

# Использование команд протокола USS для взаимодействия с приводами

# 11

Эта глава описывает стандартные команды протокола USS, позволяющие S7-200 управлять приводом MicroMaster. Команды протокола USS являются характерной особенностью STEP 7-Micro/WIN 32 Toolbox.

STEP 7-Micro/WIN 32 Toolbox упрощает управление приводами MicroMaster за счет включения предварительно скомпонованных подпрограмм и программ обработки прерываний, специально разработанных для обмена данными по протоколу USS. Эти программы появляются в качестве команд в папке библиотек дерева команд STEP 7-Micro/WIN. При помощи этих новых команд вы можете управлять физическим приводом и параметрами чтения/записи для привода.

Когда вы выбираете команду протокола USS, автоматически добавляются одна или более связанных с ней подпрограмм (от USS1 до USS7). Вам нет необходимости уделять внимание этим подпрограммам.

## Обзор главы

Раздел	Описание	Страница
11.1	Требования команд протокола USS	11–2
11.2	Последовательность программирования	11–4
11.3	Команды протокола USS	11–5
11.4	Подключение приводов	11–17
11.5	Настройка приводов	11–18

## 11.1 Требования команд протокола USS

Команды протокола USS требуют от вашего CPU следующих ресурсов:

- память емкостью 1250–1750 байтов (в зависимости от используемых команд)
- (только) порт 0
- 8 подпрограмм и 3 программы обработки прерываний
- V-память емкостью 400 байтов и 16-байтовый буфер для некоторых команд

### Использование ресурсов

Команда протокола USS использует ресурсы так, как указано ниже:

- Порт 0: Когда порт 0 используется для обмена данными по протоколу USS, он не может использоваться для какой-либо другой цели, включая обмен данными со STEP 7-Micro/WIN. Команда USS\_INIT управляет назначением порта 0 протоколу USS или PPI. Как только порт 0 был назначен для использования USS, порт 0 можно использовать снова для обмена данными с STEP 7-Micro/WIN только путем программного переназначения с использованием другой команды USS\_INIT или посредством установки переключателя режимов в положение STOP. Прекращение связи с приводами заставляет приводы остановиться. Рекомендуется, чтобы во время разработки программы с приложениями протокола USS разработчик использовал CPU 226 или EM 277 PROFIBUS-DP в сочетании с платой PROFIBUS CP в PC. Это позволяет использовать второй коммуникационный порт для контроля приложения со стороны STEP 7-Micro/WIN, пока выполняется протокол USS.
- Ячейки SM: команды протокола USS влияют на все ячейки SM, относящиеся к свободно программируемой связи через порт 0.
- Объем памяти программы пользователя: кроме памяти, занимаемой каждой командой, в программе пользователя имеются дополнительные затраты памяти величиной до 1750 байтов, занимаемых программами поддержки протокола USS.
- V-память: для переменных USS резервируется блок V-памяти размером 400 байтов, начиная с ячейки, назначенной пользователем. Для некоторых команд требуется 16-байтовый буфер обмена данными, начинающийся в ячейке, назначенной пользователем. Рекомендуется назначать уникальный буфер каждому экземпляру команд протокола USS.
- Подпрограммы и программы обработки прерываний: протокол USS воспринимает 9 подпрограмм и 3 программы обработки прерываний.

### Конфигурация таблицы глобальных символов

После того как вы ввели адрес первого символа, таблица автоматически вычисляет и назначает оставшуюся часть символов таблицы. На рис. 11–1 показана вкладка USS в таблице символов. Вы должны назначить начальную V-ячейку для таблицы, занимающей 400 байтов.

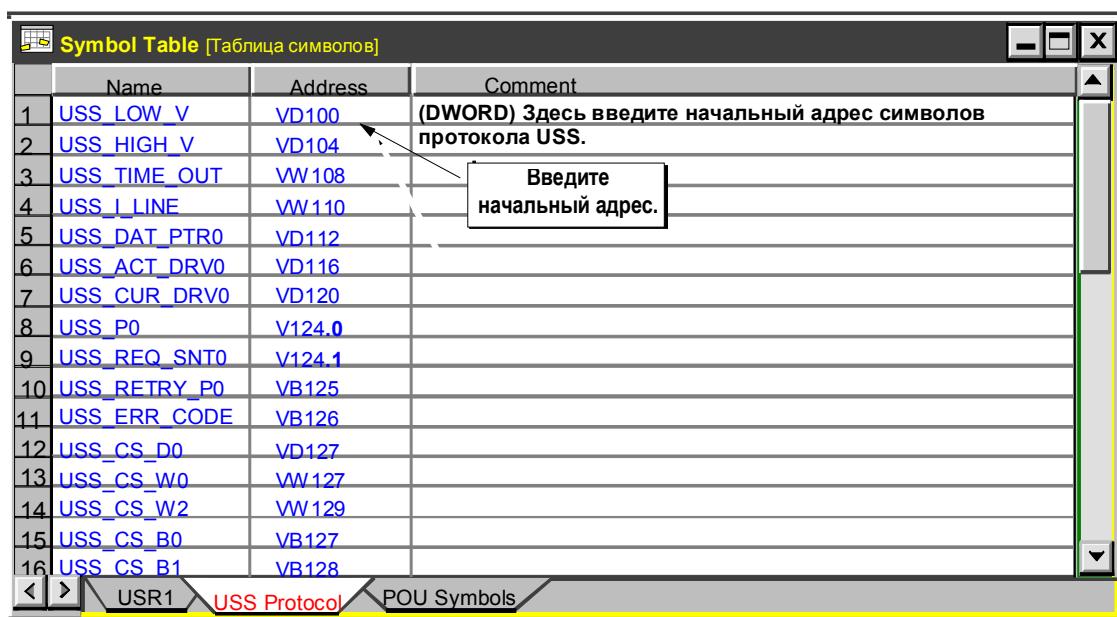


Рис. 11-1. Конфигурация таблицы символов

### Время обмена данными с приводом

Обмен данными с приводом является асинхронным по отношению к циклу сканирования CPU. За время одной коммуникационной транзакции с приводом обычно завершается несколько циклов сканирования CPU. Это зависит от числа имеющихся приводов, скорости передачи и времени цикла сканирования CPU. Таблица 11–1 показывает времена коммуникационных транзакций. Как только команда USS\_INIT назначает порт 0 протоколу USS, CPU регулярно опрашивает все активные приводы через интервалы времени, указанные в таблице 11–1. Параметр времени ожидания каждого привода должен устанавливаться с учетом этого времени опроса.

Таблица 11–1. Времена обмена данными с приводом

Скорость передачи (Бод)	Время между опросами активных приводов (мс)
1200	(460 максимум/230 обычно) * количество приводов
2400	(240 максимум/120 обычно) * количество приводов
4800	(130 максимум/65 обычно) * количество приводов
9600	(80 максимум/40 обычно) * количество приводов
19200	(50 максимум/25 обычно) * количество приводов

### Ограничение

В отдельный момент времени может быть активной только одна команда READ\_PM или WRITE\_PM. Выход Done каждой команды должен сообщать о завершении прежде, чем логика программы пользователя запустит новую команду. Для каждого привода используйте только одну команду DRV\_CTRL.

