**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： 2022.4-2022.5

姓名： szx、tqy 组别： 420-1

专业： 电子科学与技术、电子科学与技术 学号： 3210000000/3210000000

**1 实验目的和要求**

**1.1 实验目的**

（1）焊接DIY智能插座；

（2）完成智能插座的芯片和主板调试；

（3）实现智能插座的软硬件联调。

**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）支持实时查看电流电压状态。

**2 主要仪器设备**

电烙铁、万用表、小台灯、可调光台灯、小风扇、 示波器、 PC。

**3 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。

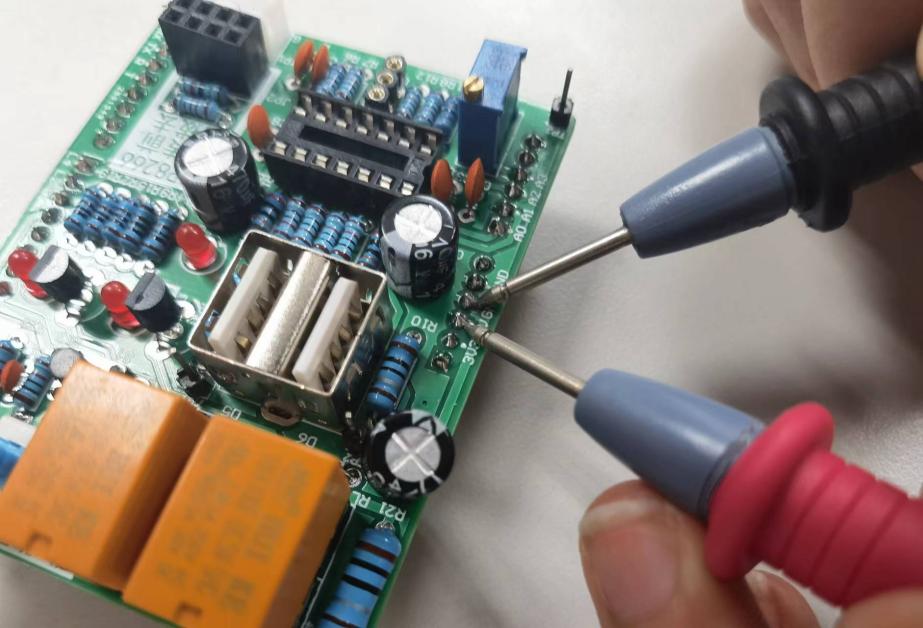
**4 实验结果和分析**

1. **电装部分**

**焊接顺序：未按照模块进行焊接，而是根据元件高低，由电阻开始，再到二极管、三极管、开关等元器件，按由低到高的顺序进行安装。**

**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） | **5.092V** |
| 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）： | **2.091V** |

