

# Ввод примера программы

# 4

Данное руководство описывает версию 2.0 программного обеспечения STEP 7–Micro/WIN. Предыдущие версии этого программного обеспечения функционируют несколько иначе.

Данная глава объясняет, как с помощью программного обеспечения STEP 7–Micro/WIN выполнить следующие задачи:

- Ввод примера программы для резервуара для смешивания с двумя питающими насосами.
- Создание таблицы символов, таблицы состояний/принудительного задания и блока данных.
- Контроль программы.

STEP 7–Micro/WIN предоставляет Вам обширную Online–помощь. Для того, чтобы получить помощь по текущим вопросам, выберите один из пунктов меню **Hilfe [Помощь]** или нажмите **F1**.

## Обзор главы

Раздел	Описание	Страница
4.1	Создание программы для прикладного примера	4–2
4.2	Создание проекта	4–6
4.3	Создание таблицы символов	4–8
4.4	Ввод программы в форме контактного плана	4–10
4.5	Создание таблицы состояний/принудительного задания	4–14
4.6	Загрузка и контроль примера программы	4–15

## 4.1 Создание программы для прикладного примера

### Предпосылки к выполнению примера программы

Если Вы полностью создадите и загрузите в CPU пример программы из этой главы, то Вы можете обрабатывать эту программу с помощью Вашего S7-200. На рис. 4-1 показаны компоненты, которые требуются для выполнения и контроля примера программы:

- кабель PC/PPI или плата MPI и кабель RS-485 для подключения Вашего компьютера к CPU S7-200
- CPU S7-200
- Имитатор входных сигналов или тумблеры либо нажимные переключатели (8).
- Сетевой кабель и источник питающего напряжения.
- STEP 7-Micro/WIN версии 2.0

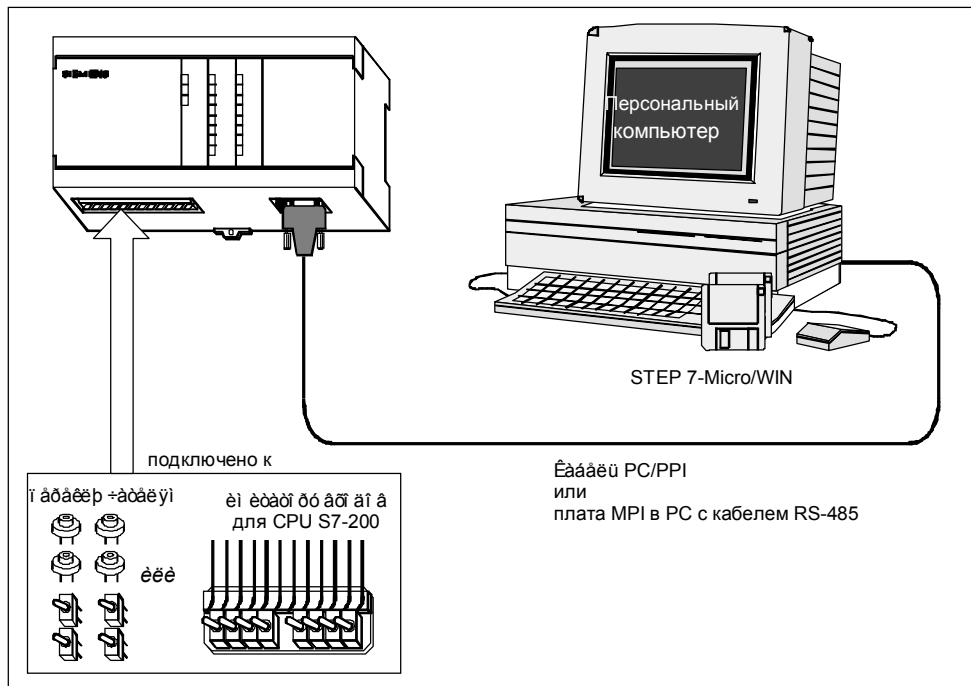


Рис. 4-1. Предпосылки для выполнения примера программы

### Прикладной пример "Резервуар для смешивания"

На рис. 4–2 показана схема резервуара для смешивания. Резервуар для смешивания может применяться в различных приложениях, например, для смешивания красок. В данном приложении две питающие трубы, подводящие разные вещества, производят подачу сверху внутрь резервуара. Через отдельную трубу в нижней части резервуара вытекает готовая смесь красок. С помощью примера программы можно управлять процессом наполнения, контролировать уровень наполнения и управлять циклами нагревания и смешивания. Процесс описывается следующим образом:

- Шаг 1: Наполнить резервуар с краской 1
- Шаг 2: Наполнить резервуар с краской 2
- Шаг 3: Закончить подачу, если замкнут переключатель "Резервуар полный"
- Шаг 4: Оставить насос включенным, если пусковой переключатель открыт
- Шаг 5: Начать цикл нагревания и смешивания
- Шаг 6: Включить двигатель смесителя и вентиль пара
- Шаг 7: Выпустить краску из резервуара
- Шаг 8: Подсчитывать циклы

