

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ
ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания

по организации и проведению самостоятельной работы
по дисциплине

«ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННО-
ИЗМЕРИТЕЛЬНЫЕ И УПРАВЛЯЮЩИЕ СИСТЕМЫ»

для студентов направления подготовки

11.04.04 «Электроника и наноэлектроника»

(профиль: «Интеллектуальные программируемые системы и
устройства»)

Ставрополь
2026

УДК
ББК

Рецензент:

Настоящие методические рекомендации предназначены для организации самостоятельной работы по дисциплине «Интеллектуальные информационно-измерительные и управляющие системы» для студентов очной формы обучения направления подготовки 11.04.04 Электроника и нанoeлектроника, профиль Интеллектуальные программируемые системы и устройства. В методических рекомендациях приведён план график выполнения самостоятельной работы по дисциплине, представлены вопросы для обсуждения, а также задания и вопросы для формирования и контроля владения компетенциями и список основной литературы, рекомендуемой для самостоятельной работы. Представленный список дополнительной литературы, позволяющий студентам более глубоко познакомиться с изучаемым предметом.

Методические рекомендации к самостоятельной работе студентов составлены в соответствии с требованиями ФГОС ВО по направлению подготовки 11.04.04 Электроника и нанoeлектроника, профиль Интеллектуальные программируемые системы и устройства, рабочими учебными планами и программами дисциплины «Автоматизированные системы обеспечения надежности и качества электронных средств».

СОДЕРЖАНИЕ

Введение	4
<u>1 ОБЩАЯ ХАРАКТЕРИСТИКА САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТА</u> <u>ПРИ ИЗУЧЕНИИ ДИСЦИПЛИНЫ «ИНТЕЛЛЕКТУАЛЬНЫЕ</u> <u>ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ И УПРАВЛЯЮЩИЕ</u> <u>СИСТЕМЫ».....</u>	<u>5</u>
1.1 Цель и задачи самостоятельной работы студентов.....	5
1.2 Тематика выполнения СРС, указания по форме контроля.....	9
2 ПЛАН-ГРАФИК ВЫПОЛНЕНИЯ СРС.....	11
3 КОНТРОЛЬНЫЕ ТОЧКИ И ВИДЫ ОТЧЕТНОСТИ ПО НИМ.....	11
4 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ИЗУЧЕНИЮ ТЕОРЕТИЧЕСКОГО МАТЕРИАЛА.....	12
4.1 Работа с литературой.....	12
4.2 Критерии оценки для проверки уровня обученности в рамках текущего контроля.....	19
5 СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ.....	21
5.1 Основная литература.....	21

Введение

Современные стандарты подготовки специалистов предусматривают значительный объем внеаудиторной работы. Освоение программы курса предполагает получение как теоретических знаний по проблемам моделирования, так и формирование навыков практической работы. Поэтому самостоятельная работа в рамках курса ориентирована как на теоретический, так и на практический аспект.

Самостоятельная работа студентов - это выполнение теоретических и практических заданий студентами по усвоению изучаемой дисциплины «Автоматизированные системы обеспечения надежности и качества электронных средств».

1 ОБЩАЯ ХАРАКТЕРИСТИКА САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТА ПРИ ИЗУЧЕНИИ ДИСЦИПЛИНЫ «ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ И УПРАВЛЯЮЩИЕ СИСТЕМЫ

»

1.1 Цель и задачи самостоятельной работы студентов

Целью самостоятельной работы является закрепление и углубление знаний, полученных студентами на лекциях, подготовке к текущим лабораторным и практическим занятиям, промежуточным формам контроля знаний и к итоговому контролю.

Дидактические цели самостоятельных занятий:

- формирование профессиональных умений;
- формирование умений и навыков самостоятельного умственного труда;
- мотивирование регулярной целенаправленной работы по освоению специальности;
- развитие самостоятельности мышления;
- формирование убежденности, волевых черт характера, способности к самоорганизации;
- овладение технологическим учебным инструментом.

Задача самостоятельной работы студентов - систематическое изучение дисциплин в течение семестра, закрепление и углубление полученных знаний и навыков, подготовка к предстоящим занятиям, а также формирование культуры умственного труда и самостоятельности в поиске и приобретении новых знаний и умений, и, в том числе, формирование компетенций.

Основная тенденция инноваций в области образования определяется как переход от «научения к изучению»

Именно систематическое изучение учебной дисциплины позволяет студенту достигнуть уровня требований федерального государственного образовательного стандарта высшего профессионального образования (ФГОС ВПО) к профессиональной подготовленности, которые сформулированы следующим образом: знать, уметь, владеть.

В современной науке существует несколько классификаций самостоятельной работы студентов. Один из вариантов такой классификации представлен в табл. 1.

Таблица 1.

Виды самостоятельной работы студентов

Виды	Содержание
Репродуктивная	Повторение учебного материала, самостоятельный просмотр, прочтение, конспектирование учебной литературы; прослушивание, запоминание, заучивание и пересказ магнитофонных записей лекций, Интернет-ресурсы и др.
Познавательного-поисковая	Написание курсовых, контрольных работ и рефератов. Разработка сообщений, эссе, докладов, докладов с презентациями. Подготовка выступлений на практических и семинарских занятиях, проработка литературы по дисциплинарным проблемам, и др.
Творческая	Подготовка дипломной работы (дипломного проекта), научных статей, рефератов, участие в научно-исследовательской работе, в студенческих и научно-практических конференциях.

Студенты в ходе выполнения самостоятельной работы должны руководствоваться ориентировочной основой деятельности на каждом этапе:

1 этап - определить цели самостоятельной работы;

2 этап - конкретизировать познавательные (практические или проблемные) задачи;

3 этап - оценить собственную готовность к самостоятельной работе по решению познавательных задач;

4 этап - выбрать оптимальный способ действий (технологии, методы и средства), ведущий к достижению поставленной цели через решение конкретных задач;

5 этап - спланировать (самостоятельно или с помощью преподавателя) программу самостоятельной работы;

6 этап - реализовать программу самостоятельной работы.

Планирование и контроль преподавателем самостоятельной работы студентов необходим для успешного ее выполнения. Преподаватель заранее планирует систему самостоятельной работы, учитывает все ее цели, формы, отбирает учебную и научную информацию и методические средства коммуникаций, продумывает свое участие и роль студента в этом процессе.

Вопросы для самостоятельной работы студентов, указанные в рабочей программе дисциплины, предлагаются преподавателями в начале изучения дисциплины. Студенты имеют право выбирать дополнительно интересующие их темы для самостоятельной работы.

Содержание деятельности преподавателя и студента при выполнении самостоятельной работы представлено в табл. 2.

Таблица 2

Содержание деятельности при выполнении самостоятельной работы

Основные характеристики	Деятельность преподавателя	Деятельность студентов
Цель выполнения СР	• объяснить смысл и цель самостоятельной работы; • дать подробный инструктаж о требованиях, предъявляемых к самостоятельной работе и методах ее выполнения; • продемонстрировать образец самостоятельной работы	• понять и принять цель самостоятельной работы как лично-значимую; • познакомиться с требованиями и образцами самостоятельной работы
Мотивация	• раскрыть теоретическую и практическую значимость выполнения самостоятельной работы, • сформировать познавательную потребность студента и готовность к выполнению самостоятельной работы; • мотивировать студента на достижение цели	• сформировать у себя познавательную потребность в выполнении самостоятельной работы; • сформировать целевую установку и принять решение о выполнении самостоятельной работы
Управление	• осуществлять управление через воздействие на каждом этапе процесса выполнения самостоятельной работы; • дать оптимальные технологии выполнения самостоятельной работы	самому осуществлять управление самостоятельной работой (проектировать, планировать, рационально распределять время и т.д.) на основе предложенных технологий

Продолжение таблицы 2

Основные характеристики	Деятельность преподавателя	Деятельность студентов
Контроль и коррекция выполнения	• осуществлять входной контроль, предполагающий выявление начального уровня готовности студента к выполнению самостоятельной работы; • намечать дальнейшие пути выполнения самостоятельной работы; • осуществлять итоговый контроль конечного результата выполнения самостоятельной работы	• осуществлять текущий и итоговый операционный самоконтроль за ходом выполнения самостоятельной работы; • самоанализ и исправление допущенных ошибок и внесение корректив в работу; • ведение поиска оптимальных способов выполнения самостоятельной работы; • осуществлять рефлексию к собственной деятельности
Оценка	• давать оценку самостоятельной работе на основе сличения результата с образцом; • давать методические рекомендации по выполнению самостоятельной работы выявлять затруднения и типичные ошибки; подчеркивать положительные и отрицательные стороны; • устанавливать уровень и определять уровень продвижения студента и тем самым сформировать у него мотивацию достижения успеха в учебной деятельности	дать оценку собственной работе, своим познавательным возможностям и способностям сопоставляя достигнутый результат с целью самостоятельной работы

Самостоятельная работа включает те разделы курса, которые не получили достаточного освещения на лекциях по причине ограниченности лекционного времени и большого объема изучаемого материала.

Методическое обеспечение самостоятельной работы состоит из:

- 1) Определения учебных вопросов, которые студенты должны изучить самостоятельно;
- 2) Подбора необходимой учебной литературы, обязательной для проработки и изучения;
- 3) Поиска дополнительной научной литературы, к которой студенты могут обращаться по желанию, если у них возникает интерес в данной теме;
- 4) Определения контрольных вопросов, позволяющих студентам самостоятельно проверить качество полученных знаний;

5) Организации консультаций преподавателя со студентами для разъяснения вопросов, вызвавших у обучающихся затруднения при самостоятельном освоении учебного материала.

1.2 Тематика выполнения СРС, указания по форме контроля

Тематика СРС соответствует рабочей программе дисциплины «ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ И УПРАВЛЯЮЩИЕ СИСТЕМЫ» по направлению подготовки 11.04.04 «Электроника и наноэлектроника», профиль. Темы разделов дисциплины и виды самостоятельной работы даны в таблице 3.

Таблица 3 – Темы разделов дисциплины и виды самостоятельной работы

№	Наименование разделов и тем дисциплины, их краткое содержание; вид самостоятельной работы	Форма контроля	Зачетные единицы (часы)
2 семестр			
	Подготовка к лекционным занятиям	Выборочный опрос	0,148 (1,20)
	Подготовка к практическим занятиям	Опрос в рамках допуска к работе	0,089 (2,4)
1	Основы теории надежности		0,287 (7,75)
	Самостоятельной изучение темы «Программирование в MathCad».	Выборочный опрос	0,287 (7,75)
2	Методы расчета надежности невосстанавливаемых и восстанавливаемых объектов		0,6481 (17,5)
	Самостоятельной дополнительное изучение темы «Модели пассивных компонентов».	Выборочный опрос	0,0923 (2,5)
	Самостоятельной дополнительное изучение темы «Модели биполярных транзисторов».	Выборочный опрос	0,0923 (2,5)
	Самостоятельной изучение дополнительное темы «Модели МОП-транзисторов».	Выборочный опрос	0,0923 (2,5)
	Самостоятельной изучение дополнительное темы «Макромодели».	Выборочный опрос	0,0923 (2,5)
	Самостоятельной дополнительное изучение темы «Источники сигналов и питания».	Выборочный опрос	0,0923 (2,5)

	Самостоятельной дополнительное изучение темы «Директивы управления заданием».	Выборочный опрос	0,0923 (2,5)
	Самостоятельной дополнительное изучение темы «Моделирование цифро-аналоговых устройств в систем PSpice».	Выборочный опрос	0,0923 (2,5)
3	Прогнозирование показателей качества и надежности компонентов радиоэлектронных средств		0,3704 (10,0)
	Самостоятельной дополнительное изучение темы «Типы данных языка VHDL».	Выборочный опрос	0,0923 (2,5)
	Самостоятельной дополнительное изучение темы «Синтаксис языка VHDL».	Выборочный опрос	0,0923 (2,5)
	Самостоятельной дополнительное изучение темы «Описание VHDL-объекта».	Выборочный опрос	0,0923 (2,5)
	Самостоятельной изучение дополнительное темы «Описание цифровых микросхем на языке VHDL».	Выборочный опрос	0,0923 (2,5)
3 семестр			
4	Оценка надежности и качества РЭС.		0,3019 (8,15)
	Самостоятельной дополнительное изучение темы «Моделирование цифровых и цифро-аналоговых схем в ANALOG OR MIXED-SIGNAL CIRCUIT.».	Выборочный опрос	0,0923 (2,5)
	Самостоятельной дополнительное изучение темы «Моделирование цифровых схем в проекте PC BOARD и PROGRAMMABLE LOGIC».	Выборочный опрос	0,1167 (3,15)
	Самостоятельной дополнительное изучение темы «Проектирование печатных плат в модуле LAYOUT PLUS».	Выборочный опрос	0,0923 (2,5)
5	Расчет надежности в подсистеме «Автоматизированной системы обеспечения надёжности и качества аппаратуры» (АСОНИКА)		0,3704 (10,0)
	Самостоятельной дополнительное изучение темы «Интерфейс программы MultiSim».	Выборочный опрос	0,0923 (2,5)
	Самостоятельной дополнительное изучение темы «Виртуальные приборы MultiSim».	Выборочный опрос	0,0923 (2,5)
	Самостоятельной дополнительное изучение темы «Виртуальные компоненты MultiSim».	Выборочный опрос	0,0923 (2,5)
	Самостоятельной дополнительное изучение темы «Виды анализа в MultiSim».	Выборочный опрос	0,0923 (2,5)
	Выполнение расчетно-графической работы	опрос	0,555(15)
	Итого		2,67 (72,0)

2 ПЛАН-ГРАФИК ВЫПОЛНЕНИЯ СРС

Подготовка к лекциям и практическим занятиям осуществляется непосредственно перед каждым занятием. План-график выполнения самостоятельной работы

Таблица 4 - План-график выполнения самостоятельной работы

Код реализуемой компетенции	Вид деятельности студентов	Итоговый продукт самостоятельной работы	Средства и технологии оценки	Объём часов
2,3 семестр				
ПК-1 ПК-2 ПК-3 ПК-6	Подготовка к лекционным занятиям	конспект	собеседование	1,2
ПК-1 ПК-2 ПК-3 ПК-6	Подготовка к практическим занятиям	конспект	собеседование	2,4
ПК-1 ПК-2 ПК-3 ПК-6	Самостоятельное изучение источников и литературы по разделу №№1-5	конспект	собеседование	53,95
ПК-1 ПК-2 ПК-3 ПК-6	Выполнение курсового проекта	Курсовой проект	защита	25
Итого:				108,00

3 КОНТРОЛЬНЫЕ ТОЧКИ И ВИДЫ ОТЧЕТНОСТИ ПО НИМ

приведен в таблице 4.

Контроль знаний студентов включает формы текущего и итогового контроля. Текущий контроль осуществляется в процессе изучения курса и включает в себя оценку работы студентов на лабораторных занятиях, а также защиту контрольной работы. Рейтинговая оценка знаний студентов в рамках текущего контроля приведена в таблице 5.

Таблица 5 - Рейтинговая оценка знаний студентов.

№ п/п	Вид деятельности студентов	Сроки выполнения	Количество баллов
2 семестр			
1.	Зачетное занятие по практическим занятиям № 1–6 с проверкой выполнения разноуровневых индивидуальных заданий по моделированию.	7 неделя	10

2.	Зачетное занятие по практическим занятиям № 7–11 с проверкой выполнения разноуровневых индивидуальных заданий по моделированию. 1-е тестирование.	12 неделя	12
3.	Зачетное занятие по практическим занятиям № 12–16 с проверкой выполнения разноуровневых индивидуальных заданий по моделированию. 2-е тестирование.	16 неделя	23
4.	Выполнение реферат	16 неделя	20
	Итого за 2 семестр		55
	Итого		55

4 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ИЗУЧЕНИЮ ТЕОРЕТИЧЕСКОГО МАТЕРИАЛА

Опрос по темам, изученным самостоятельно, может проходить в интерактивной форме в виде дискуссий или презентаций.

Выполнение заданий ведется в тетрадях для СРС или в электронном виде. Второй вид может быть использован для конспектирования.

Подготовка к практическим занятиям представляет собой внеаудиторную самостоятельную работу студентов и заключается в чтении:

- конспекта соответствующей лекции (если она читалась по данной теме);
- соответствующего раздела учебника и первоисточников (основная и дополнительная литература, интернет-ресурсы).

4.1 Работа с литературой

Важной составляющей самостоятельной внеаудиторной подготовки является работа с литературой ко всем видам занятий: семинарским, практическим, при подготовке к зачетам, экзаменам, тестированию, участию в научных конференциях.

Умение работать с литературой означает научиться осмысленно пользоваться источниками. Прежде чем приступить к освоению научной литературы, рекомендуется чтение учебников и учебных пособий.

Существует несколько методов работы с литературой.

Один из них - самый известный - метод повторения: прочитанный текст можно заучить наизусть. Простое повторение воздействует на память механически и поверхностно. Полученные таким путем сведения легко забываются.

Наиболее эффективный метод - метод кодирования: прочитанный текст нужно подвергнуть большей, чем простое заучивание, обработке. Чтобы основательно обработать информацию и закодировать ее для хранения, важно произвести целый ряд мыслительных операций: прокомментировать новые данные; оценить их значение; поставить вопросы; сопоставить полученные сведения с ранее известными.

Для улучшения обработки информации очень важно устанавливать осмысленные связи, структурировать новые сведения.

Изучение научной, учебной и иной литературы требует ведения рабочих записей.

Форма записей может быть весьма разнообразной: простой или развернутый план, тезисы, цитаты, конспект.

План - первооснова, каркас какой-либо письменной работы, определяющие последовательность изложения материала.

План является наиболее краткой и потому самой доступной и распространенной формой записей содержания исходного источника информации. По существу, это перечень основных вопросов, рассматриваемых в источнике. План может быть простым и развернутым. Их отличие состоит в степени детализации содержания и, соответственно, в объеме.

Преимущество плана состоит в следующем.

Во-первых, план позволяет наилучшим образом уяснить логику мысли автора, упрощает понимание главных моментов произведения.

Во-вторых, план позволяет быстро и глубоко проникнуть в сущность построения произведения и, следовательно, гораздо легче ориентироваться в его содержании.

В-третьих, план позволяет - при последующем возвращении к нему - быстрее обычного вспомнить прочитанное.

В-четвертых, с помощью плана гораздо удобнее отыскивать в источнике нужные места, факты, цитаты и т. д.

Выписки - небольшие фрагменты текста (неполные и полные предложения, отдельные абзацы, а также дословные и близкие к дословным записи об излагаемых в нем фактах), содержащие в себе квинтэссенцию содержания прочитанного.

Выписки представляют собой более сложную форму записей содержания исходного источника информации. По сути, выписки - не что иное, как цитаты, заимствованные из текста. Выписки позволяют в концентрированной форме и с максимальной точностью воспроизвести в произвольном (чаще последовательном) порядке наиболее важные мысли автора, статистические и даталогические сведения. В отдельных случаях — когда это оправданно с точки зрения продолжения работы над текстом - вполне допустимо заменять цитирование изложением, близким к дословному.

Тезисы - сжатое изложение содержания изученного материала в утвердительной (реже опровергающей) форме.

Отличие тезисов от обычных выписок состоит в следующем. *Во-первых*, тезисам присуща значительно более высокая степень концентрации материала. *Во-вторых*, в тезисах отмечается преобладание выводов над общими рассуждениями. *В-третьих*, чаще всего тезисы записываются близко к оригинальному тексту, т. е. без использования прямого цитирования.

Исходя из сказанного, нетрудно выявить основное преимущество тезисов: они незаменимы для подготовки глубокой и всесторонней аргументации письменной работы любой сложности, а также для подготовки выступлений на защите, докладов и пр.

Аннотация - краткое изложение основного содержания исходного источника информации, дающее о нем обобщенное представление. К написанию аннотаций прибегают в тех случаях, когда подлинная ценность и пригодность исходного источника информации исполнителю письменной работы окончательно неясна, но в то же время о нем необходимо оставить краткую запись с обобщающей характеристикой. Для указанной цели и используется аннотация.

Характерной особенностью аннотации наряду с краткостью и обобщенностью ее содержания является и то, что пишется аннотация всегда после того, как (хотя бы в предварительном порядке) завершено ознакомление с содержанием исходного источника информации. Кроме того, пишется аннотация почти исключительно своими словами и лишь в крайне редких случаях содержит в себе небольшие выдержки оригинального текста.

Резюме - краткая оценка изученного содержания исходного источника информации, полученная, прежде всего, на основе содержащихся в нем выводов. Резюме весьма сходно по своей сути с аннотацией. Однако, в отличие от последней, текст резюме концентрирует в себе данные не из основного содержания исходного источника информации, а из его заключительной части, прежде всего выводов. Но, как и в случае с аннотацией, резюме излагается своими словами - выдержки из оригинального текста в нем практически не встречаются.

Конспект - сложная запись содержания исходного текста, включающая в себя заимствования (цитаты) наиболее примечательных мест в сочетании с планом источника, а также сжатый анализ записанного материала и выводы по нему.

Для работы над конспектом следует:

- определить структуру конспектируемого материала, чему в значительной мере способствует письменное ведение плана по ходу изучения оригинального текста;
- в соответствии со структурой конспекта произвести отбор и последующую запись наиболее существенного содержания оригинального текста — в форме цитат или в изложении, близком к оригиналу;
- выполнить анализ записей и на его основе - дополнение записей собственными замечаниями, соображениями, "фактурой", заимствованной из других источников и т. п. (располагать все это следует на полях тетради для записей или на отдельных листах-вкладках);
- завершить формулирование и запись выводов по каждой из частей оригинального текста, а также общих выводов.

Систематизация изученных источников позволяет повысить эффективность их анализа и обобщения. Итогом этой работы должна стать логически выстроенная система сведений по существу исследуемого вопроса.

Необходимо из всего материала выделить существующие точки зрения на проблему, проанализировать их, сравнить, дать им оценку.

Обзор - краткое точное изложение сущности какого-либо вопроса, темы на основе одной или нескольких книг, монографий или других первоисточников. Обзор

должен содержать основные фактические сведения и выводы по рассматриваемому вопросу.

Обзор выполняет следующие функции:

- информативная;
- поисковая;
- справочная;
- сигнальная;
- индикативная;
- коммуникативная.

Степень выполнения этих функций зависит от содержательных и формальных качеств обзора, а также от того, кто и для каких целей их использует. Язык обзора должен отличаться ясностью, точностью, краткостью и простотой. Содержание следует излагать объективно от имени автора.

При оценке обзора учитывается не только качество реферирования прочитанной литературы, но и аргументированное изложение собственных мыслей студента по рассматриваемому вопросу. Результат работы студента оценивается преподавателем по бальной системе. Также допускается оценивать работы, удовлетворяющие или не удовлетворяющие предъявляемым требованиям, «зачтено» или «не зачтено» соответственно.

Письменный обзор должен содержать следующие структурные элементы: титульный лист, заполненный по единой форме; оглавление с указанием всех разделов обзора и номерами страниц; введение объемом не более 1,5-2 печатные страницы; основная часть, которая содержит одну или несколько глав, состоящих из 23 параграфов (пунктов, разделов); заключение, которое содержит главные выводы основной части, и в котором отмечается выполнение задач и достижение цели, сформулированных во введении; приложения, включающие график и таблицы (если таковые имеются); библиографическое описание использованных источников оформленных по ГОСТ 7.82-2001, ГОСТ 7.1-2003. В тексте обзора обязательны ссылки на первоисточники.

Печатный вариант работы выполняется на белой бумаге формата А4 (210х297 мм). Текст работы излагается на одной стороне листа. Все линии, цифры, буквы и знаки работы должны быть черного цвета.

Текст обзора, рисунки, формулы, таблицы, а также номера страниц не должны выходить за пределы двухсантиметровой рамки листа А4. Номера страниц должны быть проставлены внизу по центру. При использовании текстового редактора Word, для выполнения этих условий необходимы следующие настройки:

- размер бумаги А4;
- поля слева, сверху, справа по 2 см, нижнее поле 2,5 см, расстояние от нижнего края страницы до нижнего колонтитула 2 см;
- номер страницы - внизу по центру.

Основной текст обзора быть должен быть набран шрифтом Times New Roman, размер 14 пт, начертание обычное, через полуторный интервал, выравнивание по ширине страницы. Для оформления таблиц и подписей к рисункам допускается Times New Roman, размер 12 пт.

Название каждой главы начинается с новой страницы, объем главы не может быть меньше 5 страниц. Заголовки и подзаголовки должны быть выделены и отличаться от основного текста (шрифтом, жирностью, курсивом и пр.). Подзаголовки следует отделять от основного текста сверху двумя строками, снизу - одной. В тексте должны отсутствовать сокращения, кроме общепринятых ГОСТ 7.88-2003, общепринятые или необходимые сокращения при первоначальном употреблении должны быть расшифрованы. Каждый рисунок, график или таблица в письменном обзоре должны быть пронумерованы и иметь заголовок или подпись. При наличии в обзоре ссылок на использованные научные или нормативные источники, сноски должны быть оформлены в соответствии с установленной формой по ГОСТ 7.32-2001.

Письменный обзор должен быть переплетен в обложку или помещен в папку-скоросшиватель (картонную или пластиковую).

Письменный обзор должен быть предоставлен в установленный преподавателем срок. В случае несвоевременного представления работы, Письменный обзор не проверяется преподавателем и не зачитывается как выполненный.

В процессе работы над обзором можно выделить 4этапа:

- вводный - выбор темы, работа над планом и введением;
- основной - работа над содержанием и заключением обзора;
- заключительный - оформление обзора;
- защита обзора (на практическом занятии, экзамене, студенческой конференции и т.д.)

Работа над обзором начинается с выбора темы исследования. Заинтересованность автора в проблеме определяет качество проводимого исследования и соответственно успешность его защиты. Выбирая круг вопросов своей работы, не стоит спешить воспользоваться списком тем, предложенным преподавателем. Надо попытаться сформулировать проблему своего исследования самостоятельно. При определении темы обзора нужно учитывать и его информационную обеспеченность. С этой целью, во-первых, можно обратиться к библиотечным каталогам, а во-вторых, проконсультироваться с преподавателем и библиотекарем.

Выбрав тему обзора и изучив литературу, необходимо сформулировать цель работы и составить план обзора. Возможно, формулировка цели в ходе работы будет меняться, но изначально следует ее обозначить, чтобы ориентироваться на нее в ходе исследования. Определяясь с целью дальнейшей работы, параллельно надо думать над составлением плана: необходимо четко соотносить цель и план работы.

План - это точный и краткий перечень положений в том порядке, как они будут расположены в обзоре, этапы раскрытия темы. Существует два основных типа плана: простой и сложный (развернутый). В простом плане содержание обзора делится на параграфы, а в сложном на главы и параграфы. При работе над планом обзора необходимо помнить, что формулировка пунктов плана не должна повторять формулировку темы.

При работе над введением необходимо опираться на навыки, приобретенные при написании изложений и сочинений. В объеме обзора введение, как правило, составляет 1 -2 машинописные страницы. Введение обычно содержит вступление, обоснование актуальности выбранной темы, формулировку цели и задач обзора, краткий обзор литературы и источников по проблеме, историю вопроса и вывод. Содержание обзора должно соответствовать теме, полно ее раскрывать. Все рас-

суждения нужно аргументировать. Обзор показывает объективное отношение автора к излагаемому материалу. Следует стремиться к тому, чтобы изложение было ясным, простым и точным.

Заключение - самостоятельная часть обзора. Оно не должно быть переложением содержания работы. Заключение должно содержать основные выводы в сжатой форме, а также оценку полноты и глубины решения тех вопросов, которые вставали в процессе изучения темы.

Объем заключения не должен превышать 2 печатных страниц.

Типичными ошибками, допускаемыми студентами при подготовке обзора, являются:

- недостаточное обоснование актуальности, практической и теоретической значимости полученных результатов, поверхностный анализ используемого материала;
- неглубокие критические оценки и рекомендации по решению исследуемой проблемы;
- поверхностные выводы и предложения;
- нарушение требований к оформлению письменного обзора;
- использование информации без ссылок на источник.

4.2 Критерии оценки для проверки уровня обученности в рамках текущего контроля

Оценка «отлично» выставляется студенту, если студент показал глубокое, прочное и аргументированное освоение программного учебного материала, при этом поставленный вопрос раскрыт последовательно, четко и логически стройно, в полном исчерпывающем объеме, основные категории, понятия и термины учебного курса формулировались правильно, не допущено при ответе ошибок

Оценка «хорошо» выставляется студенту, если студент показал твердое знание программного учебного материала, при этом поставленный вопрос раскрыт грамотно и по существу, в достаточно полном объеме, основные категории, понятия и термины

учебного курса формулировались правильно, допущены при ответе отдельные неточности или одна ошибка

Оценка «удовлетворительно» выставляется студенту, если студент показал знание только основной части учебного материала без его частных деталей, при этом поставленный вопрос раскрыт с нарушением логической последовательности, не в полном объеме; были допущены неточные формулировки основных категорий, понятий и терминов учебного курса, а также ошибки (не более двух) или ряд незначительных неточностей, не исказивших существенно суть ответа.

Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки (более двух), существенно исказившие его суть. Оценка неудовлетворительно выставляется также, если отсутствует ответ на вопрос, либо студент отказался его сдавать.

По результатам опроса в рамках текущего контроля выставляется рейтинговый балл по каждому рейтинговому мероприятию. Рейтинговый балл, выставляемый студенту за текущее контрольное мероприятие, сданное студентом в установленные графиком контрольных мероприятий сроки, определяется следующим образом:

Таблица 6 – Расчет рейтинговых баллов по результатам текущего контроля

Уровень выполнения контрольного задания	Рейтинговый балл (в % от максимального балла за контрольное задание)
Отличный	100
Хороший	80
Удовлетворительный	60
Неудовлетворительный	0

Конкретное значение рейтингового балла определяется распределением рейтинговых баллов по контрольным мероприятиям.

6 СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

6.1 Основная литература

1. Афонин В.В. Моделирование систем / Афонин В.В., Федосин С.А.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 269 с. [Электронный ресурс]
2. Сильвашко, С.А. Программные средства компьютерного моделирования элементов и устройств электроники : учебное пособие / С.А. Сильвашко, С.С. Фролов - Оренбург : ОГУ, 2014. - 170 с. : ил., схем. - Библиогр.: с. 162163 [Электронный ресурс].
3. Болотовский Ю.И. OrCAD. Моделирование: учебное пособие/ Болотовский Ю.И., Таназлы Г.И.— М.: СОЛОН-ПРЕСС, 2009.— 200 с. [Электронный ресурс]

6.2. Дополнительная литература:

1. Поляков А.К. Языки VHDL и VERILOG в проектировании цифровой аппаратуры: учебное пособие/ Поляков А.К.— М.: СОЛОН-ПРЕСС, 2009.— 314 с.— [Электронный ресурс]
2. Дьяконов В.П. MATLAB 6.5 SP1/7 + Simulink 5/6. Обработка сигналов и проектирование фильтров / Дьяконов В.П.— М.: СОЛОН-ПРЕСС, 2009.— 577 с. [Электронный ресурс]
3. Перельройзен Е.З. Проектируем на VHDL [Электронный ресурс]/ Перельройзен Е.З.— М.: СОЛОН-ПРЕСС, 2008.— 448 с.— [Электронный ресурс]
4. Бибило П.Н. Синтез логических схем с использованием языка VHDL [Электронный ресурс]/ Бибило П.Н.— М.: СОЛОН-ПРЕСС, 2009.— 384 с.— [Электронный ресурс]
5. А. А. Самарский, А. П. Михайлов Математическое моделирование. Москва: Физматлит, 2001. 320 с.
6. Антипенский Р. В. Схемотехническое проектирование и моделирование радиоэлектронных устройств / Р. В. Антипенский, А. Г. Фадин. - М.: Техносфера, 2007. - 128 с.

7. Плис А. И., Сливина Н.А. MathCAD: математический практикум для экономистов и инженеров. М.: Финансы и статистика, 1999. - 656 с.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего
образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к практическим занятиям

по дисциплине «Интеллектуальные информационно-измерительные и
управляющие системы » для студентов направления подготовки

11.04.04 Электроника и наноэлектроника

Направленность (профиль)

Интеллектуальные программируемые системы и устройства

Ставрополь 2026 г.

Содержание

Введение	6
1 Основы программирования программируемых логических контроллеров	7
1.1 Среда программирования ПЛК <i>CoDeSys</i> 2.3	7
1.2 Программирование на языке <i>LD</i>	13
1.3 Программирование функциональных блоков и функций	24
1.4 Практическая работа 1 «Изучение технических характеристик и основ программирования контроллера ОВЕН ПЛК150»	31
1.4.1 Техническое описание лабораторного стенда	31
1.4.2 Общие технические характеристики	32
1.4.3 Состав лабораторного стенда	33
1.4.4 Схема подключения	34
1.4.5 Особенности выполнения требований техники безопасности при эксплуатации	35
1.4.6 Общие сведения об изучаемом оборудовании	36
1.4.7 Пример разработки пользовательской управляющей программы для	41
контроллера ПЛК150	41
1.4.8 Загрузка программы в контроллер и запуск программы	43
1.4.9 Порядок проведения практической работы и варианты заданий	48
1.4.10 Требования к отчёту	53
1.5 Контрольные вопросы	53
2 Изучение технических характеристик и основ программирования панели оператора ОВЕН СП270	55
2.1 Общие сведения об изучаемом оборудовании	55
2.2 Основы программирования панели оператора	56
2.3 Пример реализации пульта оператора для управления простейшим технологическим процессом	71
2.4 Практическая работа 2 «Разработка человеко-машинного интерфейса системы управления с применением панели оператора ОВЕН СП270	75
2.4.1 Выполнение практической работы	76

2.4.2 Требования к отчёту.....	77
2.5 Контрольные вопросы.....	77
3 Изучение технических характеристик и основ программирования программируемого реле ОВЕН ПР114.....	78
3.1 Программирование реле ОВЕН ПР114	78
3.2 Лабораторная работа 3 «Разработка управляющей программы для программируемого реле ОВЕН ПР114».....	89
3.2.1 Общие сведения об изучаемом оборудовании	90
3.2.2 Пример разработки пользовательской управляющей программы для программируемого реле ПР114.....	94
3.2.3 Загрузка программы	95
3.2.4 Отладка программы.....	96
3.2.5 Порядок проведения практической работы	97
3.2.6 Требования к отчёту.....	97
3.3 Контрольные вопросы.....	98
4 Изучение систем автоматизации технологических процессов с использованием программируемого реле ОВЕН ПР114 (1 вариант симуляции объекта)	99
4.1 Общие сведения об объекте автоматизации.....	99
4.2 Практическая работа 4 «Разработка программы для управления работой модели методической печи»	101
4.2.1 Общие положения	101
4.2.2 Пример разработки управляющей программы для реле ОВЕН ПР114 для управления имитацией методической печи.....	102
4.2.3 Задания для самостоятельного выполнения	107
4.2.4 Требования к отчёту.....	108
4.3 Контрольные вопросы.....	108
5 Изучение построения распределенных систем управления с использованием GSM- модема ОВЕН ПМ-01.....	110
5.1 Распределенные системы управления на основе GSM-модема	110
5.1.1 Основы конфигурирования и работы модема	112
5.1.2 Передача данных с помощью SMS-сообщений в текстовом режиме	115
5.1.3 Обмен данными по TCP/IP протоколу.....	115

5.1.4 Программирование таймера автоматической перезагрузки	116
5.2 Практическая работа 5 «Разработка управляющей программы для передачи данных через GSM-модем»	117
5.2.1 Общие сведения об изучаемом оборудовании	117
5.2.2 Задание для самостоятельного выполнения	122
5.2.3 Требования к отчёту	
123 5.3 Контрольные вопросы	
123 Список использованных источников	
124	

Введение

Современный уровень развития науки и техники обуславливает внедрение микропроцессорных средств автоматизации для управления технологическими процессами и производствами [1 – 3]. Обладающая большими возможностями микропроцессорная техника позволяет обеспечить точную и быструю обработку информации в любом месте технологического объекта или технологического процесса. Использование непрерывно развивающегося технического и программного обеспечения микропроцессорных приборов значительно ускорило разработку и реализацию цифровых систем автоматизации и управления. Одной из важнейших задач при проектировании таких систем является создание удобных для использования автоматизированных рабочих мест.

Применение контрольно-измерительных приборов и средств автоматизации фирмы «ОВЕН» [4] в ряде отраслей промышленности позволяет решить целый комплекс задач, связанных с управлением сложными производственными процессами, уменьшить влияние человеческого фактора и обеспечить безопасность производства.

Учебное пособие предназначено для практического изучения настройки, программирования и эксплуатации микропроцессорного контроллера ПЛК150 [5], графической панели оператора с сенсорным управлением СП270 [6], программируемого реле ПР114 [7] и GSM-модема ПМ-01 фирмы «ОВЕН».

Учебное пособие основано на методических указаниях [8], предназначенных для изучения элементов автоматизации лабораторного стенда «Система автоматического управления ОВЕН».

1 Основы программирования программируемых логических контроллеров

1.1 Среда программирования ПЛК CoDeSys 2.3

Для подготовки программ логических контроллеров ПЛК150, рассматриваемых в учебном пособии, служит среда программного обеспечения *CoDeSys* [8, 9].

CoDeSys является современным инструментом для программирования контроллеров в стандарте МЭК 61131-3, обладает рядом особенностей, выделяющих его среди других систем: быстрое внедрение, эффективные средства ввода, высокая производительность.

Для запуска программы необходимо на рабочем столе *Windows* дважды щелкнуть курсором мыши по соответствующему ярлыку. После этого откроется пустое окно программы *CoDeSys 2.3* (рисунок 1.1), в котором можно создать новый проект или открыть уже созданный проект.

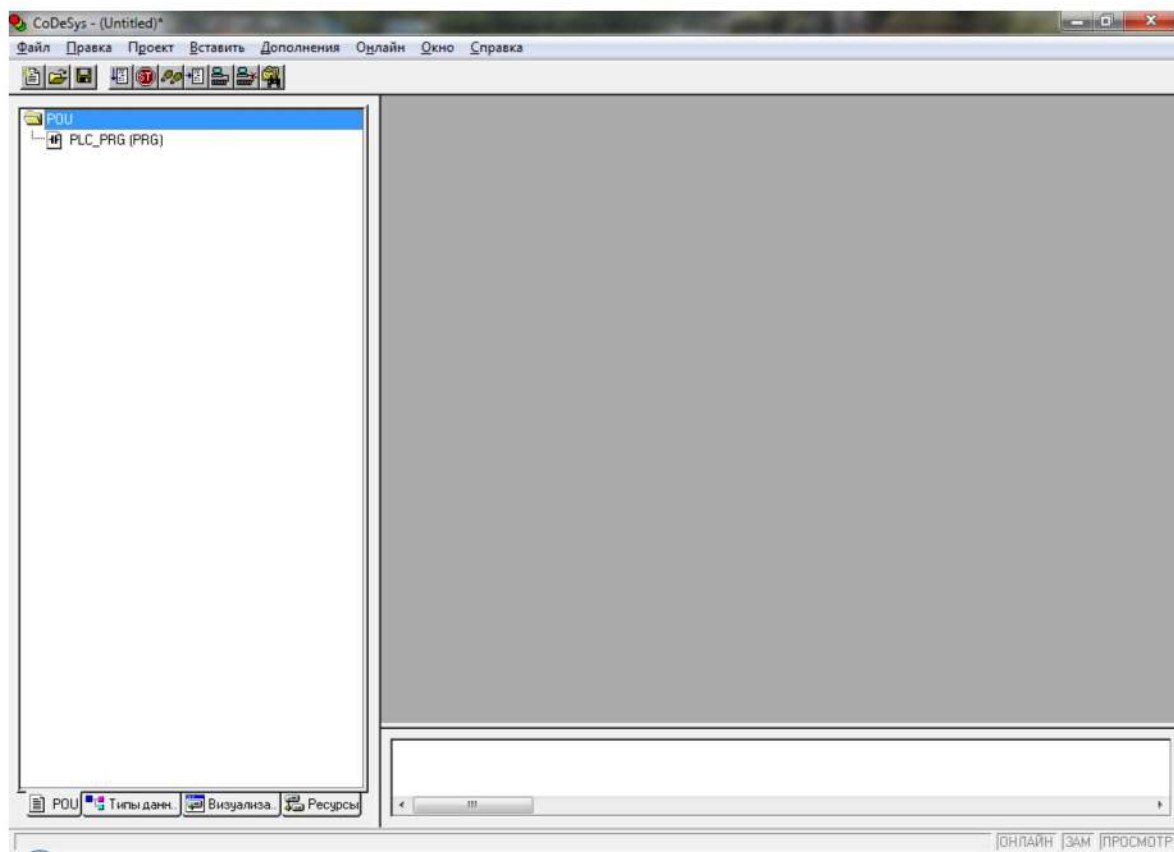


Рисунок 1.1 – Стартовое окно *CoDeSys 2.3*

Для создания нового проекта необходимо в выпадающем меню «Файл» выбрать пункт «Создать», после чего появится окно конфигурирования проекта «Настройка целевой платформы» (рисунок 1.2).

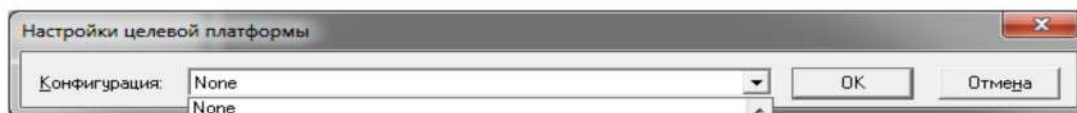


Рисунок 1.2 – Окно выбора контроллера

В выпадающем списке вариантов конфигурации (*target*-файлов) выбрать используемый в стенде контроллер ПЛК150.А-L. *Target*-файл содержит информацию о ресурсах контроллера: количестве и типах входов и выходов, интерфейсов, памяти, дополнительных устройствах и т.д., с которыми работает программная среда *CoDeSys*. После выбора контроллера появится окно общих настроек выбранного ПЛК, в котором необходимо выставить такие же значения параметров, как представлено на рисунке 1.3. После завершения настройки необходимо нажать «ОК».

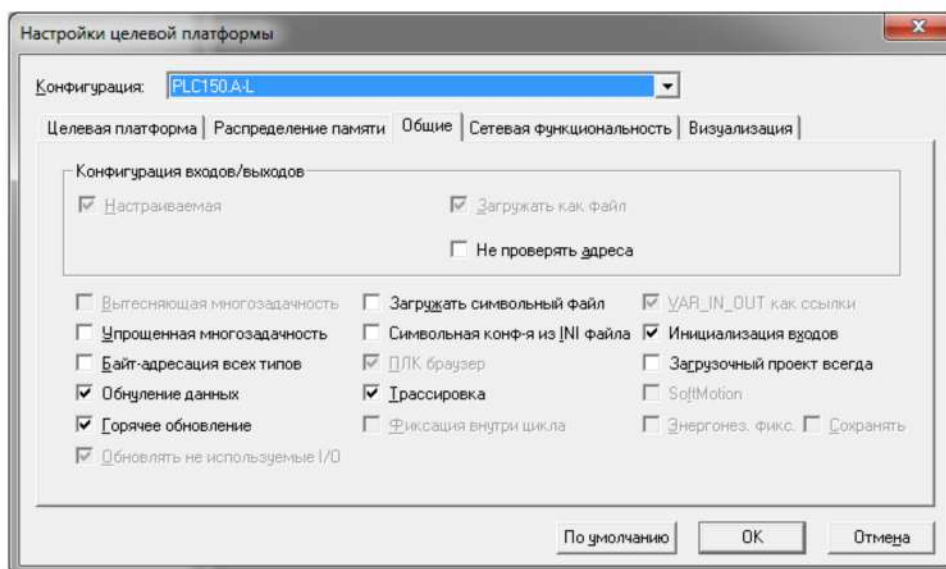


Рисунок 1.3 – Окно настройки параметров контроллера

После конфигурирования проекта откроется окно настройки программных компонентов (*POU*) проекта (рисунок 1.4), в котором выбирается имя и тип *POU* и язык реализации управляющей программы. После завершения настройки необходимо нажать «ОК».

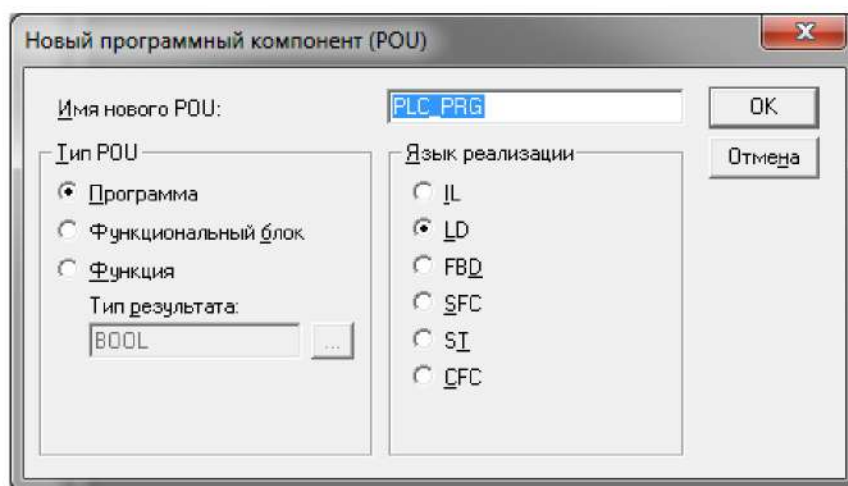


Рисунок 1.4 – Окно настройки *POU* проекта

CoDeSys поддерживает следующие программные компоненты (*POU*):

- программный блок (*PRG*);
- функциональный блок (*FB*);
- функция (*FC*).

CoDeSys поддерживает следующие языки программирования:

- текстовые: список инструкций (*IL*) и структурированный текст (*ST*);
- графические: язык последовательных функциональных схем (*SFC*), язык функциональных блокковых диаграмм (*FBD*), язык релейных диаграмм (*LD*).

Кроме того, *CoDeSys* включает поддержку основанного на языке *FBD* редактора непрерывных функциональных схем (*CFC*).

В *CoDeSys* введено несколько языков для того, чтобы пользователь мог применить язык, наиболее удобный для него: программисты выбирают язык *IL* или *ST*; специалисты, имеющие опыт работы с релейной логикой, выбирают язык *LD*; специалисты по системам автоматического управления (*CAU*) и схемотехнике выбирают привычный для них язык *FBD*.

Выбор одного из шести языков определяется не только предпочтениями пользователя, но и смыслом решаемой задачи.

Если исходная задача формулируется в терминах последовательной обработки и передачи сигналов, то для нее проще и нагляднее использовать язык *FBD*. Если задача описывается как последовательность срабатываний контактов и реле, то для нее нагляднее всего будет язык *LD*. Для задач, которые изначально

формулируются в виде сложного разветвленного алгоритма, удобнее будет язык *ST*.

В результате конфигурирования откроется рабочее окно программы, представленное на рисунке 1.5.

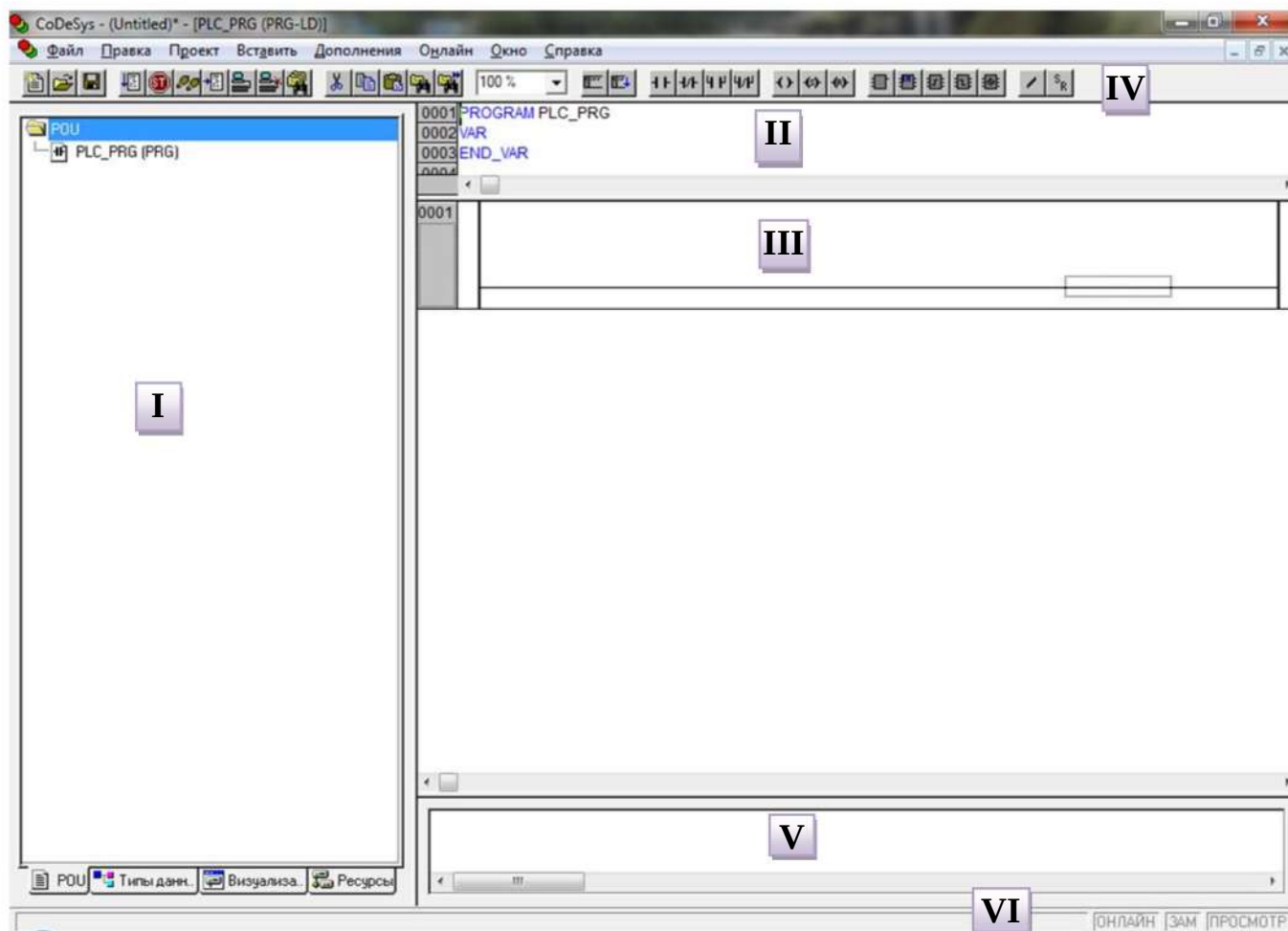


Рисунок 1.5 – Рабочее окно программы *CoDeSys 2.3*

Всю рабочую область программы можно условно разделить на несколько частей:

- I – окно менеджера проекта;
- II – область объявления переменных проекта;
- III – рабочая область;
- IV – панели инструментов;
- V – окно сообщений;
- VI – строка состояния.

В менеджере проекта все настройки проекта распределены по четырём закладкам:

- 1) «Менеджер программных компонентов» (*POU*), где можно редактировать текущие или добавлять новые компоненты (программы, блоки, функции);
- 2) «Менеджер данных», позволяющая добавлять и редактировать уже созданные базы данных проекта;
- 3) «Менеджер визуализаций», на которой имеется возможность управления (создание и редактирование свойств) экранов для системы визуализации;
- 4) «Менеджер ресурсов ПЛК», объединяющий ряд меню, необходимых для конфигурирования свойств ПЛК – различные библиотеки, конфигураторы контроллера, задач и тревог, менеджеры библиотек и параметров и ряд других.

В рабочей области проекта производится непосредственно набор управляющей программы.

Панель инструментов (рисунок 1.6) используется для создания пользовательской управляющей программы, также компиляции, отладки и загрузки проекта в память ПЛК.



Рисунок 1.6 – Панель инструментов

При выборе в окне настроек различных языков программирования, например, *LD*, *FBD* и *SFC*, панель инструментов принимает различный вид. В таблице 1.1 представлено краткое описание функций, доступ к которым осуществляется с панели инструментов при программировании на языке LD, а также представлены пиктограммы панели инструментов для других языков МЭК 61131-3: *FBD* и *SFC*.

Далее в соответствии с поставленной задачей необходимо разработать алгоритм автоматизации управления объектом либо в виде логических уравнений, либо в виде схемы алгоритма. Разработчик присваивает входным, выходным сигналам и внутренним переменным адреса в соответствии с конфигурацией контроллера.

Таблица 1.1 – Краткое описание функций панели инструментов

Язык	Пиктограмма	Описание
LD		Вставка нового сегмента до текущего сегмента
		Вставка нового сегмента после текущего сегмента
		Вставка замыкающего контакта
		Вставка размыкающего контакта
		Вставка параллельного замыкающего контакта
		Вставка параллельного размыкающего контакта
		Вставка выходной катушки
		Установка состояния <i>SET</i> обмотки выходного реле
		Установка состояния <i>RESET</i> обмотки выходного реле
		Установка функционального блока
		Установка функционального блока со входом <i>EN</i>
		Установка детектора переднего фронта
		Установка детектора заднего фронта
		Установка таймера <i>TON</i>
		Установка инверсии
		Установка функции <i>SET/RESET</i> на обмотке выходного реле
FBD		Добавление входа в функциональный блок
		Добавление выхода из функционального блока
		Добавление элемента схемы (функционального блока)
		Присваивание
		Переход
		Возврат
		Инверсия
		<i>SET/RESET</i>
SFC		Шаг-переход (сверху)
		Шаг-переход (снизу)
		Альтернативная ветвь (справа)
		Альтернативная ветвь (слева)
		Параллельная ветвь (справа)
		Параллельная ветвь (слева)
		Безусловный переход
		Переход - безусловный переход
		Использовать МЭК-шаги

Адресация контроллера имеет следующие особенности:

- прямое указание адреса дает способ непосредственного обращения к конкретной области памяти;
- использование области памяти через объявленную особым образом пере-

менную (входную, выходную или локальную).

Прямой адрес образуется из префикса «%», префиксов области памяти и размера, одного или нескольких целых чисел, разделенных точкой.

Префиксы области памяти: *I* – входы, *Q* – выходы, *M* – память данных.

Префиксы размера: *X* – один бит, *B* – байт (8 бит), *W* – слово (16 бит), *D* – двойное слово (32 бит).

Примеры:

- 1) %QX7.5 и %7.5 – бит 7.5 в области выходов;
- 2) %IW215 – 215-е слово в области входов;
- 3) %QB7 – байт 7 в области выходов;
- 4) %MD48 – двойное слово в позиции памяти 48;
- 5) %IW2.5.7.1 – зависит от конфигурации ПЛК.

По умолчанию конфигурация ПЛК имеет вид, представленный в таблице 1.2.

Таблица 1.2 – Конфигурация ПЛК по умолчанию

Входные сигналы				Выходные сигналы			
Дискретные		Аналоговые		Дискретные		Аналоговые	
Адрес ПЛК	Адрес пользователя	Адрес ПЛК	Адрес пользователя	Адрес ПЛК	Адрес пользователя	Адрес ПЛК	Адрес пользователя
%IX0.0	I1	%ID3.0	AI1	%QX1.0	Q1	%QD7.0	AQ1
%IX0.1	I2	%ID4.0	AI2	%QX1.1	Q2	%QD8.0	AQ2
%IX0.2	I3	%ID5.0	AI3	%QX1.2	Q3		
%IX0.3	I4	%ID6.0	AI4	%QX1.3	Q4		
%IX0.4	I5						
%IX0.5	I6						

1.2 Программирование на языке LD

При программировании на языке *LD* в виде релейно-контактной схемы программа разделяется на сегменты. Каждый сегмент представляет собой отдельную цепь, по которой может протекать ток. Шина питания находится слева (вертикальная линия). Таким образом, каждому логическому уравнению соответствует свой сегмент.

По умолчанию в блоке *PLCPRG* (*PRG*) уже есть пустой шаблон сегмента. Для создания нового сегмента необходимо на панели инструментов нажать пиктограмму  «Цепь (Перед)» или  «Цепь (После)».

Рассмотрим ввод программы на языке лестничных диаграмм (*LD*) для реализации логического уравнения

$$Q1 = I1 \cdot \overline{I2} + I3, \quad (1.1)$$

где $Q1$ – логическая переменная, связанная с выходом $\%QX1.0$;

$I1$, $I2$, $I3$ – логические переменные, связанные с соответствующими дискретными входами, которые описаны в таблице 1.2;

$\overline{I2}$ – инверсное значение логической переменной $I2$.

В адресах контроллера уравнение (1.1) примет следующий вид:

$$\%QX1.0 = \%IX0.0 \cdot \overline{\%IX0.1} + \%IX0.2, \quad (1.2)$$

где $\%QX1.0$, $\%IX0.0$, $\overline{\%IX0.1}$, $\%IX0.2$ – переменные уравнения (1.1), записанные в адресах контроллера.

Шаг 1. Установить в токовую цепь (сегмент) обмотку (рисунок 1.7, а). Для этого нажать на кнопку .

Шаг 2. Щёлкнуть по тому месту (непосредственно по линии), куда нужно установить замыкающий (нормально открытый) контакт (рисунок 1.7, б).

Шаг 3. Вставить замыкающий контакт, нажав на пиктограмму  (рисунок 1.7, в).

Шаг 4. Вставить параллельный замыкающий контакт, нажав на пиктограмму  (рисунок 1.7, г).

Шаг 5. Выделить место, куда необходимо поставить нормально замкнутый (размыкающий) контакт (рисунок 1.7, д).

Шаг 6. Вставить размыкающий контакт, нажав на пиктограмму  (рисунок 1.7, е).

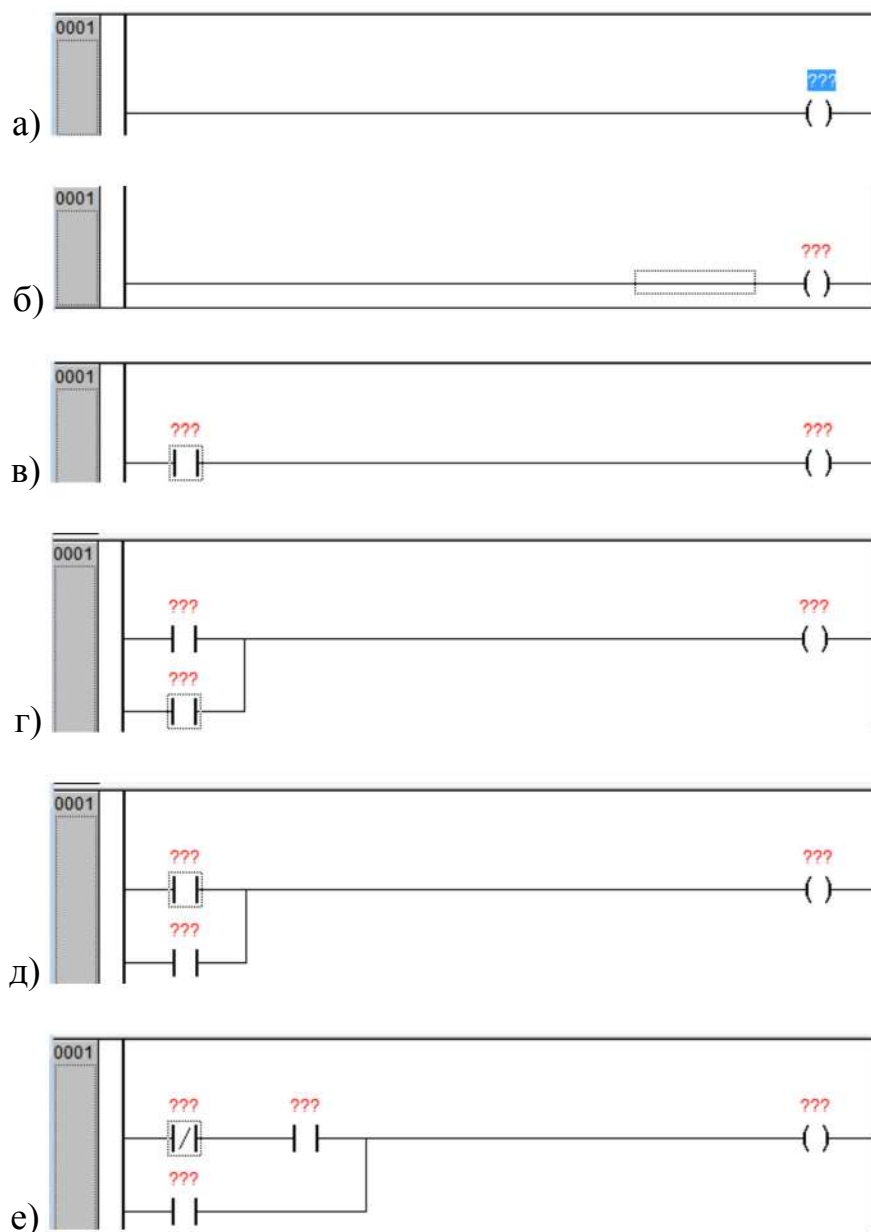


Рисунок 1.7 – Порядок программирования управляющей программы

Шаг 7. Теперь необходимо ввести адреса контактов. Для этого щёлкнуть по знакам «???» над контактом и ввести адрес (ввод адреса завершается нажатием клавиши «ENTER») (рисунок 1.8).

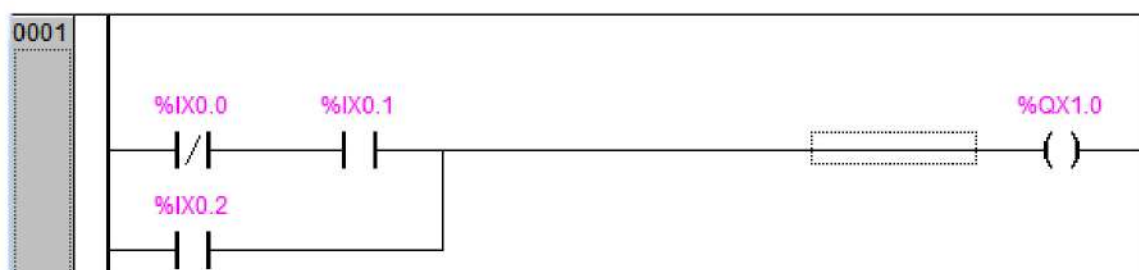


Рисунок 1.8 – Присвоение адресов компонентам программы

Для удобства пользователя при разработке программы доступна символьная адресация. Для этого необходимо предварительно объявить переменные в окне задания переменных проекта следующим образом:

- выбрать элемент на схеме, ввести имя переменной и нажать «*ENTER*»;
- в открывшемся окне выбрать класс, тип, адрес переменной и нажать «ОК» (рисунок 1.9).

После объявления всех входных и выходных переменных программа примет вид, представленный на рисунке 1.10, а окно задания переменных – на рисунке 1.11.

Для удаления какого-либо элемента программы необходимо его выделить и нажать на клавиатуре клавишу *Delete*. Для отмены предыдущего действия необходимо выбрать в меню «Правка» пункт «Отменить».

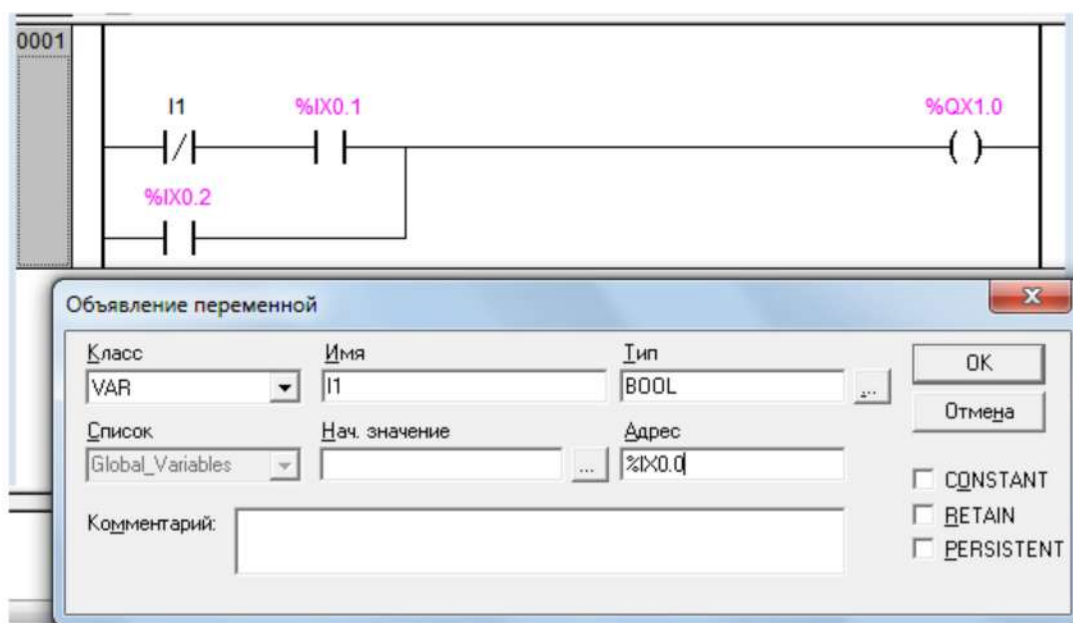


Рисунок 1.9 – Задание переменных

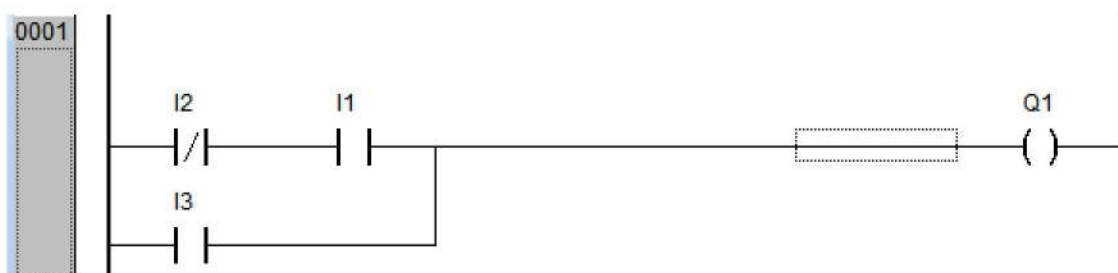


Рисунок 1.10 – Вид программы с использованием обозначенных переменных

```

0001 PROGRAM PLC_PRG
0002 VAR
0003 END_VAR
0004 VAR_INPUT
0005     I1 AT %IX0.0: BOOL;
0006     I2 AT %IX0.1: BOOL;
0007     I3 AT %IX0.2: BOOL;
0008 END_VAR
0009 VAR_OUTPUT
0010     Q1 AT %QX1.0: BOOL;
0011 END_VAR

```

Рисунок 1.11 – Окно задания переменных

В случае некорректного ввода адреса или символического имени программа высвечивает некорректное обозначение красным цветом.

После ввода программы необходимо скомпилировать ее. Для этого в выпадающем меню «Проект» необходимо выбрать пункт «Компилировать», либо нажать клавишу *F11*. Лог компиляции программы будет выведен в окно сообщений. Если программа набрана правильно (рисунок 1.12, а), то последняя строка будет иметь вид: «0 ошибок, 0 предупреждений». В том случае, если в программе будут найдены ошибки (рисунок 1.12, б), то они будут выделены в логе красным цветом. Также в последней строке будет: «X ошибок, X предупреждений». После исправления ошибок необходимо еще раз откомпилировать программу. После компиляции программа готова к загрузке в память ПЛК.

а) Проверка конфигурации задач
Реализация POU 'SFCActionControl'
Реализация POU 'AnalyzeExpression'
Реализация POU 'AppendErrorString'
Реализация POU 'PLC_PRG'
Ревизия конфигурации параметра
Конфигурация аппаратуры
Ошибка 3622: Невозможно создать символьный файл 'UNTITLED.SYM'
Ошибка 3622: Невозможно создать символьный файл 'UNTITLED.SDB'
2 ошибок, 0 предупреждений.

б) Распределение данных
Проверка конфигурации задач
Реализация POU 'PLC_PRG'
Ревизия конфигурации параметра
Конфигурация аппаратуры
POU индексов: 85 (0%)
Использовано данных: 38 из 131072 байт (0.03%)
Энергонезависимых данных: 0 из 4096 байт (0.00%)
0 ошибок, 0 предупреждений.

Рисунок 1.12 – Примеры сообщений после компиляции пользовательской программы

Рассмотрим реализацию логического уравнения (1.1) на других языках:

1) на языке инструкций (*IL*) программа будет иметь достаточно простой и понятный вид

LDN %IX0.1 (загрузить инверсный контакт);

AND %IX0.0 (загрузить контакт);

OR %IX0.2 (загрузить параллельный контакт);

ST %QX1.0 (результат выдать на обмотку);

2) разработка программы на языке функциональных блоковых диаграмм (*FBD*) ведется в следующей последовательности:

– в свойствах проекта настроить язык программирования *FBD*;

– на панели инструментов левой клавишей мыши нажать пиктограмму . На рабочем поле в новом сегменте должно появиться изображение элемента (рисунок 1.13, а), в поле названия элемента указать тип элемента *AND* («И»);

– около входов элемента указать адреса %IX0.0 и %IX0.1 сверху вниз соответственно;

– на входе, соответствующем адресу %IX0.1, нажать правую клавишу мыши и выбрать пункт «Инверсия» (рисунок 1.13, б);

– установить курсор на выходной части элемента «И»;

– левой клавишей мыши нажать пиктограмму . В поле названия элемента указать тип элемента *OR* («ИЛИ»), напротив свободного входа указать адрес %IX0.2 (рисунок 1.13, в);

– установить курсор на выходной части элемента *OR* («ИЛИ»);

– левой клавишей мыши нажать пиктограмму . В сформированном выходе необходимо указать адрес %QX1.0. Программа примет итоговый вид (рисунок 1.13, г).

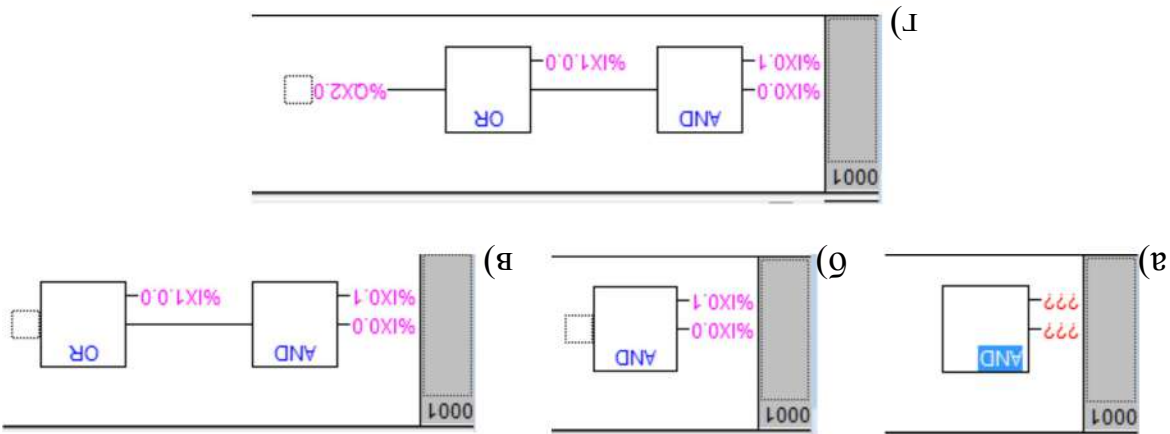
- таймер-формирователь импульса (*TP*);
- таймер с задержкой включения (*TON*);
- таймер с задержкой выключения (*TOF*);
- часы реального времени (*RTC*).

Для программирования контроллера ПЛК150 доступно четыре вида таймеров: ма примет следующий вид (рисунок 1.14, в).

- 4) во входной цепи *RESET* на месте «??» установить адрес *%IX0.1* Программ-
3) в поле «??» над блоком вписать название блока *SR1*;
В сегменте устанавливается блок *SR*-триггера (рисунок 1.14, б);
tion Blocks», в которой выбрать функциональный блок *SR*, после чего нажать «OK».
функциональные блоки». В открывшемся перечне раскрыть закладку «Bistable Func-
ке «Стандартные функциональные блоки» раскрыть библиотеку «Стандартные
струментов нажать пиктограмму . В появившемся окне (рисунок 1.14, а) в заклад-
2) установить курсор в новом сегменте. Левой клавишей мыши на панели ин-
%IX0.0;
1) установить выходную обмотку с адресом *%OX1.0* и контакт с адресом
выключится. Для этого воспользуемся функцией *SR*-триггера:

если кратковременно включить тумблер *I1* (адрес *IX0.0*), то включится выход *Q1* (адрес *QX1.0*), если кратковременно включить тумблер *I2* (адрес *IX0.1*), то выход *Q1* выключится. Для этого воспользуемся функцией *SR*-триггера:

Рисунок 1.13 – Программа на языке FBD



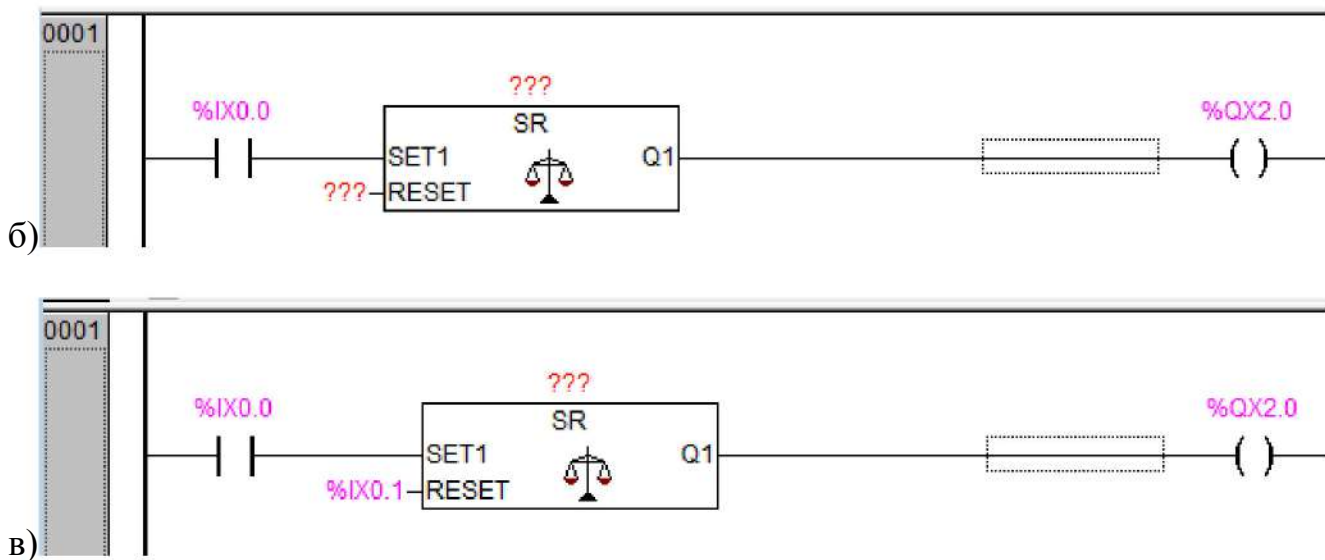
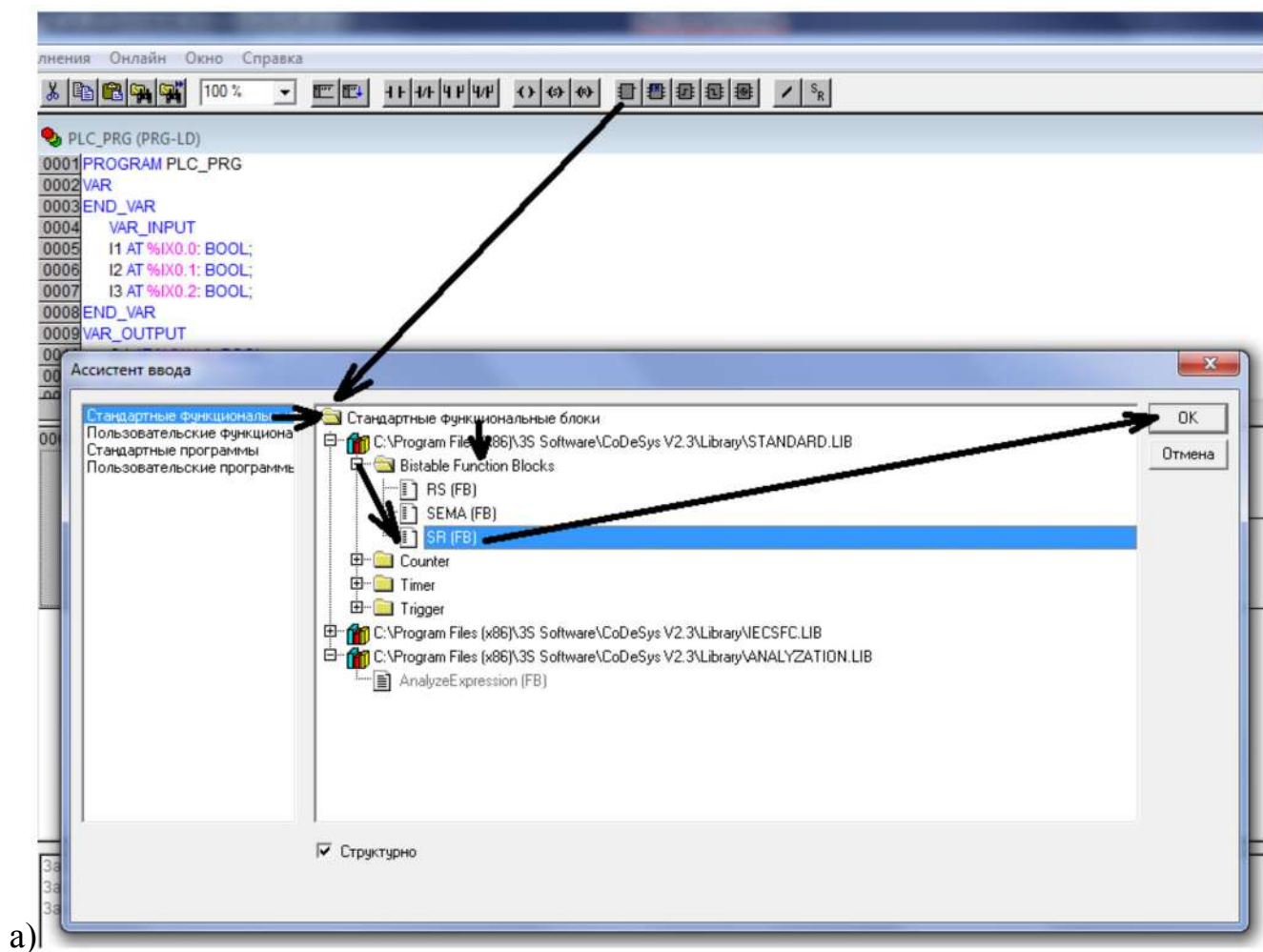


Рисунок 1.14 – Пример использования функциональных блоков

В таблице 1.3 приведены примеры реализации таймеров на языках *IL* и *ST*.

На рисунке 1.15 представлено графическое изображение блоков таймеров, ис-

пользуемое в языке *FBD*. Блок, реализующий таймер, имеет следующие входы и выходы: *IN* – вход запуска, *EN* – вход разрешения, *PT* – уставка (заданное время), *Q* – выход (состояние таймера), *ET* – оставшееся время.

Таблица 1.3 – Примеры реализации таймеров на языках *IL* и *ST*

Вид таймера	Пример объявления	Пример <i>IL</i>	Пример <i>ST</i>
<i>TP</i>	<i>TPInst: TP</i>	<i>CAL TPInst</i> (<i>IN:=VarBOOL1, PT:=T#5s</i>) <i>LD TPInst.Q</i> <i>ST VarBOOL2</i>	<i>TPInst (IN:=VarBOOL1,</i> <i>PT:=T#5s)</i> <i>VarBOOL2:=TPInst.Q</i>
<i>TON</i>	<i>TONInst: TON</i>	<i>CAL TONInst</i> (<i>IN:=VarBOOL1, PT:=T#5s</i>) <i>LD TONInst.Q</i> <i>ST VarBOOL2</i>	<i>TONInst</i> (<i>IN:=VarBOOL1,</i> <i>PT:=T#5s)</i>
<i>TOF</i>	<i>TOFInst: TOF</i>	<i>CAL TOFInst</i> (<i>IN:=VarBOOL1, PT:=T#5s</i>) <i>LD TOFInst.Q</i> <i>ST VarBOOL2</i>	<i>TOFInst</i> (<i>IN:=VarBOOL1,</i> <i>PT:=T#5s)</i> <i>VarBOOL2:=TOFInst.Q</i>

Для задания уставки времени используется идентификатор *T#*.

Уставку времени можно задать в *BCD*-коде с использованием следующего синтаксиса: *T#AAd_BBh_CCm_DDs_EEEms*, где *AA* = сутки, *BB* = часы, *CC* = минуты, *DD* = секунды и *EEE* = миллисекунды.

Каждый таймер использует структуру, хранящуюся в блоке данных, для сохранения данных о времени. При этом назначается блок данных, когда устанавливается команда таймера.

В режиме *RUN* текущее значение таймера уменьшается на одну единицу через интервал времени, установленный базой времени, до тех пор, пока значение времени не станет равным нулю, что соответствует срабатыванию таймера.

Для программирования контроллера ПЛК150 доступно три вида счетчиков:

- инкрементный (на сложение) счетчик (*CTU*);
- декрементный (на вычитание) счетчик (*CTO*);
- инкрементный/декрементный (реверсивный) счетчик (*CTUD*).

В таблице 1.4 приведены примеры реализации счетчиков на языках *IL* и *ST*.

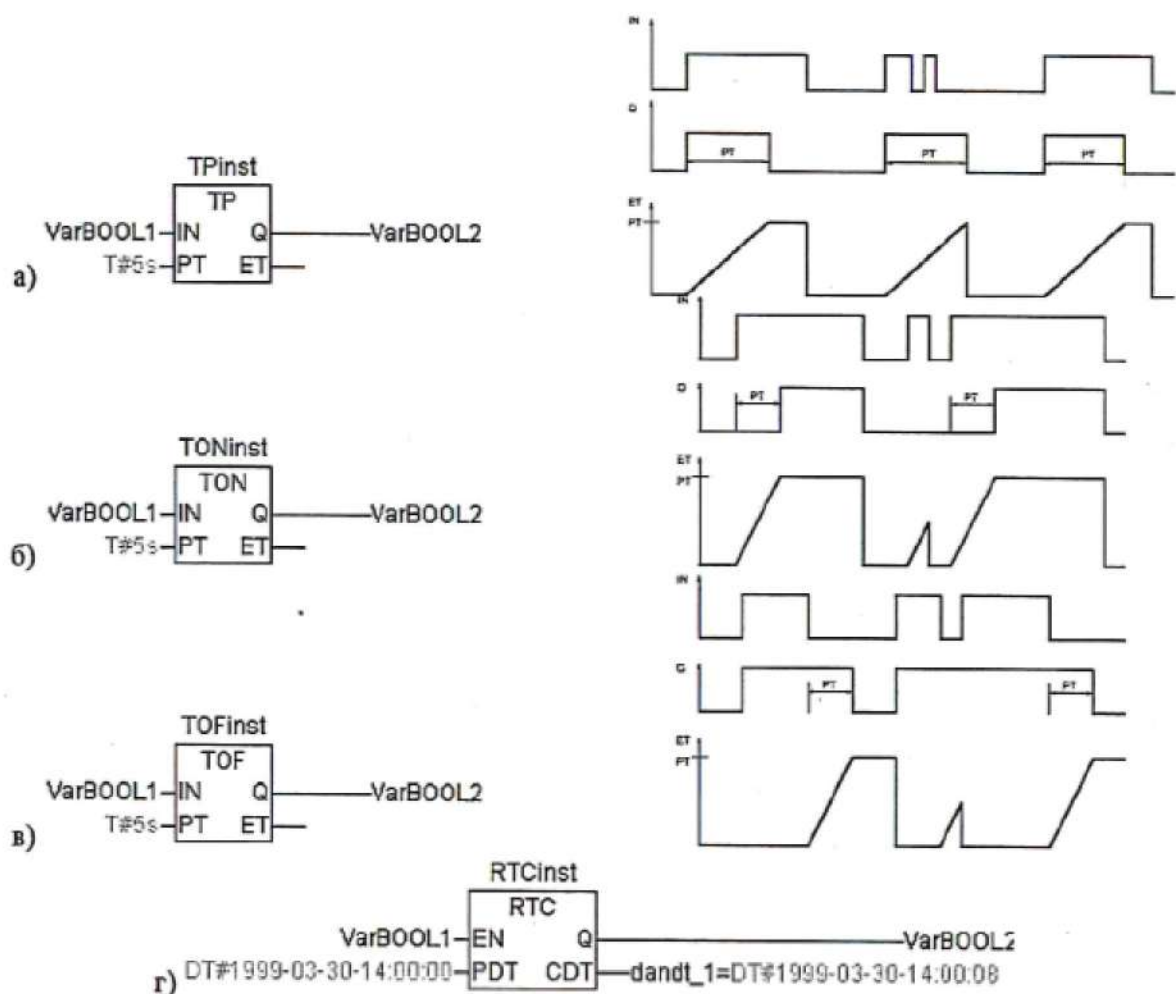


Рисунок 1.15 – Таймеры и их временные диаграммы работы

На рисунке 1.16 представлено графическое изображение блоков счетчиков, используемое в языке *FBD*. Блок, реализующий счетчик, имеет следующие входы и выходы: *CU* – вход на сложение, *CD* – вход на вычитание, *PV* – задание (уставка) счетчика, *RESET* – вход сброса, *LOAD* – вход загрузки задания счетчика, *Q*, *QU*, *QD* – выход (состояние) счетчика, *CV* – текущее значение счетчика в двоичном формате.

Уставка счетчика задается либо константой в формате <значение>, либо из области памяти (*I*, *Q*, *M*). Нарастающим фронтом сигнала на входе *LOAD* в счетчик заносится уставка с входа *PV*. При нарастающем сигнале на входе *CU* значение счетчика увеличивается на «1». При нарастающем фронте сигнала на входе *CD* значение счетчика уменьшается на «1», если значение счетчика больше, чем «0». При нарастающем фронте сигнала на входе *RESET* счетчик сбрасывается (в счетчик записывается значение «0»).

Таблица 1.4 – Примеры реализации счетчиков на языках *IL* и *ST*

Счетчик	Пример объявления	Пример IL	Пример ST
<i>CTU</i>	<i>CTUInst:</i> <i>CTU</i>	<i>CAL CTUInst(CU:=VarBOOL1,</i> <i>RESET:=VarBOOL2,</i> <i>PV:=VarINT1)</i> <i>LD CTUInst.Q</i> <i>ST VarBOOL3</i> <i>LD CTUInst.CV</i> <i>ST VarINT2</i>	<i>CTUInst(CU:=VarBOOL1,</i> <i>RESET:=VarBOOL2,</i> <i>PV:=VarINT1)</i> <i>VarBOOL3:=CTUInst.Q;</i> <i>VarINT2:=CTUInst.CV;</i>
<i>CTD</i>	<i>CTDInst:</i> <i>CTD</i>	<i>CAL CTDInst(CD:=VarBOOL1,</i> <i>LOAD:=VarBOOL2,</i> <i>PV:=VarINT1)</i> <i>LD CTDInst.Q</i> <i>ST VarBOOL3</i> <i>LD CTDInst.CV</i> <i>ST VarINT2</i>	<i>CTDInst(CD:=VarBOOL1,</i> <i>LOAD:=VarBOOL2,</i> <i>PV:=VarINT1)</i> <i>VarBOOL3:=CTDInst.Q;</i> <i>VarINT2:=CTDInst.CV;</i>
<i>CTUD</i>	<i>CTUDInst:</i> <i>CTUD</i>	<i>CAL CTUDInst(CU:=VarBOOL1,</i> <i>RESET:=VarBOOL3,</i> <i>LOAD:=VarBOOL4,</i> <i>PV:=VarINT1)</i> <i>LD CTUDInst.Q</i> <i>ST VarBOOL5</i> <i>LD CTUDInst.QD</i> <i>ST VarBOOL6</i> <i>LD CTUDInst.CV</i> <i>ST VarINT2</i>	<i>CTUDInst(CU:=VarBOOL1,</i> <i>RESET:=VarBOOL3,</i> <i>LOAD:=VarBOOL4,</i> <i>PV:=VarINT1)</i> <i>VarBOOL5:=CTUDInst.Q;</i> <i>VarBOOL6:=CTUDInst.QD;</i> <i>VarINT2:=CTUDInst.CV;</i>

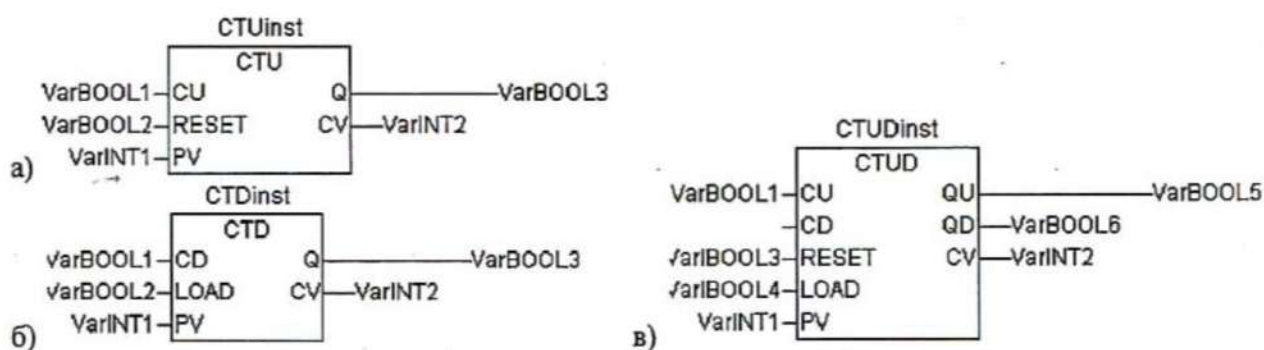


Рисунок 1.16 – Счетчики

Значение реверсивного счетчика остается неизменным, если оба счетных входа имеют нарастающий фронт.

В счетчике *CTU* выход *Q* устанавливается в «1», когда значение СП достигает значения *PV*. В счетчике *CTD* выход *Q* устанавливается в «1», когда значение *CV*

достигает «0».

В счетчике *CTUD* выход *QU* устанавливается в «1», когда значение *CV* больше или равно значению *PV*, а выход *QD* – когда значение *CV* достигает «0».

Кроме таймеров и счетчиков в программируемом контроллере ОВЕН ПЛК150 используется стандартный набор операндов и функций, присущий всем контроллерам:

- логические;
- арифметические;
- операции сдвига;
- операции сравнения и т.д.

1.3 Программирование функциональных блоков и функций

Среда разработки *CoDeSys 2.3* предусматривает возможность упростить программирование, если возникает необходимость в программировании управления однотипными технологическими операциями. Например, рассматриваемые ниже в качестве примера испытания бензинового и дизельного двигателей. Испытания в принципе одинаковые, только аппараты (кнопки, датчики и устройства индикации) свои. В этом случае необходимо запрограммировать функциональный блок *FB*, который может многократно вызываться в программе.

Например, создадим функциональный блок *FB1*, которому дадим символическое имя «*ENGINE* [двигатель]». При кратковременном появлении команды «*turn_on* [включить]» двигатель должен включиться и сформировать сигнал «*on* [включен]». При достижении фактической скорости «*fact* [факт]» заданного значения «*preset* [задание]» формируется сигнал «*ready* [готов]». Двигатель отключается по сигналу «*turn_off* [отключить]» или по сигналу «*trouble* [неисправность]».

Последовательность программирования сводится к следующим этапам.

Этап 1. Создать новый проект, в котором создать три программных компонента *POU*: программный блок *PLC_PRG*, функциональный блок *ENGINE* и функция *VENT* (рисунок 1.17).

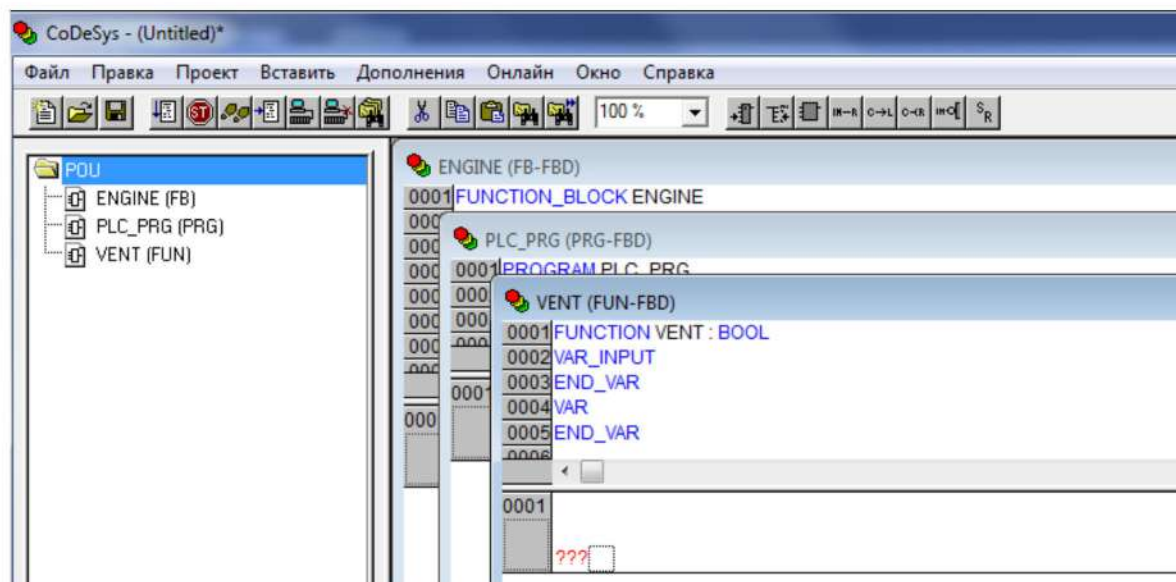


Рисунок 1.17 – Создание программных компонентов

Этап 2. Объявить переменные для функционального блока *ENGINE* (рисунок 1.18, а):

FUNCTION_BLOCK ENGINE

VAR_INPUT

turn_on: BOOL;

turn_off: BOOL;

trouble: BOOL;

fact: INT;

preset: INT;

END_VAR

VAR_OUTPUT

on: BOOL;

ready: BOOL;

END_VAR

VAR

END_VAR .

Этап 3. Объявить переменные для основной программы *PLC_PRG* (таблица 1.5 и рисунок 1.18, б).

