

APP 控制 MiCOKit 开发实例教程 (基于 APICloud)

摘要 (Abstract)

本文档仅介绍如何使用 **MiCOKit 开发套件**，开发一个简单的，使用 APP 控制 MiCOKit 单个功能或多个功能的应用实例过程。

适用对象 (Suitable Readers)

本实例适用于 MiCOKit 开发套件的开发者。并适合所有 MiCO-物联网 (IoT) 设备开发者参考。

获取更多帮助 (More Help)

MiCO 开发团队向您推荐：MiCO 开发者学习网站：<http://mico.io/> (开发者中心)，获取更多最新资料。

手机微信“扫一扫”关注：“MiCO 总动员”公众号，获取 MiCO 团队小伙伴最新活动信息。



登录上海庆科官方网站：<http://mxchip.com/>，获取公司最新产品信息。

版权声明 (Copyright Notice)

Copyright (c) 2015 MDWG Trust and the persons identified as the document authors. All rights reserved.

目 录

APP 控制 MiCOKit 开发实例教程 (基于 APICloud)	1
1. 概述	2
1.1. 注册并登录 APICloud.....	2
1.2. 登录 IDE.....	3
1.3. 同步项目	3
1.4. 固件烧录	5
1.5. 应用包结构	5
2. 项目移植.....	6
2.1. 代码移植	6
2.2. 应用设置	8
2.3. 添加模块	10
2.4. 语音识别模块	11
2.5. 云编译	12
2.6. 生成 APP	13
2.7. 扫描 APP 二维码 , 安装 APP.....	13
2.8. APP 控制设备	13
3. 项目解析.....	15
3.1. 设备列表	15
3.2. 控制设备	16
4. 版本更新说明.....	17

1. 概述

本文档会逐步引导您，快速开发一个简单的 APP 应用，控制 MiCOKit 开发板。

准备工作

(1) 请至 <http://www.apicloud.com/dev>，下载并安装 APICloud IDE 开发环境，支持 Windows 系统。

(2) 请至 http://mico.io/wiki/doku.php?id=micokit_sdk，下载最新 MiCOKit SDK 开发包（含 APP 控制 Demo）。

