



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

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

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Тема работы
Программно - методическое обеспечение для изучения языка программирования FBD программного пакета ISaGRAF

УДК 004.438FBD

Обучающийся

Группа	ФИО	Подпись	Дата
158T92	Вэй Дуньюй		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Литвинов Р. В.	к.физ.-мат.н., доцент		

Консультант

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

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

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

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ООД ШБИП	Сечин А. И.	д.т.н., профессор		

Нормоконтроль (при наличии)

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

ДОПУСТИТЬ К ЗАЩИТЕ:

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

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ООП

Код компетенции	Наименование компетенции
Универсальные компетенции	
УК(У)-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
УК(У)-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений
УК(У)-3	Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде
УК(У)-4	Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(-ых) языке(-ах)
УК(У)-5	Способен воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах
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 безопасные условия жизнедеятельности, в том числе при возникновении чрезвычайных ситуаций
УК(У)-9	Способен проявлять предприимчивость в профессиональной деятельности, в т.ч. в рамках разработки коммерчески перспективного продукта на основе научно-технической идеи
Общепрофессиональные компетенции	
ОПК(У)-1	Способен использовать основные закономерности, действующие в процессе изготовления продукции требуемого качества, заданного количества при наименьших затратах общественного труда
ОПК(У)-2	Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и
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,

Код компетенции	Наименование компетенции
	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
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
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
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
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

Код компетенции	Наименование компетенции
	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
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
ПК(У)-21	способен составлять научные отчеты по выполненному заданию и участвовать во внедрении результатов исследований и разработок в области автоматизации технологических процессов и производств, автоматизированного управления жизненным циклом продукции и ее качеством
ПК(У)-22	Способен участвовать: в разработке программ учебных дисциплин и курсов на основе изучения научной, технической и научно-методической литературы, а также собственных результатов исследований; в постановке и модернизации отдельных лабораторных работ и практикумов по дисциплинам профилей направления; способностью проводить отдельные виды аудиторных учебных занятий (лабораторные и практические), применять новые образовательные технологии, включая системы компьютерного и дистанционного обучения



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

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

УТВЕРЖДАЮ:
Руководитель ООП
_____ Громаков Е.И.
(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

Обучающийся:

Группа	ФИО
158Т92	Вэй Дуньюй

Тема работы:

Программно - методическое обеспечение для изучения языка программирования FBD программного пакета ISaGRAF	
<i>Утверждена приказом директора (дата, номер)</i>	№ 33-43/с от 02.02.2023

Срок сдачи обучающимся выполненной работы:	10.06.2023 г.
--	---------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Объект исследования – язык FBD в системе ISaGRAF. Используемое оборудование - компьютер и контроллер КРОСС. Особых требований к среде эксплуатации нет, а вред окружающей среде ничтожен.
Перечень разделов пояснительной записки подлежащих исследованию, проектированию и разработке	1. Теоретический обзор предметной области; 2. Теоретические исследования функций устройств; 3. Реализация различных функций системы ISaGRAF; 4. Разработать учебные материалы для использования в обучении
Перечень графического материала	График, рисунок, таблицы применяемых в работе.

Консультанты по разделам выпускной квалификационной работы	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Былкова Татьяна Васильевна, доцент ОСГН ШБИП, к.э.н.
Социальная ответственность	Сечин Александр Иванович, профессор ООД ШБИП, д.т.н.

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	02.02.2023
---	------------

Задание выдал руководитель / консультант (при наличии):

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Литвинов Рудольф Викторович.	к.физ.-мат.н., доцент		02.02.2023

Задание принял к исполнению обучающийся:

Группа	ФИО	Подпись	Дата
158Т92	Вэй Дуньюй		02.02.2023

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 (ООП) – 15.03.04 Автоматизация технологических процессов и производств (Автоматизация технологических процессов и производств в нефтегазовой отрасли)

Уровень образования – Бакалавриат

Отделение школы (НОЦ) – Отделение автоматизации и робототехники

Период выполнения – Весенний семестр 2022 /2023 учебного года

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Обучающийся:

Группа	ФИО
158Т92	Вэй Дуньюй

Тема работы:

Программно-методическое обеспечение для изучения языка программирования FBD программного пакета ISaGRAF.
--

Срок сдачи обучающимся выполненной работы:	10.06.2023 г.
--	---------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
29.05.2023 г.	<i>Основная часть ВКР</i>	60
30.05.2023 г.	<i>Раздел «Социальная ответственность»</i>	20
30.05.2023 г.	<i>Раздел «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»</i>	20

СОСТАВИЛ:

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Литвинов Р.В.	к.физ-мат.н.		02.02.2023

СОГЛАСОВАНО:

Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОАР ИШИТР	Громаков Е.И	к.т.н., доцент		02.02.2023

Обучающийся

Группа	ФИО	Подпись	Дата
158Т92	Вэй Дуньюй		02.02.2023

Реферат

Выпускная квалификационная работа содержит: 101 с., 72 рис., 15 табл., 8 источников, 1 прил..

Ключевые слова: контроллер , методическое указание, программный продукт ISaGRAF, функциональный блок. Объектом исследования является программно-методическое обеспечение для изучения программного пакета ISaGRAF.

Цель работы: разработка программно-методического обеспечения 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 обеспечения для изучения программного пакета ISaGRAF на базе контроллеров "КРОСС".

В результате исследования был написан методический указатель для изучения программного пакета ISaGRAF.

Изучены программные языки FBD в системе ISaGRAF; Рассмотрен и изучен контроллер «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: письменное методическое 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.....	10
1 Программный пакет ISaGRAF.....	11
1.1 Язык функциональных блочных диаграмм (Язык FBD)	16
2 Лабораторный стенд для изучения языков программирования пакета ISaGRAF.....	18
2.1 Структурная схема лабораторного стенда.....	18
2.1.1 Краткое описание аналогового вычислительного комплекса.....	18
2.1.2 Описание контроллера «КРОСС».....	19
2.1.3 Схема внешних соединений.....	21
2.2 Состав и технические характеристики контроллера «КРОСС».....	22
2.2.1 Программное обеспечение контроллера.....	22
2.2.2 Основные сведения об операционной системе OS-9.....	23
3 Методическое обеспечение.....	25
4 Социальная ответственность.....	29
4.1 Правовые и организационные вопросы обеспечения безопасности.....	29
4.2 Производственная безопасность	29
4.3 Анализ опасных и вредных производственных факторов.....	30
4.4 Экологическая безопасность.....	31
4.5 Безопасность в чрезвычайных ситуациях.....	32
5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение....	38
5.1 Планирование научно-исследовательских работ.....	38
5.2 Бюджет научно-технического исследования.....	41
5.3 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования.....	44
Заключение	47
Список используемой литературы:	48
Приложение А (обязательное) Методические указания к выполнению лабораторной работы	49

Введение

В современном мире без автоматизации это равносильно отсутствию мотыг в эпоху земледелия. С помощью технологий автоматизации производительность труда человека значительно повысилась по сравнению с прошлым, а технологии автоматизации все еще развиваются, преследуя более высокие цели. На современном этапе технологии автоматизации не только дают заводам очень высокую эффективность, но и часто выполняют работу, которую не могут выполнять рабочие. Можно сказать, что сила предприятия, ориентированного на производство, тесно связана с задействованными передовыми технологиями автоматизации.

Для внедрения системы автоматического управления технологическими процессами необходимо широко использовать новейшие компьютерные и механические технологии производства, а также развивать высококлассные таланты.

Использование испытательных стендов на базе современных промышленных контроллеров и программного комплекса ISaGRAF, а также программно-методическое обеспечение лабораторных работ в процессе обучения является важным условием подготовки квалифицированных специалистов в области автоматизации.

Цель исследования — изучить язык FBD в программном комплексе ISaGRAF и подготовить поддержку метода для лабораторных работ. Лабораторная работа с современными промышленными контроллерами и позволяет студентам автоматике ТПУ получить практические навыки языка FBD в стандартной системе ISaGRAF.

1 Программный пакет ISaGRAF

Программа ISaGRAF является одним из наиболее успешных программных продуктов, применяющих стандарт IEC 1131-3 и полностью учитывающих требования международного стандарта GUI (Graphical User Interface).

Это программное обеспечение может загрузить разработанную программу в контроллер для дальнейшего использования. ISaGRAF также предоставляет ядро интерпретатора, которое преобразует пользовательские алгоритмы в функции. При использовании этих функций пользователи могут сосредоточиться на обработке машинно-зависимых кодов, а не обрабатывать их один за другим, тем самым экономя время и деньги пользователей.

Возможности ISaGRAF System Toolkit включают написание собственных драйверов ввода-вывода, добавление функциональных блоков, определяемых рынком, и взаимодействие с системами более высокого уровня. Это отличный набор инструментов, которые могут реализовать другие более полезные функции в соответствии с вашими интересами.

Системный инструментарий ISaGRAF включает в себя платформу разработки приложений, файлы базы данных проекта, уровни связи, дополнительные каналы связи и виртуальные машины. Виртуальная машина работает на целевой платформе (конфигурация) и запускает приложение автоматизации, скомпилированное с платформы разработки приложений.

Стандарт МЭК 1131-3В содержит пять языков программирования, которые учтены в ISaGRAF, а именно [1,2]: FBD, SFC, LD, ST и IL.

На рисунке 1.1 [2] показано основное содержание программы ISaGRAF, работающей за один цикл.



Рисунок 1.1 – Основные операции цикла

Работа ISaGRAF подобна часам, которые крутятся снова и снова. Следовательно, основной единицей измерения времени по отношению к нему является период, как показано на рис. 1.2.

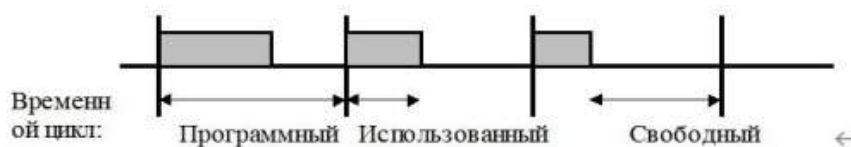


Рисунок 1.2 – Временной цикл

Применение временного цикла имеет много преимуществ, например возможность оценить производительность всего продукта путем тестирования его временной характеристики, что является важной функцией.

Система ISAGRAF не позволяет использовать несколько языков программирования в одной программе.

Система ISAGRAF предоставляет мощный браузер переменных, который поддерживает иерархические структуры и сетку для просмотра определений переменных. Для каждого ресурса в программе разработчики могут определять переменные, используя простые типы данных или определяемые ими настраиваемые типы, такие как массивы и структуры. Новые переменные могут быть определены во время разработки программы. Предопределенные переменные могут быть вставлены в программы для экономии времени

редактирования. ISAGRAF 5 поддерживает несколько новых типов данных, таких как 8-битные и 16-битные целые числа без знака и т. д.

Устройства для обработки входов и выходов (I/O) могут быть как простыми, так и сложными, поддерживающими различные типы и ориентации, и могут принимать ряд типов данных (например, логические, целочисленные и в реальном времени), которые могут управлять реальностью, состоянием или необходимой информацией. Для каждого канала можно задать усиление и смещение сигналов, выраженное в технических единицах, что увеличивает независимость между аппаратным обеспечением и приложением. Например, изменения на входе или выходе в определенном диапазоне аппаратных значений могут не обязательно требовать фундаментальных изменений в программном обеспечении, а просто переопределения коэффициентов усиления и смещения для канала. Эта гибкость позволяет нам настраивать работу канала и оптимизировать производительность устройства без внесения существенных изменений в программное обеспечение.

Среда разработки приложений

ADE автоматизирует задачи на протяжении всего жизненного цикла системы, включая проектирование, запуск, эксплуатацию и техническое обслуживание. Инструментарий ISaGRAF является основой среды разработки приложений, предоставляя разработчикам интуитивно понятную среду для создания собственных систем автоматизации и расширения основных компетенций.

В ISaGRAF проект можно разделить на одно или несколько колец или ресурсов ПЛК, определив аппаратные платформы и связи. Эти ресурсы являются виртуальными объектами во время выполнения, тогда как конфигурация и коммуникационные соединения представляют собой физическое развертывание проекта. Ресурс состоит из нескольких блоков управления программой (POUS), которые можно написать и запустить на языке программирования IEC 61131-3 с помощью блок-схем, функций и функциональных блоков. Он также поддерживает программирование функциональной блок-схемы IEC 61499.

Для разработки программ среда разработки приложений ISaGRAF предоставляет простые в использовании графические и текстовые редакторы, а также поддерживает такие функции, как перетаскивание, вырезание и вставка, что значительно повышает эффективность работы. Кроме того, все объявления переменных включены в словарь ISaGRAF, который также можно импортировать из внешних баз данных. При программировании вы можете вставить необходимые переменные в программу простым щелчком мыши. Продукт также позволяет создавать функциональные модули, а менеджер по разработке программ помогает разработчикам проектировать модули и интерфейс, упрощая повторное использование кода и экономя время и деньги во время разработки.

Используя среду разработки приложений, вы можете исследовать многие аспекты системы. Представление ресурсов показывает вам специальные возможности приложения. Представление аппаратного обеспечения демонстрирует и позволяет управлять работой ресурсов на диаграмме аппаратного обеспечения.

Подключение физического адреса ввода/вывода

Платформа разработки программ ISaGRAF предоставляет мощный редактор соединений ввода/вывода для подключения не зависящих логических переменных к физическим каналам ввода/вывода целевого промышленного компьютера или ПЛК в проекте. С помощью этого редактора разработчики могут легко назначать переменные и каналы, связанные друг с другом, обеспечивая эффективную и надежную передачу данных между программными и аппаратными компонентами системы. Просто щелкнув мышью, программисты могут определять соединения без глубоких знаний программного или аппаратного обеспечения. Логическое программирование полностью отделено от конфигурации линий ввода-вывода, что позволяет разработчикам приложений беспрепятственно выполнять перенос на различные аппаратные платформы.

Моделирование и отладка

Программа ISaGRAF позволяет выполнять отладку без необходимости для программистов возвращаться к основному логическому блоку или запоминать правила синтаксиса для каждого языка. Вместо этого платформа автоматически проверяет программное обеспечение и языковые правила, исправляет или предлагает правильное использование каждого языка и предоставляет подробную онлайн-справку. Эта функция экономит время и усилия разработчиков при отладке и повышает качество кода, обеспечивая соответствие отраслевым стандартам.

Точно так же моделирование использует мощный инструмент отладки, который быстро диагностирует производительность проекта программы и указывает области кода, требующие оптимизации. Кроме того, эмулятор также имеет инструменты просмотра для редактирования и запуска сценариев моделирования, что эффективно сокращает время отладки и проверки. При отсутствии определенной аппаратной платформы разработчики могут тестировать целые проекты приложений в лаборатории. Симулятор в платформе разработки приложений позволяет пользователю тестировать функциональность каждого модуля индивидуально или глобально, с возможностью контролировать статус выполнения программы и проверять статус любой переменной.

Текст для ознакомления

Генератор текста платформы разработки приложений ISaGRAF создает непрерывный печатный текст, описывающий элементы проекта и историю их изменений. Пользователи могут настраивать текстовое содержимое, включая включаемые компоненты проекта, в соответствии с заданными требованиями. Все изображения, текстовые программы и структуры проекта, таблицы соединений ввода/вывода, таблицы переменных, справочные стандарты включены в окончательный файл. Печатный текст может быть сгенерирован в желаемом формате.

1.1 Язык функциональных блочных диаграмм (Язык FBD)

FBD — это графический язык программирования для программирования ПЛК, который является частью стандарта IEC 61131-3. Язык состоит из ряда цепочек, каждая из которых имеет разные символы или формы, выполненные сверху вниз. Инструкция Перейти к метке позволяет изменить порядок выполнения схемы, который подходит для условного программирования и циклов. Библиотека и пользовательские блоки могут использоваться для программирования, и они могут быть написаны на FBD или других языках IEC 61131-3.

К блочным элементам относятся подпрограммы, функции И, ИЛИ, НЕ, триггеры, таймеры, счетчики, блоки обработки аналоговых сигналов и блоки математических операций. Каждая цепочка состоит из элементов, соединенных графическими выражениями. Следующий блок подключается к выходу текущего блока, создавая цепочку. Блоки в цепочке выполняются в определенном порядке. Результаты расчетов схемы могут быть сохранены во внутренних переменных или переданы на выход ПЛК.

Дополнительные сведения о FBD

Стандартные базовые функциональные блоки определяют программу FBD в ISaGRAF. Каждый блок предоставляет фиксированное количество входных и выходных портов, где каждый порт соответствует определенному типу. Переменные функционального блока подключаются к входу и выходу через соединительные линии. Эти линии являются односторонними и соединяют логические точки на диаграмме. Входная переменная относится к входному функциональному блоку, выходной функциональный блок к другому входному блоку и выходной функциональный блок к выходной переменной. Концы соединения должны быть одного типа.

Для вывода результатов оператор RETURN должен быть подключен к логическому выходу функционального блока, указывающего условие остановки выполнения программы и предотвращения выполнения остальной части диаграммы. Используйте переходы и теги для управления потоком в графике.

Невозможно соединить другие объекты с правым концом тега или символа перехода. Примеры именованя: >>LAB.... Помечает расположение именем "LAB". Когда левый конец линии соединения с символом перехода имеет значение TRUE, выполнение переходит к отмеченному маркеру.

Когда правый конец одиночной соединительной линии подключен к входу функционального блока, можно добавить логическое НЕ. Создайте небольшой цикл, который требует, чтобы оба конца имели логический тип.

2 Лабораторный стенд для изучения языков программирования пакета ISaGRAF

2.1 Структурная схема лабораторного стенда

В центральном блоке контроллера CROSS расположены модуль центрального процессора, модуль Ethernet LAN, модуль питания и модуль расширения интерфейса. В состав центрального блока CB1 контроллера CROSS входят submodule и модуль центрального процессора SM2-CPU-1.5, оба из которых имеют общую лицевую панель.

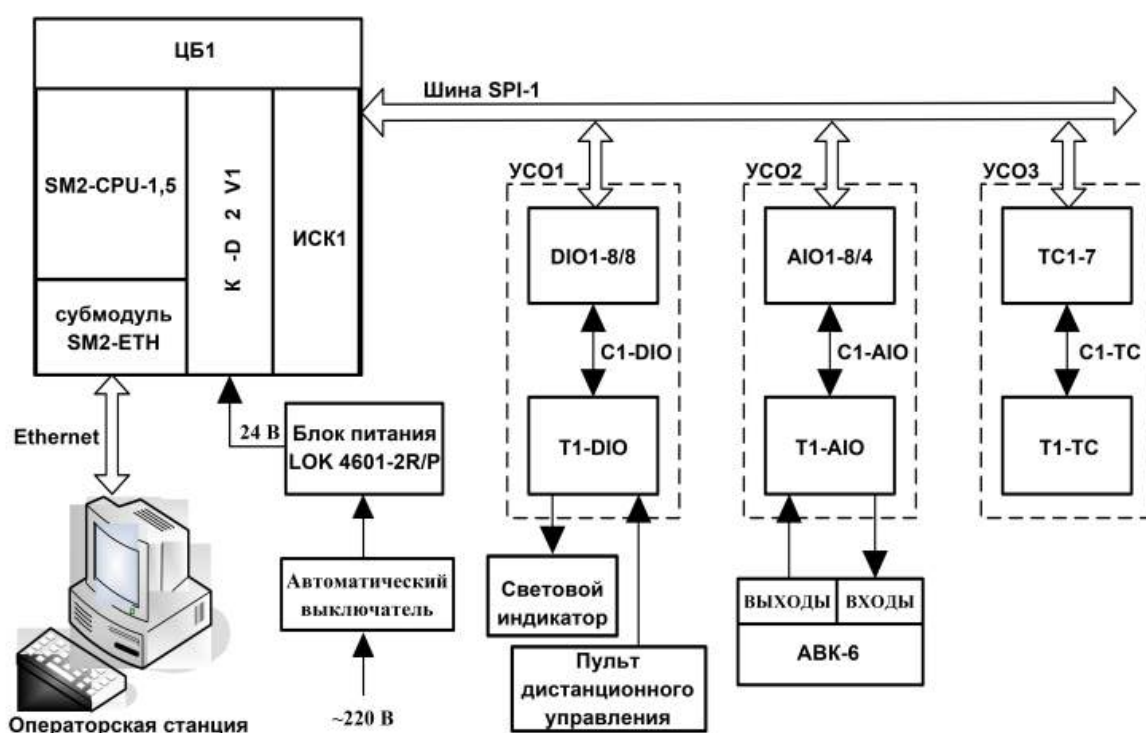


Рисунок 2.1 - Структурная схема лабораторного стенда

2.1.1 Краткое описание аналогового вычислительного комплекса

Состав АВК представлен на рисунке 2.2. В системе автоматического управления АСУ используется математическое моделирование динамического процесса, поэтому очень важно изучить систему. АВК широко используется в физико-математических дисциплинах, широко изучается инженерами и студентами.

МЭМ-6 и ВИМС составляют аналоговую вычислительно-индикаторную измерительную систему АВК-6.

С МЭМ-6 не нужно иметь специальных навыков в области стандартного компьютерного программирования, можно программировать по структурной математической модели динамического процесса.

Вообще говоря, АВК-6 состоит из множества самостоятельных модулей и узлов. Для удобства использования все компоненты могут быть размещены в одном корпусе. Использование одного агрегата позволяет быстро и легко настраивать и обслуживать весь комплект. Корпус АВК-6 представляет собой четырехсекционный каркас, причем части каждого модуля могут быть вставлены в соответствующие им отсеки, которые затем привинчены к корпусу.

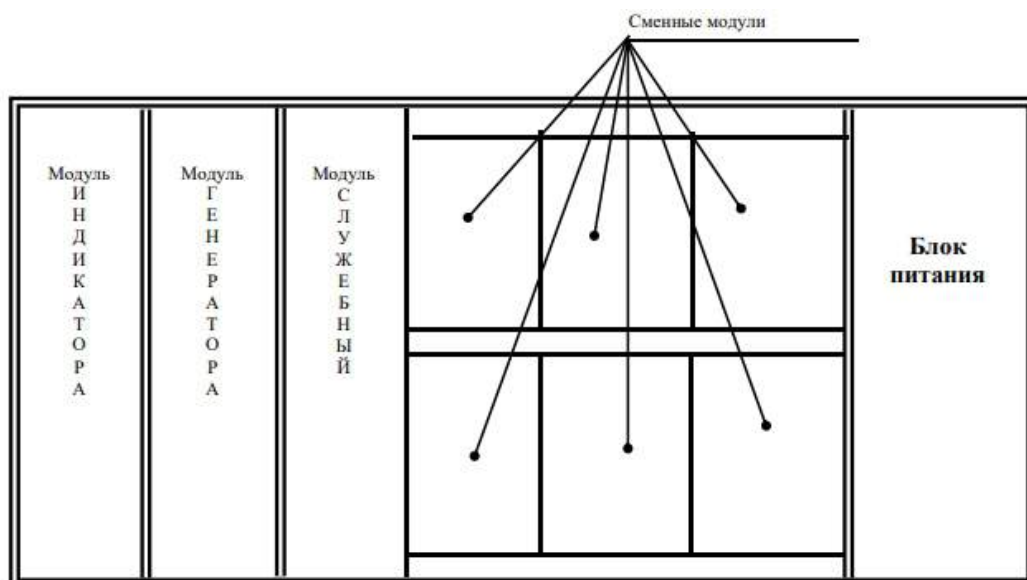


Рисунок 2.2 - Состав АВК-6

2.1.2 Описание контроллера «КРОСС»

Контроллер CROSS (рисунок 2.3) состоит из центрального блока, модуля ODR (DIO1-8/8, AIO1-8/4 и TC1-7), клеммной колодки (T1-DIO, T1-AIO и T1-TC).

