UNIVERSIDADE FEDERAL DE ITAJUBÁ

ORLANDO JOSE DE SOUZA BECO – 2016002046

TRABALHO FINAL DE PROGRAMAÇÃO EMBARCADA E LABORATÓRIO DE PROGRAMAÇÃO EMBARCADA

PROF. DR. OTÁVIO DE SOUZA MARTINS GOMES

1° SEMESTRE DE 2021

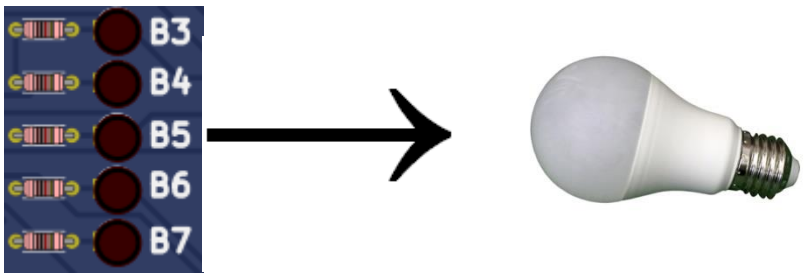
1. Descrição da Proposta

O trabalho consiste no desenvolvimento de um Projeto com o PICSimLab utilizando a maior parte dos conceitos apresentados na disciplina. A partir disso, tomou-se como objetivo desenvolver um controlador para um quarto com as seguintes funcionalidades:

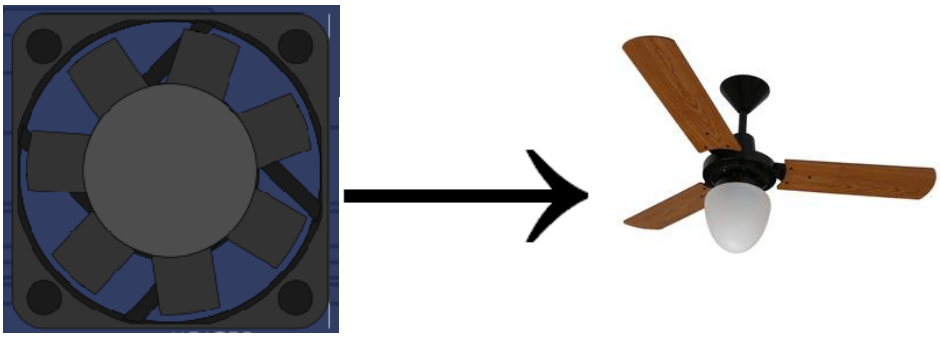
- Ligar/Desligar a lâmpada do quarto;
- Ligar/Desligar o ventilador;
- Controlar a potência do ventilador;
- Trancar/Destrancar a porta via fechadura eletrônica;
- Ligar/Desligar o aquecedor.

Para a realização deste projeto, foram utilizadas os seguintes componentes do PICSimLab:

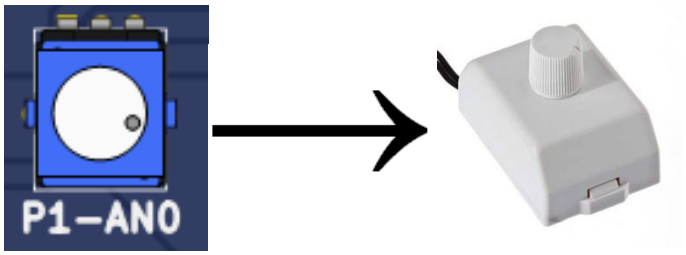
- Leds para simular a lâmpada do quarto;



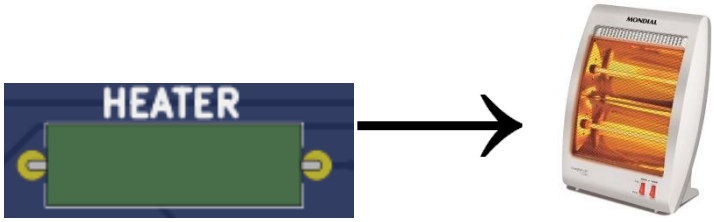
- Ventoinha (cooler) para simular o ventilador;



