SIMATIC

S7-200 Примеры

Группа	Тема
3	**Использование NETR и NETW**

Краткое описание

Этот пример объясняет использование команд чтения (NETR) и записи (NETWs) в сети для CPU 214, CPU 212 и TD 200. Простая программа использует сетевые команды для зажигания LEDs на одном CPU, отображающие входы другого, и для отображения значений каждого байта входов (IB0) и байта выходов (QB0) на TD200.

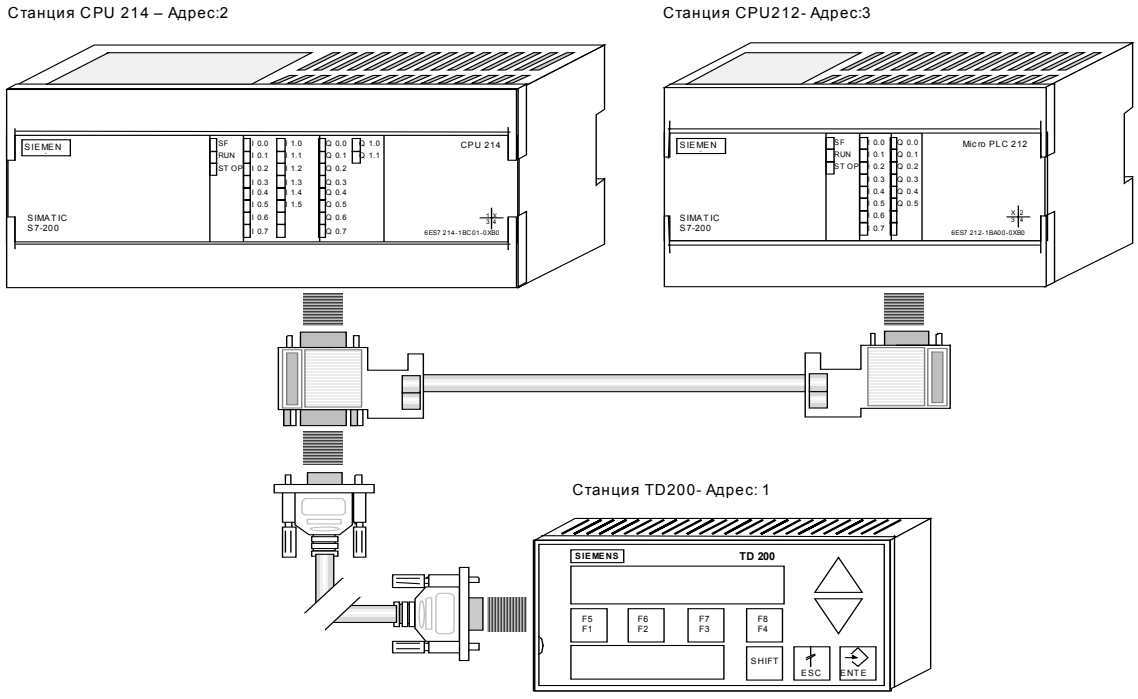


Рисунок 48.1 Конфигурация сети

Описание программы

Часть 1: Аппаратура

Этот пример показывает правильное использование команд NETR и NETW вместе с CPU 214, CPU 212 и TD 200. На Рисунке 48.1 показано структура сети для нашего примера.

Каждое устройство имеет изменяемый сетевой адрес для предотвращения коллизий при обмене данными. ПЛК имеет по умолчанию адрес 2, программатор - 0, а большинство других устройств (таких как TD200) - 1. Т.к. при использовании значений по умолчанию в локальной сети находятся две станции с адресом 2 (CPU 212 и CPU 214), Вы должны установить для одного ПЛК (например для 212) другой адрес. В нашем примере 214 имеет адрес 2, 212 - 3, а TD200 - 1. Вы можете изменить адрес станции используя или Micro/DOS или Micro/Win (Более подробную информацию Вы найдете в руководстве пользователя).