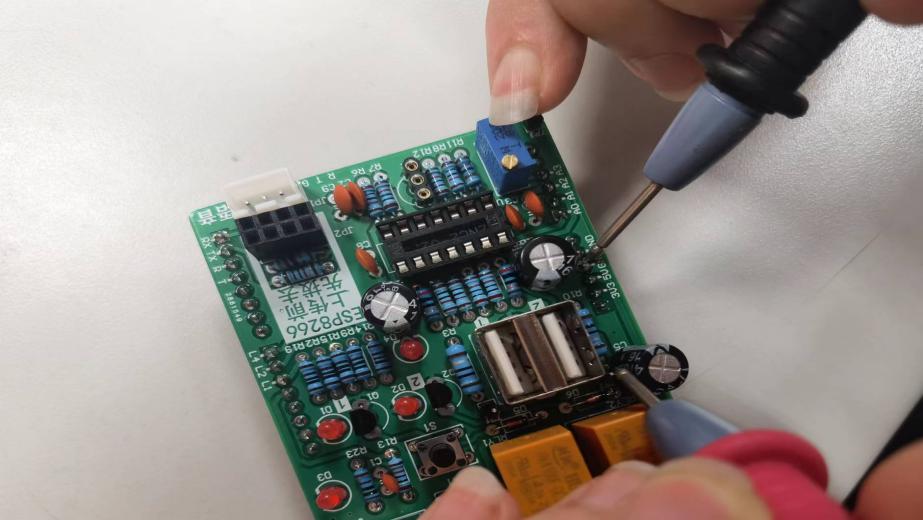


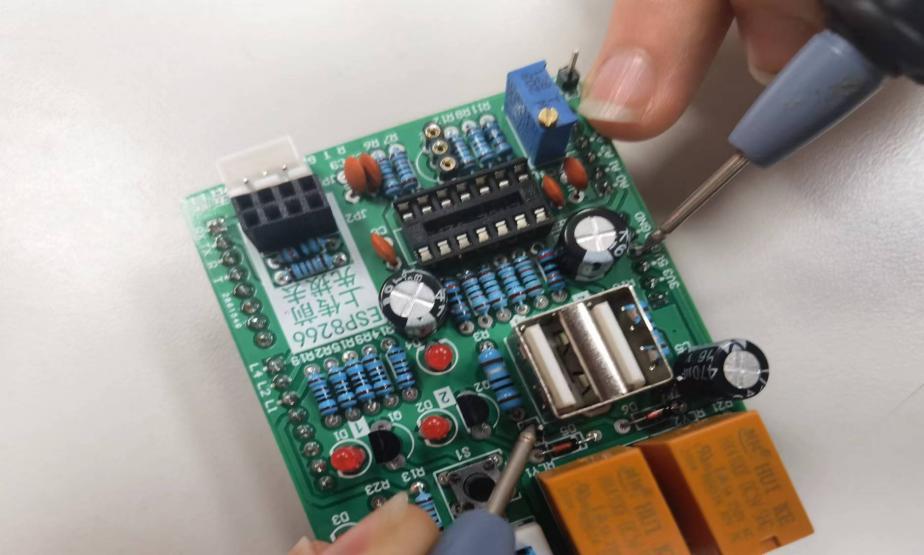
**供电通路信号路径分析**

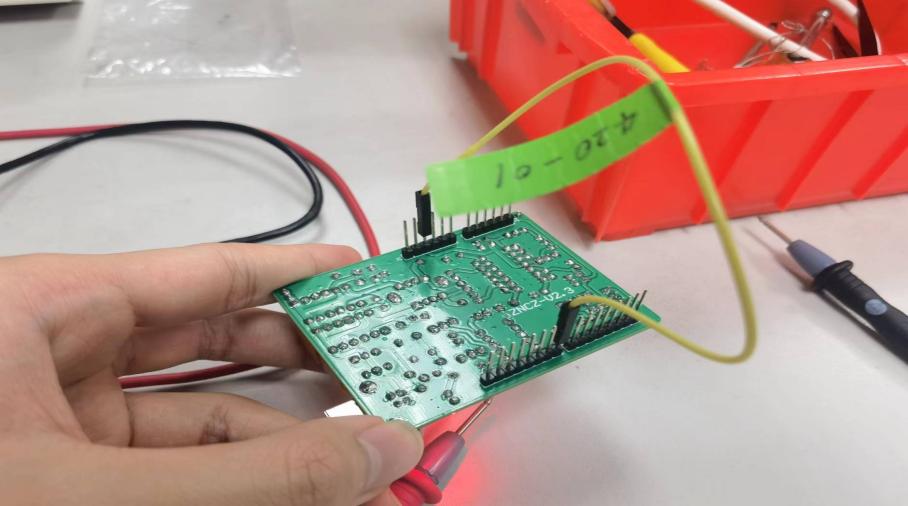
电路通道：5V的引脚通过高、低电平，可控制三极管Q的通断，从而决定继电器RLY的通断，最终决定USB供电插座JK2上的USB口是否有5V电源输出。供电电流将从USB口的VCC引脚流出，经过外接用电器，从USB口的GND口流回。

