TOMSK POLYTECHNIC UNIVERSITY
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

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

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Программно-методическое обеспечение для изучения программного пакета ISaGRAF

УДК 004.415.2:004.451:004.43

Студент

Группа	ФИО	Подпись	Дата
158Т82	Цзян Ялун		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Кузьминская Е.В.	к.т.н., доцент		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШБИП	Былкова Т.В.			

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ООД ШБИП	Авдеева И.И.			

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Громаков Е. И.	к.т.н., доцент		

Томск – 2022 г.

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 управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни.
УК(У)-7	Способен поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности.
УК(У)-8	Способен создавать и поддерживать в повседневной жизни и в профессиональной деятельности безопасные условия жизнедеятельности для сохранения природной среды, обеспечения устойчивого развития общества, в том числе при угрозе и возникновении чрезвычайных ситуаций и военных конфликтов.
УК(У)-9	Способен проявлять предприимчивость в практической деятельности, в т.ч. в рамках разработки коммерчески перспективного продукта на основе научно-технической идеи.
УК(У)-10	Способен принимать обоснованные экономические решения в различных областях жизнедеятельности.
УК(У)-11	Способен формировать нетерпимое отношение к коррупционному поведению.
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	Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.
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 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 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	Способен участвовать: в разработке планов, программ, методик, связанных с автоматизацией технологических процессов и производств, управлением процессами, жизненным циклом продукции и ее качеством, инструкций по эксплуатации оборудования, средств и систем автоматизации, управления и сертификации и другой текстовой документации, входящей в конструкторскую и технологическую документацию, в работах по экспертизе технической документации, надзору и контролю за состоянием технологических процессов, систем, средств автоматизации и управления, оборудования, выявлению их резервов, определению причин недостатков и возникающих неисправностей при эксплуатации, принятию мер по их устранению и повышению эффективности использования.
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.

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

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
_____ __ _____ Громаков Е.И.
(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
158Т82	Цзян Ялун

Тема работы:

Программно-методическое обеспечение для изучения программного пакета ISaGRAF	
Утверждена приказом директора (дата, номер)	

Срок сдачи студентом выполненной работы:	

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе *(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.).*	Объект проектирования: сеть MESH с ESP32; Цель проектирования: изучить концепцию сети MESH, спроектировать, собрать и протестировать сеть MESH с ESP32. Ожидаемый эффект: созданная сеть MESH должна иметь функции удаленного беспроводного управления, автоматического подключения, автоматического ремонта, световой индикации подключения и т.д.
Перечень подлежащих исследованию, проектированию и разработке вопросов *(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).*	Теоретический обзор предметной области; Теоретическая проработка функционала устройства; Теоретический обзор способов дистанционного; Управление, дистанционный выключатель светодиодного индикатора со смартфона; Разработка блок-схемы, создание программного кода.

Перечень графического материала *(с точным указанием обязательных чертежей)*	

Консультанты по разделам выпускной квалификационной работы
(с указанием разделов)

Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Былкова Т.В.
Социальная ответственность	Авдеева И.И.

Названия разделов, которые должны быть написаны на русском и иностранном языках:

Все

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	

Задание выдал руководитель / консультант (при наличии):

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

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
158Т82	Цзян Ялун		

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

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 2021/2022 учебного года

Форма представления работы:

Бакалаврская работа

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
27.05.2022 г.	Основная часть ВКР	60
30.05.2022 г.	Раздел «Социальная ответственность»	20
30.05.2022 г.	Раздел «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»	20

СОСТАВИЛ:

Руководитель ВКР

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

СОГЛАСОВАНО:

Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Громаков Е.И.	к.т.н.		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО		
158Т82	Цзян Ялун		
Школа	ИШИТР	**Отделение (НОЦ)**	ОАР
Уровень образования	Бакалавриат	**Направление/ специальность**	15.03.04 Автоматизация технологических процессов и производств

Тема ВКР:

Программно-методическое обеспечение для изучения программного пакета ISaGRAF	
Исходные данные к разделу «Социальная ответственность»:	
Введение	Объект исследования: образец беспроводного измерительного устройства (БИУ) Область применения:лаборатория_ Размеры помещения:площадь 20м²,высота помещения 4 м, 5 метров в длину и 4 метра в ширину Количество и наименование оборудования рабочей зоны: 4 компьютера, 2 ESP32, 1 STM32, 4 линии передачи данных, 1 датчик, 1 OLED-экран Рабочие процессы, связанные с объектом исследования, осуществляющиеся в рабочей зоне: 1.Проектировать методы беспроводных и проводных соединений. 2. Разработать логическую схему в соответствии с поставленной целью 3, Написание программного кода 4.Экспериментировать и улучшатькод 5. Подготовка отчетов
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения	• Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 25.02.2022). • ГОСТ 12.0.003-2015 Опасные и вредные производственные факторы. Классификация. Перечень опасных и вредных факторов. • ГОСТ 12.1.004-91 ССБТ. Пожарная безопасность. Общие требования. • ГОСТ 12.1.038-82 ССБТ. Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов. • ГОСТ 22269-76 Система «человек-машина». Рабочее место оператора. Взаимное расположение элементов рабочего места. Общие эргономические требования.

	• ГОСТ 12.1.030-81 ССБТ. Электробезопасность. Защитное заземление. Зануление. • СанПиН 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания. • СП 52.13330.2016 Естественное и искусственное освещение. • СП 51.13330.2011 Защита от шума.
2. Производственная безопасность при разработке проектного решения	Вредные факторы: • Недостаточная освещенность; • Несоответствующие параметры микроклимата; • Повышенный уровень шума; • Повышенный уровень вибрации; • Нервно-психологические перегрузки, связанные с активным наблюдением за технологическим процессом Опасные факторы: • Короткое замыкание; • Статическое электричество; • Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий;. Требуемые средства коллективной и индивидуальной защиты от выявленных факторов: беруши, защитные ограждения, огнетушитель, устройства защитного отключения.
3. Экологическая безопасность при разработке проектного решения	***Воздействие на селитебную зону:*** отсутствует. ***Воздействие на литосферу:*** в виде отходов, возникших при поломке персонального компьютера, люминесцентных ламп и других электроприборов. Также стоит учесть отходы макулатуры. ***Воздействие на гидросферу:*** продукты жизнедеятельности персонала. ***Воздействие на атмосферу:*** отсутствует.
4. Безопасность в чрезвычайных ситуациях при разработке проектного решения	Возможные ЧС техногенного характера – пожар (возгорание) регламентировано ГОСТ Р 22.0.02-94 Наиболее типичная ЧС_пожар (возгорание) регламентировано ГОСТ Р 22.0.02-94
Дата выдачи задания для раздела по линейному графику	23.04.2022

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ООД ШБИП	Авдеева Ирина Ивановна			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
158Т82	Цзян Ялун		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
158Т82	Цзян Ялун

Школа	ИШИТР	**Отделение школы (НОЦ)**	ОАР
Уровень образования	Бакалавр	**Направление/специальность**	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
ечень 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
ечень графического материала *(с точным указанием обязательных чертежей)*:	
1. *Альтернативы проведения НИ* 2. *График проведения и бюджет НИ* 3. *Оценка ресурсной, финансовой эффективности НИ*	

а выдачи задания для раздела по линейному графику	

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент ОСГН ШБИП	Былкова Т.В.	к.э.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
158Т82	Цзян Ялун		

Реферат

Выпускная квалификационная работа содержит: 123 страницы, 114 рисунков, 13 таблиц, 16 источников, 1 приложение.

Ключевые слова: контроллер, методическое указание, программный продукт ISaGRAF, функциональный блок.

Объектом исследования является программно-методические обеспечение для изучения программного пакета ISaGRAF.

Цель работы: разработка программно-методического обеспечение для изучения программного пакета ISaGRAF на базе контроллеров "КРОСС". Разработанное обеспечение позволит улучшить практические навыки по программированию на языках в соответствии со стандартом МЭК 1131-3..

В процессе исследования программно-методического обеспечение для изучения программного пакета ISaGRAF на базе контроллеров "КРОСС".

Пояснительная записка ВКР была выполнена в текстовом редакторе WPS Word 2019.

В результате исследования был написание методического указания ля изучения программного пакета ISaGRAF.

Изучены различные программные языки в системе ISaGRAF, а именно языке FBD, LD, ST и IL; Рассмотрен и изучен контролер «КРОСС»; Получены практические навыки по программированию контроллеров «КРОСС» в системе ISaGRAF; Разработан проект методического указания по лабораторной работе, описывающее подробно работу с системой ISaGRAF и программирование на различных языках контроллера «КРОСС»; Разработанное методическое указание (приложение А) будет использоваться на лабораторных работах для студентов направления 15.03.04 «Автоматизация технологических процессов и производств».

Степень внедрения: письменно методические указание.

Область применения: студенты учатся использовать программный пакет ISaGRAF в учебном процессе.

В будущем планируется доработка программного кода для повышения производительности и расширения функционала.

Содержание

Введение 16
1 Программный пакет ISaGRAF 19
1.1 Язык последовательных функциональных схем 25
1.2 Язык функциональных блочных диаграмм.....28
1.3 Язык релейных диаграмм 31
1.4 Язык структурированного текста 33
1.5 Язык инструкций 34
2 Сравнительный анализ операций на языке FBD, LD,ST и IL 37
2.1 Логическое ИЛИ в системе ISaGRAF 37
2.2 П-регулятор в системе ISaGRAF 38
2.3 ПИД-регулятор в системе ISaGRAF 40
3 Лабораторный стенд для изучения языков программирования пакета ISaGRAF 44
3.1 Структурная схема лабораторного стенда 44
3.2 Состав и технические характеристики контроллера «КРОСС» 52
3.3 Программное обеспечение контроллера 52
4 Социальная ответственность 54
4.1 Правовые и организационные вопросы безопасности. 54
4.2Производственная безопасность 55
4.3 Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения 56
4.4 Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения 57
4.4.1 Экологическая безопасность 58
4.5 Безопасность в чрезвычайных ситуациях 58
5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение .. 62
5.1Планирование научно-исследовательских работ 62
5.2 Бюджет научно-технического исследования 66
5.3Определение ресурсной (ресурсосберегающей), финансовой, бюджетной,с оциальной и экономической эффективности исследования 69
ЗАКЛЮЧЕНИЕ 72
Список используемой литературы 73
Приложение А (обязательное) Методические указания к выполнению лабораторной работы «Программно-методическое обеспечение для изучения программного пакета ISaGRAF» 75

Введение

В современном мире трудно представить себе работу и быстрое развитие крупных производственных предприятий без средств автоматизации. Компании, занимающиеся добычей, переработкой и сбытом сырья благодаря использованию технологий автоматизации могут ускорить производственные процессы, улучшить качество своей продукции, а также значительно снизить риск возникновения несчастных случаев в процессе производства и оперативно их устранить. Помимо вышеперечисленных аспектов производственного процесса, большое значение имеет также высокая технико-экономическая эффективность, которая может быть достигнута при использовании автоматизированных систем управления технологическими процессами (АСУТП).

В связи с развитием современной экономики и высоким уровнем научно-технического прогресса необходимым условием внедрения автоматизированных систем управления технологическими процессами является широкое применение новейших компьютерных и машиностроительных технологий и необходимость подготовки высококвалифицированных кадров в области автоматизации управления и компьютерных технологий, способных проектировать, внедрять, применять и обслуживать средства автоматизации на всех уровнях технологического процесса.

Подготовка квалифицированных специалистов в области автоматизации затруднена без использования в учебном процессе лабораторных стендов, выполненных на базе современных промышленных контроллеров и программных пакетов ISaGRAF, а также программно-методического обеспечения для выполнения лабораторных работ.

Целью преддипломной практики является разработка программного обеспечения на различных языках с использованием программного комплекса

ISaGRAF и подготовка методического обеспечения для проведения лабораторных работ. Выполнение лабораторных работ на стенде с промышленными контроллерами Cross позволит студентам Томского политехнического университета приобрести практические навыки разработки управляющих программ на основе типовых форм визуализации процессов управления из пакета ISAGRAF.

1 Программный пакет ISaGRAF

Одной из самых успешных разработок программных продуктов, которые учитывают стандарт МЭК 1131-3 и учитывает в полном объеме требования международного стандара GUI (Graphical User Interface) является программа ISaGRAF.

Д анная программа позволяет разработанное программное обеспечение загружать в контроллер для дальнейшей работы с ним. Также в данном программном продукте предусмотрено ядро интерпретатора, которое позволяет переводить алгоритм пользователя в рабочее время выполнения. Применение данных функций позволяет концентрировать машинно-зависимый код, что в свою очередь, позволяет снизить денежные расходы при использовании другого контроллера.

Системный инструментарий ISaGRAF может писать собственные драйверы ввода-вывода, добавлять указанные рынком функциональные блоки, подключаться к более высоким системам или облегчать маркировку продуктов. Это окажется уникальным набором инструментов с повыш е н н о й интеллектуальной производительностью в соответствии с собственными интересами.

Э т о т эскиз включает платформу разработки приложений, файл базы данных проекта, коммуникационный уровень, дополнительный канал связи и виртуальную машину. Виртуальная машина запускается на целевой платформе (конфигурации) и выполняет приложение автоматизации, скомпилированное из платформы разработки приложений. В IEC 61131-3 виртуальная машина выполняет ресурсы при работе контроллера в соответствии с конфигурацией. Конфигуратор (работающий контроллер) может запускать несколько ресурсов (виртуальных машин). А н а л о г и ч н ы м образом, приложения автоматизации, скомпилированные из среды разработки приложений, могут быть распределены по нескольким контроллерам запуска.

Стандарт МЭК 1131-3В содержит пять языков программирования,

которые учтены в ISaGRAF, а именно [1,2]:

– язык последовательных функциональных схем (SFC);
– язык функциональных блочных диаграмм (FBD);
– язык релейных диаграмм (LD);
– язык структурированный текст (ST);
– язык инструкций (IL).

На рисунке 1.1 [2] приведены основные операции пакета ISaGRAF во временном цикле.

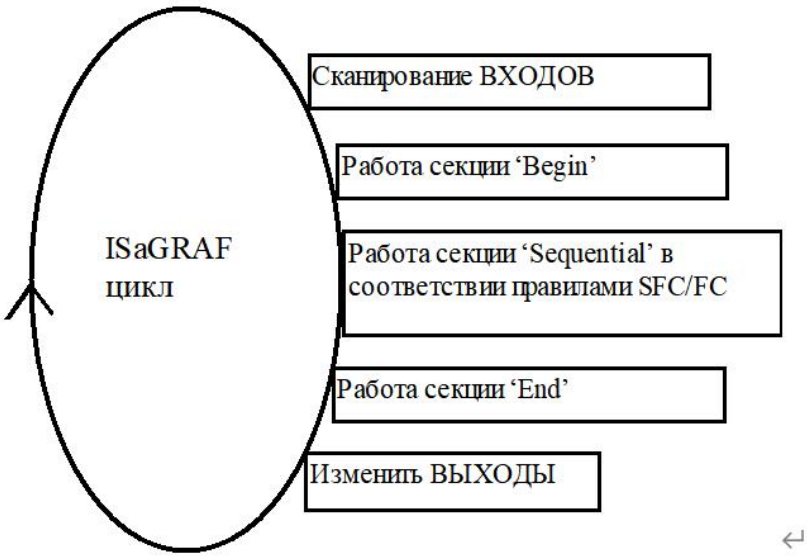


Рисунок 1.1 – Основные операции цикла

Таким образом, ISaGRAF считается системой синхронизации, которая работает по часам. Основной единицей измерений времени считается период, который показан на рисунке 1.2.

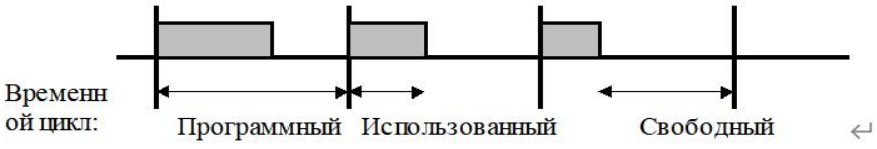


Рисунок 1.2 – Временной цикл

Применение временного цикла позволяют гарантировать, что:

– входные переменные сохраняют свое значение;

– устройства вывода изменяются один раза;

– глобальные переменные из различных программ согласуются между собой;

– временными реакция всего продукта можно управлять и оценивать его работу.

Необходимо отметить, что в системе ISaGRAF не допускается использовать в одной программе несколько языков, так как изначально при создании программы выбирается язык программирования и он не может быть изменен в дальнейшей работе, конечно же, есть исключения языки FBD и LD.

Используйте мощный браузер переменных ISaGRAF для объявления данных. Этот браузер поставляется с иерархической структурой переменных и сеткой для представления их определений. Для каждого ресурса разработчик может объявить переменные, используя простые тип ы данных или определяемые пользователем типы, такие как массивы или структуры. Новые переменные также могут быть объявлены во время разработки программы. Кроме того, уже объявленные переменные могут быть вставлены в любую программу для экономии времени редактирования. |ISaGRAF 5 поддерживает ряд новых типов данных: 8-битные беззнаковые целые числа, 16-битные целые числа и т.д.

Платформа разработки приложений ISaGRAF хранит все данные проекта (за исключением файлов ресурсов проекта МЭК) в базе данных MS-Access. Любое внешнее использование может записывать или читать эти данные через текстовые запросы, предоставляемые библиотеками динамических соединений Prohook или Propi. Ресурсы разработки приложений ISaGRAF могут совместно использовать переменные через механизм связывания. Связка представляет собой прямое соединение двух переменных. Связывание может осуществляться в рамках одного проекта или между разными проектами. Для управления обменом данными привязки используется сетевой драйвер. Механизм связывания может быть усилен функциональным блоком связи.

Функциональный блок связи позволяет программировать и детализировать обмен информацией между подключенными ресурсами, используя стандартные функциональные блоки.

Устройства ввода/вывода могут быть простыми или сложными, допускающими смесь типов и ориентаций, и могут принимать смесь булевых, целочисленных, реального времени или определяемых пользователем типов структур, способных управлять действительностью, стату с о м или любой необходимой информацией ИВ/О. После выбора устройства переменная может быть подключена к каналу простым щелчком мыши. В каждом канале усиление и смещение могут быть определены для переменных в инженерных единицах, что повышает независимость между аппаратным устройством и приложением. Например, изменение ввода/вывода в пределах определенного диапазона аппаратных значений может не потребовать фундаментального изменения приложения, а только переопределения коэффициента усиления и смещения канала.

При активации пользователем валидация стенда в lSaGRAF поддерживает любой язык ввода в синтаксисе исходного кода. Продукт также выполняет тест на соответствие приложению для проверки правильности подключения функциональных модулей. В случае ошибки при проверке программист берет сообщение об ошибке из прямого ресурса. В целом, lSaGRAF генерирует код, который может быть одновременно загружен в целевое оборудование и запущен, или смоделирован в прикладной платформе. Для того чтобы проверить проект, необходимо собрать код проекта. Этот раздел также очень полезен для проверки синтаксиса, и все обнаруженные ошибки можно легко найти одним щелчком мыши

Среда разработки приложений

Среда разработки приложений (ADE) облегчает задачи автоматизации на протяжении всего жизненного цикла системы - от проектирования системы,

запуска, эксплуатации до технического обслуживания. Среда разработки приложений является отправной точкой для использования инструментария ISaGRAF. Она предоставляет интуитивно понятную среду, помогающую разработчикам создавать собственные системы автоматизации и расширять ваши основные компетенции.

В ISaGRAF проект можно разделить на одно или несколько колец ПЛК или ресурсов, определив их аппаратные платформы и определив связи между ними. Конфигурации (аппаратные платформы, содержащие один или несколько ресурсов) и коммуникационные соединения (представляют физическое распределение проекта), а ресурсы представляют виртуальные объекты времени выполнения. Ресурсы содержат ряд программных единиц, называемых единицами управления программой или "POUS" "". Эти устройства управления программами могут программироваться на языке МЭК 61131-3 с использованием блок-схем, функций и функциональных блоков, а также поддерживают программирование функциональных блок-схем по МЭК 61499.

Для разработки программ среда разработки приложений ISaGRAF предоставляет мощный и интуитивно понятный графический и текстовый редактор, включая функции перетаскивания и вырезания и вставки для повышения удобства использования. Все переменные объявляются в словаре ISaGRAF Dictionary или импортируются из внешних баз данных. При программировании переменная может быть вставлена в программу одним щелчком мыши.

Кроме того, вы можете писать различные функциональные модули. Менеджер программы разработки приложений позволяет разработчику определить эти модули, их операции и интерфейсы, чтобы сформировать полный проект приложения. Такой подход позволяет легко повторно использовать эти единицы кода в дальнейшем, экономя время и деньги при разработке.

Используя среду разработки приложений, вы можете рассмотреть

различные аспекты вашей системы. Перспектива ресурсов знакомит вас с функциональными аспектами вашего приложения. Перспектива аппаратного обеспечения представляет и помогает вам управлять исполнением ресурсов в схеме аппаратного обеспечения.

Соединения ввода/вывода (адресация физических входов/выходов)

Для того чтобы установить связь аппаратно-независимых логических переменных в проектах ISaGRAF с физическими каналами ввода-вывода целевого промышленного компьютера или ПЛК, платформа разработки приложений ISaGRAF предоставляет мощный редактор соединений ввода-вывода. Простым щелчком мыши программист может определить соединения без необходимости углубленного изучения программного или аппаратного обеспечения. Полное разделение логического программирования и конфигурирования линий ввода-вывода обеспечивает возможность импорта программ разработки приложений на различные аппаратные платформы.

Моделирование и отладка

В приложениях ISaGRAF отладка не требует от разработчика возвращаться к базовому логическому блоку управления процессом или помнить точные правила синтаксиса всех языков. На протяжении всего процесса разработки платформа разработки приложений проверяет правила синтаксиса, заданные программой и языком, исправляя или подсказывая разработчику правильное использование каждого языка. Подробная гипертекстовая онлайн-справка включает перекрестные ссылки на стандарт IEC 61131-3.

Аналогичным образом, использование моделирования включает в себя мощный инструмент отладки. Этот инструмент быстро диагностирует производительность проекта приложения и показывает сегменты кода, которые необходимо оптимизировать. Симулятор также включает инструменты просмотра для редактирования и запуска сценариев моделирования, что значительно сокращает время отладки и проверки. Без целевой аппаратной

платформы программист может проверить весь проект приложения в лаборатории. Эмулятор в платформе разработк и приложений позволяет пользователю тестировать функциональность каждого модуля по отдельности или в целом. В то же время эмулятор может отслеживать выполнение программы и просматривать состояние любой переменной.

Самоописательный текст

П л а т ф о р м а разработки приложений ISaGRAF обладает мощной возможностью записи текстовых сообщений. Генератор текста создает полный, непрерывный текст для печати, описывающий все элементы проекта и историю их изменений. Пользователь может определить фактическое содержание текста, включая компоненты проекта, включенные в соответствии с заданными требованиями. Все графические и текстовые процедуры, а также вся структура проекта, таблицы соединений входов/выходов, таблицы переменных, справочные стандарты и т.д. в к л ю ч а ю т с я в окончательный текстовый файл. Печатаемый текст может соответствовать желаемому формату и может быть сгенерирован в текстовый файл или файл word..

1.1 Язык последовательных функциональных схем

Язык SFC используется для описания работы последовательных процессов. Программы SFC можно вводить с помощью графического редактора lSaGRAF SFC. SFC является ядром стандарта lEC1131-3, другие языки обычно используются для описания операций в рамках шагов и логических условий для осуществления переходов. графический редактор lSa GRAF SFC позволяет пользователю вводить Он сочетает в себе редактирование графики и текста, поэтому можно редактировать как диаграмму таблицы SFC, так и соответствующие операции и условные темы SFC.

П р и построении SFC-диаграммы пользователю достаточно ввести основные компоненты программы, а все одиночные соединительные линии, связывающие элементы (по горизонтали или вертикали), будут нарисованы

редактором SFC автоматически. После того, как пользователь определился с необходимой позицией и выбрал компонент SFC на панели инструментов редактора, символ вставляется в текущую позицию. Также можно использовать следующие операции на клавиатуре:

При построении SFC-диаграммы пользователю достаточно ввести основные компоненты программы, а все одиночные соединительные линии, связывающие элементы (по горизонтали или вертикали), будут нарисованы редактором SFC автоматически. Пользователю необходимо определиться с желаемой позицией и выбрать компоненты SFC на панели инструментов редактора.

Создание ответвлений и слияний

Ветви и слияния всегда проводятся слева направо. Левосторонние ветви ветвей и слияний должны располагаться в области диаграммы. Их тип (одинарная или двойная линия) можно выбрать с помощью соответствующей кнопки на панели инструментов.

Создание ответвлений и слияний

Используйте эти кнопки на панели инструментов для размещения начальных и конечных точек дополнительных ветвей. Левый угол ветви и слияния должен существовать до того, как будет вставлена новая ветвь. Правые углы ответвлений и слияний имеют тот же тип, что и левые углы основного пути. Правый угол не может быть размещен, если не существует левого угла основного пути.

Вставить шаг макроса

Эта кнопка вставляет макрошаг в диаграмму основной таблицы. Содержимое тела макрошага должно существовать где-то в той же программе S FC.

Изменить нумерацию ступеней и переходов

Идентифицирует каждый шаг или переход в диаграмме таблицы программы SFC логическим номером. Команда

"Редактировать/перенумеровать" позволяет пользователю заставить редактор автоматически установить номер ссылки в цифровом порядке для любого шага или перехо д а в редактируемой в данный момент программе SFC. При изменении нумерации шага нумерация всех переходов на этот шаг также автоматически обновляется. (Эта команда также может быть применена к макрошагам и начальным шагам).

Достижение ступени или перехода напрямую

Команда "Редактировать / Перейти к" позволяет пользователю напрямую обратиться к существующему шагу или переходу, при этом экран будет автоматически прокручиваться для отображения выбранного шага или перехода.

Поиск и замена текста

Команда "Правка/Найти и заменить" может быть использована для поиска и замены текстовых строк во всей программе (все шаги и переходы). Команда Найти/Заменить открывает диалоговое окно для ввода искомого текста и открывает непосредственно область программир о в а н и я Уровня 2, если совпадение найдено в программе Иерархия.

Начальная часть программы предназначена для описания предварительных операций с устройствами ввода и определения значений отфильтрованных переменных верхнего уровня. В последовательных программах используются переменные данные. Для выполнения защищенных операций над переменными и последующей отправки их значений на выходные устройства в основном используется конечная программа.

Программа последовательного сегмента SFC может управлять программой SFC низкого уровня, которая называется дочерний программой SFC. Под дочерними программы SFC понимается параллельные программы запущенные, удаленные, замороженные и перезапущенные родительской программой. Данные программы SFC могут применять локальные переменные и предопределенные слова. Указанные программы (родительская и дочерняя)

могут быть описаны в SFC.

Если дочерняя программа запускается благодаря родительской программе, тогда устанавливается маркер SFC (активация) на каждом первом шаге дочерней программы. Данная команда получается благодаря оператору GSTART.

Если дочерняя программа удаляется благодаря родительской программе, тогда родительная программа производит очистку всех используемых маркеров на каждом шаге дочерней программы. Эта команда описывается оператором GKILL.

Если дочерняя программа замораживается, тогда родительская программа SFC очищает все значения в подпрограмме и сохраняет их состояние в памяти. Эта команда описывается оператором GFREEZE [2].

Если дочерняя программа перезапускается, тогда происходит восстановление всех ранее полученных значений SFC, которые были стерты при заморозке программы. Команда описывается оператором GRST [2].

Программа последовательного сегмента FC может управлять другими программами FC. Родительская программа FC блокируется (ожидает), пока не будет выполнена подпрограмма FC. Родительская FC и подпрограмма не могут выполняться одновременно.

1.2 Язык функциональных блочных диаграмм

Язык функциональных блочных диаграмм (FBD) - это графический язык. Она позволяет программисту писать сложные программы, используя функции, доступные в библиотеке ISaGRAF, и связывать их линиями в области блок-схемы..

Диаграммы FBD описываются функциональными блоками, которые имеют входные и выходные переменные (рисунок 1.3), связанные между собой через линии связи. Выходные данные одного блока могут быть входом для другого блока.

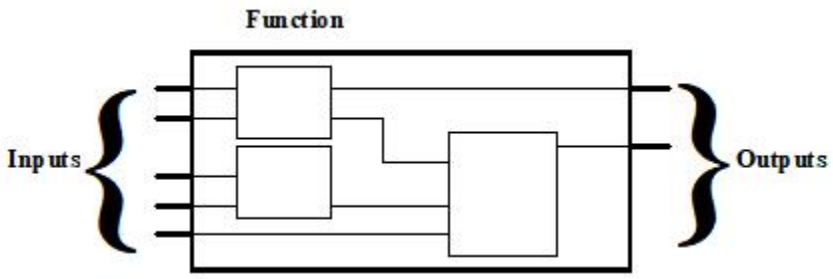


Рисунок 1.3 – Функциональный блок

Функциональные блоки представляют собой прямоугольники, которые имеют несколько входных и выходных точек. Входные и выходные переменные расположены с левого и правого краев соответственно. Функциональный блок предназначен для выполнения одной функции, имя которой обозначается символом в самом блоке, как это показано на рисунке 1.4.

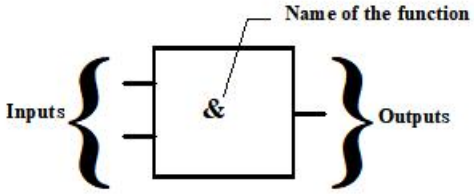


Рисунок 1.4 – Функциональный блок с именем

Каждая входная переменная или константа блока FBD имеет один тип и свой вход в блок [2].

Выходные переменные должны быть одного типа с выходом блока и связаны с ним. В FBD случаются случаи обратно вызывающей программы под которой понимают случай, если выходом является имя редактируемой программы.

Рассмотрим несколько команд используемых в FBD. Например, команда <RETURN> обозначает условное завершение программы и может служить выходом блока. Пример такой команды приведен на рисунке 1.5.

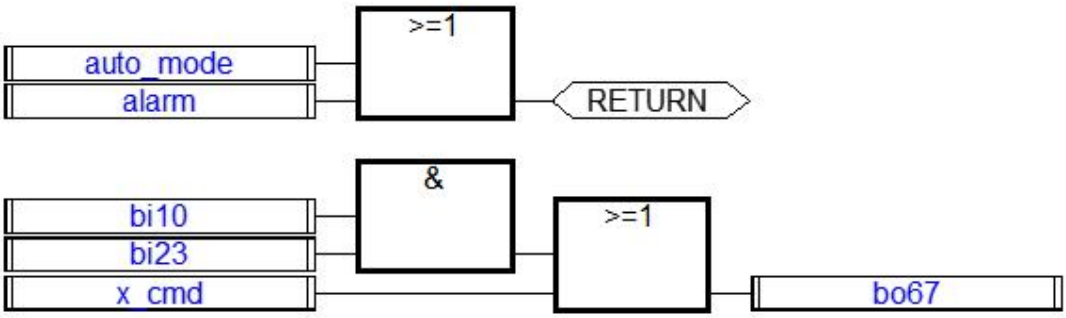


Рисунок 1.5 – Пример FBD программы с командой RETURN

Также в FBD используются управления диаграммой такие команды как прыжки и метки. К таким командам не могут присоединяться никакие другие объекты.

Если к входу блока подключен правый конец линии связи, то программа может быть завершена логическим отрицанием. Отрицание на блоке обозначается символом маленькое кольцо на входе в блок, пример которого показан на рисунке 1.7.

Рисунок 1.7 – Пример FBD программы использующей метки и прыжки

Функциональные блоки имеют один выход, который может быть входной переменной в другие блоки, пример которых показан на рисунке 1.8.