Во время выполнения нашей программы 214 ведет себя как активная станция; Micro/DOS и Micro/Win не поддерживают при программировании сеть с несколькими активными станциями (только, если Вы не используете карту MPI; в этом случае этот абзац не имеет значения). Без карты MPI Вы должны убрать все активные станции кроме одной (программатора) из сети для правильной работы программатора. Когда Вы подсоединяете программатор к 214, 212 илиTD 200, по крайней мере два устройства в сети думают, что они являются активными станциями: TD 200 и программатор. Когда 214 находится в состоянии RUN и установлен в режим PPI+, он ведет себя как еще одна активная станция и продолжает мешать программатору. Простейший способ избежать эту проблему - это отсоединить ПЛК, который Вы программируете от сети, перевести его в состояние STOP и подключить его напрямую к программатору (Рисунок 48.2).

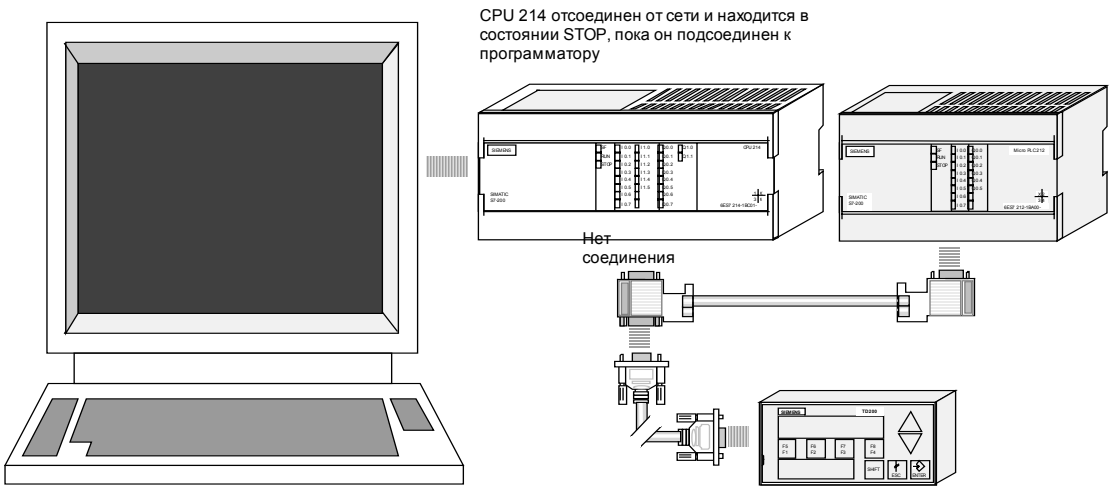


Рисунок 48.2 Возможное соединение с программатором

Если Вы не хотите отсоединять ПЛК от сети, Вы просто можете перевести ПЛК в состояние STOP и отсоединить TD 200 (Рисунок 48.3).

