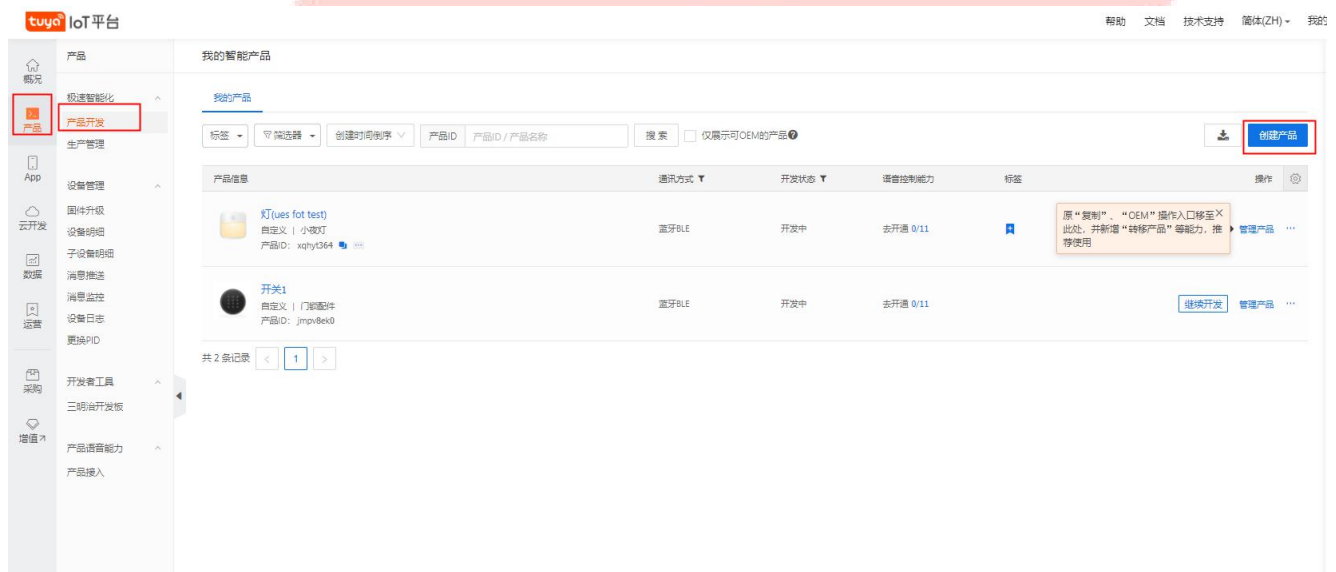


# 涂鸦协议开发说明

## 一、测试用授权码申请

1.打开涂鸦 IOT 平台 <https://auth.tuya.com/>，注册并登陆。打开产品->产品开发界面，点击创建产品。



2.选择一个具有自研模组 SDK 开发方式的产品类型，例如小夜灯填写产品名称，型号，通讯协议选择蓝牙，点击创建产品。

创建产品

海量成熟方案，超多个性化面板，极速落地产品智能化，找智能方案，到方案中心

前往方案中心

标准类目

行业解决方案

电工

照明

传感

大家电

小家电

厨房电器

运动健康

安防监控

网关中控

户外出行

节能能源

教育娱乐

工农业

其他

找不到品类?

