

SIMATIC

S7-200 Примеры

Группа	Пример к теме
2	Управление Быстрым счетчиком

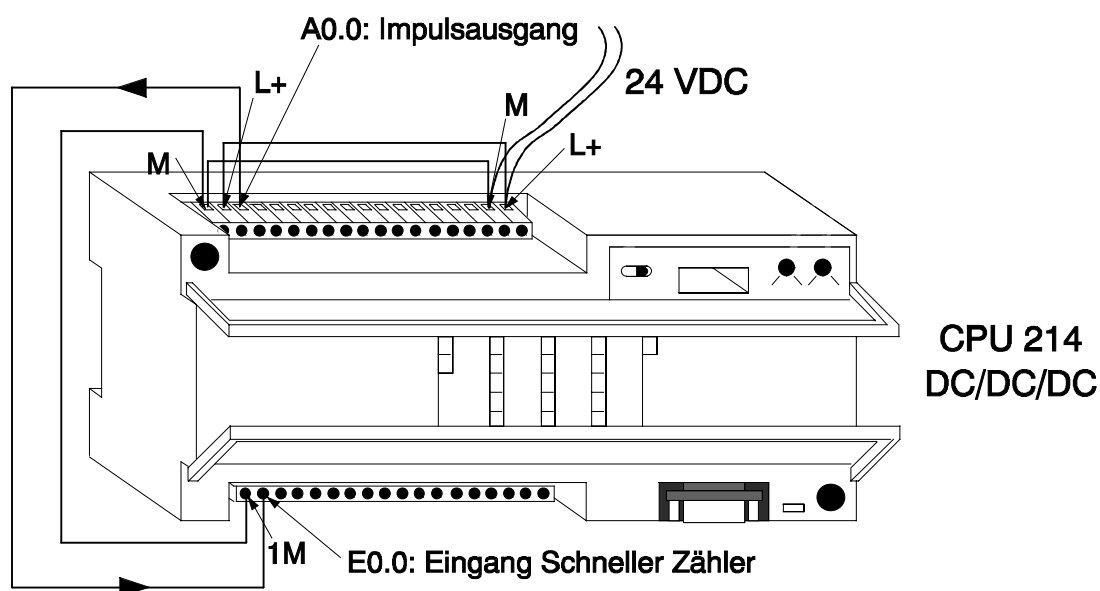
Краткое описание

В данном примере применения поясняется функция быстрого счетчика в SIMATIC S7-200. Быстрый счетчик может использоваться в различных конфигурациях, для того чтобы обрабатывать входные сигналы от датчиков, как напр. датчик перемещения.

