

ALIBABA CLOUD

阿里云

生活物联网平台
快速入门

文档版本：20201015

 阿里云

法律声明

阿里云提醒您阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
<code>Courier</code> 字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
<i>斜体</i>	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1. 开发自有品牌项目的产品	05
1.1. 概述	05
1.2. 创建项目	05
1.3. 创建产品并定义功能	06
1.4. 配置App	06
1.5. 添加设备	08
1.6. 开发设备	09
1.7. 调试设备连云	12
2. 开发天猫精灵生态的产品	16
2.1. 概述	16
2.2. 创建项目	16
2.3. 创建产品并定义产品功能	17
2.4. 配置App	18
2.5. 添加设备	18
2.6. 开发设备	19
2.7. 调试设备连云	24

1. 开发自有品牌项目的产品

1.1. 概述

生活物联网平台是阿里云IoT针对生活领域推出的物联网平台，以解决家电智能化的问题。本文带您了解生活物联网平台的简要使用流程。

原理图



准备工作

- 请确保您已经注册了阿里云账号，并完成实名认证。注册操作请参见[阿里云账号注册流程](#)。
- 请确保账号已开通[生活物联网平台](#)服务。
- 请安装好设备固件开发所需的Linux开发环境，建议使用64位Ubuntu 16.04开发环境。
- 请安装好用于烧录设备证书和固件的串口烧录调试工具。设备使用SDK和证书接入生活物联网平台。由于各类设备的烧录方式略有差异，本文以烧录经平台认证的MK3060模组为例，安装了SecureCRT串口烧录工具。

操作步骤

1. **创建项目**：项目不仅便于您管理产品，还可以实现多方协同工作。
2. **创建产品并定义功能**：产品相当于同类设备的集合，例如，产品可以是某种型号的设备。您可以通过属性、服务和事件三个维度定义产品的功能。平台将根据您定义的功能构建出产品的数据模型，用于云端与设备端的数据通信。
3. **配置App**：当前生活物联网领域，消费者通常使用App绑定并控制设备。平台提供App服务，您可以通过配置App参数项，轻松实现人机互动。
4. **添加设备**：设备指某个具体设备。每个设备拥有自己的设备证书，用于连接生活物联网平台。平台提供测试设备，测试设备的证书不能用于量产，仅供调试使用。
5. **开发设备**：平台提供设备端SDK，通过简单开发，设备即可具备上云能力。
6. **调试设备连云**：公版App连接设备后，通过App和控制台（云端）调试真实设备，验证设备端、云端、App端，三端上下行数据通信。

1.2. 创建项目

使用生活物联网平台的第一步是创建项目。项目间数据相互隔离，您可以为不同客户设置不同项目，或将某个项目独立授权给其他阿里云账号，方便多方协同工作或运营中心独立交付。

创建项目 物联网平台 飞燕

操作步骤

1. 登录[生活物联网平台](#)的控制台。
2. 单击创建项目。
3. 配置项目名称。

下图为配置示例，实际使用中建议配置为您的客户名称。方便后期项目单独转发、授权或统计时易于辨别。



- 单击**确定完成**。项目创建成功后，自动跳转至项目主页面。

后续步骤

创建产品并定义功能

1.3. 创建产品并定义功能

产品相当于一类设备的集合，同一产品下的设备具有相同的功能。每个项目中，可以新建多个产品。产品创建完成后，您可以通过定义产品功能将实际产品抽象成由属性、服务、事件所组成的数据模型，便于云端管理和数据交互。

前提条件

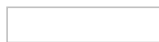
已完成项目的创建，详细操作请参见[创建项目](#)。

一、创建产品

- 在项目主页面，单击**创建新产品**。
- 配置产品参数。

产品配置的参数介绍请参见[创建产品](#)。

此处以创建Wi-Fi智能插座为例，参数配置如下图所示。



- 单击**确认完成**。产品创建成功后，自动进入产品的功能定义页面。

二、定义产品功能

平台为每个品类提供了默认标准功能。创建产品后，功能定义页面自动显示该产品默认的标准功能。

如果默认显示的标准功能已满足您的需求，可单击下一步跳过以下操作步骤。


如果不能满足您的需求，您可以根据以下步骤添加或修改功能。

- 在产品的功能定义页面，单击**标准功能**对应的**添加功能**。



- 单击左侧选择功能区域中的功能类型，系统自动添加到右侧的已选功能中。



 **说明** 如果该品类下的标准功能不能满足您的需求，可以在其他类型中搜索类似功能。如果依然没有符合需求的，建议自定义功能类型，具体操作请参见[新增自定义功能](#)。

- 单击**确定完成**。

后续步骤

配置App

1.4. 配置App

在智能生活领域，终端消费者往往通过App或语音控制与实际设备互动。生活物联网平台提供了App服务，简化了App开发任务。您可以通过简单配置，实现App与平台间的数据通信，达到人机互动的效果。