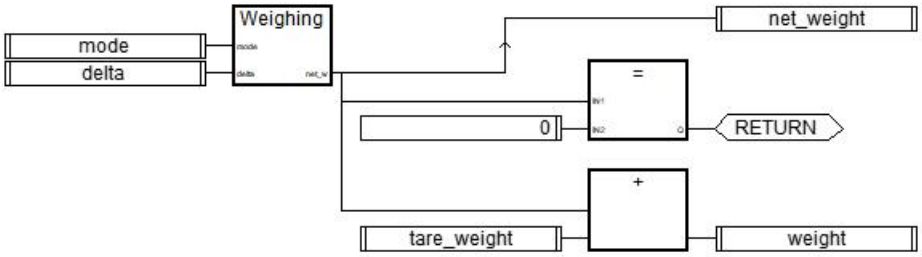


Рисунок 1.8 – Пример FBD с блоком SUB PROGRA

1.3 Язык релейных диаграмм

Графическое представление логических уравнений, объединяющих контакты (входы) и повороты (выходы) реализовано на языке релейных диаграмм [2].

Этот язык использует графику символы, что позволяет описывать булевые данные.

Пример LD диаграммы, которая ограничена вертикальными линиями справа и слева приведена на рисунке 1.9. Такие вертикальные линии принято называть силовыми рельсами.

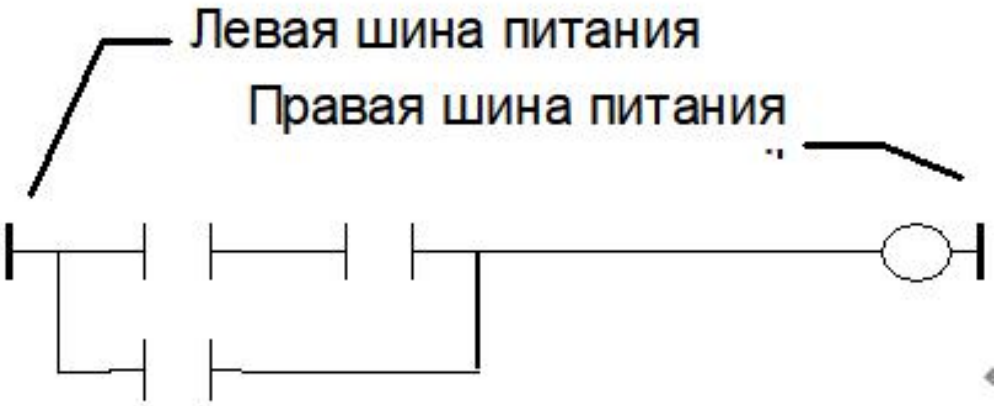


Рисунок 1.9 – LD диаграмма

Также в языке LD для соединения силовых шин между собой используются силовые линии, пример которых приведен на рисунке 1.10.

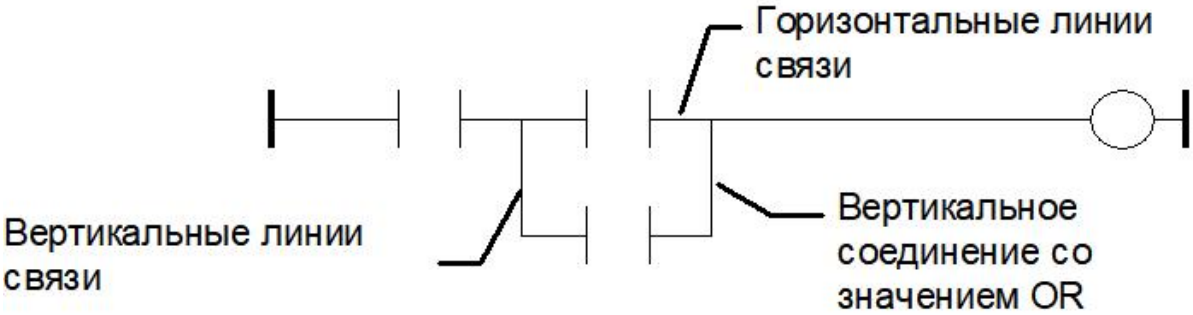


Рисунок 1.10 - Горизонтальные или вертикальные линии

Если отрезки линии непосредственно соединены, они имеют одинаковый

булевский или логический статус. Отрезок линии может быть TRUE или FALSE в обоих положениях. когда горизонтальная линия соединена с вертикальной дорожкой слева, статус TRUE.

Если использовать различные комбинации вертикальных и горизонтальных линий, то можно построить множество соединений, которые подчиняются логическим правилам.

На рисунке 1.11 приведен пример с левым множественным соединением, которая объединяет несколько горизонтальных линий. В таком примере булевское состояние правых концов соответствуют логическому ИЛИ(OR) всех левых концов.

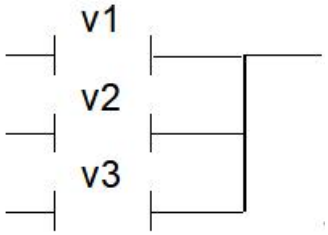


Рисунок 1.11 – Множественное левое соединение

На рисунке 1.12 приведен пример объединение одной горизонтальной линией.

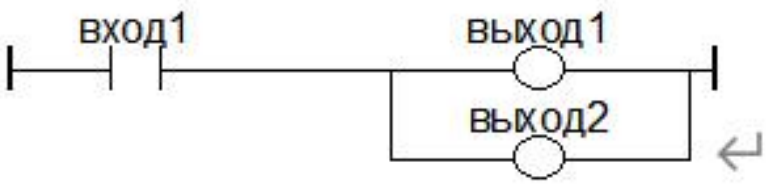


Рисунок 1.12 - Пример множественного правого соединения

На рисунке 1.13 приведен пример с множественным соединением с левой и правой сторон.

Рисунок 1.13 - Множественное двухстороннее соединение

1.4 Язык структурированного текста

Структурный язык высокого уровня, применяемый для автоматизации процессов реализован на языке структурированного текста (ST). Это язык программирования, разработанный для программируемых логических контроллеров (ПЛК) и являющийся одним из нескольких языков, поддерживаемых соответствующим стандартом IEC 61131-3.

Все языки МЭК 61131-3 поддерживают общие элементы МЭК 61131. Переменные и вызовы функций определяются Общими элементами МЭК 61131, поэтому в одной и той же программе можно использовать различные языки из МЭК 61131-3.

Язык программирования структурированных файлов похож на PASCAL и C, поэтому сложные вычисления, которые трудно выполнить с помощью лестничных диаграмм, могут быть созданы с использованием тех же методов программирования, что и в микрокомпьютерах и персональных компьютерах. Часто используемые программы и циклы могут быть легко использованы повторно путем создания FB (функциональных блоков).

Для того, чтобы лучше читать разработанную программы используют неактивные разделители с определенными установленными правилами:

– в строке нельзя писать более одного оператора;
– для передвижения сложных операторов применяется табуляция;
– использовать комментарии.

Также в языке ST применяются операторы и операнды. Для одного

выражения операнды должны быть одинаковыми по типу

Для следующих объектов могут быть использованы стандартные вызовы функций: подпрограммы; библиотечные функции и функциональные блоки; «С» функции и функциональные блоки; функции преобразования типов.

Выражение ST - это комбинация оператора ST и операнда переменной или константы. Для каждого отдельного выражения (комбинации операнда и оператора ST) тип операнда должен быть одинаковым. Это индивидуальное выражение имеет тот же тип, что и его операнды, и может быть использовано в более сложных выражениях.

Круглые скобки используются для разделения подразделов выражения и четкого определения приоритета операций. Для сложного выражения, когда круглые скобки не указаны, порядок операций задается приоритетом по умолчанию между операторами ST.

1.5 Язык инструкций

Язык IL (Instruction List) в МЭК 61131-3 - это язык низкого уровня, очень похожий на язык ассемблера, который является стандартным языком, основанным на языке списков инструкций производителей ПЛК во всем мире, и может использоваться для описания поведения функций, функциональных блоков и программ, а также поведения действий и переходов в последовательных функциональных блок-схемах.

Язык IL может использоваться для вызовов, таких как условные и безусловные вызовы функциональных блоков и функций, а также для выполнения присвоений и условных или безусловных пересылок внутри зон, и его не только легко изучить, но и очень просто реализовать и загрузить в ПЛК без компиляции. Однако в языке программирования IL отсутствуют эффективные инструменты для решения больших и сложных задач управления, поэтому его обычно не используют для решения больших и сложных задач управления.

Язык IL - это язык, состоящий из серии инструкций. Каждая инструкция начинается с новой строки, а полная инструкция состоит из оператора, за которым следуют операнды, которые являются переменными и константами, определенными в "общих элементах" МЭК 61131-3. Некоторые операторы могут принимать несколько операндов, которые разделяются запятыми. Перед инструкцией может стоять знак, за ним - двоеточие, а после операнда - комментарий.

Язык программирования таблиц IL характеризуется

- инструкции просты и легки в освоении и подходят для программирования небольших, более простых систем управления.

- Операторы используются для манипулирования переменными, вызова функций и функциональных блоков всех основных типов данных.

- язык, который может быть интерпретирован непосредственно в ПЛК и подходит для большинства производителей ПЛК.

- Языки программирования IL сложнее преобразовать в другие языки программирования, а программы, написанные на других языках программирования, легко преобразуются в языки программирования со списком инструкций. В таблице 1.1 представлены стандартные операторы, которые используются в языке IL.

Таблица 1.1 – Стандартные операторы языка IL

Оператор	**Модификатор**	**Операнд**	**Описание**
LD	N	переменная, константа	Загружает операнд
ST	N	переменная	Запоминает текущий результат
S		ВОО переменная	Устанавливает на TRUE
R		ВОО переменная	Сбрасывает на FALSE
AND	N (	ВОО	логическое И
&	N (	ВОО	логическое И
OR	N (	ВОО	логическое ИЛИ
XOR	N (	ВОО	исключающее ИЛИ
ADD	(	переменная, константа	Сложение
SUB	(	переменная, константа	Вычитание
MUL	(	переменная, константа	Умножение
DIV	(	переменная, константа	Деление
GT	(	переменная, константа	Проверить: >
GE	(	переменная, константа	Проверить: >=
EQ	(	переменная, константа	Проверить: =
LE	(	переменная, константа	Проверить: <=
LT	(	переменная, константа	Проверить: <
NE	(	переменная, константа	Проверить: <>
CAL	C N	Экземпляр функционального блока	Вызов функционального блока
JMP	C N	Имя	Прыжок на метку
RET	C N	метка	Возврат из подпрограммы
)			Выполнить задержанную операцию

2 Сравнительный анализ операций на языке FBD, LD,ST и IL

В данной главе приводятся различные операции, которые реализовываются на языках FBD, LD, ST и IL.

2.1 Логическое ИЛИ в системе ISaGRAF

На рисунках 2.1 - 2.4 приведены лицевые панели на различных языках системы ISaGRAF для логическое операции ИЛИ.

Состояние линии соединения на правом конце - это логическое ИЛИ(OR) состояния левого конца и значения переменной контакта.

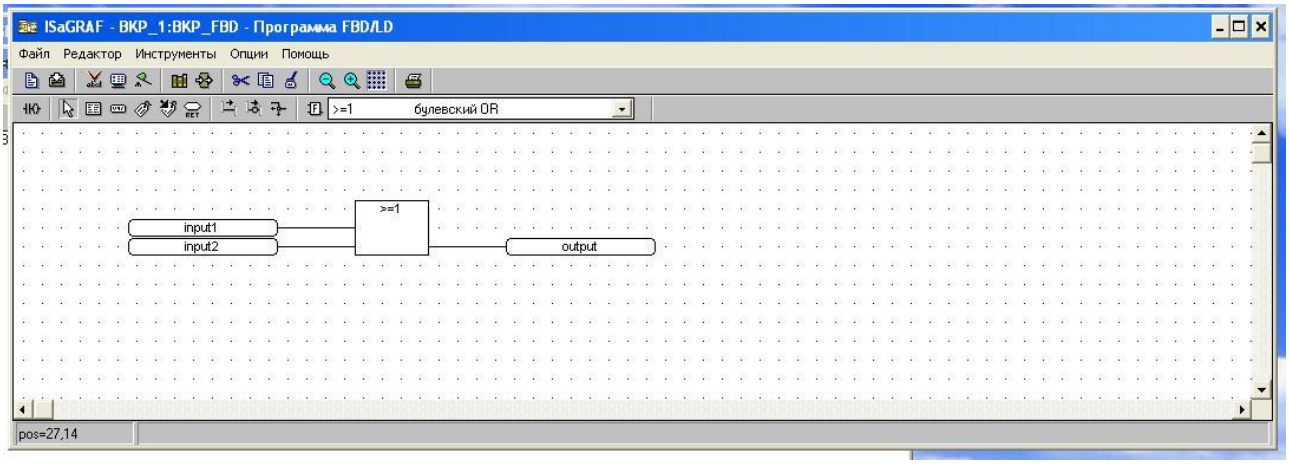


Рисунок 2.1 – Программа логической ИЛИ на языке FBD

Создание программы на языке FBD, реализующую операцию «логическое ИЛИ» в режиме симуляции. При этом сигналы не заводятся в контроллер и не выводятся из него, а используется симулятор ISaGRAF.

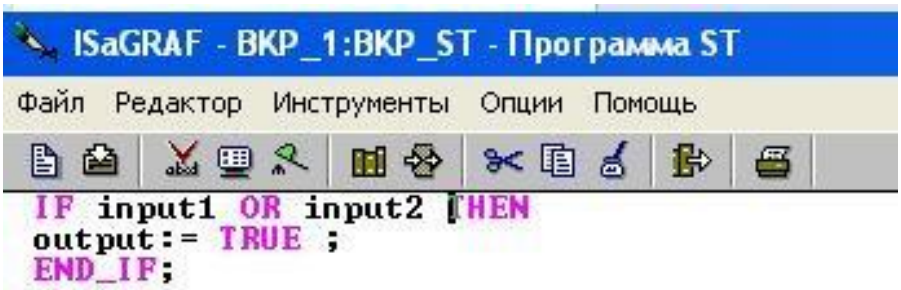


Рисунок 2.2 – Программа логической ИЛИ на языке ST

Создание программы в системе ISaGRAF на языке ST, реализующую аналоговый Логическое ИЛИ с сигнализацией по верхнему и нижнему предельному значению регулируемой величины.

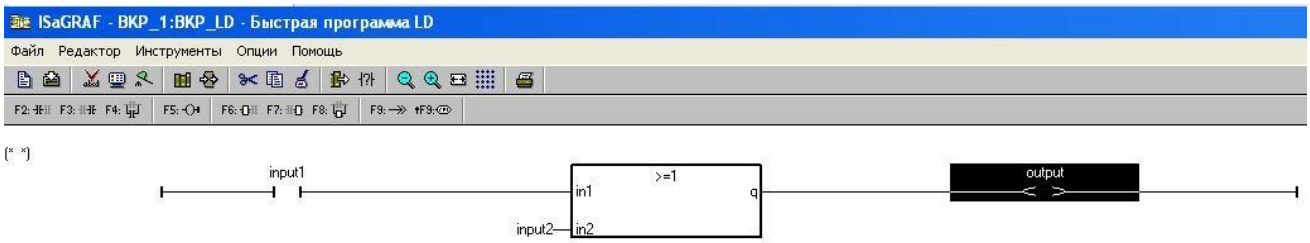


Рисунок 2.3 – Программа логической ИЛИ на языке LD

Создание программы в системе ISaGRAF на языке LD, реализующую аналоговый Логическое ИЛИ с сигнализацией по верхнему и нижнему предельному значению регулируемой величины.

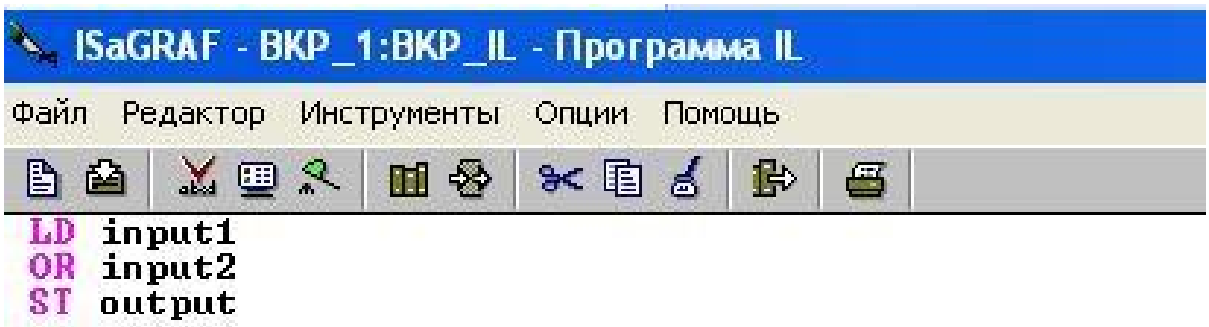


Рисунок 2.4 – Программа логической ИЛИ на языке IL

Создание программы в системе ISaGRAF на языке LD, реализующую аналоговый Логическое ИЛИ с сигнализацией по верхнему и нижнему предельному значению регулируемой величины.

Если линия связи слева от символа перехода находится в состоянии TRUE, выполнение программы перейдет к соответствующему выходу.

2.2 П-регулятор в системе ISaGRAF

Программы, реализованные на трех языках пакета ISaGRAF для

выполнения Р-регуляции и сигнализации, показаны на рисунках 2.5 - 2.7.

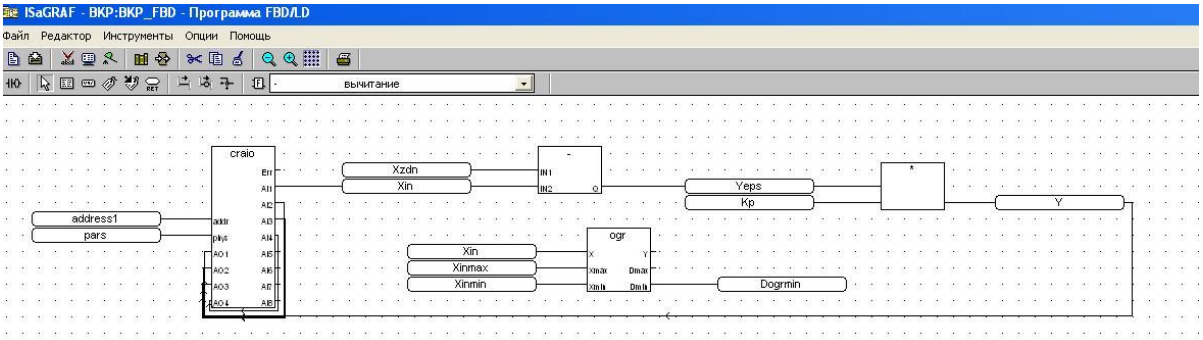


Рисунок 2.5 – Программа П-регулятор на языке FBD

Рассмотрим процесс составления программы, осуществляющую регулирование по П-закону и сигнализацию, реализованную с помощью пакета ISaGRAF.

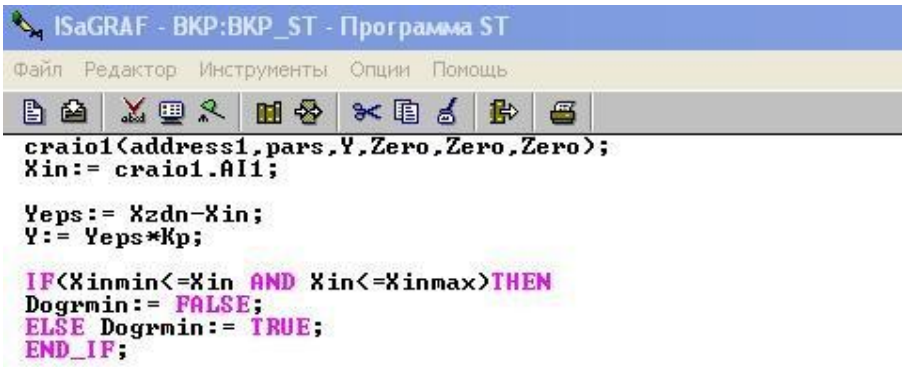


Рисунок 2.6 – Программа П-регулятор на языке ST

Рассмотрим процесс компиляции программы, которая реализует кондиционирование и сигнализацию по Р-закону, используя пакет ISaGRAF.

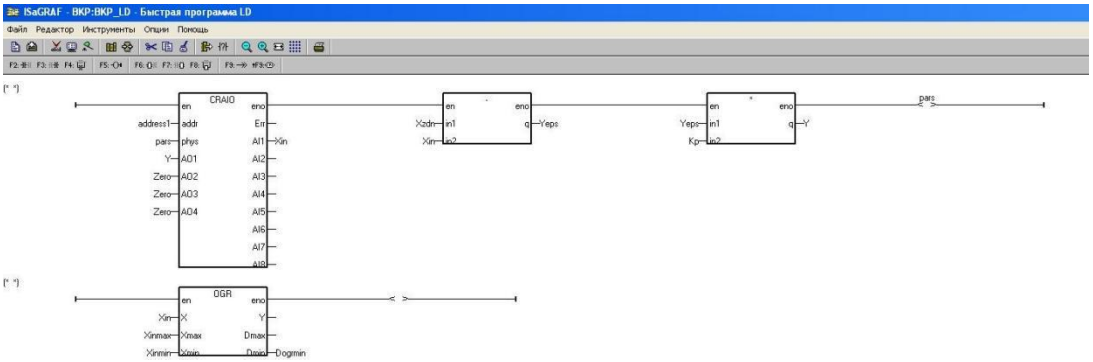


Рисунок 2.7 – Программа П-регулятор на языке LD

Рассмотрим процесс компиляции программы, которая реализует кондиционирование и сигнализацию по P-закону, используя пакет ISaGRAF.

Для сигнала "Xin", выводимого с фиксированного адреса модуля, находится разность с заданным сигналом Xzdn и определяется ошибка регулирования "Yeps". Блок умножения выполняет функцию P-регулятора, умножая сигнал ошибки "Yeps", поступающий в блок регулятора, на масштабный коэффициент "Kp". Выход контроллера - "Y".Для сигнала "Xin", который не находится в интервале (Xinmin, Xinmax), блок сигнализации OGR подает сигнал тревоги.

2.3 ПИД-регулятор в системе ISaGRAF

Программы, реализованные на четырёх языках пакета ISaGRAF для выполнения П-регуляции и сигнализации, показаны на рисунках 2.8- 2.11.

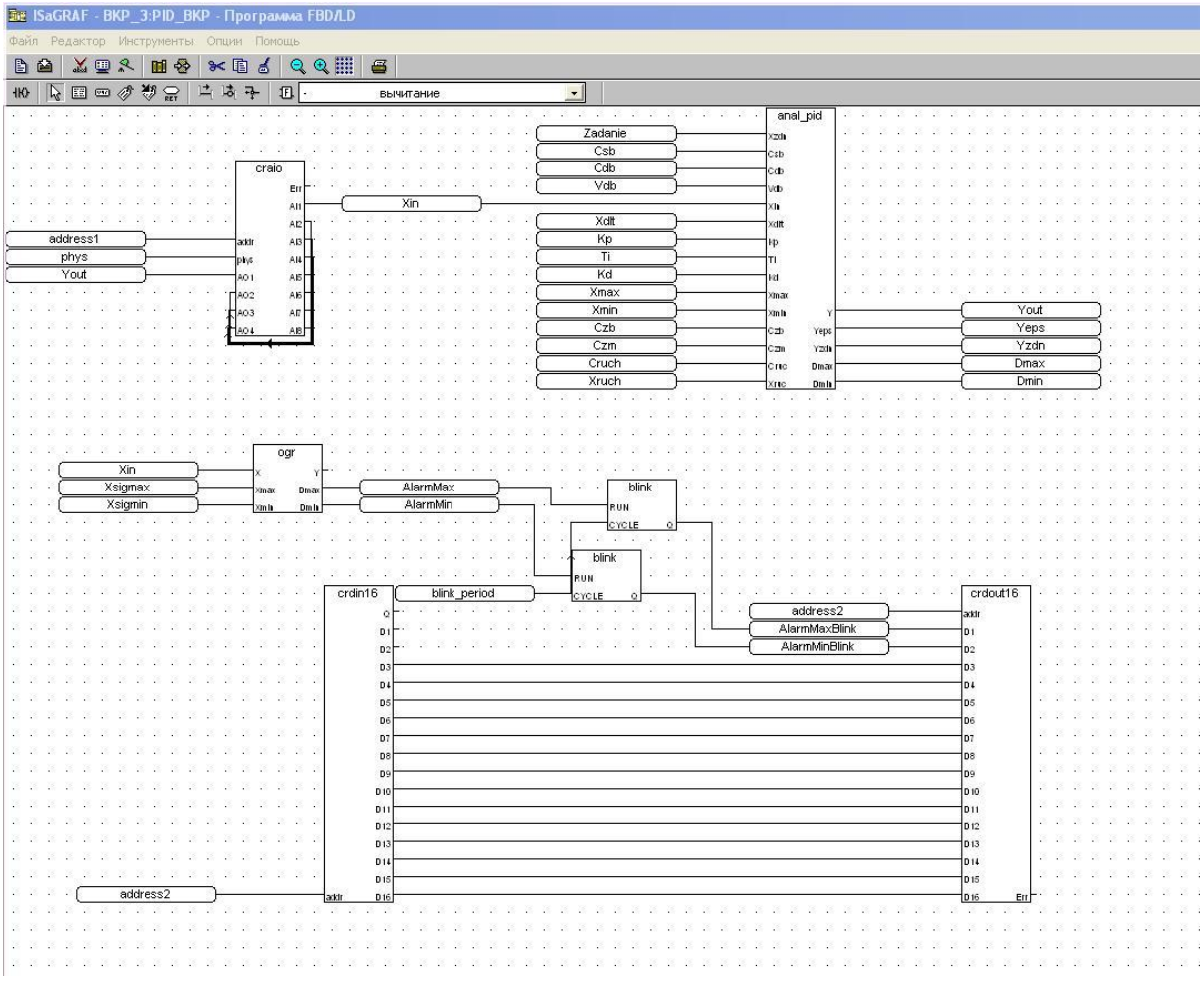


Рисунок 2.8 – Программа ПИД-регулятор на языке FBD

На рисунке 2.8 представлен фрагмент созданной программы ПИД-регулятора, который описывает сигнализацию по верхнему и нижнему предельному значению регулируемой величины.

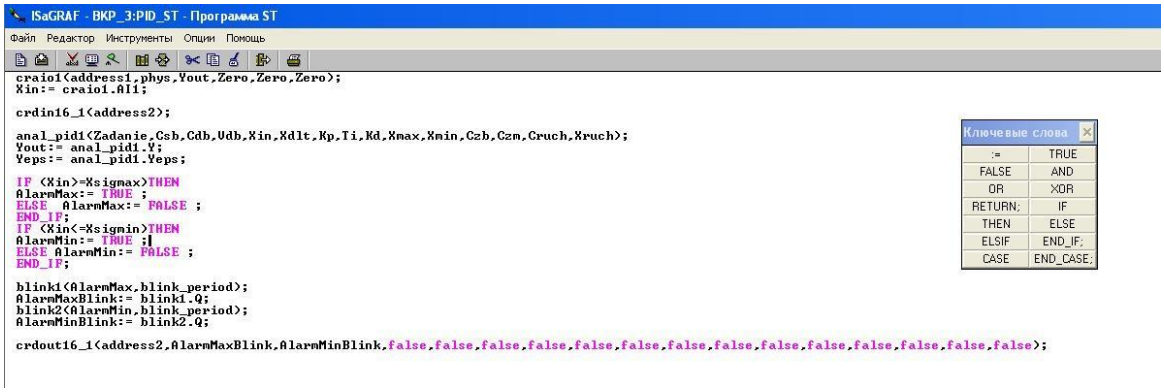


Рисунок 2.9 – Программа ПИД-регулятор на языке ST

На рисунке 2.9 приведен фрагмент ПИД-регулятора на языке ST, который описывает сигнализацию по верхнему и нижнему предельному значению регулируемой величины.

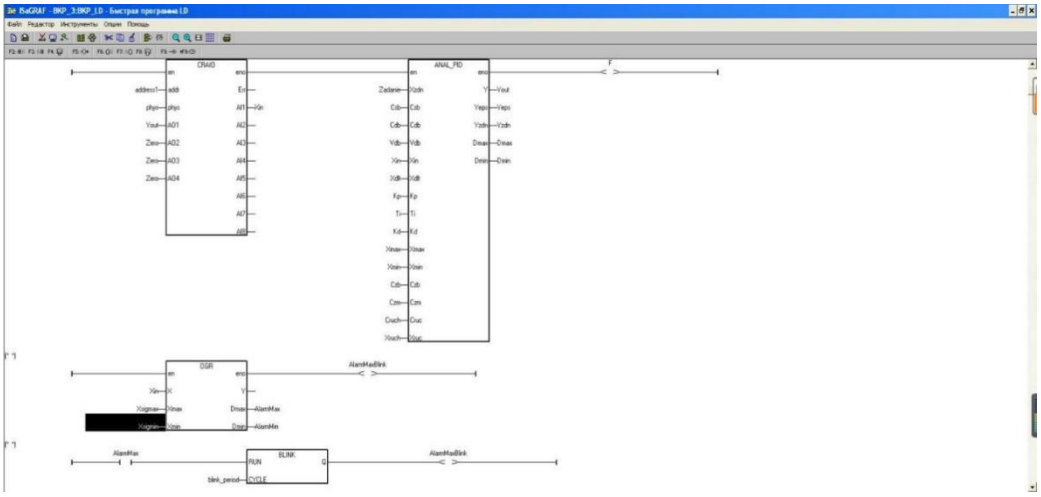


Рисунок 2.10 – Программа ПИД-регулятор на языке LD

На рисунке 2.10 приведен фрагмент ПИД-регулятора на языке LD, который описывает сигнализацию по верхнему и нижнему предельному значению регулируемой величины.

ISaGRAF - BKP_3:IL - Программа IL

Файл Редактор Инструменты Опции Помощь

```
LD address1
ST craio1.addr
LD phys
LD Yout
ST craio1.AO1
CAL craio1

LD Zadanie
ST anal_pid1.Xzdn
LD Csb
LD Cdb
LD Udb
LD craio1.AI1
ST anal_pid1.Xin
LD Xdlt
ST anal_pid1.Xdlt
LD Kp
ST anal_pid1.Kp
LD Ti
ST anal_pid1.Ti
LD Kd
ST anal_pid1.Kd
LD Xmax
ST anal_pid1.Xmax
LD Xmin
ST anal_pid1.Xmin
LD Czb
LD Czm
LD Cruch
LD Xruch
ST anal_pid1.Xruc
CAL anal_pid1

LD craio1.AI1
ST ogr1.X
LD Xmax
ST ogr1.Xmax
LD Xmin
ST ogr1.Xmin
CAL ogr1

LD ogr1.Dmax
ST blink1.RUN
LD blink_period
ST blink1.CYCLE
CAL blink1

LD ogr1.Dmin
ST blink2.RUN
LD blink_period
ST blink2.CYCLE
CAL blink2

LD address2
ST crdin16_1.addr
CAL crdin16_1

LD address2
ST crdout16_1.addr
LD blink1.Q
ST crdout16_1.D1
LD blink2.Q
ST crdout16_1.D2
CAL crdout16_1
```

Рисунок 2.11– Программа ПИД-регулятор на языке IL

На рисунке 2.11 приведен фрагмент ПИД-регулятора на языке IL, который описывает сигнализацию по верхнему и нижнему предельному значению регулируемой величины.

Уставка g, (называется zadanie), для объекта управления y (называется Xin) выходной сигнал подается обратно на контроллер, который таким образом производит управляющее воздействие U (называется Yout).

Функциональный блок ограничения **ogr** совместно с блоками **blink** позволяет реализовать сигнализацию по верхнему и нижнему допустимому значению регулируемой величины. Если значение сигнала на первом входе блока аналогового ввода-вывода (выхода объекта) превысит 2,5 мА, начнет мерцать первый световой индикатор, а если значение станет меньше 1мА, будет щелкать первое реле.Программно-методическое обеспечение

.