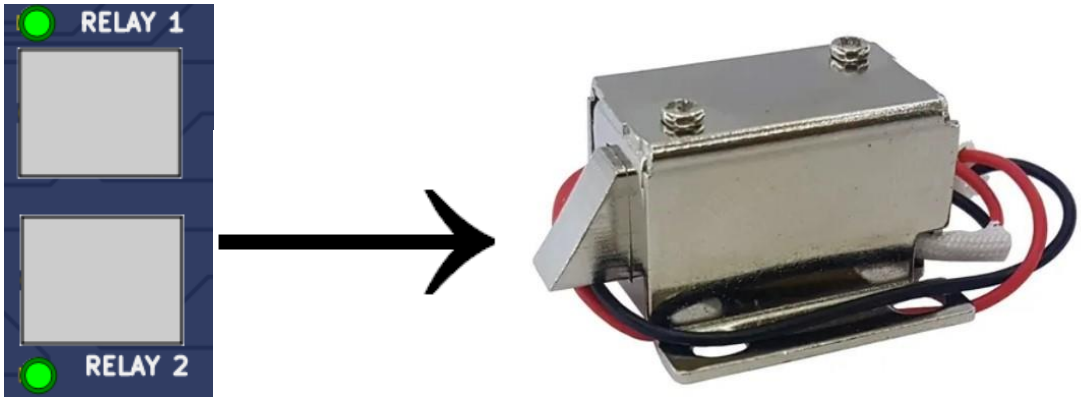
- Potenciômetro para o controle de velocidade da ventoinha do cooler;



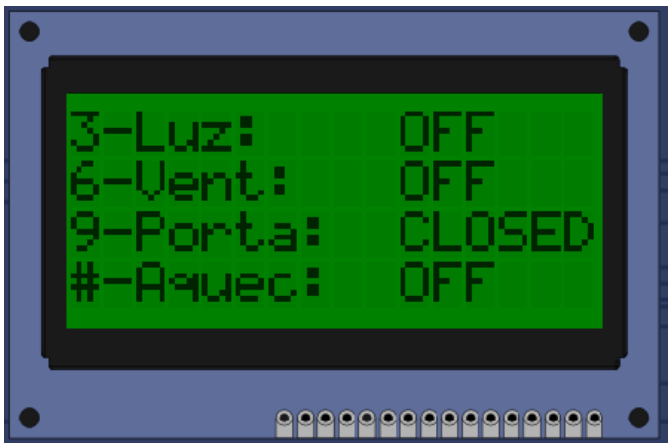
- Heater para simular o aquecedor;



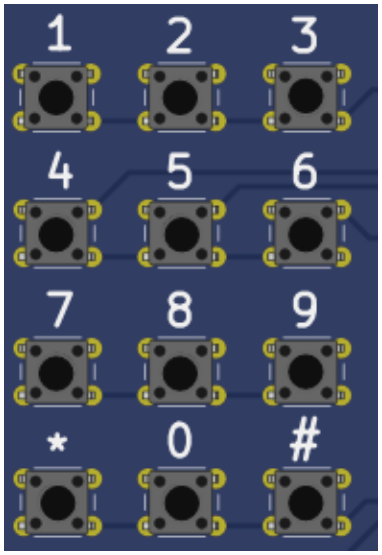
- Relés para mostrar o estado da fechadura da porta;



- LCD hd44780 16x4 para visualização dos estados dos componentes;



- Teclado matricial para selecionar opções desejadas mostradas no LCD.



2. Microcontrolador

Para o desenvolvimento deste projeto, foi utilizado como base a placa PICGenios com o microcontrolador PIC18f4520, com 256 bytes de memória EEPROM e 32 kbits de memória Flash. Este é um microcontrolador de 8 bits com núcleo de 14 bits, fabricado pela microchip. Abaixo, podemos ver a imagem do microcontrolador e da placa real.

