

NOTAS DE AULAS: MICROCONTROLADORES

TIMER 0

Prof. João Perea Martins
Dep. De Computação, FC–UNESP
E-mail: joao.perea@unesp.br

Programação do TIMER

O TIMER do PIC permite que uma determinada rotina, conceitualmente chamada de **Interrupção do Timer**, seja acionada periodicamente, em intervalos regulares de tempo. Isto é útil para fazer o controle de qualquer sistema que seja acionado em função do tempo, como relógios, alarmes, controle industrial, controle residencial, eletrodomésticos, etc.

A programação do timer depende dos registradores:

- 1) **OPTION_REG**
- 2) **INTCON**
- 3) **TMR0**.

1) Registrador OPTION_REG

Bit7				Bit0			
x	x	TOCs	x	PSA	PS2	PS1	PS0

TOCs: O timer precisa de um clock físico (sequencia constante de pulsos quadrados) para operar. Normalmente usamos o próprio clock do PIC, cuja frequência (F_{osc}) é 4MHz

$TOCs = 0$. O clock do TIMER é $F_{osc}/4$

$TOCs = 1$ O clock do TIMER vem de uma fonte externa colocada em RA4

PSA: O PSA determina onde o Prescaler será usado, para o Timer ou WDT. Neste caso ele será usado para o timer. O prescaler, em eletrônica é um divisor de frequência.

$PSA = 0$ O prescaler (divisor) é associado ao TIMER

$PSA = 1$ O prescaler é associado ao WDT

PS2/PS1/PS0: Estes três bits são usados para efeitos matemáticos, pois definem a grandeza do divisor (prescaler). Determinam o valor de divisão de contagem, conforme tabela abaixo. Esses três bits irão influenciar o cálculo do registrador **TMRO** a ser explicado no item 3 abaixo.

Tabela 1. Bits do Prescaler

PS2	PS1	PS0	Valor do Divisor (Prescaler)
0	0	0	2
0	0	1	4
0	1	0	8
0	1	1	16
1	0	0	32
1	0	1	64
1	1	0	128
1	1	1	256

Exemplo de Configuração Típica: **OPTION_REG** = 'xx0x0yyy'

2) Registrador INTCON

(Está replicado com espelho em todos os bancos)

Bit7				Bit0			
GIE	x	TOIE	x	x	TOIF	x	x

GIE: Permite a ativação individual das interrupções, portanto deve ser ativado.

GIE = 1 As rotinas de interrupções podem ser controladas individualmente

GIE = 0 Todas as rotinas de interrupções serão desativadas

TOIE: Habilita (liga) a interrupção do timer e portanto deve ser ativado.

TOIE = 1 Habilita a interrupção do TIMER

TOIE = 0 Desabilita a interrupção do TIMER

TOIF é um flag que indica a ocorrência de um pedido (acionamento) de interrupção do TIMER. Este flag é acionado por hardware, mas deve ser resetado por software. O seu valor é conforme as especificações:

TOIF = 1 Ocorreu um acionamento de interrupção do TIMER

TOIF = 0 NÃO ocorreu um acionamento de interrupção do TIMER

Exemplo de Configuração Típica por **byte**: $INTCON = '1x1xx0xx'$

Exemplo de Configuração Típica do $INTCON$ **por bit**:

$INTCON, GIE = 1$

$INTCON, TOIE = 1$

$INTCON, TOIF = 0$

Ambas as configurações acima (por byte ou por bit) produzem o mesmo resultado.

3) Registrador TMR0

(Banco 0)

O registrador TMR0 é como se fosse uma variável, cujo valor é automaticamente incrementado pelo PIC em intervalos regulares de tempo.

Quando este registrador sofre um overflow (estouro de contagem), então o flag TOIF é colocado em “1” e ocorre automaticamente um desvio para o endereço 0x0004 que é o endereço das interrupções.

O valor de TMR0 é calculado conforme a equação:

$$N_{int} = \frac{\left(\frac{Clock}{4}\right) / Prescaler}{(256 - TMR0)}$$

Onde:

Nint é o número de vezes que a interrupção do timer será executada por segundo. Essa é uma escolha do programador.

