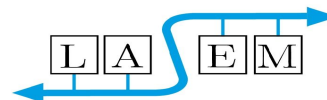

Page 10 of 10

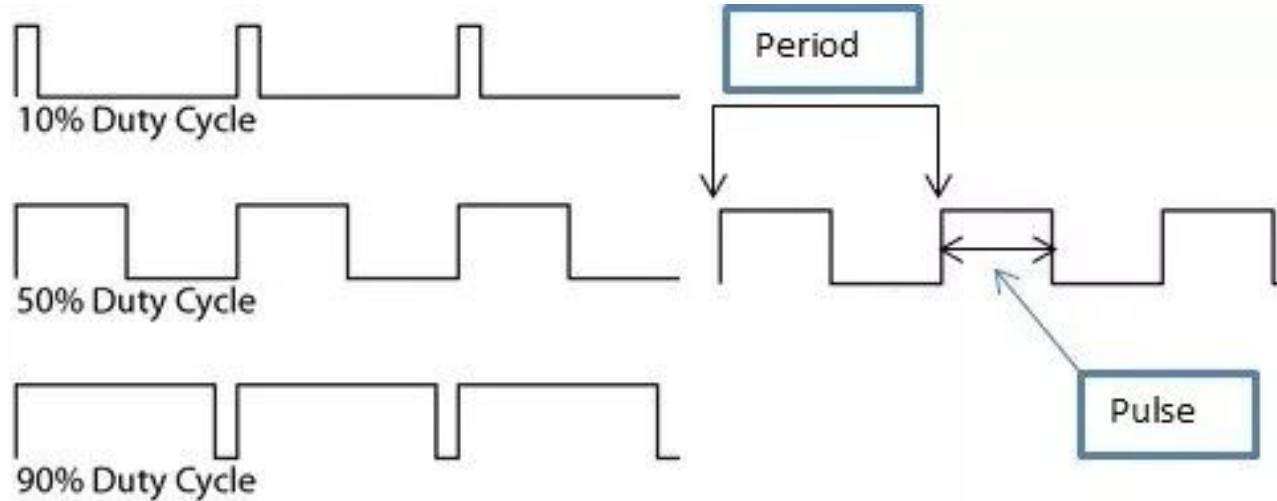


Roteiro

- Modo Compare
 - Conceitos
- Modo PWM
 - Conceitos
 - HandsOn
- Modo Capture
 - Conceitos
 - HandsOn

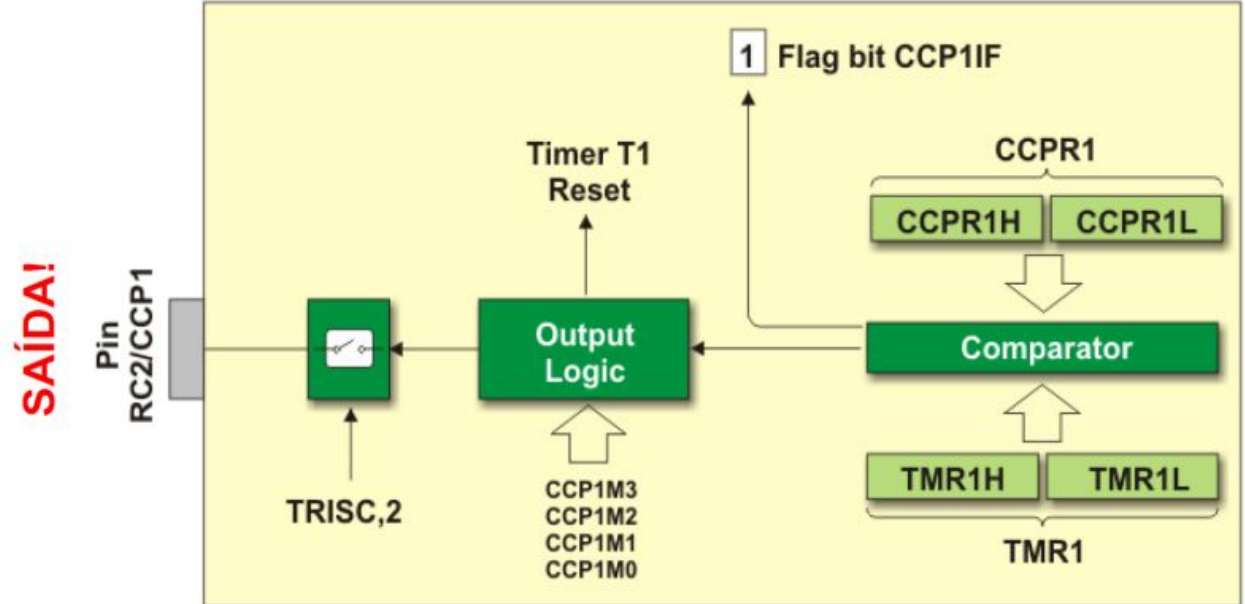
Módulo CCP (Compare/Capture/PWM)