已选品类 重新选择

小夜灯

已选方案 重新选择

小夜灯  
适用于小夜灯等设备  
开发方式: 自定义方案  
方案详情

3 完善产品信息

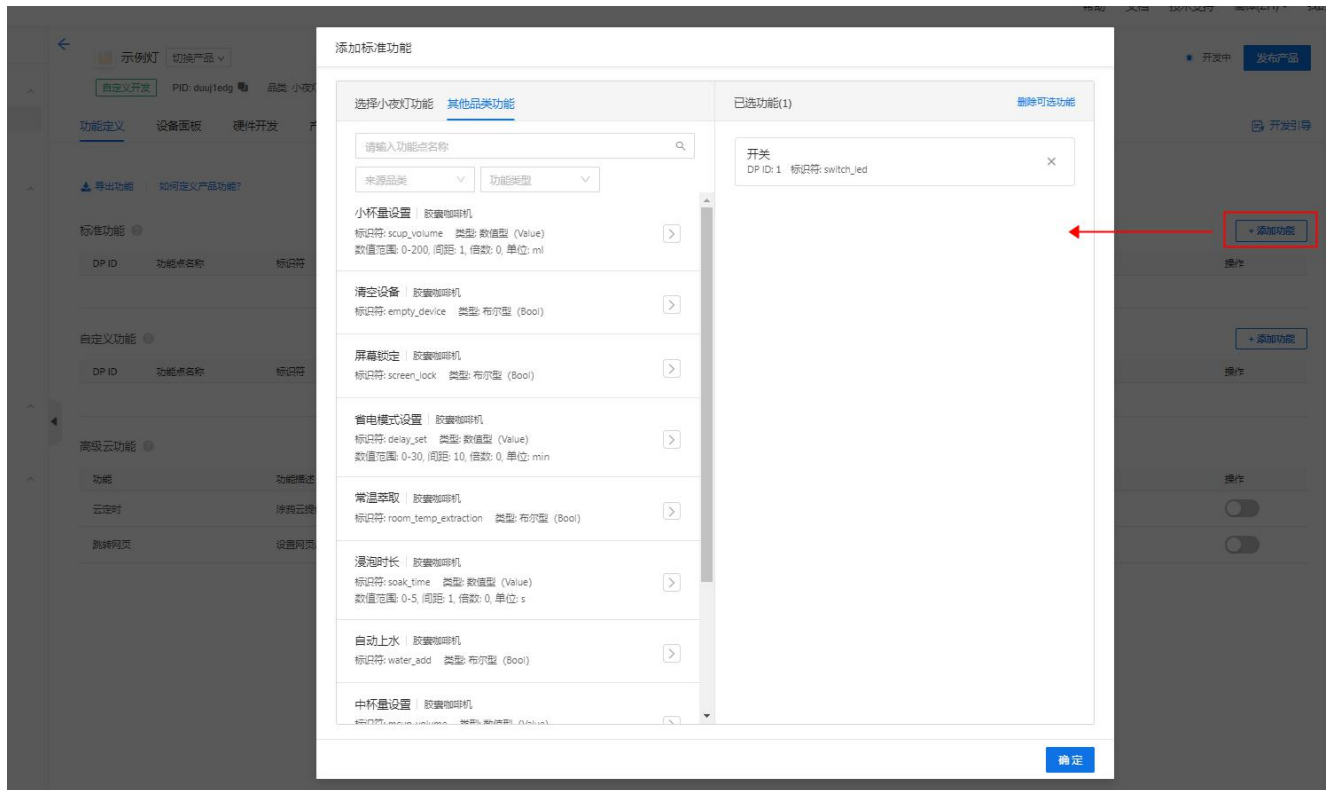
\* 产品名称: 示例灯

产品型号: 01

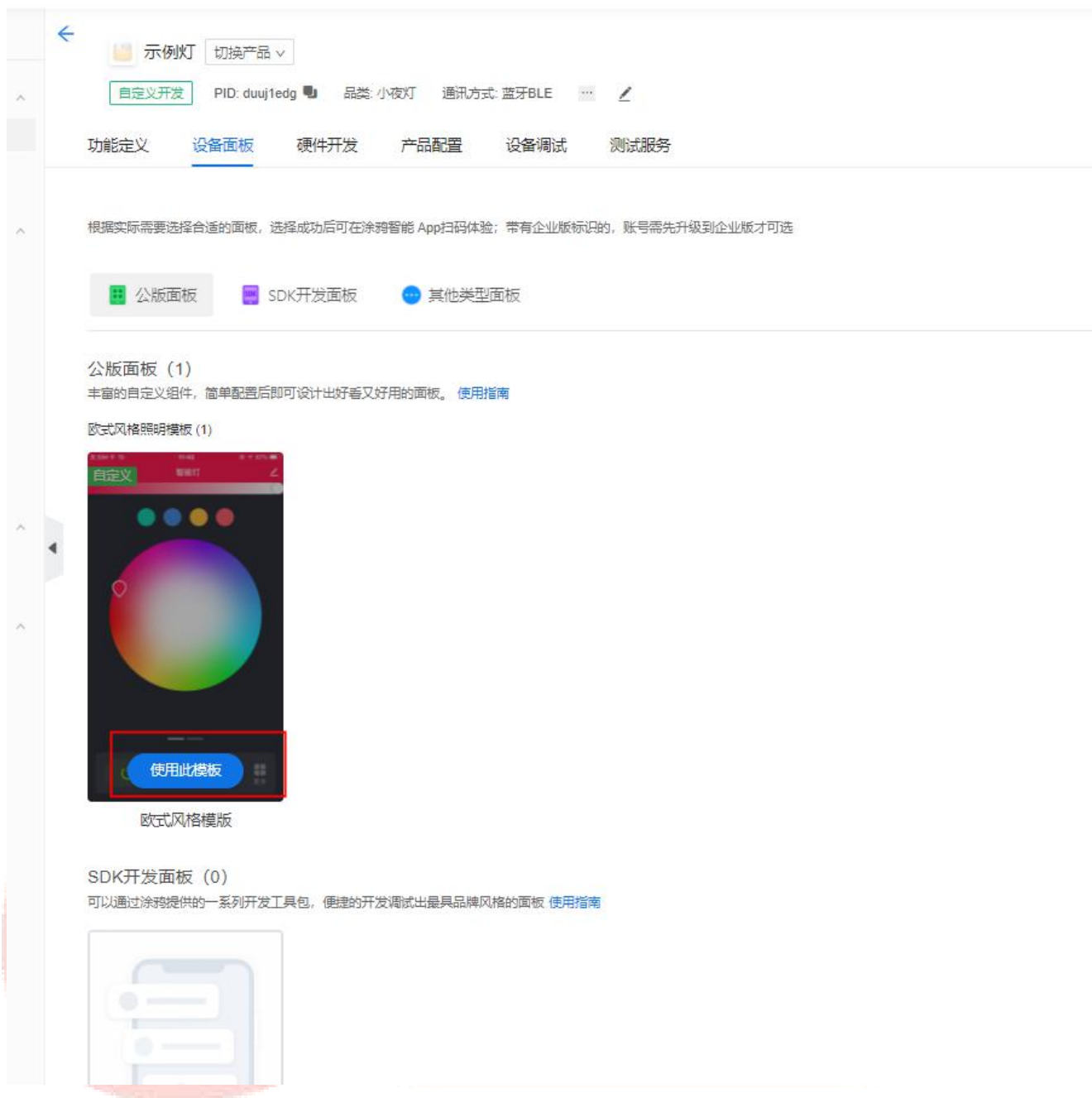
\* 通讯协议: ☐ WiFi-蓝牙 ☐ Wi-Fi ☐ 蓝牙Mesh(SIG) ☐ Zigbee ☒ 蓝牙

创建产品

### 3.进入产品开发页面，选择要添加的功能。



#### 4.在设备面板界面选择一个面板



5.在硬件开发界面选云端对接方式为自研模组 SDK 开发，注意：如果所选的产品类型无此对接方式，将无法申请调试用授权码。填写好芯片平台，操作系统，编译工具链和模组名称后，点击确定。

产品

概述

产品

App

云开发

数据

运营

采购

增值

产品

极速智能化

产品

生产管理

设备管理

固件升级

设备明细

子设备明细

消息推送

消息监控

设备日志

更换PID

开发者工具

三明治开发板

产品语音能力

产品接入

示例灯 切换产品

自定义开发 PID: duuj1edg 品类: 小夜灯 通讯方式: 蓝牙BLE

功能定义 设备面板 硬件开发 产品配置 设备调试 测试服务

根据您的选择对接方式，选择模组以及相应固件，同时可以下载开发资料、购买调试模组。

已选云端对接方式

涂鸦标准模组MCU SDK开发 自研模组SDK开发

开发 SDK

请选择您的芯片平台、操作系统及编译工具链类型，获取对应的嵌入式程序开发SDK [找不到您想要的SDK?](#)

