позволяет осуществлять сложное арифметическое управление и требует высокого уровня навыков программирования. Недостаток в том, что код должен быть преобразован в машинный язык во время компиляции, что может привести к длительному времени компиляции, медленному выполнению, а также плохой интуитивности и простоте использования.



5. IL.

Преимущества: легкость запоминания и освоения, соответствие лестничным диаграммам (ЛД), легкая взаимозаменяемость и проверка программ, отсутствие ограничений на размер экрана и элементы ввода при программировании и отладке. Недостаток: как и в случае с лестничными диаграммами, описание программы для сложной системы не является достаточно четким.

LD	A
ANDN	B
ST	C

3.6 Разработка программы для САР уровня на основе языка программирования FBD

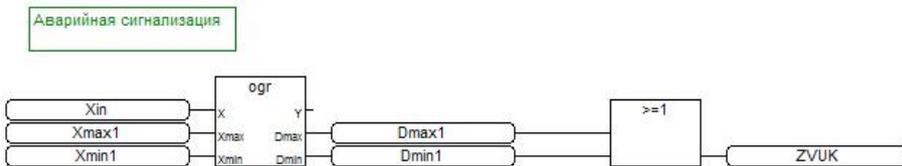
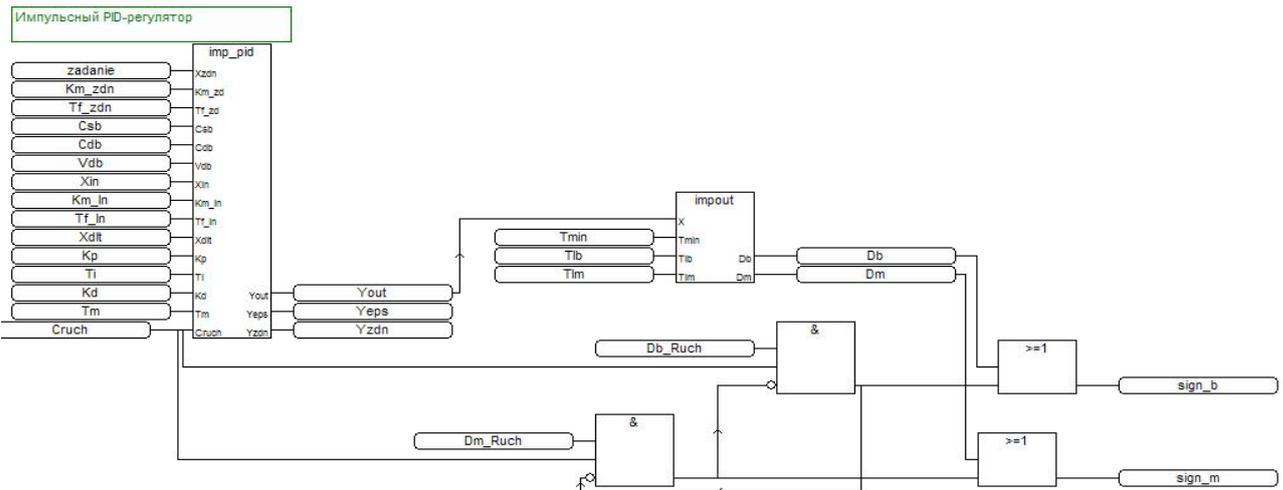
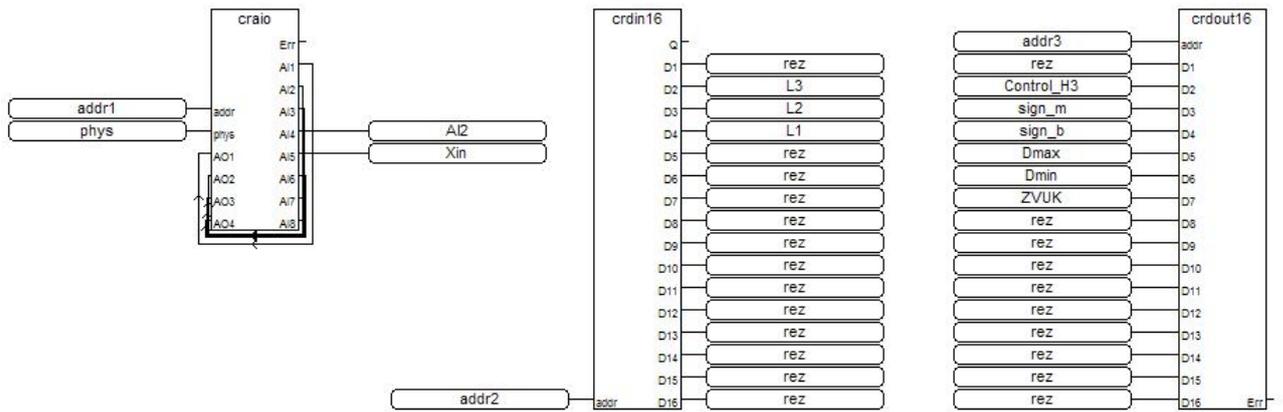
FBD представляет логику управления в виде блок-схем, похожих на язык программирования цифровых схем логических вентилей.

Язык FBD прост в освоении для тех, кто имеет базовые знания о цифровых схемах. Язык программирования использует ячейки, похожие на с-гейты и or-гейты, для представления отношений логических операций, где левая сторона ячейки является входной переменной для логической операции, а правая - выходной переменной; сигналы также идут слева направо, и отдельные функциональные блоки могут быть соединены последовательно друг с другом, или могут быть вставлены промежуточные сигналы.

Количество ячеек для объединения логических операций перед каждым конечным выходом ограничено, а количество выходных результатов одного и того же набора логических операций также варьируется в зависимости от операционной системы; после расширения можно представлять не только различные простые логические операции, но и сложные операции и операционные функции.

3.6.1 Программа управления гидравлическим объектом

Программа управления гидравлическим объектом представлена на рисунке 10.



Регулирование уровня в ЕЗ и управление насосом НЗ

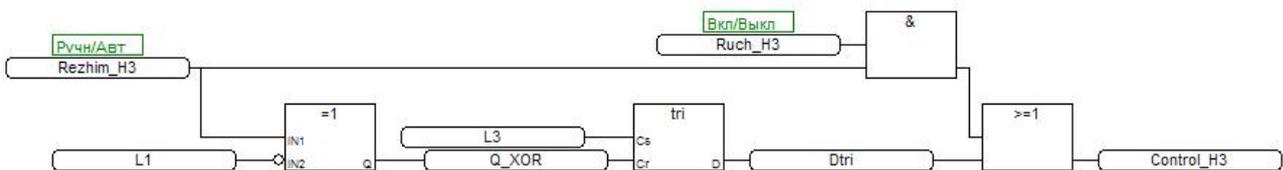
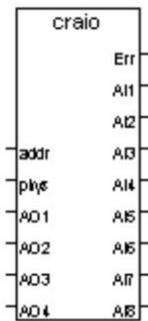


Рисунок 10 – Программа управления гидравлическим объектом

Ниже приводится описание используемых в программе функциональных блоков и программных модулей, предназначенных для автоматического регулирования, ручного дистанционного управления и сигнализации.

3.6.2 Функциональные блоки используемые при создании программы

Ввод-вывод аналоговых сигналов с модуля УСО КРОСС (craio).



Входы: **addr** – сетевой адрес модуля;

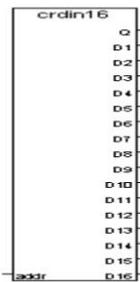
phys – признак необходимости преобразования (TRUE – преобразовывать в физическую величину; FALSE – не преобразовывать, выдавать сигнал в процентах от диапазона измерения);

AI[1-8] – значения входов.

Выходы: **AO[1-4]** – значения выходов; **err** – код ошибки.

Этот функциональный блок нужен нам для подключения входных и выходных переменных непосредственно к входам/выходам аналогового модуля АЮ1-8/4.

Ввод дискретных сигналов с модуля УСО КРОСС (crdin16).



addr – сетевой адрес модуля; **D[1-16]** – состояние дискретных входов; **Q** – код ошибки.

Данный функциональный блок необходим нам для связи входных переменных непосредственно с входами дискретного модуля ДЮ1-8/8.

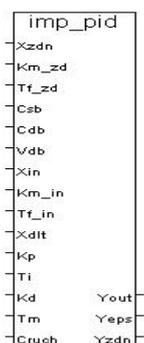
Вывод дискретных сигналов с модуля УСО КРОСС (crdout16).



addr – сетевой адрес модуля; **D[1-16]** – состояние дискретных входов; **Err** – код ошибки.

Данный функциональный блок необходим нам для связи выходных переменных непосредственно с выходами дискретного модуля ДЮ1-8/8.

Импульсный ПИД-регулятор.



Xzdn – вход задания;

Km_zd – постоянная масштабирования Xzdn;

Tf_zd – постоянная фильтра Xzdn;

Csb – Включение статической балансировки;

Cdb – Включение динамической балансировки;

Vdb – Скорость динамической балансировки;

Xin – Вход параметра;

Km_in – постоянная масштабирования Xin;

Tf_in – постоянная фильтра Xin;

Xdlt – Зона нечувствительности;

Kp – Коэффициент пропорциональности;

Ti – Постоянная времени интегрирования;

Kd – Коэффициент времени дифференцирования;

Tm – Время полного хода исполнительного механизма;

Cruch – включение ручного режима;

Yout – Основной выход алгоритма;

Yeps – Сигнал рассогласования;

Yzdn – Выход текущего задания;

Дифференциального уравнения имеет вид:

$$U(t) = K_p * \left(\varepsilon(t) + \frac{1}{T_i} \int_0^t \varepsilon(\tau) d\tau + T_d \frac{d\varepsilon(\tau)}{d\tau} \right), \text{ где}$$

U(t)- выходной сигнал;

$\varepsilon(t)$ - сигнал ошибки;

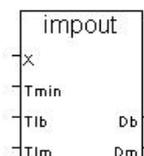
Кп, Ти, Тд – пропорциональная, интегральная, и дифференциальная составляющие регулятора соответственно.

Разностное уравнение регулятора:

$$U[k] = U[k-1] + g_0 * \varepsilon[k] + g_1 * \varepsilon[k-1] + g_2 * \varepsilon[k-2], \text{ где}$$

$$g_0 = Kn * \left(1 + \frac{Td}{To}\right); \quad g_1 = -Kn * \left(1 + 2 * \frac{Td}{To} - \frac{To}{Tu}\right); \quad g_2 = Kn * \frac{Td}{To}.$$

Импульсатор.



name: IMP – импульсатор;

Импульсатор использует RC-осциллятор или другую колебательную цепь для генерации периодического выходного импульсного сигнала. Основная структура состоит из осциллятора, делителя частоты, буферного усилителя и выходной цепи. В генераторе импульсов осциллятор генерирует сигнал квадратной волны с фиксированной частотой и амплитудой и делит его на нужную частоту импульса с помощью делителя частоты.

Методы регулировки импульсаторы При использовании генератора импульсов необходимо произвести правильную регулировку, чтобы получить более точный и стабильный выходной сигнал. Ниже приведены некоторые распространенные методы настройки генератора импульсов: Настройка частоты: Изменение частоты колебаний путем изменения емкости или сопротивления в цепи генератора.

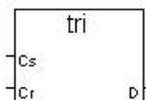
В прикладных программах правильная настройка импульсаторы имеет большое значение для обеспечения правильной работы схемы. Понимание принципа работы генератора импульсов и распространенных методов настройки поможет нам правильно использовать этот компонент.

Если $X > T_{lb}$, то выходе $D_b = 1$ (больше), если $X < T_{lm}$, то на выходе $D_m = 1$ (меньше).

$T_i = 0$ (т.е. импульсы не генерируются).

Алгоритм не содержит каскадных входов и выходов и осуществляет блокировку процедуры обратного отсчета.

RS-Триггер.



name: TRI - RS-триггер

description:

Алгоритм применяется для запоминания дискретных сигналов.

call:

BOOLEAN Cs // команда установки

BOOLEAN Cr // команда сброса

return:

BOOLEAN D // выход

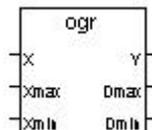
notes:

Cs означает set. Когда Cs =1 и Cr=0, поскольку Cs =1, независимо от того, будет ли состояние D перед установкой 0 или 1, мы знаем из таблицы истинности, что на выходе всегда будет 0! В беззатворном устройстве это состояние называется "установлено в единицу".

R означает сброс, при установке Cr=1 и Cs =0, поскольку Cr=1, независимо от того, что состояние не D перед установкой равно 0 или 1, на выходе будет результат 0, т.е. D=0, т.е. мы называем Cr=1, если хотим, чтобы состояние флип-флопа могло быть сброшено обратно в 0. (Конечно, после D=0, так как Cs =0, D=0, поэтому существует не-D=1.) В не-гейте это состояние называется "установлено в ноль".

Ограничение ogr.

Алгоритм применяется для ограничения верхнего и/или нижнего пределов диапазона сигнала. Два дискретных выхода D_{max} и D_{min} определяют достижение верхнего и нижнего пределов сигнала.

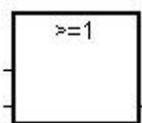


Если $X > X_{max}$, то выход $D_{max}=1$, $Y=D_{max}$.

Если $X < X_{min}$, то выход $D_{min}=1$, $Y=D_{min}$.

Если $X_{min} < X < X_{max}$, то выход $Y=0$.

Логическое ИЛИ.



Замечание: для этого оператора количество входов может быть больше, чем два.

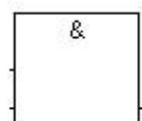
Аргументы:

Inputs BOOLEAN

Outputs BOOLEAN логическое ИЛИ двух и более входов.

Описание: логическое ИЛИ двух и более значений.

Логическое И.



Замечание: для этого оператора количество входов может быть больше, чем два.

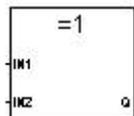
Аргументы:

Inputs BOOLEAN

Outputs BOOLEAN логическое И двух и более входов.

Описание: логическое И двух и более значений.

Исключающее ИЛИ.



Аргументы:

IN1 BOOLEAN

IN2 BOOLEAN

Q BOOLEAN исключающее ИЛИ двух входов.

Описание: исключающее ИЛИ двух значений.

3.6.3 Переменные использованные в программе

Переменные использованные в программе представлены в таблица 9.

Таблица 9 – Переменные программы:

Имя переменной	Тип данных	Атрибуты	Формат	Начальное значение
Zadanie	Целые-действительные	Внутренняя	Вещественная	0
Vdb		Константа	Вещественная	0
Xin		Внутренняя	Вещественная	0
Xdlt		Константа	Вещественная	0,001
Kp		Внутренняя	Вещественная	1
Ti		Внутренняя	Вещественная	15
Kd		Внутренняя	Вещественная	0,1
Tm		Константа	Вещественная	20
Yout		Внутренняя	Вещественная	0
Yeps		Внутренняя	Вещественная	0
Yzdn		Внутренняя	Вещественная	0
Tmin		Внутренняя	Вещественная	0
Tlb		Внутренняя	Вещественная	0
Tlm		Внутренняя	Вещественная	0
Xogr		Внутренняя	Вещественная	0
Xmax		Внутренняя	Вещественная	50
Xmin		Внутренняя	Вещественная	15
AI2		Внутренняя	Вещественная	0
Km_zdn		Внутренняя	Вещественная	0
Tf_zdn		Внутренняя	Вещественная	0
Km_In		Внутренняя	Вещественная	0
Tf_In		Внутренняя	Вещественная	0
addr1		Константа	Целая	40
addr2		Константа	Целая	30
addr3		Константа	Целая	30
Xmax1		Внутренняя	Вещественная	60
Xmin1		Внутренняя	Вещественная	10

Csb	Булевские	Внутренняя		FALSE
Cdb		Внутренняя		
phis		Внутренняя		
Db		Внутренняя		
Dm		Внутренняя		
Dmax		Внутренняя		
Dmin		Внутренняя		
Cruch		Внутренняя		
Rezhim_H3		Внутренняя		
L1		Внутренняя		
L2		Внутренняя		
L3		Внутренняя		
Dtri		Внутренняя		
Ruch_H3		Внутренняя		
Control_H3		Внутренняя		
Dmaxl		Внутренняя		
Dminl		Внутренняя		
ZVUK		Внутренняя		
Db_Ruch		Внутренняя		
Dm_Ruch		Внутренняя		
sign_b		Внутренняя		
sign_m		Внутренняя		
rez		Внутренняя		
Q_XOR		Внутренняя		

3.6.4 Алгоритм работы управляющей программы

Алгоритм работы управляющей программы представлен на рисунке 11.



Рисунок 11 – Алгоритм работы управляющей программы

При опросе каналов контроллер получает данные о значениях дискретных и аналоговых сигналов на входах, после чего происходит получение данных по интерфейсу Enternet о введённых значениях посредством MasterSCADA (таких

как параметры настройки регулятора и т.д.). В зависимости от состояния уровня в емкости Е3 программой формируются сигналы управления насосом НЗ: при заполнении емкости Е3 происходит срабатывание сигнализатора уровня L3, по сигналу которого включается насос НЗ. При достижении уровня жидкости минимального значения происходит срабатывание сигнализатора уровня L1, вследствие чего насос НЗ отключается.

Также программа обеспечивает поддержание (регулирование) уровня жидкости в ёмкости Е2 при определённом задании и формирование сигналов предупредительной световой и аварийной звуковой сигнализации при превышении или понижении входной величины установленных порогов.

После чего происходит передача данных через OPC-сервер в операторскую станцию по интерфейсу Ethernet для использования SCADA - системой MasterSCADA.

3.6.5 Организация ввода/ вывода аналоговых и дискретных сигналов

Организация ввода/ вывода аналоговых и дискретных сигналов реализована в отдельной программе (bl_vv) (Рисунок 12).

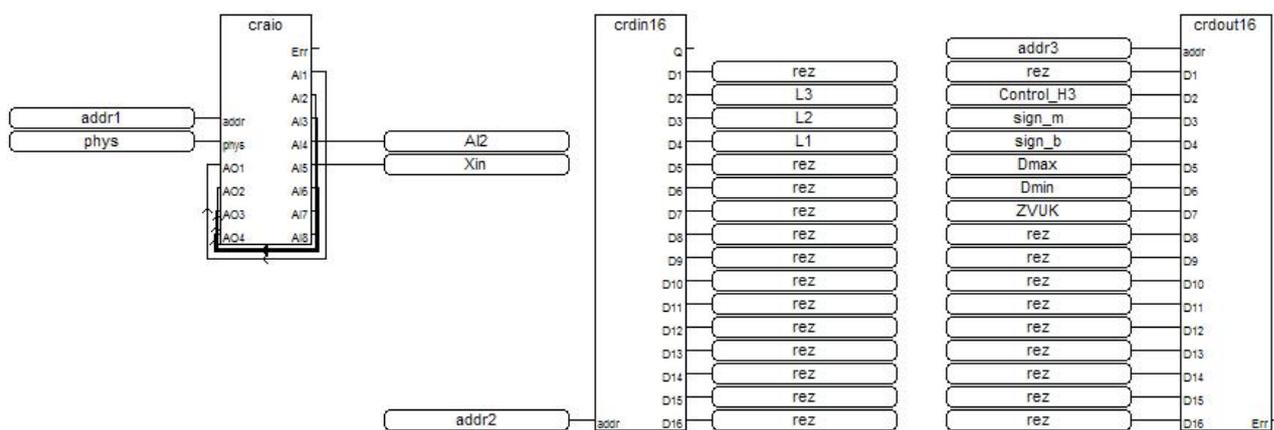


Рисунок 12 – Организация ввода/ вывода аналоговых и дискретных сигналов

Функциональные блоки программы:

Craio – обеспечивает ввод двух аналоговых сигналов: X_{in} – значение уровня жидкости в E2 и AI2 – положение исполнительного механизма в процентах.

Crdin16, Crdout16 – обеспечивают ввод и вывод, соответственно, дискретных сигналов.

Для контроля и управления гидравлическим объектом задействованы следующие входы (выходы).

Аналоговые:

AI4 – сигнал положения исполнительного механизма;

AI5 – сигнал значения уровня в емкости E2;

Дискретные:

DI2 – уровень L3;

DI3 – уровень L2;

DI4 – уровень L1;

DO2 – сигнал на включение/ отключение насоса НЗ;

DO3 – сигнал «Меньше»;

DO4 – сигнал «Больше»;

DO5 – сигнализация по max;

DO6 – сигнализация по min;

DO7 – звуковая сигнализация.

3.6.6 Программа регулирования уровня в емкости E2 входящая в состав программы управления гидравлическим объектом

Данная подпрограмма должна обеспечивать поддержание уровня жидкости в ёмкости E2 в определённом значении. Регулирование уровня осуществляется изменением расхода на притоке по ПИД закону с использованием функционального блока импульсного ПИД-регулятора.

Также данной программой обеспечивается возможность ручного управления ИМ. При значении переменной Cruch – True (Истина) значение сигнала на выходе Y_{out} блока *imp-pid* равно 0 и с помощью переменных

Db_ruch (больше) и Dm_ruch (меньше) можно подавать сигналы управления на выходы (sign_b, sign_m) блока crout16 (Рисунок 13)

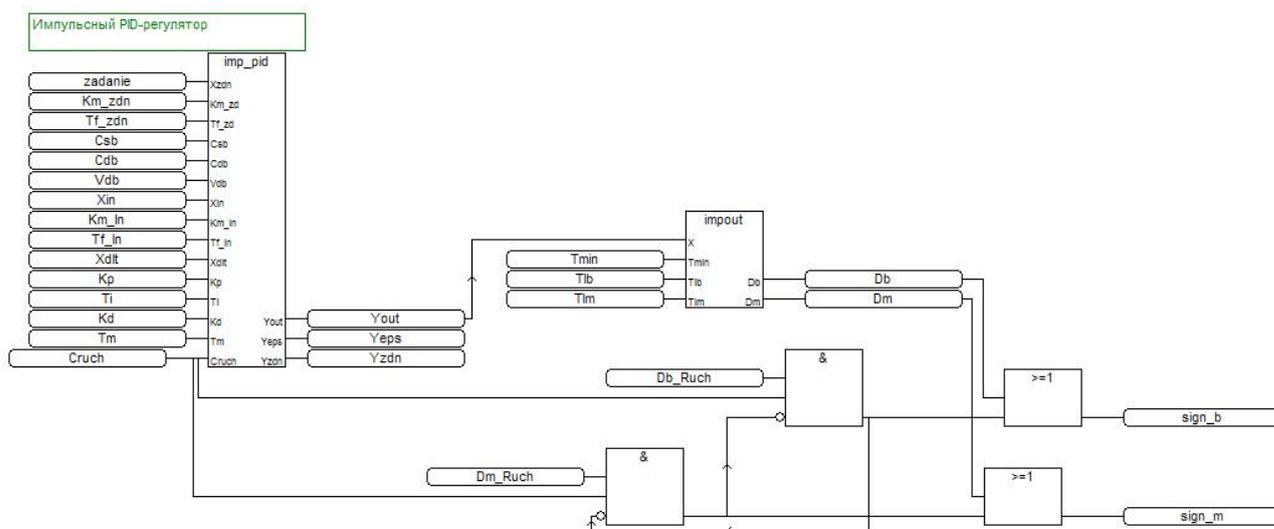


Рисунок 13 – Программа регулирования уровня на языке FBD

Функциональные блоки программы:

Imp_pid – реализует импульсный ПИД-регулятор на входе и выходе аналоговый сигнал.

Impout- импульсатор (на входе аналоговый сигнал на выходе импульсы «Б», «М» с заданной скважностью).

& (И), >=1 (ИЛИ) – служат для реализации ручного управления ИМ.

3.6.7 Программа управления насосом НЗ входящая в состав программы управления гидравлическим объектом

Для правильной работы гидравлического объекта необходимо обеспечить включение насоса при достижении жидкости верхнего дискретного датчика уровня (L3) в ёмкости ЕЗ, и отключение насоса при откачивании жидкости (отключение нижнего дискретного датчика уровня (L1))

Программа управления насосом реализована с использованием RS-триггера (Рисунок 14).

Регулирование уровня в ЕЗ и управление насосом НЗ

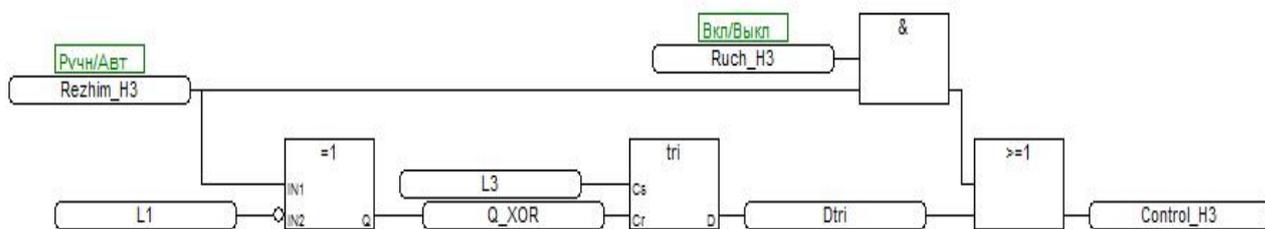


Рисунок 14 – Программа управления насосом НЗ на языке FBD

Переменная L3 отражает состояние уровня L3 в емкости ЕЗ, L1 состояние уровня L1 в емкости ЕЗ. Переменная Rezhim_H3 служит для выбора ручного/ автоматического режима управления насосом НЗ. Переменная Control_H3 привязана к дискретному выходу контроллера, с которого поступает сигнал на управление насосом.

Функциональные блоки программы:

Tri – Rs триггер предназначен для управления насоса по уровню.

=1 - исключающее OR предназначен для реализации функции режима ручного/ автоматического управления насосом НЗ.

& - предназначен для возможности включения ручного управления НЗ.

3.6.8 Программа световой сигнализации входящая в состав программы управления гидравлическим объектом

Данная подпрограмма обеспечивает формирование дискретных сигналов предупредительной световой и аварийной звуковой сигнализации при превышении или понижении входной величины (Xin) установленных значений порогов (Xmax, Xmin). Реализовано это с помощью функциональных блоков OGR. Формирование сигналов приведено на рисунке 15.

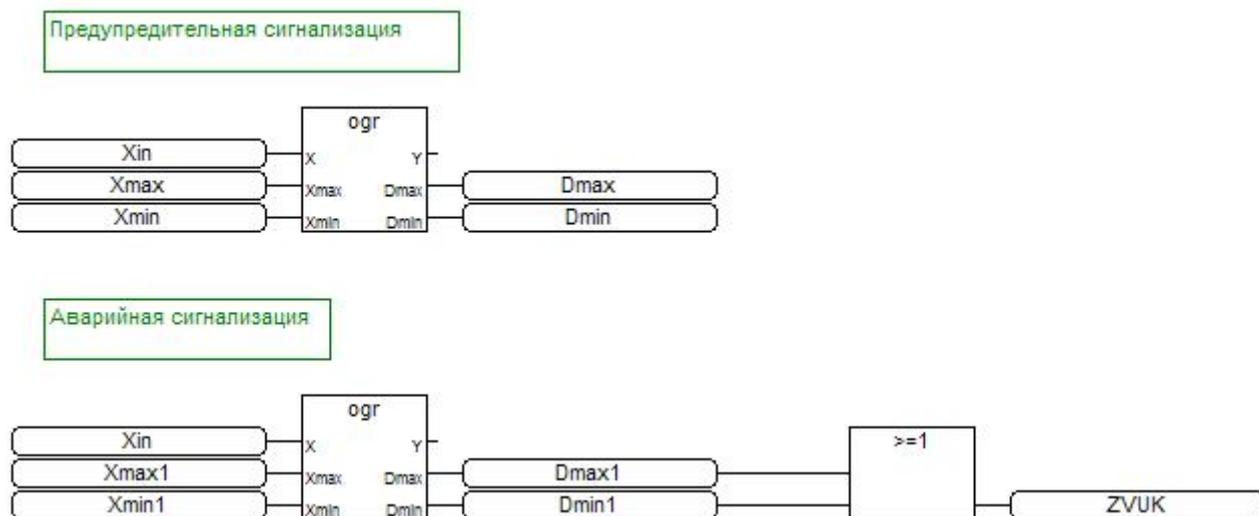


Рисунок 15 – Программа организации предупредительной световой и аварийной звуковой сигнализации на языке FBD

Функциональные блоки программы:

Ogr – обеспечивает формирование сигналов после сравнения с уставками).

>=1 - ИЛИ, обеспечивает формирование звукового сигнала.

3.7 Разработка программы для САР уровня на основе языка программирования ST

Язык программирования ST - это язык высокого уровня, который уже не ограничивается логическим управлением, но с модернизацией оборудования и сложными приложениями в него интегрируются разнообразные функции, такие как управление движением, системы технического зрения, шинная связь, обработка данных, коммуникационные сети, контроль безопасности и регистрация неисправностей. Он может реализовывать такие функции, как выбор, итерация и операторы перехода.

Кроме того, языки структурированного текста легко читать и понимать, особенно если они снабжены значимыми идентификаторами и аннотациями. В сложных системах управления структурированный текст может значительно сократить объем кода и облегчить решение сложных системных задач, однако его недостатком является то, что отладка не является интуитивно понятной, а компиляция выполняется относительно медленно.

3.7.1 Программа управления гидравлическим объектом

Программа управления гидравлическим объектом представлена на рисунке 16.

```
(*Начало программы*)
(*Объявление блоков ввода-вывода*)
(*В словаре в "FB экземпляры" объявляем блок craio и присваиваем ему имя craio111*)
(*Вызываем функциональный блок аналогового ввода-вывода*)
craio111(addr1, phys, AO_1, AO_2, AO_3, AO_4);
AI2:= craio111.AI4;
Xin:= craio111.AI5;

(*В словаре в "FB экземпляры" объявляем блок crdin16 и присваиваем ему имя crdin_16*)
(*Вызываем функциональный блок дискр ввода*)
crdin_16(addr2);
L3:= crdin_16.D2;
L2:= crdin_16.D3;
L1:= crdin_16.D4;

(*В словаре в "FB экземпляры" объявляем блок crdout16 и присваиваем ему имя crdout_16*)
(*Вызываем функциональный дискр вы вода*)
crdout_16(addr3, rez, Control_H3, sign_m, sign_b, Dmax, Dmin, ZVUK, rez, rez, rez, rez, rez, rez, rez, rez, rez);

(*ИМПУЛЬСНЫЙ ПИД-РЕГУЛЯТОР*)
(*В словаре в "FB экземпляры" объявляем блок imp_pid и присваиваем ему имя imp_pid111*)
(*Вызываем функциональный блок imp_pid*)
imp_pid111(zadanie, Km_zdn, Tf_zdn, Csb, Cdb, Vdb, Xin, Km_In, Tf_In, Xdlt, Kp, Ti, Kd, Tm, Cruch);(*описали блок imp_pid*)
Yout:= imp_pid111.Yout; (*выход блока imp_pid*)

(*В словаре в "FB экземпляры" объявляем блок improut и присваиваем ему имя improut111*)
(*Вызываем блок improut*)
improut111(Yout, Tmin, Tlb, Tlm);
Db:= improut111.Db; (*выход блока improut Db*)
Dm:= improut111.Dm; (*выход блока improut Dm*)

(*Описание Ручного режима управления и выходов регулятора*)
a:= Db_Ruch AND Cruch AND NOT b; (*создаём переменную на выходе & (Для Больше) для описания режима ручного управления*)
b:= Dm_Ruch AND Cruch AND NOT a; (*создаём переменную на выходе & (Для Меньше) для описания режима ручного управления*)

sign_b:= (Db_Ruch AND Cruch AND NOT b)OR Db;
sign_m:= (Dm_Ruch AND Cruch AND NOT a)OR Dm;

(*ПРЕДУПРЕДИТЕЛЬНАЯ СИГНАЛИЗАЦИЯ*)
(*Описываем блок org - реализация предупредительной световой сигнализации*)
(*В словаре в "FB экземпляры" объявляем блок ogr и присваиваем ему имя ogr111*)
(*Вызываем блок ogr*)
ogr111(Xin, Xmax, Xmin);

Dmax:= ogr111.Dmax; (*1, если X > Xmax*)
Dmin:= ogr111.Dmin; (*1, если X < Xmin*)

(*АВАРИЙНАЯ СИГНАЛИЗАЦИЯ*)
(*Описываем блок org2 - реализация аварийной звуковой сигнализации*)
(*В словаре в "FB экземпляры" объявляем блок ogr и присваиваем ему имя ogr222*)
(*Вызываем блок ogr*)
ogr222(Xin, Xmax1, Xmin1);

Dmax1:= ogr222.Dmax; (*1, если X > Xmax*)
Dmin1:= ogr222.Dmin; (*1, если X < Xmin*)

(*Создаём блок "ИЛИ" - звуковая сигнализация при достижении MIN и MAX аварийных значений*)
ZVUK:= Dmax1 OR Dmin1;

(*УПРАВЛЕНИЕ РАБОТОЙ НАСОСА Н3*)
(*Создаём блок "Искл ИЛИ"*)
Q_ili:= Rezhim_H3 XOR NOT L1;

(*Реализуем алгоритм работы насоса Н3*)
(*В словаре в "FB экземпляры" объявляем блок tri и присваиваем ему имя tri111*)
(*Вызываем блок ogr*)
tri111(L3, Q_ili);
Dtri:= tri111.D; (*выход блока tri*)

(*Создаём конец проги для управления Н3: блоки "И", "ИЛИ"*)
Control_H3:= (Ruch_H3 AND Rezhim_H3) OR Dtri;

(*Объявляем переменные в словаре*)
(*Конец программы*)
```

Рисунок 16 – Программа управления гидравлическим объектом на языке ST

При создании программы были применены те же блоки, что и в программе на языке FBD. Дополнительно, для реализации ручного управления регулирующим клапаном, срабатывания сигнализации и алгоритма работы насоса НЗ использовались операторы языка OR, XOR, AND и др.

Экземпляр функционального блока должен быть вызван в словаре, где ему присваивается имя.

Имя экземпляра функционального блока позволяет вызвать функциональный блок из библиотеки ISaGRAF или пользовательской библиотеки и передать параметры возврата.

Синтаксис: (*вызвать функциональный блок*)

<blockname> (<p1>, <p2>...);

(*получить возвращаемые параметры*)

<result> := <blockname>. <ret_param1>;

...

<result> := <blockname>. <ret_paramN>;

Пример: создание функционального блока OGR на языке ST.

(*вызов функционального блока ST программой*)

(*объявить экземпляр функционального блока в словаре*)

(*объявить переменные в словаре*)

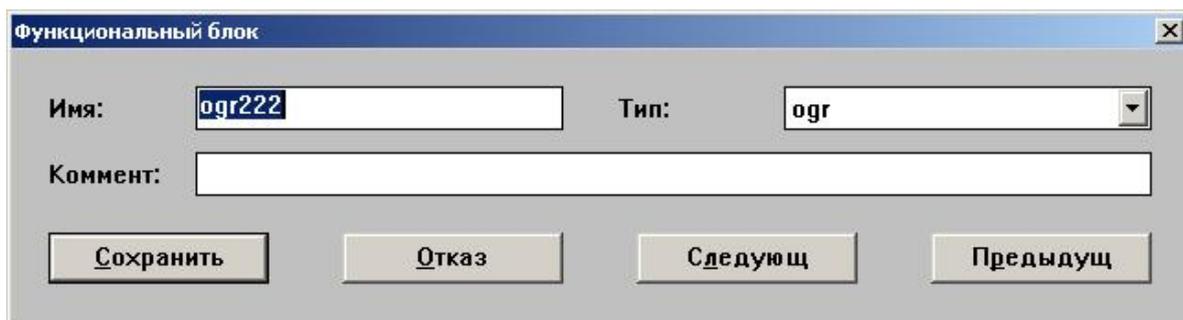


Рисунок 17 – Объявление функционального блока в словаре

(*вызываем функциональный блок из языка ST*)

ogr222(Xin, Xmax1, Xmin1);

(*получаем возвращаемые параметры*)

Dmax1:= ogr222.Dmax; (*1, если X > Xmax*)

$Dmin1 := ogr222.Dmin; (*1, \text{если } X < Xmin*)$

Пример: создание элемента ИЛИ

$ZVUK := Dmax1 \text{ OR } Dmin1;$

Операторы :=, OR выбираются из вспомогательного окна «Ключевые слова».

