

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное  
образовательное учреждение  
высшего профессионального образования  
«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ  
ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»**

**СЕНСОРНЫЕ ЭЛЕКТРОННЫЕ  
УСТРОЙСТВА В ЭЛЕКТРОПРИВОДЕ И  
МЕХАТРОНИКЕ**

**Методические указания к курсовой работе  
по дисциплине  
Сенсорные электронные устройства  
мехатронных систем**

**Казань 2020**

УДК 621.316.71:681.3

ББК 31.291

М59

**М59 Микропроцессорные средства в электроприводе и мехатронике:**

Метод. указания к практическим занятиям /

Сост.: Н.К. Андреев, Н.А. Малёв, О.В. Козелков. – Казань: Казан.

гос. энерг. ун-т, 2020. – 39 с.

Приведены описания работы и программирования контроллера, лабораторного стенда с контроллером, а также указания к практическим занятиям по темам: «Программирование ПЛК SIMATIC S7-200», «Реализация 7-сегментной индикации», «Модель «Калькулятор», «Регулятор уровня рабочей жидкости в резервуаре».

Предназначены для студентов, обучающихся по направлениям подготовки 15.04.06 «Мехатроника и робототехника», специализированной подготовки магистра по программе «Приборостроение», специализированной подготовки магистра по программе «Микропроцессорные средства и программное обеспечение измерений».

УДК 621.316.71:681.3

ББК 31.291

## ВВЕДЕНИЕ

Целью дисциплин «Информационно-сенсорные модули объектов мехатроники» (ИСМОМ) и «Сенсорные электронные устройства мехатронных систем» (СЭУМС), является изучение общих закономерностей построения автоматических систем управления (АСУ) электроприводами, мехатронных систем, промышленными установками и технологическими комплексами. В процессе обучения в соответствии с государственным образовательным стандартом у студента должно формироваться умение применять микропроцессорные средства в АСУ промышленных установок и технологических комплексов а также приборостроении.

Изучение данных курсов на кафедре Приборостроение и мехатроника предполагает, что студент ранее освоил технологию программирования микроконтроллеров, основы математической логики, элементы аналоговой и дискретной автоматики, методы анализа и синтеза аналоговых и цифровых регуляторов. Поэтому в настоящем пособии главное внимание уделяется дальнейшему углублению знаний по алгоритмизации и программированию микропроцессорных средств. Основной упор сделан на формирование умений применять программируемые логические контроллеры (ПЛК) в АСУ технологических процессов. Обучение ведется на базе стенда Simatic S7-200 с аппаратными средствами фирмы СИМЕНС, изготовленного на ЗАО «ЭКОИНВЕСТ». Стенд, кроме ПЛК, снабжен имитаторами лифта и перекрестка с соответствующими элементами индикации и датчиков. Кроме того, стенд был дополнен разработанным А.С. Малационом и Н.К. Андреевым регулятором уровня рабочей жидкости в резервуаре. В качестве основного средства программирования используется язык релейно-контактных схем в составе программного пакета STEP 7-Micro/WIN 32.

Пособие включает в себя описание практических занятий. Практические занятия разбиты по темам, которые выполняются в течение восьми часов каждая. Предусматривается не только написание программ, но и сборка электрической схемы какой-либо установки, программирование ее работы, тестирование, отладка программы и контроль правильности работы установки в целом. Целью практических занятий является получение навыков работы с лабораторным комплексом «Программируемые логические контроллеры», ознакомление с языком релейно-контактных схем (LAD) контроллера.

## **ТЕМА № 1**

### **ПРОГРАММИРОВАНИЕ ПЛК SIMATIC S7-200**

#### **Цель работы**

Получение базовых навыков работы с аппаратной и программной частью лабораторного комплекса, предназначенного для автоматизации автономных объектов управления на базе программируемого логического контроллера (ПЛК) SIMATIC S7-200 фирмы SIEMENS.

#### **Задание на работу**

Требуется выполнить подключение кнопок, датчиков, исполнительных механизмов к контроллеру, а также разработать и записать в память ПЛК простейшие программы управления согласно приведенному ниже плану.

1. Реализовать программу проверки логических схем «И», «ИЛИ» с использованием двух кнопок. В случае единичного состояния должна загораться лампа. Использовать панель с ПЛК и с 7-сегментным индикатором.

2. Реализовать программу включения и отключения объекта управления (ОУ) с помощью кнопок ПУСК и СТОП. Предусмотреть автоматическое отключение ОУ при срабатывании датчика. Использовать панель с ПЛК и ЛИФТОМ.

3. Реализовать программу отключения ОУ с использованием таймера. При нажатии кнопки ПУСК лампа должна гореть 5 секунд, после чего отключиться.

#### **Порядок выполнения работы**

1. Ознакомиться с устройством, конструктивным исполнением, характеристиками и областью применения ПЛК SIMATIC S7-200. Обратит особое внимание на назначение каждого модуля контроллера, индикаторов и переключателя выбора режима работы ПЛК.

2. Изучить основные элементы языка программирования LAD, принятые в среде программирования STEP 7-Micro WIN 32.

3. Составить список входов и выходов в соответствии с вариантом программы и описанием алгоритма работы.

4. Разработать рабочую программу на языке программирования LAD и набрать её в среде STEP 7.

5. Выполнить подключение ПЛК к объекту управления согласно списку входов и выходов.

6. Проверить программу на наличие синтаксических и иных ошибок, произвести компиляцию рабочей программы в машинный код.

7. При отсутствии ошибок в программе и схеме подключения включить питание системы, перевести переключатель режима работы контроллера в положение TERM и осуществить программирование ПЛК.

8. Запустить исполнение программы контроллером через ПК. Проверить её работу на стенде. В случае ошибок необходимо остановить исполнение программы, исправить недочеты и заново загрузить программу в память ПЛК.

9. Выполнить все задания на лабораторную работу, показать результаты преподавателю.

10. Написать отчет о проделанной работе.

### **Содержание отчёта**

Отчёт по работе должен содержать: титульный лист, цели работы, задания на работу, краткий теоретический обзор, необходимый для выполнения работы, таблицы входов и выходов, листинг программ с пояснением их работы, выводы по проделанной работе.

### **Общая характеристика и область применения системы SIMATIC S7-200**

Система SIMATIC S7-200 представляет собой аппаратно-программный комплекс, предназначенный для автоматизации автономных объектов управления, а также для работы в составе децентрализованных систем распределенного управления.

В состав комплекса (рис. 1.1) входят: программируемый логический контроллер (ПЛК) S7-200, программное обеспечение STEP 7-Micro/WIN 32, с помощью которого контроллер программируется на персональном компьютере, набор реальных и виртуальных моделей объектов автоматизации.

Контроллеры S7-200 относятся к микромощной линии ПЛК SIMATIC S7 и могут применяться в тех областях, в которых использование более мощных ПЛК ранее было нецелесообразным вследствие высокой цены и больших габаритов. ПЛК S7-200 предназначены для решения задач управления в составе небольших систем автоматизации.

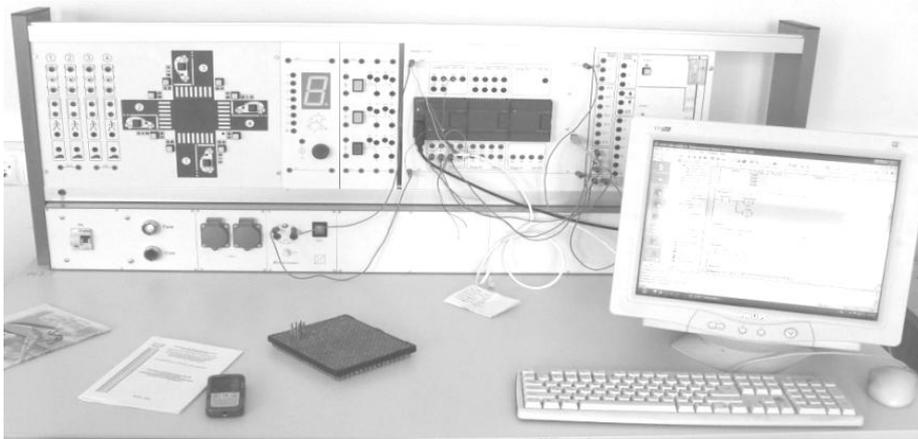


Рис. 1.1. Лабораторный комплекс с ПЛК S7-200

В состав аппаратных средств ПЛК (рис. 1.2) входит ЦПУ (CPU) – центральное процессорное устройство (2) и один или несколько модулей расширения (3), используемых в тех случаях, когда необходимо увеличить число точек ввода/вывода или обеспечить подключение аналоговых датчиков, средств связи для создания промышленной сети и сенсорных панелей человеко-машинного интерфейса. Функциональные блоки системы устанавливаются на монтажной шине DIN (3).

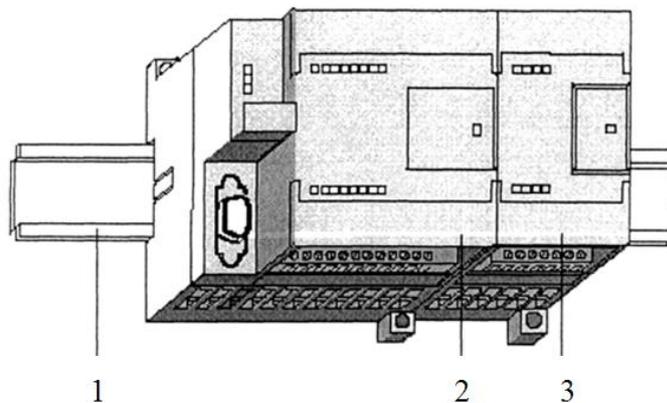


Рис. 1.2. Компоновка блоков ПЛК S7-200 на шине DIN

Центральный процессор (ЦП) представляет собой автономное устройство. Он выполняет программу пользователя и хранит данные, необходимые для управления процессом. Наличие последовательного коммуникационного порта позволяет подключать к ЦПУ программатор или другие устройства информационного обмена (компьютер, текстовый дисплей и др.).

## Устройство и конструктивное исполнение ПЛК

Контроллеры серии S7-22x представляют собой базовый блок, в состав которого входят центральный процессор, устройства памяти, модуль питания и монтажные колодки дискретных входов/выходов с винтовыми клеммами для подключения датчиков и нагрузок.