3 Лабораторный стенд для изучения языков программирования пакета ISaGRAF

3.1 Структурная схема лабораторного стенда

Лабораторный стенд для изучения языков программирования пакета ISaGRAF расположен в отделении автоматизации и робототехники ТПУ. На рисунке 3.1 приведена схема используемого стенда. Данный макет включает в себя персональный компьютер, автоматический выключатель (АВ), контроллер «КРОСС», аналоговый вычислительный комплекс АВК-6.

В контроллеры «КРОСС» входят центральный блок, блок питание, три модуля (УСО) и терминальные блоки. Центральный блок в свою очередь состоит из центрального процессора, модуля локальной сети ETHERNET, модуля питания и расширителя интерфейса.

Через субмодуль SM2-ETH, который соединен с локальной сетью ETHERNE происходит соединение центрального процессора контроллера SM2-CPU-1,5 с операторской станцией Т. На одном посадочном месте центрального блока ЦБ1 расположены субмодуль и плата центрального процессора, которые имеют общую лицевую панель.

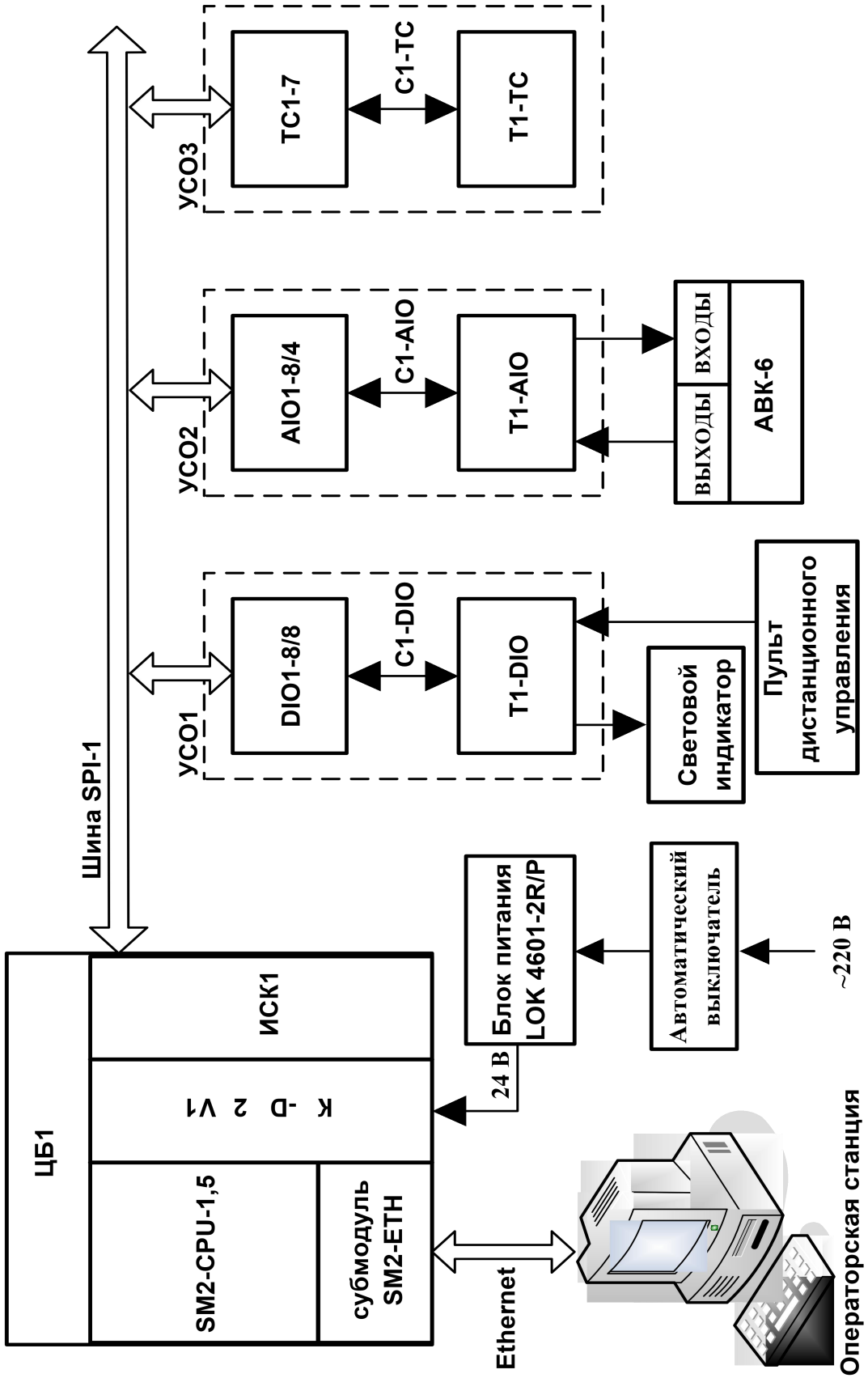


Рисунок 3.1 - Структурная схема лабораторного стенда

3.1.1 Краткое описание аналогового вычислительного комплекса

На рисунке 3.2 приведен состав аналогового вычислительного комплекса (АВК). АВК необходим для исследования систем автоматического управления (САУ) с помощью математических моделирований динамических процессов. АВК широко используется инженерами и студентами для изучения дисциплин с физико-математическим уклоном.

АВК-6 состоит из маловой аналогово-вычислительной машины (МЭМ-6) и системы индикационно-измерительных средств (СИИС)

МЭМ-6 позволяет программировать ее по структурно-математической модели изучаемого динамического процесса и при этом не нужны специальные знания и умения в области стандартного программирования ЭВМ.

Вообще АВК-6 состоит из отдельных модулей и блоков, но для простоты пользования они все заключены в один корпус. Использование одного блока позволяет быстро и удобно настроить и отремонтировать полностью комплекс. Корпус АВК-6 представляет собой каркас с четырьмя отсеками. Каждый блок модуля вставляется в свой отсек и винтами крепится к корпусу комплекса.

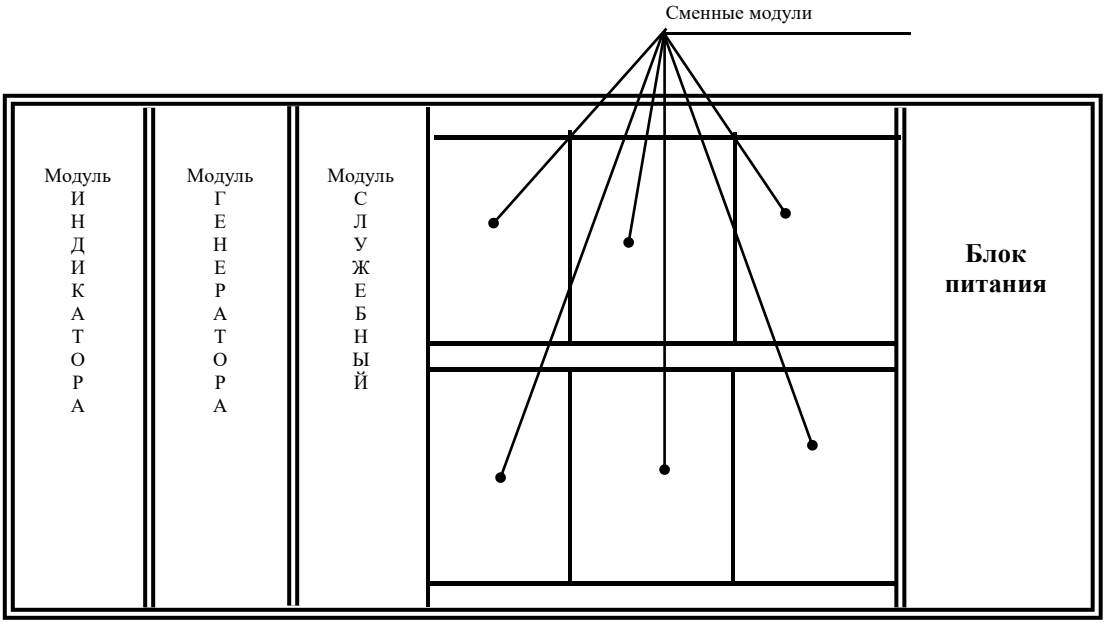


Рисунок 3.2 - Состав АВК-6

3.1.2 Описание контроллера «КРОСС»

В контроллер «КРОСС» (рисунок 3.3) включен центральный блок, модуль УСО (DIO1-8/8, AIO1-8/4 и TC1-7), терминальные блоки (T1-DIO, T1-AIO и T1-TC), блок питания LOK-4601-2R/P-ONE/ и гибкие соединители.

В свою очередь, центральный блок состоит из:

– модуля процессора SM2-CPU-1,5, входящего в центральный процессор и сетевой контроллер (субмодуль SM2-ETH);

– модуля питания KP-DC24V1;

– модуля ИСК1- расширитель интерфейса SPI;

– базового монтажного блока SMART2-BASE.

К центральному блоку подключен модуль УСО через ИСК1 шиной SPI-1, а с терминальными блоками гибкими соединителями C1-TC, C1-AIO, C1-DIO.

Состав контроллера приведен в таблице 3.1 [1]. Модули ввода/вывода (модули УСО) соединяются с центральным блоком ЦБ1 соединителем SPI через модули ИСК1, имеющие по два порта интерфейса SPI группами до восьми штук.

Контроллер имеет следующие характеристики, в диапазоне которых подтверждена их работоспособность:

– диапазон рабочих температур от 5°C до 50°C;

– влажность до 95 % без конденсата;

– не требует принудительной вентиляции в диапазоне рабочих температур;

– гарантия - 18 месяцев со дня ввода в эксплуатацию, сопровождение - 10 лет.

Параметры интерфейсных сигналов приведены в таблице 3.2 [1].

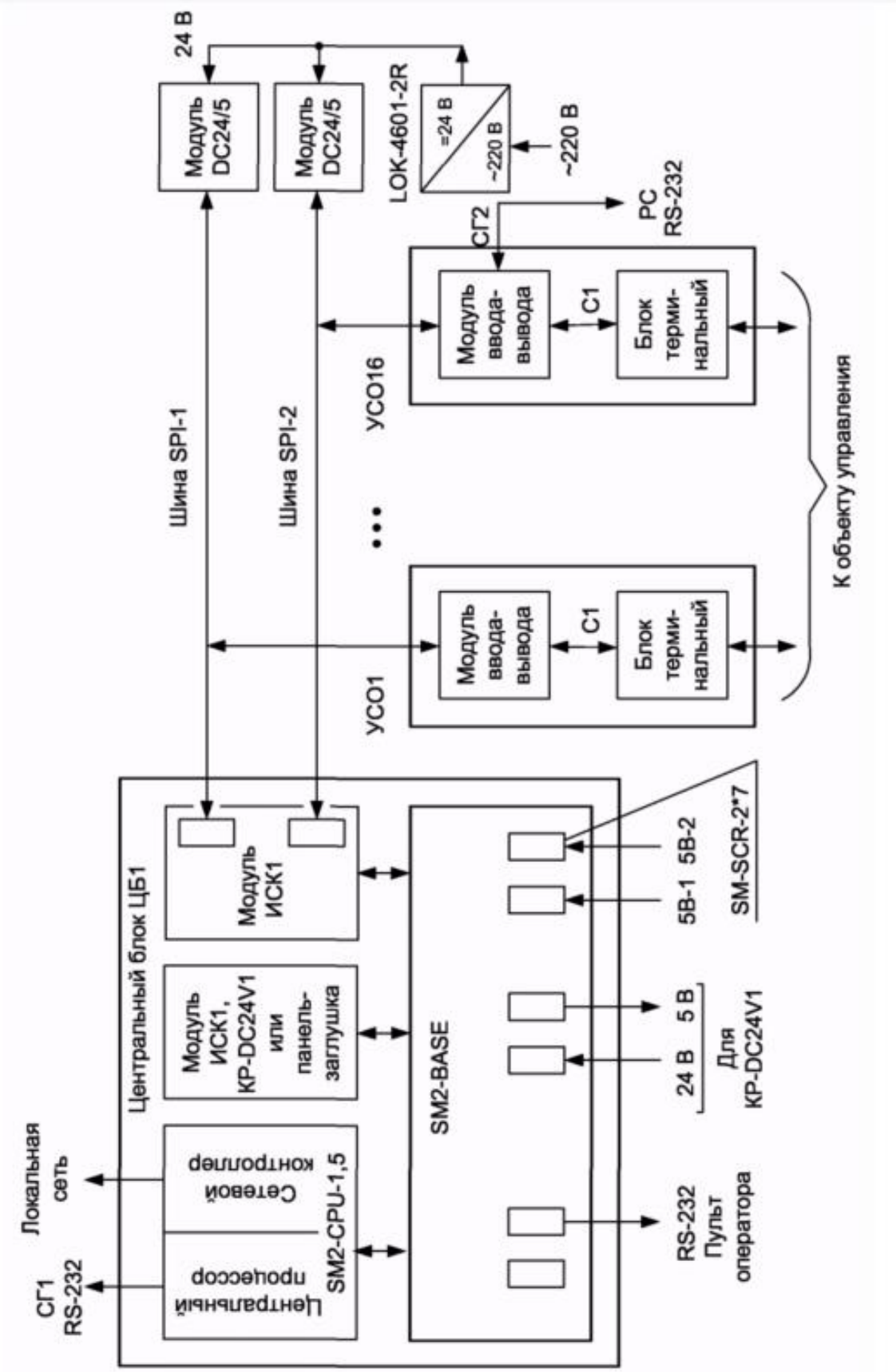


Рисунок 3.3- Архитектура контроллера

У контроллера есть измерительные каналы, которые могут преобразовывать аналоговые сигналы в цифровой код (АЦП) и из цифрового кода в постоянный ток (ПАП).

Метрологические характеристики контроллера приведены в таблице 3.1.

Таблица 3.1 – Метрологические характеристики контроллера

Модуль	**Сигналы**		**Пределы допускаемой основной приведенной (γо, %) или абсолютной (A_0, °С) погрешности**	**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
	ТПП(К), (0-20,620) мВ	(0-1730,0) °С	± 3,5 °С *	0,5 АО
	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(J), (0-64,948) мВ	(0-1120,0)°С	± 2,2 °С *	
	ТМК(Т), (0-20,255) мВ	(0-390,0) °С	± 0,8 °С *	
	ТХКн(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(N), (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(L), (0-64,771) мВ	(0-780,0) °С	±1,6°С *	
	ТСМ50', (50-92,800) Ом	(0-200,0) °С	± 0,4 °С	1,0 АО
TR1-8	ТСП50, (50-99,937) Ом	(0-266,0) °С	± 0,5 °С	1,0 АО
	ТСП50', (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	
	ТСН100, (100-199,467) Ом	(0-151,0)°С	± 0,3 °С	
AI1-8	Напряжение постоянного тока (0-10) В	13 бит	± 0,2%	0,5γо
	Постоянный ток (0-5), (0-20), (4-20)м А	13 бит	± 0,2%	
AIO1-8/4, AIO1-8/0	Напряжение постоянного тока (0-10) В	13 бит	± 0,2%	
	Постоянный ток (0-5), (0-20), (4-20) мА	13 бит	± 0,2%	
AIO1-8/4, AIO1-0/4	12 бит	Постоянный ток (0-5) мА	± 0,2%	
	12 бит	Постоянный ток (0-20), (4-	± 0,2%	

У контроллера имеется возможность выбора питания, которое может быть от сети переменного однофазного тока с номинальным напряжением 220 В с предельными отклонениями от минус 33 до плюс 22 В, частотой (50 ± 1) Гц и коэффициентом высших гармоник до 5 % или от внешнего нестабилизированного источника постоянного тока напряжением (24 ± 6) В.

3.1.3 Схема внешних соединений

Схема внешних соединений представлена на рисунке 3.4.

В данном лабораторном комплексе в качестве объекта управления была взята математическая модель САУ, набранная на аналоговом вычислительном комплексе АВК-6.

Выходы терминального блока T1-AIO соединяется с входами АВК- 6, при этом один провод общий.

Через автоматический выключатель на блоке питания, выдающим постоянное напряжение 24 В на контакты 3 и 13 клеммы с нечетной нумерацией слота «Slot B» подается питание с напряжением 220 В. Потом полученное напряжение 24 В преобразуется в 5 В, которое питает центральный процессор и шину SPI-1 через ИСК1. Питание ИСК1 осуществляется путем соединения контактов клеммы с четной нумерацией слота «Slot B» с контактами клеммы с нечетной нумерацией слота «Slot C».

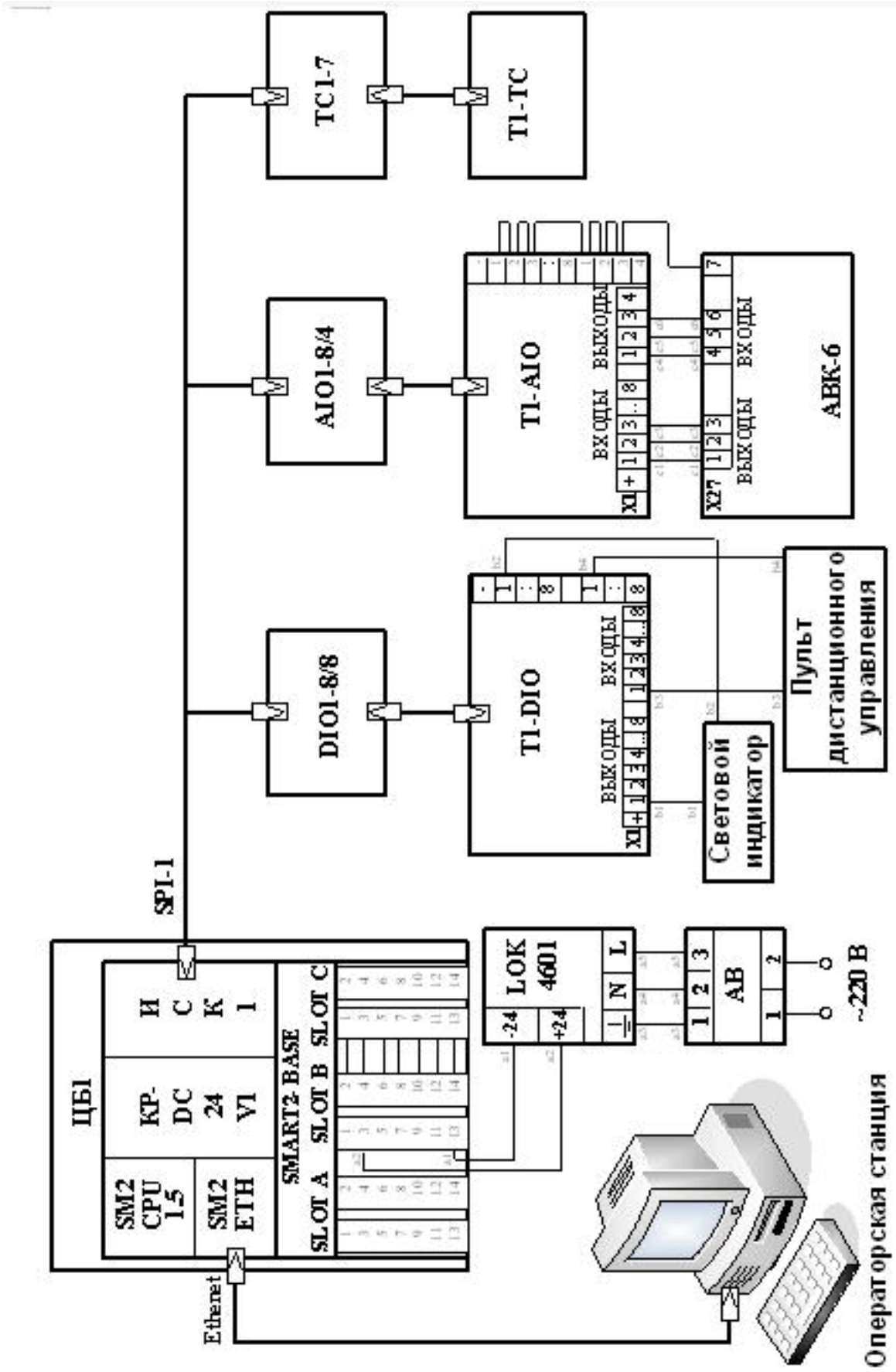


Рисунок 3.4 - Схема внешних соединений

3.2 Состав и технические характеристики контроллера «КРОСС»

3.2.1 Программное обеспечение контроллера

Программное обеспечение (ПО) контроллера состоит из резидентного ПО (встроенного в контроллер), программы эмулятора пульта настройки (ЭПН), конфигуратора и сервисной программы **«CrossTest»** для проверки модулей УСО.

В состав резидентного программного обеспечения (РПО) контроллера входят операционная система реального времени OS-9 и исполнительная среда системы технологического программирования ISaGRAF.

3.2.2 Основные сведения об операционной системе OS-9

Операционная система OS-9 - высокопроизводительная система, базирующаяся на микроядре с наращиваемой модульной архитектурой, поддерживает малые, средние и большие системы, обладает развитой иерархической системой ввода/вывода, включая коммуникации по последовательным портам, дисковые устройства, сетевые протоколы Ethernet, TCP IP, ISDN, сетевые файловые системы 8MB и NFS.

Конфигурация OS-9 в контроллере «КРОСС» содержит ядро - OS-9/68K Kernel версии V3.0.3 и ISP-пакет поддержки протоколов сети ИНТЕРНЕТ TCP IP, включая программный интерфейс sockets, серверы FTP, WEB, TELNET.

OS-9 обеспечивает:

1.гарантированное время реакции на программные и аппаратные прерывания и время переключения между задачами, что соответствует требованиям "жесткого" реального времени;

2.многозадачность;

3.механизмы синхронизации процессов;

4.надежную файловую структуру;

5.наличие службы времени;

6.развитую сетевую поддержку и другие механизмы реального времени в соответствии со стандартом POSEX.

Благодаря встроенной операционной системе контроллер по своим свойствам ставится в один ряд с универсальными компьютерами. Поэтому семейство «КРОСС» можно смело назвать "Два - в одном" (универсальный компьютер плюс промышленный контроллер).

Таким образом, в контроллерах произошло объединение мощи операционной системы реального времени OS-9 и традиций мира PLC, что дает дополнительные возможности:

– пользователь, умеющий программировать на языках программирования PLC, может программировать контроллеры привычными для него методами, не изучая OS-9;

– специалист, знакомый с OS-9, может наслаждаться программированием на языке C;

– специалист-универсал в полной мере может одновременно использовать эти обе возможности при решении своих задач, что дает фантастическую гибкость.

4 Социальная ответственность

Целью данной работы было изучение использования программного пакета ISaGRAF и понимание студентами принципов работы с программным пакетом ISaGRAF с помощью примеров и подробных операционных шагов.

Проектирование и выполнение выпускной квалификационной работы осуществлялось в Томском политехническом университете, корпус 10, комната 113. Пишется соответствующий программный код и проводятся эксперименты на контроллере КРОСС, чтобы убедиться, что программный код работает правильно. Для этой цели также было написано руководство по использованию программного пакета ISaGRAF.

В этой работе был разработан комплекс организационных, правовых, технических и институциональных мер, которые минимизируют негативное влияние длительного рабочего дня на компьютеры, также были рассмотрены вопросы техники безопасности, экологической и пожарной безопасности и даны рекомендации по созданию оптимальных условий труда.

Специфическая ситуация и стиль работы разработчиков характеризуется значительным умственным напряжением, большими нагрузками на визуальное оборудование, неподвижностью и напряжением в шейно-грудном и поясничном отделах позвоночника, что приводит к усталости, изменениям в функциональном состоянии центральной нервной системы и болям в запястьях, локтях, кистях, пальцах и спине. Боли в глазах и головные боли могут возникать при длительной работе перед экраном.

Проект не оказывает негативного социального или экологического воздействия, однако в ходе работы по написанию программ могут образовываться твердые отходы, такие как бумага, лампочки, отходы от продуктов питания и личной гигиены, а также отходы от канцелярских принадлежностей.

4.1 Правовые и организационные вопросы безопасности.

Согласно ТК РФ, N 197-ФЗ работник аудитории 101, 10 корпуса ТПУ имеет право на:

- рабочее место, соответствующее требованиям охраны труда;

- обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний в соответствии с федеральным законом;

- отказ от выполнения работ в случае возникновения опасности для его жизни и здоровья вследствие нарушения требований охраны труда, за исключением случаев, предусмотренных федеральными законами, до устранения такой опасности;

- обеспечение средствами индивидуальной и коллективной защиты в соответствии с требованиями охраны труда за счет средств работодателя;

Внеочередной медицинский осмотр в соответствии с медицинскими рекомендациями с сохранением за ним места работы (должности) и среднего заработка во время прохождения указанного медицинского осмотра.

4.2 Производственная безопасность

В данной выпускной квалификационной работе производится разработка программно-методического обеспечения для изучения программного пакета ISaGRAF на базе контроллеров "КРОСС". При этом используются такие программные пакеты, как ISaGRAF . Таким образом, в дипломном проекте используется следующее оборудование: ПК и промышленный контроллер КРОСС-500.

Работа заключается в написании программного обеспечения, его тестировании и разработке методического пособия, которые будут использованы для обучения студентов.

Для выбора факторов использовался ГОСТ 12.0.003-2015. Перечень опасных и вредных факторов, характерных для проектируемой производственной среды представлен в виде таблицы 4.1.

Таблица 4.1 - Опасные и вредные факторы при разработке

Факторы (по ГОСТ 12.0.003-2015)	**Нормативные документы**
• Неудовлетворительное освещение рабочей зоны;	СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95*
• Неудовлетворительный микроклимат	ГОСТ Р 56502-2020 Системы обеспечения микроклимата. Оценка энергетической эффективности систем отопления и вентиляции при проектировании зданий
• 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
• Опасность поражения электрическим током	ГОСТ 12.1.038-82 ССБТ. Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов.
• Короткое замыкание	ГОСТ 12.1.030-81 ССБТ. Электробезопасность. Защитное заземление. Зануление.
• Статическое электричество	ГОСТ 12.1.019-2017 ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты

4.3 Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения

При реализации разработанного проекта может возникнуть ряд вредных факторов, которые могут повлиять на здоровье, моральное, психическое состояние человека, экологию окружающей среды.

С целью предупреждения вредного необходимо, чтобы его уровень на рабочих местах с ПК не превышал допустимых значений (50дБА), установленных СП 51.13330.2011. Защита от шума. Ш у м создается вентилятором, установленным в ПК и принтером. При выполнении работ с использованием ПК в производственных помещениях уровень вибрации не

должен превышать допустимых значений вибрации для рабочих мест (категория 3, тип "в") в соответствии с действующими санитарно-эпидемиологическими нормативами. В соответствии с ГОСТ 12.1.012-2004 ССБТ. Вибрационная безопасность. Общие требования СН 2.2.4/2.1.8.566–96 «Производственная вибрация, вибрация в помещениях жилых и общественных зданий», значение виброускорения не более 83 дБ, а виброскорость не более 75 дБ.

Для уменьшения шума в помещении с ПК, как правило, применяют метод акустической обработки помещений, используя для облицовки ограждающих поверхностей звукопоглощающие материалы с максимальными коэффициентами звукопоглощения (α) в интервале частот (63 – 8000) Гц. С этой целью на потолках и стенах размещают перфорированные панели с звукопоглощающим наполнителем (минеральной ватой). Панели укрепляют непосредственно на поверхности ограждения или с отнесением от него на расстояние 20 см. В последнем случае применение звукопоглощающей облицовки более эффективно.

Дополнительным звукопоглощением могут служить однотонные занавеси из плотной ткани, гармонирующие с окраской стен и подвешенные в складку на расстоянии (15 – 20) см от оконного стекла. Ширина занавеси должна быть в 2 раза больше ширины окна. Снизить уровень шума можно также, используя для печати малошумные лазерные принтеры.

4.4 Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения

В данном подразделе рассматривается характер воздействия проектируемого решения на окружающую среду. Выявляются предполагаемые источники загрязнения окружающей среды, возникающие в результате реализации предлагаемых в ВКР решений.

4.4.1 Экологическая безопасность

В данном подразделе рассматривается характер воздействия проектируемого решения на окружающую среду. Выявляются предполагаемые источники загрязнения окружающей среды, возникающие в результате реализации предлагаемых в ВКР решений.

Анализ влияния объекта исследования на окружающую среду

П р и разработке программного обеспечения и методической поддержке программного комплекса ISaGRAF, вна селитебную зонуселитебную зону, атмосферу не отсутствуетоказывается воздействие .нНа литосферулитосферу в виде отходов. В основном это связано с выходом из строя персональных компьютеров, люминесцентных ламп и других электроприборов. Макулатура также заслуживает внимания. Воздействие на гидросферуВлияние на гидросферу также оказывают отходы жизнедеятельности человека. Весь процесс происходил в комнате 113 в здании 10, но в остальном никаких других воздействий не было. – как утилизация проходит?Техногенные отходы должны собираться для централизованной обработки.

Процесс исследования представляет из себя работу с информацией, такой как технологическая литература, статьи, ГОСТы и нормативно-техническая документация, а также разработка математической модели с помощью различных программных комплексов. Таким образом процесс исследования не имеет влияния негативных факторов на окружающую среду

4.5 Безопасность в чрезвычайных ситуациях

П р и проведении исследований наиболее вероятной ЧС является возникновение пожара в помещении 113, 10 корпуса ТПУ. Пожарная безопасность должна обеспечиваться системами предотвращения пожара и противопожарной защиты, в том числе организационно-техническими меропри ятиями.

Основные источники возникновения пожара:

1) неработоспособное электрооборудование, неисправности в проводке, розетках и выключателях. Для исключения возникновения пожара по этим причинам необходимо вовремя выявлять и устранять неполадки, а также проводить плановый осмотр электрооборудования,

2) электрические приборы с дефектами. Профилактика пожара включает в себя своевременный и качественный ремонт электроприборов,

3) перегрузка в электроэнергетической системе (ЭЭС) и короткое замыкание в электроустановке.

П о д пожарной профилактикой понимается обучение пожарной технике безопасности и комплекс мероприятий, направленных на предупреждение пожаров.

Пожарная безопасность обеспечивается комплексом мероприятий:

• обучение, в т.ч. распространение знаний о пожаробезопасном поведении (о необходимости установки домашних индикаторов задымленности и хранения зажигалок и спичек в местах, недоступных детям);

• пожарный надзор, предусматривающий разработку государственных норм пожарной безопасности и строительных норм, а также проверку их выполнения;

• обеспечение оборудованием и технические разработки (установка переносных огнетушителей и изготовление зажигалок безопасного пользования).

В соответствии с ТР «О требованиях пожарной безопасности» для административного жилого здания требуется устройство внутреннего противопожарного водопровода.

Согласно ФЗ-123, НПБ 104-03 «Проектирование систем оповещения людей о пожаре в зданиях и сооружениях» для оповещения о возникновении пожара в каждом помещении должны быть установлены дымовые оптико- электронные автономные пожарные извещатели, а оповещение о пожаре должно осуществляться подачей звуковых и световых сигналов во все помещения с

постоянным или временным пребыванием людей.

Аудитория 113, 10 корпуса ТПУ оснащена первичными средствами пожаротушения: огнетушителями ОУ-3 1шт., ОП-3, 1шт. (предназначены для тушения любых материалов, предметов и веществ, применяется для тушения ПК и оргтехники, класс пожаров А, Е.). Типы используемых огнетушителей при пожаре в электроустановках приведены в таблице 16.

Таблица 16 - Типы используемых огнетушителей при пожаре в электроустановках

Напряжение, кВ	**Тип огнетушителя (марка)**
До 1,0	порошковый (серии ОП)
До 10,0	углекислотный (серии ОУ)

Согласно НПБ 105-03 помещение, предназначенное для проектирования и использования результатов проекта, относится к типу П-2а (таблица 17).

Таблица 17 - Категории помещений по пожарной опасности

Категория помещения	**Характеристика веществ и материалов, находящихся (обращающихся) в помещении**
П-2а	Зоны, расположенные в помещениях, в которых обращаются твердые горючие вещества в количестве, при котором удельная пожарная нагрузка составляет не менее 1 мегаджоуля на квадратный метр.

