

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Оренбургский государственный университет»

А. И. Сергеев, А. М. Черноусова, А. С. Русяев

ПРОГРАММИРОВАНИЕ КОНТРОЛЛЕРОВ СИСТЕМ АВТОМАТИЗАЦИИ

Рекомендовано ученым советом федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет» в качестве учебного пособия для студентов, обучающихся по программам высшего образования по направлению подготовки 15.04.04 Автоматизация технологических процессов и производств

Оренбург
2016

УДК 004.41/92(076)
ББК 32.973-018я7
С32

Рецензент – кандидат технических наук В. В. Тугов

С32 **Сергеев, А. И.**
Программирование контроллеров систем автоматизации: учебное пособие / А. И. Сергеев, А. М. Черноусова, А. С. Русяев; Оренбургский гос. ун-т. – Оренбург: ОГУ, 2016. – 125 с.
ISBN 978-5-7410-1649-7

Учебное пособие содержит теоретические сведения о программируемых логических контроллерах (ПЛК), программируемых реле и других элементах систем автоматизации. Представлен материал о принципах программирования различных устройств с помощью специализированных и универсальных средств разработки программного обеспечения. В пособие включены методические указания по пяти лабораторным работам. Для самоподготовки по каждой теме приводятся контрольные вопросы.

Учебное пособие предназначено для изучения элементов систем автоматизации на основе применения ПЛК студентами всех форм обучения по программам высшего образования направления подготовки 15.04.04 Автоматизация технологических процессов и производств. Учебное пособие может быть использовано обучающимися других направлений подготовки, а также инженерно-техническими работниками предприятий, занимающихся разработкой систем автоматизации.

Учебное пособие подготовлено в рамках реализации проекта по совершенствованию содержания и технологий целевого обучения студентов в интересах организаций оборонно-промышленного комплекса («Новые кадры ОПК–2016»)

УДК 004.41/92(076)
ББК 32.973-018я7

ISBN 978-5-7410-1649-7

© Сергеев А. И.,
Черноусова А. М.,
Русяев А. С., 2016
© ОГУ, 2016

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 Лабораторная 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 лабораторной 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 Лабораторная работа 2 «Разработка человеко-машинного интерфейса системы управления с применением панели оператора ОВЕН СП270	75
2.4.1 Выполнение лабораторной 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 лабораторной 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 Лабораторная 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 Лабораторная 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.

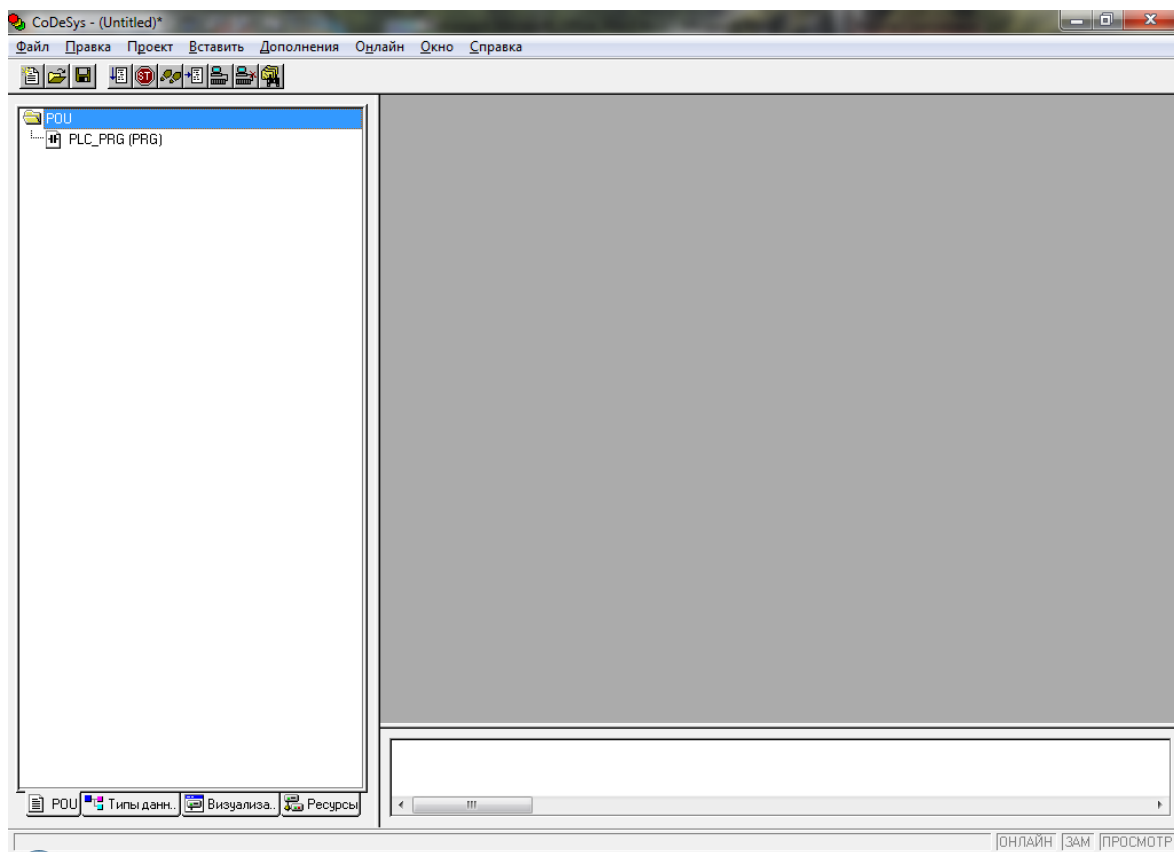


Рисунок 1.1 – Стартовое окно *CoDeSys 2.3*

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.A-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».

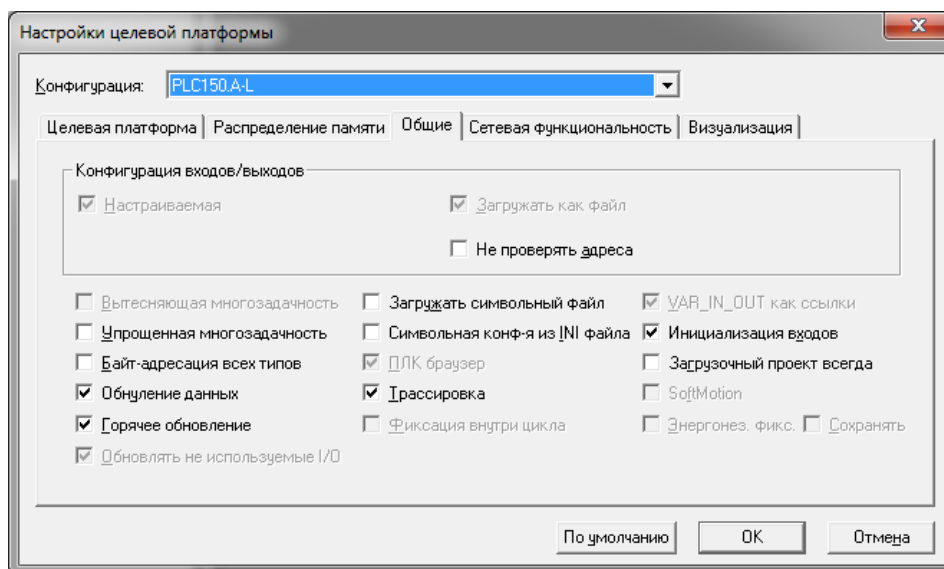


Рисунок 1.3 – Окно настройки параметров контроллера

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».

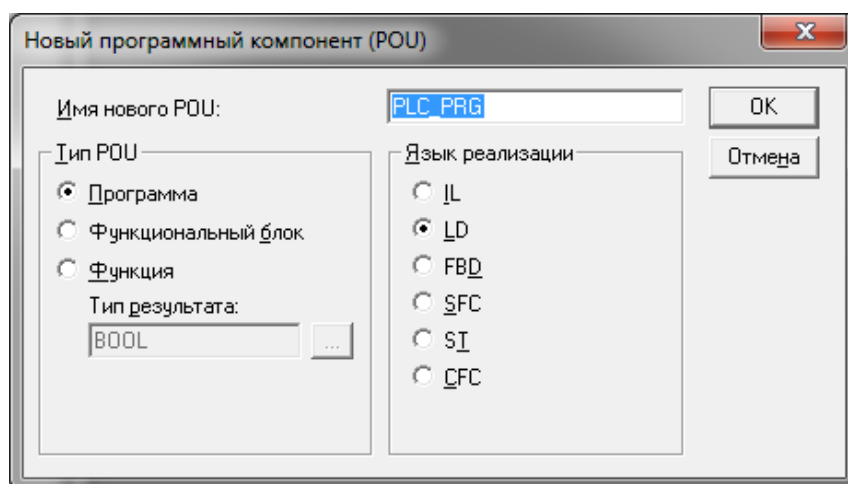


Рисунок 1.4 – Окно настройки *POU* проекта

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.

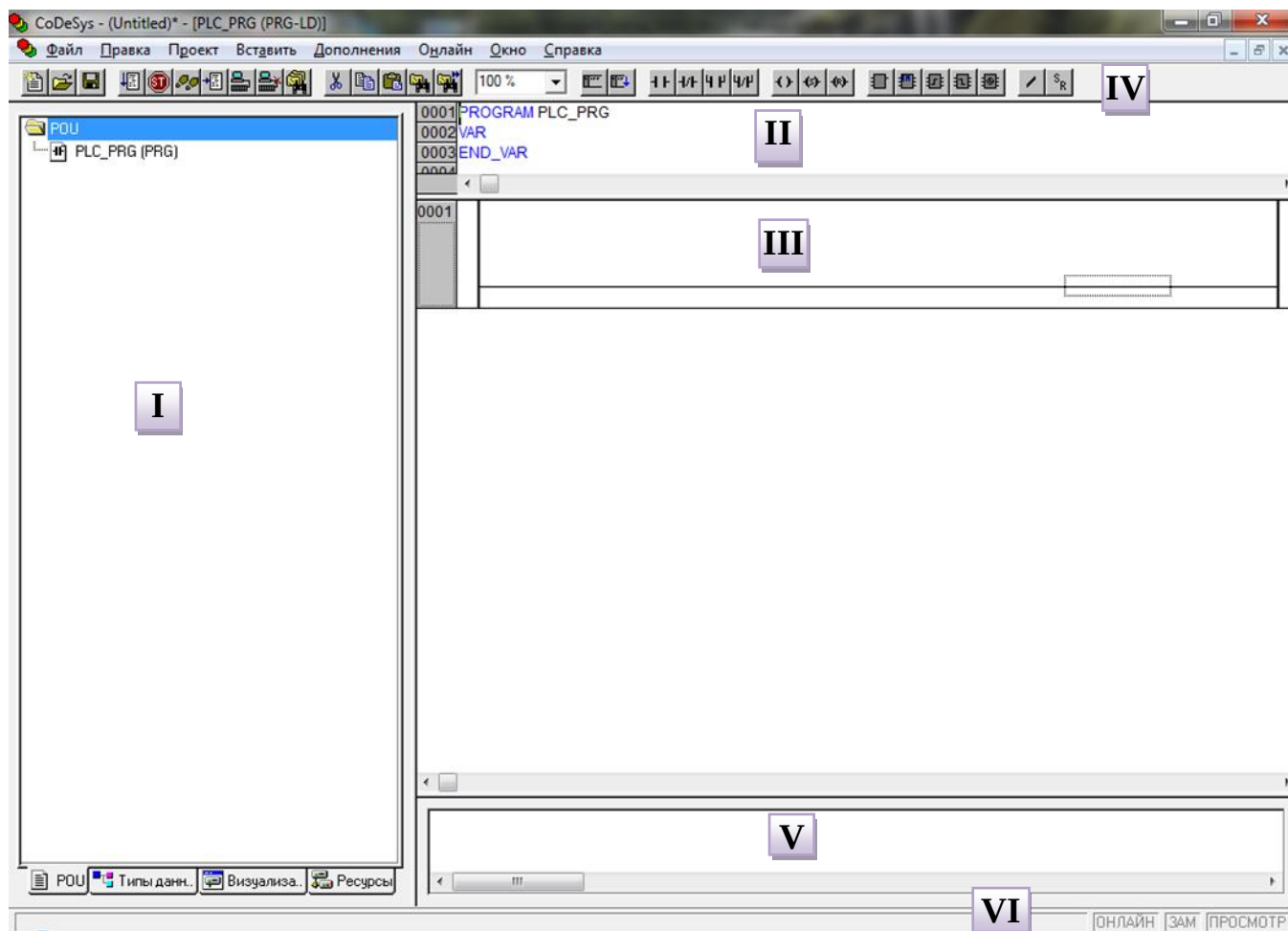


Рисунок 1.5 – Рабочее окно программы *CoDeSys* 2.3

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).

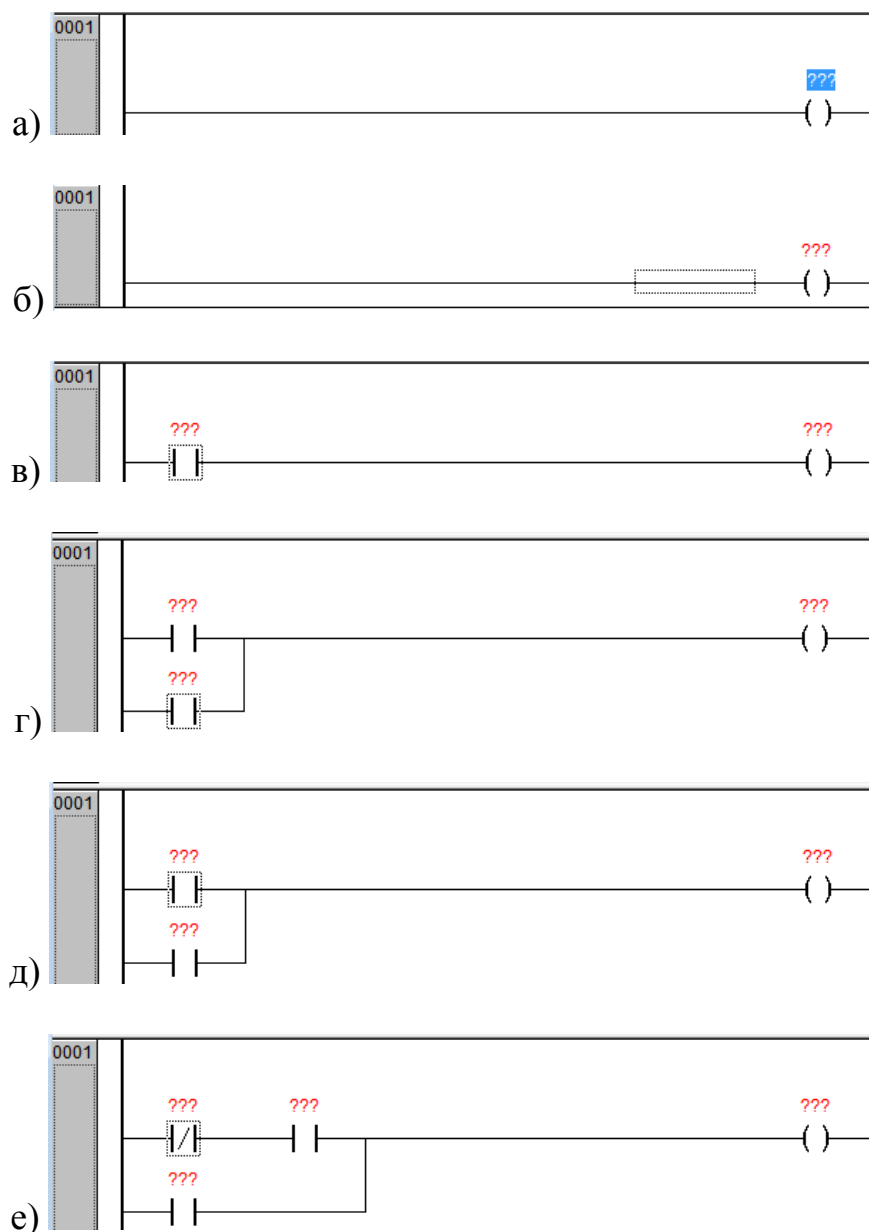


Рисунок 1.7 – Порядок программирования управляющей программы

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).

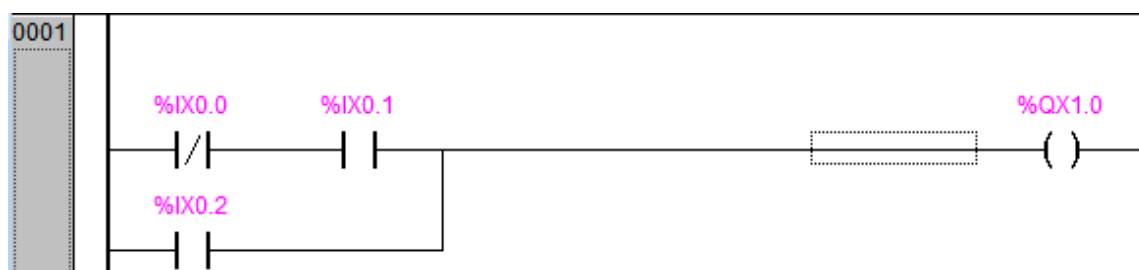


Рисунок 1.8 – Присвоение адресов компонентам программы

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».

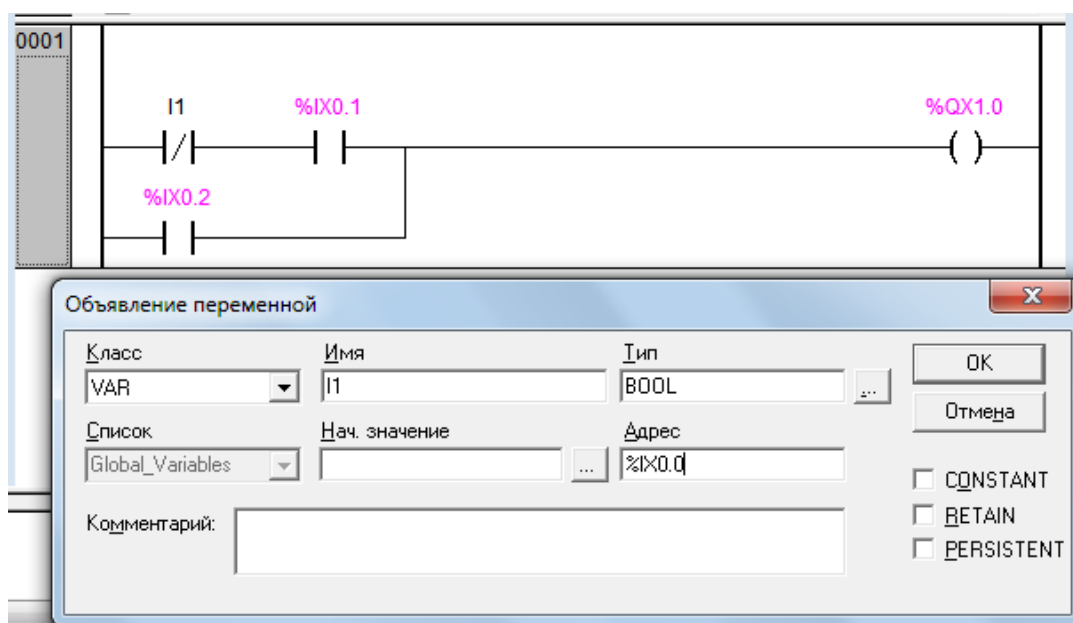


Рисунок 1.9 – Задание переменных



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 символичный файл 'UNTITLED.SYM'
Ошибка 3622: Невозможно создать символичный 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).

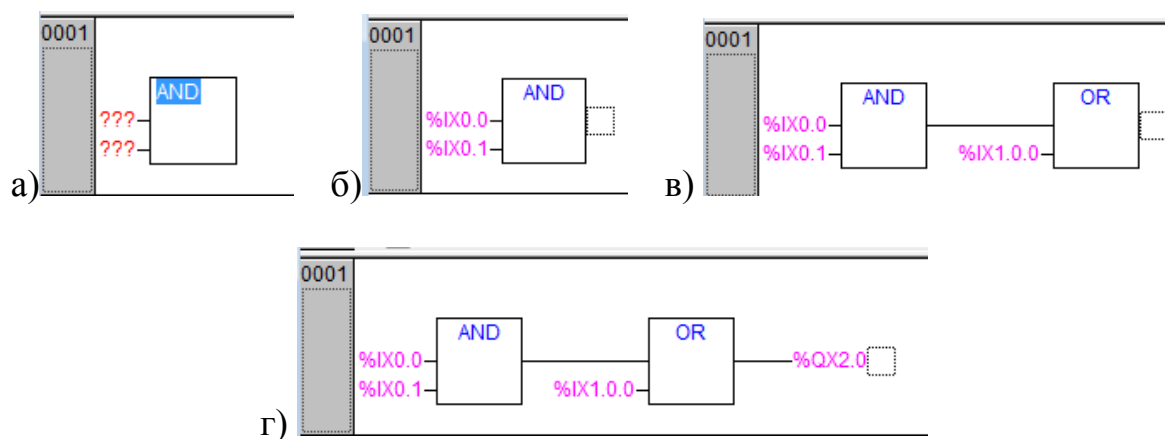


Рисунок 1.13 – Программа на языке FBD

Рассмотрим вставку функции на примере создания следующей программы: 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:

1) установить выходную обмотку с адресом *%OX1.0* и контакт с адресом *%IX0.0*;

2) установить курсор в новом сегменте. Лево́й клавишей мыши на панели инструментов нажать пиктограмму . В появившемся окне (рисунок 1.14, а) в закладке «Стандартные функциональные блоки» раскрыть библиотеку «Стандартные функциональные блоки». В открывшемся перечне раскрыть закладку «Bistable Function Blocks», в которой выбрать функциональный блок *SR*, после чего нажать «ОК». В сегменте установится блок *SR* -триггера (рисунок 1.14, б);

3) в поле «???» над блоком вписать название блока *SR1*;

4) во входной цепи *RESET* на месте «???» установить адрес *%IX0.1* Программа примет следующий вид (рисунок 1.14, в).

