

SIMATIC

S7-200 Примеры

Группа

2

Пример к теме

Управление широтно-импульсной модуляцией
--

Краткое описание:

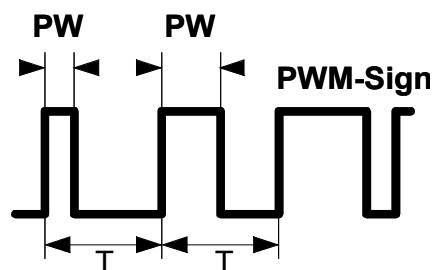
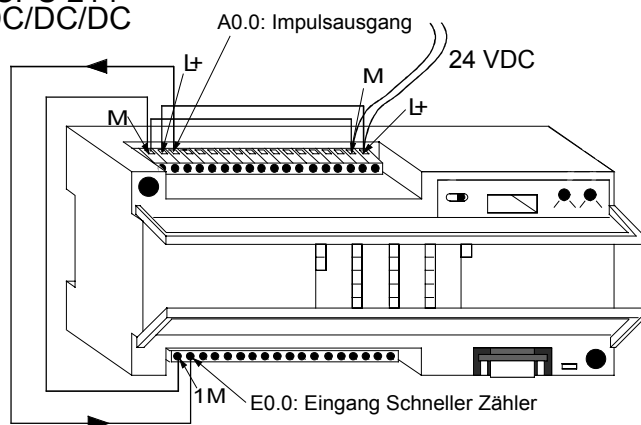
CPU-214 S7-200 содержит функцию выдачи прямоугольных сигналов на выходы A0.0 и A0.1. При этом длительность периода и ширина импульса устанавливаются независимо друг от друга. Ширина импульса соответствует времени, в течение которого выходной сигнал за период имеет состояние „1“.

Данный пример описывает широтно-импульсную модуляцию (ШИМ), где на выход A0.0 выдается сигнал, ширина импульса которого увеличивается от периода к периоду на 0,5 с. Длительность периода составляет 5 с, а стартовая ширина импульса 0,5 с. Если будет достигнуто максимальное значение ширины импульса 4,5 с, то ширина импульса снова начинает уменьшаться на 0,5 с, пока значение не станет равно нулю. Этот процесс повторяется периодически.

Для проведения такой широтно-импульсной модуляции необходимо связать выход A0.0 со входом E0.0, также как массу напряжения питания с массой входов, для того чтобы имелась возможность управлять ШИМ из программы.