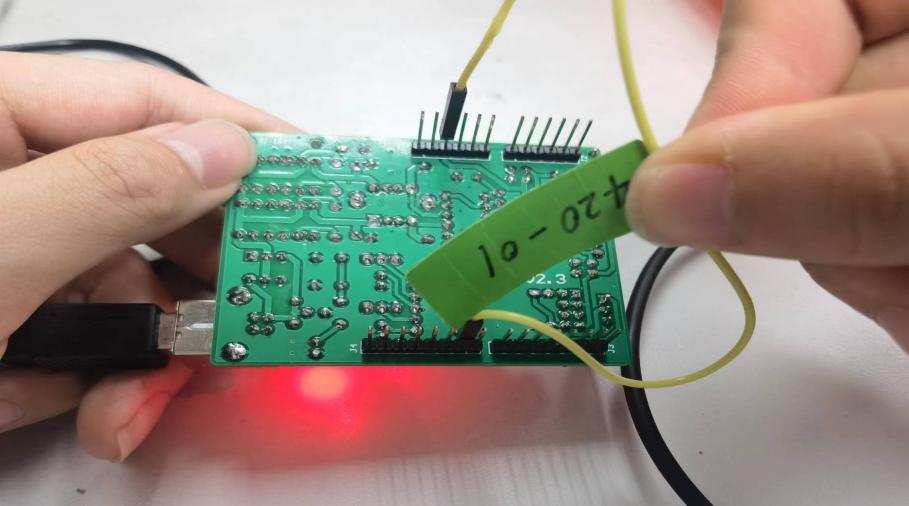
**测试二、USB插座供电控制电路测试**

|  |  |
| --- | --- |
| **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） 5.071V 。 |
| 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） 0.005V 。 |
| 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） 5.072V 。 |
| 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） 0.027V 。 |



