Министерство образования и науки Российской Федерации

Федеральное агентство по образованию

Саратовский государственный технический университет

**ПРОГРАММИРОВАНИЕ ПРОМЫШЛЕННЫХ**

**КОНТРОЛЛЕРОВ SIMATIC S7-200**

Методические указания к лабораторной работе

по курсам «Системы реального времени», «Проектирование АСОИУ»

для студентов специальности 230102

дневной формы обучения

*Одобрено*

*редакционно-издательским советом*

*Саратовского государственного*

*технического университета*

**Саратов 2006**

**Введение**

В методических указаниях рассматривается методика разработки программ управления промышленными контроллерами Siemens SIMATIC S7-200 в системе программирования STEP 7– Micro/WIN.

Методические указания содержат описание контроллера Siemens SIMATIC S7 - 200, основные приемы программирования в системе STEP 7 – Micro/WIN, пример решения учебной задачи, список вопросов для самопроверки, требования к оформлению отчета о лабораторной работе и список литературы, необходимой для выполнения данной лабораторной работы.

Методические указания предназначены для студентов специальности 230102 «Автоматизированные системы обработки информации и управления», а также могут быть использованы студентами других специальностей, связанных с решением задач автоматизации управления технологическими процессами в реальном времени.

***Цель работы: освоение системы программирования STEP 7– Micro/WIN и разработка с ее помощью программ управления промышленными контроллерами для решения задач автоматизации управления технологическими процессами в реальном времени.***

1. **общие сведения о контроллерах**

**Siemens SIMATIC S7-200**

Серия S7–200 – это ряд микропрограммируемых промышленных логических контроллеров (ПЛК, PLC), которые могут управлять разнообразными прикладными системами автоматизации. Компактная конструкция, расширяемость, низкая стоимость и мощная система команд микроконтроллеров S7–200 создают идеальное решение для управления малыми приложениями. Промышленные контроллеры этой серии собираются по модульному принципу. Контроллер в общем случае состоит из одного центрального единственного процессора (CPU); нескольких (0-7) моделей аналогового и дискретного ввода-вывода, а также коммуникационных модулей (Profibus, Ethetnet, AS-интерфейс) и модулей позиционирования.

Семейство S7–200 включает 4 варианта CPU (221, 222, 224, 226) различающиеся объемом памяти, числом дискретных входов/выходов и подключаемых модулей. Это многообразие обеспечивает множество технических характеристик и обеспечивает рентабельность решения задач автоматизации. Процессорные модули содержат один или два последовательных коммуникационных порта. Порты поддерживают три варианта протоколов: точка–точка (PPI), многоточечный (MPI) и свободно программируемый режим (например, протокол Modbus).

При изучении работы с контроллером будет использоваться учебный стенд, изображённый на рис.1 и имеющий следующую конфигурацию: персональный компьютер, CPU S7–222, один модуль аналогового ввода/вывода EM235, кабель связи PC/PPI, персональный компьютер с установленной системой программирования STEP 7– Micro/WIN. CPU имеет три режима работы: STOP, RUN, TERM. Для программирования контроллера используется режим TERM.

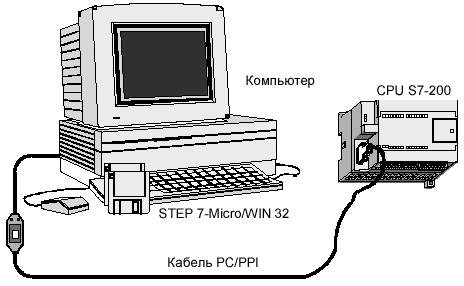


Рис.1. Компоненты системы с микроконтроллером S7–200

Для отладки программ и проверки их функционирования может использоваться программный эмулятор контроллера S7-200 (рис.2) (http://inicia.es/de/canalPLC). Для использования эмулятора необходимо, находясь в системе программирования STEP 7– Micro/WIN, сохранить на диске программу в формате AWL (экспортировать проект). Затем загрузить эту программу с диска в эмулятор контроллера S7-200 и выполнить проверку ее функционирования.