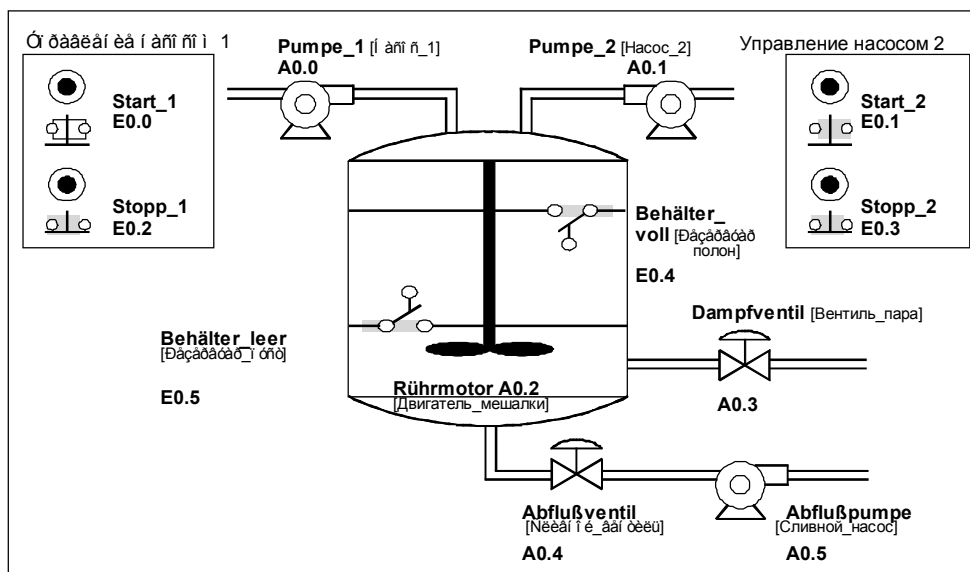


Рис. 4-2. Пример программирования: Резервуар для смешивания

### Пример программы в виде AWL и KOP

Пример программы можно вводить как в виде списка команд (AWL), так и в виде контактного плана (KOP). В табл. 4–1 показан пример программы в виде списка команд, а на рис. 4–3 этот пример показан в виде контактного плана. Разделы 4.2–4.4 проведут Вас шаг за шагом через необходимые задачи к вводу программы.

Таблица 4–1. Пример программы в виде списка команд

AWL	Описание
NETWORK	//Наполнить резервуар с краской 1.
LD "Start_1"	
O "Pumpe_1"	
U "Stopp_1"	
UN "Behälter_voll"	
= "Pumpe_1"	
NETWORK	//Наполнить резервуар с краской 2.
LD "Start_2"	
O "Pumpe_2"	
U "Stopp_2"	
UN "Behälter_voll"	
= "Pumpe_2"	
NETWORK	//Установить меркер, если достигнут
LD "Behälter_voll"	//максимальный уровень наполнения.
S "Max_Füllstand", 1	
NETWORK	//Запустить таймер, если достигнут
LD "Max_Füllstand"	//максимальный уровень наполнения.
TON "Mischzeit", +100	
NETWORK	//Включить двигатель смесителя.
LDN "Mischzeit"	
U "Max_Füllstand"	
= "Rührmotor"	
= "Dampfventil"	//Дать стечь смеси красок.
NETWORK	
LD "Mischzeit"	
UN "Behälter_leer"	
= "Abflußventil"	
= "Abflußpumpe"	//Подсчитывать каждый цикл.
NETWORK	
LD "Behälter_leer"	
U "Mischzeit"	
LD "Rücksetzen"	
ZV "Zyklusähler", +12	//Сбросить меркер, если достигнут минимальный уровень
	наполнения или истекло время.
NETWORK	
LD "Behälter_leer"	
U "Mischzeit"	
R "Max_Füllstand", 1	//Закончить главную программу.
NETWORK	
MEND	

Пояснения к символическим переменным, использованным в программе:  
 Max\_Füllstand - максимальный уровень наполнения; Mischzeit - время смешивания; Rücksetzen - сброс; Zyklusähler - счетчик циклов; смысл остальных переменных описан на рис. 4-2.

