

# 蓝牙MESH网关开发板快速上手手册

---

1. 概述
2. 软硬件准备
3. 硬件开发环境搭建
  - 3.1 开发板介绍
  - 3.2 硬件连接
4. 软件开发环境搭建
  - 4.1 使用Windows开发环境
    - 4.1.1 概述
    - 4.1.2 CDK工具下载
    - 4.1.3 Windows调试环境安装
    - 4.1.4 编译烧录
      - 4.1.4.1 网关应用编译烧录
      - 4.1.4.2 PHY6220 Controller烧录
- 5 网关运行
  - 5.1 创建产品
    - 5.1.1 在飞燕云创建网关及子设备产品
      - 5.1.1.1 创建项目
      - 5.1.1.2 创建产品并获取五元组信息
    - 5.1.2 在OCC创建网关及子设备产品
  - 5.2 开发板上电
  - 5.3 网络连接
    - 5.3.1 设备配网
    - 5.3.2 手动配置 WiFi 密码
    - 5.3.3 连接到网络
  - 5.4 添加mesh子设备
  - 5.5 向生活物联网平台注册网关及子设备
- 附录 使用Linux开发环境
  - 1 安装YocTools

2 工具链安装

3 调试环境安装

4 编译烧录

## 1. 概述

本文介绍蓝牙MESH网关开发板环境搭建及如何使用CDK/linux环境进行应用编译、烧录。通过本文的指引，开发者可以快速上手蓝牙mesh网关开发板的使用。

## 2. 软硬件准备

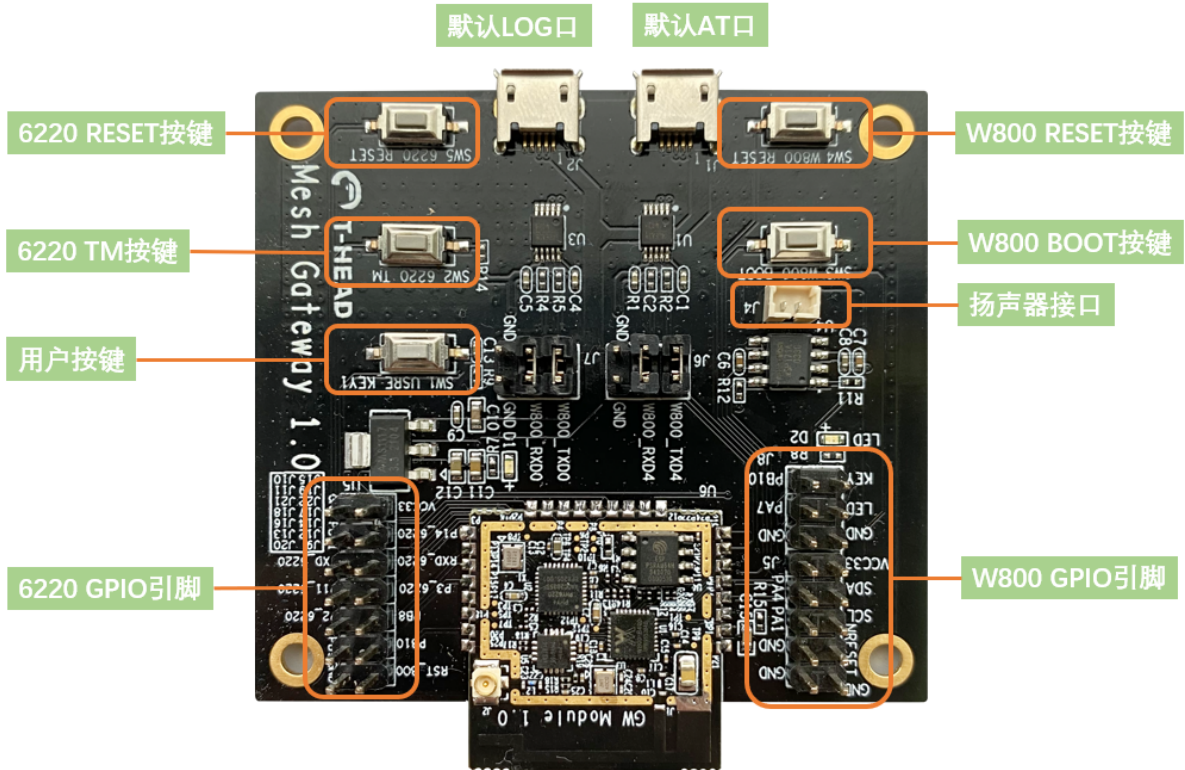
栏目	名称	下载方式
网关SDK包	ble_mesh_gateway_sdk_vxxx x	/
子设备SDK包	ble_mesh_gateway_device_s dk_xxxx	/
网关开发板	蓝牙MESH网关开发板	/
子设备开发板	蓝牙MESH开发板	/
仿真器	CKLink	/
Windows开发工具	剑池CDK集成开发环境 (v2.8.3)	参考后面章节
Linux开发工具	YocTools	参考后面章节
工具链	gcc version 6.3.0 (C-SKY Tools V3.10.22 Minilibc abiv2 B20200502)	参考后面章节
PHY6220固件下载工具	PhyPlusKit	<a href="#">phyplus下载链接</a>

## 3. 硬件开发环境搭建

### 3.1 开发板介绍

蓝牙MESH网关一方面作为MESH网络的配网器和节点，组网并接收和处理MESH网络的数据包；另一方面具备通过WiFi连接到云平台的能力，可以将MESH中的子节点纳入到云端进行统一管理。与此对应，网关开发板使用PHY6220芯片进行BLE及MESH的组网通信，使用W800芯片作为WiFi及主控芯片，管理MESH网络中的子设备。

蓝牙MESH网关开发板硬件如下



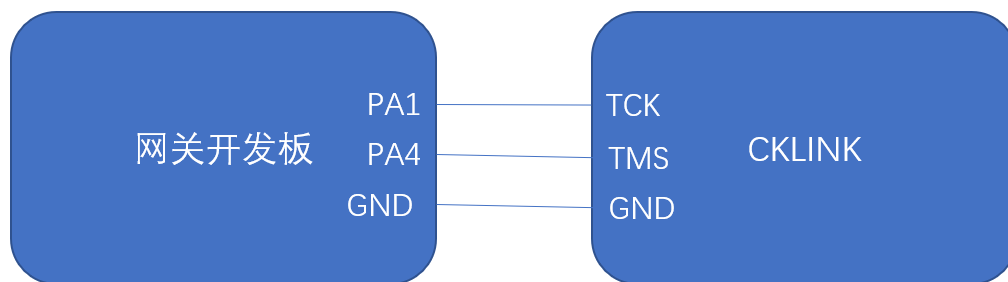
- 主要几个外设接口说明如下

丝印	功能	说明
J2	网关LOG口，主要用于供电以及调试打印	波特率1000000
J1	网关AT口，主要用于AT收发	波特率1000000
PA1	W800 JTAG口，TCK	连接CKLINK的TCK脚
PA4	W800 JTAG口，TMS	连接CKLINK的TMS脚
P3_6220	PHY6220 JTAG口，TCK	/
P2_6220	PHY6220 JTAG口，TMS	用于PHY6220调试
TXD_6220	PHY6220 LOG口，TX	/
RXD_6220	PHY6220 LOG口，RX	主要用于PHY6220烧录、打印。串口烧录需使用上位机工具PhyPlusKit,波特率为115200；烧录完成后按SW5复位按键，并将波特率切到1000000，串口恢复为LOG口

- 各管脚具体定义可参考网关模组原理图以及网关底板原理图
- 各跳线帽复用功能参考[蓝牙MESH网关开发板硬件用户手册](#)。
- 为保证上手使用时各硬件功能正常，用户在使用时需要仔细检查各跳线帽连接是否正确

## 3.2 硬件连接

开发板上电后，使用micro usb线连接网关的LOG口及AT口，并且通过PA1/PA4引脚与CKLINK连接，如下图所示



## 4. 软件开发环境搭建

如之前3.1章节所介绍，网关开发板上使用W800作为WiFi及主控芯片，负责连接云及管理子设备的功能。下面介绍W800芯片的软件开发环境搭建。

## 4.1 使用Windows开发环境

### 4.1.1 概述

Windows开发环境采用剑池CDK开发工具。剑池CDK开发工具以极简开发为理念，是专为IoT应用开发打造的集成开发环境。它在不改变用户开发习惯的基础上，全面接入云端开发资源，结合图形化的OSTracer、Profiling等调试分析工具，加速用户产品开发。

