

SIMATIC

S7-200 Примеры

Группа

5

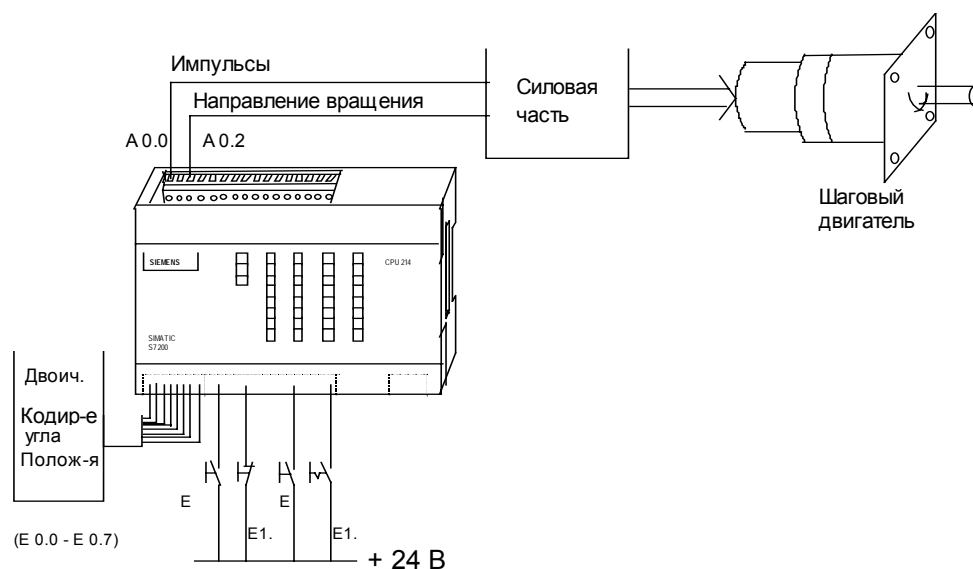
Пример к теме

Управляемое позиционирование привода шагового двигателя через импульсные выходы S7-200 CPU 214 DC/DC/DC

Краткое описание

В задачах позиционирования различают режимы *регулирования* и *управления*. Шаговые двигатели не требуют постоянного регулирования положения и могут работать в режиме управления. В данном примере программирования производится *относительное* позиционирование с приводом шагового двигателя через встроенные импульсные выходы CPU 214. Хотя при таком режиме позиционирования базовая точка не требуется, здесь описаны также простейшие действия по заданию базовой метки, так как на практике появляется необходимость, присвоить оси фиксированную точку отсчета. Затем пользователь задает ЦПУ угол позиционирования в двоичном виде через входной байт. Программа пользователя вычисляет число необходимых шагов позиционирования и приказывает ЦПУ выдать соответствующее число управляющих импульсов.

