

# 智能插座 DIY 项目总结报告

## 一，电装

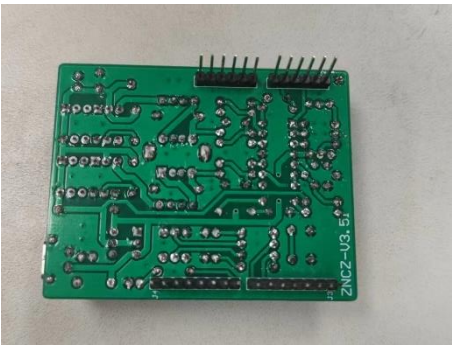
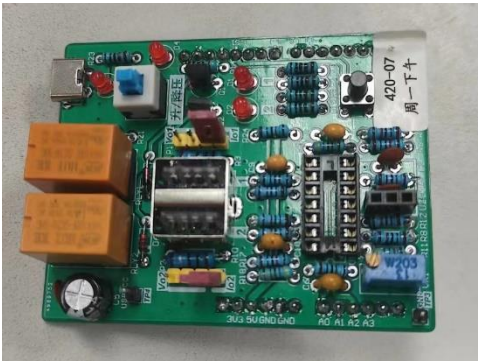
### 1，电装过程

本组采用从低到高全部组装完再测试的过程。这样使得焊接过程更方便些。

电装顺序为：若干小号电阻→R3、R10、R21 三个大号电阻→TYPE-C 接口→D6、D5 两个二极管→C1、C2、C3 三个电容→C4、C6、C9、C10 四个电容→四个 LED 灯→Q1、Q2 三极管→C5 电容→S1 按钮→LM324AN→RYL1、RYL2 继电器→LM35DZ→SW1 自锁开关→VR1 滑动变阻器→JK2 双联 USB 插座→P1、P2 排针→TP3、TP4 两个测试环→反面 J1、J2、J3、J4 四个排针

遇到的问题：电装过程中本应是 12 个 10K 电阻和 6 个 1K 电阻，但是得到的材料刚好反了过来，初装时也没有发现，直到装配到一半发现电阻大小不对，于是进行了艰难的拆卸。所以装电阻时务必先读好数，不要按照仅材料的个数下定论。

成品图：



### 2，测试

#### (1) 测试一：

检测项目	检测结果
电源空载时的输出电压：	5.094V
插上电源后 D3 状态（亮\灭）	亮
插上电源后，标注 5V 处(J1 的 3 脚)的电压 (即以万用表直流电压档测量标注 5V 处对地线 GND 的电压):	5.092V

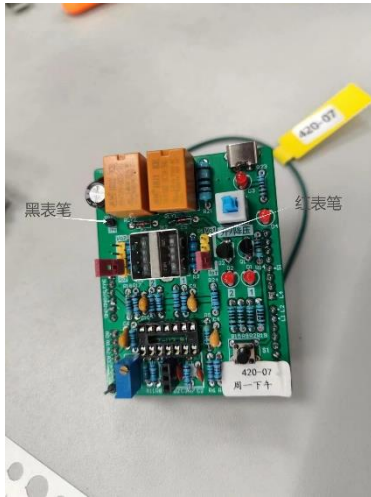


#### (2) 测试二

检测项目	检测结果
以杜邦线连接标注 L1 处 (J3 的 7 脚) 至标注 5V (J1 的 3 脚), 观察到的现象 :	D1 (亮/灭) __亮__, 继电器 RLY1 (吸合/断开) __吸合__, USB 供电插座 1 的供电电压 (即 P1) __5.046__V。
以杜邦线连接标注 L1 处 (J3 的 7 脚) 至标注 GND (J1 的 4 或 5 脚), 观察到的现象 :	D1 (亮/灭) __灭__, 继电器 RLY1 (吸合/断开) __断开__, USB 供电插座 1 的供电电压 (即 P1) __0.017__V。
以杜邦线连接标注 L2 处 (J3 的 8 脚) 至标注 5V (J1 的 3 脚), 观察到的现象 :	D2 (亮/灭) __亮__, 继电器 RLY2 (吸合/断开) __吸合__, USB 供电插座 2 的供电电压 (即 P2) __5.046__V。
以杜邦线连接标注 L2 处 (J3 的 8 脚) 至标注 GND (J1 的 4 或 5 脚), 观察到的现象 :	D2 (亮/灭) __灭__, 继电器 RLY2 (吸合/断开) __断开__, USB 供电插座 2 的供电电压 (即 P2) __0.019__V。

