

Prática L6 – Display de 7 segmentos.

Objetivos:

- Criar um *firmware* com foco no controle de displays de 7 segmentos.
- Simular o *firmware* no Proteus.
- Testar o firmware no kit didático XM118.
- Agrupar os arquivos gerados para envio pelo Moodle.

Display de 7 segmentos.

Os displays de 7 segmentos são displays formados por conjuntos de LEDs dispostos de forma a apresentar a figura de um dígito decimal quando ligados ordenadamente.

Para simplificar as conexões, os anodos ou catodos de todos os LEDs são conectados a um terminal comum do display, caracterizando o display como catodo comum ou anodo comum, respectivamente.

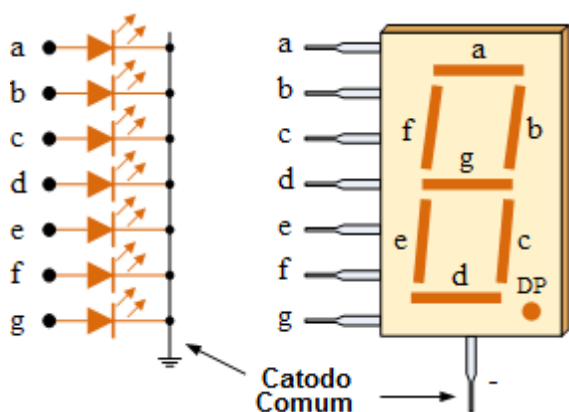


Figura 1a: Display Catodo Comum

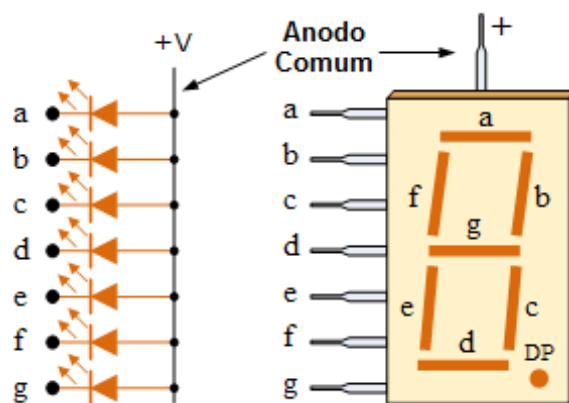


Figura 1b: Display Anodo Comum

A exibição dos dígitos é feita ao acender determinados segmentos enquanto outros são mantidos apagados. A figuras 2 apresenta a ligação dos LEDs relacionando a posição de cada um de seus segmentos.

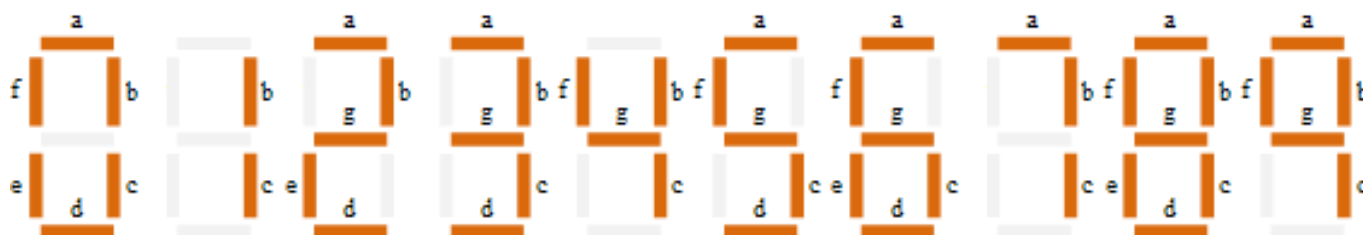


Figura 2: Dígitos criados com segmentos de LEDs.

Configuração das portas de entrada e saída.

As portas de I/O no PIC18F4550 compartilham seus pinos com outros periféricos como entrada do conversor A/D e saídas do módulo de comunicação serial. Desta forma, para que as portas funcionem de acordo com o desejado, em alguns casos, há a necessidade de configurar outros registradores além do TRIS.