Схема включения



Industrial automation

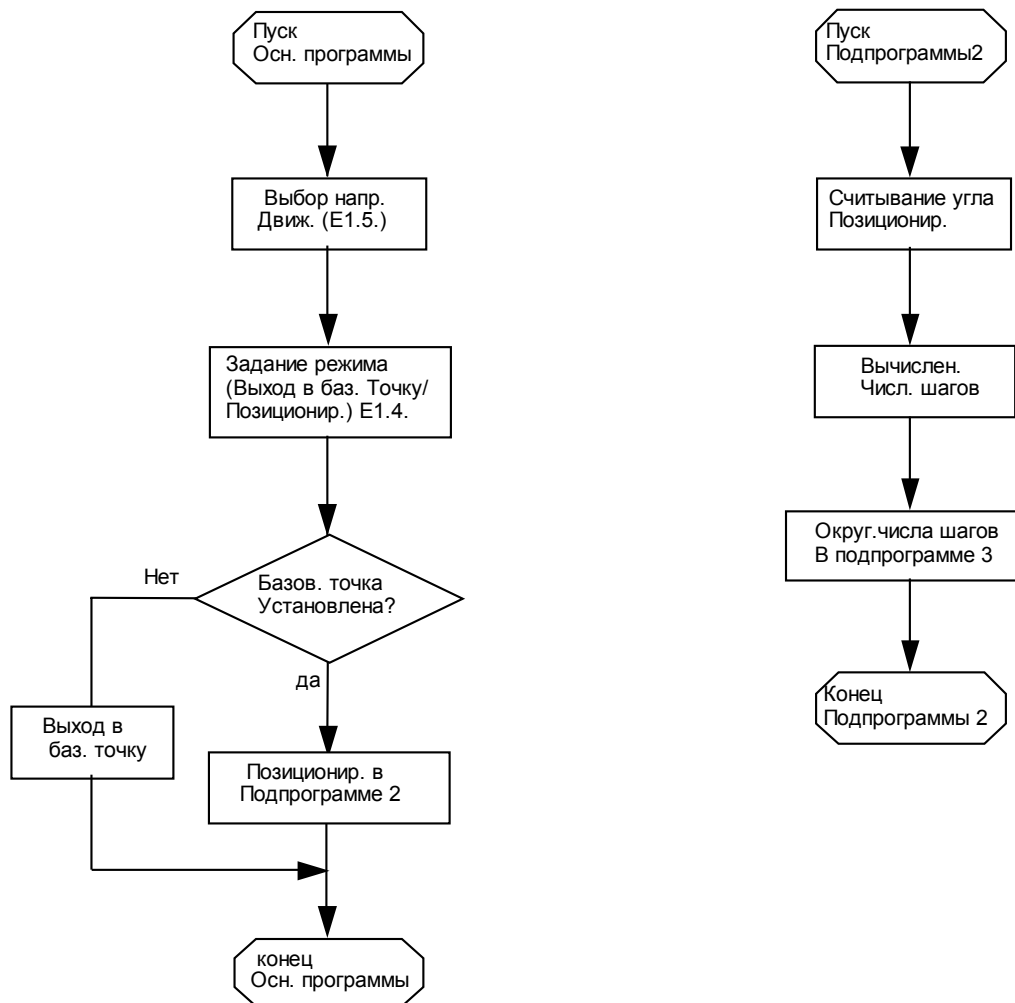
Elincom Group

 European Union: www.elinco.eu
 Russia: www.elinc.ru

Аппаратные средства

Кол-во	Приборы	Изготовитель/Заказной номер
1	SIMATIC S7-200 CPU 214	Siemens/ 6ES7 214 - 1AC00 - 0XB0
1	PC/PPI-Кабель	Siemens/ 6ES7 901 - 3BF00 - 0XA0
1	Программатор или PC	
1	Шаговый двигатель с соответствующими силовой частью и кабелем связи	
1	Кабель для управляющих сигналов к силовой части	
9	Переключатель	
3	Кнопка	

Структура программы



Описание программы вкл. листинг

В первом цикле программы (SM0.1=1) производится предустановка важнейших параметров. Выбор направления вращения и блокировка кнопок производится аналогично примеру применения No. 22.

Установка и стирание базовой точки

Если базовая точка ещё не установлена, то нажатие кнопки „ПУСК двигателя“ (E 1.0) запустит выход в базовую точку. ЦПУ выдаёт при этом максимальное возможное число управляющих импульсов. В желаемой базовой точке нажмите кнопку „Установка/стирание базовой точки“ (E1.4). При этом будет вызвана подпрограмма, останавливающая двигатель. В заключение меркер базовой точки M0.3 будет установлен в 1, а выход A1.0 покажет новый режим работы „Позиционирование активизировано“.

Если нажата кнопка E1.4 и активизировано „Позиционирование“ (M0.3=1), то происходит переключение в режим работы „Выход в базовую точку“. В подпрограмме при этом будет сброшен M0.3 и сброшена индикация „Позиционирование активизировано“. Дополнительно система управления будет подготовлена к выдаче максимального числа управляющих импульсов. Повторным нажатием кнопки E1.4 можно производить переключение с одного режима работы на другой. Если сигнал от кнопки приходит при работающем двигателе, то двигатель в любом случае будет остановлен.

На практике, вместо нажатия кнопки вручную, на привод жёстко крепится датчик базовой точки, позволяющий повторно выезжать в базовую метку.

Управляемое позиционирование

Если задана базовая точка (M0.3=1) и отсутствует блокировка (см. Пример No. 22), то будет проведено относительное позиционирование. Система управления считывает при этом в подпрограмме 2 угол позиционирования в двоичной форме из байта входов E0 в байт меркеров M0.11. Соответствующее этому углу число импульсов рассчитывается по следующей формуле:

$$n = \frac{\varphi}{360^\circ} * S.$$

n = число управляющих импульсов

(= угол перемещения в градусах

S = число шагов на оборот двигателя.

Используемый в примере шаговый двигатель работает в режиме полушага с S=1000. Вычисленное число шагов округляется затем в подпрограмме 3. Теперь если на E1.0 появится сигнал „ПУСК двигателя“, то ЦПУ выдаст на выход A0.0 вычисленное число управляющих импульсов и двигатель повернется на соответствующее число шагов. Для внутреннего маркирования состояния „Двигатель в движении“ меркер M 0.1 устанавливается в 1.

После полной выдачи импульсов будет обработана подпрограмма прерываний 0 (ср. Пример No. 22, стр. 3). M0.1 здесь сбрасывается, чтобы позволить новое включение привода. Как и в примере No. 22 этот шаг не включен в диаграмму работы, с тем чтобы оставить график наглядным.

Отключение привода

Вращение двигателя может быть, аналогично 1му примеру (стр. 3), в любой момент завершено нажатием кнопки „ОСТАНОВ двигателя“ (E1.1). Необходимые для этого команды находятся в Подпрограмме 0.

Размер программы составляет 155 слов.

Подробнее о последовательностях импульсов Вы найдёте в главе 6.3 “Быстрые операции вывода”. Подробную информацию о подпрограммах прерываний Вы найдёте в главе 6.2 “Подпрограммы прерываний” в руководстве по программированию SIMATIC S7-200.