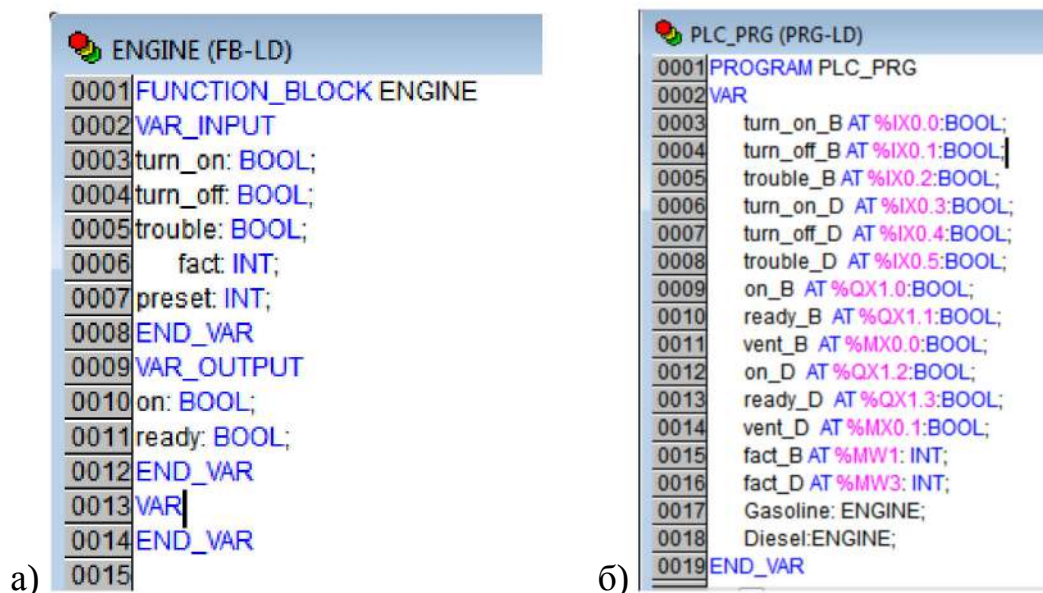


Рисунок 1.18 – Объявление переменных

Таблица 1.5 – Переменные для основной программы *PLC_PRG*

Символическое имя	Адрес на модуле	Адрес ПЛК	Тип данных	Комментарии
Входные переменные				
<i>turn_on_B</i>	<i>I1</i>	<i>%IX0.0</i>	<i>BOOL</i>	Включить бензиновый
<i>turn_off_B</i>	<i>I2</i>	<i>%IX0.1</i>	<i>BOOL</i>	Отключить бензиновый
<i>trouble_B</i>	<i>I3</i>	<i>%IX0.2</i>	<i>BOOL</i>	Неисправность бензинового
<i>turn_on_D</i>	<i>I4</i>	<i>%IX0.3</i>	<i>BOOL</i>	Включить дизель
<i>turn_off_D</i>	<i>I5</i>	<i>%IX0.4</i>	<i>BOOL</i>	Отключить дизель
<i>trouble_D</i>	<i>I6</i>	<i>%IX0.5</i>	<i>BOOL</i>	Неисправность дизеля
Выходные переменные				
<i>on_B</i>	<i>Q1</i>	<i>%QX1.0</i>	<i>BOOL</i>	Бензиновый включен
<i>ready_B</i>	<i>Q2</i>	<i>%QX1.1</i>	<i>BOOL</i>	Заданная скорость достигнута
<i>vent_B</i>	<i>Q3</i>	<i>%MX0.0</i>	<i>BOOL</i>	Вентилятор бензинового включен
<i>on_D</i>	<i>Q5</i>	<i>%QX1.2</i>	<i>BOOL</i>	Дизель включен
<i>ready_D</i>	<i>Q6</i>	<i>%QX1.3</i>	<i>BOOL</i>	Заданная скорость достигнута
<i>vent_D</i>	<i>Q7</i>	<i>%MX0.1</i>	<i>BOOL</i>	Вентилятор дизеля включен
Промежуточные переменные				
<i>fact_B</i>		<i>MW1</i>	<i>INT</i>	Фактическая скорость бензинового двигателя
<i>fact_D</i>		<i>MW3</i>	<i>INT</i>	Фактическая скорость дизеля
Таймеры				
<i>tim_B</i>		<i>TON</i>		Таймер вентилятора бензинового
<i>tim_D</i>		<i>TON</i>		Таймер вентилятора дизеля

Этап 4. Разработать программу, управляющую включением и отключением

двигателя, как описано выше.

Набранная программа на языке лестничных диаграмм имеет вид, представленный на рисунке 1.19.

Созданный блок *ENGINE* должен управлять и контролировать работу бензинового и дизельного двигателей. Различные заданные скорости хранятся в отдельных ячейках памяти.

Этап 5. Запрограммировать вызов функционального блока в программном блоке. Для этого в сегменте программного блока необходимо установить курсор в заданном положении, на панели инструментов нажать пиктограмму  и установить 2 блока на поле сегмента. В поле названия блока необходимо списать название пользовательского блока (*ENGINE*). Один блок необходимо назвать «*Gasoline*», другой – «*Diesel*». Соответственно один блок будет отвечать за работу бензинового двигателя, а второй – за работу дизельного (рисунок 1.20, а). В поля напротив входов и выходов необходимо установить названия соответствующих переменных. Программный блок после этого примет вид, представленный на рисунок 1.20, б.

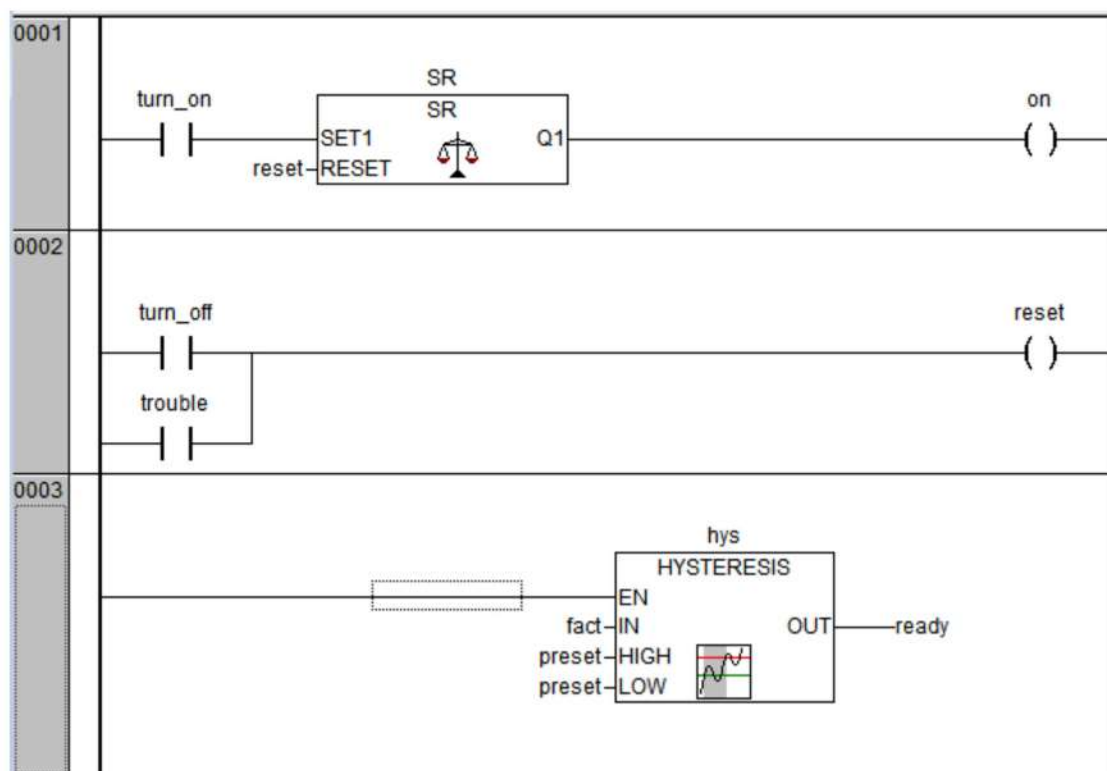


Рисунок 1.19 – Функциональный блок *ENGINE*

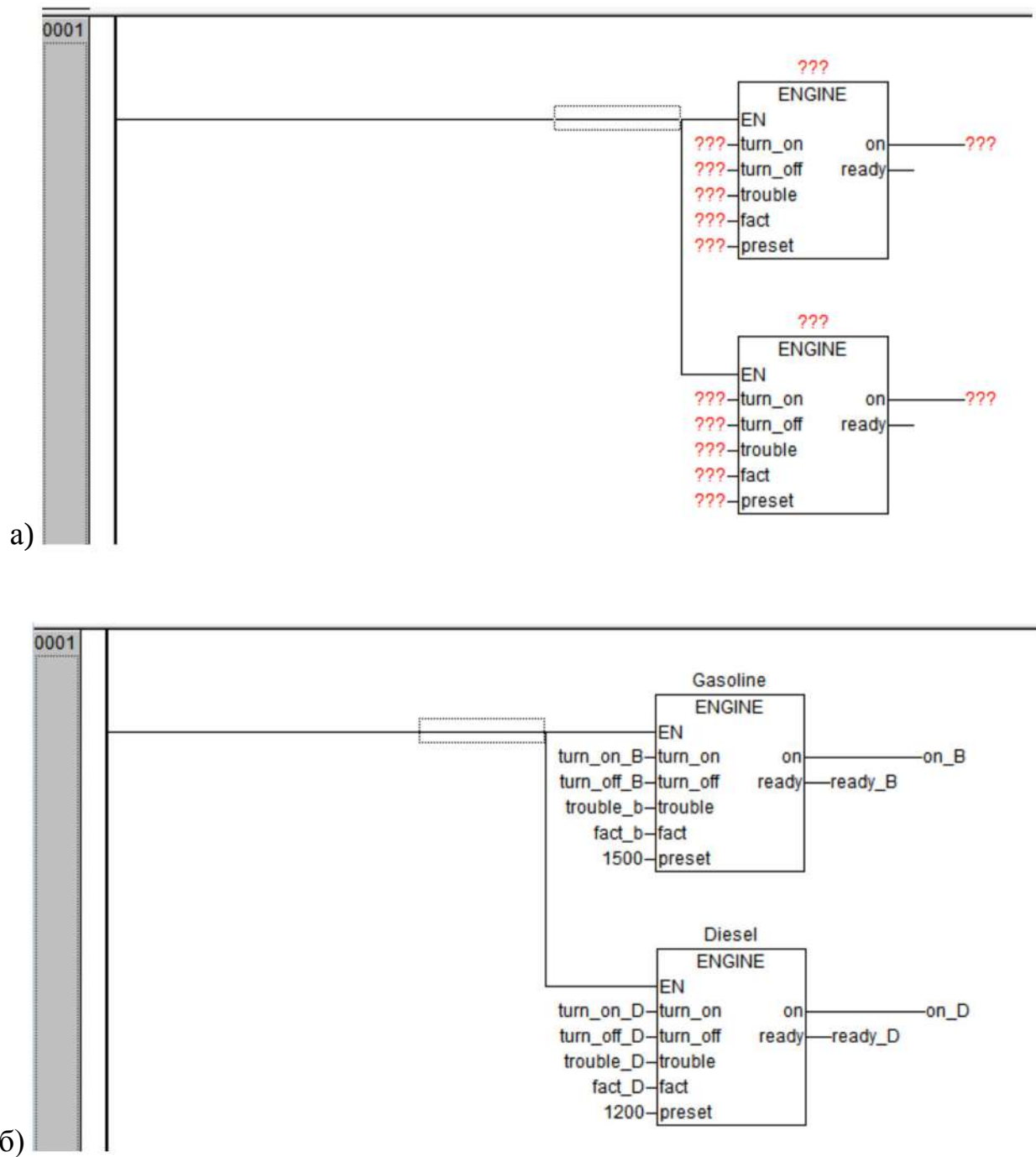


Рисунок 1.20 – Сегмент программы управления испытаниями двигателей

Напротив внутренней переменной «*preset*» блока «*Gasoline*» необходимо установить значение 1500. Таким образом, определена максимальная скорость для бензинового двигателя. Аналогично установить значение 1200 напротив внутренней переменной «*preset*» блока «*Diesel*» (для дизельного двигателя).

Этап 6. Сохранить программу.

Этап 7. Запрограммировать управление двигателями.

Функции, как и функциональные блоки, расположены в иерархии программы ниже программного блока. Однако в отличие от функционального блока, функции не нужен блок данных. У функций параметры также перечисляются в окне объявления переменных, но статические данные не разрешаются.

В рассматриваемой задаче предусматривается, что каждый двигатель как бензиновый, так и дизель имеет свой вентилятор. Вентилятор включается при работе двигателя и продолжает еще работать 4 с после отключения двигателя. Для управления этими двигателями программируется функция *VENT*.

Необходимо в «Менеджере проекта» дважды щелкнуть на строке *VENT*, чтобы открыть ее. Открывается окно для программирования. Далее нужно объявить переменные:

```
FUNCTION VENT: BOOL
VAR_INPUT
    tum_on: BOOL;
END_VAR
VAR
    tim: TON;
END VAR
```

Этап 8. Запрограммировать управление вентилятором. Набранная программа на языке *FBD* имеет вид, представленный на рисунке 1.21.

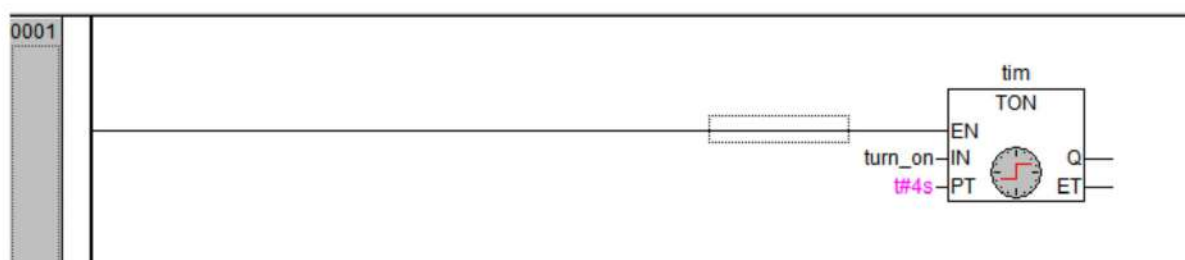


Рисунок 1.21 – Программирование функции *VENT*

Этап 9. Вызов функции *VENT* в программном блоке выполняется аналогично вызову функционального блока. Все переменные функции снабжаются в программ-

ном блоке соответствующими адресами бензинового или дизельного двигателя, которые уже отражены в таблице 1.5.

В программном блоке *PLC_PRG* вставить новый сегмент. Затем в этом сегменте необходимо установить курсор, на панели инструментов нажать пиктограмму  и установить два блока на поле сегмента. В поле названия блока необходимо списать название пользовательской функции (*VENT*). Одна функция будет отвечать за работу вентилятора бензинового двигателя, а вторая – за работу вентилятора дизельного двигателя (рисунок 1.22, а). Далее в соответствующих полях ввести символьные имена из таблицы 1.5, чтобы программа приняла вид, соответствующий рисунку 1.22, б. Сохранить программу.

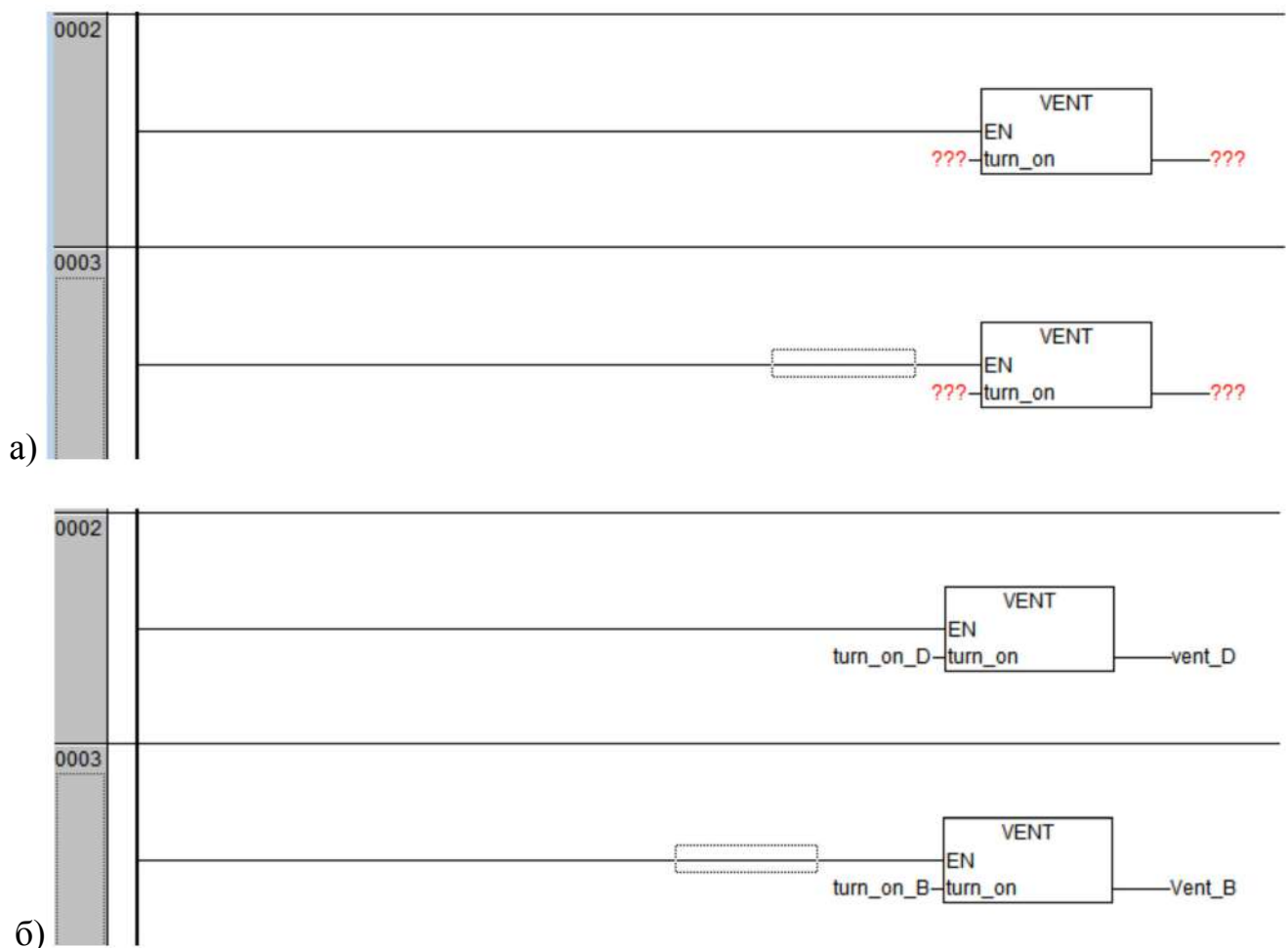


Рисунок 1.22 – Сегмент программы управления вентилятором

1.4 Лабораторная работа 1 «Изучение технических характеристик и основ программирования контроллера ОВЕН ПЛК150»

Цель работы: ознакомиться с устройством и техническими характеристиками программируемого контроллера ОВЕН ПЛК150, приобрести навыки программирования с применением программного обеспечения *CoDeSys 2.3*, изучить основные функции и операции.

Задание на выполнение работы:

- изучить назначение, технические характеристики контроллера ПЛК150, основные узлы и возможности лабораторного стенда;
- изучить принципы программирования ОВЕН ПЛК150;
- выполнить синтез системы автоматизации согласно выданного варианта задачи;
- составить программу для ввода в контроллер;
- пройти тестирование по основам программирования контроллера ПЛК150;
- освоить графическую среду программного обеспечения *CoDeSys 2.3*;
- сконфигурировать оборудование;
- набрать на ноутбуке подготовленную программу, откомпилировать ее и ввести в контроллер, убедиться в правильности работы программы.

При выполнении лабораторной работы предполагается индивидуализация обучения по принципу «каждому студенту (бригаде студентов) свой вариант задания». Поэтому в лабораторной работе предусмотрены варианты индивидуальных заданий. По своему усмотрению преподаватель может упростить или усложнить задачу варианта в зависимости от подготовки обучаемого. Выполнение лабораторной работы рассчитано на четыре академических часа работы в лаборатории и 3–4 часа самостоятельной работы для домашней подготовки и оформления отчета по работе.

1.4.1 Техническое описание лабораторного стенда

Лабораторный стенд «Система автоматического управления ОВЕН «САУ-ОВЕН» [11] (далее стенд) предназначен для обучения студентов различных

специальностей, изучающих дисциплины, связанные с автоматизацией различных отраслей промышленности. Стенд позволяет изучить:

- технические характеристики и основы программирования промышленных средств автоматизации (программируемого логического контроллера, панели оператора, программируемого реле, преобразователя частоты);
- основы построения систем управления с использованием распределенной периферии;
- основы реализации систем автоматизации на базе современных средств автоматизации.

Стенд выполнен для работы в лабораторных условиях (невзрывоопасная окружающая среда, не содержащая агрессивных газов и паров, ненасыщенная водяными парами и токопроводящей пылью).

1.4.2 Общие технические характеристики

Общие технические характеристики лабораторного стенда представлены в таблице 1.6. Внешний вид лабораторного стенда приведен на рисунке 1.23.

Стенд выполнен в настольном исполнении в стоечном варианте и включает в себя ноутбук, стойку с семью съемными модулями и комплект вспомогательного оборудования.

Таблица 1.6 – Основные технические характеристики стенда

Параметр	Значение
Напряжение электрической плиты, В	1×220
Частота напряжения питающей сети, Гц	50
Потребляемая мощность, не более, ВА	500
Габаритные размеры, мм, Ш×В×Г	850×610×300
Масса, кг	25
Диапазон рабочих температур, °С	От +10 до +35
Относительная влажность воздуха, %	80
Количество проводимых лабораторных работ	12

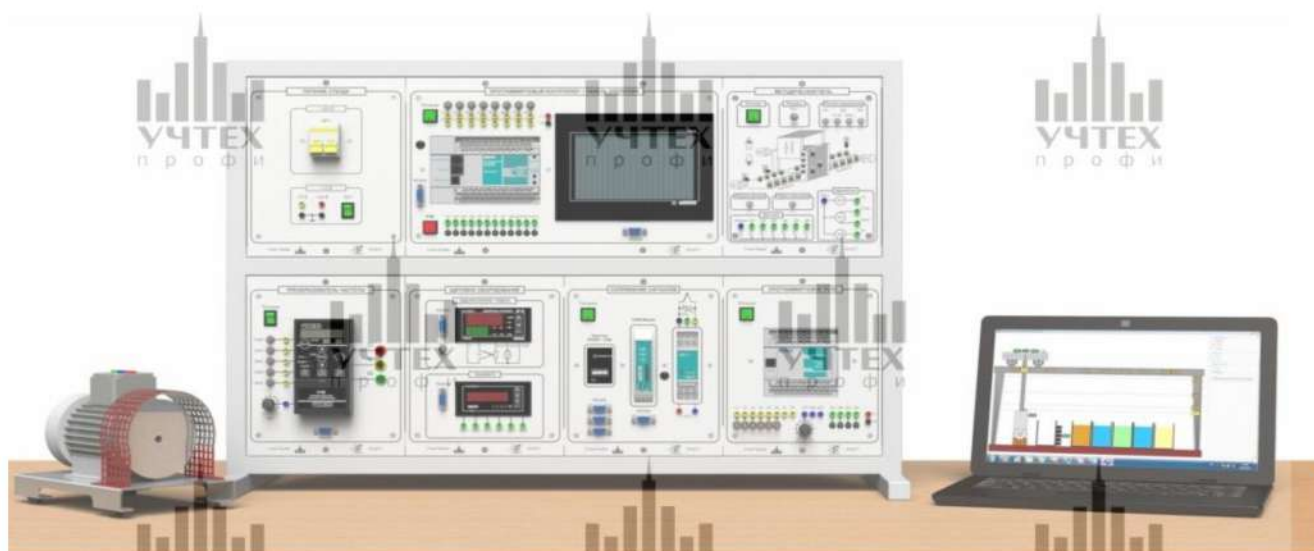


Рисунок 1.23 – Внешний вид стенда САУ-ОВЕН

Все модули выполнены по однотипной конструкции. Каждый модуль состоит из лицевой панели и защитного кожуха. К лицевой панели крепится исследуемое оборудование. Лицевые панели содержат различную полезную для пользователя информацию (мнемосхемы, название модуля, подписи к различной аппаратуре и др.). Разъемы, необходимые для обеспечения работы стенда, вынесены на задние стенки кожухов.

1.4.3 Состав лабораторного стенда

В силу своего назначения стенд содержит основные электромеханические узлы, силовые преобразователи, измерительные элементы, а также другое механическое и электрическое оборудование, охватывающее основные разделы курсов, посвященных изучению различного оборудования, входящего в состав систем промышленной автоматизации.

Лабораторный стенд содержит следующие основные элементы:

- модуль питания стенда (ПС);
- модуль программируемого контроллера и панели оператора (ПЛКиПО);
- модуль имитации физического объекта «Методическая печь» (МП);
- модуль преобразователя частоты (ПЧ);
- модуль щитового оборудования (ЩО);

- модуль согласования сигналов (СС);
- модуль программируемого реле (ПР);
- каркас 2х4;
- электромашинный агрегат;
- ноутбук;
- комплект соединительных проводов и кабелей;
- сопроводительная документация, программное и методическое обеспечение.

1.4.4 Схема подключения

Схема подключения силового напряжения и слаботочного питания между модулями с тыльной стороны стенда показана на рисунке 1.24. Перечень соединительных проводов представлен в таблице 1.7.

Данные подключения производятся однократно при наладке стенда, в дальнейшем при проведении лабораторных работ необходимости переключать их не возникает. Подключения на лицевой стороне стенда производятся непосредственно перед выполнением конкретной лабораторной работы.

Таблица 1.7 – Перечень соединительных проводов

Тип	Примечание
<i>Кабели силового и низковольтного питания</i>	
1 Сетевой кабель питания	Для подачи однофазного питания 220 В на стенд
2 Межмодульный кабель питания	Для подачи однофазного питания 220 В на модули стенда
3 Межмодульный кабель питания DBF-DB9 М	Для подачи низковольтного питания +24 В, +5 В
4 Удлинитель с розеткой на 220 В	Для подачи питания на стенд и ноутбук
<i>Информационные кабели</i>	
5 Кабель USB А – USB В	Для связи модулей с ПК через интерфейс USB

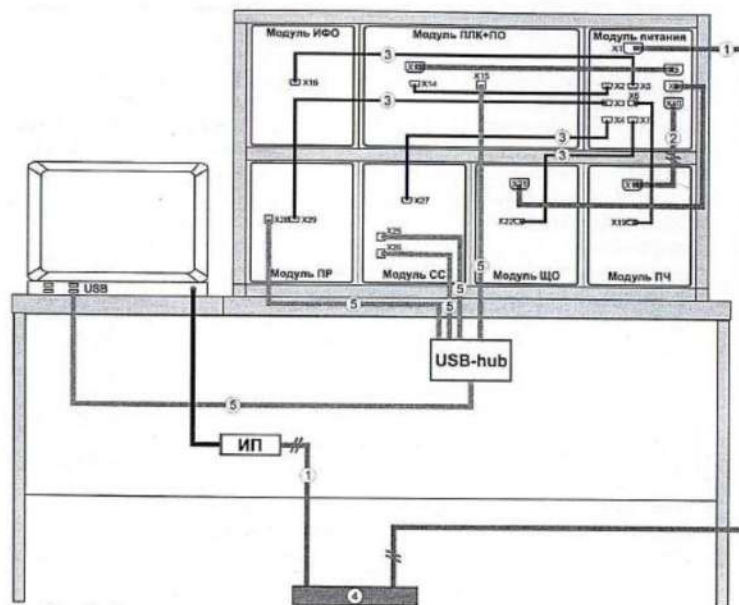


Рисунок 1.24 – Схема подключения лабораторного стенда (вид с тыльной стороны)

1.4.5 Особенности выполнения требований техники безопасности при эксплуатации

Лабораторный стенд содержит в своем составе цепи напряжением 220 В, а также вращающиеся части электромашинного агрегата, тем самым является источником повышенной опасности. К работе со стендом не допускаются лица, не ознакомившиеся с техникой безопасности при проведении лабораторных занятий. Функциональные возможности стенда и его долговечность требуют соблюдения мер по эксплуатации, направленных на безопасность проведения исследований и обеспечивающих долговечность работы самого стенда:

- 1) запрещается подключать цепи электромашинного агрегата к сторонним источникам питания, не предусмотренным составом стенда, так как это может привести к выходу из строя электрической машины;
- 2) запрещается перегружать элементы стенда повышенными токами и напряжениями;
- 3) для измерительных приборов и датчиков запрещается превышать измеряемый диапазон;
- 4) при срабатывании защиты или возникновении сигнала о защите в одном или нескольких модулях необходимо незамедлительно прекратить выполнение ла-

бораторной работы и обесточить лабораторный стенд; при следующем включении необходимо ликвидировать причину, вызвавшую срабатывание защиты;

5) сборку и разборку схемы, переключение проводов, а также некоторые оперативные переключения, неоговоренные в учебном пособии, необходимо выполнять исключительно при выключенном питании;

6) стенд имеет различные уровни слаботоочного напряжения: +5 В, +24 В. При неправильном подключении слаботоочных напряжений возможен выход из строя оборудования.

1.4.6 Общие сведения об изучаемом оборудовании

Ниже представлено краткое описание аппаратной части, используемой в данной лабораторной работе. Все необходимое оборудование расположено в модуле ПЛК+ПО, внешний вид на рисунке 1.25.

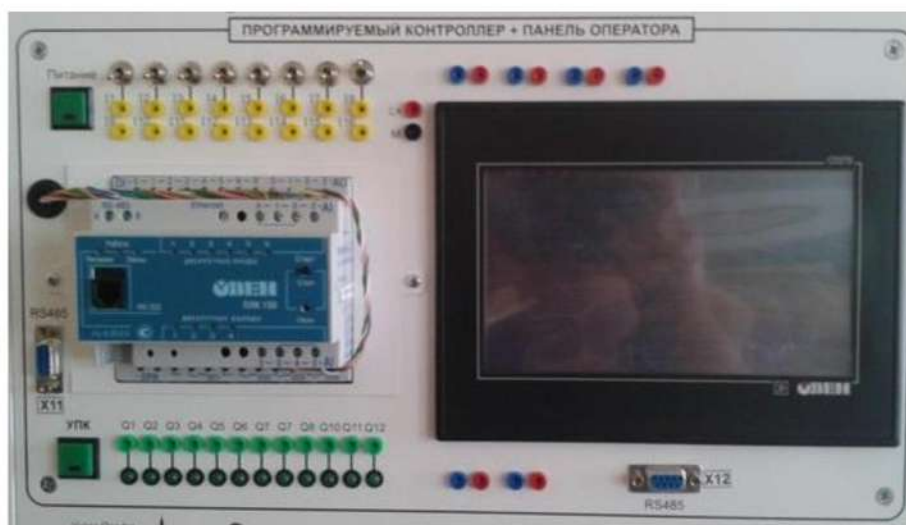


Рисунок 1.25 – Модуль программируемого логического контроллера и панели оператора (модуль ПЛК+ПО)

Программируемый логический контроллер (ПЛК) ОВЕН ПЛК150, представляющий собой моноблочный контроллер, имеет в своем составе мощный микропроцессор, встроенные дискретные входы и выходы, порты *RS232*, *RS485*, *USB*, *ETHERNET*. Кроме этого он отличается компактной конструкцией и гибкой конфигурацией. Все это обеспечивает универсальность и достаточную мощ-

ность для управления различными устройствами автоматизации, и делает контроллер ОВЕН ПЛК150 прекрасным решением для управления широким спектром приложений.

Контроллер ПЛК150 может быть использован как:

- специализированное устройство управления выделенным локализованным объектом;
- устройство мониторинга локализованного объекта в составе комплексной информационной сети;
- специализированное устройство управления и мониторинга группой локализованных объектов в составе комплексной информационной сети.

Внешний вид ПЛК представлен на рисунке 1.26.

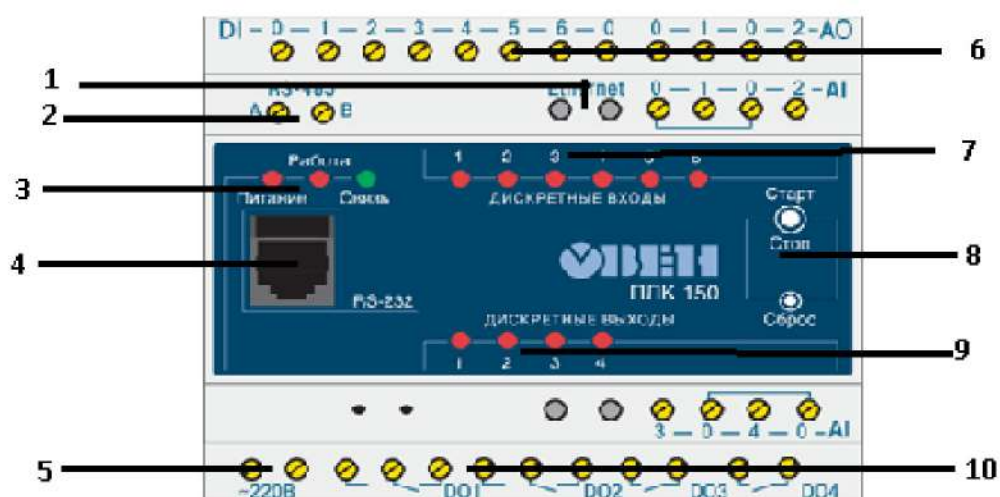


Рисунок 1.26 – Внешний вид ОВЕН ПЛК150

Описание основных элементов ПЛК представлено в таблице 1.8.

Таблица 1.8 – Описание основных элементов ПЛК

Элемент	Функциональное назначение
1	2
Разъем Ethernet	Подключение к среде программирования CoDeSys 2.3 Подключение к соответствующей информационной сети для связи с ПК, ПЛК или другими устройствами автоматизации
RS485	
RS232	

Продолжение таблицы 1.8

1		2
Индикаторы состояния ПЛК	Связь	Отображает наличие связи со средой программирования <i>CoDeSys 2.3</i>
	Питание	Отображает наличие питания контроллера
	Работа	Отображает выполнение пользовательской программы
Клеммный блок		Подключение питания контроллера
Клеммный блок		Подключение входных цепей контроллера
Индикаторы состояния входов		Индикация состояния входов программируемого контроллера
Кнопки управления	СТАРТ/СТОП	Предназначена для запуска и остановки программы в контроллере
	F1	Предназначена для перезагрузки контроллера (нажимается тонким заостренным предметом)
Индикаторы состояния выходов		Индикация состояния выхода программируемого контроллера
Клеммный блок		Подключение выходных цепей контроллера

Технические характеристики программируемого логического контроллера ОВЕН ПЛК150 представлены в таблице 1.9.

Таблица 1.9 – Технические характеристики ОВЕН ПЛК150

Параметр		Значение
1		2
Модель		ПЛК 150-220. А-L
Напряжение питания переменного напряжения, В		От 90 до 264
Потребляемая мощность, Вт		6
Напряжение и выходной ток встроенного источника		24 В постоянного тока, 0,62 А
Встроенные цифровые входы/выходы		6/4
Встроенные аналоговые входы/выходы		4/2
Языки программирования		Все языки по МЭК 61131-3
Память	Пользовательская оперативная, Мб	8
	Пользовательская постоянная (для хранения архивов и файлов), Мб	4
	Retain, Кб	4
Время выполнения цикла ПЛК (типовое), мс		1
Время выполнения цикла ПЛК (минимальное настраивание), мкс		250

Продолжение таблицы 1.9

1	2
Интерфейсы связи	RS232, RS485, OВЕН, Ethernet 100 Base-T
Протоколы передачи данных	ModBus-RTU, ModBus-ASCII, DCON, ModBus-TCP, Gateway
Среда программирования, не ниже	CODESYS 2.3.8.1
Дополнительное оборудование	Часы реального время, встроенный источник звукового сигнала, функциональная кнопка

Ниже представлены схемы подключения входных и выходных цепей ОВЕН ПЛК150 (рисунок 1.27).

Технические характеристики входных и выходных цепей контроллера представлены в таблице 1.10.

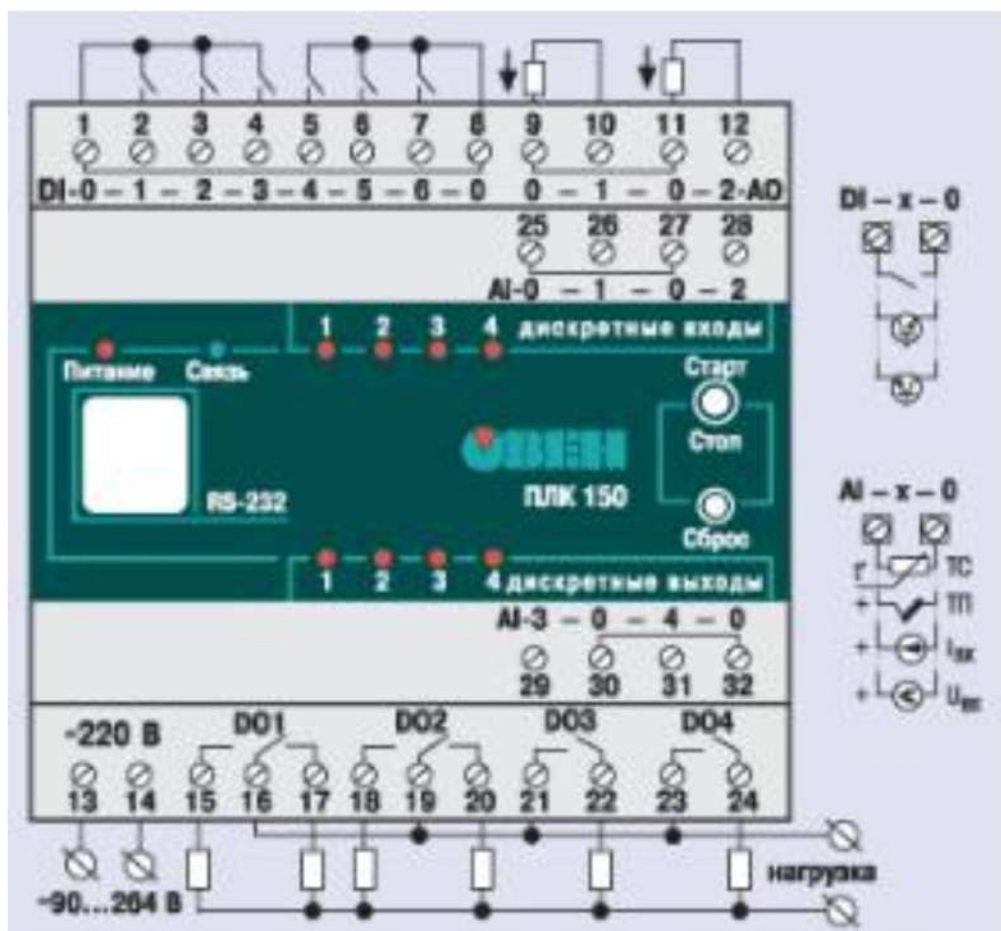


Рисунок 1.27 – Схема подключения питания, входов и выходов к ОВЕН ПЛК150

Таблица 1.10 – Технические характеристики входных и выходных цепей

Параметр		Значение
<i>Дискретные входы</i>		
Число дискретных входов		6
Тип		Сухой контакт
Максимальная частота сигнала, подаваемого на дискретный вход, кГц	При программной обработке	1
	При применении аппаратного счетчика и обработчика энкодера	10
Гальваническая изоляция дискретных входов		Есть, групповая
<i>Дискретные выходы</i>		
Число выходов		4
Тип		реле
Ток коммутации (макс.), А		2
Гальваническая изоляция дискретных выходов		Есть, индивидуальная
<i>Аналоговые входы</i>		
Число входов		4
Тип	Напряжение, В	От 0 до 1, от 0 до 10
	Ток, мА	От 0 до 5, от 0(4) до 20
	Сопротивление, кОм	От 0 до 5
Разрядность встроенного АЦП, бит		16
Время опроса одного аналогового входа, с		1,5
Гальваническая изоляция аналоговых входов		нет
<i>Аналоговые выходы</i>		
Число выходов	4	
Тип	Напряжение, В	От 0 до 10
	Ток, мА	От 4 до 20
Разрядность встроенного АЦП, бит		10
Гальваническая изоляция аналоговых выходов		Есть, групповая

Для имитации пульта оператора используется шесть трехпозиционных тумблеров I1-I6, подключенных к входам ПЛК. Индикация включенного состояния входных и выходных цепей ПЛК осуществляется за счет встроенных в него светодиодов. Для подключения внешних входных и выходных аналоговых и дискретных сигналов, например, при совместной работе ПЛК и физического объекта, используются блоки гнезд входных и выходных сигналов, а также блок гнезд для подключения питания +24 В постоянного тока.

На лицевой стороне модуля расположены разъёмы для подключения к сети RS485 и блоки гнезд для подключения входных и выходных дискретных и аналого-

вых сигналов. Блок гнезд для подключения входных аналоговых сигналов представляет собой группу из четырех изолированных каналов. Для каждого канала используется своя пара клемм. В зависимости от программной конфигурации возможно подключение как сигналов по напряжению от 0 до 10 В, так и токовых сигналов от 0(4) до 20 мА. В свою очередь блок гнезд для подключения выходных аналоговых сигналов представляет собой группу из двух изолированных каналов. Для каждого канала используется своя пара клемм. Также в зависимости от программной конфигурации возможно подключение как сигналов по напряжению от 0 до 10 В, так и токовых сигналов от 4 до 20 мА.

На тыльной стороне модуля расположены разъёмы, предназначенные для подключения питания модуля, подключения контроллера к USB-порту ПК для программирования ПЛК.

1.4.7 Пример разработки пользовательской управляющей программы для контроллера ПЛК150

В качестве примера рассмотрим программу управления движением тележки.

Условие задачи. Тележка перемещается по прямолинейному пути. Имеется два фиксированных положения $П0$ и $П1$. Исходное положение тележки – $П0$.

При кратковременном нажатии на кнопку «Пуск» тележка движется вперед до положения $П1$, стоит в положении $П1$ 5 с и возвращается назад в положение $П0$. В положении $П0$ немедленно возникает реверс привода тележки. Тележка вновь движется в положение $П1$, стоит в нем 5 с и возвращается в $П0$. В положении $П0$ вновь возникает команда на движение до положения $П1$ и т.д. после 10 циклов перемещения в положение $П1$ при возвращении тележки в положение $П0$ возникает сигнал «Конец цикла» $КЦ$.

Решение. Для управления движением тележки следует предусмотреть счетчик ($СЧ$) со счетом до 10, задержку времени (таймер) для формирования задержанного сигнала $П1^{\uparrow}$ и память $P_{П}$ о нажатии на кнопку «Пуск». Этому описанию соответствует логическое уравнение

$$P_{II} = (Пуск + P_{II}) \cdot \overline{КЦ}, \quad (1.3)$$

где P_{II} – команда с блока памяти;

$Пуск$ – команда с кнопки «Пуск»;

$\overline{КЦ}$ – инверсное значение сигнала «Конец цикла».

Команда на перемещение тележки вперед (B) возникает в положении $Π0$, если отсутствует сигнал счетчика $СЧ$, и сохраняется при наличии сигнала памяти пуска P_{II} до достижения положения $Π1$. Этому описанию соответствует логическое уравнение

$$B = (Π0 \cdot \overline{СЧ} + B) \cdot \overline{Π1} \cdot P_{II}, \quad (1.4)$$

где B – команда на перемещение тележки вперед;

$Π0, Π1$ – сигналы положений, соответственно $Π0, Π1$;

$\overline{СЧ}$ – инверсное значение сигнала счетчика;

$\overline{Π1}$ – инверсное значение сигнала положения $Π1$.

Команда на перемещение тележки назад (H) возникает через 5 с нахождения в положении $Π1$ и сохраняется при наличии переменной P_{II} до достижения положения $Π0$. Тогда логическое уравнение для команды H имеет вид

$$H = (Π1^{\uparrow} + H) \cdot \overline{Π0} \cdot P_{II}, \quad (1.5)$$

где $Π1^{\uparrow}$ – задержанный сигнал положения $Π1$;

$\overline{Π0}$ – инверсное значение сигнала положения $Π0$.

Сигнал конца цикла ($КЦ$) возникает в положении $Π0$ при наличии сигнала счетчика $СЧ$ о завершении 10 циклов перемещения в положение $Π1$ и сигнала P_{II} . Сигнал $КЦ$ сохраняется до очередного нажатия на кнопку «Пуск». Тогда логическое уравнение для рассматриваемого сигнала имеет вид

$$КЦ = (Π0 \cdot СЧ \cdot P_{II} + КЦ) \cdot \overline{Пуск}, \quad (1.6)$$

где $\overline{\text{Пуск}}$ – инверсное значение команды с кнопки «Пуск».

В приведенной ниже программе (рисунок 1.28) использован счетчик на вычитание. При нажатии на кнопку «Пуск» в счетчик вводится уставка 10. Счетчик считает число циклов перемещения в положение $П1$, при достижении этого положения число в счетчике уменьшается на единицу. Счетчик сбрасывается по сигналу конца цикла $КЦ$. Признаком окончания цикла является число в счетчике равное нулю, при котором на выходе счетчика «0». Поэтому в уравнениях команд B и $КЦ$ необходимо использовать инверсию сигнала $СЧ$ по отношению к ранее записанным уравнениям:

$$B = (П0 \cdot СЧ + B) \cdot \overline{П1} \cdot P_{II} , \quad (1.7)$$

$$КЦ = (П0 \cdot \overline{СЧ} \cdot P_{II} + КЦ) \cdot \overline{\text{Пуск}} . \quad (1.8)$$

Сброс счетчика производится сигналом конца цикла $КЦ$.

В таблице 1.11 представлена адресация команд и сигналов для ОВЕН ПЛК150. Соответственно, на рисунке 1.28 представлена программа управления движением тележки.

Таблица 1.11 – Адресация команд и сигналов для ОВЕН ПЛК150

Команда, сигнал	$П0$	$П1$	$Пуск$	B	H	$КЦ$	P_{II}	$СЧ$	$П1^{\uparrow}$
Адрес на модуле	$I1$	$I2$	$I3$	$Q1$	$Q2$	$Q3$			
Адрес ПЛК	$\%IX0.0$	$\%IX0.1$	$\%IX0.2$	$\%QX1.0$	$\%QX1.1$	$\%QX1.2$	$\%MX0.0$	$\%MX0.1$	$\%MX0.2$
Символьный адрес	$p1$	$p2$	$start$	$forw$	rev	$cycle$	mem	$count$	tim

1.4.8 Загрузка программы в контроллер и запуск программы

Компиляция и загрузка программы в контроллер осуществляется в следующей последовательности.

Шаг 1. В выпадающем меню «Проект» выбрать пункт «Компилировать» либо воспользоваться «быстрой» клавишей $F11$. Пройдет компиляция проекта, по итогам которой сформируется лог компиляции (рисунок 1.12). Если ошибок в проекте нет,

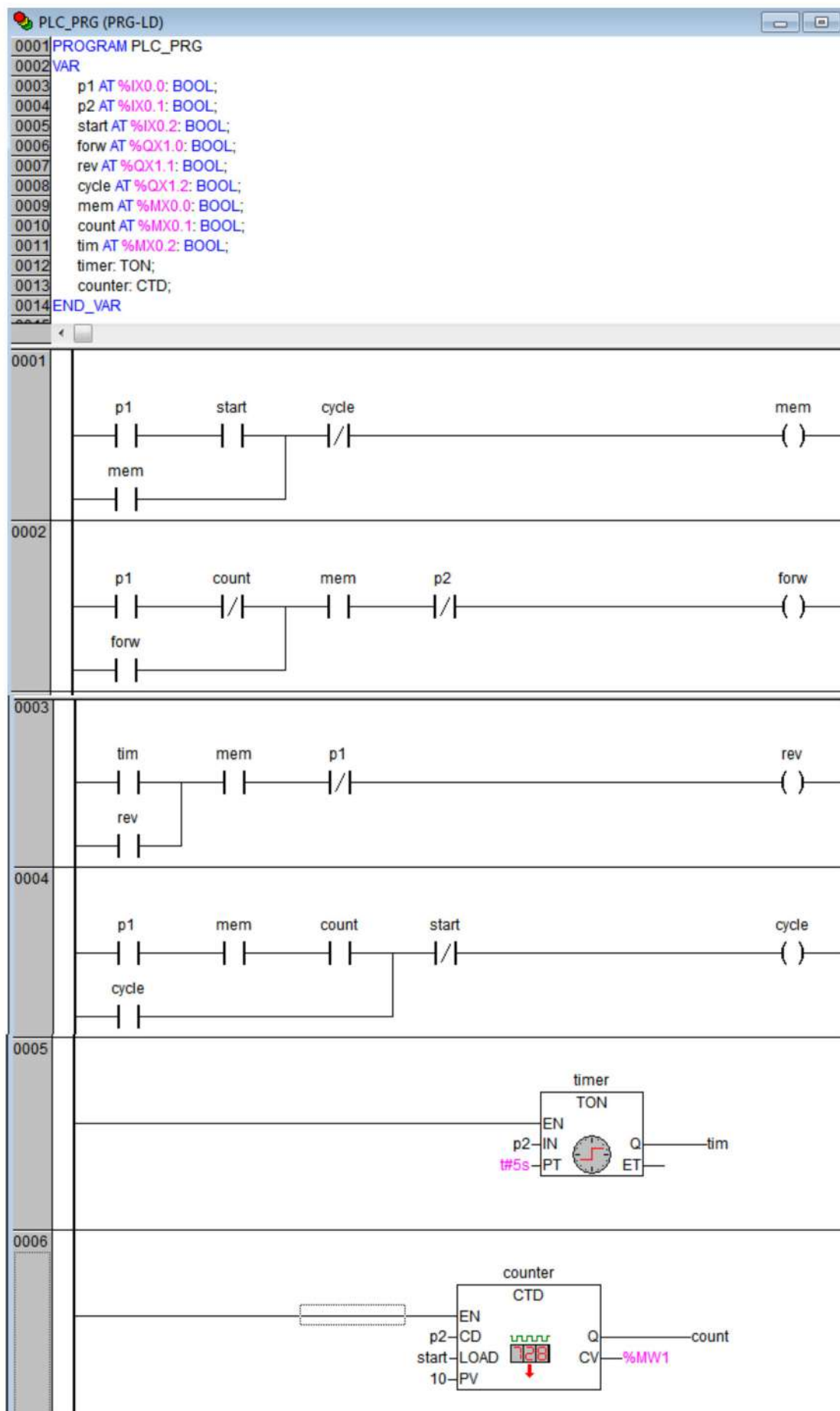


Рисунок 1.28 – Программа управления движением тележки

то весь текст будет серого цвета и в последней строке будет запись «0 ошибок, 0 предупреждений». В противном случае необходимо будет исправить все обозначенные ошибки.

Шаг 2. Настроить параметры порта связи с ПЛК150. Для этого в выпадающем меню «Онлайн» выбрать пункт «Параметры связи». В открывшемся окне нажать кнопку «New», в результате чего откроется окно создания нового канала связи (рисунок 1.29, а), в котором нужно указать название канала связи, например, «open_» и выбрать строку *Tcp/Ip (Level 2)* и нажать «ОК».

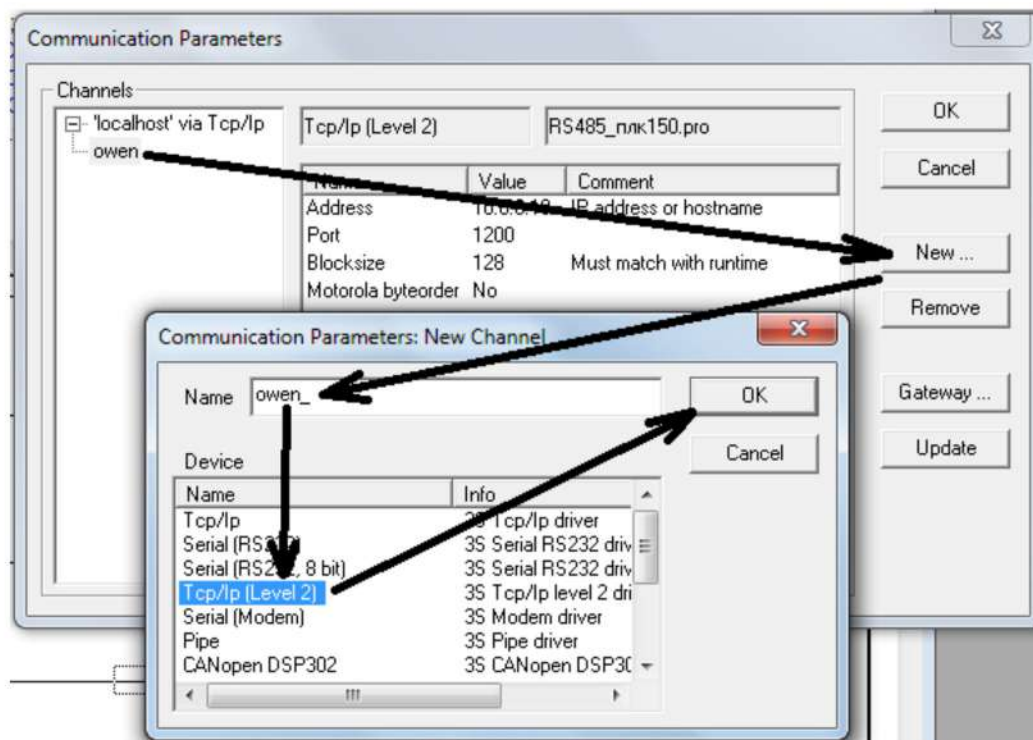
Шаг 3. В открывшемся окне настроек порта связи (рисунок 1.29, б) указать следующие значения параметров: *Address* -10.0.6.10.

Шаг 4. Проверить настройки сетевой карты. IP-адрес ноутбука должен быть в той же подсети, что и ПЛК (IP-адрес: 10.0.6.5, маска: 255.255.0.0). При необходимости настроить сетевые параметры ноутбука.

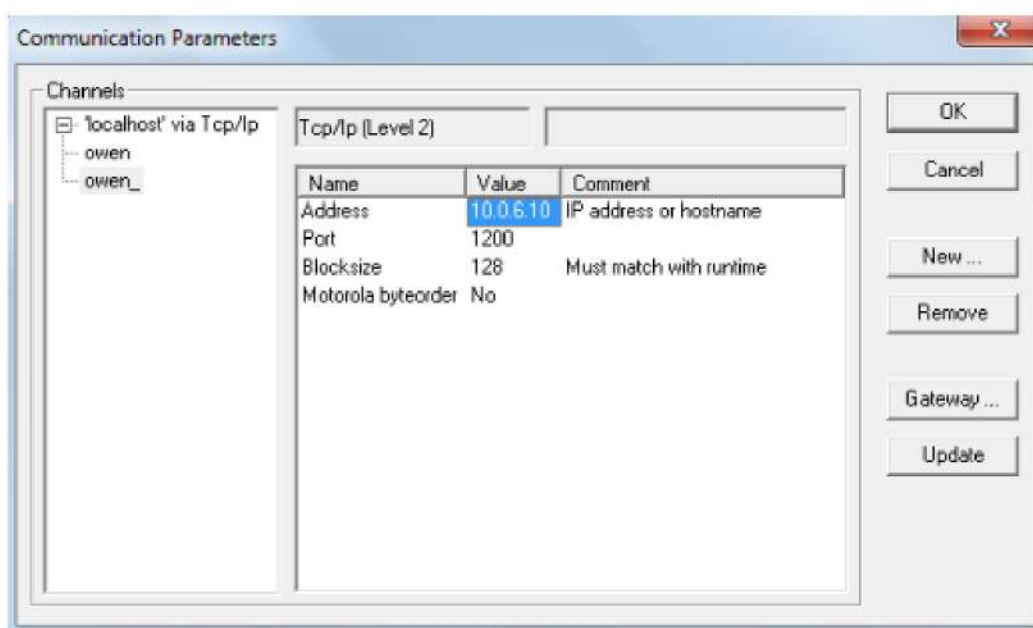
Шаг 5. Создать загрузочный проект. Для этого в выпадающем меню «Онлайн» выбрать пункт «Создание загрузочного проекта». В итоге будут созданы файлы загрузочного проекта.

Шаг 6. Загрузить проект в память ПЛК. Для этого в выпадающем меню «Онлайн» выбрать пункт «Подключение», в появившемся окне (рисунок 1.30, а) выбрать «НЕТ». После этого в выпадающем меню «Онлайн» выбрать пункт «Загрузка». Прогресс процесса загрузки представлен на рисунок 1.30, б. В итоге проект полностью загружен в память ПЛК. Запуск контроллера осуществляется нажатием кнопки «СТАРТ/СТОП» либо выбором пункта «СТАРТ» в выпадающем меню «Онлайн».

Во время работы контроллера доступен режим мониторинга хода выполнения программы. *CoDeSys* 2.3 переходит в режим мониторинга сразу же после перехода в онлайн-режим работы. В этом режиме будут отображаться состояния контактов и обмоток, а также значения счетчиков, таймеров и других функции в реальном времени (рисунок 1.31).

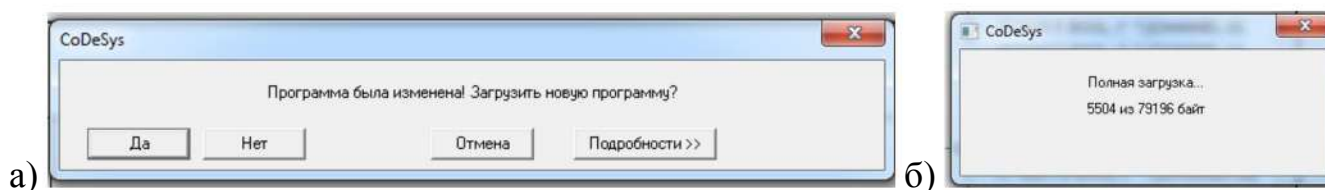


a)



б)

Рисунок 1.29 – Настройка параметров связи



a)

б)

Рисунок 1.30 – Загрузка проекта в память ПЛК

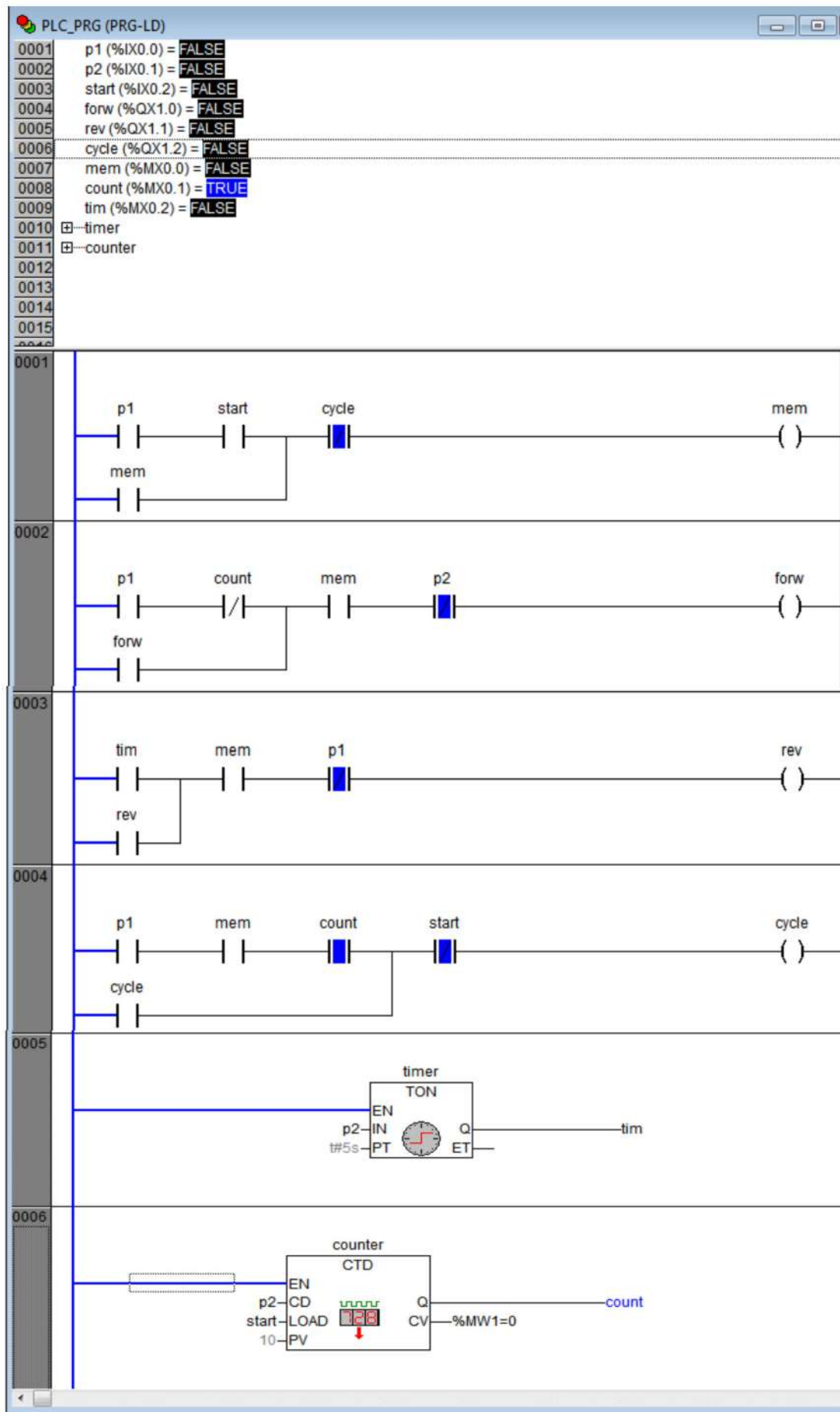


Рисунок 1.31 – Режим мониторинга

1.4.9 Порядок проведения лабораторной работы и варианты заданий

При подготовке к лабораторной работе обучающийся должен в соответствии с таблицей 1.12 выбрать свой вариант задания.

Таблица 1.12 – Варианты заданий

Номер бригады	1 и 5			2 и 6			3 и 7			4 и 8		
Член бригады	1	2	3	1	2	3	1	2	3	1	2	3
Номер варианта	1	2	3	4	5	6	7	8	9	10	11	12

Для заданного варианта необходимо разработать алгоритм ее решения. В соответствии с заданием определиться с адресацией входных и выходных переменных. Затем определиться с адресацией вводимых промежуточных переменных и с адресацией используемых таймеров и счетчиков. Составить программу для контроллера ПЛК150.

Запустить программу в работу и убедиться в правильности ее работы. При наличии ошибок в работе устранить их и продемонстрировать преподавателю работу правильно функционирующей программы.

Ниже приведены задания для самостоятельной работы.

Вариант 1. Перекладыватель служит для перемещения листов со стола на постоянно вращающийся рольганг. Перекладыватель имеет подъемник листов, установленный на тележке. Подъемник представляет собой поперечину, перемещающуюся по вертикальным стойкам сверху вниз и обратно. Поперечина имеет на конце электромагнит для притягивания листов.

В исходном положении перекладыватель стоит в положении *Л0* над столом, а его подъемник в крайнем верхнем положении *КВ*. При поступлении листа на стол подается сигнал наличия листа, подъемник включается для движения вниз. При касании листа подъемник останавливается и включается электромагнит. Через 2 с включается подъемник на подъем, достигает положения *КВ* и останавливается. Включается тележка и перемещается в положение *П1* над рольгангом, где останавливается.

ливается и электромагнит отключается. Лист падает на рольганг. Через 2 с тележка движется в положение $Л0$. Цикл повторяется при поступлении очередного сигнала наличия листа. Включение системы в работу – нажатием кнопки «Пуск».

Вариант 2. В контроллере программным путем необходимо реализовать генератор импульсов. Время наличия импульса равно 1 с, время его отсутствия – 2 с. При нажатии на кнопку «Пуск» начинается счет импульсов. При прохождении 10 импульсов загорается лампа $Л1$, при прохождении следующих 10 импульсов – лампа $Л2$, аналогично включаются лампы $Л3$ и $Л4$. После загорания лампы $Л4$ (т.е. после прохождения 40 импульсов) счет прекращается и все лампы горят до нажатия на кнопку «Стоп». Генератор импульсов необходимо реализовать программным путем.

Вариант 3. Автоматизировать перемещение пуансона штампованного прессы. Рабочий заправляет металлическую ленту (заготовку) и нажимает кнопку «Пуск». Пуансон из крайнего верхнего положения $Л0$ движется вниз, при своем движении до положения $Л1$ входит в матрицу, выбивая из ленты деталь необходимой формы, и возвращается в положение $Л0$. Чисто механическим устройством лента перемещается на один шаг и через 2 с пуансон совершает очередное движение.

Лента рассчитана на изготовление 10 деталей, поэтому через 10 ходов пуансона формируется сигнал «Конец цикла».

При подаче питания на систему автоматизации пуансон из любой точки должен прийти в положение $Л0$.

Вариант 4. На программируемом контроллере реализовать систему управления толкателем с кривошипно-шатунным механизмом с реверсивным электродвигателем.

Толкатель имеет два фиксированных положения $Л0$ и $Л1$. При подаче питания на систему толкатель из любого положения движется вперед медленно до положения $Л0$. При появлении кратковременного сигнала «Пуск» толкатель из положения $Л0$ медленно движется до положения $Л1$, стоит там в течение 2 с, а затем электродвигатель включается для движения назад быстро и возвращается в положение $Л0$, где формируется сигнал «Конец цикла». Цикл повторяется при нажатии кнопки «Пуск».

При нажатии кнопки «Стоп» толкатель из любого положения на быстрой скорости возвращается в положение ЛЮ.

Вариант 5. Автоматизировать управление крышкой котла.

При кратковременном нажатии кнопки «Открыть» крышка поднимается до крайнего верхнего положения, в котором поворачивается, открывая доступ к содержимому котла. При кратковременном нажатии на кнопку «Заккрыть» крышка поворачивается в положение точно над котлом и затем опускается, закрывая котел.

Вариант 6. Автоматизировать управление дверью (воротами) въезда/выезда гаража. Нормальное (исходное) положение двери закрытое. При кратковременном нажатии кнопки «Открыть» включается звонок, и через 5 с включается привод двери на открывание. При полном открытии двери привод двери отключается сразу, а звонок звенит еще 2 с.

Закрывание дверей – аналогично: при кратковременном нажатии кнопки «Заккрыть» включается звонок, через 5 с включается привод двери на закрывание. При полном закрытии двери привод двери отключается, а звонок звенит еще 2 с.

Вариант 7. Печь сопротивлений содержит три нагревательных элемента (НЭ), каждый из которых питается от собственного тиристорного преобразователя (ТП) соответственно ТП1, ТП2, ТП3. Сигнал «1» на входе ТП соответствует максимально допустимому току НЭ, сигнал «0» – минимально допустимому току НЭ. Синтезировать схему, обеспечивающую алгоритм работы группы ТП, представленный в таблице 1.13.

Таблица 1.13 – Алгоритм работы группы ТП

Номер импульса генератора	–	1	2	3	4	5	6	7	8	9	...
Состояние ТП1	0	1	0	0	1	1	0	1	0	0	...
Состояние ТП2	0	0	1	0	0	1	1	0	1	0	...
Состояние ТП3	0	0	0	1	1	0	1	0	0	1	...

Начало работы осуществляется после нажатия на кнопку «Пуск». Цикл управления печью сопротивления состоит из 6 тактов. Переход с одного такта на другой осуществляется при поступлении очередного импульса от генератора импульсов.

Циклы управления непрерывно следуют друг за другом.

При поступлении команды «Стоп» прекращается подача импульсов, и все ТП отключаются. Для возобновления работы необходимо нажать на кнопку «Пуск». Генератор импульсов реализуется программным путем.

Вариант 8. Манипулятор (рука со схватом) служит для подачи заготовок из накопителя в пресс. Рука манипулятора может перемещаться вперед/назад из крайнего заднего *КЗ* в крайнее переднее *КП* положение и назад. Схват имеет электромагнитный механизм зажима/разжима заготовок.

Исходное положение манипулятора перед накопителем – в положении *КЗ*, схват под действием пружин зажат. При поступлении запроса со станка и наличии заготовки в накопителе включается электромагнит разжима схвата. Рука идет в крайнее переднее положение *КП*, через 2 с схват зажимает заготовку (электромагнит схвата отключается), и рука возвращается в положение *КЗ*. Схват разжимается (электромагнит схвата включается), и через 2 с рука идет в положение *КЗ* (исходное положение руки). Если есть запрос со станка, а накопитель пуст, то включается сигнальная лампа «Нет детали» и кратковременный звуковой сигнал (в течение 5 с). Световой сигнал снимается кнопкой «Сброс». Включение системы в работу – нажатием кнопки «Пуск».

Вариант 9. Автоматизировать процесс безупорной остановки длинномерного металла на рольганге *P1*.

На рольганг *P1* металл передается с предыдущего рольганга *P0*, наличие металла на котором контролируется датчиком *Д0*. Вдоль рольганга *P1* расположены датчики, контролирующие наличие металла последовательно в положениях *П1*, *П2* и *П3*. Нормальное положение рольганга *P1* – неподвижное. При наличии сигнала с датчика *Д0* и отсутствии металла на рольганге *P1*, рольганг *P1* включается и металл перемещается по рольгангу *P1*. При достижении передним концом металла положения *П1* происходит снижение скорости *P1* до «ползучей» скорости, а в положении *П2* привод рольганга *P1* отключается. Если по каким либо причинам металл достиг положения *П3*, то привод рольганга *P1* реверсируется и работает на пониженной скорости, пока металл не уйдет из положения *П3*. При остановке переднего конца

металла между положениями $П2$ и $П3$ формируется сигнал разрешения работы механизмов уборки металла с $P1$.

Включение системы в работу – при нажатии на кнопку «Пуск».

Вариант 10. Автоматизировать работу толкателя с кривошипно-шатунным механизмом с нереверсивным электродвигателем.

При нажатии кнопки «Пуск» кривошип из положения $\alpha = 0^\circ$ движется до положения $\alpha = 120^\circ$. В этом положении происходит переход на пониженную (малую) скорость, кривошип идет до положения $\alpha = 180^\circ$ и останавливается. Через 3 с электродвигатель автоматически вновь включается, и на повышенной скорости шатун идет в положение $\alpha = 0^\circ$. Стоит в нем 3 с, включается и движется к положению $\alpha = 120^\circ$ и на пониженной скорости до $\alpha = 120^\circ$ и т.д. После 5 толканий толкатель останавливается в исходном положении (кривошип – в положении $\alpha = 0^\circ$) и формируется сигнал «Конец цикла».

Очередной цикл из пяти толканий – после очередного нажатия на кнопку «Пуск». При нажатии на кнопку «Стоп» толкатель из любого положения на повышенной скорости движется в положение $\alpha = 0^\circ$ и останавливается.

Вариант 11. Тележка движется по кольцевому пути, на котором есть 3 фиксированных положения $П0$, $П1$ и $П2$. Привод тележки – от нереверсивного электродвигателя.

При подаче питания из любого положения тележка должна идти в положение $П0$. При нажатии на кнопку «Пуск» тележка без задержки движется в положение $П1$, стоит там в течение 3 с, затем движется в $П2$, стоит там в течение 4 с, движется в положение $П0$, стоит там в течение 5 с, движется в $П1$, стоит там в течение 3 с и т.д. То есть тележка непрерывно движется по кольцевому пути с остановками в фиксированных положениях.

При нажатии на кнопку «Стоп» тележка должна без остановки идти в $П0$ и там ожидать очередного нажатия кнопки «Пуск».

Вариант 12. Автоматизировать линию сортировки изделий. По конвейеру движутся низкие изделия, но редко могут встречаться и высокие. По ходу конвейера установлены две фотоголовки на расстоянии друг от друга, равном половине шири-

ны изделия. Первая по ходу конвейера фотоголовка настроена на низкие изделия (нижняя фотоголовка), вторая – на высокие изделия (верхняя фотоголовка).

При прохождении низкой детали перекрывается только нижняя фотоголовка, а при высоких – вначале нижняя, а затем верхняя фотоголовки. На выходе конвейера детали ударяются о сортировочную планку, и высокие изделия поступают в правый накопитель (левое положение планки), низкие – в левый (правое положение планки). Планка перебрасывается пневмоприводом. Для переброски планки влево или вправо достаточно хотя бы кратковременно подать команду на ее перемещение влево или вправо. Включение системы в работу – нажатием кнопки «Пуск».

1.4.10 Требования к отчёту

Отчет должен содержать:

- название лабораторной работы;
- цель работы;
- условия задачи по варианту, принятые обозначения переменных, логические функции для выходных и промежуточных переменных, при необходимости циклограмму работы оборудования;
- логические функции в адресах программируемого контроллера;
- программу для реализации системы управления;
- методику экспериментальной проверки функционирования реализованной системы управления и результаты проверки;
- выводы по работе.

1.5 Контрольные вопросы

1.5.1 Что означает термин «конфигурирование проекта»?

1.5.2 Что означает термин «*target*-файл», где и как он используется в программной среде *CoDeSys* 2.3?

1.5.3 Что такое объявление переменных, и как это используется при программировании контроллера?

1.5.4 Можно ли в ПЛК150 реализовать таймер с уставкой времени 0,07 с?

1.5.5 При каких условиях на выходе таймеров контроллера формируется сигнал «0» и сигнал «1»?

1.5.6 При каких условиях на выходе счетчиков контроллера формируется сигнал «0» и сигнал «1»?

1.5.7 Что означает создание функционального блока (*FB*) при программировании контроллера?

1.5.8 Что означает создание и открытие функции (*FC*) при программировании контроллера?

1.5.9 Какие основные элементы включает лабораторный стенд САУ-ОВЕН?

1.5.10 Какими достоинствами обладает программируемый логический контроллер ОВЕН ПЛК150?

1.5.11 В качестве каких устройств может использоваться контроллер ПЛК150?

2 Изучение технических характеристик и основ программирования панели оператора ОВЕН СП270

2.1 Общие сведения об изучаемом оборудовании

Все необходимое оборудование для выполнения лабораторной работы расположено в модуле программируемого логического контроллера и панели оператора, внешний вид которого представлен на рисунке 1.25.

В данной лабораторной работе используется сенсорная панель оператора ОВЕН СП270, внешний вид которой представлен на рисунке 2.1. В таблице 2.1 представлено описание основных элементов панели оператора. Технические характеристики панели оператора СП270 представлены в таблице 2.2.

Таблица 2.1 – Описание основных элементов панели оператора

Элемент	Функциональное назначение
Корпус	Размещение компонентов панели
Дисплей с сенсорной клавиатурой	Отображение и ввод информации
Разъём «PLC»	Связь панели оператора с внешними устройствами автоматизации по интерфейсам RS232 и RS845
Разъём «DOWNLOAD»	
Резиновая прокладка	Защита корпуса панели от проникновения влаги и пыли при установке в щит
Разъём питания (не показан)	Подключение питания панели оператора

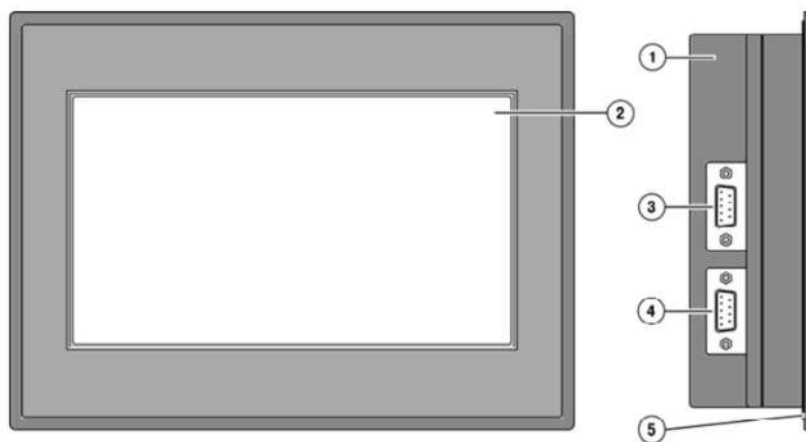


Рисунок 2.1 – Внешний вид панели оператора ОВЕН СП270

Таблица 2.2 – Технические характеристики панели оператора СП 270

Параметр		Значение
Модель		СП270
Напряжение питания постоянного напряжения, В		от 22 до 26
Потребляемый ток, мА		150
Тип дисплея		цветной <i>TFT</i>
Рабочая зона дисплея, мм		156 x 88
Разрешение, пикс.		480 x 234
Отображаемые цвета		256
Память	программ (Flash-RAM), МБ	4
	данных (SDRAM), КБ	4
Интерфейсы связи RS232		(2 канала), RS485 (1 канал)
Протоколы передачи данных		<i>ModBus-RTU, ModBus-ASCII</i>
Режимы работы		<i>Master, Slave</i>
Звуковой излучатель		Есть
Часы реального времени		Есть

Для включения питания модуля используется кнопка с фиксацией и подсветкой. На лицевой стороне модуля расположены разъёмы для подключения к сети RS485. На тыльной стороне модуля расположены разъёмы, предназначенные для подключения питания модуля, подключения панели оператора к USB-порту ПК для программирования.

2.2 Основы программирования панели оператора

Работа по программированию панели оператора ОВЕН СП270 сводится к следующему:

- в программе «Конфигуратор СП200» создать и сконфигурировать проект;
- сформировать и отредактировать структуру проекта (добавить в проект новые компоненты, отредактировать параметры элементов или удалить компоненты из проекта);
- сохранить и записать подготовленный проект в панель оператора.

Конфигуратор позволяет формировать и сохранять пользовательские экраны, отображаемые на экране панели в процессе эксплуатации. Пользовательские экраны

включают наборы базовых элементов, каждый из которых позволяет выполнять определенную задачу проекта. К базовым элементам экрана относятся буквы (русские или английские) и символы, пиктограммы (индикатор, график, линейка, регистр для отображения/редактирования данных, кнопки переключения экранов и другие элементы).

Совокупность экранов образует проект, который можно загрузить в панель или сохранить в виде файла на жестком диске ноутбука.

Создание и конфигурирование нового проекта сводится к следующим действиям.

Шаг 1. Для запуска программы необходимо на рабочем столе *Windows* дважды щелкнуть курсором мыши по соответствующему ярлыку. После чего откроется пустой экран программы «Конфигуратор СП200» (рисунок 2.2), в котором можно создать новый проект или открыть уже созданный проект.

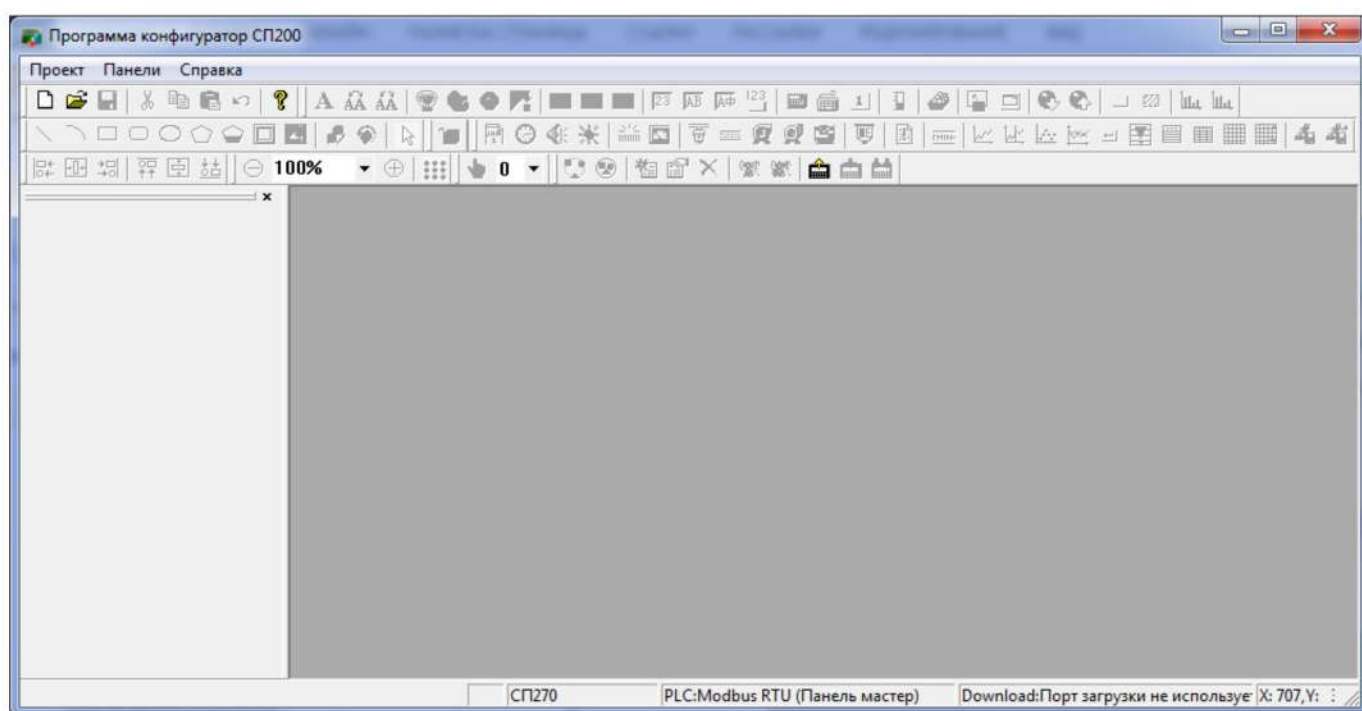


Рисунок 2.2 – Стартовый экран программы «Конфигуратор СП200»

Шаг 2. После вызова команды «Проект/Новый проект» появится окно (рисунок 2.3, а) в котором можно выбрать панель оператора.

Шаг 4. Указать название проекта, его автора и, если необходимо, какие-то комментарии.

По завершению конфигурирования откроется главное окно проекта (рисунок 2.4).

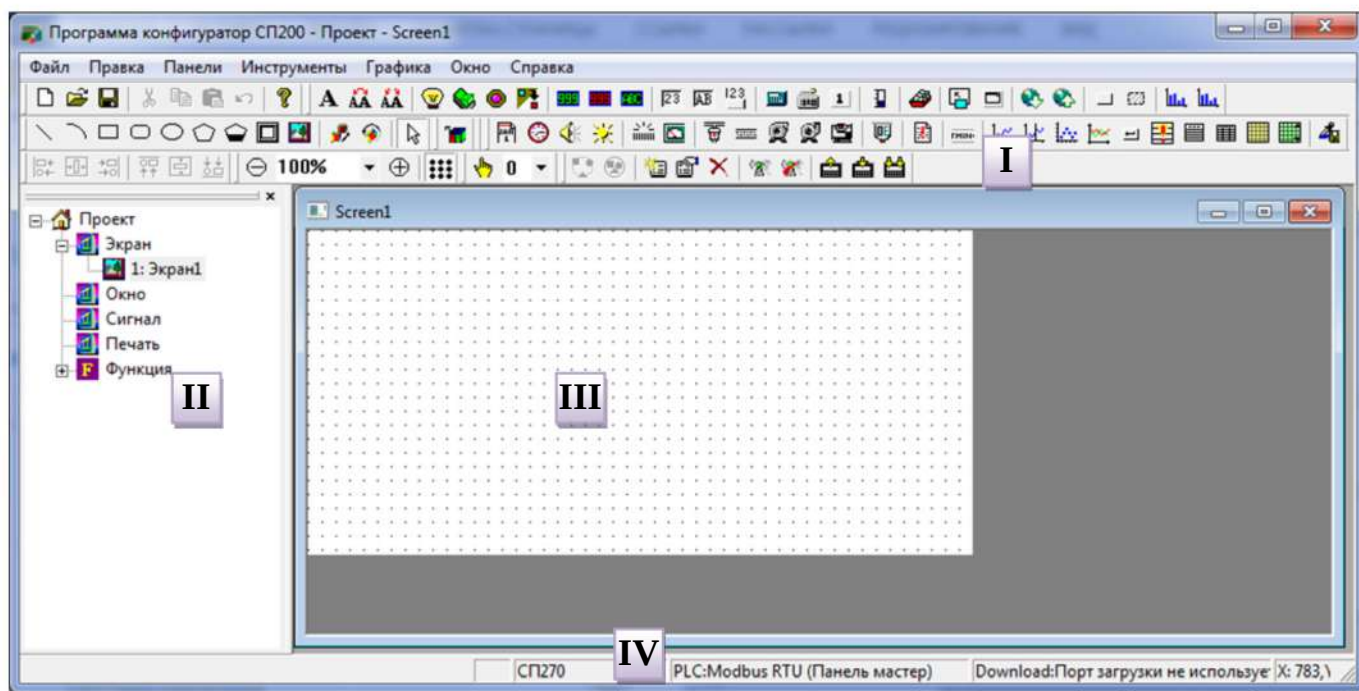


Рисунок 2.4 – Главное окно проекта

Всю рабочую область главного окна проекта можно разделить на следующие области:

- I – панель инструментов программы;
- II – область отображения структуры текущего проекта;
- III – область отображения текущего компонента проекта;
- IV – строка состояния.

При редактировании экранов сенсорной панели оператора доступно несколько панелей инструментов (рисунок 2.5). Краткое описание основных элементов этих панелей представлено в таблице 2.3.

Для правильной работы панели оператора для элементов проекта, обменивающихся значениями регистров (или битов) с ПЛК, необходимо задать адрес связан-

ного регистра (бита). К таким элементам относятся «лампы», «кнопки», «переключатели» и т.д., отображаемые на экране панели.

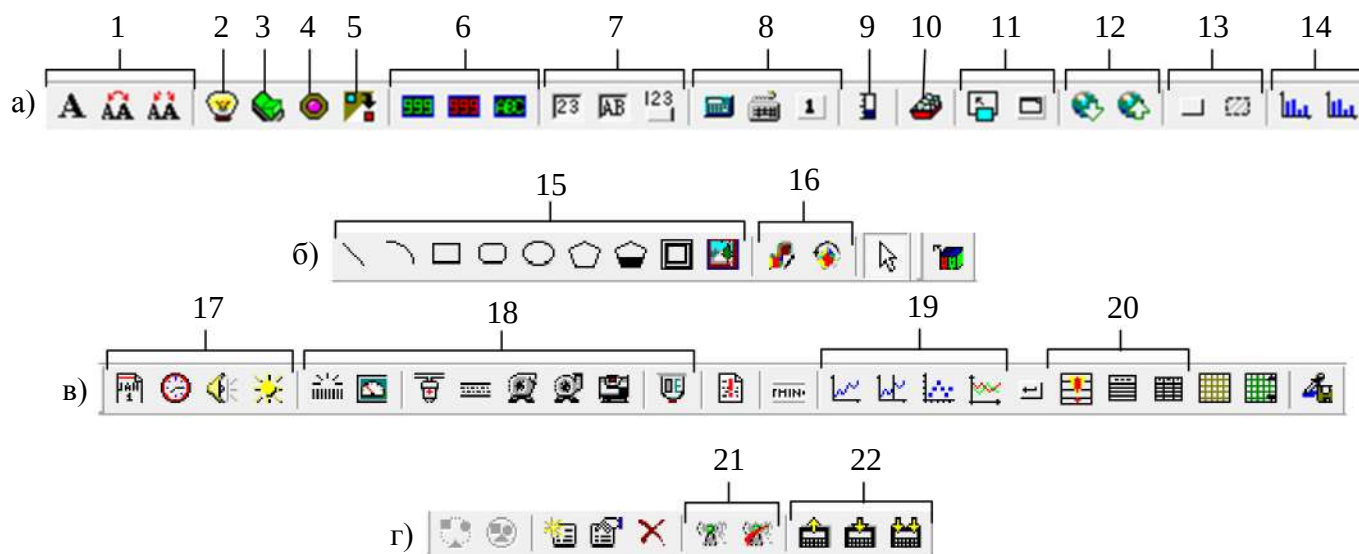


Рисунок 2.5 – Панели инструментов

Таблица 2.3 – Краткое описание основных элементов панели инструментов

Номер панели	Описание	Номер панели	Описание
1	Работа с текстом	12	Обмен данными с ПЛК
2	Лампа	13	Функциональные объекты
3	Управление битом	14	Гистограмма
4	Переключатель с индикацией	15	Графические примитивы
5	Переход экрана	16	Движение и поворот объекта
6	Дисплеи	17	Сервис (дата, время, подсветка)
7	Ввод данных	18	Приборы-примитивы
8	Клавиатуры	19	Графики
9	Вертикальная линейка	20	События
10	Динамическое отображение	21	Эмуляция
11	Работа с окнами	22	Работа с панелью в Онлайн-режиме

Настройка управления производится на вкладке «Управление» окна «Свойства» элемента проекта. Настройка управления выполняется в зависимости от того, в каком режиме и по какому протоколу должна работать панель оператора с текущим проектом. Независимо от того, по какому протоколу и в каком режиме работает

панель, элементы могут обращаться к внутренней оперативной или энергонезависимой памяти панели.

Задание параметров управления элементами осуществляется в следующей последовательности:

- в поле «Устройство» (рисунок 2.6) необходимо указать порт, по которому будет происходить обмен. Это может быть порт «PLC» или «Download»; порты становятся доступными для выбора только в том случае, если в соответствии с настройками проекта, панель работает в режиме «мастер»;
- указать адрес устройства, с которым будет происходить связь;
- в поле «Регистр» указать способ (команду) связи с переменной (регистром) внешнего устройства, с которого (в который) будет считываться (записываться) значение; в этом же поле указывается адрес регистра, с которым будет происходить связь, способы связи зависят от типа выбранного протокола обмена.

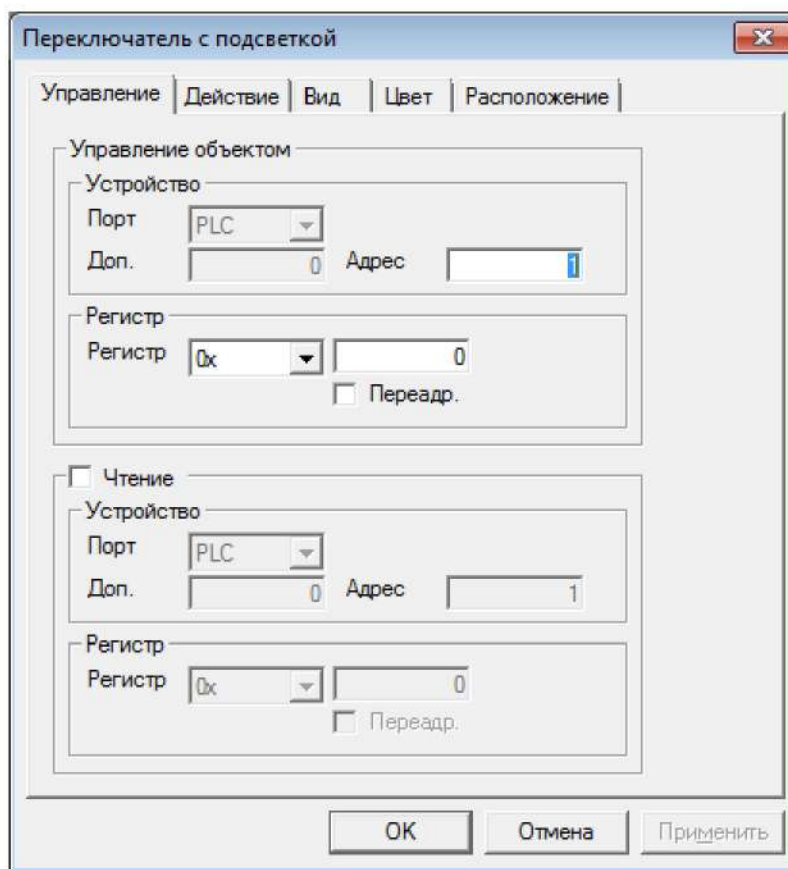


Рисунок 2.6 – Окно задания параметров управления элементом

Панель может обращаться к своей внутренней памяти. Для этого необходимо выбирать следующие способы связи:

- PSW – оперативная память панели, один адрес занимает один регистр типа word, максимальный размер зависит от свободной оперативной памяти (4 кбайт);

- PFW – энергонезависимая память панели, один адрес занимает один регистр типа word, максимальный размер зависит от свободной энергонезависимой памяти (4 Мбайт);

- PSB – оперативная память панели, один адрес занимает один бит, максимальный размер зависит от свободной оперативной памяти (4 кбайт).

Панели оператора поддерживают несколько режимов работы по протоколу Modbus:

- *Modbus RTU Master* – работа по протоколу *Modbus RTU*, панель выполняет функцию мастера сети;

- *Modbus RTU Slave* – работа по протоколу *Modbus RTU*, функцию мастера сети выполняет ПЛК, панель работает в подчиненном режиме (Slave);

- *Modbus ASCII Master* – работа по протоколу *Modbus ASCII*, панель выполняет функцию мастера сети.

При работе панели по протоколу *Modbus* в режиме мастера сети доступны дополнительные команды для связи с ПЛК:

- 0х – запрос к ПЛК будет производиться по функции 0х01 (*Read Coil Status*);

- 1х – запрос к ПЛК будет производиться по функции 0х02 (*Read Input Status*);

- 3х – запрос к ПЛК будет производиться по функции 0х03 (*Read Holding Registers*);

- 4х – запрос к ПЛК будет производиться по функции 0х04 (*Read Input Registers*).

Команды 3х и 4х доступны при работе с цифровыми индикаторами, графиками и любыми другими элементами, которые могут отображать на экране панели более двух состояний, таким образом, работа панели сводится к работе с регистрами ПЛК.

Команды 0x и 1x доступны при работе с кнопками, сигнальными окнами и любыми другими элементами, которые могут отображать на экране панели не более двух состояний (работа панели сводится к работе с битами ПЛК).

При записи значения регистра в ПЛК панель формирует посылку с функцией записи 0x06, или с функцией 0x05 – если необходимо записать значение бита.

При работе панели по протоколу *Modbus* в подчиненном режиме (*Slave*), ПЛК или любой другой мастер сети может обращаться к ее внутренним регистрам.

Если пользователю необходимо вывести значения регистров или битов на экран панели, то на вкладке «Управление» в поле «Регистр» необходимо указать область памяти: PSW или PSB. Чтение и запись значений регистров происходит по функциям *Modbus* 0x03, 0x06, 0x16 (для памяти PSW – адресация во внутреннюю память панели SDRAM; адресация происходит по регистрам) и 0x01, 0x05 (для памяти PSB- адресация во внутреннюю память панели; адресация происходит по битам).

Также при задании адресов регистров доступна функция переадресации. При установке флажка в поле переключателя «Переадресация» становится доступна кнопка, расположенная справа от переключателя (рисунок 2.7 а).

По нажатию этой кнопки открывается дополнительная форма ввода (рисунок 2.7 б), в которой можно задать дополнительный номер регистра для перенаправления в этот регистр информации из связанного регистра.

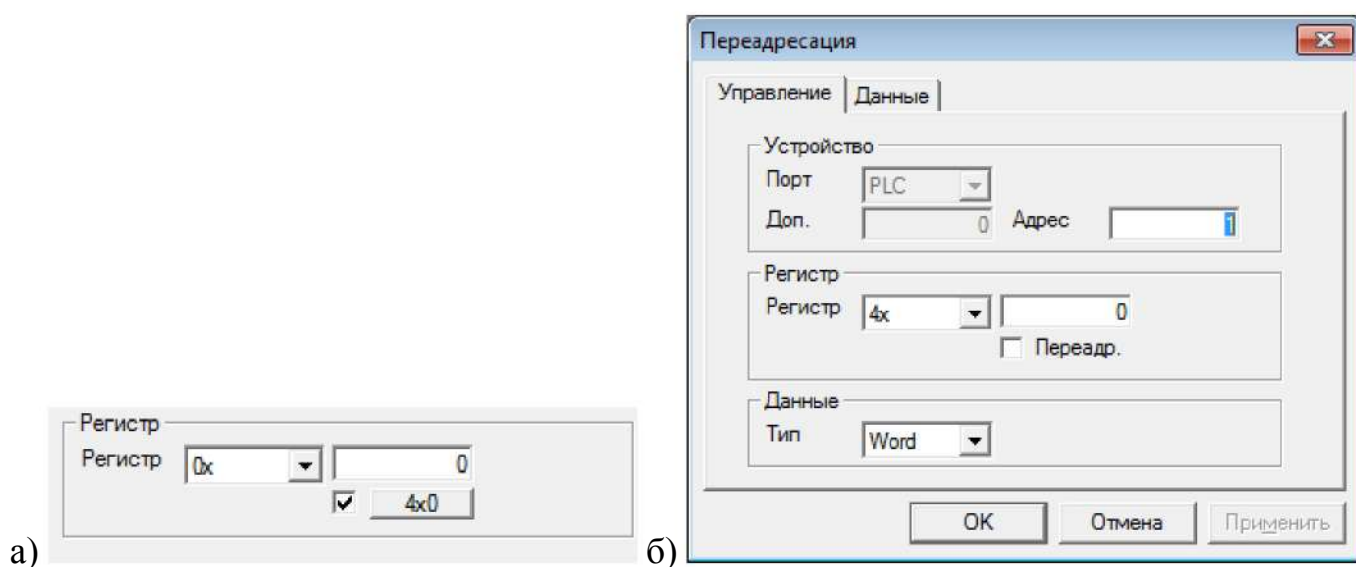


Рисунок 2.7 – Функция переадресации

Особенности использования переадресации:

- все значения вводятся в десятичном формате;
- память *PFW* имеет ограниченный ресурс перезаписи (50000 циклов);
- при использовании внутренней памяти поле задания адреса ПЛК неактивно;
- при использовании портов связи в *slave*-режиме, пользователь может использовать только внутреннюю память;
- не использовать адреса регистров внутренней памяти меньше чем 256.

