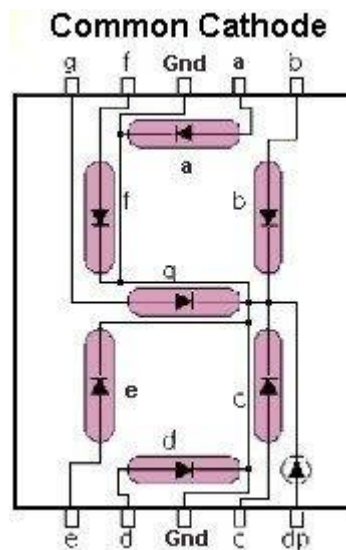




Faculdade de Computação e Engenharia Elétrica
Microcontroladores e Microprocessadores – Prof. Dr. Elton Alves
Experimento 10 – Projeto em Linguagem C no PIC16F628A

- **Objetivos:**

- Realizar o desenvolvimento de programas em linguagem C, simulando semáforos acionados por botões;
- Utilizar display de 7 segmentos do tipo cátodo comum (Figura a seguir) com PIC16F628A.



➤ **Criar um projeto 1 (Semáforo com Botão) para compilar o arquivo fonte, no software Mplab:**

```
#include <xc.h>
```

```
#pragma config FOSC = INTOSCIO      // Oscillator Selection bits (HS oscillator:  
High-speed crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN)  
#pragma config WDTE = OFF           // Watchdog Timer Enable bit (WDT disabled)  
#pragma config PWRTE = OFF          // Power-up Timer Enable bit (PWRT disabled)  
#pragma config MCLRE = OFF          // RA5/MCLR/VPP Pin Function Select bit  
//(RA5/MCLR/VPP pin function is MCLR)  
#pragma config BOREN = OFF          // Brown-out Detect Enable bit (BOD disabled)  
#pragma config LVP = OFF            // Low-Voltage Programming Enable bit  
//(RB4/PGM pin has digital I/O function, HV on MCLR must be used for  
//programming)  
#pragma config CPD = OFF            // Data EE Memory Code Protection bit (Data  
//memory code protection off)  
#pragma config CP = OFF             // Flash Program Memory Code Protection bit  
//(Code protection off)
```

```

/*
 *
 */
#define _XTAL_FREQ 4000000

#define LED1 RB0
#define LED2 RB1
#define LED3 RB2
#define bot1 RA0

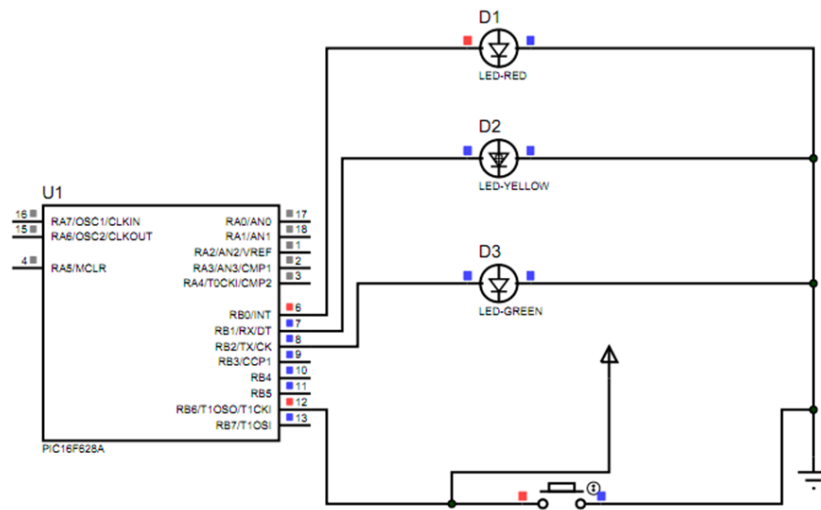
void main (){
    PORTB=0x00; //0
    PORTA=0x00; //0
    TRISB=0x00; //0
    TRISA=0xFF; // Configura para ser Entrada 1
    while(1)
    {
        LED1=1;
        LED2=0;
        LED3=0;
        __delay_ms(500);
        LED1=0;
        LED2=1;
        LED3=0;
        __delay_ms(500);
        LED1=0;
        LED2=0;
        LED3=1;
        __delay_ms(500);

        //botão acionado
        if((bot1==0)){
            __delay_ms(100);
            if (bot1==0){
                LED1=1;
                LED2=0;
                LED3=0;
                __delay_ms(3000);

            }
            bot1=1;
        }
    }
}

```

- Montar o circuito eletrônico a seguir, no software Proteus, e testar o código do projeto 1:



- Criar um projeto 2 (para compilar o arquivo fonte):

```
#include <xc.h>
```

```
#pragma config FOSC = INTOSCIO      // Oscillator Selection bits (HS oscillator:
High-speed crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN)
```

```
#pragma config WDTE = OFF           // Watchdog Timer Enable bit (WDT disabled)
```

```
#pragma config PWRTE = OFF          // Power-up Timer Enable bit (PWRT disabled)
```

```
#pragma config MCLRE = OFF          // RA5/MCLR/VPP Pin Function Select bit
//(RA5/MCLR/VPP pin function is MCLR)
```

```
#pragma config BOREN = OFF          // Brown-out Detect Enable bit (BOD disabled)
```

```
#pragma config LVP = OFF            // Low-Voltage Programming Enable bit //(RB4/PGM
pin has digital I/O function, HV on MCLR must be used for //programming)
```

```
#pragma config CPD = OFF            // Data EE Memory Code Protection bit (Data
//memory code protection off)
```

```
#pragma config CP = OFF
```

```
#define _XTAL_FREQ 4000000
```

```
unsigned char const mapa_segmento[10] =
```

```
{0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F};
```

```
//Variaveis que representam os números.
```

```

int main()
{
    int contador; // Variavel usada para incremento e controle dos numeros que aparecem
no display (de 0 a 9)

    TRISB = 0; //Define todas as portas B como saída
    PORTB = (mapa_segmento[0]); // Força o display a apresentar o número 0
(Inicio do programa)

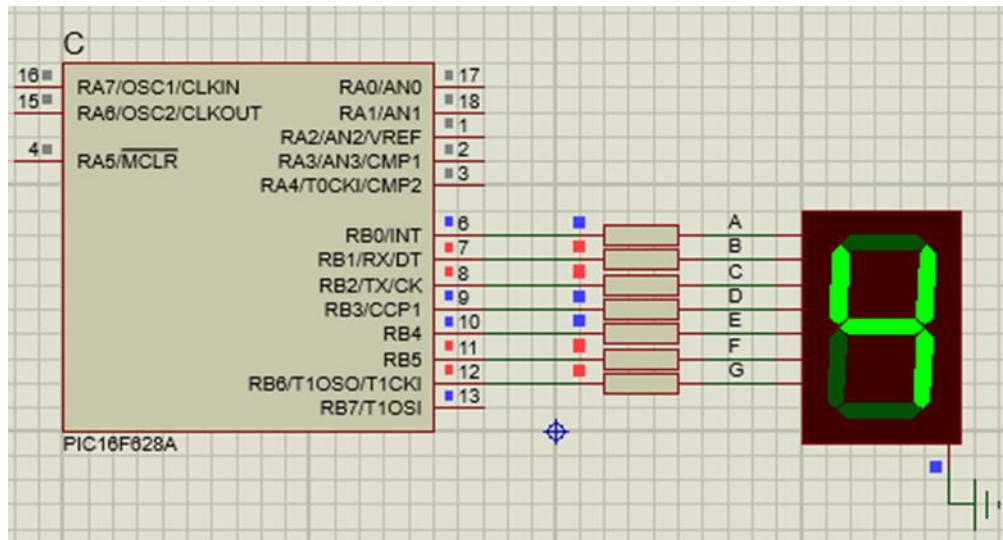
    while(1) //Loop infinito
    {
        __delay_ms(4000); // Espera 4 segundos
        // Sinal Verde do apagado

        do // faça...enquanto
        {
            {
                for (contador = 0; contador < 10; contador++) // Efetua a troca dos números
que aparecem no display
                {
                    PORTB = (mapa_segmento[contador]); //Seta no display os números de
0 a 9

                    __delay_ms(400); // Espera menos de meio segundo
                }
            }
        } while (contador < 10); // enquanto contador for menor que 10
    }
    return 0;
}

```

- **Simular o circuito eletrônico a seguir no Proteus, utilizando o código compilado no projeto 2.**
- **O esquema a seguir, representa a contagem de tempo (0-9) em um display de 7 segmentos (cátodo comum).**