- Medição e Geração de sinais digitais temporizados



Módulo CCP (Compare)

- Gera eventos temporizados
- Base para o PWM
- Base para RTOS

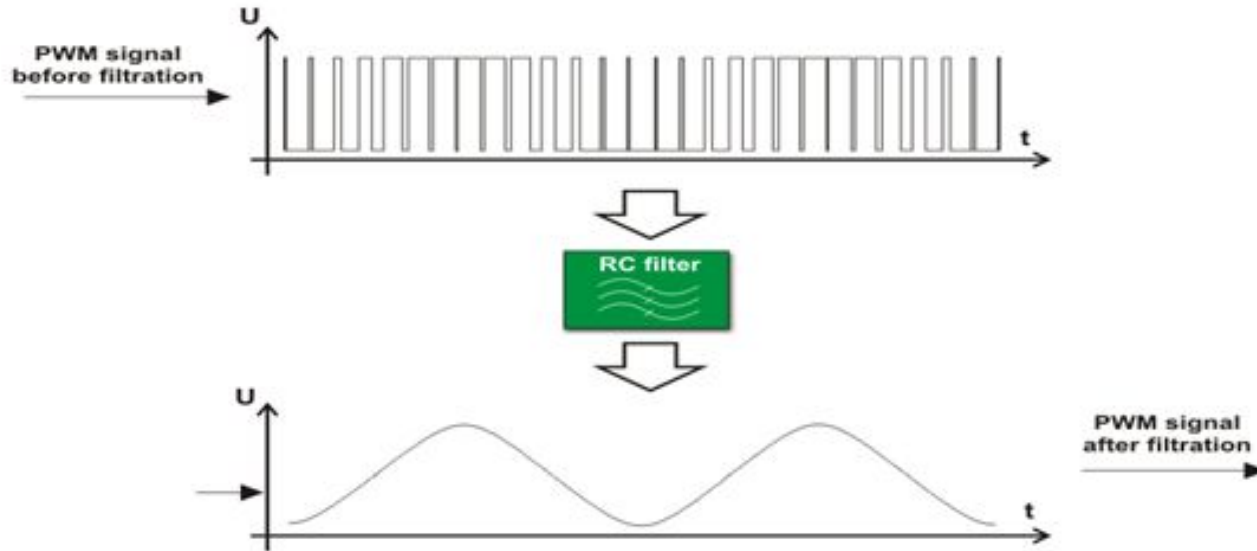


Módulo CCP (Compare)

- Valor do registrador CCPx é constantemente comparado com o valor atual de TMR1. Quando os valores coincidem, dependendo da configuração (CCPxM3–CCPxM0), os eventos escolhidos ocorrem:
 - 1000: (Set): CCPxIF é setado
 - 1001: (Clear): CCPxIF é zerado
 - 1010: (Interrupt): CCPxIF é setado + geração de interrupção
 - 1011: (Special Event): CCPxIF é setado + (CCP1 reseta TMR1 CCP2 reseta TMR1 e inicia conversão A/D