\* 芯片平台: 请选择芯片平台

\* 操作系统: 请选择操作系统

\* 编译工具链: 请选择编译工具链

\* 模组名称: 输入模组名称

确定

下载资料

公共资料

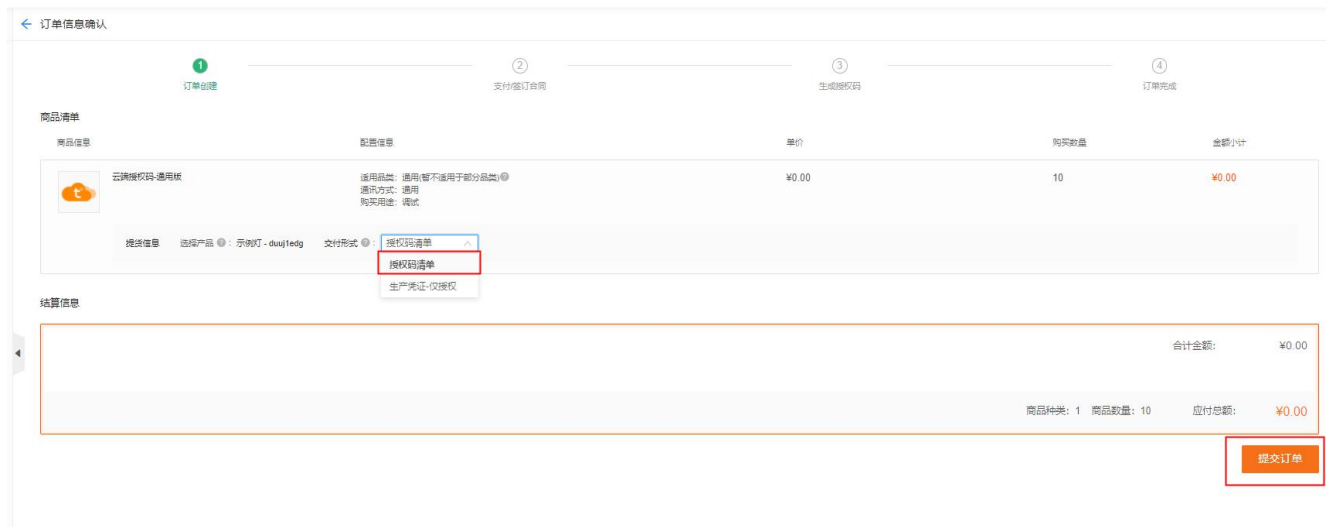
将开发板(模组)与控制板链接，通过涂鸦智能 App进行联网测试

标准功能	<a href="#">查看</a>
配置 App 界面	<a href="#">查看</a>
自定义功能	<a href="#">查看</a>
固件升级	<a href="#">查看</a>
多语言管理	<a href="#">查看</a>
推送设备消息	<a href="#">查看</a>
配置场景联动	<a href="#">查看</a>