Для работы панели оператора с ПЛК необходимо сконфигурировать программируемый контроллер. В зависимости от того, в каком режиме будет работать панель оператора, конфигурирование ПЛК несколько отличается.

Для работы панели оператора с ПЛК в режиме «*master-slave*» в программной среде *CoDeSys* 2.3 нужно выполнить следующее.

Шаг 1. Запустить ПО *CoDeSys* 2.3 через соответствующий ярлык на рабочем столе, и создать проект для программируемого контроллера.

Шаг 2. В закладке «Ресурсы» менеджера проекта двойным щелчком левой клавиши мыши выбрать пункт «Конфигурация ПЛК». В открывшемся окне выделить строку «*PLC150A-L*», нажать правой клавишей мыши и в появившемся меню выбрать «добавить подэлемент» → «*ModBus (Slave)*» (рисунок 2.8, а). В закладке «Параметры модуля» в поле «Address» необходимо установить требуемый сетевой адрес ПЛК – 1;

Шаг 3. Раскрыть строку «*ModBus (Slave) [VAR]*». В появившейся строке «*ModBus[FIX]*», нажать правой клавишей мыши (рисунок 2.8, б) и в появившемся меню выбрать «Добавить Подэлемент» → «*RS-485-1*» (сеть *RS485*, первый порт).

Шаг 4. В правом окне в закладке «Параметры модуля» настроить необходимые параметры связи с панелью оператора (рисунок 2.8, в).

Шаг 5. Выделить строку «*ModBus (Slave) [VAR]*», нажать правой клавишей мыши и в появившемся меню выбрать «Добавить Подэлемент» → «8 bits». В правом окне «Базовые параметры» можно указать дополнительные комментарии. Аналогично создать требуемое количество битовых (8 *bits*) и байтовых (2 *byte*, 4 *byte*) пе-

ременных. Раскрыв созданный регистр, можно увидеть (рисунок 2.8, г), какой адрес имеет каждый элемент регистра (бит, слово и т.д.).

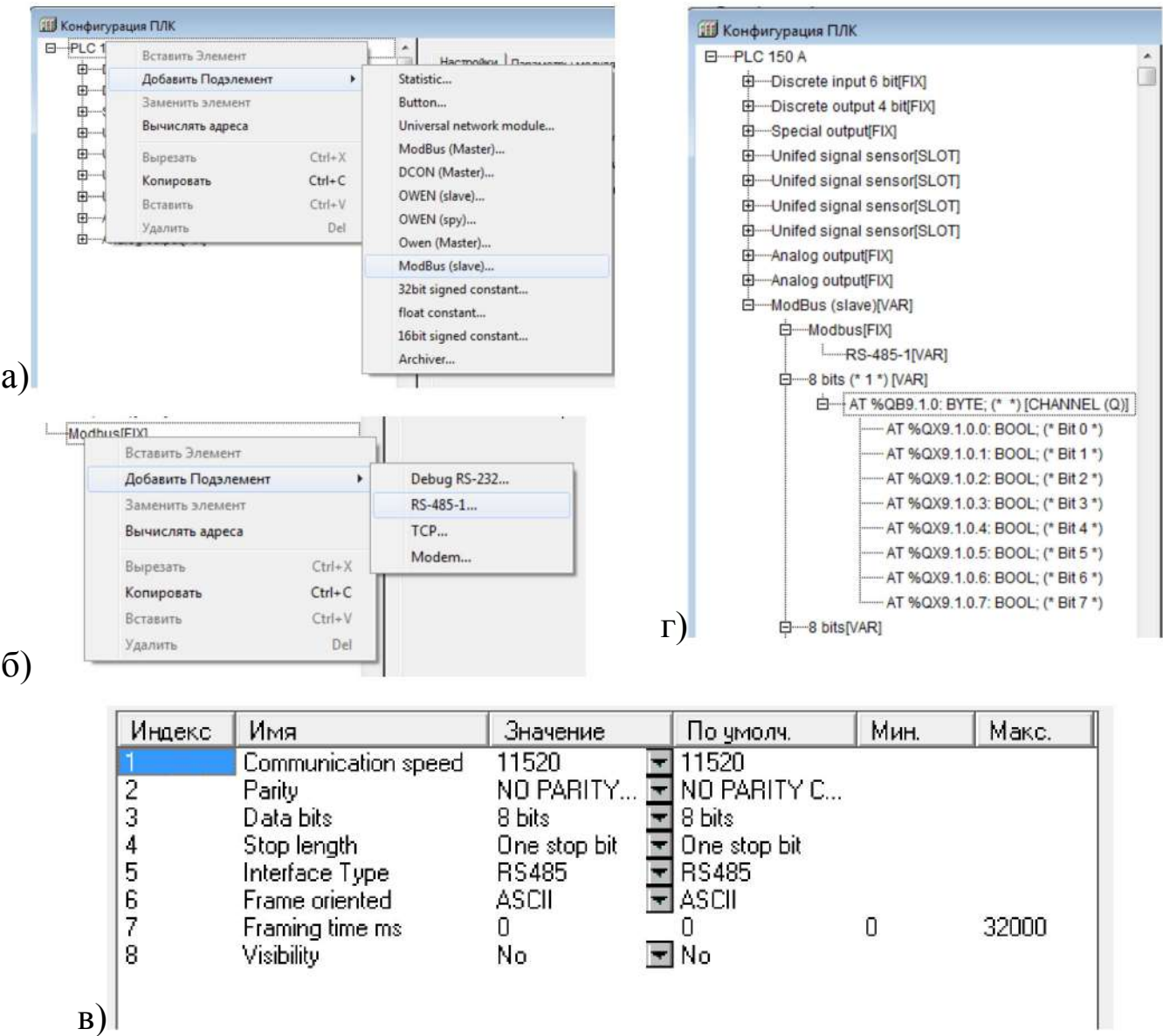


Рисунок 2.8 – Добавление *ModBus* подключения (*slave*) в ПЛК

Шаг 6. Сконфигурировать аналоговый вход ПЛК150. Для этого в окне «Конфигурация ПЛК» выбрать строку «*Unified signal sensor*» и в закладке «Параметры модуля» в поле «*Type of sensoг*» необходимо установить требуемый тип входного сигнала – от 0 до 10 В (рисунок 2.9, а) и диапазоны изменения сигнала – $A_{in} = 0$, $A_{out} = 10$.

Шаг 7. Сконфигурировать аналоговый выход ПЛК150. Для этого в окне «Конфигурация ПЛК» выбрать строку «*Analog output [Fix]*» и в закладке «Параметры модуля» в поле «*Type*» необходимо установить требуемый тип входного сигнала – от 0 до 10 В (рисунок 2.9, б).

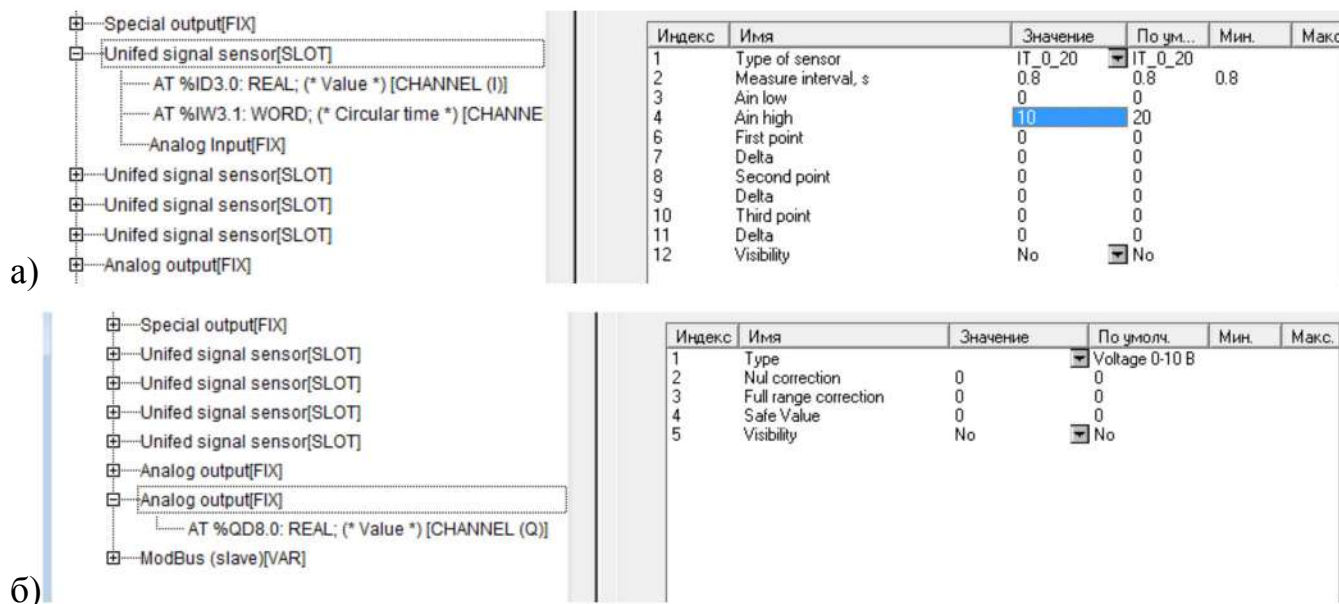


Рисунок 2.9 – Настройка аналоговых входов и выходов ПЛК 150

Шаг 8. Составить управляющую программу.

Для демонстрации обмена между панелью оператора и ПЛК используется демо-программа, представленная ниже (рисунок 2.10). Ввиду особенностей обмена данными между панелью и ПЛК в программе реализовано две функции для преобразования форматов данных:

а) *INTtoREAL* (преобразование числа из типа *INT* в тип *REAL*);

б) *REALtoINT* (преобразование числа из типа *REAL* в тип *INT*).

Для работы панели оператора с ПЛК в режиме «*slave-master*» в программной среде *CoDeSys 2.3* нужно выполнить следующее.

Шаг 1. Запустить ПО *CoDeSys 2.3* через соответствующий ярлык на рабочем столе и создать проект для программируемого контроллера.

Шаг 2. В закладке «Ресурсы» менеджера проекта выбрать пункт «Конфигурация ПЛК». В открывшемся окне выделить строку «*PLC150A-L*», нажать правой кла-

вишей мыши и в появившемся меню выбрать «Добавить подэлемент» → «ModBus (Master)».

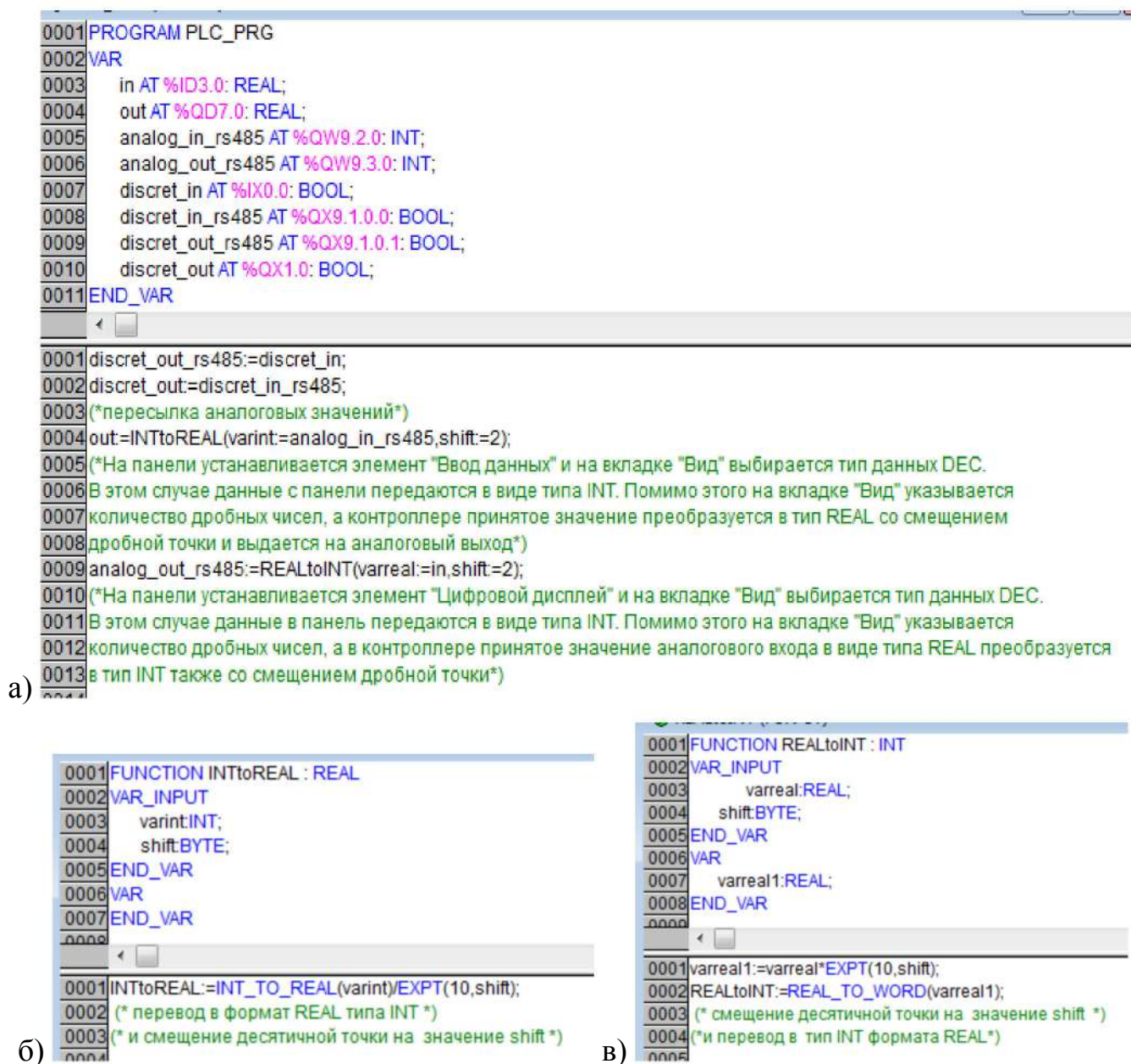


Рисунок 2.10 – Пример управляющей программы (а) и прикладных функций *INTtoREAL* (б) и *REALtoINT* (в) для работы по сети RS485

Шаг 3. Выделить строку «*DEBUG RS-232(SLOT)*», нажать правой клавишей мыши и в появившемся меню выбрать «Заменить элемент» → «RS-485-1» (рисунок 2.11, а).

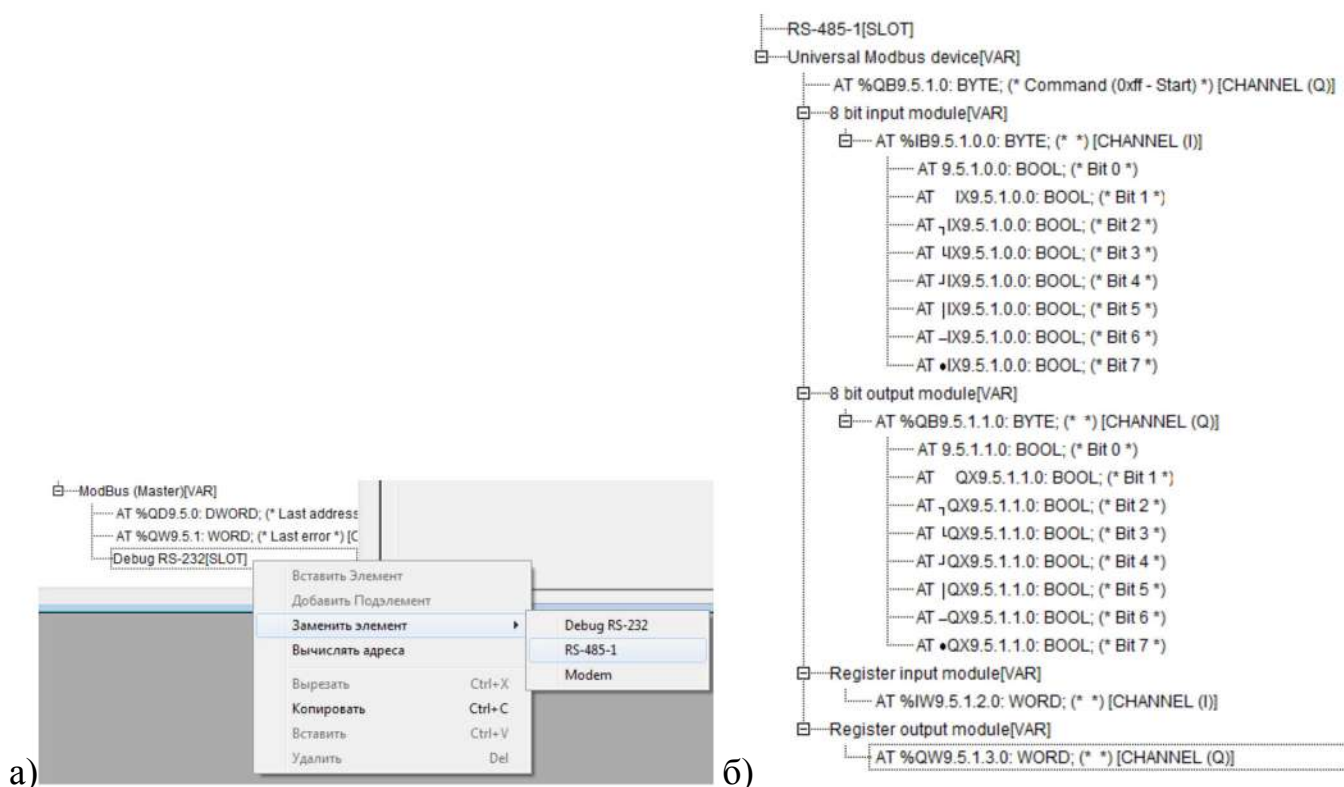


Рисунок 2.11 – Адреса сетевых переменных

Шаг 4. В правом окне в закладке «Параметры модуля» настроить необходимые параметры связи с панелью оператора (рисунок 2.8, в).

Шаг 5. Выделить строку «*ModBus (master) [VAR]*», нажать правой клавишей мыши и в появившемся меню выбрать «Добавить Подэлемент» → «*Universal Modbus device*».

Шаг 6. Выделить строку «*Universal Modbus device*», нажать правой клавишей мыши и в появившемся меню выбрать «Добавить Подэлемент» → «*8 bit input module*». В правом окне «Базовые параметры» можно указать дополнительные комментарии. Аналогично создать требуемое количество битовых и регистровых переменных. Раскрыв созданный регистр, можно увидеть (рисунок 2.11, б), какой адрес имеет каждый элемент регистра (бит, слово и т.д.).

Шаг 7. Настроить параметры переменных. Для этого выбрать соответствующий регистр. В закладке «Параметры модуля» установить параметры в соответствии с таблицей 2.4.

Таблица 2.4 – Адресация переменных

Тип переменной	Формат данных	Адрес (Address)	Команд (Command)
8 bit input module	BYTE	257	Read coil status (0x01)
8 bit input module	WORD	258	Read holding registers (0x0f)
Register output module	WORD	260	Preset single register (0x06)

Шаг 8. Аналоговые вход и выход настраивается идентично первому рассмотренному варианту связи ПЛК и панели оператора.

Шаг 9. Управляющая программа будет идентичной первому рассмотренному варианту связи ПЛК и панели оператора. Особенность лишь состоит в объявлении переменных. Сетевые переменные имеют адреса, представленные на рисунке 2.12.

```

0001 PROGRAM PLC_PRG
0002 VAR
0003   in AT %ID3.0: REAL;
0004   out AT %QD7.0: REAL;
0005   analog_in_rs485 AT %IW9.1.2.0: INT;
0006   analog_out_rs485 AT %QW9.1.3.0: INT;
0007   discret_in AT %IX0.0: BOOL;
0008   discret_in_rs485 AT %IX9.1.0.0.0: BOOL;
0009   discret_out_rs485 AT %QX9.1.1.0.0: BOOL;
0010   discret_out AT %QX1.0: BOOL;
0011 END_VAR
0012

```

Рисунок 2.12 – Объявление переменных при работе ПЛК в *master* - режиме

Программирование панели оператора осуществляется в следующей последовательности.

Шаг 1. Настроить свойства обмена между панелью оператора и программируемым контроллером для режимов работы панели оператора *Slave* и *Master*.

Настройки в режиме *Slave* будут иметь следующий вид (рисунок 2.13 а), а в режиме *Master* – рисунок 2.13, б.

Шаг 2. Расположить на экране панели оператора элементы для демонстрации сетевого обмена между ПЛК и панелью оператора: переключатель с индикацией,

– на панели инструментов нажать пиктограмму с выбранным элементом (таблица 2.3) либо в выпадающем меню «Инструменты» выбрать в закладках требуемый элемент;

– с помощью курсора, имеющего вид креста с привязанным контуром, расположить элемент, щелкнув левой клавишей мыши в требуемом месте экрана. Экран примет, следующий вид (рисунок 2.14).

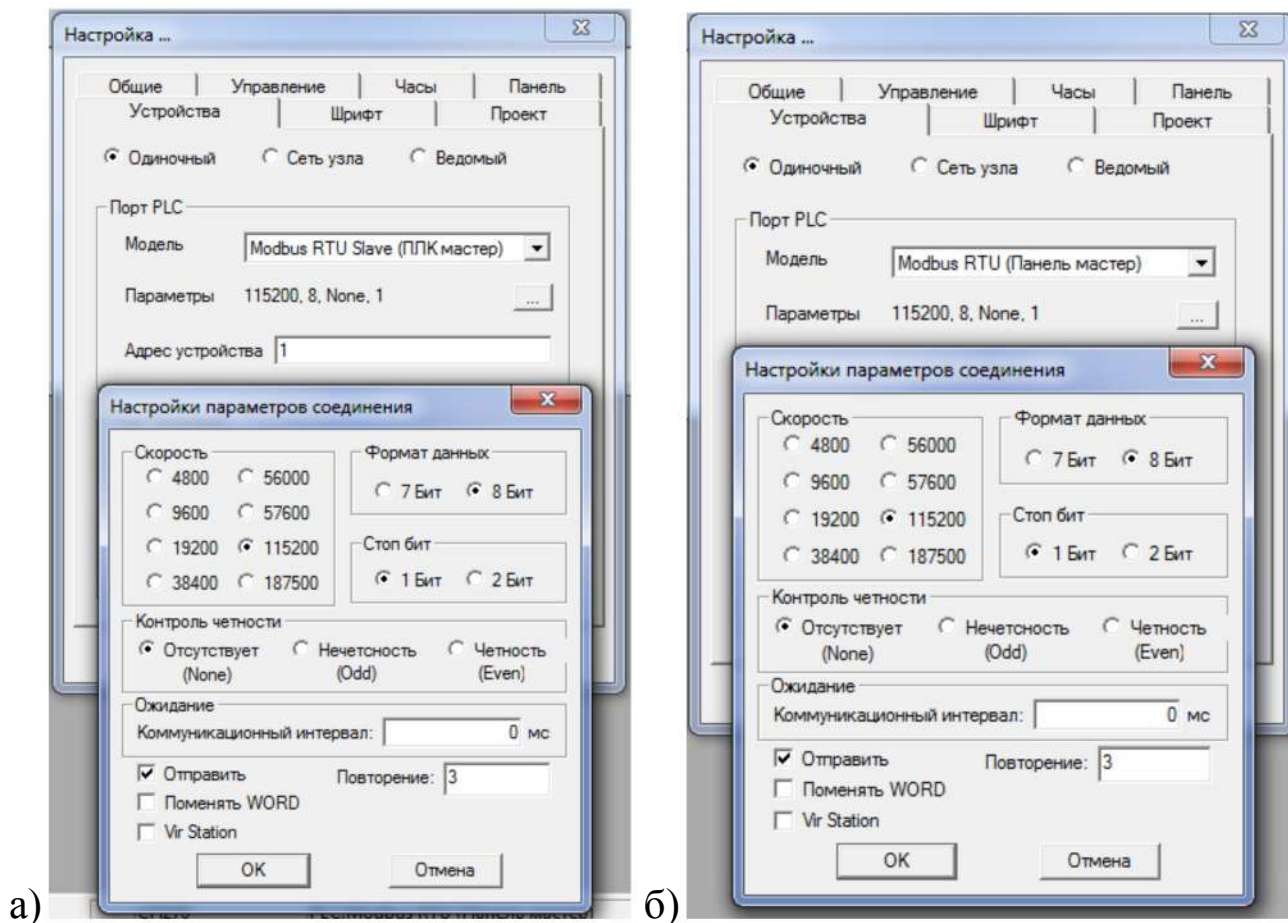


Рисунок 2.13 – Сетевые настройки панели оператора

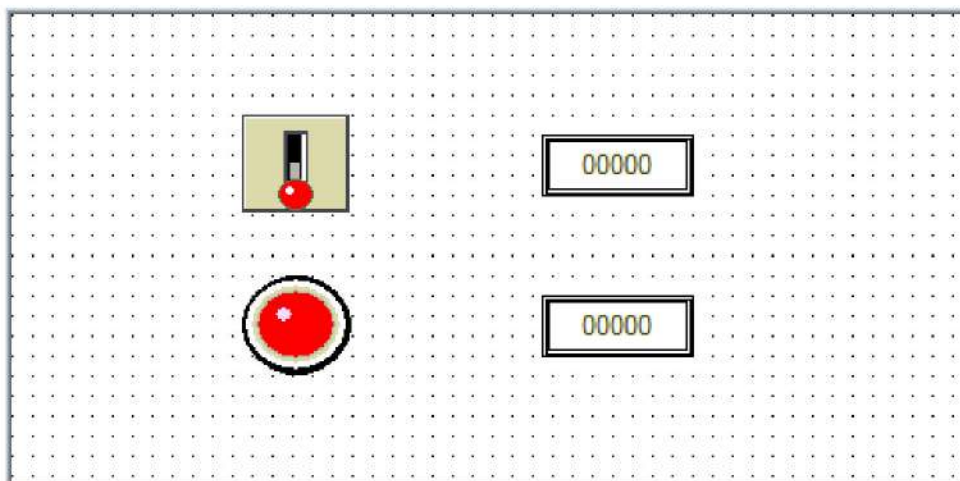


Рисунок 2.14 – Пример реализации экрана панели управления

Шаг 3. Задать параметры управления можно в непосредственно открывшемся окне либо позже, дважды щелкнув левой клавишей мыши по выбранному объекту. Настройка параметров управления каждого конкретного элемента производится в соответствии с ранее описанной методикой:

- для настройки параметров управления элементов экрана панели оператора, работающего в режиме *Master*, необходимо воспользоваться таблицей 2.5;
- для настройки параметров управления элементов экрана панели оператора, работающего в режиме *Slave*, необходимо воспользоваться таблицей 2.6.

Таблица 2.5 – Настройка параметров управления элементов экрана панели оператора, работающего в режиме *Master*

Тип элемента	Адрес устройства	Регистр		Данные	Тип
		Команда	Адрес		
Переключатель	1	0x	0	–	–
Лампа	1	0x	1	–	–
Ввод данных	1	4x	1	<i>Word</i>	<i>DEC</i>
Цифровой дисплей	1	3x	2	<i>Word</i>	<i>DEC</i>

Таблица 2.6 – Настройка параметров управления элементов экрана панели оператора, работающего в режиме *Slave*

Тип элемента	Адрес устройства	Регистр		Данные	Тип
		Команда	Адрес		
Переключатель	0	<i>PSB</i>	257	–	–
Лампа	0	<i>PSB</i>	256	–	–
Ввод данных	0	<i>PSW</i>	258	<i>Word</i>	<i>DEC</i>
Цифровой дисплей	0	<i>PSW</i>	260	<i>Word</i>	<i>DEC</i>

2.3 Пример реализации пульта оператора для управления простейшим технологическим процессом

Необходимо запрограммировать пульт оператора для управления тележкой для транспортировки деталей между пролетами цеха. Имеется два фиксированных положения *Л0* и *Л1*. На каждом положении расположена кнопка отправки в проти-

воположное положение, причем каждая из этих кнопок является виртуальным элементом панели управления.

Шаг 1. На основе описания технологического процесса составлены следующие логические уравнения:

– $B = (K_{н.в} + B) \cdot \overline{П1}$ – команда на движение тележки вперед;

– $H = (K_{н.н} + H) \cdot \overline{П0}$ – команда на движение тележки назад;

– $K_{н.в}$ – кнопка для движения тележки вперед;

– $K_{н.н}$ – кнопка для движения тележки назад.

Шаг 2. Создать и сконфигурировать проект для панели оператора. Панель оператора должна быть сконфигурирована в *master*-режиме для работы в сети RS485.

Шаг 3. В открывшемся проекте, кроме автоматически созданного, создать еще два экрана. Для этого в менеджере проекта выбрать строку «Экран», нажать правую клавишу мыши и выбрать «Добавить». В открывшемся окне (рисунок 2.15, а) выбрать имя экрана «АВТ» и номер экрана «2». Аналогично поступить для двух остальных экранов: Экран №1 – «МЕНЮ», экран №3 – «РУЧ». При необходимости можно изменить фон экрана и добавить какие-либо комментарии.

Шаг 4. Далее на экране «РУЧ» необходимо создать кнопки «Вперед» и «Назад». В панели инструментов необходимо выбрать пиктограмму  «Кнопка управления битом и установить две кнопки на экране. В окнах настроек управления необходимо выбрать настройки в соответствии с таблицей 2.7.

Шаг 5. На экранах создается необходимое количество кнопок переходов. Например, на экране «МЕНЮ» – две кнопки «АВТ» и «РУЧ» переходов на экраны соответственно автоматического и ручного режимов работы. На экране «АВТ» тоже устанавливаются две кнопки переходов на экраны «МЕНЮ» и «РУЧ». Аналогично для экрана «РУЧ» – кнопки «МЕНЮ» и «АВТ». Для установки кнопок переходов необходимо воспользоваться пиктограммой  на панели инструментов. В окне настроек кнопки перехода на экран (рисунок 2.15, б) необходимо указать номер

экрана, на который осуществляется переход, и в закладке «Кнопка» → «Содержимое» – название кнопки.

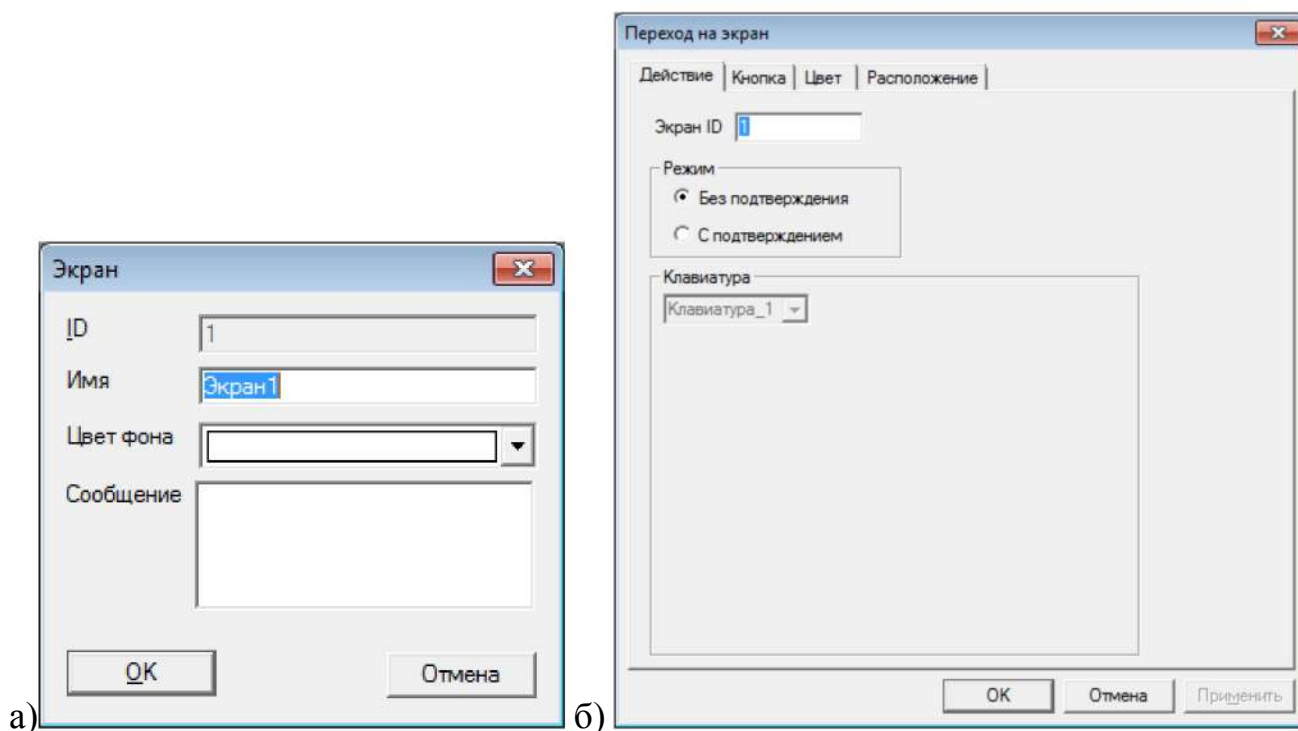


Рисунок 2.15 – Настройка свойств экрана (а) и кнопок переходов (б)

Таблица 2.7 – Настройки управления

Элемент	Регистр	Действие	Кнопка–содержимое
Кнопка 1	0x0	Мгновенно включить	Вперед
Кнопка2	0x1	Мгновенно включить	Назад
Лампа 1	0x2	—	В
Лампа2	0x3	—	Н

Шаг 6. Программирование индикаторов (лампы) на экране панели оператора ведется в ранее описанной последовательности и в соответствии с таблицей 2.7.

Шаг 7. На основе приведенных логических уравнений создается управляющая программа в среде *CoDeSys* 2.3, которая в дальнейшем записывается в контроллер ОВЕН ПЛК150. Контроллер должен быть сконфигурирован в *slave*-режиме для работы в сети *RS485*.

Таким образом, экраны, реализуемые на панели оператора, имеют вид, представленный на рисунке 2.16, а управляющая программа – на рисунке 2.17.

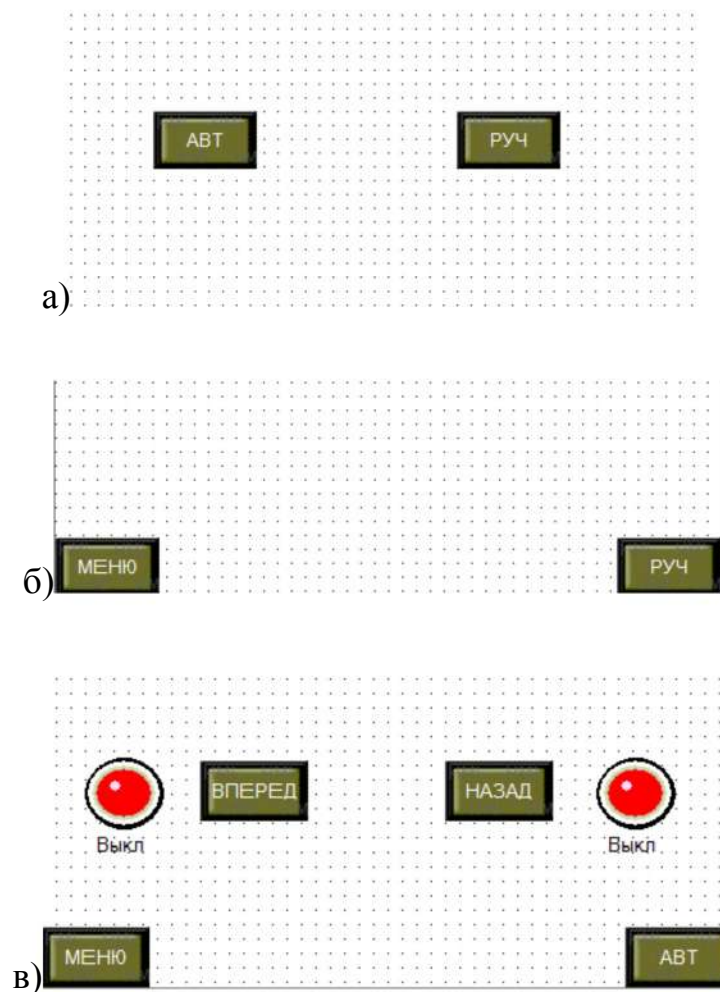


Рисунок 2.16 – Вид экранов панели оператора

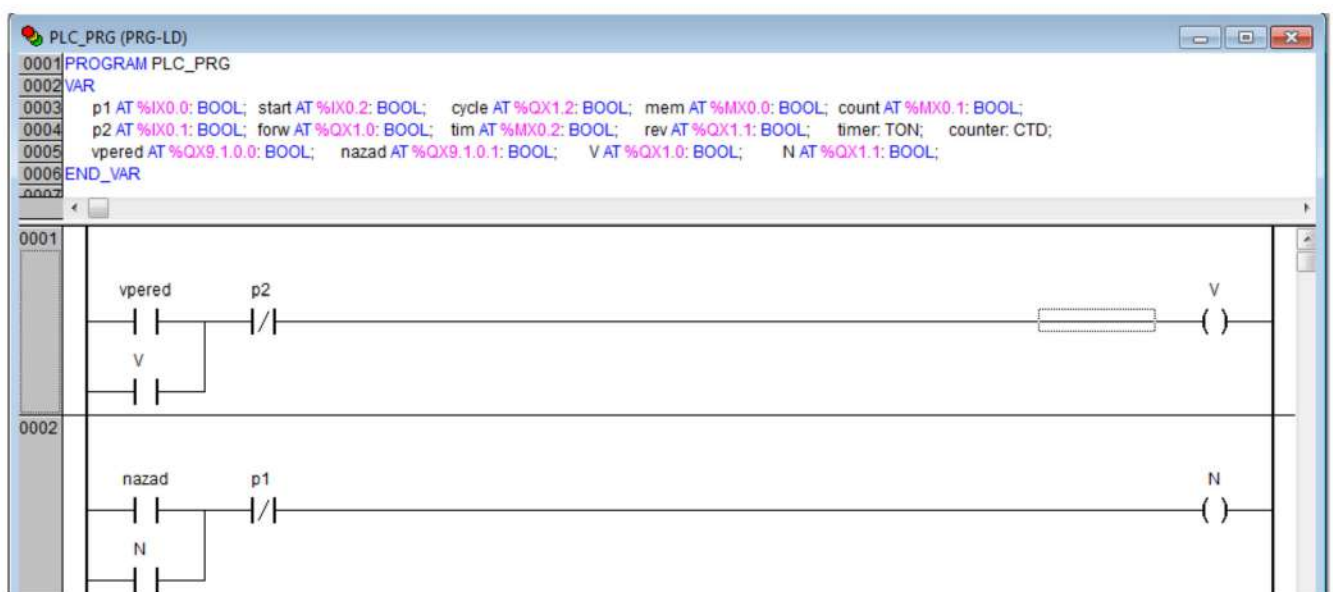


Рисунок 2.17 – Программа управления тележкой

Запись проекта в память панели оператора осуществляется в следующей последовательности:

- в выпадающем меню «Файл» выбрать пункт «Загрузить» или на клавиатуре нажать «быструю» комбинацию клавиш *CTRL+D*;
- сразу после этого начнется загрузка проекта в память панели оператора;
- отображение процесса загрузки будет отображаться на экране монитора;
- по завершению загрузки необходимо нажать «ОК».

2.4 Лабораторная работа 2 «Разработка человеко-машинного интерфейса системы управления с применением панели оператора ОВЕН СП270

Цель работы: изучить функциональные возможности панели оператора СП270 и приобрести навыки ее программирования для автоматизации управления заданным объектом.

Задание на выполнение работы:

- изучить возможности и особенности лабораторного стенда;
- изучить назначение и технические характеристики панели оператора ОВЕН СП270;
- изучить принципы программирования панели оператора;
- освоить графическую среду программного обеспечения «Конфигуратор СП200»;
- с помощью программы «Конфигуратор СП200» на ПК набрать подготовленные экраны панели оператора;
- записать проект в память панели оператора;
- сконфигурировать контроллер ОВЕН ПЛК150 для работы в сети RS485;
- набрать на ноутбуке подготовленную дома программу и откомпилировать ее;
- ввести программу в контроллер и убедиться в правильности её работы;
- при совместной работе панели оператора и программируемого контроллера проверить правильность функционирования пульта оператора;
- подготовить отчёт и сделать выводы по работе.

2.4.1 Выполнение лабораторной работы

Особенностью данной лабораторной работы является наличие объекта автоматизации, имеющего датчики, контролирующие положение механизмов объекта и исполнительные устройства, обеспечивающие перемещение механизмов объекта с заданной скоростью. Объект автоматизации, его датчики и исполнительные устройства представлены конвейерной лентой, двух датчиков крайних положений и перемещаемых по ленте предметов.

Схема лабораторной установки перед началом работы полностью собрана и никаких изменений в процессе работы не предусматривается. Программы предыдущих бригад студентов должны быть удалены из ноутбука.

В ходе лабораторной работы необходимо:

- ознакомиться с составом лабораторной установки, конструктивным исполнением компонентов стенда и соединениями между ними;
- включить компьютер;
- создать управляющую программу для ПЛК150;
- создать проект пульта управления и индикации для панели оператора СП270;
- включить блок питания сети и убедиться в наличии питания всех компонентов стенда;
- записать подготовленные проекты в ПЛК150 и панель СП270;
- при необходимости проверки правильности работы панели оператора в режимах *Master* и *Slave* для задания определенного уровня напряжения на аналоговом входе ПЛК150 можно воспользоваться потенциометром модуля «Программируемое реле». Для этого выход потенциометра AI1 модуля ПР подключить на красную клемму аналогового входа AI1 модуля ПЛК, а общую клемму M модуля ПР – к синей клемме аналогового входа AI1 модуля ПЛК. Для контроля уровня напряжения на аналоговом выходе AQ1 ПЛК можно воспользоваться мультиметром (в комплект стенда не входит);
- проверить правильность программирования контроллера и панели оператора согласно выданного задания.

2.4.2 Требования к отчёту

Отчет должен содержать:

- название работы;
- цель работы;
- описание задачи, обозначение переменных, содержательное описание формирования логических функций, логические функции, соответствующие работе объекта с учетом ручного и автоматического режимов работы;
- физические и символьные адреса переменных, отредактированные и проверенные программы для ПЛК150 и панели оператора СП270 с комментариями к ним;
- рисунки с экранов программируемого пульта и комментарии к ним;
- описание методики и экспериментальной проверки работоспособности системы автоматизации в автоматическом и ручном режимах;
- выводы по работе.

2.5 Контрольные вопросы

2.6.1 Каково назначение панели оператора?

2.6.2 Какие режимы работы есть у панели оператора?

2.6.3 Как осуществляется переход от одной картинке к другой, если пульт включает в себя несколько картинок?

2.6.4 Как обеспечивается адресация элементов программируемого пульта управления?

2.6.5 Какие этапы выполняются при создании и конфигурировании нового проекта?

2.6.6 Какие способы связи используются при обращении панели к своей внутренней памяти?

2.6.7 Какие режимы работы по протоколу Modbus поддерживают панели оператора?

2.6.8 В чем разница между режимами работы «*master*» и «*slave*»?

3 Изучение технических характеристик и основ программирования программируемого реле ОВЕН ПР114

Программируемое реле (ПР) – разновидность программируемых логических контроллеров. Основное применение программируемые реле нашли в качестве средств автоматизации локальных контуров, отдельных агрегатов машин и механизмов, для бытового применения.

На основе ПР интуитивно и понятно строятся различные системы автоматического управления, например, системы управления насосным оборудованием, сверлильными станками. Компактные размеры и простота программирования позволяют разрабатывать на базе программируемых реле элементы системы «умный дом».

Стандартными средствами описания и построения программ для данных устройств являются языки релейной логики (LD) или функциональных блоков (FBD), разработанные специально для инженеров, занятых в области автоматизации промышленности и производства.

Простота языка программирования, легкость перехода от морально устаревших систем автоматизации на базе релейно-контакторных схем к микропроцессорным устройствам, позволили программируемым реле занять надежную позицию на рынке устройств автоматизации.

3.1 Программирование реле ОВЕН ПР114

Средой программирования ПР ОВЕН является *OWEN Logic* [12]. Для запуска программы *OWEN Logic* необходимо на рабочем столе *Windows* дважды щелкнуть курсором мыши по соответствующему ярлыку. После чего откроется пустое окно программы, в котором можно создать новый проект или открыть уже созданный проект. Для этого необходимо в выпадающем меню «*File*» выбрать пункт «Новый проект». Откроется окно выбора реле, в котором необходимо выбрать заданный тип реле (рисунок 3.1). После этого нажать «ОК» и окно программы примет следующий вид (рисунок 3.2). При запуске программы также происходит автонастройка порта

связи. В случае успешной настройки в нижнем правом углу будет указан тип прибора и номер порта, через который обеспечивается связь с реле.

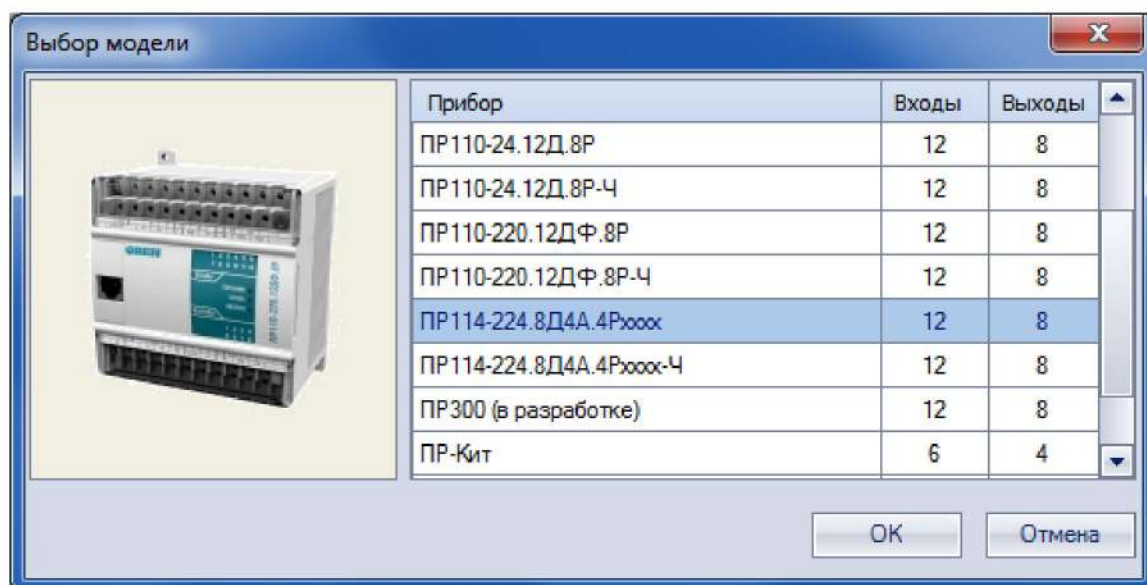


Рисунок 3.1 – Закладка диалогового окна выбора модели программируемого реле

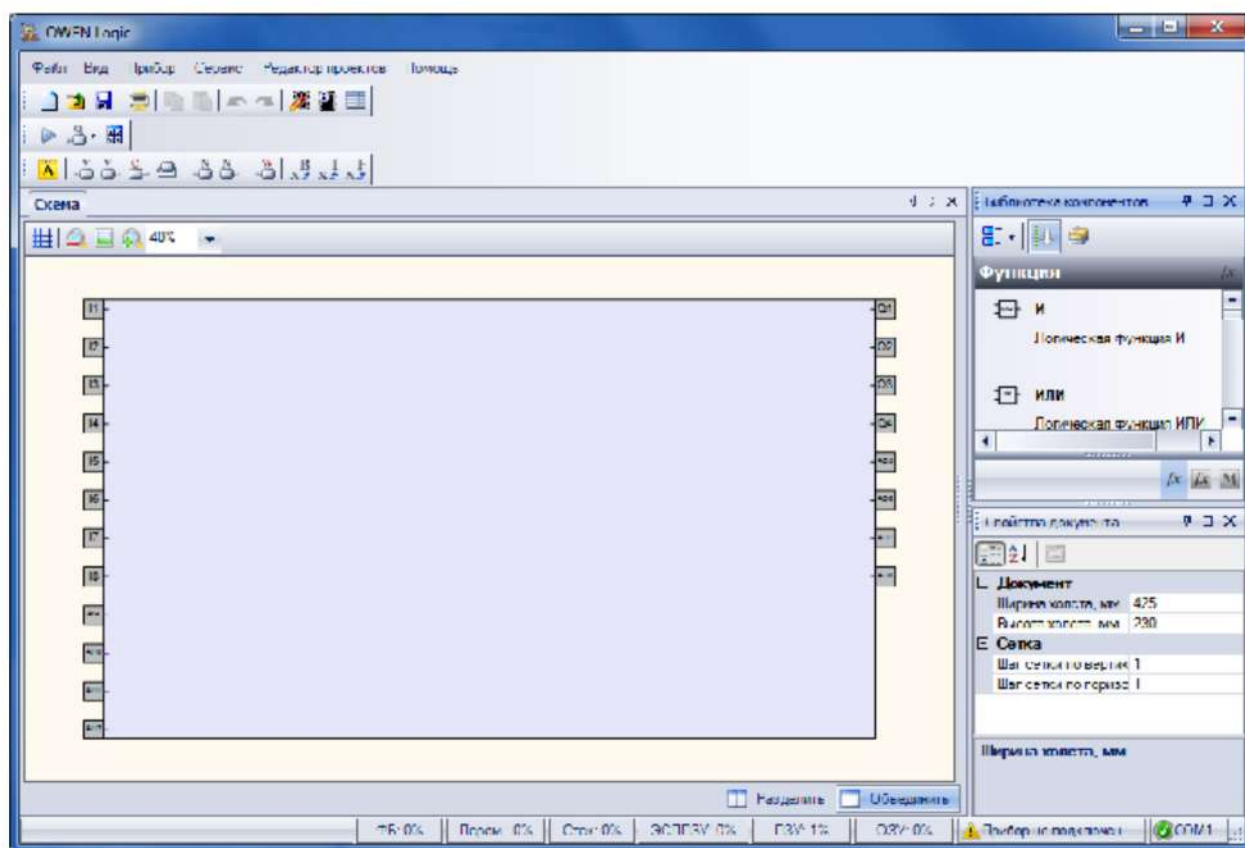


Рисунок 3.2 – Окно программы *OWEN Logic*

Настроить параметры порта также можно, выбрав пункт «Настройка порта» (рисунок 3.3) в выпадающем меню «Прибор». В открывшемся меню необходимо выбрать требуемые параметры порта, после чего нажать «ОК».

Программирование логического модуля осуществляется на языке функциональных блоков (*FBD*).

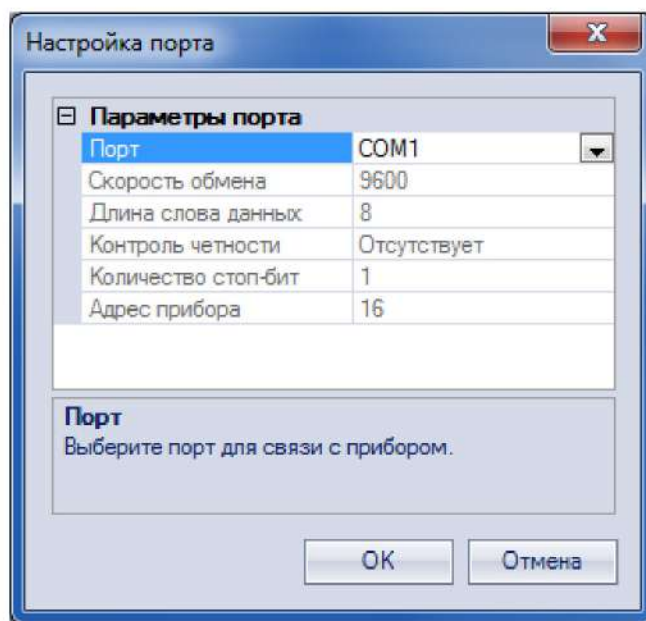


Рисунок 3.3 – Окно настройки параметров порта связи

FBD является графическим языком и наиболее удобен для программирования процессов прохождения сигналов через функциональные блоки. Язык *FBD* удобен для схемотехников, которые легко могут составить электрическую схему системы управления на «жесткой логике», но не имеют опыта программирования.

Функциональные блоки представляют собой фрагменты программ, которые могут быть многократно использованы в разных частях программы и которым соответствует графическое изображение, принятое при разработке функциональных схем электронных устройств.

Язык *FBD* может быть использован для программирования функций, функциональных блоков и программ. Функциональные блоки инкапсулируют (скрывают внутреннее содержание) данные и методы, чем напоминают объектно-

ориентированные языки программирования, но не поддерживают наследование и полиморфизм.

Типичным применением языка *FBD* является описание «жесткой логики» и замкнутых контуров систем управления. Язык функциональных блоков является удобным также для создания и пополнения библиотеки типовых функциональных блоков, которую можно многократно использовать при программировании задач промышленной автоматизации. К типовым блокам относятся блок таймера, ПИД регулятора, блок секвенсора, триггера, генератора импульсов, фильтра, и т. п.

Функциональные блоки (ФБ) являются не просто частью языка *FBD*, они применяются также для моделирования и проектирования систем автоматизации. Функциональные блоки могут быть использованы также для поддержания всего жизненного цикла системы, включая проектирование, изготовление, функционирование, валидацию и обслуживание. Описанию и применению функциональных блоков посвящены, помимо МЭК 61131-3, еще и стандарты МЭК 61499, МЭК 61804 и ISO 16484-3.

Всё рабочее окно программы (рисунок 3.2) разделено на несколько рабочих областей:

- I – главное окно проекта «Схема», в котором ведется программирование;
- II – панель инструментов для программирования, компиляции, *off/line* - симуляции и загрузки программы и т.д.;
- III – окно библиотеки компонентов, которые можно использовать при наборе программы;
- IV – окно свойств выделенного объекта (схемы или любого элемента схемы);
- V – информационная строка параметров программы и состояния связи.

В окне схемы, где ведется набор пользовательской программы, слева и справа от наборного поля располагаются точки подключения входных сигналов *I1 – I8* и выходных команд *Q1 – Q4*. На наборном поле устанавливаются элементы программы, которые соединяются друг с другом соединительными линиями в заданном пользователем порядке.

В библиотеке компонентов доступны следующие функции, которые можно разделить на несколько групп:

- И, ИЛИ, НЕ, исключающее ИЛИ (логические функции);
- ADD, SUB, MUL, DIV, MOD и т.д. (арифметические функции);
- EQ, GT, SEL, JGT(функции сравнения);
- SHL, SHR (сдвиговые функции);
- EXTRACT, RITBIT, DC32, CD32 (битовые функции).

Также в пользовательских программах можно использовать следующие функциональные блоки:

- таймеры (TP, TON, TOF);
- счетчики (CT, CTN, CTU);
- триггеры (RS, SR, RTRIG, FTRIG, DTRIG);
- генераторы (BL/NR) .

Кроме этого используются вспомогательные функции, доступ к которым осуществляется с панели инструментов (рисунок 3.4).

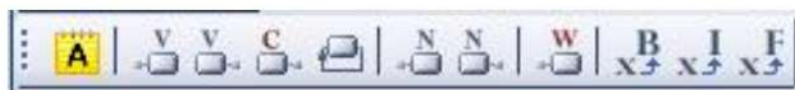


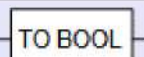
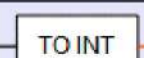
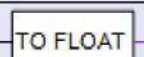
Рисунок 3.4 – Панель инструментов с вспомогательными функциями

В таблице 3.1 представлено описание функций, доступ к которым осуществляется с панели инструментов.

Таблица 3.1 – Описание функций панели инструментов

Пиктограмма	Обозначение на схеме	Описание
1	2	3
		Блок комментариев
		Создание блока выходной переменной
		Создание блока входной переменной

Продолжение таблицы 3.1

1	2	3
		Создание блока константы
		Создание обратной связи
		Создание сетевой выходной переменной
		Создание сетевой входной переменной
		Создание блока записи в ФБ
		Преобразователь в булево значение
		Преобразователь в целочисленное значение
		Преобразователь в число с плавающей точкой

Ниже рассматривается пример программирования уравнения (1.1), представленного в виде, удобном для выбора логических функций:

$$Q1 = (I1 \cdot \overline{I2}) + I3. \quad (3.1)$$

Шаг 1. Установить на поле схемы элемент НЕ (рисунок 3.5). Для этого выбрать его в закладке «Логические функции» окна библиотеки компонентов. Нажать левой клавишей мыши на иконке элемента и. переместить его в требуемое место на схеме, где отжать левую клавишу мыши.

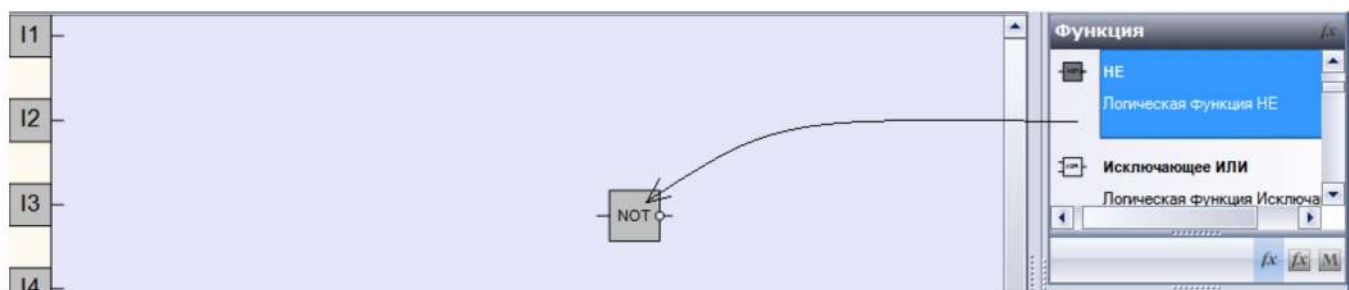


Рисунок 3.5 – Добавление логического элемента НЕ

Шаг 2. Аналогичным образом установить остальные элементы программы: И и ИЛИ. После чего программа примет вид, представленный на рисунке 3.6.

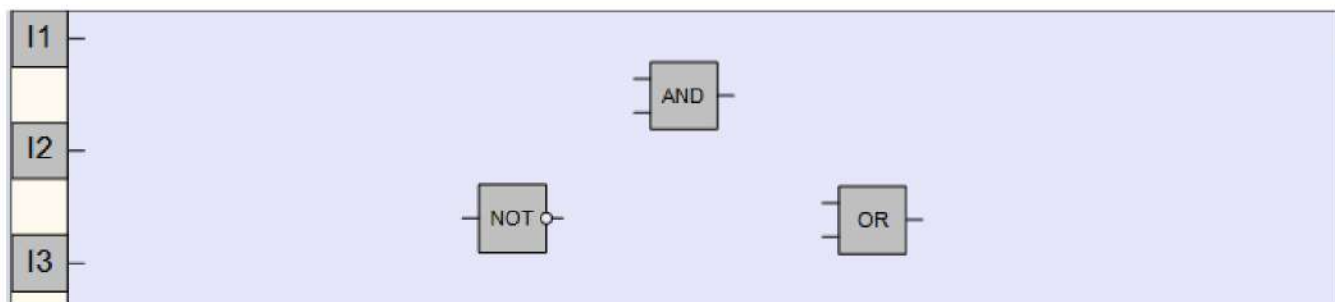


Рисунок 3.6 – Поле программы с логическими элементами НЕ, И, ИЛИ

Шаг 3. Далее необходимо соединить входы, выходы и элементы программы между собой в заданном порядке:

- выход *I1* соединить с верхним входом элемента *AND* (И);
- выход *I2* – с входом *NOT* (НЕ);
- выход *NOT*– с нижним входом *AND*;
- выход *AND* с верхним входом *OR* (ИЛИ);
- выход *I3* – с нижним входом *OR*;
- выход *OR* с входом *Q1*.

Для соединения блоков между собой необходимо установить курсор на входе/выходе блока таким образом, чтобы появился красный квадрат. После чего, нажав левую клавишу мыши, вести курсор до нужного выхода/входа другого элемента до появления красного квадрата, где нужно отпустить левую клавишу мыши.

Программа примет следующий вид (рисунок 3.7). Таким образом, разработана программа на языке FBD, реализующая логическое уравнение вида (3.1).

Для удаления какого-либо элемента программы необходимо его выделить и нажать на клавиатуре *Delete*. Для отмены предыдущего действия необходимо выбрать «Отменить» на панели инструментов.

Для редактирования какого-либо элемента программы необходимо его выделить и в окне свойств изменить данные выбранного элемента.

Кроме функций для разработки пользовательской программы можно использовать функциональные блоки.

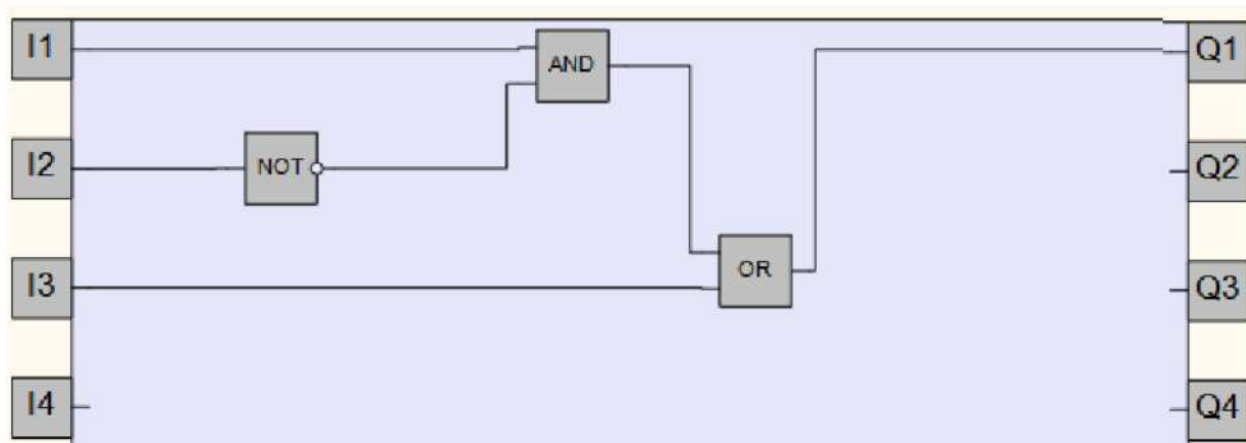


Рисунок 3.7 – Внешний вид программы после установки связей

В программируемом реле ПР114 реализуется пять типов триггеров. На рисунке 3.8 для каждого типа триггера приведено изображение блока и временные диаграммы их работы, из которых без дополнительных комментариев ясны особенности их работы.

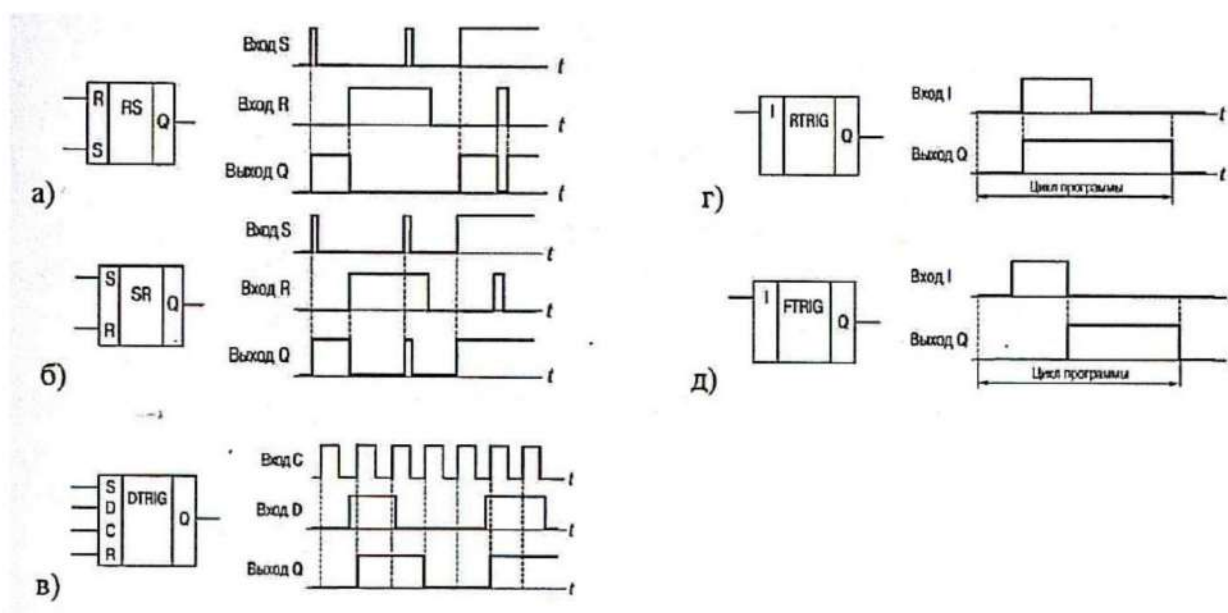


Рисунок 3.8 – Триггеры: *RS*-триггер (а), *SR*-триггер (б), *D*-триггер (в), детектор переднего фронта импульса (г), детектор заднего фронта импульса (д)

Блоки триггеров имеют следующие входы/выходы: *S* – пуск, *R* – сброс, *C* – тактовый импульс, *D* и *I* – импульс, *Q* – выход (состояние триггера). В качестве

примера рассмотрим использование RS – триггера. В данном случае он применяется для реализации следующей программы (рисунок 3.9): если кратковременно включить вход $I1$, то включится выход $Q1$, если кратковременно включить вход $I2$, выход $Q1$ выключится. Для создания программы необходимо выполнить следующие действия:

- в окне библиотеки компонентов выбрать закладку «Функциональный блок»;
- в закладке «Триггеры» выбрать RS -триггер и установить его на поле схемы;
- соединить выход $I1$ с S -входом RS -триггера, а выход $I2$ – с R - входом;
- выход Q RS -триггера соединить со входом $Q1$.

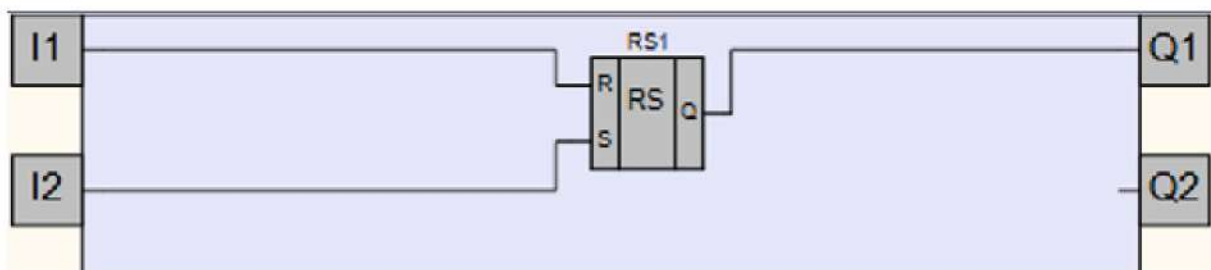


Рисунок 3.9 – Фрагмент программы с использованием RS -триггера

В программируемом реле ПР114 реализуется три разновидности таймеров. На рисунке 3.10 для каждого типа таймера приведено изображение блока и временные диаграммы их работы, из которых можно понять особенности их работы.

Блок таймера имеет следующие входы/выходы: I – вход запуска, T – уставка (заданное время), Q – выход (состояние таймера).

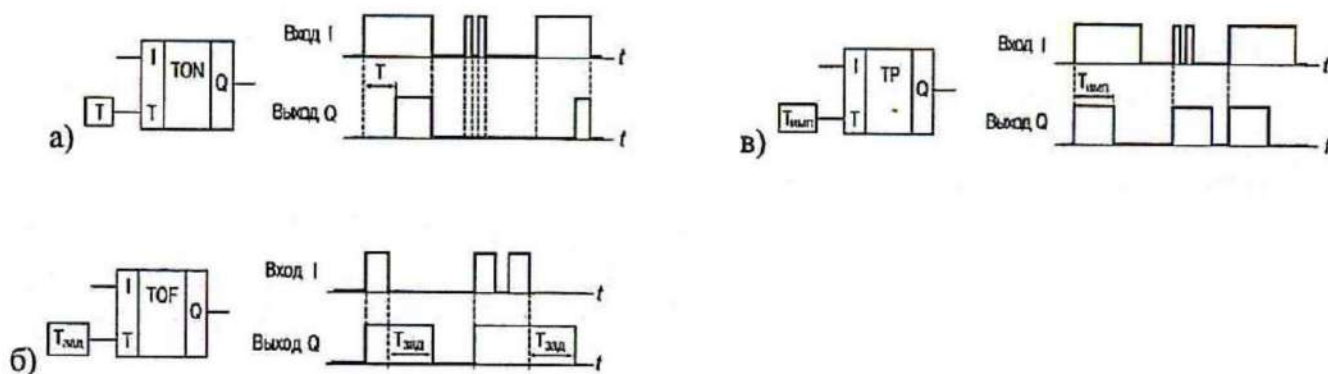


Рисунок 3.10 – Таймеры: а – с задержкой включения (TON), б – с задержкой выключения (TOF), в – формирователь импульса (TP)

Конфигурирование таймера происходит через окно настроек (рисунок 3.11, а). В нем можно назначить уставку времени, которая может быть в нескольких форматах: сек (секунды), мин (минуты), час (часы) и день (дни), а также добавить какие-то комментарии.

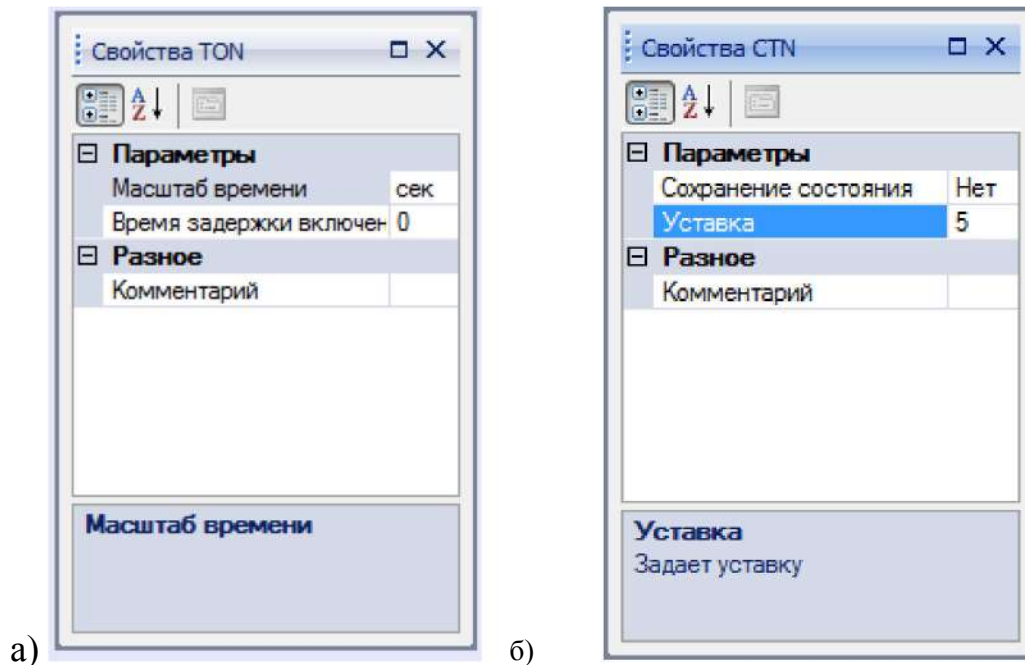


Рисунок 3.11 – Окна настроек таймера (а) и счетчика (б)

В программируемом реле ПР114 реализовано три вида счетчиков. На рисунке 3.12 представлено изображение блока и временные диаграммы, описывающие работу всех трех счетчиков. На рисунке 3.12 указаны следующие обозначения: U – инкрементный счетный вход, D – декрементный счетный вход, R – вход сброса, N – уставка, C – счетный вход, I – импульсный вход, Q – выход (состояние) счетчика.

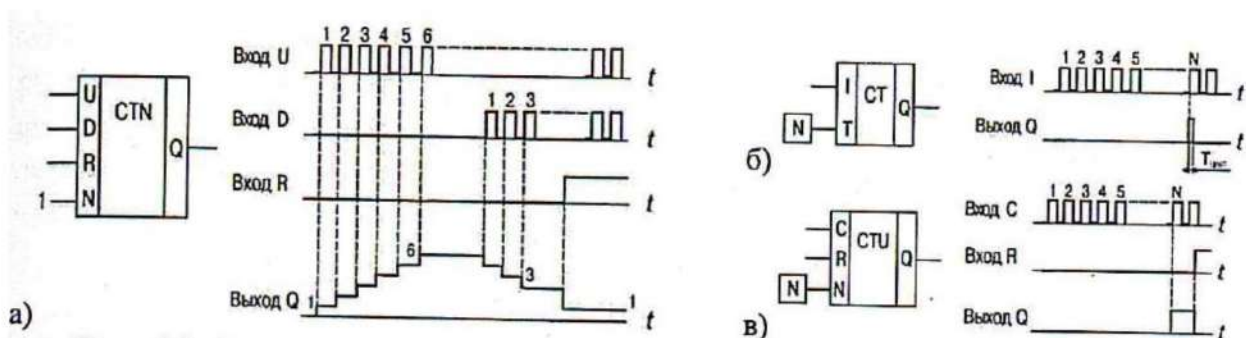


Рисунок 3.12 – Счетчики : а – универсальный (CTN), б – инкрементный счетчик с автосбросом (CT), в – инкрементный счетчик (CTU)

Конфигурирование счетчика происходит через окно настроек (рисунок 3.10, б). В нем можно назначить уставку и добавить комментарии.

Программируемое реле ПР 114 имеет четыре встроенных аналоговых входа, характеристики которых можно увидеть в соответствующем разделе данного пособия.

Для работы с аналоговыми входами в программируемом реле ПР114 доступны три режима работы компаратора (рисунок 3.13):

- сравнение на равенство (*EQ*);
- сравнение на большее значение (*GT*);
- условная операция сравнения (*SEL*).

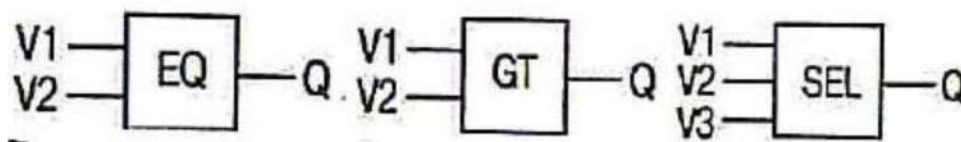


Рисунок 3.13 – Символьные обозначения функций компаратора

В качестве примера рассмотрим использование компаратора с функцией *SEL* для реализации следующей программы (рисунок 3.14): при выключенном входе *I6* на выход *AO5* передается значение с блока константы, соответственно, при включенном входе *I6* – значение с аналогового входа *AI9*:

- в окне библиотеки компонентов выбрать закладку «Функциональный блок»;
- в закладке «Функции сравнения» выбрать функцию *SEL* и установить его на поле схемы;
- для использования аналогового сигнала, поступающего со входа *AI9* необходимо использовать блок–преобразователь *TO INT* (таблица 3.1);
- для задания константы необходимо использовать блок константы (таблица 3.1), при этом в окне свойств блока в строке «тип константы» выбрать «целочисленное», в строке «значение константы» – 10;
- так как аналоговые выходы ПР114 работают с числами в формате *FLOAT* (число с плавающей точкой), то для преобразования целочисленного значения на

выходе блока *SEL* необходимо установить блок-преобразователь *TO FLOAT* (таблица 3.1);

- в окне свойств аналогового выхода *AO6* в строке «тип выхода» выбрать «аналоговый»;

- соединительными линиями подключить все блоки в заданном порядке.

После этих операций программа на схеме примет следующий вид (рисунок 3.14, а). На рисунке 3.14, б представлен пример программы с использованием функции *EQ*.

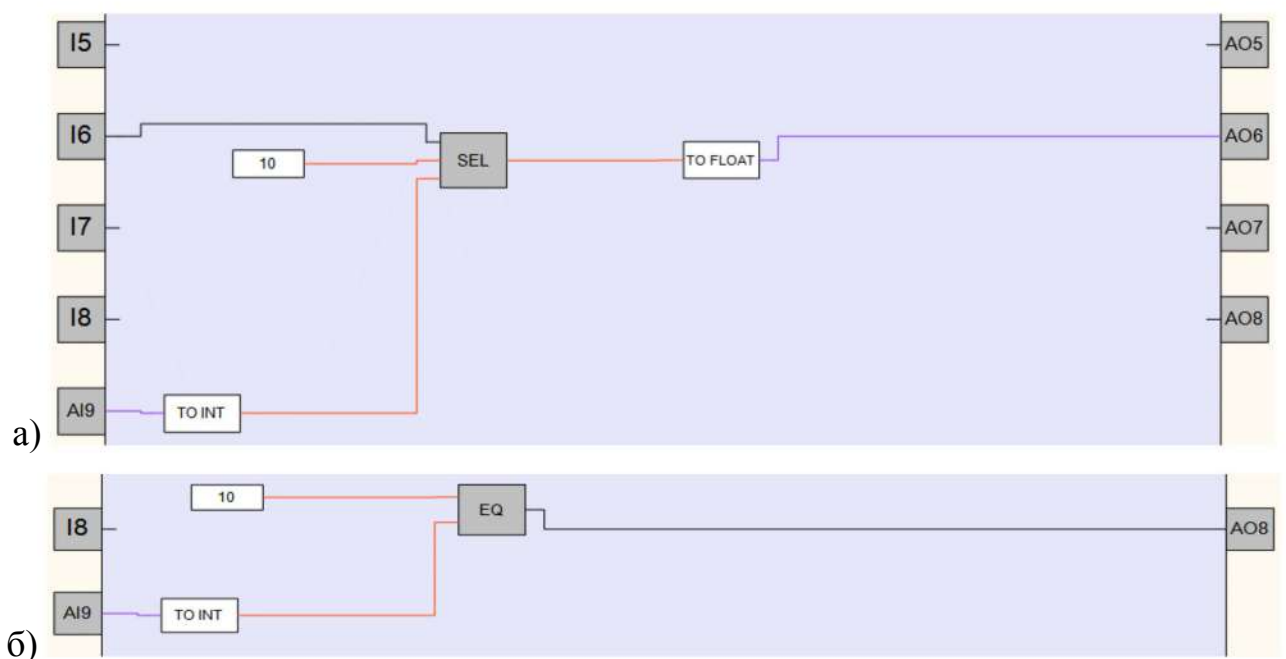


Рисунок 3.14 – Пример использования *SEL*-компаратора (а) и *EQ*-компаратора (б)

3.2 Лабораторная работа 3 «Разработка управляющей программы для программируемого реле ОВЕН ПР114»

Цель работы: ознакомиться с устройством и техническими характеристиками программируемого реле ОВЕН ПР114, приобрести навыки программирования в специализированной среде программирования *Owen Logic*, изучить основные функции и операции.

Задание на выполнение работы:

- изучить назначение, технические характеристики программируемого реле ОВЕН ПР114, основные узлы и возможности лабораторного стенда;
- изучить систему команд и принципы программирования ОВЕН ПР114;
- выполнить синтез системы автоматизации согласно выданного варианта задачи;
- составить программу для ввода в программируемое реле;
- пройти тестирование по основам программирования ОВЕН ПР114;
- освоить визуальную среду программирования *Owen Logic*;
- сконфигурировать оборудование;
- набрать на ноутбуке подготовленную программу, откомпилировать ее и ввести в программируемое реле;
- проверить работоспособность и правильность выполнения программы.

3.2.1 Общие сведения об изучаемом оборудовании

При проведении данной лабораторной работы используется программируемое реле ОВЕН ПР114. Внешний вид модуля программируемого реле представлен на рисунке 3.15. Внешний вид самого реле ОВЕН ПР114 представлен на рисунке 3.16.

В соответствующих таблицах представлено описание элементов логического модуля и его технические характеристики (таблицах 3.2 и 3.3).



Рисунок 3.15 – Внешний вид модуля



Рисунок 3.16 – Внешний вид программируемого реле ОВЕН ПР114

Таблица 3.2 – Описание элементов логического модуля

Элемент		Функция
1 Клеммный блок		Подключение входных цепей и питания контроллера
2 Разъем RS232 (Прог.)		Подключение к среде программирования <i>Owen Logic</i> . Подключение к соответствующей информационной сети для связи с ПК, ПЛК или другими устройствами автоматизации.
3 Клеммный блок		Подключение выходных цепей и интерфейса RS485 контроллера.
4 Индикаторы состояния входов		Индикация состояния входов программируемого контроллера.
5 Индикаторы состояния ПЛК	Питание	Отображает наличие питания реле.
	Связь	Отображает процесс обмена по интерфейсу UART.
	Авария	Светится при аварийных ситуациях и в процессе записи пользовательской программы в прибор. Мигает: внутреннее программное обеспечение повреждено. Непрерывно горит: записана некорректная программа.
6 Индикаторы состояния выходов		Индикация состояния выходов программируемого контроллера.

Таблица 3.3 – Технические характеристики модуля

Параметр	Значение
Модель	ОВЕНПР114
Напряжение питания, В постоянного напряжения	от 20 до 375
Потребляемая мощность, не более, ВА	10
Выходное напряжение встроенного источника питания, В постоянного напряжения	24±3
Ток нагрузки встроенного источника питания, не более, мА	140
Гальваническая изоляция	Есть
Встроенные цифровые входы/выходы	8/4
Тип входов	транзисторный
Тип выходов	реле
Встроенные аналоговые входы	4
Преобразователь для программирования	ПР-КП20
Среда программирования	<i>Owen Logic</i>
Память, байт	16384 (ПЗУ) /2560 (ОЗУ)
Количество экземпляров функциональных блоков в пользовательской программе	16384
Количество <i>Retain</i> -переменных	20
Интерфейс программирования	UART
Сетевые параметры	9600-8-none-1

Ниже представлены схемы подключения входных (рисунок 3.17) и выходных (рисунок 3.18) цепей программируемого реле ОВЕН ПР114.

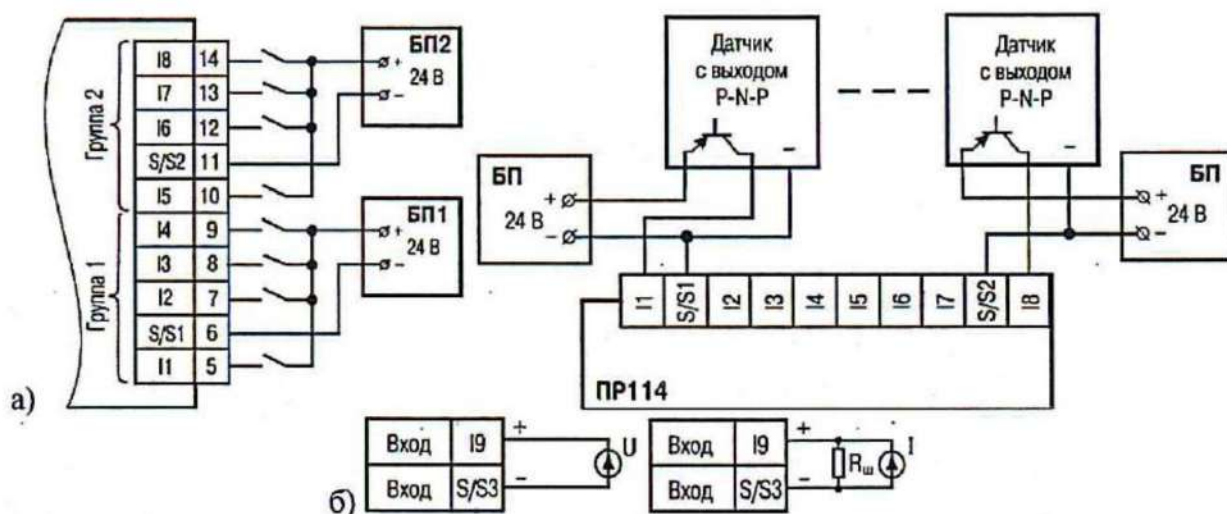


Рисунок 3.17 – Схема подключения входных цепей программируемого реле ОВЕН ПР114

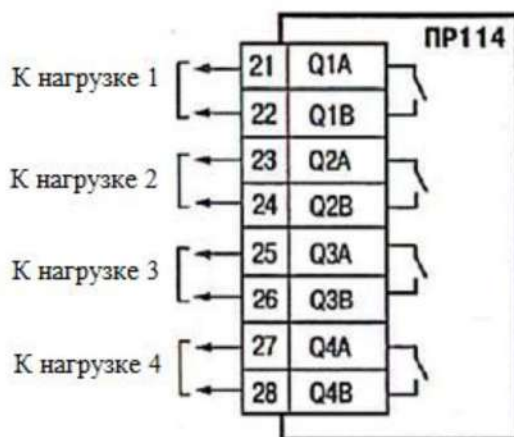


Рисунок 3.18 – Схема подключения выходных цепей реле ОВЕН ПР 114

Технические характеристики входных и выходных цепей программируемого реле ОВЕН ПР114 представлены в таблице 3.4. Реле ПР114 является устройством со свободно-программируемой логикой, работа которого определяется программой, которая разрабатывается на ПК в соответствующей среде программирования пользователем.

Пользовательская программа записывается в энергонезависимую *Flash*-память реле. По окончании процедуры записи реле автоматически перезагрузится, и

программа пользователя запустится на выполнение. Также программа пользователя начинает выполняться после подачи напряжения питания. Время установления рабочего режима после записи пользовательской программы не более 0,5 с.

По включению напряжения питания, перед началом выполнения пользовательской программы, реле выполняет настройку аппаратных ресурсов и самотестирование. Самотестирование включает в себя проверку целостности встроенного программного обеспечения прибора и корректности пользовательской программы. Если самотестирование прошло успешно, реле переходит к основной работе (Рабочий режим). В противном случае, прибор переходит в Аварийный режим.

Для подготовки программ при проведении лабораторных работ служит программа *Owen Logic* – специализированная среда программирования на основе визуального языка графических диаграмм *FBD (Function Block Diagram)*.

Таблица 3.4 – Технические характеристики входных и выходных цепей программируемого реле ОВЕН ПР114

Параметр	Значение
1	2
<i>Цифровые входы</i>	
Количество	8 (<i>I1 – I8</i>)
Потенциальная развязка, кВ переменного напряжения	1,5 (групповая по 4 входа)
Напряжение логической 1, В постоянного напряжения	От 15 до 30 (при токе от 2,0 до 4,0 мА)
Напряжение логического 0, В постоянного напряжения	От 3 до 5 (при токе не более 0,1 мА)
Тип подключаемых устройств	Механические коммутационные устройства и устройства с выходными транзисторными точками <i>PnP</i> – типа
<i>Аналоговые входы</i>	
Количество	4 (<i>I9 – I12</i>)
Диапазон измерения, В постоянного напряжения	От 0 до 10
Период обновления результатов измерения четырех каналов, мс	0,5
Предел погрешности, % от диапазона измерения	±0,5

Продолжение таблицы 3.4

1		2
<i>Цифровые выходы</i>		
Количество		4 (Q1 – Q4)
Тип		Реле (замыкающий контакт)
Электрическая изоляция, кВ переменного напряжения		1,5 (индивидуальная)
Длительный ток нагрузки, А	для цепи 30 В постоянного напряжения	5
	для цепи 250 В переменного напряжения	10
Время переключения из 0 в 1 / из 1 в 0, мс		20

Программное обеспечение *Owen Logic* позволяет пользователю разрабатывать коммутационную программу по собственному алгоритму с последующей записью ее в энергонезависимую память реле.

Одновременно среда программирования может работать только с одним проектом. При создании проекта не требуется подключения ПР.

Последовательность операций при разработке проекта следующая:

- 1) установка среды программирования *OWEN Logic* на ПК;
- 2) запуск среды программирования;
- 3) создание нового проекта (для конкретной модели ПР) или открытие существующего проекта для редактирования;
- 4) сохранение проекта на жестком диске ПК;
- 5) отладка проекта в режиме симуляции;
- 6) загрузка проекта в ПР.

3.2.2 Пример разработки пользовательской управляющей программы для программируемого реле ПР114

В качестве примера программирования реле ПР114 рассмотрим программу управления движением тележки.

Условие задачи. Тележка перемещается по прямолинейному пути. Имеется два фиксированных положения *П0* и *П1*. Исходное положение тележки – *П0*.

Условие задачи подробно представлено в пункте 1.4.7.

Решение. В таблице 3.5 представлена адресация команд и сигналов для реле ПР114. На рисунке 3.19 представлена разработанная программа, реализующая управление движением тележки. Для удобства проектирования в программе применены промежуточные переменные, которые используются в качестве входных и выходных сигналов отдельных сегментов программы.

Таблица 3.5 – Адресация команд и сигналов для реле ПР114

Команда, сигнал	П0	П1	Пуск	В	Н	КЦ	Рп	СЧ	$П^{\uparrow}$
Адрес	<i>I1</i>	<i>I2</i>	<i>I3</i>	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>	–	<i>CTU1</i>	<i>TON1</i>
Символ	П0	П1	Пуск	Вперед	Назад	КЦ	Рп	СЧ	–

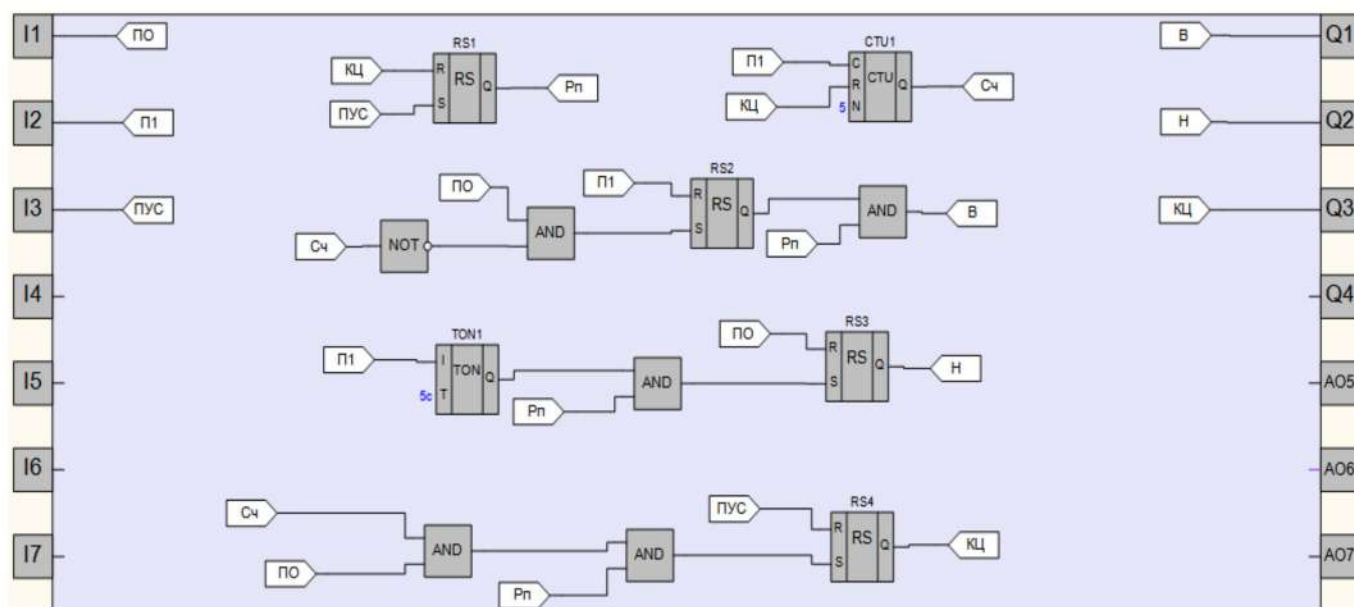


Рисунок 3.19 – Программа управления движением тележки

3.2.3 Загрузка программы

Загрузка программы осуществляется следующим образом: в выпадающем меню «Прибор» необходимо выбрать пункт «Записать программу в прибор». После че-

го начнется загрузка программы в память программируемого реле. Процесс загрузки отображается с помощью шкалы процесса.

3.2.4 Отладка программы

Для предварительной отладки пользовательской программы в программном обеспечении *OWEN Logic* доступен режим симулятора. Для перехода в этот режим и запуска программы в режиме симуляции необходимо выполнить следующую последовательность действий:

- на панели инструментов нажать пиктограмму «Режим симулятора». При этом окно схемы примет вид, представленный на рисунке 3.20;
- в появившейся панели симулятора (рисунок 3.21) нажать кнопку «Запуск симулятора в режиме реального времени»;
- нажимая на изображения входов, проверить правильность выполнения составленной программы.

Состояния входов и выходов отдельных блоков можно наблюдать соответственно около входной или выходной линии блока. Также можно контролировать состояния таймеров и счетчиков.

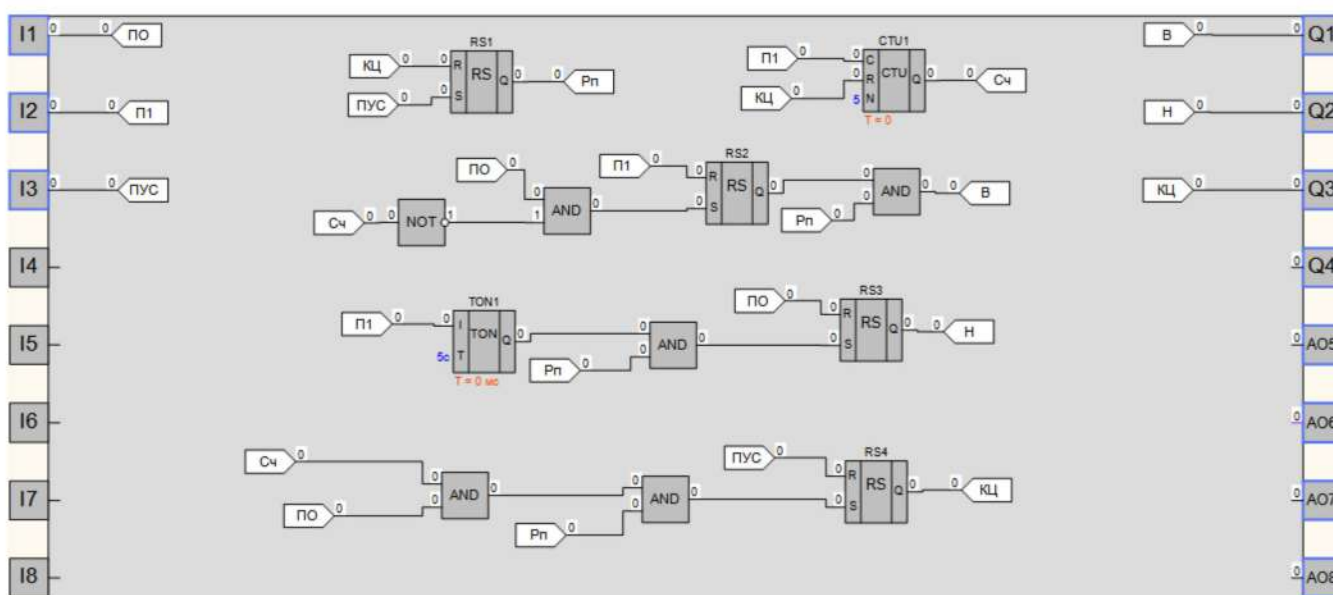


Рисунок 3.20 – Режим симулятора



Рисунок 3.21 – Панель управления симулятора

3.2.5 Порядок проведения лабораторной работы

При подготовке к лабораторной работе студент должен в соответствии с таблицей 1.12 выбрать свой вариант задания.

