Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет информационных технологий и управления

Кафедра систем управления

А. В. Марков, Т. В. Ляхор

ЭЛЕМЕНТЫ И УСТРОЙСТВА СИСТЕМ УПРАВЛЕНИЯ. ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Рекомендовано УМО по образованию в области информатики и радиоэлектроники в качестве пособия для специальности 1-53 01 07 «Информационные технологии и управление в технических системах»

Библиотека БГУИР

Минск БГУИР 2018

УДК 681.58(076.5)
ББК 32.96-04я73
М27

Р е ц е н з е н т ы:

кафедра информационных технологий Белорусского государственного университета (протокол №2 от 05.10.2017);

доцент кафедры управления информационными ресурсами Академии управления при Президенте Республики Беларусь, кандидат технических наук, доцент Н. И. Белодед;

заместитель генерального директора по научной и инновационной работе государственного научного учреждения «Объединенный институт проблем информатики Национальной академии наук Беларуси», доктор военных наук, доцент С. В. Кругликов

Марков, А. В.

М27 Элементы и устройства систем управления. Лабораторный практикум : пособие / А. В. Марков, Т. В. Ляхор. – Минск : БГУИР, 2018. – 82 с. : ил.

ISBN 978-985-543-401-7.

Содержит основные положения и принципы работы программируемых логических контроллеров и основных элементов управления.

Приведены описания и порядок выполнения 6 лабораторных работ по курсу «Элементы и устройства систем управления».

УДК 681.58(076.5)
ББК 32.96-04я73

ISBN 978-985-543-401-7

© Марков А. В., Ляхор Т. В., 2018
© УО «Белорусский государственный университет информатики и радиоэлектроники», 2018

Библиотека БГУИР

СОДЕРЖАНИЕ

Библиотека БГУИР

1 АРХИТЕКТУРА ПРОГРАММНОГО И АППАРАТНОГО ОБЕСПЕЧЕНИЯ

1.1 Описание лабораторного оборудования

Лабораторный стенд «Промышленная автоматика OMRON» предназначен для обучения студентов специальности «Информационные технологии и управление в технических системах» работе, связанной с автоматизацией различных отраслей промышленности.

Промышленный логический контроллер Omron CP1L-E позволяет производить симуляцию любого технологического процесса малой и средней сложности. Он легко программируется и позволяет использовать режим мониторинга.

Лабораторный стенд поставляется в подготовленном к работе состоянии. Все необходимые для работы компоненты закреплены на монтажной рейке и готовы к эксплуатации. Во время выполнения работы студенты могут соединять различные рабочие точки стенда между собой для передачи цифровых и аналоговых сигналов.

В состав стенда входят:

– программируемый логический контроллер Omron CP1L-E, предназначенный для выполнения программ управления различными технологическими процессами;

– источник питания Omron S8VK-G06024, предназначенный для обеспечения питания 24 В постоянного напряжения;

– Ethernet-коммутатор Omron W4S1-03B, предназначенный для интеграции в единую сеть Ethernet различных устройств автоматики (ПК, компактного промышленного контроллера и панели оператора);

– преобразователь частоты Omron 3G3MX2-AB002-E, предназначенный для регулирования частоты вращения трехфазного асинхронного электродвигателя;

– трехфазный асинхронный электродвигатель АИС71В4У3, закрепленный на лицевой панели стенда;

– программируемый терминал Omron NB5Q-TW01B, предназначенный для реализации программируемых сенсорных пультов оператора;

Библиотека БГУИР

– весоизмерительный тензометрический модуль в сборе со специальным кронштейном и винтом, регулирующим прижимное усилие, предназначенный для имитации измерения массы, давления или веса;

– измеритель Omron K3HB-V, предназначенный для совместной работы с весоизмерительным тензометрическим модулем.

Также на макете присутствуют такие элементы, как реле, светодиодные лампы индикации, потенциометры, гнезда для соединений, тумблеры, маховик и вольтметр.

Внешний вид лабораторного стенда представлен на рисунке 1.1, а его электрические принципиальные схемы – на рисунках 1.2 и 1.3.

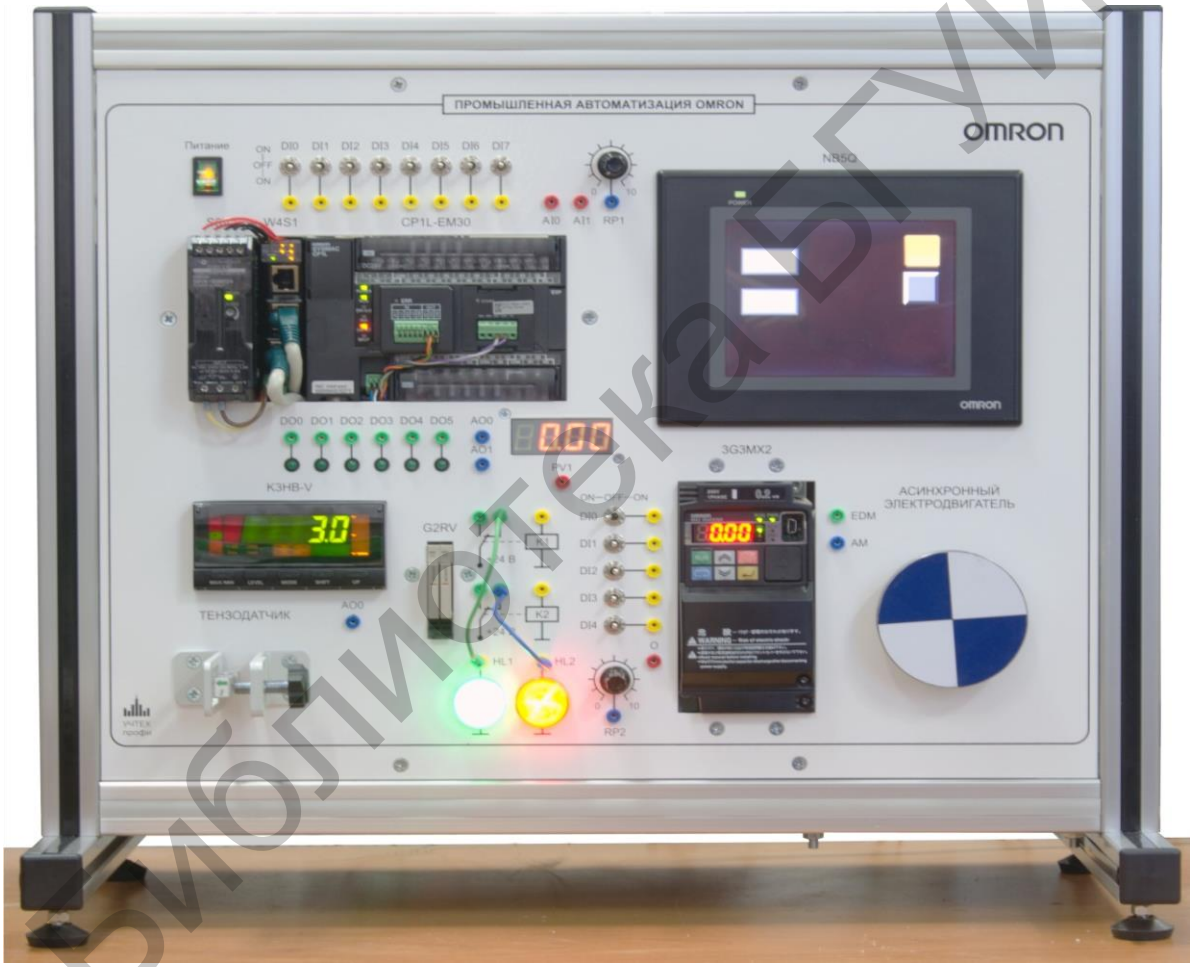


Рисунок 1.1 – Лабораторный стенд «Промышленная автоматика Omron»

Библиотека БГУИР

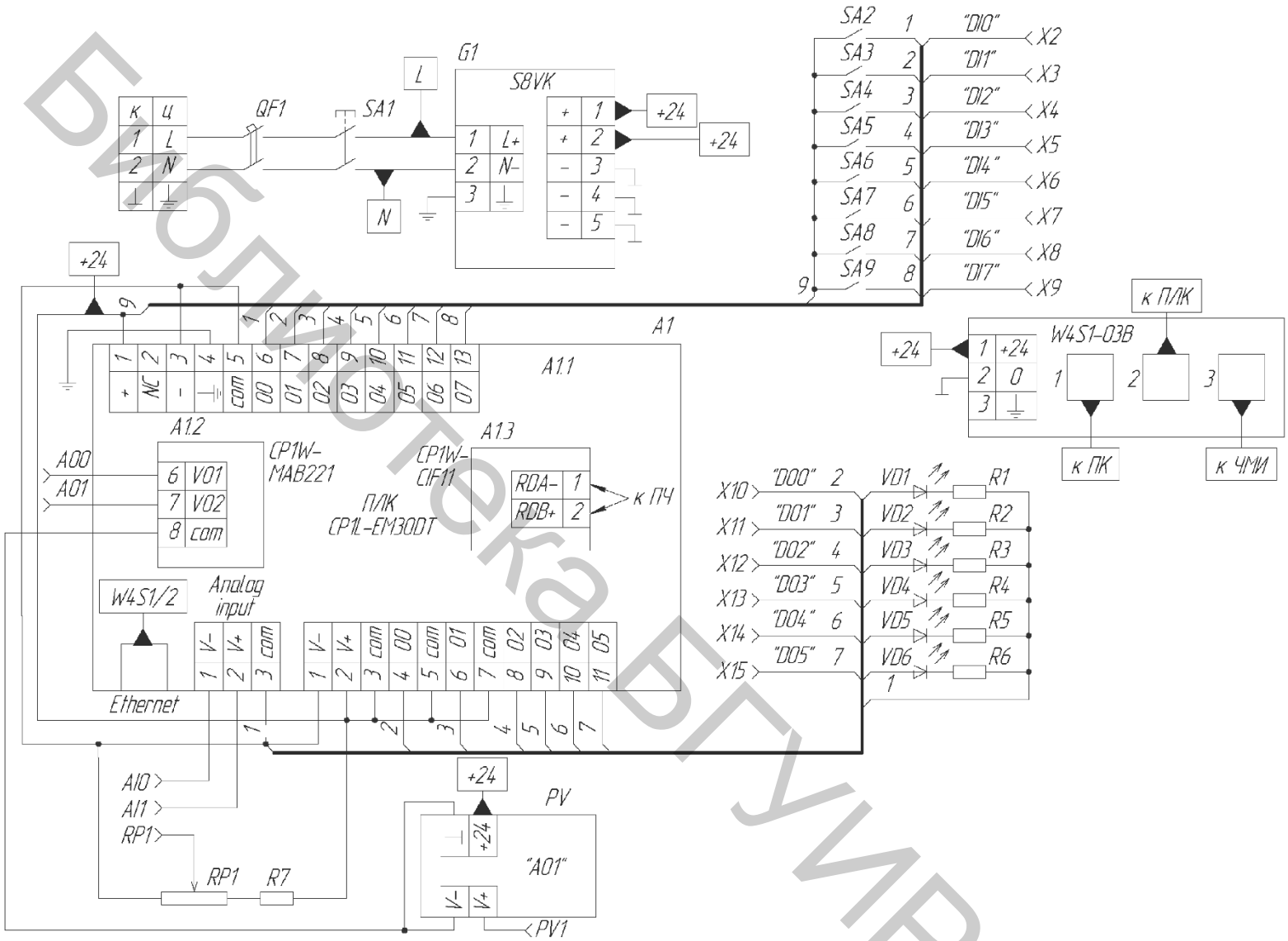


Рисунок 1.2 – Схема электрическая принципиальная. Часть первая

Библиотека БГУИР

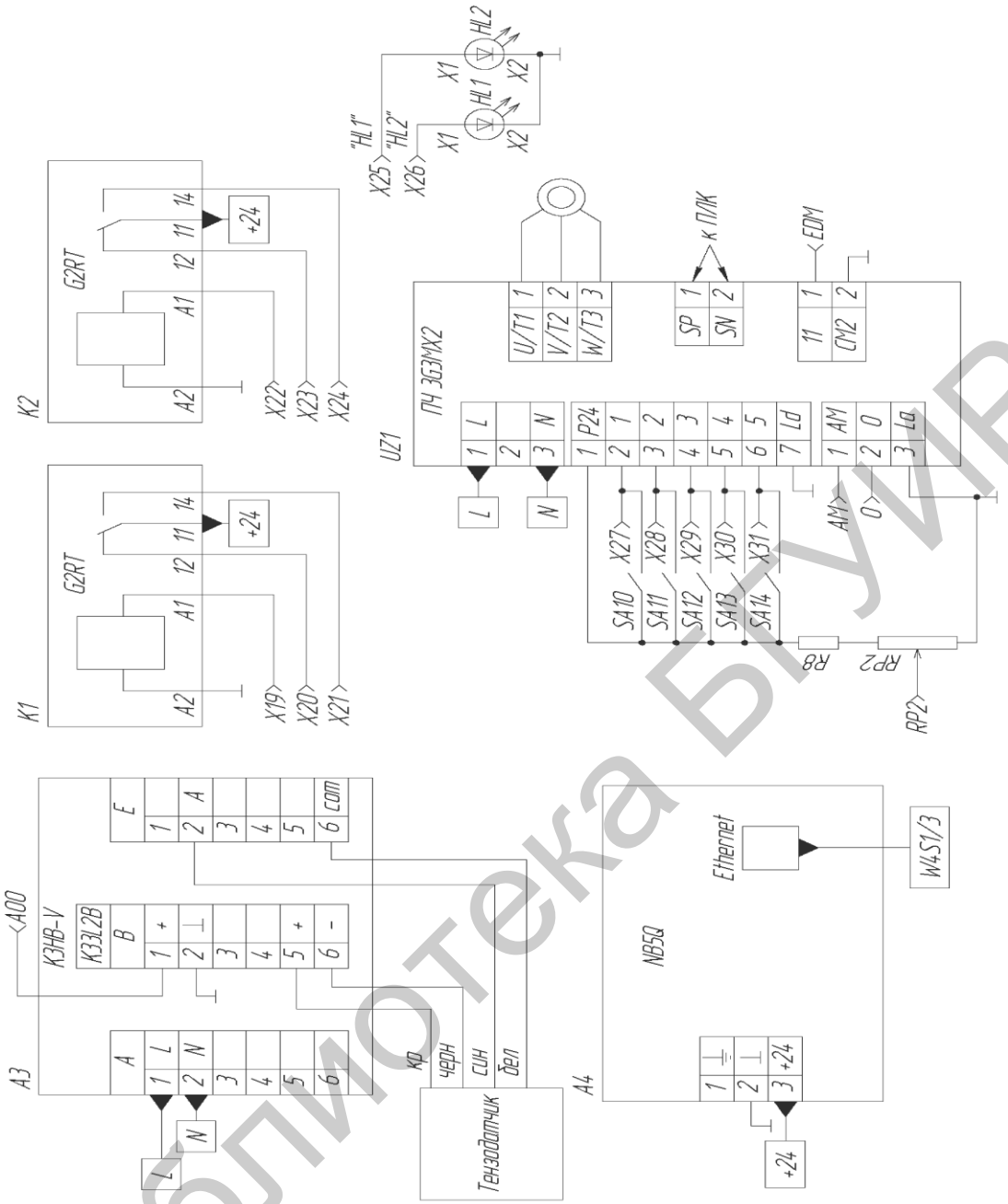


Рисунок 1.3 – Схема электрическая принципиальная. Часть вторая

1.2 Архитектура программного обеспечения

Запуск программы CX-Programmer осуществляется через соответствующий ярлык на рабочем столе.

После запуска программы появляется окно, представленное на рисунке 1.4.

Библиотека БГУИР

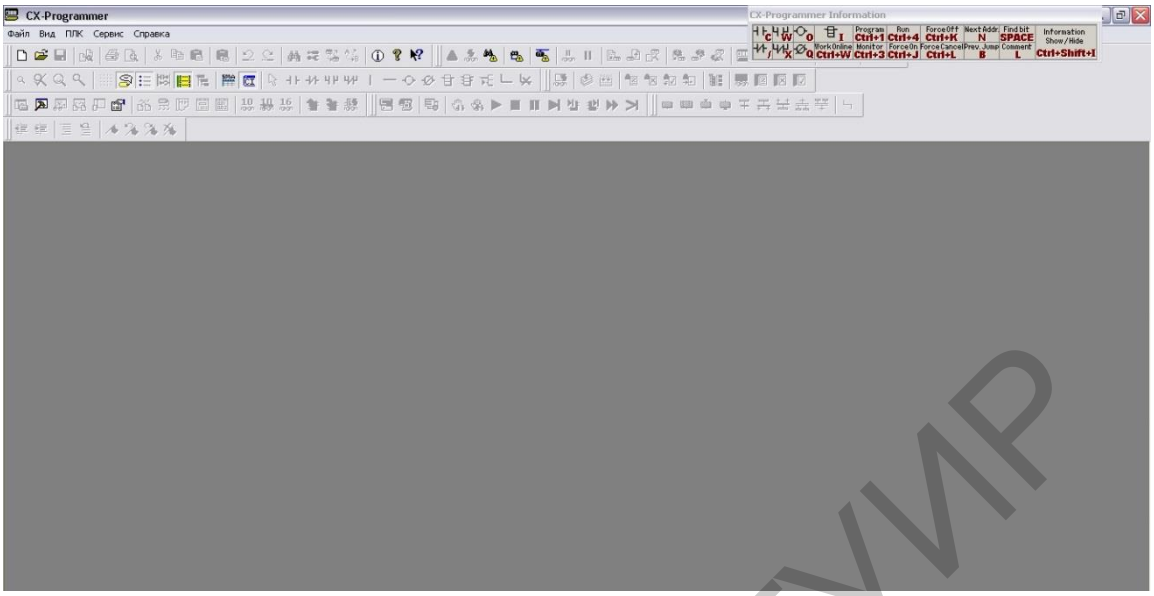


Рисунок 1.4 – Рабочее пространство программы CX-Programmer

Для того чтобы создать новый файл объекта, необходимо выбрать в главном окне программы в выпадающем меню «Файл» пункт «Создать».

При этом должно появиться окно (рисунок 1.5), в котором необходимо задать нужное имя контроллера (поле «Имя устройства»), тип контроллера – CP1L-E (поле «Тип устройства»), а также тип связи с контроллером – Ethernet (FINS/TCP) (поле «Тип сети»). Выбрав нужные параметры, следует нажать кнопку «OK» для подтверждения выбора или «Отмена» – для отмены.

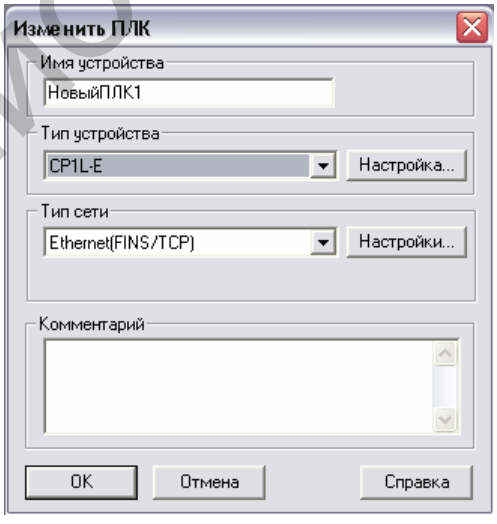


Рисунок 1.5 – Окно выбора параметров контроллера

Для настройки сетевого адреса ПЛК необходимо напротив поля «Тип сети» нажать кнопку «Настройки». В открывшемся окне выбрать закладку «Driver»

Библиотека БГУИР

(рисунок 1.6), в которой необходимо установить переключатель в положение «Ethernet – Hub connection». В полях «IP Address» и «Port Number», ставших активными, установить следующие значения: адрес – 192.168.0.130, порт – 9600.

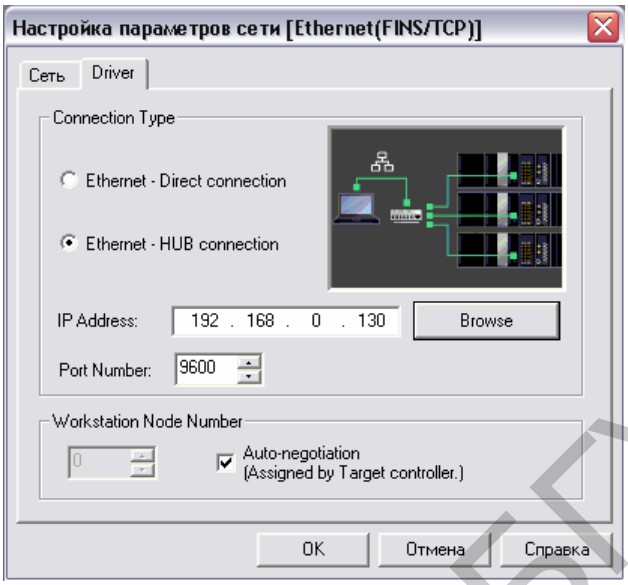


Рисунок 1.6 – Окно настройки параметров сети

Для открытия уже существующего проекта необходимо в выпадающем меню «Файл» выбрать пункт «Открыть». Появится окно выбора проекта (рисунок 1.7).

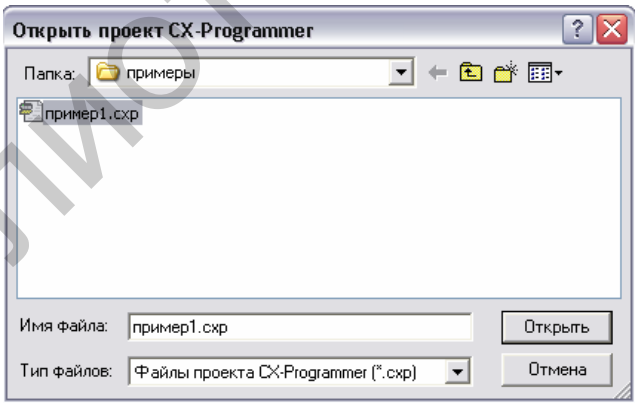


Рисунок 1.7 – Окно выбора существующего проекта

Для выбора необходимо выделить указателем мыши нужный проект и нажать кнопку «Открыть» для подтверждения или «Отмена» – для отмены выбора.

После этого вид программы должен измениться – появится окно менеджера проекта и окно редактирования программы (рисунок 1.8).

Библиотека БГУИР

Всю рабочую область программы CX-Programmer (рисунок 1.8) можно разделить на несколько областей:

- окно менеджера проекта;
- рабочая область;
- панели инструментов;
- строка сообщений;
- строка состояния.

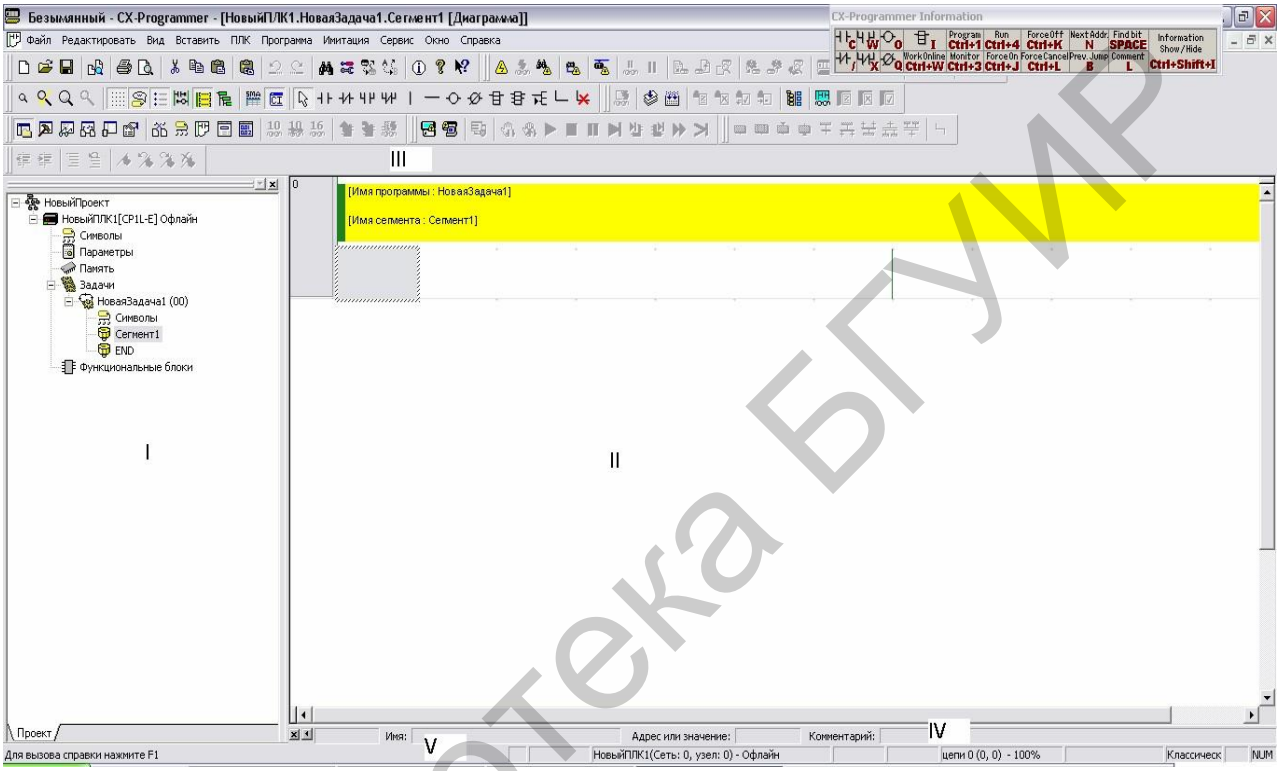


Рисунок 1.8 – Главное окно программы CX-Programmer

В зависимости от состояния проекта те или иные пункты и элементы будут либо в активном состоянии (подсвечены), либо нет. Представленная структура проекта используется для надлежащего хранения и размещения всех данных и программ.