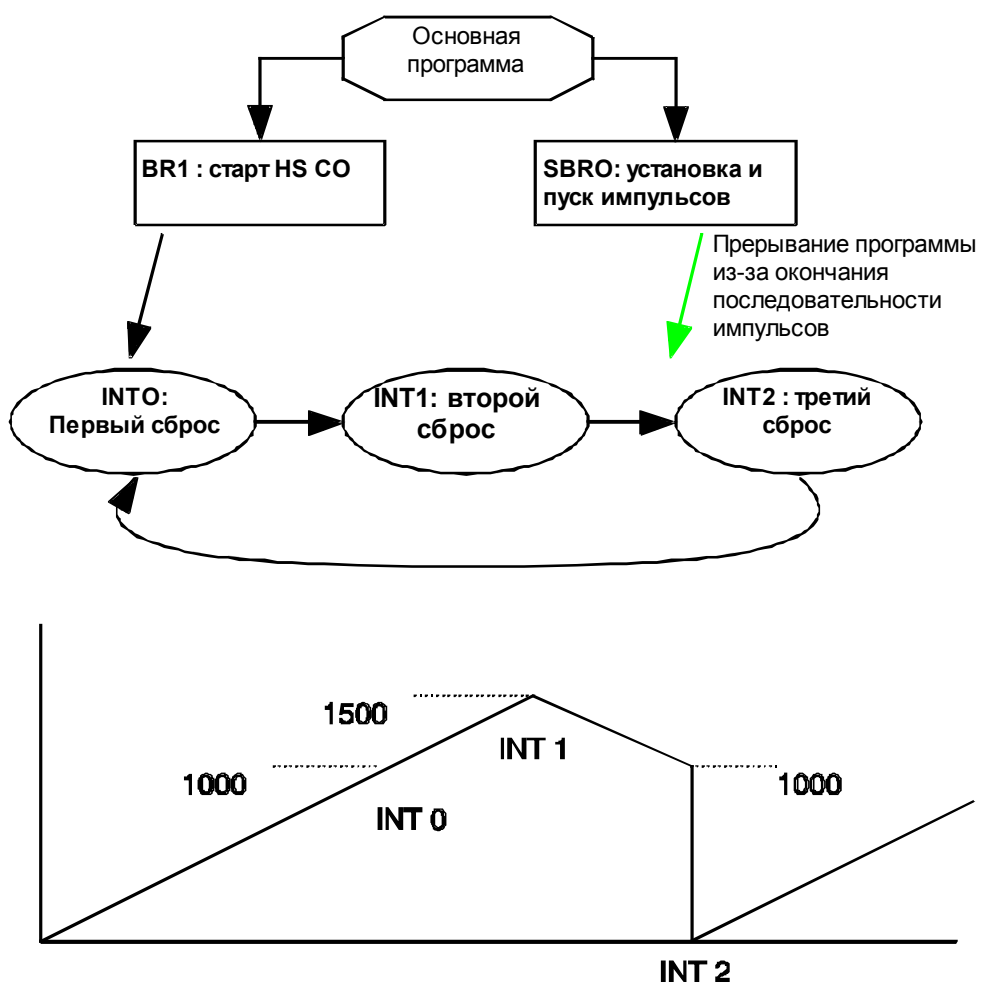
Импульсные выходы будут использоваться в данном, для того чтобы создать быстрые сигналы. При этом появляется возможность генерировать последовательность импульсов (ПТО), а также модулировать ширину импульсов (ШИМ), для того чтобы напр., управлять серводвигателем. Так как в данном применении используется последовательность импульсов, то можно использовать только **CPU 214 DC/DC/DC**.

Данное применение призвано показать, как должна быть структурирована программа, для того чтобы использовать быстрый счетчик и функцию последовательности импульсов в простейших конфигурациях.

Схема включения



Структура программы



Описание программы включая листинг

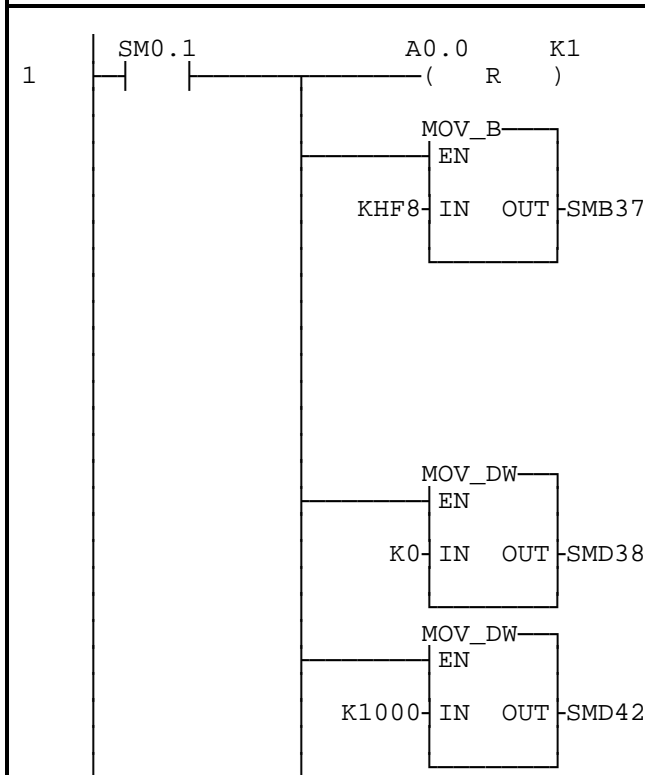
В этом примере описывается функция быстрого счетчика S7-200 DC/DC/DC. Он считает быстрее чем ПЛК может обработать цикл. Используемый счетчик это 2 килогерцовый программный счетчик, который встроен в S7-212. S7-214 имеет дополнительно два аппаратных счетчика по 7 КГц. Все счетчики вместе требуют 10 байт в памяти данных, для управления, хранения значений, счета и чтения текущего состояния.