Для программирования контроллера ПЛК150 доступно четыре вида таймеров:

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

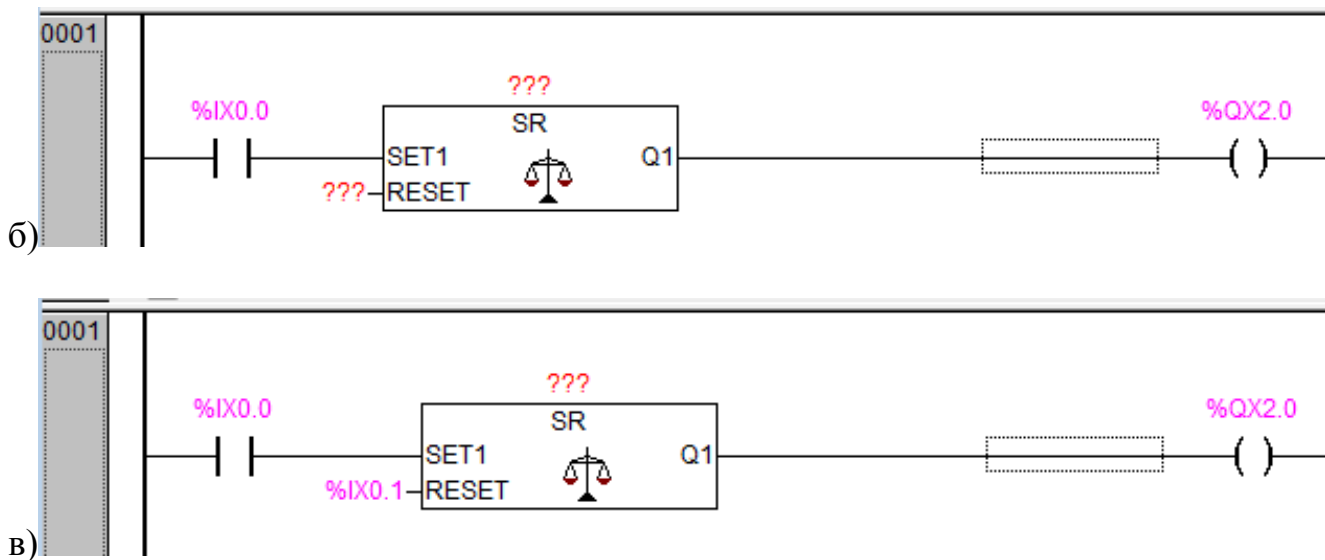
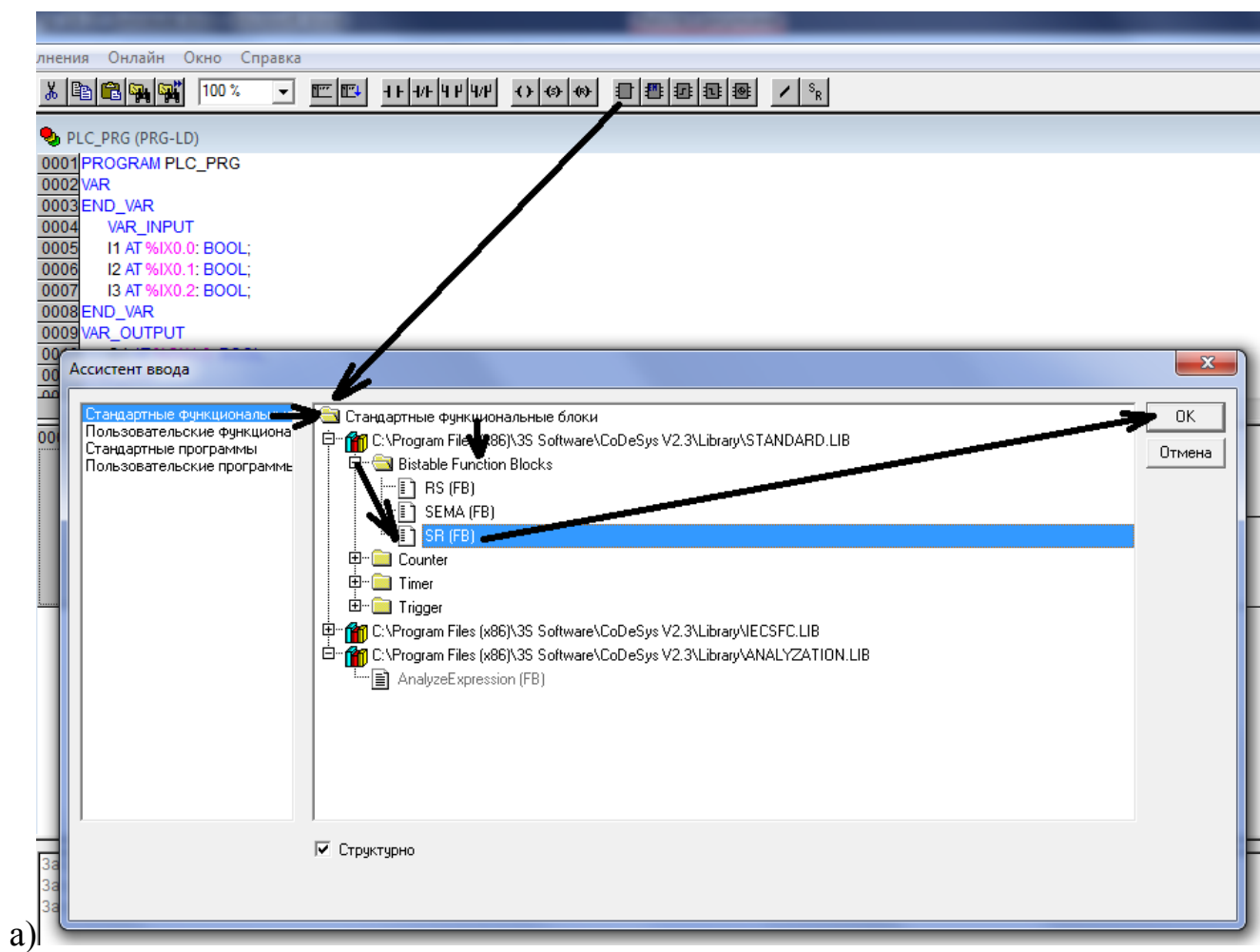


Рисунок 1.14 – Пример использования функциональных блоков

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*.

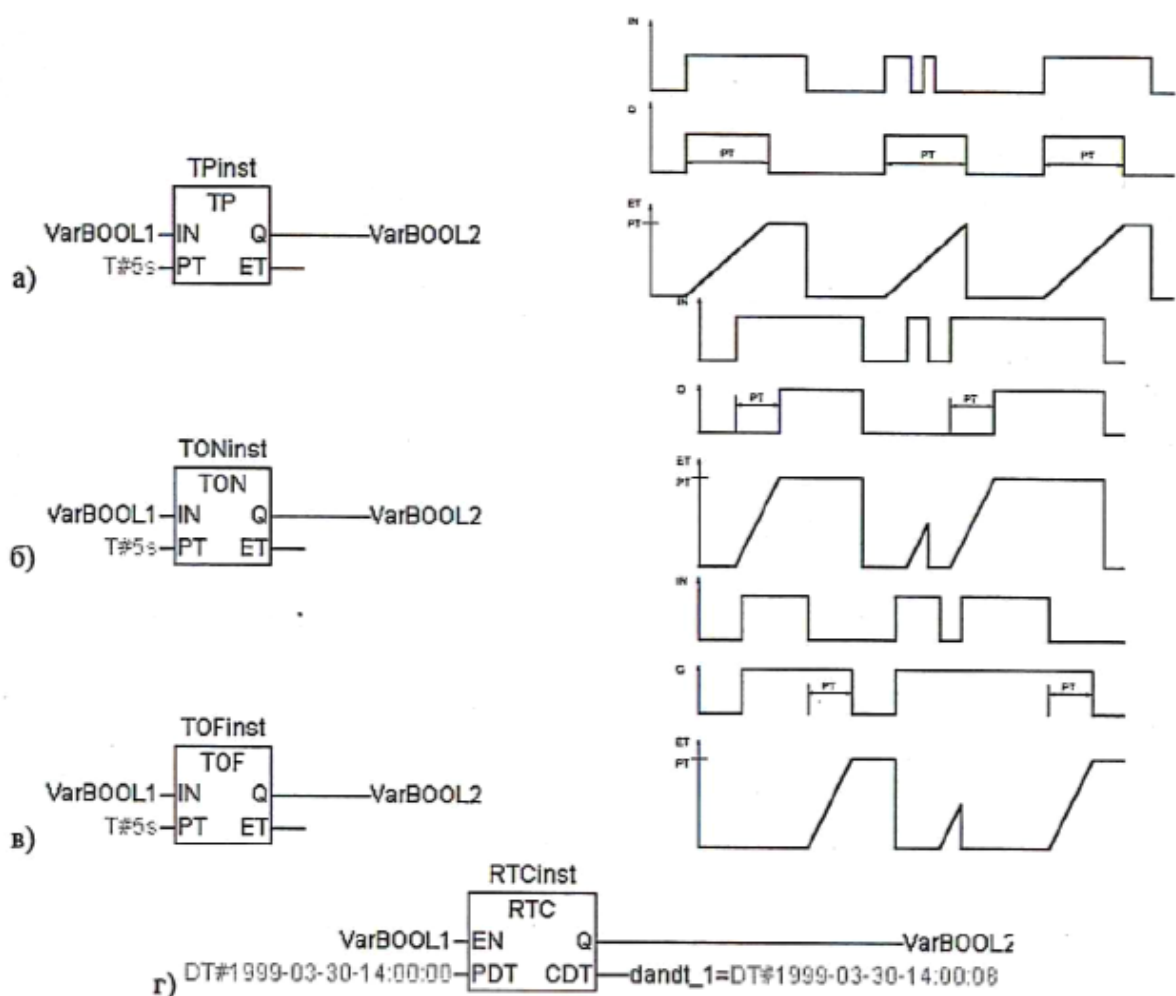


Рисунок 1.15 – Таймеры и их временные диаграммы работы

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>

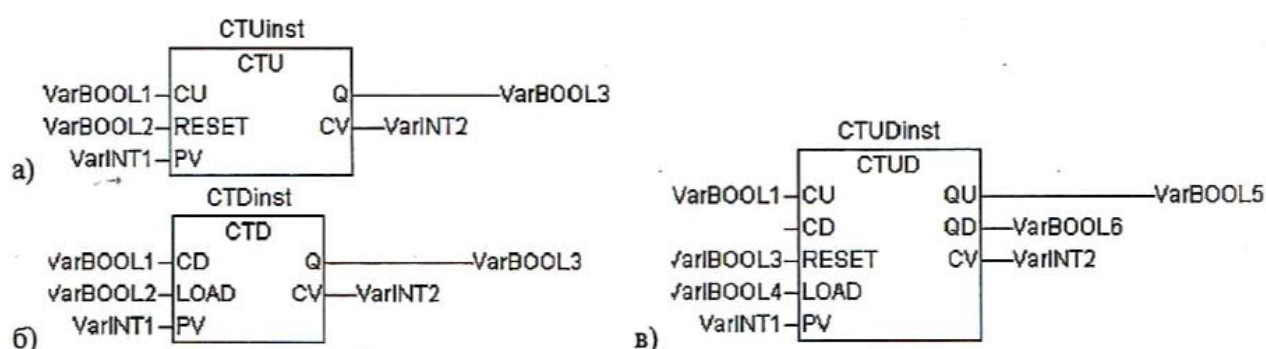


Рисунок 1.16 – Счетчики

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).

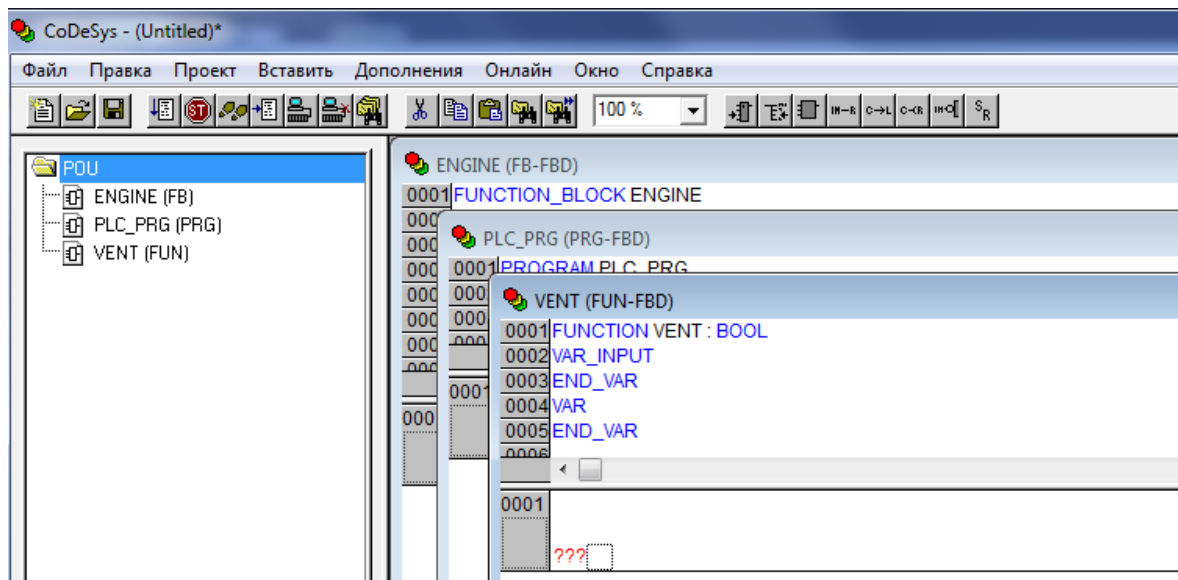


Рисунок 1.17 – Создание программных компонентов

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).

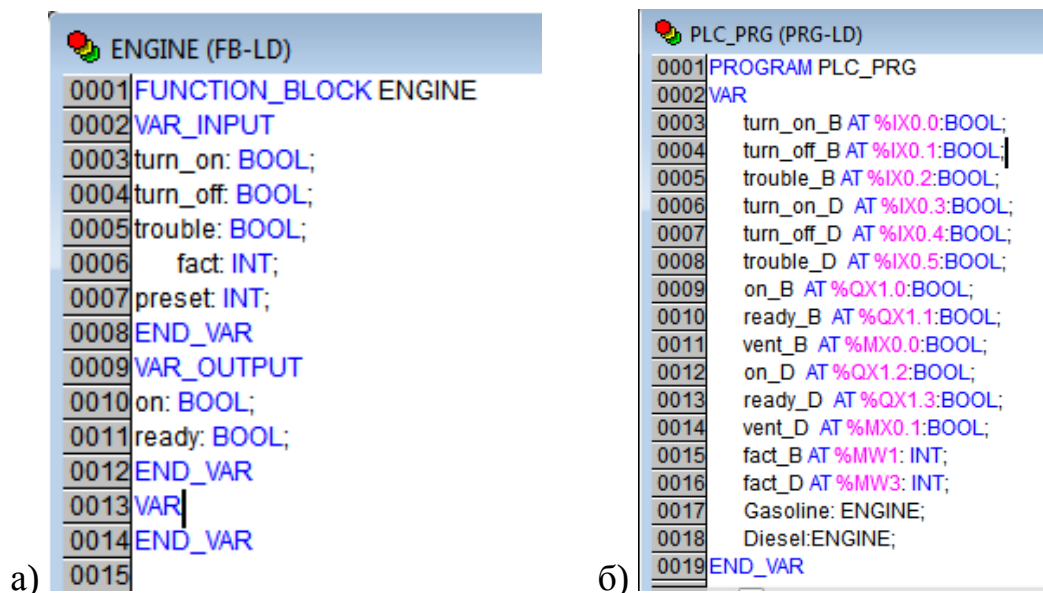


Рисунок 1.18 – Объявление переменных

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.

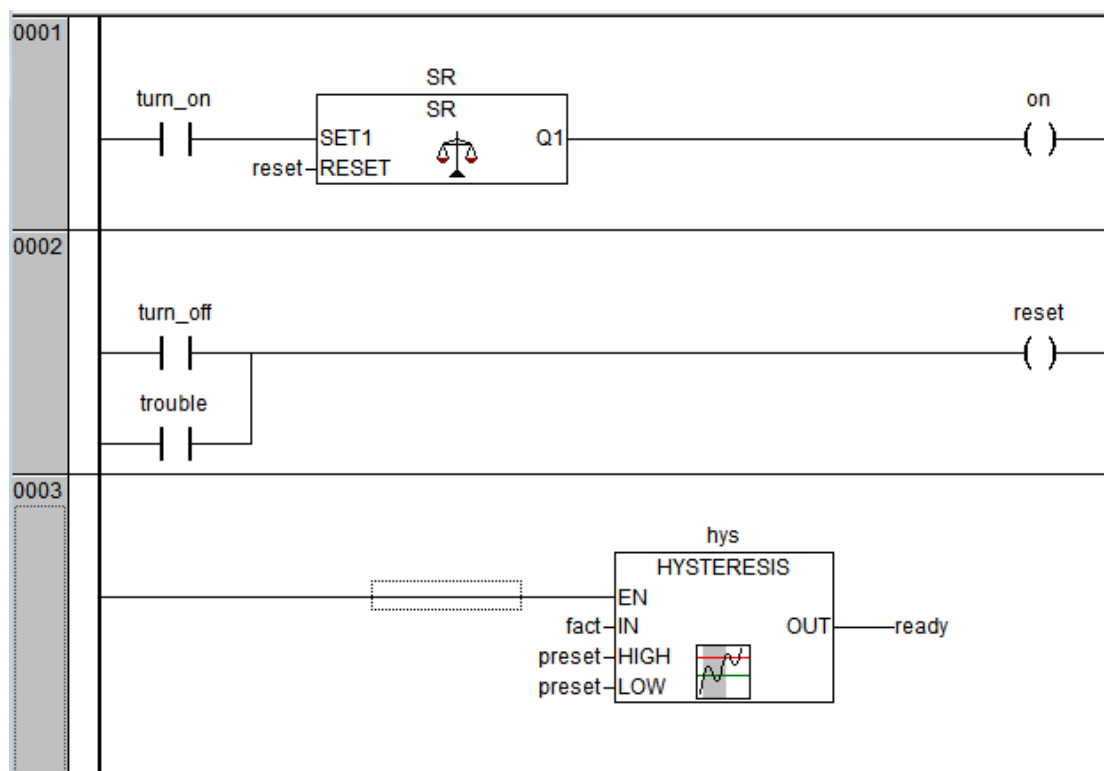


Рисунок 1.19 – Функциональный блок *ENGINE*

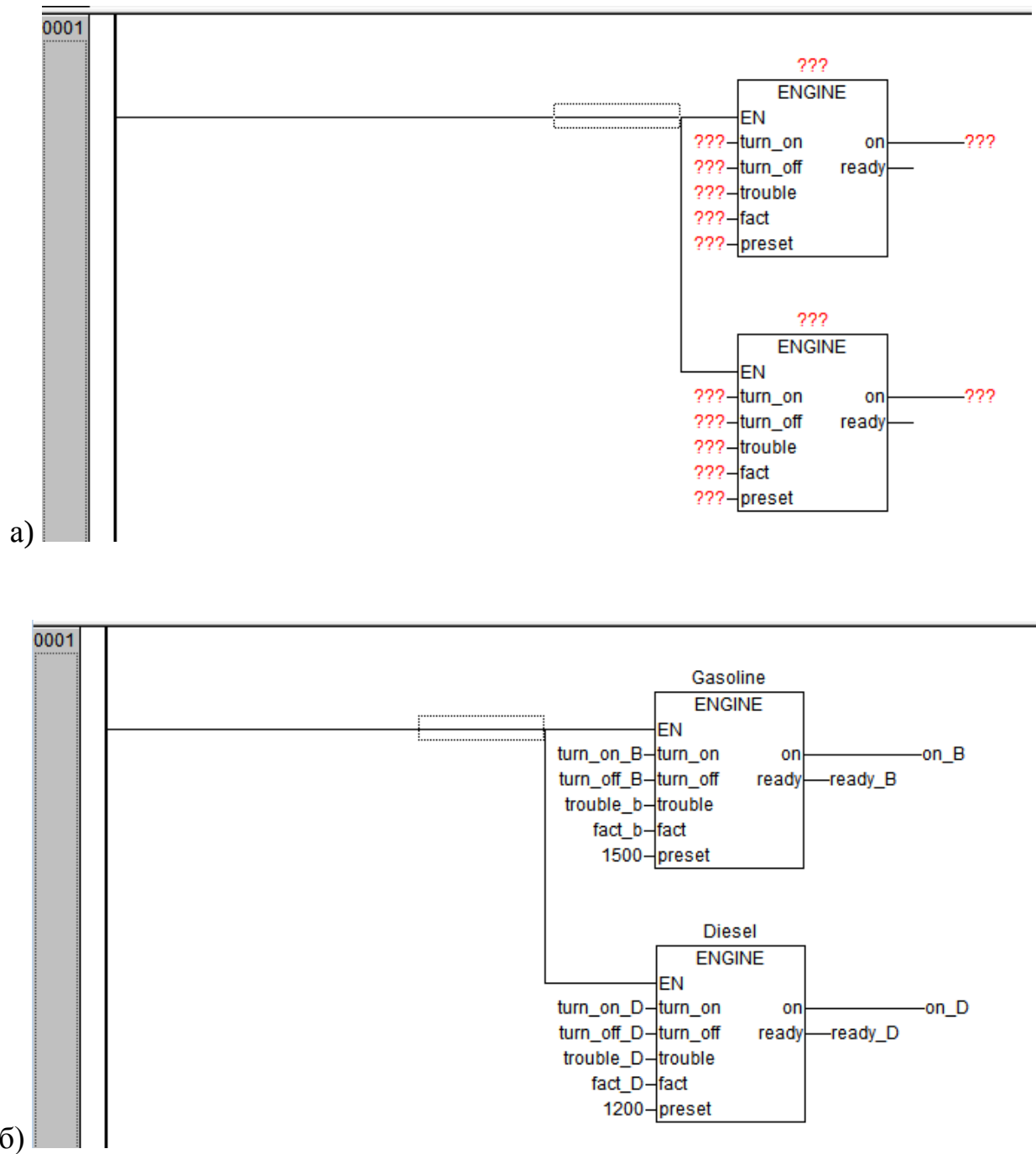


Рисунок 1.20 – Сегмент программы управления испытаниями двигателей

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.

