

电子信息工程训练
之
产品研发过程中的电装与调试
智能插座DIY系统测试

李培弘

系统测试的目的

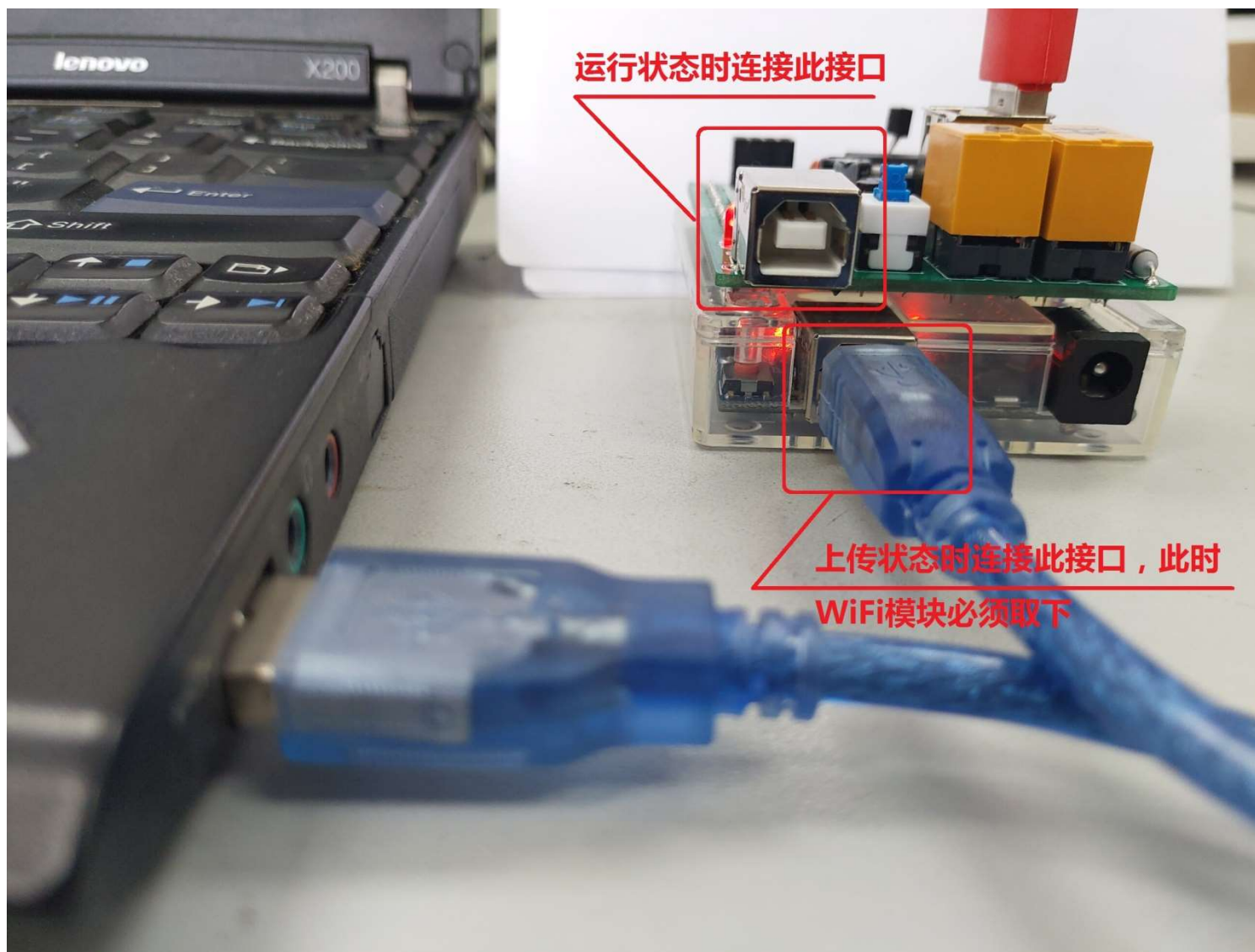
- 通过对系统每项功能在预设的典型场景中的使用测试，评价设计对前期需求分析的完成度。
- 在测试中发现系统设计或实现中存在的Bugs。
- 系统测试后一般会回溯之前的研发过程，以便修复测试中存在的Bugs。
- 在当前的电子信息系统产品生命周期中，系统测试-研发修改-系统再测试-发布系统修正，这种循环会在产品研发时，甚至产品销售部署投入实际运行后，反复迭代进行，直至该产品结束生命周期（一般是产品供应商宣布不再对该产品进行维护升级为止）。

重要提示！

- 今天的实验中，你的硬件板卡会处在2种状态下：
 1. 上传智能插座固件代码到Arduino板中，简称“**上传状态**”：用**USB**线缆，从**PC的USB**接口连接至**Arduino**板的**USB-B**接口；且**Wi-Fi**模块必须从电路中取下。
 2. 运行智能插座固件，简称“**运行状态**”：用**USB**线缆，从直流稳压电源的**5V USB**供电口连接至你们电装的智能插座电路板的**USB-B**接口；且**Wi-Fi**模块应该安装上。

