

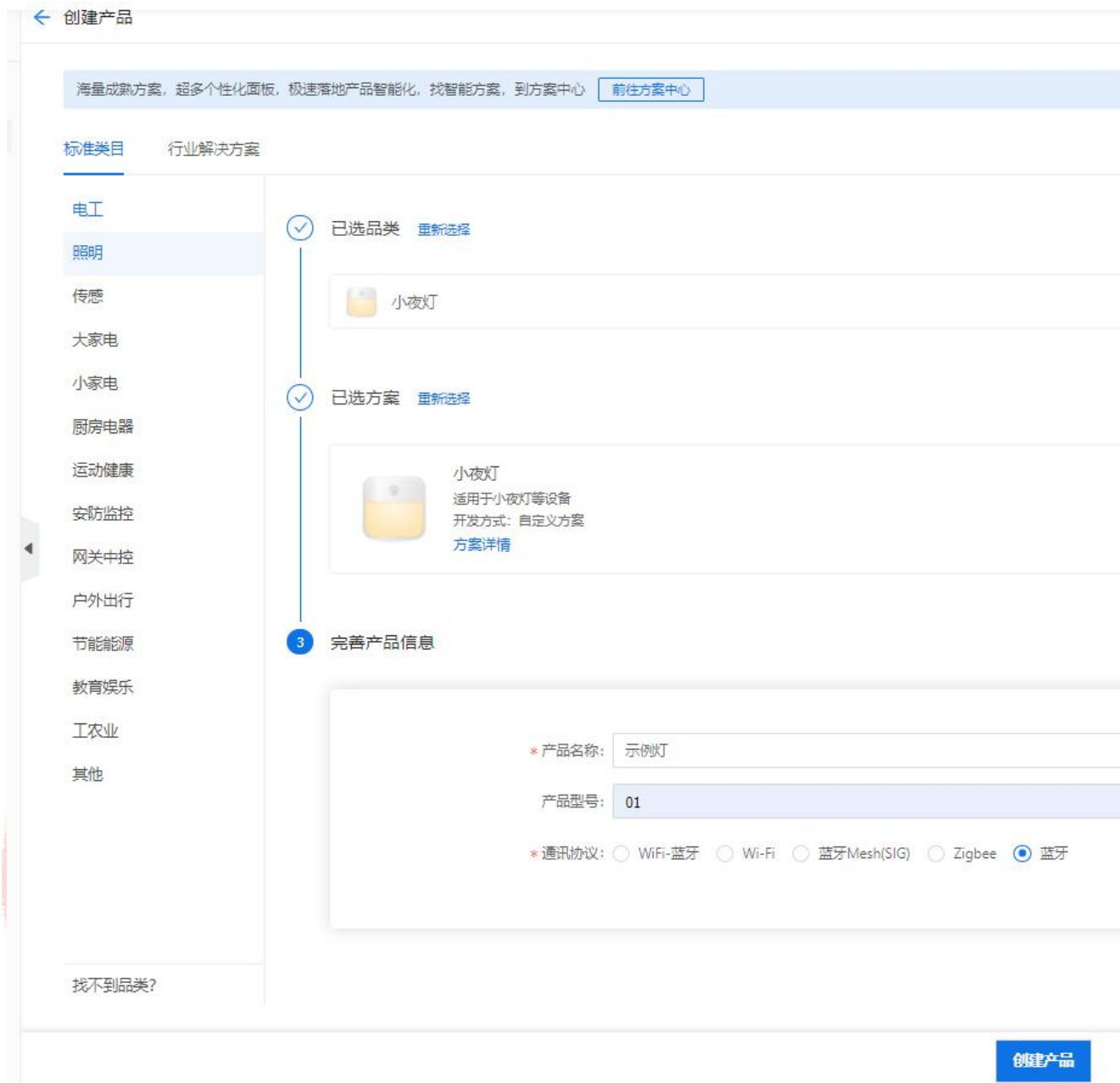
涂鸦协议开发说明

一、测试用授权码申请

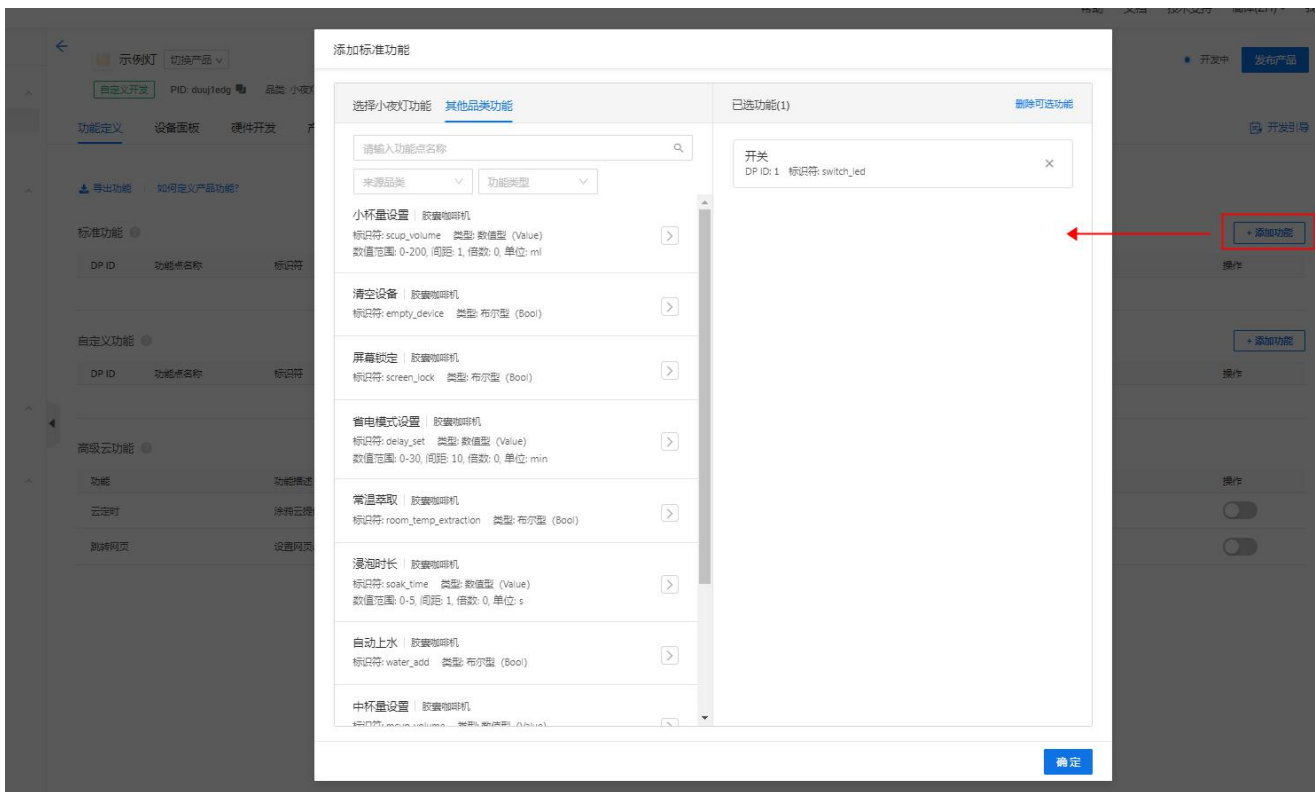
1.打开涂鸦 IOT 平台 <https://auth.tuya.com/>，注册并登陆。打开产品->产品开发界面，点击创建产品。



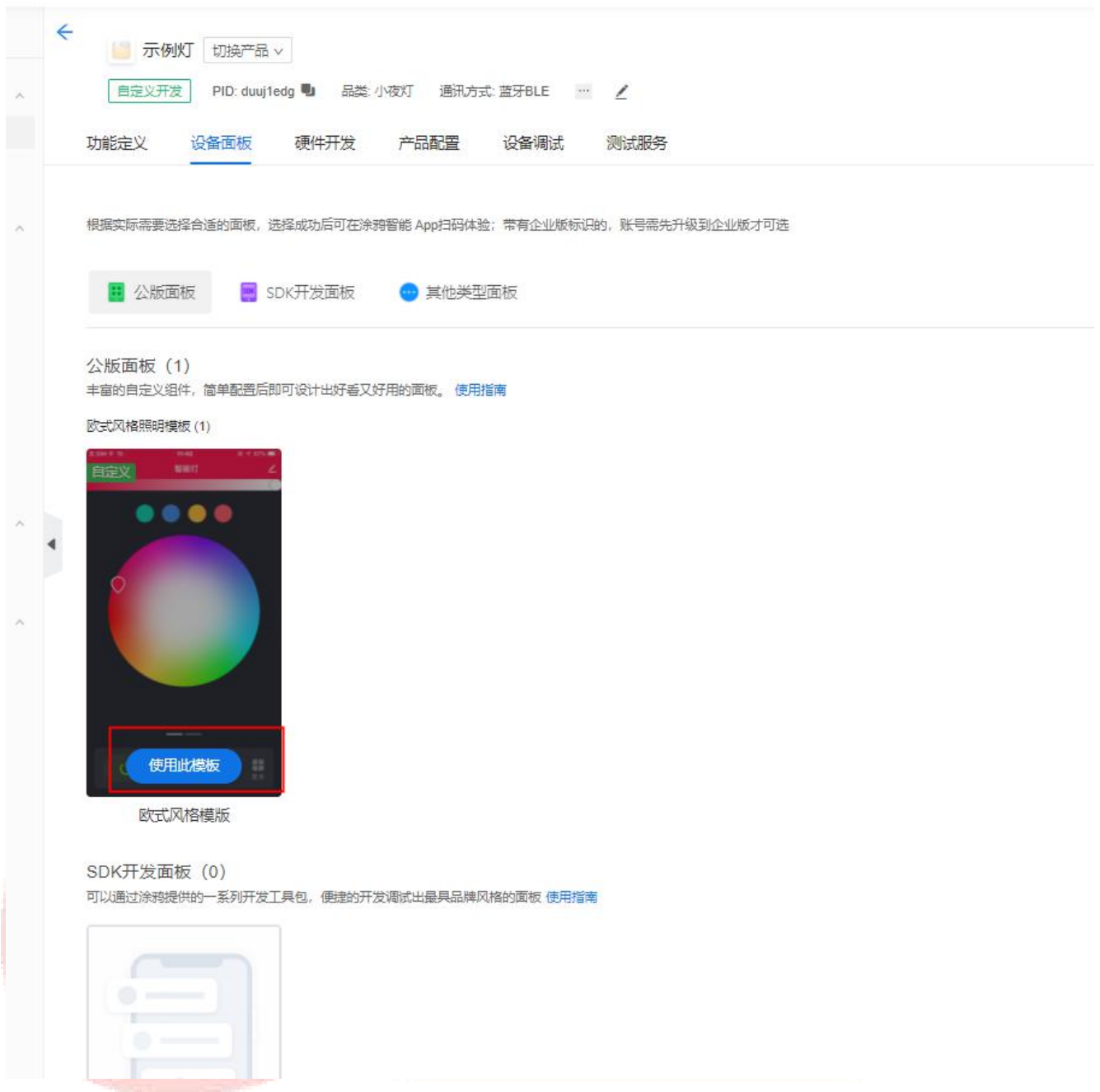
2.选择一个具有自研模组 SDK 开发方式的产品类型，例如小夜灯填写产品名称，型号，通讯协议选择蓝牙，点击创建产品。



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



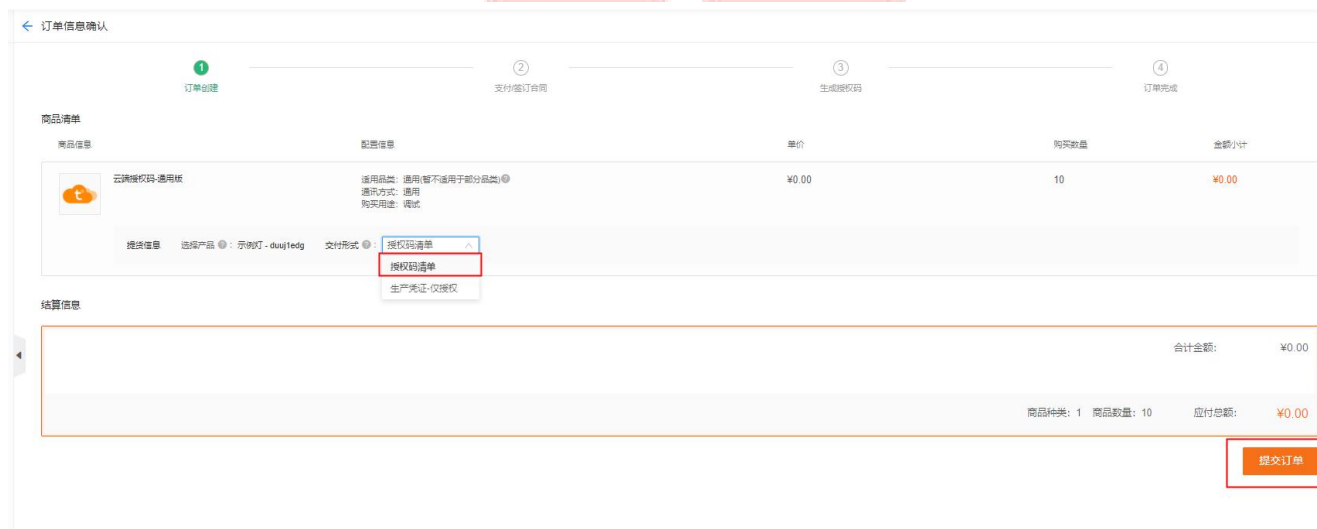
5.在硬件开发界面选云端对接方式为自研模组 SDK 开发，注意：如果所选的产品类型无此对接方式，将无法申请调试用授权码。填写好芯片平台，操作系统，编译工具链和模组名称后，点击确定。