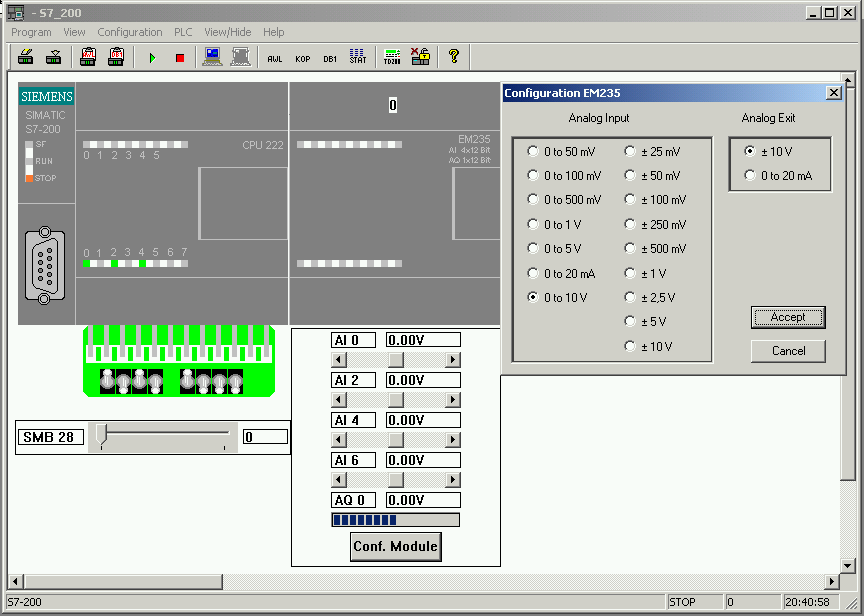


Рис.2. Интерфейс программного эмулятора контроллера S7-200

**2. Структура программы и цикл сканирования**

При управлении задачей или процессом CPU S7-200 непрерывно выполняет программу. Программа для CPU S7-200 создается при помощи системы программирования STEP 7 - Micro/WIN и состоит из трех основных элементов: главная программа, программы обработки прерываний, подпрограммы.

Главная программа – это основная часть программы, где размещаются команды, управляющие вашим приложением. Команды главной программы выполняются последовательно и однократно в каждом цикле сканирования CPU. Из главной программы можно вызывать различные подпрограммы и программы обработки прерываний.

Программы обработки прерываний – эти необязательные элементы программы выполняются при каждом возникновении события, вызывающего прерывание.

Подпрограммы – эти необязательные элементы программы выполняются только тогда, когда они вызываются из главной программы или программы обработки прерываний.

CPU S7-200 предназначен для циклического выполнения ряда заданий, включая программу пользователя. Такое циклическое выполнение заданий называется циклом сканирования (рис.3). В течение цикла сканирования CPU выполняет все или большинство из следующих задач: 1) считывание состояния входов; 2) выполнение программы; 3) обработка любых коммуникационных запросов; 4) выполнение самодиагностики CPU; 5) запись результатов в выходы.

РАЕ –Таблица отображения входов

РАА – Таблица отображения выходов

STEP 7 – программа.

* Маркеры
* Таймеры
* Счетчики
* **…**

0

1

n

2

**…**

0

1

n

2

**…**

Входы

Выходы

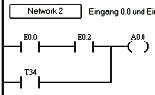


Рис.3. Выполнение цикла сканирования

CPU S7-200 позволяет принудительно устанавливать значения дискретных входов и выходов, а также до 16 значений внутренней памяти или аналоговых входов и выходов. Этот прием широко используется при отладке программ (подробнее смотри системное руководство C79000 в разделе 4 «Основные концепции программирования CPU S7-200»).

##### 3. Языки программирования и редакторы SIMATIC S7-200

В CPU S7-200 доступны две системы команд: IEС 1131–3 и SIMATIC. Система команд SIMATIC предназначена для CPU S7-200. Команды SIMATIC имеют самые короткие времена выполнения. Система команд IEC 1131–3 ограничивается командами, которые соответствуют международному стандарту МЭК 1131. Стандартные команды IEC 1131-3 описаны в системном руководстве C79000-G7076-C233-02 в разделе 10 «Инструкции контроллера S7-200». Эти команды объединяются в группы:

1. поразрядные логические команды;
2. команды сравнения IEC;
3. команды таймеров;
4. команды счетчиков;
5. арифметические команды;
6. числовые функции;
7. команды пересылки;
8. логические команды;
9. команды сдвига и циклического сдвига;
10. команды преобразования.