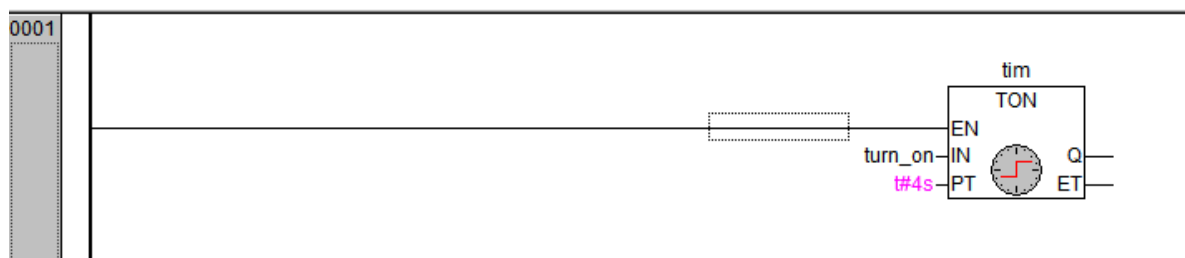


Рисунок 1.21 – Программирование функции *VENT*

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.

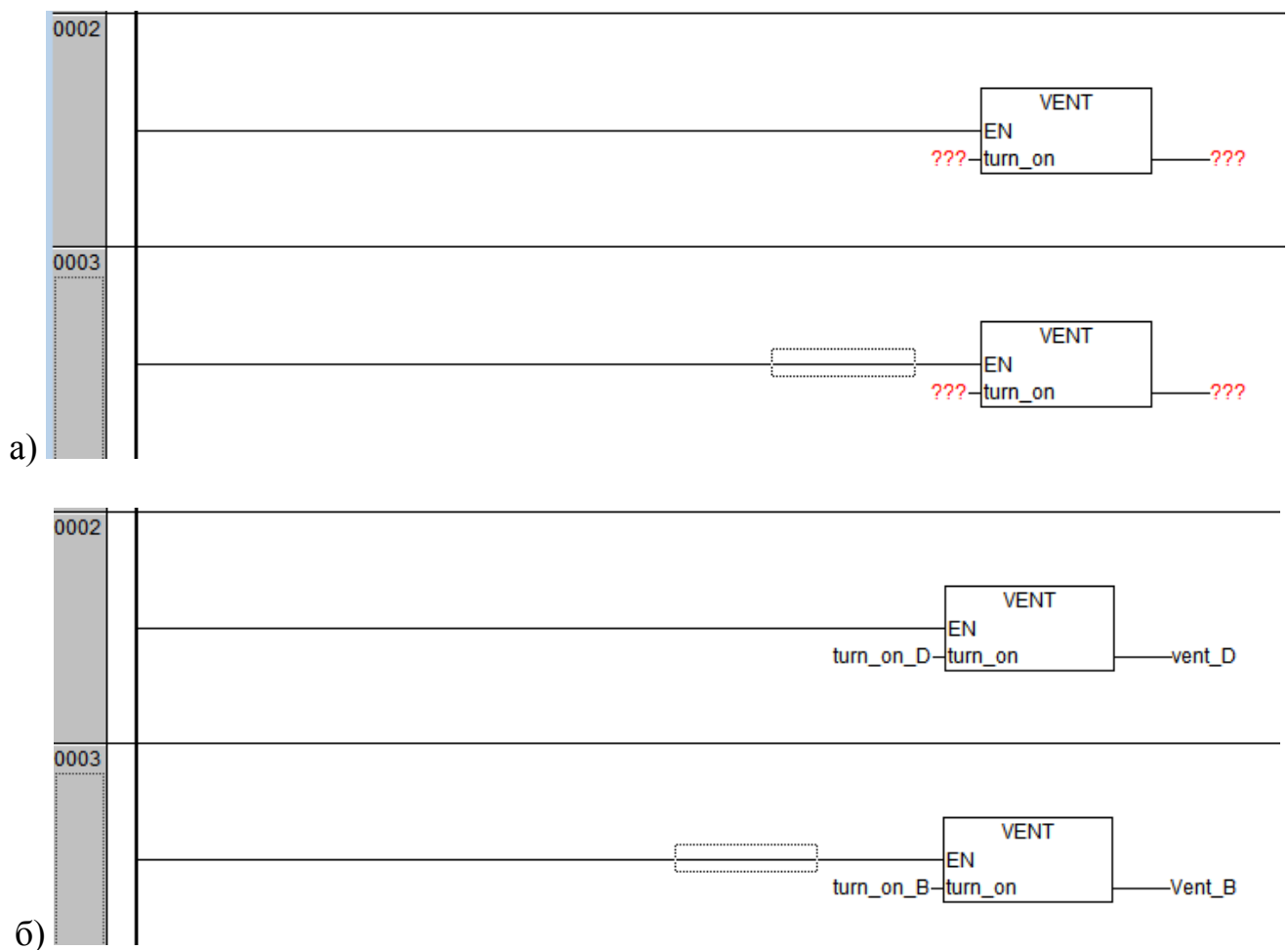


Рисунок 1.22 – Сегмент программы управления вентилятором

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

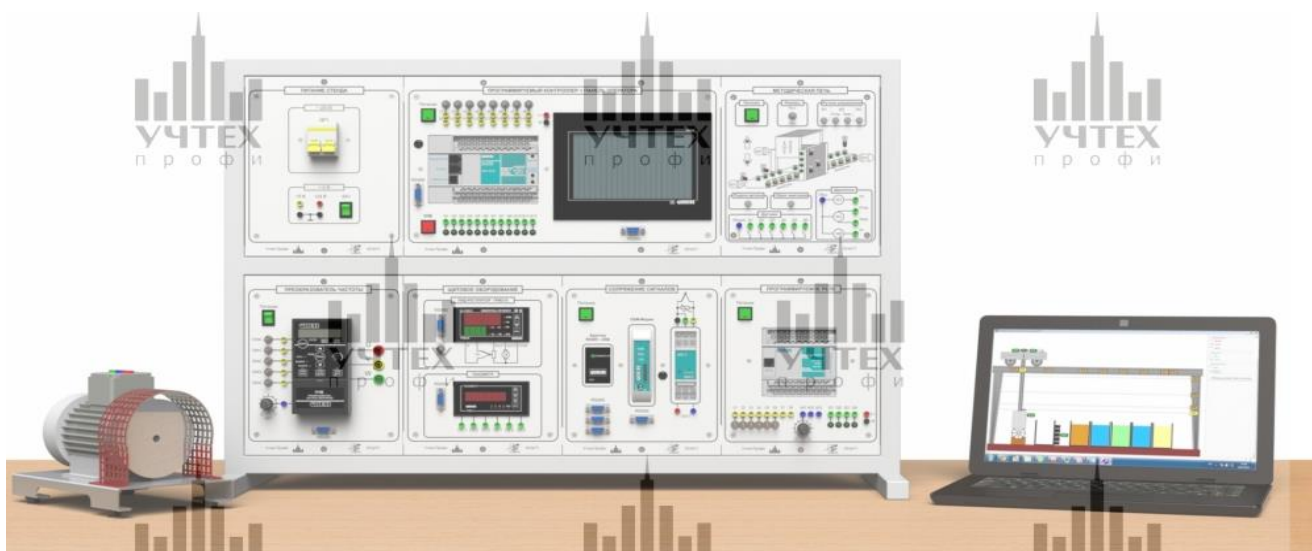


Рисунок 1.23 – Внешний вид стенда САУ-ОВЕН

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

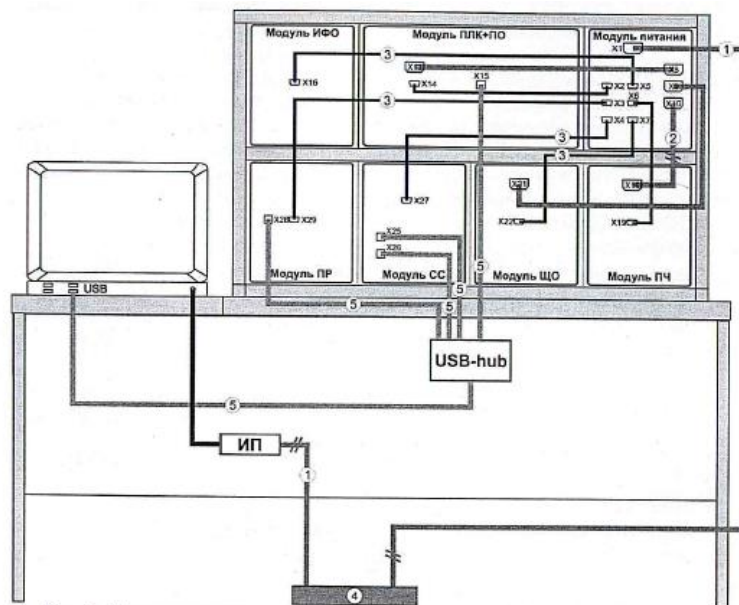


Рисунок 1.24 – Схема подключения лабораторного стенда (вид с тыльной стороны)

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.

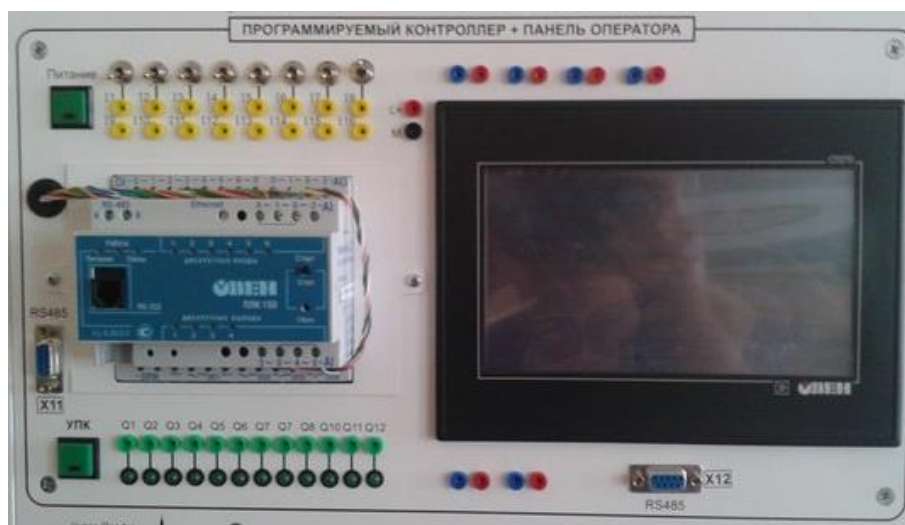


Рисунок 1.25 – Модуль программируемого логического контроллера и панели оператора (модуль ПЛК+ПО)

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.

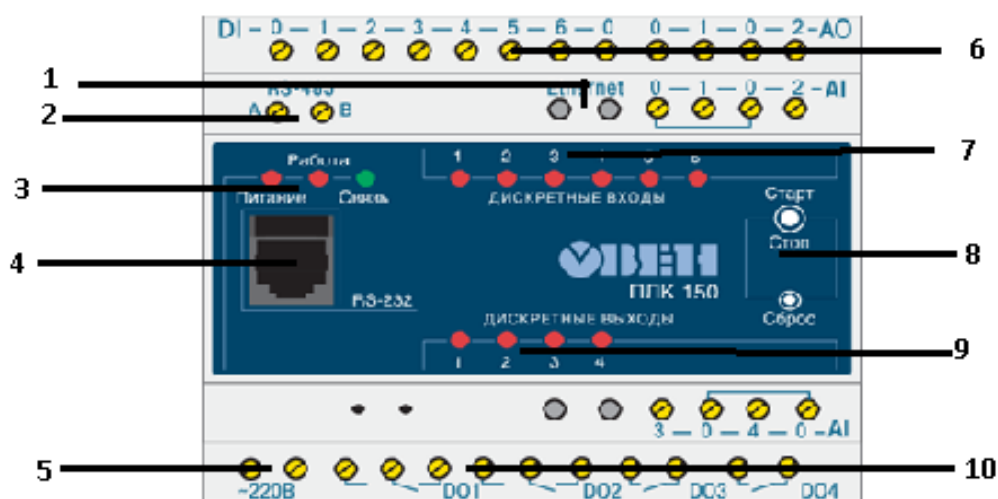


Рисунок 1.26 – Внешний вид ОВЕН ПЛК150

Описание основных элементов ПЛК представлено в таблице 1.8.

Таблица 1.8 – Описание основных элементов ПЛК

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

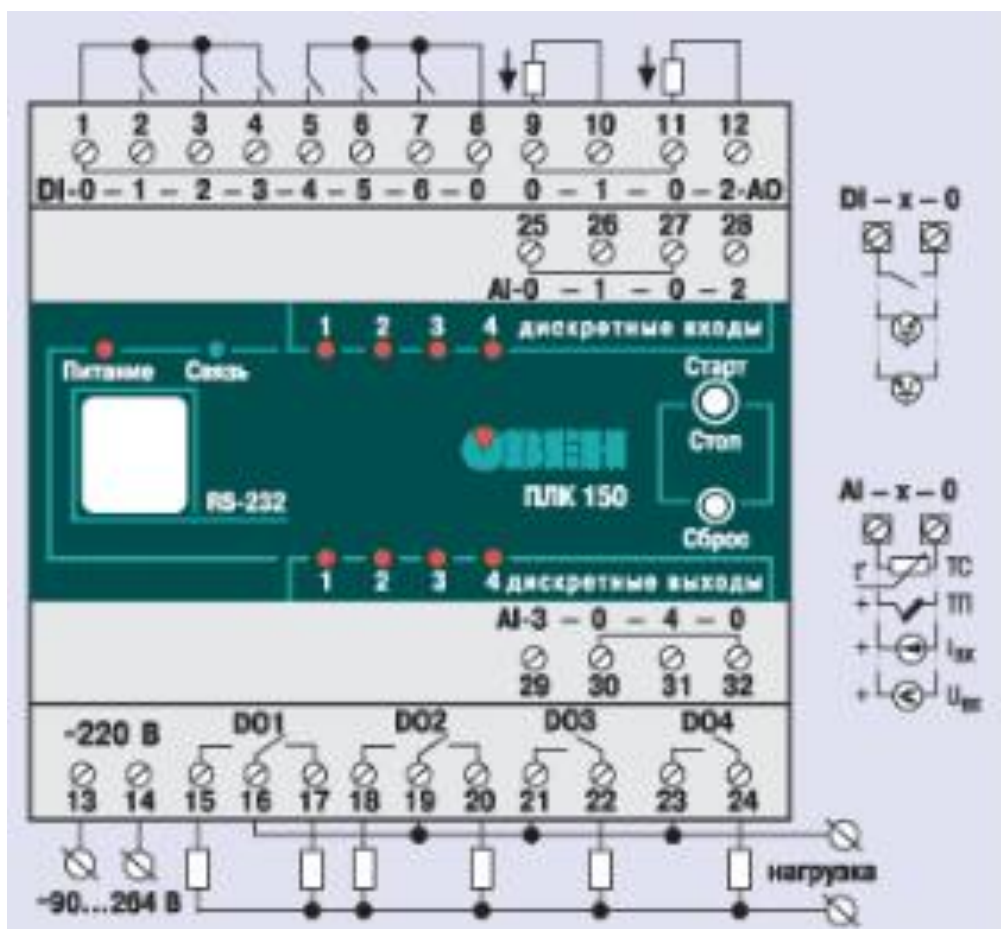


Рисунок 1.27 – Схема подключения питания, входов и выходов к ОВЕН ПЛК150

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 $Л0$ и $Л1$. Исходное положение тележки – $Л0$.

При кратковременном нажатии на кнопку «Пуск» тележка движется вперед до положения $Л1$, стоит в положении $Л1$ 5 с и возвращается назад в положение $Л0$. В положении $Л0$ немедленно возникает реверс привода тележки. Тележка вновь движется в положение $Л1$, стоит в нем 5 с и возвращается в $Л0$. В положении $Л0$ вновь возникает команда на движение до положения $Л1$ и т.д. после 10 циклов перемещения в положение $Л1$ при возвращении тележки в положение $Л0$ 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 $Л1^{\uparrow}$ и память P_L 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,

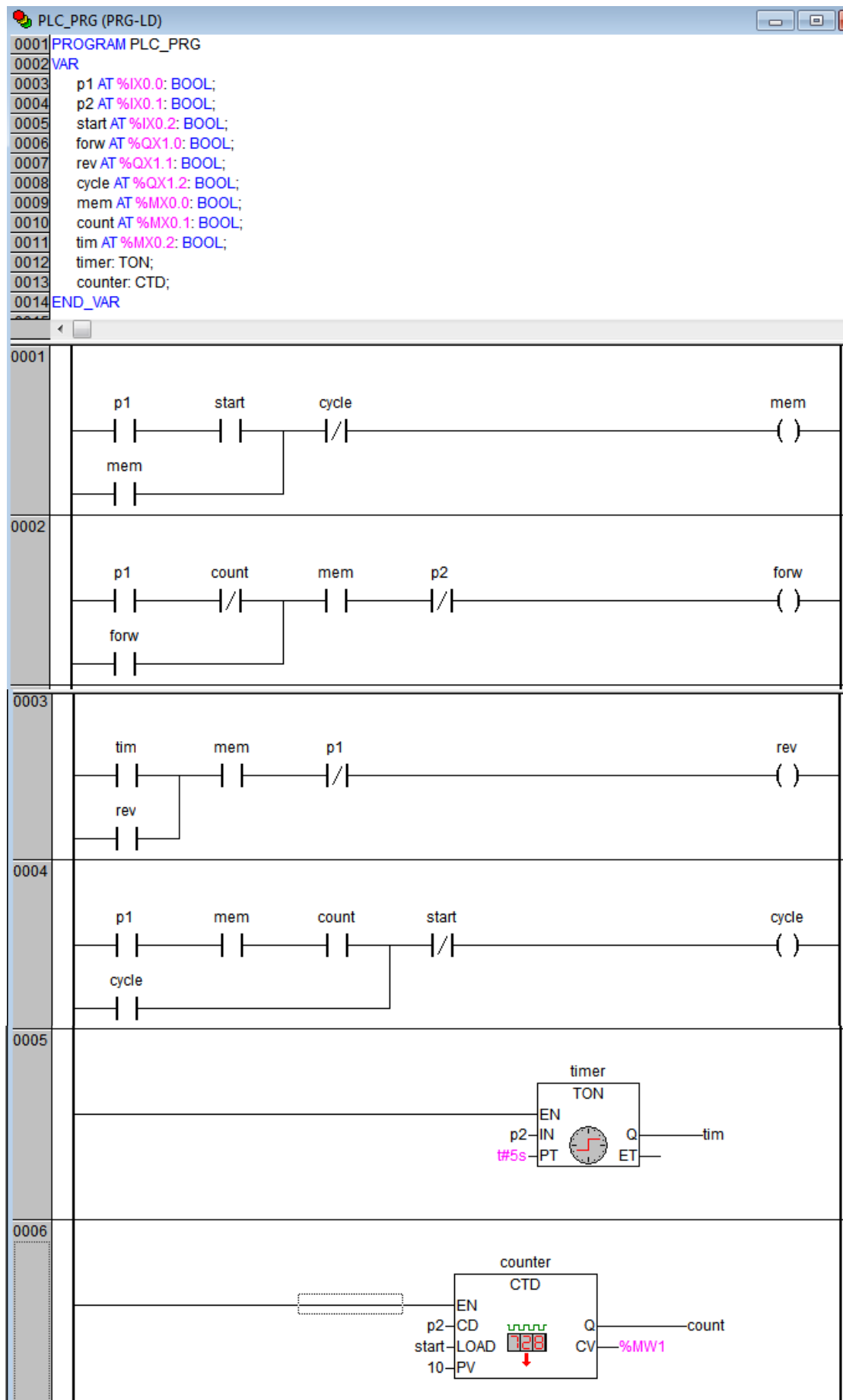


Рисунок 1.28 – Программа управления движением тележки

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, «owen_» 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).