Менеджер проекта имеет иерархическую структуру:

1. В пункте «Параметры» (рисунок 1.9) можно выбрать режим работы контроллера, в который он переходит при подаче питания. Для этого надо в закладке «Запуск» выбрать требуемый режим работы.

2. В закладке «Символы» обеспечивается возможность присвоения символического имени конкретному адресу, ячейке памяти (рисунок 1.10).

Библиотека БГУИР

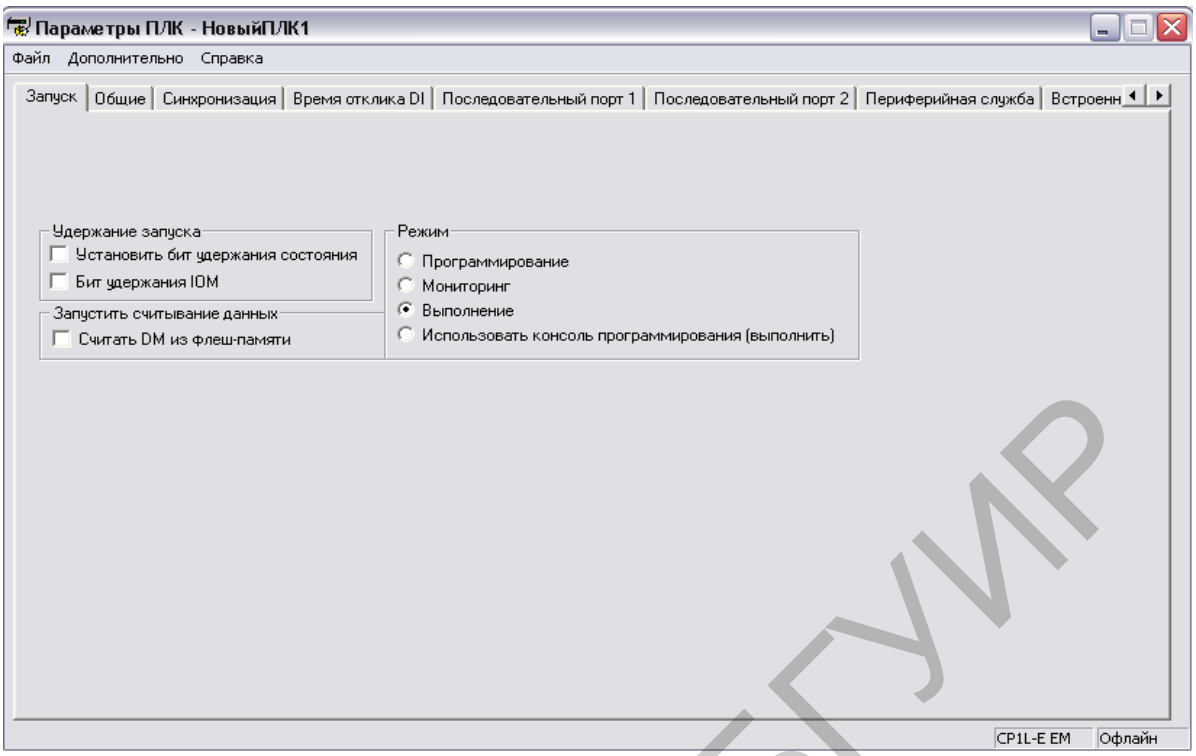


Рисунок 1.9 – Окно настройки свойств контроллера

Название	Тип данных	Адрес / значение	Расположе...	Испо...	Комментарий
P_0_02s	BOOL	CF103		Работа	Бит тактовых импульсов 0,02 с
P_0_1s	BOOL	CF100		Работа	Бит тактовых импульсов 0,1 с
P_0_2s	BOOL	CF101		Работа	Бит тактовых импульсов 0,2 с
P_1min	BOOL	CF104		Работа	Бит тактовых импульсов 1 мин
P_1s	BOOL	CF102		Работа	Бит тактовых импульсов 1,0 с
P_AER	BOOL	CF011		Работа	Флаг ошибки доступа
P_CIO	WORD	A450		Работа	Параметр области CIO

Рисунок 1.10 – Внешний вид закладки «Символы»

3. С помощью закладки «Задачи» осуществляется создание программ и управление несколькими программами для одного контроллера в рамках одного проекта.

4. В закладке «Сегмент» ведется разработка основной программы (рисунок 1.8).

5. С помощью закладки «Функциональный блок» осуществляется создание функциональных блоков и управление ими.

Доступ к операциям по работе с закладками осуществляется путем однократного нажатия правой кнопки мыши. В выпадающем списке пользователю доступны различные манипуляции над пунктами менеджера проекта.

Панели инструментов содержат наиболее часто используемые инструменты и предназначены для повышения комфорта при использовании программного обеспечения.

Библиотека БГУИР

Предварительно перед работой с управляющей программой необходимо выполнить программное конфигурирование используемого оборудования. То есть кроме выбора типа ПЛК необходимо настроить сетевой адрес ПЛК в сети Ethernet (рисунок 1.11) и при необходимости сетевые параметры встроенного последовательного порта 2, в котором установлена плата расширения RS485 (ModBus).

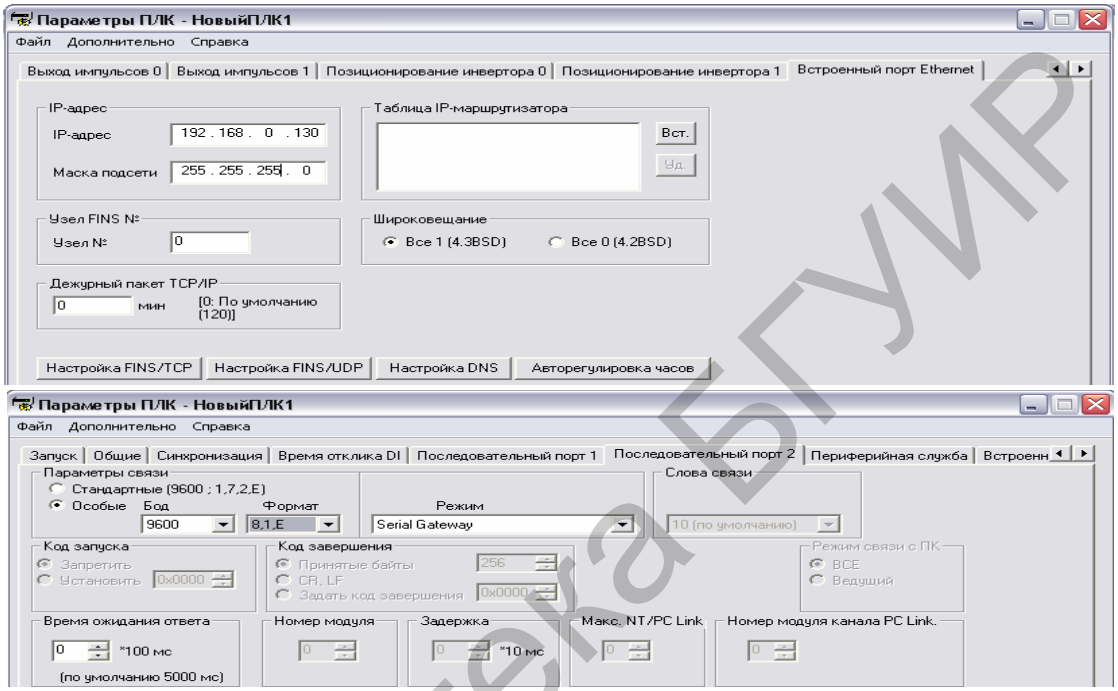


Рисунок 1.11 – Настройка сетевых возможностей ПЛК

1.3 Архитектура аппаратных средств

В лабораторном комплексе (приложение А) используется промышленный логический контроллер Omron CP1L-E. Он представляет собой компактный промышленный контроллер с множеством встроенных функций [1]. На рисунке 1.12 показан внешний вид контроллера.

Библиотека БГУИР

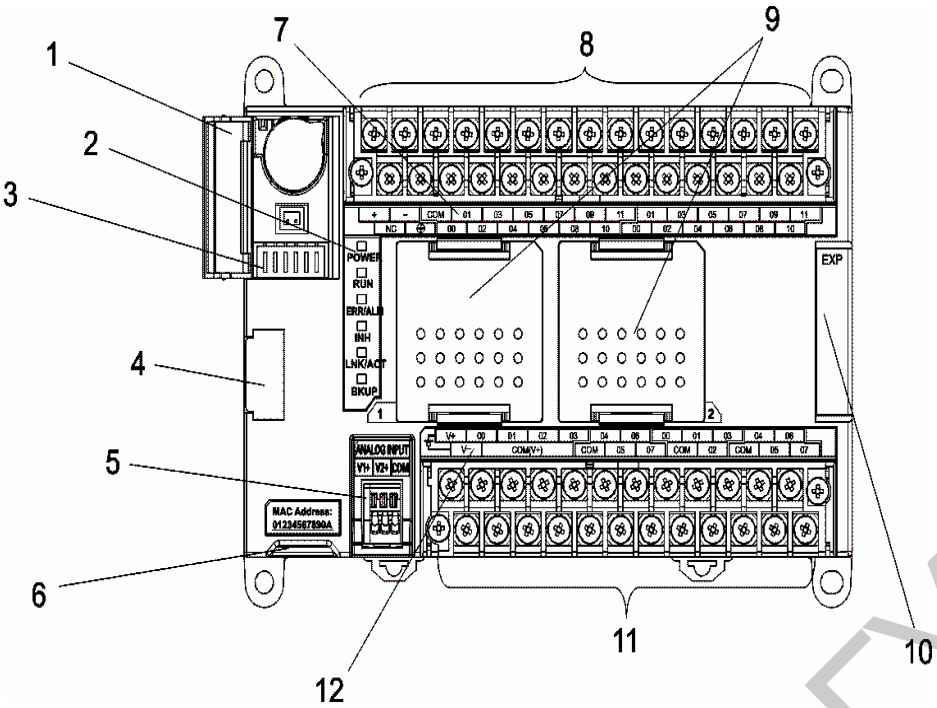


Рисунок 1.12 – Omron CP1L-E

На рисунке 1.13 представлены функциональные схемы входов контроллера.

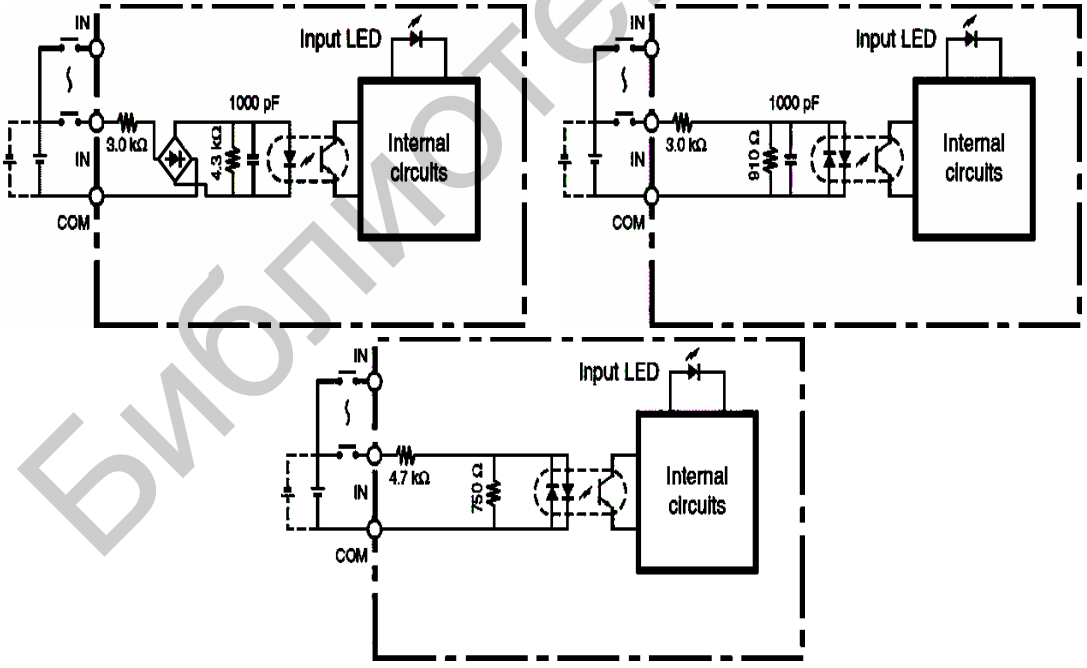


Рисунок 1.13 – Функциональные схемы входов

Библиотека БГУИР

Общая характеристика входов контроллера представлена в таблице 1.1.

Таблица 1.1 – Характеристики входов контроллера

Параметр	Значение		
	Высокоскоростной счетный вход	Вход прерывания и быстрого отклика	Обычный вход
	CIO0.00…0.03	CIO0.04…0.09	CIO0.10… 0.11
Входное напряжение	24В		
Тип подключения	Двух- и трехпроводные датчики		
Входное сопротивление	3,0 кОм		4,7 кОм
Входной ток	7,5 мА		5 мА
Напряжение уровня логической 1	От 17 В		От 14,5 В
Напряжение уровня логического 0	До 5 В, 1 мА		
Задержка включения/выключения	До 2,5/2,5 мкс	До 50/50 мкс	До 1/1 мс

На рисунке 1.14 представлена схема [2] подключения входов к клеммнику ПЛК.

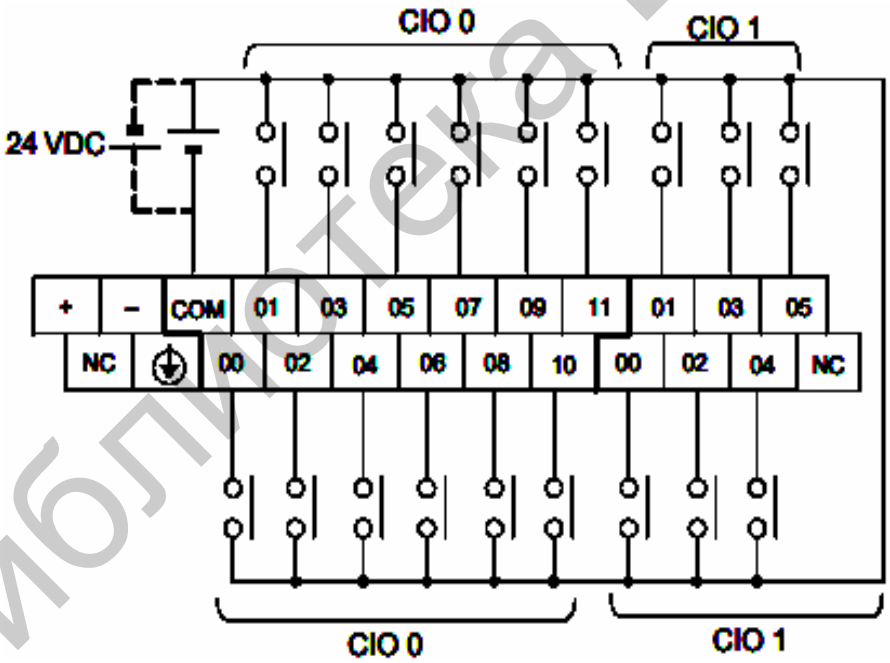


Рисунок 1.14 – Схема подключения входных сигналов для CP1L-E

На рисунке 1.15 представлена схема подключения выходов к клеммнику ПЛК.

Библиотека БГУИР

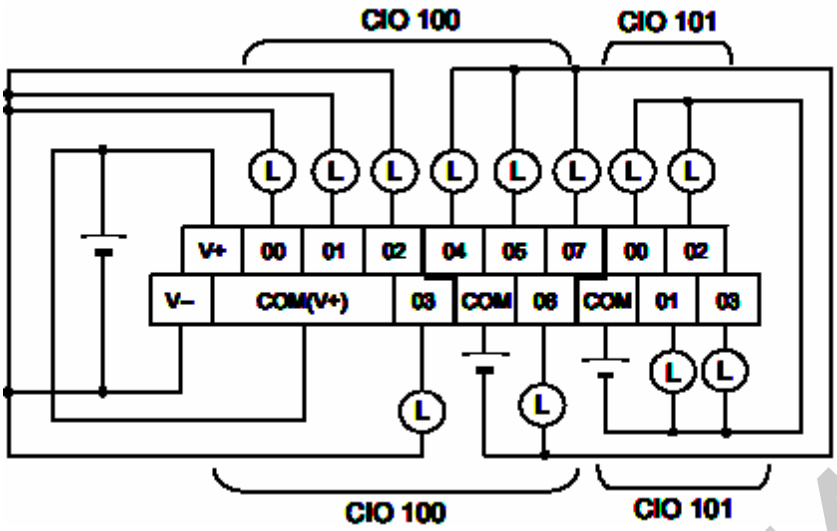


Рисунок 1.15 – Схема подключения выходных сигналов для CP1L-E

На 18 входов имеется одна общая точка. При этом внешний источник питания подсоединяется одним полюсом к общей точке COM, а другим – к используемому входу ПЛК [3] (полярность не имеет значения).

Общая характеристика выходов ПЛК представлена в таблице 1.2.

Таблица 1.2 – Характеристики выходов контроллера

		Входы	Значение
Тип выхода		Все	Транзисторный
Нагрузка при 4,5–30 В	Максимальная	Все	0,3 А/выход, 0,9 А/блок, 3,6 А/модуль
	Минимальная	Все	10 мА/выход
Ток утечки		Все	0,1 мА
Задержка включения/выключения		CIO100.00…100.03	До 0,1/0,1 мс
		CIO100.04…100.07 CIO101.00…101.03	До 0,1/1 мс

На рисунке 1.16 представлен пример распределения памяти [4] для системы, включающей в себя модуль ЦПУ CP1L-EM30DT1-D и три модуля расширения :CP1W-AD041 (аналоговый ввод, 4 канала), CP1W-DA041 (аналоговый вывод, 4 канала) и CP1W-40EDT1 (дискретный ввод/вывод, 24/16 входов/выходов). В соответствующих полях показаны адреса соответствующих входов и выходов модуля ЦПУ и модулей расширения.

Библиотека БГУИР

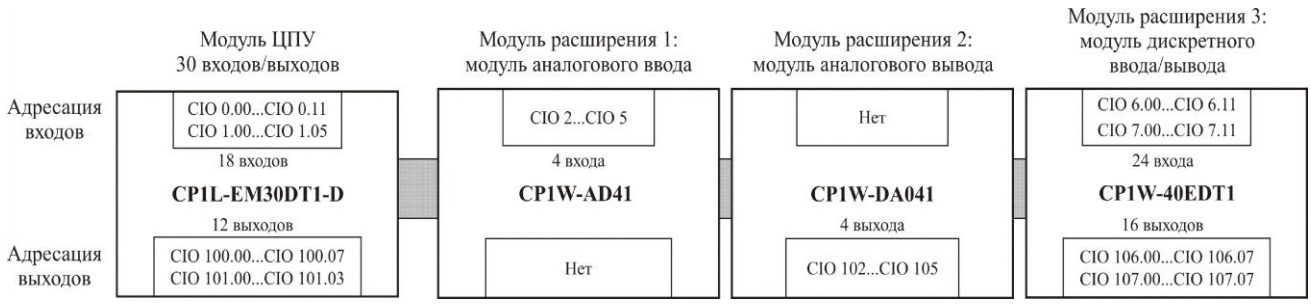


Рисунок 1.16 – Пример распределения памяти модулей ЦПУ и расширения

В таблице 1.3 представлена карта памяти для ПЛК CP1L-E.

Таблица 1.3 – Характеристики выходов контроллера

Адресация памяти ПЛК	Пользовательская адресация	Описание
0B100…0B7FF	–	Зарезервировано системой
0B800…0B801	TK0…TK31	Область флагов задач
0B802…0B83F	–	Зарезервировано системой
0B840…0B9FF	A0…A447	Вспомогательная область (только чтение)
0BA00…0BBFF	A448…A959	Вспомогательная область (чтение/запись)
0BC00…0BDFF	–	Зарезервировано системой
0BE00…0BEFF	T0000…T4095	Флаги выполнения таймеров
0BF00…0BFFF	C0000…C4095	Флаги выполнения счетчиков
0C000…0D7FF	CIO0…CIO6143	Область ввода/вывода
0D800…0D9FF	H000…H511	Область хранения H
0DA00…0DDFF	–	Зарезервировано системой
0DE00…0DFFF	W0…W511	Рабочая область
0E000…0EFFF	T0000…T4095	Текущее значение таймеров
0F000…0FFFF	C0000…C4095	Текущее значение счетчиков
10000…17FFF	D0…D32767	Память данных DM
18000…FFFFF	–	Зарезервировано системой

Библиотека БГУИР

2 ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ LD

2.1 Языки программирования ПЛК

Языки программирования для программируемых логических контроллеров описывает третья часть стандарта IEC 61131. Эта часть определяет синтаксис и семантику единого набора языков, который состоит из двух текстовых языков – список инструкций (IL – Instruction List) и структурированный текст (ST – Structured Text) – и трех графических – релейно-контактные схемы, или релейные диаграммы (LD – Ladder Diagram), диаграммы функциональных блоков (FBD – Function Block Diagram) и последовательные функциональные схемы (SFC – Sequential Function Chart).

В данных языках заложена методология структурного программирования, позволяющая пользователю представить автоматизируемый процесс в наиболее простой и понятной форме.

Языки IEC 61131-3 базируются на следующих принципах:

– вся программа разбивается на множество функциональных элементов – Program Organization Units (POU), каждый из которых может состоять из функций, функциональных блоков и программ; любой элемент IEC-программы может быть сконструирован иерархически из более простых элементов;

– стандарт требует строгой типизации данных; указание типов данных позволяет легко обнаруживать большинство ошибок в программе до ее исполнения;

– имеются средства для исполнения разных фрагментов программы в разное время, с разной скоростью, а также параллельно;

– для выполнения операций в определенной последовательности, которая задается моментами времени или событиями, используется специальный язык последовательных функциональных схем (SFC);

– стандарт поддерживает структуры для описания разнородных данных;

– стандарт обеспечивает совместное использование всех пяти языков, поэтому для каждого фрагмента задачи может быть выбран любой, наиболее удобный, язык;

– программа, написанная для одного контроллера, может быть перенесена на любой контроллер, совместимый со стандартом МЭК 61131-3.

Библиотека БГУИР

Для программирования ПЛК Omron CP1L-E используются следующие стандартизованные языки:

– LD (Ladder Diagram) – графический язык. Представляет собой программную реализацию электрических схем на базе электромагнитных реле.

– FBD (Function Block Diagram) – графический язык. Функциональный блок выражает некую подпрограмму. Каждый блок имеет входы (слева) и выходы (справа). Программа создается путем соединения множества функциональных блоков.

– ST (Structured Text) – по структуре и синтаксису ближе всего к языку программирования Паскаль. Удобен для написания больших программ и работы с аналоговыми сигналами и числами с плавающей точкой.

2.2 Язык релейных диаграмм

Язык релейных диаграмм (LD), или релейно-контактных схем (РКС) – графический язык, реализующий структуры электрических цепей. Предназначен для программирования промышленных контроллеров (ПЛК). Синтаксис языка удобен для замены логических схем, выполненных на релейной технике. Он ориентирован на инженеров по автоматизации, работающих на промышленных предприятиях. Обеспечивает наглядный интерфейс логики работы контроллера, облегчающий не только задачи собственно программирования и ввода в эксплуатацию, но и быстрый поиск неполадок в подключаемом к контроллеру оборудовании.

В силу своей простоты Ladder Diagram не подойдет для описания громоздких алгоритмов, поскольку он не поддерживает подпрограммы, функции и инкапсуляцию. Однако, благодаря возможности включения в LD функций и функциональных блоков, выполненных на других языках, сфера применения языка практически не ограничена.

Графически LD-диаграмма представлена в виде двух вертикальных шин питания (рисунок 2.1).