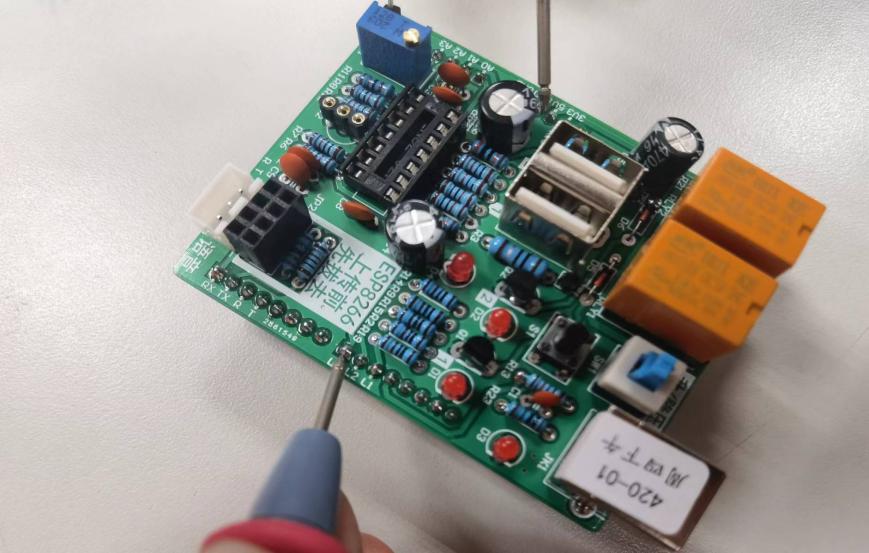


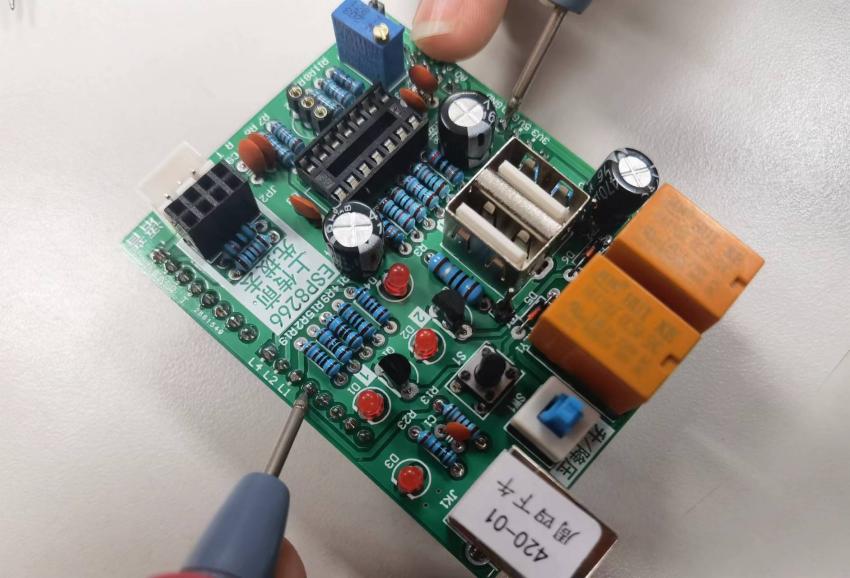


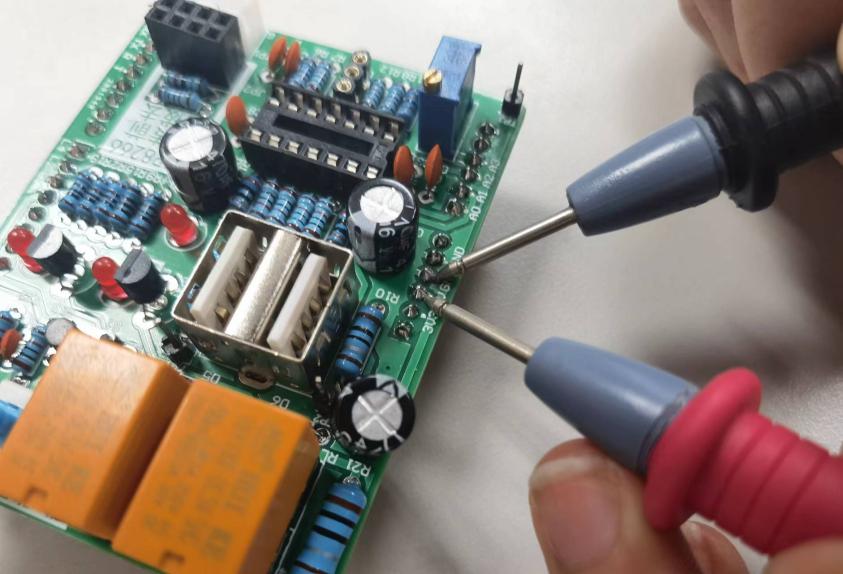


**测试三、指示灯与按钮电路测试**

|  |  |
| --- | --- |
| **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 5.088 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。 |

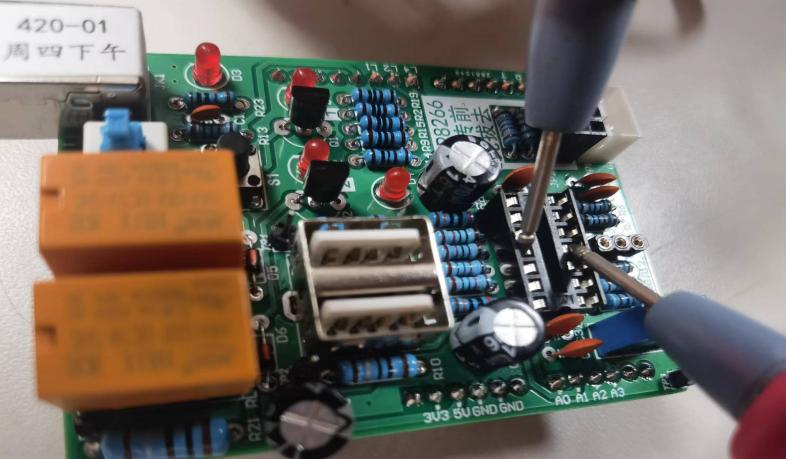






**测试四：电装总体完成后的初步通电测试**

|  |  |
| --- | --- |
| **检测项目** | **检测结果** |
| LM324AN的电源电压（V）： | 5.092V |
| LM35DZ的电源电压（V）： | 5.093V |





**遇到的问题：**

1. 在测量过程中，时常触碰到其他引脚，导致测量值突变和电路的短路。
2. 焊接时，由于电解电容过大，导致引脚TP2无法插入孔中。

**解决及效果：**

1. 只用笔尖触碰测量点，不会在碰到其他引脚，但万用表读数会发生抖动。
2. 将TP2引脚周围的塑料层进行一定的修剪，使得其成功插入其中，但也导致TP2引脚周围空间较小，给测量带来一定难度。

**（二）调试部分**

**测试一**

|  |  |
| --- | --- |
| **项目** | **测量值** |
| 当前实际室温（摄氏度） | 24.9 |
| 经你完成调试后测试程序显示温度（摄氏度） | 25 |
| 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 | TP1输出电压：4.417V | 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： 4.403 V |
| 发送字符a | TP1输出电压：0.169V | 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： 0.173 V |
| 发送字符B | TP2的电压：4.420V | 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： 4.419 V |
| 发送字符b | TP2的电压：0.135V | 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： 0.135 V |



**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 | 286.44 | 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） | 68.37 | mA |
| 插座电压值 | 4.88 | V |
| 温度测量值（环境温度） | 24.9 | ℃ |
| 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 | 293.25 | mA |
| 风扇中速档电流值 | 314.88 | mA |
| 风扇快速档电流值 | 332.24 | mA |

