**《Arduino程序设计项目实践》报 告**

**题 目：** 基于Arduino的家庭环境检测系统

目录

[第一章 绪论 3](#_Toc44661478)

[1.1实训设计背景与意义 3](#_Toc44661479)

[1.2实训设计任务 3](#_Toc44661480)

[1.3实训具体内容 4](#_Toc44661481)

[第二章 硬件设计 5](#_Toc44661482)

[2.1硬件设计思路 5](#_Toc44661483)

[2.2硬件选择 5](#_Toc44661484)

[2.2.1 DHT11温湿度传感器 6](#_Toc44661485)

[2.2.2 OLED显示器 7](#_Toc44661486)

[2.2.3无源蜂鸣器 8](#_Toc44661487)

[2.2.4 LED小灯 9](#_Toc44661488)

[2.3电路连接 9](#_Toc44661489)

[第三章 软件设计 10](#_Toc44661490)

[3.1输入元件程序设计 11](#_Toc44661491)

[3.2输出元件程序设计 12](#_Toc44661492)

[3.2.1 OLED显示器 12](#_Toc44661493)

[3.2.2 LED灯 13](#_Toc44661494)

[3.2.3 无源蜂鸣器 14](#_Toc44661495)

[第四章 系统调试 14](#_Toc44661496)

[第五章 实物展示 15](#_Toc44661497)

[第六章 总结 17](#_Toc44661498)

[附件：程序清单 17](#_Toc44661499)

# 第一章 绪论

## 1.1实训设计背景与意义

随着生活水平的提高和科技水平的发展，人们对自身和家庭安全方面的考虑也越来越多，越来越多的安全监测系统和设备进入到人们的生活中，如家庭防火系统、家庭防盗系统、家庭远程监控装置等。各种现代高科技逐渐被用于各行业和领域，为人类社会生产力的发展形成了强大动力。本设计将温湿度检测系统应用到家庭当中，通过检测家庭中温湿度的变化反应家中环境的变化，在环境出现异常时，能够进行一定的报警，让系统使用者及时发现家庭中存在的安全隐患，从而快速解决安全问题，具有一定的社会意义和应用价值。

## 1.2实训设计任务

本设计的任务主要是应用Arduino Uno主控制器，搭载温湿度传感器和led、蜂鸣器、OLED显示器等，实现温湿度的检测功能和报警功能，在环境出现异常的情况下，能够及时通知系统使用者处理异常，起到很好的安全防范的作用。

## 1.3实训具体内容

实训的具体内容如下：

设计一个家庭安全监测系统，主要可以检测温度和湿度这两个关键的环境变量，这两个关键变量将通过温湿度传感器DHT11获取，获取到数据后进行判断，当环境出现异常，比如温度过高时，能够触发蜂鸣器进行警报，提示使用者温度异常，疑似有火情；另外，当环境中湿度变化时可以触发LED的亮灭，当湿度较低时，点亮LED灯，提醒使用者当前环境干燥，可以进行一定的加湿处理。当前环境的温湿度数据进行一定的可视化，将通过OLED显示器显示出来，方便使用者随时监测环境的情况。

1. 学习Arduino Uno的使用方法，熟悉Arduino的编程语法，掌握其控制与运算原理，将其作为主控制器接收来自传感器的数据并进行数据解析进而控制一些元件的工作。
2. 学习并掌握温湿度传感器DHT11的使用，掌握其库函数的用法。
3. 学习并掌握OLED显示器的使用，掌握其文本显示的方法、变量显示的方法。
4. 掌握LED、蜂鸣器的使用。

# 第二章 硬件设计

## 2.1硬件设计思路

硬件设计包括对硬件的选材、通讯方式、接线方式的设计，主控制器为Arduino Uno，其芯片为ATMEGA328P；主要应用到的传感器为DHT11温湿度传感器，可以检测温湿度的信息；输出元件有OLED、蜂鸣器、LED。传感器将获取环境的温湿度信息，并将其传输至Arduino，Arduino获得数据后进行数据解析，并且根据设置好的阈值进行声光报警的设计，同时将解析出的温湿度信息进行OLED显示器的显示。

## 2.2硬件选择

硬件选材上主要用到以下材料：

1. 主控制器：Arduino Uno
2. 传感器：DHT11温湿度传感器
3. 显示元件：OLED液晶显示器
4. 发光元件：LED小灯
5. 发声元件：无源蜂鸣器
6. 面包板、数据线、杜邦线若干

