

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Ключевая Юрай Сергеевич
Должность: Ректор
Дата подписания: 23.09.2026 09:01:08
Уникальный программный ключ:
3beb265d5d589e7ff4c954946f3ad99a1e70ac14

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра «Кибернетических систем»

ПРОЕКТИРОВАНИЕ МИКРОПРОЦЕССОРНЫХ СИСТЕМ АВТОМАТИЗАЦИИ

*Методические указания к выполнению курсовой работы
направление: 15.03.04 – «Автоматизация технологических процессов и
производств»
программа: «Автоматизация технологических процессов и производств в
нефтяной и газовой промышленности»
для всех форм обучения*

Составители

Н.В. Попова

старший преподаватель,

В.В. Козлов

кандидат технических наук, доцент

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 6FA44C50384686A8E7BD5E27735179BC
Владелец: Ефремова Вероника Васильевна
Действителен: с 14.06.2022 до 07.09.2023

Тюмень 2019

Проектирование микропроцессорных систем автоматизации: методические указания к выполнению курсовой работы по дисциплине «Проектирование микропроцессорных систем автоматизации», для обучающихся на направлении 15.03.04 – «Автоматизация технологических процессов и производств», программа – академический бакалавриат, квалификация – бакалавр для всех форм обучения[Текст] / сост. Н.В. Попова, В.В. Козлов Тюменский индустриальный университет.– 2-е изд., испр.– Тюмень: Издательский центр БИК ТИУ 201.–32 с.

Методические указания рассмотрены и рекомендованы к изданию на заседании кафедры Кибернетических систем

«_5_» __02__ 2019 года, протокол № 8.

Аннотация

Методические указания и индивидуальные задания по дисциплине «Проектирование микропроцессорных систем автоматизации» обучающихся по направлению: 15.03.04 – «Автоматизация технологических процессов и производств» направление: «Автоматизация технологических процессов и производств в нефтяной и газовой промышленности», программа – академический бакалавриат, квалификация – бакалавр, для всех форм обучения
Данная дисциплина изучается в одном семестре.

Приведены основные теоретические сведения из дисциплины.

Содержание

Пояснительная записка	4
1 Разработка автоматизированной системы управления	11
1.1 Алгоритм выполнения курсовой работы	13
1.2 Этапы проектирования автоматизированных систем управления технологическим процессом	14
1.3 Примеры тем для курсовой работы:	12
2 Содержание пояснительной записки	13
2.1 Текстовая часть	13
2.2 Обязательный перечень приложений	14
2.3 Требования к разрабатываемому программному обеспечению	14
3 Оформление пояснительной записки	15
4 Оформление списка литературы	17
4.1 Электронный ресурс	17
Рекомендуемая литература	18
Приложение А Оформление титульного листа	19
Приложение Б Функциональная схема	20
Приложение В Алгоритм	19
Приложение Г Программа	20
Приложение Д Основной экран	22
Приложение Е Краткая информация для написания программ на языках FBD и IL	25

Пояснительная записка

Общепризнанным является тот факт, что сегодня и в ближайшем будущем основу стабильности экономики страны будет составлять развитый топливно-энергетический комплекс.

Не вызывает сомнений и то, что управление технологическими процессами необходимо осуществлять с помощью средств автоматизации.

Умение ориентироваться на переполненном сегодня рынке устройств автоматизации и главное грамотно построить безопасную, надежную, удовлетворяющую современным техническим требованиям автоматизированную систему контроля и управления технологическим процессом, является обязательным атрибутом выпускника специальности «Специальные разделы микропроцессорных систем автоматизации и управления».

Для закрепления полученных навыков по построению систем автоматизации предлагается выполнить курсовую работу, целью которой является разработка проекта автоматизации технологической установки, используемой в топливно-энергетическом комплексе России.

Цель работы: Рассмотрение ключевых аспектов автоматизации технологических процессов, которые существенно пересекаются с проблемами развития отечественной нормативной базы.

Выполнив курсовую работу, студент может проанализировать, точно определить характеристики и отладить каналы передачи информации для систем контроля и управления.

На основании требований обучающимся составляется функциональная спецификация, которая определяет функции, выполняемые контроллером для пользователя после завершения проектирования, уточняя тем самым, насколько устройство соответствует предъявляемым требованиям.

Она включает в себя описания форматов данных, как на входе, так и на выходе, а также внешние условия, управляющие действиями контроллера.

Функциональная спецификация и требования пользователя являются критериями оценки функционирования контроллера после завершения проектирования. Может потребоваться проведение нескольких итераций, включающих обсуждение требований и функциональной спецификации с потенциальными пользователями контроллера

За последнее десятилетие появилось несколько технологических языков. Более того, Международной Электротехнической Комиссией разработан стандарт IEC-1131, концентрирующий все передовое в области языков программирования для систем автоматизации технологических процессов. Этот стандарт требует от различных изготовителей ПЛК предлагать команды, являющиеся одинаковыми и по внешнему виду, и по действию.

Стандарт специфицирует 5 языков программирования:

- Sequential Function Chart (SFC) – язык последовательных функциональных блоков;
- Function Block Diagram (FBD) – язык функциональных блоковых диаграмм;
- Ladder Diagrams (LAD) – язык релейных диаграмм;
- Statement List (STL) – язык структурированного текста;
- Instruction List (IL) – язык инструкций.

STL и IL являются неким программистским «эсперанто» – они взяли в себя наиболее общие операторы языков типа Pascal и ассемблер. Они обеспечивают совместимость стандарта с ранними версиями программного обеспечения производителей контроллеров.

Язык SFC позволяет осуществлять программирование на алгоритмическом уровне, но предполагает конечную реализацию программы на других языках. Язык LAD отдает дань поклонникам стиля, когда программы похожи на электрические схемы релейной логики.

Язык FBD напоминает создание схем на логических элементах.

Наибольшее распространение в настоящее время получили языки LAD, STL и FBD. Охарактеризуем их более детально.

Язык релейных диаграмм (язык контактного плана) является графическим языком программирования. При создании программы в форме контактного плана используются графические компоненты, с помощью которых строятся логические сети, сходные с электрическими релейными схемами.

В LAD можно использовать следующие элементы и основные правила:

- контакты: каждый контакт представляет собой переключатель, через который при замкнутом состоянии может протекать ток.
- катушки: каждая катушка представляет собой реле, которое включается при протекании тока.
- блоки: каждый блок представляет собой функцию, которая выполняется, когда к блоку течет ток.
- сети: сеть образует полную цепь тока, который течет от левой шины тока через замкнутые контакты к катушкам или блокам – за счет этого они активизируются.

Графическое представление элементов программы воспринимается более легко, чем текстовое, поэтому этот язык и ему подобные максимально наглядны (особенно для тех, кто привык работать с электрическими схемами), позволяет исключить многие ошибки программирования свойственные «обычным» языкам. Этот язык можно рекомендовать для использования начинающими программистами.

Рассмотрим составление программы с помощью релейных диаграмм на нескольких небольших примерах.

Всем возможным типам переменным в контроллерах присвоены системные имена. Обозначения «I» и «Q» присвоены соответственно дискретным входам и выходам.

Для элементов логической схемы, алгоритма предусматриваются специальные обозначения, напоминающие обозначения элементов электрических схем. На рис. 1.20 показаны, последовательно включенные, ключ (вход) I0.1 и катушка реле (выход) Q0.1. Вертикальная линия условно определяет собой цепь подачи питания. Выход Q0.1 активизируется тут же, как только появится сигнал на входе I0.1.

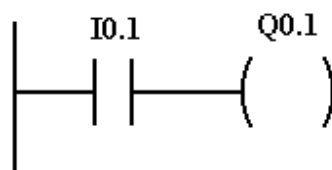


Рисунок 1.1 - Пример представления одной логической цепи в языке LAD.

Выход Q0.1 в программе на рисунке 1.1 активизируется тогда, когда появится сигнал на входе I0.3 и сигналы хотя бы на одном из входов I0.1 или I0.2.