При использовании команд IEC 1131–3 параметры команды автоматически проверяются на правильность формата данных.

Система программирования STEP 7-Micro/WIN предоставляет выбор из трех различных редакторов для создания управляющих программы: Ladder Logic, Function Block Diagram или Statement List.

Редактор контактного плана (*Ladder Logic*)

Редактор контактного плана (LAD) STEP 7-Micro/WIN позволяет формировать программы, имеющие сходство с электрической монтажной схемой (рис.4). Программы в формате LAD позволяют CPU эмулировать протекание электрического тока от источника питания через ряд логических состояний входов, которые, в свою очередь, разблокируют логические состояния выходов. Логика обычно подразделяется на малые легко понимаемые сегменты, которые в англоязычной литературе часто называются rungs, т.е. ступеньки. Программа выполняется слева направо и затем сверху вниз, как предписано программой. Как только CPU достигает конца программы, он начинает снова с вершины программы.

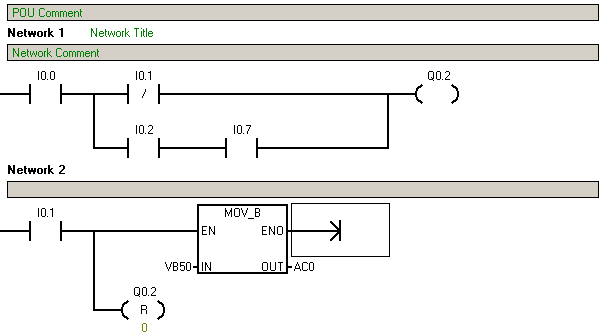


Рис.4. Пример программы в формате LAD

Команды представляются графическими символами (рис.9) и относятся к трем основным типа:

1. контакты представляют логические состояния, являются аналогами выключателей, кнопок;
2. катушки обычно представляют логические результаты, аналогичные лампам, пускателям электродвигателей, промежуточным реле, состояниям внутренних выходов и т.д.;
3. блоки представляют дополнительные команды, такие как таймеры, счетчики или математические команды.

Достоинства редактора LAD:

1. редактор LAD прост в использовании для начинающих программистов;
2. графическое представление легче понимается и поэтому является популярным;
3. редактор LAD можно использовать и с системой команд SIMATIC и IEC 1131–3;
4. для отображения программы, созданной при помощи редактора LAD, всегда можно использовать редактор STL.

Редактор функционального плана

(*Function Block Diagram)*

Редактор функционального плана (FBD) STEP 7-Micro/WIN позволяет программировать в форме функциональных (логических) блоков (рис.5), напоминающих элементы блок-схем. Нет никаких контактов и катушек, как в редакторе LAD, но имеются эквивалентные команды, которые появляются как команды в форме блоков. Логика программы выводится из соединений между этими командами в форме блоков. То есть выход одной команды (например, блока AND) может использоваться для того, чтобы разрешить другой команде (например, таймеру) создать необходимую логику управления. Такая концепция соединений позволяет решать широкий круг логических задач.

