

**Государственное бюджетное образовательное учреждение
высшего образования Московской области
«Университет «Дубна»
Филиал «Протвино»
Кафедра «Автоматизация технологических процессов и
производств»**

В.А. Коковин

**ЛАБОРАТОРНЫЕ РАБОТЫ ПО ДИСЦИПЛИНЕ
«АВТОМАТИЗАЦИЯ ТЕХНОЛОГИЧЕСКИХ
ПРОЦЕССОВ И ПРОИЗВОДСТВ»**

ЭЛЕКТРОННОЕ МЕТОДИЧЕСКОЕ ПОСОБИЕ

**Рекомендовано
кафедрой автоматизации технологических процессов и
производств
филиала «Протвино» государственного университета «Дубна»
в качестве методического пособия для студентов,
обучающихся по направлению
«Автоматизация технологических процессов и производств»**

**Протвино
2017**

Рецензент:

кандидат технических наук, доцент, доцент кафедры
«Транспортные средства и бортовые информационно-управляющие
системы»

ФГБОУ ВО «Московский технологический университет»

В.В. Плешаков

Коковин, В.А

К59 Лабораторные работы по дисциплине «Автоматизация технологических процессов и производств»: электронное методическое пособие / В.А. Коковин. — Протвино, 2017. — 67 с.

Методическое пособие содержит описание четырех лабораторных работ по дисциплине «Автоматизация технологических процессов и производств» и предназначено для изучения принципов построения систем автоматизации, разработки программ на языке стандарта МЭК 61131-3, с использованием программируемых логических контроллеров и персональных компьютеров.

Методическое пособие предназначено для студентов, обучающихся по направлению «Автоматизация технологических процессов и производств».

ББК 34.5я 73

© Государственное бюджетное образовательное учреждение
высшего образования Московской области
«Университет «Дубна», филиал «Протвино», 2017
© Коковин В.А., 2017

Введение

Автоматизация технологических процессов и производств повышает эффективность машиностроительного производства, во многом определяя конкурентоспособность изделий на современном рынке. При автоматизации производства следует исходить не только из возможностей существующей технологии, но и из возможностей применения новых высокоэффективных технологических процессов, в основе которых лежат последние достижения современной науки и техники [1].

Современное автоматизированное машиностроительное производство базируется на гибких производственных модулях (ГПМ), объединенных в гибкую производственную систему (ГПС) с транспортно-накопительной подсистемой и автоматическими складами. В состав ГПМ, как правило, входят ЧПУ и системы манипулирования. Для управления электроавтоматикой станка используются программируемые логические контроллеры. Логические уравнения в них можно описать как с помощью принципиальной схемы, так и соответствующих команд. В современных системах ЧПУ (числовое программное управление) для управления электроавтоматикой станка применяются языки высокого уровня, потому что сложные функции, например, обработку аналоговых сигналов или специальных алгоритмов регулирования, нельзя удовлетворительно описать с помощью команд контроллера. Поддержка программирования контроллера на языке высокого уровня становится обязательным условием применения его в системах ЧПУ.

Развитие промышленного оборудования, в том числе и металлорежущих станков, идет по пути повышения уровня автоматизации выполняемых операций, что предъявляет особые требования к электроавтоматике. Одновременно с увеличением объема решаемых задач при проектировании возникают проблемы, связанные с уменьшением размеров станций электрооборудования, простотой его наладки, диагностикой неисправностей и их оперативным устранением, повышением надежности работы. Эти проблемы привели к решению задачи синтеза функций электроавтоматики на новом уровне — с использованием программных методов решения.

На мировом рынке автоматизированных систем металлообработки ведущую роль занимает немецкая фирма *SIEMENS*. Эта фирма выпускает большой набор систем ЧПУ и ПЛК (программируемых логических контроллеров) различного уровня сложности,

широко представленный на российском рынке. В настоящее время фирма *SIEMENS* является законодателем в разработке, создании и применении ПЛК и выпускает четыре семейства ПЛК — S7-200, 300, 400 и 1200.

Исходя из этого, в основе учебных стендов для выполнения лабораторных работ по дисциплине «Автоматизация технологических процессов и производств» применяются ПЛК фирмы *SIEMENS*.

Цель данного учебно-методического пособия — помочь студентам, в рамках выполнения лабораторных работ по дисциплине «Автоматизация технологических процессов и производств», изучить особенности использования ПЛК в современных производствах и разработать программы на языках промышленной автоматизации стандарта МЭК 61131-3.

1 Принципы построения ПЛК

Типичный ПЛК представляет собой блок, имеющий определенный набор выходов и входов, для подключения датчиков и исполнительных механизмов [2]. Логика управления описывается программно на основе микрокомпьютерного ядра. Одинаковые ПЛК могут реализовывать совершенно разные функции, в отличие от контроллеров, выполненных на основе микросхем с жесткой логикой. При этом изменение функций не требует переделок аппаратной части. Связь с объектами управления (для воздействия на объект через выходы и для получения информации через входы) осуществляется через аппаратно реализованные входы и выходы ПЛК. Как правило, входы и выходы унифицированы и отвечают определенному промышленному стандарту. На рис. 1 представлен принцип работы ПЛК.

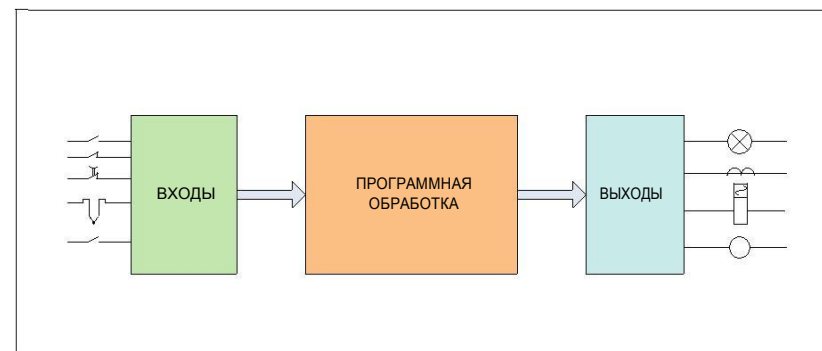


Рис. 1. Принцип работы ПЛК

Реализация заданного алгоритма с помощью ПЛК выполняется циклически и заключается в периодическом считывании состояния входов, программной обработке логических уравнений и формировании через выходы последовательности воздействий на объект управления. Современные ПЛК имеют значение времени рабочего цикла в пределах единиц миллисекунд. Поскольку время реакции большинства исполнительных устройств значительно выше времени рабочего цикла, ограничений на использование ПЛК нет. В некоторых случаях необходимо зафиксировать на входе ПЛК импульс малой длительности (гораздо меньше времени

цикла). В этом случае используют специальную реализацию входов с элементами памяти (триггер).

Пользовательская программа, загружаемая в ПЛК, работает с мгновенными значениями входов, при этом предполагается, что значения входов во время рабочего цикла не изменяются. Это фундаментальный принцип построения ПЛК сканирующего типа [2]. Время цикла может меняться от объема выполняемого программного кода. Некоторые задачи автоматизации, например связанные с автоматическим регулированием, требуют детерминированности рабочего цикла. В этом случае предусмотрен контроль времени цикла и если отдельные ветви кода управляющей программы выполняются слишком быстро, в рабочий цикл добавляются искусственные задержки [2].

2 Системы автоматизации *SIMATIC*

SIMATIC (фирма *SIEMENS*) является базовой системой автоматизации всех отраслей промышленного производства, объединяющей в своем составе широкую гамму стандартных программных и аппаратных компонентов, поддерживающих множество стандартных открытых интерфейсов для выполнения специализированных расширений [3].

Семейство *SIMATIC* включает в свой состав:

- Программируемую аппаратуру управления.
- Периферийные устройства систем распределенного ввода-вывода.
- Программаторы и промышленные компьютеры.
- Промышленное программное обеспечение *SIMATIC*.
- Микро комплекты автоматизации.
- Модульные системы автоматизации *CBA (Component based Automation)*.
- Аппаратуру и программное обеспечение систем технического зрения *Machine Vision*.

Рассмотрим более подробно особенности реализации ПЛК в различных семействах *SIMATIC*, основные характеристики и уровень решаемых задач с помощью этих ПЛК.

2.1 Логические модули *LOGO!*

Логические модули *LOGO!* из-за своих характеристик занимают особое место в линейке контроллеров *SIMATIC*. На рис. 2.1 изображен внешний вид семейства логических модулей *LOGO!*. Они обладают уникальными техническими и потребительскими

свойствами, и представляют собой компактные, экономичные и универсальные решения для построения простых устройств автоматического управления. Логические модули *LOGO!* это:

- Простота обслуживания, удобное и простое программирование
- "Все в одном": интегрированный дисплей и клавиатура, программируемая логика, библиотеки встроенных функций, входы и выходы.
- 34 встроенные функции, до 130 функций на программу.
- Программирование со встроенной клавиатуры без использования программатора и специального программного обеспечения.

Применение логических модулей *LOGO!* позволяет получать компактные, удобные и экономичные решения для построения простейших устройств автоматического управления.

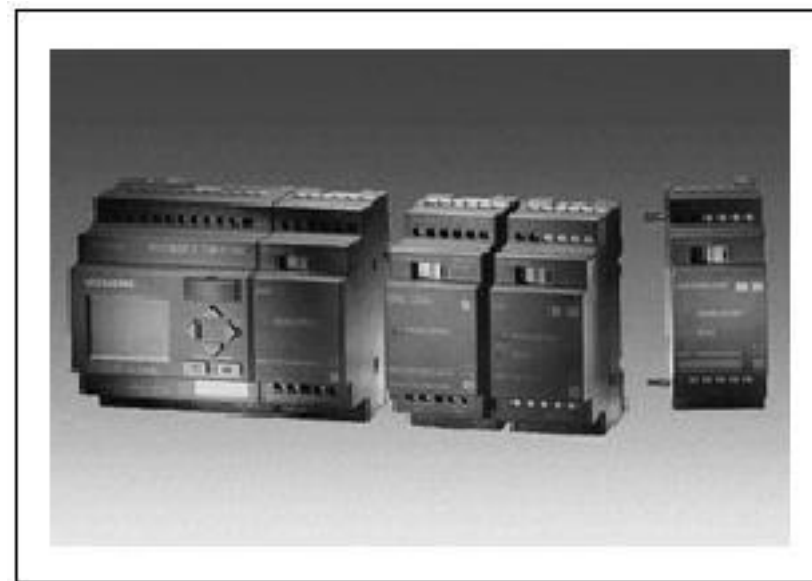


Рис. 2.1. Логические модули *LOGO!*

Программируемость и универсальность модулей позволяет использовать их:

- Для управления освещением, прерывателями, тентами, дверями и воротами, системами контроля доступа, защитными и технологическими барьерами, вентиляторами и т. д.
- В шкафах управления и силовых шкафах.

- В системах управления насосами, небольшими прессами, компрессорами и т. д.
- В системах управления поливом в оранжереях.
- В судовых системах.
- В системах предварительной обработки сигналов и других системах.

Для максимальной адаптации к требованиям решаемой задачи в моделях *LOGO!* предусмотрена возможность подключения дополнительных модулей ввода-вывода и коммуникационных модулей.

2.2 ПЛК семейства *SIMATIC S7-200*

Программируемые логические контроллеры *SIMATIC S7-200* предназначены для построения относительно простых и недорогих систем автоматического управления и могут использоваться для замены существующих релейно-контактных схем. Контроллеры поддерживают мощную систему команд и способны выполнять логические операции, математические операции с фиксированной и плавающей точкой, поддерживать алгоритмы ПИД регулирования и позиционирования и т. д. Для организации обмена данными контроллеры позволяют использовать:

- *PPI (Point to Point Interface)*,
 - *MPI (Multi Point Interface)*,
 - *AS-Interface*,
 - *PROFIBUS*,
 - *Industrial Ethernet*,
 - *Internet*.
 - Широкий спектр модулей ввода-вывода дискретных и аналоговых сигналов.
 - 4 коммуникационных модуля, обеспечивающие возможность подключения к *AS-Interface*, сети *PROFIBUS DP* (только ведомое устройство) и *Industrial Ethernet*.
 - Модуль модема EM 241.
 - Модуль позиционирования EM 253.
- На рис. 2.2 представлено семейство ПЛК *S7-200*.

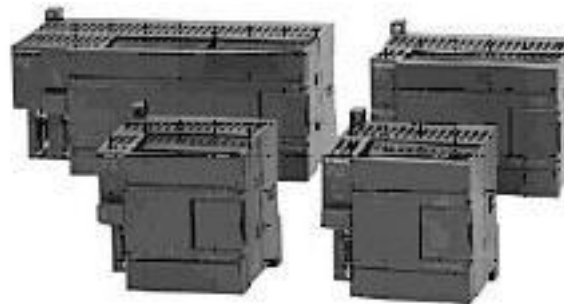


Рис. 2.2. ПЛК семейства *S7-200*

В основе семейства *S7-200* лежит функциональная модульность, которая позволяет разрабатывать приложения в зависимости от конкретной задачи. Ведущее место в наборе модулей занимает центральный процессорный модуль (*CPU*).

2.3 ПЛК семейства *SIMATIC S7-300*

Модульные программируемые логические контроллеры *SIMATIC S7-300* предназначены для решения задач автоматического управления низкой и средней степени сложности (рис. 2.3). ПЛК этого семейства наиболее массово используются в промышленной автоматизации. Они характеризуются:

- Наличием центральных процессоров, оснащенных набором встроенных входов и выходов и поддерживающих целый ряд технологических функций на уровне своей операционной системы.
- Наличием центральных процессоров для построения систем противоаварийной защиты и
- автоматики безопасности с поддержкой соответствующих функций на уровне операционной системы.

ПЛК *S7-300* находят применение для автоматизации машин специального назначения, текстильных и упаковочных машин, машиностроительного оборудования, оборудования для производства технических средств управления и электротехнического оборудования, в системах автоматизации судовых установок и систем водоснабжения и т. д.

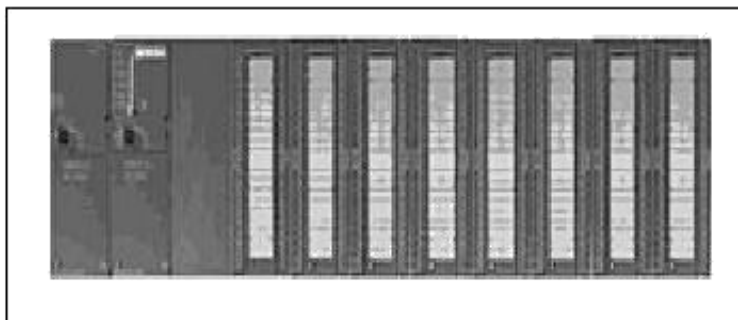


Рис. 2.3. ПЛК семейства S7-300

2.4 ПЛК семейства *SIMATIC S7-400*

Программируемые логические контроллеры *SIMATIC S7-400* предназначены для решения задач автоматического управления средней и высокой степени сложности (рис. 2.4).

- Высокая вычислительная мощность, комплексный набор ко-манд, наличие *MPI* интерфейса, способность работать в локаль-ных и глобальных вычислительных сетях делают контроллер ис-ключительно мощным.

- Поддержка мультипроцессорных конфигураций.

- Набор встроенных функций, всеобъемлющая диагностика, парольная защита, удобная система подключения внешних цепей, отсутствие ограничений на порядок размещения модулей позволяют создавать многообразные конфигурации систем управления

- Широкий спектр центральных процессоров, сигнальных, функциональных, коммуникационных и интерфейсных модулей позволяют в максимальной степени адаптировать контроллер к выполнению поставленных задач. Поддержка функций “горячей” замены модулей.