注意：上述2种状态的切换必须要在你的智能插座系统完全断电的情况下进行，即**必须首先断开USB线缆与PC或者与直流稳压电源的连接后再进行！**

重要提示！



组网测试方案1

- （推荐方案）将自己的安卓手机开放热点。其热点名（亦即程序代码中的ssid）由各自确定，但建议由英文字母和数字组成（其他字符未经预先测试不一定有效）；而且其命名要注意和其他组的热点名称有所不同，建议名中包含各自的教室房间号和各自的座位号。密码（亦即程序代码中的password）也建议由英文字母和数字组成（其他字符未经预先测试不一定有效）。这样智能插座上电启动后就会连接该手机热点以实现与该手机的联网。

组网测试方案2

- 仅有苹果手机的小组使用自己的小白路由器组网。需要设置自己的小白路由器配置，即ssid（请设置为“xiaobai_yyy_xx”，其中yyy为420或者421，xx为你的桌号）和密码、智能插座固件代码内的ssid和密码设置。并用U盘将我们提供的HD app复制到小白路由器上，并完成app安装。

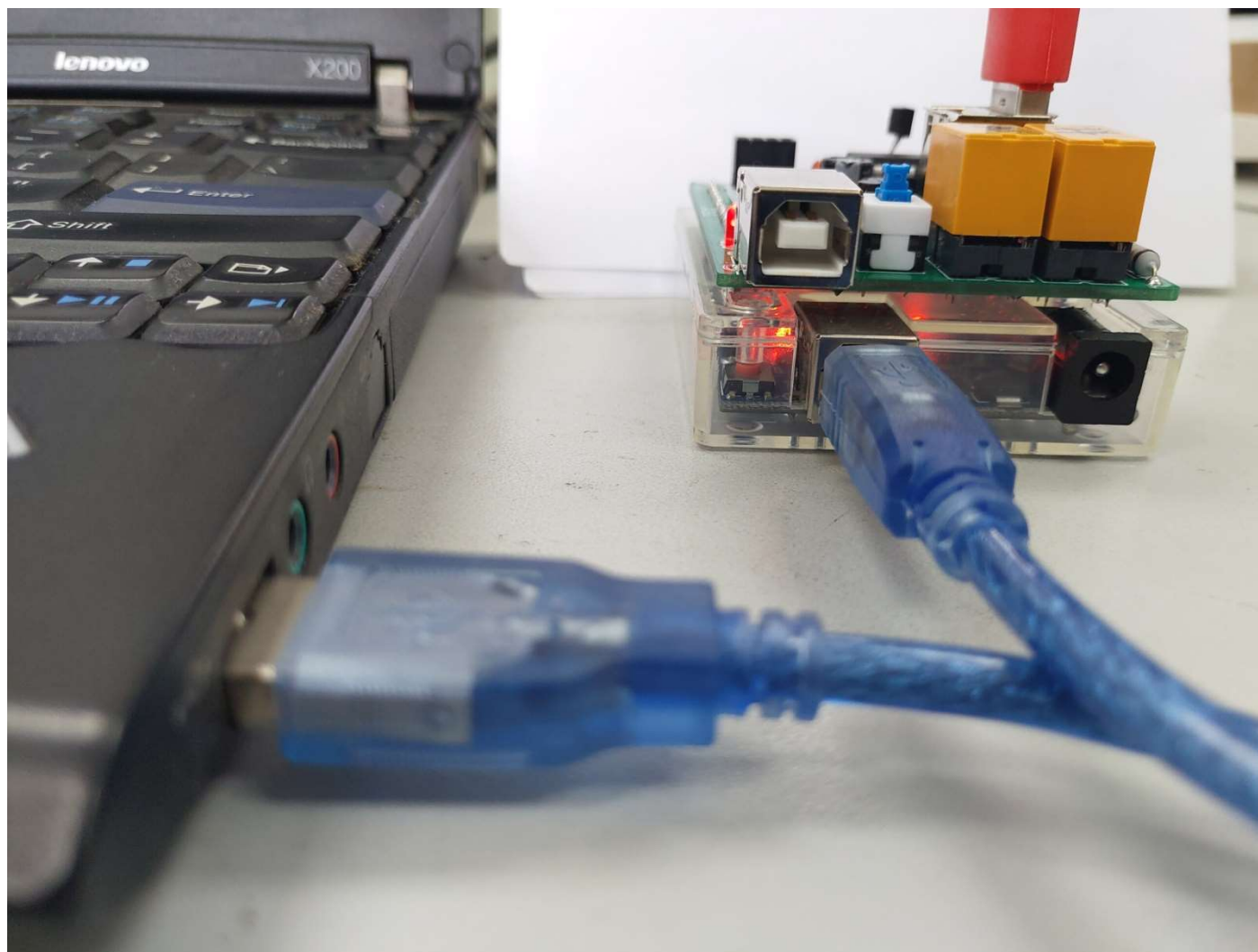
组网测试方案3（不推荐）

- （实用现场方案）2个教室分用2个路由器，在程序代码中需修改变量
ssid="SMART_SOCKET_420"(或"SMTSCKT_420")，
以及ssid="SMART_SOCKET_421"
(或"SMTSCKT_421")。它们分别对应420和421两个教室的热点，各房间注意不要连错！密码两个教室相同，即程序代码中的变量
password="12345678"。所有智能插座与手机wifi均通过其实现组网。在自己的安卓手机上安装我们提供的app。不同组之间通过组号来区分，详见智能插座固件代码内的组号设置与手机app使用时的组号设置。请务必使用自己的座位组号，并连接自己教室的路由器，且每组仅一台手机安装运行app连接你的智能插座。切记！