### 4.1.2 CDK工具下载

- 登录[平头哥芯片开放社区](#)，进入栏目"开发中心->集成开发环境->CDK"，下载最新版本的剑池CDK集成开发环境安装包。下载完成后，点击setup.exe，根据CDK安装向导提示完成安装。

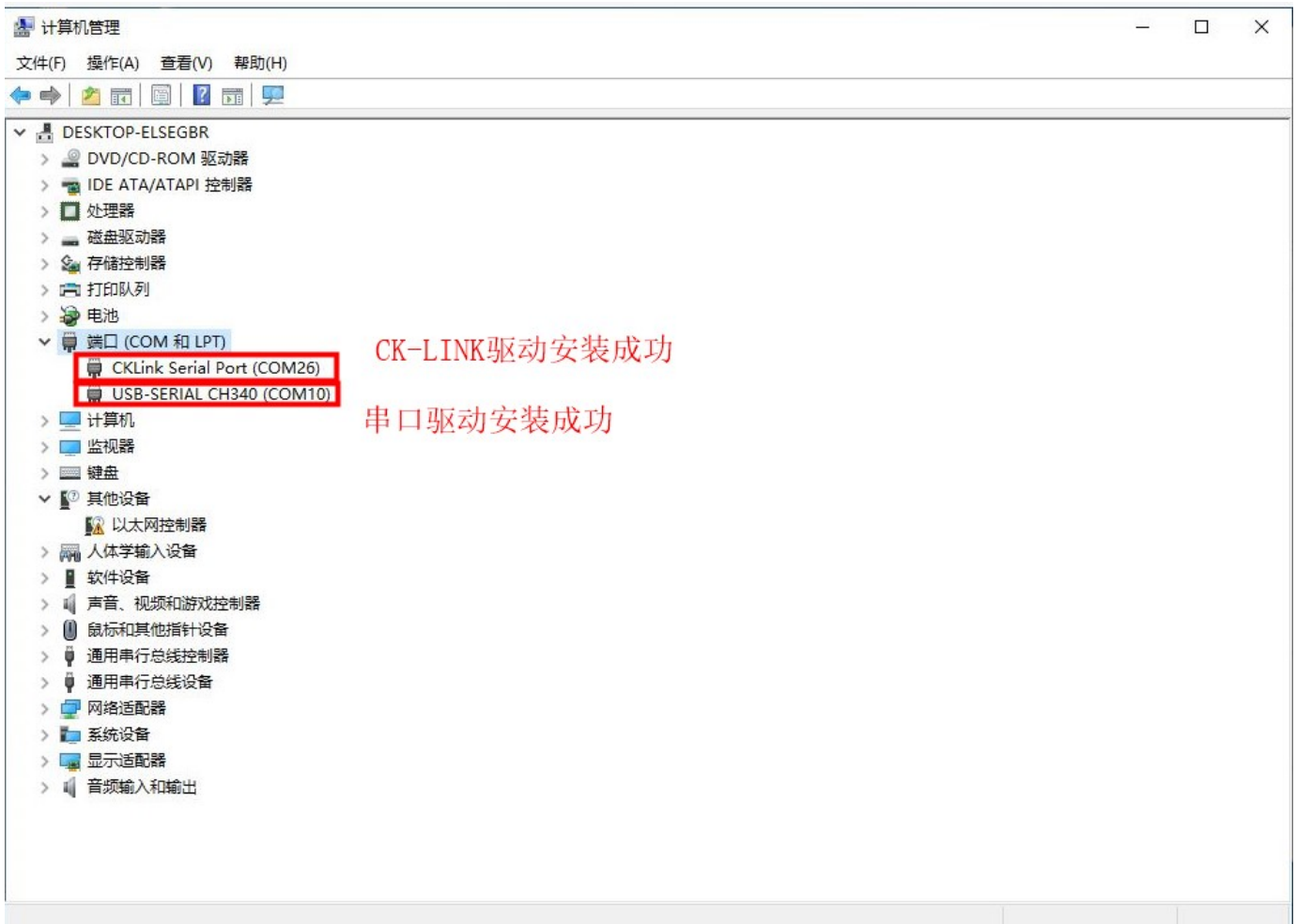
注意：

1. 建议CDK不要安装在C盘，否则需要管理员权限运行
2. 下载或安装过程中有可能被防火墙拦截，需要将其添加到防火墙白名单中

### 4.1.3 Windows调试环境安装

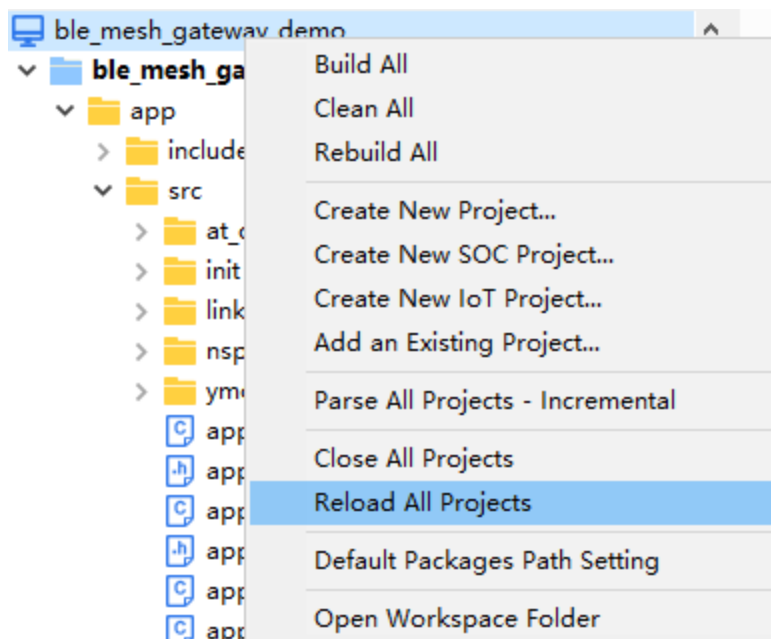
#### 检查驱动状态

- CDK 工具安装后将自动安装CK-LINK驱动，另外还需手动安装USB转串口驱动。插入开发板调试串口和JTAG调试口后，打开计算机设备管理器，确认串口驱动以及JTAG调试驱动安装无误。



#### 4.1.4 编译烧录

本小节介绍网关编译烧录的基本操作。在实际运行时，需要参考5.1.2章节获取到KP后编译并烧录网关才可以正确运行。按照5.1.2章节方法获取kp后用户可以手动reload project，以保证CDK有加载新的kp文件。



#### 4.1.4.1 网关应用编译烧录

- 编译

双击ble\_mesh\_gateway\_demo中的ble\_mesh\_gateway\_demo.cdkws CDK工程，点击CDK工具栏的

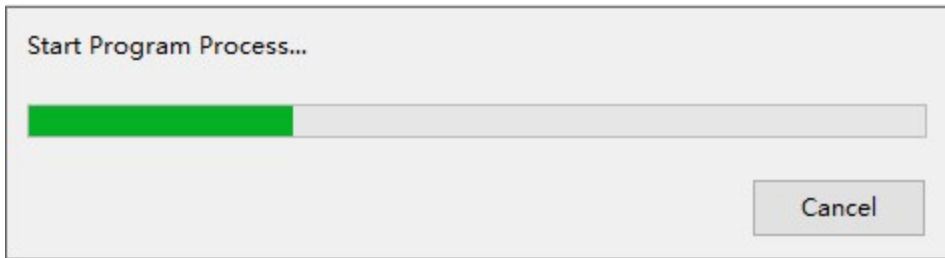


按钮，等待几分钟编译完成，如下图所示：

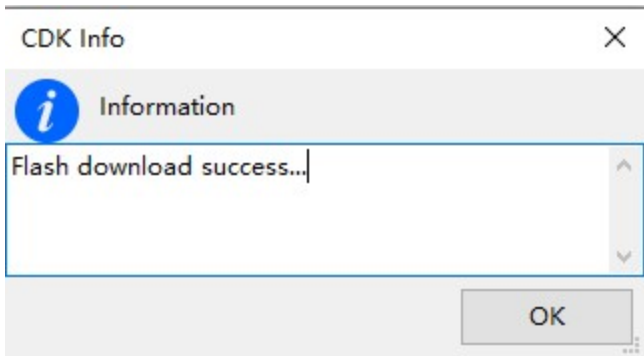
```
[INFO] Generated output files ...
-----
          boot, 0, 0, 0x08002400, 0x0000dc00, 0x08010000, boot
          misc, 0, 0, 0x08010000, 0x000c0000, 0x080d0000
          prim, 1, 2, 0x080d0400, 0x000efc00, 0x081c0000, prim
          imtb, 0, 0, 0x081c0000, 0x00001000, 0x081c1000, imtb
          kv, 0, 0, 0x081c1000, 0x00008000, 0x081c9000
boot,      43088 bytes
prim,     686996 bytes
imtb,      4096 bytes
-----
Create wm bin files
generate normal image completed.
Done
====0 errors, 0 warnings, total time : 25s435ms====
```