Состав контроллера показан в таблице 2.1 [1].

Модуль ввода-вывода (модуль ODR) подключается к центральному блоку СВ1 посредством разъема SPI с помощью разъема ISK1, Модуль ISK1 имеет два интерфейсных порта SPI, в каждой группе до 8.

Параметры сигнала интерфейса приведены в таблице 2.3[1].

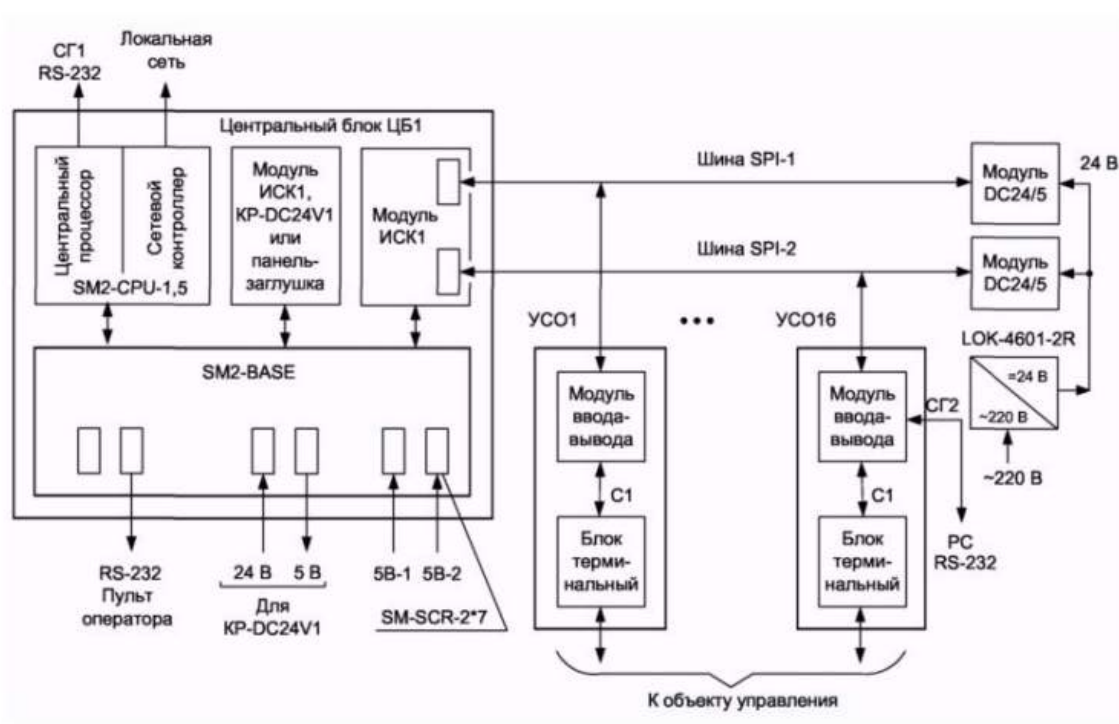


Рисунок 2.3- Архитектура контроллера

Контроллер имеет каналы измерения и может выполнять преобразование АЦП и ЦАП Конкретные значения см. в таблице 2.1.

Контроллер имеет возможность выбора источника питания. Питание может быть от однофазной сети переменного т о к а, номинальное напряжение 220 В, максимальное отклонение может достигать от минус 33 до 22 В, частота (50 ± 1) Гц , и он может выдерживать коэффициент гармоник до 5 %. Кроме того, контроллер может также принимать питание от внешнего нестабильного источника постоянного напряжения (24 ± 6) В.

Таблица 2.1 – Метрологические характеристики контроллера

Модуль	Сигналы		Пределы допускаемой основной приведенной (уо, %) или абсолютной (Аб, °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%	0,5у
	ТПП(К), (0-20,620) мВ	(0-1730,0) °С	± 3,5 °С *	
	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(Ј), (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(Н), (0-46,425) мВ	(0-1270,0) °С	± 2,5 °С *	
	ТВР(А1), (0-33,252) мВ	(0-2450,0) °С	± 4,9 °С *	
	ТВР(А2), (0-26,773) мВ	(0-1760,0) °С	± 3,5 °С *	
	ТВР(А3), (0-26,31 6) мВ	(0-1760,0) °С	± 3,5 °С *	
	ТХК(Л), (0-64,771) мВ	(0-780,0) °С	± 1,6 °С *	
	ТСМ50', (50-92,800) Ом	(0-200,0) °С	± 0,4 °С	
TR1-8	ТСП50, (50-99,937) Ом	(0-266,0) °С	± 0,5 °С	1,0АО
	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	
АП -8	Напряжение постоянного тока (0-10) В	13 бит	± 0,2%	0,5уо
	Постоянный ток (0-5), (0-20), (4-20)м А	13 бит	± 0,2%	
АЮ1-8/4, АЮ1-8/0	Напряжение постоянного тока (0-10) В	13 бит	± 0,2%	
	Постоянный ток (0-5), (0-20), (4-20) мА	13 бит	± 0,2%	
АЮ1-8/4, АЮ1-0/4	12 бит	Постоянный ток (0-5) мА	± 0,2%	
	12 бит	Постоянный ток (0-20), (4-	± 0,2%	

2.1.3 Схема внешних соединений

На рисунке 2.4 представлены внешние соединения в лабораторной системе.

Объектом управления является аналогово-вычислительный комплекс АНК-6, входом которого является математическая модель САУ.

номером, называемого «Слот В». Это напряжение питает блок управления (контроллер), который состоит из резидентного программного обеспечения (встроенного в контроллер), программного эмулятора для настройки, инструмента конфигурации и сервисной программы «CrossTest» для проверки УСО (унифицированной системы на кристалле).) модули. Контроллер является объектом управления, а входом является математическая модель САУ (системно-аналитического блока) на аналоговом вычислительном комплексе АНК-6.

2.2.2 Основные сведения об операционной системе OS-9

OS-9 — это высокопроизводительная операционная система, основанная на микроядерной архитектуре с модульной масштабируемой конструкцией. Он поддерживает малые, средние и большие системы и имеет развитую иерархическую систему ввода-вывода, включая последовательные порты, дисковые устройства и сетевые протоколы, такие как Ethernet, WAN и ISDN.

Конфигурация ОС-9 на контроллере "Кросс" включает в себя ядро - OS-9/68K Kernel версии V3.0.3 и пакет ISP с поддержкой сетевых протоколов, включая Интернет-протокол TCP/IP, в том числе поддержку программного интерфейса socket, и поддержка серверов FTP, Web и Telnet.

Операционная система ОС-9 обеспечивает следующие гарантированные возможности:

1. Гарантированное своевременное реагирование как на программные, так и на аппаратные прерывания и своевременное переключение задач, что соответствует требованиям систем «жесткого» реального времени.
2. Полная поддержка многозадачности.
3. Механизмы синхронизации процессов.
4. Надежная структура файловой системы.
5. Служба времени.

Благодаря этим функциям OS-9 представляет собой передовую, эффективную и масштабируемую операционную систему для малых, средних и крупных систем, включая те, которым требуются производительность и

возможности высоконадежной, эффективной и масштабируемой многозадачной операционной системы.

Благодаря интегрированной операционной системе реального времени контроллер по своим возможностям можно поставить на один уровень с универсальными компьютерами.

В контроллерах объединены мощь операционной системы реального времени ОС-9 и традиции мира ПЛК, обеспечивающие дополнительные возможности.

- Пользователь, разбирающийся в языках программирования ПЛК, может программировать контроллеры знакомыми методами без изучения ОС-9.

- Специалист, знакомый с ОС-9, может получить удовольствие от программирования на языке Си.

- Специалист, знакомый с обоими, может одновременно использовать обе эти возможности при решении своих задач, обеспечивая беспрецедентный уровень гибкости.

3 Методическое обеспечение

Разработанное в процессе выполнения ВКР программно-методическое обеспечение позволяет ознакомиться с назначением, составом, функциональными возможностями системы программирования ISaGRAF. А также данное методическое обеспечение позволяет студентам приобрести практические навыки в составлении различных программ на языке программирования FBD.

В разработанное методическое обеспечение входят методические указания по выполнению трёх лабораторных работ:

- 1.«Программный пакет ISaGRAF»;
- 2.«Составление и проверка работоспособности логических программ на языке FBD»;
- 3.«Система автоматического регулирования на языке FBD»;

Методическое обеспечение по выполнению лабораторной работы __ «Программный пакет ISaGRAF» позволяет подробно ознакомиться с назначением, функциональными возможностями системы программирования ISaGRAF, а также позволяет обучиться работе в программном пакете. В методическом обеспечении приводится цель работы, краткие сведения о системе ISaGRAF, краткий обзор языков программирования, методические указания по работе в программном пакете, задание на лабораторную работу, содержание отчёта, контрольные вопросы.

Методическое обеспечение по выполнению лабораторной работы «Составление и проверка работоспособности логических программ на языке FBD» позволяет более подробно изучить язык FBD, составить программу. В методическом обеспечении приводится цель работы, методические указания по реализации программ, задание на лабораторную работу, содержание отчёта, контрольные вопросы.

Методическое обеспечение по выполнению лабораторной работы «Система автоматического регулирования» позволяет реализовать программу аналогового ПИД-регулятора с сигнализацией по верхнему и нижнему

предельному значению регулируемой величины в системе ISaGRAF на языке FBD. Объект управления моделируется на АВК-6. Методическое обеспечение позволяет получить практические навыки по реализации систем автоматического регулирования на базе контроллера КРОСС. В методическом обеспечении приводится цель работы, методические указания по реализации системы автоматического регулирования, задание на лабораторную работу, содержание отчёта, контрольные вопросы.

Полный текст данного методического указания представлен в приложениях А данной выпускной квалификационной работы.

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
158Т92	Вэй Дуньюй

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

Исходные данные к разделу «Социальная ответственность»:

Введение	<p>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</p> <p>Количество и наименование оборудования рабочей зоны: 2 компьютера, 2 ESP32, 1 STM32, 2 линии передачи данных, 1 датчик, 1 OLED-экран Рабочие процессы, связанные с объектом исследования, осуществляющиеся в рабочей зоне:</p> <ol style="list-style-type: none"> 1. Проектировать методы беспроводных и проводных соединений. 2. Разработать логическую схему в соответствии с поставленной целью 3. Написание программного кода 4. Экспериментировать и улучшать код 5. Подготовка отчёта
----------	--

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<p>1. Производственная безопасность</p> <p>Анализ показателей шума и вибрации</p> <ul style="list-style-type: none"> • установление соответствие показателей нормативному требованию; <p>Анализ показателей микроклимата</p> <ul style="list-style-type: none"> • показатели температурные, скорости движения воздуха, запыленности. <p>Анализ освещенности рабочей зоны</p> <ul style="list-style-type: none"> • типы ламп, их количество, соответствие нормативному требованию освещенности; • при расчете освещения указать схему размещения светильников на потолке согласно проведенному расчету. <p>Анализ электробезопасности</p> <ul style="list-style-type: none"> • наличие электроисточников, характер их опасности; • установление класса электроопасности помещения, а также безопасные номиналы тока, напряжения, сопротивления заземления. 	<ul style="list-style-type: none"> • 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.
--	---

<ul style="list-style-type: none"> • при расчете заземления указать схему размещения заземлителя согласно проведенному расчету. <p>Анализ пожарной безопасности</p> <ul style="list-style-type: none"> • присутствие горючих материалов, тем самым, присутствие повышенной степени пожароопасности. • категории пожароопасности помещения, марки огнетушителей, их назначение. • Разработать схему эвакуации при пожаре. 	<ul style="list-style-type: none"> • 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.
<p>2. Экологическая безопасность:</p> <ul style="list-style-type: none"> • защита селитебной зоны • анализ воздействия объекта на атмосферу (выбросы); • анализ воздействия объекта на гидросферу (сбросы); • анализ воздействия объекта на литосферу (отходы); • разработать решения по обеспечению экологической безопасности со ссылками на НТД по охране окружающей среды. 	<p>Воздействие на селитебную зону: отсутствует.</p> <p>Воздействие на литосферу: в виде отходов, возникших при поломке персонального компьютера, люминесцентных ламп и других электроприборов. Также стоит учесть отходы макулатуры.</p> <p>Воздействие на гидросферу: продукты жизнедеятельности персонала.</p> <p>Воздействие на атмосферу: отсутствует.</p>
<p>3. Безопасность в чрезвычайных ситуациях:</p> <ul style="list-style-type: none"> • перечень возможных ЧС при разработке и эксплуатации проектируемого решения; • выбор наиболее типичной ЧС; • разработка превентивных мер по предупреждению ЧС; • разработка действий в результате возникшей ЧС и мер по ликвидации её последствий. 	<p>Возможные ЧС техногенного характера – пожар (возгорание) регламентировано ГОСТ Р 22.0.02-94</p> <p>Наиболее типичная ЧС_пожар (возгорание) регламентировано ГОСТ Р 22.0.02-94</p>

Дата выдачи задания для раздела по линейному графику	26.02.20 г.
--	-------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ООД ШБИП	Сечин Александр Иванович	д.т.н., профессор		26.02.20 г.

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
158Т92	Вэй Дуньюй		26.02.20 г.

4 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

4.1 Правовые и организационные вопросы обеспечения безопасности

Согласно ТК РФ, N 197-ФЗ работник аудитории 113, 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;
- 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 этой выпускной квалификационной работе осуществляется разработка программно-методического обеспечения для обучения языку программирования FBD программного пакета ISaGRAF на базе контроллеров “КРОСС”. Для этого используются программные пакеты, такие как ISaGRAF. В дипломном проекте применялось следующее оборудование: ПК и промышленный контроллер КРОСС-500.

Эта работа состоит из создания программного обеспечения и разработки учебных пособий для обучения студентов.

Для выбора коэффициентов используется ГОСТ 12.0.003-2015. Опасности, характерные для исследовательской среды, перечислены в таблице 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 <u>зданий</u>
Повышенный уровень шума и вибрации на рабочем месте	П 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 опасных и вредных производственных факторов

Реализация разработанного проекта может способствовать возникновению множества вредных факторов, которые могут оказать неблагоприятное влияние на самочувствие человека, его психическое состояние, условия окружающей среды и вызвать некоторые проблемы со здоровьем.

Для того чтобы предотвратить воздействие на людей уровень шума на рабочих местах с персональными компьютерами должен быть не выше допустимого значения (50 дБА), установленного СП 50.13330.2011. Шум создается вентилятором, установленным в ПК и принтере. При работе с ПК в производственных помещениях уровень вибрации должен быть не выше допустимых значений вибрации для рабочих мест (категория 3, тип “в”) по действующим санитарно-эпидемиологическим нормативам. Согласно ГОСТ 12.1.012-2004 С СБТ. 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.

Акустическая обработка помещения часто применяется для снижения шума в помещениях с ПК, с использованием звукопоглощающих материалов, обращенных к закрытым поверхностям. Диапазон частот этих поверхностей составляет (63 - 8000) Гц, где коэффициент звукопоглощения (α) максимален. Таким способом на потолок и стены устанавливаются перфорированные панели с наполнителем из минеральной ваты. Эти панели можно прикрепить непосредственно к поверхности забора или расположить на расстоянии 20 см от забора. Последний более эффективен.

Вы также можете использовать однотонные шторы, чтобы усилить эффект звукопоглощения. Шторы должны быть из плотной ткани, под цвет стены и свисать (15-20) см от оконного стекла. Ширина шторы должна быть в два раза больше ширины окна. Кроме того, его можно распечатать на малошумном лазерном принтере, чтобы снизить уровень шума.

Проект может привести к возникновению множества вредных факторов, которые могут оказать неблагоприятное влияние на здоровье человека и окружающую среду. Во избежание воздействия на здоровье человека уровень шума на рабочих местах, где используются персональные компьютеры, не должен превышать допустимого значения (50 дБА), установленного СП 50.13330.2011 . При работе с ПК в производственных помещениях уровень вибрации должен быть не выше допустимых значений вибрации для рабочих мест (категория 3, тип “в”) по действующим санитарно-эпидемиологическим нормативам . Акустическая обработка помещения может применяться для снижения шума в помещениях с ПК, с использованием звукопоглощающих материалов, обращенных к закрытым поверхностям. Шторы также могут использоваться для усиления эффекта звукопоглощения.

4.4 Экологическая безопасность

В этом подразделе оценивается воздействие проектного решения на окружающую среду. Выявлены потенциальные источники загрязнения окружающей среды в результате реализации решений, предложенных в ВКР.

При программно-методическом обеспечении разработки пакета ISaGRAF загрязнения литосферы в виде отходов в населенных пунктах и атмосфере отсутствовали. В основном это связано с повреждением персональных компьютеров, люминесцентных ламп и других электроприборов. О макулатурной бумаге также следует позаботиться. Воздействие на гидросферу Человеческие отходы также загрязняют гидросферу. Весь процесс проводился только в комнате 113, корпус 10, и никаких других эффектов не было. Искусственные отходы должны собираться и утилизироваться централизованно.

Процесс исследования представляет собой обработку информации, такой как техническая литература, статьи, ГОСТ, нормативно-технические документы, и разработку математических моделей с использованием различных программных комплексов.

Следовательно, исследовательский процесс не окажет негативного воздействия на окружающую среду.

4.5 Безопасность в чрезвычайных ситуациях

На момент исследования наиболее вероятным аварийным событием был пожар в кабинете 113 корпуса 10 ТПУ. В целях обеспечения пожарной безопасности необходимы системы противопожарной защиты и противопожарной защиты, включающие организационно-технические мероприятия.

Основными причинами возгорания являются:

1) Электрооборудование не работает, или неисправна проводка, розетка или выключатель. Во избежание пожаров, вызванных этими причинами, необходимо своевременно обнаруживать и устранять проблемы, а также регулярно проверять электрооборудование,

2) Прибор неисправен. Способ предотвращения пожара – своевременное и квалифицированное обслуживание электроприборов,

3) Система питания перегружена или в электрическом устройстве произошло короткое замыкание.

Противопожарные мероприятия включают в себя обучение пожарной безопасности и комплекс мероприятий, направленных на предотвращение пожаров.

Для обеспечения пожарной безопасности необходимо принять следующие меры:

- обучение людей тому, как избегать пожаров;
- Пожарный надзор, разрабатывающий нормы пожарной безопасности и строительные нормы и проверяющий их выполнение;
- Оборудование и технологии для установки переносных огнетушителей и производства зажигалок для безопасного использования.

В соответствии с ТР «О требованиях пожарной безопасности», внутренние противопожарные водопроводы – это технические требования пожарной безопасности, которые необходимо соблюдать в административных жилых зданиях.

Согласно ФЗ-123, НПБ 104-03 «Проектирование систем оповещения людей о пожаре в зданиях и сооружениях» Для обнаружения и оповещения о пожаре в каждом помещении должен быть независимый фотоэлектрический датчик дыма. При этом все помещения, в которых проживает или пребывает персонал, должны иметь звуковую и световую сигнализацию для подачи звука и света.

В целях предотвращения возгорания помещение 113 корпуса 10 ТПУ оборудовано первичными средствами пожаротушения: 1 огнетушитель ОУ-3 и 1 огнетушитель ОП-3. Эти огнетушители могут применяться для тушения пламени любых материалов, изделий и веществ, пригодных для тушения возгораний компьютеров и оргтехники, относящихся к классам пожаров А и Е. Типы используемых огнетушителей при пожаре в электроустановках приведены в таблице 4.2.

Таблица 4.2 - Типы используемых огнетушителей при пожаре в электроустановках.

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

Правила категорирования зданий и помещений по взрывопожарной и пожарной опасности указаны в главе 8 Техрегламента о требованиях пожарной безопасности, СП12.13130.2009. Согласно положениям документа уровень взрывопожароопасной кабинета 113 корпуса 10 ТПУ - В1-В4. И на этот стандарт следует ориентироваться при подборе огнетушителей.

Важными объектами 10 корпуса ТПУ являются устройства пожарной автоматики и сигнализации. Они могут автоматически отключать электрооборудование и системы вентиляции в случае пожара, снижая риск распространения огня. В то же время они также могут сообщить соответствующему персоналу о пожаре и принять своевременные меры по тушению пожара (начальный этап) для обеспечения безопасной эвакуации студентов и персонала.

В случае пожара в компьютерном зале очень важен план эвакуации. Вот несколько практических планов:

1. Знать планировку компьютерного зала: В случае пожара нужно знать, как покинуть компьютерный зал. Знание планировки вашего компьютерного зала может помочь вам быстро найти ближайший выход.

2. Сохраняйте спокойствие. Во время пожара очень важно сохранять спокойствие. Не паникуйте, но покиньте компьютерный зал как можно быстрее.

3. Используйте огнетушитель: если вы обнаружите небольшой огонь, попробуйте использовать огнетушитель, чтобы потушить источник огня. Однако, если вы не знаете, как пользоваться огнетушителем или если огонь слишком большой, немедленно эвакуируйтесь.

4. Следуйте путям эвакуации: В случае пожара очень важно следовать путям эвакуации. Следуйте указателям и указателям и покиньте компьютерный зал как можно быстрее.

5. Убедитесь, что весь персонал покинул компьютерный зал: прежде чем покинуть компьютерный зал, убедитесь, что весь персонал ушел. Если вы обнаружите, что кто-то все еще находится в компьютерной комнате, немедленно сообщите об этом и помогите эвакуироваться.

Вывод: Исследователи должны полностью понимать пожарную опасность исследовательской среды, иметь полное представление о действиях, которые могут привести к пожарам, и быть знакомыми с планами эвакуации и путями эвакуации в случае пожара.

В данной статье в качестве объекта исследования взят анализ опасных и вредных производственных факторов в проекте разработки, а также рассмотрены проблемы шума, вибрации, экологической безопасности и аварийной безопасности посредством анализа литературы и экспериментальной проверки. Основные выводы следующие:

Шум является одной из наиболее значительных опасностей в девелоперских проектах, влияющих на здоровье и производительность труда персонала. Для снижения уровня шума в этой статье предлагаются такие меры, как использование звукопоглощающих материалов и принтеров с низким уровнем шума, чтобы уровень шума не превышал 50 дБА.

Вибрация является еще одной серьезной опасностью в проектах разработки, вызывая усталость и травмы персонала. Для снижения воздействия вибрации на персонал в данной работе рекомендуется использовать антивибрационные прокладки, регулировку рабочего времени и другие методы, а скорость вибрации не превышать 75 дБ.

