



**INSTITUTO FEDERAL**  
Rio Grande do Sul  
Campus Farroupilha

---

# **PIC18F4550: Temporizadores**

Prof. Matheus Ribeiro



# TIMER0 – REGISTRADOR T0CON

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0

- ▶ TMR0ON – Habilita (1) / Desabilita (0) o contador
- ▶ T08BIT – contador de 8 bits (1) ou 16 bits (0)
- ▶ T0CS – fonte de clock: pino RA4/T0CKI (1) ou interno (0)
- ▶ T0SE (somente se T0CS=0) – atualização da contagem na descida (1) ou subida (0) de T0CKI
- ▶ T0PS2:T0PS0 – Configuração do prescaler (se PSA = 0)

000 (1:2)	010 (1:8)	100 (1:32)	110 (1:128)
001 (1:4)	011 (1:16)	101 (1:64)	111 (1:256)



# TIMER0 – MODO TEMPORIZADOR

- ▶ Frequência de clock do temporizador =  $1T_{cy} (F_{osc}/4)$
- ▶ Exemplo: se  $F_{osc} = 4\text{MHz}$ 
  - Frequência do clock do Timer0 =  $1\text{MHz}$  e período  $1\mu\text{s}$
  - Temporização máxima sem uso do prescaler:
    - Para temporizador de 8 bits  $\rightarrow 256 * 1\mu\text{s} = 256\mu\text{s}$
    - Para temporizador de 16 bits  $\rightarrow 65.536 * 1\mu\text{s} \approx 65,5\text{ms}$



# TIMERO – MODO TEMPORIZADOR

- ▶ Se o PRESCALER for configurado, ocorre a divisão na entrada de clock do temporizador.
- ▶ Exemplo: se  $T0PS2:T0PS0 = 111$ , ocorre a divisão do clock por 256
  - Novo clock do temporizador →  $1\text{MHz} / 256 \approx 3.9\text{KHz}$
  - Novo período →  $1\mu\text{s} * 256 = 256\mu\text{s}$
  - Temporização máxima com PRESCALER máximo
    - Para temporizador de 8 bits →  $256 * 256 * 1\mu\text{s} = 65,536\text{ms}$
    - Para temporizador de 16 bits →  $256 * 65.536 * 1\mu\text{s} = 16,78\text{s}$



# TIMER0 – MODO TEMPORIZADOR

- ▶ Indicação do término de contagem
  - Para temporizador de 8 bits: 0xFF → 0x00
  - Para temporizador de 16 bits: 0xFFFF → 0x0000
- ▶ Quando ocorre o reinício da contagem, INTCONbits.TMR0IF recebe 1
- ▶ Antes de iniciar nova contagem, limpar o flag:

```
INTCONbits.TMR0IF = 0;
```



# TIMER0 – MODO TEMPORIZADOR

- ▶ Atribuição do valor inicial da contagem:

- Para temporizador de 8 bits:

- $\text{TMR0L} = 0\text{xFE};$

- Para temporizador de 16 bits:

- $\text{TMR0H} = 0\text{xFF};$

- $\text{TMR0L} = 0\text{xFE};$

- ▶ Também pode ser feita a leitura da contagem atual:

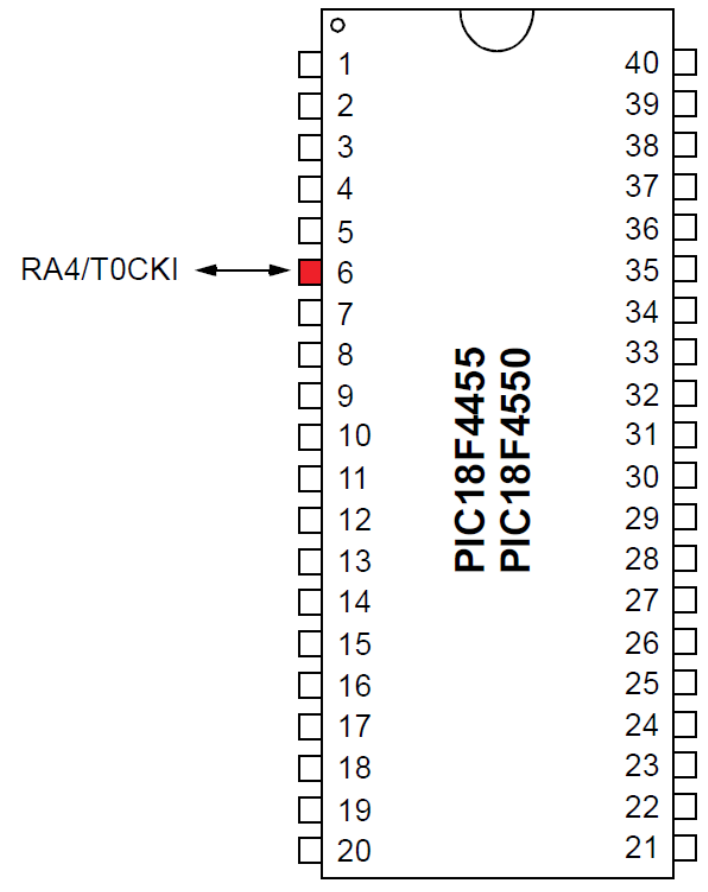
- $\text{VALOR} = \text{TMR0L};$

- $\text{VALOR} = \text{VALOR} + (\text{TMR0H} * 256);$



# TIMER0 – MODO CONTADOR

- ▶ Clock a partir do pino RA4/T0CKI
  - Configurar o pino como entrada digital
  - Configurar se a atualização será na borda de subida ou descida acordo com a aplicação
- ▶ O PRESCALER e os registradores TMR0L e TMR0H operam da mesma forma que para o temporizador



# TIMER0 – EXEMPLO DE INICIALIZAÇÃO

```
// Timer0: ON, 16 bits, clock interno, PRESCALER ativo, 1:32  
T0CON = 0x84;
```

```
// Limpa flag de overflow do Timer0  
INTCONbits.TMR0IF = 0;
```

```
// Carrega valores iniciais do contador (0x85EE = 34.286)  
TMR0H = 0x85;  
TMR0L = 0xEE;
```





# TIMER0 – EXEMPLO DE UTILIZAÇÃO

```
while(1)
{
    // Aguarda atingir o overflow
    while(!INTCONbits.TMR0IF);
    PORTDbits.RD0 = !PORTDbits.RD0;

    // Limpa flag e recarrega os valores iniciais
    INTCONbits.TMR0IF = 0;
    TMR0H = 0x85;
    TMR0L = 0xEE;
}
```



# TIMER1 – REGISTRADOR T1CON

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON

- ▶ TMR1ON – Habilita (1) / Desabilita (0) o contador
- ▶ RD16 – Leitura/escrita em 1 operação de 16 bits (1) ou 2 operações de 8 bits (0)
- ▶ T1RUN – fonte de clock: oscilador interno (1) ou outra fonte (0)
- ▶ T1CKPS1:T1CKPS0 – Configuração do prescaler – 00 (1:1), 01 (1:2), 10 (1:4), 11 (1:8)
- ▶ TMR1CS – modo contador ou economia de energia (1) / temporizador (0)
- ▶ T1SYNC (somente no modo contador) – modo síncrono (0) ou assíncrono (1)
- ▶ T1OSCEN (modo de economia de energia) – habilita (1) ou desliga o oscilador interno (0)

# TIMER1 – MODO TEMPORIZADOR

- ▶ Frequência de clock do temporizador =  $1T_{cy} (F_{osc}/4)$
- ▶ Exemplo: se  $F_{osc} = 4\text{MHz}$ 
  - Frequência do clock do Timer0 =  $1\text{MHz}$  e período  $1\mu\text{s}$
  - Temporização máxima sem uso do prescaler:
    - Somente 16 bits  $\rightarrow 65.536 * 1\mu\text{s} = 65,536\text{ms}$