1.1. 注册并登录 APICloud

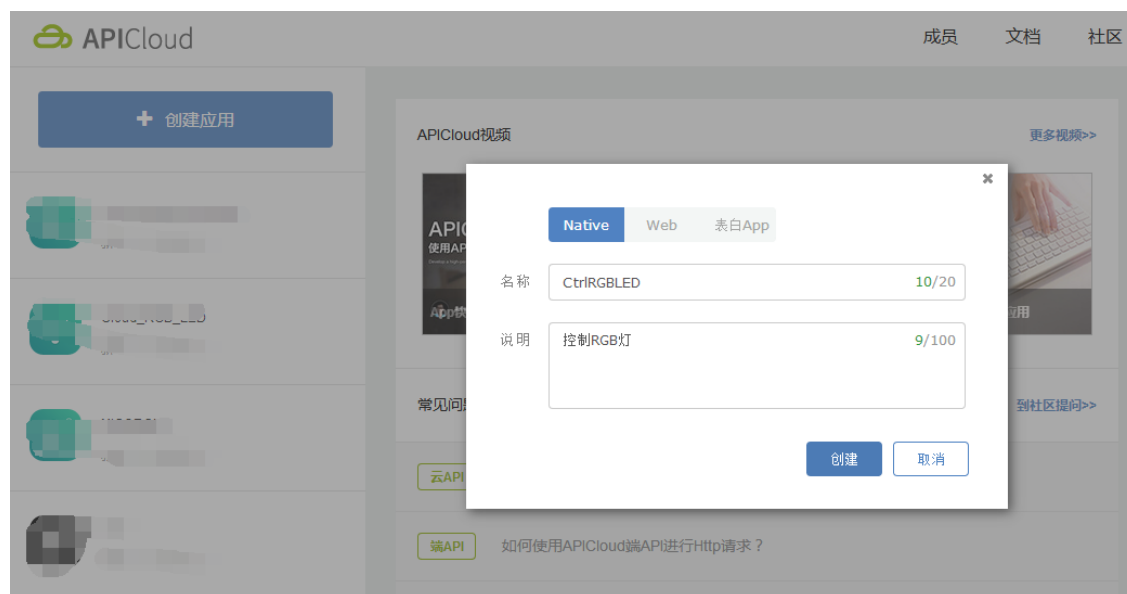
首先，注册并登录 APICloud 主页，如下图，进入控制台页面。



点击下图，左上角“创建应用”。



如下图：选择“Native”，填写“名称”及“说明”，应用创建完成。



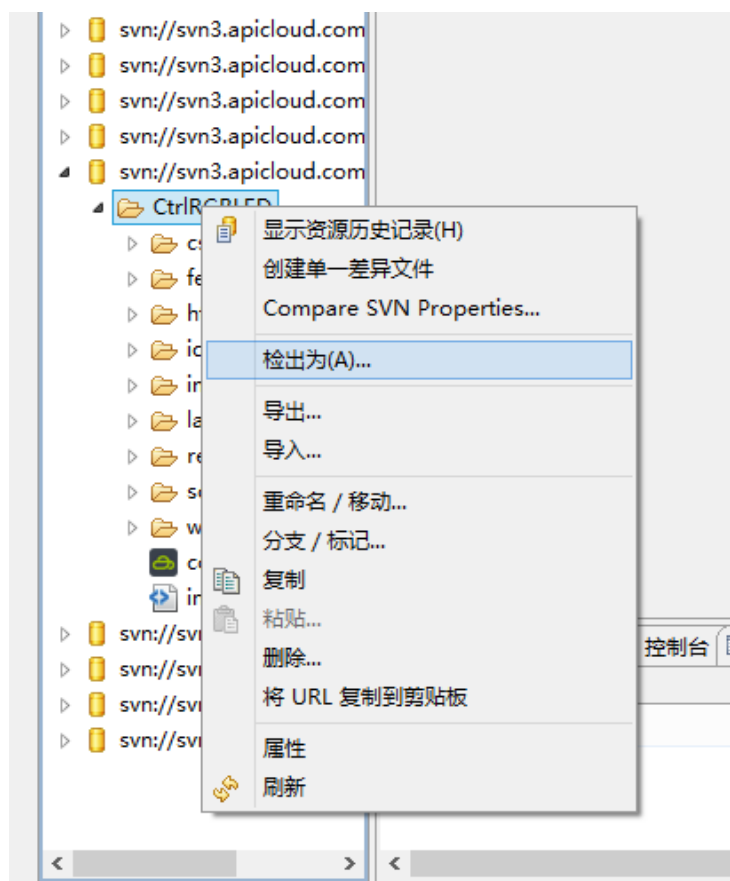
1.2. 登录 IDE

用以上注册的 APICloud 账号登录 IDE。

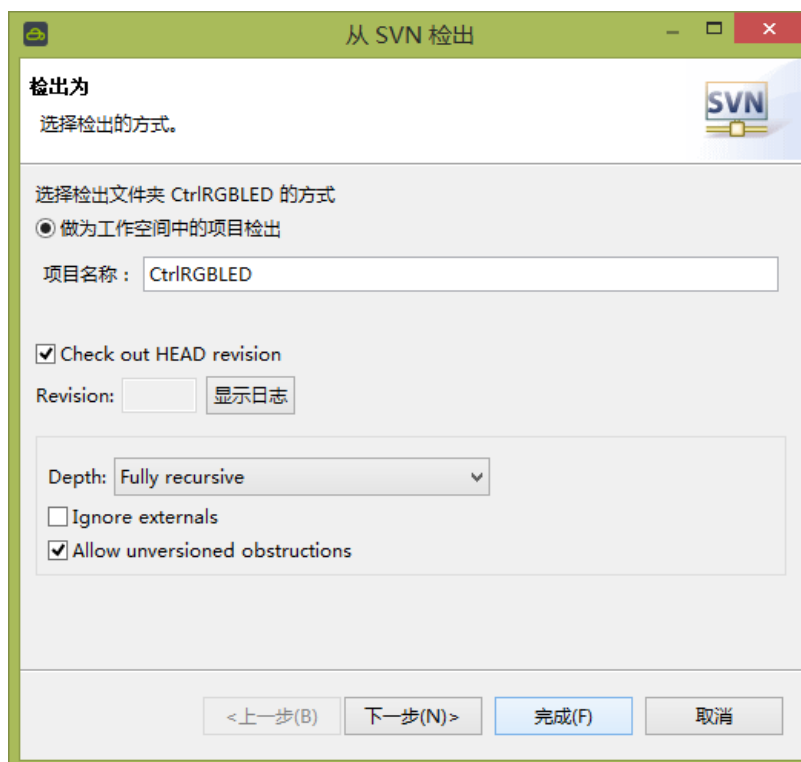


1.3. 同步项目

- 1) 登录后，左侧选择“云端资源库”，根据 APICloud 创建的应用 ID 选择 SVN 项目。
- 2) 选择项目，右键“检出为”。



3) 点击“完成”，应用创建完成。

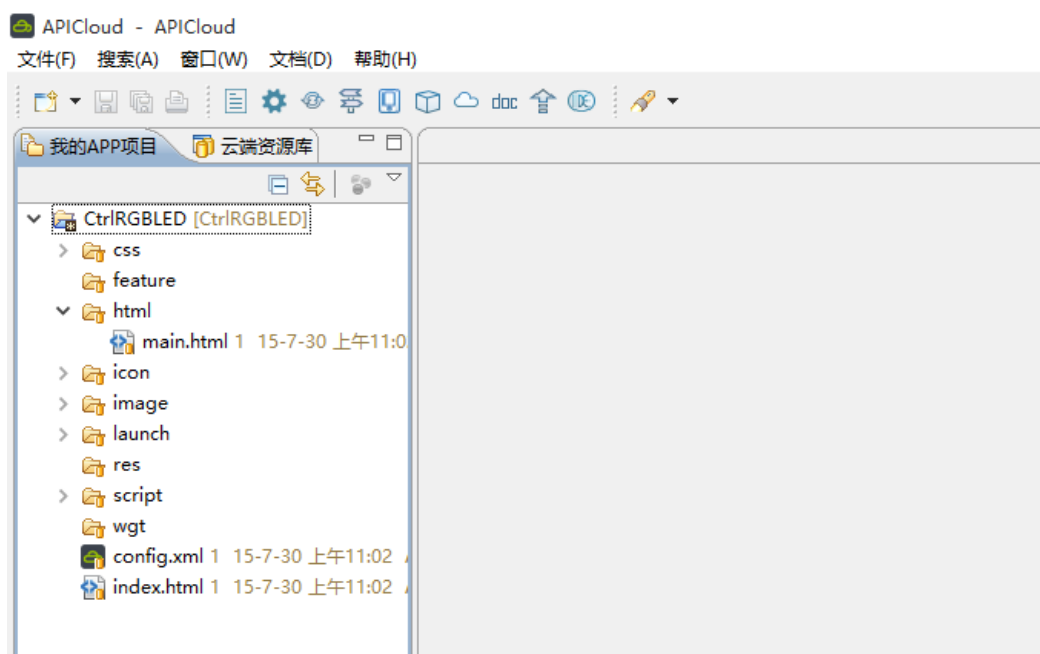


1.4. 固件烧录

三个 APP 对应的 MiCOKit 固件工程分别为：

序号	APP 文件	对应工程
1	APICloud_RGB_LED	Demos/Cloud_RGB_LED
2	APICloud_Read_Humiture	Demos/Cloud_Read_Humiture
3	APICloud_RGB_Humiture	Demos/Cloud_RGB_Humiture