Внутренний модуль питания базового блока CPU 22x включает два источника постоянного тока: первый работает с напряжением 5 В и обеспечивает питание аппаратной логики, второй рассчитан на напряжение 24 В и служит для питания цепей датчиков. Все доступные пользователю элементы базового блока расположены на его передней панели.

Все элементы управления контроллера закрыты защитными крышками (рис. 1.3).

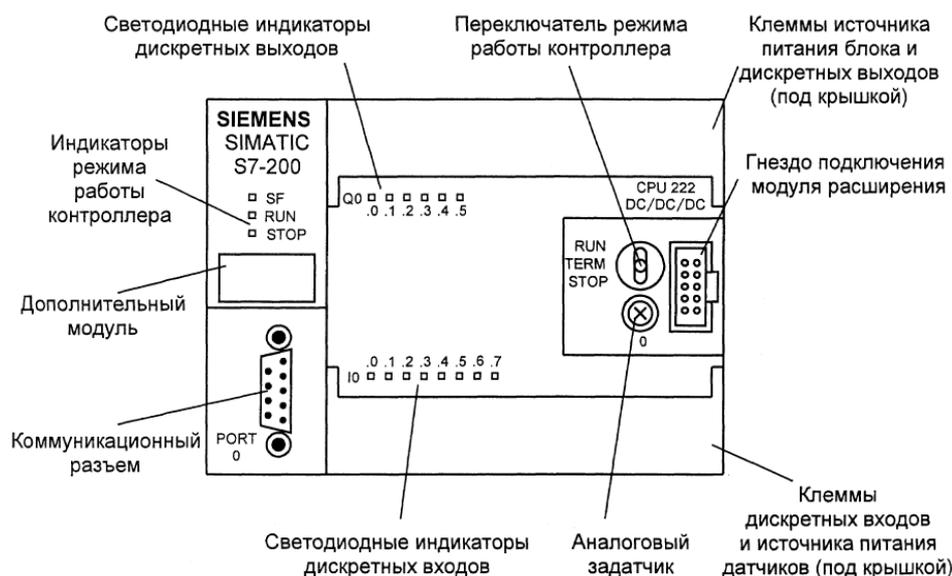


Рис. 1.3. Расположение органов управления и сигнализации на базовом блоке CPU S7-222

Под верхней крышкой размещаются винтовые клеммы дискретных выходов контроллера и контактов внешнего источника питания контроллера. Под нижней крышкой – винтовые клеммы дискретных входов контроллера и источника питания цепей датчиков. С правой стороны базового блока имеется передняя крышка, закрывающая гнездо магистрального порта (предназначен для подключения внешнего модуля расширения), переключатель режима работы контроллера (табл. 1.1) и потенциометр аналогового задатчика.

Переключатель режима работы контроллера позволяет выбрать один из возможных режимов работы ПЛК. Текущий режим работы контроллера индицируется светодиодными индикаторами.

Для подключения модуля расширения входов/выходов предназначен специальный разъем. Каждый модуль имеет гибкий шлейф с многоконтактной вилкой, которая должна быть вставлена в этот разъем.

С левой стороны базового блока установлено гнездо для вставки дополнительного модуля (картриджа памяти или картриджа батареи) в определенных конфигурациях контроллера. Картридж памяти используется либо для длительного хранения программы пользователя, либо для задания новой программы. Картридж батареи может использоваться для увеличения времени хранения данных сверх срока. С левой стороны блока размещается 9-контактный разъем коммуникационного порта, позволяющий реализовать неизолированный интерфейс RS-485. При этом скорость передачи информации при программировании (режим PPI) составляет 9600 бод.

Таблица 1.1

## Режимы работы контроллера

Положение переключателя	Описание
RUN	Контроллер работает в режиме выполнения программы. Переключение с режима «RUN» на режим «STOP» может быть в случае возникновения ошибки в процессе работы
TERM	При нахождении переключателя в положении «TERM» можно программным путем устанавливать режимы работы контроллера «RUN» и «STOP»
STOP	ПЛК прекращает исполнение программы и переходит в состояние останова. При редактировании прикладной программы и загрузке ее в память контроллер должен находиться в режиме работы «STOP»

При подключении к порту RS-232 стандартного компьютера используется кабель для программирования PC/PPI с конвертером. При этом программаторы SIMATIC подключаются к порту контроллера напрямую посредством кабеля, входящего в комплект их поставки.

В качестве элементов сигнализации в моделях CPU 22x используются светодиодные индикаторы. Выделяют три группы подобных индикаторов: индикаторы состояния, индикаторы входов, индикаторы выходов (табл. 1.2).

Светодиодные индикаторы состояния ПЛК

Индикатор	Цвет	Описание
SF	красный	индицирует системную неисправность
RUN	зеленый	ПЛК в режиме исполнения программы пользователя
STOP	желтый	ПЛК в режиме останова (исполнение прикладной программы прервано)
IX.X	зеленый	означает наличие логического сигнала от объекта управления
QX.X	зеленый	означает наличие логических сигналов на выходах ПЛК

### Общие сведения о программировании ПЛК

При разработке программ для ПЛК используют специализированные, проблемно-ориентированные языки, понятные и доступные специалистам в области автоматике. Для работы над проектами автоматизации с применением программируемых контроллеров SIMATIC S7-22x фирмой «Siemens» разработано специальное программное обеспечение STEP 7-Micro/WIN 32.

Международной электротехнической комиссией (IEC) разработан стандарт, охватывающий многие аспекты программирования ПЛК. Рекомендованная IEC система команд получила название IEC 1131-3. Фирма «Siemens» входит в ряд производителей ПЛК, поддерживающих систему команд IEC 1131-3, поэтому в пакете STEP 7-Micro/WIN32 предусмотрены средства разработки программ в соответствии со стандартом IEC 1131-3. Однако основной системой команд для ЦПУ S7-200 является система команд SIMATIC, для работы с которой в пакете предусмотрены редакторы языков программирования STL, LAD и FBD. Студентам предлагается изучить язык LAD ввиду его простоты и сходства с релейно-контактными схемами, принятыми в автоматике.

Язык LAD (Ladder Diagram, «лестничная логика») является графическим языком. Поэтому при написании программы на этом языке пользователь создает ее фрагменты в форме вычислительных логических цепочек, используя графические элементы языка (рис. 1.4).

При создании LAD-программы используются следующие базовые графические элементы, представляющие собой:

- контакт – ключ, пропускающий ток в замкнутом состоянии;

- катушка – реле, срабатывающее при прохождении тока через обмотку;
- бокс (ящик) – функция, исполняющаяся при активизации данного элемента;
- вычислительные цепочки – завершенная цепь, ток в которой проходит от шины питания через замкнутые контакты и активизирует катушки или боксы.

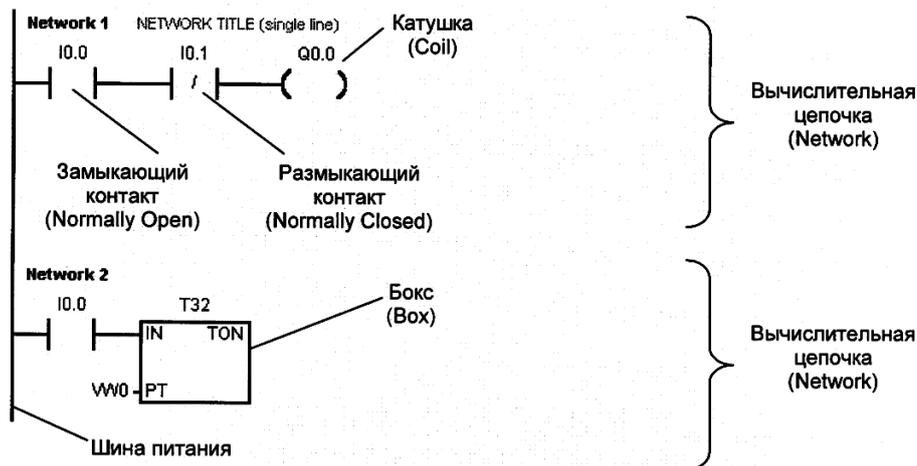


Рис. 1.4. Базовые элементы языка LAD

Контакты в LAD-программе соответствуют операндам, над которыми выполняются логические вычисления. При этом выполняемая логическая функция определяется характером соединения контактов: последовательному соединению соответствует функция логического умножения (And), параллельному – функция логического сложения (Or). Операция присваивания результата логических вычислений данной дискретной величине обозначается катушкой.

Вид вычислительной цепочки в языке LAD напоминает обычную релейно-контактную схему, что упрощает реализацию управляющих функций, заданных логическими уравнениями.

Наглядность языка LAD делает его предпочтительным при переходе от логики релейно-контактных схем к ее программной реализации при модернизации оборудования, а также при разработке новых прикладных программ, особенно на начальном этапе освоения программирования ПЛК. Рассмотрим общую структуру программы, принятую в пакете STEP 7. Прикладная (пользовательская) программа может включать три части: основную программу, подпрограмму и программу обработки прерываний.

**Основная часть программы (Main Program)** содержит инструкции, которые управляют приложением. Инструкции основной программы выполняются последовательно один раз в течение одного цикла

сканирования ЦПУ начиная с инструкций первой вычислительной цепочки. Заканчивается цикл инструкцией окончания основной программы, после исполнения которой процессор переходит к новому циклу сканирования.

**Подпрограммы (Subroutines)** являются необязательными элементами программы пользователя. Они исполняются только по вызову из основной программы. Обычно в подпрограммы выносят повторяющиеся последовательности инструкций, реализующие некоторую законченную функцию (например, формирование заданного временного промежутка, обновление уставок и т.д.). Применение подпрограмм позволяет сократить объем прикладной программы в целом, а также использовать готовые подпрограммы при разработке новых проектов.

**Программы обработки прерываний (Interrupt Routine)** также являются необязательными. Они исполняются каждый раз при наступлении некоторых событий. При этом выполнение основной программы прерывается и ее состояние запоминается. По окончании работы программы обработки прерывания происходит возврат в основную программу, и далее её выполнение возобновляется с прерванного места. Прикладная программа для ПЛК SIMATIC S7-200 создается в виде отдельных структурных блоков проекта, включающих вышеуказанные части программы. Редактирование блоков производится независимо друг от друга, а связь между ними в проекте устанавливается соответствующими инструкциями.