Экологическая безопасность – это вопрос, который нельзя игнорировать в девелоперских проектах, связанных с утилизацией отходов, охраной водных ресурсов и т.д. В данной статье анализируются возможные источники

загрязнения окружающей среды в девелоперских проектах, такие как выход из строя электрооборудования, макулатура, человеческие экскременты и т. д., и предлагаются соответствующие методы очистки, такие как своевременное техническое обслуживание, переработка и централизованная очистка.

Аварийная безопасность — это вопрос, который необходимо учитывать в проектах развития, включая предотвращение пожаров, системы противопожарной защиты и другие аспекты. В данной статье указываются возможные причины пожара в проекте разработки, такие как перегрузка электрооборудования, дефекты электрооборудования и т. д., и предлагаются соответствующие противопожарные меры, такие как регулярный осмотр, противопожарная подготовка и т. д.

В данной статье проводится относительно всестороннее и глубокое исследование по анализу опасных и вредных производственных факторов в девелоперских проектах, а также приводятся определенные справочные и руководящие указания по обеспечению безопасности персонала и окружающей среды.

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
158Т92	Вэй Дуньюй

Школа	ИШИТР	Отделение школы (НОЦ)	ОАР
Уровень образования	Бакалавриат	Направление/специальность	15.03.04 Автоматизация технологических процессов и производств

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

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	Среднерыночные цены РФ для определения стоимости материальных ресурсов. Нормативные документы НИ ТПУ, ФЗ «О минимальном размере оплаты труда» для определения оплата труда исполнителей проекта.
2. <i>Нормы и нормативы расходования ресурсов</i>	Тариф электроэнергии 3,16 руб. кВт/ч., 30% районный коэффициент
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	Отчисления в социальные внебюджетные фонды 30%;

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. <i>Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения</i>	Оценить потенциальных потребителей исследования
2. <i>Планирование и формирование бюджета научных исследований</i>	Представить план этапов работ, определить трудоёмкость и построить календарный график, сформировать бюджет НИ.
3. <i>Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования</i>	Определить интегральный показатель эффективности научного исследования.

Перечень графического материала (с точным указанием обязательных чертежей):

1. Перечень этапов, работ и распределение исполнителей 2. Временные показатели проведения НИ 3. Бюджет НИ 4. Оценка характеристик вариантов исполнения 5. Сравнительная эффективность разработки.

Дата выдачи задания для раздела по линейному графику	01.03.2023.г
---	--------------

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент ОСГН, ШБИП	Былкова Татьяна Васильевна	канд.экон.наук		01.03.2023.г

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
158Т92	Вэй Дуньюй		01.03.2023.г

5 ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ

5.1 Планирование научно-исследовательских работ

Есть два исполнителя, руководитель проекта (НР) и инженер, и лучше всего использовать линейную диаграмму. Чтобы построить его, выше в хронологическом порядке в таблице:

Таблица 5.1.1 - Список задач и исполнители

Этапы работы	Исполнители	Загрузка исполнителей
Постановка целей и задач, получение исходных данных	НР	НР – 100%
Составление и утверждение ТЗ	НР,И	НР – 100% И – 20%
Разработка календарного плана	НР,И	НР – 100% И – 20%
Подбор и изучение материалов по тематике	НР,И	НР – 60% И – 100%
Обсуждение литературы	НР,И	НР – 70% И – 100%
Ознакомление с пакетом программирования контроллера Кросс – Isagraf	И	И – 100%
Исследование и анализ прогпамов на языке FBD	И	И – 100%
Разработка программного обеспечения	НР,И	НР – 40% И – 100%
Разработка методического обеспечения	НР,И	НР – 30% И – 100%
Технико-экономическое обоснование проекта	НР,И	НР – 30% И – 100%
Оценка безопасности и экологичност	НР,И	НР – 40% И – 100%
Составление и оформление пояснительной записки	И	И – 100%
Защита дипломного проекта	И	И – 100%

Расчет продолжительности рабочего сеанса производится экспертным путем. Для определения возможных (ожидаемых) значений продолжительности работы используется следующая формула:

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

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

Чтобы построить линейный график, вам нужно рассчитать продолжительность периода в будних днях, а затем перевести его в календарные дни. Расчет продолжительности каждой фазы в рабочих днях ($T_{рд}$) производится по следующей формуле:

$$T_{рд} = \frac{t_{ож}}{K_{вн}} \cdot K_{д}$$

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

$K_{вн}$ – коэффициент выполнения работ, учитывающие влияние внешних факторов на соблюдение заданной продолжительности, особенно возможный $K_{вн} = 1$;

$K_{д}$ – коэффициент, учитывающие дополнительное время на компенсацию непредвиденных задержек и координационную работу ($K_{д} = 1-1,2$; в этих пределах точное значение остается за исполнителем). Примем коэффициенты равными: $K_{вн} = 1; K_{д} = 1,2$. Расчет продолжительности фазы в календарных днях производится по формуле:

$$T_{кд} = T_{рд} \cdot T_{к}$$

где $T_{кд}$ – Продолжительность этапа (в календарных днях);

$T_{к}$ – Календарный коэффициент, который позволяет перейти от продолжительности рабочего дня в рабочий день к соответствующей продолжительности в календарном дне, и рассчитывается по следующей формуле:

$$T_{к} = \frac{T_{кал}}{T_{кал} - T_{вд} - T_{пд}}$$

Где: $T_{кал}$ – календарные дни ($T_{кал} = 366$); $T_{вд}$ – выходные дни ($T_{вд} = 52$); $T_{пд}$ – праздничные дни ($T_{пд} = 14$). В нашем примере расчеты производятся для шестидневной рабочей недели, $T_{к} = 1,22$.

Таблица 5.1.2 - Трудозатраты на выполнение проект

Этапы работы	Исполнители	Продолжительность работ, дни	Трудоемкость работ по исполнителям чел.-дн
--------------	-------------	------------------------------	--

		tmin	tmax	тож	НР	И	НР	И
Постановка целей и задач, получение исходных данных	НР	1	2	1.4	1.68	0	2.05	0
Составление и утверждение ТЗ	НР,И	1	3	1.8	2.16	0.43	2.64	0.52
Разработка календарного плана	НР,И	2	5	3.2	3.84	0.77	4.68	0.94
Подбор и изучение материалов по тематике	НР,И	3	4	3.4	2.45	4.08	2.99	4.98
Обсуждение литературы	НР,И	1	2	1.4	1.18	1.68	1.44	2.05
Ознакомление с пакетом программирования контроллера Кросс – Isagraf	И	5	10	7	0	8.4	0	10.24
Исследование и анализ прогамов на языке FBD	И	6	12	8.4	0	10.08	0	12.3
Разработка программного обеспечения	НР,И	10	15	12	5.76	14.4	7.03	17.57
Разработка методического обеспечения	НР,И	7	12	9	3.24	10.8	3.95	13.18
Технико-экономическое обоснование проекта	НР,И	2	4	2.8	1.01	3.36	1.23	4.1
Оценка безопасности и экологичности работы	НР, И	1	2	1.4	0.5	1.68	0.61	2.05
Составление и оформление пояснительной записки	И	7	14	9.8	0	11.76	0	14.35
Защита дипломного проекта	И	1	2	1.4	0	1.68	0	2.05
Итого:				63	21.82	67.44	26.62	82.28

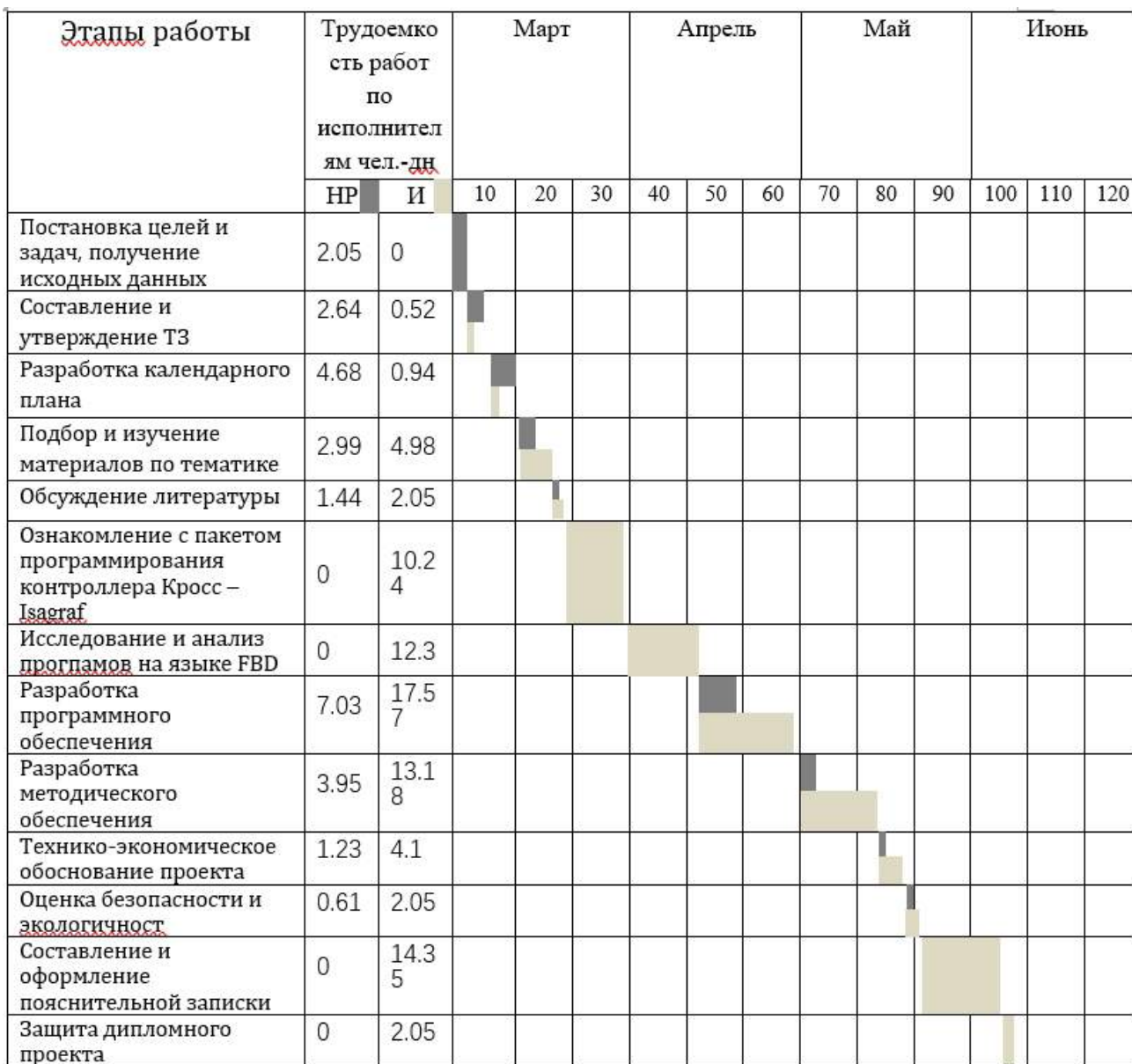


Рисунок 5.1.3 - Линейный график работ

5.2 Бюджет научно-технического исследования

В таблице 5.2.1 показан расчет стоимости материалов.

Таблица 5.2.1 - Расчет стоимости материала

Наименование материалов	Цена за ед., руб	Кол-во	Сумма, руб.
USB Flash Drive 32 Gb	300	1	300
Ручки	70	2	140
Карандаш	100	1	100
Тетрадь	120	1	120
Бумага для принтера формата А4	200	1	200
Картридж для принтера	1200	1	1200
Итого:			2060

Расчеты тарифов на электроэнергию для исследовательских целей приведены в таблице 5.2.2.

Таблица 5.2.2 - Затраты энергии на исследования

Наименование оборудования	Время работы оборудования $t_{\text{ОБ}}$, час	Потребляемая мощность $P_{\text{ОБ}}$, кВт	Затраты $\Delta_{\text{ОБ}}$, руб.
Персональный компьютер	543,3	0,3	162,99
Принтер	2,1	0,2	0,42
Контроллер КРОСС	489,2	0,2	97,84
Итого:			261,25

В статье «Амортизационные расходы» рассчитывается амортизация оборудования, используемого в ходе проекта.

Используется формула

$$C_{\text{АМ}} = \frac{N_{\text{А}} * C_{\text{ОБ}} * t_{\text{рф}} * n}{F_{\text{Д}}},$$

где $N_{\text{А}}$ – Годовая норма амортизации оборудования;

$C_{\text{ОБ}}$ – Балансовая стоимость оборудования, включая ТЗР.

$F_{\text{Д}}$ – Фактическое годовое финансирование часов работы соответствующего оборудования.

Для ПК (300 рабочих дней при шестидневной рабочей неделе) можно принять $F_{\text{Д}} = 300 * 8 = 2400$ часа;

Для принтера $F_{\text{Д}} = 500$ часов;

Для контроллера КРОСС $F_{\text{Д}} = 2400$ часов;

$t_{\text{рф}}$ – Фактическое время работы устройства во время выполнения

Для ПК $t_{\text{рф}} = 543,3$ часа. Для принтера $t_{\text{рф}} = 2,1$ часов. Для контроллера КРОСС $F_{\text{Д}} t_{\text{рф}} = 489,2$ час

n – Количество задействованных единиц однотипной техники. При использовании нескольких видов оборудования выполните соответствующие расчеты по формуле, а затем подведите итоги. Для определения НС следует

обратиться к Постановлению Правительства РФ «О классификации основных средств, включаемых в амортизационные группы». Он позволяет получить расчетную стоимость амортизации (полезного использования) оборудования СА.

Для ПК возьмем СА=4 года. Далее определяется НА как величина обратная СА, в данном случае это $1/4 = 0,25$. Для принтера СА= 2 года, НА= 0,5. Для контроллера КРОСС СА= 3 года НА= 0,33;

САМ(ПК) = $(0,25*50000*543,3*1)/2400 = 2829,69$ руб; САМ (Пр) = $(0,5*20000*2,1*1)/500 = 42$ руб; САМ (К) = $(0,33*30000*489,2 *1)/2400 = 2038,33$ руб. Итого начислено амортизации 4910,02 руб.

Расчет базового оклада осуществляется в зависимости от сложности каждого этапа и размера месячного оклада исполнителя. Среднедневная заработная плата рассчитывается по следующей формуле:

$$ЗП_{дн-г} = MO/25.44,$$

Учитывая, что в году 300 рабочих дней, получается в среднем 25 рабочих дней в месяц (6 рабочих дней в неделю).

Расчет полной стоимости заработной платы приведен в таблице 2.1. Время, затраченное каждым исполнителем в течение рабочего дня, округленное до целого числа, взятое из таблицы 1.1. Для учета в ее составе надбавок, доплат и районных надбавок используются следующие ряды коэффициентов: КПП = 1,1; Кдоп.ЗП = 1,188; Кр = 1,3. Следовательно, для перевода из тарифной (базовой) суммы доходов исполнителей, связанных с участием в проекте, в соответствующие им полные доходы (окладную часть сметы) необходимо первую умножить на интегральный коэффициент $K_{и} = 1,1 * 1,188 * 1,3 = 1,699$. Вышеуказанные значения для Кдоп. ЗР указаны для шестидневной недели.

Таблица 5.2.3 - Затраты на заработную плату

Исполнитель	Оклад, руб./мес	Среднедневная ставка, руб./раб.день	Затраты времен и,раб.дни	Коэффициент	Фонд з/платы, руб
НР	23556,35	925,96	27	1,699	42476,56
И	14670,25	576,66	82	1,699	80339,22
Итого:					122815,78

Отчисление во внебюджетные фонды = $122815,78 * 0,3 = 36844,73$ руб. В статье «Прочие расходы» отражается стоимость проекта, которая не учтена в предыдущих статьях, и они должны быть равны 10%.

Добавим 10% к сумме, т.е. $С_{\text{проч.}} = (С_{\text{мат}} + С_{\text{зп}} + С_{\text{соц}} + С_{\text{эл.об.}} + С_{\text{ам}} + С_{\text{проч.}}) \cdot 0,1$.

Для нашего примера это $С_{\text{проч.}} = (2060 + 122815,78 + 36844,73 + 261,25 + 4910,02) \cdot 0,1 = 19275,55$ руб.

Проведя расчет по всем статьям сметы затрат на разработку, можно определить общую себестоимость проекта «Программно-методическое обеспечение для изучения языка программирования FBD программного пакета ISaGRAF»

Таблица 5.2.4 Смета затрат на разработку проекта

Статья затрат	Условное обозначение	Сумма, руб.
Материалы и покупные изделия	$C_{\text{мат}}$	2060
Основная заработная плата	$C_{\text{зп}}$	122815,78
Отчисления в социальные фонды	$C_{\text{соц}}$	36844,73
Расходы на электроэнергию	$C_{\text{эл}}$	261,25
Амортизационные отчисления	$C_{\text{ам}}$	4910,02
Прочие расходы	$C_{\text{проч}}$	19275,55
Итого:		212031,08

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

Определение эффективности основано на расчете интегральных показателей эффективности научных исследований. Его выводы связаны с определением двух средневзвешенных значений: финансовой эффективности и эффективности использования ресурсов.

Общий показатель финансовой эффективности научных исследований может быть получен в процессе оценки бюджета на проведение научных

исследований. Интегральный финансовый показатель разработки определяется как:

$$I_{финр} = \frac{\Phi_p}{\Phi_{max}},$$

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

Φ_p – стоимость исполнения;

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

$$I_{финр \text{ текущая разработка}} = 212031,08 / 208176,21 = 1,018$$

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

Результирующее значение общего финансового показателя развития отражает соответствующее численное увеличение бюджета затрат на разработку (значение больше 1) или соответствующее численное уменьшение стоимости разработки (значение меньше 1).

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

$$I_p = \sum a \cdot b,$$

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	0.3	3	2
2.Теоретический уровень	0.2	4	3
3.Возможность реализации	0.5	3	3

Итого	1	3.2	3
-------	---	-----	---

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

$$I_{\text{текущая разработка}} = I_p / I_{\text{финр}} = 3.14$$

$$I_{\text{вариант исполнения 1}} = I_p / I_{\text{финр}} = 3$$

Сравнительная эффективность проекта

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

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

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

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

$$\mathcal{E}_{\text{ср 1}} = 3.14 / 3 = 1.05$$

Отразим результаты расчета сравнительной эффективности разработки в таблице 5.3.2.

Таблица 5.3.2 – Сравнительная эффективность разработки

№ п/п	Показатели	Разработка	Исп1
1	Интегральный финансовый показатель разработки	1,02	1
2	Интегральный показатель ресурсоэффективности разработки	3,2	3
3	Интегральный показатель эффективности	3,14	3
4	Сравнительная эффективность вариантов исполнения	1,01	

В процессе оценки бюджета исследования и определения общих финансовых показателей, показателей эффективности использования ресурсов можно сделать вывод, что рассчитанные финансовые показатели и показатели общей эффективности свидетельствуют об эффективности программы

Заключение

В результате выпускной квалификационной работы:

- изучил язык FBD в системе ISaGRAF;
- рассмотрел и изучил контроллер «КРОСС»;
- приобрел практические навыки программирования контроллера «КРОСС» в системе ISaGRAF;
- разработал лабораторный проект. методическое пособие, подробно описывающее работу системы ISaGRAF и программирование контроллера КРОСС на различных языках;
- разработанное руководство (Приложение А) будет использовано для лабораторных работ студентов направления 15.03.04 «Технический Автоматизация процессов и производства».

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 Спириин 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.

ПРИЛОЖЕНИЕ А

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ЛАБОРАТОРНОЙ РАБОТЫ

**«Программно-методическое обеспечение для изучения языка программирования
FBD программного пакета ISaGRAF»**

Вэй Дуньюй

**Программно-методическое обеспечение для изучения языка
программирования FBD 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

2023

СОДЕРЖАНИЕ

1.СИСТЕМА ПРОГРАМИРОВАНИЯ КОНТРОЛЛЕРОВ ISaGRAF	3
1.1Краткие сведения о системе ISaGRAF	3
1.2 Архитектура проекта	3
1.3 Основные типы данных	4
1.4 Работа в системе разработки ISaGRAF	5
2 СТРУКТУРНАЯ СХЕМА ЛАБОРАТОРНОГО КОМПЛЕКСА	7
2.1 Структурная схема лабораторного комплекса.....	7
2.2 Аналоговый вычислительный (АВК-6).....	8
2.2.1 Назначение АВК-6.....	8
2.2.2 Общие характеристики и состав АВК-6	9
2.2.3 Конструкция АВК-6	9
3 ЯЗЫК ФУНКЦИОНАЛЬНЫХ БЛОКОВ FBD.....	10
3.1 Основной формат функциональных блоков FBD.....	11
3.2 Оператор RETURN.....	12
3.3 Прыжки и метки.....	12
3.4 Логическое отрицание	13
3.5 Вызов функций и функциональных блоков из FBD	13
3.6 Создание функциональных блоков	14
3.7 Объявление переменных.....	20
4 Примеры создания программ в системе ISaGRAF на языке FBD.....	24
4.1 Создание простейшей программы, реализующей операцию логического ИЛИ на языке FBD в режиме симуляции	24
4.2 Создание элементарной программы аналогового ввода/вывода	35
4.3 Создание элементарной программы дискретного ввода/вывода	40
4.4 Создание элементарной программы аналогового и дискретного ввода/вывода	43
4.5 Создание программы, реализующей ПИД-регулятор.....	46

Цель работы: получение практических навыков программирования контроллера КРОСС-500 в системе ISaGRAF на языках FBD

1. СИСТЕМА ПРОГРАММИРОВАНИЯ КОНТРОЛЛЕРОВ ISaGRAF

1.1 Краткие сведения о системе ISaGRAF

Для программирования промышленного контроллера КРОСС-500 используется система ISaGRAF.

ISaGRAF — это французская реализация стандарта Международной электротехнической комиссии (МЭК) 61131-3. Система состоит из двух частей: средств разработки ISaGRAF Workbench и ядра интерпретатора ISaGRAF Target, которое находится в контроллере. ISaGRAF Workbench включает в себя редактор, средство отладки и препроцессор и позволяет пользователю создавать программы с использованием языков, определенных стандартом IEC 61131-3: SFC, FBD, LD, ST и IL.

После разработки программы пользователя ее можно загрузить в контроллер для выполнения. Ядра интерпретатора ISaGRAF Target транслируют пользовательские алгоритмы на лету во время выполнения, снижая затраты, связанные с переключением на другой контроллер.

Для программирования промышленного контроллера КРОСС-500 с использованием ISaGRAF такие языки, как SFC, FBD, LD, являются графическими, а ST и IL — текстовыми.

1.2 Архитектура проекта

Система ISagraf состоит из нескольких программных модулей, включая SFC, FBD, LD, ST и IL. Он имеет иерархическую структуру и поддерживает несколько языков программирования. Программы — это логические единицы, описывающие операторы и переменные, связанные с процессом. Система

разделена на четыре основных раздела: начало, последовательность, конец и функция.

В начальной секции программы выполняются в начале каждого цикла целевой системы. Последовательный раздел содержит программы, которые управляются динамическими правилами SFC и выполняются последовательно. Конечные программы выполняются в конце каждого цикла целевой системы. Секция функций содержит программы, которые могут быть вызваны любой программой в проекте, а также имеет возможность вызывать другие программы в той же секции.

