|  |  |
| --- | --- |
| MiCO Documentation Working Group (MDWG) | Jacky Xu |
| Track Number: 0010CN | MXCHIP Co., Ltd |
| Version: 1.0 | 2015.8.3 |
| Category: Technical Manual |  |

APP控制MiCOKit开发实例教程

摘要（Abstract）

本文档仅介绍如何使用**MiCOKit开发套件（MiCOKit-3288或MiCOKit-3165）**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。

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 1](#_Toc426561713)

[1. 概述 2](#_Toc426561714)

[2. 准备工作 2](#_Toc426561715)

[2.1. 注册并登录APICloud 2](#_Toc426561716)

[2.2. 登录IDE 3](#_Toc426561717)

[2.3. 同步项目 3](#_Toc426561718)

[2.4. 固件烧录 5](#_Toc426561719)

[2.5. 应用包结构 5](#_Toc426561720)

[3. 项目移植 6](#_Toc426561721)

[3.1. 代码移植 6](#_Toc426561722)

[3.2. 应用设置 8](#_Toc426561723)

[3.3. 云编译 12](#_Toc426561724)

[3.4. 生成APP 13](#_Toc426561725)

[3.5. 扫描APP二维码，安装APP 13](#_Toc426561726)

[3.6. APP控制设备 13](#_Toc426561727)

[4. 项目解析 16](#_Toc426561728)

[4.1. 设备列表 16](#_Toc426561729)

[4.2. 控制设备 17](#_Toc426561730)

[5. 版本更新说明 18](#_Toc426561731)