Переменные данной программы те же, что и для программы на языке FBD.

В результате мы получаем реализацию аварийной сигнализации с возможностью вывода затем переменной ZVUK на блок дискретного вывода.

Описанные блоки (элементы) программы на языке ST эквиваленты схеме на языке FBD, приведённой на рисунке 18.

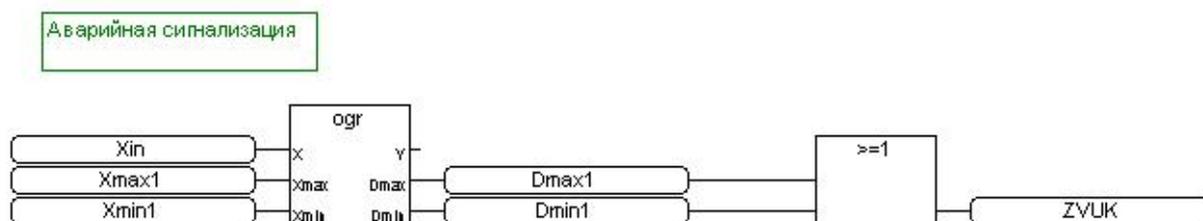


Рисунок 18 – Аналог программы на языке FBD

3.7.2 Организация ввода / вывода аналоговых и дискретных сигналов

Программа ввода / вывода аналоговых и дискретных сигналов представлена на рисунке 19.

```
(*Объявление блоков ввода-вывода*)
(*В словаре в "FB экземпляры" объявляем блок craio и присваиваем ему имя craio111*)
(*Вызываем функциональный блок аналогового ввода-вывода*)
craio111(addr1, phys, AO_1, AO_2, AO_3, AO_4);
AI2:= craio111.AI4;
Xin:= craio111.AI5;

(*В словаре в "FB экземпляры" объявляем блок crdin16 и присваиваем ему имя crdin_16*)
(*Вызываем функциональный блок дискр ввода*)
crdin_16(addr2);
I3:= crdin_16.D2;
I2:= crdin_16.D3;
I1:= crdin_16.D4;

(*В словаре в "FB экземпляры" объявляем блок crdout16 и присваиваем ему имя crdout_16*)
(*Вызываем функциональный дискр вы вода*)
crdout_16(addr3, rez, Control_H3, sign_m, sign_b, Dmax, Dmin, ZVUK, rez, rez, rez, rez, rez,
```

Рисунок 19 - Программа ввода / вывода аналоговых и дискретных

сигналов

Функциональные блоки программы:

Craio – обеспечивает ввод двух аналоговых сигналов: X_{in} – значение уровня жидкости в E2 и AI2 – положение исполнительного механизма в процентах.

Cr_{din}16, Cr_{dout}16 – обеспечивают ввод и вывод, соответственно, дискретных сигналов.

Для контроля и управления гидравлическим объектом задействованы следующие входы (выходы).

Аналоговые:

AI4 – сигнал положения исполнительного механизма;

AI5 – сигнал значения уровня в емкости E2;

Дискретные:

DI2 – уровень L3;

DI3 – уровень L2;

DI4 – уровень L1;

DO2 – сигнал на включение/ отключение насоса НЗ;

DO3 – сигнал «Меньше»;

DO4 – сигнал «Больше»;

DO5 – сигнализация по max;

DO6 – сигнализация по min;

DO7 – звуковая сигнализация.

3.7.2 Программа регулирования уровня в емкости E2 входящая в состав программы управления гидравлическим объектом

Данная подпрограмма должна обеспечивать поддержание уровня жидкости в ёмкости E2 в определённом значении. Регулирование уровня осуществляется изменением расхода на притоке по ПИД закону с использованием функционального блока импульсного ПИД-регулятора.

Также данной программой обеспечивается возможность ручного управления ИМ. При значении переменной Cr_{uch} – True (Истина) значение

сигнала на выходе Y_{out} блока imp_pid равно 0 и с помощью переменных Db_ruch (больше) и Dm_ruch (меньше) можно подавать сигналы управления на выходы ($sign_b$, $sign_m$) блока $crout16$ (Рисунок 20)

```
(*ИМПУЛЬСНЫЙ ПИД-РЕГУЛЯТОР*)
(*В словаре в "FB экземпляры" объявляем блок imp_pid и присваиваем ему имя imp_pid111*)
(*Вызываем функциональный блок imp_pid*)
imp_pid111(zadanie, Km_zdn, Tf_zdn, Csb, Cdb, Vdb, Xin, Km_In, Tf_In, Xdlt, Kp, Ti, Kd, Tm, Cruch);(*описали
Yout:= imp_pid111.Yout; (*выход блока imp_pid*)

(*В словаре в "FB экземпляры" объявляем блок impout и присваиваем ему имя impout111*)
(*Вызываем блок impout*)
impout111(Yout, Tmin, Tlb, Tlm);
Db:= impout111.Db; (*выход блока impout Db*)
Dm:= impout111.Dm; (*выход блока impout Dm*)

(*Описание Ручного режима управления и выходов регулятора*)
a:= Db_Ruch AND Cruch AND NOT b; (*создаём переменную на выходе & (Для Больше) для описания режима ручного уп
b:= Dm_Ruch AND Cruch AND NOT a; (*создаём переменную на выходе & (Для Меньше) для описания режима ручного уп

sign_b:= (Db_Ruch AND Cruch AND NOT b)OR Db;
sign_m:= (Dm_Ruch AND Cruch AND NOT a)OR Dm;
```

Рисунок 20 – Программа регулирования уровня на языке FBD

Функциональные блоки программы:

Imp_pid – реализует импульсный ПИД-регулятор на входе и выходе аналоговый сигнал.

$Impout$ - импульсатор (на входе аналоговый сигнал на выходе импульсы «Б», «М» с заданной скважностью).

AND(И), OR (ИЛИ) – служат для реализации ручного управления ИМ.

3.7.3 Программа управления насосом НЗ входящая в состав программы управления гидравлическим объектом

Для правильной работы гидравлического объекта необходимо обеспечить включение насоса при достижении жидкости верхнего дискретного датчика уровня (L3) в ёмкости ЕЗ, и отключение насоса при откачивании жидкости (отключение нижнего дискретного датчика уровня (L1)

Программа управления насосом реализована с использованием RS-триггера (Рисунок 21).

```

(*УПРАВЛЕНИЕ РАБОТОЙ НАСОСА НЗ*)
(*Создаём блок "Искл ИЛИ"*)
Q_ili:= Rezhim_H3 XOR NOT L1;

(*Реализуем алгоритм работы насоса НЗ*)
(*В словаре в "FB экземпляры" объявляем блок tri и присваиваем ему имя trill1*)
(*Вызываем блок ogr*)
trill1(L3, Q_ili);
Dtri:= trill1.D; (*выход блока tri*)

(*Создаём конец проги для управления НЗ: блоки "И", "ИЛИ"*)
Control_H3:= (Ruch_H3 AND Rezhim_H3) OR Dtri;

```

Рисунок 21 – Программа управления насосом НЗ на языке FBD

Переменная L3 отражает состояние уровня L3 в емкости E3, L1 состояние уровня L1 в емкости E3. Переменная Rezhim_H3 служит для выбора ручного / автоматического режима управления насосом НЗ. Переменная Control_H3 привязана к дискретному выходу контроллера, с которого поступает сигнал на управление насосом.

Функциональные блоки программы:

Tri – Rs триггер предназначен для управления насоса по уровню.

XOR - исключающее OR предназначен для реализации функции режима ручного/ автоматического управления насосом НЗ.

AND (И) – предназначен для возможности включения ручного управления НЗ.

3.7.4 Программа организации световой сигнализации входящая в состав программы управления гидравлическим объектом

Данная подпрограмма обеспечивает формирование дискретных сигналов предупредительной световой и аварийной звуковой сигнализации при превышении или понижении входной величины (X_{in}) установленных значений порогов (X_{max} , X_{min}). Реализовано это с помощью функциональных блоков OGR. Формирование сигналов приведено на рисунке 22.

```

(*ПРЕДУПРЕДИТЕЛЬНАЯ СИГНАЛИЗАЦИЯ*)
(*Описываем блок org - реализация предупредительной световой сигнализации*)
(*В словаре в "FB экземпляры" объявляем блок ogr и присваиваем ему имя ogr111*)
(*Вызываем блок ogr*)
ogr111(Xin, Xmax, Xmin);

Dmax:= ogr111.Dmax; (*1, если X > Xmax*)
Dmin:= ogr111.Dmin; (*1, если X < Xmin*)

(*АВАРИЙНАЯ СИГНАЛИЗАЦИЯ*)
(*Описываем блок org2 - реализация аварийной звуковой сигнализации*)
(*В словаре в "FB экземпляры" объявляем блок ogr и присваиваем ему имя ogr222*)
(*Вызываем блок ogr*)
ogr222(Xin, Xmax1, Xmin1);

Dmax1:= ogr222.Dmax; (*1, если X > Xmax*)
Dmin1:= ogr222.Dmin; (*1, если X < Xmin*)

(*Создаём блок "ИЛИ" - звуковая сигнализация при достижении MIN и MAX аварийных значений*)
ZVUK:= Dmax1 OR Dmin1;

```

Рисунок 22 – Программа организации предупредительной световой и аварийной звуковой сигнализации на языке FBD

Функциональные блоки программы:

Ogr – обеспечивает формирование сигналов после сравнения с уставками).

OR(ИЛИ) - обеспечивает формирование звукового сигнала.

Созданная программа по реализуемым функциям аналогична программе написанной на языке FBD, что позволяет отработать навыки написания программы на языке ST, проверяя результаты работы обеих программ на лабораторном стенде.

3.8 Расчёт параметров настройки регулятора

Расчёт параметров настройки был произведён ранее в выпускной квалификационной работе, посвященной разработке данного лабораторного комплекса.

Структурная схема системы автоматического регулирования уровня приведена на рис 23.

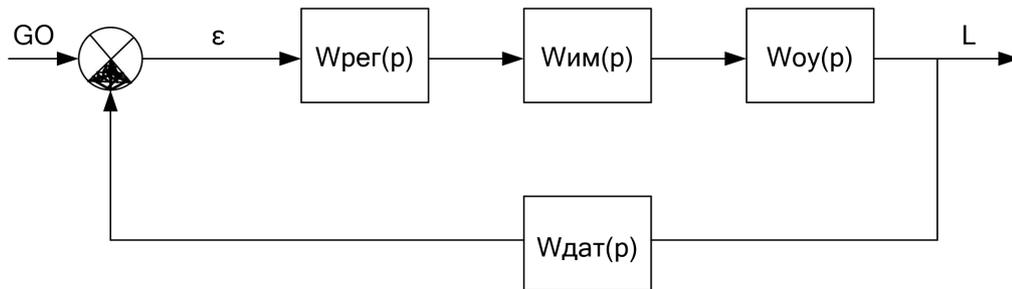


Рисунок 23 – Структурная схема САУ уровня

Передаточные функции отдельных устройств имеют следующий вид.

Передаточная функция регулятора

$$W_p(p) = K_n \frac{T_M}{T_i} \left[1 + T_{ip} \frac{T_i^2 K_{\partial p}}{\left(\frac{1}{8} T_i K_{\partial p} + 1 \right)^2} \right].$$

Передаточная функция исполнительного механизма

$$W_{им}(p) = \frac{K_{им}}{S(S+1)}.$$

Передаточная функция объекта управления

$$W_o(p) = \frac{K_o}{T_o p}.$$

Передаточная функция датчика

$$W_{\partial}(p) = K_{\partial} = 0,39.$$

Коэффициенты передаточных функций были определены ранее в выпускной квалификационной работе, посвященной разработке лабораторной работы, выполненной на базе контроллера ЭЛСИ-Т. Численные значения коэффициентов равны: $K_{им}=0,000337 \text{ рад/\%}$; $K_o=1 \text{ м/(м}^3/\text{с)}$; $K_{\partial}=0,39 \text{ а/м}$; $T_o=1 \text{ с}$; $T_M=120 \text{ с}$.

Модель САУ в пакете MATLAB представлена на рисунке 24.

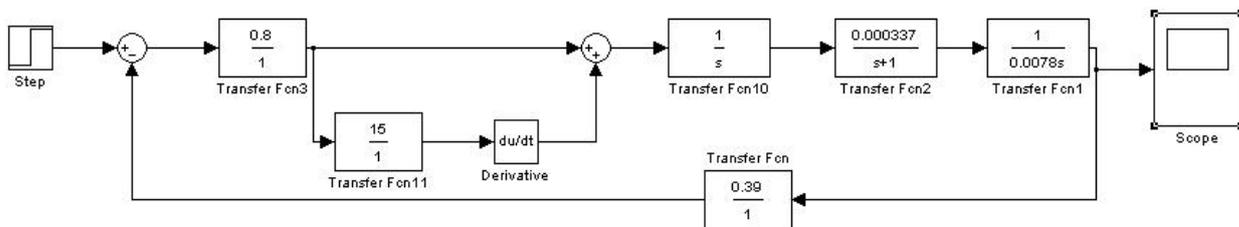


Рисунок 24 – Модель САР.

В результате моделирования были определены коэффициенты регулятора:
 $K_p = 1$; $T_i = 15$.

Кривая переходного процесса САР с данными коэффициентами регулятора приведена на рис.

Показатели качества САР следующие:

- 1) время переходного процесса;
- 2) перерегулирование.

График переходного процесса в САР полученный в результате моделирования САР с данными параметрами в MATLAB представлен на рисунке 25 .

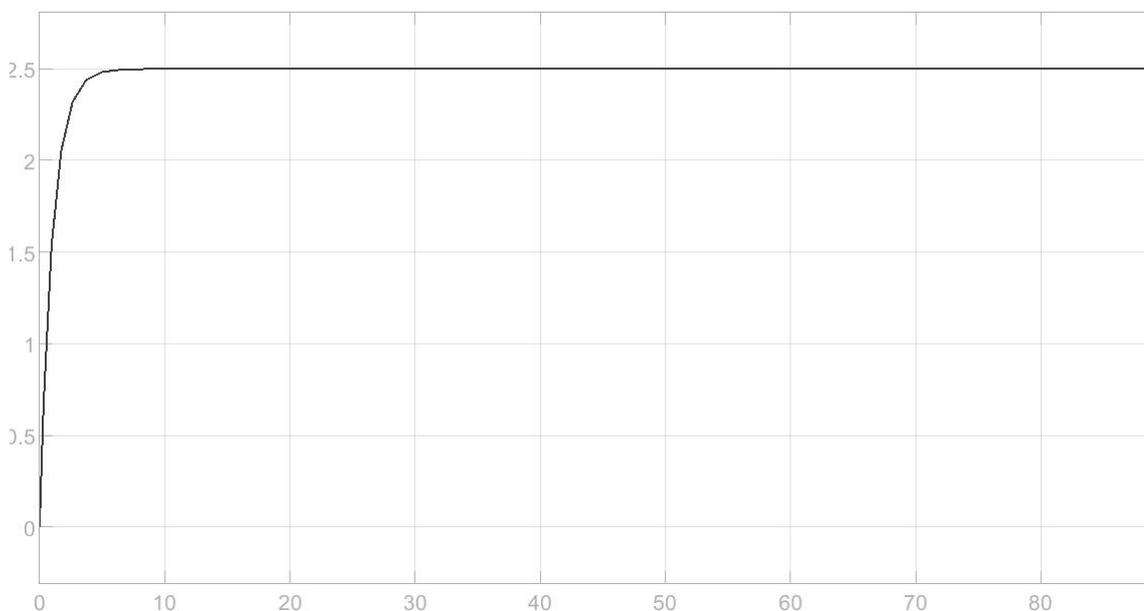


Рисунок 25 – График переходного процесса в САР

4 ПРИМЕНЕНИЕ SCADA-ПАКЕТА MasterSCADA ДЛЯ ВИЗУАЛИЗАЦИИ РАБОТЫ ГИДРАВЛИЧЕСКОГО ОБЪЕКТА

4.1 Назначение SCADA- систем

Система SCADA - это компьютерная система DCS и автоматизированная система мониторинга электроэнергии; она имеет широкий спектр применения и может использоваться во многих областях, таких как сбор данных и контроль мониторинга, а также управление процессами в электроэнергетике, металлургии, нефтяной, химической, газовой и железнодорожной промышленности.

В электроэнергетической системе система SCADA является наиболее широко используемой, а развитие технологии - наиболее зрелым. Она занимает важное место в удаленной системе, которая может контролировать и управлять эксплуатационным оборудованием на месте для реализации функций сбора данных, управления оборудованием, измерения, настройки параметров и всех видов сигнализации, то есть, мы знаем "четыре удаленные" функции.

4.2 SCADA – пакет MasterSCADA 2.2

4.2.1 Основные преимущества пакета

Объект в MasterSCADA - это основная единица разрабатываемой системы, которая имеет особые свойства, позволяющие считать его объектом в понятиях традиционной объектной идеологии. Вот эти свойства:

- Объект обладает набором файлов, трендов и т.д., все они жестко вписаны в него. В разработанной системе каждый файл всегда относится к какому-то объекту, в MasterSCADA не просто тренд, и не просто мнемоника. Эти файлы могут быть какими-то общими файлами, принадлежащими мастер-объекту, или файлами специфического характера, имеющими любой вложенный уровень объектов;

- Ограничивает границы видимости. Использование "чужих" данных в файлах объекта не допускается. Только входы и выходы, принадлежащие данному объекту или подчиненному объекту, считаются "своими" данными.

Например, если вы хотите использовать выход другого объекта в тренде, вы вынуждены создать свой собственный вход, соединить его с "чужим" выходом, а затем вы сможете использовать эти данные, но уже со своего собственного входа. На первый взгляд, это может оказаться неудобным, поскольку необходимо выполнить дополнительное действие. Однако преимущества такого подхода окупятся сторицей при создании большой и сложной системы;

– По умолчанию все настройки унаследованы от "родительского" объекта. Каждый элемент в дереве системы и объектов имеет страницы свойств, которые позволяют настраивать каждый компонент индивидуально. Такое обилие настроек может потребовать большой работы от разработчика системы. Но поскольку большинство настроек объекта одинаковы, это можно сделать только один раз, и все вложенные элементы автоматически воспримут настройки родительского элемента, т.е. "унаследуют" их. Исключением будут только те настройки и только для тех элементов, которые разработчик самостоятельно изменил;

– Допускается использовать один и тот же объект несколько раз со всеми созданными для него файлами, в том числе при разработке нескольких систем. Вы имеете возможность разрабатывать объект с вложенными объектами, документами, файлами вложенных объектов и т.д. При желании вы можете разработать сложный объект. Теперь, когда вы создаете копию объекта, все его настройки, документы, параметры файлов и внутренние ссылки хранятся в копии, и вам больше никогда не придется делать то же самое.

– При необходимости вам нужно будет лишь изменить то, что должно отличаться в копии. Если вы считаете, что объект пригоден для разработки на других системах, вы можете разместить его в библиотеке, и он будет сохранен вместе со всей документацией, настройками и внутренними ссылками. Это обеспечивает возможность независимым разработчикам параллельно создавать объекты для системы.

4.2.2 Пользовательский интерфейс MasterScada

В пользовательском интерфейсе присутствуют различные элементы. Дерево проекта, палитра библиотечных элементов, окно редактирования файлов и производительности и т.д.

Окно проекта включает в себя четыре основные части (рис. 26):

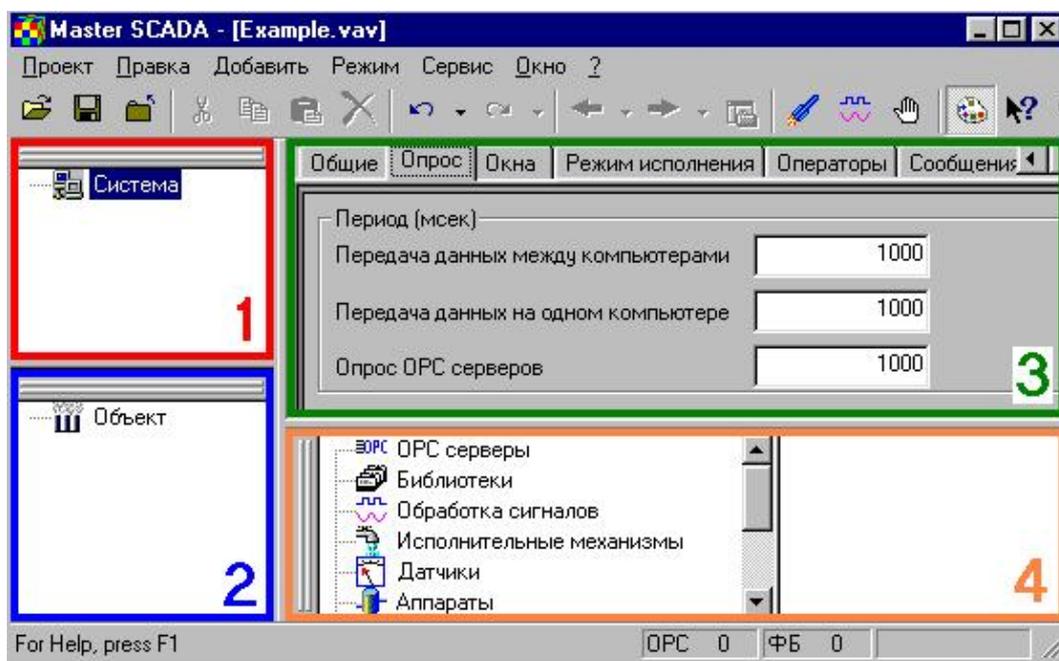


Рисунок 26 - Пользовательский интерфейс MasterScada

- Дерево системы, в котором отображены элементы конфигурации такие, как компьютеры, OPC серверы, и т. д.;
- Дерево объектов, включающее в себя объекты, переменные, группы переменных, функциональные элементы;
- Страницы свойств элементов, на которых производятся все необходимые настройки элементов;
- 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.

4.2.3 Библиотеки

Объектная часть проекта формируется из элементов типа (объектов, групп переменных и переменных) и библиотечных элементов. Существует три типа библиотечных элементов:

- функциональные блоки;
- визуальные функциональные блоки;
- библиотечные объекты (часть проекта, помещенная в библиотеку для повторного использования) - содержит все настройки, файлы и ссылки исходного объекта; внешние ссылки автоматически восстанавливаются при вставке проекта, если доступны необходимые источники и приемники данных.

Простая и наглядная библиотека функциональных блоков содержит более ста стандартных функциональных блоков, которые могут быть расширены путем программирования новых функциональных блоков на основе типовых блоков в соответствии с открытым методом.

Библиотеки объектов формируются путем помещения в библиотеку любой части "объектной" части проекта, включая макеты из функциональных блоков (макроблоки).

Отдельные редакторы (например, вспомогательные) могут располагать собственными библиотеками типовых элементов.

4.2.4 Мнемосхемы

Количество мнемоник в проекте не ограничено.

Количество мнемоник объектов ограничено - объект может иметь все типы мнемоник, разрешенные в проекте. Количество типов может быть расширено пользователем.

Количество элементов мнемоники не ограничено. В качестве мнемоелемента может быть использован любой элемент управления ActiveX, и любое из его свойств может быть динамически настроено, как и одно из стандартных свойств (положение, размер, прорисовка, мигание и т.д.).

Основным способом создания плана памяти является перетаскивание объектов, визуальных функциональных блоков и переменных из дерева проекта,

которые уже имеют все необходимые функции (изображения, динамизация, окна управления и т.д.). Например, для переменной можно выбрать отображение в виде текстового значения или в виде одного из устройств на панели, а объект можно представить в виде кнопки с эскизом ее макета.

Поддерживаются все стандартные графические форматы: bmp, gif, jpg, avi. Все импортируемые изображения и видеоклипы могут отображаться в настраиваемом режиме прозрачности, при этом динамизируя любые другие свойства.

4.2.5 Тренды

Графики могут быть отображены для любого количества архивируемых и неархивируемых переменных. Переменные вставляются в тренд путем перетаскивания из проекта и автоматически наследуют диапазоны и единицы измерения. Тренды реального времени и исторические тренды "бесшовно" объединяются в одном окне. Каждое перо может иметь свой диапазон осей значений. Соотношение времени и значения можно изменять во время просмотра.

4.3 Программа визуализации процесса управления гидравлическим объектом в пакете MasterSCADA

Данная программа позволяет контролировать и управлять технологическим процессом в реальном времени.

Результатом создания программы являются следующие элементы визуализации:

– мнемосхема установки «Гидравлический объект» представленная на рисунке 27 – здесь отражается весь процесс работы установки;

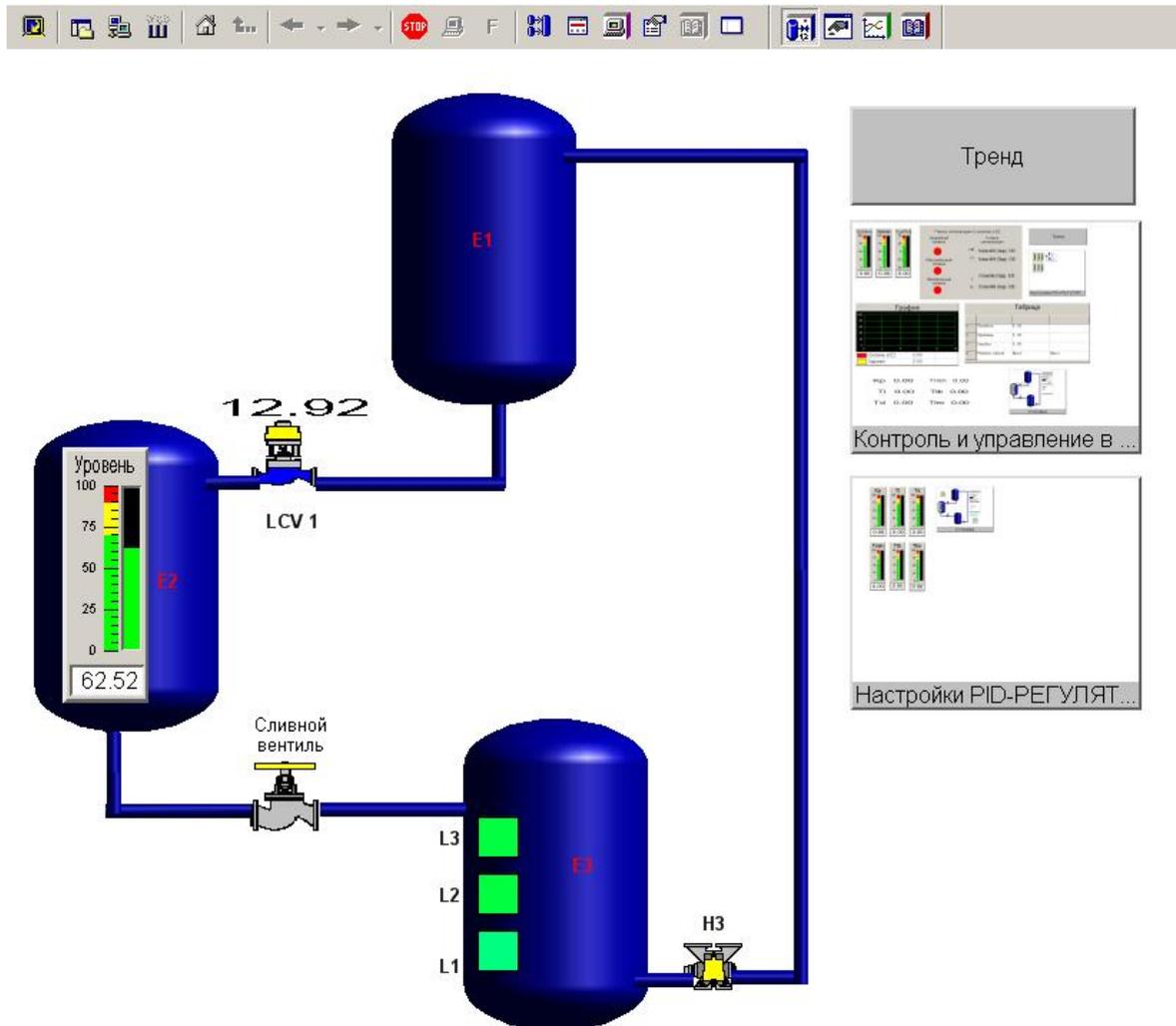


Рисунок 27 – Основная мнемосхема проекта

– отдельное Окно «Контроль и управление в E2» (рисунок 28.) – с помощью этого окна осуществляется контроль параметров и управление ИМ ёмкости E2;

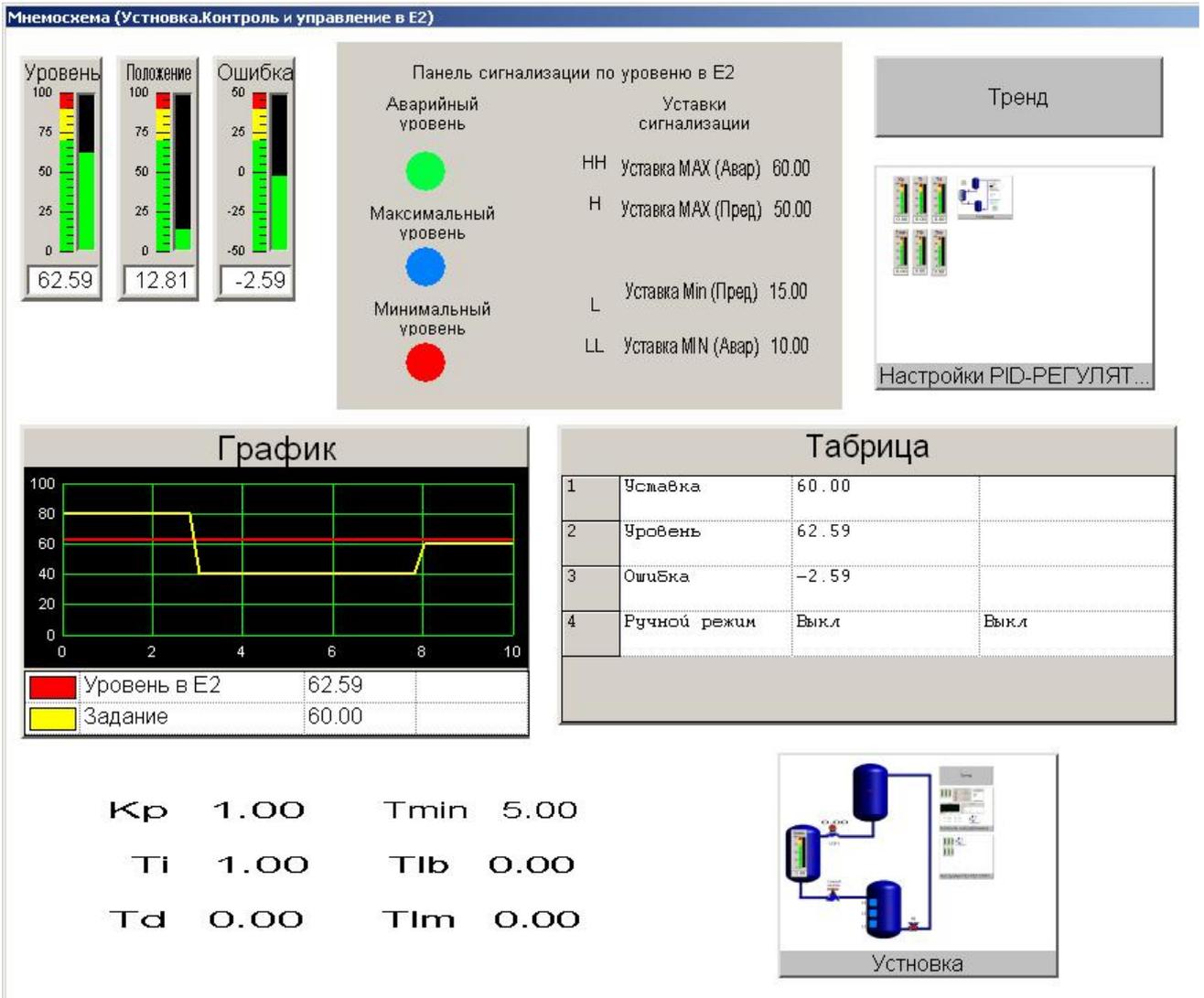


Рисунок 28 – Окно «Контроль и управление в E2»

– окно настроек ПИД-регулятора (рисунок 29) – позволяет в любой момент времени произвести корректировку коэффициентов ПИД-регулятора;

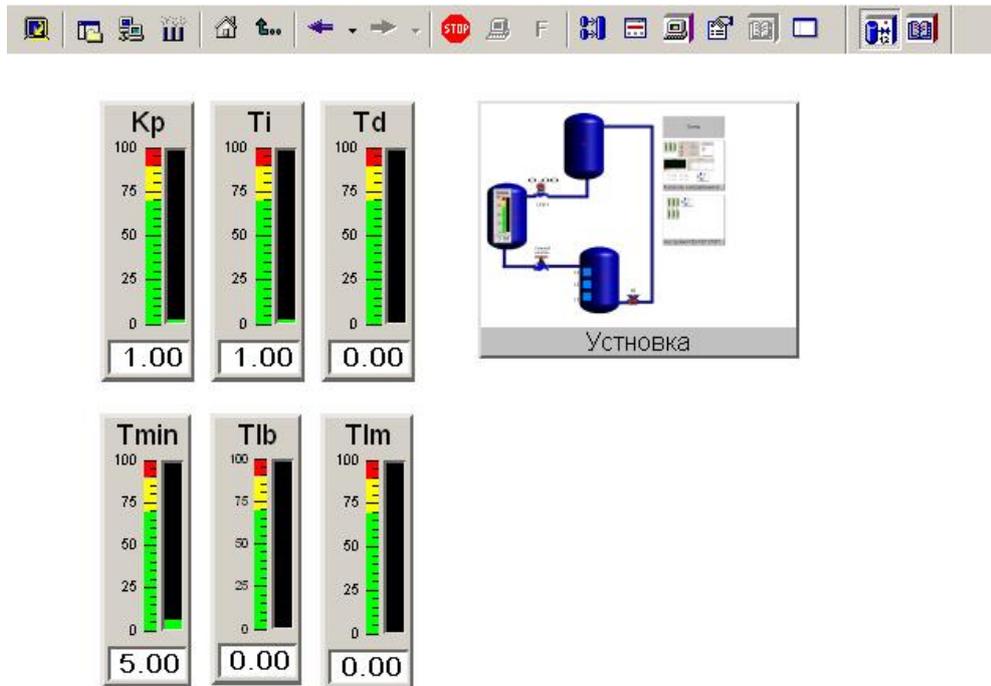


Рисунок 29 – Окно настроек ПИД-регулятора

– тренд (рисунок 30).

