

Введение

Дорогой пользователь S7-200,

использование программируемых контроллеров (PLC) в автоматизации производства постоянно растет, так как это позволяет уменьшить стоимость продукции при увеличении ее объемов. Уверенный темп технического развития является другой причиной того, почему все больше и больше проблем автоматизации решаются с помощью программируемых контроллеров.

Микроконтроллер PLC S7-200, в особенности, применяется все больше и больше, потому что он сочетает мощность с привлекательной ценой и простым управлением.

Чтобы Вам делать первые шаги в мире S7-200 было просто и доступно, мы создали специальный стартовый набор.

Этот одночасовой пример предназначен для того, чтобы помочь Вам научиться основным приемам, чтобы в состоянии потом применить S7-200 в пределах минимально возможного времени.

А теперь, мы желаем Вам простого и быстрого старта и всяческих успехов!



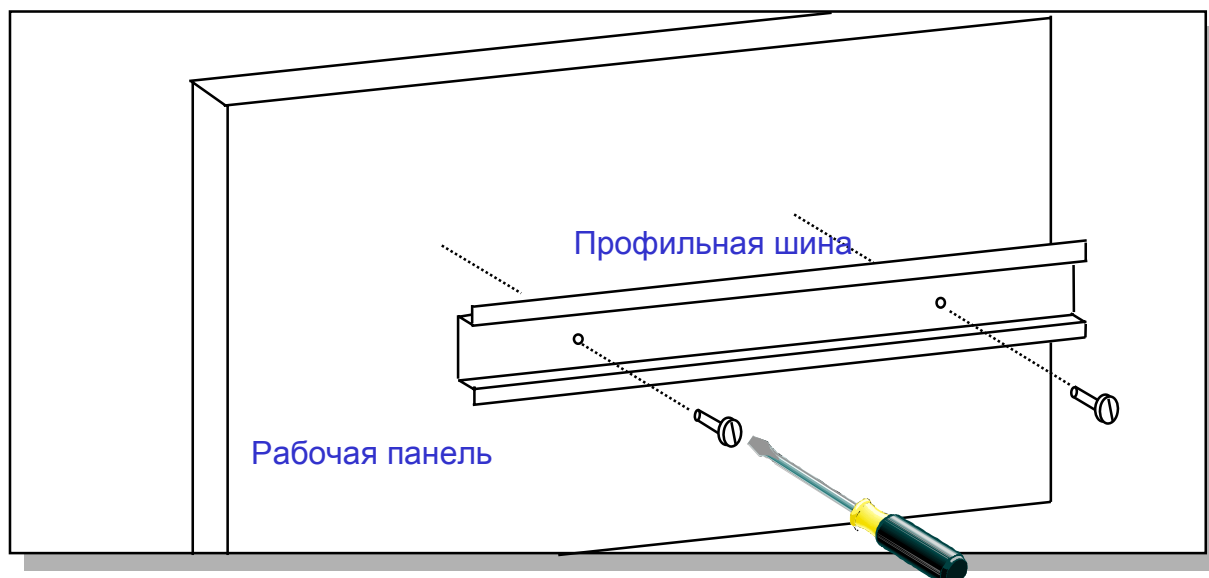
Содержание

Оборудование	Установка оборудования (Монтаж) Органы управления S7-200 (CPU 212) 6 Подключение устройства Схема учебного устройства Схема подключения S7-200 (CPU 212)	5 7 8 9
Программное обеспечение	Инсталляция программы под Windows 95 Инсталляция программы под Windows 3.1x Запуск STEP 7-Micro/WIN Система помощи	11 12 13 14
Основные установки	Установка скорости передачи Конфигурирование порта Первый тест на работоспособность	15 16 17
1-й пример программы	Записываем свою первую программу Чтение 1-го программного примера Загрузка программы в контроллер Тестирование программы Логические элементы Преобразование электрической схемы Элементы программы 1 Наблюдение статуса программы (online) Структура элемента программы	19 20 21 22 23 24 25 26 27
Модернизации примера	1. Операция "И" Вставка логического элемента Вставка операнда и тестирование Удаление ... 2. Операция "ИЛИ" Вставка элемента "ИЛИ" 3. Таймер задержки Функция таймера задержки Программирование таймера задержки Разговариваем просто Создание нового проекта Поздравления	29 30 31 32 33 34 35 36 37 38 40 43
Приложение	О битах, байтах и словах Адресные области в S7-200 Выполнение циклической программы	48 49 50

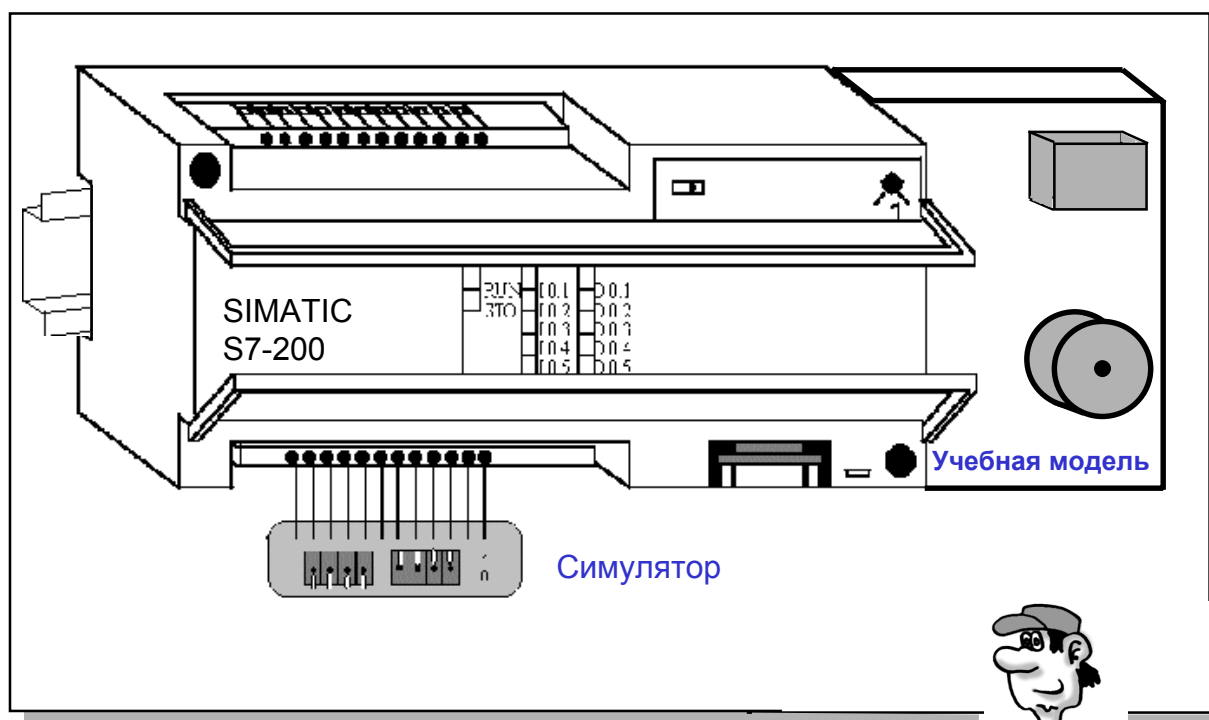


Структура документа показывается внизу на каждой странице. Тот раздел, который Вы открыли, выделен голубой рамкой с тенью.

Установка оборудования (Монтаж)



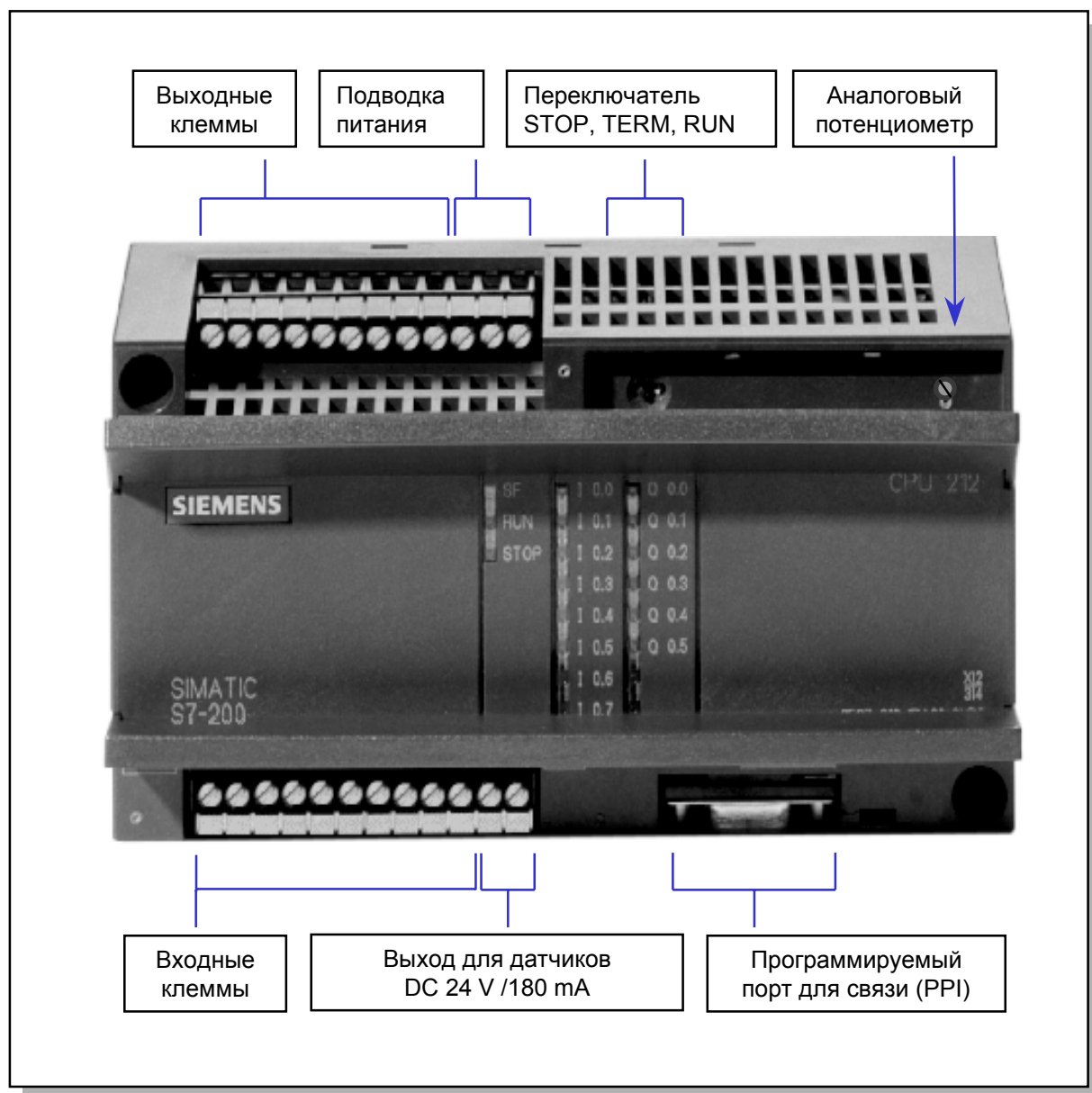
1. Установите профильную шину на рабочую панель, как указано выше.



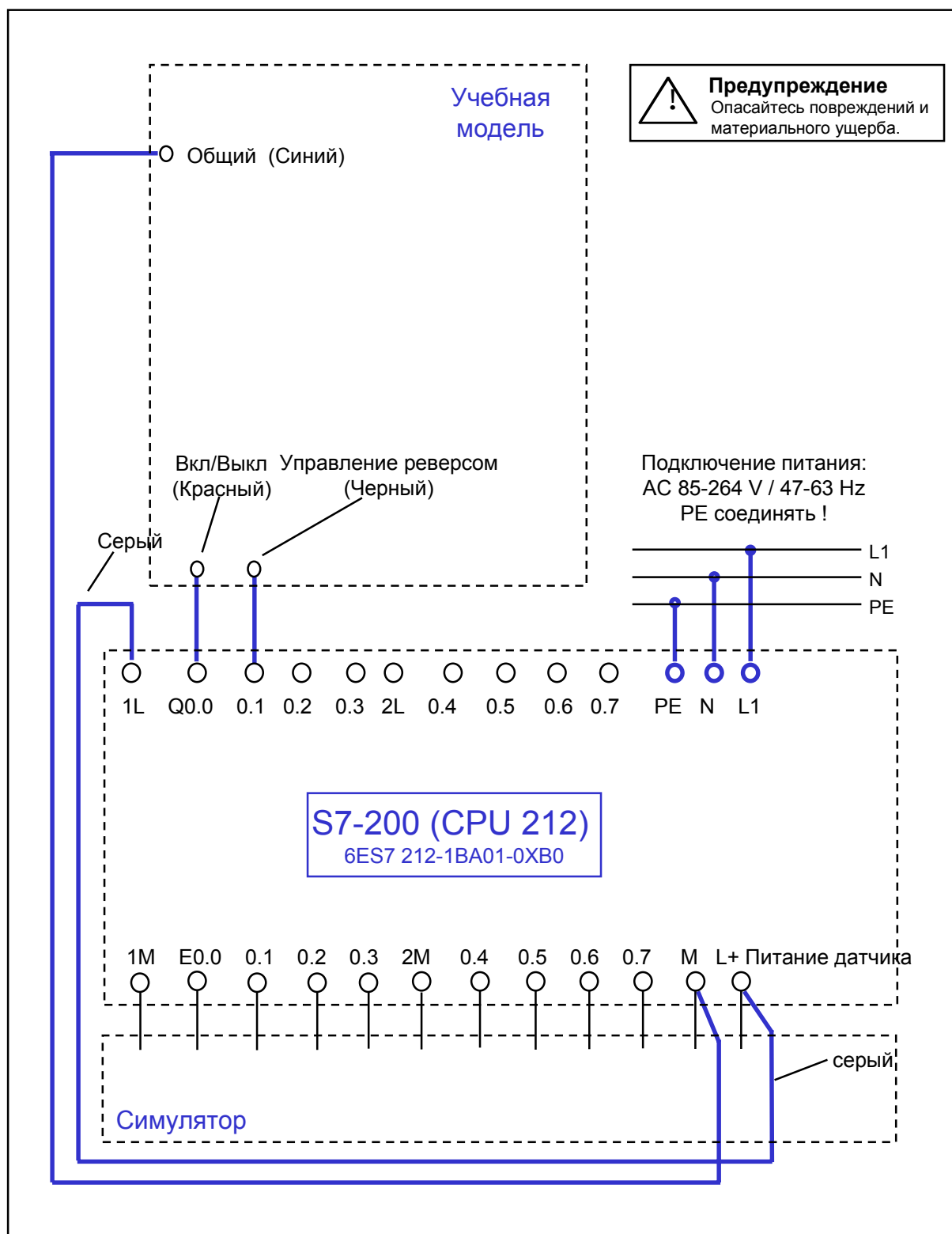
2. Подключите симулятор к входным клеммам S7-200, которые расположены под нижней крышкой на передней стороне контроллера.
3. Установите контроллер на профильную шину.
4. Установите учебную модель рядом с контроллером.



Органы управления S7-200 (CPU 212)



Подключение устройства



5. Соедините кабели, выделенные жирным шрифтом, как показано на диаграмме выше. Серый провод только закреплен на учебной модели механически. Концы его должны быть подключены к клеммам L+ и 1L.