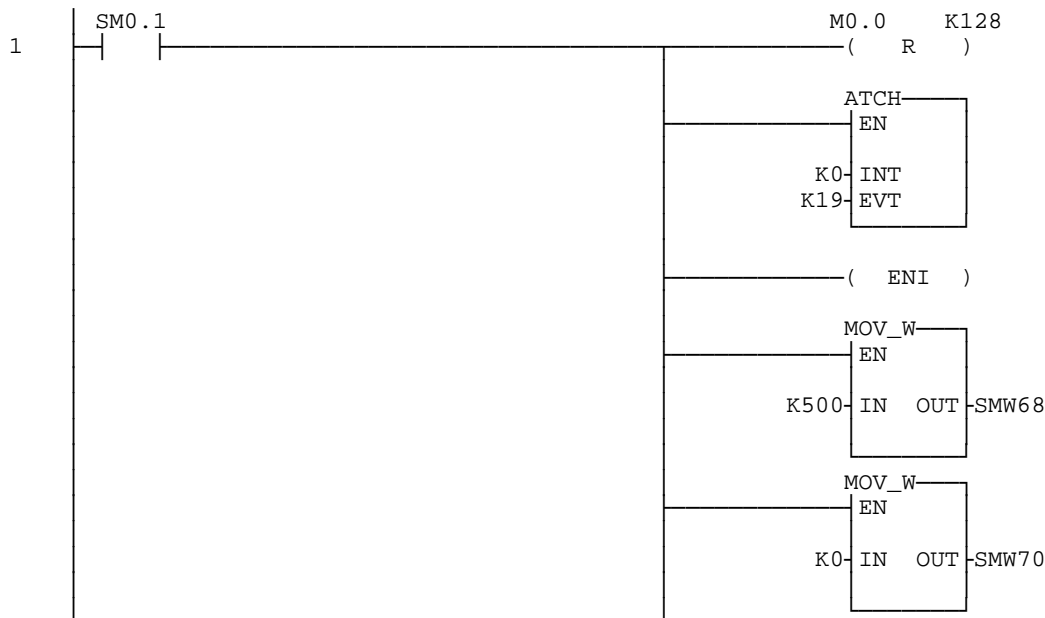
KOP (S7-MicroDOS)

AWL (TOOLITE2)

Основная программа

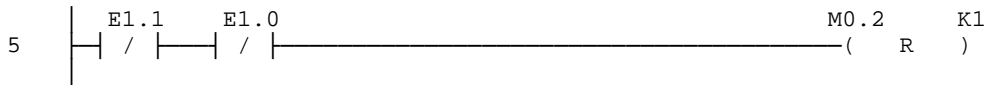
// TITEL = Управляемое позиционирование через импульсные выходы

// Входы:	E0.0 - E0.7	Угол позиционирования в градусах (двоичный код)
//	E1.0	Кнопка „ПУСК двигателя“
//	E1.1	Кнопка „ОСТАНОВ двигателя“
//	E1.4	Кнопка „Установить/стереть базовую точку“
//	E1.5	Переключатель направления движения
// Выходы:	A0.0	Импульсный выход
//	A0.2 (0=вправо/1=влево)	Сигнал направления движения
//	A1.0	Индикация режима работы
// Меркеры:	M0.1	Двигатель в движении
//	M0.2	Меркер блокировки
//	M0.3	Меркер базовой точки
//	MD8, MD12	Вспомогательный меркер



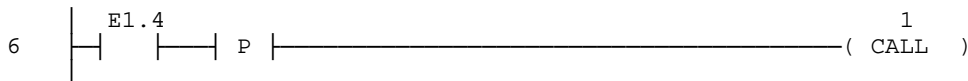
LD	SM0.1 // Устанавливается только в первом цикле
R	M0.0, 128 // Сброс MD0 - MD12
ATCH	0, 19 // Назначение подпрограммы прерываний 0 (Завершена последовательность импульсов)
ENI	// Прерывание деблокировано
MOVW	500, SMW68 // Длительность периода T = 500 мсек
MOVW	0, SMW70 // Ширина импульса = 0 при широтно-импульсной модуляции
MOVD	16#1999997C, SMD72 // макс. число импульсов для выхода в базовую точку
// Деблокировка вращения влево	
2	
LDN	M0.1 // Двигатель выключен
U	E1.5 // а переключатель направления движения = 1
S	A0.2, 1 // Деблокировка вращения влево
// Деблокировка вращения вправо	
3	
LDN	M0.1 // Двигатель выключен
UN	E1.5 // ... а переключатель направления движения = 0
R	A0.2, 1 // Деблокировка вращения вправо
// Активирование блокировки	
4	
LD	E1.1 // Нажата кнопка "ОСТАНОВ двигателя"
S	M0.2, 1 // Блокировка активирована