**出现的问题：**测量风扇电流时，电流示数在不断发生周期性的改变，所以测量值取的是变化居中的值作为实际测量值。

**（三）放大器参数测试**

|  |  |
| --- | --- |
| **温度放大器电路运行参数(V)** | |
| **Vti** | **0.290** |
| **Vtadj** | **2.927** |
| **Vt+** | **0.291** |
| **Vt-** | **0.291** |
| **Vto** | **2.994** |

|  |  |
| --- | --- |
| **电压放大器电路运行参数（V）** | |
| **Vusbvcc** | **2.468** |
| **V+** | **2.461** |
| **V-** | **0.078** |
| **Vo** | **2.456** |

|  |  |
| --- | --- |
| **USB1电流测量放大器运行参数（V）** | |
| **V1in** | **0.135** |
| **V1+** | **0.133** |
| **V1-** | **0.105** |
| **V1o** | **2.255** |

|  |  |
| --- | --- |
| **USB2电流测量放大器运行参数（V）** | |
| **V2in** | **0.125** |
| **V2+** | **0.123** |
| **V2-** | **0.104** |
| **V2o** | **2.264** |

1. **代码测试**

**Text 1:记录按下多少次s1**

**/\***

**Serial Call and Response**

**Language: Wiring/Arduino**

**This program sends an ASCII A (byte of value 65) on startup**

**and repeats that until it gets some data in.**

**Then it waits for a byte in the serial port, and**

**sends three sensor values whenever it gets a byte in.**

**Thanks to Greg Shakar and Scott Fitzgerald for the improvements**

**The circuit:**

**\* potentiometers attached to analog inputs 0 and 1**

**\* pushbutton attached to digital I/O 2**

**Created 26 Sept. 2005**

**by Tom Igoe**

**modified 24 April 2012**

**by Tom Igoe and Scott Fitzgerald**

**This example code is in the public domain.**

**http://www.arduino.cc/en/Tutorial/SerialCallResponse**

**\*/**

**// constants won't change. Used here to set a pin number :**

**const int relayPin9 = 9; // the pin number of the relay pin**

**const int relayPin10 = 10; // the pin number of the relay pin**

**const int ledPin8 = 8; //the pin number of the led pin**

**const int keyPin11 = 11; //the pin number of the key pin**

**int acc;**

**bool keyin, lastkey;**

**void setup()**

**{**

**// set the digital pin as output:**

**pinMode( relayPin9, OUTPUT );**

**pinMode( relayPin10, OUTPUT );**

**pinMode( ledPin8, OUTPUT );**

**pinMode(keyPin11, INPUT);**

**// start serial port at 9600 bps:**

**Serial.begin(115200);**

**while (!Serial) {**

**; // wait for serial port to connect. Needed for Leonardo only**

**}**

**digitalWrite(ledPin8, HIGH);**

**digitalWrite(relayPin9, LOW);**

**digitalWrite(relayPin10, LOW);**

**Serial.println("UART ready...");**

**acc = 0;**

**keyin = 1;**

**lastkey = keyin;**

**}**

**void loop()**

**{**

**delay(6.5); // 添加函数，保证避免毛刺产生**

**keyin = digitalRead(keyPin11);**

**if (Serial.available() > 0) {**

**// get incoming byte:**

**char inChar = (char)Serial.read();**

**switch (inChar)**

**{**

**case '0':**

**acc = 0;**

**Serial.print("Keystrokes: ");**

**Serial.print(acc);**

**Serial.print(" Key input: ");**

**Serial.println(keyin);**

**break;**

**case 'r':**

**Serial.print("Keystrokes: ");**

**Serial.print(acc);**

**Serial.print(" Key input: ");**

**Serial.println(keyin);**

**break;**

**default:**

**break;**

**}**

**}**

**if(keyin != lastkey)**

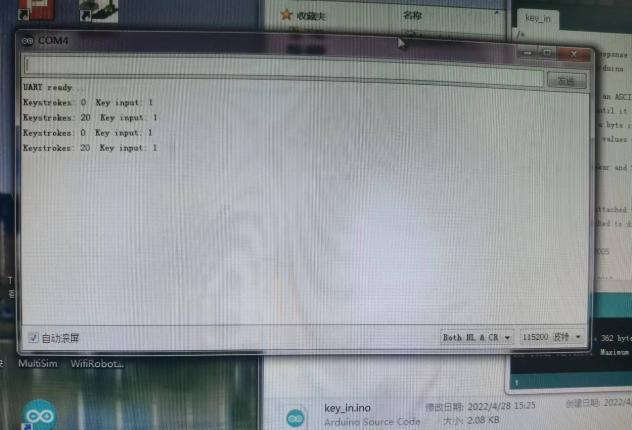
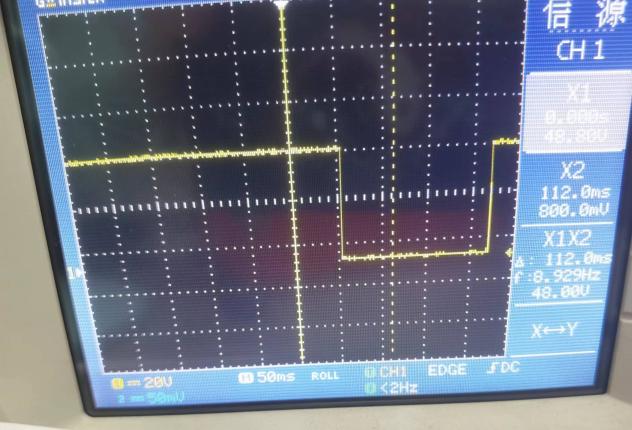
**{**

