

## Prática L9 – Módulo CCP: modo PWM.

### Objetivos:

- Criar um *firmware* com foco na geração de um sinal PWM.
- Simular o *firmware* no Proteus.
- Testar o firmware no kit didático XM118.
- Agrupar os arquivos gerados para envio pelo Moodle.

### Módulo CCP – Modo PWM

No modo PWM, o módulo CCP pode gerar sinais modulados em largura de pulso (PWM - Pulse Width Modulation), que consiste em uma onda quadrada com período fixo e largura de pulso variável.

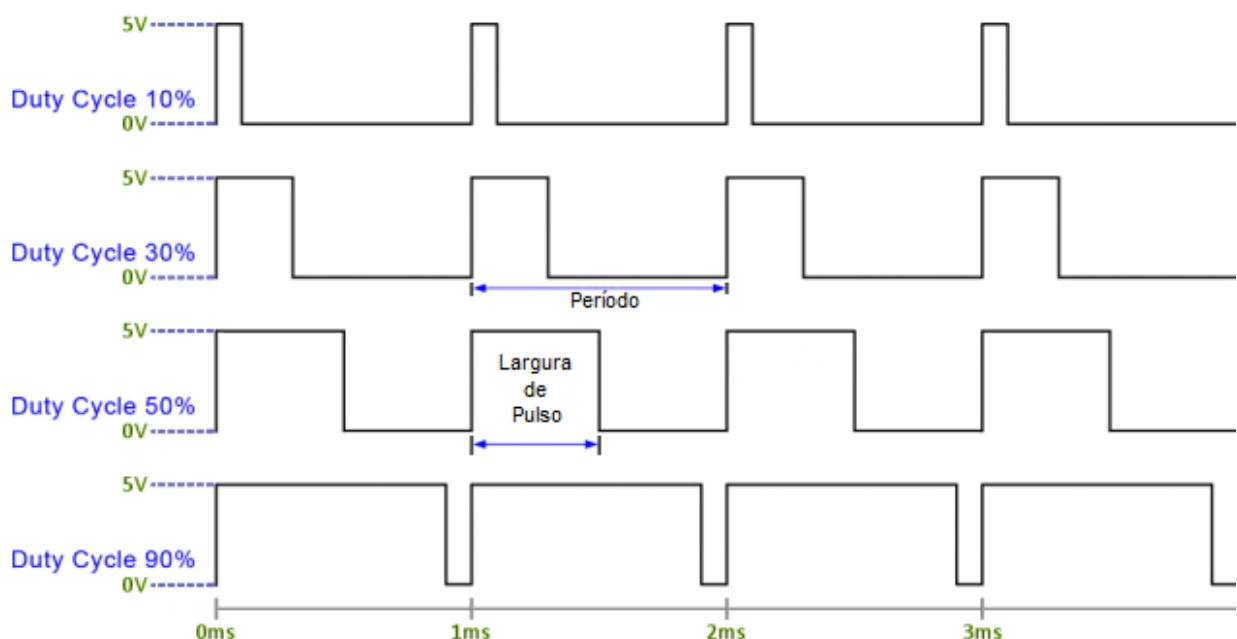


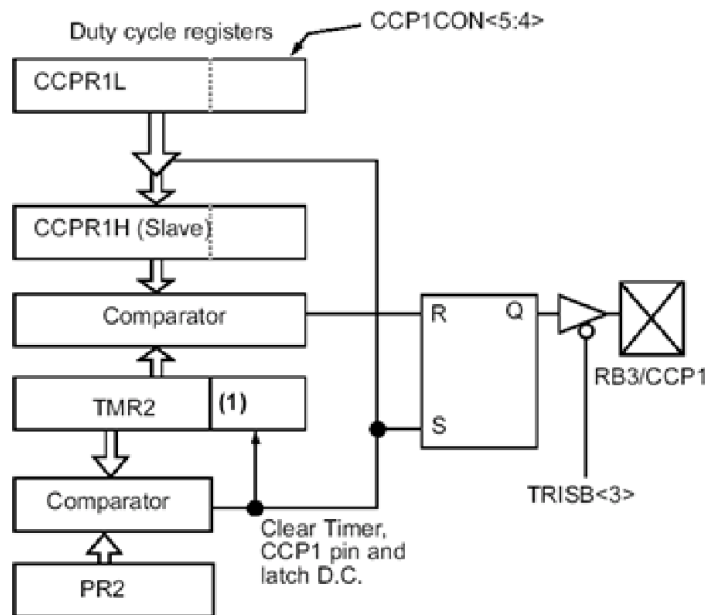
Figura 1: Sinais PWM com diferentes *duty cycles*.

A largura do pulso em estado lógico alto dividida pelo período do sinal define o *duty cycle* (ou ciclo de trabalho) do sinal PWM.

A maior parte das aplicações de PWM para microcontroladores se aproveita da propriedade da energia de um sinal retangular ser proporcional ao seu *duty cycle* (a energia de um sinal está relacionada com a área entre o sinal e o eixo do tempo). Vamos imaginar que um sinal PWM é aplicado a uma lâmpada. Um *duty cycle* de 100% (sinal sempre em '1') fará a lâmpada acender em sua potência máxima; já um sinal com 70% de tempo em alto entrega a lâmpada 70% da potência máxima, e assim por diante. Essa propriedade é utilizada no acionamento de cargas DC, controle de motores, etc.

