**

**项目报告**

**课程名称**  IT技术创新项目实践

**项目名称** 改造普通灯变成WiFi控制灯

**班级与班级代码** 19计算机科学与技术（实验班）

**实验室名称（或课室）** 实验楼802

**专 业 计算机科学与技术**

**任课教师 肖银皓**

**学 号：** 19251106105

**姓 名：** 伍嘉明

**实验日期：** 2022 年 4 月 20 日

广东财经大学教务处 制

**姓名 伍嘉明 项目报告成绩**

**评语：**

**指导教师（签名）**

**年 月 日**

**改造普通灯变成WiFi控制灯**

1. **项目目的**
   * + 1. 改造普通usb灯为WiFi控制灯，通过手机浏览器传输指令，实现对普通灯的开关控制。
2. **项目环境**
   * + 1. NodeMCU（ESP8266）开发板一块
       2. 高低电平继电器模块一个
       3. 普通USB灯一个
       4. 11V转3.3V、5V变压器
       5. 手机一台（可开移动热点和浏览网页）
       6. 移动电源
       7. 杜邦线若干、USB线一根、USB转DC电源线一根
       8. Arduino IDE 1.8.19
       9. Visual Studio Code
3. **项目原理**
   * + 1. ESP8266是一款物联网WiFi芯片，基于ESP8266可以开发物联网串口WiFi模块，可将物理设备连接到WiFi无线网络上，进行互联网或局域网通信，实现联网功能。本次实现采用ESP8266的STA（Station）模式，即ESP8266开发板作为网络服务器端，与同一局域网内的客户端，如手机，进行HTTP协议通信。
       2. 用户在客户端开启移动热点，ESP8266开发板连接该WiFi热点，就会生成一个IP地址，并通过Arduino IDE的串口监视器输出。考虑到便捷性，不用每次都控制WiFi灯都查看串口监视器，可以设置固定IP地址，只要该IP地址与手机的IP地址在同一局域网内，即可进行通信，从而实现一个较为完整的系统。
       3. ESP8266的数字输出作为高低电平继电器模块的数字输入。在本次实验中，任意边沿来临时高低电平继电器模块输出接通，任意边沿再次来临则高低电平继电器模块输出断开，以实现开关的效果。
       4. 11V转3.3V、5V变压器的作用是通过连接一个电源，可以同时给多个不同电压需求的设备供电。
4. **项目步骤与结果**
   * + 1. 在Arduino IDE编写控制WiFi板的代码。
   1. 总体程序设计

程序流程如下图1，控制WiFi板全部代码见附录代码连接。代码主要由setup()函数和loop()函数组成。setup()函数的内容用于初始化设置，如设置输出引脚和连接WiFi，只运行一次；loop()函数检查是否有客户端设备通过网络向ESP8266网络服务器发送请求，一直循环运行。

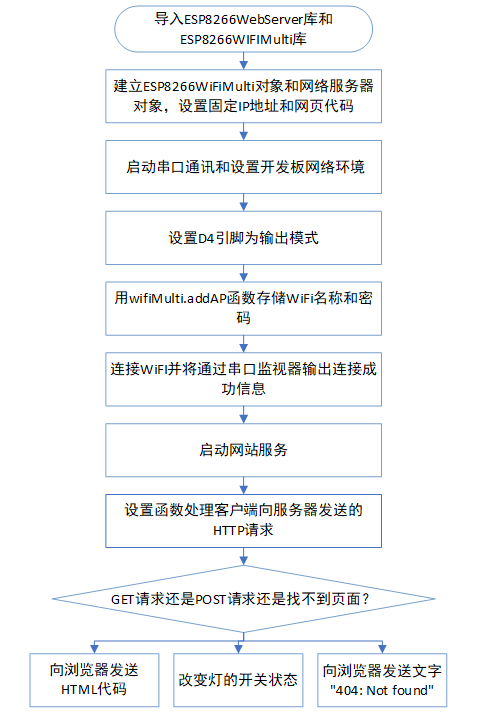


图 1

* 1. 设置WiFi板的固定IP地址和网络环境。

调用IPAddress定义四个变量:local\_IP、gateway、subnet和dns,分别为本地ip、网关ip、子网掩码和dns的ip，如图2。然后调用WIFI.config函数完成IP配置，用函数的返回结果做判断，如果设置成功返回ture，设置失败，返回false，如图3。