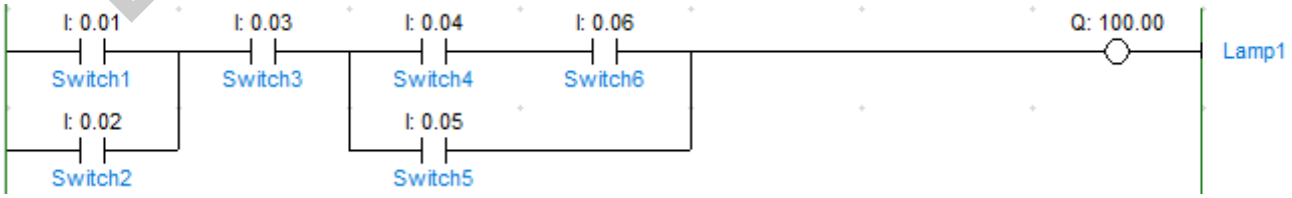
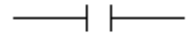


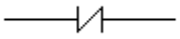
Рисунок 2.1 – Схема LD из одной цепи

Библиотека БГУИР

Между шинами расположены цепи, образованные соединением контактов. Нагрузкой каждой цепи служит реле. Количество контактов в одной цепи произвольно, реле одно. Каждое реле имеет контакты, которые можно использовать в других цепях. При необходимости можно включать несколько реле, однако последовательное включение не допускается.

Основными элементами языка являются контакты, которые можно образно уподобить паре контактов реле, или кнопки. Различаются нормально замкнутые и нормально разомкнутые контактные элементы, которые можно сопоставить с нормально замкнутыми и нормально разомкнутыми кнопками в электрических цепях. Графически базовые элементы языка релейно-контактных схем обозначаются следующим образом:

 – нормально разомкнутый контакт;

 – нормально замкнутый контакт;

 – катушка реле.

В LD каждому контакту ставится в соответствие логическая переменная, определяющая его состояние. Ее имя ставится над контактом и служит его названием. Если контакт замкнут, то переменная имеет значение TRUE, если разомкнут – FALSE. Последовательное соединение контактов или цепей соответствует логической операции И/AND, параллельное – ИЛИ/OR. Нормально замкнутый (инверсный) контакт равнозначен логической операции НЕ. Набор данных способов соединений образуют базис Буля.

Итог логической цепочки копируется в целевую переменную, которая называется катушка (англ. coil). Это слово имеет обобщенный образ исполнительного устройства, поэтому в русскоязычной документации обычно говорят о выходе цепочки, хотя можно встретить и частные значения термина, например, катушка реле. Катушки реле также могут быть инверсными. В таком случае в соответствующую логическую переменную копируется инверсное значение состояния цепи.

Приведенная на рисунке 2.1 схема эквивалентна следующему логическому выражению:

Lamp1 := (Switch1 OR Switch2) AND Switch3 AND ((Switch4 AND Switch6) OR Switch5);

LD-программа выполняется последовательно слева направо и сверху вниз. В каждом рабочем цикле однократно выполняются все цепи, входящие в сеть. Любая переменная в рамках одной цепи всегда имеет одно и то же значе-

Библиотека БГУИР

ние. Если даже реле в цепи изменит переменную, то новое значение поступит на контакты только в следующем цикле. Цепи, расположенные ниже, получат новое значение переменной сразу, а расположенные выше – только в следующем цикле.

Строгий порядок выполнения цепей очень важен. Благодаря жесткому порядку выполнения LD-программы сохраняют устойчивость при наличии обратных связей.

Порядок выполнения цепей диаграммы можно принудительно изменять, используя метки (labels) и переходы (jumps).

Метку можно ставить только в начало цепи. Имена меток подчинены правилам наименования переменных. Для наглядности можно закончить метку двоеточием. Двоеточие не образует новой метки. Так, «M1: » и «M1» это одно и то же. Цепь может иметь только одну метку и один переход. Переход равнозначен выходному реле и выполняется, если выходная переменная имеет значение ИСТИНА. Переход может быть инверсным, в этом случае он выполняется при значении цепи ЛОЖЬ. Используя переход, можно пропустить выполнение части диаграммы. Пропущенные цепи не сбрасываются, то есть не выполняются, а именно остаются в том положении, в котором были ранее. Переход вверх допускается и позволяет создавать циклы. Проверка условий окончания цикла является обязанностью программиста.

Существует специальный переход RETURN, который прекращает выполнение LD-диаграммы. Если RETURN встречается в основной программе, рабочий цикл прерывается. В функциях и функциональных блоках происходит возврат в место вызова. Иными словами, использование перехода RETURN аналогично по смыслу оператору RETURN в текстовых языках.

2.3 Пример программирования на языке LD в CX-Programmer

Для того чтобы начать программирование, в дереве проекта необходимо выбрать пункт «Сегмент1», при этом активизируется окно редактирования программы, представляющее собой поле, ограниченное левой и правой шинами, а также панель инструментов.

При программировании лестничными диаграммами (LD) в виде релейно-контактной схемы программа разделяется на ступени. Каждая ступень представляет собой аналог электрической цепи, по которой может протекать ток. Шина питания находится слева (вертикальная линия).

Библиотека БГУИР

По умолчанию в открытом сегменте уже есть пустой шаблон ступени. При начале работы в ней автоматически создается вторая ступень. Таким образом, нет необходимости специально создавать каждый раз новую ступень сегмента программы. При этом пользователь может принудительно создавать новые ступени выше и ниже текущей ступени с помощью выпадающего меню, вызываемого однократным щелчком правой кнопки мыши (рисунок 2.2).

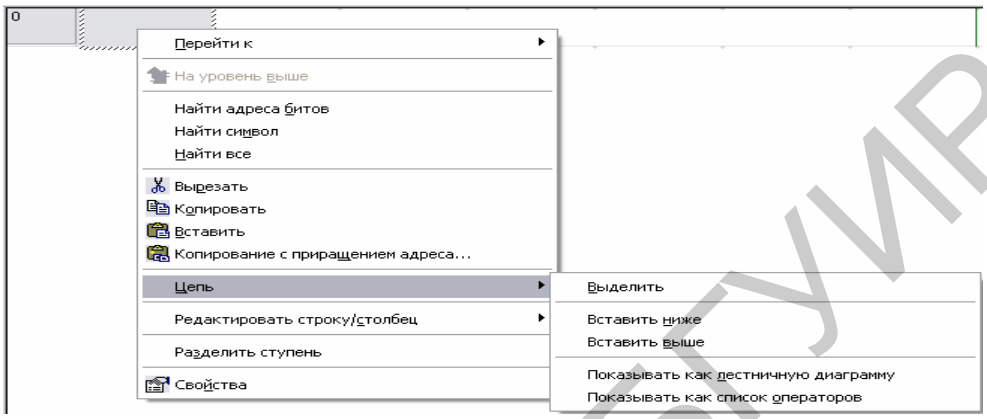


Рисунок. 2.2 – Создание дополнительных ступеней сегмента программы

Далее необходимо программно реализовать простое логическое уравнение:

$$100.00 = (0.00 \cdot \overline{0.01}) + 0.02. \qquad (2.1)$$

Для реализации необходимо выполнить следующие действия:

1. Установить в цепь сегмента программы замыкающий (нормально открытый) контакт. Для этого на панели инструментов выбрать «Создать контакт» и нажать в необходимом положении на поле программы.

2. В открывшемся окне ввести адрес контакта 0.00. После чего дважды нажать «ОК». Аналогичным образом ввести закрытый контакт 0.01. Для создания параллельного контакта 0.02 необходимо выбрать на панели инструментов «Создать вертикальное соединение» и установить вертикальную линию после контакта 0.01.

3. Вставить контакт 0.02 и соединить его с вертикальной линией с помощью инструмента «Создать горизонтальное соединение».

4. На панели инструментов выбрать «Создать катушку» и установить элемент рядом с правой вертикальной шиной. В открывшемся окне указать адрес катушки 100.00 и дважды нажать «ОК».

5. Провести соединительную горизонтальную линию от катушки до остальной части схемы.

Библиотека БГУИР

С левой стороны ступени для удобства пользователя есть индикация правильности использования команд. При ошибках в программировании будет отображаться красная линия. Причем необходимо отметить, что красная линия в процессе программирования говорит лишь о том, что цепь не завершена.

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

В программном пакете CX-Programmer удобно пользоваться символической адресацией, когда любому адресу может быть присвоено удобное для пользователя символическое имя (тэг).

Для открытия редактора символов необходимо в менеджере проекта выбрать пункт «Символы» и дважды щелкнуть по нему левой кнопкой мыши. В открывшемся окне можно создавать новые и редактировать уже существующие тэги.

Далее необходимо создать 4 тэга в соответствии с рисунком 2.3.

Название	Тип данных	Адрес / значение	Расположе...	Испо...	Комментарий
кнопка1	BOOL	0.00		Вход	
кнопка2	BOOL	0.01		Вход	
тумблер	BOOL	0.02		Вход	
лампа	BOOL	100.00		Выход	

Рисунок 2.3 – Использование символьной адресации

Для этого необходимо в столбце «Название» нажать правую кнопку мыши, в выпадающем окне выбрать «Вставить символ» (рисунок 2.4).

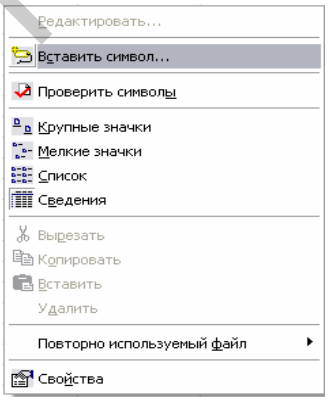


Рисунок 2.4 – Вставка символа через контекстное меню

Затем в открывшемся окне (рисунок 2.5) указать имя тэга, тип данных и адрес, после чего нажать «ОК». Далее аналогично присвоить тэги другим адресам. После этого необходимо закрыть окно «Символы». Внешний вид программы с символьной адресацией примет вид, представленный на рисунке 2.6.

Библиотека БГУИР

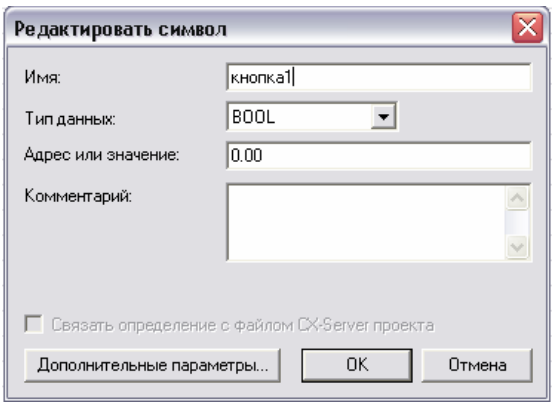


Рисунок 2.5 – Процесс создания символьного имени (тэга)

Рисунок 2.6 – LD-диаграмма с использованием тэгов

2.4 Примеры работы с памятью ПЛК с использованием стандартных битовых команд

На рисунке 2.7 приведен пример использования команд лестничной диаграммы (AND, NOT, OR) на языке LD. Ограничений по количеству использования любой из этих команд или порядка их применения нет, если программа вмещается в отведенную память.

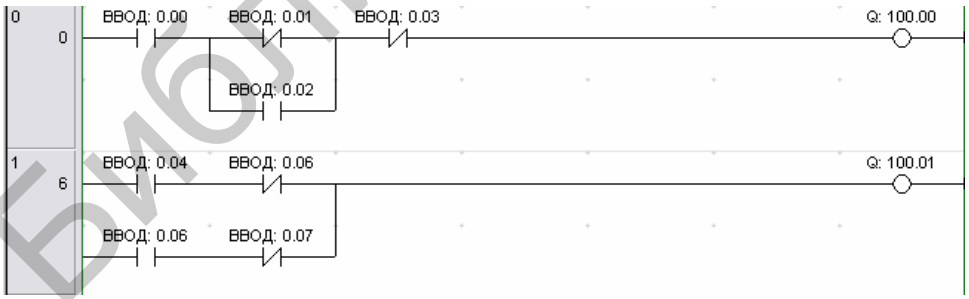


Рисунок 2.7 – Пример программы на языке LD

При вводе команд лестничной диаграммы появляется окно, например, как на рисунке 2.8, в котором необходимо ввести адрес новой переменной и задать соответствующие параметры. Для задания области памяти перед адресом пишется сокращенное название нужной области (например, CIO, D и

Библиотека БГУИР

т. д.). Если перед адресом ничего нет, то по умолчанию воспринимается как адрес в области CIO. Также в этом окне можно выбрать фронт сигнала, по которому будет срабатывать контакт («Различать фронт»).

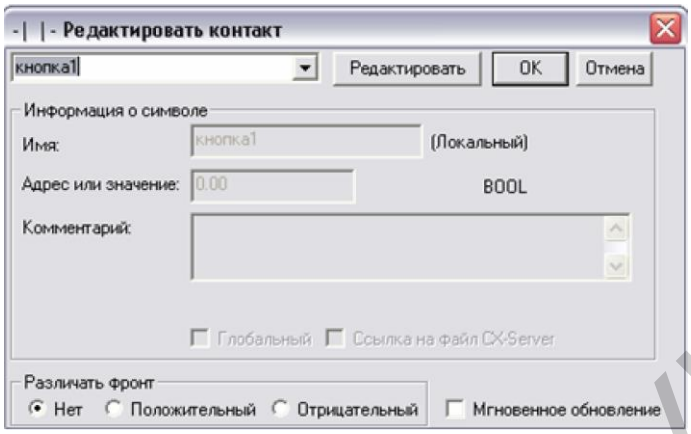


Рисунок 2.8 – Окно выбора параметров команд релейно-контактной схемы

Затем нужно нажать «OK» для подтверждения или «Отмена» – для отмены. Далее появляется окно, в котором можно ввести комментарий к выбранному контакту или выходу. После этого необходимо опять нажать «OK» для подтверждения своих действий или «Отмена» – для их отмены.

2.5 Работы с регистрами ПЛК с использованием команды перемещения MOV

Рассмотреть использование данной команды можно на примере создания следующей программы: входной сигнал должен включить соответствующий ему выход, например, вход 0.00 (адрес CIO0.00) – выход 100.00 (CIO100.00), вход 0.01 (CIO0.01) – выход 100.01 (CIO100.01), вход 0.02 (CIO0.02) – выход 100.02 (CIO100.02) и т. д. То есть необходимо состояние слова CIO0 полностью перенести в слово CIO100. Для этого можно воспользоваться функцией перемещения MOV:

– в новой ступени установить контакт и в открывшемся окне в выпадающем списке выбрать тэг «P_On» – флаг «Всегда включен»;

– выбрать на панели инструментов «Создать команду» и установить команду после ранее установленного контакта. В открывшемся окне в выпадающем списке выбрать команду MOV (необходимо ввести либо имя инструкции, либо соответствующий цифровой код) и нажать кнопку «Подробности». В поле «Операнды» ввести переменные для инструкции. В верхней строке

Библиотека БГУИР

необходимо указать адрес источника – 0 (CIO0), во второй строке сверху – адрес приемника – 100 (CIO100) (рисунок 2.9).

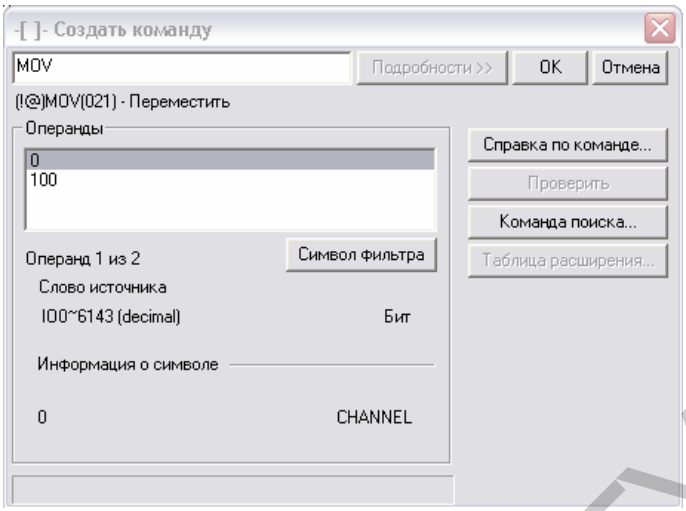


Рисунок 2.9 – Использование команды MOV

Кроме того, при необходимости ввода константы перед числом необходимо установить символ «#». С помощью кнопки «Команда поиска» также можно ввести необходимую инструкцию. Для этого надо найти ее в списке и нажать кнопку «ОК» или «Отмена». Здесь также можно ввести необходимый комментарий. Краткую справку по инструкции можно получить, нажав кнопку «Справка по команде».

После ввода всех необходимых параметров нужно нажать кнопку «ОК». После этого программа примет вид, представленный на рисунке 2.10.

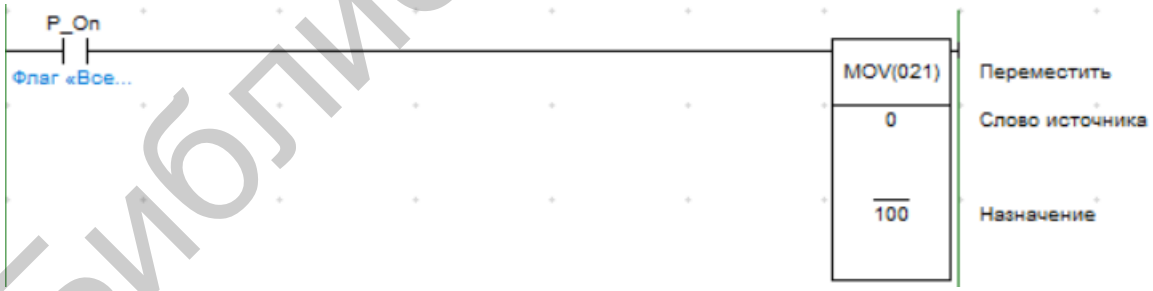


Рисунок 2.10 – Внешний вид команды MOV

2.6 Работа с системными таймерами и счетчиками

Существенной особенностью многих программ для ПЛК является подсчет чего-либо. Например, от ПЛК можно потребовать подсчитать число предметов в партии или зарегистрировать, сколько раз произойдет некоторое событие, а для

Библиотека БГУИР

больших электродвигателей – зафиксировать количество включений. Поэтому неудивительно, что все ПЛК содержат те или иные виды счетчиков.

Работа счетчика характеризуется двумя числами. Первое из них – это собственно отсчет (часто называемый накопленным значением), который увеличивается или уменьшается, когда происходит переход сигнала от низкого логического уровня на суммирующем входе счетчика. Накопленное значение может быть сброшено до исходного состояния путем подачи логической 1 на вход очистки счетчика. Как и время, заданное таймером, подсчитанное число можно прочитать и использовать в других фрагментах программы.

Следующее число – это уставка, которую можно рассматривать как заданную цель для счетчика. Если подсчитанное значение достигает уставки, подсчет заканчивается и появляется сигнал об окончании работы счетчика. Значение уставки может быть изменено программным способом, например: если речь идет о последовательности предметов, оператор может изменить количество предметов в партии, введя это значение с клавиатуры или при помощи программируемого терминала.

Счетчики у ПЛК Omron делятся на два вида – считающие в BCD-формате и в десятичном. Способ применения одинаковый. Но у первых необходимо вносить данные, соответственно, в BCD. Это может оказаться полезным при работе с часами реального времени, т. к. именно в таком формате хранится текущая дата/время. Минусы при его использовании: максимальное значение в одном слове – 9999 и большее время на обработку инструкций.

Инициирование счетчика происходит путем его сброса. Это связано с тем, что значение счетчика сохраняется при отключении питания (некоторое время без батарейки). Считает он в обратную сторону, так что, если просмотреть значение регистра C_0 (текущее значение, PresentValue, PV), то увидим 10, а после переустановки W0.00 – 9.

Также существует реверсивный счетчик. Его отличия от обычного:

– вход не один, а два: по одному текущее значение прибавляется (прямой счет), по второму – убавляется (обратный счет);

– при сбросе текущее значение счетчика становится равным 0.

Флаг окончания счета устанавливается только в двух случаях:

– при использовании входа обратного счета был получен 0;

– при использовании входа прямого счета было получено установленное значение (Set Value).

Библиотека БГУИР

Аналогично счетчикам все таймеры разделяются на считающие в BCD-формате (двоично-десятичный) и BIN-формате. Дополнительно таймеры разделяются по единицам измерения: 100 мс, 10 мс и 1 мс. 32-битную модификацию имеют только 100-мс таймеры. Считают таймеры с уставки до нуля. Тип таймера определяется при запуске его определенной инструкцией.

Алгоритм таймера является одним из самых распространенных и используемых алгоритмов. Таймеры могут использоваться в программах управления запуском и остановкой двигателей, турбин, конвейеров и т. д. Существующие таймеры из стандартных библиотек, без дополнительных надстроек, не могут управлять запуском и остановкой какого-либо двигателя в силу отсутствия тех или иных функций. В каждой из рассматриваемых сред есть свой алгоритм таймера, в некоторых даже не один.

Таймеры запускаются подачей на них сигнала. После пропадания сигналов 0.00, 0.01 текущие значения T0, T1 (регистры) таймеров становятся равными уставке (даже если счет не закончен) и снимаются T0, T1 (биты) флаги окончания счета.

Также существует накопительный счетчик (100 мс, 16-бит, BCD/BIN). Он отличается тем, что считает от нуля до уставки, после снятия сигнала счета не сбрасывает свое текущее значение и после восстановления сигнала продолжает счет с того значения, на котором остановился. Сброс таймера осуществляется дополнительным входом сброса.

В качестве примера использования данных команд можно рассмотреть решение следующей задачи: при включении входа ПЛК 0.00 (адрес CIO0.00) на 5 с и более включается выход 100.00 (CIO100.00). При наличии 10 включений входа 0.00 (адрес CIO0.00) включается выход 100.01 (CIO100.01). Сброс счета происходит при включении входа 0.01 (адрес CIO0.01). На рисунке 2.11 представлены временные диаграммы.

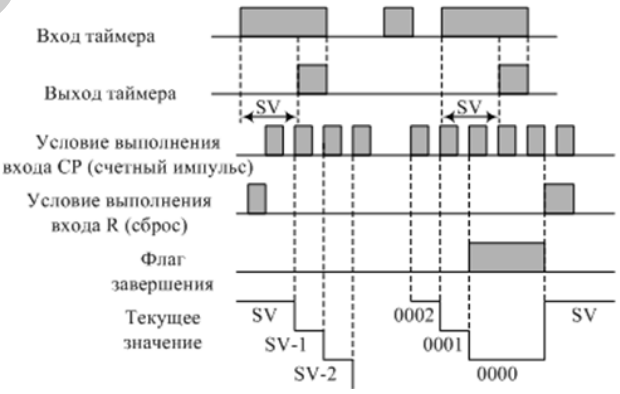


Рисунок 2.11 – Временные диаграммы

Библиотека БГУИР

Программа разрабатывается в определенной последовательности:

– в первой ступени программы установить команду таймера TIM, в поле операндов указать номер таймера – 0000 и уставку – #50 (5 с);

– для включения таймера в работу на вход таймера установить контакт 0.00;

– во второй ступени выход таймера T0000 подключить к катушке 100.00;

– в третьей ступени программы установить команду счетчика CNT, в поле операндов указать номер счетчика – 0000 и уставку – #10 (10 включений);

– на счетный вход счетчика подключить контакт 0.00, на сбросовый – контакт 0.01;

– в четвертой ступени выход счетчика C0000 подключить к катушке 100.01.

На рисунке 2.12 представлена разработанная на языке LD программа.

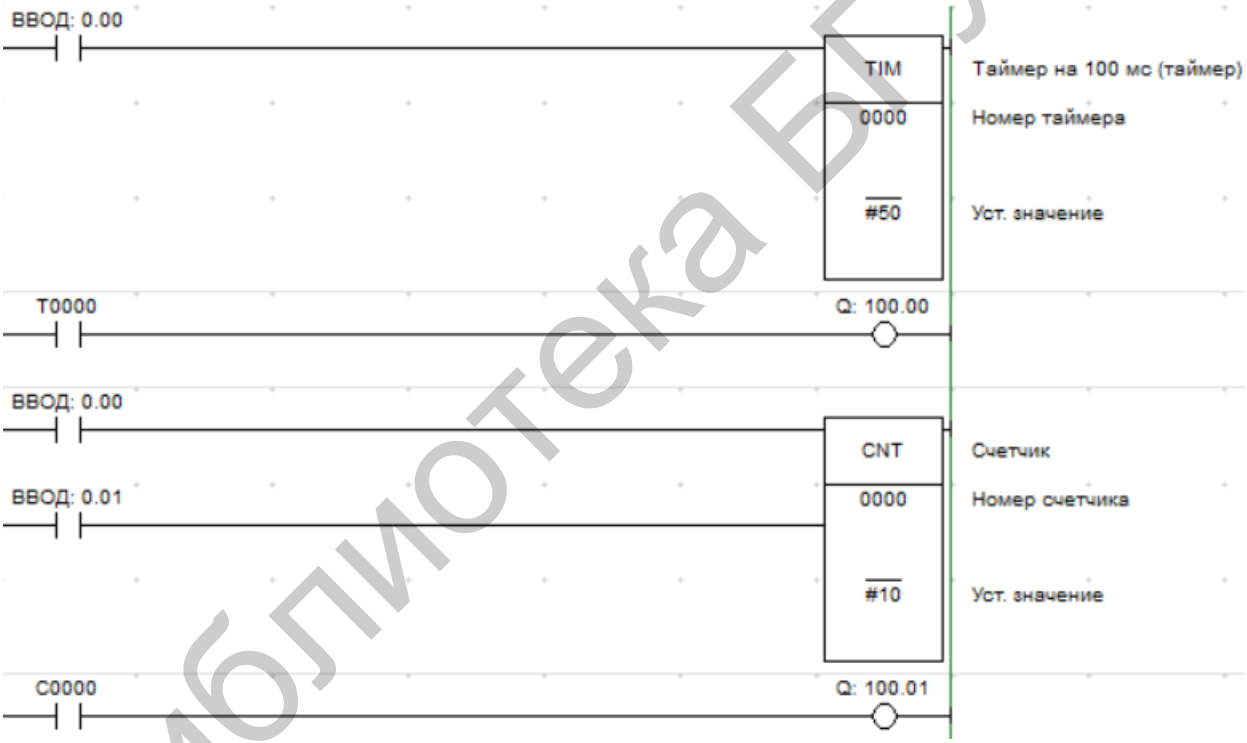


Рисунок 2.12 – Инициализация таймера и счетчика

Область операндов таймера и счетчика содержит два значения: номер таймера или счетчика N (может задаваться только в числовом формате) и уставка или заданное значение SV (может задаваться как в числовом формате, так и косвенно, словом из области данных CIO, W, H, A, T, C, D). При этом каждый номер таймера или счетчика в диапазоне от 0 до 4095 можно использовать в качестве определителя только для одного таймера или счетчика.