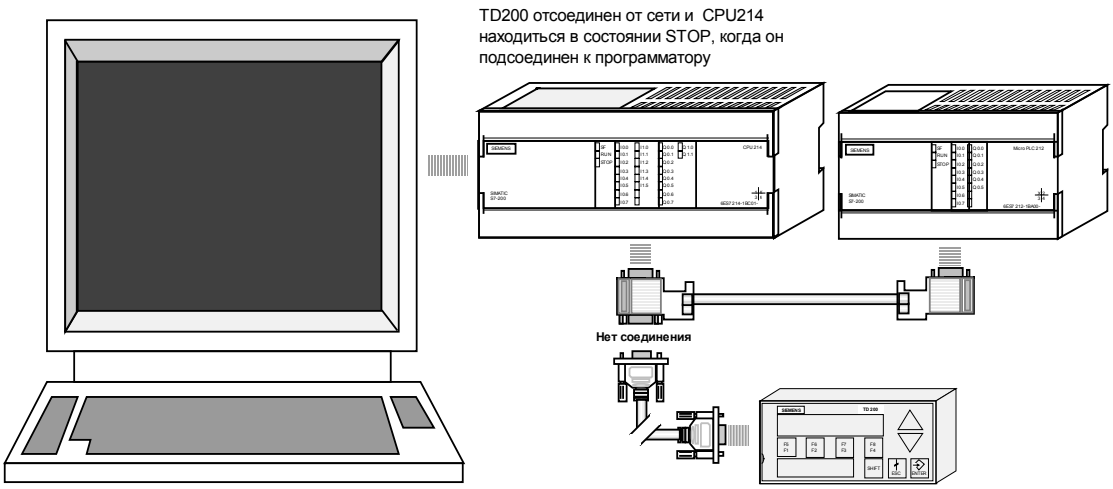


Figure 48.3 2 Другое возможное соединение с программатором

Как было сказано выше, в нашем примере мы будем использовать две сетевые команды CPU214: NETR и NETW (Network-Read и Network-Write). Каждая из этих команд требует два параметра: первый байт из таблицы (находящийся в V-памяти) и порт (который всегда имеет значение 0, т.к. CPU 214 имеет только один порт). Таблица (или буфер) состоит из 7 байт "заголовка" и от 1 до 16 байт "данных". На Рисунке 48.4 показана более подробная информация о различных составляющих таблицы.

Выполнено (функция завершена) : 0 = нет	1 = да
Активна (функция обрабатывается) : 0 = нет	1 = да
Ошибка (функция вернула ошибку) : 0 = нет	1 = да

Байт смещен. 7 … 0

0	B	A	O	0	Код ошибки	
1	Адрес удаленной станции					
2	Указатель на Область данных Удаленной станции (I,Q,M or V)					
3						
4						
5						
6	Длина данных					
7	Бит данных 0					
8	Бит данных 1					
	…					
22	Бит данных 15					

Адрес удаленной станции: адрес ПЛК, к данным которого обращаются.

Указатель на область данных удаленной станции : косвенный указатель на адресуемые данные.

Длина данных : число байтов данных, адресуемых в удаленной станции (1 – 16 байт).

Область данных для приема или передачи: 1 – 16 байт, зарезервированных для данных. NETR записывает сюда полученные данные, а NETW записывает сюда данные для удаленной станции.

Рисунок 48.4 Таблица для NETR и NETW

Часть 2: Программа

Программа, включенная в данный пример (загружаемая в CPU 214), зажигает светодиоды на выходах CPU 212 соответственно входам CPU 214, и наоборот. Она также использует TD 200 для отображения значений IB0 и QB0 для обоих 214 и 212.

Во время первого цикла в первом сегменте выполняется несколько шагов. Во-первых, SMB30 устанавливается равным 2, устанавливается режим PPI+ и сообщается 214, что он является активной станцией. Во-вторых, очищается область памяти для трех таблиц (используемые NETR и NETW), которые создаются V200 (мы назовем ее Таблица 1), V250 (Таблица 2) и V300 (Таблица 3). Программа использует Таблицу 1 для адресации станции 3, чтения/записи IB0 и для данных, размером 1. Таблицы 2 и 3 тождественны Таблице 1, но обе для чтения/записи QB0.

Следующие три сегмента проверяют биты "обработано" для таблиц 1, 2 и 3 соответственно. Т.к. ни один из этих битов не устанавливается во время первого цикла, программа ничего не делает и переходит к сегменту 5, который использует SM0.0 для хранения отображаемых битов сообщений TD 200. Мы потом еще раз вернемся к сегментам 2 - 4.