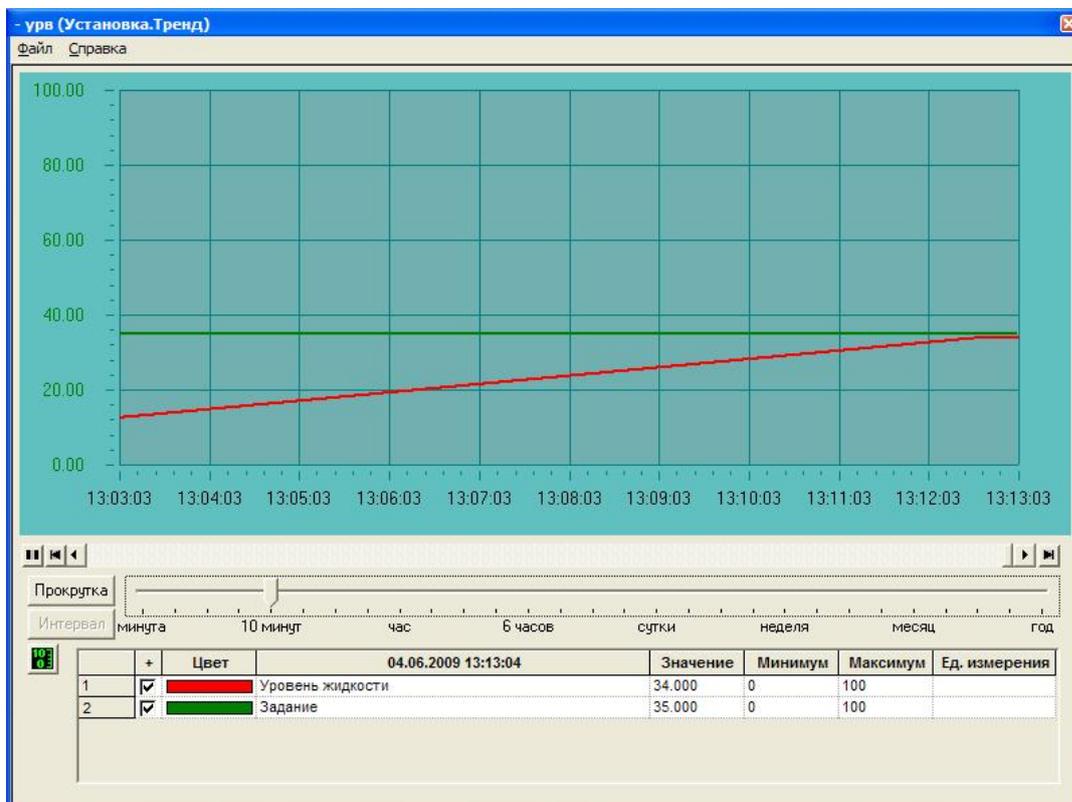


Рисунок 30 – Окно тренда

Также реализована возможность перехода между этими документами.

Мнемосхемы представляют собой набор статических и динамических объектов. К статическим объектам относятся:

- изображение ёмкостей и трубных соединений;
- изображение ручного клапана вводящего в САР возмущающее воздействие;
- изображение надписей.

К динамическим относятся:

- динамический текст, для отображения значений уровня, задания и положения ИМ, задания коэффициентов ПИД-регулятора;
- гистограмма уровня;
- процент заливки по значению уровня в ёмкостях E2 и E3;
- панель ручного управления исполнительным механизмом;
- тренд отображающий значение задания и уровня и др.

После запуска программы отображается основное окно проекта, а так же гистограмма уровня E2, значение уставки, состояния индикаторов уровня в E3 и др. Возможен переход между окнами и вызов тренда.

Управление исполнительными механизмами возможно как из окна основной мнемосхемы, так и, вызвав окно «Контроль и управление в E2».

При выходе значения уровня в емкости E2 за установленные пределы предупредительной сигнализации будут загораться световые индикаторы: меньше 15% – зелёный, больше 50 % – красный. Также будет подан звуковой сигнал при аварийных значениях уровня меньше 10% и больше 60%.

5 МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

5.1 Методические указания по выполнению лабораторной работы «Создание программ управления работой установки «Гидравлический объект» в системе ISaGRAF на основе языков программирования FBD и ST»

Целью работы является получение практических навыков программирования в системе ISaGRAF на примере создания программ для САР уровня на языках программирования FBD и ST, на базе контроллера КРОСС.

Данные методические указания содержат :

Теоретический материал, необходимый для написания программ для двух языков программирования – FBD и ST, а также примеры создания программ на этих языках.

Заданием на выполнение лабораторной работы является составление программы обеспечивающей управление и контроль параметров установки «Гидравлический объект» на заданном преподавателем языке, проверки программы на наличие ошибок и проверки работоспособности программы на лабораторном стенде.

В методических указаниях по выполнению лабораторной работы приводится последовательность выполнения данной работы, необходимый перечень действий.

Проверка правильности выполнения данной лабораторной работы осуществляется преподавателем. В качестве результата выполнения работы являются оформление отчёта и ответы на следующие контрольные вопросы:

1. Приведите краткое описание гидравлического объекта?
2. Можно ли описать блоки ввода-вывода в основной программе?
3. Для чего используется OPC-сервер?
4. Когда используется язык программирования ST?
5. Как описывается функциональный блок на языке ST?

Ответы на данные вопросы позволяют оценить полученные знания, а реализация программы позволяет оценить практические навыки, полученные в процессе выполнения работы.

После проделанной работы студенты оформляют отчет и делают краткие выводы.

5.2 Методические указания по выполнению лабораторной работы «Создание программы визуализации процесса управления установкой «Гидравлический объект» в пакете MasterSCADA»

Целью работы является получение навыков работы в пакете Master SCADA при создании программ визуализации процессов контроля, регулирования и сигнализации.

Данные методические указания содержат :

Теоретический материал, необходимый для создания программы визуализации управления технологическим процессом в пакете Master SCADA, а также примеры создания как отдельных элементов программы и динамизации их свойств, так и в целом пример создания программы визуализации управления лабораторной установкой «Гидравлический объект».

Заданием на выполнение лабораторной работы является создание программы визуализации САР уровня в пакете Master SCADA, обеспечивающей управление и контроль параметров установки «Гидравлический объект». А также проведение отладки и проверки работоспособности программы на лабораторном стенде, используя в качестве аппаратного и программного обеспечения нижнего уровня контроллер КРОСС с исполнительной системой ISaGRAF.

В методических указаниях по выполнению лабораторной работы приводится последовательность выполнения данной работы, необходимый перечень действий.

Проверка правильности выполнения данной лабораторной работы осуществляется преподавателем. В качестве результата выполнения работы являются оформление отчёта и ответы на следующие контрольные вопросы:

1. Назначение SCADA-систем.
2. Перечислите наиболее распространённые на западном и российском рынках SCADA-системы.
3. Перечислите отличительные особенности пакета MasterScada.
4. Для чего используется OPC-сервер?

Ответы на данные вопросы позволяют оценить полученные знания, а реализация программы позволяет оценить практические навыки, полученные в процессе выполнения работы.

После проделанной работы студенты оформляют отчет и делают краткие выводы.

6 КОНСТРУКТОРСКАЯ ЧАСТЬ

В данном разделе приведено описание конструкции лабораторного комплекса "Система автоматического регулирования уровня на базе промышленного контроллера КРОСС", общий вид которого приведён на рисунке 31.

В состав лабораторного комплекса входят следующие устройства.

1. Стенд контроллера, на котором расположены:
 - блок питания контроллера;
 - модули ввода/вывода аналоговых и дискретных сигналов;
 - блок реле;
 - блок сигнализации.
2. ЦБ1.
3. Модуль УСО DIO1 8\8.
4. Модуль УСО AIO1 8\4.
5. Терминальный блок T1-DIO.
6. Терминальный блок T1-AIO.
7. Блок реле.
8. Блок сигнализации.
9. Монитор компьютера.
10. Системный блок компьютера.
11. Стол с выдвигной столешницей для клавиатуры.
12. Гидравлический объект.

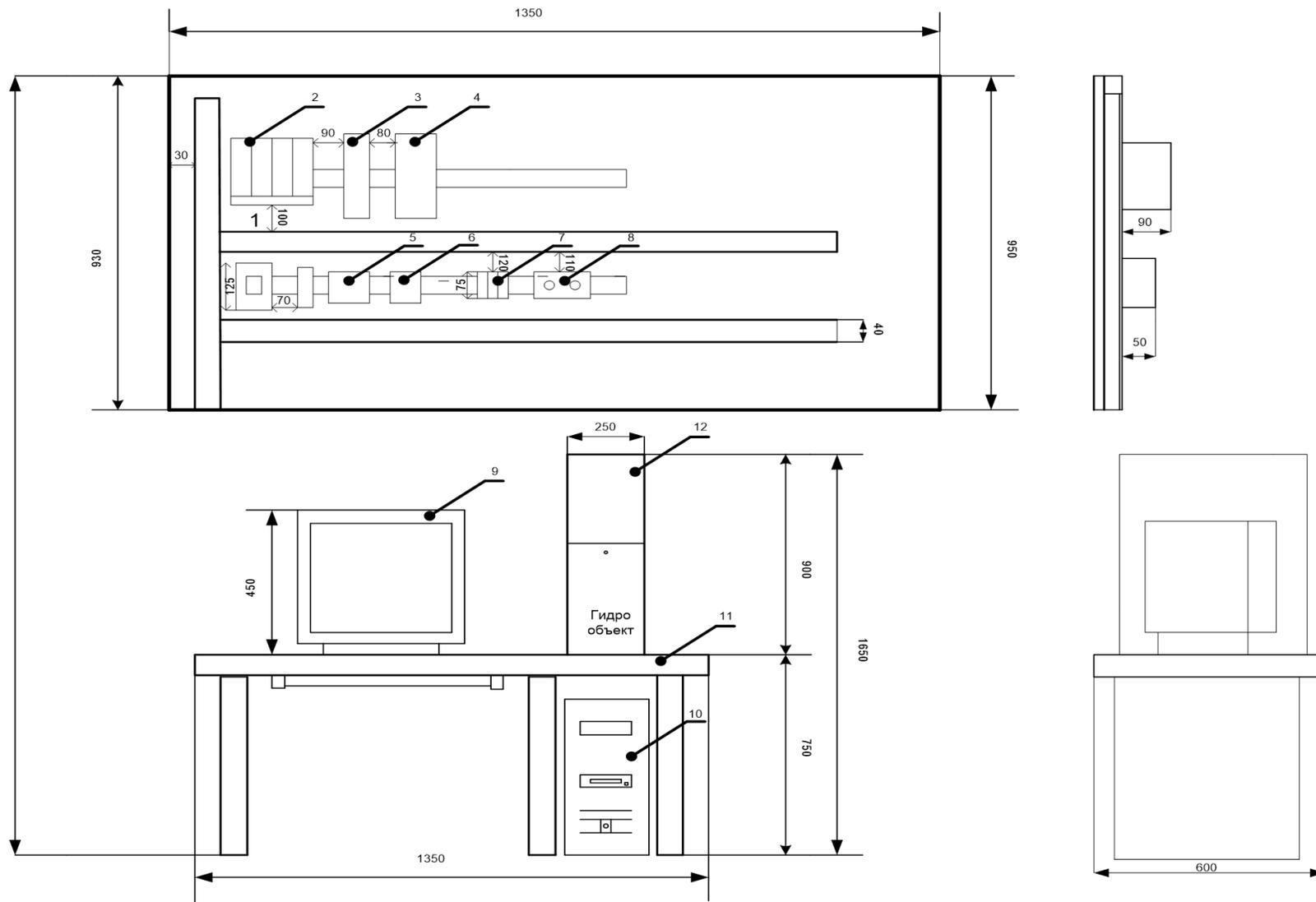


Рисунок 31 – Общий вид лабораторного комплекса

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСООБЪЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа 158Т91		ФИО Ли Яньчэн	
Школа	ИШИТР	Отделение школы (НОЦ)	ОРА
Уровень образования	Бакалавриат	Направление/специальность	15.03.04 Автоматизация технологических процессов и производств

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Стоимость материальных ресурсов определялась по средней рыночной стоимости. Оклады в соответствии с окладами сотрудников организации.
2. Нормы и нормативы расходования ресурсов	30 % районный коэффициент
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	30 % отчисления во внебюджетные фонды

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	Представить возможные альтернативы проведения НИ с позиции ресурсоэффективности и ресурсосбережения.
2. Планирование и формирование бюджета научных исследований	Разработать план научно-исследовательских работ и рассчитать затраты.
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	Определить интегральный показатель эффективности научного исследования

Перечень графического материала (с точным указанием обязательных чертежей):

1. Оценка конкурентоспособности технических решений
2. Матрица SWOT
3. Альтернативы проведения НИ
4. График проведения и бюджет НИ
5. Оценка ресурсной, финансовой и экономической эффективности НИ

Дата выдачи задания для раздела по линейному графику

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШБИП	Якимова Татьяна Борисовна	Кандидат экономических наук		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
158Т91	Ли Яньчэн		

6 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

6.1 планирование научно-исследовательских работ

Так как исполнителей два человека, а именно научный руководитель проекта (НР) и инженера(И), предпочтительным является линейный график. Для его построения хронологически упорядоченные вышеуказанные сведения сведены в таблицу 10:

Таблица 10 - Перечень работ и продолжительность их выполнения

Этапы работы	Исполнители	Загрузка исполнителей
Постановка целей и задач, получение исходных данных	НР	НР – 100%
Составление и утверждение ТЗ	НР, И	НР – 100% И – 10%
Разработка календарного плана	НР, И	НР – 100% И – 10%
Подбор и изучение материалов по тематике	НР, И	НР – 60% И – 100%
Обсуждение литературы	НР, И	НР – 60% И – 100%
Ознакомление с пакетом программирования контроллера Кросс, Isagraf и MasterSCADA	И	И – 100%
Разработка программного обеспечения	НР, И	НР – 30% И – 100%
Разработка методического обеспечения	НР, И	НР – 20% И – 100%
Технико-экономическое обоснование проекта	НР, И	НР – 40% И – 100%
Оценка безопасности и экологичности работы	НР, И	НР – 30% И – 100%
Составление и оформление пояснительной записки	И	И – 100%
Защита дипломного проекта	И	И – 100%

Расчет продолжительности этапов работ осуществляется экспертным способом. Для определения вероятных (ожидаемых) значений продолжительности работ тож применяется формула (1):

$$t_{ож} = \frac{3 * t_{\min} + 2 * t_{\max}}{5} \quad (1)$$

где t_{\min} – минимальная продолжительность работы, дн.; t_{\max} – максимальная продолжительность работы, дн.;

Для построения линейного графика необходимо рассчитать длительность этапов в рабочих днях, а затем перевести ее в календарные дни. Расчет продолжительности выполнения каждого этапа в рабочих днях (ТРД) ведется по формуле (2):

$$T_{РД} = \frac{t_{ож}}{K_{ВН}} * K_{Д} \quad (2)$$

где $t_{ож}$ – продолжительность работы, дн.;

$K_{ВН}$ – коэффициент выполнения работ, учитывающий влияние внешних факторов на соблюдение предварительно определенных длительностей, в частности, возможно $K_{ВН} = 1$;

$K_{Д}$ – коэффициент, учитывающий дополнительное время на компенсацию непредвиденных задержек и согласование работ ($K_{Д} = 1-1,2$; в этих границах конкретное значение принимает сам исполнитель). Берем коэффициенты, равные: $K_{ВН} = 1$; $K_{Д} = 1,1$. Расчет продолжительности этапа в календарных днях ведется по формуле (3):

$$T_{КД} = T_{РД} * T_{К} \quad (3)$$

где $T_{КД}$ – продолжительность выполнения этапа в календарных днях;

$T_{К}$ – коэффициент календарности, позволяющий перейти от длительности работ в рабочих днях к их аналогам в календарных днях, и рассчитываемый по формуле (4):

$$T_{К} = \frac{T_{КАЛ}}{T_{КАЛ} - T_{ВД} - T_{ПД}} \quad (4)$$

где $T_{КАЛ}$ – календарные дни ($T_{КАЛ} = 366$); $T_{ВД}$ – выходные дни ($T_{ВД} = 52$); $T_{ПД}$ – праздничные дни (Включает зимние и летние каникулы) ($T_{ПД} = 66$).

В нашем случае расчет ведем для шестидневной рабочей недели,
 $T_k=1.48$.

Таблица 11 - Трудозатраты на выполнение проекта:

Этап	Исполнители	Продолжительность работ, дни			Трудоёмкость работ по исполнителям чел.-дн.			
		t_{\min}	t_{\max}	$t_{\text{ож}}$	НР($T_{pд}$)	И	НР($T_{кд}$)	И
Постановка целей и задач, получение исходных данных	НР	1	4	2,2	2,24	0,00	3,32	0,00
Составление и утверждение ТЗ	НР, И	1	2	1,4	1,54	0,15	2,28	0,20
Разработка календарного плана	НР, И	1	2	1,4	1,54	0,15	2,28	0,20
Подбор и изучение материалов по тематике	НР, И	3	8	5	3,3	5,5	4,88	8,14
Обсуждение литературы	НР, И	1	2	1,4	0,92	1,54	1,37	2,28
Ознакомление с пакетом программирования контроллера Кросс ,Isagraf И MasterSCADA	И	8	12	9,6	0,00	10,56	0,00	15,63
Разработка программного обеспечения	НР, И	10	15	12	3,96	13,2	5,86	19,54
Разработка методического обеспечения	НР, И	5	7	5,8	1,28	6,38	1,89	9,44
Технико-экономическое	НР, И	2	4	2,8	1,23	3,08	1,82	4,56
Оценка безопасности и экологичности работы	НР, И	2	4	2,8	0,92	3,08	1,37	4,56
Составление и оформление пояснительной записки	И	6	10	7,6	0,00	8,36	0,00	12,37
Защита дипломного проекта	И	1	2	1,4	0,00	1,54	0,00	2,28
Итого:					16,93	53,54	25,06	79,24

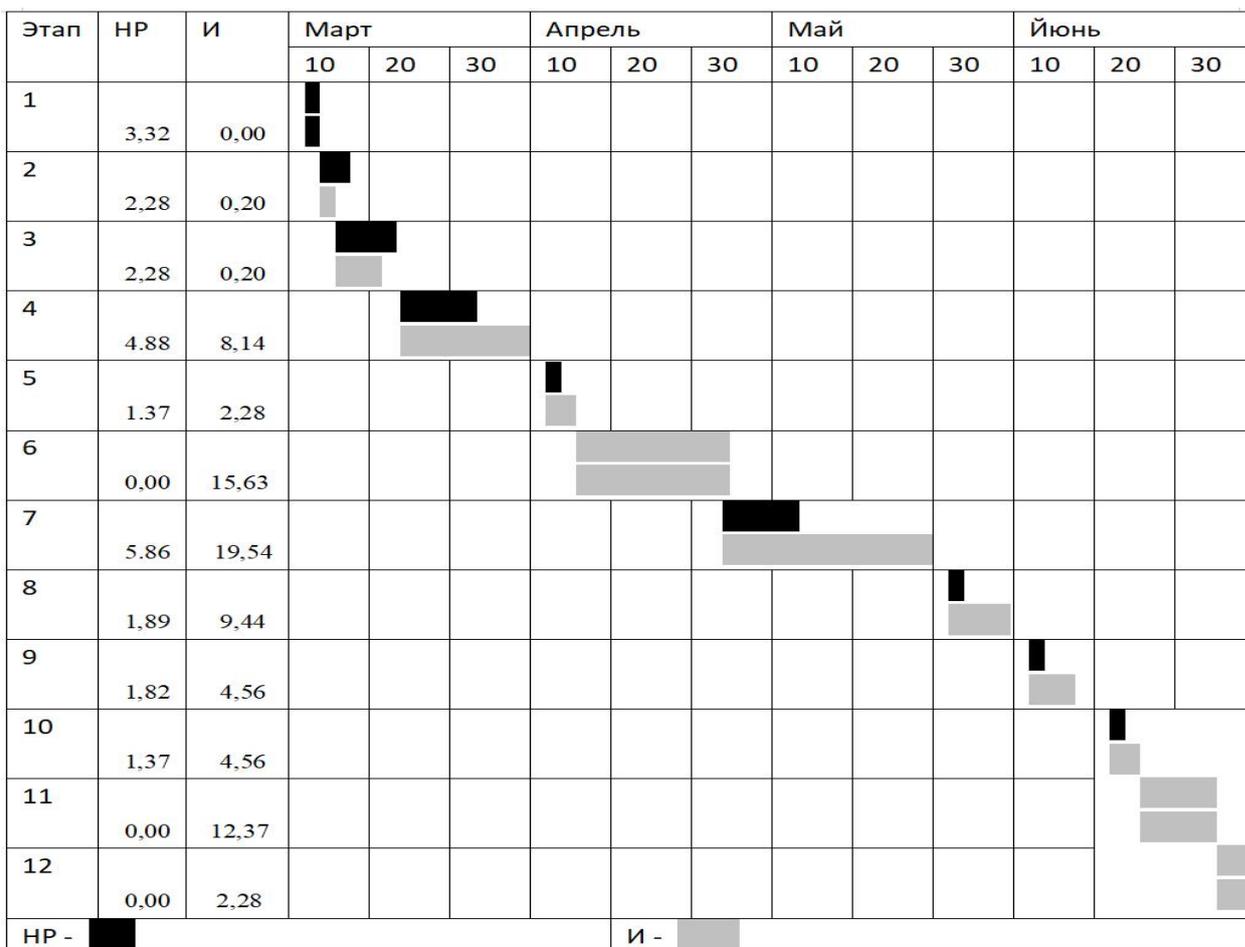


Рисунок 32 - Линейный график работ

6.2 Бюджет научно-технического исследования

В таблице 11 приведены расчеты затрат на материалы.

Таблица 11 - Расчет затрат на материалы

Наименование материалов	Цена за ед., руб.	Кол-во	Сумма, руб.
USB Flash Drive 4 Gb	300	1	300
Ручка	8	1	8
Карандаш	5	1	5
Тетрадь	10	2	20
Бумага для принтера формата А4	650	1	650
Картридж для принтера	2350	1	2350
Итого:			3333

Расчет затраты на электроэнергию для технологических целей приведен в таблице 12.

Таблица 12- Затраты на электроэнергию технологическую

Наименование оборудования	Время работы оборудования $t_{об}$, час	Потребляемая мощность $P_{об}$, кВт	Затраты $Э_{об}$, руб.
Персональный компьютер	300	0,5	788,54
Принтер	3	0,3	4,73
Контроллер КРОСС	350	0,2	367,98
Итого:			1 161,25

В статье «Амортизационные отчисления» рассчитывается амортизация используемого оборудования за время выполнения проекта.

Используется формула:

$$C_{AM} = \frac{N_A * Ц_{об} * t_{рф} * n}{F_D}$$

где N_A – годовая норма амортизации единицы оборудования;

$Ц_{об}$ – балансовая стоимость единицы оборудования с учетом ТЗР.

F_D – действительный годовой фонд времени работы соответствующего оборудования.

Для ПК (250 рабочих дней при шестидневной рабочей неделе) можно принять $F_D = 250 * 8 = 2000$ часа;

Для принтера $F_D = 500$ часов;

Для контроллера КРОСС $F_D = 2000$ часов;

$t_{рф}$ – фактическое время работы оборудования в ходе выполнения проекта, учитывается исполнителем проекта.

Для ПК $trф = 300$ часа. Для
принтера $trф = 3$ часов.

Для контроллера КРОСС ФД $trф = 350$ часа.

n – число задействованных однотипных единиц оборудования.

При использовании нескольких типов оборудования расчет по формуле делается соответствующее число раз, затем результаты суммируются.

Для определения НА следует обратиться к постановлению правительства РФ «О классификации основных средств, включенных в амортизационные группы». Оно позволяет получить рамочные значения сроков амортизации (полезного использования) оборудования СА.

Для ПК возьмем СА= 2,5 года. Далее определяется НА как величина обратная СА, в данном случае это $1/2,5 = 0,4$.

Для принтера СА= 2 года, НА= 0,5.

Для контроллера КРОСС СА= 2,5 года НА= 0,4;

$САМ(ПК) = (0,4 * 70000 * 300 * 1) / 2000 = 4200$ руб;

$САМ(Пр) = (0,5 * 25000 * 3 * 1) / 500 = 75$ руб;

$САМ(К) = (0,4 * 45000 * 350) / 2000 = 3150$ руб.

Итого начислено амортизации 7425 руб.

Расчет основной заработной платы выполняется на основе трудоемкости выполнения каждого этапа и величины месячного оклада исполнителя. Среднедневная тарифная заработная плата (ЗПдн-т) рассчитывается по формуле:

$$ЗПдн-т = MO/24,$$

учитывающей, что в году 250 рабочих дней и, следовательно, в месяце в среднем 24 рабочих дня (при шестидневной рабочей неделе).

Расчеты затрат на полную заработную плату приведены в таблице 12 Затраты времени по каждому исполнителю в рабочих днях с округлением до целого взяты из таблицы 11. Для учета в ее составе премий, дополнительной зарплаты и районной надбавки используется следующий ряд коэффициентов:

$$КПР = 1,1; Кдоп.ЗП = 1,188; Кр = 1,3.$$

Таким образом, для перехода от тарифной (базовой) суммы заработка исполнителя, связанной с участием в проекте, к соответствующему полному заработку (зарплатной части сметы) необходимо первую умножить на интегральный коэффициент $K_{и} = 1,1 * 1,188 * 1,3 = 1,699$. Вышеуказанное значение $K_{доп.ЗП}$ применяется при шестидневной рабочей неделе.

Таблица 13 - Затраты на заработную плату

Исполнитель	Оклад, руб./мес.	Среднедневная ставка, руб./раб.день	Затраты времени, раб.дни	Коэффициент	Фондз/плат руб.
НР	25000	1041,67	20	1,6 99	35395,95
И	12000	500,00	80	1,6 99	67960,00
Итого:					103355,95

$$\text{Отчисления во внебюджетные фонды} = 103355,95 * 0,3 = 31006,79 \text{ руб.}$$

В статье «Прочие расходы» отражены расходы на выполнение проекта, которые не учтены в предыдущих статьях, их следует принять равными 10% от суммы всех предыдущих расходов, т.е.

Спроч. = (Смат + Сзп + Ссоц + Сэл.об. + Сам + Снп) · 0,1. Для нашего примера это

Спроч. = $(3333,00 + 103355,95 + 31006,79 + 1\ 161,25 + 7425,00 + 0,00) \cdot 0,1 = 14628,2$ руб.

Проведя расчет по всем статьям сметы затрат на разработку, можно определить общую себестоимость проекта «Программно-методическое обеспечение для изучения цифровых регуляторов, построенных на нечеткой логике».

Таблица 14 Смета затрат на разработку проекта

Статья затрат	Условное обозначение	Сумма, руб.
Материалы и покупные изделия	$C_{\text{мат}}$	3333,00
Основная заработная плата	$C_{\text{зп}}$	103355,95
Отчисления в социальные фонды	$C_{\text{соц}}$	31006,79
Расходы на электроэнергию	$C_{\text{эл.}}$	1 161,25
Амортизационные отчисления	$C_{\text{ам}}$	7425,00
Непосредственно учитываемые расходы	$C_{\text{нр}}$	0,00
Прочие расходы	$C_{\text{проч}}$	14628,2
Итого:		160910,2

6.3 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Определение эффективности происходит на основе расчета интегрального показателя эффективности научного исследования. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

Интегральный показатель финансовой эффективности научного исследования получают в ходе оценки бюджета затрат исполнения научного

исследования. Интегральный финансовый показатель разработки определяется к

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}}$$

где $I_{\text{финр}}^{\text{исп.}i}$ – интегральный финансовый показатель разработки;

Φ_{pi} – стоимость исполнения;

Φ_{max} – максимальная стоимость исполнения научно-исследовательского проект.

$$I_{\text{финр-текущая-разработка}} = \frac{160910,2}{160000,0} = 1,0057$$

$$I_{\text{финр-вариант-исполнения}} = 1$$

Полученная величина интегрального финансового показателя разработки отражает соответствующее численное увеличение бюджета затрат разработки в размах (значение больше единицы), либо соответствующее численное удешевление стоимости разработки в размах (значение меньше единицы, но больше нуля).

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_p = \sum a * b$$

где I_p – интегральный показатель ресурсоэффективности;

a – весовой коэффициент;

b – балльная оценка, устанавливается экспертным путем по выбранной шкале оценивания;

Таблица 15 – Сравнительная оценка характеристик проекта

Критерии	Весовой коэффициент параметра	Оценка	
		Текущий проект	Вариант исполнения 1
1. Уровень новизны	0,4	4	2
2. Теоретический уровень	0,1	2	2
3. Возможность реализации	0,5	4	4
Итого	1	3,8	3

Интегральный показатель эффективности определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{\text{текущая-разработка}} = \frac{I_P}{I_{\text{финр}}} = 3,75$$

$$I_{\text{вариант-исполнения}} = \frac{I_P}{I_{\text{финр}}} = 3$$

Сравнительная эффективность проекта:

$$\mathcal{E}_{\text{ср}} = \frac{I_{\text{фа}}^p}{I_{\text{фб}}^p}$$

где $\mathcal{E}_{\text{ср}}$ – сравнительная эффективность проекта;

$I_{\text{фа}}^p$ – интегральный показатель разработки;

$I_{\text{фб}}^p$ – интегральный технико-экономический показатель аналога

$$\mathcal{E}_{\text{ср1}} = \frac{3,75}{3} = 1,25$$

Отразим результаты расчета сравнительной эффективности разработки в таблице 16.

Таблица 16 – Сравнительная эффективность разработки

№ п/п	Показатели	Разработка	Исп1
1	Интегральный финансовый показатель разработки	1,06	1,06
2	Интегральный показатель ресурсоэффективности разработки	3,8	3
3	Интегральный показатель эффективности	3,58	2,83
4	Сравнительная эффективность вариантов исполнения	1,25	

Оценивая бюджет исследования и определяя консолидированные финансовые показатели, можно сделать вывод, что рассчитанные финансовые показатели и консолидированные показатели эффективности показали, что проект является эффективным.

Поэтому проект не ориентирован на экономический результат, а направлен на обеспечение дополнительной программно-методической поддержки студентов по направлению 15.03.04.

6.4 Вывод

В ходе выполнения работы было проведено экономическое обоснование к разработке лабораторного комплекса на базе промышленного микропроцессорного контроллера КРОСС.

Затем была рассчитана смета затрат на разработку данной системы и смета затрат на ее годовую эксплуатацию. Были просчитаны материальные затраты, используемые при разработке проекта которые составляют 3333 рублей. Рассчитана основная заработная плата инженера программиста и руководителя которая составляет 103355,95 руб. Отчисления во внебюджетные фонды 31006,79 руб. Была просчитана амортизация основных фондов 7425 руб. Прочие расходы составляют 14628,2 руб. Расходы на

электроэнергию составляют 1 161,25 руб. Затраты на реализацию проекта составляют 160910,2 руб.

Данный лабораторный комплекс позволяет:

- получить навыки по настройке регуляторов;
- позволяет изучить основы программирования промышленного микропроцессорного контроллера «КРОСС»;
- помогает подробнее изучить систему программирования ISaGRAF;
- получить навыки по созданию мнемосхем объектов управления в MasterSCADA.

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
158T91	Ли Яньчэн

ШКОЛА	ИШИТР	Отделение	ОРА
Уровень образования	Бакалавриат	Направление /специальность	15.03.04 Автоматизация технологических процессов и производств

Тема дипломной работы: «Система автоматического регулирования уровня на базе контроллера КРОСС»

Исходные данные к разделу «Социальная ответственность»:

- Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения)

Правовые и организационные вопросы обеспечения безопасности

- специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства;
- организационные мероприятия при компоновке рабочей зоны.

Объект исследования – КРОСС-500, ISaGRAF и MasterSCADA
Область применения – системы автоматизации технологических объектов широкого класса (простых и сложных, медленных и быстрых, сосредоточенных и распределенных в пространстве) в различных отраслях с непрерывными и дискретными технологическими процессами (энергетические, химические, нефте- и газодобывающие, машиностроительные, сельскохозяйственные, пищевые производства, производство стройматериалов, предприятия коммунального хозяйства т.п.).

Приводится перечень НТД, используемой в данном разделе.
1, Белов, Сергей Викторович. Безопасность жизнедеятельности и защита окружающей среды (техносферная безопасность) [Электронный ресурс]: учебник для бакалавров / С. В. Белов. — 4-е изд. — Мультимедиа ресурсы.
2, Безопасность жизнедеятельности. Безопасность технологических процессов и производств. Охрана труда: учебное пособие для вузов / П. П. Кукин.
3, Лесенко, Георгий Георгиевич. Безопасность труда в приборостроении / Г. Г. Лесенко, Ю. И. Борисенко. — Киев: Техника, 1988. — 128 с.: ил. — Техника безопасности. — Библиогр.: с. 127.
4, Правовое обеспечение информационной безопасности [Текст] : учеб. пособие для вузов / под ред. С. Я. Казанцева.- 2-е изд., испр. и доп. - М. : Академия, 2007. - 240 с.

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Производственная безопасность Анализ показателей шума и вибрации

- установление соответствие показателей нормативному требованию;

Анализ показателей микроклимата

- показатели температурные, скорости движения воздуха, запыленности.

Коллективные средства защиты:

- Защитный колпак: для фильтрации и изоляции токсичных газов.
- Замок: для защиты лабораторных помещений.