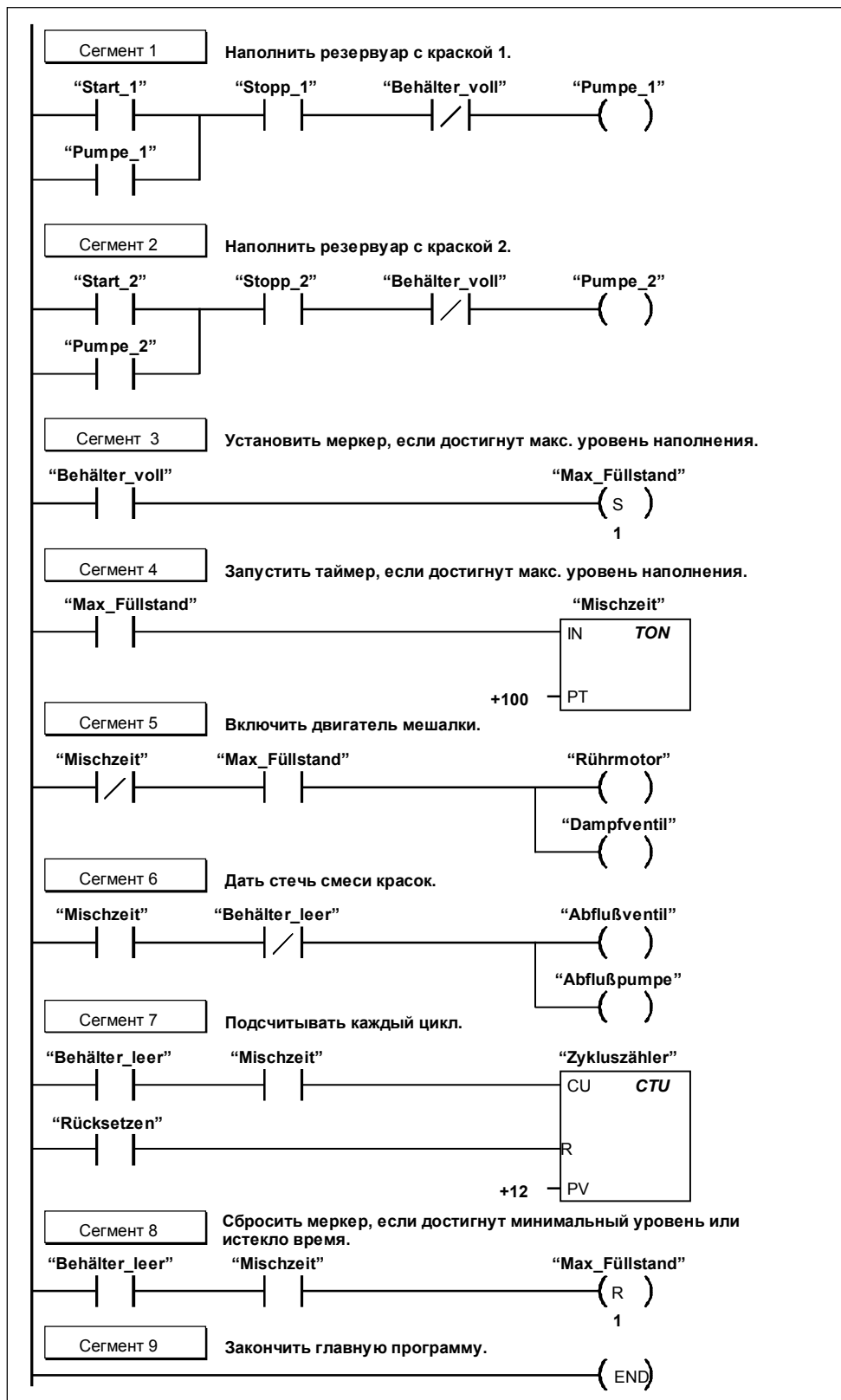


Рис. 4-3. Пример программы в виде контактного плана

## 4.2 Создание проекта

### Создание нового проекта

При создании или открытии проекта STEP 7–Micro/WIN вызывает следующие редакторы: редактор AWL или KOP (OB1) в зависимости от предварительной установки, редактор блоков данных (DB1), редактор таблицы состояний/ принудительного задания и редактор таблиц символов.

Для создания нового проекта выберите команду меню **Projekt** → **Neu [Проект → Новый]** (см. рис. 4–4)

либо щелкните мышью на следующей кнопке на панели инструментов:  .

В ответ на это открывается диалоговое окно, в котором Вы можете выбрать свой CPU из раскрывающегося спискового блока.

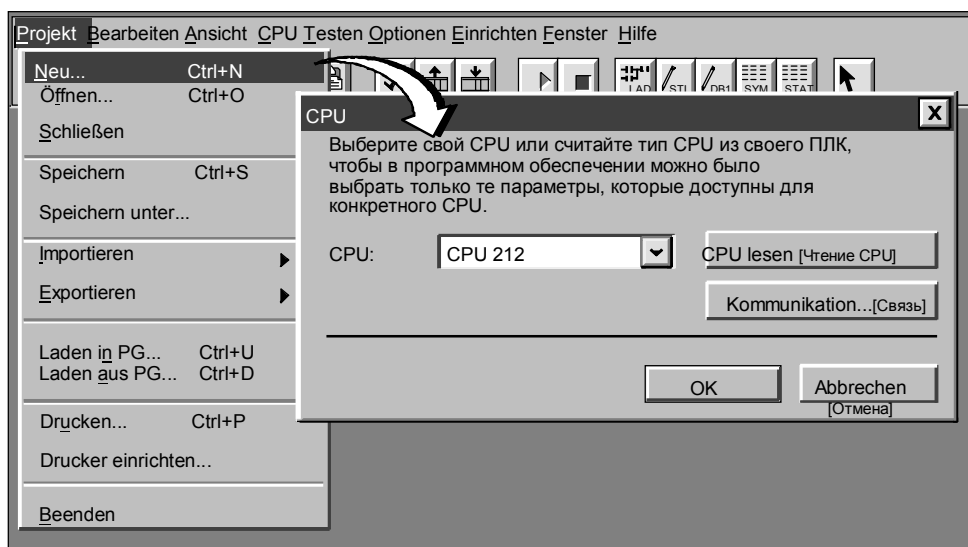


Рис. 4-4. Создание нового проекта и выбор CPU

Пояснения к меню Projekt на рис. 4-4: Neu - Новый; Öffnen - Открыть; Schließen - Закреть; Speichern - Сохранить; Speichern unter - Сохранить как; Importieren - Импортировать; Exportieren - Экспортировать; Laden in PG - Загрузить в PG; Laden aus PG - Загрузить из PG; Drucken - Печатать; Drucker einrichten - Настройка принтера; Beenden - Конец.

### Присвоение имени примеру программы

Вы можете дать имя Вашему проекту в любой момент времени. В данном примере для того, чтобы сохранить проект под определенным именем, действуйте следующим образом □ см. рис. 4–5):

1. Выберите команду меню **Projekt** → **Speichern unter** [Проект → Сохранить как].
2. В поле "Dateiname" ["Имя файла"] введите следующее имя: **projekt1.prj**
3. Щелкните мышью по "OK".