Во время первого цикла в последнем сегменте устанавливается бит "обработано" для первой таблицы. На первый взгляд этот сегмент кажется не на правильном месте (Почему он не на первом месте в программе и откладывается обработка SM0.1 ?), но есть два довода, чтобы он находился здесь. Во-первых: Когда SMB30 устанавливается равным 2, чтобы работать в режиме PPI+ как активная станция, то потом до конца цикла не выполняется больше никакое действие. Таким образом, после установки SMB30 во время первого цикла, больше не происходит никаких изменений до второго цикла; также, после устанавки его во время двенадцатого цикла, не происходит никаких изменений до тринадцатого цикла. Во-вторых, команды NETR и NETW не работают, до тех пор пока SMB30 не воспримит изменения. Таким образом, если мы установили бит "обработано" в начале программы, при вызове NETW будет получена ошибка, и (конечно) не установиться бит "обработано" и программа будет находиться в режиме ожидания, наблюдая за битом "обработано" до перезапуска. Эта позиция позволяет программе опросить его, если это необходимо, до исполнения NETR и NETW.

Вы может быть удивлены всей этой игрой с битами"обработано". Идя такова: каждый раз, когда CPU 214 получает двойной запрос из сети, он помещает их в очередь. Чем больше очередь, тем менее регулярно выполняются команды NETR и NETW, и вся система работает медленнее. В нашем примере это избегается при помощи вышеупомянутого бита"обработано".

Когда выполняется команда NETR или NETW и адресуется какая-то таблица, то устанавливается бит "обрабатывается". Когда команда выполнена, бит "обрабатывается" очищается, а бит "обработано" устанавливается. Используя бит"обработано", мы играем с сегментами так, что в любое время активен только один сетевой запрос. Таким образом, когда бит "обработано" для 1 установлен, 2 - вызывается; когда для 2 - установлен, 3 - вызывается; когда для 3 - установлен, 1 - вызывается; и так далее (см. Рисунок 48.5). Конечно, в сегментах выполняется больше, чем было описано выше.