- 烧录

点击工具栏的  开始烧录：



烧录完成后，显示烧录成功窗口：



**注意：**用CDK烧录前确保DebugServer关闭

**注意：**若烧录失败，可以尝试按下开发板上的SW4 W800复位按键；或者先长按SW3，再按下SW4使W800进入烧录模式

#### 4.1.4.2 PHY6220 Controller烧录

网关开发板上内置了一颗PHY6220 Controller芯片，用于和蓝牙子设备进行BLE通信。第一次拿到网关开发板，同时需要烧录PHY6220 Controller的固件，其烧录过程需使用PhyPlusKit工具。后续使用不需要重复烧录。

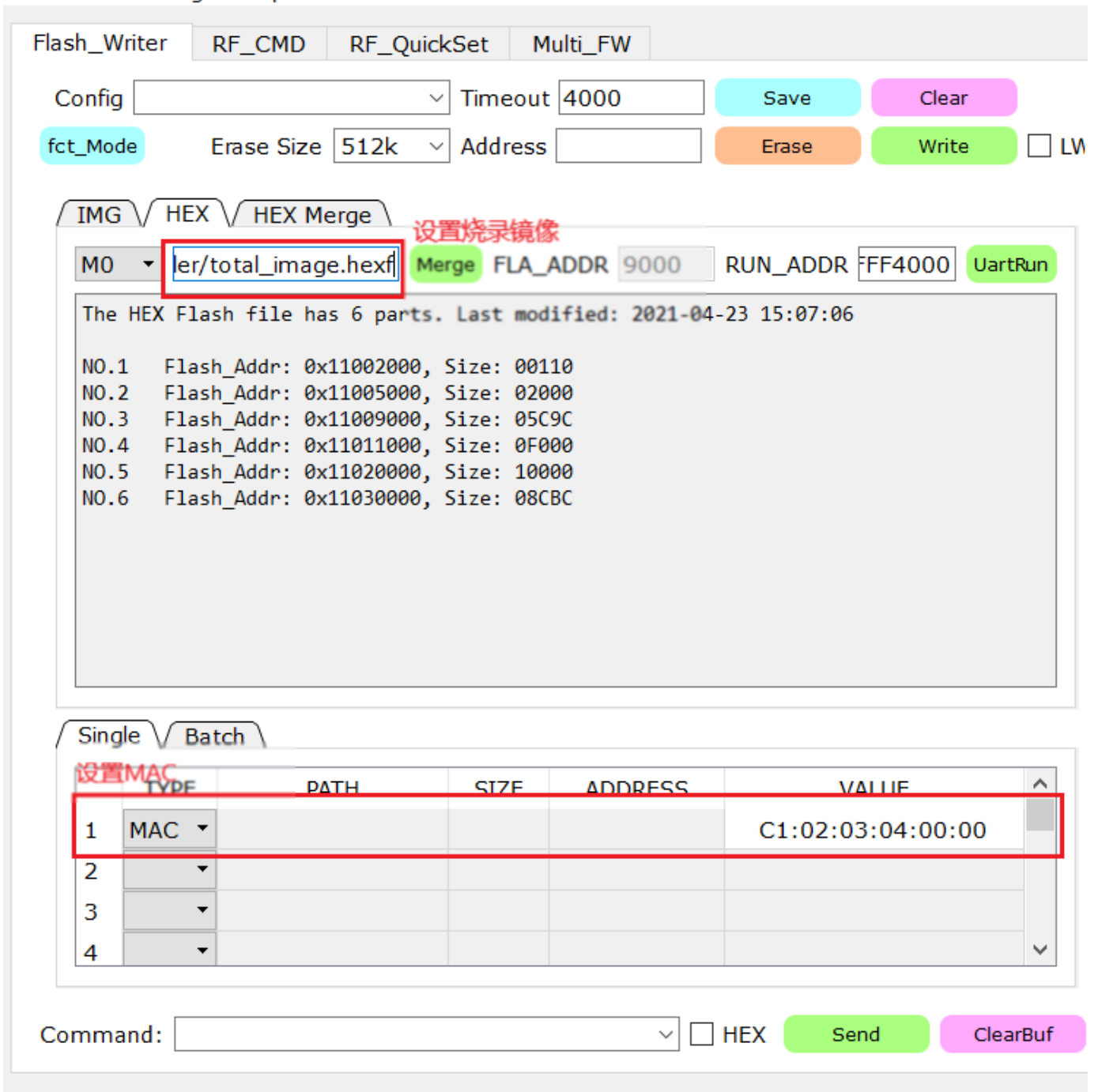
固件烧录步骤如下：

- 网关开发板TXD\_6220 – RXD\_6220 –GND连接PC，PhyPlusKit打开串口，并设置波特率为115200，点击【Connect】。按下网关 SW2 按钮的同时按下SW5按钮，使网关进入烧录模式，如下图所示：





- 双击PhyPlusKit HEX框 选择boards\IoTGW\_CB800\controller下的total\_image.hexf controler镜像，并设置蓝牙MAC地址，如下图所示：



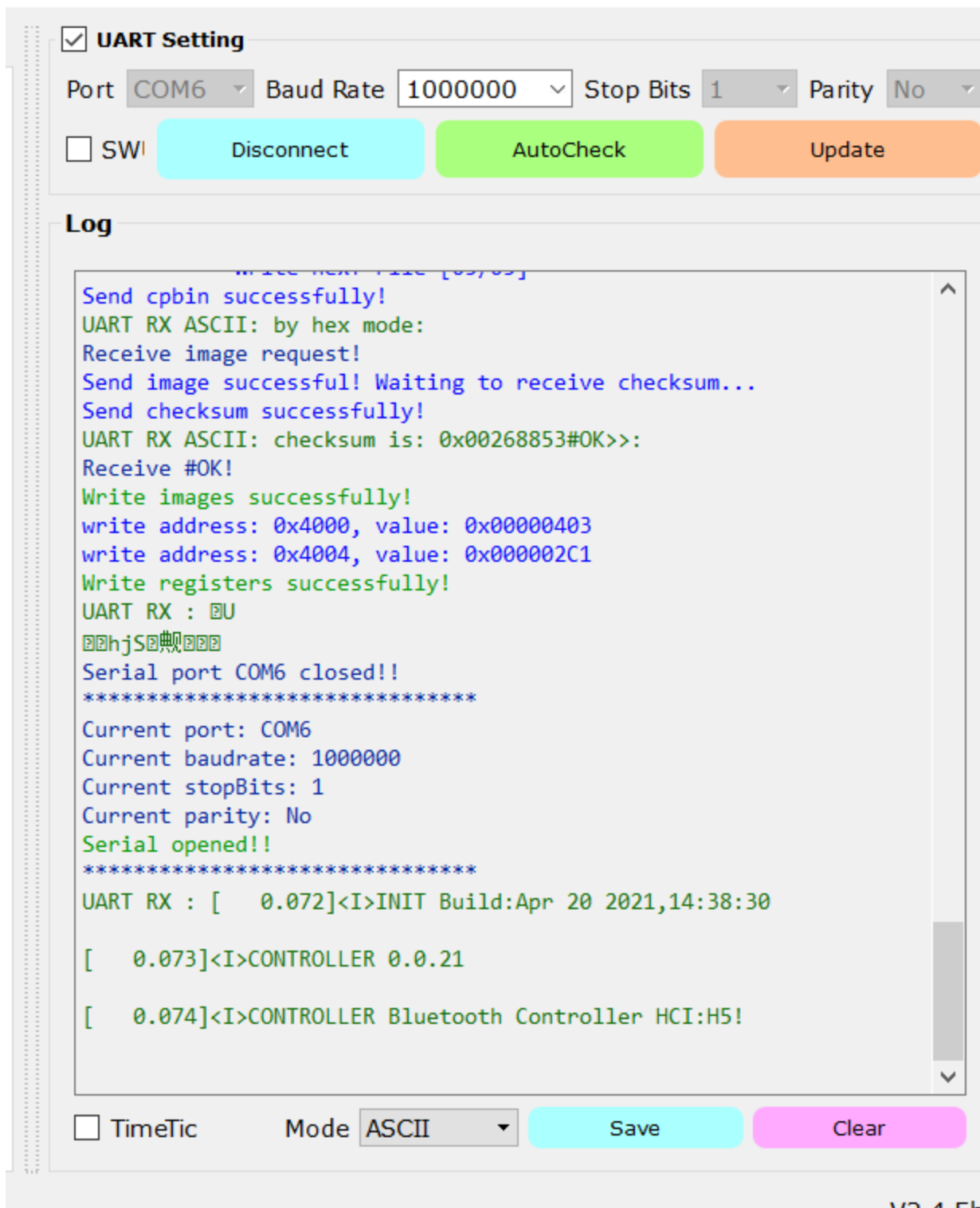
- 点击PhyPlusKit烧录按钮 **Write**，进入烧录，（若出现烧录失败，可尝试点击擦除按钮 **Erase** 再烧录）如下图所示：

```
Send image successful! Waiting to receive checksum...
Send checksum successfully!
UART RX ASCII: checksum is: 0x00222425#OK>>:
Receive #OK!
Receive >>: successful!

=====Write hexf File [04/05]=====
Send cpbin successfully!
UART RX ASCII: by hex mode:
Receive image request!
Send image successful! Waiting to receive checksum...
Send checksum successfully!
UART RX ASCII: checksum is: 0x00504f15#OK>>:
Receive #OK!
Receive >>: successful!

=====Write hexf File [05/05]=====
Send cpbin successfully!
UART RX ASCII: by hex mode:
Receive image request!
Send image successful! Waiting to receive checksum...
Send checksum successfully!
UART RX ASCII: checksum is: 0x00268853#OK>>:
Receive #OK!
Write images successfully!
write address: 0x4000, value: 0x00000403
write address: 0x4004, value: 0x000002C1
Write registers successfully!
```