## 11.2 Последовательность программирования

Последовательность программирования при использовании команд протокола USS указана ниже:

1. Поместите команду USS\_INIT в программу пользователя. Это автоматически добавит в программу несколько скрытых подпрограмм и программ обработки прерываний. Команда USS\_INIT должна вызываться только на один цикл обработки программы, чтобы инициализировать или изменить параметры обмена данными по USS. Для получения дополнительной информации о команде USS\_INIT см. страницу 11–5.
2. Назначьте первой ячейке в таблице глобальных символов USS адрес V-памяти. Все другие адреса назначаются автоматически. Всего требуется 400 последовательных байтов. См. на рис. 11–1 вкладку USS в таблице символов.
3. Поместите в свою программу только по одной команде DRV\_CTRL на активный привод. Вы можете добавлять столько команд READ\_PM и WRITE\_PM, сколько желаете, но только одна из них может быть активной в каждый конкретный момент времени.
4. Конфигурируйте параметры привода, чтобы согласовать используемые скорость передачи и адрес с программой. Обратитесь к пункту “Настройка привода” в разделе 11.5.
5. Подключите коммуникационный кабель между CPU и приводами. Очень важно, чтобы любая подключаемая к приводу аппаратура управления (например, ПЛК) соединялась с той же самой шиной “земля” или нейтральной точкой “звезды”, что и привод, с использованием короткого толстого кабеля.



### Предостережение

Внутренняя связь оборудования с различными опорными потенциалами может вызвать протекание нежелательных токов через соединительный кабель. Эти нежелательные токи могут вызывать ошибки передачи данных или повреждать оборудование. Чтобы предотвратить протекание нежелательных токов, убедитесь, что все оборудование, которое вы собираетесь соединить при помощи коммуникационного кабеля, либо совместно использует общую опорную точку для электрических цепей, либо гальванически развязано. См. «Заземление и опорная точка цепи при использовании изолированных цепей» в разделе 2.3. Экран должен быть заземлен на шасси или соединен с клеммой 1 в 9-контактном разъеме. Рекомендуется соединить клемму 2-0V в приводе MicroMaster с заземлением на шасси.

---

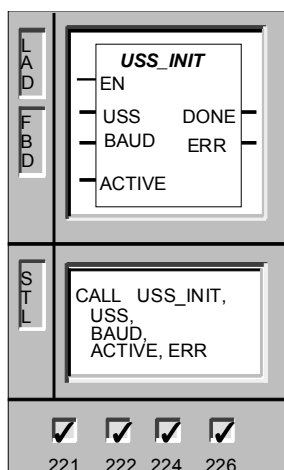
### Примечание

Если вы не можете считывать все переменные в блоках команд USS, то **выберите в меню пункт View → Zoom [Вид → Масштабирование], а затем увеличьте ширину сетки.**

---

### 11.3 Команды протокола USS

#### USS\_INIT



Команда **USS\_INIT** используется для того, чтобы деблокировать и инициализировать или блокировать обмен данными с приводом MicroMaster. Команда USS\_INIT должна быть выполнена без возврата кода ошибки, прежде чем можно будет использовать любую другую команду протокола USS. Эта команда завершается, и сразу устанавливается бит Done (флаг готовности) перед переходом к следующей команде.

Команда выполняется в каждом цикле обработки программы, когда включен вход EN. Команда USS\_INIT должна выполняться ровно один раз для каждого изменения состояния обмена данными. Поэтому вход EN должен включаться импульсом через элемент обнаружения фронта сигнала.

Если протокол USS был запущен, то он должен блокироваться путем выполнения новой команды USS\_INIT прежде, чем может быть сделано изменение в параметрах инициализации.

Значение входа USS выбирает протокол обмена данными. Значение 1 назначает порт 0 протоколу USS и деблокирует этот протокол. Значение 0 назначает порт 0 протоколу PPI и блокирует протокол USS.

BAUD устанавливает скорость 1200, 2400, 4800, 9600 или 19200 Бод.

ACTIVE указывает, какие приводы являются активными. Некоторые приводы поддерживают только адреса с 0 по 30. Рис. 11–2 показывает описание и формат входа активного привода. Любой привод, отмеченный как ACTIVE, автоматически опрашивается в фоновом режиме работы для того, чтобы управлять приводом, собирать данные о состоянии и предоставлять простой последовательного канала связи в приводе. Для вычисления времени между опросами состояния обратитесь к таблице 11–1 на стр. 11–3. Для получения информации о том, как настроить параметр времени простоя последовательного канала связи (P093), обратитесь к разделу 11.5.

MSB				LSB			
31	30	29	28	3	2	1	0
D31	D30	D29			D2	D1	D0
D0 Бит активности привода 0; 0 = привод не активен, 1 = привод активен							
D1 Бит активности привода 1; 0 = привод не активен, 1 = привод активен							
.							
D31 Бит активности привода 31; 0 = привод не активен, 1 = привод активен							

Рис. 11–2. Описание и формат входа активного привода

Когда команда USS\_INIT завершается, включается выход DONE. Выходной байт ERR содержит результат выполнения команды. Таблица 11–6 на странице 11–16 определяет возможные ошибочные состояния, которые могут следовать из выполнения команды.

В таблице 11–2 показаны операнды и типы данных подпрограмм USS.

Таблица 11–2. Операнды и типы данных подпрограммы USS\_INIT

Входы/выходы	Операнды	Типы данных
USS	VB, IB, QB, MB, SB, SMB, LB, AC, константа, *VD, *AC, *LD	BYTE
BAUD	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, константа, AC *VD, *AC, *LD	WORD
ACTIVE	VD, ID, QD, MD, SD, SMD, LD, AC, константа, *VD, *AC, *LD	DWORD
DONE	I, Q, M, S, SM, T, C, V, L	BOOL
ERR	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	BYTE

На рис. 11–3 показано, как подпрограмма USS\_INIT используется в LAD, FBD и STL.

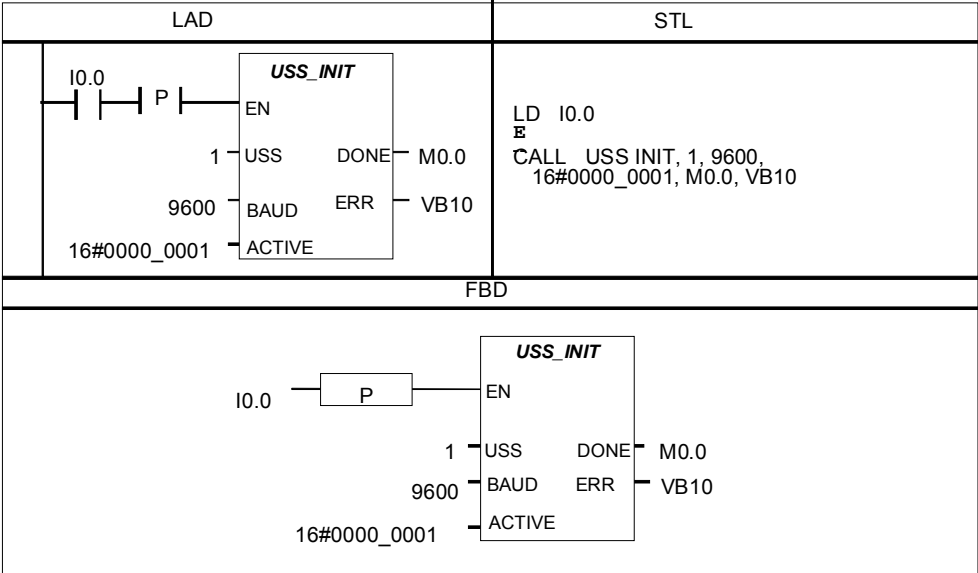
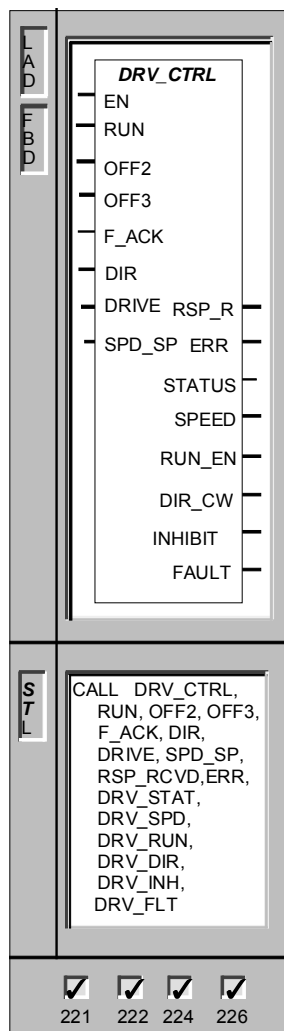