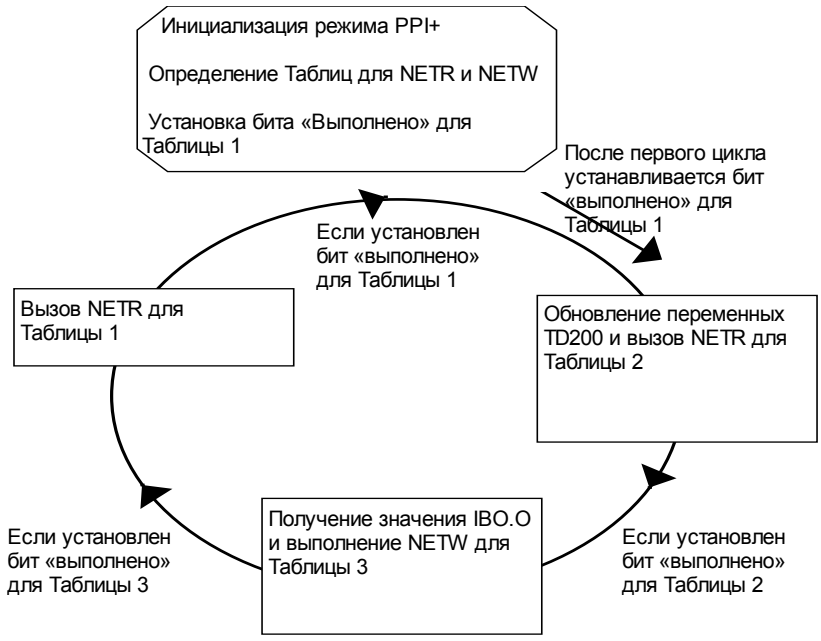


Рисунок 48.5 Диаграма состояний

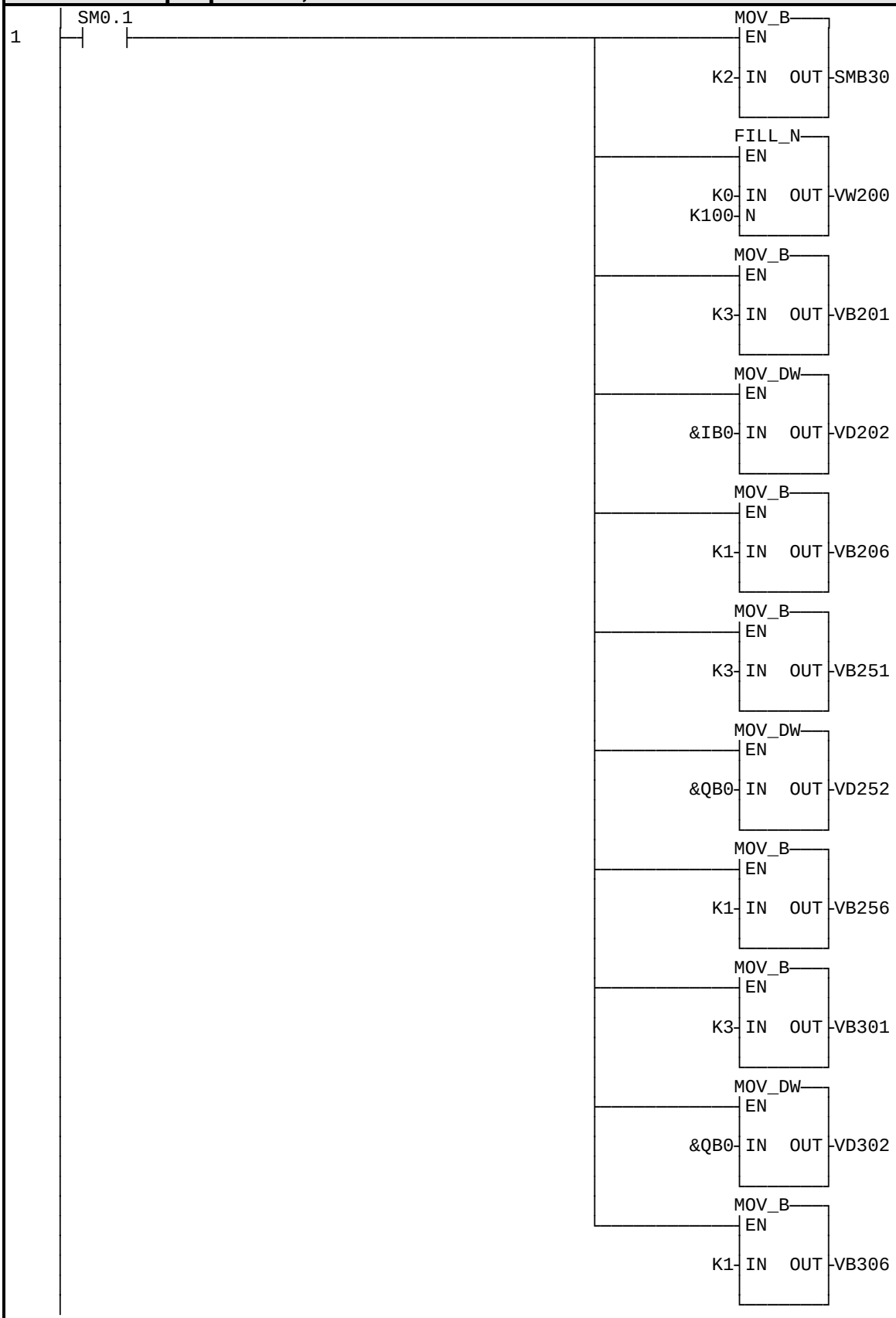
Если бит "обработано" для Таблицы 1 установлен, сегмент 2 копирует значение IB0 в область данных Таблицы 3 и вызывает NETW для Таблицы 3 записывая это значение в QB0 CPU 212.

Если установлен бит "обработано" для Таблицы 2, сегмент 3 берет значение из области данных Таблицы 1 (равное значению IB0 CPU 212) и записывает его в QB0 CPU 214 и в одну из переменных сообщений TD 200. Он также копирует IB0 и QB0 CPU 214 в переменные сообщений TD 200. И наконец он копирует данные из Таблицы 2 (значение QB0 CPU 212), завершая процесс обновления переменных сообщений TD200 перед вызовом NETR для Таблицы 2, чтобы получить новое значение QB0 CPU 212.