- 烧录完成后，点击【Disconnect】断开串口，并将波特率设置为1000000，重新连接串口，按下SW5复位键，可以看到如下图所示输出：



## 5 网关运行

在完成上述软硬件环境搭建并了解如何编译烧录后，我们就进入到了如何运行网关的环节。要正确运行网关，还需要如下步骤：



经过这些步骤，网关才可以正确进行后续的鉴权、向云端注册等操作。下面针对这些步骤依次介绍。

## 5.1 创建产品

获取KP需要创建网关及子设备产品，而生成KP所需要的一些元素（如五元组）需要在飞燕云或其它三方云去获取。因此在occ上创建网关及子设备产品前需要先在其它云上创建好产品。下面以生活物联网平台（飞燕云）为例，介绍创建产品的步骤。

### 5.1.1 在飞燕云创建网关及子设备产品

登录[生活物联网平台](#)后，按照如下流程进行产品的创建：



具体步骤参考[生活物联网帮助文档-快速入门](#)。在创建产品和设备成功后，即可获得设备五元组信息。

#### 5.1.1.1 创建项目

登录平台后，需要先创建一个项目。数据会根据项目进行隔离，以确保客户的数据安全。每个项目可以独立授权给其他阿里云账号，可以用于项目的多方开发协作、和运营中心的交付。每个项目会有ID，在调用云端服务接口时，会使用到该ID。


全部项目 自建项目 授权项目 创建新项目

### 新建项目


名称

 8/20

类型



自有品牌项目



天猫精灵生态项目

提供消费级智能设备服务，可在全球任意地区使用，自动连接最近的数据中心。

确定 取消

### 5.1.1.2 创建产品并获取五元组信息

在创建项目后，就可以在项目中创建网关及子设备的产品。每个项目中，可以新建多个产品，一个产品代表一种设备型号。

创建网关产品如下图所示

新建产品

产品信息

\* 产品名称  
蓝牙Mesh网关

\* 所属品类  
网络设备 / 网关 功能定义

节点类型

\* 节点类型  
 设备  网关

连网与数据

\* 连网方式  
WiFi

\* 数据格式  
ICA 标准数据格式 (Alink JSON)

\* 使用 ID<sup>2</sup> 认证  
 是  否

创建子设备产品如下图所示

(注意：配置产品为“接入网关”且与网关之间的协议为“自定义”)

新建产品

产品信息

\* 产品名称  
蓝牙mesh灯

\* 所属品类  
电工照明 / 灯 功能定义

节点类型

\* 节点类型  
 设备  网关

\* 是否接入网关  
 是  否

连网与数据

\* 接入网关协议  
自定义

\* 数据格式  
ICA 标准数据格式 (Alink JSON)

确认 取消

在新建产品成功后，在云平台会产生该类产品的基本信息，其中包括产品的Product Key、Product Secret及Product Id。如下图所示（图中对关键信息做了脱敏处理）

## 蓝牙Mesh网关

更新时间: 2021-05-10 11:52:24

### 基本信息

所属分类: 网关

节点类型: 网关

通讯方式: WIFI

数据格式: ICA标准数据格式 (推荐)

Product Key: a1N0xsTF

Product Secret: aTlyZw5S6Z

Product Id: 6957

认证方式: 设备密钥

### 模组

品牌: 未认证

型号: 未认证

创建时间: 2021-01-07



同时, 创建产品成功后进入对应产品页面, 会出现对产品配置 **功能定义**、**人机交互**、**设备调试**、**批量投产** 的页签, 如下图所示:

生活物联网平台 / 蓝牙Mesh网关 / 功能定义

1 功能定义 2 人机交互 3 设备调试 4 批量投产

标准功能 查看物模型

类型	名称	标识符	数据类型	数据定义	操作
! 暂无数据					

自定义功能 查看

类型	名称	标识符	数据类型	数据定义	操作
属性	白名单	add_whitelist	text (字符串)	数据长度: 10240	查看

蓝牙Mesh网关  
更新时间: 2021-05-10 11:52:24

**基本信息**  
所属分类: 网关  
节点类型: 网关  
通讯方式: WIFI  
数据格式: ICA标准数据格式 (推荐)  
Product Key: a1N0xsTF  
Product Secret: \*\*\*\* 显示  
Product Id: 6957  
认证方式: 设备密钥

**模组**  
品牌: 未认证  
型号: 未认证

平台客服

**功能定义** 页签: 进行网关及子设备产品功能的定义, 即物模型的配置。

**人机交互** 页签: 配置网关的配网方式, 对于使用公版云智能APP的用户, 需要打开 **使用公版App控制产品** 选项

1 功能定义 2 人机交互 3 设备调试 4 批量投产

选择交互端 配置项默认用于您创建的自有APP, 如启用公版APP, 相关配置可同时用于自有APP和公版APP。

尚未创建自有APP (默认) 如您仅使用公版APP, 则无需创建。 [创建自有APP](#)

使用公版App控制产品 可以直接从应用市场下载公版App, 用于控制智能设备。



网关SDK中默认配网方式为蓝牙辅助配网，因此这里配网选择BLE辅助配网，备选方案可选择设备热点配网，如下图

The screenshot shows a configuration page with a sidebar on the left containing menu items: 产品展示 (Product Display), 分享方式 (Sharing Method), 设备面板 (Device Panel), 配网引导 (Network Configuration Guide), 多语言管理 (Multi-language Management), and 设备告警 (Device Alerts). The '配网引导' item is selected and highlighted. The main content area is titled '配网方案选择' (Network Configuration Selection). It includes a '产品自定义配网' (Product Custom Network) section with a sub-header '根据自身产品情况，配合阿里云IoT提供的Wi-Fi配网方案，设置优先进入的默认配网方案，以及当默认方案失败后的备选方案。' (According to the product situation, combine the Wi-Fi network configuration scheme provided by Alibaba Cloud IoT, set the default network configuration scheme that enters first, and the alternative scheme after the default scheme fails). Below this are two dropdown menus: '默认配网方式:' (Default Network Configuration Method) set to '蓝牙辅助配网' (Bluetooth-assisted network) and '备选配网方式:' (Alternative Network Configuration Method) set to '设备热点配网' (Device hotspot network). There is also a link for '零配方式 (需设备支持): 点击配置' (Zero-configuration method (requires device support): Click configuration). Under '配网入口:' (Network Configuration Entry), there are two checkboxes: '扫码/本地自动发现' (Scan code/local automatic discovery) which is checked, and '支持手动选择产品列表/搜索' (Support manual selection of product list/search) which is unchecked.