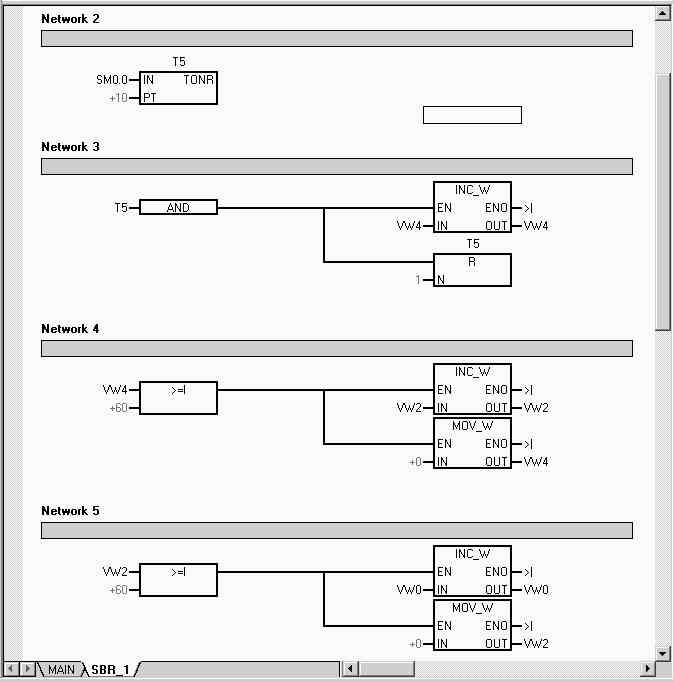
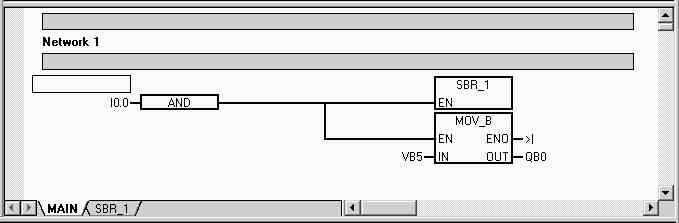


Рис.5. Пример программы в формате FBD

Достоинства редактора FBD:

1. графический стиль представления в форме логических элементов хорош для последующего составления блок – программы;
2. редактор FBD можно использовать и с системой команд SIMATIC и IEC 1131–3;
3. для отображения программы, созданной при помощи редактора FBD, всегда можно использовать редактор STL.

На рис.5 показан пример программы, созданной при помощи редактора FBD. Данная программа определяет длительность работы привода (тормоза, автоматы и т.д.). Она основана на том, что при работающем приводе на вход «I0.0» подается сигнал 24 В, а при остановленном двигателе напряжение отсутствует. Появление сигнала запускает счет времени, а при отсутствии сигнала счет времени прерывается, пока сигнал не появится снова. Число сосчитанных часов заносится в слово данных «VW0», минуты – в слово данных «VW2», а секунды – в слово данных «VW4». (Пример находится в файле «S7-200\Tips&Trics\S72\_16r.pdf»).

Цикл программы начинается с опроса состояния входа «I0.0». Если логическое состояние «1», то вызывается «Подпрограмма 1». В «Подпрограмме 1» таймер «T5» (задержка включения с запоминанием) устанавливается на одну секунду. По прошествии этой секунды устанавливается бит таймера «T5», а значение слова меркеров «MW4» для значения секунд увеличивается на 1. Для того чтобы при следующем цикле таймер снова мог быть запущен, бит таймера «T5» сбрасывается. Если меркер секунд достигает значения 60, то меркер минут «MW2» повышается на 1, а меркер секунд «MW4» снова сбрасывается в ноль (знак «>» служит здесь для повышения надежности). Если меркер минут достигает значения 60, то он сбрасывается в ноль, а слово данных «VW0», работающее как память для прошедших часов, увеличивается на 1. Кроме того, меркер минут «MW2» копируется в слово данных «VW2» для текущего значения минут, а меркер секунд «MW4» - в слово данных «VW4» для текущего значения секунд. На этом работа подпрограммы завершается. Последняя инструкция в конце основной программы служит только для вывода текущего значения секунд на светодиодный индикатор.

Редактор списка команд (*Statement List*)

Редактор списка команд (STL) STEP 7-Micro/WIN позволяет создавать управляющие программы, вводя мнемонику команд. Редактор STL больше подходит для опытных программистов, хорошо знакомых с ПЛК и логическим программированием. Редактор STL позволяет также создавать программы, которые нельзя создать с помощью редакторов LAD и FBD, так как STL - это «родной язык» CPU, а не язык для графического редактора, где применяются ограничения для правильного рисования схем.

На рис.6 показан пример программы в форме STL. Данная программа реализует пример шаговой цепочки, в каждом шаге выполняется определенное действие. Шаги выполняются последовательно, причем шаг выполняется только, если выполнены все предусмотренные для этого условия. Условия и активные выходы для каждого шага приведены в таблице.

