

20-канальный ШИМ контроллер для управления сервомашинками

Автор: Илья Мамонтов
Сопровождающий проект: Есть
Семейство микросхем: Все
Среда разработки: PSoC Designer 4.3
Ссылки на дополнительные документы: Нет

Резюме

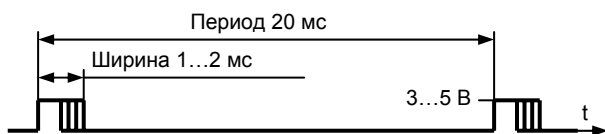
Эта статья описывает реализацию 20-канальной системы управления сервоприводами (сервомашинками), где для генерации широтно-модулированных импульсов используются только два Модуля Пользователя PWM8. Поддерживаются стандартные сервомашинки с длительностью управляющих импульсов от 1 до 2 мс и периодом повторения 20 мс. Используя описанный метод, вы можете строить свои системы, с числом каналов до 64-х (применяя 100-выводные модификации микросхем семейства PSoC).

Введение

Такие области техники, как радиоуправляемые системы и в особенности – робототехника, требуют применения большого количества сервомашинки одновременно.

Сервомашинка – это устройство, создающее вращательное или возвратно-поступательное движение в ответ на входной управляющий сигнал. Типичная машинка содержит электромотор, редуктор, датчик положения и схему управления. Для управления обычно используется последовательность импульсов с изменяемой шириной (широтно-импульсная модуляция или ШИМ, см. Рисунок 1). Импульсу длительностью 1,5 мс соответствует среднее положение сервомашинки, импульсам длительностью 1 и 2 мс – крайние положения.

Рисунок 1. Стандартный управляющий сигнал



Основная проблема, которая обычно возникает – как управлять двумя или более машинками одновременно. Одно из решений – использовать коммерческие продукты (обычно с числом каналов до 8) и соединять

их в цепочку для получения требуемого количества каналов.

Другое эффективное решение заключается в использовании микроконтроллеров (МК) – они могут не только сгенерировать требуемые сигналы, но и одновременно быть задействованы для других задач, например – для высокоуровневого управления.

Устаревший способ с использованием МК – это генерировать ШИМ-импульсы непосредственно микропрограммой, с помощью установки-сброса выходных бит и организацией требуемых задержек (программными циклами, прерываниями от таймера и т.д.). Основным недостатком подобных алгоритмов заключается в том, что они требуют согласованной работы во времени всех частей программы и использования различных уловок для того, чтобы устранить (или сократить) возможные флуктуации выходного сигнала, так называемый «джиттер» (дрожание фронтов импульсов). Кроме того подобные алгоритмы очень сильно нагружают центральный процессор. Основной способ избежать этого – использовать аппаратные генераторы ШИМ, которые присутствуют во всех современных МК.

К сожалению, количество ШИМ-каналов большинства МК обычно ограничено четырьмя, и к тому же 8-разрядные ШИМ-генераторы не позволяют получить число градаций длительности импульса более чем 12 (результат того, что сервомашинки требуют изменения ширины импульсов от 1 до 2 мс при периоде их повторения 20 мс, т.е. эффективный диапазон будет составлять $2^8 \times (2-1)/20 = 12$ значений).

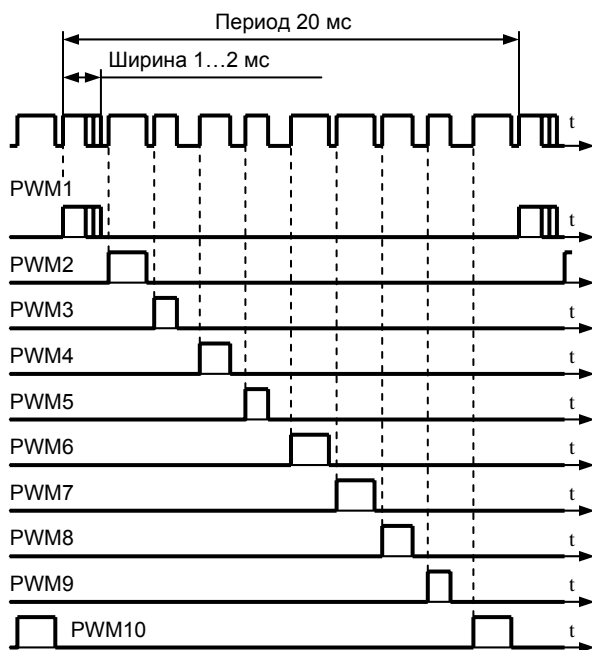
Использование микроконтроллеров PSoC помогло бы частично преодолеть эти ограничения. К примеру, кристаллы семейства CY8C29XXX имеют на борту 16 цифровых блоков, что позволяет сконфигурировать до 8 двухблочных (т.е. 16-разрядных) ШИМ-генераторов. Но главная особенность всех микроконтроллеров PSoC – это возможность переконфигурирования внутренней структуры во время работы и... было бы глупо не воспользоваться этим!