### 2.2.1 DHT11温湿度传感器

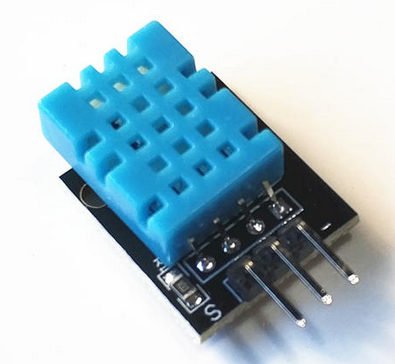


图2.1 DHT11温湿度传感器实物图

1）硬件描述

此传感器是一款含有已校准数字信号输出的温湿度复合传感器。它应用专用的数字模块采集技术和温湿度传感技术，确保产品具有极高的可靠性与卓越的长期稳定性。传感器包括一个电阻式感湿元件和一个NTC测温元件，并与一个高性能8位单片机相连接。因此该产品具有品质卓越、超快响应、抗干扰能力强、性价比极高等优点。每个DHT11传感器都在极为精确的湿度校验室中进行校准。校准系数以程序的形式储存在OTP内存中，传感器内部在检测信号的处理过程中要调用这些校准系数。单线制串行接口，使系统集成变得简易快捷。超小的体积、极低的功耗，信号传输距离可达20米以上，使其成为各类应用甚至最为苛刻的应用场合的最佳选则。因此，本设计选择DHT11作为环境检测的传感器。

2）硬件参数：

1. 工作电压 ：3.3V-5v
2. 尺寸大小：40mm x 28mm
3. 重量大小：3g
4. 信号类型：数字信号
5. 供电电流：最大2.5mA
6. 温度范围：0-50℃，误差±2℃
7. 湿度范围：当环境温度在 0 ℃时为30-90%RH;当环境温度在25℃时为20-90%RH ;当环境温度在50℃时为20-80%RH
8. 响应时间：1/e(63%) 6-30s
9. 测量分辨率分别为：温度（8bit）湿度（8bit）
10. 采样周期间隔：<1s

### 2.2.2 OLED显示器

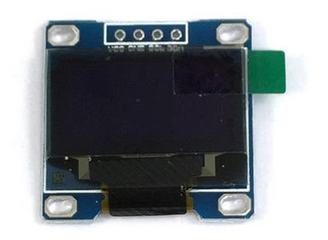


图2.2 OLED液晶显示器实物图

1）硬件描述

0.96寸OLED模块采用高亮度，低功耗的OLED屏，显示颜色纯正，在阳光下有很好的可视效果。模块供电可以是3.3V也可以是5V，不需要修改模块电路，OLED屏具有多个控制指令，可以控制OLED的亮度、对比度、开关升压电路等指令，操作方便，功能丰富，因此，本设计选用这款OLED作为显示元件。

2）硬件参数

1. 高分辨率：12864（和12864LCD相同分辨率，但该OLED屏的单位面积像素点多）
2. 超广可视角度：大于160°
3. 超低功耗：正常显示时0.06W
4. 宽供电范围：直流3V-5V
5. 工业级：工作温度范围-30℃~70℃
6. 体积小：27mm27mm\*2mm
7. 通信方式：SPI、IIC
8. 亮度、对比度可以通过程序指令控制
9. 使用寿命不少于16000小时
10. OLED屏幕内部驱动芯片：SSD1306

### 2.2.3无源蜂鸣器



图2.3 无缘蜂鸣器实物图

 蜂鸣器是一种一体化结构电子讯响器，采用直流电源供电，广泛应用于计算机、打印机、复印机、报警器、电子玩具、汽车电子设备、电话机、定时器等电子产品中作发声器件。蜂鸣器主要分为有源蜂鸣器和无源蜂鸣器，本设计使用的是无源蜂鸣器，通过控制PWM的输出可以控制蜂鸣器的鸣叫频率。

### 2.2.4 LED小灯

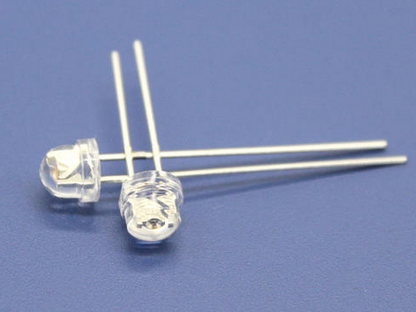


图2.4 LED发光二极管

Led小灯为一种发光二极管，能够发出不同颜色的灯光，本设计使用led的目的在于提醒使用者环境的情况，当湿度不符合我们的要求时，led能够进行闪烁，从而提醒使用者。