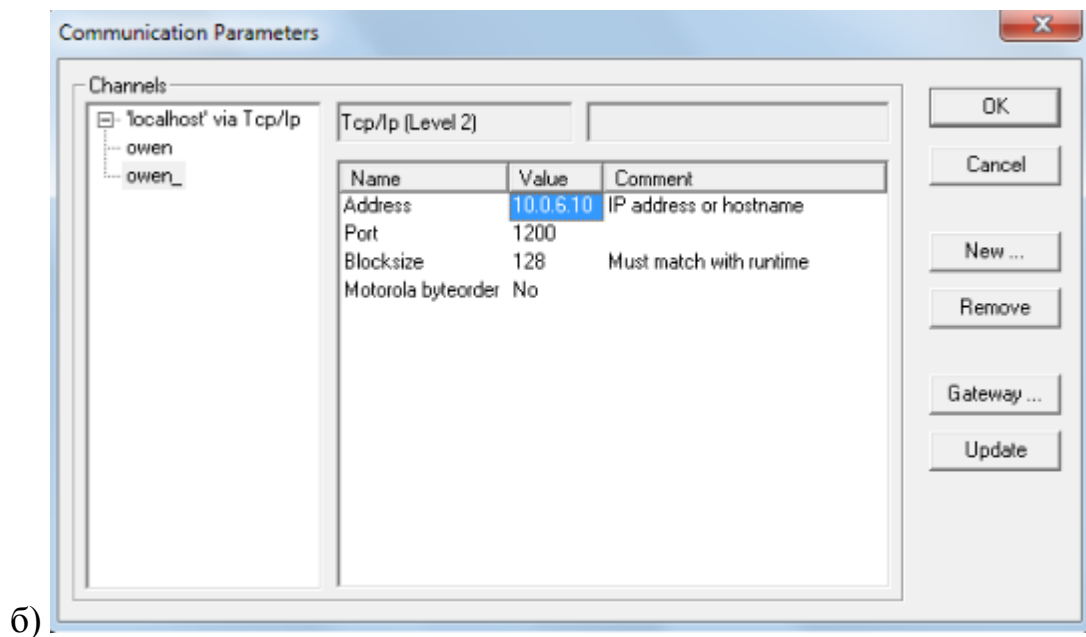
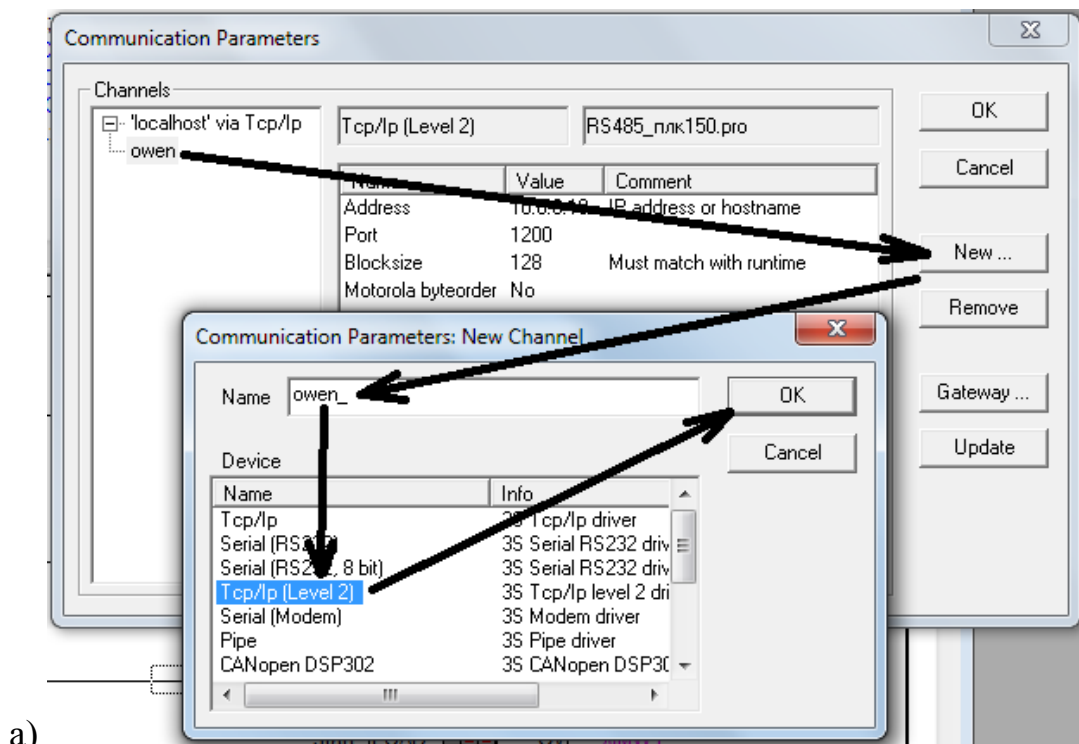


Рисунок 1.29 – Настройка параметров связи

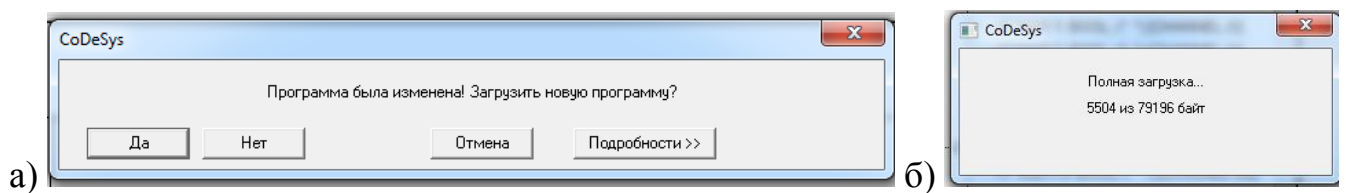


Рисунок 1.30 – Загрузка проекта в память ПЛК

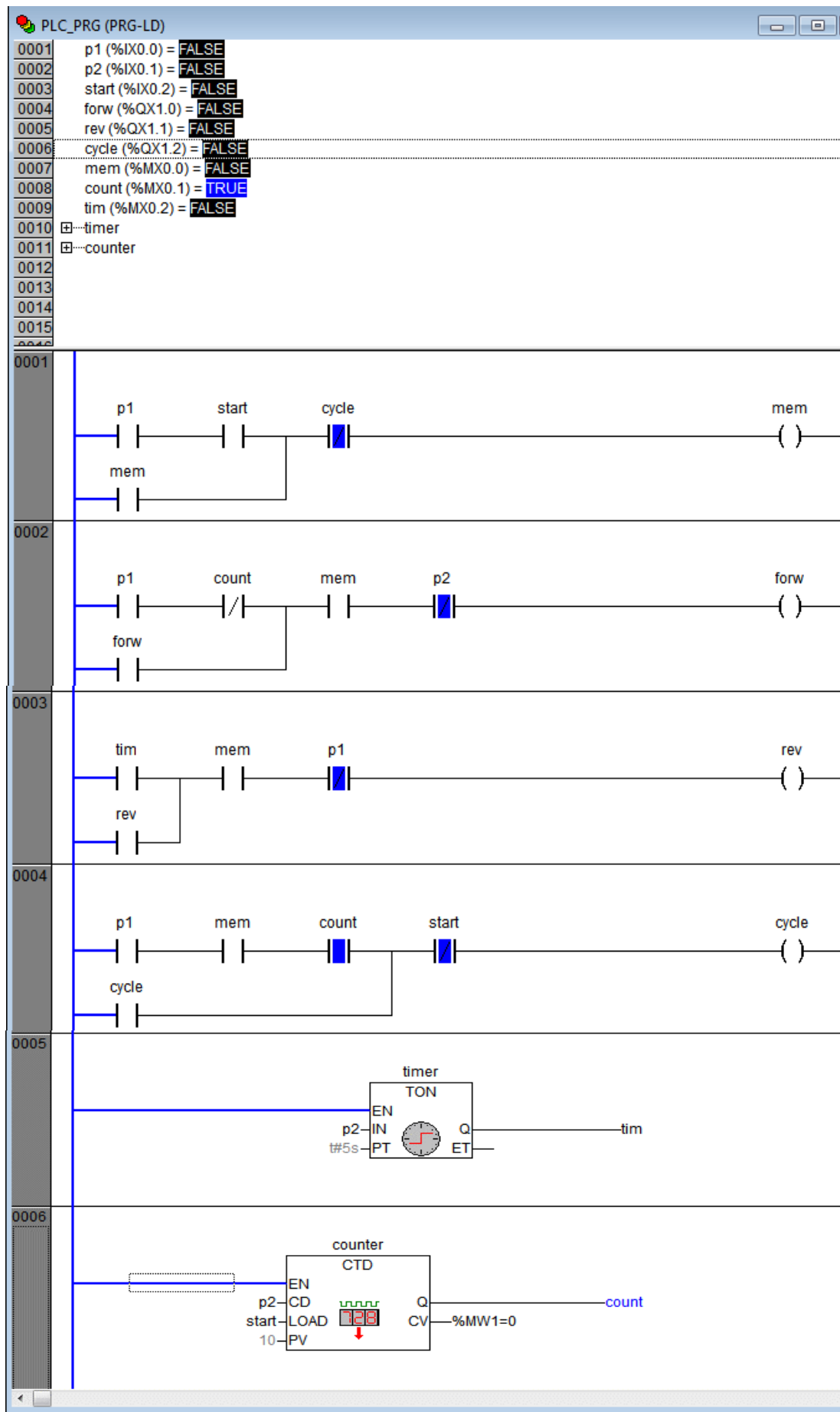


Рисунок 1.31 – Режим мониторинга

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

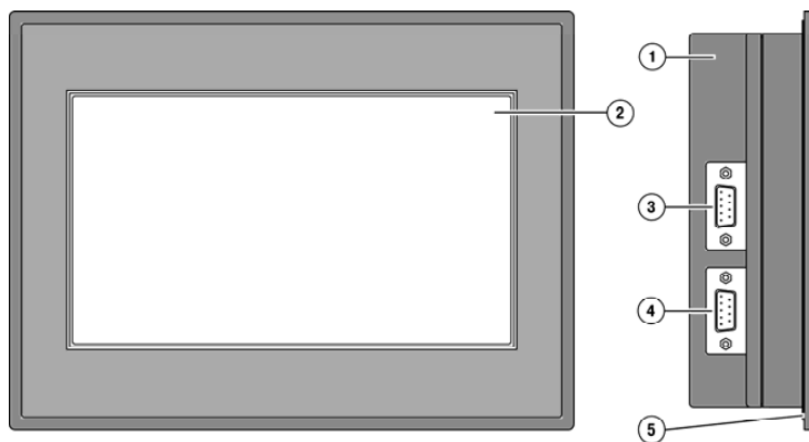


Рисунок 2.1 – Внешний вид панели оператора ОВЕН СП270

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.

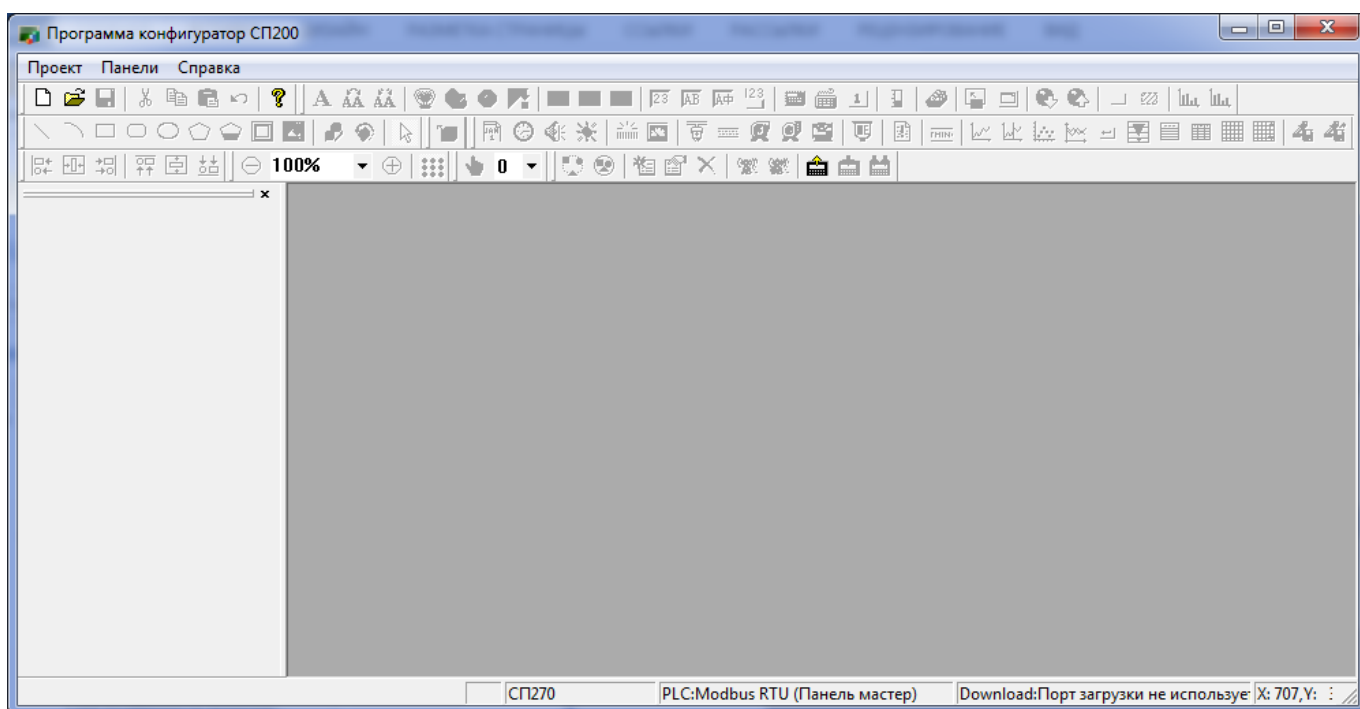


Рисунок 2.2 – Стартовый экран программы «Конфигуратор СП200»

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

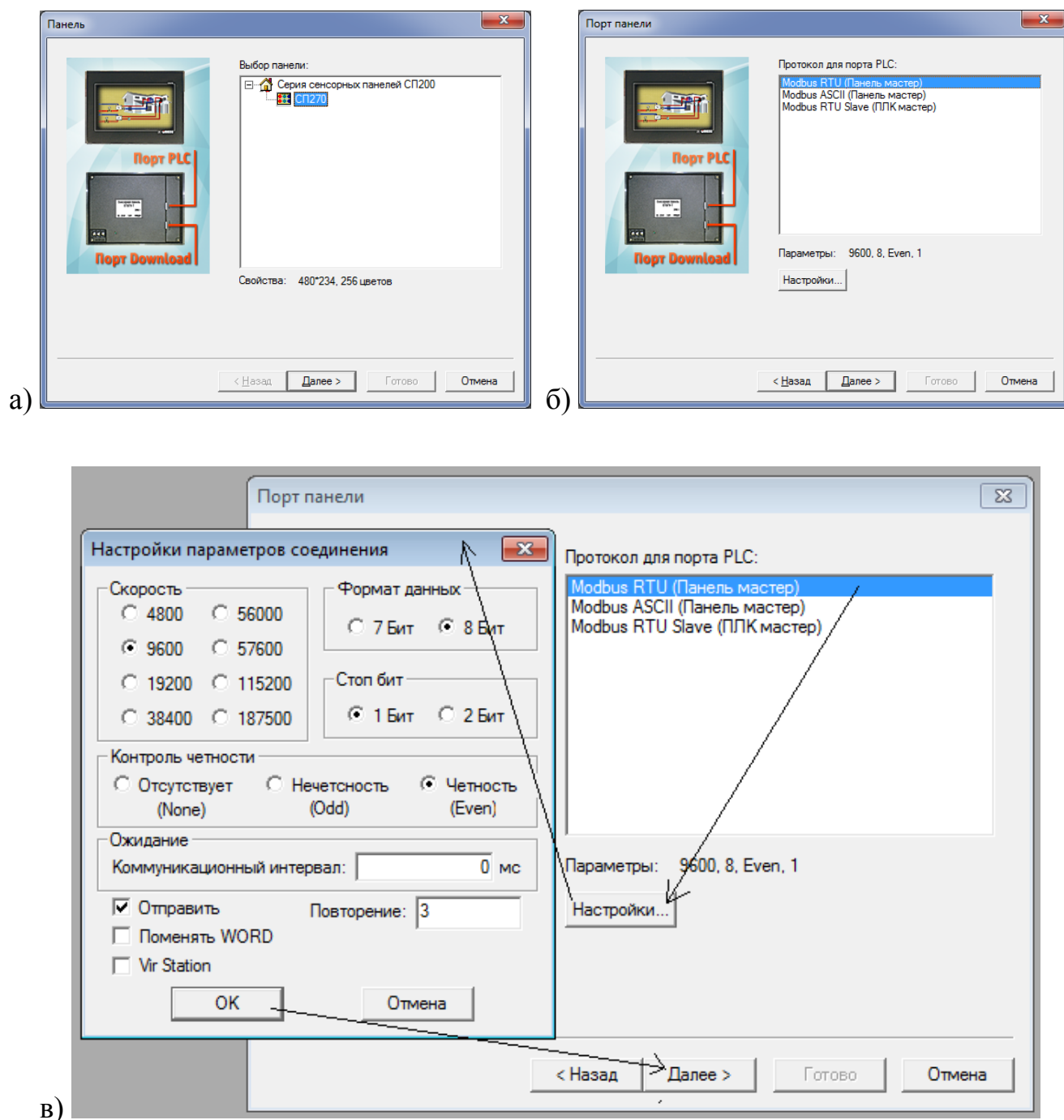


Рисунок 2.3 – Окна настройки проекта в программе «Конфигуратор СП200»

Шаг 3. В следующем окне можно выбрать протокол для порта ПЛК и режим работы «master/slave» (рисунок 2.3, б). После выбора протокола необходимо задать его настройки с помощью нажатия соответствующей кнопки в соответствии с рисунком 2.3, в).

Шаг 3. В следующем окне необходимо пропустить настройку, так как порт «Download» не используется. Для этого необходимо просто нажать «Далее».

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

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

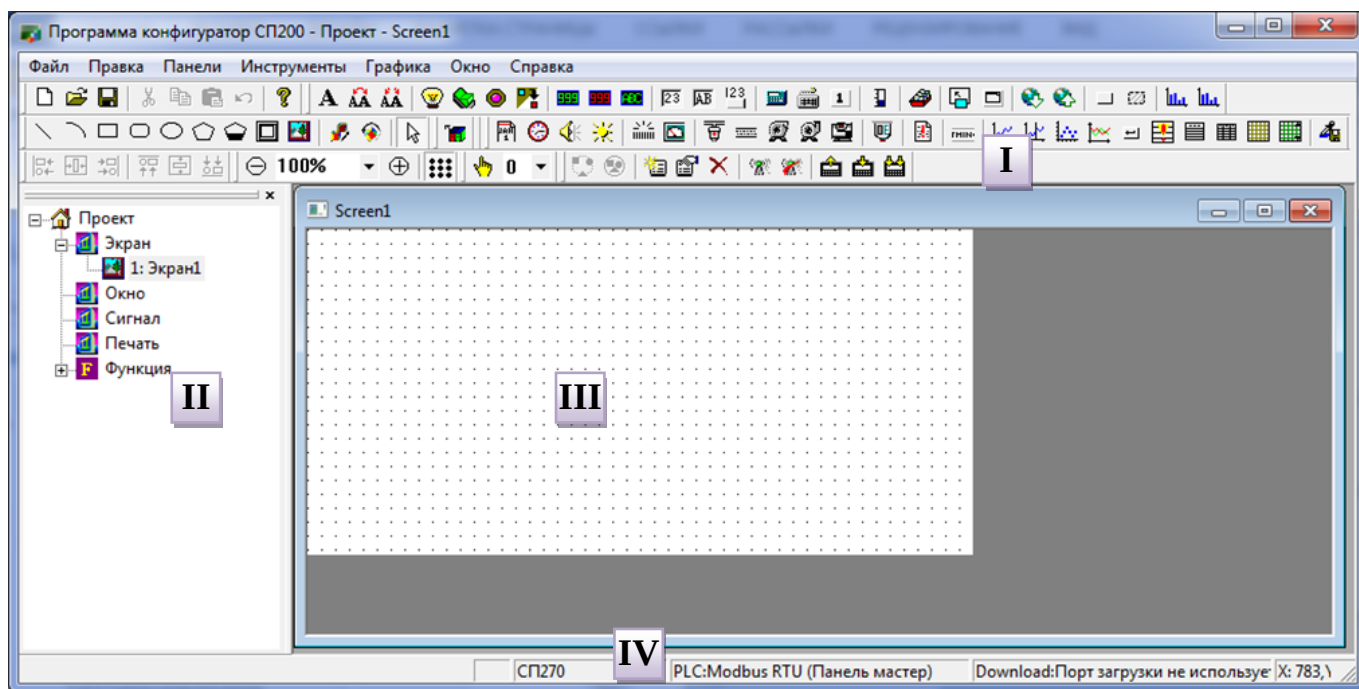


Рисунок 2.4 – Главное окно проекта

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.

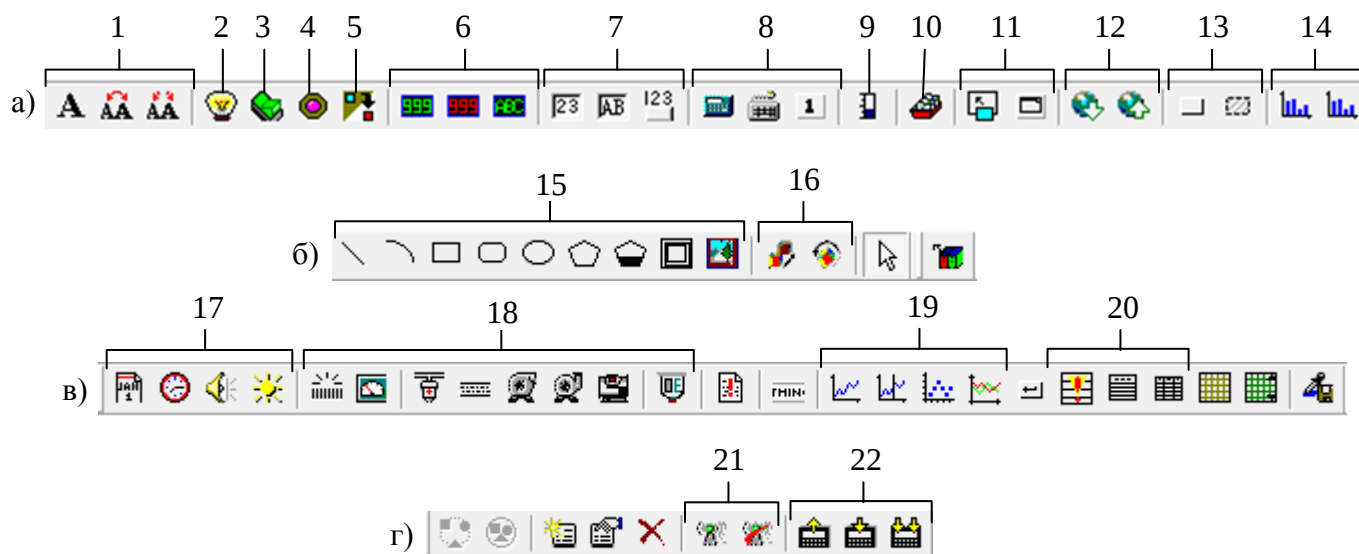


Рисунок 2.5 – Панели инструментов

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.

Переключатель с подсветкой

Управление | Действие | Вид | Цвет | Расположение

Управление объектом

Устройство

Порт: PLC

Доп.: 0 Адрес: 1

Регистр

Регистр: 0x Переадр. ☐

☐ Чтение

Устройство

Порт: PLC

Доп.: 0 Адрес: 1

Регистр

Регистр: 0x Переадр. ☐

OK Отмена Применить

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.