Циклические операции, не зависящие от времени, описываются программами в разделе «Начало» и «Конец», а последовательные операции описываются программами в разделе «Последовательный».

В системе ISaGRAF язык SFC используется для описания основных и дополнительных программ в разделе Sequential. Этот язык не подходит для описания циклических программ в разделах «Begin» и «End».

Секция Begin предназначена для считывания сигналов с датчиков и выполнения предварительных операций с устройствами ввода и отфильтрованными переменными. Секция End предназначена для отправки сигналов на датчики и выполнения защитных операций с переменными перед их отправкой на устройства вывода.

1.3 Основные типы данных

В системе ISaGRAF каждая константа, выражение или переменная, используемые в программе, должны иметь определенный тип. К основным типам объектов, которые может представлять программа, относятся:

BOOLEAN: логическое значение, которое может быть TRUE или FALSE

ANALOG: Непрерывное или дискретное (целое или действительное) значение, которое может изменяться во времени.

TIMER: значение, основанное на времени, которое ограничено одним днем или меньше.

MESSAGE: строка символов

Важно отметить, что переменная TIMER (на основе времени) используется только для данных, связанных со временем, и не может использоваться для долговременного хранения информации.

Программы также имеют собственный поток данных, который можно представить в виде графиков. Типы данных, которыми можно обмениваться между программами, включают:

ВВОД: значение данных, предоставляемое устройствами ввода программы, такими как датчики или операторы.

ВЫХОД: значение данных, отправляемое из программы на ее устройства вывода, такие как экраны дисплея или приводы.

СОСТОЯНИЕ ПРОГРАММЫ: значение данных, отражающее текущее состояние или статус программы, например «РАБОТАЕТ» или «ОСТАНОВЛЕНО».

Эти типы данных используются во всей системе ISaGRAF, чтобы обеспечить согласованность, надежность и безопасность программ. Определяя определенные типы для различных элементов программы, система может легче интерпретировать и анализировать поведение программы, что может помочь выявить ошибки или ошибки до того, как они вызовут более серьезные проблемы.

1.4 Работа в системе разработки ISaGRAF

В ISaGRAF шаги по созданию программы следующие:

1. Запустите систему ISaGRAF и создайте новый проект.
2. Начните создавать программу, указав имя, язык и стиль программы.
3. Объявите переменные, которые будут использоваться в программе.
4. После написания программы проверьте наличие ошибок с помощью команды File/Check. Если ошибок нет, сохраните программу и закройте окно редактирования.
5. Сгенерируйте код проекта с помощью команды «Файл/Создать

приложение» в окне «Диспетчер программ» или нажмите кнопку «Создать код приложения».

6. Чтобы протестировать программу, вы можете запустить симуляцию с помощью команды «Начать», а затем запустить симуляцию с помощью команды «Выполнить/Пуск» в окне «Диспетчер симуляции».

7. Если вам нужно изменить программу, вы можете сделать это, перезапустив симуляцию и изменив программу, или повторно открыв окно редактирования программы и внося изменения.

Важно отметить, что разные стили программы, такие как «Начало», «Последовательный» и «Конец», имеют разное поведение и используются для разных целей. Например, программы Begin выполняют инициализацию и устанавливают значения переменных до начала выполнения любых других программ, в то время как программы Sequential выполняют задачи в определенном порядке, а программы End выполняют действия по завершению работы и очистке.

В целом, процесс создания программы в ISaGRAF включает тщательное сочетание объявления переменных, написания кода, тестирования кода и исправления любых возникающих ошибок или проблем. С практикой вы сможете научиться использовать эту систему для создания надежных и эффективных программ контроля и мониторинга процессов.

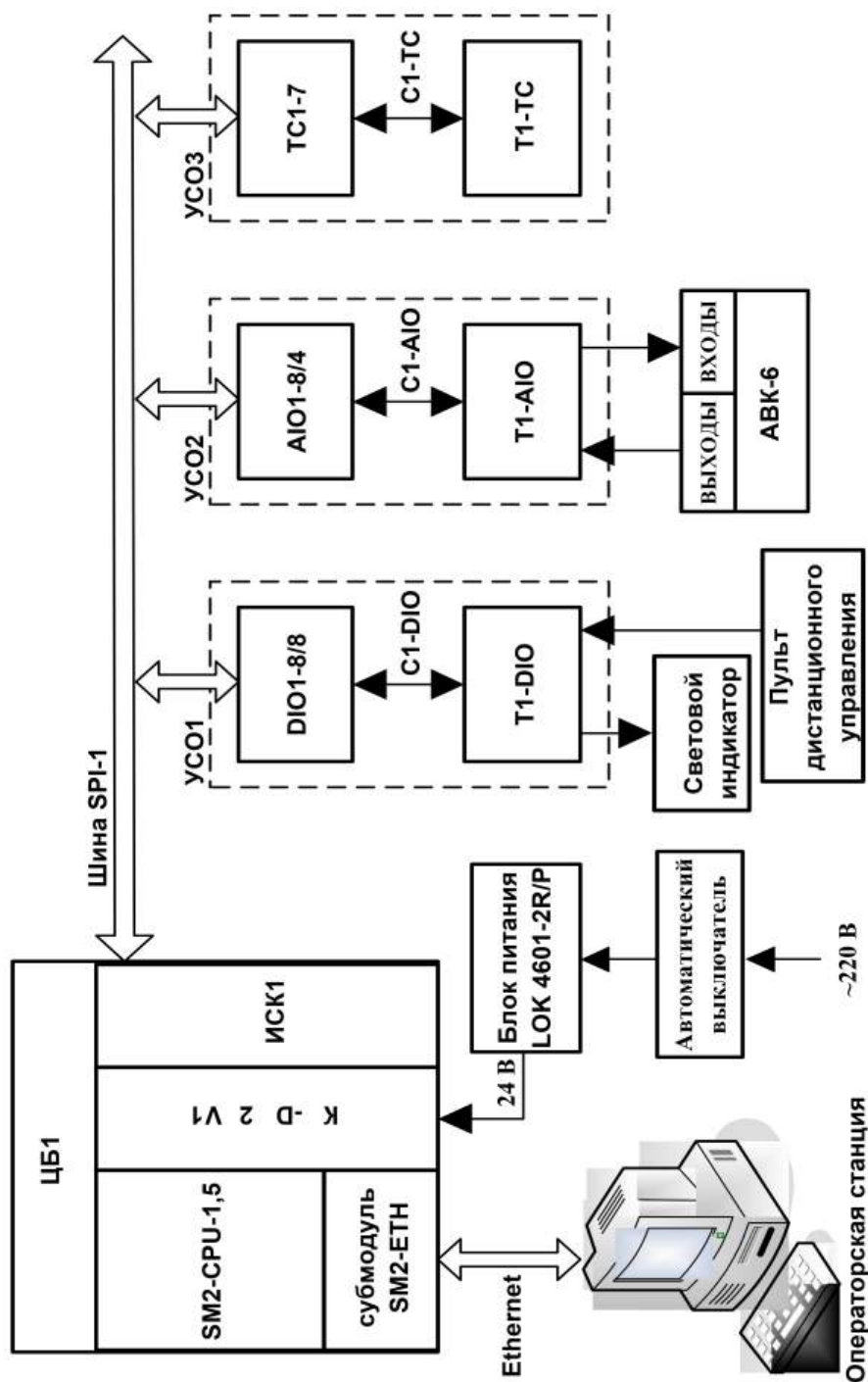
2 СТРУКТУРНАЯ СХЕМА ЛАБОРАТОРНОГО КОМПЛЕКСА

2.1 Структурная схема лабораторного комплекса

Лабораторный комплекс состоит из аналогово-вычислительного комплекса АНК-6, персонального компьютера под управлением Windows 2000 и контроллера Кросс-500. АНК-6 используется для создания математической модели системы управления, которая служит объектом управления лабораторным комплексом. Контроллер Cross-500 получает входные данные от датчиков и отправляет выходные сигналы на исполнительные механизмы, которые используются для управления изучаемыми физическими процессами.

Персональный компьютер под управлением Windows 2000 служит пользовательским интерфейсом для лабораторного комплекса, позволяя операторам контролировать систему и при необходимости настраивать параметры управления. Сочетая мощь аналоговых вычислений с удобным интерфейсом персонального компьютера, этот лабораторный комплекс обеспечивает эффективный и действенный способ изучения и контроля физических процессов.

Структурная схема комплекса приведена на рисунке 2.1.



2.2 Аналоговый вычислительный (АВК-6)

2.2.1 Назначение АВК-6

АВК-6 способен выполнять широкий спектр сложных расчетов и моделирования, что делает его идеальным инструментом как для студентов, так и для специалистов в этих областях. Используя АВК-6, пользователи могут получить более глубокое понимание принципов работы динамических систем и их практических приложений, таких как моделирование и анализ систем управления для различных типов машин и оборудования. Практический подход к обучению и использованию АВК-6 дает ценный опыт математического моделирования, моделирования и решения задач, который можно применять в различных областях, включая инженерию, физику и математику.

2.2.2 Общие характеристики и состав АВК-6

Аналоговый вычислительный комплекс АВК-6 состоит из следующих компонентов:

1. Малая Аналитическая Машина 6 (МЭМ-6);
2. Система индикаторно-измерительных приборов (СИИС);
3. система FMI.

В целом АВК-6 обеспечивает удобный и эффективный подход к моделированию и изучению динамических процессов в широком диапазоне областей, включая технику управления, электротехнику и машиностроение. Благодаря использованию МЭМ-6 и системы FMI пользователи могут получить более глубокое понимание принципов и приложений динамических систем, а также развить технические навыки и улучшить способность решать проблемы.

2.2.3 Конструкция АВК-6

Аналоговый вычислительный комплекс АНК-6 представляет собой

портативное модульное устройство, состоящее из нескольких функциональных компонентов, в том числе аналоговой вычислительной машины «Малая Аналитическая Машина 6» (МЭМ-6), которая предназначена для проведения математического моделирования и имитационного моделирования при исследовании динамических процессов.

Корпус комплекса заключен в надежное и прочное шасси, а каждый блок и модуль, включая модуль питания, крепятся к шасси с помощью винтов для удобства установки и ремонта. Модульная конструкция также упрощает техническое обслуживание, так как каждый блок и модуль могут быть доступны и удалены по мере необходимости. В целом АНК-6 представляет собой мощный инструмент для изучения и моделирования динамических процессов, а его модульная конструкция делает его удобным и гибким для использования в различных условиях, включая инженерные, физические и математические.

Расположение блока, модулей на корпусе АВК 6 показано на рисунке 2.2

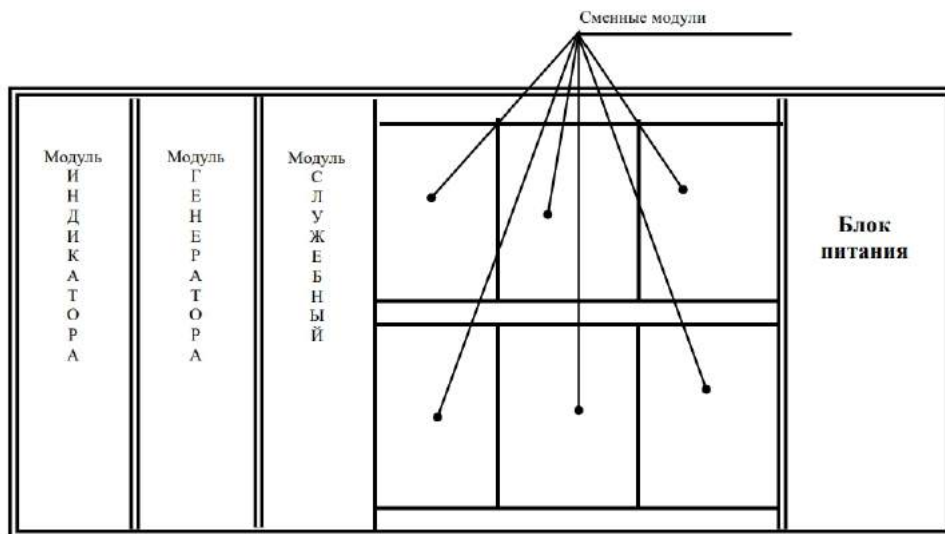


Рисунок 2.2 - Состав АВК 6

3 ЯЗЫК ФУНКЦИОНАЛЬНЫХ БЛОКОВ FVD

ISaGRAF — это библиотека функциональных блоков для инженерных расчетов, которая имеет большую базу пользователей, особенно среди электротехнических дисциплин. Библиотека обычно используется для проектирования электронных схем, включая аналоговые, цифровые схемы и

схемы со смешанными сигналами. По сравнению с классическими схемами релейной логики, FBD предлагает гораздо более наглядный и простой подход к пониманию электронных схем. FBD стал важным инструментом при проектировании и анализе сложных схем, требующих высокого уровня детализации и точности. Фактически, он был неотъемлемой частью инженерной учебной программы университетов во многих странах, включая Соединенные Штаты и Европу, и часто преподается в рамках программ получения инженерных степеней или технических дипломов.

Таким образом, он считается наиболее часто используемым языком для проектирования электронных схем с широким набором функций и возможностей, которые делают его подходящим для различных приложений, от бытовой электроники до аэрокосмической техники.

3.1 Основной формат функциональных блоков FBD

Блок FBD определяет функцию между входными и выходными переменными, и эта функция состоит из набора элементарных функциональных блоков. Входные и выходные переменные подключаются к блокам с помощью линий (рисунок 3.1), а выход одного функционального блока может быть подключен к входу другого функционального блока. Соединив выход одного функционального блока со входом другого, можно создать цепочку функций, позволяющую формировать сложные схемы. Этот процесс соединения блоков и формирования цепей имеет важное значение при проектировании электронных схем. Предоставляя визуальное представление схемы, FBD позволяет инженерам легко понять и проанализировать поведение схемы, что делает его жизненно важным инструментом при разработке цифровых и аналоговых схем.

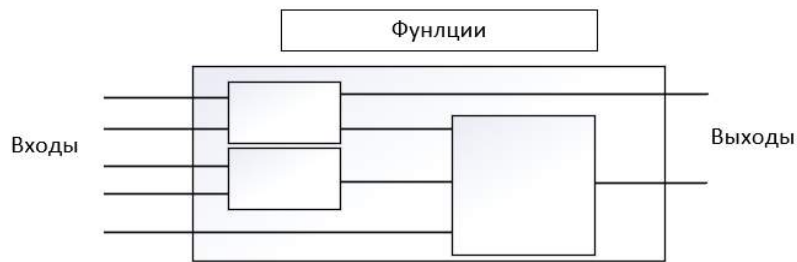


Рисунок 3.1 Описание функций на языке FBD



Рисунок 3.2 Графическое обозначение функциональных блоков

3.2 Оператор RETURN

Оператор "<RETURN>" используется для условного завершения программы: если выход блока, подключенного к оператору, является TRUE, оставшие части диаграммы не выполняются. Пример FBD программы, использующей оператор <RETURN >, представлен на рисунке 3.3.

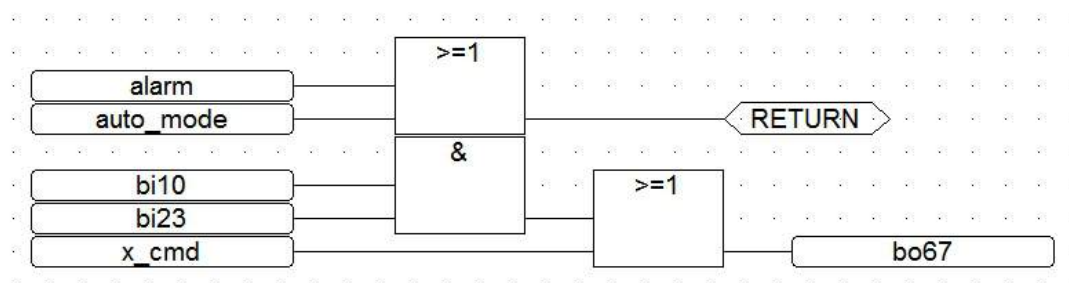


Рисунок 3.3 — Пример программы, использующей оператор <RETURN >

3.3 Прыжки и метки

Прыжки и метки используются для управления Исполнением программ на диаграммах FBD управляют переходы и метки. Никакие другие объекты не могут быть подключены к правой стороне метки или символа перехода. Используются следующие обозначения:

>LAB перейти к метке (называемой "LAB")

LAB: определение метки (называемой "LAB")

Если соединение слева от символа перехода находится в состоянии ИСТИНА,

выполнение программы перейдет к соответствующей метке.

Пример программы FBD, использующей метки и переходы, показан на рис. 3.4. В этом примере, если manual=TRUE и b1=TRUE одновременно, выполнение программы перейдет к метке NOMODIF, а средний функциональный блок (>=1) выполняться не будет.

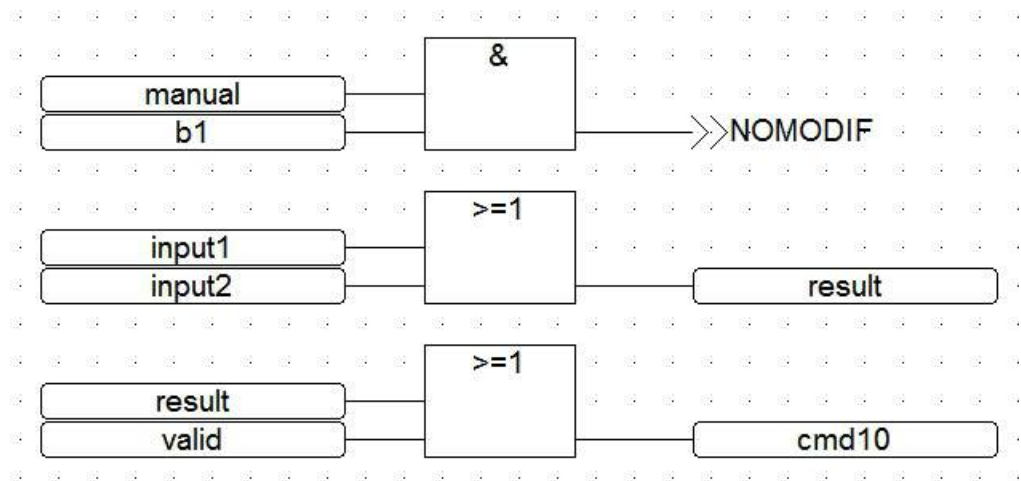


Рисунок 3.4 - Пример FBD программы с использованием меток и прыжков

3.4 Логическое отрицание

На самом деле единственная линия, соединяющая правый край функционального блока, может иметь на конце логическую отрицательную окружность. Отрицание представляет собой обратную сторону условия. При использовании логического отрицания в соединении строк как левый, так и правый край строки должны иметь тип BOOLEAN. Это позволяет программе инвертировать результат выражения, чтобы сделать результат противоположным исходному.

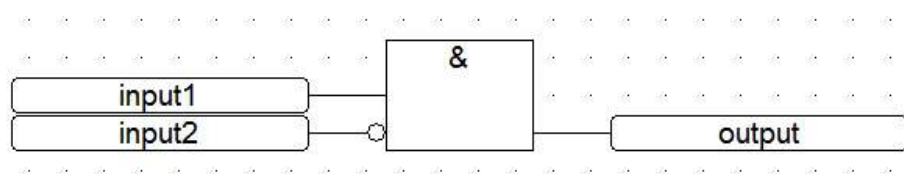


Рисунок 3.5 - Пример FBD программы, использующей логическое отрицание

3.5 Вызов функций и функциональных блоков из FBD

В FBD (диаграмме функциональных блоков) функции и подпрограммы можно вызывать с использованием прямоугольной формы, которая представляет имя подпрограммы, функции или блока. Функция или подпрограмма имеет единственный возвращаемый параметр, который является возвращаемым значением. Функциональные блоки могут иметь несколько выходных параметров.

При вызове подпрограммы или функции возвращаемое значение передается вызывающему оператору в качестве входного параметра. Если вызывается функциональный блок, выходные параметры возвращаются как отдельные значения для использования в остальной части программы.

Очень важно убедиться, что подпрограмма или функция и вызывающий оператор имеют совместимые типы параметров, иначе может возникнуть ошибка времени выполнения. Кроме того, рекомендуется убедиться, что подпрограмма или функция правильно определена и реализована, прежде чем вызывать ее из диаграммы FBD.

3.6 Создание функциональных блоков

Для создания программы на FBD используйте библиотеку базовых функций арифметики, логики и динамики. Выберите нужный функциональный блок в панели инструментов и перетащите его в рабочее пространство программы с помощью мыши, как показано на рисунке 3.6.

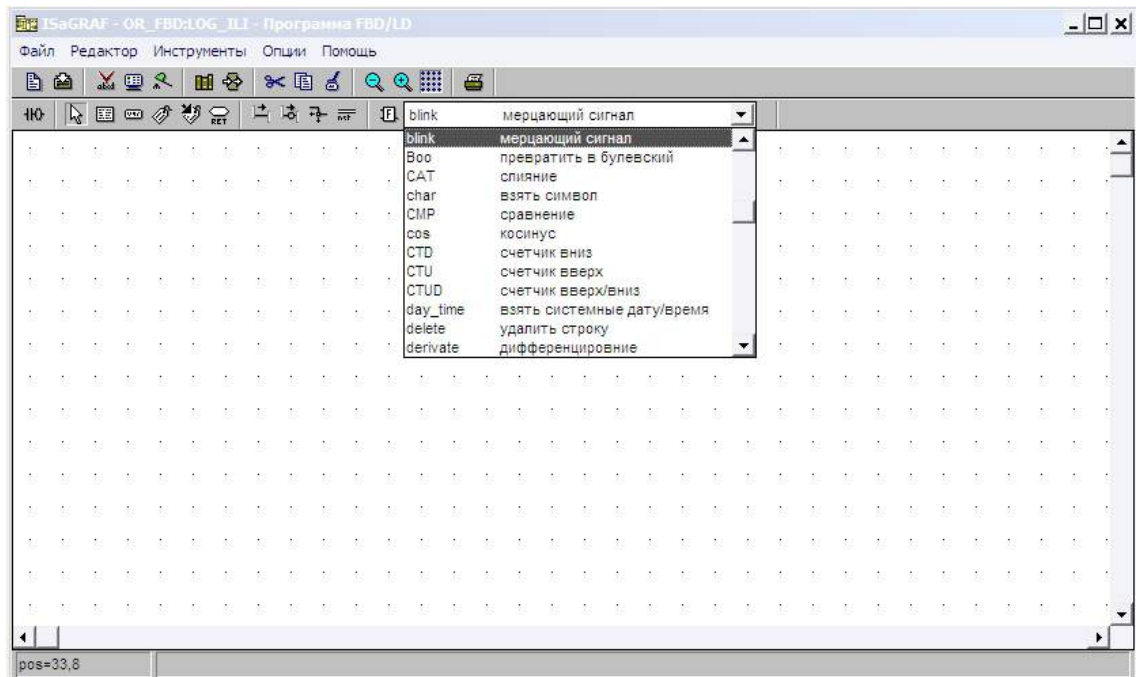


Рисунок 3 Окно программы на языке FBD

Для открытия окна дважды щелкните мышью на блоке. Кнопка Информация позволяет получить больше сведений о функциональном блоке (рисунок 3.7).