**Network 1 // шаг 1**

LD I0.1 // Условие пуска

AN I0.0 // и нет сброса

AN M0.0 // и не установлены след. шаги

AN M0.1 //

AN M0.2 //

AN M0.3 //

AN M0.4 //

S M0.0, 1 // 1-й шаг установить

**Network 2**

LD M0.0 // 1-й шаг

S Q0.2, 2 // установить выходы

TON T37, +50// Интервал времени 2-го шага

**Network 3 // шаг 2**

LD T37 //По оконч. 1-го интервала врем.

A M0.0 // и после выполнения 1-го шага R R M0.0, 1 // 1-й шаг сбросить,

S M0.1, 1 // а 2-ой шаг установить

**Network 4**

LD M0.1 // 2-ой шаг

S Q0.1, 1 // Установить выходы

S Q0.4, 1 //

R Q0.2, 2 // Сбросить выходы

TON T38,+50// Интервал времени 3-го шага

**Network 5 // шаг 3**

LD T38 //По оконч. 2-го интервала врем.

A M0.1 // и после выполнения 2-го шага

R M0.1, 1 // 2-ой шаг сбросить,

S M0.2, 1 // а 3-й шаг установить

**Network 6**

LD M0.2 // 3-й шаг

S Q0.0, 1 // Установить выходы

S Q0.5, 1

R Q0.1, 1 // Сбросить выходы

R Q0.4, 1

**Network 7 // шаг 4**

LD I0.2 // Появилось усл.вкл. 4-го шага

A M0.2 // и после выполнения 3-го шага

R M0.2, 1 // 3-й шаг сбросить,

S M0.3, 1 // а 4-й шаг установить

**Network 8**

LD M0.3 // 4-й шаг

S Q0.1, 1 // Установить выходы

S Q0.3, 1 //

R Q0.0, 1 // Сбросить выходы

TON T39, +50// Интервал времени 5-го шага

**Network 9 // шаг 5**

LD I0.3 //Появилось усл.вкл. 5-го шага

A T39

A M0.3 // и после выполнения 4-го шага

R M0.3, 1 // 4-й шаг сбросить,

S M0.4, 1 // а 5-й шаг установить

**Network 10**

LD M0.4 // 5-й шаг

R Q0.1, 1 // Сброс выходов

R Q0.5, 1

**Network 11**

LD I0.0 // Если кнопка сброса нажата,

R M0.0, 5 // шаговые маркеры сбросить,

R Q0.0, 6 // все 6 выходов сбросить

Рис.6. Пример программы в форме STL

Таблица

Условия и активные выходы для каждого шага

|  |  |  |
| --- | --- | --- |
| Шаг | Предварительные условия | Активные выходы |
| 1 | I0.1 установлен | Q0.2 - Q0.3 |
| 2 | Выдержка 5 сек (таймер T37) | Q0.1 - Q0.4 |
| 3 | Выдержка 5 сек (таймер T38) | Q0.0 - Q0.5 |
| 4 | I0.2 установлен | Q0.1 - Q0.3 - Q0.5 |
| 5 | Выдержка 5 сек (таймер T39) & I0.3 установлен | Q0.3 |
|  | Сброс шаговой цепочки (I0.0 установлен) | нет |

Шаговая цепочка состоит из пяти шагов, выполняемых последовательно. Один шаг состоит из установки и сброса определенных выходов. Для того чтобы шаг мог быть сделан, должны быть выполнены определенные предварительные условия, например, нажатие выключателя или окончание времени ожидания. Кроме того, нажатие выключателя «I0.0» позволяет сбросить шаговую цепочку в любой момент времени. (Пример находится в файле «S7-200\Tips&Trics\S72\_18r.pdf»)

**4. Типы данных и способы адресации** **SIMATIC S7-200**

CPU S7–200 предоставляет специализированные области памяти для обеспечения более быстрой и эффективной обработки данных, при этом обеспечивается прямая и косвенная адресация областей памяти CPU.