Для заданного варианта задачи необходимо разработать алгоритм ее решения. В соответствии с заданием определиться с адресацией входных и выходных переменных. Затем определиться с адресацией вводимых промежуточных переменных и параметрами используемых таймеров и счетчиков. Составить программу для программируемого реле ПР114.

С помощью программного обеспечения *OWEN Logic* набрать подготовленную программу. При необходимости проверить правильность работы программы на симуляторе, после чего записать ее в ПР114. Затем запустить программу в работу и убедиться в правильности ее работы. При наличии ошибок в работе устранить их и продемонстрировать преподавателю работу правильно функционирующей программы.

3.2.6 Требования к отчёту

Отчет должен содержать:

- название работы;
- цель работы;
- условия задачи по варианту, принятые обозначения переменных, логические функции для выходных и промежуточных переменных, при необходимости циклограмму работы оборудования;
- логические функции в адресах программируемого реле ПР114;
- программу для реализации системы управления на реле ПР114;
- методику экспериментальной проверки функционирования реализованной системы управления и результаты проверки;
- выводы по работе.

3.3 Контрольные вопросы

3.3.1 В чем достоинства программируемого реле?

3.3.2 Что является стандартными средствами описания и построения программ для программируемого реле?

3.3.3 Что является средой программирования программируемого реле OWEN?

3.3.4 Как осуществляется выбор необходимого типа программируемого реле?

3.3.5 Каково отличие входов $I_9 - I_{12}$ от входов $I_1 - I_8$?

3.3.6 Что представляют собой функциональные блоки?

3.3.7 Для чего применяются функциональные блоки?

3.3.8 На какие рабочие области разбито окно программы *OWEN Logic*?

3.3.9 Какие функциональные блоки используются в пользовательских программах?

3.3.10 Как реализуются входные и выходные переменные?

3.3.11 Какие типы таймеров доступны для программирования?

3.3.12 Как в пользовательской программе реализуется таймер?

3.3.13 Какие типы счетчиков доступны для программирования?

3.3.14 Как в пользовательской программе реализуется счётчик?

3.3.15 Как в пользовательской программе реализуется самоподхват в уравнениях типа $Q1 = (I1 + Q1) \cdot I2$?

3.3.16 Как в пользовательской программе реализуется аналоговый компаратор?

3.3.17 Для чего предназначен потенциометр на лицевой панели модуля программируемого реле?

4 Изучение систем автоматизации технологических процессов с использованием программируемого реле ОВЕН ПР114 (1 вариант симуляции объекта)

4.1 Общие сведения об объекте автоматизации

При проведении занятий используется модуль имитации физического объекта «Методическая печь» (рисунок 4.1), который предназначен для демонстрации работы методической нагревательной печи. Все исполнительные органы методической печи (транспортеры, подъемная крышка) реализованы в виде мнемосхем на лицевой панели модуля. Имитация выполняемых действий осуществляется с помощью светодиодов и имеет два режима работы: ручной – управление имитацией с помощью пульта ручного управления, расположенного также на лицевой панели, автоматический – управление осуществляется от программируемого логического контроллера либо от программируемого реле с помощью записанных в них управляющих программ. Выбор режима производится с помощью соответствующего тумблера «Руч./Авт.».

Для подачи новой детали в процессе имитации работы методической печи и сброса имитации в начало во время исполнения программы на лицевой панели модуля расположены соответствующие кнопки «Подача детали» и «Сброс имитации». Через блоки гнезд «Датчики» и «Двигатели» осуществляется подключение имитации датчиков и исполнительных устройств методической печи соответственно к входам и выходам ОВЕН ПР114. Питание модуля включается с помощью кнопки с фиксацией и подсветкой. Технические характеристики модуля представлены в таблице 4.1.

Таблица 4.1 – Технические характеристики модуля

Параметр	Значение
Количество датчиков	6
Количество исполнительных механизмов	3
Режимы работы	ручной /автоматический

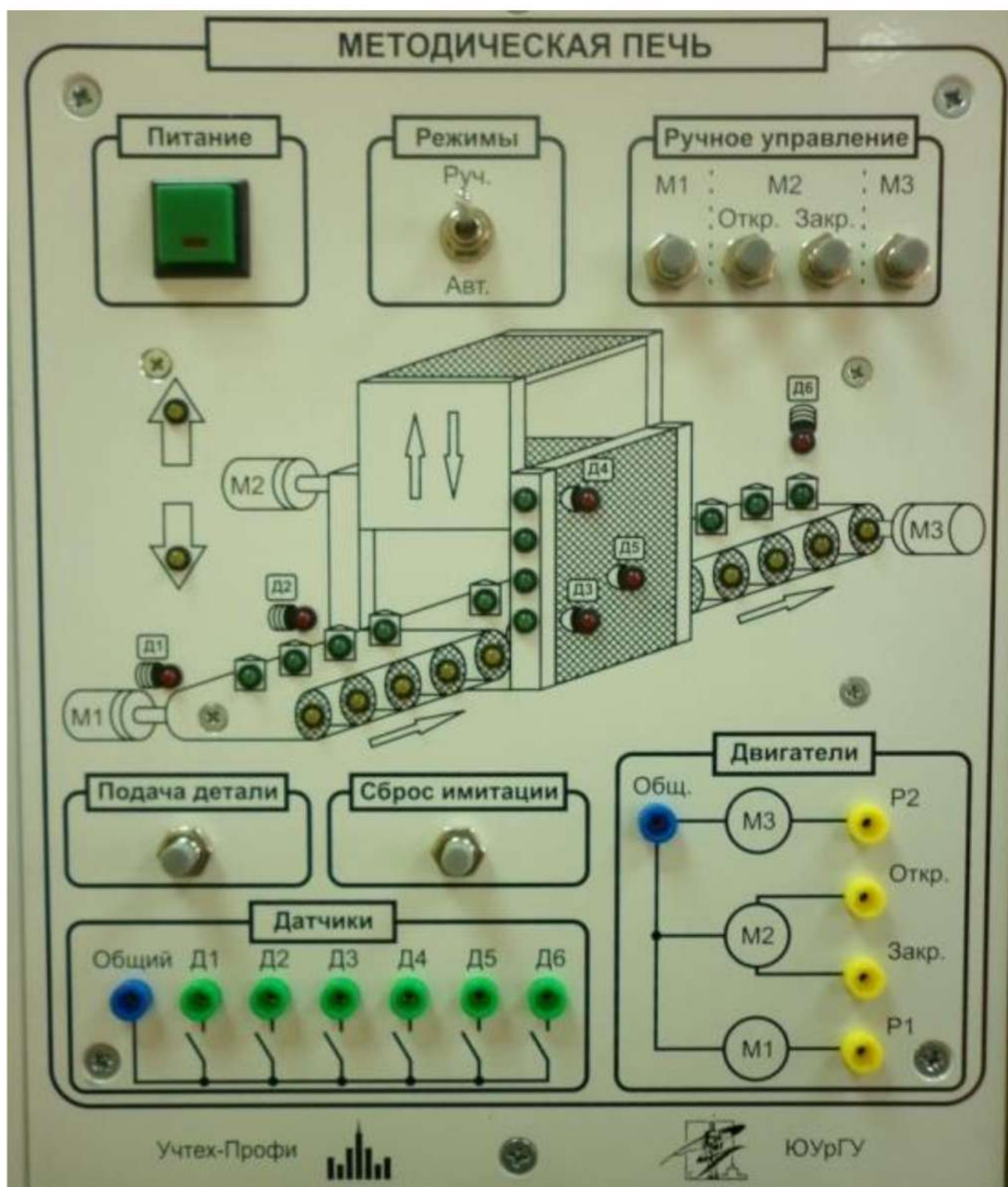


Рисунок 4.1 – Модуль имитации физического объекта (модуль ИФО)

Также при проведении занятия используется модуль программируемого реле ОВЕН ПР114. Описание модуля и краткие технические характеристики изучаемого оборудования приведено в разделе 3.

4.2 Лабораторная работа 4 «Разработка программы для управления работой модели методической печи»

Цель работы: приобрести навыки алгоритмизации управления технологическим объектом и реализации системы автоматизации управления объектом на базе программируемых устройств автоматики (программируемого реле ОВЕН ПР114).

Задание на выполнение работы:

- уточнить адреса датчиков имитации физического объекта, сигналы с которых поступают на входы программируемого реле ОВЕН ПР114;
- уточнить адреса управляющих сигналов и команд, поступающих из реле ПР114 на исполнительные элементы имитации физического объекта;
- составить алгоритм управления автоматизируемым объектом (составить логические функции для выходных сигналов и промежуточных переменных);
- записать логические функции в адресах ОВЕН ПР114;
- составить программу для ОВЕН ПР114;
- ввести подготовленную программу для программируемого реле в ПК, провести ее компиляцию и записать её в память реле;
- собрать схему для проверки правильности работы системы автоматизации на базе программируемого реле ОВЕН ПР114;
- подать питание на модуль имитации физического объекта «Методическая печь»;
- подавая с имитации пульт управления, расположенного на лицевой панели модуля «Методическая печь», команды, проверить работу системы управления, как в ручном, так и в автоматическом режимах работы;
- составить отчет и заключение по работе.

4.2.1 Общие положения

Логические функции, описывающие работу системы автоматизации, предлагается записывать на основе содержательного описания работы системы автомати-

зации. Последовательность подготовки программ для логического модуля будет следующей.

Шаг 1. Изучается объект автоматизации заданного варианта. Уточняется последовательность работы оборудования, уточняется назначение датчиков, число и назначение исполнительных устройств. Уточняется, какие элементы будут использоваться для управления заданным объектом как в автоматическом, так и в ручном режимах работы.

Шаг 2. Для каждой выходной и для каждой промежуточной переменной составляется содержательное описание, по которому затем непосредственно записывается логическая функция (уравнение). Содержательное описание – это словесный портрет включения переменной, условий ее сохранения и условий ее отключения. Логические функции составляются непосредственно по содержательному описанию, основываясь на том, что союзам разговорной речи И, ИЛИ, НЕ соответствуют логические операции соответственно конъюнкция, дизъюнкция и инверсия. В качестве элементов памяти рекомендуется использовать самоблокировки. При этом следует помнить, что наличие в одном логическом уравнении двух или более самоблокировок часто приводит к неработоспособности системы. Поэтому такие функции необходимо подвергнуть тщательному анализу на их работоспособность. Если они оказались действительно неработоспособными, то следует избавиться от самоблокировок введением промежуточных переменных.

Шаг 3. Полученные логические функции необходимо переписать в адресах программируемого реле и по ним составить программу в соответствии с изложенными в разделе 3 правилами составления программы.

4.2.2 Пример разработки управляющей программы для реле ОВЕН ПР114 для управления имитацией методической печи

В качестве примера рассмотрим один из вариантов управления имитацией методической печи. Упрощенный вид имитации методической печи представлен на рисунок 4.1. Описание работы приведено выше.

Алгоритм управления методической печью представлен ниже. Возможно два режима работы методической печи: ручной и автоматический.

В ручном режиме управление имитацией методической печи полностью осуществляется с лицевой панели модуля ИФО с помощью тумблера переключения режима работы «Руч. / Авт.» и четырех кнопок, с помощью которых происходит включение в работу отдельных приводов методической печи.

В автоматическом режиме работа осуществляется с использованием программы, записанной в память программируемого реле. При срабатывании датчика Д1 (наличие детали) происходит включение рольганга 1. Он работает до срабатывания датчика Д2, после чего останавливается. Включается подъем заслонки печи. После полного открытия крышки (включен датчик Д4) одновременно включаются рольганги 1 и 2. После отключения датчика Д2 рольганг 1 выключается, а рольганг 2 работает до срабатывания датчика Д5, после чего также останавливается. Крышка закрывается. После полного закрытия крышки (включен датчик Д3) начинается процесс нагрева, который длится 10 с. По истечении времени нагрева крышка поднимается. После полного открытия крышки включается в работу рольганг 2, который работает до срабатывания датчика Д6, где деталь уходит с рольганга 2 на следующую технологическую линию, а крышка печи опускается. После завершения цикла система приходит в исходное состояние и ожидает появления новой детали.

Команда на движение рольганга 1 «Р1» возникает в двух случаях: когда есть наличие детали в положении Д1 и сохраняется до срабатывания датчика Д2 и при наличии детали в положении Д2 и полностью открытой заслонке печи и сохраняется до перехода детали в положение Д5. Этому содержательному описанию команды «Р1» соответствует система логических функций:

$$P11 = (Д1 + P11) \cdot Д2,$$

$$P12 = (Д2 + P12) \cdot Д4 \cdot Д5,$$

$$P1 = P11 + P12,$$

где $P11$, $P12$ – промежуточные переменные;

$P1$ – команда на движение рольганга 1;

$D2, D5$ – сигнал отключения датчиков $D2$ и $D5$ соответственно;

$D1, D2, D5$ – сигнал срабатывания соответствующего датчика.

Команда на движение заслонки печи вверх «Откр.» возникает также в двух случаях. Во-первых, деталь находится в положении $D2$, нет памяти о том, что заслонка открывалась в текущем цикле, заслонка находится в полностью закрытом состоянии и сохраняется до тех пор, пока не сработает датчик открытого положения заслонки $D4$. Во-вторых, когда деталь находится в закрытой печи более 10 с и сохраняется до тех пор, пока заслонка полностью не откроется:

$$\begin{aligned}O1 &= (D3 + O1) \cdot D4 \cdot \overline{Z2} \cdot D2, \\O2 &= (D5^{10c} \cdot D3^{10c} + O2) \cdot D4, \\Откр &= O1 + O2,\end{aligned}$$

где $Откр$ – команда на движение заслонки печи вверх;

$O1, O2$ – промежуточные переменные;

$D4$ – сигнал отключения датчика открытого положения заслонки $D4$;

$D3$ – сигнал соответствующий полному закрытию крышки;

$D5^{10c}, D3^{10c}$ – сигналы срабатывания датчиков, с задержкой 10 с;

$\overline{Z2}$ – инверсное значение промежуточной переменной $Z2$.

Команда на движение заслонки печи вниз «Закр.» возникает в двух случаях. Во-первых, рольганги 1 и 2 выключены, деталь находится в положении $D5$, заслонка находится в открытом состоянии и сохраняется до тех пор, пока не сработает датчик закрытого положения заслонки $D3$. Во-вторых, когда рольганг 1 выключен, деталь вышла из печи, заслонка печи открыта и сохраняется до тех пор, пока заслонка полностью не закроется. Этому содержательному описанию команды «Закр.» соответствует система логических функций:

$$\begin{aligned}Z1 &= (D4 + Z1) \cdot D3 \cdot p1 \cdot D5, \\Z2 &= (D4 + Z2) \cdot D3 \cdot p1 \cdot D5, \\Откр &= Z1 + Z2.\end{aligned}$$

Промежуточная переменная $p1$ выполняет роль памяти о том, что деталь побывала в печи. Она сохраняется до появления новой детали на рольганге 1 в положении Д1:

$$p1 = (Д5 \cdot Д3 + p1) \cdot Д1.$$

Команда на движение рольганга 2 «P2» возникает также в двух случаях:

- во-первых, когда заслонка печи открыта, деталь находится в положении Д2 и сохраняется до тех пор пока деталь не окажется в положении Д5;
- во-вторых, когда заслонка печи открыта, деталь находится в положении Д5, есть память $p3$ и сохраняется до перехода детали в положение Д6.

Этому содержательному описанию команды «P2» соответствует система логических функций:

$$\begin{aligned} P21 &= (Д2 + P21) \cdot Д5 \cdot Д4, \\ P22 &= (Д4 \cdot Д5 \cdot p1 + P22) \cdot Д6, \\ P2 &= P21 + P22. \end{aligned}$$

Адресация команд и сигналов, используемых в программе управления объектом, представлена в таблице 4.2.

В соответствии с представленными функциями и таблицей соответствия сигналов и команд адресам ОВЕН ПР114 на рисунке 4.2 представлен пример программы управления имитацией физического объекта «Методическая печь» для программируемого реле.

Необходимо набрать данную программу, провести ее компиляцию, записать в память программируемого реле ОВЕН ПР114, после чего проверить правильность ее работы.

После того, как реле подготовлено к работе, необходимо приступить непосредственно к управлению ИФО «Методическая печь». Для этого необходимо выполнить следующие операции:

Таблица 4.2 – Адресация команд и сигналов, используемых в программе управления объектом

Обозначение функции	Адрес в ПР114	Обозначение функции	Адрес в ПР114
<i>Входные сигналы</i>		<i>Промежуточные переменные</i>	
Д1	I1	P11	P11
Д2	I2	P12	P12
Д3	I3	O1	O1
Д4	I4	O2	O2
Д5	I5	31	31
Д6	I6	32	32
<i>Выходные сигналы</i>		P21	P21
P1	Q1	P22	P22
P2	Q2	p1	p1
Откр.	Q3		
Закр.	Q4		

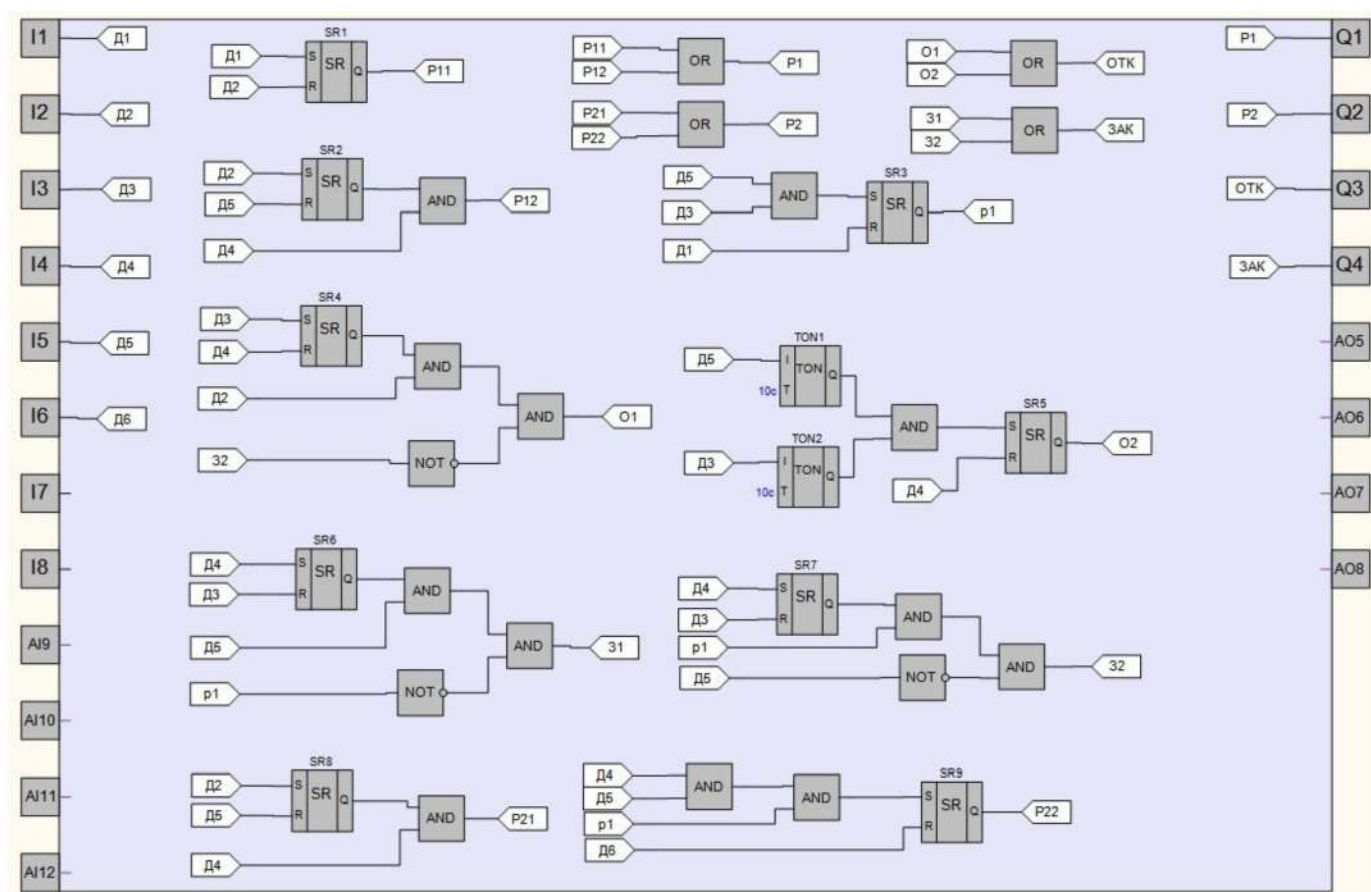


Рисунок 4.2 – Программа для реле ПР114 на языке FBD

- собрать схему для проверки правильности набранной программы в соответствии с адресацией. При этом общие клеммы модуля «Методическая печь» необходимо подключить к клемме «+» модуля программируемого реле;
- на модуле ИФО выбрать режим «Авт.»;
- с помощью кнопок «Подача детали» и «Сброс имитации», расположенных на лицевой панели ИФО, управлять работой «методической печи». При обнаружении ошибок в работе объекта необходимо произвести необходимые изменения в программе ОВЕН ПР114, добиваясь полного соответствия работы объекта заданию.

4.2.3 Задания для самостоятельного выполнения

Вариант 1. Автоматизировать работу всех механизмов методической печи по описанному в примере алгоритму. Ввести счетчик количества деталей. При достижении количества обработанных деталей равного 10 сформировать сигнал об окончании программы. Сигнал должен быть импульсным. Время выключенного состояния – 2 с, время включенного состояния – 3 с.

Вариант 2. Разработать систему автоматизации методической печи на основе базового варианта. Дополнительно установить регулируемое время нагрева заготовки. Уставка таймера должна регулироваться с помощью внешнего потенциометра на лицевой панели программируемого реле, подключенного к аналоговому входу реле.

Вариант 3. Автоматизировать работу всех механизмов методической печи по описанному в примере алгоритму. Ввести счетчик количества деталей. Уставка счетчика должна задаваться от тумблеров на лицевой панели программируемого реле, имитирующих пульт оператора. При достижении заданного количества обработанных деталей сформировать длительный сигнал об окончании программы.

Вариант 4. Автоматизировать работу всех механизмов методической печи по описанному в примере алгоритму. По окончании цикла необходимо сформировать сигнал «Конец цикла» и предусмотреть возможность сброса этого сигнала с помощью внешнего сигнала. В таком режиме при появлении на рольганге 1 новой детали работу не начинать, а сформировать еще один сигнал «Ожидание».

Вариант 5. Автоматизировать работу всех механизмов методической печи по описанному в примере алгоритму. Дополнительно установить три режима нагрева с различным временем нагрева заготовки. Необходимо задать три различных временных уставки таймера. Выбор требуемой уставки должен осуществляться с помощью тумблеров на лицевой панели программируемого реле, имитирующих пульт оператора.

Вариант 6. Автоматизировать работу всех механизмов методической печи по описанному в примере алгоритму. С помощью таймеров определить общее время цикла и время выполнения отдельных технологических операций.

4.2.4 Требования к отчёту

Отчет должен содержать:

- название работы;
- цель работы;
- краткое описание автоматизированного объекта;
- обозначение сигналов и команд при управлении объектом, их символьные и непосредственные адреса в программируемом реле ОВЕН ПР114;
- содержательное описание формирования промежуточных переменных и выходных команд ОВЕН ПР114 и составленные на основе содержательного описания соответствующие логические функции;
- логические функции в адресах (символьных) ОВЕН ПР114;
- программу для программируемого реле на языке *FBD*;
- методику экспериментальной проверки правильности работы реализуемой системы и результаты этой проверки;
- выводы по работе.

4.3 Контрольные вопросы

4.3.1 Объясните принцип работы ИФО «Методическая печь».

4.3.2 В каких режимах может работать ИФО «Методическая печь»?

- 4.3.3 Как реализованы исполнительные органы методической печи?
- 4.3.4 Как осуществляется имитация выполняемых действий?
- 4.3.5 Как осуществляется выбор режима работы?
- 4.3.6 Как осуществляется подготовка программы для логического модуля?
- 4.3.7 Что такое «содержательное описание»?
- 4.3.8 Что используется в качестве элементов памяти?
- 4.3.9 Что может привести к неработоспособности системы?
- 4.3.10 Для чего вводятся промежуточные переменные?
- 4.3.11 Как осуществляется управление имитацией методической печи в ручном режиме?
- 4.3.12 Как осуществляется управление имитацией методической печи в автоматическом режиме?
- 4.3.13 С помощью каких устройств возможно управление ИФО?
- 4.3.14 Какие действия выполняются после подготовки реле к работе?
- 4.3.15 Для чего предназначен датчик Д5?
- 4.3.16 Почему и для чего в программе используются *SR*-триггеры, а не *RS*-триггеры?

5 Изучение построения распределенных систем управления с использованием GSM-модема ОБЕН ПМ-01

5.1 Распределенные системы управления на основе GSM-модема

Современные распределенные системы автоматизации и управления всё более широко используют *GSM*-сети для передачи данных [13]. *GSM*-сети характеризуются развитой во всех регионах инфраструктурой, высокой надежностью, возможностью быстрого развертывания и невысокими стоимостными характеристиками. Модем ОБЕН ПМ-01 совместно с *SIM*-картой предназначен для удаленного обмена данными через беспроводные системы связи стандарта *GSM* с оборудованием, оснащенным последовательными интерфейсами связи, рисунок 5.1.



Рисунок 5.1 – Внешний вид модема ОБЕН ПМ-01

Краткие технические характеристики модема ОБЕН ПМ-01 представлены в таблице 5.1.

Таблица 5.1 – Технические характеристики модема ОБЕН ПМ-01

Характеристика	Значение
1	2
Напряжение питания, В постоянного напряжения	от 10 до 30
Максимальная потребляемая мощность, не более, ВА	15
Рабочий частотный диапазон	EGSM900, DCS1800, PCS1900

Продолжение таблицы 5.1

1		2
Класс выходной мощности передатчика		4(EGSM900/GSM85). 1(DCS1800/PCS1900)
Класс мобильного оборудования		B
GPRS мультислот		класс 10
Скорость обмена в режиме CSD, бнт/с		9600
Скорость обмена в режиме GPRS, бит/с	Прием	до 85600
	Передача	до 42800
Поддерживаемые типы		SMS, SMS-MO, SMS-MT, SMS-CB
Интерфейс связи с ноутбуком		RS232, RS485
Диапазон скоростей обмена по последовательному интерфейсу связи, бит/с		от 1200 до 115200
Тип антенного соединителя		гнездо SMA

На рисунке 5.2 представлена структурная схема модема, поясняющая его принцип работы.

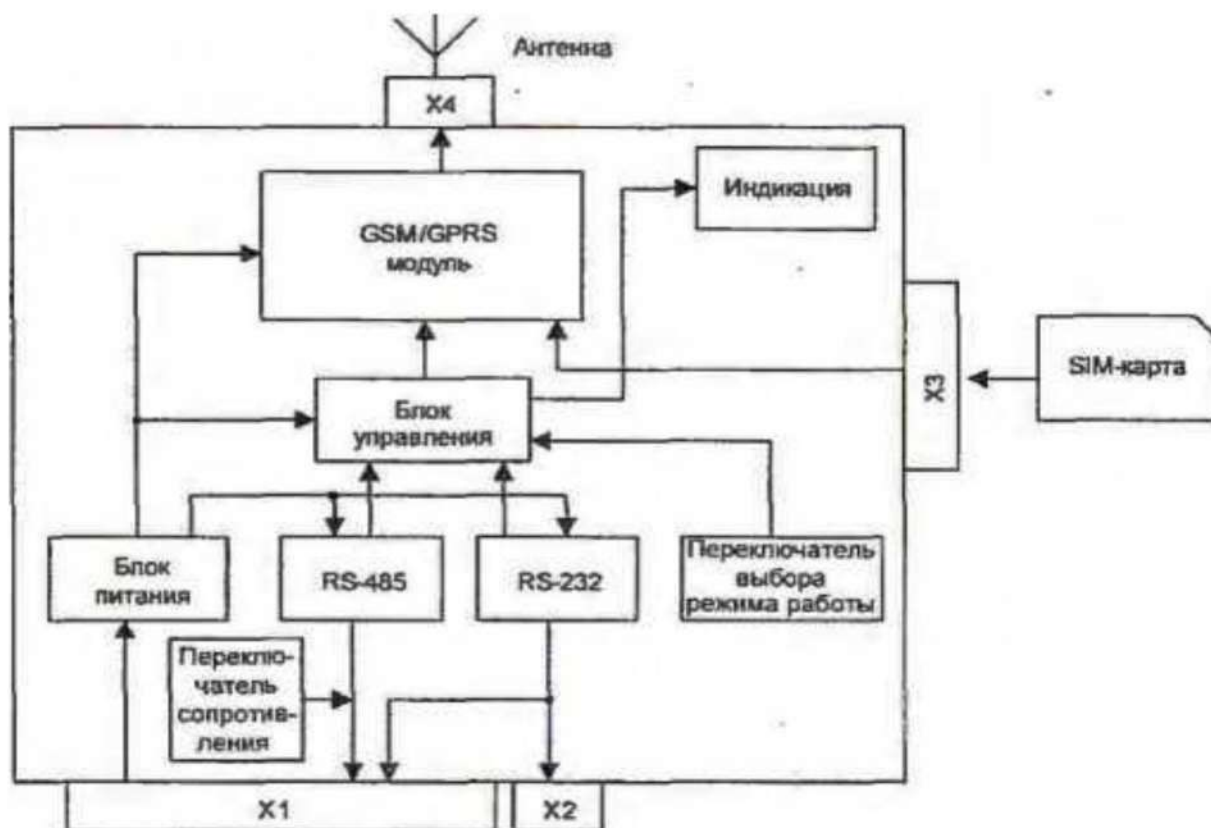


Рисунок 5.2 – Структурная схема модема

Блок управления выполняет функции запуска GSM/GPRS модуля, коммутации сигналов последовательных интерфейсов в GSM/GPRS модуль, управления индикацией.

Соединители *X1*, *X2* служат для подключения линий последовательных интерфейсов RS232 и RS485.

Модем может функционировать в двух режимах: активный (прием/передача данных) и режим настройки таймера автоматической перезагрузки.

На передней панели модема находится сдвоенный двухпозиционный переключатель (рисунок 5.1). С помощью первого переключателя подключается внутреннее сопротивление согласования для линии интерфейса RS485 с номинальным значением сопротивления 120 Ом. Выбор режима работы модема осуществляется с помощью второго переключателя.

Индикация модема состоит из трех индикаторов красного свечения. Назначение индикации модема приведено в таблице 5.2.

Таблица 5.2 – Назначение индикации модема

Индикатор	Функция
TX	Сигнализирует о прохождении данных по интерфейсу RS232/RS485 в направлении от модема к управляющему прибору
RX	Сигнализирует о прохождении данных по интерфейсу RS232/RS485 в направлении от управляющего прибора к модему
GSM	Выключен – нет напряжения питания, модем выключен Мигает с периодом 3 с – модем зарегистрирован в сети GSM Мигает с периодом 0,8 с – модем производит поиск сети GSM Мигает с периодом 0,3 с – модем производит обмен данными по GPRS

5.1.1 Основы конфигурирования и работы модема

Модем может выполнять следующие функции:

- прием и передача SMS;
- прием и передача данных с помощью CSD;
- прием и передача данных с помощью GPRS;
- работа с последовательными интерфейсами RS232 и RS485;
- управление приемом и передачей данных по последовательным интерфейсам RS232 и RS485 с помощью AT-команд в соответствии со стандартами GSM 07.05 и GSM 07.07;

– индикация наличия обмена данными по последовательным портам *RS485* или *RS232*;

– индикация наличия регистрации в сети *GSM* и наличия передачи данных в режиме *GPRS*;

– функция автоматической периодической перезагрузки модема в соответствии с заданными настройками.

Управление работой модема осуществляется с помощью *AT*-команд по последовательным интерфейсам. Модем ПМ01 позволяет выдавать данные в оба интерфейса одновременно. Внешние устройства должны разделять во времени данные или команды, посылаемые ими в модем по интерфейсам *RS232* и *RS485*. Список наиболее широко используемых команд приведен в таблице 5.3. В системах связи стандарта *GSM* передача данных может осуществляться различными способами:

- передача данных по *CSD*;
- передача данных по *GPRS*;
- передача данных с помощью *SMS*-сообщений.

Для передачи данных требуется предварительно установить соединение с удаленным абонентом (кроме передачи данных с помощью *SMS*-сообщений). При передаче данных с помощью *GPRS* модем обеспечивает поддержку *TCP/IP* и *UDP* протоколов. При передаче данных с помощью *SMS*-сообщений модем обеспечивает поддержку текстового и *PDU*-режимов *SMS*-сообщений.

Таблица 5.3 – Наиболее часто используемые команды модема

Наименование команды ¹	Ответ на команду ²	Описание команды
1	2	3
<i>ATE0</i>	<i>ATE0</i> OK	Отключить эхо
<i>ATE1</i>	OK	Включить эхо
<i>AT+IPR</i> =<скорость>	OK	Настроить скорость обмена по последовательному порту
<i>AT&W</i>	OK	Записать ранее произведенные настройки в память
<i>AT+CPIN?</i>		Запросить необходимость ввода <i>PIN</i> -кода
	+ <i>CPIN: SIM PIN OK</i>	Требуется ввод <i>PIN</i> -кода

Продолжение таблицы 5.3

1	2	3
<i>AT+CPIN?</i>	<i>+CPIN: READY OK</i>	Не требуется ввода <i>PIN</i> -кода
<i>AT+CPIN</i> =< <i>PIN</i> -код>	OK	Ввод <i>PIN</i> -кода
<i>AT+CREG?</i>	<i>+CREG:<n>,<m></i>	Запросить информацию о регистрации в сети <i>GSM</i> . <i>n</i> может принимать значения: 0 – отчет о регистрации в сети выключен; 1 – отчет о регистрации в сети включен. <i>m</i> может принимать значения: 0 – не зарегистрирован; 1 – зарегистрирован; 2 – идет поиск сети; 3 – в регистрации отказано; 5 – зарегистрирован в роуминге.
<i>ATD</i> <номер>	<i>CONNECT</i> <скорость>	Установка <i>CSD</i> -соединения
<i>AT+CMGS</i> =< "номер"> <текст> <i>Ctrl+Z</i> ³	<текст> <i>OK</i>	Отправка <i>SMS</i> -сообщения
¹ Наименование команды должно формироваться в соответствии с таблицей кодировки символов ASCII. После наименования команды должен следовать символ перевода строки– <i>0Dh</i>. ² Ответ модема начинается и заканчивается последовательностью символов перевода строки и возврата каретки– <i>0Dh 0Ah</i>. ³ Команду <i>Ctrl+Z</i> (<i>L4</i>) необходимо посылать после окончания ввода текста.		

Для внешнего принудительного перезапуска внутреннего *GSM/GPRS* модуля модема используется сигнал *RESET* который реализован в виде кнопки сброса на лицевой панели модуля. Для включения/выключения модуля необходимо замкнуть эти контакты между собой на время от 1 до 5 с.

5.1.2 Передача данных с помощью SMS-сообщений в текстовом режиме

Для передачи данных с помощью SMS-сообщений в текстовом режиме необходимо отправить в модем следующие команды:

AT+CMGF=1 – установка текстового режима для SMS-сообщений;

AT+CMGS= <"номер">

<текст>

ctrl+Z

Между вводом команды *AT+CMGS=*<"номер"> и вводом текста необходимо дождаться ответа модема в виде символа ">".

Прием SMS-сообщения индицируется отчетом:

CMT1:<память>,<индекс>,

где <память> – тип памяти, в которой сохранено сообщение,

<индекс> – порядковый номер сообщения в памяти.

Прочитать принятое сообщение можно командой: *AT+CMGR=*<индекс>, 0.

5.1.3 Обмен данными по TCP/IP протоколу

Для обмена данными по TCP/IP протоколу (по GPRS) необходимо отправить в модем следующий набор команд:

– *AT + CGATT = 1* – подключение модема к сервису GPRS;

– *AT + CSTT =* «точка доступа», «имя», «пароль» – выбор GPRS сервиса для передачи данных, задание точки доступа в сеть GPRS, имени пользователя и пароля;

– *AT + CIICR* – установка соединения с сервисом GPRS;

– *AT + CIFSР* – получение локального IP-адреса;

– *AT + CIPSTART =* «TCP», «xx.xx.xx.xx», «порт» – установка соединения по TCP/IP протоколу с сервером по заданному номеру порту, где xx.xx.xx.xx – IP-адрес сервера;

– *AT + CIPSEND* <данные> *ctrl + Z* – передача данных по установленному соединению.

После ввода команды *AT + CIPSTART*, в случае успешной установки соедине-

ния, модем выдаст отчет: *CONNECT OK*. Между вводом команды *AT + CIPSEND* и вводом данных следует дождаться ответа модема в виде символа «>».

5.1.4 Программирование таймера автоматической перезагрузки

Для установки периода таймера автоматической перезагрузки модема выполнить следующие действия.

Шаг 1. Подключить модем к ноутбуку с помощью интерфейса *RS232*.

Шаг 2. Установить переключатель выбора режима работы модема в положение «*Off*».

Шаг 3. Подать питание на модем.

Шаг 4. Запустить на ПК программу *HyperTerminal* или ее аналог, в программе установить номер *COM*-порта, к которому подключен модем, и следующие параметры обмена:

- скорость 9600 бит/с;
- 8 бит данных;
- 1 стоп бит;
- нет контроля четности;
- нет управления потоком.

Шаг 5. Для задания периода таймера подать команду *SETTM X*, где *X* – число от 0 до 170 (при задании значения 0 таймер будет отключен, значения *X* от 1 до 170 – период перезагрузки в часах).

Шаг 6. Если команда принята, модем выдаст ответ *OK*, если команда введена неверно, модем выдаст ответ *ERROR*.

Шаг 7. Для проверки текущего значения периода таймера необходимо подать команду *GETTM*, модем выдаст ответ *GETTM X*, где *X* – текущее значение периода таймера перезагрузки.

Шаг 8. Отключить питание модема.

5.2 Лабораторная работа 5 «Разработка управляющей программы для передачи данных через GSM-модем»

Цель работы: ознакомиться с устройством и техническими характеристиками GSM-модема ОВЕН ПМ-01 (далее модема), изучить его основные функции и приобрести навыки использования модема в составе распределенных систем управления.

Задание на выполнение работы:

- изучить возможности и особенности лабораторного стенда;
- изучить назначение, технические характеристики GSM-модема ОВЕН ПМ-01;
- изучить принципы работы модема в составе систем автоматизации;
- выполнить синтез системы автоматизации согласно выданного варианта задачи;
- составить программу для ввода в программируемый контроллер ПЛК150 с учетом использования модема;
- пройти тестирование по функциональным возможностям и режимам работы GSM-модема ОВЕН ПМ-01;
- при необходимости сконфигурировать модем;
- набрать на ноутбуке подготовленную программу, откомпилировать ее и ввести в контроллер;
- провести экспериментальные исследования вариантов режимов работы модема;
- убедиться в правильности работы программы.

5.2.1 Общие сведения об изучаемом оборудовании

В зависимости от предложенного студенту варианта в работе могут быть использованы программируемый контроллер ОВЕН ПЛК150 и панель оператора ОВЕН СП270.

Изучаемое оборудование расположено в модуле сопряжения сигналов, внешний вид которого представлен рисунке 5.3.

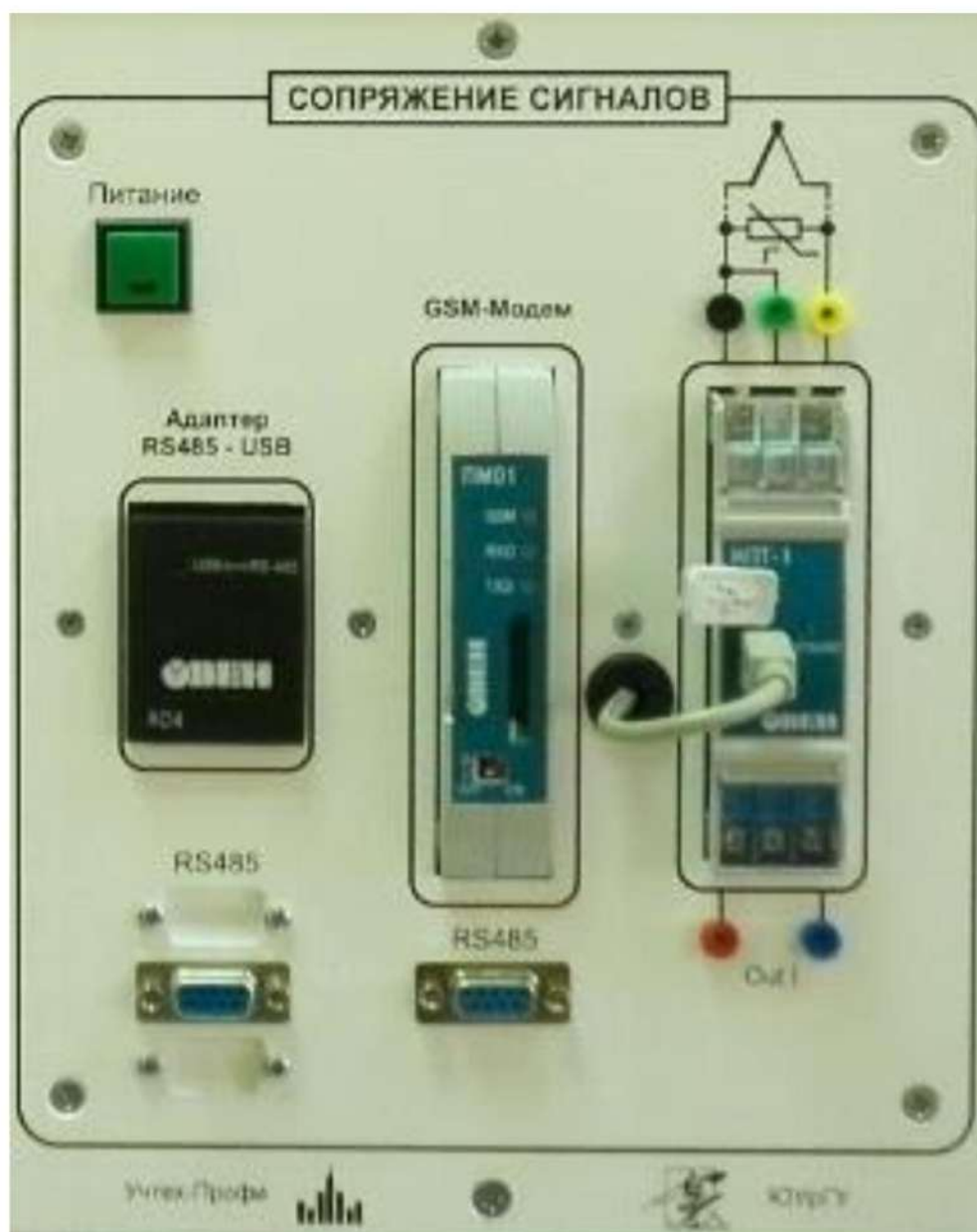


Рисунок 5.3 – Модуль сопряжения сигналов

Как было уже описано ранее, обмен данными с использованием *GSM*-модема может осуществляться двумя способами: посредством *SMS*-сообщений и с помощью подключения к сети *Internet* по *GPRS*.

В данной лабораторной работе предлагается освоить первый способ передачи данных, так как он наиболее простой и менее затратный. При необходимости освоения второго способа обмена информацией (через *GPRS*) следует обратиться к документации.

Лабораторная работа по настройке передачи данных посредством *SMS*-

сообщений выполняется в следующем порядке.

Шаг 1. Соединить между собой разъёмы *RS485* программируемого контроллера ПЛК150 и *GSM*-модема. Проверить наличие *SIM*-карты в картоприёмнике. Подать питание на используемые в лабораторной работе модули стенда.

Шаг 2. С помощью среды программирования *CoDeSys 2.3* разработать управляющую программу для передачи данных через *GSM*-модем:

- запустить *CoDeSys 2.3* и при настройке проекта выбрать язык программирования *CFC* (рисунок 5.4);

- в окне конфигурирования добавить модуль «*Universal network module*», в котором далее заменить порт *RS232* на *RS485*. В параметрах модуля установить скорость обмена данными (рисунок 5.5, а), соответствующую скорости, выбранной в *GSM*-модеме (по умолчанию – 9600 бод);

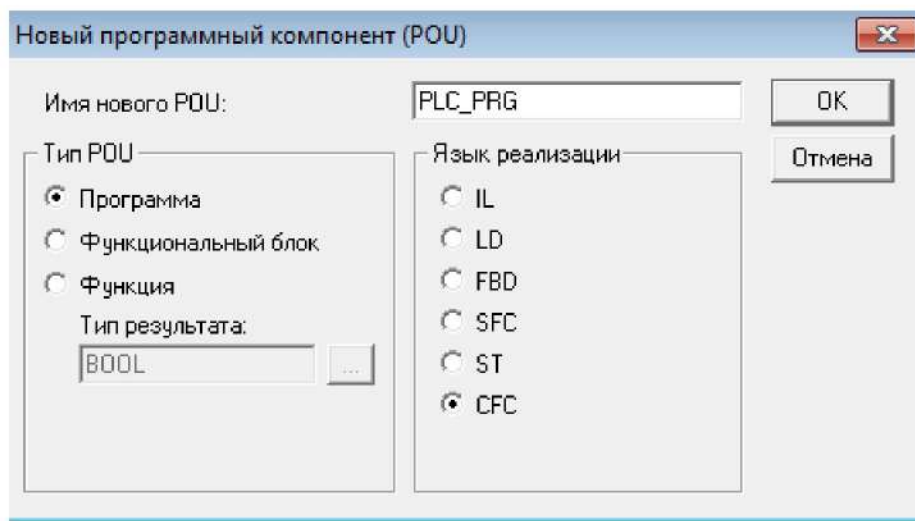


Рисунок 5.4 – Выбор языка программирования (*CFC*)

- в закладке «Параметры модуля» изменить минимальную длительность цикла с 1 до 50 мс (рисунок 5.5, б);

- объявить переменные программы в соответствующей области (рисунок 5.6):

- а) функциональные блоки *OpenPort*, *FB_SMS_SR* и *FB_SMS_CFG*;

- б) текст отправляемого и текст последнего принятого *SMS*-сообщения (*smsText* и *InText*);

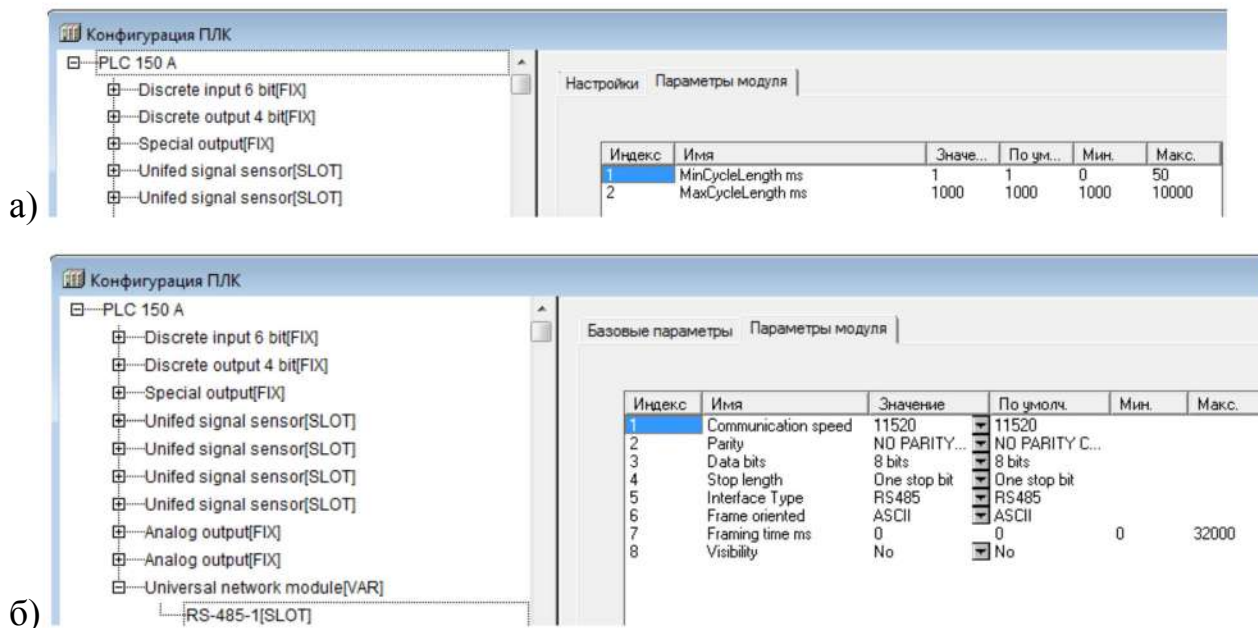


Рисунок 5.5 – Конфигурирование порта RS485 программируемого контроллера

```

PLC_PRG (PRG-CFC)
0001 PROGRAM PLC_PRG
0002 VAR
0003   x: WORD;
0004
0005   com: OpenPort;
0006   sms: FB_SMS_SR;
0007   cfg: FB_SMS_CFG;
0008   devNum: WORD := 0;
0009   smsText: STRING := "privet mir";
0010   smsNum: STRING := 79033929465;
0011
0012   inText: STRING;
0013   inNum: STRING;
0014   smsSend AT %IX0.0: BOOL;
0015   comSend AT %QX1.0: BOOL;
0016
0017 END_VAR

```

Рисунок 5.6 – Объявление переменных

в) номер абонента, кому отправить SMS-сообщение и номер отправителя последнего входящего SMS-сообщения (*smsNum* и *InNum*);

г) биты управления и контроля отправки SMS-сообщений (*smsSend* и *comSend*);

– в закладке проекта «Ресурсы» найти менеджер библиотек и запустить его двойным щелчком левой клавиши мыши;

- в открывшемся окне выбрать область, в которой перечислены все подключенные библиотеки, и выделить строку ниже последней библиотеки;
- нажать правую клавишу мыши и в открывшемся окне выбрать пункт «Добавить библиотеку» (рисунок 5.7, а);

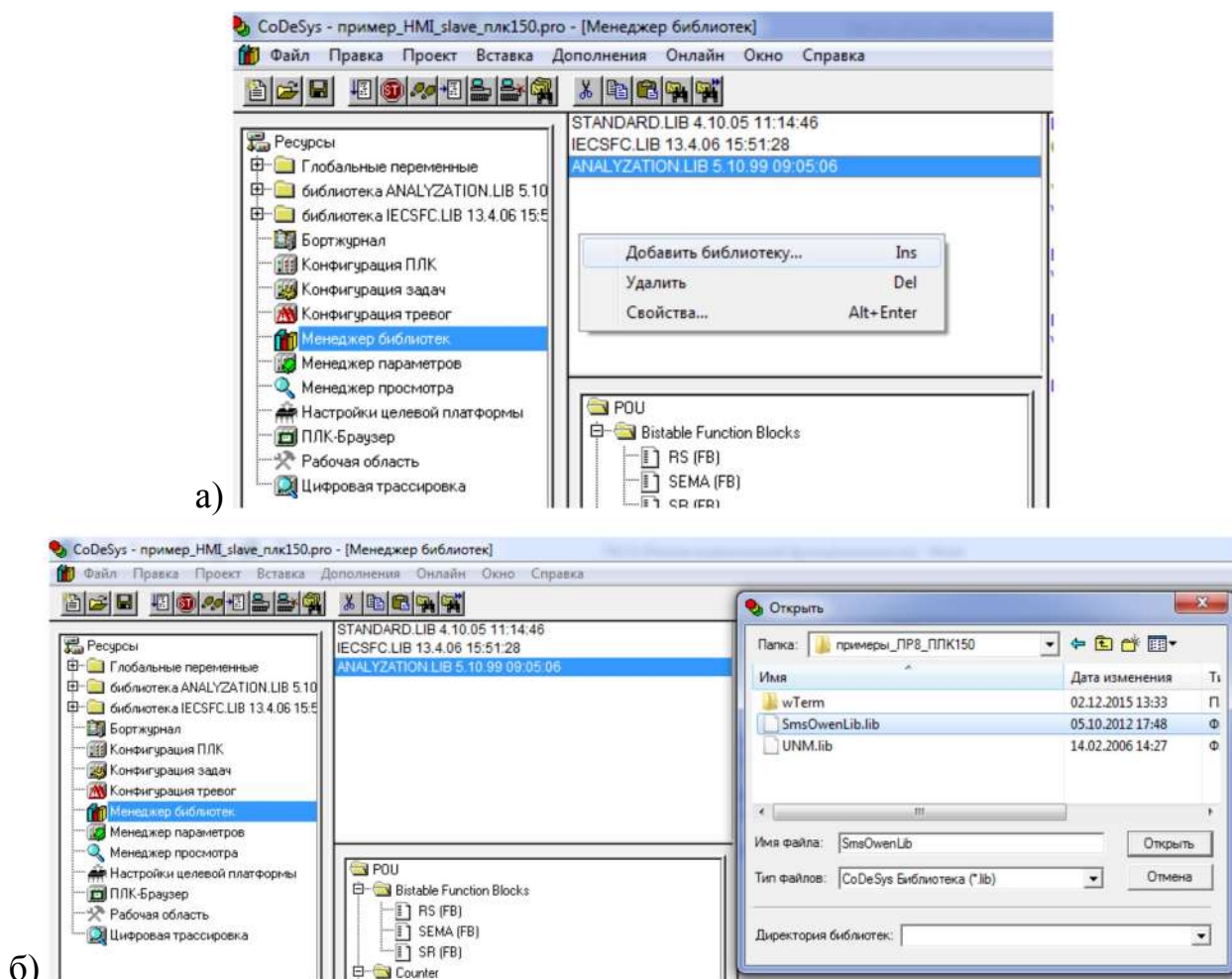


Рисунок 5.7 – Добавление библиотек

- через окно проводника выбрать требуемые для подключения библиотеки: *SmsOwenLib.lib* и *UNM.lib* (рисунок 5.7, б) и нажать «Открыть»;
- в рабочей области с использованием функциональных блоков из добавленных библиотек (*OpenPort*, *FB_SMS_SR* и *FB_SMS_CFG*) разработать программу, представленную на рисунке 5.8;
- записать управляющую программу в память программируемого контроллера

ПЛК150, после чего перевести его в режим «Работа» нажатием кнопки «СТАРТ/СТОП».

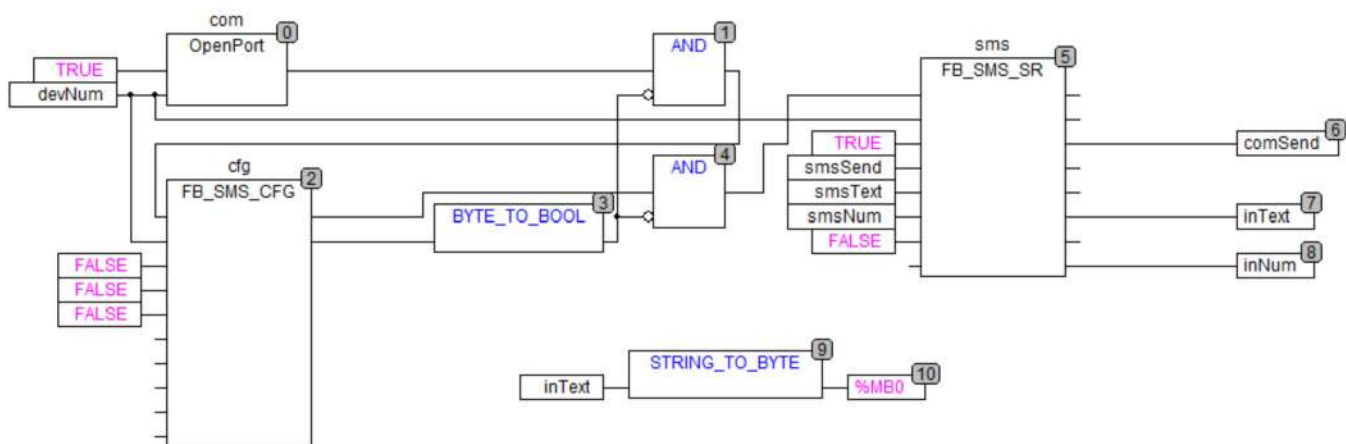


Рисунок 5.8 – Управляющая программа для получения/отправки SMS-сообщений

Шаг 3. Проверка правильности функционирования разработанной системы:

- подать питание на модули лабораторного стенда, используемые при проведении лабораторной работы;
- включением тумблера *I1* на лицевой панели программируемого контроллера и панели оператора инициировать отправку SMS-сообщения на телефон, указанный в строке «*smsNum*». Для удобства можно использовать свой собственный номер;
- отправить со своего телефона на номер SIM-карты, установленной в модем, SMS-сообщение, например, с каким-нибудь числом. В результате модем должен принять SMS-сообщение и должны включиться выходы контроллера, соответствующие числу в двоичном коде;
- при необходимости внести изменения в настройки и управляющую программу контроллера.

5.2.2 Задание для самостоятельного выполнения

Для управляющей программы, разработанной в первой лабораторной работе, реализовать функцию запуска и остановки с помощью SMS-сообщений.

5.2.3 Требования к отчёту

Отчет должен содержать:

- название работы;
- цель работы;
- порядок выполнения лабораторной работы;
- физические и символические адреса переменных, отредактированная и проверенная программа для контроллера ПЛК150 с комментариями;
- описание методики и экспериментальной проверки работоспособности системы автоматизации;
- выводы по работе.

5.3 Контрольные вопросы

5.3.1 Чем характеризуются *GSM*-сети?

5.3.2 Для чего предназначен модем ОВЕН ПМ-01?

5.3.3 Из каких устройств состоит модем ОВЕН ПМ-01?

5.3.4 Что означает термин «конфигурирование контроллера»?

5.3.5 В какой последовательности осуществляется конфигурирование контроллера ПЛК150?

5.3.6 Какие библиотеки *CoDeSys* 2.3 отвечают за связь программируемого контроллера с *GSM*-модемом?

5.3.7 Как осуществляется запись программ в контроллер ПЛК150?

5.3.8 Какие функции выполняет в системе *GSM*-модем?

5.3.9 Какие способы передачи данных можно реализовать с помощью *GSM*-модема?

5.3.10 В каких режимах может работать *GSM*-модем?

5.3.11 Как проверить правильность функционирования разработанной системы?

Список использованных источников

- 1 Беккер, В.Ф. Технические средства автоматизации. Интерфейсные устройства и микропроцессорные средства: учеб. пособие / В.Ф. Беккер. – М.: РИОР: ИНФРА-М, 2015. – 2-е изд. – 152 с. . – ISBN 978-5-369-01198-0 (РИОР).
- 2 Технические средства автоматизации. Программно-технические комплексы и контроллеры : учеб. пособие / И. А. Елизаров, Ю. Ф. Мартемьянов, А. Г. Схиртладзе, С. В. Фролов. – М.: «Издательство Машиностроение-1», 2004. – 180 с. – ISBN 5-94275-104-8.
- 3 Борисов, А. М. Средства автоматизации и управления : учеб. пособие / А. М. Борисов, А. С. Нестеров. – Челябинск : Изд-во ЮУрГУ, 2007. – 207 с.
- 4 ОВЕН. Оборудование для автоматизации: сайт компании ОВЕН. – Электрон. дан. – Москва: ОВЕН, [2009] . – Режим доступа : <http://www.owen.ru/>
- 5 ПЛК150. Контроллер программируемый логический : руководство по эксплуатации. – Электрон. дан. – Москва: ОВЕН. – 47 с. – Режим доступа : http://www.owen.ru/uploads/re_plc150_2274.pdf
- 6 СП270. Панель оператора : руководство по эксплуатации КУВФ.421449.003 РЭ. – Электрон. дан. – Москва: ОВЕН. – 25 с. – Режим доступа : http://www.owen.ru/uploads/re_sp270_1467.pdf
- 7 Программируемое реле ПР114 : руководство по эксплуатации. – Электрон. дан. – Москва: ОВЕН. – 64 с. – Режим доступа : http://www.owen.ru/uploads/re_pr114_1956.pdf
- 8 Нестеров А. С. Лабораторный стенд «Система автоматического управления ОВЕН»: метод. указ. к проведению лабораторных работ / А. С. Нестеров. – Челябинск: «Учтех-Профи», 2015. – 88 с.
- 9 CoDeSys v.3 – Первые шаги. – Электрон. дан. – Москва: ОВЕН, [2009] . – Режим доступа : http://www.owen.ru/catalog/codesys_v3/19944443 .
- 10 Руководство пользователя по программированию ПЛК в CoDeSys 2.3. – Электрон. дан. – Смоленск: ПК Пролог, 2006. – 453 с. – Режим доступа : http://www.kipshop.ru/CoDeSys/steps/codesys_v23_ru.pdf .

СОДЕРЖАНИЕ Часть 2

ПРАКТИЧЕСКАЯ РАБОТА № 1

Создание узла АРМ.....	4
------------------------	---

ПРАКТИЧЕСКАЯ РАБОТА № 2 Создание стрелочного прибора, привязка к аргументу	13
--	----

ПРАКТИЧЕСКАЯ РАБОТА № 3 Добавление функции управления.....	22
--	----

ПРАКТИЧЕСКАЯ РАБОТА № 4 Простейшая обработка данных.....	33
--	----

БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	41
-------------------------------	----

ПРАКТИЧЕСКАЯ РАБОТА № 1

Создание узла АРМ

Цель работы – создать систему мониторинга, содержащую один узел – автоматизированное рабочее место (АРМ), с использованием механизма автопостроения каналов TRACE MODE методом «от шаблона экрана». В дальнейшем в состав системы будут введены функции управления, реализовано взаимодействие с приложением Windows по протоколу обмена DDE, а в завершение будет показано подключение аналогового сигнала от модуля удаленного ввода.

Порядок выполнения работы

Загрузим инструментальную систему двойным щелчком левой клавиши (ЛК) мыши по иконке рабочего стола Windows и с помощью иконки инструментальной панели создадим **Новый проект**.

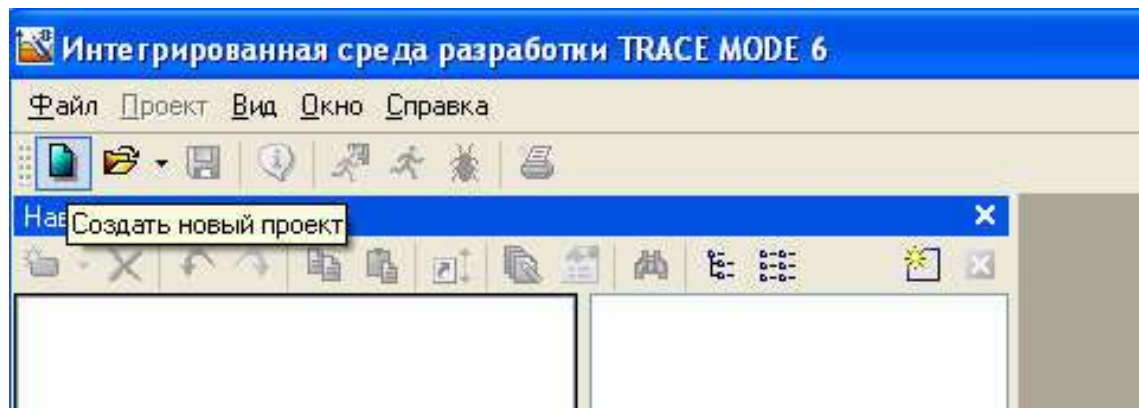


Рис. 1. Создание **Нового проекта**

После нажатия ЛК мыши на экранной кнопке «Создать», в левом окне «Навигатора проекта» отобразится «дерево проекта», содержащее слои «Ресурсы», «Система» (с созданным узлом АРМ «RTM_1»), «Источники/Приемники» и «Библиотеки_компонентов».

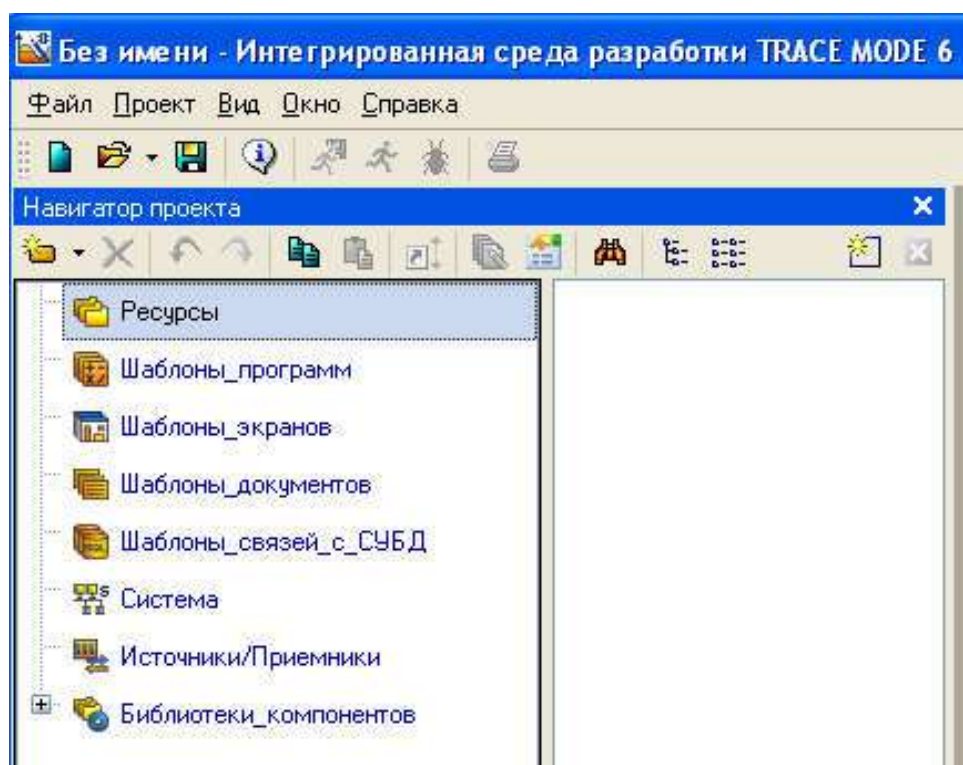


Рис. 2. Навигатор проекта

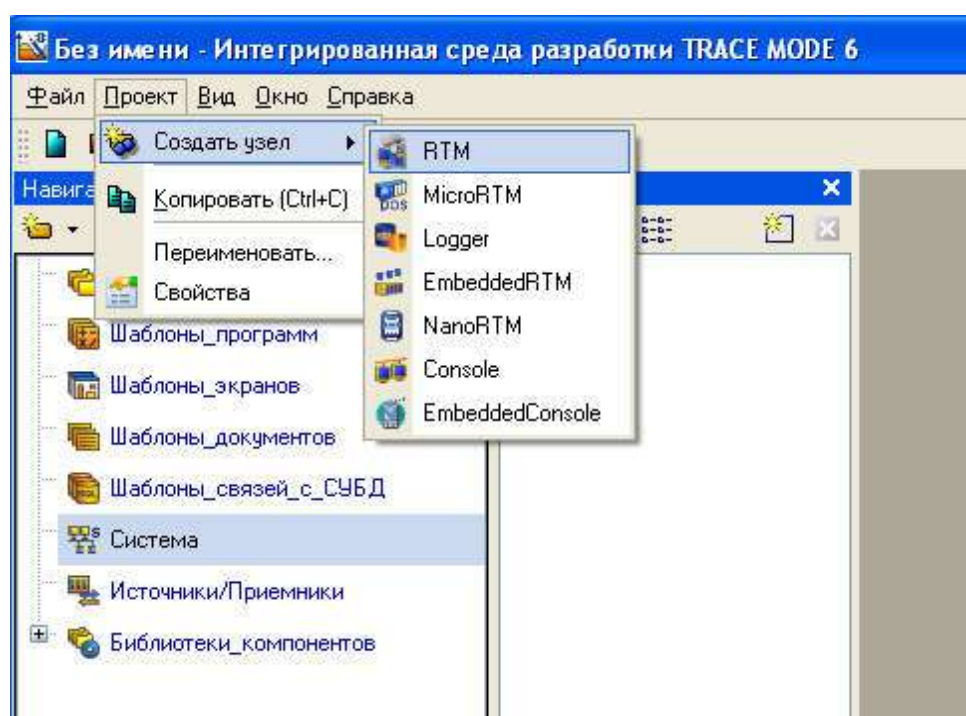


Рис. 3. Выбор узла «RTM»

В правом окне Навигатора проекта отобразится содержимое узла «RTM_1» – пустая группа «Каналы» и один канал класса «CALL Экран#1», вызывающий соответствующий компонент – шаблон экрана,

предназначенный для отображения с помощью графических элементов (ГЭ) средств человеко-машинного интерфейса на узле «RTM_1»:

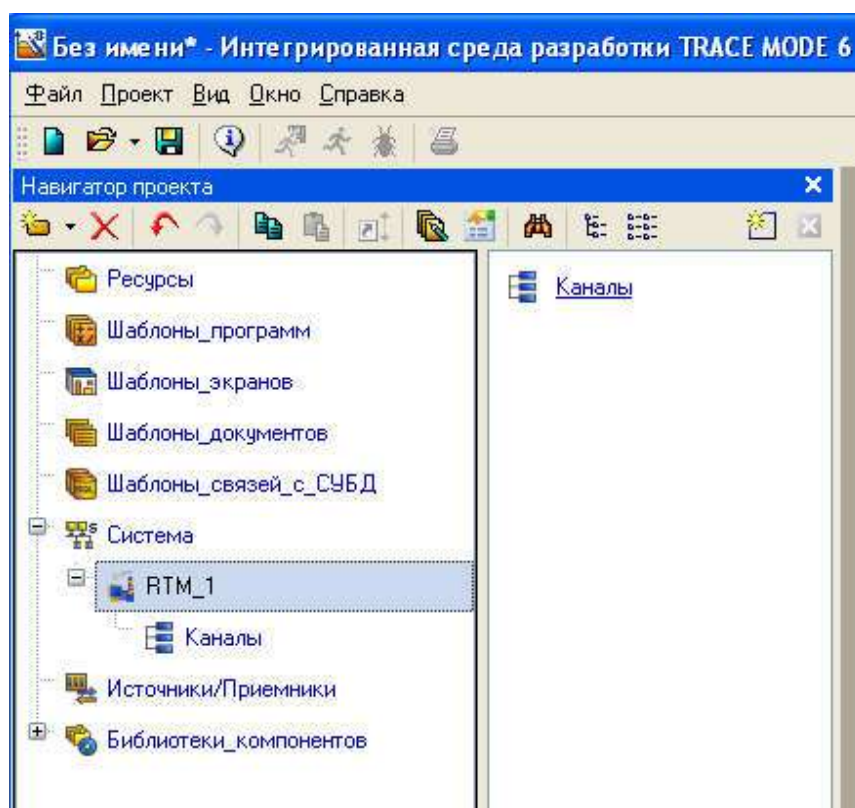


Рис. 4. Пустая группа «Каналы»

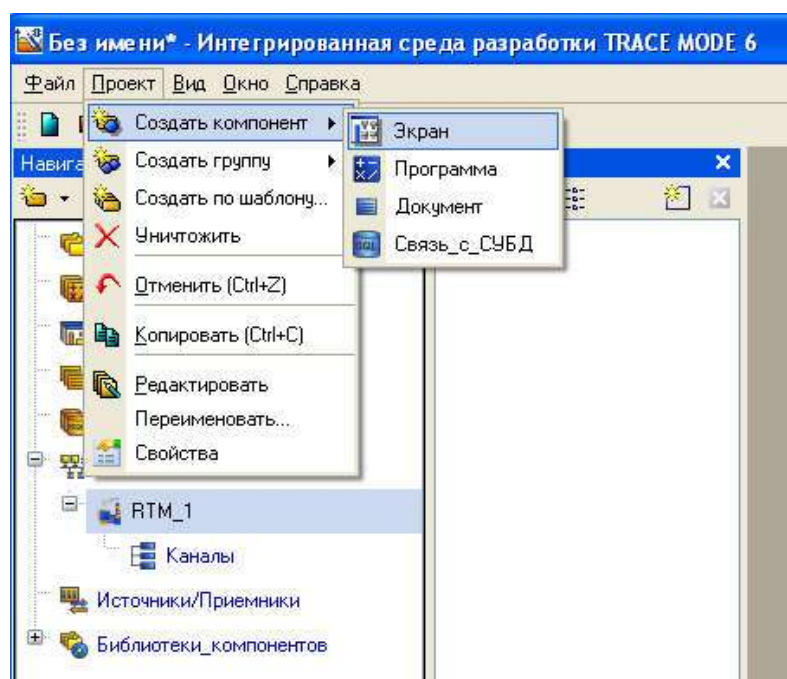


Рис. 5. Создание компонента «Экран»

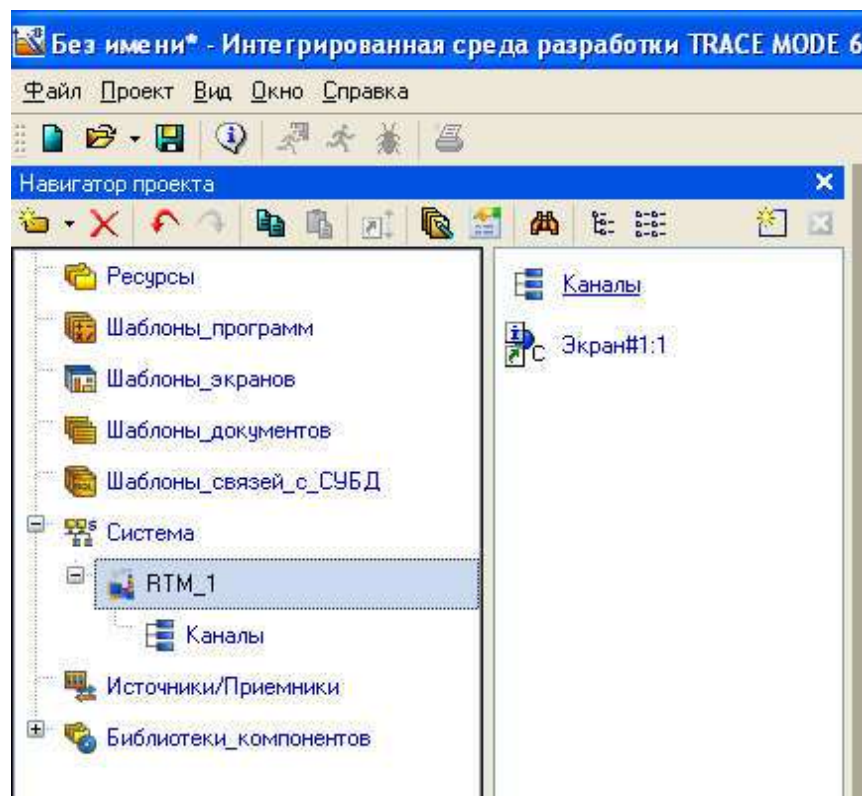


Рис. 6. Содержимое узла «RTM_1»

Создание графического экрана

Двойным щелчком ЛК на компоненте «Экран#1» откроем окно графического редактора.

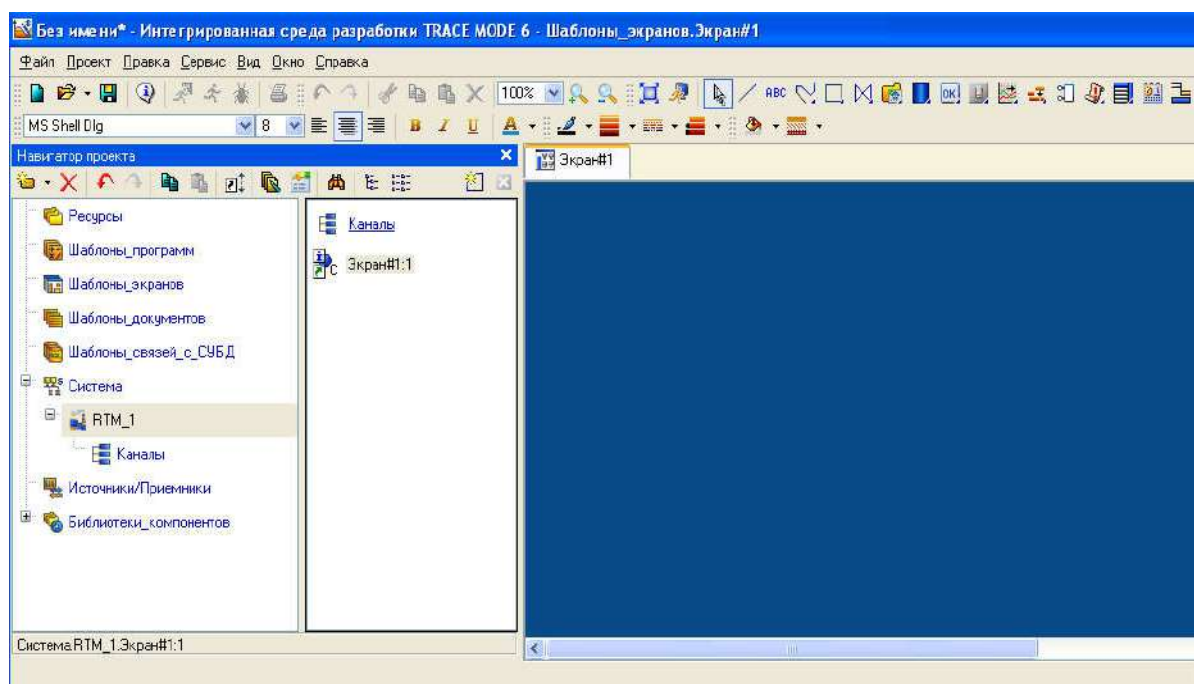


Рис. 7. Окно графического редактора «Экран#1»

Создание статического текста

Разместим в левом верхнем углу экрана статический текст – надпись **«Значение параметра»**.

Для этого выполним следующие действия: на панели инструментов графического редактора ЛК мыши выделим иконку ГЭ **«Текст»**; в поле графического редактора установим прямоугольник ГЭ, для этого:

- зафиксируем ЛК точку привязки – левый верхний угол;
- развернем прямоугольник движением курсора до необходимого размера;
- зафиксируем ЛК выбранный ГЭ:

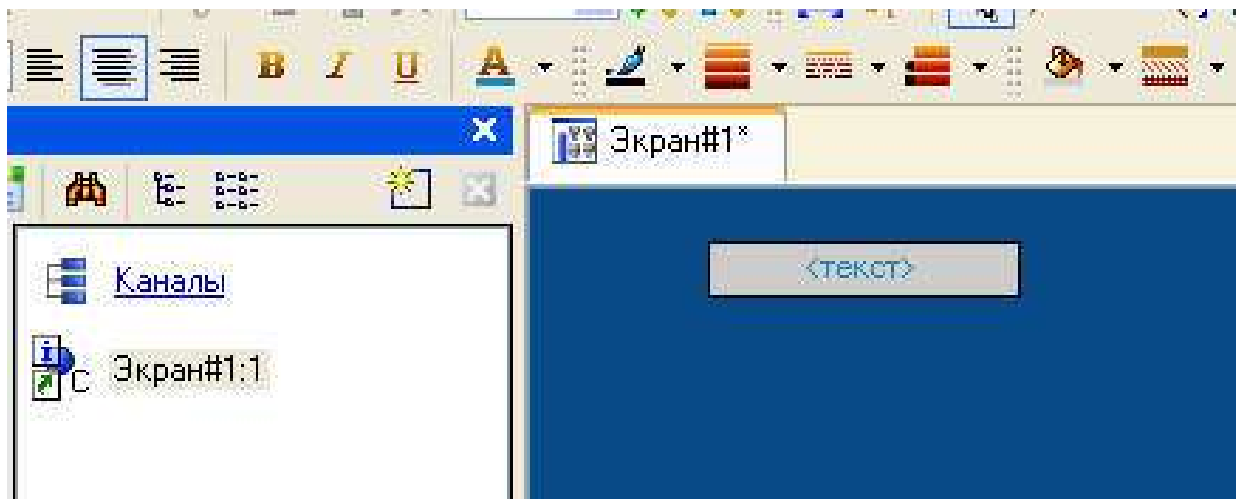


Рис. 8. Графический элемент **«Текст»**

- для перехода в режим редактирования атрибутов размещенного ГЭ выделим ЛК иконку на панели инструментов;
- двойным щелчком ЛК по ГЭ откроем окно его свойств;
- в правом поле строки **«Текст»** наберем **«Значение параметра»** и нажмем на клавиатуре клавишу **«Enter»**;

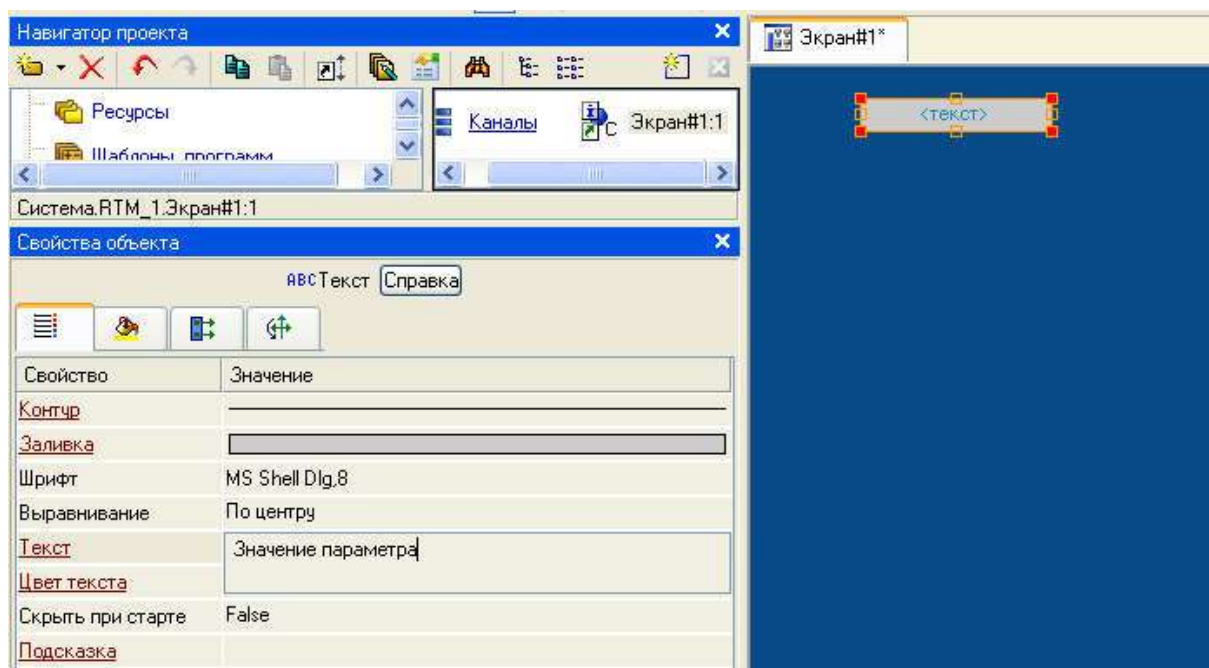


Рис. 9. Ввод текста

• закроем окно свойств щелчком ЛК по иконке, ГЭ примет следующий вид:

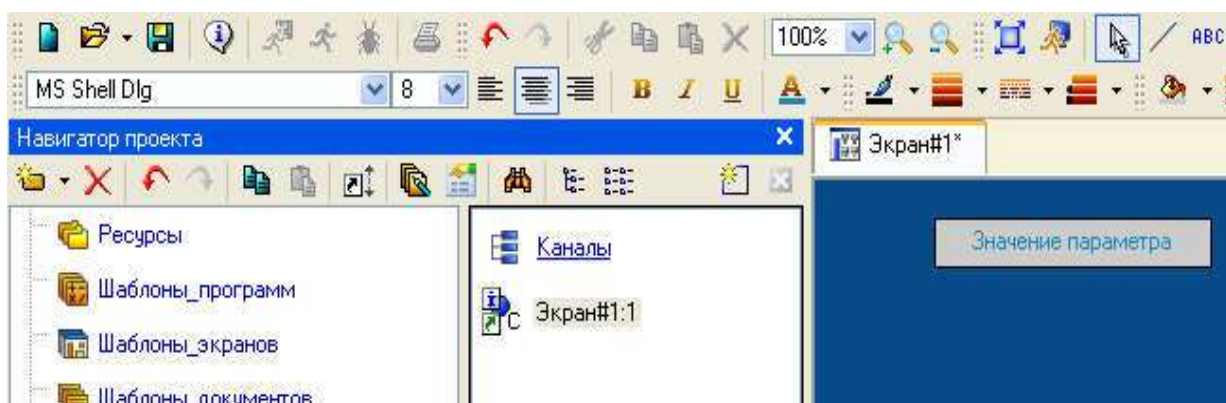


Рис. 10. Результат создания графического элемента – надписи
«Значение параметра»

Создание динамического текста, создание аргумента экрана в процессе настройки динамического текста

Подготовим на экране вывод динамического текста для отображения численного значения какого-либо источника сигнала (внешнего или внутреннего) путем указания динамизации атрибута «Текст» ГЭ. Определим назначение аргумента шаблона экрана. Для этого необходимо выполнить следующие действия:

- создадим и разместим новый ГЭ справа от ГЭ с надписью «Значение параметра»;
- откроем свойства вновь размещенного ГЭ;
- двойным щелчком ЛК на строке «Текст» вызовем меню «Вид индикации»;

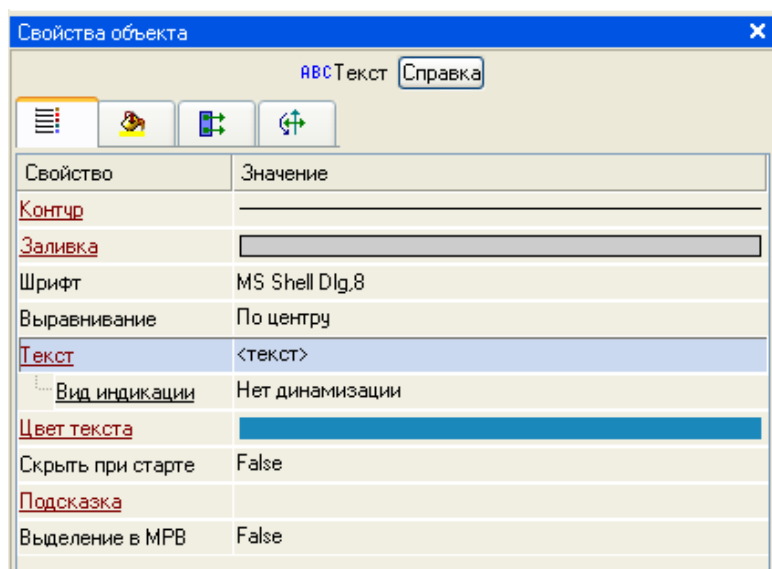


Рис. 11. «Свойства объекта» – «Текст» – «Вид индикации»

- в правом поле строки щелчком ЛК вызовем список доступных типов динамизации атрибута;
- из всех предлагаемых типов выберем ЛК «Значение»;



Рис. 12. Выбор параметра «Значение»

- в открывшемся меню настройки параметров динамизации «Вид индикации» выполним щелчок ЛК в правом поле строки «Привязка»;

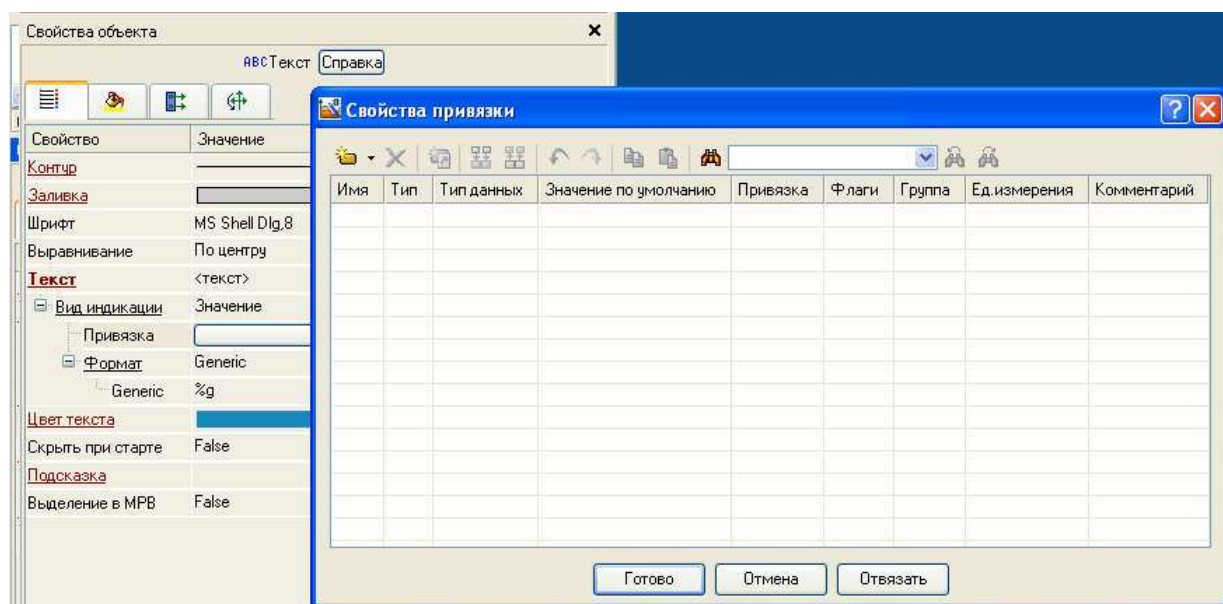


Рис. 13. Выбор параметра «Привязка»

- в открывшемся окне «Свойства привязки», нажмем ЛК по иконке на панели инструментов и тем самым создадим аргумент шаблона экрана «ARG_000» ;

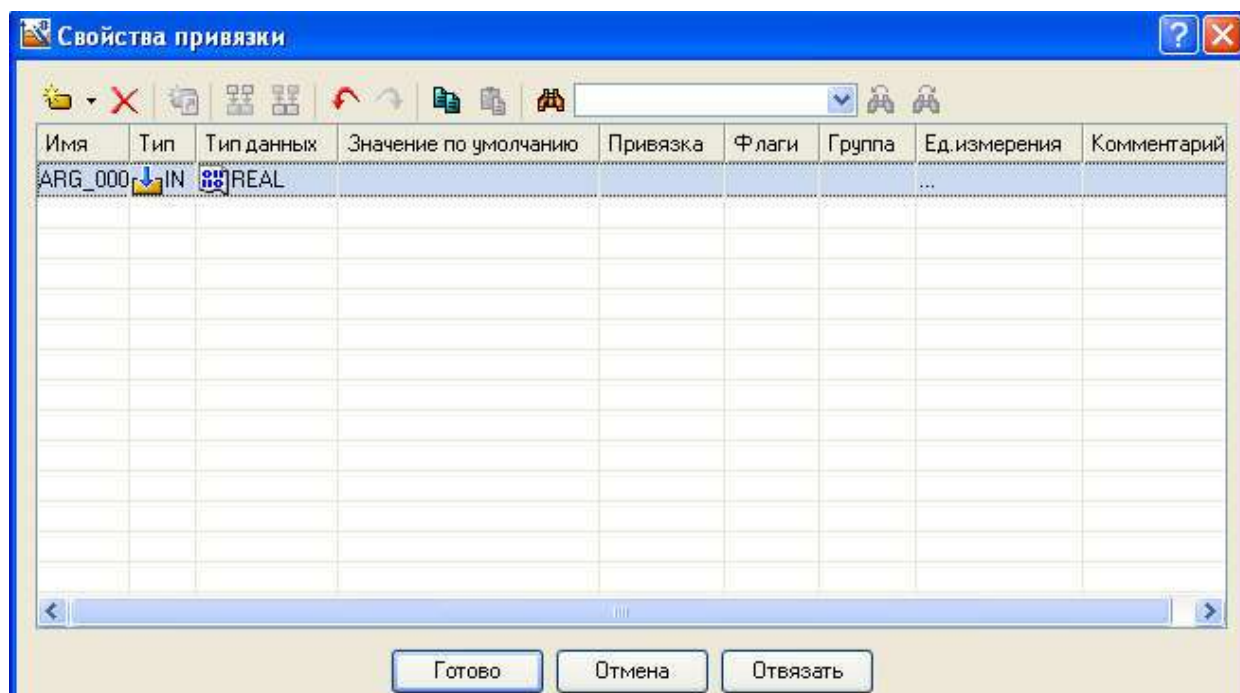


Рис. 14. Окно «Свойства привязки» с аргументом шаблона экрана «ARG_000»

- двойным щелчком ЛК выделим имя аргумента и изменим его, введя с клавиатуры «**Параметр**» (здесь и в дальнейшем ввод данных с клавиатуры будем завершать нажатием клавиши «Enter»);

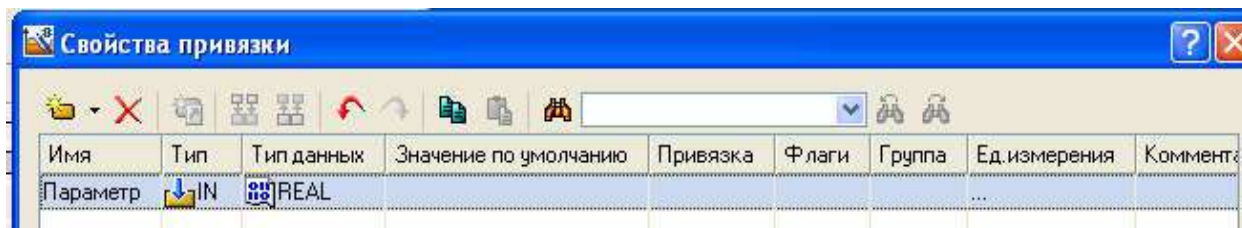


Рис. 15. Изменение имени аргумента «ARG_000» на «Параметр»

- подтвердим связь атрибута «Текст» ГЭ с данным аргументом щелчком ЛК по экранной кнопке «**Готово**»;
- закроем окно свойств ГЭ.

Графический экран будет иметь следующий вид:

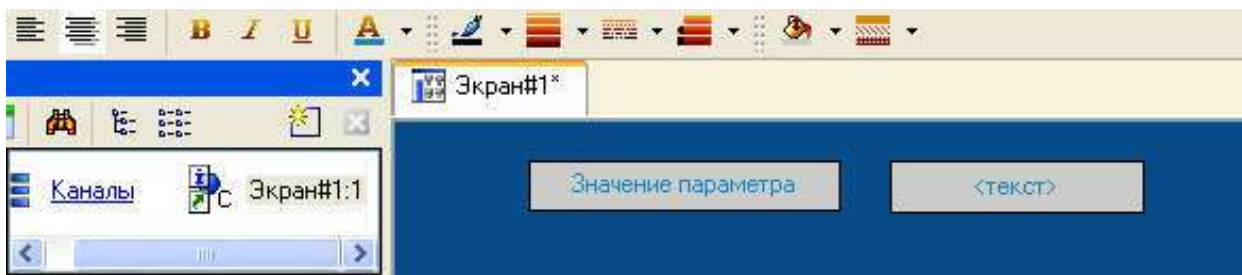


Рис. 16. Графический экран «Экран#1»

ПРАКТИЧЕСКАЯ РАБОТА № 2

Создание стрелочного прибора, привязка к аргументу

Цель работы – научиться создавать стрелочные приборы, задавать границы и производить настройку установок, создавать генератор синусоидальных сигналов и привязку его к каналу.