- Возможность подключения к одному базовому блоку кон-троллера до 21 стойки расширения.

Модификации:

- Стандартное исполнение *SIMATIC S7-400*.

- Резервированная система автоматизации *SIMATIC S7-400H*.

- Программируемые контроллеры для построения систем про-тивоаварийной защиты и автоматики безопасности *SIMATIC S7-400F/FH*.

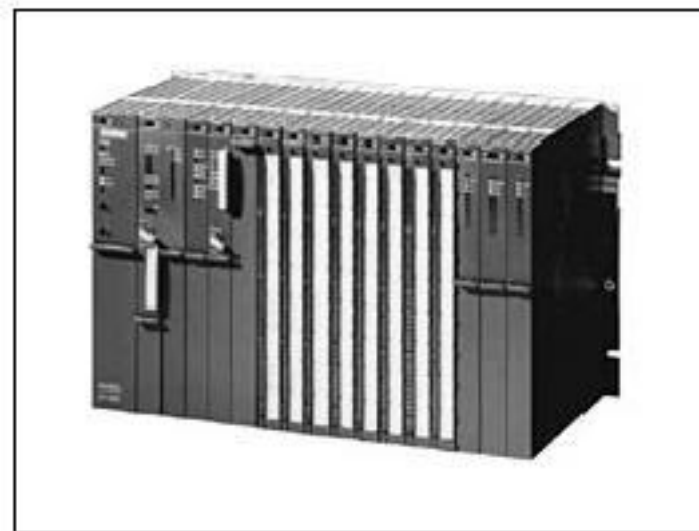


Рис. 2.4. ПЛК семейства S7-400

2.5 ПЛК семейства *SIMATIC S7-1200*

Программируемые контроллеры *SIMATIC S7-1200* имеют мо-дульную конструкцию и универсальное назначение (рис. 2.5). Они способны работать в реальном масштабе времени, могут исполь-зоваться для построения относительно простых узлов локаль-ной автоматики или узлов комплексных систем автоматического управления, поддерживающих интенсивный коммуникационный обмен данными через сети *Industrial Ethernet/ PROFINET*, а также *PtP (Point-to-Point)* соединения.

Контроллеры *S7-1200* имеют компактные пластиковые корпу-са со степенью защиты *IP20*, могут монтироваться на стандарт-ную 35 мм профильную шину *DIN* или на плоскую поверхность и сохраняют работоспособность в диапазоне температур от 0 до +50 °С. Они способны обслуживать от 10 до 284 дискретных и от 2 до 51 аналогового канала ввода-вывода.

В составе программируемого контроллера *S7-1200* находят применение модули центральных процессоров (*CPU*); коммуни-кационные модули (*CM*); модули (*SM*) и платы (*SB*) ввода-вывода дискретных и аналоговых сигналов; 4-канальный коммутатор *Industrial Ethernet (CSM 1277)* и модуль блока питания (*PM 1207*).

Для конфигурирования и программирования новой серии кон-троллеров используется программное обеспечение *Simatic Step 7*

Basic, которое позволяет работать одновременно и с контроллерами *Simatic S7-1200*, и с панелями серии *Basic Panels*. Пакет *Simatic Step 7 Basic v10.5* со встроенным ПО *WinCC Basic* предоставляет пользователю множество новых, интуитивно понятных возможностей по совместной разработке проектов для контроллера и панелей. Это значительно повышает удобство и эффективность работы.

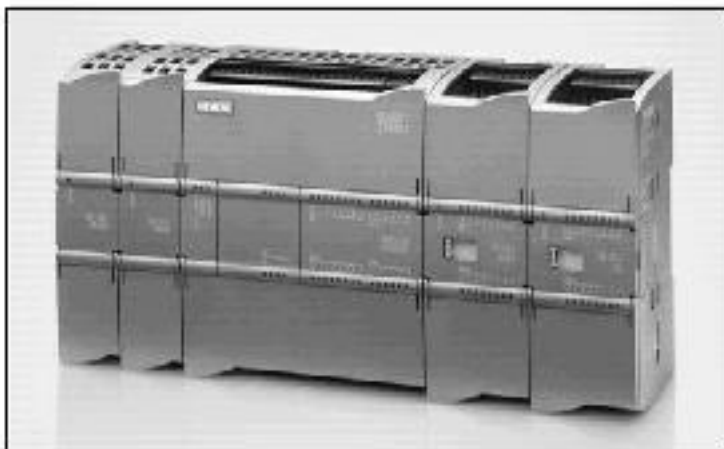


Рис. 2.5. ПЛИК семейства S7-1200

Область применения:

- управление производственным оборудованием в различных областях промышленности;
- обмен данными между удалёнными объектами;
- задачи управления перемещением и позиционирование приводов;
- управление инженерными системами зданий и сооружений;
- задачи регулирования на основе *PID*-алгоритмов. Центральные процессоры

В серии S7-1200 используется 3 модели центральных процессоров, отличающихся производительностью, объемами встроенной памяти, количеством и видом встроенных входов и выходов и другими показателями. Каждая модель имеет три модификации:

- С напряжением питания =24 В, дискретными входами = 24 В и дискретными выходами =24 В/0.5А на основе транзисторных ключей.

- С напряжением питания =24 В, дискретными входами = 24 В и дискретными выходами с замыкающими контактами реле и нагрузочной способностью до 2 А на контакт.

- С напряжением питания ~115/230 В, дискретными входами = 24 В и дискретными выходами с замыкающими контактами реле и нагрузочной способностью до 2 А на контакт.

Каждый центральный процессор S7-1200 оснащен встроенным интерфейсом *Ethernet*, имеет возможность подключения до трех коммуникационных модулей и установку одной платы ввода-вывода. Дополнительно к CPU 1212C может подключаться до 2, к CPU 1214C — до 8 сигнальных модулей (SM).

Коммуникации

Основным коммуникационным портом для контроллеров S7-1200, является встроенный *Ethernet*-интерфейс. С его помощью производится программирование, диагностика, обмен данными с другими системами автоматизации или устройствами человеко-машинного интерфейса. Связь с устройствами, имеющими другие интерфейсы, возможна с помощью подключаемых дополнительно коммуникационных модулей, имеющих интерфейсы RS 232 или RS 485, что позволяет устанавливать *PtP*-соединения с использованием различных протоколов обмена данными.

Программное обеспечение *Simatic Step 7 Basic* позволяет осуществить комплексный подход к проектированию систем автоматизации на основе контроллеров S7-1200 (рис. 2.6 — окно пакета *Simatic Step 7 Basic*).



Рис. 2.6. Программное обеспечение для ПЛИК семейства S7-1200

Программный пакет объединяет в себе средства создания кон-фигурации оборудования, программирования, отладки програм-мы и *on-line* диагностики работы. *Simatic WinCC Basic*, является частью *Simatic Step 7 Basic* и позволяет создавать проекты для панелей оператора серии *Basic Panels*. Подобная интеграция и ис-пользование единого программного обеспечения даёт несомнен-ное преимущество при проектировании и дальнейшем обслужи-вании системы.

3 Обзор семейства ПЛК S7-200 фирмы SIEMENS

В основе семейства S7-200 лежит функциональная модуль-ность, которая позволяет разрабатывать приложения в зависимо-сти от конкретной задачи. Ведущее место в наборе модулей зани-мает центральный процессорный модуль (CPU). CPU S7-200 со-стоит из микропроцессора, встроенного или внешнего источника питания, входных и выходных цепей, находящихся в компактном корпусе (рис. 3.1). После загрузки программы S7-200 содержит логику, необходимую для контроля и управления входными и вы-ходными устройствами в разрабатываемом приложении.

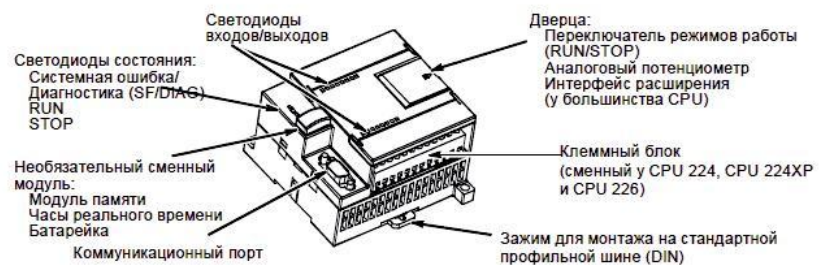


Рис. 3.1. Структура и состав ПЛК семейства S7-200

Фирма *Siemens* предлагает различные модели CPU S7-200 с разнообразными характеристиками производительности и функ-циями, чтобы обеспечить создание эффективных решений для са-мых разнообразных приложений. В табл. 3.1 дается краткое срав-нение функций различных CPU

Таблица 3.1

Сравнение характеристик модулей CPU S7-200

Параметры	CPU 221	CPU 222	CPU 224	CPU 224XP	CPU 226
Физические размеры	90 x 80 x 62	90 x 80 x 62	120.5 x 80 x 62	140 x 80 x 62	190 x 80 x 62
Программная память:	4096 байт	4096 байт	8192 байт	12288 байт	16384 айт
Память данных	2048 байт	2048 байт	8192 байт	10240байт	10240 байт
Арифметика с плавающей точкой	Да	Да	Да	Да	Да
Цифровые встроенные входы/ выходы	6 вх./ 4вых.	8 вх./ 6вых	14 вх./ 10вых	14 вх./ 10вых	24 вх./ 16вых
Аналоговые встроенные входы/ выходы	-	-	-	2 вх. / 1 вых.	
Модули расширения	0 моду-лей	2 модуля	7 модулей	7 модулей	7 моду-лей
Коммуникацион-ные порты	1 RS-485	1 RS-485	1 RS-485	2 RS-485	2 RS-485

В таблице 3.2 представлены области памяти и функции CPU S7-200.

Таблица 3.2

Области памяти и функции CPU S7-200

Параметры	CPU 221	CPU 222	CPU 224	CPU 224XP	CPU 226
Размер программы пользователя	4096 байт	4096 байт	8192 байт	12288 байт	16384 байт
Размер данных пользователя	2048 байт	2048 байт	8192 байт	10240 байт	10240 байт
Регистр входов образа процессов	I0.0-I15.7	I0.0-I15.7	I0.0-I15.7	I0.0-I15.7	I0.0-I15.7
Регистр выходов образа процессов	Q0.0-Q15.7	Q0.0-Q15.7	Q0.0-Q15.7	Q0.0-Q15.7	Q0.0-Q15.7

Продолжение таблицы 3.2

Параметры	CPU 221	CPU 222	CPU 224	CPU 224XP	CPU 226
Память переменных (V)	VB0-VB2047	VB0-VB2047	VB0-VB8191	VB0-VB10239	VB0-VB10239
Локальная память (L)	LB0-LB63	LB0-LB63	LB0-LB63	LB0-LB63	LB0-LB63
Битовая память (M)	M0.0-M31.7	M0.0-M31.7	M0.0-M31.7	M0.0-M31.7	M0.0-M31.7
Специальная память (SM) только чтение	SM0.0-SM179.7 SM0.0-SM29.7	SM0.0-SM299.7 SM0.0-SM29.7	SM0.0-SM549.7 SM0.0-SM29.7	SM0.0-SM549.7 SM0.0-SM29.7	SM0.0-SM549.7 SM0.0-SM29.7
Таймеры	256 (T0-T256)	256 (T0-T256)	256 (T0-T256)	256 (T0-T256)	256 (T0-T256)
Задержка включения с запоминанием	T0, T64, T1-T4, T65-T68	T0, T64, T1-T4, T65-T68	T0, T64, T1-T4, T65-T68	T0, T64, T1-T4, T65-T68	T0, T64, T1-T4, T65-T68
1 мс	T5-T31, T69-T95	T5-T31, T69-T95	T5-T31, T69-T95	T5-T31, T69-T95	T5-T31, T69-T95
10 мс					
100 мс	T32, T96	T32, T96	T32, T96	T32, T96	T32, T96
Задержка вкл./выкл.	T33-T36, T97-T100	T33-T36, T97-T100	T33-T36, T97-T100	T33-T36, T97-T100	T33-T36, T97-T100
1 мс	T37-T63, T101-T255	T37-T63, T101-T255	T37-T63, T101-T255	T37-T63, T101-T255	T37-T63, T101-T255
10 мс					
100 мс					
Счетчики	C0-C255	C0-C255	C0-C255	C0-C255	C0-C255
Скоростные счетчики	HC0-HC5	HC0-HC5	HC0-HC5	HC0-HC5	HC0-HC5
Аккумуляторные регистры	AC0-AC3	AC0-AC3	AC0-AC3	AC0-AC3	AC0-AC3
Вызов/Подпрограмма	0-63	0-63	0-63	0-63	0-63
Программы обработки прерываний	0-127	0-127	0-127	0-127	0-127
PID-регуляторы	0-7	0-7	0-7	0-7	0-7

4 Программное обеспечение STEP-7 Micro/Win 4.0

Пакет STEP 7 Micro/WIN 4.0 предназначен для программирования контроллеров семейства SIMATIC S7-200. Подключение центрального процессора S7-200 к программатору или компьютеру производится через PC/PPI кабель или через MPI кабель и ком-

муникационный процессор CP 5511/ CP 5611, установленный в программатор/компьютер. Допускается дистанционное программирование контроллеров S7-200 через сеть Industrial Ethernet. В последнем случае компьютер/программатор должен быть оснащен сетевой Ethernet-картой, например, коммуникационным процессором CP 1512/CP 1612.

- Пакет программирования контроллеров SIMATIC S7-200, работающий под управлением операционных систем Windows 95/98/ NT/ME/2000/XP.

- Для решения как простых, так и достаточно сложных задач автоматического управления.

- Большой объем встроенных функций, облегчающих процесс разработки и отладки программы.

- Конфигурирование и настройка параметров устройств человеко-машинного интерфейса, предназначенных для работы с SIMATIC S7-200.

- Расширение функциональных возможностей за счет подключения дополнительных библиотек Instruction Library и пакетов TP-Designer и SIMATIC MicroComputing.

Пакет STEP 7 Micro/WIN 4.0 характеризуется следующими показателями:

1. Простая структура программы, состоящая из одного организационного блока, вызываемых подпрограмм и подпрограмм обслуживания прерываний. Дополнительно может создаваться блок данных.

2. Программирование на языках STL, LAD и FBD.

3. Работа в LAD и FBD с использованием символики IEC 61131-1 или S7-200.

4. Возможность использования символьной адресации. Распределение ресурсов и определение соответствия символьных имен физическим адресам выполняется в таблице символов. Вывод программы на печать с символьной или абсолютной адресацией.

5. Простая для изучения и интуитивно понятная система команд. Множество модификаций различных команд (например, команды пересылки данных MOVE).

6. Набор функций настройки центрального процессора для определения аппаратной конфигурации, времени фильтрации сигналов, размеров области памяти, значений сигналов, выводимых на выходы в случае перехода центрального процессора в режим "STOP", установка часов реального времени и паролей различных уровней, а также сетевых адресов.

7. Набор мастеров для выполнения операций конфигурирования: текстового дисплея *TD 200*, ПИД регуляторов, коммуникационных связей между центральными процессорами, скоростных счетчиков, модема *EM 241*, коммуникационного процессора *CP 243-1 (IT)*, модуля позиционирования *EM 253*.

8. Интерактивная контекстно-зависимая система помощи.

9. Интерактивная работа с центральным процессором: перевод центрального процессора в режим *RUN* или *STOP*, загрузки программы из памяти центрального процессора в программатор/компьютер, загрузки программы из программатора/компьютера в память центрального процессора, сравнение текущей версии программы *STEP 7 Micro/WIN* с программой, загруженной в память центрального процессора.