В корпусе 10 ТПУ имеется пожарная автоматика, сигнализация. В случае возникновения загорания необходимо обесточить электрооборудование, отключить систему вентиляции, принять меры тушения (на начальной стадии) и обеспечить срочную эвакуацию студентов и сотрудников в соответствие с планом эвакуации.

ВЫВОД по разделу социальная ответственность:

В этом разделе рассматриваются организационные и правовые аспекты

обеспечения безопасности, а также анализируются нормативные документы, касающиеся завершающего этапа карьерного роста. На изученных рабочих местах значение всех производственных факторов соответствует нормам. Согласно Правилам электробезопасности, категория помещения соответствует категории 1 - "помещения без повышенной опасности". Согласно Правилам техники безопасности и охраны трудаПравилам техники безопасности и охраны труда, при эксплуатации электроустановок персонал должен иметь допуск по электробезопасности группы 1. Согласно СанПиН 1.2.3685-21 "Санитарные правила и требования к обеспечению населения безопасными и (или) безвредными факторами среды обитания", категория тяжелых работ в лаборатории относится к категории Iб (работа, выполняемая сидя, стоя или при ходьбе и связанная с физической нагрузкой). Пожар является наиболее типичной чрезвычайной ситуацией. Соблюдение правил и мер, рекомендованных в данном разделе, поможет избежать аварийных ситуаций.

5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

5.1 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 сведены в таблицу:

Таблица 5.1.1 - 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	И – 100%
Исследование и анализ прогпамов на языке FBD,ST,LD и IL	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 $t_{ож}$ применяется формула (1):

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

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 (**$T_{РД}$**) ведется по формуле (2):

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

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; Кд=1,2. Расчет продолжительности этапа в календарных днях ведется по формуле:

$$T_{КД} = T_{РД} \cdot T_{К}, \quad (3)$$

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:

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

где $T_{КАЛ}$ – календарные дни ($T_{КАЛ}$ = 366); $T_{ВД}$ – выходные дни ($T_{ВД}$ = 52);

$T_{ПД}$ – праздничные дни ($T_{ПД}$ = 14).

В нашем случае расчет ведем для шестидневной рабочей недели, Тк=1.22

Таблица 5.1.2 - Трудозатраты на выполнение проекта

Этап	Исполнители	Продолжительность работ, дни			Трудоемкость работ по исполнителям чел.-дн.			
		t_{min}	t_{max}	$t_{ож}$	НР	И	НР	И
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Постановка целей и задач, получение исходных данных	НР	1	2	1,4	1,68	0,00	2,05	0,00
Составление и утверждение ТЗ	НР, И	1	2	1,4	1,68	0,17	2,05	0,20
Разработка календарного плана	НР, И	2	3	2,4	2,88	0,29	3,51	0,35
Подбор и изучение материалов по тематике	НР, И	5	7	5,8	4,18	6,96	5,09	8,49
Обсуждение литературы	НР, И	1	2	1,4	1,01	1,68	1,23	2,05
Ознакомление с пакетом программирования контроллера Кросс – Isagraf	И	5	10	7	0,00	8,40	0,00	10,25
Исследование и анализ прогпамов на языкеFBD,ST,LD и IL	И	7	12	9	0,00	10,80	0,00	13,18
Разработка программного обеспечения	НР, И	10	15	12	4,32	14,40	5,27	17,57
Разработка методического обеспечения	НР, И	7	10	8,2	1,97	9,84	2,40	12,00
Технико-экономическое	НР, И	2	4	2,8	1,34	3,36	1,64	4,10
Оценка безопасности и экологичности работы	НР, И	2	4	2,8	1,01	3,36	1,23	4,10
Составление и оформление пояснительной записки	И	7	12	9	0,00	10,80	0,00	13,18
Защита дипломного проекта	И	1	2	1,4	0,00	1,68	0,00	2,05
Итого:				71,60	20,06	80,14	24,48	97,77

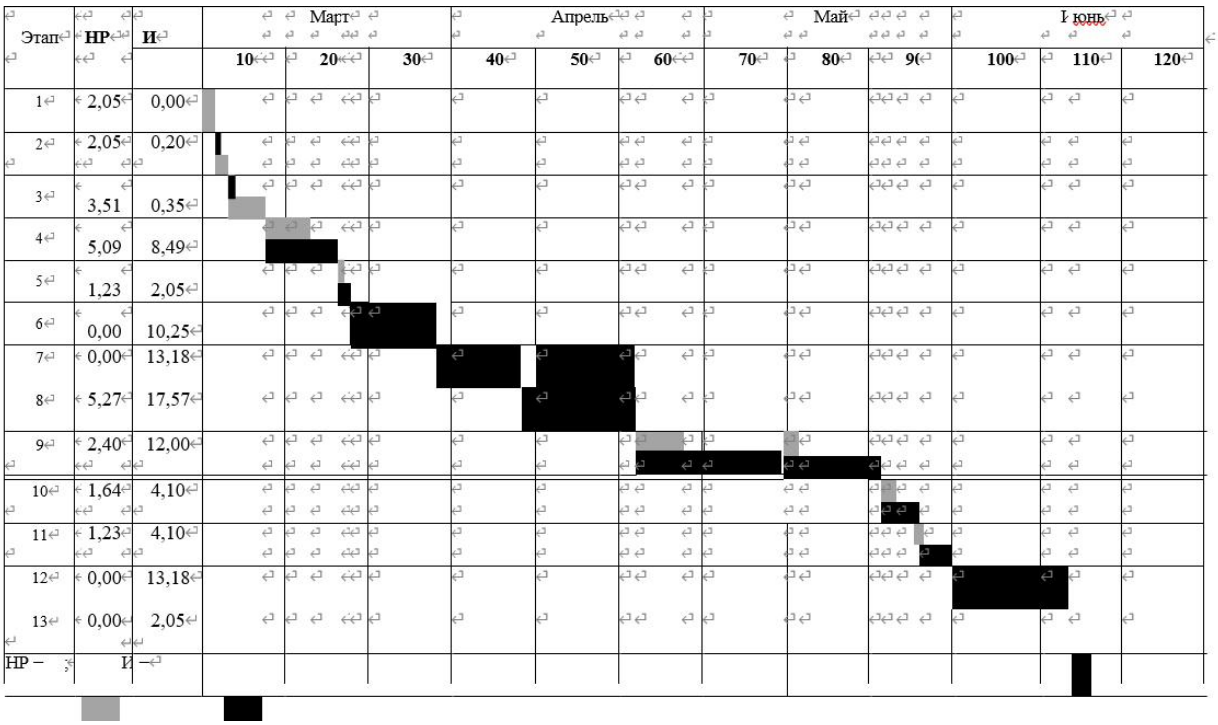


Рисунок 6.1.1 - Линейный график работ

5.2 Бюджет научно-технического исследования

В таблице 5.2.1 приведены расчеты затрат на материалы.

Таблица 5.2.1 - Расчет затрат на материалы

Наименование материалов	**Цена за ед., руб.**	**Кол-во**	**Сумма, руб.**
USB Flash Drive 4 Gb	500	1	500
Ручка	10	1	10
Карандаш	5	1	5
Тетрадь	12	1	12
Бумага для принтера формата А4	250	1	250
Картридж для принтера	1550	1	1550
Итого:			**2327**

Расчет затраты на электроэнергию для технологических целей приведен в таблице 5.2.2.

Таблица 5.2.2 - Затраты на электроэнергию технологическую

Наименование оборудования	Время работы оборудования $t_{ОБ}$, час	Потребляемая мощность $P_{ОБ}$, кВт	Затраты $Э_{ОБ}$, руб.
Персональный компьютер	641,1	0,3	**1 011,06**
Принтер	5,0	0,2	**5,26**
Контроллер КРОСС	512,9	0,1	**269,62**
Итого:			**1 285,93**

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

Используется формула:

$$С_{АМ} = \frac{Н_А * Ц_{ОБ} * t_{рф} * n}{F_Д},$$

где НА – годовая норма амортизации единицы оборудования;

ЦОБ – балансовая стоимость единицы оборудования с учетом ТЗР.

FД – действительный годовой фонд времени работы соответствующего оборудования.

Для ПК (300 рабочих дней при шестидневной рабочей неделе) можно принять FД = 300 * 8 = 2400 часа;

Для принтера FД = 500 часов;

Для контроллера КРОСС FД = 2400 часов;

tрф – фактическое время работы оборудования в ходе выполнения проекта, учитывается исполнителем проекта.

Для ПК tрф =641,1 часа. Для принтера tрф =5 часов.

Для контроллера КРОСС FД tрф = 512,9 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*45000*641,1*1)/2400 = 4808,25 руб; САМ (Пр) = (0,5*12000*5*1)/500 = 60 руб;

САМ (К) = (0,4*25000*512,9 *1)/2400 = 2173,08 руб.

Итого начислено амортизации 7 041,33 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:

$$ЗП_{дн\text{-}т} = МО/25.58,$$

учитывающей, что в году 300 рабочих дней и, следовательно, в месяце в среднем 25 рабочих дня (при шестидневной рабочей неделе).

Расчеты затрат на полную заработную плату приведены в таблице 5.2.1. Затраты времени по каждому исполнителю в рабочих днях с округлением до целого взяты из таблицы 5.1.1. 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 Ки

= 1,1* 1,188*1,3 = 1,699. Вышеуказанное значение Кдоп.ЗП применяется при шестидневной рабочей неделе.

Таблица 5.2.3 - Затраты на заработную плату

Исполнитель	**Оклад, руб./мес.**	**Среднедневная ставка, руб./раб.день**	**Затрат ы времен и,раб.дни**	**Коэффици ент**	**Фонд з/платы, руб.**
НР	**23 264,86**	930,59	20	1,699	**31 722,79**
И	**14 874,45**	594,98	80	1,699	**81 006,89**
Итого:					**112 729,67**

Отчисления во внебюджетные фонды = 112 729,67 * 0,3 = 33 818,9 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.

$C_{проч.} = (C_{мат} + C_{зп} + C_{соц} + C_{эл.об.} + C_{ам} + C_{нп}) \cdot 0,1$. Для нашего примера это

Спроч. = (2 443.45 + 112 729,67 + 33 818,9 + 1 285,93 + 7 041,33) · 0,1 =

15 487,58 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 5.2.4 Смета затрат на разработку проекта

Статья затрат	**Условное обозначение**	**Сумма, руб.**
Материалы и покупные изделия	$С_{мат}$	2 443.45
Основная заработная плата	$С_{зп}$	112729,67
Отчисления в социальные фонды	$С_{соц}$	33818,90
Расходы на электроэнергию	$С_{эл.}$	1285,93
Амортизационные отчисления	$С_{ам}$	7041,33
Непосредственно учитываемые расходы	$С_{нр}$	0,00
Прочие расходы	$С_{проч}$	15487,58
Итого:		**1703,41**

5.3 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 как:

$$I_{финр} = \frac{Ф_р}{Ф_{max}}, \quad (6)$$

где $I_{финр}$ – интегральный финансовый показатель разработки;

$Ф_р$ – стоимость исполнения;

$Ф_{max}$ – максимальная стоимость исполнения научно-исследовательского

проект.

$$I_{финр\ текущая\ разработка} = \frac{\mathbf{172806,86}}{170363,31} = 1,014$$

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

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

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

$$I_p = \sum a \cdot b\ , \qquad (7)$$

где I_p – интегральный показатель ресурсоэффективности;

a – весовой коэффициент;

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

Таблица 5.3.1 – 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:

$$I_{текущая\ разработка} = \frac{I_p}{I_{финр}} = 3,75$$

$$I_{вариант\ исполнения\,1} = \frac{I_р}{I_{финр}} = 3$$

Сравнительная эффективность проекта:

$$Э_{ср} = \frac{I^{p}_{фа}}{I^{p}_{фb}}, \quad (8)$$

где $Э_{ср}$– сравнительная эффективность проекта;

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

$I_{фb}{}^{p}$ – интегральный технико-экономический показатель аналога.

$Э_{ср\,1} = \frac{3{,}75}{3} = 1{,}25$

Отразим результаты расчета сравнительной эффективность разработки в таблице 5.3.2.

Таблица 5.3.2 – 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	

В ходе оценки бюджета затрат научного исследования и определения интегрального финансового показателя, показателя ресурсоэффективности можно сделать вывод, что рассчитанный финансовый показатель и интегральный показатель эффективности показали, что это проект является эффективным.

Итак, учебное время, посвященное изучению программного обеспечения ISaGRAF, до сих пор было минимальным и охватывало только теоретические курсы. Разработанный проект позволяет продемонстрировать практическое применение системы управления КРОСС и тем самым улучшить знания студентов в соответствующих направлениях современных исследований систем автоматизации.

Поэтому проект не ориентирован на экономические результаты, а направлен на оказание дополнительной программно-методической поддержки студентам направления 15.03.04.

ЗАКЛЮЧЕНИЕ

В результате выпускной квалификационной работы были:

- изучены различные программные языки в системе ISaGRAF, а именно языке FBD, LD, ST и IL;
- рассмотрен и изучен контролер «КРОСС»;
- получены практические навыки по программированию контроллеров «КРОСС» в системе ISaGRAF;
- разработан проект методического указания по лабораторной работе, описывающее подробно работу с системой ISaGRAF и программирование на различных языках контроллера «КРОСС»;
- разработанное методическое указание (приложение А) будет использоваться на лабораторных работах для студентов направления 15.03.04 «Автоматизация технологических процессов и производств».

Список используемой литературы

1 Контроллер «КРОСС»: Руководство по эксплуатации ЯЛБИ.421457.018РЭ. — Чебоксары, ОАО «Завод электроники и механики»— 111 с.

2 GRST [2] GFREEZE [2]

3 Контроллеры «КРОСС», ТРАССА-500, Р-130ISA: Руководство по эксплуатации. Система технологического программирования ISaGRAF ЯЛБИ.420146.001РЭ. — Чебоксары, ОАО «Завод электроники и механики». — 114 с.

4 ISaGRAF Version 3.4. Руководство пользователя.

5 Любашин А.Н. Что такое ISaGRAF? // Мир компьютерной автоматизации On Line [Электронный ресурс]. - Режим доступа: http://www.mka.ru. - Загл. с экрана.

6 Спирин Н.А., Лавров В.В. Информационные системы в металлургии: Конспект лекций (отдельные главы из учебника для вузов). - Екатеринбург: Уральский государственный технический университет - УПИ, 2004. - 495 с.

7 Изерман Р. Цифровые системы управления: Пер. с англ.- М.: 1984.-541с.

8 Безопасность жизнедеятельности/Под ред. С.В. Белова. - М.: Высш.шк.,1999.

9 Охрана окружающей среды /Под ред.С.В.Белова.-М.:Высш.шк.,1991 г. 9 СанПиН 2.2.2/2.4.1340-03 "Гигиенические требования к персональным электронно-вычислительным машинам и организации работы". Утверждены 30.05.2003.

10 СанПиН 2.2.4.548-96 "Гигиенические требования к микроклимату производственных помещений". Утверждены. 1.10.1996.

11 Сибаров Ю. Г.и др. Охрана труда в вычислительных центрах. - шк.,1991 - М.:Машиностроение,1990.

12 СанПиН 2.2.2/2.4.1340-03 ”Гигиенические требования к естественному,

искусственному и совмещенному освещению жилых и общественных зданий”

13 Безопасность жизнедеятельности. Расчёт искусственного освещения. Методические указания к выполнению индивидуальных заданий для студентов дневного и заочного обучения всех специальностей. — Томск: Изд. ТПУ, 2000.

14 ПУЭ /Изд. Главэнергонадзор России, 6-е изд. — М.,1998.

15 Баранов А.Н., Пчелинцев В.А. Пожарная безопасность: Учебное пособие. - М.: издательство АСВ, 1997 — 176 с.

16 Долин П.А. Справочник по технике безопасности. - М.: Энергоатомиздат, 1984. - 824

Приложение А

(Обязательное)

Методические указания к выполнению лабораторной работы «Программно-методическое обеспечение для изучения программного пакета ISaGRAF»

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

УТВЕРЖДАЮ
и.о. Директора ИШИТР

_____А.Ю. Демин
«__»_____2022 г.

Цзян Ялун

Программно-методическое обеспечение для изучения программного пакета ISaGRAF

Методические указания к выполнению лабораторных работ для студентов IV курса, обучающихся по направлению 15.03.04«Автоматизация технологических процессов и производств»

Издательство
Томского политехнического университета
2022

УДК 000000
ББК 00000
А00

Цзян Ялун
С46 Программирование контроллере КРОСС в системе ISaGRAF на языках FBD,LD,ST и IL: методические указания к выполнению лабораторной №4 по курсу «Автоматизация технологических процессов и производств»/ Цзян Ялун, Кузьминская Е.В. Скороспешкин М.В.; Томский поли- технический университет. – Томск : Изд-во Томского политехни- ческого университета, 2022. – 59 с.

УДК 000000
ББК 00000

Методические указания рассмотрены и рекомендованы к изданию методическим семинаром отделения ОАР ИШИТР
«__» _____2022 г.

Зав. отделением <название>
кандидат технических наук _____*И.О. Фамилия*

Председатель учебно-методической
комиссии _____*И.О. Фамилия*

Рецензент

Заместитель генерального директора ОАО «НИКИ» г. Томска
С.А. Окунев

© ФГАОУ ВО НИ ТПУ, 2022
© Цзян Ялун, Кузьминская Е.В.
Скороспешкин М.В., 2022

Цель работы: Описать процесс создания и отладки нескольких программ на четырех разных языках в пакете ISaGRAF для ознакомления студентов

1 Примеры создания программ в системе ISaGRAF на языке FBD

1.1 Создание программы логического ИЛИ на языке FBD

Создайте программу ISaGRAF в FBD для реализации логического ИЛИ в режиме моделирования, т.е. когда сигналы не вводятся и не выводятся непосредственно на контроллер, а используется симулятор ISaGRAF.

Для запуска системы разработки ISaGRAF дважды щелкните левой кнопкой (ЛК) мыши на иконке isagraf на рабочем столе или воспользуйтесь меню Windows «ПУСК / ПРОГРАММЫ»/ isagraf. Откроется менеджер управления проектами (рисунок 1.1).

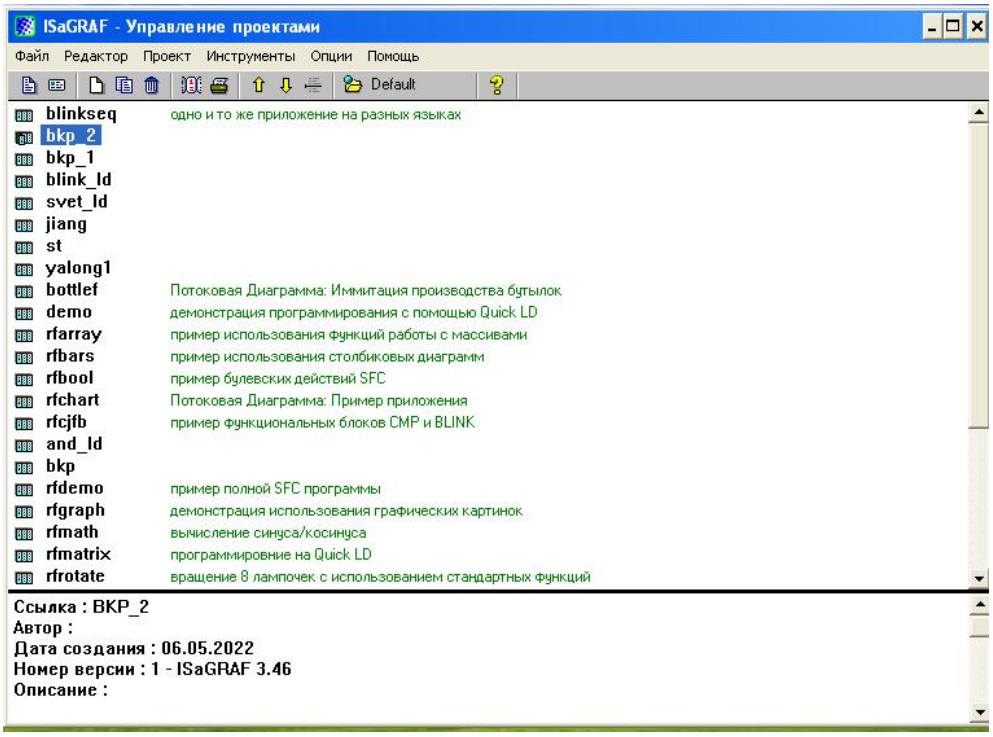


Рисунок 1.1 – Окно менеджера управления проектами ISaGRAF

Чтобы создать новый проект, используйте команду меню «Файл/Новый» или кнопку «Создать новый проект». В появившемся окне (рисунок 1.2) введите имя вашего проекта, например, «BKP_1» (обязательно используйте

английский язык), конфигурацию В/В выберите (ничего) и нажмите кнопку Принять.

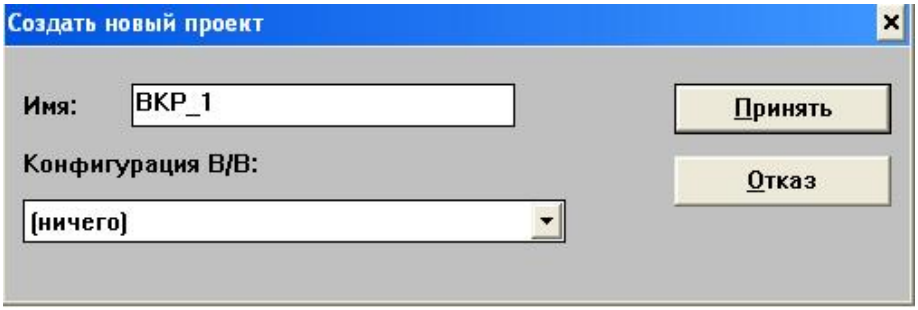


Рисунок 1.2 – Создание нового проекта

После того как ваш проект появится в списке проектов. Двойной щелчок на имени проекта откроет окно "Программы" Менеджера программ (рисунок 1.3).

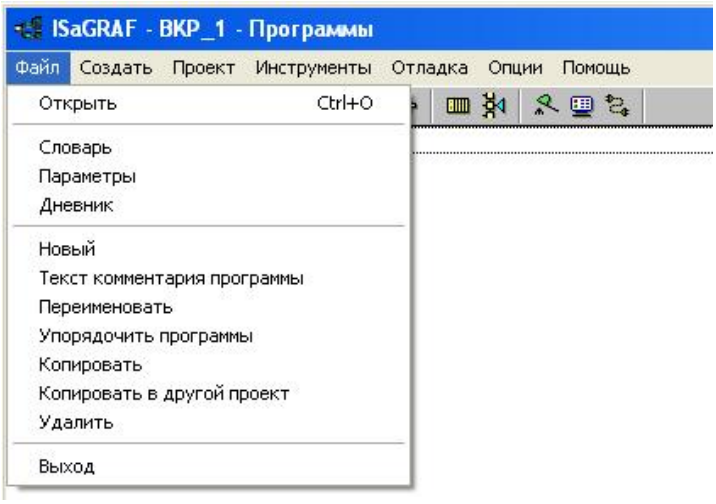


Рисунок 1.3 - Окно Менеджера Программ

Теперь открыто окно менеджера программ, которое является пустым, так как ни одна программа не была определена.Для создания новой программы используйте команду меню Менеджера Программ «Файл/Новый» или кнопку Создать новую программу. В появившемся окне, приведенном на рисунке 1.4, присвойте новой программе имя, например BKP_FBD, комментарий (не обязателен), выберите язык FBD: Функциональная Блочная Диаграмма, стиль Begin : Основная программа и нажмите кнопку «Принять».

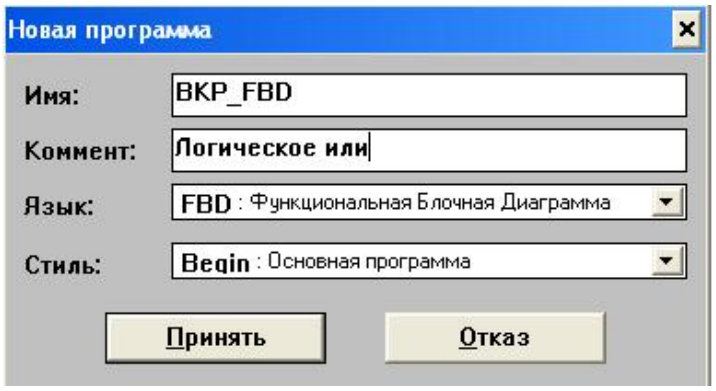


Рисунок 1.4 – Диалоговое окно создания новой программы

Получите доступ к этой программе, дважды щелкнув по названию программы BKP_FBD. Появиться окна редактора FBD/LD, приведенное на рисунке 1.5, в котором необходимо ввести и отредактировать программу.

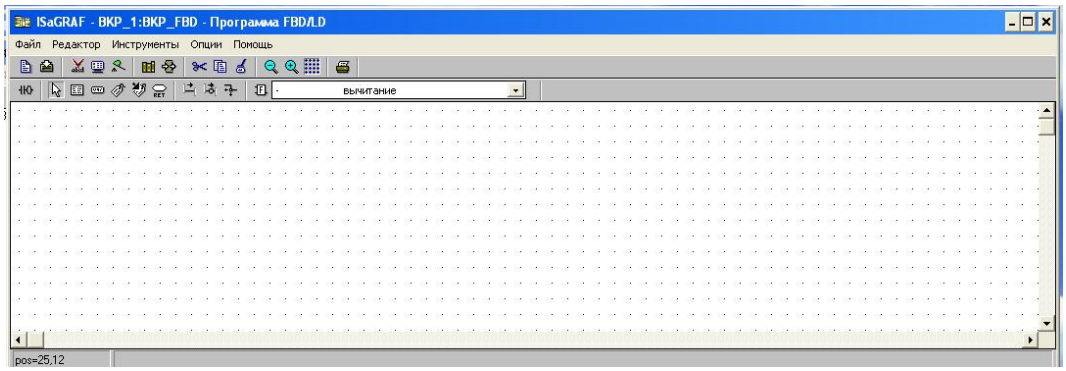


Рисунок 1.5 – Окно редактора FBD/LD

Все переменные должны быть объявлены перед вводом программы. Исп ользуйте команду меню «Файл/Словарь» или кнопку Словарь, чтобы открыть словарь. Появившееся окно словаря показано на рисунке 1.6.

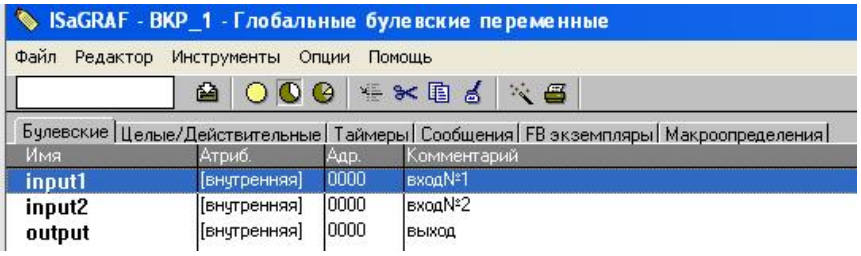


Рисунок 1.6 – Окно словаря

В окне словаря выберите вкладку Булевские и произведите двойной щелчок в выделенной верхней строке рабочего поля, или можно воспользоваться кнопкой Вставить объект. В появившемся окне (рисунок 1.7) задайте параметры переменной: имя input1, комментарий (комментарий

необязателен), атрибуты: Внутренняя, начальное значение False (по умолчанию) и нажмите кнопку «Сохранить».

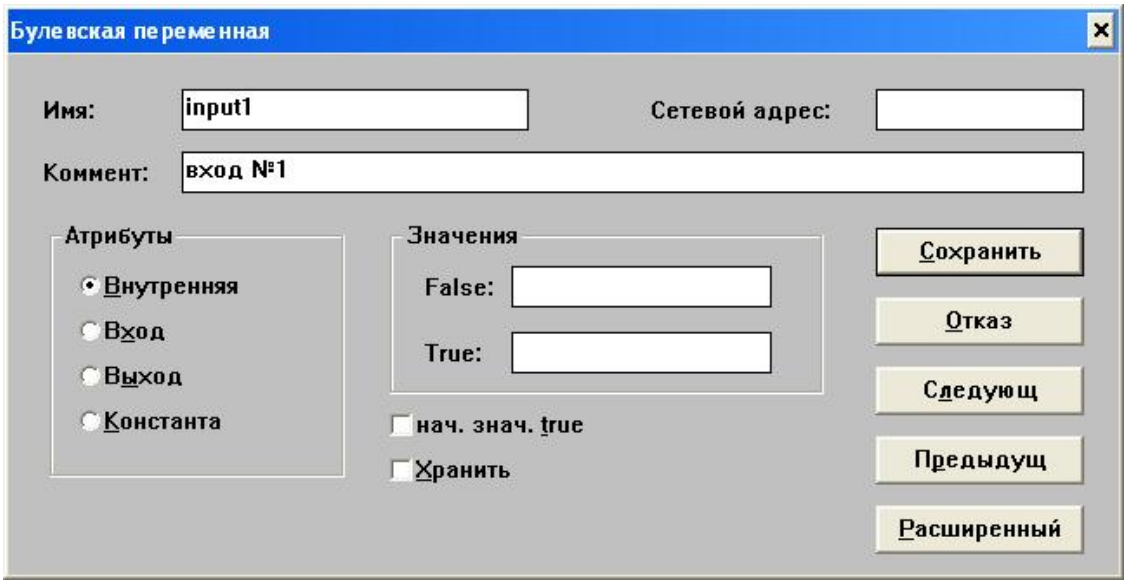


Рисунок 1.7 – Окно задания параметров переменной input1

Аналогично установите параметры второй входной переменной input2 и выходной переменной output (рисунки 1.8, 1.9). Выберите «Файл/Сохранить» в меню словаря, чтобы сохранить все изменения и закрыть словарь.

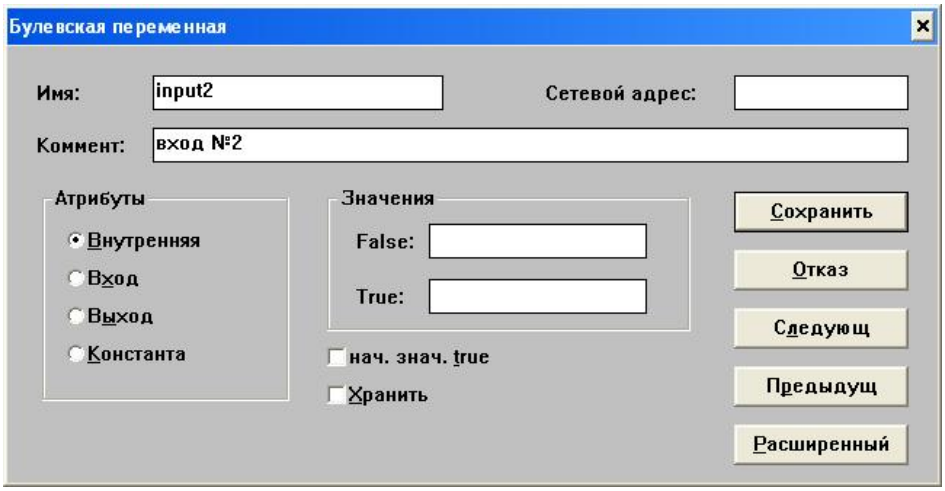


Рисунок 1.8 - Окно задания параметров переменной input2

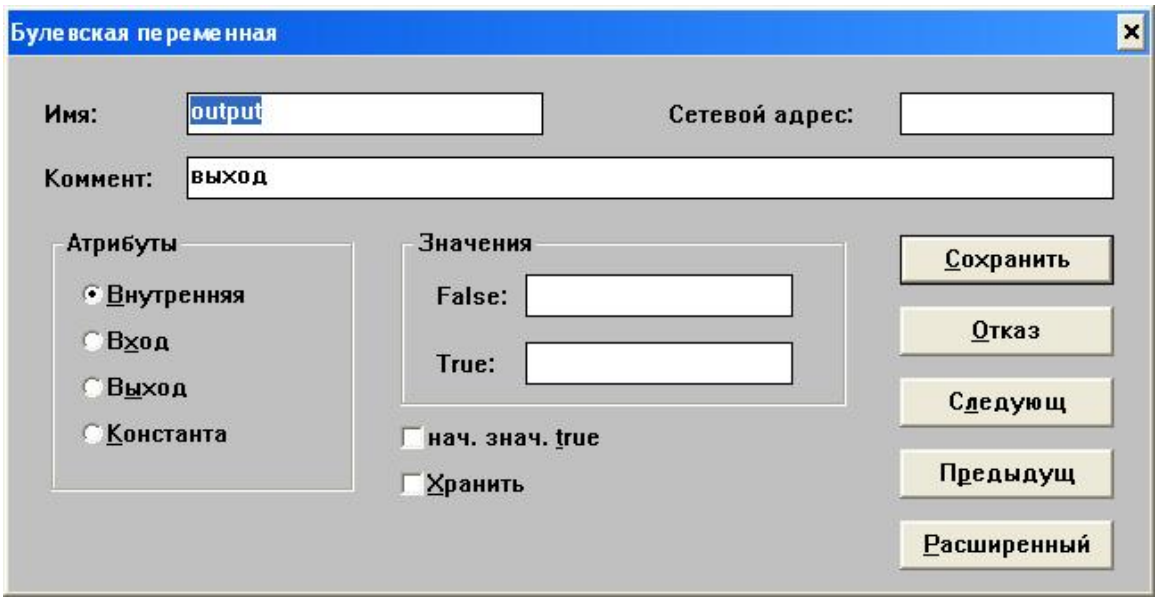


Рисунок 1.9 - Окно задания параметров переменной output

Теперь необходимо заняться вводом и редактированием программы. Вернитесь в окно редактора FBD/LD «Программа FBD/LD», приведенного на рисунке 1.10.

