**实验报告**

课程名称： 电子工程训练（甲） 任课老师： 金向东/马洪庆/李培弘

实验名称： 智能插座DIY 实验日期： 2021.4-2021.5

姓名： 许嘉怡、杜颜竹君 组别： 421-05

专业： 信息工程、微电子科学与工程 学号： 3200102569/3200105864

**1 实验目的和要求**

**1.1 实验目的**

（1）小组DIY智能插座；

（2）完成智能插座的电装与调试。

**1.2 实验要求**

（1）支持Wi-Fi远程开关遥控；

（2）支持插座电压、温度以及2路电流值的测量；

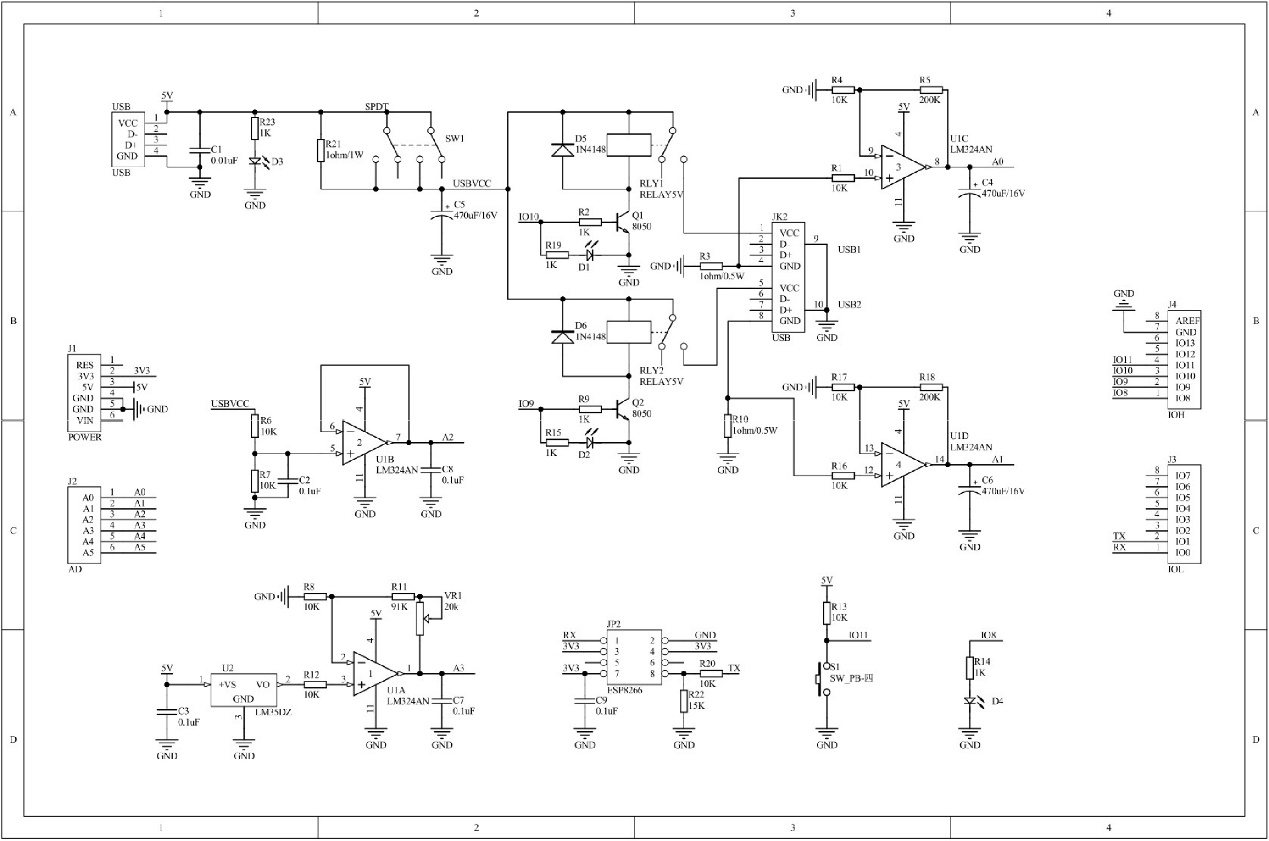
（3）支持各路插座的定时开关和延时开关；

（4）针对插座的过电压、欠电压、超高/低温范围过电流、过功率的断电保护自控制。

（5）支持安卓系统的APP遥测、遥控。

（6）……