智能插座固件代码的上传

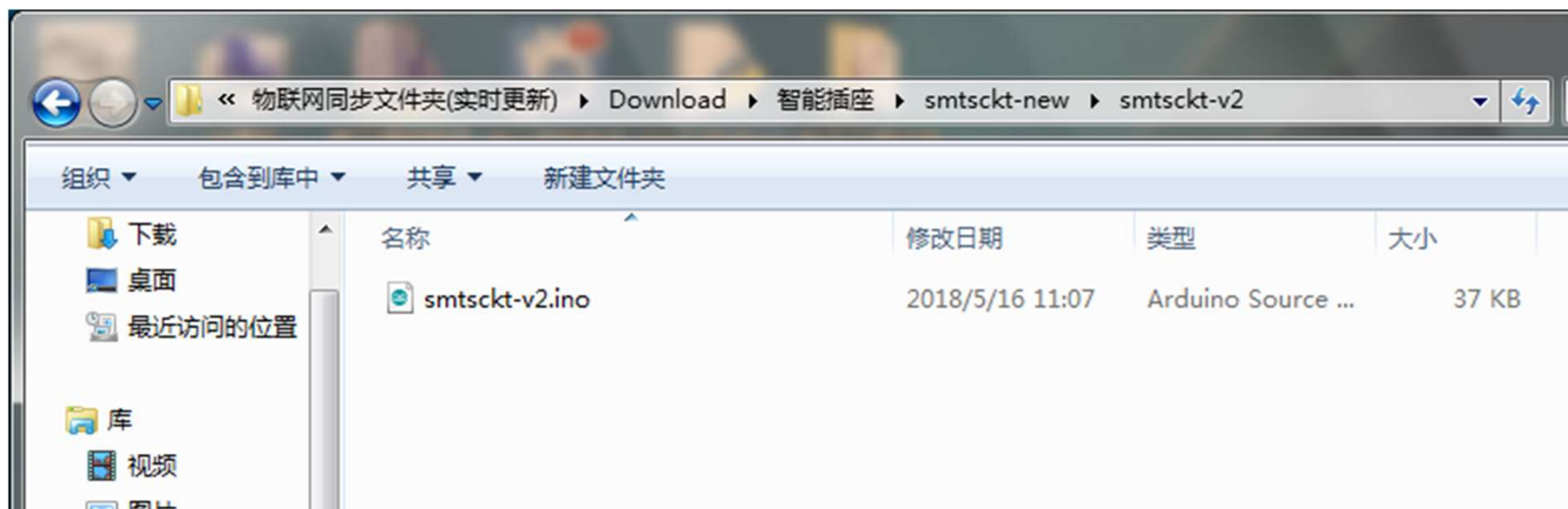
- 在智能插座系统硬件置于“**上传状态**”，将课程提供的**sketch**代码上传到你的智能插座板中。
- 该**sketch**代码名称为：**smtsckt-v4.ino**，一般在你D盘的课程相关目录中。
- **smtsckt-v4.ino**为实现智能插座所有设计功能的最终完整固件源代码。