Размер программы составляет 96 слов.

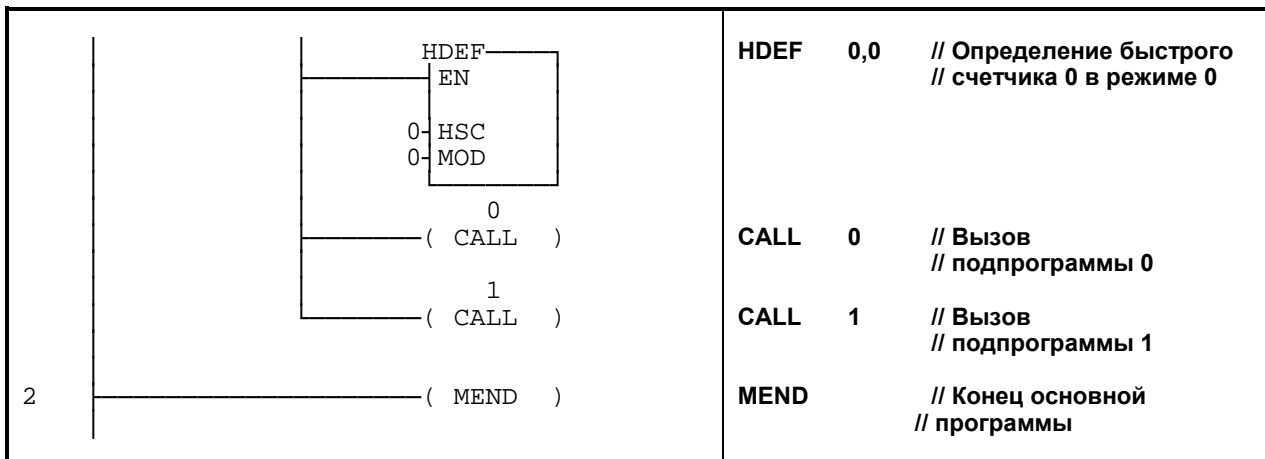
Более подробную информацию о быстром счетчике Вы найдете в Главе 6.1 Руководства по программированию SIMATIC S7-200, дальнейшую информацию о последовательности импульсов Вы найдете в Главе 6.3.

KOP (S7-MicroDOS)**AWL (TOOLITE2)****Основная программа**

// Основная программа сбрасывает выход A0.0, так как он необходим для
// функции импульсов. Кроме того инициализируется быстрый счетчик HSC0
// и вызываются подпрограммы 0 и 1. При этом HSC0 запускается
// со следующими свойствами: возможна актуализация значения, прямой
// счет. Программа завершается, если выполняется число циклов указанное
// в SBR 0 (SMD72).

