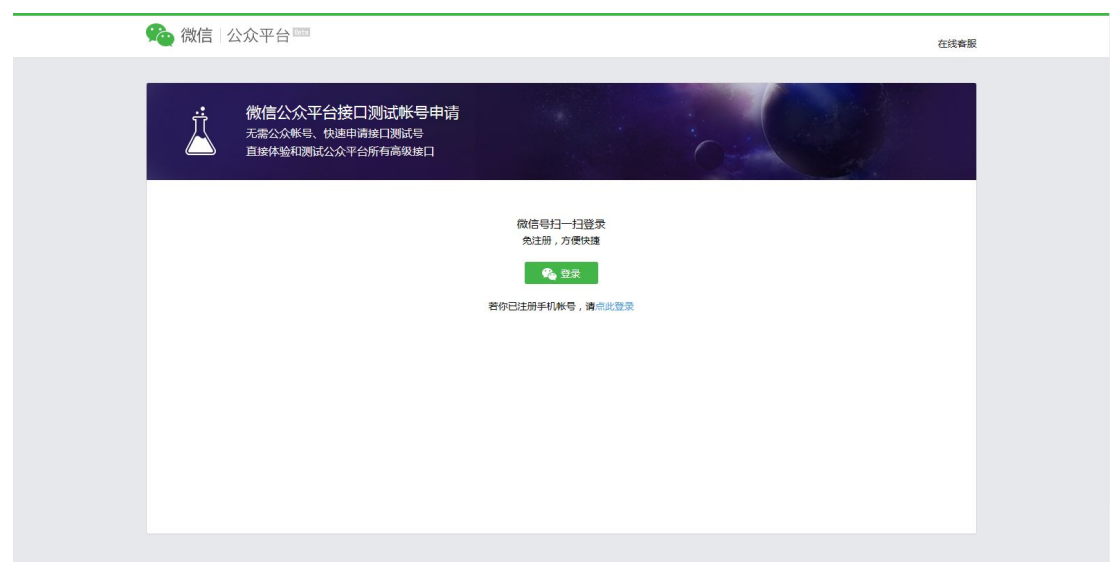


Hi, 小伙伴们, 还记得之前安信可的《3000 个微信板子免费送》活动吗? 你的板子是否已经玩转起来了呢, 现在, 安信可已经对接了微信直连云, 并开放 sdk 给大家下载, 快来跟着一起体验吧~

1 新建产品

1.1 使用测试帐号登录公众号后台

mp.weixin.qq.com/debug/cgi-bin/sandbox?t=sandbox/login



1.2 开启设备功能

		获取用户地理位置	无上限	开启
功能服务	推广支持	生成带参数二维码	100000	
		长链接转短链接接口	1000	
	界面丰富	自定义菜单	详情 ▾	
	素材管理	素材管理接口	详情 ▾	
	智能接口	语义理解接口	1000	
	设备功能	设备功能接口	无上限	开启
	多客服	获取客服聊天记录	5000	
		客服管理	详情 ▾	
		会话控制	详情 ▾	
	网页帐号	网页授权获取用户基本信息	无上限	修改
	基础接口	判断当前客户端版本是否支持指定JS接口	无上限	

注意, 目前微信测试帐号不兼容 chrome, 尽量使用 IE 浏览器打开, 否则会出现无法找到设

置选项的问题。

1.3 新建产品 LED

开启设备功能后，点击设备功能后面的【设置按钮】

功能服务	界面丰富	自定义菜单	1000	详情
	素材管理	素材管理接口		详情
	智能接口	语义理解接口	1000	
	设备功能	设备功能接口	无上限	设置 关闭
	多客服	获取客服聊天记录	5000	
		客服管理		详情
		会话控制		详情
	网页帐号	网页授权获取用户基本信息	无上限	修改

点击【添加产品】，

设备功能

微信硬件平台开发中心

产品管理

设置

未通过认证的产品最多不超过5个，当前还可添加4个产品

+ 添加产品

产品	连接类型	总授权配额	剩余授权配额	剩余授权比	全部状态	操作
 测试设备 8002	Wi-Fi	100	37	37%	未认证	产品详情 编辑 删除

1 / 1

跳转

并按照如下配置设备

编辑产品信息

1 基础资料登记2 产品能力登记

产品名称

测试设备4/15

选择接入方案

即该产品接入微信硬件平台使用的数据传输方案 [详细介绍](#)

☒ 微信硬件云标准接入方案
若选择微信硬件云标准接入方案，设备消息将以Json格式发送到开发者在 [设备功能/设置](#) 填写的服务器地址

☒ 设备直连微信硬件云通道 设备可通过微信硬件平台提供的直连SDK，直接与微信硬件云对接

☐ 厂商云连接微信硬件云通道 设备连接厂商服务器后，可通过设备openAPI与微信硬件云对接

☐ 平台基础接入方案 设备不对接微信硬件云，设备消息将以xml格式发送到开发者在公众平台/基本配置填写的服务器地址

产品描述

产品简要介绍或功能说明

led

3/100

产品品类

家用电器五金家装LED灯

[搜索品类](#)