(图片中断电)

测试插座 1 :



测试插座 2 :



### (3) 测试三

检测项目	检测结果
电路上电, 以杜邦线连接标注 L4 处 (J4 的 1 脚) 至标注 5V (J1 的 3 脚), 观察到的现象 :	D4 (亮/灭) __亮__。 这里通过标注 4 所施加的 5V 电压, 就是后续系统能通过软件输出的二进制控制信号“1”。
电路上电, 以杜邦线连接标注 L4 处 (J4 的 1 脚) 至标注 GND (J1 的 4 或 5 脚), 观察到的现象 :	D4 (亮/灭) __灭__。 这里通过标注 4 所施加的 0V 电压, 就是后续系统能通过软件输出的二进制控制信号“0”
电路上电, 按下 S1 按钮并保持, 测量 IO11 (J4 的 4 脚) 电平 (即该点的对地电压) :	IO11 的电平__5.093__V。 这个就是你按下按钮状态下, 系统能通过软件检测到的信号。
电路上电, 松开 S1 按钮, 测量 IO11 (J4 的 4 脚) 电平 (即该点的对地电压) :	IO11 的电平__0.019__V。 这个就是你松开按钮状态下, 系统能通过软件检测到的信号。

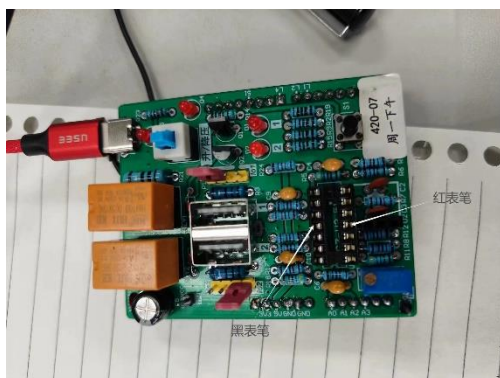
测试图 :



#### (4) 测试四

检测项目	检测结果
LM324AN 的电源电压 (V)	5.093V
LM35DZ 的电源电压 (V)	5.093V

测 LM324AN :



测 LM35DZ :



遇到的问题：在进行测试二测量 USB 接口供电电压时，不能正确测出，通过分析电路图发现是跳线帽未正确放置，以及红黑表笔测试点有问题。应用跳线帽将四针连起来，黑表笔在 TP4 端，才可测其供电电压。

## 二，调试

### (一) 标定

#### 1, 温度值标定

测试项目	测量值
当前实际室温 (摄氏度)	25.3°C
经你完成调试后测试程序显示温度 (摄氏度)	25.36°C
是存在严重的元器件离散性问题？(是\否)	否