Порядок выполнения работы

Применим для отображения параметра новый тип ГЭ – «**Стрелочный прибор**». Для этого необходимо выполнить следующие действия:

- выделим двойным щелчком ЛК на инструментальной панели графического редактора иконку  и выберем из появившегося меню иконку «**Стрелочный прибор**»:

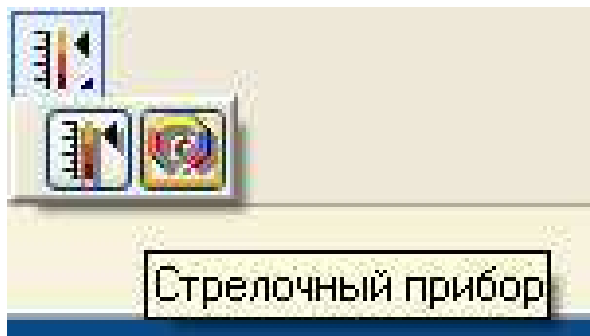


Рис. 17. Создание стрелочного прибора

- установим ГЭ, выбрав его размер таким, чтобы все элементы графики и текста на нем были разборчивы и симметричны;
- перейдем в режим редактирования и откроем окно свойств ГЭ;
- щелчком ЛК на экранной кнопке «**Основная привязка**» откроем окно табличного редактора аргументов шаблона экрана;
- ЛК выберем уже имеющийся аргумент «**Параметр**»;
- подтвердим выбор щелчком ЛК на кнопке «**Готово**»;

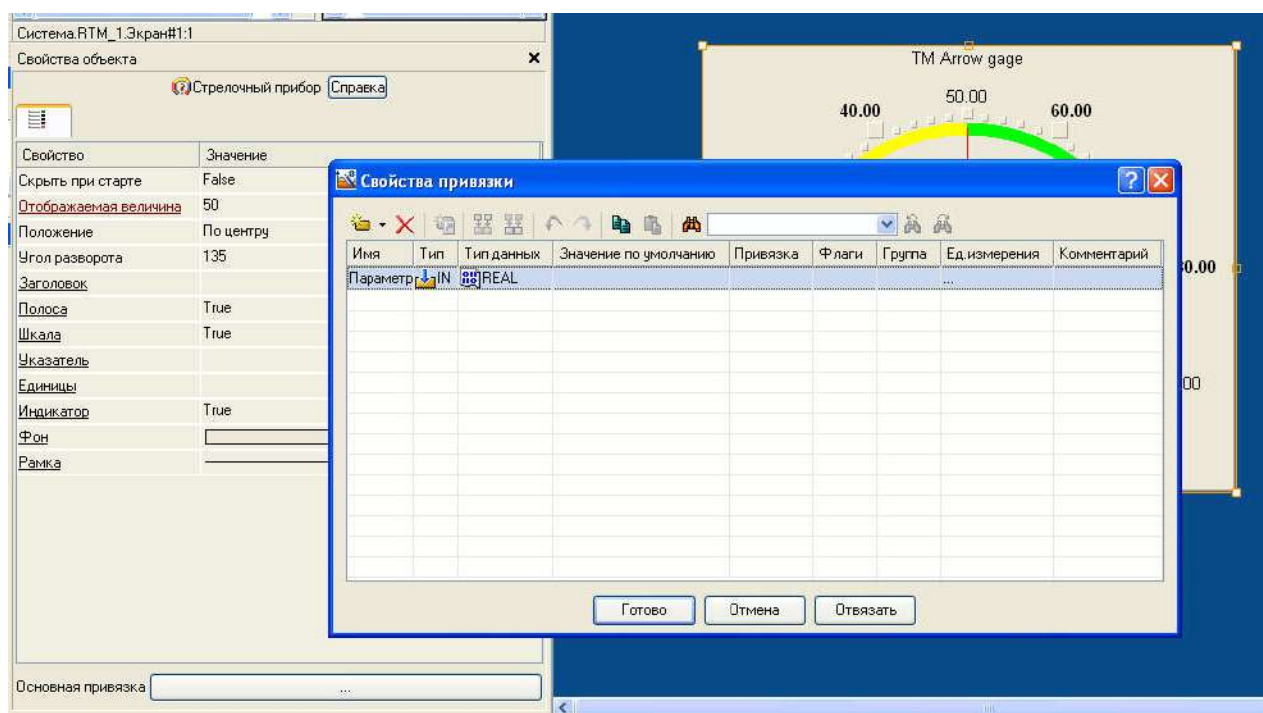


Рис. 18. Создание привязки

- двойным щелчком ЛК откроем атрибут «Заголовок» и в строке «Текст» введем надпись «Параметр»;

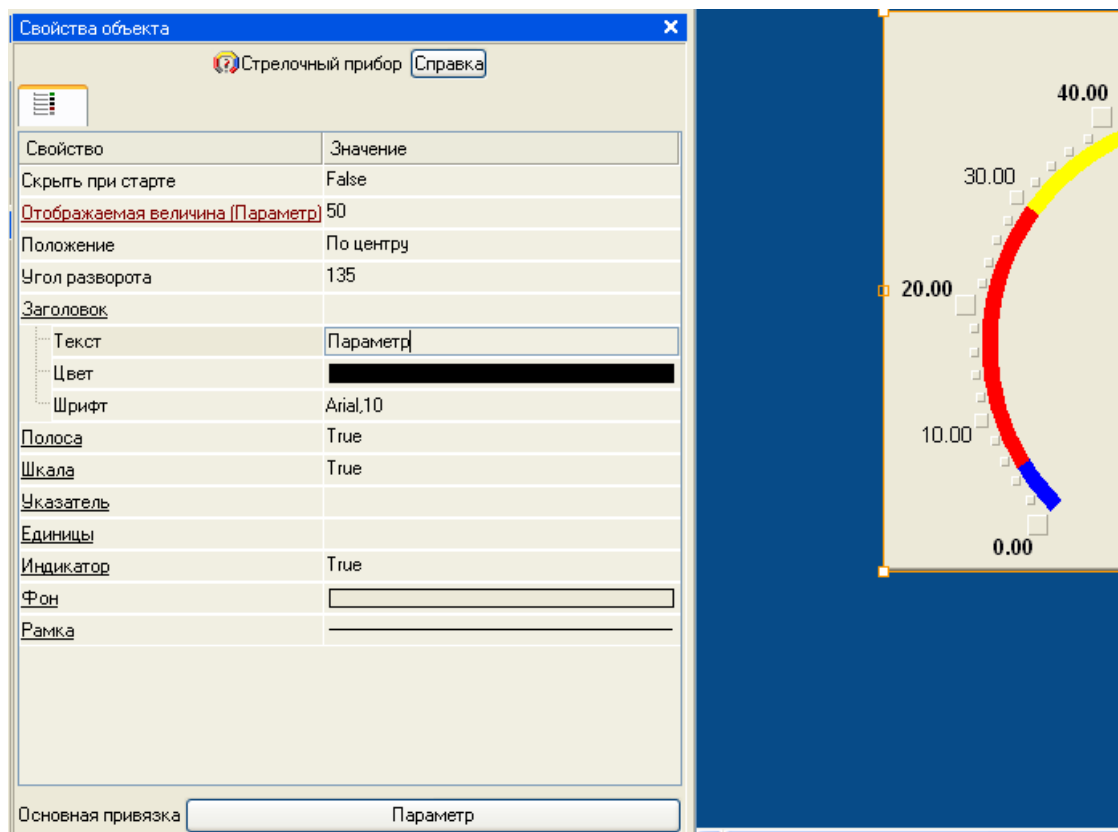


Рис. 19. Редактирование параметров объекта

- закроем окно свойств ГЭ.

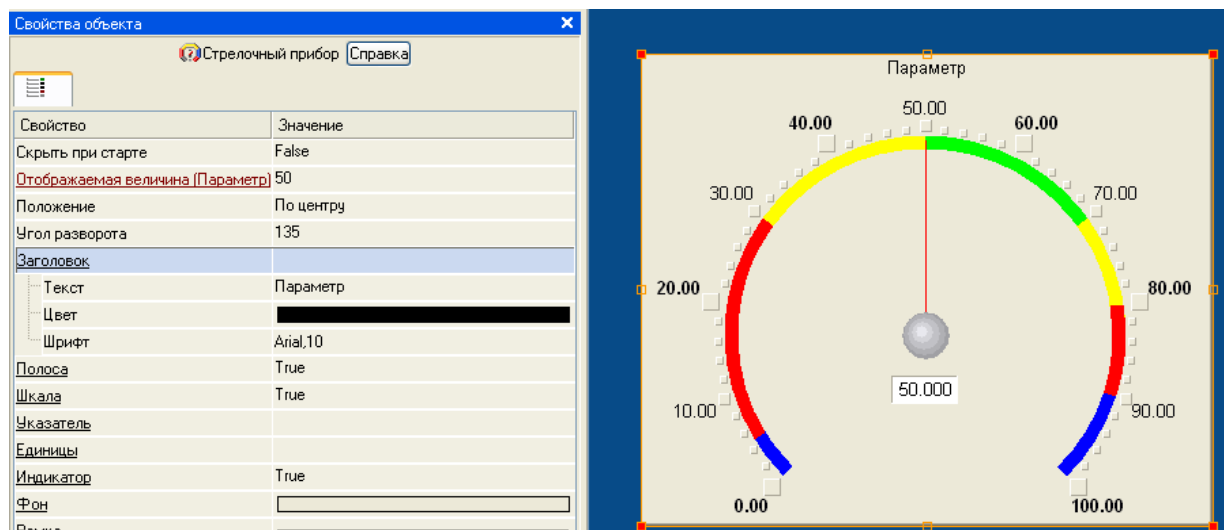


Рис. 20. Стрелочный прибор

Для проверки правильности привязок ГЭ к аргументам экрана можно воспользоваться режимом эмуляции. Переход в режим эмуляции осуществляется с помощью иконки  на панели инструментов. На экране графического редактора выводится окно задания значения аргумента в соответствующем поле:

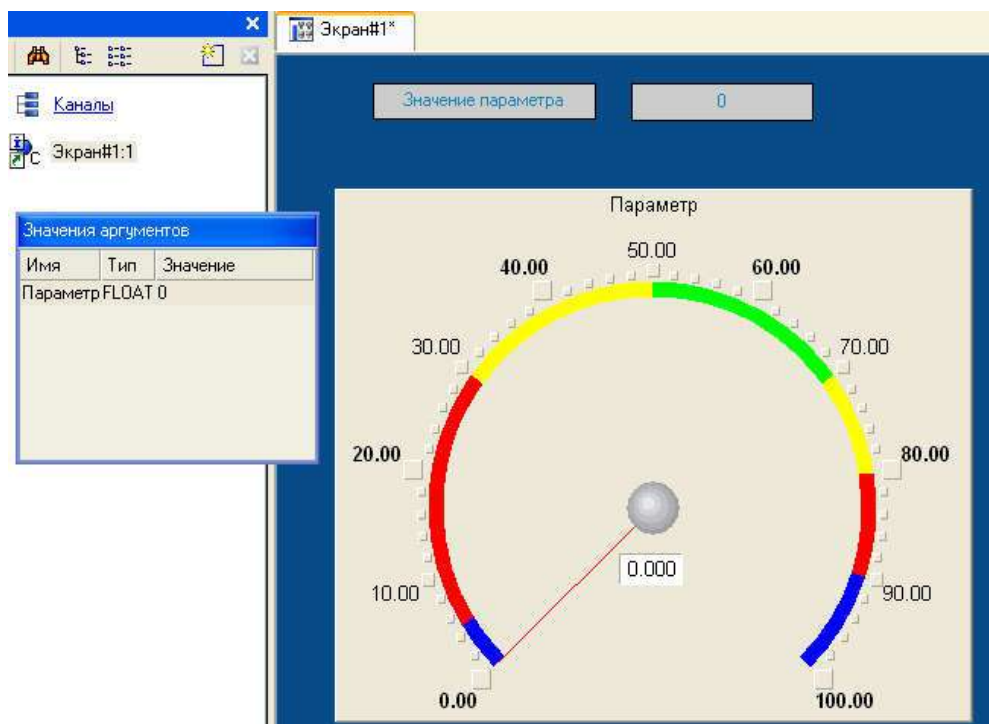


Рис. 21. Запуск режима эмуляции

Вводим «Значение аргумента», равное 12, «Стрелочный прибор» отображает введенное значение. Второй ГЭ на экране – «Значение параметра» также отображает введенное значение, привязки выполнены правильно.

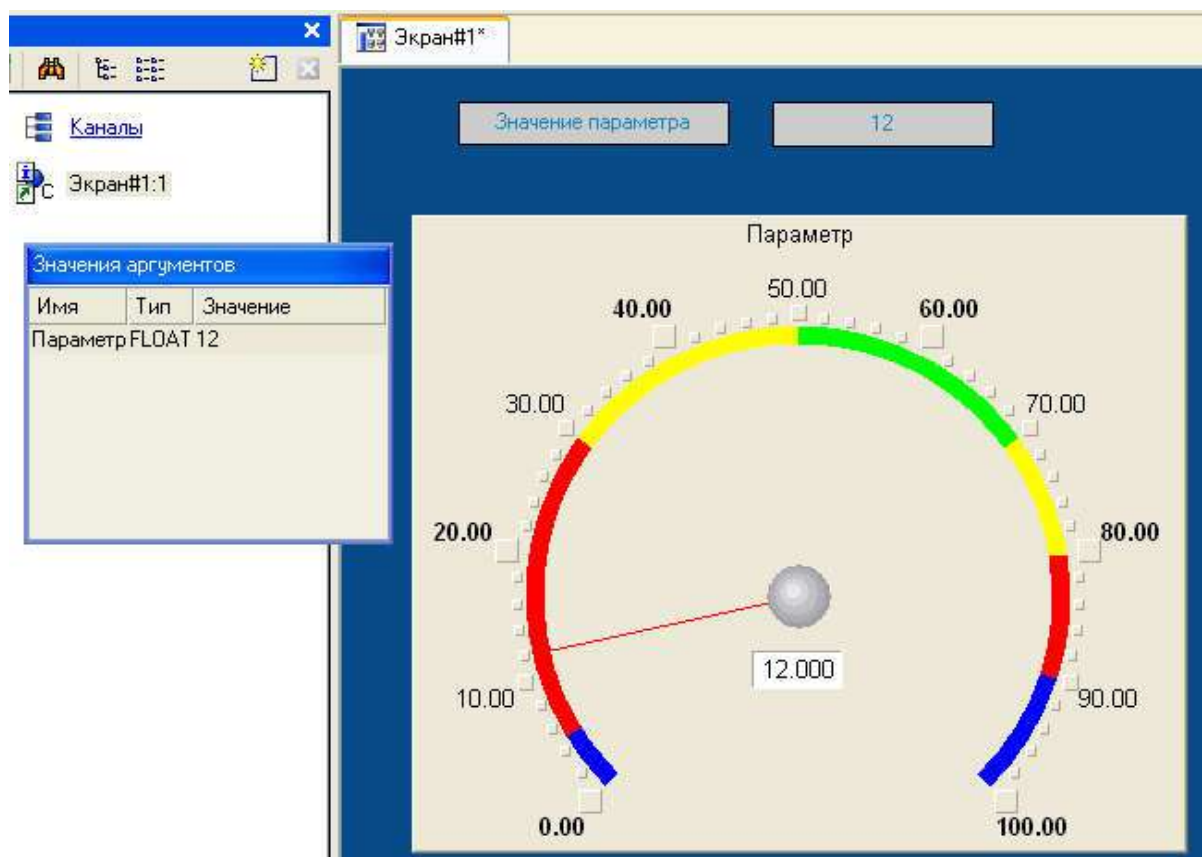


Рис. 22. Эмуляция

Выход из режима эмуляции – повторное нажатие ЛК по иконке



Автопостроение канала

Для создания «Канала» в узле проекта по аргументу шаблона экрана воспользуемся процедурой автопостроения:

- в слое «Система» выберем ЛК узел «RTM_1»;
- в поле компонентов узла ЛК выберем «Экран#1»;
- щелчком правой кнопки мыши (ПК) вызовем контекстное меню;
- в нем щелчком ЛК откроем свойства компонента «Экран#1»:

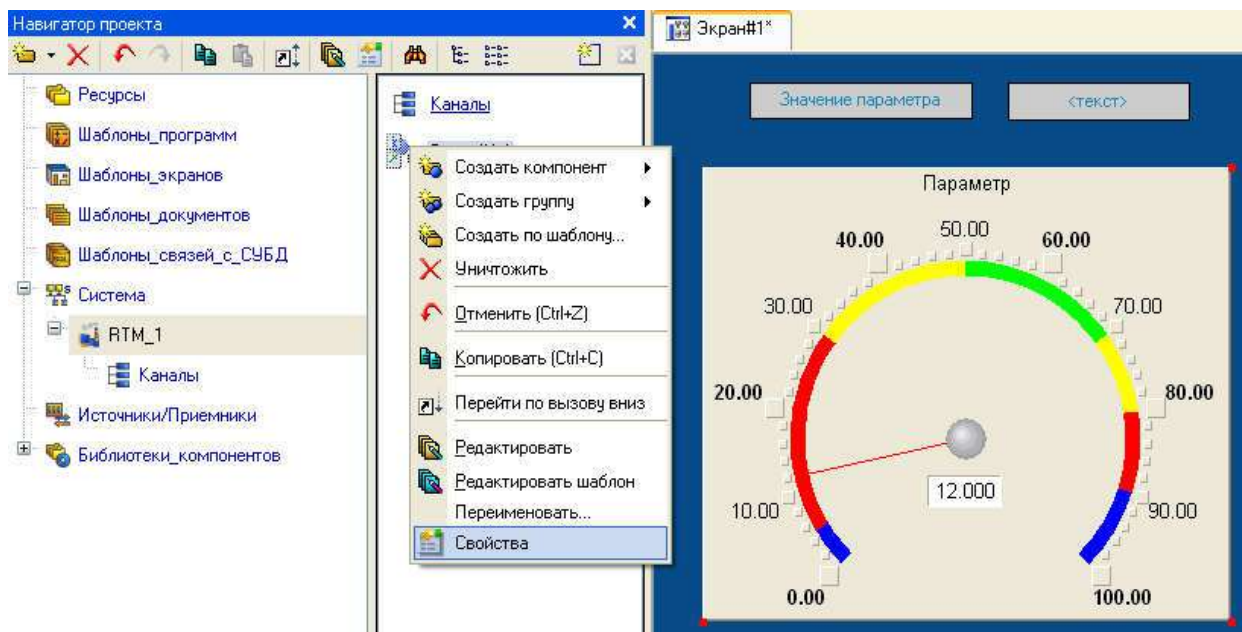


Рис. 23. Редактирование свойства

- выберем ЛК вкладку «Аргументы»:

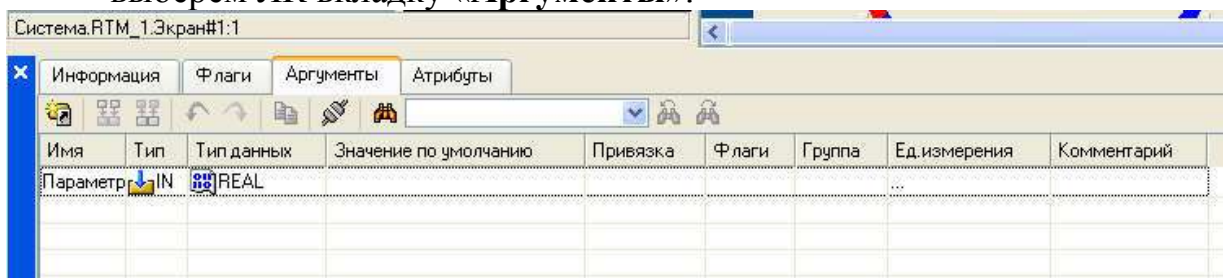


Рис. 24. Вкладка «Аргументы»

- выделим ЛК аргумент «Параметр» и с помощью иконки  создадим канал класса «Float» типа «Input» с именем «Параметр»:

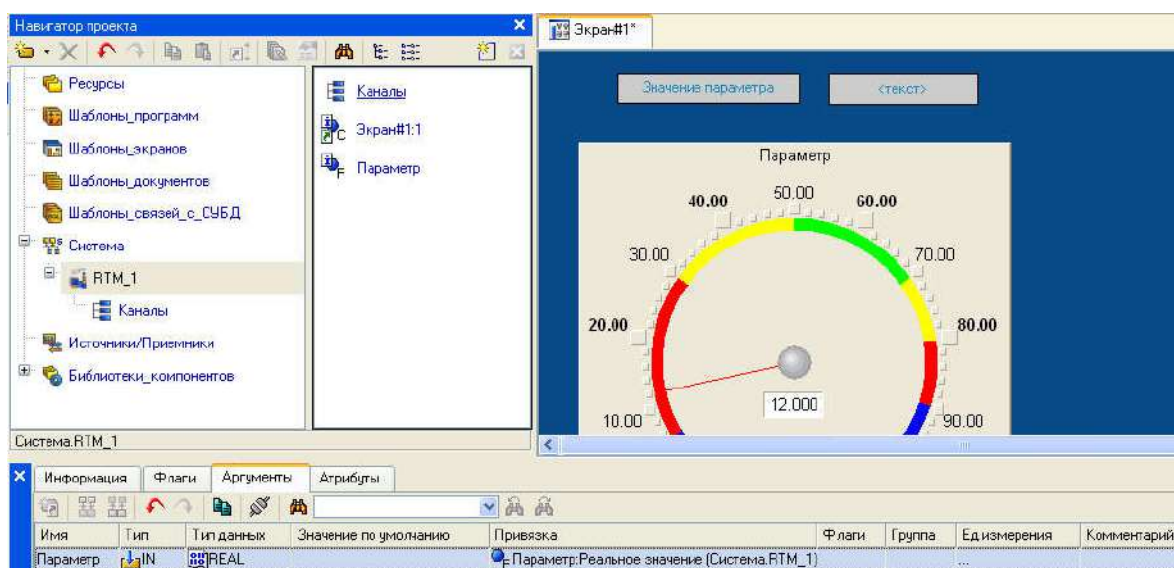


Рис. 25. Создание канала с именем «Параметр»

Задание границ и установок

Двойным щелчком ЛК по каналу «**Параметр**» откроем бланк редактирования его атрибутов и заполним раздел «**Границы**» следующим образом:

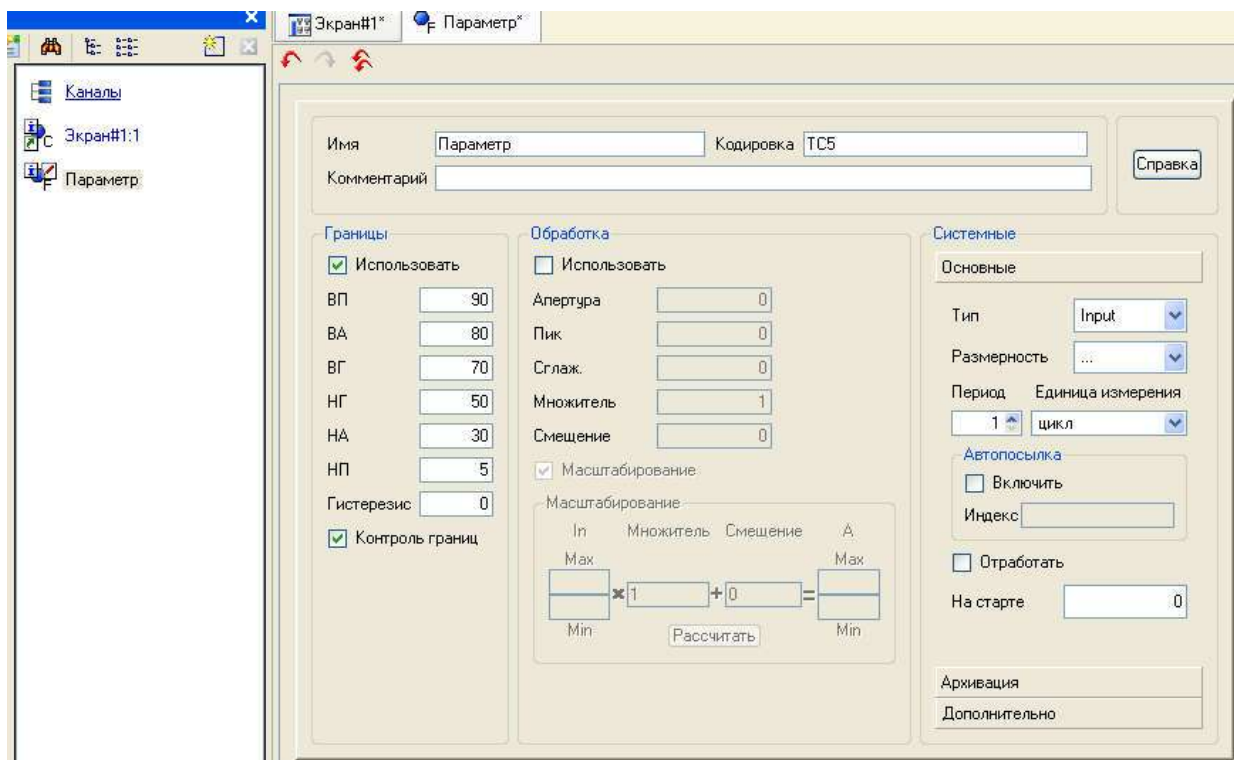


Рис. 26. Бланк редактирования атрибутов – заполнение раздела «**Границы**»

Создание генератора синусоиды и привязка его к каналу

Введем в состав проекта источник сигнала – внутренний генератор синусоиды, свяжем его с созданным каналом и опробуем в работе выполненные средства отображения. Для этого сделаем следующие действия:

- откроем слой «**Источники/Приемники**» и через ПК создадим в нем группу компонентов «**Генераторы**»:

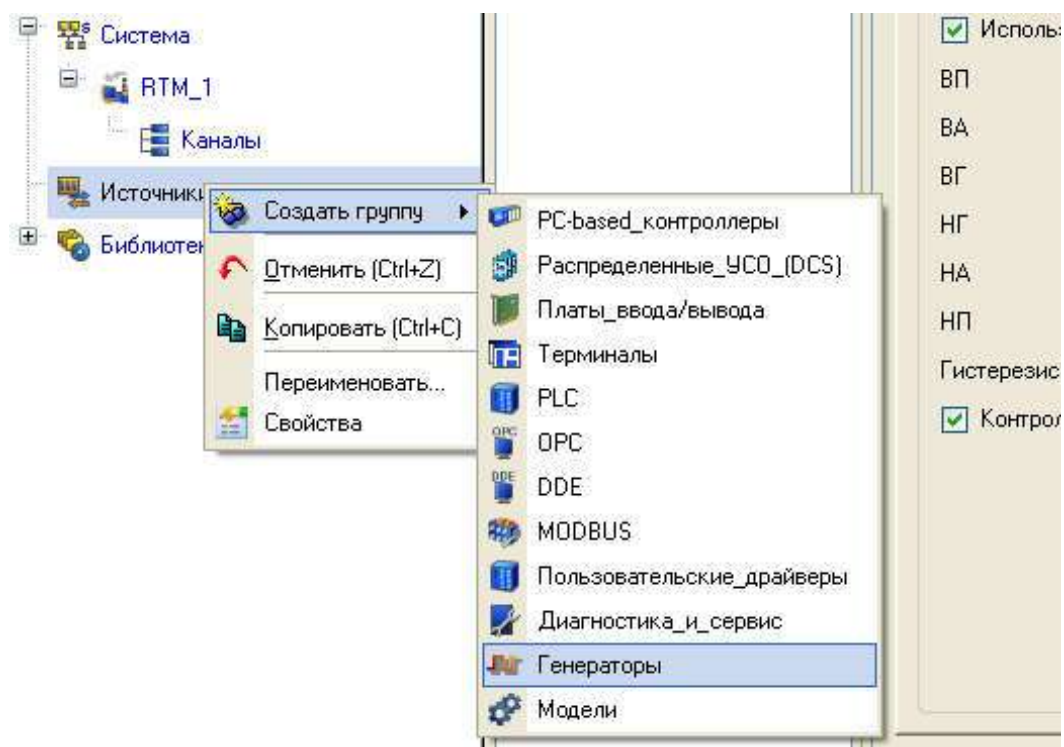


Рис. 27. Группа «Генераторы»

- двойным щелчком ЛК откроем группу «Генераторы_1» и через ПК создадим в ней компонент «Синусоида»;

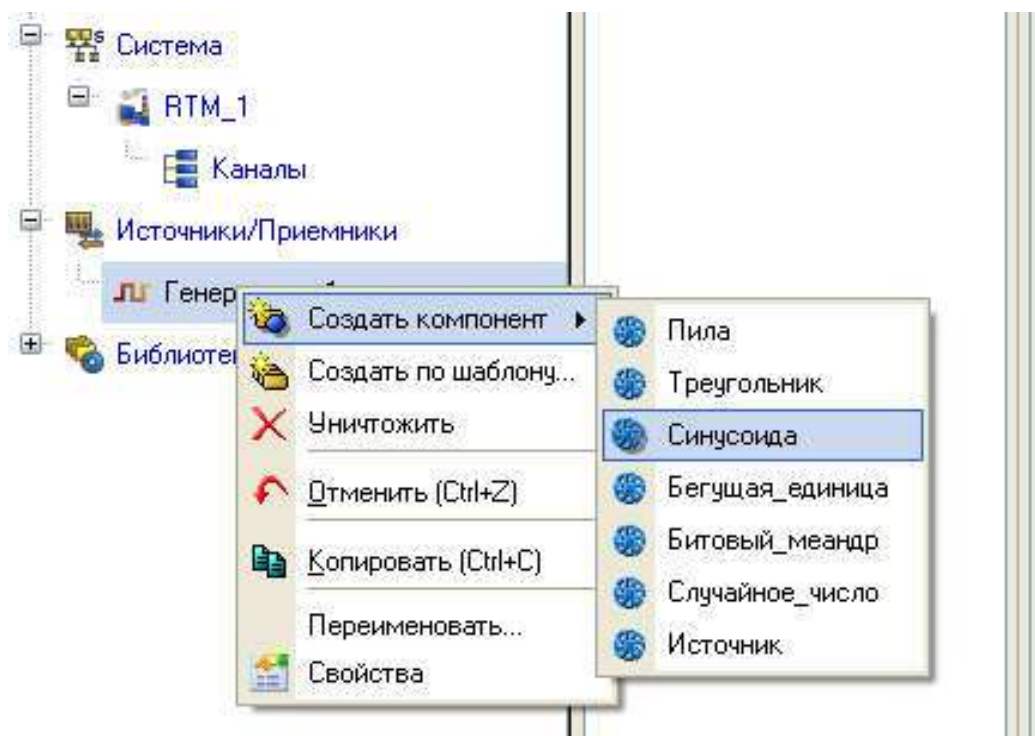


Рис. 28. Выбор компонента «Синусоида»

- захватим с помощью ЛК созданный источник и, не отпуская ЛК, перетащим курсор на узел «RTM_1» в слое «Система», а затем, в открывшемся окне компонентов, на канал «Параметр». Отпустим ЛК.

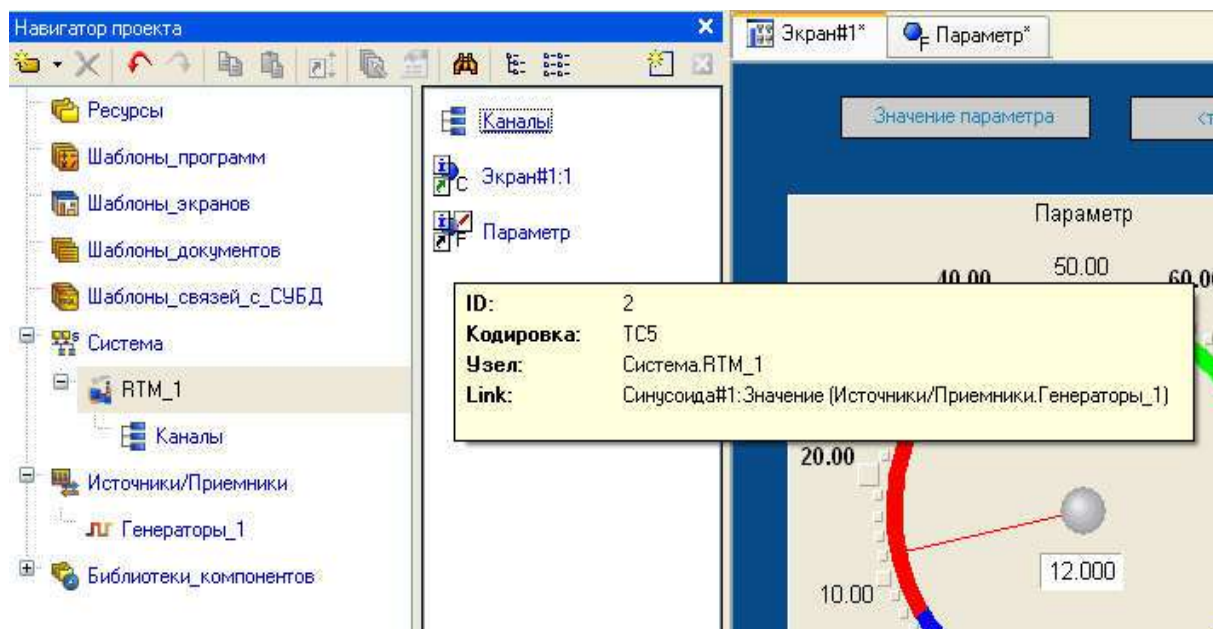


Рис. 29. «Навигатор проекта»

Запуск проекта

- Сохраним проект с помощью иконки , задав в открывшемся окне имя **Laba_1.prj**;
- на инструментальной панели выберем ЛК иконку  и подготовим проект для запуска в реальном времени;
- ЛК выделим в слое «Система» узел «RTM_1», а после, нажав ЛК иконку  на инструментальной панели, запустим профайлер;
- запуск/останов профайлера осуществляется с помощью иконки на его инструментальной панели или клавишной комбинации «Ctrl+R».

В открывшемся окне ГЭ справа от надписи «**Значение параметра**» должно показываться изменение синусоидального сигнала. То же значение должен отображать и стрелочный прибор:

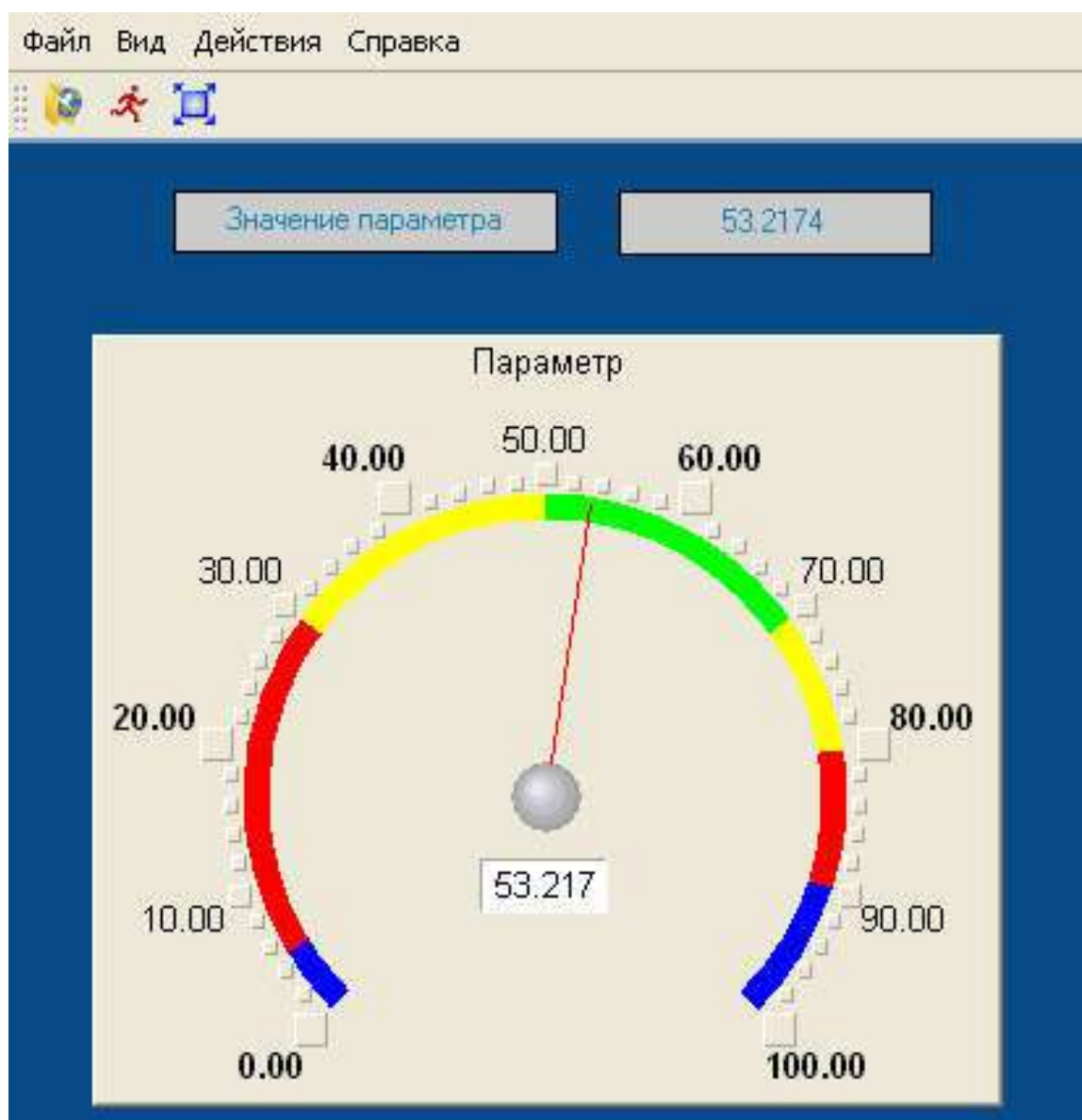


Рис. 30. Эмуляция проекта в реальном времени

ПРАКТИЧЕСКАЯ РАБОТА № 3

Добавление функции управления

Цель работы – научиться создавать графический элемент, позволяющий реализовать ввод числовых значений с клавиатуры, а также графический элемент, предназначенный для совместного просмотра изменений значений каналов узла во времени и отслеживания предыстории – тренд.

Порядок выполнения работы

Введем в состав графического экрана ГЭ, позволяющий реализовать ввод числовых значений с клавиатуры. Создадим новый аргумент шаблона экрана для их приема.

Редактирование графического экрана

- вызовем графический экран на редактирование;
- на инструментальной панели графического редактора выберем ЛК иконку ГЭ Кнопка – ;
- с помощью мыши разместим ГЭ в поле экрана под ГЭ .

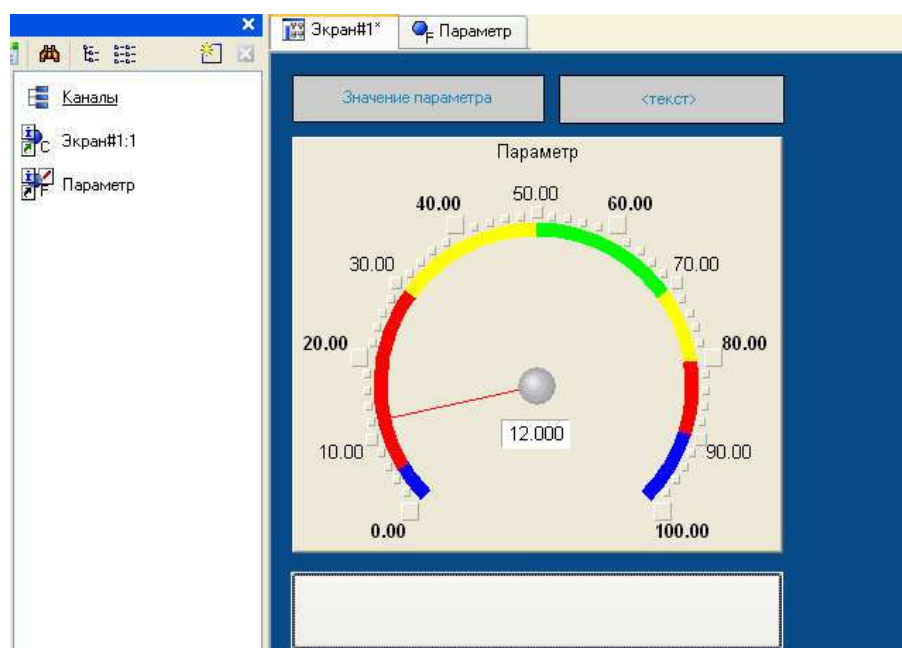


Рис. 31. Графический экран

- перейдем в режим редактирования , выделим ГЭ  ЛК и вызовем окно его свойств:

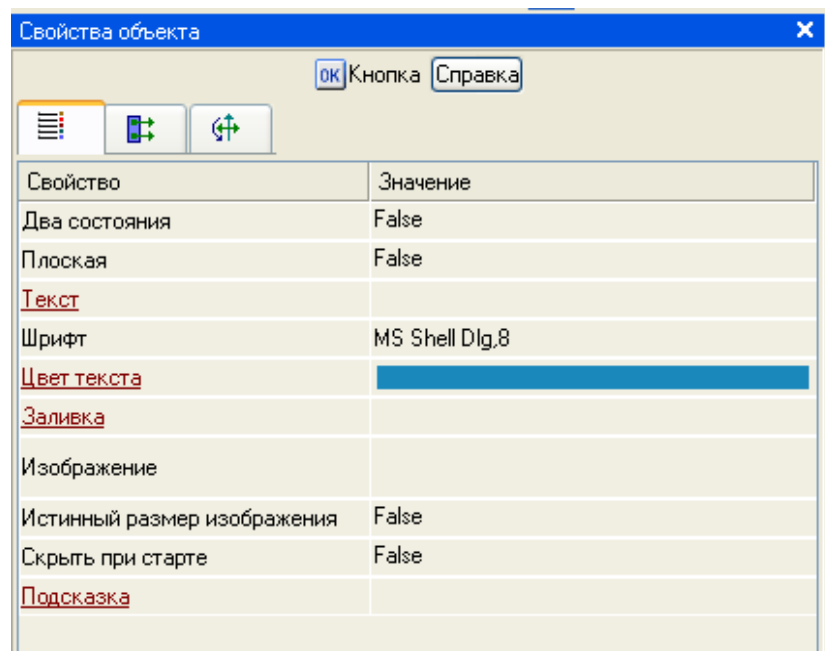


Рис. 32. Редактирование графического экрана

- в поле «Текст» введем надпись «Управление»;
- откроем бланк «События»  и ПК раскроем меню «По нажатию (mousePressed)»;
- выберем из списка команду «Добавить Передать значение»;

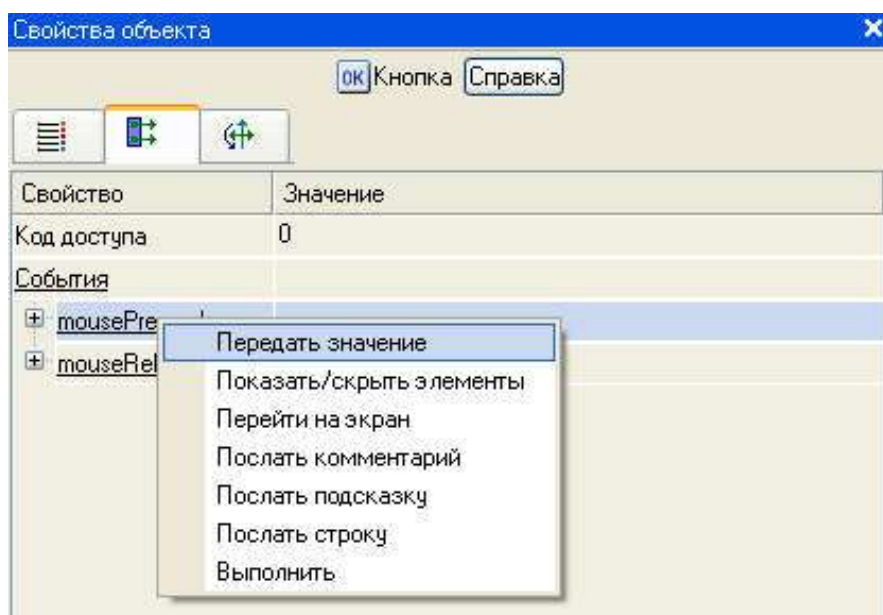


Рис. 33. Свойства объекта

- в раскрывшемся меню настроек выбранной команды в поле «**Тип передачи**» выберем из списка «**Ввести и передать**»:

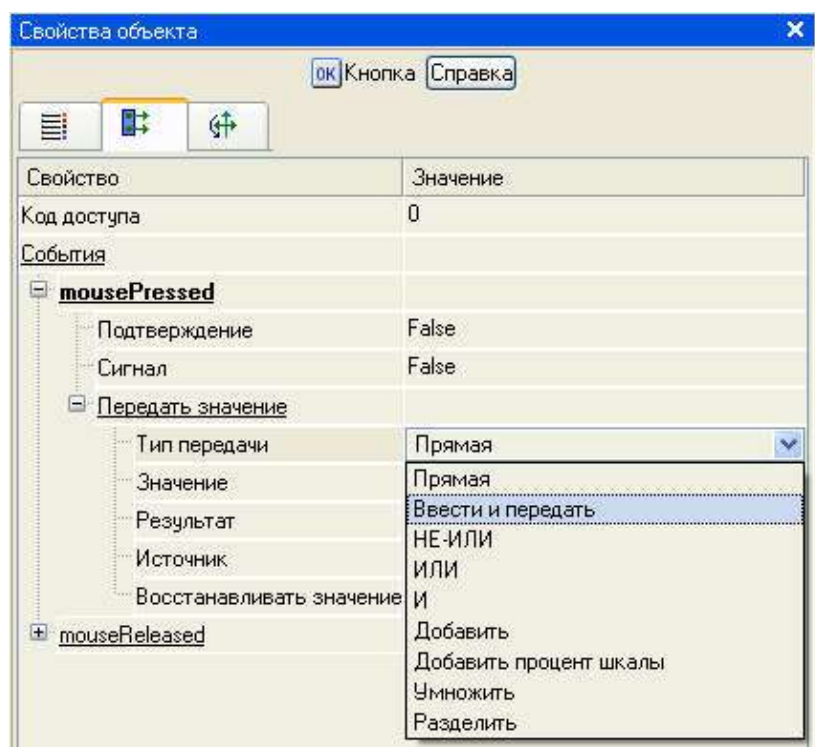


Рис. 34. Редактирование свойств

- щелчком ЛК в поле «**Результат**» вызовем табличный редактор аргументов;
- создадим еще один аргумент и зададим ему имя «**Управление**»;

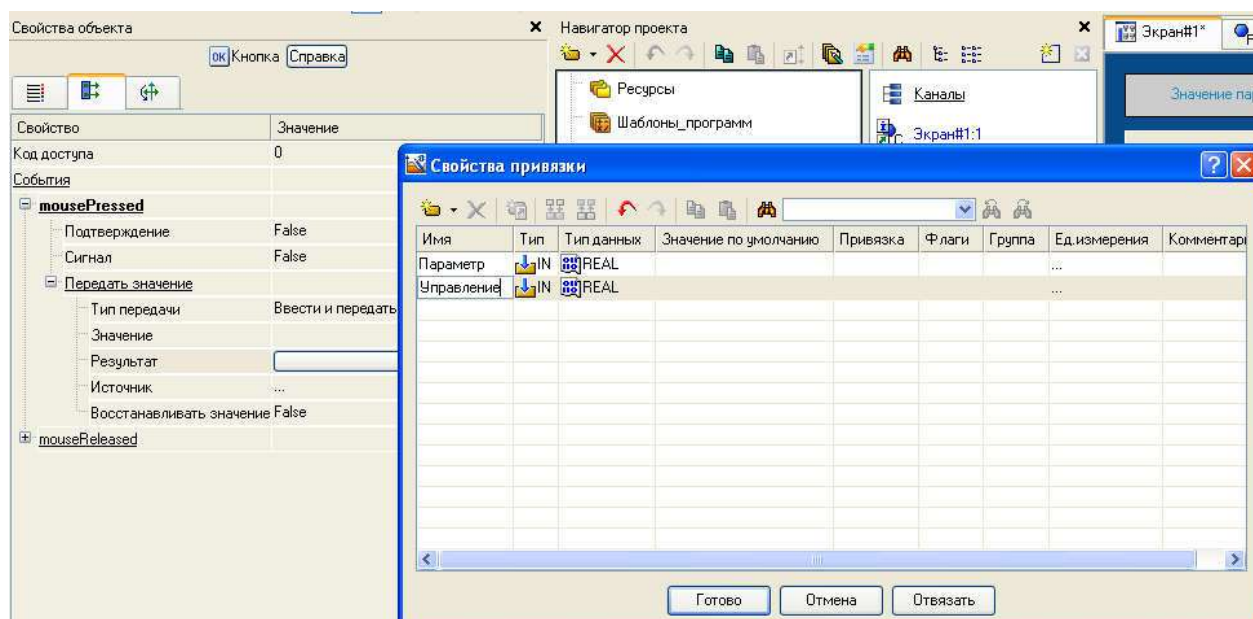


Рис. 35. Свойства привязки «Управление»

- изменим тип аргумента на «IN/OUT», кнопкой «Готово» подтвердим привязку атрибута ГЭ к этому аргументу:

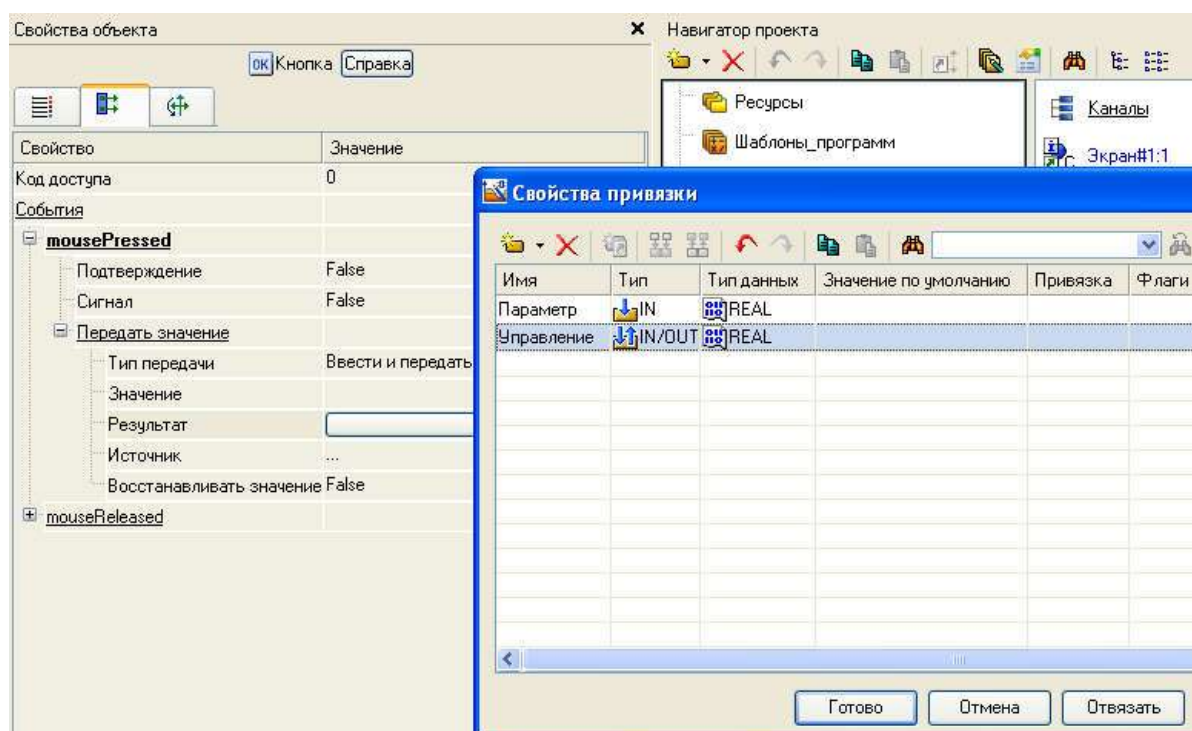


Рис. 36. Изменение типа аргумента «Управление»

- закроем окно свойств ГЭ

Далее выполним размещение ГЭ «Текст» для отображения вводимого с клавиатуры значения. Воспользуемся уже имеющимся на графическом экране ГЭ путем его копирования/вставки и перепривязки. Для этого:

- выделим ЛК ГЭ «Текст», выполненный для отображения аргумента «Параметр»:

- с помощью иконки  на панели инструментов или комбинацией клавиш «Ctrl+C» скопируем выделенный ГЭ «Текст» в буфер обмена;

- далее с помощью иконки  или комбинацией клавиш «Ctrl+V» извлечем копию ГЭ из буфера обмена и поместим ее на графический экран;

- переместим, удерживая нажатой ЛК, копию ГЭ «Текст» справа от размещенного на экране ГЭ «Кнопка»;

- двойным щелчком ЛК на перемещенном ГЭ «Текст» откроем окно его свойств;

- двойным щелчком ЛК на строке «Текст» вкладки основных свойств



перейдем к настройке динамизации данного атрибута ГЭ;

- в правом поле строки «Привязка» щелчком ЛК откроем табличный редактор аргументов шаблона экрана;

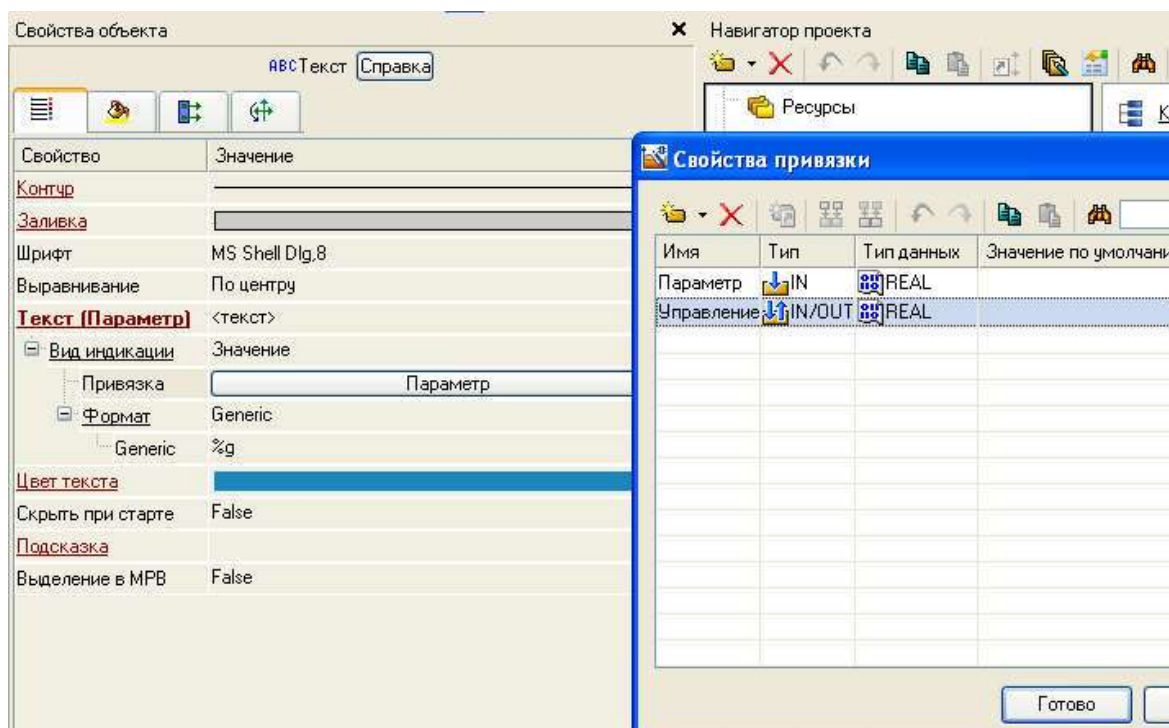


Рис. 37. Свойства привязки

- выделим ЛК в списке аргумент «Управление» и щелчком ЛК по экранной кнопке «Готово» подтвердим привязку атрибута ГЭ «Текст» к данному аргументу шаблона экрана;

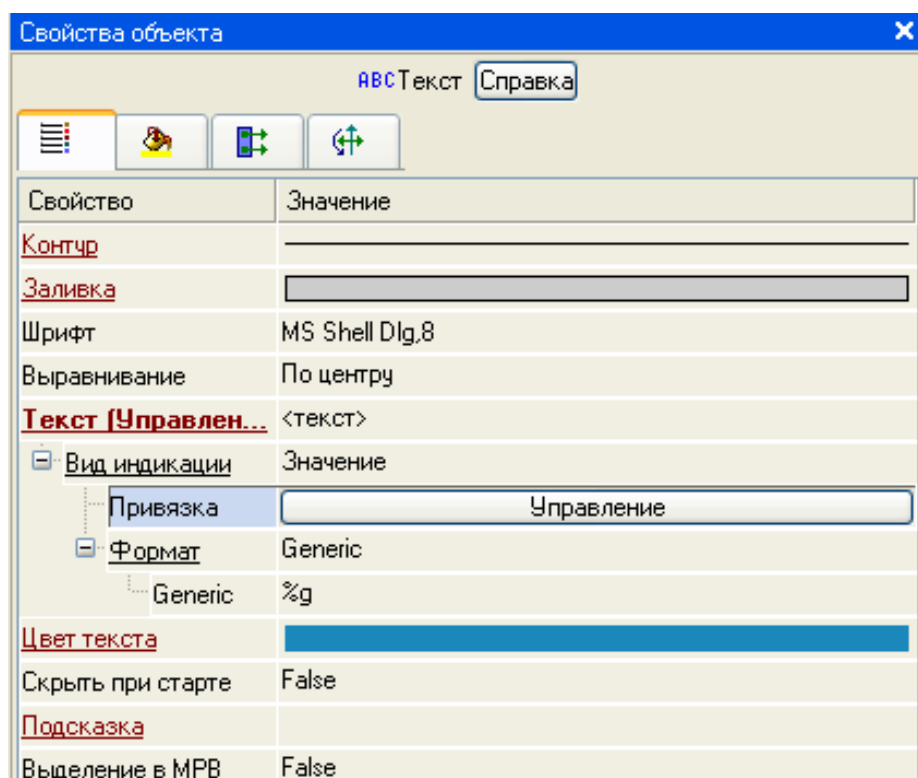


Рис. 38. Редактирование «Свойства объекта»

- закроем окно свойств ГЭ «Текст».

Привязка аргумента экрана к каналу

Создадим по аргументу «Управление» шаблона экрана новый канал, отредактируем привязку атрибута канала к аргументу шаблона экрана. Для этого:

- в слое «Система» откроем узел «RTM_1»;
- щелчком ПК вызовем через контекстное меню свойства компонента «Экран#1»:

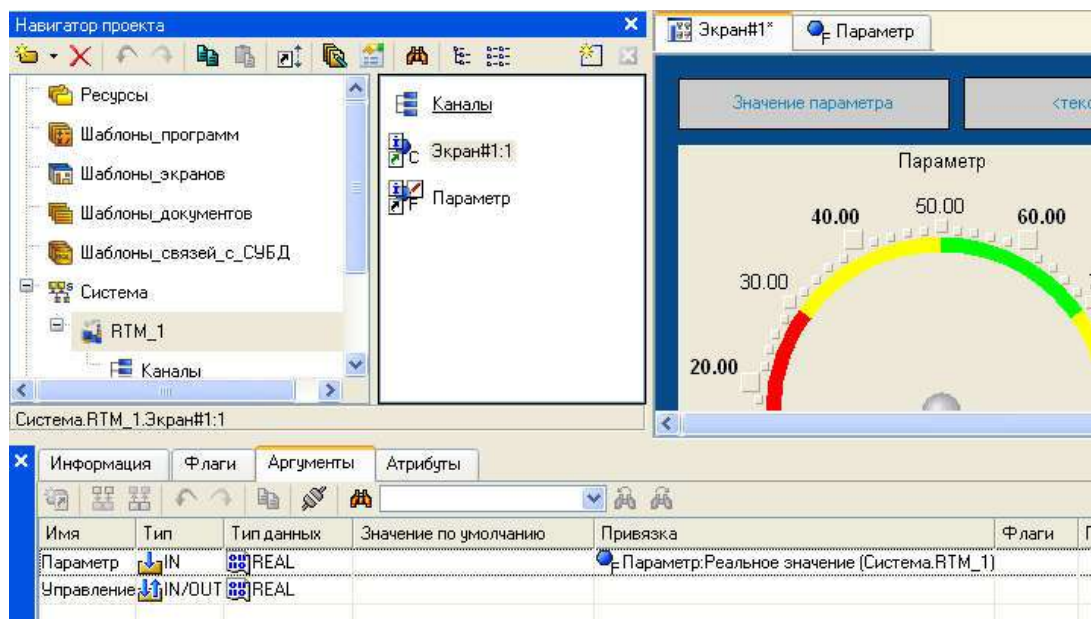


Рис. 39. Навигатор проекта

- выберем вкладку «Аргументы», ЛК выделим аргумент «Управление» и с помощью «иконки»  выполним автопостроение канала;
- в результате в узле «RTM_1» будет создан канал с именем «Управление»:

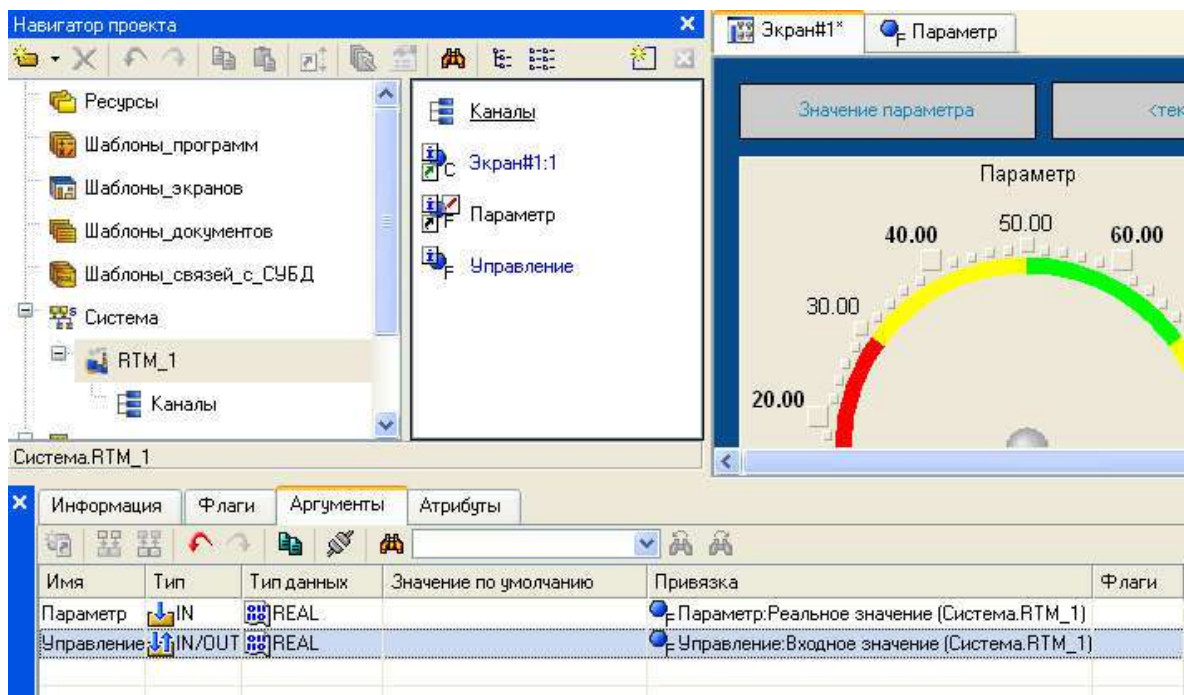


Рис. 40. Редактирование вкладки «Аргумент»

- двойным щелчком ЛК в поле «Привязка» аргумента «Управление» вызовем окно настройки связи, выберем в нем атрибут «Входное значение» канала «Управление» и кнопкой «Привязка» подтвердим связь аргумента экрана «Управление» с атрибутом «Входное значение» канала «Управление»:

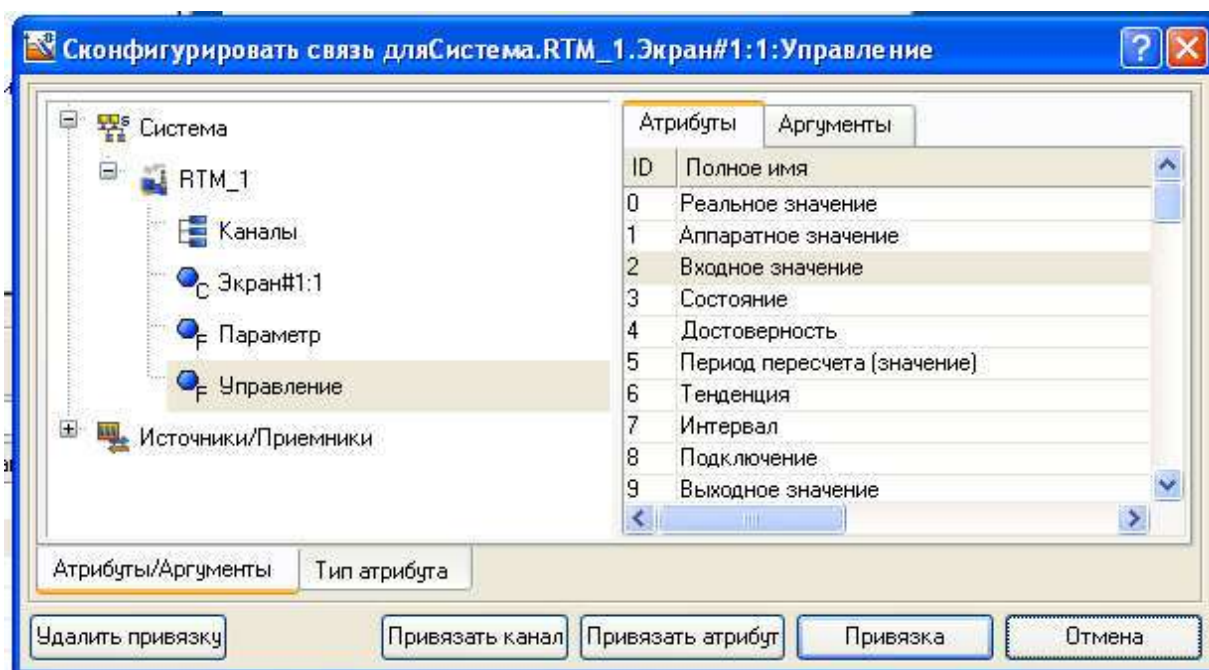


Рис. 41. Настройка связи

- закроем окно свойств компонента «Экран#1».

Размещение ГЭ «Тренд»

Дополним созданный экран новым ГЭ для совместного просмотра изменений значений каналов узла во времени и отслеживания предыстории – трендом.

В правой части графического экрана разместим ГЭ «Тренд»  для вывода значений «Параметр» и «Управление».

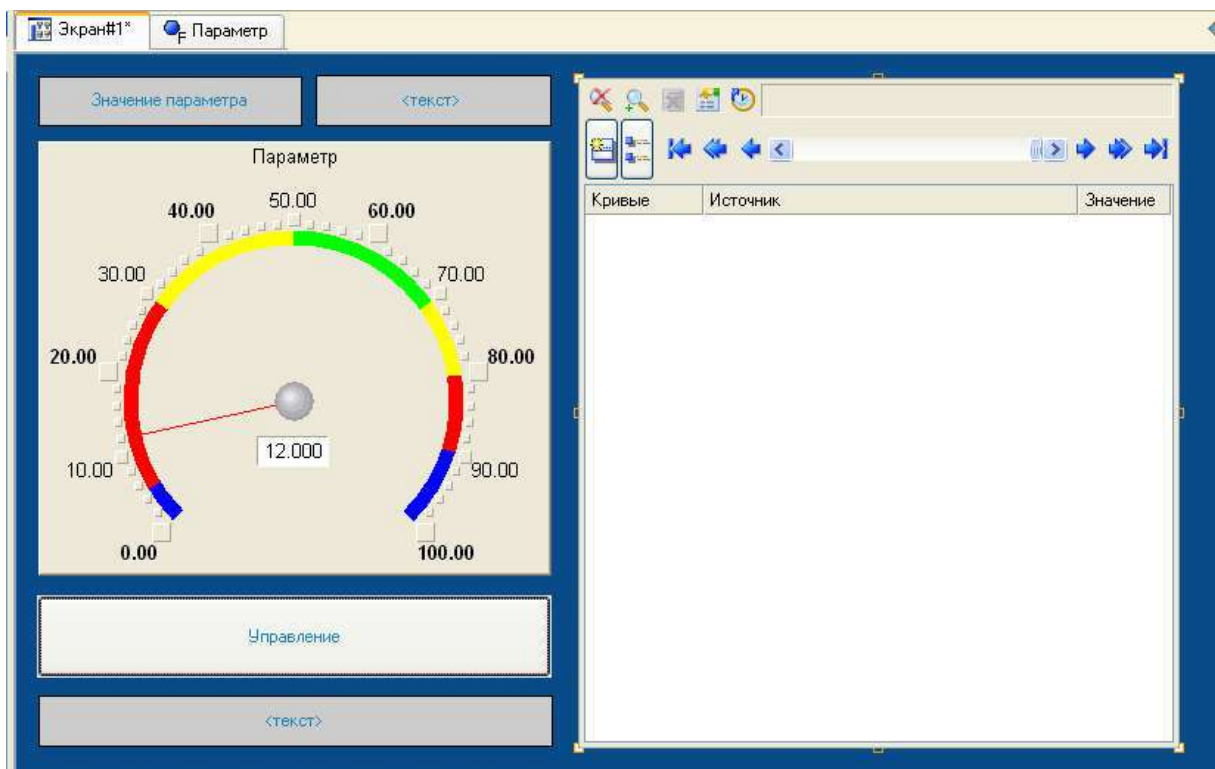


Рис. 42. Размещение ГЭ «Тренд»

Основные свойства ГЭ  оставим заданными по умолчанию. Перейдем во вкладку  и, выделив ЛК строку «**Кривые**», с помощью ПК создадим две новых кривых. Настроим для них привязки к существующим аргументам, толщину и цвет линий:



Рис. 43. Редактирование свойств ГЭ «Тренд»

ГЭ примет вид:

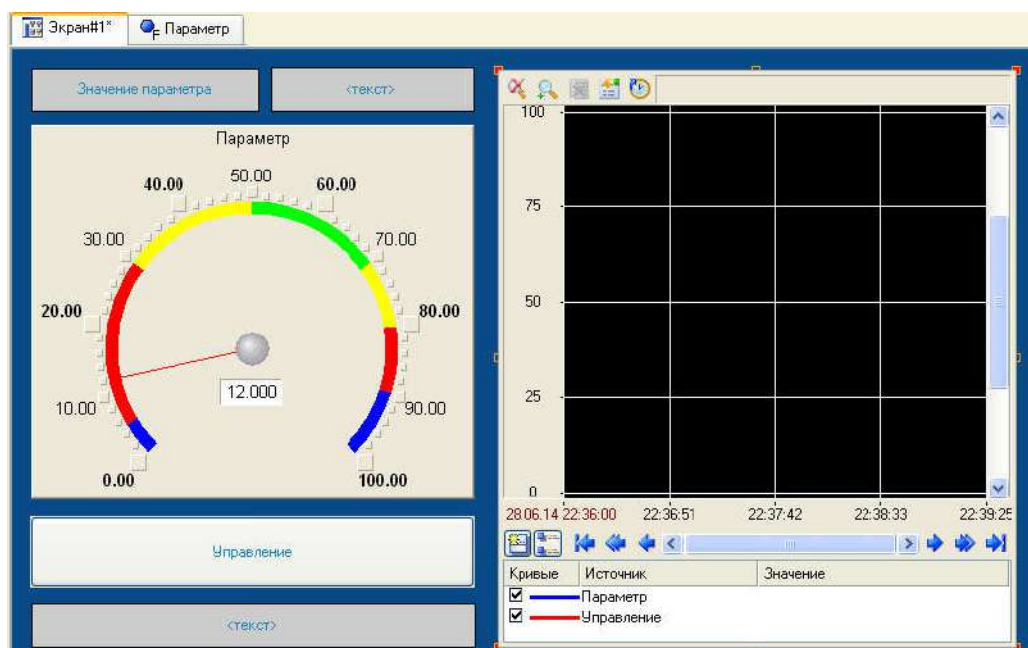


Рис. 44. Графический элемент «Тренд»

Запуск проекта

- Сохраним проект;
- на инструментальной панели выберем ЛК  для запуска проекта в реальном времени;
- с помощью иконки  на инструментальной панели запустим проект на исполнение.

Теперь с помощью кнопки «Управление» вводим величину управляющего воздействия, равную 55, и наблюдаем результат в соседнем поле и тренде:

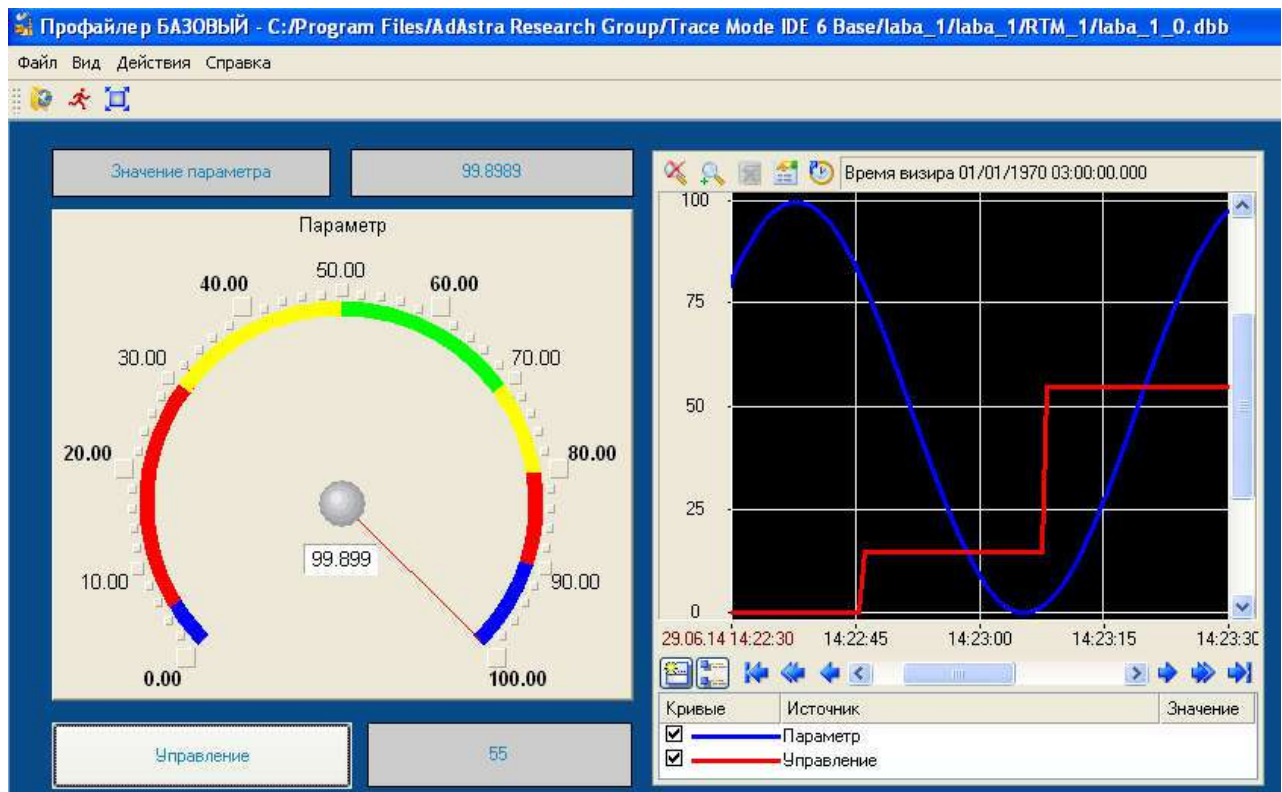


Рис. 45. Запуск «Проекта»

ПРАКТИЧЕСКАЯ РАБОТА № 4

Простейшая обработка данных

Цель работы – научиться основам программирования в TRACE MODE.

Порядок выполнения работы

С помощью нового компонента проекта – шаблона программы свяжем два имеющихся канала операцией сложения. Будем суммировать реальные значения каналов «**Параметр**» и «**Управление**», а результат помещать во вновь созданный аргумент экрана «**Сумма**» (с отображением на ГЭ «**Текст**» и «**Тренд**») без создания дополнительного канала в узле проекта.

Доработка графического экрана

- Скопируем два первых ГЭ – «**Значение параметра**» и «**Текст**» и разместим их ниже ГЭ «**Кнопка**»;
- изменим статический текст первого ГЭ на «**Сумма**»:

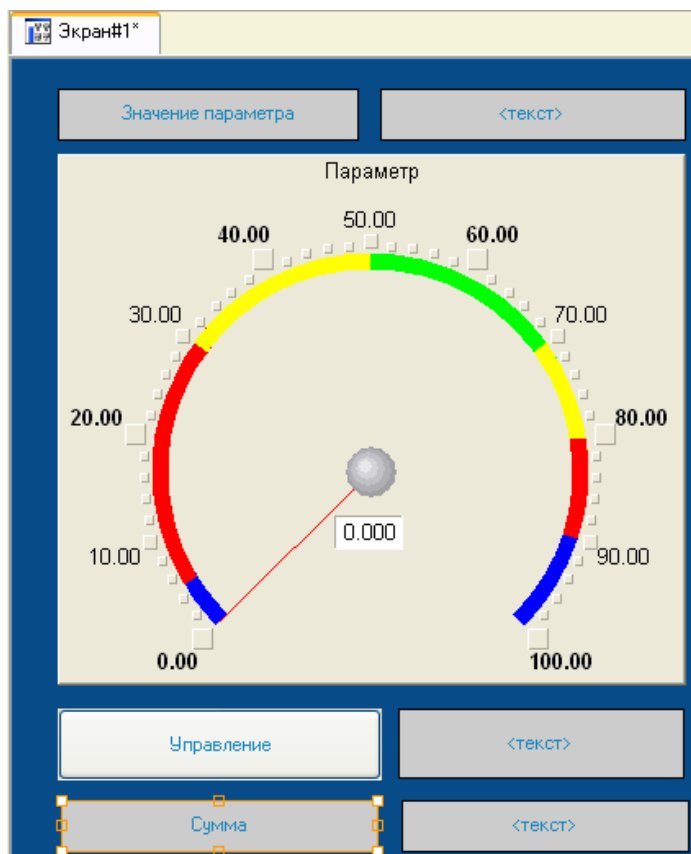


Рис. 46. ГЭ «Стрелочный прибор»

- динамику второго ГЭ привяжем к новому – третьему – аргументу шаблона экрана типа «IN» с именем «Сумма», который создадим в процессе привязки:

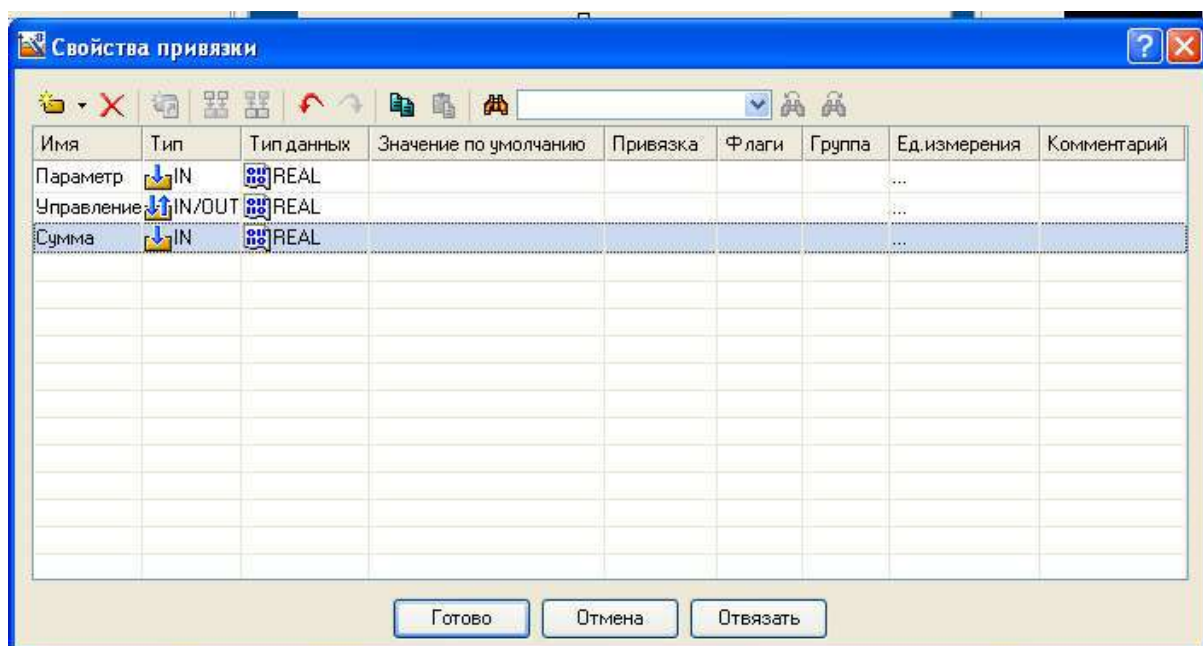


Рис. 47. «Свойства привязки»

- добавим еще одну кривую на тренд с привязкой к аргументу «Сумма».

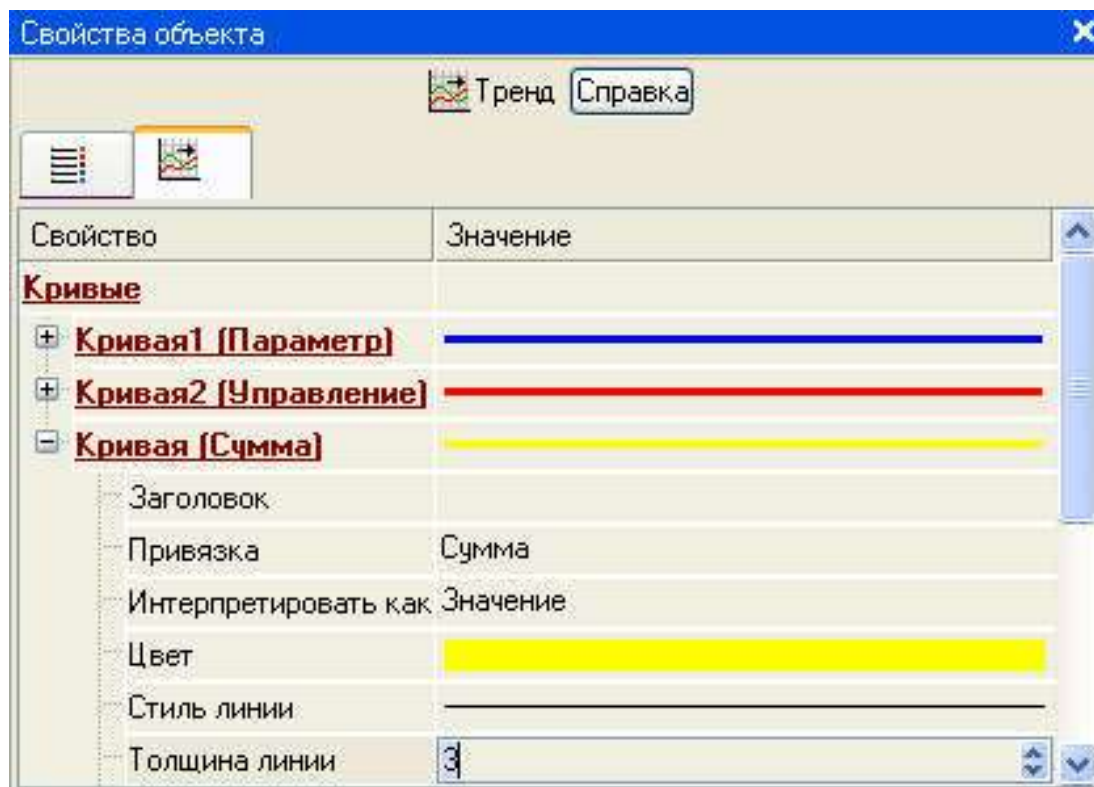


Рис. 48. Редактирование «Свойства объекта»

Создание программы на языке Техно ST

Создадим программу, в которой сумма двух аргументов, связанных с атрибутами «**Реальное значение**» каналов «**Параметр**» и «**Управление**», будет помещается в третий аргумент с именем «**Сумма**». В дальнейшем воспользуемся возможностью связывания аргументов шаблонов для вывода на экран результата работы программы без создания дополнительного канала.

Двойным щелчком ЛК откроем узел «**RTM_1**»:

- создадим в нем компонент «**Программа**»;

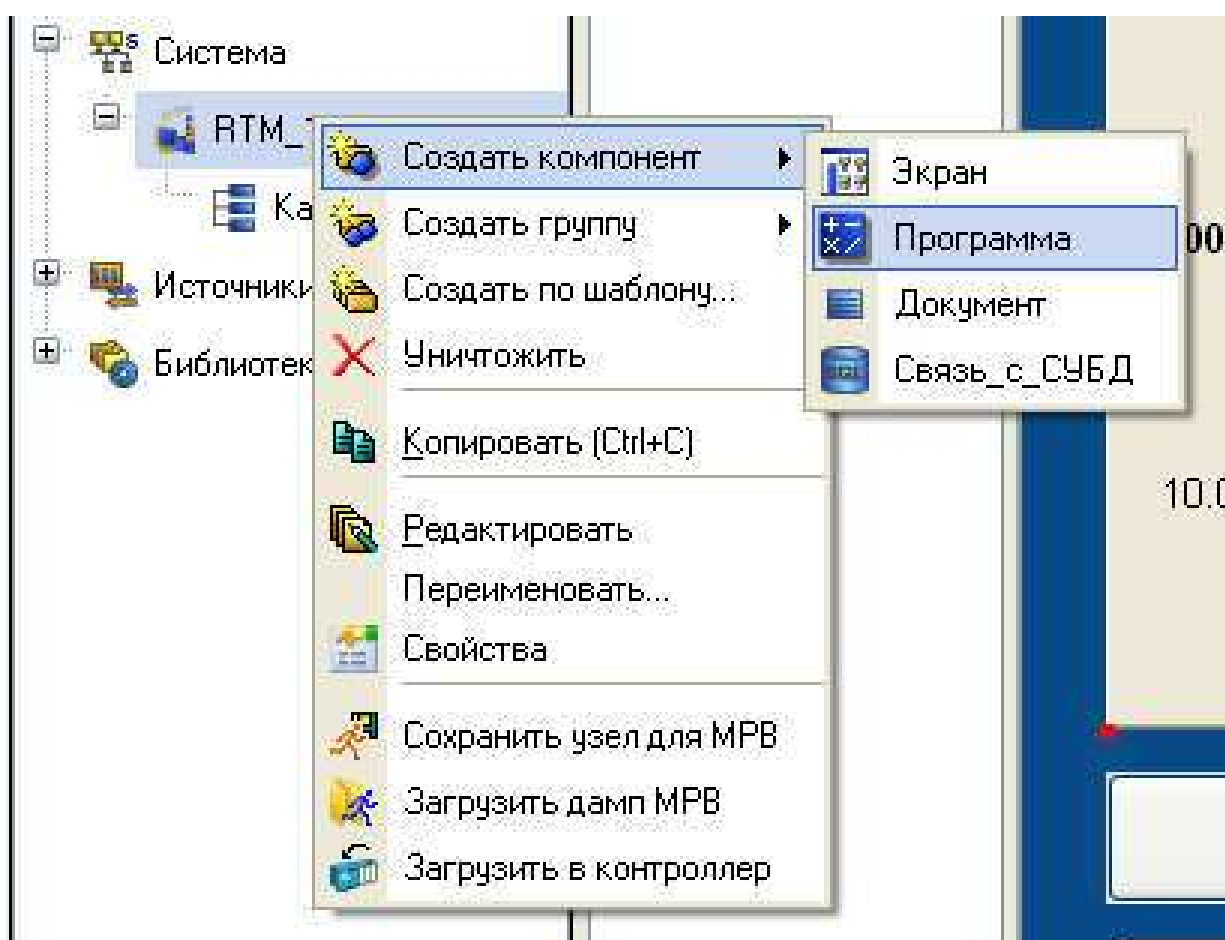


Рис. 49. Создание компонента «**Программа**»

- двойным щелчком ЛК по компоненту «**Программа#1**» перейдем в режим редактирования программы:

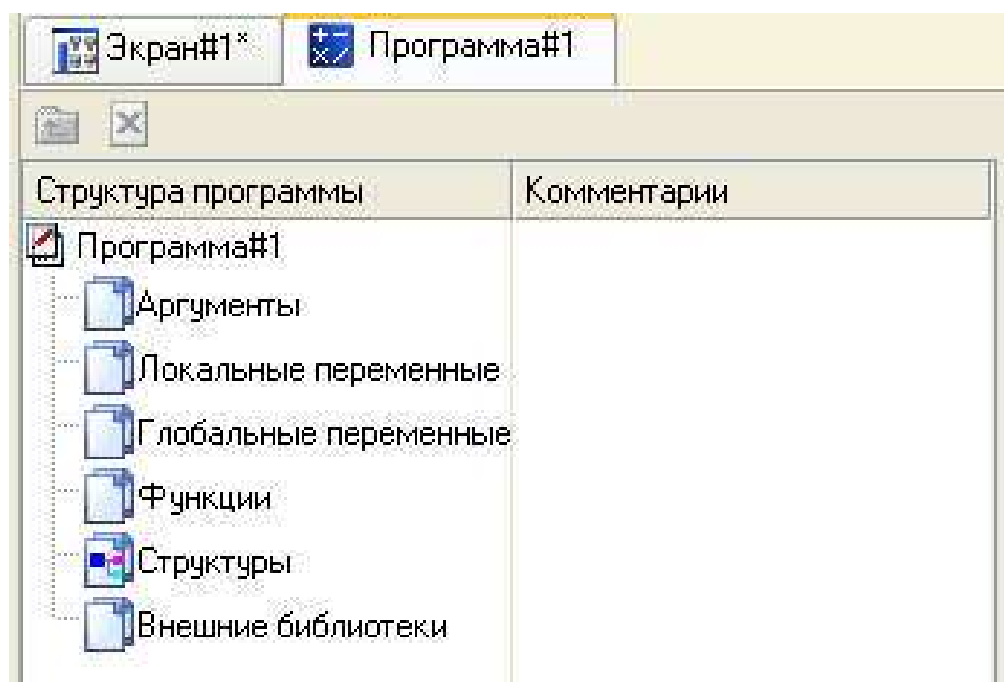


Рис. 50. Редактирование компонента «Программа#1»

- выделением ЛК в дереве шаблона «Программа#1» строки «Аргументы» вызовем табличный редактор аргументов;
- с помощью иконки  создадим в редакторе аргументов три аргумента с именами «Параметр», «Управление» и «Сумма». При этом первые два аргумента должны быть типа «IN», а третий – «OUT»;

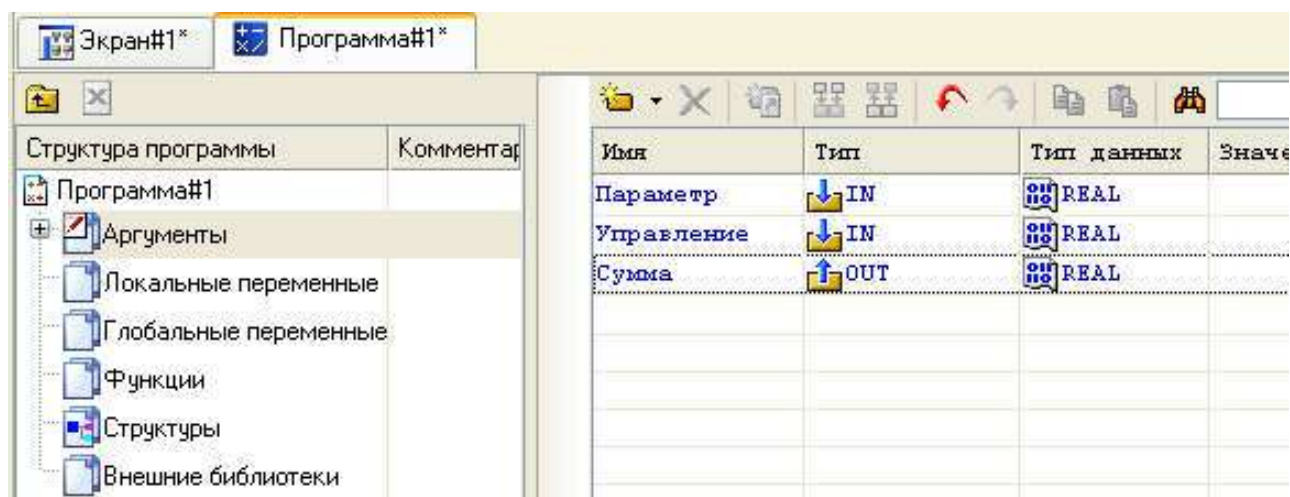


Рис. 51. Редактирование аргумента

- выделим ЛК в дереве шаблона строку «Программа#1» и в открывшемся диалоге «Выбор языка» выберем язык «ST»:

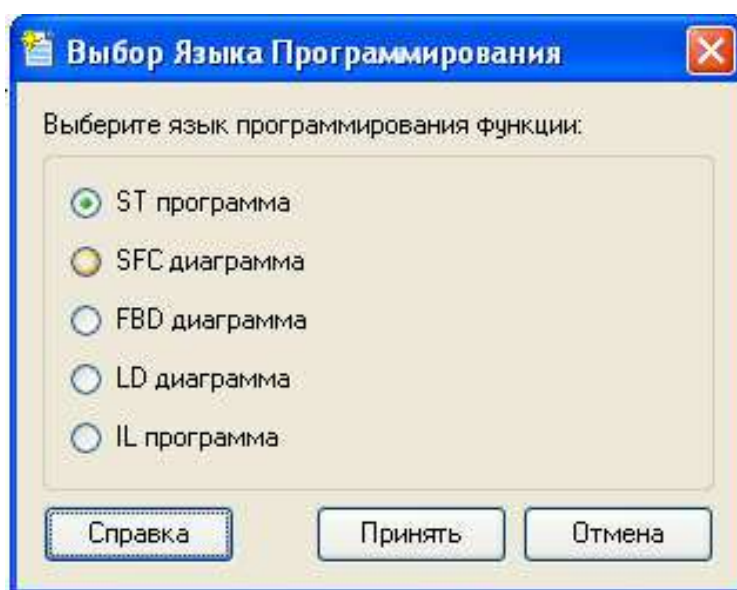


Рис. 52. Выбор языка программирования

- по нажатию экранной кнопки «Принять» в открывшемся окне редактора программ с объявленными переменными наберем следующую строку:



Рис. 53. Редактор программы

- с помощью иконки  на инструментальной панели редактора или нажатием горячей клавиши F7 скомпилируем программу и убедимся в успешной компиляции в окне «Вывод» (Output), вызываемом из инструментальной панели с помощью иконки .