连接类型

☒ Wi-Fi☐ 蓝牙☐ 蜂窝网络☐ 有线连接☐ 网关子设备☐ 其它

产品添加方式

产品的二维码类型 [查看详情](#)

☐ 每个产品唯一的二维码☒ 每种型号唯一的二维码

产品配置方式

产品支持微信配网或蓝牙发现能力时，请填写配置说明

☒ 微信配网
使用加密 ☐ 使用 ☒ 不使用

请分步骤填写AirKiss引导文本，该内容将在用户使用AirKiss配置网络显示 [查看示例](#)

步骤一 长按指示键5/20

例如：长按指示键

[+添加步骤](#)

☐ 蓝牙发现

下一步

产品能力需要依据微信官方文档：<http://iot.weixin.qq.com/wiki/new/index.html?page=4-5> 进行定义，这里我们以一个微信标准面板为准，只定义开关、亮度调节。

● 请根据产品实际情况选择对应的产品能力，勾选后平台将自动开启相应接口权限。若未实现已勾选的能力，产品将无法通过认证审核。能力详情参考《产品能力集参考手册》、《产品能力定义指引》。

产品基础能力

- ☒ 连接能力 即设备具备微配网、局域网发现、蓝牙发现、近场通讯等能力
- ☒ 微配网 通过微信AirKiss协议配置设备上网
- ☒ 局域网发现 设备可在局域网内被微信发现
- ☐ 蓝牙发现 设备可通过蓝牙被微信发现
- ☐ 近场通讯 设备可以与微信近场通讯
- ☐ 子设备管理 可动态添加删除配件或子设备
- ☐ 消息接收能力 即设备具备接收音乐、地理位置、图片、文件等微信消息的能力

产品标准能力

- ☒ 设备状态类 设备可提供其运行状态、开关状态、电池状态、声光电提醒等设备状态的能力
- ☒ 运行状态 设备处于运行中、待机中或不可用等状态，该项为必选项
- ☒ 开关状态
- ☐ 电池状态
- ☐ 声光电提醒 设备可通过震动、发光或声音提醒发出提示
- ☒ 家电能力 家电类设备可通过微信实现如温度调节等具体能力（支持的能力类别将陆续更新）
- ☐ 插座相关
- ☐ 空调相关
- ☐ 风量调节
- ☐ 音量调节
- ☒ 灯

设备面板 即设备遥控面板，用户可通过此面板操控设备

- ☐ 不使用面板
- ☐ 自定义面板
- ☒ 微信标准面板 此类设备类型系统提供标准面板，由系统根据产品标准能力自动生成，不支持修改。

此设备有以下标准面板可供选择，请选择其中一个：



至此我们已完成了微信公众后台的配置。

1.4 获取一个 licence

通过文档 <http://iot.weixin.qq.com/wiki/new/index.html?page=3-4-6> 得知，每个设备必须要有一个微信分配的 Licence，设备才可以正常运行，首先替换下面的 appid 和 secret 为自己公众号后台的信息

https://api.weixin.qq.com/cgi-bin/token?grant_type=client_credential&appid=APPID&secret=SECRET

测试号管理

测试号信息

appid

appsecret

获取到一个有效的 access_token

https://api.weixin.qq.com/cgi-bin/token?grant_type=client_credential&appid=&secret=
["access_token": , "expires_in": 7200]

再找到公众号后台的产品 ID，

产品	连接类型	总授权配额	剩余授权配额	剩余授权比	全部状态	操作
 测试设备 7888	Wi-Fi	100	100	100%	未认证	产品详情 编辑 删除
 测试设备 8002	Wi-Fi	100	26	26%	未认证	产品详情 编辑 删除

1 / 1 跳转

使用上面获取到的 access_token 和产品 ID 替换下面链接中的信息

https://api.weixin.qq.com/device/getqrcode?access_token=ACCESS_TOKEN&product_id=PRODUCT_ID

获取到如下信息

https://api.weixin.qq.com/device/getqrcode?access_token=
{"base_resp": {"errcode": 0, "errmsg": "ok"}, "deviceid": "12345678901234567890", "qrticket": "12345678901234567890", "devicelicence": "12345678901234567890"}

保存好获取到的数据，下面固件要用。

2 编译固件

下载安信可开放的 SDK,

打开 `airkiss_cloud.h` 改写 `licence` 为自己的实际 `licence`

```
#define DEVICE_LICENCE "FIXME"
```

