

# 基于设备端SDK的数据上云使用说明

- 设备端SDK简介
- 云端配置
- 设备端配置
- 工具链接

## 1. 设备端SDK简介

在阿里云的IaaS和PaaS层云产品的基础上，搭建了一套完整的、更贴近IoT生活领域的公有云平台。开发者可以通过设备端SDK&API、移动端SDK&API、云端API，实现移动应用和云端对设备进行联网、认证和控制。

设备端SDK是集成在设备端AliOS Things内，目的是将设备连接到云端平台，上报和接收业务报文。设备端SDK是基于MQTT协议开发的。

一个IoT设备通过设备端SDK连云的完整流程是先进行云端产品开发、云端测试设备添加；再进行设备端开发，以及功能调试。

**本文重点讲述设备端的数据通过设备端SDK发到云端，其他的内容只是简单介绍**

## 2 云端配置

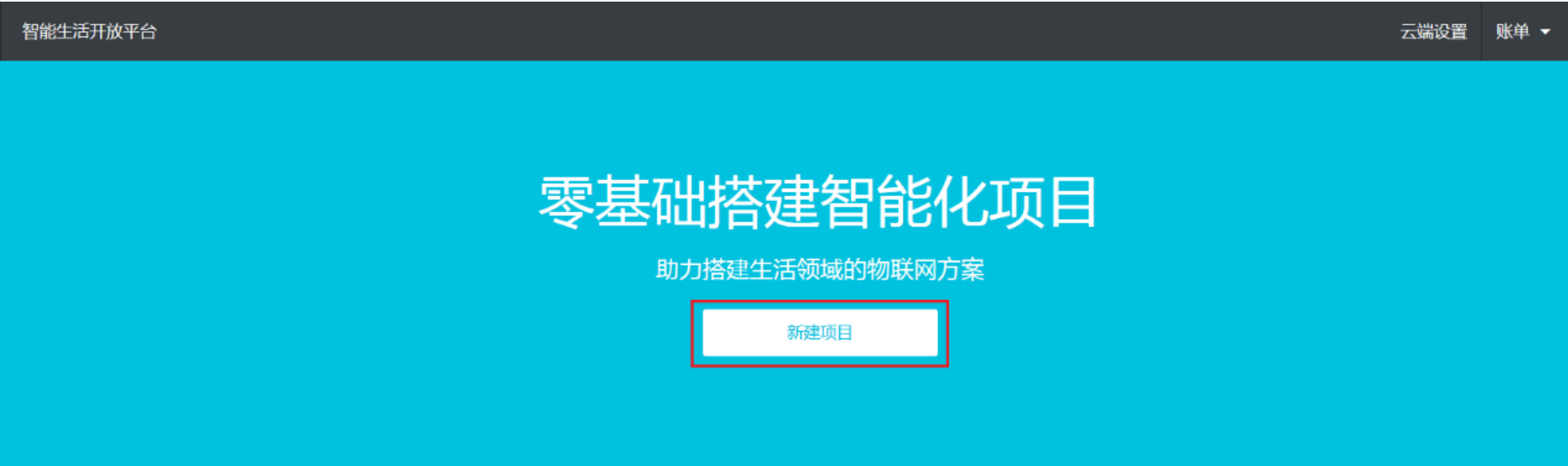
在产品开发前，首先保证云端的产品、设备已成功配置。如果已经在云端已经完成了该工作（例如：与aos\example\linkkitapp\目录下原始用例对应的云端配置），则可以跳过该章节；如果用户想创建一个新产品、新设备，可以参考该章节完成。

云端平台将每一类具有相同功能定义的物联网设备以产品的方式进行管理，以ProductKey作为区分，一款产品包含特定的名称、型号、通讯方式及扩展类描述等信息。产品开发前，请确定已经完成以下工作

- 拥有阿里云平台的使用账号，如果没有，您需要进行账号注册；
  - 正常登录开发平台；

平台连接：<https://living.aliyun.com>

### 2.1 创建项目



### 2.2 创建产品



添加产品，让设备连接云端

研发智能家电，从开发到量产4步走

新建产品



自定义APP应用，让用户控制设备

关联相关设备，研发APP应用2步走

自主研发APP

新建产品

基于阿里IoT认证模组、标准属性和SDK，快速完成模组智能化对接，同时提供专业的人机交互界面。适合简单的家电家装产品。

所属分类

家居安防	>	风扇	加湿器	取暖器	空气盒子	除湿器	新风机	地暖
环境电器	>	净水器	空气净化器					
电工照明	>							
大家电	>							
厨房电器	>							
个护健康	>							
其他	>							

确认

取消

2.产品名称、数据格式设置

新建空气盒子

\* 产品名称

智能空气盒子

\* 节点类型

设备

网关

\* 通讯方式

WIFI

蜂窝(2/3/4G)

蓝牙BLE

ZigBee

以太网

\* 数据格式

ICA标准数据格式 ( 推荐 )

