

Universidade Federal do Rio Grande do Sul
Escola de Engenharia
Departamento de Sistemas Elétricos de Automação e Energia
ENG10032 Microcontroladores

Roteiro de Laboratório 6
Pulse-Width-Modulation – PWM

Prof. Walter Fetter Lages

24 de setembro de 2019

1 Objetivo

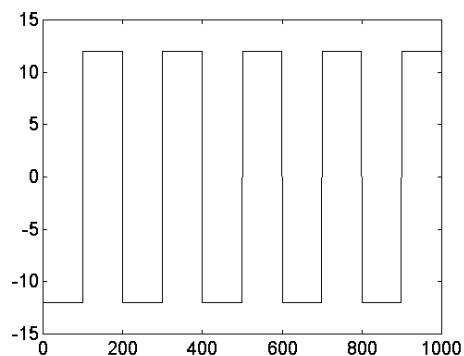
O objetivo deste laboratório é explorar as saídas PWM da Galileo Gen2. Em especial, tem-se como objetivos entender como utilizar um sinal PWM para representar um sinal analógico e como utilizá-lo para comandar o chaveamento de uma etapa de potência, de forma a poder variar a tensão média sobre um motor.

2 Fundamentação Teórica

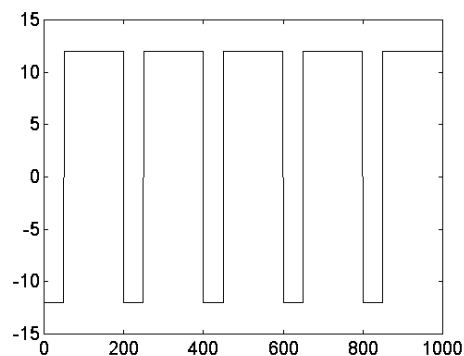
Modulação é a operação de variar uma, ou mais, características (amplitude, frequência, fase, etc.) de um sinal, denominado portadora, em função de outro sinal, chamado sinal modulante.

Em particular, na modulação por largura de pulso (PWM) a portadora é um sinal digital de frequência constante e o sinal modulante é um sinal analógico. O ciclo de trabalho (% do período em que o sinal fica em nível lógico alto) da portadora é variado em função da amplitude do sinal modulante, como mostra a Figura 1.

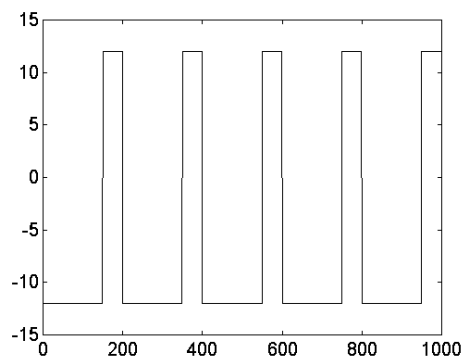
Outra característica importante do sinal PWM é que o seu valor médio é proporcional à amplitude sinal modulante e, portanto, **o sinal modulante pode ser recuperado utilizando-se apenas um filtro passa-baixas. Isso é particularmente interessante para o acionamento de dispositivos de potência, pois, tipicamente, a dinâmica do próprio dispositivo já atua como filtro passa-baixas.** Neste laboratório isto será explorado variando-se a luminosidade de um LED, já que o olho humano filtra as piscadas produzidas pelo PWM e percebe-se apenas o valor médio da luminosidade. Se o mesmo sinal fosse aplicado (através de *drivers* apropriados) à



(a) Ciclo de trabalho de 50%.



(b) Ciclo de trabalho de 75%.



(c) Ciclo de trabalho de 25%.

Figura 1: Sinal PWM.

um motor D.C., o efeito seria o mesmo de variar a tensão aplicada nos seus terminais, pois a indutância e a resistência do enrolamento do motor agem como um filtro passa-baixas e o motor percebe apenas a tensão média.

Note que apesar do sinal PWM ser um sinal digital, o ciclo de trabalho continua sendo uma grandeza analógica e portanto, a princípio, um sinal PWM pode representar um sinal analógico com precisão infinita. No entanto, dependendo de como o sinal PWM é gerado (principalmente quando o modulador é totalmente digital), esta precisão pode ser limitada devido à quantização.

Esta capacidade do sinal PWM de representar um sinal analógico é utilizada em servomotores de pequeno porte, utilizados em aeromodelismo, por exemplo. Tipicamente é utilizado um sinal PWM com período de 20 ms e a largura do pulso é utilizada para representar a referência de posição para o servo. Um pulso de 1500 μs (1520 μs nos servos produzidos pela Futaba) representa a posição neutra do eixo (em geral, o centro da faixa de movimento) e pulsos mais largos represen-

tam posições no sentido anti-horário, enquanto pulsos mais estreitos representam posições no sentido horário. Valores típicos são 2500 μ s para a posição máxima no sentido anti-horário e 500 μ s para a posição máxima no sentido horário, mas isto não é padrão e varia de servo para servo, muitas vezes até entre dispositivos do mesmo modelo.