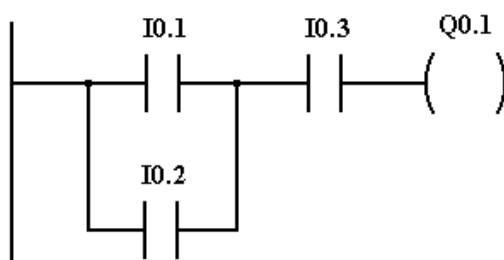


Рисунок 1.2 Пример представления в языке LAD цепи реализующей логическую функции 2ИЛИ-И.

На рисунке 1.2 представлена программа, включающая в себе блок таймера. Указанный в программе таймер, имеет дискретность отсчета 10 мс.

Уставка таймера равна 300 и он сработает через 3000 мс после поступления сигнала на вход I0.1. При этом замкнутся управляющий контакт таймера T32 и выход Q0.1 будет активизирован.

Все действия программы выполняются по очереди – шаг за шагом, поэтому программа состоит из двух цепей.

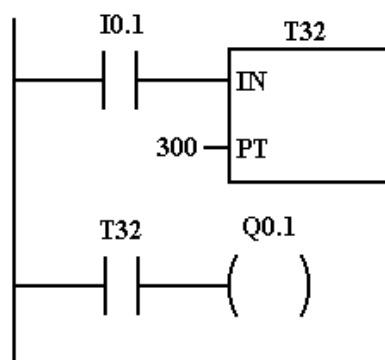


Рисунок 1.3- Пример представления в языке LAD программы с таймером.

Программа, представленная на рисунке 1.3 будет по очереди выполнять сложение, умножение и деление соответствующих переменных после поступления сигнала на вход I0.1.

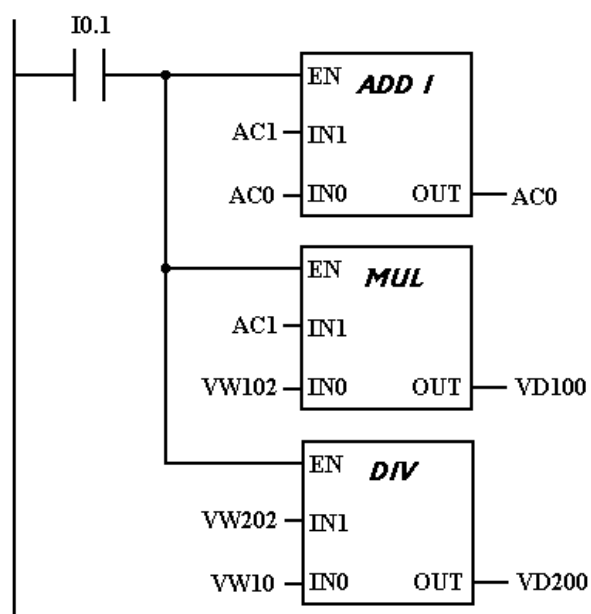


Рисунок 1.4 - Пример представления в языке LAD программы для выполнения арифметических операций

Язык функциональных блокковых диаграмм FBD (язык функционального плана) описывает функции между входными переменными и выходными переменными функциональных блоков (рисунок 1.4). Эти функции описываются в виде сочетания элементарных функциональных блоков. Каждый функциональный блок представляется прямоугольником, внутри которого имеется обозначение функции, выполняемой блоком.

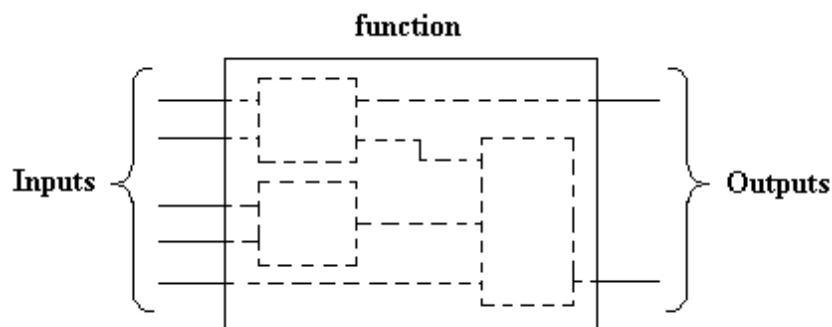


Рисунок 1.4-. Функциональный блок языка FBD.

Выход функционального блока может быть соединен с другими блоками. Логика программы выводится из соединений между этими блоками. Иными словами, выход одной команды (например, блока AND) может использоваться для того, чтобы разрешить другой команде (например, таймеру) создать необходимую логику управления. На рисунке 1.5 показан элемент программы, созданной при помощи языка функционального плана.

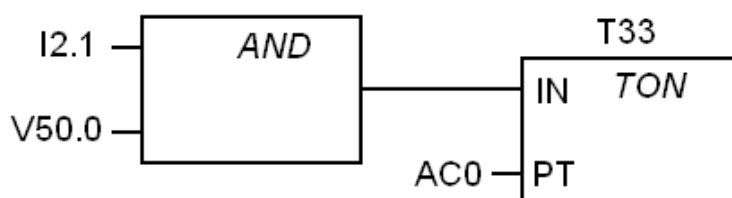


Рисунок 1.5- Вид программы на языке FBD

В качестве библиотечных блоков используются не только элементарные функции, но и алгоритмы П-, ПИ-, ПИД-регулирования, фильтрация сигналов, стабилизация заданных параметров. Набор из математических и статистических функциональных блоков позволяет просто организовать необходимые вычисления и обработку сигналов.

Приведем основные формальные правила языка FBD:

- функциональные блоки могут располагаться в поле программы произвольно;
- не может быть свободных (несоединенных) входов и выходов функционального блока;
- любая связь (NET) может иметь имя переменной;
- входы и выходы функциональных блоков, присоединенные к связям, имеющим одинаковые имена, считаются соединенными;
- очередность выполнения блоков в программе: слева направо, сверху вниз.

Переменные FBD-программ присоединяются к входным/выходным точкам функциональных блоков.

На входе FBD-блока может быть

- константное выражение;
- любая внутренняя или входная переменная;
- выходная переменная.

На выходе FBD-блока может быть любая внутренняя или выходная переменная.

В таблице 1.1 приведены примеры базовых функций языка FBD.

Программы на языке FBD напоминают электрические принципиальные схемы логических устройств и формально соблюдают алгоритмы их работы. Однако, несмотря на всю схожесть с электрическими схемами, язык FBD содержит метки, операторы условного и безусловного переходов, которые свойственны традиционным процедурным языкам программирования. Графический стиль представления программы в форме логических элементов удобно использовать для последующего составления ее блок-схемы.

Таблица 1.1 Примеры базовых функций языка FBD

Функции двоичного типа	NOT, AND, OR, XOR, SET, RESET
Функции управления программой	RETURN, GOTO, CALL, TSTART, TSTOP, GSTART, GSTOP
Арифметические функции	ADD, SUB, DIV, MUL
Функции сравнения	=, <>, >, <, >=, <=
Математические функции	ABS, EXPT, LOG, SQRT
Тригонометрические функции	ACOS, ASIN, ATAN, COS, SIN, TAN

Язык структурированного текста STL представляет собой язык программирования, в котором каждая команда программы содержит операцию, мнемоника которой определяет ту или иную функцию контроллера. Эти операции связываются в одну программу так, чтобы создать систему управления приложением.

По способу представления программа созданная в списке команд очень похожа на обычный ассемблер. Например, программа представленная на рисунке 1.6 в этой форме будет выглядеть следующим образом:

```
LD I0.1
+ AC1, AC2
= AC0
MUL AC1, VD102
= VD100
DEL VW202, VW10
= VD200
```

Рисунок 1.6 - STL

Пакеты по созданию программ для ПЛК, как правило, предоставляют возможность выбора языка представления. Например, Вы можете

предпочитать создавать программы в среде, более ориентированной на графическое представление, тогда как кто-то другой может предпочитать стиль языка ассемблера, основанный на текстовом представлении. Более того, обычно предоставляется возможность переходить от одной к другой форме представления уже созданной программы, то есть работать с ней в той форме представления, которая является более привычной и удобной.

В настоящее время в практике проектирования АСУ ТП появились так называемые «коробочные» решения в области проектирования АСУ ТП нижнего (уровня контроллеров) и верхнего уровня (уровень SCADA-систем), а также уровня предприятия (уровня АСУП - MES- и ERP-систем).

Однако применение SCADA-систем (особенно для разработчиков проектов) - это огромный набор технологий.

