Capture/Compare/PWM

O módulo CCP (capture/compare/pwm) é uma funcionalidade do microcontrolador que permite fazer análise detalhada de pulsos externos e também de gerar pulsos PWM.

As funcionalidades deste módulo estão atreladas aos timers:

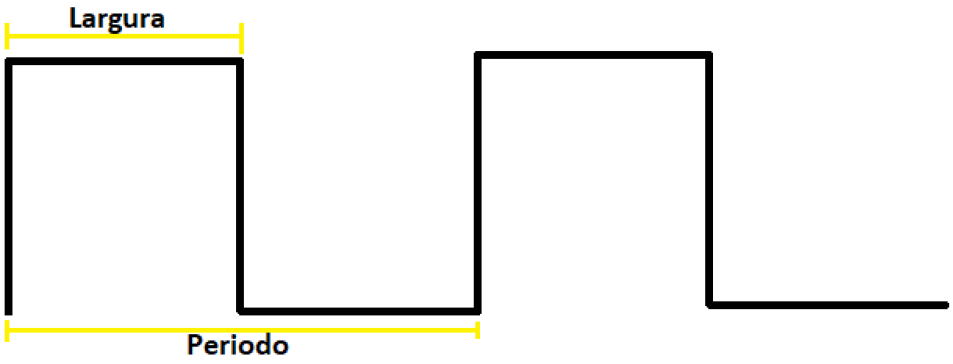
MODO CCP	TIMER COMPARTILHADO
CAPTURE	Timer1 e Timer3
COMPARE	Timer1 e Timer3
PWM	Timer2

No momento que você aciona o módulo CCP, você perde a funcionalidade do timer correspondente a ele. Ou seja, o selecionado será exclusivo para o funcionamento do módulo CCP. O PIC18f45k22 possui 5 módulos CCP. Cada um deles pode ser atrelado a algum timer (menos timer0). Essa escolha do timer é feita utilizando os registradores CCPTMRS0 e CCPTMRS1, onde está definido cada timer que cada módulo consome.

Modo capture

Relacionado a leitura de sinais. É possível medir o período e a largura de pulso de um sinal externo. A medição é feita com base no tempo, podendo ser gerado pelo timer 1. Portanto, saber utilizar o timer 1 corretamente é muito importante neste ponto.

Uma medida muito importante que pode ser adquirida com este periférico é o Duty Cycle. O duty cycle consiste na razão entre a largura do pulso e seu período. Em outras palavras, o tempo que o sinal permaneceu ligado divido pelo tempo total do sinal. Essa razão afeta diretamente a tensão resultante gerada pelo sinal.

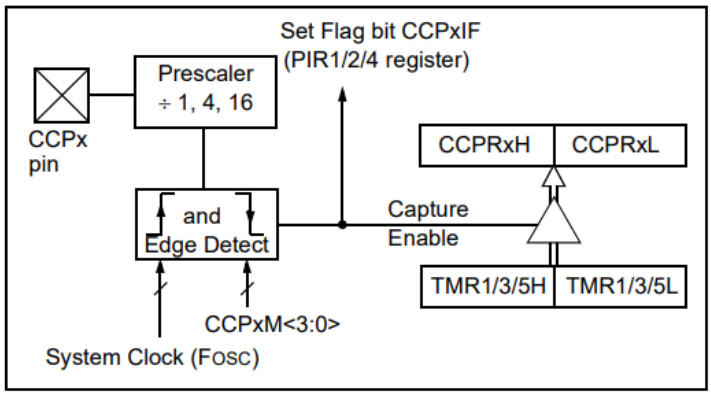


$$DutyCycle = \frac{Largura}{Periodo}$$

Tanto a largura quanto o período são valores obtidor pelo registrador TMR1.

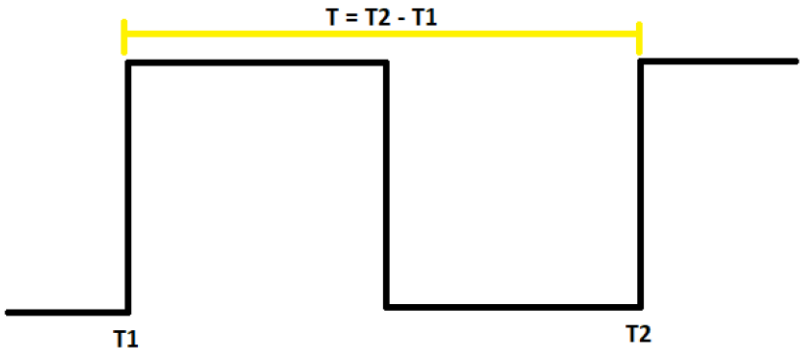
Funcionamento

O módulo Compare identifica automaticamente uma borda no sinal. Pode ser uma borda de decida, subida ou um conjunto de bordas (definido no registrador CCPxCON). Quando essa borda chega, os valores de TMR1L e TMR1H são salvos automaticamente nos registradores CCPR1L e CCPR1H, isto pode ser utilizado para calcular a largura do pulso.