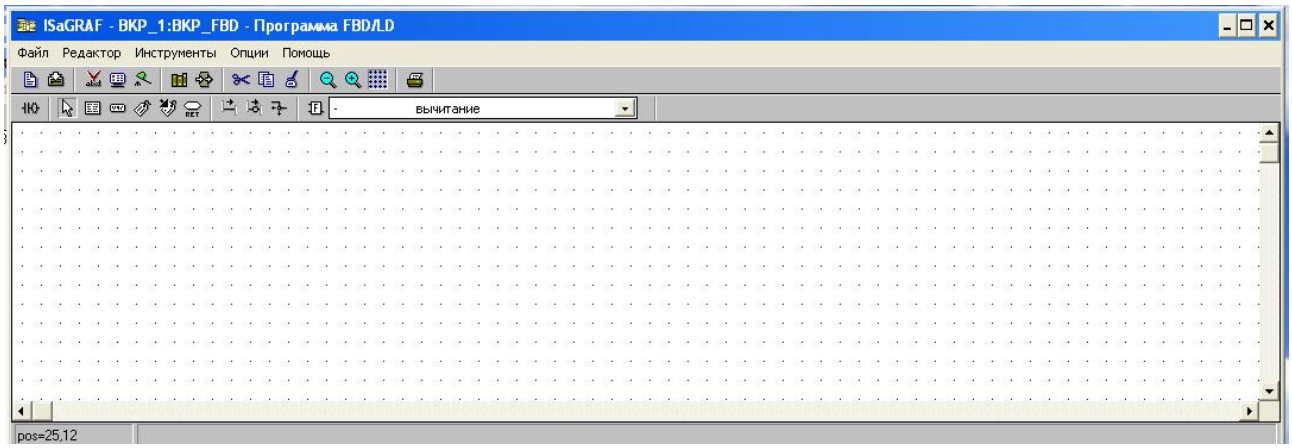


Рисунок 1.10 – Редактор FBD/LD

Выберите в панели управления редактора функциональный блок логического ИЛИ (>=1 – булевский OR) , как показано на рисунке 1.11.

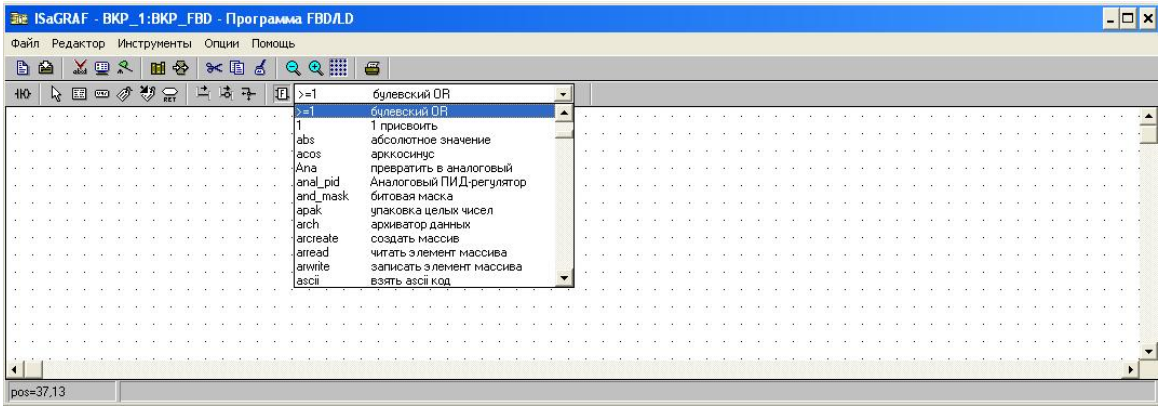


Рисунок 1.11 – Выбор блока булевский OR

Нажмите левую кнопку мыши в белой рабочей области редактора, и появится функциональный блок булевский OR. Если затем дважды щелкнуть по изображению блока, откроется окно, в котором можно изменить свойства блока, для чего можно изменить количество входов (рис. 1.12). Оставьте количество входов по умолчанию, т.е. 2.

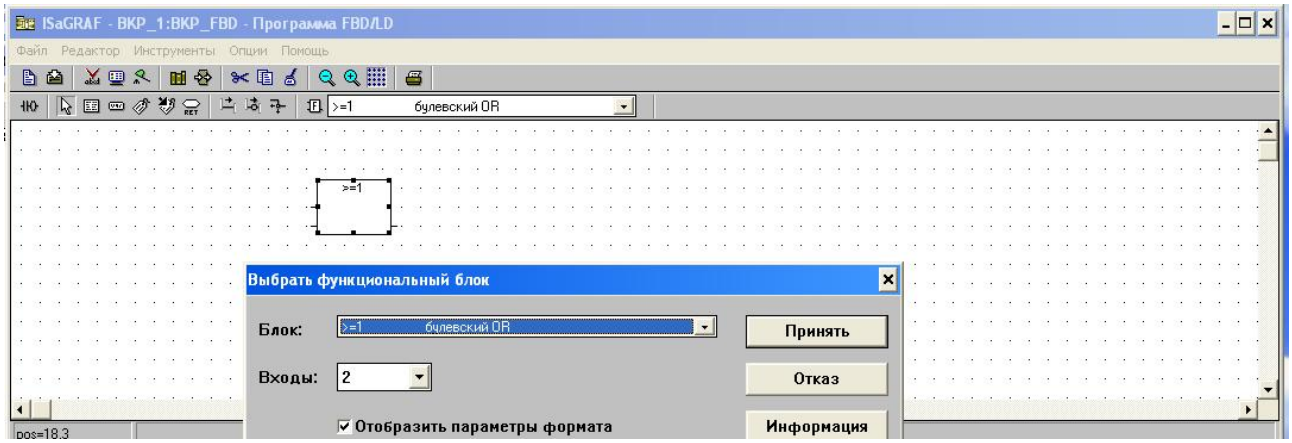


Рисунок 1.12 – Окно свойств блока булевский OR

Вход и выход блока булевский OR теперь должны быть связаны с переменными, которые вы ранее объявили в словаре.Выберите в панели управления иконку «Вставить» переменную, нажмите ЛК мыши в свободном месте рабочей области редактора, появиться окно выбора переменной, в котором выберите переменную input1 и нажмите кнопку «Принять» (рисунок 1.13).

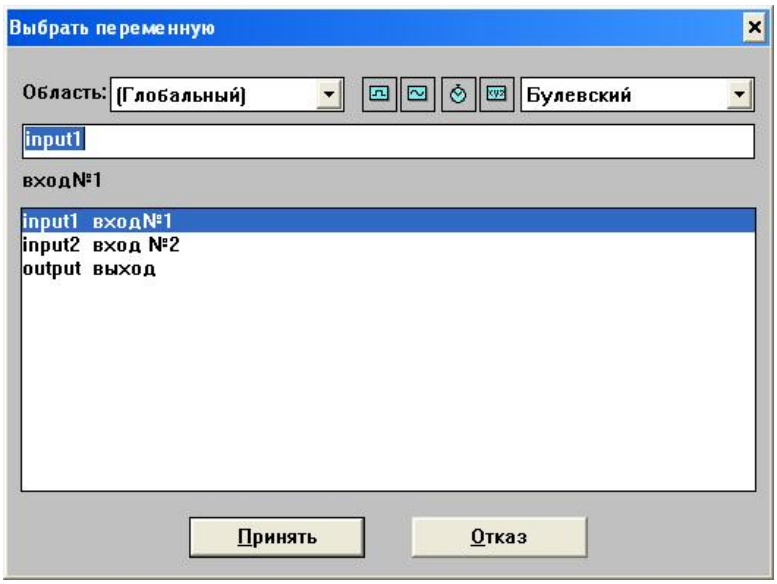


Рисунок 1.13- Окно выбора переменной

В рабочей области появиться изображение переменной в прямоугольнике с закругленными углами, то есть переменная input1 вставлена. Далее снова щелкните ЛК мыши в свободном месте рабочей области экрана, в появившемся окне выбора переменной выберите input2, нажмите кнопку «Принять» и будет вставлена переменная input2. Аналогичным образом вставьте в рабочую область переменную output.

После этого нажмите в панели управления редактора нажмите иконку Выбрать, нажмите ЛК мыши на изображение переменной input1 в рабочей области и не отпуская ЛК мыши аккуратно переместите переменную, установив ее перед верхним входом блока булевский OR. Аналогичным образом разместите остальные переменные возле соответствующих входов и выхода функционального блока.

Теперь соедините переменную input1 с верхним входом блока линией связи. Для этого нажмите на иконке Начертить линию связи, затем ЛК мыши нажмите на правом краю изображения переменной и, не отпуская ЛК мыши проведите линию от переменной до верхнего входа блока булевский OR. Аналогичным образом соедините остальные переменные с соответствующим входом и выходом блока. После этого программа примет полностью завершенный вид, приведенный на рисунке 1.14.

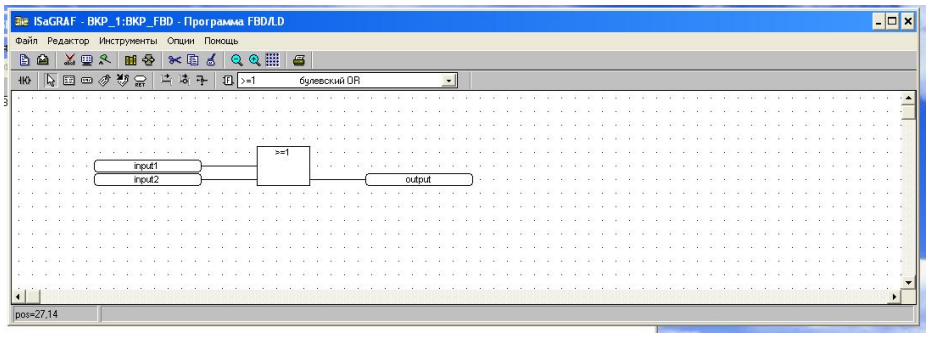


Рисунок 1.14 – Полностью введенная программа логического ИЛИ на языке FBD

После того как программа создана и переменные объявлены, необходимо проверить ее на правильность синтаксиса написания, для чего выберите команду меню «Файл/Проверить». Если ошибок нет, выйдите из генератора кода. Сохраните программу командой «Файл/Сохранить» из меню редактора FBD/LD, в диалоговом окне нажмите кнопку Принять.

Теперь закройте редактор FBD/LD, так как в ближайших шагах вы будете запускать симулятор. А симулятор нельзя запускать, если открыты дочерние окна, при не выполнении этого условия даже может случится «зависание» системы ISaGRAF. Сгенерируйте код командой меню «Создать/Создать» приложение или нажмите кнопку Создать код приложения в окне Диспетчера Программ. Теперь для проверки работоспособности запустите программу в режиме симуляции командой меню «Отладка/Симуляция» или кнопкой «Симуляция».

Во время работы симулятора можно открыть программу, дважды нажав ЛК мыши на имени программы BKP_FBD в окне Отладка программ, после этого откроется окно редактора, где можно будет посмотреть, как она работает (рисунок 1.15).

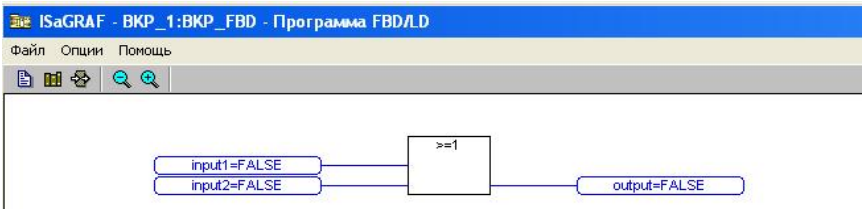


Рисунок 1.15 – Окно программы в режиме симуляции

Ч т о б ы изменить значение входной переменной во время работы программы, дважды щелкните на изображении переменной и в появившемся окне выберите TRUE или FALSE (рисунок 1.16).

Рисунок 1.16 – Окно изменения значения переменной

Экспериментируя с различными значениями input1 и input2, мы можем увидеть, что программа фактически выполняет операцию логического ИЛИ - то есть, значение выходной переменной становится TRUE, если хотя бы одна из входных переменных input1 или input2 равна TRUE. на рисунке 1.17 показана программа в режиме моделирования, когда input1 = TRUE.

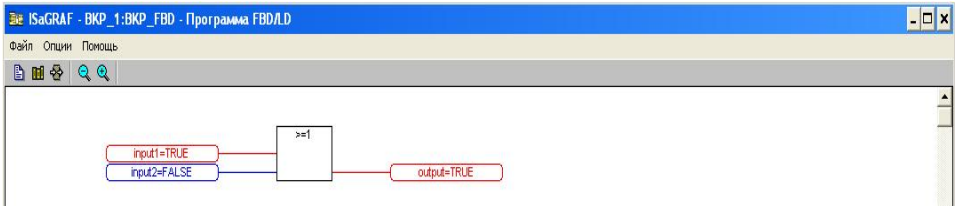


Рисунок 1.17 - Работа программы в режиме симуляции при значении input1=TRUE

Для завершения работы программы в окне Отладчик выберите команду «Файл/Выход».

1.2 Составление программы П-регурятора на языке FBD

Рассмотрим процесс составления программы, осуществляющую регулирование по П-закону и сигнализацию, реализованную с помощью пакета ISaGRAF.

Объявление переменных

Теперь необходимо открыть программу “p”, нажав дважды ЛК по ее имени.

Прежде чем выберем какие – либо блоки, необходимо объявить в Словаре все переменные используемые для реализации нашей программы. Вызов словаря осуществляется при помощи меню «Файл/Словарь» или кнопкой в менеджере программ. Переменные объявляются в соответствии с типом данных:

Булевские – логическая величина (рисунок 1.18);

Целые/Действительные – целая или действительная непрерывная величина (рисунок 1.19);

Таймеры – временная величина;

Сообщения – строка символов;

FB экземпляры – экземпляры функциональных блоков, используемые при программировании на языках ST и IL;

Макроопределения – синонимы, используемые для замены текстовых трок.

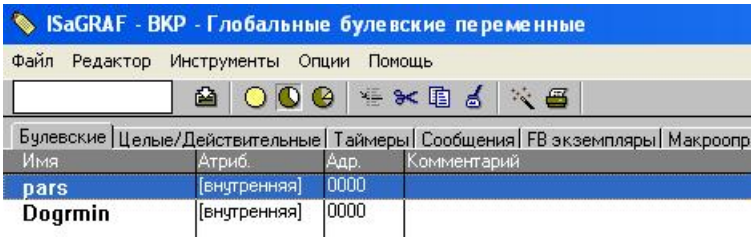


Рисунок 1.18- Глобальные булевские переменные

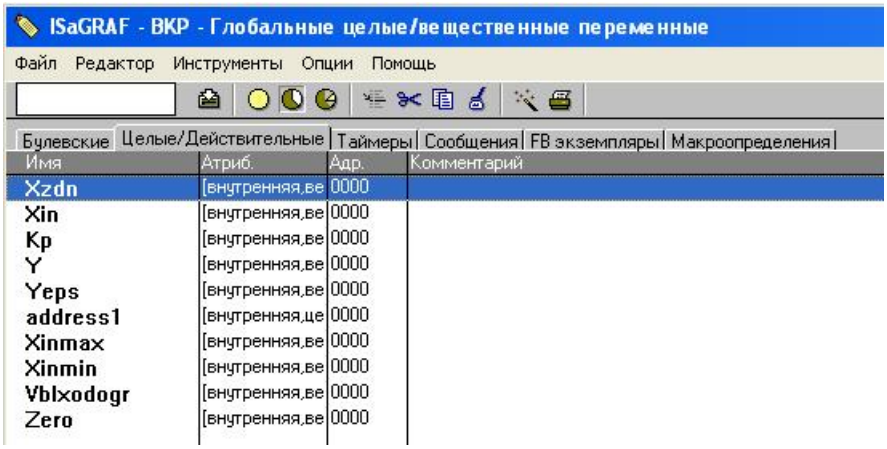


Рисунок 1.19- Глобальные целые/вещественные переменные

Программа ввода/вывода аналоговых сигналов.

Вызываем функциональный блок ввода-вывода аналоговых сигналов с модуля УСО КРОСС (craio), размещаем его на экране и осуществляем соединение блока с переменными.

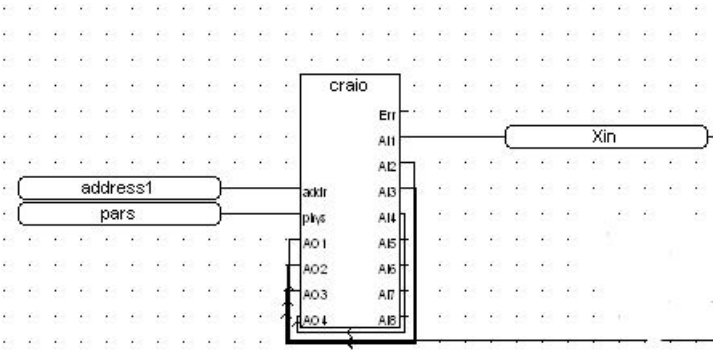


Рисунок 1.20 - Программа ввода/вывода аналоговых сигналов

Блок П - регулирования.

Для реализации П-регулятора воспользуемся двумя простыми операциями: вычитанием и умножением. Соответственно Сначала находим ошибку регулирования Y_{eps}, используя разницу между задающим сигналом X_{zdn} и выходом объекта управления X_{in}. Блок умножения выполняет функцию П-регулятора, умножая входящий в блок регулятора сигнал ошибки Y_{eps} на коэффициент пропорциональности K_P. На выходе регулятора получаем Y.

В итоге программа П-регулятора будет выглядеть как на рисунке 1.21.

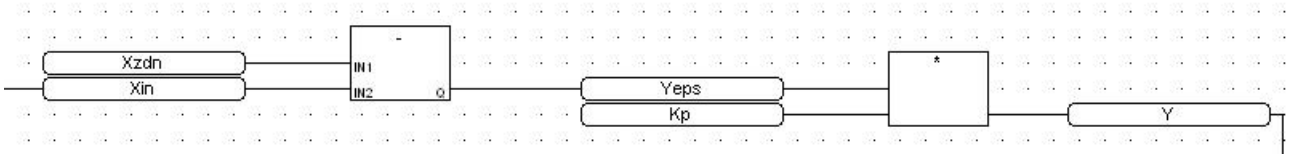


Рисунок 1.21 - Программа П-регулятора

Блок сигнализации.

Вызываем функциональный блок сигнализации с модуля УСО КРОСС (ogr), размещаем его на экране и осуществляем соединение блока с переменными.

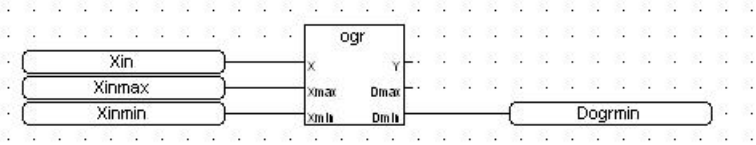


Рисунок 1.22 - Блок сигнализации

Проверка на ошибки

После того как программа была написана, необходимо проверить её на ошибки. Проверка осуществляется с помощью кнопки . Результат проверки

будет показан в отдельном окне (рисунок 1.23).

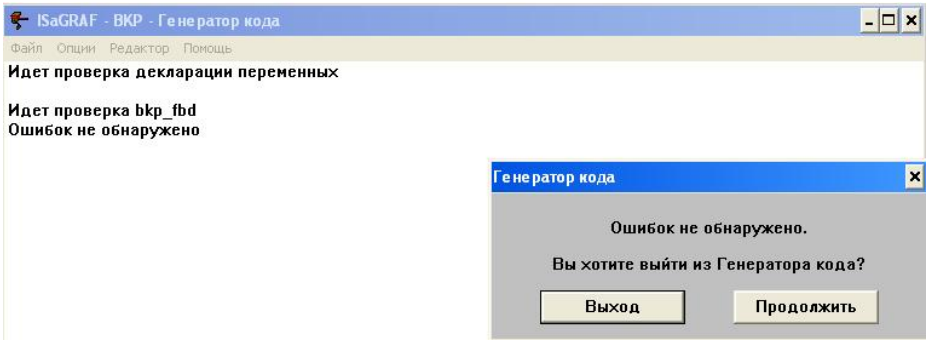


Рисунок 1.23 - Результат проверки на ошибки программы

Настройка опций компилятора

Следующим шагом является установление связи между программой и контроллером. Для этого перейдите в меню «Создать/Опции Компилятора» (рисунок 1.24) и выберите тип процессора. Наш контроллер содержит процессор типа Intel, поэтому необходимо проверить код TIC code for Motorola (рисунок 1.25).

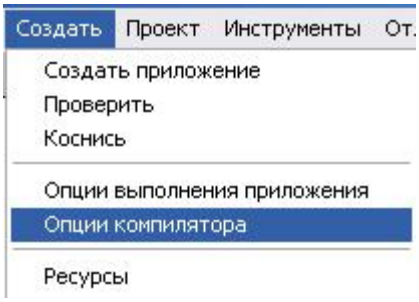


Рисунок 1.24 - Меню создать

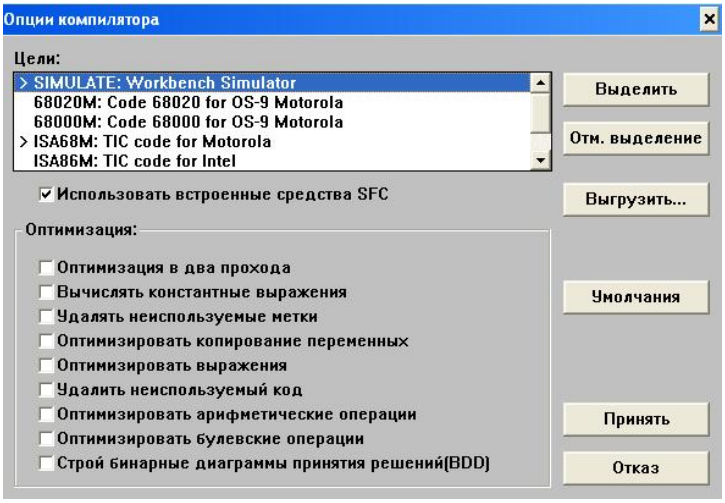


Рисунок 1.25 - Опции компилятора

Настройка связи

Для настройки связи в контекстном меню окошка программы выбираем «Отладка» и в раскрывшимся списке нажимаем на «Установление связи». В открывшемся окне (рисунок 1.26) настраиваем параметры связи ПК - ПЛК, далее жмем на «Установка» и прописываем интернет адрес контроллера (рисунок 1.27).

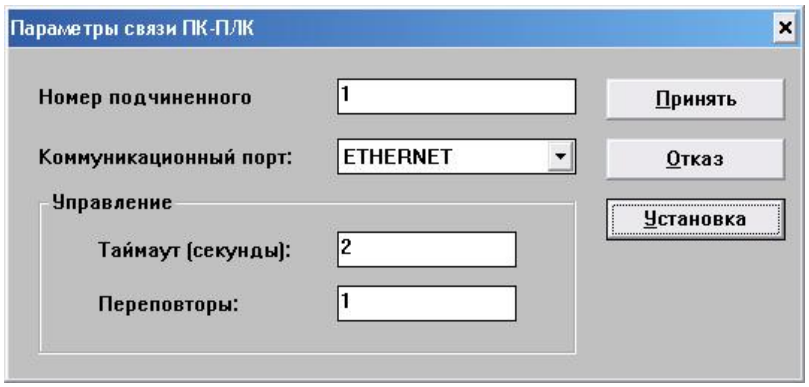


Рисунок 1.26 - Параметры связи ПК-ПЛК

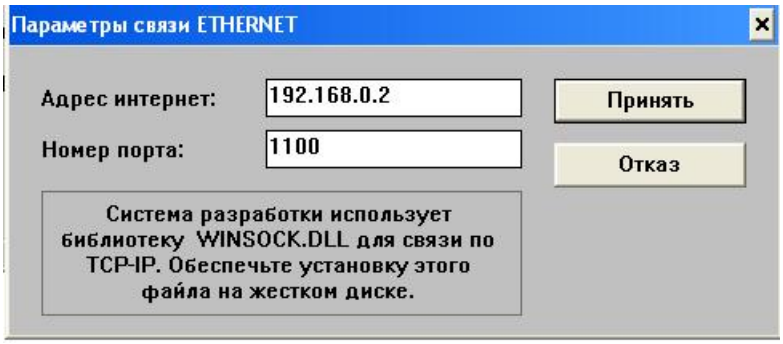


Рисунок 1.27 - Параметры связи ETHERNET

Генерация кода приложения

После завершения всех настроек используйте кнопку, расположенную на панели управления, чтобы создать код приложения для загрузки в контроллер. Результат будет отображен в отдельном окне (рисунок 1.28).

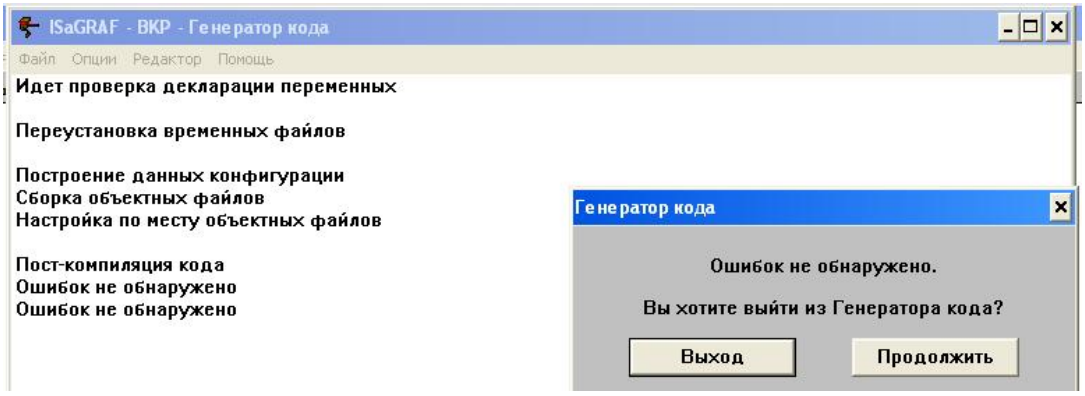


Рисунок 1.28 - Окно генератора кода

Окно отладчика

Поскольку мой компьютер не был подключен к контроллеру, я открыл окно отладки для отладки(рисунок 1.29).

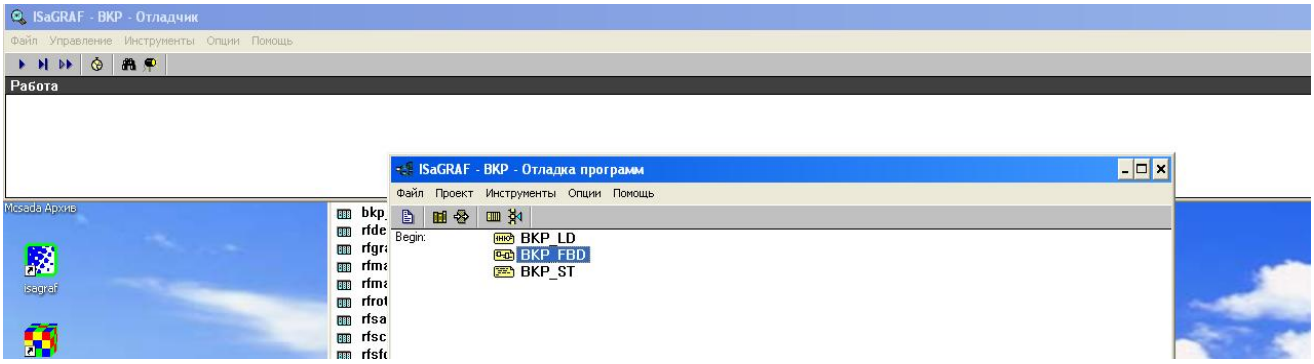


Рисунок 1.29 - Окно отладки

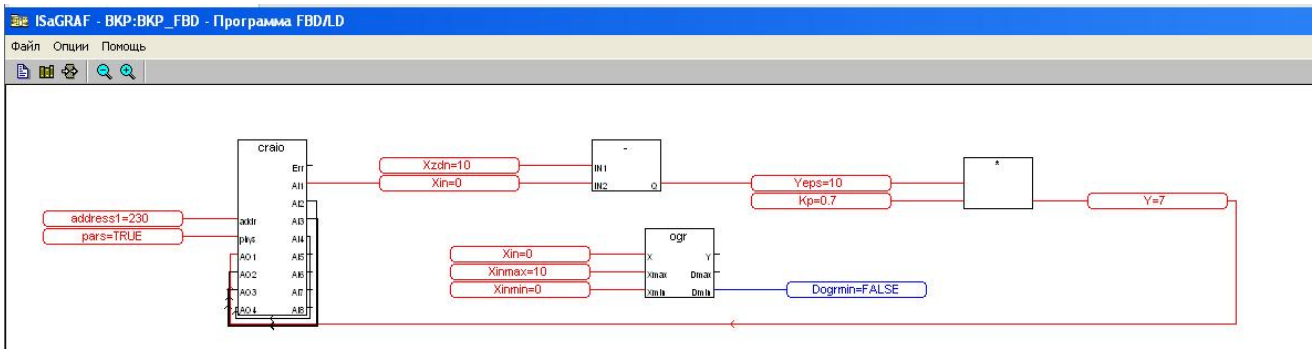


Рисунок 1.30 - Рабочая программа в среде ISaGRAF

Внимание :Здесь модуль УСО КРОСС (craio) только вход,не может получить выход(рисунок 1.31)

craio	КРОСС дискретный ввод
craio	КРОСС дискретный ввод
crdin16	КРОСС дискретный ввод
crdout16	КРОСС дискретный ввод
crgetid	возвращает id для модуля с указ
crlaster	get last error for cross module
CTD	счетчик вниз
CTU	счетчик вверх
CTUD	счетчик вверх/вниз
data_r_a	receive 16 integer variables
data_r_b	receive 16 digital variable
data_r_r	receive 16 real variables
data_s_a	send 16 integer variable
data_s_b	send 16 digital variable

Рисунок 1.31 - Блока craio

1.3 Составление программы ПИД-регурятора на языке FBD

1.Объявление переменных

ISaGRAF - BKP_3 - Глобальные булевские переменные

Файл Редактор Инструменты Опции Помощь

Булевские | Целые/Действительные | Таймеры | Сообщения | FB экземпляры | Макроопределения

Имя	Атриб.	Адр.	Комментарий
phys	[константа]	0000	
Csb	[константа]	0000	Включение статистической балансировки
Cdb	[константа]	0000	Включение динамической балансировки
Czb	[константа]	0000	Сигнал запрета в направлении "Больше"
Czm	[константа]	0000	Сигнал запрета в направлении "Меньше"
Cruch	[внутренняя]	0000	Включение ручного режима
Dmax	[внутренняя]	0000	Ограничение по максимуму
Dmin	[внутренняя]	0000	Ограничение по минимуму
AlarmMax	[внутренняя]	0000	
AlarmMin	[внутренняя]	0000	
AlarmMaxBlink	[внутренняя]	0000	
AlarmMinBlink	[внутренняя]	0000	
DostMaxBlink	[внутренняя]	0000	
DostMinBlink	[внутренняя]	0000	
F	[внутренняя]	0000	

Рисунок 1.32 - Глобальные булевские переменные

ISaGRAF - BKP_3 - Глобальные целые/вещественные переменные

Файл Редактор Инструменты Опции Помощь

Булевские | Целые/Действительные | Таймеры | Сообщения | FB экземпляры | Макроопределения

Имя	Атриб.	Адр.	Комментарий
address1	[константа,цел	0000	Адрес craio
address2	[константа,цел	0000	Адрес crdout16,crdin16
Xin	[внутренняя,ве	0000	Входной сигнал
Yout	[внутренняя,ве	0000	Основной выход регулятора
Zadanie	[внутренняя,ве	0000	Уставка регулятора
Vdb	[константа,вещ	0000	Скорость динамической балансировки
Xdlt	[константа,вещ	0000	Размер зоны нечувствительности
Kp	[внутренняя,ве	0000	Коэффициент пропорциональности
Ti	[внутренняя,ве	0000	Постоянная времени интегрирования
Kd	[внутренняя,ве	0000	Постоянная времени дифференцирования
Xmax	[константа,вещ	0000	Уровень ограничения по максимуму
Xmin	[константа,вещ	0000	Уровень ограничения по минимуму
Xruch	[внутренняя,ве	0000	Ручное задание
Yeps	[внутренняя,ве	0000	Выход рассогласования
Yzdn	[внутренняя,ве	0000	Выход текущего задания
Xsigmax	[внутренняя,ве	0000	Максимум для сигнализации
Xsigmin	[внутренняя,ве	0000	Минимум для сигнализации
Zero	[внутренняя,ве	0000	Переменная с нулевым значением
Kpom	[константа,вещ	0000	
Impul	[внутренняя,ве	0000	
Y	[внутренняя,ве	0000	
Kpom1	[внутренняя,ве	0000	
Dosomax	[внутренняя,ве	0000	
Dosimin	[внутренняя,ве	0000	
Pomexa	[внутренняя,ве	0000	
alfa	[внутренняя,ве	0000	
Ypered	[внутренняя,ве	0000	

Рисунок 1.33 - Глобальные целые/вещественные переменные

ISaGRAF - BKP_3 - Глобальные таймеры

Файл Редактор Инструменты Опции Помощь

Булевские | Целые/Действительные | Таймеры | Сообщения | FB экземпляры | Макроопределения

Имя	Атриб.	Адр.	Комментарий
blink_period	[внутренняя]	0000	
Tblink1	[внутренняя]	0000	
Tblink2	[внутренняя]	0000	

Рисунок 1.34 - Глобальные Таймеры

2.Программа ввода/вывода аналоговых сигналов.