# TIMER1 – MODO TEMPORIZADOR

- ▶ Se o PRESCALER for configurado, ocorre a divisão na entrada de clock do temporizador.
- ▶ Exemplo: se  $T1CKPS1:T1CKPS0 = 11$ , ocorre a divisão do clock por 8
  - Novo clock do temporizador →  $1\text{MHz} / 8 = 125\text{KHz}$
  - Novo período →  $1\mu\text{s} * 8 = 8\mu\text{s}$
  - Temporização máxima com PRESCALER máximo
    - Somente temporizador de 16 bits →  $8 * 65.536 * 1\mu\text{s} \approx 524,3\text{ms}$



# TIMER1 – MODO TEMPORIZADOR

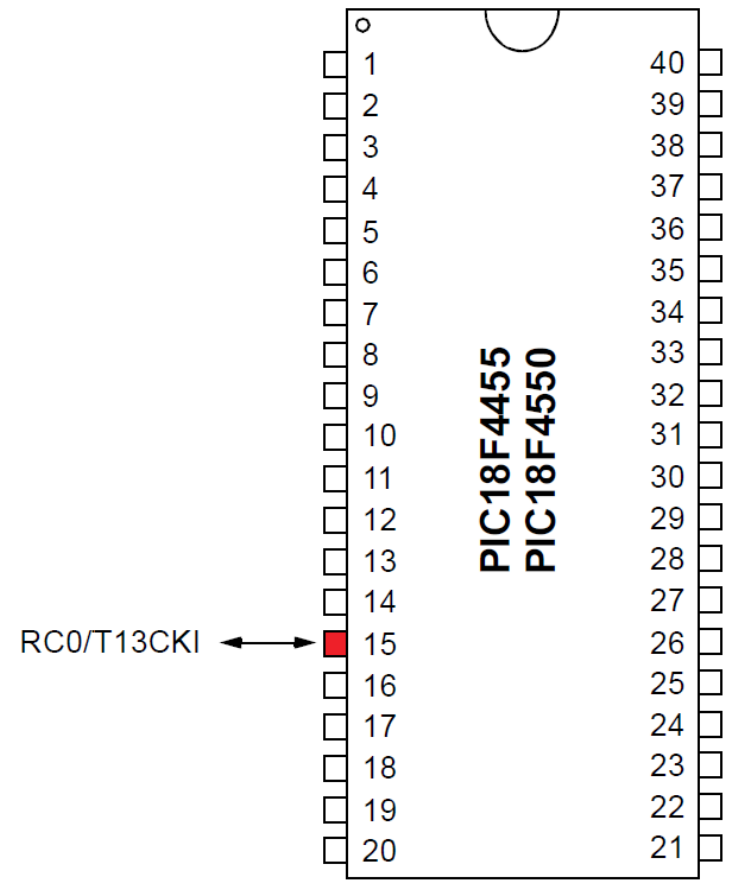
- ▶ Indicação do término de contagem
  - Na transição 0xFFFF → 0x0000
- ▶ Quando ocorre o reinício da contagem, PIR1bits.TMR1IF recebe 1
- ▶ Antes de iniciar nova contagem, limpar o flag:

```
PIR1bits.TMR1IF = 0;
```