Medindo o período de um sinal com Capture

Você deverá setar interrupções para o Capture. Sempre que uma borda de subida (ou a borda que você selecionou no prescaler) for lida, a interrupção será chamada. Então para ler um período de um sinal, basta salvar o valor nos registradores CCPR (aqueles que copiam o valor do TMR1) em duas interrupções e depois subtrair um pelo outro, assim você terá o período do sinal.



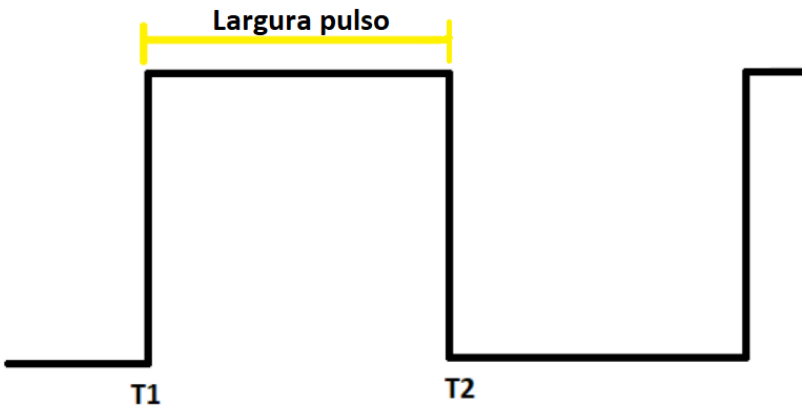
Para realizar este processo, você deve:

- Habilitar as interrupções globais
- Habilitar interrupção pelo módulo CCP
- Configurar o timer1 de forma adequada à situação
- Configurar a função interrupção de forma a capturar T1 e T2

Medindo a largura de pulso com Capture

A única diferença entre o método anterior e este, é configurar a borda de ativação como sendo de descida logo após a primeira captura. Então, no meio da leitura, você alterará a configuração da borda no registrador CCP1CON.

Desta forma, na primeira subida ocorrerá uma interrupção e na primeira descida ocorrerá outra interrupção. Assim, você terá a largura do pulso.



Para todos os casos do modo capture, é preciso se preocupar com a largura mínima de pulso. Cada instrução em nosso sistema leva 0,5 microssegundo para ser executada. Portanto, em frequências muito altas a leitura pode não acontecer apropriadamente pois a largura de pulso é menor que o tempo mínimo que o microcontrolador precisa para executar as instruções.

Encontrado Duty Cycle com módulo compare

O Duty Cycle une os dois conhecimentos que trabalhamos até aqui. Ele consiste na relação entre o período e a largura do pulso, dado em uma porcentagem. Em outras palavras, é a porção ligada do sinal. Precisaremos criar uma função interrupção combinada para extrair tanto o período quanto a largura do sinal.

Mexer com leituras sincronizadas pode gerar todo o tipo de problema, desde imprecisão até quebra total do resultado final. Isso ocorre devido a forma que o microcontrolador executa instruções.

Nós temos que lembrar que a execução das instruções levam tempo (0,5 us em nosso caso), o próprio timer tem seu tempo de overflow que pode ocorrer no meio da leitura. Por isso, tenha em mente que é importante ajustar seu código de acordo com a frequência que irá trabalhar. Para frequências muito altas, talvez seja necessário utilizar um cristal oscilador mais rápido. Vai depender da aplicação.

Adicionando a frequência à leitura

O período deve estar na unidade correta, que é em microssegundos. Até esse momento, trabalhamos com ele como se fosse segundos, algo que não teve problema devido ao tipo de cálculo até agora. Para converter na unidade correta, dividimos período por 1 milhão e depois convertemos para frequência.

```
float f_periodo;
f_periodo = (float) periodo / 1000000;
frequencia = 1./f_periodo;
```

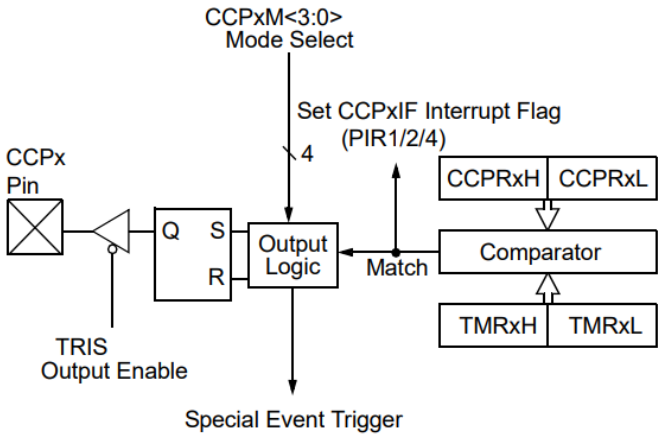
Modo Compare

É um modo interessante para gerar sinais. Com o modo compare, o programador precisa escrever um valor nos registradores CCPR1H e CCPR1L. Quando o