Рис. 11–3. Использование подпрограммы USS\_INIT в LAD, FBD и STL

## DRV\_CTRL



Команда **DRV\_CTRL** используется для управления АКТИВНЫМ приводом MicroMaster. Команда DRV\_CTRL размещает выбранные команды в коммуникационном буфере. Команды из буфера передаются адресованному приводу (параметр DRIVE), если этот привод был выбран в параметре ACTIVE команды USS\_INIT. Каждый привод должен иметь только одну команду DRV\_CTRL.

Бит EN должен быть включен, чтобы деблокировать команду DRV\_CTRL. Как правило, эта команда всегда деблокирована.

Бит RUN (RUN/STOP) указывает, включается ли привод (1) или выключается (0). Когда бит RUN включается, привод MicroMaster получает команду начать вращение с заданной скоростью в заданном направлении. Чтобы привести в действие привод, нужно выполнить следующее:

- DRIVE должен быть выбран как ACTIVE в USS\_INIT.
- OFF1 и OFF2 должны быть установлены в 0.
- FAULT и INHIBIT должны быть равны 0.

Когда бит RUN выключается, приводу MicroMaster передается команда линейно понижать скорость до останова двигателя.

Бит OFF2 используется для того, чтобы разрешить приводу MicroMaster вращаться по инерции до останова. Бит OFF3 используется, чтобы подать приводу MicroMaster команду остановиться быстро.

Бит F\_ACK (Fault Acknowledge [подтверждение неисправности]) используется для того, чтобы подтвердить неисправность привода. Когда F\_ACK переходит с низкого уровня на высокий, привод сбрасывает сигнал неисправности (FAULT).

Бит DIR (направление) указывает в каком направлении привод должен вращаться (0 - против часовой стрелки, 1 - по часовой стрелке).

Вход DRIVE (адрес привода) является адресом привода MicroMaster, которому должна передаваться команда DRV\_CTRL. Действительными являются адреса с 0 по 31.

SPD\_SP (установка скорости) – это скорость привода, заданная как процент от полной скорости (от -200,0% до 200,0%). Отрицательные значения SPD\_SP заставляют привод изменить направление его вращения на противоположное.

### Примечание

Каждому приводу должна назначаться только одна команда DRV\_CTRL.

Бит RSP\_R (Response Received [ответ принят]) подтверждает ответ привода. Все АКТИВНЫЕ приводы опрашиваются с целью получения информации о самом последнем состоянии привода. Каждый раз, когда CPU получает ответ привода, RSP\_R бит включается на один цикл обработки программы, и обновляются все следующие значения.

ERR – это байт ошибки, содержащий результат самого последнего коммуникационного запроса к приводу. Таблица 11–6 на странице 11–16 определяет возможные ошибочные состояния, которые могут следовать из выполнения команды.

STATUS - это необработанное значение слова состояния, возвращаемое приводом. Рис. 11–4 показывает биты состояния для стандартного слова состояния и главной обратной связи.

SPEED - это скорость привода, выраженная в процентах от полной скорости (от –200,0% до 200,0%).

---

#### **Примечание**

Некоторые приводы сообщают скорость только в форме положительного значения. Если скорость отрицательная, то привод сообщает скорость в форме положительного значения, но изменяет бит DIR\_CW (направление) на обратный.

---

RUN\_EN (RUN Enable [деблокировка RUN]) показывает, работает ли привод (1) или остановлен (0).

DIR\_CW показывает направление вращения привода (0 - против часовой стрелки, 1 - по часовой стрелке).

INHIBIT показывает состояние бита запрета в приводе (0 - не запрещен, 1 - запрещен). Для того, чтобы сбросить бит запрета, нужно выключить бит FAULT, а также выключить входы RUN, OFF2 и OFF3.

FAULT [неисправность] показывает состояние бита неисправности (0 – нет неисправности, 1 - неисправность). Привод отображает код неисправности. (Обратитесь к справочному руководству для вашего привода). Чтобы сбросить бит FAULT, устраните причину неисправности и включите бит F\_ACK.

Таблица 11–3 показывает операнды и типы данных подпрограммы DRV\_CTRL.



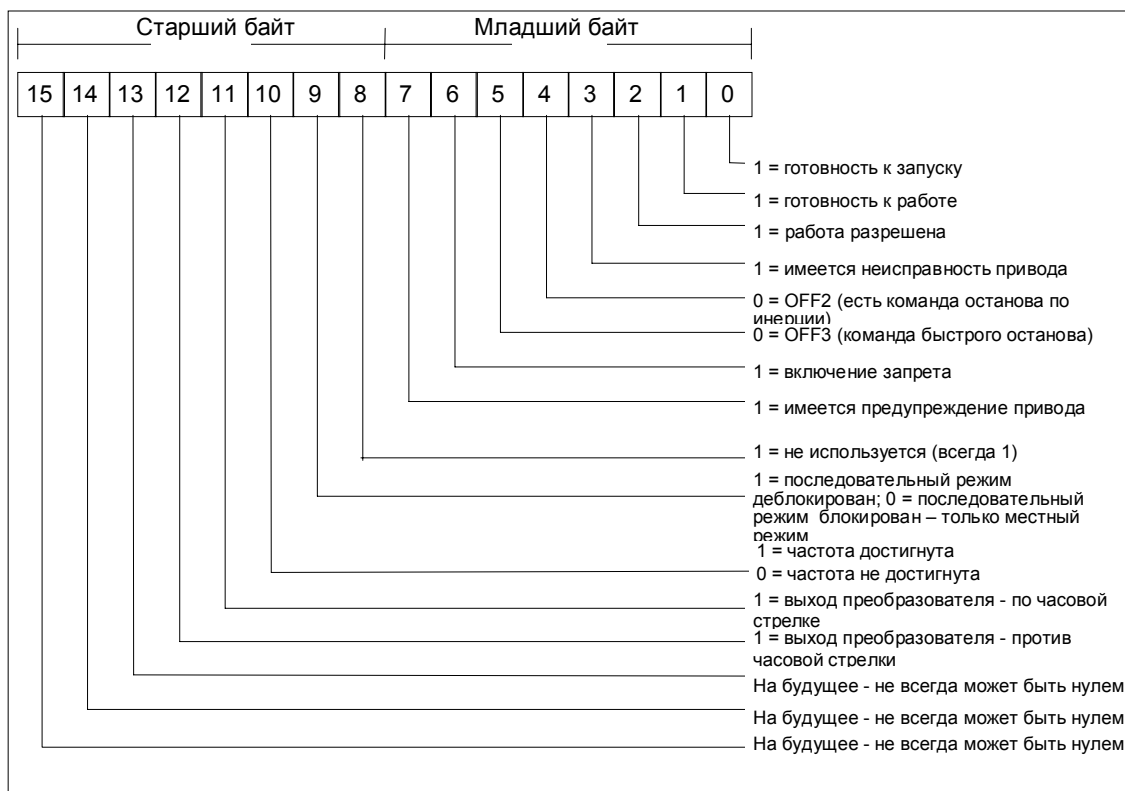


Рис. 11–4. Биты состояния для стандартного слова состояния и главной обратной связи.