**2 实验原理**

**3 主要仪器设备**

电烙铁、万用表、小台灯、可调光台灯、小风扇等。

**4 操作方法和实验步骤**

（1）了解智能插座电路设计方法、理解电路结构。

（2）手工焊接PCB。

（3）电路模块测试，进行对硬件的功能验证与纠错。

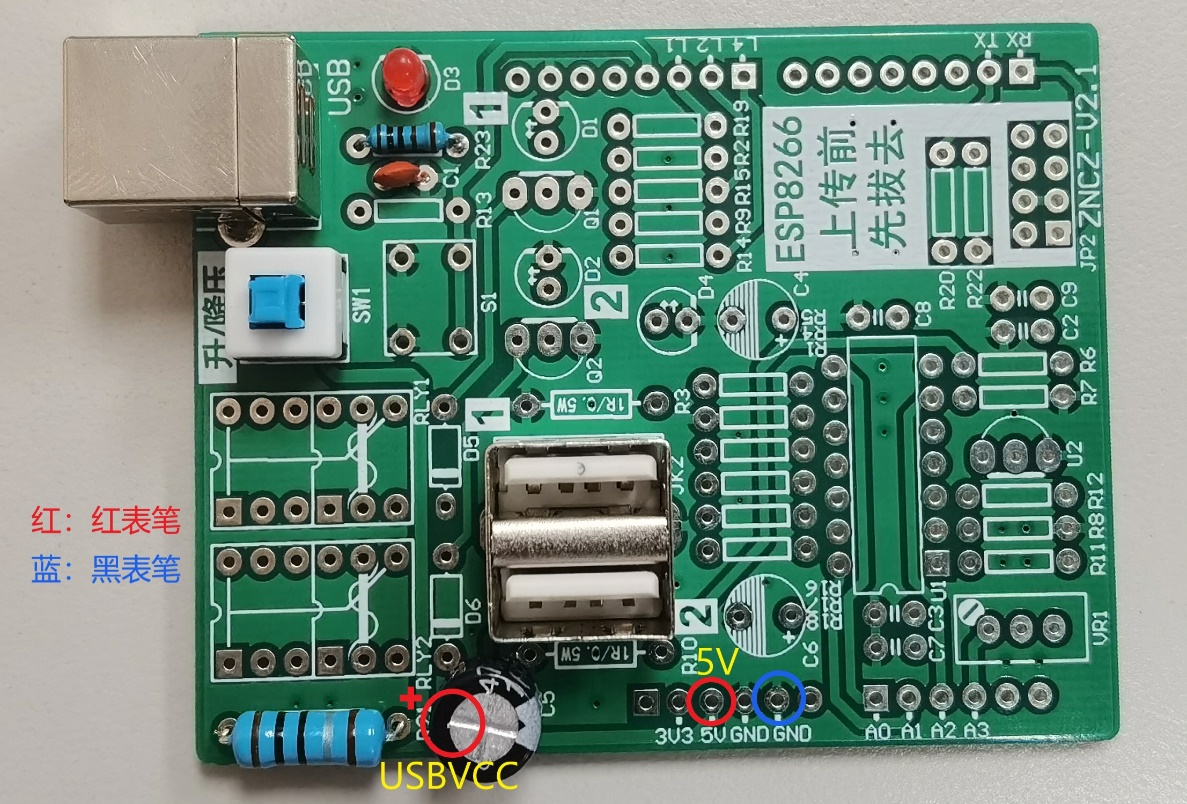
（4）系统测试，结合软件进行对全系统的功能验证与纠错。

**5 实验结果和分析**

**（一）电装部分**

**测试一、供电电路测试**

|  |  |
| --- | --- |
| **检测项目** | **检测结果** |
| 插上电源后LED3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） | 5.088V |
| 插上电源后，USBVCC的电压（即以万用表直流电压档测量USBVCC点对地线GND的电压）（V）： | 5.068V |