Библиотека БГУИР

3 ОСНОВЫ РАБОТЫ С ЧАСТОТНЫМИ ПРЕОБРАЗОВАТЕЛЯМИ

3.1 Применение и принципы построения частотных преобразователей

В настоящее время в промышленности и на транспорте отчетливо просматривается тенденция увеличения относительной доли асинхронного регулируемого электропривода переменного тока сравнительно с регулируемым электроприводом постоянного тока. Замена электропривода постоянного тока асинхронным закономерна, она обусловлена прежде всего более высокими технико-конструкционными показателями асинхронных двигателей, такими, как масса, надежность, перегрузочная способность. При всем этом асинхронный привод более сложен в вопросах реализации регулирования момента и скорости.

Современные системы векторного управления позволяют просто и эффективно управлять таким сложным объектом, как асинхронный двигатель с короткозамкнутым ротором, что в свою очередь позволяет существенно расширить область его применения, почти полностью вытесняя из автоматизированных управляемых приводов двигатели постоянного тока. Это связано в первую очередь с развитием силовой электроники, позволяющей создавать надежные и относительно дешевые преобразователи, а также с развитием быстродействующей микроэлектроники, способной реализовать алгоритмы управления практически любой сложности.

В настоящее время все большее распространение получают системы электропривода на базе асинхронного двигателя (АД) с короткозамкнутым ротором. Известно, что главным преимуществом АД в регулируемом электроприводе является простота конструкции. Здесь нет никаких скользящих контактов, а единственный узел, требующий периодического контроля, – это подшипники.

Несмотря на видимую сложность процесса электромеханического преобразования энергии в АД, использование принципа векторного управления позволяет раздельно формировать потокосцепление ротора и моментообразующую составляющую вектора токов статора, что делает управление АД аналогичным управлению двигателем постоянного тока независимого возбуждения. Для реализации принципа векторного управления необходима информация о векторе потокосцепления ротора. В общепромышленных электроприводах для ее получения используются наблюдатели, точность которых напрямую зависит от точности определения параметров схемы замещения АД. Кроме того, часто для получения информации о текущем значении частоты вращения двигателя вме-

Библиотека БГУИР

сто датчиков используются специальные алгоритмы идентификации, точность которых также зависит от точности определения параметров схемы замещения.

В последнее время в связи с развитием элементной базы и появлением программных средств, которые облегчают анализ и синтез сложных систем, значительно распространился интерес к системам автоматического управления с наблюдателями состояния (НС). Необходимость использования НС обусловлена их способностью оценивать значение координат, которые невозможно или очень тяжело измерить непосредственно (с помощью соответствующих датчиков). Наличие дополнительной информации о системе разрешает улучшить качество регулирования и расширяет возможности для автоматизации технологических процессов.

Основное назначение преобразователя частоты (ПЧ) – это получение переменного по частоте и амплитуде напряжения для управления трехфазным асинхронным двигателем. Большинство современных преобразователей частоты (ПЧ) от долей до сотен киловатт построены одинаково: сеть переменного тока – неуправляемый выпрямитель – шины постоянного тока – конденсатор LC-фильтра – автономный инвертор напряжения с широтно-импульсной модуляцией (ШИМ) – асинхронный двигатель АД, к которому приложено переменное трехфазное напряжение с регулируемой частотой f и амплитудой U (управление инвертором осуществляется блоком управления). Используя широтно-импульсную модуляцию, можно формировать любые нужные формы кривой тока, учитывая изменяющиеся в процессе работы параметры нагрузки. В современных качественных преобразователях частоты ШИМ позволяет при любой требуемой выходной частоте преобразователя изменять нужным образом амплитуду напряжения, управляя магнитным потоком двигателя, и формировать при любой нагрузке на валу близкую к синусоидальной форму тока двигателя.

Полностью реализовать широкие возможности ШИМ удалось лишь в последние 10–15 лет с появлением на рынке совершенных ключей, в частности, транзисторных модулей IGBT напряжением до 1200 В, током до 600 А и частотой коммутации до 30 кГц, а также средств управления ими.

Блок-схема силовой части преобразователя с промежуточным звеном постоянного тока (так называемый U-инвертор) приведена на рисунке 3.1.

Выпрямитель состоит из неуправляемой одно- или трехфазной мостовой схемы и преобразует переменное напряжение сети в постоянное. Далее оно сглаживается в промежуточном контуре конденсатором.

Библиотека БГУИР

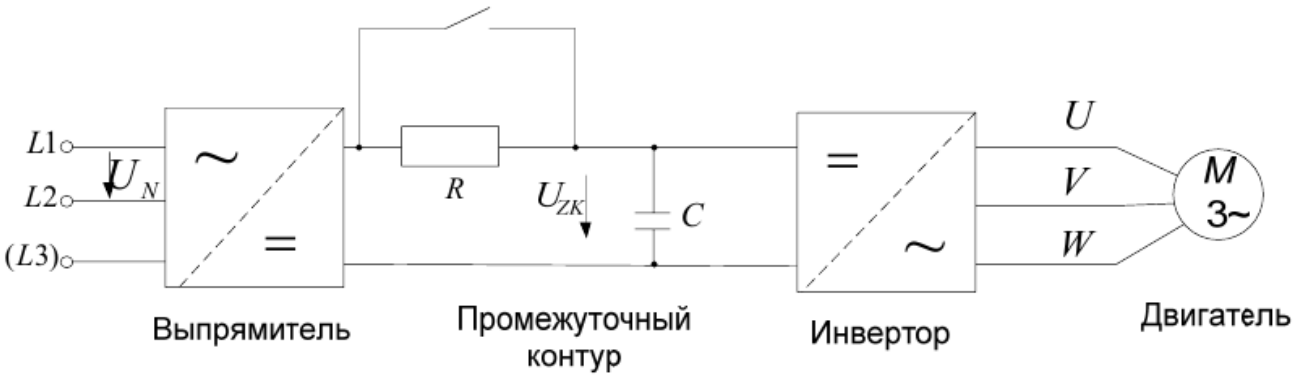


Рисунок 3.1. – Структурная электрическая схема преобразователя частоты

Во время зарядки конденсатора промежуточного контура протекает очень большой кратковременный ток. Это может вывести из строя входной предохранитель или даже выпрямитель. Ток зарядки должен быть ограничен допустимой величиной. Это достигается включением балластного резистора R последовательно с конденсатором, который активизируется только при включении преобразователя. После зарядки конденсатора резистор выключается, например, контактным реле.

Большая емкость конденсатора требуется для сглаживания напряжения промежуточного звена. После выключения инвертора из сети конденсатор сохраняет высокое напряжение в течение определенного времени. Это отображается зарядным светодиодом.

На выходе устанавливается инвертор (рисунок 3.2). При инвертировании постоянного тока в переменный в ПЧ используются транзисторы, которые работают в переключающем режиме.

Ток через обмотку двигателя протекает тогда, когда по меньшей мере один из верхних (T1, T3 и T5) и один из нижних транзисторов (T4, T6 и T2) включены (рисунок 3.3).

Посредством циклического переключения силовых ключей ток меняется в трех выходных фазах, которые постоянно сдвинуты на 120° относительно друг друга. Получается симметричная трехфазная система, частота которой зависит от длительности цикла срабатывания выходных ключей инвертора.

Амплитуда определяется величиной отношения времени включения ко времени выключения транзисторов. При широтно-импульсной модуляции синусоидального сигнала это отношение мало в начале и конце полуволны и велико в середине.

Библиотека БГУИР

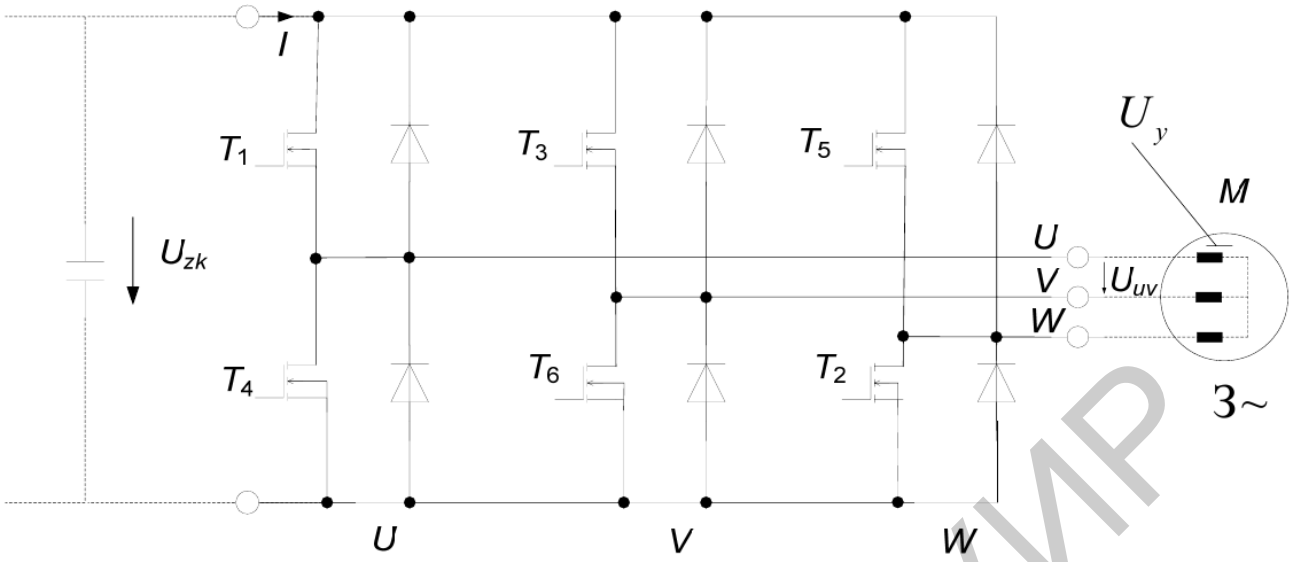


Рисунок 3.2. – Электрическая принципиальная схема трехфазного инвертора

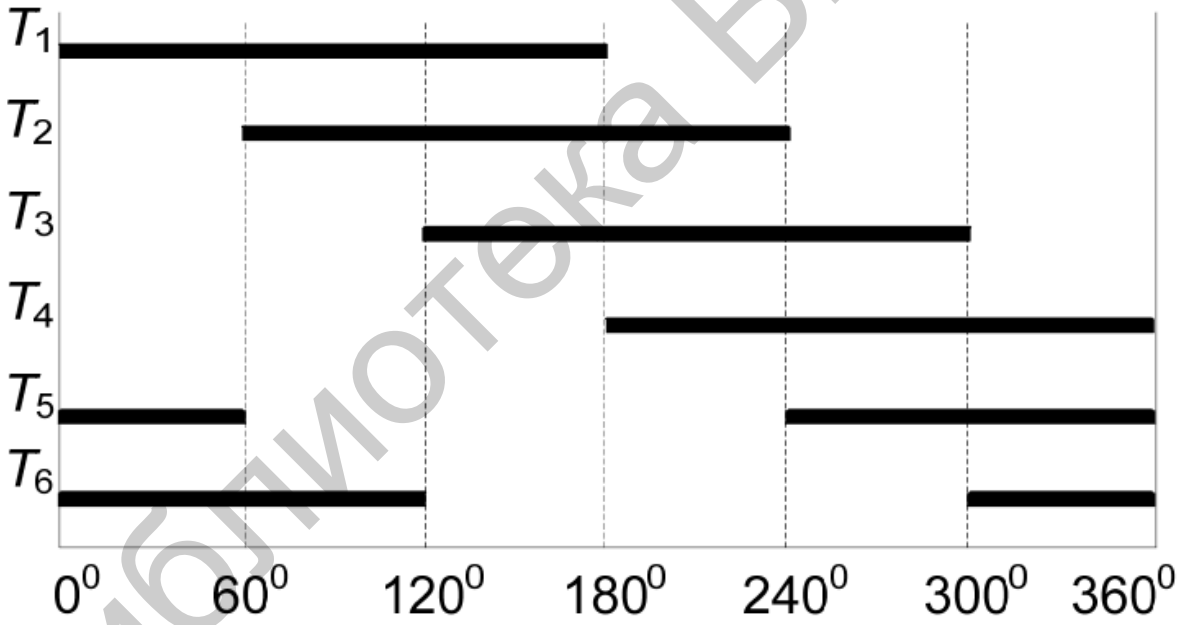


Рисунок 3.3. – Порядок срабатывания выходных ключей инвертора

Библиотека БГУИР

3.2 Основные функции преобразователей частоты Omron MX2

Линейка преобразователей частоты серии MX2 [5] компании Omron насчитывает более 12 моделей, охватывающих двигатели мощностью от 0,1 до 15 кВт. Каждая модель имеет исполнение на входное напряжение ~240 или ~400 В.

Внешний вид ПЧ MX2 представлен на рисунке 3.4.

Рисунок 3.4. – Внешний вид и устройство лицевой панели ПЧ MX2

Основные характеристики преобразователей частоты серии MX2:

– ПЧ 200 и 400 В, на мощность от 0,1 до 15 кВт, с характеристиками для двух режимов нагрузки;

– встроенная функция простого программирования (EzSQ);

– порт RS485 MODBUS RTU во всех моделях по умолчанию, также доступны другие сетевые опции;

– новая функция ограничения тока;

– 16 программируемых ступеней скорости (16 с помощью 4 входов или 8 скоростей побитово с помощью 7 входов);

Библиотека БГУИР

– ПИД-регулятор автоматически корректирует скорость двигателя для поддержания заданного значения переменной процесса;

– защита с помощью пароля от непреднамеренного изменения параметров;

– управление синхронным двигателем (двигателем с постоянными магнитами);

– пятистрочный ЖК-дисплей с возможностью чтения и записи (функции копирования) и часами реального времени для протоколирования аварийных отключений.

Конструкция ПЧ MX2 устраняет многие ставшие привычными компромиссы между скоростью, крутящим моментом и коэффициентом полезного действия. Основные рабочие характеристики:

– высокий пусковой момент: 200 % при 0,5 Гц;

– продолжительная работа с крутящим моментом 100 % в диапазоне скоростей 1:10 (6/60 Гц/5/50 Гц) без ухудшения показателей работы двигателя;

– возможность включения/выключения охлаждающего вентилятора для продления его срока службы.

Модельный ряд преобразователей частоты Omron MX2 включает в себя несколько исполнений:

– однофазные модели класса 200 В;

– трехфазные модели класса 200 В;

– трехфазные модели класса 400 В.

Схема подключения ПЧ Omron MX2 представлена на рисунке 3.5.

Библиотека БГУИР

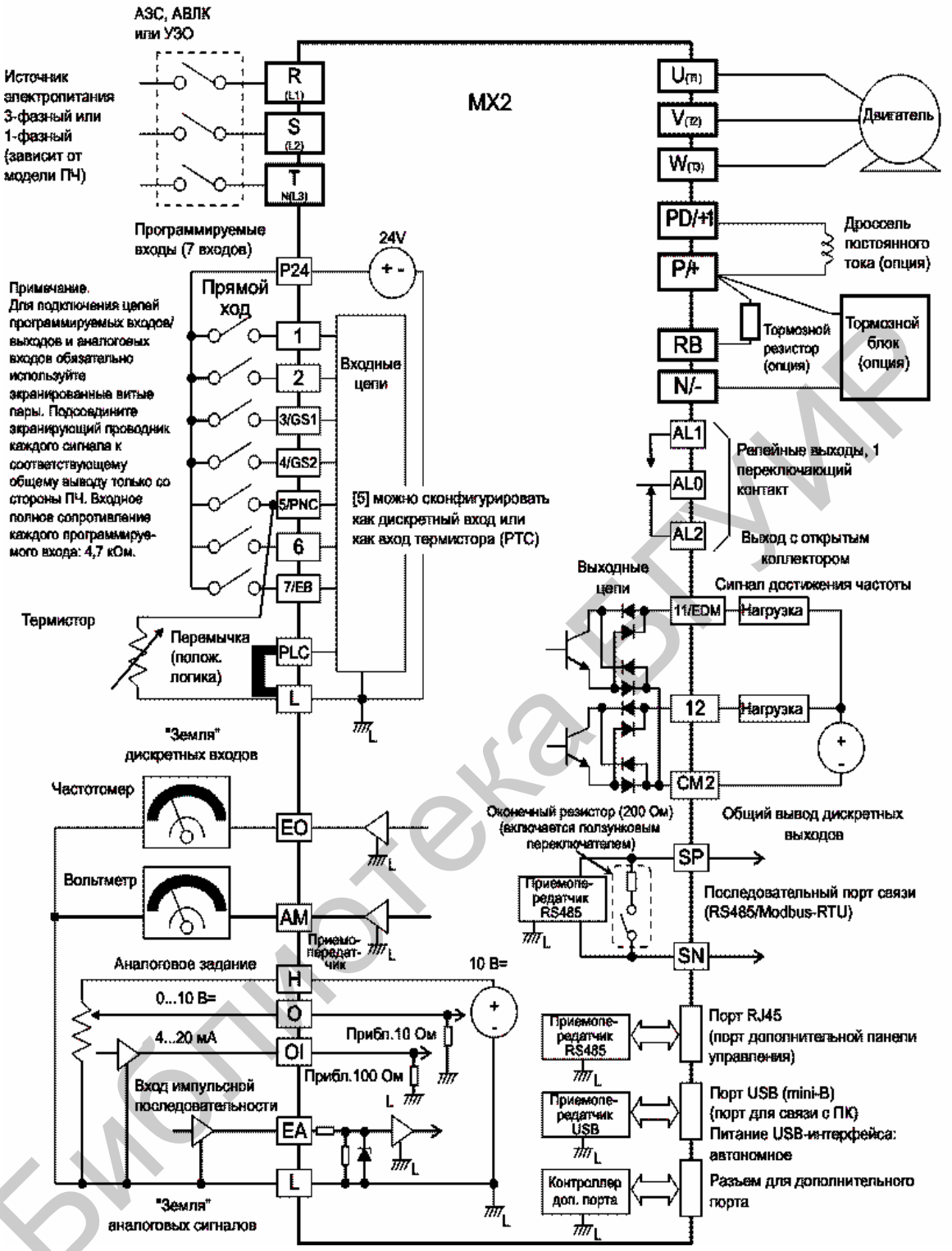


Рисунок 3.5. – Схема подключения преобразователя частоты Omron MX2

Клавишная панель, расположенная на лицевой панели преобразователя частоты серии MX2, содержит все необходимые органы управления и индикации для программирования параметров и контроля их значений [6]. Располо-

Библиотека БГУИР

жение органов управления и индикации на клавишной панели показано на рисунке 3.6.

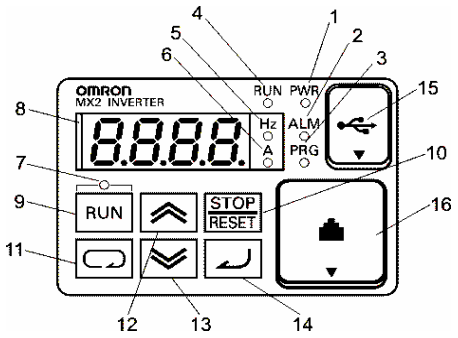


Рисунок 3.6 – Внешний вид клавишной панели

Назначение каждого индикатора и клавиши описано в таблице 3.1.

Таблица 3.1 – Описание полей класса

Элемент	Назначение
Светодиод «POWER» (Питание)	Включен (светится зеленым светом), когда на ПЧ подано питание
Светодиод «ALARM» (Ошибка)	Включен (светится красным светом), когда ПЧ находится в состоянии аварийного выключения
Светодиод «Program» (Программа)	Включен (светится зеленым светом), когда индикатор отображает изменяемый параметр. Мигает в случае ввода ошибочного значения
Светодиод «RUN» (Ход)	Включен (светится зеленым светом), когда ПЧ управляет двигателем
Светодиод контроля (Гц)	Включен (светится зеленым светом)при индикации данных, связанных с частотой
Светодиод контроля (А)	Включен (светится зеленым светом)при индикации данных, связанных с током
Светодиод команды «Ход»	Включен (светится зеленым светом), когда команда «Ход» выдается с панели управления (клавиша «Ход» действует)
Семисегментный дисплей	Отображает параметры, контрольные значения и т. п.
Клавиша «Run» (Ход)	По нажатии этой клавиши ПЧ включает двигатель

Входы 1–7 – это идентичные друг другу программируемые входы общего назначения. Электрическая схема входных цепей допускает использование внутреннего (изолированного) источника питания +24 ВПЧ или внешнего источника питания. Также возможен выбор между отрицательной или положительной логикой управления входами.

Библиотека БГУИР

В случае отрицательной логики активному состоянию входа (логическая 1) соответствует низкий потенциал на входе, в то время как высокий потенциал соответствует неактивному состоянию (логический 0). В случае положительной логики вход переводится в активное состояние (логическая 1) высоким потенциалом.

Положительную и отрицательную логику управления также можно интерпретировать с позиции направления протекания тока входного сигнала.

Если для входа выбран тип контакта «НО», то в неактивном состоянии (логический 0) ток в цепи не течет, а в активном состоянии (логическая 1) в цепи с отрицательной логикой ток течет в направлении от входа к источнику сигнала, а в цепи с положительной логикой – от источника сигнала в направлении входа.

Выбор типа контакта «НЗ» не меняет направление протекания токов, но меняет логику управления на противоположную: в активном состоянии ток в цепи не течет, а в неактивном – течет.

ПЧ MX2 имеет несколько режимов работы [7]:

- ход;
- стоп;
- программа;
- мониторинг.

Режимы «Ход» и «Программа» – это два независимых, но не противоположных друг другу режима, поэтому светодиоды «RUN» и «PRG» не несут полной информации о текущем режиме работы ПЧ. Из приведенной диаграммы состояний (см. рисунок 3.7) видно, что режим «Ход» противоположен режиму «Стоп», а режим «Программа» противоположен режиму «Мониторинг» (контроль). Благодаря тому что режимы «Ход» и «Программа» не являются взаимоисключающими, оператор может изменять некоторые параметры преобразователя частоты, не прекращая работу оборудования.

Возникновение ошибки во время работы вызывает аварийное отключение выхода преобразователя частоты, как показано на рисунке 3.7.

Аварийное событие, такое как перегрузка двигателя, приводит к тому, что преобразователь частоты выходит из режима «Ход» и снимает питание с электродвигателя. В режиме аварийного отключения любая попытка запустить двигатель игнорируется. Вы должны сбросить ошибку нажатием клавиши «Stop/Reset».

Программа управления двигателем в ПЧ MX2 может использовать один из двух алгоритмов формирования синусоидального тока методом

Библиотека БГУИР

ШИМ (рисунок 3.8). Оба алгоритма формируют частоту на выходе преобразователя одинаковым образом. Выбранный алгоритм управления также является основой для настройки других параметров.

Рисунок 3.7 – Диаграмма состояний преобразователя частоты

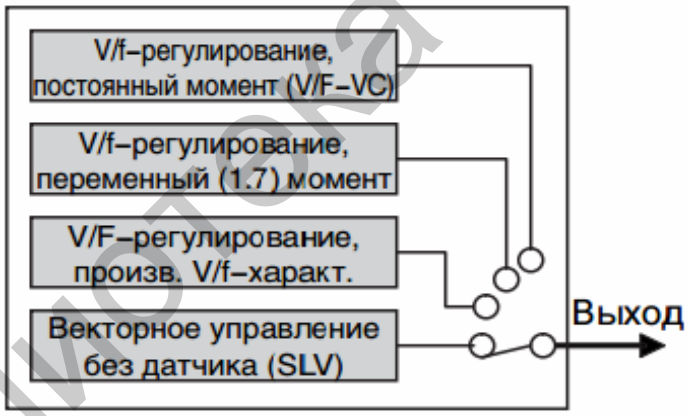


Рисунок 3.8 – Алгоритмы управления ПЧ

3.3 Конфигурирование частотного преобразователя Omron MX2 в ручном режиме

Конфигурирование ПЧ MX2 [8] заключается в задании требуемых значений параметров. Совокупность заданных значений параметров привода называется набором параметров. Набор параметров задает определенный режим работы ПЧ. В большинстве случаев применения потребуется дополнительно настроить всего несколько специальных параметров.