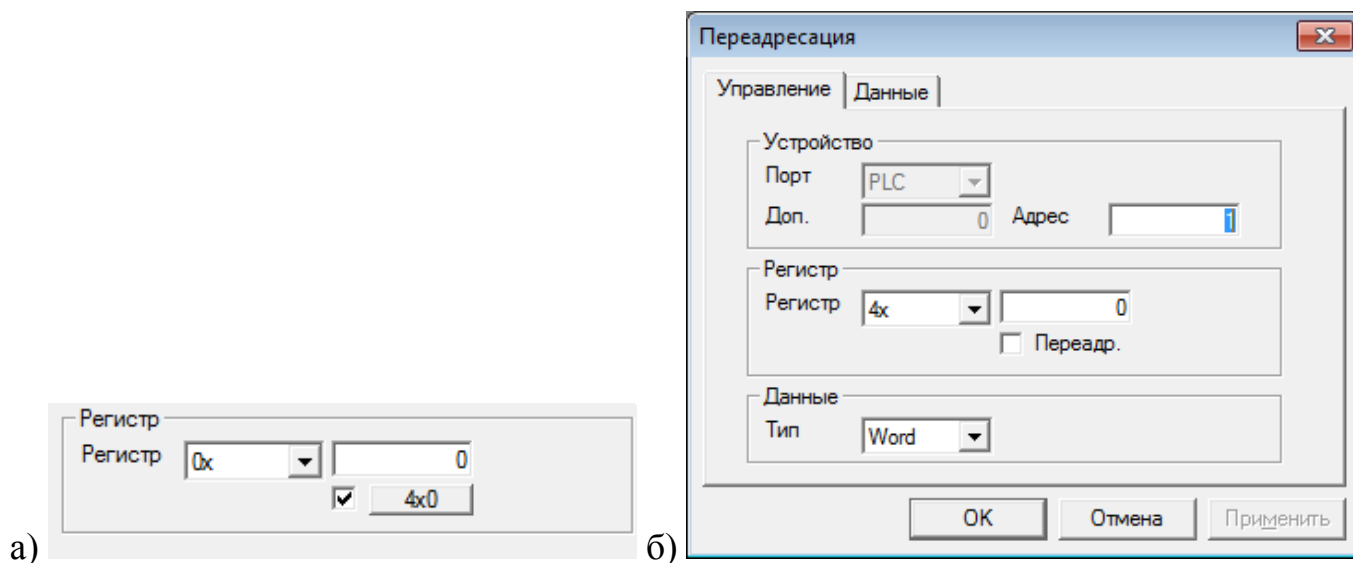


Рисунок 2.7 – Функция переадресации

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.).

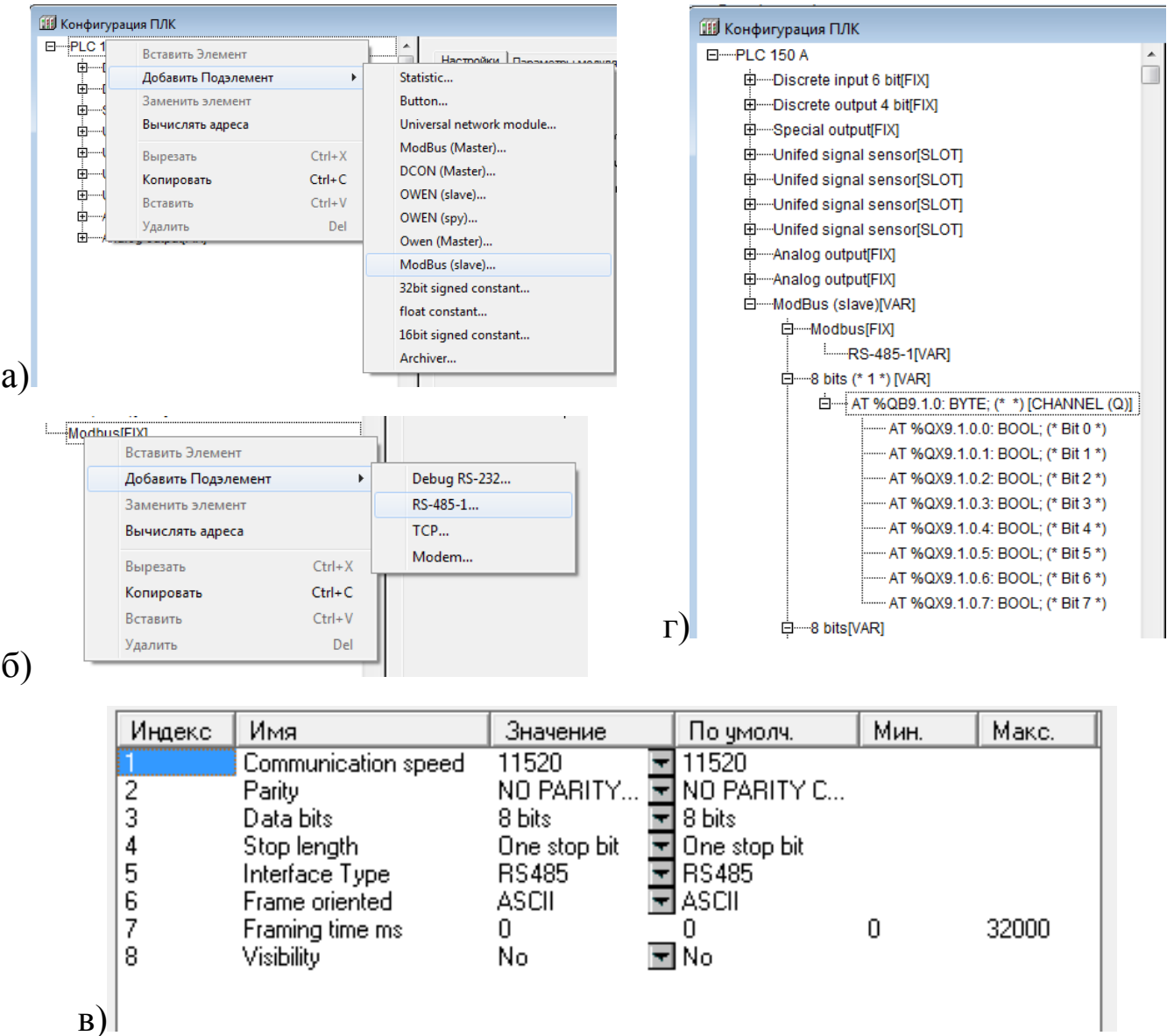


Рисунок 2.8 – Добавление *ModBus* подключения (*slave*) в ПЛК

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).

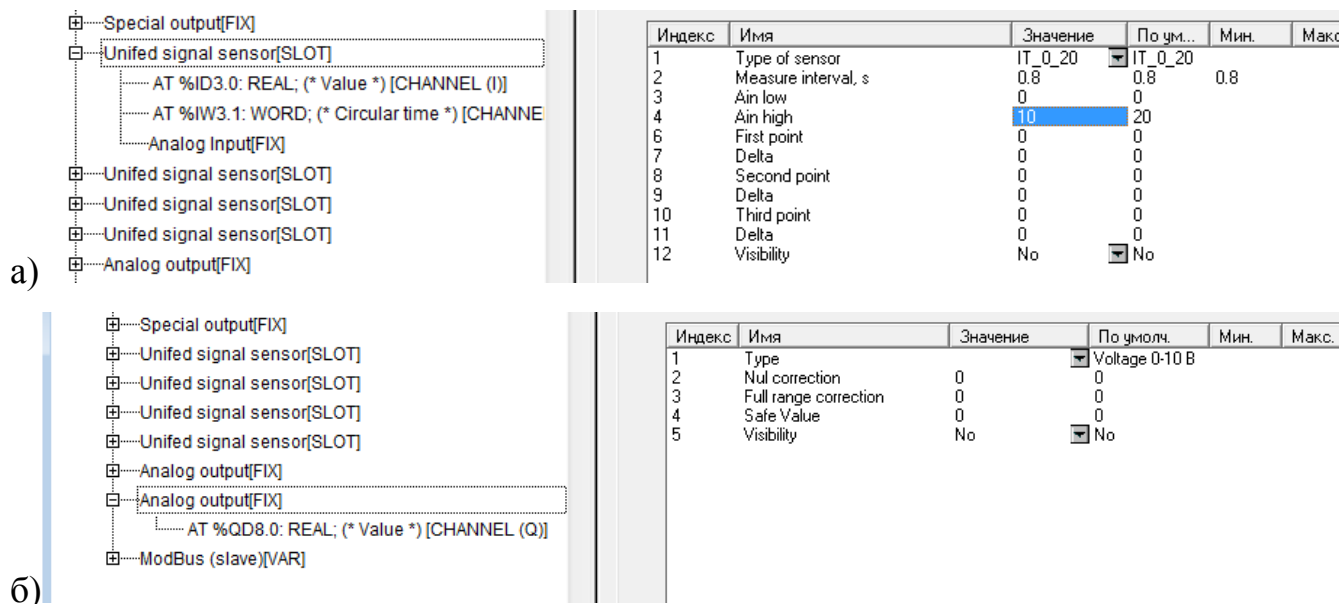


Рисунок 2.9 – Настройка аналоговых входов и выходов ПЛК 150

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)».

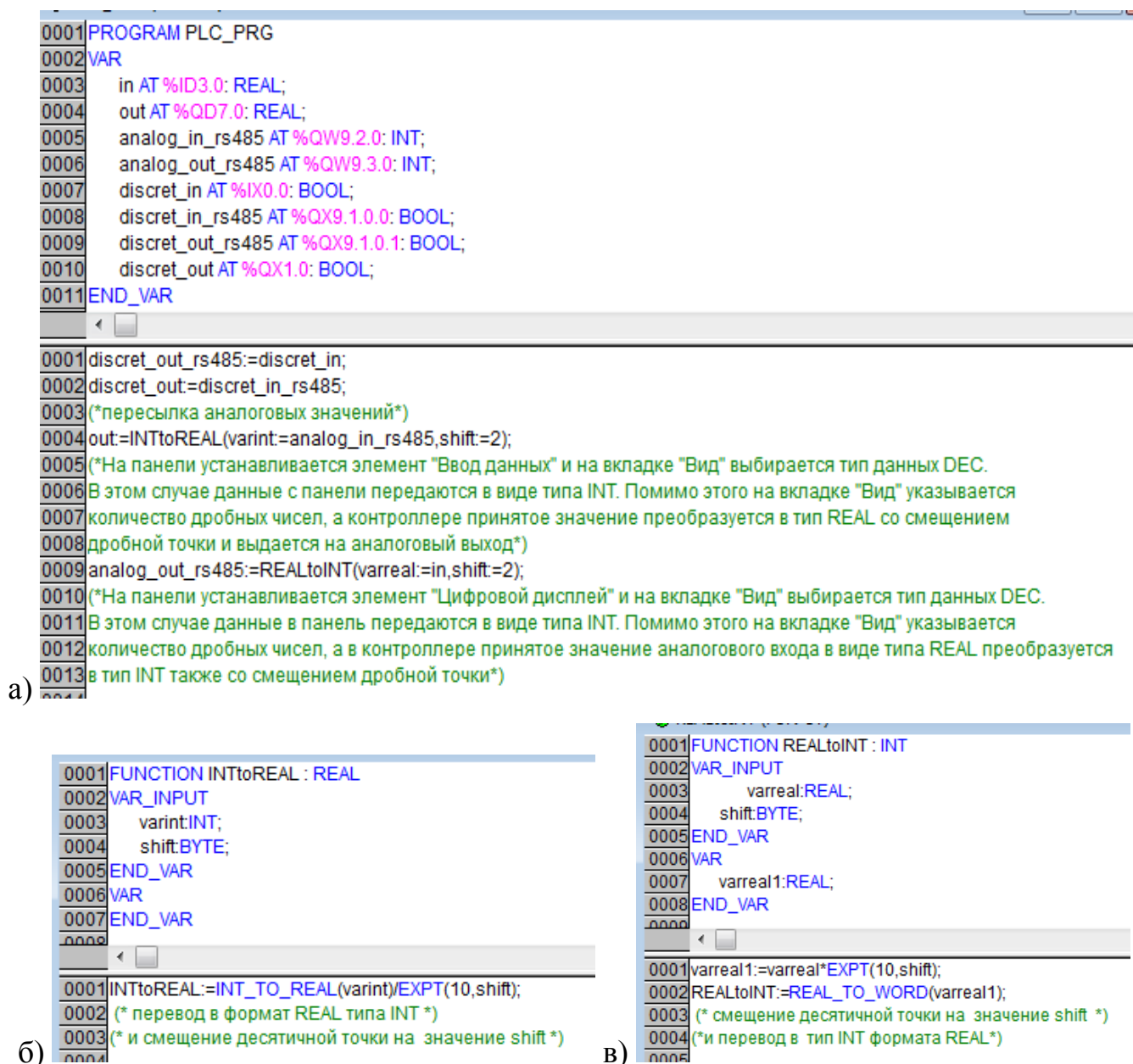


Рисунок 2.10 – Пример управляющей программы (а) и прикладных функций *INTtoREAL* (б) и *REALtoINT* (в) для работы по сети RS485

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).

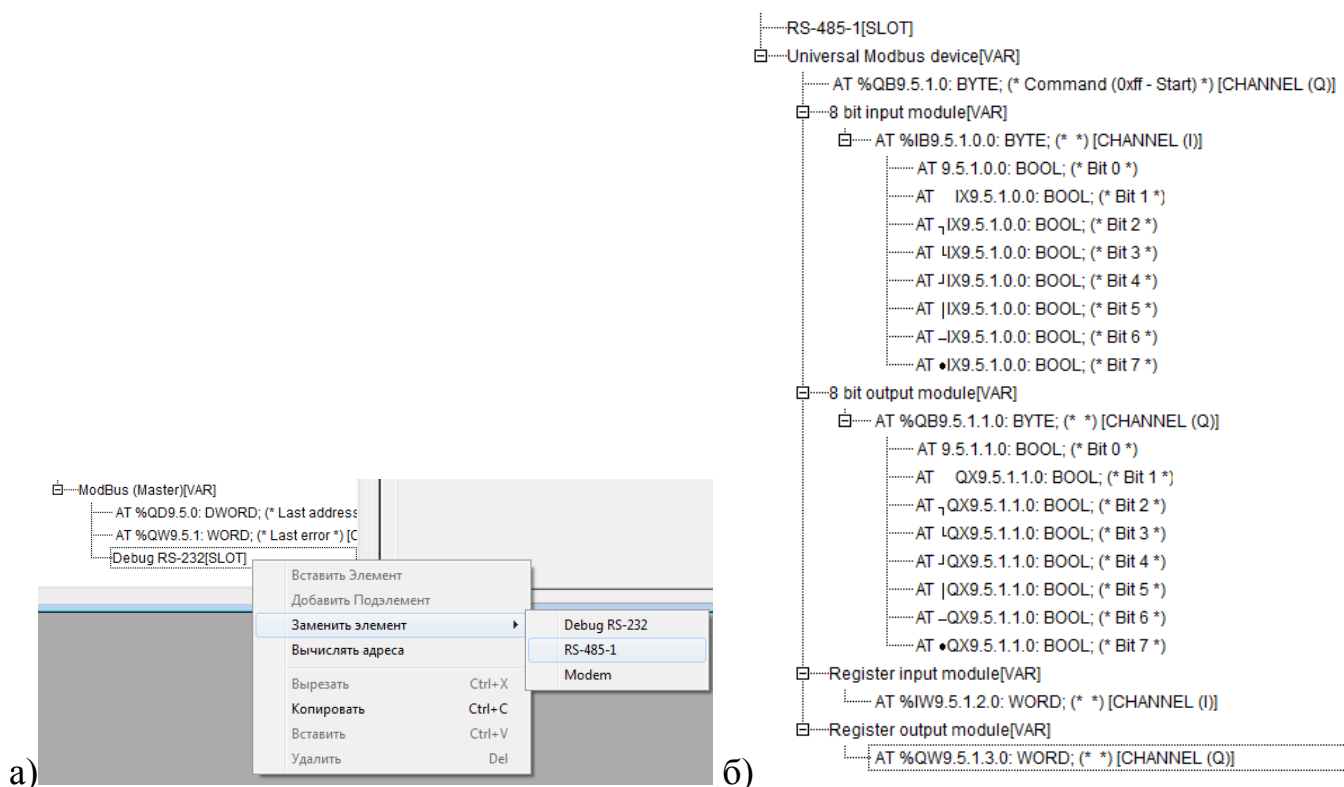


Рисунок 2.11 – Адреса сетевых переменных

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).

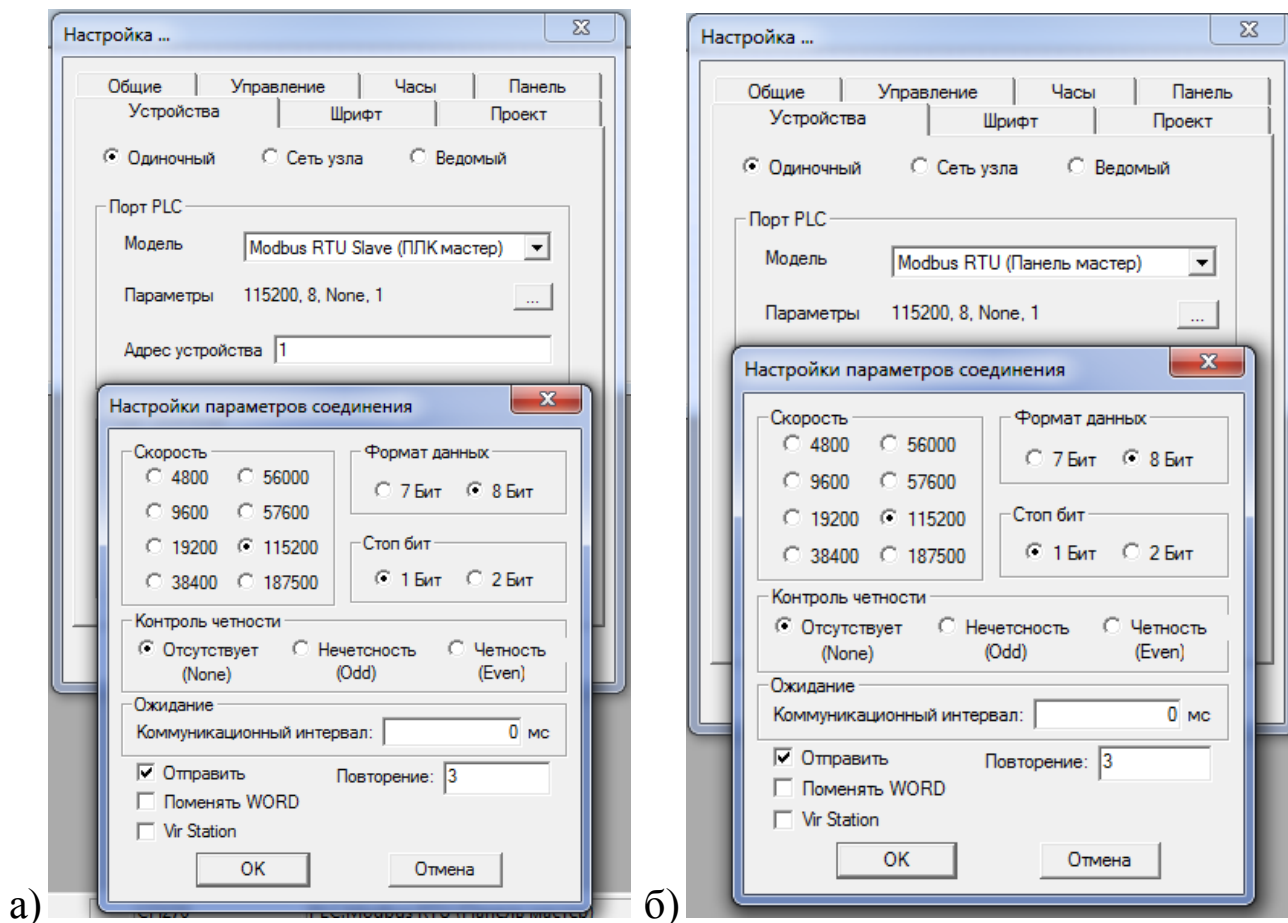


Рисунок 2.13 – Сетевые настройки панели оператора

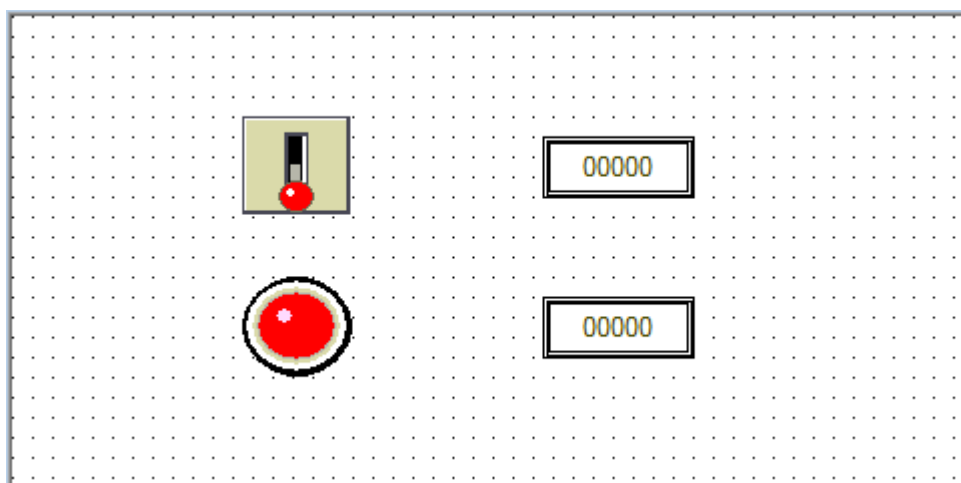


Рисунок 2.14 – Пример реализации экрана панели управления

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.

