

15 分钟制作智能插座（串口接入方式）

版本	编写/修订说明	修订人	修订日期	备注
2.0.0	创建文档	王科岩	20180605	

目录

目录.....	2
1 概要.....	4
2 软硬件准备.....	4
2.1 云平台账号.....	4
2.2 嵌入式开发套件.....	5
2.3 APP.....	5
2.3.1 调试 APP.....	5
2.3.2 插座 APP.....	6
3 产品开发.....	7
3.1 新建产品.....	7
3.2 添加功能点.....	8
3.3 设备开发.....	10
3.3.1 获取产品 ID 及秘钥.....	10
3.3.2 修改源码.....	10
4 功能调试.....	14

4.1 创建应用.....	14
4.2 APP 登录.....	15
4.3 设备绑定.....	16
4.3.1 wifi 设备/有线设备.....	16
4.3.2 2G/3G/4G/NB-IOT 设备.....	18
4.4 设备调试.....	19
5 插座 APP.....	21
5.1 开关.....	21
5.2 定时.....	21
5.3 分享.....	22

1 概要

青莲云作为领先的物联网安全服务商，为了让开发者不必关心数据的加密传输、网络链路的安全通信，适配了一系列 wifi、蓝牙、NB-IOT 模组。青莲云提供不同平台的嵌入式 SDK，开发者可以通过简单的调试，立即拥有强大的后端云能力，专注于具体业务研发。

本文档描述了如何使用青莲云提供的服务制作一款安全联网智能插座。开发者利用自身熟悉的平台，仅需 15 分钟，即可制作一款功能强大的智能插座。

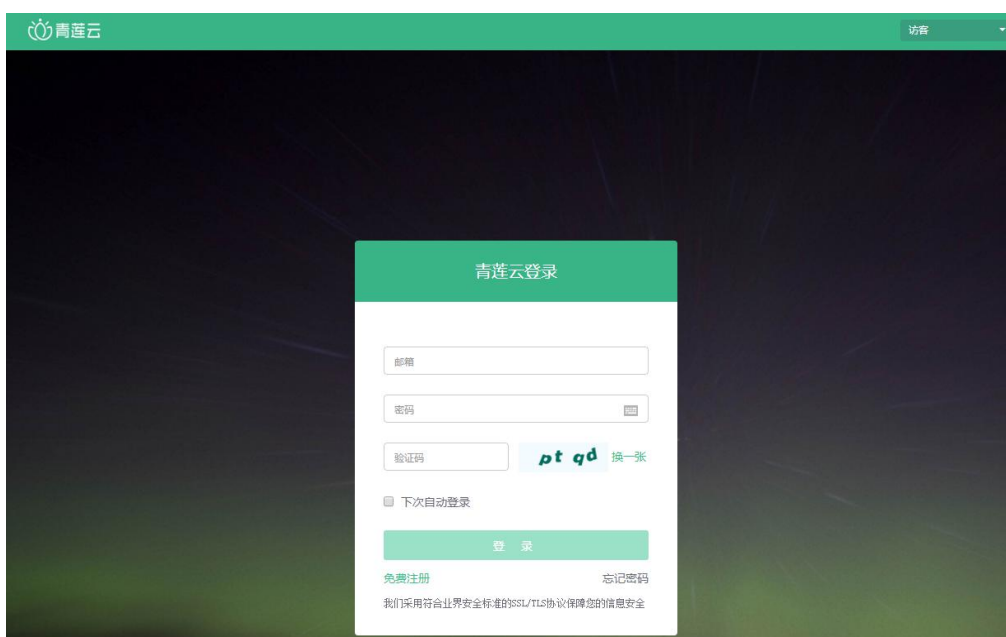
插座功能如下：

- 1) 安全快速联网
- 2) 本地按键控制插座开关
- 3) 远程控制开关
- 4) 定时控制开关
- 5) 设备分享

2 软硬件准备

2.1 云平台账号

登录青莲云官网 <https://www.qinglianyun.com/>，注册、登录账号



2.2 嵌入式开发套件

选择熟悉的硬件平台，到 <https://www.qinglianyun.com/Home/Doc/index> 下载相应的固件和 MCU 示例。本文以 STM32 处理器+ESP8266 模组为例。