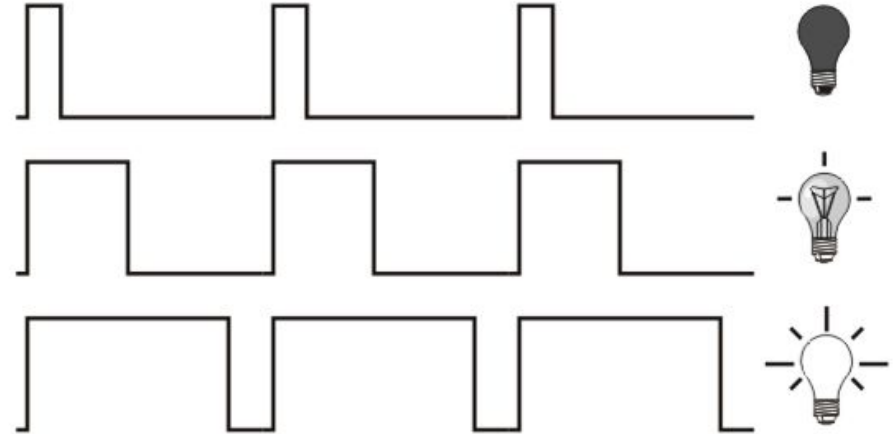
PWM: Pulse Width Modulation

- Maneira digital de interagir com um mundo analógico



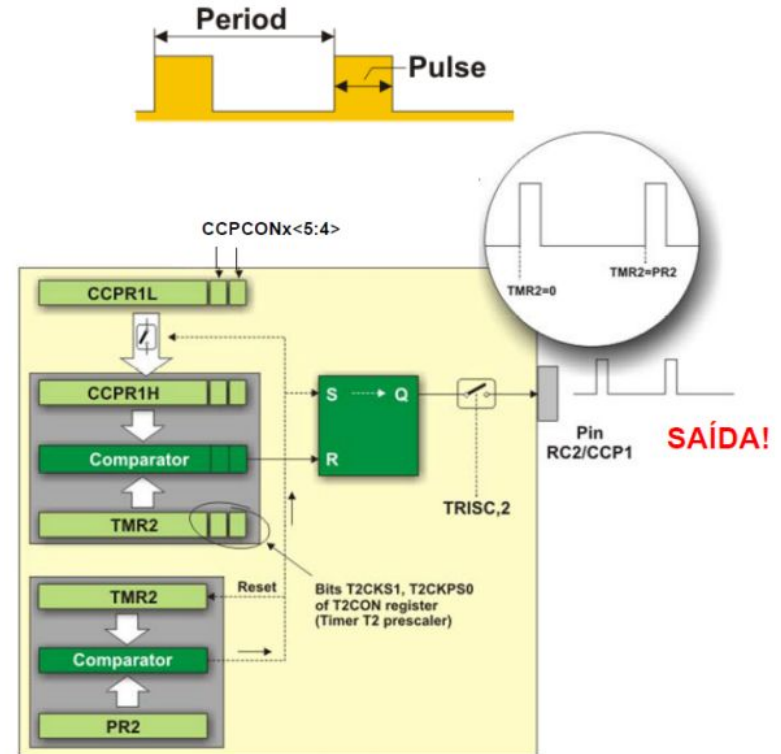
PWM: Pulse Width Modulation

- Gera um sinal com frequência e ciclo de trabalho configuráveis
- Sinal possui um tempo em nível Alto e outro em Nível Baixo
- Em média a intensidade do sinal é controlada



PWM: Pulse Width Modulation

- Utiliza o timer 2
 - 10 Bits de resolução
 - Utiliza o mesmo pino (CCPx) como saída
- Diversas utilidades
 - Acionar Motores
 - Ajustar Intensidade de iluminação



Configuração do CCP (PWM)

- Nosso modelo de PIC (PIC18F4550) possui dois módulos CCP: CCP1 e CCP2
- Frequência do PWM definida pelo timer2
 - CCP1 e CCP2 têm a mesma frequência se ambos forem utilizados dessa forma
 - CCP1 e CCP2 podem ter ciclos de trabalho diferentes
- Utilização no módulo de captura e comparação de sinais (Timer1)
 - Resolução de 16 bits para Capture e Compare

Configuração do PWM

1. Definir o período do PWM a partir do registrador PR2;
2. Determinar o ciclo de trabalho do PWM configurando os registradores CPR1L e CCP1CON<5:4>;
3. Definir CCP1 como Saída Configurando TRISC ;
4. Determinar o valor do prescale do TMR2 e iniciar o timer 2 escrevendo em T2CON;
5. Configurar o Modulo CCP para operação com PWM (pag 145)

Hands-On: PWM

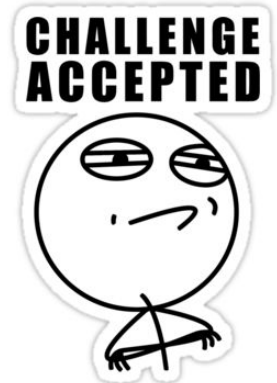
- Fazer o download do header file contendo o protótipo das funções de configuração (pwm.h)
- Implementar o source file (pwm.c)
- Implementar o arquivo main.c:
 - Ler o valor de um potenciômetro (RA0) e refletir esse valor como duty cycle para alimentar o cooler.
 - Exibir o valor nos displays de 7 segmentos
 - Testar no kit