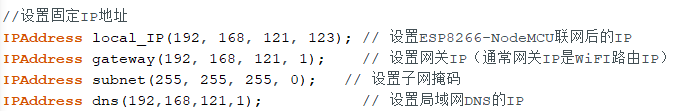


图 2

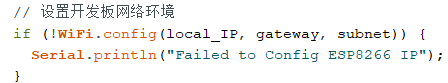


图 3

* 1. ESP8266网络服务器的主要工作代码

当ESP8266开发板利用begin函数开启网络服务以后，每当有客户端向服务器发送HTTP请求时，利用on函数设置HTTP请求回调函数。利用onNotFound函数处理页面找不到的情况，代码如下图4。

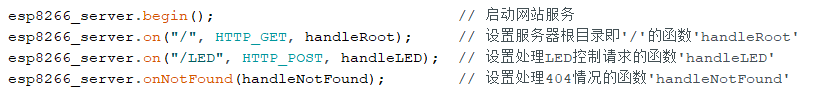


图 4

* 1. 处理请求开灯或关灯的http请求

每次ESP8266网络服务器接收到来自浏览器的开灯或关灯请求时，将当前D4引脚的电平输出状态改成相反的输出状态，核心代码为：digitalWrite(D4,!digitalRead(D4));

即如果原来是高电平则改成低电平输出，原来低电平则改成高电平输出，从而控制灯的开关状态。

* + - 1. 在Visual Studio Code编写网页代码。

网页界面如图5，网页详细代码见附录代码链接。核心代码为：

<form action="/LED" method="POST\">

LampController

<input type="submit" value="turn on/off">

</form>

用户在客户端点下灯开关按钮，提交表单，即向ESP8266网络服务器发送一个POST 请求。

接下来是压缩html代码，将html代码以字符串的形式写入控制WiFi板的代码。压缩html代码时，在Visual Studio Code将双引号用单引号替换，否则html里的双引号会与两端表示字符串的双引号结合，提前结束html代码，引发程序错误。删除空格和换行。