模组整机固件（供串口接入使用）

基于通信模组的固件，适用于MCU+模组的硬件方案。开发者使用时只需MCU支持串口即可使用青莲云的服务。

固件



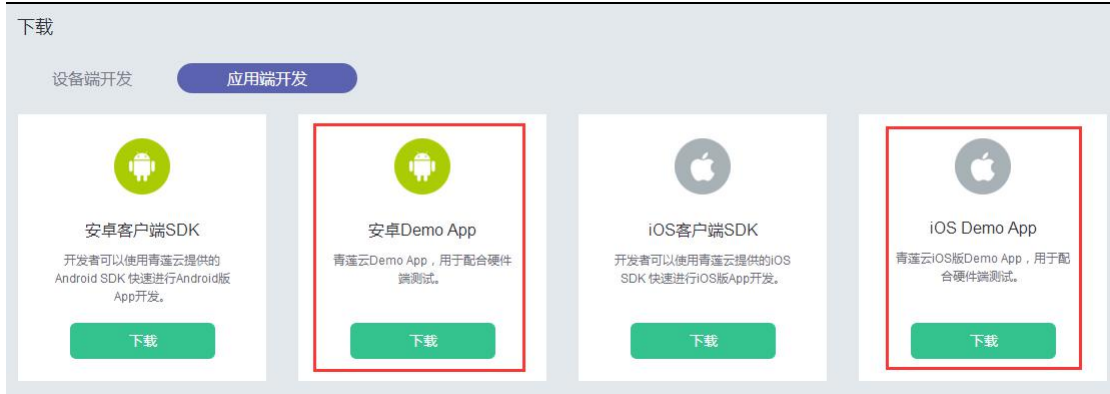
MCU示例



2.3 APP

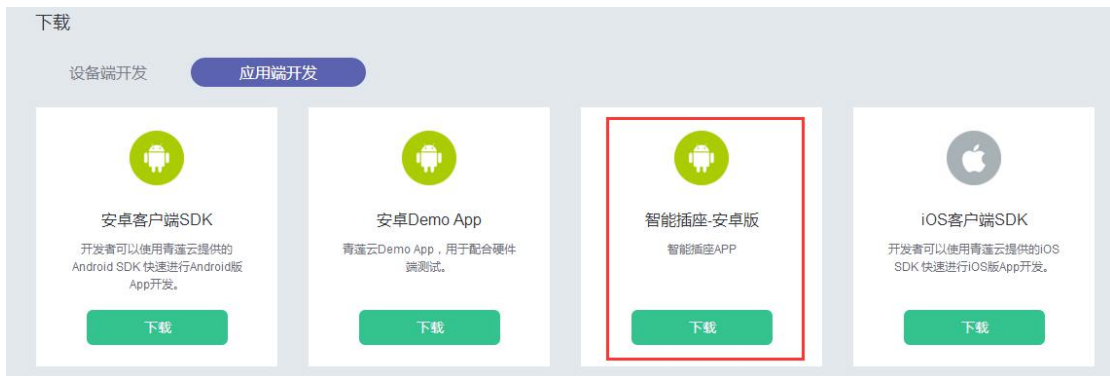
2.3.1 调试 APP

到 <https://www.qinglianyun.com/Home/Doc/mobdownload> 下载调试 APP，安卓或苹果版本都可以。本文以安卓为例。



2.3.2 插座 APP

调试完成, 到 <https://www.qinglianyun.com/Home/Doc/mobdownload> 下载插座APP 进行测试。



3 产品开发

3.1 新建产品

进入青连云控制台，添加一个智能插座产品。

The screenshot displays the Qinglian Cloud (青连云) user interface. At the top, there is a navigation bar with links for '首页' (Home), '安全产品和服务' (Security Products and Services), '行业解决方案' (Industry Solutions), '平台基础能力' (Platform Basic Capabilities), '文档与下载' (Documents and Downloads), and '关于我们' (About Us). A red box highlights the '进入控制台' (Enter Console) button in the top right corner.