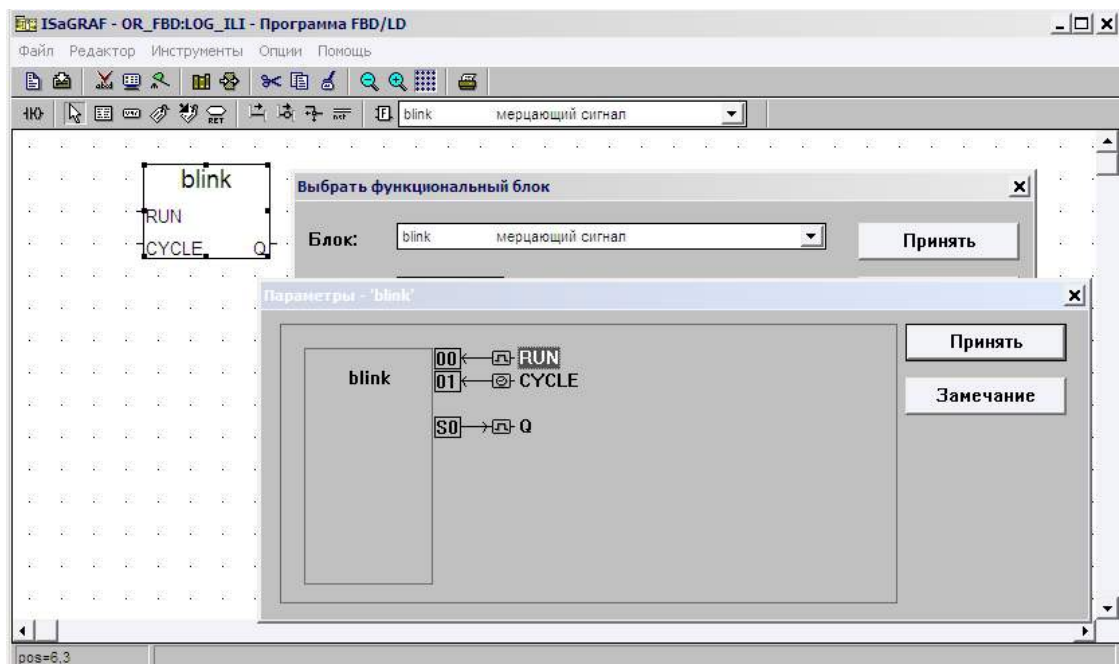
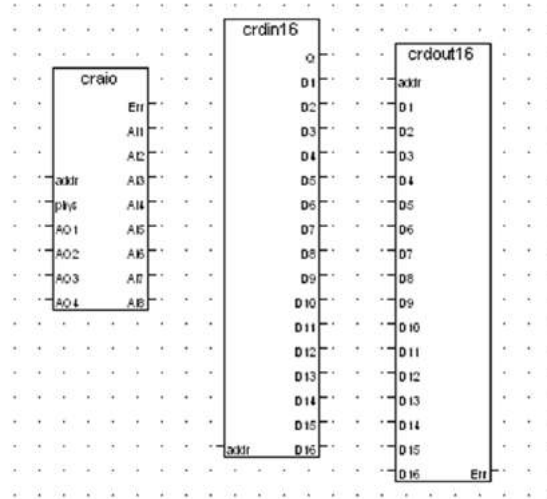


Рисунок 3.7 Параметры блока

Очень важно правильно определить входные и выходные параметры. Кроме того, важно убедиться, что все входные параметры всегда имеют значения перед выполнением блока. Кроме того, мы должны быть осторожны с передачей недопустимых значений, таких как отрицательные целые числа, когда тип

входной переменной установлен на неотрицательное целое число. Это может привести к неожиданным ошибкам и неверным результатам.

Для реализации данной программы понадобятся следующие функциональные блоки:



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

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

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

Аналоговые:

AI4 - сигнал положения исполнительного механизма;

AI5 - сигнал значения уровня в емкости E2;

Дискретные:

DI2 - уровень L3;

DI3 - уровень L2;

DI4- уровень L1;

DO2 - сигнал на включение/ отключение насоса Н3;

DO3 - сигнал «Меньше»;

DO4 - сигнал «Больше»;

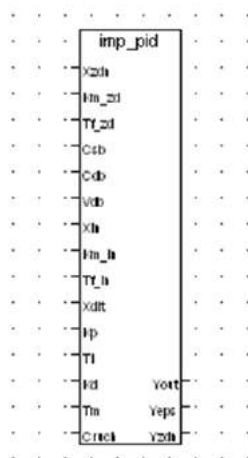
DO5 - сигнализация по та.х;

DO6 - сигнализация по min;

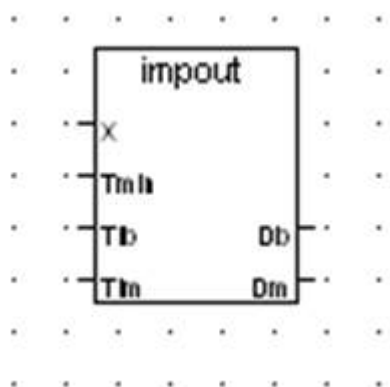
DO7 - звуковая сигнализация;

Регулирование импульсное `imp_pid`

Алгоритм используется для создания ПИД-регулятора с импульсным ВЫХОДОМ.



Impout - формирователь импульсного вывода по заданной скважности.



Входы:

X - скважность;

Tmin - минимальная длительность импульса;

Tlb - время люфта на «больше»;

Tlm - время люфта на «меньше»;

Выходы: Db - дискретный сигнал на «больше»;

Dm - дискретный сигнал на «меньше».

Назначение.

Алгоритм применяется в тех случаях, когда контроллер должен управлять исполнительным механизмом постоянной скорости.

Алгоритм переводит сигнал, который создают алгоритмы контроллера (в том числе, алгоритм импульсного регулирования), в ряд импульсов с переменной

скважностью, и посылает импульсные сигналы на выходы алгоритма Db (выход больше) и Dm (выход меньше).

Описание алгоритма- Алгоритм включает широтно-импульсный модулятор (ШИМ), который переводит входной сигнал X в ряд импульсов с скважностью Q , которая зависит от входного сигнала: $Q=X/100$. При скважности $Q=1$. Если $X>0$, то импульсы появляются в выходной цепи “ больше”, если $X<0$, то в цепи меньше“. При $X=0$ выходной сигнал обнуляется.

Параметр «Т» определяет минимальную длительность выходного импульса.

Параметры T1b и T1t определяют время дополнительного импульса выборки лофта исполнительного механизма в сторону соответственно большего или меньшего изменения его движения. «Т», «T1b» и «T1t» задаются в секундах и округляются до ближайшего значения, кратного времени цикла контроллера».

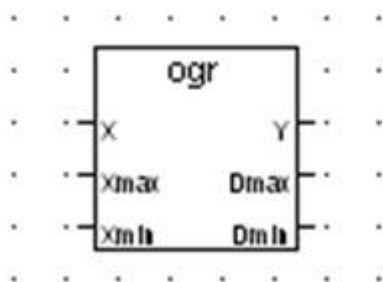
Ограничение ogr

1. Алгоритм используется для установки верхней и (или) нижней границы диапазона сигнала.

2. Алгоритм предназначен для обеспечения работы исполнительного механизма в определенном диапазоне при нормальной работе и для ограничения диапазона сигналов при нештатных состояниях, таких как неисправность или внешнее вмешательство.

3. Верхний и нижний пределы могут быть установлены исходя из характеристик и требований исполнительного механизма и общей системы управления.

4. Два дискретных выхода Dmax и Dmin могут запускать аварийные сигналы или вызывать отключение механизма при достижении пределов, обеспечивая безопасность и надежность системы.



Входы: X ;

Xmax - верхняя граница ограничения;

Xmin - нижняя граница ограничения;

Выходы: Y

Dmax - достижение верхней границы;

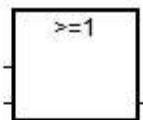
Dmin - достижение нижней границы;

Если $X_{max} > x > X_{min}$, то $Y = X$, $D_{max} = 0$, $D_{min} = 0$;

Если $X = X_{max}$, $D_{max} = 1$, $D_{min} = 0$;

Если $X = X_{min}$, $D_{max} = 0$, $D_{min} = 1$;

Логическое ИЛИ



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

Аргументы:

(inputs)

BOOLEAN

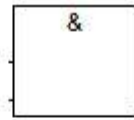
output

BOOLEAN логическое ИЛИ двух или более входов

Описание:

Логическое ИЛИ двух или больше значений.

Логическое И



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

Аргументы:

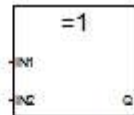
(inputs)
output

BOOLEAN
BOOLEAN логическое И двух или более входов

Описание:

Логическое И двух или больше значений.

Логическое ИЛИ



Аргументы:

IN1
IN2
Q

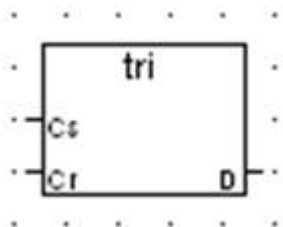
BOOLEAN
BOOLEAN
BOOLEAN логическое ИЛИ двух или более входов

Описание:

Логическое ИЛИ двух или больше значений.

Триггер tri (RS-триггер)

Алгоритм применяется для запоминания дискретных сигналов.



Cs - команда установки

Cr- команда сброса

У - выход

S является «установочным входом», что означает, что если Cr=0 и Cs=1 (что является логическим значением, равным истине), то выходной сигнал D=1. Затем этот выходной сигнал запоминается или удерживается на месте, так что его значение остается равным 1 независимо от любых изменений значения входного сигнала Cs.

Cs Cr D

0 0 D(i-1)


1 0 1

0 1 0

1 1 0

Где D(i-1) предыдущее значение выхода.

3.7 Объявление переменных

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

Затем переходим на вкладку с нужным типом переменной (например Целые] Действительные) и двойным щелчком мыши выбираем строку в поле. В открывшемся окне указываем параметры переменной: имя, комментарий (если требуется), атрибуты и т.д. (рисунок 3,8).

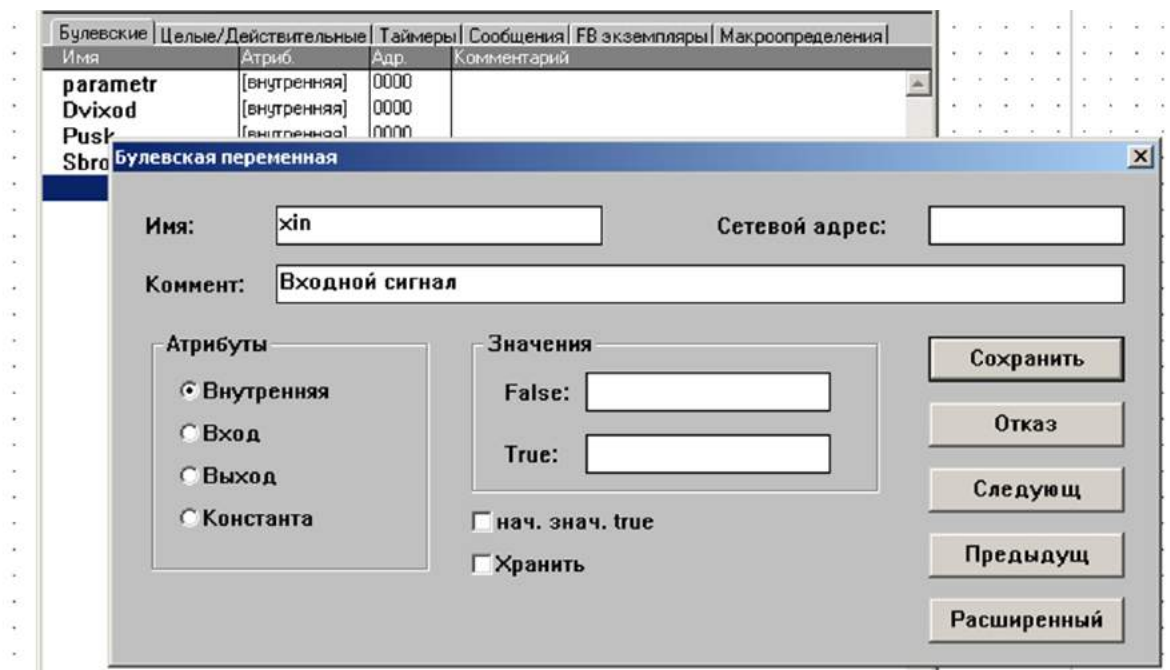


Рисунок 3.8 Окно параметров переменной

После определения переменных закрываем словарь, сохранив все

изменения. В таблице 1 перечислены все переменные, нужные для проекта.

Таблица 1 — Переменные используемые и создании программы

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

Db		Внутренняя	
Dm		Внутренняя	
Dmax		Внутренняя	
Dmin		Внутренняя	
Cruch		Внутренняя	
Rezhim_H3		Внутренняя	
L1		Внутренняя	
L2		Внутренняя	
L3		Внутренняя	
Dtri		Внутренняя	
Ruch_H3		Внутренняя	
Control_H3		Внутренняя	
Dmax1		Внутренняя	
Dmin1		Внутренняя	
ZVUK		Внутренняя	
Db_Ruch		Внутренняя	
Dm_Ruch		Внутренняя	
Sign_b		Внутренняя	
Sign_m		Внутренняя	
rez		Внутренняя	
Q_XOR		Внутренняя	

Когда все переменные определены, нужно поместить их в рабочее пространство программы. для этого на панели управления выбираем значок VAR - Вставить переменную. Выбираем нужную переменную и добавляем её в программу.

Теперь необходимо связать переменные входом (выходом) функционального блока с помощью линий связи, предварительно нажав иконку



- **Начертить линию связи.** Нажав ЛС мыши на переменной и, не отпуская ее, ведем линию до нужного входа функционального блока и отпускаем ЛК. Аналогично связываем другие переменные (рисунок 3.9).

Связать можно также выход одного блока с входом другого.

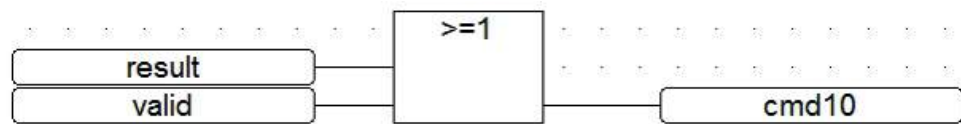


Рисунок 3.9 Функциональный блок с переменными

В процессе создания программы, а также после ее написания, необходимо выполнять проверку её на синтаксис. Для этого нужно зайти в меню **ФАЙЛ/ ПРОВЕРИТЬ**.

4 Примеры создания программ в системе ISaGRAF на языке FBD

4.1 Создание простейшей программы, реализующей операцию логического ИЛИ на языке FBD в режиме симуляции

Чтобы запустить программное обеспечение для разработки системы ISaGRAF, вы можете либо дважды щелкнуть значок ISaGRAF на рабочем столе, либо выбрать меню «Пуск» в Windows (кнопка Windows на клавиатуре), а затем нажать «Программы», чтобы открыть Диспетчер программ. Диспетчер программ отобразит список установленных программ.

После открытия Program Producer найдите программу ISaGRAF в списке и нажмите кнопку «Выполнить», чтобы запустить программное обеспечение для разработки системы ISaGRAF.

Кроме того, вы можете щелкнуть правой кнопкой мыши значок ISaGRAF на рабочем столе и выбрать опцию «Открыть», чтобы запустить программное обеспечение для разработки системы ISaGRAF.

Кроме того, вы можете ввести команду «ISaGRAF» в окне командной строки, чтобы запустить программное обеспечение для разработки системы ISaGRAF. Откройте окно командной строки, нажав клавишу «Windows» + «R» на клавиатуре, введите «cmd» (без кавычек) и нажмите клавишу «Enter». Затем введите «ISaGRAF» (без кавычек) и нажмите «Enter», чтобы запустить программное обеспечение для разработки системы ISaGRAF.

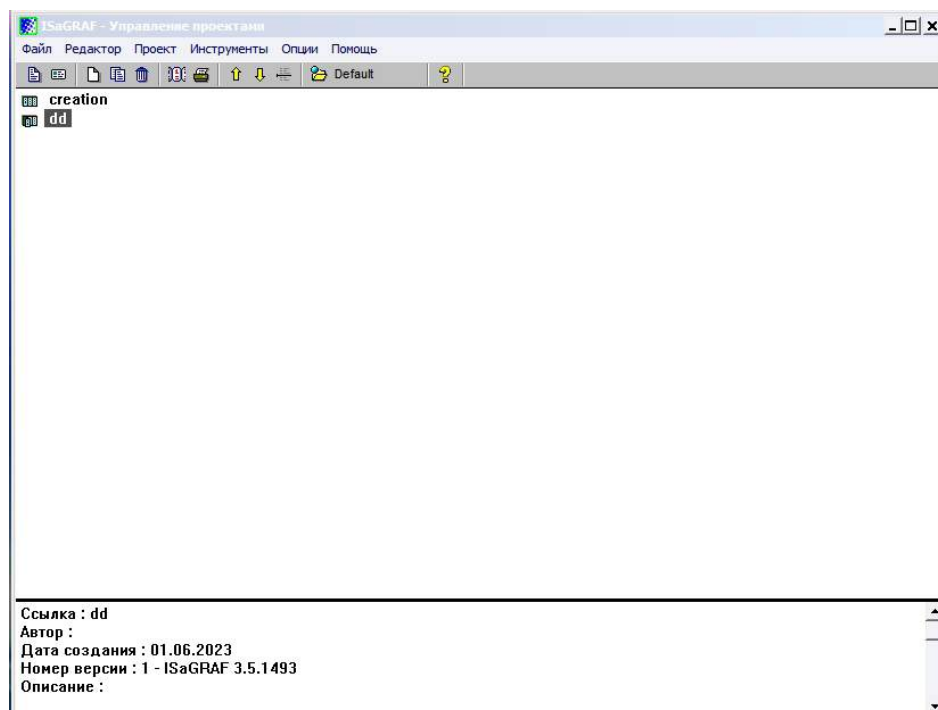


Рисунок 4.1 — Окно менеджера управления проектами ISaGRAF

Чтобы создать новый проект в ISaGRAF, выполните следующие действия:

1. Откройте программное обеспечение ISaGRAF и щелкните меню «Файл» или выберите «Файл» на панели инструментов в верхней части окна.
2. Выберите «Новый» в раскрывающемся меню. Кроме того, вы можете нажать комбинацию клавиш «Ctrl + N».
3. Заполните или выберите необходимую информацию, как показано на рисунке 4.2.

После создания проекта, в котором вы можете приступить к проектированию и моделированию контура управления.

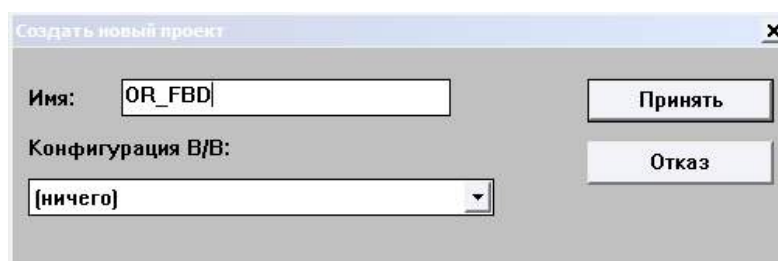


Рисунок 4.2 Создание нового проекта

После того, как вы создадите новый проект в ISaGRAF, он появится в списке проектов в главном окне программы. Вы можете дважды щелкнуть имя

проекта в списке, чтобы открыть окно проекта. Когда вы дважды щелкните имя проекта, программное обеспечение ISaGRAF откроет новое окно с деталями проекта. Это окно будет содержать древовидное представление проекта контура управления, а также холст, на котором вы можете просматривать и редактировать проект контура управления. Кроме того, программное обеспечение ISaGRAF также предоставляет встроенную справочную систему, которая дает подробные объяснения каждого компонента контура управления и функций, которые они выполняют. Доступ к этой встроенной справочной системе можно получить из меню «Справка» или нажав клавишу «F1» на клавиатуре во время работы с ISaGRAF.

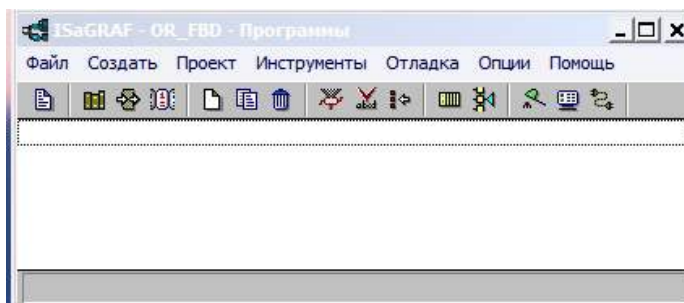


Рисунок 4.3 - Окно Менеджера Программ

Чтобы создать новую программу в ISaGRAF, выполните следующие действия: 1. Откройте программное обеспечение ISaGRAF и щелкните меню «Файл» или выберите «Файл» на панели инструментов в верхней части окна. 2. Выберите «Новый» в раскрывающемся меню. Кроме того, вы можете нажать комбинацию клавиш «Ctrl + N», чтобы создать новую программу. 3. В диалоговом окне «Создать новую программу» введите название вашей программы, например, «LOG ILI» на английском языке, и (необязательно) введите полезный комментарий о назначении программы. 4. Выберите язык FBD (функциональная блок-схема) из списка доступных языков программирования в раскрывающемся списке «язык FBD: Функциональная Блочная Диаграмма». 5. Выберите стиль «программа» из списка доступных стилей в раскрывающемся списке «Стиль». 6. Нажмите кнопку «Принять», чтобы создать новую программу. Новая программа теперь появится в списке программ в окне «Программы». Вы

можете дважды щелкнуть имя программы, чтобы открыть новое окно проектирования контура управления, где вы можете начать проектирование и моделирование контура управления с помощью программного обеспечения ISaGRAF.

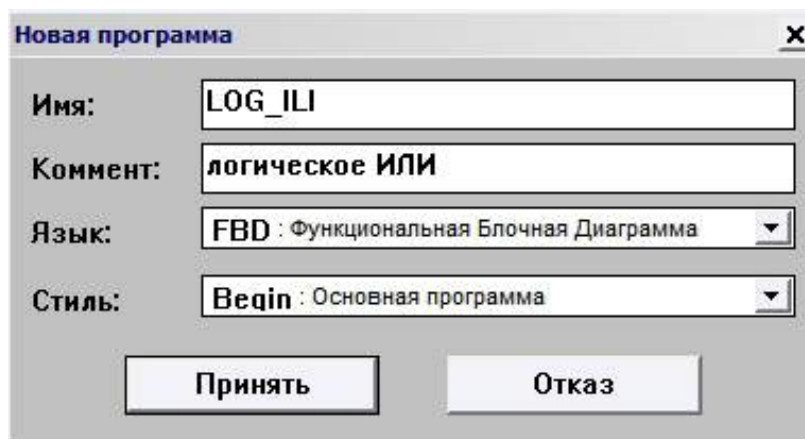


Рисунок 4.4 Диалоговое окно создания новой программы

Откройте программу, дважды щелкнув мышью на названии программы LOG_ILI. Откроется окно редактора FBD/LD, показанное на рисунке 4.5, в котором нужно будет создавать и редактировать программу.