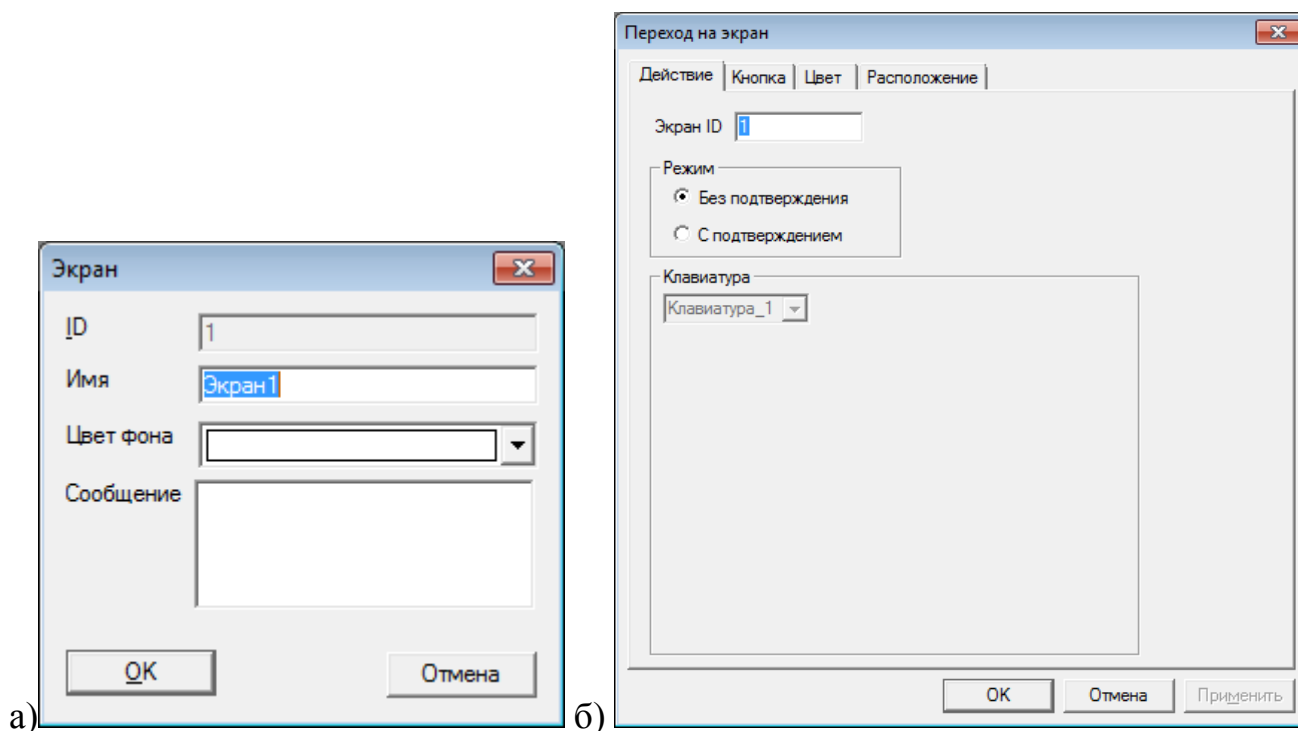


Рисунок 2.15 – Настройка свойств экрана (а) и кнопок переходов (б)

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.

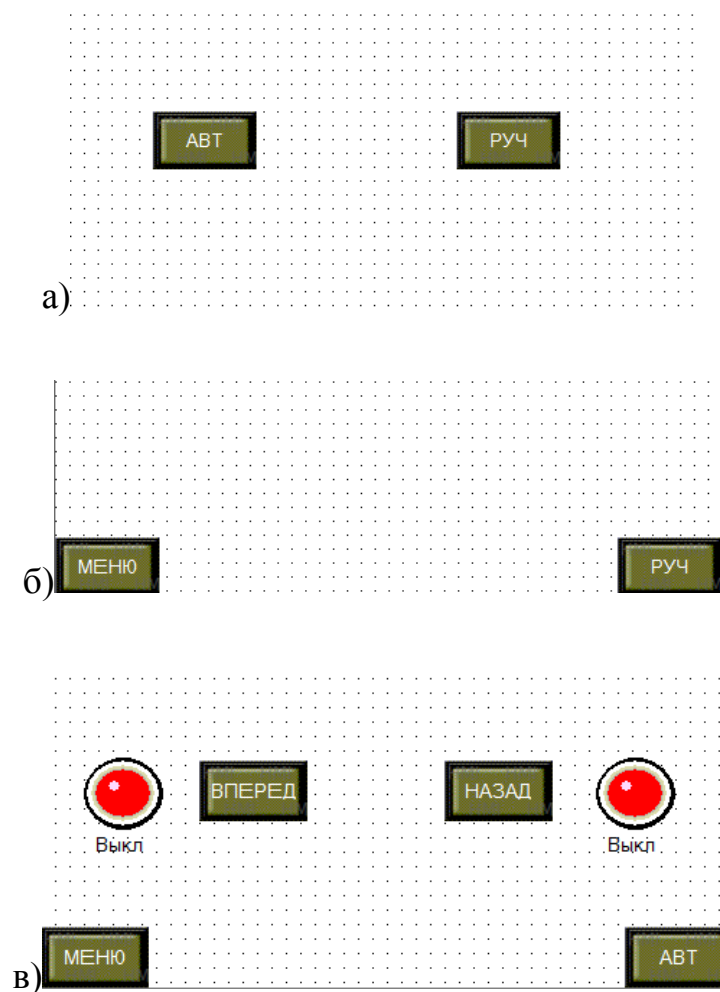


Рисунок 2.16 – Вид экранов панели оператора

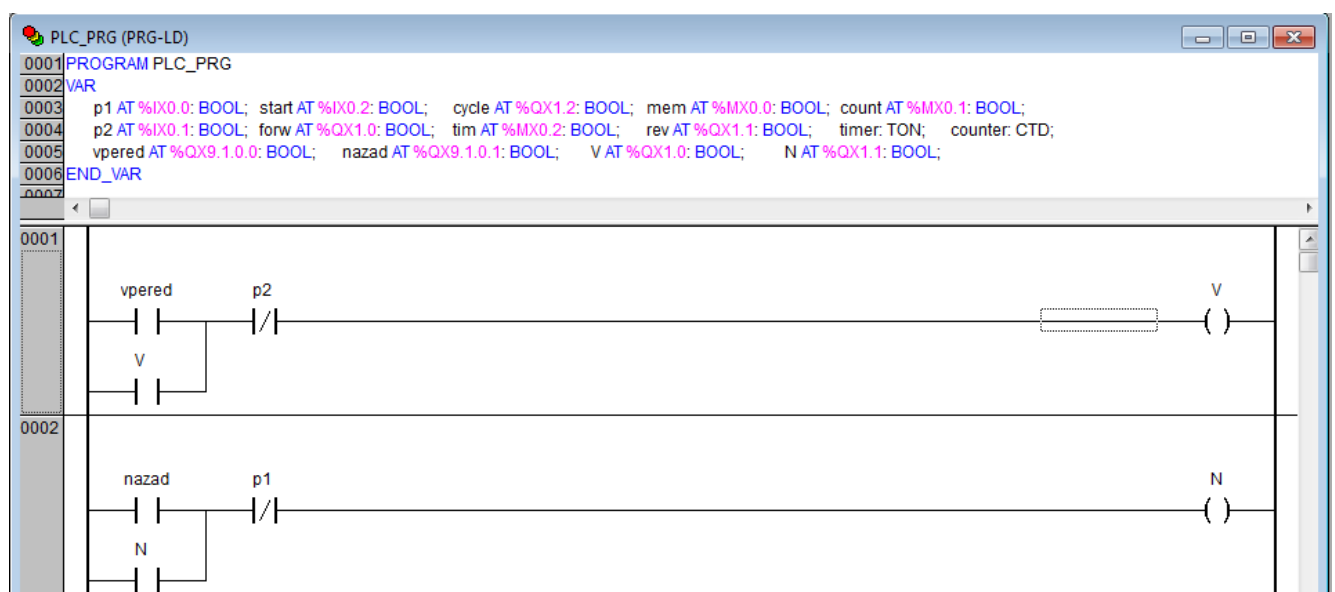


Рисунок 2.17 – Программа управления тележкой

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.

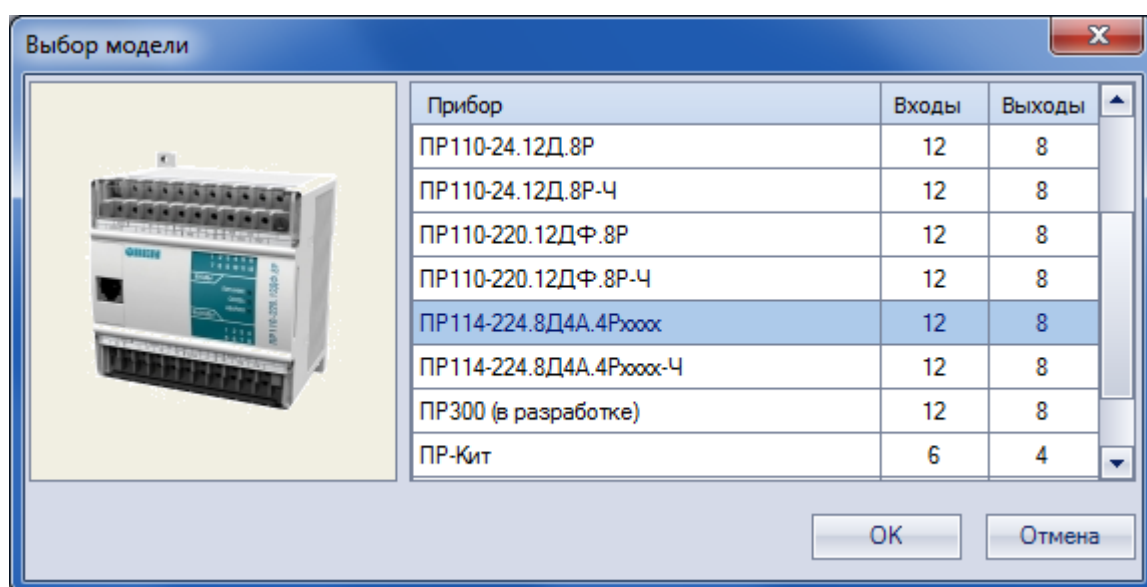


Рисунок 3.1 – Закладка диалогового окна выбора модели программируемого реле

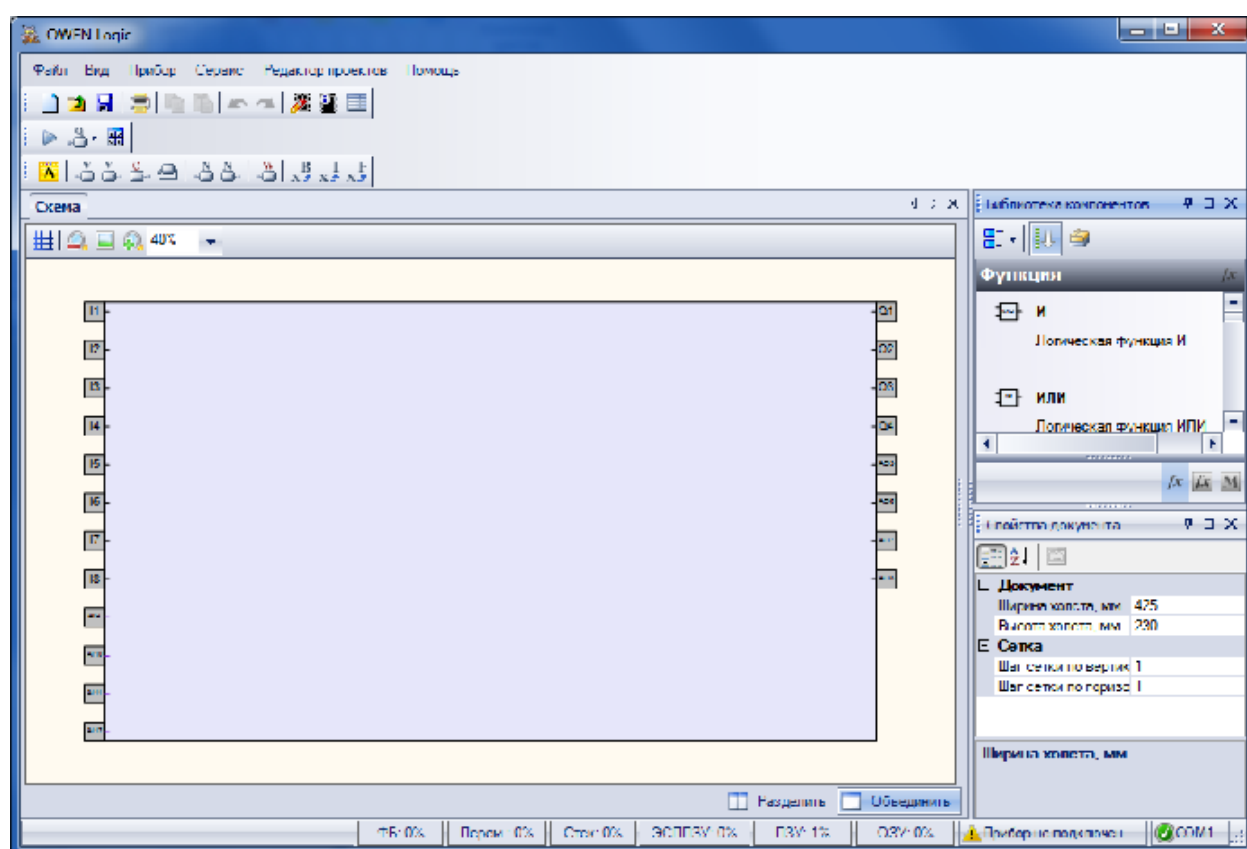


Рисунок 3.2 – Окно программы *OWEN Logic*

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*).

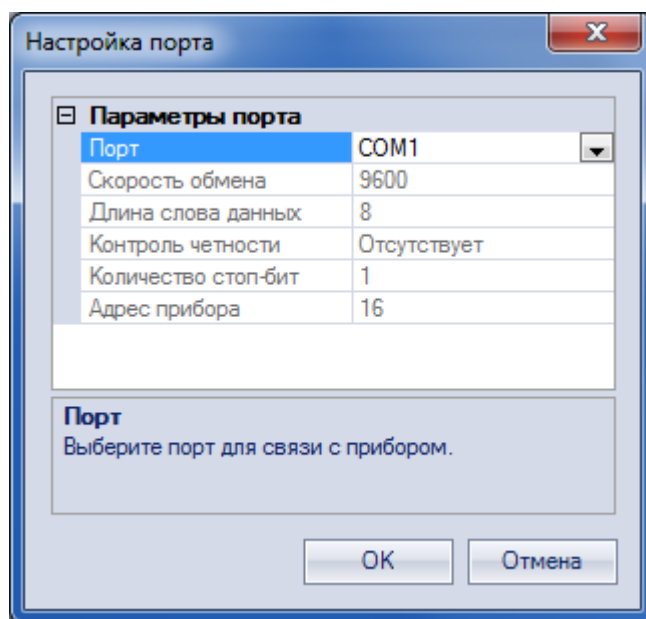


Рисунок 3.3 – Окно настройки параметров порта связи

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).



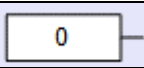
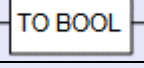
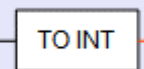
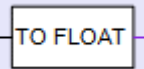
Рисунок 3.4 – Панель инструментов с вспомогательными функциями

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.

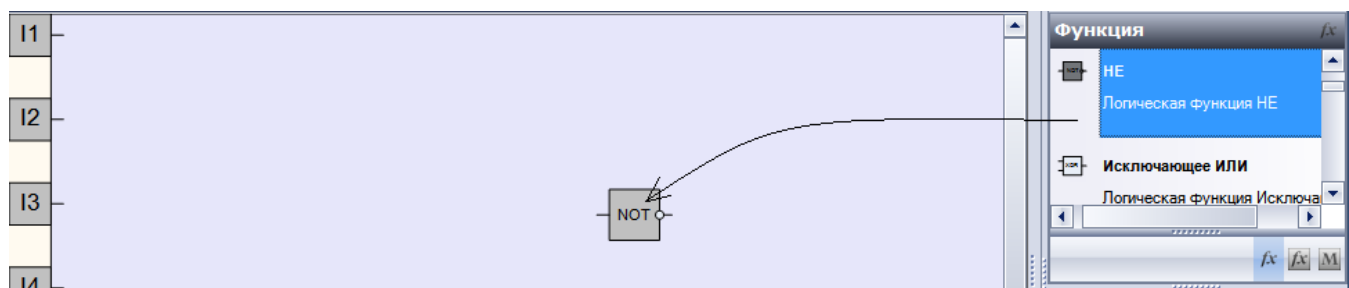


Рисунок 3.5 – Добавление логического элемента НЕ

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

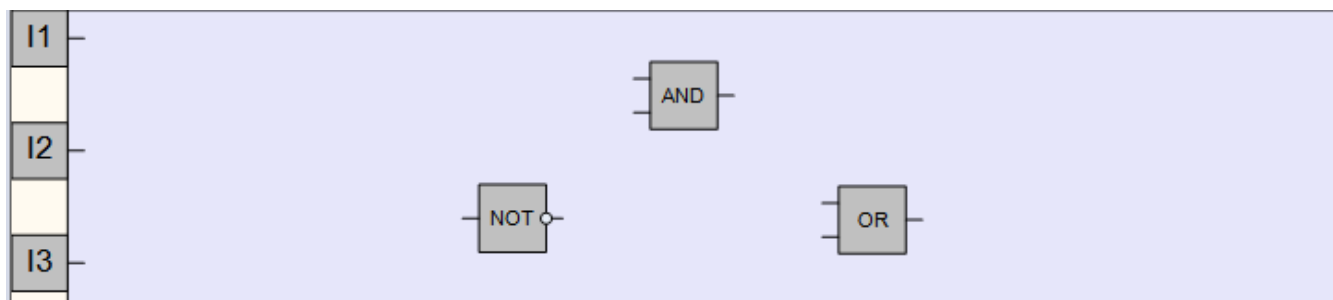


Рисунок 3.6 – Поле программы с логическими элементами НЕ, И, ИЛИ

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.

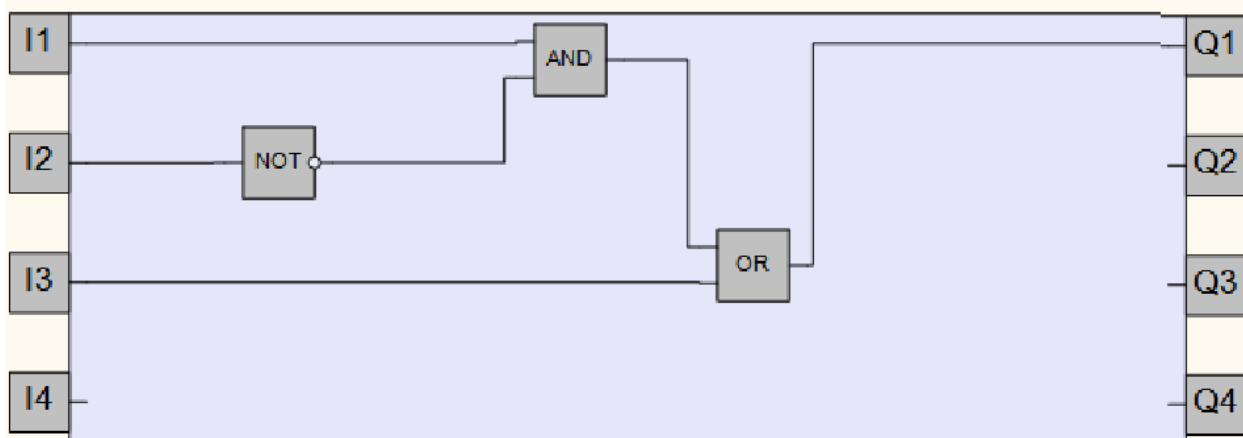


Рисунок 3.7 – Внешний вид программы после установки связей

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.

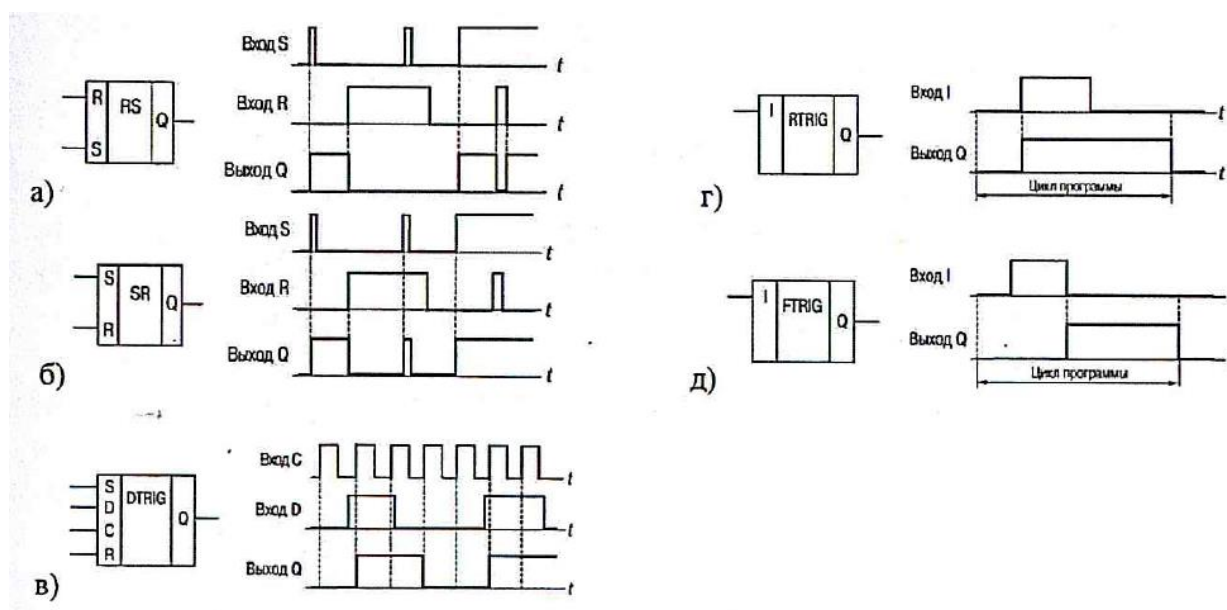


Рисунок 3.8 – Триггеры: *RS*-триггер (а), *SR*-триггер (б), *D*-триггер (в), детектор переднего фронта импульса (г), детектор заднего фронта импульса (д)

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 $Q1$.