智能插座固件代码的上传



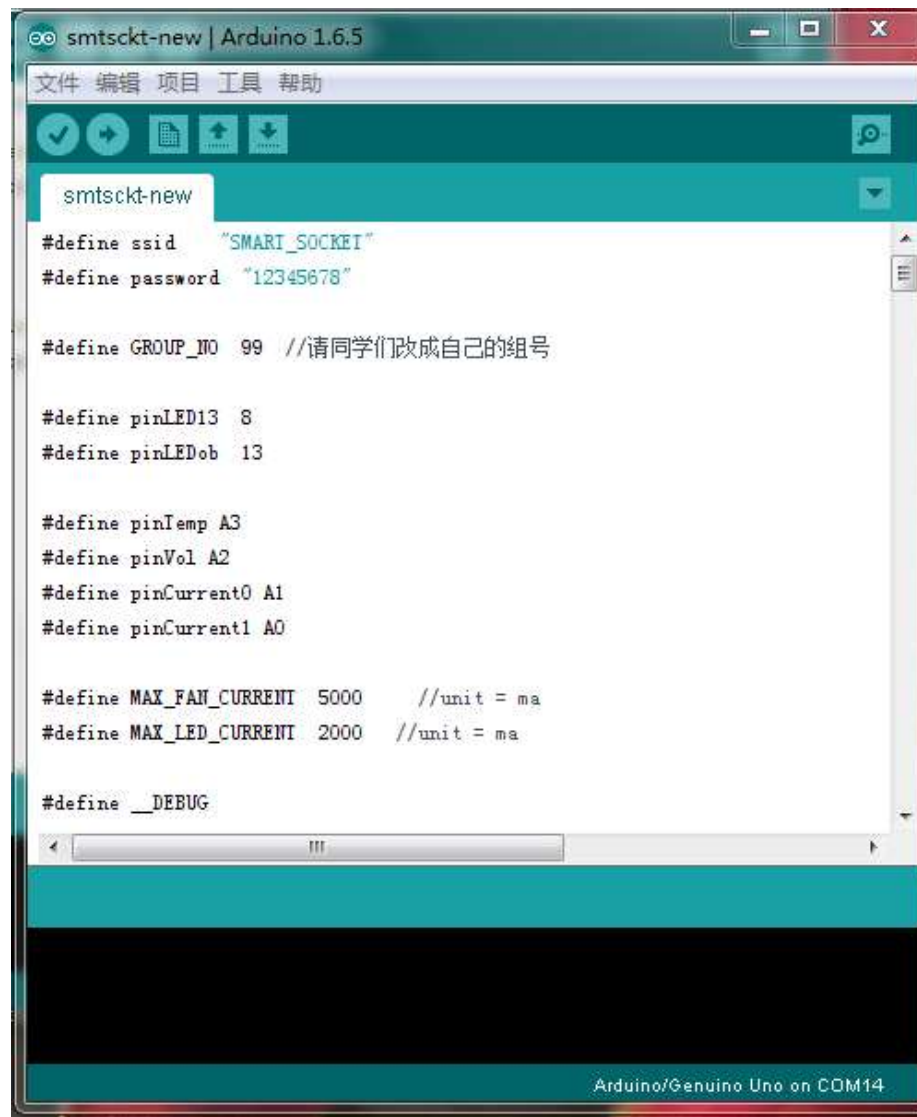
智能插座固件源代码文件

- 双击之即可在Arduino IDE中打开该源代码文件——**smtsckt-v4.ino**



使用方案2时的 智能插座固件代码的修改

- 在Arduino IDE中，打开项目SmtSckt-v4。
- 将程序中的ssid（即图中的“SMART_SOCKET”）和password（即图中的“12345678”）改成你刚才在小白路由器上所设置的ssid和密码那样



```
smtsckt-new | Arduino 1.6.5
文件 编辑 项目 工具 帮助

smtsckt-new

#define ssid "SMART_SOCKET"
#define password "12345678"

#define GROUP_NO 99 //请同学们改成自己的组号

#define pinLED13 8
#define pinLEDo 13

#define pinTemp A3
#define pinVol A2
#define pinCurrent0 A1
#define pinCurrent1 A0

#define MAX_FAN_CURRENT 5000 //unit = ma
#define MAX_LED_CURRENT 2000 //unit = ma

#define __DEBUG

Arduino/Genuino Uno on COM14
```

使用方案2时的 小白路由器的配置

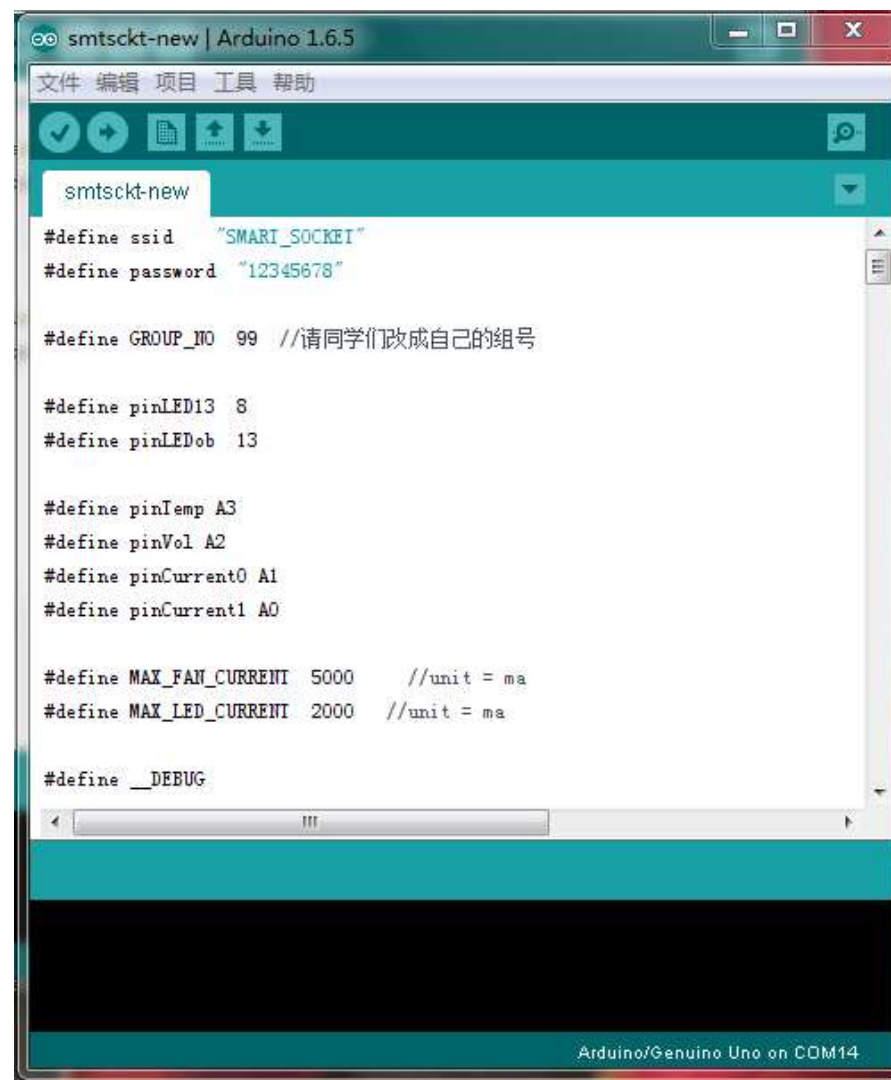
- 利用之前课程所学，先将你的小白路由器恢复成出厂设置，然后将小白路由器的SSID设置成“xiaobai_yyy_xx”，其中yyy为420或421房号，xx为你的桌号；并设置小白路由器的密码（可考虑8个字符长度的数字串），以保证实验中各组网络通信不相互干扰。