# 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开发套件（MiCOKit-3288或MiCOKit-3165）。

# 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mico.io的MiCOKit SDK发布中心，下载最新SDK\_MiCOKit\_V2.2.0.6（含APP控制Demo）。

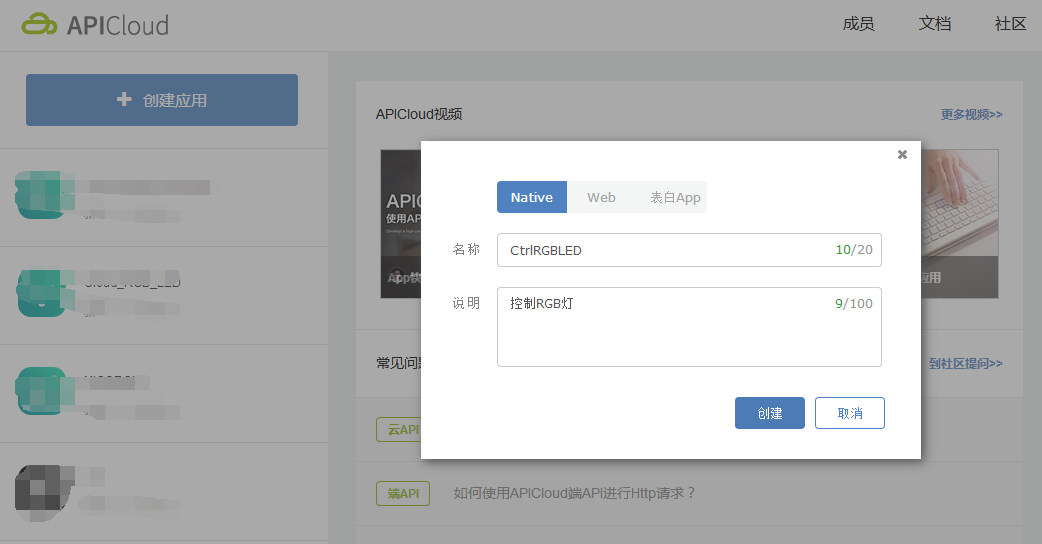
## 注册并登录APICloud

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。



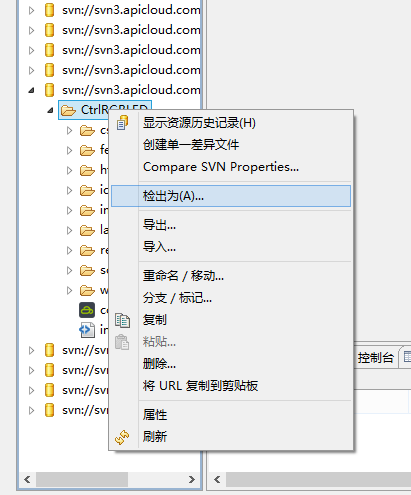
## 登录IDE

用以上注册的APICloud账号登录IDE。



## 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”。



1. 点击“完成”，应用创建完成。

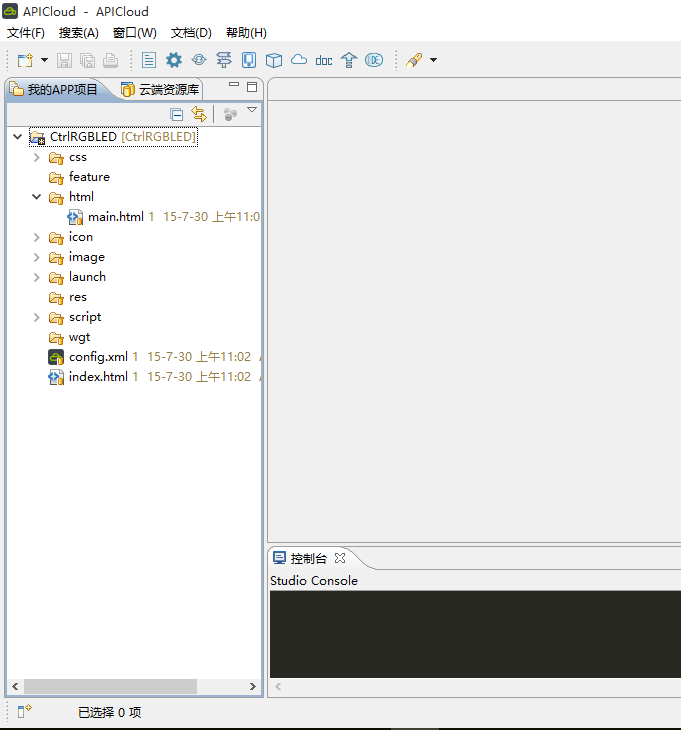


## 固件烧录

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 |

## 应用包结构

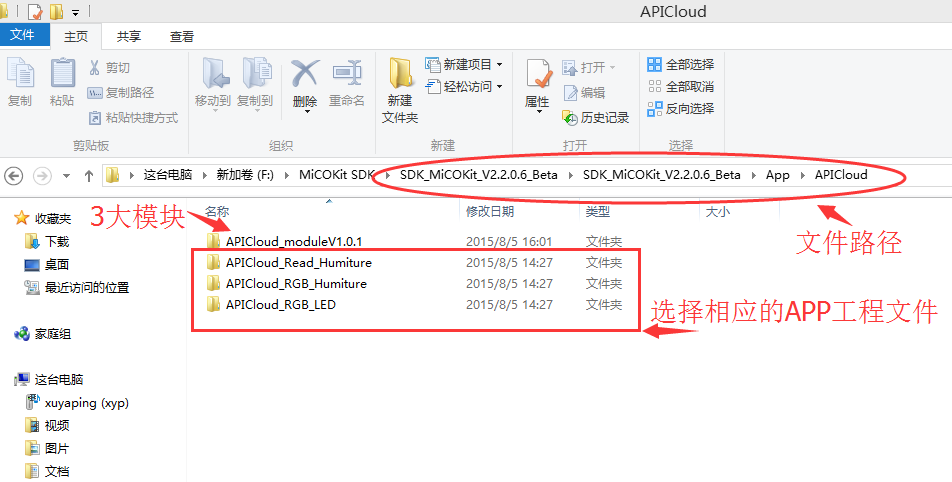


“config.xml”和 “index.html”必须包含，其它均为可选。“config.xml”是配置文件，“index.html”是启动页面，“icon”为图标文件目录，“launch”为启动图片目录（更多介绍详见Widget包结构说明文档）。

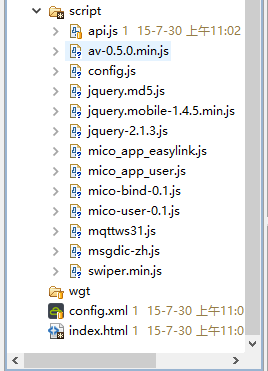
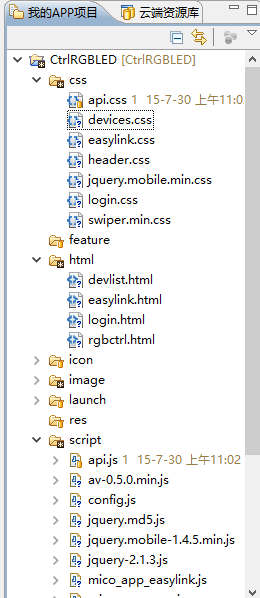
# 项目移植