Таблица 11–3. Операнды и типы данных подпрограммы DRV\_CTRL

Входы/выходы	Операнды	Типы данных
RUN	I, Q, M, S, SM, T, C, V, L, поток сигнала	BOOL
OFF2	I, Q, M, S, SM, T, C, V, L, поток сигнала	BOOL
OFF3	I, Q, M, S, SM, T, C, V, L, поток сигнала	BOOL
F_ACK	I, Q, M, S, SM, T, C, V, L, поток сигнала	BOOL
DIR	I, Q, M, S, SM, T, C, V, L, поток сигнала	BOOL
DRIVE	VB, IB, QB, MB, SB, SMB, LB, AC, константа, *VD, *AC, *LD	BYTE
SPD_SP	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD, константа	REAL
RSP_R	I, Q, M, S, SM, T, C, V, L	BOOL
ERR	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	BYTE
STATUS	VW, T, C, IW, QW, SW, MW, SMW, LW, AC, AQW, *VD, *AC, *LD	WORD
SPEED	VD, ID, QD, MD, SD, SMD, LD, AC, *VD, *AC, *LD	REAL
RUN_EN	I, Q, M, S, SM, T, C, V, L	BOOL
DIR_CW	I, Q, M, S, SM, T, C, V, L	BOOL
INHIBIT	I, Q, M, S, SM, T, C, V, L	BOOL
FAULT	I, Q, M, S, SM, T, C, V, L	BOOL

На рис. 11–5 показано, как подпрограмма DRV\_CTRL должна использоваться в LAD, FBD и STL.

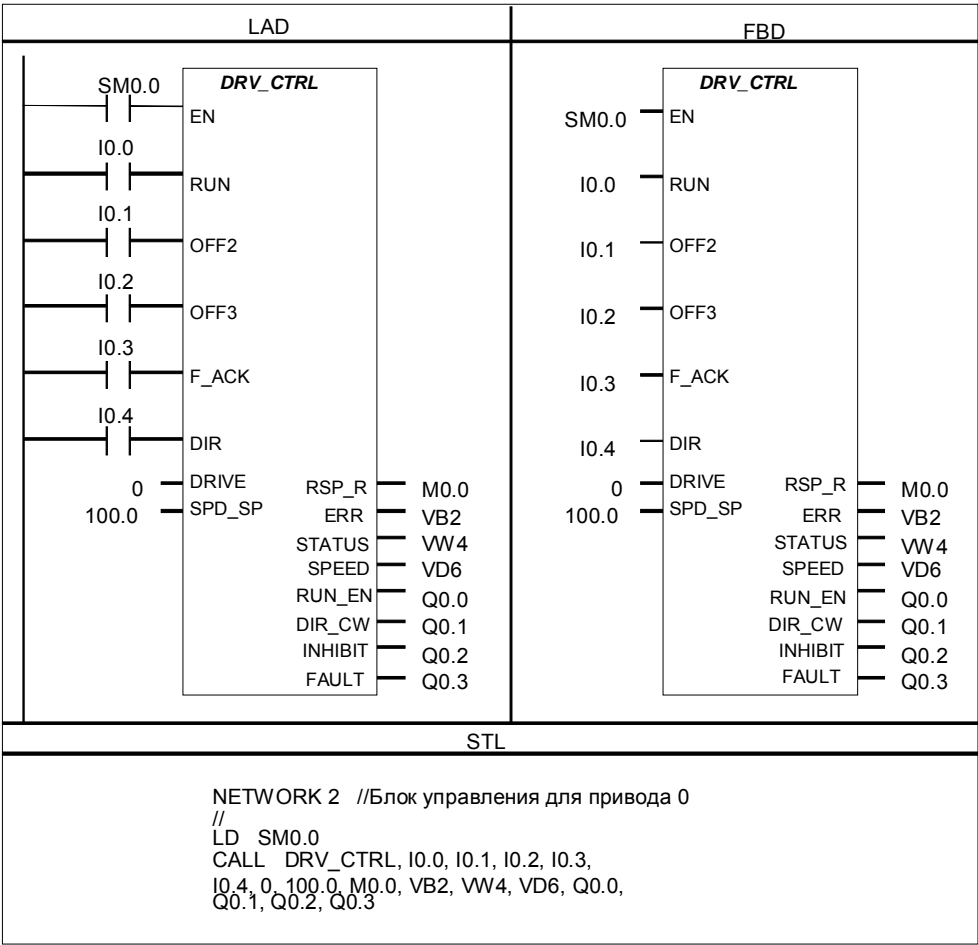
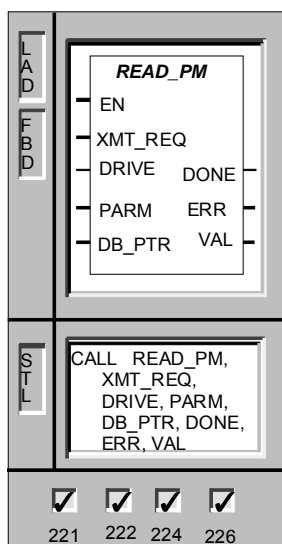


Рис. 11–5. Использование подпрограммы DRV\_CTRL в LAD, FBD и STL

## READ\_PM



Команда **READ\_PM** считывает параметр привода. Транзакция READ\_PM завершается, когда привод MicroMaster подтверждает получение команды или когда регистрируется ошибочное состояние. Пока этот процесс ожидает ответа, продолжает выполняться логическое сканирование программы. Бит EN, чтобы деблокировать передачу запроса, должен быть включен и оставаться включенным до тех пор, пока не установится бит DONE, сообщая о завершении процесса. Запрос READ\_PM передается приводу MicroMaster в каждом цикле обработки программы, когда вход XMT\_REQ включен. Поэтому вход XMT\_REQ должен включаться импульсно через элемент обнаружения фронта сигнала, вызывающий передачу одного запроса для каждого положительного фронта на входе EN.

Вход DRIVE – это адрес привода MicroMaster, которому должна передаваться команда READ\_PM. Действительными адресами отдельных приводов являются адреса с 0 по 31.

PARM является номером параметра.

На вход DB\_PTR должен подаваться адрес 16-байтового буфера. Этот буфер используется командой READ\_PM, чтобы сохранять результаты команды, поданной на привод MicroMaster.

Когда команда READ\_PM завершается, выход DONE включается, и выходной байт ERR содержит результат выполнения команды. Таблица 11–6 на странице 11–16 определяет возможные ошибочные состояния, которые могут следовать из выполнения команды.

VAL – это возвращаемое значение параметра.

### Примечание

В каждый конкретный момент времени может быть активна только одна команда READ\_PM или WRITE\_PM.

Таблица 11–4 показывает операнды и типы данных подпрограммы READ\_PM.

