



João Hugo Soares de Campos

Projeto de Chuveiro Eletrônico feito para PICgenios com PIC18F4520

PASSO 1: Conhecimento Prévio.

Necessário conhecimento básico de linguagem C e necessário conhecer como funciona PICgenios e o microcontrolador PIC18F4520.

PASSO 2: Planejamento do Chuveiro.

Conhecimento sobre os componentes necessários para o funcionamento do chuveiro.

Para esse projeto será utilizado um Display LCD (modelo HD44780), um sensor de temperatura e aquecedor via PWC, um keypad e um display de 7 segmentos.

PASSO 3: Definições das bibliotecas utilizadas.

Para a construção do projeto foram necessárias a utilização de 7 bibliotecas diferentes, algumas não foram explicitamente chamadas dentro do programa, mas necessárias para o funcionamento, além de 4 bibliotecas padrão da linguagem C.

```
#include <pic18f4520.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include "config.h"
#include "atraso.h"
#include "keypad.h"
#include "lcd.h"
#include "ssd.h"
#include "bits.h"
#include "adc.h"
```

No qual pic18f4520.h, string.h, stdio.h e stdlib.h são bibliotecas padrões e config.h, atraso.h, keypad.h, lcd.h, ssd.h, bits.h e adc.h são bibliotecas desenvolvidas pelo professor Rodrigo Maximiano para a manipulação do microcontrolador PIC18F4520 e seus periféricos.



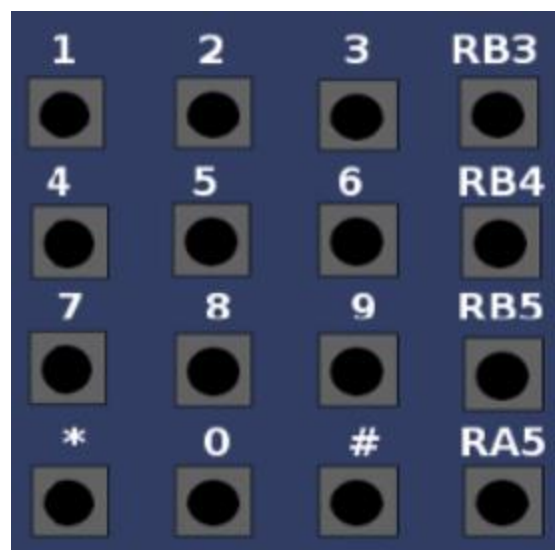
PASSO 4: DEFINIÇÕES DE PERFIS

Para o funcionamento do programa é necessário definir 4 perfis diferentes, em que 1 será o perfil utilizado durante todo o programa e outros 3 serão as possíveis escolhas para usuário, cada perfil personalizável terá 3 temperaturas diferentes.

```
char Perfil_Utilizado[6], str[6];  
int Temp1, Temp2, Temp3;  
  
char Perfil_1[6] = "P1";  
int Perfil_1_Temp1 = 31;  
int Perfil_1_Temp2 = 41;  
int Perfil_1_Temp3 = 51;  
  
char Perfil_2[6] = "P2";  
int Perfil_2_Temp1 = 32;  
int Perfil_2_Temp2 = 42;  
int Perfil_2_Temp3 = 52;  
  
char Perfil_3[6] = "P3";  
int Perfil_3_Temp1 = 33;  
int Perfil_3_Temp2 = 43;  
int Perfil_3_Temp3 = 53;
```

PASSO 5: ESCOLHA DOS PERFIS

Após iniciar o programa, será necessário definir o perfil que será utilizado entre as 3 possíveis opções, para escolher o programa é utilizado 3 botões ao lado do keypad, as teclas de interrupção externa, RB3, RB4 e RB5.