В настоящей работе рассматривается интегрированная среда разработки Trace Mode версии 6.10.

1 Разработка автоматизированной системы управления

1.1 Алгоритм выполнения курсовой работы

- 1 Производится выбор технологической установки (объекта), используемой в нефтегазодобывающем и топливно-энергетическом комплексе (например: блок насосный агрегатов с маслосистемой, блок сепараторов и так далее).
- 2 Разрабатывается схема автоматизации.
- 3 Составляется перечень сигналов для КИПиА
- 4 Производится выбор средств автоматизации (датчики, исполнительные устройства).
- 5 Производится выбор программируемого логического контроллера, на основании таблицы КИПиА.
- 6 Разрабатывается блок-схема алгоритма управления технологическим объектом.
- 7 Составляется программа для контроллера согласно разработанному алгоритму.
- 8 Процесс создания систем диспетчерского контроля и управления технологическим процессом
- 9 Оформляется пояснительная записка согласно ниже приведенным требованиям.

1.2 Этапы проектирования автоматизированных систем управления технологическим процессом

Исходными данными для курсового проекта являются:

- разделы регламента (или рабочих инструкций) с максимально возможным по детализации описанием технологии;
- разработка функциональной схемы объекта;
- различного рода ведомости и спецификации средств КИПиА;
- перечень контролируемых и регулируемых параметров;
- документация и внешний вид существующих щитов КИП с вторичными приборами;
- разводка параметров по существующим вторичным приборам;
- чертежи основных технологических агрегатов, помогающие лучше нарисовать мнемосхемы;
- заполненные образцы отчетных документов различной периодичности.

Создание системы АСУ ТП при проектировании делится на следующие этапы:

Процесс создания систем диспетчерского контроля и управления

технологическим процессом для программного пакета TRACE MODE можно разбить на следующие этапы:

- детализация технических требований на создаваемую диспетчерскую систему контроля и управления;
- разработка проектно – сметной документации в сокращенном или полном объеме;
- сбор и изучение исходных данных;
- создание статических частей графических экранов интерфейса оператора;
- заполнение графических экранов интерфейса оператора динамическими элементами;
- тестирование системы в автономном режиме (без УСО);
- тестирование системы в рабочем режиме (с УСО).

Проектирование систем автоматизации в TraceMode может быть осуществлено несколькими способами:

- проектирование «от шаблонов»;
- проектирование от технологии;
- проектирование от топологии;
- смешанное проектирование.

Порядок разработки АСУ можно изменить. К примеру, можно сначала создать шаблоны программ, а потом - шаблоны экранов.

Также к любому из этапов можно вернуться. К примеру, сначала создать шаблоны экранов и программ для одного объекта, а затем экран и программу - для другого объекта.

1.3 Примеры тем для курсовой работы:

Тема : «Применение микропроцессорного контроллера в системе автоматизации объекта нефтегазодобычи» для обучающихся на заочной форме обучения

Вариант	Объект
1	Сепаратора первой ступени на ДНС
2	Газосепаратор на ДНС
3	Трубчатый подогреватель нефти
4	Абсорбер в цеха осушки газа
5	Испаритель на УКПГ
6	Насосного агрегата на НПС
7	Электродегидратор на КСП
8	Низкотемпературный сепаратор на УКПГ
9	ГПА (нагнетатель, маслосистема) на КС
10	Концевая сепарационная установка на ЦПС
11	Десорбер на УКПГ

12	Применение микропроцессорной техники в системе автоматизации КВГ
----	--

Выбор варианта (для заочной, заочно сокращенной форм обучения) осуществляется по последней цифре № зачетной книжки.

Тема : «Применение микропроцессорного контроллера в системе автоматизации объекта нефтегазодобычи» для обучающихся дневной форме обучения

- 1 Применение микропроцессорной техники в системе автоматизации котла ДЕ
- 2 Применение микропроцессорной техники в системе автоматизации перекачивающего агрегата на НПС
- 3 Применение микропроцессорной техники в системе автоматизации блока сепараторов на ДНС-6
- 4 Применение микропроцессорной техники в системе автоматизации магистрального насосного агрегата (МНА) на НПС

2 Содержание пояснительной записки

2.1 Текстовая часть

- титульный лист (приложение А)
- список сокращений
- введение (актуальность проблемы, цель работы и основные задачи)
- общая характеристика объекта управления (предназначение и место в технологической цепи)
- структура автоматизированной системы управления объемы автоматизации, (датчики, исполнительные механизмы – контроллер, операторский интерфейс)
- обоснование выбора контроллера (обзор рынка контроллеров, критерии выбора, рассмотреть не менее трех контроллеров наиболее подходящего класса)
- разработка и описание алгоритмов управления технологической установкой
- тип используемого кабеля и интерфейса, для связи компонентов системы автоматизации
- обоснование выбора операторского интерфейса. (Произвести сравнительную характеристику не менее трех операторских интерфейсов).
- заключение (основные результаты)

- список использованных источников (по тексту ссылки на каждый источник)

2.2 Обязательный перечень приложений

- схема автоматизации технологической установки (выполненной с помощью программ AutoCAD, КОМПАС, Visio)
 - схема внешних электрических соединений контроллера (выполненной с помощью программ AutoCAD, КОМПАС, Visio)(очная форма обучения)
 - блок-схема алгоритма управления технологической установкой
 - перечень сигналов (таблица КИПиА)
 - текст программы для контроллера, языки программирования FBD, LD, STL,техноLT(от 5 до10 страниц)
 - мнемосхемы НМІ (для очной формы обучения)
- Пример приложений приведен в Приложениях Б,В,Г,Д.

2.3 Требования к разрабатываемому программному обеспечению

Разрабатывается согласно алгоритму управления технологической установкой, наличие комментариев на русском языке к каждой строке.

Обязательным является освещение следующих вопросов:

- Обработка информации, поступающей от датчиков с аналоговым выходным сигналом, от сигнализаторов, а также по сетевому протоколу (если такие предусмотрены проектом);
- запуск/останов агрегата, открытие/закрытие запорной арматуры, управление регулирующим органом (в зависимости от объекта);
- реакция на нештатные ситуации;
- составление конфигурации контроллера и описание модулей выбранного контроллера, а так же расчет энергопотребления модулей с учетом 15% запаса;
- заполнить таблицу КИПиА. (пример таблица 3.1).

Таблица 2.1 –Пример перечня сигналов.

№	Наименование параметра	Единицы измерения	Обозначение	Пределы измерения параметра	Наименование прибора	Поз. На схеме	Пределы измерения прибора	Класс точности	Диапазон вх/вых сигнала	Тип сигнала				
										DI	DO	AI	AO	FI
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

1	Уровень в РВС-1	М	LT	0,4-4	Альбатрос ДУУ2М	1а	4-25	±0,5	4-20 мА			+		
2	Сигнализатор уровня в РВС-1	М	LA	0,4-4	Rosemount 2120	2а	-	±1 мм	24 В	+				
3	Автоматическая задвижка 1	%	NS/HS	0-100	МЭП	Аз.1	0-100	-	24 В	++	++			

Продолжение таблицы 2.1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Автоматическая задвижка 2	%	NS/HS	0-100	МЭП	Аз.2	0-100	-	24 В	++	++			
5	Сигнализатор уровня в ЕП-1	М	LSA	0-1,8	Rosemount 2120	5а	-	±1 мм	24 В	+				

3 Оформление пояснительной записки

Записка должна быть выполнена с использованием компьютера и принтера на одной стороне листа белой бумаги формата А4.

Цвет шрифта должен быть черным, высота букв, цифр и других знаков — не менее 1,8 мм (14 кегль).

Междустрочный интервал – полуторный (для таблиц допускается одинарный), тип шрифта – Times New Roman, абзацный отступ – 1,25 см, выравнивание по ширине текста.

Текст записки следует печатать, соблюдая следующие размеры полей:

- нижнее - 20 мм,
- верхнее - 20 мм,
- левое - 30 мм,
- правое -10 мм.

Каждый пункт, подпункт и перечисления записывают с красной строки.

Содержание, список сокращений, введение, заключение, список использованных источников следует начинать с новой страницы.

Список использованных источников должен содержать порядковый номер, который в тексте записки (в месте его упоминания) следует приводить в квадратных скобках ([]).