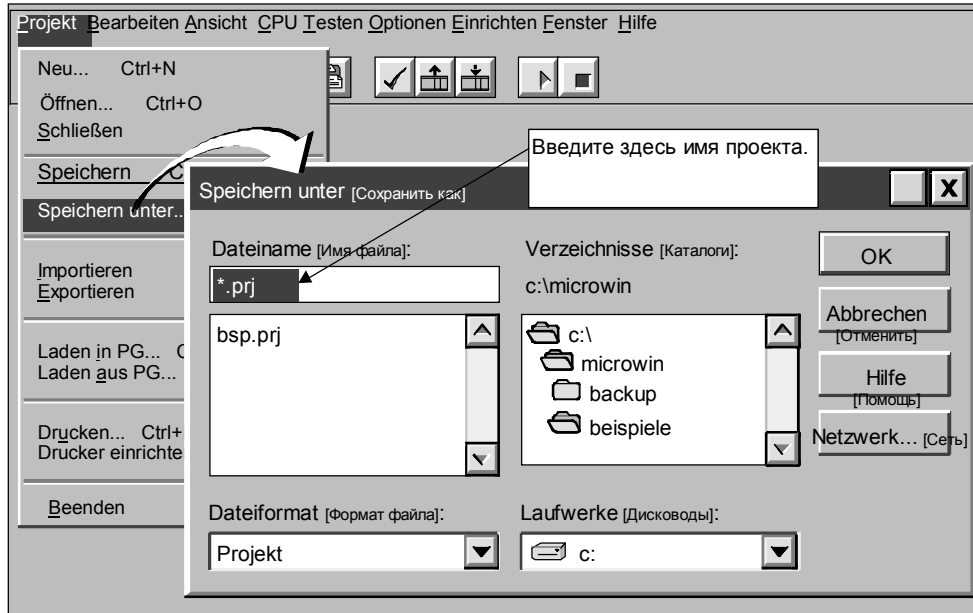


Рис. 4-5. Присвоение имени примеру проекта

## 4.3 Создание таблицы символов


### Вызов редактора таблиц символов

Если Вы хотите поставить в соответствие абсолютным адресам в примере программы символические имена, откройте редактор таблиц символов. Для этого дважды щелкните мышью на пиктограмме редактора или на кнопке “Восстановить/ максимизировать” на этой пиктограмме (в Windows 95). Вы также можете выбрать команду меню **Ansicht** → **Symboltabelle...** [**Вид** → **Таблица символов**].

### Ввод символических имен

На рис. 4–6 показан список абсолютных адресов и соответствующих им символических имен для примера программы. Для ввода символических имен действуйте следующим образом:

1. Выделите первое поле в столбце “Symbolischer Name” [“Символическое имя”] и введите **Start\_1**.
2. Нажмите клавишу ввода, чтобы выделить первое поле в столбце “Adresse” [“Адрес”]. Введите адрес **E0.0** и нажмите клавишу ввода. В ответ на это выделяется первое поле в столбце “Kommentar” [“Комментарий”]. (Комментарии являются необязательными, однако они могут быть очень полезными для того, чтобы документировать отдельные элементы программы.)
3. Нажмите клавишу ввода, чтобы начать ввод в следующей строке. Повторяйте приведенные выше шаги до тех пор, пока Вы не введете все остальные адреса и символические имена.
4. Сохраните таблицу символов с помощью команды меню **Projekt** → **Speichern** [**Проект** → **Сохранить**].



Symbolischer Name	Adresse	Kommentar
Start_1	E0.0	Инициализация таймера 1
Start_2	E0.1	Пусковая кнопка для краски 2
Stopp_1	E0.2	Кнопка СТОП для краски 1
Stopp_2	E0.3	Кнопка СТОП для краски 2
Behälter_voll	E0.4	Выключатель-ограничитель максим. уровня
Behälter_leer	E0.5	Выключатель-ограничитель миним. уровня
Rücksetzen	E0.7	Управление сбросом счетчика
Pumpe_1	A0.0	Насос для краски 1
Pumpe_2	A0.1	Насос для краски 2
Rührmotor	A0.2	Двигатель смесителя
Dampfventil	A0.3	Пар для нагрева смеси красок в резервуаре
Abflußventil	A0.4	Вентиль для слива смеси красок
Abflußpumpe	A0.5	Насос для откачки смеси красок
Max_Füllstand	M0.1	Меркер
Mischzeit	T37	Таймер для управл. процес. смеш. и нагрева
Zykluszähler	Z30	Считает кол-во циклов смешив. и нагрева

Рис. 4-6. Таблица символов для примера программы

### Программирование с применением символических адресов

Прежде чем вводить программу, установите символическую адресацию. Для этого выберите команду меню **Ansicht** → **Symbolische Adressierung** [**Вид** → **Символическая адресация**]. Если слева от пункта меню находится галочка, то символическая адресация активизирована.

#### Указание

Символические имена чувствительны к регистру символов. Имя, которое Вы вводите, должно по используемым большим и малым буквам в точности соответствовать имени в таблице символов. В случае расхождений курсор остается на элементе, а у нижнего края панели управления в статусной строке отображается сообщение “Ungültiger Parameter” [“Недействительный параметр”].





## 4.4 Ввод программы в форме контактного плана

### Вызов редактора KOP

Для вызова редактора KOP дважды щелкните мышью на его пиктограмме у нижнего края панели управления. На рис. 4–7 показаны некоторые из важных инструментов, которые Вы будете использовать в редакторе KOP.

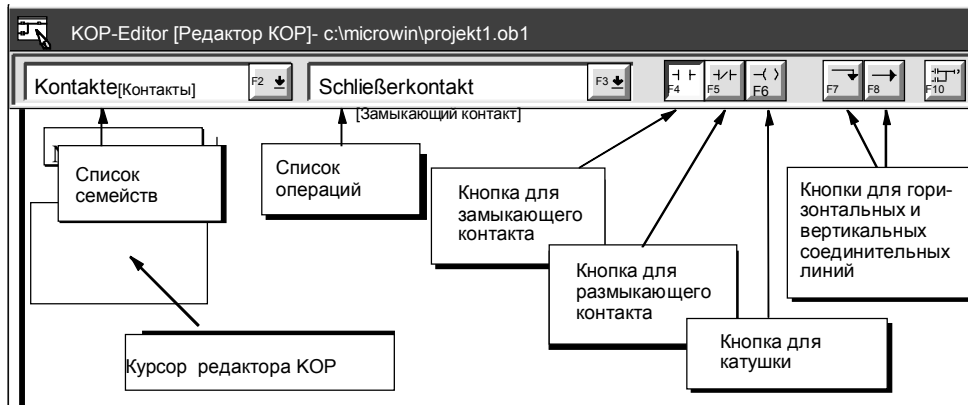


Рис. 4-7. Основные инструменты в редакторе KOP

### Ввод первого элемента сегмента

Для того, чтобы ввести первый сегмент примера программы, действуйте следующим образом:

1. Дважды щелкните мышью на пронумерованном ключевом слове Network [Сегмент] или рядом с ним, чтобы выделить поле комментария. Введите представленный на рисунке 4–8 комментарий.
2. Нажмите клавишу ввода или клавишу ↓. В ответ на это курсор KOP в следующей строке перемещается на первое поле слева.
3. Выберите замыкающий контакт, нажав функциональную клавишу **F4** или выбрав сначала семейство "Kontakte" ["Контакты"], а затем "Schließerkontakt" ["Замыкающий контакт"] в списке операций.
4. Нажмите клавишу ввода или переместите указатель мыши внутри курсора KOP и дважды щелкните мышью. В результате на экране отображается замыкающий контакт с символическим именем "Start\_1" сверху. (Каждый раз, когда Вы вводите контакт, программное обеспечение указывает E0.0 в качестве предварительно установленного адреса. В данном примере E0.0 соответствует символическому имени "Start\_1".)
5. "Start\_1" является первым элементом сегмента 1. Подтвердите ввод этого первого элемента и символического имени, нажав клавишу ввода. При этом курсор KOP переходит во второе поле этой строки.

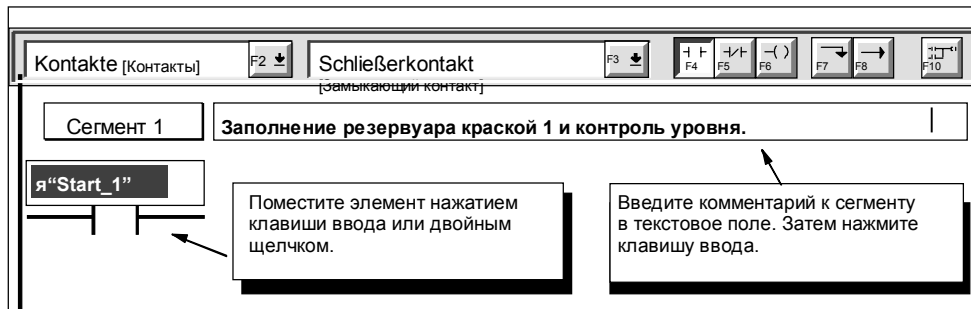


Рис. 4-8. Ввод комментария к сегменту и первого элемента в КОР

При вводе следующих контактов первого сегмента действуйте следующим образом:

1. Нажмите клавишу ввода, чтобы ввести второй элемент. В результате отображается замыкающий контакт с предварительно установленным символическим именем "Start\_1".
2. Введите **Stopp\_1** и нажмите клавишу ввода. При этом курсор перейдет в следующее поле.
3. Щелкните мышью на кнопке "Размыкающий контакт" (F5) на панели инструментов и нажмите клавишу ввода. На экране отображается размыкающий контакт. Выше контакта появляется символическое имя "Start\_1".
4. Введите **Behälter\_voll** и нажмите клавишу ввода.

Введенный Вами сегмент КОР должен выглядеть так, как показано на рисунке 4–9.

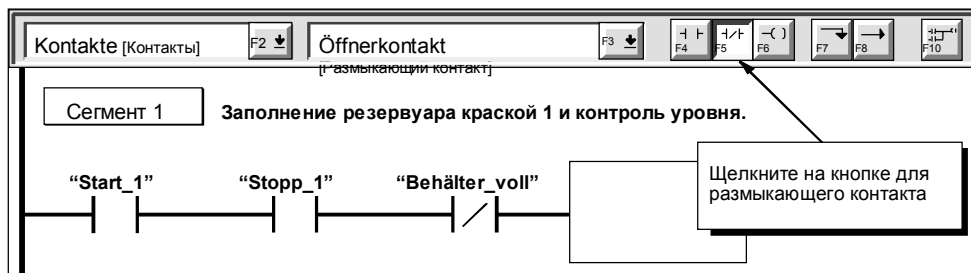


Рис. 4-9. Ввод следующих элементов КОР

Теперь курсор КОР находится справа от размыкающего контакта "Behälter\_voll" ["Резервуар\_полный"]. Для дополнения сегмента действуйте следующим образом (см. рисунок):

1. Щелкните мышью на кнопке "Катушка" (F6) и нажмите клавишу ввода или переместите указатель мыши внутрь курсора КОР и дважды щелкните мышью. В ответ на это отображается катушка с именем "Pumpe\_1" ["Насос\_1"]. (Каждый раз, когда Вы вводите катушку, в качестве предварительной установки отображается адрес A0.0. В данном примере программы A0.0 соответствует символическому имени "Pumpe\_1".)
2. Подтвердите катушку и символическое имя, нажав клавишу ввода.
3. Переместите курсор опять на первый элемент данного сегмента, используйте для этого мышью или клавишу ←.
4. Щелкните мышью на кнопке "Вертикальная соединительная линия" (F7) и нажмите клавишу ввода. В результате между первым и вторым контактом протягивается вертикальная соединительная линия.

## Ввод примера программы

- Щелкните мышью на кнопке "Замыкающий контакт" (F4) на панели инструментов и затем нажмите клавишу ввода. В ответ на это отображается контакт с именем "Start\_1".
- Введите **Pumpe\_1** и нажмите клавишу ввода.  
Теперь первый сегмент введен полностью.