Figura 1 – Microcontrolador PIC18F4520

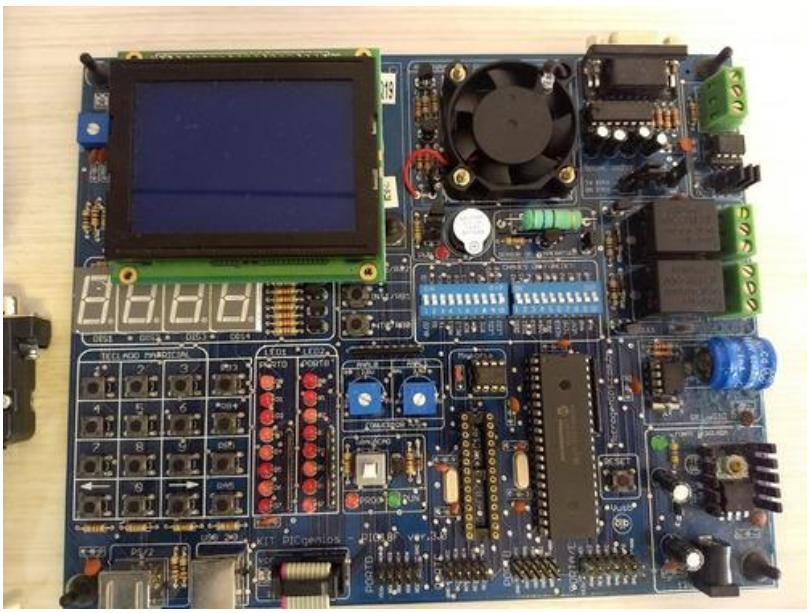


Figura 2 – Placa PICGenios

3. Desenvolvimento

Para o desenvolvimento deste projeto, foram abordados diversos conceitos além dos dispositivos já mencionados anteriormente, tal como PWM para o controle da velocidade das ventoinhas do cooler.

Para o desenvolvimento deste projeto, optou-se pela utilização de funções secundárias que estão no arquivo main.c e sua chamada na função main, quando necessário. Abaixo temos trechos do código com a sua descrição:

- No início, foram feitos os includes das bibliotecas necessárias para o desenvolvimento das funções que seriam utilizadas neste programa:

```
#include <pic18f4520.h>
#include "config.h"
#include "bits.h"
#include "lcd.h"
#include "keypad.h"
#include "delay.h"
#include "pwm.h"
#include "adc.h"
```

- Em seguida, foram declaradas as variáveis que seriam utilizadas em mais de uma função neste programa, seja ela a main ou alguma outra:

```
int estadoPorta, estadoVent, estadoLuz, estadoAquec;
unsigned int tecla;
unsigned char vel;
```

- Depois, foram inicializadas as funções que foram desenvolvidas e que são utilizadas no programa:

```
void escolheOpcoes(void);    //Escolha de opções através do teclado
void luz(void);              //Liga/Desliga Luz
void vent(void);             //Liga/Desliga o Ventilador
void porta(void);            //Abre/Fecha a fechadura da porta
void aquec(void);            //Liga/Desliga o Aquecedor
void lcdPrint(void);         //Mostra as informações fixas no LCD
void lcdPrintInicio(void);   //Mostra os estados iniciais dos componentes
void contrVent(void);        //Realiza o controle da ventoinha
```

- A seguir, segue cada uma das funções inicializadas acima:

```
void escolheOpcoes(void) {
    kpDebounce();
    if (kpRead() != tecla) {
        tecla = kpRead();
        for (int aux = 0; aux < 12; aux++) {
            if (bitTst(tecla, aux)) {
                if (aux == 11) {        //tecla 3
                    luz();
                }
                if (aux == 10) {        //tecla 6
                    vent();
                }
                if (aux == 9) {         //tecla 9
                    porta();
                }
                if (aux == 8) {         //tecla #
                    aquec();
                }
            }
        }
    }
}
```

```
void luz(void) {
    if (estadoLuz == 1) {        // se ligado no acionamento
        estadoLuz = 0;
        PORTB = 0x00;            // desliga
        lcdCommand(L_L1 + 10);
        lcd_str("OFF   ");
    } else {                     // se não
        estadoLuz = 1;
        PORTB = 0xF8;            // liga
        lcdCommand(L_L1 + 10);
        lcd_str("ON    ");
        ;
    }
}

void vent(void) {
    if (estadoVent == 1) {       // se ligado no acionamento
        estadoVent = 0;          // liga
        lcdCommand(L_L2 + 10);
        lcd_str("OFF   ");
    } else {                     // se não
        estadoVent = 1;          // liga
        lcdCommand(L_L2 + 10);
        lcd_str("ON  ");
    }
}
void contrVent(void) {
    if (estadoVent) {                    // se ligado
        vel = (adcRead(0)*20) / 204;     // le o potenciômetro 1
        pwmSet1(vel);                    // controla a velocidade
    } else {                             // se não
        pwmSet1(0);                      // desliga
    }
}

void porta(void) {
    if (estadoPorta == 1) {      // se fechada no acionamento
        estadoPorta = 0;
        PORTCbits.RC0 = 0;       // manda abrir
        PORTEbits.RE0 = 0;
        lcdCommand(L_L3 + 10);
        lcd_str("OPEN   ");
    } else {                     // se aberta no acionamento
        estadoPorta = 1;
        PORTCbits.RC0 = 1;       // manda fechar
        PORTEbits.RE0 = 1;
        lcdCommand(L_L3 + 10);
        lcd_str("CLOSED ");
    }
}
```

```
void aquec(void) {
    if (estadoAquec == 1) {          // se ligado no acionamento
        estadoAquec = 0;
        PORTCbits.RC5 = 0;           // desliga
        lcdCommand(L_L4 + 10);
        lcd_str("OFF ");
    } else {                         // se não
        estadoAquec = 1;
        PORTCbits.RC5 = 1;           // liga
        lcdCommand(L_L4 + 10);
        lcd_str("ON  ");
    }
}

void lcdPrint(void) {        // Informações que não mudam
    lcdCommand(L_CLR);
    lcdCommand(L_L1);
    lcd_str("3-Luz:");
    lcdCommand(L_L2);
    lcd_str("6-Vent:");
    lcdCommand(L_L3);
    lcd_str("9-Porta:");
    lcdCommand(L_L4);
    lcd_str("#-Aquec:");
}

void lcdPrintInicio(void) {          // Informações de inicialização
    lcdCommand(L_CLR);
    lcdCommand(0x0C);
    lcdCommand(0x40);
    lcdCommand(L_L1);
    lcd_str("Inicializando...");
    atraso_s(2);
    lcdCommand(L_L2);
    lcd_str("   Smart Room   ");
    atraso_s(2);
    lcdCommand(L_L4);
    lcd_str("by: Orlando Beco");
    atraso_s(3);
    lcdPrint();
    lcdCommand(L_L1 + 10);           // Estados iniciais dos componentes
    lcd_str("OFF");
    lcdCommand(L_L2 + 10);
    lcd_str("OFF");
    lcdCommand(L_L3 + 10);
    lcd_str("OPEN");
    lcdCommand(L_L4 + 10);
    lcd_str("OFF");
}
```

