

# Soundbox 天猫精灵使用说明

## 1. SDK 配置

### (1) 实例选择

在 `api/model_api.h` 下，通过配置 `CONFIG_MESH_MODEL` 选择天猫精灵例子。

```
01 model_api.h
1  #ifndef __MODEL_API_H__
2  #define __MODEL_API_H__
3
4  //< Detail in "MshMDLv1.0.1"
5  #define SIG_MESH_GENERIC_ONOFF_CLIENT 0 // examples/generic_onoff_client.c
6  #define SIG_MESH_GENERIC_ONOFF_SERVER 1 // examples/generic_onoff_server.c
7  #define SIG_MESH_VENDOR_CLIENT 2 // examples/vendor_client.c
8  #define SIG_MESH_VENDOR_SERVER 3 // examples/vendor_server.c
9  #define SIG_MESH_ALIGENIE_SOCKET 4 // examples/AliGenie_socket.c
10 #define SIG_MESH_LIGHT_LIGHTNESS_SERVER 5 // examples/light_lightness_server.c
11 // more...
12
13 //< Config which example will use in <examples>
14 #define CONFIG_MESH_MODEL SIG_MESH_ALIGENIE_SOCKET
15
16
```

### (2) 板级配置

在 `apps/mesh/board/br23/board_ac635n_demo_cfg.h` 下，配置打印引脚与 AD\_KEY 引脚

```
19 //***** UART *****//
20 //***** UART *****//
21 //***** UART *****//
22 //***** UART *****//
23 #define TCFG_UART0_ENABLE ENABLE_THIS_MODULE //串口打印模块使能
24 #define TCFG_UART0_RX_PORT NO_CONFIG_PORT //串口接收脚配置（用于打印可以选择NO_CONFIG_PORT）
25 #define TCFG_UART0_TX_PORT IO_PORTA_00 //串口发送脚配置
26 #define TCFG_UART0_BAUDRATE 1000000 //串口波特率配置
27
```

```
//***** adkey *****//
//***** adkey *****//
//***** adkey *****//
//***** adkey *****//
#define TCFG_ADKEY_ENABLE ENABLE_THIS_MODULE //DISABLE_THIS_MODULE //是否使能AD按键
#define TCFG_ADKEY_PORT IO_PORTB_01 //AD按键端口（需要注意选择的IO口是否支持AD功能）
/*AD通道选择，需要和AD按键的端口相对应：
AD_CH_PA1 AD_CH_PA3 AD_CH_PA4 AD_CH_PA5
AD_CH_PA9 AD_CH_PA1 AD_CH_PB1 AD_CH_PB4
AD_CH_PB6 AD_CH_PB7 AD_CH_DP AD_CH_DM
AD_CH_PB2
*/
#define TCFG_ADKEY_AD_CHANNEL AD_CH_PB1
#define TCFG_ADKEY_EXTERN_UP_ENABLE ENABLE_THIS_MODULE //是否使用外部上拉
```

### (3) 实例配置

LED 指示灯引脚配置

```
const u8 led_use_port[] = {
    IO_PORTA_01,
    IO_PORTA_02,
};
```

## 2. 实例介绍

### （1）天猫精灵实例简介

该实例按照阿里巴巴 IoT 开放平台关于天猫精灵蓝牙 mesh 软件基础规范，根据硬件品类规范描述自己为一个 插座，通过天猫精灵语音输入进行发现连接(配网)和控制设备

### （2）实际操作

- 编译工程并下载到目标板，接好串口，接好演示用 LED 灯，上电或者复位设备
- 天猫精灵连接到互联网上
- 上电天猫精灵，长按设备上的语音按键，让设备进入待连接状态
- 手机应用商店下载天猫精灵 APP，APP 上登陆个人中心
- 打开手机 WLAN，将天猫精灵通过手机热点连接到互联网上
- 通过天猫精灵进行配网和控制

### （3）配网对话

用户：“天猫精灵，搜索设备”

天猫精灵：“发现一个智能插座，是否连接”

用户：“连接”

天猫精灵：“连接成功。。。 ”

语音控制 插座命令 (可通过 IoT 开放平台 添加自定义语音命令)

#### 命令

#### 效果

天猫精灵，打开插座      开发板上 **LED** 灯打开

天猫精灵，关闭插座      开发板上 **LED** 灯关闭

### （4）消息处理函数

- SIG Generic OnOff Server 回调

结构体 root\_models 里的 Generic\_OnOff\_Server 注册了回调 gen\_onoff\_srv\_op 来对用户数据进行处理

当收到 BT\_MESH\_MODEL\_OP\_GEN\_ONOFF\_GET 等注册消息时，就会调用 gen\_onoff\_get 等对应的回调函数进行用户数据处理

```
/*
 * @brief OnOff Model Server Op Dispatch Table
 */
/*-----*/
static const struct bt_mesh_model_op gen_onoff_srv_op[] = {
    { BT_MESH_MODEL_OP_GEN_ONOFF_GET, 0, gen_onoff_get },
    { BT_MESH_MODEL_OP_GEN_ONOFF_SET, 2, gen_onoff_set },
    { BT_MESH_MODEL_OP_GEN_ONOFF_SET_UNACK, 2, gen_onoff_set_unack },
    BT_MESH_MODEL_OP_END,
};
```

- Vendor Model 回调

结构体 `vendor_srv_op` 里的 `Vendor_Server_Model` 注册了回调 `vendor_srv_op` 来对用户数据进行处理

当收到 `VENDOR_MSG_ATTR_GET` 等注册消息时，就会调用 `vendor_attr_get` 等对应的回调函数进行用户数据处理

```
/*  
 * @brief Vendor Model Server Op Dispatch Table  
 */  
/*-----*/  
static const struct bt_mesh_model_op vendor_srv_op[] = {  
    { VENDOR_MSG_ATTR_GET,    ACCESS_OP_SIZE, vendor_attr_get },  
    { VENDOR_MSG_ATTR_SET,    ACCESS_OP_SIZE, vendor_attr_set },  
    { VENDOR_MSG_ATTR_CONFIRM, 1, vendor_attr_cfm },  
    BT_MESH_MODEL_OP_END,  
};
```