Библиотека БГУИР

ПЧ MX2 имеет достаточно большой перечень программируемых параметров, которые для удобства пользователя разбиты по группам:

- группа «F» – основные параметры профиля;
- группа «A» – стандартные функции;
- группа «B» – функции точной настройки;
- группа «C» – функции программируемых входов и выходов;
- группа «H» – константы двигателя;
- группа «P» – функции карт расширения.

Набор базовых параметров с доступными в конкретном случае диапазонами настройки представлен в таблице 3.2.

Таблица 3.2 – Диапазоны настройки параметров ПЧ

Код параметра / Имя функции	Диапазон настройки или контроля значений
A001 / Выбор способа ввода задания частоты	00: цифровая панель (ручка регулировки) (возможно при использовании 3G3AX-OP01) / 01: клемма / 02: цифровая панель (F001) / 03: интерфейс Modbus / 04: дополнительная плата / 06: импульсная последовательность / 07: EzSO / 10: результат вычисления
A002 / Выбор способа подачи команды «Ход»	01: клемма / 02: цифровая панель / 03: интерфейс Modbus / 04: дополнительная плата
A003 / Основная частота	От 30 до максимальной частоты
A004 / Максимальная частота	От основной частоты до 400 (1000)
A005 / Выбор входов O/OI	00: переключение между входом O и входом OI / 02: переключение между входом O и потенциометром клавишной панели / 03: переключение между входом OI и потенциометром клавишной панели
A019 / Выбор ступенчатого переключения скорости	00: двоичный (выбор 16 ступеней с помощью 4 входов) / 01: битовый (выбор 8 ступеней с помощью 7 входов)
A020 / Задание ступенчатого переключения скорости 0	0,0 / от пусковой частоты до максимальной
A021...A035 / Задание ступенчатого переключения скорости 1, 2, ...15	0,0 / от пусковой частоты до максимальной
A038 / Частота толчкового хода	От пусковой частоты до 9,99
A039 / Выбор способа остановки для толчкового хода	00: остановка на выбеге в толчковом режиме, выключение во время работы / 01: торможение до остановки в толчковом режиме, выключение во время работы / 02: торможение постоянным током в толчковом режиме, выключение во время работы / 03: остановка на выбеге в толчковом режиме, включение во время работы / 04: торможение до остановки в толчковом режиме, включение во время работы / 05: торможение постоянным током в толчковом режиме, включение во время работы

Библиотека БГУИР

В зависимости от уровня доступа часть параметров доступна для просмотра и редактирования на стандартном уровне (параметр B031=0), а часть – на высоком уровне доступа (параметр B031=10).

Все параметры ПЧ можно сбросить к их исходным заводским (принимаемым по умолчанию) значениям, соответствующим зоне эксплуатации. Для инициализации ПЧ необходимо выполнить следующие действия:

– выбрать режим инициализации с помощью параметра b084 (таблица 3.3);

– если b084=02, 03 или 04, выбрать инициализируемые данные в b094;

– если b084=02, 03 или 04, указать код страны в b085;

– задать b180=01.

На дисплее в течение нескольких секунд должны отобразиться приведенные на рисунке 3.9 коды, после чего инициализация завершается, и на дисплее отображается код d001.

Таблица 3.3 – Группа параметров «b»

Код функции	Название	Описание
b084	Режим инициализации (параметров или журнала аварийных отключений)	Выбрать инициализируемые данные: 00: Инициализация выключена / 01: Очистка журнала аварийных отключений / 02: Инициализация всех параметров / 03: Очистка журнала аварийных отключений и инициализация всех параметров / 04: Очистка журнала аварийных отключений, инициализация всех параметров и программы EzSQ
b094	Выбор инициализируемых данных	Выбрать инициализируемые параметры: 00: Все параметры / 01: Все параметры, кроме входных/выходных клемм и интерфейса связи / 02: Только параметры, зарегистрированные в Uxxx / 03: Все параметры, кроме параметров, зарегистрированных в Uxxx, и параметра b037
b085	Выбор зоны для начальных данных	Выбрать зону применения преобразователя частоты для инициализации данных: 00: (Япония/США) / 01: (Европа)
b180	Запуск инициализации	Служит для выполнении инициализации в соответствии с введенными значениями параметров b084, b085 и b094: 00: Инициализация выключена / 01: Выполнить инициализацию

Библиотека БГУИР

Рисунок 3.9 – Процесс инициализации

ПЧ MX2 имеет встроенную функцию автонастройки, которая позволяет добиться оптимального качества управления двигателем за счет автоматического измерения констант двигателя. Автонастройка эффективна только для векторного управления без датчика обратной связи.

В ПЧ MX2 реализовано два режима автонастройки:

1. Автонастройка при остановленном двигателе (H001=01). В процессе автонастройки двигатель не вращается. Этот режим используется в том случае, если вращение двигателя при автонастройке нежелательно. При этом постоянные двигателя I0 (ток холостого хода) и J (момент инерции) не измеряются. (Значение I0 можно посмотреть при частоте 50 Гц в режиме V/f-регулирования).

2. Автонастройка с вращением двигателя (H001=02). Во время автонастройки для вращения двигателя применяется специальный алгоритм управления. Однако при автонастройке не обеспечивается достаточный вращающий момент, что может быть проблемой для грузоподъемных систем.

На рисунке 3.10 представлена процедура автонастройки, в таблице 3.4 – функции, которые необходимо использовать при автонастройке ПЧ. При выборе автонастройки без вращения шаги 4 и 5 пропускаются.

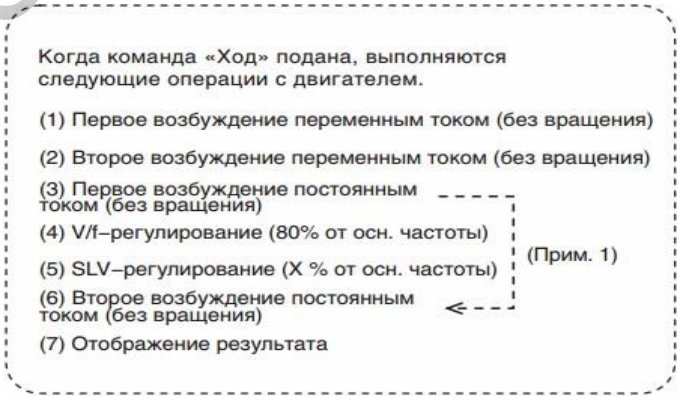


Рисунок 3.10 – Процесс автонастройки

Библиотека БГУИР

Таблица 3.4 – Функции автонастройки преобразователя частоты

Код функции	Название	Описание
H001	Выбор типа автонастройки	00: Отключена / 01: Включена, без вращения двигателя / 02: Включена, с вращением двигателя
H002	Выбор модели двигателя	00: Стандартный двигатель Hitachi / 02: Использовать данные автонастройки
H003	Мощность двигателя	0,1 / 0,2 / 0,4 / 0,75 / 1,5 / 2,2 / 3,7 / 5,5 / 7,5 / 11 / 15 / 18,5
H004	Количество полюсов двигателя	2 / 4 / 6 / 8 / 10
A003	Базовая частота	Установить частоту от 30 Гц до максимальной частоты
A082	AVR: Выбор номинального напряжения	Выбрать наиболее близкое значение: 200В класс: 200 / 215 / 220 / 230 / 240

Для проведения процедуры автонастройки необходимо выполнить следующие действия:

– предварительно обязательно отключить торможение постоянным током, простое позиционирование, отключить вход задания момента, в противном случае константы двигателя будут измерены неверно;

– задать мощность и число полюсов двигателя: 0,2 и 4;

– выбрать тип автонастройки;

– установить базовую частоту: 50 Гц;

– выбрать номинальное напряжение: 380 В;

– выбрать источник команды «Ход»: управление с панели;

– для начала автонастройки нажать «RUN», по окончании автонастройки должно появиться одно из двух сообщений: автонастройка успешно проведена или ошибка автонастройки.

3.4 Конфигурирование частотного преобразователя Omron MX2 в дистанционном режиме

Конфигурирование ПЧ MX2 может производиться также на ПК с помощью программы CX-Drive [9] в операционных системах Windows через USB-порт с использованием кабеля USB-A – USB-B-mini.

Минимальные требования к ПК:

– процессор, необходимый для работы соответствующей ОС;

– свободное место на жестком диске не менее 500 Мб;

Библиотека БГУИР

– дисплей с параметрами XGA (1024x768, High Color 16);
– клавиатура и мышь.

Программное обеспечение CX-Drive имеет следующие функциональные возможности:

– выбор типа привода;
– установка параметров привода;
– установка параметров связи;
– мониторинг состояния привода в режиме онлайн;
– отображение трендов в режиме онлайн;
– осуществление дистанционного пробного пуска;
– осуществление инициализации привода.

Запуск программы CX-Drive осуществляется через соответствующий ярлык на рабочем столе. Стартовое окно изображено на рисунке 3.11.

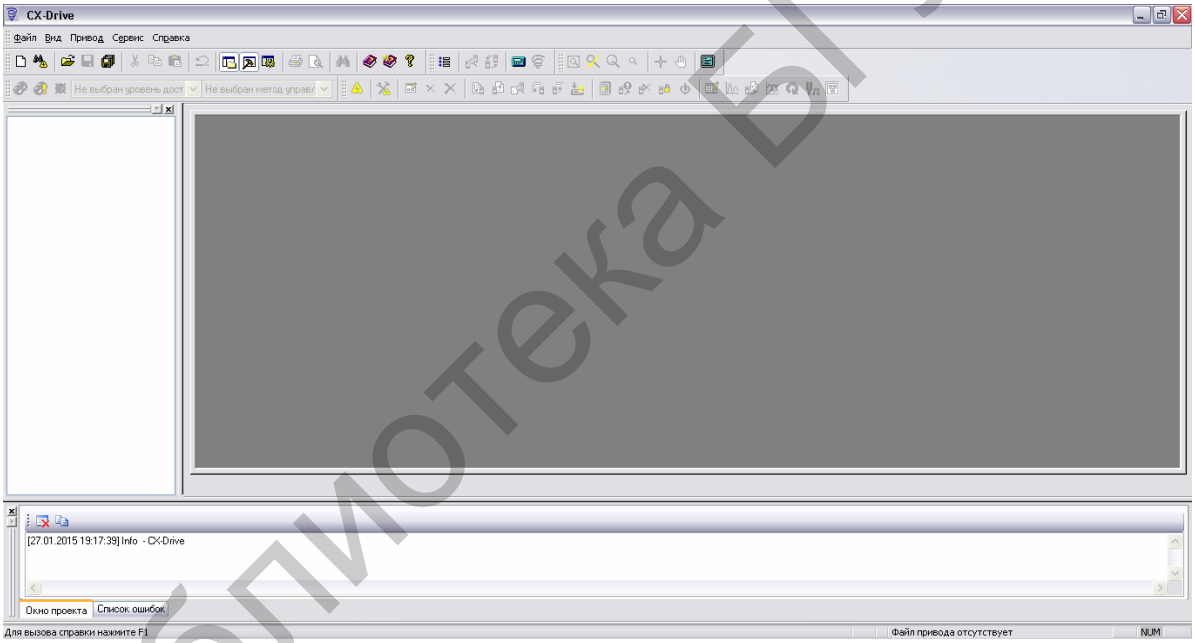


Рисунок 3.11 – Стартовое окно программы CX-Drive

Для того чтобы создать новый файл объекта, необходимо выбрать в главном окне программы в выпадающем меню «Файл» пункт «Создать». При этом должно появиться окно (рисунок 3.12), в котором необходимо задать нужное имя ПЧ (поле «Имя привода»), тип ПЧ – инвертор 3G3MX2 (поле «Тип привода»), а также тип связи с ПЧ – Прямое (USB) (поле «Тип соединения»).

Библиотека БГУИР

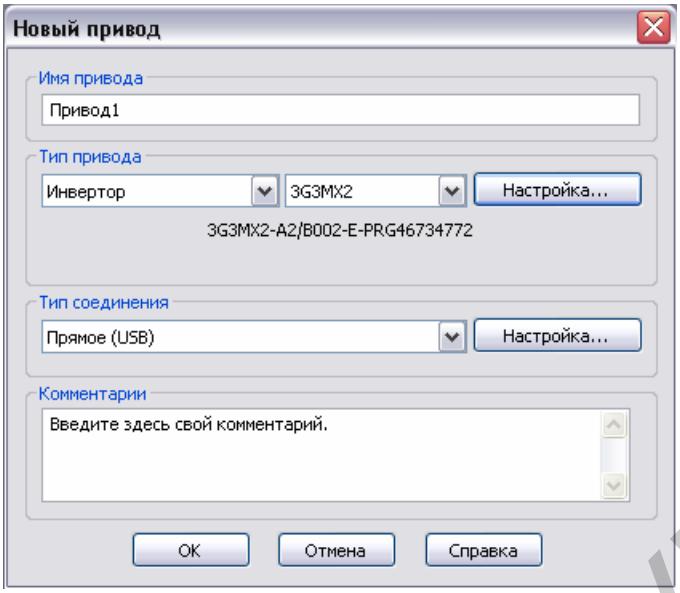


Рисунок 3.12 – Создание нового проекта

При необходимости можно ввести комментарии к проекту. Выбрав нужные параметры, следует нажать кнопку «OK» для подтверждения выбора или «Отмена» – для отмены. Настройки типа ПЧ дополнительно настраиваются в окне «Настройка типа привода» (рисунок 3.13).

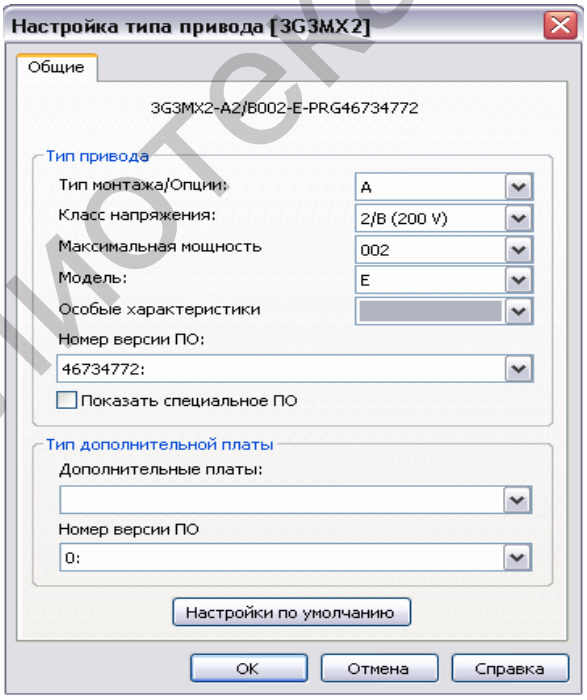


Рисунок 3.13 – Окно настройки параметров нового проекта

Для открытия уже существующего проекта необходимо в выпадающем меню «Файл» выбрать пункт «Открыть». В открывшемся окне выбрать не-

Библиотека БГУИР

обходимый проект и нажать «ОК». После этого вид программы должен измениться – появится окно менеджера проекта и окно редактирования программы (рисунок 3.14).

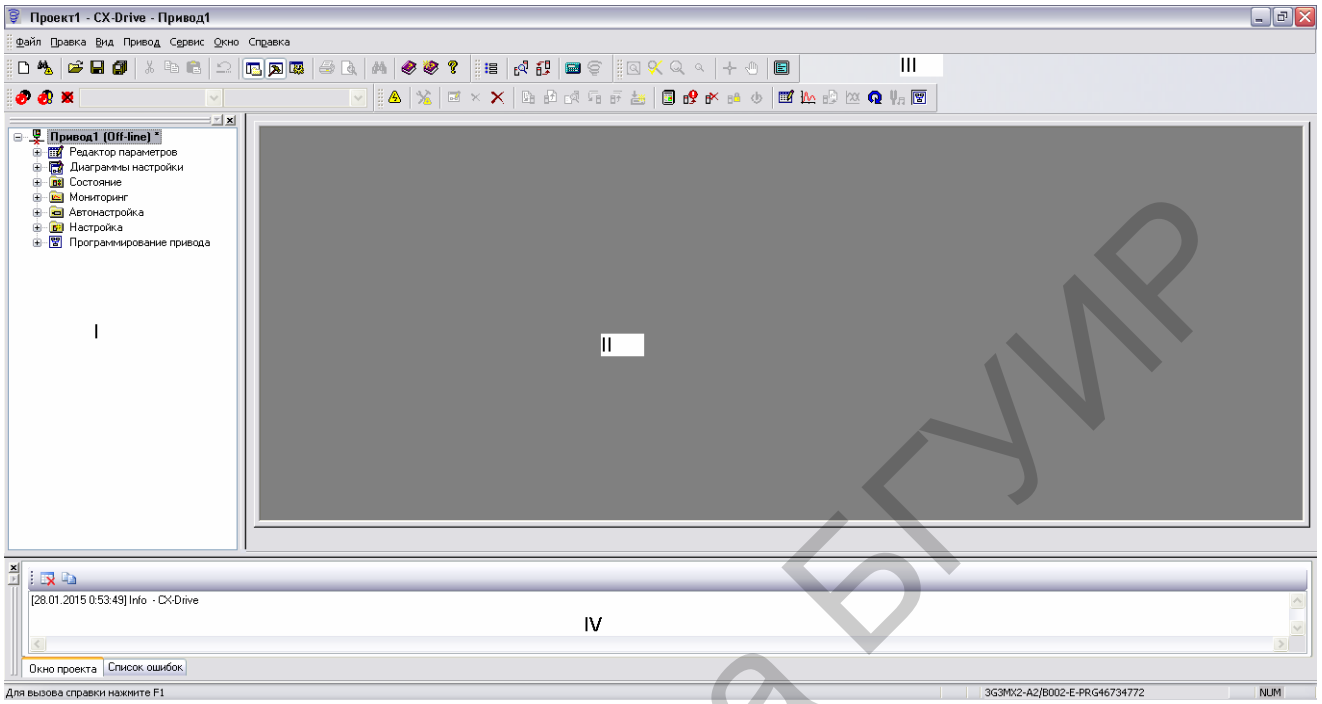


Рисунок 3.14 – Главное окно программы CX-Drive

Всю рабочую область программы CX-Drive (см. рисунок 3.14) можно разделить на несколько областей:

1) окно менеджера проекта;
2) рабочая область;
3) панели инструментов;
4) строка сообщений;
5) строка состояния.

В окне менеджера проекта все параметры, настройки и дополнительные функциональные возможности расположены иерархически и могут быть задействованы пользователем в зависимости от режима работы ПЧ.

Режим работы «Редактор параметров». В отдельных закладках по папкам (рисунок 3.15) сгруппированы все параметры, доступные для просмотра и редактирования.

Библиотека БГУИР

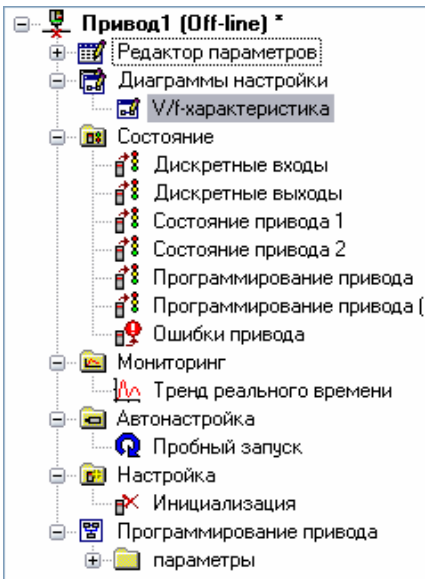


Рисунок 3.15 – Режим работы «Редактор параметров»

Для просмотра или редактирования какого-либо параметра необходимо открыть соответствующую папку и дважды щелкнуть левой кнопкой мыши по нужной группе параметров. Для редактирования параметров в открывшемся окне (рисунок 3.16) необходимо щелкнуть по нему и в строке ввести заданное значение параметра.

Таким образом и осуществляется конфигурирование всех необходимых параметров при настройке ПЧ MX2 на заданный режим работы.

Рисунок 3.16 – Конфигурирование параметров ПЧ MX2

Режим работы «Диаграммы настройки» (см. рисунок 3.15). В закладке находится вложение «V/f-характеристика». То есть на отдельном экране (рисунок 3.17) представляется настроенная с помощью соответствующих параметров V/f-характеристика ПЧ MX2.

Библиотека БГУИР

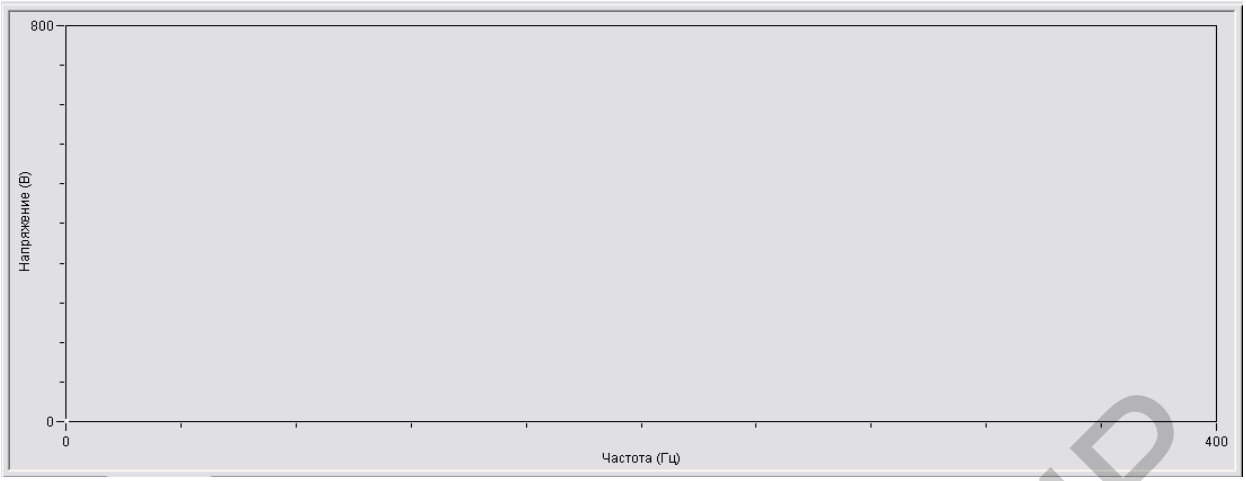


Рисунок 3.17 – V/f-характеристика

Режим работы «Состояние» (см. рисунок 3.15). В закладках находятся меню, в которых располагаются все контрольные данные (рисунок 3.18), дающие полное представление о состоянии ПЧ. Использование этой закладки актуально только при работе с ПЧ в режиме онлайн.

	Название	Описание	Название сигнала	Значение
☑	d005.1	Состояние настраиваемого входа	0: FW (ход вперед/стоп)	
☑	d005.2	Состояние настраиваемого входа	1: RV (ход назад/стоп)	
☑	d005.3	Состояние настраиваемого входа	12: EXT (внешнее отключение выхо	
☑	d005.4	Состояние настраиваемого входа	18: RS (сброс инвертора)	
☑	d005.5	Состояние настраиваемого входа	2: CF1 (выбор предустановленной	
☑	d005.6	Состояние настраиваемого входа	3: CF2 (выбор предустановленной	
☑	d005.7	Состояние настраиваемого входа	6: JG (толчковый ход)	

Рисунок 3.18 – Контроль состояния дискретных входов ПЧ MX2

Режим работы «Мониторинг» (см. рисунок 3.15). В закладке расположен график «Тренд реального времени» (рисунок 3.19), с помощью которого в режиме онлайн можно наблюдать за изменением выбранного параметра.