The main content area features a large banner for the '正式发布《2017智能硬件安全白皮书》' (Official Release of '2017 Smart Hardware Security White Paper'). Below the banner is a '点击下载' (Click to Download) button.

Below the banner is a navigation bar with '授权管理' (Authorization Management), '下载' (Download), '文档' (Documents), and '测试账号' (Test Account). A sidebar on the left contains '产品中心' (Product Center) with sub-items: '产品概览' (Product Overview), '产品功能' (Product Features), '告警设置' (Alert Settings), and '设备中心' (Device Center).

The main content area shows a '产品列表' (Product List) table with columns: '产品名称' (Product Name), '产品ID' (Product ID), '设备类型' (Device Type), '产品型号' (Product Model), '功能点数量' (Number of Features), '设备包数' (Number of Device Packages), '接入设备数' (Number of Connected Devices), '创建时间' (Creation Time), and '操作' (Operations). A red box highlights the '+ 添加产品' (Add Product) button in the top right of the table.

Below the table is the '添加新产品' (Add New Product) form. It consists of three steps: 1. '添加一个新的智能硬件' (Add a new smart hardware), 2. '添加功能点' (Add features), and 3. '完成' (Complete). The first step is active and highlighted with a red box. The form fields are:

- 产品名称 (Product Name): 智能插座 (Smart Plug)
- 产品型号 (Product Model): qly_00001
- 接入方式 (Access Method): 直连青连云 (Directly connect to Qinglian Cloud)
- 设备类型 (Device Type): 智能家电 (Smart Home Appliances)
- 网络类型 (Network Type): WIFI
- 产品描述 (Product Description): 基于ESP8266 (Based on ESP8266)

A '下一步' (Next Step) button is located at the bottom of the form.

3.2 添加功能点

插座只有一个开关的功能，添加一个功能点即可，点击“产品功能”，添加功能点。添加功能点完成后，记录功能点的编号，后面做设备端开发时会用到相应编号。



新建产品功能

功能名称

开关

*用于描述该功能的中文说明，如：开关、增大音量

变量名

dp_switch

*该功能对应的程序变量名，此处填写的变量名应和MCU开发中的变量名保持一致，如：switch

传输类型

可上报可下发

*可上报可下发：设备数据可上报至云端，云端指令可下发给设备
只上报：只支持设备数据上报至云端
只下发：只支持云端指令下发给设备

数据类型

布尔型

*用于定义非真即假的二值型变量，如开关，0代表关、1代表开

真值描述

开

假值描述

关

保存

取消

产品中心

产品功能

您可在该页面设置产品的功能点。例如灯的开关，颜色，传感器的温湿度等。

+ 添加功能点 + 导入功能点 导出功能点

编号	功能名称	变量名	类型	数据类型	描述	操作
1	开关	dp_switch	可上报可下发	布尔型(1)	0关,1开	编辑 删除

当前第1页, 共1页

1 > >> GO

这个编号设备开发会用到

3.3 设备开发

3.3.1 获取产品 ID 及密钥

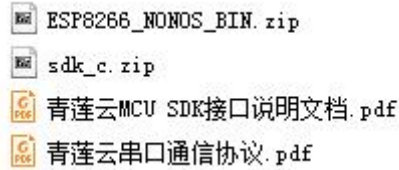
到“产品概览”中查看创建的产品 ID 及产品密钥，这是确保设备能够合法入网的标识符，需要写到设备端的代码中，记录下这两个值。



3.3.2 修改源码

解压下载好的嵌入式开发套件压缩包。

ESP8266 文件夹包含如下内容。ESP8266_NONOS_BIN.zip 包含模组固件及烧录说明，请按文档说明烧录固件。sdk_c.zip 为青连云提供的串口协议的 MCU SDK，具体实现的功能及接口的调用方法可以参考《青连云 MCU SDK 接口说明文档.pdf》。如果使用的功能较少，为了节省 MCU 资源也可以参考《青连云串口通信协议.pdf》自行实现串口协议的解析处理。



STM32 文件夹包含一个完整的例子工程，目录结构如下所示。其中 sdk_c 为青莲云提供的串口协议的 MCU SDK 源码，USER/QLY 目录包含 SDK 的接口文件。



