

SIMATIC

S7-200 Примеры

Группа

3

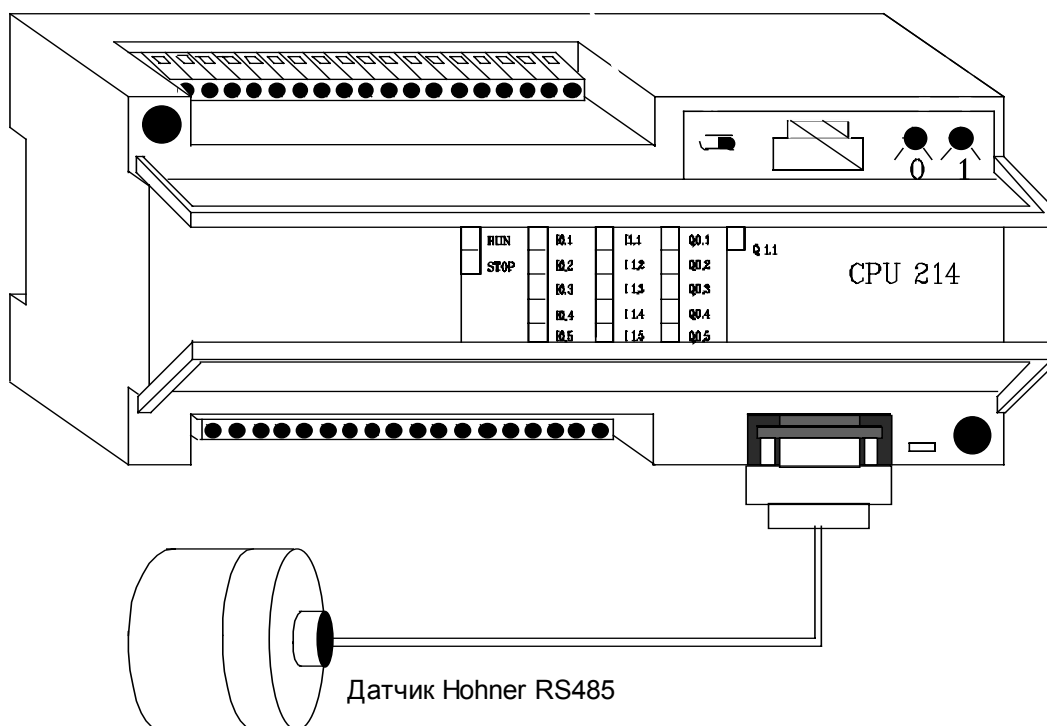
Тема

Подключение датчика RS485 к PPI интерфейсу S7-200

Краткое описание

Данный пример описывает как м.б. считано значение с датчика положения RS485 (производитель: Hohner Engineering) используя PPI интерфейс S7-200.

Схема



Industrial automation

Elincom Group

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

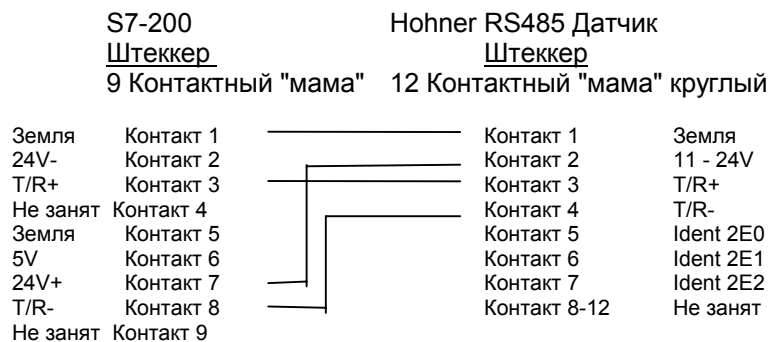
Описание аппаратуры

Существует два типа датчиков, имеющих разное разрешение:

1. Датчик с разрешением 13 бит: Тип SVE10-06213
2. Датчик с разрешением 25 бит: Тип SHE86-10214

Производитель: Hohner Electrical Engineering, D-78532 Tuttlingen/Germany

Теоретически возможно подключить 8 датчиков к интерфейсу RS485.

Распайка кабеля:

Проблемы с интерфейсом

При использовании датчика Hohner вместе с S7-200 могут возникнуть следующие проблемы:

1. Ограничение по току для PPI интерфейса S7-200.

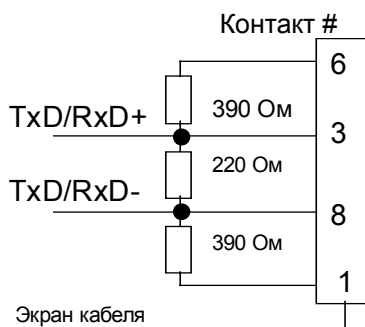
Допустимый ток не должен превышать 120mA, а для датчика Hohner необходимо 140mA. Поэтому датчик питается от отдельного 24V источника питания.

Возможно другое решение с использованием EMC-платы. Данная плата изолирует интерфейс RS485 от датчика. Напряжение питания при этом для интерфейса S7-200 не превышает 20%, или < 30mA.

2. Помехи от датчика.

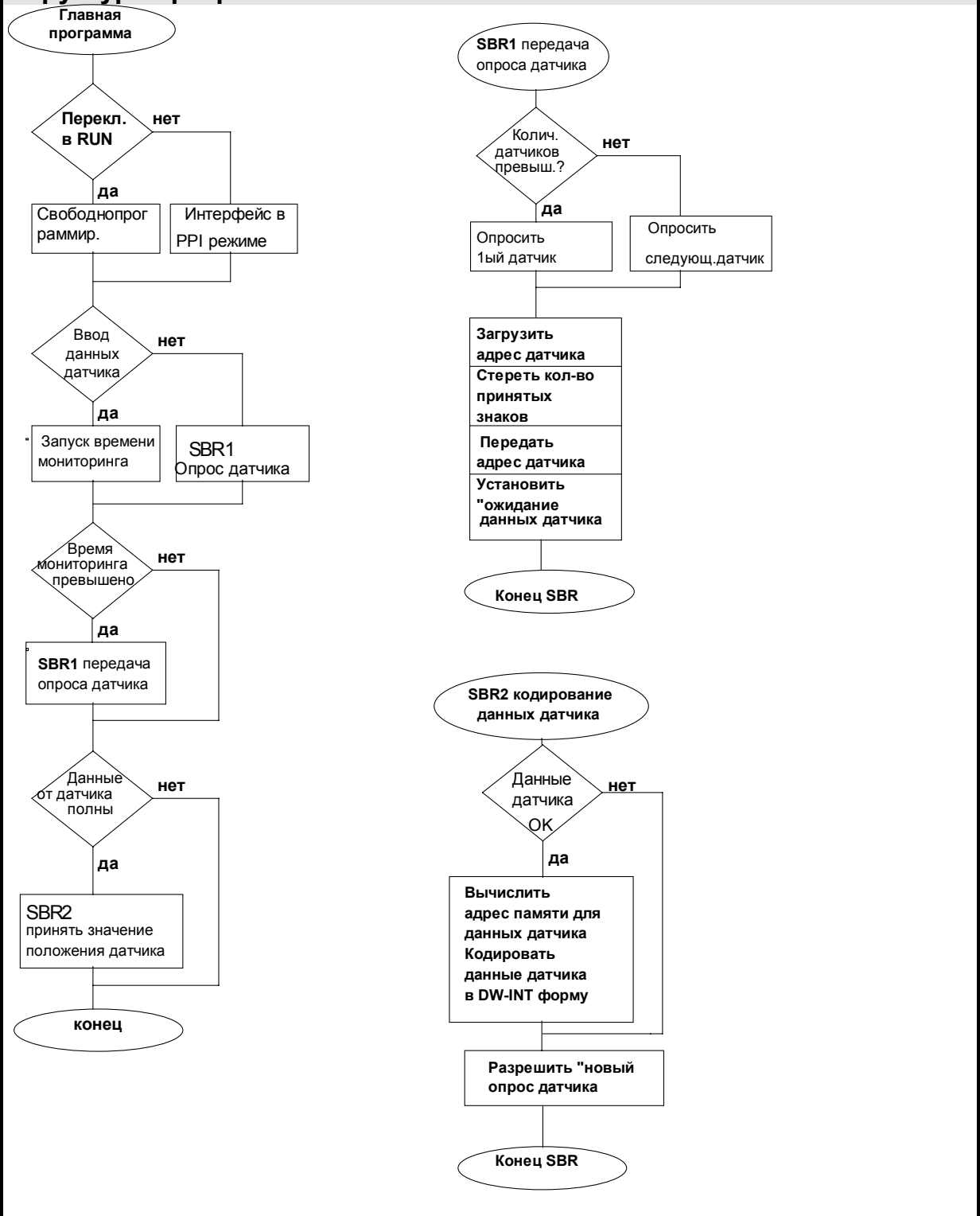
Уровень помех от датчика м.б. уменьшен при использовании терминатора:

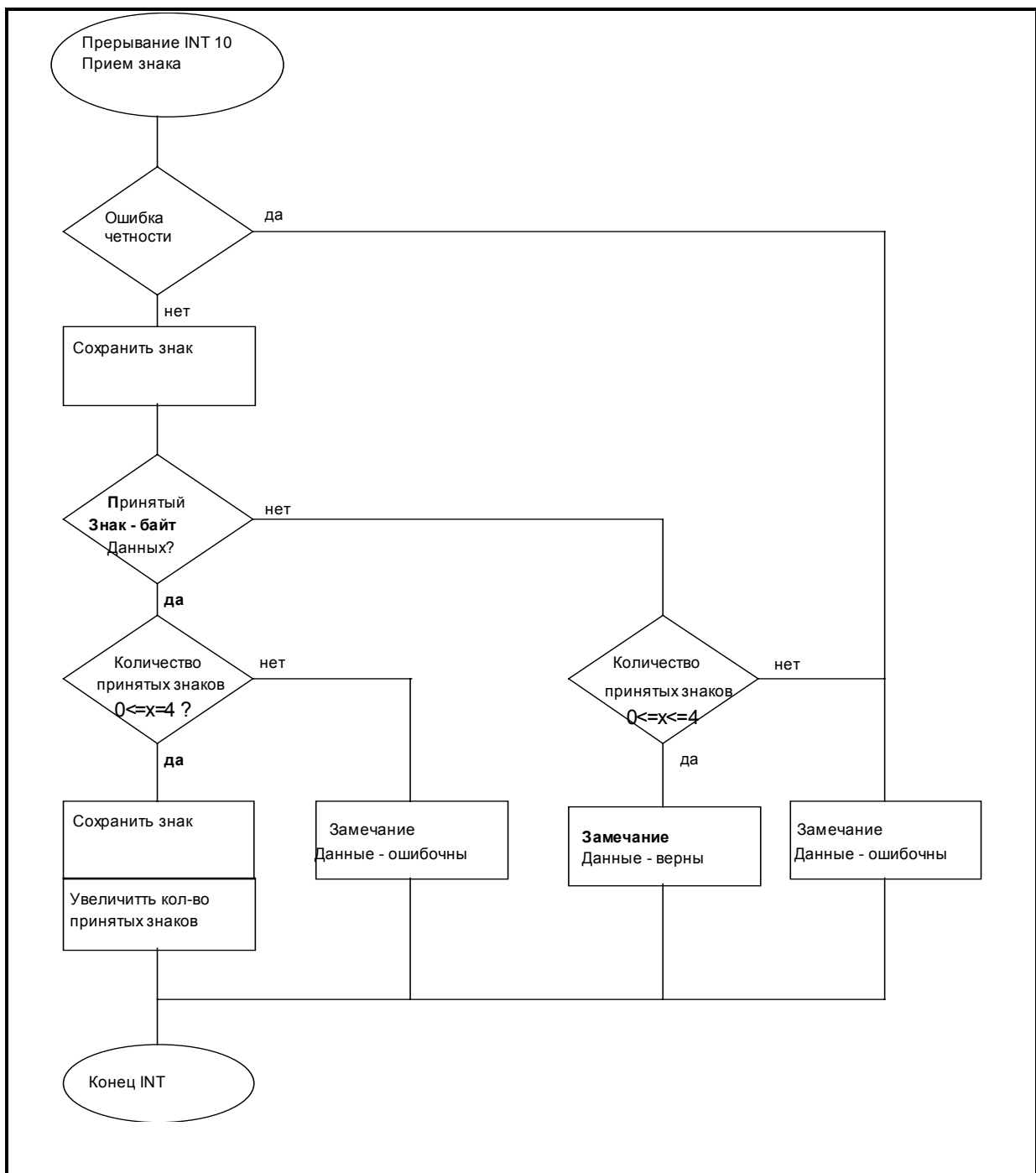
6ES7-972-0B00-0XA0, или комбинации резисторов, как показано на рисунке ниже:

**3. Слишком короткое время ожидание между двумя передаваемыми словами.**

Время ожидания между двумя словами для датчика Hohner равно только 1 бит времени. Это м.б. слишком маленьким промежутком времени для CPU 212 или CPU 214 для того, чтобы сбросить блок UART. UART не будет готов для приема следующего знака после 2 бит времени и может возникнуть ошибка четности. Данная проблема м.б. решена путем увеличения времени ожидания датчика до 2 бит времени.

Структура программы





Описание программы

// TITLE = КОММУНИКАЦИЯ

// Инициализация:

Во время запуска - Подпрограмма 0 - устанавливаются параметры для PPI интерфейса, скорость передачи и кадр символа. Дополнительно устанавливается адрес датчика RS485.

Используемые данные:

Тип	Адрес	Описание
VD	200	Указатель на буфер передачи
VB	204	Количество байт данных для передачи (1)
VB	205	Передаваемый байт
VB	210	Количество принятых переданных байт
VB	211	Текущий принятый Байт 1
VB	212	Текущий принятый Байт 2
VB	213	Текущий принятый Байт 3
VB	214	Текущий принятый Байт 4
VB	215	Текущий Номер Датчика
VB	216	Максимальное количество датчиков
VB	220	Шинный Адрес Первого Датчика
VB	221	Чтение Значения в MSB
VB	222	Чтение Значения в
VB	223	Чтение Значения в
VB	224	Чтение Значения в LSB
VB	225	Шинный Адрес Второго Датчика
VB	226	Чтение Значения в MSB
VB	227	Чтение Значения в
VB	228	Чтение Значения в
VB	229	Чтение Значения в LSB
..		
VB	255	Шинный Адрес Восьмого Датчика
VB	256	Чтение Значения в MSB
VB	257	Чтение Значения в
VB	258	Чтение Значения в
VB	259	Чтение Значения в LSB
M	0.0	Чтение данных датчика активно
M	0.1	Данные верны
M	0.2	Принятые ошибочные данные <1 байта, >4 байт и Ошибка Четности
M	0.3	Вспомогательный Меркер <1 байта, >4 байт и Ошибка Четности
T	33	Время мониторинга; датчик не откликается 0.1 сек.
SM	0.0	Всегда "1"
SM	0.1	Завершить перезапуск
SM	0.7	Переключатель в положении "RUN"
SMB	2	Принятый байт последовательного интерфейс
SM	3.0	Ошибка Четности
SM	4.0	Переполнение очереди
SM	4.4	Разрешить прерывания
SM	4.5	Передатчик в ожидании
SMB	30	Контрольный Байт Интерфейса RS485

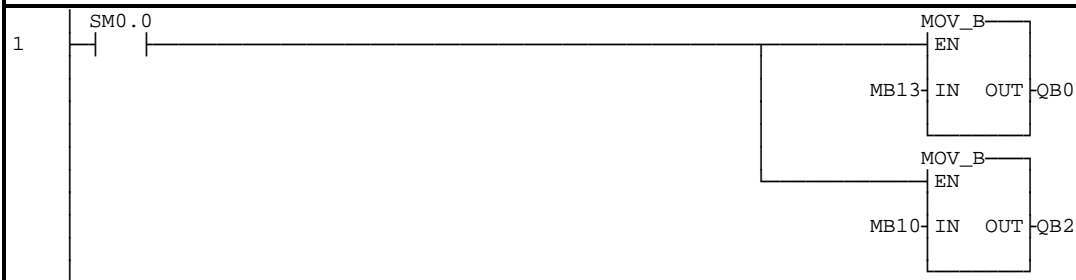
SM	SM	SM	Значение	Описание
30.7	30.6		00	Без контроля четности
			01	Контроль на четность
			10	Без контроля четности
			11	Контроль на нечетность
30.5		0	8 бит на знак	
		1	7 бит на знак	
30.4	30.3	30.2	000	38.4KB (214) / 19,2KB (212)
			001	19.2KB
			010	9,6 KB
			011	4.8 KB
			100	2,4 KB
			101	1.2 KB
			110	0.6 KB
			111	0.3 KB
30.1	30.0		00	PPI протокол
			01	Свободнопрограммируемый протокол
			10/11	PPI протокол