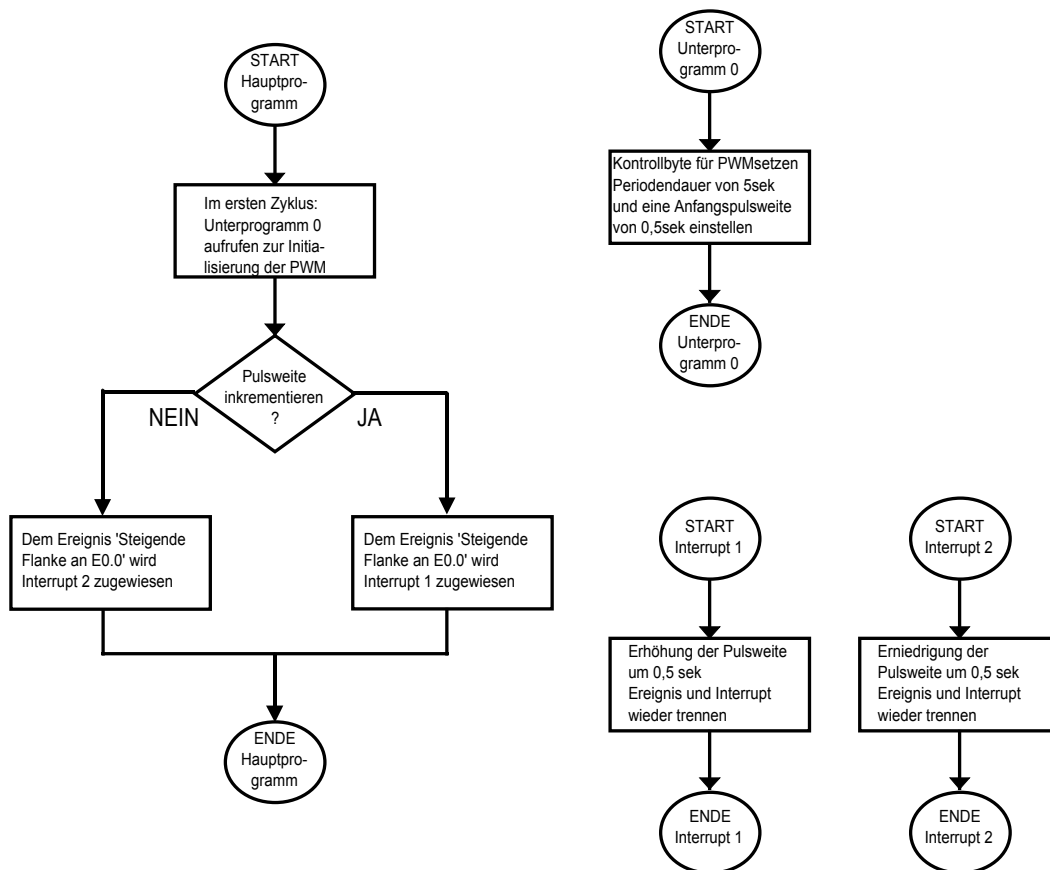
Схема включения:

CPU 214
DC/DC/DC



T = Periodendauer
PW = Pulsweite

Структура программы:



Описание программы вкл. листинг:

Байт специальных меркеров SMB67 служит для инициализации широтно-импульсной модуляции на выходе A0.0. Это инициализация содержит деблокировку ШИМ и установки, позволяющие изменять длительность периода и ширину импульса, а также выбрать временной базис (здесь в мс). В подпрограмме 0 будут установлены соответствующие контрольные байты. Команда ENI производит глобальную деблокировку всех прерываний. По команде PLS 0 операционная система запоминает установленные значения и инициализирует так называемый "РТО/ШИМ-генератор", который будет производить ШИМ на выходе A0.0.

Длительность периода 5 с задается передачей значения 5000 в слово специальных меркеров SMW68. Начальная ширина импульса устанавливается на 0,5 с, путем записи значения 500 в слово специальных меркеров SMW70.

Данная инициализация производится в первом цикле, путем логического связывания вызова подпрограммы с битом специальных меркеров SM0.1 (First Scan Flag). Инициализация и установка ШИМ будет производится повторно, после завершения процесса ШИМ, т.е. когда текущая ширина импульса станет равной 0.

Вспомогательный меркер M0.0 служит для установки, будет ли ширина импульса увеличиваться или уменьшаться. Во время инициализации он устанавливается, для того чтобы могло производиться инкрементирование. Выход A0.0 связан со входом E0.0, так что выходной сигнал появляется и на входе E0.0. После того как будет выдан первый импульс, событие '0' (положительный фронт на E0.0) будет назначено подпрограмме прерывания 1 (INT1) с помощью команды 'ATCH'.

INT1 повышает текущее значение импульса на 0,5 с а затем команда 'DTCH' разрывает связь этого прерывания с INT1, причем прерывание снова будет заблокировано. Если ширина импульса при следующем увеличении станет равна длительности периода, то вспомогательный меркер M0.0 будет сброшен. При этом событие '0' назначается подпрограмме прерывания 2, которая уменьшат ширину импульса после каждого импульса на 0,5 с. Когда ширина импульса станет равна 0, снова будет вызвана подпрограмма инициализации (подпрограмма SBR 0).

Размер программы составляет 66 слов.

Более подробную информацию о широтно-импульсной модуляции Вы найдете в Главе 6.3 "Быстрые операции на выходах" Руководства по программированию SIMATIC S7-200.