### **Среда разработки STEP 7-Micro/WIN 32**

При запуске программы STEP 7-Micro/WIN 32 (рис. 1.5) на экран монитора выводится Главное окно. Оно содержит строку заголовка с наименованием проекта и кнопками управления окном.

В Главном окне доступны следующие панели:

– панель Главного меню – все функции STEP 7-Micro/WIN 32 доступны через пункты этого меню с помощью системы дополнительных раскрывающихся подменю;

– стандартная Панель инструментов – расположена под панелью главного меню и содержит множество часто используемых при разработке и тестировании программы функций. Большинство этих функций включены в состав аналогичных панелей всех Windows-приложений. Стандартная панель инструментов STEP 7-Micro/WIN 32 также содержит кнопки-пиктограммы, дублирующие ряд функций главного меню;

– панель инструментов Редактора располагается непосредственно под панелью стандартных инструментов, включает набор инструментов для работы с различными редакторами языков программирования.

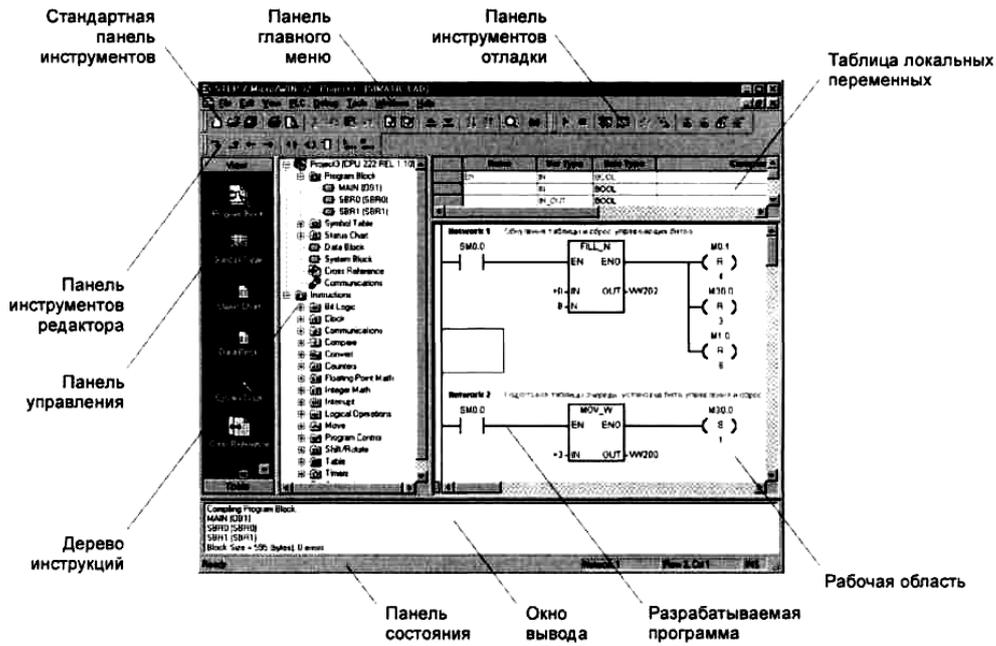


Рис. 1.5. Вид главного окна пакета программирования STEP 7

- панель инструментов Отладки включает кнопки пуска и останова программы, мониторинга значений переменных, а также ряд кнопок специальных функций отладки;
- панель Управления содержит две группы кнопок, открывающих доступ к различным частям проекта;
- панель Состояния – на ней у нижней границы главного окна отображается состояние операций, выполняемых пользователем в STEP 7-Micro WIN 32. В зависимости от режима работы на панели Состояния может отображаться либо информация, связанная с редактированием (режим редактирования), либо информация о состоянии процесса и контроллера при работе в режиме реального времени (online);
- рабочая область – предназначена для ввода инструкций разрабатываемой программы;
- таблица локальных переменных – выводится только при работе с подпрограммами и программами обработки прерываний, содержит значения локальных переменных.

## Правила построения LAD-программы

Каждая вычислительная цепочка LAD-программы состоит из графических элементов, соответствующих определенным инструкциям машинного языка, – контактов, катушек и боксов. Вместе с тем процедура создания вычислительных цепочек регламентируется определенными правилами размещения в них графических элементов языка:

– каждая вычислительная цепочка должна начинаться контактом, но нельзя заканчивать цепочку контактом;

– катушка служит для завершения логической цепочки и не может находиться в ее начале; отдельно взятая вычислительная цепочка может иметь несколько катушек на параллельных участках; последовательное соединение катушек не допускается;

– боксы, используемые для выполнения определенных функций, могут располагаться в вычислительной цепочке последовательно, один за другим, если они имеют выход ENO; к этому выходу могут также подключаться дальнейшие инструкции цепочки; бокс, не имеющий выхода ENO, заканчивает вычислительную цепочку.

Любую вычислительную цепочку LAD-программы можно представить как сетку, состоящую из ячеек, в каждой из которых может быть размещена инструкция, значение параметра или участок линии. Сетка имеет 32 ячейки по горизонтали и 32 ячейки по вертикали; данные размеры определяют предельное количество элементов вычислительной цепочки.

### Основные элементы языка LAD

**Нормально открытый контакт** (рис. 1.6). Будет замыкаться при состоянии бита, указанного в качестве адреса, равного 1. Если состояние сигнала по указанному адресу равно 0, то контакт разомкнут. В случае подключения датчика или кнопки к входам ПЛК, над контактом указывается адрес входа, например I0.1 означает, что задействован **вход 1**, базового модуля ЦПУ (обозначается 0). Область памяти, которая может быть использована для адреса: I, Q, M, L, D, T, C.

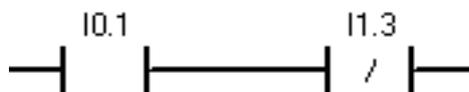


Рис. 1.6. Нормально открытый и нормально закрытый контакты

**Нормально закрытый контакт** (рис. 1.6). Если состояние сигнала по указанному адресу равно 0, то контакт замкнут. Контакт разомкнется при состоянии бита, указанного в качестве адреса, равного 1. Адрес входа I1.3 означает, что задействован **вход 3** первого модуля расширения. Область памяти, которая может быть использована для адреса, аналогична нормально открытому контакту.

**Выходная катушка** (рис. 1.7). Работает как катушка в цепи управления релейно-контактной схемы.

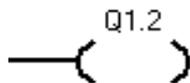


Рис. 7. Выходная катушка

Если к катушке подводится ток, то бит адреса устанавливается в 1. Выходную катушку можно установить только на правом конце логической цепи. Разрешается использование нескольких параллельно соединенных катушек в одной логической цепочке. Область памяти адреса: I, Q, M, L, D.

**Катушки установки и сброса бита** (рис. 1.8). Инструкция установить бит (S) выполняется только тогда, когда результат предыдущей инструкции равен 1. После установки бита указанного адреса в 1 результат останется неизменным до тех пор, пока не будет применена инструкция сброс бита (R).

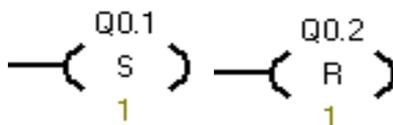


Рис. 1.8. Катушки установки и сброса бита

При протекании тока через катушку сброса бита указанный над катушкой адрес бита сбрасывается в 0. Область памяти адреса может быть: I, Q, M, L, D, T, C.

**Таймер** (рис. 1.9). Инструкция таймера является выходной функцией вычислительной цепочки. Таймеры формируют заданные интервалы времени путем подсчёта определенного количества фиксированных задержек времени – дискрет. При задании таймера в программе должны быть определены значение дискретности и количество дискрет (уставка), определяющих требуемый интервал времени.

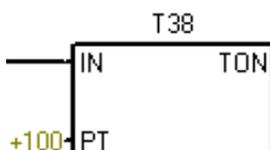


Рис. 1.9. Таймер с задержкой включения

Приняты следующие обозначения: Txx – номер таймера (адрес бита), TON – обозначение типа таймера, PT – значение уставки, IN – вход, по которому начинается отсчёт. Таймер с задержкой включения (TON) начинает отсчёт времени, когда его вход принимает значение 1, и продолжает счет, пока

на входе сохраняется единичное значение. Счет заканчивается по достижении значения  $PT$ , и одновременно устанавливается бит таймера. Сброс таймера происходит при сбросе входа. Номер таймера определяет дискрету времени. Т33–Т36 имеют дискрету 10 мс, Т101–Т255 дискрету 100 мс, Т32 и Т96–дискрету 1 мс. Требуемое значение задержки определяется по формуле:

$$T = PT\Delta t, \quad (1)$$

где  $PT$  – значение уставки,  $\Delta t$  – значение дискреты (зависит от номера таймера).

### Пояснения к выполнению заданий

**Задание 1.** Из ранее пройденных курсов известно, что логической схеме «И» соответствует последовательное соединение контактов, а схеме «ИЛИ» – параллельное. Для проверки достаточно использовать в качестве входных переменных две кнопки, а в качестве выходного значения – лампу. Горящая лампа будет соответствовать значению «1», а не горящая «0».

К панели с 7-сегментным индикатором следует подвести напряжение 24 В, соблюдая полярность. От клеммы +24 В соединить проводом верхние клеммы двух нормально открытых контактов. Нижние же их клеммы подключить к входам ПЛК. Выход ПЛК соединить проводом с клеммой лампы. Далее нужно написать программу на языке LAD, используя те адреса, к которым были подключены кнопки и лампа, загрузить программу в память ПЛК и проверить результаты.

**Задание 2.** Существует несколько способов программной реализации включения и отключения объекта управления с помощью кнопок и датчиков. Первый связан с использованием обычной выходной катушки и подключением дополнительного нормально открытого контакта с таким же адресом, как у выходной катушки, параллельно контакту «Пуск» (рис. 1.10).

