

```

/*
CopyRights 2020/7/18
    机械工程学院
    机械电子工程专业
    17 级机械电子工程 2 班盛梓茂
    《基于 arduino 和 esp8266 的智能家居设计》
AllRights Reserved
*/
#include<SD.h>
#include<SPI.h>
#include<dht11.h>    //测温度和湿度
#include<MsTimer2.h>
#include<math.h>
#include<LiquidCrystal_I2C.h>    // lcd 点亮显示数
#include<math.h>
#define tempGet A0    //温湿度传感器接口
#define Aout A1    //有毒传感器接收口
#define clockLed 9    //时间响应口 ,亮灯口
#define timeinterval 60    //设置时间间隔，即每过多长时间闹钟闹一
次（把闹钟用小灯代替）
#define lightLed A2    //室外光线读取口
#define homeLight 8    //大厅的灯为 A3 口
#define pinCS 10 // Pin 10 on Arduino Uno    //CS 口
#define jidianqi A3
#define timeinterval2 4    //存储时间间隔 4s
int RECV_PIN = 7;
//IRrecv irrecv(RECV_PIN);
//decode_results results;
void saveSD(float temperature,float humidity,float gasvalue);
void musicClock();    // 灯亮
struct ss getTemh();
void lcdShownumber();
float gasValue(); //对有毒气体的模拟值进行分析运算返回影响
void serialShow() ;
dht11 DHT11;
File myFile;    //文件定义
LiquidCrystal_I2C lcd(0x27,16,2);
/*int SDA = A4;
int SCL = A5;*/
struct ss    //温湿度变量
{
    float temperature;
    float humidity;
}result;

```

```

static int lastConnectionTime = millis()/1000;
static int lastConnectionTime2 = millis()/1000;
float gasvalue;
float value_balance = 80;
int pinBuzzer = 3;
void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
    pinMode(pinCS, OUTPUT);
    pinMode(lightLed,INPUT);
    pinMode(homeLight,OUTPUT);
    pinMode(Aout,INPUT);
    pinMode(clockLed,OUTPUT);
    pinMode(pinBuzzer,OUTPUT);
    pinMode(RECV_PIN,INPUT);
    lcd.begin(16,2);
    lcd.setBacklight(HIGH);
}
void loop(){
    int Keytemp = 0;           //存储按键值
    alarmClock();              //时间到了就触发中断自动响铃 ? 如何让程序在相邻的同
    时还亮灯有毒气体电压数值
    result = getTemh();        //读取温度和湿度值
    gasvalue = gasValue();     //读取有毒气体电压
    if(gasvalue>value_balance)//电压超过 50 警报 3 秒 //有毒气体报警器
    {
        police();
        // digitalWrite(2,HIGH);
        // delay(3000);
        analogWrite(jidianqi,1020);
    }
    else
    {
        analogWrite(jidianqi,100);

        saveSD(result.temperature,result.humidity,gasvalue); //保存到 sd 卡

        serialShow();      //串口显时
        lcdShownumber();
    }
    void saveSD(float temperature,float humdity,float gasvalue)
    {
        if(millis()/1000-lastConnectionTime2>timeinterval2) //4s 存一次
        {

```

```

        lastConnectionTime2 = millis()/1000;
    if(SD.begin())
    {
        Serial.println("SD card is ready to use.");
    }
    else
    {
        Serial.println("SD card initialization failed");
        return;
    }
    myFile = SD.open("test.txt", FILE_WRITE);
    if (myFile)
    {
        myFile.print("ShengZlmao and ShiDawen");
        myFile.print(",");
        myFile.print("time:");
        myFile.print(millis()/1000);
        myFile.print(",");
        myFile.print("temperature:");
        myFile.print(temperature);
        myFile.print(",");
        myFile.print(" humidity:");
        myFile.print(humidity);
        myFile.print(",");
        myFile.print("gas_value:");
        myFile.println(gasvalue);
        myFile.close(); // close the file
    }
    else
    { Serial.println("error opening test.txt");} // if the file didn't open, print an error:
}
}

void alarmClock() //当时间间隔到达时自动响铃
{
    //int nowtime = nowTime();
    if(millis()/1000-lastConnectionTime>timeinterval)
    {
        musicClock();
        lastConnectionTime = millis()/1000;
    }
}

void musicClock() // 灯亮

```

```

{
    digitalWrite(clockLed,HIGH);
    delay(1500);           //时间到了灯亮 1.5s 后熄灭
    digitalWrite(clockLed,LOW);
    delay(20);
}
struct ss getTemh()//测得温度并将温度反映
{
    Serial.println("\n");
    int chk = DHT11.read(tempGet);
    Serial.print("Read sensor:");
    switch(chk)
    {
        case DHTLIB_OK:           Serial.println("OK!");break;
        case DHTLIB_ERROR_CHECKSUM:Serial.println("CHECKSUM ERROR!");break;
        case DHTLIB_ERROR_TIMEOUT: Serial.println("TIMEOUT ERROR!");break;
        default:                   Serial.println("UNKNOWN ERROR!");break;
    }
    struct ss result;
    result.temperature = DHT11.temperature;
    result.humidity = DHT11.humidity;
    return result;
    Serial.print("Temperature:(摄氏度)");
    Serial.println((float)DHT11.temperature,2);
    Serial.print("Humidity:(%)");
    Serial.println((float)DHT11.humidity,2);
    delay(2000);
}
void lcdShownumber()
{
    lcd.init();
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("t:");
    lcd.print(millis()/1000);//反映系统当前时间。
    lcd.print("s");
    lcd.print(" ");
    lcd.print("tem:");
    lcd.print((int)result.temperature);
    lcd.print((char)223);
    lcd.print("C");
    lcd.setCursor(0,1);
    lcd.print("hum:");
    lcd.print((int)result.humidity);
}

```

```

        lcd.print("%");
        lcd.print(" ");
        lcd.print("PPM:");
        lcd.print((int)gasvalue);
    }
    float gasValue() //对有毒气体的模拟值进行分析运算返回影响
    {
        float ppm    = analogRead(Aout);

        return ppm;
    }

    void police()
    {
        long frequency = 300;
        tone(pinBuzzer,frequency);
        delay(1000);
        noTone(pinBuzzer);
        delay(2000);
    }

    void serialShow()
    {
        Serial.print("nowtime_s:");
        Serial.println(millis());
        Serial.print("Temperature_sheshidu:");           //串口显示当前温度和湿度值
        Serial.println((float)result.temperature,2);
        Serial.print("Humidity_%:");
        Serial.println((float)result.humidity,2);
        Serial.print("ppm:");
        Serial.println(gasvalue);
        /*Serial.print("ppm:");
        Serial.println(gasDealwith(gasvalue));*/
    }

```