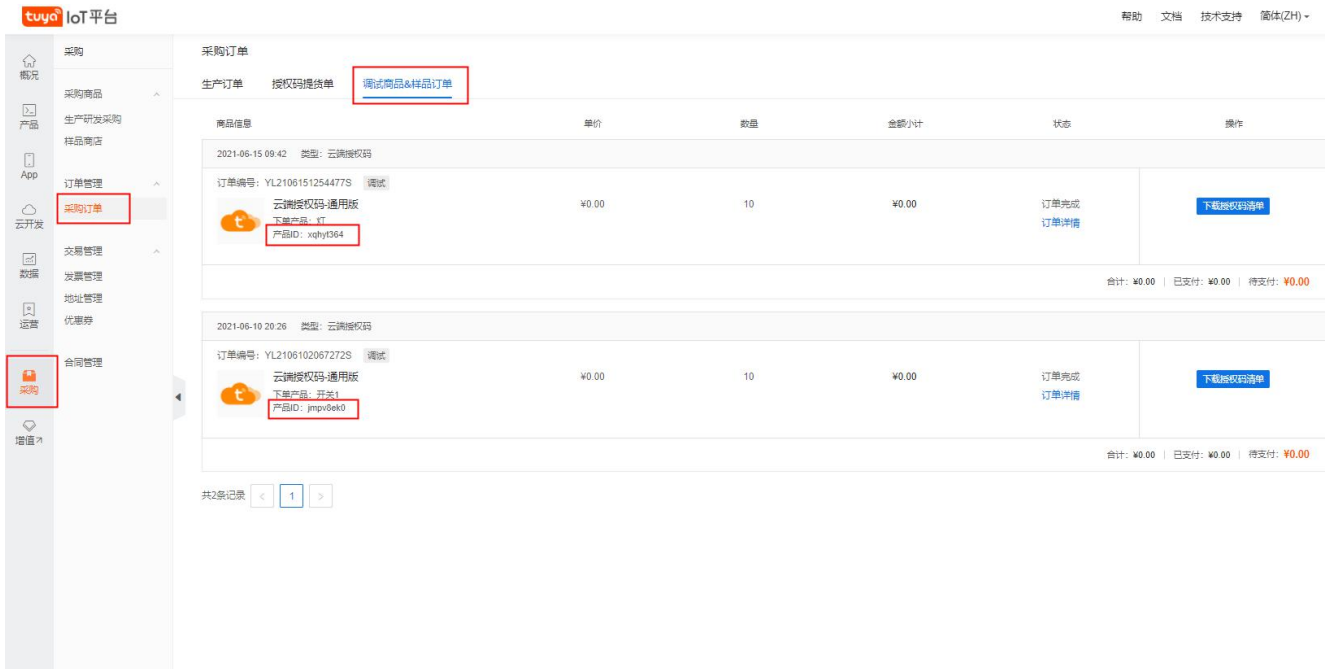
6.之后点击免费获取 10 个激活码



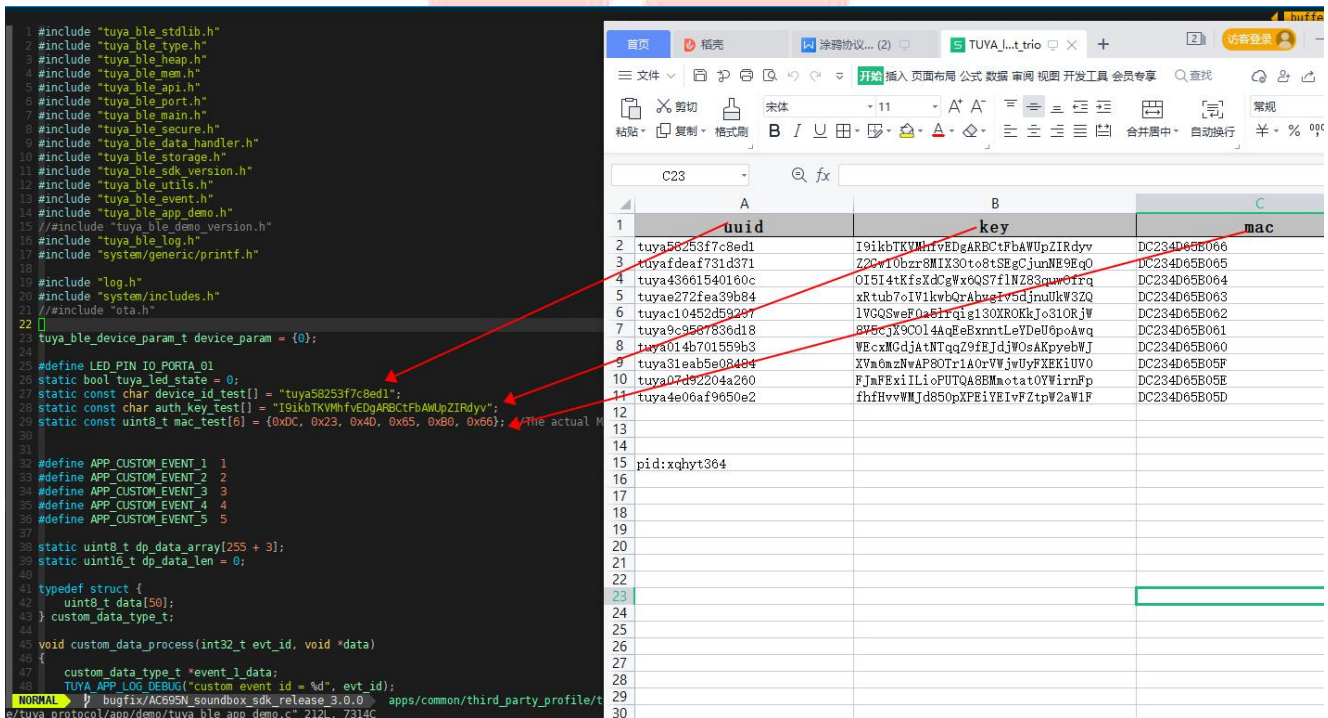
7.在订单确认界面选择交付形式为：授权码清单。点击提交订单。



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 tuya_ble_app_demo.h 里的 APP_PRODUCT_ID

```
1 tuya_ble_app_demo.c 2 tuya_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 } tuya_light_mode;
31 */
32
33
34 void tuya_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. isd_config.ini 文件配置