确认

取消

3.产品功能设置

功能定义

功能概述

由于智能生活开放平台已加入ICA联盟，设备数据需符合ICA标准，新功能需审核。

标准功能

2

自定义功能

0

暂无自定义功能，无需提交审核。

标准功能

根据产品的设备类型，我们已为您自动创建了标准功能，您还可以添加可选功能。

查看JSON

新增

功能类型	功能名称	标识符	数据类型	数据定义	操作
事件	故障上报	Error	-	事件类型：信息	编辑
属性	电源开关	PowerSwitch	布尔型	布尔值： 0 - 关闭 1 - 打开	编辑

自定义功能

如果标准功能无法满足您的需求，您还可以添加自定义功能。

新增

功能类型	功能名称	标识符	数据类型	数据定义	操作
------	------	-----	------	------	----

智能空气盒子

更新时间：2018-04-15 11:18:31

基本信息

编辑

所属分类：空气盒子  
节点类型：设备  
通讯方式：WIFI  
数据格式：ICA标准数据格式 ( 推荐 )  
Product Key：a1qiX043iQl  
Product Secret：\*\*\*\* 查看

功能定义

查看

标准功能：2  
自定义功能：0

创建时间：2018-04-15

由于自定义功能需要两天的审核时间，在演示过程中，我们只从标准功能中添加。

新增可选功能

选择功能：

空气盒子

其他类型

TVOC\_VOC浓度

属性

标识符：TVOC 适用类别：空气盒子

当前湿度

属性

标识符：CurrentHumidity 适用类别：空气盒子

当前温度

属性

标识符：CurrentTemperature 适用类别：空气盒子

电池电量

属性

标识符：BatteryPercentage 适用类别：空气盒子

接收信号强度

属性

>

已选功能：

当前湿度

属性

标识符：CurrentHumidity

当前温度

属性

标识符：CurrentTemperature

确定

取消

点击“编辑”键可以修改产品功能的具体信息

标准功能

根据产品的设备类型，我们已为您自动创建了标准功能，您还可以添加可选功能。

查看JSON

新增

功能类型	功能名称	标识符	数据类型	数据定义	操作	
属性	当前温度	可选	CurrentTemperature	浮点型(双精度)	取值范围：-40 ~ 55 单位：°C / 摄氏度	编辑 删除
事件	故障上报	必选	Error	-	事件类型：信息	编辑
属性	当前湿度	可选	CurrentHumidity	整数型	取值范围：0 ~ 100 单位：% / 百分比	编辑 删除
属性	电源开关	必选	PowerSwitch	布尔型	布尔值： 0 - 关闭 1 - 打开	编辑

自定义功能

如果标准功能无法满足您的需求，您还可以添加自定义功能。

新增

功能类型	功能名称	标识符	数据类型	数据定义	操作
暂无自定义功能					

下一步

2.3 进入产品开发

通过以下两步进入产品开发流程

1

功能定义

2

设备调试

3

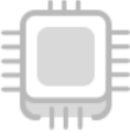
人机交互

4

批量生产

选择模组

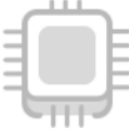
常用



品牌：不限  
型号：不限

使用选中模组

模组信息



品牌：不限 已认证  
型号：不限


重选模组

采购模组

推荐前往阿里IoT云市场进行采购,经阿里测试和大规应用验证的通讯模组。

前往购买


设备调试



功能定义

确定设备的功能属性


>



SDK集成

基于SDK进行设备开发

>



调试设备

50个免费激活码调试开发

开始开发

请前往一站式开发平台 开发设备

智能空气盒子

更新时间：2018-04-15 11:18:31

基本信息

编辑

所属分类：空气盒子  
节点类型：设备  
通讯方式：WIFI  
数据格式：ICA标准数据格式 (推荐)  
Product Key：a1qiX043iQl  
Product Secret：\*\*\*\* 

查看

功能定义

查看

标准功能：4  
自定义功能：0

创建时间：2018-04-15

2.4 创建设备

通过“新增测试设备”按键增加一个测试设备

智能空气盒子 

开发中

所属分类: 空气盒子

产品型号: -

节点类型: 设备

通讯方式: Wi-Fi

数据格式: Alink

创建时间: 2018-04-14 18:20:08

ProductKey: a1Enz5qfa8e 

复制

ProductSecret: \*\*\*\*\* 

显示

产品描述: -

开发文档

编辑

发布产品

功能定义

设备开发

扩展信息

SDK 集成

根据SDK集成文档，进行设备端的开发和调试。

SDK	相关资源
C SDK	<a href="#">下载SDK</a> <a href="#">集成文档</a>
Node.js SDK	<a href="#">下载SDK</a> <a href="#">集成文档</a>

设备列表

产品开发阶段允许添加最多50个测试设备，上线发布后将不再限制设备接入数。

已添加设备 1/50

在线调试

新增测试设备

为该设备添加该产品中唯一的设备名称

新增测试设备

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。

DeviceName :

aos\_air\_box\_01

确认

取消

2.5 获取产品信息

设备创建成功后，云端的配置已经完成，记录以下信息，在后续的设备端开发中会使用。

2.5.1 获取到产品的ProductKey与ProductSecret

产品接入 > 产品详情

智能空气盒子 开发中

开发文档 编辑 发布产品

所属分类: 空气盒子

产品型号: -

节点类型: 设备

通讯方式: Wi-Fi

数据格式: Alink

创建时间: 2018-04-14 18:20:08

ProductKey: a1Enz5qfa8e 复制

ProductSecret: \*\*\*\*\* 显示

产品描述: -

2.5.2 获取设备的激活凭证

设备列表 产品开发阶段允许添加最多50个测试设备，上线发布后将不再限制设备接入数。 已添加设备 1/50 在线调试 新增测试设备

<input type="checkbox"/>	DeviceName	状态	最后上线时间	操作
<input type="checkbox"/>	aos_air_box_01	● 未激活	-	查看 调试 激活凭证