**设备调试** 页签：申请每个设备的Device Name及Device Secret信息。这两个信息连同上文提到的Product Key、Product Secret及Product Id共同构成设备的五元组信息。

The screenshot shows a dialog box titled '新增测试设备' (Add Test Device) with a close button (X) in the top right corner. Inside the dialog, there is a yellow warning box with a red exclamation mark icon. The text in the warning box reads: 'DeviceName可以是MAC地址、IMEI号或自定义SN等，须确保产品下唯一，为空将由系统自动颁发，您可以烧录到设备中，并上报到云端进行鉴权认证。使用蓝牙协议设备，需要使用Mac地址充当DeviceName，以确保设备的正常使用。' (DeviceName can be MAC address, IMEI number or custom SN, etc., must ensure uniqueness under the product, if empty it will be issued by the system automatically, you can burn it into the device and report it to the cloud for authentication. For Bluetooth protocol devices, you need to use Mac address as DeviceName to ensure the normal use of the device). Below the warning box is a label 'DeviceName' with a question mark icon. Underneath is a text input field with the placeholder text '请输入DeviceName，为空将由系统自动颁发' (Please enter DeviceName, if empty it will be issued by the system automatically). At the bottom right of the dialog are two buttons: '确定' (Confirm) and '取消' (Cancel).

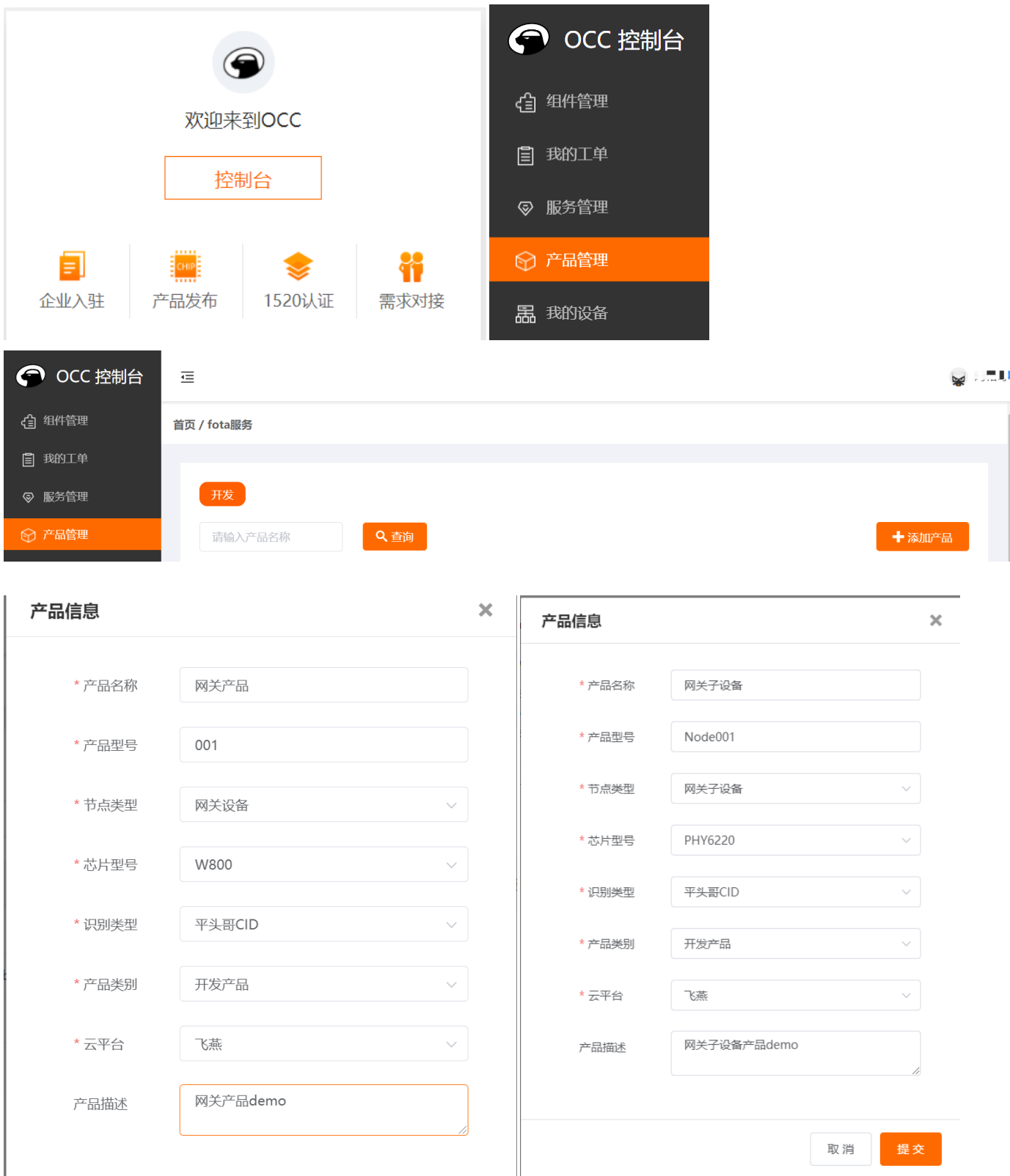
**注意：**

- 网关及子设备在申请五元组信息时，一定是以**设备mac地址**（小写无冒号）作为DeviceName。
- 网关获取mac地址可以通过cli输入wifi\_mac命令，子设备的mac地址需要使用4.1.4.2介绍的工具进行烧录
- 用户在第一次进入 **设备调试** 页签时需要选择 **认证模组/芯片**，用户可选择 **未认证**