Para que o perfil seja escolhido e que não seja pulado pelo usuário, somente esses três botões faz com que o programa continue, é utilizado um “while”, o programa é impossibilitado de continuar.

```
////////ESCOLHER PERFIL////////

lcd_cmd(L_CLR);
lcd_cmd(L_L1);
lcd_str("Escolher perfil");
lcd_cmd(L_L2+4);
lcd_str("Aperte: ");
atraso_ms(2000);

lcd_cmd(L_CLR);
lcd_cmd(L_L1);
lcd_str("P1 -RB3");
lcd_cmd(L_L1 + 8);
lcd_str("P2 -RB4");
lcd_cmd(L_L2);
lcd_str("P3 -RB5");

TRISB = 0x38;
while(PORTBbits.RB3 && PORTBbits.RB4 && PORTBbits.RB5);

if (PORTBbits.RB3 == 0){
    strcpy(Perfil_Utilizado,Perfil_1);
    Temp1 = Perfil_1_Temp1;
    Temp2 = Perfil_1_Temp2;
    Temp3 = Perfil_1_Temp3;
}

if (PORTBbits.RB4 == 0){
    strcpy(Perfil_Utilizado,Perfil_2);
    Temp1 = Perfil_2_Temp1;
    Temp2 = Perfil_2_Temp2;
    Temp3 = Perfil_2_Temp3;
}

if (PORTBbits.RB5 == 0){
    strcpy(Perfil_Utilizado,Perfil_3);
    Temp1 = Perfil_3_Temp1;
    Temp2 = Perfil_3_Temp2;
    Temp3 = Perfil_3_Temp3;
}
```



PASSO 6: DEFINIR A TEMPERATURA

Após definir o perfil, é necessário definir a temperatura para o chuveiro, igual anteriormente, o programa te define as opções já pré-definidas, e trava o programa até que seja escolhida pelas teclas de interrupção externa, RB3, RB4 e RB5.

```
////////ESCOLHER TEMPERATURA////////

lcd_cmd(L_CLR);
lcd_cmd(L_L1);
lcd_str(" Definir Temp");
lcd_cmd(L_L2+4);
lcd_str("Aperte: ");
atraso_ms(2000);

lcd_cmd(L_CLR);
lcd_cmd(L_L1);
itoa(Temp1, str);
lcd_dat(str[2]);
lcd_dat(str[3]);
lcd_dat(str[4]);
lcd_str("-RB3");
lcd_cmd(L_L1 + 8);
itoa(Temp2, str);
lcd_dat(str[2]);
lcd_dat(str[3]);
lcd_dat(str[4]);
lcd_str("-RB4");
lcd_cmd(L_L2);
itoa(Temp3, str);
lcd_dat(str[2]);
lcd_dat(str[3]);
lcd_dat(str[4]);
lcd_str("-RB5");

TRISB = 0x38;
while(PORTBbits.RB3 && PORTBbits.RB4 && PORTBbits.RB5);

int Temp_Escolhida = 0;;

if (PORTBbits.RB3 == 0){
    Temp_Escolhida = Temp1;
}

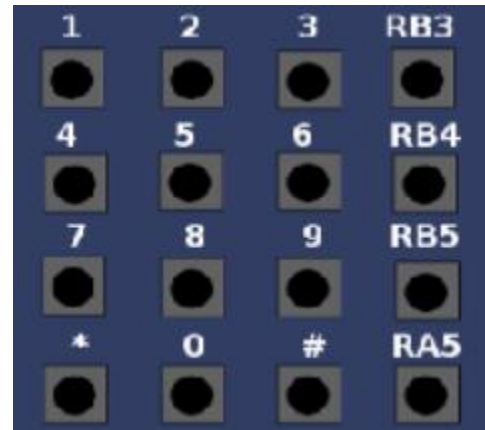
if (PORTBbits.RB4 == 0){
    Temp_Escolhida = Temp2;
}

if (PORTBbits.RB5 == 0){
    Temp_Escolhida = Temp3;
}
```



PASSO 6: DEFINIR TEMPO DE BANHO

A partir daqui começaram a aparecer problemas, para definir o tempo de banho foram necessários utilizar o teclado keypad e o display LCD simultaneamente, além de que seria necessário converter os números digitados todos em segundos, sendo que para o usuário, após os ":" no display LCD, seriam minutos convertidos para segundos.



```
//////////TEMPO BANHO//////////
lcd_cmd(L_L1);
lcd_str("Tempo de banho ");
lcd_cmd(L_L2);
lcd_str("Definir: ");
lcd_cmd(L_L2 + 10);
lcd_str("00:00");

ADCON1 = 0x00;

unsigned int tecla = 0;
int cont = 0, total = 0, i = 0, valor, Tempo_Total = 0;
char tempo[6];

lcd_cmd(L_L2 + 10);
kpInit();

while(total != 4) {
    kpDebounce();

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

        if (bitTst(tecla, 3)) { //Tecla 1
            lcd_dat('1');
            tempo[i] = '1';
            valor = 1;
            i++;
            cont++;
            total++;
        } else if (bitTst(tecla, 7)) { //Tecla 2
            lcd_dat('2');
            tempo[i] = '2';
            valor = 2;
            i++;
            cont++;
            total++;
        } else if (bitTst(tecla, 11)) { //Tecla 3
            lcd_dat('3');
            tempo[i] = '3';
            valor = 3;
            i++;
            cont++;
            total++;
        } else if (bitTst(tecla, 2)) { //Tecla 4
```