查看设备激活凭证

设备激活凭证，请烧录到设备中

ProductKey :	a1Enz5qfa8e	复制
DeviceName :	aos_air_box_01	复制
DeviceSecret :	zlmMgPN6kJ4KH2YyrH4DM6l5w8V5uVDY	复制

一键复制

确定

2.5.3 获取物的描述信息

所属分类: 空气盒子	产品型号: -	节点类型: 设备
通讯方式: Wi-Fi	数据格式: Alink	创建时间: 2018-04-14 18:20:08
ProductKey: a1Enz5qfa8e 复制	ProductSecret: ***** 显示	
产品描述: -		

- 功能定义
- 设备开发
- 扩展信息

**功能概览** 由于智能生活开放平台已加入ICA联盟，设备数据需符合ICA标准，新功能需审核。

<div><div>标准功能</div><div>4</div></div>	<div><div>自定义功能</div><div>0</div></div>
<div><div></div><div>暂无自定义功能，无需提交审核。</div></div>	

**标准功能** 根据产品的设备类型，我们已为您自动创建了标准功能，您还可以添加可选功能。

查看JSON

新增

查看JSON

根据产品功能定义，自动生成对应的Alink JSON格式协议。

```
1 {
2   "schema": "https://iot-tsl.oss-cn-shanghai.aliyuncs.com/schema.json"
3   "profile": {
4     "productKey": "a1Enz5qfa8e"
5   },
6   "services": [
7     {
8       "outputData": [],
9       "identifier": "set",
10      "inputData": [
11        {
12          "identifier": "PowerSwitch",
13          "dataType": {
14            "specs": {
15              "0": "关闭",
16              "1": "打开"
17            },
18            "type": "bool"
19          },
20          "name": "电源开关"
21        }
22      ]
23    }
24  ]
25 }
```

关闭

2.6 名词解释

名词	说明
产品	一组具备共同功能定义的设备集合，以ProductKey作为区分，一款产品包含特定的名称、型号、通讯方式及扩展类描述等信息
设备	具备唯一标识的具体设备，以设备凭证作为区分，一般包括能够与云端独立进行交互的直连设备，也包括通过网关接入云端的子设备
ProductKey	由平台颁发的产品唯一标识，11位长度的英文数字随机组合
ProductSecret	由平台颁发的产品加密密钥，通常与ProductKey成对出现，可用于一型一密的认证方案
DeviceName	用户注册设备时生成的设备唯一编号，支持系统自动生成，也可支持用户添加自定义编号，产品维度内唯一



名词	说明
DeviceSecret	设备密钥，与DeviceName成对出现，可用于一机一密的认证方案
设备激活凭证	设备在云端预注册后颁发的设备认证证书，包含ProductKey、DeviceName和DeviceSecret三元组，开发者可下载到本地（ excel文件 ），用于产线批量烧录，设备上线时根据该凭证进行认证鉴权
属性	即property，设备的功能模型之一，一般用于描述产品固有的特性，例如检测的温度、湿度等
事件	即events，设备端post给云端的消息，新建一个property时，其event应该有property的identifier，这个在建立property时，event自动生成，不过其method为thing.event.property.post
服务	即service，产品提供给云端的能力,云端可以通过sevice，向设备发起一系列命令

### 3 终端设备开发

#### 3.1 修改设备凭证

设备凭证（即设备四元组，参考2.5.1和2.5.2章节）是从云端获取的。临时验证的话可修改到头文件aos\framework\protocol\alink-ilop\sdk-encap\imports\iot\_import\_product.h中（当前多个凭证信息是由编译宏控制，编译宏定义位于文件aos\example\linkkitapp\linkkitapp.mk）；如果是产品化，则需要自行开发工具，在出厂前烧录到固件中。  
如果只需要拿用例中已创建好的设备做验证，则可以跳过该步骤。

```
#define PRODUCT_KEY          "a1Enz5qfxxx"
#define PRODUCT_SECRET      "ouov8ya7p3Ichxxx"
#define DEVICE_NAME         "aos_air_boxxx"
#define DEVICE_SECRET       "zImMgPN6kJ4KH2YyrH4DM6l12345uxxx"
```