使用方案1时的 智能插座固件代码的修改

- 打开你安卓手机的热点，记下其ssid和密码。
代码修改与方案2类似。

使用方案3（不推荐）时的 智能插座固件代码的修改

- 在Arduino IDE中，打开项目smtsckt-v4。
- 将程序中的ssid（即图中的“SMART_SOCKET”）和password（即图中的“12345678”）改成你房间的无线热点连接配置。
- 将程序中的GROUP_NO修改成你的座位组号。



```
smtsckt-new | Arduino 1.6.5
文件 编辑 项目 工具 帮助

smtsckt-new

#define ssid "SMART_SOCKET"
#define password "12345678"

#define GROUP_NO 99 //请同学们改成自己的组号

#define pinLED13 8
#define pinLED0b 13

#define pinTemp A3
#define pinVol A2
#define pinCurrent0 A1
#define pinCurrent1 A0

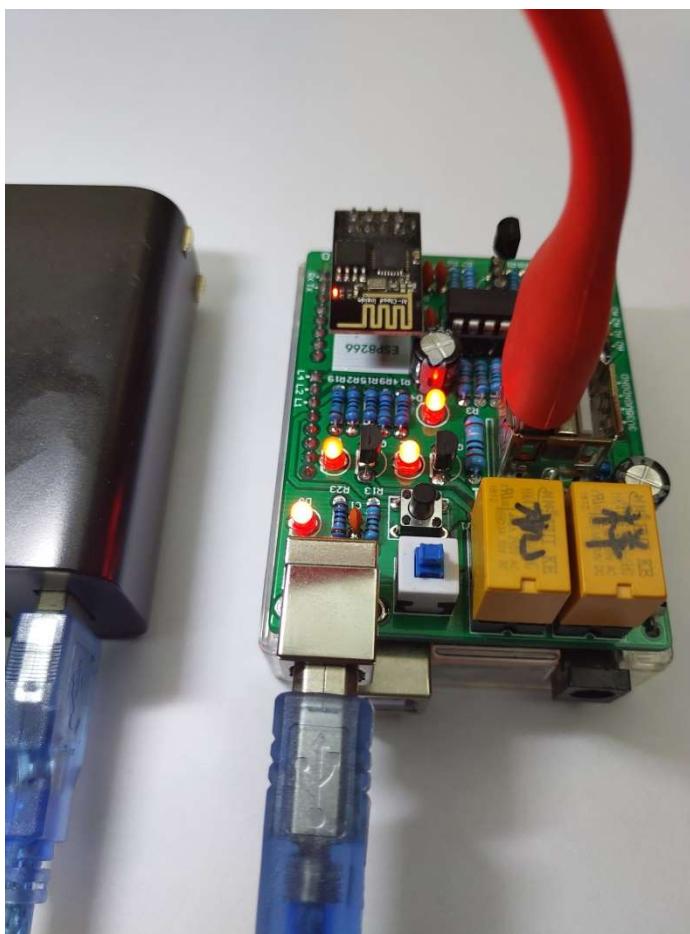
#define MAX_FAN_CURRENT 5000 //unit = ma
#define MAX_LED_CURRENT 2000 //unit = ma

#define __DEBUG

Arduino/Genuino Uno on COM14
```