打开 `airkiss_lan.h` 修改以下两行代码为自己的实际数值

```
#define DEVICE_TYPE "FIXME"
```

```
#define DEVICE_ID "FIXME"
```

编译固件，注意配置编译选项为

```
BOOT?=new
```

```
APP?=1
```

```
SPI_SPEED?=40
```

```
SPI_MODE?=QIO
```

```
SPI_SIZE_MAP?=2
```

3 烧写固件

固件烧写地址依次为：

BIN File	Address
boot_v1.5.bin	0x0
user1.1024.new.2	0x01000
esp_init_data_default.bin	0x0FC000
blank.bin	0x0FE000

固件烧写流程请参考 wiki.ai-thinker.com

4 配置 airkiss

开发板烧写固件后，重新上电，长按功能按键，开发板进入配置网络模式，LED 不停的变换颜色，

此时，打开手机微信，扫描产品型号二维码，进入如下界面：



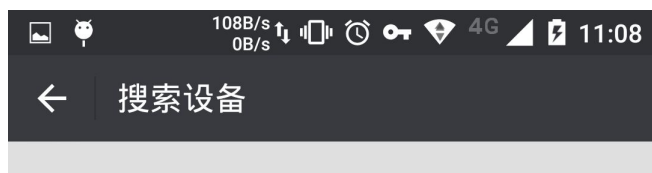
点击【配置网络上网】



输入密码，点击【连接】开始配置
配置成功
图

5 绑定设备

点击返回上一步，搜索设备，



正在搜索可添加的测试设备

若搜不到要添加的设备，请尝试再次[配置设备](#)
[上网](#)



搜索到设备，点击设备名称进行绑定

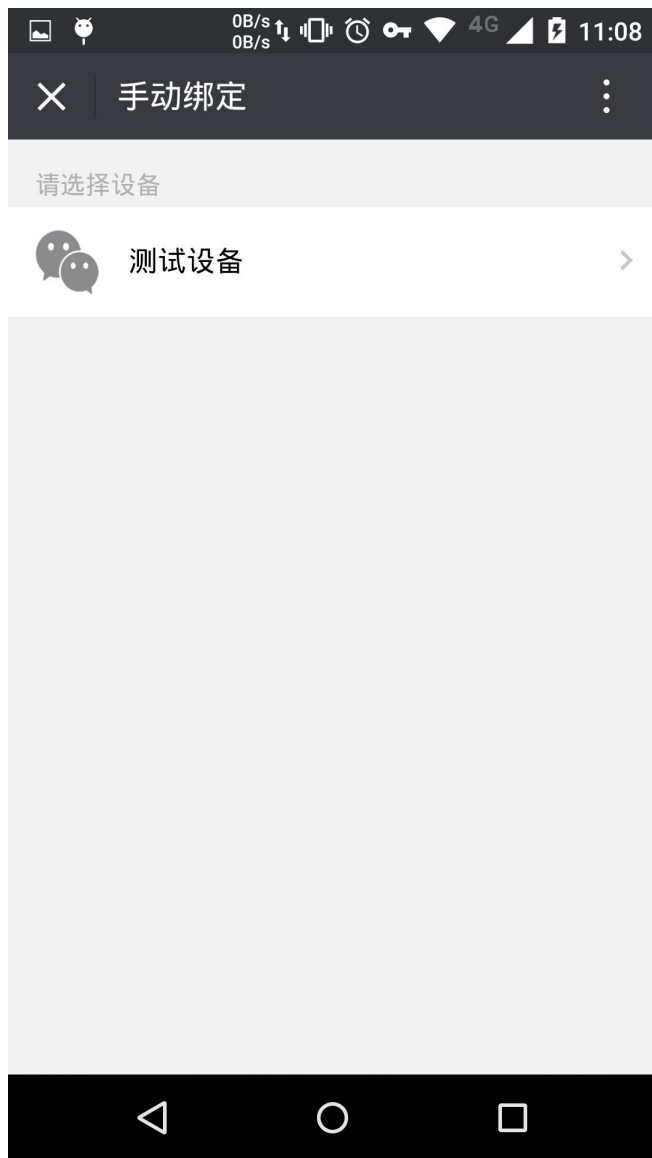


6 控制

进入公众号，点击我的设备
或直接在微信中打开网址

<https://hw.weixin.qq.com/devicectrl/panel/device-list.html?appid=wx11f098dfa51348c5>