Применяемые формулы и справочные данные обязательно должны иметь ссылки на источники их получения.

Схемы, рисунки, графики и тому подобное должны быть единообразными по оформлению и размещаться по тексту сразу же после ссылки на них.

Таблицы и рисунки имеют сквозную нумерацию и сопровождаются краткой надписью. Допускается нумерация таблиц и рисунков в пределах раздела.

Нумерация таблиц размещается в левом верхнем углу в одну строку через тире с названием таблицы. Заголовки граф и строк следует писать с прописной буквы.

Пример обозначения таблицы, нумерация которой производится в пределах раздела.

Таблица 3.1 – Название таблицы

	Название	Название	Название	Название

При переносе части таблицы на другую страницу пишут слово "Продолжение".

Пример:

Продолжение таблицы 3.2

	Название	Название	Название	Название

Допускается применять размер шрифта в таблице меньший, чем в тексте (12,10 кегель)

Рисунки следует располагать в работе непосредственно после текста, в котором они упоминаются впервые, или на следующей странице.

Рисунки должны иметь название, которое помещают под рисунком, ниже помещают поясняющие данные (подрисуночный текст).

Рисунок обозначается словом «Рисунок», которое ставится перед поясняющими данными. Если нумерация производится в пределах раздела, например, Рисунок 1.2., то цифра 1 - номер раздела, цифра 2 - порядковый номер рисунка.

Страницы в пояснительной записке должны иметь сквозную нумерацию, начиная с титульного листа (на титульном листе номер страницы не ставится), включая все страницы с рисунками, таблицами, приложениями. Нумерация страниц указывается внизу в рамках.

Приложения оформляются как продолжение пояснительной записки на последующих страницах. Каждое приложение следует начинать с новой страницы с указанием наверху по центру страницы прописными буквами слова «Приложение», обозначенное буквами, например «Приложение А, Б, В, Г, Д, Е, Ж, И, К, Л, М» (за исключением букв Ё, З, Й, О, Ч, Ь, Ы, Ъ).

Каждое приложение сопровождается заголовком.

Пояснительная записка должна быть выполнена согласно требованиям единой системы конструкторской документации (ЕСКД), ГОСТ 2.105-95 (Общие требования к текстовым документам), ГОСТ 2.106-96 (текстовые документы), ГОСТ 2.104-2006 (Основные надписи). Примеры оформ-

ления страниц пояснительной записки представлены в приложении Б.

При акцентировании внимания на определенных терминах, формулах, теоремах, разрешается использовать полужирный шрифт.

Опечатки, опiski и графические неточности, обнаруженные в процессе подготовки отчета, допускается исправлять подчисткой или закрашиванием белой краской и нанесением на том же месте исправленного текста (графики) машинописным способом или черными чернилами, пастой или тушью — рукописным способом.

Повреждения листов текстовых документов, помарки и следы не полностью удаленного прежнего текста (графики) не допускаются.

Фамилии, названия учреждений (организаций) и другие имена собственные в тексте пояснительной записки приводят на языке оригинала. Допускается указывать имена собственные и приводить названия учреждений (организаций) в переводе на русский язык с добавлением (при первом упоминании) оригинального названия. Имена следует писать в следующем порядке: фамилия, имя, отчество или – фамилия, инициалы через пробелы, при этом не допускается перенос инициалов отдельно от фамилий.

Не допускаются сокращения следующих слов и словосочетаний: «так как...», «так называемый...», «таким образом...», «так что...», «например...».

Пояснительная записка выводится на печать, переплетается пружиной с прозрачной обложкой либо в мягкую обложку.

4 Оформление списка литературы

- 1 Харазов В.Г. Интегрированные системы управления технологическими процессами [Текст] : учебное пособие. – М.: - СПб Профессия, 2009 г. – 592 с.
- 2 Андреев Е.Б., Попадько В.Е. Технические средства систем управления технологическими процессам в нефтяной и газовой промышленности [Текст]: Учебное пособие. – М.: РГУ нефти и газа им. И.М. Губкина, 2004. – 272 с.

4.1 Электронный ресурс

- 3 Официальный сайт фирмы ФГУП СПО «Аналитприбор» [Электронный ресурс]. URL: <http://www.gazanalizator.ru> (дата обращения: 10.02.2019).
- 4 Официальный сайт фирмы ОАО «РОСС» [Электронный ресурс]. URL: <http://www.ross.com.ru> (дата обращения: 14. 02.2019).

Рекомендуемая литература

- 1 Андреев Е.Б., Попадько В.Е. Технические средства систем управления технологическими процессами в нефтяной и газовой промышленности[текст]: Учебное пособие. – М.: РГУ нефти и газа им. И.М. Губкина, 2004. – 272 с.
- 2 Густав Олссон, Джангуидо Пиани. Цифровые системы автоматизации и управления. [текст]: – СПб.: Невский Диалект, 2001. – 557 с.
- 3 Парр Э. Программируемые контроллеры : руководство для инженера [текст]: / Э. Парр ; перевод. 3-го англ. изд. – М. : БИНОМ. Лаборатория знаний, 2007. – 516 с.
- 4 Агеев, Олег Алексеевич. Информационно-измерительная техника и электроника. Преобразователи неэлектрических величин [Текст] : Учебное пособие / О. А. Агеев. - 2-е изд., испр. и доп. - М. : Издательство Юрайт, 2018. - 158 с. - (Университеты России). - <http://www.biblio-online.ru/book/E9083298-A69E-4EAF-9F37-679125167739>
- 5 Макуха, Владимир Карпович. Микропроцессорные системы и персональные компьютеры [Текст : Электронный ресурс] : Учебное пособие / В. К. Макуха. - 2-е изд., испр. и доп. - М. : Издательство Юрайт, 2018. - 175 с. - (Университеты России). - Режим доступа: <http://www.biblio-online.ru/book/4F29CE67-3B2B-4289-BA38-9FDE247F3D62>
- 6 Гуревич, Владимир Игоревич. Уязвимости микропроцессорных реле защиты. Проблемы и решения [Электронный ресурс] / В. И. Гуревич. - Москва : Инфра-Инженерия, 2017. - 256 с : ил. - Режим доступа: <https://e.lanbook.com/book/95769>
- 7 Кузяков, Олег Николаевич. Проектирование систем на микропроцессорах и микроконтроллерах [Текст : Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по направлению подготовки 220400 - Управление в технических системах УрФО / О. Н. Кузяков ; ТюмГНГУ. - Тюмень : ТюмГНГУ, 2014. - 104 с. : ил., табл. - Режим доступа: <http://elib.tyuiu.ru/>

Приложение А
Оформление титульного листа

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра «Кибернетических систем»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе по дисциплине
«проектирование микропроцессорных систем автоматизации»
на тему:

**ПРИМЕНЕНИЕ ПРОГРАММИРУЕМОГО ЛОГИЧЕ-
СКОГО КОНТРОЛЛЕРА В СИСТЕМЕ АВТОМАТИЗАЦИИ
НЕФТЕПЕРЕКАЧИВАЮЩЕЙ СТАНЦИИ**

(шифр направления, номер приказа , год)

РУКОВОДИТЕЛЬ:

_____ (должность Ф.И.О.)

РАЗРАБОТЧИК:

Студент группы № группы

_____ (Ф.И.О.)

Тюмень 2019

Приложение Б

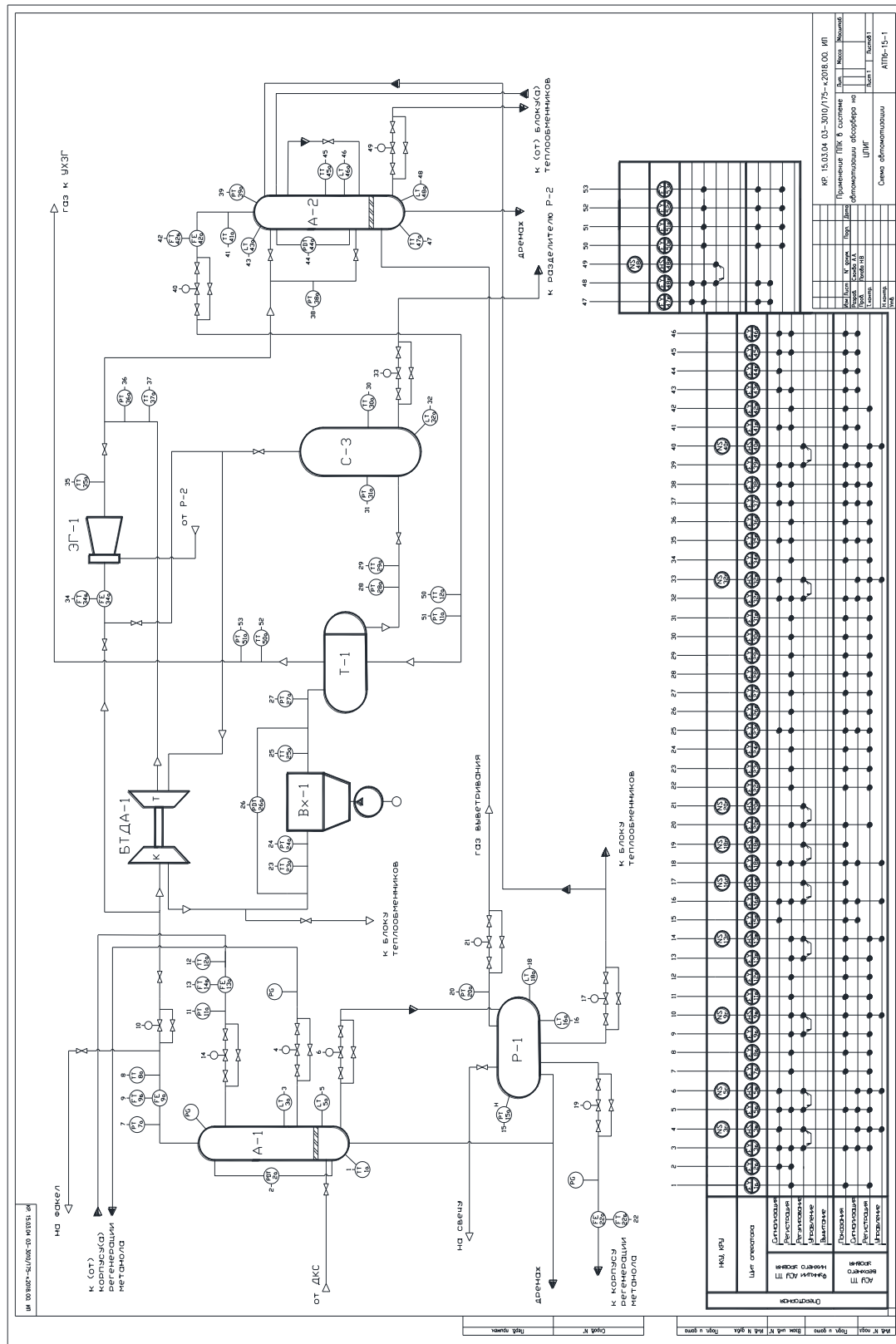
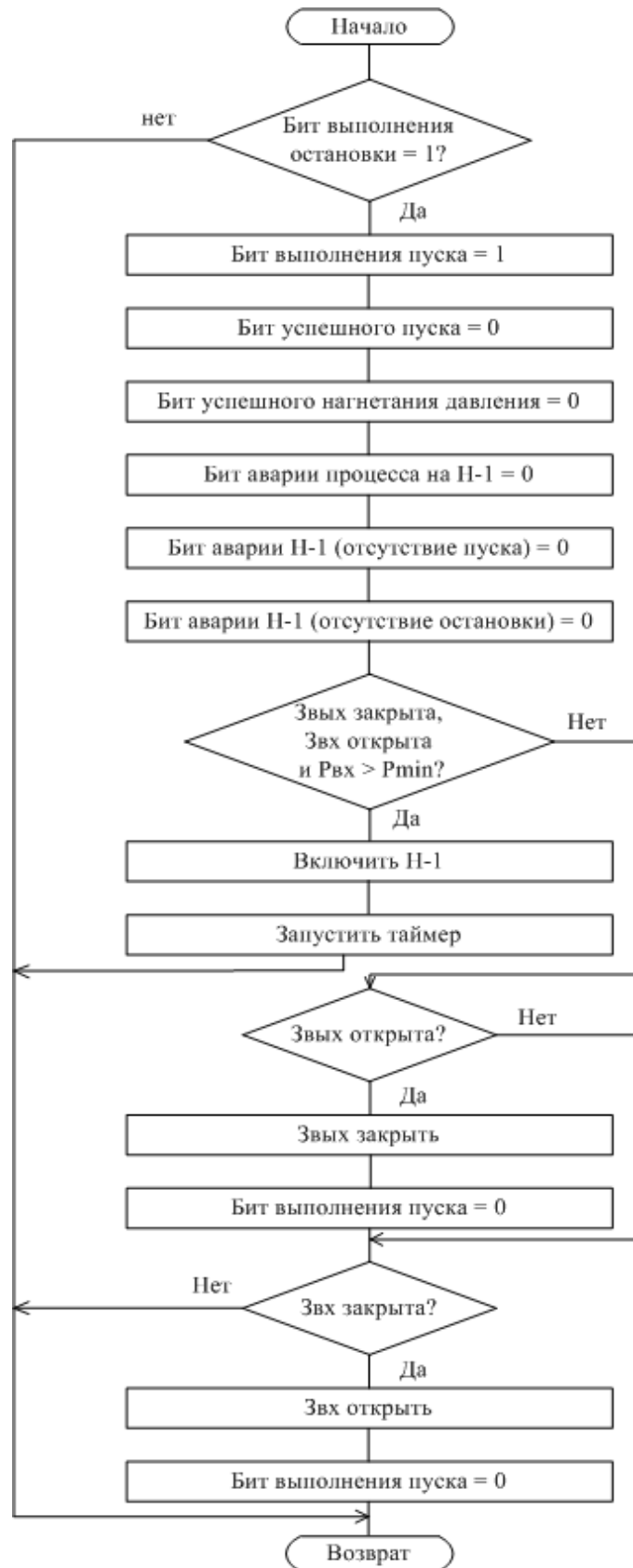