Схема учебного устройства

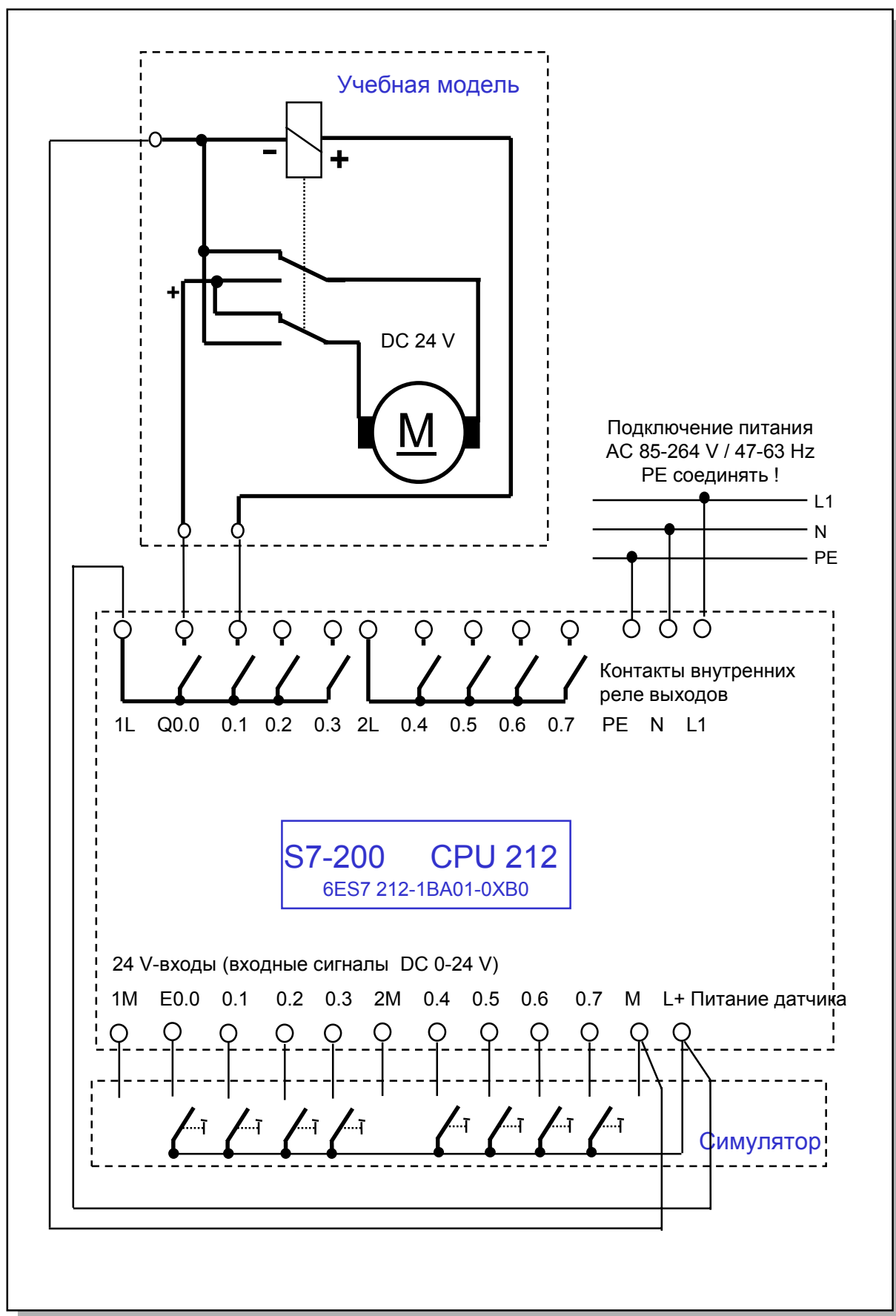
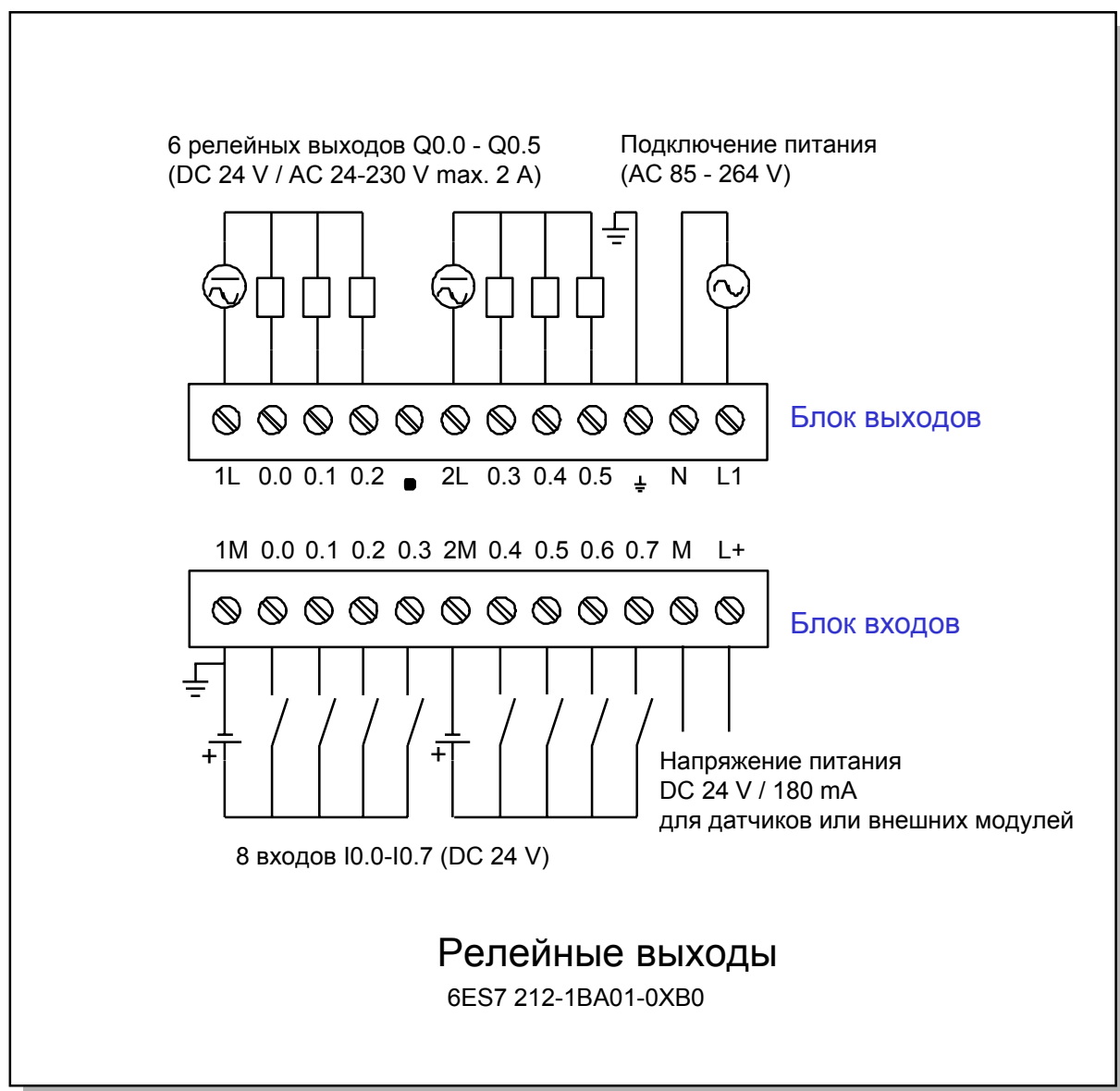
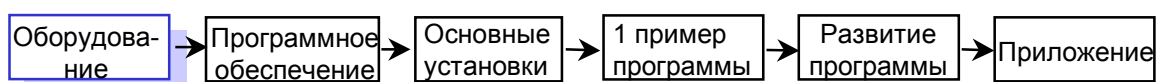


Схема подключения S7-200 (CPU 212)





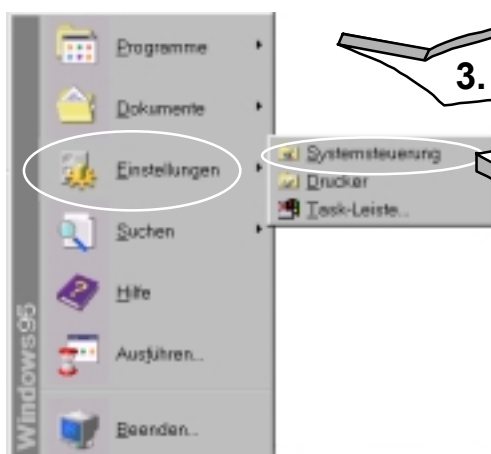
Инсталляция программного обеспечения под Windows 95

Для инсталляции программного обеспечения STEP7-Micro/WIN Вам необходим персональный компьютер (PC) или устройство программирования (PG) с операционной системой фирмы Microsoft Windows 3.1x или Windows 95.

1. Вставьте дискету 1 в ваш компьютер, когда он уже работает.



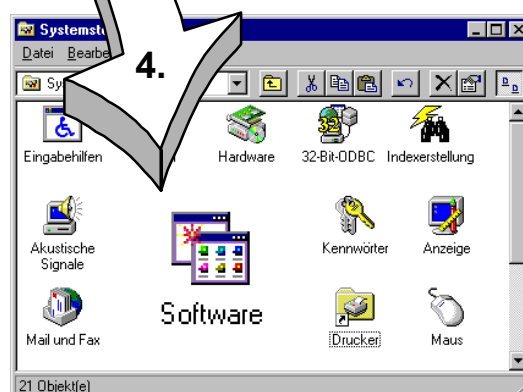
2. Вызовите стартовое меню кнопкой **Start** из строки задач.



3. Откройте **Control Panel** (**Systemsteuerung**) из меню **Settings** (**Einstellungen**).

4. Вызовите инсталлятор программ, дважды нажав мышью на пиктограмму **Add/Remove Programs** (**Software**)

Выберите кнопку **Install** (**Installieren**) и кнопку **Continue** (**Weiter**) в следующем диалоговом окне.

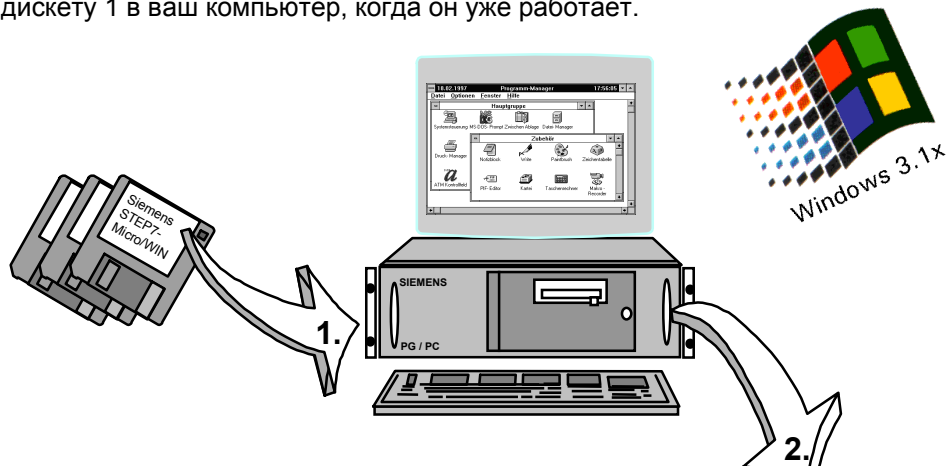


Инсталлятор найдет программу установки S7-Micro/WIN на дискете 1.

Затем запустите программу инсталляции. В конце инсталляции появится "Micro/WIN" в меню **Start > Programs** (**Start > Programme**).

Инсталляция программного обеспечения под Windows 3.1x

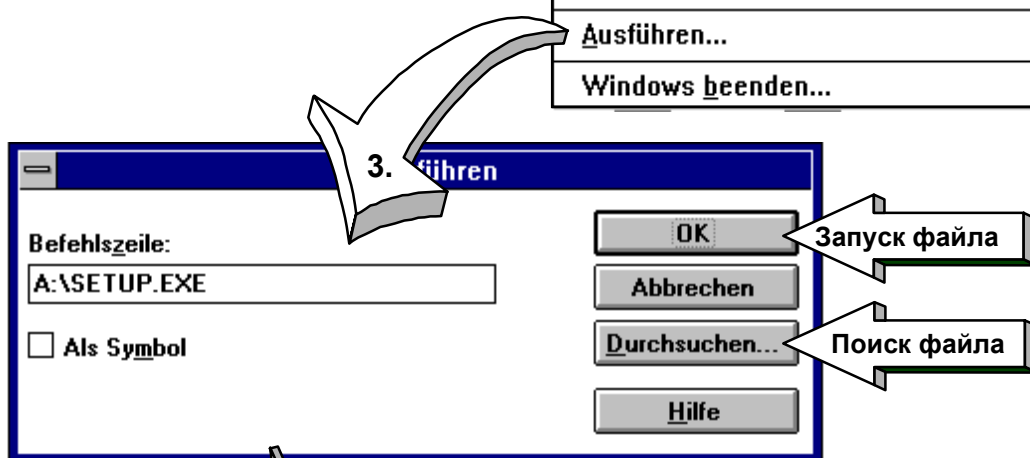
1. Вставьте дискету 1 в ваш компьютер, когда он уже работает.



2. Выберите из меню **File (Datei)** раздел **Run (Ausführen)**.



3. Найдите устройство A:\ и файл SETUP.EXE и выберите его. Запустите файл, кликнув мышью кнопку **OK**.

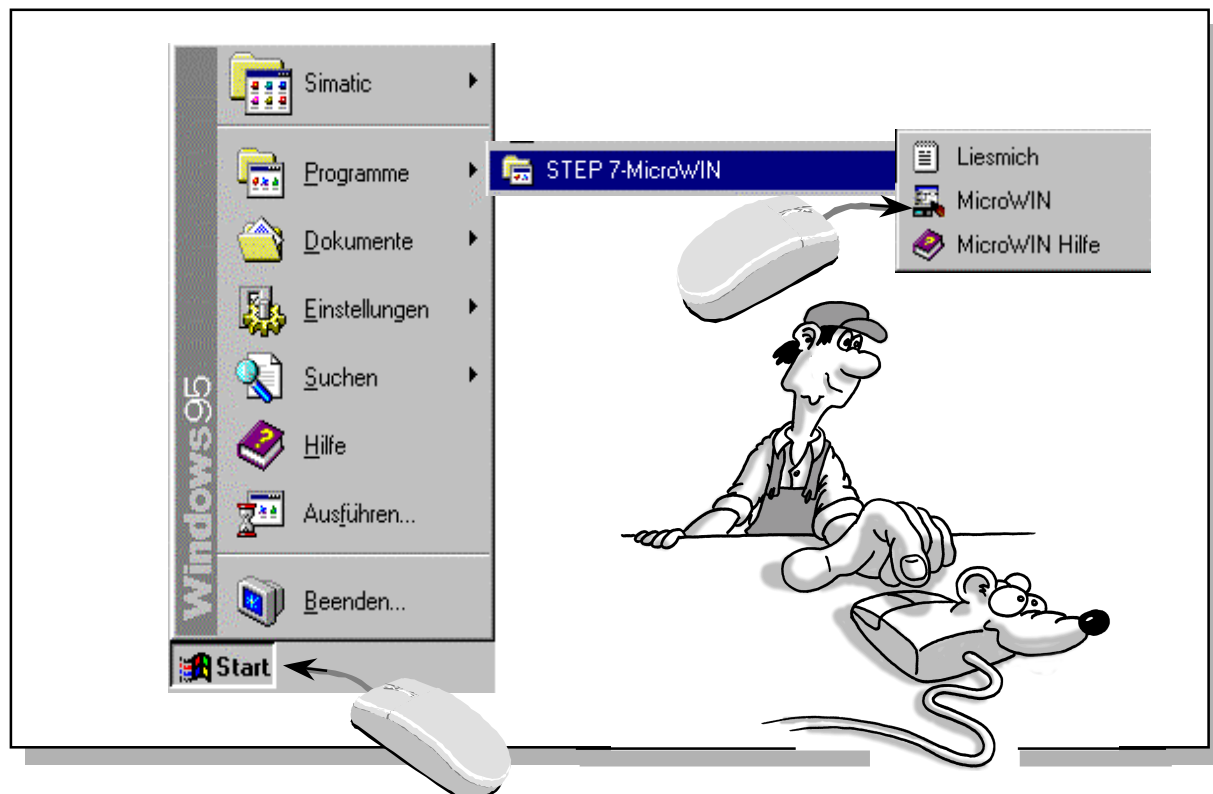


4. Следующие инструкции по инсталляции выполняются в соответствии с программой установки. В конце инсталляции вы увидите программную группу STEP 7-Micro/WIN.



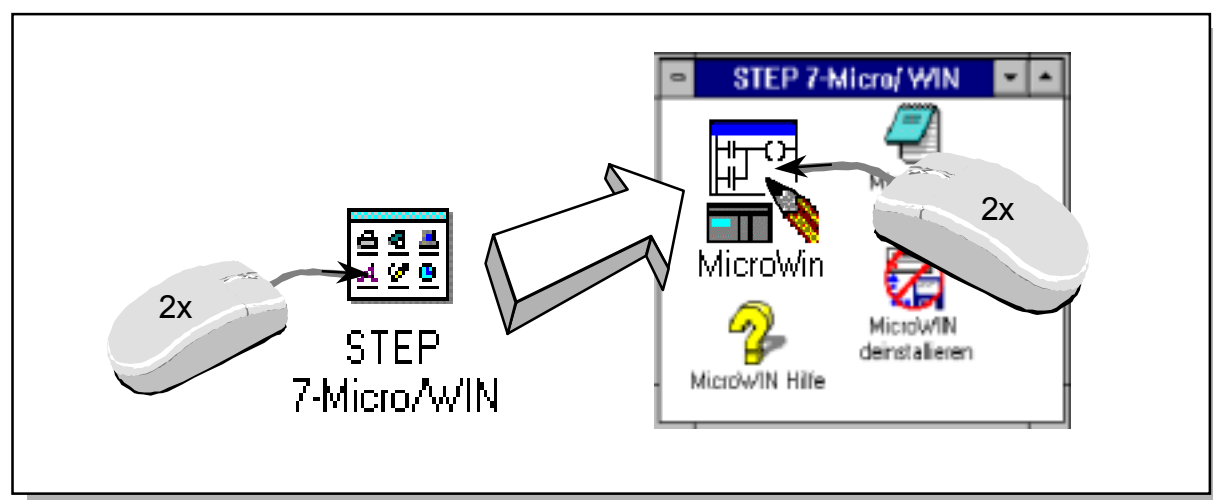
Запуск STEP 7-Micro/WIN

Windows 95



В разделе *Programs (Programme)*, которое вы вызываете из стартового меню, вы найдете папку *STEP 7-Micro/WIN*. Она содержит стартовую пиктограмму *STEP 7-Micro/WIN*. Запустите программу кликнув мышью на этой строке.