## 2.3电路连接

电路连接采用Altium Designer软件绘制，温湿度传感器为5V供电，其中信号输出端连接到Arduino的8号I/O引脚上；OLED显示器的通讯方式为IIC通讯，因此除了供电的两根线外，只需要一根信号线SDA和一根时钟线SCK就可以。无源蜂鸣器正极接到3号I/O引脚，原因在于Arduino的3号引脚支持PWM输出，从而可以控制蜂鸣器的鸣叫频率；LED小灯正极接到4号I/O引脚。

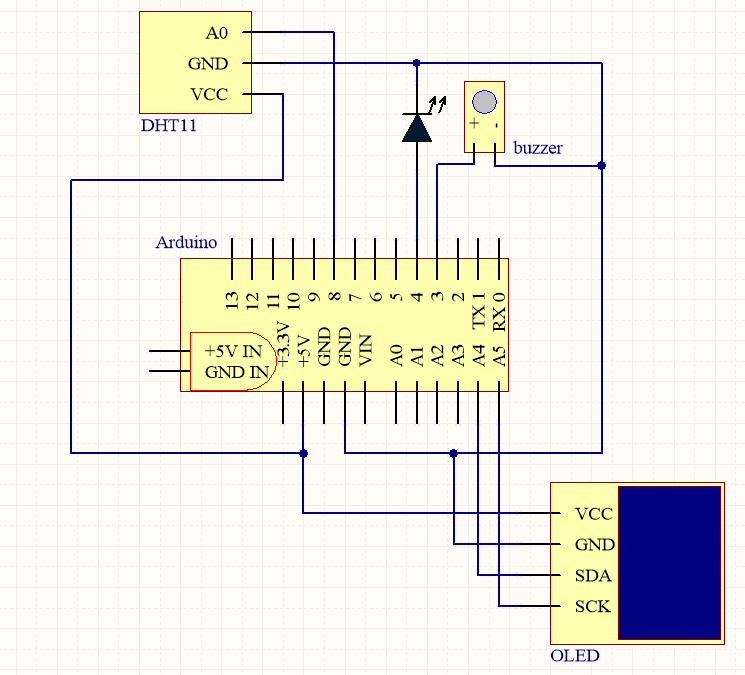


图2.5 硬件接线图

# 第三章 软件设计

本设计的主控制器为Arduino Uno，控制该控制器最常用的语言为C/C++混合编程，本设计也使用此最为广泛的语言，编译与上传代码的环境为Arduino IDE软件。软件方面的设计主要分为两大部分，第一为输入元件的程序编写，即DHT11传感器方面的内容；第二部分为输出元件的程序编写，主要为OLED的显示设计、LED的闪烁和蜂鸣器的鸣叫。

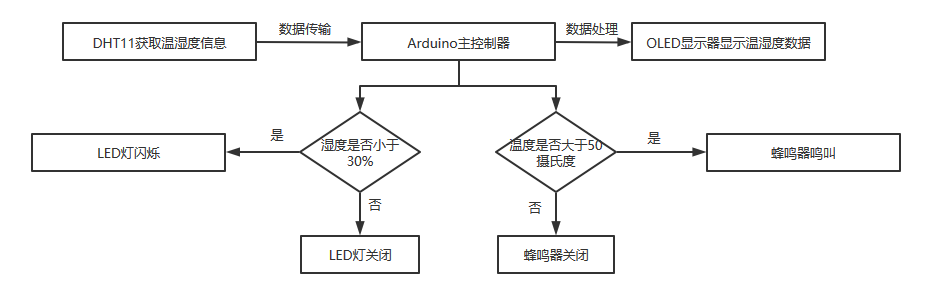


图3.1程序设计流程图

## 3.1输入元件程序设计

本设计的输入元件即传感器选择的是DHT11温湿度传感器，在程序设计中，本设计引入其库函数“DHT.h”，通过初始化传感器和调用获取温湿度的函数可以直接得出数据处理过的温湿度变量，程序中再根据这些变量去控制输出元件。

部分程序讲解：

#include <DHT.h>

DHT11传感器库引入；

#define DHTPIN 8

此句话定义了DHT11传感器信号线所接的为Arduino的数字引脚8；

#define DHTTYPE DHT11

进行宏定义；

DHT dht(DHTPIN, DHTTYPE);

定义一个对象，并且进行引脚输入和类型匹配；

dht.begin();

温湿度传感器初始化；

void get\_temp\_humi()             //获取温湿度数据子函数

{

  HH = int(dht.readHumidity());      //获得湿度

  TT = int(dht.readTemperature());   //获得温度

  Serial.print("Temp: ");           //串口显示温湿度信息

  Serial.println(TT);

  Serial.print("Humi: ");

  Serial.println(HH);