该文件需要添加一下内容

```
USERIF_ADR=AUTO;
```

```
USERIF_LEN=1K;
```

```
USERIF_OPT=1;
```

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 connctet 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
124         break;
125     case TUYA_BLE_CB_EVT_DP_DATA_WTH_TIME_REPORT_RESPONSE:
126         TUYA_APP_LOG_INFO("received dp data report response result code =%d", event->dp_response_data.status);
127         break;
128     case TUYA_BLE_CB_EVT_DP_DATA_WITH_FLAG_REPORT_RESPONSE:
129         TUYA_APP_LOG_INFO("received dp data with flag report response sn = %d , flag = %d , result code =%d", event->dp_with_flag_
130     esponse_data.sn, event->dp_with_flag_response_data.mode
131         , event->dp_with_flag_response_data.status);
132
133         break;
134     case TUYA_BLE_CB_EVT_DP_DATA_WITH_FLAG_AND_TIME_REPORT_RESPONSE:
135         TUYA_APP_LOG_INFO("received dp data with flag and time report response sn = %d , flag = %d , result code =%d", event->dp_w
136     th_flag_and_time_response_data.sn,
137         event->dp_with_flag_and_time_response_data.mode, event->dp_with_flag_and_time_response_data.status);
138
139         break;
140     case TUYA_BLE_CB_EVT_UNBOUND:
141         TUYA_APP_LOG_INFO("received unbound req");
142
143         break;
144     case TUYA_BLE_CB_EVT_ANOMALY_UNBOUND:
145         TUYA_APP_LOG_INFO("received anomaly unbound req");
146
147         break;
148     case TUYA_BLE_CB_EVT_DEVICE_RESET:
```

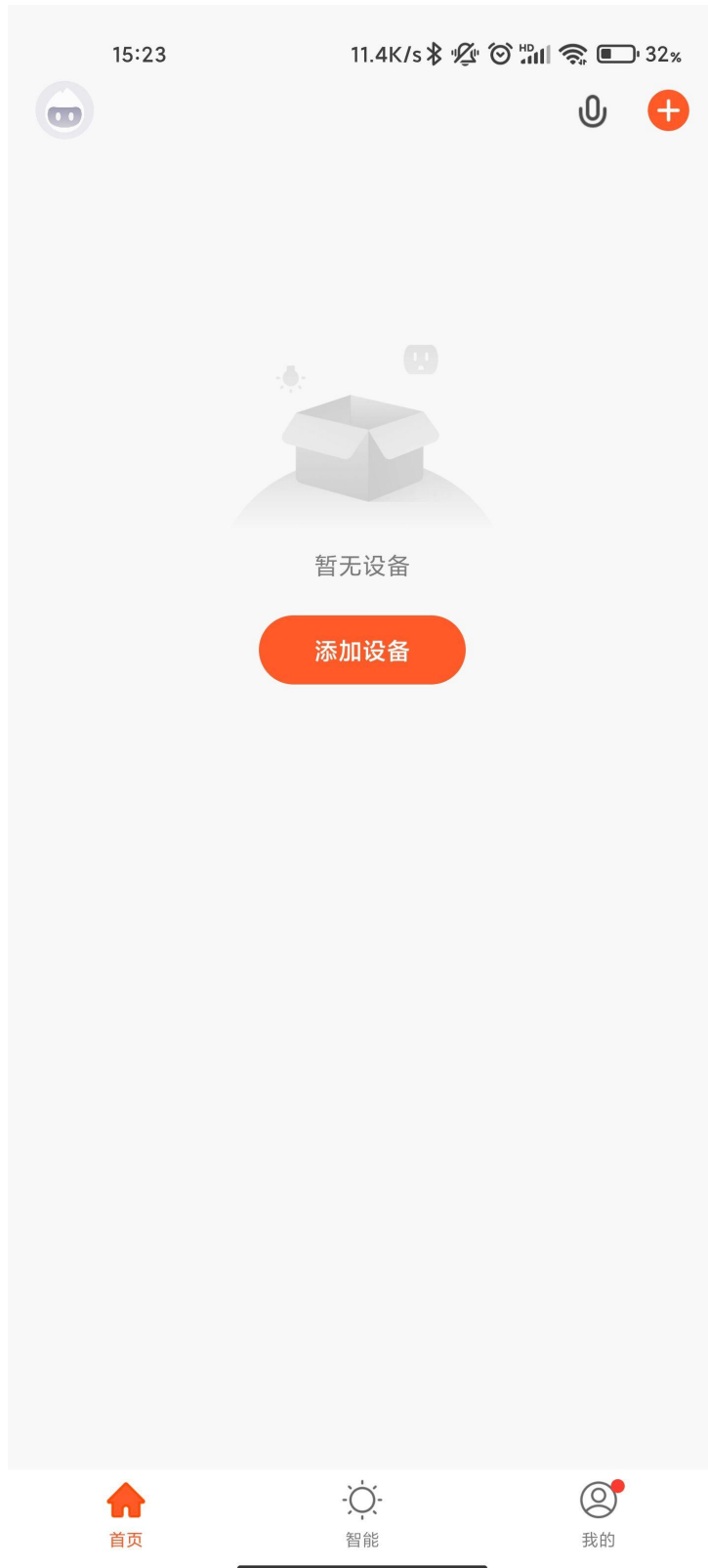


```
8
9 void tuya_data_parse(tuya_ble_cb_evt_param_t *event)
10 {
11     uint8_t *buf = event->dp_received_data.p_data;
12     uint32_t sn = event->dp_received_data.sn;
13     put_buf(buf, event->dp_received_data.data_len);
14     struct {
15         uint8_t id;
16         uint8_t type;
