

Universidade Federal de Itajubá - Unifei

Guilherme de Freitas Marques

PROJETO NO SIMULADOR PICSImLab - SECADOR DE CABELO

Itajubá - MG
2021

SUMÁRIO

INTRODUÇÃO	3
DESENVOLVIMENTO E FUNCIONAMENTO	4
DIFICULDADES ENCONTRADAS	7
CONCLUSÃO	8
REFERÊNCIAS	9

1. INTRODUÇÃO

Um sistema embarcado é um computador construído para o único propósito da sua aplicação, ao invés de prover um sistema computacional generalizado. Essa definição independe da sua construção, pode abranger desde lógicas combinacionais e processadores de 8-bits, até processadores de 64-bits multi-cores e ASICs. Sistemas embarcados são utilizados na maioria das áreas que envolvem eletrônica, como na produção de eletrodomésticos, aparelhos de comunicação, equipamentos médicos e industriais, brinquedos etc.

Microcontroladores são dispositivos compostos por um processador dedicado com finalidade de executar uma aplicação específica, sendo utilizados em projetos de sistemas eletrônicos, tendo como principal função a execução de uma aplicação, ou conjunto de aplicações, relacionadas como um único sistema.

Neste projeto será desenvolvido um sistema embarcado que simulará as funções de um secador de cabelo. O microcontrolador simulado será o PIC18F4520 (Imagem 1) e a plataforma de simulação utilizada será a do PICSIMLAB (Imagem 2).

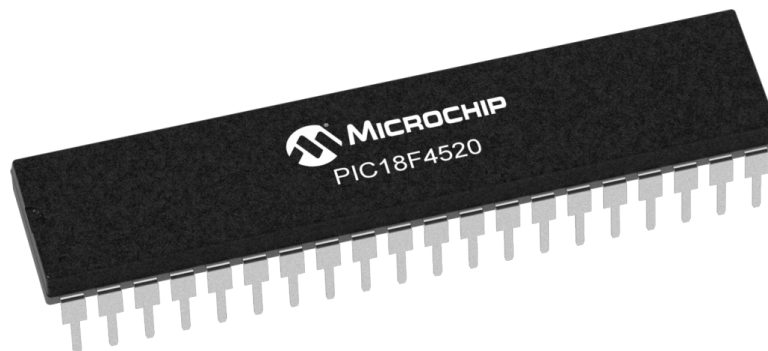


Imagem 1 - PIC18F4520

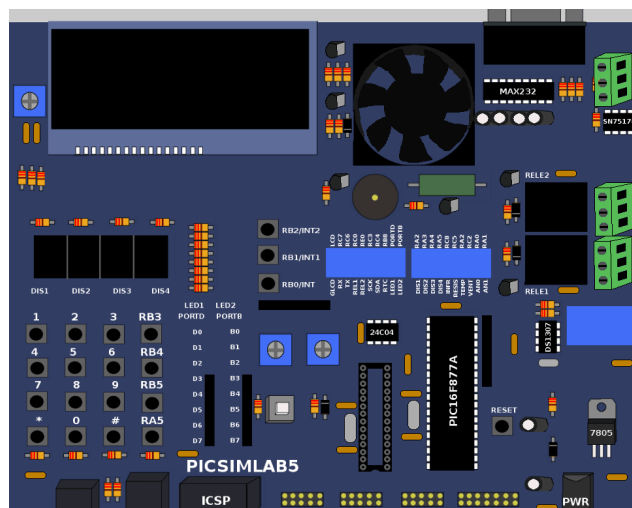


imagem 2 - PICSIMLAB

2. DESENVOLVIMENTO E FUNCIONAMENTO

2.1. Os Componentes

O projeto foi pensado e desenvolvido tendo em foco a simulação de um secador de cabelo similar ao real. Primeiramente foi pensado o que cada componente do PICSimLab iria representar, levando em consideração como seria implementado em código suas diferentes funções. Acabou-se por decidir que o cooler seria o motor (Imagem 3), o heater o aquecedor (Imagem 4), o buzzer (Imagem 5) simularia o barulho do motor, o teclado o painel de acesso do usuário (Imagem 6), o display LCD a tela de informações (Imagem 7), os relays os diferentes modos de saída de vento (Imagem 8) e os LEDs o indicador visual mais simples da ativação dos componentes (Imagem 9).