10. Сохранение всех данных проекта в файле.

Более подробно о структуре пакета и интерфейсе можно посмотреть в разделе **Help STEP 7 Micro/WIN**. Кроме того в разделе 7 настоящего пособия подробно рассмотрен пример по созданию программы, компиляции, сохранению и загрузки в контроллер.

5 Основы ПЛК семейства S7-200

5.1 Выполнение логики управления с помощью ПЛК S7-200

Основной принцип действия ПЛК S7-200 достаточно прост:

- ПЛК считывает состояние входов.
- Программа, хранящаяся в ПЛК, использует эти входы для анализа логики управления. Во время обработки программы ПЛК S7-200 обновляет данные.

- S7-200 записывает данные на выходы.

S7-200 выполняет последовательность задач неоднократно. Эта регулярная обработка задач называется циклом. Как показано на рис. 5.1, ПЛК выполняет в цикле большинство или все из следующих задач:

- Чтение входов: ПЛК копирует состояние физических входов в регистр входов образа процесса.

- Выполнение логики управления в программе:

ПЛК выполняет команды программы и сохраняет значения в различных областях памяти.

- Обработка запросов на обмен данными: ПЛК выполняет все задачи, необходимые для обмена данными.

- Самодиагностика *CPU*: ПЛК проверяет, чтобы встроенное программное обеспечение, программная память и все модули расширения работали надлежащим образом.

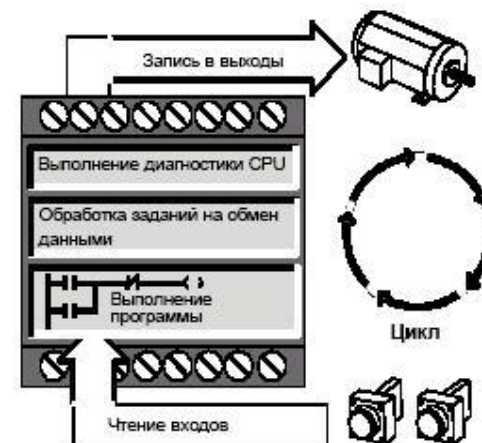


Рис. 5.1. Цикл S7-200

- **Запись в выходы:** Значения, хранящиеся в регистре выходов образа процесса, записываются в физические выходы.

Выполнение цикла зависит от того, находится ли ПЛК в состоянии *STOP* или в состоянии *RUN*. В состоянии *RUN* программа выполняется; в состоянии *STOP* программа не выполняется.

Чтение входов

Цифровые входы: В начале цикла текущие значения цифровых входов считываются, а затем записываются в регистр входов образа процесса.

Аналоговые входы: ПЛК S7-200 не обновляет аналоговые входы модулей расширения автоматически как часть цикла, если не активизирована фильтрация аналоговых входов. Аналоговый фильтр обеспечивает стабильность сигналов. Структура ПЛК позволяет активизировать аналоговый фильтр для каждого входа. Если фильтр для аналогового входа активизирован, то ПЛК обновляет этот аналоговый вход один раз за цикл, выполняет функцию фильтрации и сохраняет отфильтрованное значение внутри. Это отфильтрованное значение затем предоставляется в распоряжение всякий раз, когда программа пользователя обращается к этому аналоговому входу.

Исполнение программы

На этом участке цикла ПЛК S7-200 обрабатывает программу с первой команды до последней. Пользователь может непосредственно управлять входами и выходами и получать, таким обра-

зом, доступ к ним во время исполнения основной программы или программы обработки прерываний. Если в программе используются прерывания, то программы обработки прерываний, которые ставятся в соответствие прерывающим событиям, хранятся как часть основной программы. Однако программы обработки прерываний исполняются не как составная часть нормального цикла, а только тогда, когда происходит прерывающее событие (оно возможно в любом месте цикла).

Обработка запросов на обмен данными
На участке цикла, выделенном для обработки коммуникаций, ПЛК обрабатывает все сообщения, полученные из коммуникационного порта или от интеллектуальных модулей ввода/вывода.

Самодиагностика CPU
На этом участке цикла ПЛК S7-200 проверяет надлежащую работу CPU, области памяти и состояние модулей расширения.

Запись в цифровые выходы
В конце каждого цикла ПЛК записывает значения, хранящиеся в регистре выходов образа процесса, в цифровые выходы. (Аналоговые выходы обновляются немедленно, независимо от цикла.)

5.2 Доступ к данным

ПЛК S7-200 хранит информацию в различных местах памяти, которые имеют однозначные адреса. Вы можете явно указать адрес в памяти, к которому вы хотите обратиться. Благодаря этому ваша программа имеет прямой доступ к информации. Табл. 5.1 показывает диапазон целых значений, которые могут быть представлены с помощью данных различной длины.

Таблица 5.1

Десятичные и шестнадцатеричные диапазоны для данных различной длины

Представление	Байт (B)	Слово (W)	Двойное слово (D)
Целое без знака	от 0 до 255 от 0 до FF	от 0 до 65 535 от 0 до FFFF	от 0 до 4 294 967 295 от 0 до FFFF FFFF
Целое со знаком	от -128 до +127 от 80 до 7F	от -32 768 до +32 767 от 8000 до 7FFF	от -2 147 483 648 до +2 147 483 647 от 8000 0000 до 7FFF FFFF
Вещественное IEEE 32-битовое с плавающей точкой	Неприменимо	Неприменимо	от +1.175495E-38 до +3.402823E+38 (положительное) от -1.175495E-38 до -3.402823E+38 (отрицательное)

Для обращения к биту в некоторой области памяти вы необходимо указать адрес бита. Этот адрес состоит из идентификатора

области памяти, адреса байта и номера бита. На рис. 5.2 показан пример обращения к биту (адресация в формате «байт.бит»). В этом примере за областью памяти и адресом байта ($I = input$ [вход], 3 = байт 3) следует точка («.»), чтобы отделить адрес бита (бит 4).

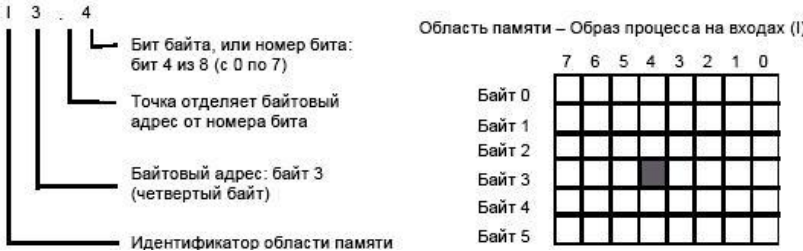


Рис. 5.2. Адресация байт. бит

Применяя формат байт, бит, можно обратиться к данным в большинстве областей памяти (V, I, Q, M, S, L и SM) как к байтам, словам или двойным словам. Если необходимо обратиться к байту, слову или двойному слову данных в памяти, то нужно указать эти адреса подобно адресу бита. Таким образом указывается идентификатор области, обозначение длины данных и начальный адрес байта, слова или двойного слова, как показано на рис. 5.3.

К данным в других областях памяти (напр., T, C, HC и аккумуляторы) необходимо обращаться, указывая в качестве адреса идентификатор области и номер элемента.

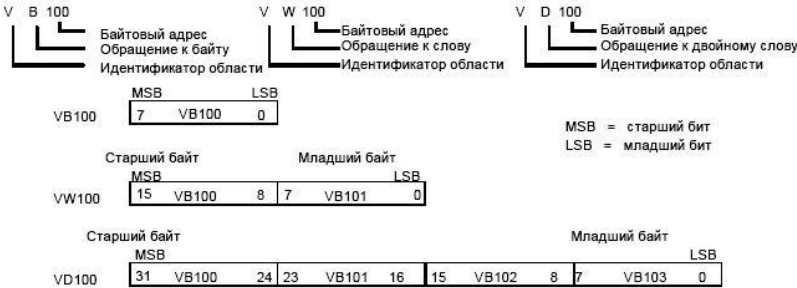


Рис. 5.3. Обращение к одному и тому же адресу в формате байта, слова и двойного слова

5.3 Обращение к данным в областях памяти Регистр входов образа процесса: I

В начале каждого цикла ПЛК S7–200 опрашивает физические входы и записывает полученные значения в регистр входов образа процесса. К образу процесса можно обратиться в формате бита, байта, слова и двойного слова:

Бит: $I[\text{адрес байта}].[адрес бита]$ I0.1

Байт, слово или двойное слово: $I[\text{длина}][\text{начальный адрес байта}] IB4$

Регистр выходов образа процесса: Q

В конце цикла ПЛК копирует значения, хранящиеся в регистре выходов образа процесса, в физические выходы. К образу процесса можно обратиться в формате бита, байта, слова и двойного слова:

Бит: $Q[\text{адрес байта}].[адрес бита]$ Q1.1

Байт, слово или двойное слово: $Q[\text{длина}][\text{начальный адрес байта}] QB5$

Область памяти переменных: V

Память переменных можно использовать для хранения промежуточных результатов операций, выполняемых в программе. В памяти переменных можно хранить также другие данные, имеющие отношение к процессу или к решению выполняемой задачи автоматизации. К памяти переменных можно обратиться в формате бита, байта, слова и двойного слова:

Бит: $V[\text{адрес байта}].[адрес бита]$ V10.2

Байт, слово или двойное слово: $V[\text{длина}][\text{начальный адрес байта}] VW100$

Область битовой памяти: M

Биты памяти (меркеры) можно использовать как управляющие реле для хранения промежуточных результатов операций или другой управляющей информации. К битам памяти можно обратиться в формате бита, байта, слова и двойного слова:

Бит: $M[\text{адрес байта}].[адрес бита]$ M26.7

Байт, слово или двойное слово: $M[\text{длина}][\text{начальный адрес байта}] MD20$

Таймеры: T

ПЛК S7–200 имеет в своем распоряжении таймеры, которые отсчитывают приращения времени с разрешениями (шагами базы времени) 1 мс, 10 мс или 100 мс. С таймером связаны две переменные:

- Текущее значение: это 16-битовое целое со знаком хранит количество времени, отсчитанное таймером.

- Бит таймера устанавливается или сбрасывается, когда текущее значение становится равным предустановленному значению. Предустановленное значение вводится как часть таймерной команды. Обращение к обоим этим элементам данных происходит через адрес таймера ($T + \text{номер таймера}$). Происходит ли обращение к биту таймера или к текущему значению зависит от используемой команды: команды с операндами в битовом формате обращаются к биту таймера, тогда как команды с операндами в формате слова обращаются к текущему значению. Как показано на рис. 5.4, команда «Нормально открытый контакт» обращается к биту таймера, а команда «Передать слово» обращается к текущему значению таймера.

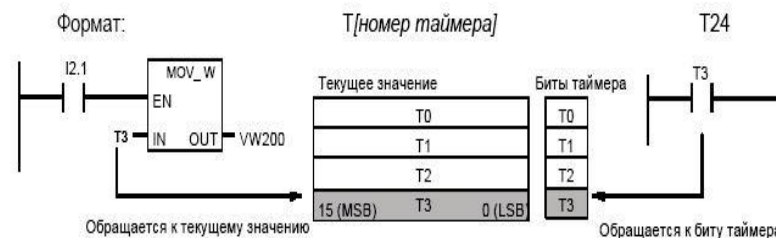


Рис. 5.4. Обращение к биту или к текущему значению таймера

Счетчики: C

ПЛК S7–200 имеет в своем распоряжении три вида счетчиков, которые подсчитывают нарастающие фронты на счетных входах счетчика: один вид счетчиков ведет прямой счет, другой считает только в обратном направлении, а третий вид считает в обоих направлениях. Со счетчиком связаны две переменные:

- Текущее значение: это 16-битовое целое со знаком хранит счетное значение, накопленное счетчиком.

- Бит счетчика устанавливается или сбрасывается, когда текущее значение становится равным предустановленному значению. Предустановленное значение вводится как часть команды счетчика.

Обращение к обоим этим элементам данных происходит через адрес счетчика ($C + \text{номер счетчика}$). Выполняется ли обращение к биту счетчика или к текущему значению, зависит от ис-

пользуемой команды: команды с операндами в битовом формате обращаются к биту счетчика, тогда как команды с операндами в формате слова обращаются к текущему значению. Как показано на рис. 5.5, команда «Нормально открытый контакт» обращается к биту счетчика, а команда «Передать слово» обращается к текущему значению счетчика.

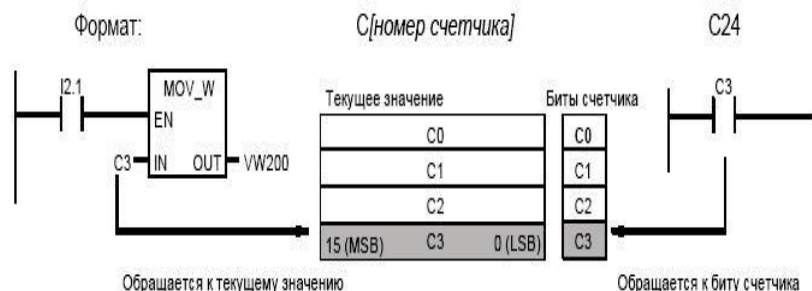


Рис. 5.5. Обращение к биту или к текущему значению счетчика

Скоростные счетчики: HC подсчитывают быстрые события независимо от цикла CPU. Скоростные счетчики имеют в своем распоряжении 32-битовое целое счетное значение (текущее значение). Для обращения к счетному значению скоростного счетчика необходимо ввести его адрес, указав область памяти (HC) и номер счетчика (напр., HC0). Текущее значение скоростного счетчика защищено от записи и может быть адресовано только в формате двойного слова (32 бита).

Формат: $HC[\text{номер скоростного счетчика}]$ HC1

Аккумуляторы: AC – это элементы чтения/записи, которые могут использоваться как память. Например, вы можете использовать аккумуляторы для передачи параметров в подпрограммы и из них или для хранения промежуточных результатов расчетов. ПЛК S7-200 имеет в своем распоряжении четыре 32-битовых аккумулятора (AC0, AC1, AC2 и AC3). К данным в аккумуляторах можно обратиться в формате бита, слова или двойного слова. Длина данных, к которым производится обращение, зависит от команды, которая используется для обращения к аккумулятору. Как показано на рис. 5.6, при обращении к аккумулятору в формате бита или слова используются младшие 8 или 16 битов значения, хранящегося в аккумуляторе. При обращении к аккумулятору в формате двойного слова используются все 32 бита.

Формат: $AC[\text{номер аккумулятора}]$ AC0

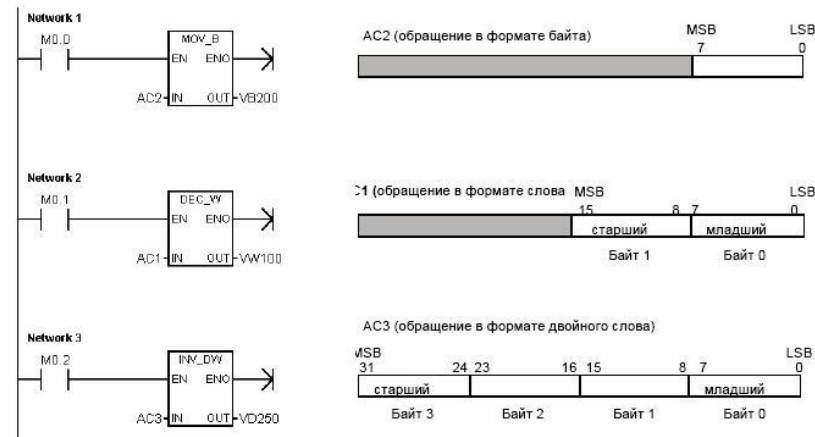


Рис. 5.6. Обращение к аккумуляторам

Специальные биты памяти: SM предоставляют средство для обмена данными между CPU и выполняемой программой. Можно использовать эти биты для выбора и управления некоторыми специальными функциями CPU S7-200, например: бит, который устанавливается только в первом цикле; бит, который устанавливается и сбрасывается с фиксированной частотой, или бит, который указывает на состояние арифметической или иной команды. (Подробную информацию о специальных битах памяти вы найдете в прил. D.) К SM-битам можно обращаться в формате бита, слова или двойного слова:

Бит: $SM[\text{адрес байта}].[адрес бита]$ SM0.1

Байт, слово или двойное слово: $SM[\text{длина}][\text{начальный адрес байта}]$ SMB86

Память локальных данных: L

ПЛК S7-200 имеет в своем распоряжении 64 байта локальной памяти, из которых 60 могут быть использованы в качестве промежуточной памяти или для передачи формальных параметров в подпрограммы.