Тип	Прерывание	EVENT	Описание
-----	------------	-------	----------

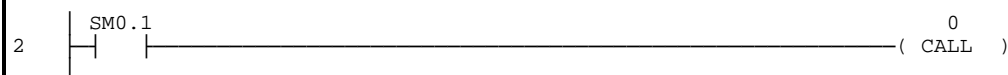
INT	--	9	Прерывание передачи данных
INT	10	8	Прием Данных

Тип	Адрес	Описание
SBR	0	Инициализация
SBR	1	Опрос передающего датчика
SBR	2	Сохранить данные датчика

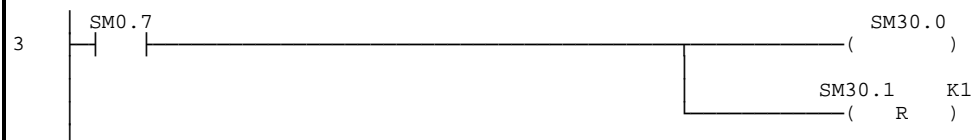
```
// Подключен датчик Hohner-RS485 с разрешением 13 и 25 бит
// данных на байт, 7 бит используются как биты данных.
// Выходы Q0.0 - Q1.1 установлены на 13 бит разрешения
//
// Данные проверяются на:
// - Сумму установленных битов, получаемых с контрольным байтом
// - Ошибка Четности
```



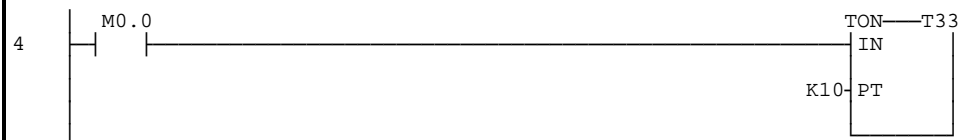
```
LD SM0.0 // Каждый цикл
MOV MB13 QB0 // Скопировать байт памяти13 в Бит выходов 0
MOV MB10 QB2 // Скопировать байт памяти10 в Бит выходов 2
```



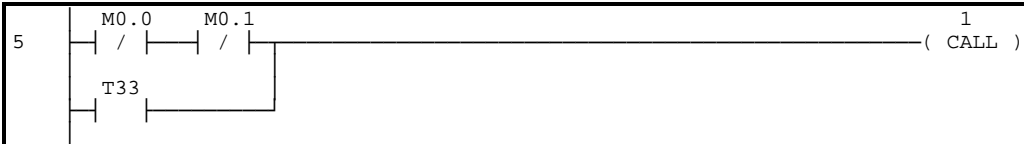
```
// Инициализация коммуникации
LD SM0.1 // 1ый цикл программы
CALL 0 // Вызов Подпрограммы инициализации
```



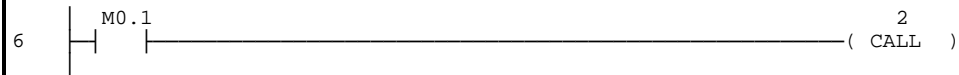
```
// Установить интерфейс PPI или свободнопрограммируемый интерфейс
LD SM0.7 // Если RUN, установить свободнопрограммируемый
// интерфейс
= SM30.0 // Если TERM, установить PPI
R SM30.1,1 // 00 = PPI ; 01 = свободнопрограммируемый
// интерфейс
```



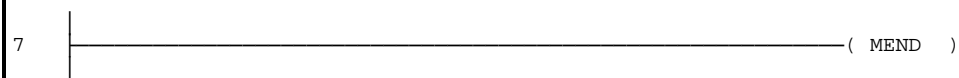
```
// Запуск таймера
LD M0.0 //Повторить опрос датчика, если он
TON T33,10 //не отвечает в течении 0.1 сек
```



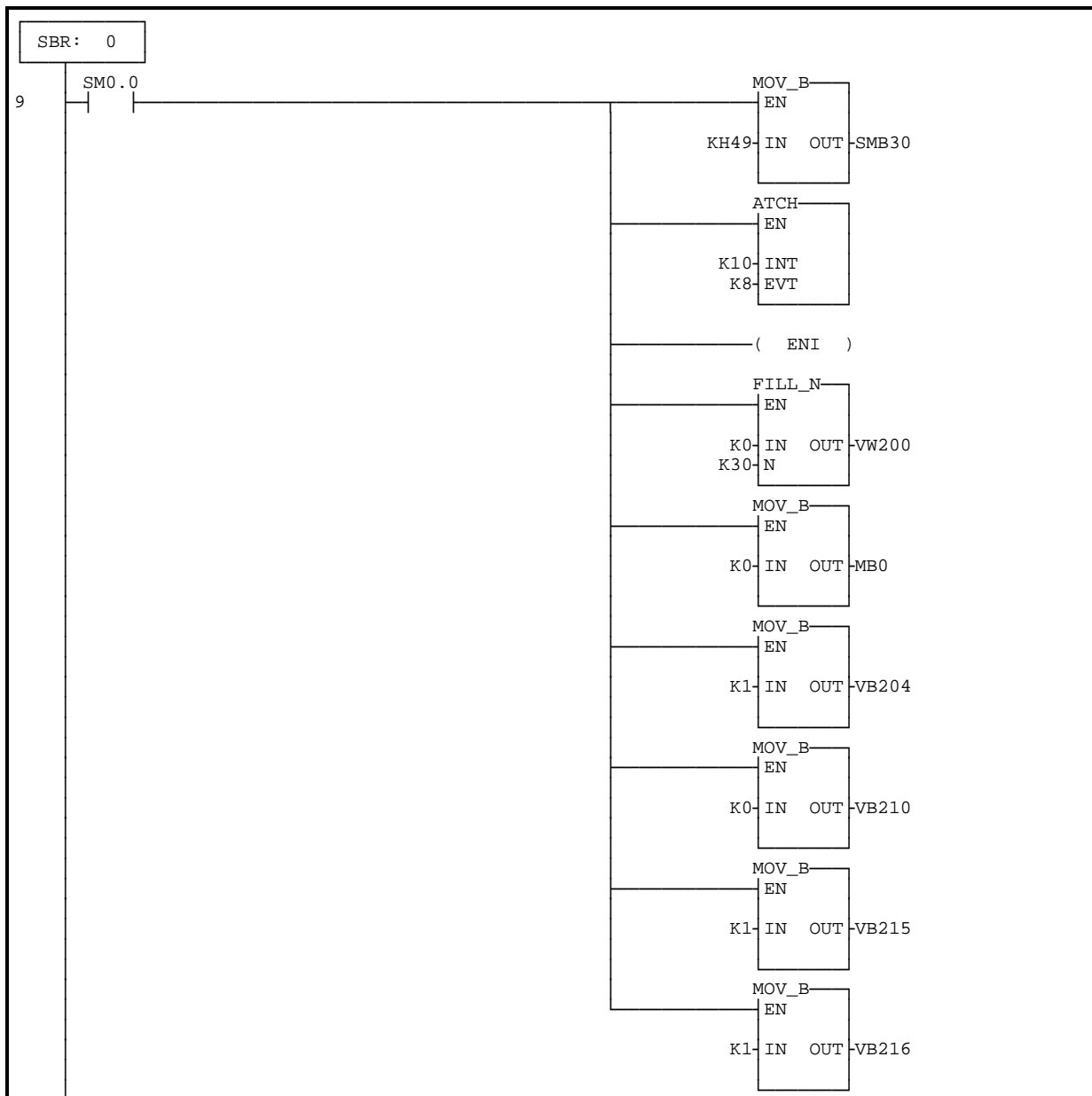
```
// Передача
LDN M0.0           // не M 0.0 - прочитанные данные датчика активны
O T33             // ИЛИ T33 активен
AN M0.1           // НЕ M 0.1 - данные датчик верны
CALL 1            // заполнить и передать буфер передачи
```



```
// Кодирование принятых данных
LD M0.1           // Данные датчика верны
CALL 2            // Сохранить принятые данные
```



```
MEND // ***** Конец главной программы *****
```

