# ESP8266 WiFi 控制继电器的实现

系统描述

使用ESP8266模块以STA模式连接到wifi，通过电脑搭建的TCP Server服务向其发出指令，ESP8266接收到指令之后，向继电器发出通断指令，实现对12V灯泡开关的控制。

硬件准备

ESP8266模块

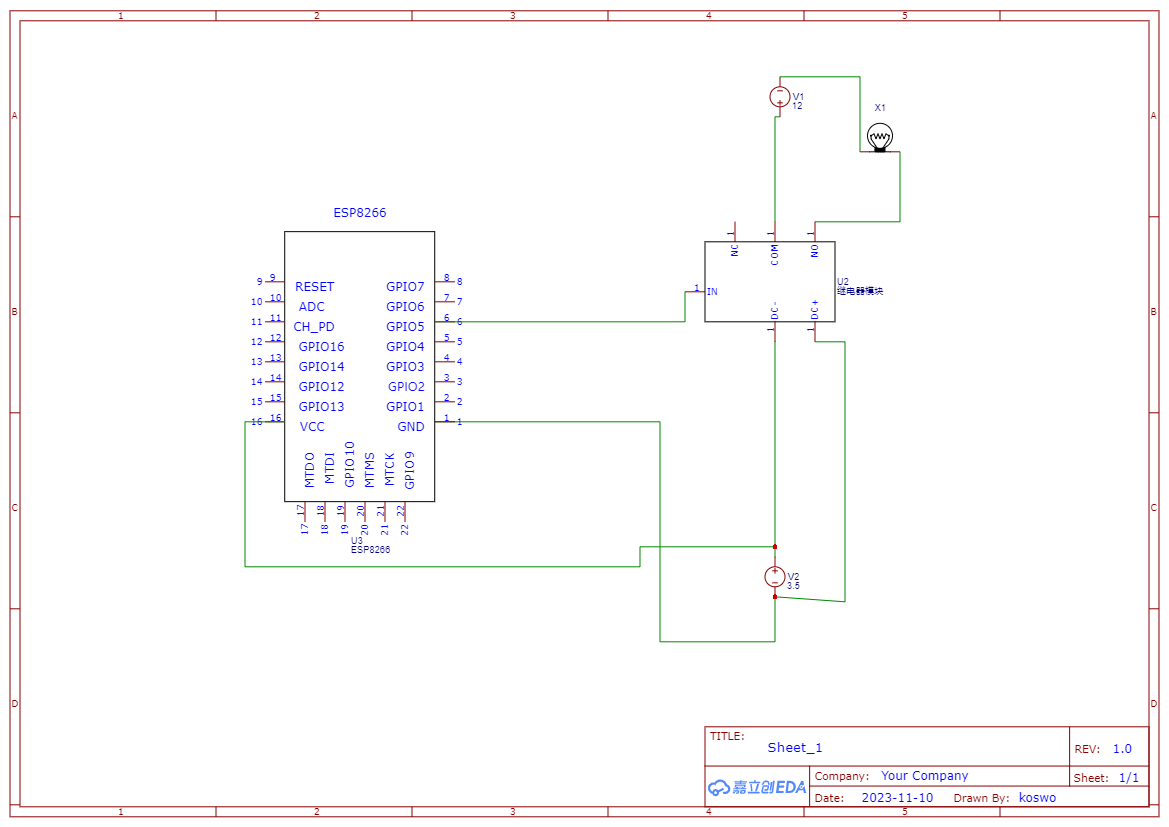
继电器模块

Arduino IDE

12V灯泡

继电器电源

接线图如下



软件准备

安装Arduino IDE并配置ESP8266开发板支持库

代码框架



代码分析

//设置网络

#include <ESP8266WiFi.h>//加载库文件

const char\* ssid     = "Group\_5";

const char\* password = "12345678";//设置AP的SSID及密码

const uint16\_t port = 2222;

const char \* host = "192.168.0.102";//设置TCP Server的IP和端口号

WiFiClient client; //创建一个tcpclient连接

//setup函数部分

void setup() {

Serial.begin(115200);//设置波特率

 pinMode(5, OUTPUT);//配置输出端口

WiFi.mode(WIFI\_STA);//设置WIFI模式

WiFi.begin(ssid, password);

