



```
#include <reg51.h>
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
#define LCD P2
```

```
#define ADC P3
```

```
#define SHTAddr 0x80
```

```
//bool data type
```

```
typedef enum { false, true } bool;
```

```
bool key_pressed=false;
```

```
bool welcome = true;
```

```
bool mainScreen=true;
```

```
//LED Pins
```

```
sbit LED_UP = P1^4;  
sbit LED_DOWN = P1^5;  
sbit LED_OK = P1^6;  
sbit LED_BACK = P1^7;
```

```
//Button Pins
```

```
sbit UP = P0^4;  
sbit DOWN = P0^5;  
sbit OK = P0^6;  
sbit BACK = P0^7;
```

```
//
```

```
sbit RS = P1^0;  
sbit E = P1^1;  
sbit eoc = P0^0;  
sbit oe = P0^1;  
sbit start = P0^2;
```

```
//SCL, SDA
```

```
sbit SCL=P1^2;  
sbit SDA=P1^3;
```

```
//variables
```

```
int choice=0;  
float num;  
float humidity=0;  
char result[1];  
unsigned char ack1, hum[2];
```

```
const float bal_res=5000;
const float max_adc=256;
const float beta=3965;
const float room_temp=298.15;
const float res_room_temp=5000;
```

```
//LCD functions
```

```
void delay(int);
void LCD_init(void);
void cmd(char);
void dispString(char*);
void LCD_disp(char);
```

```
//Sensor functions
```

```
void readThermistor();
void readHum();
void readHum1();
```

```
//ADC functions
```

```
void read_ADC();
```

```
//I2C functions
```

```
void I2C_init(); //initialize I2C pins
void I2C_start(); //send start bit on I2C
void I2C_restart(); //send start bit on I2C
unsigned char I2C_writeByte(unsigned char); //send data on I2C
unsigned char I2C_readByte(); //read value from I2C
void I2C_sendACK(); //send ACK bit on i2c
void I2C_sendNACK(); //send NACK bit on i2c
```

```
void I2C_stop(); //send stop bit on I2C
```

```
//Menu functions
```

```
void startScreen();
```

```
void optionSelect();
```

```
void mainScreenP();
```

```
void goBack();
```

```
void logOut();
```

```
// Main function
```

```
void main(void)
```

```
{
```

```
    //INPUT-OUTPUT defining section
```

```
    OK=1; //making input
```

```
    UP=1; //making input
```

```
    DOWN=1; //making input
```

```
    BACK=1; //making input
```

```
    LED_UP=0; //making output
```

```
    LED_DOWN=0; //making output
```

```
    LED_OK=0; //making output
```

```
    LED_BACK=0; //making output
```

```
    ADC=0xFF; // making input
```

```
    LCD=0x00; //making output
```

```
    eoc=1;
```

```
    oe=0;
```

```
    start=0;
```

```
//Initializing the LCD
```

```
LCD_init();
```

```
//Initializing the I2C communication
```

```
I2C_init();
```

```
while(1)
```

```
{
```

```
    OK=1;//re-establish OK as input
```

```
    if(key_pressed==false)
```

```
    {dispString("Press any key");
```

```
    cmd(0xC0);
```

```
    dispString("to continue...");}
```

```
    while(key_pressed==false)
```

```
    {
```

```
        if(UP==0)
```

```
        {
```

```
            key_pressed=true;
```

```
            LED_UP=1;
```

```
            delay(100);
```

```
            LCD_init();
```

```
            dispString("Starting up...");
```

```
            delay(400);
```

```
            LED_UP=0;
```

```
        }
```

```
    else
```

```
        if(DOWN==0)
```

```
{
    key_pressed=true;
    LED_DOWN=1;
    delay(100);
    LCD_init();
    dispString("Starting up...");
    delay(400);
    LED_DOWN=0;
}
else
    if(OK==0)
    {
        key_pressed=true;
        LED_OK=1;
        delay(100);
        LCD_init();
        dispString("Starting up...");
        delay(400);
        LED_OK=0;
    }
else
    if(BACK==0)
    {
        key_pressed=true;
        LED_BACK=1;
        delay(100);
        LCD_init();
        dispString("Starting up...");
        delay(400);
```

```

        LED_BACK=0;
    }
}

if(key_pressed==true)
{
    if(welcome==true)
    {
        LCD_init();
        startScreen();
        welcome=false;
    }
    if(welcome==false)
    {
        if(mainScreen==true)
        {
            while(OK==1)
            {
                startScreen();
                optionSelect();
                mainScreenP();
                if(BACK==0)
                {
                    OK=0;
                    logOut();
                }
            }
            mainScreen=false;
        }
    }
}

```

```

if(mainScreen==false)
{
    //mainScreen=true;
    if(choice==1)
    {
        LCD_init();
        readThermistor();
        goBack();
    }
    else
        if(choice==2)
        {
            readHum();
            delay(500);
            readHum1();
            goBack();
        }
        else
        {
            mainScreen=true;
        }
    }
}
}
}
}

