Faculdade de Computação e Engenharia Elétrica
Microcontroladores e Microprocessadores – Prof. Dr. Elton Alves
Experimento 9 – Projeto em Linguagem C no PIC16F628A

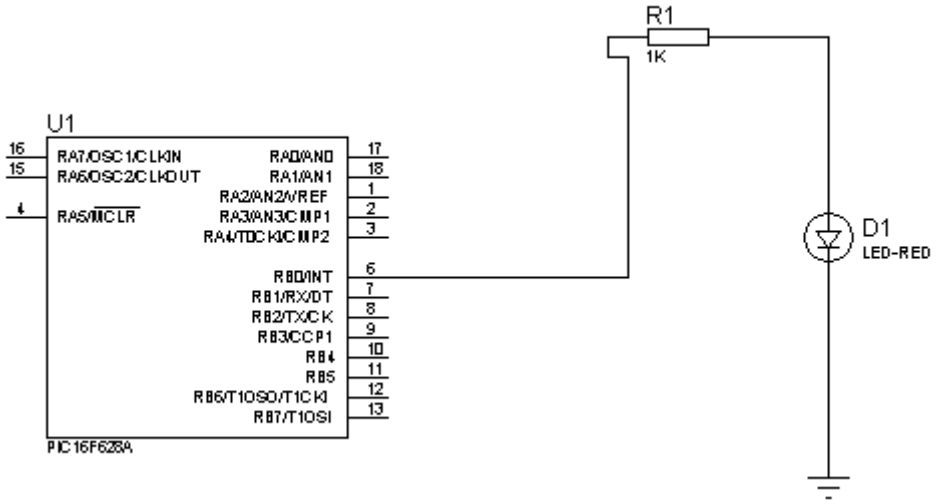
- **Objetivos:**

- Programar o PIC16F628A em linguagem C para piscar um LED.

➢ **Criar um projeto 1 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 LED RB0

void main (){
   PORTB=0x00;
   TRISB=0x00;
   while(1)
   {
      LED=1;
      __delay_ms(1000);
      LED=0;
      __delay_ms(1000);
   }
}
```

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

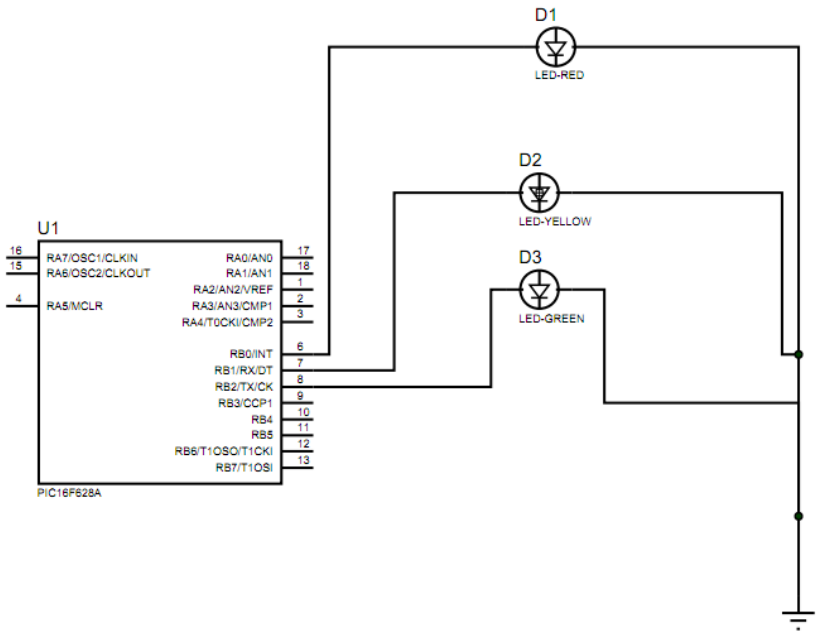


- **Criar um projeto 2 para compilar o arquivo fonte:**

```
*
 * File:   Experimento91.c
 * Author: Elton Alves
 *
 * Created on 22 de Julho de 2021, 11:08
 */