<p>Анализ освещенности рабочей зоны</p> <ul style="list-style-type: none"> - типы ламп, их количество, соответствие нормативному требованию освещенности; - при расчете освещения указать схему размещения светильников на потолке согласно проведенному расчету. <p>Анализ электробезопасности</p> <ul style="list-style-type: none"> - наличие электроисточников, характер их опасности; - установление класса электроопасности помещения, а также безопасные номиналы тока, напряжения, сопротивления заземления. - при расчете заземления указать схему размещения заземлителя согласно проведенному расчету. <p>Анализ пожарной безопасности</p> <ul style="list-style-type: none"> - присутствие горючих материалов, тем самым, присутствие повышенной степени пожароопасности. - категории пожароопасности помещения, марки огнетушителей, их назначение. - Разработать схему эвакуации при пожаре. 	<ul style="list-style-type: none"> - Устройство заземления: для защиты от повышенного статического электричества. <p>Индивидуальные средства защиты:</p> <ul style="list-style-type: none"> - Средства защиты органов дыхания (респираторы, противогазы и противопылевые маски). - Защита рук (перчатки, щитки для запястий). - Средства защиты головы (каска, капюшоны). - Средства защиты кожи (защитная одежда, резиновые сапоги и т.д.). - Медицинские средства защиты (личная аптечка). <p>Для сетей с напряжением 220 В сопротивление изоляции должно быть не менее 380 кОм. Напряжение постоянного тока. Входное напряжение от минус 5 до 65 мВ.</p> <p>Категорию пожароопасности помещения: взрывопожароопасность (Б).</p> <p>Марки огнетушителей, их назначение:</p> <ul style="list-style-type: none"> - Воздушно-пенный огнетушитель: огнетушитель, заряд и конструкция которого обеспечивают получение и применение воздушно-механической пены низкой или средней кратности для тушения пожаров. - Воздушно-пенный огнетушитель: огнетушитель, заряд и конструкция которого обеспечивают получение и применение воздушно-механической пены низкой или средней кратности для тушения пожаров.
<p>2. Экологическая безопасность:</p> <ul style="list-style-type: none"> - защита селитебной зоны - анализ воздействия объекта на атмосферу (выбросы); - анализ воздействия объекта на гидросферу (сбросы); - анализ воздействия объекта на литосферу (отходы); - разработать решения по обеспечению экологической безопасности со ссылками на НТД по охране окружающей среды. 	<ul style="list-style-type: none"> - защита селитебной зоны: отсутствует. - анализ воздействия объекта на атмосферу (выбросы): Картридж принтера при работе выделяет небольшое количество пыли. Рассмотрите возможность сокращения использования. - анализ воздействия объекта на гидросферу (сбросы): продукты жизнедеятельности персонала. - анализ воздействия объекта на литосферу

	<p>(отходы): Отходы, образующиеся при поломке компьютеров, ламп подсветки дисплеев и других электроприборов. Также следует рассмотреть возможность утилизации оборудования.</p> <ul style="list-style-type: none"> - Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 25.02.2022). - ГОСТ 12.1.004-91 ССБТ. Пожарная безопасность. Общие требования. - ГОСТ 12.1.030-81 ССБТ. Электробезопасность. Защитное заземление. Зануление. - СП 52.13330.2016 Естественное и искусственное освещение.
<p>3. Безопасность в чрезвычайных ситуациях:</p> <ul style="list-style-type: none"> - перечень возможных ЧС при разработке и эксплуатации проектируемого решения; - выбор наиболее типичной ЧС; - разработка превентивных мер по предупреждению ЧС; - разработка действий в результате возникшей ЧС и мер по ликвидации её последствий. 	<p>Рассматриваются 2 ситуации ЧС:</p> <ol style="list-style-type: none"> 1) природная – сильные морозы зимой; 2) техногенная – исключить несанкционированное проникновение посторонних на рабочее место (большая вероятность проведения диверсии). <p>Предотвращение возможных технических аварийных ситуаций в присутствии только испытателей, обеспечивая при этом нормальную температуру в помещении: пожар (возгорание) регламентировано ГОСТ Р 22.0.02-94 Наиболее типичная ЧС_пожар (возгорание) регламентировано ГОСТ Р 22.0.02-94.</p>

Дата выдачи задания для раздела по линейному графику	26.04.2023 г.
--	---------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ТПУ	Сечин Александр Иванович	Доктор технических наук		30.05.2023 г.

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
158Т91	Ли Яньчэн		30.05.2023 г.

7 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

Целью данной работы было изучение использования КРОСС-500 и реализация системы автоматического регулирования уровня путем программирования программного пакета ISaGRAF и моделирования с помощью программного пакета MasterSCADA. Разработка и прохождение квалификационной работы в Томском политехническом университете, 10 этаж, комната 113. Был написан соответствующий программный код и проведены эксперименты на кросс-контроллере, чтобы убедиться в правильности работы программного кода. Для этого также было подготовлено руководство по использованию программного пакета КРОСС-500.

В этой работе был разработан комплекс организационных, правовых, технических и институциональных мер по минимизации негативного воздействия длительного рабочего времени на компьютеры, а также рассмотрены вопросы техники безопасности, экологической и пожарной безопасности и даны рекомендации по созданию оптимальных условий труда.

Негативного социального или экологического воздействия, связанного с данным проектом, нет, однако в ходе работы по программированию могут образовываться твердые отходы, такие как бумага, лампочки, пищевые отходы и отходы личной гигиены, а также отходы от канцелярских принадлежностей.

7.1 Правовые и организационные вопросы обеспечения безопасности

Работа в опасных и вредных условиях труда, или в работа в условиях исключительной температуры или загрязнения выдача средств защиты с обязательной сертификацией или декларацией соответствия средства защиты с обязательной сертификацией или декларацией о соответствии выдаются в соответствии со следующими критериями:

Стандарты на средства индивидуальной защиты (СИЗ) устанавливаются Правительством Российской Федерации.

Поскольку данный вид работ предполагает возможные риски для здоровья и безопасности, необходимо обеспечить работников всеми необходимыми средствами защиты - рабочими перчатками для уменьшения травм от острых металлических кромок. Работа в помещении должна быть комфортной и хорошо освещенной.

Перед станком должно быть ровное и удобное место, а освещение должно быть достаточным для работы.

В помещении должно быть как естественное, так и искусственное освещение.

Внутренняя отделка помещения должна быть выполнена из диффузных отражающих материалов диффузные отражающие материалы с коэффициентом отражения 0,7 - 0,8 для потолков; 0,5 - 0,6 для стен; 0,3 - 0,5 для полов.

В помещении должна быть аптечка первой помощи. Взрывоопасные и легковоспламеняющиеся материалы, они должны быть покрыты огнезащитными материалами.

Рабочее место располагается на 1 этаже в Томском политехническом университете, корпусе 10, комнате 113. помещение представляет собой комнату размером 10 м на 5 м, высотой 3 м, 2 окон выходящих на запад, в помещении находится 8 единиц технологического оборудования.

7.2 Производственная безопасность

В заключительной работе для идентификации, основанной на контроллере КРОСС, мы исследовали разработку программного обеспечения и его использование, использовали для исследования программный пакет ISaGRAF.

Исходя из этого отсутствия опасности, был использован программный пакет для контроллера КРОСС -500. Поэтому в наших экспериментах

необходимо использовать следующее оборудование: компьютер и промышленный контроллер КРОСС-500.

Исследование данного проекта подразумевает написание программ, их тестирование и разработку учебных пособий, готовые экспериментальные пособия будут использоваться для обучения новых студентов.

Для выбора факторов использовался ГОСТ 12.0.003-2015. Перечень опасных и вредных факторов, присутствующих в производственной среде экспериментального проекта, представлен в виде таблицы 17.

Таблица 17 - Опасные и вредные факторы при разработке:

Факторы (по ГОСТ 12.0.003-2015)	Нормативные документы
<ul style="list-style-type: none"> • Неудовлетворительное освещение рабочей зоны; 	СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95*
<ul style="list-style-type: none"> • Неудовлетворительный микроклимат 	ГОСТ Р 56502-2020 Системы обеспечения микроклимата. Оценка энергетической эффективности систем отопления и вентиляции при проектировании зданий
<ul style="list-style-type: none"> • Повышенный уровень шума и вибрации на рабочем месте; 	СП 51.13330.2011. Защита от шума. ГОСТ 12.1.012-2004 ССБТ. Вибрационная безопасность. Общие требования
<ul style="list-style-type: none"> • Статическое электричество 	ГОСТ 12.1.019-2017 ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты

7.2.1 Анализ опасных и вредных производственных факторов

1) Источник возникновения фактора;

Нестандартное освещение, нестандартные параметры микроклимата, чрезмерный шум, излучение и токи, запыленность и загазованность воздуха в рабочей зоне, вибрация, движущиеся детали машин и механизмов,

образование промышленных отходов, взрыво- и пожароопасность и другие чрезвычайные ситуации.

2) Воздействие фактора на организм человека;

Нестандартное освещение может вызвать зрительное напряжение и усталость у экспериментатора; чрезмерный шум, излучение и токи, пыль и загрязнение воздуха в рабочей зоне могут вызвать слуховое и психическое напряжение у экспериментатора; вибрация, движущиеся детали машин и механизмов, образование промышленных отходов, опасность взрыва и пожара и другие чрезвычайные ситуации потенциально могут создать угрозу личной безопасности экспериментатора.

3) Анализ максимальных норм, допустимых для каждого фактора риска;

Для воздействия вредного шума уровень шума ПК на рабочем месте не должен превышать допустимого значения (50 дБА) согласно СП 51.13330.2011.

Для ослабления шума на рабочем месте обычно решается акустической обработкой с использованием звукопоглощающих материалов с максимальным коэффициентом поглощения (альфа) в диапазоне частот (63-8000) Гц в качестве облицовки закрытых поверхностей.

4) Для снижения воздействия факторов принимаются коллективные и индивидуальные меры.

Режим работы оборудования был рационализирован за счет применения электронных приборов с низким уровнем излучения, использования специально организованного освещения, а также выдачи персоналу лаборатории спецодежды, защитных масок и защитной обуви.

7.2.2 Анализ показателей шума и вибрации

Предельно допустимый уровень шума в цехе должен быть не более 80дБА, что соответствует ГОСТ 12.1.003-83. Под воздействием шума, превышающего 85-90 дБА, снижается слуховая чувствительность.

Для снижения шума можно использовать следующие методы:

- уменьшение шума в источнике;
- изменение направленности излучения;
- рациональная планировка предприятий и цехов;
- акустическая обработка помещений;
- уменьшение шума на пути его распространения.

7.2.3 Анализ показателей микроклимата

Согласно ГОСТ 12.1.005-88 "Настройки ССБТ. Гигиенические требования к воздуху, температуре и пыли в рабочей зоне", различные климатические факторы в рабочей зоне оцениваются в зависимости от периода, вида работ и тепла, выделяемого во время работы.

- тёплый, характеризующийся среднесуточной температурой наружного воздуха $+10^{\circ}\text{C}$ и выше;
- холодный и переходный – ниже $+10^{\circ}\text{C}$.

Участникам экспериментов рекомендуется сократить использование приборов высокой мощности для уменьшения выделения тепла, а также носить маски для лица и перчатки во время работы для уменьшения опасности.

7.2.4 Анализ освещенности рабочей зоны

Убедитесь, что каждый отдельный испытательный стенд имеет две лампы естественного освещения. Анализ естественного и искусственного освещения в тестовой зоне, а также разработка норм согласно СанПиН 2.2.1/1278-03 и СНиП 23-05-95.

Потолок также является отражающей поверхностью, и лаборатории следует контролировать яркость потолка до уровня менее 200 кд/м^2 , поскольку уменьшение естественного света приведет к недостаточному освещению в мастерской, так как плохое освещение в цехе вызвано

снижением уровня естественного света из-за загрязнения световых отверстий, стен и стеклянных поверхностей потолка.

Рекомендуется отделывать лаборатории материалами с подходящей отражательной способностью для потолков и стен, а также устанавливать достаточное искусственное освещение для своевременного дополнительного освещения при снижении естественного света.

7.2.5 Анализ электробезопасности

Электронное лабораторное оборудование и персональные компьютеры, используемые в лабораториях, генерируют электромагнитные поля, которые могут оказывать специфическое воздействие на здоровье. Длительное воздействие интенсивных электромагнитных полей на организм человека может привести к ряду проблем со здоровьем, например, к развитию сердечно-сосудистых и неврологических заболеваний, органов дыхания и нервной системы. Нарушения сердечно-сосудистой и нервной систем, органов дыхания и нервной системы при воздействии интенсивности выше предельно допустимого уровня.

Источниками электромагнитного излучения являются персональные компьютеры, блоки питания, трансформаторы, лабораторное оборудование и т.д. Ниже перечислены необходимые стандарты:

Диапазоны частот	Предельно допустимая энергетическая экспозиция		
	По электрической составляющей, $(В/м)^2 \times ч$	По магнитной составляющей, $(А/м)^2 \times ч$	По плотности потока энергии $(мкВт/см^2) \times ч$
30 кГц - 3 МГц	20000,0	200,0	-
3 - 30 МГц	7000,0	Не разработаны	-
30 - 50 МГц	800,0	0,72	-
50 - 300 МГц	800,0	Не разработаны	-
300 МГц - 300 ГГц	-	-	200,0

7.2.6 Анализ пожарной безопасности

Пожарная профилактика основывается на исключении условий, необходимых для горения, и использования принципов обеспечения 75

безопасности. При обеспечении пожарной безопасности решаются следующие задачи:

возгорание; локализация возникших пожаров; защита людей и материальных ценностей; предотвращение пожаров; тушение пожара.

Для предотвращения пожаров необходимо:

1) Рабочие должны быть проинструктированы о мерах предосторожности и проведены инструктажи.

2) В каждом цехе должны быть предусмотрены меры по эвакуации.

3) Обязаны присутствовать средства пожаротушения (в качестве первичных средств пожаротушения пенные огнетушители ОХВП-10, углекислотные огнетушители ОУ-2, ОУ-5, и ОУ-8 1-у штуку на 700м² площади, ящики с песком 1-ин на 500м² площади).

4) Инструкции на случай пожара должны быть вывешены на видном месте, с надлежащим указанием плана эвакуации, а также безопасного доступа и всеобщего доступа к номерам экстренных служб для всех.

5) Необходимо установить и усовершенствовать систему пожарной сигнализации.

6) Интегрировать систему пожарной сигнализации лаборатории с системой пожарной сигнализации на пожарной станции, чтобы обеспечить возможность первого реагирования в случае пожара.

7.3 Экологическая безопасность

В этом разделе описывается воздействие на окружающую среду и приводятся примеры решений. Анализируется воздействие на окружающую среду загрязняющих газов, сточных вод и других отходов, образующихся в ходе экспериментов, выявляются опасности и описываются решения.

Эксперименты представляют собой разработку программного обеспечения и комплекс решений системной поддержки ISaGRAF, КРОСС-500 и MasterSCADA, для атмосферы, с меньшим влиянием на скальные образования. В основном это связано с выходом из строя персональных

компьютеров, люминесцентных ламп и других электроприборов. Заслуживают внимания и корзины для макулатуры. Воздействие на водные ресурсы также происходит из-за отходов жизнедеятельности человека.

Использование экспериментального оборудования приостанавливается во время расчетов или анализа результатов, чтобы уменьшить количество тепла, выделяемого аппаратурой; сточная вода, используемая во время экспериментов, перерабатывается, чтобы избежать воздействия на водные ресурсы; макулатура, пыль и другой мусор, образующийся во время экспериментов, отделяются и перерабатываются.

7.4 Безопасность в чрезвычайных ситуациях

Чрезвычайная ситуация - это ряд опасных событий или непредвиденных обстоятельств, которые подвергают жизнь опасности или даже приводят к ее прекращению. К таким событиям относятся перебои в подаче электроэнергии, резкие изменения температуры, избыточное давление, попадание токсичных веществ, длительное излучение высокой интенсивности, чрезмерный шум и вибрация и т.д.

Основные причины чрезвычайных ситуаций в основном внутренние и включают в себя: слишком старое оборудование, плохие контакты питания, дефектные проектные экспериментальные методы, опасные и сложные методики, а также неопытный экспериментальный персонал. Внешние чрезвычайные ситуации, включая: например, стихийные бедствия, перебои в подаче электроэнергии и т.д.

Для чрезвычайных ситуаций природного характера: перед проведением экспериментов проводятся имитации предсказуемых стихийных бедствий для получения соответствующих ответных действий и разработки методов самопомощи в непредсказуемых ситуациях. Для технических проблем: сократить или даже запретить вход в лабораторию лицам, не являющимся персоналом, чтобы уменьшить непредсказуемые опасности;

иметь технического консультанта в каждой лаборатории, чтобы попытаться избежать опасностей, вызванных техническими проблемами.

7.5 Заключение по разделу социальная ответственность

В этом разделе рассматриваются вопросы социальной ответственности и юридической безопасности при проведении испытаний. В нем также анализируются соответствующие положения законодательства и техники безопасности.

На исследованных рабочих местах все производственные факторы соответствуют стандартам. Согласно Правилам электробезопасности, категория помещения соответствует категории 1 - "нет помещений повышенной опасности". Согласно Правилам по охране труда, при эксплуатации электроустановок персонал должен иметь допуск по электробезопасности 1 группы. Тяжелые работы в лабораториях относятся к категории Ib согласно 1.2.3685-21 "Санитарные правила и требования к обеспечению безопасных и безопасных условий труда людей".

8 СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Контроллер для распределённых открытых систем КРОСС. Руководство по эксплуатации ЯЛБИ.421457.018 РЭ
2. ISaGRAF Version 3.4. Руководство пользователя.
3. Любашин А. Н. Что такое ISaGRAF? – М.: АО РТСофт
4. ИнСАТ MasterSCADA. Основы проектирования :
Учебник по MasterSCADA
5. Ключев А.С. Автоматическое регулирование: учебник для вузов.- М.: Высш. Шк., 1986.-351 с.
6. Изерман Р. Цифровые системы управления: Пер. с англ.- М.: 1984.- 541 с.
7. Корчагина Р.Л. Техничко-экономические обоснования при разработке радиоэлектронных приборов и устройств. / Учебное пособие по дипломному проектированию. – Л.: Механический институт, 1988.
8. Баранов А.Н., Пчелинцев В.А. Пожарная безопасность: Учебное пособие. – М.: издательство АСВ, 1997 – 176 с.
9. СанПиН 2.2.2.542-96. Гигиенические требования к ВДТ и ПЭВМ, организации работы.
10. Безопасность жизнедеятельности/Под ред. С. В. Белова. - М.: Высш.шк.,1999.
11. Охрана труда в машиностроении/Под ред. Е.Я. Юдина.- М.:Машиностроение,1985.
12. Сибаров Ю. Г.и др. Охрана труда в вычислительных центрах. - шк.,1991 - М.:Машиностроение, 1990.

13. Охрана окружающей среды /Под ред. С.В. Белова. – М.: Высш.шк., 1991.
14. ПУЭ /Изд. Главэнергонадзор России, 6-е изд. – М.,1998.
15. СН 245-71.
16. Долин П.А. Справочник по технике безопасности. - М.: Энергоатомиздат, 1984. - 824 с.

**ПРИЛОЖЕНИЕ А.
МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

Ли Яньчэн

**СОЗДАНИЕ ПРОГРАММ УПРАВЛЕНИЯ ЛАБОРАТОРНОЙ
УСТАНОВКОЙ «ГИДРАВЛИЧЕСКИЙ ОБЪЕКТ» В СИСТЕМЕ
ISaGRAF НА ЯЗЫКАХ ПРОГРАММИРОВАНИЯ FBD И ST**

Содержание

1 Структурная схема лабораторного комплекса	1
2 Создание программы контроля уровня воды с использованием языка FBD в пакете ISaGRAF	4
2.1 Создайте новый проект и дайте ему имя	4
2.2 Создание функциональных блоков	5
2.3 Объявление переменных	11
3 Написание программы для САР уровня на языке FBD	14
3.1 Ввод-вывод аналоговых и дискретных сигналов	14
3.2 Импульсный ПИД-регулятор с возможностью ручного управления ИМ	14
3.3 Создание предупредительной и аварийной сигнализации	15
3.4 Алгоритм управления насосом НЗ	15
3.5 Получите полную программу работы	16
4 Отладка программы	17
5 Создание программы на языке структурированного текста ST	19
5.1 Основной синтаксис ST	19
5.2 Выражения и скобки	20
5.3 Вызов функциональных блоков	20
5.4 Пример создания функционального блока OGR на языке ST	21
5.5 Пример создания элемента ИЛИ	21
6 Написание программы для САР уровня на языке ST	22
6.1 Создание блоков ввода-вывода	22
6.2 Реализация импульсного ПИД-регулятора	23
6.3 Описание Ручного режима управления и выходов регулятора	23
6.4 Реализация предупредительной и аварийной сигнализации	24
6.5 Реализация алгоритма работы насоса НЗ	24
6.6 Получите полную программу работы	25
7 Задание на лабораторную работу	26
8 Методические указания по выполнению отдельных пунктов задания ..	27
9 Контрольные вопросы	28

Цель работы: получение практических навыков программирования в системе ISaGRAF на примере создания программ для САР уровня на языках программирования FBD и ST, на базе контроллера КРОСС.

1 Структурная схема лабораторного комплекса

В рассматриваемую в данной работе систему управления гидравлическим объектом входят:

- гидравлический объект;
- стенд с контроллером;
- персональный компьютер- операторская станция.

Структура экспериментальной системы контроля уровня жидкости показана на рисунке 1.

Экспериментальная установка горизонтального контроля состоит из трех резервуаров для воды E1, E2 и E3, горизонтальных датчиков, клапанов и насоса давления воды.

Резервуар E1 является резервуаром для воды, который заполняется водой в начале, а затем вода из резервуара E1 перекачивается в резервуар E3 в результате регулировки уровня воды или ручного воздействия. Резервуар E1 также имеет клапан, ведущий в резервуар E2, к которому подключен привод (EA) с мембранным исполнительным механизмом (PO). Устройство управления может переключаться между автоматическим и ручным режимом, в котором оно управляется кнопкой, а в автоматическом режиме - контроллером.

Двигатель исполнительного механизма вращает мембранный вентиль посредством червячной передачи. Мембранный вентиль исполняет роль регулирующего органа. Исполнительный механизм оснащён концевыми выключателями, которые обеспечивают выключение электродвигателя при достижении крайних положений вентиля. Также на валу ИМ расположен указатель положения (УП) реостатного типа.

В ёмкости E2 расположен аналоговый датчик уровня (ДУ). При заполнении водой зазора между электродами, питающимися от внешнего источника, изменяется напряжение. Нормирующий преобразователь (Н.П) преобразует это напряжение в унифицированный сигнал в виде тока от 0 до 20 мА, этот сигнал поступает на аналоговый вход контроллера.

Ёмкость E2 соединена с ёмкостью E3 через вентиль мембранного типа, служащего для внесения возмущения в систему, путём изменения слива из E2 в E3.

Ёмкость E3 оснащена тремя дискретными датчиками уровня (Д.ДУ 1..3), расположенными на разном уровне, и насосом, который также может управляться либо с панели управления, либо с контроллера.

Функции управления и регулирования в системе выполняет промышленный микропроцессорный контроллер «КРОСС».

В качестве операторской станции в системе используется компьютер типа IBM PC на базе процессора Celeron 2800, с установленной SCADA системой Master.SCADA и системой программирования ISaGRAF.

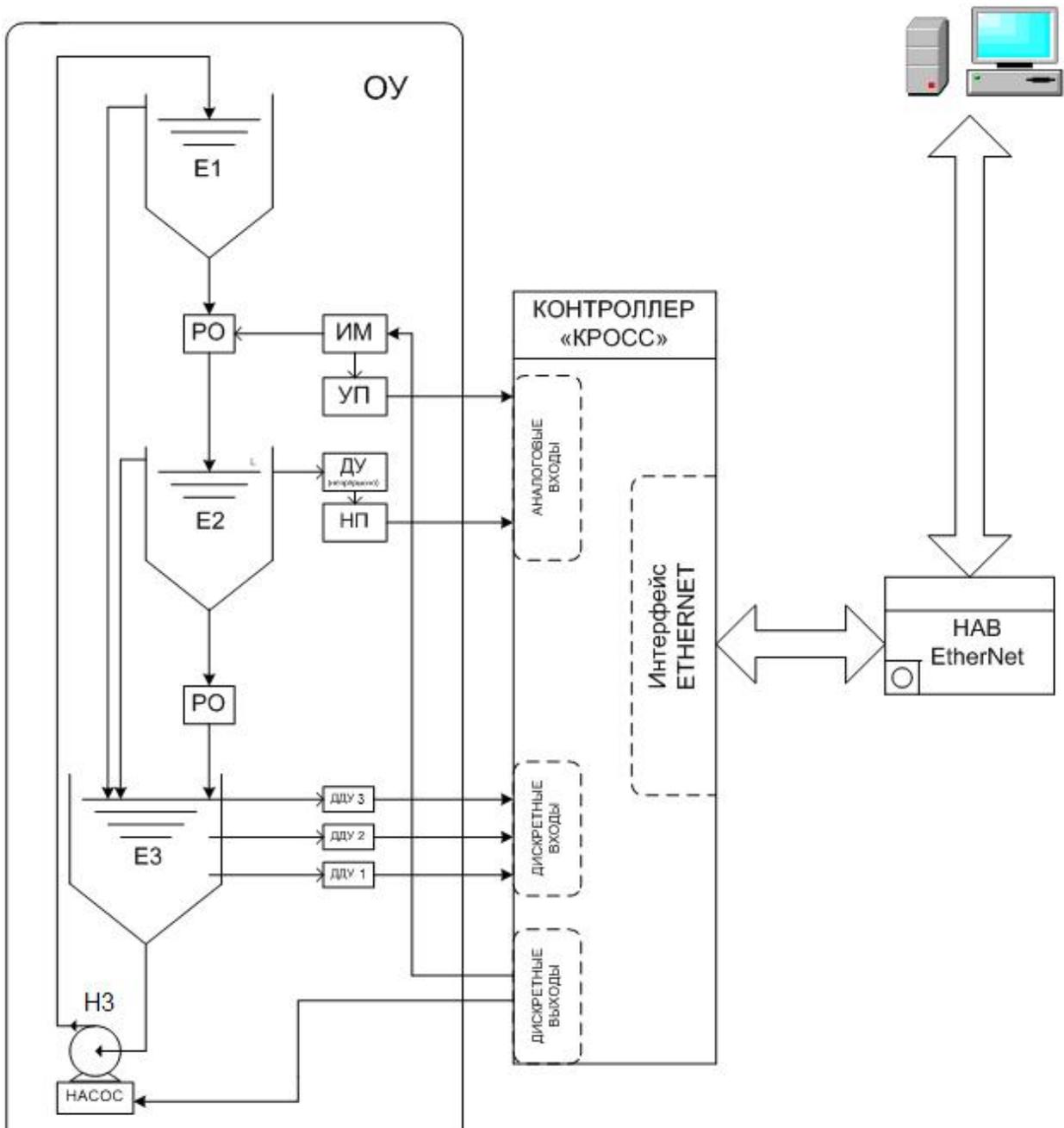


Рисунок 1 -Структурная схема лабораторного комплекса

2 Создание программы контроля уровня воды с использованием языка FBD в пакете ISaGRAF

2.1 Создайте новый проект и дайте ему имя

Программный пакет ISaGRAF использует язык FBD для создания программы автоматического контроля уровня воды, ручного управления приводом (ЕС) и системы сигнализации предельного значения уровня воды.

Запустите приложение через меню **ПУ.СК. / ПРОГ.РАММЫ. / ISaGRAF** или, дважды нажав Л.К мыши по ярлыку на рабочем столе .

Создадим новый проект через меню **ФАЙЛ. / НОВЫЙ** или иконку **СОЗДАТЬ НОВЫЙ ПРОЕКТ**. После этого на экране появится окно (рисунки 1), в котором необходимо задать имя своему проекту, придерживаясь следующих правил:

- Длина имени не должна превышать 8 символов;
- Первый символ должен быть буквой;
- Следующие символы могут быть буквой, цифрой или символом подчёркивания;
- Прописной и строчные регистры не различаются.

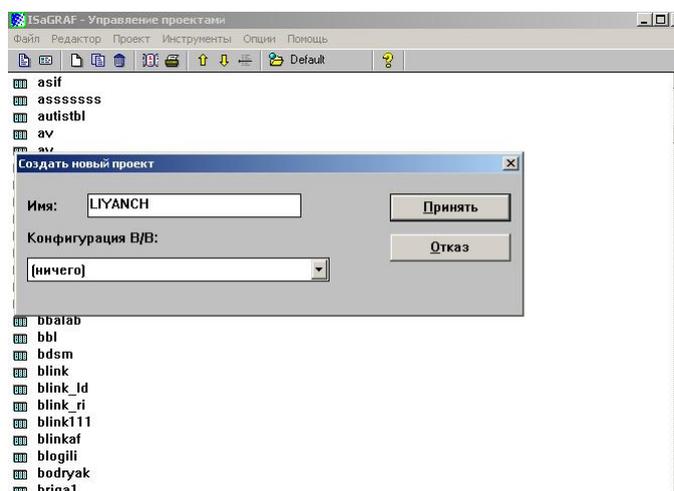


Рисунок 1 – Создание нового проекта

После присвоения имени новый созданный элемент появится в списке элементов. Пользуясь командой **Редактор. / Текст комментария**, добавьте комментарий описания проекта. Нажав два раза Л.К мыши на проекте,

появится окно **ПРО.ГРАММ.Ы**, в котором вы должны создать новую программу через меню **Ф.АЙЛ. / НОВ.ЫЙ**. Новой программе нужно присвоить имя (.main.), а также выбрать язык (.FBD – Функциональная Блочная Диаграмма), комментарий и стиль (Begin: основная программа) (рисунок 2).

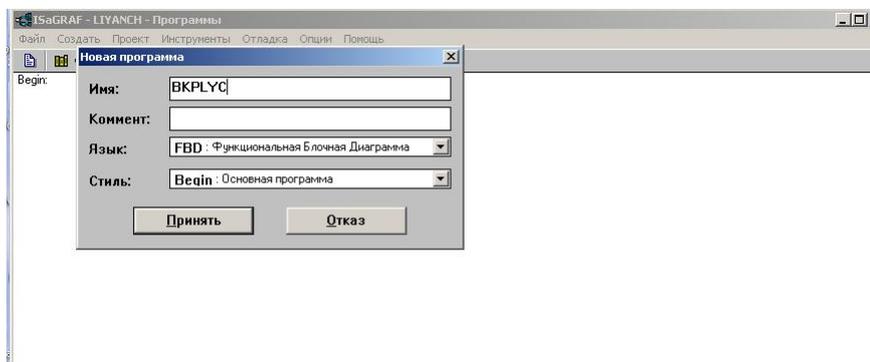


Рисунок 2 – Создание новой программы

Нажав два раза ЛК мыши на имени программы, вы попадёте в рабочую область вашей программы (рисунок 3).

2.2 Создание функциональных блоков

Язык функциональных блоков F.BD построен на библиотеке элементарных арифметических, логических и динамических функций. Выберите в панели управления нужный функциональный блок. Затем вставьте его в рабочую область программы ЛК мыши, как показано на рисунке 3.

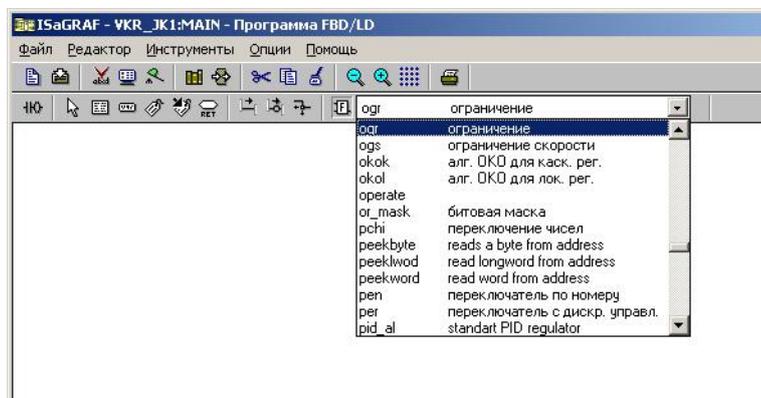


Рисунок 3 – Окно программы на языке FBD

Нажав два раза ЛК по изображению блока, откроется окно. Получить дополнительную информацию о данном функциональном блоке можно нажав, кнопку Информация (рисунок 4).

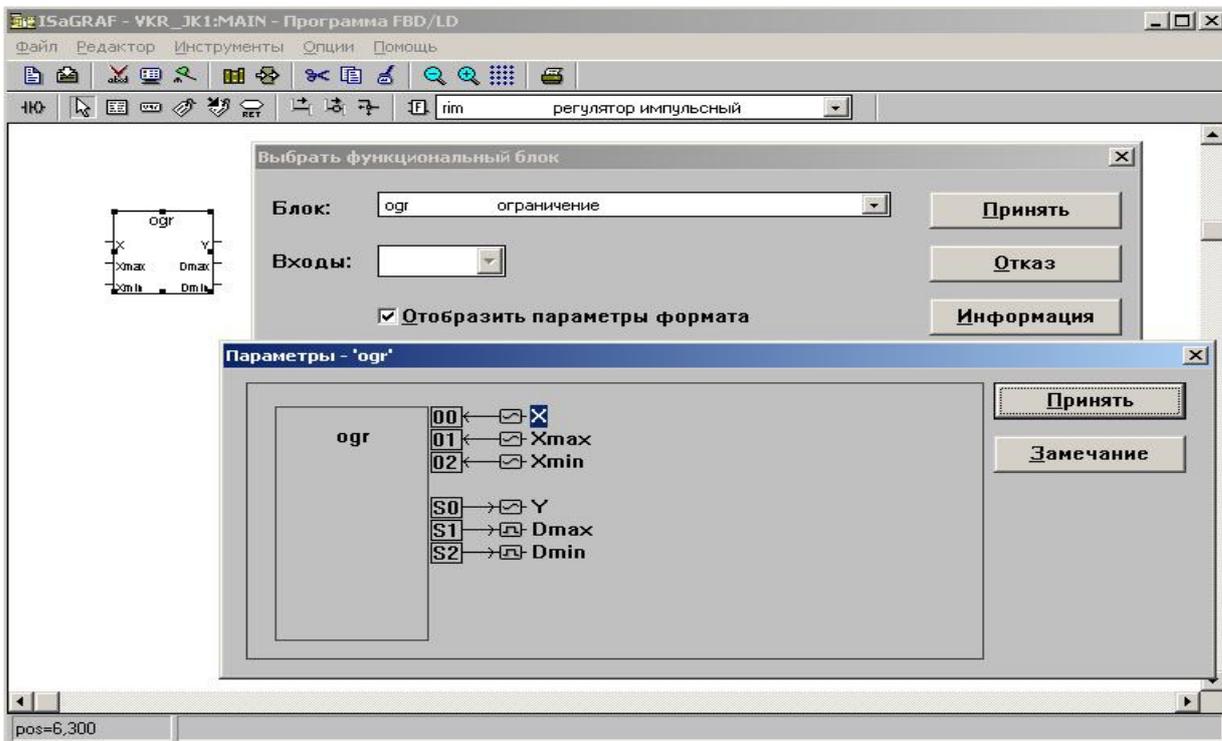


Рисунок 4 – Параметры блока

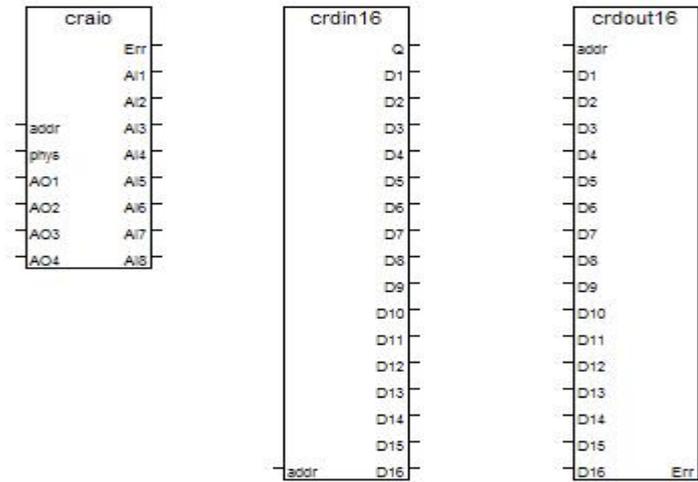
Каждый функциональный блок имеет свою фиксированную функцию. Входные и выходные переменные двух функциональных блоков должны быть связаны во время написания программы. Кроме того, тип каждой переменной в блоке должен совпадать с типом соответствующего входа, как указано.

Входом блока FBD может быть внутренняя переменная, константа или переменная.

Выходом блока FBD может быть внутренняя переменная, выходная переменная или имя программы (только для подпрограмм).

Для реализации системных программ уровня управления необходимы следующие функциональные блоки:

Блоки ввода-вывода аналоговых, Блоки ввода дискретных и Блоки вывода сигналов craio , crdin16, crdout16



Addr – адрес модуля (craio – 40, crdin16 - 30, crdout16 - 30);

Phys – признак необходимости преобразования (true – преобразовывать в физическую величину, false – в процентах от диапазона измерения);

Для контроля и управления гидравлическим объектом задействованы следующие входы (выходы).