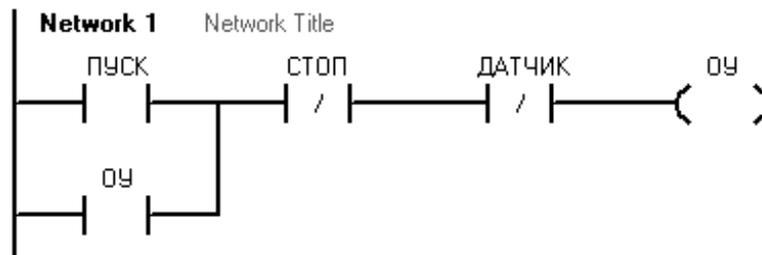


Рис. 1.10. Листинг программы

Такую схему принято называть схемой с самоподхватом. Пользователю нет необходимости держать постоянно кнопку «ПУСК». Отключение ОУ произойдет, если будет нажата кнопка «СТОП» или сработает датчик.

Задачу также можно решить с помощью катушек установки и сброса бита (рис. 1.11). При нажатии кнопки «ПУСК» выходная катушка принимает единичное значение. ОУ будет работать до тех пор, пока не будет выполнена логическая цепочка сброса бита (R) выходной катушки.

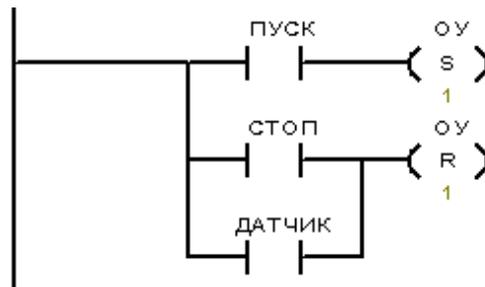


Рис. 1.11. Листинг программы

**Задание 3.** Для успешного выполнения задания необходимо выбрать номер таймера и по формуле (1) посчитать значение уставки времени РТ. Первая вычислительная цепочка будет аналогична программе на рис. 10, за исключением того, что на месте нормально закрытого контакта датчика будет такой же контакт с адресом таймера. Вторая цепочка будет состоять из таймера с выбранным ранее номером и уставкой. Таймер начнёт отсчёт после замыкания нормально открытого контакта (с адресом выходной катушки, подключённой ко входу IN) первой вычислительной цепочки.

### Контрольные вопросы

1. Назначение и область применения системы SIMATIC S7-200.
2. Опишите устройство и конструктивное исполнение ПЛК S7-200.
3. Какие языки программирования используются при написании программ для ПЛК?
4. В чём удобство языка LAD?
5. Из каких частей состоит прикладная программа?
6. Перечислите основные элементы языка LAD и правила создания вычислительных цепочек.
7. Какими способами можно реализовать включение и автоматическое отключение объекта управления? В чём их сходство и отличие?
8. Как посчитать уставку времени для таймера? На что влияет номер таймера?

## **ТЕМА № 2**

### **РЕАЛИЗАЦИЯ 7-СЕГМЕНТНОЙ ИНДИКАЦИИ**

#### **Цель работы**

Реализация программы вывода на 7-сегментном индикаторе номера нажатой кнопки с последующим подсчётом количества нажатий, на базе программируемого логического контроллера (ПЛК) SIMATIC S7-200 фирмы SIEMENS.

#### **Задание на работу**

Требуется выполнить подключение кнопок и 7-сегментного индикатора к контроллеру, а также разработать и записать в память ПЛК управляющую программу, работающую по следующему алгоритму.

1. При нажатии на кнопку «1» на 7-сегментном индикаторе должна высветиться цифра 1 и продолжать светиться до тех пор, пока не будет нажата другая кнопка. Аналогично с кнопкой «2» и «3». При одновременном нажатии нескольких кнопок система не должна никак реагировать.

2. При 8-кратном нажатии кнопки «3» на индикаторе должна высветиться цифра 8.

3. При нажатии кнопки «Сброс» на индикаторе не должен светиться ни один сегмент.

#### **Порядок выполнения работы**

1. Изучить инструкции SEG, MOV\_B, STU языка программирования LAD, принятые в среде программирования STEP 7-Micro/WIN 32, необходимые для выполнения лабораторной работы.

2. Составить список входов и выходов в соответствии с заданием на лабораторную работу.

3. Разработать рабочую программу на языке программирования LAD и набрать её в среде STEP 7.

4. Выполнить подключение ПЛК к объекту управления согласно списку входов и выходов.

5. Проверить программу на наличие синтаксических и иных ошибок, произвести компиляцию рабочей программы в машинный код.

6. При отсутствии ошибок в программе и схеме подключения включить питание системы, перевести переключатель режима работы контроллера в положение TERM и осуществить программирование ПЛК.

7. Запустить исполнение программы контроллером через ПК. Проверить её работу на стенде. При наличии ошибок необходимо остановить исполнение программы, исправить недочеты и заново загрузить программу в память ПЛК.

8. Выполнить все пункты задания на лабораторную работу, показать результаты преподавателю.

9. Написать отчет о проделанной работе.

### Содержание отчёта

Отчёт по работе должен содержать: цели работы, задания на работу, краткий теоретический обзор, необходимый для выполнения работы, таблицу входов и выходов, листинг программы с пояснением её работы, выводы по проделанной работе.

### Инструкции языка LAD

**Инструкция SEG.** Символ на 7-сегментном индикаторе формируется отдельными сегментами (рис. 2.1, а). Поэтому для вывода на индикатор определенной цифры нужно соединить выходы его сегментов с цифровыми выходами контроллера и записать логические единицы на выходы, соединенные с выводами сегментов, которые должны светиться, оставив на всех остальных выходах нулевые значения. При этом должен соблюдаться определенный порядок подключения индикатора к выходам контроллера (рис. 2.1, б).

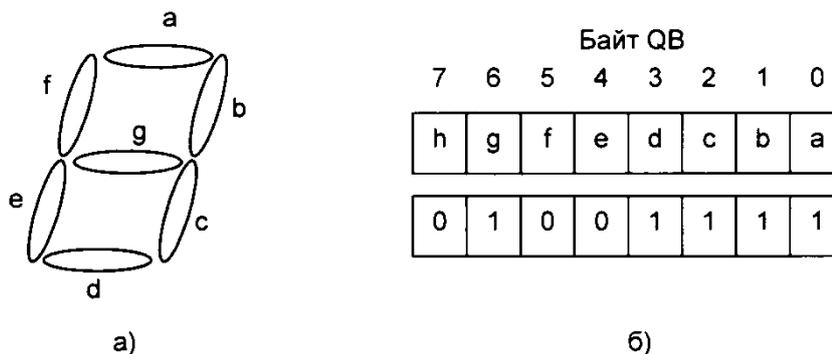


Рис. 2.1. Формирование символа на 7-сегментном индикаторе:  
а) матрица индикатора; б) кодирование символа

Если необходимо вывести на индикаторе цифру от 0 до 9, проще всего использовать специальную инструкцию SEG (рис. 2.2). В инструкции приняты следующие обозначения: EN – вход, по которому активируется

инструкция, IN – вход, по которому записывается в десятичном формате число для отображения, OUT – выход для записи адреса контроллера, ENO – дополнительный вывод для подключения новых инструкций.

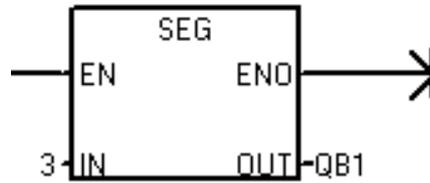


Рис. 2.2. Инструкция SEG

Для правильной работы инструкции необходимо записать число 0–9 по входу IN и адрес выводов контроллера, например QB1, по выходу OUT.

**Инструкция MOV\_B.** Иногда при работе над программой возникает задача вывести на индикатор нестандартный символ или букву русского или латинского алфавита (рис. 2.3).

Символ	Сегменты дисплея	Выходной код		Символ	Сегменты дисплея	Выходной код
0		0011 1111		8		0111 1111
1		0000 0110		9		0110 0111
2		0101 1011		A		0111 0111
3		0100 1111		B		0111 1100
4		0110 0110		C		0011 1001
5		0110 1101		D		0101 1110
6		0111 1101		E		0111 1001
7		0000 0111		F		0111 0001

Рис. 2.3. Коды формирования HEX-символов на индикаторе

Для вывода на индикатор, например буквы F, в байт выхода контроллера следует записать код 0111 0001 (или – в шестнадцатеричной форме – 71). Для этой цели подходит инструкция MOV\_B (рис. 2.4). Отличие от инструкции SEG заключается в синтаксисе задания отображаемой цифры по входу IN. Необходимо набирать 16#«код в шестнадцатеричной форме».

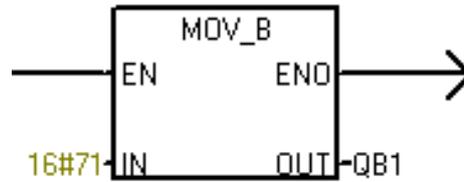


Рис. 2.4. Инструкция MOV\_B

**Алгоритм получения кода произвольного символа:**

- 1) записать в обратном порядке обозначение сегментов индикатора: h g f e d c b a;
- 2) под каждой буквой записать «1», если этот сегмент должен светиться, «0» – если сегмент гореть не должен;
- 3) получившийся двоичный код перевести в шестнадцатеричный формат;
- 4) результат записать по входу IN инструкции MOV\_B.

**Инструкция суммирующего счётчика CTU.** Суммирующий счётчик (рис. 2.5) инкрементирует свои показания (начиная с 0) каждый раз, когда вход счёта по возрастанию CU изменяет своё состояние из 0 в 1. По достижении текущим значением уставки PV счётчик прекращает работу и устанавливает свой бит Sxxx.

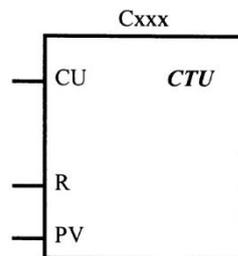


Рис. 2.5. Инструкция суммирующего счётчика

Счётчик сбрасывается (т.е. обнуляется его битовое и текущее значения), когда вход сброса R переключается в состоянии 1 или когда выполняется инструкция сброса Reset для данного счётчика. Максимальное значение суммирующего счётчика, по достижении которого происходит его останов, равно 32767.

**Пояснения к выполнению заданий**

Программу можно условно поделить на три части: вывод номера нажатой клавиши на индикатор, подсчёт количества нажатий и очистка индикатора от изображений.

Первая часть программы работает с использованием инструкции SEG, по входу IN которой записываются значения 1, 2 или 3 в зависимости от номера нажатой кнопки. Для предотвращения одновременного нажатия нескольких кнопок целесообразно включить в вычислительную цепочку последовательно соединённые нормально закрытые контакты кнопок, которые нажимать не следует.