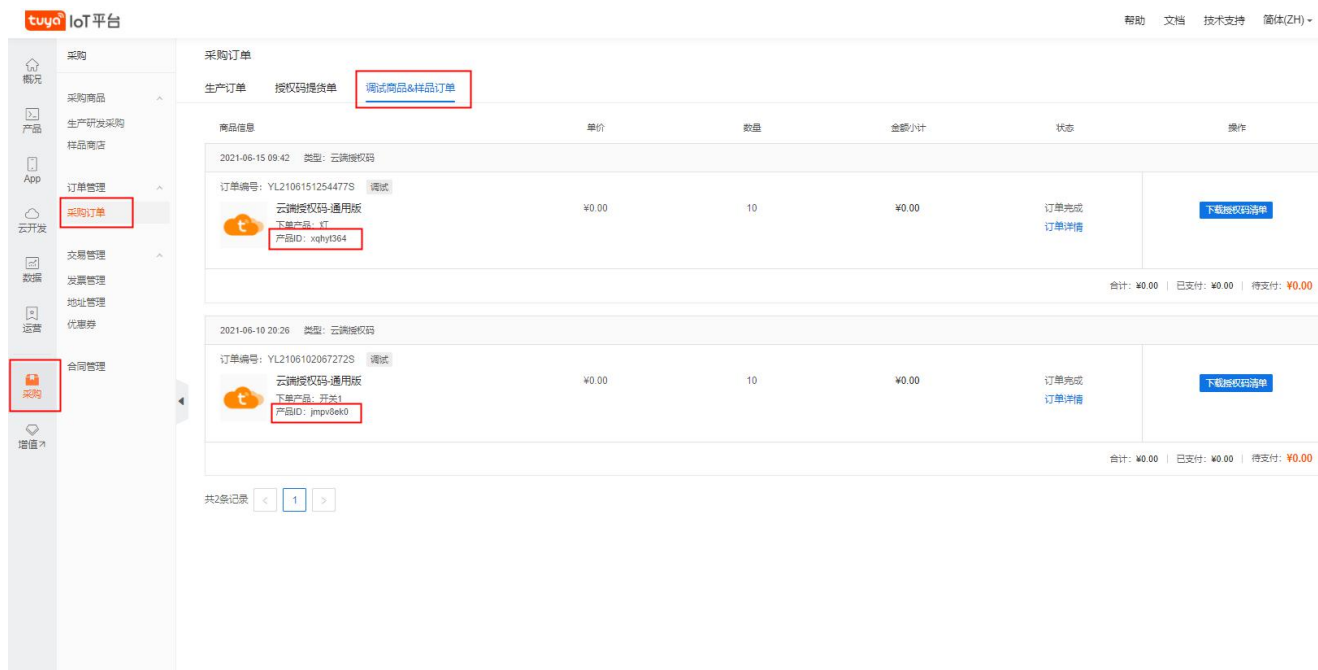
## 6.之后点击免费获取 10 个激活码



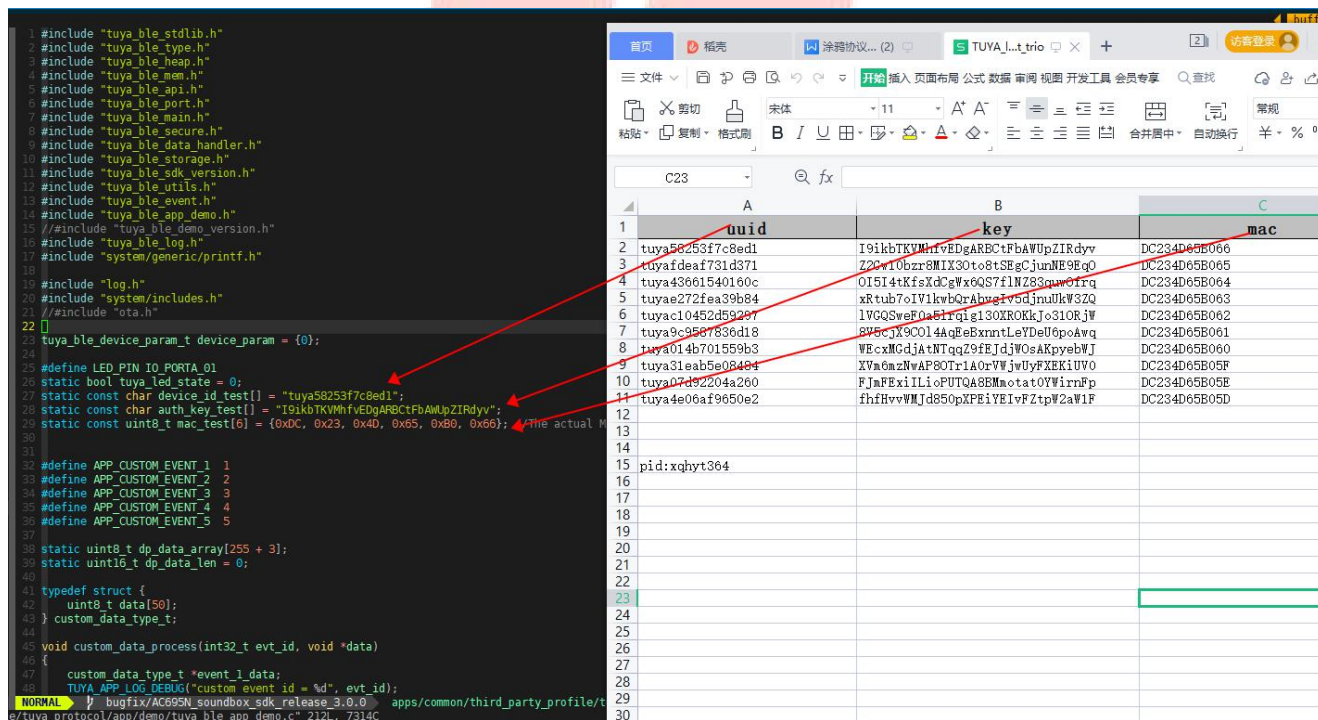
## 7.在订单确认界面选择交付形式为：授权码清单。点击提交订单。



8.提交订单后，在样品订单里就可以下载授权码了。授权码清单和每个产品的 pid 要对应使用。



9.申请的授权码对应代码 tuya\_ble\_app\_demo.c 里的 device\_id\_test, auth\_key\_test, mac\_test。





10.pid 对应 tuyu\_ble\_app\_demo.h 里的 APP\_PRODUCT\_ID

```
1 tuyu_ble_app_demo.c 2 tuyu_ble_app_demo.h
1 #ifndef TUYA_BLE_APP_DEMO_H_
2 #define TUYA_BLE_APP_DEMO_H_
3
4
5 #ifdef __cplusplus
6 extern "C" {
7 #endif
8
9
10
11 #define APP_PRODUCT_ID "xqhyt364"
12
13 #define APP_BUILD_FIRMNAME "tuya_ble_sdk_app_demo_nrf52832"
14
15 //固件版本
16 #define TY_APP_VER_NUM 0x0100
17 #define TY_APP_VER_STR "1.0"
18
19 //硬件版本
20 #define TY_HARD_VER_NUM 0x0100
21 #define TY_HARD_VER_STR "1.0"
22
23
24 /*
25 typedef enum {
26     white,
27     colour,
28     scene,
29     music,
30 } tuyu_light_mode;
31 */
32
33
34 void tuyu_ble_app_init(void);
35
36
37 #ifdef __cplusplus
38 }
39 #endif
40
41 #endif //
```



## 二、涂鸦协议程序开发说明

### 1. SDK 配置

打开 SDK 的板级配置 apps/soundbox/board/br23/board\_ac695x\_smartbox/board\_ac695x\_smartbox.h, 按如下配置。

```
1 board_ac695x_smartbox.h
590 #define TCFG_SC7A20_EN 0
591 #define TCFG_STK8321_EN 0
592 #define TCFG_GSENOR_USER_IIC_TYPE 0 //0:软件IIC 1:硬件IIC
593
594 //*****
595 // 系统配置 //
596 //*****
597 #define TCFG_AUTO_SHUT_DOWN_TIME 0 //没有蓝牙连接自动关机时间
598 #define TCFG_SYS_LVD_EN 1 //电量检测使能
599 #define TCFG_POWER_ON_NEED_KEY 0 //是否需要按按键开机配置
600 #define TWFG_APP_POWERON_IGNORE_DEV 3700//上电忽略挂载设备, 0时不忽略, 非0则n毫秒忽略
601
602 //*****
603 // AI配置 //
604 //*****
605
606 #define CONFIG_APP_BT_ENABLE // AI功能、流程总开关
607 #define CONFIG_IDY_CODE_DEBUG //工程整理代码导致编译不过, 加一下调试代码让工程编译通过
608
609
610 #ifdef CONFIG_APP_BT_ENABLE
611 #define TRANS_DATA_EN 0
612 #define SMART_BOX_EN 0
613 #define TUYA_DEMO_EN 1
614 #else
615 #define TRANS_DATA_EN 0
616 #define SMART_BOX_EN 0
617 #define TUYA_DEMO_EN 0
618 #endif
619
620 #if (SMART_BOX_EN) //rcsp需要打开ble
621 #define CONFIG_DOUBLE_BANK_ENABLE 0
622 #define RCSP_UPDATE_EN 1 //是否支持rcsp升级
623 #define OTA_TWS_SAME_TIME_ENABLE 0 //是否支持TWS同步升级
624 #define UPDATE_MD5_ENABLE 1 //升级是否支持MD5校验
625 #define RCSP_FILE_OPT 1
626 #define JL_EARPHONE_APP_EN 1
627 #define TCFG_LFN_EN 1
628 #define TCFG_BS_DEV_PATH_EN 1
629 #else
630 #define OTA_TWS_SAME_TIME_ENABLE 0
631 #define RCSP_UPDATE_EN 0
632 #define UPDATE_MD5_ENABLE 0 //升级是否支持MD5校验
633 #define RCSP_FILE_OPT 0
634 #define JL_EARPHONE_APP_EN 0
635 #endif
636
637 #if RCSP_UPDATE_EN
```