}

上面定义了一个子函数，其中HH变量用于存储湿度数据，TT变量用于存储温度数据，并且开启了串口，使用者可以打开Arduino的串口监视器查看当前的温湿度数据。该子函数将在循环函数loop里面调用。

## 3.2输出元件程序设计

输出元件有3个，分别为OLED显示器、LED、蜂鸣器。

### 3.2.1 OLED显示器

由上述电路图可以知道，OLED的数据线SDA接到A4，时钟线SCK接到A5，因为Arduino的IIC接口正是A4\A5，当然，A4\A5接口也可以作为普通数字IO口来用，在这里是属于引脚复用。

部分程序讲解：

#include "U8glib.h"

OLED显示器库函数引入；

U8GLIB\_SH1106\_128X64 u8g(U8G\_I2C\_OPT\_NONE);

设置通讯方式为IIC通讯；

void dis\_play()    //OLED数据显示子函数

{

  u8g.firstPage(); //  OLED首页

  do

  {

    char buffer1[1]; //定义字符buffer1

        //  将int转为char显示出来

    itoa(TT, buffer1, 10);        //10：10进制显示

    u8g.setFont(u8g\_font\_gdr10r); //设置字体

    u8g.drawStr(25, 26, "Temp:"); //第26行25列显示“Temp：”

    u8g.drawStr(70, 26, buffer1); //第26行70列显示温度数据

    char buffer2[1];  //定义字符buffer2

    //  将int转为char显示出来

    itoa(HH, buffer2, 10);        //10：10进制显示

    u8g.setFont(u8g\_font\_gdr10r); //设置字体

    u8g.drawStr(25, 60, "Humi:"); //第60行25列显示“Humi：”

    u8g.drawStr(70, 60, buffer2); //第60行70列显示湿度数据

  } while (u8g.nextPage());       //执行配置

}

上述代码为OLED显示文本和数据的子函数，在第26行25列显示温度数据，在第26行70列显示湿度数据，并且该子函数将在循环函数loop里面调用。

### 3.2.2 LED灯

LED灯的功能在于当湿度小于某个阈值的时候闪烁，这里设置的值为30，湿度过小，代表环境干燥，提示用户适当进行加湿。

部分代码讲解：

int led = 4;

定义LED引脚4；

pinMode(led, OUTPUT);

设置LED引脚为输出模式；

if (HH < 30)

  {

    digitalWrite(led, HIGH);

    delay(50);

    digitalWrite(led, LOW);

    delay(50);

  }

当湿度小于30的时候，LED灯将进行间隔50ms的闪烁，提醒用户当前环境的湿度情况。

### 3.2.3 无源蜂鸣器

无源蜂鸣器在该设计的作用是当环境的温度大于50度的时候，产生鸣叫，从而提醒使用者环境温度超标，可能有火情出现。

部分代码讲解：

int buzzer = 3;

定义蜂鸣器所接引脚为3；

pinMode(buzzer, OUTPUT);

设置蜂鸣器所接引脚为输出模式；

  if (TT > 50)

    analogWrite(buzzer, 100);

  else

    analogWrite(buzzer, 0);

