

# Microcontroladores

## *Trabalho Prático 3*

**Carlos Abreu,**

Instituto Politécnico de Viana do Castelo, [cabreu@estg.ipvc.pt](mailto:cabreu@estg.ipvc.pt)

Instituto Politécnico de Viana do Castelo  
Escola Superior de Tecnologia e Gestão  
2024

Carlos Abreu  
[www.estg.ipvc.pt/~cabreu](http://www.estg.ipvc.pt/~cabreu)

**Curso:**

CTeSP em Sistemas Eletrónicos e Computadores



## Objetivo Pedagógico

Compreender a arquitetura dos *timers* do *PIC18F4580*. Saber configurar e utilizar os *timers* do *PIC18F4580* no desenvolvimento de software utilizando diferentes técnicas de programação.

### Sumário:

**Duração:** 5 horas

1. Arquitetura dos *Timers* no *PIC18F4580*
2. Utilização dos *timers* com recurso a polling
3. Utilização das interrupções geradas pelos *timers*

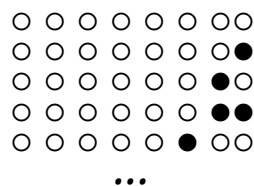
## 1. Introdução

Antes de iniciar a realização deste trabalho prático estude a secção 11.0 do datasheet do *PIC18F4580*, referente ao Timer 0. De seguida, utilizando as ferramentas de desenvolvimento ao seu dispor realize as seguintes tarefas.

## 2. Timers - Exercícios

**Exercício 1** Desenvolva um contador binário de 8 bits usando os LEDs presentes no seu Kit de desenvolvimento. O contador inicia a contagem em  $0x00$  (i.e., todos os LEDs no estado "OFF") e termina em  $0xFF$  (i.e., todos os LEDs no estado "ON"). O contador deverá incrementar o seu valor até chegar ao máximo e de seguida reinicia a contagem a partir do zero. Os valores lógicos "1" e "0" são representados pelos LEDs no estado "ON" (●) e "OFF" (○), respectivamente.

*Exemplo:*



A contagem de tempo entre incrementos consecutivos do contador deverá ser realizada pelo Timer 0, com recurso a polling.

**Nota:** Complete a seguinte função de forma a obter atrasos de duração igual a  $t$  ms. Justifique todas as opções que tomar. Descreva de forma clara e sucinta o funcionamento da função e comente o código linha a linha.

```
void my_delay_ms(unsigned int t)
{
    TOCONbits.T08BIT = ____;
    TOCONbits.T0CS   = ____;
    TOCONbits.PSA     = ____;
    TOCONbits.T0PS    = ____;
    TOCONbits.TMR0ON  = ____;
    while(t)
    {
        TMR0 = ____;
        while(!INTCONbits.TMR0IF);
        INTCONbits.TMR0IF = ____;
        t--;
    }
    return;
}
```

**Exercício 2** Modifique o software que desenvolveu no exercício anterior de forma a utilizar as interrupções do timer em vez de polling. Desenvolva uma rotina de serviço às interrupções de alta prioridade que permita substituir a função `my_delay_ms(unsigned int t)`.

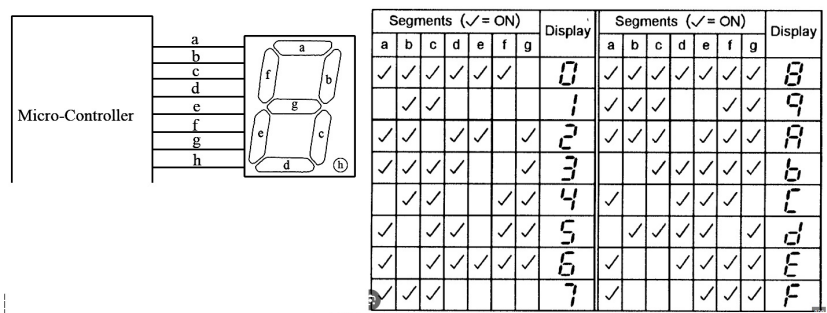
---

**Nota:** Consulte as secções 9.0 e 11.0 do datasheet do *PIC18F4580*.

---

**Exercício 3** Desenvolva um software que use o *Timer 0* para gerar uma onda quadrada com uma frequência de  $2^n$  Hz no pino *RD0*. Inicialmente o valor de  $n$  é 0. Sempre que o utilizador prime o Botão 1 o valor de  $n$  incrementa uma unidade, até ao máximo de 5. A frequência da onda quadrada volta ao valor inicial sempre que o valor de  $n$  for superior a 5. Em simultâneo, deverá mostrar no display de 7-segmentos (Figura 1) o valor de  $n$ .

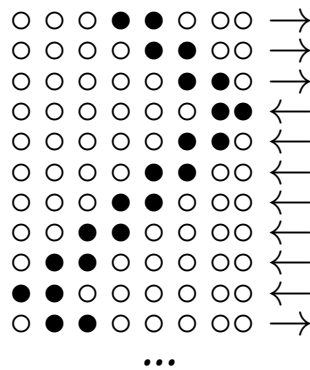
**Importante:** Não poderá usar polling neste exercício.



**Figura 1**

Tabela de verdade para o display de 7-Segmentos.

**Exercício 4** Usando o Timer 0 e as suas interrupções, desenvolva um software que faça um varrimento aos LEDs do seu kit de desenvolvimento conforme este exemplo:



**Nota:** Explore a utilização do operador de shift bit a bit.

O sentido do varrimento é indicado pelas setas à direita. A alteração do estado dos LEDs dá-se a cada 25 ms.