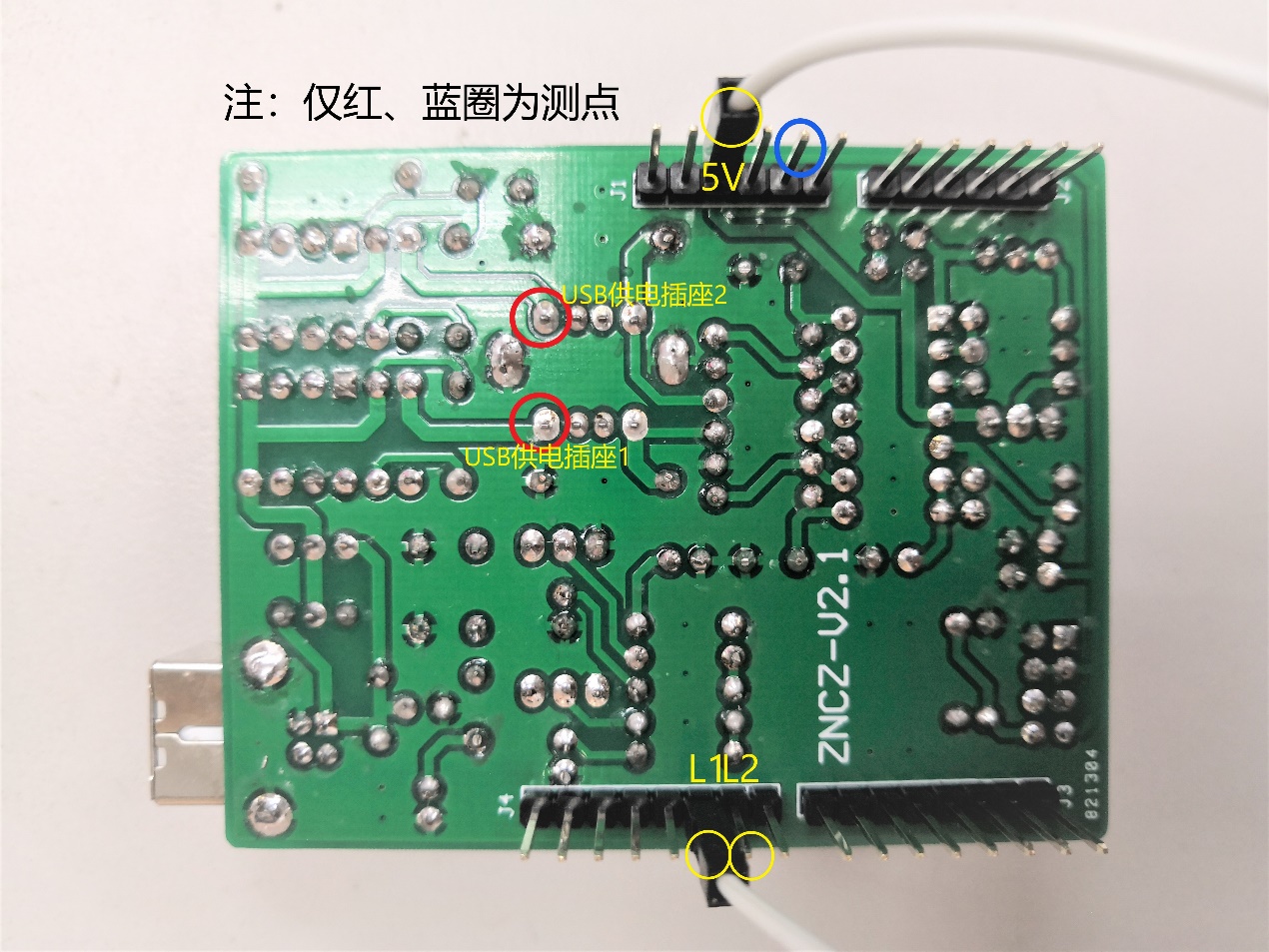


**USB2供电通路信号路径分析**

USB2的电路通道：引脚IO9（电路板上标注L2处）通过高、低电平（即+5V/0V），可控制三极管Q2的导通/截至，从而决定继电器RLY2的通断，最终决定USB供电插座JK2上的USB2口是否有5V电源输出。供电电流将从USB2口的VCC引脚流出，经过外接用电器，从USB2口的GND流回。