Além das funções acima, há também o programa principal, a main deste projeto, que faz a chamada para a função escolheOpções, que caso uma operação seja escolhida, ela é direcionada para a função que realiza o processo necessário.

```
void main(void) {
    ADCON1 = 0x06;          //configurações iniciais
    TRISA = 0xC3;
    TRISB = 0x07;
    TRISC = 0x00;
    TRISD = 0x00;
    TRISE = 0x00;
    tecla = 16;             //inicialização de variáveis
    estadoPorta = 0;
    estadoVent = 0;
    estadoLuz = 0;
    estadoAquec = 0;

    adcInit();              //inicialização de funções
    pwmInit();
    lcdInit();
    kpInit();
    lcdPrintInicio();

    for (;;) {              //loop infinito
        escolheOpcoes();    //aguarda leitura de opção
        contrVent();        //controla velocidade da ventoinha
    }
}
```

A função contrVent está neste loop no programa principal porque ela não deve depender de nada além da variável que define o estado do Cooler, assim, caso a variável do Cooler indique que o Cooler está ligada, está função faz o controle da velocidade através do potenciômetro P1.

Link do github para os códigos desenvolvidos e as bibliotecas utilizadas neste projeto: https://github.com/orlandobeco/Projeto_ProgEmbarcada

4. Simulação

A simulação foi realizada no PICSimLab versão 0.8.8 e tem a seguinte interface gráfica:

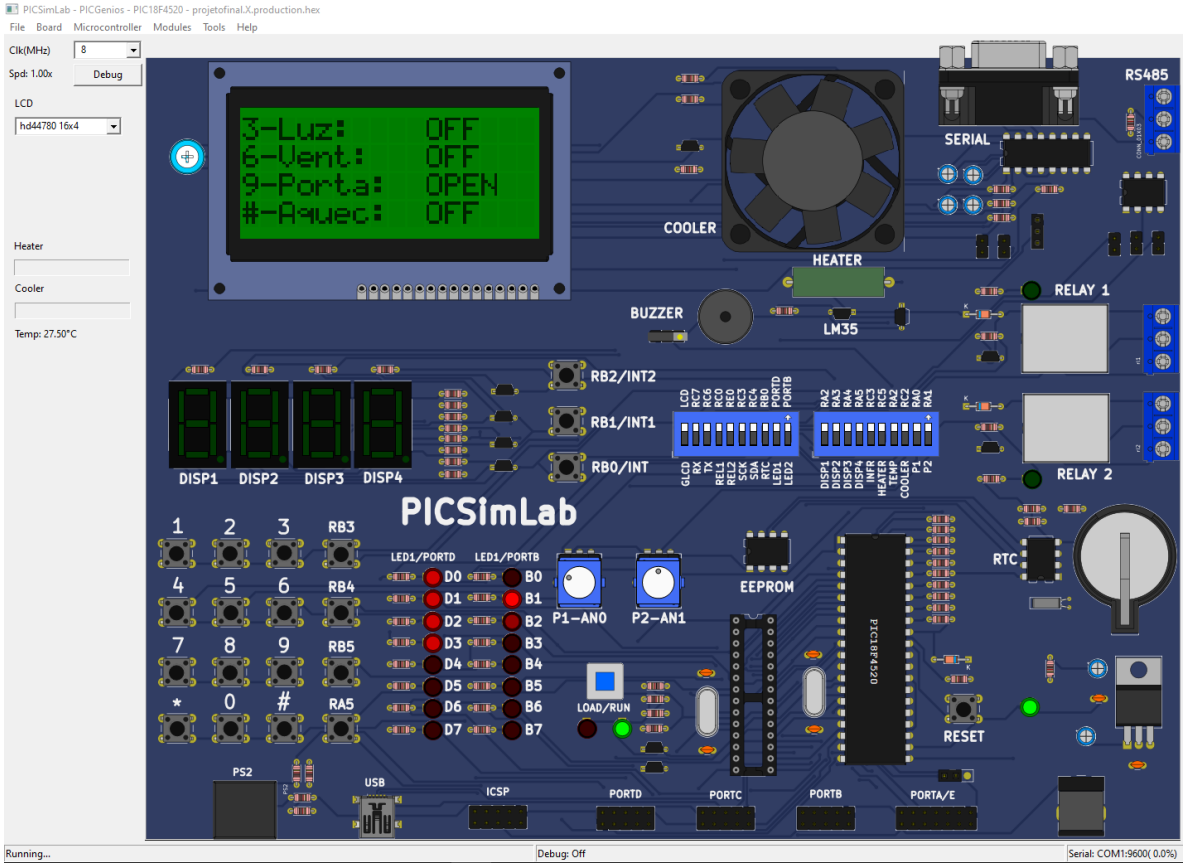


Figura 3 – Simulação do Projeto

Nesta simulação, primeiro são apresentadas informações iniciais no display LCD, como mostra abaixo:

Figura 4 – Inicialização da simulação

Em seguida, aparece a tela da Figura 4 e assim pode-se realizar a simulação do projeto, controlando os estados dos componentes: Luz, Ventilador, Porta e Aquecedor pelas teclas 3, 6, 9 e #, respectivamente. Como falado anteriormente, cada um desses componentes tem um representante na placa, e desta forma, especificamos o estado de cada um deles da seguinte maneira:

- Luz → ON – Leds de B3 a B7 ligados.
  OFF – Leds de B3 a B7 desligados.
- Ventilador → ON – Cooler ativado para ser controlado pelo potenciômetro.
  OFF – Cooler desativado.
- Porta → OPEN – Relés desligados, representa que a fechadura não está acionada.
  CLOSED – Relés ligados, representa que a fechadura está acionada.
- Aquecedor → ON – Heater está ligado.
  OFF – Heater está desligado.

Para melhor visualização, no link abaixo pode ser observado o funcionamento do projeto pelo PICSimLab: https://youtu.be/_EaX3yIjUf4

5. Dificuldades encontradas

A principal dificuldade encontrada neste projeto foi para realizar a multiplexação entre os LEDs e o display de 7 segmentos, desta forma, optou-se por utilizar outras funcionalidades que não estavam previstas no inicio deste projeto, tal como o uso do potenciômetro e do Heater e deixar de lado o display de 7 segmentos, que serviria para mostrar o horário atual, da forma HHMM.

Outra dificuldade encontrada foi o controle via PWM do Heater. Não foi possível chegar a este controle via PWM, que seria regulado pelo potenciômetro P2, então optou-se por utilizar apenas em 2 modos, ON/OFF, que não deixa de representar alguns aquecedores do mercado, que não possuem o controle de potência de aquecimento

6. Conclusão

Após o desenvolvimento, conclui-se que o projeto atendeu as expectativas de trabalhar com conhecimentos adquiridos na matéria teórica e relacionar com alguma aplicação real. O uso do PICSimLab para a simulação atendeu as expectativas e foi possível simular todo o projeto sem problemas. O controle do quarto por microcontrolador é uma proposta simples, mas que poderia atender diversas pessoas, dado a facilidade que proporcionaria para as pessoas. Este é um projeto que pode e deve ser evoluído, com acréscimo de novas funcionalidades, além de poder ser personalizado facilmente caso a caso.