➤ Criar um projeto 3 para compilar o arquivo fonte, no software Mplab:

```
#include<xc.h>
```

```
#pragma config FOSC = INTOSCIO      // Oscillator Selection bits (HS oscillator:
High-speed crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN)
#pragma config WDTE = OFF           // Watchdog Timer Enable bit (WDT disabled)
#pragma config PWRTE = OFF          // Power-up Timer Enable bit (PWRT disabled)
#pragma config MCLRE = OFF           // RA5/MCLR/VPP Pin Function Select bit
// (RA5/MCLR/VPP pin function is MCLR)
#pragma config BOREN = OFF           // Brown-out Detect Enable bit (BOD disabled)
#pragma config LVP = OFF             // Low-Voltage Programming Enable bit //(RB4/PGM
pin has digital I/O function, HV on MCLR must be used for //programming)
#pragma config CPD = OFF             // Data EE Memory Code Protection bit (Data
//memory code protection off)
#pragma config CP = OFF
```

```
#define _XTAL_FREQ 4000000
```

```
//Define a utilização do clock interno de 4 Mhz
```

```
#define _XTAL_FREQ 4000000
```

```
unsigned char const mapa_segmento[10] =
{0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F};
//Variaveis que representam os números.
```

```
int main()
```

```
{
```

```
    int contador; // Variavel usada para incremento e controle dos numeros que
aparecem no display (de 0 a 9)
```

```
    TRISB = 0x00; //Define todas as portas B como saída
```

```
    PORTB = 0x00;
```

```
    PORTA=0x00;
```

```

TRISA=0x00;
PORTB = (mapa_segmento[0]); // Força o display a apresentar o número 0 (Início
do programa)

// SEMAFORO
// TRISA3 = 0; //A0 Como pino de saída VERMELHO
//TRISA4 = 0; //A4 Como pino de saída AMARELO
//TRISA2 = 0; //A4 Como pino de saída VERDE

while(1) //Loop infinito
{
    __delay_ms(4000); // Espera 4 segundos
    PORTB = (mapa_segmento[0]); // Força o display a apresentar o número 0 (Fim
do ciclo e início de outro)
    RA7 = 0;      // Sinal Vermelho apagado
    RA6 = 0;      // Sinal Amarelo apagado
    RA2 = 1;      // Sinal Verde do acesso

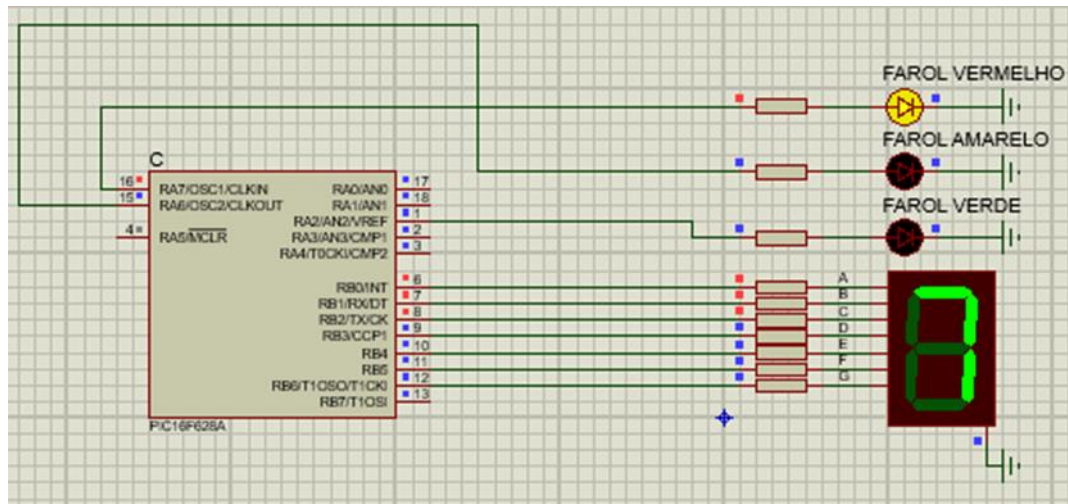
    __delay_ms(2000); // Espera 2 segundos
    RA7 = 0;      // Sinal Vermelho apagado
    RA6 = 1;      // Sinal Amarelo acesso
    RA2 = 0;      // Sinal Verde do apagado

    __delay_ms(4000); // Espera 4 segundos
    RA7 = 1;      // Sinal Vermelho acesso
    RA6 = 0;      // Sinal Amarelo apagado
    RA2 = 0;      // Sinal Verde do apagado

do // faça...enquanto
{
    if(RA7==1)
    {
        for (contador = 0; contador < 10; contador++) // Efetua a troca dos números
que aparecem no display
        {
            PORTB = (mapa_segmento[contador]); //Seta no display os números
de 0 a 9
            __delay_ms(400); // Espera menos de meio segundo
        }
    }
} while (contador < 10); // enquanto contador for menor que 10
}
return 0;
}

```

- **Simular o circuito eletrônico a seguir no Proteus, utilizando o código compilado no projeto 3.**
- **O esquema a seguir, representa um semáforo de contagem de tempo (0-9) em um display de 7 segmentos (cátodo comum), quando o sinal está no vermelho.**



ATIVIDADE AVALIATIVA:

- Desenvolver um código em C, utilizando o software Mplab, para:
 - Inserir um botão, para acionar o semáforo;
 - O semáforo será controlado pelo usuário;
 - Após ser acionado, o sinal passará para a condição fechado (led vermelho) em seguida se iniciará contagem de tempo regressivo de 9 até 0;
 - Em seguida o sinal será aberto.