Аналоговые:

Ai4 – сигнал положения исполнительного механизма;

Ai5 – сигнал значения уровня в емкости E2;

Дискретные:

DI2 – уровень L3;

DI3 – уровень L2;

DI4 – уровень L1;

DO2 – сигнал на включение/ отключение насоса Н3;

DO3 – сигнал «Меньше»;

DO4 – сигнал «Больше»;

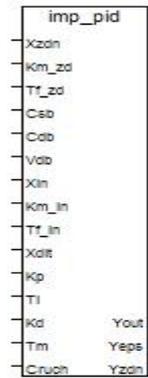
DO5 – сигнализация по max;

DO6 – сигнализация по min;

DO7 – звуковая сигнализация;

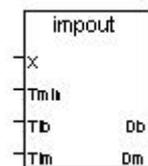
Регулирование импульсное imp_pid

ПИД-регулятор с импульсным выходом выдает регулируемый импульсный сигнал на следующий функциональный блок.



Этот алгоритм используется для построения ПИД-регуляторов для работы с приводами с постоянной скоростью. Алгоритм часто используется в сочетании с алгоритмом импульсного выхода, который преобразует аналоговый выходной сигнал алгоритма RIM в серию импульсных сигналов для управления исполнительным механизмом. Помимо генерации управляющего сигнала, алгоритм также вычисляет ковариационный сигнал, фильтрует ковариационный сигнал и создает мертвую зону.

Impout – формирователь импульсного вывода по заданной скважности.



Импульсатор использует RC-осциллятор или другую колебательную цепь для генерации периодического выходного импульсного сигнала. Основная структура состоит из осциллятора, делителя частоты, буферного усилителя и выходной цепи. В генераторе импульсов осциллятор генерирует сигнал квадратной волны с фиксированной частотой и амплитудой и делит его на нужную частоту импульса с помощью делителя частоты.

Методы регулировки импульсаторы При использовании генератора импульсов необходимо произвести правильную регулировку, чтобы получить более точный и стабильный выходной сигнал. Ниже приведены некоторые распространенные методы настройки генератора импульсов: Настройка частоты: Изменение частоты колебаний путем изменения емкости или сопротивления в цепи генератора.

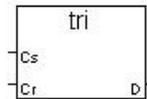
В прикладных программах правильная настройка импульсаторы имеет большое значение для обеспечения правильной работы схемы. Понимание принципа работы генератора импульсов и распространенных методов настройки поможет нам правильно использовать этот компонент.

Если $X > T_{lb}$, то выходе $D_b = 1$ (больше), если $X < T_{lm}$, то на выходе $D_m = 1$ (меньше).

$T_i = 0$ (т.е. импульсы не генерируются).

Алгоритм не содержит каскадных входов и выходов и осуществляет блокировку процедуры обратного отсчета.

RS-Триггер.



name: TRI - RS-триггер

description:

Алгоритм применяется для запоминания дискретных сигналов.

call:

BOOLEAN Cs // команда установки

BOOLEAN Cr // команда сброса

return:

BOOLEAN D // выход

notes:

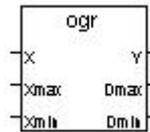
Cs означает set. Когда $C_s = 1$ и $C_r = 0$, поскольку $C_s = 1$, независимо от того, будет ли состояние D перед установкой 0 или 1, мы знаем из таблицы истинности, что на выходе всегда будет 0! В беззатворном устройстве это состояние называется "установлено в единицу".

R означает сброс, при установке $C_r = 1$ и $C_s = 0$, поскольку $C_r = 1$, независимо от того, что состояние не D перед установкой равно 0 или 1, на выходе будет результат 0, т.е. $D = 0$, т.е. мы называем $C_r = 1$, если хотим, чтобы

состояние флип-флопа могло быть сброшено обратно в 0. (Конечно, после $D=0$, так как $Cs = 0$, $D=0$, поэтому существует не- $D=1$.) В не-гейте это состояние называется "установлено в ноль".

Ограничение ogr.

Алгоритм применяется для ограничения верхнего и/или нижнего пределов диапазона сигнала. Два дискретных выхода D_{max} и D_{min} определяют достижение верхнего и нижнего пределов сигнала.



Если $X > X_{max}$, то выход $D_{max}=1$, $Y=D_{max}$.

Если $X < X_{min}$, то выход $D_{min}=1$, $Y=D_{min}$.

Если $X_{min} < X < X_{max}$, то выход $Y=0$.

Логическое ИЛИ.



Замечание: для этого оператора количество входов может быть больше, чем два.

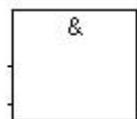
Аргументы:

Inputs BOOLEAN

Outputs BOOLEAN логическое ИЛИ двух и более входов.

Описание: логическое ИЛИ двух и более значений.

Логическое И.



Замечание: для этого оператора количество входов может быть больше, чем два.

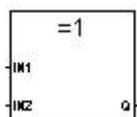
Аргументы:

Inputs BOOLEAN

Outputs BOOLEAN логическое И двух и более входов.

Описание: логическое И двух и более значений.

Исключающее ИЛИ.



Аргументы:

IN1 BOOLEAN

IN2 BOOLEAN

Q BOOLEAN исключающее ИЛИ двух входов.

Описание: исключающее ИЛИ двух значений.

2.3 Объявление переменных

Для использования переменных их сначала необходимо объявить в словаре. Для этого вызываем **СЛОВАРЬ** из меню **ФАЙЛ**. Или нажав иконку  на панели и инструментов.

Далее выбираем закладку с необходимым типом переменной (например **Целые/ Действительные**) и дважды щелкаем ЛК мыши в выделенной строке рабочего поля. В появившемся окне задаём параметры переменной: имя, комментарий(если необходимо), атрибуты и т.д. (рисунок 5).

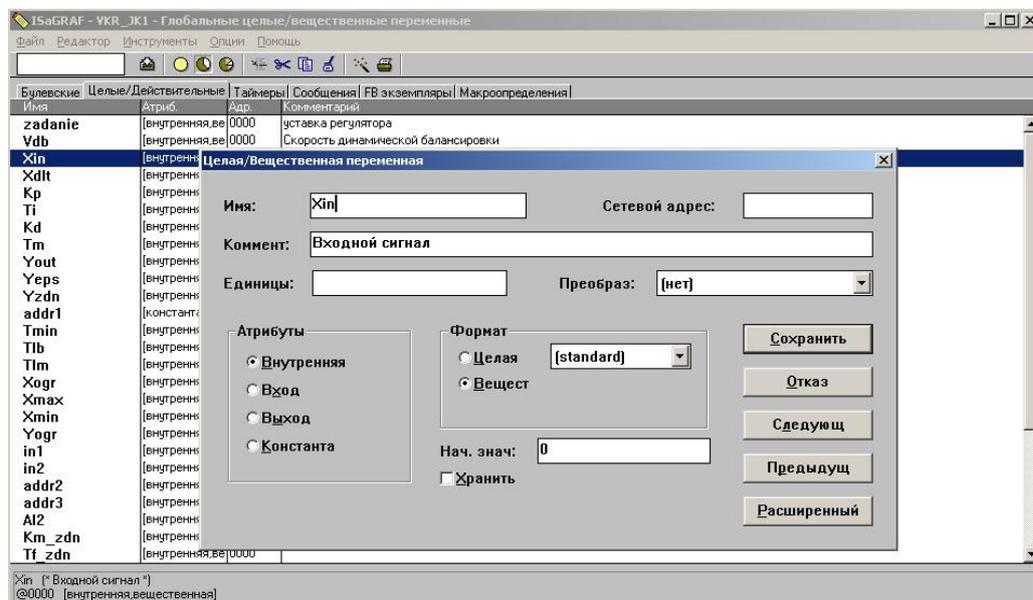


Рисунок 5 – Окно параметров переменной

После объявления переменных закрываем словарь, сохраняя все изменения. В таблице 1 приведены все переменные, необходимые для проекта.

Таблица 1 – Переменные используемые при создании программы

Имя переменной	Тип данных	Атрибуты	Формат	Начальное значение
Zadanie	Целые-действительные	Внутренняя	Вещественная	0
Vdb		Константа	Вещественная	0
Xin		Внутренняя	Вещественная	0
Xdlt		Константа	Вещественная	0,001
Kp		Внутренняя	Вещественная	1
Ti		Внутренняя	Вещественная	15
Kd		Внутренняя	Вещественная	0,1
Tm		Константа	Вещественная	20
Yout		Внутренняя	Вещественная	0
Yeps		Внутренняя	Вещественная	0
Yzdn		Внутренняя	Вещественная	0
Tmin		Внутренняя	Вещественная	0
Tlb		Внутренняя	Вещественная	0
Tlm		Внутренняя	Вещественная	0
Xogr		Внутренняя	Вещественная	0
Xmax		Внутренняя	Вещественная	50
Xmin		Внутренняя	Вещественная	15
AI2		Внутренняя	Вещественная	0
Km_zdn		Внутренняя	Вещественная	0
Tf_zdn		Внутренняя	Вещественная	0
Km_In		Внутренняя	Вещественная	0
Tf_In		Внутренняя	Вещественная	0
addr1		Константа	Целая	40
addr2		Константа	Целая	30
addr3		Константа	Целая	30
Xmax1		Внутренняя	Вещественная	60
Xmin1		Внутренняя	Вещественная	10

Csb	Булевские	Внутренняя		FALSE
Cdb		Внутренняя		
phis		Внутренняя		
Db		Внутренняя		
Dm		Внутренняя		
Dmax		Внутренняя		
Dmin		Внутренняя		
Cruch		Внутренняя		
Rezhim_H3		Внутренняя		
L1		Внутренняя		
L2		Внутренняя		
L3		Внутренняя		
Dtri		Внутренняя		
Ruch_H3		Внутренняя		
Control_H3		Внутренняя		
Dmaxl		Внутренняя		
Dminl		Внутренняя		
ZVUK		Внутренняя		
Db_Ruch		Внутренняя		
Dm_Ruch		Внутренняя		
sign_b		Внутренняя		
sign_m		Внутренняя		
rez		Внутренняя		
Q_XOR		Внутренняя		

Когда все переменные заданы, необходимо вставить их в рабочую область программы. Для этого в панели управления выбираем иконку VAR – **Вставить переменную**. Выбираем необходимую переменную и вставляем её в программу.

Теперь необходимо связать переменные с входом (выходом) функционального блока с помощью линий связи, предварительно нажав иконку  - **Начертить линию связи**. Нажав ЛК мыши на переменной и, не отпуская её, ведём линию до нужного входа функционального блока и отпускаем ЛК. Аналогично связываем другие переменные (рисунок 6).

Связать можно также выход одного блока с входом другого.

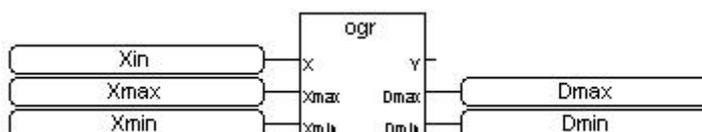


Рисунок 6 – Функциональный блок с переменными

В процессе создания программы, а также после ее написания, необходимо выполнять проверку её на синтаксис. Для этого нужно зайти в меню **ФАЙЛ/ПРОВЕРИТЬ**. Если ошибок нет, то выходим из генератора кода.

3 Написание программы для САР уровня на языке FBD

Теперь перейдём непосредственно к реализации проекта.

Для более наглядного вида основная программа представлена в виде отдельных подпрограмм, реализующих разные функции. Рассмотрим их по отдельности.

3.1 Ввод-вывод аналоговых и дискретных сигналов

Для реализации ввода-вывода аналоговых и дискретных сигналов используем блоки `crain0`, `crdin16`, `crdout16` (Рисунок 7):

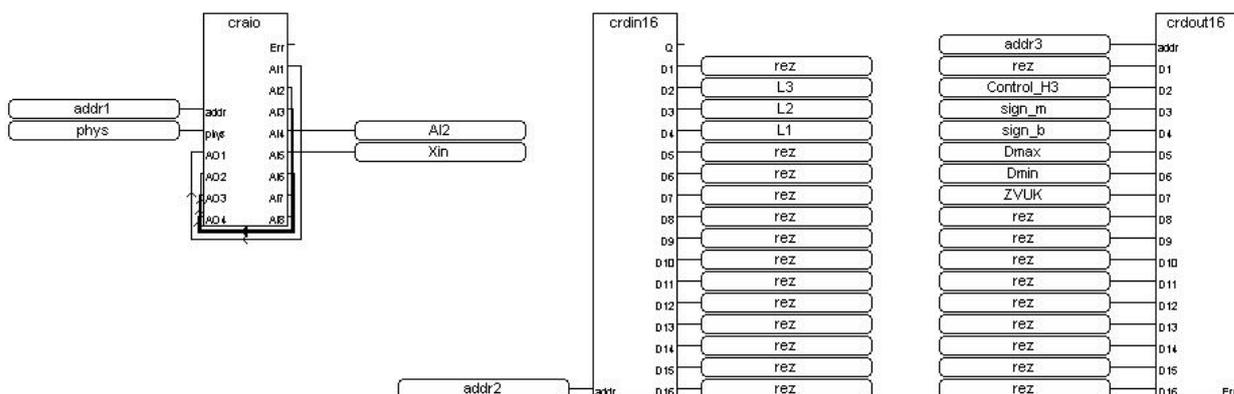


Рисунок 7 – Фрагмент программы: блоки ввода/ вывода

3.2 Импульсный ПИД-регулятор с возможностью ручного управления ИМ

Данная подпрограмма должна обеспечивать поддержание уровня жидкости в ёмкости E2 при определённом задании. Регулирование уровня осуществляется изменением расхода на притоке по ПИД закону с использованием функционального блока ПИД-регулятора. Ручное управление реализовано на элементах И и ИЛИ (рисунок 8).

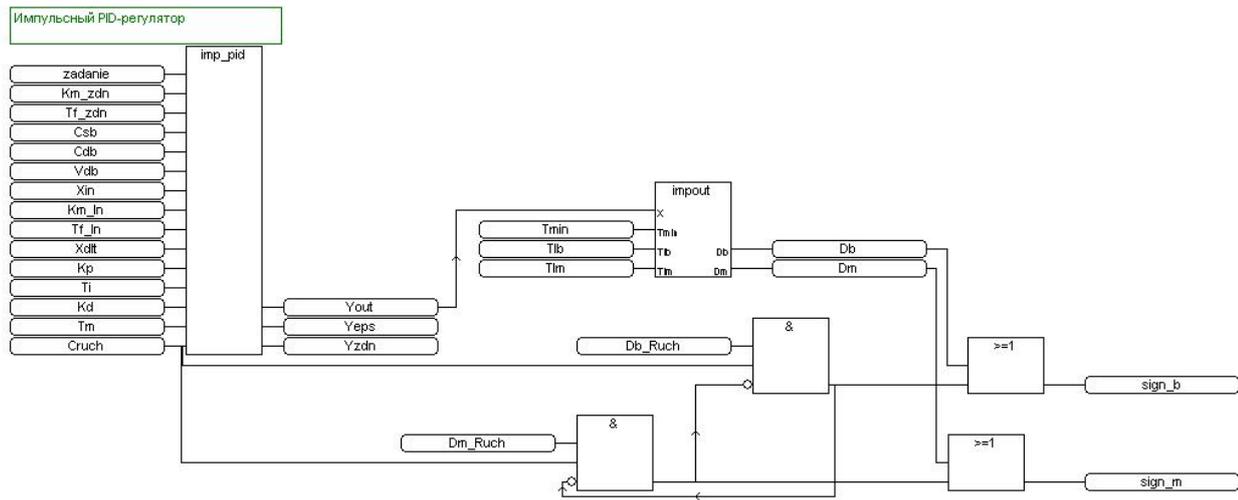


Рисунок 8 - Фрагмент программы. Реализация импульсного регулирования

3.3 Создание предупредительной и аварийной сигнализации

Задача: реализовать срабатывание предупредительной световой и аварийной звуковой сигнализации при превышении или понижении входной величины установленных порогов. Сделаем это с помощью функциональных блоков OGR (рисунок 9, 10).



Рисунок 9



Рисунок 10 - Фрагмент программы. Предупредительная и аварийная сигнализация

3.4 Алгоритм управления насосом НЗ

При наполнении емкости ЕЗ происходит срабатывание сигнализатора уровня L3, по сигналу которого включается насос НЗ. При достижении уровня

жидкости минимального значения происходит срабатывание сигнализатора уровня L1, в следствии чего насос Н3 отключается .

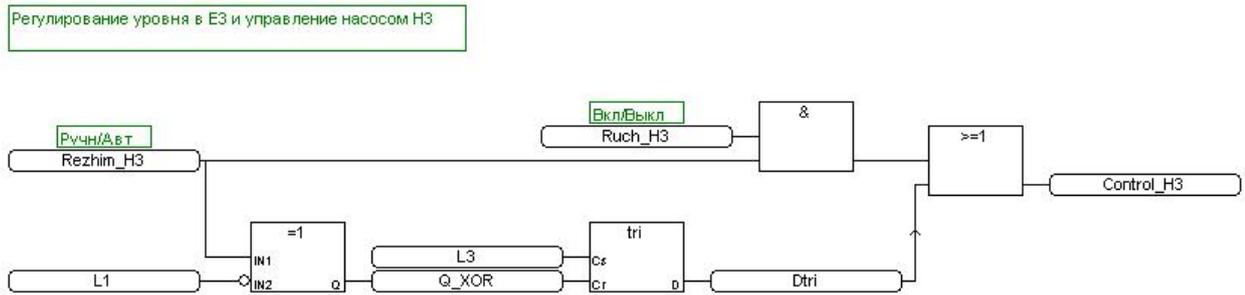


Рисунок 11 - Фрагмент программы. Управление насосом Н3

3.5 Получите полную программу работы

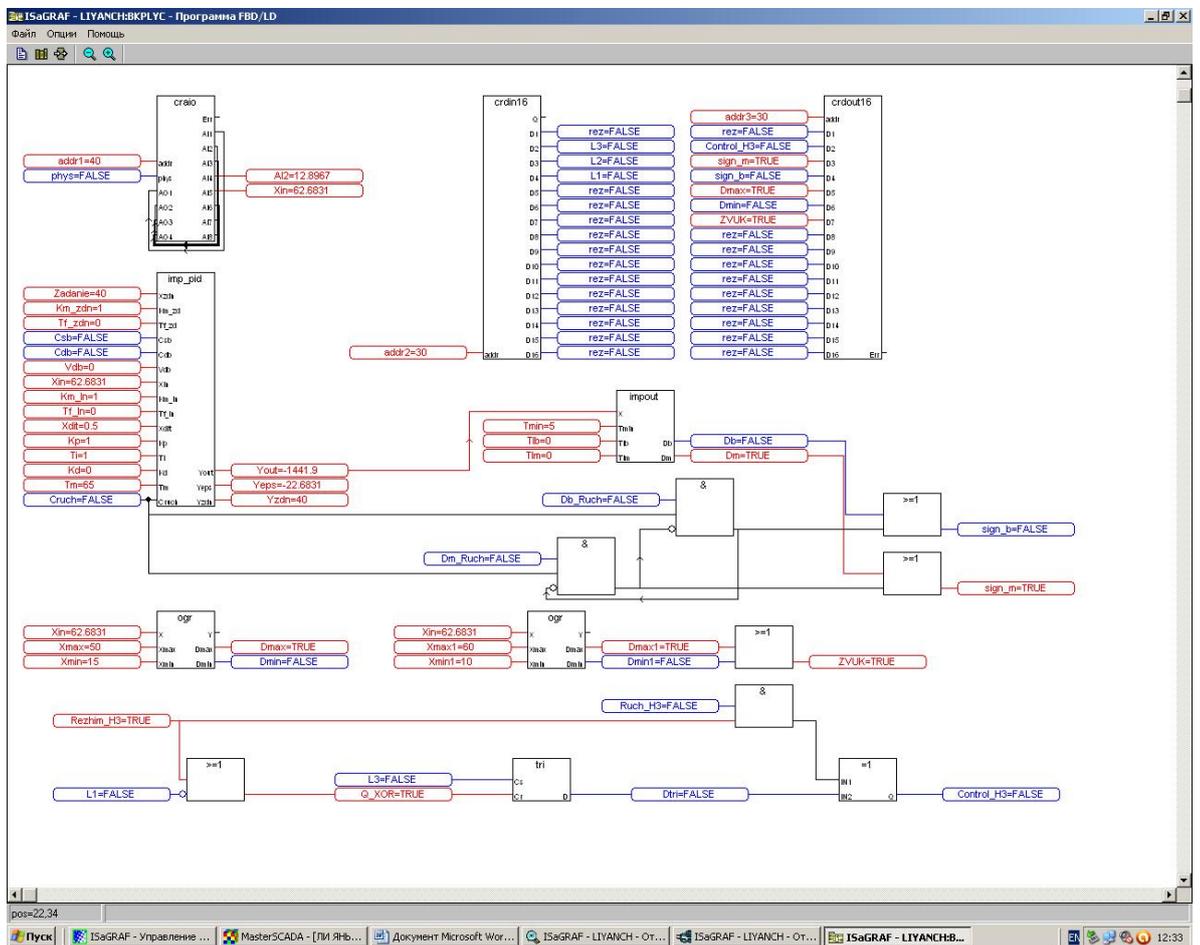


Рисунок 12 - Схема работы, когда уровень воды в баке E2 ниже установленного значения

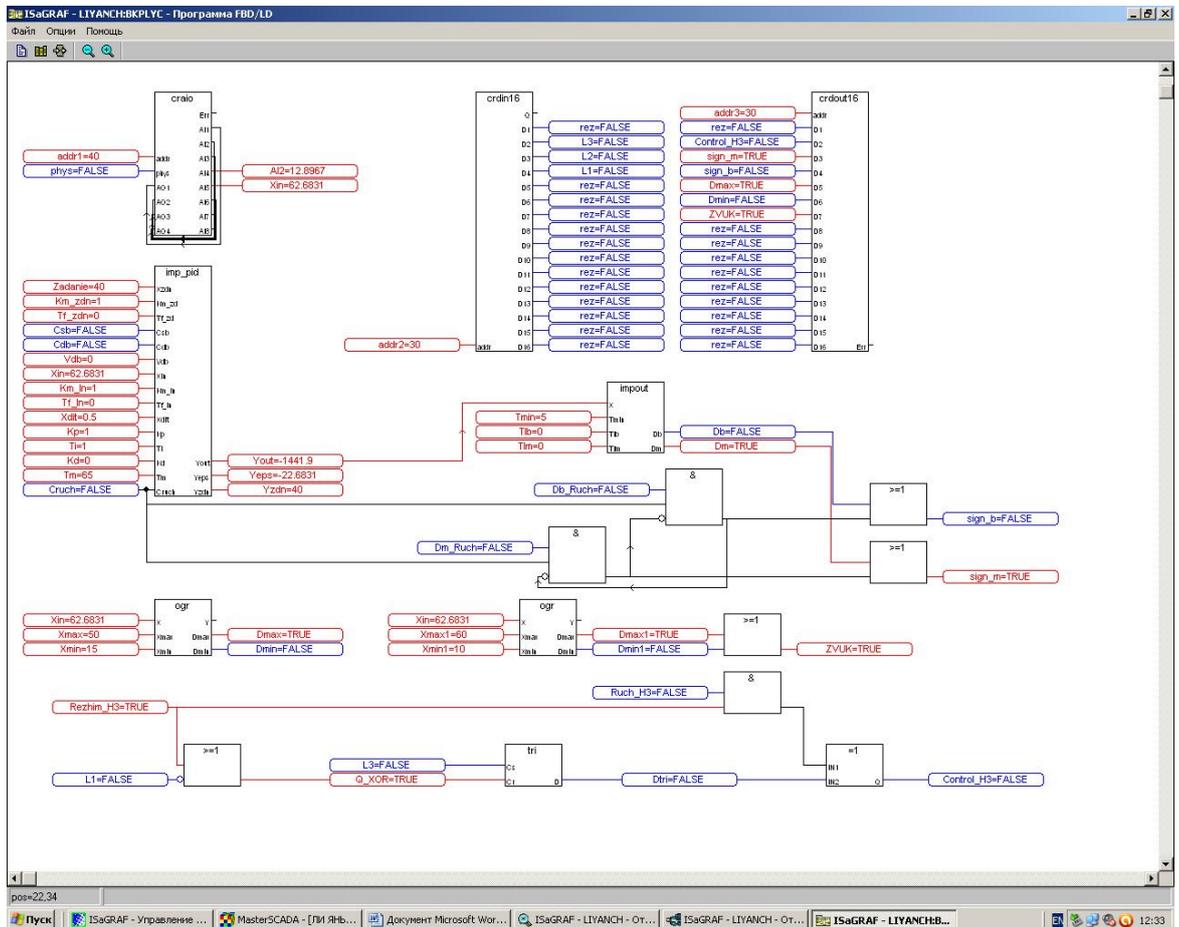


Рисунок 13 - Схема работы, когда уровень воды в баке E2 больше установленного значения

4 Отладка программы

После того, как программа написана и объявлены переменные, ее необходимо проверить.

Сначала необходимо установить опции компилятора для взаимодействия целевой задачи ISaGRAF с контроллером КРОСС. Для этого заходим в меню **ОПЦИИ/ ОПЦИИ КОМПИЛЯТОРА** и устанавливаем метку напротив TIC for Motorola, т.к. процессор контроллера КРОСС произведен компанией Motorola (рисунок 14).

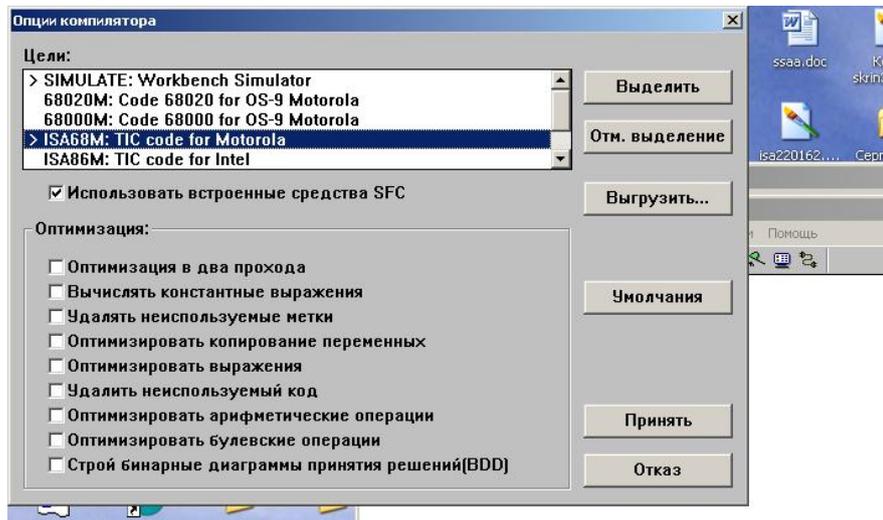


Рисунок 14 – Опции компилятора

Далее необходимо открыть окно ISaGRAF – ПРОГРАММЫ, выбрать команду СОЗДАТЬ/ СОЗДАТЬ КОД ПРИЛОЖЕНИЯ или нажать кнопку . Эта команда создаёт код приложения и проверяет программу на наличие синтаксических ошибок. Если ошибок нет, то выходим из генератора кода.

Следующим шагом будет установление связи между ПК и контроллером КРОСС. Открываем окно ISaGRAF – ПРОГРАММЫ, заходим в меню ОТЛАДКА/ УСТАНОВЛЕНИЕ СВЯЗЕЙ. Устанавливаем коммуникационный порт ETHERNET, а в установке прописываем Интернет адрес 192.168.0.2 – это адрес контроллера, номер порта 1100 (рисунок 15).

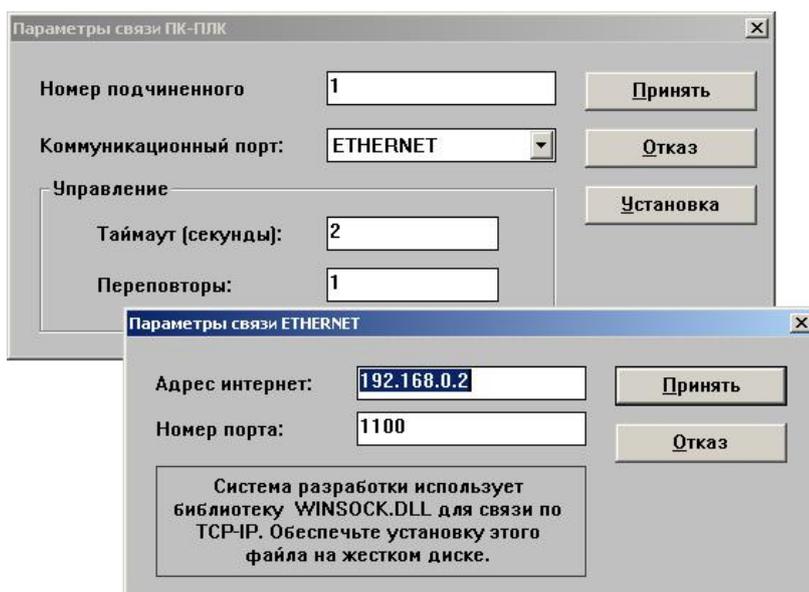


Рисунок 15 – Параметры соединения с контроллером

Программа готова к отладке контроллером. Включите контроллер при помощи автоматического выключателя на лабораторном стенде.

Открываем окно приложения ISaGRAF – Программы FBD/ LD. Запускаем отладчик с помощью меню **ОТЛАДКА/ ОТЛАДКА** или иконкой  на панели управления. При этом на экране появится окно отладчика.

Заходим в меню отладчика **ФАЙЛ/ ОСТАНОВИТЬ ПРИЛОЖЕНИЕ** или нажав на иконку **STOP**, тем самым остановив приложение, которое было загружено в контроллер раньше. После остановки ранее загруженного приложения при помощи команды **ФАЙЛ/ ЗАГРУЗИТЬ** загружаем свою программу. (Предварительно можно провести отладку программы в режиме симуляции, нажав , или, выбрав **ОТЛАДКА/ СИМУЛЯЦИЯ**)

Убедитесь в наличие воды в ёмкостях установки.

Подайте напряжение на установку «Гидравлический объект».

Теперь можно проводить отладку созданной программы: подбирать коэффициенты, имитировать дискретные сигналы, анализировать работу программы.

5 Создание программы на языке структурированного текста ST

ST – это структурный язык высокого уровня разработанный для процессов автоматизации. Этот язык, в основном, используется для создания сложных процедур, которые не могут быть легко выражены при помощи графических языков.

5.1 Основной синтаксис ST

Программы для программирования на языке ST пишутся полностью на языке ST. Имена, используемые в исходном тексте ST-программ (идентификаторы переменных, константы, ключевые слова), разделяются активными разделителями (пробелы, подстановочные знаки и табуляции) или активными разделителями со специальным значением (например, разделитель > обозначает сравнение больше, чем). Комментарии могут быть добавлены к

тексту, конечно, в зависимости от программиста. Комментарии должны начинаться с (*) и заканчиваться *. Каждый комментарий должен заканчиваться точкой с запятой.

Основные операторы языка ST:

- оператор присвоения :=
- 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.

Неактивные разделители можно свободно вставлять между активными разделителями, константами и идентификаторами. В отличие от неформальных языков, таких как PL, неактивные разделители могут быть вставлены в любое место программы. Чтобы улучшить читаемость программ, при использовании неактивных разделителей следует придерживаться следующих правил:

- Не пишите более одного оператора в строке;
- Используйте табуляцию для сдвига сложных операторов;
- Вводите комментарии для улучшения читаемости строк и параграфов.

5.2 Выражения и скобки

Программы ST содержат ряд выражений с операторами и операндами между ними, что позволяет вычислять константы и переменные. При выполнении операции между переменными или константами операнды выражения должны быть одного типа.

5.3 Вызов функциональных блоков

Экземпляр функционального блока должен быть вызван в словаре, где ему присваивается имя.

Имя экземпляра функционального блока вызывает функциональный блок из библиотеки ISaGRAF или из пользовательской библиотеки и возвращает возвращаемые параметры.

Синтаксис: (*вызвать функциональный блок*)

<blockname> (<p1>, <p2>...);

(*получить возвращаемые параметры*)

<result> := <blockname>. <ret_param1>;

...

<result> := <blockname>. <ret_paramN>;

5.4 Пример создания функционального блока OGR на языке ST

(*вызов функционального блока ST программой*)

(*объявить экземпляр функционального блока в словаре*) (Рисунок 16)

(*объявить переменные в словаре*)

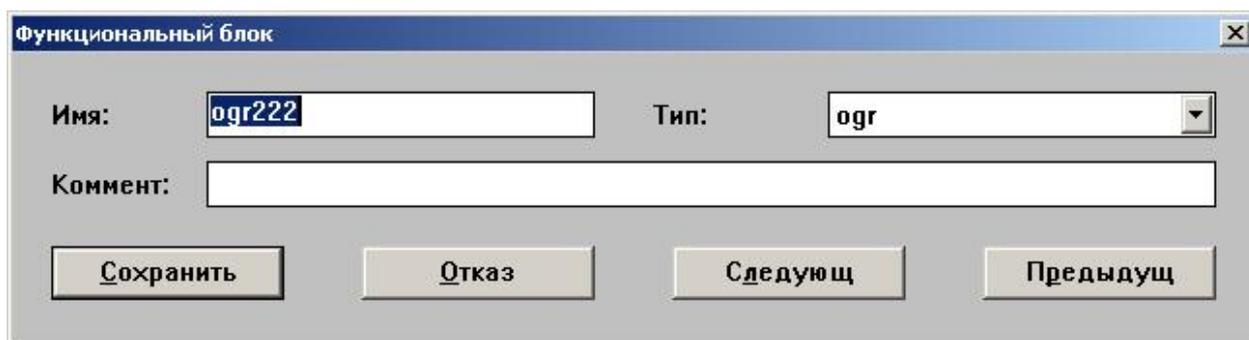


Рисунок 16 – Объявление функционального блока в словаре

(*вызываем функциональный блок из языка ST*)

ogr222(Xin, Xmax1, Xmin1);

(*получаем возвращаемые параметры*)

Dmax1:= ogr222.Dmax; (*1, если X > Xmax*)

Dmin1:= ogr222.Dmin; (*1, если X < Xmin*)

5.5 Пример создания элемента ИЛИ

(*Присваиваем переменной ZVUK результат логического элемента ИЛИ*)

ZVUK:= Dmax1 OR Dmin1;

Операторы :=, OR выбираются из вспомогательного окна «Ключевые слова».

В результате мы получаем реализацию аварийной сигнализации с возможностью вывода затем переменной ZVUK на блок дискретного вывода.

Описанные блоки (элементы) программы на языке ST эквиваленты схеме на языке FBD, приведённой на рисунке 17.