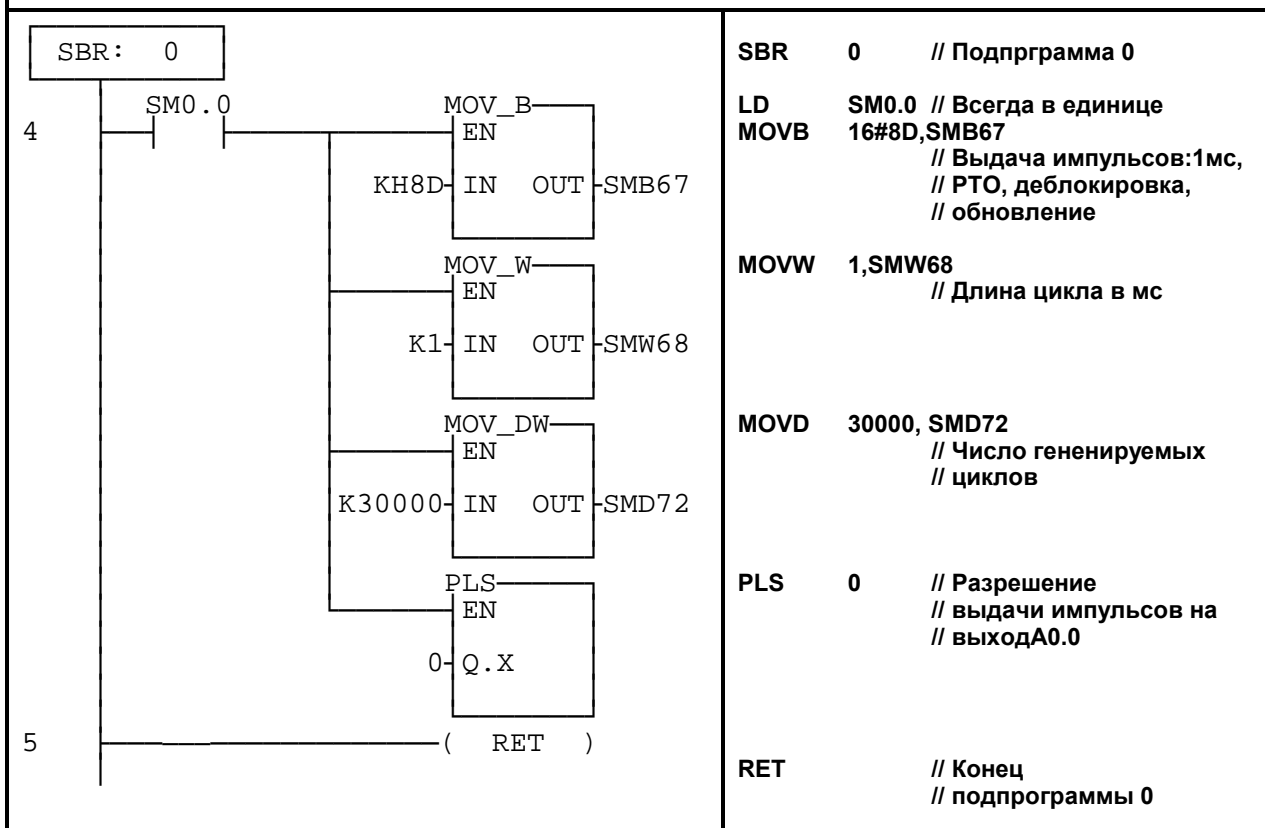


```
LD      SM0.1 // Первый цикл
R       A0.0,1 // Сброс выхода
           // A0.0 для импульсов
MOV_B   16#F8,SMB37
           // Загрузка контрольных
           // битов для быстрого
           // счетчика HSC0
           // (деблокировка HSC0,
           // обновление текущих
           // значений, обновление
           // сбросов, обновление
           // направления счета и
           // числа циклов. Команда
           // HSC использует эти
           // контрольные биты,
           // для конфигурирования
           // быстрого счетчика.
MOV_D   0,SMD38
           // Начальное значение
           // быстрого счетчика
           // HSC0
MOV_D   1000,SMD42
           // Первое значение
           // останова от HSC0
```