操作步骤

1. 单击下一步，进入产品的人机交互页面。

2. 选择App版本。

平台支持两种交互终端：公版免开发App和自有品牌App（需根据平台提供的SDK、插件自行开发）。

本示例选用公版App控制设备。您需要打开使用公版App控制产品的开关。

3. 在左侧导航栏中选择分享方式，配置App的分享方式。在使用App绑定设备时，设备的归属关系受该分享方式约束。由于公版App对设备分享的安全要求，公版App默认为授权式，且不可更改。

4. 在左侧导航栏中选择设备面板，配置产品的App中的面板显示。

i. 单击未设置，在弹出的选择产品面板对话框中选择任一主题面板。

ii. 安装公版App“云智能”。

单击立即安装，并选择开发版页签，然后扫码下载开发版的公版App。

说明

公版App分为开发版与用户版，两者区别如下。

- 开发版App供开发者在设备开发期间调试设备使用，只可以扫码下载。
- 用户版App供消费者使用，可以扫码下载，也可以在全球主流应用商店中搜索“云智能”下载。用户版根据区域还分为全球版、国内版、海外版，扫描二维码后跳转至相应的应用商店。

iii. （可选）使用云智能App扫描界面预览处的二维码，您可以体验产品界面。

5. 在左侧导航栏中选择产品展示，配置产品在App中展示的图标和产品名称。

i. （可选）单击更换图标，修改产品在App中的展示图标。

ii. 配置产品在App中的中文展示名称。

产品在公版App中按“品牌+名称+型号”的样式展示，产品名称的配置规则如下。

- 支持中文、英文字母、数字、连接号（-）、at符号（@）、中文圆括号和英文圆括号。
- 内容不能含有“测试”字样。
- 不能超过30个字符（一个中文算1个字符）。

说明

在该页面配置的产品名称会自动同步到App多语言页面的中文页签中。

iii. 单击保存完成。

6. 在左侧导航栏中选择配网引导，配置设备在App上的配网引导页面。

选择公版App，且品类为灯、插座、开关时，支持以下两种配网方式。

- 品类标准配网

品类标准配网的配网引导文案不可修改，您只需配置产品的配网类型和配网入口即可。

o 自定义配网

不想使用品类标准配网时，您可以单击配网引导右上角的我要自定义配网，切换到自定义配网方式。

本示例以品类标准配网为例，请您根据以下步骤来操作。如果您需要自定义配网请参见配置自有品牌项目下的自定义配网。

- i. 配置标准配网类型。平台支持Wi-Fi和BLE+Wi-Fi两种方式，请根据产品的实际连网方式来选择。
- ii. (可选) 配置产品在公版App中的配网入口。

品类标准配网的默认配网入口为App扫码和本地自动发现，您还可以选中手动选择产品列表/搜索复选框，为产品增加品类配网入口。

当您选中手动选择产品列表/搜索复选框时，您配置的产品名称需符合规范，设备真实量产时产品名称需经过平台审核。审核通过后产品名称才可以在公版App的配网列表中显示。

iii. 单击确认并保存设置完成。

7. 在左侧导航栏中选择多语言管理，配置多种语言展示在App中的产品名称和产品功能。

i. 配置产品名称。

说明 如果在产品展示页面中已配置了产品名称，此处会自动显示。如果此处修改后，产品展示页面中也会同步更新。

设备真实量产时，品类为灯、插座、开关，且配网引导选中手动选择产品列表/搜索复选框时，配置的产品名称需经过审核后才可以使用。如果审核不通过，需要重新修改后再提交审核，直至审核通过。

ii. 配置产品功能的名称。产品功能分为简单功能和复杂功能，您需要配置各属性值在App上显示的功能名称。

品类为灯、插座、开关时，平台为您自动生成默认的功能名称，您可以直接使用。其余的品类则需要您自行配置。

iii. 单击保存完成。

iv. (可选) 如果您需要切换App语言，请切换至相应语言的页签，同样方法配置产品名称和产品功能即可。

8. (可选) 配置其余的App参数项。

本示例中仅配置了选择公版App时必须填的参数项，其余参数项的介绍和配置方法请参见配置人机交互。

后续步骤

添加设备

1.5. 添加设备

设备必须使用平台颁发的唯一设备证书 (ProductKey、DeviceName、DeviceSecret) 才能接入生活物联网平台。实际生产中，您可以购买激活码以获取设备证书。考虑到量产前的测试需求，平台为每个产品免费提供50个测试设备。测试设备的设备证书仅供产品开发阶段测试使用，无法用于量产。