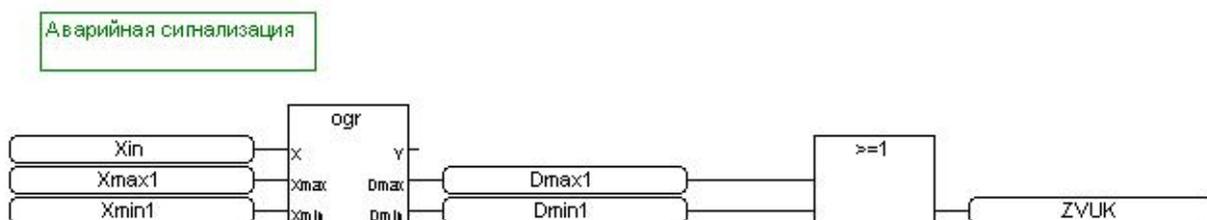


Рисунок 17 – Аналог программы на FBD

6 Написание программы для САР уровня на языке ST

6.1 Создание блоков ввода-вывода

Описываем блок аналогового ввода-вывода:

(*В словаре в «FB экземпляры» объявляем блок craio и присваиваем ему имя craio111*)

(*Вызываем функциональный блок craio*)

```
craio111(addr1, phys, AO_1, AO_2, AO_3, AO_4);
```

```
AI2:= craio111.AI4; (*положение исполнительного механизма*)
```

```
Xin:= craio111.AI5; (*уровень в E2*)
```

Описываем блок дискретного ввода:

(*В словаре в «FB экземпляры» объявляем блок crdin16 и присваиваем ему имя crdin_16*)

(*Вызываем функциональный блок crdin16*)

```
crdin_16(addr2);
```

```
L3:= crdin_16.D2;
```

```
L2:= crdin_16.D3;
```

```
L1:= crdin_16.D4;
```

Аналогично описываем блок дискретного вывода:

crdout_16(addr3, rez, Control_H3, sign_m, sign_b, Dmax, Dmin, ZVUK, rez, rez, rez, rez, rez, rez, rez, rez);

6.2 Реализация импульсного ПИД-регулятора

Описываем блок IMP_PID:

(*В словаре в «FB экземпляры» объявляем блок im_pid и присваиваем ему имя im_pid111*)

(*Вызываем функциональный блок im_pid*)

im_pid111(zadanie, Km_zdn, Tf_zdn, Csb, Cdb, Vdb, Xin, Km_In, Tf_In, Xdlt, Kp, Ti, Kd, Tm, Cruch);(*описали блок imp_pid*)

Yout:= im_pid111.Yout; (*Выход блока imp_pid*)

Описываем блок improut:

(*В словаре в «FB экземпляры» объявляем блок improut и присваиваем ему имя improut111*)

(*Вызываем функциональный блок improut*)

improut111(Yout, Tmin, Tlb, Tlm);

Db:= improut111.Db; (*выход блока improut Db*)

Dm:= improut111.Dm; (*выход блока improut Dm*)

6.3 Описание Ручного режима управления и выходов регулятора

Сначала опишем переменные a и b – выходы элементов &, для того чтобы использовать их значения при реализации Ручного режима управления:

a:= Db_Ruch AND Cruch AND NOT b; (*создаём переменную a на выходе & - для реализации сигнала Больше*)

b:= Dm_Ruch AND Cruch AND NOT a; (*создаём переменную b на выходе & - для реализации сигнала Меньше*)

С помощью элементов AND и OR реализуем выдачу сигналов Больше и Меньше.

$$\text{sign_b} := (\text{Db_Ruch AND Cruch AND NOT b}) \text{OR Db};$$
$$\text{sign_m} := (\text{Dm_Ruch AND Cruch AND NOT a}) \text{OR Dm};$$

6.4 Реализация предупредительной и аварийной сигнализации

Описываем блок org – реализация предупредительной световой сигнализации.

Как создать данный блок было показано выше.

$$\text{ogr111}(\text{Xin}, \text{Xmax}, \text{Xmin});$$
$$\text{Dmax} := \text{ogr111.Dmax}; (*1, \text{ если } X > \text{Xmax}*)$$
$$\text{Dmin} := \text{ogr111.Dmin}; (*1, \text{ если } X < \text{Xmin}*)$$

Описываем блок org2 – реализация аварийной звуковой сигнализации:

$$\text{ogr222}(\text{Xin}, \text{Xmax1}, \text{Xmin1});$$
$$\text{Dmax1} := \text{ogr222.Dmax}; (*1, \text{ если } X > \text{Xmax}*)$$
$$\text{Dmin1} := \text{ogr222.Dmin}; (*1, \text{ если } X < \text{Xmin}*)$$

Создаём блок ИЛИ – звуковая сигнализация при достижении уровня минимальных или максимальных значений:

$$\text{ZVUK} := \text{Dmax1 OR Dmin1};$$

6.5 Реализация алгоритма работы насоса НЗ

Создаём блок «Исключающее ИЛИ»:

$$\text{Q_ili} := \text{Rezhim_H3 XOR NOT L1};$$

Описываем блок tri:

(*В словаре в «FB экземпляры» объявляем блок tri и присваиваем ему имя tri111*)

(*Вызываем функциональный блок tri*)

tri11(L3, Q_ili);

Dtri:= tri11.D; (*Выход блока tri*)

Создаём фрагмент (конец) программы для управления насосом НЗ:

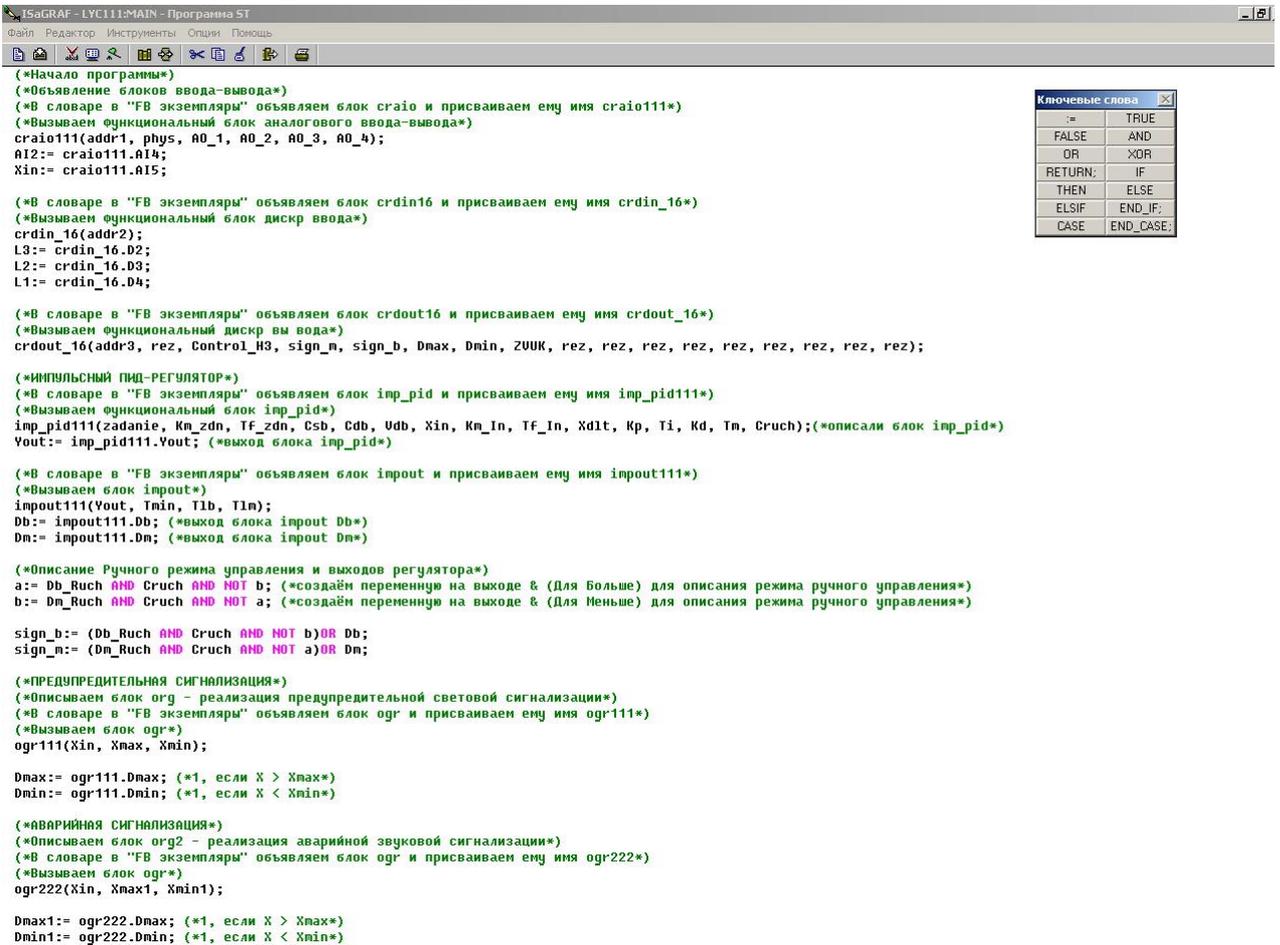
Control_НЗ:= (Ruch_НЗ AND Rezhim_НЗ) OR Dtri;

(*Объявляем переменные*)

(*Конец программы*)

Для загрузки и отладки программы проделываем операции аналогично приведенным ранее, при создании программы на языке FBD.

6.6 Получите полную программу работы



```
(*Начало программы*)
(*Объявление блоков ввода-вывода*)
(*В словаре в "FB экземпляры" объявляем блок craio и присваиваем ему имя craio111*)
(*Вызываем функциональный блок аналогового ввода-вывода*)
craio111(addr1, phys, AO_1, AO_2, AO_3, AO_4);
AI2:= craio111.AI4;
Xin:= craio111.AI5;

(*В словаре в "FB экземпляры" объявляем блок crdin16 и присваиваем ему имя crdin_16*)
(*Вызываем функциональный блок дискр ввода*)
crdin_16(addr2);
L3:= crdin_16.D2;
L2:= crdin_16.D3;
L1:= crdin_16.D4;

(*В словаре в "FB экземпляры" объявляем блок crdout16 и присваиваем ему имя crdout_16*)
(*Вызываем функциональный дискр вы вода*)
crdout_16(addr3, rez, Control_НЗ, sign_n, sign_b, Dmax, Dmin, ZUOK, rez, rez, rez, rez, rez, rez, rez, rez, rez);

(*ИМПУЛЬСНЫЙ ПИД-РЕГУЛЯТОР*)
(*В словаре в "FB экземпляры" объявляем блок imp_pid и присваиваем ему имя imp_pid111*)
(*Вызываем функциональный блок imp_pid*)
imp_pid111(zadanie, Km_zdn, TF_zdn, Csb, Cdb, Udb, Xin, Km_In, TF_In, Xdt, Kp, Ti, Kd, Tm, Cruch);(*описали блок imp_pid*)
Yout:= imp_pid111.Yout; (*выход блока imp_pid*)

(*В словаре в "FB экземпляры" объявляем блок inprout и присваиваем ему имя inprout111*)
(*Вызываем блок inprout*)
inprout111(Yout, Tmin, Tlb, Tln);
Db:= inprout111.Db; (*выход блока inprout Db*)
Dn:= inprout111.Dn; (*выход блока inprout Dn*)

(*Описание Ручного режима управления и выходов регулятора*)
a:= Db_Ruch AND Cruch AND NOT b; (*создаём переменную на выходе & (Для Больше) для описания режима ручного управления*)
b:= Dn_Ruch AND Cruch AND NOT a; (*создаём переменную на выходе & (Для Меньше) для описания режима ручного управления*)

sign_b:= (Db_Ruch AND Cruch AND NOT b)OR Db;
sign_n:= (Dn_Ruch AND Cruch AND NOT a)OR Dn;

(*ПРЕДУПРЕДИТЕЛЬНАЯ СИГНАЛИЗАЦИЯ*)
(*Описываем блок ogr - реализация предупредительной световой сигнализации*)
(*В словаре в "FB экземпляры" объявляем блок ogr и присваиваем ему имя ogr111*)
(*Вызываем блок ogr*)
ogr111(Xin, Xmax, Xmin);

Dmax:= ogr111.Dmax; (*1, если X > Xmax*)
Dmin:= ogr111.Dmin; (*1, если X < Xmin*)

(*АВАРИЙНАЯ СИГНАЛИЗАЦИЯ*)
(*Описываем блок ogr2 - реализация аварийной звуковой сигнализации*)
(*В словаре в "FB экземпляры" объявляем блок ogr и присваиваем ему имя ogr222*)
(*Вызываем блок ogr*)
ogr222(Xin, Xmax1, Xmin1);

Dmax1:= ogr222.Dmax; (*1, если X > Xmax*)
Dmin1:= ogr222.Dmin; (*1, если X < Xmin*)
```

=	TRUE
FALSE	AND
OR	XOR
RETURN:	IF
THEN	ELSE
ELSIF	END_IF;
CASE	END_CASE;

```

(*АВАРИЙНАЯ СИГНАЛИЗАЦИЯ*)
(*Описываем блок ogr2 - реализация аварийной звуковой сигнализации*)
(*В словаре в "FB экземпляры" объявляем блок ogr и присваиваем ему имя ogr222*)
(*Вызываем блок ogr*)
ogr222(Xin, Xmax1, Xmin1);

Dmax1:= ogr222.Dmax; (*1, если X > Xmax*)
Dmin1:= ogr222.Dmin; (*1, если X < Xmin*)

(*Создаём блок "ИЛИ" - звуковая сигнализация при достижении MIN и MAX аварийных значений*)
ZUUK:= Dmax1 OR Dmin1;

(*УПРАВЛЕНИЕ РАБОТОЙ НАСОСА НЗ*)
(*Создаём блок "Искл ИЛИ"*)
Q_ili:= Rezhim_N3 XOR NOT L1;

(*Реализуем алгоритм работы насоса НЗ*)
(*В словаре в "FB экземпляры" объявляем блок tri и присваиваем ему имя tri111*)
(*Вызываем блок ogr*)
tri111(L3, Q_ili);
Dtri:= tri111.D; (*выход блока tri*)

(*Создаём конец проги для управления НЗ: блоки "И", "ИЛИ"*)
Control_N3:= (Ruch_N3 AND Rezhim_N3) OR Dtri;

(*Объявляем переменные в словаре*)
(*Конец программы*)

```

Рисунок 18 - Диаграмма полной программы горизонтальной регулировки, отредактированная в ST

7 Задание на лабораторную работу

1. В системе ISaGRAF составьте программу, обеспечивающую регулирование уровня, управление насосом и сигнализации на заданном преподавателем языке программирования (FBD, ST). Необходимо реализовать:

- регулирование уровня в ёмкости E2 на основе импульсного ПИД-регулятора
(imp_pid);
 - срабатывание световой сигнализации при достижении уровня жидкости в ёмкости E2 предупредительных порогов: min 15%, max 50%;
 - срабатывание звуковой сигнализации при достижении уровня жидкости в ёмкости E2 аварийных порогов: min 10%, max 55%;
 - алгоритм работы насоса НЗ: включение – при достижении уровня жидкости в ёмкости E3 верхнего уровня (L3), отключение – при достижении нижнего уровня (L1).
2. Проверить программы на наличие ошибок.
3. Проверить функциональность программы при условии, что запрос не содержит ошибок.

4. Подготовить отчет о работе, который должен содержать

- цель работы;
- структурную схему лабораторного комплекса;
- текст программы;
- ответы на контрольные вопросы.
- вывод

8 Методические указания по выполнению отдельных пунктов

задания

Перед выполнением задания ознакомьтесь с конструкцией гидравлического объекта согласно п. 1.

Создайте новый проект на основе п. 2 и п.3 данного методического пособия. Обратите внимание, что в секции BEGIN созданы две программы: отдельно для описания блоков ввода-вывода и основная программа для САР уровня.

При написании программы не забывайте объявлять переменные в словаре.

В процессе создания программы, а также после ее написания, выполняйте проверку её на синтаксис. Для этого нужно зайти в меню **ФАЙЛ/ ПРОВЕРИТЬ** или нажать на кнопку . Если ошибок нет, то выходим из генератора кода.

При конфигурировании блоков ввода-вывода задействуйте следующие входы (выходы):

Аналоговые:

AI4 – сигнал положения исполнительного механизма;

AI5 – сигнал значения уровня в емкости E2;

Дискретные:

DI2 – уровень L3;

DI3 – уровень L2;

DI4 – уровень L1;

DO2 – сигнал на включение/ отключение насоса Н3;

DO3 – сигнал «Меньше»;

DO4 – сигнал «Больше»;

DO5 – сигнализация по max;

DO6 – сигнализация по min;

DO7 – звуковая сигнализация;

Также при конфигурировании блоков ввода-вывода необходимо связать свободные входы/ выходы между собой, либо, применив свободную переменную «REZ»(резерв), как показано в п. 3.1.

9 Контрольные вопросы

6. Приведите краткое описание гидравлического объекта?
7. Можно ли описать блоки ввода-вывода в основной программе?
8. Для чего используется OPC-сервер?
9. Когда используется язык программирования ST?
10. Как описывается функциональный блок на языке ST?

**ПРИЛОЖЕНИЕ Б.
МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

Ли Яньчэн

**СОЗДАНИЕ ПРОГРАММЫ ВИЗУАЛИЗАЦИИ ПРОЦЕССА
УПРАВЛЕНИЯ УСТАНОВКОЙ «ГИДРАВЛИЧЕСКИЙ ОБЪЕКТ»
В ПАКЕТЕ MasterSCADA**

Содержание

Цель работы	1
1 Понять, как работает пакет Master SCADA и типы систем SCADA	1
2 Функции MasterScada и соответствующая информация	3
3 Принципиальная схема лабораторного оборудования для гидравлического контроля	6
4 Создание визуального представления программы в пакете MasterScada	8
4.1 Создайте новый проект	8
4.2 Создание основной мнемосхемы	12
4.3 Создание экспериментальной программы памяти	15
4.4 Создание индикации уровня в емкости E2	16
4.5 Создание изображения регулирующего клапана с импульсным управлением	17
4.6 Создание команды «Задание»	19
4.7 Создание Окна Управления «Настройки PID-регулятора»	19
4.8 Создание индикации уровня в E3 и управления насосом H3.	21
5 Создание Тренда	23
6 Создание окна «Контроль и управление в E2»	24
7 Создание таблицы	28
8 Создание графика	29
9 Запуск программ визуализации	29
10 Контрольные вопросы	31

Цель работы: Освоить навыки работы с пакетом Master SCADA, используя программы программирования в ISaGRAF, подключить пакет ISaGRAF к пакету Master SCADA для взаимодействия данных через пакет OPC, а затем создать программы визуализации для управления, регулирования и вывода сигналов.

1 Понять, как работает пакет Master SCADA и типы систем SCADA

Системы SCADA часто используются при разработке программного обеспечения и при проектировании начального оборудования автоматического управления, что уменьшает условия разработки программного обеспечения и увеличивает вероятность правильного функционирования программы, обеспечивая высокое качество разработки и эксплуатации.

Используемые сегодня системы SCADA состоят из трех основных структурных компонентов, как показано на рисунке 1.1.

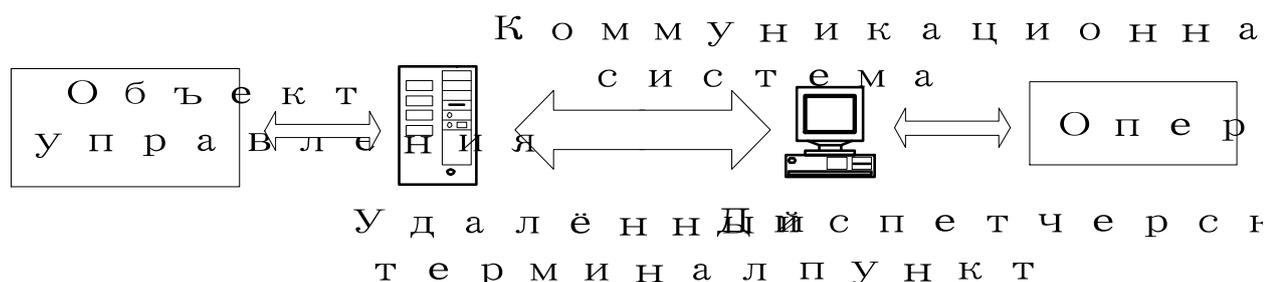


Рисунок 1.1 - Основные структурные компоненты SCADA системы

Удаленный терминальный блок (RTU) выполняет соответствующие задачи управления быстро и в реальном времени.

Главный терминальный блок (MTU) выполняет расширенную обработку данных в реальном времени и эффективно и обеспечивает интерфейс между человеком и системой.

Система связи (CS) необходима для передачи данных с удаленных точек (станций, терминалов) на центральный интерфейс оператора и для передачи сигналов управления в центр управления.

К SCADA-системам предъявляются следующие *основные требования*:

- надежность системы (технологическая и функциональная);
- безопасность управления;
- точность обработки и представления данных;
- простота расширения системы.

Система SCADA - это чрезвычайно многофункциональная система программирования с функциональным набором всех программных пакетов, которые могут быть индивидуально запрограммированы и смоделированы, в комплекте с определением и реализацией программы. Ниже перечислены основные функции и инструменты системы SCADA:

- 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.;
- возможность работы прикладной системы с наборами параметров, рассматриваемых как единое целое (ресурсы, или установки).

Программные пакеты SCADA сегодня очень широко используются, например, в банках или компаниях с большим количеством устройств системы SCADA. Ниже приводится сравнение некоторых наиболее популярных систем SCADA, используемых на Западе и в России (табл. 1.1).

Таблица 1.1 - Знакомство с распространенными программными пакетами SCADA, доступными сегодня

Фирма разработчик	Страна	Фирма распространитель в России	Название SCADA- пакета
AdAstra Res. Group	Россия	AdAstra Res. Group	TRACE MODE
Intellution	США	Индасофт	FIX
Ci Technologies	Австралия	РТСофт	Citect
Wonderware	США	РТСофт	InTouch
Iconics	США	ПроСофт	Genesis
Инсат	Россия	Инсат	VNS MasterSCADA

Самая большая разница между вышеперечисленными системами SCADA заключается только в сложности программирования и стоимости разработки программного обеспечения, кроме того, компаниям, использующим системы SCADA, необходимо учитывать возврат инвестиций.

2 Функции MasterScada и соответствующая информация

MasterScada - это программный пакет SCADA с функциями моделирования и инструмент разработки системы SCADA, который вместе с сопутствующим программным обеспечением (например, ISaGRAF) позволяет снизить трудозатраты, сократить рабочее время и улучшить работоспособность системы. передача.

В дополнение к поддержке всех стандартов (XML, HTML, ODBC, OLE, COM/DCOM, ActiveX и т.д.) и спецификаций интерфейса, предоставляются все функции, необходимые для взаимодействия внешних приложений и системных данных.

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.

Пользовательский интерфейс MasterScada (рис. 2.1) представляет собой монолитную форму. В зависимости от типа редактируемого файла, окно редактирования открывает страницу настроек или нужный внешний редактор.

Проект состоит из двух частей. Раздел "Система" описывает техническую структуру внедряемой системы. Раздел "Объекты" описывает иерархию атрибутов технических объектов, подлежащих мониторингу, и документирует каждый объект.

Окно проекта состоит из четырех основных частей (рисунок 2.1):

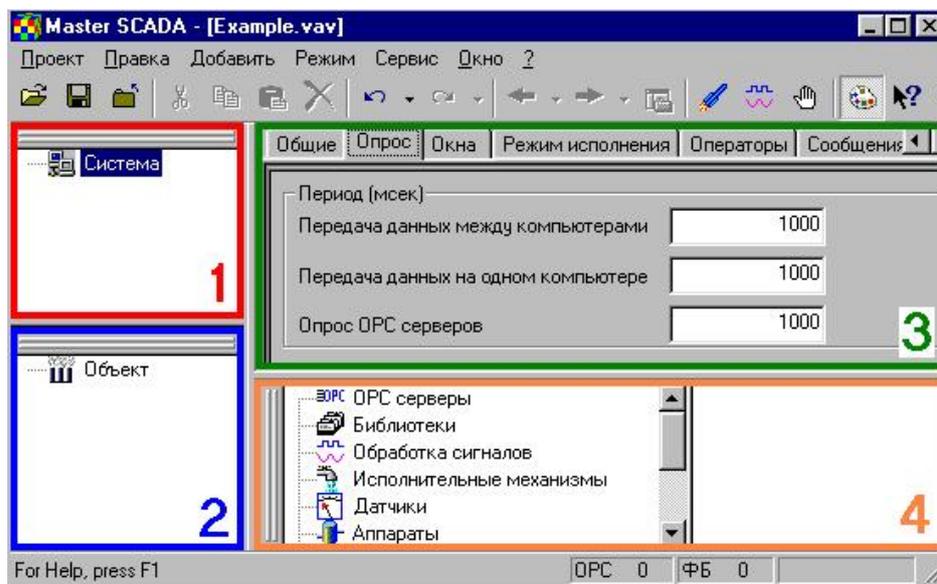


Рисунок 2.1 - Окно проекта MasterSCADA

1) Дерево системы, в котором отображены элементы конфигурации такие, как компьютеры, ОРС серверы, и т. д.

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

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

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

Системное дерево, дерево объектов и палитра изменяют размер, цвет, атрибуты и могут быть настроены, остальное относится к странице свойств.

Встроенные программные средства позволяют:

- Вести обработку данных;
- Составлять сценарии наступления событий;
- Создавать схемы и мнемосхемы, тренды;
- Выдавать сообщения и составлять рапорты;
- Вести архивирование;
- Обмениваться данными по сети;
- Ограничивать права доступа и вести контроль действий оператора;
- Отлаживать созданные программы;
- И повышать надежность работы.

Режимы работы

Существует три возможных варианта работы в режиме исполнения.

- Рабочий режим. Включается командой "Пуск" из меню "Режим", или кнопкой  на панели инструментов. 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. Включается командой ".Отладка." из меню "Режим.", или кнопкой  на панели инструментов. 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. Включается командой ".Имитация." из меню ".Реж.им.", или кнопкой  на панели инструментов. В этом режиме на все входы, не имеющие связей, вместо констант будет подаваться имитация в соответствии с настройками Системы.

3 Принципиальная схема лабораторного оборудования для гидравлического контроля

В рассматриваемую в данной работе систему управления гидравлическим объектом входят:

- гидравлический объект;
- стенд с контроллером;
- персональный компьютер - операторская станция.

На рисунке 3.1 показана схема гидравлической системы управления.

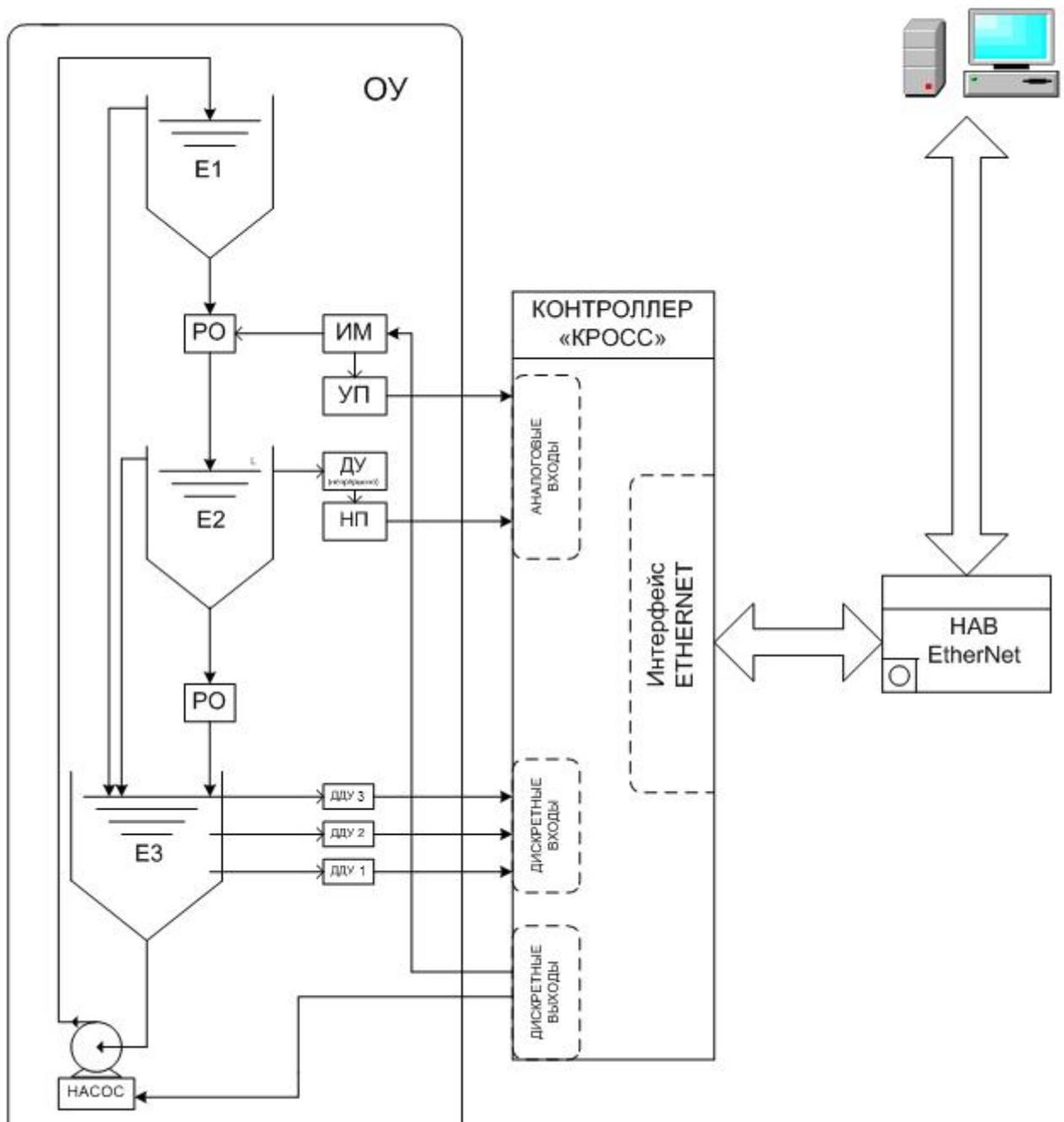


Рисунок 3.1 - Структурная схема лабораторного комплекса

Резервуар для воды E1 является напорным резервуаром. Первоначально вода в бак E1 добавляется сверху, а при проведении эксперимента вода из бака E1 перекачивается в бак E3. Резервуар E1 также оснащен водопроводом, ведущим в резервуар E3 и выходом в резервуар E2, к которому подключен регулирующий орган (РО) - привод (ИМ) с индикатором положения (УП).

В резервуаре E2 находится датчик уровня жидкости (ДУ). При изменении уровня воды в емкости E2 датчик уровня генерирует выходной сигнал, который

преобразуется преобразователем (Н.П) в сигнал в виде тока 0-20 мА, который передается на аналоговый вход контроллера.

Между резервуаром Е2 и резервуаром Е3 установлен клапан для управления уровнем воды в резервуаре Е2 путем увеличения или уменьшения потока воды из Е2 в Е3 во время эксперимента.

Ёмкость Е3 оснащена тремя дискретными датчиками уровня (ДДУ), расположенными на разном уровне, а также насосом, который может управляться либо с панели управления, либо с контроллера, при помощи которого жидкость закачивается в емкость Е1, после чего через регулирующий орган поступает в Е2.

Функции управления и регулирования в системе выполняет промышленный микропроцессорный контроллер КРОСС.

В качестве операторской станции в системе используется компьютер типа IBM PC на базе процессора CELERON 2800, с установленной SCADA системой MasterScada и системой программирования ISaGRAF.

4 Создание визуального представления программы в пакете MasterScada

Совместимость переменных из SCADA-системы с переменными в ISaGRAF после подключения через OPC-сервер. На этой основе можно реализовать обмен данными и выполнить визуальное моделирование.

4.1 Создайте новый проект

Запустите программу MasterScada. Введите имя проекта во всплывающем окне Create Project (рис. 4.1). Если необходимо ввести пароль, то вы сможете получить доступ к проекту (рис. 4.2). Если пароль не нужен, то можно просто нажать ОК.

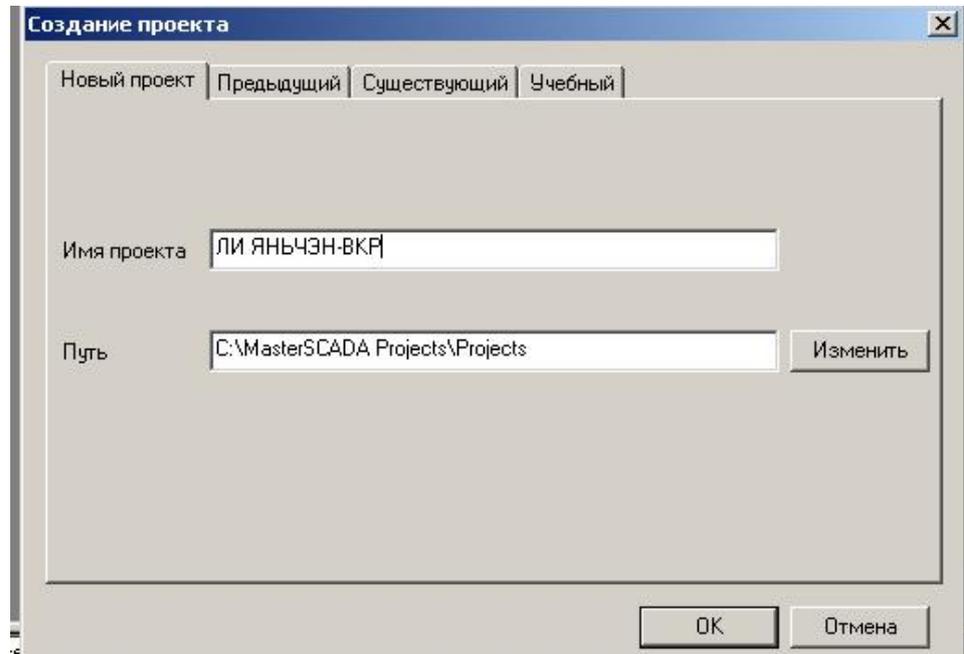


Рисунок 4.1 – Окно создания проекта

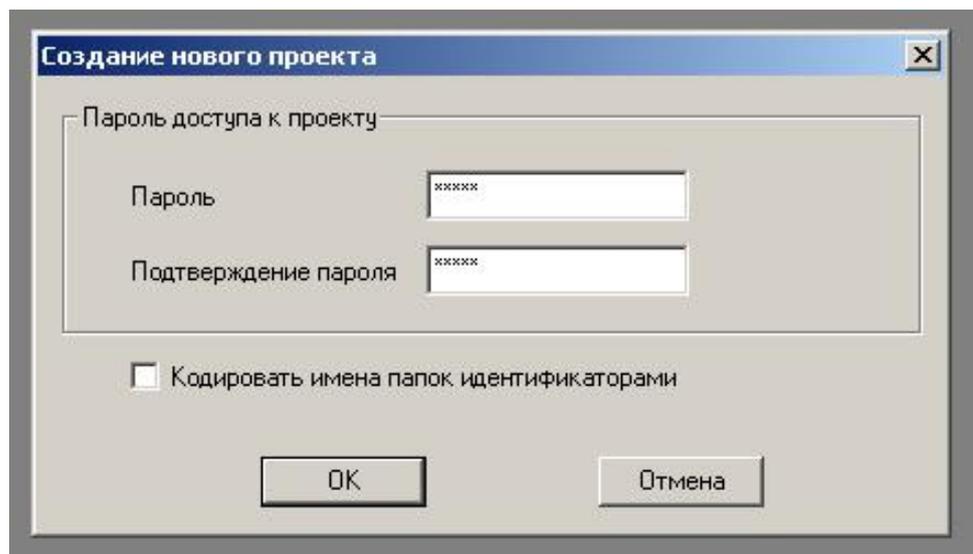


Рисунок 4.2 – Окно установления пароля

В итоге появится окно проекта (рис. 4.3):

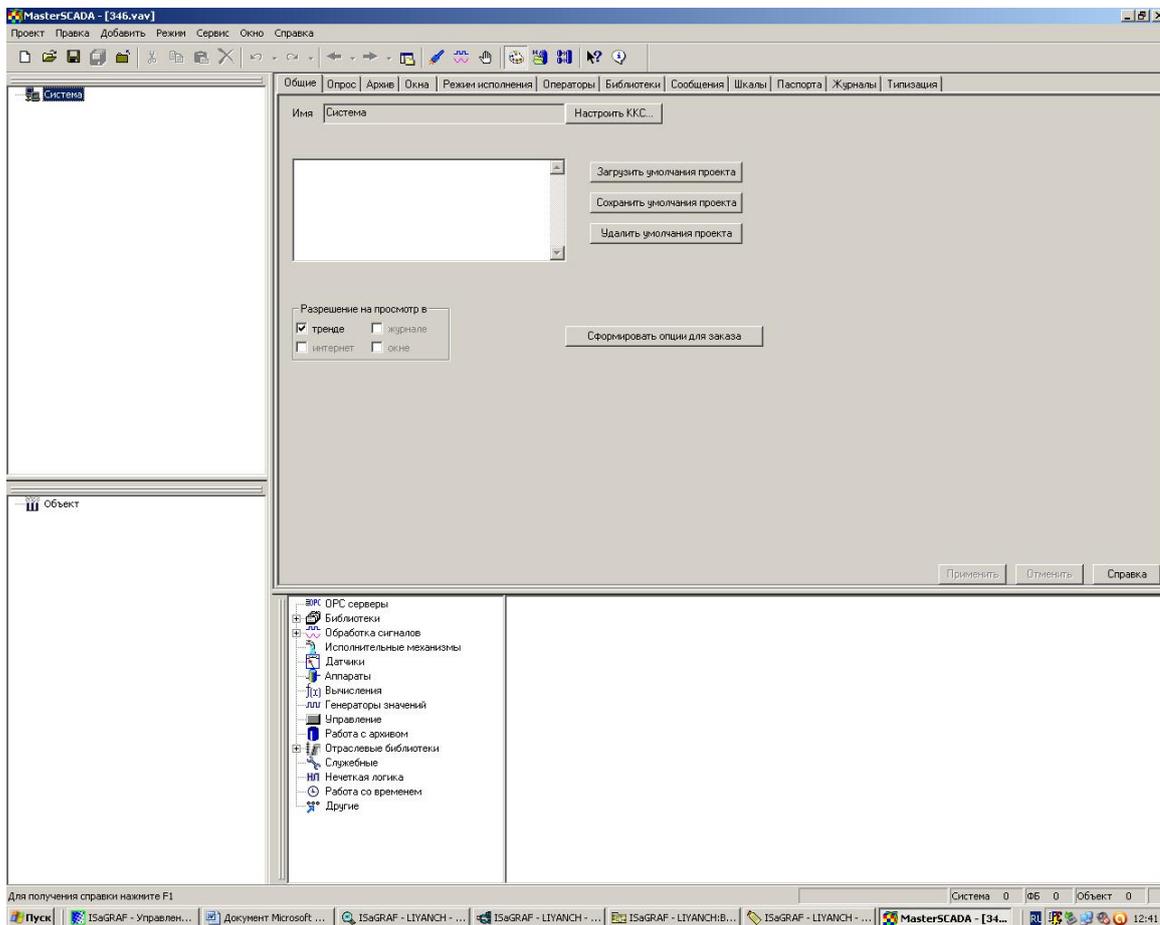


Рисунок 4.3 – Окно проекта

Вы хотите реализовать соединение между контроллером и ISaGRAF. Выберите объект "Система" в дереве системы, щелкните на нем правой кнопкой мыши (ПК) и выберите ВСТАВИТЬ/КОМПЬЮТЕР (рис. 4.4). Измените поле "Имя", введя "titan2", и не забудьте нажать на ПРИМЕНИТЬ.