Чтобы получить многоканальную широтно-импульсную модуляцию, я скомбинировал два метода – аппаратную генерацию импульсов и программную манипуляцию над «внутренностями» PSoC.

Как это работает внутри PSoC

Главная идея получения многоканальной ШИМ заключается в генерации цепочки поджатых друг к другу импульсов и перенаправлении их в разные каналы в соответствующие моменты времени. Период стандартного сигнала управления легко делится на 10 интервалов, как это показано на Рисунке 2. В каждый обозначенный интервал выход модуля, генерирующего импульсы, соединяется только с одним из нескольких выходов микросхемы.

Рисунок 2. Генерирование многоканальной ШИМ



Для генерации я использую 8-битный Модуль Пользователя PWM8, который позволяет получить эффективное разрешение длительности импульса 7 бит (из-за того, что минимальная ширина импульса составляет 50% от максимальной). Такое разрешение достигается при значениях от 0 до 127, загружаемых в Регистр Сравнения (Compare Register). В этом случае уровень на активной стадии выходного импульса будет

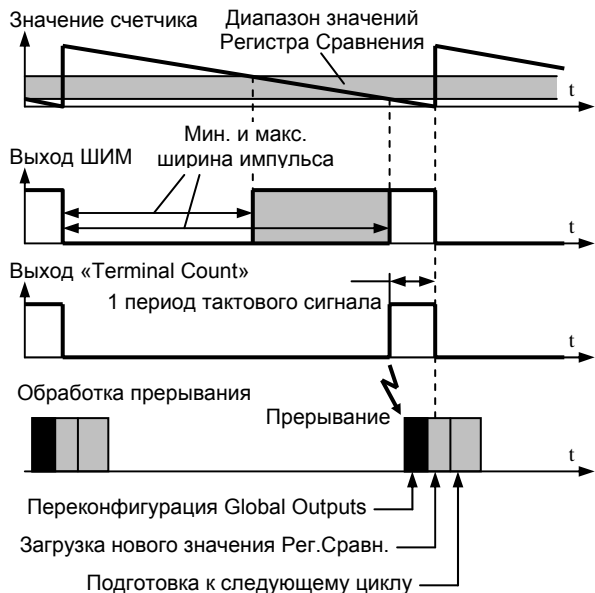
низким, так что потребуется инверсия сигнала перед тем, как он будет подан на выход. Эта операция может быть реализована посредством функции «НЕ» элемента LUT – специального блока PSoC, управляющего шиной Global Output Bus.

Чтобы последовательно перенаправлять импульсы на выходные ножки, используются некоторые уловки.

1. Для перенаправления используются специальные регистры назначения Port Global Select Registers (PRTxGS). Они задают, какой источник – биты регистра данных Data Register (PRTxDR) или линии шины Global Output Bus – будут управлять ножками микросхемы. На каждом промежутке времени только одна ножка из нескольких должна принимать импульс с модуля PWM8, логические уровни на остальных ножках должны быть низкими. В связи с этим регистр данных Data Register должен быть предварительно установлен в состояние «все нули».
2. PSoC-микросхемы имеют особую внутреннюю архитектуру – выходная ножка может соединяться только с одной определенной линией шины Global Output Bus. Поэтому, чтобы управлять всеми восемью ножками порта, вы должны управлять всеми восемью соответствующими линиями шины. К счастью, к одной и той же линии могут подключаться до нескольких ножек от разных портов (это зависит от количества выводов PSoC).
3. (в дополнение к п.2) Только четыре линии шины Global Output Bus могут управляться от одного источника (блока LUT). Чтобы управлять всеми 8-ю линиями шины, вы должны задействовать два блока LUT. Но из-за ограниченных возможностей в части межсоединений, эти два блока должны быть соседними по ряду (как показано на Рисунке 5). В результате этого мы имеем специфическое распределение каналов по ножкам микросхемы.
4. Самое подходящее время для перекоммутации выходов – это момент перед началом следующего импульса (на Рисунке 3 активная часть импульса обозначена как «Мин. и макс. ширина импульса»). Для «поимки» этого момента используются прерывания по окончании счета от модуля PWM8 (Interrupts on Terminal Count, или TC).