CPU S7–200 хранит информацию в различных ячейках памяти, имеющих уникальные адреса. При явном указании адреса в памяти, к которому нужно обратиться, обеспечивается прямая адресация. Это позволяет программе иметь прямой доступ к информации.

Для обращения к биту в области памяти указывается адрес, который включает в себя идентификатор области памяти, адрес байта и номер бита. На рис.7 показан пример обращения к биту (адресация «байт.бит»). В этом примере за областью памяти и адресом байта («I» – вход (input), а «3» – байт номер 3) следует точка («**.**»), чтобы отделить адрес бита (бит 4).



Рис. 7. Обращение к биту данных в памяти CPU (адресация «байт.бит»)

Используя байтовый формат адреса, можно обращаться к данным в различных областях памяти CPU (V, I, Q, M, S, L и SM) как к байтам, словам или двойным словам. Для доступа к байту, слову или двойному слову в памяти CPU вы должны указать адрес таким же способом, как и при указании адреса бита. Адрес включает идентификатор области, обозначение размера данных и начальный байтовый адрес байта, слова или двойного слова (рис. 8). Доступ к данным в других областях памяти CPU (таким, как T, C, HC и аккумуляторы) производится с помощью формата адреса, включающего идентификатор области и номер элемента.

Контроллер может адресоваться к следующим областям памяти:

1. регистрам входов образа процесса («I»);
2. регистрам выходов образа процесса («Q»);
3. области памяти переменных («V»);
4. битовой памяти (меркеров) («M»);
5. области памяти реле управления последовательностью операций («S»);
6. битов специальной памяти («SM»);
7. области локальной памяти («L»);
8. счетчиков («C»);
9. аналоговых входов («AI»);
10. аналоговых выходов («AQ»);
11. аккумуляторам («AC»);
12. скоростных счетчиков («HC»).

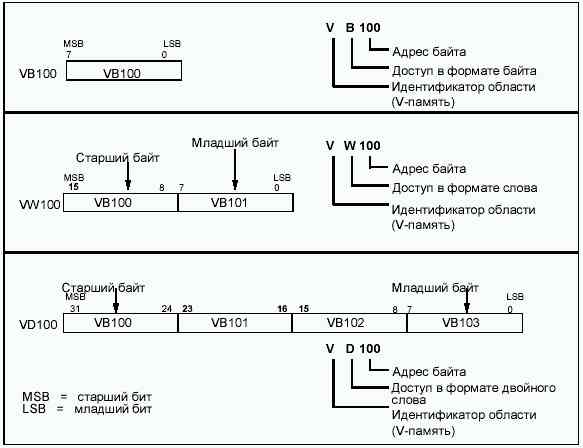
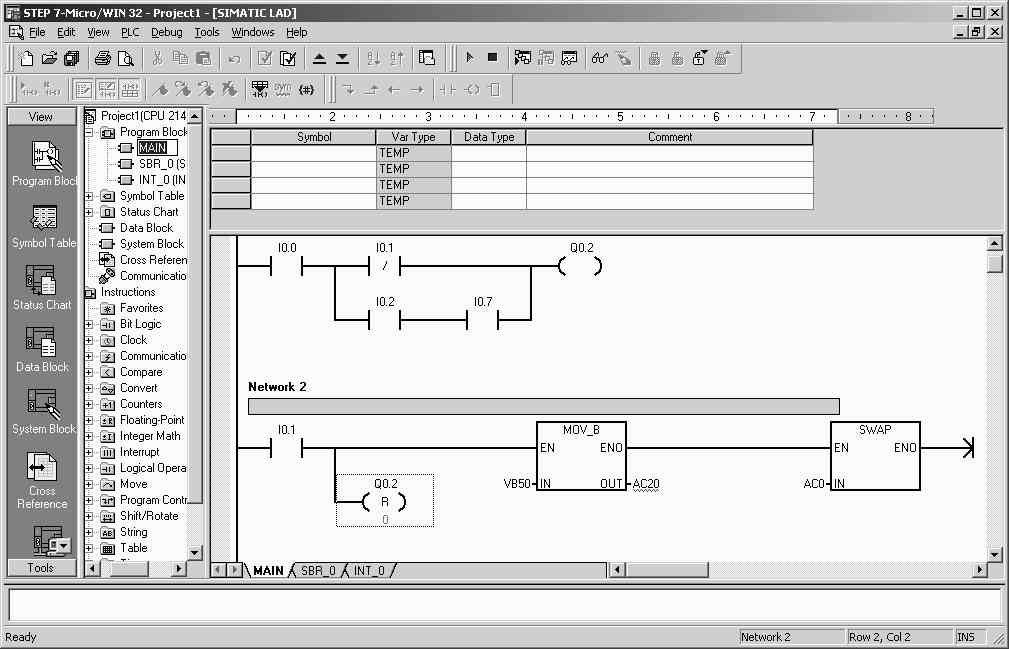