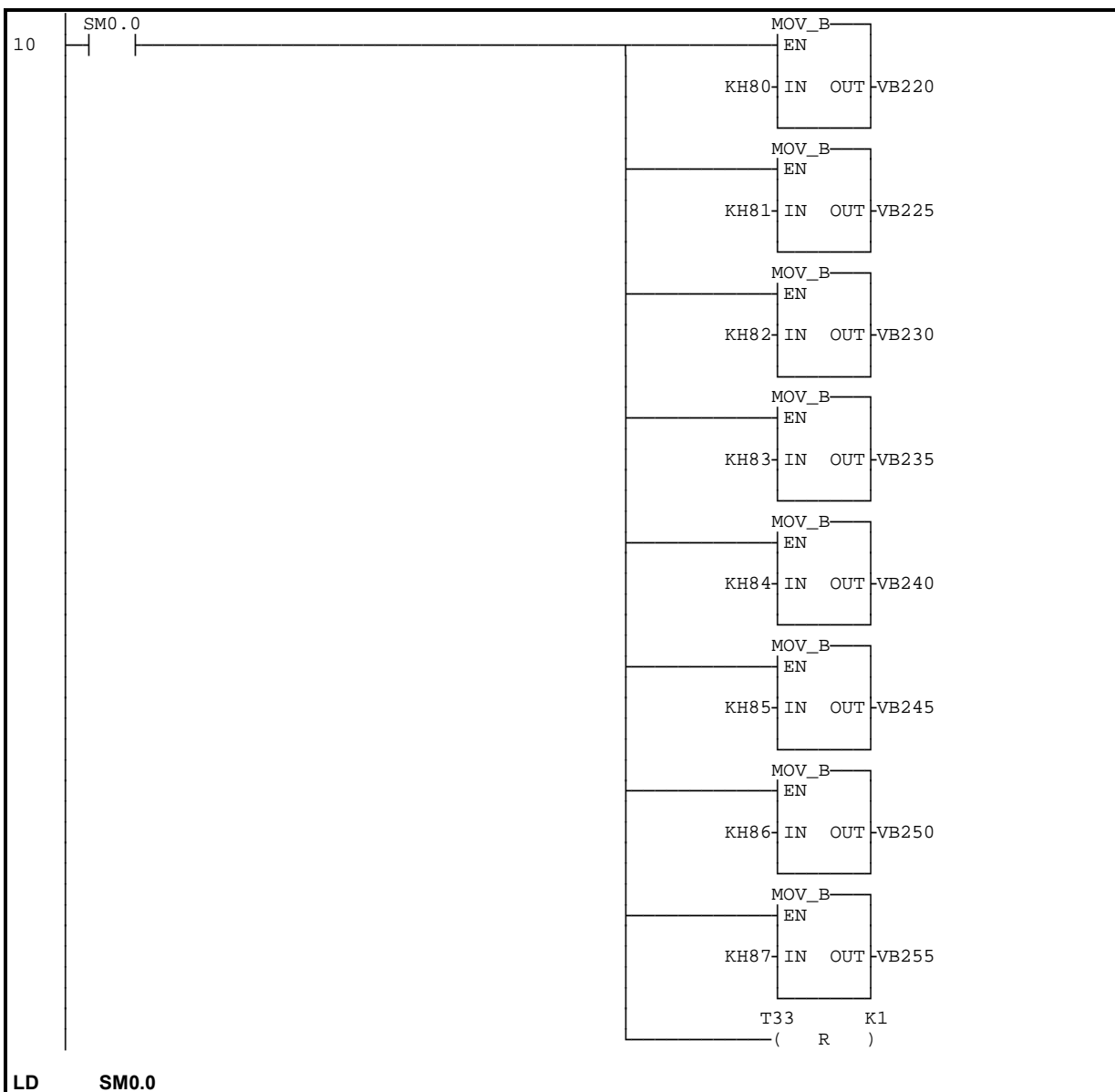
// Инициализация передачи

```
SBR 0
LD SM0.0
MOVB 16#49,SMB30
```

```
ATCH 10,8
ENI
FILL 0,VW200,30
MOVB 0,MB0
MOVB 1,VB204
MOVB 0,VB210
MOVB 1,VB215
MOVB 1,VB216
```

// ПОДПРОГРАММА 0

```
// RLO=1
// 9600 Бод, с контролем четности, 8бит/знак
// Бит 7/6 01 - с контролем четности
// Бит 5 0 - 8 бит/знак
// Бит 4/3/2 = 010 - 9.6 кВ
// Бит 4/3/2 = 001 - 19.2 кВ
// Бит 1/0 = 01 - свободнопрограммируемый интерфейс
// Знак приема INT 10 на EVENT 8
// Разрешить прерывания
// Очистить VW200-VW258
// Очистить Меркер
// Количество переданных байт = 1
// Знак счетчика CC = 0
// Текущий Указатель Датчика = 1
// Номер Датчика = 1
```



LD SM0.0

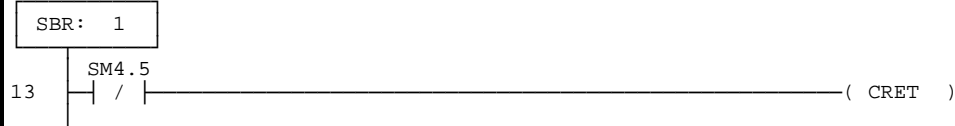
```

MOV_B 16#80,VB220 // Ввести Адрес для 1го Датчика
                // 1ый Номер Датчика = 0
MOV_B 16#81,VB225 // Ввести Адрес для 2го Датчика
                // 2ой Номер Датчика = 1
MOV_B 16#82,VB230 // Ввести Адрес для 3го Датчика
                // 3ий Номер Датчика = 2
MOV_B 16#83,VB235 // Ввести Адрес для 4го Датчика
                // 4ый Номер Датчика = 3
MOV_B 16#84,VB240 // Ввести Адрес для 5го Датчика
                // 5ый Номер Датчика = 4
MOV_B 16#85,VB245 // Ввести Адрес для 6го Датчика
                // 6ой Номер Датчика = 5
MOV_B 16#86,VB250 // Ввести Адрес для 7го Датчика
                // 7ой Номер Датчика = 6
MOV_B 16#87,VB255 // Ввести Адрес для 8го Датчика
                // 8ой Номер Датчика= 7
R      T33,1      // Сбросить таймер

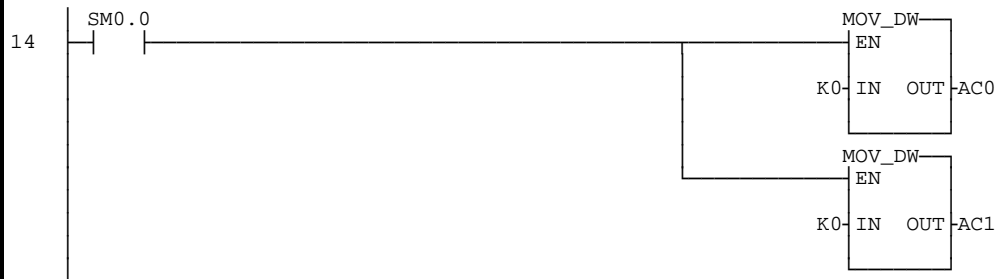
```



RET // Конец инициализации



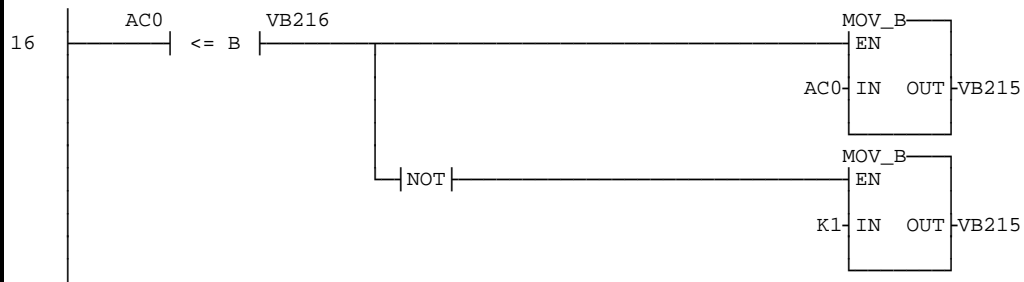
// Передача опроса Датчика
SBR 1 // *** ПОДПРОГРАММА 1 ***
LDN SM4.5 // Обрабатывается передатчик
CRET // Конец