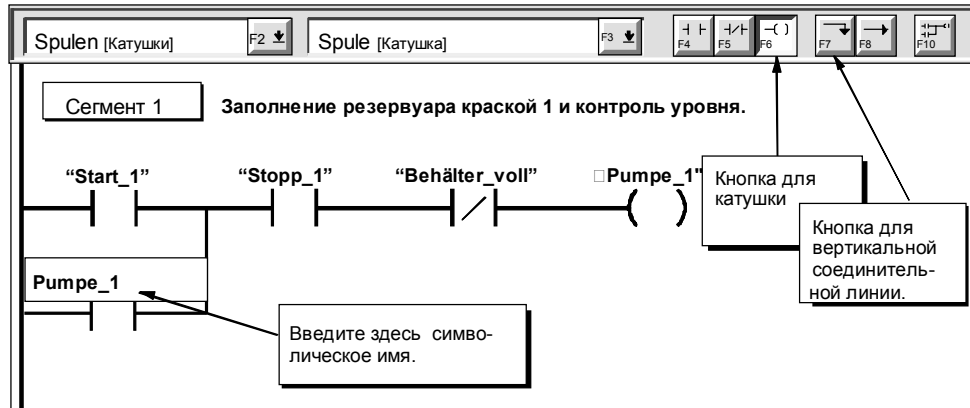


Рис. 4-10. Дополнение первой сети

## Ввод второго сегмента

Для ввода второго сегмента примера программы действуйте следующим образом:

- Переместите курсор на второй сегмент, используя для этого мышью или клавишу ↓.
- Введите комментарий, показанный на рисунке 4-11, в поле комментария сегмента. □Так как комментарий для сегмента 2 подобен комментарию сегмента 1, то Вы также можете выделить этот текст в сегменте 1, скопировать и вставить в поле комментария сегмента 2. Затем Вы должны лишь заменить номер краски с 1 на 2.)
- Повторите шаги, которые Вы выполняли при вводе сегмента 1. Используйте при этом символические имена, представленные на рисунке 4-11.
- После ввода сегмента 2 переместите курсор на сегмент 3.

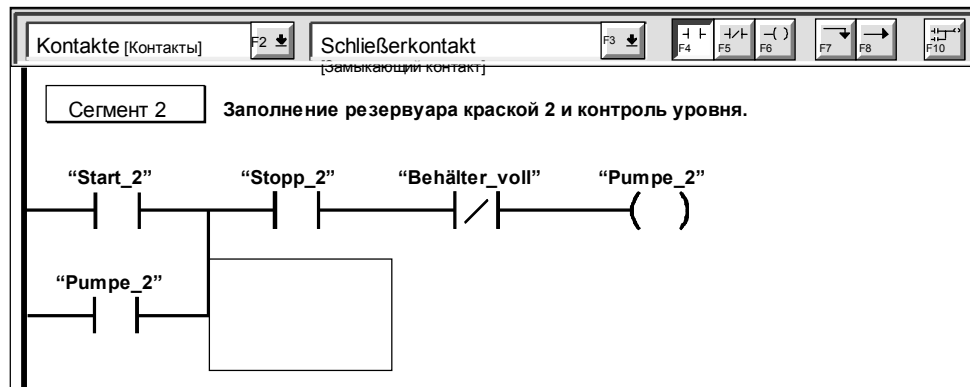



Рис. 4-11. Ввод второго сегмента

### Ввод остальных сегментов

При вводе остальных сегментов действуйте точно так же, как было описано для предыдущих сегментов. Остальные сегменты представлены на рисунке 4–3.


### Компиляция программы

После ввода программы проверьте синтаксис, выбрав команду меню **CPU** → **Übersetzen** [**CPU** → **Компиляция**] или щелкнув мышью на следующей кнопке для компиляции: .

Если Вы ввели сегменты правильно, то отображается сообщение "Übersetzung erfolgreich" ["Компиляция успешна"]. Это сообщение показывает, кроме того, количество сегментов и занимаемый программой объем памяти. Если Вами при вводе допущена ошибка, то сообщение показывает, в каком сегменте содержится ошибка.

### Сохранение примера программы

Сохраните Ваш проект с помощью команды меню **Projekt** → **Speichern** [**Проект** → **Сохранить**] или

щелкните мышью на следующей кнопке для сохранения: . Все компоненты проекта Вашего примера сохраняются.

## 4.5 Создание таблицы состояний/принудительного задания

### Создание таблицы состояний/принудительного задания

Для контроля состояния определенных элементов в Вашем примере программы Вы должны создать таблицу состояний/принудительного задания, в которой содержатся элементы, которые Вы хотели бы наблюдать во время обработки программы. Для открытия редактора таблицы дважды щелкните мышью на его пиктограмме у нижнего края панели управления. Для того, чтобы ввести элементы из примера программы, действуйте следующим образом:

1. Выделите первое поле в столбце "Adresse" ["Адрес"] и введите **Start\_1**.
2. Подтвердите Ваш ввод с помощью клавиши ввода. Этот элемент может отображаться только в двоичном формате ("1" или "0"), поэтому Вы не можете изменить формат.
3. Выделите следующую строку. Повторите описанные выше шаги для остальных элементов (см. рис. 4-12).

Совет: Если Вы выбрали адресное поле и строка под этим полем пустая, то Вы можете путем нажатия клавиши ввода автоматически увеличивать адрес на 1 в каждой последующей строке. Подробную информацию о работе с таблицей состояний/принудительного задания можно получить через Online-помощь.

С помощью команды меню **Bearbeiten** → **Reihe einfügen** [Редактирование → Вставка строки] (или с помощью клавиши INSERT) Вы вставляете новую строку, а именно, каждый раз над строкой, в которой находится курсор.