#### 2, 电流值标定

测量电流方式：将跳线帽插在连续的一侧，红色端插在靠近跳线帽的排针上，黑色端

插在靠外的排针上，然后接红黑表笔。

先测 P1 处：

(1) 采样电流数据

Val（串口监视器读取电流值）	I（P1 处测量的实际电流值）
588mA	100mA
873mA	150mA

(2) 计算求解 k，b 求得最终的标定公式为：

$$I = \text{val} \times 0.175 - 2.9 \quad (\text{mA})$$

(3) 修改后的代码

```
Sensor1 = getCurrent(1);
Sensor2 = getVoltage();
Sensor3 = getTemperature();

// send sensor values:
Serial.print("Socket #1 current: ");
Serial.print((float)Sensor0*0.175-2.9);
Serial.print(" mA");
Serial.print("\n");
Serial.print("Socket #2 current: ");
Serial.print((float)Sensor1*0.178-4.308);
Serial.printin(" mA");
Serial.print("\n");
Serial.print("Voltage input: ");
Serial.print((float)Sensor2);
Serial.printin(" mV");
Serial.print("\n");
Serial.print("Temperature: ");
Serial.print((float)Sensor3/10);
Serial.printin(" x0.1 Deg Cel");
Serial.printin("\n");
Serial.printin("\n");
delay(500);
```

(4) 改变后的数据

Val（串口监视器读取电流值）/mA	I（P1 处测量的实际电流值）/mA
79.70	80
89.85	90
99.82	100
109.97	110
119.95	120
130.10	130
139.82	140
149.90	150
159.89	160
169.80	170

(5) 另一路电流，按照同上步骤，测得标定公式为

$$I = \text{val} \times 0.178 - 4.308 \quad (\text{mA})$$

(二) 测试任务

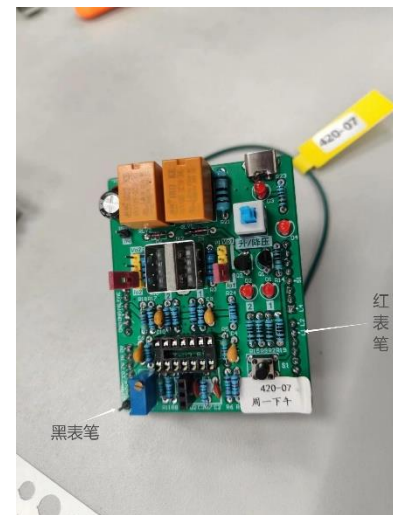
1, 测试任务一

输出命令	1#和 2#插座的状态（通/断）	控制电压（用电压表测量 L1 或 L2）
发送字符 A	智能插座 1#插座的输出电压， TP1 的电压： __4.988__V	J4 的_L1__（选 L1 或 L2，即你程序中输出控制信号的那个引脚）脚电压： __2.951__V
发送字符 a	智能插座 1#插座的输出电压， TP1 的电压： __0.014__V	J4 的_L1__（选 L1 或 L2，即你程序中输出控制信号的那个引脚）脚电压： __0.000__V

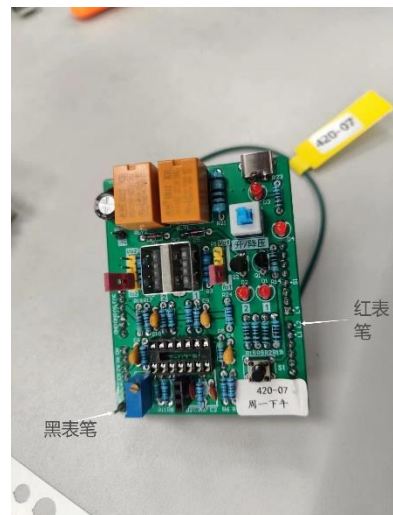


发送字符 B	智能插座 2#插座的输出电压， TP2 的电压： __4.985__ V	J4 的_L2_ (选 L1 或 L2, 即你程序中输出控制信号的那个引脚) 脚电压： __2.948__ V
发送字符 b	智能插座 2#插座的输出电压， TP2 的电压： __0.000__ V	J4 的_L2_ (选 L1 或 L2, 即你程序中输出控制信号的那个引脚) 脚电压： __0.000__

测试 L1:



测试 L2:



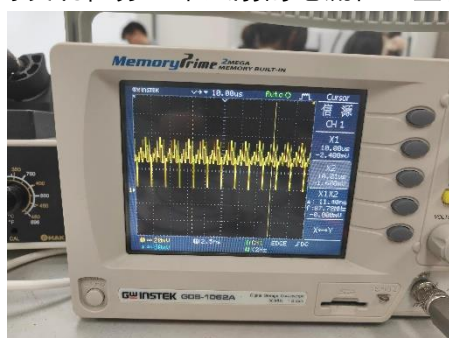
## 2, 测试任务二

项目	测量值	单位
1#插座外接的用电器名称	小台灯	
1#插座电流值	148.8	mA
2#插座外接的用电器名称	可调光台灯	
2#插座电流值 (最亮时)	228.34	mA
2#插座电流值 (最暗时)	37.17	mA
插座电压值	4.962	V
温度测量值 (环境温度)	25.76	°C
温度测量值 (手指触碰温度)	29.97	°C

## 3, 测试任务三

项目	测量值	单位
风扇慢速档电流值	348.0	mA
风扇中速档电流值	265.0	mA
风扇快速档电流值	193.0	mA

测量时发现任务三中风扇的电流值一直在变化，不稳定。测量其供电电压的输出波形，得到



是由于风扇调速使用了 PWM 的控制方法，导致电压不断波动，电流也不稳定。

### （三）软硬件联调任务

#### 1, 任务一

添加的代码段：

```
float set[10]={0.0},ave=0.0;int i=0;    //定义全局变量数组 set 和平均值 ave，计数变量 i

void loop()
{
    // read analog inputs:
    Sensor0 = getCurrent(0);
    Sensor1 = getCurrent(1);
    Sensor2 = getVoltage();
    Sensor3 = getTemperature();

    //average
    if(i<=10&& i>0){
        set[i]=Sensor0*0.175-2.9;        //将接收到的电流值以此赋值到数组中
        i++;
    }else{                                //如果数组赋满，新的值从头开始替换历史值
        i=0;
        set[0]=Sensor0*0.175-2.9;
        i++;
    }
    int sum=0;
    for(int a=0; a<10; a++){
        sum+=set[a];
    }
    ave=sum/10;                           //计算平均值

    // send sensor values:
    Serial.print("Socket #1 current: ");
```

代码采用的是移动平均法，原理就是将接收的电流值赋值到数组，然后取整个数组的平均值，如果数组已满则从头开始替代，每个输出的值都是十个相邻的历史值的平均，从而使输出结果稳定。但由于刚开始时，数组还未被赋满，但是默认平均值 =sum/10，所以开始的值会偏小，经过一小段加载时间后会趋于平稳。

#### 2, 任务二

添加的代码段

```
Sensor3 = getTemperature();
```

```
if(Sensor2<4880){
    digitalWrite(pinLED,HIGH);
```

```

}else{
    digitalWrite(pinLED,LOW);
}
// send sensor values:
Serial.print("Socket #1 current: ");

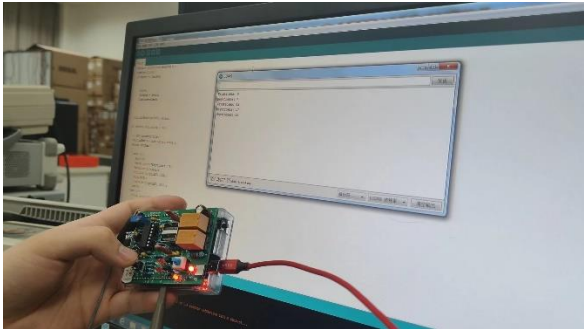
```

当测得电压值小于 4880mV 时，D4 点亮；不小于 4880mV 时，D4 熄灭。其中 4880mV 是通过按 SW1 测量出来的正常电压和降压后电压中间的阈值电压。