Как видно из Рисунка 2, максимальное количество каналов, которое можно получить от одной последовательности импульсов – 10. Чтобы увеличить это число, вы должны сгенерировать две или более ШИМ-последовательности, т.е. использовать дополнительные модули PWM8. Это не является существенным недостатком – вы теряете в ресурсах PSoC, но выигрываете в задержках управления, ведь новые каналы будут работать одновременно с имеющимися (обратите внимание на диаграммы Рисунка 4, где импульсы четных и нечетных каналов попарно синхронны).

Рисунок 3. Подробности формирования ШИМ-импульса



Прерывания по окончании счета (Terminal Count) используются также для перезагрузки регистра сравнения (Compare Register) новым значением – чтобы

получить требуемую ширину следующего импульса. Подробности показаны на Рисунке 3 (здесь ширина импульса «Terminal Count» сильно преувеличена). Важно отметить, что тот промежуток времени, где производится переконфигурация выходов, должен находиться внутри паузы между соседними импульсами, т.е. внутри импульса Terminal Count. Это обеспечивается:

- установкой надлежащих параметров модуля PWM (значения регистра Compare Register находятся в диапазоне 0...127, условие сравнения (Compare Type) выбрано «Less Than or Equal To», т.е. «меньше или равно»),
- оптимизацией кода обработки прерывания (та часть кода, которая осуществляет переконфигурирование, располагается в начале обработчика прерывания, а количество инструкций минимизировано).

Реализация контроллера

Схема 20-канального ШИМ контроллера показана на Рисунке 4, внутренняя конфигурация PSoC – на Рисунке 5.

Рисунок 4. Схема контроллера и диаграммы выходных сигналов.

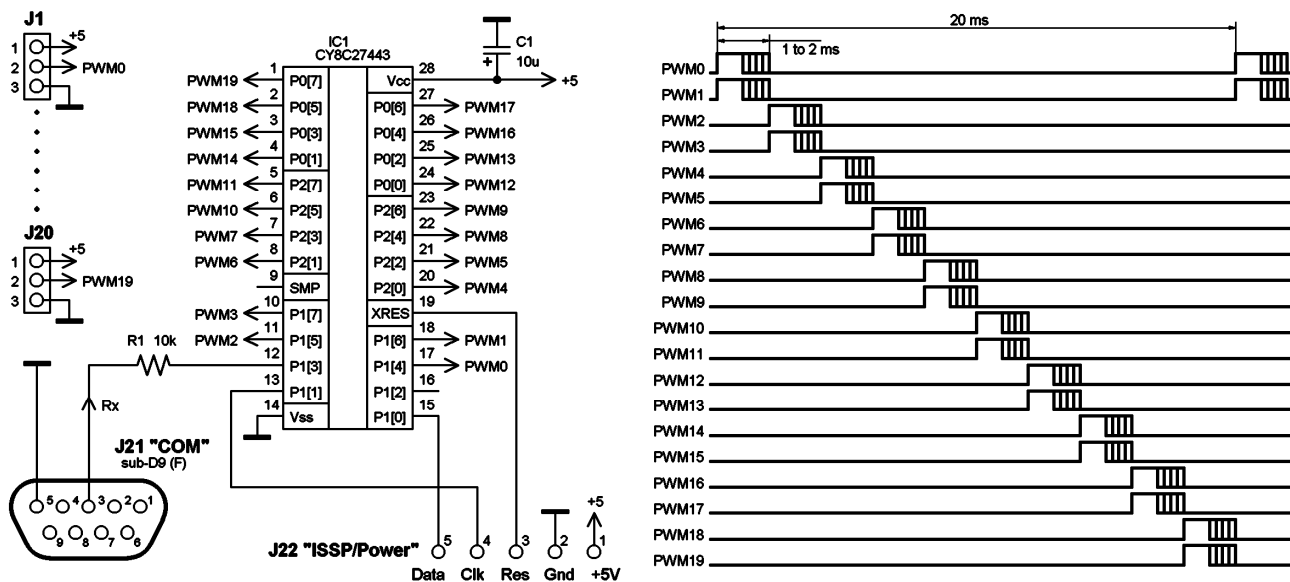
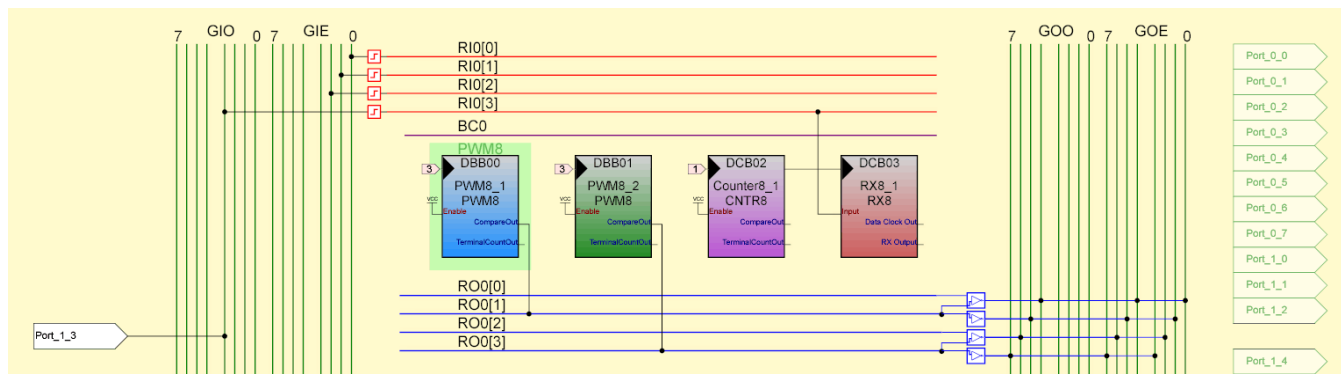