Таблица 11–4. Операнды и типы данных подпрограммы READ\_PM

Входы/выходы	Операнды	Типы данных
XMT_REQ	I, Q, M, S, SM, T, C, V, L, поток сигнала, обусловленный элементом обнаружения положительного фронта сигнала.	BOOL
DRIVE	VB, IB, QB, MB, SB, SMB, LB, AC, константа, *VD, *AC, *LD	BYTE
PARM	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, константа, AC *VD, *AC, *LD	WORD
DB_PTR	&VB	DWORD
DONE	I, Q, M, S, SM, T, C, V, L	BOOL
ERR	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	BYTE
VAL	VW, T, C, IW, QW, SW, MW, SMW, LW, AC, AQW, *VD, *AC, *LD	WORD

На рис. 11–6 показано, как подпрограмма READ\_PM используется в LAD, FBD и STL.

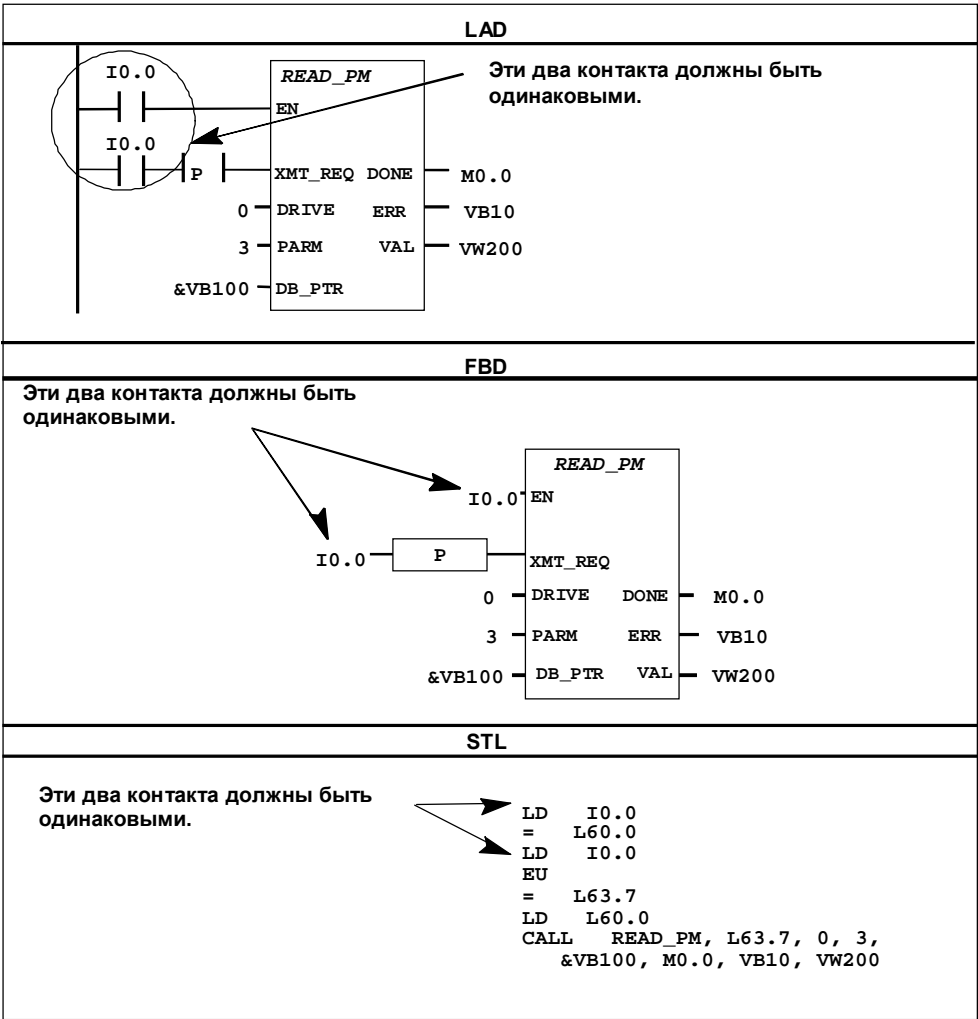
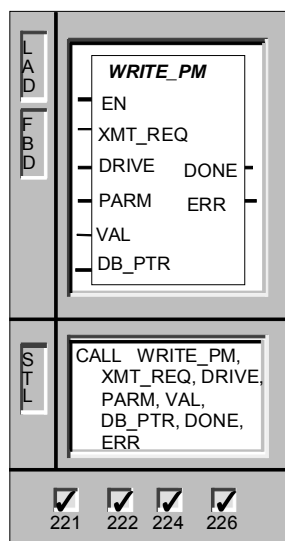


Рис. 11–6. Использование подпрограммы READ\_PM в LAD, FBD и STL

**WRITE\_PM**

Команда **WRITE\_PM** записывает параметр привода в заданную ячейку. Транзакция WRITE\_PM завершается, когда привод MicroMaster подтверждает получение команды или когда регистрируется ошибочное состояние. Пока этот процесс ожидает ответа, продолжает выполняться логическое сканирование программы.

Бит EN, чтобы деблокировать передачу запроса, должен быть включен и оставаться включенным до тех пор, пока не установится бит DONE, сообщая о завершении процесса. Запрос WRITE\_PM передается приводу MicroMaster в каждом цикле обработки программы, когда вход XMT\_REQ включен. Поэтому вход XMT\_REQ должен включаться импульсно через элемент обнаружения фронта сигнала, вызывающий передачу одного запроса для каждого положительного фронта на входе EN.

Вход DRIVE – это адрес привода MicroMaster, которому должна передаваться команда WRITE\_PM. Действительными адресами отдельных приводов являются адреса с 0 по 31.

PARM является номером параметра. VAL – это значение параметра, подлежащее записи.

На вход DB\_PTR должен подаваться адрес 16-байтового буфера. Этот буфер используется командой WRITE\_PM, чтобы сохранять результаты команды, поданной на привод MicroMaster.

Когда команда WRITE\_PM завершается, выход DONE включается, и выходной байт ERR содержит результат выполнения команды. Таблица 11–6 на странице 11–16 определяет возможные ошибочные состояния, которые могут следовать из выполнения команды.

**Примечание**

В каждый конкретный момент времени может быть активна только одна команда READ\_PM или WRITE\_PM.

**ПРЕДОСТЕРЕЖЕНИЕ**

Когда вы используете команду WRITE\_PM для обновления набора параметров в ЭСППЗУ, вы должны гарантировать, что не превышаете максимальное количество циклов записи (приблизительно 50 000), установленное для ЭСППЗУ. Превышение максимального количества циклов записи приводит к искажению запоминаемых данных и последующей потере данных. Количество циклов чтения не ограничивается. Если требуются частые записи в параметры привода, то вы должны сначала обнулять P971 (элемент управления памятью ЭСППЗУ).

В таблице 11–5 показаны операнды и типы данных подпрограммы WRITE\_PM.

Таблица 11–5. Операнды и типы данных подпрограммы WRITE\_PM

Входы/выходы	Операнды	Типы данных
XMT_REQ	I, Q, M, S, SM, T, C, V, L, поток сигнала, обусловленный элементом обнаружения положительного фронта сигнала.	BOOL
DRIVE	VB, IB, QB, MB, SB, SMB, LB, AC, константа, *VD, *AC, *LD	BYTE
PARM	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, константа, AC *VD, *AC, *LD	WORD
VAL	VW, IW, QW, MW, SW, SMW, LW, T, C, AIW, константа, AC *VD, *AC, *LD	WORD
DB_PTR	&VB	DWORD
DONE	I, Q, M, S, SM, T, C, V, L	BOOL
ERR	VB, IB, QB, MB, SB, SMB, LB, AC, *VD, *AC, *LD	BYTE

На рис. 11–7 показано, как подпрограмма WRITE\_PM используется в LAD, FBD и STL.