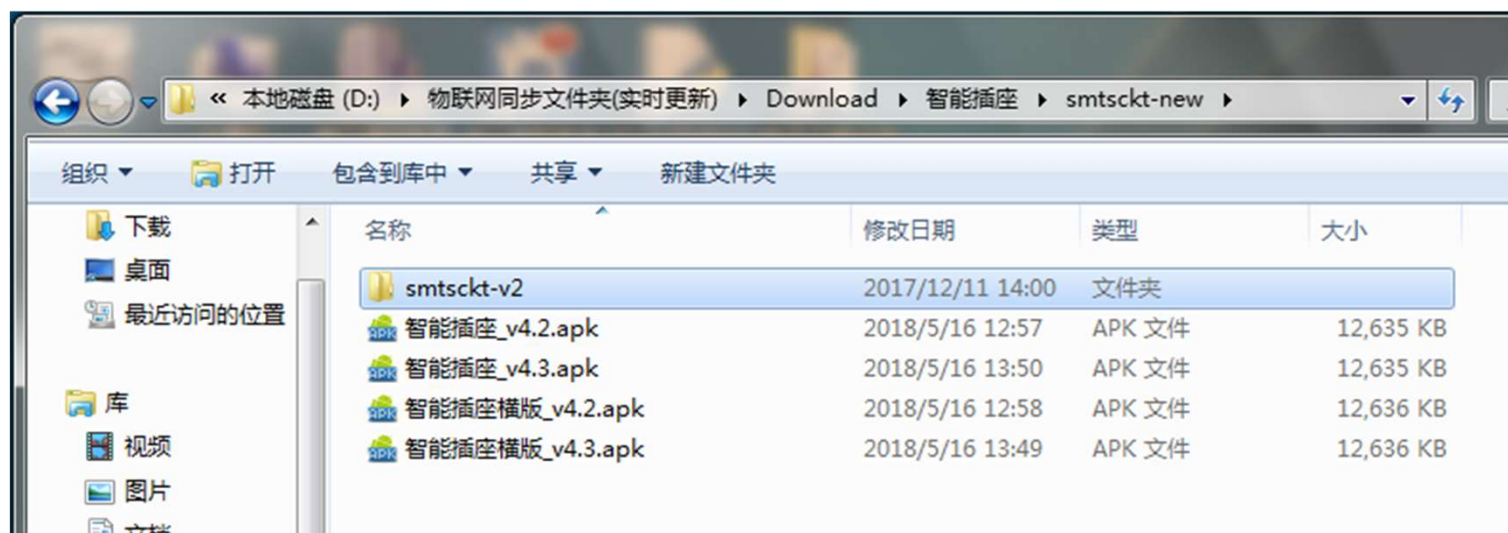
智能插座硬件系统的运行

- 智能插座系统的“运行状态”如下图所示：



在手机或小白路由器上安装APP

- 在手机上安装App: 智能插座_4.3.apk
- 或者在小白路由器上安装App: 智能插座横版_4.3.apk (注: 另一v4.2版是用于较旧的手机安卓系统, 且无语音调试功能)
- 注意: 每组仅允许一台手机运行app连接其智能插座。多台手机同时连接智能插座可能导致不可预料的错误!



智能插座综合功能系统测试

- 置智能插座硬件系统于“运行状态”下，上电启动智能插座；
- 在手机或小白路由器上启动运行App。
- 如你选择方案3组网方式（不推荐），则你的手机应附着到教室的公共热点（ssid见本课件前面内容，密码12345678），并在App中设置你的座位组号。

重要提示

- 手机WLAN热点的AP频段必须设置在2.4GHz频段上。
- 建议运行智能插座app时，建议关闭移动网络流量。
- 智能插座上电运行时，各指示灯会出现闪烁现象，这是其内部固件的有意设计，以便前期调试时了解固件的运行状况，大家测试时不必理会。当D1、D2和D4均稳定地持续亮起，一般表明智能插座已连上热点，但此时还未和app连上。
- 当app界面正中央的“1”变由灰变绿，且界面底部的进度条从左完全滚动显示至右时，则表明app与智能插座连上。
- 因为实验过程中教室里无线设备较多，且开始时会有很多人同时在进行无线通信连接，可能会造成部分设备通信受阻，请大家在首次连接时保持一定的耐心，稍稍多等待一些时间。

如何进行系统测试？

- 探索尝试App中的全部各项功能，注意各功能的操作步骤。
- 预先设计好针对各项功能的测试步骤和测试场景（或数据），实际检验你发现的各项功能的运行情况（要求说明你的各项测试的目的，所采用的测试步骤，采用了那些测试场景或测试数据，结果成功与否；是否存在问题；如果存在问题，分析问题存在的原因.....）

通用系统测试方法介绍

- 我们在进行一个系统测试时需要让你的测试工作尽可能覆盖到被测系统的各方面，这就是评价测试工作完备性的一一测试覆盖率。
- 我们在开始动手测试一个系统之前，会提前针对每项测试准备一个或多个测试步骤，准备好一套或多套的输入数据，并预测系统正常情况下的执行状态或输出结果，以便实际测试时与实际情况相对照，判断系统执行正确与否。这种测试前就准备好的测试步骤或输入数据被称为：**测试用例**。
- 从系统测试效率来看，我们期望**以最少的测试用例达到最全面的测试覆盖率**。这就要求对你的测试用例精心筛选，以达到较高的系统测试效率。
- **测试用例的选择设计，反应了一个产品测试的完善程度和测试效率，同时也体现了一个测试工程师的实际工作经验与测试水平。**

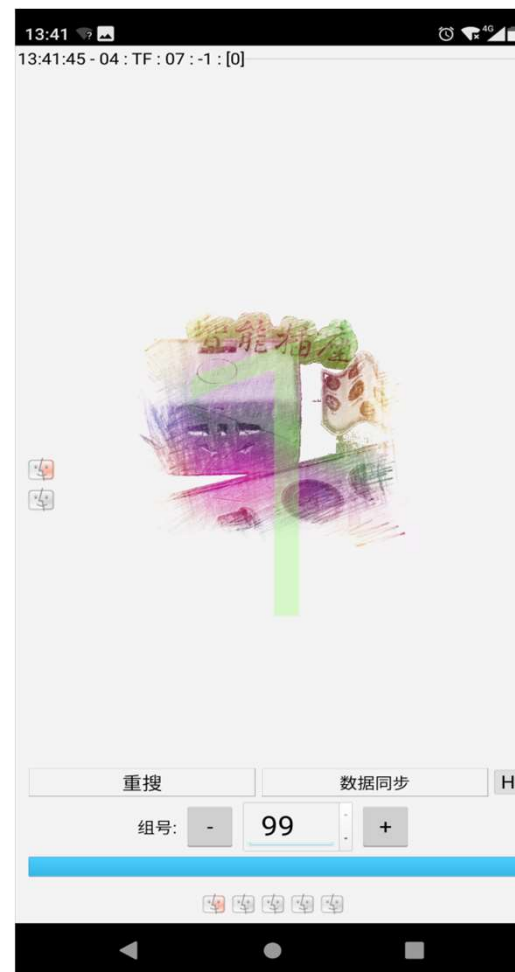
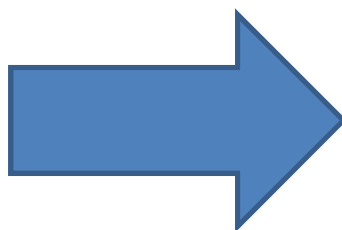
系统测试过程提示