**测试二、USB插座供电控制电路测试**

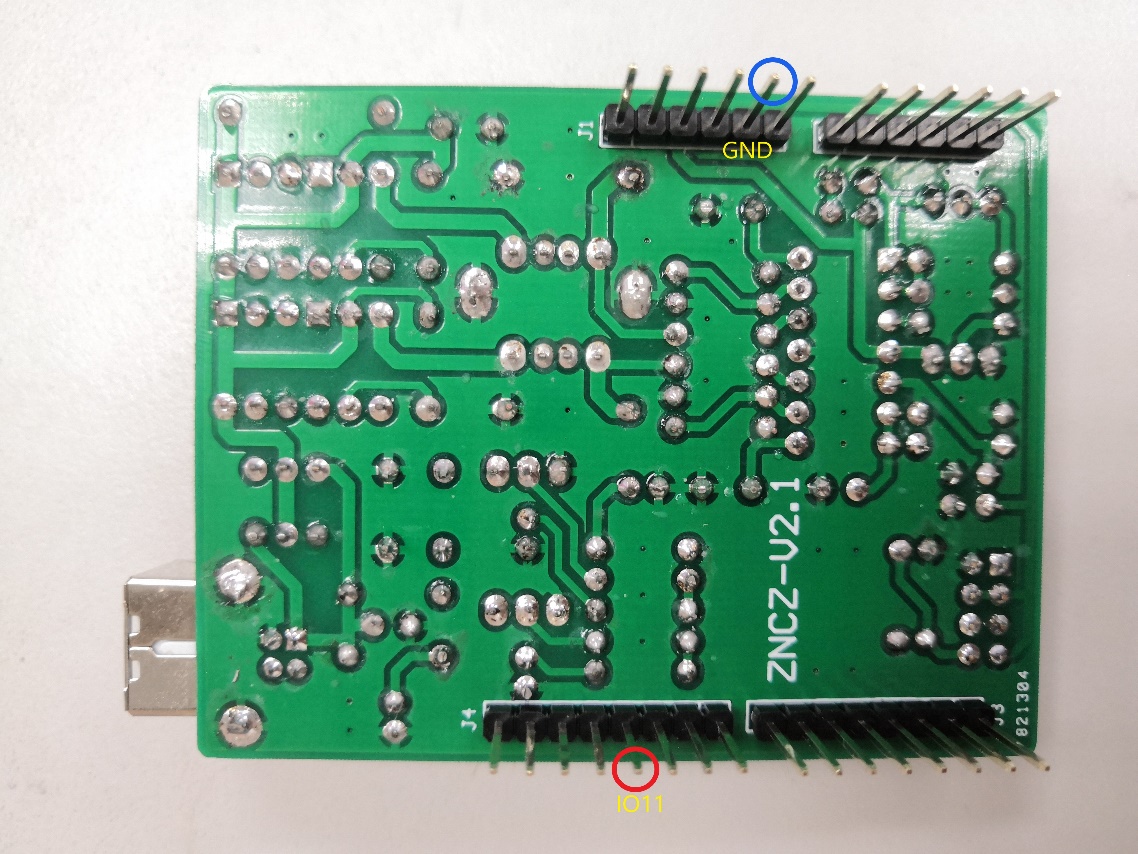
|  |  |
| --- | --- |
| **检测项目** | **检测结果** |
| 以杜邦线连接标注L1处（J4的3脚）至标注5V（J1的3脚），观察到的现象： | LED1（亮/灭） 亮 ，继电器RLY1（吸合/断开） 吸合 ，USB供电插座1的供电电压（V） 5.107V 。 |
| 以杜邦线连接标注L1处（J4的3脚）至标注GND（J1的4或5脚），观察到的现象： | LED1（亮/灭） 灭 ，继电器RLY1（吸合/断开） 断开 ，USB供电插座1的供电电压（V） 0V 。 |
| 以杜邦线连接标注L2处（J4的2脚）至标注5V（J1的3脚），观察到的现象： | LED2（亮/灭） 亮 ，继电器RLY2（吸合/断开） 吸合 ，USB供电插座2的供电电压（V） 5.110V 。 |
| 以杜邦线连接标注L2处（J4的2脚）至标注GND（J1的4或5脚），观察到的现象： | LED2（亮/灭） 灭 ，继电器RLY2（吸合/断开） 断开 ，USB供电插座2的供电电压（V） 0V 。 |