4. Таймер T37 и счетчик Z30 могут по мере надобности отображаться в других форматах. Если Вы выделили поле в столбце "Format", то Вы можете с помощью клавиши пробела вызывать друг за другом все форматы, которые действительны для соответствующего элемента. В данном примере выберите для таймера и счетчика формат "Mit Vorzeichen" ["Со знаком"].

Сохраните Вашу таблицу состояний/принудительного задания с помощью команды меню **Projekt** →

**Speichern** [Проект → Сохранить] или щелкните мышью на следующей кнопке для сохранения:

Adresse	Format	Aktueller Wert	Wert ändern in
"Start_1"	Binär	2#0	
"Start_2"	Binär	2#0	
"Stopp_1"	Binär	2#0	
"Stopp_2"	Binär	2#0	
"Behälter_voll"	Binär	2#0	
"Behälter_leer"	Binär	2#0	
"Rücksetzen"	Binär	2#0	
"Pumpe_1"	Binär	2#0	
"Pumpe_2"	Binär	2#0	
"Rührmotor"	Binär	2#0	
"Dampfventil"	Binär	2#0	
"Abflußventil"	Binär	2#0	
"Abflußpumpe"	Binär	2#0	
"Max_Füllstand"	Binär	2#0	
"Mischzeit"	Mit Vorzeichen	+0	
"Zykluszähler"	Mit Vorzeichen	+0	

Рис. 4-12. Таблица состояний/принудительного задания для примера программы

Пояснения к таблице на рис. 4-12.

Наименования столбцов: Adresse - Адрес; Format - Формат; Aktueller Wert - Текущее значение; Wert ändern in - Изменить значение на.



Форматы: Binär - двоичный; Mit Vorzeichen - со знаком.

## 4.6 Загрузка и контроль примера программы

На следующем шаге Вы должны загрузить Вашу программу в CPU и затем перевести CPU в режим RUN. Тогда Вы сможете тестировать Вашу программу и наблюдать ее обработку.

### Загрузка проекта в CPU

Для того, чтобы Вы могли загрузить программу в CPU, последний должен находиться в состоянии STOP. Чтобы перевести CPU в режим STOP и загрузить программу, действуйте следующим образом:

1. Поставьте переключатель режимов работы CPU в положение TERM или STOP.
2. Выберите команду меню **CPU → Stop** или щелкните мышью на кнопке STOP  в главном окне.
3. Подтвердите этот процесс посредством "Ja" ["Да"].
4. Выберите команду меню **CPU → Laden aus PG... [CPU → Загрузить из PG...]** или щелкните мышью на кнопке для загрузки из PG в главном окне: .
5. В диалоговом окне "Laden aus PG" ["Загрузка из PG"] Вы можете указать, какие компоненты Вы хотите загрузить в CPU. Затем нажмите клавишу ввода или щелкните мышью на кнопке "OK".

Сообщение уведомит Вас о том, был ли процесс загрузки успешным.

---

#### Указание


STEP 7-Micro/WIN не проверяет, являются ли операнды и адреса входов и выходов в Вашей программе допустимыми для соответствующего CPU. Если Вы загружаете в CPU программу, операнды и адреса которой находятся вне области, допустимой для CPU, или операции которой не поддерживаются CPU, то эта программа не может загрузиться в CPU. Вам отображается сообщение об ошибке.

Обращайте внимание на то, чтобы операнды и адреса входов и выходов, а также операции в Вашей программе были допустимыми для CPU, с которым Вы работаете.

---

### Перевод CPU в режим работы RUN

Если программа была загружена в CPU успешно, то после этого Вы можете перевести CPU в режим RUN.

1. Выберите команду меню **CPU → RUN** или щелкните мышью на кнопке для режима работы RUN  в главном окне.
2. Подтвердите этот процесс посредством "Ja" ["Да"].

### Наблюдение статуса KOP

Если Вы включили статус KOP, то отображается текущее состояние событий в Вашей программе.

Откройте окно редактора KOP и выберите команду меню **Testen** → **KOP-Status ein** [Тестирование → Статус KOP включен].

Если Вы подключили к входным клеммам Вашего CPU имитатор входных сигналов, то Вы можете включать разные переключатели и следить за потоком сигнала, а также за обработкой программы. Например, если Вы включаете входы **E0.0** и **E0.2**, а переключатель входа **E0.4** ("Behälter\_voll" [Резервуар полон]) выключен, то поток сигнала первого сегмента будет полным. Тогда сегмент становится таким, как показано на рисунке 4–13.

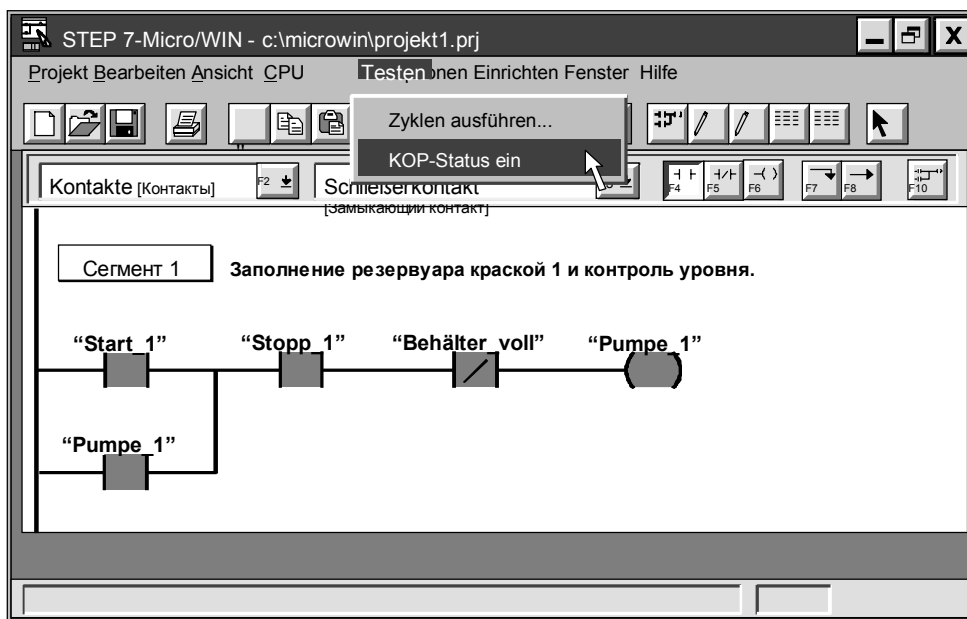


Рис. 4-13. Контроль состояния первого сегмента




Пояснения к рис. 4-13.