## 2. 应用开发

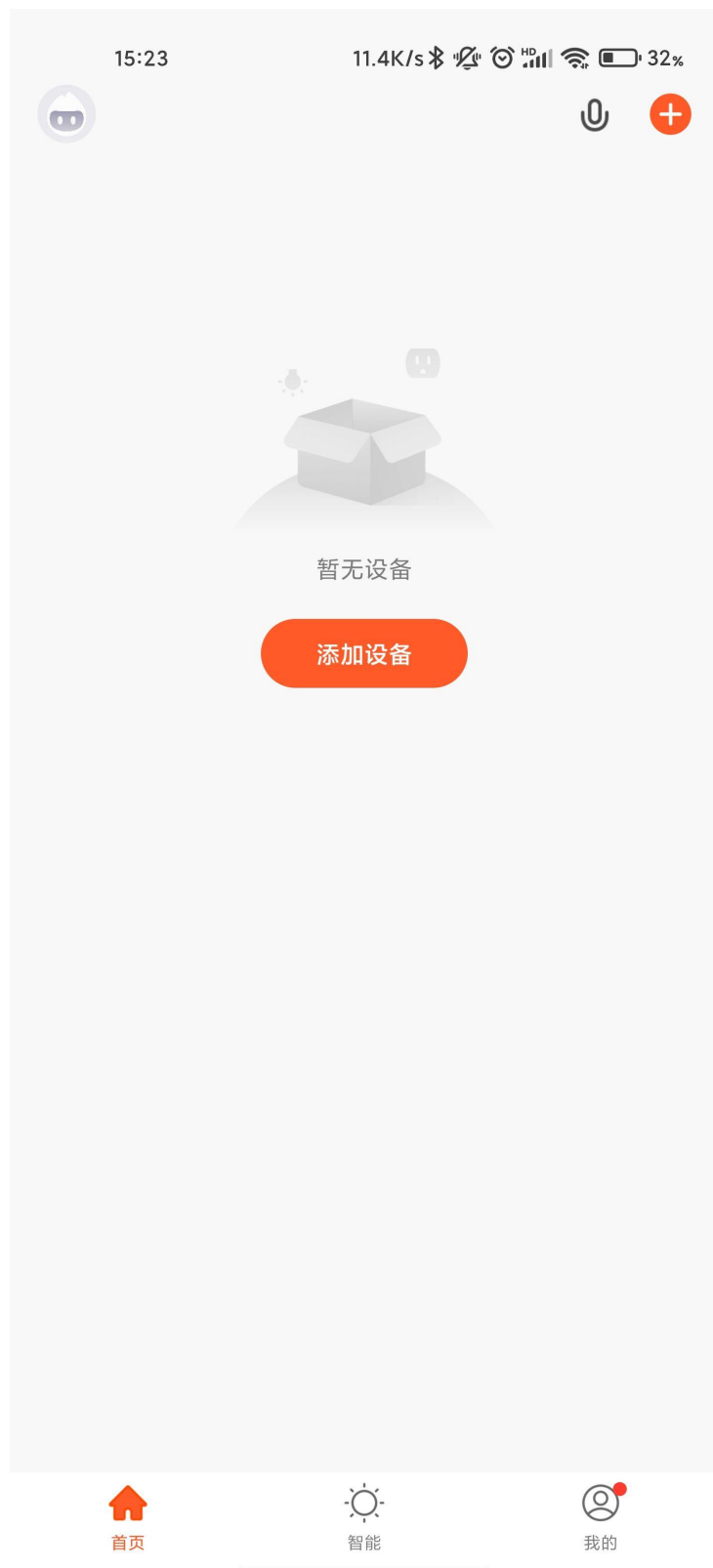
设备在收到 APP 端的消息后会进入 tuya\_ble\_app\_demo.c 文件里的 tuya\_cb\_handler 回调函数里,控制消息对应 TUYA\_BLE\_CB\_EVT\_DP\_WRITE 类型。通过 tuya\_data\_parse 函数对消息进行解析后,对设备进行控制操作。控制消息的结构参考 TUYA BLE SDK User Guide.pdf 的 5.10 节。

```
104 }
105 }
106
107 static uint16_t sn = 0;
108 static uint32_t time_stamp = 1587795793;
109 static void tuya_cb_handler(tuya_ble_cb_evt_param_t *event)
110 {
111     int16_t result = 0;
112     printf("tuya cb handler event->evt=%d\n", event->evt);
113     switch (event->evt) {
114     case TUYA_BLE_CB_EVT_CONNECTE_STATUS:
115         TUYA_APP_LOG_INFO("received tuya ble conncet status update event,current connect status = %d", event->connect_status);
116         break;
117     case TUYA_BLE_CB_EVT_DP_WRITE:
118         tuya_data_parse(event);
119         tuya_ble_dp_data_report(event->dp_write_data.p_data, event->dp_write_data.data_len); //1
120         break;
121     case TUYA_BLE_CB_EVT_DP_DATA_REPORT_RESPONSE:
122         TUYA_APP_LOG_INFO("received dp data report response result code =%d", event->dp_response_data.status);
123         break;
124     case TUYA_BLE_CB_EVT_DP_DATA WITH TIME REPORT_RESPONSE:
125         TUYA_APP_LOG_INFO("received dp data report response result code =%d", event->dp_response_data.status);
126         break;
127     case TUYA_BLE_CB_EVT_DP_DATA WITH FLAG REPORT_RESPONSE:
128         TUYA_APP_LOG_INFO("received dp data with flag report response sn = %d , flag = %d , result code =%d", event->dp_with_flag_
129         response_data.sn, event->dp_with_flag_response_data.mode
130         , event->dp_with_flag_response_data.status);
131         break;
132     case TUYA_BLE_CB_EVT_DP_DATA WITH FLAG AND TIME REPORT_RESPONSE:
133         TUYA_APP_LOG_INFO("received dp data with flag and time report response sn = %d , flag = %d , result code =%d", event->dp_w
134         th_flag_and_time_response_data.sn,
135         event->dp_with_flag_and_time_response_data.mode, event->dp_with_flag_and_time_response_data.status);
136         break;
137     case TUYA_BLE_CB_EVT_UNBOUND:
138         TUYA_APP_LOG_INFO("received unbound req");
139         break;
140     case TUYA_BLE_CB_EVT_ANOMALY_UNBOUND:
141         TUYA_APP_LOG_INFO("received anomaly unbound req");
142         break;
143     case TUYA_BLE_CB_EVT_DEVICE_RESET:
144         break;
145     }
```