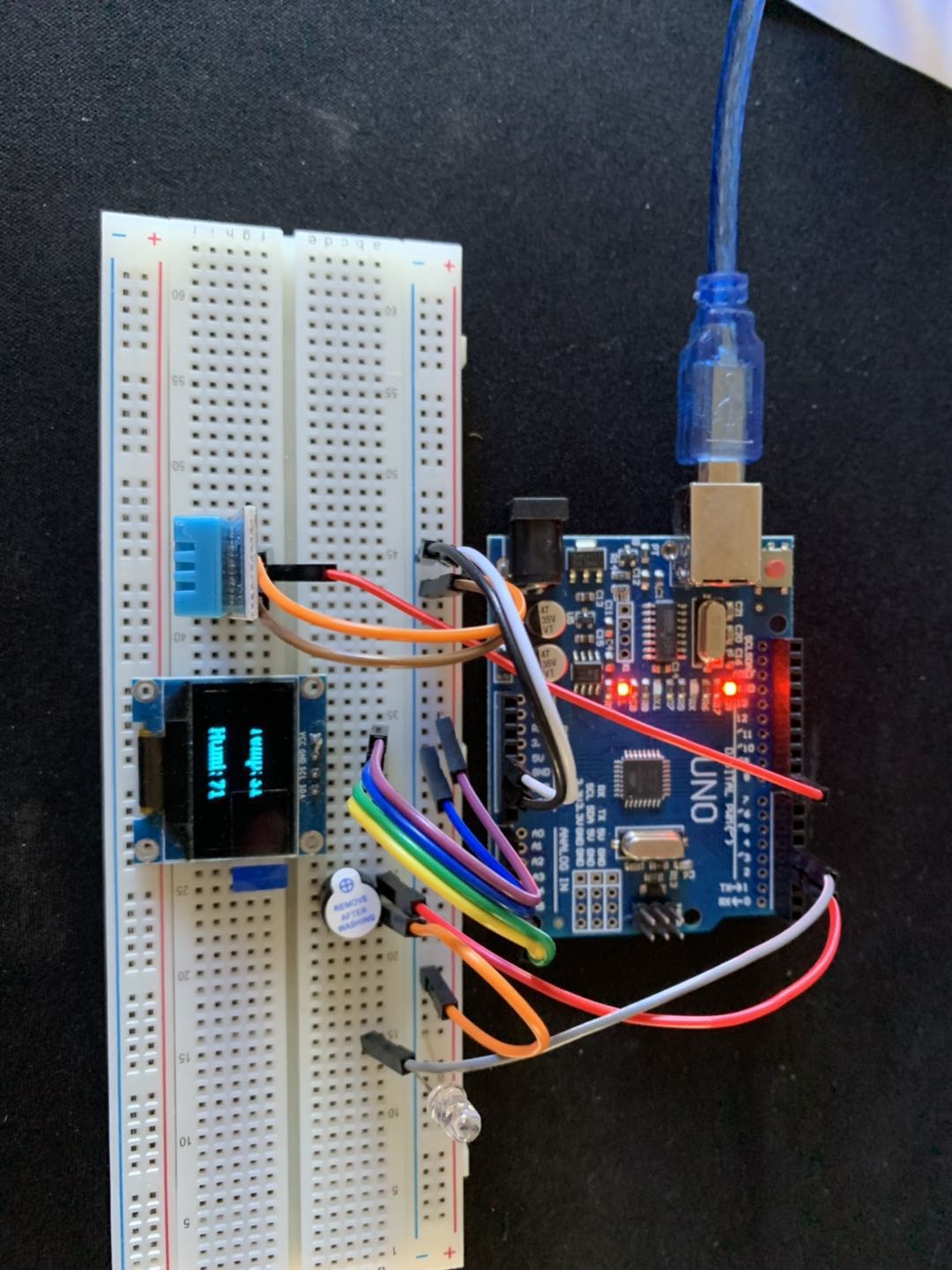
当温度数值大于50度的时候，蜂鸣器响起，提醒使用者当前环境的温度值超标，可能出现火情，当温度值小于50的时候，蜂鸣器不鸣叫。