// Подавление блокировки



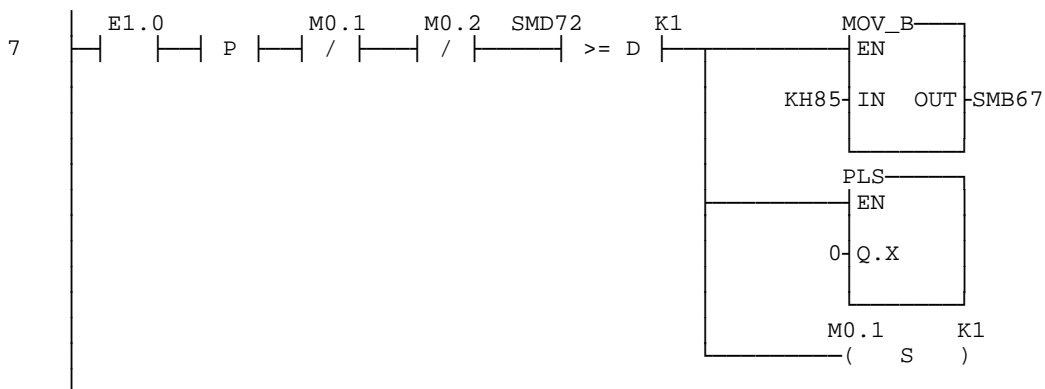
LDN E1.1
 UN E1.0 // Обе кнопки деблокированы
 R M0.2, 1 // Подавление блокировки

// Задание режима работ (Выход в базовую точку / Позиционирование)



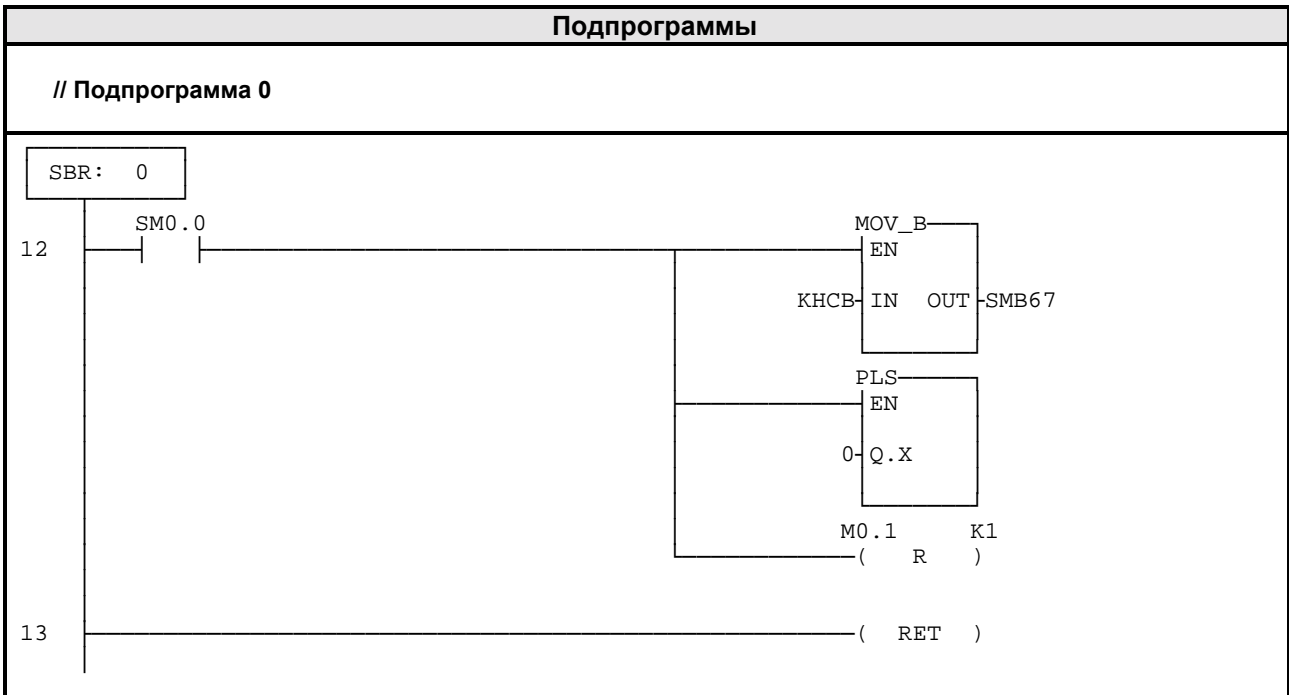
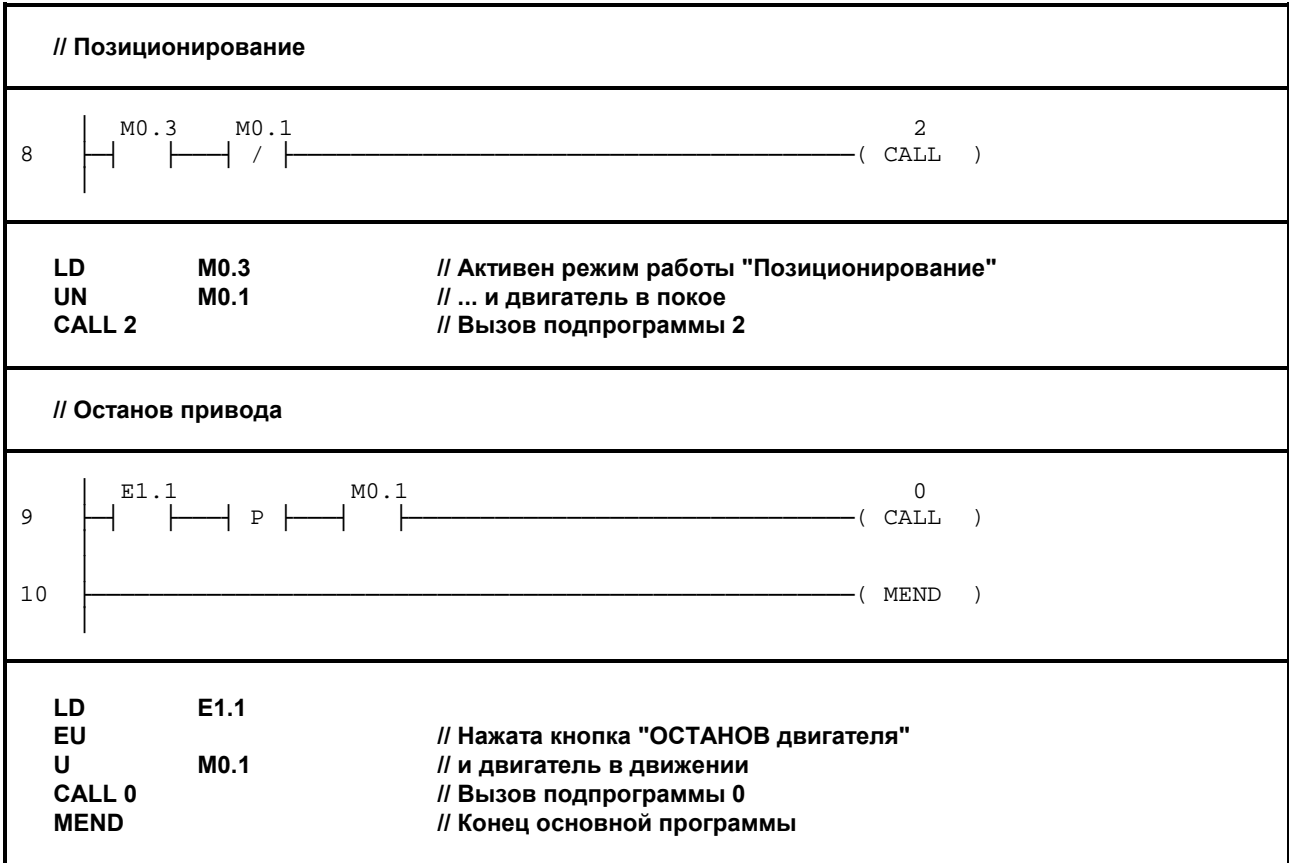
LD E1.4
 EU // Нажата кнопка "Установка/стирание базовой точки"
 CALL 1 // Вызов подпрограммы 1