需要修改 2 个文件

◆ main.c

- 1) SDK 中的示例代码用于快速理解各接口的调用方法，这里将例子中定时主动上传数据的代码注释掉。

```
if(0x04 == g_esp8266_status)
{
    if((++count_m)%0x0007FFFF == 0) /* timer */
    {
#if 0
        if(0 == test_type) /* dp test */
        {
            qlcloud_data_up();
            test_type = 1;
        }
        else if(1 == test_type) /* push test */
        {
            qlcloud_custom_data_up();
            test_type = 2;
        }
        else /* throughput test */
        {
            qlcloud_data_up_max();
            test_type = 0;
        }
    }
#endif
}
```

◆ qlcloud_interface.c

插座可以通过本地按键控制开关，也可以通过 APP 远程控制开关，这两种方式都会触发本地开关状态改变，并将开关状态上报至云端。

- 1) qlcloud_interface.c 起始位置有对产品 ID 和产品密钥的定义，替换成 3.3.1 节从官网获取的，注意，替换产品密钥时，需将官网的一串字符串转换成相应的十六进制编码，即在每个字节前增加 0x 作为开头。如密钥开头是 7879，替换时应改为 0x78, 0x79。

```
uint32_t PRODUCT_ID = 14;  
uint8_t PRODUCT_KEY[] = {0x78, 0x79, 0x78, 0x79, 0x78, 0x79, 0x78, 0x79, 0x78, 0x79, 0x78, 0x79, 0x78, 0x79, 0x78, 0x79};  
uint8_t MCU_VER[] = "01.16";
```

- 2) 控制插座的开关需根据具体的硬件电路实现，本例中使用 g_switch 变量模拟插座的打开和关闭。

```
uint8_t g_switch = 0; /* 0: off, 1 on*/  
  
uint8_t get_switch_status()  
{  
    return g_switch;  
}  
  
void set_switch_status(uint8_t on_off)  
{  
    g_switch = on_off;  
}
```

- 3) 实现开启/关闭插座的函数。

首先设置插座的开关。

接着将此状态上报至云端，注意一定要在设备状态改变后，上报一条最新值。

上报数据时，功能点编号要与 3.2 节添加的功能点编号对应。

```
void switch_on_off(uint8_t on_off)
{
    uint8_t seq;

    set_switch_status(on_off);

    dp_up_add_boo(1, on_off);

    qlcloud_upload_dps(&seq);
}
```

4) 实现本地按键控制及 APP 控制功能

```
void dp_down_handle_switch(uint8_t in_data, uint32_t in_len)
{
    uint8_t on_off = bytes_to_bool(in_data);

    switch_on_off(on_off);

    if(0 == on_off)
    {
        printf("DP \"switch\" off\r\n");
    }
    else
    {
        printf("DP \"switch\" on\r\n");
    }
}
```

4 功能调试

4.1 创建应用

不同的 APP 可以绑定、控制不同的产品，青连云对此功能做了权限控制。此功能适用于同一个厂商开发了多款 APP，比如 APP1 可以控制某款插座，APP2 可以控制某款空调。应用授权即给予 APP（应用）对某产品的控制权。

1) 进入控制台，点击“授权管理”。



2) 选择需要授权的 APP，如还未添加，请先添加应用 APP。



3) 添加应用时可以勾选该应用可控制的产品。



4) 保存后可以看到添加了一款 APP

APP ID 及 APP Token 用于标示 APP，后面我们会用到；

包含产品是上一个步骤勾选过的产品，即此 APP 可控制的产品；

二维码包含了 APP ID 和 APP Token 及一些其他信息；

当需要添加或删除包含的产品时，可以选择编辑该应用。



4.2 APP 登录

- 1) 打开调试 APP，首先输入 APP ID 和 APP Token，可以手动输入到文本框，也可以通过右上角扫码，扫描上一个步骤提到的二维码。

调试App 扫码

调试准备：

- 1、登陆青莲云官网www.qinglianyun.com，点击进入控制台
- 2、在产品列表页，添加产品，如：智能灯
- 3、在“授权管理”的“应用授权”中，添加应用，将需要用该应用控制的产品添加到应用
- 4、将应用的ID和token分别输入到上面的输入框，即可开始调试