**测试三、指示灯与按钮电路测试**

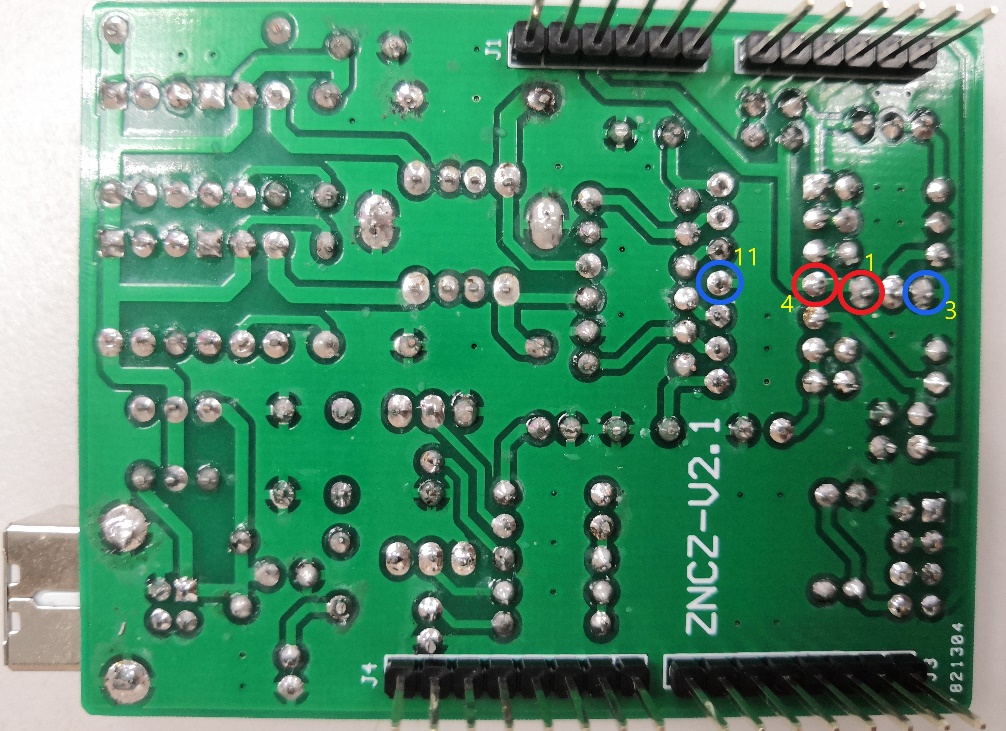
测试D4的控制是否良好：电路上电，若以杜邦线连接标注L4处（J4的1脚，即IO8）至标注5V（J1的3脚），D4亮；且以杜邦线连接标注L4处（J4的1脚，即IO8）至标注GND（J1的4或5脚），D4灭，则D4控制良好。

|  |  |
| --- | --- |
| **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： | LED4（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： | LED4（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 0 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 5.151 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。 |



**测试四、电装总体完成后的初步通电测试**

|  |  |
| --- | --- |
| **检测项目** | **检测结果** |
| LM324AN的电源电压（V）： | 5.164V |
| LM35DZ的电源电压（V）： | 5.155V |