Подсчёт количества нажатий легко осуществить с помощью суммирующего счётчика, в качестве уставки у которого будет стоять цифра 8. При установке бита счётчика Sxxx в «1» нормально открытый контакт с адресом Sxxx замкнётся и сигнал поступит на вход инструкции SEG.

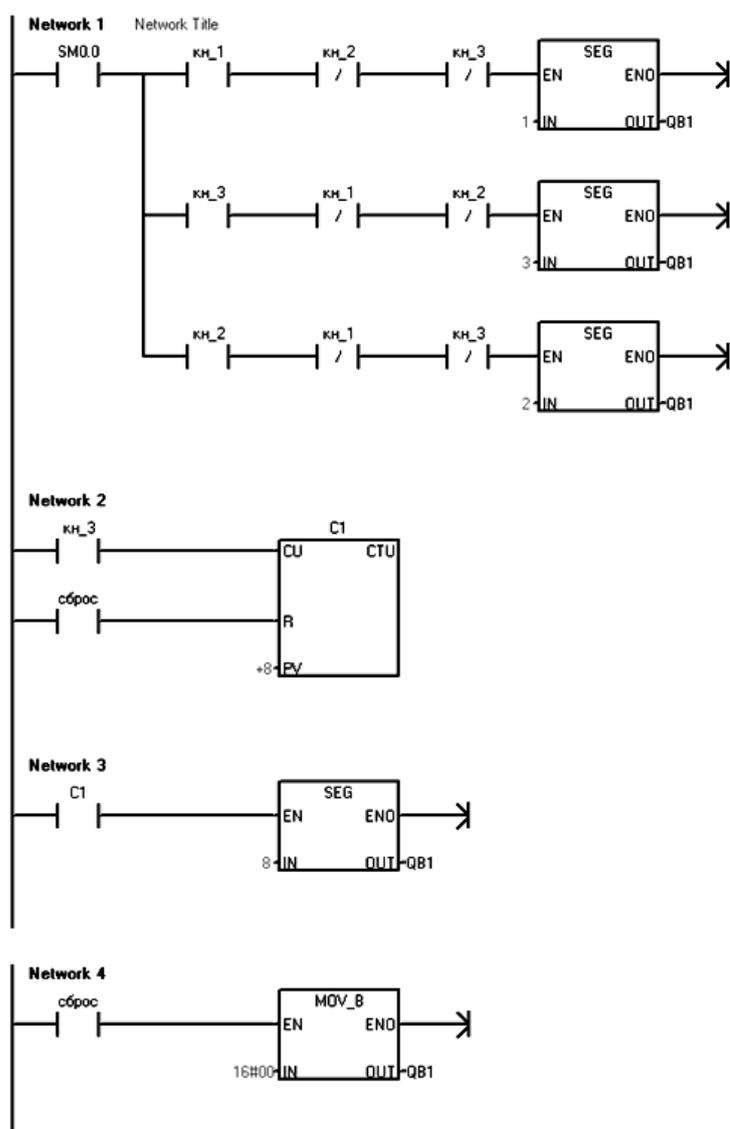


Рис. 2.6. Листинг программы

Сброс счётчика и отчистка индикатора осуществляется с помощью последовательно установленных нормально открытого контакта кнопки «Сброс» и инструкции MOV\_B.

### Контрольные вопросы

1. Какие инструкции 7-сегментной индикации вы изучили?
2. В чём особенности применения этих инструкций?
3. Изложите последовательность действий при получении кода для произвольного символа.
4. Как работает суммирующий счётчик?
5. Какому символу будет соответствовать код 16#39?
6. По какому входу сбрасывается счётчик?
7. Возможно ли присвоить счётчику номер 256?

## **ТЕМА № 3**

### **МОДЕЛЬ «КАЛЬКУЛЯТОР»**

#### **Цель работы**

Реализация программы модели калькулятора с использованием изученных ранее элементов языка LAD на базе ПЛК SIMATIC S7-200 фирмы SIEMENS.

#### **Задание на работу**

Требуется разработать алгоритм и рабочую программу «Калькулятор», работа которой аналогична обычному калькулятору. При этом используются три цифры (1, 2, 3) и операции сложения или вычитания. При нажатии первой цифры, кнопки операции и второй цифры на индикаторе высвечивается получившийся результат. Так как используемые цифры однозначные, то результат необходимо высвечивать сразу после нажатия второй цифры. Необходимо предусмотреть кнопку сброса, при нажатии на которую осуществляется сброс введенных цифр и очистка индикатора.

#### **Порядок выполнения работы**

1. Составить список входов и выходов в соответствии с заданием на лабораторную работу.
2. Разработать рабочую программу на языке программирования LAD и набрать её в среде STEP 7.
3. Выполнить подключение ПЛК к объекту управления согласно списку входов и выходов.
4. Проверить программу на наличие синтаксических и иных ошибок, произвести компиляцию рабочей программы в машинный код.
5. При отсутствии ошибок в программе и схеме подключения включить питание системы, перевести переключатель режима работы контроллера в положение TERM и осуществить программирование ПЛК.
6. Запустить исполнение программы контроллером через ПК. Проверить её работу на стенде. В случае ошибок необходимо остановить исполнение программы, исправить недочеты и заново загрузить программу в память ПЛК.
7. Выполнить все пункты задания на лабораторную работу, показать результаты преподавателю.
8. Написать отчет о проделанной работе.

## Содержание отчёта

Отчёт по работе должен содержать: цели работы, задания на работу, краткий теоретический обзор, необходимый для выполнения работы, таблицу входов и выходов, листинг программы с пояснением её работы, выводы по проделанной работе.

### Пояснения к выполнению задания

Для выполнения задания необходимо использовать стенд «Лифт». На нем есть необходимое количество кнопок и 7-сегментный индикатор. После подключения соединительных проводов к контроллеру подаются напряжения на стенд. При нажатии кнопки на стенде на соответствующий разъем подается напряжение, а с данного разъема с помощью соединительного провода напряжение подается на входной порт контроллера.

Перед составлением алгоритма работы программы необходимо составить список входов и выходов. Входными переменными будут цифры и вычислительные операции, а выходной переменной будут байт, формирующий число на 7-сегментном индикаторе. Затем необходимо сопоставить в соответствии со списком входов и выходов память данных входных и выходных переменных. Данный список следует занести в таблицу символов среды разработки STEP 7-Micro/Win32. Таким образом, каждой переменной будет соответствовать определенная область памяти. В процессе разработки рабочей программы «Калькулятор» на языке LAD нет необходимости вручную писать адрес переменных, достаточно, нажав правой кнопкой мыши на поле адреса, выбрать из списка нужную переменную. Данные символьные переменные берутся из таблицы символов Symbol Table. Вычислительная операция, осуществляемая контроллером, может не являться вычислительной в полном смысле этого слова. То есть, можно задать такой алгоритм, при котором число будет высвечиваться на 7-сегментном индикаторе как результат определенной последовательности нажатий клавиш. Для вывода чисел на 7-сегментном индикаторе следует использовать инструкцию MOV\_B или SEG. Если же надо зафиксировать двукратное нажатие клавиши (например, при осуществлении операции «1+1» или «2+2»), то можно использовать суммирующий счетчик STU.

Нажатием клавиши Compile All производится компиляция и проверка на ошибки. В случае возникновения ошибок количество ошибок указывается в самой нижней строке среды разработки. После устранения всех ошибок нажимается кнопка Download, которая осуществляет программирование контроллера. Подключив соединительные провода к контроллеру,

необходимо проверить разработанную программу на стенде. Необходимо убедиться, что при нажатии соответствующих кнопок загораются определенные светодиоды на контроллере, аналогично – посмотреть состояние светодиодов, отвечающих за выходные порты контроллера. При наличии несоответствий исходному заданию нажать кнопку программного статуса и в режиме реального времени, нажимая на кнопки стенда, найти ошибки в программе и исправить их.