1.5. 应用包结构

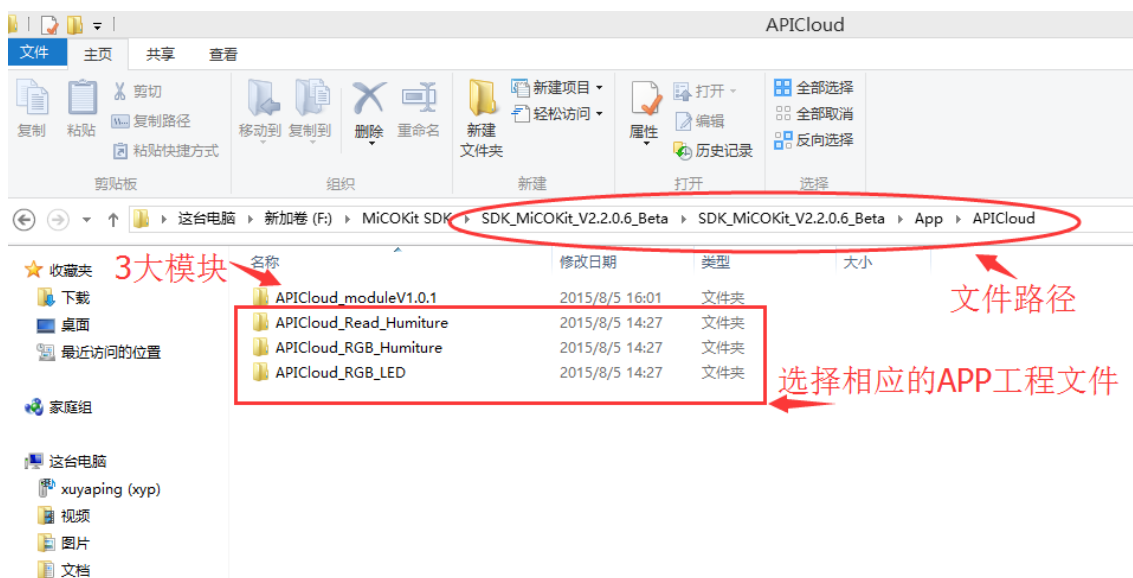


“config.xml” 和 “index.html” 必须包含，其它均为可选。“config.xml” 是配置文件，“index.html” 是启动页面，“icon” 为图标文件目录，“launch” 为启动图片目录（更多介绍详见 Widget 包结构说明文档）。