开始调试

- 2) 用手机号或微信注册登录



4.3 设备绑定

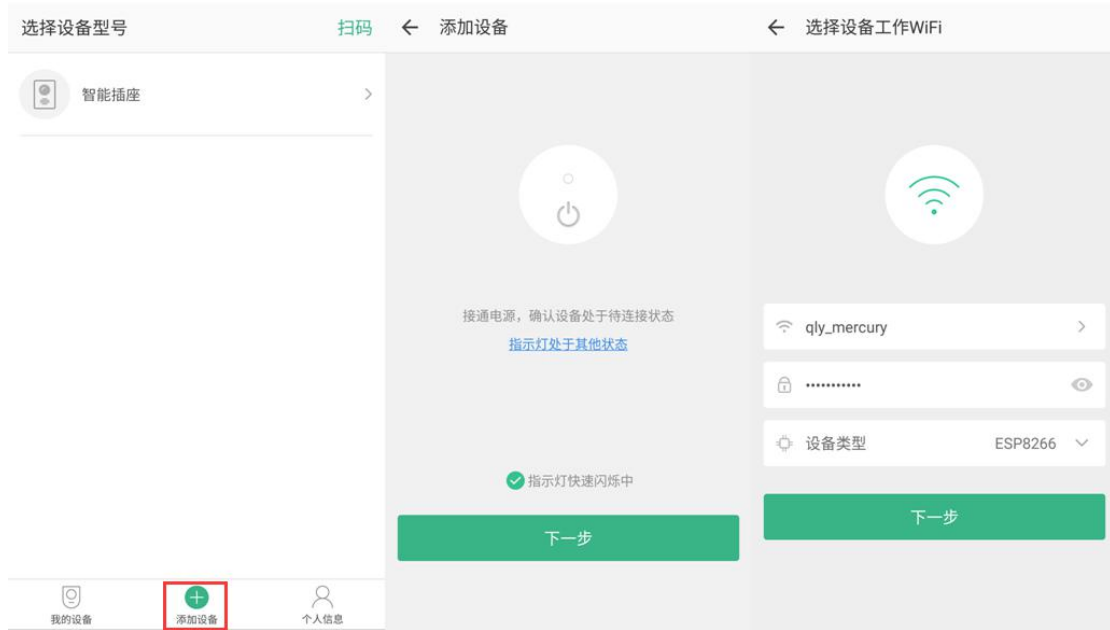
4.3.1 wifi 设备/有线设备

此类设备由 MAC 地址作为设备标识符，支持局域网、扫码两种方式进行设备绑定

◆ 局域网绑定

手机要连接到 wifi 路由器上，与设备在同一局域网中

添加设备，输入 wifi 路由器的 SSID 和密码



绑定成功，可以在“我的设备”找到绑定的设备，左侧的小绿点表示设备在线。

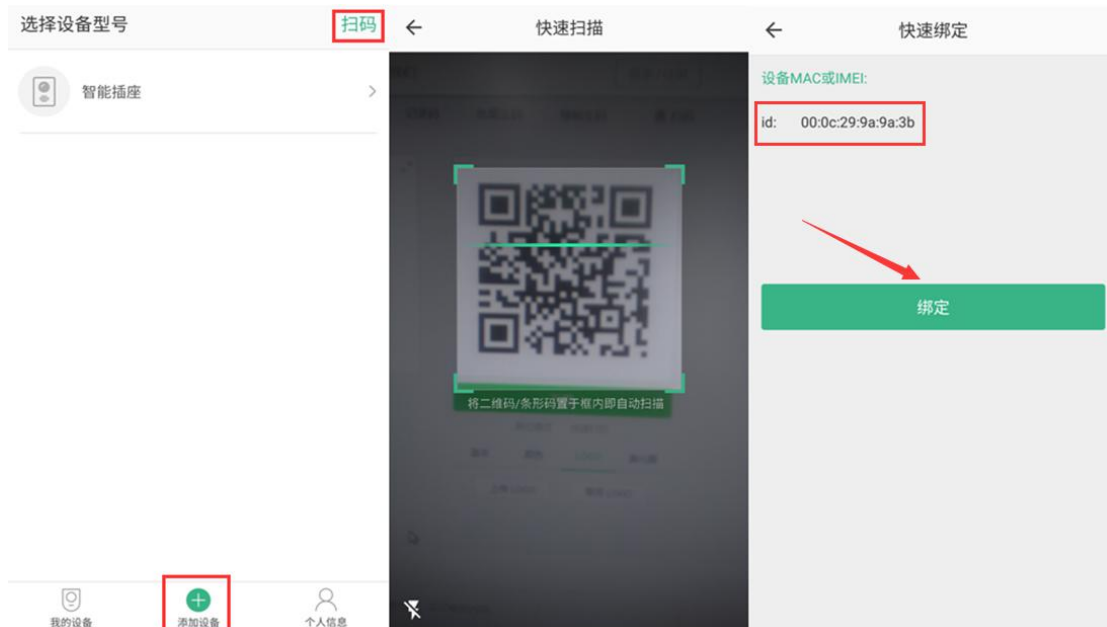


◆ 扫码绑定