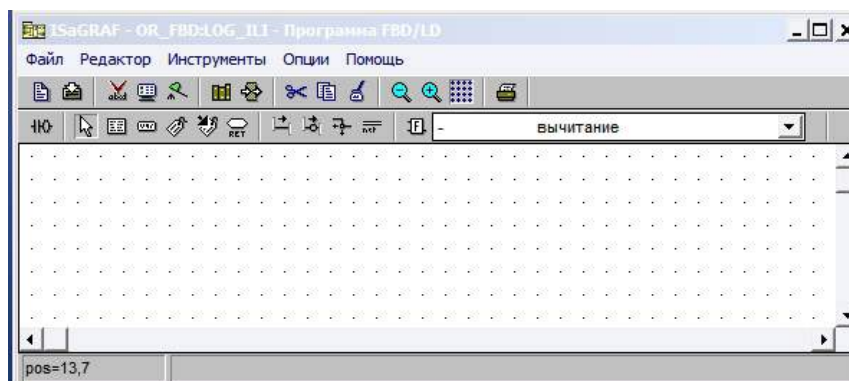


Рисунок 4.5 — Окно редактора FBD/LD

Чтобы открыть словарь ISaGRAF, выполните следующие действия: 1. Откройте программное обеспечение ISaGRAF и щелкните меню «Файл» или выберите «Файл» на панели инструментов в верхней части окна. 2. Выберите «Новый» в раскрывающемся меню. Кроме того, вы можете нажать комбинацию клавиш «Ctrl + N», чтобы открыть новый словарь. 3. В окне «Словарь» вы

увидите список всех доступных символов (переменных, констант и т. д.) и их значений в программе ISaGRAF. 4. Словарь также позволяет выполнять поиск определенных символов, а также просматривать соответствующие описания и примеры их использования в сценарии проектирования контура управления.

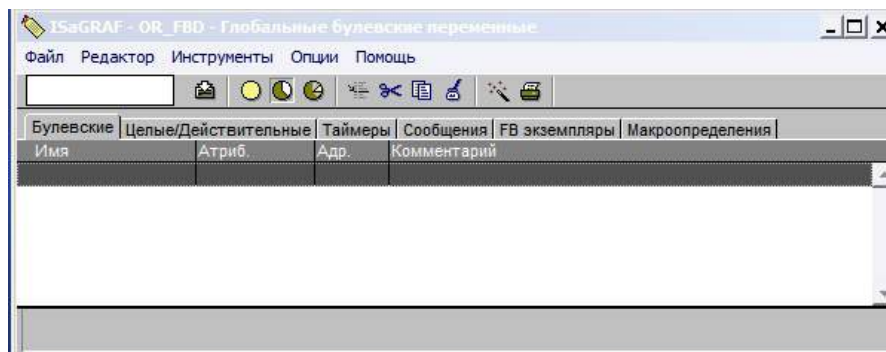


Рисунок 4.6 Окно словаря

Чтобы создать новую логическую переменную в словаре ISaGRAF, выполните следующие действия: 1. Откройте программное обеспечение ISaGRAF и щелкните меню «Файл» или выберите «Файл» на панели инструментов в верхней части окна. 2. Выберите «Новый» в раскрывающемся меню. Кроме того, вы можете нажать комбинацию клавиш «Ctrl + N». 3. В окне «Словарь» выберите вкладку «Логические значения» в левой части окна. 4. В разделе окна «Рабочая область» нажмите кнопку «Вставить новый объект». Кроме того, вы можете щелкнуть в выделенной области раздела «Рабочая область» и дважды щелкнуть. 5. В появившемся диалоговом окне «Новый объект» введите имя новой логической переменной, например, «input1». Вы также можете предоставить полезный комментарий о назначении переменной, но это не обязательно. 6. В разделе «Свойства» диалогового окна выберите «Внутренние» из списка доступных значений «Домен», чтобы создать внутреннюю переменную. 7. В разделе «Значение по умолчанию» диалогового окна можно оставить значение по умолчанию «0» или ввести другое значение. 8. После того, как вы ввели всю необходимую информацию в диалоговое окно, нажмите кнопку «Сохранить», чтобы сохранить новую логическую переменную в свой словарь.

The screenshot shows a window titled "Булевская переменная" (Boolean variable) with a close button (X) in the top right corner. The window contains the following fields and controls:

- Имя:** input1
- Сетевой адрес:** (empty field)
- Коммент:** вход №1
- Атрибуты:**
 - Внутренняя
 - Вход
 - Выход
 - Константа
- Значения:**
 - False: (empty field)
 - True: (empty field)
 - нач. знач. true
 - Хранить
- Buttons:** Сохранить, Отказ, Следующ, Предыдуц, and Расширенный.

Рисунок 4.7 Окно задания параметров переменной input1

Точно так же снова другая выходная переменная и выходная переменная.

The screenshot shows a window titled "Булевская переменная" (Boolean variable) with a close button (X) in the top right corner. The window contains the following fields and controls:

- Имя:** input2
- Сетевой адрес:** (empty field)
- Коммент:** вход №2
- Атрибуты:**
 - Внутренняя
 - Вход
 - Выход
 - Константа
- Значения:**
 - False: (empty field)
 - True: (empty field)
 - нач. знач. true
 - Хранить
- Buttons:** Сохранить, Отказ, Следующ, Предыдуц, and Расширенный.

Рисунок 4.8 - Окно задания параметров переменной input2

The screenshot shows a window titled "Булевская переменная" (Boolean variable) with a close button (X) in the top right corner. The window contains the following fields and controls:

- Имя:** output
- Сетевой адрес:** (empty field)
- Коммент:** выход
- Атрибуты:**
 - Внутренняя
 - Вход
 - Выход
 - Константа
- Значения:**
 - False: (empty field)
 - True: (empty field)
 - нач. знач. true
 - Хранить
- Buttons:** Сохранить, Отказ, Следующ, Предыдуц, and Расширенный.

Рисунок 4.9 - Окно задания параметров переменной output

Теперь вернитесь к интерфейсу, показанному на рис. 4.10, чтобы настроить функциональный блок.

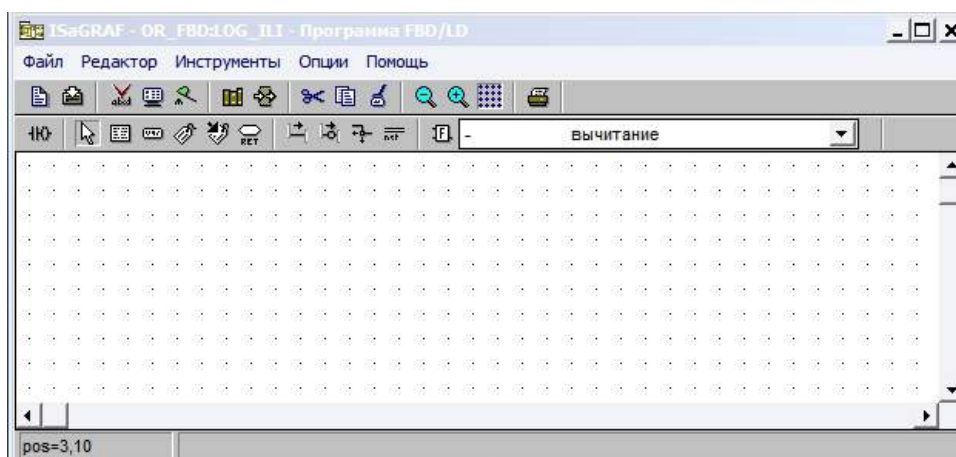


Рисунок 4.10 Редактор FBD/LD

Выберите функциональный блок или, как показано на рисунке 4.11.

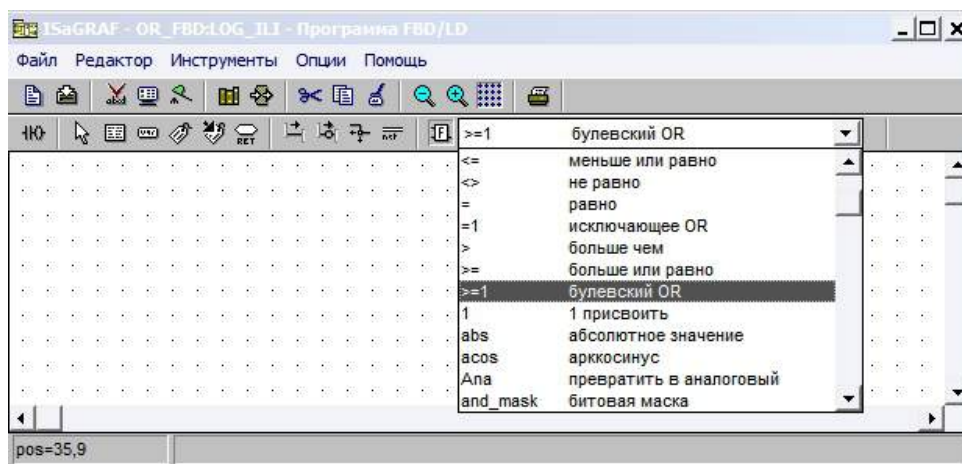


Рисунок 4.11 – Выбор блока булевский OR

Нажмите мышью внутри белой рабочей зоны редактора, и в ней появится изображение функционального блока булевский OR. Если после этого дважды щелкнуть мышью на изображении блока, то откроется окно, где можно изменить свойства блока.

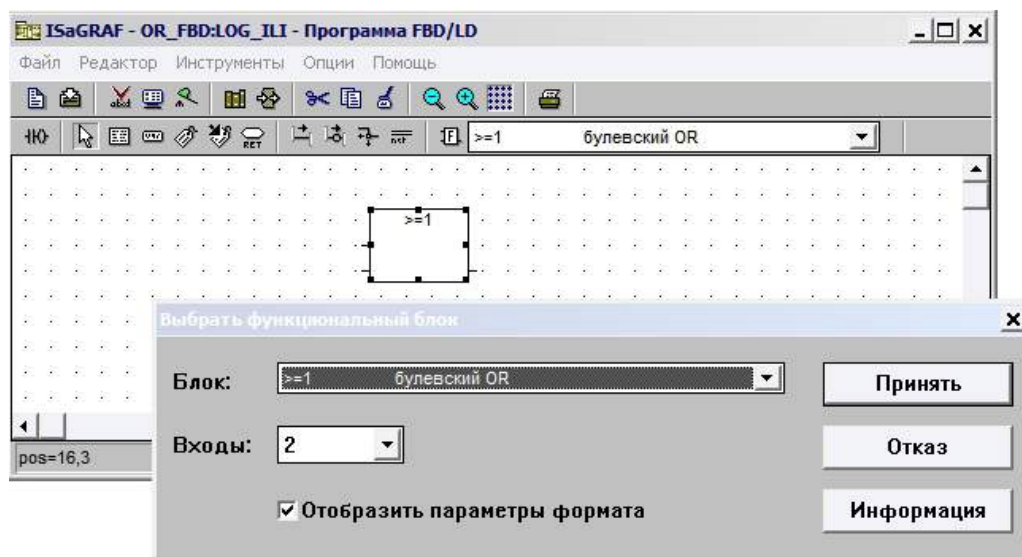


Рисунок 4.12 Окно свойств блока булевский OR

Если в окне свойств блока, показанном на рисунке 4.12, нажать кнопку Информация, то откроется информационное окно с справочной информацией о параметрах блока (рисунок 4.13).

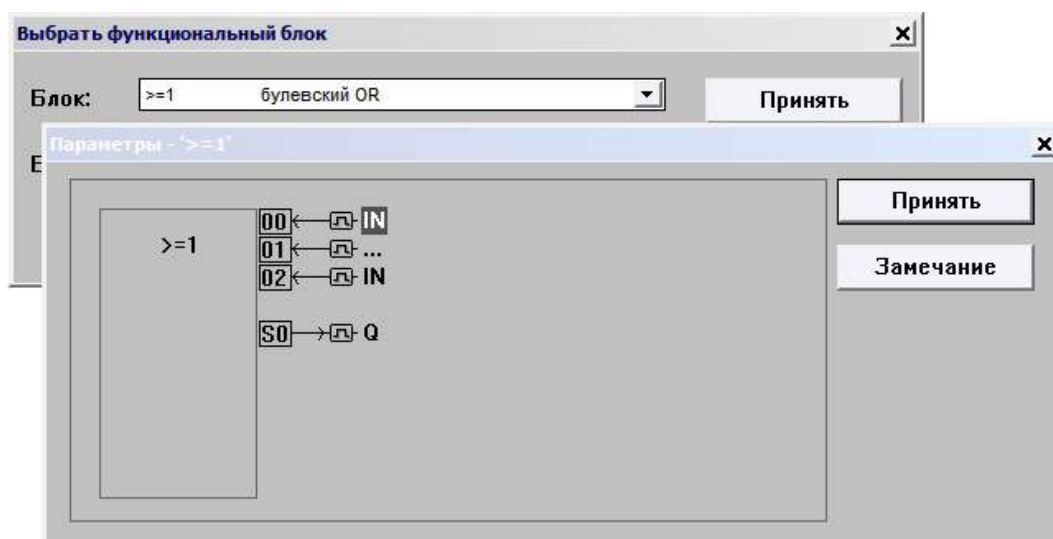


Рисунок 4.13 Окно с информацией о параметрах блока

Для получения дополнительной информации о функциональном блоке нажмите кнопку Замечание в окне, показанном на рисунке 4.13, после чего откроется окно подробного описания блока (Рисунок 4.14).

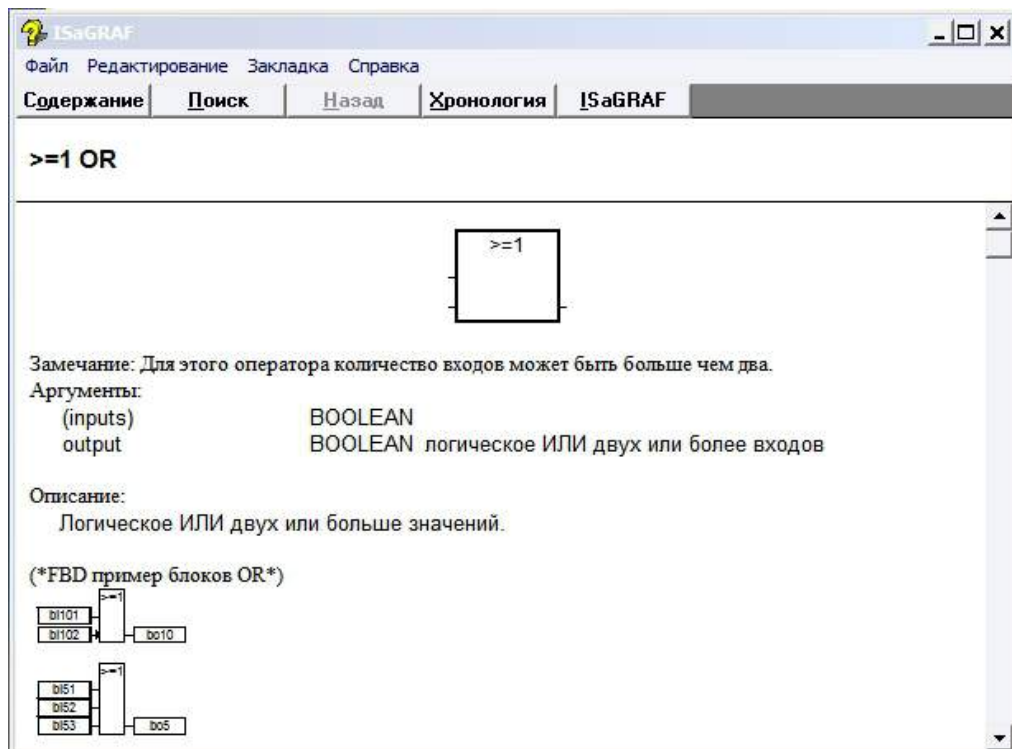


Рисунок 4.14 Окно подробного описания функционального блока

Теперь входы и выход блока булевский OR нужно связать с переменными, которые вы ранее уже определили в словаре. Выберите на панели управления значок Вставить переменную, нажмите мышью в свободном месте рабочей зоны редактора.

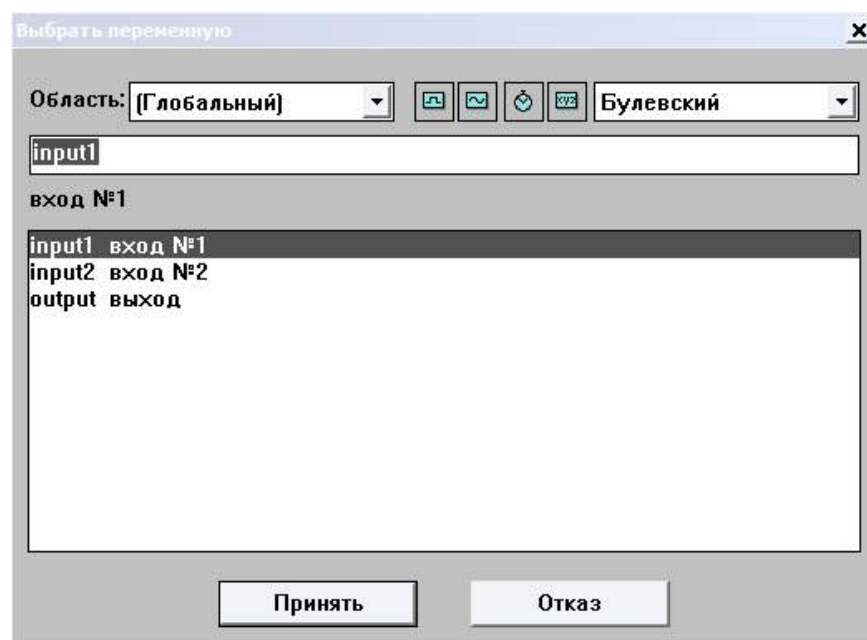




Рисунок 4.15- Окно выбора переменной

Затем таким же образом вытащите две другие переменные.

После этого нажмите на панели управления редактора значок  Выбрать, нажмите мышью на изображение переменной input1 в рабочей зоне и не отпуская мышью аккуратно переместите переменную, поставив ее перед верхним входом блока булевский OR. Таким же образом расположите остальные переменные рядом с соответствующими входами и выходом функционального блока.

Чтобы соединить входные и выходные переменные проекта контура управления, выполните следующие действия: 1. Откройте проект контура управления в программном обеспечении ISaGRAF и выберите вкладку «Диаграмма» в строке меню в верхней части окна. 2. В окне проектирования контура управления выберите инструмент «Соединения»  в меню слева. 3. Щелкните и перетащите мышью от верхнего входа логического функционального блока OR к метке «input1» в окне проектирования контура управления. Отпустите кнопку мыши, чтобы нарисовать линию соединения. 4. Повторите этот шаг, чтобы соединить другие входные и выходные переменные проекта контура управления с соответствующими входами и выходами различных функциональных блоков в проекте. 5. Когда все соединения будут нарисованы, нажмите кнопку «ОК», чтобы сохранить проект контура управления.

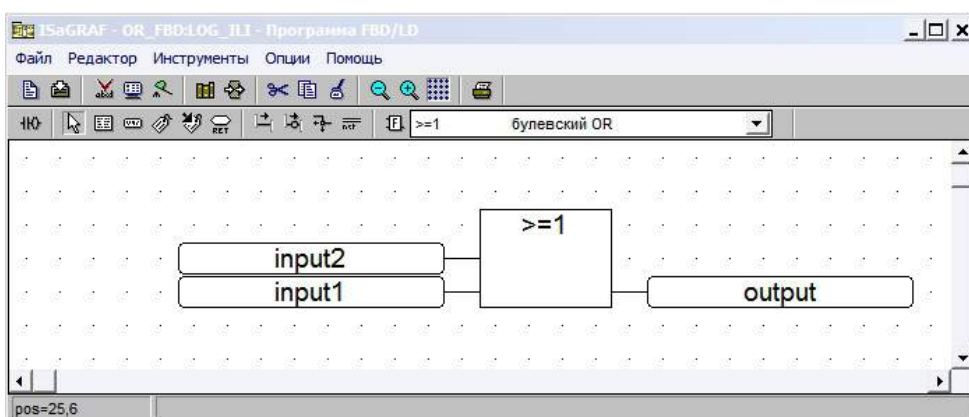


Рисунок 4.16 - Полностью введенная программа логического ИЛИ

Чтобы проверить синтаксис проекта контура управления, созданного в программном обеспечении ISaGRAF, выполните следующие действия: 1. Откройте программное обеспечение ISaGRAF и выберите вкладку «Диаграмма»

в строке меню в верхней части окна. 2. Выберите команду "Проверить" из меню "Файл" или нажав комбинацию клавиш "Ctrl+Alt+V". 3. В появившемся диалоговом окне «Проверить» установите флажок «Проверить синтаксис». 4. Если в синтаксисе проекта контура управления нет ошибок, вы можете нажать кнопку «ОК», чтобы закрыть диалоговое окно и сохранить проект. 5. Для сохранения проекта контура управления в виде нового файла выберите команду «Сохранить» из меню «Файл» или нажав комбинацию клавиш «Ctrl+S». В появившемся диалоговом окне «Сохранить как» введите имя нового файла и выберите формат файла. 6. Нажмите кнопку «Сохранить», чтобы сохранить проект контура управления.

Перед запуском симулятора закройте окно редактора FBD/LD, иначе ISaGRAF может «зависнуть». Чтобы сгенерировать код, выберите создать/Создать приложение в меню или нажмите кнопку Создать код приложения в Диспетчере Программ. Затем запустите программу в режиме симуляции, выбрав Отладка/Симуляция в меню или нажав кнопку Симуляция.

Открыть редактор и посмотреть работу программы в симуляторе можно, щелкнув мышью дважды по имени программы LOG ILI в окне Отладка программ (рисунок 4.17).

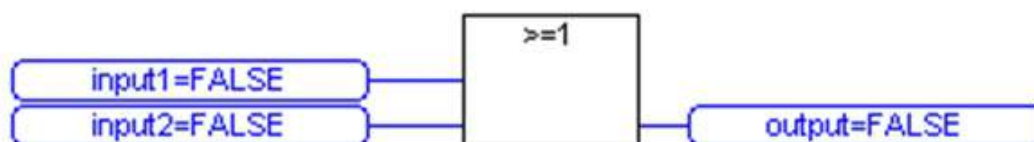


Рисунок 4.17 - Окно программы в режиме симуляции

Двойной щелчок мыши по изображению входной переменной позволит изменить ее значение на TRUE или FALSE во время работы программы. Окно выбора значения показано на рисунке 4.18.