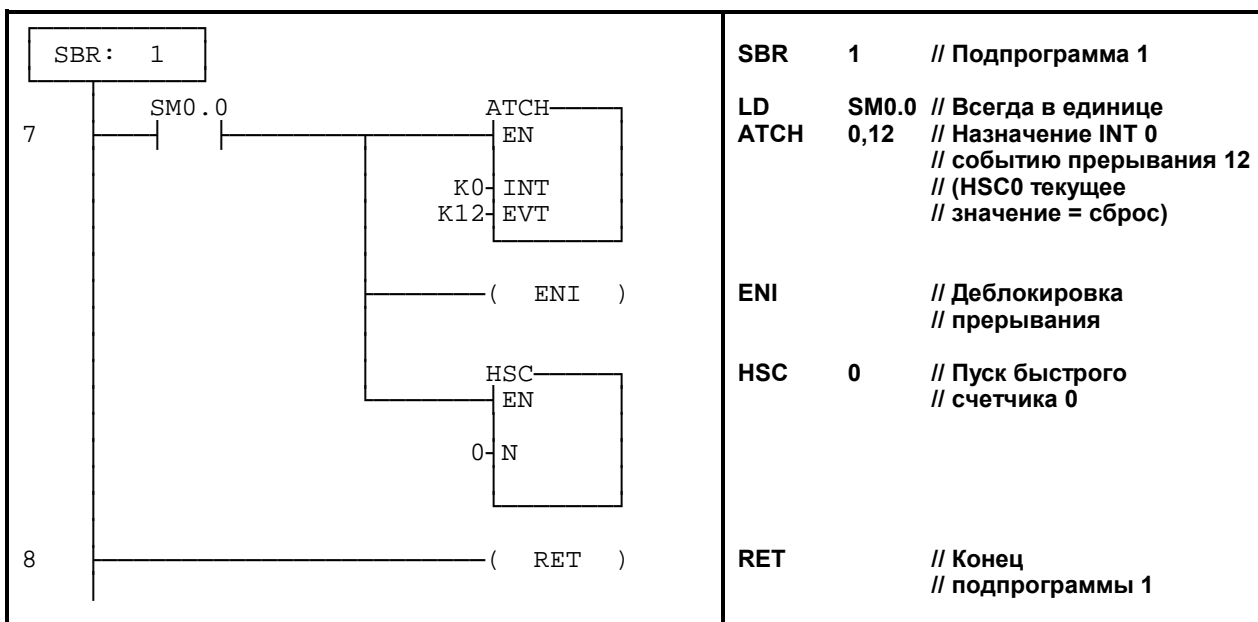


Подпрограммы

// Подпрограмма 0 служит для инициализации и деблокировки выдачи импульсов.
 // В байте специальных меркеров SMB67 определяется вывод: последовательность импульсов,
 // временой базис, обновление значений и деблокировка. Слово SMW68 содержит
 // время цикла как мультипликатор временного базиса. В двойном слове специальных меркеров
 // задается число генерируемых циклов.