#### 3.2 修改TSL信息

在产品功能定义完成后，平台将自动生成JSON格式的TSL（即 Thing Specification Language），用来描述物的模型，获取方式请参考2.5.3章节。该信息经过“压缩并转义”后，转换成C语言下的json格式，这样可以添加到设备端的TSL信息中。“压缩并转义”的方式可以参考下面的代码，或者链接：  
<https://www.sojson.com/yasuo.html>

```
void c_json(char* str,char* outfile)
{
    int i = 0;
    FILE *f;
    int len = strlen(str);
    f=fopen(outfile,"w");

    for(i = 0; i < len; i++,str++)
    {
        if((*str) == 0x20 || (*str) == 0x0a || (*str) == 0x0d)
        {
            continue;
        }
        else
        {
            if(*str == 0x22)
            {
                fprintf(f,"%c",0x5c);
            }
            fprintf(f,"%c",*str);
        }
    }
    fprintf(f,"\n\n");
    fclose(f);
}

/* file_in is the original json tsl, file_out is c_json */
void dofile(char *file_in,char *file_out)
{
    FILE *f;long len;char *data;

    f=fopen(file_in,"rb");
    fseek(f,0,SEEK_END);
    len=ftell(f);
    fseek(f,0,SEEK_SET);

    data=(char*)malloc(len+1);
    fread(data,1,len,f);
```



```
fclose(f);

c_json(data,file_out);

free(data);
}
```

在创建物的对象时，需要转义后的TSL信息，其修改可以参考文件在linkkit\_app.c。

```
const char TSL_STRING[] = "{\"schema\":\"https://iot-tsl.oss-cn-shanghai.aliyuncs.com/schema.json\",\"profile\":{\"productKey\":\"a1Enz5qfxx\"}"};
```

3.3 设备端SDK函数接口

设备端SDK服务例程初始化、处理流程需要在配网完成之后，建议注册在事件CODE\_WIFI\_ON\_GOT\_IP里。设备端SDK示例分为以下步骤顺序完成：服务程序初始化，进入事件分发处理，创建对象，设置物对象的TSL属性、上报云端。后面会介绍该过程，具体请参考（aos\example\linkkitapp下的用例代码，入口函数为linkkit\_main）

3.3.1 设备端SDK服务程序初始化

名称	描述
函数	int linkkit_start(int max_buffered_msg, int get_tsl_from_cloud, linkkit_loglevel_t log_level, linkkit_ops_t ops, linkkit_cloud_domain_type_t domain_type, void user_context)
说明	启动设备端SDK服务，与云端建立连接并安装回调函数
返回值	0: 服务程序启动成功; -1: 服务程序启动失败

参数	类型	描述	备注
max_buffered_msg	int	消息队列大小	此参数影响事件消息最大暂存数，频繁业务可适当增大此参数，建议值 [16~32]
get_tsl_from_cloud	int	获取 TSL 路径	0: 从本地获取物模型，非0: 从云端获取物模型
log_level	linkkit_loglevel_t	设置打印的日志等级	[0~5]: 数字越大, 打印越详细
ops	linkkit_ops_t*	设备端SDK事件回调函数	设备端SDK通过回调函数将事件通知给用户
domain_type	linkkit_cloud_domain_type_t	云服务器域名	linkkit_cloud_domain_sh: 上海域名linkkit_cloud_domain_usa: 美国域名
user_context	void*	用户上下文环境指针	用户自定义结构体的指针, 回调函数中传出

事件类型以及对应的事件回调函数如下所示,每当设备端SDK处理完相应的事件后，都会执行用户注册的回调函数，具体实现可参考linkkit\_app.c内的实现：

```
typedef enum {
    dm_callback_type_property_value_set = 0,
    dm_callback_type_service_requested,
#ifdef RRPC_ENABLED
    dm_callback_type_rrpc_requested,
#endif /* RRPC_ENABLED */
    dm_callback_type_cloud_connected,
    dm_callback_type_cloud_disconnected,
    dm_callback_type_new_thing_created,
    dm_callback_type_thing_disabled,
    dm_callback_type_thing_enabled,
    dm_callback_type_raw_data_arrived,

    dm_callback_type_number,
} dm_callback_type_t;

static linkkit_ops_t alinkops = {
    .on_connect      = on_connect,
    .on_disconnect   = on_disconnect,
    .raw_data_arrived = raw_data_arrived,
    .thing_create    = thing_create,
    .thing_enable    = thing_enable,
    .thing_disable   = thing_disable,
    .thing_call_service = thing_call_service,
```

```
        .thing_prop_changed = thing_prop_changed,
    };

    linkkit_start(16, 0, 5, &alinkops, linkkit_cloud_domain_sh, sample_ctx);
```

3.2.2 创建对象

根据设备描述信息（tsl），创建物的对象，并添加到设备端SDK。如果创建成功，会在注册的thing\_create函数里返回物的唯一标志符，后续只需要通过该标识符来管理对象。设备描述信息（tsl）的获取，可以参考2.1.6章节

名称	描述
函数	void <i>linkkit_set_tsl</i> (const char tsl, int tsl_len)
说明	从本地读取 TSL 文件,生成物的对象并添加到设备端SDK中
返回值	0: 成功; -1: 物的唯一标识符

参数	类型	描述	备注
tsl	const char*	设备描述信息( JSON 类型)	tsl 必须和设备三元组所属的产品的 Alink JSON 格式协议相同
tsl_len	int	设备描述信息长度	-