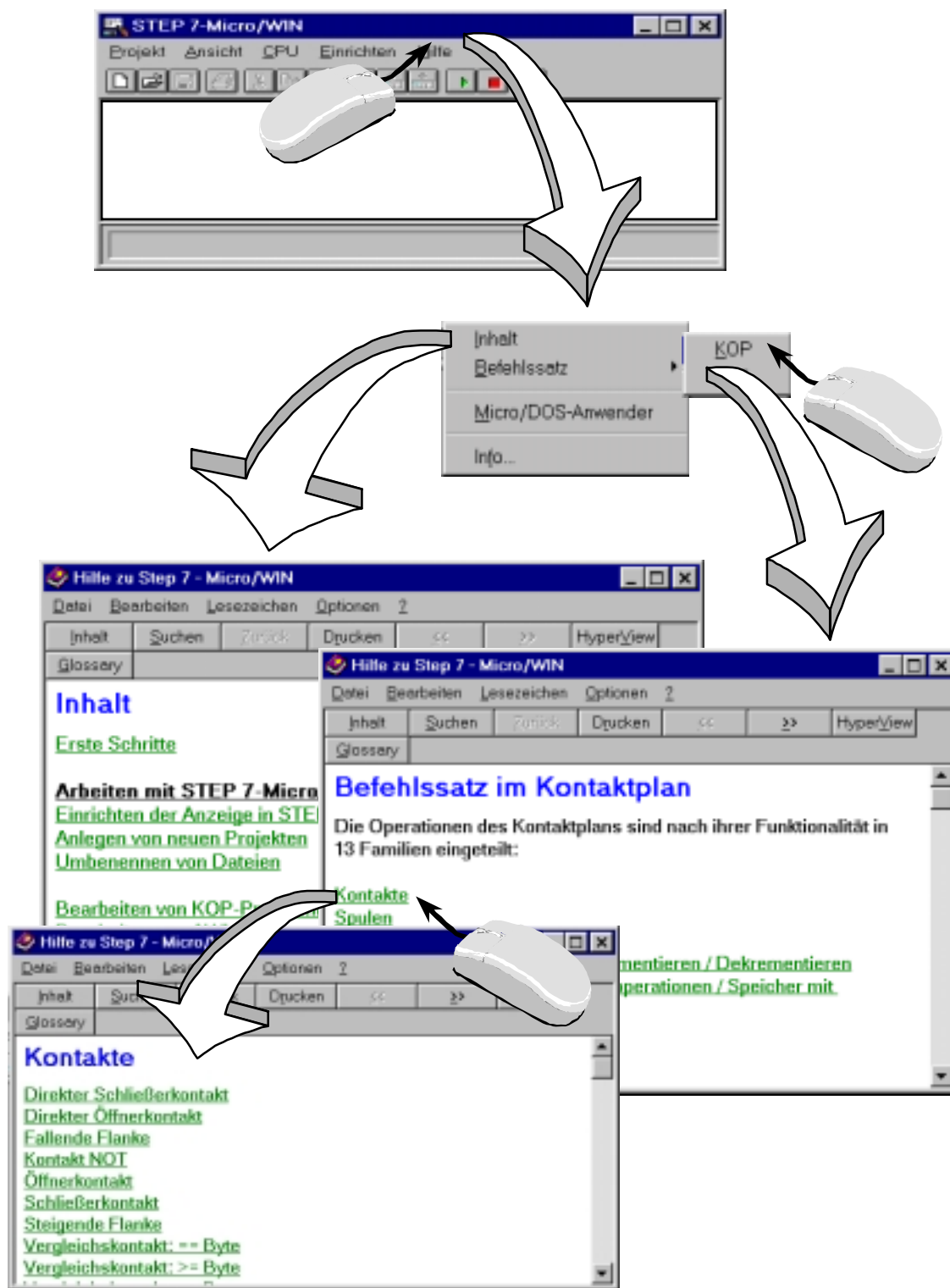
Windows 3.1x



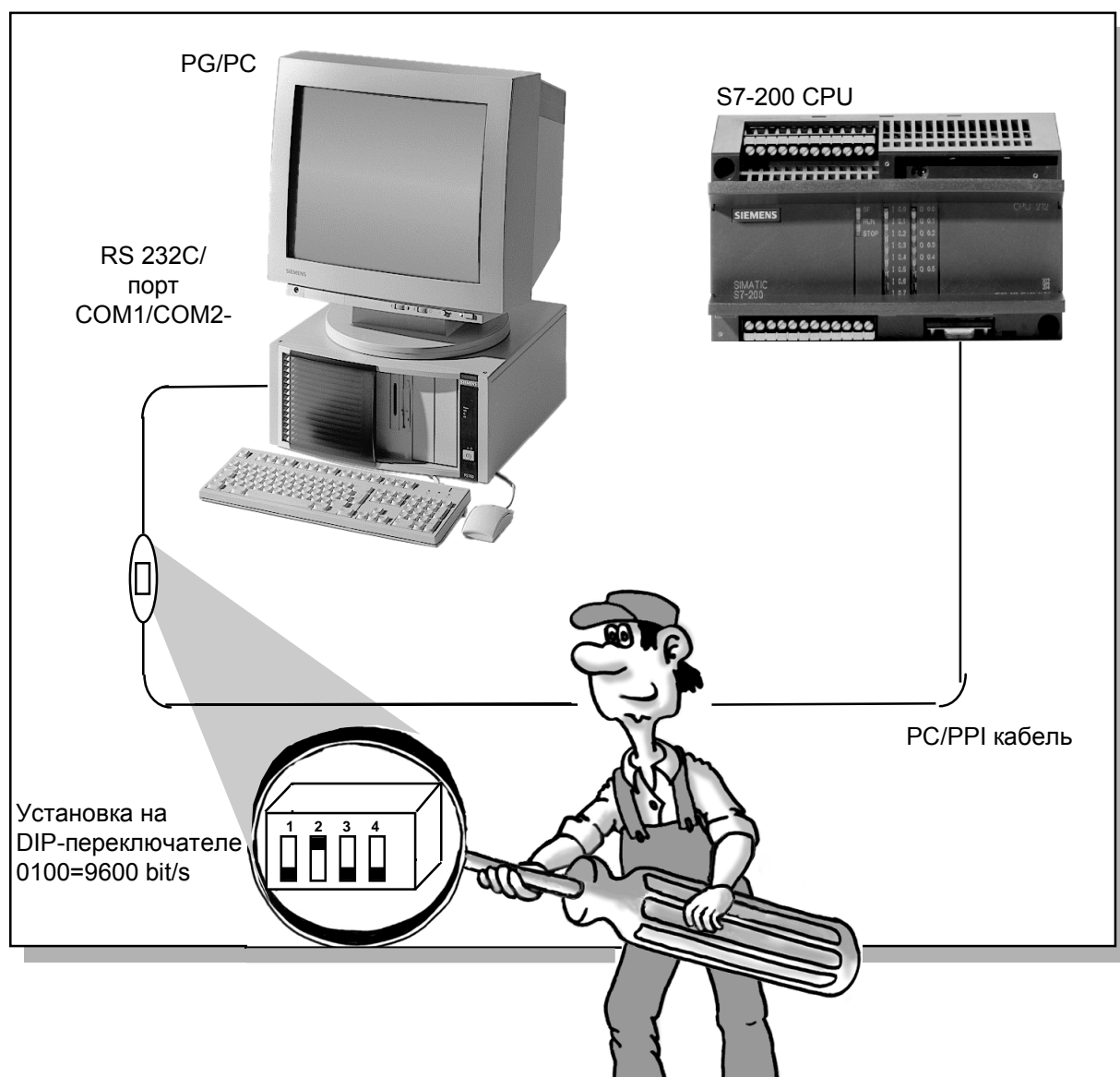
Запустите STEP 7-Micro/WIN под Windows 3.1x, кликнув мышью иконку *Micro/WIN*. Вы найдете ее в программной группе *STEP 7-Micro/WIN*.

Система помощи

STEP 7-Micro/WIN имеет удобную систему быстрой помощи подобно другим Windows-приложениям. Выбирая в меню раздел *Help (Hilfe)*, Вы можете получить информацию по разделам (*Contents/Inhalt*) или по инструкциям (*Instruction Sets/Befehlssatz*) STEP 7-Micro/WIN.



Установка скорости передачи



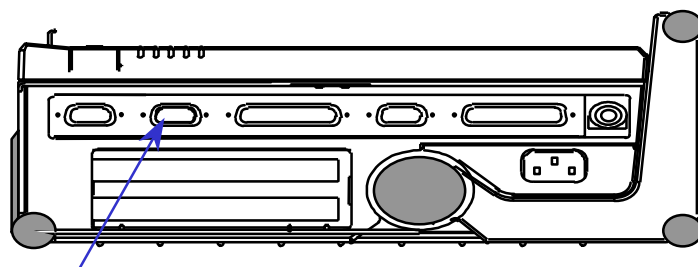
PC/PPI-кабель соединяет PC с контроллером S7-200.

На Вашем PC или PG Вы должны использовать последовательный порт с 9-штырьковым разъемом (или 25-штырьковым с адаптером), например, COM2.

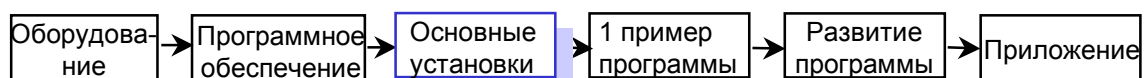
S7-200 передает данные со скоростью 9600 bit/s. Установите скорость передачи на PC/PPI-кабеле согласно схеме, представленной на рисунке. Затем вставьте PC/PPI -кабель в PC и S7-200.

Используйте одну и ту же фазу питающего напряжения на PC и контроллере во избежание различий в напряжении.

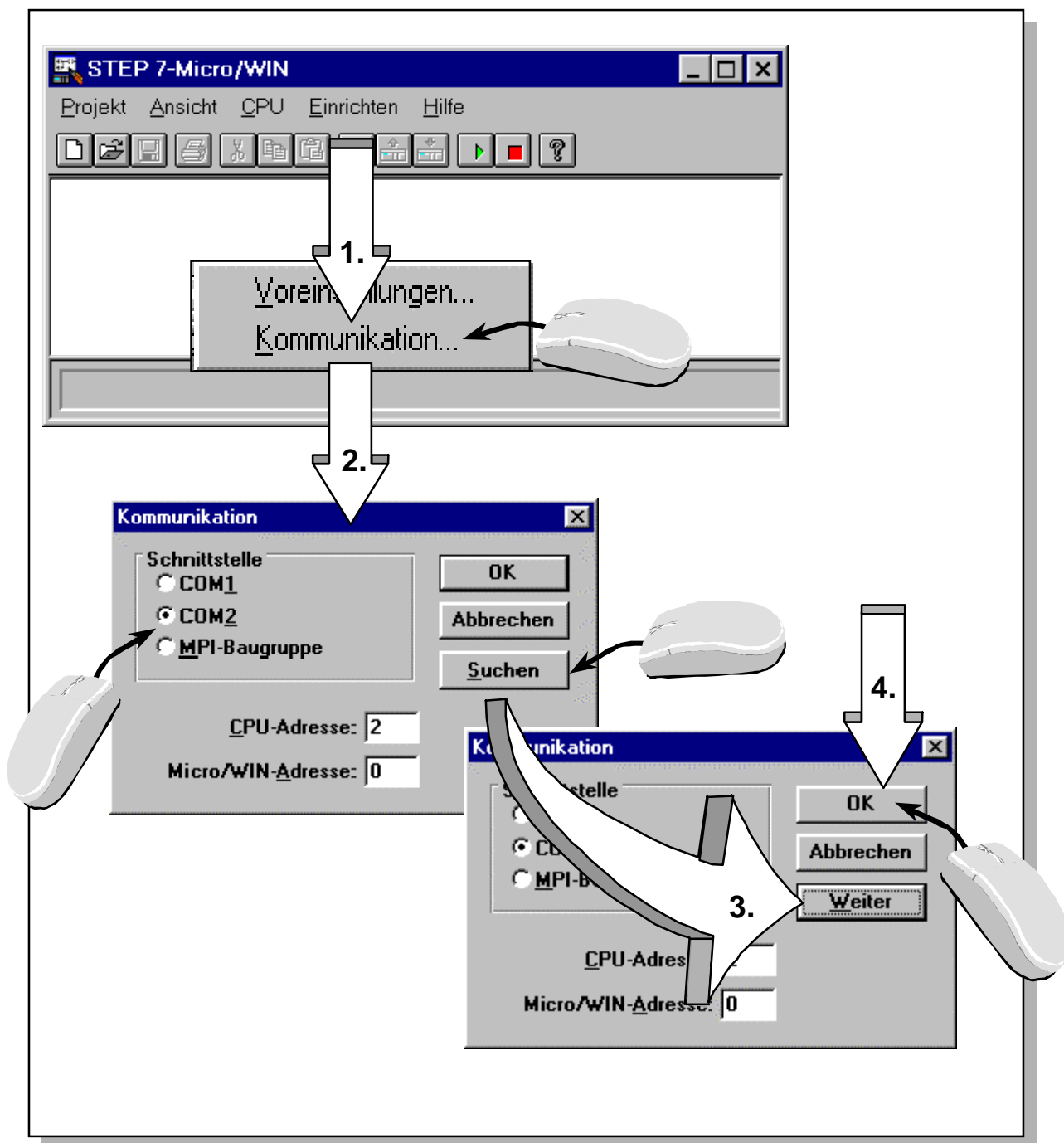
Подайте рабочее напряжение на контроллер (загораются индикаторы STOP или RUN).



COM2-порт на программаторе PG 740 (вид слева)



Конфигурирование порта на PC/PG



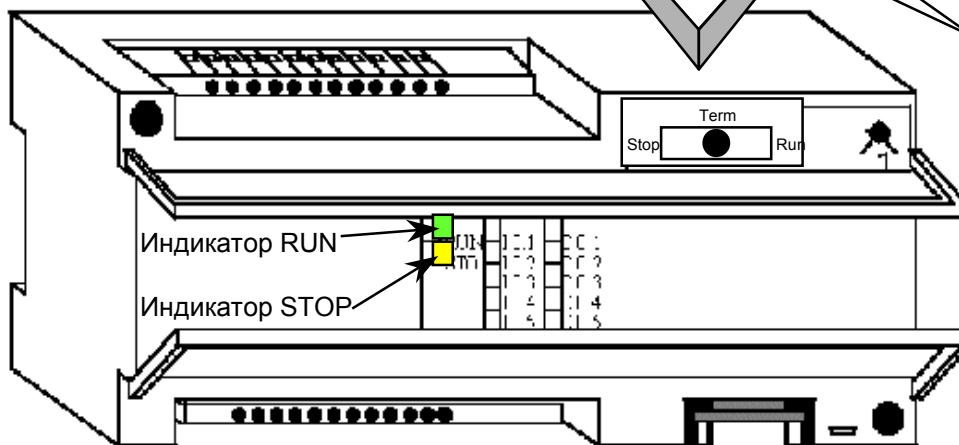
1. Выберите пункт меню *Setup* > *Communication* (*Einrichten* > *Kommunikation*).
2. Выберите последовательный порт (COM1/COM2), в который Вы вставили PC/PPI-кабель.
3. Вы можете использовать кнопку *Find* (*Suchen*) чтобы проверить, правильно ли сконфигурирован порт.
Если порт сконфигурирован правильно, на кнопке *Find* (*Suchen*) появится текст "Next" ("Weiter"). Если нет, то появится сообщение об ошибке. В случае ошибки выберите другой порт.
4. Подтвердите корректную конфигурацию клавишей *OK*.

Первый тест на работоспособность

1. Установите переключатель режимов в контроллере в положение Term. Переключатель находится под верхней крышкой контроллера..

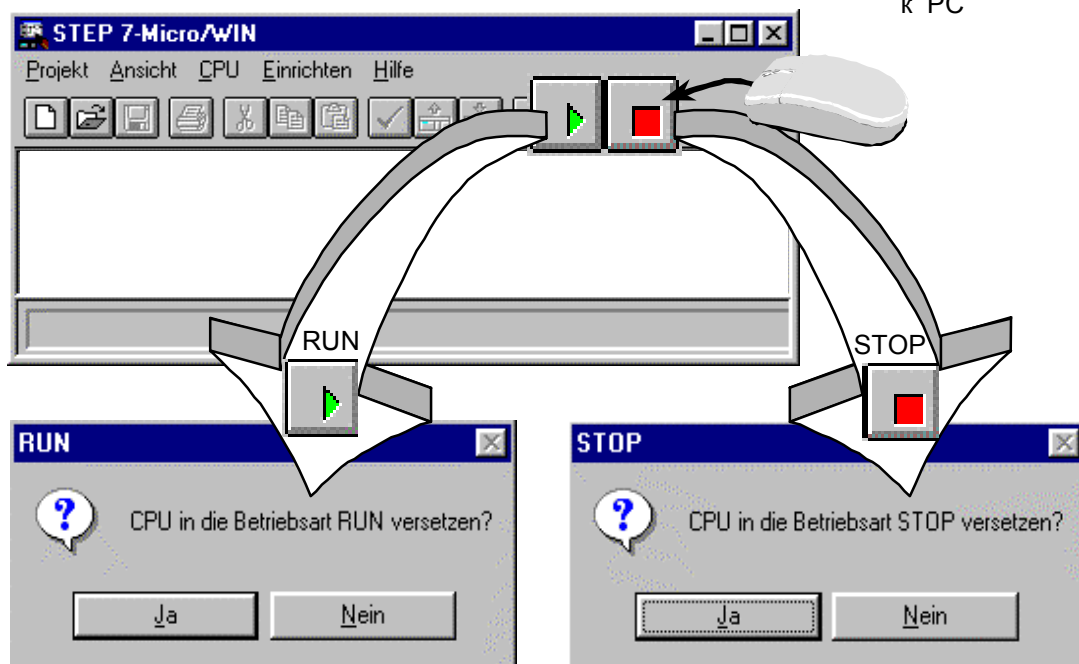
TERM

Только в положении **TERM** возможно дистанционное управление режимом (RUN или STOP) от PC/PG.



2. Со стороны PC переключите S7-200 в режим STOP и обратно в RUN.

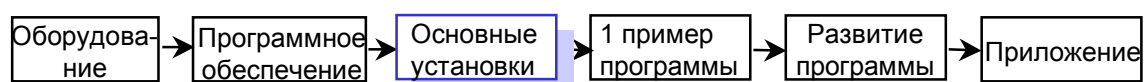
PC/PPI-кабель к PC



В режиме RUN ("Работа") в контроллере горит зеленый индикатор RUN. В состоянии STOP ("Останов") горит желтый индикатор STOP.

Если Вы смогли переключить режимы контроллера дистанционно от PC, то соединение между PC и контроллером сконфигурировано правильно.

Если режимы не менялись, проверьте соединительный кабель между PC и контроллером, установленную скорость передачи на кабеле PC/PPI или убедитесь, что правильный порт выбран в меню **Setup > Communication (Einrichten > Kommunikation)**.



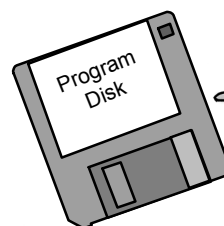
Записываем свою первую программу

Отлично, первый рабочий тест был успешно выполнен.

Контроллер запускается и передача данных работает. Но как мне записать программу?



Сейчас ты быстро научишься как по шагам программировать основные функции в контроллере, используя за основу небольшой программный пример на дискете.



Прежде всего, ты должен научиться открывать программный пример с дискеты (или винчестера) и передавать его в контроллер с помощью STEP 7-Micro/WIN.



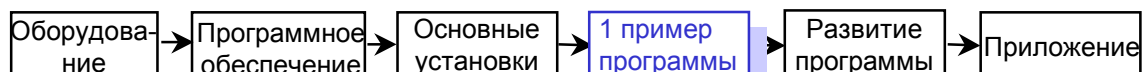
После этого, ты сам познакомишься с функциями программы, с передачей и тестированием ее.



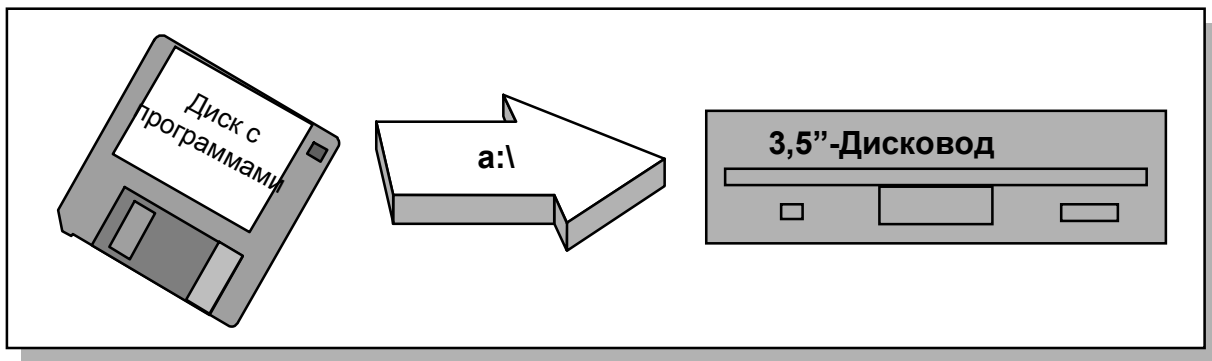
Основные краткие знания о "логических операциях" ты сможешь получить путем анализа элементов учебной программы и понимании инструкций в ней.