Память локальных данных похожа на память переменных с одним существенным отличием. Память переменных доступна глобально, тогда как память локальных данных доступна локально. Глобальная доступность означает, что к адресу в этой области

памяти можно обратиться из любой организационной единицы программы (из основной программы, подпрограммы или подпрограмм обработки прерываний). Локальная доступность означает, что эта область памяти ставится в соответствие определенной организационной единице программы. ПЛК S7–200 выделяет 64 байта локальной памяти для главной программы, 64 байта — для каждого уровня вложенности подпрограмм и 64 байта — для про-грамм обработки прерываний.

К области локальных данных, поставленной в соответствие основной программе, не имеют доступа подпрограммы и программы обработки прерываний. Подпрограмма не может обращаться к области локальных данных основной программы, программы обработки прерываний или другой подпрограммы. Аналогично программа обработки прерываний не имеет доступа к области локальных данных основной программы или подпрограммы.

ПЛК S7–200 выделяет область локальных данных по мере необходимости. Это значит, что при выполнении основной программы области локальных данных для подпрограмм и программ обработки прерываний не существуют. Если возникает прерывание или вызывается подпрограмма, то по потребности выделяется локальная память. Вновь выделенная локальная память может снова использовать те же адреса, которые использовались другой подпрограммой или программой обработки прерываний.

ПЛК S7–200 не инициализирует область локальных данных к моменту ее назначения, поэтому она может содержать любые значения. Если при вызове подпрограммы передаются формальные параметры, то ПЛК сохраняет значения передаваемых параметров в соответствующих адресах области локальных данных, выделенной этой подпрограмме.

Адреса в области локальных данных, которые не получили значений при передаче формальных параметров, не инициализируются и при выделении могут содержать произвольные значения.

Бит: $L[\text{адрес байта}].[адрес бита] L0.0$

Байт, слово или двойное слово: $L[\text{длина}] [\text{начальный адрес байта}] LB33$

5.4 Косвенная адресация областей памяти S7–200 с помощью указателей

Косвенная адресация использует указатель для доступа к данным в памяти. Указатели — это ячейки памяти, имеющие размер двойного слова, которые содержат адрес другой ячейки памяти.

В качестве указателей можно использовать только ячейки памяти переменных и локальных данных или аккумуляторные регистры (AC1, AC2 или AC3). Для создания указателя необходимо использовать команду «Переместить двойное слово». Эта команда передает адрес косвенно адресованной ячейки памяти в ячейку указателя. Указатели могут также передаваться в подпрограмму в качестве параметров.

ПЛК S7–200 дает возможность использования указателей для косвенной адресации следующих областей памяти: I, Q, V, M, S, AI, AQ, SM, T (только текущее значение) и C (только текущее значение). Косвенную адресацию нельзя использовать для обращения к отдельному биту или к областям памяти HC или L.

Если необходимо косвенно обратиться к данным, расположенным по некоторому адресу в памяти, можно создать указатель на этот адрес, введя амперсанд (&) и соответствующий адрес. Входному операнду команды должен предшествовать амперсанд (&), чтобы указать на необходимость перемещения в ячейку, обозначенную в выходном операнде команды (указателе), адреса ячейки памяти, а не ее содержимого. Ввод астериска (*) перед операндом команды указывает, что этот операнд является указателем. Как показано на рис. 5.7, ввод *AC1 указывает, что AC1 является указателем на слово, на которое ссылается команда «Переместить слово» (MOVW). В этом примере значения, хранящиеся в VB200 и VB201, перемещаются в аккумулятор AC0.

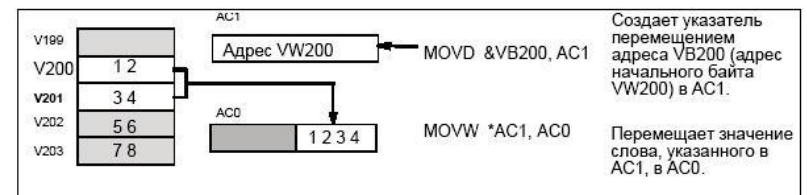


Рис. 5.7 Создание и использование указателя

Как показано на рис. 5.8, можно изменить значение указателя. Так как указатели имеют размер 32 бита, то для изменения значений указателей используйте операции над двойными словами. Для изменения значений указателей могут использоваться такие простые математические операции, как сложение или инкрементирование. Косвенную адресацию с помощью указателей удобно использовать в часто повторяющихся операциях, например, ког-

да необходимо извлекать из памяти циклические управляющие коды, при этом через указатель выполняется увеличение (инкрементирование) или уменьшение (декрементирование) программного счетчика.

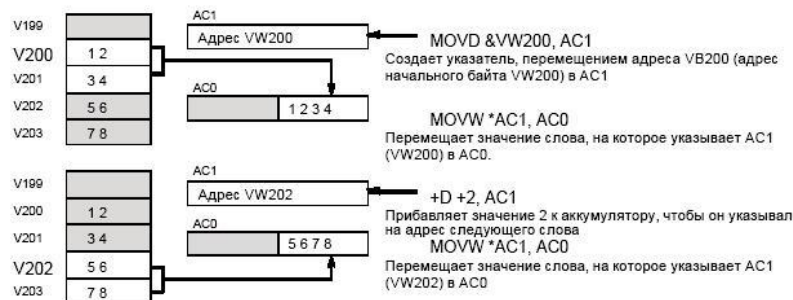


Рис. 5.8. Изменение указателя

5.5 Установка режима работы CPU S7-200

ПЛК S7-200 имеет два режима работы: *STOP* и *RUN*. Индикаторы состояния на передней панели CPU указывают на текущий режим работы. В состоянии *STOP* ПЛК не выполняет программы, и можно загрузить в CPU программу или конфигурацию CPU. В режиме *RUN* ПЛК исполняет программу.

Для изменения режима работы ПЛК снабжен переключателем режимов. С помощью переключателя режимов (он находится под передней крышкой CPU) можно установить режим работы вручную: установка переключателя режимов в *STOP* прекращает исполнение программы; установка переключателя режимов в *RUN* запускает исполнение программы, а установка переключателя режимов в режим *TERM* (терминал) не изменяет режима работы.

Если питание прерывается, когда переключатель режимов находится в положении *STOP* или *TERM*, ПЛК при восстановлении питания автоматически переходит в состояние *STOP*. Если питание прерывается, когда переключатель режимов находится в положении *RUN*, ПЛК при восстановлении питания переходит в режим *RUN*.

STEP 7-Micro/WIN в режиме *online* дает возможность изменить режим работы ПЛК S7-200. Чтобы это программное обеспечение могло управлять режимом работы, необходимо вручную перевести переключатель режимов работы на CPU в положение *TERM*

или *RUN*. Для изменения режима работы вы можете использовать команды меню **PLC > STOP [ПЛК > STOP]** или **PLC > RUN [ПЛК > RUN]** или соответствующие кнопки на панели инструментов.

Для перевода ПЛК S7-200 в состояние *STOP* можно использовать в своей программе команду *STOP*. Это позволяет прекратить исполнение своей программы в зависимости от логики обработки программы.

6 Лабораторные стенды на базе ПЛК для выполнения лабораторных работ

Для выполнения лабораторных работ, представленных в данном учебно-методическом пособии, используются учебные стенды, в основе которых используется ПЛК семейства S7-200 фирмы Siemens. В составе каждого стенда используется персональный компьютер с предустановленной программой *STEP 7-Micro/WIN*, для разработки программного приложения и загрузки исполняемого кода в контроллер.

6.1 Стенд CLC-221

Назначение:

Стенд CLC-221 [4] разработан на базе контроллера CPU 221 семейства S7-200 и предназначен для изучения элементов и систем автоматизации, языка программирования *STEP 7*, пакета программирования *MicroWin*, а также для выполнения лабораторных работ по курсу «Автоматизация технологических процессов и производств» (рис. 6.1).

Состав стенда:

Стенд состоит из базового шкафа управления (БШУ), дополнительных выносных устройств (ДВУ) и персонального компьютера (ПК). Такое деление позволяет обеспечить создание различных объектов управления и разнообразие программ, решающих задачи автоматизации этих объектов.

БШУ имеет следующие габаритные размеры: 1200 × 600 × 120мм.

Состав аппаратуры размещаемой в БШУ:

- ПЛК CPU 221, 220B;
- двигатель DC 12V;

- устройство коммутации;
- коммутационное поле (набор тумблеров и кнопок);
- автомат вводной (двухфазный, 16А, 220В);
- релейный модуль;
- блок питания низковольтный (типа АТХ 300);
- устройство PPI — интерфейс с внешним РС;
- набор коммутационных колодок;
- индикационный модуль.

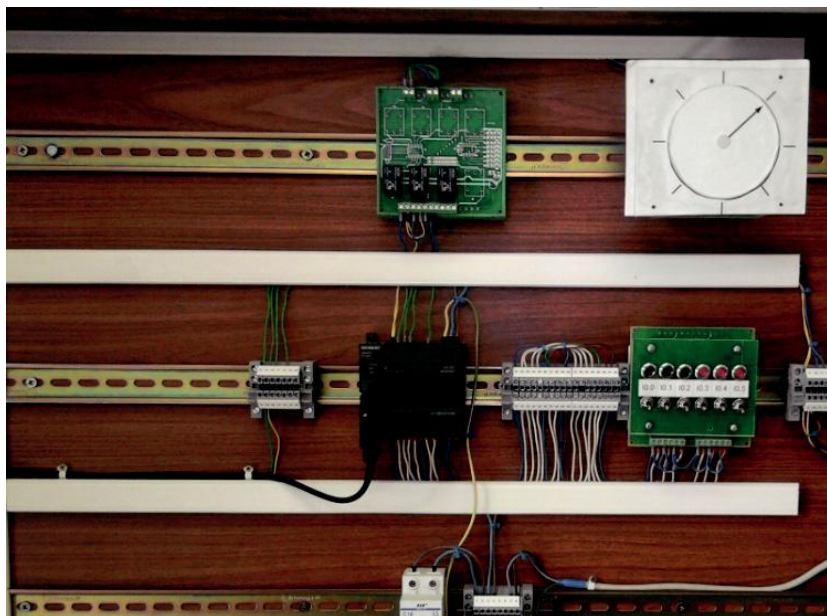


Рис. 6.1. Общий вид стенда CLC-221

Программируемый логический контроллер (ПЛК) CPU-221 ПЛК CPU 221 — наиболее простой и дешевый центральный процессор семейства, предназначенный для построения автоном-ных узлов локальной автоматики. Оснащен шестью встроенными дискретными входами и четырьмя дискретными выходами (см. табл. 3.2). Данный ПЛК не позволяет производить подключение модулей расширения ввода-вывода. Технические характеристики представлены в табл. 6.1.

Таблица 6.1

<i>Технические характеристики</i>	<i>Количество</i>	<i>Примечание</i>
Количество входов процессора	6	Изоляция вход-ных цепей — оптоэлектронная
<i>Напряжение питания входных цепей</i>		
номинальное значение, В	24	
входное напряжение логической единицы, В	15..35	
входное напряжение логического нуля, В	0...5	
Количество входов в группе	2 и 4	
Входной ток логической единицы, мА	до 4	
<i>Задержка распространения</i>		
для стандартных входов (конфигурируется), мс	0.2 ... 12.8	
для входов прерывания (I0.0...I0.3), мс	0.2 ... 12.8	
для счетных входов (вкл./откл.) (I0.0 ... I0.5), мс	0.03	
Количество выходов процессора	4	Тип выходного ключа - реле
<i>Напряжение питания выходных цепей L+/L1:</i>		
Номинальное значение, В	24	
Допустимый диапазон изменений, В	20.4..28.8	
Выходное напряжение логической единицы, В	18.6	
<i>Изоляция - оптоэлектронная</i>		
Количество выходов в группе	4	
<i>Максимальный выходной ток лог. 1</i>		
при температуре 45°C, А	0.75	
при температуре 55°C, А	0.75	
Максимальный выходной ток логического нуля, мА	0.1	
<i>Максимальный суммарный выходной ток:</i>		
при температуре 45°C, А	3.00	
при температуре 55°C, А	3.00	
<i>Задержка включения:</i>		
стандартных выходов Q0.2, Q0.3, мс	15	
импульсных выходов Q0.0, Q0.1, мс	2	
<i>Задержка отключения:</i>		
стандартных выходов Q0.2, Q0.3, мс	100	
импульсных выходов Q0.0, Q0.1, мс	10	

Габариты 90 × 80 × 62мм.

Масса 0.27кг.

Память программ:

- объем 4Кбайт/1.3К инструкций;

- тип энергонезависимая, *EEPROM*;

Объем памяти данных 2048 слов;

Защита данных необслуживаемая;

- запись *DB1* во встроенное *EEPROM*;

- сохранение данных, состояний флагов, счетчиков и таймеров в *RAM* с подпиткой от буферного конденсатора или от буферной батареи (если она установлена).

Время сохранения данных:

- типовое — 50 часов (без буферной батареи); 200 дней (с буферной батареей);

- минимальное— 8 часов при 40°C.

Время заряда буферного конденсатора не менее 20 минут (до 60% емкости).

Языки программирования: STEP 7 Micro/Win. Методы представления программ : STL, LAD и FBD. Организация программы 1×OB1, 1×DB, 1×SDB, подпрограммы с или без передачи параметров.

Методы выполнения программы:

- циклический (*OB1*);

- по аппаратным прерываниям;

- по временным прерываниям (1 ... 255мс).

Количество уровней вложения подпрограмм, до

64. *Парольная защита программы* — 3-уровневая.

Набор команд:

основной: — логические операции, адресация результата, сохранение, счет, загрузка;

— передача, сравнение, сдвиг, вращение, вызов подпрограмм с передачей параметров.

расширенный: — инструкции управления ШИМ и ЧИМ, инструкции переходов, циклов, преобразования типов данных;

— арифметические инструкции сложения, вычитания, умножения, деления, извлечения квадратного корня (целочисленная математика и математика с плавающей запятой).

Время выполнения логической команды — 0.37мкс.

Контроль времени цикла 300мс

(перенастраиваемое). *Количество флагов:*

- общее: 256;

- с сохранением состояний при сбоях в питании в *EEPROM*: 0 ... 112 (конфигурируемый параметр). В *RAM* с подпиткой от конденсатора или батареи: 0 ... 255 (конфигурируемый параметр).

Количество счетчиков:

- общее: 256;

- с сохранением состояний при сбоях в питании в *RAM* с подпиткой от конденсатора или батареи: 0 ... 255 (конфигурируемый параметр);

- числовой диапазон счета — 0 ... 32767.