Desafio

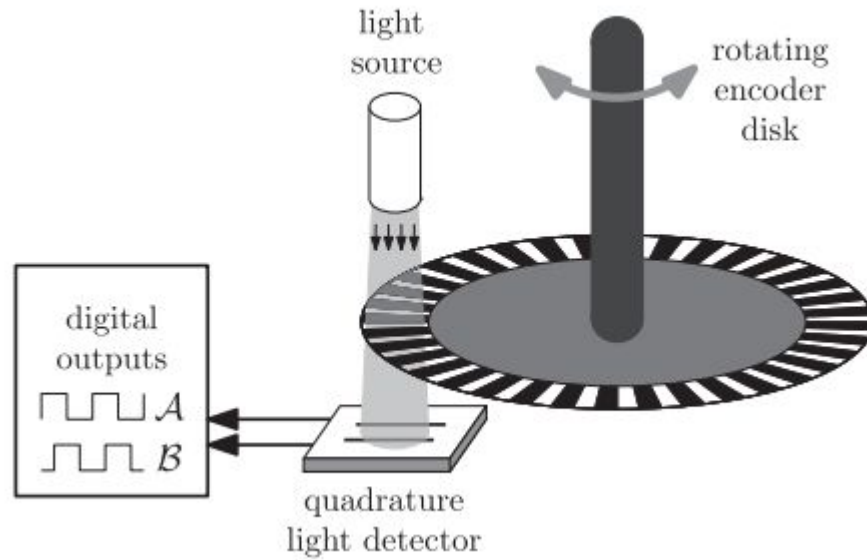
A partir da biblioteca desenvolvida, implementar um projeto de controle de temperatura

- Ler temperatura (sensor conectado ao pino RA0) gerado pelo resistor de potência da placa;
- A partir dessa leitura elaborar uma estratégia de controle para o duty cycle de aquecimento (PWM2 conectado ao resistor) e de resfriamento (Cooler conectado ao CCP1);
- Exibir a temperatura nos displays de sete segmentos



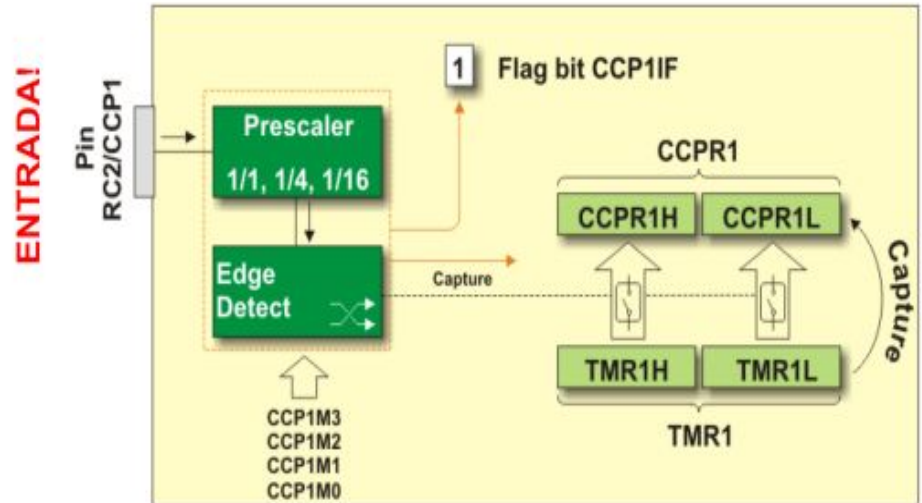
CCP Modo Capture

- Captura e Mede o tempo entre 2 eventos externos



CCP Modo Capture

- Valor do registro TMR1 (Timer 1) é copiado para o registro CCPRx em um dos eventos de borda, configurado por (CCPxM3 – CCPxM0) no CCPxCON:
 - 0100: Borda de descida (1 → 0)
 - 0101: Borda de subida (0 → 1)
 - 0110: Na 4ª borda de subida
 - 0111: Na 16ª borda de subida



HandsOn - Capture

- Fazer o download do header file contendo o protótipo das funções de configuração (pwm.h)
- main.c
 - Capture_Setup ()
 - Desenvolver uma aplicação que leia um sinal analógico (a partir de um potenciômetro) e refletir seu valor na velocidade de um motor. Capturar o sinal do encoder do motor para calcular a velocidade de rotação. Exibir a frequência de rotação em 2 displays de 7 segmentos
 - Simular no Proteus