## 代码移植

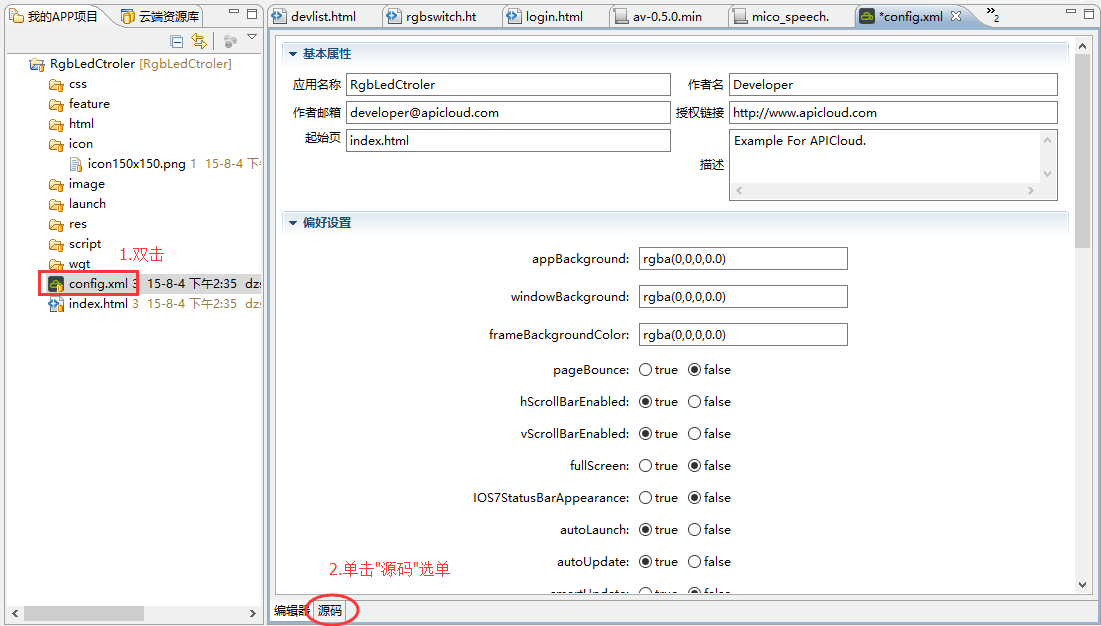
打开SDK\_MiCOKit\_V2.2.0.6\_Beta，找到对应的APP文件，



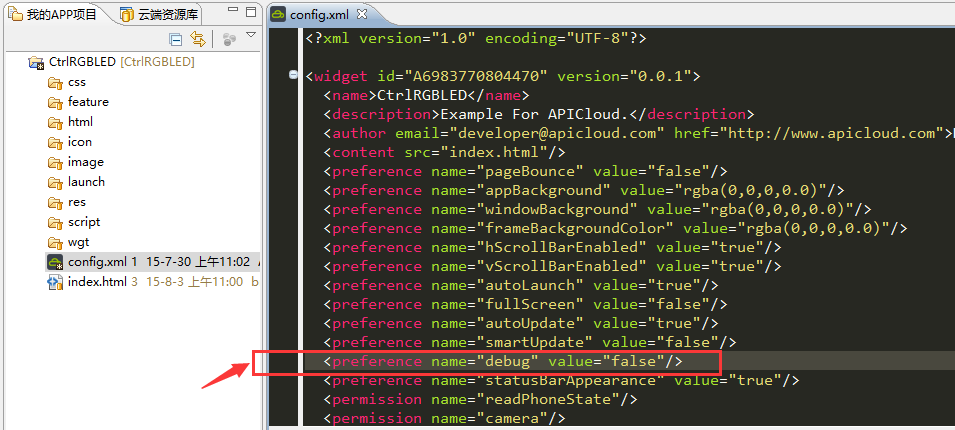
将相应文件夹中的所有文件，直接覆盖到CtrlRGBLED工程下 用户本地的文件，覆盖后如下图：