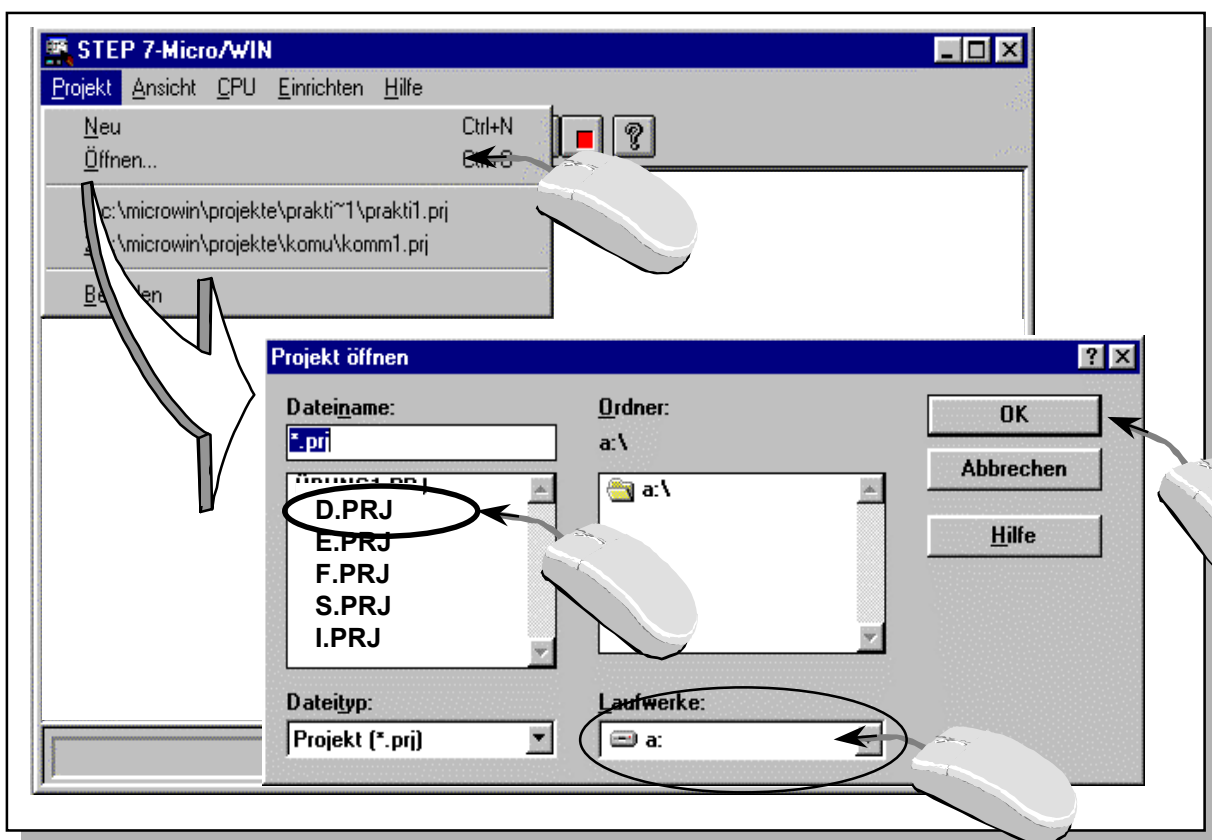
В разделе "Больше упражнений" ты научишься наиболее важным знаниям, необходимым для записи своей собственной программы.



Чтение 1-го программного примера



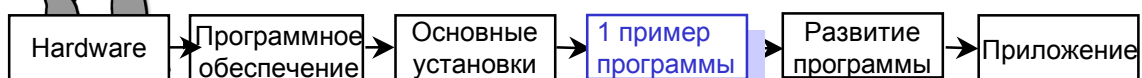
1. Вставьте дискету с упражнением из стартового набора в дисковод 3.5". Дискета находится в конверте на последней странице руководства.



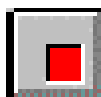
2. Из меню **Projekt** > **Open** (**Projekt** > **Öffnen**) загрузите 1-е программное упражнение с дискеты в редактор STEP 7-Micro/WIN. Буква в имени программы означает язык комментариев в программе (для английского -E.PRJ, немецкого -D.PRJ). Сперва Вы должны выбрать имя вашего дисковода 3.5".



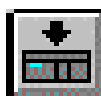
STEP 7-Micro/WIN загрузил программное упражнение в рабочую память PC/PG и показывает первые элементы программы на экране.



Загрузка 1-й учебной программы в контроллер



Щелкните мышью на эту кнопку для того, чтобы переключить контроллер в режим STOP. При этом переключатель режимов в контроллере должен быть в положении TERM, а сам контроллер находится в режиме RUN (загорится индикатор STOP).
Внимание: Это означает, что подключенная модель больше не управляется.



Щелкните мышью на эту кнопку для загрузки программы, которая отображается на экране, в контроллер.



Щелкните мышью на эту кнопку для переключения контроллера в режим RUN. При этом переключатель в контроллере должен быть в положении TERM, а сам контроллер находится в режиме STOP (загорится индикатор RUN).
Внимание: Подключенная модель начнет немедленно работать, подчиняясь действиям вашей программы.



Предупреждение

Опасайтесь повреждений и материального ущерба

Вы также можете загрузить программу из контроллера в PC/PG.



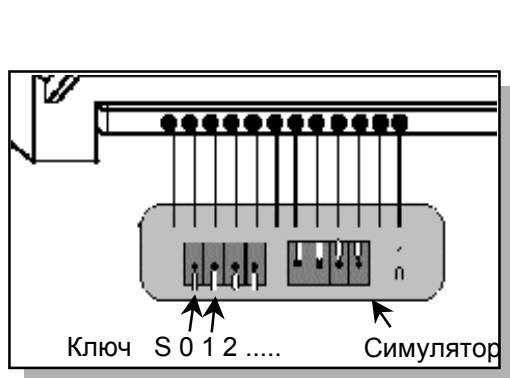
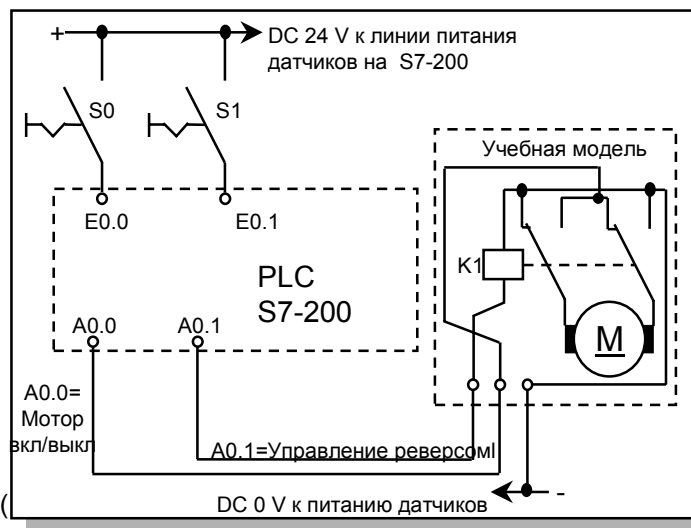
Щелкните мышью на эту кнопку для загрузки программы из контроллера в PG/PC. Это переписшет текущую программу, отображаемую на мониторе. Вы всегда, следовательно, сможете сделать так, чтобы актуальную версию вашей программы иметь на диске или дискете.

Функционирование и тестирование учебной программы

В **упражнении к программе 1** ключ S0 используется для включения мотора. Ключ S1 используется для управления направлением вращения мотора.

В **учебном комплекте** переключатели S0 и S1 находятся на симуляторе. Симулятор подключает 24 V к входам и I0.1 (E0.0 и E0.1). Учебная модель подключена к выходам контроллера Q0.0 (A0.0) -Мотор вкл/выкл- и Q0.1 (A0.1) - Управление реверсом-.

Состояние сигнала входа I0.0 (E0.0) связано в **программе** с выходом Q0.0 (A0.0). Состояние сигнала входа I0.1 (E0.1) связано с выходом Q0.1 (A0.1).



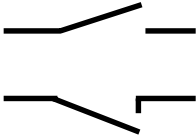

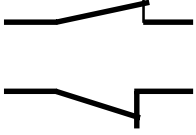

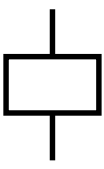
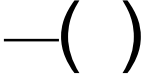
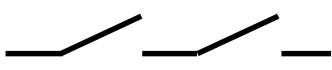
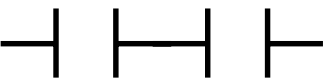
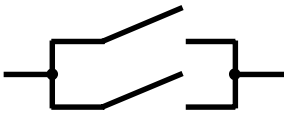
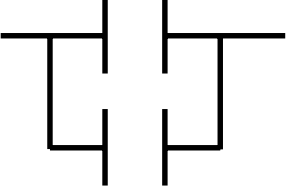

Сейчас будем тестировать его! Управляющее напряжение подключено. Установка правильно соединена. Вы уже загрузили программу с дискеты в программный редактор и передали ее в контроллер (PLC). PLC находится в режиме RUN (горит зеленый индикатор RUN). Сейчас будем управлять ключами S0 и S1 и наблюдать за работой модели.

Действие	Реакция		
Ключ S0 включен	Индикатор I0.0 горит	Индикатор Q0.0 горит	Мотор вращается
Ключи S0 и S1 включены	Индикаторы I0.0 и I0.1 горят	Индикаторы Q0.0 и Q0.1 горят	Мотор вращается в другую сторону



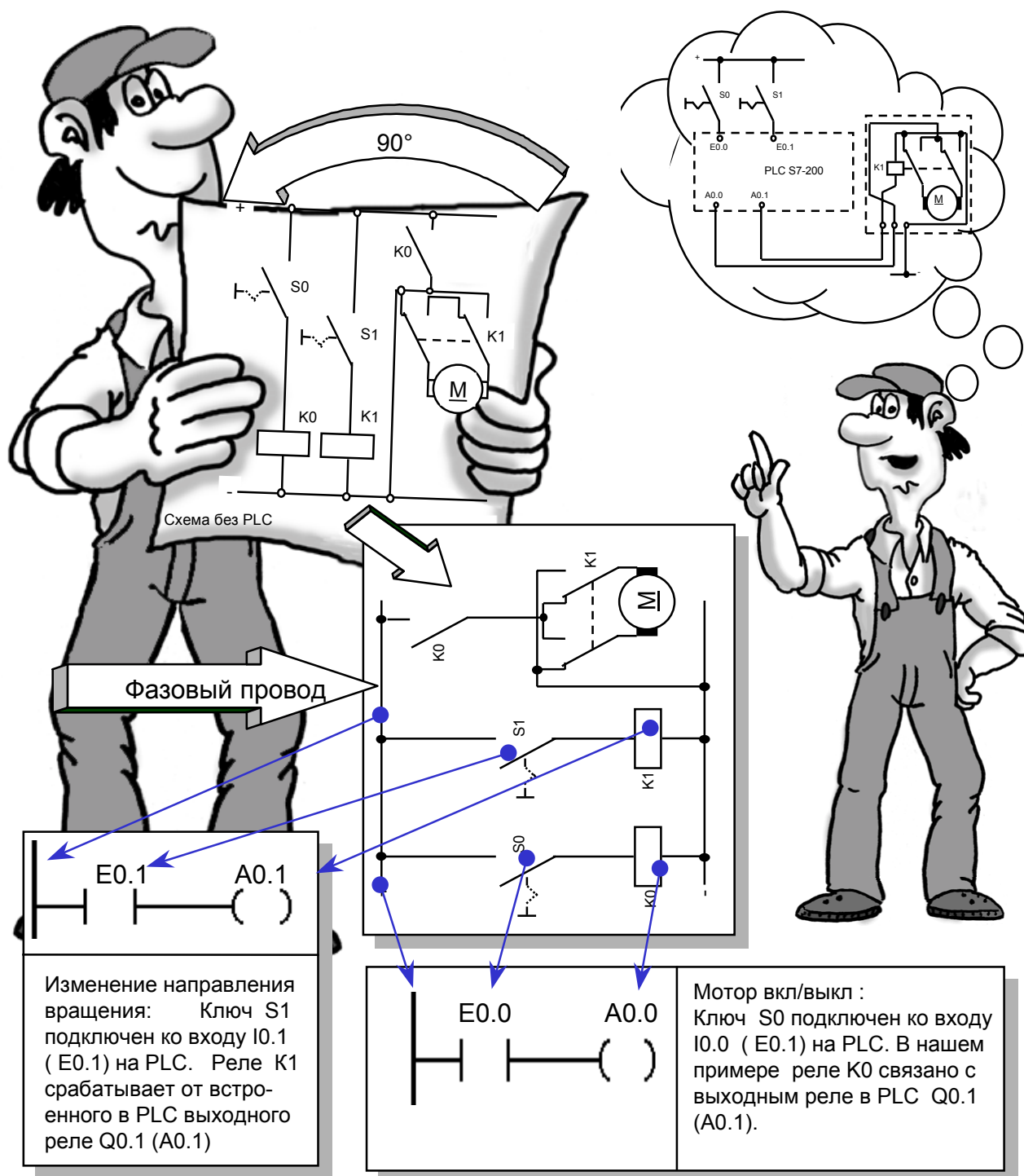
Индикаторы с I0.0 по I0.7 индицируют состояние сигналов на входах I0.0- I0.7 (E0.0 - E0.7). Индикаторы с Q0.0 по Q0.7 индицируют состояние сигналов на выходах Q0.0- Q0.7 (A0.0 - A0.7). Символы I и Q обычно используются в мире для обозначения входов и выходов.

Логические элементы

Контактор		Инструкция для PLC в соответствии с функцией
	<p>Опрос : Ток течет? Если да, тогда результат опроса истина . (Результат равен "1")</p>	
	<p>Опрос : Ток не течет? Если да(нет тока) , тогда результат опроса истина . (Результат равен "0")</p>	
	<p>Катушка : Если значение "истина" и ток проходит через катушку, то она активируется. (Катушка запускается).</p>	
	<p>Последовательная цепочка: (Логика "И"). Первый ключ И второй ключ должны быть замкнуты для протекания тока.</p>	
	<p>Параллельная цепочка: (Логика "ИЛИ"). Первый ключ ИЛИ второй ключ должны быть замкнуты для протекания тока.</p>	
	<p>Это элемент управляющей программы обозначает конец главной программы.</p>	

В цифровой логике состояниями сигналов являются "0" или "1". Состояние "0" обозначает "ложь", а состояние "1" - "истину". Можно также сказать, что опросом является "0" (ложь) или "1" (истина).

Преобразование электрической схемы



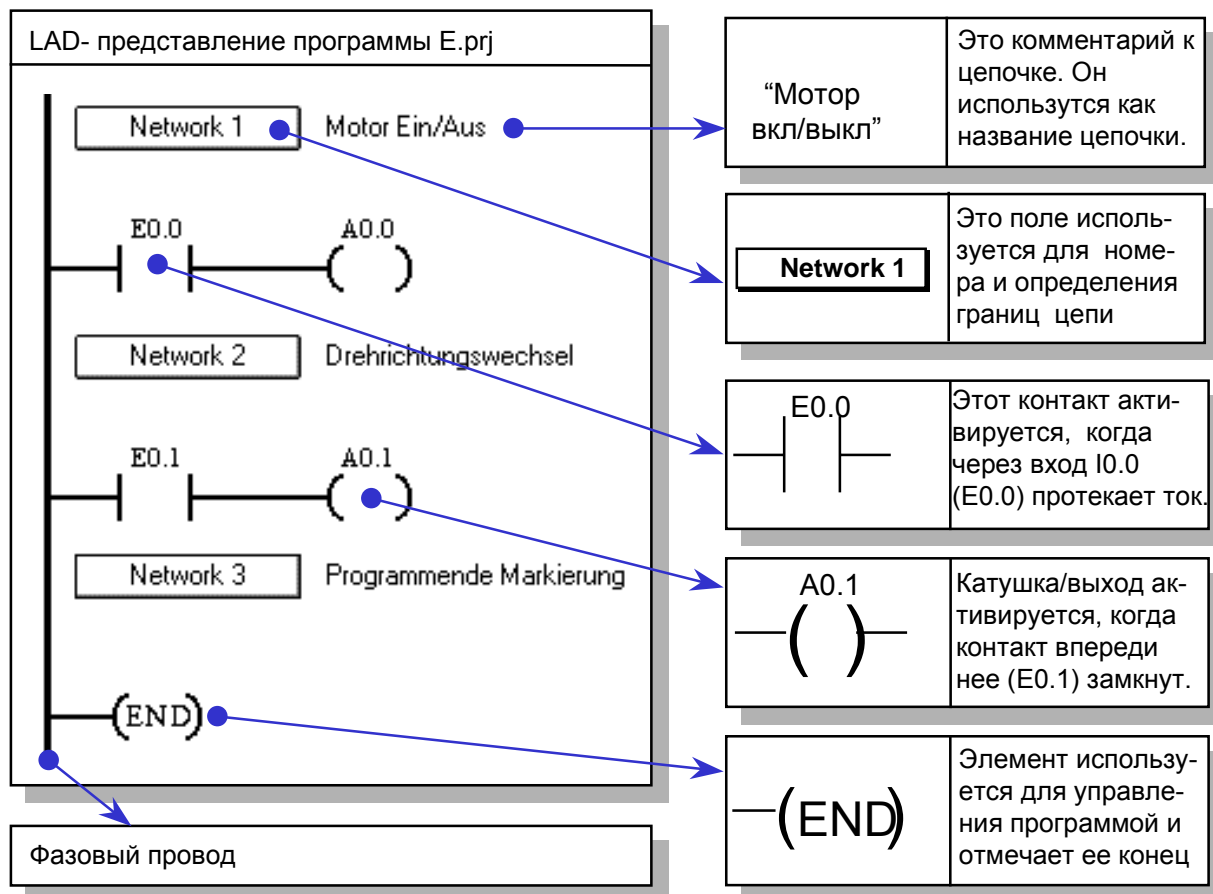
Как Вам преобразовать электрическую схему в программу для PLC?