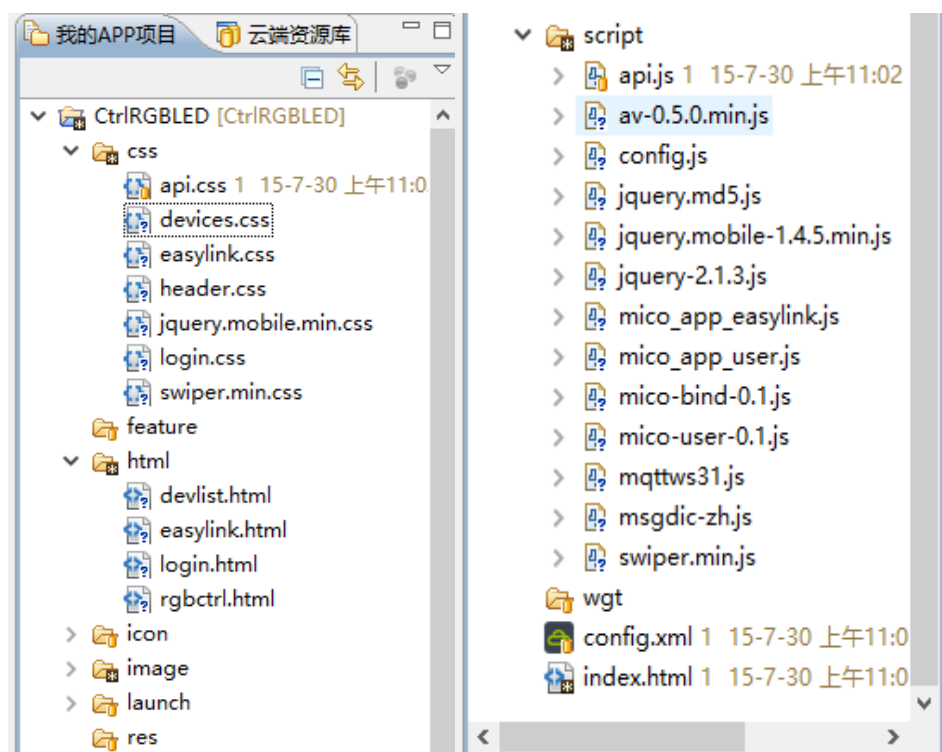
2. 项目移植

2.1. 代码移植

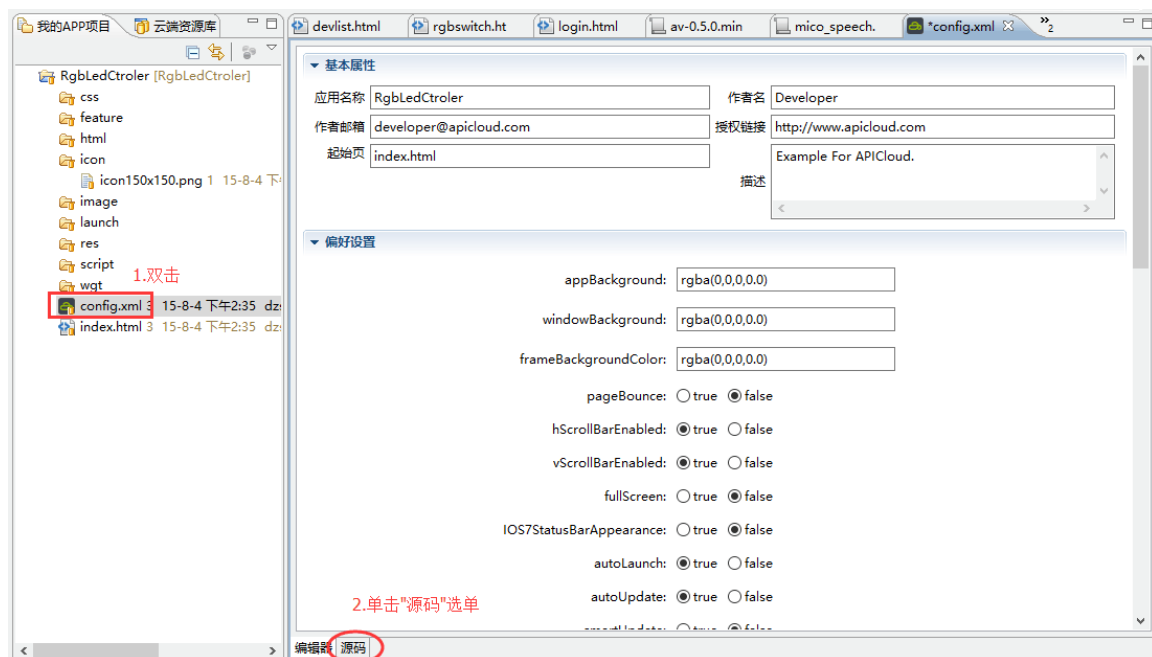
打开 MiCOKit SDK，找到对应的 APP 文件，



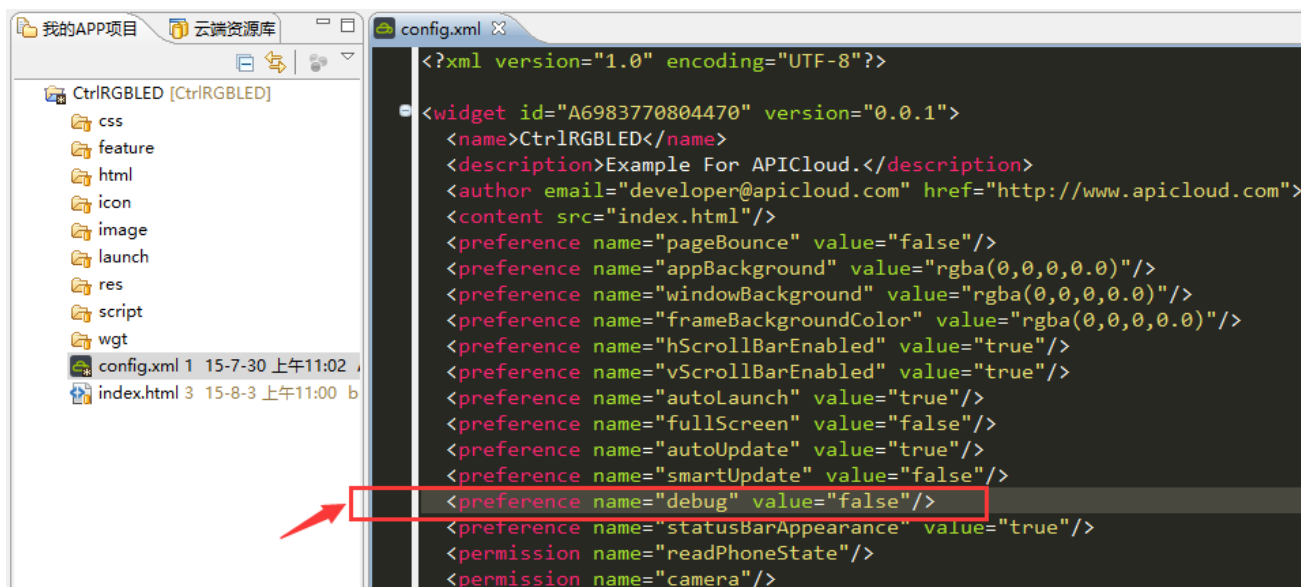
将相应文件夹中的所有文件，直接覆盖到 CtrlIRGBLED 工程下 用户本地的文件，覆盖后如下图：