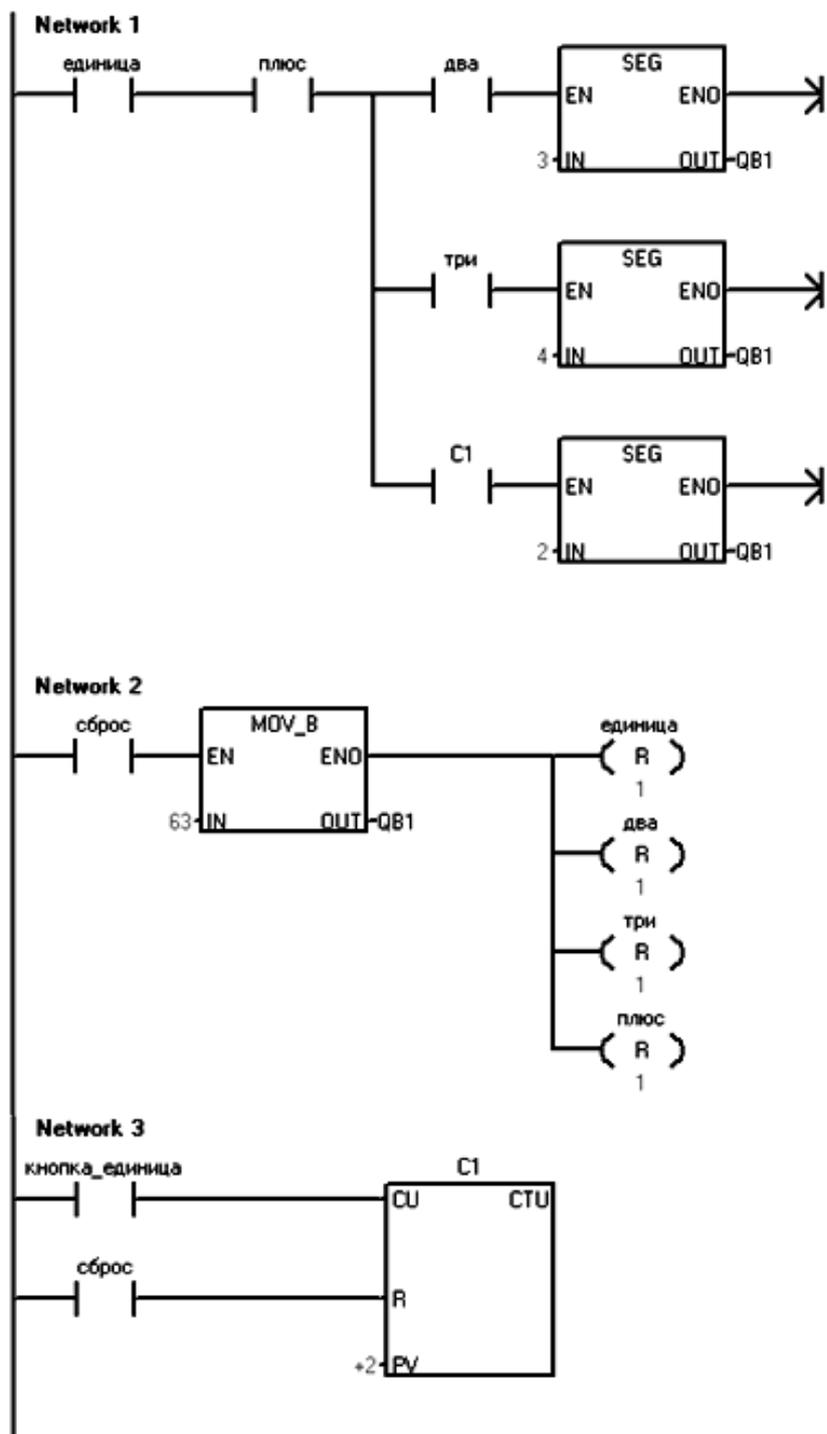


Рис. 3.1. Листинг программы, реализующей операцию сложения

### Контрольные вопросы

1. Какие сегменты памяти необходимо использовать для решения поставленной задачи?
2. Возможно ли решить данную задачу, используя меньшее число сегментов памяти?
3. Какую инструкцию SEG или MOV удобнее использовать для вывода цифр на 7-сегментный индикатор? Почему?
4. Для чего в данной программе необходимо использовать суммирующий счетчик STU?
5. Зачем необходима таблица символов, ведь программа будет работать и без нее?
6. Какой код необходимо задать для инструкции MOV\_B, чтобы отобразить на индикаторе знак минус «-»?

## **ТЕМА № 4**

### **РЕГУЛЯТОР УРОВНЯ РАБОЧЕЙ ЖИДКОСТИ В РЕЗЕРВУАРЕ**

#### **Цель работы**

Получить навыки работы с реальным объектом автоматизации, выполненным на базе лабораторного комплекса «Программируемые логические контроллеры». Ознакомиться с конструкцией, принципом работы и управления регулятора уровня рабочей жидкости в резервуаре. Применить полученные ранее знания, на базе модульного микроконтроллера SIMATIC S7-200 фирмы «Siemens» написать несколько программ управления «регулятором» и проверить его работу.

#### **Задание на работу**

Требуется разработать алгоритмы и рабочие программы для различных режимов работы регулятора уровня рабочей жидкости в резервуаре.

**ВАРИАНТ 1.** Емкость должна наполняться водой после нажатия кнопки «Пуск наполнение» и отключаться по достижении жидкостью верхнего положения. При нажатии кнопки «ПУСК слив» должен включиться сливной насос и отключиться – по достижении жидкостью нижнего уровня. Также необходимо предусмотреть кнопку «СТОП», останавливающую процесс как наполнения, так и слива.

**ВАРИАНТ 2.** Используя переключатель «выбора уровня жидкости», реализовать программу наполнения и слива воды до заданного уровня (0,  $\frac{1}{2}$ , 1). В случае выведения переключателя из работы должна выполняться программа варианта 1. Выполнить индикацию кнопок и текущего уровня жидкости на панели управления.

**ВАРИАНТ 3.** Разработать и проверить программу задержки отключения клапана и насоса на 5 секунд для предотвращения ложных срабатываний датчиков уровня из-за волнения жидкости на поверхности.

#### **Содержание отчёта**

Отчёт по работе должен содержать: цели работы, задания на работу, краткий теоретический обзор, необходимый для выполнения работы, структурную схему регулятора, таблицу входов и выходов, листинг программы с пояснением её работы, выводы по проделанной работе.

## Порядок выполнения работы

1. Ознакомиться со структурной схемой регулятора уровня жидкости. Сопоставить блоки и элементы схемы с установкой, расположенной в аудиториях А-318, А-321.

2. Составить список входов и выходов в соответствии с вариантом программы и описанием алгоритма работы. Рекомендуется использовать обозначения, принятые в съёмной панели управления.

3. Разработать алгоритм работы программ.

4. Разработать рабочую программу «Регулятора» на языке LAD.

5. Определить схему подключений ПЛК к объекту управления.

6. Произвести компиляцию рабочей программы в машинный код.

7. Проверить программу на наличие синтаксических и иных ошибок.

8. При отсутствии ошибок в программе осуществить программирование ПЛК.

9. Показать преподавателю рабочую программу.

10. Открыть задвижку для сливного насоса и кран для заливного клапана. Убедиться в отсутствии течи.

11. Произвести сборку схемы на съёмной панели управления согласно разработанной таблице символов.

12. Подать питание 220 В на щит управления, расположенный в аудитории А-318.

13. Разбиться на группы по 2–4 человека. Одна группа следит за работой регулятора и в случае неисправности отключает его от сети и перекрывает воду. Вторая управляет регулятором со съёмной панели управления. Учащиеся должны проверить правильность работы программы на включение исполнительных органов и их отключение по заданному уровню, а также наличие индикации уровня жидкости в резервуаре.

14. Написать отчёт о проделанной работе.

### Общие сведения о регуляторе уровня жидкости в резервуаре

Универсальный регулятор уровня жидкости представляет собой автоматическую систему регулирования уровня воды в ёмкостях любого объема. Устройство может работать как на заполнение, так и на откачивание воды.

Для реализации такого устройства необходимо, чтобы при достижении заданного уровня срабатывал датчик, вырабатывая электрический сигнал, удобный для дальнейшего преобразования.

Алгоритм функционирования регулятора уровня жидкости в резервуаре должен быть достаточно гибким и содержать следующие задачи автоматизации:

- наполнение или слив до заданного уровня;
- поддержание уровня жидкости;
- индикация уровня жидкости;
- работа в режимах ручного и автоматического управления.

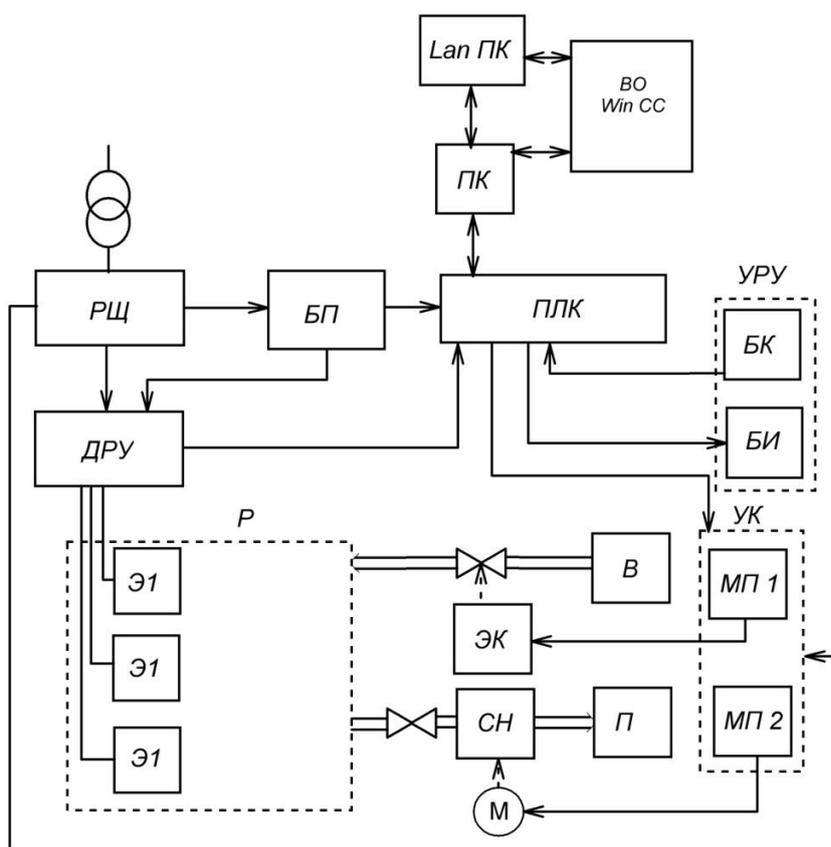


Рис. 4.1. Структурная схема регулятора уровня жидкости

На рис. 4.1 приведена структурная регулятора уровня жидкости. В схеме приняты следующие условные обозначения: БК – блок кнопок; БИ – блок индикации; БП – блок питания; В – вода; ВО Win CC – виртуальный объект; ДР – датчик-реле уровня; М – двигатель; МП – магнитный пускатель; ПЛК – программируемый логический контроллер; СН – сливной насос; П – потребитель; ПК – персональный компьютер; Lan ПК – сетевой компьютер; Р – резервуар; РЩ – распределительный щит; УРУ – устройство ручного управления; УК – устройство коммутации; Э – электрод; ЭК – электромагнитный клапан.

Рассмотрим работу структурной схемы регулятора. Питание из сети поступает на распределительный щит, откуда подаётся на блок питания и устройства коммутации. Блок питания преобразует напряжение питающей

сети в напряжение, пригодное для питания контроллера и датчиков-реле уровня.

Резервуар представляет собой ёмкость с установленными в ней датчиками уровня. В качестве датчиков уровня выступают три электрода, расположенные на разных уровнях. Срабатывание датчиков происходит в момент замыкания электрода на заземлённый корпус резервуара через электропроводную жидкость.

Сигналы с электродов обрабатываются датчиком-реле уровня, который коммутирует и передаёт их на ПЛК. Временная диаграмма работы датчика-реле уровня представлена на рис. 4.2.