Поверните Вашу схему на 90° влево. Ваш фазовый провод будет тогда расположен слева, а земляной провод - справа. Посередине Вы будете видеть переключательные элементы Вашей схемы. Схемная логика машины (например, реле времени или триггеры), которая используется в качестве инструментов, записывается теперь вместе с переключателями, контактами и управляющими реле и т.д. и управляется от PLC.

Элементы управления, такие как переключатели входов, переключатели направлений и т.д., со стороны входа и силовые ключи (такие как контакты мотора, переключатели реверса, и т.д.) со стороны выхода не могут быть описаны в PLC.

Элементы программы 1

Посмотрите внимательно на структуру PLC-программы, представленной в контактном плане, т.е. на языке LAD (KOP). Этот тип представления наиболее близко подходит к электрической схеме.



STL-программа

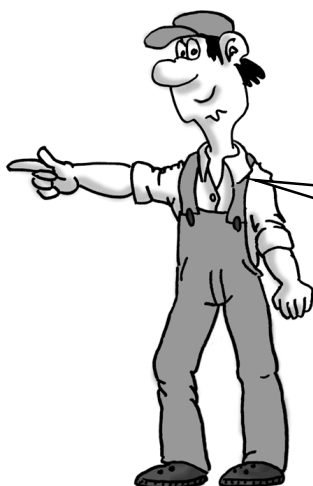
```

NETWORK
LD    E0.0
=     A0.0

NETWORK
LD    E0.1
=     A0.1

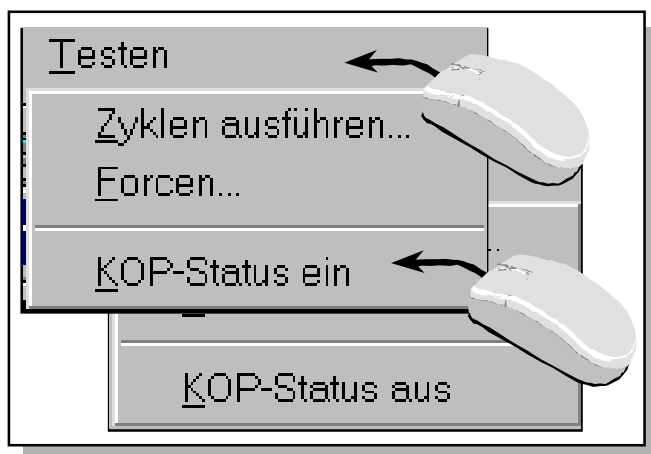
NETWORK
MEND
    
```

Лист с инструкциями, пример которого показан на иллюстрации “STL-программа, является другим типом представления программы. Вы можете переключиться между этими двумя типами в меню [View > LAD/STL \(Ansicht > KOP/AWL\)](#)



Цепочки или сегменты используются для структурирования программы. Вы вставляете отдельный элемент схемы в цепочку. Инструкция конца программы END (MEND в STL) должна быть в отдельной цепочке.

Наблюдение статуса программы (online)



Выберите раздел меню *Debug > Ladder Status On* (*Testen > KOP-Status ein*) для активации режима наблюдения за состоянием программы в LAD-представлении.

Вы можете теперь наблюдать состояние операндов в контроллере.

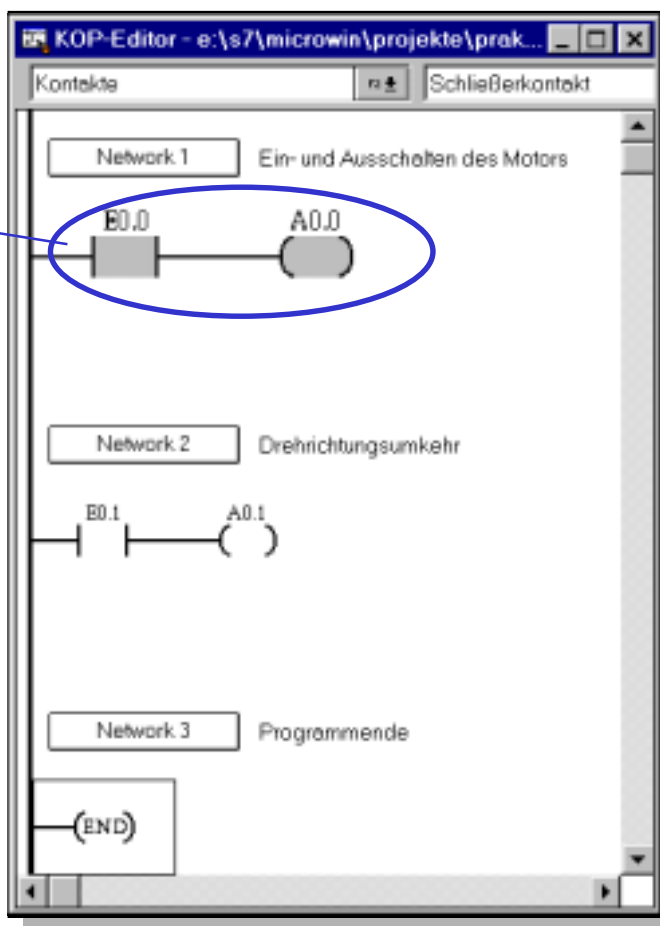
Выберите *Debug > Ladder Status Off* (*Testen > KOP-Status aus*) для снятия режима статуса.

В примере контакт S0 соединен с входом 0.0. Если Вы активируете ключ и будете наблюдать за LAD-программой в режиме статуса, то операции, которые пропускают ток, т.е. истинны (=“1”), заполняются серым цветом.

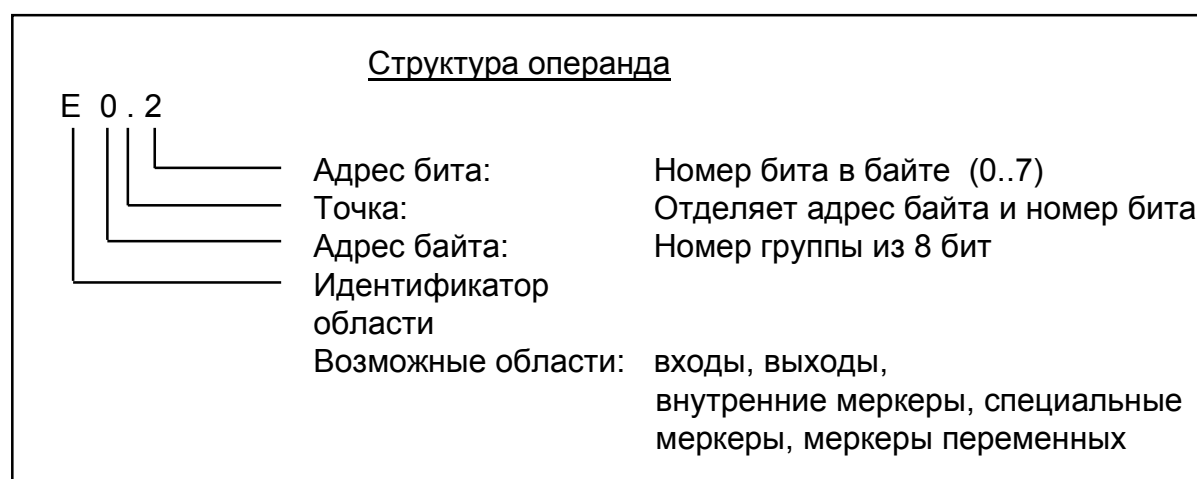
Статус операций циклически читается из контроллера и обновляется на экране.

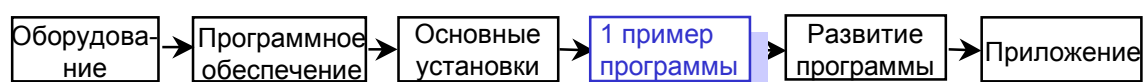
В этом контексте, средства Online-связи позволяют взглянуть на контроллер изнутри с помощью PC/PG, наблюдая за текущим состоянием и циклическими изменениями операндов.

Очень быстрые события не могут быть отслежены, так как время, требуемое для передачи выходных данных на экран, довольно большое.



Структура элемента LAD-программы





1. Модификация программы: операция “И”

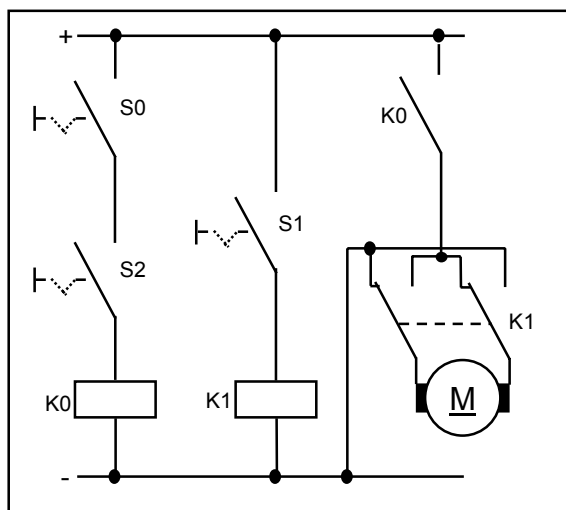


Схема без контроллера

Задача:

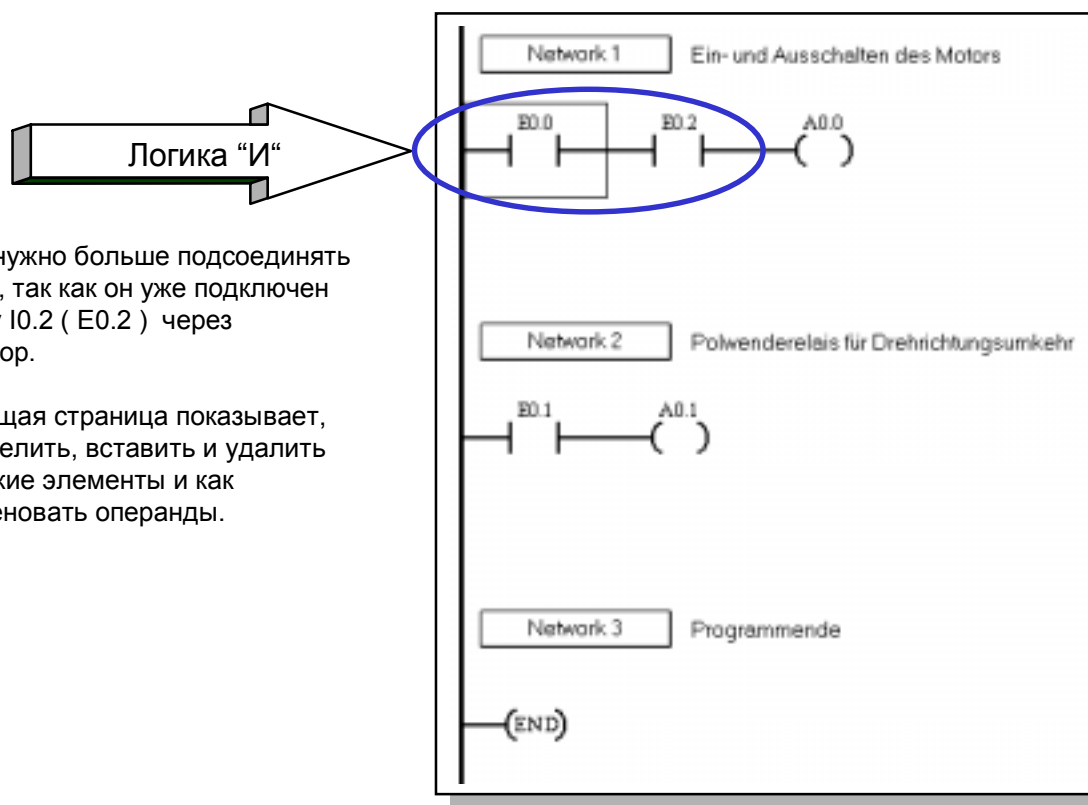
Для того чтобы включать мотор, также с ключом S0 активируется и ключ S2.

Как и в 1-м варианте, ключ S1 будет использован для управления реверсом.

Описание функции показанной выше:

Мотор запускается, когда S0 И S2 активируются. В LAD- представлении это означает: когда контакты I0.0 И I0.2 (E0.0 И E0.2) закрыты, ток течет через катушку Q0.0 (A0.0). Контакты соединены последовательно (логика “И”).

В нашей программе логика “И” представлена следующим образом:



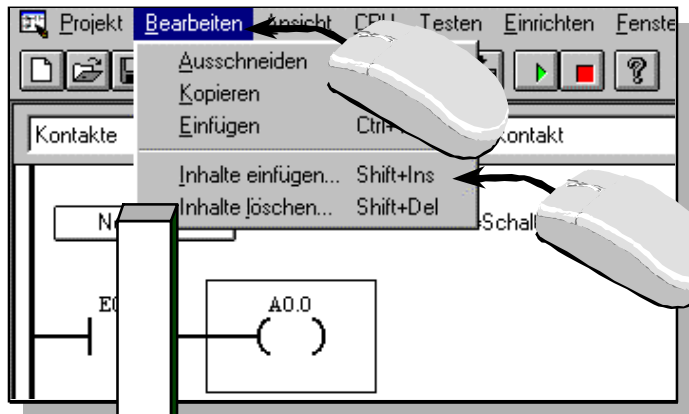
LAD- диаграмма схемы

Вам не нужно больше подсоединять ключ S2, так как он уже подключен ко входу I0.2 (E0.2) через симулятор.

Следующая страница показывает, как выделить, вставить и удалить логические элементы и как переименовать операнды.

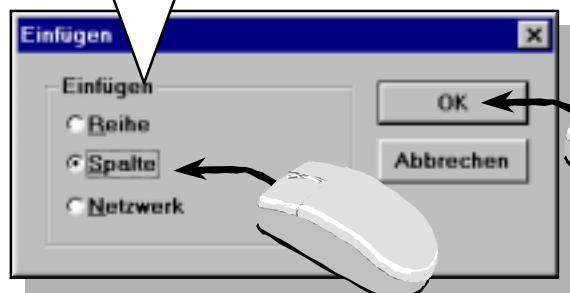
Вставка логического элемента

Если Вы далее захотите соединить нормально открытый контакт I0.2 последовательно между контактом I0.0 (E0.0) и катушкой Q0.0 (A0.0), то сперва необходимо найти свободное место.

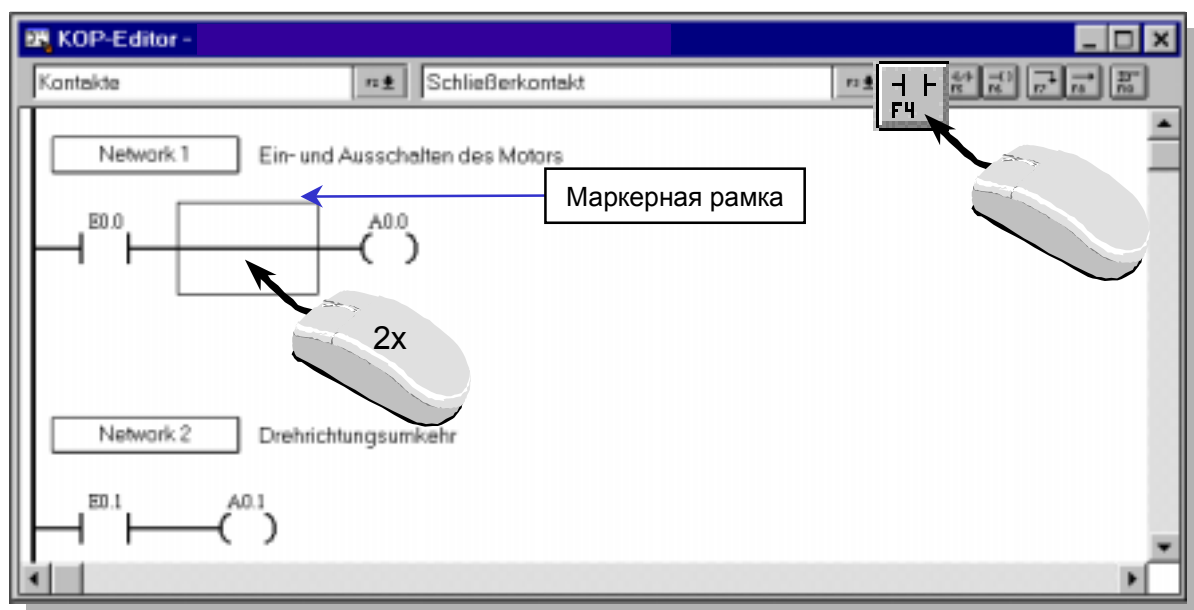


Выберите меню **Edit > Insert** (**Bearbeiten > Inhalte einfügen**).
Диалоговое окно предлагает Вам определить, что вставлять.

Перед этим щелкните мышью на то место, где должен быть расположен новый элемент (выделенное место показывается рамкой). В нашем примере это будет Q0.0 (A0.0)



В этом окне Вы можете выбрать, что вставлять. Для текущей задачи Вам необходимо выбрать **Column (Spalte)**-столбец. Для подтверждения выбора щелкните на кнопку **OK**, и в программу вставится столбец. Теперь Вы можете вставить наш элемент.

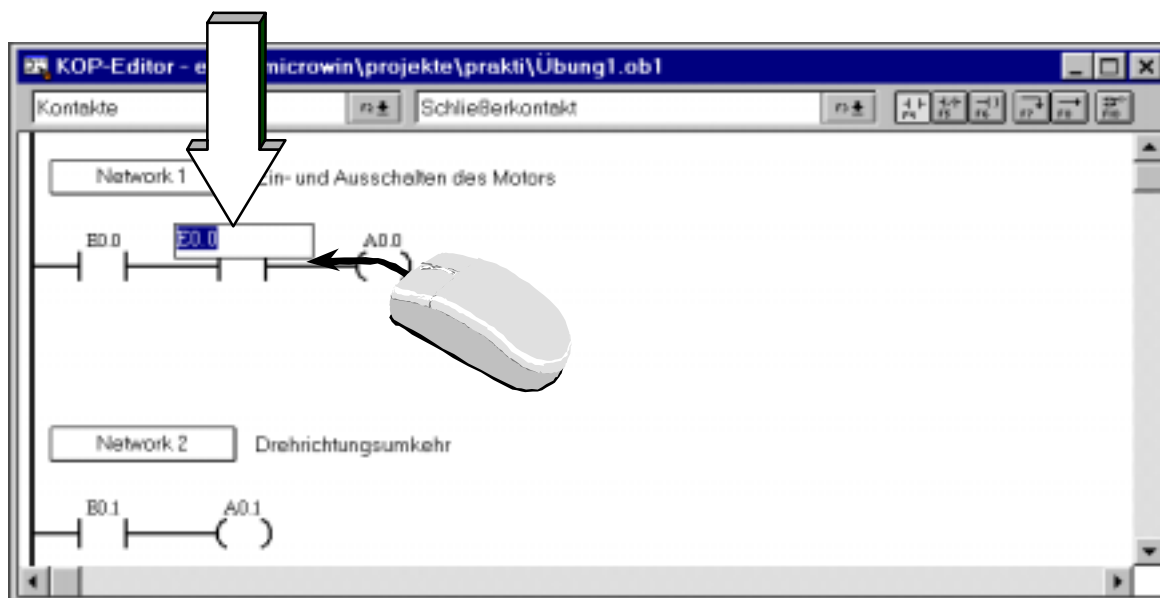


Одним щелчком мыши Вы можете выбрать логическую операцию.
Двойным щелчком мыши в маркерной рамке вставится выделенная логическая операция.

