

Prática 6:

Controle de motores DC usando PWM e motores de passo

6.1 – Introdução e objetivos

O propósito deste experimento é ilustrar uma aplicação envolvendo modulação por largura de pulso (PWM). Nesta modulação podemos alterar a largura do pulso (tempo em que a onda está em nível alto) mantendo constante o período de saída conforme ilustra a Figura 1. Note que o ciclo de trabalho da onda (duty cycle), que corresponde à relação "tempo em alto/período da onda" pode produzir um valor de potência eficaz que varia com o ciclo de trabalho. A modulação PWM pode ser empregada para outros propósitos, inclusive o de comunicação.

A ideia do experimento é usar a técnica de PWM para controlar a potência de um pequeno motor DC de um cooler a partir da variação do ciclo de trabalho.

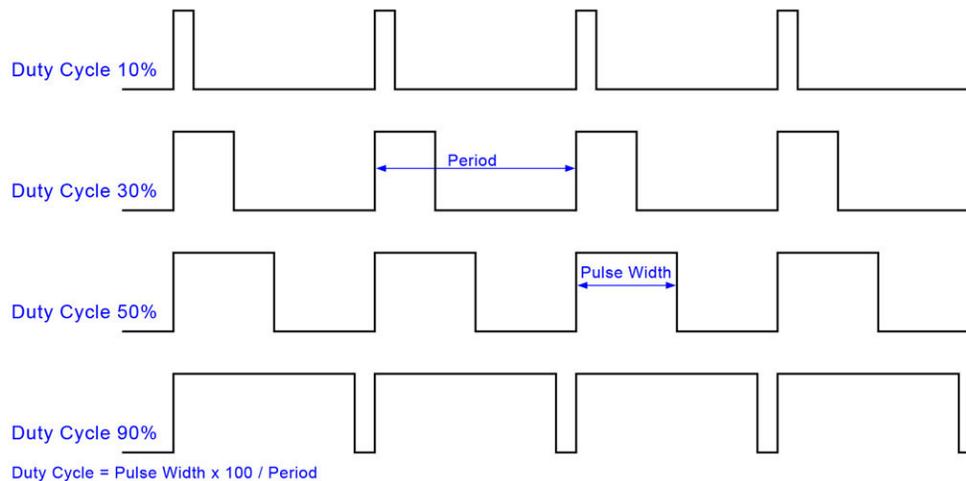


Figura 1 - Ilustração da modulação por largura de pulso (neste caso a frequência permanece fixa) e a variação do ciclo de trabalho para mudar a potência eficaz entregue a um dispositivo. Quanto maior o ciclo de trabalho, maior o tempo em alto e portanto maior a potência entregue ao dispositivo.

7.2 – O circuito

Para esta prática deve ser montado o circuito da Figura 2

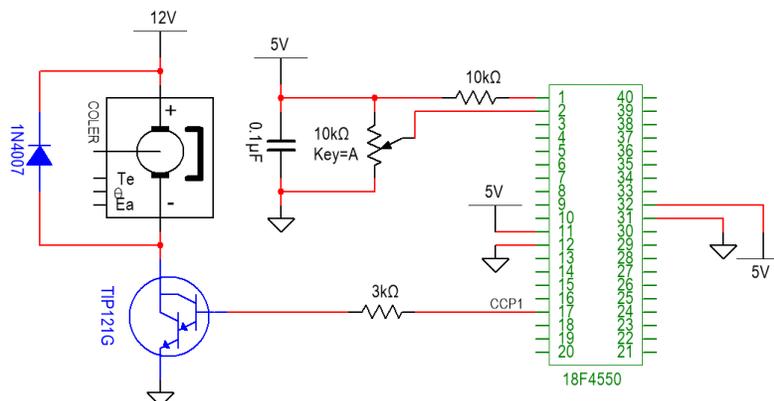


Figura 2 - Circuito



7.3 – O código

Crie um novo programa com o código abaixo. Usando a função `set_pwm1_duty`, cujo argumento deve ser uma variável de 8bits, é possível controlar o ciclo de trabalho da saída PWM1 que no caso do PIC18f4550 é o pino CCP1 (pino 17) deste microcontrolador. Neste código, a ciclo de trabalho é controlado por um potenciômetro cujo valor é lido na linha 19.

```
1  #include "main.h"
2
3  void main()
4  {
5      int16 valor;
6      ...
7      setup_timer_0(RTCC_INTERNAL);
8      setup_timer_1(T1_DISABLED);
9      setup_timer_2(T2_DIV_BY_16, 255, 1);
10     setup_timer_3(T3_DISABLED|T3_DIV_BY_1);
11     setup_ccp1(CCP_PWM);
12     set_pwm1_duty(512);
13     setup_comparator(NC_NC_NC_NC);
14     ...
15     set_adc_channel(0);
16     while(TRUE)
17     {
18         delay_us(20);
19         valor = read_adc();
20         set_pwm1_duty(valor);
21     }
22 }
```