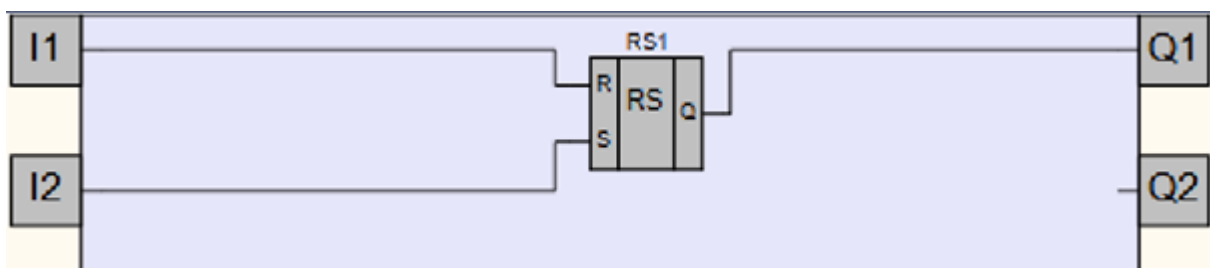


Рисунок 3.9 – Фрагмент программы с использованием RS -триггера

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).

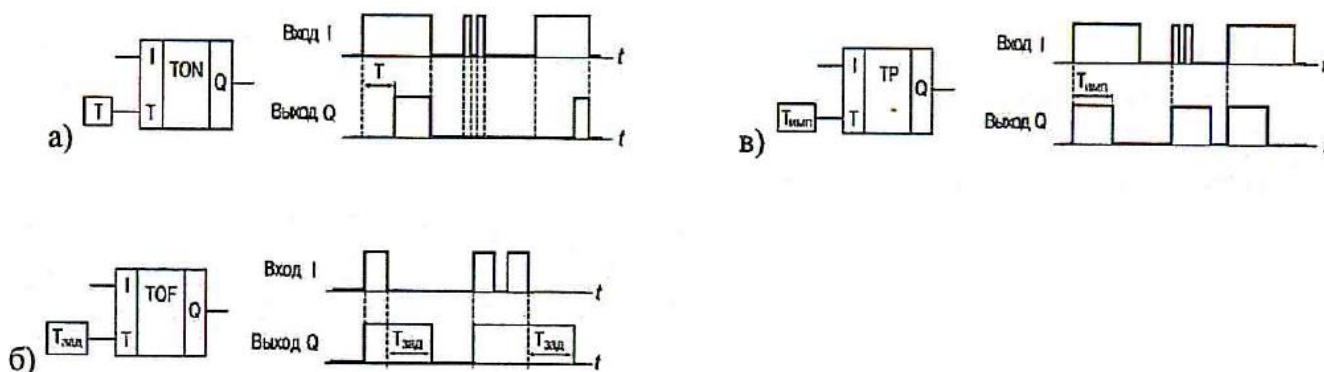


Рисунок 3.10 – Таймеры: а – с задержкой включения (TON), б – с задержкой выключения (TOF), в – формирователь импульса (TP)

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.

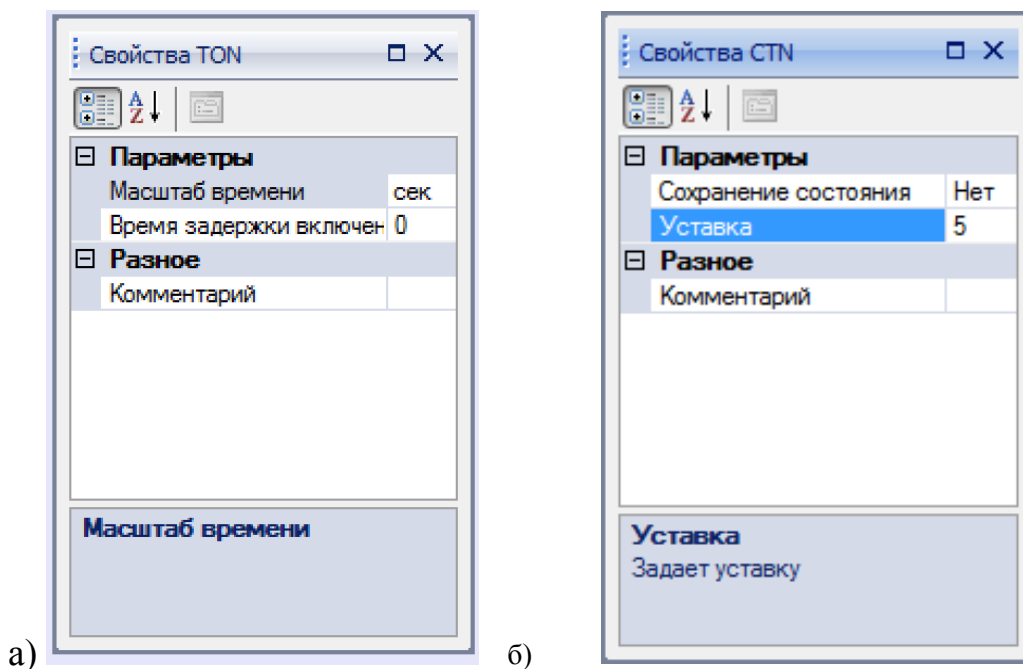


Рисунок 3.11 – Окна настроек таймера (а) и счетчика (б)

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.

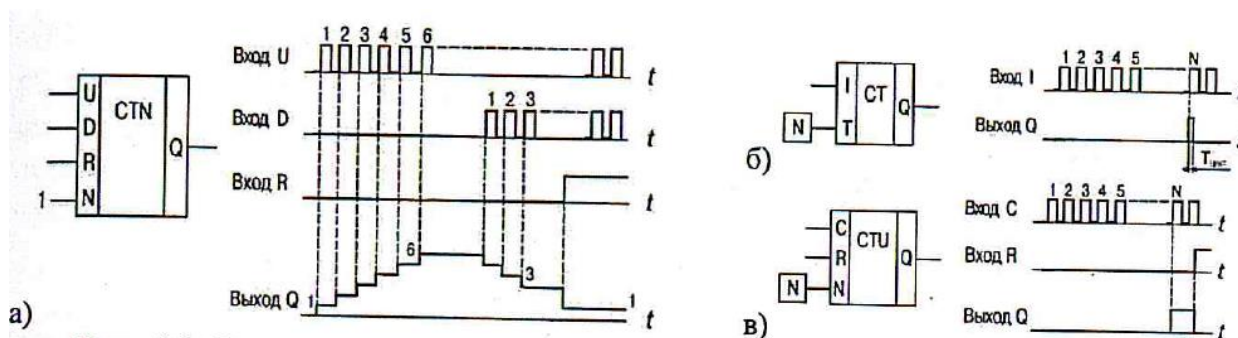


Рисунок 3.12 – Счетчики : а – универсальный (CTN), б – инкрементный счетчик с автосбросом (CT), в – инкрементный счетчик (CTU)

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*).

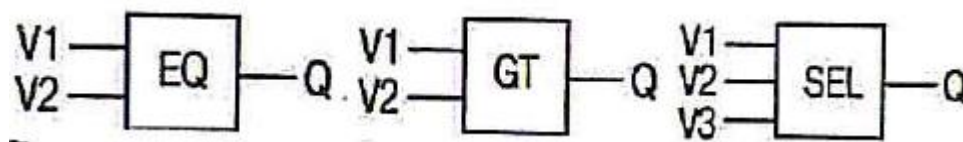


Рисунок 3.13 – Символьные обозначения функций компаратора

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*.

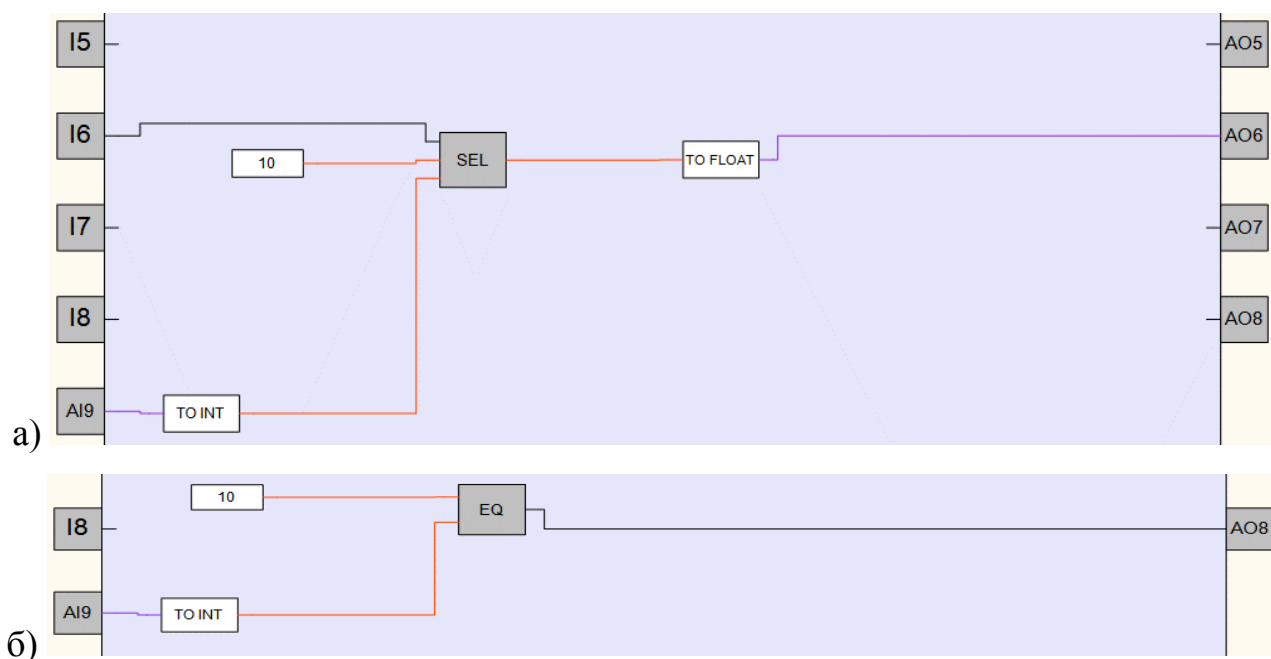


Рисунок 3.14 – Пример использования *SEL*-компаратора (а) и *EQ*-компаратора (б)

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.

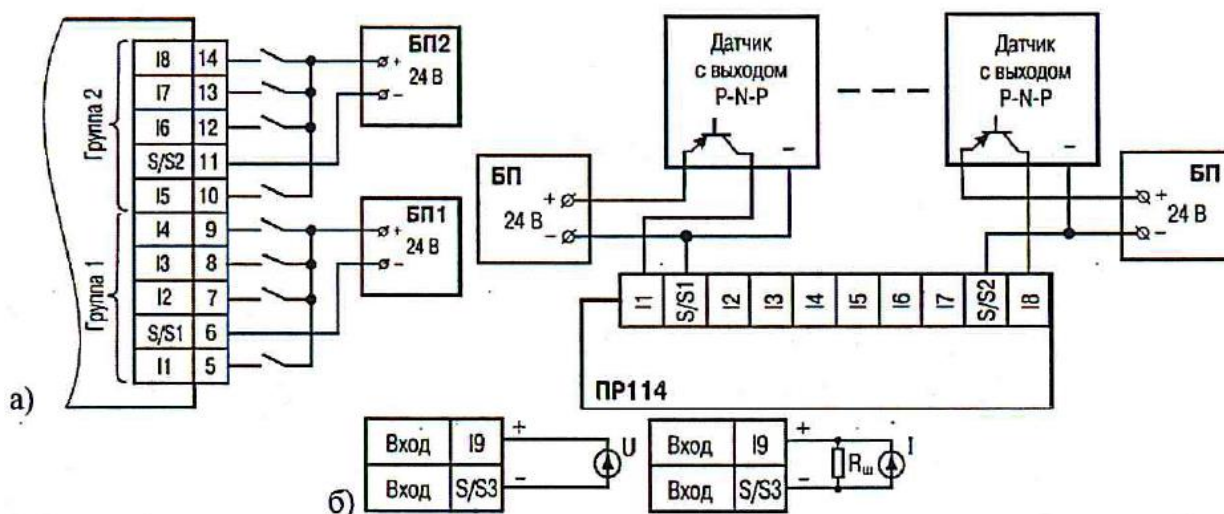


Рисунок 3.17 – Схема подключения входных цепей программируемого реле ОВЕН ПР114

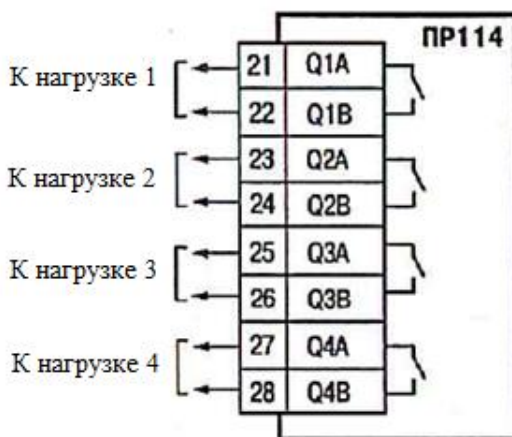


Рисунок 3.18 – Схема подключения выходных цепей реле ОВЕН ПР 114

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	$m^{\uparrow}$
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	–

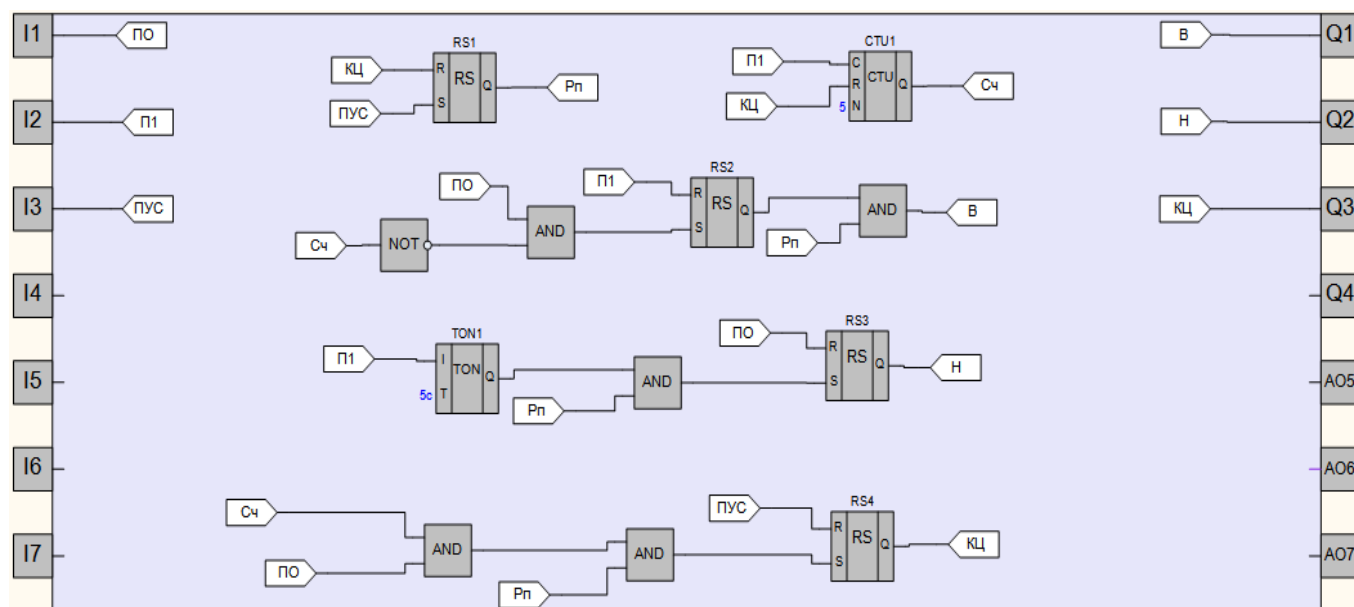


Рисунок 3.19 – Программа управления движением тележки

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.

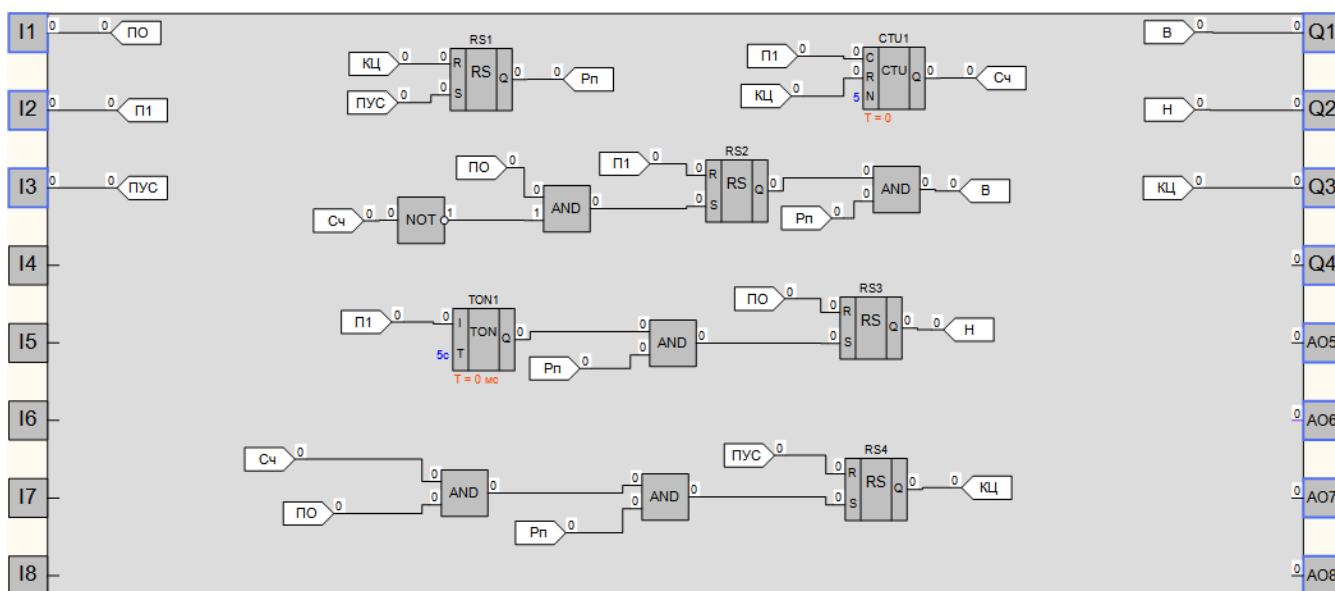


Рисунок 3.20 – Режим симулятора



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 $I9 - I12$ от входов $I1 - I8$?

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

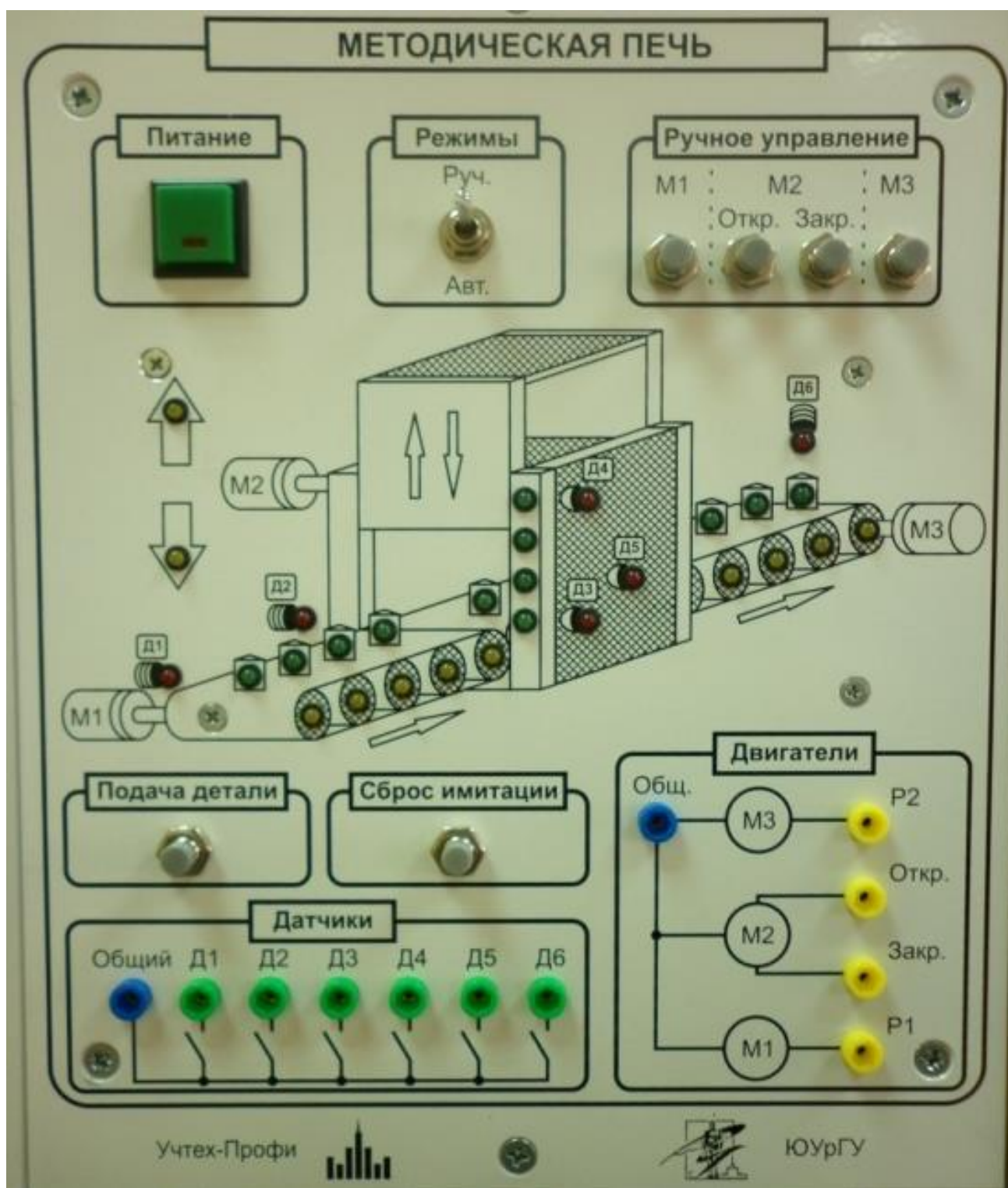


Рисунок 4.1 – Модуль имитации физического объекта (модуль ИФО)