Quando não se deseja usar o conversor A/D e em seu lugar deseja-se usa as portas de I/O digitais, deve-se configurar o registrador ADCON1 com os quatro bits menos significativos em nível alto. Esta configuração garante que todo o PORT A seja configurado como porta de I/O digital. Ex:

```
ADCON1 = 0x0F;           // PORTA configurado como I/O digital
```

Os possíveis valores de configuração do registrador ADCON1 em função de seus quatro bits menos significativos são:

PCFG3: PCFG0	AN12	AN11	AN10	AN9	AN8	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
0000	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	A	A	A	A	A	A	A	A	A
0111	D	D	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D

A = Analog input

D = Digital I/O

Para que o PORT B inicialize com a função de porta de I/O digital, deve-se utilizar a seguinte configuração em nível de pré-processamento:

```
#pragma config PBDEN = OFF           // Pinos do PORTB começam como digitais
```

Como pode ser verificado no esquema elétrico do kit didático XM118, os botões SW12 e SW13, que estão respectivamente ligadas à RB0 e RB1, não possuem resistores de *Pull-Up* externos. Neste caso, deve-se ligar os resistores de *Pull-Up* internos. Para isso o bit RBPU do registrador INTCON2 deve ser apagado. Ex:

```
INTCON2bits.RBPU = 0;           // Ativa resistores de Pull-Up para o PORT B
```

Maiores informações sobre estas configurações estão disponíveis na seção 10 do *datasheet* do PIC18F4550.

ATIVIDADES

1- Criar o circuito de simulação.

Acesse a plataforma Moodle e faça o *download* do arquivo: L6.pdsprj

Salve o arquivo no seguinte diretório de trabalho:

{Disciplina}_{Turma}_{Grupo}_{Número do roteiro}

Ex: C:\EC45C_C51A_B1_L6

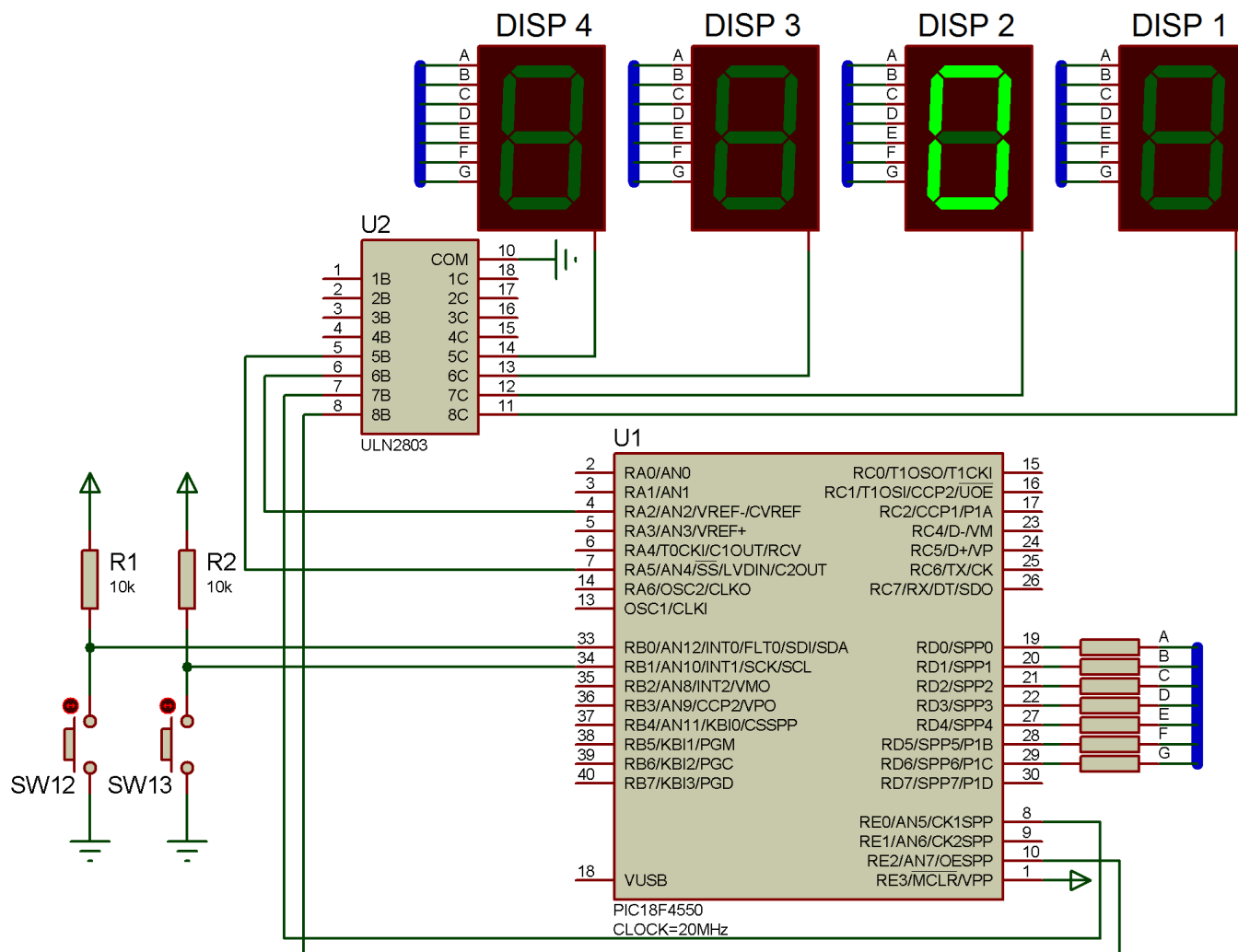


Figura 3: Circuito da atividade L6.

2- Desenvolvimento do *firmware*.

Crie um novo projeto no MPLAB X denominado L6 e salve-o no diretório de trabalho criado na atividade acima.

Desenvolva um *firmware* capaz de apresentar o valor de uma variável com valor máximo de 9999 nos quatro displays de 7 segmentos disponíveis. Esse valor deve ser incrementado por RB0 e decrementado por RB1. O incremento deve ser de 10 unidades e o decremento deve ser de 5 unidades. O tempo de varredura de cada display deve ser de aproximadamente 5 ms.

Observações sobre o kit didático XM118:

Possui um cristal de 20 MHz. Os displays de 7 segmentos são do tipo catodo comum. Os botões SW12 e SW13 controlam os pinos de I/O RB0 e RB1, respectivamente. Se pressionados impõem nível lógico 0, mas não possuem resistores de *Pull-Up* externos.

Utilize o seguinte código para iniciar seu projeto:

```
#include <xc.h>

// Configurações
#pragma config PLLDIV = 5           // PLL para 20MHz
#pragma config CPUDIV = OSC1_PLL2  // PLL desligado
#pragma config FOSC = HS            // Fosc = 20MHz; Tcy = 200ns
#pragma config WDT = OFF            // Watchdog timer desativado
#pragma config PBADEN = OFF         // Pinos do PORTB começam como digitais
#pragma config LVP = OFF            // Desabilita gravação em baixa tensão
#pragma config DEBUG = ON           // Habilita debug
#pragma config MCLRE = ON           // Habilita MCLR e desabilita RE3 como I/O

#define _XTAL_FREQ 20000000         // uC opera com cristal de 20 MHz
```

3- Simulação do *firmware* no Proteus.

Simule o *firmware* no Proteus e apresente seu funcionamento ao professor.

4- Gravação e execução do código de máquina no microcontrolador.

Grave o código de máquina no microcontrolador e apresente o funcionamento para o professor.

5- Envio dos resultados para plataforma Moodle.

Compacte o diretório de trabalho com o projeto do *firmware* L6 em um arquivo .zip.

Nomeie o arquivo obedecendo o seguinte formato:

{Disciplina}_{Turma}_{Grupo}_{Número do roteiro}.zip

ex: EC45C_C51A_G1_L6.zip

Envie o arquivo compactado acessando a atividade “L6” no Moodle.