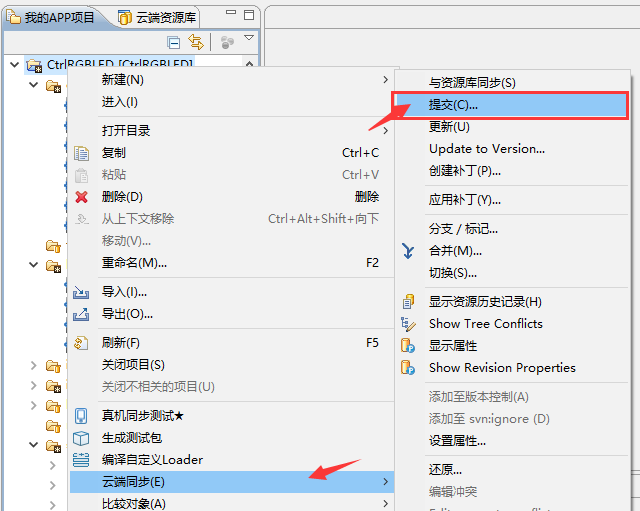
修改配置参数，打开config.xml文件，如下图：

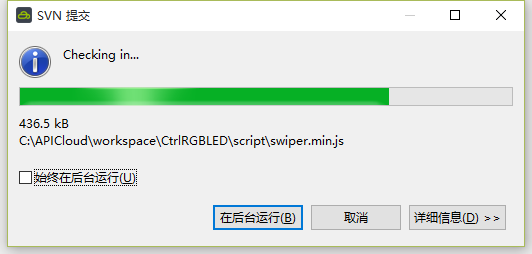


将源码中的debug的value设置为false，如下图。



提交到云端。



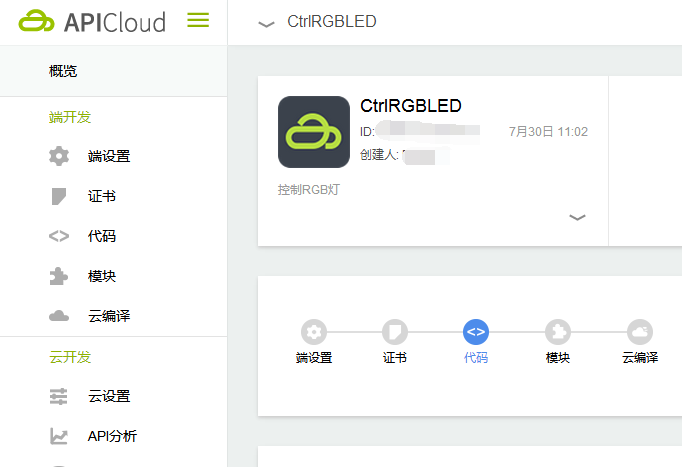


## 应用设置

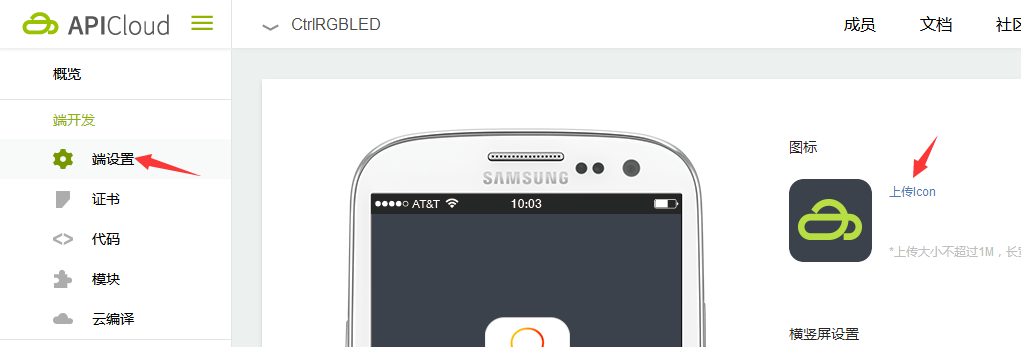
打开APICloud 控制台页面，找到之前创建的应用，如下图。



点击应用



修改图标和启动页



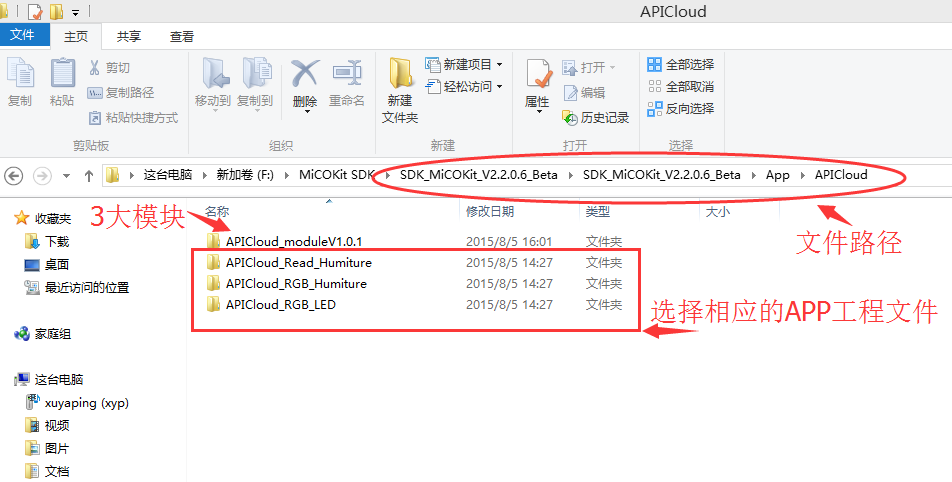
图标



启动页

## 添加模块

打开SDK\_MiCOKit\_V2.2.0.6\_Beta，找到对应的APP的模块文件，如下图：APICloud\_moduleV1.0.1.

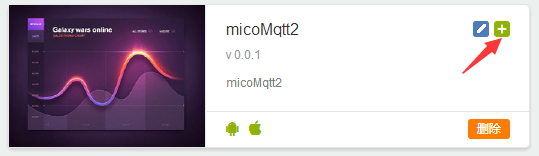


请按照以下序号顺序进行上传自定义模块，如下图：已上传micoMdns 压缩包。