**acc++;**

**lastkey = keyin;**

**}**

**}**

****

**（按S1十次）**

**加入delay函数，电路保持当前状态一段时间，保证电路获得反应，从而避免毛刺，使测量值准确。**

**Text 2: 通过SW1开关调整电压，并使D4具有预警功能**

**/\***

**Serial Call and Response**

**Language: Wiring/Arduino**

**This program sends an ASCII A (byte of value 65) on startup**

**and repeats that until it gets some data in.**

**Then it waits for a byte in the serial port, and**

**sends three sensor values whenever it gets a byte in.**

**Thanks to Greg Shakar and Scott Fitzgerald for the improvements**

**The circuit:**

**\* potentiometers attached to analog inputs 0 and 1**

**\* pushbutton attached to digital I/O 2**

**Created 26 Sept. 2005**

**by Tom Igoe**

**modified 24 April 2012**

**by Tom Igoe and Scott Fitzgerald**

**This example code is in the public domain.**

**http://www.arduino.cc/en/Tutorial/SerialCallResponse**

**\*/**

**// constants won't change. Used here to set a pin number :**

**const int relayPin9 = 9; // the pin number of the relay pin**

**const int relayPin10 = 10; // the pin number of the relay pin**

**const int ledPin8 = 8; //the pin number of the led pin**

**const int keyPin11=11;**

**unsigned int Sensor0 = 0; // 6 analog sensors**

**unsigned int Sensor1 = 0;**

**unsigned int Sensor2 = 0;**

**unsigned int Sensor3 = 0;**

**float fSensor2;**

**void setup()**

**{**

**// set the digital pin as output:**

**pinMode( relayPin9, OUTPUT );**

**pinMode( relayPin10, OUTPUT );**

**pinMode( ledPin8, OUTPUT );**

**pinMode(keyPin11, INPUT);**

**// start serial port at 9600 bps:**

**Serial.begin(115200);**

**while (!Serial) {**

**; // wait for serial port to connect. Needed for Leonardo only**

**}**

**//digitalWrite(ledPin10,HIGH);**

**// pinMode(2, INPUT); // digital sensor is on digital pin 2**

**// establishContact(); // send a byte to establish contact until receiver responds**

**digitalWrite(ledPin8, LOW);**

**digitalWrite(relayPin9, LOW);**

**digitalWrite(relayPin10, LOW);**

**Serial.println("UART ready...");**

**}**

**void loop()**

**{**

**// inByte = Serial.read();**

**// read analog input, divide by 4 to make the range 0-255:**

**Sensor0 = analogRead(A0) / 4; //i0**

**/// Serial.println(Sensor0);**

**// Sensor0 =Sensor0 \* 0.042;**

**Sensor0 = Sensor0 \* 4.2 / 5 \* 220;**

**// Sensor0 =Sensor0 \* 2.6/5\*220;**

**Sensor1 = analogRead(A1) / 4; //i1**

**/// Serial.println(Sensor1);**

**// Sensor1 = Sensor1 \* 0.026;**

**Sensor1 = Sensor1 \* 4.2 / 5 \* 220;**

**// Sensor1 = Sensor1 \* 2.6/5\*220;**

**Sensor2 = analogRead(A2) / 4;**

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

**Sensor3 = analogRead(A3) / 4;**

**Sensor3 = Sensor3 \* 0.178;**

**// send sensor values:**

**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();**

**delay(500);**

**if (Serial.available() > 0) {**

**// get incoming byte:**

**char inChar = (char)Serial.read();**

**Serial.println(inChar);**

**switch (inChar)**

**{**

**case 'a':**

**digitalWrite(relayPin10, LOW);**

**break;**

**case 'A':**

**digitalWrite(relayPin10, HIGH);**

**break;**

**case 'b':**

**digitalWrite(relayPin9, LOW);**

**break;**

**case 'B':**

**digitalWrite(relayPin9, HIGH);**

**break;**

**default:**

**break;**

**}**

**}**

**if(fSensor2>4.8){ //通过fSensor的值来反应当前电压，在高于此值是D4亮起预警**

**digitalWrite(ledPin8,HIGH );**

**}**

**else{**

**digitalWrite(ledPin8, LOW);**

**}**

**}**

1. **系统测试**
2. **智能插座能够正常与手机热点和App连接**
3. **App能够控制1、2USB接口（手动控制）**

****

1. **能够对电压等问题进行警告**

****

1. **能够进行延时开关和定时开关**

****

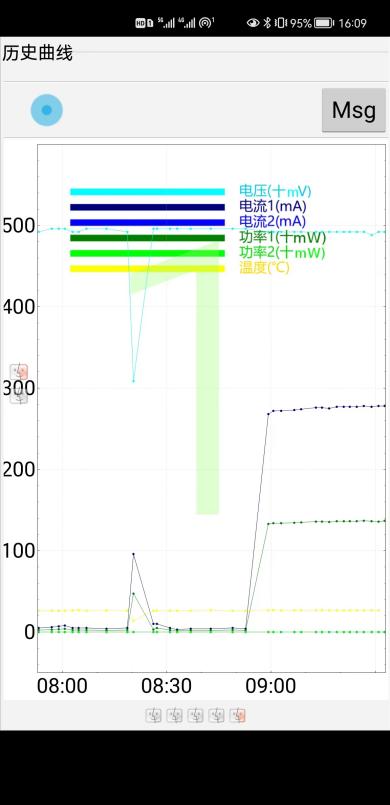
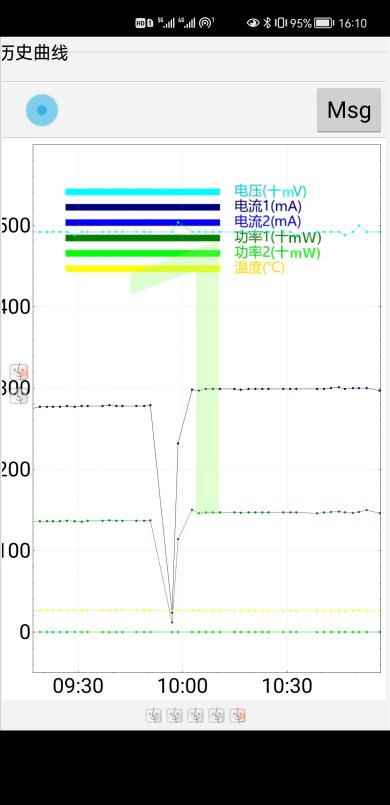
1. **超/欠电压设置正常，能够进行告警**

**设置最高电压为4.9V， 超过时开始告警，USB接口自动断开，SW1弹起。**

**温度设置正常，能够进行告警**

****

1. **能够实时显示插座的电压电流等参数**

****

**5 讨论与心得**

1. **电装要按照一定的顺序，或是模块、或是元器件种类、大小等，不能随意焊接。**
2. **组队队员之间要配合默契，一个人测量、一个人记录、控制代码等。**
3. **使用测试代码时，要先理解代码使用的目的，读懂代码，不能盲目地直接上传就开始实验。**
4. **做实验要有耐心，尤其是在系统测试的时候，App与插座的连接较慢且很不稳定，所以要耐心等待，不然会花费更多的时间。**