Для реализации данной программы нам потребуются следующие функциональные блоки:

1) Ввод-вывод аналоговых сигналов с модуля УСО КРОСС (craio), приведен на рисунке 1.35.

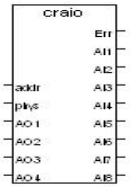


Рисунок 1.35 – Блок craio

Входы: addr (INT) – сетевой адрес модуля;

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

AO1-AO4(REAL) – значения выходных каналов модуля.

Выходы: AI1-AI8 (REAL) – значения входных каналов модуля;

Err (INT) – код ошибки.

Блок craio необходим нам для связи входных и выходных переменных непосредственно с входами-выходами аналогового модуля AIO-8/4.

2) Ввод дискретных сигналов с модуля УСО КРОСС (crdin16), приведен на рисунке 1.36.

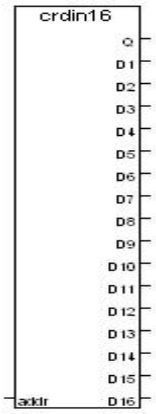


Рисунок 1.36 – Блок crdin16

Входы: addr (INT) – сетевой адрес модуля.

Выходы: D1-D16 (BOOLEAN) – значения входных каналов модуля;

Q (INT) – код ошибки.

Функциональный блок crdin16 необходим нам для связи входных переменных непосредственно с входами дискретного модуля DIO1-8/8.

3) Вывод дискретных сигналов с модуля УСО КРОСС (crdout16), приведен на рисунке 1.37.

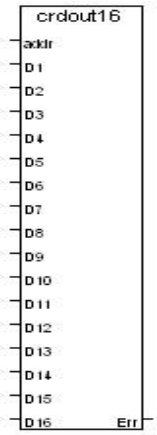


Рисунок 1.37 – Блок crdout16

Входы: addr (INT) – сетевой адрес модуля;

D1-D16(BOOLEAN) – значения выходных каналов модуля.

Выходы: Err (INT) – код ошибки.

Функциональный блок crdout16 необходим нам для связи выходных переменных непосредственно с выходами дискретного модуля DIO1-8/8.

4) Мерцающий сигнал (blink), приведен на рисунке 1.38.

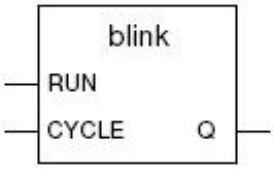


Рисунок 1.38 – Функциональный блок blink

Описание: генерирует периодический булевый сигнал.

Аргументы:

Входы: RUN (BOOLEAN) режим работы: TRUE – разрешение/FALSE –сброс выхода в FALSE

CYCLE (TMR) время периода;

Выход: Q (BOOLEAN) выходной периодический сигнал.

5) ограничение (OGR), приведен на рисунке 1.39.

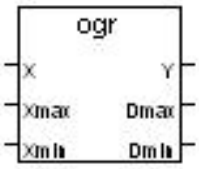


Рисунок 1.39 - Функциональный блок OGR

6) аналоговый ПИД-регулятор (anal_pid), приведен на рисунке 1.40.

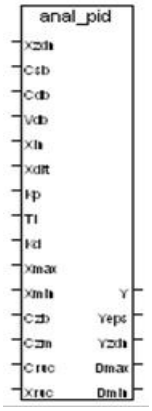


Рисунок 1.40 - Функциональный блок anal_pid

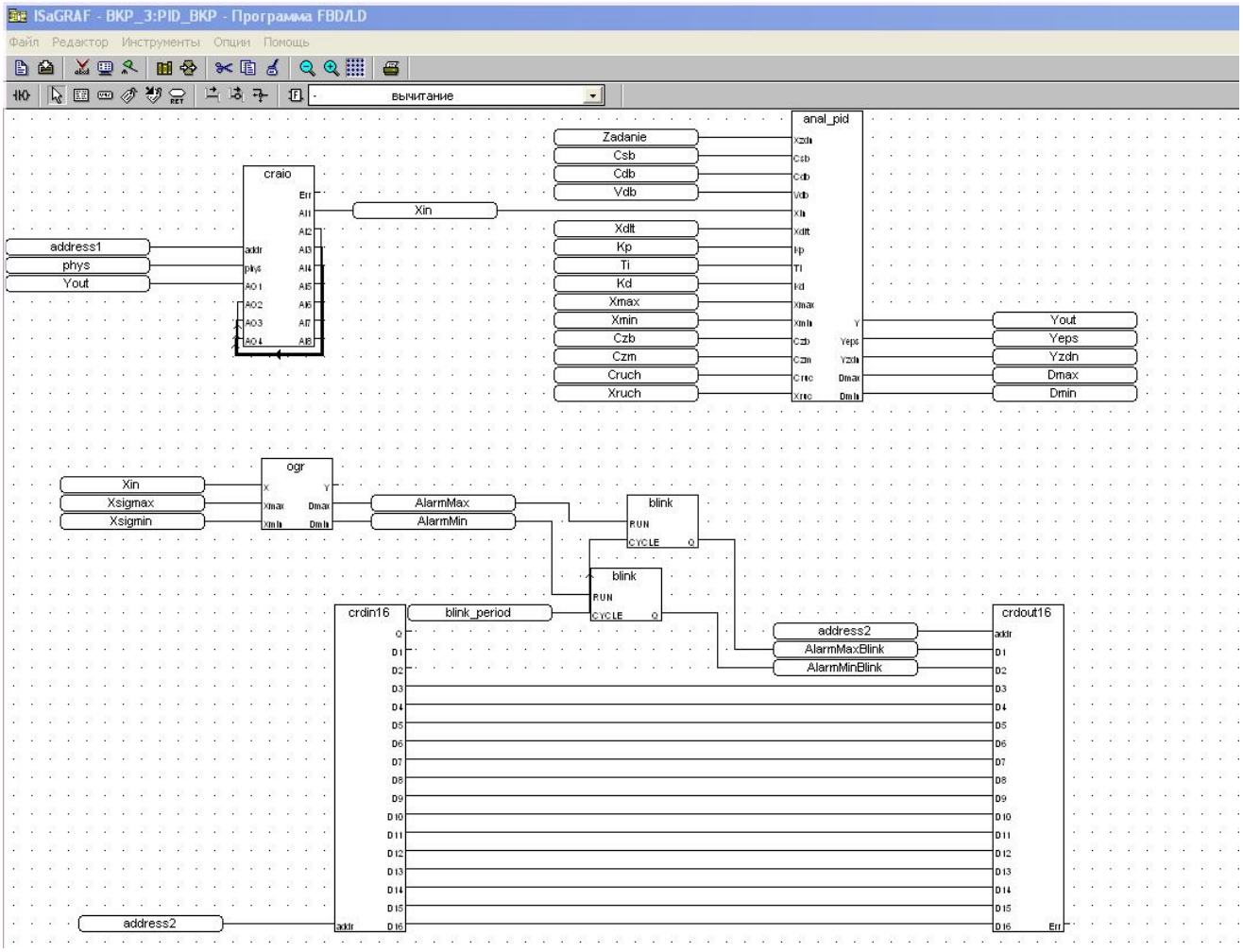


Рисунок 1.41– Программа реализации аналогового ПИД-регулятора на FBD.

После ввода программы необходимо выполнить ее проверку на правильность синтаксиса описания командой «Файл/Проверить». Если все нормально, нужно сохранить программу командой «Файл/Сохранить».

Теперь необходимо установить опции компилятора для правильной генерации кода приложения. Для этого, находясь в редакторе FBD необходимо зайти в меню Опции/Опции компилятора и двойным щелчком ЛК мыши установить галочку напротив TIC code for Motorola, как показано на рисунке 1.42. После этого нажать кнопку Принять.

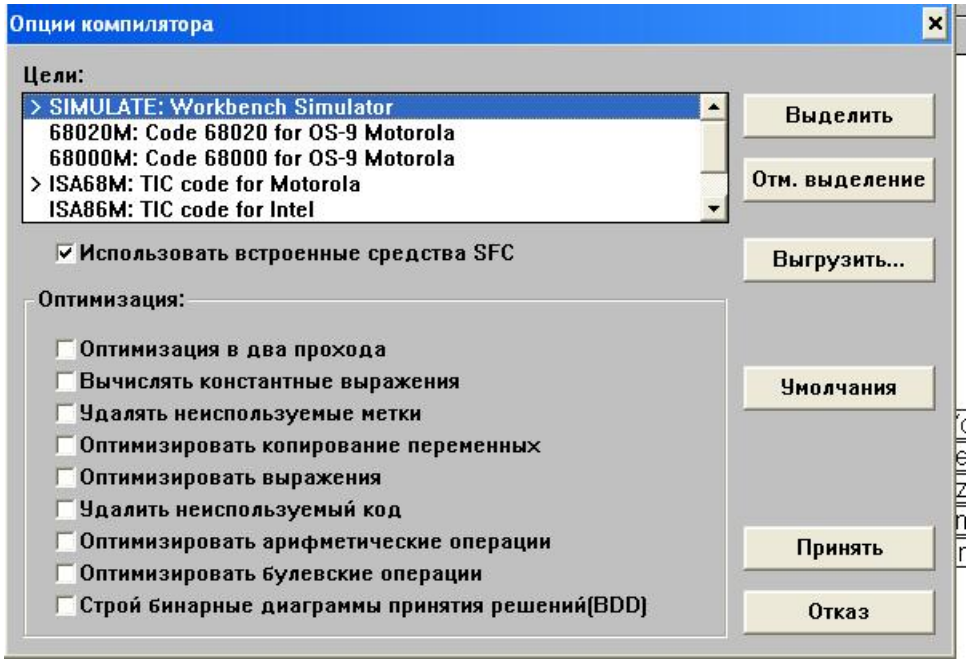


Рисунок 1.42 – Окно опций компилятора

Далее необходимо скомпилировать программу. Для этого нужно зайти в окно «Программы» и выбрать команду меню Создать/Создать приложение или нажать кнопку Создать код приложения. Команда создаст код приложения и проверит программу на наличие синтаксических ошибок. Если ошибок нет, нужно выйти из генератора кода.

Далее нужно установить связь между ПК и контроллером КРОСС. Для этого необходимо открыть окно ISaGRAF «Программы», зайти в меню Отладка/Установление связей. В появившемся окне (рисунок 1.43) установить коммуникационный порт ETHERNET, нажать кнопку Установка и в

следующем окне прописать интернет адрес 192.168.0.2 (это адрес контроллера), номер порта 1100.

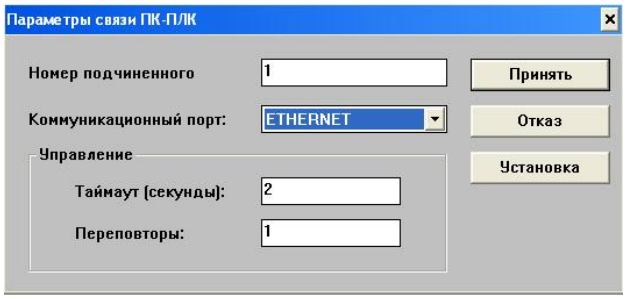


Рисунок 1.43 – Окно параметров связи ПК-ПЛК

Программа готова к отладке контроллером. Состояние программы в режиме отладки приведено на рисунке 1.44. Убедитесь, что САР действительно работает, для чего прямо в процессе работы программы измените уставку. Для этого выполните двойной щелчок ЛК мыши на изображении переменной Zadanie в окне Программа FBD/LD и задайте новое значение уставки.

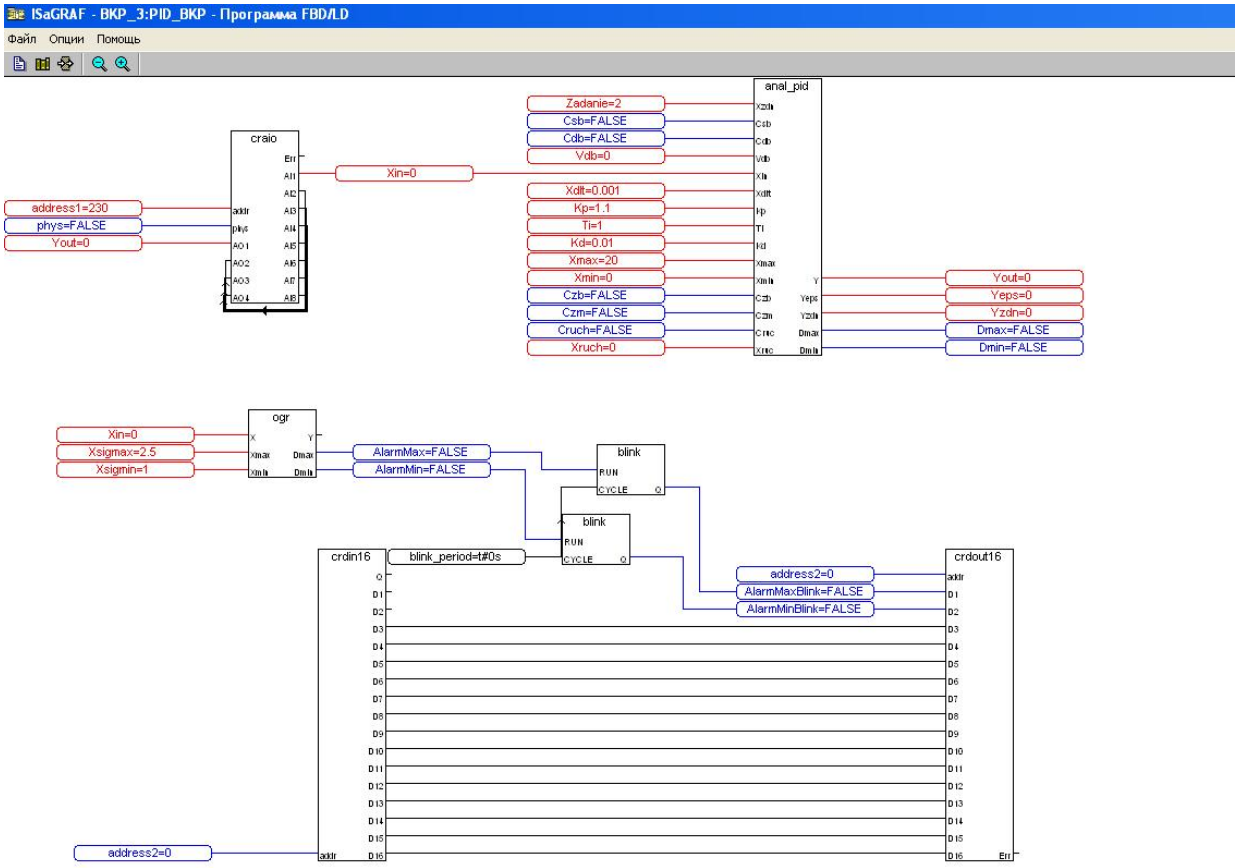


Рисунок 1.44 – Состояние программы аналогового ПИД-регулятора в режиме отладки

Пример создания программы в системе ISaGRAF на языке ST

1.4 Создание программы логического ИЛИ на языке ST

Выполняя последовательно действия, приведенные в п.1.1 по созданию проекта и программы, создайте новый проект с именем «BKP_1», а в нем новую программу с именем «BKP_ST», но при создании программы в диалоговом окне выберите язык не FBD, а ST.

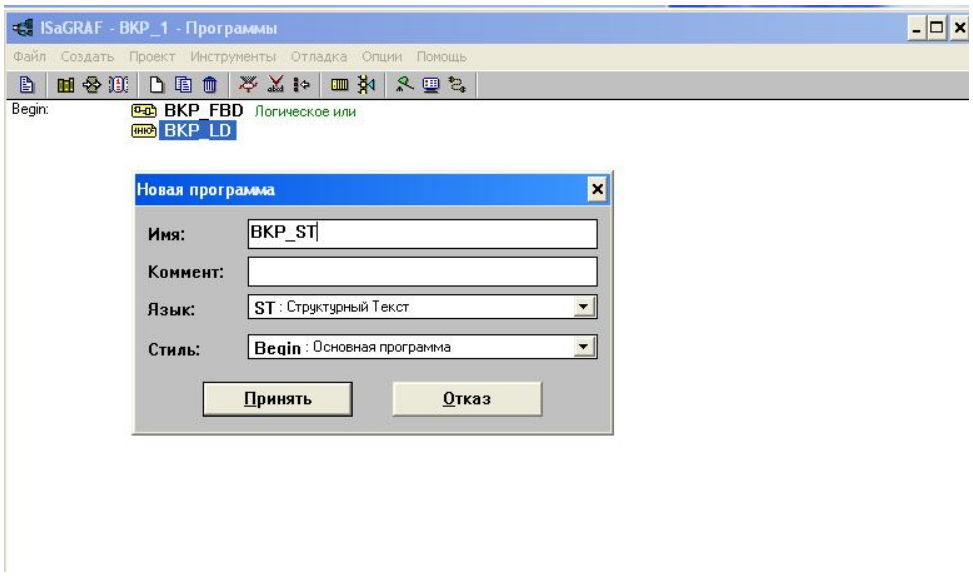


Рисунок 1.45 – Диалоговое окно создания новой программы

После объявления переменных введите программу в текстовый редактор ST (окно программы ST). Текст программы для ввода (список) показан ниже. Примечание: Текст программы содержит множество комментариев, заключенных в круглые скобки (**). Эти комментарии предназначены для облегчения понимания программы, но вводить их в редакторе не обязательно.

IF input1 OR input2 THEN

output:=TRUE;

END_IF;

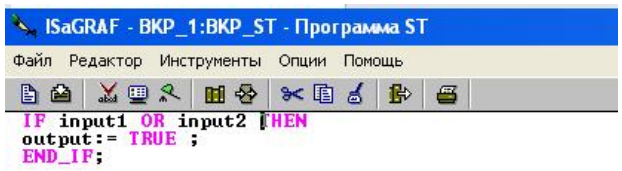


Рисунок 1.46 – Полностью введенная программа логического ИЛИ на языке ST

Действия по настройке параметров компилятора, установления связей с

контроллером, созданием кода приложения (проверкой) и отладкой программы необходимо осуществить в соответствии с действиями, выполняемыми при создании программы из раздела 3 данного методического пособия.

Понаблюдать за значениями переменных и изменять значение переменной Zadanie (Уставка) можно, если во время работы программы открыть Словарь (рисунок 1.47). Для изменения значения Задания (Уставки) выполните двойной щелчок ЛК мыши на имени переменной Zadanie и в появившемся окне задайте новое значение переменной.

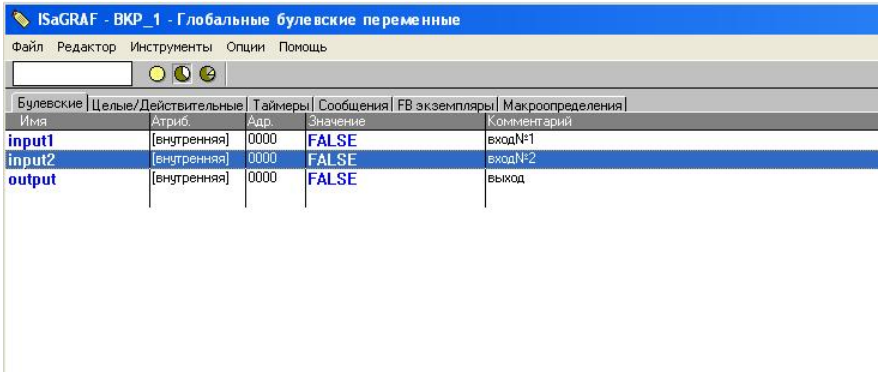


Рисунок 1.47 – Окно словаря во время работы программы.

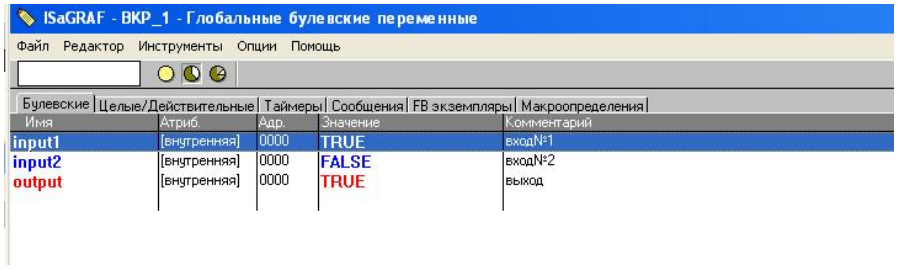


Рисунок.1.48 - Работа программы в режиме симуляции при значении input1=TRUE

1.5 Составление программы П-регурятора на языке ST

1.Объявление переменных

Теперь необходимо открыть программу “p”, нажав дважды ЛК по ее имени. Прежде чем выберем какие – либо блоки, необходимо объявить в Словаре все переменные используемые для реализации нашей программы. Вызов словаря осуществляется при помощи меню Файл, Словарь или кнопкой в менеджере программ. Переменные объявляются в соответствии с типом данных:

Булевские – логическая величина;

Целые/Действительные – целая или действительная непрерывная величина;

Таймеры – временная величина;

Сообщения – строка символов;

FB экземпляры – экземпляры функциональных блоков, используемые при программировании на языках ST и IL (рисунок 1.49);

Макроопределения – синонимы, используемые для замены текстовых строк.

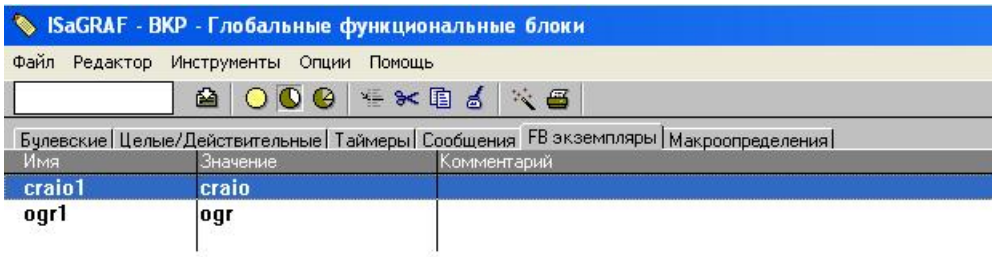


Рисунок 1.49 - Глобальные FB экземпляры

Написание кодов П-регурятора

```
craio1(address1,pars,Y,Zero,Zero,Zero);
Xin:= craio1.AI1;

Yeps:= Xzdn-Xin;
Y:= Yeps*Kp;

IF(Xinmin<=Xin AND Xin<=Xinmax)THEN
Dogrmin:= FALSE;
ELSE Dogrmin:= TRUE;
END_IF;
```

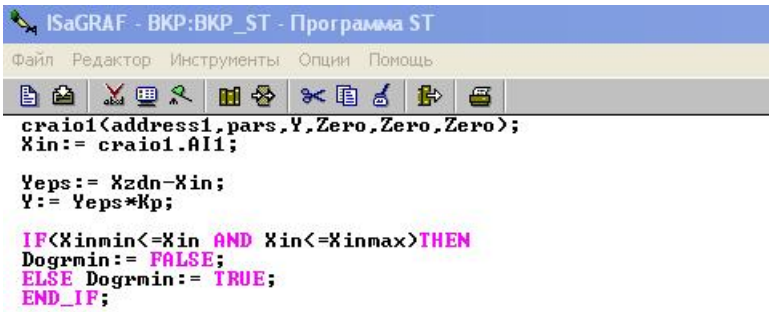


Рисунок 1.50 – Полностью введенная программа П-регурятора на языке ST

После того как программа создана и переменные объявлены, необходимо проверить ее на правильность синтаксиса написания, для чего выберите команду меню «Файл/Проверить». Если ошибок нет, выйдите из генератора кода(рисунок 1.51).

Рисунок 1.51 - Окно Генератор кода

Сохраните программу командой «Файл/Сохранить», в диалоговом окне нажмите кнопку «Принять» (рисунок 1.52).

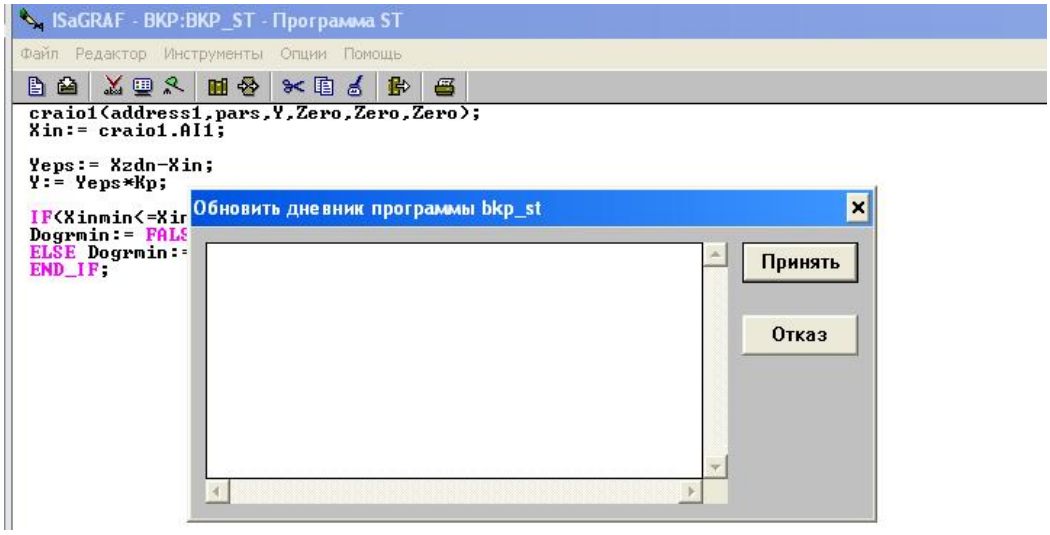


Рисунок 1.52 - Сохраните программу

После того как были закончены все настройки, сгенерируем код приложения для загрузки в контроллер контекстное меню «Создать/ Создать приложение» (рисунок 1.53). Далее необходимо произвести загрузку программы в контроллер, для этого необходимо запустить отладчик меню «Отладка/Симуляция» (рисунок 1.54), при этом на экране появится окно отладчика.

Рисунок 1.53 - Создать приложение

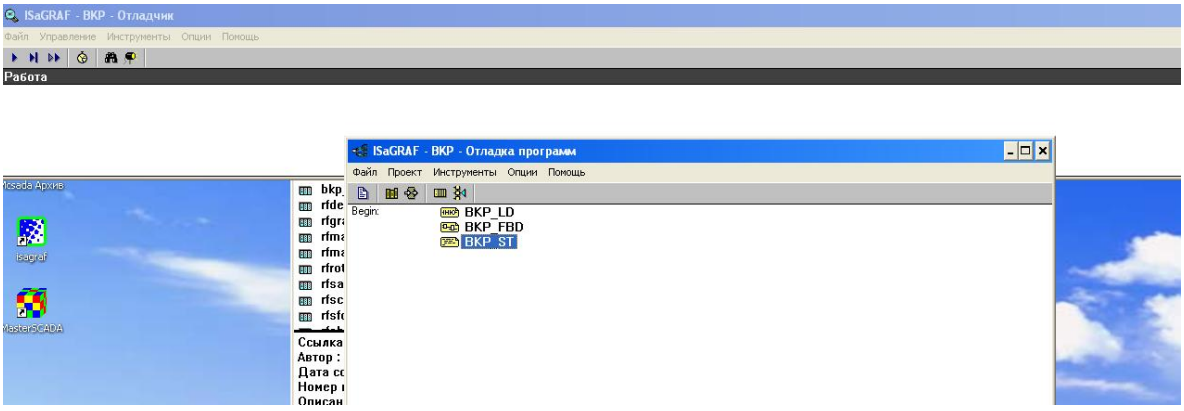


Рисунок 1.54 - Окно симуляцию

Во время работы программы открыть Словарь (рисунок 1.55). Для изменения значения Задания (Уставки) выполните двойной щелчок ЛК мыши на имени переменной и в появившемся окне задайте новое значение переменной.