代码示例

```
int linkkit_main()
{
    sample_context_t* sample_ctx = &g_sample_context;
    int execution_time = 0;
    int get_tsl_from_cloud = 0;

    execution_time = execution_time < 1 ? 1 : execution_time;

    memset(sample_ctx, 0, sizeof(sample_context_t));
    sample_ctx->thing_enabled = 1;
    linkkit_start(16, get_tsl_from_cloud, linkkit_loglevel_debug, &alinkops, linkkit_cloud_domain_sh, sample_ctx);
    if (!get_tsl_from_cloud) {
        linkkit_set_tsl(TSL_STRING, strlen(TSL_STRING));
    }

    aos_post_delayed_action(1000, linkkit_action, sample_ctx);
    return 0;
}
```

3.2.3 事件分发处理

设备端SDK服务程序初始化之后，需要周期性地地进行进行事件分发处理，建议放到任务或delay事件中执行。该函数会根据事件回调的类型来执行相应的回调函数，linkkit\_app.c的用例代码中已默认调用了该接口。

名称	描述
函数	int linkkit_dispatch()
说明	事件分发函数,触发linkkit_start安装的用户回调
返回值	0: 成功; -1: 失败

代码示例

```
void linkkit_action(void *params)
{
    int ret;
    sample_context_t* sample_ctx = params;

    linkkit_dispatch();

    aos_post_delayed_action(200, linkkit_action, sample_ctx);
}
```

3.2.4 设置对象的TSL属性

设置TSL属性，就是给已创建的对对象的TSL属性赋值，赋值完成后，可以通过接口linkkit\_post\_property将对象属性上报到云端，也可以通过linkkit\_trigger\_event将事件上报云端。

名称	描述
函数	int linkkit_set_value(linkkit_method_set_t method_set, const void thing_id, const char identifier, const void value, const char value_str)
说明	根据identifier设置物对象的 TSL 属性，如果标识符为struct类型、event output类型或者service output类型，使用点'.'分隔字段；例如"identifier1.identifier2"指向特定的项
返回值	0: 物对象 TSL 属性设置成功; -1: 物对象 TSL 属性设置失败

参数	类型	描述	备注
method_set	linkkit_method_set_t	设置 TSL 属性的类型	linkkit_method_set_property_value:设置模型property value linkkit_method_set_event_output_value:设置物模型event output value linkkit_method_set_service_output_value:设置物模型service output value
thing_id	const void*	物的唯一标识符	物对象的标识符,可从 thing_create 回调获得设备端SDK创建的 thing id
identifier	const char*	TSL 属性标示符	物对象 TSL 属性的标识符
value	const void*	输入的属性值	和 value_str, 二选一必填
value_str	const char*	输入的字符串格式的 属性值	和 value, 二选一必填

3.2.5 上报设备属性到云端

名称	描述
函数	int linkkit_post_property(const void thing_id, const char property_identifier)
说明	上报设备属性到云端
返回值	0: 上报成功; -1: 上报失败linkkit_post_property败

参数	类型	描述	备注
thing_id	const void*	物的唯一标识符	物对象的标识符,可从 thing_create 回调中获得该值
property_identifier	const char*	属性标示符	上报的物对象的TSL属性标识符

物对象的TSL标识符请参考物的TSL字符串或下图中的“标识符”列

标准功能

根据产品的设备类型，我们已为您自动创建了标准功能，您还可以添加可选功能。

查看JSON

新增

功能类型	功能名称	标识符	数据类型	数据定义	操作
属性	当前温度 <span>可选</span>	CurrentTemperature	浮点型(双精度)	取值范围：-40 ~ 55 单位：°C / 摄氏度	<a href="#">编辑</a> <a href="#">删除</a>
<span>+</span> 事件	故障上报 <span>必选</span>	Error	-	事件类型：信息	<a href="#">编辑</a>
属性	当前湿度 <span>可选</span>	CurrentHumidity	整数型	取值范围：0 ~ 100 单位：% / 百分比	<a href="#">编辑</a> <a href="#">删除</a>
属性	电源开关 <span>必选</span>	PowerSwitch	布尔型	布尔值： 0 - 关闭    1 - 打开	<a href="#">编辑</a>

```
int linkkit_data_publish(const char* identifier, const void* value, const char* value_str)
{
    int ret = 0;
    ret = linkkit_set_value(linkkit_method_set_property_value.g thing_id.identifier.value.value_str);
}
```

```

    if(0 != ret) {
        return ret;
    }

    ret = linkkit_post_property(g_thing_id,identifier);
    return ret;
}

int data_publish_test()
{
    int ret = 0;

    /*温度值为24.6*/
    ret = linkkit_data_publish("CurrentTemperature",NULL,"24.6");
    if(0 != ret) {
        return ret;
    }
    /*湿度值为43%*/
    ret = linkkit_data_publish("CurrentHumidity",NULL,"43");
    if(0 != ret) {
        return ret;
    }
    return 0;
}

```

3.2.5 上报事件到云端

linkkit\_set\_value的第一个参数值选择linkkit\_method\_set\_event\_output\_value，则可以设置完事件（event）的TSL信息，之后可以通过linkkit\_trigger\_event将事件发送到云端。

名称	描述
函数	int linkkit_trigger_event(const void <i>thing_id</i> , const char event_identifier)
说明	上报设备事件到云端
返回值	0: 上报成功; -1: 上报失败

参数	类型	描述	备注
thing_id	const void*	物的唯一标识符	物对象的标识符,可从 thing_create 回调中获得该值
event_identifier	const char*	事件标示符	上报的事件标识符