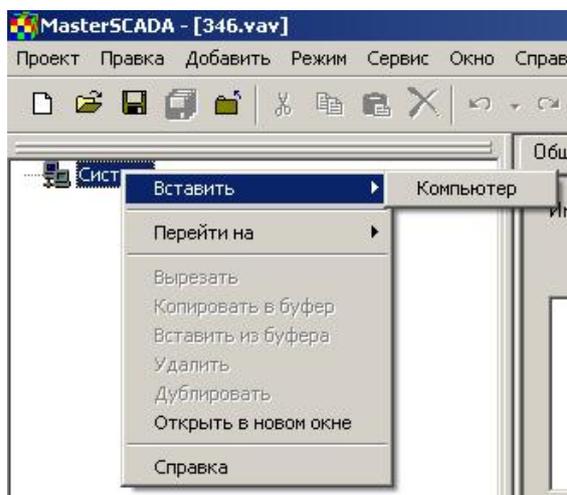


Рисунок 4.4 – Добавление компьютера

Потом нажав ПК на «titan2», добавьте OPC сервер (рис. 4.5).

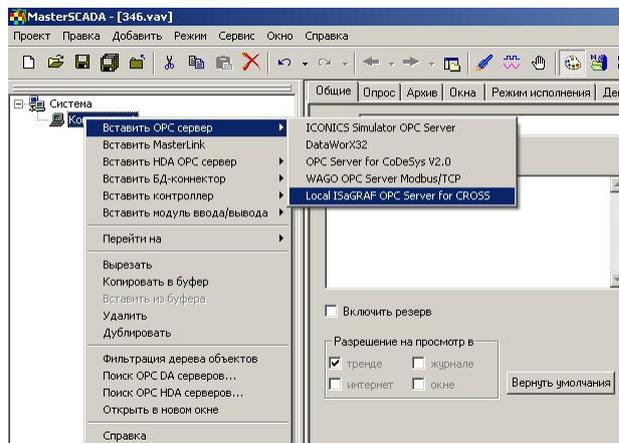


Рисунок 4.5 – Добавление OPC сервера

В появившемся окне установите флажок, как показано на рисунке 4.6, и все переменные, используемые в ISaGRAF, будут доступны в MasterScada. После добавления переменные показаны на рисунке 4.7.

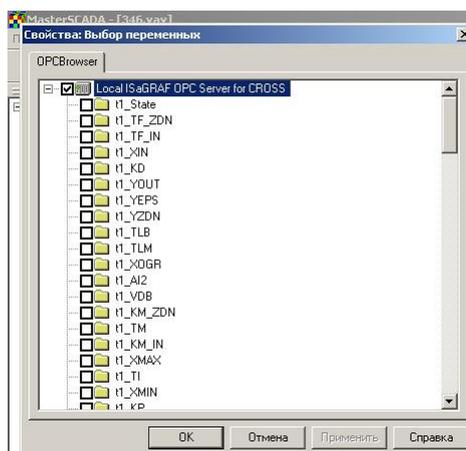


Рисунок 4.6 – Выбор переменных

Существует три основных вида OPC переменных:

- для чтения (отображается в дереве значком выхода ➔);
- для записи (отображается в дереве значком входа ➔);
- для чтения и записи (отображается в дереве значком ↔).

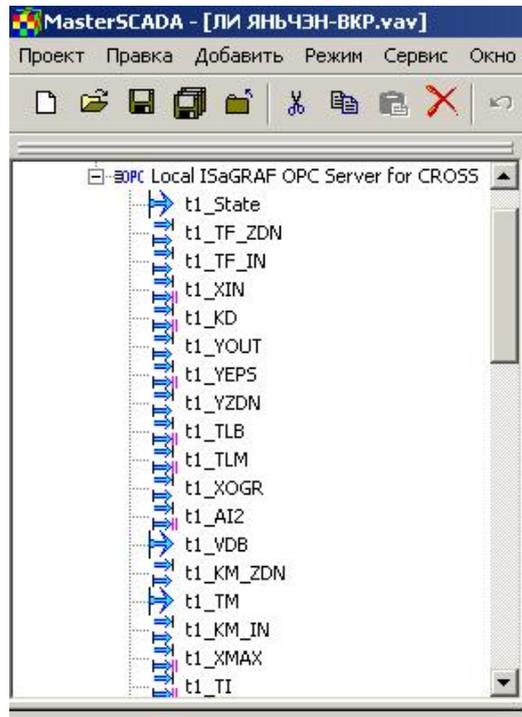


Рисунок 4.7 – Переменные в дереве системы

4.2 Создание основной мнемосхемы

Для создания графической части проекта необходимо использовать переменные, графики, отображения значений и кнопки в дереве частичных объектов.

Графика в основном состоит из трех резервуаров с водой, а также клапанов, водопроводных труб, напорных насосов и т.д. (рис. 4.8).

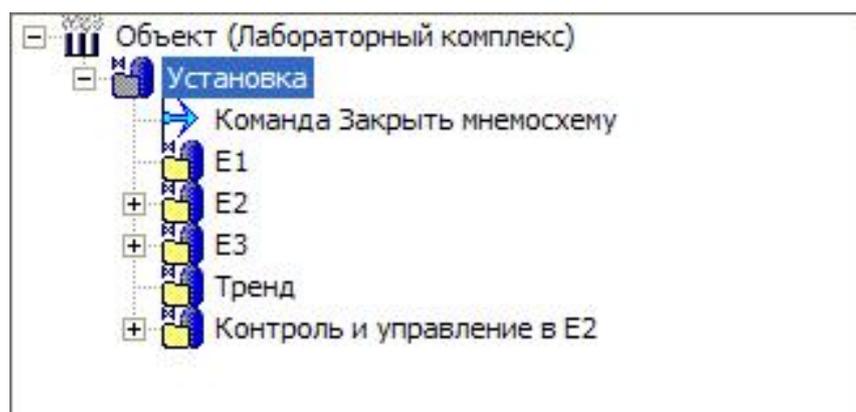


Рисунок 4.8 – Дерево объекта

Создайте и выполните следующие элементы по порядку.

Во-первых, в отображении "Объект" измените имя на "titan2" на странице свойств и сохраните его (рис. 4.9).

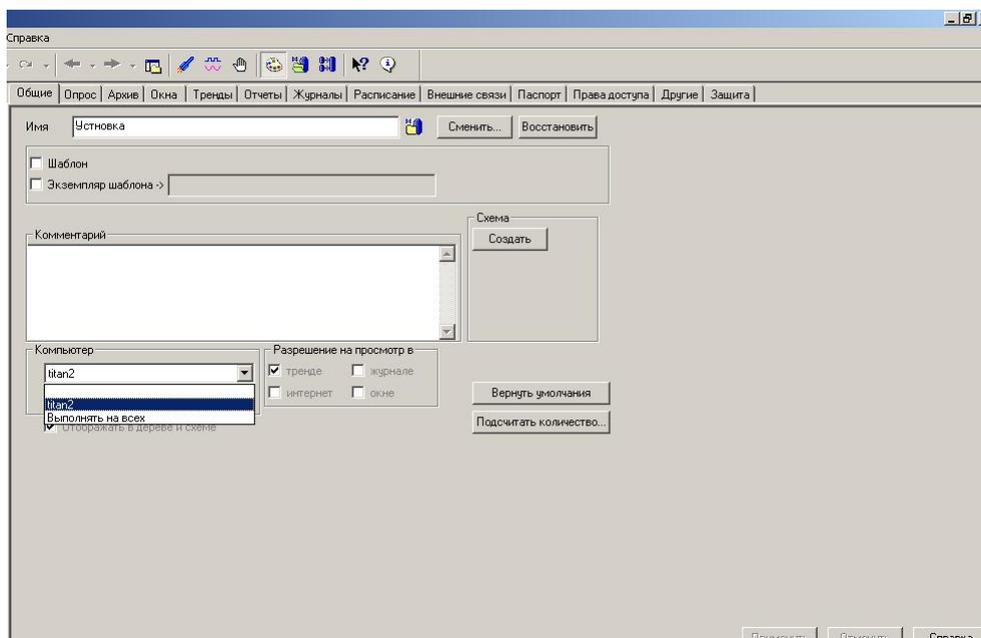


Рисунок 4.9 – Страница свойств элемента «Объект»

Затем нажмите и выберите **ВСТАВИТЬ / ОБЪЕКТ**, как показано на рисунке (Рисунок 4.10).

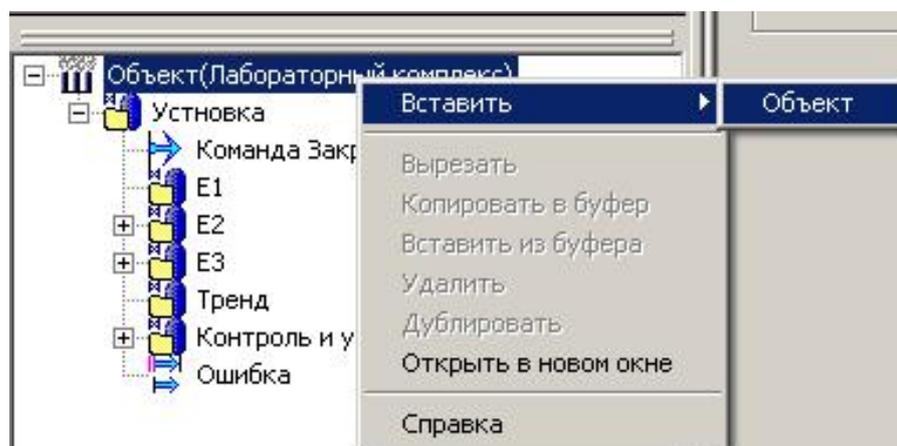


Рисунок 4.10 – Добавление объекта

На его странице свойств выберите компьютер "titan2". Все остальные объекты в проекте будут созданы таким же образом. Для того чтобы визуализатор запускался с отображением этой страницы первой, необходимо создать мнемонику.

Выберите объект «Установка» и на странице свойств перейдите в раздел «Окна», по умолчанию должна быть выбрана мнемоника и нажмите на кнопку **РЕДАКТИРОВАТЬ** (рис. 4.11).

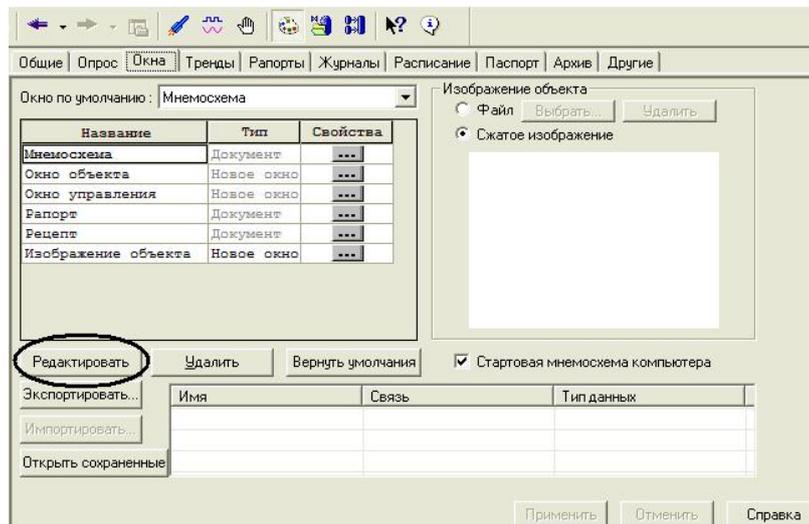


Рисунок 4.11 – Создание мнемосхемы

Затем щелкните по мнемонике, и в окне редактора мнемоники, с правой стороны, появятся различные модули, которые можно использовать (Рисунок 4.12):

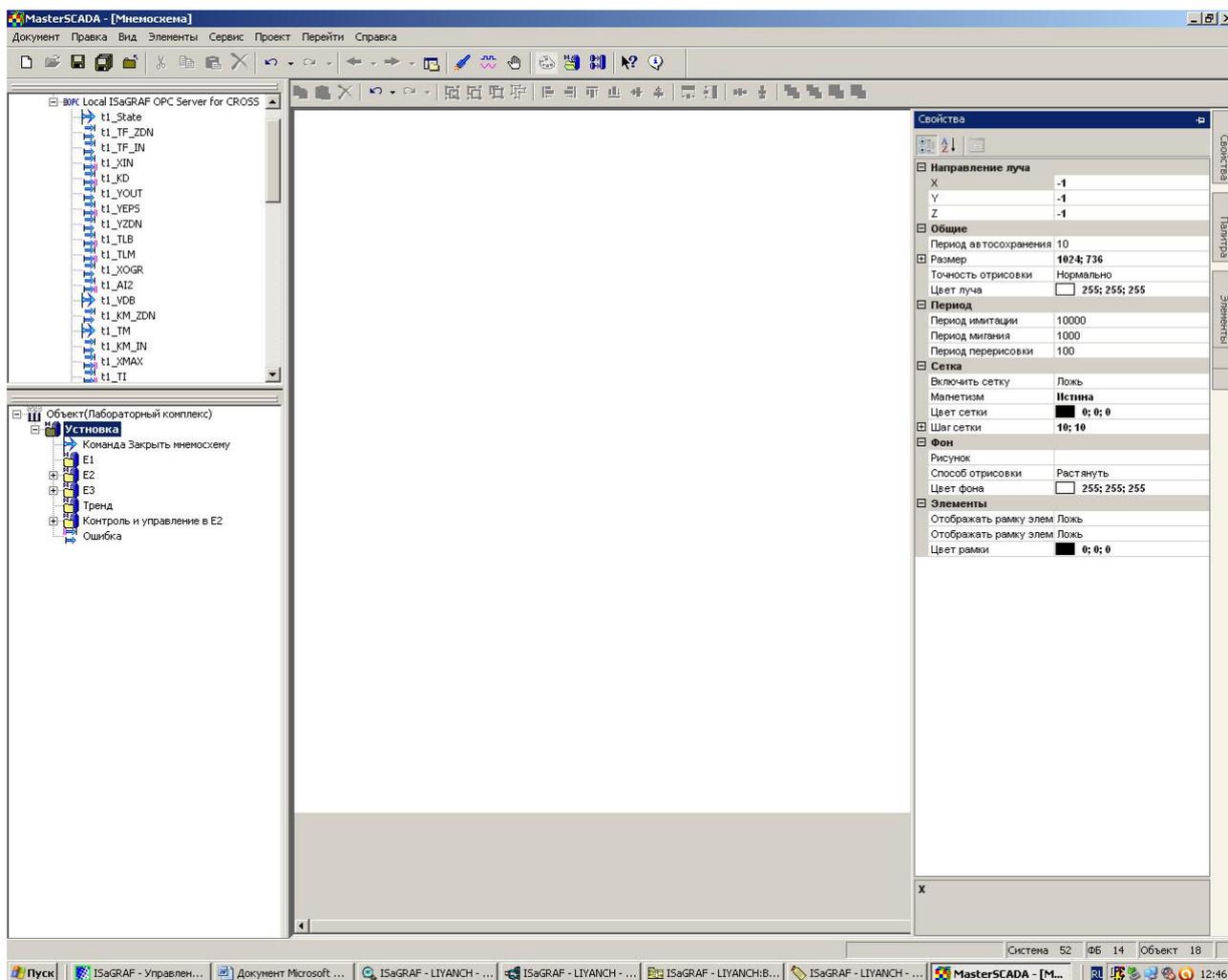


Рисунок 4.12 – Окно редактора мнемосхема

4.3 Создание экспериментальной программы памяти

Создайте резервуары для воды и водопроводные трубы, необходимые для эксперимента.

Создайте три емкости по отдельности (рис. 4.13). Начните с выбора цилиндра в правом элементе, затем перетащите его в программу памяти и выберите подходящий размер.

Также можно щелкнуть правой кнопкой мыши и выбрать свойства, чтобы изменить цвет резервуара, цвет уровня и т.д.

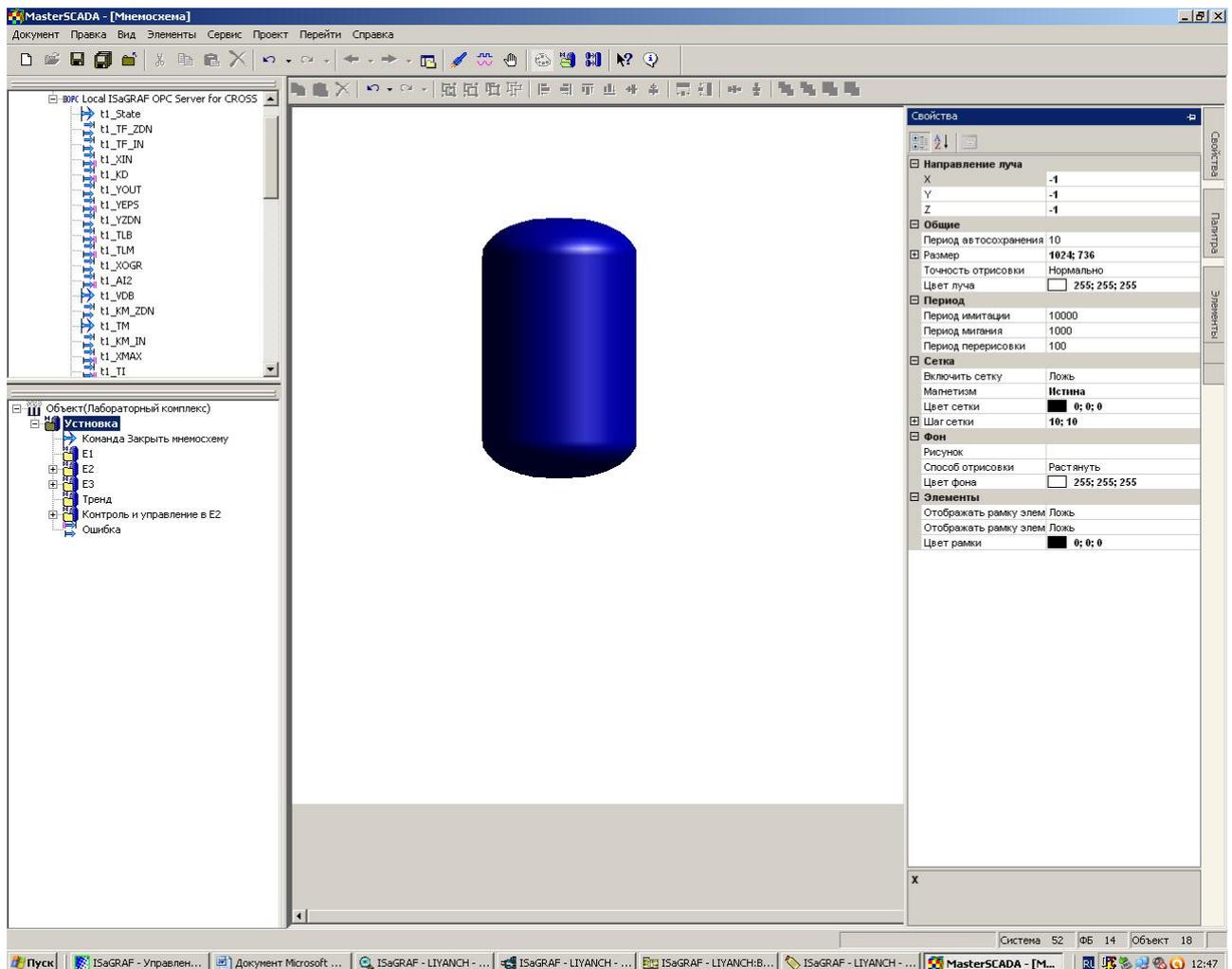


Рисунок 4.13 – Создание объемных элементов

Перетащите **ГРАФИЧЕСКИЕ ПРИМИТИВЫ/ТЕКСТ** с правой стороны внутри программы памяти. напишите необходимый текст в характеристиках и измените цвет шрифта, размер и т.д.

Между тремя резервуарами также необходимо создать соединение водопроводной трубы. Перетащите трубу с правой стороны и измените длину и

диаметр трубы по необходимости. Соединения труб требуют еще одного изогнутого соединения труб. (Рисунок 4.14)

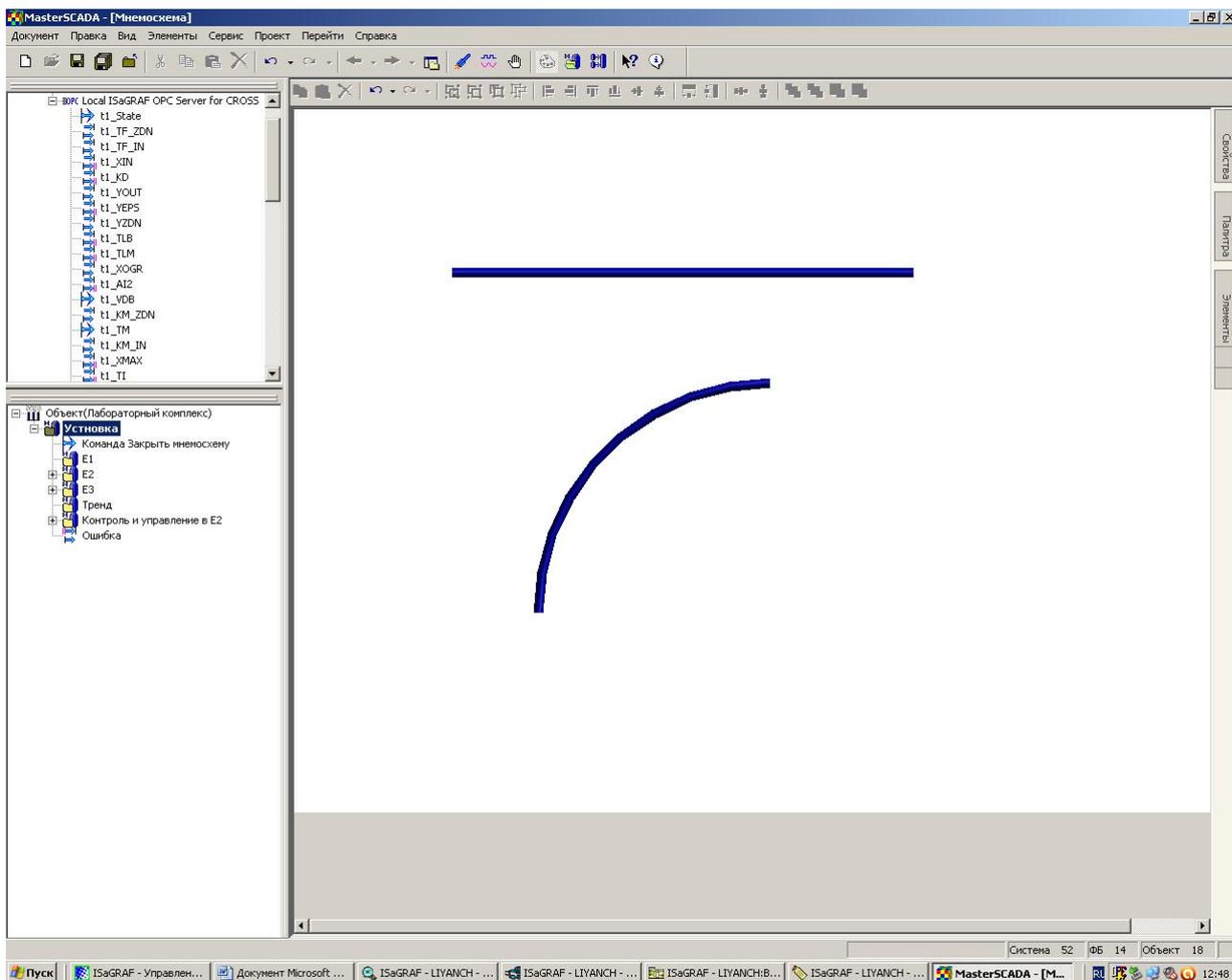


Рисунок 4.14 – Трубопровод

4.4 Создание индикации уровня в емкости E2

Выберите объект "E2", нажмите на РС и выберите INSERT/VALUE. перейдите в Properties, чтобы изменить значение с именем "Level" и типом Analog. Затем перетащите OPC-переменную t1_XIN в дерево объектов. (Рисунок 4.15)

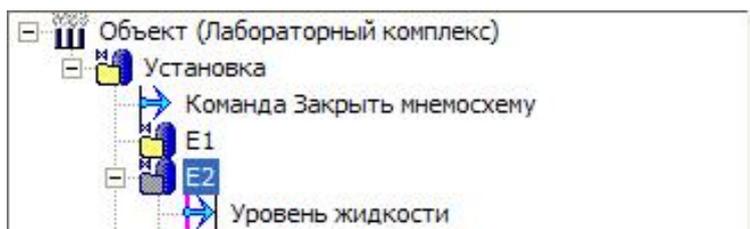


Рисунок 4.15 – Создание значения в дереве объектов

Затем уровень в дереве объектов перетаскивается в программу памяти. Введите характеристики индикатора и вы можете изменить цвет, верхнее и нижнее граничные значения и т.д. (рис. 4.16).



Рисунок 4.16 – Индикатор уровня

4.5 Создание изображения регулирующего клапана с импульсным управлением

Затем необходимо создать управляющий клапан с импульсным управлением. В исходной палитре управляющий клапан перетаскивается в объект E2. Управляющий клапан представляет собой визуальный функциональный блок с функциональными входами и выходами. (Рисунок 4.17).

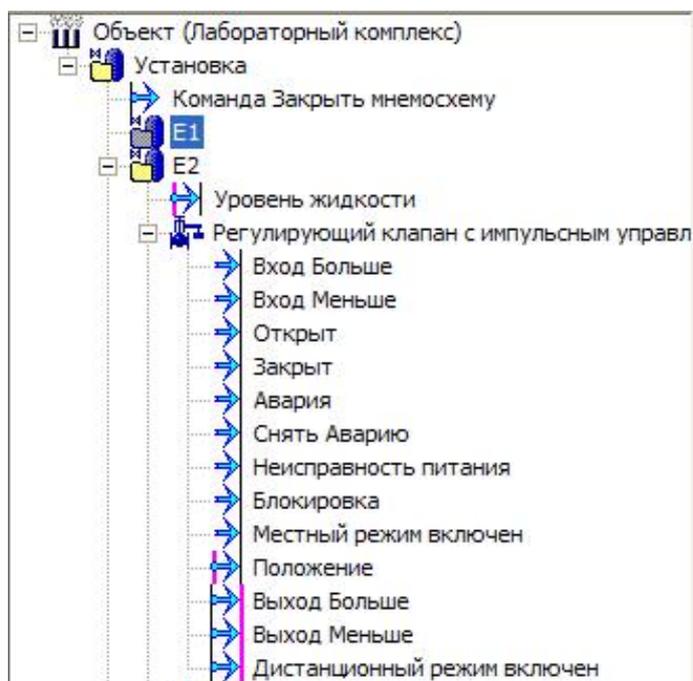


Рисунок 4.17 – Дерево объектов. Регулирующий клапан

Для реализации связи с переменными OPC необходимы следующие соединения.

T1_AI2 – на вход «Положение»

T1_DB_RUCH – «Выход Больше»

T1_DM_RUCH – «Выход Меньше»

T1_CRUCH – Дистанционный режим включен

Над ИМ можно создать блок отображения значений. Это можно сделать непосредственно из дерева системы (t1_AI2. Можно зайти в характеристики, чтобы изменить цвет, размер шрифта и т.д. Рис. 4.18).



Рисунок 4.18 – Визуализация ИМ

В режиме запуска, моделирования или ввода в эксплуатацию окно управления клапаном можно открыть вручную (Рисунок 4.19) где можно выбрать его открытие или закрытие.

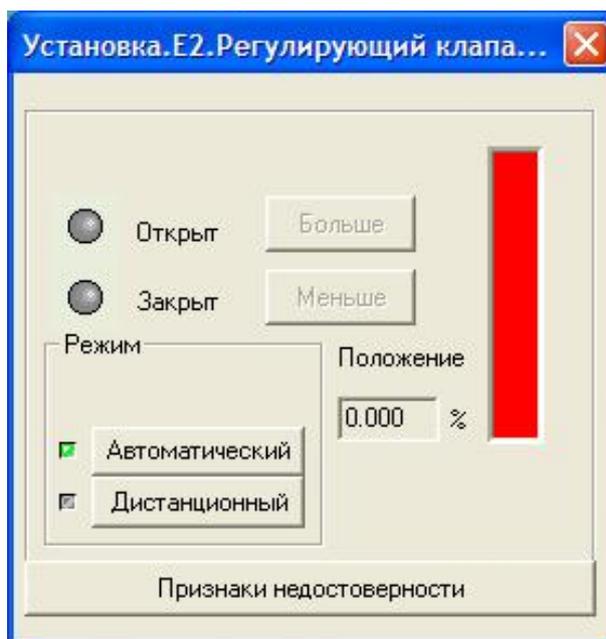
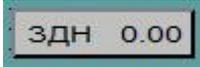


Рисунок 4.19 – Окно управления регулирующим клапаном

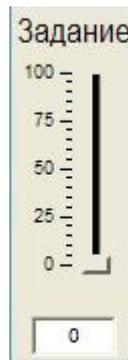
4.6 Создание команды «Задание»

В дереве объектов выберите "Объект E2" и затем выберите ВСТАВИТЬ / КОМАНДА. перетащите t1_ZADANIE из OPC в КОМАНДА.

По желанию вы можете перетащить его из дерева объектов в программу памяти.

Редактируемым значением –  или

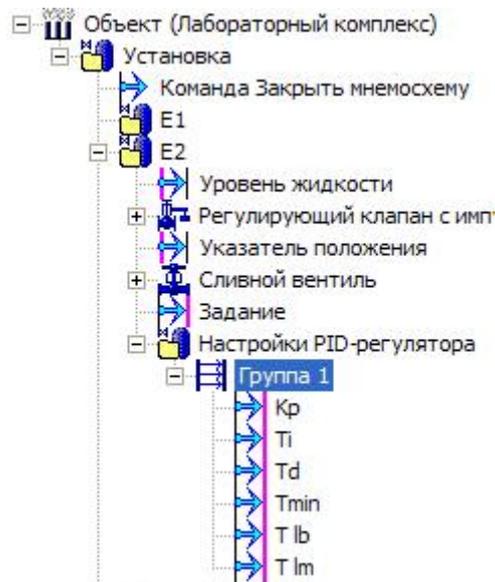
Слайдером –



4.7 Создание Окна Управления «Настройки PID-регулятора»

Для задания коэффициентов PID-регулятора в данном проекте реализовано отдельное окно управления. С помощью этого окна можно производить подстройку коэффициентов в любой момент времени, вызвав это окно нажатием кнопки «Настройки PID» в правом верхнем углу на основной мнемосхеме.

В дереве объектов создайте Объект «Настройки PID-регулятора». Выделите его ПК мышью, выберите ВСТАВИТЬ/ГРУППУ. Это будет группа команд для задания коэффициентов PID-регулятора. Создайте шесть команд и свяжите их с соответствующими OPC-переменными.



В настройках ПИД-регулятора в дереве объектов, в свойствах, в разделе "Окно". Выберите CONTROL Window/NEW Window (Рисунок 4.20).