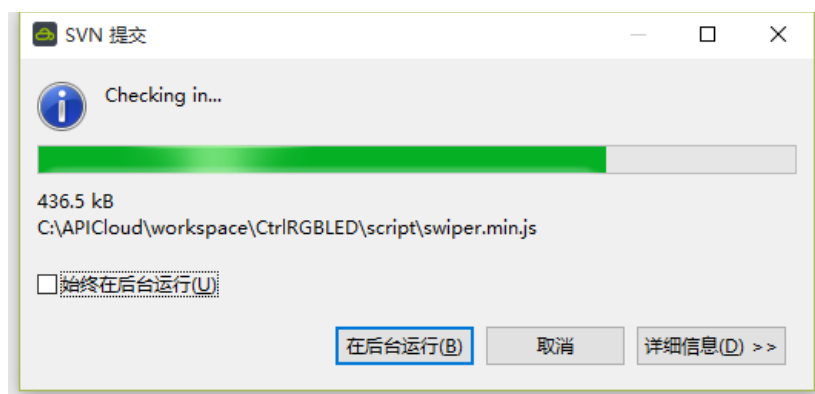
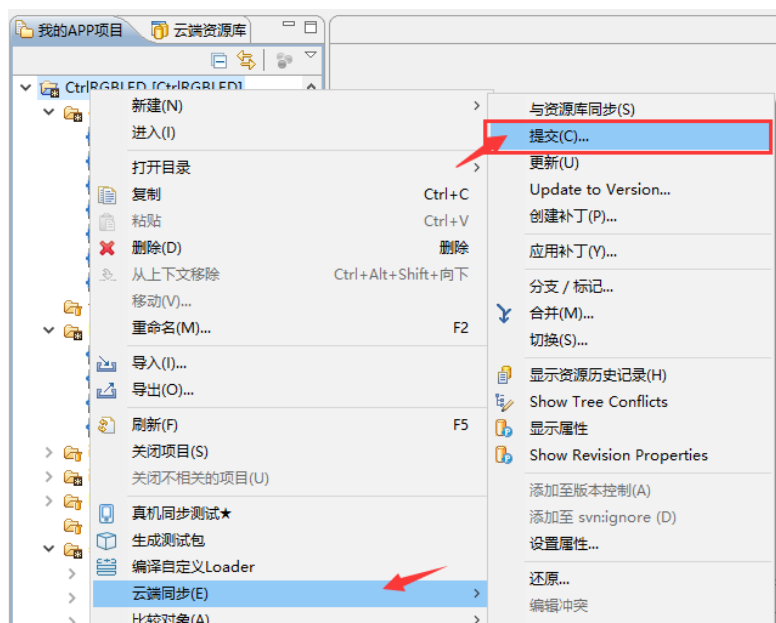
修改配置参数，打开 config.xml 文件，如下图：