LD SM0.0 // RLO=1
MOVD 0,AC0 // Очистить ACCU0
MOVD 0,AC1 // Очистить ACCU1



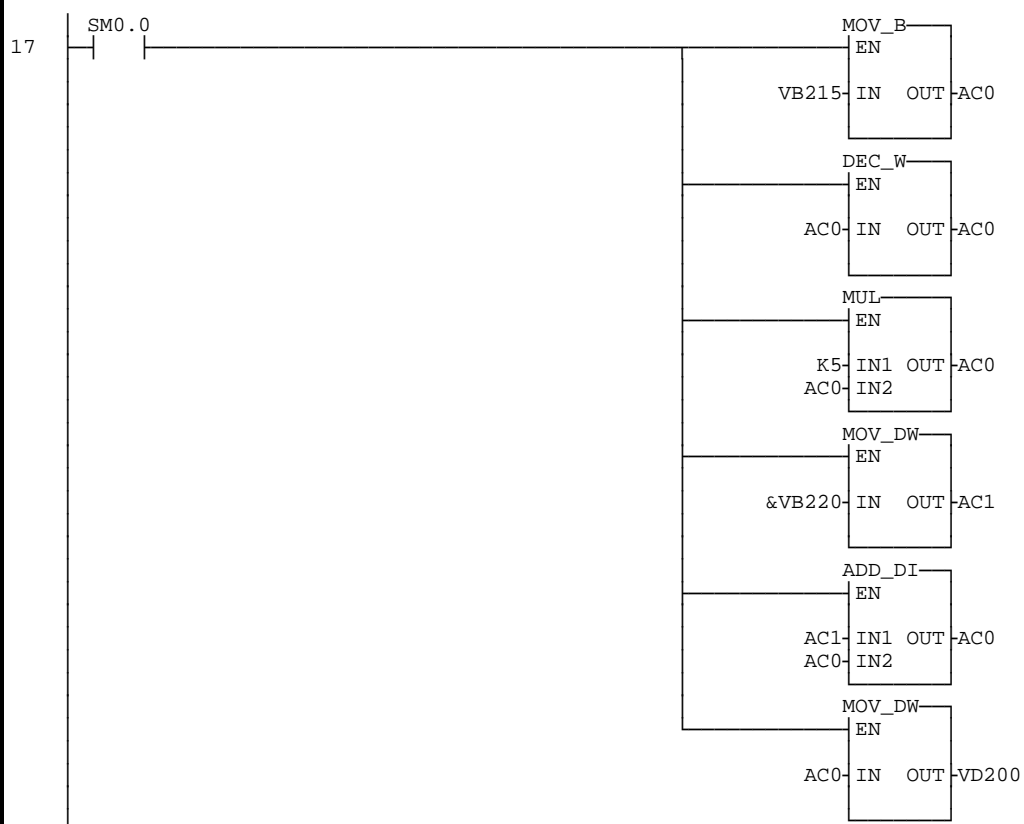
LD SM0.0 // RLO=1
MOVVB VB215,AC0 // Увеличить указатель на адрес датчика
INCW AC0



```

LDB<= AC0,VB216           // Номер Датчика превышен ?
MOV_B AC0,VB215           // Сохранить новый указатель на адрес датчика
NOT
MOV_B 1,VB215             // иначе, установить указатель на адрес датчика
                           // на нижнее граничное значение

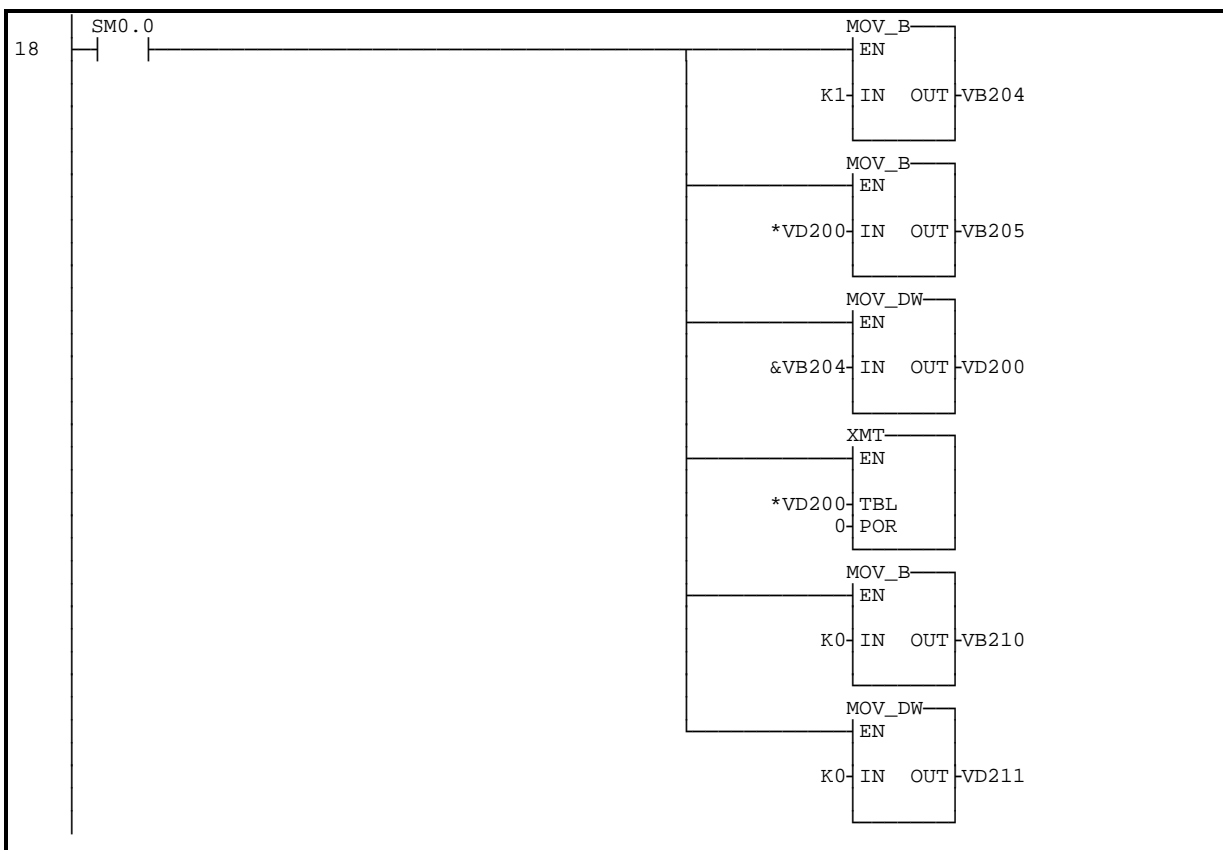
```



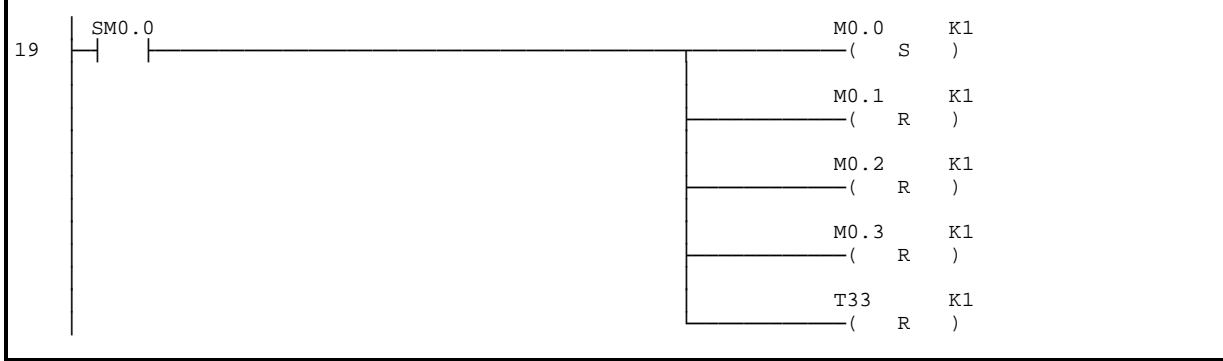
```

// Вычислить указатель на адрес датчика
LD      SM0.0              // RLO=1
MOV_B  VB215,AC0          // ((тек. № датчика - 1) * 5) + базовый адрес (VB220)
DECW   AC0                //
MUL    5,AC0              //
MOVD   &VB220,AC1
+D     AC1,AC0
MOVD   AC0,VD200

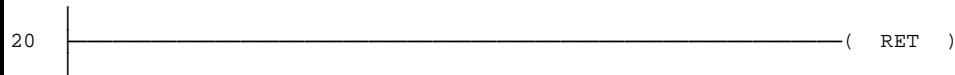
```



```
// Загрузить переданные данные (адрес датчика)
LD SM0.0 // RLO=1
MOVB 1,VB204 // Количество переданных байт = 1
MOVB *VD200,VB205 // Загрузить переданные данные
MOVD &VB204,VD200 // Установить указатель передачи на буфер передачи
XMT *VD200,0 // Передача
MOVB 0,VB210 // Сбросить счетчик символов
MOVD 0,VD211 // Очистить буфер приема
```



```
LD SM0.0 // RLO=1
S M0.0,1 // Установить "чтение данных датчика активно"
R M0.1,1 // Сбросить " данные датчика верны"
R M0.2,1 // Сбросить " данные датчика ошибочны"
R M0.3,1 // Сбросить "прием ОК"
R T33,1 // Сбросить таймер
```