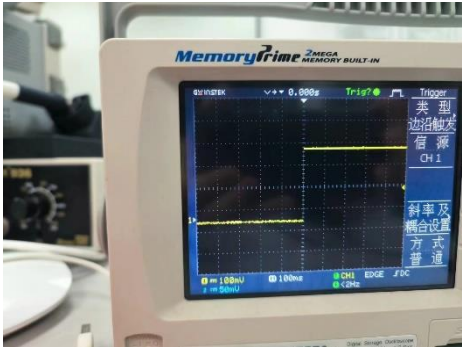
### 3，任务四

s1 按钮的程序读取值（0/1）	
按下 s1 时	0
松开 s1 时	1

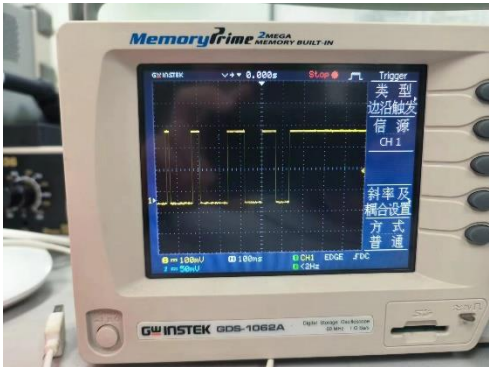
测试中发现，慢按时计数正常，快按时计数偏高，会出现误读情况。



连接示波器 CH1，调节触发方式为“普通”，触发类型为“边沿触发”；触发耦合为“DC”，触发抑制为“关”，触发噪声抑制为“关”。调节触发电平“ LEVEL”，调节量程旋钮和扫描周期旋钮使得图像清晰，得出按下/松开按钮时的电平变化为



连续快速点击时波形会出现毛刺



分析后得出误读出现的原因是由于按钮的机械特性，在快速按压过程中按钮高频抖动，导致了电压不稳定，出现误读的情况。

修改后的代码：

```
time1=millis();

//设置按钮敏感度（通过调整时间阈值）
if(keyin != lastkey)
{
    if(time1-lasttime>100){
        acc++;
        lastkey = keyin;
        lasttime=time1;
    }
}
//digitalWrite(pinLED, keyin);
```

修改后无论快按慢按都可准确计数。

（四）遇到的问题

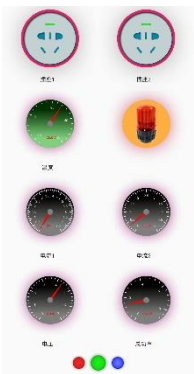
- 1，在任务一中尝试使用加权平均法，即定义历史平均值 a，计数变量 k，当前所得值 sensor，输出的平均值每次都做  $a=a*(k-1)/k+sensor/k$  的运算，输出 a。结果由于缺乏风扇换挡时的重置操作失败，所以仍使用移动平均法。
- 2，在将示波器接入智能插座时，探头误触了 USB 外壳，导致突然断电，且不能重连。最终得知是由于短路触发了电脑 USB 口的保护机制，重启电脑即可。

三，测试

1，测试项目一

- （1）测试目的：测试智能插座可以连一台手机。
- （2）测试用例：组员两部手机
- （3）测试过程及结果：组员 F 先开热点，并修改 SSID 和密码，源代码段上也修改，随后发现可以连接。组员 L 开同样 SSID 和密码的热点，也可以连接。两部手机同时开热点，只有一部可以连接。

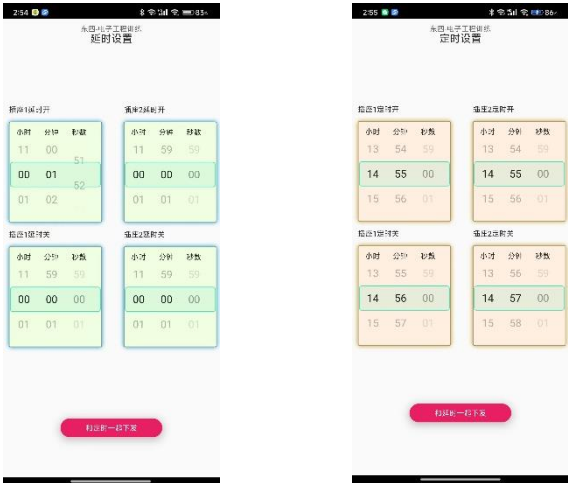
组员 F 开热点	可连
组员 L 开热点	可连
同时开	只连一部





2，测试项目二

- (1) 测试目的：智能插座具备定时开关，延时开关功能
- (2) 测试用例：插座一二分别设定定时开关与延时开关
- (3) 测试过程及结果：插座一设置 14:55 开，14:56 关；插座二设置 14:57 关。插座一二均设置延迟一分钟开，延迟两分钟关。结果发现其均准时开关。