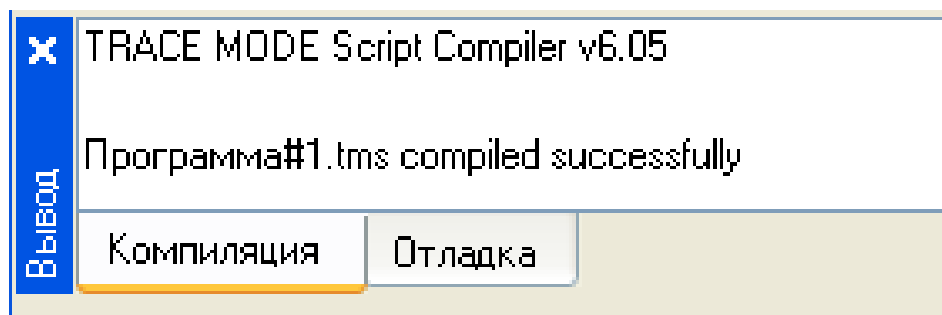


Рис. 54. Компиляция программы

Привязка аргументов программы

Выполним привязку аргументов программы к атрибутам каналов:

- вызовем свойства компонента «**Программа#1**» через контекстное меню;
- выберем вкладку «**Аргументы**»;
- двойным нажатием в поле «**Привязка**» свяжем аргументы программы с атрибутами каналов – аргумент «**Параметр**» к реальному значению канала «**Параметр**», аргумент «**Управление**» к реальному значению канала «**Управление**»;

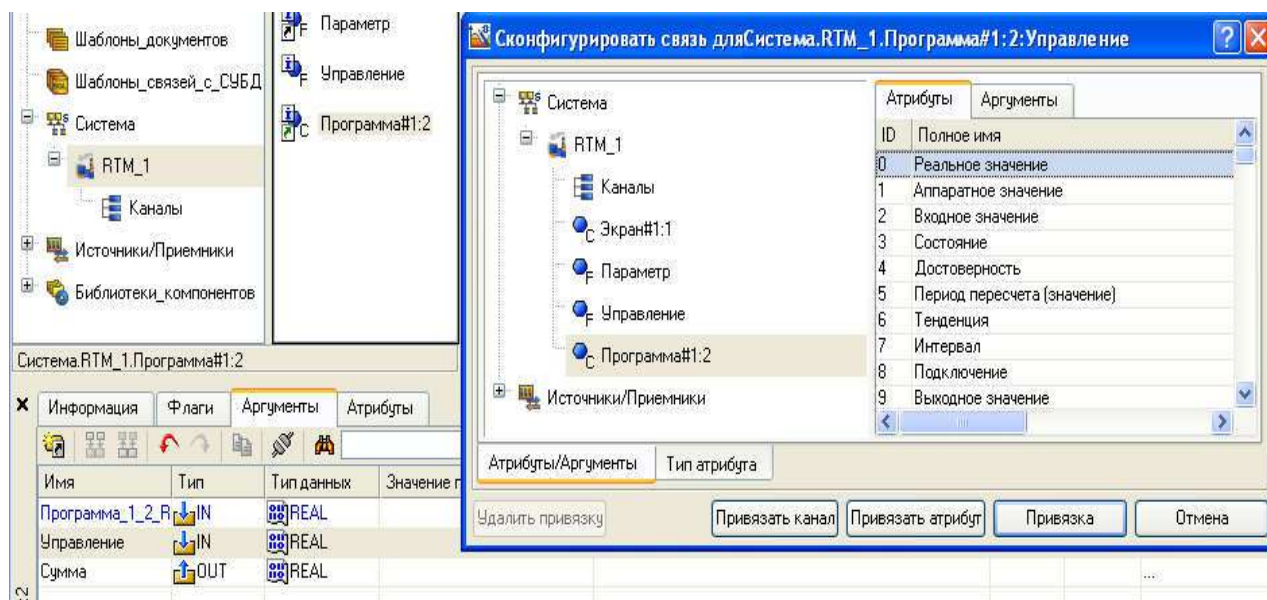


Рис. 55. Редактирование привязки

- двойным щелчком в поле «**Привязка**» аргумента программы «**Сумма**» вызовем окно настройки связи, выберем в левом окне канал класса «**CALL Экран#1**», а в правом откроем вкладку «**Аргументы**» и укажем в

ней аргумент «Сумма», затем щелчком ЛК по экранной кнопке «Привязка» подтвердим связь:

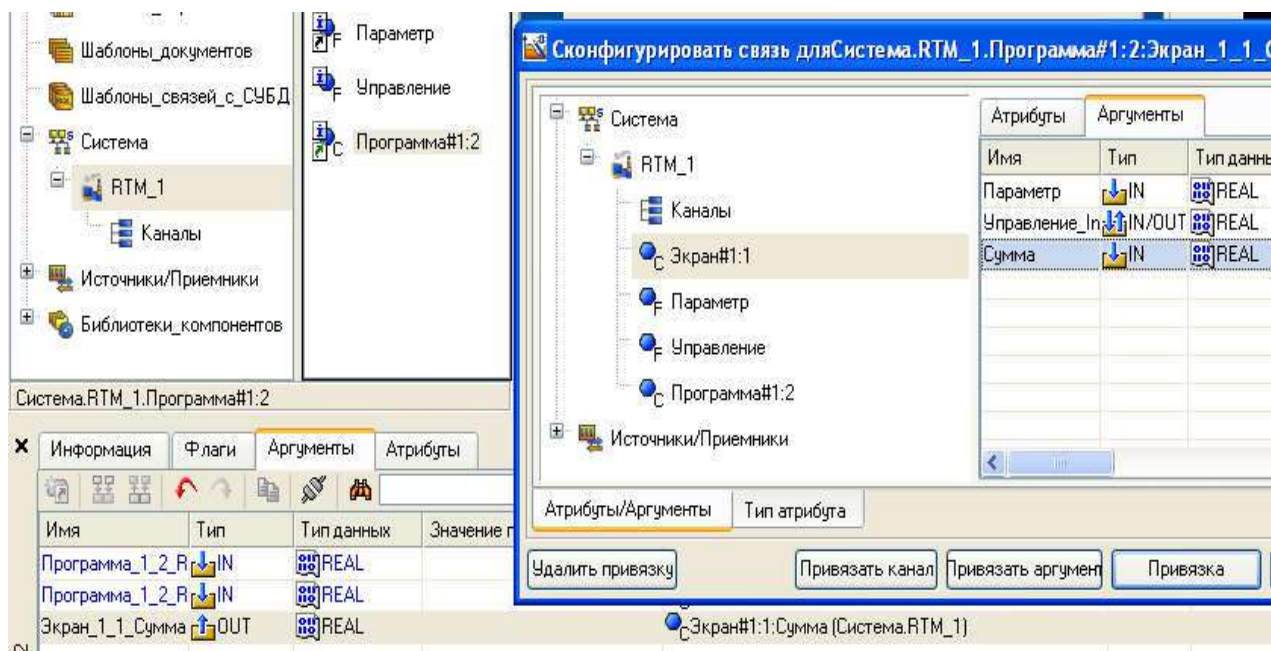


Рис. 56. Настройка связи

В результате будем иметь:

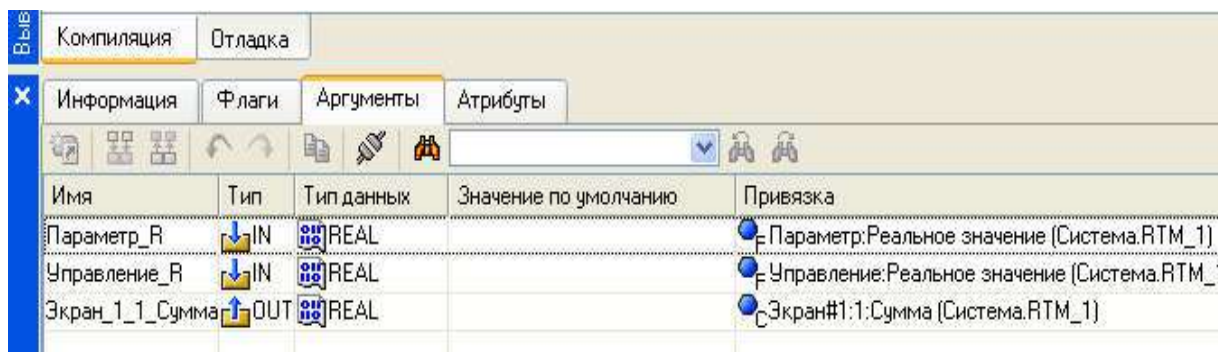


Рис. 57. Результат редактирования

- Закроем окно свойств компонента «Программа#1».

Запуск проекта

- Сохраним проект с помощью иконки ;
- на инструментальной панели выберем ЛК иконку  для запуска проекта в реальном времени;

• с помощью иконки  на инструментальной панели запустим режим исполнения.

Вводя теперь с помощью кнопки «**Управление**» значения параметров будем наблюдать соответствующее изменение реального значения канала «**Управление**» и результата работы программы суммирования:

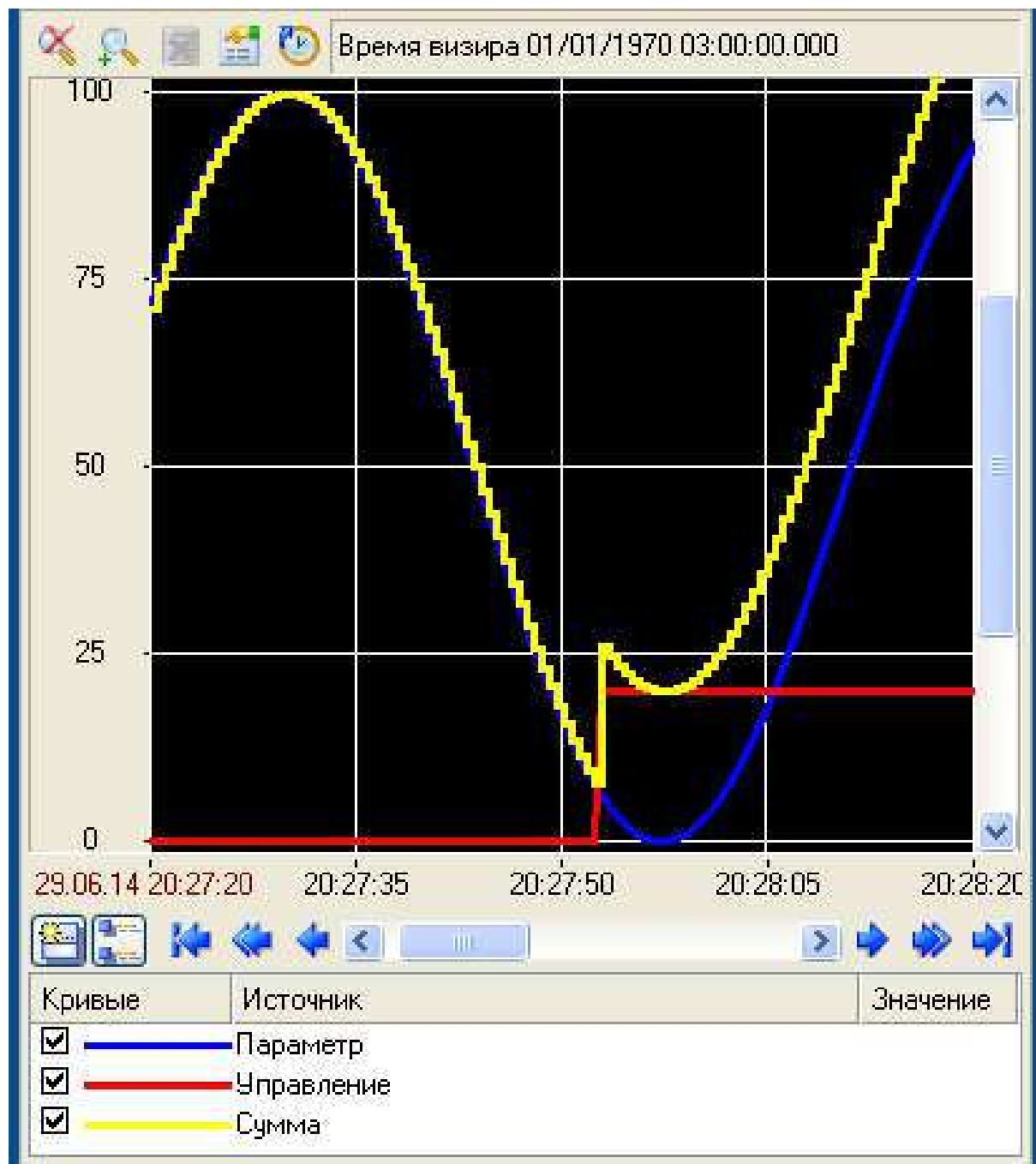


Рис. 58. Запуск проекта

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Мезенцев А. А., Павлов В. М. САПР TRACE MODE 6: учебно-методическое пособие / Томский политехнический университет. – Томск: 2012 – 137 с.
2. Trace Mode IDE 6.09 Base: Справка. -М.: AdAstra Research Group, Ltd., 2014.
3. Сайт TRACE MODE . Режим доступа: www.tracemode.ru.
4. Пьявченко Т. А. Автоматизированные информационно-управляющие системы с применением SCADA-системы Trace Mode: учебное пособие для студентов вузов, обучающихся по направлению подготовки «Автоматизация технологических процессов и производств» / – Санкт-Петербург: Лань, 2015. - 335 с.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ часть 2
к практическим занятиям
по дисциплине «Интеллектуальные
информационноизмерительные и управляющие системы » для
студентов направления подготовки
11.04.04Электроника и нанoeлектроника
Направленность (профиль)
Интеллектуальные программируемые системы и устройства

Учебно-методическое пособие для
выполнения практических работ

.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего
образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания к курсовому проектированию

ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ И УПРАВЛЯЮЩИЕ
СИСТЕМЫ

Направление подготовки 11.04.04 Электроника и наноэлектроника

Направленность (профиль) Интеллектуальные программируемые системы и
устройства.

Год начала обучения 2026

Реализуется в 3 семестре

Методические указания к курсовому проектированию по дисциплине «Интеллектуальные информационно-измерительные и управляющие системы» предназначены для обучающихся 11.04.04 Электроника и нанoeлектроника Направленность (профиль) Интеллектуальные программируемые системы и устройства.

Указания составлены на основе требований федерального государственного образовательного стандарта высшего образования и содержат: постановку цели и задач курсовой работы, структуру курсовой работы, а также процедуру ее проверки и защиты. В конце представлены рекомендуемые источники литературы, в том числе действующие версии норм и стандартов. По многим пунктам методических указаний даны примеры или ссылки, где с ними можно ознакомиться. Приведена ориентировочная тематика курсовых работ.

Содержание

Введение	5
1 Теоретические сведения из области автоматизированных систем управления технологическими процессами	7
2 Общие правила и рекомендации по выполнению курсовой работы	14
3 План выполнения курсовой работы	16
4 Структура и содержание курсовой работы.....	18
4.1 Введение	19
4.2 Основная часть курсовой работы.....	19
4.2.1 Обоснование необходимости разработки АСУ ТП	20
4.2.2 Характеристики и описание автоматизируемого технологического процесса (объекта).....	21
4.2.3 Анализ процесса автоматизации заданного объекта на основе обзора информационных источников	21
4.2.4 Техническое задание	21
4.2.5 Разработка локальной системы управления технологическим процессом (объектом)	22
4.2.6 Разработка общей структуры системы управления	24
4.2.7 Расчёт и выбор технической реализации регулятора	29
4.2.8 Выбор технической реализации остальных элементов системы управления	30
4.2.9 Разработка мнемосхемы системы управления в SCADA-системе	31
4.2.10 Разработка алгоритмов управления.....	36
4.2.11 Расчёт показателей качества системы	37
4.3 Заключение	38
4.4 Список использованных источников	38
4.5 Приложения.....	38
5 Утверждение и защита курсовой работы.....	39
6 Формулировка тематики курсовых работ.....	40
7 Источники информации по автоматизированным системам управления технологическими процессами	42
7.1 Основная литература	42
7.2 Дополнительная литература	44
7.3 Периодические издания	45

7.4 Интернет-ресурсы.....	46
Список использованных источников.....	48
Приложение А.....	49

Введение

Современные производственные системы являются интегрированными комплексами, охватывающими все уровни управления производством.

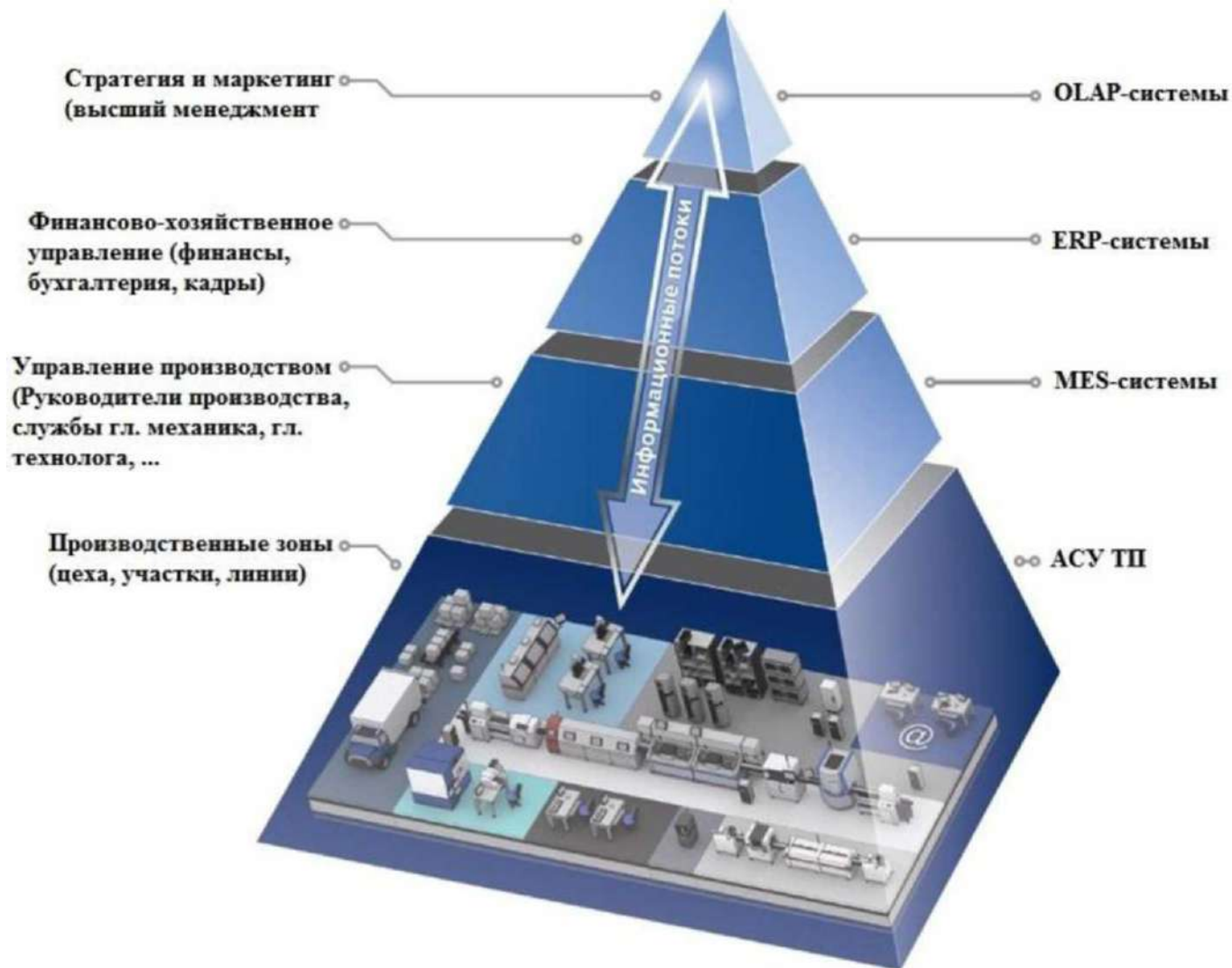


Рисунок 1 - Информационно-управляющая структура производственного

Структура современного производства представлена на рисунке 1. предприятия

Иногда между ERP (англ. *Enterprise Resource Planning*, планирование ресурсов предприятия) и MES (от англ. *manufacturing execution system*, система управления производственными процессами) помещают уровень APS (сокр. от англ. *Advanced Planning & Scheduling* - усовершенствованное планирование) - программное обеспечение для производственного планирования, а уровень OLAP

(англ. *online analytical processing*, интерактивная аналитическая обработка) может иметь другое название, например, BI (*Business intelligence* - бизнес-интеллект, корпоративный интеллект).

На нижнем уровне пирамиды управления находится АСУ ТП - система, предназначенная для выработки управляющих воздействий на технологический объект в соответствии с критерием оптимальности и учётом поступающей об объекте информации, а также действующих возмущений.

АСУ ТП, как видно из рисунка 1, служит основанием или базой для всех остальных уровней, поэтому изучение этой системы является важным звеном в обучении современного инженера.

В курсовой работе по дисциплине «Автоматизация технологических процессов и производств» обучающийся должен продемонстрировать умение самостоятельно решать задачи автоматизации технологических процессов, творческое мышление, навыки в использовании новейших технологий из области автоматики и автоматизации, а также применять полученные знания к конкретной проблеме.

Целью курсовой работы является закрепление теоретических и практических знаний по автоматизации технологических процессов и производств и формированию умений по проектированию систем автоматизации технологических процессов.

Для достижения этой цели необходимо решить следующие задачи:

- закрепить знания по терминологии из области автоматики и автоматизации технологических процессов;
- получить навыки выбора технических средств автоматизации;
- получить навыки по разработке функциональных схем автоматизации;
- научиться составлять техническое задание на проект по автоматизации агрегата, участка.

В результате выполнения курсовой работы формируется компетенция:

- ПК-2 способностью проводить вычислительные эксперименты с использованием стандартных программных средств с целью получения математических моделей процессов и объектов автоматизации и управления.

1 Теоретические сведения из области автоматизированных систем управления технологическими процессами

Современные производственные предприятия представляют собой сложные многоиерархичные системы управления, которые сопровождаются интенсивными информационными потоками и находятся в условиях переменчивой неопределенной внешней среды. Все это приводит к усложнению и повышению требований к управляющим системам всех уровней, особенно уровня реализации технологического процесса, представленного автоматизированными системами управления технологическими процессами (АСУ ТП).

Автоматизированная система управления технологическим процессом - *человеко-машинная система управления, обеспечивающая автоматизированный сбор и обработку информации, необходимой для оптимизации управления технологическим объектом в соответствии с принятым критерием.*

Современный этап развития АСУ ТП характеризуется внедрением систем на основе единых технических средств под управлением программируемых логических контроллеров (ПЛК) и информационно-программной поддержкой диспетчерских программных комплексов, называемых SCADA-системами.

SCADA (аббр. от англ. **Supervisory Control And Data Acquisition** - диспетчерское управление и сбор данных) - *программный пакет, предназначенный для разработки или обеспечения работы в реальном времени систем сбора, обработки, отображения и архивирования информации об объекте мониторинга или управления.* При наличии средств программирования логических контроллеров такие системы приобретают способность управлять режимами технологического процесса и к ним прибавляется термин SoftLogic.

Типичная структура современных АСУ ТП на основе ПЛК и SCADA-систем показана на рисунке 2.

Цель АСУ ТП определяется, исходя из общей цели управления предприятием, назначения системы, типа технологического объекта, для которого она разработана.

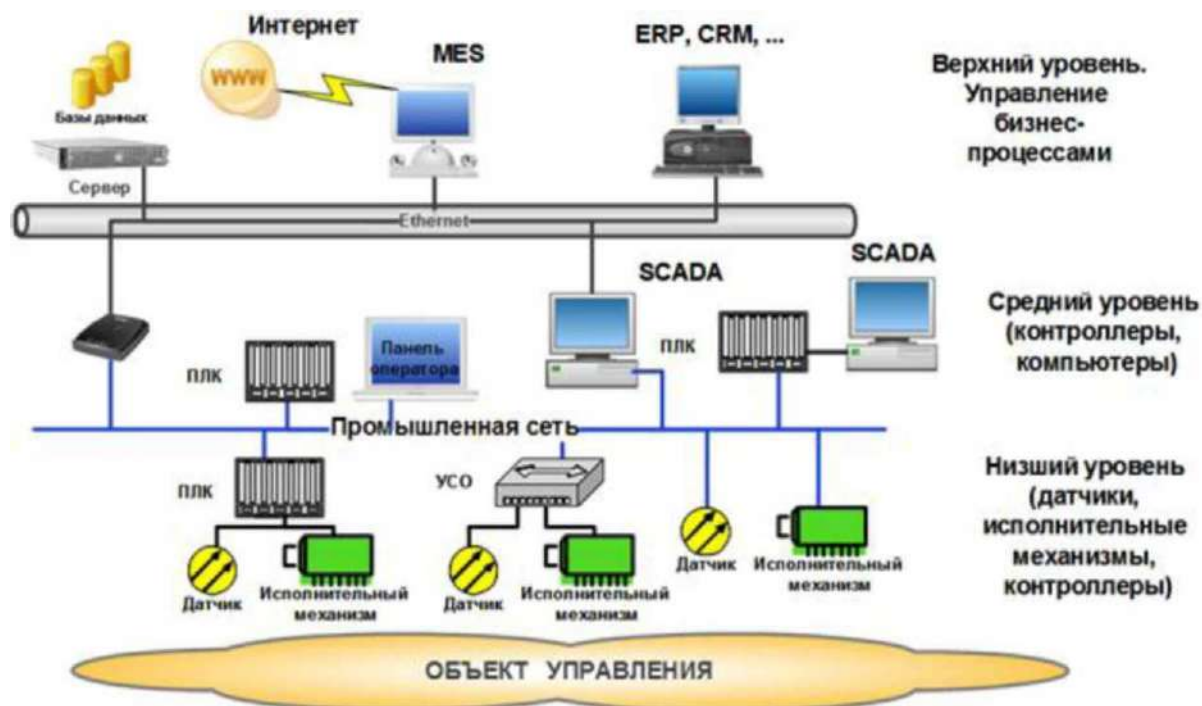


Рисунок 2 - Типичная структура АСУ ТП с ПЛК и SCADA-системой

Технологический объект управления (ТОУ) - это совокупность технологического оборудования и реализованного на нем по соответствующим регламентам технологического процесса.

АСУТП предназначены для выработки и реализации управляющих воздействий на ТОУ в соответствии с принятым критерием управления и с помощью современных средств ЭВМ.

Критерий управления - это соотношение, характеризующее качество работы ТОУ в целом, выраженный в числовой форме и принимающий конкретные фиксированные значения.

Одними из распространенных целей создания АСУ ТП могут быть:

- увеличение прибыли предприятия;
- повышение качества продукции;
- экономия электроэнергии и других производственных ресурсов;
- улучшение технологии производства продукта;
- повышение эффективности использования производственных ресурсов;
- улучшение тяжелых условий труда;

- обеспечение безопасности ТОУ;
- повышение экологичности производства;
- оптимизация режимов функционирования ТОУ.



Рисунок 3 - Классы функций АСУ ТП

Классификация функций АСУ ТП показана на рисунке 3.

К **основным управляющим функциям АСУ ТП** относятся выработка и реализация рациональных управляющих воздействий на основе имеющейся информации для эффективного выполнения технологического процесса.

Информационная функция АСУ ТП состоит в обеспечении управляющей функции соответствующей информацией: состоянием технологического процесса через значение его параметров, косвенными вычисляемыми показателями, оценку и прогноз состояния производственного комплекса, подготовка информации в выше-стоящие уровни управления, формирование сведений в надлежащей форме для оперативного персонала и тому подобное.

Содержание **вспомогательных функций** определяется внутренними потребностями самой АСУ ТП в целях нормального функционирования системы,

например, ремонт оборудования, утилизация отходов, обеспечение материалами и тому подобное.

Как и любая автоматизированная система АСУ ТП включает несколько видов



Рисунок 4 - Виды обеспечения АСУ ТП

обеспечения (рисунок 4).

Математическое обеспечение АСУ ТП представляет собой эксплуатационную документацию, содержащую описание методов и алгоритмов, разработанных при создании данной АСУ ТП и обеспечивающих решение технологических задач [1].

Техническое обеспечение АСУ ТП включает аппаратные и технические средства, обеспечивающие хранение, обработку, представление и передачу информации по системе и из системы на вышестоящие уровни, а также устройства и оборудование, реализующие технологический процесс.

Программное обеспечение АСУ ТП - совокупность программ, обеспечивающих функционирование всех цифровых вычислительных средств АСУ ТП (контроллеры, серверы, рабочие и инженерные станции, программаторы, панели оператора), а также решающих все функциональные задачи на этапах разработки, наладки, тестирования и эксплуатации системы.

Информационное обеспечение АСУ ТП - совокупность входных и выходных сигналов, а также массивов информации, формируемых в АСУ ТП, характеризующих состояние технологического оборудования и эксплуатационной документации, используемых при создании и функционировании АСУ ТП.

Организационное обеспечение АСУ ТП - совокупность документов, устанавливающих порядок и правила функционирования оперативного персонала АСУ ТП, а также организационные мероприятия, направленные на успешное внедрение системы и на безопасное ведение технологического процесса.

Метрологическое обеспечение АСУ ТП - совокупность технических средств, требований, положений, правил, норм и методик, направленных на достижение единства и требуемой точности измерений в составе АСУ ТП.

Лингвистическое обеспечение АСУ ТП - языки программирования, описания и манипулирования данными, описания алгоритмов управления АСУ ТП.

Методическое обеспечение АСУ ТП - документы, в которых содержится состав АСУ ТП, правила эксплуатации ее компонентов, последовательные операции для решения задач управления и инструкции по обслуживанию оборудования.

Алгоритмическое обеспечение АСУ ТП - совокупность алгоритмов функционирования подсистем АСУ ТП, которая реализуется в виде программ. Например, алгоритмы первичной обработки информации (ПОИ), стабилизации, оптимизации, программно-логического управления и так далее.

Правовое обеспечение АСУ ТП - представляет собой совокупность норм (выраженную в нормативных актах), которые устанавливают и закрепляют организацию этих систем, их цели, задачи, структуру и функцию, правовой статус АСУ ТП и всех её звеньев, и регламентируют процессы создания и функционирования

АСУ ТП.

Классификация АСУ ТП показана на рисунке 5.

Если все функции АСУ ТП выполняются без участия человека, то такие АСУ ТП называются **автоматическими**, а когда часть функций остается закрепленной за человеком - **автоматизированными**.

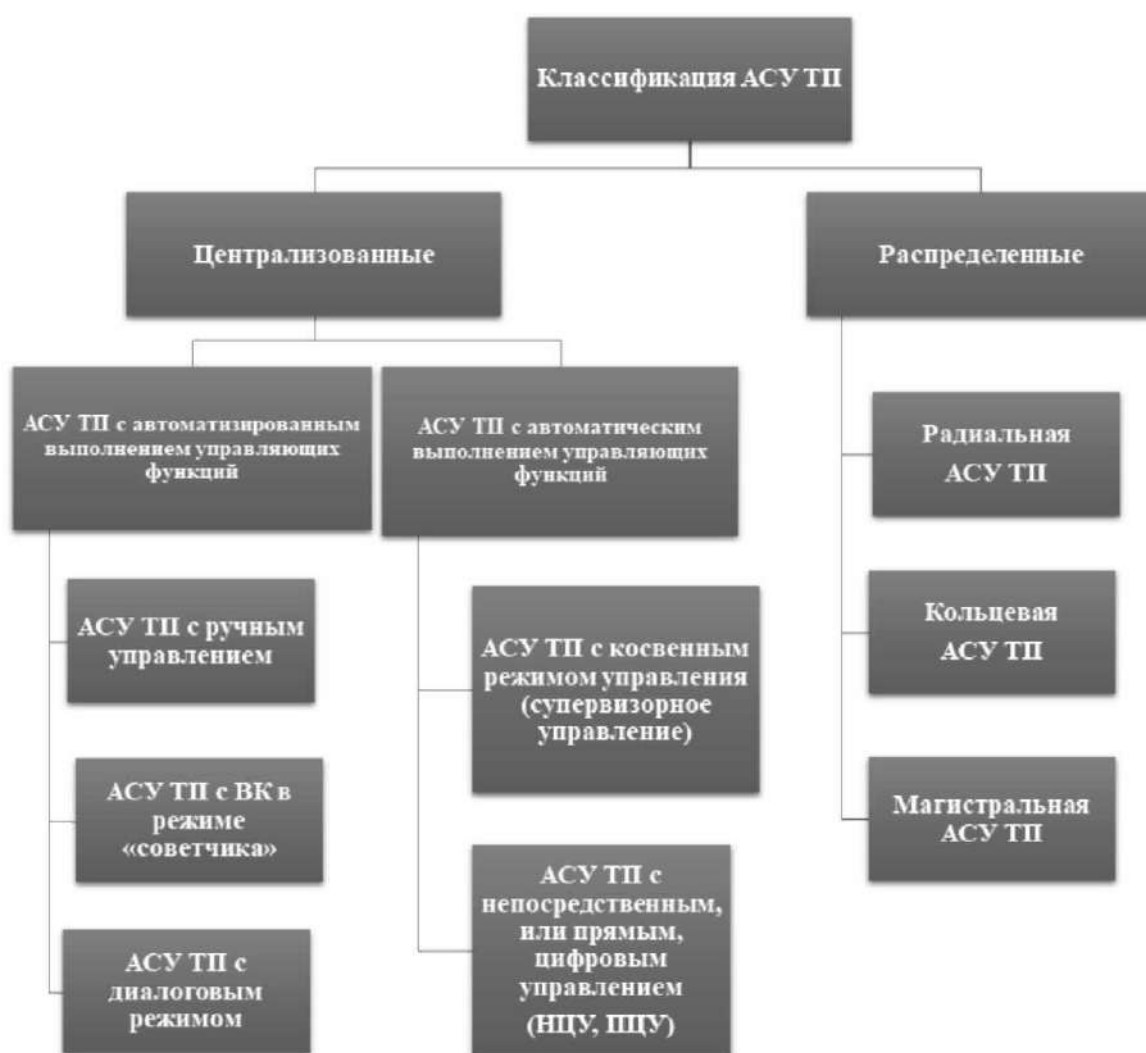


Рисунок 5 - Классификация АСУ ТП

Автоматические АСУ ТП подразделяются на системы с **супервизорным управлением** и **прямым цифровым управлением** [2].

Супервизорная АСУ ТП обеспечивает стабилизацию параметров технологического процесса, заданных уставками, на оптимальном уровне путем выработки управляющих воздействий в реальном времени. Оператор в этом случае

может отслеживать протекание технологического процесса через устройства отображения информации, но не вмешиваться в него или реагировать в случае чрезвычайных ситуаций.

При прямом цифровом управлении управляющее вычислительное устройство само формирует управляющие воздействия для коррекции параметров технологического процесса, взаимодействуя с полевым уровнем. Задача оператора, как и в предыдущем случае - это вмешательство по мере аварийных или других внештатных ситуаций.

Автоматизированные АСУ ТП существуют в трёх разновидностях - **АСУ ТП с ручным управлением, АСУ ТП с вычислительной системой в советующем режиме и АСУ ТП в диалоговом режиме.**

В АСУ ТП с ручным управлением основные функции выработки управляющих воздействий остаются за человеком, а вычислительный комплекс только предоставляет необходимую для принятия решения информацию.

АСУ ТП с вычислительным комплексом в советующем режиме оставляет за человеком-оператором выработку управляющих воздействий, но информационно-вычислительный комплекс может вырабатывать рекомендации по оптимальному решению для оператора.

При диалоговом режиме АСУ ТП оператор может корректировать процесс формирования информационно-вычислительным комплексом рекомендаций по управляющим воздействиям

2 Общие правила и рекомендации по выполнению курсовой работы

Учебный план направления подготовки 27.03.04 Управление в технических системах включает выполнение курсовой работы по дисциплине «Автоматизация технологических процессов и производств» в шестом семестре.

Курсовая работа представляет собой самостоятельную работу студента под руководством преподавателя, выполняемую с определенной целью путем постановки и решения задач.

Задачи, предусматриваемые в данной курсовой работе, включают освоение работ на стадиях технического и рабочего проектирования, а также получение навыков выбора и расчёта технических средств автоматизации.

Приступая к курсовой работе, студенту прежде всего необходимо очень тщательно и детально изучить технологический процесс (объект), провести его полный анализ, который выявит достоинства и недостатки существующей формы организации технологического процесса, используемых в нем технических средств. Следующим этапом будет анализ информационных источников: учебников, пособий, справочников, периодической литературы, технических норм и ГОСТов, патентов, интернет-источников и так далее, который заканчивается обобщением и формулировкой предложений по автоматизации технологического процесса.

Работа над курсовой по дисциплине «Автоматизация технологических процессов и производств» является частью подготовки студента к выполнению выпускной квалификационной работы. Помимо этого, курсовая работа развивает общие навыки работы с литературой и способности к самостоятельному анализу и обобщению информации.

Рекомендуется для реализации курсовой работы использовать только проверенные и общеизвестные программные средства, которые широко используются на практике при проектировании автоматизированных систем.

Графическая часть курсовой работы содержит рисунки, схемы, графики, диаграммы, таблицы, которые должны выполняться в соответствии с документом [3].

Объём курсовой работы составляет 25-30 страниц на листах формата А4 без учёта приложений. Рекомендуется количество листов свыше обозначенного объёма переносить в приложения.

Темы курсовой работы разрабатываются и утверждаются на обучающей кафедре, но могут включать и сторонние предложения студентов, если будут представлены веские основания целесообразности разработки или студент получает задание от предприятия, на котором проходит практику.

Работа над курсовой начинается с выдачи задания. При выполнении работы предусмотрены индивидуальные консультации между студентом и руководителем курсовой работы для уточнения или помощи при возникающих трудностях.

За выполнение и защиту курсовой работы студент получает оценку. Лучшие работы могут быть рекомендованы к изданию в сборниках научных студенческих работ или в журналах, а также к применению их результатов в организациях, к которым данная работа имеет отношение.

Тематика курсовой работы выбирается либо из предложенных преподавателем тем, либо дается, исходя из производственного задания, либо предлагается самостоятельно и согласуется с руководителем. Ориентировочный шаблон тематик курсовой работы представлен в разделе № 6. На выбор темы студенту отводится недельный срок, после которого преподаватель имеет право утвердить за студентом тему по своему усмотрению.

3 План выполнения курсовой работы

Предложенные в пункте 6 тематики курсовых работ представляют собой примерные направления разработки и не могут рассматриваться, как прямое руководство. Темы работы в пределах одной группы текущего года уникальны для каждого студента, повторения исключаются.

При анализе технологического процесса могут возникнуть две ситуации:

- на анализируемом объекте уже присутствует техническое оснащение автоматизации, и задача курсовой работы заключается в его модернизации. Название темы в этом случае содержит слово «модернизация», после которого идет полное название АСУ ТП на объекте с привязкой к технологическому процессу, который эта система автоматизирует;

- объект исследования еще не имеет своей АСУ ТП или ее подсистем, и задача студента заключается в проектировании одной из функциональных частей (модуля) системы. Наименование темы в этом случае начинается со слов «автоматизация процесса» или «проектирование системы», а далее следует полное название АСУ ТП и технологического процесса, к которому она имеет отношение.

Содержание или план курсовой работы разрабатывается обучающимся в самостоятельном порядке сразу после утверждения темы, составленный план передается на согласование руководителю, после чего начинается процесс выполнения курсовой работы.

Ориентировочное содержание курсовой работы и сроки выполнения этапов

Таблица 1 - План выполнения курсовой работы

№ этапа	Этапы	Сроки
1	Выбор темы и утверждение ее у руководителя	1-ая неделя
2	Разработка структуры и содержания курсовой работы и утверждение их у руководителя	2-3-я недели

представлены в таблице 1.

Продолжение таблицы 1

№ этапа	Этапы	Сроки
3	Анализ литературных источников	2 - 3 - я недели
4	Написание основной части курсовой работы	4 - 10-я недели
5	Корректировка основной части курсовой работы после проверки руководителя	11 - 14-я недели
6	Написание заключения и выводов по курсовой работе, оформление курсовой работы	13 - 14-я недели
7	Сдача готовой курсовой работы руководителю на проверку	15-я неделя
8	Защита курсовой работы	16-я неделя

4 Структура и содержание курсовой работы

Курсовая работа содержит пояснительную записку и графическую часть.

Шаблонная структура курсовой работы включает следующие разделы:

- титульный лист;
- задание на курсовую работу;
- содержание;
- аннотация;
- введение;
- основная часть:
- заключение;
- список использованных источников;
- приложения.

Графическая часть может включать следующие элементы:

- общий вид автоматизируемого технологического объекта с размещением средств автоматизации;
- схему алгоритмов управления объектом или листинги программ в SCADA-системе;

- функциональную схему автоматизации;
- вид настроек регулятора и результаты моделирования переходных процессов системы управления;
- мнемосхема системы управления, изображенная средствами SCADA-системы.

Пример оформления титульного листа приведён в приложении А.

Аннотация представляет собой краткое содержание работы по разделам, актуальность и перечисление объёма частей пояснительной записки (страниц), а также графической части.

4.1 Введение

Во введении должна быть аргументированно поставлена проблема, решаемая курсовой работой. Последовательность изложения материала введения в типовом виде может быть следующей:

- актуальность работы;
- краткое описание предметной области, выявление противоречий между состоянием теории и практики, завершаемое постановкой проблемы;
- современное состояние теории и практики по сформулированной проблеме;
- краткая характеристика и структура предприятия или технологического процесса;
- выведенные из проблемы предметной области цели и задачи работы;
- перечисление методов и средств, которые будут применяться для решения поставленных задач;
- актуальность работы.

Объём введения составляет 2 - 3 страницы.

4.2 Основная часть курсовой работы

Основная часть должна перекликаться с заданием на курсовую работу и выполняться строго в соответствии с его пунктами. Основная часть вместе с подпунк-

тами формулируется совместно с руководителем, но, как правило, типовой состав включает следующие пункты:

- обоснование необходимости разработки АСУ ТП;
- характеристики и описание автоматизируемого технологического процесса (объекта);
- анализ процесса автоматизации заданного объекта на основе обзора информационных источников;
- техническое задание (ТЗ);
- разработка локальной системы управления технологическим процессом (объектом);
- разработка общей структуры системы управления;
- расчёт и выбор технической реализации регулятора;
- выбор технической реализации остальных элементов системы управления;
- разработка мнемосхемы системы управления в SCADA-системе;
- разработка алгоритмов управления;
- расчёт показателей качества системы.

4.2.1 Обоснование необходимости разработки АСУ ТП

Раздел содержит характеристики объекта автоматизации, краткое описание его конструкции и принципа работы, приводятся основные технические параметры, оценивается уровень автоматизации.

Оформление необходимости разработки АСУ ТП разделяется на два аспекта: социально-экономический и потребности самого технического объекта. Социально-экономический аспект учитывает выполнение задачи в масштабах всего предприятия или организации, указывается каким образом решаемая проблема улучшает его деятельность и функционирование в целом. Последнее излагается путем указания источников повышения эффективности, которыми могут быть:

- повышение производительности труда;
- сокращение используемой рабочей силы;

- улучшение качества продукции;
- повышение надежности технических средств;
- повышение экологичности производства;
- социальный эффект: освобождение от монотонного неквалифицированного труда, освобождение от вредных и опасных условий труда, получение людьми дополнительных качеств и возможностей для выполнения специфических задач и тому подобное.

Второй аспект предполагает построение логической цепочки модернизации процесса (объекта) через устранение недостатков, возможность автоматизации или повышение ее уровня.

4.2.2 Характеристики и описание автоматизируемого технологического процесса (объекта)

Раздел посвящён описанию общей структуры и особенностям технологического процесса, подлежащего автоматизации, например, входящие в процесс технологические операции или различные варианты реализации технологического процесса. Здесь же приводится описание документации по процессу, регламентирующей требования к функциональному обеспечению АСУ ТП.

4.2.3 Анализ процесса автоматизации заданного объекта на основе обзора информационных источников

На данном этапе проводится обзор зарубежной и отечественной литературы, в том числе периодической и интернет-источников. Цель этого процесса состоит в поиске новых методов решения поставленной задачи и недостатков существующих систем. Если прямой поиск не даёт результатов, подобных разрабатываемой системе, то ведется поиск аналогов в других областях.

Результатом изучения информационных источников становится оформление технического задания на разработку системы.

4.2.4 Техническое задание

Техническое задание - это один из важнейших документов на создание системы. Без технического задания невозможно окончательное урегулирование отношений между заказчиком и разработчиком системы.

Техническое задание разрабатывается в соответствии с требованиями ГОСТ 34.602-89 [4], как окончательное оформление представления о системе, которую необходимо спроектировать.

В курсовой работе желательно уделить внимание следующим пунктам технического задания:

- назначение и цели создания (развития) системы;
- требования к системе;
- источники разработки.

В источниках разработки указываются все информационные материалы и технические аналоги или методики, на основании которых проводится разработка системы.

Все эти пункты описаны в приведенном ГОСТ 34.602-89 [4]. Полный состав технического задания выглядит следующим образом:

- 1) общие сведения;
- 2) назначение и цели создания (развития) системы;
- 3) характеристика объектов автоматизации;
- 4) требования к системе;
- 5) состав и содержание работ по созданию системы;
- 6) порядок контроля и приемки системы;
- 7) требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие;
- 8) требования к документированию;

9) источники разработки.

На объём технического задания в полном составе отводится не более 10 страниц, а технического задания с содержанием только основных пунктов - не более 3 страниц. Все пункты технического задания должны быть отражены в работе.

4.2.5 Разработка локальной системы управления технологическим процессом (объектом)

Перед началом проектирования системы управления технологическим процессом (объектом) по составленному техническому заданию стоит определиться, как именно система будет взаимодействовать с объектом или его частями и какую именно роль будет играть в общей структуре технологического комплекса.

Первое представление о проектируемой системе и ее коммуникациях с внешней средой начинают с составления структурной схемы системы. Разработка структурной схемы может быть разбита по двум направлениям:

- внутренние взаимосвязи системы;
- внешние взаимосвязи системы.

Внутренние взаимосвязи определяют взаимодействие между объектом и системой управления, которое можно пояснить следующими пунктами:

- выходные величины процесса (объекта), которые отражают собой его количественное и качественное состояние;
- параметры, для управления которыми проектируется система;
- параметры, которые необходимо контролировать для получения информации, необходимой для выработки управляющих воздействий.

Далее выстраивается структура системы путем изучения связей между элементами. Например, рассматривается иерархия соподчинения частей системы. Фактически система может состоять как из одного устройства, так и из нескольких вплоть до масштабов отдельных участков или цехов. Пример типовой структуры АСУ ТП рассмотрен на рисунке 2. В нее могут входить датчики и исполнительные

механизмы, образующие полевой уровень, затем идет уровень управления контроллеров, подчиняющийся уровню АРМов операторов. Если одного контроллера недостаточно, то может вводиться уровень с модулями расширения микропроцессорных функций, либо еще один нижний уровень контроллеров.

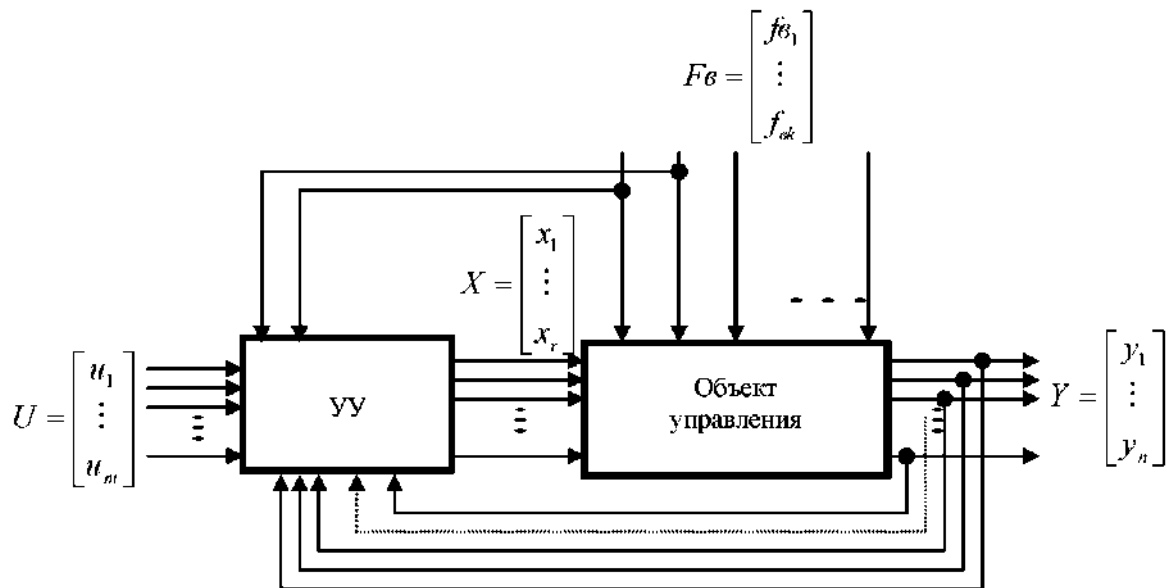
В самом общем виде структура любой автоматизированной системы может быть представлена в следующем виде (рисунок 6) [5].

На рисунке 6 вектор U - это уставки, заданные значения параметров управления, вектор X - внутренние параметры самого объекта (переменные состояния), B_v - вектор возмущающих внешних воздействий, Y - вектор текущих значений выходных параметров объекта.

Внешние взаимосвязи системы могут быть представлены через иерархию управления и положение проектируемой системы в ней. Стоит учитывать и характер процесса управления. Система может быть либо централизованно управляемой (в этом случае она представляет собой локальную систему управления), либо управление может быть распределённым, и в этом случае уровень иерархии управления усложняется, а в систему добавляются дополнительные устройства коммуникации, интерфейсы взаимодействия между модулями и устройства синхронизации и координации.

4.2.6 Разработка общей структуры системы управления

На текущем этапе развития производства разработка локальных систем встречается редко, поэтому чаще всего студенту приходится иметь дело с многоуровневыми системами, но выделение локальных подсистем облегчает процесс проектирования.



Основой обеспечения верхнего уровня управления являются программные комплексы, называемые SCADA-системами. При небольшой структурной сложности системы управления можно использовать и специализированное программное обеспечение, разработанное для контроллеров отдельных фирм, которое частично выполняет те же функции. Например, среда разработки CoDeSys или программное обеспечение OWEN Logic компании Owen.

Общая структура системы управления может выстраиваться через выделение контролируемых величин или контуров управления. Рекомендуется учитывать и используемые виды управления: стабилизация, слежение или программное управление.

Графическим оформлением и завершением данного этапа служит функциональная схема автоматизации.

Функциональная схема автоматизации (ФСА) представляет собой чертёж, на котором схематически условными обозначениями изображаются: технологическое оборудование, коммуникации, органы управления и средства автоматизации (приборы, регуляторы и так далее) с указанием связей между технологическим оборудованием и элементами системы управления, а также связей между отдельными элементами системы.

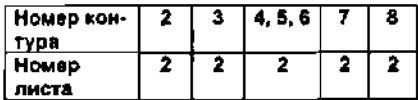
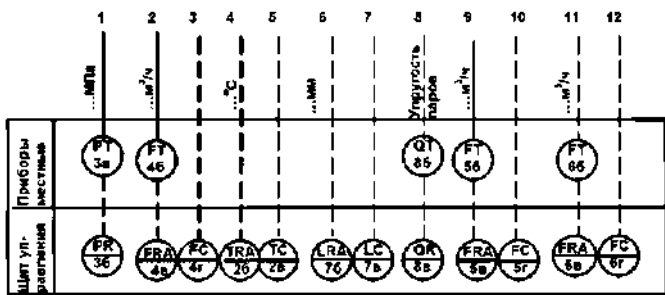
При построении функциональных схем оборудование и установки изображаются в упрощённом виде, но достаточно полно, чтобы давать ясное представление о принципе работы системы и взаимодействии всех средств автоматизации между собой.

Используются два способа иллюстрации ФСА:

- развернутый, при котором на схеме изображают состав и место расположения технических средств автоматизации каждого контура контроля и управления (рисунок 7);

- упрощенный, при котором на схеме изображают основные функции контуров контроля и управления (без выделения входящих в них отдельных технических средств автоматизации и указания места расположения) (рисунок 8).

При развёрнутом способе ФСА делится на 2 области: верхнюю с изображением технологического оборудования и тех средств автоматизации, которые располагаются непосредственно на оборудовании или рядом с ним, и нижнюю, где демонстрируется таблица с остальными средствами автоматизации, причём в первой строке идут приборы, конструктивно не связанные с оборудованием (приборы местные), а в остальных могут располагаться щиты управления (например, щит контроллеров или ПАЗ, приборы на щите) или подключения к SCADA-системе (рисунок 9).



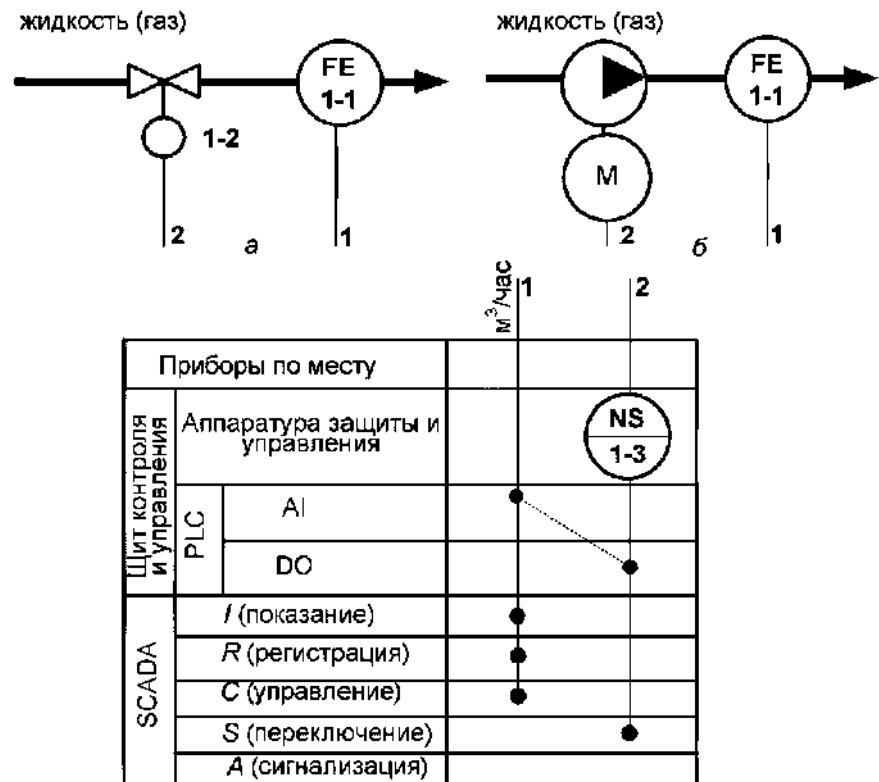


Рисунок 9 - Пример выполнения схемы автоматизации с подключением к системе SCADA

Конструктивно в изображении оборудования можно выделить следующие элементы:

- технологические коммуникации (изображаются по [6], [7]);
- средства автоматизации (приборы, исполнительные и измерительные устройства и тому подобное) (изображаются по ГОСТ 21.208-2013 [8]).

В упрощённой схеме ФСА отсутствуют пульты и щиты управления, изображается только верхняя область с оборудованием, а средства автоматизации располагаются там же рядом с датчиками.

Развёрнутый способ является более предпочтительным по причине большей полноты, детализации и наглядности.

Технологическое оборудование допускается не изображать на схеме в случаях, когда точки контроля и управления в технологических цехах немногочисленны (например, в рабочей документации по диспетчеризации). В этом случае в верхней части схемы вместо изображения технологического оборудования

приводят таблицу по рисунку 10, в графах которой указывают наименование оборудования и коммуникаций.

При разработке функциональной схемы автоматизации рекомендуется учитывать следующие соображения:

- АСУ ТП должна строиться по возможности на базе серийно выпускаемых средств автоматизации и вычислительной техники;
- желательно, чтобы выбираемые в качестве средств автоматизации приборы и оборудование принадлежали к Государственной системе промышленных приборов (ГСПП);
- классы точности аппаратуры диктуются требованиями технологического процесса, подлежащего автоматизации;
- при выборе технических средств следует учитывать их совместимость с уже имеющимися на предприятии оборудованием и устройствами;
- выбор АСУ ТП централизованного типа на основе управления от программируемых логических контроллеров и ЭВМ или распределенной АСУ ТП с применением контроллерной сети позволяет упростить отображение информации о ходе технологического процесса, если это диктуется экономической целесообразностью

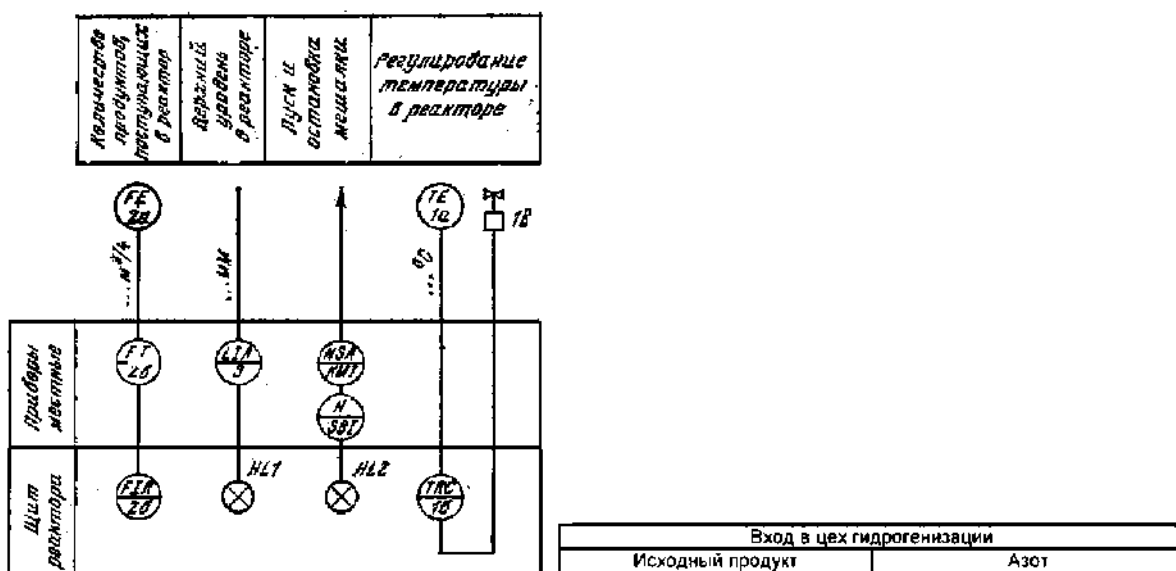


Рисунок 10 - Пример выполнения схемы автоматизации без изображения технологического оборудования
технического решения данного случая.

4.2.7 Расчёт и выбор технической реализации регулятора

В процессе проектирования локальных замкнутых систем автоматического управления (САУ) выбирают структуру и характеристики управляющего устройства, обеспечивающие требуемые параметры качества и устойчивость системы.

САУ может функционировать в статике или динамике. В случае, если интервал между двумя моментами появления возмущающего воздействия больше времени длительности переходного процесса в объекте, режим считается статическим. При сопоставимости времени переходного процесса и времени между последовательными возмущающими воздействиями принимается динамический режим.

Если на объект действуют интенсивные возмущения, которые не подлежат измерению, а сам объект имеет управляемость по выбранному выходу, то для управления этим объектом выбирается САУ замкнутого типа с регулированием по отклонению. Проектирование такой САУ заключается в выборе типового регулятора и расчёте его настроечных параметров для обеспечения уставок по качеству и устойчивости.

При невозможности достижения заданных показателей качества и устойчивости с помощью типового регулятора используют многоконтурное регулирование.

Если объект обладает низкой инерционностью и к нему не предъявляются требования высокой точности стабилизации выходной переменной, то возможно построение регулятора на основе двухпозиционного реле.

Предполагается использование и других типов регулирования: по возмущению, комбинированного, адаптивного, экстремального и так далее.

Перед выбором регулятора необходимо иметь следующие данные:

- динамические параметры процесса (объекта) (передаточный коэффициент, постоянные времени элементов, время запаздывания);
- заданные показатели качества управления (длительность переходного процесса, максимальное перерегулирование, установившееся значение процесса, число колебаний переходной характеристики и тому подобное).

На первом этапе лучше упрощать задачу, выбирая самый элементарный регулятор, и усложнять ее только по мере необходимости - в виду отсутствия удовлетворительного результата управления.

Для синтеза регулятора можно воспользоваться методами теории автоматического управления [9], используемыми для построения корректирующих цепочек.

Обобщение практических случаев автоматизации демонстрирует, что для большинства объектов выбор и расчёт регуляторов может быть выполнен методом последовательной коррекции с использованием типовых регуляторов. Метод заключается в последовательном включении в прямую цепь САУ корректирующих звеньев с передаточными функциями стандартных законов регулирования.

В качестве типовых регуляторов чаще всех используются пропорциональный (П-регулятор), интегрирующий (И-регулятор), пропорционально-интегрирующий (ПИ-регулятор), пропорционально-интегрально-дифференцирующий регулятор (ПИД-регулятор).

Основной стандартной настройкой, применимой для многих задач управления, является настройка на технический оптимум (оптимум по модулю). В тех случаях, когда требуется более высокая точность регулирования, применяют стандартную настройку на симметричный оптимум.

Техническая реализация регулятора в двух- и трехуровневых АСУ ТП обеспечивается программируемыми логическими контроллерами или командными аппаратами типа реле, контакторов или переключателей.

Рекомендуются широко распространённые программируемые логические контроллеры российских и зарубежных фирм: Овен, Segnetics, ICP DAS, Omron, Siemens, Schneider Electric, Mitsubishi Electric, VIPA.

Исходными данными для выбора программируемых контроллеров являются: число и параметры входов-выходов (род тока, номинальные напряжения и ток и др.); условия эксплуатации; конструктивное исполнение; функциональные возможности, интерфейсы связи и другие.

4.2.8 Выбор технической реализации остальных элементов системы управления

В данном подразделе выбираются такие модули системы, как датчики и исполнительные механизмы (неизменяемые элементы системы).

Выбор датчика обусловлен главным образом той физической величиной, которую он измеряет, а также диапазоном ее изменения, допустимой погрешностью, влиянием параметров окружающей среды, частотными характеристиками и тому подобным.

Ведущими фирмами по производству датчиков являются Метран, Овен, Georgin, Siemens, Omron, Pepperl+Fuchs, Scaime, Autonics Corporation, IFM и другие.

Исполнительное устройство выбирается, исходя из соответствия принципа действия и конструкции задачам автоматизации, а также по мощности, необходимой для воздействия на управляемый объект. Другим соображением по выбору исполнительного механизма может быть вид регулирующего органа, который, как правило, представляет собой его часть. Наиболее распространенными регулирующими органами являются клапаны, существенными факторами выбора которых служат:

- измеряемая среда (газ, пар, жидкость);
- температура среды;
- диаметр условного прохода клапана;
- пропускная способность;
- давление среды, действующее на запорную арматуру;
- тип привода.

Рекомендации по выбору исполнительных механизмов приведены, например, в [5].

4.2.9 Разработка мнемосхемы системы управления в SCADA-системе

Мнемосхема технологического процесса (объекта) - это наглядное графическое изображение функциональной схемы автоматизации объекта, включающее совокупность сигнальных устройств и сигнальных изображений оборудования, а также внутренних связей контролируемого объекта, размещаемых на диспетчерских пультах, операторских панелях или выполненных на персональном компьютере.

В процессе управления объектом мнемосхема служит для оператора важнейшим источником информации о его состоянии, включая аварийную сигнализацию и отражение протекающих процессов. К мнемосхеме предъявляются следующие требования:

- мнемосхема должна наглядно представлять схему системы в целом (на одном или нескольких переключающихся экранах) и связи между отдельными элементами объекта, а также достаточно детально отображать функциональные схемы агрегатов или устройств; демонстрировать характер взаимодействия данной системы с подсистемами и внешней средой; обеспечивать сигнализацию состояния важнейших технологических агрегатов; сигнализировать об авариях и сбоях в системе; обеспечивать быстрое нахождение резервов для локализации аварий;
- в мнемосхему должны включаться только те элементы, которые участвуют в процессе управления, не должно быть лишней и избыточной информации;
- при проектировании компоновки мнемосхемы должна учитываться последовательность действий оператора в процессе управления: оценка поступившей информации, принятие решения и выработка управляющего воздействия на выбранные объекты для нормализации режима работы, контроль за результатами реализованного управления;
- наиболее важные с точки зрения процесса управления элементы должны выделяться формой, цветом, размерами;
- мнемосхему следует разбивать по отдельным объектам, агрегатам или устройствам, входящим в систему, демонстрируя их особенности, отличные от других элементов;

- желательно, чтобы схема совпадала или отражала пространственное положение объектов для выработки и ускорения осмысления ее оператором;
- мнемосхема должна проектироваться с как можно большей унификацией. Одинаковые элементы следует обозначать на ней идентично, выделяя как можно больше таких частей;
- мнемонические символы должны быть единым алфавитом, желательно максимально коротким, с чёткой идентификацией и отличием символов друг от друга;
- желательно, чтобы мнемонический символ отражал функциональный признак объекта, а обозначения контуров не пересекались, чтобы не запутывать схему;
- сигнализирующие элементы, как правило, имеют зеленый свет для нормальной работы и красный для аварийного состояния, причём смена состояния должна сопровождаться прерывистым мигающим световым сигналом;
- соединительные линии на мнемосхеме должны быть прямыми и сплошными и иметь наименьшее число пересечений;
- при разработке мнемосхемы следует учитывать ее воздействие на психофизическое состояние оператора и не перегружать схему раздражающими оттенками (например, красным) или большим количеством мигающих элементов;
- мнемосхемы могут разбиваться на виды в зависимости от функций и взаимодействия с ними оператора (рисунок 11). В случае операторских мнемосхем технологический объект бывает сосредоточенным в едином пространстве, а в случае диспетчерских - распределённым по нескольким агрегатам, оборудованию, аппаратам и установкам или даже территориально. Что операторские, что диспетчерские мнемосхемы могут быть разделены далее в зависимости от того, как именно с ними работает оператор: контактно или только получает информацию о ходе процесса. Существуют и другие виды классификации, например, по способу кодирования информации (условные и символические).

Перечисленные требования заложены в так называемые принципы проектирования мнемосхем:

- *принцип лаконичности*, согласно которому мнемосхема должна быть простой, не должна содержать лишних, затемняющих элементов, а отображаемая информация должна быть четкой, конкретной и краткой, удобной для восприятия и дальнейшей переработки;



Рисунок 11 - Классификация мнемосхем

- *принцип обобщения и унификации* состоит в использовании наиболее существенных особенностей управляемых объектов, то есть на мнемосхеме не следует применять элементы, обозначающие несущественные конструктивные особенности системы, а символы сходных объектов и процессов необходимо по возможности объединять и унифицировать;

- *принцип акцента* - элементы контроля и управления на мнемосхемах, наиболее существенные для оценки состояния, принятия решения и воздействия на управляемый объект, следует выделять размерами, формой или цветом;

- *принцип автономности* предусматривает необходимость обособления друг от друга участков мнемосхемы, соответствующих автономно контролируемым и управляемым объектам и агрегатам. Эти обособленные участки должны быть четко отграничены от других и, согласно принципу структурности, должны иметь

завершенную, легко запоминающуюся и отличающуюся от других структуру. Структура должна отражать характер объекта и его основные свойства;

- *принципом пространственного соотнесения элементов контроля и управления* расположение контрольно-измерительных и индикаторных приборов -должно быть четко согласовано с расположением соответствующих им элементов управления, то есть должен соблюдаться закон совместности стимула и реакции;
- *принцип использования привычных ассоциаций* и стереотипов предполагает применение на мнемосхемах таких условных обозначений параметров, которые ассоциируют с общепринятыми буквенными обозначениями этих параметров. Желательно применять, если это возможно, вместо абстрактных знаков символы, ассоциирующиеся с объектами и процессами.

Пример мнемосхемы приведен на рисунке 1 2.

Мнемосхема в курсовой работе создается средствами SCADA Trace Mode, либо CoDeSys или Owen Logik. Пример разработки мнемосхемы в SCADA Trace Mode 6.06 можно найти по следующему пути, скачав бесплатную версию с сайта adastra.ru после быстрой регистрации: запустив файл `tmdevenv.exe`, заходим в опцию Справка — Содержание. Появляется окно, у которого слева находится дерево меню, показанное на рисунке 1 3.

Раскрывая меню, следуем в пункт Быстрый старт — Часть первая — Создание простейшего проекта — Создание графического экрана. Другой вариант разработки



Рисунок 1 2 - Пример мнемосхемы

мнемосхемы производственного участка можно посмотреть в этом же меню по следующему пути: Быстрый старт — Часть вторая — Создание экранов АРМ.

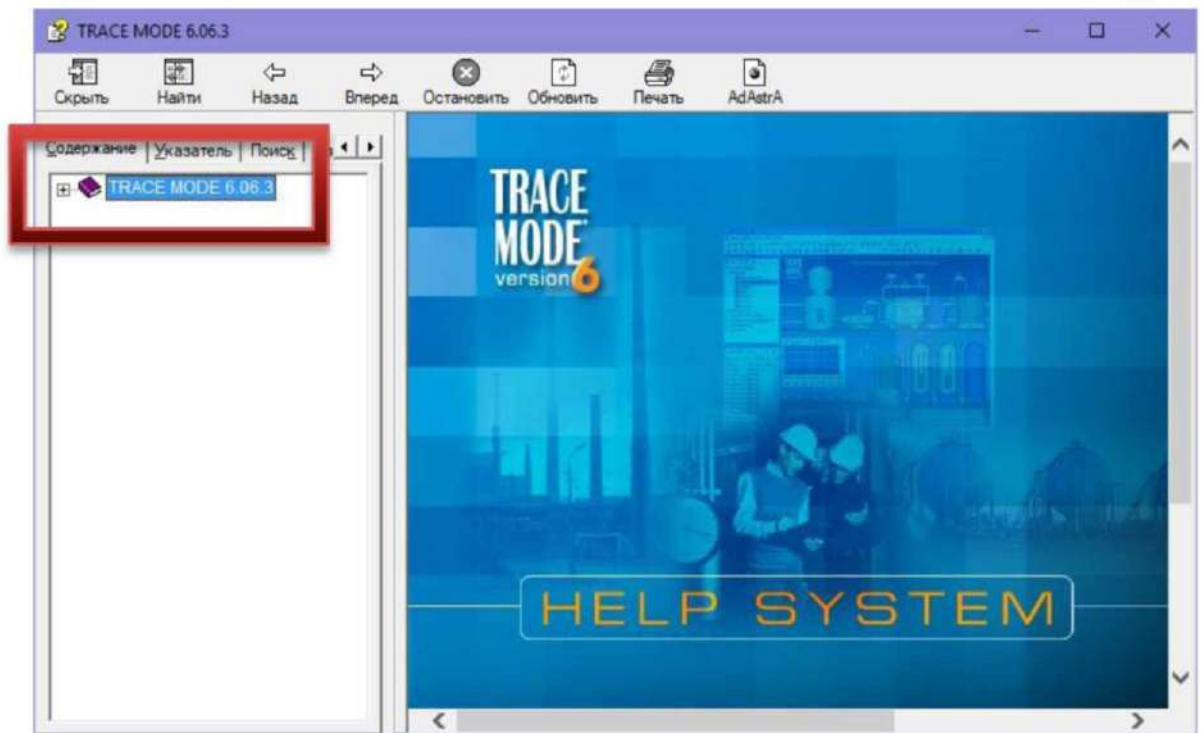


Рисунок 13 - Окно «Содержание» интегрированной среды разработки SCAD A Trace Mode

4.2.10 Разработка алгоритмов управления

Типовая последовательность разработки алгоритма управления включает следующие этапы:

- 1 Анализ циклограммы работы объекта управления или схемы его функционирования. Если циклограмма работы или схема отсутствуют, то, используя словесное описание алгоритма, необходимо их разработать.

- 2 Составление таблиц входных и выходных сигналов программируемого контроллера (таблица 2). Таблицы должны содержать наименование и условное обозначение сигналов, их источник или приёмник, адресацию сигналов и их привязку к контактам разъёмов контроллера.

- 3 В некоторых случаях является полезным описать алгоритм (циклограмму) уравнениями алгебры логики.

Итогом этапа разработки алгоритмов управления становится информация, используемая для разработки программ для программируемого логического

Таблица 2 - Пример упрощённой формы таблицы сигналов входа-выхода

№	Код контура	Позиция КИП	Наименование параметра	Шкала		Ед. Изм.	Тип сигнала	Предупредительная сигнализация		Предаварийная сигнализация	
				мин	макс			L	H	LL	HH
1	2	3	4	5	6	7	8	9	10	11	12

контроллера.

Языки программирования ПЛК регламентированы стандартом [10]. Документ определяет пять языков программирования ПЛК: *текстовые* - список инструкций (IL) и структурированный текст; *графические* - релейно-контактные схемы (LD) и функциональные блочные диаграммы (FBD); *смешанный (и графический и текстовый)* - последовательная функциональная схема (SFC).

В курсовой работе можно использовать любой из этих языков для создания программ управления. Эти языки стандартно входят в следующие среды: SCADA Trace Mode, CoDeSys, Owen Logik, Step 7, UltraLogik и тому подобное.

Описанные языки могут применяться для запуска или останова исполнительных механизмов, включения/отключения сигнализации, реализации законов управления физическими величинами через контроллер (ПИД-регулирование температуры, давления и так далее) и тому подобного.

Тексты спроектированных программ выносят в приложения пояснительной записки курсовой работы.

4.2.11 Расчёт показателей качества системы

Предполагается, что до выполнения этого этапа обучающийся уже имеет полученную средствами теории автоматического управления математическую модель системы и конкретные параметры настройки регулятора(ов).

Проверить качество работы спроектированной системы можно по виду переходных процессов объекта, подавая на его вход ступенчатое воздействие. В теории автоматического управления уже были изучены методы оценки показателей качества при ступенчатом воздействии, такие как: время переходного процесса, максимальная амплитуда выходного сигнала, перерегулирование и тому подобное.

Реализовать моделирование системы и проверку показателей качества можно средствами таких программных средств, как MathCAD, SamSim, Matlab, SimInTech, VisSim.

Окончательные результаты моделирования помещаются в курсовую работу вместе с настройками регуляторов.

4.3 Заключение

Объём заключения составляет не более двух страниц текста. В заключении оформляются основные выводы по работе:

- насколько полно была решена поставленная задача и какие методы при этом использовались;
- итоговые результаты анализа спроектированной системы;
- краткое описание возможных направлений дальнейшей разработки.

4.4 Список использованных источников

Список источников идёт в конце работы после заключения, но перед приложениями. Любой источник информации, используемый в курсовой работе, должен быть пронумерован и упомянут в этом списке. Оформление списка использованных источников осуществляется в соответствии с [3].

4.5 Приложения

В приложения выносятся слишком большие графические иллюстрации или иные материалы, которые носят дополнительный, поясняющий характер к тексту

курсовой работы. Такими материалами могут быть: таблица, схемы, схемы алгоритмов, листинги программ, экранные формы, копии или шаблоны документов и тому подобное. Оформление приложений осуществляется в соответствии с [3].

5 Утверждение и защита курсовой работы

Оформленная курсовая работа предоставляется руководителю в электронной форме не позднее двух недель до начала сессии.

В случае обнаружения ошибок: несоответствия стандартам оформления, недостаточности объёма работы, либо нарушений требований к работе, а также несоответствия предложенной теме курсовая работа возвращается для доработки и исправления замечаний.

Руководитель вправе возвращать работу не один раз в зависимости от полноты устранения выявленных ошибок.

Утверждение работы после полной коррекции завершается подписью руководителя на титульном листе, что означает допуск к её защите.

На защиту студент готовит доклад на пять минут и презентацию, представляющие его работу. Структура доклада повторяет структуру курсовой работы. В докладе излагаются только самые важные с точки зрения спроектированной системы моменты, а также цели и задачи работы.

Таким образом, защита курсовой работы проходит в присутствии руководителя в форме:

- доклада по выполненному труду;
- презентации по курсовой работе;
- ответов на вопросы.

При оценке работы учитывается полнота разработки, выполнение намеченных задач, качество проведенного анализа системы, эффективность выбранных методов исследования, а также умение аргументированно и в логической последовательности излагать материал.

В случае отсутствия завершённой работы в намеченный срок или отрицательного результата ее защиты, кроме случаев с уважительными причинами,

студент считается имеющим академическую задолженность и не допускается к сдаче экзамена или зачёта по дисциплине. Следует заметить, что оценка курсовой работы по окончании обучения идёт в приложение к диплому.

6 Формулировка тематики курсовых работ

В соответствии с рабочей программой и содержанием дисциплины «Автоматизация технологических процессов и производств» в качестве ориентировочных шаблонных формулировок предлагаются следующие темы курсовой работы:

- 1) Автоматизация процесса получения смеси с определённой концентрацией компонента;
- 2) Автоматизация процесса извлечения твёрдой фазы из жидкости в отстойниках;
- 3) Автоматизация процесса центрифугирования;
- 4) Автоматизация процесса фильтрации жидкостей;
- 5) Автоматизация процесса фильтрации газов;
- 6) Автоматизация процесса флажной очистки газов;
- 7) Автоматизация процесса очистки газов электрической дугой;
- 8) Проектирование системы автоматизации процесса нагрева продукта;
- 9) Автоматизация процесса нагрева продукта в трубчатой печи;
- 10) Проектирование системы автоматизации процесса вакуумной выпарки;
- 11) Автоматизация процесса кристаллизации разбавленного продукта;
- 12) Автоматизация процесса регулирования расхода продукта с коррекцией по составу целевых продуктов;
- 13) Автоматизация процесса регулирования состава абсорбента в колонне;
- 14) Автоматизация процесса адсорбции;
- 15) Автоматизация процесса десорбции в кипящем слое;
- 16) Автоматизация процесса сушки влажного материала;
- 17) Автоматизация процесса противоточной сушки влажного материала;
- 18) Автоматизация процесса ленточной сушки влажного материала;
- 19) Модернизация процесса струйной сушки влажного материала;

20) Автоматизация процесса радиационной сушки влажного материала;

21) Автоматизация процесса измельчения материала.

Данные темы представлены исключительно в плане демонстрации различных

названий тематик курсовой работы или примерного направления исследования, поэтому выбираемые студентами тематики могут и не совпадать с предложенным списком.

7 Источники информации по автоматизированным системам управления технологическими процессами

7.1 Основная литература

1 СТО 70238424.27.140.010-2010. Автоматизированные системы управления технологическими процессами ГЭС И ГАЭС. Условия создания. Нормы и требования. -Введ. 2010-09-30. -М.: НПИВЭЛ, 2010. - 39 с.

2 Старостин, А.А. Технические средства автоматизации и управления: учеб. пособие / А.А. Старостин, А.В. Лаптева. - Екатеринбург: Изд-во Урал, ун-та, 2015. - 168 с - ISBN 978-5-7996-1498-0.

3 СТО 02069024.101-2015. РАБОТЫ СТУДЕНЧЕСКИЕ. Общие требования и правила оформления. - Введ. 2016-02-08. - Оренбург: ОГУ, 2015. - 89 с. - Режим доступа: [http://www.osu.ru/docs/official/standart/standart_101-2015 .pdf](http://www.osu.ru/docs/official/standart/standart_101-2015.pdf).

4 ГОСТ 34.602-89. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. - Введ. 1990-01-01. -М.: Изд-во стандартов, 1989. - 11 с.

5 Проектирование систем автоматизации технологических процессов: Справочное пособие / Под ред. А.С. Ключева. - М.: Энергоатомиздат, 1990. - 464 с. -ISBN 978-5-903034-44-4.

6 ГОСТ 2.784-96. Единая система конструкторской документации. Обозначения условные графические. Элементы трубопроводов. - Введ. 1998-01-01. - М.: Стандартинформ, 2005, 2012. - 24 с.

7 ГОСТ 2.785-70. Единая система конструкторской документации. Обозначения условные графические. Арматура трубопроводная. - Введ. 1971-01-01. - М.: ИПК Издательство стандартов, 1998, Стандартинформ, 2012. - 30 с.

8 ГОСТ 21.208-2013. Система проектной документации для строительства. Автоматизация технологических процессов. Обозначения условные приборов и средств автоматизации в схемах. - Введ. 2014-11-01. - М.: Стандартинформ, 2015. -28 с.

9 Теория автоматического управления: в 2-х ч./ Под ред. Воронова А.А. - М.: Высш. школа, 1986. - Ч.1 - 367с.; Ч.2 - 504 с.

10 ГОСТ Р МЭК 61131-3-2016. Контроллеры программируемые. Часть 3. Языки программирования. - Введ. 2017-04-01. - М.: Стандартиформ, 2016. - 227 с.

11 Скрябин, В.А. Автоматизация производственных процессов в машиностроении: Учебник / В.А. Скрябин, А.Г. Схиртладзе, А.Е. Зверовщиков - М.: КУРС, НИЦ ИНФРА-М, 2017. - 320 с.: 60х90 1/16 (Переплёт 7БЦ) ISBN 978-5-906818-607. - Режим доступа: <http://znanium.com/bookread2.php?book=752393>.

12 Фурсенко, С.Н. Автоматизация технологических процессов: учебное пособие / С.Н. Фурсенко, Е.С. Якубовская, Е.С. Волкова. - М.: НИЦ ИНФРА-М; Мн.: Нов. знание, 2015. - 377 с.: ил.; 60х90 1/16. - (Высшее образование). (п) ISBN 978-5-16-010309-9. - Режим доступа: <http://znanium.com/bookread2.php?book=483246>.

13 Клепиков, В.В. Автоматизация производственных процессов: учеб. пособие / В.В. Клепиков, Н.М. Султанзаде, А.Г. Схиртладзе. - М.: ИНФРА-М, 2017. - 208 с. - (Высшее образование: Бакалавриат). - www.dx.doi.org/10.12737/18466. - ISBN 978-5-16-011109-4, ISBN-online 978-5-16-103175-9. - Режим доступа: <http://znanium.com/bookread2.php?book=883959>.

14 Автоматизация технологических процессов и производств: учебное пособие / А.А. Иванов. - 2-е изд., испр. и доп. - М.: Форум: НИЦ ИНФРА-М, 2017. -224 с.: 60х90 1/16. - (Высшее образование). - ISBN 978-5-91134-948-6. - Режим доступа: <http://znanium.com/bookread2.php?book=795655>.

15 Проектирование автоматизированных систем производства: учебное пособие / В.Л. Конюх. - М.: КУРС: НИЦ ИНФРА-М, 2014. - 312 с.: 60х90 1/16. (переплет) ISBN 978-5-905554-53-7. - Режим доступа: <http://znanium.com/bookread2.php?book=449810>.

16 Шишов, О.В. Программируемые контроллеры в системах промышленной автоматизации: учебник / О.В. Шишов. - М.: ИНФРА-М, 2019. - 365 с. + Доп. материалы [Электронный ресурс]. - (Высшее образование: Бакалавриат).

-www.dx.doi.org/10.12737/17505. - ISBN 978-5-16-011205-3, ISBN-online 978-5-16-103331-9. - Режим доступа: <http://znanium.com/bookread2.php?book=978937>.

17 Шишов, О.В. Технические средства автоматизации и управления: учебное пособие / Шишов О.В. - М.: НИЦ ИНФРА-М, 2018. - 396 с. - ISBN 978-5-16-010325-9. - Режим доступа: <http://znanium.com/bookread2.php?book=973005>.

18 Проектирование АСУТП с использованием SCADA-систем: учебное пособие / А.В. Герасимов, А.С. Титовцев; Министерство образования и науки России, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Казанский национальный исследовательский технологический университет». - Казань: Издательство КНИТУ, 2014. - 128 с.: табл., ил. - Библиогр.: с. 96. - ISBN 978-5-7882-1514-3; То же [Электронный ресурс]. - Режим доступа: <http://biblioclub.ru/mdex.php?page=book&id=427985> (24.01.2019).

19 ГОСТ 21.408-2013. Система проектной документации для строительства. Правила выполнения рабочей документации автоматизации технологических процессов. - Введ. 2014-11-01. - М.: Стандартинформ, 2014. - 39 с.

20 Федоров, Ю.Н. Справочник инженера по АСУТП: проектирование и разработка: учебно-практическое пособие: в 2 т. / Ю.Н. Федоров. - 2-е изд. - Москва; Вологда: Инфра-Инженерия, 2017. - Т. 1. - 449 с.: ил., схем., табл. - Библиогр. в кн. - ISBN 978-5-9729-0122-7; то же [Электронный ресурс]. - Режим доступа: <http://biblioclub.ru/index.php?page=book&id=466779>.

21 Федоров, Ю.Н. Справочник инженера по АСУТП: проектирование и разработка / Ю.Н. Федоров. - Москва: Инфра-Инженерия, 2008. - 928 с. - ISBN 978-5-9729-0019-0; То же [Электронный ресурс]. - Режим доступа: <http://biblioclub.ru/index.php?page=-book&id=70501> (24.01.2019).

7.2 Дополнительная литература

1 Лукас, В.А. Теория управления техническими системами: компактный учебный курс для вузов / В.А. Лукас. - 3-е изд., перераб. и доп. - Екатеринбург: Изд-во

УГГГА, 2002. - 675 с.

2 Кошкидько, В.Г. Основы программирования в системе MATLAB: учебное пособие / В.Г. Кошкидько, А.И. Панычев - Таганрог: Южный федеральный университет, 2016. - 84 с.: ISBN 978-5-9275-2048-0. - Режим доступа:

<http://znanium.com/bookread2.php?book=991834>.

3 Тимохин, А.Н. Моделирование систем управления с применением Matlab: учеб. пособие / А.Н. Тимохин, Ю.Д. Румянцев; под ред. А.Н. Тимохина. - М.: ИН-ФРА-М, 2019. - 256 с. + Доп. материалы [Электронный ресурс]. - (Высшее образование: Бакалавриат). - www.dx.doi.org/10.12737/14347. - ISBN 978-5-16-010185-9, ISBN-online 978-5-16-102042-5. - Режим доступа: <http://znanium.com/bookread2.php?book=1004245>.

7.3 Периодические издания

- Информационные технологии в проектировании и производстве: журнал. - М.: Агентство «Роспечать», 2016.

- Автоматизация. Современные технологии: журнал. - М.: Агентство «Роспечать», 2018.

- Вестник машиностроения: журнал. - М.: Агентство «Роспечать», 2018.

- Машиностроитель: журнал. - М.: Агентство «Роспечать», 2016.

- Автоматика и телемеханика: журнал. - М.: Наука, 2016.

- СТИН: журнал. - М.: Агентство «Роспечать», 2016.

- Мехатроника, автоматизация, управление: журнал. - М.: Агентство «Роспечать», 2018.

- Справочник. Инженерный журнал: журнал. - М.: Агентство «Роспечать», 2018.

- Информационно-измерительные и управляющие системы: журнал. - М.: Агентство «Роспечать», 2018.
- Автоматизация в промышленности: журнал. - М.: Агентство «Роспечать», 2016.

7.4 Интернет-ресурсы

- АСУТП.ги - средства и системы компьютерной автоматизации. - Режим доступа: <http://www.asutp.ru>.
- SCADA TRACE MODE. SCADA системы для АСУ ТП. - Режим доступа: <http://www.adastra.ru>.
- Сайт издательства «Новые технологии» - теоретические и прикладные научно-технические журналы, обеспечивающие научной, производственной, обзорно-аналитической и образовательной информацией руководящих работников и специалистов промышленных предприятий, научных академических и отраслевых организаций, а также учебных заведений в области приоритетных направлений развития науки и технологий. - Режим доступа: <http://www.novtex.ru/>.
- «Открытое образование» - современная образовательная платформа, предлагающая онлайн-курсы по базовым дисциплинам, изучаемым в российских университетах. - «Открытое образование», Каталог курсов, Политех: «Современная промышленная электроника». - Режим доступа: <https://openedu.ru/course/spbstu/MODIEL/>.
- «Открытое образование» - современная образовательная платформа, предлагающая онлайн-курсы по базовым дисциплинам, изучаемым в российских университетах. - «Открытое образование», Каталог курсов, Университет ИТМО: «Элементы систем автоматического управления». - Режим доступа: <https://openedu.ru/course/ITMOUniversity/ACSE/>.
- «Открытое образование» - современная образовательная платформа, предлагающая онлайн-курсы по базовым дисциплинам, изучаемым в российских университетах. - «Открытое образование», Каталог курсов, Университет ИТМО: «Линейные системы автоматического управления». - Режим доступа: <https://openedu.ru/course/ITMOUniversity/LINACS/>.
- «Открытое образование» - современная образовательная платформа, предлагающая онлайн-курсы по базовым дисциплинам, изучаемым в российских

университетах. - «Открытое образование», Каталог курсов, МИСиС: «Теория автоматического управления. Нелинейные системы автоматического управления». -Режим доступа: <https://openedu.ru/course/misis/TAU/>.

- Сайт компании ОВЕН - ведущего российского разработчика и производителя контрольно-измерительных приборов и средств автоматизации для различных отраслей промышленности. - Режим доступа: <https://www.owen.ru>.

Список использованных источников

1 СТО 70238424.27.140.010-2010. Автоматизированные системы управления технологическими процессами ГЭС И ГАЭС. Условия создания. Нормы и требования. - Введ. 2010-09-30. - М.: НП ИНВЭЛ, 2010. - 39 с.

2 Старостин, А.А. Технические средства автоматизации и управления: учеб. пособие / А.А. Старостин, А.В. Лаптева. - Екатеринбург: Изд-во Урал. ун-та, 2015. - 168 с. - ISBN 978-5-7996-1498-0.

3 [СТО 02069024.101-2015. РАБОТЫ СТУДЕНЧЕСКИЕ](http://www.osu.ru/docs/official/standart/standart_101-2015.pdf). Общие требования и правила оформления. - Введ. 2016-02-08. - Оренбург: ОГУ, 2015. - 89 с. - Режим доступа: [http://www.osu.ru/docs/official/standart/standart_101-2015 .pdf](http://www.osu.ru/docs/official/standart/standart_101-2015.pdf).

4 ГОСТ 34.602-89. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. - Введ. 1990-01-01. - М.: Изд-во стандартов, 1989. - 11 с.

5 Проектирование систем автоматизации технологических процессов: справочное пособие / Под ред. А.С. Ключева. - М.: Энергоатомиздат, 1990. - 464 с. - ISBN 978-5-903034-44-4.

6 ГОСТ 2.784-96. Единая система конструкторской документации. Обозначения условные графические. Элементы трубопроводов. - Введ. 1998-01-01. - М.: Стандартинформ, 2005, 2012. - 24 с.

7 ГОСТ 2.785-70. Единая система конструкторской документации. Обозначения условные графические. Арматура трубопроводная. - Введ. 1971-01-01. - М.: ИПК Издательство стандартов, 1998, Стандартинформ, 2012. - 30 с.

8 ГОСТ 21.208-2013. Система проектной документации для строительства. Автоматизация технологических процессов. Обозначения условные приборов и средств автоматизации в схемах. - Введ. 2014-11-01. - М.: Стандартинформ, 2015. - 28 с.

9 Теория автоматического управления: в 2-х ч./ Под ред. Воронова А.А. - М.: Высш. школа, 1986. - Ч.1 - 367с.; Ч.2 - 504 с.

10 ГОСТ Р МЭК 61131 -3-2016. Контроллеры программируемые. Часть 3.
Язы-
ки программирования. - Введ. 2017-04-01. - М.: Стандартиформ, 2016. - 227 с.

SCADA-система

Методические указания для
выполняющих курсовые работы

Методические указания по выполнению курсовых работ

В документе описывается пример выполнения курсовой работы. Справочные сведения помечены форматированием.

Задание

Тема: Автоматизированная система управления (АСУ) электронагревателями.

Описание: В каждой из трех комнат необслуживаемого технического помещения находится температурный датчик и два электронагревателя (рис. 1). Температура в комнатах в зимнее время не должна опускаться ниже плюс 5 °С.

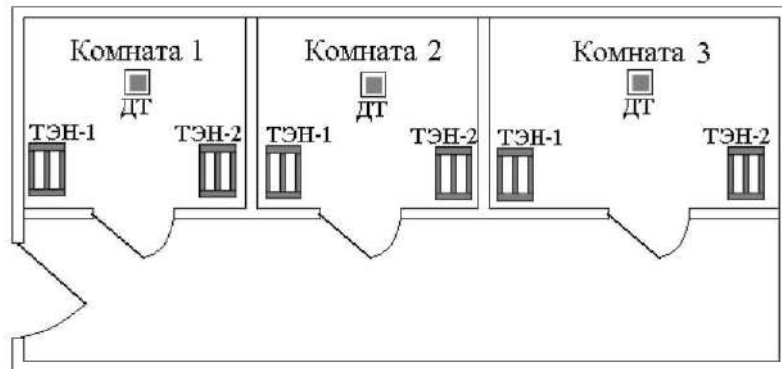


Рис. 1. Схема технического помещения.

ДТ - датчик температуры;

ТЭН-1, ТЭН-2 - трубчатые электронагреватели.

Задание: Реализовать АСУ электронагревателями

АСУ электронагревателями в каждой комнате должна работать в двух режимах:

1. **Ручной.** Включение и отключение электронагревателей осуществляется оператором.
2. **Автоматический.** Включение и отключение электронагревателей осуществляется в зависимости от температуры в комнате. При температуре ниже плюс 5 °С электронагреватели должны включиться, при достижении температуры плюс 15 °С — должны выключиться. Также предусмотреть возможность изменения оператором температуры включения и выключения электронагревателя (т.е. уставок).

Последовательность работ:

1. Составить перечень сигналов. Таблица сигналов должна содержать следующие поля: №, шифр и наименование сигнала, тип сигнала - аналоговый / дискретный, назначение сигнала - вход / выход, единицы измерения, шкала (границы изменения).
2. Реализовать технологическую программу в системе программирования контроллеров KLogis. Для эмуляции значения температурного датчика в комнатах использовать функциональный блок ДемоСигнал в режиме синусоиды со следующими параметрами: период 2 минуты, амплитуда 0..50 .
3. Создать технологическую мнемосхему в модуле Визуализация. Отобразить схему технического помещения. Реализовать:
 - а) управление и отображение режима работы АСУ (ручной/автоматический);
 - б) управление работой электронагревателей каждой из комнат в ручном режиме;
 - в) отображение значения температуры в каждой из комнат, с отображением единиц измерения,
причём падение температуры в каждой из комнат отобразить синим цветом датчика, нормальное значение - желтым цветом.
 - г) работу каждого из электронагревателей отобразить красным цветом, простой - серым.

4. Настроить БД технологических параметров, добавить в нее все имеющиеся параметры.
5. Настроить сигнализацию, срабатывающую при температуре в каждой из комнат ниже 5 С (аварийный минимум).
6. Настроить отчеты:
 - а) средняя температура в комнатах за сутки;
 - б) включение и отключение электронагревателей за сутки.

Общие сведения

SCADA-система «КАСКАД» представляет собой мощный инструмент для наблюдения, анализа и управления процессами в системах автоматизации в различных областях промышленности. Она имеет в своем распоряжении все необходимые инструменты, присущие современным пакетам данного класса, и уникальные особенности.

«КАСКАД» — модульная система, имеющая клиент-серверную архитектуру. Можно выделить две основные группы программных модулей SCADA-системы «КАСКАД» — серверные и клиентские. *Серверная* часть отвечает за сбор данных с различных устройств, их обработку, ведение архивов, отслеживание нарушений в технологическом процессе. К ней относятся такие модули, как Сервер Доступа к Данным (СДД), Модули Доступа к Данным (МДД) и Модули Обработки Данных (МОД).

Клиентская часть обеспечивает взаимодействие с пользователем: наблюдение за ходом технологического процесса, управление процессом, просмотр накопленных архивов, формирование отчетов. К ним относятся следующие модули: Диалоговая среда контроля и управления (модуль Визуализации), модуль просмотра истории и Модуль Формирования рапортов.

Поэтому разработка проекта в SCADA-системе КАСКАД состоит из нескольких этапов.

Задачи по параметрической настройке проекта решаются с помощью программы Конфигуратор. Назначение Конфигуратора - свести все модули SCADA-системы «КАСКАД» в единую оболочку, позволяющую удобно создавать и настраивать проекты.

Открыть Конфигуратор можно либо через проводник (или другой файловый менеджер), запустив на выполнение файл Configurator.exe. Окно программы Конфигуратор представлено на рис. 2.



Рис. 2. Конфигуратор

Конфигуратор имеет пять вкладок, находящихся в левой части окна: выбор проекта, описание проекта, настройка проекта, сервер и клиенты, утилиты. В правой части окна отображается содержание этих вкладок.