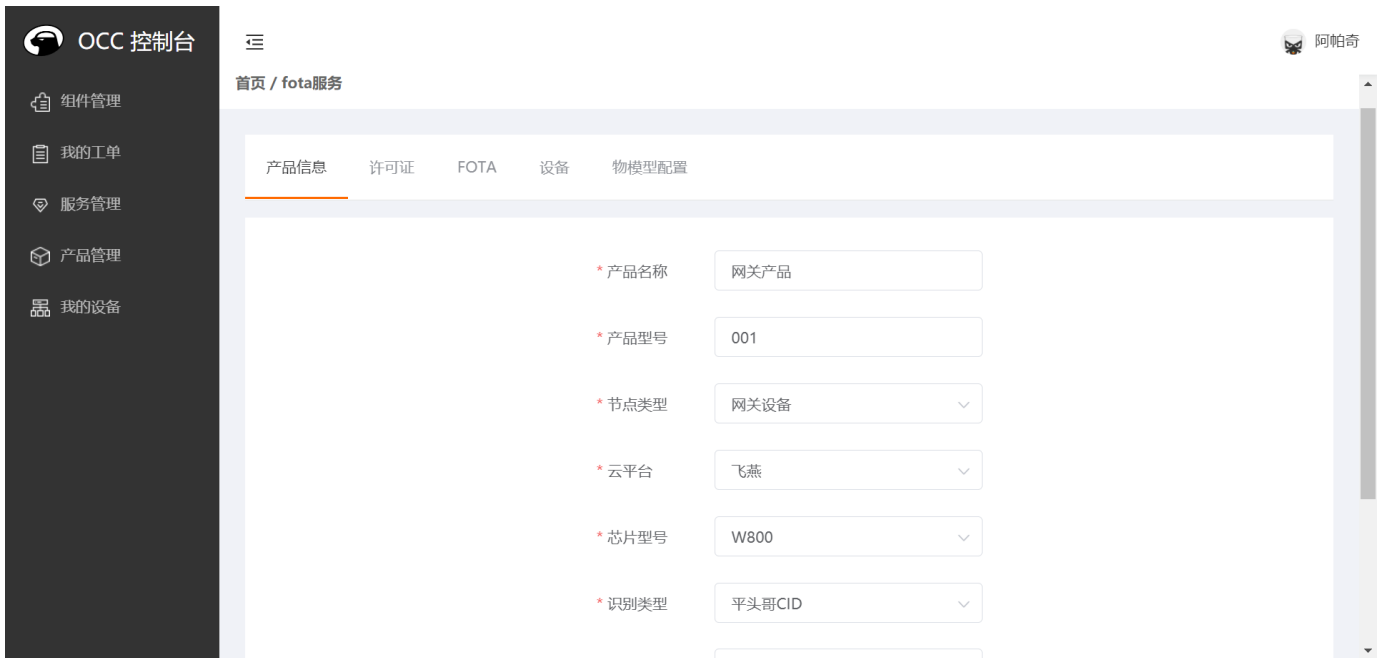
**批量投产** 页签：这里可以找到配网二维码。

## 5.1.2 在OCC创建网关及子设备产品

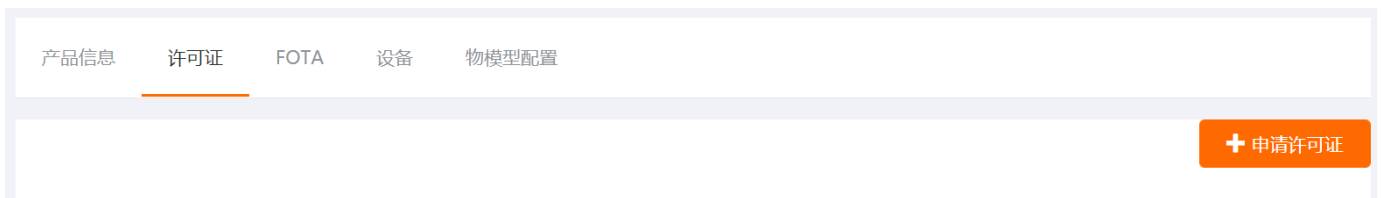
**创建产品：**登录**平头哥芯片开放社区**，进入控制台->产品管理->添加产品->填写产品信息，选择节点类型和对应的云平台



创建产品成功后，点击对应产品的“编辑”按钮进入产品页面进行配置。如下图所示，在配置页面可以对产品的许可证（KP）、FOTA及物模型等进行编辑和配置。



**许可证配置：**网关及子设备的许可证信息需要分别配置和申请。通过点击“编辑”按钮进入上述步骤创建的对应产品（如网关/子设备）后，即可申请对应设备的许可证信息。点击“许可证”->“申请许可证”



申请许可证之后，点击【生成】，将产品对应的MAC地址及5.1.1.2节得到的五元组信息填写到csv文件中（可参考平台示例），然后将编辑好的csv文件上传，即可生成许可证（KP）并进行下载。

许可证名	许可证描述	总量	审批状态	下载地址	操作
网关许可证	网关许可证	1	● 可用		<a href="#">编辑</a> <a href="#">生成</a> <a href="#">删除</a>

该类型产品需上传设备信息用于许可证生成 (示例)

device\_demo.csv

下载对应的KP文件夹后，将文件夹解压，在解压文件夹下pieces目录下可看到两个分别为Hex格式及二进制格式的文件，名称即为该设备的CID，如下图所示

名称	修改日期	类型	大小
5dacdb0904403000fd51d1bcc4dd96cd	2021/6/4 16:12	文件	1 KB
5dacdb0904403000fd51d1bcc4dd96cd.hex	2021/6/4 16:12	HEX 文件	2 KB

文件夹下二进制格式的文件就是我们用于烧录的KP文件，对于网关的KP，用户需要将烧录的kp文件rename为fcds，放到boards/loTGW\_CB800/bootimngs文件夹，然后按照4.1.4章节的介绍编译并进行烧录。

由于KP在生成时上传了云平台的鉴权信息（如生活物联网平台的五元组），因此烧录完成并重启后固件会自动加载对应的鉴权信息，联网后完成网关到云端的鉴权注册。

**FOTA配置：**网关及子设备产品均支持差分升级，因此FOTA在配置中需要添加当前设备烧录的固件作为差分基础固件。然后上传需要升级的固件，在该差分基础包的基础上生成差分升级包。FOTA配置的具体操作指南参考文档

- [FOTA服务-使用流程](#)

**物模型配置：**点击【产品管理】->选择产品，点击【编辑】->点击【物模型配置】页签，进入物模型配置页面。OCC提供了一个简单的js脚本编辑器。新建产品的脚本输入框中会自动填充物模型模版，在模版基础上可以根据实际场景开发。其中：`cloud_to_device`和`device_to_cloud`不能修改名称，参数dev是系统内置，在线调试不需要单独传入。编写好的脚本会自动存入草稿，也可以通过点击脚本编辑器下方的保存按钮，存入到草稿。编辑好物模型脚本后，可以对其进行简单测试：

a.模拟输入下，选择对应的模拟类型

b.输入模拟的设备上报数据或者设备接收数据，单击执行

在线调试脚本能正常可用后，单击提交，将脚本发布到OCC。

发布之后可以在脚本编辑器的右上方点击的草稿或线上运行脚本切换草稿和线上脚本。

编辑器界面如下：



编辑javascript(ECMAScript5)脚本 (当前展示为: 草稿, 您可以点击 [运行线上版本](#) 查看)

```
7 * example=> {"powerstate":1}->8202 01 01
8 * @param dev 内置全局参数, 无需输入
9 * @param cloud_data 下行数据, 不能为空
10 * @returns
11 */
12 cloud_to_device: function (dev, cloud_data) {}
13 // cloud_data may be a json-string, convert to a json object
14 var json = JSON.parse(cloud_data);
15 if (json) {
16 // convert to the device protocol
17 var data = todo_convert_to_device(json);
18 // send device protocol to the underlying physical link
19 return dev.send_to_device(data);
20 }
21
22 return -1;
23 },
24
25 /**
26 * convert the device protocol to the IoT cloud platform object
27 * model protocol and send it to the cloud
28 * example=> 8202 01 01->{"powerstate":1}
29 * @param dev 内置全局参数, 无需输入
30 * @param dev_data 上行数据, 不能为空
31 * @returns
```

模拟输入    运行结果

## 5.2 开发板上电

在完成上述产品创建步骤并正确烧录KP文件后，即可对开发板进行上电操作：

- 打开串口调试工具，配置波特率为 1000000 ， 数据位 8，校验位 None，停止位 1，流控 None
- 按开发板上的复位按键SW4，复位开发板，开发板正常上电log打印如下

```

1 Welcome boot2.0!
2 build: May 10 2021 21:22:51
3 load img & jump to [prim]
4 load&jump 0x8013000,0x8013000,1104524
5 all copy over..j 0x08013100
6 [ 0.893]<I>INIT Build:Jun 2 2021,16:18:12
7 [ 0.963]<I>INIT find 7 partitions
8 [ 1.832]<D>app 1.0.0
9
10 [ 1.949]<I>app enter nomal mode
11 [ 1.963]<D>occ already has triples, return
12 [ 1.965]<D>app_fota 1.0.0
13 [ 2.034]<D>app_net start net config
14 _wifi_pair_thread, 338
15 [ 2.045]<D>SL_PROV call ble config start
16 [BZ]no AC get from kv
17 [ 2.065]<D>app PWM led off.
18 [ 2.780]<I>AOSBT bt_init:No ID address. App must call settings_load()
19 [ 2.810]<I>AOSBT bt_dev_show_info:Identity: 00:11:00:11:00:11 (public)
20 [ 2.822]<I>AOSBT bt_dev_show_info:HCI: version 4.0 (0x06) revision 0x0120,
    manufacturer 0x0504
21 [ 2.838]<I>AOSBT bt_dev_show_info:LMP: version 5.0 (0x09) subver 0x0208
22 Bluetooth init succeed.
23 [BZ]Breeze adv per device
24 [BZ]Breeze unbind
25 [BZ]need awss

```

## 5.3 网络连接

### 5.3.1 设备配网

首次上电时，开发板没有配置 WiFi 用户名密码，无法连接无线路由器。此时，需要使用APP对设备进行配网，开发板目前默认配网方式为BLE辅助配网。配网需要在应用商城下载云智能APP，其开发和使用过程参考[BLE辅助配网最佳实践](#)及[WIFI配网概述](#)。