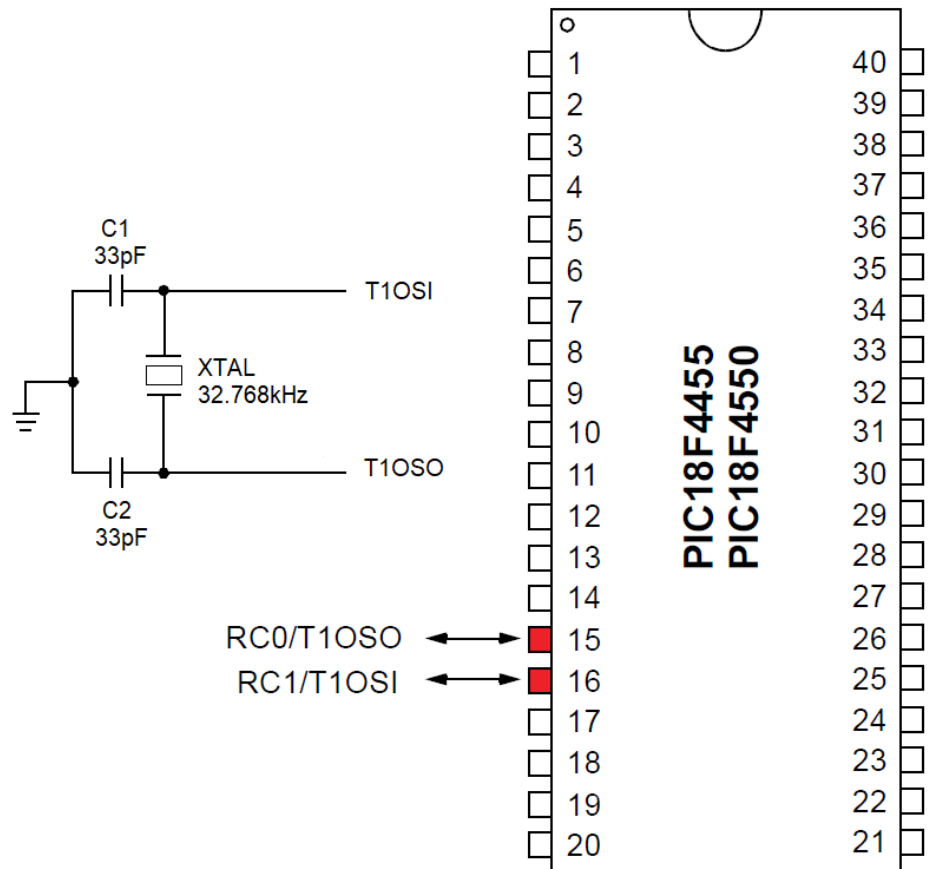
# TIMER1 – MODO CONTADOR

- ▶ Clock a partir do pino RC0/T13CKI
  - Configurar o pino como entrada digital
- ▶ O PRESCALER e os registradores TMR1L e TMR1H operam da mesma forma que para o temporizador



# TIMER1 – OSCILADOR INTERNO

- ▶ Clock é obtida através de um oscilador interno de baixo consumo ajustado para 32KHz
  - É necessário incluir um ressonador (cristal + capacitores) entre os pinos RC0/T1OSO e RC1/OSI.
  - Pinos automaticamente configurados como entradas



# TIMER1 – EXEMPLO DE INICIALIZAÇÃO

```
// Timer1: PRESCALER 1:1, clock Fosc/4  
T1CON = 0x01;
```

```
// Limpa flag de overflow do Timer1  
PIR1bits.TMR1IF = 0;
```

```
// Carrega valores iniciais do Timer1 (0x0000)  
TMR1H = 0x00;  
TMR1L = 0x00;
```





# TIMER1 – EXEMPLO DE UTILIZAÇÃO

```
while(1)
{
    // Aguarda atingir o overflow
    while(!PIR1bits.TMR1IF);

    PORTDbits.RD0 = !PORTDbits.RD0;

    // Limpa o flag e recarrega os valores iniciais
    PIR1bits.TMR1IF = 0;
    TMR1H = 0x00;
    TMR1L = 0x00;
}
```



# TIMER2 – REGISTRADOR T2CON

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0

- ▶ TMR2ON – Habilita (1) / Desabilita (0) o contador
- ▶ T2CKPS1:T2CKPS0 – Configuração do prescaler  
00 (sem divisão)      01 (1:4)      1X (1:16)
- ▶ T2OUTPS3:T2OUTPS0 – Configuração do postscaler

0000 (1:1)	0100 (1:5)	1000 (1:9)	1100 (1:13)
0001 (1:2)	0101 (1:6)	1001 (1:10)	1101 (1:14)
0010 (1:3)	0110 (1:7)	1010 (1:11)	1110 (1:15)
0011 (1:4)	0111 (1:8)	1011 (1:12)	1111 (1:16)