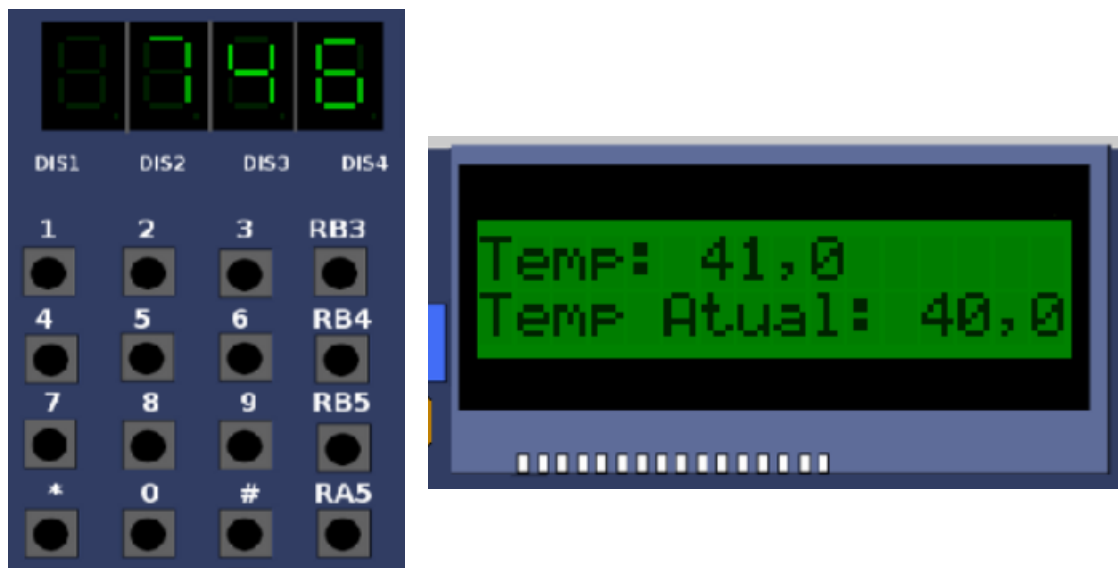


```
} else if (bitTst(tecla, 2)) { //Tecla 4
    lcd_dat('4');
    tempo[i] = '4';
    valor = 4;
    i++;
    cont++;
    total++;
} else if (bitTst(tecla, 6)) { //Tecla 5
    lcd_dat('5');
    tempo[i] = '5';
    valor = 5;
    i++;
    cont++;
    total++;
} else if (bitTst(tecla, 10)) { //Tecla 6
    lcd_dat('6');
    tempo[i] = '6';
    valor = 6;
    i++;
    cont++;
    total++;
} else if (bitTst(tecla, 1)) { //Tecla 7
    lcd_dat('7');
    tempo[i] = '7';
    valor = 7;
    i++;
    cont++;
    total++;
} else if (bitTst(tecla, 5)) { //Tecla 8
    lcd_dat('8');
    tempo[i] = '8';
    valor = 8;
    i++;
    cont++;
    total++;
} else if (bitTst(tecla, 9)) { //Tecla 9
    lcd_dat('9');
    tempo[i] = '9';
    valor = 9;
    i++;
    cont++;
    total++;
} else if (bitTst(tecla, 4)) { //Tecla 0
    lcd_dat('0');
    tempo[i] = '0';
    valor = 0;
    i++;
    cont++;
    total++;
}
if (cont != 0 && cont % 2 == 0){
    lcd_cmd(0b00010100);
    cont = 0;
}
if (i == 1){
    Tempo_Total += valor*300;
}
if (i == 2){
    Tempo_Total += valor*30;
}
if (i == 3){
    Tempo_Total += valor*5;
}
if (i == 4){
    Tempo_Total += valor;
}
}
}
```