### 5.3.2 手动配置 WiFi 密码

用户也可以选择通过串口命令行指令快速配置 WiFi 密码，步骤如下：

- 串口中输入指令分别设置 WiFi 参数(ssid, password)，其中 {ssid} 为 wifi 名称，{password} 为 wifi 密码，如果AP配置为OPEN方式，则输入的密码会被忽略。

要以换行符 `\n` 或 `\r\n` 结尾

```

1 kv set wifi_ssid test
2 kv set wifi_psk 12345678

```

- 可以通过 `kv get wifi_ssid/wifi_psk` 来获取设置的ssid/密码，验证设置是否正确

Plain Text | 复制代码

```
1 > kv get wifi_ssid
2 kv get test
3 > kv get wifi_psk
4 kv get 123456
```

- 按开发板复位键，重启开发板

### 5.3.3 连接到网络

使用上述两种方式之一获取到AP的SSID/KEY之后，就会发起连接。等待并观察到串口打印输出 wifi 连接成功

Plain Text | 复制代码

```
1 [ 7.801532][I][netmgr ]start dhcp
2 [ 7.855701][I][netmgr ]IP: 192.168.1.103
```

也可以通过命令 `ifconfig` 检查网络连接状态，当网络连接成功，通过ifconfig会有如下信息

Plain Text | 复制代码

```
1 >ifconfig
2
3 wifi0 Link encap:WiFi HWaddr 28:6d:cd:63:bb:20
4 inet addr:172.16.2.182
5 GWaddr:172.16.2.1
6 Mask:255.255.254.0
7 DNS SERVER 0: 172.16.2.1
8
9 WiFi Connected to f8:8e:a1:39:29:e0 (on wifi0)
10 SSID: pingtougelIOTtest2G
11 channel: 1
12 signal: -31 dBm
```

此时可以ping通平头哥OCC社区

Plain Text | 复制代码

```
1 > ping occ.t-head.cn
2 ping occ.t-head.cn(203.119.214.112)
3 from 203.119.214.112: icmp_seq=1 time=38 ms
4 from 203.119.214.112: icmp_seq=2 time=37 ms
5 from 203.119.214.112: icmp_seq=3 time=39 ms
```

如果Wi-Fi连接失败，会显示“WiFi Not connected”

```

1 > ifconfig
2
3 wifi0 Link encap:WiFi HWaddr 28:6d:cd:54:49:c9
4     inet addr:0.0.0.0
5     Gwaddr:0.0.0.0
6     Mask:0.0.0.0
7     DNS SERVER 0: 208.67.222.222
8
9 WiFi Connected to 00:00:00:00:00:00 (on wifi0)
10  SSID: NETGEAR_HYJ
11  channel: 31
12  signal: -31 dBm
13
14  WiFi Not connected

```

## 5.4 添加mesh子设备

添加到网关的子设备，同样需要获取KP文件，其过程参考5.1.2章节的介绍及[PHY6220开发板快速上手手册](#)。在正确烧录KP的子设备处于配网状态时，用户可以使用网关的AT口手动输入AT指令进行配网，AT口波特率1000000，具体AT指令可参考[网关方案AT指令集](#)。

```

1 /*使能AT*/
2 AT+MESHAT=1
3 /*配置网关*/
4 AT+MESHPROVCONFIG=1,FF
5 AT+MESHPROVAUTOPROV
6 AT+MESHPROVEN=1
7 /*扫描节点*/
8 AT+MESHPROVSHOWDEV=1
9 /*扫描到节点后停止扫描节点*/
10 AT+MESHPROVSHOWDEV=0
11
12 /*绑定某个节点*/
13 AT+MESHADDDEV=00:E0:4C:00:00:03,00,9402820d00e04c000003b10600000000,00,01 /*这
    里mac地址和uuid等信息需要从scan到的信息中拷贝*/
14 /*配置节点*/
15 AT+MESHAUTOCONFIG=0x0002 /*这里需要是设备实际的unicast addr, unicast addr是设备在绑
    定后回复+MESHAPPKEYADD:0002,0中的第一个参数*/

```

上述配网过程也可以通过网关配套的上位机来实现，具体操作可以参考[蓝牙网关上位机快速上手手册](#)。

## 5.5 向生活物联网平台注册网关及子设备

当通过5.3介绍的步骤确认网络通畅后，网关会连接到生活物联网平台，关键打印信息如下



```

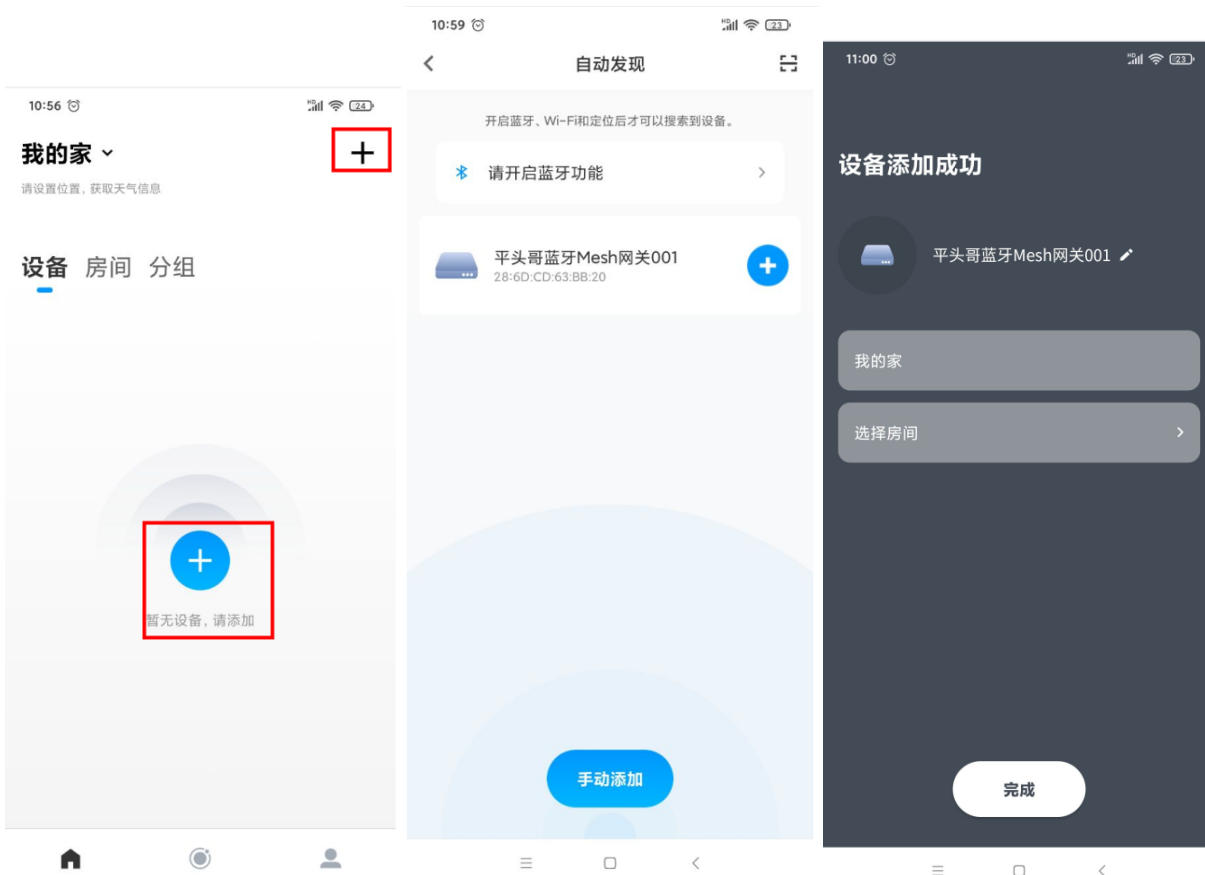
1  [prt] log level set as: [ 5 ]
2  [ 127.737]<A>AOS IOTX_CONN_CLOUD
3  .....
4      PK : a1N0xsTPVQM
5      DN : 286dcd63bb20
6      DS : 4df54539f0
7      PID : example.demo.partner-id
8      MID : example.demo.module-id
9      SM : TLS + Direct
10     TS : 2524608000000
11     .....
12     .....
13         Host : public.iot-as-mqtt.cn-shanghai.aliyuncs.com
14         Port : 1883
15     ClientID :
16     a1N0xsTPVQM.286dcd63bb20|securemode=2,tokenType=0,token=76547429348979C3BE431
17     608D41676E1,_v=sdk-c-
18     2.3.0_FY_1.6.6,timestamp=2524608000000,signmethod=hmacsha1,lan=C,pid=example.
19     demo.partner-id,mid=example.demo.module-id,authtype=custom-
20     ilop,_fy=1.6.6,_ss=1|
21         CA : ('... HMUfpIBvFSDJ3gyI ...')
22     .....
23 [ 127.930]<D>tls LD CA root Cert
24 [ 127.962]<D>tls SSL/TLS struct
25 [ 127.979]<I>tls Conn /public.iot-as-mqtt.cn-shanghai.aliyuncs.com/1883
26 [ 128.078]<D>tls Handshake
27 [ 130.750]<D>tls Verify X.509
28 [ 130.761]<I>tls certverify ret 0x00
29 [ 131.051]<I>gateway Device Initialized, Devid: 0
30 [ 132.200]<I>gateway Cloud Connected
31 [ 132.208]<I>gateway do update subdev