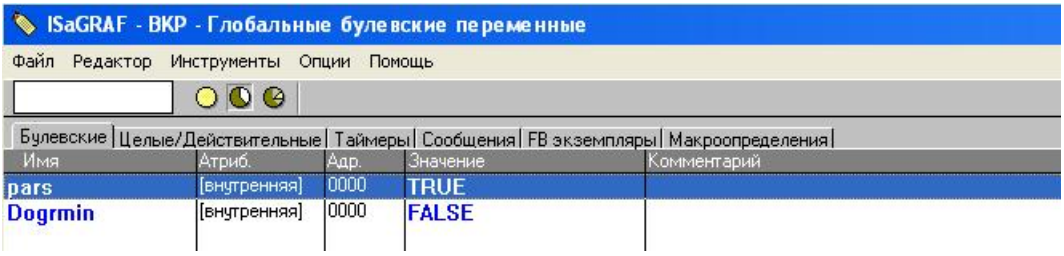


Рисунок 1.55 - Словарь

1.6 Составление программы ПИД-регурятора на языке ST

1.Объявление переменных

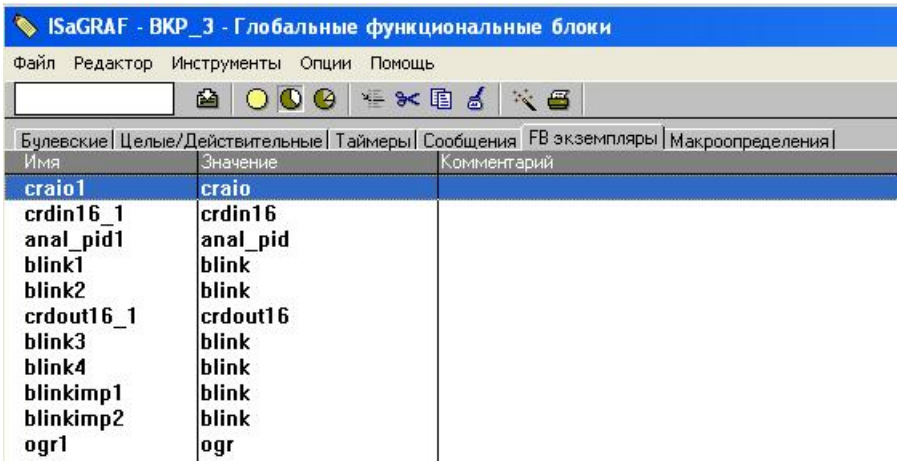


Рисунок 1.56 - Глобальные FB экземпляры

Текст программы для ввода (список) показан ниже.

```
craio1(address1,phys,Yout,Zero,Zero,Zero);
Xin:= craio1.AI1;

crdin16_1(address2);

anal_pid1(Zadanie,Csb,Cdb,Vdb,Xin,Xdlt,Kp,Ti,Kd,Xmax,Xmin,Czb,Czm,C
ruch,Xruch);
Yout:= anal_pid1.Y;
Yeps:= anal_pid1.Yeps;

IF (Xin>=Xsigmax)THEN
```

AlarmMax:= TRUE ;

ELSE AlarmMax:= FALSE ;

END_IF;

IF (Xin<=Xsigmin)THEN

AlarmMin:= TRUE ;

ELSE AlarmMin:= FALSE ;

END_IF;

blink1(AlarmMax,blink_period);

AlarmMaxBlink:= blink1.Q;

blink2(AlarmMin,blink_period);

AlarmMinBlink:= blink2.Q;

crdout16_1(address2,AlarmMaxBlink,AlarmMinBlink,false,false,false,false,false,false,false,false,false,false,false,false,false,false);

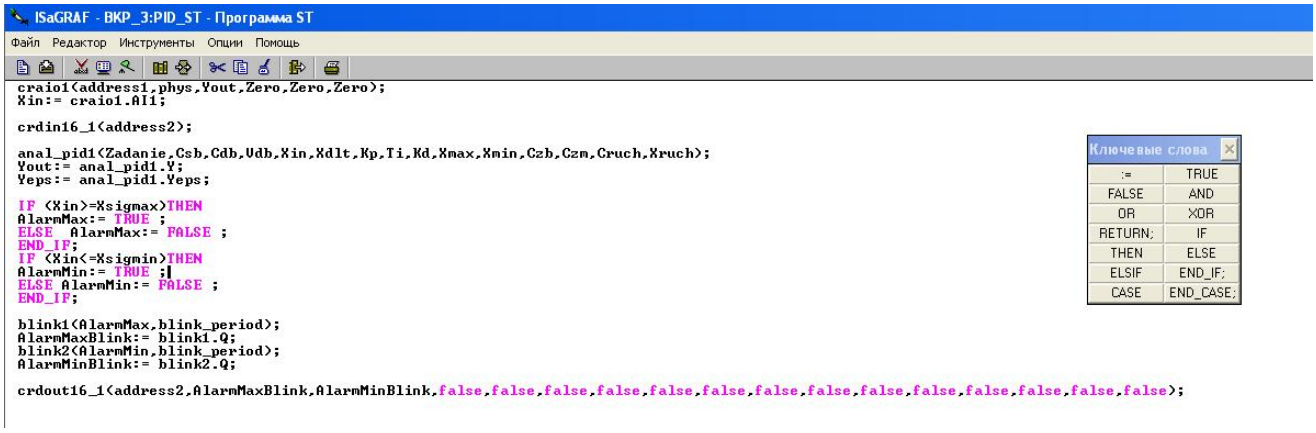


Рисунок 1.57 – Программа, реализующая ПИД-регулятора на языке ST.

Понаблюдать за значениями переменных и изменять значение переменной Zadanie (Уставка) можно, если во время работы программы открыть Словарь (рисунок 1.58). Для изменения значения Задания (Уставки) выполните двойной щелчок ЛК мыши на имени переменной Zadanie и в появившемся окне задайте новое значение переменной.

ISaGRAF - BKP_3 - Глобальные целые/вещественные переменные

Файл Редактор Инструменты Опции Помощь

Булевские | Целые/Действительные | Таймеры | Сообщения | FB экземпляры | Макроопределения

Имя	Атриб.	Адр.	Значение	Комментарий
address1	[константа,цел	0000	230	Адрес craio
address2	[константа,цел	0000	51	Адрес crdout16,crdin16
Xin	[внутренняя,ве	0000	0	Входной сигнал
Yout	[внутренняя,ве	0000	0	Основной выход регулятора
Zadanie	[внутренняя,ве	0000	2	Уставка регулятора
Vdb	[константа,вещ	0000	0	Скорость динамической балансировки
Xdlt	[константа,вещ	0000	0.001	Размер зоны нечувствительности
Kp	[внутренняя,ве	0000	1.1	Коэффициент пропорциональности
Ti	[внутренняя,ве	0000	1	Постоянная времени интегрирования
Kd	[внутренняя,ве	0000	0.01	Постоянная времени дифференцирования
Xmax	[константа,вещ	0000	20	Уровень ограничения по максимуму
Xmin	[константа,вещ	0000	0	Уровень ограничения по минимуму
Xruch	[внутренняя,ве	0000	0	Ручное задание
Yeps	[внутренняя,ве	0000	0	Выход рассогласования
Yzdn	[внутренняя,ве	0000	0	Выход текущего задания
Xsigmax	[внутренняя,ве	0000	2.5	Максимум для сигнализации
Xsigmin	[внутренняя,ве	0000	1	Минимум для сигнализации
Zero	[внутренняя,ве	0000	0	Переменная с нулевым значением
Kpom	[константа,вещ	0000	0	
Impul	[внутренняя,ве	0000	0	
Y	[внутренняя,ве	0000	0	
Kpom1	[внутренняя,ве	0000	0	
Dosomax	[внутренняя,ве	0000	0	
Dosimin	[внутренняя,ве	0000	0	
Pomexa	[внутренняя,ве	0000	0	
alfa	[внутренняя,ве	0000	0	
Ypered	[внутренняя,ве	0000	0	

Рисунок 1.58 – Окно словаря во время работы программы.

Примеры создания программ в системе ISaGRAF на языке LD

1.7 Создание программы логического ИЛИ на языке LD

Выполняя последовательно действия, приведенные в п.1.1 по созданию проекта и программы, создайте новый проект с именем «BKP_1», а в нем новую программу с именем «BKP_LD», но при создании программы в диалоговом окне выберите язык не FBD, а LD.

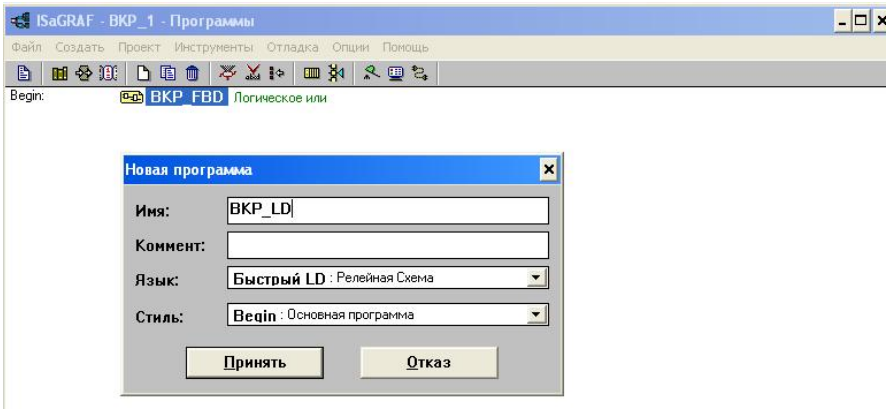


Рисунок 1.59 – Диалоговое окно создания новой программы

Программа создана, хотя она пока пустая, и ее имя появилось в окне Менеджера Программ. Далее необходимо зайти в программу, дважды нажав ЛК

мыши на имя программы BKP_LD. Появится окно редактора Quick LD «Быстрая программа LD», приведенное на рисунке 1.60, в котором нужно будет вводить и редактировать программу.

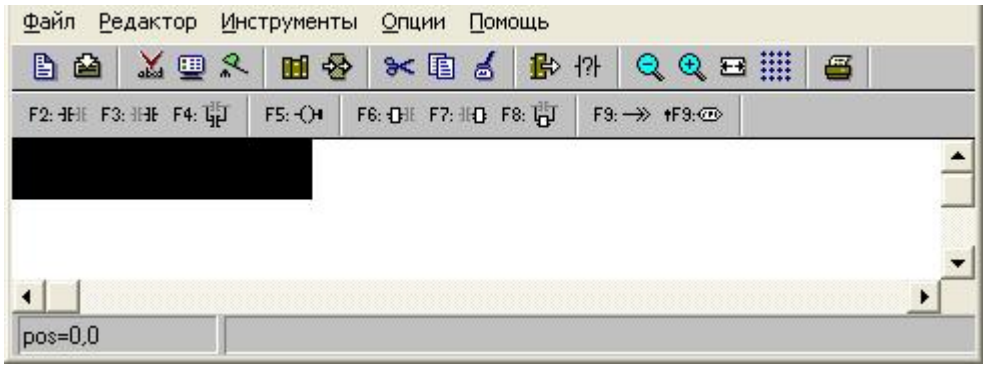


Рисунок 1.60 – Окно редактора Quick LD

Перед вводом программы нужно объявить все переменные. Для этого необходимо открыть словарь при помощи команды меню «Файл/Словарь» или кнопки Словарь. Появившееся окно словаря приведено на рисунке 1.61.

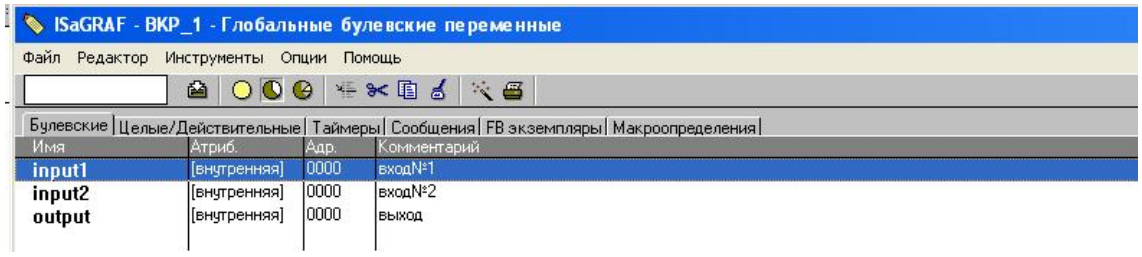


Рисунок 1.61 – Окно словаря

В окне словаря выбрать вкладку Булевские и произвести двойной щелчок в выделенной верхней строке рабочего поля, или можно воспользоваться кнопкой Вставить объект. В появившемся окне (рисунок 1.62) необходимо задать параметры переменной: имя input1, комментарий (комментарий необязателен), атрибуты: Внутренняя, начальное значение False (по умолчанию) и нажать кнопку «Сохранить».

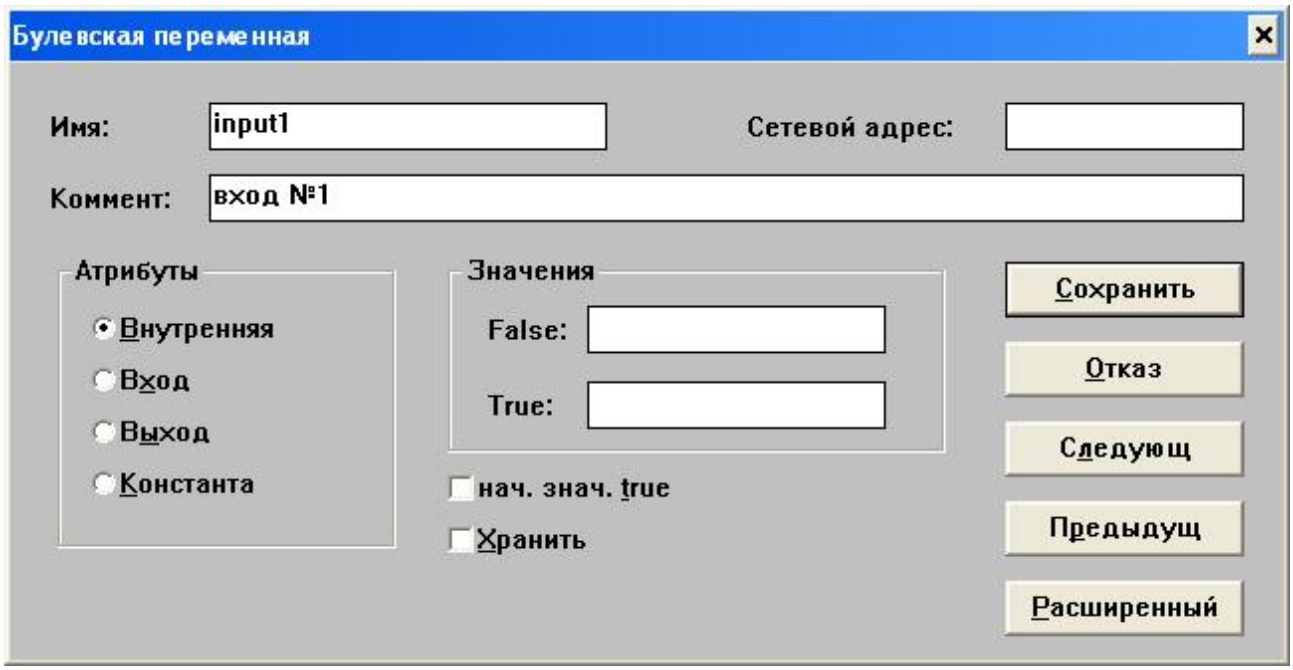


Рисунок 1.62 – Окно задания параметров входной переменной input1

Параметры второй входной переменной input2 и выходной output1 необходимо задать аналогичным образом (рисунки 1.63, 1.64). Далее сохранить все изменения командой меню словаря Файл/Сохранить и закрыть словарь.

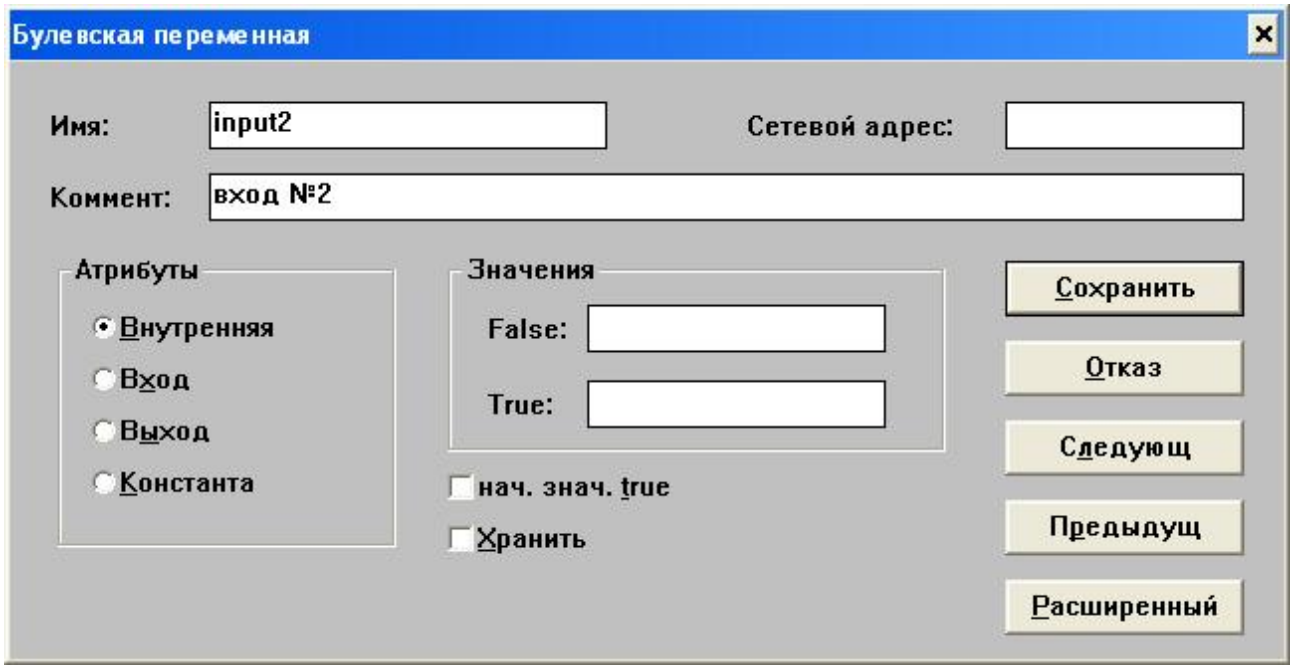


Рисунок 1.63 - Окно задания параметров входной переменной input2

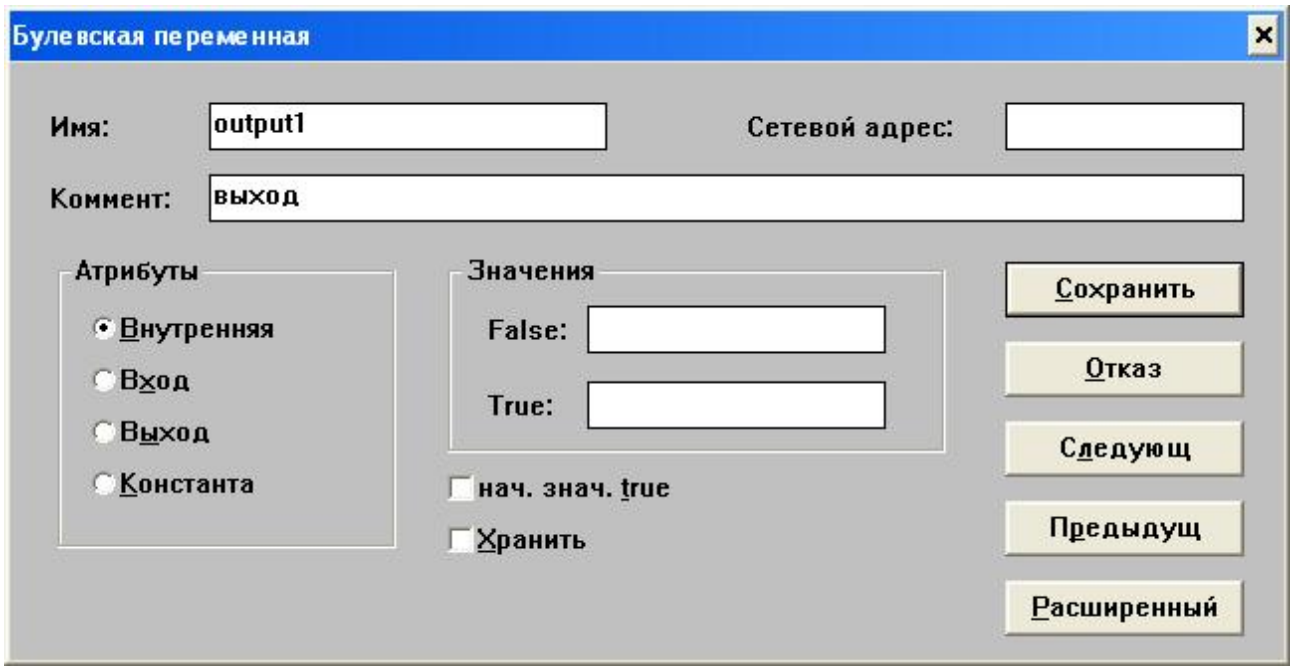


Рисунок 1.64 - Окно задания параметров выходной переменной output1

Теперь необходимо заняться вводом и редактированием программы. Вернёмся в окно редактора Quick LD «Быстрая программа LD», приведенного на рисунке 1.65.

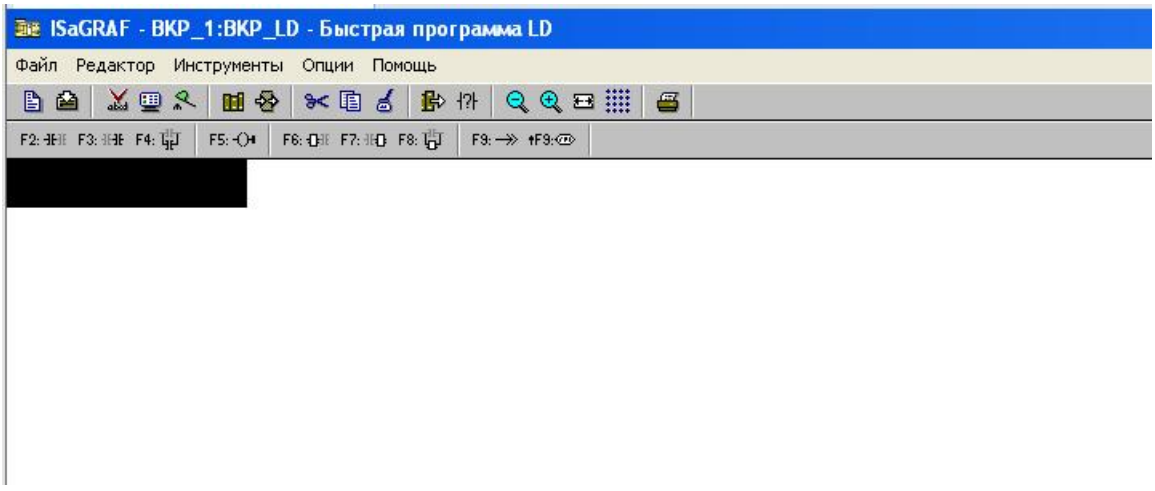


Рисунок 1.65 – Редактор Quick(Быстрый) LD

Необходимо обратить внимание, что на панелях редактора имеются подсказки по использованию функциональных клавиш клавиатуры ПК. Нажать на клавиатуре ПК клавишу F2, что означает ввод команды Контакт налево. На экране появится один прямой контакт, один прямой виток, а также левая и

правая силовые шины (рисунок 1.66).

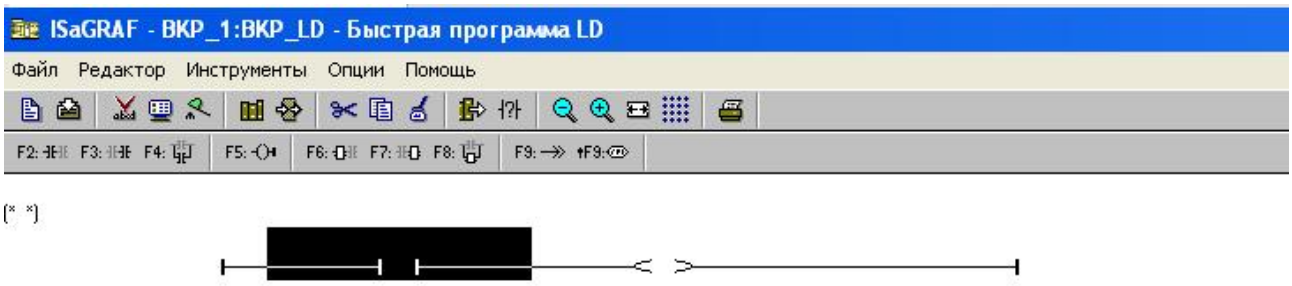


Рисунок 1.66 – Окно редактора после нажатия клавиши F2

Теперь нужно нажать на клавиатуре клавишу F3, что означает ввод команды Контакт направо. На экране появится еще один прямой контакт (рисунок 1.67).Кроме того, при желании можно еще и ввести комментарий Логическое ИЛИ, дважды нажав ЛК мыши на надпись (* *).

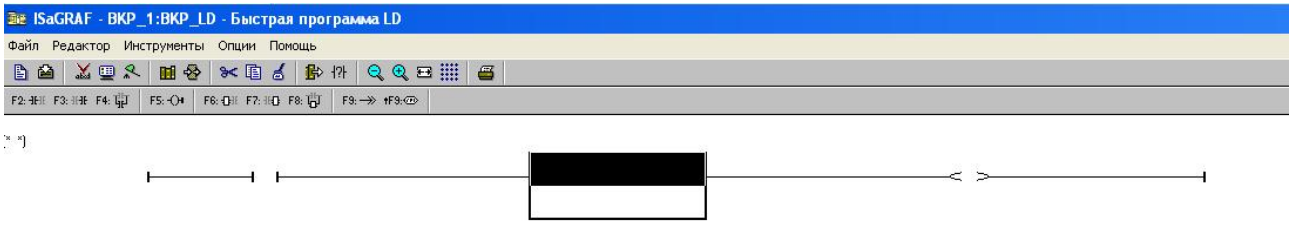


Рисунок 1.67 – Окно редактора после нажатия клавиши F3

Далее необходимо связать переменные с символами LD. Курсор можно перемещать по схеме, используя стрелки клавиатуры. Переместить курсор на самый левый контакт и нажать на клавиатуре клавишу Enter. В появившемся окне выбрать переменную с именем input1 (рисунок 1.68) и нажать кнопку «Принять».

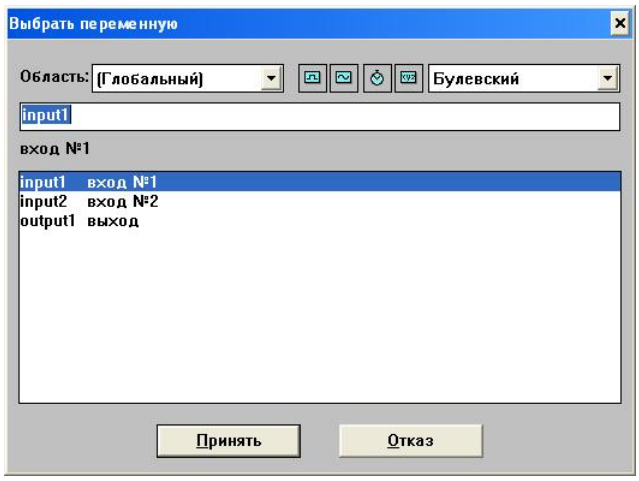


Рисунок 1.68 – Окно выбора переменной

Аналогичным образом второй контакт необходимо связать с переменной input2, а виток – с переменной output1. Теперь программа полностью завершена, полученный результат приведен на рисунке 1.69. Далее необходимо сохранить программу командой меню «Файл/Сохранить» и нажать кнопку Принять.

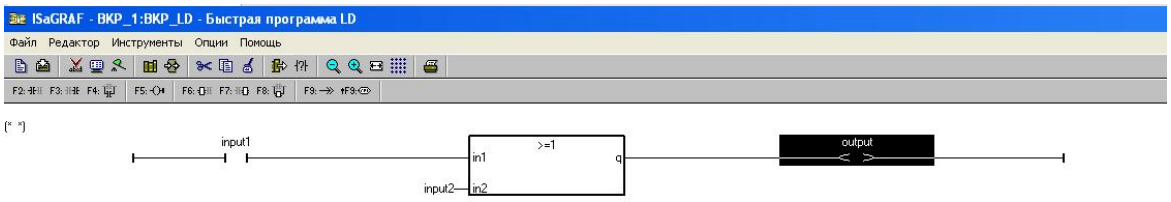


Рисунок 1.69 – Полностью введенная программа

После того как программа создана и переменные объявлены, необходимо проверить ее на правильность синтаксиса написания, для чего выбрать команду меню «Файл/Проверить». Если ошибок нет, выйти из генератора кода. Далее сохранить программу командой Файл/Сохранить из меню редактора, в появившемся диалоговом окне нажать кнопку Принять.

Теперь необходимо закрыть редактор Быстрая программа LD, так как в ближайших шагах необходимо запустить симулятор. А симулятор нельзя запускать, если открыты дочерние окна, при не выполнении этого условия может случится «зависание» системы ISaGRAF. Далее необходимо сгенерировать код командой меню «Создать/Создать» приложение или нажать кнопку Создать код приложения в окне Диспетчера Программ. Теперь для проверки работоспособности необходимо запустить программу в режиме

симуляции командой меню «Отладка/Симуляция» или кнопкой Симуляция.

Во время работы симулятора можно открыть программу, дважды нажав ЛК мыши на имени программы ВКР_LD в окне Отладка программ, после этого откроется окно редактора, где можно будет посмотреть, как она работает (рисунок 1.70).

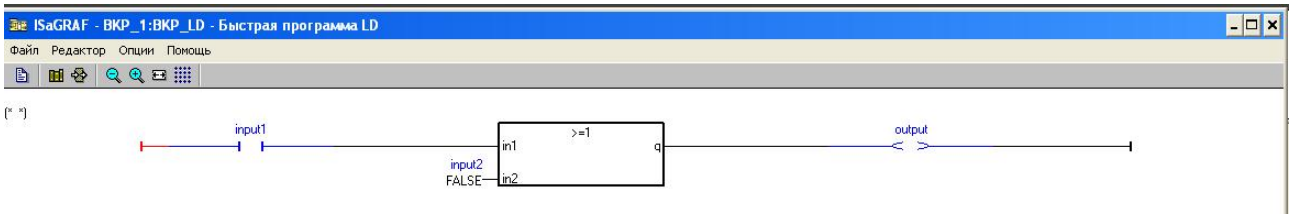


Рисунок 1.70 – Окно программы в режиме симуляции

Если какая-либо булевая переменная отображается в режиме симуляции в окне Редактирования синим цветом, это означает что ее значение равно False, а если красным, то True. Щелкнув два раза ЛК мыши на входной переменной input1, можно изменить ее значение с False на True или наоборот (рисунок 1.72).

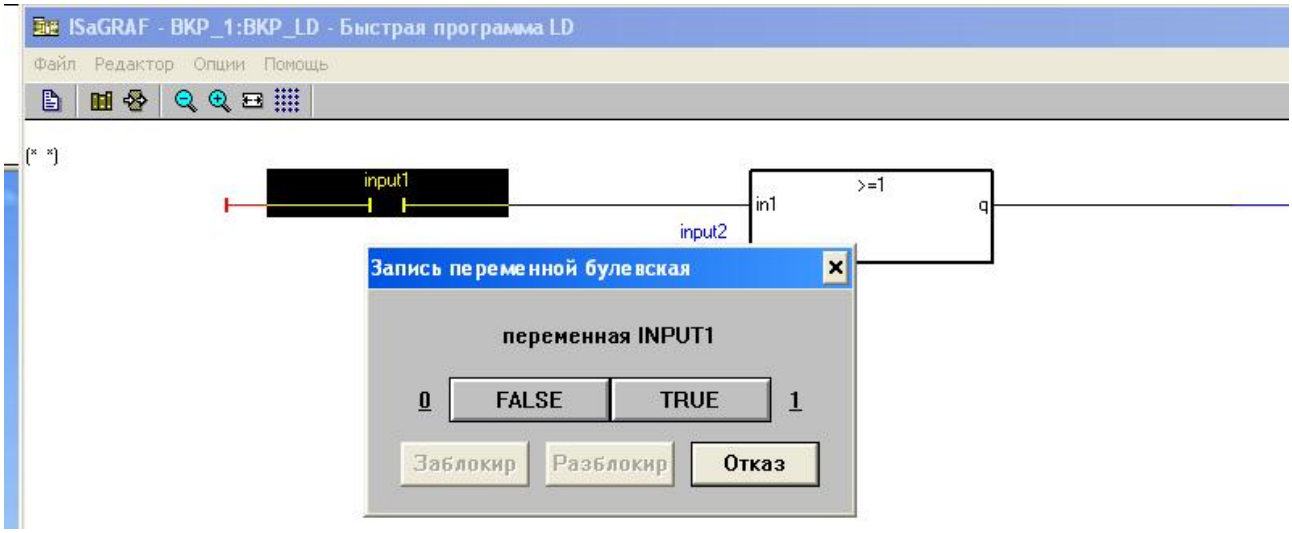


Рисунок 1.72 – Окно изменения значения переменной

Аналогичным образом можно изменить и значение переменной input2. Поэкспериментировав в разными значениями переменных input1 и input2, можно убедиться, что программа действительно выполняет операцию логического ИЛИ – то есть значение выходной переменной output1 становится равным True (при этом она отображается красным цветом) тогда и только тогда, когда значения обеих входных переменных равны True.