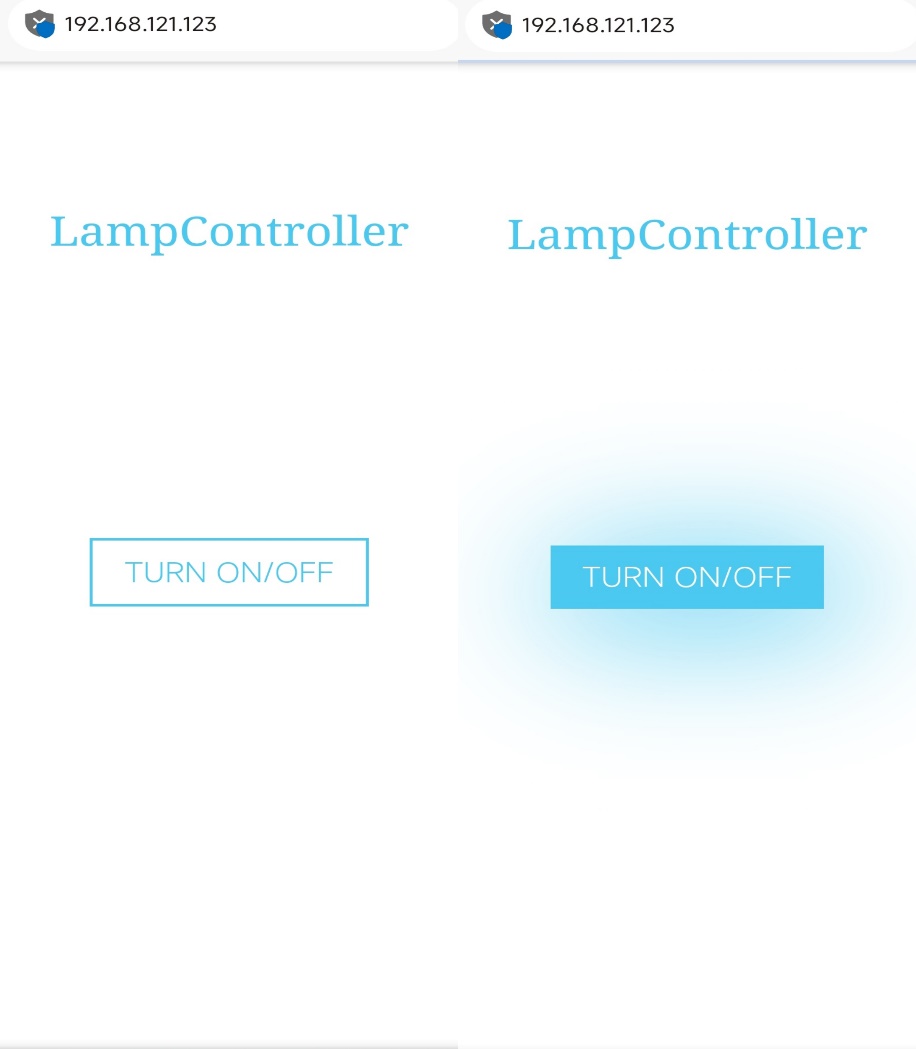


图 5 左：点击按钮前；右点击按钮后

* + - 1. 烧录。

烧录时，要注意保持WiFi板的端口连接正常。

* + - 1. 接线。总体接线如图6。

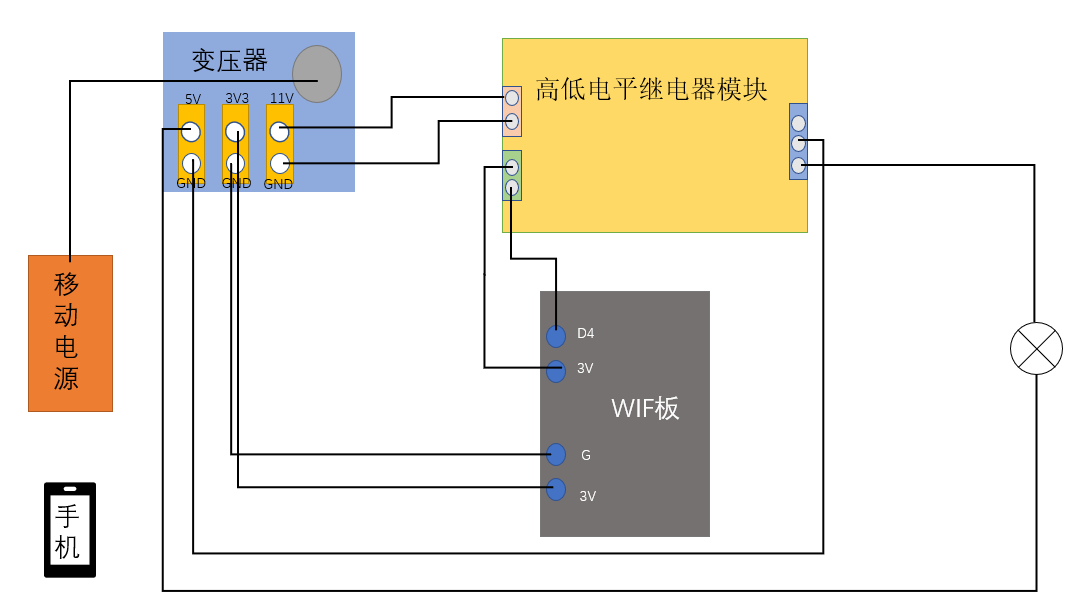


图 6

* 1. 给变压器、继电器接通电源和选择模式。

将变压器连接上移动电源，用杜邦线将变压器11V引脚和继电器电源正极输入连接，GND引脚和电源负极输入连接。继电器通电后，长按“M”按键来解锁设置，选择延时功能模式F--1，再选择开关功能模式0101，任意边沿自锁，即任意边沿来临时输出接通，任意边沿再次来临则输出断开，如图7。

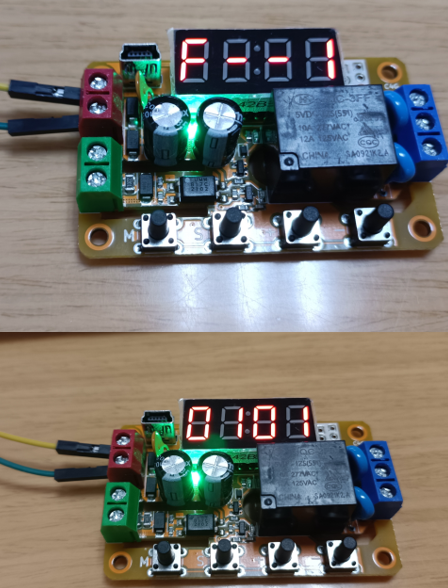


图 7

* 1. 连接WiFi板和变压器、电器。