**电装小结**

在完成了全部电装任务的基础上，总结自己的电装经验：

（1）在报告中列出你的电装元器件顺序。

顺序：供电电路-调试中超、欠电压状态模拟发生电路-将剩余电子元器件由低到高进行电装。其中供电电路与调试中超、欠电压状态模拟发生电路电装顺序如下。

供电电路：R23，C1，D3，USB插座。

调试中超、欠电压状态模拟发生电路：R21，C5，SW1，双联USB插座JK2。

原因：最初按照pdf中提供的测试顺序，分模块进行电装；但发现安装USB插座后导致焊接过程变得困难，如出现板子不平衡、较低的元器件难以紧贴PCB板、引脚较短的元器件容易掉出等降低电装效率的情况，故改变策略，参照物料表按从低到高的顺序安装剩余元件，之后再统一进行测试。

（2）列出你在上述电装期间遇到的各类问题。

①由于先焊接了双联USB插座JK2，在后续焊接过程中经常出现板子不平衡的情况；

②后续焊接时高度较低的电子元器件容易漏出，例如电阻更易一高一低；

③红黑表笔测量数据时偶尔出现接触到其他引脚的情况，导致示数不稳定或出错。

（3）分析问题并给出纠错解决方案与解决后的效果。

①焊接技术需要提升，另外也可以借助其他的工具或者请同组同学帮忙固定板子；

②利用镊子夹紧引脚或固定元器件位置，同组同学进行焊接，或在焊接好后再次加热焊锡，调整元器件高低不平的情况；

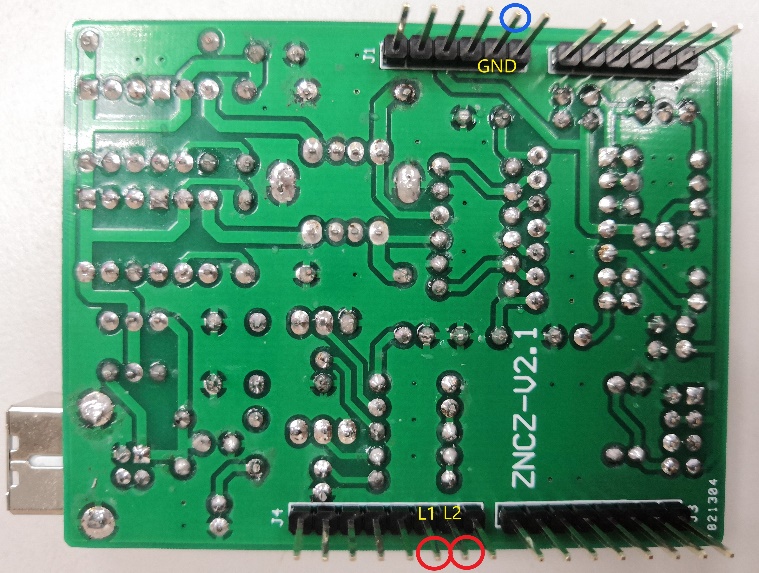
③将红黑表笔头部的凹槽尽可能贴紧引脚，找准表笔角度，使示数符合预期且保持稳定。

**（二）调试部分**

|  |  |
| --- | --- |
| **项目** | **测量值** |
| 当前实际室温（摄氏度） | 23℃ |
| 经你完成调试后测试程序显示温度（摄氏度） | 23℃ |
| 是存在严重的元器件离散性问题？（是/否） | 否 |

|  |  |  |
| --- | --- | --- |
| **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 | 2#通 | 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： 3.432 V |
| 发送字符a | 2#断 | 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： -1.492 V |
| 发送字符B | 1#通 | 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： 3.424 V |
| 发送字符b | 1#断 | 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： -1.384 V |

**测试一**



**测试二**

|  |  |  |
| --- | --- | --- |
| **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 | 266.11 | 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） | 338.18 | mA |
| 2#插座电流值（最暗时） | 64.68 | mA |
| 插座电压值 | 4.96 | V |
| 温度测量值（环境温度） | 23 | ℃ |
| 温度测量值（手指触碰温度） | 32 | ℃ |