Outra característica importante do PWM é que, se o sinal for filtrado, podemos obter níveis analógicos proporcionais ao *duty cycle*.

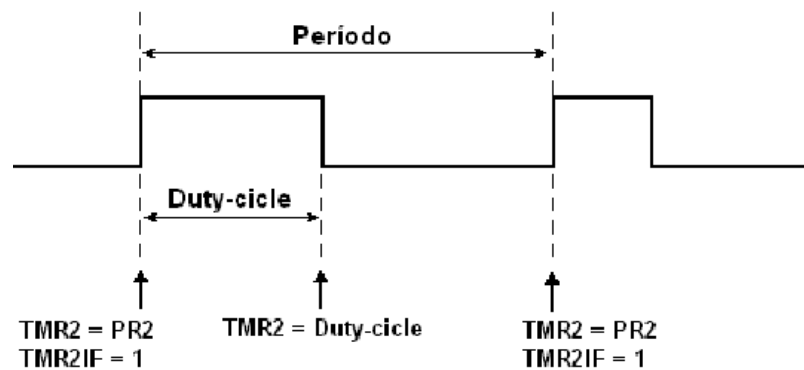
O CCP precisa de uma base de tempo para designar o período do sinal PWM e utiliza o Timer 2 para conseguir essa base. Isso pode ser observado no diagrama de blocos do PWM.



**Figura 2:** Diagrama funcional do módulo CCP em modo PWM.

A geração do sinal PWM é feita da seguinte forma: cada vez que TMR2 coincide com PR2, o pino RB3/CCP1 é setado e TMR2 é reiniciado. Isso nos dá a frequência do sinal. O *duty cycle* é definido comparando o CCPR1H concatenado com os bits 5 e 4 de CCP1CON com TMR2 concatenado com mais dois bits, da pré-escala ou gerados pelos ciclos Q. Quando há a coincidência, o pino RB3/CCP1 é zerado. As concatenações nos dão 10 bits de resolução. Esse processo pode ser observado na figura a seguir.

O registro CCPR1H não é acessível para leitura no modo PWM. O *duty cycle* é configurado através de CCP1L e dos bits 4 e 5 de CCP1CON. Esse valor 'é atualizado em CCPR1H e nos bits de latch a cada período.



**Figura 3:** Diagrama funcional do módulo CCP em modo PWM.

Configurar o PWM se resume a estabelecer sua frequência e seu *duty cycle*. De posse desses dois parâmetros, é possível calcular os valores de cada registro.

O período é o inverso da frequência e é configurado através de PR2 e da pré-escala do TMR2 e pode ser calculado através da equação 1:

$$T_{PWM} = (PR2 + 1) \times 4 \times T_{OSC} \times PS \quad (1)$$

Onde:

$T_{PWM}$  = Período do PWM;

$PR2$  = Registro de comparação do Timer2;

$T_{OSC}$  = Período de um clock do oscilador externo;

$PS$  = pré-escala do TMR2.

Para encontrar o valor de PR2 a partir de um dado período, pode ser usada a equação 2.



Desenvolva um *firmware* capaz de:

- Configurar o CCP2 para gerar na saída RC1 um sinal PWM com frequência de 4 kHz e razão cíclica variável a partir de ajustes do usuário.
- Ajustar a razão cíclica do sinal PWM a partir de comandos do usuário por meio dos botões conectados a RB0 e RB1. Utilize as interrupções externas INT0 e INT1.
  - A cada aperto do botão conectado a RB0, a razão cíclica deve ser incrementada em 10%. Caso haja aperto do botão RB0 e a razão cíclica já for máxima (100%), mantenha-a em 100 %;
  - A cada aperto do botão conectado a RB1, a razão cíclica deve ser decrementada em 10%. Caso haja aperto do botão RB0 e a razão cíclica já for mínima (0%), mantenha-a em 0 %.
- Escrever no display alfanumérico as seguintes informações:
  - Linha 1: 12 Vdc Lâmpada
  - Linha 2: Duty-Cycle = XX%, onde XX é o valor percentual da razão cíclica do sinal PWM em questão.
  - Atualize o LCD somente quando houver modificação no valor da razão cíclica.

Observações sobre o kit didático XM118:

Possui um cristal de 20 MHz.

Para que o sinal de PWM chegue à lâmpada, o pino 4 do *DIP switch* CH3 deve ser ligado.

### **3- Simulação do *firmware* gerado no Proteus.**

Simule o *firmware* no Proteus e apresente o funcionamento ao professor.

### **4- Gravação e execução do código de máquina no microcontrolador**

Grave o código de máquina no microcontrolador e apresente o funcionamento ao professor.

### **5- Envio dos resultados para plataforma Moodle.**

Compacte o diretório de trabalho com o projeto do *firmware* L9 em um arquivo .zip.

Nomeie o arquivo obedecendo o seguinte formato:

{Disciplina}\_{Turma}\_{Grupo}\_{Número do roteiro}.zip

Ex: EC45C\_C51A\_G1\_L9.zip

Envie o arquivo compactado acessando a atividade “L9”.