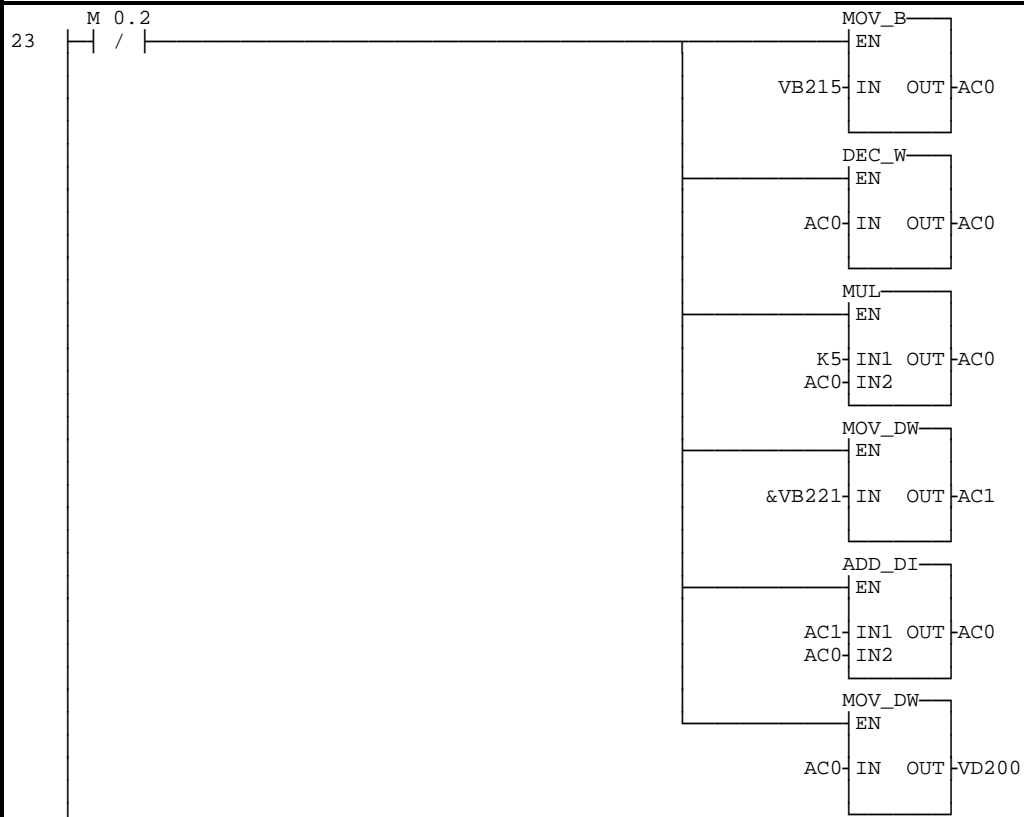
RET // Конец ПОДПРОГРАММЫ 1



// Сохранить данные датчика
 // Первым приходит Младший Байт, затем Старший Байт
 // В каждом байте используются Биты 0-6

//V211.0....V214.3 = 25 Биты Датчик

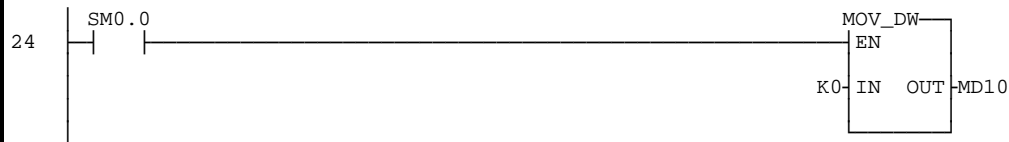
SBR 2 // ПОДПРОГРАММА 2
 LD SM0.0 // RLO = 1
 MOVD 0,AC0 // Очистить ACCU 0
 MOVD 0,AC1 // Очистить ACCU 1



```

LDN M0.2           // Данные датчика верны?
                   // Подсчитать адрес памяти
MOVW VB215,AC0     // ((тек. № датчика - 1) * 5) + Базовый Адрес (VB221)
DECW AC0
MUL 5,AC0
MOVD &VB221,AC1
+D AC1,AC0
MOVD AC0,VD200

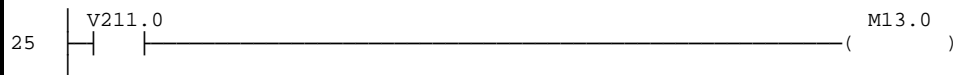
```



```

LD SM0.0           // RLO=1
MOVD 0,MD10       // Очистить Вспомогательный Меркер

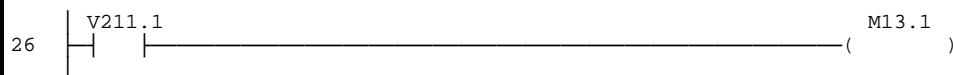
```



```

LD V211.0         // "Скомпоновать" данные датчика
= M13.0

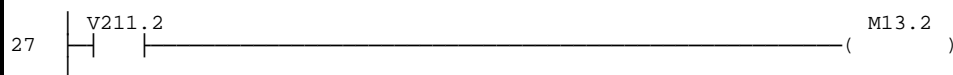
```



```

LD V211.1
= M13.1

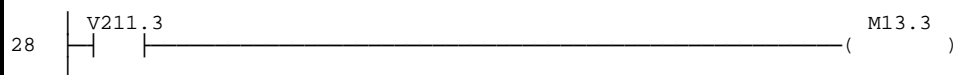
```



```

LD V211.2
= M13.2

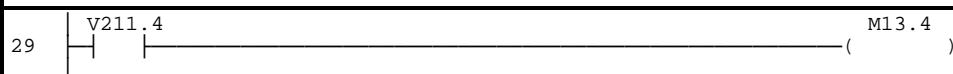
```



```

LD V211.3
= M13.3

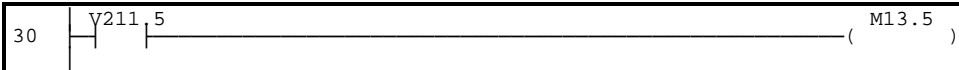
```



```

LD V211.4
= M13.4

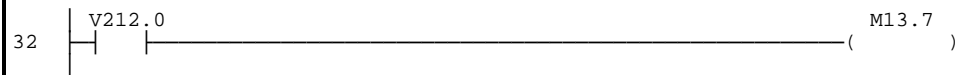
```