// Запуск привода



LD E1.0
 EU // Нажата кнопка "ПУСК двигателя"
 UN M0.1 // ... и двигатель в покое
 UN M0.2 // ... и блокировка неактивна
 UD>= SMD72, 1 // и число шагов >= 1

MOVB 16#85, SMB67 // Активирование функции выдачи импульсов РТО
 PLS 0 // Пуск выдачи импульсов на А0.0
 S M0.1, 1 // Установка меркера "Двигатель в движении"



```

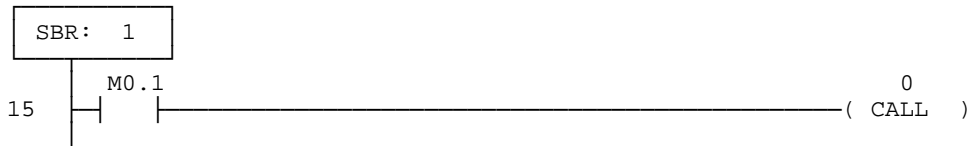
SBR 0           // Подпрограмма 0 "Останов привода"
LD              SM0.0
MOV            16#CB, SMB67 // Активирование широтно-импульсной модуляции
PLS           0           // Останов выдачи импульсов на A0.0
R             M0.1, 1     // Сброс меркера "Двигатель в движении"
RET