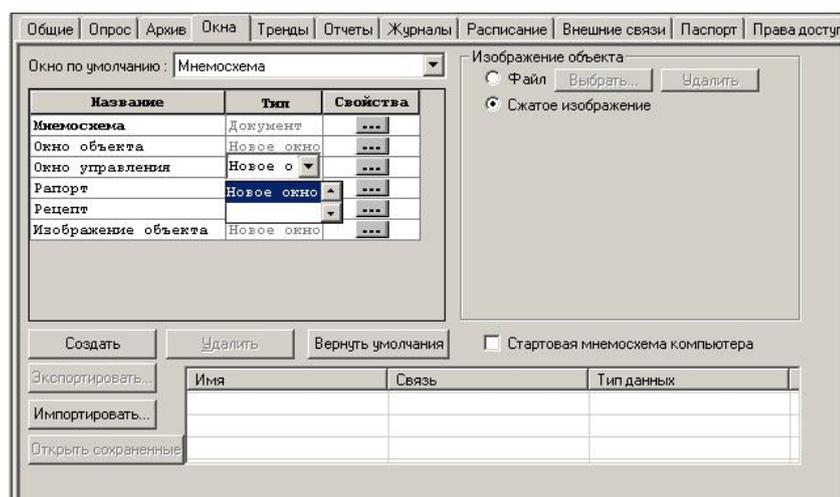


Рисунок 4.20 – Создание окна «Настройки PID-регулятора»

Нажмите кнопку Редактировать. Создайте изображение данного окна с помощью Графических примитивов из раздела Палитра, переместив в него Команды из Объекта «Настройки PID-регулятора».

Теперь на основной мнемосхеме необходимо создать кнопку для вызова окна «Настройки PID-регулятора».

4.8 Создание индикации уровня в Е3 и управления насосом Н3.

Реализация индикации уровня в Е3 и управления насосом Н3 производится аналогично емкости Е2.

В дереве объектов создайте в Объекте «Установка» Объект «Е3». Добавьте значения для трёх дискретных уровней, создайте функциональный блок Насос (рисунок 4.21). Установите связи с ОРС-переменными, перетащите и разместите элементы на мнемосхеме.

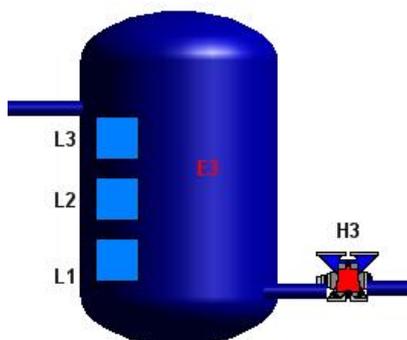
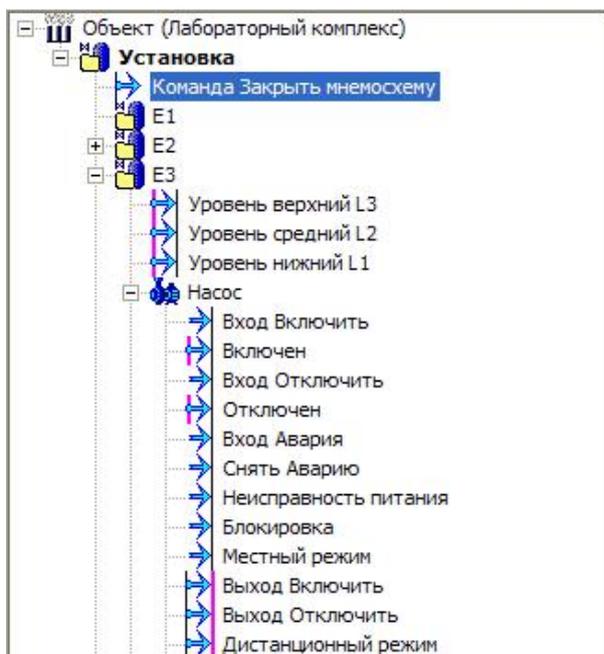


Рисунок 4.21 – Визуализация емкости Е3 и насоса Н3

Для динамизация свойств, а именно, соответствию процента заливки уровням L1, L2, L3, использовались одинаковые изображения трёх цилиндров, каждый со своим процентом заливки (динамизация процента заливки проводится аналогично показанному ранее для Е2 (рисунок 4.22):

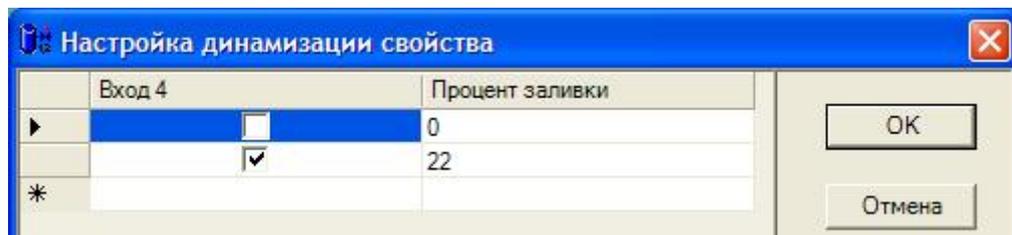


Рисунок 4.22 – Настройка процента заливки для изображения емкости соответствующая уровню L1.

Динамизируемые свойства и источники приведены в таблице 6.

Таблица 6

Элемент	Динамизируемое свойство	Источник	№ входа
Цилиндр 3	Процент заливки	Уровень L1	Вход 4
	Отключен	Уровень L2	Вход 5
Цилиндр 5	Процент заливки	Уровень L2	Вход 5
	Отключен	Уровень L3	Вход 6
Цилиндр 6	Процент заливки	Уровень L3	Вход 6
	Отключен	-	-

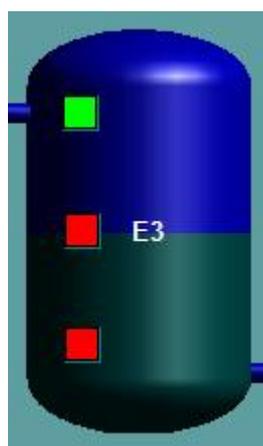


Рисунок 4.23 – Динамизация свойств ёмкости E3

Полная карта памяти показана на рисунке 4.23:

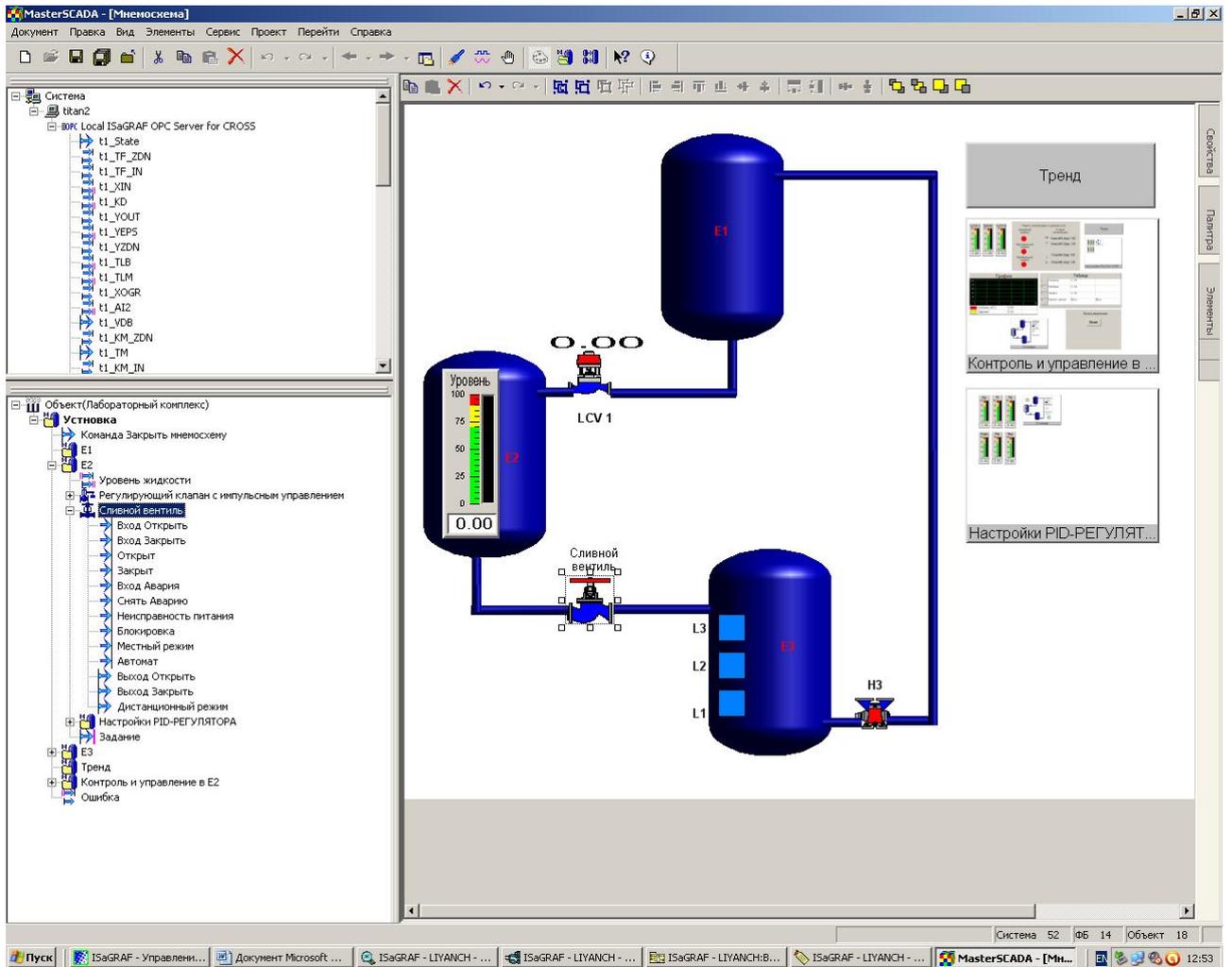


Рисунок 4.24 – Основная мнемосхема проекта

5 Создание Тренда

Создайте тенденцию, затем откройте вкладку Тенденции. После добавления перетащите переменные, которые необходимо отобразить, в этот тренд. (Рисунок 5.1).

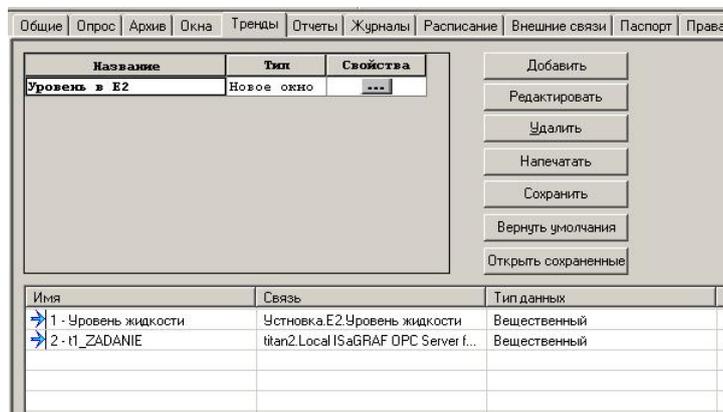


Рисунок 5.1 – Создание тренда

Нажмите на кнопку "Редактировать". Откроется и отобразится график тренда (Рисунок 5.2).

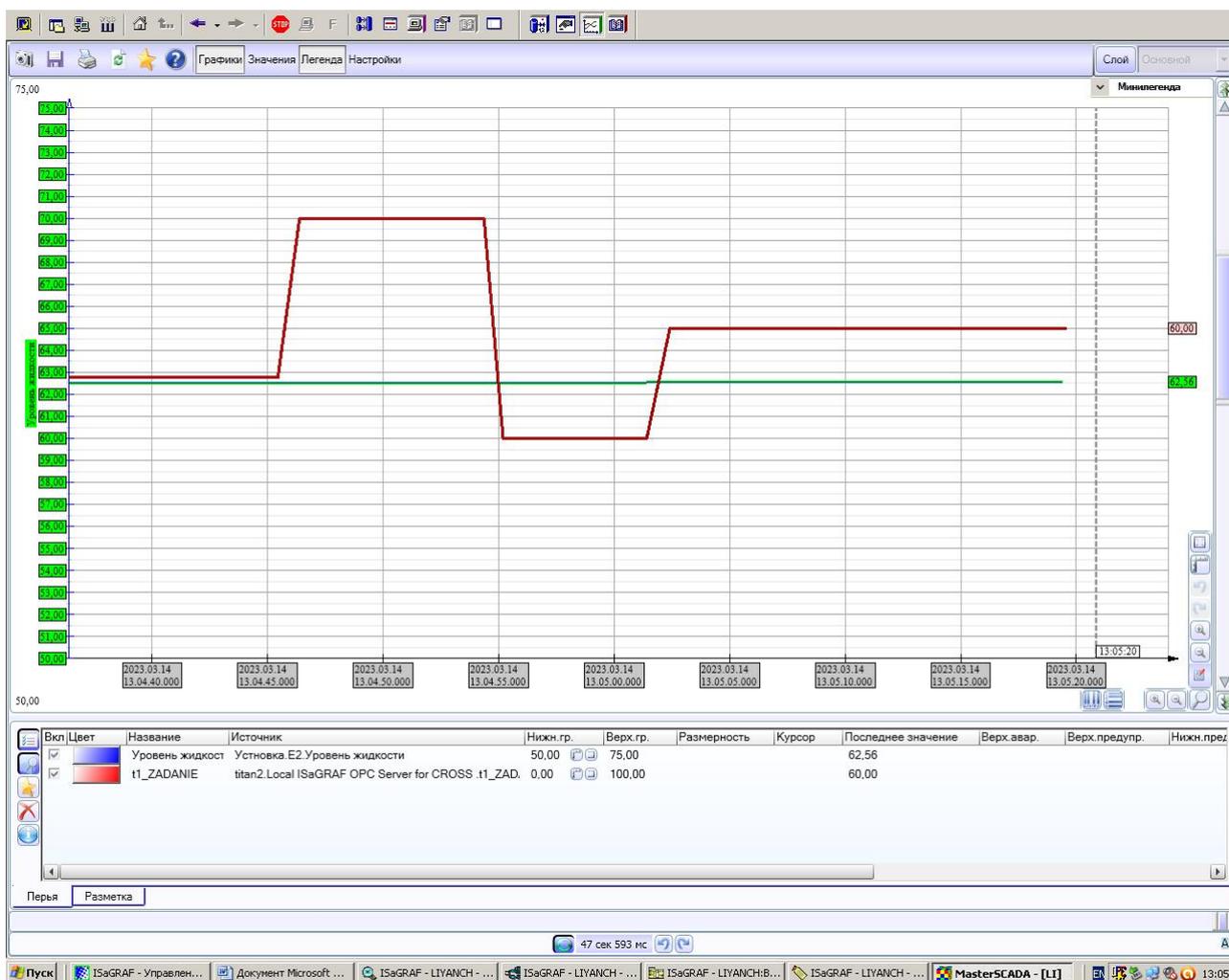


Рисунок 5.2 – Окно тренда

6 Создание окна «Контроль и управление в E2»

Это окно создано с целью более широкого использования возможностей MasterSCADA. С помощью этого окна можно будет дополнительно контролировать и управлять параметрами в ёмкости E2. В дереве объектов создайте Объект «Контроль и управление в E2». На странице свойств элементов откройте закладку Окна. Выберите МНЕМОСХЕМА – НОВОЕ ОКНО. Нажмите кнопку Редактировать. В конечном итоге окно будет иметь вид, представленный на рисунке 6.1. Вызвать данное окно можно будет, нажав соответствующую кнопку на основной мнемосхеме.

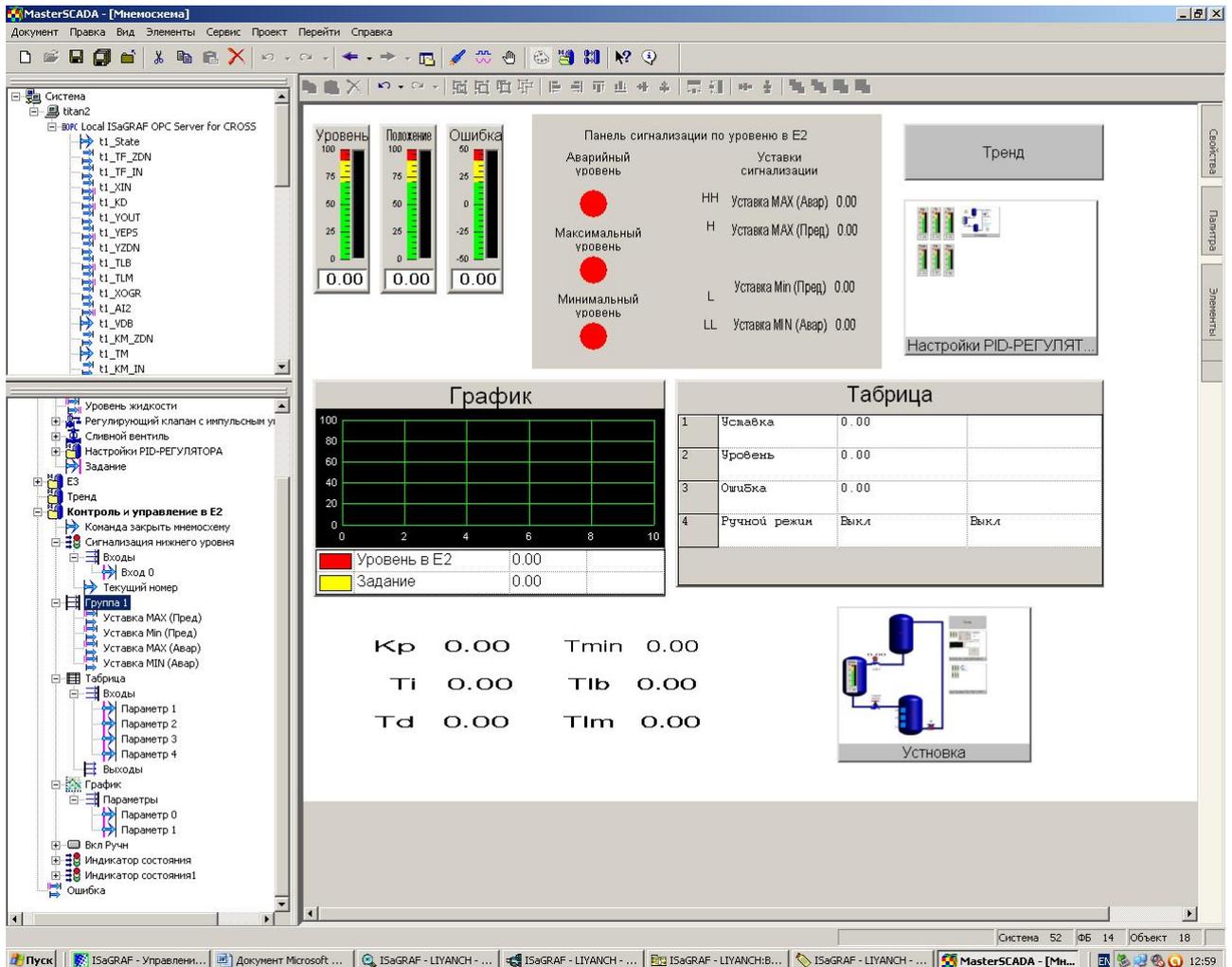


Рисунок 6.1 – Окно «Контроль и управление в E2»

Рассмотрим по отдельности как создавать элементы этого окна.

Создайте на мнемосхеме индикаторы значений «Уровень», «Ошибка», «Положение ИМ», перетащив созданные ранее значения ПК мыши и, выбрав вид «Индикатор», настройте необходимые свойства (рисунок 6.2).



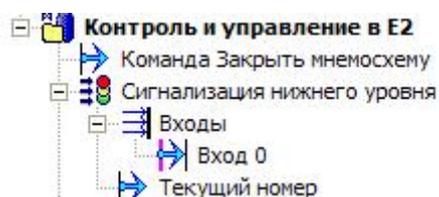
Рисунок 6.2 – Индикаторы значений Уровня, Ошибки, положения ИМ
Панель сигнализации по уровню

Создайте элемент визуализации сигнализации превышения или понижения уровня в Е2. Для этого создайте предварительно на мнемосхеме основу Панели сигнализации по уровню с помощью Графических примитивов (рисунок 6.3).



Рисунок 6.3 – Основа для Панели сигнализации

Далее в Палитре элементов выберите ДАТЧИКИ/ИНДИКАТОР СОСТОЯНИЯ и вставьте его в Объект «Контроль и управление в Е2» в дереве объектов.



Выделите этот элемент. На Странице свойств элементов в закладке Общие присвойте имя «Сигнализация нижнего уровня». На закладке Цвета определите цвет индикатора при достижении предупредительной и аварийной границ (рисунок 6.4).

Вход	Цвет
По умолчанию	Красный
Вход 0	Синий

Рисунок 6.4 – Изменение цвета индикатора сигнализации

В дереве объектов установите связи входов данного элемента с ОРС-переменными:

Вход 0 – t1_DMIN – предупредительная сигнализация

Перетащите на основу элемент  Сигнализация верхнего уровня .

Аналогично создайте визуализацию сигнализации по верхнему уровню (рисунок 6.5).



Рисунок 6.5 – Создание Панели сигнализации

Создадим элемент визуализации сигнализации для аварийного уровня. При достижении уровня аварийных значений раздается звуковой сигнал. На мнемосхеме это отображается миганием соответствующего индикатора (Рисунок 6.6).

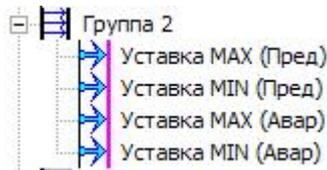


Рисунок 6.6 – Визуализация аварийного уровня

Для этого сначала было создано изображение динамика, а затем проведена динамизация свойства – мерцание.

Далее реализуем возможность задания уставок сигнализации.

В Объекте «Контроль и управление в E2» создайте Группу, добавьте в неё четыре команды, установите связи с ОРС-переменными.



Перетащите команды на основу, определите свойства (рисунок 6.7).

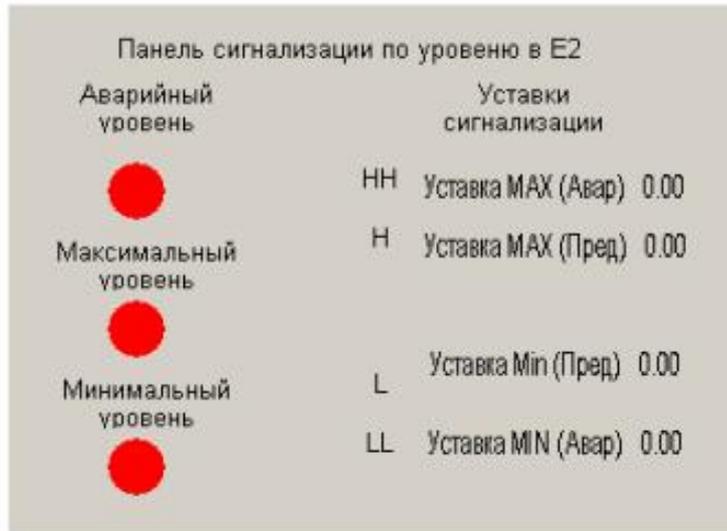


Рисунок 6.7 – Панель сигнализации по уровню в E2

7 Создание таблицы

Цель таблицы - более прямое представление переменных с помощью числового отображения.

Сначала перейдите в палитру, выберите CONTROL/TABLE и перетащите ее в объект "E2".

Таблица перетаскивается в программу памяти, и имя переменной может быть изменено автономной машиной.

1	Уставка	0
2	Уровень	0
3	Ошибка	0
4	Ручной режим	Выкл

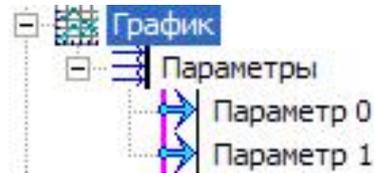
Установка.Контроль и управление в E2.Таблица

Рисунок 7.1 - Таблица

8 Создание графика

Функциональный блок графической визуализации помогает наглядно представить тенденцию изменения переменных.

В палитре инструментов выберите  объект (ЛК)  и перетащите его в объект "E2".



Перетащите переменную, соответствующую ОРС, в соответствующую позицию, а затем перетащите ее в экран программы памяти (рис. 8.1).



Рисунок 8.1 - График

9 Запуск программ визуализации

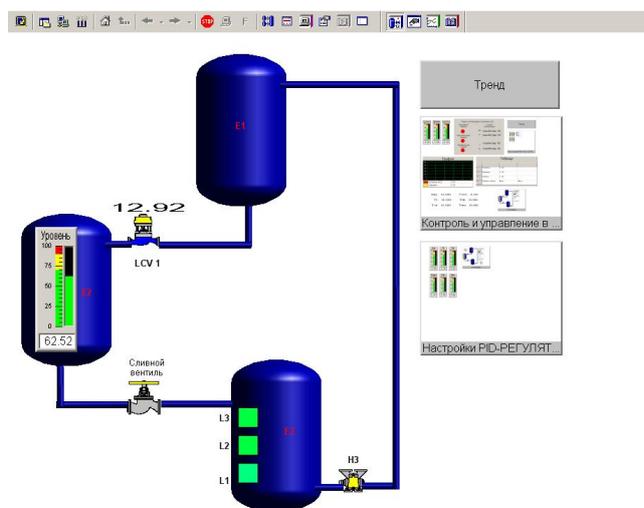


Рисунок 9.1 - Схема работы системы основной программы

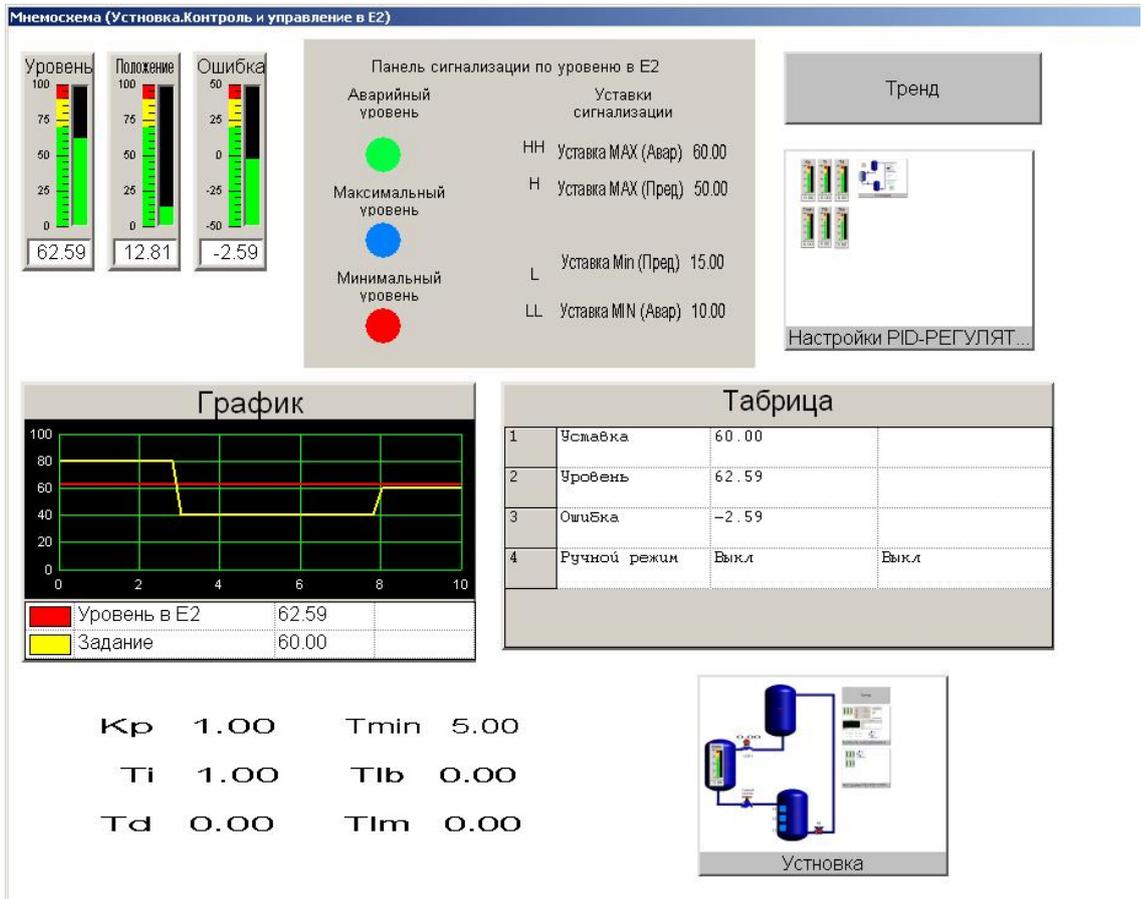


Рисунок 9.2 - Схема работы «Контроль и управление в E2»

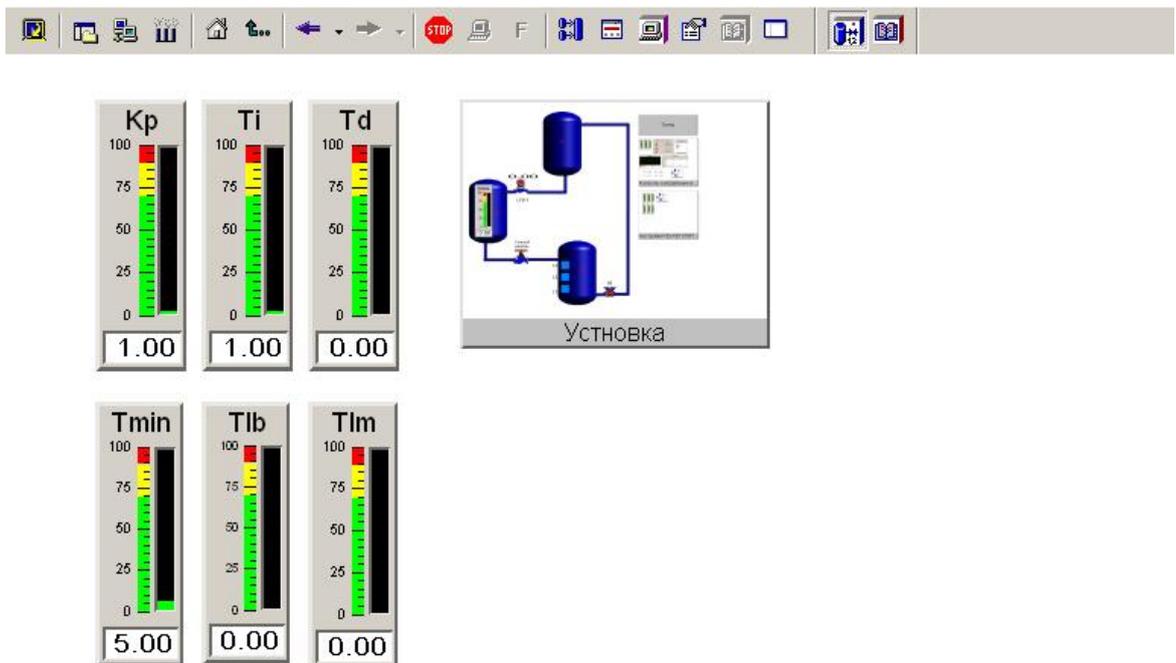


Рисунок 9.3 - Схема работы «Настройки PID-регулятора»

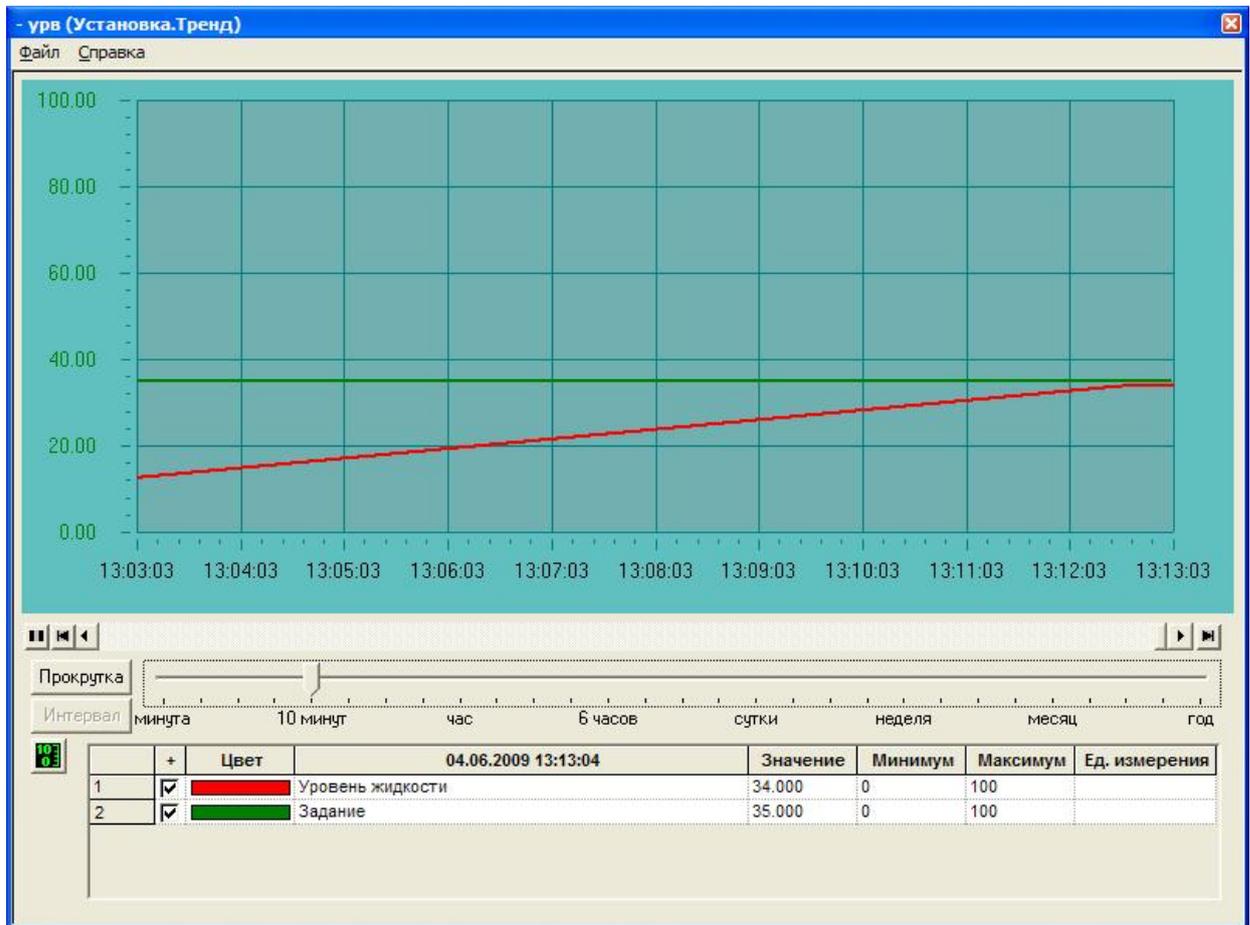


Рисунок 9.4 - Схема работы Тренда

10 Контрольные вопросы

11. Назначение SCADA-систем.
12. Перечислите наиболее распространённые на западном и российском рынках SCADA-системы.
13. Перечислите отличительные особенности пакета MasterScada.
14. Для чего используется OPC-сервер?

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы было разработано программно-методическое обеспечение лабораторного комплекса в системе технологического программирования ISAGRAF с применением языков программирования FBD и ST.

Создана программа визуализации объекта управления в SCADA пакете MasterSCADA 2.2.

Восстановлена работоспособность лабораторного стенда.

Основным результатом данного проекта являются методические указания по выполнению лабораторных работ, целью которых является получение практических навыков при создании:

- программ в системе программирования ISaGRAF;
- программ визуализации в SCADA пакете MasterSCADA.

Методические указания по выполнению лабораторных работ будут в дальнейшем использоваться студентами ТПУ, обучающимися по специальности 210100 - «Управление и информатика в технических системах» при изучении курса «Автоматизированные информационно-управляющие системы», а так же студентами других специальностей.