Кроме того, наблюдать за значениями переменных и изменять значения входных переменных в режиме симуляции можно используя Словарь Команда меню «Файл/Словарь» – рисунок 1.73. Для изменения значения переменной нужно выполнить двойной щелчок ЛК мыши на имени нужной переменной в словаре.

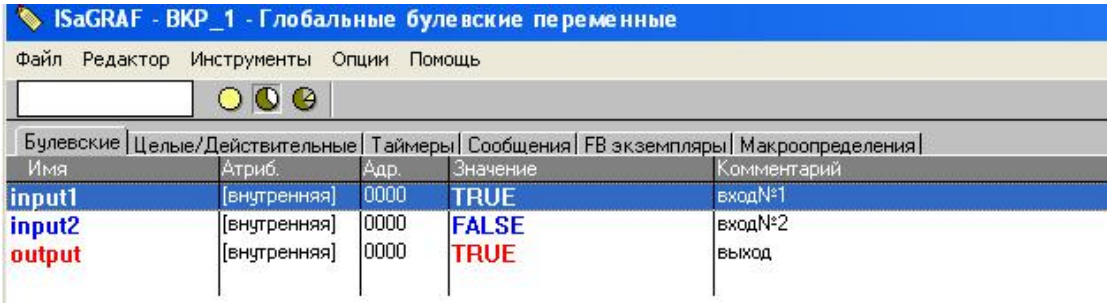


Рисунок 1.73 – Окно словаря в режиме симуляции

Для завершения работы программы в окне Отладчик необходимо выбрать команду «Файл/Выход».

1.8 Составление программы П-регурятора на языке LD

Создать программу П-регулятора на языке LD(рисунок 1.74)

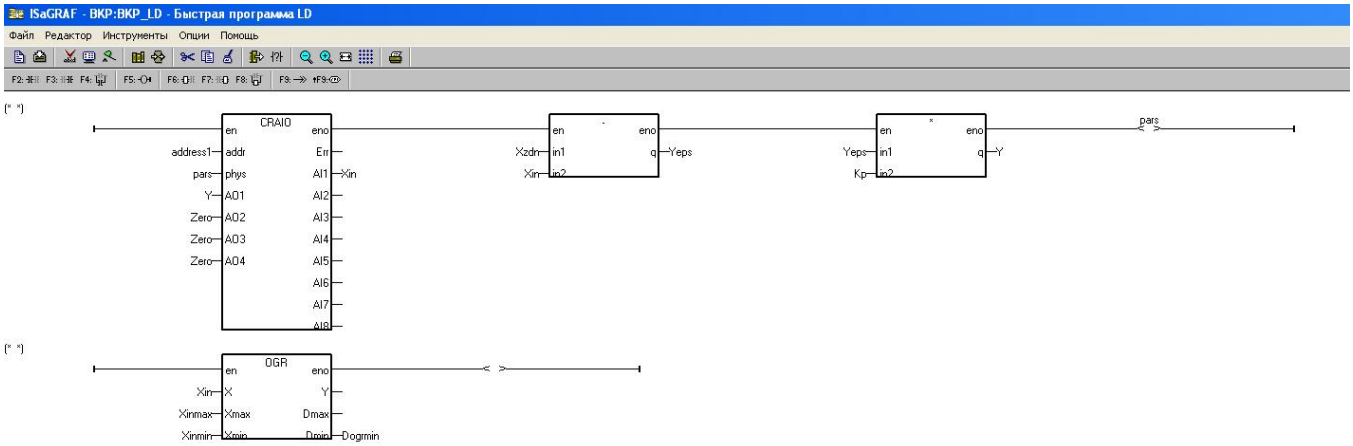


Рисунок 1.74 – П-регулятор

После того как программа создана и переменные объявлены, необходимо проверить ее на правильность синтаксиса написания, для чего выбрать команду меню «Файл/Проверить». Если ошибок нет, выйти из генератора кода. Далее сохранить программу командой «Файл/Сохранить» из

меню редактора, в появившемся диалоговом окне нажать кнопку Принять.

После того как были закончены все настройки, сгенерируем код приложения для загрузки в контроллер контекстное меню «Создать/ Создать приложение». Далее необходимо произвести загрузку программы в контроллер, для этого необходимо запустить отладчик меню «Отладка/Симуляция», при этом на экране появится окно отладчика (рисунок 1.75).

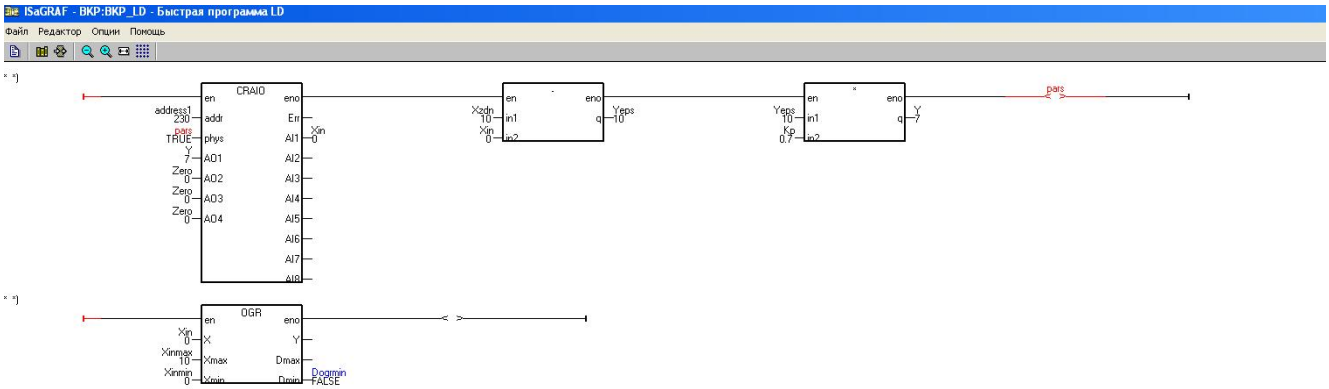
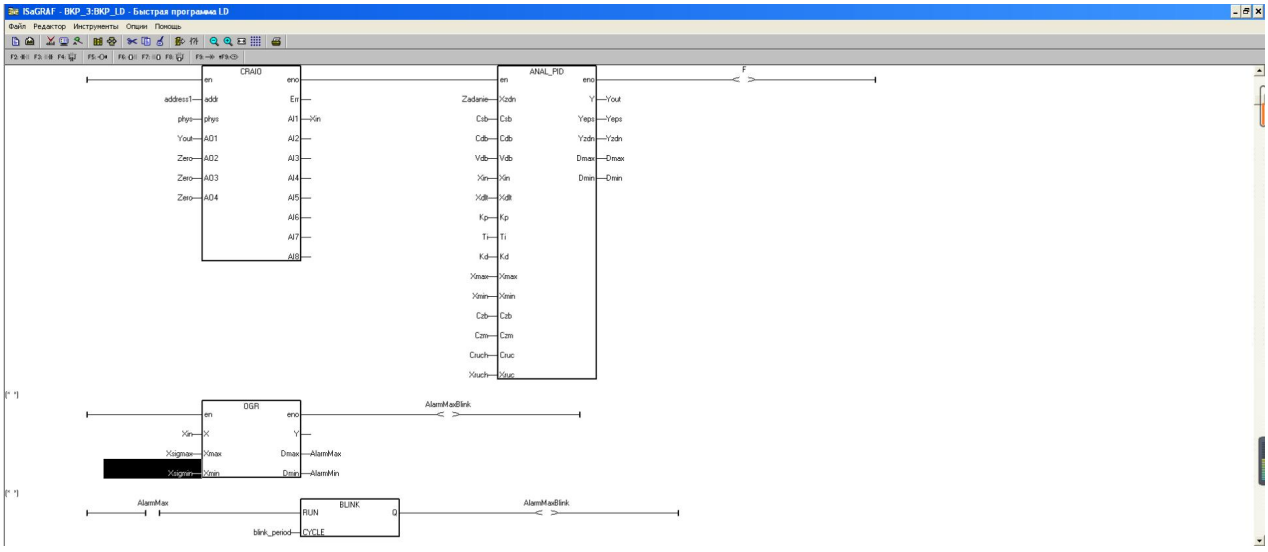


Рисунок 1.75 – Окно отладчика

1.9 Составление программы ПИД-регурятора на языке LD

Создать программу П-регулятора на языке LD(рисунок 1.76)



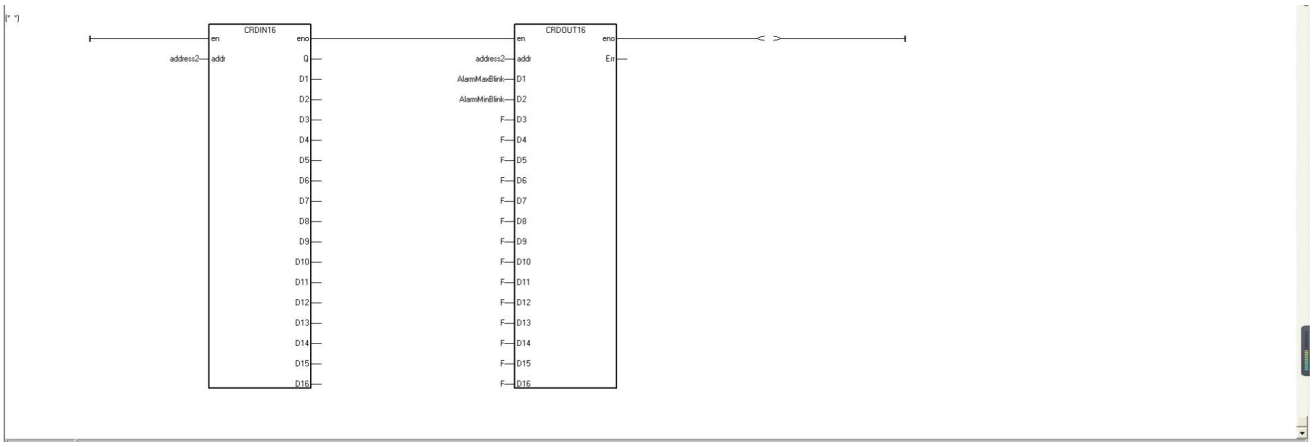
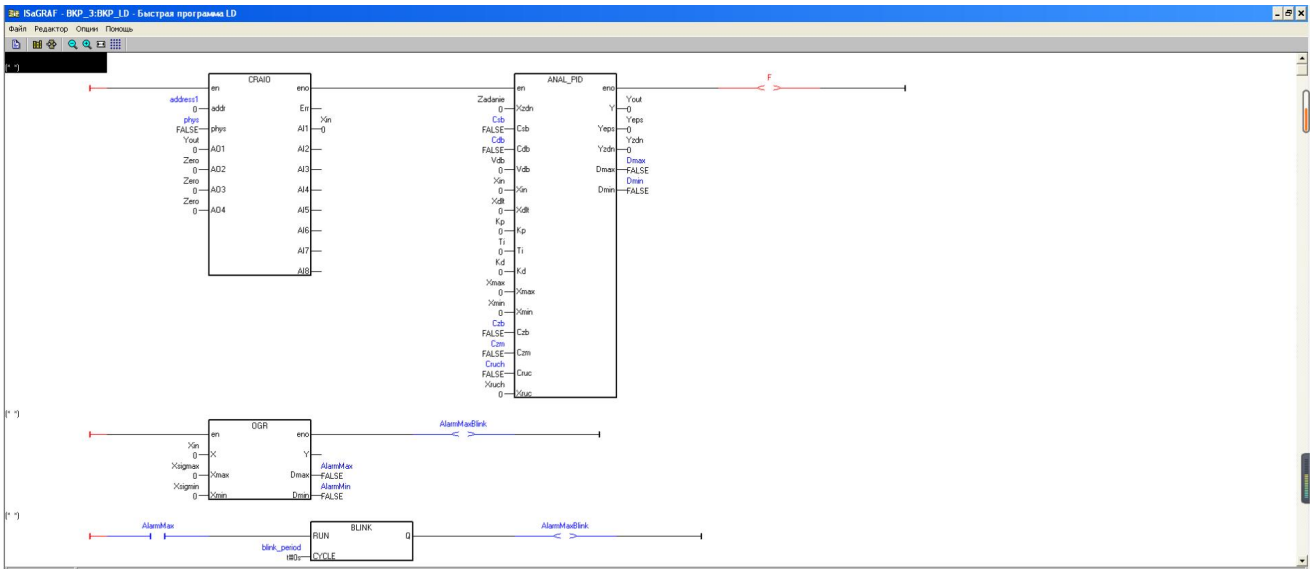


Рисунок 1.76 – ПИД-регулятор

После того как программа создана и переменные объявлены, необходимо проверить ее на правильность синтаксиса написания, для чего выбрать команду меню «Файл/Проверить». Если ошибок нет, выйти из генератора кода. Далее сохранить программу командой «Файл/Сохранить» из меню редактора, в появившемся диалоговом окне нажать кнопку «Принять».

После того как были закончены все настройки, сгенерируем код приложения для загрузки в контроллер контекстное меню «Создать/ Создать приложение». Далее необходимо произвести загрузку программы в контроллер, для этого необходимо запустить отладчик меню «Отладка/Симуляция», при этом на экране появится окно отладчика(рисунок 1.77).



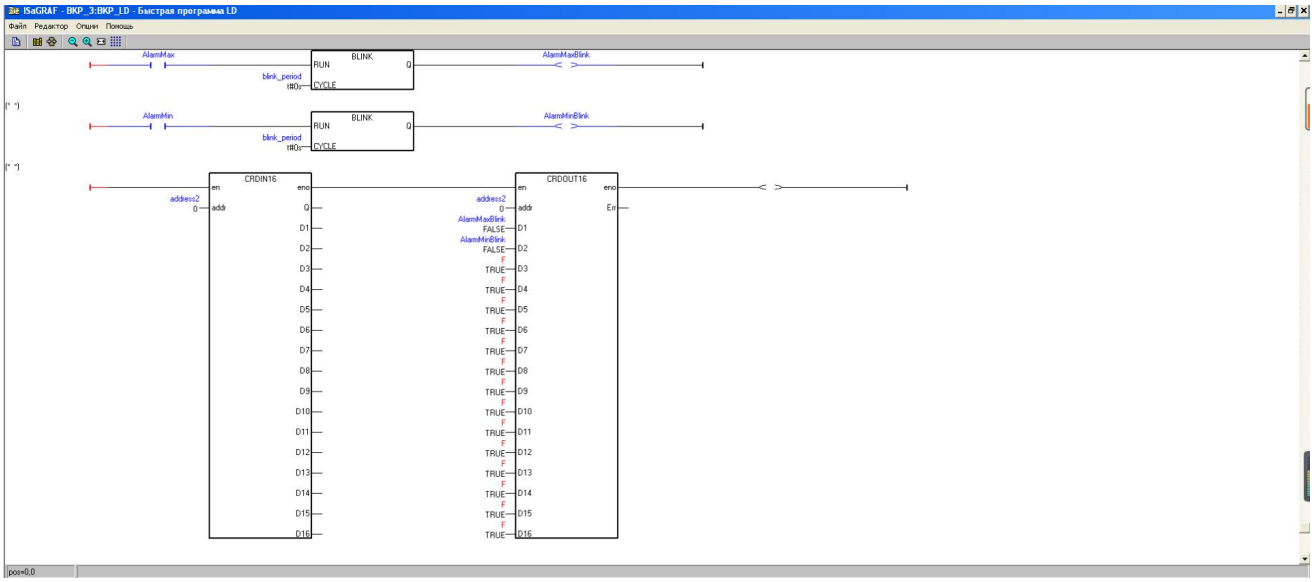


Рисунок 1.77 – Окно отладчика

Примеры создания программ в системе ISaGRAF на языке IL

1.10 Создание программы логического ИЛИ на языке IL

Выполняя последовательно действия, приведенные в п.1.1 по созданию проекта и программы, создайте новый проект с именем «BKP_1», а в нем новую программу с именем «BKP_IL», но при создании программы в диалоговом окне выберите язык не FBD, а IL.

.

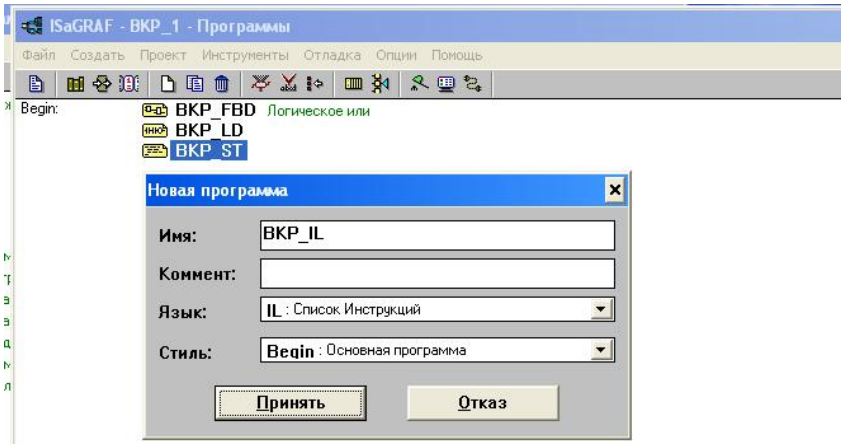


Рисунок 1.78 – Диалоговое окно создания новой программы

После объявления переменных введите программу в текстовый редактор IL (окно программы IL). Текст программы для ввода (список) показан ниже.

LD input1

OR input2

ST output

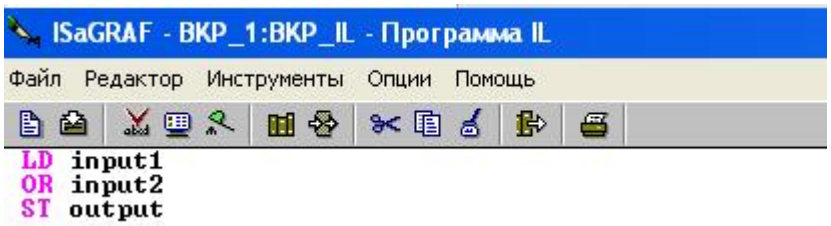


Рисунок 1.79 – Полностью введенная программа логического ИЛИ на языке IL

П о с л е создания программы и объявления переменных необходимо проверить их синтаксическую правильность, выбрав команду меню «Файл/Проверять». Если ошибок нет, выйдите из генератора кода.

С о х р а н и т е программу с помощью команды «Файл/Сохранить» и нажмите кнопку «Принять» в диалоговом окне.

П о с л е завершения всех настроек сгенерируем код приложения для загрузки в контроллер после контекстного меню «Создать/ Создать приложение» необходимо загрузить приложение в контроллер, для этого необходимо запустить отладчик меню «Отладка/Симуляция», после чего на экране появится окно отладчика(рисунок 1.80).

К о г д а приложение запущено, откройте словарь. Чтобы изменить переменную, дважды щелкните левой кнопкой мыши на имени переменной и задайте новое значение переменной в появившемся окне(рисунок 1.81).

Рисунок 1.80 – Окно программы в режиме симуляции

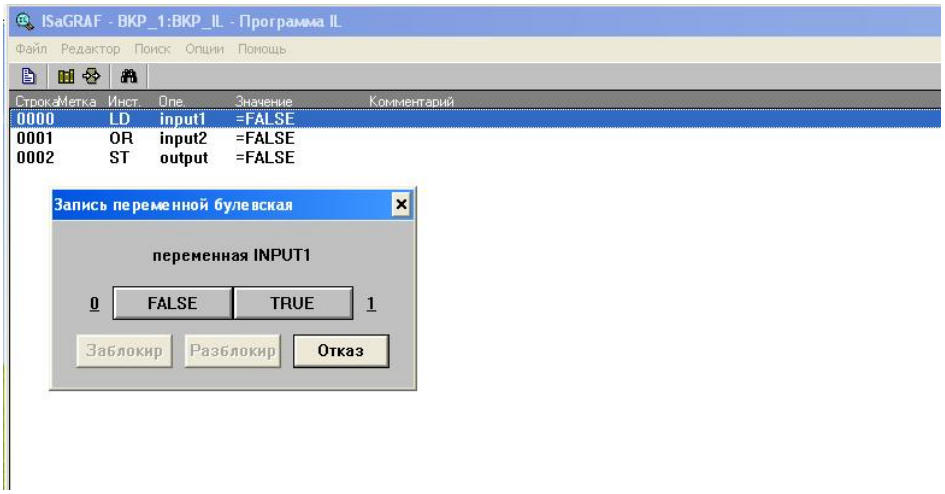


Рисунок 1.81 – Окно изменения значения переменной

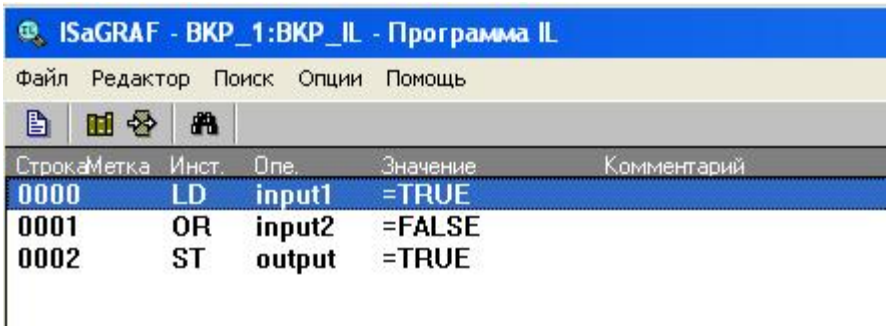


Рисунок 1.82 - Работа программы в режиме симуляции при значении input1=TRUE

1.11 Составление программы ПИД-регурятора на языке IL

Написать коды программы ПИД-регулятора на языке IL(рисунок 1.83). Текст программы для ввода (список) показан ниже.

```
LD address1
ST craio1.addr
```

```
LD phys
LD Yout
ST craio1.AO1
CAL craio1

LD Zadanie
ST anal_pid1.Xzdn
LD Csb
LD Cdb
LD Vdb
LD craio1.AI1
ST anal_pid1.Xin
LD Xdlt
ST anal_pid1.Xdlt
LD Kp
ST anal_pid1.Kp
LD Ti
ST anal_pid1.Ti
LD Kd
ST anal_pid1.Kd
LD Xmax
ST anal_pid1.Xmax
LD Xmin
ST anal_pid1.Xmin
LD Czb
LD Czm
LD Cruch
LD Xruch
ST anal_pid1.Xruc
CAL anal_pid1

LD craio1.AI1
ST ogr1.X
LD Xmax
ST ogr1.Xmax
LD Xmin
ST ogr1.Xmin
CAL ogr1

LD ogr1.Dmax
ST blink1.RUN
LD blink_period
ST blink1.CYCLE
CAL blink1
```

```
LD ogr1.Dmin
ST blink2.RUN
LD blink_period
ST blink2.CYCLE
CAL blink2

LD address2
ST crdin16_1.addr
CAL crdin16_1

LD address2
ST crdout16_1.addr
LD blink1.Q
ST crdout16_1.D1
LD blink2.Q
ST crdout16_1.D2
CAL crdout16_1
```

ISaGRAF - BKP_3:IL - Программа IL

Файл Редактор Инструменты Опции Помощь

```
LD address1
ST craio1.addr
LD phys
LD Yout
ST craio1.AO1
CAL craio1

LD Zadanie
ST anal_pid1.Xzdn
LD Csb
LD Cdb
LD Vdb
LD craio1.AI1
ST anal_pid1.Xin
LD Xdlt
ST anal_pid1.Xdlt
LD Kp
ST anal_pid1.Kp
LD Ti
ST anal_pid1.Ti
LD Kd
ST anal_pid1.Kd
LD Xmax
ST anal_pid1.Xmax
LD Xmin
ST anal_pid1.Xmin
LD Czb
LD Czm
LD Cruch
LD Xruch
ST anal_pid1.Xruc
CAL anal_pid1

LD craio1.AI1
ST ogr1.X
LD Xmax
ST ogr1.Xmax
LD Xmin
ST ogr1.Xmin
CAL ogr1

LD ogr1.Dmax
ST blink1.RUN
LD blink_period
ST blink1.CYCLE
CAL blink1

LD ogr1.Dmin
ST blink2.RUN
LD blink_period
ST blink2.CYCLE
CAL blink2

LD address2
ST crdin16_1.addr
CAL crdin16_1

LD address2
ST crdout16_1.addr
LD blink1.Q
ST crdout16_1.D1
LD blink2.Q
ST crdout16_1.D2
CAL crdout16_1
```

Рисунок 1.83 – Программ ПИД-регулятор

После того как были закончены все настройки, сгенерируем код приложения для загрузки в контроллер контекстное меню «Создать/ Создать приложение»(рисунок 1.84). Далее необходимо произвести загрузку программы в контроллер, для этого необходимо запустить отладчик меню «Отладка/Симуляция»(рисунок 1.85), при этом на экране появится окно отладчика(рисунок 1.86).

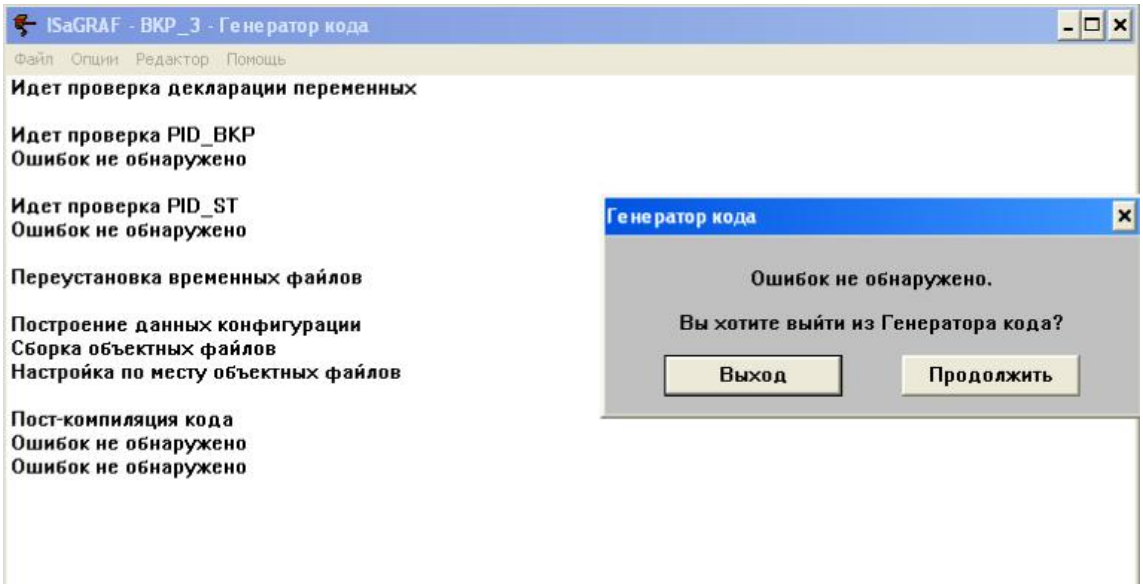


Рисунок 1.84 – Создать приложение

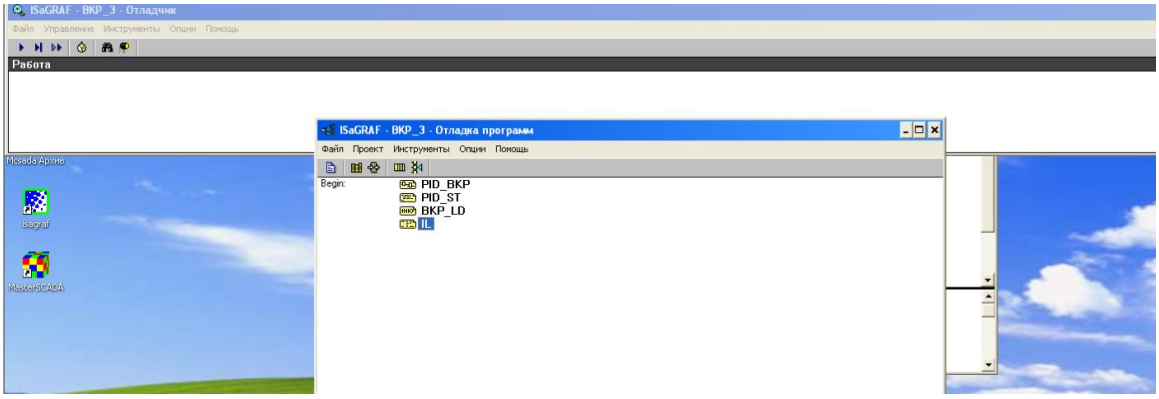


Рисунок 1.85 – Окно симуляцию

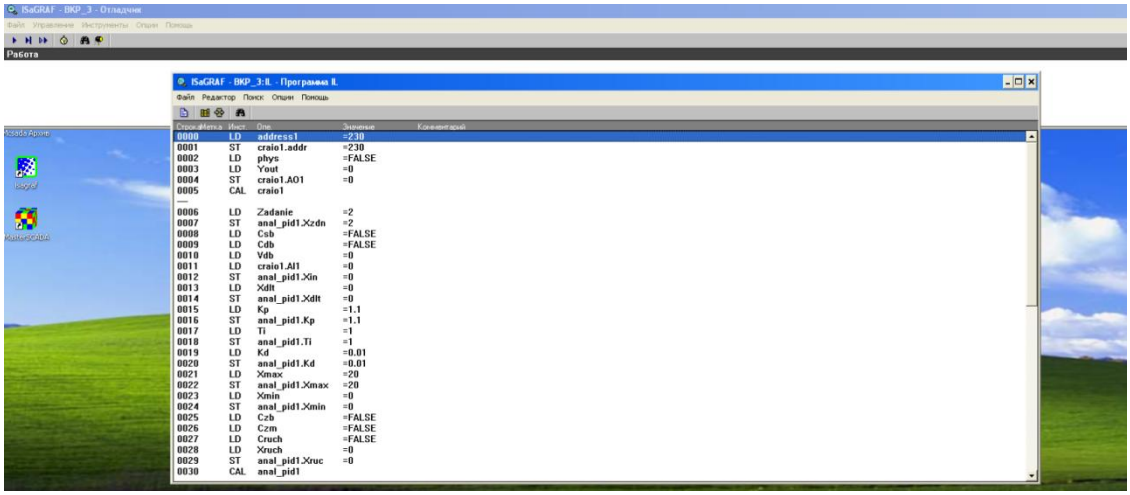


Рисунок 1.86 – окно отладчика

Учебное издание

АНИКЕЕНКО Владимир Михайлович
МАРЬИН Сергей Сергеевич
АНИСИМОВА Ольга Александровна

Программно-методическое обеспечение для изучения программного пакета ISaGRAF

Методические указания к выполнению лабораторных работ для студентов IV курса, обучающихся по направлению 150304 «Автоматизация технологических процессов и производств»

Отпечатано в Издательстве ТПУ в полном соответствии с качеством предоставленного оригинал-макета

Подписано к печати 00.00.2022. Формат 60x84/16. Бумага «Снегурочка».
Печать CANON. Усл. печ. л. 0,00. Уч.-изд. л. 0,00.
Заказ 000-22. Тираж 100 экз.

Издательство
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