发现的 bug：在延迟开关中出现显示 bug。假设将延迟定为一分钟，那么开始计时后，分钟处没有减一位变为 0，仍然为 1，但是秒处正常减少，并且在过一分钟后成功执行指令。随后又设置了两分钟、三分钟，均发现开始计时过程中分钟数没有先减一的显示问题，但是计时功能正常。由于这是软件开发的 bug，所以不会修。

3，测试项目三

- (1) 测试目的：测试智能插座可以针对超欠电压、超电流、超功率告警并自动断开插座供电。
- (2) 测试用例：用控制变量法，测试某一项时保证其余项均处于正常范围的前提下。分别进行了如下测试：最小电压设置为 5.0V，最大电压设置为 4.5V，最大电流 1、2 设置为 0.1A，最大功率 1、2 为 0.3W；
- (3) 测试过程及结果：将风扇接入 USB 插头并工作，保证其余处于正常范围时，改变测试项，观察结果。

测试项	结果
最小电压 5.0V	app 上警告并且断电
最大电压 4.5V	app 上警告并且断电
(风扇接 1) 最大电流 1 设置为 0.1A	app 上警告并且断电
(风扇接 1) 最大功率 1 设置为 0.3W	app 上警告并且断电
(风扇接 2) 最大电流 2 设置为 0.1A	app 上警告并且断电
(风扇接 2) 最大功率设置为 0.3W	app 上警告并且断电
设置最小电压 4.0V，最大电压 5.0V，最大电流 3.0A，最大功率 8.0W	(接 1) 正常工作，不警告
	(接 2) 正常工作，不警告

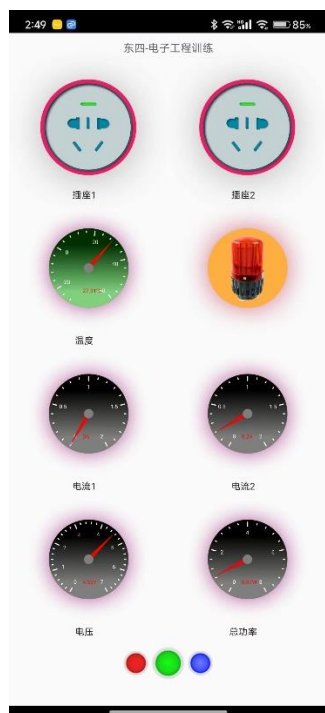
4，测试项目四

(1) 测试目的：测试插座有开关手动控制，温度、电压、电流和功率显示的功能

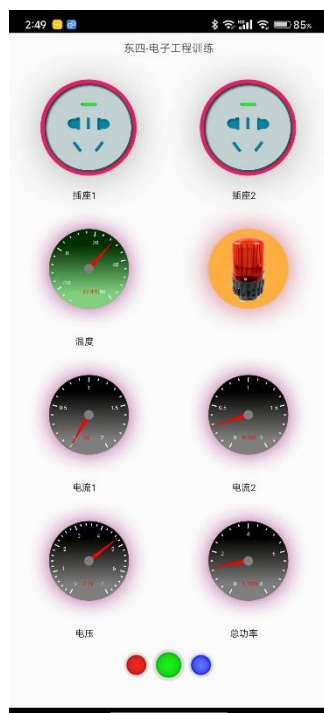
(2) 测试用例：风扇接入插座，调三档，在 app 上观察其显示。

(3) 测试过程及结果：

一档



二档



三档