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。



## 语音识别模块

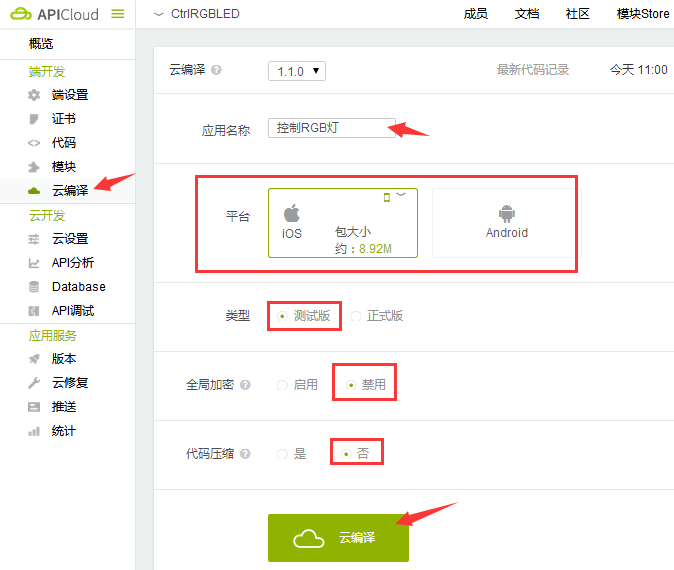
其中，APICloud\_RGB\_Humiture的APICloud设置时，需要在“模块列表”中搜索“SpeechRecognizer”，添加进来，才可支持语音识别功能哦！

“SpeechRecognizer”的添加示意图如下：



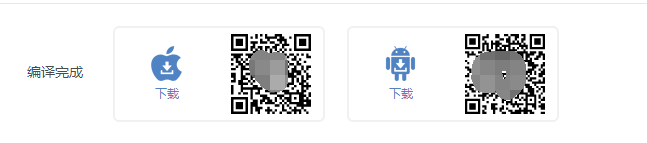
## 云编译

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。



## 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。

## 扫描APP二维码，安装APP

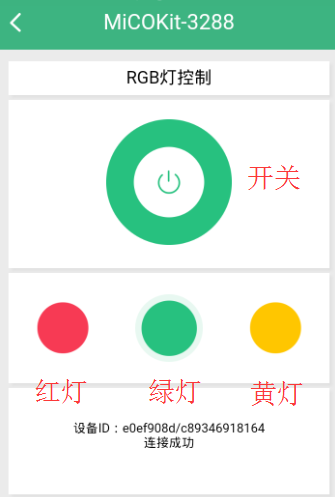
根据不同的手机系统，扫描3.5生成的APP二维码，即可下载，安装（下载安装方法可参考MiCO总动员APP使用）。