Если установлен бит "обработано" для Таблицы 3, вызывается NETR для Таблицы 1, чтобы получить значение IB0 CPU 212.

Более подробную информацию о сетевых командах Вы найдете в Главе 6, “Специальные команды” в *Руководстве по программированию S7-200*.

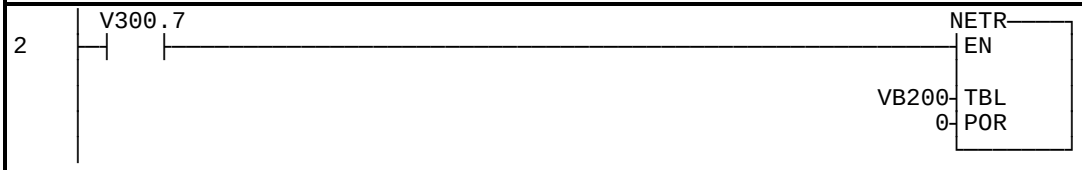
Описание программы, включая листинг



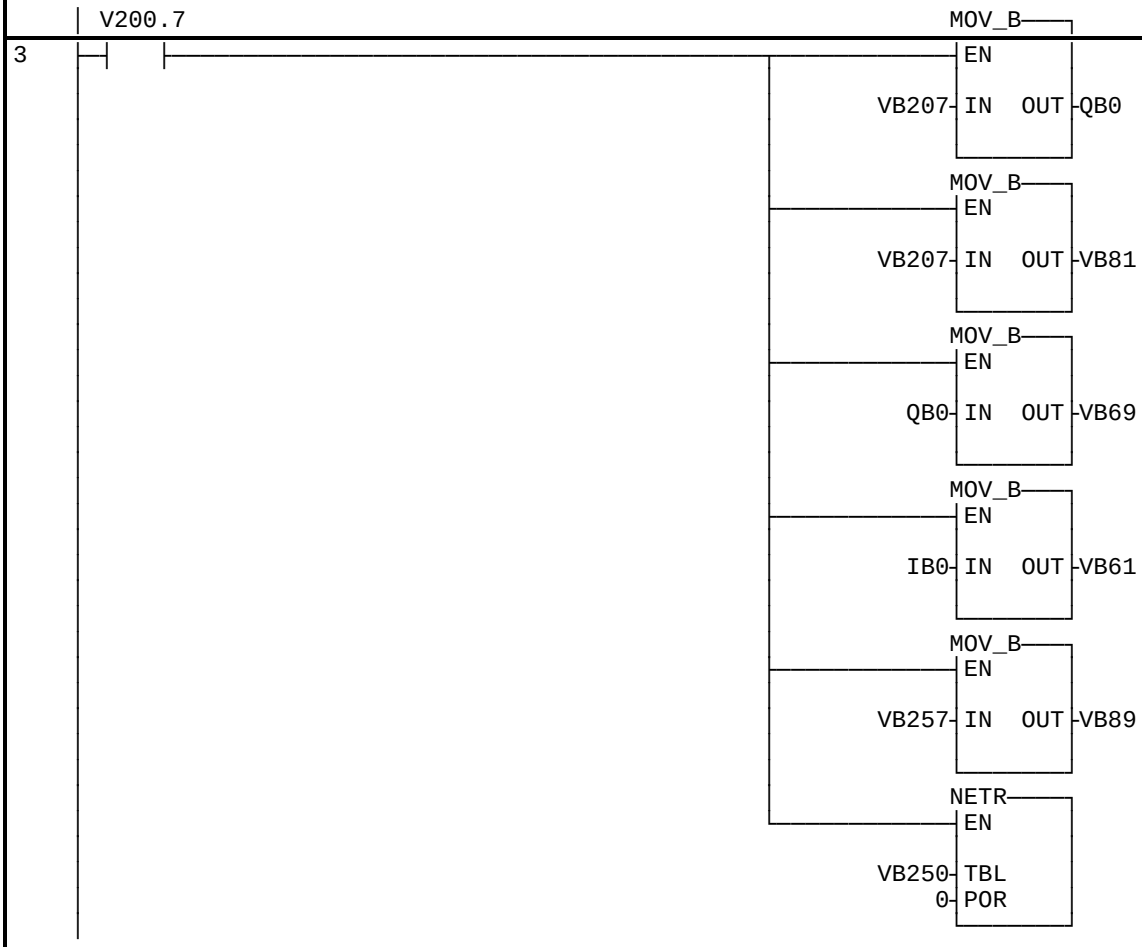

```
LD      SM0.1           // Выполнить этот сегмент только во время первого цикла

MOVB    2, SMB30        // Установить CPU в режим PPI+
FILL    0, VW200, 100   // Очистить область памяти, используемую для таблиц
                        //  *** Установить Таблицу 1 @ VB200 (для NETR) ***
```