第一步，杜邦线将WiFi板的G引脚和变压器的GND引脚、WiFi板的3V引脚和变压器的3V引脚连接。第二步，杜邦线连接高低电平继电器模块的数字输入正极和WiFi板3V引脚、数字输入负极和D4引脚，如图8。

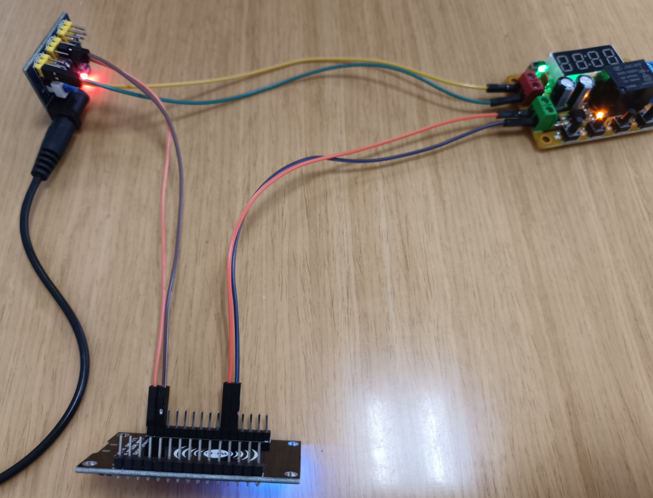


图 8

* 1. 连接变压器、继电器和USB灯。

第一步，将USB灯的插头端剪开，并去除绝缘皮，使得电源+5V导线（红色）裸露出来，地线（黑色）裸露出来。第二步，剪开两根杜邦线，并去除结缘皮，如图9。第三步，将杜邦线和导线分别绞合在一起，拧紧，如图10。将USB灯的正极（红色导线）连接的杜邦线接入变压器5V引脚，连接地线（黑色）的杜邦线接入继电器的常开触点。第四步，用杜邦线将5V引脚下的GND引脚与公共触点连接。至此，整个完成装置，效果图如图11所示。演示视频见附录视频链接。