Количество таймеров:

- общее: 256;

- с сохранением состояний при сбоях в питании в *RAM* с подпиткой от конденсатора или батареи: 0 ... 63 (конфигурируемый параметр);

- диапазоны выдержек времени 4 таймера: 1мс ... 30с. 16 таймеров: 10мс ... 5мин. 236 таймеров: 100мс ... 54мин.

Коммуникационный интерфейс:

- количество - 1;

- тип - RS 485.

Режимы работы:

- режим *PPI* интерфейса для программирования контроллера и подключения программатора, компьютера (через *PC/PPI* кабель), текстового дисплея *TD 200* или панели оператора, связи с другим *S7-200*. Скорость передачи: 9.6, 19.2 или 187.5 Кбит/с.

Для программирования микроконтроллера *CPU 221* серии *S7-200* используется пакет *STEP 7 Micro/WIN 4.1*. Подключение центрального процессора *S7-200* к программатору или компьютеру производится через *PC/PPI*-кабель или через *MPI*-кабель и коммуникационный процессор *CP 5511/ CP 5611*, установленный в программатор/компьютер. Допускается дистанционное программирование контроллеров *S7-200* через сеть *Industrial Ethernet*. В последнем случае компьютер/программатор должен быть оснащен сетевой *Ethernet*-картой, например, коммуникационным процессором *CP 1512/ CP 1612*.

Двигатель DC 12V

Двигатель *DC 12V* предназначен для проведения лабораторных работ, где изучается программное включение/выключение двигателя и смена направления вращения двигателя.

Устройство коммутации

Устройство коммутации предназначено для задания логических уровней на входы программируемого контроллера, работающего в составе стенда.

Коммутационное поле

Коммутационное поле предназначено для коммутации входных и выходных устройств стенда и позволяет задать тип соединений между различными по функциям наборами клемм.

Для выполнения заданного варианта работы клеммы соответствующих полей соединяются между собой.

Вводной автомат (16А, 220В)

Вводной автомат (16А, 220В) служит для подачи и отключения питания, а также для отключения питания стенда при экстремальных условиях.

Релейный модуль

Релейный модуль предназначен для управления работой двигателя постоянного тока (включение и смена направления вращения).

Блок питания АТХ-300

Блок питания АТХ-300 служит для питания интегральных схем и устройств низковольтной логики.

Индикационное поле

Индикационное поле предназначено для отображения состояния выходов ПЛК в процессе выполнения программы. Индикационное поле содержит ряд светодиодных индикаторов, имеющих разные цвета.

PPI-интерфейс

Все модели центральных процессоров S7-200 оснащены встроенным PPI (*Point To Point Interface*) интерфейсом (RS 485), который может быть использован для подключения программатора, устройств человеко-машинного интерфейса, других контроллеров S7-200. Каналы связи выполняются витой парой. В такой сети один центральный процессор может поддерживать одновременную связь с несколькими центральными процессорами S7-200, текстовыми дисплеями или панелями оператора. Скорость передачи данных может достигать 187.5 Кбит/с. PPI-интерфейс может быть использован для организации связи:

- с программаторами PG702, Field PG, Power PG;
- через PC/PPI-кабель с AT-совместимыми компьютерами;
- с текстовыми дисплеями TD200, панелью оператора TP070, а также панелями оператора OP3, OP7, OP17, OP27 и OP37;
- с другими контроллерами S7-200.

Для более подробного изучения устройства и принципов работы стенда можно использовать [3].

6.2 Стенд CLC-224

Назначение:

Стенд CLC-224 [5] разработан на базе CPU-224 и предназначен для изучения элементов и систем автоматизации, языка программирования STEP 7, пакета программирования MicroWin, а также для выполнения лабораторных работ по курсу «Автоматизация» «Программное обеспечение систем управления» (рис. 6.2).

Состав стенда:

Стенд CLC-224 состоит из базового шкафа управления (БШУ), и персонального компьютера (ПК). Такое деление позволит обеспечить создание различных объектов управления и разнообразие программ, решающих задачи автоматизации этих объектов.

БШУ имеет следующие габаритные размеры: 1200 × 600 × 125 мм. Его корпус выполнен из ламинированной ДСП-плиты. Все устройства устанавливаются в шкаф на DIN-профиль 35/7.5 ДКС 02140. Шкаф имеет две дверцы, закрываемые на замок. Подводка питания (220В) выполнена через специализированный трехконтактный силовой разъем на вводной автомат 16А, 220В. Все корпуса внутренних устройств заземлены. Подготовленная в пакете MicroWin 4.0 программа загружается в CPU-224 через внешний разъем интерфейса PPI.

Состав аппаратуры размещаемой в БШУ:

- Ø CPU-224 — программируемый логический контроллер семейства S7-200 фирмы Siemens;
- Ø вводной автомат С 16 (двухфазный, 16А, 220В);
- Ø блок питания DRP-240-24;
- Ø устройство коммутации (набор тумблеров и кнопок);
- Ø коммутационное поле;
- Ø транзисторный адаптер;
- Ø шаговый двигатель 42SPM — 24DJA;
- Ø устройство PPI-интерфейс с внешним PC.

Основное назначение ПЛК — исполнять загружаемую программу, через анализ входных сигналов, поступающих на объект управления, и создание управляющих воздействий.

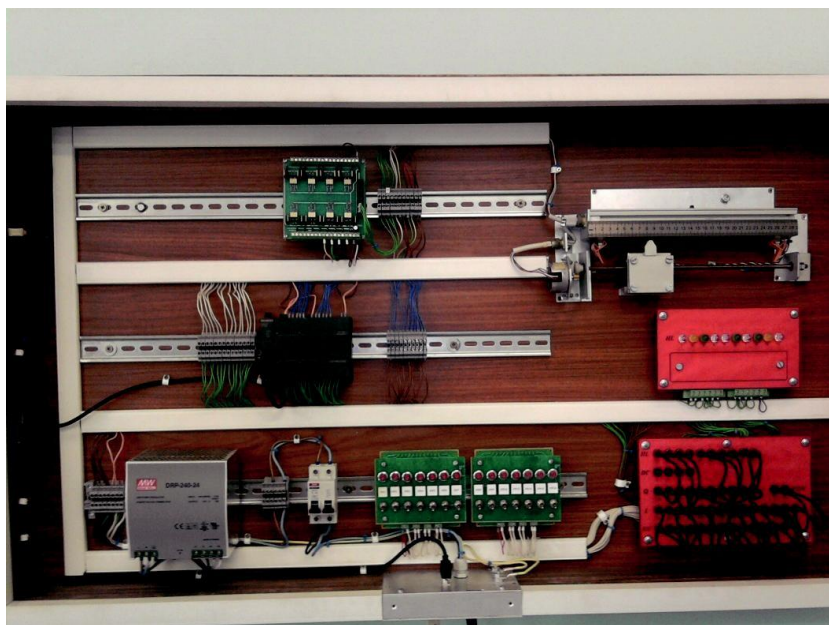


Рис. 6.2. Общий вид стенда CLC-224

Программируемый логический контроллер (ПЛК) CPU-224 ПЛК CPU 224 — блок центрального процессора семейства S7-200, предназначенный для построения автономных узлов локальной автоматики. Оснащен 14 встроенными дискретными входами и 10 дискретными выходами (см. табл. 3.2). Данный ПЛК позволяет производить подключение до 7 модулей расширения ввода-вывода. Технические характеристики представлены в табл. 6.2

Таблица 6.2

<i>Технические характеристики</i>	<i>Количество</i>	<i>Примечание</i>
Количество входов процессора	14	Изоляция входных цепей — оптоэлектронная
<i>Напряжение питания входных цепей</i>		
номинальное значение, В	24	
входное напряжение логической единицы, В	15..35	
входное напряжение логического нуля, В	0...5	
Количество входов в группе	2 и 4	
Входной ток логической единицы, мА	до 4	
<i>Задержка распространения</i>		
для стандартных входов (конфигурируется), мс	0.2 ... 12.8	
для входов прерывания (I0.0...I0.3), мс	0.2 ... 12.8	
для счетных входов (вкл./откл.) (I0.0...I0.5), мс	0.03	
Количество выходов процессора	10	Тип выходного ключа — транзистор, изоляция — оптоэлектронная
<i>Напряжение питания выходных цепей L+/L1:</i>		
Номинальное значение, В	24	
Допустимый диапазон изменений, В	20.4..28.8	
Выходное напряжение логической единицы, В	18.6	
Количество выходов в группе	10	
<i>Максимальный выходной ток лог. 1</i>		
при температуре 45°C, А	0.75	
при температуре 55°C, А	0.75	
Максимальный выходной ток логического нуля, мА	0.1	
<i>Максимальный суммарный выходной ток:</i>		
при температуре 45°C, А	5.00	
при температуре 55°C, А	5.00	
<i>Задержка включения:</i>		
стандартных выходов Q0.2, Q0.3, мс	15	
импульсных выходов Q0.0, Q0.1, мс	2	
<i>Задержка отключения:</i>		
стандартных выходов Q0.2, Q0.3, мс	100	
импульсных выходов Q0.0, Q0.1, мс	10	

Масса 0.37кг.

Память программ:

- объем 8Кбайт/ 1.3К инструкций;
- тип энергонезависимая, *EEPROM*;

Объем памяти данных 8192 слов;

Защита данных необслуживаемая;

- запись *DB1* во встроенное *EEPROM*;

- сохранение данных, состояний флагов, счетчиков и таймеров

в *RAM* с подпиткой от буферного конденсатора или от буферной батареи (если она установлена).

Время сохранения данных:

- типовое — 50 часов (без буферной батареи); 200 дней (с буферной батареей);

- минимальное — 8 часов при 40°C.

Время заряда буферного конденсатора не менее 20 минут (до 60% емкости).

Языки программирования *STEP 7 Micro/Win*. Методы представления программ *STL*, *LAD* и *FBD*. Организация программы 1× *OB1*, 1× *DB*, 1× *SDB*, подпрограммы с или без передачи параметров.

Методы выполнения программы:

- циклический (*OB1*);
- по аппаратным прерываниям;
- по временным прерываниям (1 ... 255мс).

Количество уровней вложения подпрограмм до 64. Парольная защита программы 3-уровневая.

Набор команд:

основной — логические операции, адресация результата, сохранение, счет, загрузка. Передача, сравнение, сдвиг, вращение, вызов подпрограмм с передачей параметров.

расширенный — инструкции управления ШИМ и ЧИМ, инструкции переходов, циклов, преобразования типов данных. Арифметические инструкции сложения, вычитания, умножения, деления, извлечения квадратного корня (целочисленная математика и математика с плавающей запятой).

Время выполнения логической команды 0.37мкс.

Контроль времени цикла 300мс

(перенастраиваемое). Количество флагов:

- общее: 256;

- с сохранением состояний при сбоях в питании в *EEPROM*: 0 ... 112 (конфигурируемый параметр). В *RAM* с подпиткой от конденсатора или батареи: 0 ... 255 (конфигурируемый параметр).

Количество счетчиков:

- общее: 256;

- с сохранением состояний при сбоях в питании в *RAM* с подпиткой от конденсатора или батареи: 0 ... 255 (конфигурируемый параметр);

- числовой диапазон счета 0 ... 32767.

Количество таймеров:

- общее: 256;

- с сохранением состояний при сбоях в питании в *RAM* с подпиткой от конденсатора или батареи: 0 ... 63 (конфигурируемый параметр);

- диапазоны выдержек времени 4 таймера: 1мс... 30с. 16 таймеров: 10мс ... 5мин. 236 таймеров: 100мс ... 54мин.

Коммуникационный интерфейс:

- количество: 1;

- тип — RS 485.

Режимы работы:

- режим *PPI* интерфейса для программирования контроллера и подключения программатора, компьютера (через *PC/PPI* кабель), текстового дисплея *TD 200* или панели оператора, связи с другим *S7-200*. Скорость передачи 9.6, 19.2 или 187.5 Кбит/с.

Для программирования микроконтроллера *CPU 224* серии *S7-200* используется пакет *STEP 7 Micro/WIN 4.1*. Подключение центрального процессора *S7-200* к программатору или компьютеру производится через *PC/PPI* кабель или через *MPI* кабель и коммуникационный процессор *CP 5511/ CP 5611*, установленный в программатор/ компьютер. Допускается дистанционное программирование контроллеров *S7-200* через сеть *Industrial Ethernet*. В последнем случае компьютер/ программатор должен быть оснащен сетевой *Ethernet* картой, например, коммуникационным процессором *CP 1512/ CP 1612*.

Вводной автомат (16А, 220В)

Вводной автомат (16А, 220В) служит для подачи и отключения питания, а так же отключения питания стенда при экстремальных условиях.

Блок питания *DRP-240-24*

Блок питания *DRP-240-24* служит для питания *CPU 224*, датчиков, интегральных схем и устройств низковольтной логики.

Устройство коммутации (набор тумблеров и кнопок)

Устройство коммутации предназначено для задания логических уровней на входы программируемого контроллера, работающего в составе стенда.

Коммутационное поле

Коммутационное поле предназначено для коммутации входных и выходных устройств стенда и позволяет задать тип соединений между различными по функциям наборами клемм.

Для выполнения заданного варианта работы клеммы соответствующих полей соединяются между собой.

Транзисторный адаптер

Транзисторный адаптер служит для согласования выходных уровней напряжения ПЛК и исполнительных устройств (шагового двигателя, индикаторов и т. д.).

Индикационное поле

Индикационное поле предназначено для отображения состояния выходов ПЛК в процессе выполнения программы. Индикационное поле содержит ряд светодиодных индикаторов, имеющих разные цвета.

Шаговый двигатель 42SPM — 24DJA

Шаговый двигатель предназначен для перемещения макета суппорта по винту. Перемещение суппорта ограничено концевыми переключателями, состояние которых может считывать управляющая программа. Кроме того по краям макета размещены концевые выключатели безопасности, которые отключают напряжение питания с шагового двигателя в случае их срабатывания.

PPI-интерфейс

Все модели центральных процессоров *S7-200* оснащены встроенным *PPI (Point To Point Interface)* интерфейсом (*RS 485*), который может быть использован для подключения программатора, устройств человеко-машинного интерфейса, других контроллеров *S7-200*. Каналы связи выполняются витой парой. В такой сети один центральный процессор может поддерживать одновременную связь с несколькими центральными процессорами *S7-200*, текстовыми дисплеями или панелями оператора. Скорость передачи данных может достигать 187.5 Кбит/с. *PPI* интерфейс может быть использован для организации связи: с программа-

торами *PG702*, *Field PG*, *Power PG*, через *PC/PPI* кабель с *AT*-совместимыми компьютерами, с текстовыми дисплеями *TD200*, панелью оператора *TP070*, а также панелями оператора *OP3*, *OP7*, *OP17*, *OP27* и *OP37*, с другими контроллерами *S7-200*.

Для более подробного изучения устройства и принципов работы стенда можно использовать [4].

6.3 Стенд CLA-117

«Пневмопривод» Назначение:

Стенд *CLA-117* «Пневмопривод» [6] (рис. 6.3) предназначен для изучения работы пневмопривода на базе линейных пневмоцилиндров в рамках лабораторных работ по дисциплинам «Пневмо- гидропривод» и «Автоматизация технологических процессов и производств в машиностроении». Пневмоцилиндрами можно управлять как в ручном режиме, формируя управляющее воздействие от тумблеров, так и программно с помощью ПЛК *CPU 224*, включенного в состав стенда.