PASSO 7: CHUVEIRO FUNCIONAR

Essa obviamente foi a parte mais complicada do projeto inteiro, desde fazer um contador usando os 4 displays de 7 segmentos funcionar corretamente com o valor em segundos, fazer com que o leitor de temperatura digital da placa funcionar corretamente, ligar e desligar o aquecedor nas horas certas para que ele não fique ligando desnecessariamente, além de que deixando a possibilidade da temperatura puder ser aumentada ou diminuída pelo o usuário durante o funcionamento.



É necessário informar o usuário que é possível aumentar ou diminuir a temperatura escolhida e configurar as portas que serão utilizadas durante o funcionamento, ligando juntamente os displays de 7 segmentos e o conversor A/D.

```
////////CHUVEIRO//////////
lcd_cmd(L_CLR);
lcd_cmd(L_L1);
lcd_str("Aumenta Temp-RB3");
lcd_cmd(L_L2);
lcd_str("Diminui Temp-RB4");
atraso_ms(2000);
lcd_cmd(L_CLR);

ADCON2 = 0x00;
TRISA = 0x03;
TRISC = 0x00;
TRISB = 0x18;
TRISD = 0;
ADCON1 = 0x06;

unsigned int Temp_Atual;
float Tempo_Flicker;

ssdInit();
adcInit();
```



Para que o programa funcione corretamente, é necessário que o tempo diminua com intervalos de segundos tomando cuidado para que o tempo de flicker não seja muito aparente, a temperatura seja calculada corretamente pela conversão, que seja possível alterar a temperatura escolhida durante todo o tempo e que o aquecedor ligue nos momentos certos quando a temperatura escolhida fosse maior do que a temperatura do PICGENIOS.

```
while(Tempo_Total != 0) {

    lcd_cmd(L_L1);
    lcd_str("Temp: ");
    lcd_dat((10*Temp_Escolhida/100)%10 + '0');
    lcd_dat((10*Temp_Escolhida/10)%10 + '0');
    lcd_dat(',');
    lcd_dat((10*Temp_Escolhida/1)%10 + '0');

    if (PORTBbits.RB3 == 0){
        Temp_Escolhida++;
    }

    if (PORTBbits.RB4 == 0){
        Temp_Escolhida--;
    }

    lcd_cmd(L_L2);
    lcd_str("Temp Atual: ");
    //Temp_Atual = 10*adcRead(2)/2;
    Temp_Atual = (adcRead(2)* 0.0048828125 * 100);
    lcd_dat((10*Temp_Atual/100)%10 + '0');
    lcd_dat((10*Temp_Atual/10)%10 + '0');
    lcd_dat(',');
    lcd_dat((10*Temp_Atual/1)%10 + '0');

    if (Temp_Escolhida > Temp_Atual){
        PORTCbits.RC5 = 1;
    }
    else{
        PORTCbits.RC5 = 0;
    }
    ///////////CONTADOR DISPLAY DE SETE SEGMENTOS//////////

    Tempo_Total--;
    for (int i=0; i<40; i++){

        ssdDigit(((Tempo_Total/1)%10),3);
        ssdDigit(((Tempo_Total/10)%10),2);
        ssdDigit(((Tempo_Total/100)%10),1);
        ssdDigit(((Tempo_Total/1000)%10),0);

        ssdUpdate();
        for (Tempo_Flicker = 0; Tempo_Flicker < 100; Tempo_Flicker++);
    }
}
```

É necessário calibrar ainda mais o tempo de flicker juntamente com a temperatura proveniente do conversor A/D para que o programa funcione mais bonito para os olhos.



PASSO 8: CONSIDERAÇÕES FINAIS

Após semanas de testes e muitas dores de cabeça para fazer com que o programa funcionasse do jeito mais correto possível, o programa teve varias versões em que foi necessário testar cada parte separadamente para que ele funcionasse corretamente dentro do programa.

Foi uma grande experiência trabalhar com uma picgenios, no qual sempre é necessário fazer com que cada parte funcione corretamente e que não interfira uma parte com a outra.