Serial.print("\r\nConnecting to ");

Serial.print(ssid); Serial.println(" ...");

int i=0;

while (WiFi.status() !=WL\_CONNECTED)

{

  delay(1000);

  Serial.print("Waiting for ");

  Serial.print(i++);Serial.println("s...");

}

  Serial.println("");

  Serial.println("WiFi connected!");

  Serial.print("IP address: ");

  Serial.println(WiFi.localIP());

}//通过串口发送WIFI连接状态信息

//loop函数部分

void loop() {

Serial.println("Connceting to TCP Server :");

Serial.println(host);

if (!client.connect(host,port)){

  Serial.println("Failed");

  Serial.println("Reconnect in 5 seconds");

  delay(5000);

  return;

}//通过串口发送TCP Server连接信息

Serial.println("Send The String to TCP Server :\"I am Esp8266\"");

client.println(String("I am Esp8266"));

delay(3000);

Serial.println("");

Serial.println("wait 5 sec...");//向TCP Server发送信息

while (client.available())

 {

  char val =client.read();

  if(val=='0')

  {

    digitalWrite(5,HIGH);//如果收到0，则GPIO5输出高电平

  }

  if(val=='1')

  {

    digitalWrite(5,LOW);//如果收到1，则GPIO5输出低电平

  }

 }

}

在程序的开头，为开发板载入相对应的库文件，并且对连接的AP的SSID和AP以及TCP Server的参数进行配置。在setup函数部分，配置单片机的波特率以及在串口向外输出运行状态。在loop函数部分，首先设置单片机向串口发送TCP Server连接信息以及向TCP Sever发送的数据，然后单片机根据收到的指令，对GPIO5的高低电平输出进行配置。

进行代码烧录

在Arduino IDE中，选择对应的COM端口，手动指定开发板为Generic ESP8266 Module，进行验证，检查代码中是否存在错误，并进行修改完善，修改完成后进行上传写入。

首先将ESP8266配置在写入模式，然后在Arduino IDE中进行写入操作，写入完成后，开关拨到运行模式，并按压reset按键重启ESP8266单片机，在Arduino IDE的串口监视器中检查ESP8266单片机的输出，确定ESP32是否按照预期设置正常运行，如有故障则进行相应排故。

在netassist软件中，配置以TCP Server模式运行，检查ESP8266进行连接，在在netassist软件中向ESP8266单片机发送指令，进而控制灯泡的通断。

通过传感器进行控制

对代码进行以下修改

在第一段代码中添加

int sensorvalue = 0;

int outvalue = 0;//定义传感器的值

在loop部分中添加

sensorvalue= analogRead(A0);

outvalue=map(sensorvalue,0,1024,0,330);

Serial.println(outvalue);// //设定传感器数值，在串口进行输出

  delay(3000);

  if(outvalue <= 50) {

    digitalWrite(5,LOW);

  }//若传感器输出数值小于50，则GPIO5输出低电平

  if(outvalue > 50) {

    digitalWrite(5,HIGH);

  }//若传感器输出数值大于50，则GPIO5输出高电平

问题记录

ESP8266库无法安装

使用第三方整合包对ESP8266库进行手动离线安装。

Arduino出现闪退

安装最新版本的Arduino IDE。

控制指令发出后，灯泡响应较慢

减少delay（）语句中设置的时间长度。

netassist软件中TCPServer模式下无法收发数据

检查电脑的网络防火墙设置，确定所选择的端口可以被使用，能够在局域网中被访问。

代码中出现重复定义冲突

在故障信息窗口中点击故障信息定位到对应的代码行，确定正确的配置。

Arduino IDE的串口监视器中出现乱码信息

设定波特率为ESP8266所设定的波特率。

继电器无响应

检查线路，将继电器的接线连接至正确的GPIO接口。