Рисунок Б.1- Схема автоматизации

Приложение В

Алгоритм пуска подпорного насоса



Приложение Г

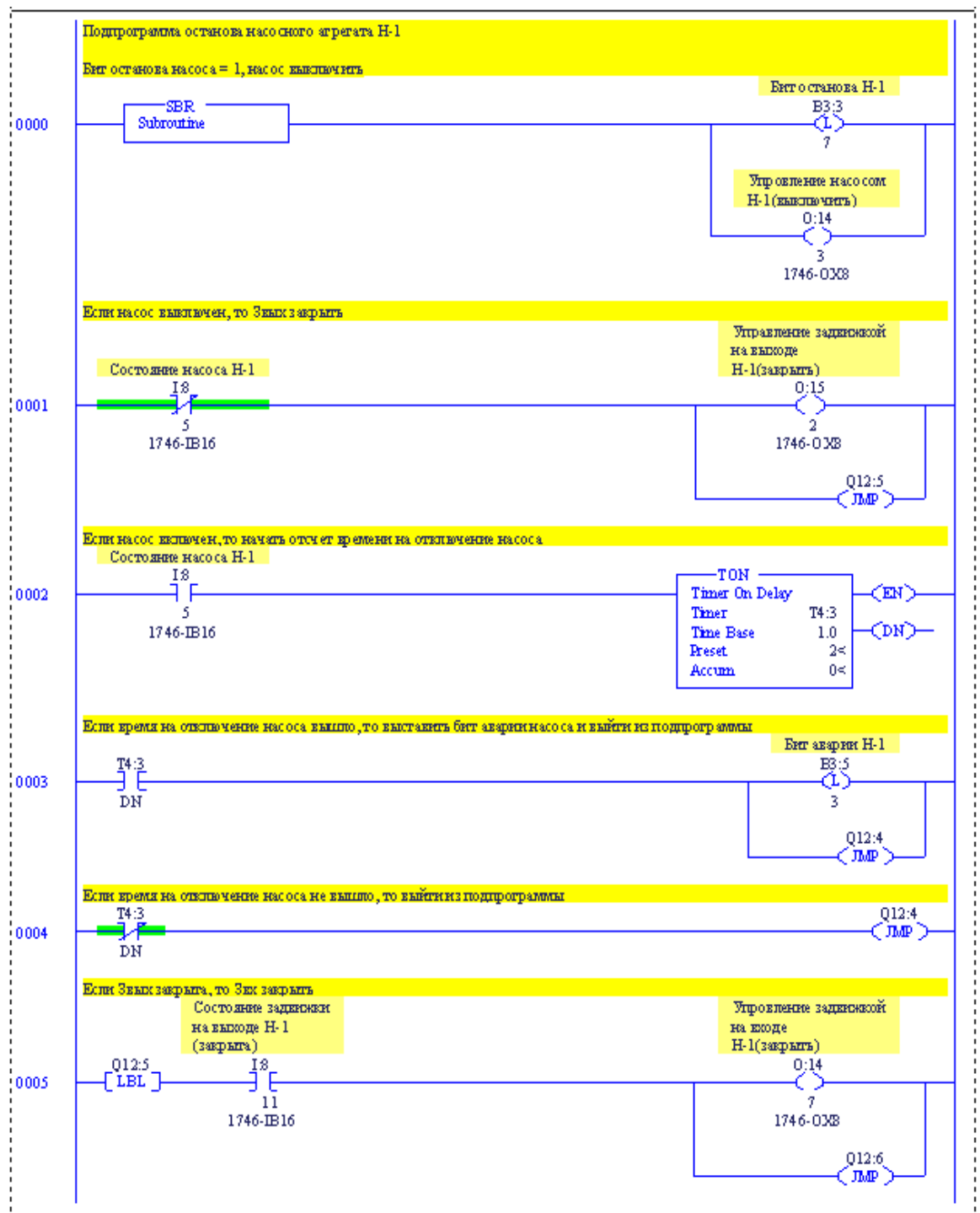


Рисунок Г.1- Подпрограмма остановки насоса LD

Приложение Г

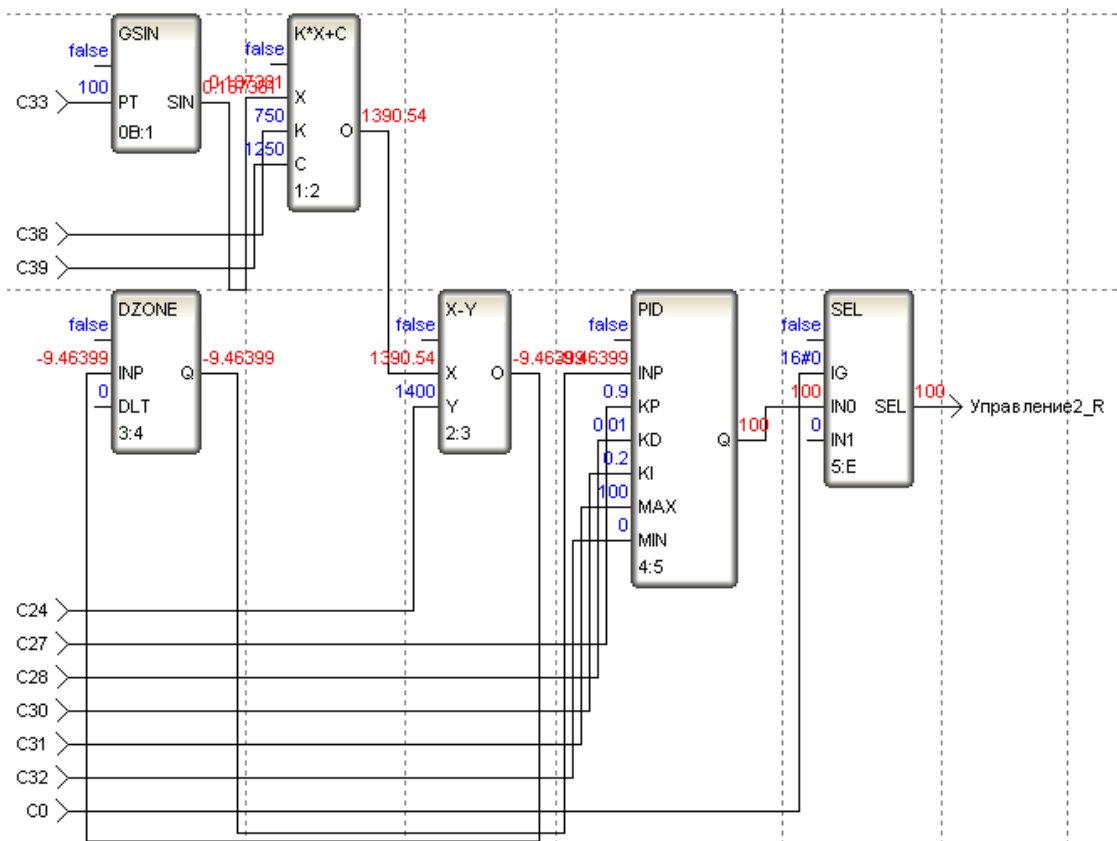


Рисунок Г.2 - Подпрограмма ПИД-регулирования на FBD

PROGRAM

```

VAR_INPUT Газ_низко : REAL; END_VAR
VAR_INPUT Газ_высоко : REAL; END_VAR
VAR_INPUT Вода_больше : REAL; END_VAR
VAR_INPUT Вода_меньше : REAL; END_VAR
VAR_OUTPUT Клапан_1 : REAL; END_VAR
VAR_OUTPUT Клапан_2 : REAL; END_VAR
    if Газ_низко == 0
    then Клапан_1 =0;
    elsif Газ_высоко == 0
    then Клапан_1 =0;
    elsif Вода_больше == 0
    then Клапан_1 =0;
    else Клапан_1 =1;
    end_if; END_PROGRAM