3.2.6 回应云端请求

名称	描述
函数	int linkkit_answer_service(const void <i>thing_id</i> , const char service_identifier, int response_id, int code)
说明	对云端服务请求进行回应
返回值	0: 上报成功; -1: 上报失败linkkit_post_property败

参数	类型	描述	备注
thing_id	const void*	物的唯一标识符	物对象的标识符,可从 thing_create 回调中获得该值
service_identifer	const char*	服务标识符	用户应该从设备端SDK的回调函数获得此标识符，使用此函数生成回复码发送给服务端
response_id	int	回复ID	response_id 应该从设备端SDK回调函数中获得此参数
code	int	回复状态码	200: 成功 400: 失败

物对象的TSL标识符请参考物的TSL字符串或下图中的“标识符”列  
设备端详细的API使用请参考文档：<https://living.aliyun.com/doc#cusllk.html>

4 功能调试

下面以MK3060为例说明设备端SDK用例的调试过程。

4.1 配网

设备配网分为一键式配网、热点配网、路由器配网和零配网四种方式，不作为该文档的重点，这里只按照一键式配网的流程进行介绍。  
下面以MK3060为例说明一键式配网调试的过程，该流程是针对未配网的单板来说明的，已成功配网的单板可跳过该步骤。并且该流程是根据Demo APP来介绍的，如果用户需要开发自己的APP请参考文档 <https://living.aliyun.com/doc#yzqy0q.html>

#### 4.1.1 手机端下载并安装配网使用的APP

APP会经常更新，这里只介绍对在MK3060上配网成功过的Demo APP；

APP下载的内网链接：[http://mtl3.alibaba-inc.com/project/project\\_build\\_config.htm?projectId=61218&buildConfigId=444184](http://mtl3.alibaba-inc.com/project/project_build_config.htm?projectId=61218&buildConfigId=444184)

只能访问外网的同学可以找相关接口人拿到Demo APP。

#### 4.1.2 单板烧写程序

单板需要将**原始用例的可执行文件**烧录得到mk3060单板中（即master最新的未修改过的代码，通过编译命令[aos make linkkitapp@mk3060](#)得到的可执行文件linkkitapp@mk3060.ota.bin）。

烧写的命令是长按boot键的同时短按reboot键，使MK3060进入bootloader，通过write 0x13200,可以烧写版本。

#### 4.1.3 单板进入配网模式

1.短按reboot键，单板复位，如果设备没有连网，则待设备进入moniter模式后，出现ssid扫描列表；

```
chan 1
[1] ssid:alibaba-inc, mac:141fba8c7f90, chn:1, auth:wpa2-8021x, aes, aes, rssi:0, adha:0
[2] ssid:alibaba-inc, mac:141fba8c8220, chn:1, auth:wpa2-8021x, aes, aes, rssi:0, adha:0
[3] ssid:Pi3-AP, mac:b827eb271f36, chn:1, auth:wpa2-psk, aes, aes, rssi:0, adha:0
[4] ssid:alibaba-guest, mac:141fba8c8ea1, chn:1, auth:open, none, none, rssi:0, adha:0
[5] ssid:alibaba-guest, mac:141fba8c7f91, chn:1, auth:open, none, none, rssi:0, adha:0
[6] ssid:alibaba-inc, mac:141fba8c8e60, chn:1, auth:wpa2-8021x, aes, aes, rssi:0, adha:0
[7] ssid:alibaba-inc, mac:141fba8c8ea0, chn:1, auth:wpa2-8021x, aes, aes, rssi:0, adha:0
[8] ssid:SL-IR4_F149, mac:bc14efeaf148, chn:3, auth:wpa2-psk, aes, tkip, rssi:0, adha:0
[9] ssid:alibaba-guest, mac:141fba8c8e61, chn:1, auth:open, none, none, rssi:0, adha:0
[10] ssid:alibaba-guest, mac:141fba8c90e1, chn:1, auth:open, none, none, rssi:0, adha:0
[11] ssid:alibaba-guest, mac:141fba8c8c61, chn:1, auth:open, none, none, rssi:0, adha:0
chan 6
[12] ssid:alibaba-guest, mac:141fba8c8041, chn:6, auth:open, none, none, rssi:0, adha:0
[13] ssid:alibaba-inc, mac:141fba8c7d60, chn:6, auth:wpa2-8021x, aes, aes, rssi:0, adha:0
[14] ssid:alibaba-inc, mac:141fba8c6f30, chn:6, auth:wpa2-8021x, aes, aes, rssi:0, adha:0
[15] ssid:alibaba-guest, mac:141fba8c6f31, chn:6, auth:open, none, none, rssi:0, adha:0
[16] ssid:alibaba-guest, mac:141fba8c7d61, chn:6, auth:open, none, none, rssi:0, adha:0
[17] ssid:Yuemewifi-3766, mac:948b03033766, chn:4, auth:wpa2-psk, aes, aes, rssi:0, adha:0
[18] ssid:alibaba-inc, mac:141fba8c8040, chn:6, auth:wpa2-8021x, aes, aes, rssi:0, adha:0
chan 11
```

