

Prática 6 de Sistemas Micro-Controlados

Os microcontroladores pic16f628a possuem implementados em hardware 3 contadores/temporizadores (Timer0, Timer1 e Timer2)

Timer0: O Timer0 é um timer de 8 bits, logo, ele pode armazenar contar até o número 255.

O Timer0 começa a incrementar a partir do zero (ou valor que você programar) e quando ele estiver com o valor 255 e passar para 256 (valor que não cabe em 8 bits) ocorre um "estouro" (overflow) de contagem. Após isso, o flag de interrupção do Timer0 é setado e o Timer0 é zerado, começando assim um novo ciclo de contagem.

Considerando o clock do microcontrolador com sendo de 4Mhz e utilizando-se o prescaler (divisor interno) por um fator de 64, teremos uma frequência de interrupções do Timer0 de 15625 Hz.

Se programarmos a função de tratamento da interrupção do Timer0 para dividir esse sinal por 125, teremos um sinal de exatamente 125Hz. Para programar o Timer0 para dividir o sinal por 125 basta carrega-lo a cada estouro com o valor 131 (256 - 125). Desta forma teremos o um sinal de frequência 1Hz.

Exemplo de código:

```
#include <16f628A.h>
#fuses INTRC,NOWDT,NOPUT,NOBROWNOUT,NOLVP,NOMCLR
#use delay(clock=4000000)

int LED =0;
int conta =0;

#int_timer0          // diretiva para a função do Timer0
void trata_t0()      // tratamento da interrupção do timer0
{
    set_timer0(131 + get_timer0());    // carrega o Timer0 com o valor desejado
    conta++;                          // incrementa contador de estouros

    if (conta==125)                  // se ocorreram 125 estouros inverte o estado do LED
    {
        LED = !LED;
        output_bit(pin_b0,LED);
        conta=0;
    }
} // fim da função de tratamento do Timer0
```

```
void main(void)
{

setup_timer_0 ( RTCC_INTERNAL | RTCC_DIV_64 ); // clock interno e dividido por 64

set_timer0(131);           // carrega o valor 131 no registrador do timer

enable_interrupts (global | int_timer0); // habilita interrupções e libera o timer0

output_B(0);           // zera a porta B

while(true)
{
// seu programa principal
}
}
```

Exercícios:

- 1 – Implemente o hardware necessário para o teste, compile, grave e teste o código;
- 2- Modifique o hardware para que dois LEDs pisquem sob o controle de duas chaves (uma para cada LED).