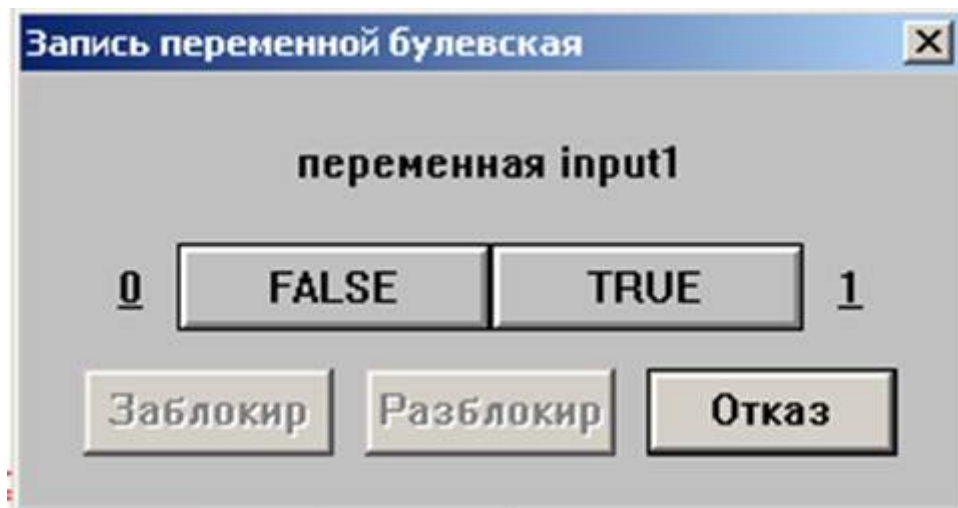


Рисунок 4.18 - Окно изменения значения переменной

Программа выполняет операцию логического ИЛИ, что можно увидеть, экспериментируя со значениями переменных `input1` и `input2`. Значение `output` становится `TRUE`, когда `TRUE` равна хотя бы одна из переменных `input1` или `input2`. На рисунке 4.19 показан пример работы программы в симуляторе при `input1=TRUE`.

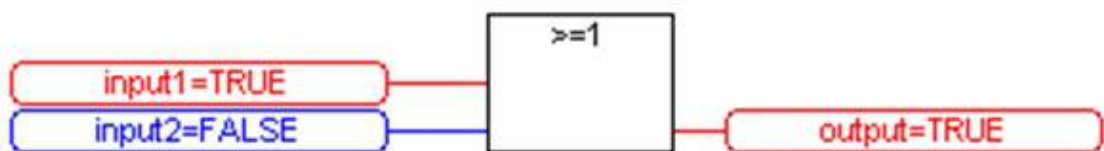


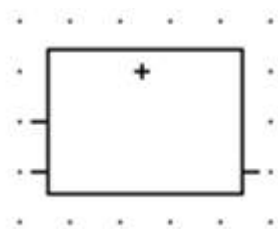
Рисунок 4.19 - Работа программы в режиме симуляции при значении `input1=TRUE`

4.2 Создание элементарной программы аналогового ввода/вывода

Следуйте действиям из п.4.1 для создания нового проекта и новой программы.

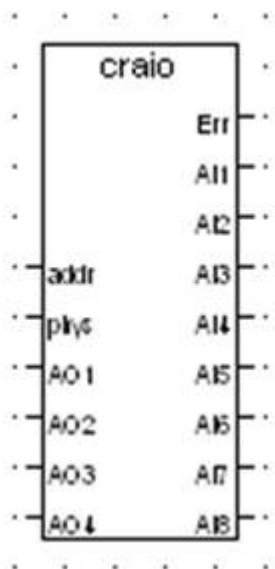
Для реализации алгоритма сложения аналогового сигнала с константой нам необходимы следующие функциональные блоки:

1. Сложение



Описание: Складывает две или несколько переменных

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



После того, как функциональные блоки будут на рабочей области, необходимо задать переменные (таблица 2).

Таблица 2

Имя переменной	Тип данных	Атрибуты	Формат	Начальное значение
Vход1	Целые действительные	Внутренняя	Вещественная	0
vixod		Внутренняя	Вещественная	0
konstanta		Константа	Вещественная	2
address		Константа	Целая	40
paramctr	Булевские	Внутренняя		TRUE

Задав переменные, помещаем их на рабочую область и соединяем с

соответствующими входами-выходами функциональных блоков (рис. 18).

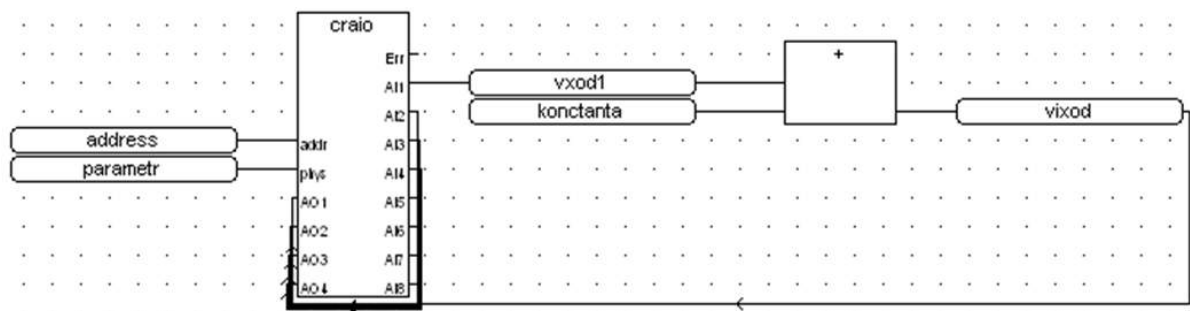


Рисунок 4.21 - Сложение аналогового сигнала с константой

Для взаимодействия целевой задачи ISaGRAF с контроллером КРОСС необходимо установить опции компилятора. Для этого выбираем в меню ОПЦИИ КОМПИЛЯТОРА (рисунок 4.22) и отмечаем TIC code for Motorola, так как процессор контроллера КРОСС от фирмы Motorola.

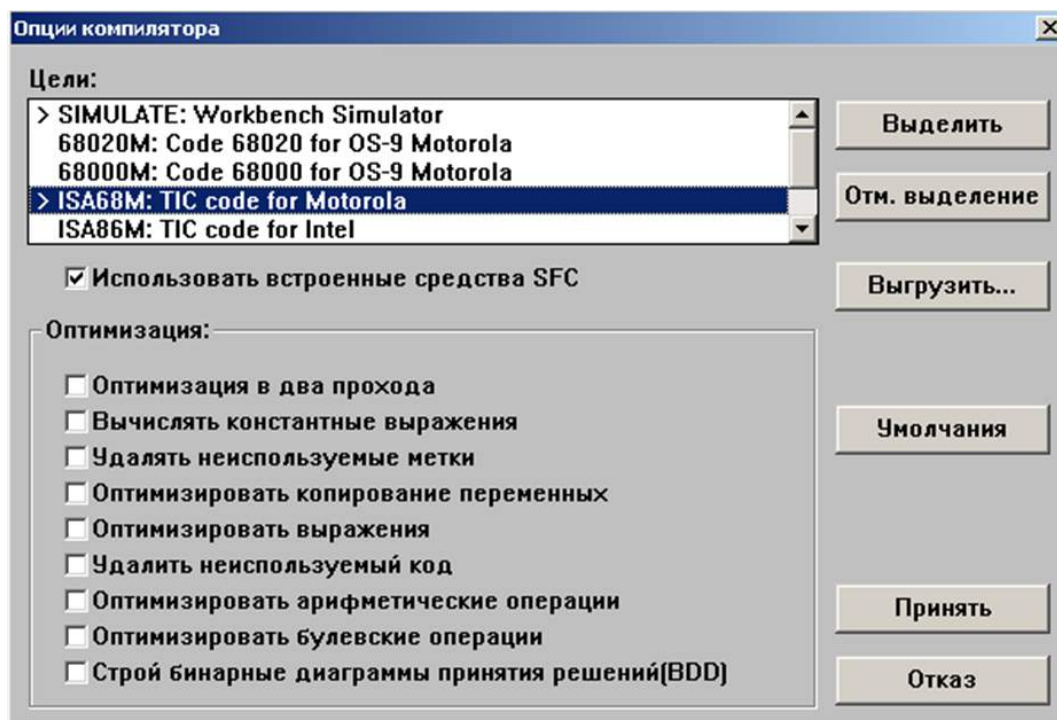


Рисунок 4.22 - Опции компилятора

Чтобы сгенерировать код для приложения и проверить программу на наличие синтаксических ошибок в программном обеспечении ISaGRAF, выполните следующие действия: 1. Откройте программное обеспечение ISaGRAF и выберите вкладку «Программа» в строке меню в верхней части окна. 2. Выберите команду «Создать код приложения» из меню «Программы» или

нажав комбинацию клавиш «Ctrl+Alt+Q». 3. В появившемся диалоговом окне «Мастер кода» следуйте инструкциям по созданию новой программы IES 61131-3. Это создаст набор файлов, содержащих исходный код для вашего проекта контура управления. 4. После того, как вы сгенерировали исходный код для проектирования контура управления, выберите команду «Проверить код» из меню «Программа» или нажав комбинацию клавиш «Ctrl + Alt + V». 5. В появившемся диалоговом окне «Подтвердить код» установите флажок «Проверить синтаксис». 6. Если ошибок в синтаксисе кода нет, можно закрыть диалоговое окно «Проверить код» и сохранить программу, выбрав команду «Сохранить» из меню «Файл» или нажав сочетание клавиш «Ctrl+S». комбинация клавиш. 7. Чтобы подключить ПК к контроллеру с помощью связи Ethernet, выполните следующие действия:

1. Откройте программное обеспечение ISaGRAF и выберите вкладку «Программа» в строке меню в верхней части окна.

2. Выберите команду "Отладка" из меню "Программы" или нажав комбинацию клавиш "Ctrl+Alt+E".

3. В появившемся диалоговом окне «Отладка» выберите вкладку «Связь».

4. Во вкладке «Связь» выберите вкладку «TCP/IP».

5. На вкладке «TCP/IP» введите IP-адрес коммуникационного порта как «192.168.0.2» и введите номер коммуникационного порта как «1100».

6. Нажмите кнопку «Подключить», чтобы установить соединение между ПК и контроллером.

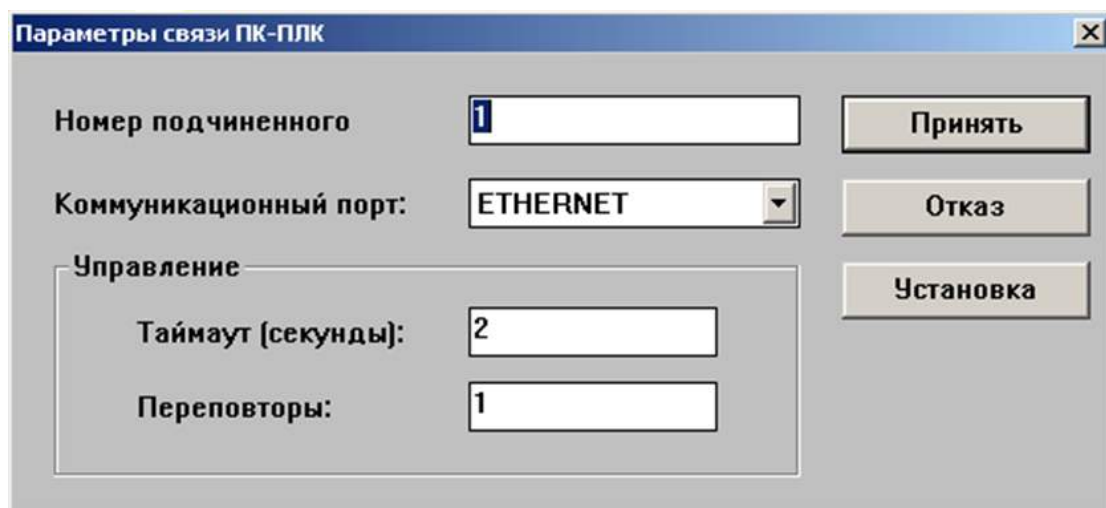


Рисунок 4.23 - Параметры соединения с контроллером

Подаем аналоговый сигнал с АВК-6 на первый вход модуля АЮ1-8/4 и наблюдаем выходной сигнал на первом выходе модуля для отладки программы контроллером. Для этого соединяем первый вход модуля с источником напряжения и устанавливаем напряжение.

Соединяем первый выход модуля с потенциометром. Включаем контроллер на лабораторном стенде автоматическим выключателем и ожидаем сообщения о подключении по локальной сети на мониторе.

Открываем окно ISAGRAF-Программы FBD/LD и запускаем отладчик из меню ФАЙЛ/ОТЛАДКА или с панели управления иконкой. На экране монитора будет окно отладчика (рисунок 4.24).

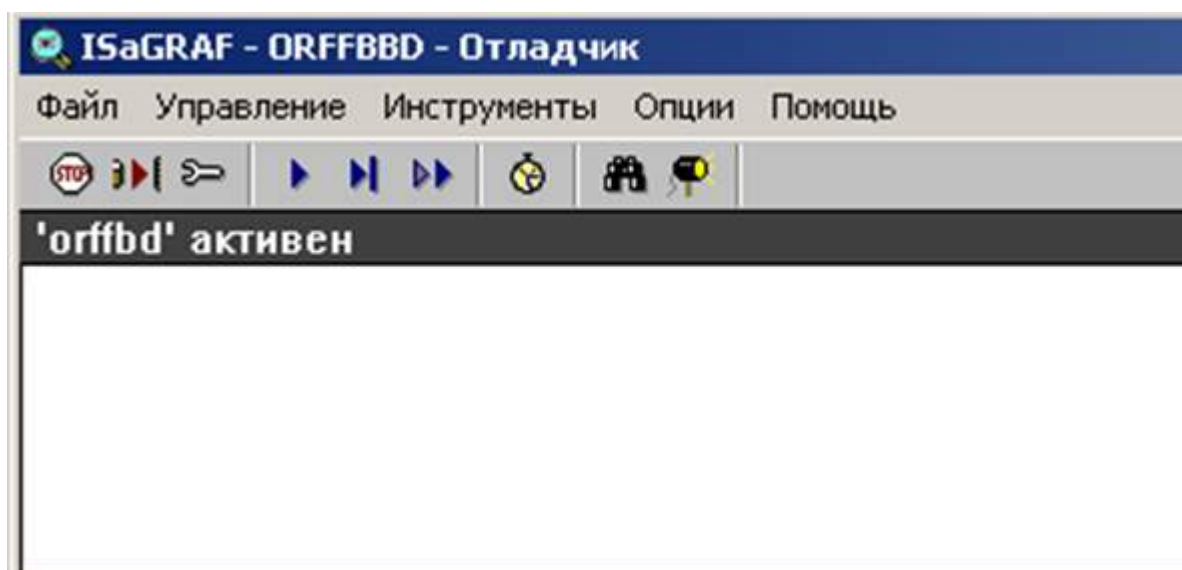


Рисунок 4.24 -Окно отладчика

Остановить приложение, которое было в контроллере раньше, можно командой **ФАЙЛ ОСТАНОВИТЬ ПРИЛОЖЕНИЕ** из меню отладчика или иконкой **STOP**. После остановки приложения загружаем свое приложение командой **ФАЙЛ ЗАГРУЗИТЬ**. После этого контроллер запускает нашу программу сложения аналогового сигнала с константой (рисунок 4.25).

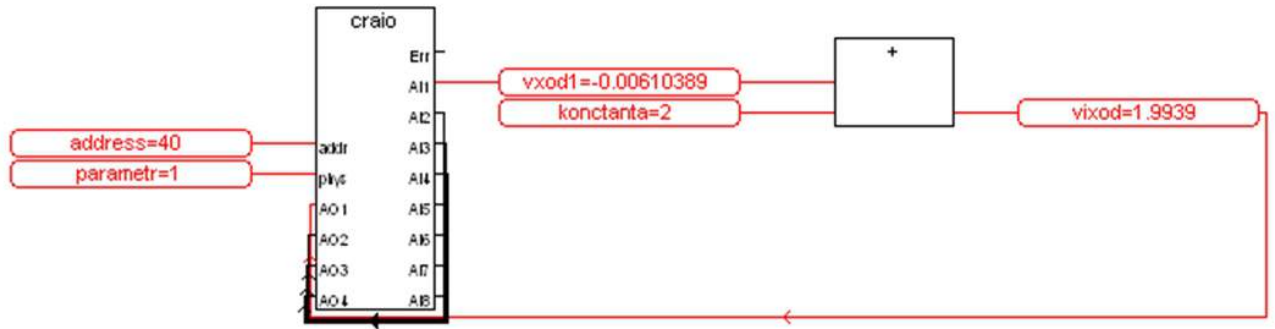


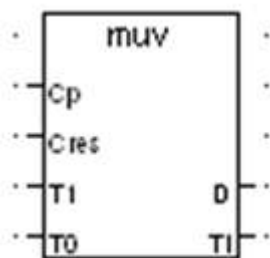
Рисунок 4.25 - Работа программы сложения аналогового сигнала с константой в режиме отладки

4.3 Создание элементарной программы дискретного ввода/вывода

Выполняя последовательно действия, приведенные в п.4.1 по созданию проекта и программы, создадим новый проект, а в нем новую программу.

Для реализации данной программы нам необходимы следующие функциональные блоки:

1. Мультивибратор



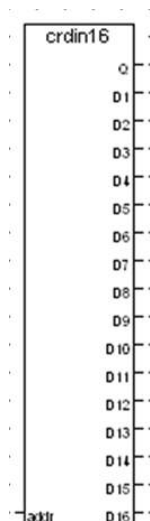
Состояние основного выхода **D** алгоритма в режиме пуска зависит от значений **T1** и **T0**:

Если $T1 > 0$ и $T0 > 0$, то формируется последовательность импульсов.

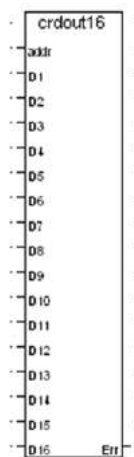
Если $T1 \leq 0$ и $T0$ -безразлично, на выходе формируется 0.

Если $T1 > 0$ и $T0 \leq 0$ на выходе формируется 1.

2. Ввод дискретных сигналов с модуля УСО КРОСС (crdin16)



3. Вывод дискретных сигналов с модуля УСО КРОСС



После помещения функциональных блоков на рабочую область нам необходимо задать переменные (таблица 3).

Таблица 3

Имя переменной	Тип данных	Атрибуты	Формат	Начальное значение
Tpause	Целые действительные	Внутренняя	Вещественная	1
Timpulsa		Внутренняя	Вещественная	1
address2		Константа	Целая	30
Address1		Константа	Целая	30
Tvixoda		Внутренняя	Вещественная	0

Dvixod	Булевские	Внутренняя		FALSE
Pusk		Внутренняя		
Sbros		Внутренняя		

Задав переменные, помещаем их на рабочую область и соединяем с соответствующими входами-выходами функциональных блоков (рисунок 4.31).

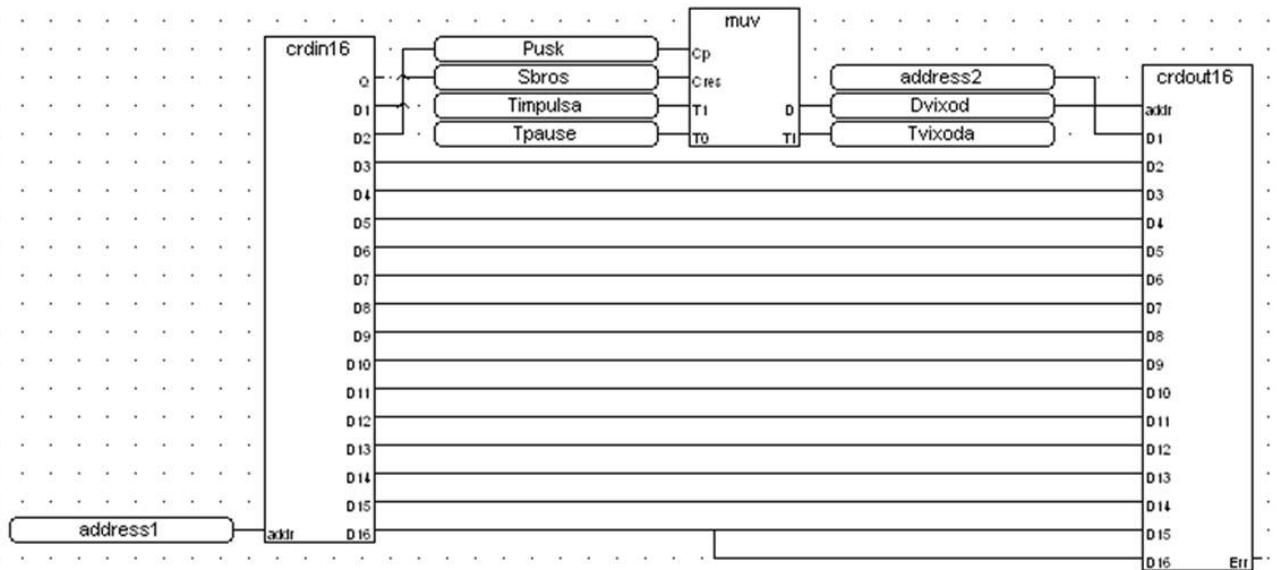


Рисунок 4.31 Реализация мигания светодиода

После этого наша программа реализации мигания светодиода в режиме отладки примет вид (рисунок 4.32):

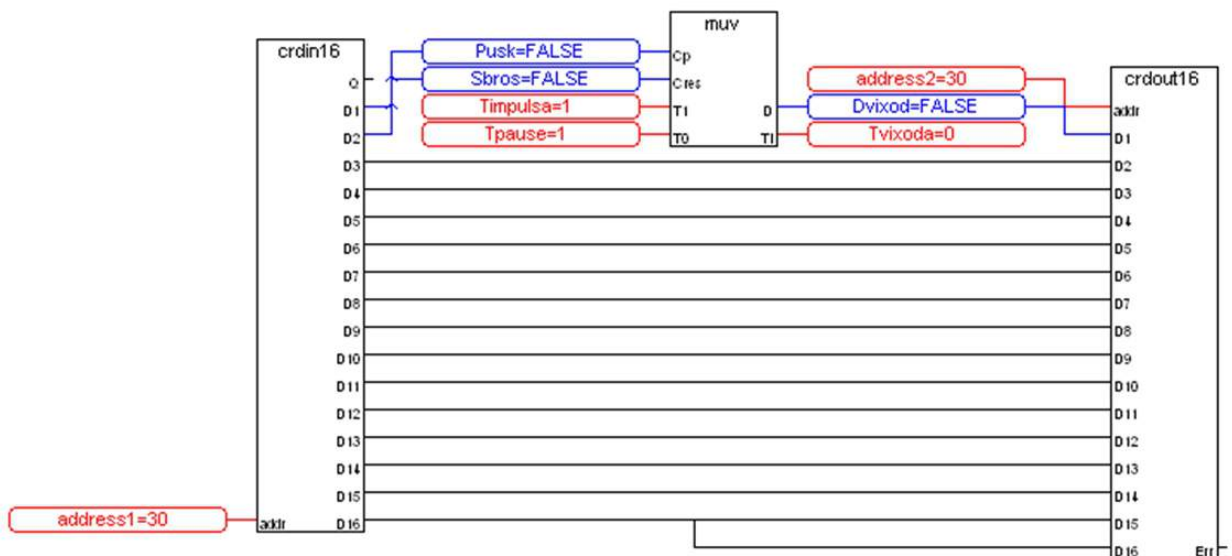


Рисунок 4.32 Состояние программы в режиме отладки

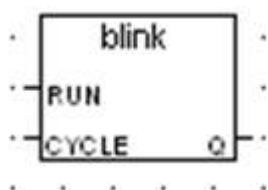
Чтобы заставить мигать светодиод, подключенный к первому выходу дискретного модуля D101-8/8, нужно удерживать нажатой кнопку, подключенную к первому входу модуля. Когда вы нажмете кнопку, значение входа Push станет TRUE и светодиод будет мигать с периодом I секунды. Когда вы отпустите кнопку, значение входа Push вернется к FALSE. Можно остановить мигание, подав на второй вход дискретного модуля лог. 1. Мультивибратор остановится, и светодиод не будет мигать.