2.再短按boot键，激活配网模式（短按boot键会调用到函数do\_awss\_active）；

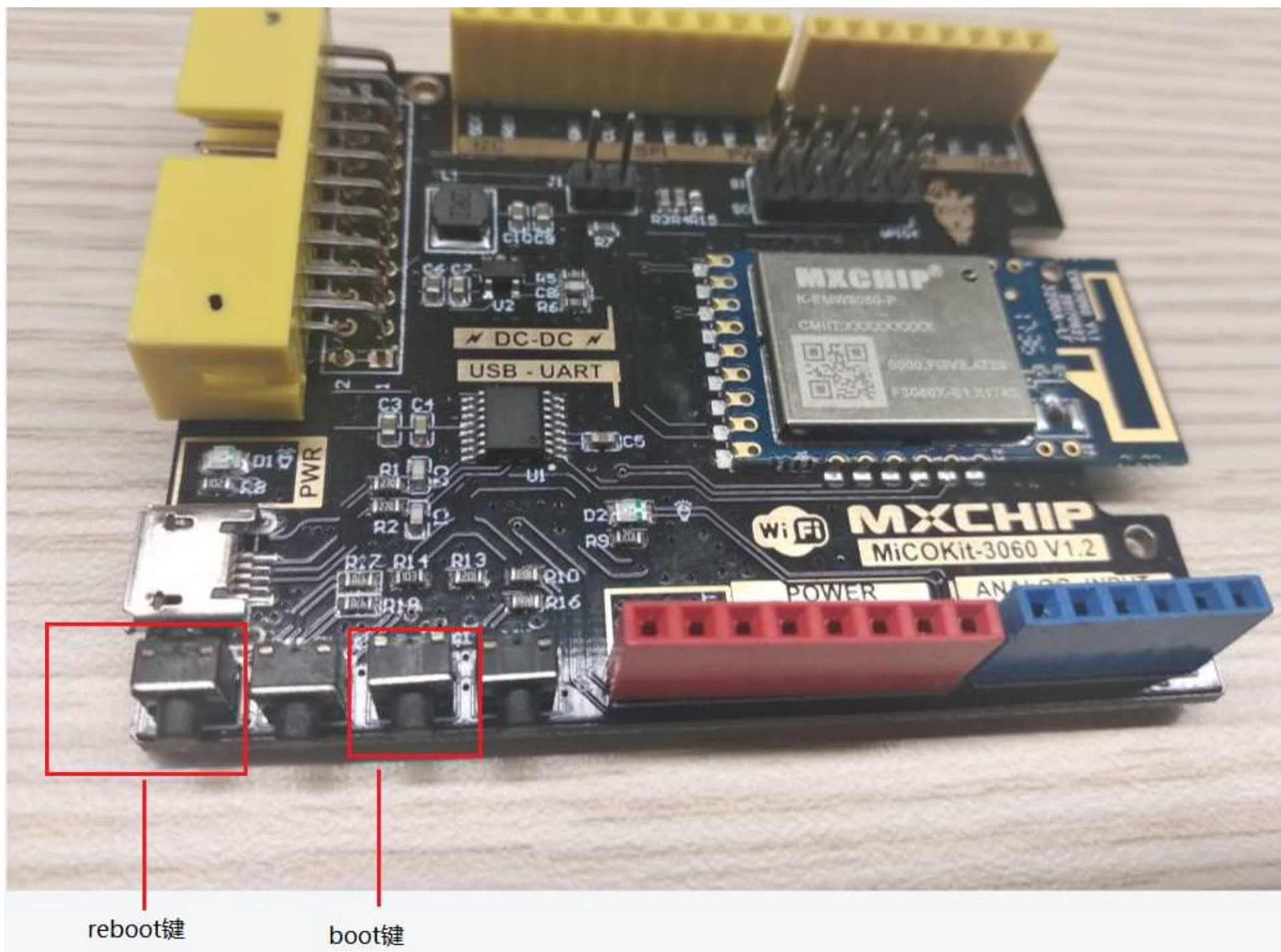
```
chan 3
chan 4
chan 6
chan 11
[63] ssid:alibaba-inc, mac:141fba8c8730, chn:11, auth:wpa2-8021x, aes, aes, rssi:0, adha:0
[008728]<v> awss config press 2
[008728]<v> do_awss_active 0
press
chan 13
chan 1
chan 6
len=360, F, sn=3040, enc=3, chn=6, src=2c3ae822081e
chan 11
```

