

高级程序设计语言智能硬件项目 说明文档

题目： 带有温湿度计与 CO 报警器功能的
Arduino 光立方设计

设计者： 杜洪超

学号： 201605130116

一、摘要

光立方既普通 LED 灯焊接而成的立方体，通过程序设计控制每一个 LED 灯的亮灭以实现三维的静态或动态图案，极具观赏性。常见的光立方有四阶或八阶，本次设计采用了较为简单的四阶光立方，既 $4*4*4$ 共 64 个雾状 LED 灯，搭配各种传感器，添加了温湿度计与 CO 报警器的功能，使用 Arduino 开发板作为智能硬件，以 C 语言实现，最终完成了本次设计。

二、设计目的与要求

2.1 设计目的

光立方本身虽具有较强的观赏性，但受硬件限制且一般被设定为通电之后无限循环，较为不便且没有很大的实用价值。本设计的目的是为了通过设计让光立方成为家居装饰，同时具有一定的实用价值。在此基础上完成了基本的设计要求。

2.2 设计要求

以基于 Arduino 的光立方为主体，通过温湿度传感器使其具备温湿度计的功能，通过 CO 浓度传感器与蜂鸣器结合使其具备 CO 报警

器功能，添加红外遥控模块以实现遥控设计与发送各种指令，通过 Wifi 模块将各种数据传输至服务器端。

三、硬件方案设计

3.1 简介

经过简单分析之后发现，单独的 Arduino UNO 单片机不足以实现所要求，因此本设计选择了 Arduino UNO 标准板与较为轻便的 Arduino Nano 各一以供开发，其中后者以前者的扩展版的形式存在，主要用来控制光立方与收发红外遥控模块的指令，前者控制各种传感器并与服务器进行交互。

CO 浓度传感器采用 MQ2 烟雾传感器，该传感器能有效监测各种易燃气体及烟雾浓度，本设计中使用 MQ2 的监测 CO 浓度功能

采用 ESP8266Wifi 模块实现无线传输

采用 DHT11 温湿度传感器收集温湿度信息

采用无源蜂鸣器

供电使用装有六节五号电池的电池盒，为并联的单片机供电

3.2 单片机资源分配情况

(1)光立方

四阶光立方分为四层，每层共 16 个 LED 灯，根据光立方层共阴，列共阳的基本设计理念，共需要 $16 \times 4 = 20$ 个输出接口，显然如果使用

单一 Arduino 控制，将占用全部可用输出口而无法交互，故使 Nano 中 D2 到 D13 12 个数字 IO 口,A0 到 A3 4 个模拟 IO 口共 16 个输出口来控制光立方阳极。由 UNO 8~11 4 个数字 IO 控制光立方阴极，共 20 个引脚。

(2)红外遥控模块

Nano 中 A4 模拟 IO 作为数据输入，接 5V 与 GND 供电

(3)温湿度传感器模块

UNO 模拟 IO A0 作为数据输入，接 5V 与 GND 供电

(4)无源蜂鸣器

UNO 模拟 IO A1 作为数据输入，接 5V 与 GND 供电

(5)MQ2 烟雾传感器

UNO 模拟 IO A1 作为数据输入，接 5V 与 GND 供电

(6)ESP8266Wif 模块

UNO 数字 IO 4, 5 作为模拟软串口进行数据传输，接两个独立 3.3V 与 GND 供电

(7)UNO 与 Nano 交互

连接两单片机串口 TX，RX 既数字 IO 0，1 实现通讯

(8)UNO，Nano 供电

将 VIN 连接焊接好的电池盒正极插针，GND 连接负极插针

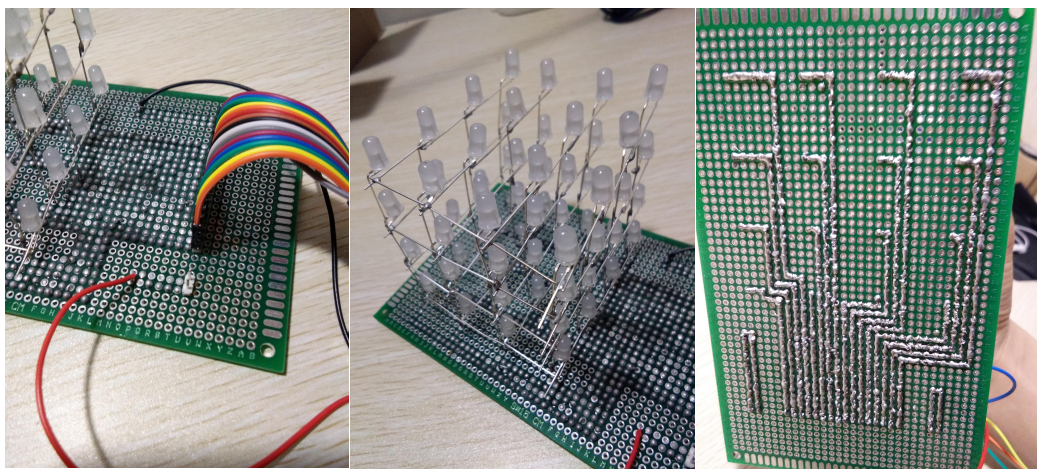
3.3 光立方与电池盒焊接

光立方采用 64 个 5mm 雾状蓝 LED，通过一定处理后焊接出 4 层，

每层 16 个 LED,再将每一层焊接起来,将预先保留出的 16 个阳极引脚插进 15cm*9cm 洞洞板,设计并焊接出底部的电路,再将 16 个插针焊接在电路板上以供单片机连接。