- 提示：系统有如下功能：
 1. 我们的智能插座可连一个手机或小白路由器。
 2. 智能插座具备定时开/关、延时开/关。
 3. 针对超/欠电压、超电流、超功率告警并自动断开插座供电。
 4. 插座开关手动控制，以及温度、电压、电流和功率显示。
 5. 与智能插座的通信连接功能按钮——“重搜”按钮，是先发断开命令，再发连接请求命令。“数据同步”是命令智能插座将其当前的状态和参数全部发往app，以便app和智能插座上的状态一致（用于通信不稳定场合或中断后重新联机用）。注：该第5项主要是我们研发时使用的小工具，一般不随最终的量产品向终端用户交付，故可不写入测试报告中。

智能插座App中的各操作界面



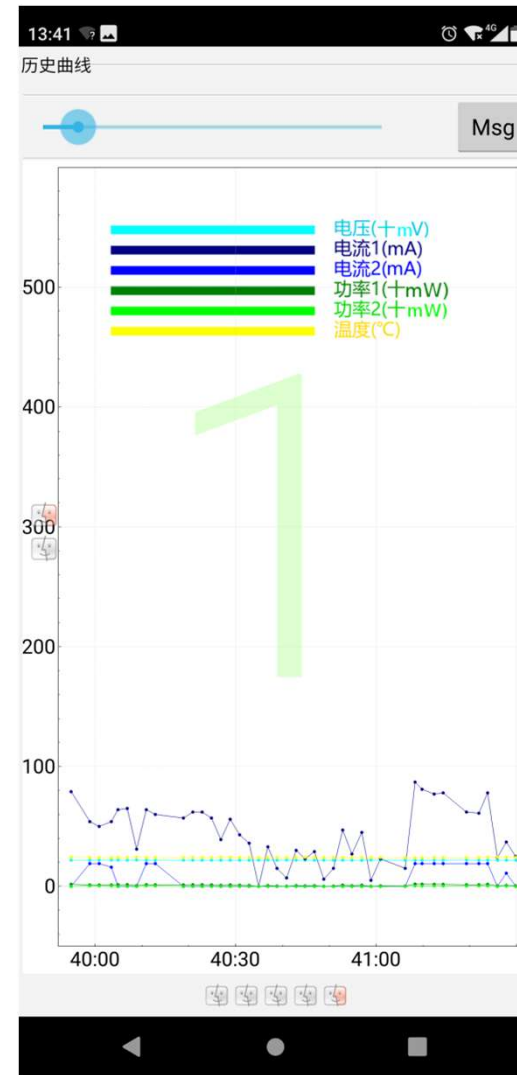
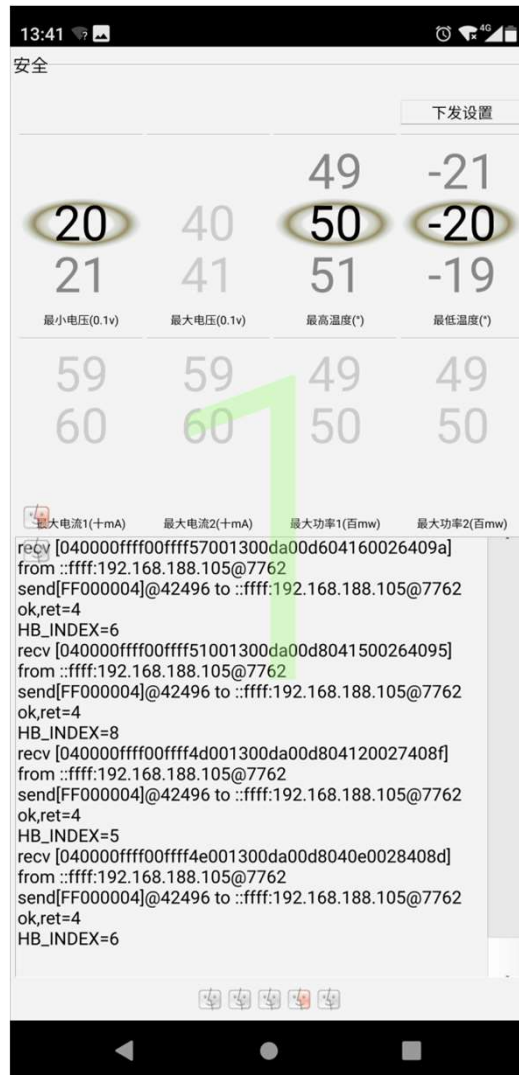
APP连
上智
能插
座后



智能插座App中的各操作界面



智能插座App中的各操作界面



注意事项！

- 注意本系统测试时，外接电源应接上板的**USB-B**电源接口，否则，测试中用到超、欠电压状态模拟发生电路将不会起作用。
- 在系统测试期间，如需重新上传固件代码，到你的智能插座硬件中，请务必在“上传模式”下进行，上传完成后，再回到“运行模式”。切记！