备注：

如果设备已配好网，reboot后不会进入moniter模式,会直接连云；

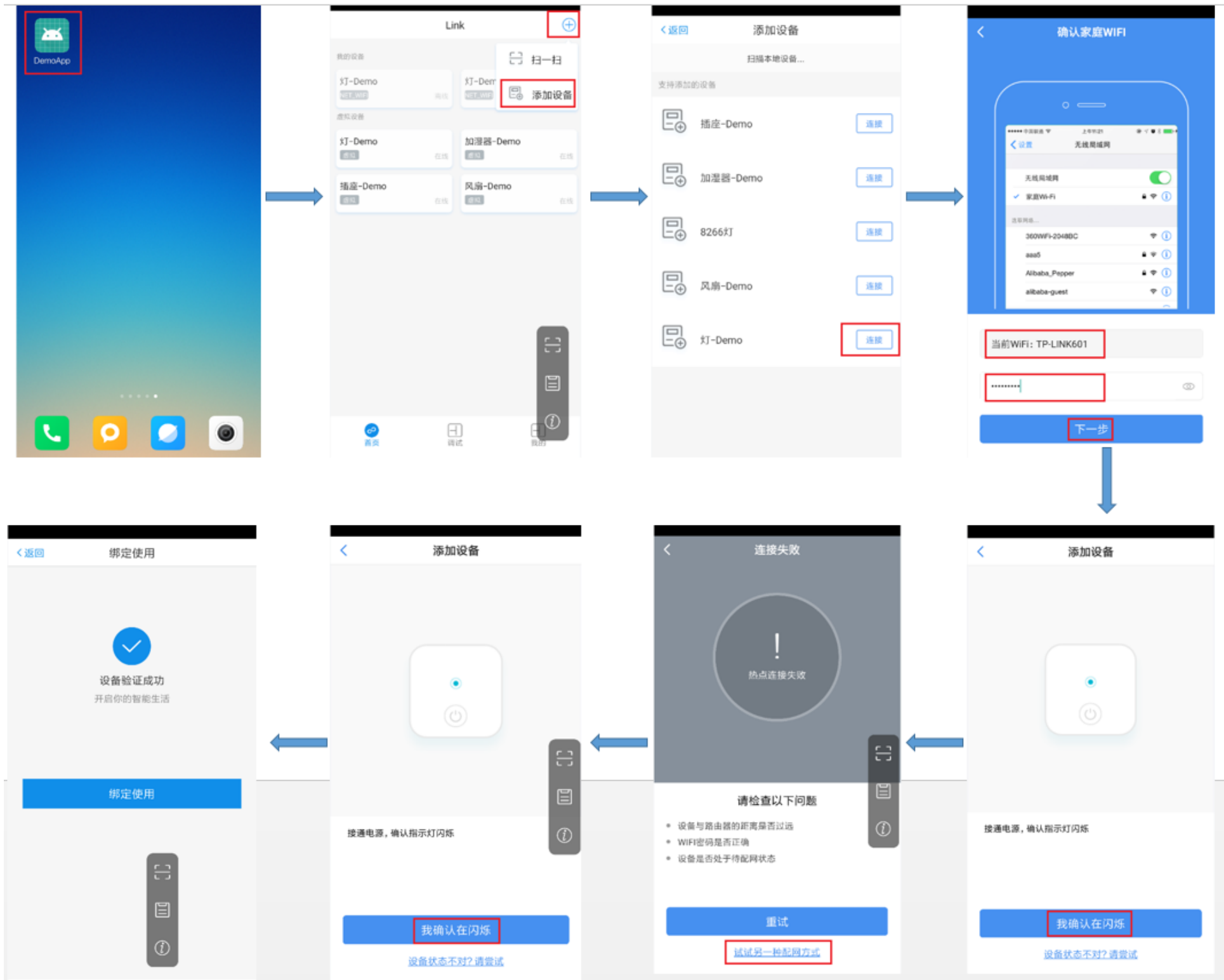
长按boot键5S，会恢复出厂设置（即调用到函数do\_awss\_reset），则可以重复1、2步骤，重新配网；





#### 4.1.4 Demo APP配网

待单板进入配网模式后，手机侧进行以下操作，完成一键式配网：



## 4.2 用例编译

编译前确保aos\example\linkkitapp目录下为修改过的用例（参考第3章介绍）

编译命令为aos make linkkitapp@mk3060

linkkitapp 为用例名称；mk3060为单板类型

编译后生成的文件为aos\out\linkkitapp@mk3060\binary\linkkitapp@mk3060.ota.bin

## 4.3 用例执行

用原始的linkkitapp可执行文件配网成功后，然后替换新的用例执行，在云端可以看到设备已经是在线状态

设备列表 产品开发阶段允许添加最多50个测试设备，上线发布后将不再限制设备接入数。				已添加设备 1/50	在线调试	新增测试设备
<input type="checkbox"/>	DeviceName	状态	最后上线时间	操作		
<input type="checkbox"/>	aos_air_box_01	● 在线	2018-04-15 12:20:27	查看	调试	激活凭证



而当设备端上传数据成功后，可以在云端看到设备上传的数据

智能空气盒子 / aos\_air\_box\_01● 在线

调试设备

产品型号: -

ProductKey: a1Enz5qfa8e [复制](#)

DeviceName: aos\_air\_box\_01 [复制](#)

DeviceSecret: \*\*\*\*\* [显示](#)

添加时间: 2018-04-14 23:55:35

激活时间: 2018-04-15 1:55:39

最后上线时间: 2018-04-15 12:20:27

固件版本: app-1.0.0-20180415.0106

运行状态	服务调用	事件管理	设备日志	扩展信息
------	------	------	------	------

设备运行状态

显示设备有上报数据的最新属性值，点击“查看数据”可以查看指定属性的历史数据。

实时刷新 ☒

当前湿度

43%

2018-04-15 12:20:29

查看数据

当前温度

24.6°C

2018-04-15 12:20:29

查看数据