操作步骤

1. 单击下一步，进入产品的设备调试页面。
2. 在选择认证模组/芯片中选择相应的模组，建议选用阿里已认证的模组。
3. 在测试设备中单击新增测试设备。
4. 配置设备名称。

DeviceName也可以不配置，由系统自动生成设备名称。

5. 单击确定，界面显示测试设备的设备证书。

需要烧录到设备中的激活凭证（ProductKey、ProductSecret、DeviceName、DeviceSecret）如下图红色框所示。

量产设备获取设备证书的方法请参见[下载设备证书](#)。

后续步骤

开发设备

1.6. 开发设备


生活物联网平台提供了已封装设备端与物联网平台交互协议的SDK。您可以直接基于这些SDK开发真实设备的固件，从而使真实设备具备连接物联网的能力。以基于AliOS Things的V1.1.0 SDK开发MK3060模组（平台已认证模组）为例介绍设备固件的开发和烧录操作。

前提条件

已完成开发环境的搭建和依赖包的安装，详细请参见[准备开发环境](#)。

一、开发设备固件

1. 下载设备端SDK代码，请参见[获取生活物联网SDK](#)。

 **说明** 本操作以含AliOS Things的V1.1.0版本生活物联网SDK为例，正式开发设备时，建议您选用最新版本SDK。

2. 将下载的ZIP包上传至开发环境中，并解压ZIP包。
3. 配置设备的身份信息，修改完成后保存退出。设备证书是平台颁发给设备的唯一身份信息，需要将`example/linkkitapp/linkkit_example_solo.c`中的demo设备证书修改为[添加设备](#)中待烧录的设备证书。
 - i. 进入SDK根目录。
 - ii. 执行 `vim example/linkkitapp/linkkit_example_solo.c` 命令修改`linkkit_example_solo.c`文件。
 - iii. 按 `i` 进入编辑模式。

iv. 修改文件中的如下参数。

```
// for demo only
#define PRODUCT_KEY   "a1U****Dgg"
#define PRODUCT_SECRET "sW*****o9B"
#define DEVICE_NAME   "kE*****6g70"
#define DEVICE_SECRET "RWY*****h005"
```

v. 按Esc键退出编辑模式，再输入 `:wq` 命令保存并关闭文件。

4. 在SDK根目录，执行编译命令。

本文在MK3060模组上编译linkkitapp程序。

```
aos make linkkitapp@mk3060
```

编译完成后，在 `out\linkkitapp@mk3060\binary\` 目录下会生成 `linkkitapp@mk3060_crc.bin` 文件。该文件为需要烧录到真实设备中的固件。

二、烧录设备固件

使用USB线将MK3060与电脑连接。此时电脑会提示安装驱动，完成驱动的安装。本文以Windows电脑为例。

1. 右击我的电脑，选择管理 > 系统工具 > 设备管理器 > 端口（COM和LPT），查看串口号。

 说明 不同系统下打开设备管理器的操作路径略有差异，请根据您的电脑的实际路径操作。

2. 打开串口调试工具SecureCRT，选择文件 > 快速连接。

3. 选择协议为Serial和设备对应串口号，设置波特率（3060模组波特率为921600），并单击连接。

4. 将拨码开关拨到如下所示位置（1置于ON，2置于OFF），按开发板上reset键，初始化MK3060。

SecureCRT中出现以下日志，则进入初始化状态。

```

=====
|          MOC108 BOOTLOADER MENU          |
|-----|
| Comamnd | Arguments    | Description    |
|-----+-----+-----|
| read   | <address> <size> | read flash    |
|-----+-----+-----|
| write  | <address>        | write flash   |
|-----+-----+-----|
| erase  | <address> <size> | erase flash   |
|-----+-----+-----|
| boot   | <mode>           | boot to APP, ATE or QC |
|-----+-----+-----|
| reboot |                  | reboot        |
|-----+-----+-----|
=====

@ Author : Snow Yang
@ Version : 1.1.0
@ Date   : Oct 12 2019 15:01:24
$

```

5. 输入 `write 0x13200` , 进入烧录模式。

```

waiting for the file to sent ...(press 'a' to abort)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

6. 选择传输 > 发送Ymodem, 选择下载到本地的固件, 单击添加, 并单击确定开始烧录固件。

烧录成功后, 出现如下提示。

```

Programing Completed Successfully!

```

7. 将拨码开关拨到如下所示位置 (1置于OFF, 2置于OFF) , 按reset键, 进入正常启动模式。

```
enter low level!
mac fc:ee:e6:xx:xx:ce
leave low level!
app_init finished
start-----hal
trace config close!!!
[000008]<A> aos framework init.
-----Firmware info-----
Host: qiheng.hc
Branch: rel_1.0.1
Hash: 86ae9xxxxxea6cb96a369eb603254
Date: Oct 12 2019 15:07:07
Kernel: AOS-R-1.3.4
LinkKit: 2.3.0
APP: app-1.0.0-20191012150607
MQTT direct: on
Region env: MAINLAND-ONLINE