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		

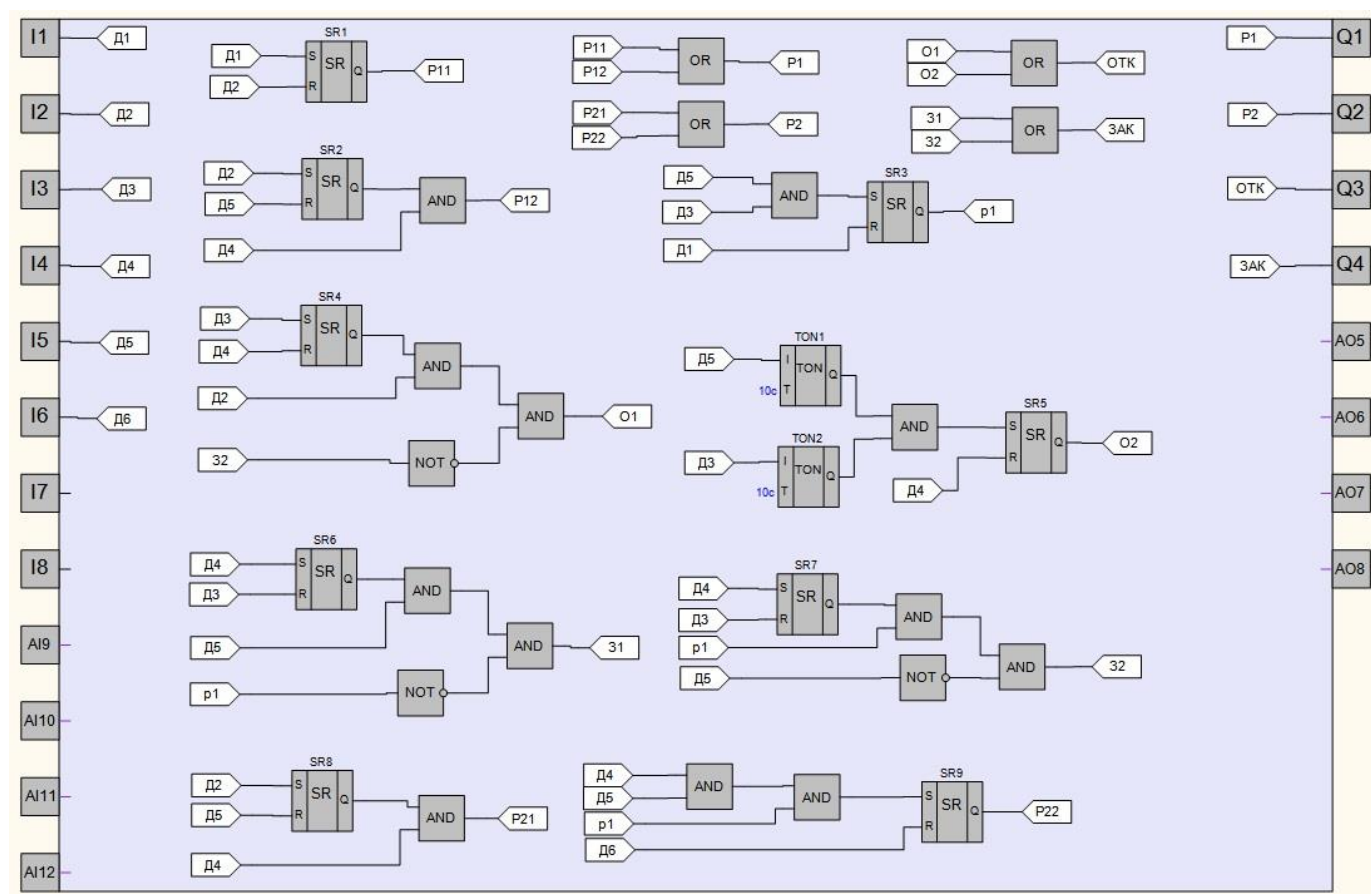


Рисунок 4.2 – Программа для реле ПР114 на языке FBD

- 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.

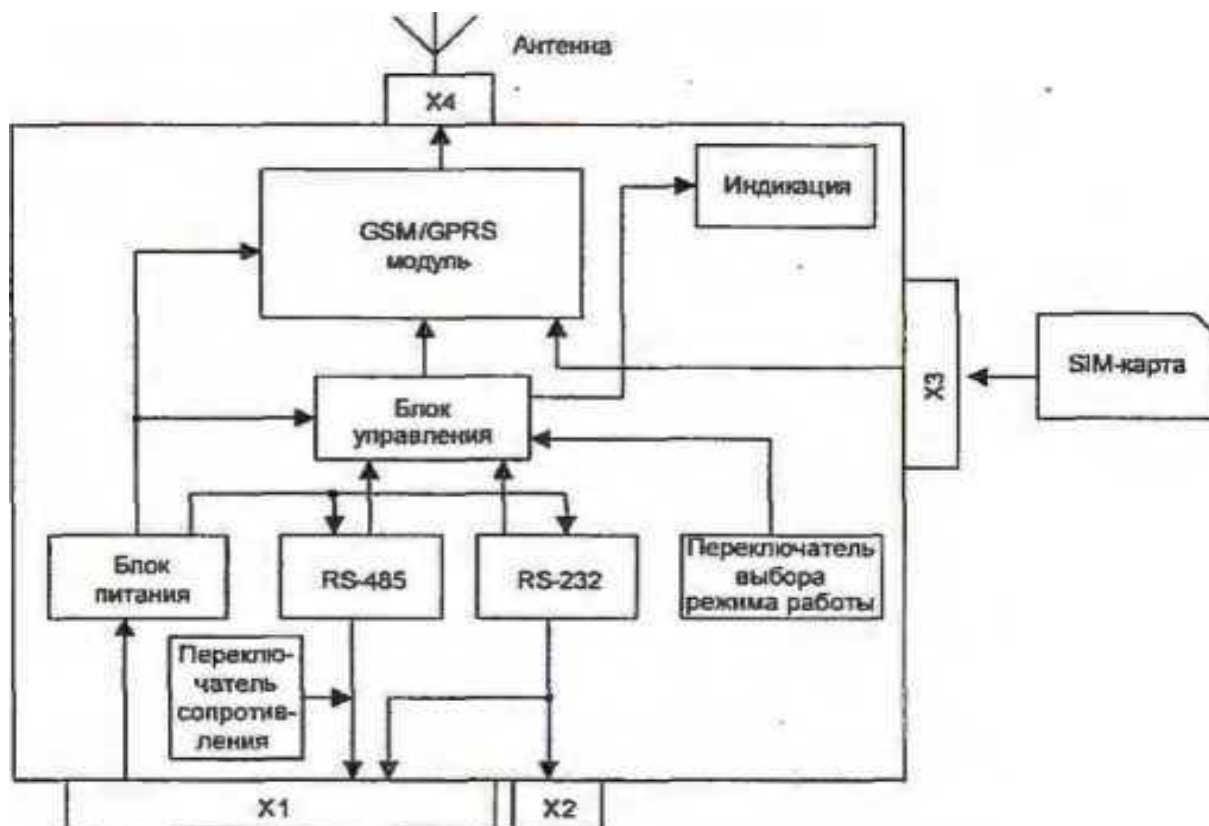


Рисунок 5.2 – Структурная схема модема

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.

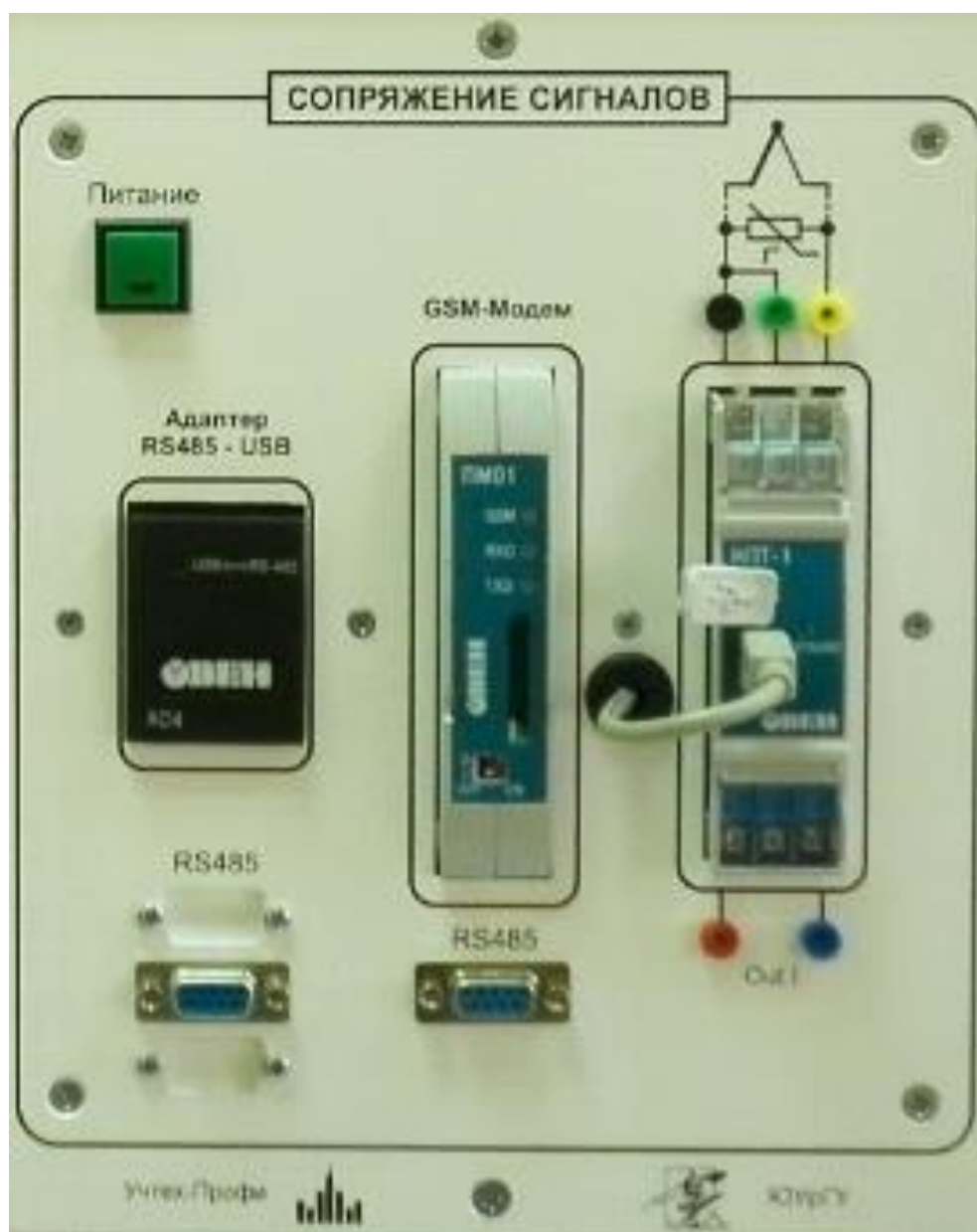


Рисунок 5.3 – Модуль сопряжения сигналов

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);

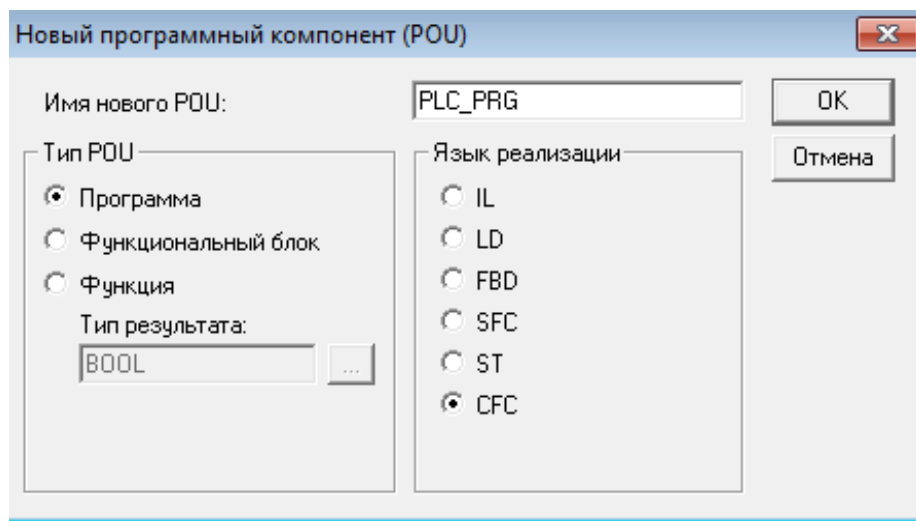


Рисунок 5.4 – Выбор языка программирования (*CFC*)

- 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*);

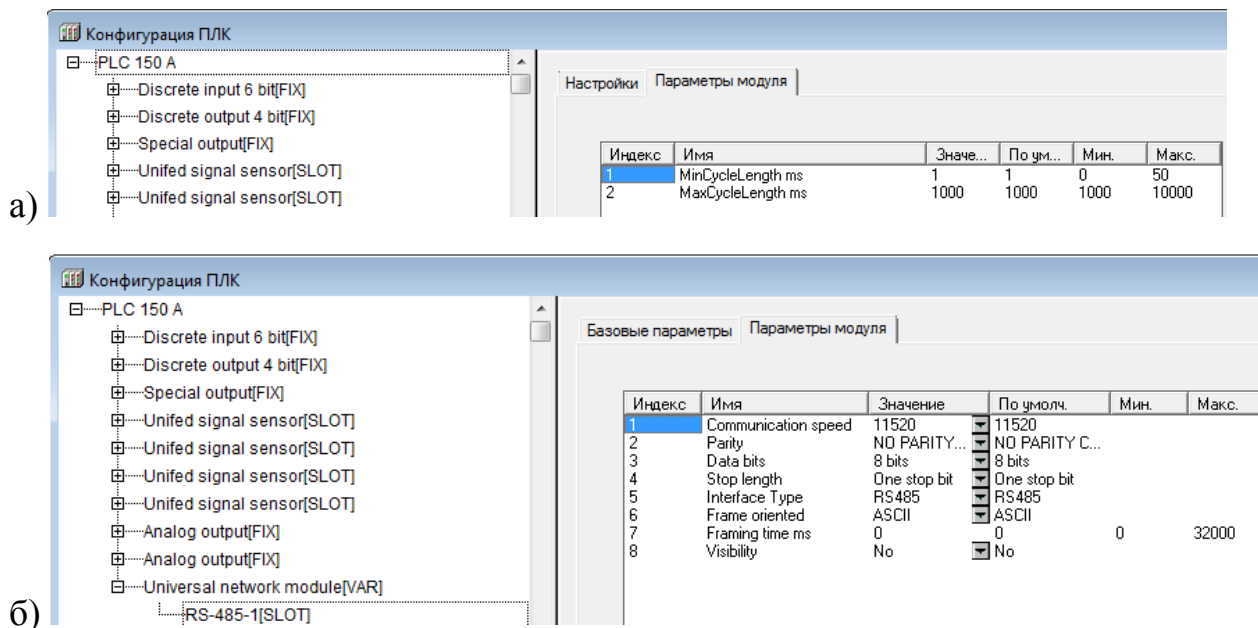


Рисунок 5.5 – Конфигурирование порта RS485 программируемого контроллера

```

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);

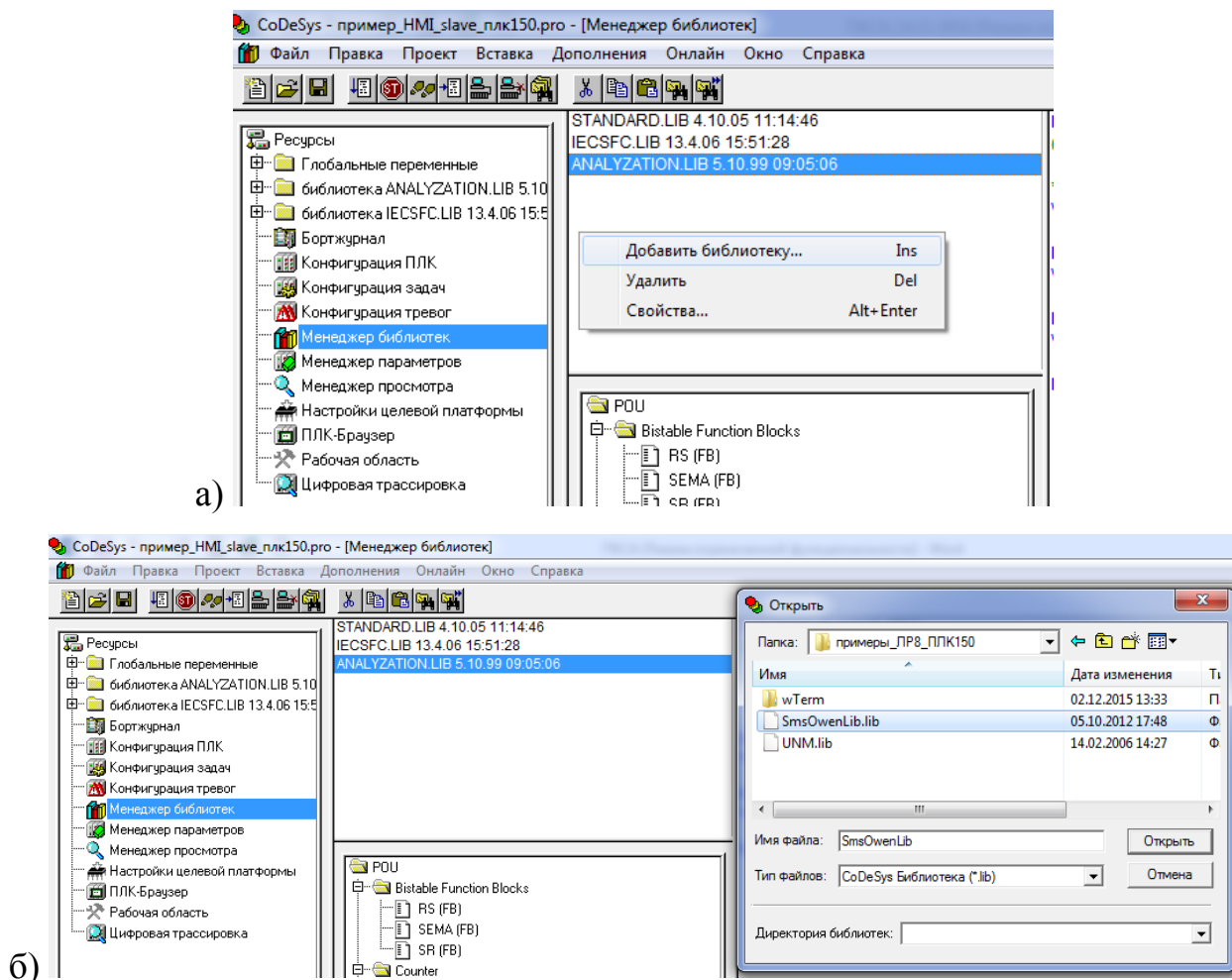


Рисунок 5.7 – Добавление библиотек

- 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».

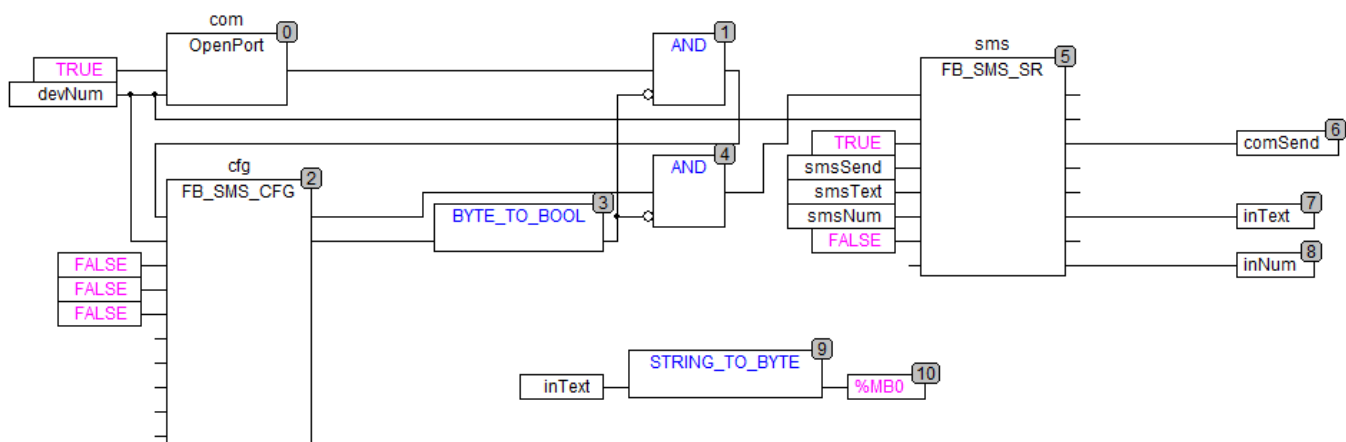


Рисунок 5.8 – Управляющая программа для получения/отправки SMS-сообщений

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 .

11 Учебное оборудование. Лабораторные стенды на заказ : сайт компании ООО НПП «Учтех — Профи». – Электрон. дан. – Челябинск. – Режим доступа : www.uralstend.ru.

12 Программное обеспечение *OWEN Logic* : сайт компании ОВЕН. – Электрон. дан. – Москва: ОВЕН, [2009] . – Режим доступа : http://www.owen.ru/catalog/programmnoe_obespechenie_owen_logic/opisanie

13 GSM/GPRS модемы ПМ01-24.В, ПМ01-24.АВ, ПМ01-220.В и ПМ01-220.АВ : руководство по эксплуатации. – Электрон. дан. – Москва: ОВЕН, [2009]. – 29 с. – Режим доступа : http://www.owen.ru/uploads/re_pm01_2321.pdf

Учебное пособие

Александр Иванович Сергеев
Антонина Михайловна Черноусова
Александр Сергеевич Русяев

**ПРОГРАММИРОВАНИЕ КОНТРОЛЛЕРОВ СИСТЕМ
АВТОМАТИЗАЦИИ**

ISBN 978-5-7410-1649-7