# TIMER2 – MODO TEMPORIZADOR

- ▶ Frequência de clock do temporizador =  $1T_{cy} (F_{osc}/4)$
- ▶ Exemplo: se  $F_{osc} = 4\text{MHz}$ 
  - Frequência do clock do Timer0 =  $1\text{MHz}$  e período  $1\mu\text{s}$
  - Temporização máxima sem uso do prescaler:
    - Somente 8 bits  $\rightarrow 256 * 1\mu\text{s} = 256\mu\text{s}$



# TIMER2 – MODO TEMPORIZADOR

- ▶ Uso do PRESCALER:
  - ▶ Exemplo: se  $T2CKPS1:T2CKPS0 = 11$ , ocorre a divisão do clock por 16
    - Novo clock do temporizador  $\rightarrow 1\text{MHz} / 16 = 62,5\text{KHz}$
    - Temporização máxima:  $16 * 256 * 1\mu\text{s} \approx 4,1\text{ms}$
- ▶ Uso do POSTSCALER:
  - Define o número de “casamentos” entre os registradores TRM2 e PR2.
  - Esta característica é utilizada no gerador de PWM.



# TIMER2 – EXEMPLO DE INICIALIZAÇÃO

```
// Timer2: POSTSCALER 1:16, ON, PRESCALER 1:16  
T2CON = 0xFF;
```

```
// Limpa flag de overflow do Timer2  
PIR1bits.TMR2IF = 0;
```

```
// Carrega valores iniciais do Timer2 (período máximo)  
PR2 = 0xFF;
```



# TIMER2 – EXEMPLO DE UTILIZAÇÃO

```
while(1)
{
    // Aguarda atingir o overflow
    while(!PIR1bits.TMR2IF);

    PORTDbits.RD0 = !PORTDbits.RD0;

    // Limpa o flag e recarrega os valores iniciais
    PIR1bits.TMR2IF = 0;
}
```



# TIMER3 – REGISTRADOR T3CON

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RD16	T3CPP2	T3CKPS1	T3CKPS0	T3CPP1	$\overline{\text{T3SYNC}}$	TMR3CS	TMR3ON

- ▶ TMR3ON – Habilita (1) / Desabilita (0) o contador
- ▶ RD16 – Leitura/escrita em 1 operação de 16 bits (1) ou 2 operações de 8 bits (0)
- ▶ T1CKPS1:T1CKPS0 – Configuração do prescaler  
00 (1:1)      01 (1:2)      10 (1:4)      11 (1:8)
- ▶ TMR3CS – modo contador ou economia de energia (1) / temporizador (0)
- ▶  $\overline{\text{T3SYNC}}$  (somente no modo contador) – modo síncrono (0) ou assíncrono (1)
- ▶ T1OSCEN (modo de economia de energia) – habilita (1) ou desliga o oscilador interno (0)



# TIMER3 – REGISTRADOR T3CON

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RD16	T3CPP2	T3CKPS1	T3CKPS0	T3CPP1	$\overline{\text{T3SYNC}}$	TMR3CS	TMR3ON

- ▶ T3CPP2:T3CCP1  
00 - Timer1 é o clock para CCP1 e CCP2  
01 - Timer3 é o clk para CPP2 e Timer para CCP1  
1x - Timer3 é o clock para CCP1 e CCP2
- ▶ Operação similar ao Timer1





# TIMER3 – EXEMPLO DE INICIALIZAÇÃO

```
// Timer3: 8 bits, PRESCALER 1:1, clock externo  
T3CON = 0x03;
```

```
// Limpa flag de overflow do Timer3  
PIR2bits.TMR3IF = 0;
```

```
// Carrega valores iniciais do Timer3 (0xFFFD = 65.533)  
TMR3H = 0xFF;  
TMR3L = 0xFD;
```



# TIMER3 – EXEMPLO DE UTILIZAÇÃO

```
while(1)
{
    // Aguarda atingir o overflow
    while(!PIR2bits.TMR3IF);

    PORTDbits.RD0 = !PORTDbits.RD0;

    // Limpa o flag e recarrega os valores iniciais
    PIR2bits.TMR3IF = 0;
    TMR3H = 0xFF;
    TMR3L = 0xFD;
}
```