将源码中的 debug 的 value 设置为 false，如下图。



提交到云端。

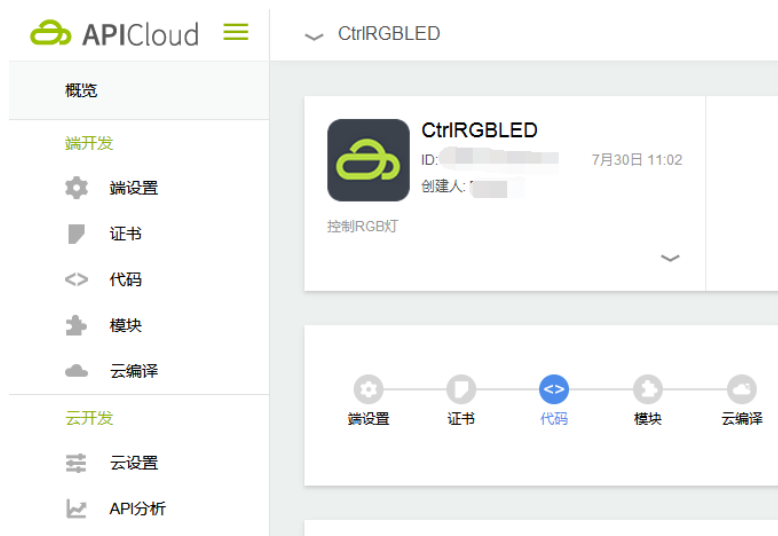


2.2. 应用设置

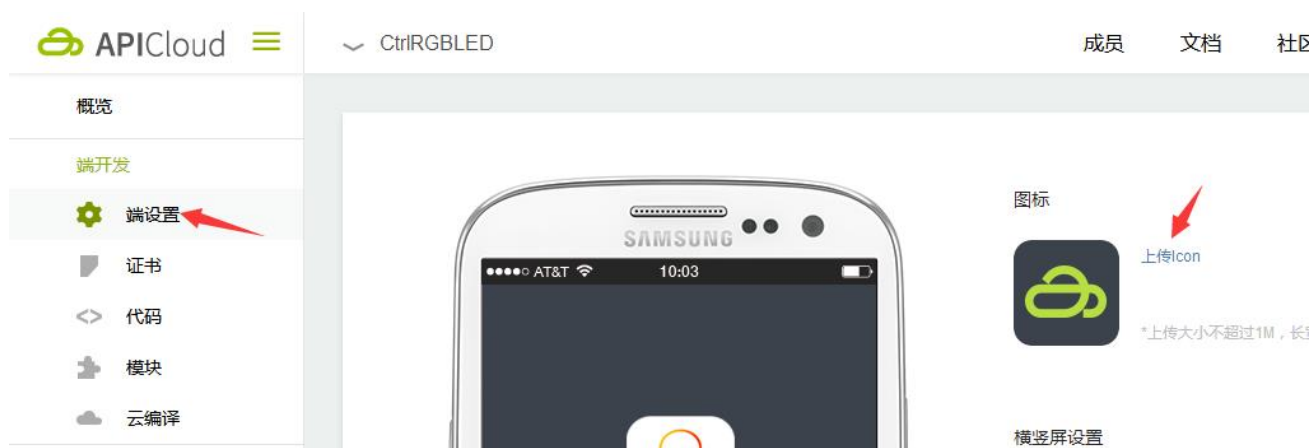
打开 APICloud 控制台页面，找到之前创建的应用，如下图。



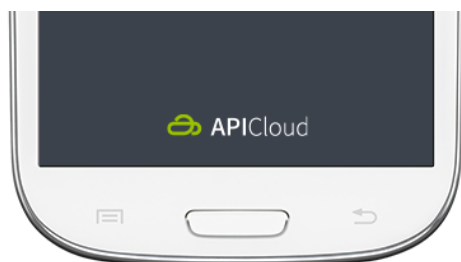
点击应用



修改图标和启动页



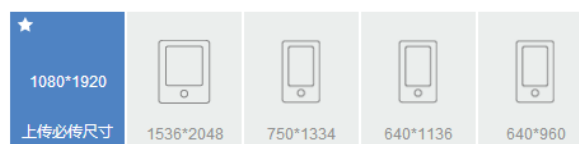
图标



启动页

必传尺寸为1080*1920（长1720px-2120px之间，宽880px-1280px之间）系统将智能适配640*1136、640*960、1536*2048、800*1280、720*1280、400*800等分辨率的机型，以帮助你快速创建应用。

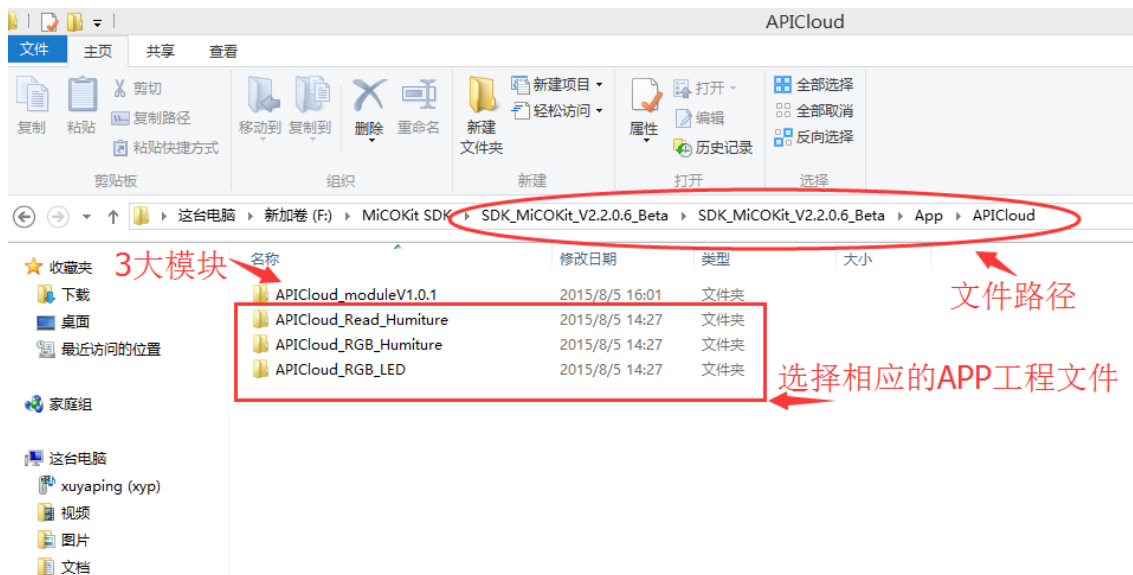
*针对iOS机型分别单独上传标准分辨率的启动页。



启动页

2.3. 添加模块

打开 MiCOKit SDK，找到对应的 APP 的模块文件，如下图：APICloud_moduleV1.0.1.