#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


void main (){
   PORTB=0x00;
   TRISB=0x00;
   while(1)
   {
      LED1=1;
      LED2=0;
      LED3=0;
       __delay_ms(200);
      LED1=0;
      LED2=1;
      LED3=0;
      __delay_ms(200);
      LED1=0;
      LED2=0;
      LED3=1;
      __delay_ms(200);
   }
}
```

- **Simular o circuito eletrônico a seguir no Proteus, utilizando o código compilado no projeto 2.**



ATIVIDADE AVALIATIVA:

- Desenvolver um programa em linguagem C para controlar um cruzamento e ruas com 3 tempos, com o circuito correspondente no simulador Proteus e hardware na protoboard.

- Tempo 1: Av. VP7 em ambos sentidos (os dois são iguais)
- Tempo 2: Av. VP8 sentido Norte-Sul
- Tempo 3: Av. VP8 sentido Sul-Norte

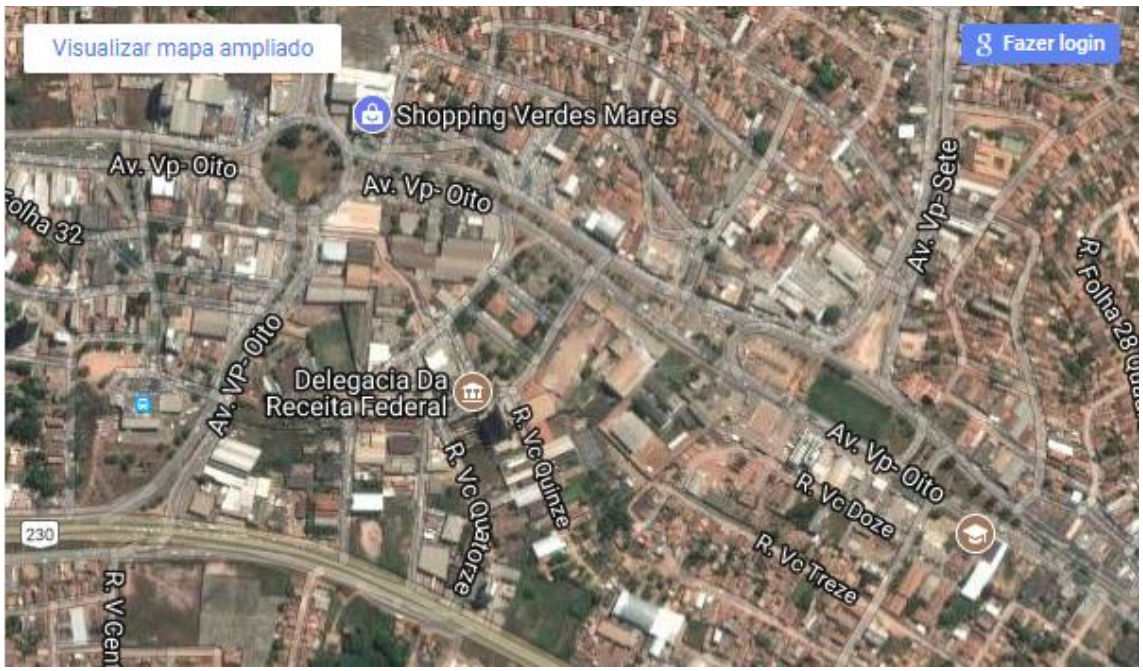
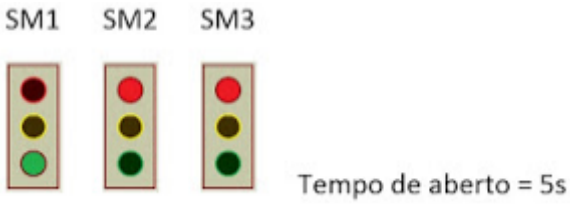
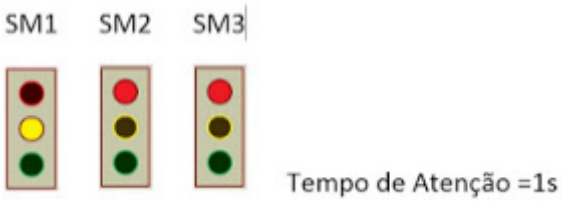


Diagrama de estados:

- Semáforo 1 VP7 aberto e demais fechados.



- Semáforo 1 VP7 em atenção e demais fechados.



- Semáforo 2 VP8 sentido Norte-Sul aberto e demais fechados.

- Semáforo 2 VP8 sentido Norte-Sul atenção e demais fechados.

- Semáforo 3 VP8 sentido Sul-Norte aberto e demais fechados.

- Semáforo 3 VP8 sentido Sul-Norte em atenção e demais fechados.