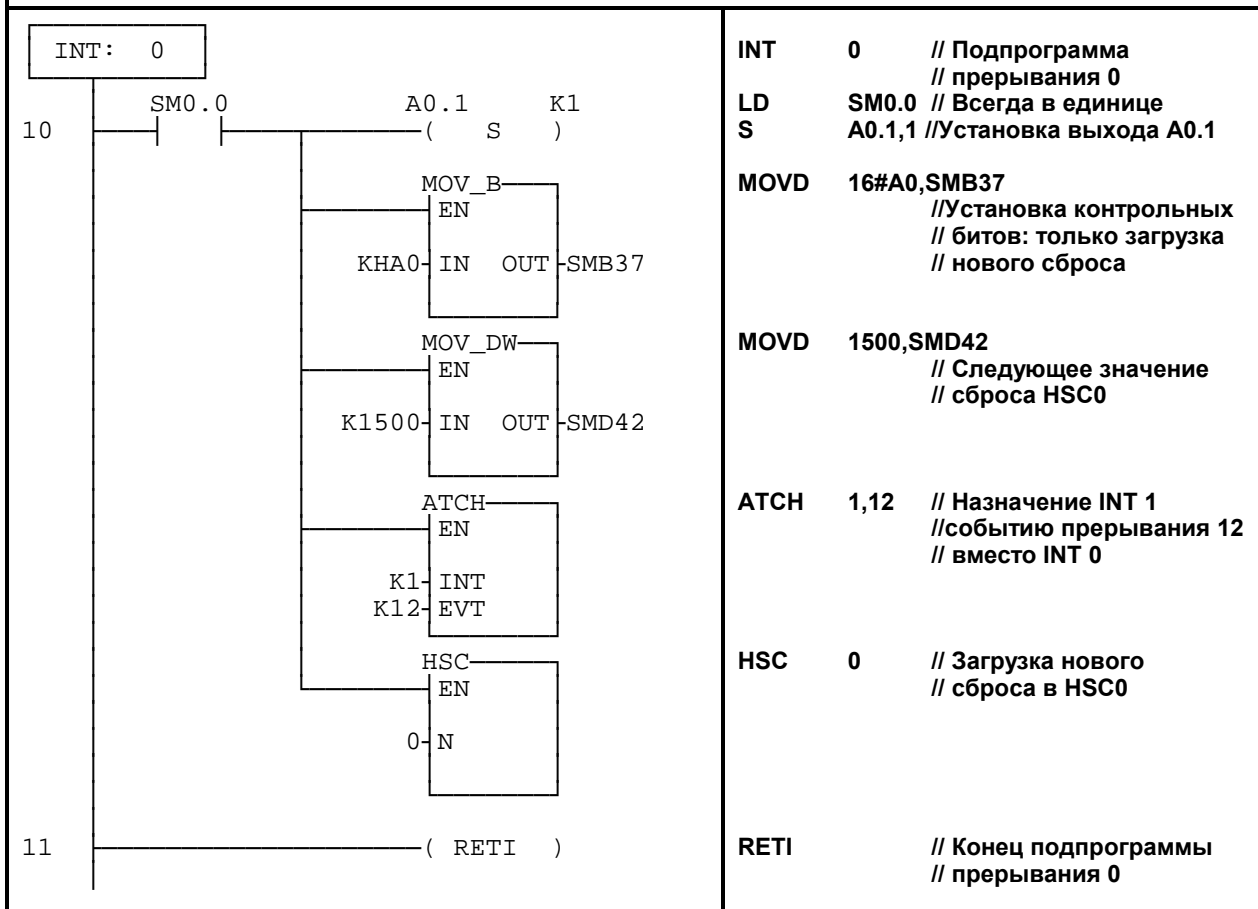


// Подпрограмма 1 запускает быстрый счетчик HSC0 и назначает
 // подпрограмму 0 событию прерывания 12 (HSC0 текущее значение = сброс).
 // Это событие возникает, если число сосчитанных импульсов (текущее значение)
 // достигает текущего значения останова (сброс).
 // Происходит деблокировка прерывания.