请按照以下序号顺序进行上传自定义模块，如下图：已上传 micoMdns 压缩包。



说明：填写模块名称，概要，版本，点击上传（包含 android 和 iOS 版本的模块一起上传），模块名和文件名一致，版本为 0.0.1。

上传成功后，点击加号，添加此模块：



模块添加成功后界面如下图（注意：以下三个模块都需要添加：1. micoBind，2.micoMdns, 3.micoMqtt2）

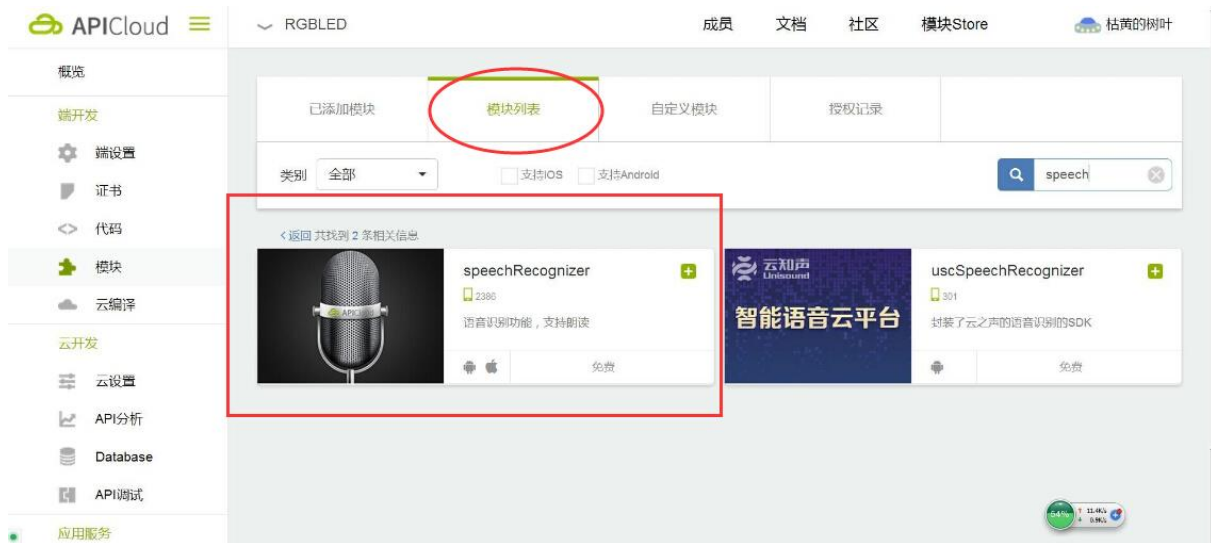
其中，mam 和 push 是默认存在的。



2.4. 语音识别模块

其中，APICloud_RGB_Humiture 的 APICloud 设置时，需要在“模块列表”中搜索“SpeechRecognizer”，添加进来，才可支持语音识别功能哦！

“SpeechRecognizer” 的添加示意图如下：



2.5. 云编译

打开 APICloud 页面-控制台，点击左侧“云编译”，如下图。参数设置：修改 APP 名称，选择 APP 安装的平台，可以 iOS 和 Android 一起选择。因为没有证书，所以只能选择测试版本，其他默认，点击云编译。



2.6. 生成 APP

等 40-50 秒后会生成两个二维码，如下图：



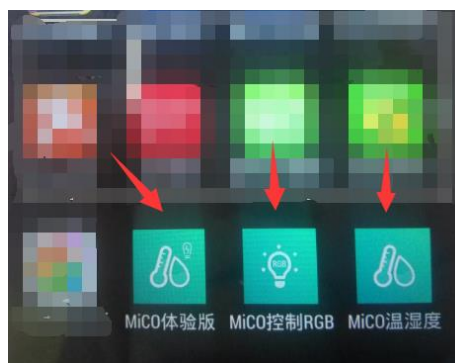
至此你完成了第一步，项目被你成功移植过来了。

2.7. 扫描 APP 二维码，安装 APP

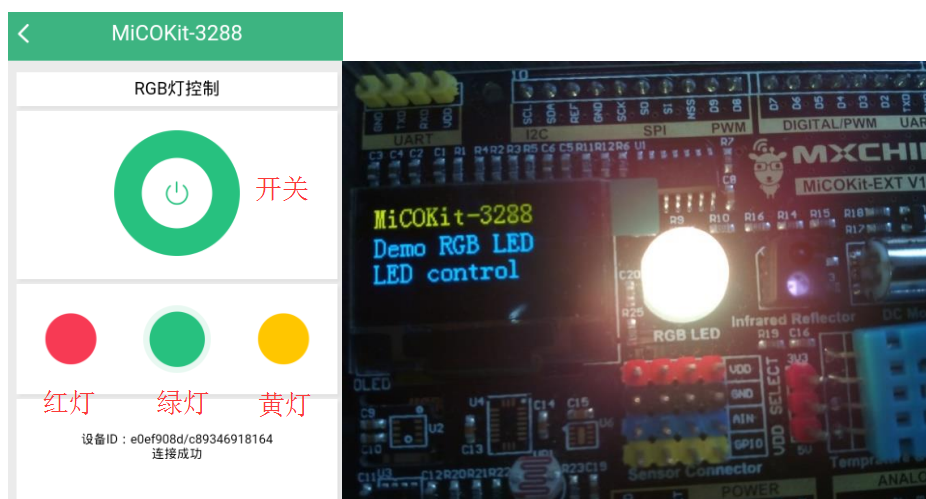
根据不同的手机系统，扫描 3.5 生成的 APP 二维码，即可下载，安装（下载安装方法可参考 MiCO 总动员 APP 使用）。