[000012]<A>pk[a1U****Dgg]
[000012]<A> ps[sw*****o9B]
[000012]<A> dn[kE*****6g70]
[000014]<A> ds[RWY*****h005]
```

8. 检查打印日志中的设备凭证（ProductKey、ProductSecret、DeviceName、DeviceSecret）是否与[添加设备](#)中生成的设备凭证一致。如果不正确可以通过以下命令修改。

```
linkkey <ProductKey> <DeviceName> <DeviceSecret> <ProductSecret>
```

此时设备进入待配网状态，可通过App连接设备。

后续步骤

调试设备连云

1.7. 调试设备连云

公版App连接真实设备后，通过App和控制台（云端）调试真实设备，并在控制台上管理您的设备，验证设备端、云端、App端，三端上下行数据通信。

前提条件

完成云智能App的账号注册。

操作步骤

1. 打开云智能App，选择设备当前所在的国家/地区。
2. 设备按reset键，当SecureCRT工具出现以下日志，设备进入待配网状态。

```
INFO: network_start no valied ap confing
INFO: power_cycle_awss power cycle count: 1
INFO: check_and_start_awss awss type: one key
[000020]<A> do_awss_active 0

[000020]<F> enable awss

[000022]<A> awss_close_dev_ap exit

[002022]<A> IOTX_AWSS_START
[002022]<I> zconfig_init

Soft_AP_start
[002034]<F> chan 1
```

3. 登录云智能App。
4. 单击右上角的加号来添加设备，并对设备配网。

```

[022862]<I> MQTT init success!
[022864]<I> Loading the CA root certificate ...
[022868]<I> ok (0 skipped)
[022868]<I> Connecting to /a1U****Dgg.iot-as-mqtt.cn-shanghai.aliyuncs.com/1883...
rate:4, pwr_gain:7
add extral movement in test
[022926]<I> ok
[022926]<I> . Setting up the SSL/TLS structure...
[022928]<I> ok
[022928]<I> Performing the SSL/TLS handshake...
[023012]<I> AP connected
[023012]<D> awss connect ssid:abc00 success //成功连接名称为abc00的WiFi
[023012]<A> IOTX_AWSS_GOT_IP
[023012]<D> -----AWSS STATIS-----
[023012]<D> name          max   min  mean  cnt   suc  crc-err passwd-err
[023012]<D> ConnRouter      5418 5418 5418  1    1    0      0
[023014]<D> Smartconfig     2002 2002 2002  1    1    0      0
[023014]<D> Smartconfig-wps  0     0    0     0    0    0      0
[023016]<D> Aha             0     0    0     0    0    0      0
[023016]<D> Zconfig        0     0    0     0    0    0      0
[023016]<D> Dev-ap         0     0    0     0    0    0      0
[023016]<D> -----
[023018]<I> The CoAP Server already init
[023018]<D> CoAPResource_register, context:0x41ee30, new node
[023018]<D> Register new resource /sys/awss/device/connectap/info/get success, count: 5
[023020]<D> CoAPResource_register, context:0x41ee30, new node
[023020]<D> Register new resource /sys/a1U****Dgg/kEa*****6g70/awss/device/connectap/info/get success, count: 6

```

设备连网成功后，在产品的设备调试页面，测试设备的状态显示为在线。



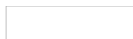
5. 调试真实设备。不仅可以通过App控制真实设备（您可以直接通过手机体验App端控制设备，此处不作介绍），还可以通过控制台调试真实设备，具体操作步骤如下。

- i. 在产品的设备调试页面，单击操作中的调试。
- ii. 选择调试真实设备页签。
- iii. 选择调试功能和方法，并单击发送指令。右侧显示设备的实时日志。



6. 管理您的设备。

您还可以在控制台项目主页面的运营中心里，管理您的真实设备和设备用户。详细介绍请参见[运营中心概述](#)。



2. 开发天猫精灵生态的产品

2.1. 概述

生活物联网平台是阿里云IoT针对生活领域推出的物联网平台，以解决家电智能化的问题。本文以开发蓝牙Mesh设备为例，带您了解生活物联网平台的简要使用流程。

原理图

天猫精灵生态项目下的产品按照是否需要通过网关入网分为以下两种情况，原理图如下所示。

- 直连设备

仅支持Wi-Fi方式直接连接生活物联网平台。



- 非直连设备

仅支持蓝牙方式（分为蓝牙Mesh和蓝牙GATT）通过网关连接生活物联网平台。



准备工作

以调试蓝牙Mesh设备的TG7100开发板为例（如需获取该开发板请通过[工单](#)联系我们），您需要准备以下内容：

- 请确保您已经注册了阿里云账号，并完成实名认证。注册操作请参见[阿里云账号注册流程](#)。
- 请确保账号已开通[生活物联网平台](#)服务。
- 请安装好设备固件开发所需的Linux开发环境，建议使用64位Ubuntu 16.04开发环境。
- 请安装好用于烧录固件和设备证书的串口工具。设备使用SDK和设备证书接入生活物联网平台。