Рисунок 5. Внутренняя конфигурация PSoC (VC1=24МГц/6=4 МГц, VC3=VC1/188)



Протокол Mini SSC II фирмы Scott Edwards Electronics стал фактическим стандартом для многоканальных устройств управления сервоприводами. Этот последовательный протокол предусматривает 3-байтовый формат команды:

- байт 1 – синхронизирующий, всегда равен 255,
- байт 2 – номер сервомашинки (от 0 до 254),
- байт 3 – положение ее привода (от 0 до 254).

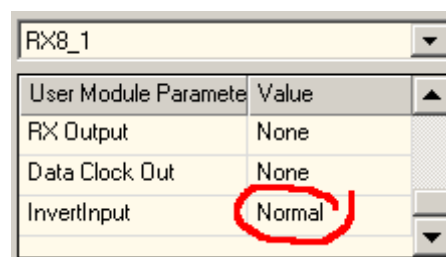
В моем проекте он поддерживан со следующими ограничениями:

- номер сервомашинки (или номер ШИМ-канала) – от 0 до 19,
- диапазон положений привода сервомашинки 0...127 (т.е. используется 7-битное разрешение, значению 64 соответствует среднее положение привода).

Протокол реализован в виде так называемого «автомата состояний» в файле *main.asm* Сопровождающего Проекта. Два модуля – Counter8_1 и RX8_1 – аппаратно поддерживают канал связи. Первый модуль является делителем частоты для получения требуемой скорости обмена данными второго модуля, 8-битного приемника. Оба модуля сконфигурированы для поддержки режима «9600,8,N,1» (что означает скорость обмена 9600 бод, 8-битовая разрядность данных, отсутствие проверки на четность, 1 стоповый бит).

Замечательной особенностью модуля RX8 является возможность инверсии входа данных (режим «Invert Input mode»). Это означает, что вы можете напрямую подсоединять вход этого модуля к COM-порту компьютера (через токоограничительный резистор, как это показано на Рисунке 4). Если вы будете использовать микросхему преобразователя уровней (например, MAX232) или другие микроконтроллеры в качестве источников сигналов с «нормальной» логикой, в окне Редактора Устройств установите значение «Normal» у параметра «InvertInput», как это показано на Рисунке 6.

Рисунок 6. Параметры модуля RX8 при использовании преобразователя MAX232 – вы должны выбрать режим «Normal»



20-канальная широтно-импульсная модуляция реализована с использованием модулей PWM8_1, PWM8_2 и с помощью манипуляций на регистрах назначения PRT0GS...PRT2GS. Весь управляющий код размещен в процедуре обработки прерывания модуля PWM8_1 (файл *PWM8_1INT.asm*).