需要提供设备 MAC 地址对应的二维码，可以到在线生成二维码的网站生成二维码，本文使用的是 <https://cli.im/> 来生成，比如要生成 MAC 地址为 00:0C:29:7C:46:EF 的二维码



注意生成二维码时，MAC 地址要加冒号作为间隔符。接着可以扫描生成的二维码进行绑定设备。



4.3.2 2G/3G/4G/NB-IOT 设备

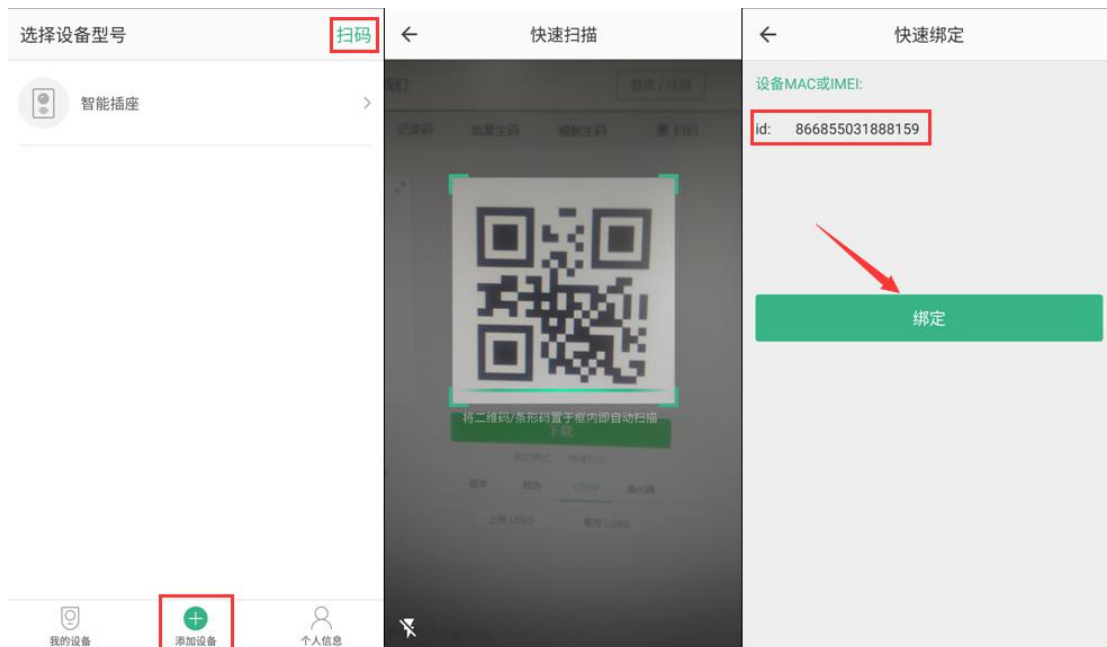
此类设备由设备的 IMEI 码作为设备标识符，仅支持扫码方式进行设备绑定

a) 扫码绑定

需要提供设备 IMEI 码对应的二维码，可以到在线生成二维码的网站生成二维码，本文使用的是 <https://cli.im/> 来生成，比如要生成 IMEI 为 866855031888159 的二维码