Библиотека БГУИР

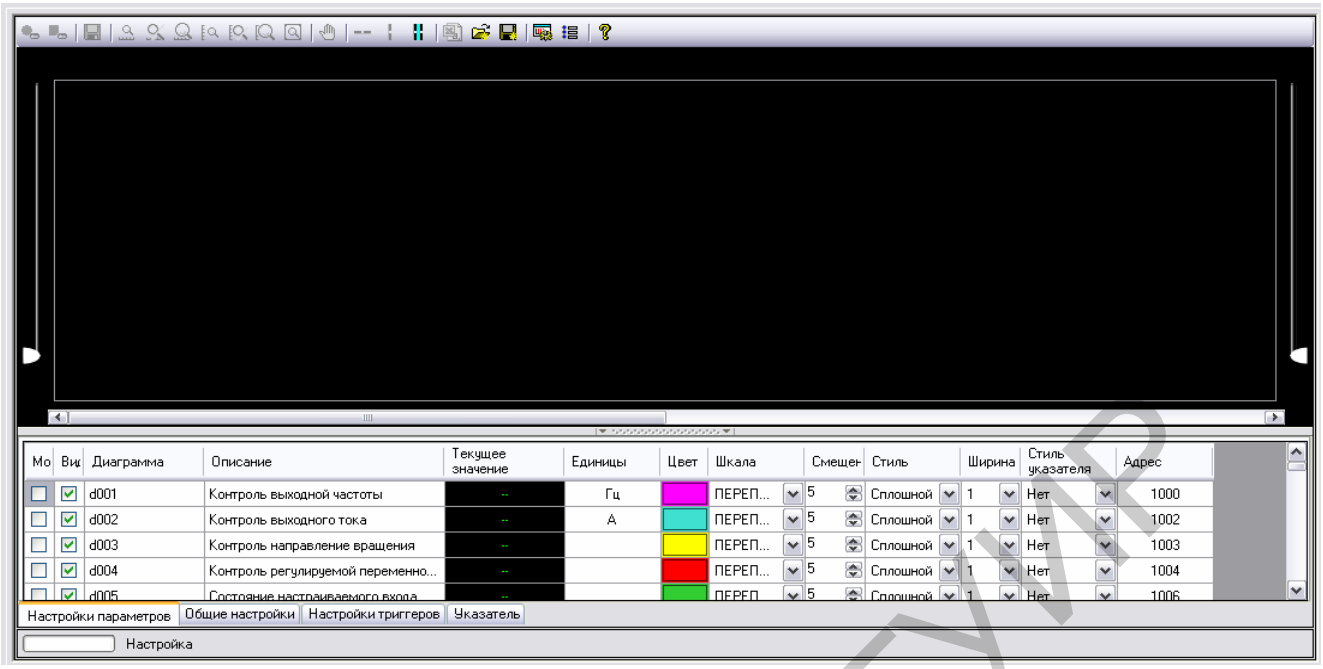


Рисунок 3.19 – Тренд реального времени

В нижней части экрана расположены закладки настройки отображаемых графиков, с помощью которых можно настроить следующие параметры:

– отображаемые на графике данные (код параметра);
– цвет;
– смещение относительно оси абсцисс;
– стиль;
– другие графические параметры отображаемых кривых.

В верхней части экрана расположен непосредственно график, на котором и отображаются тренды различных выбранных параметров.

Режим работы «Автонастройка» (см. рисунок 3.15). В закладке находится иконка пробного запуска, при нажатии на которую в режиме онлайн можно будет осуществить пробный запуск ПЧ MX2 (рисунок 3.20).

Библиотека БГУИР

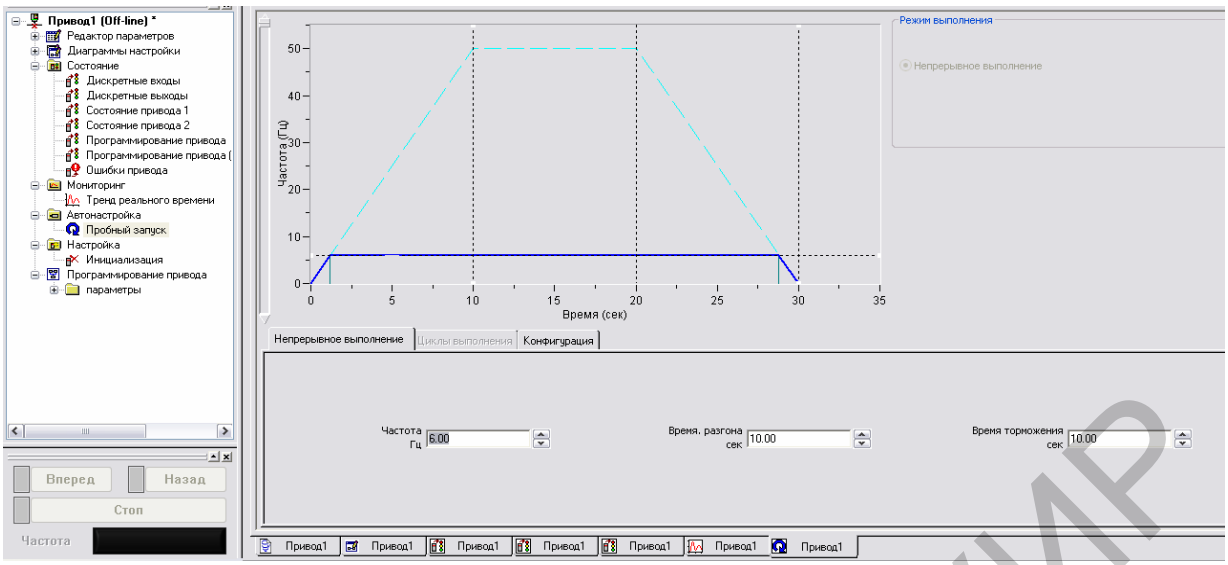


Рисунок 3.20 – Пробный запуск

Для согласия с предупреждением необходимо нажать кнопку «ОК». В открывшемся окне появится виртуальный пульт управления ПЧ, включающий в себя кнопки «Вперед», «Назад», «Стоп» и индикатор «Частота». Кроме того, на графике в рабочем окне проекта будет отображаться график изменения частоты вращения двигателя, а также пользователь с помощью соответствующих полей самостоятельно может настроить базовые параметры разгона/торможения и установившегося движения.

Режим работы «Настройка» (см. рисунок 3.15). В закладке находится иконка инициализации параметров ПЧ, при нажатии на которую в режиме онлайн можно будет произвести инициализацию (возврат к заводским уставкам) ПЧ MX2.

Для согласия с предупреждением необходимо нажать кнопку «ОК». После этого в открывшемся окне необходимо выбрать параметры инициализации в соответствующих выпадающих списках и для начала инициализации нажать соответствующую кнопку.

Режим работы «Программирование привода» (см. рисунок 3.15). В закладке находится папка «параметры», в которой в алфавитном порядке располагаются все параметры ПЧ MX2, отвечающие за работу привода в пользовательском программном режиме.

Библиотека БГУИР

4 ЯЗЫК ФУНКЦИОНАЛЬНЫХ БЛОКОВ FBD

4.1 Функциональная декомпозиция. Назначение и описание языка FBD

Функциональная декомпозиция – это метод разбиения алгоритма программы на отдельные составляющие, которые описывают определенные функции процесса. Если блоки кода программы, основанные на этих алгоритмах, встречаются часто, то в языках высокого уровня их объединяют в функции, процедуры или методы.

На автоматизированных системах управления технологическим процессом (АСУ ТП) в случае применения ПЛК большинство базовых программ, обеспечивающих чтение информации с датчиков и базовое управление объектами, написаны с использованием языка релейных диаграмм LD. Однако, когда технологический процесс усложняется, появляются новые возможности, которые реализовать на языке LD затруднительно:

- общение с другими устройствами по последовательным интерфейсам;
- точное позиционирование двигателей;
- сложные математические операции;
- преобразования аналоговых сигналов.

Также даже в случае простого технологического процесса есть участки кода, которые многократно повторяются.

Для такого случая стандартом IEC 61131 предусмотрен язык функциональных блок-диаграмм (FBD).

FBD (Function Block Diagram) – это графический язык программирования. Диаграмма FBD очень напоминает принципиальную схему электронного устройства на микросхемах. В отличие от LD «проводники» в FBD могут проводить сигналы (передавать переменные) любого типа (логический, аналоговый, время). Шины питания на FBD-диаграмме не показываются. Выходы блоков могут быть поданы на входы других блоков либо непосредственно на выходы ПЛК. Сами блоки, представленные на схеме как «черные ящики», могут выполнять любые функции.

FBD-схемы очень четко отражают взаимосвязь входов и выходов диаграммы. Если алгоритм изначально хорошо описывается с позиции сигналов, то его FBD-представление всегда получается нагляднее, чем в текстовых языках.

FBD-диаграмма строится из компонентов, отображаемых на схеме прямоугольниками. Входы изображаются слева от прямоугольника, выходы –

Библиотека БГУИР

справа. Внутри прямоугольника указывается типы и наименования входов и выходов. Для экземпляра функционального блока его наименование указывается сверху, над прямоугольником. В графических системах программирования прямоугольник компонента может содержать картинку, отражающую его тип. Размеры прямоугольника зависят от числа входов и выходов и устанавливаются графическим редактором автоматически.

Программа в FBD необязательно должна представлять большую единую схему. Как и в LD, диаграмма образуется из множества цепей, которые выполняются одна за другой.

Прямоугольники в FBD соединены линиями связи. Соединения имеют направленность слева направо. Вход блока может быть соединен с выходом блока, расположенного слева от него. Помимо этого, вход может быть соединен с переменной или константой. Соединение должно связывать переменные или входы и выходы одного типа. В отличие от компонента, переменная изображается на диаграмме без прямоугольной рамки. Ширина соединительной линии в FBD роли не играет. Стандарт допускает использование соединительных линий разной ширины и стиля для соединений разного типа.

Выполнение FBD-цепей идет слева направо, сверху вниз. Блоки, расположенные левее, выполняются раньше. Блок начинает вычисляться только после вычисления значений всех его входов. Дальнейшие вычисления не будут продолжены до вычисления значений на всех выходах. Другими словами, значения на всех выходах графического блока появляются одновременно. Вычисление цепи считается законченным только после вычисления значений на выходах всех входящих в нее элементов.

В некоторых системах программирования пользователь имеет возможность свободно передвигать блоки с сохранением связей. В этом случае ориентироваться нужно исходя из порядка соединений.

Соединители представляют собой поименованное соединение, которое можно разорвать и перенести в следующую цепь. Такой прием может понадобиться при ограниченной ширине окна редактора FBD.

Стандарт не запрещает соединения, идущие с выхода блока на свой вход или вход ранее исполняемых блоков. Обратная связь не образует цикл, подобный FOR**,** просто некоторое вычисленное значение поступит на вход при следующем вызове диаграммы. Фактически это означает неявное создание переменной, которая сохраняет свое значение между вызовами диаграммы. Для устранения неоднозначности необходимо присвоить безопасное начальное значение переменной обратной связи.

Библиотека БГУИР

Порядок выполнения FBD-цепей диаграммы можно принудительно изменять, используя метки и переходы, точно так же, как и в релейных схемах.

Метка ставится в начале любой цепи, являясь, по сути, названием данной цепи. Цепь может содержать только одну метку. Имена меток подчинены общим правилам наименования идентификаторов Международной электротехнической комиссии. Графический редактор автоматически нумерует цепи диаграммы. Эта нумерация применяется исключительно для документирования и не может заменять метки.

Переход обязательно связан с логической переменной и выполняется, если переменная имеет значение TRUE. Для создания безусловного перехода используется константа TRUE, связанная с переходом.

Оператор возврата RETURN можно использовать в FBD так же, как и переход на метку, то есть в связке с логической переменной. Возврат приводит к немедленному окончанию работы программного компонента и возврату на верхний уровень вложений. Для основной программы это начало рабочего цикла ПЛК.

4.2 Основы работы с функциональными блоками

Программный пакет CX-Programmer предусматривает возможность упрощения процедуры программирования, если возникает необходимость в разработке системы управления однотипными технологическими операциями.

В качестве примера предлагается вариант использования функциональных блоков (FB) в управляющей программе ПЛК для управления испытаниями бензинового и дизельного двигателей.

Необходимо отметить, что испытания двигателей производятся по одинаковому алгоритму.

При кратковременном появлении команды включения «Включить» двигатель должен включиться и сформировать сигнал «Включен».

При достижении фактической скорости «Фактическая скорость» заданного значения «Задание» формируется сигнал «Готов». Двигатель отключается по сигналу отключения «Отключить» или по аварийному сигналу «Неисправность».

Кроме того, каждый двигатель имеет свой вентилятор. Вентилятор включается при работе двигателя и продолжает еще работать 4 с после отключения двигателя.

В данном случае необходимо разработать два FB: для управления включением/отключением двигателя и управления включением/отключением вентилятора. Таким образом, однократно запрограммированный FB может многократно вызываться в различных сегментах программы.

Библиотека БГУИР

Разработка управляющей программы производится следующим образом:

1. Разработать FB для управления включением/отключением двигателя, которому будет присвоено символьное имя «Двигатель».

2. Определиться с символьным представлением сигналов и команд (таблица 4.1), которое задается в окне «Символы».

Таблица 4.1– Символьное представление команд и сигналов

Название	Тип данных	Адрес/ Значение	Комментарий
Бенз_Вкл	BOOL	0.00	Включить бензиновый двигатель
Бенз_Откл	BOOL	0.01	Отключить бензиновый двигатель
Бенз_Неиспр	BOOL	0.02	Неисправность бензинового двигателя
Бенз_Задание	NUMBER	1500	Заданная скорость бензинового двигателя
Бенз_Факт	INT	D100	Фактическая скорость бензинового двигателя
Бенз_Включен	BOOL	100.00	Бензиновый двигатель включен
Бенз_Готов	BOOL	100.01	Заданная скорость достигнута
Бенз_Вент_Вкл	BOOL	100.02	Вентилятор бензинового двигателя включен
Диз_Вкл	BOOL	0.03	Включить дизельный двигатель
Диз_Откл	BOOL	0.04	Отключить дизельный двигатель
Диз_Неиспр	BOOL	0.05	Неисправность дизельного двигателя
Диз_Задание	NUMBER	1200	Заданная скорость дизельного двигателя
Диз_Факт	INT	D101	Фактическая скорость дизельного двигателя
Диз_Включен	BOOL	100.03	Дизельный двигатель включен
Диз_Готов	BOOL	100.04	Заданная скорость достигнута
Диз_Вент_Вкл	BOOL	100.05	Вентилятор дизельного двигателя включен

3. Для переменных «Бенз_Задание» и «Диз_Задание» в столбце «Адрес/значение» необходимо указать значения частоты вращения бензинового и дизельного двигателей – соответственно 1500 и 1200 оборотов.

4. Для создания нового FB выбрать «Вставить функциональный блок» → «Лестничная диаграмма» (рисунок 4.1).

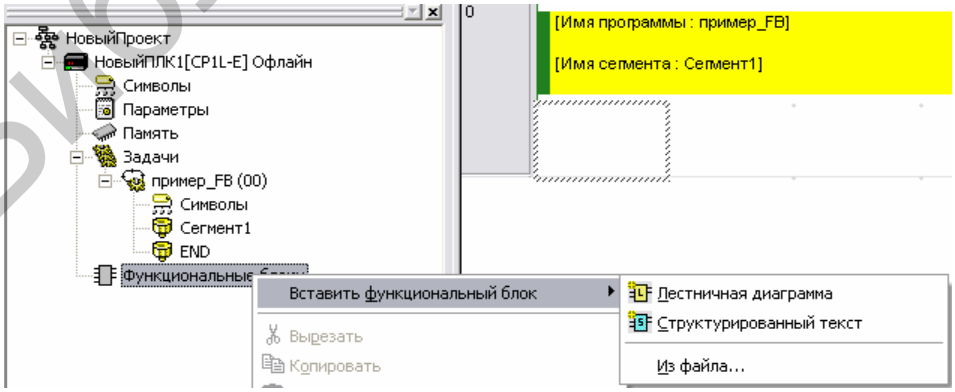


Рисунок 4.1 – Порядок вызова нового функционального блока

Библиотека БГУИР

5. В открывшемся окне присвоить функциональному блоку имя «Двигатель» и закрыть окно.

6. Для программирования FB дважды щелкнуть левой кнопкой мыши по имени блока. В основном окне программы должно появиться содержимое функционального блока.

7. Заполнить таблицу описания переменных в соответствующих закладках. Для этого открыть необходимую закладку. Далее в пустой строке нажать правой кнопкой мыши и в выпадающем списке выбрать «Вставить переменную». В окне (рисунок 4.2) выбрать параметры переменной (Имя – Тип данных – Использование – Исходное значение). После задания переменных окна примут вид, представленный на рисунке 4.3.

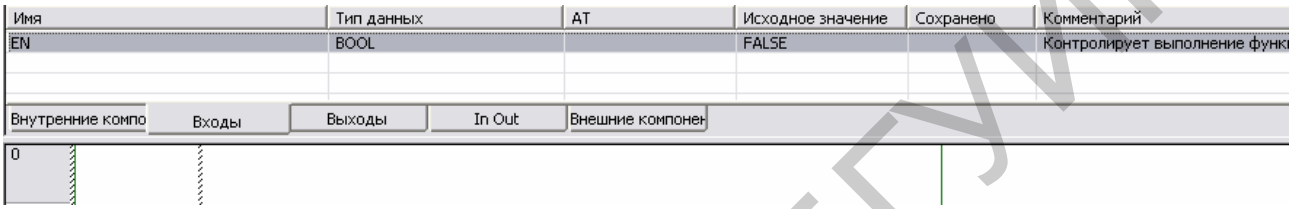


Рисунок 4.2 – Окно программирования функционального блока

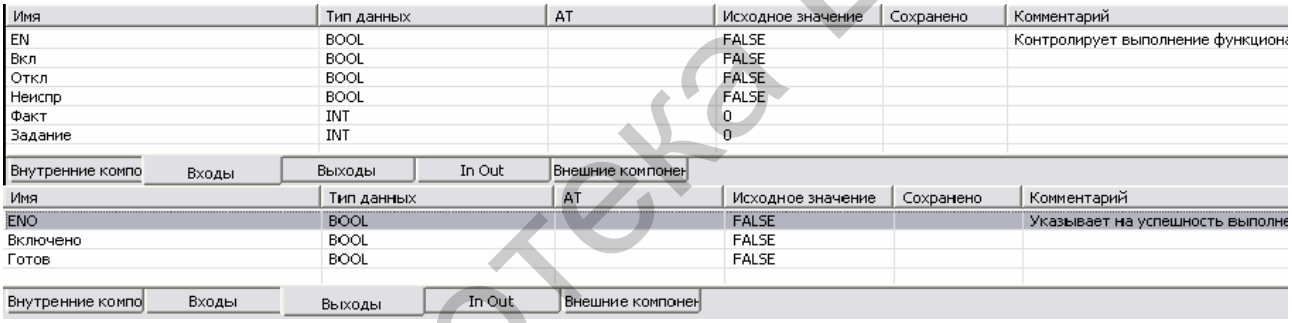


Рисунок 4.3 – Окна задания переменной функционального блока

8. В соответствии с рисунком 4.4 набрать программу управления включением и отключением двигателя на языке лестничных диаграмм. Рассматриваемый FB должен управлять и контролировать работу бензинового и дизельного двигателей.

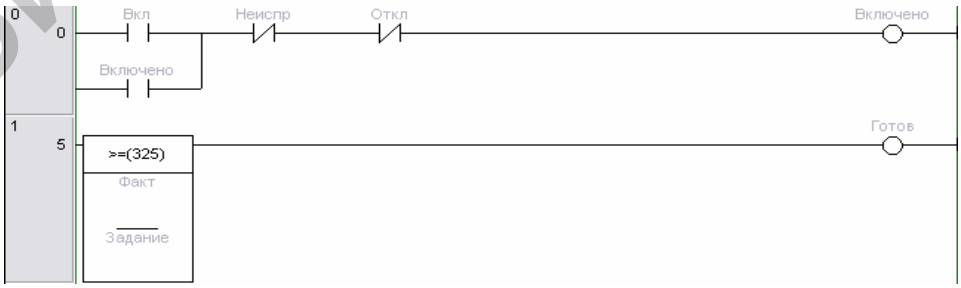


Рисунок 4.4 – Программа FB «Двигатель»

Библиотека БГУИР

9. Открыть основной сегмент программы «Сегмент 1» и в ступени 0 установить курсор в произвольном положении.

10. На панели инструментов выбрать элемент «Создать вызов функционального блока».

11. В открывшемся окне (рисунок 4.5) необходимо указать имя экземпляра FB «Бензиновый».

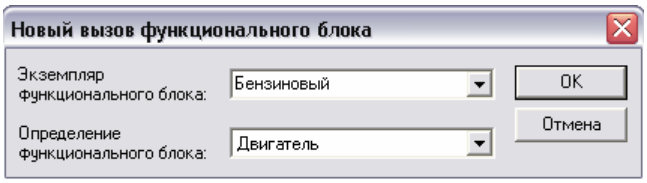


Рисунок 4.5 – Процедура вызова экземпляра FB «Двигатель»

12. Перед каждым из входов и выходов вставить параметры созданного экземпляра FB. Для этого необходимо подвести курсор к ячейке, расположенной напротив слева или справа соответственно входа или выхода, нажать левую кнопку мыши и в выпадающем списке (рисунок 4.6) выбрать необходимое символьное имя параметра.

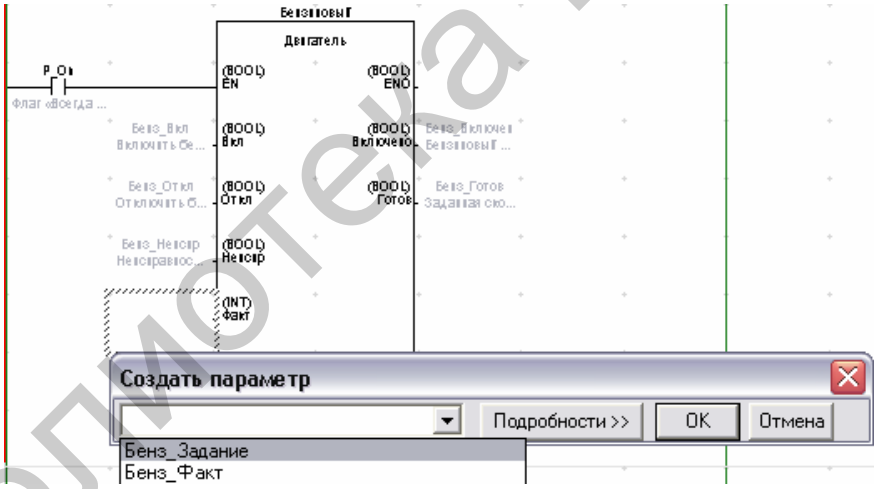


Рисунок 4.6 – Процедура вызова экземпляра FB «Двигатель»

13. В ступени номер 2 вызвать еще один экземпляр FB «Двигатель» и присвоить ему имя «Дизельный». Таким образом, основной сегмент программы примет вид, изображенный на рисунке 4.7.

14. Аналогичным образом создать FB «Вентилятор» для управления включением и отключением вентилятора. На рисунке 4.8 представлена таблица переменных, на рисунке 4.9 – программа FB «Вентилятор».

Библиотека БГУИР

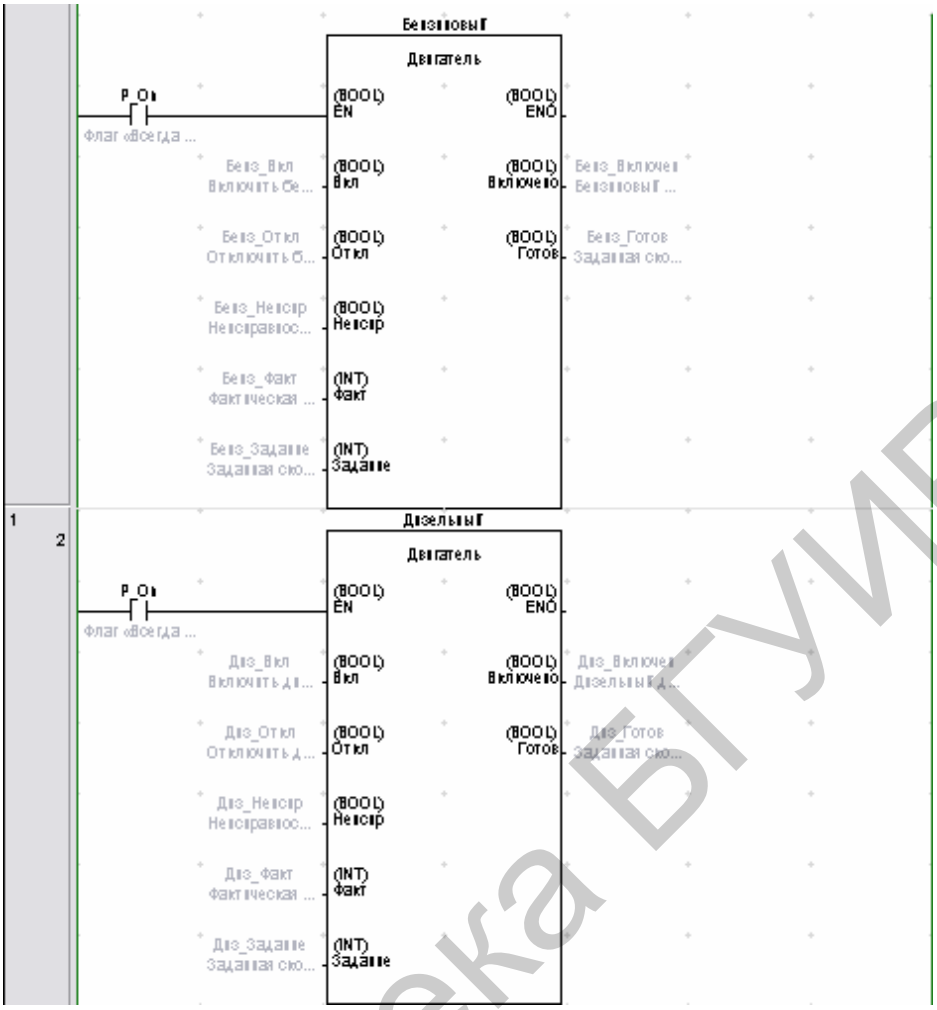


Рисунок 4.7 – Процедура вызова экземпляра FB «Двигатель»

Имя	Тип данных	AT	Исходное значение	Сохранено	Комментарий
Вент_таймер	TIMER				

Внутренние компо | Входы | Выходы | In Out | Внешние компонен

Имя	Тип данных	AT	Исходное значение	Сохранено	Комментарий
EN	BOOL		FALSE		Контролирует
Включено	BOOL		FALSE		

Внутренние компо | Входы | Выходы | In Out | Внешние компонен

Имя	Тип данных	AT	Исходное значение	Сохранено	Комментарий
ENO	BOOL		FALSE		Указывает на
Вент_включен	BOOL		FALSE		

Внутренние компо | Входы | Выходы | In Out | Внешние компонен

Рисунок 4.8 – Таблица символов

Библиотека БГУИР

Рисунок 4.9 – Программирование FB «Вентилятор»

15. В основном сегменте программы вызвать два экземпляра FB «Вентилятор» с именами «Бенз_Вент» и «Диз_Вент» соответственно и «подключить» к их входам и выходам параметры конкретного экземпляра FB, имеющие символьные имена, указанные в таблице 4.1. После этого рассматриваемый фрагмент программы примет вид, представленный на рисунке 4.10.