2.8. APP 控制设备

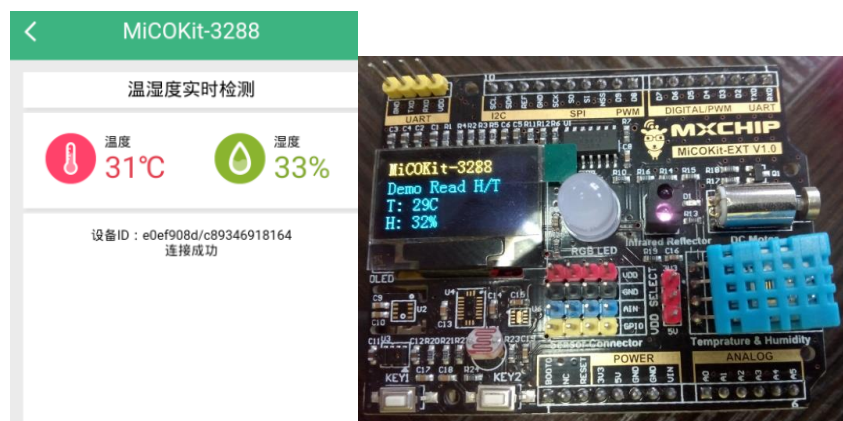
APP 安装后，三个 Demo 的 APP 图标显示如下：



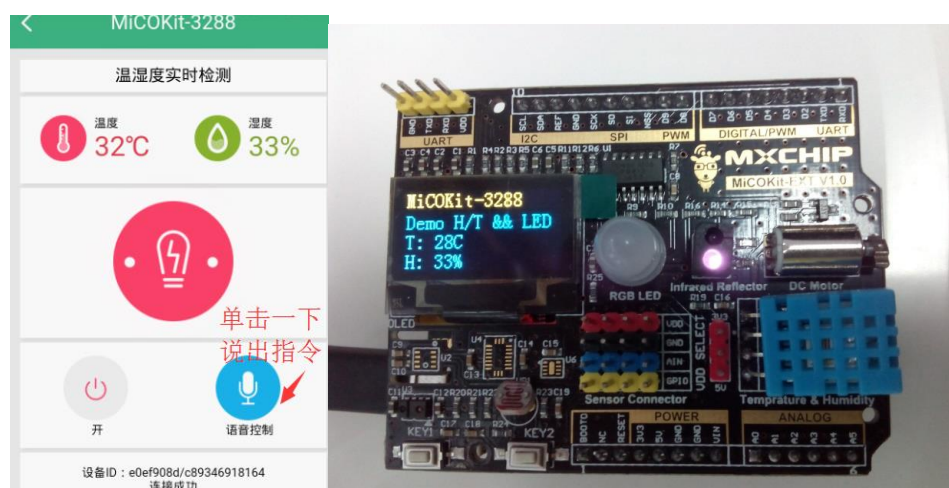
1. MiCO 控制 RGB



2. MiCO 温湿度



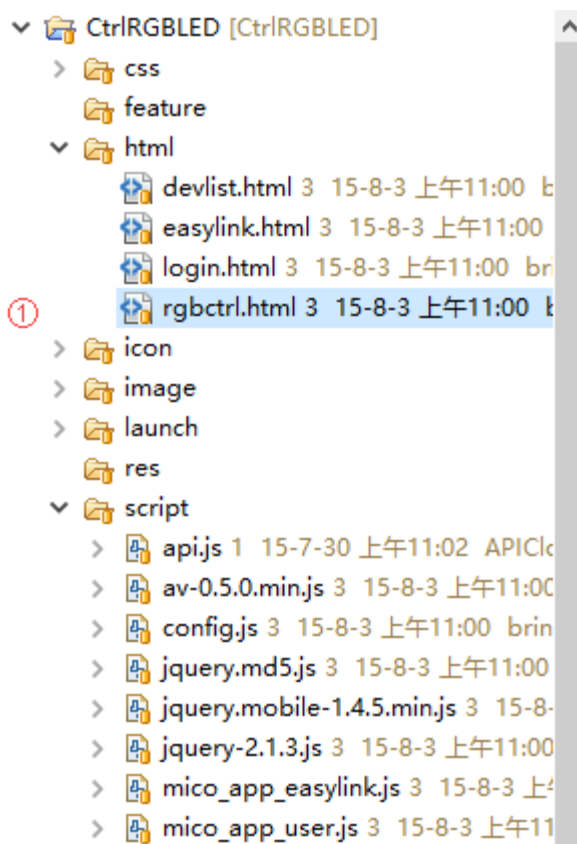
3. APP 控制 MiCOKit 综合版



3. 项目解析

此项目中文件的作用：

- 1、css，控制页面的样式，配合 html 文件中的 HTML 代码来显示
- 2、html，页面的布局
- 3、image，页面需要显示的图片
- 4、script，业务逻辑



其中：css 和 image 文件不做赘述，重点介绍下 HTML 和 script。

3.1. 设备列表

1) devlist.html 为设备列表界面，getdevlist()方法是获取设备列表以及将列表转成页面显示的元素，具体的接口是 getDevList，返回的是 json，如果需要自己设计列表界面，请参考 showDev 方法来解析 json。

```
$mico.getDevList(userToken, function(ret, err, devinfofb))
```

这里传入的 usertoken 可以通过以下方法获取，前提是用户已经登录成功，关于用户注册部分可以使用默认的，或者参考 mico.io。


```
var userToken = getUserInfo().get("userToken");
```

2) 点击设备列表里的某一个设备后，会调用 `torgbctrl()` 跳转到新的页面 (`rgbctrl.html`)，跳转时候将 `deviceid` 和 `devicename` 一起传递过去，

```
pageParam : {  
  "deviceid" : id,  
  "devicename" : name  
}
```

3.2. 控制设备

1) 获取之前传递过来的参数：

a) 获取 `deviceid` : `api.pageParam.deviceid`;

b) 获取 `devicename` : `api.pageParam.devicename`;

2) 打开页面时候判断 `deviceid` 不为空，则自动去连接 mqtt : `ez_connect(device_id)`

3) 连接 mqtt

a) `api.require("micoMqtt2")`; 先引用此模块，

b) 传递 `host`、`clientID` 和 `topic` 三个参数用于打开 mqtt。

`Host` 固定，`clientID` (需要按照此标准来定义：`v1-app-[MAC]` 版本号-app-手机 MAC(12 位))，`topic` 默认只监听 `device_id + '/out/#'` 的通道。

4) 控制 RGB 灯

a) `function publishcmd(topic, command)` 方法用于发送控制设备的指令。

注：只要包含 `api.require()` 的代码，基本都是引用的 `native` 的模块。

如果您完成到此步骤，那么恭喜您通关啦！

最简单的物联网设备已经被您开发出来了！

没有完成也不要灰心，仔细参照本文检查之前的步骤，如果还有问题，请移步至 MiCO 社区 <http://mico.io>

Good Luck !

4. 版本更新说明

日期	版本	修订人	更新内容
2015.8.5	V1.0	Jacky Xu	1. 初始版本
2015.9.8	V1.1	Jenny Liu	1. 删除对 MiCOKit 型号及 SDK 版本的指定内容，使本文适用于所有型号 MiCOKit 开发板和所有版本 MiCOKit SDK 开发包