```



```
//LCD functions
```

```
void cmd(char t)
```

```
{  
    LCD = t;  
    RS=0;  
    E=1;  
    delay(5);  
    E=0;  
}
```

```
void LCD_init()
```

```
{  
    cmd(0x38); //data init  
    cmd(0x0C); //LCD display on and cursor off  
    cmd(0x01); //clear LCD display  
    cmd(0x80); //positioning cursor at the first line  
}
```

```
void delay(int n)
```

```
{  
    int i,j;  
    for(i=0;i<n;i++)  
        for(j=0;j<100;j++)  
            {}  
}
```

```
void dispString(char *p)
```

```
{
```

```
        while(*p)
        {
            LCD_disp(*p++);
        }
    }
```

```
void LCD_disp(char x)
{
    LCD=x;
    RS=1;
    E=1;
    delay(5);
    E=0;
}
```

```
//ADC functions
void read_ADC()
{
    num=0;
    start=1;
    delay(5);
    start=0;
    while(eoc==1);
    while(eoc==0);
    oe=1;
    num=ADC;
    delay(5);
    oe=0;
}
```

```
//Menu functions
```

```
void startScreen()
```

```
{  
    cmd(0x80);  
    dispString("1.Temperature");  
    cmd(0xC0);  
    dispString("2.Humidity");  
}
```

```
void optionSelect()
```

```
{  
    if(UP==0)  
        choice=1;  
    else  
        if(DOWN==0)  
            choice=2;  
        else  
            choice=choice;  
}
```

```
void mainScreenP()
```

```
{  
    if(choice==1)  
    {  
        cmd(0x80);  
        dispString(">.Temperature");  
        cmd(0xC0);  
        dispString("2.Humidity");  
    }
```

```

    }
    else
    if(choice==2)
    {
        cmd(0x80);
        dispString("1.Temperature");
        cmd(0xC0);
        dispString(">.Humidity");
    }
}

```

```

void goBack()
{
    while(BACK==1);
    LCD_init();
    mainScreen=true;
    key_pressed=true;

}

```

```

void logOut()
{
    LCD_init();
    cmd(0x80);
    dispString("System powering");
    cmd(0xC0);
    dispString("down...");
    key_pressed=false;
    choice=0;
}

```

```
        mainScreen=true;

        welcome=true;

        delay(500);

        LCD_init();
    }
```

```
//I2C functions
```

```
void I2C_init()
```

```
{
    SDA=1;

    SCL=1;
}
```

```
void I2C_start()
```

```
{
    SDA=0;

    SCL=0;
}
```

```
void I2C_restart()
```

```
{
    SDA=1;

    SCL=1;

    SDA=0;

    SCL=0;
}
```

```
void I2C_stop()
```

```
{
    SDA=0;

    SCL=0;

    SDA=1;
}
```

```

        SCL=1;
    }
void I2C_sendACK()
{
    SDA=0;
    SCL=1;
    SCL=0;
    SDA=1;
}
void I2C_sendNACK()
{
    SDA=1;
    SCL=1;
    SCL=0;
    SDA=1;
}
unsigned char I2C_writeByte(unsigned char myData)
{
    unsigned char i;
    for(i=0;i<8;i++)
    {
        if((myData & 0x80)==0)
            SDA=0;
        else
            SDA=1;
        SCL=1;
        SCL=0;
        myData=myData<<1;
    }
}

```

```

        SDA=1;

        SCL=1;

        SCL=0;

        return SDA;
    }

    unsigned char I2C_readByte()
    {
        unsigned char i, myData=0;
        for(i=0;i<8;i++)
        {
            SCL=1;

            if(SDA)

                myData=myData | 1;

            if(i<7)

                myData=myData<<1;

            SCL=0;
        }

        return myData;
    }

```

//Sensor functions

```

void readThermistor()
{
    float rTh=0;

    float tKel=0;

    float tCel=0;

    char x=0xDF; //ascii value of degree(°) symbol

    read_ADC(); //puts the ADC value in num

```

```

    rTh=bal_res*((max_adc/num)-1);
    tKel=(beta*room_temp)/(beta+(room_temp*log(rTh/res_room_temp)));
    tCel=tKel-273.15;
    sprintf(result,"%0.2f",tCel);
    LCD_init();
    cmd(0x80);
    dispString("Temperature: ");
    //cmd(0xC0);
    dispString(result);
    LCD_disp(x);
    dispString("C");
}

void readHum()
{
    //write to slave device
    I2C_start();
    ack1=I2C_writeByte(SHTAddr);
    ack1=I2C_writeByte(0xE5);
    //ack1=I2C_writeByte();
    I2C_stop();

    //read from slave device
    I2C_start();
    I2C_writeByte(SHTAddr|1); //slave address with read byte
    hum[0]=I2C_readByte();//read MSB
    I2C_sendACK();
    hum[1]=I2C_readByte();//read LSB
    I2C_sendNACK();
    I2C_stop();
}

```



```

        //hum conversion
        humidity=((hum[0]*256.0+hum[1])*125.0)/65536.0-6;
    }
    void readHum1()
    {
        //write to slave device
        I2C_start();
        ack1=I2C_writeByte(SHTAddr);
        ack1=I2C_writeByte(0xE5);
        //ack1=I2C_writeByte();
        I2C_stop();

        //read from slave device
        I2C_start();
        I2C_writeByte(SHTAddr|1); //slave address with read byte
        hum[0]=I2C_readByte();//read MSB
        I2C_sendACK();
        hum[1]=I2C_readByte();//read LSB
        I2C_sendNACK();
        I2C_stop();

        //hum conversion
        humidity=((hum[0]*256.0+hum[1])*125.0)/65536.0-6;

        LCD_init();
        dispString("Humidity: ");
        sprintf(result,"%0.2f",humidity);
        dispString(result);
    }
}

```

```
dispString("%RH");
```

```
}
```