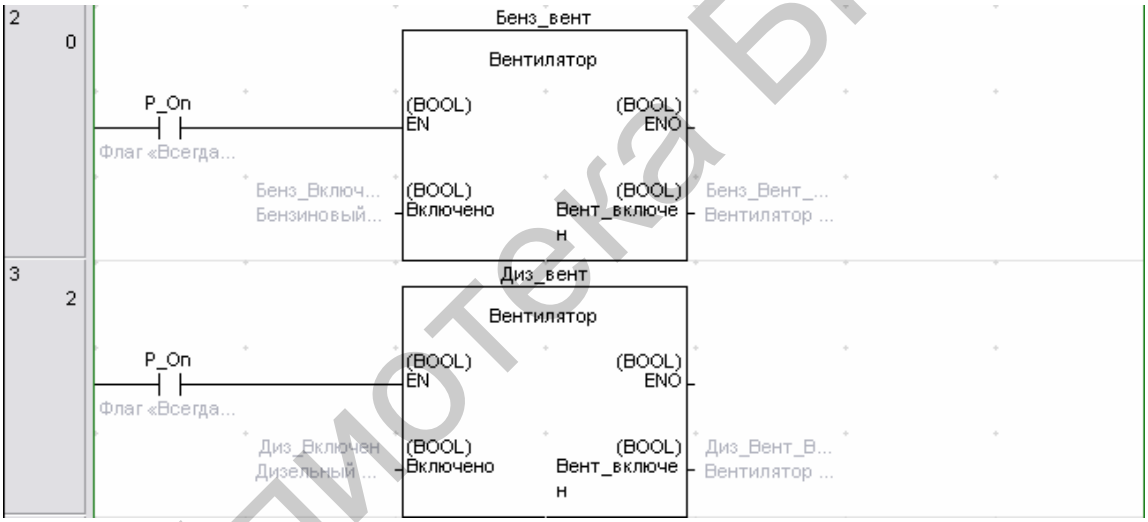


Рисунок 4.10 – Вызов FB «Вентилятор» в основном сегменте программы

Библиотека БГУИР

5 ЯЗЫК СТРУКТУРИРОВАННОГО ТЕКСТА (ST). ИСПОЛЬЗОВАНИЕ АНАЛОГОВЫХ СИГНАЛОВ

5.1 Язык структурированного текста

Это текстовый язык высокого уровня с инструкциями и синтаксисом уровня адаптированного языка Паскаль. Он позволяет программировать сложные алгоритмы обработки данных – последовательности команд с использованием переменных, вызовов функций и функциональных блоков, операторов повторения и т. д., а также применяется для описания действий внутри шагов и условий языка SFC. В основном используется в тех случаях, когда алгоритм трудно описать с помощью графических языков.

Имена, используемые в исходном коде (идентификаторы переменных, константы, ключевые слова), разделены неактивными разделителями (пробелами, символами окончания строки и табуляции) или активными разделителями, которые имеют заранее определенное значение (символ-разделитель «>» означает сравнение «больше чем», а символ «+» – операцию сложения и т. д.).

Неактивные разделители могут быть свободно введены между активными разделителями, константами и идентификаторами. В отличие от неформатных языков, таких как IL, конец строки может быть введен в любом месте программы.

Основой ST-программы служат выражения. Выражения состоят из операндов (констант и переменных) и операторов. Операторы являются «командами» языка программирования ST. Они должны заканчиваться точкой с запятой. Одна строка может содержать несколько операторов, отделяемых точками с запятой.

Тип всех операндов выражения должен быть одинаковым. Для изменения типов необходимо использовать функции преобразования типов: BOO, ANA, REAL, TMR и MSG.

Результат вычисления выражения присваивается переменной при помощи оператора присваивания «:=». Каждое выражение обязательно заканчивается точкой с запятой «;». Выражение состоит из переменных, констант и функций, разделенных операторами, например:

```
Var1 := 1+Var2 / ABS(Var2);
```

Стандартные операторы в выражениях ST имеют символьное представление, например, математические действия: +, –, *, /, операции сравнения и т. д.

Библиотека БГУИР

Выражение может включать другое выражение, заключенное в скобки, которое вычисляется в первую очередь.

Вычисление выражения происходит в соответствии с правилами приоритета операций. Первыми выполняются операции с наивысшим приоритетом.

Для того чтобы отделить подчасти выражения и явно определить приоритетность операций, используются скобки. Когда в сложном выражении нет скобок, приоритетность ST-операторов задана неявно.

В порядке уменьшения приоритета операции располагаются следующим образом:

- выражение в скобках;
- вызов функции;
- степень EXPT;
- замена знака (-);
- отрицание NOT;
- умножение, деление и деление по модулю MOD;
- сложение и вычитание (+, –);
- операции сравнения: равенство (=); неравенство (<, >);
- логические операции AND, XOR и OR.

При составлении выражений необходимо учитывать возможный диапазон изменения значений и типы переменных. Ошибки, связанные с переполнением, возникают в процессе выполнения и не могут быть обнаружены транслятором.

Оператор выбора позволяет выполнить различные группы выражений в зависимости от условий, представленных логическими выражениями. Полный синтаксис оператора IF (если):

IF <логическое выражение IF>

THEN

<выражения IF>;

[

ELSIF <логическое выражение ELSIF 1>

THEN

< выражения ELSIF 1**>** ;

…

ELSIF <логическое выражение ELSEIF n>

Библиотека БГУИР

THEN

< выражения ELSEIF n> ;

ELSE

< выражения ELSE> ;

]

END_IF

Если <логическое выражение IF> ИСТИНА, то выполняются выражения первой группы – <выражения IF>. Прочие выражения пропускаются, альтернативные условия не проверяются.

Если <логическое выражение IF> ЛОЖЬ, то одно за другим проверяются условия ELSIF. Первое истинное условие приведет к выполнению соответствующей группы выражений. Прочие условия ELSIF анализироваться не будут. Групп ELSIF может быть несколько или не быть совсем.

Если все логические выражения дали ложный результат, то выполняются выражения группы ELSE, если она есть. Если группы ELSE нет, то не выполняется ничего.

Оператор множественного выбора CASE позволяет выполнить различные группы выражений в зависимости от значения одной целочисленной переменной или выражения. Синтаксис:

CASE <целочисленное выражение> **OF**

<значение 1>:

<выражения 1>;

<значение 2>, <значение 3>:

<выражения 3>;

<значение 4>..<значение 5> :

<выражения 4> ;

…

[

ELSE

<выражения ELSE>;

Библиотека БГУИР

]

END_CASE

Если значение выражения совпадает с заданной константой, то выполняется соответствующая группа выражений. Прочие условия не анализируются (<значение 1>: <выражения 1> ;).

Если несколько значений констант должны соответствовать одной группе выражений, их можно перечислить через запятую (<значение 2> , <значение 3> : <выражения 3> ;).

Диапазон значений можно определить через двоеточие (<значение 4>..<значение 5> : <выражения 4> ;).

Группа выражений ELSE является необязательной. Она выполняется при несовпадении ни одного из условий (<выражения ELSE> ;).

Безусловно оператор CASE «слабее» оператора IF, но формат CASE не только выразителен для программиста, но и более эффективен. Использование целочисленных констант позволяет транслятору выполнить оптимизацию кода, часто весьма существенную.

Циклы **WHILE** и **REPEAT** обеспечивают повторение группы выражений, пока верно условное логическое выражение. Если условное выражение всегда истинно, то цикл становится бесконечным.

Синтаксис **WHILE:**

WHILE <Условное логическое выражение> **DO**

<Выражения — тело цикла>

END_WHILE

Условие в цикле WHILE проверяется до начала цикла. Если логическое выражение изначально имеет значение ЛОЖЬ, тело цикла не будет выполнено ни разу.

Синтаксис REPEAT:

REPEAT

<Выражения — тело цикла>

UNTIL <Условное логическое выражение>

END_REPEAT

Библиотека БГУИР

Условие в цикле REPEAT проверяется после выполнения тела цикла. Если логическое выражение изначально имеет значение ЛОЖЬ, тело цикла будет выполнено один раз.

Правильно построенный цикл WHILE или REPEAT обязательно должен изменять переменные, составляющие условие окончания в теле цикла, постепенно приближаясь к условию завершения. Если этого не сделать, цикл не закончится никогда.

Цикл **FOR** обеспечивает заданное количество повторений группы выражений. Синтаксис:

FOR <Целый счетчик> := <Начальное значение>

TO <Конечное значение>

[BY <Шаг>] **DO**

<Выражения – тело цикла>

END_FOR

5.2 Основы работы с аналоговыми сигналами

ПЛК Omron CP1L-E имеет два встроенных аналоговых входа. Кроме того, в соответствующее гнездо установлена плата расширения аналоговых входов/выходов.

Подача сигналов по напряжению от 0 до 10 В на аналоговые входы осуществляется от внешних источников через гнезда AI0 и AI1 при использовании внешних источников сигнала, например, встроенного потенциометра RP1 или выхода измерителя Omron K3HB-V. Вывод аналоговых сигналов по напряжению 0–10 В на какие-либо исполнительные устройства, например, преобразователь частоты Omron MX 2 или индикатор PV1, осуществляется через гнезда AO0 и AO1.

В таблице 5.1 приведены сводные характеристики используемых входов и выходов.

Библиотека БГУИР

Таблица 5.1 – Характеристики аналоговых входов/выходов ПЛК

Параметр	Значение					
	Аналоговые входы				Аналоговые выходы	
	AI0	AI1	AI2	AI3	AO0	AO
Физическое расположение	Встроенные аналоговые входы		Плата расширения аналоговых входов/выходов MAB221			
Тип	Напряжение 0…10 В		Напряжение 0…10 В, ток 4…20 мА		Напряжение 0…10 В	
Разрешение	1/1000		1/4000 (напряжение) 1/2000 (ток)		1/4000	
Диапазон преобразования данных	0000h…03E8h		0000h…0FA0h (напряжение) 0000h…07D0h (ток)		0000h…0FA0h	
Адрес в памяти ПЛК	A642	A643	CIO2980	CIO2981	CIO2985	CIO2986

Входные аналоговые сигналы преобразуются в цифровую форму. Соотношения между диапазонами входных сигналов и диапазонами цифровых значений для встроенных аналоговых входов и выходов платы расширения представлены на рисунках 5.1 и 5.2 соответственно.

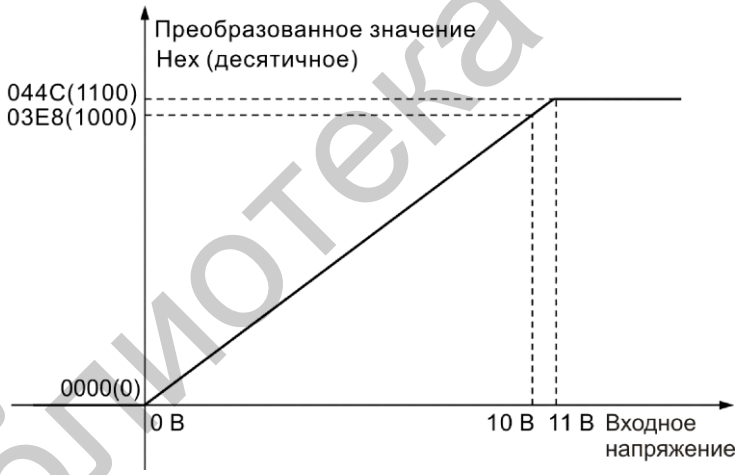


Рисунок 5.1 – Характеристика соответствия входных аналоговых сигналов с цифровыми значениями

Библиотека БГУИР

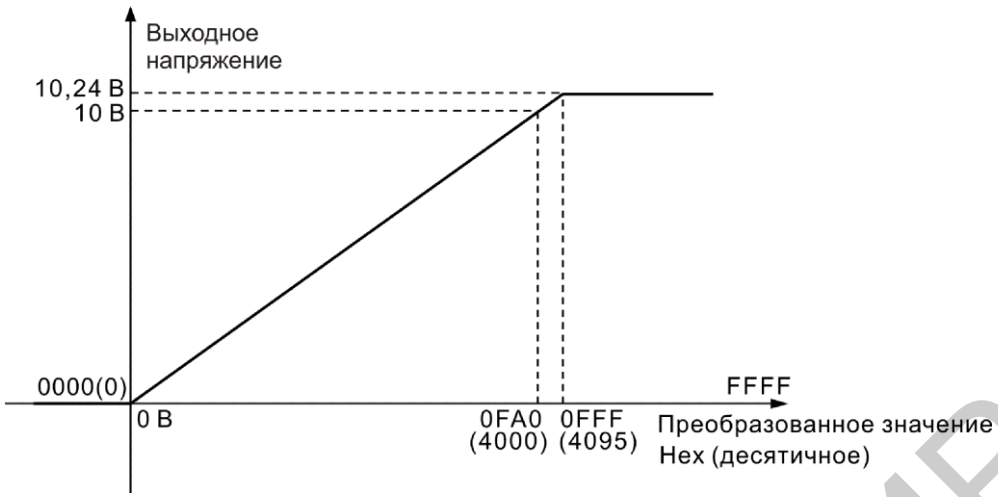


Рисунок 5.2 – Характеристика соответствия выходных аналоговых сигналов цифровым значениям

Таким образом, в данном стенде пользователю для работы доступно два встроенных аналоговых входа A642 и A643 и два аналоговых выхода CIO2985 и CIO2986.

В качестве примера использования аналоговых входов и выходов ПЛК пользователю предлагается следующая задача: необходимо реализовать прямую передачу сигнала с аналогового входа AI0 (адрес A642), напряжение на котором задается потенциометром RP1, на аналоговый выход AO0 (адрес CIO2985), подключенный на вход вольтметра PV1, на котором осуществляется индикация напряжения. При этом необходимо масштабировать входной сигнал таким образом, чтобы на выходе получить аналоговый сигнал от 0 до 10 В.

Для решения поставленной задачи необходимо разработать для ПЛК управляющую программу, выполняя следующие действия:

1. Запустить программу CX-Programmer и создать новый проект.

2. В сегменте программы набрать простую программу, представленную на рисунке 5.3. Команда *(420) производит умножение числа, содержащегося в ячейке A642 (аналоговый вход AI0) на десятичную константу &4 и передает результат операции в ячейку по адресу CIO2985 (аналоговый выход AO0). Использование операции умножения связано с разным разрешением входного (1/1000) и выходного (1/4000) сигналов и применяется для согласования уровней напряжений на входе AI0 и выходе AO0.

Библиотека БГУИР

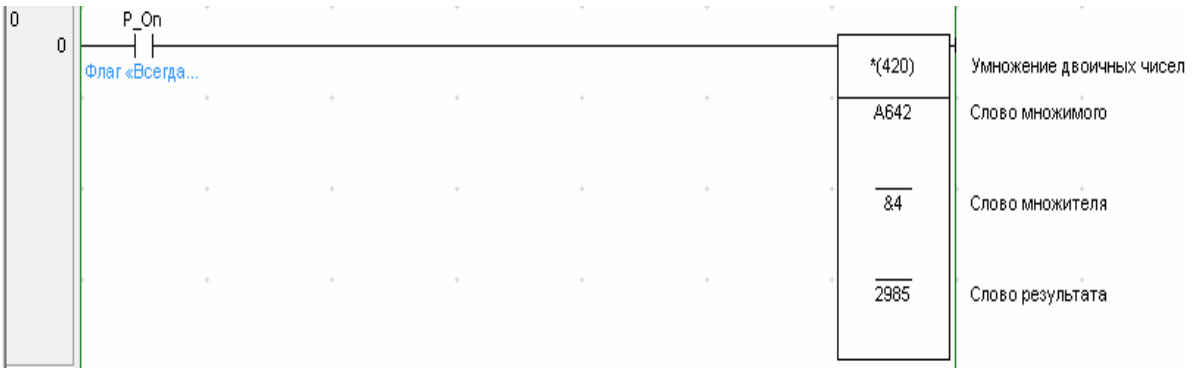


Рисунок 5.3 – Программа передачи аналогового значения со входа на выход

3. Скомпилировать программу и загрузить ее в память ПЛК.

4. Для проверки правильности работы программы потенциометр RP1 подключить к аналоговому входу AI0, аналоговый выход AO0 подключить ко входу вольтметра PV1. Плавно вращая ручку потенциометра, изменять напряжение на аналоговом входе AI0 и наблюдать изменение значения регистрируемого напряжения на вольтметре PV1. Также за работой программы можно наблюдать в режиме мониторинга на экране ПК.

5.3 Пример использования языка структурированного текста в функциональных блоках

В качестве примера рассмотрим функцию, которая должна выполнять аналог функции из предыдущего подраздела, передавать значение аналогового сигнала со входа на выход.

Для этого необходимо выполнить следующие действия:

1. В разделе «Функциональные блоки» с помощью правой кнопки мышки открыть контекстное меню, выбрать опцию «Вставить функциональный блок». В раскрывшемся списке выбрать опцию «Структурированный текст» (рисунок 5.4).

2. В открывшемся окне ввести имя создаваемого блока.

3. Двойным щелчком левой кнопки мыши открыть созданный функциональный блок. Открывшееся окно будет иметь вид, представленный на рисунке 5.5.

Библиотека БГУИР

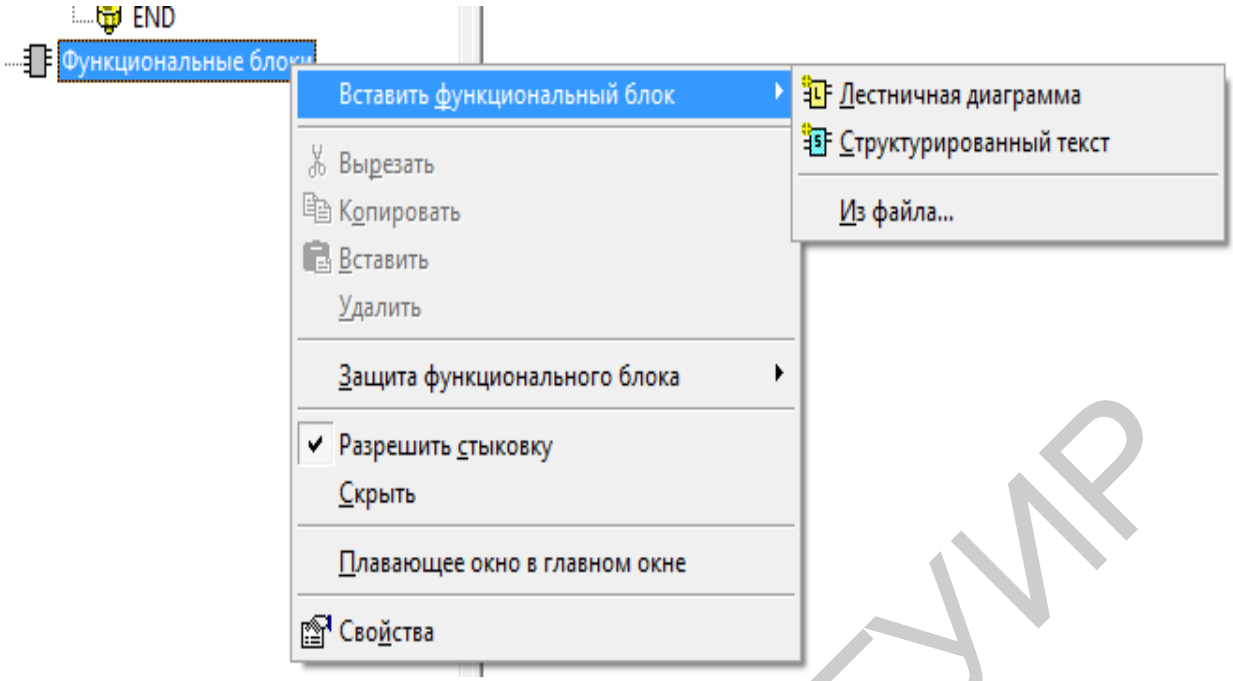


Рисунок 5.4 – Создание функционального блока структурированного текста

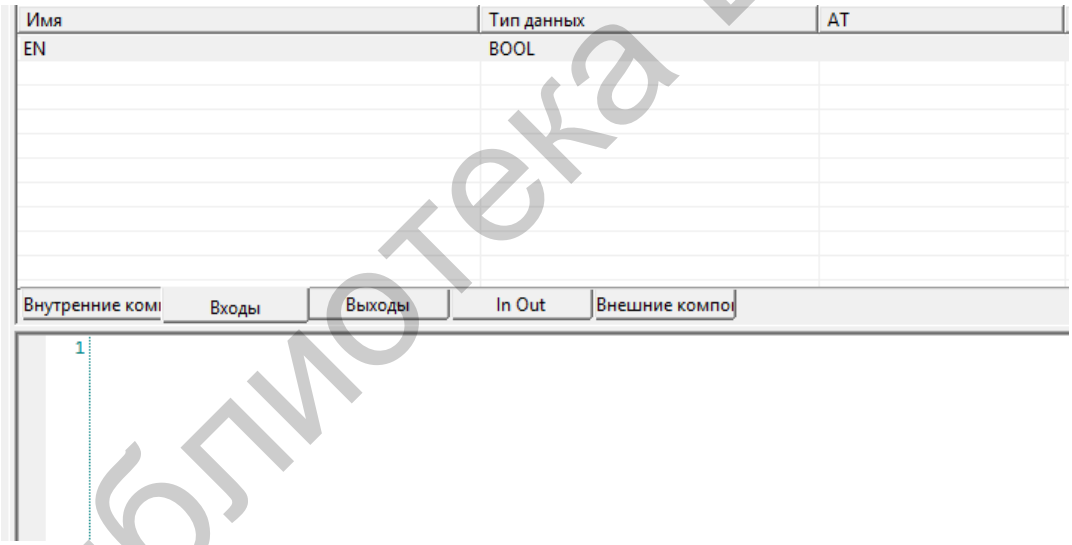


Рисунок 5.5 – Окно редактирования функционального блока

4. Во вкладке «Входы» создать переменную «input_u» типа INT (рисунок 5.6).

Библиотека БГУИР

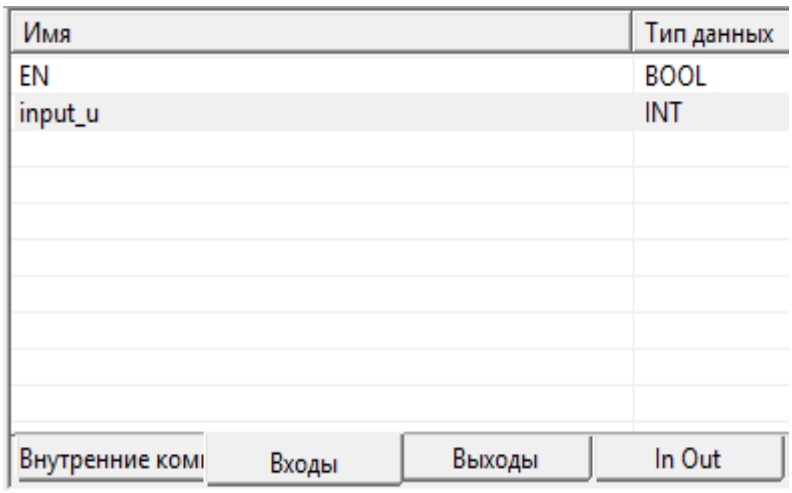


Рисунок 5.6 – Окно создания входных переменных

5. Во вкладке «Выходы», создать переменную «output_u» типа INT (рисунок 5.7).