```

其中，PK/DN/DS等为五元组信息，在5.1.1章节描述的创建产品过程中由生活物联网平台分配，在申请的KP中已经包含，无需用户再次烧录。

若网关是通过5.3.1介绍的配网方式连接到AP，那么在网关连接云端成功后app上就会出现网关设备；若网关通过5.3.2介绍的手动输入ssid/key的方式连接到AP，那么网关连接到云端后，还需要在app上本地搜索并添加网关。

**注意：** app上本地搜索并添加网关时要确保手机和网关连在同一个AP，才能保证本地发现



完成上述网关注册后，就可以通过云智能APP扫描在5.1.1章节介绍下创建的子设备的二维码，对设备进行注册。注册成功后，即可在app上显示该子设备，并通过app进行控制。



## 附录 使用Linux开发环境

### 1 安装YocTools

YocTools需要通过 pip 命令安装，YocTools 支持 python2.7+、python3.6+，建议使用 python3 作为默认 python。

- 卸载旧版本YocTools，之前如果没安装请跳过该步骤

```
1 $sudo pip uninstall yoctools
```

Shell | 复制代码

- 安装YocTools

```
1 $sudo pip install yoctools
```

Shell | 复制代码

安装完成后使用命令 `yoc -V` 查看yoctools版本，用户安装的版本应大于2.0.12版本

### 2 工具链安装

编译工具链采用gcc编译，这里采用csky-elfabiv2-gcc 工具链，安装方法如下：

- 解压工具链包csky-elfabiv2-tools.tar.gz 到Linux系统路径，这里放到了/opt目录下
- 配置工具链路径到环境变量：

Shell | 复制代码

```
1 $echo "PATH=$PATH:/opt/csky-elfabiv2-tools/bin" >> ~/.bashrc
2 $source ~/.bashrc
```

安装完成后使用命令csky-elfabiv2-gcc -v查看工具链版本，网关SDK统一使用C-SKY Tools V3.10.21 Minilibc abiv2 B20200407版本作为开发工具链

### 3 调试环境安装

#### 安装Debug Server

- 登录[平头哥芯片开放社区](#)，进入栏目“开发中心->资源下载->工具 -> Debug Server”，下载最新CSKY-DebugServer-Linux版本，并解压
- 执行 `sudo sh ./CSKY-DebugServer-linux-*.sh -i`，开始安装

注意：安装时需要sudo权限

- 系统提示 `Do you agree to install the DebugServer[yes/no]`，输入 `yes`
- 系统提示设置安装路径 `Set full installing path:`，推荐安装到默认路径 -- 直接回车
- 系统会提示 `This software will be installed to the default path: (/usr/bin/)?[yes/no/cancel]:`，输入 `yes`
- 安装成功后会提示：  
Done !  
You can use command “DebugServerConsole” to start DebugServerConsole!  
(NOTE: The full path of ‘DebugServerConsole.elf’ is /usr/bin/C-Sky\_DebugServer)

#### Linux虚拟机下安装Debug Server

很多开发者习惯在Windows系统下安装Linux虚拟机来进行开发，常见的虚拟机有WSL（Windows Subsystem for Linux），VMware，VirtualBox等。然而在Linux虚拟机下，由于Debug Server没办法获取到USB设备信息，导致部分情况下Debug Server连接失败。

因此不支持在Linux虚拟机下安装Debug Server，需要切换到Windows环境安装。

#### 运行Debug Server

- 安装完成后，在任意目录下通过命令 `sudo DebugServerConsole`，来打开 DebugServer。打开后界面如下：

```

xie@xie-VirtualBox:~/workspace/test$ DebugServerConsole
+---+
| C-Sky Debugger Server (Build: Feb 19 2020) |
| User Layer Version : 5.8.18 |
| Target Layer version : 2.0 |
| Copyright (C) 2019 Hangzhou C-SKY Microsystems co.,ltd |
+---+
C-SKY: CKLink_Lite_V2, App_ver 2.16, Bit_ver null, Clock 2526.316KHz,
      2-wire, With DDC, Cache Flush On.
+-- CPU 0 --+
C-SKY CPU ID:
      WORD[0]: 0x04a11453
      WORD[1]: 0x10000000
      WORD[2]: 0x21400417
      WORD[3]: 0x30c00006
Target Chip Info:
      CPU Type is CK804FGT, in LITTLE Endian.
      L1ICache size 32KByte.
      Bus type is AHB-LITE32.
      Signoff date is 04/0107.
      Foundry is SMIC.
      Process is 55nm.

```

- DebugServer连接成功如下:

```

***** DebuggerServer Commands List *****
singlestep/si
      execute single-step in the target
reset
      reset the target
pctrace
      show the PCFIFO(8 <= length <= 4096, default 8)
print/p
      print /x[d/f/o] *memory[$registers], eg p /x *0x20000000
      print target
set
      Set *memory[$registers]=value, eg. set $r0=0x1234
quit/q
      quit Debugger Server
help/h
      show help informations
CTRL+B ENTER
      switch input channel
*****
DebuggerServer$ Connection from 127.0.0.1 for CPU 0, at time 12:45.

```

## 4 编译烧录

- 编译

进入开发板目录，使用 `make` 命令开始编译：

```
Shell | 复制代码
1 $ cd solutions/ble_mesh_gateway_demo
2 $ make clean; make
```

- 烧录

## 全部烧录

- 上电开发板，确保红色电源指示灯被点亮；运行Debug Server并保证与开发板连接成功
- 首次运行时，需要执行全部烧录命令，保证所有分区内容都被烧录：

```
Plain Text | 复制代码
1 $ make flashall
```

注意：Bootloader烧写若异常断电可能会导致设备无法引导

第一次烧录时，会提示输入IP地址和端口号，则需要进行如下操作

- 输入命令 `ifconfig` 获取本机IP地址：

```
Shell | 复制代码
1 $ ifconfig
2 wifi0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 30.21.178.4 netmask 255.255.252.0 broadcast 30.21.179.255
4     inet6 fe80::b031:5c7e:9a91:d8d2 prefixlen 64 scopeid
5     0xfd<compat,link,site,host>
6     ether 60:f2:62:77:45:d9 (Ethernet)
7     RX packets 0 bytes 0 (0.0 B)
8     RX errors 0 dropped 0 overruns 0 frame 0
9     TX packets 0 bytes 0 (0.0 B)
10    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- 本机IP地址（本例：30.21.178.4），端口号 `1025`，则输入如下命令可进行烧录

```
Shell | 复制代码
1 $ make flashall
2 Input gdb DebugServer IP:PORT :30.21.178.4:1025
```

注意：烧录完成后，当前目录下会自动生成 `.gdbinit` 文件，以后烧录时，若本机IP地址无变化，无需更新 `.gdbinit` 文件，直接烧录即可；若本机IP地址变化，需要更新 `.gdbinit` 文件内的IP地址。

## 应用固件烧录

- 大部分应用程序都运行应用分区上，修改应用程序代码后可以只烧写相应的应用程序固件，加快烧写速度。

Plain Text | 复制代码

```
1 $ make flash
```

## 烧录成功

- 烧录成功后，可以看到所有分区烧写进度都已至100%。

Plain Text | 复制代码

```
1 Program partition: prim          address: 0x80d0400, size 664816 byte
2 erasing...
3 program 08172400, 100%
4 Program partition: imtb         address: 0x81c0000, size 4096 byte
5 erasing...
6 program 0081c1000, 100%
```