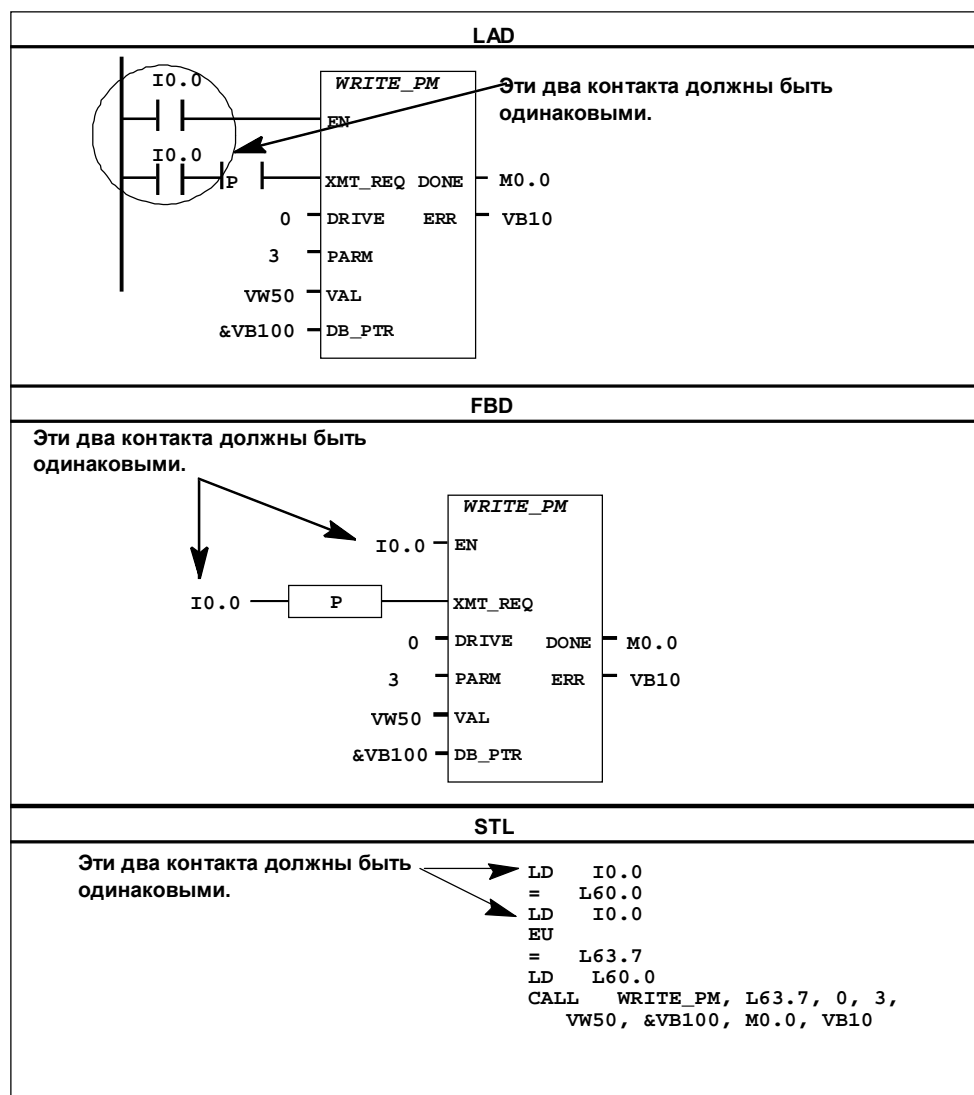


Рис. 11–7. Использование подпрограммы WRITE\_PM в LAD, FBD и STL

Таблица 11–6. Ошибки выполнения команд USS

Номер ошибки	Объяснение
0	Ошибок нет.
1	Привод не ответил.
2	В ответе привода была обнаружена ошибка контрольной суммы.
3	В ответе привода была обнаружена ошибка четности.
4	Ошибка была вызвана помехой из программы пользователя.
5	Была предпринята попытка выполнения запрещенной команды.
6	Был указан недействительный адрес привода.
7	Для протокола USS не был установлен коммуникационный порт.
8	Коммуникационный порт занят обработкой команды.
9	Вход скорости привода находится вне допустимого диапазона.
10	Длина ответа привода неверна.
11	Первый символ в ответе привода неверен.
12	Символ длины в ответе привода неверен.
13	Ответил не тот привод.
14	Адрес, поданный на DB_PTR, неверен.
15	Указанный номер параметра неверен.
16	Выбран недопустимый протокол.
17	USS активен; изменение не разрешено.
18	Установлена недопустимая скорость передачи.
19	Нет обмена данными: привод не активен.
20	Параметр или значение в ответе привода является неверен.



## 11.4 Подключение приводов

Для подключения CPU к приводу MicroMaster можно использовать стандартный кабель PROFIBUS и разъемы. Для получения дополнительной информации обратитесь к пункту Сетевые штекеры в главе 7. О том, как правильно установить смещение и оконечную нагрузку соединительных кабелей см. рис. 11–8.



### Предостережение

Внутренняя связь оборудования с различными опорными потенциалами может вызвать протекание нежелательных токов через соединительный кабель.

Эти нежелательные токи могут вызывать ошибки передачи данных или повреждать оборудование.

Чтобы предотвратить протекание нежелательных токов, убедитесь, что все оборудование, которое вы собираетесь соединить при помощи коммуникационного кабеля, либо совместно использует общую опорную точку электрических цепей, либо имеет гальваническую развязку. См. Заземление и опорная точка цепи при использовании изолированных цепей в разделе 2.3.

Экран должен соединяться с заземлением на шасси или с клеммой 1 в 9-контактном разъеме. Рекомендуется соединить клемму 2-0V в приводе MicroMaster с заземлением на шасси.

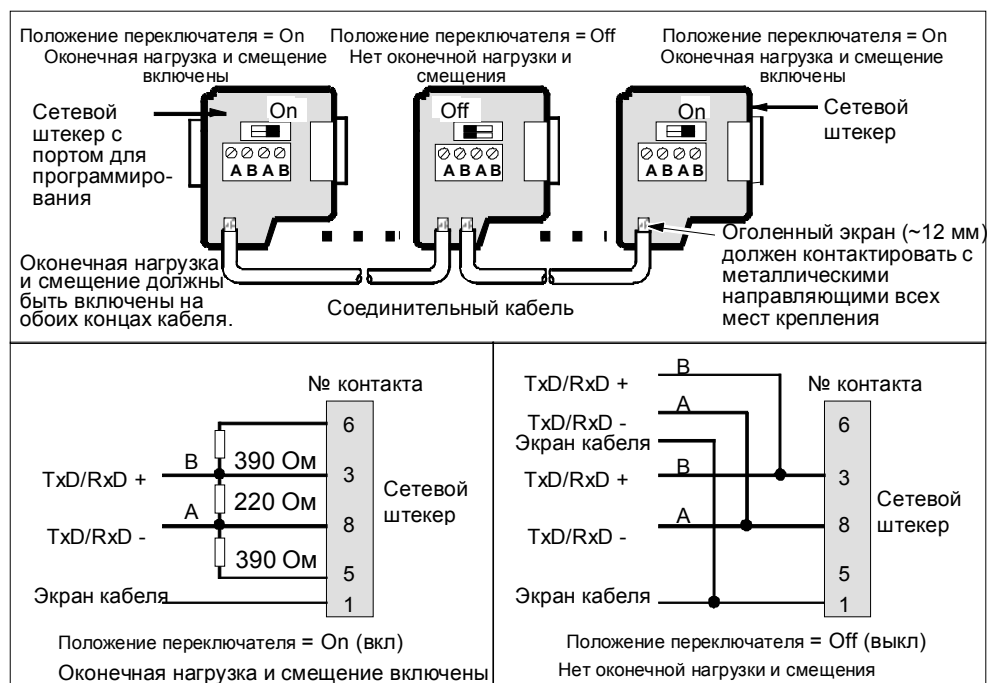


Рис. 11-8. Смещение и оконечная нагрузка соединительного кабеля

## 11.5 Настройка приводов

Перед подключением привода к ПЛК вы должны убедиться, что привод имеет указанные ниже системные параметры. Параметры устанавливаются с помощью вспомогательной клавиатуры привода.