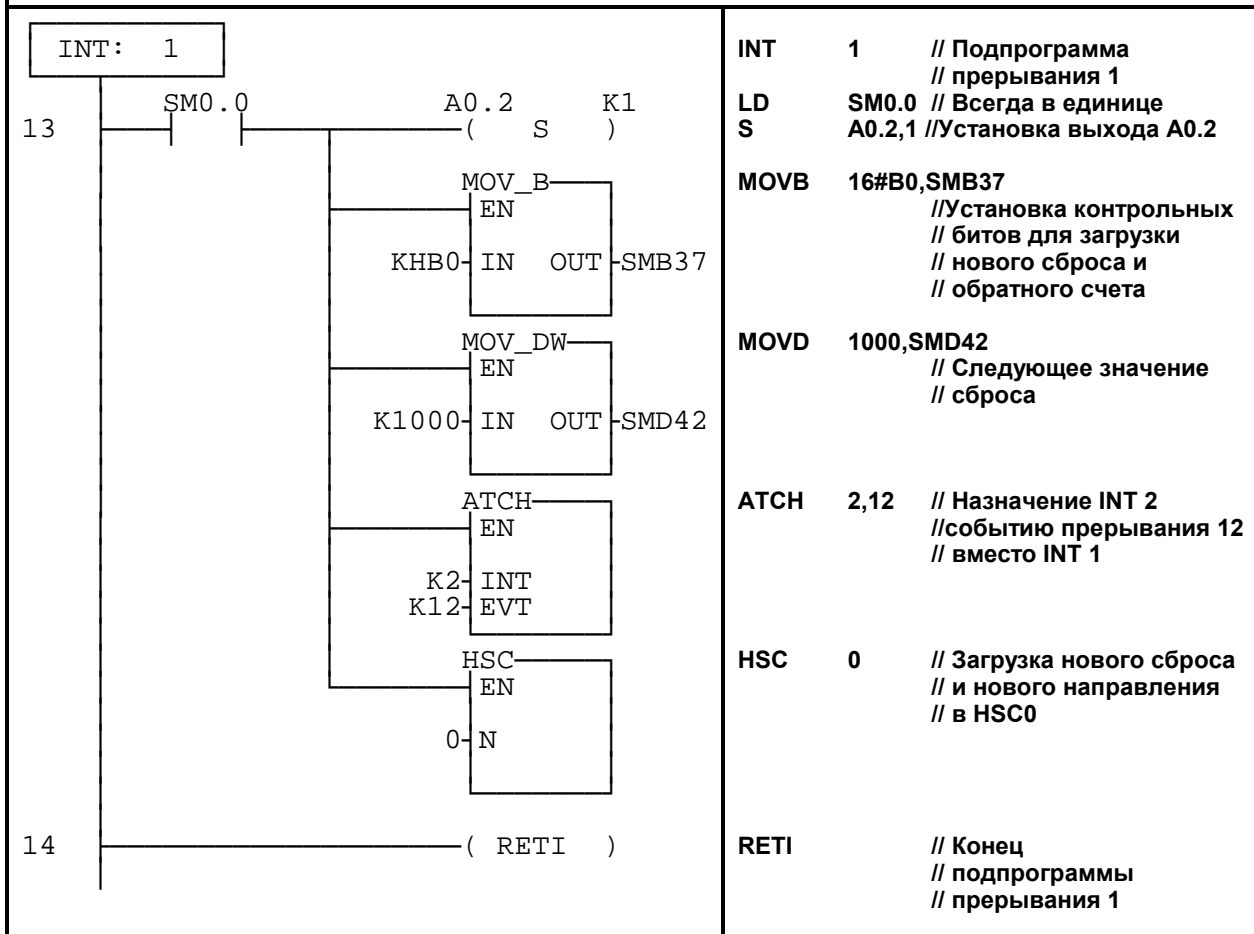


Подпрограммы прерываний

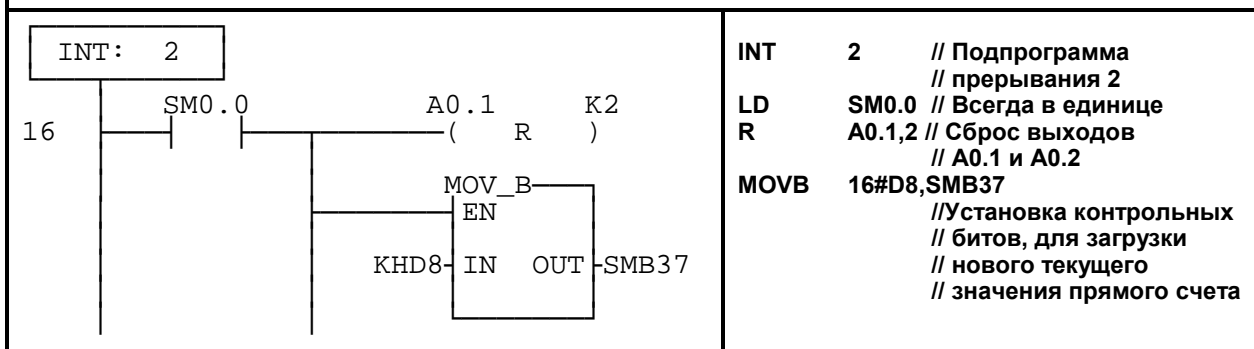
//Подпрограмма прерывания 0 будет вызвана, если достигнуто первое значение сброса (1000)
// быстрого счетчика 0. Выход A0.1 устанавливается и задается новое
// значение сброса (1500) для быстрого счетчика.
// Подпрограмма прерывания 1 назначается событию прерывания 12
// -вместо подпрограммы 0.