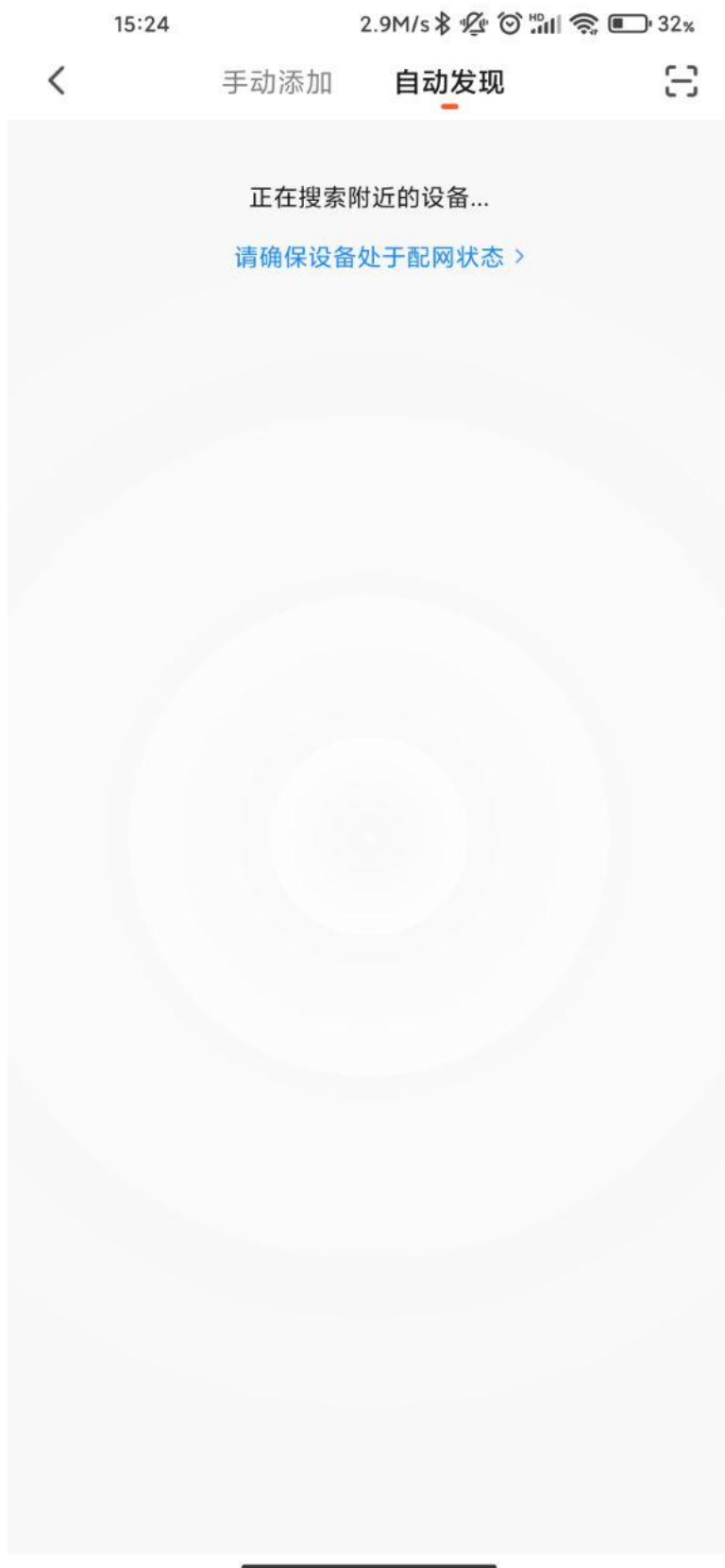
```
8
9 void tuyu_data_parse(tuya_ble_cb_evt_param_t *event)
10 {
11     uint8_t *buf = event->dp_received_data.p_data;
12     uint32_t sn = event->dp_received_data.sn;
13     put_buf(buf, event->dp_received_data.data_len);
14     struct {
15         uint8_t id;
16         uint8_t type;
17         uint16_t len;
18         uint8_t data;
19     } p_dp_data;
20
21     p_dp_data.id = buf[0];
22     p_dp_data.type = buf[1];
23     p_dp_data.len = 0x0100;
24     p_dp_data.data = buf[4];
25
26     printf("\n\n<----- tuyu_data_parse ----->");
27     printf("sn = %d, id = %d, type = %d, len = %d, data = %d", sn, p_dp_data.id, p_dp_data.type, p_dp_data.len, p_dp_data.data);
28
29     switch (buf[0]) {
30     case 1:
31         printf("tuya switch control, onoff set to: %d\n", p_dp_data.data);
32         tuyu_led_state = p_dp_data.data;
33         gpio_direction_output(LED_PIN, tuyu_led_state);
34         break;
35         //case 2:
36         //printf("tuya mode control, mode set to: %d\n", buf[3]);
37         //break;
38     default:
39         printf("unknow control msg len = %d, data:", buf[2]);
40         break;
41     }
42
43     #if (TUYU_BLE_PROTOCOL_VERSION_HIGN == 0x03)
44         tuyu_ble_dp_data_report(&p_dp_data, 5); //1
45     #endif
46     #if (TUYU_BLE_PROTOCOL_VERSION_HIGN == 0x04)
47         tuyu_ble_dp_data_send(sn, DP_SEND_TYPE_ACTIVE, DP_SEND_FOR_CLOUD_PANEL, DP_SEND_WITH_RESPONSE, &p_dp_data, 5);
48     #endif
49
50 }
```