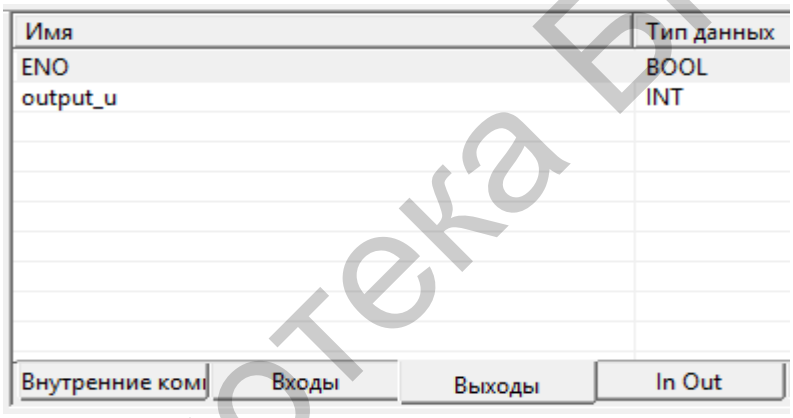


Рисунок 5.7 – Окно создания выходных переменных

6. В окне ввода программы составить следущее выражение: «output_ u := input_u · 4; » . Результат будет иметь вид, представленный на рисунке 5.8.

7. После этого необходимо скомпилировать функциональный блок. А затем вставить его в LD-цепь.

Библиотека БГУИР

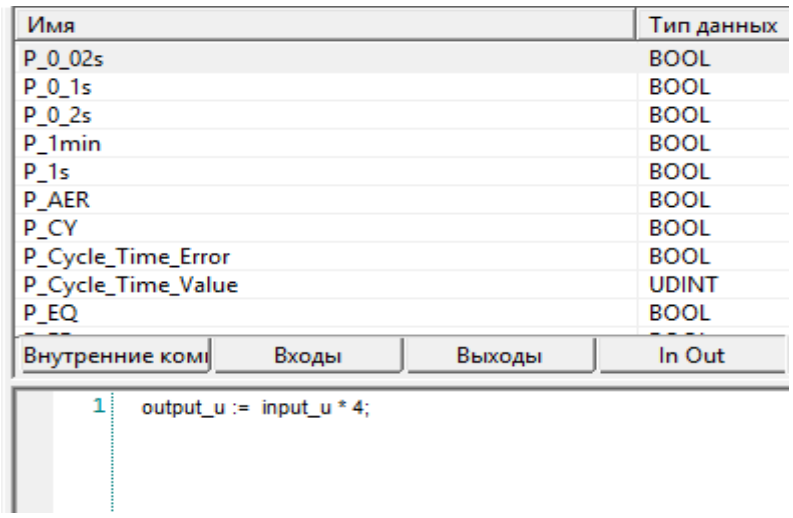


Рисунок 5.8 – Программа масштабирования аналогового значения

Библиотека БГУИР

ПРИЛОЖЕНИЕ А

(*обязательное*)

Контрольные вопросы и задания к лабораторным работам

Лабораторная работа №1
Работа с дискретными входами и выходами

Контрольные вопросы

1. Определение и назначение ПЛК.
2. Классификация языков стандарта МЭК 61131-3.
3. Двоичная система исчисления. Перевод в десятичную и обратно.
4. Шестнадцатеричная система исчисления. Перевод в десятичную и обратно.
5. Двоично-десятичная система.
6. Принципы устройства цифровых входов.
7. Описание областей памяти ПЛК.
8. Виды таймеров, принципы их работы.
9. Виды счетчиков, принципы их работы.
10. Реле. Принцип работы. Примеры использования.

Задания

Задание 1. Работа с логическими командами ПЛК.

Y – обозначения выходных сигналов ПЛК.

X – обозначения входных сигналов ПЛК.

y1 – подключен к зеленой лампе, y2 – к красной лампе.

1. $y1 = x1 \cdot x3 + x1 \cdot x4,$
   $y2 = \overline{y1}.$
2. $y1 = x2 \cdot x5 + x3 \cdot \overline{x2},$
   $y2 = \overline{y1}.$
3. $y1 = x0 \cdot x4 + x6,$
   $y2 = y1.$
4. $y1 = x1 \cdot x6 + x5,$
   $y2 = y1.$
5. $y1 = x0 + x6 + \overline{x1},$
   $y2 = \overline{y1}.$

Библиотека БГУИР

6. $y1 = x0 \cdot \overline{x1}$,
   $y2 = x0 \cdot x1$.
7. $y1 = x0 \cdot \overline{x1} + x3 \cdot x2$,
   $y2 = \overline{y1}$.

Задание 2. Работа с логическими командами ПЛК.

Отключить зеленую лампу переключателем Xi:

1) i = 5;
2) i = 0;
3) i = 2;
4) i = 3;
5) i = 6;
6) i = 4;
7) i = 1.

Задание 3. Работа с таймерами.

1. При включении зеленой лампы она горит 5 с, а потом гаснет.
2. После подачи сигнала зеленая лампа загорается через 6 с.
3. При включении зеленой лампы она горит 3 с, а потом гаснет.
4. После подачи сигнала зеленая лампа загорается через 2 с.
5. При включении зеленой лампы она горит 7 с, а потом гаснет.
6. После подачи сигнала зеленая лампа загорается через 8 с.
7. При включении зеленой лампы она горит 5 с, а потом гаснет.

Задание 4. Работа со счетчиками.

1. Включаем X1 тумблер 5 раз – загорается зеленая лампа, N+1 раз – красная (зеленая гаснет).
2. Включаем тумблер X2 3 раза – каждую секунду начинает моргать зеленая лампа.
3. Включаем тумблер X3 4 раза – через t секунд загорается зеленая лампа.
4. Включаем тумблер X4 6 раз – каждую секунду моргают (поочередно) зеленая и красная лампочки.
5. Включаем тумблер X5 7 раз – через один загораются все выходы.
6. Включаем тумблер X6 8 раз – каждую секунду моргают все выходы.
7. Вначале все лампочки горят. При включении тумблера X7 N раз гаснет половина лампочек.

Библиотека БГУИР

Задание 5. Работа с циклическими командами.

1. Переключая тумблер, организовать переключение светодиода в линейке выходов с адреса 100.00 в правую сторону.
2. Переключая тумблер, организовать переключение светодиода в линейке выходов с адреса 100.07 в левую сторону.
3. Переключая тумблер, организовать переключение светодиода в линейке выходов с адреса 100.00 в правую сторону через один.
4. Организовать переключение светодиода в линейке выходов с адреса 100.00 в правую сторону.
5. Организовать переключение светодиода в линейке выходов с адреса 100.07 в левом направлении каждую секунду.
6. Переключая i-й тумблер, организовать переключение светодиода в линейке выходов вправо. Переключая i+1-й тумблер – влево.
7. Переключая тумблер, организовать переключение светодиода в линейке выходов с адреса 100.07 в левую сторону через один.

Лабораторная работа №2
Использование аналоговых сигналов

Контрольные вопросы

1. Принцип работы ЦАП.
2. Принцип работы АЦП.
3. Устройство аналоговых входов ПЛК.
4. Устройство аналоговых выходов ПЛК.
5. ШИМ-модуляция.
6. Назвать адресацию и основные характеристики аналоговых входов и выходов ПЛК Omron CP1L-E.
7. Примеры обратной связи через аналоговые сигналы.

Задания

Задание 1. Работа с аналоговыми входами.

На аналоговые входы AI0 и AI1 подаются сигналы с потенциометров. Значения с этого входа записываются в рабочую память W и если они равны в пределах погрешности X, то необходимо зажечь зеленую лампу:

1) W = 100, X = 30;
2) W = 200, X = 50;

Библиотека БГУИР

3) W = 105, X = 80;
4) W = 54, X = 100;
5) W = 30, X = 60;
6) W = 25, X = 70;
7) W = 33, X = 40.

Задание 2. Работа с аналоговыми выходами.

Подключить аналоговый выход контроллера к вольтметру и получить на нем сигнал такой же, как с потенциометра (то есть крайнее левое положение потенциометра соответствует 0 В, а крайнее правое – 10 В).

Задание 3. Аналоговые сигналы. Повышенный уровень.

Зеленая лампа меняет свое состояние с частотой, устанавливаемой с помощью потенциометра (1 В – 1 с).

Лабораторная работа №3
Использование функциональных блоков

Контрольные вопросы

1. Что такое функциональная декомпозиция и для чего она нужна?
2. Типы данных в ПЛК.
3. Методика создания блока FBD.
4. Синтаксис языка структурного текста «ST».
5. Области действия переменных.

Задания

Задание 1. Использование функциональных блоков.

Написать управляющую программу с помощью функциональных блоков.

Вариант 1. «Сортировка деталей по размеру».

Часть 1

При включении тумблера X1 включается команда подачи для робота (выход 100.00). При выключении тумблера подача робота прекращается.

При включении тумблера X2 включается конвейер (выход 100.01), а при отключении – останавливается.

Конвейер не включается, пока не отключена подача робота.

Библиотека БГУИР

Большие, средние и мелкие детали сортируются в зависимости от сигналов на выходах датчиков: верхний (тумблер X3), средний (тумблер X4), нижний (тумблер X5). При этом зажигаются соответствующие лампы, сигнализирующие о размерах деталей: большой – выход 100.02, средней – выход 100.03 и малой величины – выход 100.04.

Часть 2

При включении тумблера 1 включается команда подачи для робота (выход 100.00). При выключении тумблера подача робота прекращается.

При включении тумблера 2 включается конвейер (выход 100.01), а при отключении – останавливается.

Конвейер не включается, пока не отключена подача робота.

С помощью потенциометра устанавливается размер детали. При срабатывании датчика (тумблер 3) в ячейку памяти сохраняется размер детали (большая, средняя, малая – шкала потенциометра делится на 3 части). При срабатывании концевого выключателя (тумблер X4) зажигаются соответствующие лампы, сигнализирующие о размерах деталей: большой – выход 100.02, средней – выход 100.03 и малой величины – выход 100.04.

Вариант 2. «Загрузка апельсинов».

Часть 1

При включении переключателя 2 включается конвейер (выход 100.01). Конвейер останавливается при выключении переключателя 2 и при попадании ящика в зону фотодатчика (переключатель 3).

При переключении тумблера 1 осуществляется подача роботом ящика на конвейер.

Конвейер не включается, пока не отключена подача робота.

Апельсины загружаются в ящик и подсчет ведется по сигналам от датчика (переключатель 4). Число апельсинов составляет 5. Как только загрузка выполнена, конвейер снова включается.

Если апельсины не загружаются в течение 10 с после срабатывания датчика, то загорается красная лампочка, сигнализирующая об аварии. Конвейер при этом включить нельзя.

Часть 2

Выполняется аналогично части 1, но количество апельсинов, которое можно погрузить в ящик задается потенциометром (от 1 до 10).

Вариант 3. «Светофор».

Часть 1

Вначале горит красный сигнал в течение n с. Последние 2 с одновременно с ним горит желтый. Затем красный и желтый гаснут и загорается зеленый на

Библиотека БГУИР

k с, причем последние 2 с он моргает с частотой 2 Гц. Зеленый гаснет и цикл повторяется.

Время n и k устанавливаются на входах функционального блока.

Часть 2

Доработать светофор, чтобы он учитывал время суток (добавить еще один функциональный блок). То есть время 6.00–21.00 он работает в стандартном режиме, а в промежутке времени 21.00–6.00 моргает желтый сигнал с частотой Z Гц.

Частота устанавливается на входе функционального блока.

Вариант 4. «Поиск максимума».

Часть 1

Создать функциональный блок, на входах которого можно с помощью переключателей 1, 2, 3, 4, 5 задать числа, которые внутри блока записываются в ячейки памяти (однократное нажатие инкрементирует значение).

Затем нужно найти максимальное из этих чисел.

Часть 2

Немного изменить задание 1. Число задается с помощью потенциометра (от 0 до 10). При включении тумблера 1 число заносится в ячейку памяти (например W1). При повторном включении тумблера 1 новое число заносится в ячейку W2. При следующем включении число заносится в W3 и т. д.

Вариант 5. «Управление вентилятором».

Часть 1

В ячейках памяти (W0, W1, W2) хранятся числа. Создать функциональный блок, который будет вычислять выражение:

W3 = 100 – W02 – (W13 + W0) – выражение реализовать на ST. Если W2 = W3, то включается функциональный блок управления вентилятором:

Тумблер 1 – Пуск. Вентилятор вращается (выход 100.00). Предусмотреть самоблокировку.

Тумблер 2 – Стоп. Вентилятор останавливается.

Часть 2

Тумблерами 1 и 2 задается диапазон чисел (однократное нажатие инкрементирует значение).

При включении тумблера 3 количество часов текущего времени проверяется на вхождение во введенный диапазон. Если попали в диапазон, то загорается зеленая лампочка, если же нет – красная.

Библиотека БГУИР

Вариант 6. «Зона хранения».

Часть 1

Составить программу для зоны хранения, которая представляет систему с двумя конвейерами и зоной временного хранения между ними. Конвейер 1 транспортирует пакеты к зоне хранения. Фотодатчик в конце конвейера 1 рядом с зоной хранения определяет, сколько пакетов доставлено в зону хранения.

Конвейер 2 транспортирует пакеты из зоны временного хранения к погрузочной площадке, где грузовые автомобили забирают пакеты для доставки их клиентам. Фотодатчик в конце конвейера 2 у зоны временного хранения определяет, сколько пакетов покидает зону хранения для отправки на погрузочную площадку. Информационное табло с пятью лампочками показывает уровень заполнения зоны временного хранения. Задание выполнить с помощью функционального блока.

Задание 2

Тумблером 1 задается количество сдвигов светодиода (от 0 до 7). Сдвиг происходит раз в секунду (использовать бит P_1s). Тумблер 2 запускает сдвиг.

Лабораторная работа №4
Работа с частотным преобразователем

Контрольные вопросы

1. Устройство и принцип работы асинхронного двигателя.
2. Способы управления асинхронным двигателем.
3. Назначение частотного преобразователя.
4. Векторное управление асинхронным двигателем.
5. Частотный преобразователь. Электрическая структурная схема.
6. Группы параметров частотного преобразователя.
7. Организация обратной связи в системах позиционирования. Виды датчиков.

Задания

Задание 1. Ручная конфигурация частотного преобразователя.

Тумблер 1 возле частотного преобразователя запускает двигатель в прямом направлении, тумблер 2 – в обратном. Осуществить управление частотой двигателя с помощью потенциометра (без контроллера). Вращение двигателя

Библиотека БГУИР

сопровождается индикацией зеленой лампочки. Если же движок стоит, то горит красная лампочка.

Задание 2. Использование обратной связи.

Четыре переключателя на контроллере устанавливают четыре типа скорости вращения двигателя. Первый переключатель запускает двигатель. С аналогового выхода ПЧ на аналоговый вход контроллера подается сигнал со сведением о текущей частоте. Текущее значение частоты хранится в слове W50 (сигнал обратной связи от ПЧ).

Задание 3. Преобразователь частоты. Повышенный уровень.

Тумблер 1 около контроллера запускает двигатель в прямом направлении, тумблер 2 – в обратном. Управление скоростью осуществляется потенциометром. Организовать часы в контроллере: вести учет времени (количество минут и часов) вращения двигателя в любом направлении. При отключении питания контроллера время не должно сбрасываться.

Если ПЧ отключил свой выход по причине ошибки, то загорается сигнал аварии (красная лампа).

Лабораторная работа №5
Программирование панели оператора

Контрольные вопросы

1. Что такое GUI?
2. Методики разработки пользовательских интерфейсов.
3. Протокол связи между программируемым терминалом и ПЛК.
4. Основы работы с NB-Designer.
5. Типы программируемых терминалов.

Задания

Задание 1

Создать на панели элементы «кнопка» и «индикатор». Индикатор показывает состояние выхода контроллера, который заводится на зеленую лампочку. Устанавливается и сбрасывается он кнопками «Включить» и «Выключить» с панели элементов.

Задание 2

Создать два экрана, представленные на рисунках А.1 и А.2. При нажатии кнопки «Настройки» переходим на другой экран, где в поле NI0 задается коли-

Библиотека БГУИР

чество нажатий (n = 1…10). При нажатии кнопки «Старт» n раз загорается индикатор BL0. В поле ND0 отображается текущее значение счетчика. При нажатии кнопки «Сброс» лампочка гаснет, и счетчик сбрасывается.

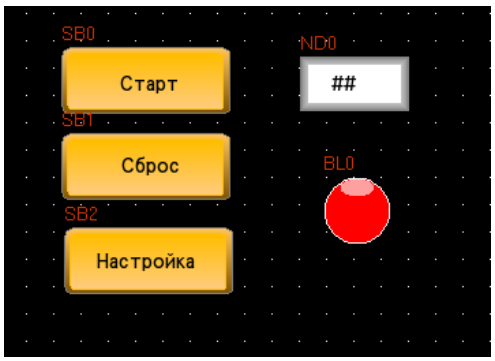


Рисунок А.1 – Основной экран

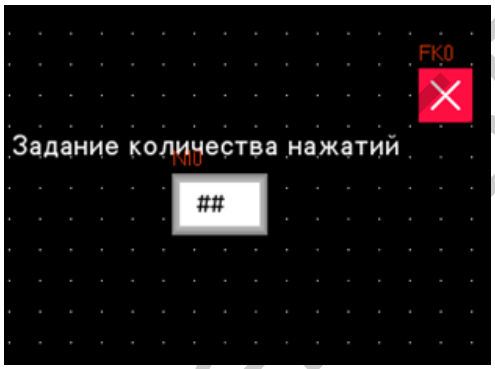


Рисунок А.2 – Экран «Настройки»

Задание 3

Часть 1

Визуализировать в панели проект «Сортировка деталей по размеру» (см. лабораторную работу №3). Добавить кнопки и индикаторы состояния для конвейера и робота, поле для ввода размера детали, индикаторы состояния измерительных датчиков.

Часть 2

Визуализировать в панели проект «Загрузка апельсинов» (см. лабораторную работу №3). Добавить кнопки и индикаторы состояния для конвейера и робота, поле для задания количества апельсинов, поле для отображения текущего количества в ящике. При аварийной ситуации вывести текст с информацией об аварии.

Часть 3

Визуализировать в панели проект «Светофор» (см. лабораторную работу №3). Предусмотреть поле для ввода текущего времени.

Библиотека БГУИР

Часть 4

Визуализировать в панели проект «Поиск максимума». В полях вводятся числа, далее при нажатии кнопки в отдельное поле выводится максимальное число.

Часть 5

Визуализировать в панели проект «Вентилятор». В поля вводятся 3 числа, рассчитывается выражение из лабораторной работы №3 и результат сохраняется в отдельное поле. Если W2 = W3, то кнопка «Управление» становится доступной на панели и при ее нажатии переходим на экран с кнопками управления вентилятором.

Часть 6

Нарисовать на панели 5 индикаторов состояния. В поле для ввода задается количество сдвигов лампочки. При нажатии кнопки происходит сдвиг лампочки каждую секунду. Если лампочка дошла до конца, то происходит сдвиг в обратном направлении.

Лабораторная работа №6
Изучение возможностей сетевого взаимодействия устройств фирмы Omron

Контрольные вопросы

1. Характеристики протокола обмена данных Modbus.
2. Характеристики протокола обмена данных Modbus TCP.
3. Характеристики протокола обмена данных CAN-OPEN.
4. Модель OSI.
5. Принцип организации сети на основе интерфейса RS-485.

Задания

Вариант 1

Источник сигнала задания частоты: ПТ.

Условие запуска регулирования: на панели устанавливаются галочки «готовности» к запуску в прямом и обратном направлении, запуск с тумблера на контроллере (тумблер 1 – прямой ход, тумблер 2 – обратный ход).

Темп разгона/торможения: 4 с.

Сигнал ОС от ПЧ (отображение на панели): текущая скорость.

Библиотека БГУИР

Вариант 2

Источник сигнала задания частоты: внешний потенциометр.

Условие запуска регулирования: пуск с панели либо тумблера (в прямом и обратном направлениях).

Темп разгона/торможения: 6 с.

Сигнал ОС от ПЧ (отображение на панели): входное напряжение.

Вариант 3

Источник сигнала задания частоты: ПТ.

Условие запуска регулирования: кнопки на панели; запуск происходит по таймеру с задержкой 5 с; значение таймера отображается на панели.

Темп разгона/торможения: 8 с.

Сигнал ОС от ПЧ (отображение на панели): выходной ток.

Вариант 4

Источник сигнала задания частоты: потенциометр.

Условие запуска регулирования: кнопки на панели или тумблер; запуск происходит по таймеру с задержкой 3 с.

Темп разгона/торможения: 10 с.

Сигнал ОС от ПЧ (отображение на панели): часы (время наработки двигателя в режиме «Ход»).

Вариант 5

Источник сигнала задания частоты: ПТ.

Условие запуска регулирования: тумблер или кнопка на панели; запуск с тумблера осуществляется по количеству нажатий (3 раза).

Темп разгона/торможения: 12 с.

Сигнал ОС от ПЧ (отображение на панели): выходной ток.

Вариант 6

Источник сигнала задания частоты: ПТ или потенциометр (выбирается галочкой в панели).

Условие запуска регулирования: кнопки на панели.

Темп разгона/торможения: 15 с.

Сигнал ОС от ПЧ (отображение на панели): текущая скорость.

Библиотека БГУИР

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

ОС – обратная связь.

ПЛК – программируемый логический контроллер.

ПТ – программируемый терминал.

ПЧ – частотный преобразователь, электронное устройство для изменения амплитуды и частоты электрического напряжения.

CX-Drive – программное обеспечение для конфигурирования, пусконаладки и обслуживания инверторов и сервоприводов фирмы Omron.

CX-Programmer – среда разработки для программирования контроллеров фирмы Omron.

Ethernet – пакетная технология передачи данных преимущественно локальных компьютерных сетей.

HMI (Human Machine Interface) – человеко-машинный интерфейс – понятие, охватывающее инженерные решения, обеспечивающие взаимодействие человека-оператора с управляемыми им машинами.

NB Designer – среда разработки, которая обеспечивает возможность создания простой HMI для программируемых терминалов Omron.

PLC (Programmable Logic Controller) – программируемый логический контроллер, который является электронной составляющей промышленного контроллера, специализированного (компьютеризированного) устройства, используемого для автоматизации технологических процессов.

Библиотека БГУИР

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Модули ЦПУ CP1H/CP1L. Руководство по программированию. – М., 2007. – 1276 с.
2. SYSMAC CP1L/CP1. Вводное руководство. – М., 2007. – 182 с.
3. Модуль ЦПУ CP1L. Руководство по эксплуатации. – М., 2010. – 829 с.
4. CP1L-EL/EM CPU Unit. Operation Manual. – Tokyo, 2012. – 800 p.
5. MX2. Born to drive machines. Model: 3G3MX2. Quick Start Guide. – Kyoto, 2004. – 48 p.
6. MX2 Компактный преобразователь частоты с векторным управлением. Руководство пользователя. – М., 2007. – 19 с.
7. Inverter MX2/RX/LX Series Drive Programming. Users Manual. – Tokyo, 2010. – 82 p.
8. Преобразователь частоты серии MX2. Инструкция по эксплуатации. – М., 2010. – 372 с.
9. CX-Drive. Operation Manual. – Kyoto, 2007. – 170 p.
10. Digital Indicators K3HB-S/-X/-V/-H User's Manual. – Kyoto, 2004. – 230 p.
11. Цифровые индикаторы серии K3HB. Технические характеристики. – М., 2009. – 327 с.
12. Программируемые терминалы. NB-Designer. Руководство пользователя. – М., 2012. – 568 с.
13. Programmable Terminals. Setup Manual. – Tokyo, 2007. – 286 p.
14. Programmable Terminals. Host connection manual. – Tokyo, 2014. – 206 p.
15. Программируемые терминалы. Вводное руководство. – Tokyo, 2012. – 124 с.

Библиотека БГУИР

Св. план 2017, поз. 24

Учебное издание

Марков Александр Владимирович
Ляхор Тимофей Васильевич

ЭЛЕМЕНТЫ И УСТРОЙСТВА СИСТЕМ УПРАВЛЕНИЯ. ЛАБОРАТОРНЫЙ ПРАКТИКУМ

ПОСОБИЕ

Редактор *М. А. Зайцева*
Корректор *Е. Н. Батурчик*
Компьютерная правка, оригинал-макет *В. М. Задоля*

Подписано в печать 10.05.2018. Формат 60×84 1/16. Бумага офсетная. Гарнитура «Таймс».
Отпечатано на ризографе. Усл. печ. л. 5,0. Уч.-изд. л. 5,1. Тираж 70 экз. Заказ 30.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий №1/238 от 24.03.2014,
№2/113 от 07.04.2014, №3/615 от 07.04.2014.
ЛП №02330/264 от 14.04.2014.
220013, Минск, П. Бровки, 6

Библиотека БГУИР