# 第四章 系统调试

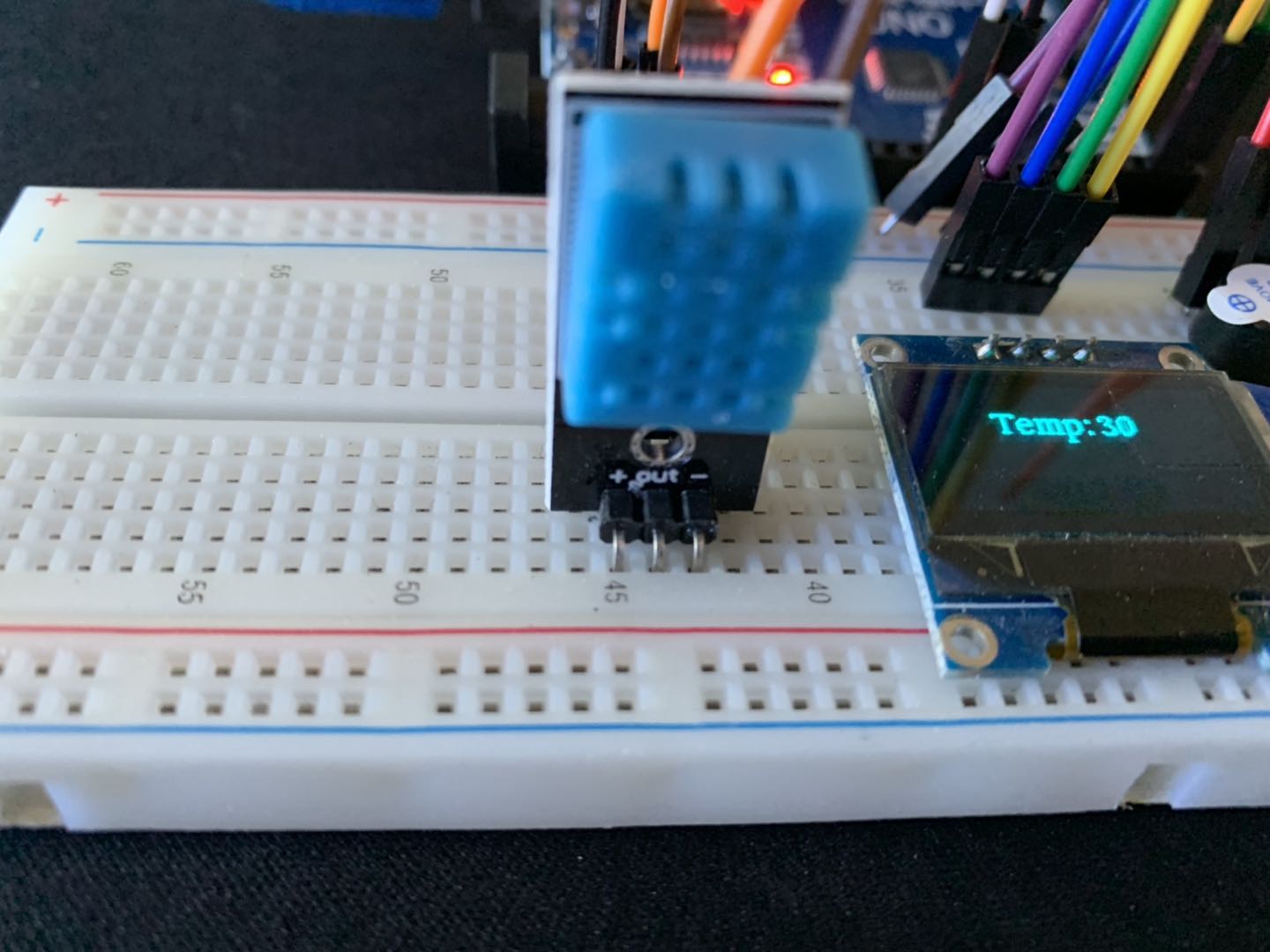
系统调试分为硬件调试和软件调试。硬件调试中主要为确保接线的正确性，软件部分主要为检测各功能是否满足预设的目标，调试发现，给Arduino上电后，OLED显示器立即显示由DHT11温湿度传感器获得的数据，可以显示出当前环境的温湿度情况，与实际温湿度情况大体一致，并且拿到干燥地方，led灯会闪烁，当环境温度超过50摄氏度的时候，无源蜂鸣器会鸣叫，调试各功能顺利，基本满足预设要求。