关于系统测试报告

- 根据上述的系统测试过程，撰写系统功能测试报告。报告包括针对**APP**中所有功能的描述，以及你的测试过程和测试用例（有了这部分描述，别人就能轻松地事后重现你做发现的**Bug**），给出测试结果。如发现有**Bug**存在，请描述之。
- 选做，新增一项功能：开机时按下**s1**按钮则将**ssid**和密码恢复到一个默认配置状态。要求修改代码实现之；并设计该项修改后的测试过程与测试用例，记录展示测试结果。报告中附上所修改的代码片段。
- 选做，是否有**Bug**（如果你发现有的话）？你能修改吗？如果你能修改的，试修改之。并在报告中附上相应的修改代码段。



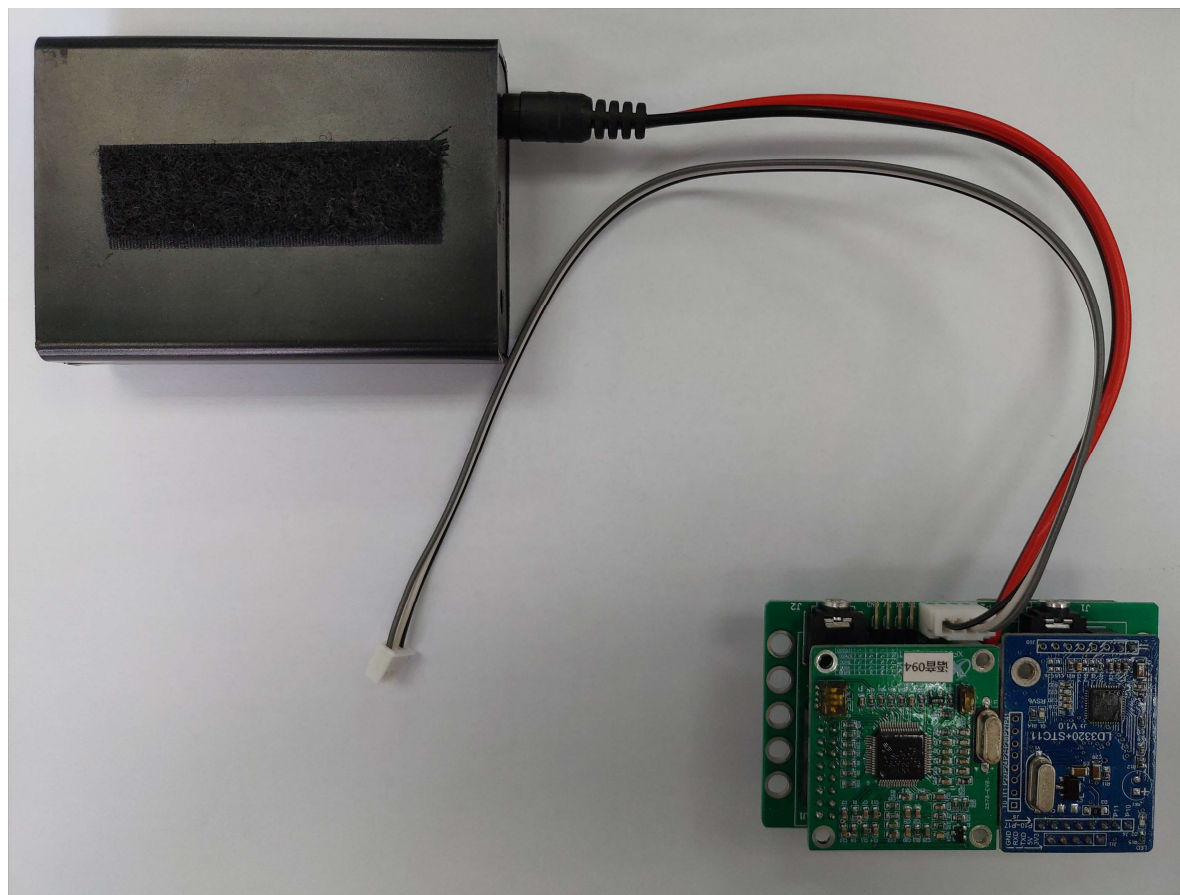
系统测试报告要求写入本项目的总结报告并一起提交。

再次提醒：关于本训练项目的总结报告

- 本训练项目的所有实验内容请写入同一份总结报告中，并一次性提交到“学在浙大”上的相应作业提交链接处。
- 要求按组上传pdf电子稿。上传文件名为“智能插座-421(或420)-周X上午(或下午)-组号-姓名1学号1-姓名2学号2-Y”，Y为你的版本号。上传“学在浙大”。
- 报告中列出你电装过程中的安装调试顺序，说明你采用该顺序的原因。总结该过程中出现的问题（如果有的话）以及获得的经验教训。
- 根据各课件中的各项测试要求（包括一些要求写入本报告的思考问答题），列出你在电装、调试以及系统测试过程中测得的各项数据列表、测试过程或者相应的程序代码片段（一般会在课件中会以“”标出）。

附：一种应对联网失败的系统调试方式

- 领取一个外接语音模块



附：一种应对联网失败的系统调试方式

- 将该语音模块与智能插座的JP1接口相连，正常启动系统各部件，则该语音模块将在智能插座运行时给出各运行状态提示，你就能通过语音了解当前系统的联网进程了。

