

Frequência interna do OHF<sup>\*1</sup> é de 48 MHz → 48 000 000 Hz [pagina 139 | 19.2 | 2º paragrafo | 1ª frase]

CLKSEL (clock select) [pagina 138 | tabela 19.1] o relógio do sistema ira derivar do OHF dividido por 2 →  
 $48 \text{ MHz} / 2 = 24 \text{ MHz} \rightarrow 24\,000\,000$

CKCON (clock control) [pagina 225 | tabela 25.1] seleciona o relógio fornecido ao timer 2, como queremos que o valor seja dividido posteriormente esta definição pode ficar por defeito [pagina 225 | tabela 25.1 | bit 4 (T2ML)]

Para utilizarmos o timer 2 no modo 16-bit com auto-reload o bit T2SPLIT do registo TMR2CN terá de ser 0 [pagina 235 | 25.2.1 | 1ª frase]

TMR2CN (timer 2 control) [pagina 239 | tabela 25.9]

Para dividirmos o relógio do sistema por 12 teremos de colocar o bit T2XCLK a 0 (definição por defeito) [pagina 239 | tabela 25.9 | bit 0]

Para inicializar a contagem utilizando o timer 2 teremos de colocar TR2 = 1 [pagina 239 | tabela 25.9 | bit 2]

A ISR será ativada quando TF2H (no modo 16-bit com auto-reload for ativada TF2H = 1), esta flag não é automaticamente limpa por hardware [pagina 239 | tabela 25.9 | bit 7]

TMR2RLL (timer 2 reload register low byte) [pagina 240 | tabela 25.10] guarda o valor do byte menos significativo do valor de reload para o Timer 2

TMR2RLH (timer 2 reload register high byte) [pagina 240 | tabela 25.11] guardar o valor do byte mais significativo do valor de reload do Timer 2.

### Configuração Timer 2

```
CLKSEL |= 0x02;  
CKCON |= 0x00;  
  
T2SPLIT = 0;  
T2XCLK = 0;  
  
TMR2RLL = reload;  
TMR2RLH = reload >> 8;  
  
TR2 = 1;
```

O texto que se encontra rasurado não necessita de ser passado para o código C, pois para obtermos o resultado pretendido poderemos deixar essas configurações por defeito.

Como a configuração a cima apresentada a frequência do relógio do timer 2 será de:

$$48 \text{ MHz} / 2 = 24 \text{ MHz} \rightarrow 24 \text{ MHz} / 12 = \mathbf{2 \text{ Mhz}} \text{ ou } \mathbf{2\,000\,000 \text{ Hz}}$$

E o seu período será de:

$$1 / 2000000 = \mathbf{0.0000005s} \text{ ou } \mathbf{0.5 \mu s}$$

Num registo de 16 bits [0xFFFF → valor máximo] podemos ter 65535 ciclos.

$$1 \text{ segundo} = 1\,000\,000 \text{ us}$$

Se inicializarmos a nossa variável de reload faltando 50 000 ciclos para o transbordo teremos que em cada transbordo terão se passado

$$50\,000 * 0.5 \text{ us} = 25000 \text{ us} \Leftrightarrow \mathbf{0.025 \text{ s}}$$

Obtenção de 1 segundo (mostrar 1 dígito e contagem do tempo):

Assim sendo necessitamos que a flag transborde um total de **40 vezes**

$$1 \text{ transbordo} - 0.025 \text{ segundos}$$

$$x \text{ transbordo} - 1 \text{ segundo}$$

$$x = 40$$

Obtenção de 0.5 segundos (mostrar 2 dígitos):

Assim sendo necessitamos que a flag transborde um total de **20 vezes**

$$1 \text{ transbordo} - 0.025 \text{ segundos}$$

$$x \text{ transbordo} - 0.50 \text{ segundos}$$

$$x = 20$$

Obtenção de 1/3 de segundos (mostrar 3 dígitos):

Como o valor não resultaria num total de transbordos limpos ( 13.(3) vezes) optamos por mostrar o primeiro dígito (aquele que se altera menos vezes) durante 0.20 segundos e os 2 restantes durante 0.40 segundos perfazendo assim 1 segundo:

1 dígito:

A flag terá de transbordar **8 vezes**

2 e 3 dígito:

A flag terá de transbordar **16 vezes**

$$1 \text{ transbordo} - 0.025 \text{ segundos}$$

$$x \text{ transbordo} - 0.40 \text{ segundos}$$

$$x = 16 \text{ vezes}$$

Obtenção de 0.25 segundos (mostrar 4 dígitos):

Assim sendo necessitamos que a flag transborde um total de **10 vezes**

$$1 \text{ transbordo} - 0.025 \text{ segundos}$$

$$x \text{ transbordo} - 0.25 \text{ segundos}$$

$$x = 10$$

Obtenção de 0.20 segundos (mostrar 5 dígitos):

Assim sendo necessitamos que a flag transborde um total de **8 vezes**

$$1 \text{ transbordo} - 0.025 \text{ segundos}$$

$$x \text{ transbordo} - 0.20 \text{ segundos}$$

$$x = 8$$