Comparator identificar que estes valores são iguais, uma das ações podem ocorrer:

- 0010: Inverte o pino CCP
- 1000: Seta o pino CCP
- 1001: Limpa o pino CCP
- 1010: Somente gera interrupção
- 1011: Inicia conversão AD (evento especial disponível para este chip)

Esta seleção de funcionalidade fica a cargo do registrador CCP1CON. O diagrama do modo compare é o seguinte:



Gerando um sinal de saída de 100Hz

- Para facilitar, configure o timer para incrementar a cada 1us
- Encontre o período através da frequência (10000us)
- Programe o CCP1CON para togglar na metade deste tempo, assim você terá 5000us de tempo ligado e 5000us de tempo desligado
- Programe a interrupção por CCP e limpe os registradores do timer

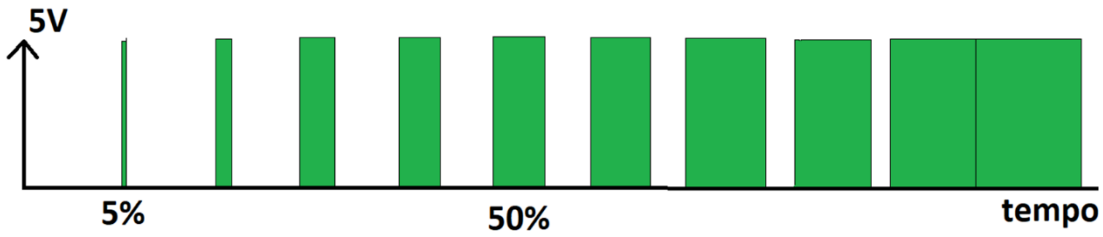
Cuide para não salvar um valor maior que o timer pode atingir. Com incremento de 1us, o tempo máximo que nosso timer atinge é de 65535us, que é bem acima dos 5000us que programamos, então está correto. Limpe os registradores do timer para que a contagem aconteça novamente.

Gerando sinal de 1Khz

O período de 1Khz é 1 milissegundo. Portanto, nas mesmas configurações do exemplo anterior, basta carregar o valor 500 (500 microssegundos, ou 0,5 milissegundo) nos registradores CCPR e já está feito.

Modo PWM

Modulação por largura de sinal. É um tipo de sinal de alta frequência onde você pode fazer o controle do Duty Cycle. Ele possui várias aplicações como: controlar tensão de saída, controlar velocidade, luminosidade, etc.



Durante um PWM a frequência do sinal deve permanecer a mesma, a única coisa que é alterada é a largura do sinal, ou seja, o duty cycle. Essa variação da largura do sinal resulta em aumento ou diminuição da tensão final.

Acionamento do PWM

- Configure o pino atrelado ao PWM como saída
- Defina o modo PWM no registrador CCP1CON
- Definir o Duty Cycle com o registrador CCPR1L
- Configurar o T2CON
- Carregar o PR2 do timer2

Características da utilização do PWM

- Utilizar o PWM compromete a utilização do timer2.
- Quem determina o **período** é o PR2 do TMR2
- Quem determina o **Duty Cycle** é o CCPR1L
- Não é necessário configurar interrupção para zerar o timer2 do PWM. Como vimos no módulo de timer
- Na segunda e última fase, o incremento do TMR2 agora vai ser comparado com o PR2, no momento que esses valores coincidirem, o sinal do CCP1 vai para 1 novamente e o ciclo reinicia.

Encontrando constantes para programar um PWM

As equações estão todas esquematizadas na página 187 do datasheet do PIC18F45k22.

PR2 (Período e frequência)

$$PR2 = \frac{Fosc}{4\ x\ Fpwm * TMR2pre} - 1$$

- PR2 → Registrador
- Fosc → Frequência do oscilador
- Fpwm → Frequência desejada para o PWM
- TMR2pre → Valor do prescaler TMR2

Encontrando o perído a partir de um PR2 definido

$$Tpwm = (PR2 + 1)x\ 4\ x\ Tosc\ x\ TMR2pre$$

- Tpwm → Período do PWM (1/Fpwm)
- PR2 → Registrador do Timer2
- Tosc → Período do oscilador primário (1/Fosc)
- TMR2pre → Valor do prescaler TMR2

CCPRL (duty cycle)

$$CCPRL = \frac{\%DCpwm\ x\ Fosc}{100\ x\ Fpwm\ x\ TMR2pre}$$

- CCPRL deve ficar entre 0 e 1023 !!!
- DCpwm → Duty Dycle desejado
- Fosc → Frequência do oscilador
- Fpwm → Frequência desejada para o PWM
- TMR2pre → Valor do prescaler TMR2

Encontrando Duty Cycle a partir de um CCPRL defindo

$$DC = \frac{CCPRL}{4\ x\ (PR2 + 1)}$$

- DC → Valor do duty cycle (decimal)
- CCPRL → Registrador
- PR2 → Registrador