Чтобы установить параметры в вашем приводе, выполните следующие шаги:

1. Восстановите в параметрах привода заводскую настройку (не обязательно). Нажмите клавишу P: на дисплее появится P000. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится P944. Нажмите клавишу P, чтобы ввести параметр.  
P944=1
2. Разрешите доступ для чтения/записи ко всем параметрам. Нажмите клавишу P. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится P009. Нажмите клавишу P, чтобы ввести параметр.  
P009=3
3. Проверьте параметры настройки двигателя вашего привода. Параметры настройки изменяются соответственно используемому двигателю (двигателям). Нажмите клавишу P. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится параметр настройки вашего привода. Нажмите клавишу P, чтобы ввести параметр.  
P081= Номинальная частота двигателя (Гц)  
P082= Номинальная скорость двигателя (оборотов в минуту)  
P083= Номинальный ток двигателя (А)  
P084= Номинальное напряжение двигателя (В)  
P085= Номинальная мощность двигателя (кВт/л.с.)
4. Установите режим управления "локальный/дистанционный". Нажмите клавишу P. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится P910. Нажмите клавишу P, чтобы ввести параметр.  
P910=1 Режим управления "дистанционный".
5. Установите скорость передачи последовательного интерфейса RS-485. Нажмите клавишу P. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока не появится P092. Нажмите клавишу P, чтобы ввести параметр. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится номер, соответствующий скорости передачи вашего последовательного интерфейса RS-485. Для ввода нажмите клавишу P.  
P092 =    3        (1200 Бод)  
             4        (2400 Бод)  
             5        (4800 Бод)  
             6        (9600 Бод – по умолчанию)  
             7        (19200 Бод)

6. Введите адрес slave-устройства. Каждым приводом (максимум 31 привод) можно управлять через шину. Нажмите клавишу P. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока не появится P091. Нажмите клавишу P, чтобы ввести параметр. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится желаемый адрес slave-устройства. Для ввода нажмите клавишу P.  
P091=0÷31.
7. Время (времена) линейного нарастания скорости (не обязательно). Это время в секундах, которое требуется двигателю для разгона до максимальной частоты. Нажмите клавишу P. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока не появится P002. Нажмите клавишу P, чтобы ввести параметр. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится желаемое время линейного разгона. Для ввода нажмите клавишу P.  
P002=0÷650.00
8. Время (времена) линейного убывания скорости (не обязательно). Это время в секундах, которое требуется двигателю для торможения до полного останова. Нажмите клавишу P. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока не появится P003. Нажмите клавишу P, чтобы ввести параметр. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится желаемое время линейного торможения. Для ввода нажмите клавишу P.  
P003=0÷650.00
9. Время ожидания последовательного канала связи. Это максимальный допустимый промежуток времени между двумя входящими пакетами данных. Это свойство используется для выключения инвертора в случае нарушения связи.  
  
Отсчет времени начинается после получения действительного пакета данных. Если следующий пакет данных не принимается в течение установленного периода времени, то инвертор отключается и отображается код ошибки F008. Установка значения в нуль выключает блок управления. Время между опросами состояния привода можно вычислить, используя таблицу 11–1.  
  
Нажмите клавишу P. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока не появится P093. Нажмите клавишу P, чтобы ввести параметр. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится желаемое время ожидания последовательного канала связи. Для ввода нажмите клавишу P.  
P093=0÷240 (0 задан по умолчанию; время в секундах)
10. Номинальная системная уставка последовательного канала связи. Это значение может изменяться, но обычно соответствует 50 Гц или 60 Гц, что определяет соответствующее 100% значение для PV или SP. Нажмите клавишу P. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока не появится P094. Нажмите клавишу P, чтобы ввести параметр. Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится желаемая номинальная системная уставка последовательного канала связи. Для ввода нажмите клавишу P.  
P094=0÷400.00

11. Совместимость с USS (не обязательно). Нажмите клавишу Р.  
Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока не появится Р095. Нажмите клавишу Р, чтобы ввести параметр.  
Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится номер, соответствующий желаемой совместимости с USS. Для ввода нажмите клавишу Р.  
Р095 = 0 разрешающая способность 0,1 Гц  
(значение по умолчанию)  
1 разрешающая способность 0,01 Гц
12. Управление памятью ЭСППЗУ (не обязательно). Нажмите клавишу Р.  
Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока не появится Р971. Нажмите клавишу Р, чтобы ввести параметр.  
Нажимайте клавишу "стрелка вверх" или "стрелка вниз", пока на дисплее не появится номер, соответствующий желаемому управлению памятью ЭСППЗУ. Для ввода нажмите клавишу Р.  
Р971 = 0 Изменения в настройке параметров  
(включая Р971) теряются  
при снятии питания.  
1 (значение по умолчанию). Изменения в настройке параметров сохраняются в периоды, когда питание снимается.
13. Приведение в действие дисплея. Нажмите Р, чтобы выйти из режима задания параметров.

## Пример программы протокола USS

На рисунках 11–9 ÷ 11–11 показан пример программы USS в LAD, FBD и STL.