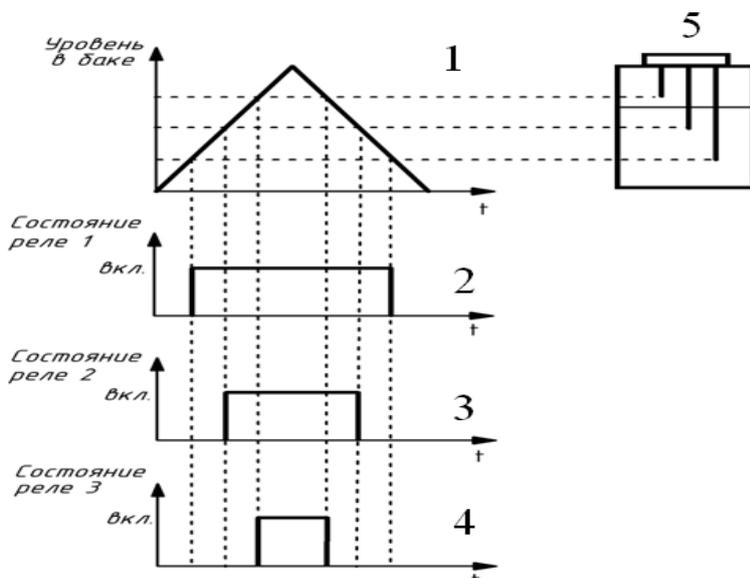


Рис. 4.2. Временная диаграмма работы ДРУ: 1 – уровень воды в резервуаре; 2 – состояние реле 1 при замыкании нижнего электрода; 3 – состояние реле 2 при замыкании среднего электрода; 4 – состояние реле 3 при замыкании верхнего электрода; 5 – расположение электродов в резервуаре

ПЛК представляет собой ядро системы управления. Он взаимодействует с объектом управления через свои входы и выходы: обрабатывает информацию, поступающую от объекта, и выдаёт управляющие сигналы на его исполнительные механизмы. Контроллер управляет технологическим процессом в соответствии с некоторой прикладной программой, разработанной пользователем.

В качестве исполнительных механизмов в этой схеме используются электромагнитный клапан и сливной насос. Включение механизмов возможно через устройства коммутации – два магнитных пускателя, управляемых сигналами с ПЛК.

Для управления технологическим процессом предусмотрены блок кнопок (ПУСК «Насос», ПУСК «Клапан», СТОП), переключатель, задающий

требуемый уровень жидкости (нижний, верхний и средний), а также индикация работы насоса, электромагнитного клапана и уровня жидкости в резервуаре.

На рис. 4.3 представлена модель регулятора уровня жидкости, а также панель управления. Панель управления обозначена на принципиальной схеме как блок УРУ.

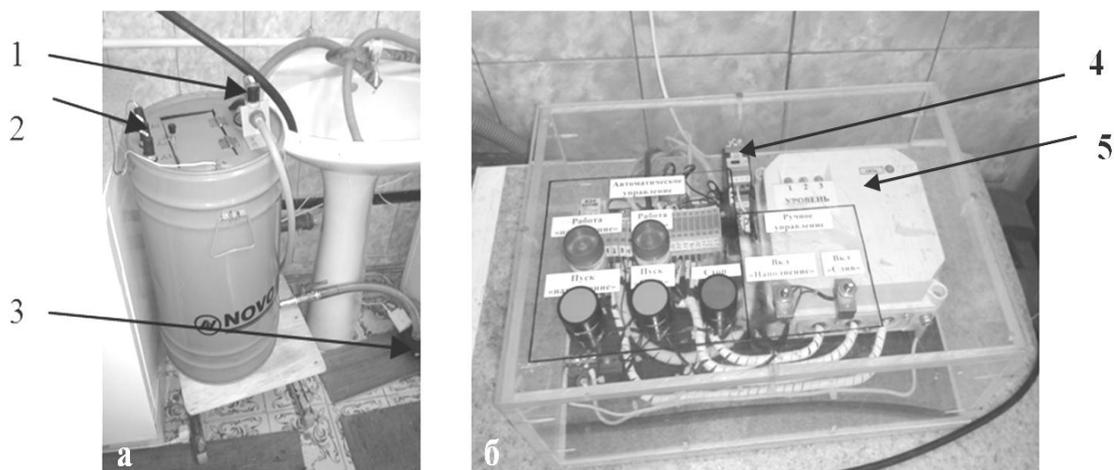


Рис. 4.3: Модель регулятора уровня жидкости:

*а)* резервуар с водой: 1 – электромагнитный клапан; 2 – датчик уровня; 3 – сливной насос;  
*б)* панель управления: 4 – магнитный пускатель; 5 – ДРУ

Объект управления (рис. 4.3, *а*) представляет собой резервуар цилиндрической формы, в корпус которого вмонтированы три электрода (2), электромагнитный клапан (1) и сливное отверстие для подключения насоса (3). Электроды предназначены для контроля уровня жидкости с помощью датчика-реле уровня, расположенного в панели управления.

Панель управления представляет собой ящик из оргстекла (рис. 4.3, *б*), на лицевую поверхность которого выведены органы ручного и автоматического управления, а также световые индикаторы работы двигателя и насоса. Внутри ящика выполнен монтаж схемы, реализующей связь ПЛК с объектом управления.

Стенд с ПЛК и регулятор уровня жидкости расположены в разных учебных аудиториях. Их удалённая связь осуществляется по десятижильному кабелю сечением  $0,25 \text{ мм}^2$ . Объект управления подключается к ПЛК с помощью специально разработанной съёмной лицевой панели (рис. 4.4). На лицевую панель выведены приборные гнёзда с кнопок, сигнальных ламп, датчиков уровня, насоса и электромагнитного клапана. Исполнительные устройства соединяются с контроллером гибкими проводами с помощью гнёзд в съёмной лицевой панели, в зависимости от технического задания.

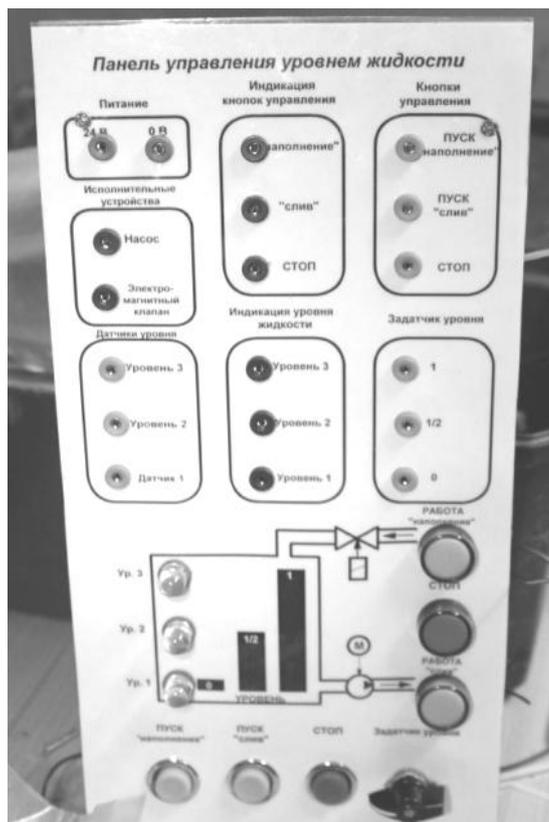


Рис. 4.4. Съёмная лицевая панель управления уровнем жидкости

В качестве исполнительных органов выбраны сливной насос и электромагнитный заливной клапан от стиральной машины «Вирпул». Общая мощность устройств – 70 Вт, напряжение питания – 220 В, частота – 50 Гц. Работа устройств осуществляется через два реле, с катушками на 24 В постоянного тока. Выбор такого реле обусловлен тем, что на дискретных выходах ПЛК SIMATIC S7-200 формируется напряжение 24 В.

Контроллер S7-200 имеет модульную конструкцию и состоит из базового блока CPU 222 (8 вх./6 вых.) и дискретного модуля расширения EM 221 (8 вх./8 вых.). В состав базового блока входят центральный процессор, модуль питания, устройства памяти, дискретные входы/выходы, коммуникационный порт RS-485. Внутренний модуль питания CPU 222 включает два источника постоянного тока: первый работает с напряжением 5 В и обеспечивает питание аппаратной логики, второй рассчитан на напряжение 24 В и служит для питания цепей датчиков. Коммуникационный порт обеспечивает возможность подключения центрального процессора к персональному компьютеру с целью программирования и диагностики, а также создания промышленных сетей.

## Пояснения к выполнению заданий

Для выполнения задания необходимо использовать стенд «Управление уровнем жидкости». На нем есть необходимое количество кнопок и индикаторов.

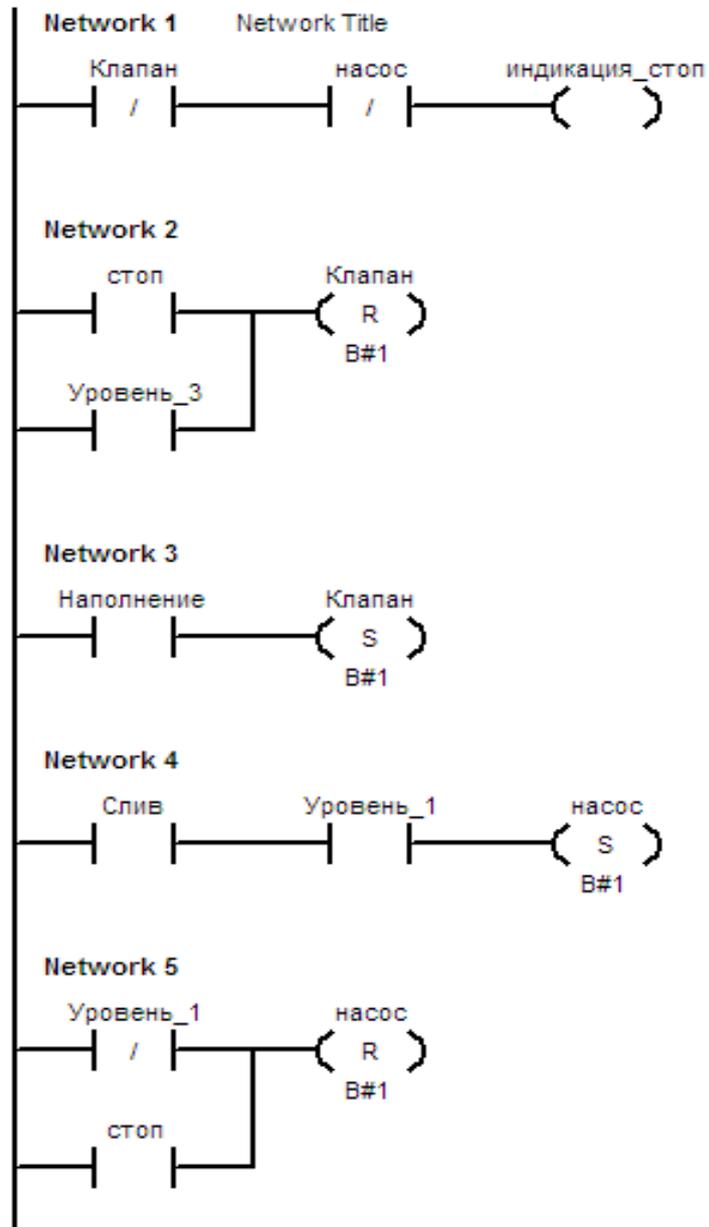
Один из возможных примеров, реализующий автоматическое отключение по достижении жидкостью нижнего или верхнего уровня (вариант 1), представлен в Приложении 1. Программа начинается с вычислительной цепочки индикации лампы «Стоп». Лампа загорается в случае, если не работает ни клапан, ни насос, т.е. реализована логическая функция «И». Цепочки Network 2 и Network 5 реализуют логическую функцию «ИЛИ». Отключение насоса и клапана происходит либо по достижении заданного уровня, либо при нажатии кнопки «Стоп». Цепочки Network 3 и Network 4 обеспечивают включение насоса либо клапана при нажатии соответствующих кнопок. Включить насос можно, если в резервуаре есть жидкость.