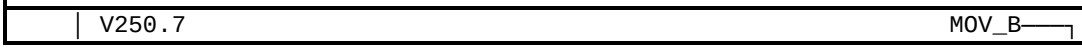
```
MOVB    3, VB201            // Адрес устройства = '3'
MOVD    &IB0, VD202         // Установить адрес для чтения = первому байту входов
MOVB    1, VB206            // Установить размер передачи = 1
                            //   *** Установить Таблицу 2 @ VB250 (для NETR) ***
MOVB    3, VB251            // Адрес устройства = '3'
MOVD    &QB0, VD252         // Установить адрес для чтения = первому байту выодов
MOVB    1, VB256            Установить размер передачи = 1
                            //   *** Установить Таблицу 3 @ VB300 (для NETW) ***
MOVB    3, VB301            // Адрес устройства = '3'
MOVD    &QB0, VD302         // Установить адрес для чтения = первому байту выодов
MOVB    1, VB306            // Установить размер передачи = 1
```

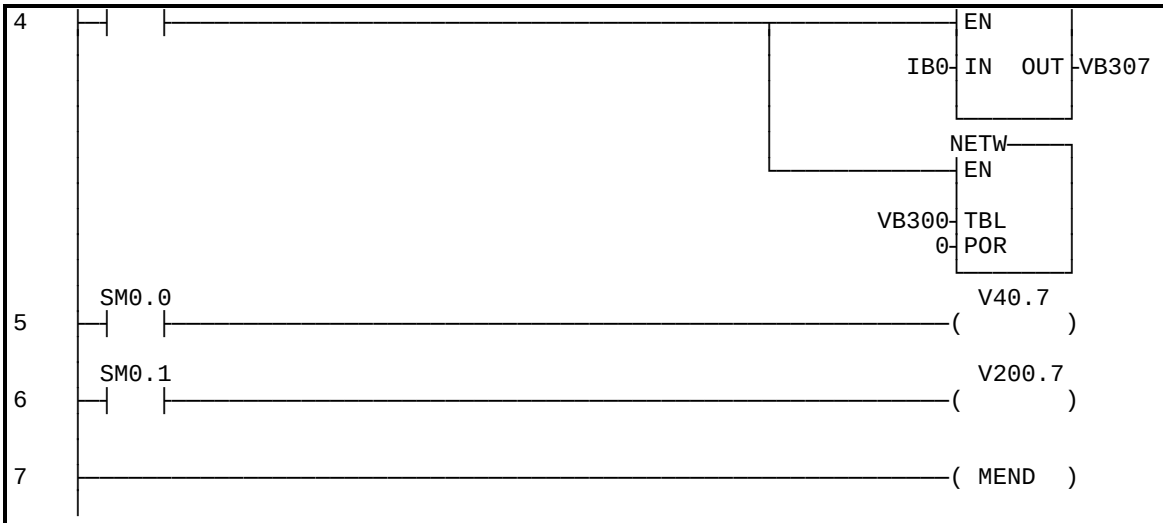



```
LD      V300.7              // Если закончена команда NETW для Таблицы 3,
NETR    VB200, 0            // Выполнить NETR для Таблицы 1
```




```
LD      V200.7              // Если закончена команда NETR для Таблицы 1
MOVB    VB207, QB0          // Передать полученное значение из Таблицы 1 в байт выходов
MOVB    VB207, VB81         // Передать полученное значение из Таблицы 1 на экран TD200
MOVB    QB0, VB69           // Передать значение байта выходов в TD200
MOVB    IB0, VB61           // Передать значение байта входов в TD200
MOVB    VB257, VB89         // Передать полученное значение из Таблицы 2 в TD200
NETR    VB250, 0        // Вызвать NETR для Таблицы 2
```






```
LD      V250.7          // Если закончена команда NETR для Таблицы 2
MOVB    IB0, VB307      // Передать значение байта входов в Таблицу 3
NETW    VB300, 0        // Вызвать NETW для Таблицы 3
                        // для записи IB0 в пассивную станцию

LD      SM0.0           // Для отображения сообщений на TD200
=       V40.7           // всегда использовать один бит

LD      SM0.1           // Только во время первого цикла
=       V200.7          // установить бит "обработано" для Таблицы 1

MEND                    // Конец
```