由于各类设备的烧录方式略有差异，本文以操作平台认证的TG7100开发板为例，请您安装好用于烧录固件的TG71XX Programmer串口烧录工具和用于烧录设备证书的SecureCRT串口烧录工具。

操作步骤

1. **创建项目**：项目不仅便于您管理产品，还可以实现多方协同工作。
2. **创建产品并定义产品功能**：产品相当于同类设备的集合，例如，产品可以是某种型号的设备。您可以通过属性、服务和事件三个维度定义产品的功能。平台将根据您定义的功能构建出产品的数据模型，用于云端与设备端的数据通信。
3. **配置App**：在智能生活领域，终端消费者往往通过App或语音控制与实际设备互动。天猫精灵生态项目下的产品固定选用天猫精灵App控制设备。您可以通过简单配置，实现人机互动的效果。
4. **添加设备**：设备指某个具体设备。每个设备拥有自己的设备证书，用于连接生活物联网平台。平台提供测试设备，测试设备的证书不能用于量产，仅供调试使用。
5. **开发设备**：平台提供设备端SDK，通过简单开发，设备即可具备上云能力。
6. **调试设备连云**：天猫精灵App连接设备后，通过App和控制台（云端）调试真实设备，验证设备端、云端、App端，三端上下行数据通信。

2.2. 创建项目

使用生活物联网平台的第一步是创建项目。项目间数据相互隔离，您可以为不同客户设置不同项目，或将某个项目独立授权给其他阿里云账号，方便多方协同工作或运营中心独立交付。

操作步骤

1. 登录[生活物联网平台](#)的控制台。
2. 单击创建项目。
3. 配置项目名称，并选择项目类型为天猫精灵生态项目。

下图为配置示例，实际使用中项目名称建议配置为您的客户名称。方便后期项目单独转发、授权或统计时易于辨别。



4. 选中我同意天猫精灵（阿里云账号“tmallgenie_iot”）创建和拥有该项目下的所有产品数据复选框。
5. 单击确定完成。项目创建成功后，自动跳转至项目主页面。

后续步骤

创建产品并定义产品功能

2.3. 创建产品并定义产品功能

产品相当于一类设备的集合，同一产品下的设备具有相同的功能。每个项目中，可以新建多个产品。产品创建完成后，您可以通过定义产品功能将实际产品抽象成由属性、服务、事件所组成的数据模型，便于云端管理和数据交互。

一、创建产品

1. 在项目主页面，单击创建新产品。
2. 配置产品参数。

产品配置的参数介绍请参见[创建产品](#)。

此处以创建蓝牙Mesh的智能灯为例，参数配置如下图所示。



3. 单击确认完成。产品创建成功后，自动进入产品的功能定义页面。

二、定义产品功能

平台为每个品类提供了默认标准功能。创建产品后，功能定义页面自动显示该产品默认的标准功能。

如果默认显示的标准功能已满足您的需求，可单击下一步跳过以下操作步骤。

如果不能满足您的需求，您可以根据以下步骤添加或修改功能。

1. 在产品的功能定义页面，单击添加功能。



2. 单击左侧选择功能区域中的功能类型，系统自动添加到右侧的已选功能中。



3. 单击确定完成。

后续步骤

配置App

2.4. 配置App

在智能生活领域，终端消费者往往通过App或语音控制与实际设备互动。天猫精灵生态项目下的产品固定选用天猫精灵App控制设备。您可以通过简单配置，实现人机互动的效果。

操作步骤

1. 单击下一步，进入产品的人机交互页面。
2. 在左侧导航栏中选择产品展示，配置产品在天猫精灵App中展示的图标和产品名称。



- i. 单击上传图片，选择产品展示的图标。

上传的App展示图需符合一定的视觉规范，详细信息请参见[产品展示图规范](#)。

- ii. 配置产品在App中的中文展示名称。

产品在公版App中按“品牌+名称+型号”的样式展示，产品名称的配置规则如下。

- 支持中文、英文字母、数字、连接号(-)、at符号(@)、中文圆括号和英文圆括号。
- 内容不能含有“测试”字样。
- 不能超过30个字符（一个中文算1个字符）。

说明 在该页面配置的产品名称会自动同步到App多语言页面的中文页签中。

- iii. 单击保存完成。

- iv. （可选）单击页面右上角的配网+App下载二维码，扫描二维码下载天猫精灵App。该二维码是二合一的二维码，您可以通过该二维码下载天猫精灵App，您还可以通过天猫精灵App扫描该二维码为设备配网。

3. 在左侧导航栏中选择设备面板，配置产品的天猫精灵App中的面板显示。

平台支持两种面板：默认控制面板和上传自主开发控制面板（需自行开发面板界面）。本示例以默认控制面板为例，您只需选择默认控制面板并单击确定即可。



4. 在左侧导航栏中选择配网引导，配置设备在天猫精灵App中的配网引导页面。

天猫精灵生态项目下的产品仅支持一键配网方式。您只需按照图中提示配置产品的配网引导图和相关提示文案即可。



5. （可选）配置其余的App参数项。

本示例中仅配置了选择公版App时必须填的参数项，其余参数项的介绍和配置方法请参见[配置人机交互](#)。

后续步骤

添加设备

2.5. 添加设备

设备必须使用平台颁发的唯一设备证书（ProductKey、DeviceName、DeviceSecret）才能接入生活物联网平台。实际生产中，您可以购买激活码以获取设备证书。考虑到量产前的测试需求，平台为每个产品免费提供50个测试设备。测试设备的设备证书仅供产品开发阶段测试使用，无法用于量产。

操作步骤

1. 单击下一步，进入产品的设备调试页面。
2. 在选择认证模组/芯片中选择相应的模组，建议选用阿里已认证的模组。
3. 在测试设备中单击新增测试设备。
4. 配置设备名称。

对于蓝牙Mesh产品，DeviceName需与设备的MAC一致。如果您手动输入DeviceName则需满足MAC地址格式要求。一般建议为空由系统自动生成设备名称。

5. 单击确定，界面显示测试设备的设备证书。

蓝牙Mesh产品需要烧录到设备中的激活凭证（ProductId、DeviceName、DeviceSecret）如下图红色框所示。

量产设备获取设备证书的方法请参见[下载设备证书](#)。

后续步骤

开发设备

2.6. 开发设备

生活物联网平台提供了已封装设备端与物联网平台交互协议的SDK。您可以直接基于这些SDK开发真实设备的固件，从而使真实设备具备连接物联网的能力。以基于AliOS Things的蓝牙Mesh SDK（V1.2.0）调试TG7100开发板为例，介绍设备固件的开发和烧录操作。

前提条件

已完成开发环境的搭建和依赖包的安装，详细请参见[准备开发环境](#)。

一、准备开发环境

1. 搭建SDK的开发环境。

建议您在64位Ubuntu下搭建设备端SDK的开发环境，并使用vim编辑代码。如果您是Win10操作系统可以在应用商店下载安装Ubuntu18.04 LTS，如果是其他操作系统可以安装虚拟机软件。该部分的操作请自行查阅网络相关文档完成。

2. 安装Ubuntu（版本16.04 X64）程序运行时库。

请您按顺序逐条执行命令。