```

Рисунок Г.3 - Подпрограмма на языке Техно LT

Приложение Д

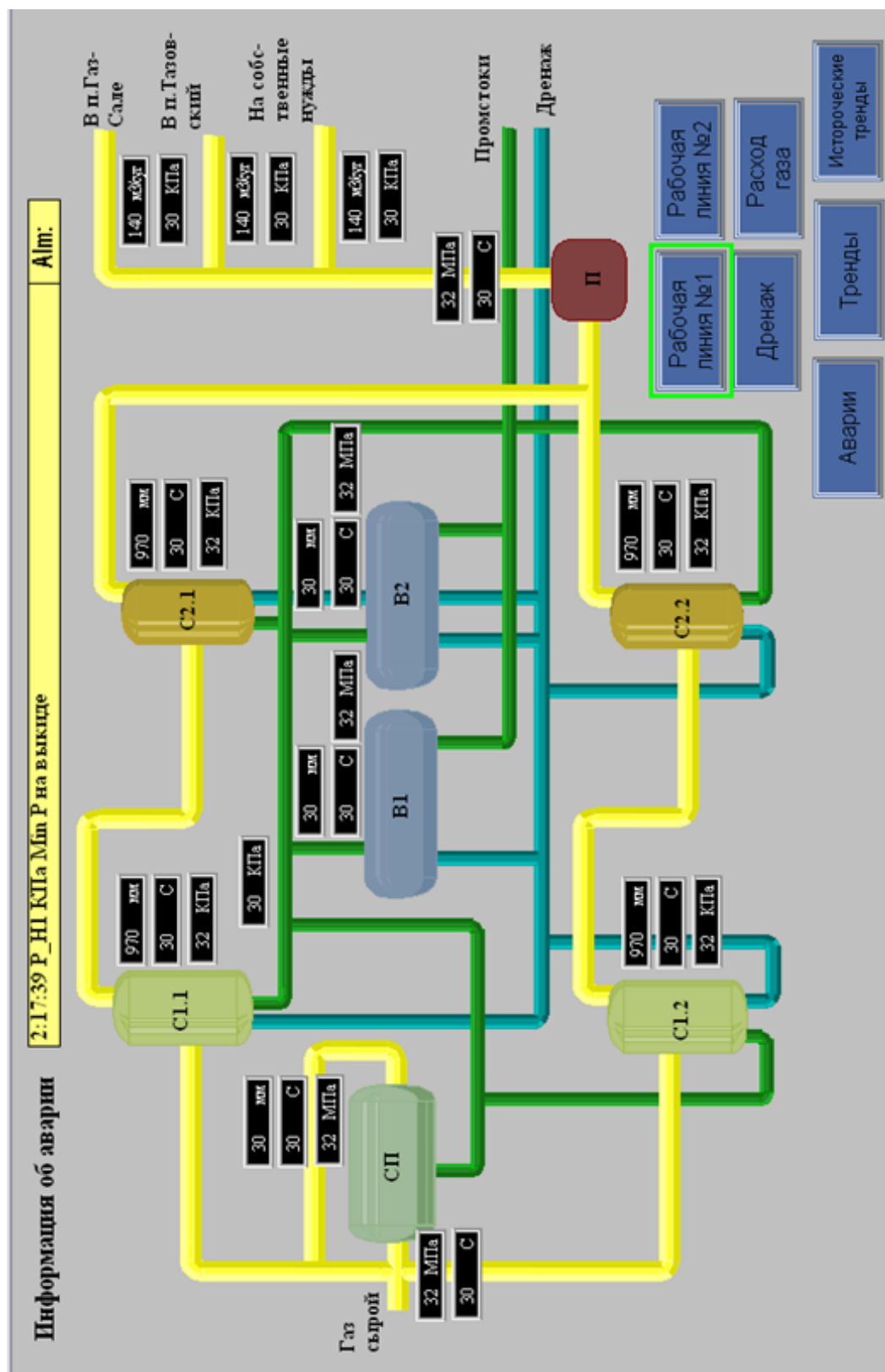


Рисунок Д.1 – Экран оператора.Главный экран

Приложение Е

Краткая информация для написания программ на языках FBD и IL

Для размещения блоков на поле окна используется визуальный язык программирования алгоритмов Техно FBD.

Блок – это графическое изображение вызова одной из функций, реализованных в языке. Это могут быть либо стандартные, встроенные в систему функции либо разработанные пользователем на языке Техно IL. Пример блока ПИД-регулятора приведен на рисунке Е.1

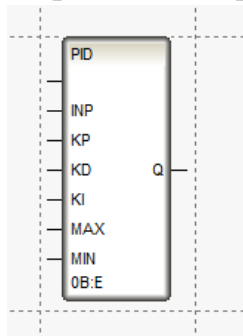


Рисунок Е.1 – Блок ПИД-регулятора

Каждый функциональный блок имеет следующие атрибуты: выполняемая функция, номер, входы и выходы. Для данного блока это: функция – PID регулирование, номер на диаграмме – 1 : 2, его входы – INP, KP, KD, KI, MAX, MIN, его выход – Q.

Функция блока выбирается перед размещением блока.

Выбор функции блока осуществляется в диалоге Меню FBD, показанном на следующем рисунке. Здесь сначала указывается функциональный раздел (в данном случае «Регулирование»), а затем в нем выбирается требуемая функция (в данном случае PID).

Номер функционального блока устанавливается системой автоматически при его размещении в рабочем поле.

Он используется только для индикации очередности выполнения функциональных блоков при вызове программы. Входы в функциональный блок на его изображении всегда расположены слева, а выходы – справа. При вычислении блока над переменными, связанными с его входами, осуществляются действия, определенные функцией блока. Полученные в результате значения присваиваются переменным, связанным с выходами блока. Вход блока может быть связан с переменной или константой, а может быть свободным.

На диаграмме они выводятся ярко-зеленым цветом. Назначение всех входов и выходов определяется функцией блока.

Исключением является только первый вход.

Он для всех блоков управляет их пересчетом. Далее все входы, используемые для формирования значений выходов, будут называться функциональными, а вход, управляющий пересчетом, – входом блокировки. Все функциональные входы и выходы блоков имеют обозначения, включающие в себя до трех символов. При отладке программы в режиме эмуляции рядом с каждым входом и выходом функционального блока выводится его текущее значение. При этом для входов выводимые значения заменяют их обозначения.

Инверсия может быть применена к любым входам блоков, реализующих логические операции: & (логически умножить), I (логически сложить), ^ (выполнить исключающее логическое сложение), и к битовым входам блоков триггеров: RS, SR, TOFF, TON, TP.

Инверсия на выходах осуществляется для всех функциональных блоков без исключения, но результат при этом превращается в логическую 1 или 0.

Для выполнения инверсии надо в режиме редактирования нажать ПК на изображении входа или выхода. При этом в его основании появится пустой круг – признак инвертирования.

На одном функциональном блоке инвертироваться может только один функциональный вход.

Существуют два вида представления данных для входов и выходов функциональных блоков:

- FLOAT;
- HEX.

Первый из них предназначен для работы с аналоговыми переменными, представленными в виде числа с плавающей точкой одинарной точности.

Второй – для работы с целыми 16-битовыми переменными. При связывании входов и выходов с разными видами представления осуществляется преобразование данных из одного вида в другой.

Типы входа и выхода функционального блока могут иметь следующие форматы:

- свободен;
- блок;
- константа;
- аргумент.

Тип входа/выхода определяет переменную, с которой он будет связан. Это может быть либо внутренняя переменная программы, либо константа, либо внешняя переменная, передаваемая в качестве аргумента при вызове программы.

Сохранение и вставка FBD-программ. FBD-программу можно сохранить в файл и затем вставить в другой узел того же или другого проек-

та. Ее можно сохранить также в виде текста на структурированном языке международного стандарта МЭК 1131-3. Это позволяет использовать данную программу в других приложениях.

Сохранение. Для сохранения FBD-программы в файл надо ее выбрать в списке диалога FBD-программа. Затем следует выполнить команду. Сохранить из меню Программа данного диалога. При этом на экране появляется стандартный диалог выбора файла. Указанная FBD-программа будет сохранена в заданный здесь файл, которому автоматически дается расширение «.frm». Сохранение в виде текста. Для сохранения FBD-программы в текстовом виде надо выполнить команду «Сохранить как текст» меню Программа. Создаваемый файл в этом случае будет иметь текстовый формат и расширение «.frm».

Вставка из файла. Любую сохраненную в файл FBD-программу можно вставить в проект. Для этого, находясь в диалоге FBD-программа, надо выполнить команду «Открыть» из меню Программа. В результате на экран будет выведен диалог выбора файла. После указания требуемого файла записанная в нем программа будет добавлена в текущий проект и ее имя появится в списке FBD-программ. Переход к редактированию FBD-программ. Для перехода к редактированию выбранной FBD программы следует выполнить команду «Редактировать» из меню Редактирование диалога FBD-программа или нажать левую кнопку мыши на соответствующей кнопке того же диалога. При этом в рабочее поле окна редактирования FBD-программ выводится диаграмма из функциональных блоков, реализующая алгоритм указанной программы.

Редактирование связей. Режим редактирования связей позволяет настроить структуру обработки данных в программе. В этом режиме доступны следующие операции: настройка входов и выходов функциональных блоков; связывание входов и выходов функциональных блоков и удаление связей; удаление блоков; перемещение блоков. Для настройки входа или выхода функционального блока надо дважды нажать на нем левой клавишей мыши. При этом на экране появится диалог.

Описание переменной. Этот диалог позволяет задать тип переменной, вид представления и значение. На приведенном ниже рисунке показан вид этого диалога. Кроме того, в этом диалоге можно ввести комментарий. Этот комментарий выводится в рамке рядом с настраиваемым входом или выходом, так же в бланках диалога «Реквизиты» для переменных типа «Аргумент» и «Константа» при настройке вызова программы. Связывание входов и выходов функциональных блоков. При разработке алгоритмов, выполняющих несколько последовательных действий, необходимо передать результаты вычислений одного блока другому. Для этого надо связать соответствующие входы и выходы этих блоков.

Для установления такой связи следует нажать левой кнопкой мыши на одном из ее концов и, удерживая кнопку мыши нажатой, перевести кур-

сор в область другого конца связи. При этом соответствующие вход и выход функциональных блоков соединятся линией.

Связанным входам и выходам автоматически присваивается тип «Блок». Связывать можно не только вход с выходом, но и вход со входом. В этом случае линии соединения не проводятся, однако связь устанавливается. При выделении любого из связанных между собой входов цвет всех остальных также меняется на красный. Чтобы посмотреть все связи выхода функционального блока, следует нажать на нем левой кнопкой мыши. При этом линии связи этого выхода с другими блоками выделяются красным цветом. Нажатие левой кнопкой мыши на входе блока не выделяет его связи. Однако при этом выделяются все входы других блоков, связанных с тем же источником.