KOP (S7-MicroDOS)	AWL (TOOLITE2)
--------------------------	-----------------------

Основная программа

// TITEL = УПРАВЛЕНИЕ ШИРОТНО-ИМПУЛЬСНОЙ МОДУЛЯЦИЕЙ

// ***** ОСНОВНАЯ ПРОГРАММА *****

<p>1</p>	<p>LD SM0.1 // В первом цикле: CALL 0 // вызов // подпрограммы 0 для // пуска ШИМ</p>
<p>2</p>	<p>LDW>= SMW70, VW0 // Если ширина импульса // >= (длина периода - // ширина импульса), R M0.0, 1 // вспомогат. меркер // сбрасывается</p>
<p>3</p>	<p>LDW= SMW70, 0 // Если ширина импульса // равна нулю, CALL 0 // Вызов // подпрограммы 0 для // нового пуска ШИМ</p>
<p>4</p>	<p>LD E0.0 // Вход E0.0 установлен U M0.0 // и вспомогательный // меркер для увеличения // установлен, ATCH 1, 0 // то INT 1 назначена // событию 0 // (положительный фронт // на E0.0)</p>

	<pre> LD E0.0 // Вход E0.0 установлен, UN M0.0 // а вспомогательный // меркер для увеличения // не установлен, // то INT 2 назначена // событию 0 (положи- // тельный фронт на E0.0) ATCH 2, 0 // Конец основной // программы MEND </pre>
--	--

Подпрограммы

```

//***** ПОДПРОГРАММА 0 *****
// SM67.0 := 1    => сохранение новой длительности периода (деблокировка)
// SM67.1 := 1    => сохранение новой ширины импульса (деблокировка)
// SM67.3 := 1    => временной базис 1мс (если 0 => временной базис 1мкс)
// SM67.6 := 1    => выбор режима ШИМ (если 0 => режим PTO)
// SM67.7 := 1    =>общая деблокировка High-Speed-Output-функций
    
```

	<pre> SBR 0 // Инициализация // ШИМ LD SM0.0 // Всегда в единице S M0.0, 1 // Установка вспом. // меркера "увеличение // ширины импульса" MOV_B 16#CB, SMB67 //Установка контрольных //байтов PTO/ШИМ- // для выхода A0.0 MOV_W 500, SMW70 // Задание начальной // ширина импульса // (здесь 500 мс) MOV_W 5000, SMW68 // длина периода // (здесь 5 с) ENI // Деблокировка всех // прерываний PLS 0 // Команда на // программирование //PTO/ШИМ-генератора MOV_W SMW68, VW0 // Копировать длину // периода в слово // данных VW0 -I 500, VW0 // Сохранить значение // 'длина периода- // ширина импульса' // в слове данных VW0 // Конец подпрограммы 0 // и возврат в // основную программу RET </pre>
--	--

Подпрограммы прерываний	
// ***** INTERRUPT SERVICE ROUTINE 1 *****	
	<p>INT 1 // Увеличение ширины // импульса</p> <p>LD SM0.0 // Всегда в единице</p> <p>+I 500, SMW70 // Увеличение ширины // импульса на 500мс</p> <p>PLS 0 // Команда на // программирование // ПТО/ШИМ-генератора</p> <p>DTCH 0 // Разделить прерывание // и назначенное ему // событие 0</p> <p>RETI // Конец ISR 1 и // возврат в // основную программу</p>
// ***** INTERRUPT SERVICE ROUTINE 2 *****	
	<p>INT 2 // Уменьшение ширины // импульса</p> <p>LD SM0.0 // Всегда в единице</p> <p>-I 500, SMW70 // Уменьшение ширины // импульса на 500мс</p> <p>PLS 0 // Команда на // программирование // ПТО/ШИМ-генератора</p> <p>DTCH 0 // Разделить прерывание // и назначенное ему // событие 0</p> <p>RETI // Конец ISR 2 и // возврат в // основную программу</p>

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.
- 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.