1. Вкладка *Выбор проекта* позволяет открыть существующий проект, либо создать новый, запустив мастер создания проектов.
2. Вкладка *Описание проекта* выводит информацию об открытом проекте
3. На вкладке *Настройка проекта* расположены кнопки вызова настроечных модулей для всех подсистем SCADA-системы «КАСКАД»
4. На вкладке *Сервер, клиенты* расположены кнопки вызова исполняемых модулей SCADA-системы «КАСКАД»
5. С вкладки *Утилиты* запускаются вспомогательные утилиты SCADA-системы.

Все кнопки расположены в той последовательности, в которой должна логически происходить настройка проекта.

1. Создание проекта

Для создания проектов разработан т.н. *Мастер создания проектов*. Для запуска мастера необходимо на вкладке *Выбор проекта* нажать кнопку «Создать новый проект» (см. рис. 1.1).

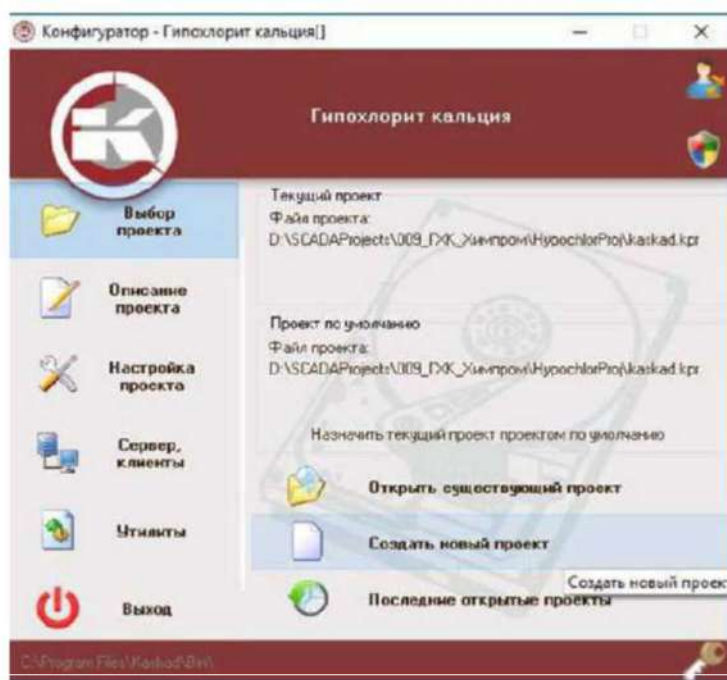


Рис. 1.1. Запуск мастера создания проектов

Процесс создания проекта состоит из нескольких шагов.

1. На первом шаге необходимо **указать путь к файлу проекта** (выбрать Папку проекта), задать название проекта и его описание (рис. 1.2). При установке флажка «Назначить проектом по умолчанию» существующий проект будет открываться по умолчанию при запуске всех модулей SCADA-системы КАСКАД.

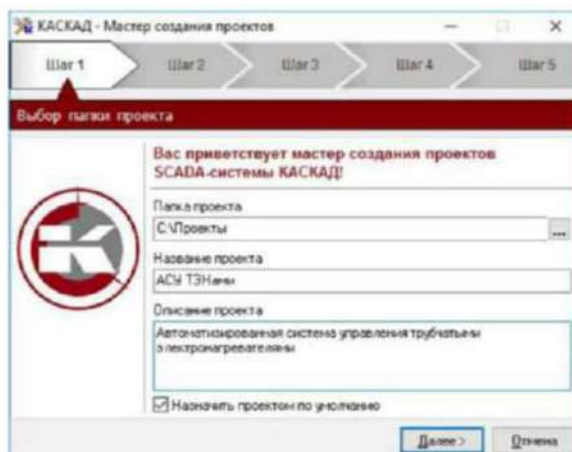


Рис. 1.2. Первый шаг создания проекта

2. Второй шаг — **добавление рабочей станции в новый проект** (рис. 1.3). В нашем случае проект состоит из одной серверной рабочей станции.

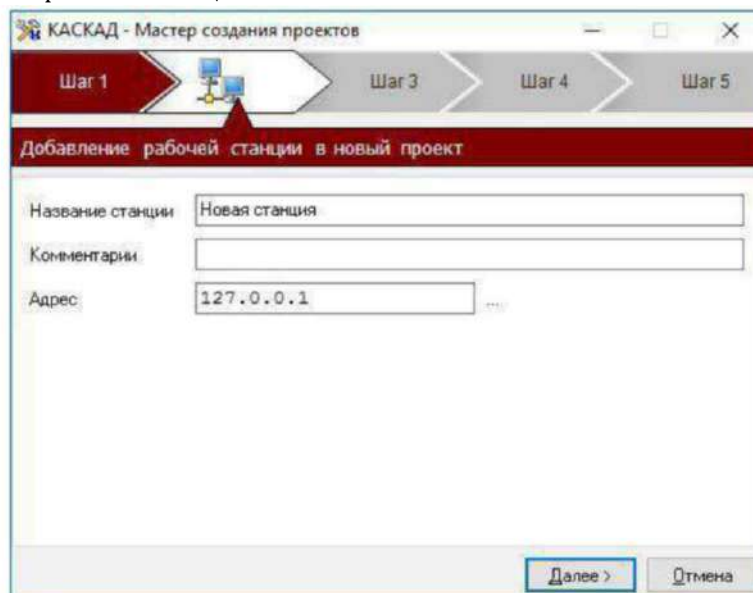


Рис. 1.3. Второй шаг создания проекта

3. **Создание/подключение базы данных пользователей** (рис. 1.4). Полный путь к БД пользователей задаётся с помощью полей редактирования *Сервер* и *Файл БД на сервере*. *Сервер* может быть выбран *Из проекта* - в качестве сервера будет служить добавленная на втором шаге *Станция*. Путь к *Файлу БД* может быть задан относительно пути к проекту.

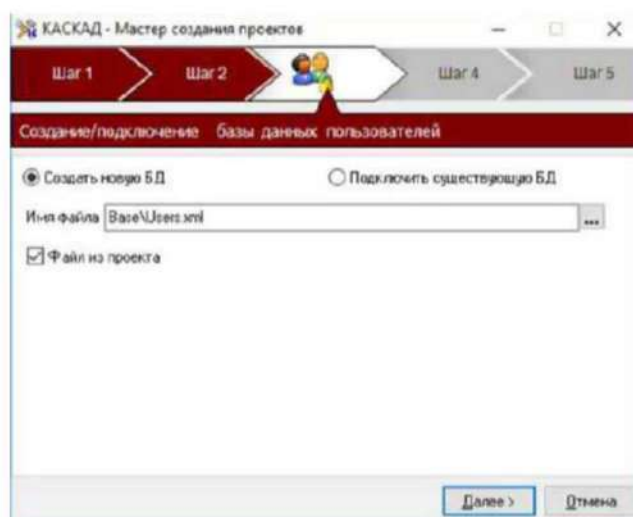


Рис. 1.4. Третий шаг создания проекта

Оставляем значения по умолчанию, нажимаем **Далее**, при этом происходит создание базы данных пользователей для проекта.

4. **Добавление пользователя с правами администратора** (рис. 1.5). По умолчанию такой пользователь имеет имя *admin*, но можно ввести любое имя. Также можно ввести Полное имя и Пароль. При установке флажка **Назначить пользователем по умолчанию** любое приложение проекта по умолчанию будет открываться с правами этого пользователя, иначе при запуске любого приложения из состава SCADA-системы необходимо будет проводить авторизацию (вводить имя пользователя и пароль).

Данный этап аналогичен п.3. Здесь также оставляем все значения по умолчанию. При нажатии на

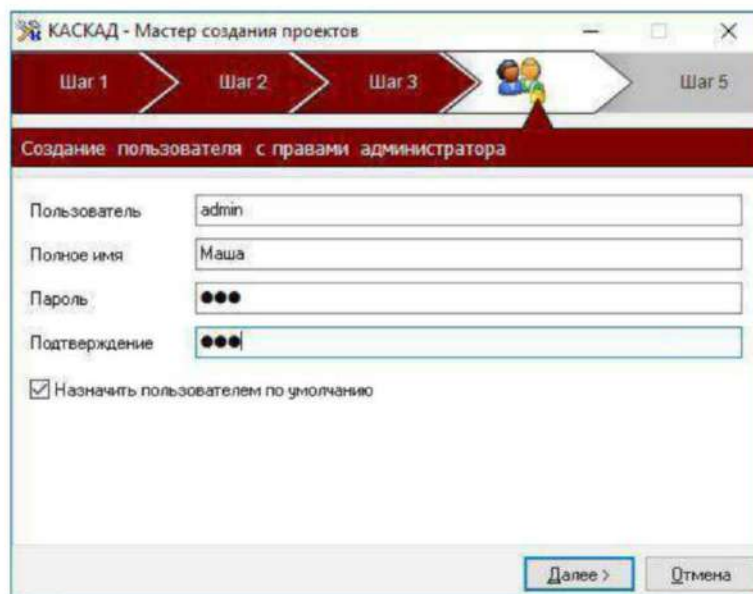


Рис. 1.5. Четвёртый шаг создания проекта 5. Заключительный этап при создании проекта — создание базы данных событий (рис. 1.6).

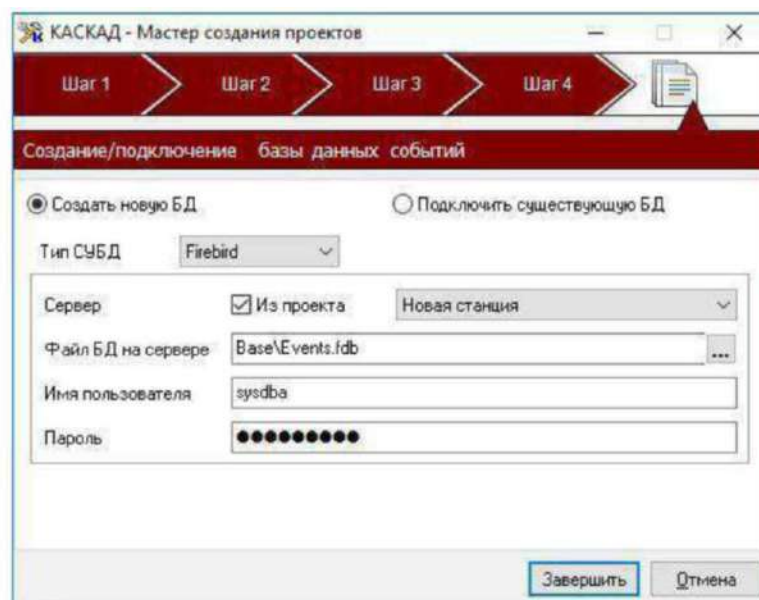


Рис. 1.6. Пятый шаг создания проекта

кнопку **Завершить** мастер заканчивает свою работу, проект создан.

Информацию о проекте можно посмотреть на вкладке **Описание проекта** (рис. 1.7).

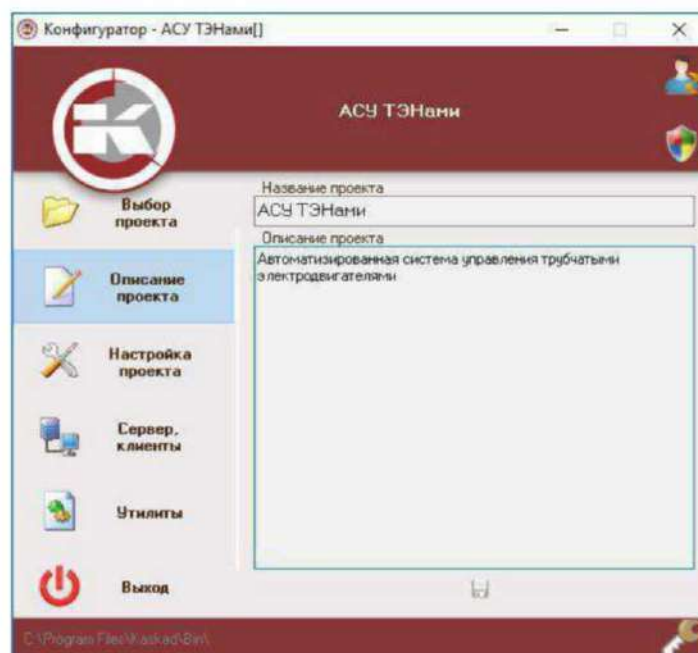


Рис. 1.7. Вкладка «Описание проекта»

Здесь можно редактировать Название и Описание проекта, которые являются информационными составляющими проекта.

2. Настройка прав пользователя

После создания проекта необходимо его настроить. Для этого предназначен основной раздел Конфигуратора - вкладка **Настройка проекта** (рис. 2.1), которая позволяет запускать конфигурационные модули SCADA-системы "КАСКАД". Здесь можно настроить права пользователей, сетевое взаимодействие, базы данных событий, контроллеры, сервер доступа к данным, паспорта, базы данных технологических параметров, алармы.

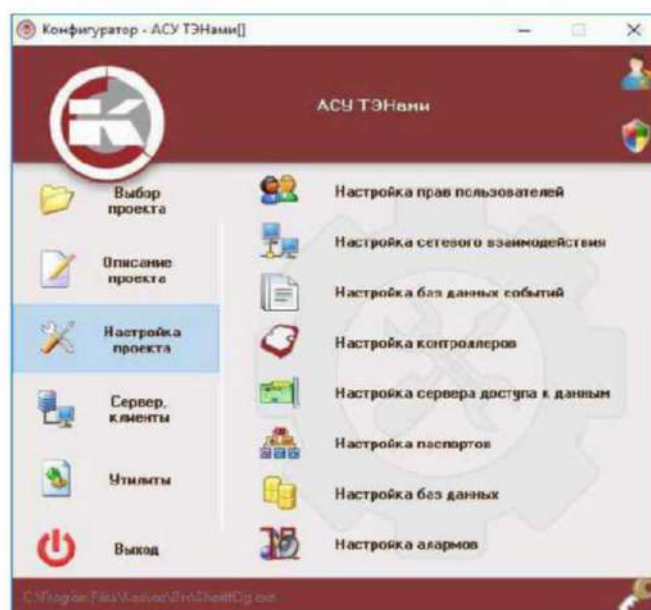


Рис. 2.1. Вкладка «Настройка проекта»

Первый этап — **Настройка прав пользователей** (рис. 2.2). При выборе этого пункта вызывается программа управления подсистемой администрирования.

Подсистема администрирования предназначена для ограничения доступа пользователей к

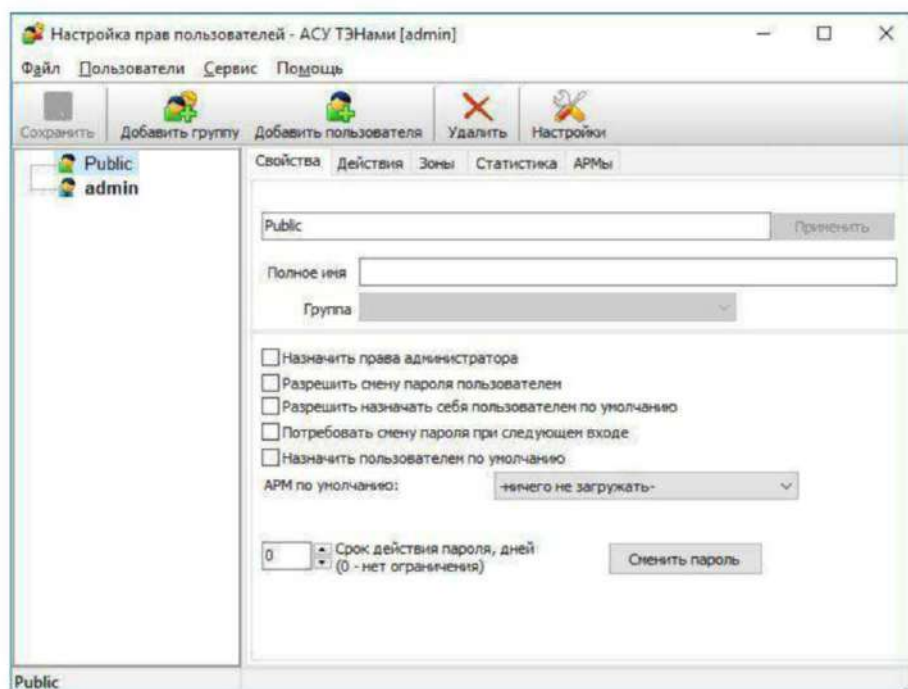


Рис. 2.2. Пользователи проекта

Выберите пользователя admin и нажмите **Сервис - Настройка**, откроется окно настроек. Необходимо добавить приложение **Систему программирования контроллеров**. Для этого нажмите на **Добавить приложение** и выберите KLogic (рис. 2.3). В списке приложений должно появиться приложение **Система программирования контроллеров**.

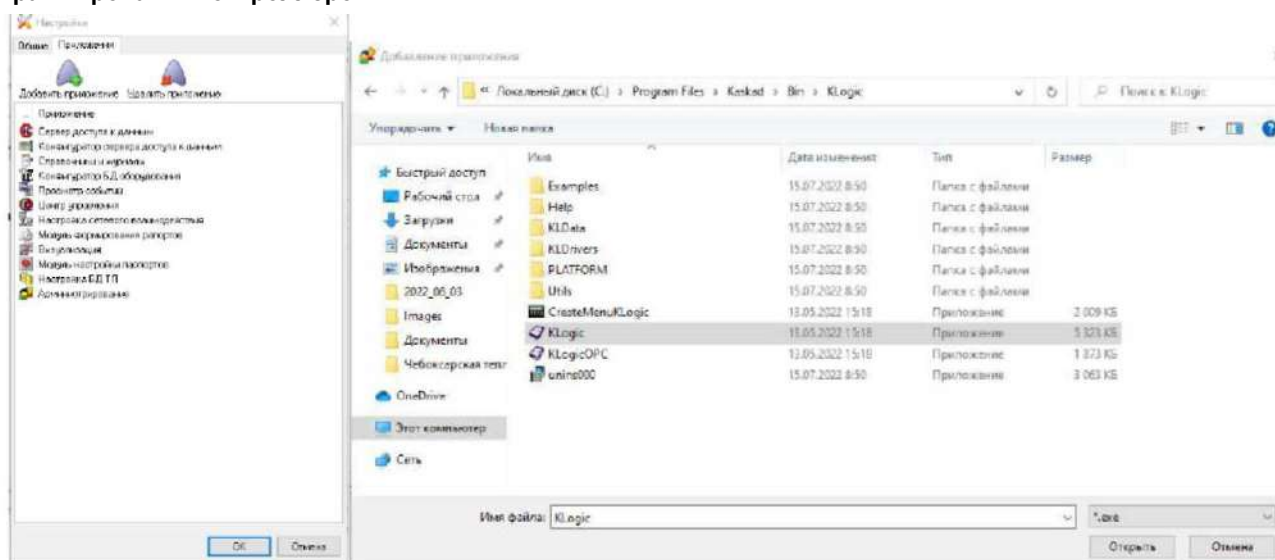


Рис. 2.3. Добавление приложения

различным ресурсам SCADA-системы «КАСКАД». Используя данный модуль, можно регистрировать и удалять приложения, доступ к функциям которых нужно запретить или разрешить, добавлять/удалять пользователей, объединять пользователей в группы, добавлять и удалять группы. Пользователь с правами администратора и именем admin уже был создан в пункте 1.4.

3. Настройка сетевого взаимодействия

Данное приложение (рис. 3.1) позволяет настроить список серверов и рабочих станций, используемых в проекте, состоящем из более чем одного компьютера.

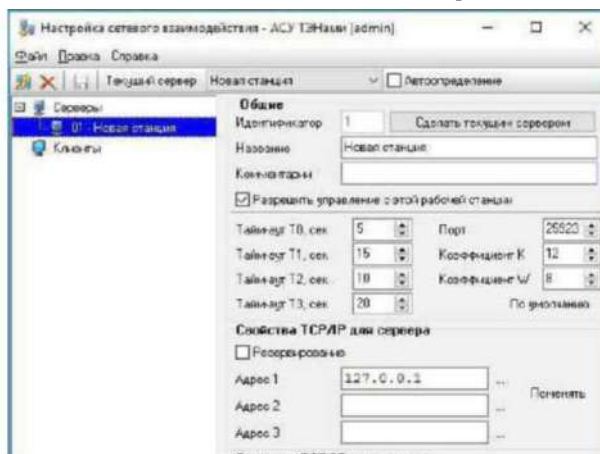


Рис. 3.1. Настройка сетевого взаимодействия

В левой части окна отображаются станции, задействованные в проекте, в правой части — настройки выбранной станции. Для случаев, когда в проекте участвует несколько компьютеров, панель **Свойства TCP/IP для сервера** определяет IP-адреса станции, по которому сервер определяет себя при запуске, а так же адрес, по которому его найдут другие сервера из списка станций, либо резервный/основной сервер в паре. В таком случае **Адрес 1** и **Адрес 2** становятся основным и резервным соответственно. **Основной** адрес определяет IP-адрес основного сервера, **резервный** - резервного. **Свойства TCP/IP для клиента** определяют IP-адрес сервера, по которому будут слать запросы к нему клиентские приложения. В простейшем случае оба адреса совпадают. В случае, когда сервера находятся в технологической сети, а клиенты к ним подключены по другому сетевому интерфейсу, возможно, адреса для сервера и для клиента будут различаться.

Сервер - компьютер, который непосредственно взаимодействует с объектом управления через специализированные устройства, т.е. связан с объектом управления.

Рабочая станция - компьютер, который не подключается непосредственно к объекту управления, но получает данные с сервера.

Настройка сетевого взаимодействия необходима для реализации механизма «единого проекта», когда директория электронного проекта может быть полностью размножена на все компьютеры (серверы/рабочие станции), участвующие в автоматизированной системе, и ни на каком из них не будет необходимости изменять какие-либо настройки, в том числе сетевые.

В пункте 1.2 уже была указана одна рабочая станция проекта.

4. Базы данных событий

Модуль просмотра и настройки архива событий используется для просмотра и настройки событий, генерируемых сервером доступа к данным (СДД), различными библиотеками и приложениями SCADA-системы «КАСКАД»: открытие/закрытие проекта, запуск СДД, загрузка и выгрузка библиотек, запуск приложений и выход из них, изменение настроек, управление технологическим процессом и т.п. В любом проекте необходимо наличие, как минимум, одной БД событий.

В данном проекте уже была создана одна база данных событий (пункт 1.2).

Основное окно программы настройки баз данных событий представлено на рис. 4.1.

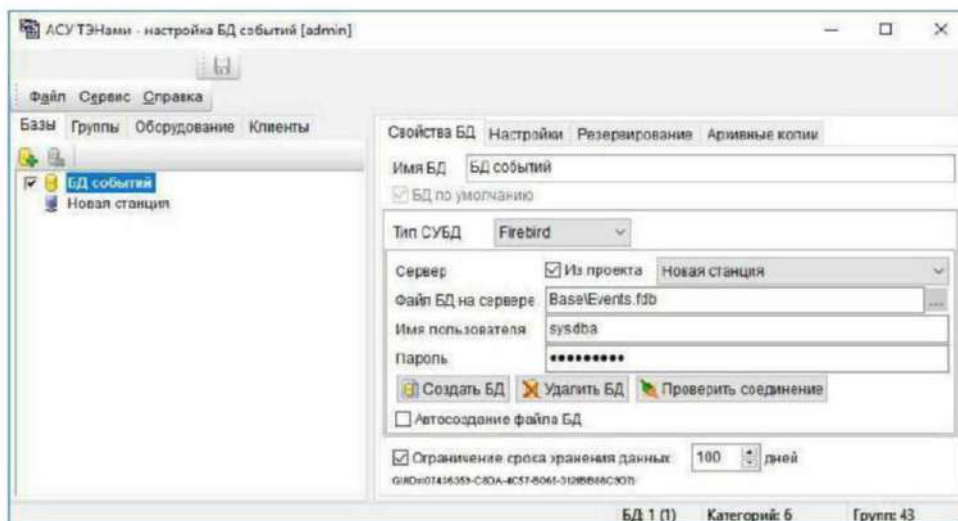


Рис. 4.1. Настройка БД событий

В верхней части окна программы находится главное меню и кнопка сохранения настроек. Слева - панель, состоящая из нескольких закладок для настройки баз, групп и клиентов. Справа - панели настройки, соответствующие выделенным элементам слева. В нижней части - строка состояния, в которой отображаются подсказки, информация об общем количестве и количестве выбранных БД, о количестве категорий и групп событий.

Правая часть состоит из четырех вкладок - **Свойства БД**, **Настройки**, **Резервирование** и **Архивные копии**.

На вкладке **Свойства БД** можно задать имя базы данных событий. Опция *БД по умолчанию* имеет смысл при наличии нескольких БД событий в проекте. Назначение кнопок **Создать файл БД**, **Удалить файл БД** и **Проверить соединение** понятно из их названия. В левом нижнем углу отображается GUID базы данных - её уникальный идентификатор.

Вкладка **Настройки** позволяет указать настройки ведения базы данных событий (рис. 4.2).

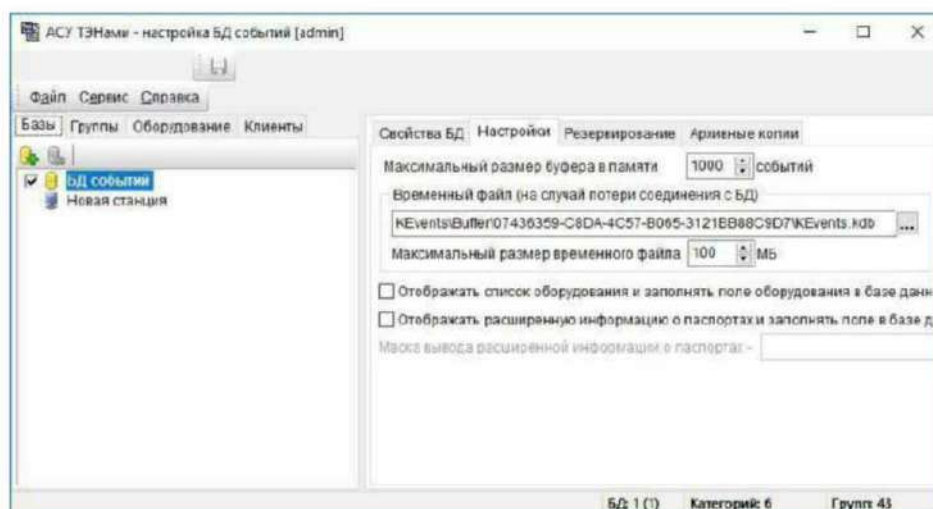


Рис. 4.2. Настройки ведения БД событий

События пишутся библиотекой регистрации событий сначала в буфер, затем - в файл БД. Буфер может разрастаться в случае потери соединения с БД. Для его ограничения существует настройка

Максимальный размер буфера в памяти - это максимальное количество событий в буфере - 1000 событий по умолчанию. В случае потери соединения с базой событий или с резервной БД (если она есть), создаётся **Временный файл**, в который пишутся события до момента восстановления связи с БД, после чего данные из **временного файла** переносятся в БД, а файл удаляется. Если соединение с базой событий не удаётся восстановить длительное время, временный файл может разрастаться. Для ограничения его размера введена настройка **Максимальный размер временного файла** (по умолчанию равен 100 мегабайтам).

Вкладка **Резервирование**. В нашем проекте Резервирование не требуется, поэтому галочка на этом пункте не установлена. Если сделать этот пункт активным, то будет вестись резервная БД событий.

Вкладка **Архивные копии**. Для нашего задания архивные копии также не требуются.

Эта вкладка служит для настройки автоматического создания архивных копий БД, целью которого является длительное хранение информации о ходе технологического процесса. Модуль создания архивных копий баз данных позволяет автоматически создавать копии БД событий, содержащие данные только за определенные пользователем интервалы времени (день, неделя, месяц, квартал, полугодие, год).

Чтобы просмотреть статистику по БД, нужно выбрать соответствующий пункт в контекстном меню списка баз данных событий (рис. 4.3).

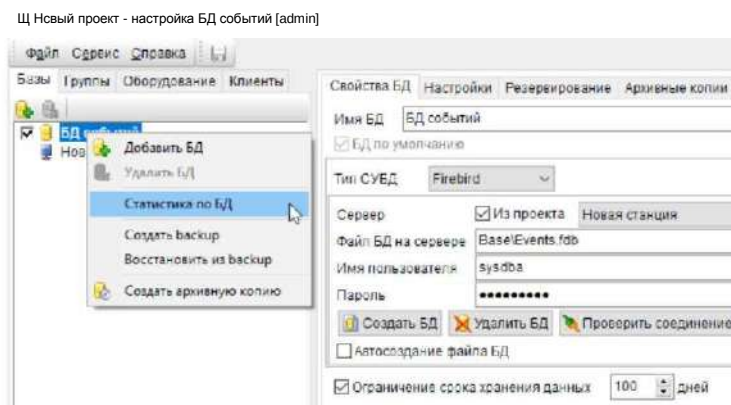


Рис. 4.3. Контекстное меню БД событий

Также с помощью контекстного меню можно создавать резервные копии (backup) базы данных, они используются для аварийного восстановления базы данных в случае возникновения проблем с файлом БД, для реального уменьшения размера файла БД после очистки.

Все события пишутся в свои группы, которые можно просмотреть во вкладке **Группы** (рис. 4.4).

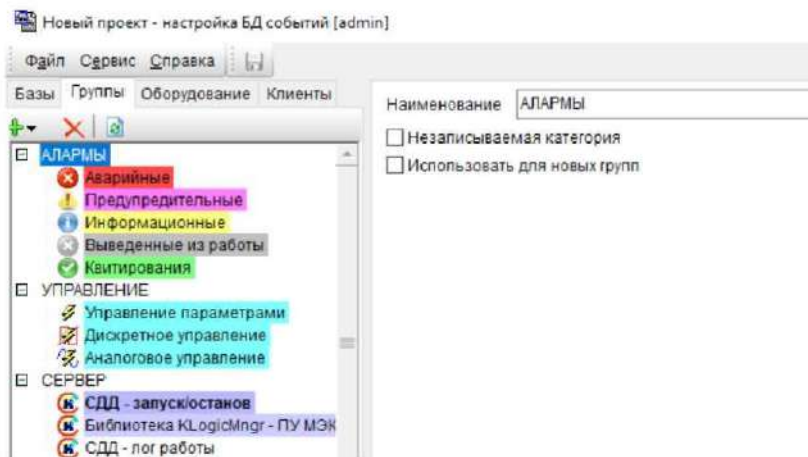


Рис. 4.4. Группы событий

Группы можно объединять в категории, количество категорий и групп не ограничено. Можно переименовывать группы и категории, менять их местами.

Вкладка **Клиенты** содержит список клиентов, т.е. станций, осуществляющих запись событий в БД (рис. 4.5).

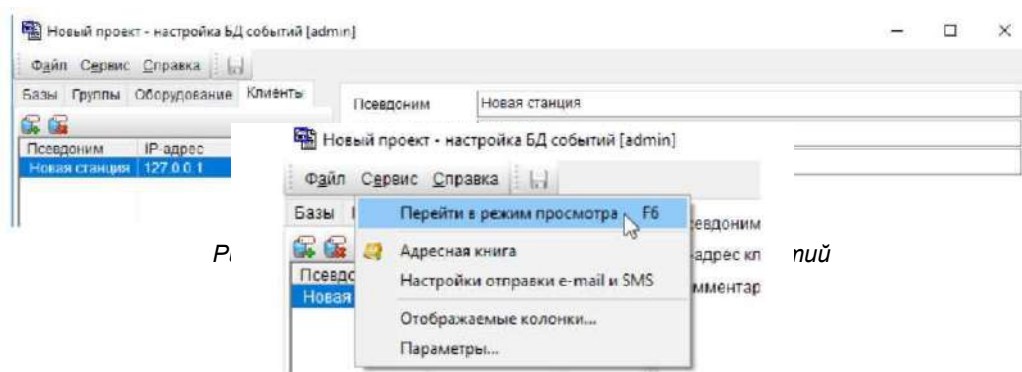


Рис. 4.6. Переход в режим просмотра

Для просмотра событий переходим в соответствующий режим (практически во всех модулях SCADA-системы КАСКАД для перехода между режимами просмотра и настройки используется «горячая» клавиша F6, также можно использовать соответствующий пункт меню Сервис), как показано на рис. 4.6:

Для нашего проекта нет ни одного события, поэтому список пустой. Список событий можно распечатать, для предварительного просмотра используется соответствующий пункт меню Файл (рис. 4.7).

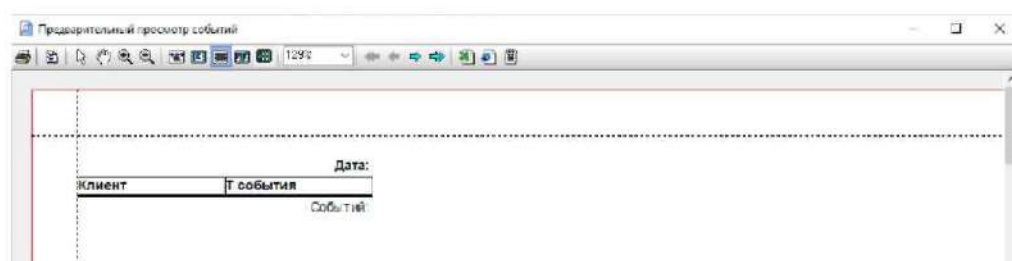


Рис. 4.7. Предварительный просмотр событий.

5. Настройка контроллеров

Общие сведения

Настройка контроллеров производится с помощью системы программирования контроллеров (СПК) KLogic, которая является инструментом технологического программирования контроллеров в составе автоматизированных систем управления технологическими процессами, оперативно-диспетчерского управления, систем коммерческого и технического учета ресурсов.

KLogic состоит из исполнительной системы (ИС), выполняющейся в контроллере, и инструментальной среды разработки (KLogic IDE), функционирующей на платформе Win32. Исполнительная система KLogic совместно с конфигурацией, создаваемой инструментальной средой, представляет собой целевую задачу, записываемую в контроллер. В конфигурацию входят технологические программы пользователя и параметры функционирования всех задач. Создание технологических программ, загрузка конфигурации в контроллер, отладка и мониторинг производятся из инструментальной среды разработки.

Инструментальная система KLogic можно работать отдельно — при запуске исполняемого файла KLogic.exe через проводник (или другой файловый менеджер), либо в составе проекта — при запуске через Конфигуратор.

Главное окно инструментальной системы **KLogic IDE** при запуске из проекта, т.е. через Конфигуратор (пункт Настройка контроллеров на вкладке Настройка проекта) представлено на рис. 5.1.

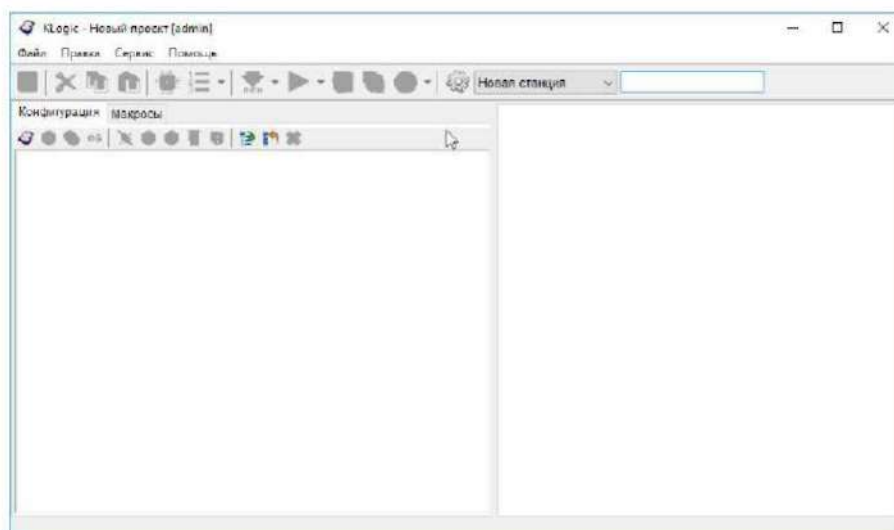


Рис. 5.1. Главное окно KLogic IDE

Окно приложения состоит из следующих элементов: строка заголовка, строка меню, стандартная панель инструментов. В левой части окна находится дерево конфигурации проекта с собственной панелью инструментов (наверху). В правой части окна — панель отображения и настройки свойств выбранного объекта конфигурации. Также в правой части рабочего окна на отдельной вкладке располагается графический редактор ФБД — функциональных блоковых диаграмм.

При выборе различных элементов дерева конфигурации, соответственно в правой части экрана происходит смена панелей для отображения свойств выбранных элементов.

В дереве конфигурации имеется контекстное меню, пункты которого становятся доступны в зависимости от того, на каком элементе дерева было вызвано контекстное меню.

Корневыми элементами дерева являются **Контроллеры**. В проекте одновременно может присутствовать любое количество конфигураций контроллеров. В каждом контроллере присутствуют группы **Задачи пользователя**, **Каналы ввода/вывода**, **Архивы**. В группе задач пользователя создаются алгоритмы, которые будут обрабатываться исполнительной системой, в группе каналов ввода/вывода описываются внешние сигналы, которые будут запрашиваться/устанавливаться исполнительной системой, в группе архивов производится настройка архивов, которые будет вести контроллер.

Добавление контроллера

Для решения поставленной задачи необходимо добавить контроллер. Это можно сделать

О

с помощью соответствующей кнопки ^ на панели инструментов конфигурации. В выпадающем меню выбираем нужный тип контроллера (рис. 5.2), в нашем случае **None-target** — виртуальный контроллер, работающий внутри сервера, он запускается автоматически при старте **Сервера доступа к данным** и является его составной частью.

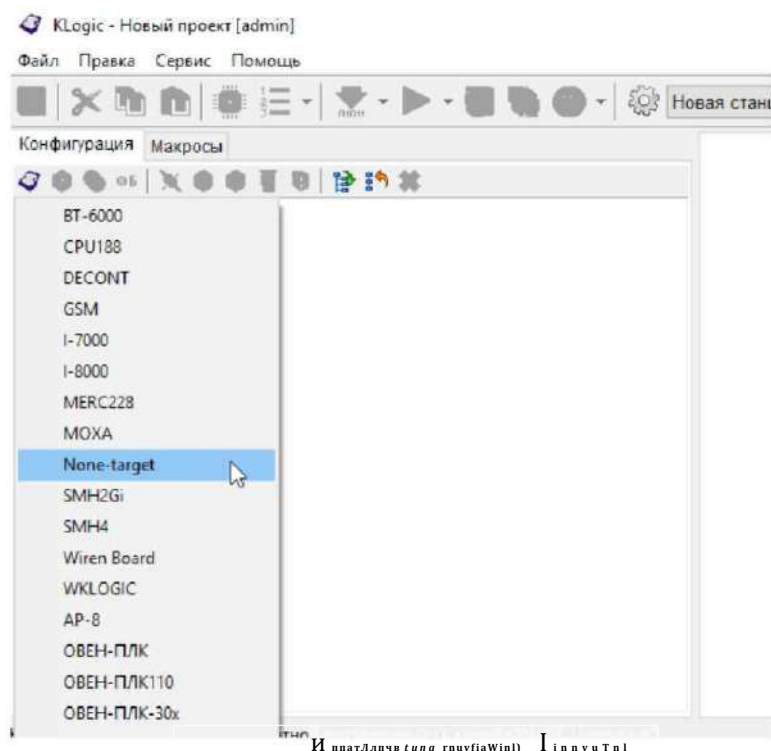


Рис. 5.2. Добавление контроллера

После добавления контроллера необходимо его настроить. Для этого предназначена панель свойств, которая появляется в правой части экрана при выделении контроллера в дереве проекта. На панели свойств есть вкладка «Связь с контроллером», содержащая одну или несколько панелей (рис. 5.3).

В панели **Общие настройки контроллера** можно сменить имя контроллера, выбрать его модификацию

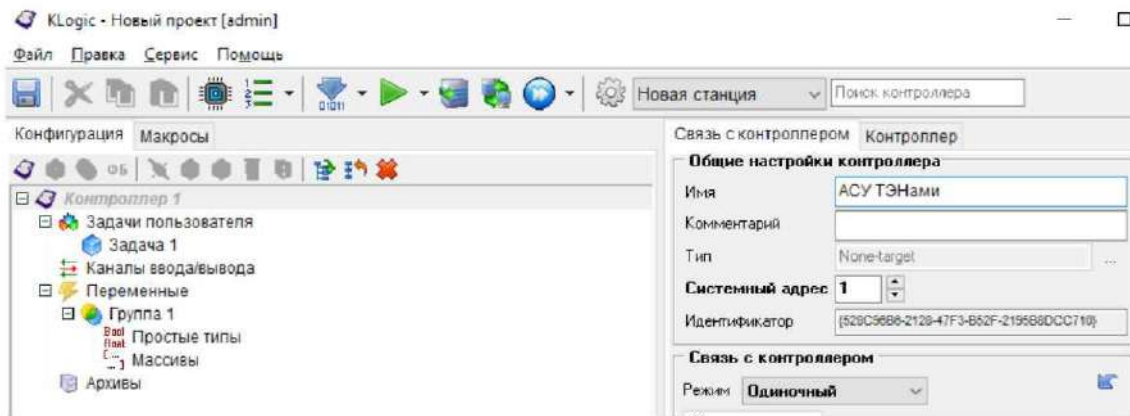


Рис. 5.3. Общие настройки контроллера

и просмотреть описание. Зададим имя контроллера АСУТЭНами.

Примечание. Для других (реальных, «железных») контроллеров на вкладке Связь с контроллером есть дополнительные панели, позволяющие настроить, например, опции связи с контроллером.

Теперь можно сохранить проект с помощью кнопки на панели инструментов, пункта меню Файл/Сохранить, или сочетания клавиш Ctrl+S.

Необходимо иметь в виду, что конфигурация KLogic состоит из нескольких файлов: общего файла проекта Имя_проекта.КЮ и файлов конфигурации контроллеров {GUID}.xml, где GUID - уникальный идентификатор контроллера. Поэтому при работе с инструментальной системой KLogic вне составе проекта (при запуске исполняемого файла KLogic.exe через проводник) рекомендуется создавать отдельную папку для хранения конфигурации KLogic.

Добавление и настройка ФБ

Группа **Задачи пользователя** дерева конфигурации KLogic является контейнером для отдельных задач пользователя, которые, в свою очередь, содержат группы и функциональные блоки. На них строятся алгоритмы, которые будут обрабатываться исполнительной системой.

Подробнее. Задача пользователя является контейнером для функциональных блоков — ФБ. Функциональные блоки могут находиться непосредственно внутри задачи пользователя, или в подгруппе. Сам элемент "Задача" можно рассматривать как корневую группу ФБ. Группы используются, в основном, для формирования удобных для восприятия пользователем, связанных между собой, групп функциональных блоков (их можно воспринимать как подпрограммы).

Ветка "Задачи пользователя" создается автоматически при добавлении нового контроллера в конфигурацию и не может быть удалена. По умолчанию эта ветка уже содержит одну пустую задачу под названием «Задача 1».

При необходимости можно добавить ещё одну или несколько задач (с помощью контекстного меню группы «Задачи пользователя» или соответствующей кнопки © на панели инструментов конфигурации). Исполнительная система позволяет одновременно исполнять много задач пользователя. Каждая задача выполняется циклически, с заданным приоритетом. Все задачи выполняются параллельно.

На вкладке свойств задачи пользователя можно настроить различные параметры задач, а также работать со списками функциональных блоков и групп. Окно настройки Задачи пользователя представлено на рис. 5.4.

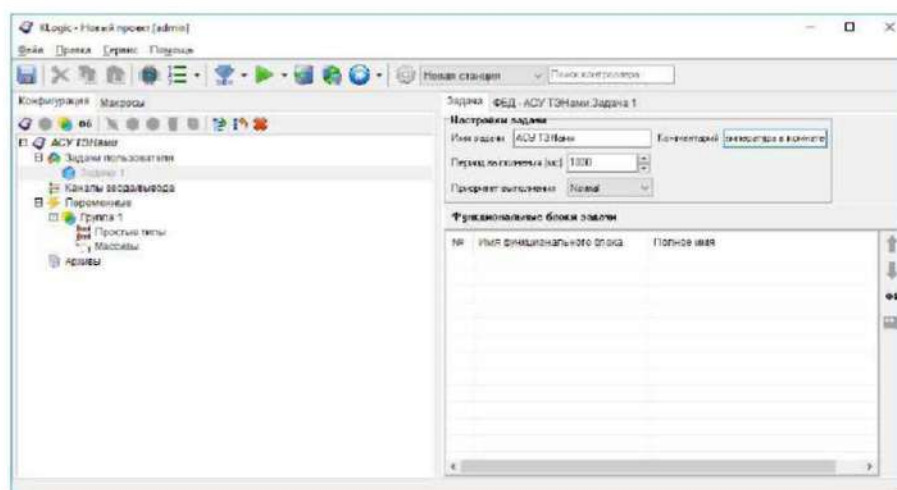


Рис. 5.4. Окно настройки задачи пользователя

Панель **"Настройки задачи"** содержит в себе основные свойства Задачи пользователя — имя, период и приоритет выполнения. Зададим имя задачи — **АСУ ТЭНами** и Комментарий — **Температура в комнатах**.

Таблица **"Функциональные блоки задачи"** содержит в себе список ФБ и групп, расположенных в этой задаче. Двойной щелчок по любому из объектов, расположенных в этом списке приведет к переходу на свойства этого объекта. Правее списка ФБ (и групп) располагаются управляющие этим

списком кнопки: перемещение ↑ вверх и ↓ вниз по списку текущего выделенного объекта,

ФБ добавление нового ФБ и -удаление текущего выделенного объекта. Мы пока ещё не добавили ни одного функционального блока, поэтому список **Функциональные блоки задачи** пустой.

Вкладка **ФБД - АСУ ТЭНами. Задача 1** — является редактором функциональных блоковых диаграмм (ФБД). Это графическое представление находящихся в данной группе функциональных блоков. Редактор позволяет создавать алгоритм управления на базе ФБ в удобном виде, с графическим представлением как самих ФБ, так и связей между ними.

Для решения поставленной задачи составим предварительно её алгоритм: при достижении значения температуры ниже плюс 5°C должен формироваться сигнал, включающий обогреватели (в данном случае ТЭН), при достижении температуры плюс 15 °C ТЭНы должны выключиться.

Для реализации данного алгоритма потребуется несколько функциональных блоков. Так, для получения текущего значения температуры в комнате согласно заданию используется ФБ **ДемоСигнал** в режиме синусоиды. Для сравнения текущей температуры с заданными уставками 5 и 15°C используется ФБ

Сравнение, используемый для сравнения одной или нескольких пар значений. Кроме того, понадобится ФБ *RS-триггер*, который применяется для запоминания дискретных сигналов. В нашем случае RS-триггер будет выдавать дискретный управляющий сигнал в зависимости от значений на его входах.

Добавить функциональные блоки можно несколькими способами:

1. Кнопка ФБ Добавить объект на панели справа таблицы "Функциональные блоки задачи" (рис. 5.5).

Рис. 5.5. Добавление ФБ с помощью кнопки на вкладке *Задача* Выбираем

ФБ *RS-Триггер* из группы *Обработка сигналов*.

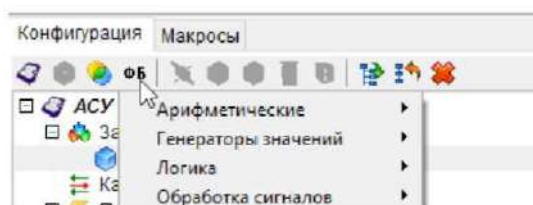


Рис. 5.6. Добавление ФБ с помощью панели инструментов конфигурации

3. Контекстное меню задачи в дереве конфигурации.

В списке выбираем ФБ *ДемоСигнал* из группы *Генераторы значений* (рис. 5.7).

2. Кнопка в"* Добавить функциональный блок на панели инструментов конфигурации (рис. 5.6).

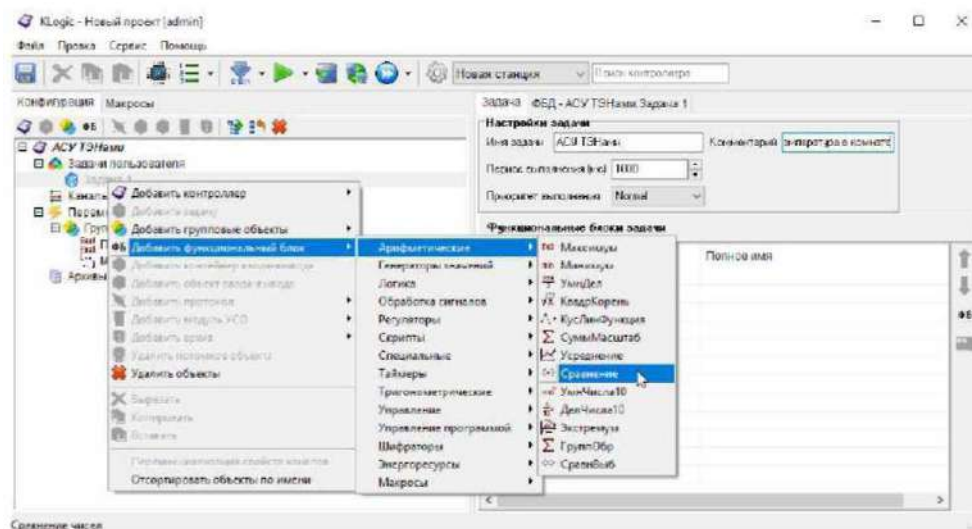


Рис. 5.7. Добавление ФБ с помощью контекстного меню задачи

4. ИзФБД.

Правой кнопкой мыши кликаем на поле редактора ФБД - Вставить объект-> Функциональный блок -> Арифметические -> Сравнение.

По умолчанию ФБ «Сравнение» имеет всего один канал для сравнения двух значений. Нам нужно сравнивать две пары значений, поэтому необходимо либо два одноканальных ФБ Сравнение, либо один двухканальный. Для добавления второго канала необходимо выделить ФБ в дереве конфигурации и в окне свойств ФБ на панели справа списка каналов нажать кнопку Добавить канал ^ВД. В результате получаем следующее (см. рисунки 5.8, 5.9).

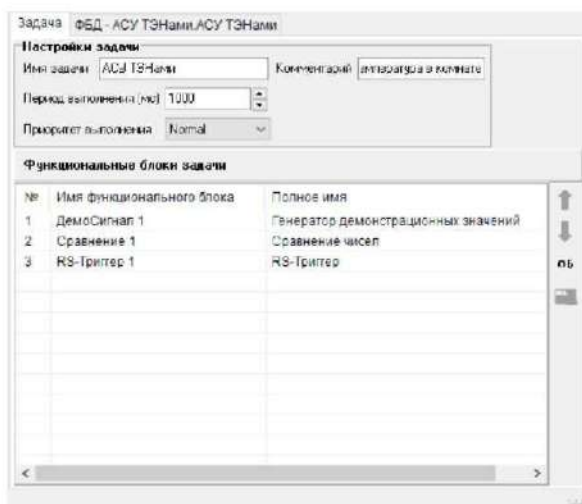


Рис. 5.8. Свойства задачи

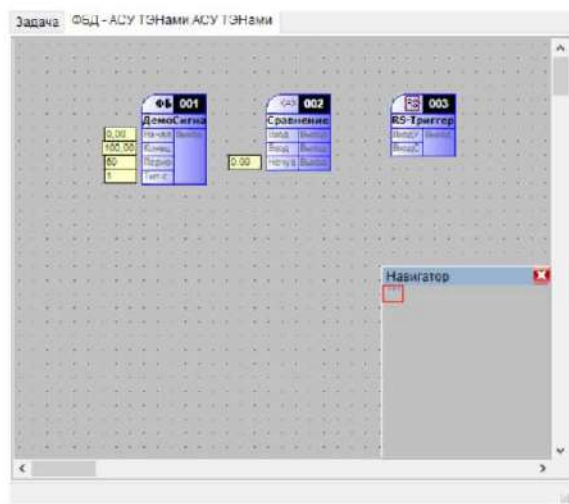


Рис. 5.9. ФБД

При желании можно расставить функциональные блоки на ФБД по своему усмотрению (просто перетаскивая их мышью за заголовок).

Теперь необходимо настроить добавленные функциональные блоки.

Каждый ФБ имеет входы и выходы. У каждого входа/выхода есть свои изменяемые настройки, такие, как шифр, начальное значение, единица измерения.

Для настройки параметров ФБ необходимо соответствующим входам присвоить начальные значения, которые будут присвоены входам сразу при включении контроллера.

К примеру, окно настройки ФБ ДемоСигнал представлено на рис. 5.10.

ФБ ДемоСигнал 1 ФБД - АСУ ТЭНами.АСУ ТЭНами

Общие настройки

Имя функционального блока:

Тип ФБ: (18) Генератор демонстрационных значений

Текущий ФБ: АСУ ТЭНами.АСУ ТЭНами.ДемоСигнал 1

Количество каналов: 1 Минимальное количество каналов: 1

Имя канала:

Объекты в составе функционального блока

№	Шифр	Тип	№ К...	Наименование	Подключений	Нач.значение
1	Начал...	вещественный вход		Начало диапазона значени...	0	0.00
2	Конеч...	вещественный вход		Конец диапазона значений...	0	50.00
3	Перио...	целочисленный вход		Период диапазона значени...	0	120
4	Тип сиг...	целочисленный вход, ка...	1	=1-синусоида,=2-пила,=3-ш...	0	1
5	Выход 1	вещественный выход, ка...	1	Значение демо сигнала	0	Неопределено

Рис. 5.10. Окно настройки ФБ ДемоСигнал

Согласно условиям, для задания температуры в комнатах нужно использовать **ФБ ДемоСигнал** в режиме синусоиды с периодом в 2 минуты и амплитудой 0.50.

Тип сигнала синусоида (значение 1 соответствующего входа) для данного функционального блока уже установлен по умолчанию.

Для входа «Период (сек)» необходимо задать начальное значение 120 (т.е. 2 минуты), для входа «Начало диапазона» - 0,0, для входа «Конец диапазона» - 50,0.

Для задания начального значения есть несколько способов. Можно выбрать канал в списке и на панели справа нажать кнопку **Определить начальное значение**. В появившемся окне ввести нужное значение (рис. 5.11) и нажать **ОК**.

Вещественное значение

Начало диапазона

Рис. 5.11. Ввод начального значения с помощью диалогового окна

Кроме того, можно воспользоваться окном свойств канала (открывающееся по двойному щелчку по строке в списке либо при выделении нужного канала в дереве конфигурации), которое выглядят следующим образом (см. рис. 5.12).

В свойствах следует поставить галочку напротив "Включить в карту адресов" для всех каналов для отображения в последующем в карте адресов (и для всех последующих ФБ).

АСУ ТЭНами ДемоСигнал 1 Период (сек) ФБД - АСУ ТЭНами АСУ ТЭНами

Общие настройки	
Шифр пользовательский	
Шифр	Период [сек]
Наименование пользовательское	
Наименование	Период, диапазон, значимый сигнал
Единица измерения	
Тип параметра	Целочисленный код
Скрывать в иерархической памяти	☐ Нет
Номер выхода	
Инвертировать значение	☐ Нет
Начальное значение	
Использовать	☒ Да
Целое	50
Адрес параметра	
Включить в карту адресов	☒ Да
МЭК-адрес параметра	21
Пересылать параметр на ПУ	Панель управления и панель мониторинга
по оппорту (3 границы контрольной шлюза)	0 БС
значимые операции в контроллере	0 БС
Модифицировать параметр (функция 3)	4
Связи	
Обработка аналогового значения на сервере	
(1) Линейная обработка	
Коэффициент масштабирования	1
Коэффициент сдвига	0
(2) Зона нечувствительности	☐ Нет
в диапазоне от	0
и до	100
выставлять значение	0

Рис. 5.12. Свойства канала

Для задания начального значения с помощью этого окна необходимо в группе Общие настройки установить галочку (Да) в поле Начальное значение/Использовать, а в поле Целое ввести само значение. Значение температуры можно будет получить на канале [ДемоСигнал.Выход 1].

Рекомендуется заполнять пользовательский шифр и наименование используемых каналов, это упростит дальнейшую разработку программы пользователя. Например, для канала [ДемоСигнал.Выход 1] необходимо указать его реальное применение, то есть Шифр «t_Kl» и наименование «Температура в комнате 1», а также единицу измерения °C. Для входов обоих каналов ФБ Сравнение можно указать значения уставок, для выходных каналов удобно задать в качестве комментария (поле «Шифр пользовательский» в окне свойств канала) назначение выходов: условные обозначения "<", "=" и ">".

Для этого нужно заполнить соответствующие поля в том же окне свойств канала. Для задания шифра можно воспользоваться и окном свойств ФБ, содержащем список его каналов.

Для отображения шифра канала необходимо включить пункт «Пользовательский шифр» в контекстном меню ФБД, как показано на рис. 5.13.

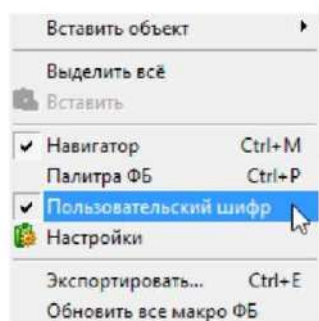


Рис. 5.13. Контекстное меню ФБД

В результате на ФБД получим следующее (рис. 5.14).

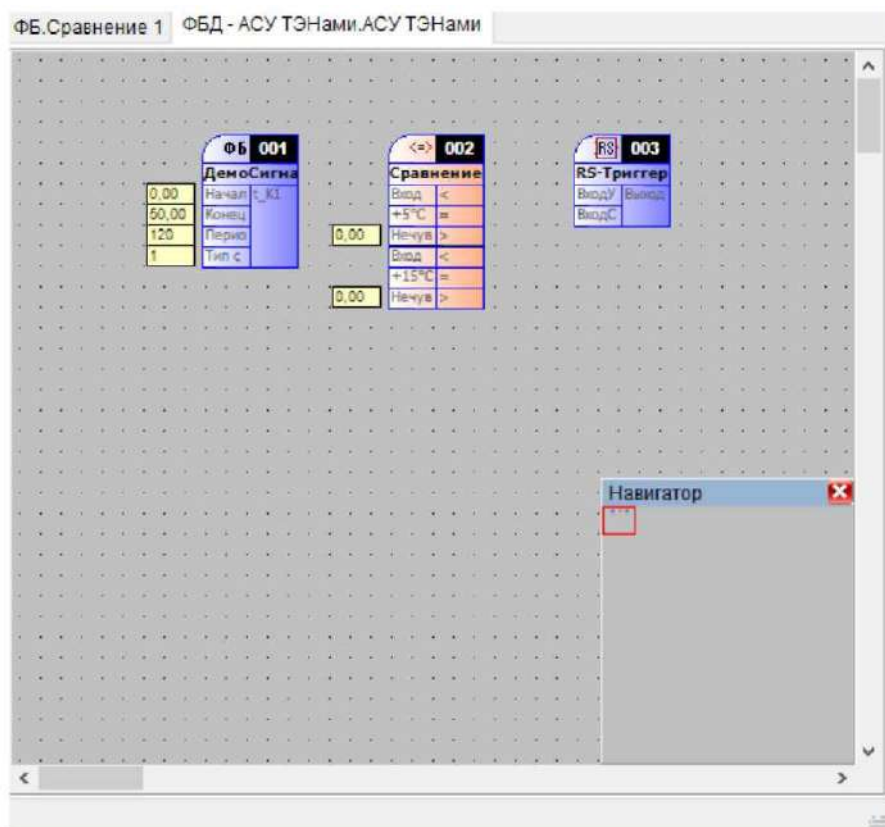


Рис. 5.14. Функциональные блоки на ФБД

Добавление протокола, модулей и каналов

Для связи с технологическим объектом контроля и управления используются различные устройства, такие как измерители, счетчики, модули ввода-вывода и др. *Протокол* описывает формат взаимодействия с устройством по коммуникационному порту, обычно оно строится на запросах контроллера к устройству и получении от него данных в двоичном формате. Проще говоря, протокол -это набор правил, регулирующих процесс приема/передачи данных.

Протоколы разделяются на две категории:

- протокол опроса внутренних модулей
- протокол опроса внешних модулей

Так как в проекте нет реального контроллера и модулей, можем использовать любой из них, например линейку модулей ICP DAS 1-7000 (в СПК KLogic для связи с ними используется протокол DCON

V

(ICOS, ADAM)). Добавляем этот протокол. Для этого можно использовать кнопку ^ на панели инструментов конфигурации либо контекстное меню строки Каналы ввода-вывода в дереве конфигурации (см. рис. 5.15).

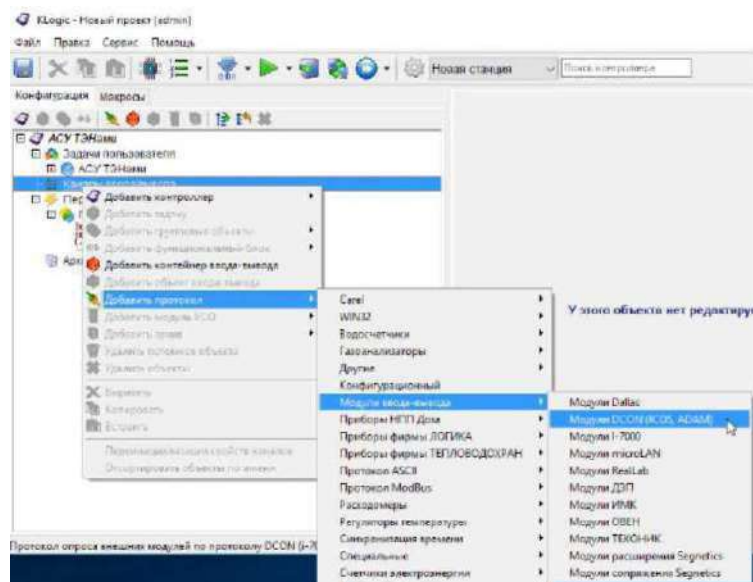


Рис. 5.15. Добавление протокола

Общие настройки протокола показаны на рис. 5.16.

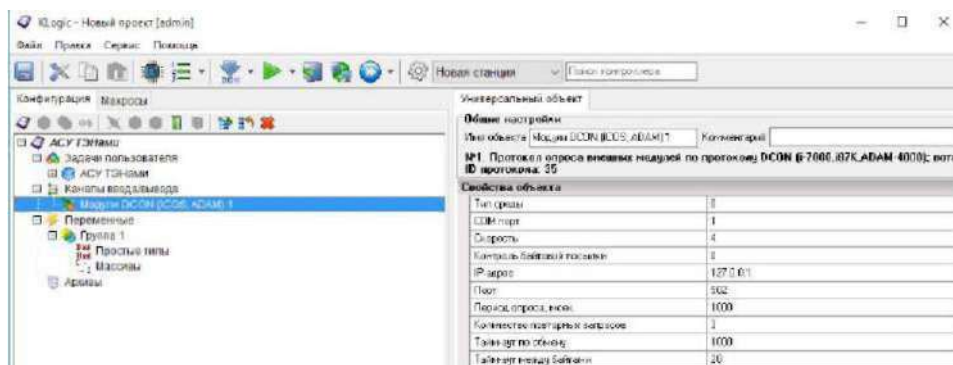


Рис. 5.16. Общие настройки протокола

При наведении курсора мыши на параметр настройки под редактором свойств появляется подсказка, объясняющая параметр. Например:

- **Тип среды** — тип среды обмена: 0 (RS-485) или 1 (Ethernet).
- **COM порт** — номер COM-порта контроллера, к которому подключены внешние модули.
- **Скорость** — скорость, бод (1 - 1200, 4 - 9600, 5 - 19200, 6 - 38400, 7 - 57600, 8 - 115200).
- **Контроль байтовой посылки** — контроль чётности (0 - нет, 1 - на нечётность, 2 - на чётность).
- и другие

Если бы в проекте использовались реальные модули, необходимо было задать эти параметры. Также при желании в этом окне можно сменить имя объекта и дописать к нему комментарий. Оставляем все параметры по умолчанию.

После настройки протокола добавляем непосредственно сами модули ввода-вывода.

Добавить модуль можно с помощью соответствующей кнопки " на панели инструментов конфигурации либо контекстного меню протокола в дереве конфигурации (см. рис. 5.17):

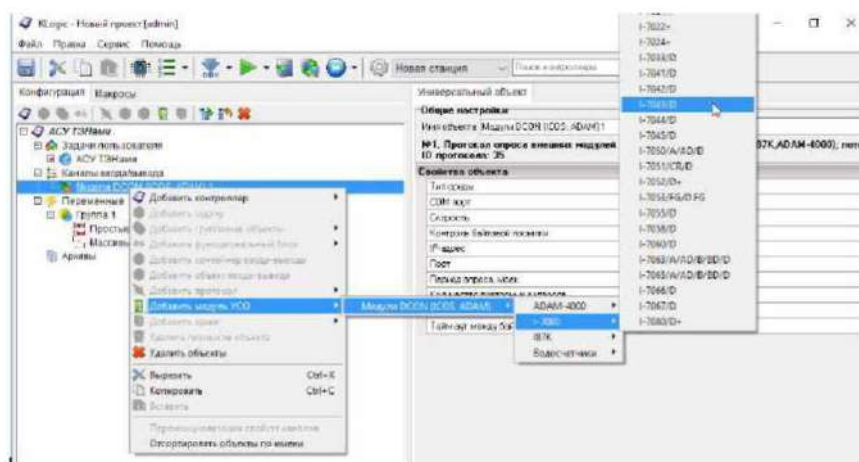


Рис. 5.17. Добавление модуля.

В нашем случае необходимо выдавать дискретный сигнал на включение электронагревателя, поэтому выбираем модуль дискретного вывода. В нашем случае можно выбрать любой модуль, например, I-7043/D- 16-ти канальный модуль дискретного вывода без изоляции.

Окно настройки модуля содержит в себе параметры модуля, подключенного по текущему протоколу — родительской ветке для текущего модуля.

Примечание. Список параметров настройки различных модулей отличаются.

У модуля I-7043/D есть несколько редактируемых свойств, которые в случае подключения реального модуля необходимо заполнить — это флаг использования CRC (контрольной суммы) и Адрес модуля (см.рис.5.18).

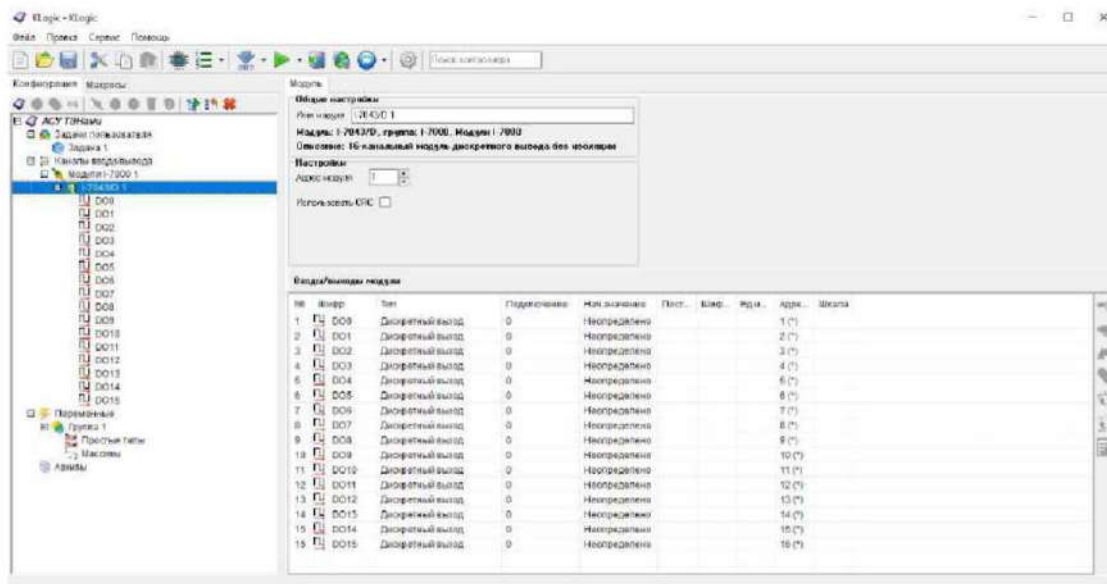


Рис. 5.18. Настройки i-7043/D

Как видно, у каждого канала модуля (как и у каналов ФБ) есть свои настройки: можно задать единицу измерения, шифр и имя, начальное значение, которое будет присвоено входу сразу при включении контроллера.

Для управления электронагревателями будем использовать дискретный выход DO0 модуля. Через окно свойств зададим ему пользовательский шифр «Вых_ТЭН1».

Дискретные каналы в дереве конфигурации и в списке каналов обозначаются значком ГЦ аналоговые — Л/.

Входы, т.е. входные каналы, из которых можно читать, обозначаются зелёной стрелкой влево <—. Выходы, т.е. каналы, в которых можно записывать значения, обозначаются красной стрелкой вправо

Связь каналов на ФБД

Редактор ФБД позволяет создавать алгоритм управления на базе ФБ в удобном виде, с графическим представлением как самих ФБ, так и связей между ними. Связи между функциональными блоками, находящимися внутри данной группы, отображаются в виде линий. Связи между ФБ, находящимися в других группах и задачах, и между модулями ввода-вывода, называются внешними, и отображаются маркером.

Между входами-выходами ФБ и каналами модулей в дереве проекта, а также входами-выходами других ФБ, отображенных на поле редактора ФБД, возможна установка связей с помощью механизма drag-n-drop — мышкой захватывается выход одного ФБ и перетаскивается на вход другого ФБ.

Создание связей между входами-выходами элементов дерева проекта производится аналогично.

Примечание 1. При создании связей между ФБ в пределах редактора ФБД кнопку мыши всё время держать нажатой не нужно, т.к. редактор ФБД запоминает состояние мыши, т.е. подвели курсор мыши к выходу одного ФБ — щёлкнули один раз, подвели курсор ко входу другого ФБ — щёлкнули ещё раз — связь готова.

Примечание 2. Для создания связи между элементом дерева и ФБД связь необходимо начинать с дерева.

Теперь необходимо создать алгоритм управления на базе добавленных функциональных блоков (ФБ).

Для сравнения текущей температуры в комнате с уставками +5 и +15 градусов используется двухканальный ФБ «Сравнение». Первый канал нужен для сравнения температуры в комнате с заданной уставкой плюс 5 градусов. То есть, если на первый вход ФБ Сравнение подать сигнал с выхода ФБ ДемоСигнал (значение температуры в комнате) и задать начальное значение второго входа, равное 5.0, то на соответствующем выходе (Меньше, Равно, Больше) первого канала ФБ будет сформирован результат сравнения в виде сигнала логической единицы (True).

Нажимаем на канал ФБ ДемоСигнал.Выход, появляется линия связи, соединяем с каналом [Сравнение.Вход 1 1], т.е. первым входом первого канала (см. рис. 5.19).

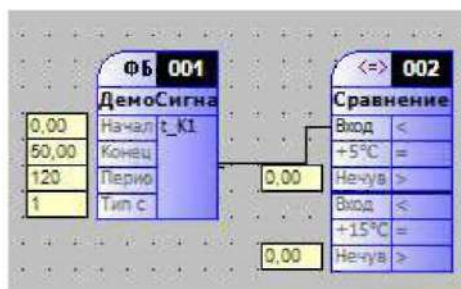


Рис. 5.19. Связь функциональных блоков на ФБД

Теперь необходимо задать уставку, т.е. величину, с которой будет сравниваться температура. То есть необходимо задать начальное значение для второго входа первого канала ФБ Сравнение. Конечно, можно воспользоваться списком входов-выходов ФБ Сравнение или же окном его свойств (см. выше). Но для задания начального значения можно воспользоваться и средствами редактора ФБД. Для этого в контекстном меню соответствующего входа ФБ (т.е. нужно нажать правой кнопкой мыши на втором входе ФБ Сравнение) выбрать команду начальное значение (см. рис. 5.20).

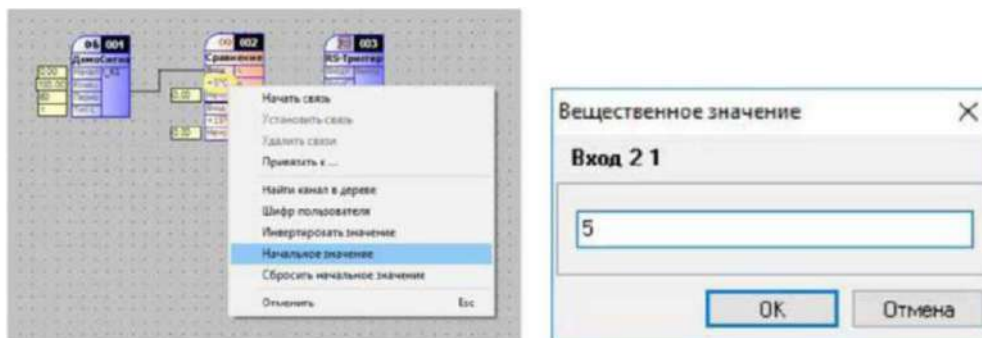


Рис. 5.20. Задание начального значения

Аналогично соединяем сигнал температуры с первым входом второго канала ФБ Сравнение и задаём уставку (плюс 15.0) второго канала.

Далее в зависимости от результата сравнения необходимо сформировать сигнал на включение/выключение нагревателя. По условию, при температуре ниже плюс 5 °С электронагреватели должны включиться, при этом температура начинает расти. При достижении температуры плюс 15 °С обогреватели должны выключиться. Далее температура в комнате будет снижаться, обогреватели при этом будут выключены вплоть до достижения температурой уставки плюс 5 градусов. То есть если температура находится в пределах от 5 до 15 градусов, состояние нагревателя не должно измениться. Вышеизложенное представлено в табл. 5.1.

Табл. 5.1. Таблица истинности

$t_{K1} < 5$	$t_{K1} > 15$	Выход	Пояснение
true	false	true	Если температура ниже 5 градусов, обогреватель включается
false	true	false	Если температура выше 15 градусов, обогреватель выключается
false	false	предыдущее состояние	При температуре от 5 до 15 градусов состояние нагревателя измениться не должно
true	true	—	невозможное состояние

Данная таблица истинности соответствует функциональному блоку «RS-триггер».

Передаём результаты сравнения температуры в комнате с уставками +5 и +15 градусов на входы ФБ RS-триггер, его выходной сигнал и будет использоваться для включения/отключения электронагревателя. Так как при понижении температуры ниже пяти градусов на выходе должен быть сигнал логической единицы, связываем канал [Сравнение.Выход Меньше 1] с входом установки (ВходУст) ФБ RS-триггер. Аналогично результат сравнения температуры с уставкой 15 (канал [Сравнение.Выход Больше 2]) подаём на вход сброса (ВходСброса). Таким образом, на выходе ФБ RS-триггер будет формироваться сигнал управления работой электронагревателя в автоматическом режиме работы системы.

Получаем следующее (см. рис. 5.21).

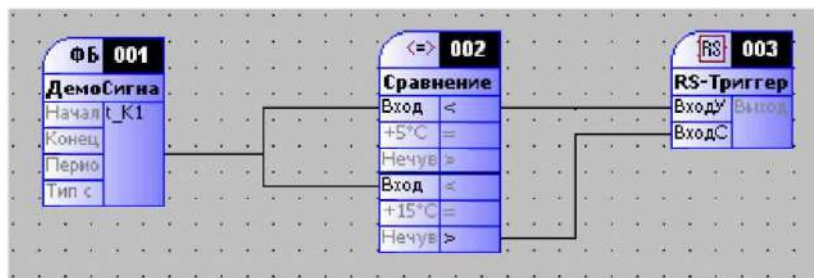


Рис. 5.21. Связанные функциональные блоки на ФБД

Для того чтобы реализовать ручной режим управления, нам понадобятся сигналы для включения автоматического режима и для управления нагревателями в ручном режиме.

Для задания таких сигналов нужен специальный ФБ, используемый для сохранения параметров и имеющий настраиваемый пользователем список переменных, которые затем можно связать с ключевыми или промежуточными значениями коэффициентов, параметров технологической программы и других значений. Таким ФБ является **Сохранение переменных**, находящийся в группе Специальные. Окно добавления канала (переменной) этого ФБ представлено на рис. 5.22.

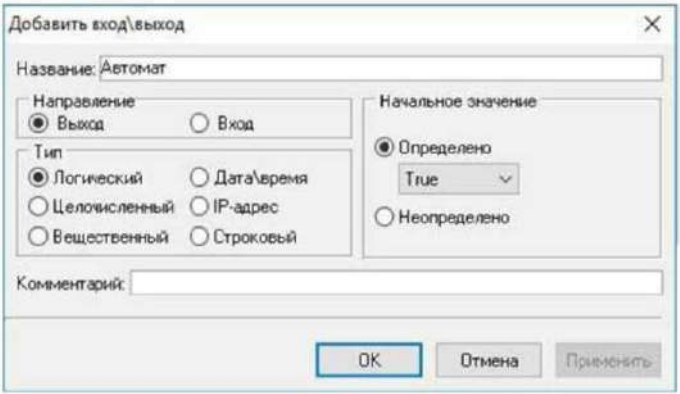


Рис. 5.22. Добавление каналов ФБ

Как видно, для переменной можно задать имя, тип, начальное значение и комментарий. Добавим два канала, т.е. две переменных логического (булевского) типа.

Табл.5.2. Список переменных

Имя	Направление	Тип	Нач. значение	Шифр	Назначение
Автомат	выход	логический	True	Авт	Включение автоматического режима
Включить	выход	логический	False	Упр_руч	Управление в ручном режиме

Канал Автомат (Авт) будем считать переключателем режима работы АСУ (ручной/автоматический), его значение по умолчанию True, то есть при запуске контроллера будет активен автоматический режим. Канал Включить (Упр_руч) будет управлять работой электронагревателя в ручном режиме.

В автоматическом режиме (Авт) выходной сигнал (Вых) должен зависеть только от выхода на RS-триггере (Упр_авт), в ручном режиме (сигнал НЕ Авт) — только от состояния канала Включить (Упр_руч).

Иными словами, состояние выхода (Вых) определяется [сигналом Упр_авт в автоматическом режиме (Авт)] ИЛИ [сигналом Упр_руч в ручном режиме (НЕ Авт)], поэтому логическую функцию можно записать следующим образом:

$$Вых = Авт \cdot Упр1 + \overline{Авт} \cdot Упр2$$

Поэтому необходимо добавить несколько ФБ, реализующих логику управления в ручном режиме и переключения между режимами. Так, в группе Логика находятся ФБ **Логическое И**, **Логическое ИЛИ** и **Логическое НЕ**.

Создаём связи между ФБ: ко входам одного ФБ «Логическое И» подключаем выходы Упр_авт и Авт, а ко входам другого ФБ «Логическое И» — выходы Авт (через ФБ Инвертор) и Упр_руч. Выходы обоих ФБ «Логическое И» подключаем ко входам ФБ «Логическое ИЛИ».

Для передачи выходного сигнала на включение/отключение электронагревателя, связываем канал [ИЛИ_1.Выход] с выбранным ранее дискретным выходом D00 добавленного ранее модуля 1-7043. Для этого нужно раскрыть дерево конфигурации так, чтобы в нем были видны каналы модуля и создать связь с помощью механизма drag-n-drop, т.е. нажать левую кнопку мыши на нужном канале модуля и не отпуская, довести курсор до канала Выход ФБ Переключатель с ДУ. Связь отображаются маркером в виде ярлыка, идущего от выхода ФБ.

В процессе работы все ФБ внутри одной задачи пользователя выполняются последовательно, сверху вниз в обход по дереву проекта, включая ФБ, находящиеся в группах. При необходимости задания порядка выполнения ФБ это можно сделать с помощью списка задачи — либо с помощью кнопок на панели справа этого списка, либо с помощью перетаскивания ФБ мышью (Drag'n'Drop) в пределах списка, а также с помощью редактора ФБД, используя соответствующий пункт контекстного меню функционального блока.

Зададим порядок выполнения функциональных блоков с учётом того, что связанные ФБ должны выполняться последовательно, то есть ФБ, выходы которых связаны с входами других ФБ, должны выполняться раньше, и, соответственно, иметь номер по порядку меньше.

Так, для ФБ Сохранение переменных необходимо задать порядок №1, так как переменные [Авт] и [Упр_руч] являются управляющими, т.е. задающими режим работы всей системы.

В результате получаем следующее (см. рис. 5.23).

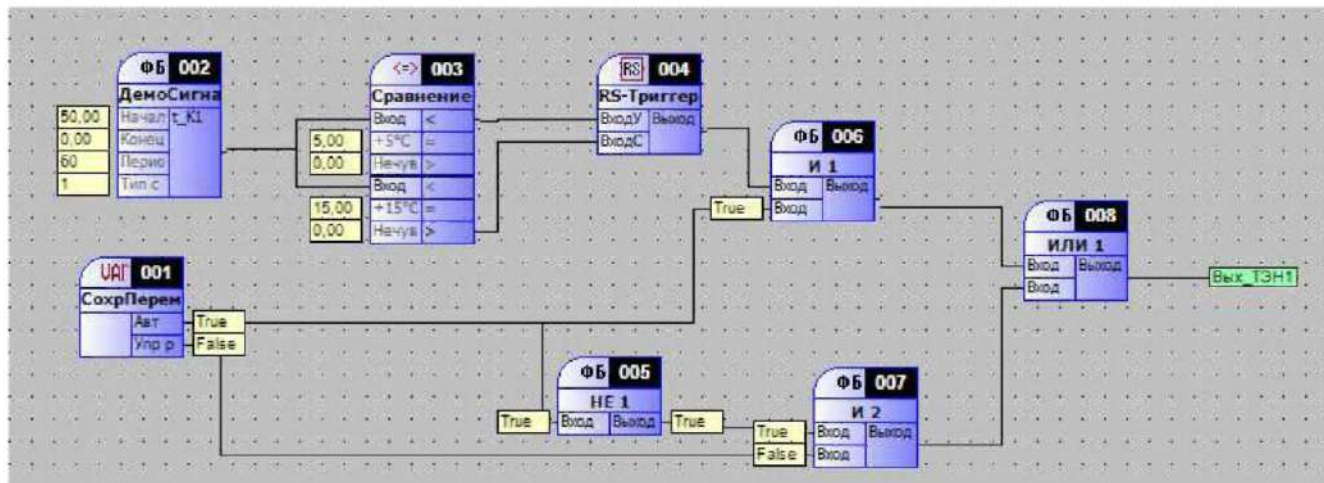


Рис. 5.23. Готовая технологическая программа

На этом написание технологической программы завершено. Данная программа реализует алгоритм, описанный в задании.

Отладка и построение конфигурации

Для проверки работы реализованной программы необходимо выполнить её отладку. Существует три режима отладки:

- без опроса удаленных модулей УСО,
- с опросом,
- режим глобальной отладки.

Первые два режима выбираются из выпадающего меню, вызываемого нажатием на стрелку вниз рядом с кнопкой отладки (см. рис. 5.25).

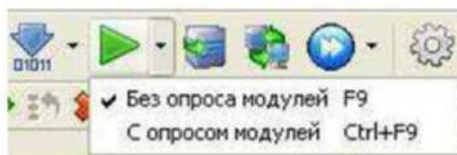


Рис. 5.25. Режимы отладки

По умолчанию используется режим *отладки без опроса модулей* («горячая» клавиша F9), представляющий собой эмуляцию работы исполнительной системы непосредственно внутри инструментальной системы. Часто этот режим применяется для проверки работы задач пользователя и отладки алгоритмов обработки информации, закладываемых в контроллер.

Отличие режима *отладки с опросом модулей* (сочетание клавиш Ctrl+F9) в том, что в этом режиме используется библиотека эмулятора KLogic, которая может реально опрашивать удаленные модули

ввода-вывода, подключенные к коммуникационным портам рабочей станции (т.е. непосредственно к компьютеру). Этот режим позволяет снимать реальные сигналы в случае наличия подключенного оборудования.

Выбранный режим запоминается, и используется при простом нажатии на кнопку отладки. Для первых двух видов отладки она запускается только для текущего контроллера конфигурации (выделенного в дереве, а если не выделено ничего, то первого).

Подробнее про другие виды отладки можно посмотреть в соответствующем разделе справочной системы.

При любом варианте отладки инструментальная система переходит в режим мониторинга входов-выходов ФБ и модулей. В режиме отладки все действия по работе с конфигурацией блокируются, также блокируются все действия панели инструментов приложения (кроме остановки отладки).

Проверяем работу алгоритма, запуская Отладку (F9) и наблюдая за значениями на каналах функциональных блоков (см. рис. 5.25).

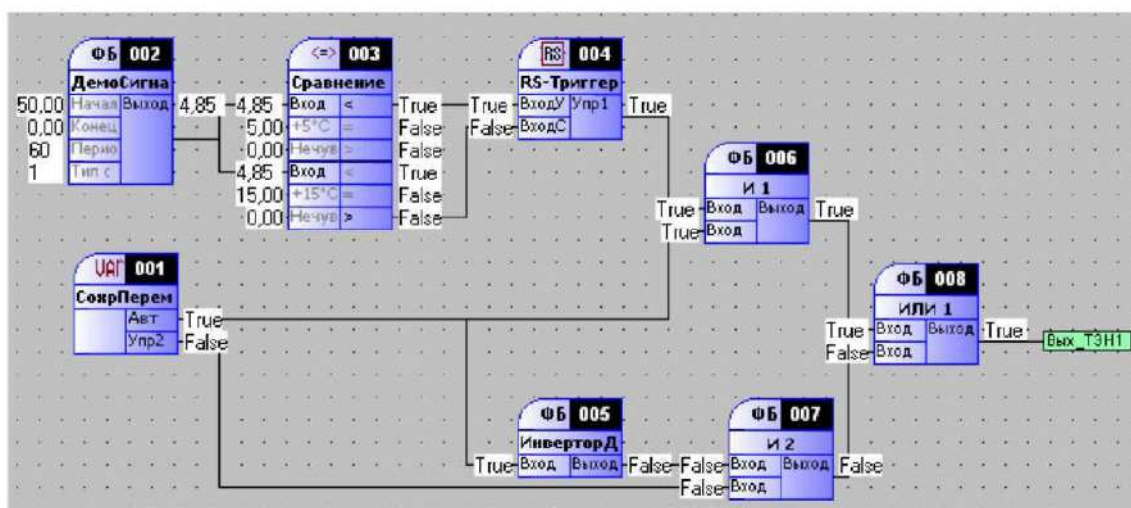


Рис. 5.25. ФБД в режиме отладки.

В автоматическом режиме изменение температуры на выходе ФБ «ДемоСигнал» сравнивается со значениями 5 и 15, на выходах ФБ «Сравнение» формируются дискретные значения, соответствующие результатам сравнения уставок температуры с текущим значением температуры, результаты сравнения попадают на ФБ «RS-триггер», вырабатывающий сигнал управления, который через ФБ «И_1» и «ИЛИ» попадает на дискретный выход модуля, связанный с электронагревателем.

В режиме отладки можно менять любые входные значения каналов функциональных блоков и наблюдать реакцию пользовательской программы на них.

Так, для перехода в ручной режим необходимо изменить состояние выхода Авт ФБ Сохранение переменных.

Записать значение в канал можно несколькими способами, но в данном случае удобнее всего это делать через ФБД:

1. Через контекстное меню поля значения входа/выхода функционального блока на ФБД
2. Нажав левой кнопкой мыши на поле значения канала ФБ

В любом случае выходит окно записи значения в канал, представленное на рис. 5.26.

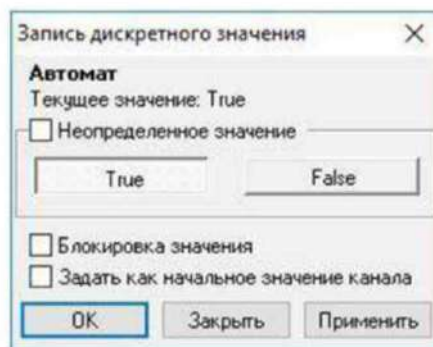


Рис. 5.26. Окно записи значения в канал

В окне отображается название (Вход 1 1), текущее значение канала (5,00), галочка для установки неопределённого значения, непосредственно поле для ввода значения, флаг установки блокировки, а также кнопки: ОК, Закреть, Применить. В заголовке окна отображается тип записываемого значения. Вид поля ввода зависит от типа значения (bool, int, float, string, DateTime). Например, для ввода булевских значений используется кнопки «Запись True» и «Запись False».

Форма не является модальной, т.е. не мешает работе с основным окном KLogic IDE. При переходе на другой параметр происходит обновление формы (текст запроса, тип, текущее значение, флаг блокировки).

Запись производится при нажатии кнопки «ОК» или «Применить», причём в первом случае окно закрывается, а во втором - остаётся. При остановке опроса либо при переходе на ветку дерева, не являющуюся каналом, окно скрывается.

Записав в канал [СохрПерем I.Abt] значение False, мы перейдём в ручной режим работы системы, при этом сигнал управления [СохрПерем.Упр_руч] через ФБ И_2 и ИЛИ проходит на дискретный выход модуля.

В KLogic существует возможность визуального контроля процесса. Для этого на ФБД с помощью контекстного меню можно добавить т.н. *Тренд*, который используется для отображения графиков изменений параметров в процессе отладки программы ФБД (см. рис. 5.28).



Рис. 5.28. Тренд.

Добавим отображение двух параметров: значение температуры в комнате (канал [ДемоСигнал.Выход]) и сигнал управления нагревателем (дискретный выход [Модуль 1-7043.DO0]):

С помощью настроек (см. рис. 5.28) можно изменять размер тренда, список отображаемых графиков и другие общие свойства тренда.

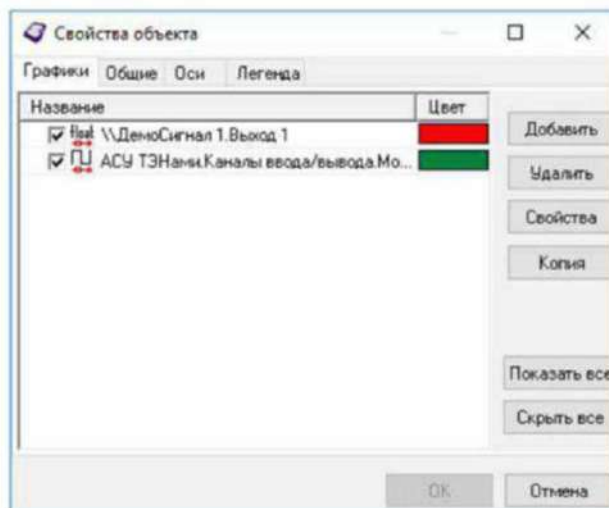


Рис. 5.28. Свойства тренда

В результате получаем график, представленный на рис. 5.29.

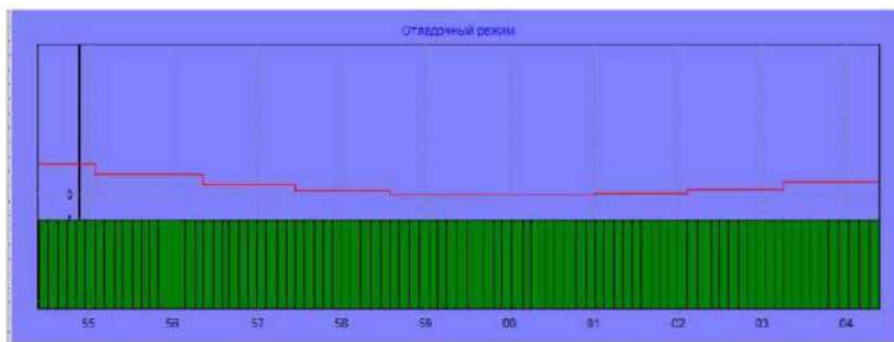


Рис. 5.29. Тренд в режиме отладки

Из графика, снятого в автоматическом режиме работы АСУ (для большей наглядности здесь уменьшена амплитуда ДемоСигнала до 20, а период до 30 сек) видно, что при понижении температуры ниже пяти градусов подаётся сигнал на включение нагревателя, при повышении температуры до плюс 15 нагреватель выключается.

Для проверки работы системы в ручном режиме надо записать в канал [СохрПерем.Автомат] логический ноль (False) и, изменяя значение переменной Упр_руч, наблюдать за изменением состояния дискретного выхода DO0 модуля 1-7043.

После того, как мы убедились, что система работает правильно, нажимаем кнопку . При этом создается конфигурация контроллера (см. рис. 5.30). Эта конфигурация будет использована для связи с верхним уровнем SCADA-системы.

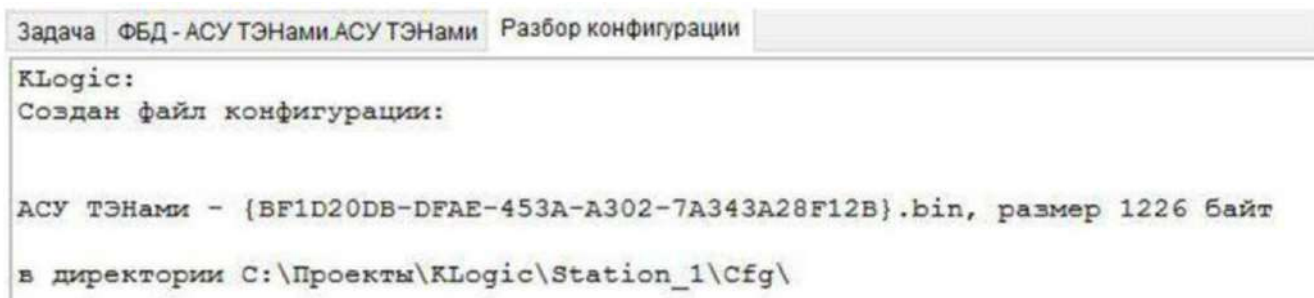


Рис. 5.30. Построение конфигурации.

6. Настройка виртуального контроллера

Для начала нужно установить службу виртуального контроллера, если вы не установили ее в процессе установки дистрибутива. Для этого через меню Пуск запустите утилиту управления службами

виртуального контроллера. Для корректной установки службы утилита должна быть запущена с правами администратора. После создания экземпляра службы и его запуска вы увидите ее текущий статус - Работает, а также сообщения в окне лога.

В конфигурации открывается настройка виртуального контроллера через Утилиты - Управление виртуальными контроллерами (см. рис. 6.1).

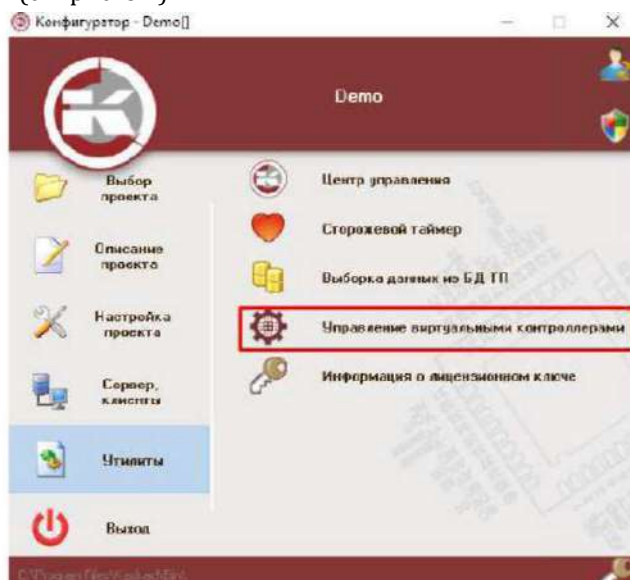


Рис. 6.1 - Управление виртуальными контроллерами При отсутствии лицензии служба способна опрашивать не более 32 тегов одного устройства. В KLogic у контроллера None-target, в качестве номера порта укажите UDP порт (см. рис. 6.2), который вы задали при создании экземпляра службы (по умолчанию это 30293, или 0x7655).