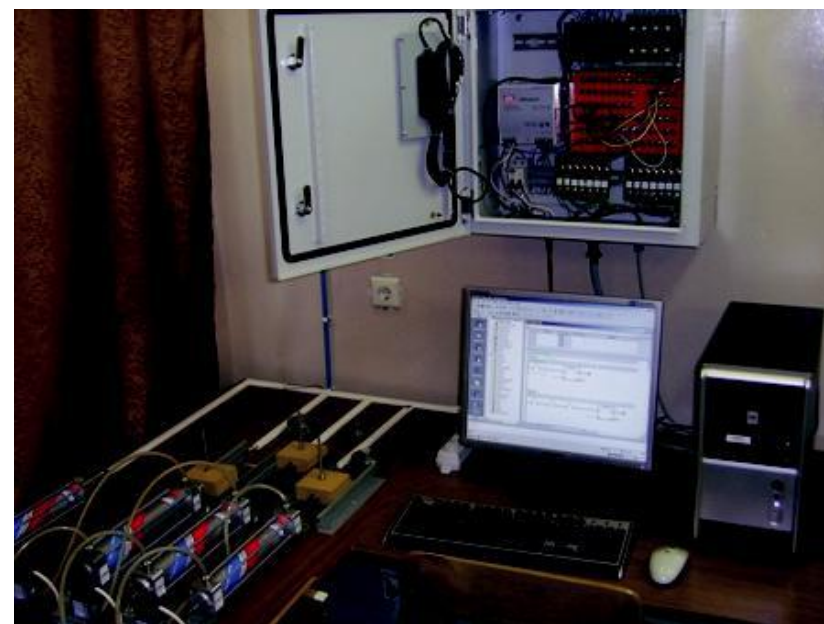


Рис. 6.3. Общий вид стенда *CLA-117*

Состав станда:

Стенд *CLA-117* состоит из базового шкафа управления (БШУ), персонального компьютера (ПК), набора пневмоцилиндров, пневмораспределителей и пневмодросселей.

БШУ имеет следующие габаритные размеры: $600 \times 600 \times 200$ мм. Его корпус выполнен из металла. Все электронные устройства устанавливаются в шкаф на *DIN*-профиль 35/7.5 ДКС 02140. Шкаф имеет дверцу, закрываемую на замок. Подводка питания 220В выполнена через специализированный трех контактный силовой разъем на вводной автомат 16А, 220В. Все корпуса внутренних устройств заземлены. Подготовленная в пакете *MicroWin 4.0* программа загружается в *CPU-224* через внешний разъем интер-фейса *PPI*.

Состав аппаратуры размещаемой в БШУ:

- ПЛК *CPU 224*, 24В;
- устройство коммутации (набор тумблеров и кнопок);
- коммутационное поле
- автомат вводной (16А, 220В);
- блок питания *DRP-240-24*;
- устройство *PPI* - интерфейс с внешним *PC*;
- набор коммутационных колодок;
- индикационный модуль.

Набор аппаратуры размещенной в БШУ соответствует аппаратуре, описанной в п. 6.2, станда *CLC-224*.

Состав внешних исполнительных устройств:

- Пневмоцилиндр 10-032× 0200;
- Пневмоцилиндр 10-040× 0200;
- Пневмоцилиндр 10-050× 0200 *LSO*;
- Пневмораспределители П-Р321-6 (=24В) — 6 шт.;
- Пневмодроссели ПДК-06-2 — 6 шт.;
- Блок подготовки воздуха П-ФПК-10.

Для работы пневмоустройств станда подведена лабораторная пневмомагистраль, по которой поступает сжатый воздух от компрессорной установки К-12. Остановимся подробнее на работе пневмоустройств и способе управления ими. На рис. 6.4 представлена упрощенная конфигурация пневмопривода станда *CLA-117*

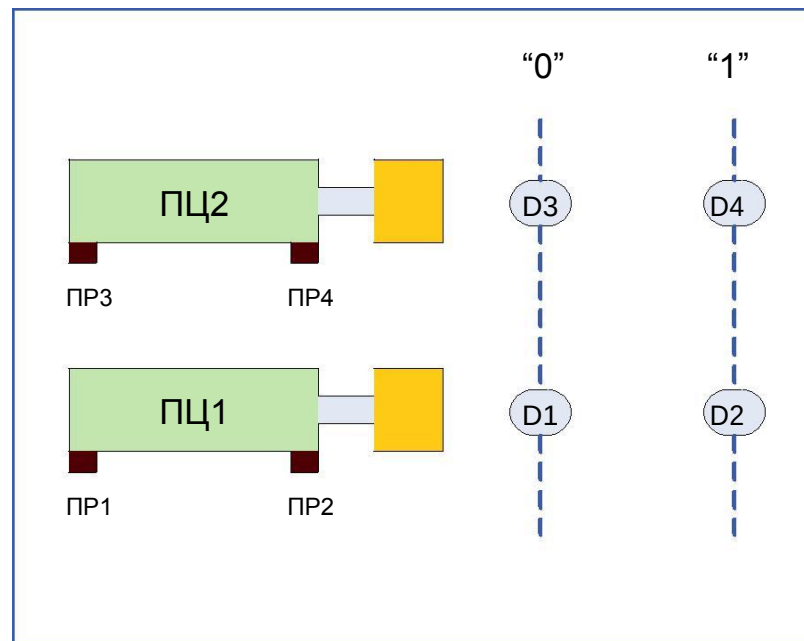


Рис. 6.4. Упрощенная конфигурация пневмопривода станда *CLA-117*

К штоку каждого пневмоцилиндра (показаны ПЦ1 и ПЦ2) прикреплена специальная платформа, которая перемещается по направляющим. На нижней части платформ размещены небольшие магниты, которые замыкают герконовые датчики *D1..D4*, установленные по краям трассы перемещения платформы. Выходы датчиков соединены с входами ПЛК, так чтобы программа могла контролировать перемещение платформы. Для управления работой пневмоцилиндров используются пневмораспределители сжатого воздуха ПР1..ПР4, соединенные с выходами ПЛК. Таким образом, при формировании воздействия на пневмораспределители (открывая сжатый воздух в одну из камер пневмоцилиндров) происходит перемещение платформы, а с помощью датчиков контролируется положение платформы.

Разработанная на ПК программа для ПЛК загружается в контроллер и запускается на исполнение. Для более подробного изучения устройства и принципов работы станда можно использовать [6].

7 Создание приложений для ПЛК семейства S7-200 в пакете Step 7 Micro/WIN ver.4.0

Рассмотрим основные приемы создания управляющего приложения для ПЛК семейства S7-200 CPU-224. На рис. 7.1 представлен интерфейс основного окна (*MAIN*) пакета *Step 7 Micro/WIN ver.4.0*. Краткое изложение возможностей этого пакета рассмотрены в п.4 «Программное обеспечение пакета *Step 7 Micro/WIN ver.4.0*».

7.1 Разработка программы в редакторе LAD

Для начала необходимо вызвать редактор программ *LAD*. Чтобы открыть редактор программ, щелкните на символе *Program Block* [Программный блок] (рис. 7.1). Необходимо обратить внимание на дерево (с левой стороны) команд и редактор программ. Дерево команд используется для вставки команд контактного плана (*LAD*) в сегменты редактора программ путем перемещения команд с помощью мышки из дерева команд в сегменты. Символы на панели инструментов предоставляют возможность быстрого вызова ко-манд меню.

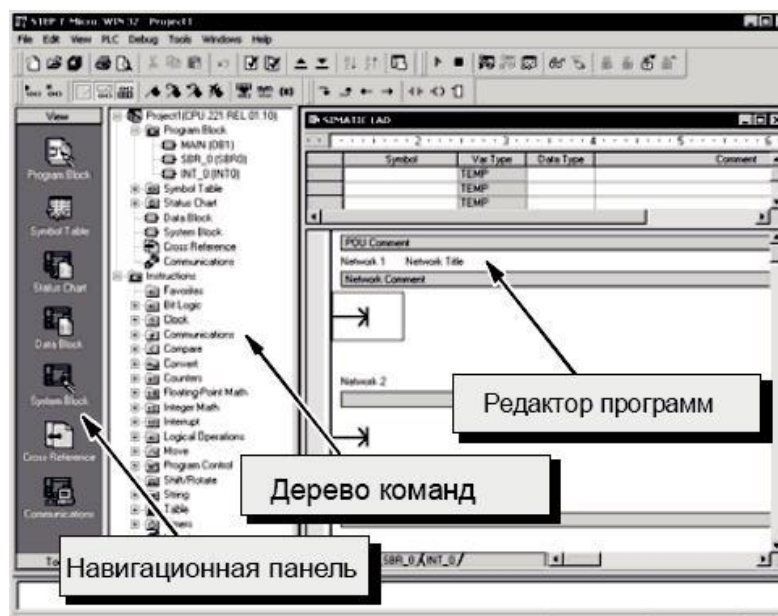


Рис. 7.1. Интерфейс пакета *Step 7 Micro/WIN*

Редактор логических схем *LAD* отображает программу в виде графического представления, имеющего сходство с электрической монтажной схемой. Логические схемы позволяют программе имитировать протекание электрического тока от источника напряжения через ряд логических условий на входах, которые, в свою очередь, активизируют логические условия на выходах. *LAD*-программа включает в себя находящуюся слева шину, находящуюся под напряжением, которая является источником потока сигнала. Замкнутые контакты позволяют потоку сигнала протекать через эти контакты к следующему элементу, а разомкнутые контакты препятствуют протеканию потока сигнала.

В качестве примера разработаем генератор периодических сигналов на выходе **Q0.0** контроллера с программно-изменяемым периодом следования. Период сигнала будет определяться уставкой таймера **T37**, работающего в диапазоне 100 мс.

Реализация алгоритма программы представлена ниже:

1. В строчке программы **Network 1** таймер **T37** запускается **M4.2** (значение бита памяти **M4.2** равно '0' при включении питания контроллера, но поскольку используется инверсия этой переменной — нормально-замкнутый контакт, то переменная **M4.2** активна) на 1,0 с (уставка таймера = 10, диапазон = 100мс).

2. Через 1,0с управляющий бит таймера **T37** становится активным (замыкается нормально разомкнутый контакт, связанный с переменной **T37** — строчка программы **Network 2**) и переменной **M4.2** присваивается активное состояние ('1').

3. В следующем цикле контроллера нормально-замкнутый контакт, связанный с переменной **M4.2**, размыкается (поскольку **M4.2** = '1') и таймер **T37** и его управляющий бит сбрасываются в '0' (**Network 1**), что приводит к переопределению переменной **M4.2** ('0').

4. Таким образом цикл работы ПЛК будет повторяться, а управляющий бит таймера (как впрочем и переменная **M4.2**) будет принимать заданные значения.

5. Для того чтобы визуальным образом проверить правильность работы программы присваивается значение бита **T37** выходу **Q0.0** контроллера (**Network 2**). При этом выход **Q0.0** на коммутационном поле стенда можно соединить с любым индикатором.

Введенная программа изображена на рис. 7.2. В этом примере используется обозначение таймера (**T37**) и его управляющего бита (**T37**). Но в первом случае речь идет о текущем значении тайме-

ра (16-битовое целое со знаком хранит количество времени, отсчитанное таймером), а во-втором — бит таймера, который устанавливается или сбрасывается, когда текущее значение таймера становится равным предустановленному значению. Подробнее о структуре таймеров можно посмотреть в разделе 5.3.

7.2 Сохранение разработанной программы

При сохранении программы создается проект, который включает в себя тип *CPU S7-200* и другие параметры. Для сохранения программы необходимо:

- Выбрать из строки меню команду **File > Save as**.
- В диалоговом окне **Save as** выбрать имя для проекта.
- Для сохранения проекта щелкнуть мышкой **OK**.

Далее можно выполнять компиляцию проекта.

7.3 Компиляция разработанной программы

Перед началом компиляции необходимо выбрать тип контроллера, на который будет загружаться программа. Разработанная программа будет загружаться на стенд, содержащий контроллер *CPU-224*, поэтому выбирается этот контроллер:

- Выбрать из строки меню команду **PLC > Type**.
- В диалоговом окне выбрать тип контроллера *CPU-224*. Для компиляции проекта выполняются следующие действия:
- Выбрать из строки меню команду **PLC > Compile**
- Результат компиляции будет представлен в нижнем окне.
- Если программа обнаружила ошибки, то их необходимо исправить и повторить компиляцию.

Получившийся после компиляции исполняемый код готов для загрузки в контроллер.

7.4 Загрузка исполняемого кода в контроллер

Каждый проект *Step 7 Micro/WIN* ставится в соответствие конкретному *CPU*. Если тип *CPU*, установленный в проекте, не соответствует подключенному *CPU*, то программа указывает на это несоответствие и предлагает сделать изменения. Для загрузки исполняемого кода необходимо:

- Выбрать из строки меню команду **File > Download**.
- Для загрузки элементов программы нажмите **OK**.

Если ваш *CPU* находится в режиме **RUN**, то программа предложит перейти в режим **STOP** и продолжить загрузку.

7.5 Исполнение программы на контроллере

Для запуска исполнения программы на контроллере необходимо контроллер перевести в режим **RUN**. Проверьте, чтобы выход **Q0.0** на коммутационном поле стенда был соединен с любым индикатором *HL*. При правильно введенной программе, после перевода контроллера *CPU-224* в режим **RUN**, выбранный индикатор будет мигать с периодом в 1,0 с.

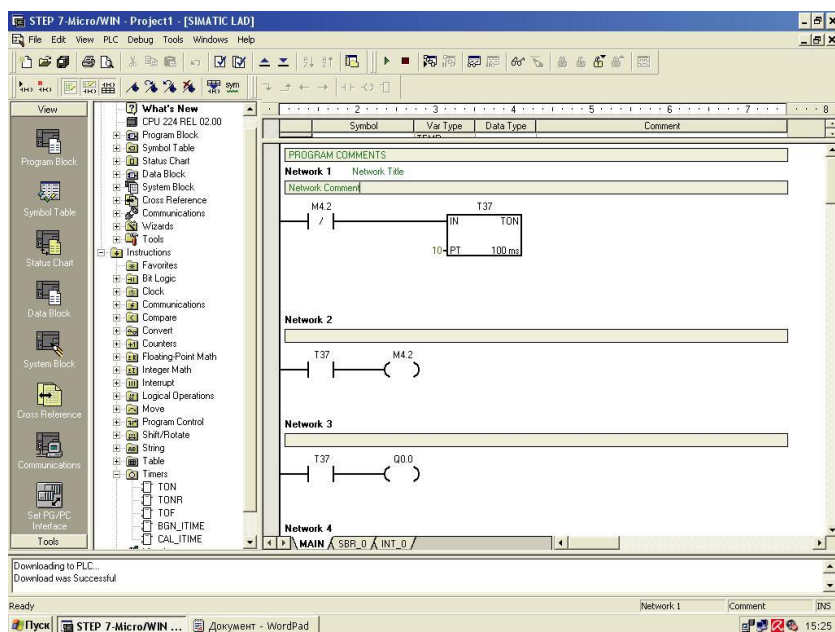


Рис. 7.2. Разработанная программа

8 Лабораторная работа №1

Разработка алгоритма, создание и исследование программы на языке STEP-7 «Управление работой двигателя постоянного тока по заданному циклу»

Цель работы — разработка цикловых управляющих программ на языке STEP-7, с использованием стандартных инструкций языка. В качестве объектов управления могут быть любые устройства электроавтоматики станка

8.1 Порядок выполнения лабораторной работы

ВАЖНО! К выполнению лабораторной работы на стенде CLC-221 допускаются студенты прошедшие инструктаж по Технике Безопасности

8.1.1. Изучить описание стенда, данное в настоящем документе и описание функциональных устройств, представленных в Приложении [4]. При изучении использовать стенд «CPU-221».

8.1.2. По описанию функциональных устройств изучить правила коммутации индикаторов и задающих кнопок, представленных на индикационном поле и на устройстве коммутации.