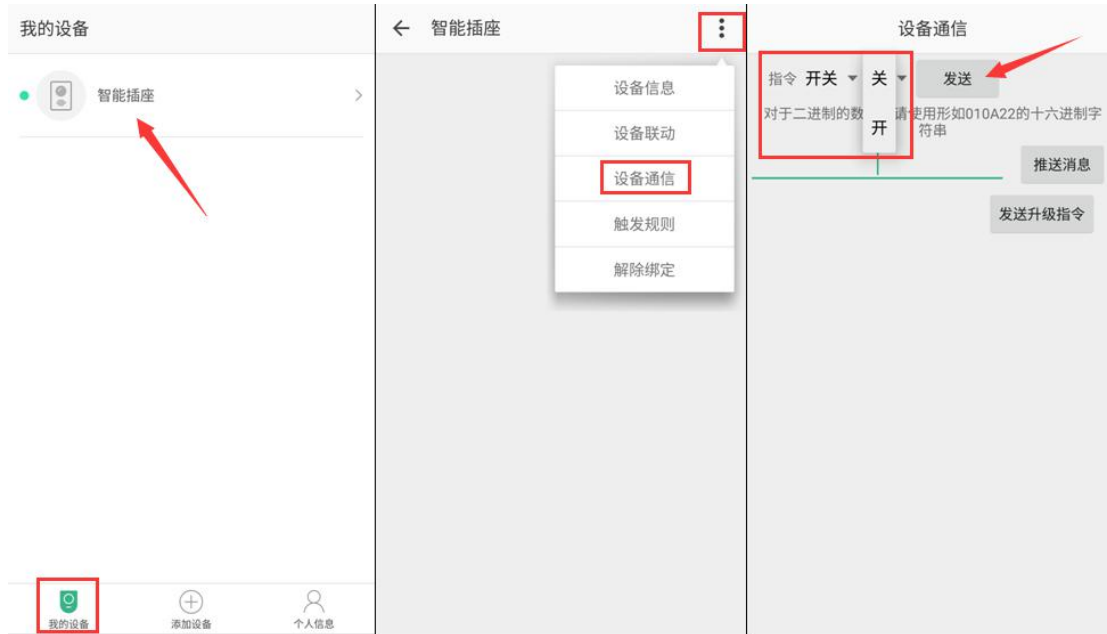


注意生成二维码时，直接输入设备的IMEI码生成即可。接着可以扫描生成的二维码进行绑定设备。



4.4 设备调试

在“我的设备”找到已经绑定的设备，点击右上角“设备通信”，可以看到此设备支持的指令，选择“开关”指令，点击“发送”，可以看到设备已经收到相应命令了。



```

RX ver:02, type:12, len:5
DP "switch" on
RX ver:02, type:12, len:5
DP "switch" off
RX ver:02, type:12, len:5
DP "switch" on
RX ver:02, type:12, len:5
DP "switch" off
    
```