```
sudo apt-get update
sudo apt-get -y install libssl-dev:i386
sudo apt-get -y install libncurses-dev:i386
sudo apt-get -y install libreadline-dev:i386
```

3. 安装Ubuntu（版本16.04 X64）依赖软件包。


请您按顺序逐条执行命令。

```
sudo apt-get update
sudo apt-get -y install git wget make flex bison gperf unzip
sudo apt-get -y install gcc-multilib
sudo apt-get -y install libssl-dev
sudo apt-get -y install libncurses-dev
sudo apt-get -y install libreadline-dev
sudo apt-get -y install python python-pip
```

4. 安装Python依赖包。

请您按顺序逐条执行命令。

```
python -m pip install setuptools
python -m pip install wheel
python -m pip install aos-cube
python -m pip install esptool
python -m pip install pyserial
python -m pip install scon
```

 **说明** 安装完成后，请您使用[aos-cube --version](#)查看aos-cube的版本号，需确保aos-cube的版本号大于等于0.5.11。

如果在安装过程中遇到网络问题，可使用国内镜像文件。

```
### 安装/升级 pip
python -m pip install --trusted-host=mirrors.aliyun.com -i https://mirrors.aliyun.com/pypi/simple/
--upgrade pip
### 基于pip依次安装第三方包和aos-cube
pip install --trusted-host=mirrors.aliyun.com -i https://mirrors.aliyun.com/pypi/simple/  setuptools
pip install --trusted-host=mirrors.aliyun.com -i https://mirrors.aliyun.com/pypi/simple/  wheel
pip install --trusted-host=mirrors.aliyun.com -i https://mirrors.aliyun.com/pypi/simple/  aos-cube
```

二、开发设备固件

1. 下载设备端SDK代码。下载[蓝牙Mesh SDK \(V1.2.0\)](#)。蓝牙Mesh SDK基于C语言开发，目前该SDK仅对部分用户开放。您需要使用阿里云账号（普通账号即可）登录<https://code.aliyun.com/>后才能下载。如果您登录后仍然没法下载，请通过[工单](#)联系我们。
2. 将下载的ZIP包上传至开发环境中，并解压ZIP包。
3. 在SDK根目录，执行编译命令。

本文档以在TG7100开发板上编译light_ctl程序为例，选择以下任一命令执行即可。