```

```

// Подпрограмма 1

```



```

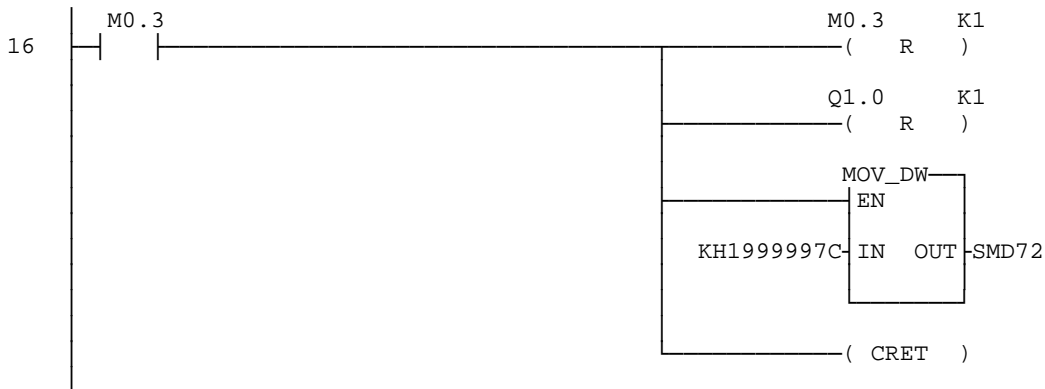
SBR 1           // Подпрограмма 1 "Задание режима работы"
LD              M0.1           // Двигатель в движении
CALL           0           // Останов привода

```

```

// Требование выхода в базовую точку

```



```

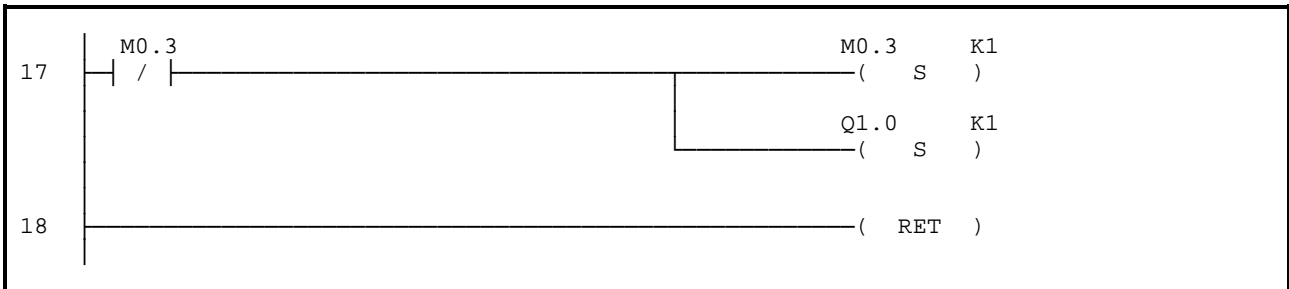
LD              M0.3           // "Позиционирование" активно
R              M0.3, 1        // Сброс меркера базовой точки
R              A1.0, 1        // Стереть сообщение "Позиционирование активно"
MOVD           16#1999997C, SMD72 // макс. число импульсов для нового выхода в баз.точку
CRET

