Relatório do Projeto Final – Ventilador de Teto

1. Introdução

Com o objetivo de criar um projeto final para a disciplina ECOP04 (Programação Embarcada) que é ministrada na Universidade Federal de Itajubá (UNIFEI), desenvolvi um programa que utiliza o emulador PicSimLab, a placa PICGenios, o microcontrolador PIC18F4520 e o software MPLAB X IDE. Este programa simula um ventilador de teto que executa os comandos recebidos do teclado matricial presente na placa. Ele também possui timers instalados que serão mostrados no SSD (Display de 7 Segmentos), um Display de LCD e uma ventoinha, que será ligada através de uma saída de PWM e simulará o ventilador.

2. Estrutura do código

O código do programa possui início a partir das inclusões das bibliotecas necessárias para o seu funcionamento. Dentre elas estão:

- pic18f4520.h
- config.h
- pwm1.h
- keypad.h
- ssd.h
- lcd.h
- timer.h
- bits.h

Após isso, a função main foi criada e foram feitas as criações das variáveis que viriam a ser utilizadas no código e as inicializações das funções dos periféricos.

```
unsigned char tecla, temp, i, Tela = 1, VelocidadeVentoinha = 1;
unsigned char EstadoVentoinha = 0, EstadoTimer = 0, EstadoLuminaria = 0;
unsigned long timerSSD = 0;

lcdInit();
kpInit();
pwmInit();
ssdInit();
timerInit();
```

Figura 2.1 – Criação das variaveis e inicialização das funções

Após a criar e iniciar as variáveis e as funções que estão presentes no código, foi criado o loop principal, onde o programa executará todo o procedimento até que a placa desligue. Afim de que o código ficasse bem resumido e compacto, utilizei estruturas de bitTst() que checariam qual bit está ligado e consequentemente qual tecla foi pressionada e o procedimento necessário perante tal fato.

```
if (bitTst(tecla, 3)) {
    if ((EstadoVentoinha % 2) == 0) {
        pwmSet1(32);
        VelocidadeVentoinha = 1;
    } else if ((EstadoVentoinha % 2) != 0) {
        pwmSet1(0);
        VelocidadeVentoinha = 1;
    }
    EstadoVentoinha++;
}
```

Figura 2.2 – Exemplo da estrutura de bitTst() presente no código

Por fim, foi utilizado um timer para garantir que o loop principal sempre executasse em 10ms, garantindo a placa estabilidade e o bom funcionamento do teclado e do display de 7 segmentos.

3. Funcionamento dos Periféricos

3.1. Teclado Matricial

Neste projeto, o teclado é o segundo mais importante periférico do projeto, ficando apenas atras da ventoinha que simulará o ventilador em si. Isso se deve, pois o teclado e suas teclas são a única forma de comunicação entre a placa e o usuário, sendo seus comandos passados através de um sistema de leitura e debounce das teclas e chegando ao microprocessador. Dessa forma, a leitura dos botões e o bom funcionamento deste periférico são de extrema importância para o projeto.

Por questões de design, adotei as teclas 1 e 2 como formas de conversar diretamente com a ventoinha, fazendo com que a tecla 1 liga e desliga a ventoinha e a tecla 2 altera a velocidade da ventoinha, tendo 3 velocidades, baixa, média e alta. As teclas 0, 5 e 8 ativam os timers no display de 7 segmentos, sendo que a tecla 5 ativa o timer de 5 minutos, a tecla 8 ativa o

timer de 10 minutos e a tecla 0 ativa o timer de 30 minutos. Já as teclas 4, 7 e “*” mostram os comandos das teclas no display de LCD.

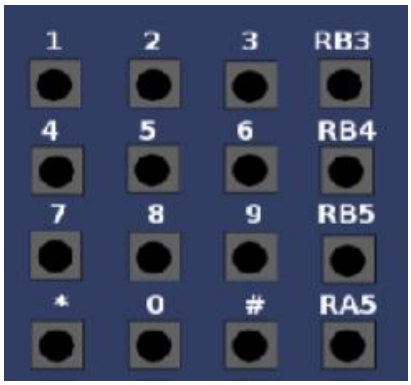

Figura 3.1.1 – Mapeamento do Teclado

3.2. Display de 7 Segmentos (SSD)

Para exibir o timer funcionando na placa, optei por utilizar os 4 displays de sete segmentos, fazendo com que os dois primeiros exibam os dígitos dos minutos e os dois últimos os dígitos dos segundos que faltam para terminar o timer. Foram programados 3 diferentes tempos que podem ser utilizados pelo timer; 5 minutos, 10 minutos e 30 minutos. Ao final do tempo programado no timer, a ventoinha desliga.

Figura 3.2.1 – SSD em funcionamento

3.3 Display de LCD

Como forma de exibir os comandos para o usuário, utilizei o display de LCD com 3 interfaces que podem ser alternadas pelas teclas presentes no teclado matricial. Na primeira interface ele verá os comandos da ventoinha, podendo o usuário ligar, desligar ou alterar a velocidade. Na segunda interface ele verá os comandos para ligar os timers de 5 e 10 minutos. E na terceira interface ele verá o comando para ligar o timer de 30 minutos.

Figura 3.3.1 – Exibição da 1ª interface no LCD

3.4 Ventoinha

Como ultimo periférico a ser utilizado na nossa placa, a ventoinha pode ser considerada como o periférico mais importante de todo o projeto, já que possui uma importantíssima missão, simular o ventilador em si. Seu acionamento é feito através de uma saída de PWM, que se parece com uma saída analógica, porém é uma saída digital que opera em pulsos, ligando e desligando em uma frequência predeterminada. Dessa forma, torna-se possível operá-la em níveis, simulando os níveis baixo, médio e alto de rotação do ventilador.

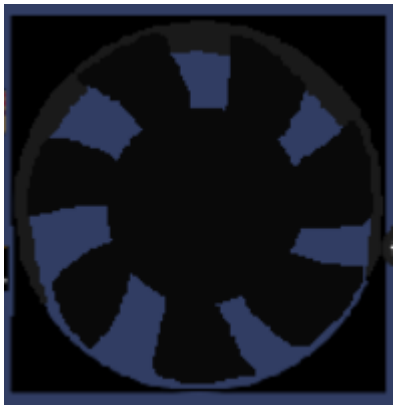

Figura 3.4.1 – Ventoinha presente no PicSimLab

4. Dificuldades na Elaboração do Projeto

A maior dificuldade encontrada durante a elaboração deste projeto foi encontrar uma forma de exibir para o usuário qual a função de cada tecla. Depois de pensar bastante sobre resolvi utilizar o Display de LCD, fazendo com que fique de fácil acesso ao usuário acessar cada uma das três interfaces presente no LCD por meio das teclas já mencionadas anteriormente no teclado. Todavia, caso o projeto fosse produzido na vida real, provavelmente não seria utilizado o display de LCD, e as funções de cada tecla seriam escritas ao lado de cada um dos botões, respectivamente. Dessa forma, não seria necessário elevar o custo do sistema e aumentaria ainda mais a acessibilidade da interface usuário-ventilador.

5. Conclusões

Foi algo gratificante poder produzir este projeto, visto que os conhecimentos que adquiri ao longo deste semestre em embarcados realmente foram postos a prova para sua realização. Pode-se dizer que o projeto como um todo foi bem sucedido, alcançando o requisitado pelos professores Rodrigo Almeida e Otávio Gomes da disciplina ECOP04 (Programação Embarcada). Fiquei satisfeito com o resultado final, visto que foi possível criar um sistema bastante coeso e completo onde os periféricos se comunicassem satisfatoriamente entre si.