Для удаления связи между блоками надо нажать левую кнопку мыши на входе в блок. Нажатие клавиши «DEL» после этого приводит к удалению выделенной связи. После удаления связи входам блоков присваивается тип «Свободен», а выходам – «Блок». Чтобы удалить функциональный блок из редактируемой программы, его надо сначала выделить. Выделение блока осуществляется нажатием левой кнопкой мыши на его изображении. Нажатие клавиши «DEL» после этого приводит к удалению из программы выделенного блока. Все связи этого блока также удаляются. Входам блоков, связанных с выходами удаляемого блока, присваивается тип «Свободен», а выходам – «Блок».

Описание языка техно IL

Текст программы, разработанной на Техно IL, представляет собой последовательность инструкций.

Каждая инструкция включает в себя описатель действий и операнды. Максимальное количество инструкций одной программы равно 12000. Описателями действий могут быть функции, операторы и операции. В качестве операндов могут использоваться переменные и константы. Переменные языка Техно IL. Язык Техно IL позволяет использовать несколько типов переменных.

Для всех типов переменных структура их имен остается одинаковой: первый символ в имени определяет тип переменной, далее за символом типа без пробела следует ее номер.

Для различных типов переменных Техно IL определены следующие идентификаторы:

- I – входные переменные;
- Q – выходные переменные;
- E – статические переменные;
- W – статические глобальные переменные;
- F – динамические переменные.

Количество доступных в программе переменных разного типа определяется типом II-программы. Кроме описанных типов, в Техно II используются еще две системные переменные:

- result – аккумулятор (X);
- CMP – признак истинности.

Константы в II-программах. В тексте программ можно использовать константы. Константы могут быть двух типов:

- целые числа в диапазоне от –127 до 127;
- любые другие числа.

Эти константы могут записываться как в десятичном, так и в шестнадцатеричном виде. Шестнадцатеричная константа должна начинаться с 0х.

Ограничений на число констант первого типа не накладывается, поэтому их в программе может быть любое количество. Количество доступных для использования констант второго типа зависит от типа программы. Для «FB» их можно использовать не более 10, а для «PRG» – не более 40.

При этом совпадающие по значению константы считаются за одну. Например, если в тексте программы три раза применяется константа 10,5, то считается, что использована только одна константа.

Операнды языка Техно II. Каждая инструкция программы содержит оператор и операнды. Оператор задает действие, которое надо осуществить с операндами. В качестве операндов в II-программе могут использоваться все описанные выше переменные, кроме CMP, а также константы. Синтаксис записи операций.

Язык инструкций предполагает следующую структуру записи операций. В одной строке записывается сначала мнемоническое обозначение операции, затем после символа разделителя (например, пробел) записывается операнд.

Техно II позволяет использовать два операнда и мнемоническое или символьное обозначение операции. Кроме того, обозначение операции может быть помещено между операндами. Например, операция сложения переменной Q1 с переменной I2 с записью результата в первую из них в соответствии со стандартом записывается следующим образом:

*LD Q1
ADD I2
ST Q1*

В Техно II эта операция может быть записана так же, а может – существенно компактнее:

ADD Q1 I2 ИЛИ + Q1 I2 ИЛИ Q1 + I2.

В одной строке программы можно записать несколько операций. В этом случае их следует разделять символом «;».

Полный список функций, меток, операторов, комментариев языка Техно IL приведен в электронной справке ТРЕЙС МОУД.

Создание новых блоков, язык техно IL. Язык инструкций (Техно IL) – это текстовый язык ТРЕЙС МОУД для разработки программ, реализующих функции обработки данных и управления.

Он является расширением IL-языка международного стандарта IEC 1131-3. Это расширение позволяет использовать более простой и интуитивный синтаксис, дополнительные функции и операторы, а также двухадресный режим.

Вызов IL-программ. Программы, разработанные на Техно IL, могут вызываться двумя способами. Первый способ – из FBD-программ. В этом случае Техно IL используется для программирования функций блоков, добавляемых в систему. Второй способ вызова IL-программ – это запуск их параллельно с пересчетом базы каналов.

Функциональные блоки. Язык Техно FBD имеет около 150 стандартных типов функциональных блоков. Кроме того, он поддерживает использование произвольно программируемых функциональных блоков. Их максимальное количество, которое можно одновременно включить в систему, равно 54. Программирование таких блоков осуществляется на языке Техно IL.

Чтобы добавить в систему функциональный блок с новой функцией, надо в диалоге Техно IL разработать программу, реализующую требуемую функцию и подключить ее к системе. При этом тип программы надо установить «FB». Вызов IL-программы, реализующей функцию данного блока, будет осуществляться при каждом вычислении FBD-программ, где он используется.

Тип IL-программы определяет способ ее вызова монитором реального времени, а также ограничения, которые накладываются на количество доступных переменных и операторов. Он задается выбором из списка в соответствующем поле диалога

Техно IL и может принимать следующие значения:

- FB – программирование блока для Техно FBD;
- PRG – метапрограмма.

Первый из них используется при программировании собственных функциональных блоков для языка Техно FBD. Тип «PGR» устанавливается для метапрограмм, запускаемых параллельно с пересчетом базы каналов.

Тексты программ типа «PGR» должны начинаться с ключевого слова «PROGRAM».

Имена IL-программы. Для каждой IL-программы определены два имени: основное и дополнительное. Они вводятся в соответствующих полях показанного выше диалога Техно IL. Текст обоих имен IL-программы может включать в себя 7 любых символов.

Первое из них используется для идентификации программы. Оно выводится во всех списках редактора базы каналов при ссылках на IL-программы. Назначение дополнительного имени зависит от типа программы. Для типа программы FB оно задает название функции программируемого блока. Это название выводится в верхней части функционального блока при его размещении в рабочем поле редактирования FBD-программ.

Если IL-программа имеет тип «PRG», то дополнительное имя используется для задания его номера. Номер программы может быть установлен от 0 до 15. Эти атрибуты используются для идентификации программы, определения доступа к ней и способа вызова. Они задаются в Диалоге Техно IL. Вход в этот диалог осуществляется по команде «Создать» меню Техно IL.

Создание IL-программ. Для создания новой IL-программы надо выполнить команду «Создать» меню Техно IL редактора базы каналов. При этом на экран выводится диалог Техно IL. Здесь в соответствующих полях следует ввести тип и имена новой программы (основное и дополнительное).

После этого надо набрать текст программы. Для набора текста предусмотрено специальное окно данного диалога. После этих операций новая программа может быть оттранслирована и добавлена в систему или сохранена в виде файла. В обоих случаях она будет в дальнейшем доступна для редактирования.

Сохранение в файл и вставка IL-программ. Любая IL-программа может быть сохранена в отдельный файл в текстовом формате. Для этого надо, находясь в диалоге Техно IL, выполнить команду «Сохранить как» из его меню «Файл». При этом на экран выводится стандартный диалог выбора файла.

Имя файла для сохранения и путь к нему можно указать любые. Однако по умолчанию предлагается использовать файл с именем, образованным из основного имени программы, и поддиректорию «ASM» рабочей директории ТРЕЙС МОУД.

Тексты программ на языке Техно IL сохраняются в файлы с расширением «il». Сохраненная ранее программа может быть снова загружена в редактор. Для этого надо, находясь в диалоге Техно IL, выполнить команду «Открыть» из его меню «Файл». При этом на экран выводится стандартный диалог выбора файла.

Учебное издание

**ПРОЕКТИРОВАНИЕ МИКРОПРОЦЕССОРНЫХ СИСТЕМ
АВТОМАТИЗАЦИИ**

Методические рекомендации по выполнению курсовой работы

Составители
ПОПОВА Надежда Владимировна,
КОЗЛОВ Василий Владимирович

В авторской редакции

Подписано в печать . Формат 60х90 1/16. Усл. печ. л.
Тираж 20 экз. Заказ №

Библиотечно-издательский комплекс
федерального государственного бюджетного образовательного
учреждения высшего образования
«Тюменский индустриальный университет».
625000, Тюмень, ул. Володарского, 38.

Типография библиотечно-издательского комплекса.
625039, Тюмень, ул. Киевская, 52.