### 三、APP 使用实例

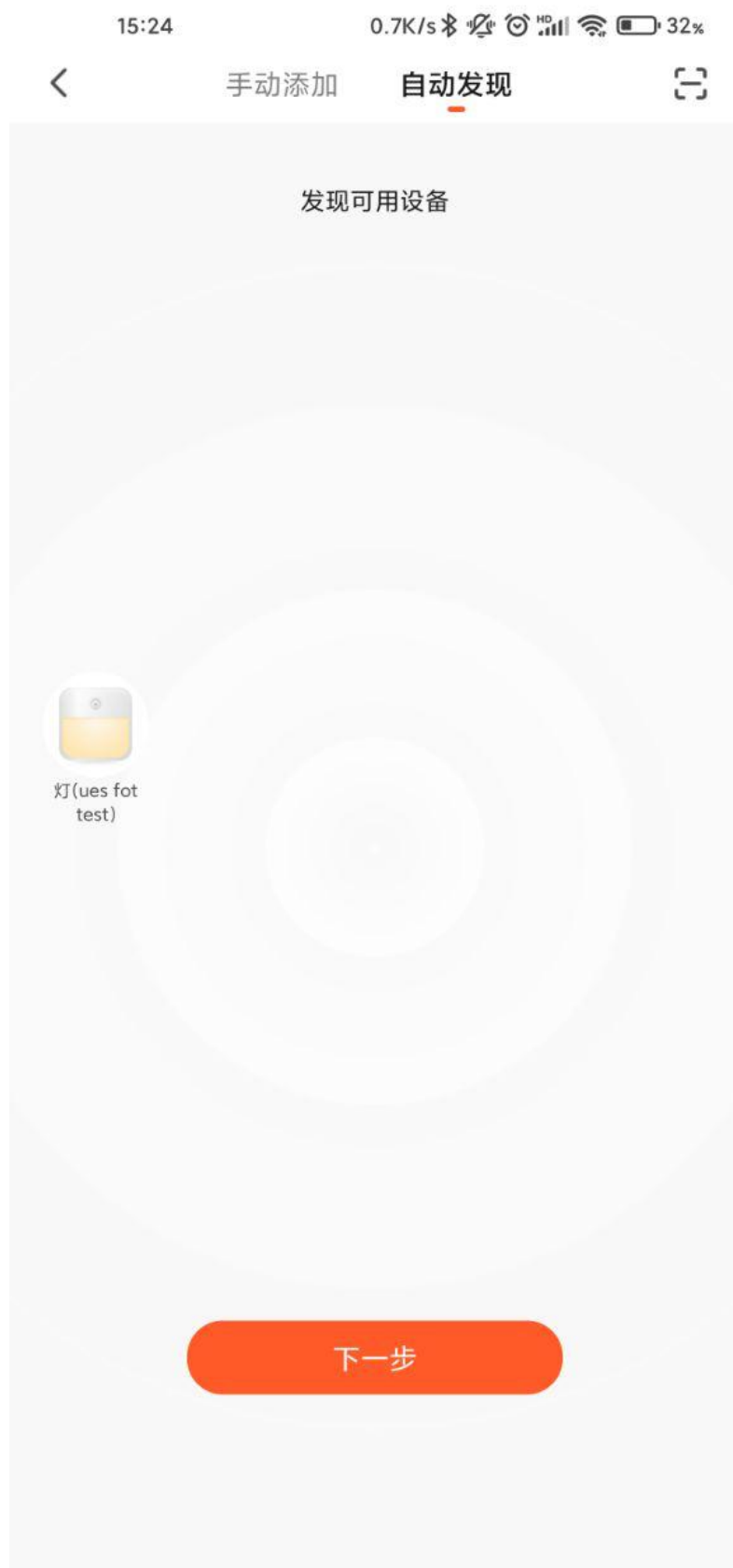
#### 1. 下载涂鸦智能手机 APP 后打开



## 2. 点击添加设备，进入自动发现页面

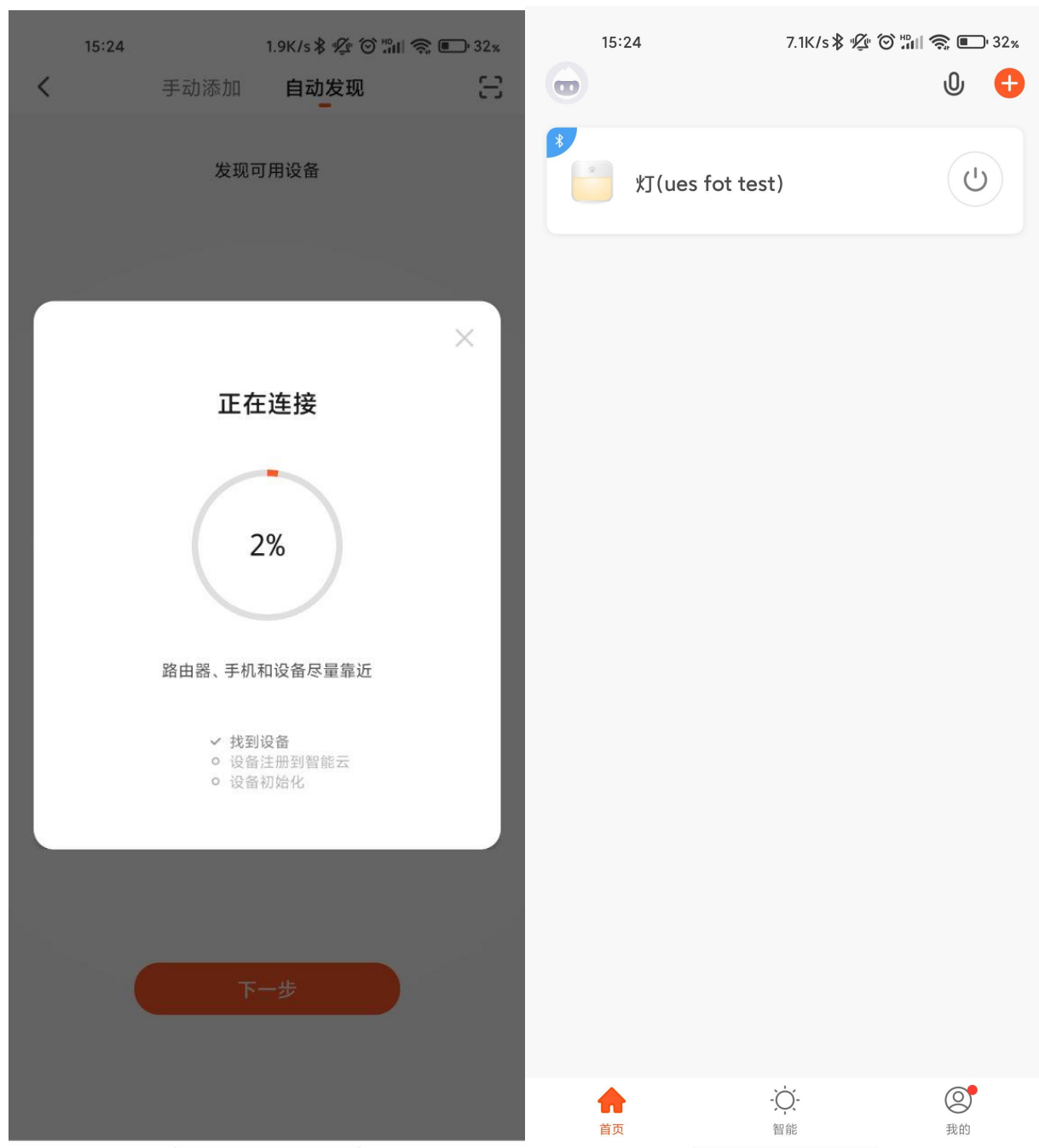