Clock é o clock do PIC que normalmente é 4MHz

Prescaler é o valor do divisor obtido através da tabela 1 acima.

Assim, por exemplo, se o programador definir que serão executadas 125 interrupções por segundo, então ele deve “chutar” um valor do prescaler e então fazer os cálculos. Supondo um prescaler “chutado” como 64, o cálculo ficaria:

$$125 = \frac{\left(\frac{4M}{4}\right) / 64}{(256 - TMR0)}$$

$$TMR0 = 131$$

Isso significa que se o registrador for “carregado” com o valor 131, então a rotina do TIMER0 será acionada 125 vezes por segundo.

4) Exemplo de Configuração dos Registradores

```
; ***** Programação do OPTION_REG
BANKSEL OPTION_REG
MOVLW      B'10000101'    ; bits 7, 6, 4= x
                        ; bit 5: tocs=0, O clock é 4Mhz/4
                        ; bit 3 =0, O prescaler é associado do Timer
                        ; bits 2,1,0 - Prescaler = 64

MOVWF      OPTION_REG

; ***** Programação do INTCON
BANKSEL INTCON
MOVLW      B'10100000'    ; bit 7: GIE = 1  Habilita Ints individualmente
                        ; bit 6, 4, 3, 1, 0 = x
                        ; bit 5: TOIE=1  Habilita a interrupção do Timer
                        ; bit 2: TOIF=0  flag do Timer
INTCON
MOVWF

; ***** Programação do TMRO
MOVLW      D'131'
MOVWF      TMR0
```

4) Exemplo de Programação

O programa abaixo faz um LED conectado ao pino RB7 do PIC16F873 piscar em intervalos de um segundo.

- A interrupção é acionada 125 vezes por segundo.
- A variável TIQUES conta quantas vezes o PIC acionou a interrupção. Depois de 125 acionamentos é que passou exatamente um segundo. Quando isso ocorrer então altero (pisco) o estado do LED, o que é feito com um comando XOR (ou exclusivo) para ficar mais simples.
- O único pino de I/O usado é o RB7 que deve ser definido como saída. Assim o registrador TRISB foi totalmente zerado e todos os pinos definidos como saída. Para facilitar isso no software foi feito simplesmente: `CLRF TRISB`.
- Todas as vezes que a rotina de interrupção do timer é acionada o programa vai para o endereço 4 e neste caso sabemos que houve um estouro no registrador TMR0. Assim, o valor do mesmo (131) dever ser restaurado na rotina do endereço 4.
- O flag TOIF também deve ser obrigatoriamente zerado na rotina de interrupção.
- A diretiva EQU permite definir o endereço de uma única variável.

```
; Programa: Pisca um LED no pino RB7 a cada 1 segundo
; Este programa está intencionalmente sem comentários a fim de forçar o aluno a pesquisá-lo...
; ...Porém, na prática, todo programa deve ter o maior número possível de comentários.
```

```
#include <p16F873.inc>
```

```
TIQUES      EQU    0X20
      ORG    0X0000
      GOTO  INICIO
```

```
;-----
```

```
      ORG    0X4
      BCF    INTCON, T0IF
      MOVLW   D'131'
      MOVWF   TMR0
      DECFSZ  TIQUES, F
      GOTO    FIM_T0
```

```
      MOVLW   B'10000000'
      XORWF   PORTB, F
      MOVLW   D'123'
      MOVWF   TIQUES
```

```
FIM_T0: RETFIE
```

```
; *****
```

```
INICIO:
```

```
      BANKSEL TRISB
      CLRF    TRISB
      BANKSEL PORTB
      BSF     PORTB, 7
```

```
      BANKSEL OPTION_REG
      MOVLW   B'10000101'
      MOVWF   OPTION_REG
```

```
      BANKSEL INTCON
      MOVLW   B'10100000'
      MOVWF   INTCON
```

```
      BANKSEL TMR0
      MOVLW   D'131'
      MOVWF   TMR0
```

```
      MOVLW   D'125'
      MOVWF   TIQUES
```

```
      GOTO    $
```

```
      END
```