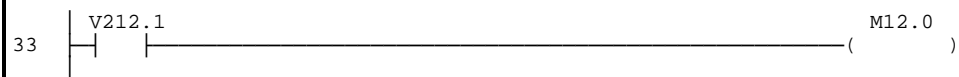
LD V211.5
= M13.5



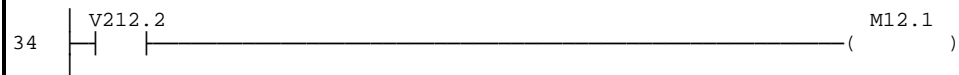
LD V211.6
= M13.6



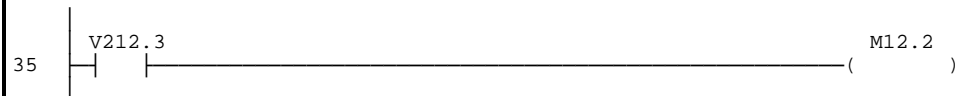
LD V212.0
= M13.7



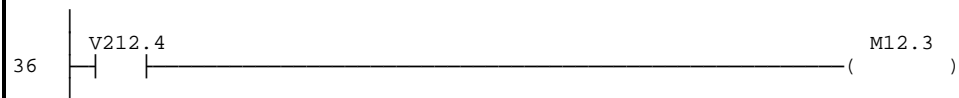
LD V212.1
= M12.0



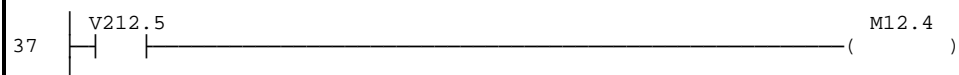
LD V212.2
= M12.1



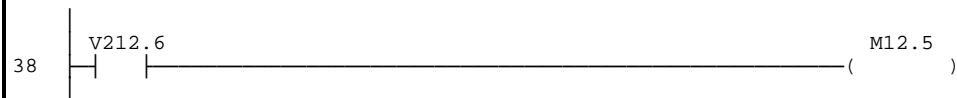
LD V212.3
= M12.2



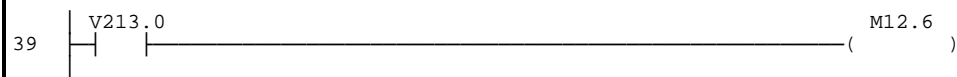
LD V212.4
= M12.3



LD V212.5
= M12.4



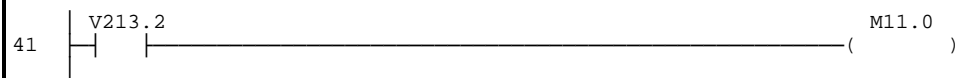
LD V212.6
= M12.5



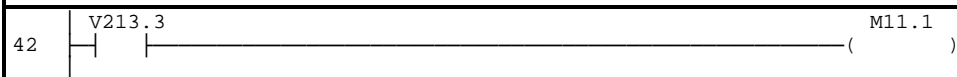
LD V213.0
= M12.6



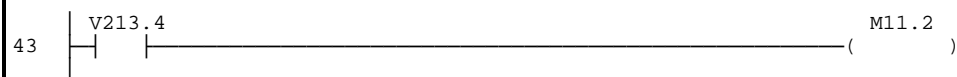
LD V213.1
= M12.7



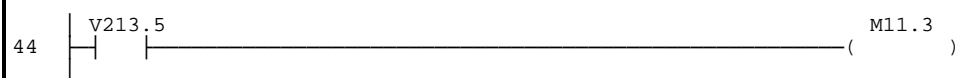
LD V213.2
= M11.0



LD V213.3
= M11.1



LD V213.4
= M11.2

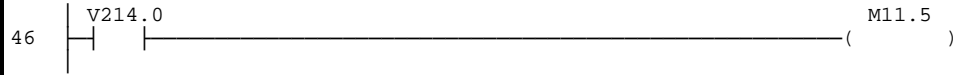


LD V213.5

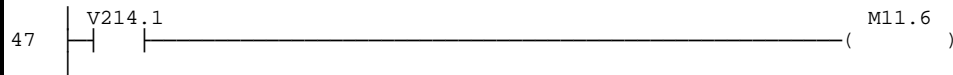
= M11.3



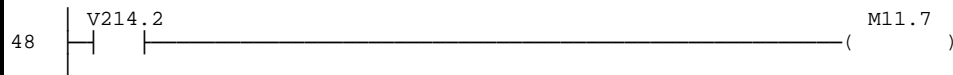
LD V213.6
= M11.4



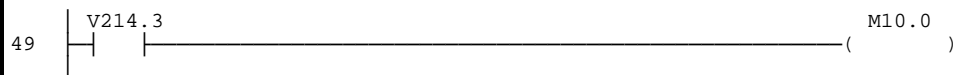
LD V214.0
= M11.5



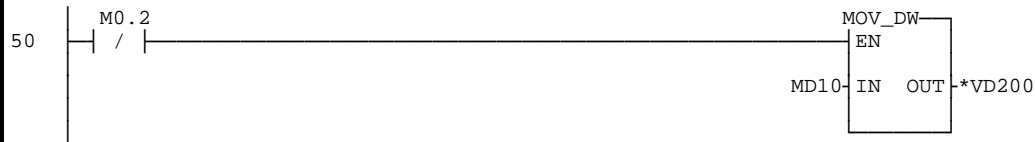
LD V214.1
= M11.6



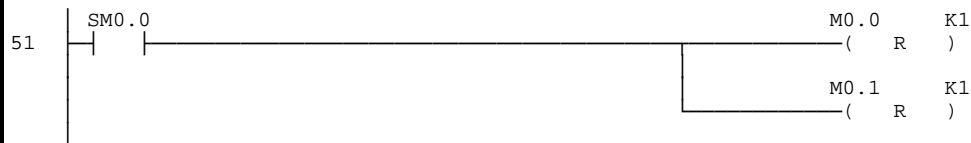
LD V214.2
= M11.7



LD V214.3
= M10.0



LDN M0.2 // Данные Датчика верны ?
MOVD MD10,*VD200 // Сохранить данные датчика

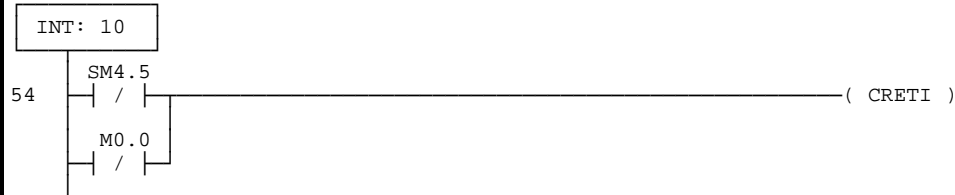


// Разрешить новый опрос датчика

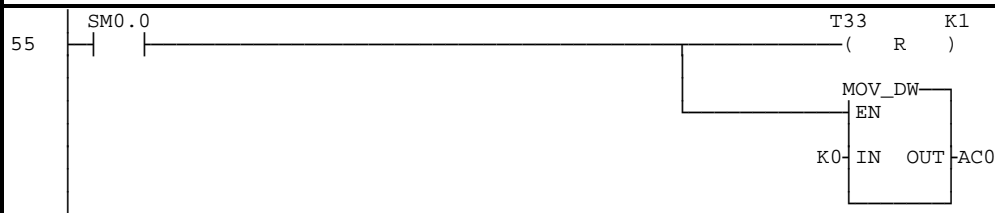
```
LD SM0.0 // RLO=1
R M0.0,1 // Установить "чтение данных датчика активно"
R M0.1,1 // Сбросить " данные датчика верны"
```

52 |-----| (RET)

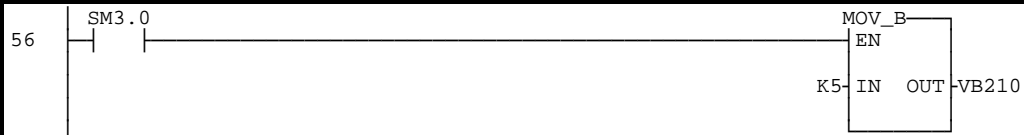
RET // Конец ПОДПРОГРАММЫ 2



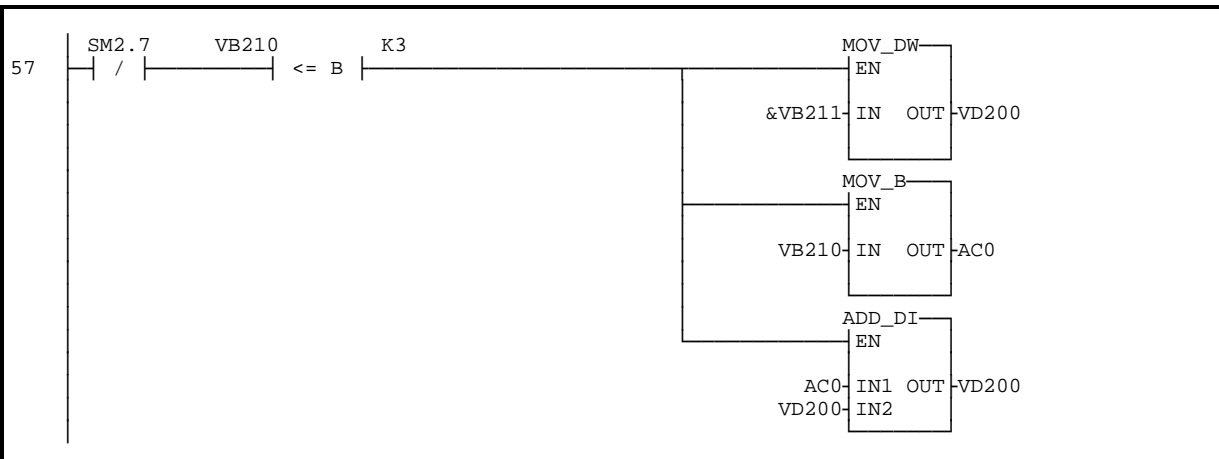
```
// Программа прерывания для приема
INT 10 // *** INT 10 ***
LDN SM4.5 // Обрабатывается Передача
ON M0.0 // данные ожидаются ?
CRETI // Конец
```