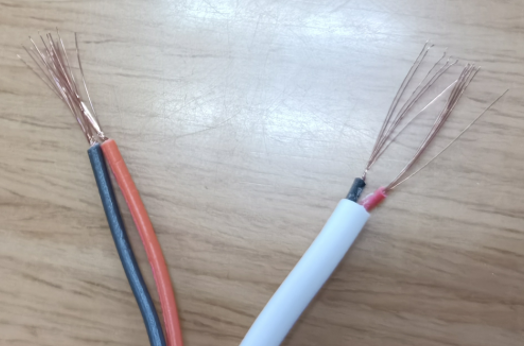


图 9

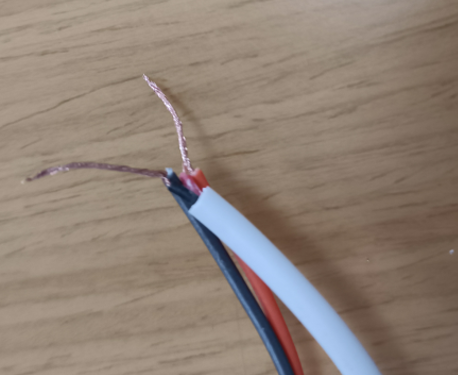


图 10

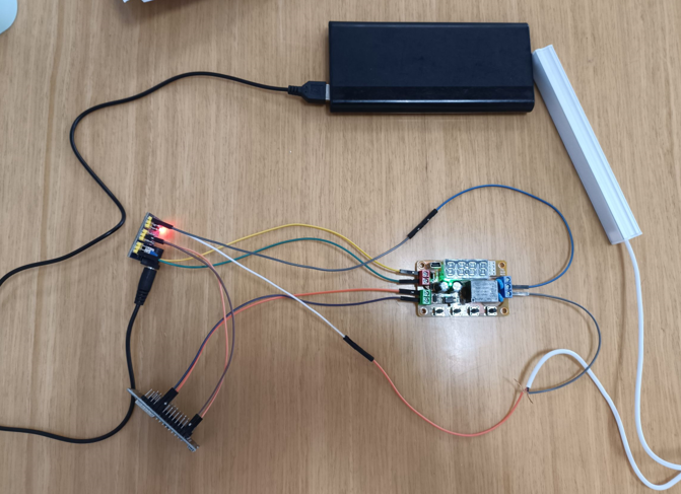


图 11

1. **项目总结**
   * + 1. 实验结果评价。

本次实验填补了我许多知识的空白，实验结果我认为该系统是比较完善的，虽然仍存在一定的瑕疵。WiFi板使用固定IP地址带来一定的便捷，但也存在一定的缺陷。如果是用手机流量开热点，由于手机IP地址是随着手机信号连接的基站的改变而改变得，有时候手机的IP地址和WiFi板的IP地址不在同一个局域网，使得两者之间无法通信，如果使用路由器连接可以解决这个缺点，WiFi板的IP地址要设置在路由器分配IP地址的范围内。该系统可以进一步完善，与云平台配合，实现远程控制。

* + - 1. 实验中遇到的主要问题的分析与处理

问题1：ESP8266 WiFi板无法连接手机热点。

分析：手机热点频段有2.4GHz和5GHz，ESP8266不支持5GHz频段，如果手机热点开启时选择了5GHz频段，会导致ESP8266 WiFi板无法连接手机热点。

处理：手机热点开启时选择2.4GHz频段即可解决。

问题2：如果使用手机热点连接的ESP8266，每次都要通过串口监视器查看WiFi板的IP地址，能否实现一个较为独立的系统，不用每次控制灯都要查看IP地址？

分析：可以通过设置WiFi板连接网络后的IP地址，只有手机与该WiFi板处于同一个局域网即可互相通信。

处理：通过函数WIFI.config设置WiFi板的网络环境。

**附录**

1. 演示视频链接：<https://b23.tv/BA0CSyV?share_medium=android&share_source=qq&bbid=XX4B4504E7F2F8068BF467926084DEB7A888F&ts=1654171605571>

1. 代码链接：

<https://github.com/Avivi210/ESP8266-WiFi-LED.git>