# 第五章 实物展示

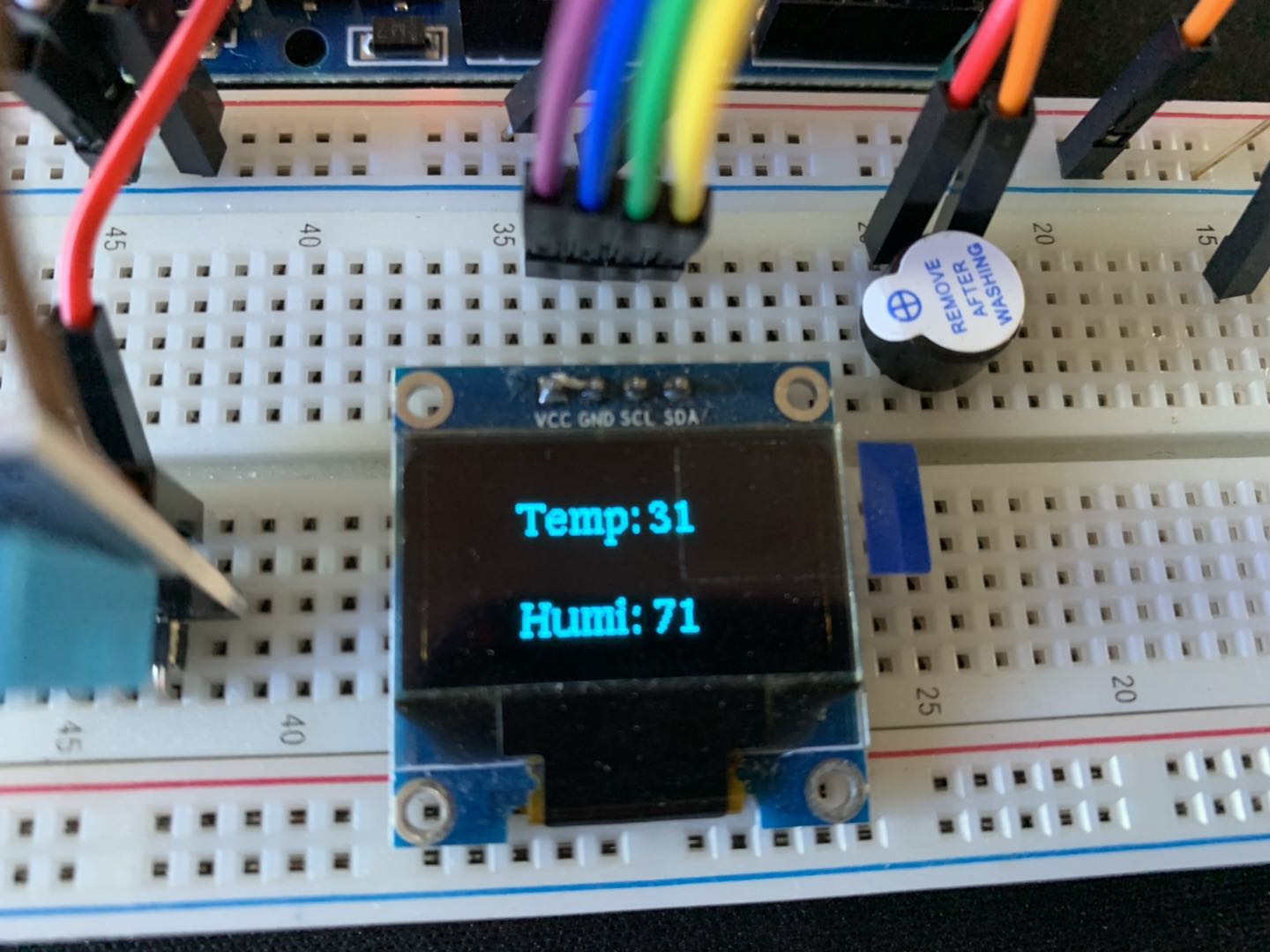
1）整体实物图



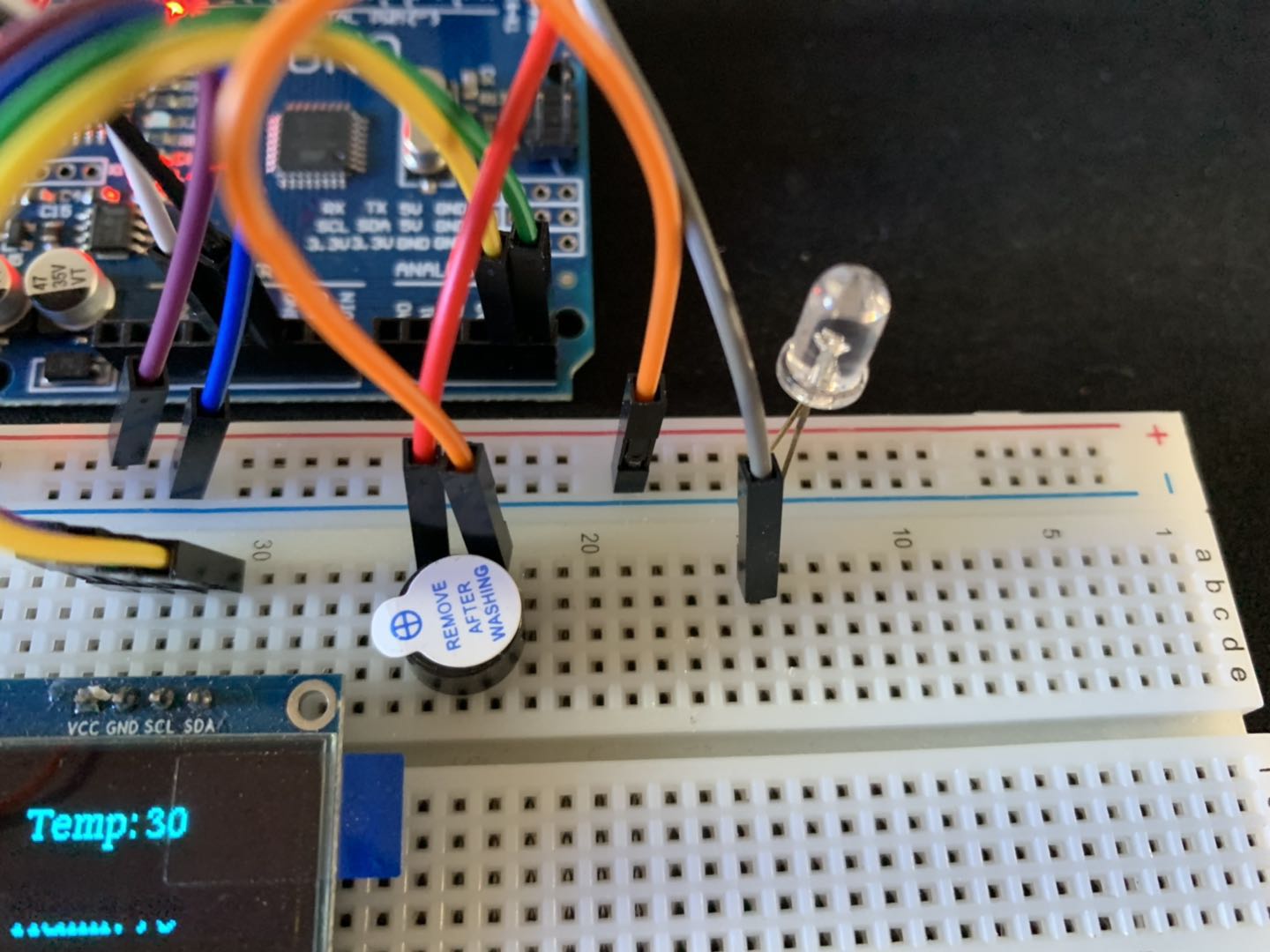
2）DHT11展示



1. OLED显示器展示



4） 蜂鸣器和LED展示



# 第六章 总结

这次实训本人设计的家庭安全监测系统，可以有效地监测家庭温度和湿度的情况，并且具有预警功能，达到很好的提醒作用。随着人们生活水平的提高和安全防范意识的日渐加强，智能防火防盗系统也在不断升级，防火防盗设备也在不断更新，希望此设计能给家庭安全设备提供一定的思路。

# 附件：程序清单

#include <DHT.h>                            //引入<DHT.h>头文件

#include "U8glib.h"                         //引入Oled头文件

U8GLIB\_SH1106\_128X64 u8g(U8G\_I2C\_OPT\_NONE); //设置通讯方式为IIC通讯

#define DHTPIN 8                            //湿度传感器OUT接8号口

#define DHTTYPE DHT11                       //定义DHT11传感器

DHT dht(DHTPIN, DHTTYPE);                   //定义湿度传感器OUT与8号口连接并且读取温湿度传感器的数值

int buzzer = 3;                             //定义蜂鸣器引脚3

int led = 4;                                //定义led引脚4

int HH;                                     //定义HH变量，用于储存湿度数据

int TT;                                     //定义TT变量，用于储存温度数据

void setup()

{

  pinMode(buzzer, OUTPUT);                 //设置蜂鸣器引脚为输出模式

  pinMode(led, OUTPUT);                    //设置led引脚为输出模式

  Serial.begin(9600); //串口波特率

  dht.begin();        //温湿度传感器初始化

}

void loop()

{

  get\_temp\_humi();           //获得温湿度子函数

  dis\_play();                //oled显示数据子函数

  if (TT > 50)                //如果温度大于50°，蜂鸣器响

