Vitor Amadeu Souza

Entendendo os

Timers

Com base no C CCS e PIC12F629

© 2022 by Cerne Tecnologia e Treinamento Ltda.

© 2022 by Vitor Amadeu Souza

Nenhuma parte desta publicação poderá ser reproduzida sem autorização prévia e escrita de **Cerne Tecnologia e Treinamento Ltda.** Este livro publica nomes comerciais e marcas registradas de produtos pertencentes a diversas companhias. O editor utiliza as marcas somente para fins editoriais e em benefício dos proprietários das marcas, sem nenhuma intenção de atingir seus direitos.

Março de 2022

Direitos reservados por:
Cerne Tecnologia e Treinamento Ltda
Produção: *Cerne Tecnologia e Treinamento*
E-mail da Empresa: *cerne@cerne-tec.com.br*
Home Page: *www.cerne-tec.com.br.com.br*
Atendimento ao Consumidor: *sac@cerne-tec.com.br*
Contato com o Autor: *vitor@cerne-tec.com.br*



FEITO NO BRASIL

Dedicatória

Como nos meus outros livros, dedico este livro a minha querida esposa Renata Leal.

"Quando falares, tuas palavras devem ser melhores que teu silêncio."

Provérbio árabe

Kits Didáticos e Gravadores da Cerne Tecnologia

A Cerne tecnologia tem uma linha completa de aprendizado para os microcontroladores da família PIC, 8051, Holtek, dsPIC, ARM e etc. Veja os detalhes de cada um nas figuras abaixo:



Kit Cerne PICLAB18F4520

- Display LCD;
- Botões e Leds;
- Entrada de 12 V;
- Medição de RPM;
- Medição de Temperatura.
- Teclado matricial.

Uma linha completa de componentes para o desenvolvimento de seus projetos eletrônicos como displays, PICs, botões, leds, cristais e etc.

Visite a nossa página na Internet e conheça melhor nossos serviços e produtos.

Cerne
Conhecimento para o Desenvolvimento

www.cerne-tec.com.br

Sumário

Introdução **6**

I. Timer 0 **7**

1. Introdução 7

2. Timer 0 7

II. Timer 1 **14**

1. Introdução 14

2. Timer 1 14

Introdução

Este livro é uma continuação da obra PIC12F Programado em C com base no C CCS, onde a proposta é estudar os 2 timers disponíveis neste processador de maneira teórica e prática.

Capítulo I

Timer 0

1. Introdução

Neste capítulo será abordado o estudo sobre o funcionamento do timer de 8 bits do PIC12F, especificamente o Timer 0.

Ao todo há 2 timers no PIC12F sendo um de 8 bits (Timer 0), e um de 16 bits (Timer1) no qual todos eles podem ser utilizados para medir tempos e criar retardos em software, por exemplo.

2. Timer 0

O timer 0 do PIC12 funciona em modo de 8 bits, o que indica que este pode contar de 0 a 255, já que 2^8 é igual a 256. O timer 0 é composto por dois registradores, sendo estes os registradores OPTION_REG e TMR0. A função do registrador TMR0 é armazenar o valor da contagem de 0 a 255, enquanto que a do OPTION_REG é controlar o funcionamento do timer, como prescaler. O funcionamento do timer funciona da seguinte forma: O registrador TMR0 é incrementado a cada ciclo de máquina (pode ser incrementado a cada 1, 2, 4, 8, 16, 32, 64, 128 e 256 ciclos de

máquina, porém por enquanto considere apenas 1 ciclo de máquina). No momento em que o registrador TMR0 passar de 255 para 0 ele reinicia a contagem em 0 para um novo ciclo. Neste momento, ocorre o “estouro” do timer (O estouro do timer é quando o TMR0 passa de 255 para 0). Desta forma, é possível contar tempo ou até mesmo cronometrar alguns eventos. A fórmula genérica para se achar o tempo de estouro do timer 0 é a seguinte:

$$T0_{ESTOURO} = CM * PRESCALER * 256$$

Em que CM é o ciclo de máquina no qual o PIC12F está funcionando, no projeto em questão que utiliza oscilador de 4 MHz o ciclo interno é de 1 µs. O valor de PRESCALER pode ser 1, 2, 4, 8, 16, 32, 64, 128 ou 256 e permite aumentar o tempo de timer, pois se for escolhida a opção 1:4, a cada 4 ciclos de máquina é que ocorrerá 1 incremento no timer.

A seguir alguns exemplos a fim de facilitar o entendimento do timer 0.

1. Encontre o tempo de estouro do timer 0, sabendo que o prescaler está configurado em 1:1 e o ciclo de máquina é de 1 µs.

$T0_{estouro}$= 1µs * 1 * 256

$T0_{estouro}$= 256 µs

2. Determine o tempo de estouro do timer 0 sabendo que o prescaler é 1:4 e o ciclo de máquina é de 1 µs.

$T0_{estouro}$= 1 µs * 4 * 256

$T0_{estouro}$= 1024 µs

3. Verifique o tempo de estouro do timer 0, sabendo que o prescaler é 1:16 e o ciclo de máquina é de 200 ns.

$T0_{estouro}$= 200 ns * 16 * 256

$T0_{estouro}$= 819,2 µs

4. Ache o tempo de estouro do timer 0 sabendo que o prescaler é 1:256 e o ciclo de máquina é de 500 ns.

$T0_{estouro}$= 500ns * 256 * 256

$T0_{estouro}$= 32768µs

Observe se o prescaler for 1:1, a cada ciclo de máquina o timer será incrementado. Já se o prescaler for 1:4, o registrador TMR0 é incrementado a cada 4 ciclos de máquina. Desta forma,

com o prescaler, há como aumentar significativamente o tempo de "estouro" do timer.

Para configurar o PRESCALER do timer 0 fazemos uso da função abaixo do C CCS:

```
setup_timer_0(RTCC_INTERNAL | RTCC_DIV_1);
```

Donde RTCC_DIV_1 refere-se ao PRESCALER podendo ser as outras opções mencionadas antes. Para iniciar o TMR0 com um valor diferente do valor de "estouro", ou seja, o valor 0 podemos usar a função `set_timer_0(VALOR)` em que VALOR pode variar de 0 a 255.

Estando habilitada a interrupção sempre que o valor do TIMER0 chegar ao estabelecido pode-se gerar um tratamento de interrupção naquele instante.

1. Sabendo que o PIC12F está funcionando com ciclo de máquina de 1µs, faça um programa que deixe o timer 0 configurado com prescaler 1:4 e encontre o tempo de estouro sabendo que o valor inicial vale 100.

$$T0_{estouro} = 1\mu s * 4 * (256-100)$$

$$T0_{estouro} = 624\ \mu s$$