```

```

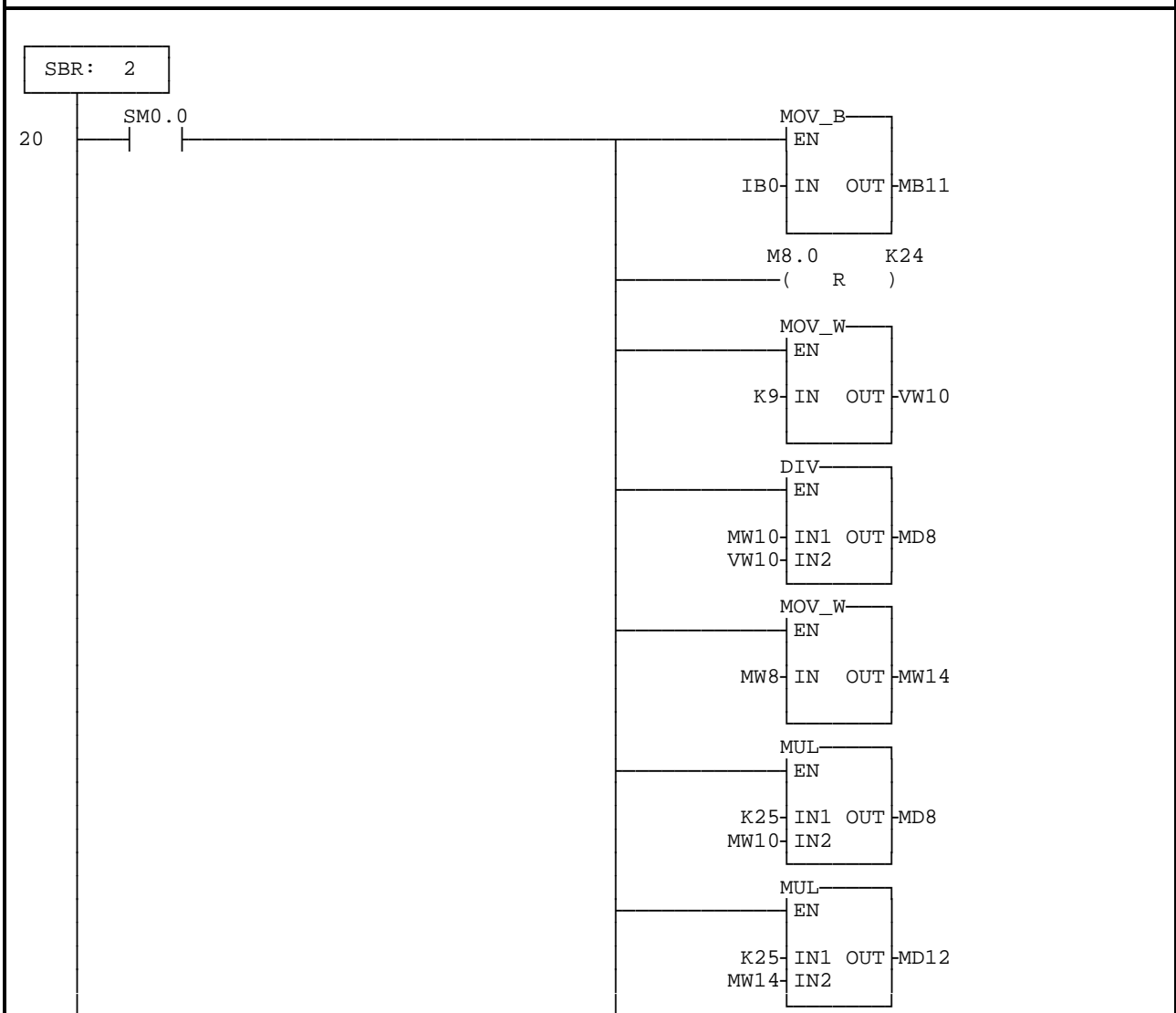
// Требование „Позиционирования“

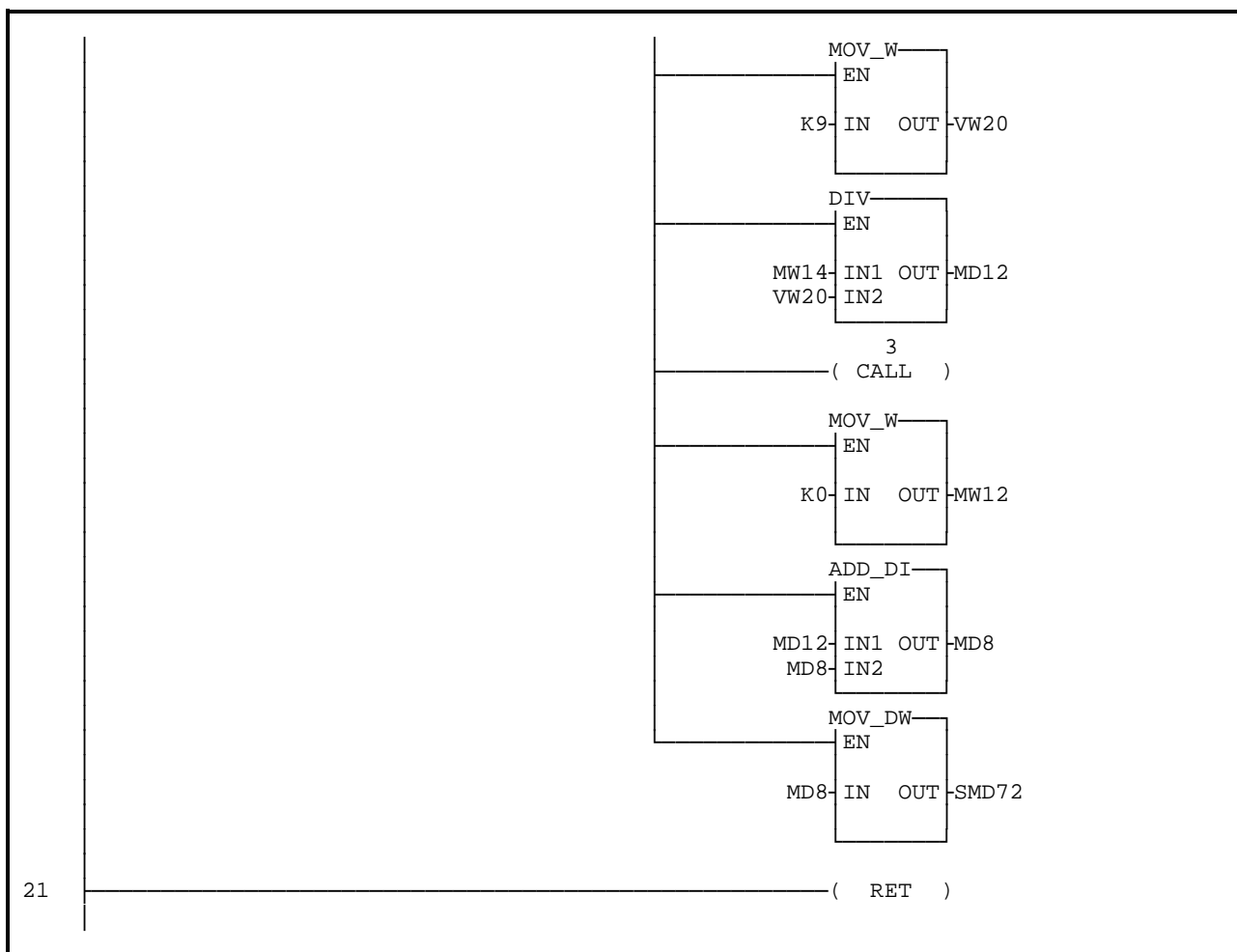
```