DP "switch" on → 开启命令
DP "switch" off → 关闭命令

设备在收到开启/关闭指令后，先执行硬件操作，接着把开启/关闭状态回传至云端。

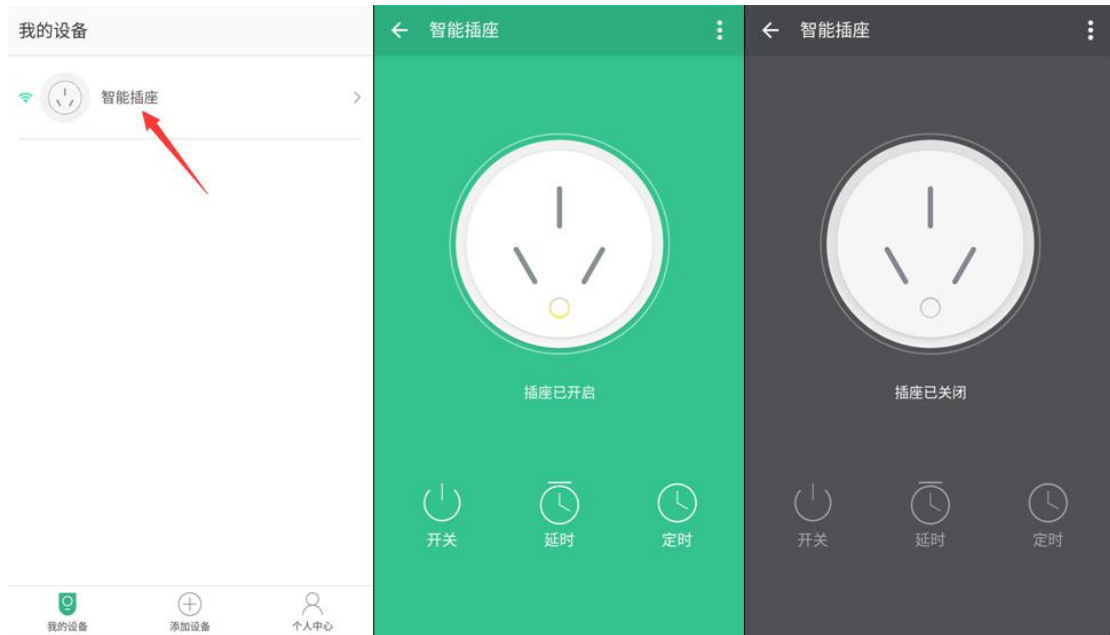
APP 把收到设备回复的数据，及收到数据的时间点打印了出来。



5 插座 APP

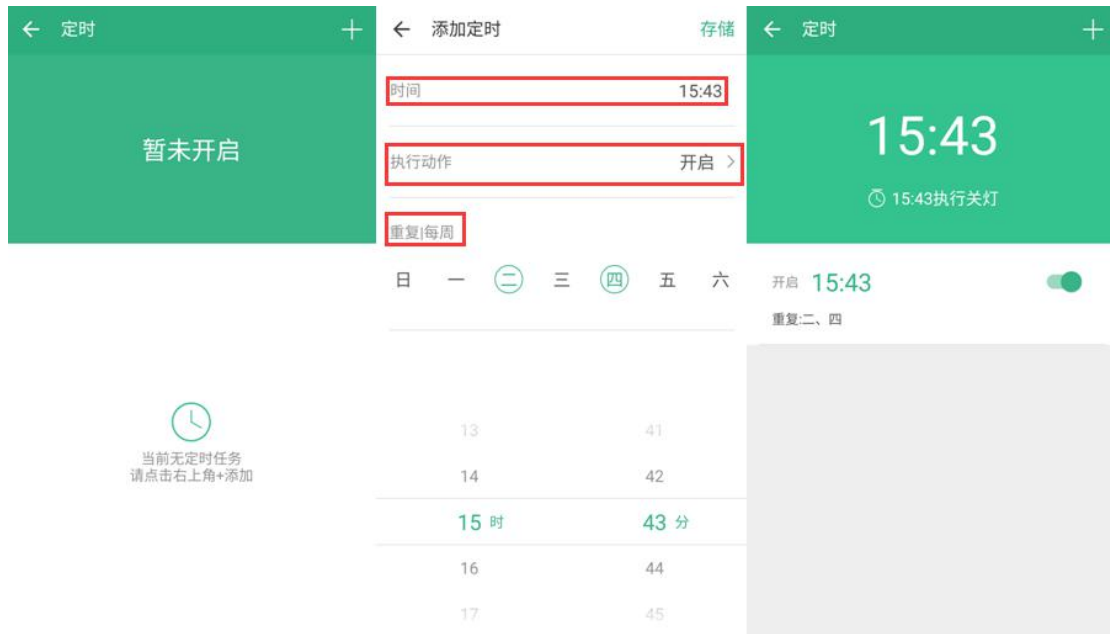
5.1 开关

调试完成后，直接安装智能插座 APP，用同样的账号登录，即可进行完整测试。



5.2 定时

可以为插座添加定时、延时操作，如图：



5.3 分享

可以将设备分享给使用该 APP 注册的其他用户