    analogWrite(buzzer, 100);

  else                        //否则，蜂鸣器不响

    analogWrite(buzzer, 0);

  if (HH < 30)               //如果湿度小于30，led闪烁

  {

    digitalWrite(led, HIGH);

    delay(50);

    digitalWrite(led, LOW);

    delay(50);

  }

  else                      //否则，led熄灭

    digitalWrite(led, LOW);

}

void get\_temp\_humi()             //获取温湿度数据子函数

{

  HH = int(dht.readHumidity());      //获得湿度

  TT = int(dht.readTemperature());   //获得温度

  Serial.print("Temp: ");           //串口显示温湿度信息

  Serial.println(TT);

  Serial.print("Humi: ");

  Serial.println(HH);

}

void dis\_play()    //OLED数据显示子函数

{

  u8g.firstPage(); //  OLED首页

  do

  {

    char buffer1[1]; //定义字符buffer1

        //  将int转为char显示出来

    itoa(TT, buffer1, 10);        //10：10进制显示

    u8g.setFont(u8g\_font\_gdr10r); //设置字体

    u8g.drawStr(25, 26, "Temp:"); //第26行25列显示“Temp：”

    u8g.drawStr(70, 26, buffer1); //第26行70列显示温度数据

    char buffer2[1];  //定义字符buffer2

    //  将int转为char显示出来

    itoa(HH, buffer2, 10);        //10：10进制显示

    u8g.setFont(u8g\_font\_gdr10r); //设置字体

    u8g.drawStr(25, 60, "Humi:"); //第60行25列显示“Humi：”

    u8g.drawStr(70, 60, buffer2); //第60行70列显示湿度数据

  } while (u8g.nextPage());       //执行配置

}