LDN M0.3 // Базовая точка еще не установлена
 S M0.3, 1 // Установить меркер базовой точки
 S A1.0, 1 // Вывод сообщения "Позиционирование активно"
 RET // Конец подпрограммы 1

// Подпрограмма 2



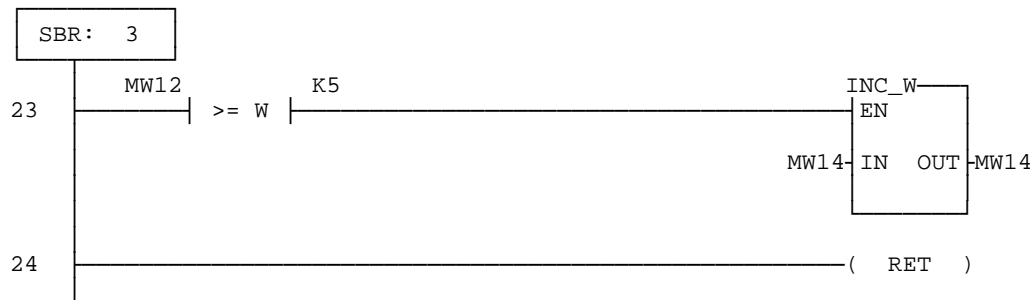


```

SBR 2          // Подпрограмма 2 "Позиционирование"
LD            SM0.0
MOVB         EB0, MB11 // Скопировать угол поворота из EB0 в MD8
R            M8.0, 24 // Стереть MB8 - MB10
MOVW        9, VW10
DIV         VW10, MD8 // Угол / 9 = q1 + r1
MOVW        MW8, MW14 // r1 занести в MD12
MUL         25, MD8 // q1*25 -> MD8
MUL         25, MD12
MOVW        9, VW20
DIV         VW20, MD12 // r1*25 / 9 = q2 + r2
CALL 3      // Округлить число шагов в подпрограмме 3
MOVW        0, MW12 // Стереть r2
+D         MD12, MD8 // Занести число шагов в MD8
MOVD        MD8, SMD72 // Передать число шагов в SMD72
RET

```

Подпрограмма 3



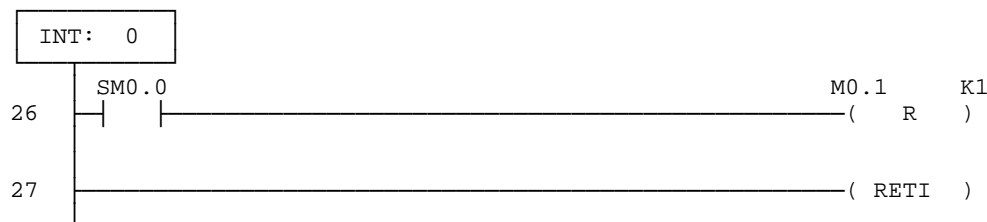
```

SBR 3           // Подпрограмма 3 "Округление числа шагов"
LDW>=   MW12, 5 // если r2 >= 5 / 9
INCW    MW14   // число шагов увеличить на 1
RET

```

Подпрограммы прерываний

// Подпрограмма прерываний 0 "Завершение выдачи импульсов"



```

INT 0           // Начало подпрограммы прерываний 0
LD      SM0.0
R       M0.1, 1 // Сброс меркера "Двигатель в движении"
RETI

```

Указания по преобразованию

Для того чтобы преобразовать TOOLITE2 AWL в S7-Micro/DOS AWL

- Установите 'K' перед каждым числом, не являющимся 16-ричной константой (напр. 4 ⇒ K4)
- Замените '16#' на 'KH' для всех 16-ричных констант (напр. 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 AWL в KOP-форму нужно начинать каждый сегмент словом 'NETWORK' и номером. Каждый сегмент в этом примере имеет свой номер на диаграмме KOP. Используйте NWENFG в меню редактора для ввода нового сегмента. Команды MEND, RET, RETI, LBL, SBR и INT требуют отдельных сегментов.

Общие указания

Примеры SIMATIC S7-200 предоставляются заказчику бесплатно. Данные примеры не привязаны к конкретной задаче и являются общей информацией о возможностях применения S7-200. Решение заказчика может отличаться от приведенного здесь.

За правильную работу системы заказчик несет ответственность сам. Мы обращаем Ваше внимание на действующие нормы Вашей страны и предписания по установке соответствующей системы. Ошибки и изменения возможны.