## APP控制设备

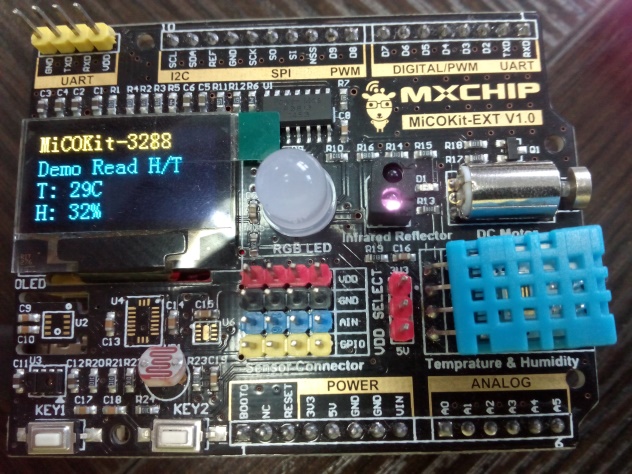
APP安装后，三个Demo的APP图标显示如下：



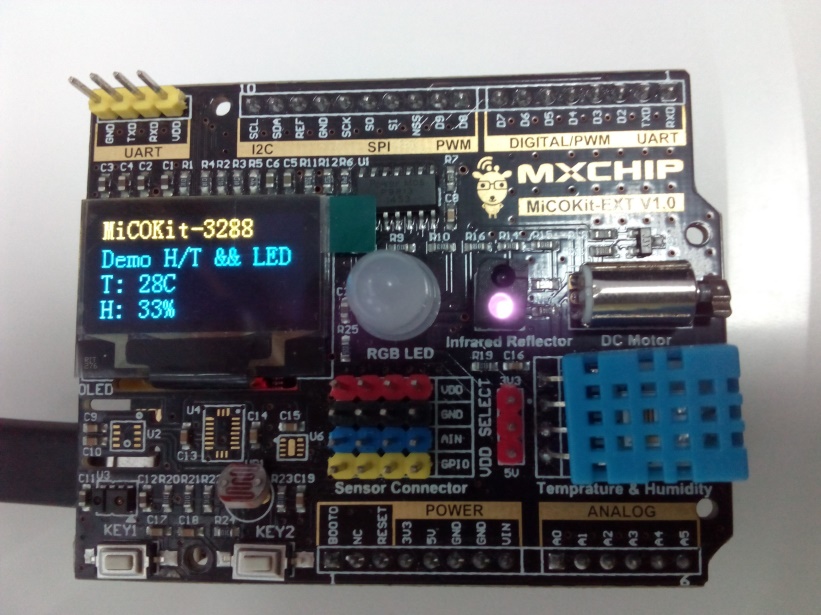
1. MiCO控制RGB



2. MiCO温湿度



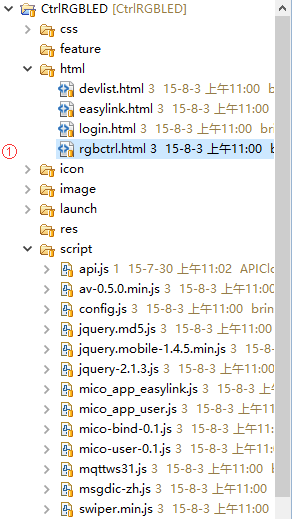
3. APP控制MiCOKit综合版



# 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。

## 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*, *devinfocb*))

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

}

## 控制设备

1. 获取之前传递过来的参数：
   1. 获取deviceid：api.pageParam.deviceid;
   2. 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
   1. api.require("micoMqtt2");先引用此模块，
   2. 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。

1. 控制RGB灯
   1. *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！

# 版本更新说明

|  |  |  |
| --- | --- | --- |
| 日期 | 版本 | 更新内容 |
| 2015.8.5 | V1.0 | 初始版本 |