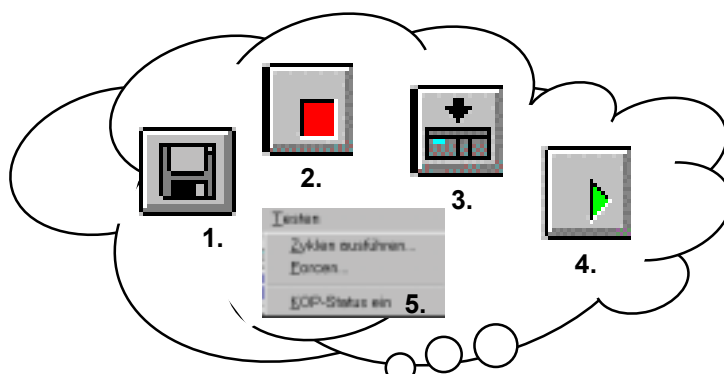
Вставка операнда и тестирование

Когда новый элемент вставлен, Вам сразу нужно определить правильно операнд.
STEP 7-Micro/WIN отображает поле ввода с операндом по умолчанию сразу после вставки элемента. Перепишите этот операнд на правильный. В нашем случае I0.2 (E0.2).

Подтвердите ввод нажатием на клавишу Enter.
Не забывай те сохранять свои изменения!



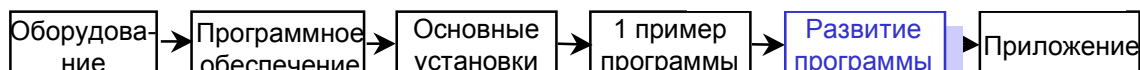
Если Вы хотите ввести или изменить символ в операнде, Вы можете выделить его и щелкнуть через некоторое время мышью на поле операнда.



Если Вы хотите тестировать изменения ...

Загрузите измененную программу в контроллер и проверьте ее.
Она будет нужна для активации ключей S0 и S2 и запуска мотора.

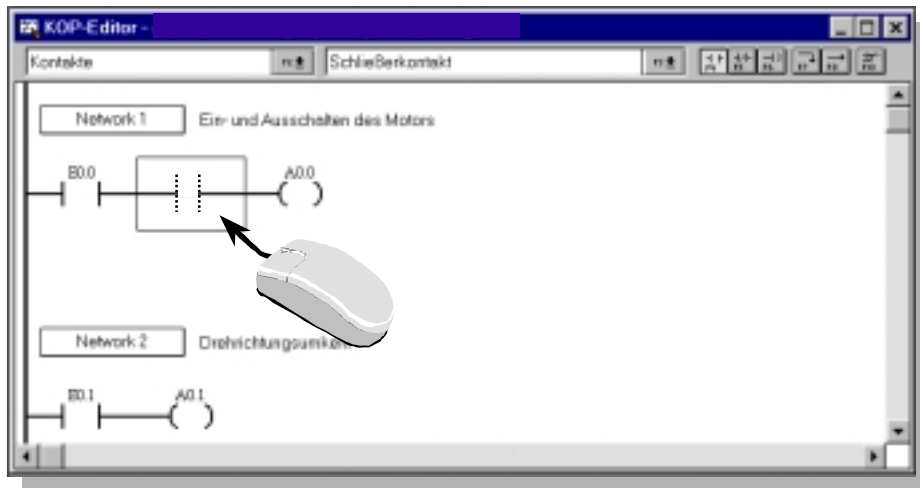
Просмотрите Вашу LAD-программу в режиме статуса и наблюдайте за состояниями ключей.



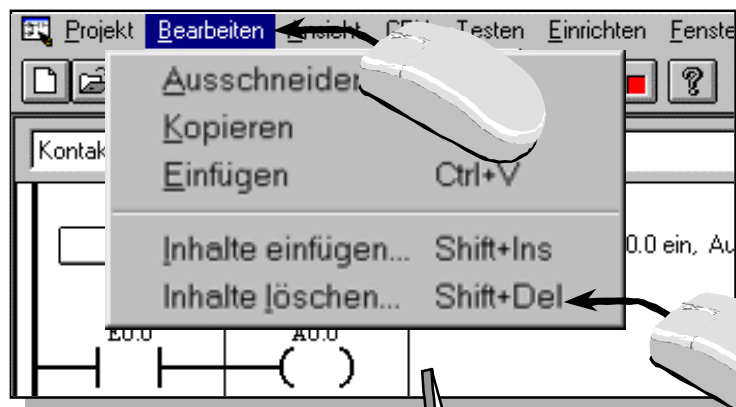
Удаление ...

Логического элемента

Если Вы ввели неправильный логический элемент и желаете удалить его, выделите мышью элемент и нажмите на клавишу **DEL**.

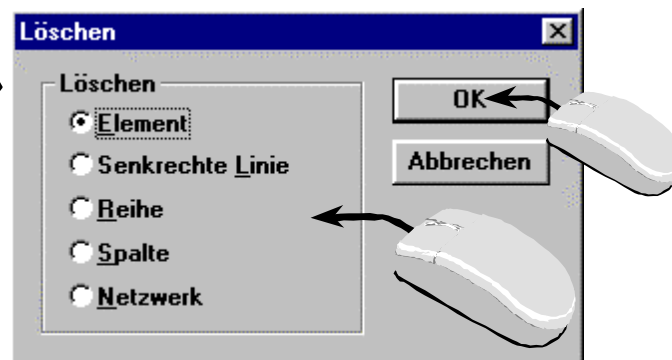


Столбцов, строк, цепочек и линий



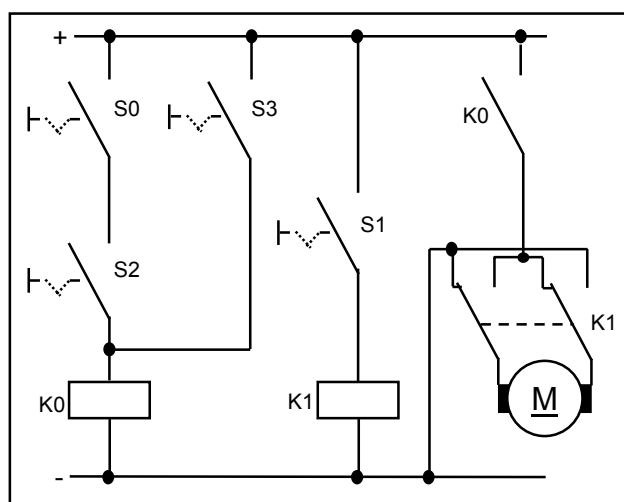
Если Вы желаете удалить столбец, строку, линию или цепочку, выделите удаляемый объект и выполните следующее:

Откройте окно Delete (Löschen) путем выбора меню **Edit > Delete** (**Bearbeiten > Inhalte löschen**).



В окне Delete (Löschen) Вы можете теперь выбрать то, что желаете удалить. Когда Вы активируете кнопку **OK**, то выбранный объект удаляется.

2-ая модификация программы: операция “ИЛИ”



Schaltung ohne SPS

Задача:

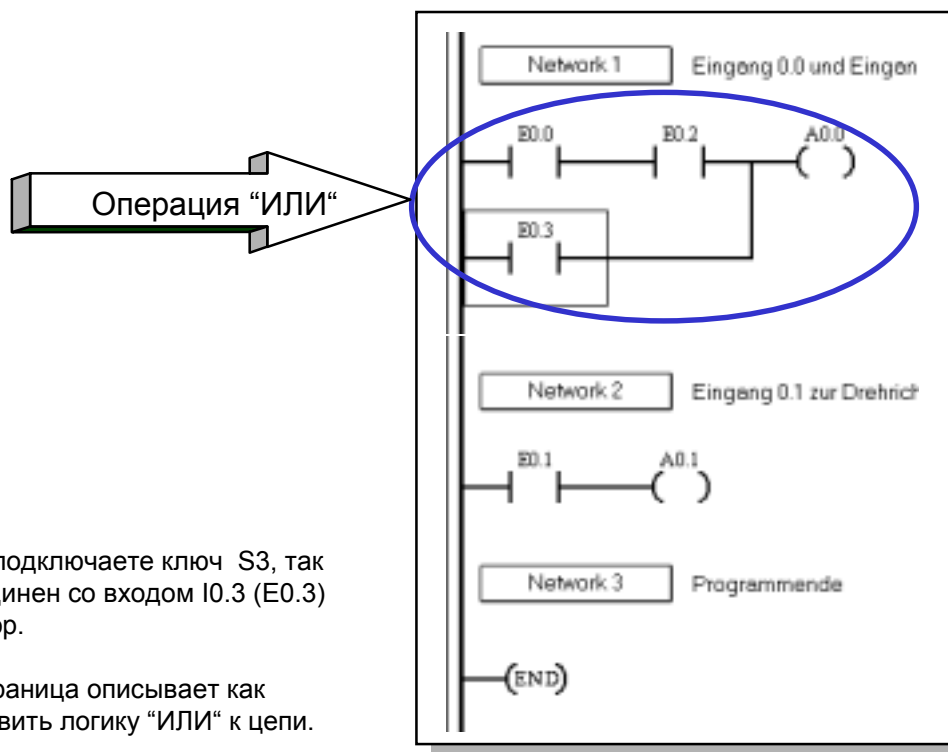
Ключи S0 и S2 в упражнении активируются для включения мотора. Ключ S3 используется одиночно как альтернатива для включения мотора. Ключ S1 используется для управления реверсом мотора.

Описание функции, показанной выше:

Когда (S0 И S2) ИЛИ S3 активируются, мотор запускается. В LAD-схеме это означает: Когда контакты I0.0 И I0.2 (E0.0 И E0.2) ИЛИ I0.3 (E0.3) закрыты, ток течет от фазового провода к катушке Q0.0 (A0.0).

S0 и S2 с S3 соединены параллельно (операция “ИЛИ”).

В нашей программе эта логика записывается следующим образом:



Вы больше не подключаете ключ S3, так как он уже соединен со входом I0.3 (E0.3) через симулятор.

Следующая страница описывает как выбрать и вставить логику “ИЛИ” к цепи.

LAD- программа для схемы

Вставка элемента “ИЛИ”

1. Выберите нормально открытый контакт.

2. Дважды щелкните мышью в позицию, в которую должен быть установлен контакт, и укажите его как I0.3 (E0.3).

3. Вам нужно еще дополнительно соединительную линию, идущую параллельно контактам I0.0, I0.2

4. Вставьте соединительную линию двойным щелчком позади контакта I0.3 (E0.3).

5. Чтобы закончить параллельную цепь, Вы соединяете строку с элементами I0.0 и I0.2 (E0.0) со строкой с I0.3 (E0.3) позади I0.2 (E0.2). Для этого, выберите вертикальную линию.

6. Дважды щелкните на элемент, позади которого Вы желаете вставить вертикальную линию. В нашем примере это элемент I0.2 (E0.2).

Ваша программа показана на предыдущей странице. Сохраните программу и загрузите ее в контроллер. Проверьте функцию.

3-я модификация программы: таймер задержки

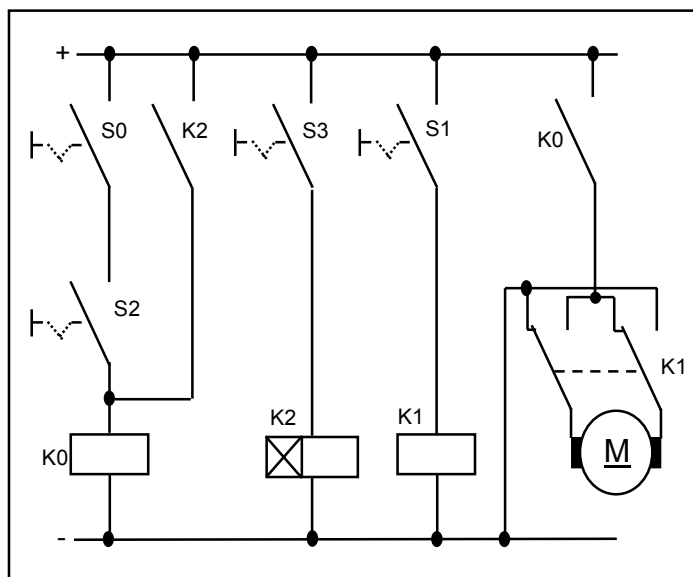
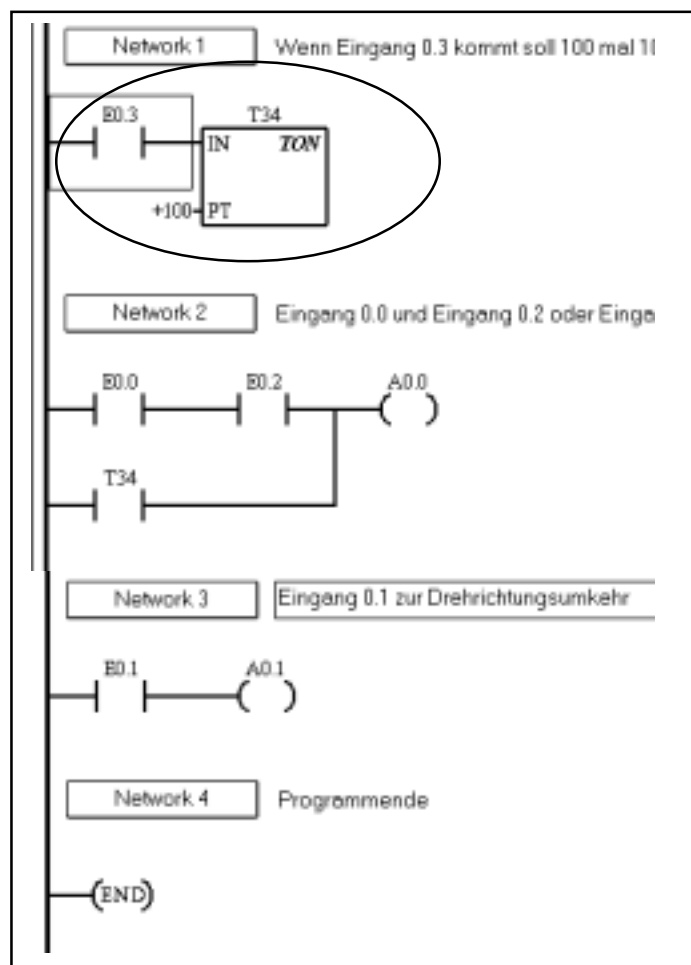


Схема без PLC

Электрических соединений для дополнительной функции таймера не требуется.

Все необходимые переключатели уже соединены. Задержка времени используется как функция контроллера при минимальном объеме программирования.

Следующая страница знакомит с функцией таймера задержки для S7-200, которая будет затем включена для изменения программы.



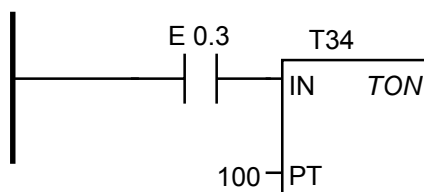
LAD- программа схемы

Задача:

В следующем изменении программы должен быть вставлен таймер задержки.

Когда вход I0.3 (E0.3) -S3 на симуляторе активирован, запускается время задержки. Выход 0.0 и, следовательно, мотор не активируется до тех пор, пока время ожидания не истечет.

Функция таймера задержки



Перед тем как начнется отсчет времени, таймер должен быть запущен. В нашем примере это будет сделано сигналом I0.3 (E0.3), поступающим на вход IN

Заданное значение в 1s записывается числом 100 у входа PT. Значение времени вычисляется через фактор PT (здесь 100) и дискретность отсчета. (Дискретность T34 10 ms , $100 \times 10 \text{ ms} = 1\text{s}$).

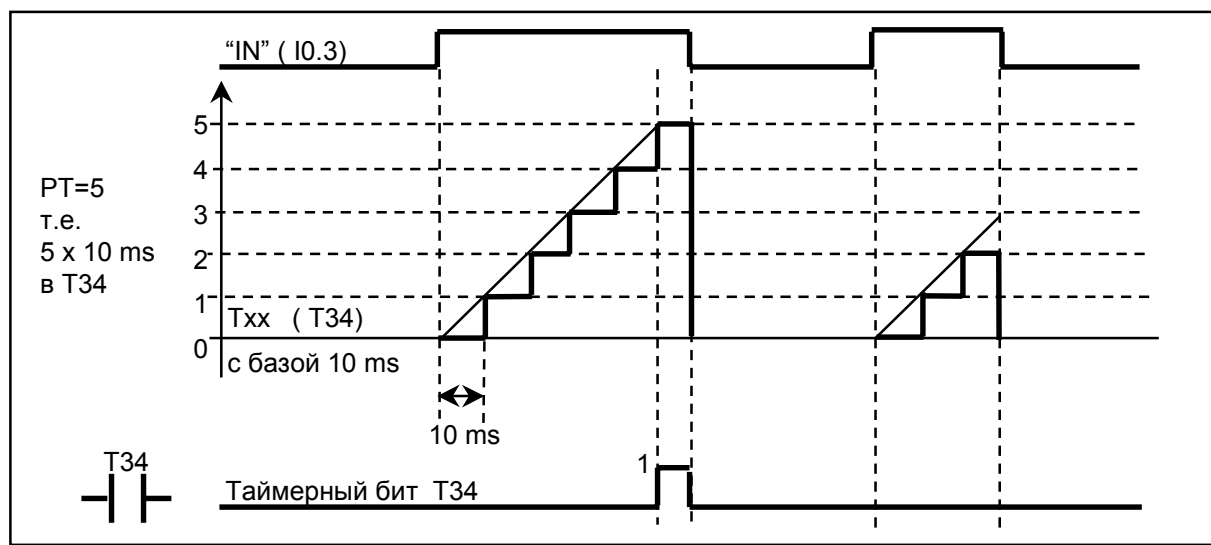
Вам требуется задержка в 1 s. Контроллер S7-200 (CPU 212) имеет 64 таймера. Таймеры обозначаются с T0 по T63. В этом примере используем таймер T34.