Меню Testen [Тестирование]: Zyklen ausführen - Выполнение циклов; KOP-Status ein - Статус KOP включен.



### Отображение текущего состояния элементов программы

С помощью таблицы состояний/принудительного задания Вы можете наблюдать и изменять текущие значения произвольных входов и выходов, а также адресов памяти. Откройте окно для таблиц состояний/принудительного задания и выберите команду меню **Testen** → **Tabellenstatus ein** [Тестирование → Статус таблиц включен] (см. рис. 4–14). Если CPU находится в режиме RUN, а входы включаются и выключаются, то таблица состояний/принудительного задания показывает текущее состояние отдельных элементов.

- Если Вы хотите запустить отображение текущего состояния элементов в Вашей программе, то щелкните мышью в таблице состояний/принудительного задания на кнопке для однократного считывания  или на кнопке для постоянного чтения .
- Если Вы хотите закончить актуализацию статуса, то щелкните мышью в таблице состояний/принудительного задания на кнопке STOP .

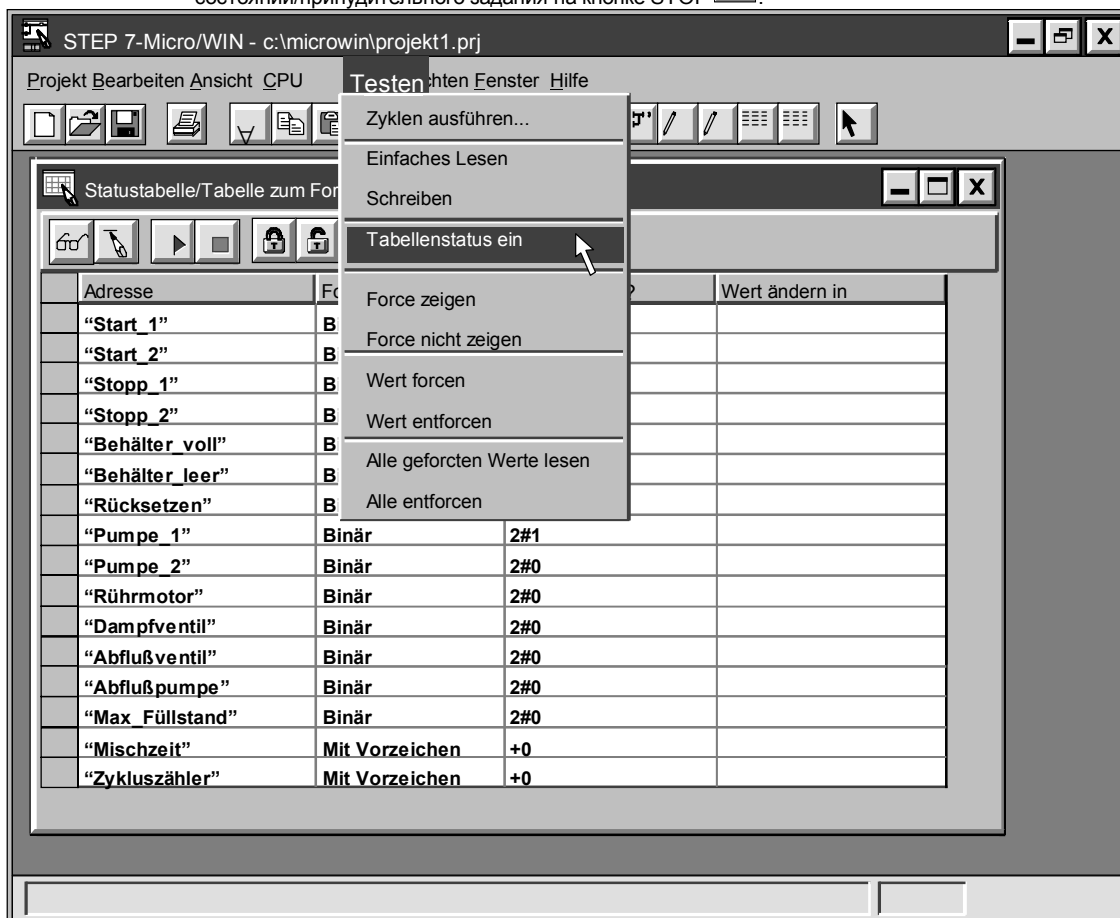


Рис. 4-14. Контроль состояний в примере программы с помощью таблицы состояний / принудительного задания

Пояснения к рис. 4-14. Меню Testen: Zyklen ausführen - Выполнение циклов; Einfaches Lesen - Простое чтение; Schreiben - Запись; Tabellenstatus ein - Статус таблицы включен; Force zeigen - Показать установки; Force nicht zeigen - Не показывать установки; Wert forcen - Установить значение; Wert entforcen - отменить установку значения; Alle geforcten Werte lesen - Прочитать все принудительно установленные значения; Alle entforcen - Отменить все принудительные установки.

*Ввод примера программы*