8.1.3. Подготовить стенд к работе, для чего:

- включить персональный компьютер (ПК), соединенный интерфейсом *PPI* с программируемым логическим контроллером *CPU-221*;

- загрузить пакет *STEP-7 Micro/Win v 4.0*;
- включить вводной автомат *C16* стенда, при этом на ПЛК загорится желтый индикатор (состояние ПЛК *STOP*).

8.1.4. Подготовить разработанную программу для загрузки в ПЛК, для чего:

- ввести исходный текст программы, разработанной на основании задания;
- откомпилировать программу, выбрав в меню пакета *PLC* и далее выбрать пункт **Compile**;
- в случае появления ошибок компиляции, устранить их и снова повторить предыдущий пункт;
- проверить коммуникационное соединение персонального компьютера и ПЛК, для чего в меню *PLC* пакета выбрать пункт **Type**. После нажатия откроется диалоговое меню, где необходимо нажать **Read PLC**. При правильном соединении ПК и ПЛК в окошке **PLC Type** будет написано: *CPU 221*;
- проверить состояние ПЛК, который должен при загрузке программы находиться в состоянии *STOP*;

- выбрать в меню **File** пакета *STEP-7 Micro/Win* пункт

Download. После нажатия откроется диалоговое меню **Download** где в пункте *Options* напротив требуемых для загрузки блоков (*Program Block, Data Block, System Block*) поставить галочки. Нажать кнопку **Download** и наблюдать процесс загрузки программы в ПЛК.

8.1.5. С помощью коммутационного поля установить требуемые соединения между индикаторами и выходами ПЛК и между задающими кнопками и входами ПЛК.

8.1.6. Перевести работу ПЛК в режим *RUN*, для этого в меню *PLC* пакета выбрать пункт **RUN**. После нажатия этой кнопки программа, загруженная в ПЛК, начинает работать.

8.1.7. Наблюдать работу двигателя постоянного тока по заданному алгоритму.

8.2 Задание для выполнения лабораторной работы

Разработать программу на языке STEP-7 программируемого автомата, который управляет набором объектов и работает по цикловому алгоритму.

Дано:

В качестве объектов управления используется двигатель постоянного тока и два индикатора (*HL1...HL2*). Задающие кнопки (*SW1...SW6*) используются для инициализации программы и останова.

Необходимо:

Обеспечить работу двигателя по цикловому алгоритму.

1. При включении ПЛК в режим *RUN* ни один индикатор не горит.

2. После нажатия кнопки инициализации *SW* (номер кнопки **START** задается в таблице), через заданное время *T0* загорается первый индикатор и горит время *T1*.

3. По завершению интервала *T1* первый индикатор гаснет и начинает вращаться двигатель по часовой стрелке — время *T2*.

4. По завершению интервала *T2* двигатель останавливается и загорается второй индикатор, который горит время *T3*.

5. По завершению интервала *T3* второй индикатор гаснет и включается двигатель с вращением против часовой стрелки, который работает время *T4*.

6. По завершению интервала *T4* двигатель останавливается и начинается новый цикл, при этом загорается первый индикатор.

7. Количество циклов N работы объектов установлено заданием.

8. При нажатии стоповой кнопки SW (номер кнопки **STOP** задается в таблице) в любой момент работы программы, циклы останавливаются. При нажатии кнопки инициализации начинает реализовываться алгоритм, начиная с п. 2.

Таблица 8.1

Варианты заданий лабораторной работы №1

№ вар-та	Первый индикатор	Второй индикатор	$T0$	$T1$	$T2$	$T3$	$T4$	SW START	SW STOP	N
1	HL1	HL2	1	2	3	4	0.5	SW1	SW2	2
2	HL2	HL3	2	3	1	0.5	3	SW2	SW3	3
3	HL3	HL4	3	2	0.5	1	4	SW3	SW4	4
4	HL4	HL1	4	3	2	0.5	1	SW5	SW6	2
5	HL2	HL3	1	3	2	4	2	SW1	SW5	4
6	HL4	HL3	2	4	3	0.5	3	SW2	SW4	2
7	HL2	HL1	3	2	0.5	1	4	SW3	SW6	3
8	HL1	HL4	2	3	1	0.5	3	SW4	SW5	4
9	HL1	HL3	1	4	2	3	0.5	SW2	SW5	2
10	HL4	HL2	0.5	4	3	2	1	SW4	SW2	4
11	HL2	HL4	2	3	3	2	4	SW1	SW6	4
12	HL3	HL1	2	0.5	1	1	3	SW5	SW4	3
13	HL3	HL1	2	3	1	0.5	3	SW2	SW1	4
14	HL4	HL2	0.5	4	3	2	1	SW3	SW2	3
15	HL2	HL1	4	1	1	3	2	SW4	SW1	2
16	HL3	HL4	2	2	0.5	3	1	SW5	SW1	3

8.3 Требования к выполнению отчета

Отчет по лабораторной работе выполняется в отдельной тетради и должен содержать:

- Название лабораторной работы, ее цель, задачи.
- Вариант задания.
- Таблица соответствия входов и выходов контроллера битам используемой памяти и функциональному назначению.
- Исходные тексты программы в редакторе LAD и в редакторе STL с комментариями.

8.4 Вопросы для самопроверки

- Основные функциональные узлы станда $CLC-221$.
- Основные инструкции языка $STEP-7$ ($MOVB$, TON , CTU , $LD = QX.X$).
- Входные и выходные каскады дискретных входов и выходов $CPU-221$.
- Организация промышленной связи в $SIMATIC S7-200$.
- Настройка параметров основного меню пакета $STEP 7 Micro/WIN$.
- Как откомпилировать и загрузить программу в ПЛК в пакете $STEP 7 Micro/WIN$.

9 Лабораторная работа №2

Разработка алгоритма, создание и исследование программы «Управление работой светофоров на перекрестке»

Цель работы — разработка цикловых управляющих программ на языке *STEP-7*, решающих задачи автоматизации объектов средней сложности.

9.1 Порядок выполнения лабораторной работы

ВАЖНО! К выполнению лабораторной работы на стенде CLC-224 допускаются студенты прошедшие инструктаж по Технике Безопасности

9.1.1. Изучить описание стенда, данное в настоящем документе и описание функциональных устройств, представленных в [5]. При изучении использовать стенд «CPU-224».

9.1.2. По описанию функциональных устройств изучить правила коммутации индикаторов и задающих кнопок, представленных на индикационном поле и на устройстве коммутации соответственно.

9.1.3. Подготовить стенд к работе, для чего:

- Включить персональный компьютер (ПК), соединенный интерфейсом *PPI* с программируемым логическим контроллером *CPU-224*.

- Загрузить пакет *STEP-7 Micro/Win v 4.0*.

- Включить вводной автомат *C16* стенда, при этом на ПЛК загорится желтый индикатор (состояние ПЛК *STOP*).

9.1.4. Подготовить разработанную программу для загрузки в ПЛК, для чего:

- Ввести исходный текст программы, разработанной на основании задания.

- Откомпилировать программу, выбрав в меню пакета *PLC* и далее выбрать пункт **Compile**.

- В случае появления ошибок компиляции, устранить их и снова повторить предыдущий пункт.

- Проверить коммуникационное соединение персонального компьютера и ПЛК, для чего в меню *PLC* пакета выбрать пункт **Type**. После нажатия откроется диалоговое меню, где необходимо нажать **Read PLC**. При правильном соединении ПК и ПЛК в окошке **PLC Type** будет написано: *CPU 224*.

- Проверить состояние ПЛК, который должен при загрузке программы находиться в состоянии *STOP*.

- Выбрать в меню **File** пакета *STEP-7 Micro/Win* пункт **Download**. После нажатия откроется диалоговое меню **Download** где в пункте **Options** напротив требуемых для загрузки блоков (*Program Block, Data Block, System Block*) поставить галочки. Нажать кнопку **Download** и наблюдать процесс загрузки программы в ПЛК.

9.1.5. С помощью коммутационного поля установить требуемые соединения между индикаторами и выходами ПЛК и между задающими кнопками и входами ПЛК.

9.1.6. Перевести работу ПЛК в режим *RUN*, для этого в меню *PLC* пакета выбрать пункт **RUN**. После нажатия этой кнопки программа, загруженная в ПЛК, начинает работать.

9.1.7. Нажать «пешеходную» кнопку и наблюдать за работой светофоров.

9.2 Задание для выполнения лабораторной работы

Разработать программу на языке *STEP-7* для автоматизации работы светофоров на перекрестке. Работа пешеходного светофора инициируется кнопкой.

Дано:

Входы контроллера соединены с пешеходными кнопками, работу которых имитируют кнопки *SW*. Выходы *QX.X* (заданные таблицей) контроллера управляют работой индикаторов светофоров и индицируют его состояние.

Необходимо:

Обеспечить работу светофора по цикловому алгоритму. Индицировать работу светофоров следующим образом:

- Первый индикатор — красный автомобильный светофор.
- Второй индикатор — желтый автомобильный светофор.
- Третий индикатор — зеленый автомобильный светофор.
- Четвертый индикатор (пешеходный) — выключен — *STOP*, включен — *GO*.

Алгоритм работы программы следующий:

1. При включении ПЛК в режим *RUN* ни один индикатор не горит.

2. С помощью кнопки инициализации *SW (START)*, необходимо выполнить инициализацию программы, так чтобы горел зеленый автомобильный и красный пешеходный.

3. При нажатии на любую «пешеходную» кнопку включается таймер *T0* на время *T0*.

4. При срабатывании управляющего бита таймера 1, загорается желтый светофор, который горит время **T1**.

5. После завершения работы таймера 1, включается красный свет автомобильного светофора и зеленый (GO) пешеходного на время **T2**. После срабатывания таймера 2, загорается красный пешеходный и желтый автомобильный.

6. Желтый светофор горит время **T3** и все светофоры возвращаются в начальное состояние.

7. Число циклов **N** задается в таблице.

8. При нажатии на стоповую кнопку в произвольное время выполняется безусловный сброс светофоров в исходное состояние.

• Исходные тексты программы в редакторе *LAD* и в редакторе *STL* с комментариями.

9.4 Вопросы для самопроверки

- Основные функциональные узлы станков *CLC-221*, *CLC-224*.
- Основные инструкции языка *STEP-7 (MOV, TON, CTU, LD =QX.X.)*
- Настройка параметров основного меню пакета *STEP 7 Micro/ WIN*.
- Основные способы доступа к данным *CPU-224*.
- Как откомпилировать и загрузить программу в ПЛК в пакете *STEP 7 Micro/WIN*.

Таблица 9.1

Варианты заданий лабораторной работы №2

№	Первый индикатор	Второй индикатор	Третий индикатор	Четвертый индикатор	T0	T1	T2	T3	T4	SW START	SW STOP	N
1	HL1	HL2	HL3	HL4	1	2	3	4	0.5	SW1	SW2	3
2	HL2	HL3	HL4	HL1	2	3	1	0.5	3	SW2	SW3	2
3	HL3	HL4	HL1	HL2	3	2	0.5	1	4	SW3	SW4	5
4	HL4	HL3	HL1	HL2	4	3	2	0.5	1	SW5	SW6	3
5	HL2	HL4	HL3	HL1	1	3	2	4	2	SW7	SW8	5
6	HL3	HL1	HL2	HL4	2	4	3	0.5	3	SW8	SW9	3
7	HL1	HL3	HL4	HL2	3	2	0.5	1	4	SW9	SW10	4
8	HL2	HL1	HL3	HL4	2	3	1	0.5	3	SW10	SW11	5
9	HL2	HL4	HL1	HL2	1	4	2	3	0.5	SW11	SW12	3
10	HL1	HL2	HL4	HL3	0.5	4	3	2	1	SW12	SW13	5
11	HL1	HL3	HL4	HL2	2	3	3	2	4	SW13	SW14	5
12	HL2	HL4	HL3	HL1	2	0.5	1	1	3	SW1	SW14	4
13	HL3	HL2	HL1	HL4	2	3	1	0.5	3	SW2	SW13	5
14	HL4	HL3	HL2	HL1	0.5	4	3	2	1	SW3	SW12	4
15	HL2	HL4	HL3	HL1	4	1	1	3	2	SW4	SW11	3
16	HL2	HL1	HL3	HL4	2	2	0.5	3	1	SW5	SW10	4

9.3 Требования к выполнению отчета

Отчет по лабораторной работе выполняется в отдельной тетради и должен содержать:

- Название лабораторной работы, ее цель, задачи.
- Вариант задания.
- Таблица соответствия входов и выходов контроллера битам используемой памяти и функциональному назначению.

10 Лабораторная работа №3

Разработка алгоритма, создание и исследование программы «Управление работой циклового устройства на пневмоцилиндрах»

Цель работы — разработка цикловых управляющих программ на языке STEP-7, решающих задачи автоматизации на основе использования гидро- пневмоприводов.

10.1 Порядок выполнения лабораторной работы

ВАЖНО! К выполнению лабораторной работы на стенде «Пневмопривод» допускаются студенты прошедшие инструктаж по Технике Безопасности.

10.1.1. Изучить описание стенда «Пневмопривод », данное в настоящем документе и описание функциональных устройств, представленных в [6]. При изучении использовать стенд «Пневмопривод».

10.1.2. По описанию функциональных устройств изучить правила коммутации индикаторов и задающих кнопок, представленных на индикационном поле и на устройстве коммутации соответственно.

10.1.3. Подготовить стенд к работе, для чего:

- Включить персональный компьютер (ПК), соединенный интерфейсом PPI с программируемым логическим контроллером CPU-224.

- Загрузить пакет STEP-7 Micro/Win v 4.0.

- Включить вводной автомат С16 стенда, при этом на ПЛК загорится желтый индикатор (состояние ПЛК STOP).

10.1.4. Подготовить разработанную программу для загрузки в ПЛК, для чего:

- Ввести исходный текст программы, разработанной на основании задания.

- Откомпилировать программу, выбрав в меню пакета PLC и далее выбрать пункт **Compile**.

- В случае появления ошибок компиляции, устранить их и снова повторить предыдущий пункт.

- Проверить коммуникационное соединение персонального компьютера и ПЛК, для чего в меню PLC пакета выбрать пункт **Type**. После нажатия откроется диалоговое меню, где необходимо нажать **Read PLC**. При правильном соединении ПК и ПЛК в окошке **PLC Type** будет написано: CPU 224.

- Проверить состояние ПЛК, который должен при загрузке программы находиться в состоянии STOP.

- Выбрать в меню **File** пакета STEP-7 Micro/Win пункт **Download**. После нажатия откроется диалоговое меню **Download** где в пункте **Options** напротив требуемых для загрузки блоков (Program Block, Data Block, System Block) поставить галочки. Нажать кнопку **Download** и наблюдать процесс загрузки программы в ПЛК.

10.1.5. С помощью коммутационного поля установить требуемые соединения между индикаторами и выходами ПЛК и между задающими кнопками и входами ПЛК.

10.1.6. Перевести работу ПЛК в режим RUN, для этого в меню PLC пакета выбрать пункт **RUN**. После нажатия этой кнопки про-грамма, загруженная в ПЛК, начинает работать.

10.3 Задание для выполнения лабораторной работы

Разработать программу на языке STEP-7 для автоматизации работы манипулятора на пневмоцилиндрах по заданному алгоритму.

Дано:

Выходы контроллера соединены с пневмо-распределителями (ПР1..ПР4) пневмоцилиндров ПЦ1 и ПЦ2 (рис. 10.1).

- ПЦ1 — обеспечивает движение руки манипулятора,
- ПЦ2 — обеспечивает работу схвата манипулятора.