Рис. 8. Сравнение доступа к DB, DW и DD по одному и тому же адресу

Подробно типы данных и режимы адресации описаны в системном руководстве «C79000-G7076-C233-02» раздел 5 «Память CPU: Типы данных и способы адресации».

**5. Пример разработки программы в Step7 - Micro/WIN**

1. Подключить контроллер к последовательному порту компьютера (рис.1).
2. Установить переключатель режима работы контроллера в положение TERM.
3. Загрузить Step7 - Micro/WIN.
4. Создать новый проект в формате LAD.
5. Написать программу в формате LAD (рис. 4).
6. Сохранить программу на диске.
7. Загрузить программу в контроллер (рис.9).
8. Создать новый проект.
9. Загрузить программу из контроллера в компьютер (рис.9).
10. Запустить программу на выполнение (рис.9).
11. Остановить программу (рис.9).
12. Установить режим отладки (рис.9).
13. Запустить и выполнить программу в режиме отладки (рис.9).
14. Управлять контроллером с помощью переключателей 1-8 и объяснить результаты выполнения программы.



Параллельная цепочка: (Логика «ИЛИ»). Первый ключ ИЛИ второй ключ должны быть замкнуты для протекания тока.

Катушка: Если значение - «Истина» и ток проходит через катушку, то она активируется. Катушка запускается

Параллельная цепочка: (Логика «И»). Первый ключ И второй ключ должны быть замкнуты для протекания тока.

Опрос: Ток течет? Если да, то результат опроса - истина. Результат «1».

Опрос: Ток не течет? Если да (нет тока), то результат опроса - истина. Результат «0»

Контакты представляют логические состояния объектов, аналогичных выключателям, кнопкам, внутренним условиям

Катушки представляют логические результаты, аналогичные объектам: лампа, пускатель эл. двигателя, промежуточное реле, состояние внутренних выходов

Блоки представляют дополнительные команды, такие как таймеры, счетчики или математические команды

Загрузить программу в контроллер

Считать программу из контроллера

Запустить программу на выполнение

Остановить выполнение программы

Включить режим отладки

Рис.9. Общий вид системы программирования Step7 - Micro/WIN

1. Остановить выполнение программы
2. Преобразовать программу из формы LAD в FBD и STL.
3. Написать программу в формате FBD (рис. 5) и повторить пункты 5-15.
4. Написать программу в формате STL (рис. 6) и повторить пункты 5-14.