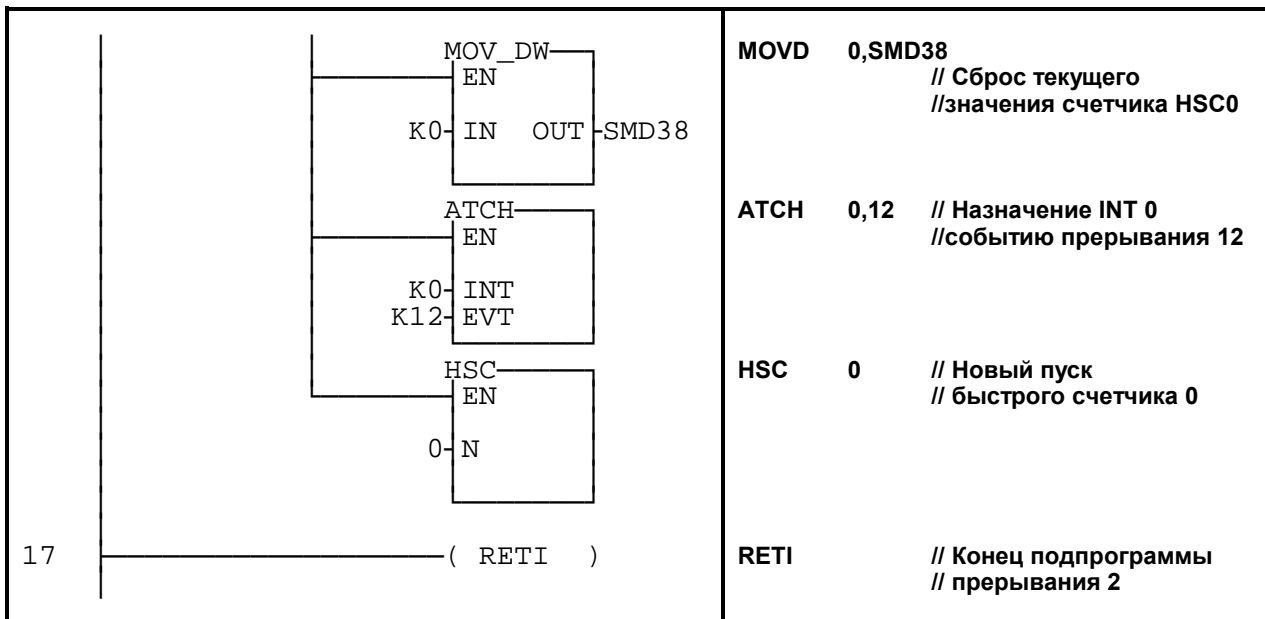


// Подпрограмма прерывания 1 вызывается, если достигнуто второе значение сброса
 // (1500) быстрого счетчика 0. Выход A0.2 устанавливается, направление счета
 // изменяется на обратное и определяется новое значение сброса (1000).
 // Подпрограмма прерывания 2 назначается событию прерывания 12



// Подпрограмма прерывания 2 вызывается, если достигнуто третье значение сброса (1000)
 // быстрого счетчика 0. Выходы A0.1 и A0.2 сбрасываются, направление счета
 // изменяется на прямое и текущее значение счетчика сбрасывается в ноль.
 // Значение сброса остается неизменным. При этом подпрограмма прерывания 0
 // назначается событию прерывания 12, выполнение программы начинается заново.
 // Программа завершается, если заданное в SBR 0 (SMD72) число циклов отработано.





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#' → '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.
- Комментарии к строкам начинающиеся с "/" в S7-Micro/DOS не возможны, зато возможны комментарии к сегментам.

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.