4.4 Создание элементарной программы аналогового и дискретного ввода/вывода

Затем следуя инструкциям из п.4.1, создайте новый проект и программу в нем.

Для реализации этой программы вам понадобятся следующие функциональные блоки:

1. Ввод-вывод аналоговых сигналов с модуля УСО КРОСС (craio);
2. Ввод дискретных сигналов с модуля УСО КРОСС (crdin16);
3. Вывод дискретных сигналов с модуля УСО КРОСС (crdout16);
4. Мерцающий сигнал (blink):

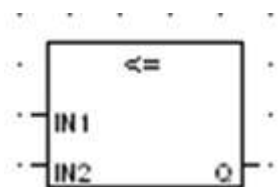


Описание: генерирует мигающий сигнал;

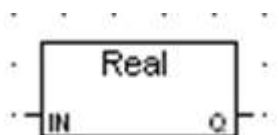
Аргументы:

RUN режим: TRUE=мигать; FALSE=сброс выхода в FALSE (BOOLEAN);
CYCLE — период мигания (TMR); Q — выходной мигающий сигнал (BOOLEAN).

5. Меньше или равно:



6. Превратить в действительный (REAL):



Описание: превращает переменную в действительную;

Аргументы:

IN - любая недействительная аналоговая величина (BOOLEAN,INT,TMR);

Q - число эквивалентное целому аналоговому (INT).

После помещения функциональных блоков на рабочую область нам необходимо задать переменные (таблица 3).

Таблица 3

Имя переменной	Тип данных	Атрибуты	Формат	Начальное значение
Vход1	Целые действительные	Внутренняя	Вещественная	0
address		Константа	Целая	40
address2		Константа	Целая	30
address 1		Константа	Целая	30
konstsnta		Константа	Вещественная	5
Vход	Булевские	Внутренняя		FALSE
VbIход		Внутренняя		
arametr		Внутренняя		TRUE
sравnenie		Внутренняя		FALSE
Blink_period	Таймеры	Внутренняя		t#1s

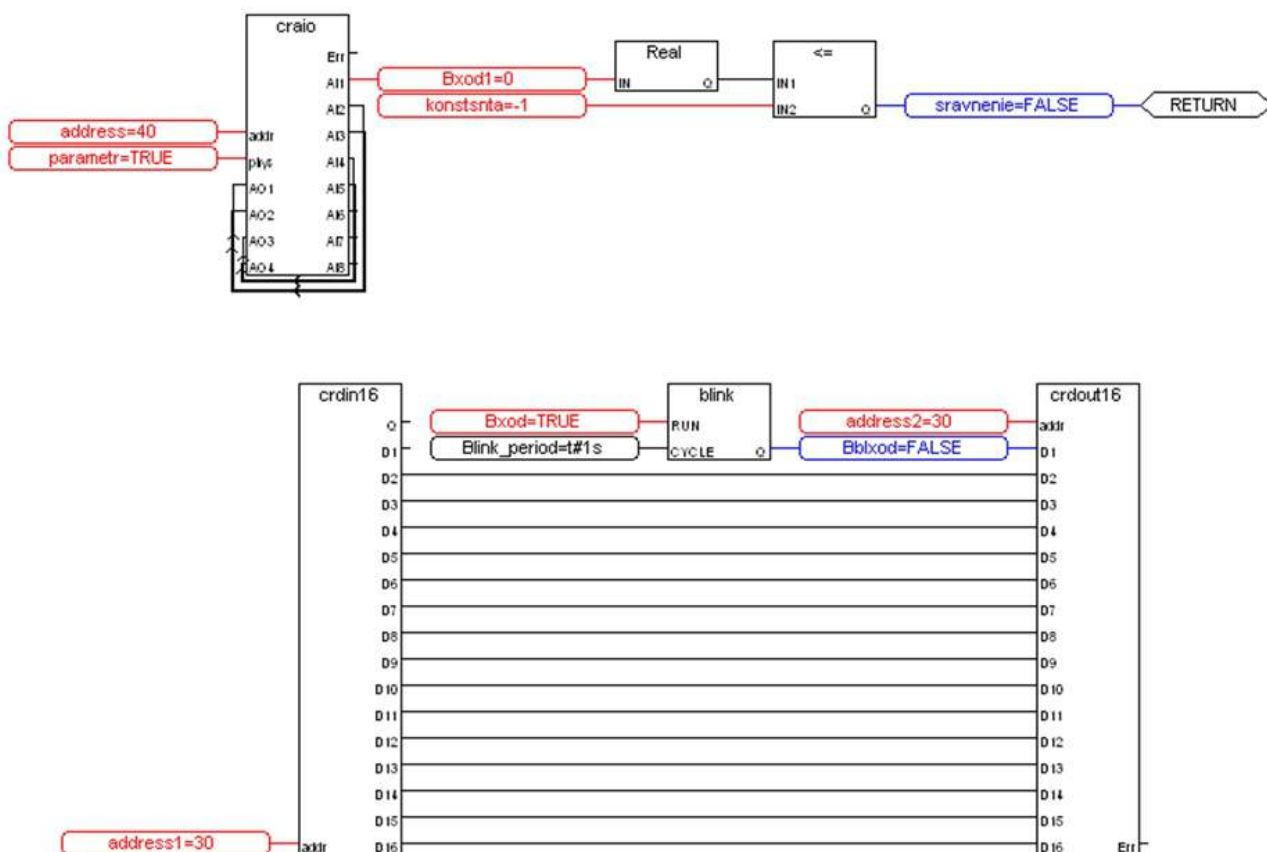


Рисунок 4.42 Состояние программы в режиме отладки

4.5 Создание программы, реализующей ПИД-регулятор

Следуя инструкциям из п.4.1, создайте новый проект и программу в нем.

Для реализации этой программы вам понадобятся следующие функциональные блоки:

1. Ввод-вывод аналоговых сигналов с модуля УС О КРОСС (craio);
2. Ввод дискретных сигналов с модуля УСО КРОСС (crdin16);
3. Вывод дискретных сигналов с модуля УСО КРОСС (crdout16);
4. Мерцающий сигнал (blink);
5. Ограничение (OGR);
6. Логическое OR (булевский OR - &);
7. Регулирование аналоговое (RAN).

Входы-выходы алгоритма RAN (таблица 4) и его функциональная схема (Рисунок 4.51) приведены ниже.

Таблица 4 - Входы-выходы алгоритма RAN

Номер	Тип	Обозначение	Вх-Вых	Назначение
-------	-----	-------------	--------	------------

01	REAL	Xzdn	Вход	Вход задания
02	BOO	Csb	Вход	Включение статической балансировки
03	BOO	Cdb	Вход	Включение динамической балансировки
04	REAL	Vdb	Вход	Скорость динамической балансировки
05	REAL	Xin	Вход	Вход параметра
06	REAL	Xdlt	Вход	Зона нечувствительности
07	REAL	Kp	Вход	Коэффициент пропорциональности
08	REAL	Ti	Вход	Постоянная времени интегрирования
09	REAL	Kd	Вход	Постоянная времени дифференцирования
10	REAL	Xmax	Вход	Уровень ограничения по максимуму
11	REAL	Xmin	Вход	Уровень ограничения по минимуму
12	BOO	Czb	Вход	Сигнал запрета в направлении «Больше»
13	BOO	Czm	Вход	Сигнал запрета в направлении «Меньше»
14	BOO	Yzdn	Вход	Включение ручного режима
15	REAL	Cruch	Вход	Сигнал ручного задания
01	REAL	Y	Выход	Основной выход алгоритма (каскадный)
02	REAL	Yeps	Выход	Сигнал рассогласования
03	REAL	Yzdn	Выход	Сигнал текущего задания

04	BOO	Dmax	Выход	Ограничение по максимуму
05	BOO	Dmin	Выход	Ограничение по минимуму

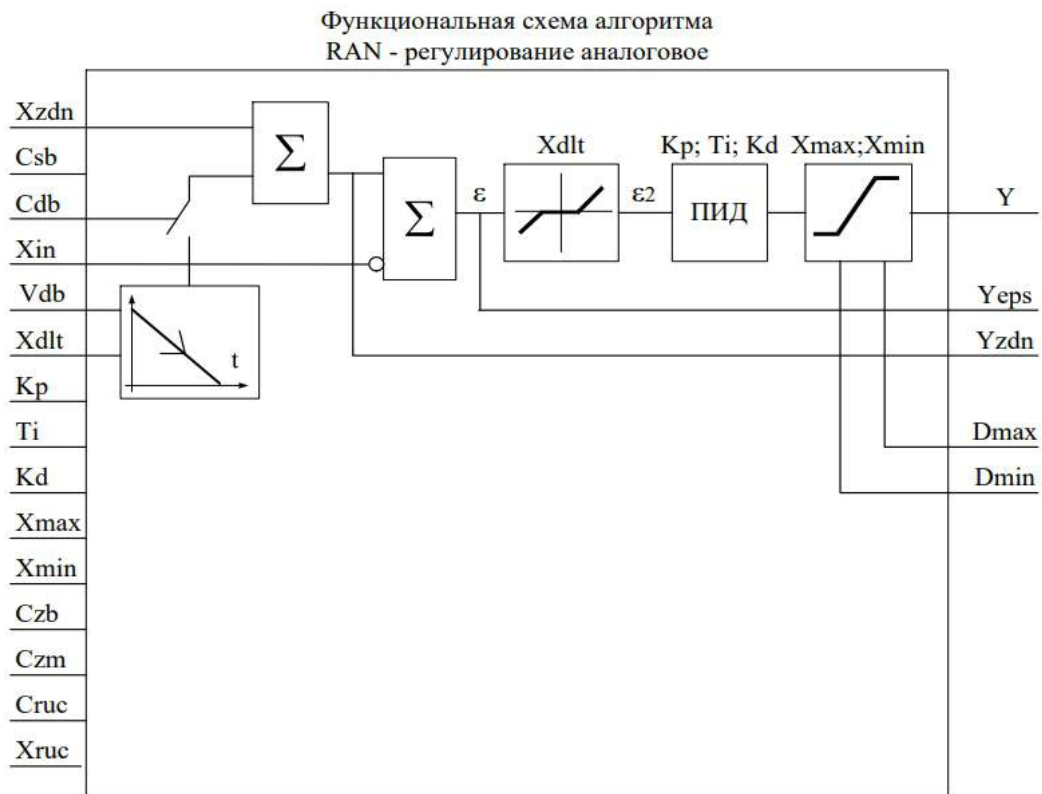


Рисунок 4.51 Функциональная схема алгоритма RAN - регулирование аналоговое

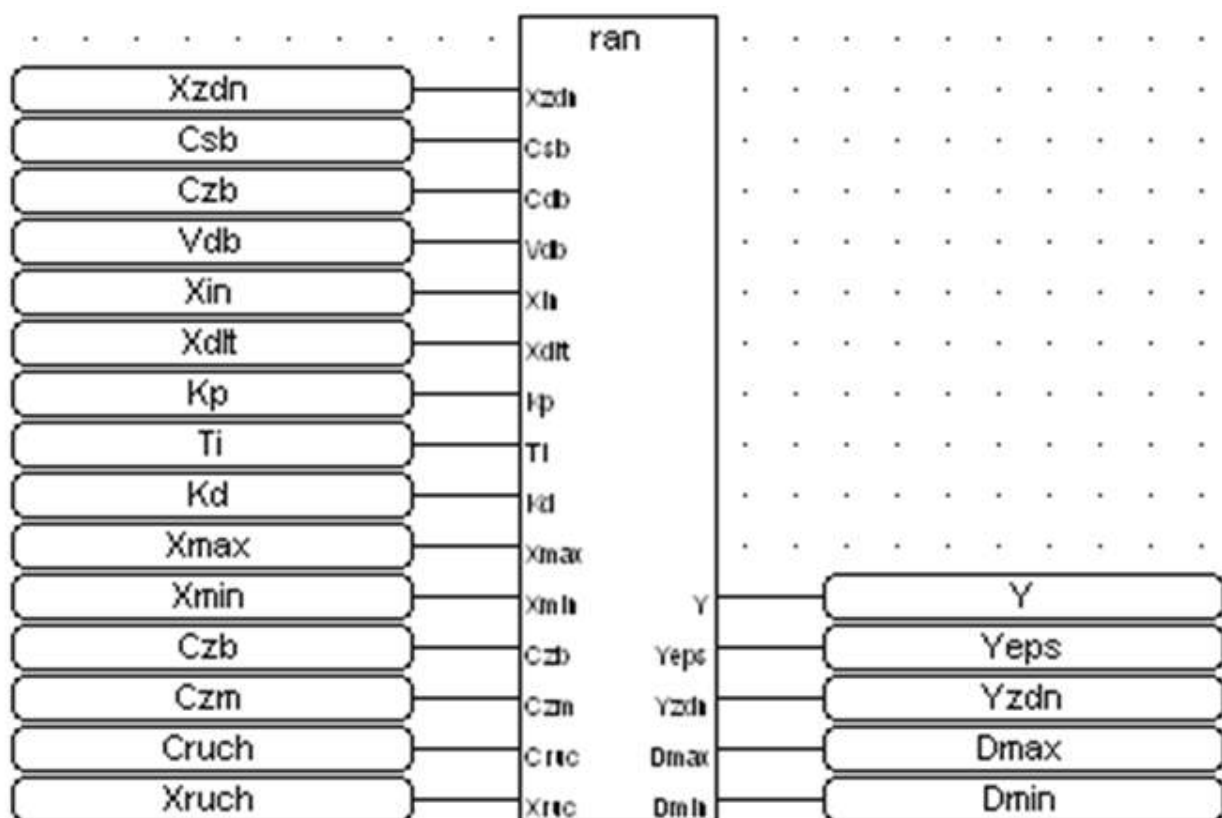


Рисунок 4.51 Функциональный блок RAN

После помещения функциональных блоков на рабочую область необходимо задать переменные (таблица 5).

Таблица 5

Имя переменной	Тип данных	Атрибуты	Формат	Начальное значение
Xzdn	Целые действительные	Константа	Вещественная	4
Vdb		Константа	Вещественная	0
Xin		Внутренняя	Вещественная	0
Xdlt		Константа	Вещественная	10^{-7}
Kp		Константа	Вещественная	2
Ti		Константа	Вещественная	1
Kd		Константа	Вещественная	3000
Xmax		Константа	Вещественная	20
Xmin		Константа	Вещественная	-20
Xruch		Константа	Вещественная	0
Y		Внутренняя	Вещественная	0
Yeps		Внутренняя	Вещественная	0
Yzdn		Внутренняя	Вещественная	0
address1		Константа	Целая	40
address2		Константа	Целая	30

Yemax	Булевские	Константа	Вещественная	0,1
Yemin		Константа	Вещественная	-0,1
VbIxodogr		Внутренняя	Вещественная	0
Czb		Константа		FALSE
Czm		Константа		
Cruch		Константа		
Dmax		Внутренняя		
Dmin		Внутренняя		
Csb		Константа		
Cdb		Константа		
BbIxod		Внутренняя		
sravntnie		Внутренняя		
Dogrmax		Внутренняя		
Dogrmin		Внутренняя		
pars	Константа			
Vxod	Внутренняя		TRUE	
Blink_period d	Тфймеры	Внутренняя		#!s

Задав переменные, помещаем их на рабочую область и соединяем с соответствующими входами-выходами функциональных блоков (Рисунок 4.52).

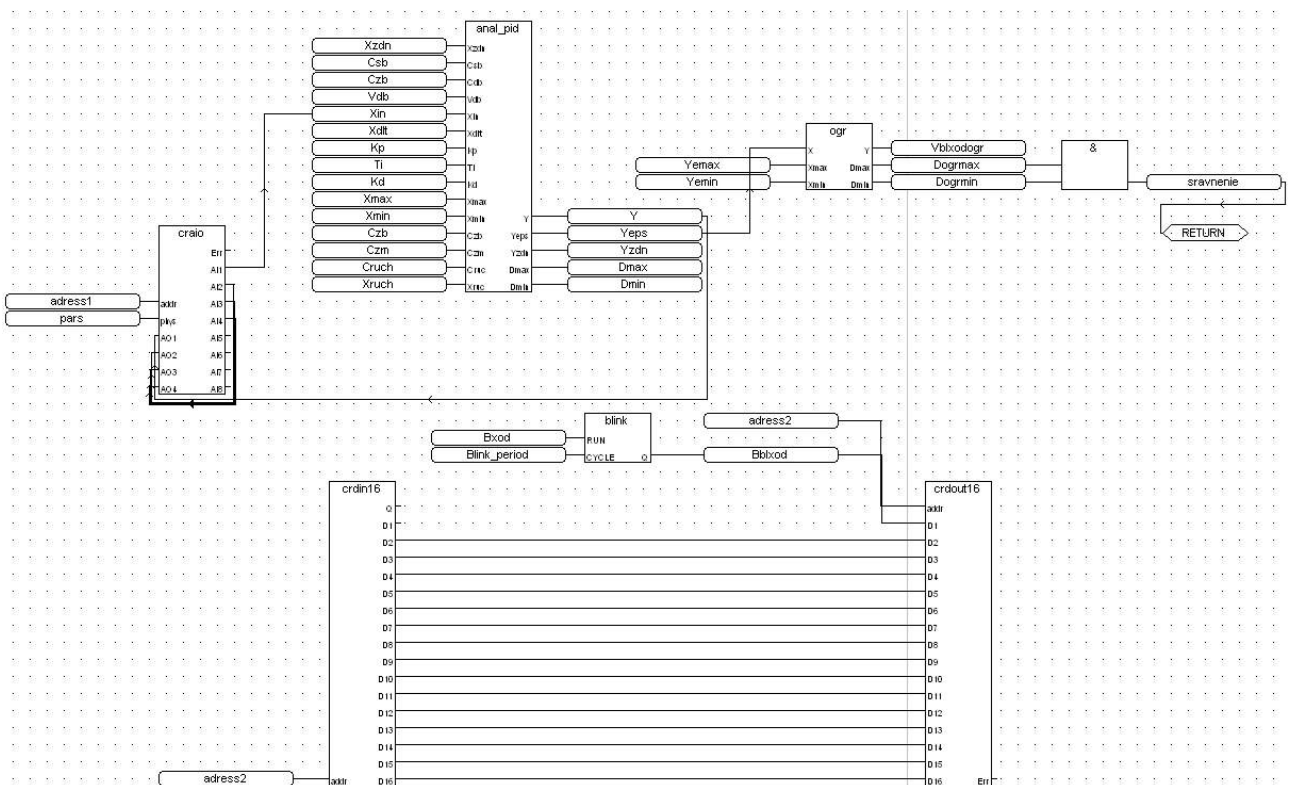


Рисунок 4.52 Реализация ПИД-регулятора с сигнализацией по ошибке

По сути, алгоритм работы программы заключается в регулировании ОУ

(рисунок 4.52). ПИД-регулятор реализован контроллером КРОСС, в который загружена программа ПИД-регулирования, написанная на языке FBD в системе ISAGRAF.

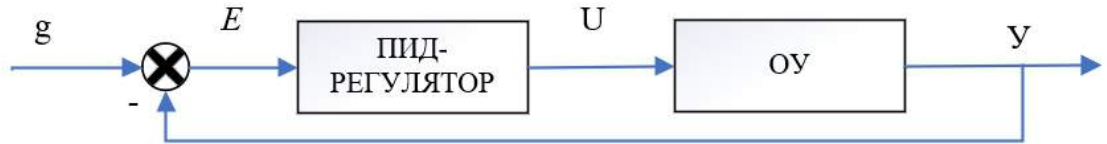


Рисунок 4.53 Структурная схема САР

Уставка g задается программно, выходное воздействие ОУ y поступает по обратной связи в контроллер, тем самым контроллер вырабатывает управляющее воздействие u .

Функциональный блок «Ограничение» задает диапазон уровня ошибки ($-0,1 \leq Y_{\text{eps}} \leq 0,1$). Если уровень ошибки выходит за этот диапазон, светодиод мерцает. Реализация мерцания светодиода дана в четвертой программе этого методического пособия.

После этого наша программа реализации ПИД-регулятора с сигнализацией по ошибке примет вид в режиме отладки (Рисунок 4.54):

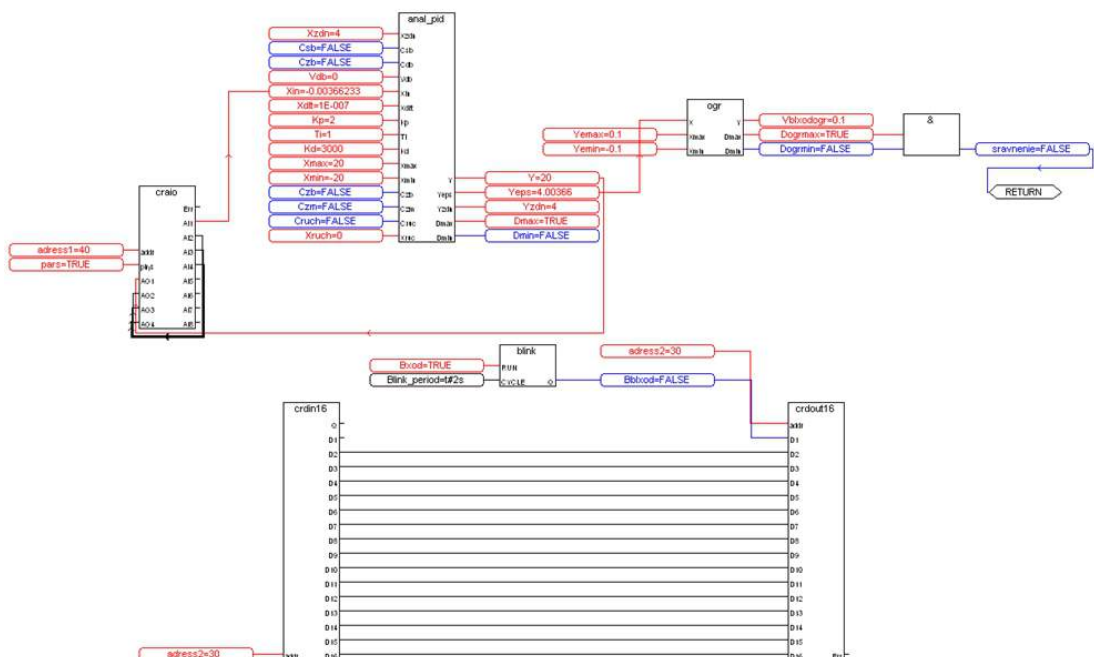


Рисунок 4.53 Состояние программы в режиме отладки