### 3. 发现到设备后点击下一步



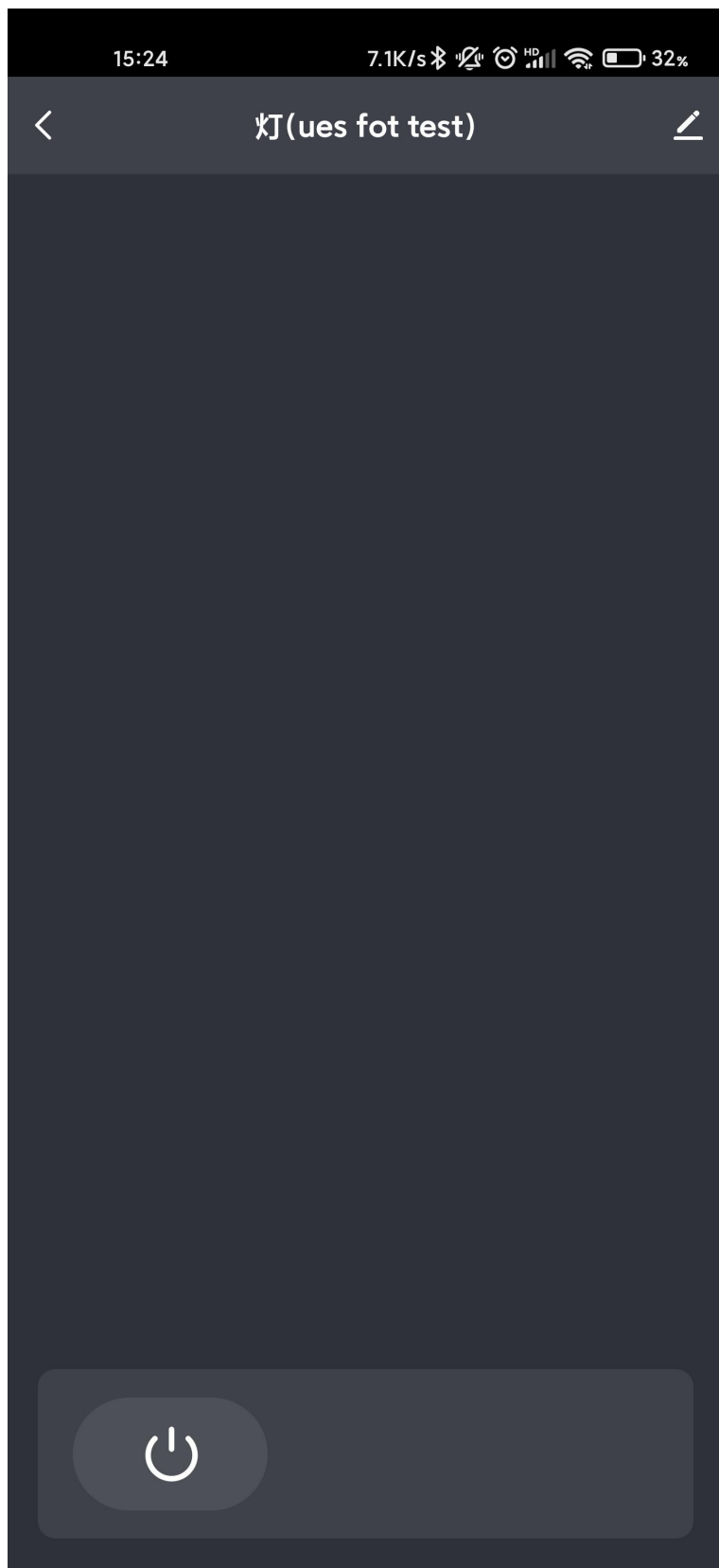


#### 4. 连接完成后在主界面会显示连接的设备





## 5. 进入设备控制面板即可控制设备



6. 点击右上角的按钮可以进行解绑或者固件升级操作