Os sinais de PWM da Galileo Gen2 são gerados pelo *chip* PCA9685 da NXP¹, que possui 16 canais de PWM com 12 bits de resolução. A frequência dos PWMs (todos usam a mesma frequência) pode ser programada entre 24 Hz e 1525 Hz. Note que a frequência máxima é relativamente baixa para acionamento de motores, além de estar na faixa audível, o que causaria um ruído acústico considerável. Cada saída do PCA9685 pode drenar 25 mA ou fornecer 10 mA a 5V.

Os PWMs são acessados no espaço do usuário através da interface sysfs exposta em `/sys/class/pwm/pwmchip0`. Neste diretório existem os seguintes pseudo-arquivos:

npwm: O número de canais de PWM suportados

export: Exporta um canal de PWM

unexport: “Desexporta” um canal de PWM

device/pwm_period: período do PWM em nanosegundos. No caso do PCA9685 todos os PWMs usam o mesmo período.

Os canais de PWM são numerados de 0 a npwm-1.

Quando um canal de PWM é exportado, será criado um diretório pwmX dentro de `/sys/class/pwm/pwmchip0`, onde X é o número do canal exportado. Os seguintes arquivos existirão no diretório `/sys/class/pwm/pwmchip0/pwmX`:

period: período do PWM em nanosegundos. No caso do PCA9685 todos os PWMs usam o mesmo período, portanto este arquivo não pode ser escrito. O período deve ser configurado através do arquivo `/sys/class/pwm/pwmchip0/device/pwm_period`, como explicado acima.

duty_cycle: duração do ciclo de trabalho do PWM em nanosegundos. Obviamente, deve ser menor do que o período.

polarity: troca a polaridade do sinal de PWM. Funciona apenas se o *hardware* suportar. A polaridade só pode ser trocada com o PWM desabilitado. Os valores possíveis são as *strings*: "normal" ou "inversed".

¹A primeira geração da Galileo utiliza um *chip* CY8C9540A da Cypress e seus PWMs tem resolução menor.

enable: habilita/desabilita o PWM. 0 desabilita, 1 habilita.

Note que apesar do PCA9685 possuir 16 canais de PWM, apenas 6 deles estão disponíveis no conector de *shield* da Galileo Gen2: pwm1, pwm3, pwm5, pwm7, pwm9 e pwm11. As demais saídas do *chip* são usadas internamente na Galileo Gen2.

Neste laboratório será utilizado o pino IO3 do conector de *shield* da Galileo Gen2 acionar um LED. Este pino é mapeado no pwm1. Vide tabela no Moodle <<http://moodle.ece.ufrgs.br>>.

3 Experimentos

1. Logue-se na Galileo como superusuário e crie o grupo pwm e inclua o seu usuário neste grupo.
2. Verifique no mapa de configuração dos pinos da Galileo Gen2 como configurar os multiplexadores para que o sinal pwm1 seja roteado para o pino IO3. Vide tabela no Moodle <<http://moodle.ece.ufrgs.br>>.
3. Faça um *script* de inicialização para configurar o uso do pwm1. Configure permissões de leitura e escrita para o grupo pwm nos arquivos `device/pwm_period`, `duty_cycle` e `enable`.
4. Projete um circuito para interfacear um LED com a Galileo Gen2 utilizando o pino IO3. Projete um resistor que funcione tanto para um *shield* que funcione com 3.3 V quanto com um *shield* que funcione com 5 V. Considere uma corrente máxima no LED de 10 mA e projete o circuito para que o LED acenda quando pino estiver em nível lógico baixo.
5. Monte o circuito com o LED para ser acionado pela Galileo Gen2 ou utilize o LED do *Grove Starter Kit* e faça um programa para acionar o LED através do PWM. Utilize uma frequência de 1 kHz e varie a luminosidade do LED, programando uma rampa de 0 % a 100 % do ciclo de trabalho do PWM.
6. Faça um programa para comandar o servo do *Grove Starter Kit*, mostrado na Figura 2. O valor do ângulo a ser posicionado o eixo do motor (-90° a 90°) deve ser passado como argumento na linha de comando. Utilize um período de 20 ms. Os valores das larguras de pulso para a posição máxima e para a posição mínima devem ser obtidos experimentalmente. A chave `sw1` no *shield* base permite selecionar a tensão `VCC` disponível nos conectores do *shield* base. Para usar o servo, configure-a para 5V. Após usar o servo, configure-a de volta para 3.3V.



Figura 2: Grove servo.

7. Suponha que se deseje acionar um dispositivo que exija 1 A e 10 V. Projete um circuito para que se possa acioná-lo por PWM através da Galileo utilizando transistores bipolares.
8. Repita o item 7 utilizando transistores MOSFET.