**Задания на разработку программ в Step7 – Micro/WIN**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Название примера** | **Сложность** | **Имя файла примера** |
| 1 | Обработка фронтов | 1 | Tips&Trics/S72\_**09**.pdf |
| 2 | Прерывания по времени | 1 | Tips&Trics/S72\_**01**.pdf |
| 3 | Прерывания по событию | 1 | Tips&Trics/S72\_**02**.pdf |
| 4 | Аналоговые потенциометры | 1 | Tips&Trics/S72\_**03**.pdf |
| 5 | Выключение по времени | 1 | Tips&Trics/S72\_**17**.pdf |
| 6 | Счетчик часов работы | 1 | Tips&Trics/S72\_**16**.pdf |
| 7 | Таймеры | 1 | Tips&Trics/S72\_**31**.pdf |
| 8 | Часы реального времени | 1 | Tips&Trics/S72\_**08**.pdf |
| 9 | Шаговая цепочка | 2 | Tips&Trics/S72\_**18**.pdf |
| 10 | Реверсивное управление двигателем | 2 | 1Hour\_primer\1step\_r.pdf |
| 11 | Цикл For-Next | 2 | Tips&Trics/S72\_**06**.pdf |
| 12 | Широтно-импульсная модуляция | 2 | Tips&Trics/S72\_**07**.pdf |
| 13 | Регулирование лампы (импульсн. выход) | 2 | Tips&Trics/S72\_**21**.pdf |
| 14 | ПИД ‑ регулятор | 2 | Tips&Trics/S72\_**32**.pdf |
| 15 | Аналоговый модуль | 3 | Tips&Trics/S72\_**34**.pdf |
| 16 | Windows интерфейс через DDE | 4 | Tips&Trics/S72\_**35**.pdf |
| 17 | Передача данных (S7-200‑PCAccess‑‑ TraceMode) | 4 |  |
| 18 | Операторский интерфейс TD 200 | 4 | Tips&Trics/S72\_**51**.pdf |
| 19 | Масштабирование | 5 | Tips&Trics/S72\_**38**.pdf |
| 20 | РТ 100 Линеаризация (EM235) | 5 | Tips&Trics/S72\_**39**.pdf |
| 21 | PID- команда | 5 | Tips&Trics/S72\_**53**.pdf |
| 22 | Демонстрация PID | 5 | Tips&Trics/S72\_**54**.pdf |

##### Содержание работы

## Ознакомиться с контроллером SIMATIC S7-200, его типами данных, способами адресации, циклом сканирования и структурой программы.

1. Изучить языки программирования для SIMATIC S7-200 и работу в системе программирования Step7 – Micro/WIN.
2. Реализовать примеры разработки программ в Step7 - Micro/WIN (пункт 5).
3. Получить у преподавателя задание на самостоятельную разработку программ в Step7 - Micro/WIN.
4. Подготовить отчет о работе.

##### Требования к оформлению отчета

1. Титульный лист.
2. Цель работы и задание на выполнение работы.
3. Контрольные вопросы и краткие ответы на них.
4. Алгоритмы выполнения программ.
5. Листинги программ.
6. Выводы по работе.

##### Вопросы для самопроверки

1. Номенклатура CPU и модулей контроллеров серии S7-200. Протоколы обмена информацией по последовательному интерфейсу.
2. Назначение и возможности системы STEP 7– Micro/WIN.
3. Особенности языков программирования Ladder Logic, Function Block Diagram и Statement List.
4. Типы данных и способы адресации SIMATIC S7-200.
5. Системы команд SIMATIC и команды IEC 1131-3.
6. Отладка и выполнение программ в системе STEP 7– Micro/WIN.

##### Список литературы

1. Деменков Н.П. Языки программирования промышленных контроллеров: учеб. пособие / Н.П. Деменков; под ред. К.А.Пупкова. М.: Изд-во МГТУ им. Н.Э.Баумана, 2004. 172 с.
2. Системное руководство Siemens SIMATIC S7-200 C79000-G7076-C233. М.: Фирма Siemens, 2003. 909 с.
3. Tips&Trics. SIMATIC S7-200 by Siemens. Версия 3.0 Вып. 10/1996.
4. Автоматизация и приводы, СА01 каталог продукции Siemens //СТА. 2006. №1.

ПРОГРАММИРОВАНИЕ ПРОМЫШЛЕННЫХ

КОНТРОЛЛЕРОВ SIMATIC S7-200

Методические указания к лабораторной работе

по курсам «Системы реального времени», «Проектирование АСОИУ»

Составили: ПЕТРОВ Дмитрий Юрьевич

ЕГОРОВ Станислав Владиславович

Рецензент В.А. Иващенко

Редактор О.А. Луконина

Лицензия ИД № 06268 от 14.11.01

Подписано в печать Формат 60х84 1/16

|  |  |  |
| --- | --- | --- |
| Бум. тип. | Усл. печ. л. 0,93 (1,0) | Уч.- изд. л. 0,9 |
| Тираж 100 экз. | Заказ | Бесплатно |

Саратовский государственный технический университет

410054 г. Саратов, ул. Политехническая, 77