

Nome: Beatriz Salomão Monteiro

Nesse relatório, irei demonstrar o desenvolvimento de um programa para controlar o tempo em que uma bomba de piscina fica ligada, que foi desenvolvido com projeto final das disciplinas de Programação Embarcada de Laboratório de Programação Embarcada.

Foi usado o ambiente de programação MPLABX IDE disponibilizado pela Microchip e o compilador foi o XC8. Foi usada a linguagem de programação C.

Suprimentos:

- MPLABX IDE v5. 40;
- PicSimLab
- Compilador XC8

### Etapa 1: Equipamentos e softwares

O PicSimLab é um simulador de um kit desenvolvimento com vários periféricos, como display de sete segmentos, display de LCD, etc.

No simulador, foi usado o display de sete segmentos para demonstração da contagem de tempo; no display de LCD está um aviso de que a bomba está ligada; e o ventilador simula a bomba ligada.

### Etapa 2: Desenvolvimento do código

Para a realização do projeto, foram disponibilizadas bibliotecas de controles dos periféricos do hardware. Cada biblioteca possuem suas funções de inicialização e configuração dos componentes.

O código foi desenvolvido com a biblioteca de PWM, que simplesmente liga o ventilador, quando a placa está em funcionamento. Também foi utilizado a biblioteca do display de sete segmentos que é usada para contar o tempo e mostrar em sua interface, assim como a biblioteca do display de LCD foi usada para demonstrar a placa em funcionamento.

### Etapa 3: Funcionamento do código no PicSimLab

Com o simulador, foi testado o programa.

Código:

```
#include "ssd.h"
#include "pwm.h"
#include "config.h"
#include "lcd.h"
```

```

void main(void) {
    unsigned int cont, i, j;
    float tempo;
    unsigned int contseg, contmin;

    TRISD = 0x00;

    timerInit();
    ssdInit();
    pwmInit();
    pwmSet(50);
    lcdInit();

    lcdData('L');
    lcdData('i');
    lcdData('g');
    lcdData('a');
    lcdData('d');
    lcdData('o');

    for (;;) {

        cont++;
        contseg++;
        contmin++;
        if (((contseg / 1000) % 10) == 6) {
            contseg = 0;
            contmin += 4000;
        }
        pwmSet(50);

        ssdDigit(((contseg / 100) % 10), 3); //1s
        ssdDigit(((contseg / 1000) % 10), 2); //10s
        ssdDigit(((contmin / 10000) % 10), 1);
        ssdDigit(((contmin / 100000) % 10), 0); //1000s
        ssdUpdate();

    }

}

```