```
./build.sh bluetooth.light_ctl tg7100b 1
```

```
aos make bluetooth.light_ctl@tg7100b
```

出现以下提示时，则编译成功。

```

AOS MEMORY MAP
=====|
| MODULE                | ROM  | RAM  |
=====|
| newlib_stub           | 0    | 270  |
| kv                    | 0    | 3615 |
| bt_common             | 0    | 7705 |
| crc                   | 0    | 128  |
| mcu_tg7100b          | 0    | 38556|
| rhino                 | 0    | 19885|
| vfs                   | 0    | 860  |
| libgcc                | 0    | 8552 |
| ref_impl              | 0    | 1972 |
| bt_mesh               | 0    | 54456|
| bt_host               | 0    | 43334|
| cli                   | 0    | 9039 |
| yloop                 | 0    | 1205 |
| board_tg7100b        | 0    | 965  |
| libc_nano             | 0    | 13100|
| genie_service        | 0    | 24533|
| auto_component        | 0    | 2    |
| driver_bt            | 0    | 18408|
| osal                  | 0    | 950  |
| log                   | 0    | 428  |
| armv6m                | 0    | 336  |
| light_ctl             | 0    | 2841 |
| *fill*                | 0    | 938  |
=====|
| TOTAL (bytes)         | 0    | 252078|
=====|
[INFO] Create bin files
sign all partitions with key:def_otp
-----
    bomtb, 0, 0, 0x11002100, 0x00001000, 0x11003100, bomtb
    FCDS, 0, 0, 0x11004000, 0x00001000, 0x11005000
    imtb, 0, 0, 0x11005000, 0x00002000, 0x11007000, imtb
    kv, 0, 0, 0x11007000, 0x00002000, 0x11009000

```

```

boot, 0, 0, 0x11009000, 0x00005000, 0x1100e000, boot
jumptb, 1, 2, 0x1100e000, 0x00001000, 0x1100f000, jumptb
prim, 1, 2, 0x1100f000, 0x00012000, 0x11021000, prim
misc, 0, 0, 0x11021000, 0x00039000, 0x1105a000
xprim, 1, 2, 0x1105a000, 0x00026000, 0x11080000, xprim

bomtb, 16 bytes
boot, 17696 bytes
jumptb, 1024 bytes
prim, 70996 bytes
xprim, 140172 bytes
imtb, 8192 bytes

-----

[out/bluetooth.light_ctl@tg7100b/binary/fota.bin]FULL==>:jumptb,size:1024,[isforce:0]
[out/bluetooth.light_ctl@tg7100b/binary/fota.bin]FULL==>:prim,size:70996,[isforce:0]
[out/bluetooth.light_ctl@tg7100b/binary/fota.bin]FULL==>:xprim,size:140172,[isforce:0]
Build complete
Making .gdbinit
build time is 9min 48s

```

4. 查看编译结果。编译成功后，在 `out/bluetooth.light_ctl@tg7100b/binary` 目录下会生成 `total_image.hex` 文件。该文件为需要烧录到设备中的固件。`fota.bin` 为可用于OTA升级的固件。

三、烧录固件

1. 使用USB线将TG7100开发板与电脑连接，并打开烧写工具TG71XX Programmer.exe。如果您的电脑没有安装硬件驱动，则还需要按照提示完成驱动的安装。
2. 选中右上角的UART Setting复选框，并如下图所示设置开发板串口的各参数。

驱动安装成功后，TG71XX Programmer.exe工具可自动识别您的串口号，示例中为COM3。

3. 单击Connect。此时，Log区域出现以下日志。

```

*****
Current port: COM3
Current baudrate: 115200
Current stopBits: 1
Current parity: No
Serial opened!!
*****

```

4. 选择Flash_writer > HEX，双击空白处（图中③所示），上传待烧录的文件。

5. 在左下方Single页签中，从TYPE列表中选择MAC，并配置MAC地址。

MAC地址需配置为设备的DeviceName。详细请参见[添加设备](#)。



6. 确认TG7100开发板的拨码开关已置于VDD（图①所示），并按reset键（图②所示）复位硬件。



此时，Log区域打印出以下日志内容。