**测试三**

|  |  |  |
| --- | --- | --- |
| **项目** | **测量值** | **单位** |
| 风扇慢速档电流值 | 259.10 | mA |
| 风扇中速档电流值 | 292.98 | mA |
| 风扇快速档电流值 | 323.22 | mA |

备注：测量值取最小电流与最大电流的平均值。

测量时读数异常：风扇电流测量值随时间规律性变化明显（由小到大再由大到小）；串口监视器中温度不断变化，而室温应基本保持不变。

测试方法：核心思想为分别改变系统、测试环境，观察读数的变化规律。例如针对风扇：将风扇更换为其他外接用电器进行测试，发现电流可以保持稳定；检查测试环境如软件、代码及arduino板、PCB板等，未发现明显问题，说明风扇系统有问题、测试环境没有问题。

**程序代码**

①在测控任务一、二中均对串口监视器的数据打印代码稍作了修改：

// send sensor values:

sysnow=millis();//用于删除delay(500)后仍每隔0.5s在串口监视器打印数据

if(sysnow-sysbefore>=500){

sysbefore=sysnow;

Serial.print("Socket #1 current: ");

Serial.print((float)Sensor0 / 100);

Serial.print(" mA");

Serial.print("\t\t");

Serial.print("Socket #2 current: ");

Serial.print((float)Sensor1 / 100);

Serial.println(" mA");

Serial.print("Voltage input: ");

Serial.print(fSensor2);

Serial.println(" V");

Serial.print("Temperature: ");

Serial.print(Sensor3, DEC);

Serial.println(" Deg Cel");

Serial.println();

Serial.println();

}

②测控任务一：

在setup函数之前定义：

const int keyPin11 = 11;

unsigned long before = 0;

unsigned long sysbefore=0;

unsigned long sysnow;

int laststate=1;//记录LED按下按键前状态，1表示闪烁，0表示停止闪烁（常亮）

int n=0;//用于记住LED4上次状态为灭（0）还是亮（1）

在loop函数中：

unsigned long now=millis();

int v1=digitalRead(keyPin11);

int v2=v1;//v2保存一次电平数据

delay(10);

v1=digitalRead(keyPin11);//v1保存10ms后的电平数据

if((v1-v2==1)||(now-before>=1000)){//上升沿触发，如果按键被弹起就切换状态（不理会按键按下）

if(v1-v2==1){//按键被弹起

delay(200);//屏蔽电平抖动

switch(laststate){

case 1:laststate=0;//之前状态为闪烁，当前切换为停止闪烁

digitalWrite(ledPin8,HIGH);//停止闪烁，常亮

while(digitalRead(keyPin11)==1);//当按键一直没有被按下，停留在循环中，保持常亮状态

break;

case 0:laststate=1;break;//之前状态为停止闪烁，当前切换为闪烁

}

}

else if(laststate==1) goto label;

else{

label:

before=now;//在闪烁状态，每隔1s切换亮灭，用变量before和now计时

if(n==1){//之前为亮，当前切换为灭

n=0;

digitalWrite(ledPin8,LOW);

}

else{//之前为灭，当前切换为亮

n=1;

digitalWrite(ledPin8,HIGH);

}

}

}

③测控任务二：

在setup函数之前定义：

unsigned long before = 0;

int n = 1; //用于记住LED4上次状态为灭（0）还是亮（1）

float fSensor3;

在loop函数中：

unsigned long now=millis();

Sensor2 = analogRead(A2) / 4;

fSensor2 = Sensor2 \* 1.718 / 220 \* 5;

if(fSensor2-fSensor3>=0.1){//串口监视器显示按下SW1按钮时Voltage input为4.92-4.96V,弹起时约为4.76V，故设置0.1V的电压差为条件

while(fSensor2-fSensor3>-0.1){

now=millis();

if(now-before>=500){

before=now;//重新开始计时

switch(n){

case 1:n=0;digitalWrite(ledPin8,LOW);break;

case 0:n=1;digitalWrite(ledPin8,HIGH);break;

