

Управление скоростью вращения двигателя постоянного тока с помощью широтно-импульсной модуляции

При регулировании скорости вращения двигателя с помощью широтно-импульсной модуляции (ШИМ, в англоязычных текстах - PWM, pulse-width modulation) на двигатель подается полное напряжение питания, но регулируется время, в течение которого оно подается.

Как правило, период подачи питания одинаков, а меняется лишь время, в течение которого оно подается. Условно говоря, мы нажимаем на кнопку питания двигателя каждую секунду, но если нам нужно только 25% мощности двигателя, то мы удерживаем кнопку нажатой всего четверть секунды, а если 50% мощности - то полсекунды. Когда же нам нужна от двигателя полная мощность, мы держим кнопку нажатой полную секунду (то есть фактически ее просто не отпускаем).

Конечно же, в большинстве реальных случаев питание подается на двигатель с частотой много выше, чем один раз в секунду, но принцип остается тем же.

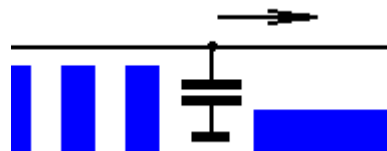


подробнее>>> shim То, что с помощью дискретного сигнала ("включено/выключено") можно управлять аналоговой величиной (частотой вращения двигателя), делает этот способ очень популярным при использовании цифровых схем управления и, в частности, микроконтроллеров.

О выборе частоты

Обычно частота ШИМ бывает довольно высока и составляет от нескольких десятков до нескольких сотен герц. В этом случае при емкостной нагрузке происходит сглаживание пульсаций питающего напряжения и фактически мы подаем на двигатель некоторое эффективное напряжение. Скажем, при напряжении питания 10В и длительности импульса 50% от периода, мы получим точно такой же результат, как если бы просто подали на двигатель напряжение 5В.

Часто именно это и требуется. Более того, это один из наиболее популярных способов регулирования аналогового напряжения в цифровых схемах.



Обратите внимание, что если емкость нагрузки мала, вы можете столкнуться с неожиданным поведением двигателя, что особенно заметно у дешевых микромоторов. Коллекторная система двигателя играет роль коммутатора, включая и выключая его обмотки; когда же на включение обмоток накладывается периодическое включение питания, результат может быть непредсказуемым, вплоть до того, что при определенном сочетании частоты ШИМ и частоты вращения двигателя, последний начнет вращаться в обратную сторону!

Тем не менее, возможны и некоторые особые случаи. Дело в том, что в обычном коллекторном двигателе постоянного тока напряжение питания не задает, строго говоря, частоту вращения. От напряжения питания зависит лишь ток в обмотках двигателя, сила тока будет определять величину крутящего момента на его валу, а уже от крутящего момента будет зависеть частота вращения.

Поэтому регулирование частоты вращения двигателя (или же скорости движения конструкции, если речь идет о ходовых двигателях) может быть связано с определенными сложностями, особенно если речь идет о малых, по отношению к номинальной, скоростях.

Предположим на вашем роботе установлены ходовые двигатели с номинальным напряжением 10В и при подаче номинального напряжения он имеет скорость 10 см/сек. Так вот, если вам нужно получить скорость в 1 см/сек, это совсем не означает, что вам надо подать на двигатель напряжение в 1В! Более того, даже если вам удастся подобрать напряжение, при котором модель будет двигаться со скоростью 10% от номинальной, эта скорость, скорее всего, значительно изменится даже при незначительном изменении условий движения.

Одним из возможных решений этой проблемы (помимо использования систем автоматического регулирования, конечно) является включение двигателя на полную мощность, но на короткое время. Модель начнет двигаться небольшими рывками, которые можно сделать практически незаметными, правильно подобрав частоту и длительность импульсов.

Этим способом можно заставить модель устойчиво двигаться с любой скоростью от 0 до максимальной, причем скорость будет оставаться постоянной в достаточно широком диапазоне нагрузок.

Владельцы роботов LEGO SpyBotisc или MindStorms могут найти описание и текст программы на NQC, реализующей данный алгоритм, в учебнике по NQC, входящем в состав пакета BrixCC, в главе VIII. More about motors.

Подобного эффекта можно добиться, если использовать ШИМ с очень низкой частотой - обычно от нескольких герц до долей герца; конкретные параметры ШИМ как правило определяются опытным путем в зависимости от конструкции и характеристик конкретной модели.

Программная реализация ШИМ

Большинство современных контроллеров имеют встроенные средства для организации ШИМ. К примеру, AT90S2313 имеет один таймер/счетчик, который может работать в режиме ШИМ, а ATmega16 - целых четыре канала ШИМ. Тем не менее, программно это тоже стоит уметь делать, хотя бы потому, что не всегда в вашем распоряжении столько линий с аппаратной ШИМ, сколько вам требуется (к примеру, для нашего [проекта](#) требуется 2 или 4 линии с ШИМ, а у используемого контроллера линия с аппаратной ШИМ только одна).

Алгоритм программной реализации ШИМ довольно прост: в начале каждого импульса устанавливаем на линии "1" и ждем, пока длительность импульса не достигнет заданного значения; когда это произойдет - сбрасываем линию в "0".

Естественно, что длительность импульса удобнее отслеживать не непрерывно, а с определенной периодичностью. Задавать длительность также удобнее ступенями (к примеру, десять ступеней регулирования будут соответствовать 10% от максимума, 20% от максимума и т.д.). Соответственно, логично для отсчета времени использовать прерывания по таймеру и в момент прерывания сравнивать текущее значение длительности импульса с заданным.

Итак, прежде всего, определяемся с частотой импульсов и числом ступеней регулирования. Произведение этих двух значений автоматически дает нам требуемую частоту прерываний таймера.

Иногда поступают и наоборот: выбирают "удобную" частоту таймера и/или число ступеней регулирования и как результат получают частоту импульсов.

К примеру, возьмем частоту импульсов около 200Гц, а скважность импульсов будем задавать с интервалом 10 процентов (т.е. 10 ступеней регулирования). В таком случае частота прерываний таймера должна быть около 2кГц. При тактовой частоте контроллера 4МГц требуемую частоту прерываний можно получить с

помощью 8-разрядного таймера, который обязательно есть в любом контроллере AVR, установив коэффициент деления частоты на его входе равным 8 (т. е. $400000/8/256 = 1,95\text{кГц}$).

Если обозначить линию как *OUT*, текущее значение времени как *pwm_t*, а заданное значение скважности как *pwm_ch*, то логика программы будет такой:

```
Подпрограмма обработки прерывания {  
    если (pwm_t = 10) pwm_t = 0;  
    если (pwm_t = 0) OUT = 1;  
    если (pwm_t >= pwm_ch) OUT = 0;  
    pwm_t = pwm_t + 1;  
}
```