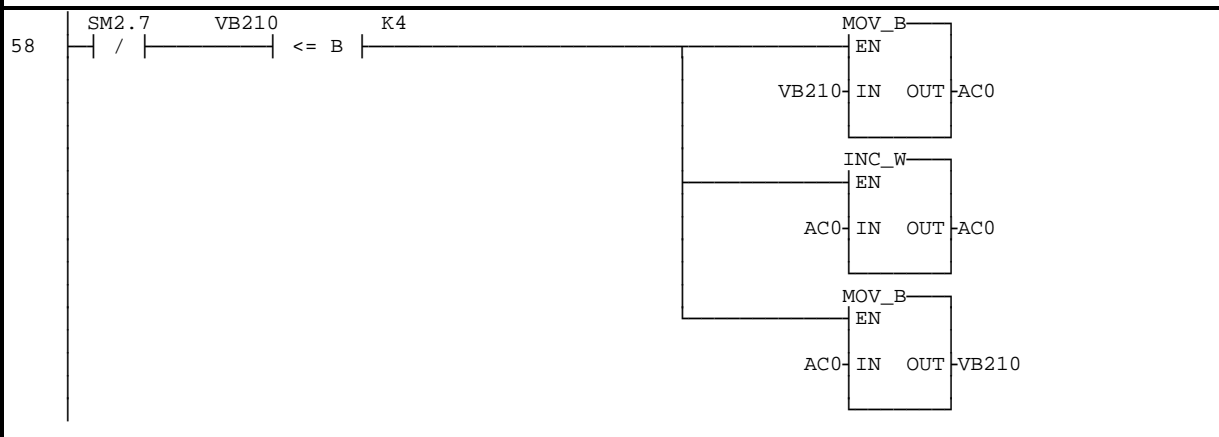
```
LD SM0.0 // RLO = 1
R T33,1 // Сбросить таймер
MOVD 0,AC0 // Очистить ACCU
```



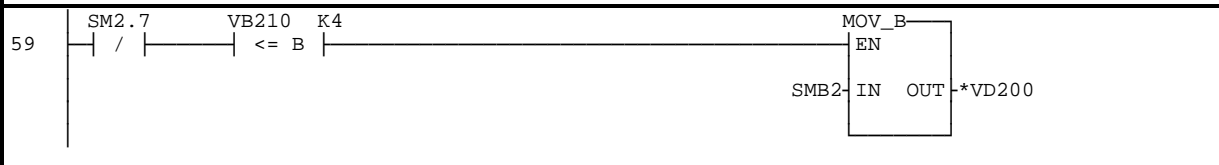
```
LD SM3.0 // Ошибка Четности
MOVB 5,VB210 // Установить счетчик символов
// в недействительное значение
```



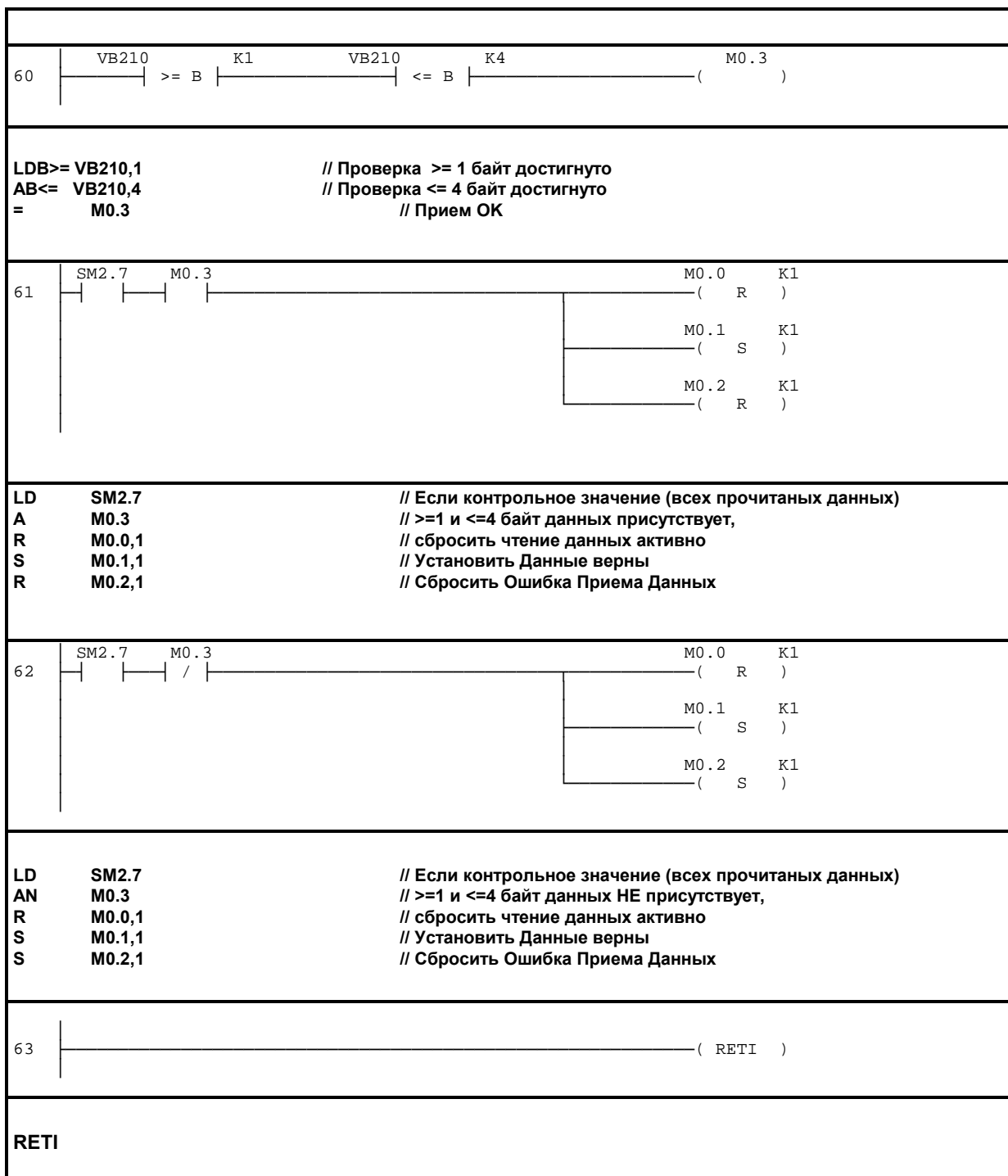
```
LDN  SM2.7           // СЛОВО ДАННЫХ датчика
AB<= VB210,3        // Счетчик Знаков <= 3
MOVD &VB211,VD200   // Загрузить адрес 1го знака буфера
MOVB VB210,AC0      // Счетчик Знаков в ACCU
+D   AC0,VD200      // Подсчитать указатель
```



```
LDN  SM2.7           // СЛОВО ДАННЫХ от Датчика
AB<= VB210,4        // Счетчик Знаков <= 4 ?
MOVB VB210,AC0      // Увеличить Счетчик Знаков
INCW AC0
MOVB AC0,VB210
```



```
LDB<= VB210,4       // Больше, чем 4 знака
AN   SM2.7          // Слово Данных от датчика
MOVB AC0,VB210     // Загрузить символ в буфер приема
```



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. Каждый 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.
- Комментарии строк, обозначенные "//" не поддерживаются в S7-Micro/DOS, но разрешены комментарии 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 соблюдения соответствующих технически безопасных предписаний при применении необходимо предпринять дополнительные меры.

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

Все права защищены. Любая форма размножение и дальнейшего распространения, в том числе и частично, допустимо только с письменного разрешения SIEMENS AG.