База времени	Txx
1 ms	T32
10 ms	T33-T36
100 ms	T37-T63

Для CPU 212 установлена дискретность времени для таймеров в 1 ms, 10 ms и 100 ms. Один таймер не может быть использован дважды.

Каждый таймер в S7-200 имеет бит статуса (время идет/ не идет). Этот бит называется таймерным битом. для таймера T34 он обозначается T34.

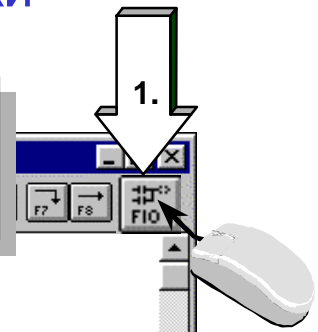
Вход I0.3 (E0.3) должен 1 s иметь значение "1", после того как бит T34 установится в "1". Если сигнал на входе "IN" сбрасывается раньше, чем выполнится задержка времени, таймерный бит не устанавливается (см. временную диаграмму ниже). Вы можете использовать таймерный бит подобно нормально открытому контакту.



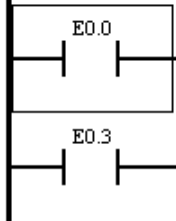
Временна диаграмма таймера для значения 50 ms

Программирование таймера задержки

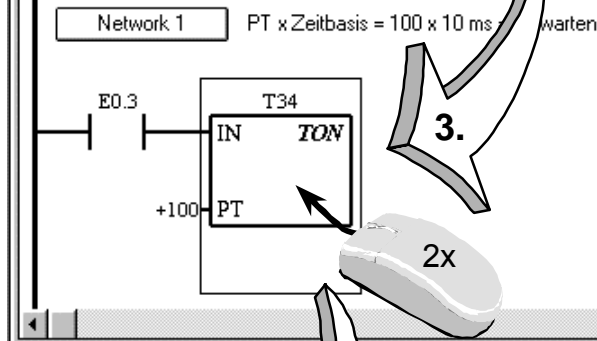
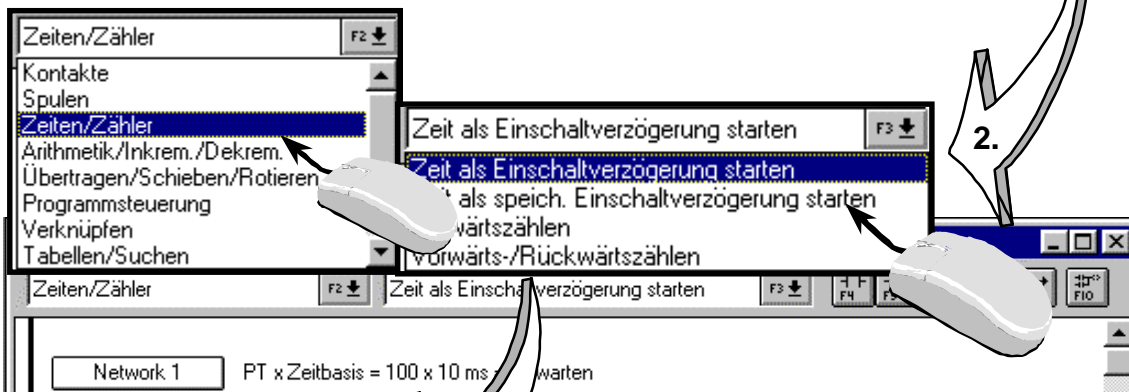
1. Чтобы оценить работу таймера, установленного в операции логическое "ИЛИ", вы должны сам таймер вставить в предыдущую цепочку. Для вставки цепочки отметьте элемент в первой цепочке и щелкните мышкой на кнопку



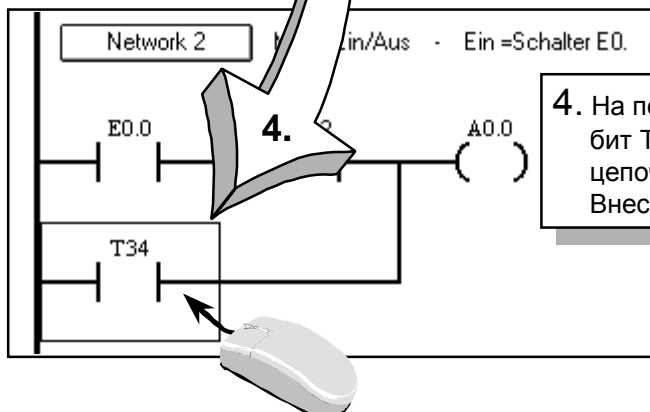
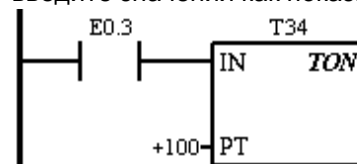
Network 1 Motor Ein/Aus - Ein =Schalter E0.0 ein, Aus = Schalter E0.0 aus



2. После того, как Вы вставите в первую позицию цепочки 1 нормально-открытый контакт для входа I0.3 (E0.3), вставьте таймер задержки во вторую позицию. Для этого клавишей F2 и мышкой выберите строку Timers/Counters (Zeiten/Zdhler), а затем F3 и мышью кликните в строку Timer-On Delay (Zeit als Einschaltverzögerung starten).



3. Вставка таймера происходит, если в отмеченной позиции дважды щелкнуть мышью. Выделите мышью каждый операнд и введите значения как показано здесь.



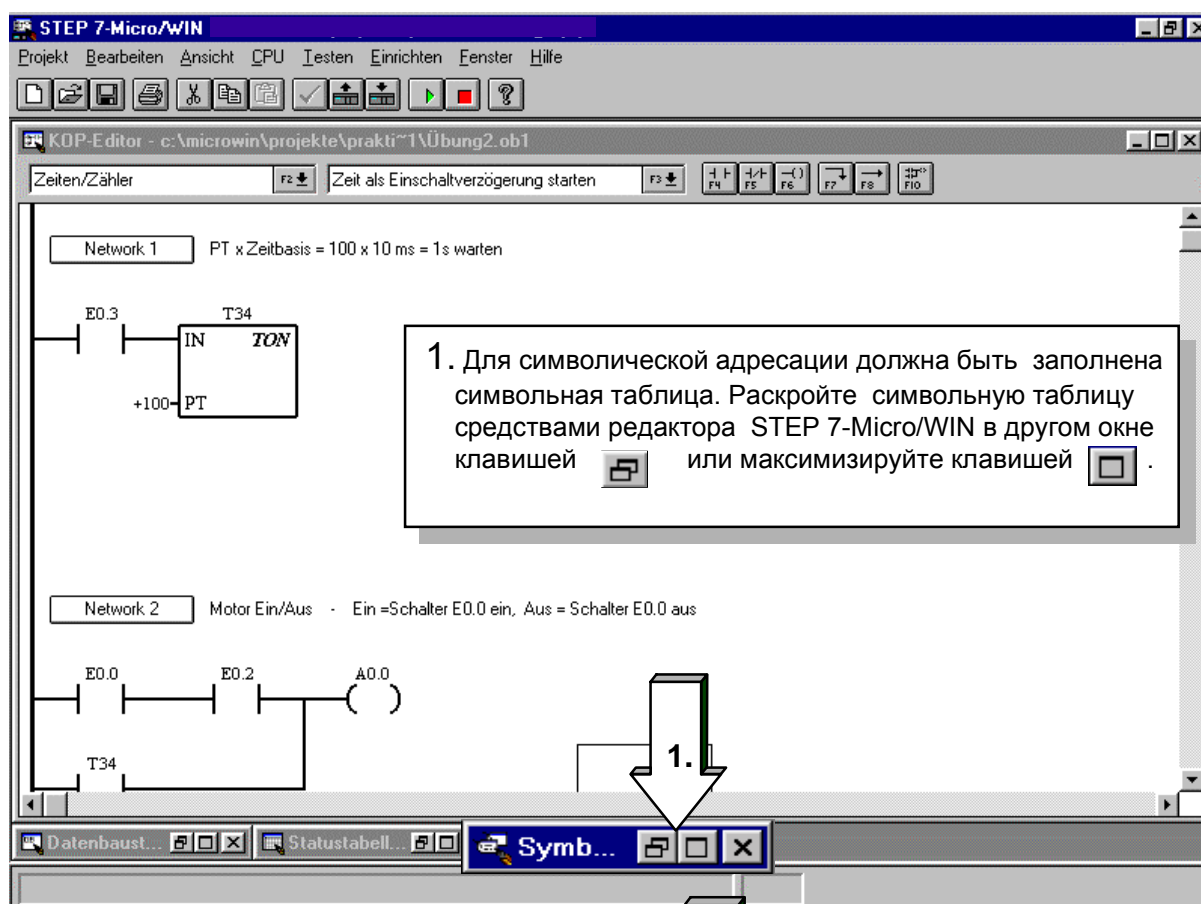
4. На последнем шаге таймерный бит T34 должен появиться в цепочке 2 в позиции I0.3 (E0.3). Внесите изменения.

Не забудьте сохранить Вашу программу. Передайте ее в контроллер и протестируйте эту функцию.



“Разговариваем просто (1)”

Пока что Вы работали с PLC-программами с операндами на PLC- языке, такими как I0.3 или T34. Но, если программа большая, то не просто читать эти операнды. Было бы очень хорошо, если бы Вы смогли применять ключ определений или другой понятный текст. Точно это означает, что Вы могли бы программировать с символической адресацией.



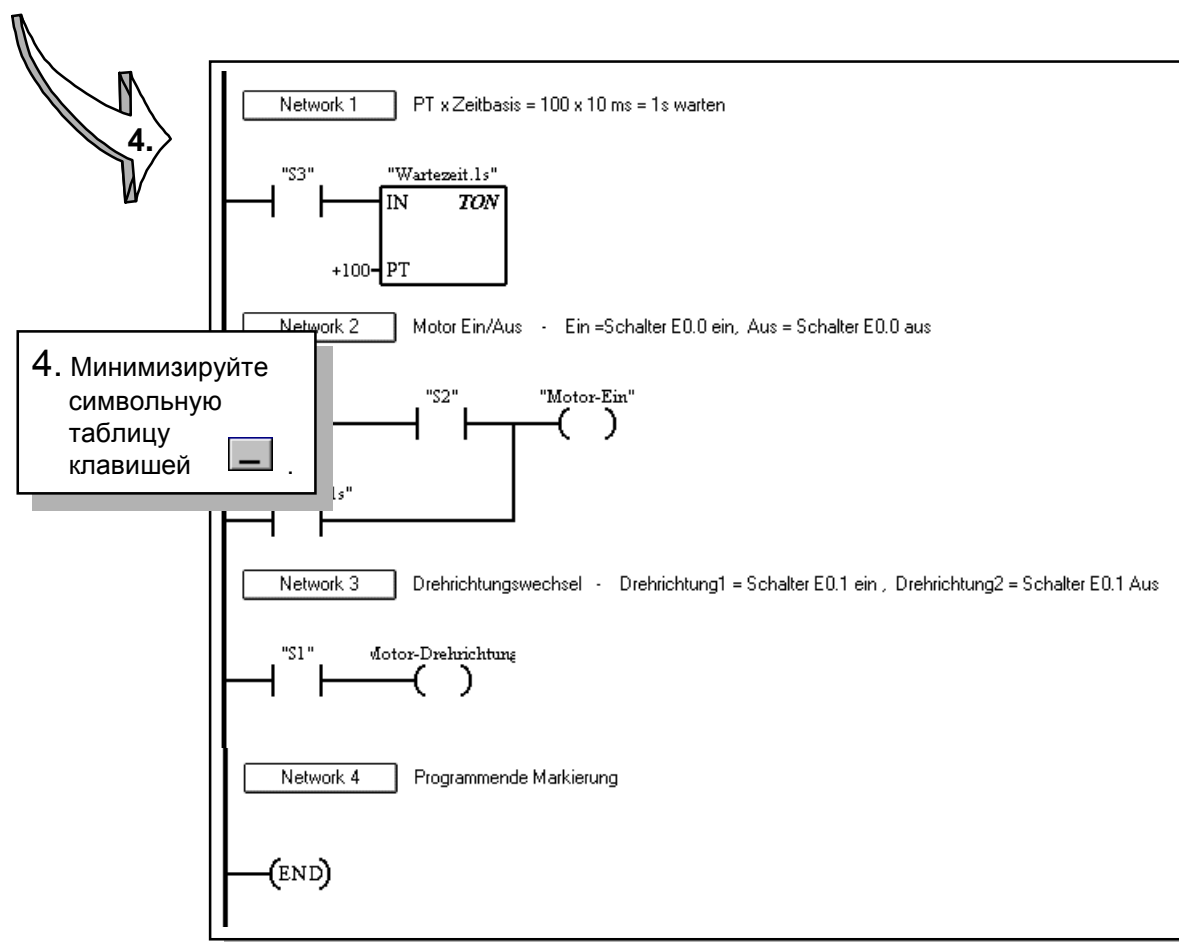
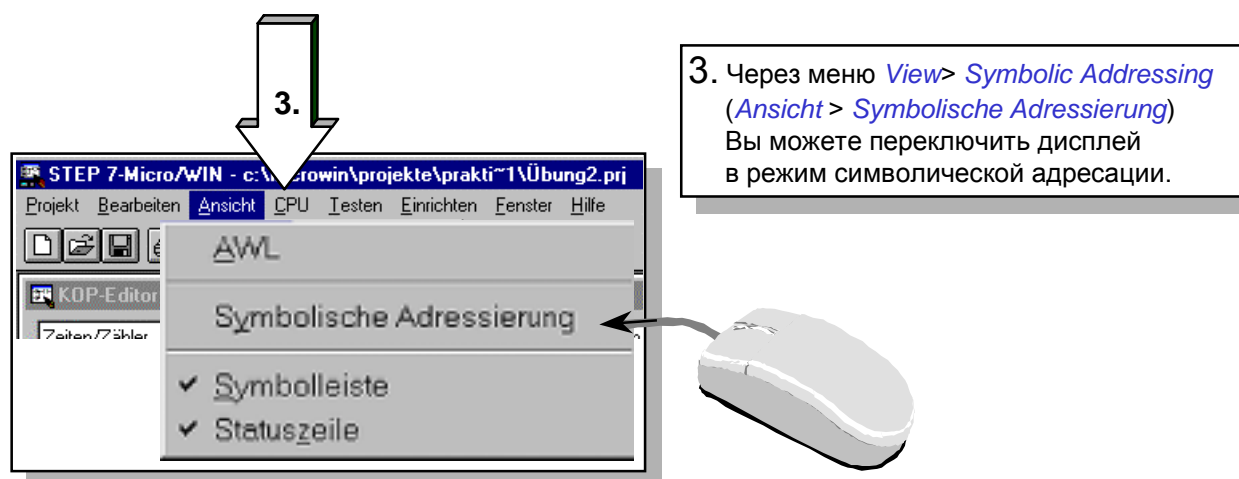
2.

Symbolischer Name	Adresse	Kommentar
S0	E0.0	Schalter 0
S1	E0.1	Schalter 1
S2	E0.2	Schalter 2
S3	E0.3	Schalter 3
Motor-Ein	A0.0	Schaltet den Motor ein
Drehrichtung	A0.1	Wechselt die Drehrichtung des Motors
Wartezeit, 1s	T34	Wartet 100 * 10ms = 1 Sekunde bis einschalten

2. В появившемся окне Вы можете редактировать символьную таблицу. Под строкой “Symbolischer Name” введите элемент, который Вы позже появится как понятный текст. Под адресом (Adresse) введите операнд, который Вы желаете заменить на символическое имя. Под комментарием (Kommentar) вы можете ввести подсказку, которую Вы найдете полезной. Не забудьте сохранить свою работу.

Продолжение на след. стр.

“Разговариваем просто (2)”



5. Рисунок выше показывает LAD-диаграмму Вашей программы с символической адресацией. Конечно, Вы можете также программировать с символической адресацией. Например, в качестве операнда Вы можете ввести “S3”, но только Вы должны быть уверены, что этот операнд имеется в символьной таблице.

Создание нового проекта (1)

Если Вы желаете написать новую свою программу, Вам необходимо иметь контейнер, чтобы “положить” туда файл программы и связанные с ним файлы. В STEP 7-Micro/WIN этот контейнер называется проектом.

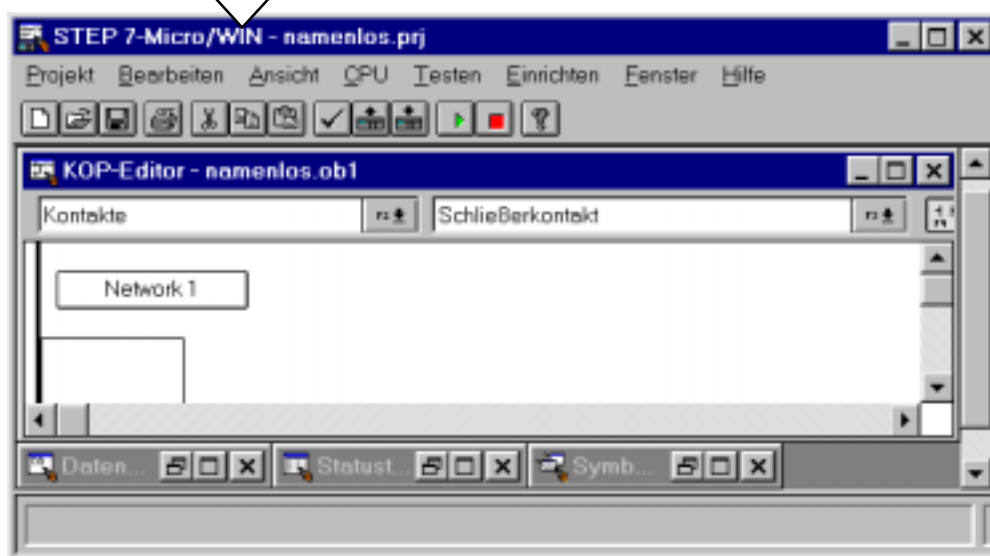
S7-200-проект содержит программный файл, например, *.OB1 и файл для символов *.sym.