Вначале переназначаются выходы микросхемы, с помощью следующего фрагмента кода:

```

;reconnect global outputs
mov A,[mask+0]
mov reg[PRT1GS],A ;port_1
mov A,[mask+1]
mov reg[PRT2GS],A ;port_2
mov A,[mask+2]
mov reg[PRT0GS],A ;port_0
    
```

где два бита 3-байтовой переменной *mask* установлены в «1» – с целью соединения двух линий выходных портов с двумя источниками ШИМ последовательностей.

Затем в Регистры Сравнения (Compare Registers) модулей PWM8_1 и PWM8_2 заносятся новые значения ширины импульса – для генерации следующего импульса. Эти значения берутся из 20-байтового массива положений сервомашинки *pwdm* (этот массив обновляется с помощью принятых команд Mini SSC II).

В конце процедуры, согласно внутренней архитектуре межсоединений, вычисляются два новых бита переменной *mask*, а также увеличивается на 1 указатель на следующее значение массива *pwm*.

Быстрое тестирование

Я разработал простейшую программу связи для управления всеми двадцатью каналами, работающую в среде Windows (скриншот показан на Рисунке 6). Zip-архив этой программы вы можете скачать со страницы загрузки статьи.

«Движки» управляют положениями приводов сервомашинки, панель «Advanced» создает дополнительные возможности при реализации и отладке ваших собственных команд.

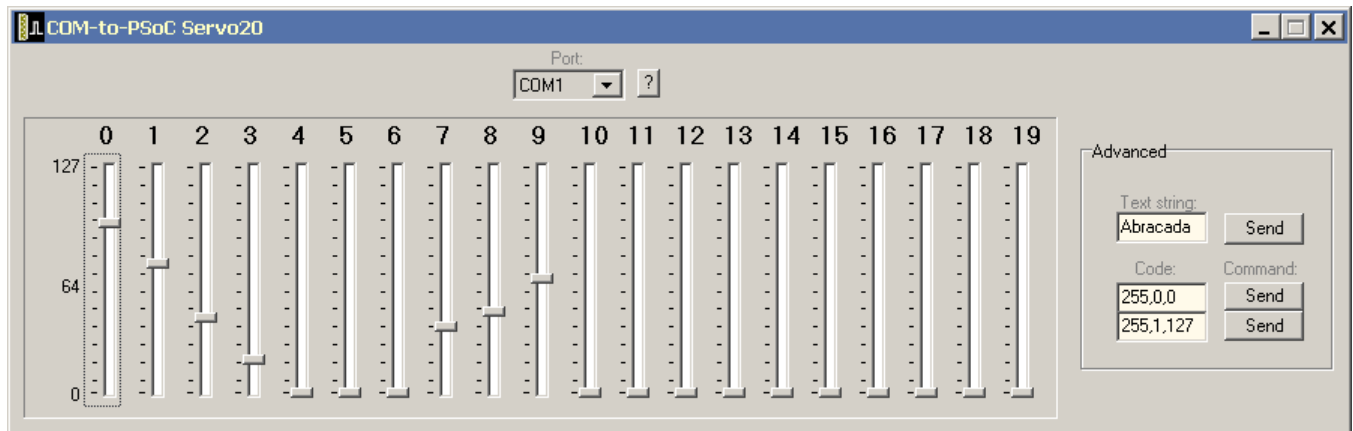
Заключение

Комбинируя аппаратную генерацию импульсов и программную манипуляцию над внутренней структурой PSoC, вы можете реализовать многоканальную систему управления сервомашинками.

Чтобы создать 20-канальный контроллер, достаточно использовать 28-выводную модификацию PSoC и задействовать только два модуля ШИМ (PWM8).

Если вы усвоили описанную методику, то сможете построить свои собственные системы, с числом каналов от 21 до 64 (используя модификации кристаллов PSoC с другим количеством выводов).

Рисунок 6. Программа связи



About the Author

Name: Ilya Mamontov

Title: Electronic Engineer

Background: Ilya graduated from Moscow Aviation Institute (Russian Federation) in 1989 and worked 6 years for spacecraft electronic engineering. He presently working for atomic industry, his region of interests is design, modernization, repair and technical service of measurement instruments.

Contact: ilka@elsite.ru subj: PSOC AN
or
illinoys@narod.ru subj: PSOC AN

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
<http://www.cypress.com>

© Cypress Semiconductor Corporation, 2006. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.