```
UART RX : cmd>>:
```

7. 单击Flash_writer页签顶部的Erase，擦除固件Flash。

Log区域打印出以下日志内容，则擦除成功。

```
Send erase successfully!  
Receive #OK!  
Erase successfully!
```


8. 单击Flash_writer页签顶部的Write，烧写固件。

Log区域打印出以下日志内容，则固件烧写成功。

```
=====Write hexf File [10/10]=====  
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: 0x0041c6a4  
#OK>>:  
Receive #OK!  
Write images successfully!  
write address: 0x4000, value: 0xB276A87A  
write address: 0x4004, value: 0x0000FA28  
Write registers successfully!
```

四、烧录设备证书

1. 断开烧写工具TG71XX Programmer.exe，并将开发板的拨码开关置于GND，并按下开发板上的reset键。
2. 打开串口调试工具SecureCRT，选择文件 > 快速连接，并如下图所示设置开发板串口的各参数。

 **说明** 其中端口号与TG71XX Programmer.exe工具自动识别的串口号一致。您也可以右击我的电脑，选择管理 > 系统工具 > 设备管理器 > 端口（COM和LPT）来查看（不同系统下打开设备管理器的操作路径略有差异，请根据您的电脑的实际路径操作）。



3. 单击连接。
4. （可选）输入AT+MESHMAC?命令，查看设备的MAC地址。

```
# +MESHMAC: 28:xx:xx:xx:xx:b2
OK
```

5. 输入AT+FACTYTEST命令，进入产测模式。命令执行后返回以下结果。

```
# OK
DUT Start
```

6. 输入以下命令，烧录设备证书。

```
AT+TRITUPLE=<product id>,<Device Secret>,<Device Name>
```

说明

- 该命令中product id、Device Secret、Device Name（即MAC），三者为添加设备中生成的设备证书。
- 在串口烧录工具中输入该命令后，还需要回车（CR）和换行（LF）。

命令执行后，返回以下结果。

```
55xxx10 aaded522xxxxxxxxc88105d10e81d 28xxxxxxxxb2
OK
```

后续步骤

调试设备连云

2.7. 调试设备连云

天猫精灵App连接真实设备后，通过App和控制台（云端）调试真实设备，并在控制台上管理您的设备，验证设备端、云端、App端，三端上下行数据通信。

前提条件

- 已有一台天猫精灵音箱，并使用天猫精灵App连网成功。
- 使用淘宝账号登录天猫精灵App，并完成设备绑定。

一、配网设备

- 按下开发板上的reset键，重启设备退出厂测模式。
- （可选）输入AT+MESHRST命令，清除配网记录，重新进入待配网状态。设备第一次配网也可以忽略该步骤，默认进入待配网状态。

```
# OK
+MESHEVT:0x04
```

3. 通过天猫精灵音箱或天猫精灵App，发现并配网设备。

设备连网成功后，在产品的设备调试页面，测试设备的状态显示为在线。

您还可以通过输入AT+MESHINF?命令，查看当前设备配网状态。此时，返回以下日志内容。

```
# +MESHINF:1 //0表示未配网，1表示已配网
OK
```

二、调试设备

不仅可以通过App控制真实设备（您可以通过手机体验天猫精灵App端控制设备，此处不作介绍），还可以通过控制台调试真实设备，具体操作步骤如下。

1. 在产品的设备调试页面，单击操作中的调试。
2. 单击登录淘宝账号。

3. 在选择调试设备的下拉列表中，选择天猫精灵App中已绑定的设备。设备选择后，页面显示该设备的Product Id、Device Name、Device ID。
4. （可选）选择调试的音箱。当您的天猫精灵App中绑定多个音箱时，您还需要选择具体的调试音箱。

5. 在线调测蓝牙设备。

调试蓝牙设备支持物模型和指令两种方式。

- 物模型

在物模型页签中，选择设备的调试功能，并单击发送指令。

示例：下图为通过物模型方式给设备下发关灯指令。在右侧日志区域可以查看设备相应的运行日志。

- 指令

在指令页签中，输入要测试指令的Opcode和参数，并单击发送指令。

示例：下图为通过指令方式给设备下发切换灯场景模式到夜灯模式。在右侧日志区域可以查看设备相应的运行日志。

6. 在线语音调试设备。

- i. 选择语音调试页签。

- ii. 在对话流区域内，输入对话的语料。平台支持标准语料和泛话语料两种。

- 标准语料

在标准语料页签中，单击平台提供的标准语料。标准语料的详细介绍请参见[查看控制口令](#)。

- 泛化语料

在泛化语料页签中，输入泛化的测试语料，并单击发送测试。

7. 管理您的设备。

您还可以在控制台项目主页面的运营中心里，管理您的真实设备和设备用户。详细介绍请参见[运营中心概述](#)。