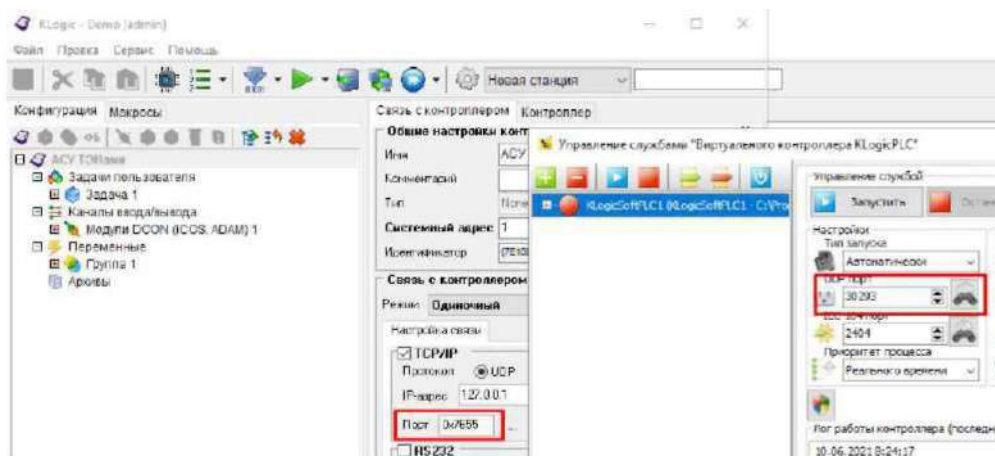


Рис.6.2-UDP порт

7. Настройка сервера доступа к данным

Сервер доступа к данным (СДД) - серверное приложение, предоставляющее централизованный механизм создания единого массива технологических параметров, их вторичной обработки по алгоритмам пользователя, ведения истории технологического процесса и обработки клиентских запросов. Сервер доступа к данным загружает соответствующие модули доступа к данным, модуль обработки паспортов и модуль регистрации технологических параметров, запускает механизмы обработки и архивации, а также предоставляет клиентским приложениям из состава SCADA-системы «КАСКАД» единый интерфейс для доступа к параметрам, формируемым в результате контроля над технологическим процессом.

В нашем случае настройка СДД не требуется, т.к. связь параметров конфигурации KLogic и других модулей SCADA-системы идёт напрямую, то есть следует просто включить СДД через Сервер,Клиенты -Сервер доступа к данным. После важных внесений изменений в других настройках, следует перезапускать СДД.

8. Настройка паспортов

Значение параметра, получаемое с контроллера или со специализированного устройства, пересчитывается в Сервере Доступа к Данным (СДД) в реальное значение параметра. Структуры, хранящие это значение, называются **паспортами**. Модуль настройки паспортов (см. рис. 8.1) предназначен для настройки паспортов различных типов для опроса и обработки данных с различных устройств. Все паспорта в едином проекте разбиты по станциям. По умолчанию выводятся паспорта той станции, которая назначена или автоматически определилась как текущая.

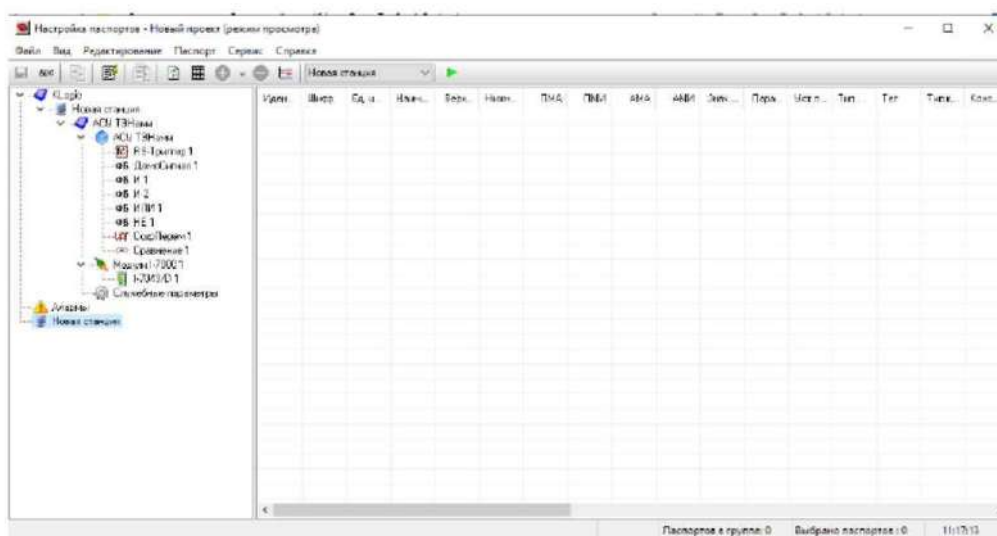


Рис. 8.1. Программа настройки паспортов

В данном проекте используется СПК KLogic, параметры конфигурации KLogic автоматически преобразуются в паспорта, поэтому дополнительной настройки не требуется - все параметры конфигурации будут сразу видны в диалоге выбора паспорта из любого клиентского приложения.

9. Регистрация технологических параметров

Информация об изменении во времени технологических параметров сохраняется в базах данных технологических параметров. Регистратор технологических параметров является одним из модулей обработки данных СДД и сохраняет информацию о ходе технологического процесса в базах данных (БД) технологических параметров. Для ведения БД используется SQL-сервер InterBase Server 6.

Важно, чтобы Firebird версии совпадали, рекомендуем везде использовать версию 3.0.

Основное окно программы настройки БД ТП имеет вид, представленный на рис. 9.1. Также, как и в других модулях, в заголовке отображается название проекта и имя пользователя, под которым произошел запуск программы.

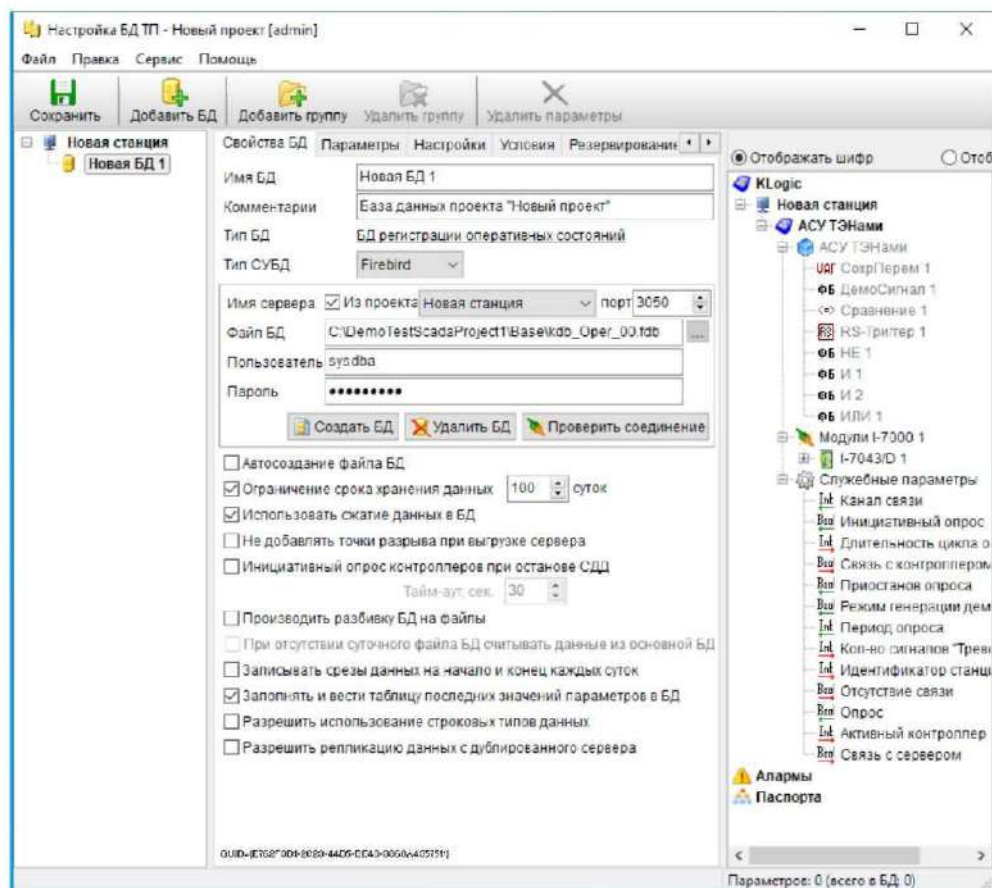


Рис. 9.1. Настройка БД ТП

Основное окно программы настройки БД ТП состоит из следующих частей:

- 1) В верхней части находится панель управления, содержащая главное меню и панель инструментов (кнопки на панели инструментов дублируют основные пункты меню);
- 2) В левой части — дерево баз данных проекта;
- 3) В средней части — несколько закладок с элементами редактирования для настройки выделенной в данный момент БД или группы параметров БД;
- 4) В правой части — дерево паспортов и тегов KLogic проекта;
- 5) В нижней части — строка состояния, в которой отображаются подсказки, а также информация о количестве параметров БД.

Добавление БД осуществляется с помощью кнопки . Также это можно сделать через меню Правка/ Добавить БД либо соответствующий пункт контекстном меню дерева баз данных. При этом появляется окно, предлагающее выбрать тип добавляемой БД (см. рис. 9.2).

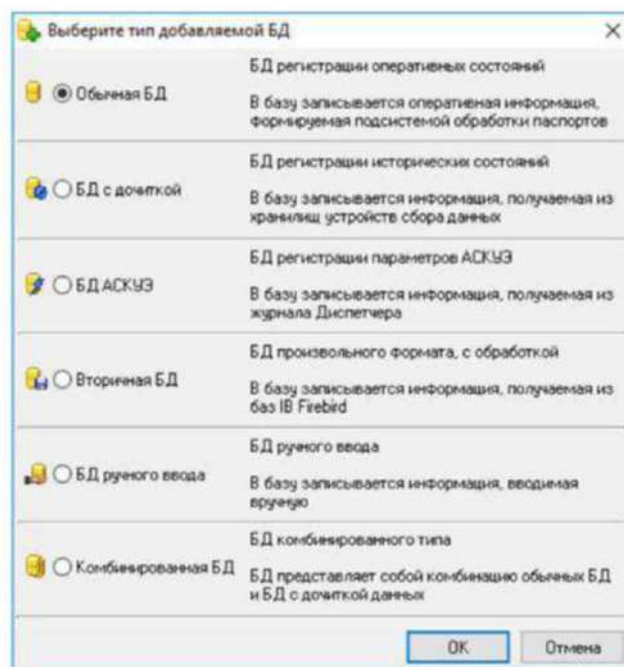


Рис. 9.2. Выбор типа БД

Существуют следующие виды БД:

1. **Обычная БД** (БД регистрации оперативных состояний). В базу записывается оперативная информация, формируемая подсистемой обработки паспортов (наиболее часто используемый тип БД);
2. **БД с дочеткой** (БД регистрации исторических состояний). В базу записывается информация, получаемая из хранилищ устройств сбора данных (данный тип БД может использоваться при условии, что контроллеры имеют собственные исторические архивы и в модуле доступа к данным для этих контроллеров реализовано чтение исторических данных);
3. **БД АСИУЭ** (БД регистрации параметров АСКУЭ). В базу записывается информация, получаемая из журнала Диспетчера;
4. **Вторичная БД** (БД произвольного типа, с обработкой). В базу записывается информация, получаемая из баз IB Firebird (первичных БД). Вторичные БД предназначены для хранения данных на серверах БД любого типа: Oracle, MS Access и т.д. В качестве их источников данных служат Обычные БД, БД с дочеткой и БД ручного ввода;
5. **БД ручного ввода**. В базу записывается информация, вводимая вручную (из объекта визуализации "Ручной ввод").

Примечание. Тип БД можно задать только для добавляемой БД, тип уже созданной БД изменить нельзя.

Создаем новую «Обычную БД», задаем имя и пишем комментарий.

Вкладка **«Свойства БД»**, представленная на рис. 9.3, даёт возможность редактировать основные свойства базы данных.

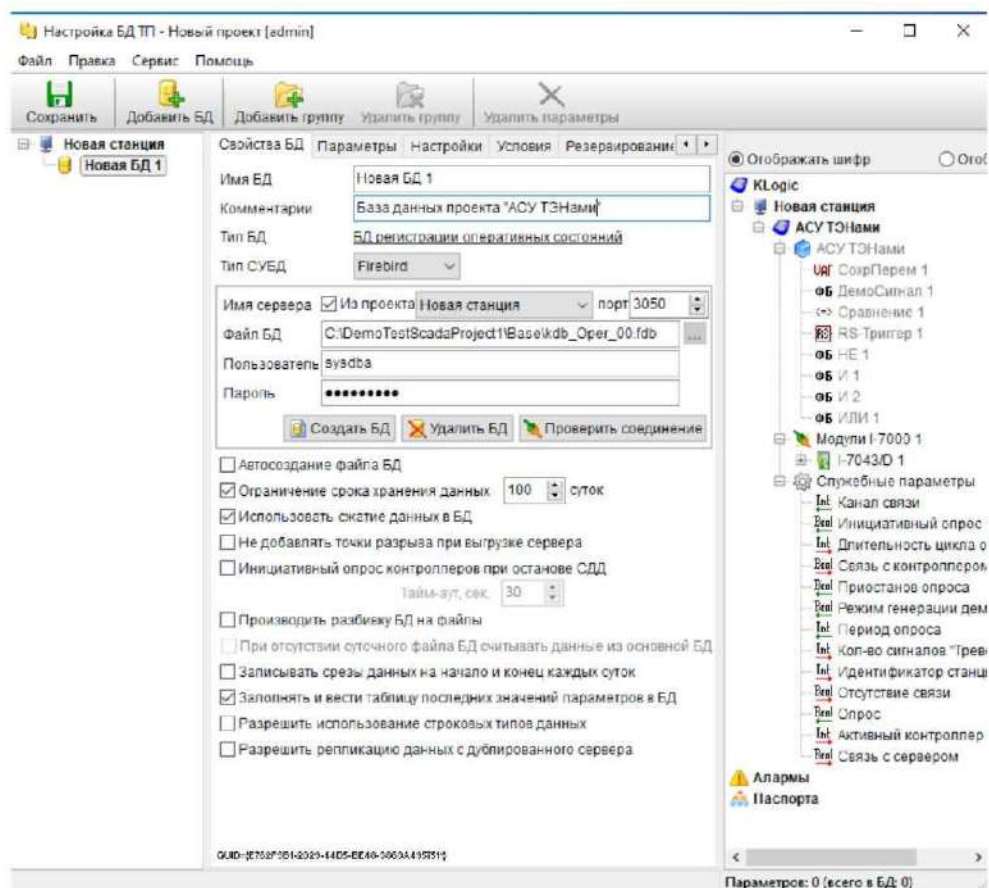


Рис. 9.3. Свойства БД

Непосредственно для создания базы данных необходимо нажать соответствующую кнопку или включить опцию «Автосоздание файла БД», при этом база будет создана автоматически.

На вкладке **«Параметры»** располагается список параметров БД, входящих в выделенную БД (или в выделенную группу параметров БД). Параметры можно добавить простым перетаскиванием в область параметров.

Добавляем необходимые параметры в БД (см. рис. 9.4).

Свойства БД	Параметры	Настройки	Условия	Резервирование	Архивные копии
Ид-р п...	И...	Тип	Зона	Шифр	Наименование
30	1	Дискретный ...	АСУ ТЭНами	Вых_ТЭН1	Дискретный выход
31	2	Дискретный ...	АСУ ТЭНами	DO1	Дискретный выход
32	3	Дискретный ...	АСУ ТЭНами	DO2	Дискретный выход
33	4	Дискретный ...	АСУ ТЭНами	DO3	Дискретный выход
34	5	Дискретный ...	АСУ ТЭНами	DO4	Дискретный выход
35	6	Дискретный ...	АСУ ТЭНами	DO5	Дискретный выход
36	7	Дискретный ...	АСУ ТЭНами	DO6	Дискретный выход
37	8	Дискретный ...	АСУ ТЭНами	DO7	Дискретный выход
38	9	Дискретный ...	АСУ ТЭНами	DO8	Дискретный выход
39	10	Дискретный ...	АСУ ТЭНами	DO9	Дискретный выход
40	11	Дискретный ...	АСУ ТЭНами	DO10	Дискретный выход
41	12	Дискретный ...	АСУ ТЭНами	DO11	Дискретный выход
42	13	Дискретный ...	АСУ ТЭНами	DO12	Дискретный выход
43	14	Дискретный ...	АСУ ТЭНами	DO13	Дискретный выход
44	15	Дискретный ...	АСУ ТЭНами	DO14	Дискретный выход
45	16	Дискретный ...	АСУ ТЭНами	DO15	Дискретный выход
46	1	Аналоговый ...	АСУ ТЭНами	Канал связи	Тип канала связи: ...
47	2	Дискретный ...	АСУ ТЭНами	Инициативны...	Команда на иници...
48	3	Аналоговый ...	АСУ ТЭНами	Длительность...	Длительность посл...
49	4	Дискретный ...	АСУ ТЭНами	Связь с контр...	Наличие связи с ко...
50	5	Дискретный ...	АСУ ТЭНами	Приостанов о...	Приостанов опроса
51	6	Дискретный ...	АСУ ТЭНами	Режим генер...	Режим генерации ...
52	7	Аналоговый ...	АСУ ТЭНами	Период опроса	Период полного пр...
53	8	Аналоговый ...	АСУ ТЭНами	Кол-во сигнал...	Количество дозвон...
54	9	Аналоговый ...	АСУ ТЭНами	Идентификат...	Идентификатор ст...
55	20	Дискретный ...	АСУ ТЭНами	Отсутствие св...	Отсутствие связи с ...
56	21	Дискретный ...	АСУ ТЭНами	Опрос	Опрос (для GSM - р...
57	22	Аналоговый ...	АСУ ТЭНами	Активный кон...	Активный контролл...
58	25	Дискретный ...	АСУ ТЭНами	Связь с сервер...	Наличие связи с се...

Рис. 9.4. Выбор параметров для записи в БД

Примечание. Размер файла базы данных зависит от количества записываемых параметров, поэтому для уменьшения объёма БДТП не следует без необходимости включать в неё все параметры проекта. Параметры, история изменения которых не представляет интереса, лучше исключить из БД.

При выделении какого-либо параметра БД, в дереве паспортов (расположенном в правой части основного окна программы) "подсвечивается" паспорт, соответствующий данному параметру. Если же паспорт, соответствующий параметру, не найден, параметр выделяется красным цветом. Для вывода окна с более подробной информацией о паспорте, необходимо дважды щёлкнуть мышью на параметре, либо выбрать пункт "Информация о параметре" из контекстного меню.

Вкладка «Настройки» (см. рис. 9.5) содержит следующие элементы редактирования (вид закладки зависит от типа БД, который выбирается при добавлении БД в проект):

Рис. 9.5. Настройки записи для БД

1. Панель настройки **периодической** записи для обычных БД.

Периодически — при выставлении этого признака запись значений параметров в БД будет осуществляться с заданным *Периодом* (в мсек).

Записывать только изменения — осуществлять запись значений параметров в БД только, если новое значение отличается от предыдущего на заданное количество процентов (*чувствительность*). Для записи по изменениям предусмотрено ещё две настройки:

Учитывать изменения временной метки — считать изменением параметра изменение его временной метки.

Максимальный интервал между записями (в сек) — производить обязательную запись, даже если значение параметра не изменилось (для редко меняющихся параметров).

2. Панель настройки **периодической** записи для БД с дочиткой.

Периодически — при выставлении этого признака запрос данных из устройств, которые имеют собственные хранилища, и запись значений параметров в БД будет осуществляться с заданной *Периодичностью дочитки* (в мин).

Задержка после запуска СДД (в мин) — начинать производить запрос данных из устройств и запись в БД не ранее, чем через заданное количество минут после запуска сервера доступа к данным.

3. Панель настройки записи **по расписанию** для обычных БД и БД с дочиткой.

По расписанию — при выставлении данного признака запись значений параметров в БД будет производиться в соответствии с заданным расписанием.

Расписание можно составить *На сутки*, *На неделю* или *На месяц*. Элементы расписания добавляются, удаляются и редактируются с помощью кнопок *Добавить*, *Удалить* и *Изменить*.

4. Панель настройки записи **по инициативе** для обычных БД и БД с дочиткой.

По инициативе — при выставлении данного признака запись значений параметров в БД будет производиться только при изменении заданного дискретного *Паспорта* (*По переднему фронту*, *По заднему фронту*, *По переходу в ошибку*, *По переходу из ошибки*). Инициативный паспорт выбирается с помощью кнопки *Выбрать*. Окно информации об инициативном паспорте вызывается с помощью кнопки *Информация*.

5. Панель настройки записи **по изменению** для обычных БД.

По изменению — производить запись в БД только тогда, когда происходит реальное изменение параметров. Уведомление об изменении (обновлении) значений параметров приходит от Сервера Доступа к Данным (в этом состоит принципиальное отличие данного вида записи от всех остальных видов).

Если *чувствительность* задать равной нулю, то все значения параметров, приходящие от СДД, будут записываться в БД, иначе — только если новое значение параметра отличается от предыдущего на заданное количество процентов.

Не записывать значения с одинаковой временной меткой — настройка, которая служит для исключения записи значений с повторяющейся временной меткой для одного и того же параметра.

В нашем случае выберем *периодическую* запись с параметром Записывать только изменения.

Вкладка «Условия» (см. рис. 9.6) отображается только для обычных БД и БД с дочиткой. Есть возможность выбрать *управляющий параметр*, по которому, в зависимости от его значения будет производиться запись в БД. Также можно временно отключить запись в БД, поставив галочку в пункте

Рис. 9.6. Условия записи в БД Вкладка «Резервирование» (см. рис. 9.7) отображается только для обычных БД и БД с дочиткой

Рис. 9.7. Настройки резервирования БД

Резервирование — признак ведения резервной БД.

Полный путь к файлу резервной БД задаётся полями редактирования *Имя сервера* и *Файл БД*. Сервер для файла резервной БД может быть выбран *Из проекта* с помощью выпадающего списка станций, зарегистрированных в проекте посредством программы "Настройка сетевого взаимодействия".

Имя пользователя и *Пароль* — имя пользователя SQL-сервера Interbase и пароль этого пользователя.

С помощью соответствующих кнопок можно *Создать файл* резервной БД, *Удалить файл* резервной БД и *Проверить соединение* с файлом резервной БД.

Временно не писать параметры в БД.

Резервная БД — полная копия данной БД, которая будет вестись одновременно с ней. Это нужно для дублирования БД на случай повреждения жесткого диска, на котором ведется данная БД, и имеет смысл только при ведении БД на другом жестком диске или на другой рабочей станции.

Резервирование базы данных нам не требуется.

Вкладка «**Архивные копии**» представлена на рис. 9.8. Для всех типов БД, кроме вторичных, существует возможность автоматического создания архивных копий, целью которого является длительное хранение информации о ходе технологического процесса. Эта возможность реализована в специальном модуле создания архивных копий баз данных технологических параметров, который позволяет автоматически создавать копии БД ТП, содержащие данные только за определенные пользователем интервалы времени (день, неделя, месяц, квартал, полугодие, год). Архивные копии баз данных могут храниться на файл-сервере, на CD и DVD-дисках и т.п., что позволяет хранить информацию о ходе технологического процесса длительное время.

Рис. 9.8. Настройки архивирования БД

Все технологические параметры, сохраняемые в БД, можно разбивать по группам. В каждой БД может быть произвольное количество групп параметров. Вложенность групп - неограниченная.

Делаем вкладку Архивные копии активной, для хранения зададим папку [C:\проекты\Archives](#).

10. Визуализация

Переходим на вкладку **Сервер, клиенты** программы Конфигуратор.

Данная закладка, изображённая на рис. 10.1, предназначена для запуска Сервера доступа к данным, а также клиентских модулей SCADA-системы "КАСКАД": Визуализация, Просмотр истории, Формирование рапортов, Просмотр архива событий.

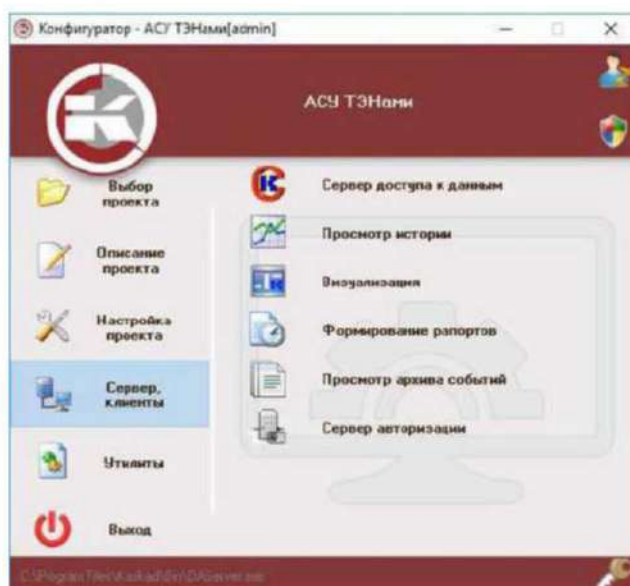


Рис. 10.1. Вкладка «Сервер, клиенты»

Следующий этап — настройка визуализации.

Диалоговая среда контроля и управления (модуль визуализации) предоставляет средства для разработки и отображения графических мнемосхем автоматизированных рабочих мест (АРМ) операторов систем промышленной автоматизации. Данный модуль является основным средством визуального контроля текущих параметров ТП, а также главным инструментом управления процессами. То есть задачей этого модуля является получение данных с одного или нескольких СДД и отображение в виде набора мнемонических схем (мнемосхем), а также предоставление пользователю возможности управления. **Мнемосхема** — совокупность сигнальных устройств и изображений оборудования и внутренних связей контролируемого объекта, размещаемых на диспетчерских пультах, операторских панелях или выполненных на персональном компьютере. Каждая мнемосхема позволяет в схематичном и понятном пользователю виде представить участок технологической схемы с отображением необходимых параметров техпроцесса.

Создаем мнемосхему. Переходим в режим редактирования (см. рис. 10.2). Как уже говорилось выше, переход между режимами просмотра и редактирования также может осуществляться с помощью «горячей» клавиши F6.

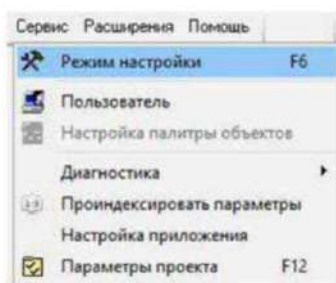


Рис. 10.2. Переход в режим настройки

Справа отображается дерево проекта, с помощью которого можно создавать, копировать, удалять мнемосхемы, группировать их по ветвям и пр. Включить отображения дерева проекта можно с помощью меню, кнопки *t*'' на панели управления, или «горячей» клавиши F3. Свойства объекта открываются по двойному щелчку на нём или по сочетанию клавиш Ctrl+E.

Свойства мнемосхемы (выходят по двойному щелчку по свободному месту мнемосхемы либо через контекстное меню мнемосхемы в дереве проекта) представлены на рис. 10.3.

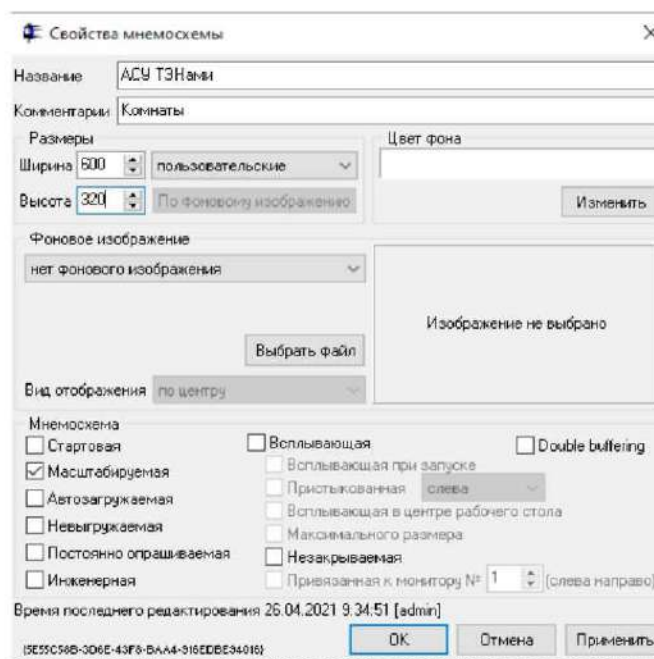


Рис. 10.3. Свойства мнемосхемы

В поле редактирования в верхней части окна можно изменять название мнемосхемы и комментарии к ней. Ниже задаются ширина и высота мнемосхемы в пикселях, и цвет фона мнемосхемы. Если для мнемосхемы выбрано фоновое изображение, и оно не соответствует заданным размерам мнемосхемы, то становится доступен флажок **Растянуть изображение**.

Включение флажка **Масштабируемая мнемосхема** разрешает масштабирование данной мнемосхемы. При включении флажка **Инженерная мнемосхема** доступ к данной мнемосхеме будет разрешен только пользователям, имеющим соответствующие права. При включении флажка **Постоянно опрашиваемая мнемосхема** данная мнемосхема опрашивается постоянно, даже если загружена другая мнемосхема. Включение флажка **Невыгружаемая мнемосхема** нужно для того, чтобы переход по мнемосхемам был более быстрым (мнемосхема не выгружается из памяти и ее загрузка происходит почти мгновенно). При включении флажка **Всплывающая мнемосхема** данная мнемосхема открывается в отдельном окне поверх других мнемосхем. **Стартовая мнемосхема** - это мнемосхема, с которой начинается загрузка программы, **автозагружаемая мнемосхема** - это мнемосхема, которая загружается автоматически при запуске программы.

Для добавления объектов на мнемосхемы можно использовать **палитру объектов** (вкладка **Палитра** окна **Дерево проекта**), основное меню или контекстное меню мнемосхемы (см. рис. 10.4).

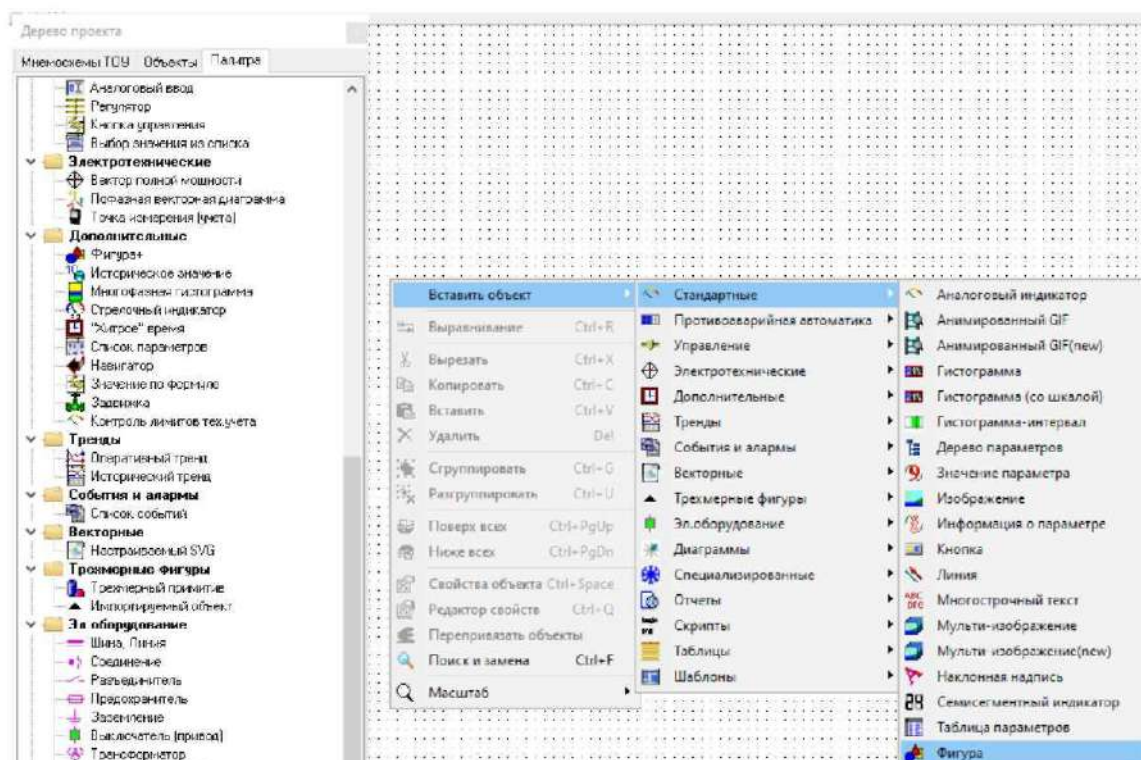


Рис. 10.4. Добавление объекта через палитру и через меню

Объекты разбиты на логические группы. Работать с объектами можно во вкладке **Объекты** окна **Дерево проекта**.

Для рисования прямоугольников-комнат выберем объект **Фигура** из группы **Стандартные**, который предназначен для отображения стандартных графических примитивов. Для настройки объекта «Фигура» предназначено окно его свойств, представленное на рис. 10.5.

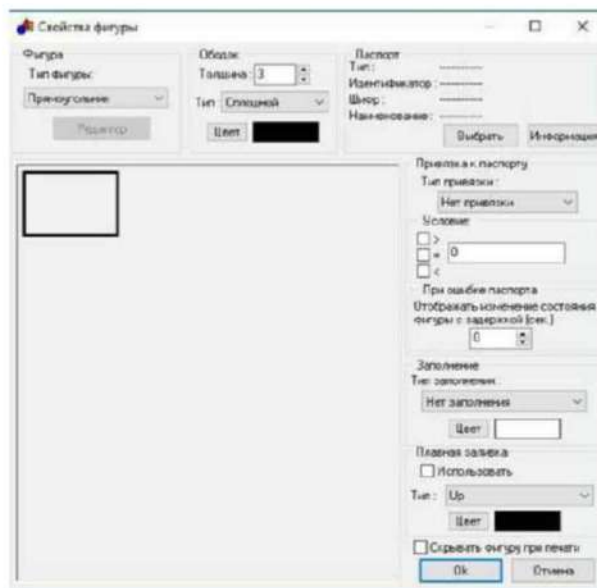


Рис. 10.5. Свойства фигуры

В свойствах указываем следующие параметры:

- Тип фигуры: Прямоугольник;

- Толщина ободка: 3;
- Заполнение: Нет.

Остальные две комнаты получим путём копирования. Скопировать любой объект на мнемосхеме можно несколькими способами: через пункты контекстного меню, сочетание клавиш либо меню Правка. В результате получим схематическое изображение, представленное на рис. 10.6.

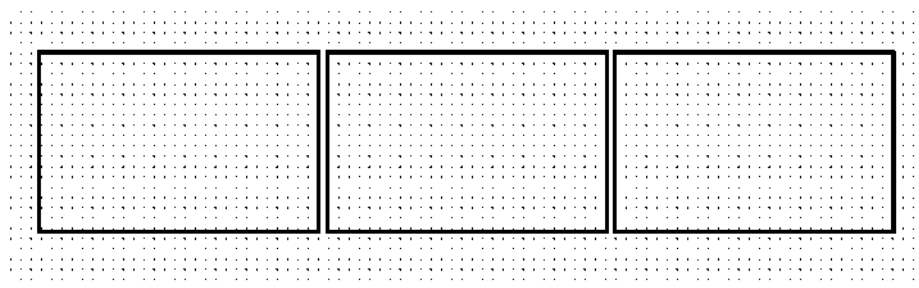


Рис. 10.6. Схематическое изображение комнат

В нашем случае на мнемосхеме необходимо

После этого добавляем элементы управления. отображать:

- 1) значение температуры;
- 2) кнопки перехода в ручной/автоматический режим;
- 3) кнопки включения/выключения при управлении в ручном режиме.

Кроме того, нужно добавить текущее время и поясняющие надписи.

Для отображения показания датчика температуры (канал ДемоСигнал.Выход 1) использовать объект **Значение параметра** из группы Стандартные (см. рис. 10.7).

можно

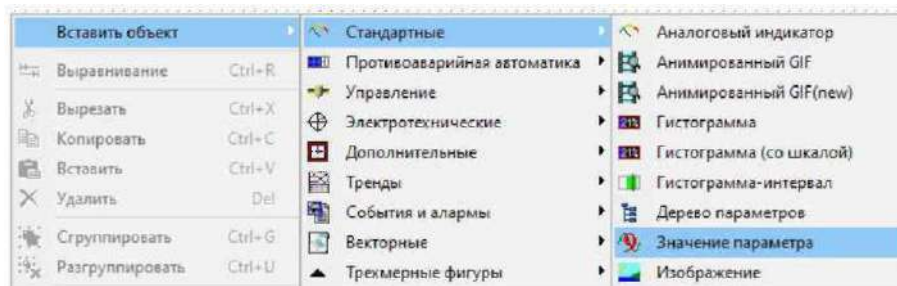


Рис. 10.7. Добавление объекта «Значение параметра»

Выбираем паспорт, к которому хотим привязать объект (см. рис. 10.8).

(Если паспорта не отображаются - см. стр.17).

Свойства объекта «Значение параметра» представлены на рис. 10.9.

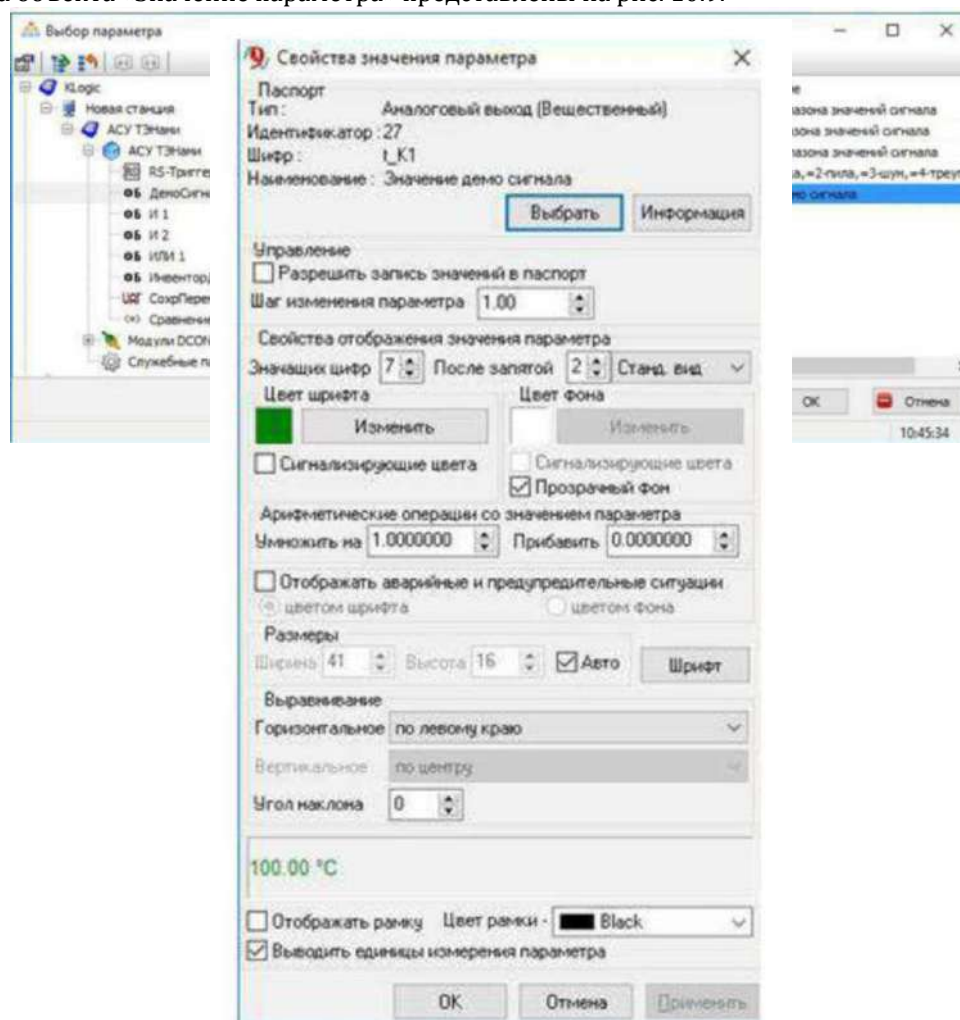


Рис. 10.9. Свойства значения параметра

В этом окне можно задать параметры отображения и форматирования текста, например, указать вывод единиц измерения параметра (флажок внизу окна).

Для перехода в ручной/автоматический режим, а также включения/выключения нагревателя при управлении в ручном режиме можно использовать объект визуализации **"Кнопка дискретного управления"** из группы Управление, который предназначен для записи значения в дискретный паспорт. Для этого на мнемосхеме отображается кнопка (с возможной надписью и картинкой), после нажатия которой и производится запись.

Выбираем параметр, в который будет записано значение при нажатии на кнопку. Для включения/отключения автоматического режима используется тег «СохрПерем.Автомат». При записи единицы система работает в автоматическом режиме, при нуле — ручном.

Какое именно значение быть записано в паспорт, определяется значением, выбранным из выпадающего списка "Действие по нажатию на кнопку". Возможны следующие варианты: «Изменение состояния» (можно записать и "0", и "1"); запись "1"; запись "0"; и пр.

Внешний вид кнопки настраивается. Кнопка может выглядеть как стандартная кнопка в Windows, либо быть плоской и прозрачной (невидимой) — для этого нужно выставить флаг "Плоская прозрачная кнопка". Кроме того, на кнопке может находиться произвольный "Текст" и "Картинка" (изображение из файла .bmp), причем для каждого состояния паспорта ("0" и "1") они могут быть разные. Также можно настроить взаимное расположение текста и картинки (см. рис. 10.10).

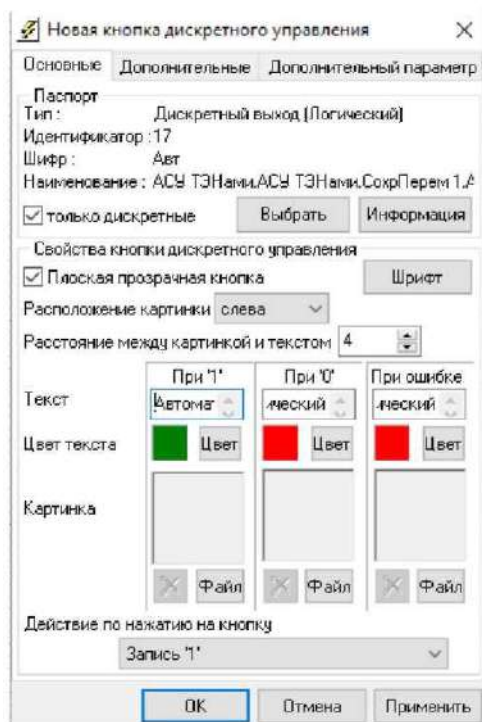


Рис. 10.10. Свойства кнопки дискретного управления

Для наглядности лучше поставить не одну кнопку для переключения режимов ручной/автоматический, а две кнопки для перехода в тот или иной режим.

Для кнопки *перехода в автоматический режим* в качестве «Действия при нажатии на кнопку» выберем «Запись 'Г'», текст на кнопке - «Автоматический» (в оба поля, и при 1, и при 0).

Кнопку *перехода в ручной режим* сделаем из первой. Для второй кнопки задаём свойства: Текст на кнопке «Ручной», Действие - «Запись '0'». После нажатия первой кнопки (перехода в автоматический режим) она блокируется, при этом становится доступна кнопка перехода в ручной режим.

Для *включения* (записи «1») и *выключения* (записи «0») *нагревателя в ручном режиме* работы системы нужно аналогичным способом добавить две кнопки дискретного управления для записи значений 1 и 0 соответственно в дискретный параметр «СохрПерем.Включить».

Для представления *состояния работы ТЭНов* можно использовать объект визуализации **"Мульти-изображение"** из группы Стандартные, который предназначен для вывода на мнемосхему изображения, зависящего от значения или состояния параметра. При этом для каждого состояния дискретного параметра будет выводиться своё изображение формата GIF.

Однако в данном случае удобно воспользоваться уже знакомым нам объектом **"Фигура"**, который также предназначен для отображения стандартных графических примитивов в динамическом режиме, т.е. данный объект позволяет отображать одну или несколько фигур в зависимости от значения дискретного параметра.

Свойства фигуры представлены на рис. 10.11. Привяжем фигуру к первому дискретному выходу DO0 модуля I-7043/D, который и осуществляет включение/отключение электронагревателя.

Для того, чтобы при изменении значения на данном выходе фигура меняла свой цвет, установим тип привязки «Две фигуры». При этом появляются две вкладки, на одной можно настроить вид фигура при 'Г,

на другой — при '0'. Для обоих случаев выбираем тип фигуры — окружность, цвет при 1 красный, при 0 серый (согласно заданию).

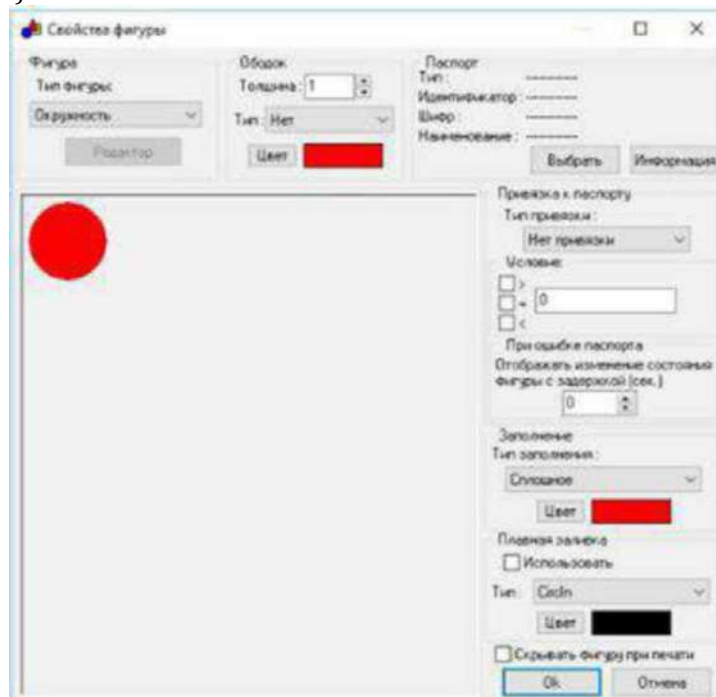


Рис. 10.11. Свойства фигуры

Для отображения *времени*, а также различных статических *текстов* (подписей, комментариев и пр.) воспользуемся объектом **Наклонная надпись** из группы Стандартные, причём для отображения времени нужно задать соответствующее значение в поле «Тип надписи» (см. рис. 10.12).

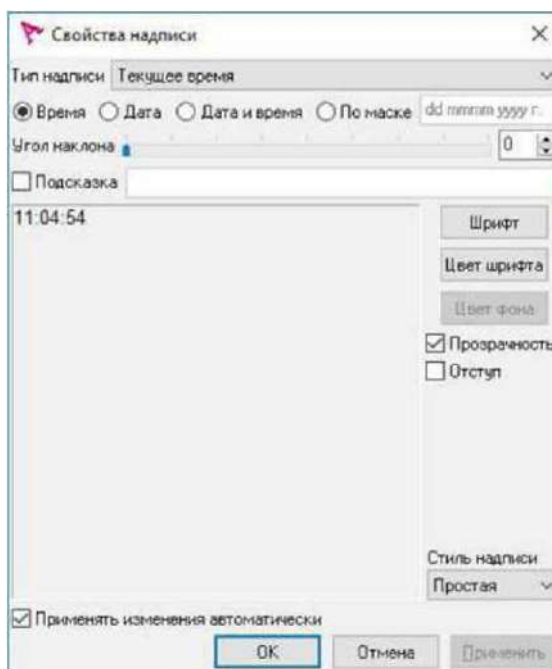


Рис. 10.12. Свойства надписи

Для *визуального контроля* процесса в Диалоговой Среде Контроля и Управления KVision существует два объекта визуализации - оперативный и исторический тренды, находящиеся в группе Тренды.

Оперативный тренд получает данные непосредственно от Сервера Доступа к Данным (СДД), поэтому позволяет увидеть историю за небольшой период времени, равному глубине тренда.

Основным же инструментом просмотра истории является **Исторический тренд**, который оперирует историческими данными, получая их не от СДД, а из баз данных регистратора. Он позволяет просматривать данные как в текущий момент времени, так и за произвольный промежуток времени из накопленных регистратором баз.

Окно свойств исторического тренда представлено на рис. 10.13.

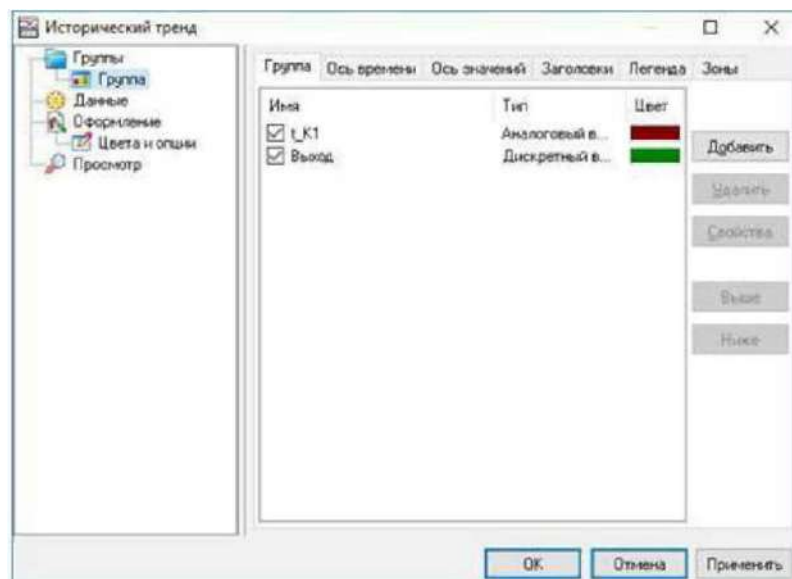


Рис. 10.13. Свойства исторического тренда

На вкладке Группы можно настроить список отображаемых трендов. Добавим два параметра. Остальные вкладки позволяют гибко настроить отображение трендов.

На вкладке Оформление можно настроить внешний вид области для отображения тренда, в частности, для экономии места на мнемосхеме можно сделать отображение тренда в виде кнопки, при нажатии которой и откроется окно с трендом.

В результате получаем следующую мнемосхему, которая представлена на рис. 10.14.



Рис. 10.14. Созданная мнемосхема

Сервер доступа к данным

Проверим работу системы посредством запуска СДД - Сервера доступа к данным.

Сервер доступа к данным - серверное приложение, предоставляющее централизованный механизм создания единого массива технологических параметров, их вторичной обработки по алгоритмам

пользователя, ведения истории технологического процесса и обработки клиентских запросов. Сервер доступа к данным загружает соответствующие модули доступа к данным, модуль обработки паспортов и модуль регистрации технологических параметров, запускает механизмы обработки и архивации, а также предоставляет клиентским приложениям из состава SCADA-системы «КАСКАД» единый интерфейс для доступа к параметрам, формируемым в результате контроля над технологическим процессом.

После запуска СДД в системном трее появляется вращающийся значок ©. При двойном щелчке по значку появляется окно СДД с двумя вкладками: Список событий и Модули.

На вкладке Список событий (рис. 10.15) можно просмотреть список произошедших событий СДД (загрузка модулей, необходимых для работы SCADA-системы «КАСКАД», добавление устройств и тегов и

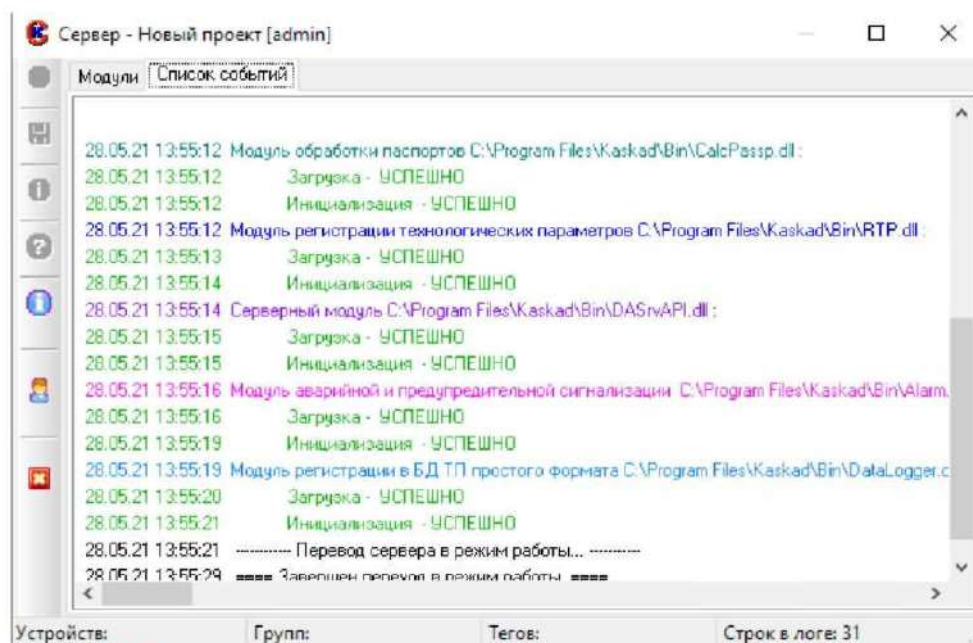


Рис. 10.15. Список событий СДД

Вкладка Модули содержит два списка и ряд кнопок (см. рис. 10.16): т.д.):

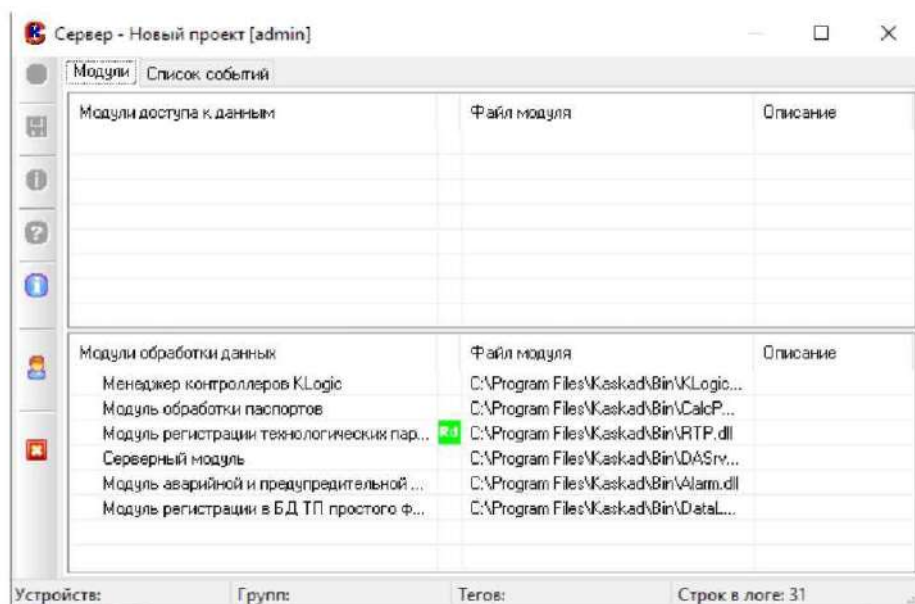


Рис. 10.16. Модули СДД

Первый список содержит конфигурации (модули доступа к данным) проекта, сведения о них (название конфигурации, файл модуля доступа к данным и его описание), а также процесс опроса, если он происходит.

На той же вкладке расположены **Модули обработки данных** - список загруженных модулей обработки данных, сведения о них (название, файл и описание модуля), а также процесс работы модуля, если модуль запущен.

С помощью кнопки **О** можно просмотреть краткую информацию о выбранном модуле обоих списков.

Проверка работы

Проверка работы - запуск СДД, просмотр элементов управления, взаимодействие с ними. Скриншот мнемосхемы в автоматическом режиме работы системы показан на рис. 10.17.



Рис. 10.17. Мнемосхема в автоматическом режиме работы

При нажатии кнопки Тренд открывается окно, представленное на рис. 10.18.

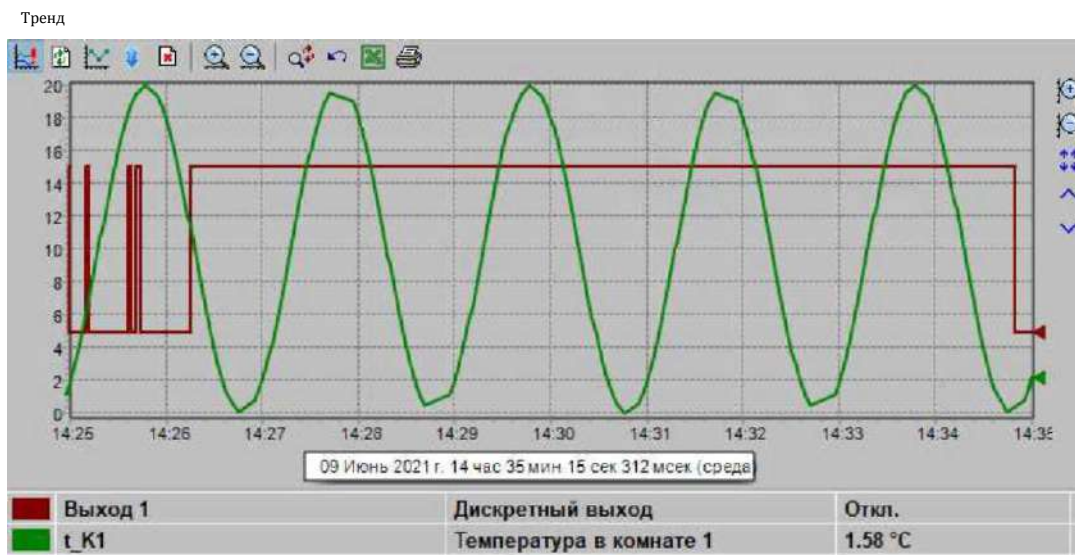


Рис. 10.18. Тренд в автоматическом режиме работы

Из тренда видно, что при понижении температуры ниже пяти градусов подаётся сигнал на включение нагревателя, при повышении температуры до плюс 15 нагреватель выключается.

Для проверки работы системы в ручном режиме в группе «Выбор режима» на мнемосхеме нажмём кнопку «Ручной». Далее, нажимая на кнопки «Вкл» и «Выкл», будем отслеживать состояние нагревателей с помощью их значков, а также графика на тренде (см. рис. 10.19).

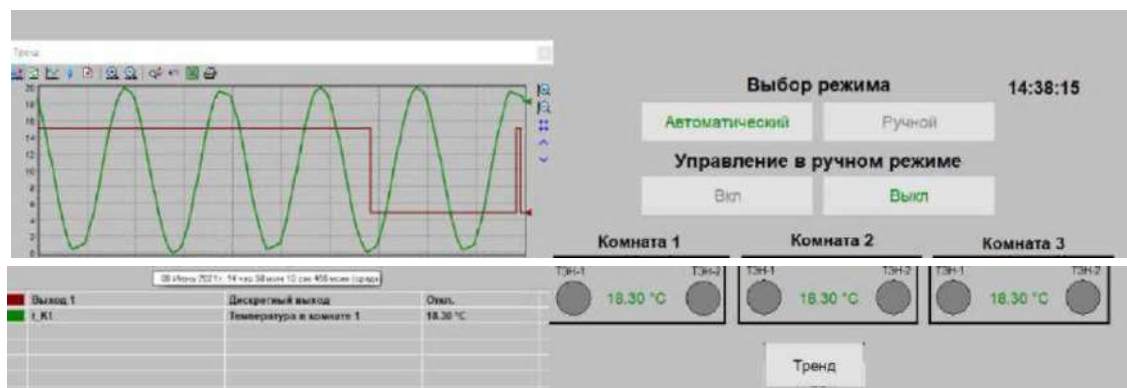


Рис. 10.19. Мнемосхема и тренд в ручном режиме работы

11. Сигнализация

Модуль звуковой сигнализации осуществляет контроль соответствия технологического процесса установленным режимам. В случае нарушений происходит информирование пользователя проигрыванием звуковых файлов. Благодаря чрезвычайно гибкой настройке модуль может быть использован также и для комментирования хода технологического процесса. В качестве звуковой информации могут быть использованы предварительно записанные речевые предупреждения. Сигнализирующее сообщение можно составлять из нескольких элементов, заикливать произвольный участок цепочки. Узел, вызвавший нарушение, отображается модулем визуализации, что позволяет немедленно принять необходимые меры. Каждому контролируемому параметру задается приоритет, что позволяет в первую очередь обрабатывать наиболее важные ситуации.

Открыть модуль можно с помощью Конфигуратора: кнопка «Настройка алармов» на вкладке Настройка проекта. При этом открывается окно настройки Конфигуратора модуля аварийной и предупредительной сигнализации, представленное на рис. 11.1.

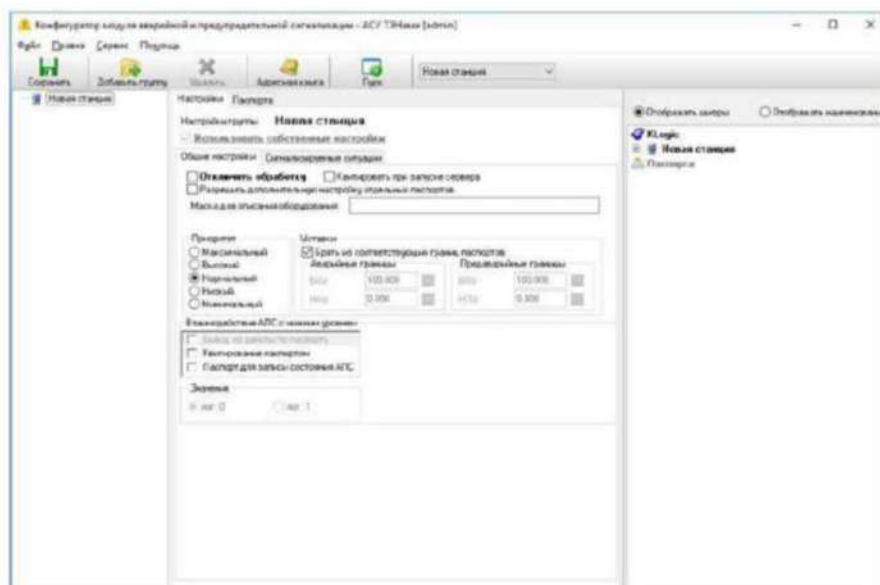


Рис. 11.1. Конфигуратор модуля аварийной и предупредительной сигнализации

Режим настройки позволяет установить общие настройки, настройки цвета, настройки сигнализируемых параметров. В режиме настройки составляется список параметров сигнализации, их настройки.

Основные настройки модуля определяются в форме Настройки. В заголовке показывается название и имя пользователя текущего проекта. В этом окне указываются основные настройки Алармов. Каждая группа содержит параметры, аварийное состояние которых будет сигнализироваться. Группа может иметь общие настройки для всех параметров. Параметры могут иметь отдельные настройки. Сигнализация может срабатывать по событию, которое можно выбрать.

Также существует три типа ситуации:

- *авария* - непосредственно выход каких-либо значений за допустимые границы;
- *предупреждение* - приближение каких-либо значений к допустимой границе;
- *сообщение* - сообщение о каком-либо событии.

Есть возможность вручную установить аварийные, предаварийные границы, либо брать их из паспорта. Можно выбрать приоритет ситуации - таким образом задается порядок сработки. Ситуацию можно называть *сквигированной*, если на нее была реакция со стороны оператора.

Также существует возможность настройки цветов сигнализации (пункт «Настройка цветов» в меню Сервис) — см. рис. 11.2.

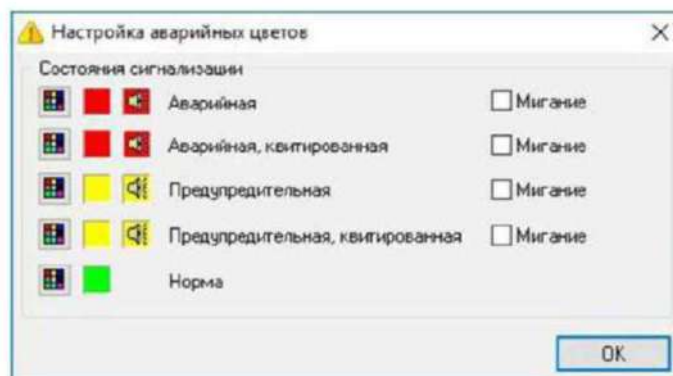


Рис. 11.2. Настройка аварийных цветов

Добавляем показание температуры: параметр ДемоСигнал 1.Температура (см. рис. 11.3).



Кроме того, для вывода информации о срабатывании аларма добавим аварийную ситуацию. Таким образом, при снижении температуры ниже 5 градусов будет срабатывать аларм.

12. Формирование рапортов



Назначение модуля формирования рапортов - получение необходимой информации об истории технологического процесса (результаты арифметической обработки значений параметров, информация о параметрах, временная статистика, мгновенные значения, выборка значений) в виде Excel-подобных таблиц, с возможностью экспорта сформированного рапорта в различные стандартные форматы представления данных, таких как Microsoft Excel, HTML и др.

Рапорт — это обработанная отчетная технологическая информация. Рапорта формируются на основе шаблонов (статической части рапортов) путем добавления в них свойств и значений заданных параметров технологического процесса, которые формируются (выбираются из базы данных технологических параметров и проходят соответствующую обработку) по алгоритмам, размещенным в этих шаблонах.

Окно программы имеет вид, представленный на рис. 12.1.

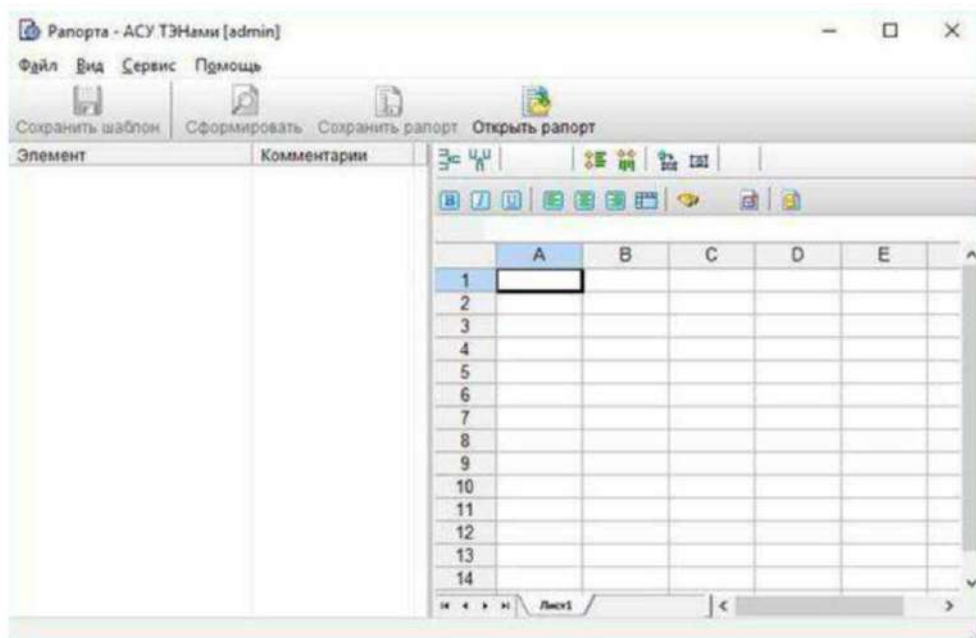


Рис. 12.1. Модуль формирования рапортов

Программа запускается в рабочем режиме с тем загруженным шаблоном, который был открыт в момент выхода из программы при последнем запуске программы. Если модуль формирования рапортов запускается в первый раз, то ни один шаблон не загружен.

Для перехода в режим настройки рапортов необходимо в меню **"Сервис"** выбрать пункт **"Режим настройки"** либо воспользоваться «горячей» клавишей F6.

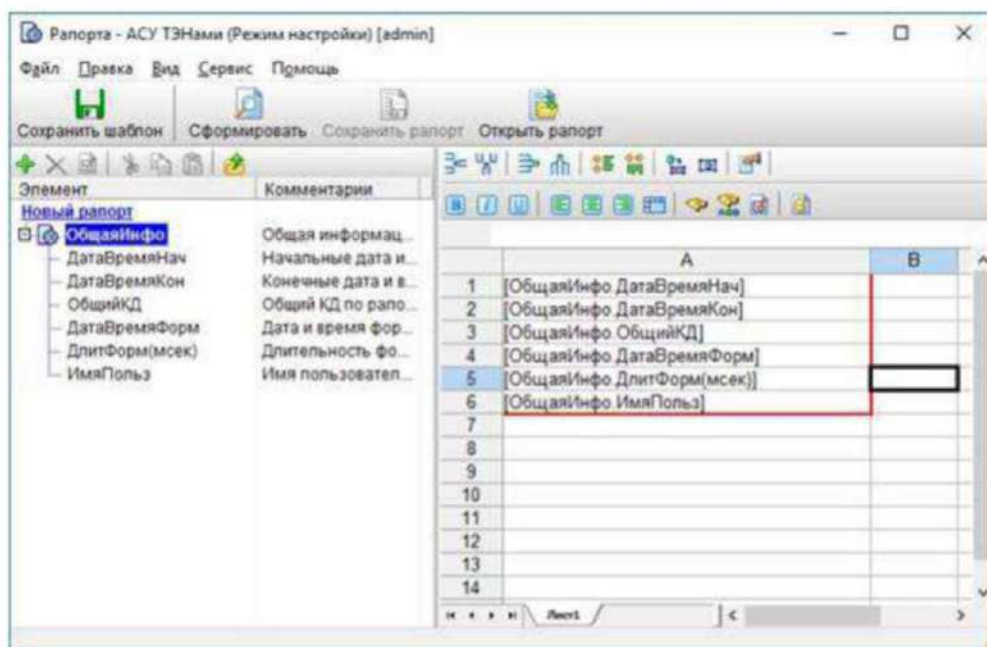


Рис. 11.2. Настройка рапортов

Основное окно программы формирования рапортов (см. рис. 12.2) состоит из следующих частей:

1. В верхней части находится панель управления, содержащая главное меню и панель инструментов (кнопки на панели инструментов дублируют основные пункты меню);
2. В левой части — дерево алгоритмов загруженного в данный момент шаблона рапорта (комментарии к элементам дерева можно скрыть/показать, выбрав в меню **"Вид"** пункт **"Комментарии в дереве рапорта"**);
3. В правой части — таблица отчёта загруженного в данный момент шаблона рапорта;
4. В нижней части — строка состояния, в которой отображаются подсказки.

Для навигации по шаблонам рапортов предназначено дерево рапортов (рис. 12.3), вызываемого по «горячей» клавише F3 или из соответствующего пункта меню Вид. При двойном щелчке левой кнопкой мыши по названию шаблона происходит открытие этого шаблона.

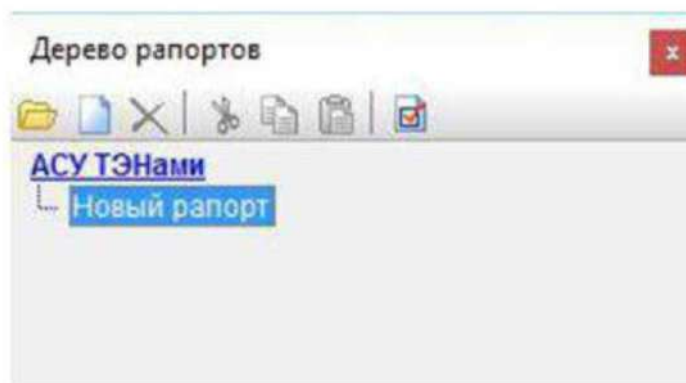


Рис. 12.3. Дерево рапортов

В режиме настройки в дереве рапортов появляется панель инструментов, предоставляющая пользователю дополнительные действия по работе с рапортами: редактирование, группировка, копирование, удаление и т.д.

Изменение взаимного расположения алгоритмов и ветвей в дереве алгоритмов осуществляется простым перетаскиванием этих элементов с помощью мыши.

При навигации по таблице отчёта происходит "подсвечивание" тех выходов в дереве алгоритмов, которым соответствуют выделенные ячейки. При этом происходит разворачивание элементов в дереве

алгоритмов. Для того, чтобы свернуть алгоритмы, можно воспользоваться кнопкой $\wedge$ на панели инструментов дерева алгоритмов основного окна программы.

В нашем случае необходимо создать два рапорта:

1. Включение и отключение электронагревателей за сутки.

После перехода в режим настройки рапорта подготовим его шаблон. В свойствах рапорта укажем название и комментарий к рапорту. Шапка таблицы будет иметь вид, представленный на рис. 12.4.

Режим работы ТЭНов

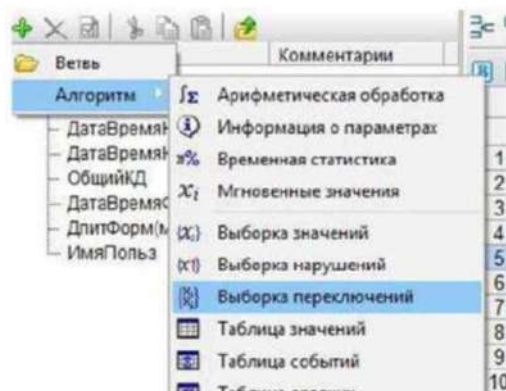
за период с начало периода
по конец периода

Рис. 12.4. Шапка таблицы

Укажем заголовок «Режим работы ТЭНов».

Для указания начального и конечного диапазона формирования рапорта перетащим из дерева алгоритмов параметры ДатаВремяНач и ДатаВремяКон (группа *ОбщаяИнфо*) в окно шаблона рапорта. С помощью контекстного меню и кнопок на панели инструментов можно задать форматирование текста и ячеек, а также объединять ячейки.

Включение и отключение ТЭНов нам нужно отслеживать. Для параметров за заданный интервал



скретного выхода, которое и нескольких дискретных включений" (см. рис. 11.5).

Рис. 12.5. Добавление алгоритма

Данный алгоритм работает с одним набором данных, представляющим из себя результирующую таблицу переключений заданных параметров, количество строк которой заранее неизвестно. При добавлении алгоритма в таблицу отчёта автоматически создаётся бэнд, привязанный к набору данных и размножающийся в готовом рапорте по числу записей в этом наборе, т.е. по числу строк (столбцов) в итоговой таблице переключений.

При добавлении алгоритма выходит окно, позволяющее настроить его параметры. Данное окно состоит из нескольких вкладок.

Вкладка «Диапазон» предназначена для настройки временного диапазона, за который будут запрашиваться данные. Оставим флажок «Наследование временного диапазона родительской ветки» включённым.

Вкладка «Параметры» предназначена для выбора дискретных параметров, по которым будут запрашиваться данные. Выберем первый дискретный выход DO0 модуля I-7043/D, который и осуществляет включение/отключение первого электронагревателя. Здесь же можно настроить текст, который будет отображаться в результирующем рапорте при соответствующих изменениях значения параметра. Если в группе присутствует несколько параметров, то для наглядности можно сделать выделение отдельных параметров цветом. Поле «Текст группы» позволяет задать дополнительный текст, который можно задать для параметров, чтобы обозначить их принадлежность к разным группам.

Вкладка «Выходы» позволяет настроить поля, которые будут отображаться в рапорте. Выберем следующие параметры: момент времени и текст значения. Отмеченные выходы будут отображаться в дереве алгоритмов основного окна программы.

Наконец, вкладка «Общие выходы» позволяет добавить в рапорт начальный и конечный диапазон времени.

В результате получилось дерево алгоритмов, представленное на рис. 11.6.

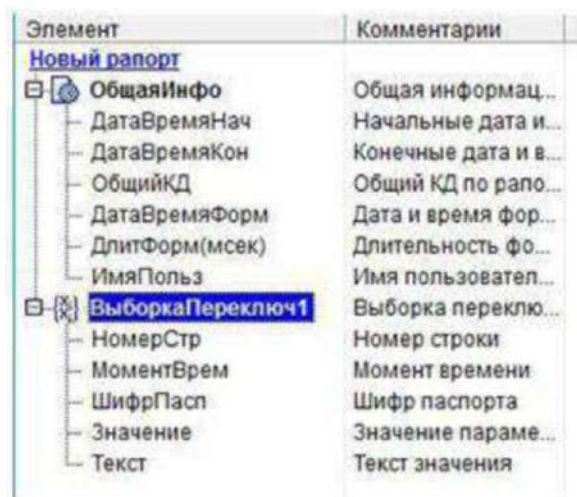


Рис. 12.6. Дерево алгоритмов

Для добавления бэнда перетащим его на лист. В запросе о заполнении ячеек необходимо указать горизонтальный бэнд, чтобы данные отображались построчно. Выше бэнда укажем название полей таблицы.

В конце шаблона необходимо указать дату и время формирования рапорта, для этого воспользуемся элементом ОбщаяИнфо/ДатаВремяФорм дерева алгоритмов.

Внимание! Для корректного отображения даты и времени необходимо настроить тип ячеек через вкладку «Число» окна «Формат ячеек», вызываемого через контекстное меню ячейки (а также кнопку на панели инструментов или сочетание клавиш Ctrl+1).

Получим следующий шаблон рапорта (рис. 12.7).

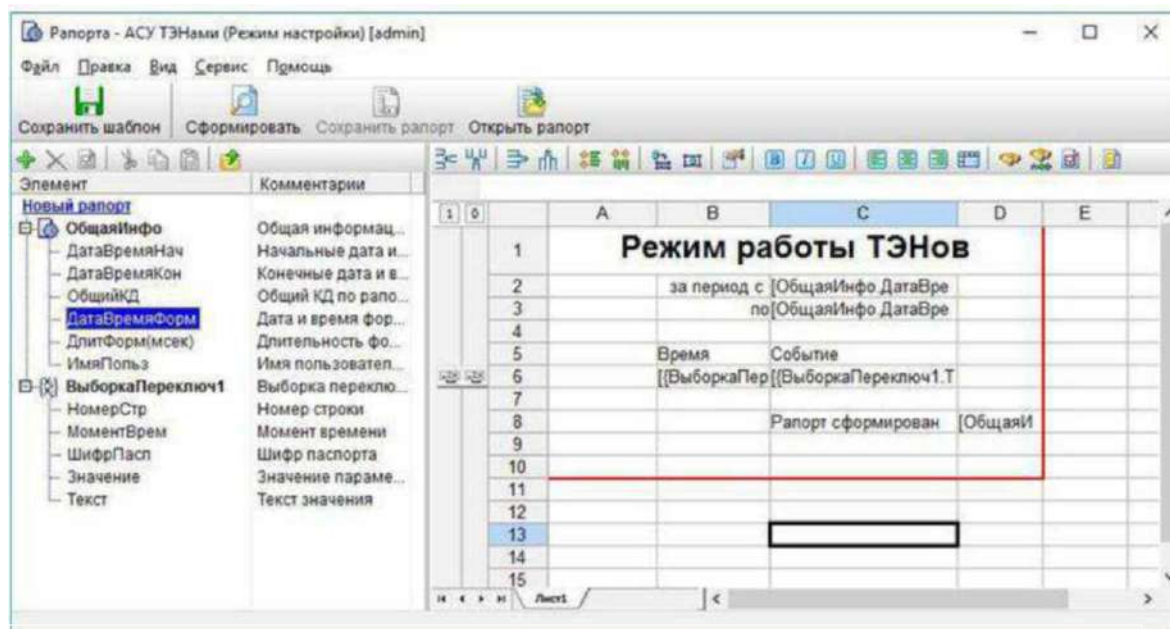


Рис. 12.7. Шаблон рапорта Для указания диапазона формирования рапорта откроем свойства рапорта (см. рис. 12.8).

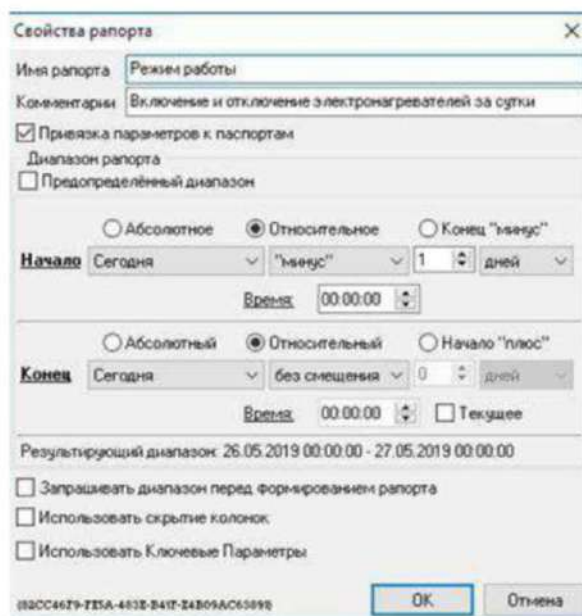


Рис. 12.8. Свойства рапорта

Так как необходимо вывести режим работы нагревателей за (предыдущие) сутки, укажем конец диапазона: «Относительный», сегодня, 0:00; начало диапазона: «Конец минус», 1 день.

Для формирования рапорта необходимо выбрать пункт "Сформировать рапорт" в меню Сервис или нажать «горячую» клавишу F9. Если в свойствах рапорта выставлен признак "Запрашивать временные настройки перед формированием рапорта", то на экране появится диалоговое окно, которое содержит панель настройки временного диапазона.

При нажатии на кнопку "OK" запустится процесс формирования рапорта. После завершения этого процесса на экране появится окно предварительного просмотра сформированного рапорта (рис. 12.9).

Режим работы ТЭНов			
в период с 08.08.2021 по 10.08.2021 0:00			
Событие	10:11	Событие: ТЭН1 Выключен	0
Событие	10:12	Событие: ТЭН1 Выключен	1
Событие	10:13	Событие: ТЭН1 Выключен	0
Событие	10:14	Событие: ТЭН1 Выключен	1
Событие	10:15	Событие: ТЭН1 Выключен	0
Событие	10:16	Событие: ТЭН1 Выключен	1
Событие	10:17	Событие: ТЭН1 Выключен	0
Событие	10:18	Событие: ТЭН1 Выключен	1
Событие	10:19	Событие: ТЭН1 Выключен	0
Событие	10:20	Событие: ТЭН1 Выключен	1
Событие	10:21	Событие: ТЭН1 Выключен	0
Событие	10:22	Событие: ТЭН1 Выключен	1
Событие	10:23	Событие: ТЭН1 Выключен	0
Событие	10:24	Событие: ТЭН1 Выключен	1
Событие	10:25	Событие: ТЭН1 Выключен	0
Событие	10:26	Событие: ТЭН1 Выключен	1
Событие	10:27	Событие: ТЭН1 Выключен	0
Событие	10:28	Событие: ТЭН1 Выключен	1
Событие	10:29	Событие: ТЭН1 Выключен	0
Событие	10:30	Событие: ТЭН1 Выключен	1
Событие	10:31	Событие: ТЭН1 Выключен	0
Событие	10:32	Событие: ТЭН1 Выключен	1
Событие	10:33	Событие: ТЭН1 Выключен	0
Событие	10:34	Событие: ТЭН1 Выключен	1
Событие	10:35	Событие: ТЭН1 Выключен	0
Событие	10:36	Событие: ТЭН1 Выключен	1
Событие	10:37	Событие: ТЭН1 Выключен	0
Событие	10:38	Событие: ТЭН1 Выключен	1
Событие	10:39	Событие: ТЭН1 Выключен	0
Событие	10:40	Событие: ТЭН1 Выключен	1
Событие	10:41	Событие: ТЭН1 Выключен	0
Событие	10:42	Событие: ТЭН1 Выключен	1
Событие	10:43	Событие: ТЭН1 Выключен	0
Событие	10:44	Событие: ТЭН1 Выключен	1
Событие	10:45	Событие: ТЭН1 Выключен	0
Событие	10:46	Событие: ТЭН1 Выключен	1
Событие	10:47	Событие: ТЭН1 Выключен	0
Событие	10:48	Событие: ТЭН1 Выключен	1

Рис. 12.9. Готовый рапорт

Примечание. Для того, чтобы не загромождать рапорт, диапазон настроен на последние 20 минут.

Назначение кнопок на панели инструментов этого окна понятно из всплывающих подсказок; более подробное описание можно посмотреть в справочной системе. Так, с помощью этих кнопок итоговый рапорт можно распечатать, экспортировать в формат Microsoft Excel, HTML или текстовый документ и пр.

2. Средняя температура в комнатах.

В этом случае необходимо получить рапорт, содержащий среднюю температуру в комнатах по суткам за определённый, настраиваемый, период.

Этот рапорт можно сделать на основе предыдущего, для этого нужно создать его копию с помощью дерева рапортов. Для копирования рапорта можно воспользоваться контекстным меню либо кнопками на панели инструментов окна дерева рапортов.

По умолчанию скопированный рапорт имеет имя Копия Оригинал. Для изменения имени воспользуемся окном свойств рапорта, кроме того, укажем комментарий к рапорту, а также диапазон времени рапорта.

Так, для получения рапорта с начала прошлой недели до текущего момента необходимо указать конец диапазона: «Сегодня», «без смещения», время «текущее»; начало диапазона: «Относительное», Начало недели, «минус», 7 дней, 0:00.

Для того чтобы у пользователя была возможность задания своего диапазона установим флажок «Запрашивать временные настройки перед формированием рапорта». При этом перед формированием рапорта будет выведено диалоговое окно с возможностью корректировки временного диапазона (рис. 12.10).

Рис. 12.10. Свойства рапорта

Для отображения шаблона рапорта необходимо дважды щёлкнуть его название в дереве рапортов.

После этого подготовим шапку таблицы: укажем заголовок, остальные элементы шапки оставим без изменения.

Бэнд, содержащий ячейки, содержащие элементы алгоритма «Выборка переключений» можно удалить с помощью контекстного меню строки либо соответствующей кнопки на панели инструментов (сам алгоритм из дерева алгоритмов тоже можно удалить аналогичным способом — через контекстное меню или кнопки на панели инструментов).

Для получения таблицы значений арифметической обработки (средних, интегралов и т.п.) одного или нескольких параметров за заданный интервал времени с заданным периодом следования используется алгоритм «Таблица средних» (см. рис. 12.11).

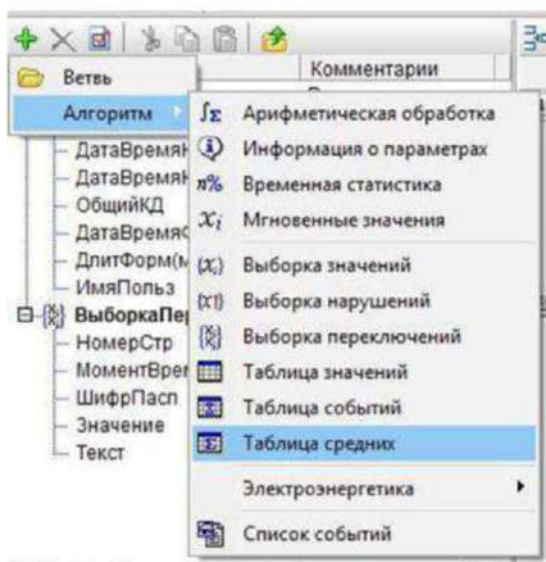


Рис. 12.11. Добавление алгоритма «Таблица средних»

Данный алгоритм работает с одним набором данных, представляющим из себя результирующую таблицу значений. При добавлении алгоритма в таблицу отчёта автоматически создаётся бэнд, привязанный к набору данных и размножающийся в готовом рапорте по числу записей в этом наборе, т.е. по числу строк (столбцов) в итоговой таблице.

При добавлении алгоритма выходит окно, позволяющее настроить его параметры. Данное окно состоит из нескольких вкладок.

Вкладка «Диапазон» предназначена для настройки временного диапазона таблицы средних. Флажок «Наследование временного диапазона родительской ветки» оставим включённым. Для указания периода, с которым будут следовать средние значения в итоговой таблице, зададим соответствующее значение (в секундах, минутах, часах или днях — к примеру, 1 день) в поле «Период следования данных в таблице».

Вкладка «Параметры» предназначена для выбора параметров, по которым будут запрашиваться данные. Выберем выход ФБ ДемоСигнал, который задаёт текущую температуру в комнате.

Вкладка «Выходы» позволяет настроить поля, которые будут отображаться в рапорте. Выберем одно поле Среднее (для отображения среднее значение параметра), которое и будет отображаться в дереве алгоритмов основного окна программы.

Наконец, вкладка «Общие выходы» позволяет добавить в рапорт номер строки, а также начальный и конечный моменты периода.

В результате получим следующее дерево алгоритмов, представленное на рис. 12.12.

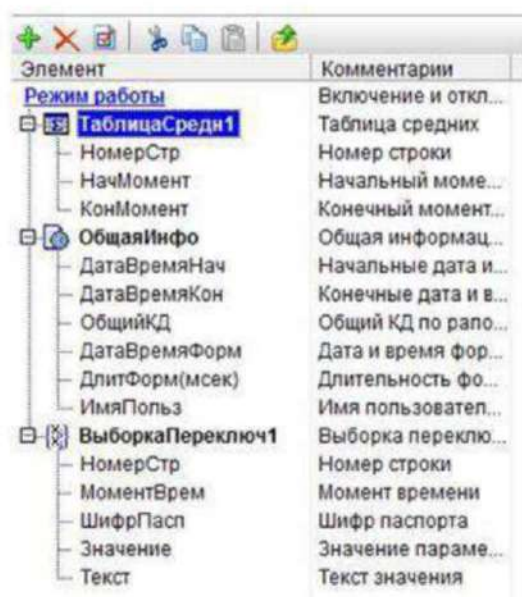


Рис. 12.12. Дерево алгоритмов

Добавим бэнд, перетащив на лист группу «Значение демо сигнала». Для построчного отображения данных создадим горизонтальный бэнд. Для указания номера строки необходимо перенести соответствующий элемент из дерева алгоритмов в строчку бэнда, на свободную ячейку (например, слева). Кроме того, укажем диапазон времени, в котором находится среднее значение, для этого добавим элементы НачМомент и КонМомент с интервалом в одну ячейку, в которую напомним дефис.

Выше бэнда укажем название полей таблицы.

В конце шаблона необходимо указать дату и время формирования рапорта, для этого воспользуемся элементами ОбщаяИнфо/ДатаВремяФорм дерева алгоритмов.

Внимание! Для корректного отображения даты и времени необходимо настроить тип ячеек через вкладку «Число» окна «Формат ячеек», вызываемого через контекстное меню ячейки (а также кнопку на панели инструментов или сочетание клавиш Ctrl+1).

И, наконец, после формирования получим готовый рапорт, представленный на рис. 12.14.

Рапорт - АСУ ТЭНов (Режим настройки) [admin]

Предварительный просмотр рапорта - Режим работы ТЭНов

594	09.06.2021 9:53	09.06.2021 9:54	
595	09.06.2021 9:54	09.06.2021 9:55	
596	09.06.2021 9:55	09.06.2021 9:56	
597	09.06.2021 9:56	09.06.2021 9:57	
598	09.06.2021 9:57	09.06.2021 9:58	
599	09.06.2021 9:58	09.06.2021 9:59	
600	09.06.2021 9:59	09.06.2021 10:00	
601	09.06.2021 10:00	09.06.2021 10:01	
602	09.06.2021 10:01	09.06.2021 10:02	
603	09.06.2021 10:02	09.06.2021 10:03	
604	09.06.2021 10:03	09.06.2021 10:04	
605	09.06.2021 10:04	09.06.2021 10:05	
606	09.06.2021 10:05	09.06.2021 10:06	
607	09.06.2021 10:06	09.06.2021 10:07	
608	09.06.2021 10:07	09.06.2021 10:08	
609	09.06.2021 10:08	09.06.2021 10:09	
610	09.06.2021 10:09	09.06.2021 10:10	
611	09.06.2021 10:10	09.06.2021 10:11	6,2976118
612	09.06.2021 10:11	09.06.2021 10:12	9,1736697
613	09.06.2021 10:12	09.06.2021 10:13	10,774646
614	09.06.2021 10:13	09.06.2021 10:14	9,2533279
615	09.06.2021 10:14	09.06.2021 10:15	10,837157
616	09.06.2021 10:15	09.06.2021 10:16	9,1442126
617	09.06.2021 10:16	09.06.2021 10:17	10,825614
618	09.06.2021 10:17	09.06.2021 10:18	9,2575423
619	09.06.2021 10:18	09.06.2021 10:19	10,744493
620	09.06.2021 10:19	09.06.2021 10:20	9,1822885
621	09.06.2021 10:20	09.06.2021 10:21	10,818671
622	09.06.2021 10:21	09.06.2021 10:22	9,2221318
623	09.06.2021 10:22	09.06.2021 10:23	10,837667
624	09.06.2021 10:23	09.06.2021 10:24	9,1447822
625	09.06.2021 10:24	09.06.2021 10:25	10,757117
626	09.06.2021 10:25	09.06.2021 10:26	9,1459386
627	09.06.2021 10:26	09.06.2021 10:27	10,770939
628	09.06.2021 10:27	09.06.2021 10:28	9,1639781
629	09.06.2021 10:28	09.06.2021 10:29	10,780333
630	09.06.2021 10:29	09.06.2021 10:30	9,1683748

Лист 1

Правое по

Страница 1 из 24

Рис. 12.14. Готовый рапорт

Получим следующий шаблон (см. рис. 12.13).

Примечание. Для того, чтобы не загромождать рапорт, диапазон настроен на последние 10 минут с интервалом в 25 секунд.

Для открытия рапортов из мнемосхемы в модуле Визуализация можно добавить кнопки формирования рапортов (группа Рапорта).

Окно настройки кнопки формирования рапорта имеет следующий вид (см. рис. 12.15).

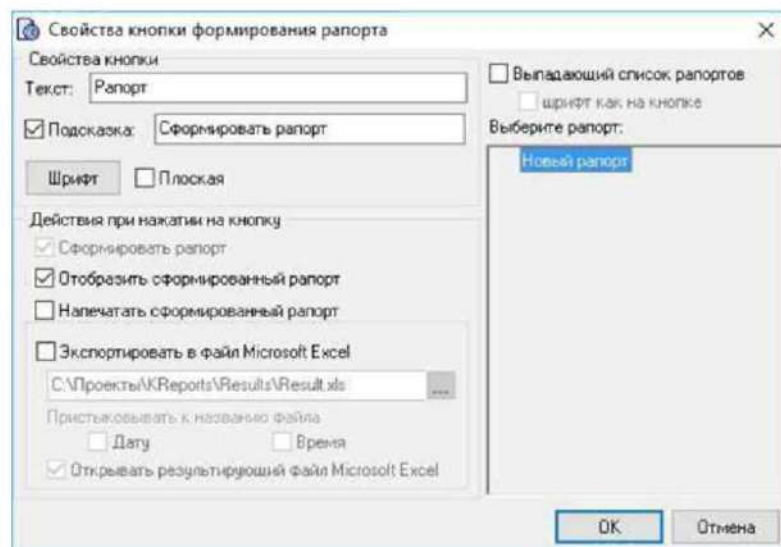


Рис. 12.15. Свойства кнопки формирования рапорта

Как видно, можно задать название для кнопки, подсказку, выбрать рапорт для формирования, настроить экспорт рапорта в файл MS Excel.

На этом разработка проекта завершена.

Заключение

В данной курсовой работе нами была поэтапно рассмотрена разработка АСУ электронагревателями в SCADA-системе «Каскад»:

1. Реализована технологическая программа в системе программирования контроллеров KLogic.
2. Создана технологическая мнемосхема в модуле Визуализация, в которой реализованы: управление и отображение режима работы АСУ (ручной/автоматический); управление работой электронагревателей каждой из комнат в ручном режиме; отображение значения температуры в комнатах с отображением единиц измерения; индикаторы для отображения работы каждого из электронагревателей.
3. Настроена БД технологических параметров.
4. Настроена сигнализация, срабатывающая при температуре в каждой из комнат ниже 5 °C (аварийный минимум).
5. Настроены отчеты: по средней температуре в комнатах за определённый интервал времени, а также по режиму работы электронагревателей за сутки.

В результате выполнения работы создана АСУ, которая может работать в двух режимах:

1. Ручной, при котором включение и отключение электронагревателей осуществляется оператором.
2. Автоматический. При этом включение и отключение электронагревателей осуществляется в зависимости от температуры в комнате. При температуре ниже плюс 5 °C электронагреватели включаются, при достижении температуры плюс 15 °C — выключаются.