S7-200- проект является также приемником дополнительных файлов, связанных с программой. Итак, первым делом Вам нужен новый проект.

1. Создайте новый проект, используя меню *Projekt > Neu*.



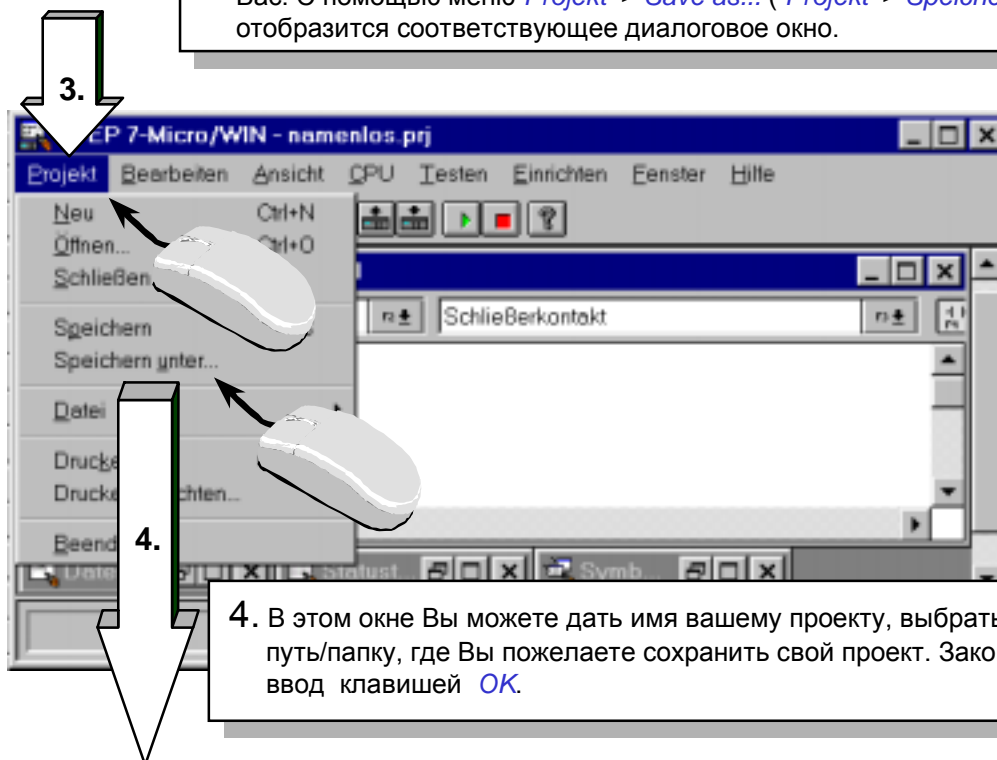
2. Результатом является проект с именем unnamed.prj (namenlos.prj) и открытый блок с именем unnamed.ob1 (namenlos.ob1). В нем Вы можете начать программировать.



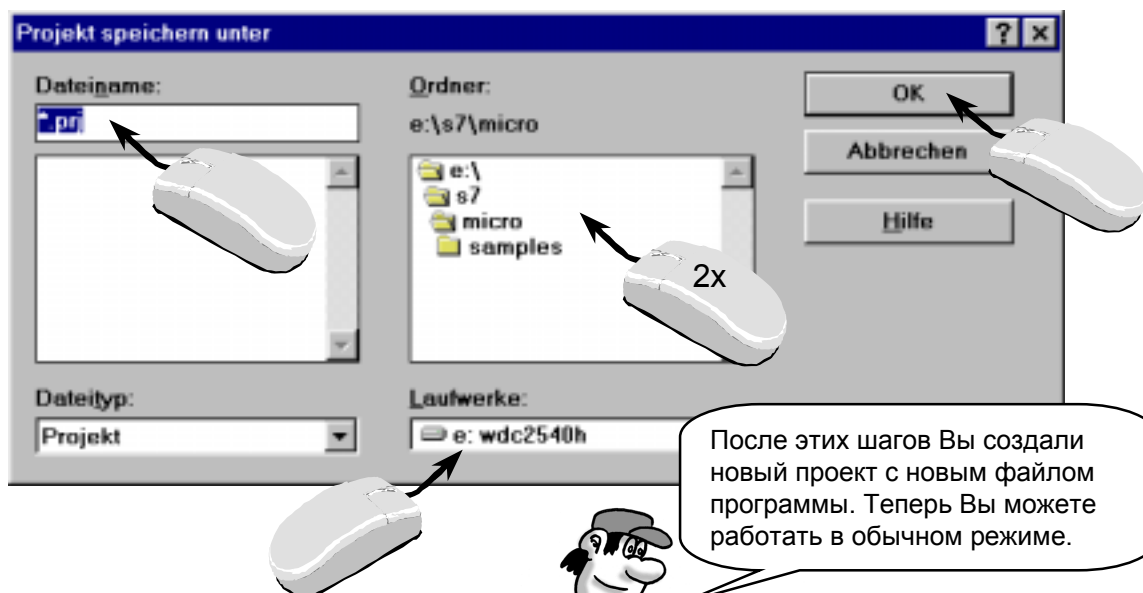
См. следующую страницу для продолжения.

Создание нового проекта (2)

3. Сохраните проект Namenlos.prj сразу, после или во время программирования под именем, которое что-нибудь означает для Вас. С помощью меню *Projekt* > *Save as...* (*Projekt* > *Speichern unter...*) отобразится соответствующее диалоговое окно.

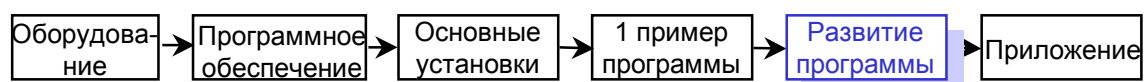


4. В этом окне Вы можете дать имя вашему проекту, выбрать диск и путь/папку, где Вы пожелаете сохранить свой проект. Закончите ввод клавишей *OK*.

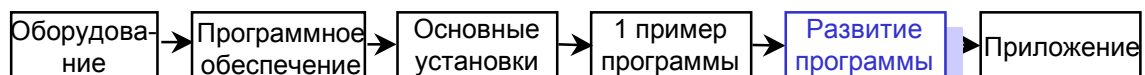


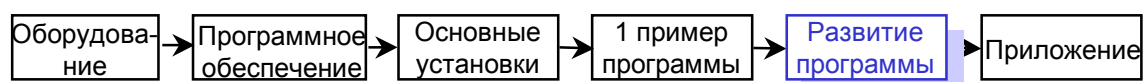
После этих шагов Вы создали новый проект с новым файлом программы. Теперь Вы можете работать в обычном режиме.





Поздравляю, теперь Вы знаете, как программировать микроконтроллер S7-200. Если Вы желаете узнать больше, то Вы найдете полезную информацию в приложении.





Вы желаете узнать больше ?

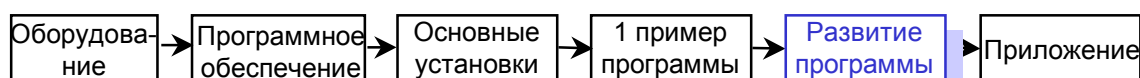
Дополнительные примеры Вы найдете в каталоге “Samples” в вашем каталоге для STEP 7-Micro/WIN или в сборнике “Tips & Tricks” (советы и приемы) для S7-200. Вы можете получить “Tips & Tricks” от наших представителей в России.

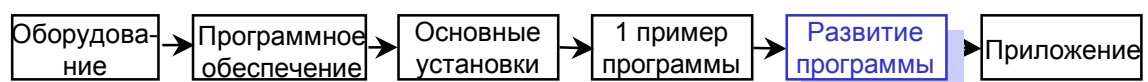
Еще больше информации Вы найдете в руководстве для S7-200. Дальнейшее обучение Вы можете пройти на курсах по S7-200 в учебных центрах Siemens в России.

Остались вопросы?
Технические проблемы?
Наши представители
будут рады помочь Вам.



Пожалуйста, свяжитесь с нашим представителем, от которого Вы получили стартовый набор. Если Вы не можете с ним связаться, пожалуйста позвоните в представительство Siemens в Москве: (095) 237 56 43 или в Германию на нашу линию SIMATIC-Hotline по тел. +49 0911 895-7000.





Приложение



О битах, байтах и словах


Наименьшая единица информации в цифровых системах известна как бит. Бит может иметь состояние "0" (т.е. фальш или недействительно) или "1" (т.е. истина).

Например, выключатель света имеет только два состояния: свет включен или свет выключен. Выключатель света, таким образом, имеет объем информации в один бит.

Состояние выключатель включен при неисправной лампе не рассматривается в этом примере.

В контроллере биты организованы в группы. Группа в 8 бит называется байтом. Каждый бит в группе расположен в определенном месте в точном соответствии с его адресом. Байт имеет байтовый адрес и битовые адреса с 0 по 7. Группа из двух байт называется словом.

Числа в контроллере представлены в двоичной системе, т.е. они имеют основание 2.

1 бит = 2^0  Если эта позиция "0" (= фальш или не истина...), он имеет значение 0, т.к. $0 * 2^0 = 0$ (содержимое * вес разряда бита)
Если эта позиция "1" (= истина), он имеет значение 1, т.к. $1 * 2^0 = 1$

В байте это бит с позицией (весом разряда) 2^0 !

1 байт =

--	--	--	--	--	--	--	--

Каждая позиция бита может иметь значение "1" или "0". Биты могут использоваться индивидуально (как ключи), так и в группе, представляя число между 0 и 255.

$(2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 255,$
 $2^0=1, 2^1=2, 2^2=4, 2^3=8,$
 $2^4=16, 2^5=32, 2^6=64, 2^7=128).$

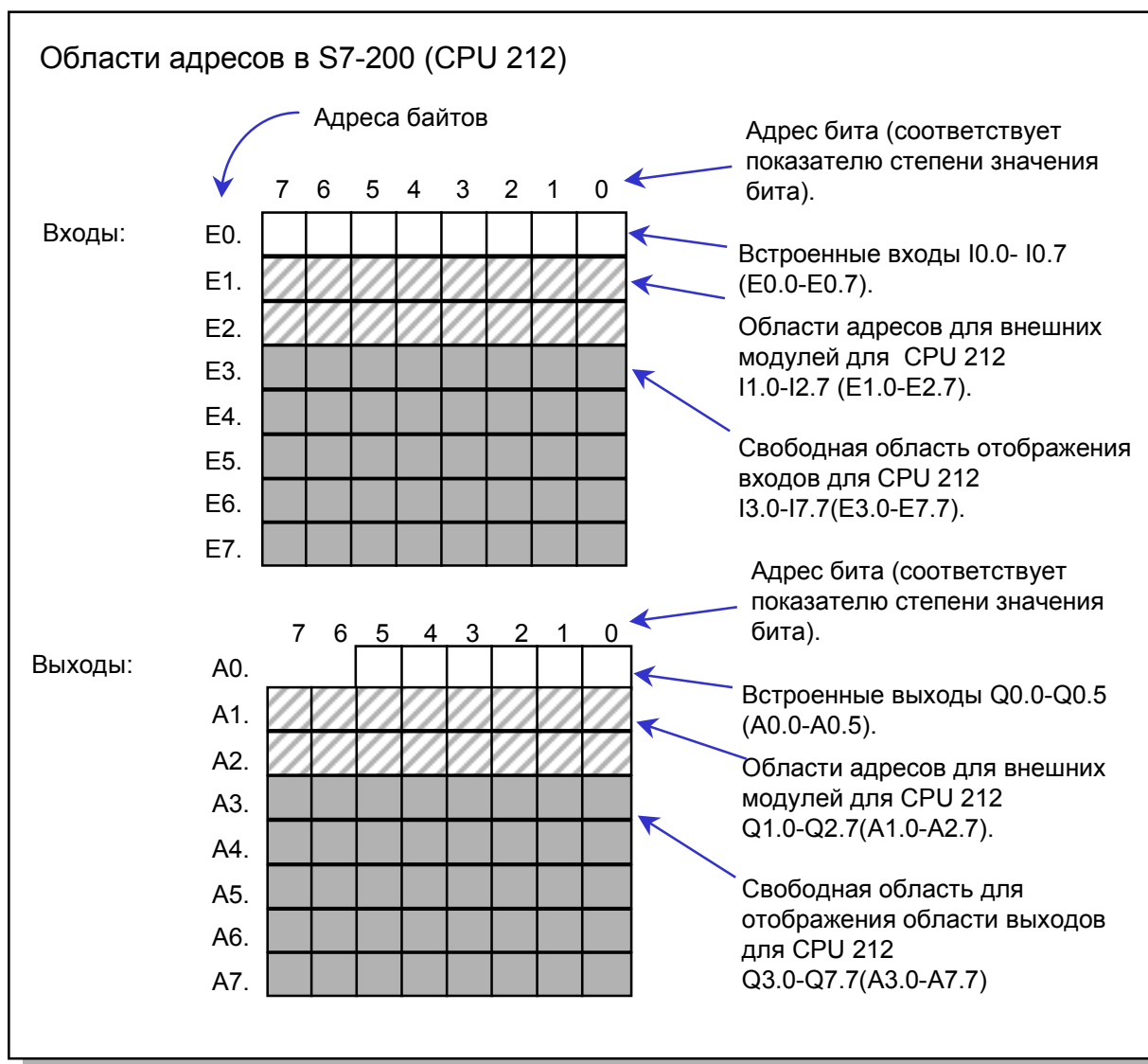
Максимальное значение байта в слове 255 !

1 слово =

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Слово в контроллере представляется числом от -32768 до +32767. Бит со значением 2^{15} используется для обозначения отрицательного числа (когда разряд 2^{15} равен "1", то число отрицательно).

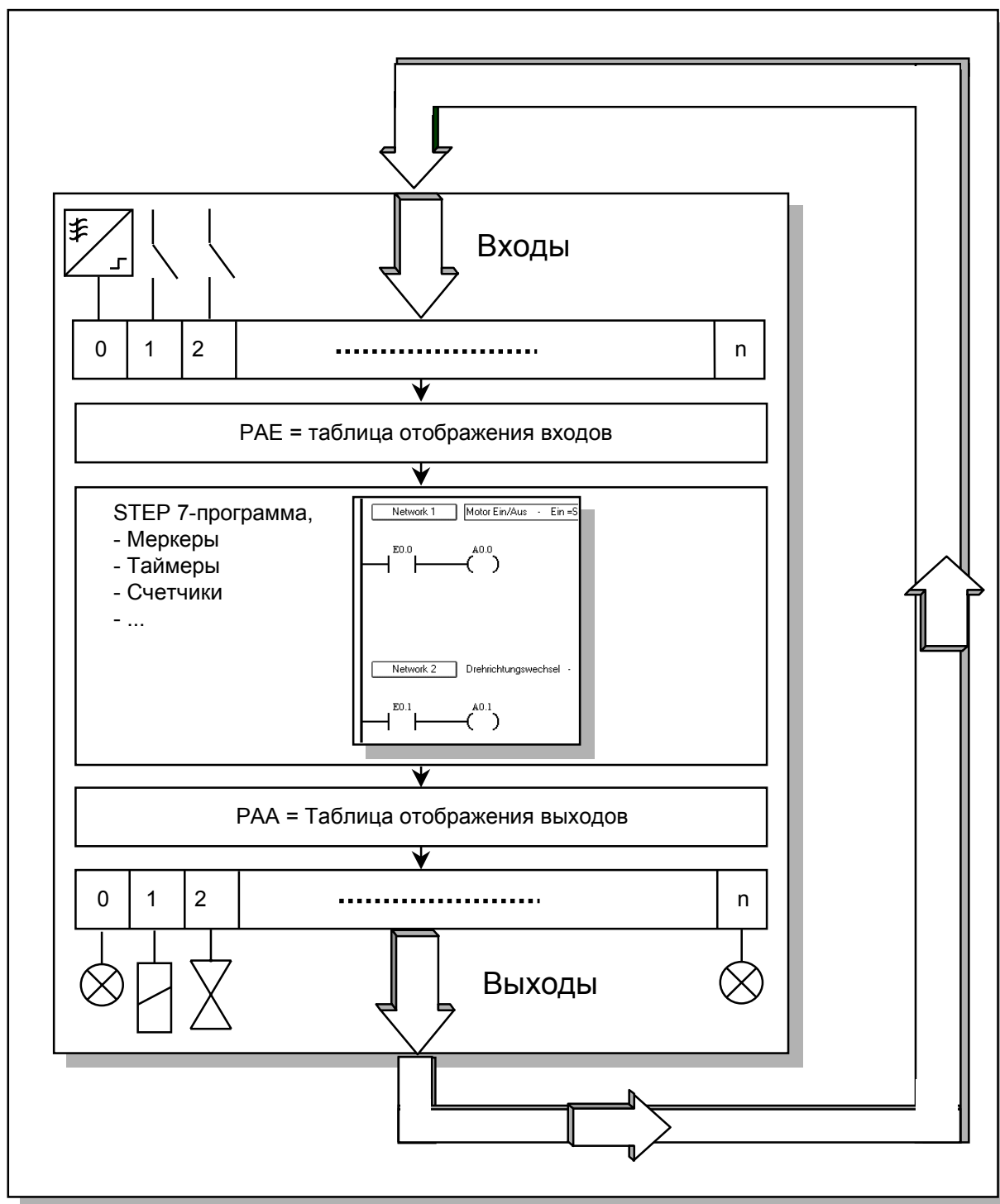
Адресные области в S7-200



Адресные области в контроллере являются областями памяти, в которых отображены входы и выходы процесса. Например, если напряжение, приложенное к входу равно "1", то этот сигнал отображается в ячейку памяти, связанную с этим входом.

Области памяти, формируемые по внешним сигналам (входы и выходы) называются областью отображения входов (РАЕ) и областью отображения выходов (РАА). Эти области памяти содержат образ сигналов от процесса.

Выполнение циклической программы в S7-200



Все контроллеры SIMATIC работают циклически. В течении каждого цикла сначала читаются входы и сохраняются в таблице отображения входов.

Управляющая программа работает на основе этой информации. Таблица отображения выходов (РАА) формируется на основе логики программы.
На последнем шаге состояния выходов из таблицы РАА передаются на физические выходы.

Затем цикл повторяется.