选中要控制的设备，点击设备名称进入控制界面



点击开关，可以看到 led 亮灭变化

拖动滑动条，可以看到 led 亮度的变化



至此，微信硬件直连云的开发体验已经结束

7 串口打印

```
ip:192.168.137.2,mask:255.255.255.0,gw:192.168.137.1

NETWORK: change to network_state
_sta_ok
PLATFORM: restart all services
AIRKISS_LAN: delete udp !
AIRKISS_LAN: airkiss_version: airkiss-2.0.0-25645(Jan 15 2016 18:12:09):xtensa-lx106-elf/gcc-4.8.2;ESP8266_IROM
AIRKISS_LAN: creat udp !
AIRKISS_CLOUD: Airkiss lib version:airkiss-3.1.0-26342(Mar 7 2016 16:00:58):xtensa-lx106-elf-ccf1cfd2;Xtensa_LX3

AIRKISS_JSON: airkiss_json_upload:
{'msg_type': 'notify', 'services': {'operation_status': {'status': 1}, 'power_switch': {'on_off': true}, 'lightbulb': {'alpha': 27}}}

AIRKISS_PORTING:url: cloud.hw.weixin.qq.com
AIRKISS_PORTING:-----start dns:5348119-----

AIRKISS_PORTING:dns_found 101.27.131.110
AIRKISS_PORTING:-----END dns:6319937-----

AIRKISS_PORTING:airkiss creat tcp
AIRKISS_PORTING:tcp socket create success !
AIRKISS_PORTING:dest ip is:101.227.131.110
AI
AIRKISS_PORTING:remote ip : 101.227.131.110:11285

AIRKISS_PORTING:tcp client connected success !

AIRKISS_APP: param saved !

AIRKISS_PORTING:airkiss tcp rec
[ 272 ] data:
AIRKISS_CLOUD: Recv Event Callback:1
AIRKISS_CLOUD: Device Login!

AIRKISS_PORTING:airkiss tcp rec
[ 222 ] data:
AIRKISS_CLOUD: Enter Resp Callback:id:1, err:0, funcid:1, len:51
{'error_code':0,'error_msg':'ok','msg_id':72132769}

AIRKISS_LAN: Finish send notify!

pm open,type:2 0
```

发送和上传日志打印:

```
AIRKISS_PORTING:airkiss tcp rec
[ 222 ] data:
AIRKISS_CLOUD: Enter Resp Callback:id:1, err:0, funcid:1, len:51
{'error_code':0,'error_msg':'ok','msg_id':72132769}

AIRKISS_LAN: Finish send notify!

pm open,type:2 0

AIRKISS_PORTING:airkiss tcp rec
[ 382 ] data:
AIRKISS_CLOUD: Recv notify Callback funcid:1, len:219
{'msg_id':72134091,'msg_type':'set','services':{'operation_status':{'status':0}, 'power_switch':{'on_off':true}, 'lightbulb':{'alpha':271}, 'user':'oFgHd44_Epf2CweddW650UQl','test_create_time':1461067197701,'data':''}}
AIRKISS_APP: alpha: 27

AIRKISS_APP: param saved !
AIRKISS_APP: msg_type: set
AIRKISS_JSON: airkiss_json_ask_set:
{'any_error_code':0,'any_error_msg':'ok','msg_id':72134091,'msg_type':'set','services':{'operation_status':{'status':0}, 'power_switch':{'on_off':true}, 'lightbulb':{'alpha':271}}}

AIRKISS_PORTING:airkiss tcp rec
[ 206 ] data:
AIRKISS_CLOUD: Enter Resp Callback:id:3, err:0, funcid:1, len:33
{'error_code':0,'error_msg':'ok'}

AIRKISS_APP: button press!!!

AIRKISS_APP: param saved !
AIRKISS_JSON: airkiss_json_upload:
{'msg_type':'notify','services':{'operation_status':{'status':1}, 'power_switch':{'on_off':true}, 'lightbulb':{'alpha':271}}}

AIRKISS_PORTING:airkiss tcp rec
[ 222 ] data:
AIRKISS_CLOUD: Enter Resp Callback:id:5, err:0, funcid:1, len:51
{'error_code':0,'error_msg':'ok','msg_id':72132770}

AIRKISS_PORTING:airkiss tcp rec
[ 382 ] data:
AIRKISS_CLOUD: Recv notify Callback funcid:1, len:218
{'msg_id':72134093,'msg_type':'set','services':{'operation_status':{'status':0}, 'power_switch':{'on_off':true}, 'lightbulb':{'alpha':271}, 'user':'oFgHd44_Epf2CweddW650UQl','test_create_time':1461067182903,'data':''}}
AIRKISS_APP: alpha: 27

AIRKISS_APP: param saved !
AIRKISS_APP: msg_type: set
AIRKISS_JSON: airkiss_json_ask_set:
{'any_error_code':0,'any_error_msg':'ok','msg_id':72134093,'msg_type':'set','services':{'operation_status':{'status':0}, 'power_switch':{'on_off':true}, 'lightbulb':{'alpha':271}}}

AIRKISS_PORTING:airkiss tcp rec
[ 206 ] data:
AIRKISS_CLOUD: Enter Resp Callback:id:6, err:0, funcid:1, len:33
{'error_code':0,'error_msg':'ok'}
```