

#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-OUPUT 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 dispaly

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,"%.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,"%.2f",humidity);

dispString(result);

dispString("%RH");

}