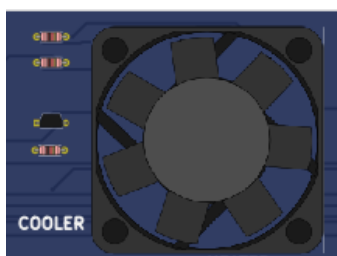


Imagem 3 - Cooler



Imagem 4 - Heater

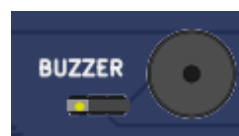


Imagem 5 - Buzzer



Imagem 6 - Teclado

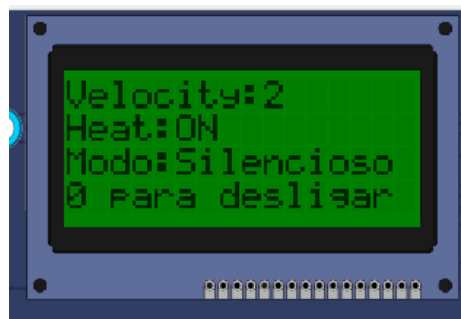


Imagem 7 - display LCD

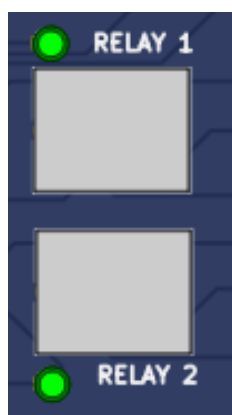


Imagem 8 - Relays

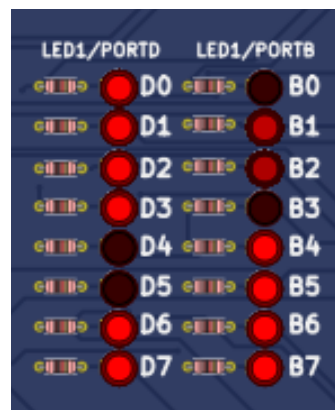


Imagem 9 - LEDs

2.2. O Código

No desenvolvimento do código, foi pensado que um secador estaria em constante funcionamento e mudanças de operação seriam feitas em tempo real, logo o loop que contém o funcionamento do secador ligado (Imagem 10) não tem nenhuma interrupção e consiste em ações baseadas na leitura de teclas do teclado.

As teclas 1, 2 e 3 configuram a velocidade do motor e a representação do estado deste nos LEDs, as teclas 4 e 5 determinam se o aquecedor está ligado ou não e a representação do estado deste nos LEDs, as teclas 7, 8 e 9 controlam o modo de saída de vento e a tecla 0 encerra o loop e muda o estado do secador para desligado.

```
while (1) {
    kpDebounce();
    if (kpRead() != tecla) {
        tecla = kpRead();
        if (bitTst(tecla, 3)) { //Tecla 1 - Velocidade de rotação lenta
            lcdPosition(0, 9);
            lcdStr("1");
            pwmSet1(32);
            PORTD = 0b10000000;
        }
        if (bitTst(tecla, 7)) { //Tecla 2 - Velocidade de rotação média
            lcdPosition(0, 9);
            lcdStr("2");
            pwmSet1(67);
            PORTD = 0b11000000;
        }
        if (bitTst(tecla, 11)) { //Tecla 3 - Velocidade de rotação alta
            lcdPosition(0, 9);
            lcdStr("3");
            pwmSet1(99);
            PORTD = 0b11100000;
        }
        if (bitTst(tecla, 2)) { //Tecla 4 - Aquecedor desliga
            lcdPosition(1, 5);
            lcdStr("OFF");
            PORTCbits.RC5 = 0;
            PORTB = 0x00;
        }
        if (bitTst(tecla, 6)) { //Tecla 5 - Aquecedor liga
```

Imagem 10 - Loop do secador ligado

O secador inicia desligado, e para ligar o usuário deve pressionar a tecla 1 (Imagem 11). Quando pressionada, é configurado as informações que serão mostradas no LCD e o funcionamento padrão (Imagem 12).

```

for (;;) {
    //Desliga o secador
    PORTCbits.RC5 = 0; //Desliga o Heater
    pwmSet2(0);        //Desliga o Buzze
    pwmSet1(0);        //Desliga o Cooler
    PORTD = 0;         //Desliga os LEDs do PORTD
    PORTB = 0;         //Desliga os LEDs do PORTB
    PORTCbits.RC0 = 0; //Desliga o relay 1
    PORTEbits.RE0 = 0; //Desliga o relay 2
    lcdPosition(0, 0);
    lcdStr("Secador OFF");
    lcdPosition(1, 0);
    lcdStr("Pressione 1");
    lcdPosition(2, 0);
    lcdStr("          ");
    lcdPosition(3, 0);
    lcdStr("          ");
    //Espera input do usuário para ligar
    while (1) {
        kpDebounce();
        tecla = kpRead();
        if (bitTst(tecla, 3)) { //Enquanto não pressionar 1, o loop continua
            break;
        }
    }
}

```

Imagem 11 - Secador desligado esperando input

```

//Configurações para ligar o secador
PORTD = 1;
lcdPosition(0, 0);
lcdStr("Velocity:1      ");
lcdPosition(1, 0);
lcdStr("Heat:OFF        ");
lcdPosition(2, 0);
lcdStr("Modo:Padrao  ");
lcdPosition(3, 0);
lcdStr("0 para desligar");
pwmSet1(32);
pwmSet2(50);

```

Imagem 12 - Configuração base padrão

2.3 Bibliotecas utilizadas e criadas

Todas as bibliotecas utilizadas foram ou disponibilizadas anteriormente ou criadas ao decorrer do curso, ou seja, nenhuma nova biblioteca foi criada. as bibliotecas incluídas foram: config.h, bits.h, lcd.h, io.h, keypad.h, pic18f4520.h e pwm.h.

3. DIFICULDADES ENCONTRADAS

Foram encontradas algumas dificuldades ao decorrer do desenvolvimento do projeto. Uma delas foi descobrir como acionar o heater, parte essencial do produto sendo simulado, o que foi resolvido com a ajuda de um colega.

Outra dificuldade foi a leitura do teclado. Ao implementar um código feito anteriormente para leitura de teclas, este não funcionava com as bibliotecas atualizadas que estavam incluídas. Logo, foi necessário analisar as bibliotecas para descobrir qual estava interferindo no funcionamento do teclado, e após descobrir que era a biblioteca responsável pelo controle do LCD, foi restaurada esta para uma configuração anterior e criada novamente a função “lcdStr” que escreve um fluxo de caracteres no LCD.

Fora estas dificuldades, o projeto fluiu normalmente e ficou pronto em alguns dias.

4. CONCLUSÃO

É entendido que o projeto cumpriu os requisitos impostos em seu desenvolvimento: Foram utilizados a maioria dos componentes do PICSIMLab para desenvolvimento da simulação de um sistema embarcado utilizando todo o conhecimento prévio da matéria.

5. REFERÊNCIAS

- <https://www.embarcados.com.br/sistemas-embarcados-e-microcontroladores/>
- <https://www.embarcados.com.br/sistema-embarcado/>