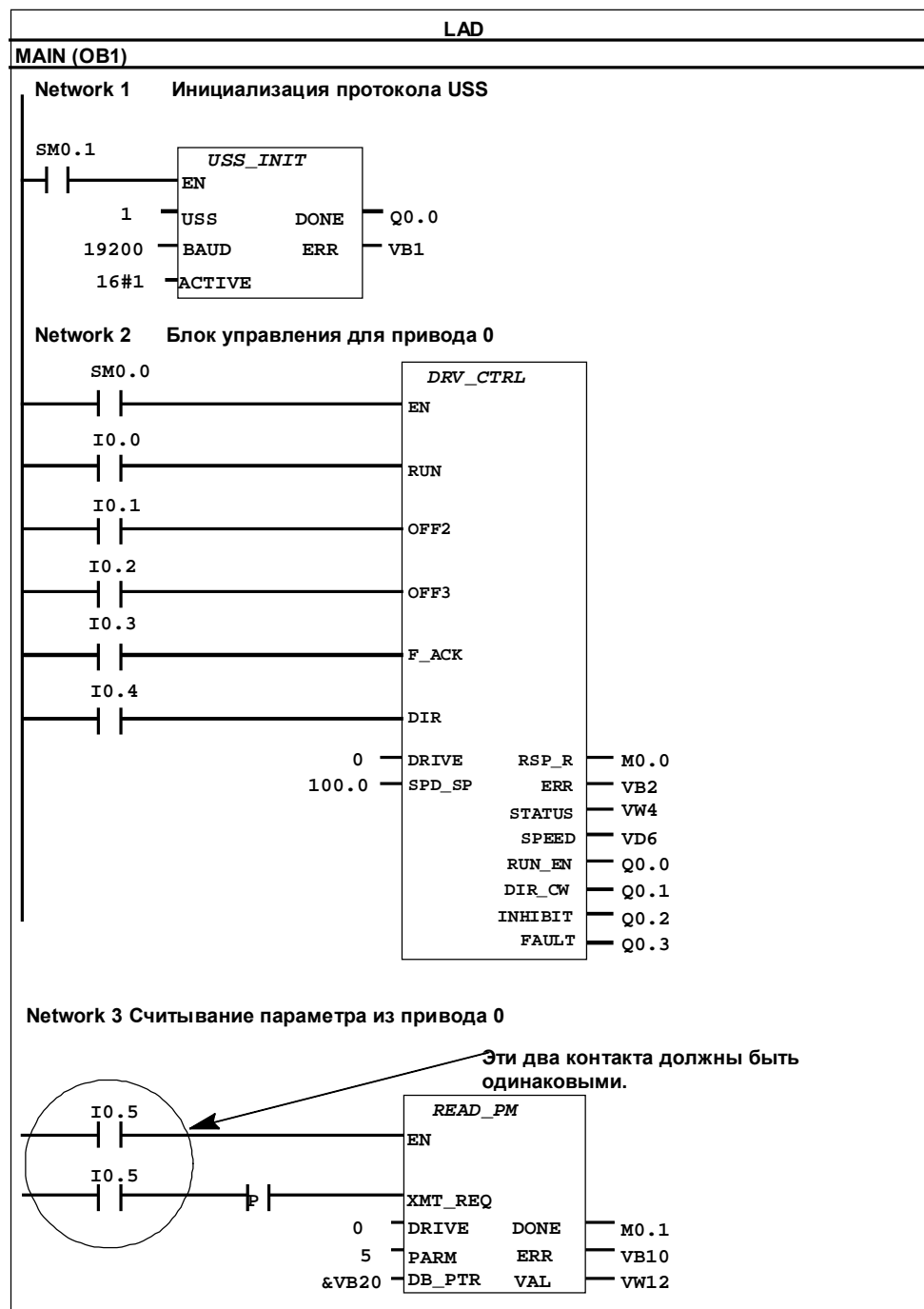


Рис. 11-9. Пример команд USS для SIMATIC LAD

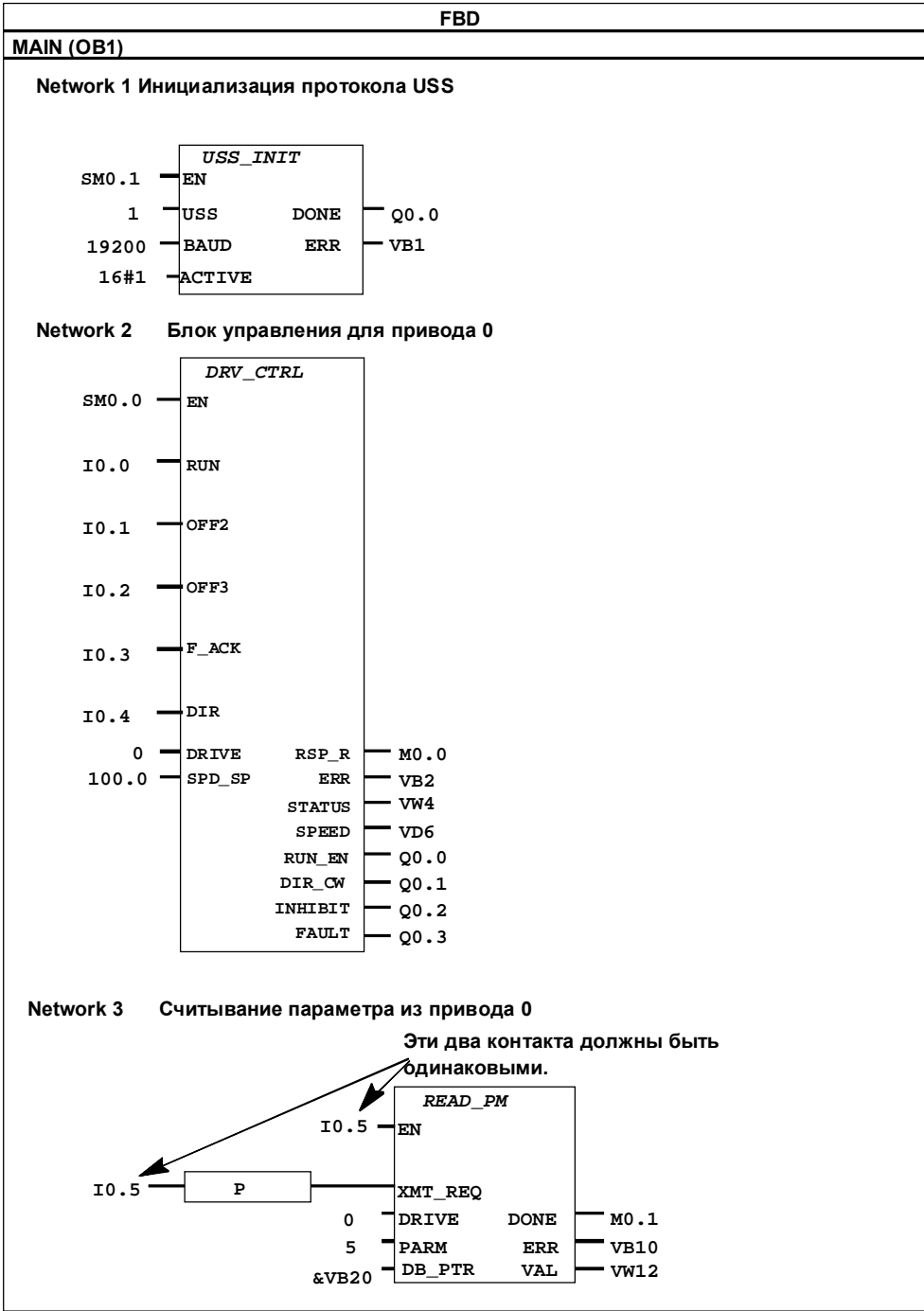


Рис. 11-11. Пример команд USS для SIMATIC FBD

```

1 NETWORK 1 //Инициализация протокола USS
2 //
3 LD SM0.1 //В первом цикле выполнения программы
4 CALL USS_INIT, 1, 19200, 16#1, Q0.0, VB1
5 //разблокировать протокол USS для порта 0 при скорости 19200 бод
6 //с активным адресом привода "0".
7 NETWORK 2 //Блок управления для привода 0
8 //
9 LD SM0.0
10 CALL DRV_CTRL, I0.0, I0.1, I0.2, I0.3, I0.4, 0, 100.0, M0.0, VB2, VW4,
11 VD6, Q0.0, Q0.1, Q0.2, Q0.3
12
13 NETWORK 3 //Считывание параметра привода 0
14 //
15 LD I0.5
16 = L60.0 //Сохранить состояние I0.1 во временной L-ячейке так, чтобы
17 //этот сегмент отображался в LAD.
18 LD I0.5
19 EU
20 = L63.7 //Сохранить импульс положительного фронта сигнала I0.2 во временной L-ячейке,
21 // которой он может передаваться для подпрограммы CW
22 LD 60.0
23 CALL READ_PM, L63.7, 0, 5, &VB20, M0.1, VB10, VW12
24
25

```

Эти два контакта должны быть одинаковыми.

Рис. 11-11. Пример команд USS для SIMATIC STL