Блок данных

```
VB0     'TD'            // Начало блока параметров TD200
VW2     16#1001         // Английский; обновлять как можно быстрее
VW4     16#0100         // 1 сообщение
VW6     50              // Сообщения начинаются с VB50
VW8     40              // Разрешения начиная с VB40
VB50    '214: I0='      // Часть сообщения
VW58    16#0030         // Конфигурация переменных
VW60    16#0000         // Значение переменных
VB62    ' Q0='          // Следующие данные сообщений
VW66    16#1030         // Конфигурация переменных
VW68    16#0000         // Значение переменных
VB70    '212: I0='      // Следующие данные сообщений
VW78    16#0030         // Конфигурация переменных
VW80    16#0000         // Значение переменных
VB82    ' Q0='          // Следующие данные сообщений
VW86    16#1030         // Конфигурация переменных
VW88    16#0000         // Значение переменных
```

Указания по преобразованию

Для того чтобы преобразовать IEC STL в S7-Micro/DOS STL

- Добавьте ‘K’ перед каждым числом, не являющимся шестнадцатеричной константой (например, 4 ⇒ K4)
- Замените ‘16#’ на ‘KH’ для всех шестнадцатеричных констант (например, 16#FF ⇒ KHFF)
- Поставьте запятые для смены полей. Используйте клавиши перемещения или клавишу TAB для перехода от поля к полю.
- Для преобразования программы S7-Micro/DOS STL в LAD-форму каждый сегмент должен начинаться со слова ‘NETWORK’ и номера. Кажiый сегмент в этом примере имеет свой номер на диаграмме LAD. Используйте команду INSNW в меню редактора для ввода нового сегмента. Команды MEND, RET, RETI, LBL, SBR и INT требуют отдельных сегментов.
- Коментарии строк, обозначенные "//" не поддерживаются в S7-Micro/DOS, но разрешены коментарии сегментов

Общие указания

Примеры применения SIMATIC S7-200 предназначены для того, чтобы дать пользователям S7-200 начальную информацию, как можно решить с помощью данной системы управления определенные задачи. Данные примеры применения S7-200 бесплатны.

В приведенных примерах программ речь идет об идеях решения без претензии на полноту или работоспособность в будущих версиях программного обеспечения S7-200 или STEP7 Micro. Для соблючения соответствующих технически безопасных предписаний при применении необходимо предпринять дополнительные меры.

Ответственность Siemens, все равно по каким правовым нормам, при возникновении ущерба из-за применеия примеров программ исключается, равно и при ущербе личным вещам, персональном ущербе или при намеренных или грубо неосторожных действиях.