Задание варианта 2 решается путем введения в вычислительные цепочки дополнительных нормально открытых контактов с переключателя выбора требуемого уровня. Пример одной из таких цепочек – в Приложении 2. Задержку отключения (вариант 3) можно реализовать следующим образом. Предположим, необходимо слить воду до нижнего уровня. Если замкнут контакт «пуск\_слив» и в резервуаре есть жидкость, то включается «насос». При условии, что уровень жидкости уменьшился до минимального (нормально закрытый контакт «уровень\_1»), в ячейку памяти M0.6 записывается единичное состояние. Контакт M0.6 замыкается и даёт разрешение на отсчёт времени таймеру T105. Таймер замкнёт свой контакт T105 по достижении уставки времени в пять секунд. Значение «1» на контакте T105 даёт разрешение на отключение насоса и сброса бита M0.6, после чего таймер обнулится и примет исходное состояние. То есть, реализуется задержка отключения насоса на пять секунд после достижения нижнего уровня. Это время позволяет дождаться завершения переходного процесса. Отключение клапана по верхнему уровню выполнено аналогичным образом (Приложение 3).

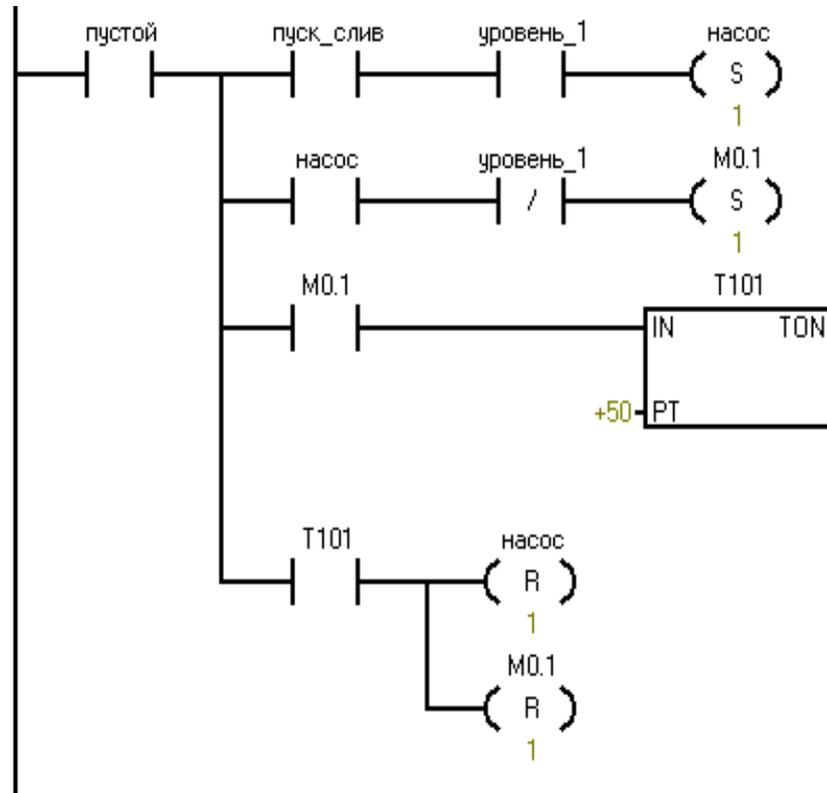
## Контрольные вопросы

1. Для чего предназначен регулятор уровня жидкости?
2. Какие выходные переменные будут присвоены ПЛК? Почему?
3. Как можно организовать индикацию уровня жидкости в резервуаре?
4. Каким образом происходит автоматическое отключение исполнительных органов по достижении критических уровней?

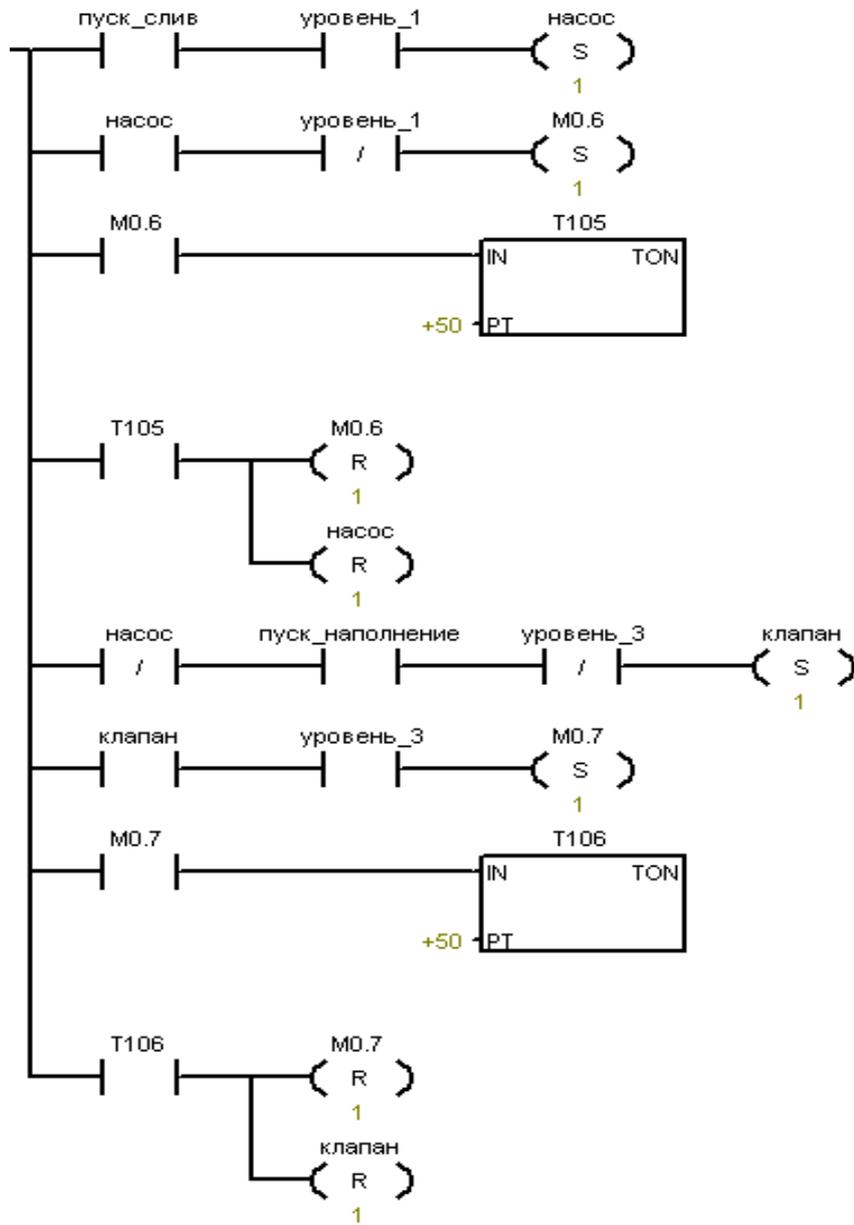
## Пример листинга программы по варианту 1



## Пример листинга программы по варианту 2



## Пример листинга программы по варианту 3



## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Техническое описание стенда «Программируемые логические контроллеры». Часть 1, «Аппаратные и программные средства ПЛК», УИЦ ЗАО Экоинвент.

2. Техническое описание стенда «Программируемые логические контроллеры». Часть 2, «Программирование ПЛК», УИЦ ЗАО Экоинвент.

3. Техническое описание стенда «Программируемые логические контроллеры». Часть 3, «Основы разработки программ для ПЛК», УИЦ ЗАО Экоинвент.

**СОДЕРЖАНИЕ**

Введение .....	3
Тема № 1. Программирование ПЛК SIMATIC S7-200 .....	4
Тема № 2. Реализация 7-сегментной индикации .....	17
Тема № 3. Модель «Калькулятор» .....	23
Тема № 4. Регулятор уровня рабочей жидкости в резервуаре.....	27
Приложение 1 .....	34
Приложение 2 .....	35
Приложение 3 .....	36
Библиографический список .....	37

*Учебное издание*

**МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА  
В ЭЛЕКТРОПРИВОДЕ И МЕХАТРОНИКЕ**

Методические указания к курсовой работе

Составители: **Андреев Николай Кузьмич,  
Малёв Николай Анатольевич  
Козелков Олег Владимирович**

Кафедра Приборостроение и мехатроника КГЭУ

Редактор редакционно-издательского отдела *К.В. Аршинова*  
Компьютерная верстка *Т.И. Лунченкова*

Подписано в печать 31.05.20.

Формат 60×84/16. Бумага «Business». Гарнитура «Times». Вид печати РОМ.  
Усл. печ. л. 2,32. Уч.-изд. л. 2,58. Тираж экз. Заказ 4627

Редакционно-издательский отдел КГЭУ, 420066, Казань, Красносельская,  
51

