

UNIVERSIDADE FEDERAL DE ITAJUBÁ

ELAINE APARECIDA INÁCIO

DECK INTELIGENTE

Este relatório visa demonstrar o processo de como foi feito o programa do controle de um deck, que será tampa de uma piscina e ao mesmo tempo poderá proporcionar uma área de lazer otimizando espaço, e isso será muito mais prático e eficiente por meio de um controle. O projeto do controle foi elaborado a fim de ser parte da nota da disciplina de Programação Embarcada (ECOP04) e da disciplina de Laboratório de Programação Embarcada (ECOP14) da Universidade Federal de Itajuba-MG.

O projeto utiliza o LCD para mostrar mensagens na tela do controle, utiliza os LEDs para demonstração do quanto aberto/fechado o deck está, a ventoinha simbolizando o motor e o teclado é para poder escolher as opções de fechamento e abertura do deck.

O programa do Controle Deck Inteligente foi feito utilizando o programa MPLAB X IDE e sua demonstração no programa PICSIMLab é feito a partir dos seguintes passos:

Passo 1:

O primeiro passo após configurar os programas é incluir as bibliotecas que serão usadas no programa, estas que foram disponibilizadas na disciplina de ECOP14 fazem configuração da placa, operações com bits, do LCD, do teclado e do microcontrolador.

Ainda antes de declarar o main estão as funções de tempo que determinam o tempo que a ventoinha ficara ligada, a princípio estavam dentro do scope da main mas para uma melhoria do código

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

#define CLR 0x01

void tempo(void) {
    int i, j, k;
    for (i = 0; i < 1500; i++) {
        for (j = 0; j < 41; j++) {
            for (k = 0; k < 3; k++); //Delay de tempo
        }
    }
}

void time(void) {
    int i, j, k;
    for (i = 0; i < 3000; i++) {
        for (j = 0; j < 41; j++) {
            for (k = 0; k < 3; k++); //Delay de tempo
        }
    }
}
```

Fonte própria

Passo 2:

O segundo passo é declarar a função main, a primeira parte do scope da main será a declaração das variáveis do programa, em seguida as strings que apareceram na tela do LCD, logo após estão as inicializações do LCD, do teclado e da ventoinha que será a simulação do motor e por fim a configuração dos LEDs. A função pwmFrequency controla a frequência em que a ventoinha gira, a princípio estava no laço for mas para melhoria do código foi colocada antes.

```
void main(void) {  
  
    //declarando as variaveis  
    unsigned int tecla = 16;  
    int i;  
    char p;  
    volatile unsigned char j, k;  
  
    //mensagens que aparecero no lcd  
    char msg1[11] = "Abrir Tudo";  
    char msg2[13] = "Abrir Metade";  
    char msg3[11] = "Fechar Tudo";  
    char msg4[16] = "Fechar Metade";  
  
    lcdInit(); //inicializa lcd  
    kpInit(); //inicializa teclado  
    pwmInit(); //inicializa a ventoinha  
  
    //configurando entrada e saida dos leds  
    TRISD = 0x00;  
    TRISB = 0x00;  
    PORTD = 0x00;  
  
    pwmFrequency(1000); //Supondo que com f=1000 a ventoinha que simboliza o motor ira rodar pra "frente"
```

Fonte própria

Passo 3:

Ainda no scope da main inicia-se o laço for, nele ira conter os comandos do programa.

O kpdebounce faz o debounce das teclas, o primeiro if ira realizar a detecção se houve alguma mudança no estado das teclas, mudança essa que conterà o que vai ser lido no LCD, como os LEDs serão acionados e a frequência em que a ventoinha ira girar.

```

for (;;) {
    kpDebounce();

    if (kpRead() != tecla) {
        tecla = kpRead();
    }
}

```

Fonte própria

Passo 4

O primeiro teste é da tecla 3 que no programa será a bit 1 da portad, caso ela seja acionada o comando "lcdCommand(CLR)" limpará a tela do LCD e o for ira percorrer a tela mostrando a respectiva mensagem no LCD. Logo após o comando na PORTAD faz o acionamento dos Leds.

A função pwmSet configura a ventoinha para liga e desligar, chamando a função do tempo em que a ventoinha fica ligada e depois com a função lcdCommand finaliza o bloco correspondente a primeira tecla.

O mesmo processo acontece para todas as outras teclas acionadas.

O código ficara da seguinte maneira:

```

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

#define CLR 0x01

void tempo(void) {
    int i, j, k;
    for (i = 0; i < 1500; i++) {
        for (j = 0; j < 41; j++) {
            for (k = 0; k < 3; k++); //Delay de tempo
        }
    }
}

```

```

void time(void) {
    int i, j, k;
    for (i = 0; i < 3000; i++) {
        for (j = 0; j < 41; j++) {
            for (k = 0; k < 3; k++); //Delay de tempo
        }
    }
}

```

```

void main(void) {

```

```

    //declarando as variaveis
    unsigned int tecla = 16;
    int i;
    char p;
    volatile unsigned char j, k;

```

```

    //mensagens que aparecero no lcd
    char msg1[11] = "Abrir Tudo";
    char msg2[13] = "Abrir Metade";
    char msg3[11] = "Fechar Tudo";
    char msg4[16] = "Fechar Metade";

```

```

    lcdInit(); //inicializa lcd
    kpInit(); //inicializa teclado
    pwmInit(); //inicializa a ventoinha

```

```

    //configurando entrada e saida dos leds
    TRISD = 0x00;
    TRISB = 0x00;
    PORTD = 0x00;

```

pwmFrequency(1000); //Supondo que com f=1000 a ventoinha que simboliza o motor ira rodar pra "frente"

```

for (;;) {
    kpDebounce();

    if (kpRead() != tecla) {
        tecla = kpRead();

        if (bitTst(tecla, 3)) { //Realiza leitura da tecla 1
            lcdCommand(CLR);
            for (i = 0; i < 11; i++) {
                lcdData(msg1[i]); //Mostra mensagem no lcd
            }
            PORTD = 0xFF;

```

```

    pwmSet(100);

    time();

    pwmSet(0);
    lcdCommand(CLR);
}

```

```

if (bitTst(tecla, 7)) { //Realiza leitura da tecla 2
    lcdCommand(CLR);
    for (i = 0; i < 13; i++) {
        lcdData(msg2[i]);
    }
    PORTD = 0xF0;

    pwmSet(100);
    tempo();

    pwmSet(0);
    lcdCommand(CLR);
}

```

```

if (bitTst(tecla, 6)) { //Realiza leitura da tecla 5
    lcdCommand(CLR);
    for (i = 0; i < 11; i++) {
        lcdData(msg3[i]);
    }
    PORTD = 0x00;

    pwmFrequency(1500); //Supondo que com f=1500 ele rode pra "tras"
    pwmSet(100);

    time();

    pwmSet(0);
    lcdCommand(CLR);
}

```

```

if (bitTst(tecla, 2)) { //Realiza leitura da tecla 4
    lcdCommand(CLR);
    for (i = 0; i < 13; i++) {
        lcdData(msg4[i]);
    }
    PORTD = 0xF0;

    pwmSet(100);

    tempo();
}

```

```
pwmSet(0);  
lcdCommand(CLR);
```

```
}
```

```
}
```

```
}
```

```
}
```