将电池盒正负极引线焊接在电路板上,通过焊电路与插针实现并联。

实物图:



四、软件设计

(说明: Arduino 中多文件编程仅能保存.cpp 文件,故各个传感器的函数文件以.cpp 格式保存,但编程语言仍为 C 语言)

1. Arduino UNO

(1)源文件构成:

①CUBE4_1.0.ino

②DHT_11.h DHT_11.cpp

③Buzzer.h Buzzer.cpp

④MQ2.h MQ2.cpp

⑤ESP8266.h ESP8266.cpp

(2)代码作用及逻辑顺序

①(主程序):

调用函数进行各个模块的初始化;

实时读取 CO 浓度,并在超过影响人体健康最低浓度时做出相应的反应;

接受经 Nano 转发的遥控指令并做出反应;

接受控制光立方的指令并做出反应;

②(温湿度传感器模块):

读取传感器数据并处理;

将数据上传至服务器;

③(无源蜂鸣器模块):

初始化;

发出蜂鸣声;

向服务器发送报警指令;

④(MQ2 烟雾传感器模块):

重置模块,初始化数据;

监测并计算 CO 浓度;

⑤(Wifi 模块):

初始化模块并连接 WIFI;

连接服务器;

2.Arduino Nano

源文件：CUBE4.ino

代码作用及逻辑顺序：

初始化光立方及红外遥控模块；

接受来自红外模块的指令并将其传往开发板 UNO；

将存在数组中的光立方指令取出一部分存到缓冲区，并依次做出相应处理；

红外遥控目前只对三个按键进行了编码，实现了打开关闭光立方，测量温湿度与重置 CO 传感器功能

(具体实现详见源码)

五、测试

1.摁下 Arduino UNO 的复位按钮，光立方进入调试模式，开始加载各个模块并连接服务器。

2.初始化完毕，光立方变为一般模式，服务器显示收到了连接。

3.摁下遥控器的 OK 按钮，光立方关闭，再摁一次，光立方打开。

4.关闭光立方，摁下 1 按键，光立方打开并进入调试模式，服务器上开始接受到处理好的温湿度信息。

5.关闭光立方，摁下 2 按键，MQ2 模块重置。

6.关闭光立方，使用打火机模拟出小范围较高浓度 CO，光立方进入调试模式，蜂鸣器报警，服务器接受到报警指令并发出报警信息。

待 CO 散去，蜂鸣器停止报警。

7.其它设计与构想因为硬件与时间限制暂时不进行实现与测试，(如控制光立方亮度，遥控控制光立方动画模式等)可供二次开发，测试结束。

演示视频网址: http://v.youku.com/v_show/id_XMjc1OTc3MjgzMg==.html

六、心得体会

本次设计从确定方向到最终完成历时二十余天，中间包括网上采购开发板与零器件环节。查阅了大量光立方设计资料，经过仔细研究设计出改造方案，在开始的焊接过程中电烙铁的学习与使用极大的提高了自己的动手能力，并考验了耐性。通过查询资料对 Arduino 的开发有了较深入的了解，虽然因为硬件限制导致制作中出现了很多问题，甚至一度中断不能继续进行，不过最终在实验了多种方法后终于找到了相应的解决办法并最终实现了预期功能。同时对 Wifi 模块与服务器的连接中对网络编程有了一定的了解。

因为本设计最初就是从实用角度出发的，所以在提高自己能力的同时，设计本身也是一件设计巧妙的装饰品与实用的报警器，同时还具备了可编程与可二次开发的特性。

综上，这次的基于 Arduino 的光立方设计让我收获良多，是一次不可多得的学习经历。

七、参考文献与主要资料

1. Massimo Banzi 著，于欣龙、郭浩赞译. 《爱上 Arduino》
人民邮电出版社
2. <https://www.arduino.cc/en/Reference/HomePage>
Arduino Reference
3. <http://zhongbest.com/2017/01/06/%E6%95%99%E7%A8%8B%E7%A4%BA%E4%BE%8B%E7%B4%A2%E5%BC%95%E9%A1%B5/>
传感器模块示例教程
4. <http://www.geek-workshop.com/thread-4989-1-1.html>
四阶光立方直驱教程