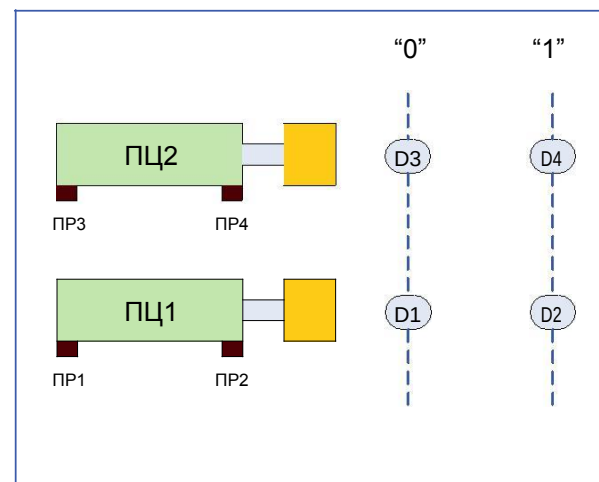


Рис. 10.1. Элементы управления пневмоцилиндров

Входы контроллера соединены с индукционными датчиками D1...D4.

- D1 — контролирует положение руки манипулятора «задвинута».
- D2 — контролирует положение руки манипулятора «выдвинута».
- D3 — контролирует состояние схвата манипулятора «зажат».
- D4 — контролирует состояние схвата манипулятора «разжат».

Необходимо:

Индицировать работу пневмоцилиндров следующим образом:

- Первый индикатор — состояние пневмораспределителя ПР1 пневмоцилиндра ПЦ1.
- Второй индикатор — состояние пневмораспределителя ПР2 пневмоцилиндра ПЦ1.
- Третий индикатор — состояние пневмораспределителя ПР3 пневмоцилиндра ПЦ2.
- Четвертый индикатор — состояние пневмораспределителя ПР4 пневмоцилиндра ПЦ2.

Обеспечить работу пневмоманипулятора по цикловому алгоритму.

1. Составить таблицу соответствия входов контроллера и датчиков, а также выходов и пневмораспределителей.
2. Запустить программу специальной кнопкой «Start».
3. Каждое состояние руки манипулятора и схвата должны индицироваться соответствующим индикатором.
4. Обеспечить цикловую работу манипулятора по заданному алгоритму. Алгоритм работы задан в табл. 10.1.

Таблица 10.1

Варианты заданий лабораторной работы №3

№ варианта	Первый индикатор.	Второй индикатор	Третий индикатор	Четвертый индикатор	Рука манипулятора	Схват манипулятора	SW START	SW STOP	N
1	HL1	HL2	HL3	HL4	выдвинута	зажат	SW1	SW2	3
					выдвинута	разжат			
					задвинута	зажат			
					задвинута	разжат			
2	HL2	HL3	HL1	HL4	задвинута	зажат	SW3	SW4	4
					задвинута	разжат			
					выдвинута	зажат			
					выдвинута	разжат			
3	HL1	HL2	HL4	HL3	выдвинута	разжат	SW5	SW6	2
					задвинута	зажат			
					выдвинута	зажат			
					задвинута	разжат			
4	HL1	HL3	HL2	HL4	задвинута	разжат	SW4	SW2	3
					выдвинута	зажат			
					задвинута	зажат			
					выдвинута	разжат			
5	HL2	HL1	HL3	HL4	выдвинута	зажат	SW1	SW4	4
					выдвинута	разжат			
					задвинута	зажат			
					задвинута	разжат			
6	HL1	HL4	HL3	HL3	задвинута	зажат	SW5	SW3	3
					задвинута	разжат			
					выдвинута	зажат			
					выдвинута	разжат			
7	HL3	HL4	HL2	HL1	выдвинута	разжат	SW6	SW2	3
					задвинута	зажат			
					выдвинута	зажат			
					задвинута	разжат			
8	HL3	HL1	HL2	HL4	задвинута	разжат	SW4	SW2	3
					выдвинута	зажат			
					задвинута	зажат			
					выдвинута	разжат			

10.4 Требования к выполнению отчета

Отчет по лабораторной работе выполняется в отдельной тетради и должен содержать:

- Название лабораторной работы, ее цель, задачи.
- Вариант задания.
- Таблица соответствия входов и выходов контроллера битам используемой памяти и функциональному назначению.
- Исходные тексты программы в редакторе *LAD* и в редакторе *STL* с комментариями.

10.5 Вопросы для самопроверки

- Основные функциональные узлы станда «Пневмопривод».
- Основные инструкции языка *STEP-7* (*MOVB*, *TON*, *ADD_DI*)
- Использование символьной адресации в пакете *STEP 7 Micro/WIN*.
- Основные способы доступа к данным *CPU-224*.
- Как откомпилировать и загрузить программу в ПЛК в пакете *STEP 7 Micro/WIN*.

11 Лабораторная работа №4

Разработка алгоритма, создание и исследование программы «Управление работой трехзвенного манипулятора по заданному алгоритму»

Цель работы — разработка цикловых управляющих программ на языке *STEP-7*, решающих задачи автоматизации вспомогательного оборудования гибких производственных модулей.

11.1 Порядок выполнения лабораторной работы

ВАЖНО! К выполнению лабораторной работы на стенде *CLC-224* допускаются студенты прошедшие инструктаж по Технике Безопасности.

11.1.1. Изучить описание станда *CLC-224*, данное в настоящем документе и описание функциональных устройств, представленных в [5]. При изучении использовать стенд *CLC-224*.

11.1.2. По описанию функциональных устройств изучить правила коммутации индикаторов и задающих кнопок, представленных на индикационном поле и на устройстве коммутации соответственно.

11.1.3. Подготовить стенд к работе, для чего:

- Включить персональный компьютер (ПК), соединенный интерфейсом *PPI* с программируемым логическим контроллером *CPU-224*.
- Загрузить пакет *STEP-7 Micro/Win v 4.0*.
- Включить вводной автомат *C16* станда, при этом на ПЛК загорится желтый индикатор (состояние ПЛК *STOP*).

11.1.4. Подготовить разработанную программу для загрузки в ПЛК, для чего:

- Ввести исходный текст программы, разработанной на основании задания.
- Откомпилировать программу, выбрав в меню пакета *PLC* и далее выбрать пункт **Compile**.
- В случае появления ошибок компиляции, устранить их и снова повторить предыдущий пункт.
- Проверить коммуникационное соединение персонального компьютера и ПЛК, для чего в меню *PLC* пакета выбрать пункт **Type**. После нажатия откроется диалоговое меню, где необходи-

мо нажать **Read PLC**. При правильном соединении ПК и ПЛК в окошке **PLC Type** будет написано: **CPU 224**.

- Проверить состояние ПЛК, который должен при загрузке программы находиться в состоянии **STOP**.
- Выбрать в меню **File** пакета **STEP-7 Micro/Win** пункт **Download**. После нажатия откроется диалоговое меню **Download** где в пункте **Options** напротив требуемых для загрузки блоков (**Program Block**, **Data Block**, **System Block**) поставить галочки. Нажать кнопку **Download** и наблюдать процесс загрузки программы в ПЛК.

11.1.5. С помощью коммутационного поля установить требуемые соединения между индикаторами и выходами ПЛК и между задающими кнопками и входами ПЛК.

11.1.6. Перевести работу ПЛК в режим **RUN**, для этого в меню **PLC** пакета выбрать пункт **RUN**. После нажатия этой кнопки про-грамма, загруженная в ПЛК, начинает работать.

11.2 Задание для выполнения лабораторной работы

Разработать программу на языке **STEP-7** для автоматизации работы трехзвенного манипулятора по заданному алгоритму.

Дано:

Механизм манипулятора соответствует цилиндрической системе координат. В этой системе звено 1 может вращаться относительно звена 0 (относительное угловое перемещение φ_{10}), звено 2 перемещается по вертикали относительно звена 1 (относительное линейное перемещение S_{21}) и звено 3 перемещается в горизонтальной плоскости относительно звена 2 (относительное линейное перемещение S_{32}). На звене 3 закреплено захватное устройство или схват, предназначенный для захвата и удержания объекта манипулирования при работе манипулятора. Функциональное назначение каждого исполнительного устройства показано на рис. 11.1.

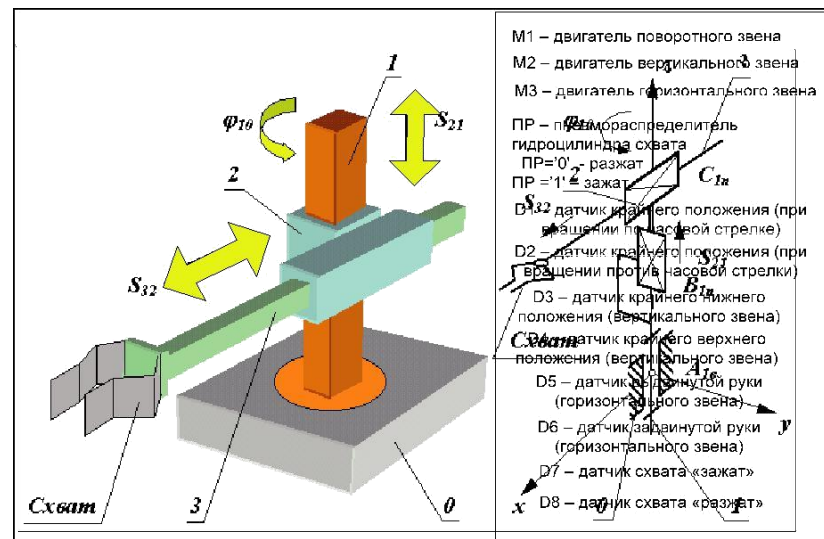


Рис. 11.1. Трехзвенный манипулятор

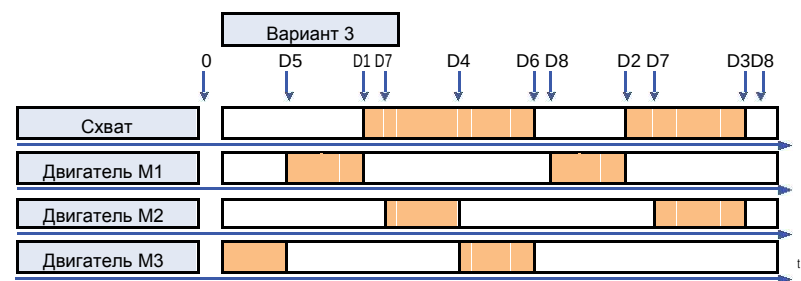
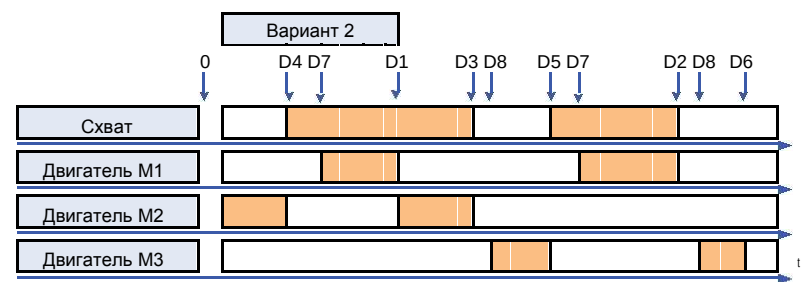
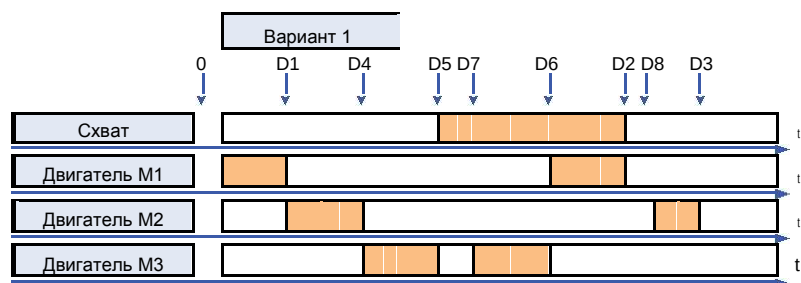
Выходы контроллера соединены:

- с пневмо-распределителем ПР пневмоцилиндра схвата;
- с двигателем M1 (**DIR1** — направление движения, **P1** — работа двигателя);
- с двигателем M2 (**DIR2** — направление движения, **P2** — работа двигателя);
- с двигателем M3 (**DIR3** — направление движения, **P3** — работа двигателя)

Входы контроллера соединены с датчиками **D1...D8**.

Необходимо:

1. Составить таблицу соответствия входов контроллера и датчиков, а также выходов контроллера и исполнительных устройств.
2. Запустить программу специальной кнопкой «Старт».
3. Установить все исполнительные устройства в исходное положение (схват разжат, рука задвинута, повернута против часовой стрелки и опущена вниз).
4. Обеспечить цикловую работу манипулятора по заданному алгоритму. Алгоритм работы представлен в виде циклограммы.



11.3 Требования к выполнению отчета

Отчет по лабораторной работе выполняется в отдельной тетради и должен содержать:

- Название лабораторной работы, ее цель, задачи.
- Вариант задания.
- Таблица соответствия входов и выходов контроллера битам используемой памяти и функциональному назначению.

- Исходные тексты программы в редакторе *LAD* и в редакторе *STL* с комментариями.

11.4 Вопросы для самопроверки

- Типы выходов в ПЛК семейства S7-200.
- Основные инструкции языка STEP-7 (*MOV_W*, *TOFF*, *SUB_DI*)
- Использование символьной адресации в пакете STEP 7 Micro/WIN.
- Основные составляющие цикла CPU семейства S7-200.
- Что такое PPI — интерфейс ПЛК.

Библиографический список

1. Ковальчук, Е. Р. Основы автоматизации машиностроительного производства: Учеб. для машиностроит. спец. вузов / Е. Р. Ковальчук, М. Г. Ко-сов, В. Г. Митрофанов и др.; под ред. Ю. М. Соломенцева. — 3-е изд., стер. — М.: Высш. Шк., 2001. — 312 с.: ил.
2. Петров, И. В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования / И. В. Петров, под ред. проф. В. П. Дьяконова. — М.: СОЛОН-Пресс, 2004 — 256 с.: ил. — (Серия «Библиотека инженера»).
3. www.siemens.com
4. Учебный стенд *CLC-221*. Паспорт и инструкция по эксплуатации.
5. Учебный стенд *CLC-224*. Паспорт и инструкция по эксплуатации.
6. Учебный стенд *CLA-117* «Пневмопривод». Паспорт и инструкция по эксплуатации.

Оглавление

Введение	3
1 Принципы построения ПЛК	5
2 Системы автоматизации <i>SIMATIC</i>	6
3 Обзор семейства ПЛК <i>S7-200</i> фирмы <i>SIEMENS</i>	14
4 Программное обеспечение <i>STEP-7</i> <i>Micro/Win 4.0</i>	16
5 Основы плк семейства <i>S7-200</i>	18
6 Лабораторные стенды на базе плк для выполнения лабораторных работ	29
7 Создание приложений для ПЛК Семейства <i>s7-200</i> в пакете <i>Step 7 Micro/WIN ver.4.0</i>	44
8 Лабораторная работа №1	48
9 Лабораторная работа №2	52
10 Лабораторная работа №3	56
11 Лабораторная работа №4	61
Библиографический список	66

Электронное учебное издание

Коковин Валерий Аркадьевич

**ЛАБОРАТОРНЫЕ РАБОТЫ ПО ДИСЦИПЛИНЕ
«АВТОМАТИЗАЦИЯ ТЕХНОЛОГИЧЕСКИХ
ПРОЦЕССОВ И ПРОИЗВОДСТВ»**

ЭЛЕКТРОННОЕ МЕТОДИЧЕСКОЕ ПОСОБИЕ

Филиал «Протвино»
государственного университета «Дубна»
142281 г. Протвино Московской обл.,
Северный проезд, д. 9