}

}

fSensor3=fSensor2;

delay(10);

Sensor2 = analogRead(A2) / 4;

fSensor2 = Sensor2 \* 1.718 / 220 \* 5;

}

}

if(fSensor2-fSensor3<=-0.1){

digitalWrite(ledPin8,HIGH);

}

**（三）系统测试部分**

注：已分别对1#与2#插座进行系统测试，结果均正常，部分测试由于篇幅仅提供其中一组插座的测试图片。

**测试一、智能插座连接手机**

功能正常，如图（注：前后共使用两个手机进行测试，故截图界面或有所不同）。

**测试二、定时/延时开/关**

分别针对1#与2#插座进行定时/延时开/关的测试，功能正常，如图（以插座2#为例）。

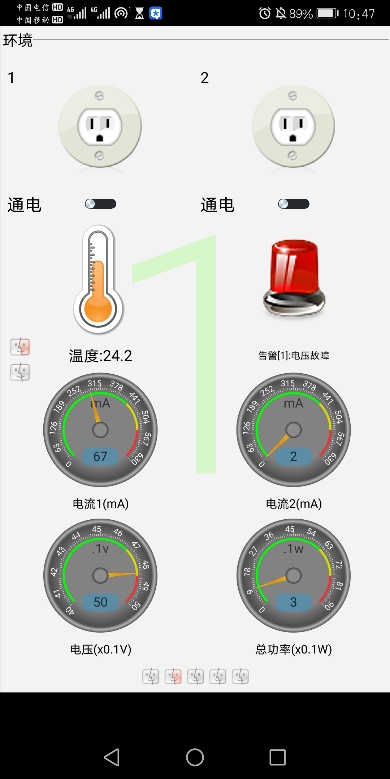
定时开/关：设置10:18定时开，10:20定时关；实际10:18定时开，10:20定时关（指app中显示时间而非手机系统时间）。

延时开/关：设置延时3min开，延时5min关；实际延时3min开，延时5min关。

**测试三、告警系统**  
①超/欠电压：

1#/2#：

设置最高电压为4.9V，按下按钮SW1，系统告警并自动断电，如图。

设置最低电压为4.9V，弹起按钮SW1，系统告警并自动断电，如图。

④高温/低温：

设置最高温度为23℃，实际为24℃，系统告警并自动断电，如图。

设置最低温度为25℃，实际为24.9℃，系统告警并自动断电，如图。

②超电流：设置最大电流为320mA，系统告警并自动断电，如图。

③超功率：

1#总功率1.57W，设置最大功率为1.5W, 系统告警并自动断电，如图。

2#总功率1.54W，设置最大功率为1.5W, 系统告警并自动断电，如图。

**测试四、插座开关手动控制**

1#/2#插座均可正常控制，温度、电压、电流和功率显示正常，见上述及下方测试图片。

备注：智能插座app显示时间与手机系统时间无法同步，偶尔出现时差较大现象；app中时间流速较实际时间更快。

**6 讨论与心得**

（1）电装部分需要细心与耐心，也要在每一个板块完成后进行测试及时发现错误；调试过程也是了解智能插座各项功能的过程，养成随时记录实验数据的习惯也很重要；系统测试时，APP的控制不是特别顺利，由于发现APP的系统时间比实际时间快，所以我们在设置定时、延时开关时对时间也做了相应的调整。

——许嘉怡

（2）

电装：焊接时尽量组员合作，一位同学焊接时，另一位同学可以负责看清电子元器件的极性与焊接位置、辨认电阻阻值、及时递送材料、固定板子等等，提高焊接效率。

调试：进行测控任务代码编写时应尽量理解各行代码含义，并考虑周全，不应忽略delay()函数带来的延迟、按钮按下导致的200ms内电平的不稳定、全局与局部变量差异等细节，从而方能使代码较为有效，也可活学活用，借助Serial.print()等函数debug，由一段代码延伸开来，常常能使人迅速学到许多额外知识。

系统测试：仔细阅读测试步骤，区分上传状态和运行状态、警惕前后使用的USB接口不同。

——杜颜竹君