17         uint16_t 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<----- tuya_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
30     switch (buf[0]) {
31     case 1:
32         printf("tuya switch control, onoff set to: %d\n", p_dp_data.data);
33         tuya_led_state = p_dp_data.data;
34         gpio_direction_output(LED_PIN, tuya_led_state);
35         break;
36         //case 2:
37         //printf("tuya mode control, mode set to: %d\n", buf[3]);
38         //break;
39     default:
40         printf("unknow control msg len = %d, data:", buf[2]);
41         break;
42     }
43     #if (TUYA_BLE_PROTOCOL_VERSION_HIGN == 0x03)
44         tuya_ble_dp_data_report(&p_dp_data, 5); //1
45     #endif
46     #if (TUYA_BLE_PROTOCOL_VERSION_HIGN == 0x04)
47         tuya_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

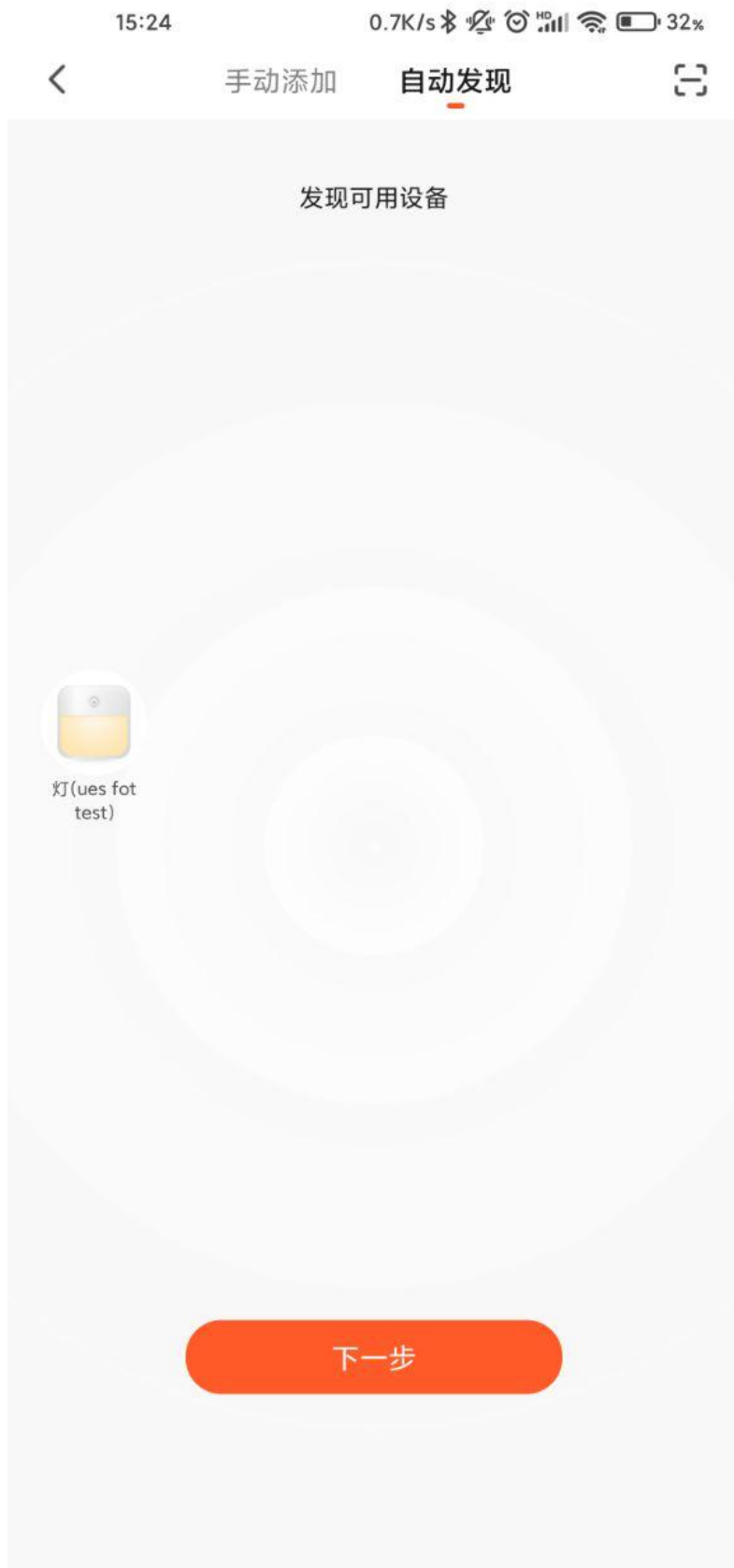
1. 下载涂鸦智能手机 APP 后打开



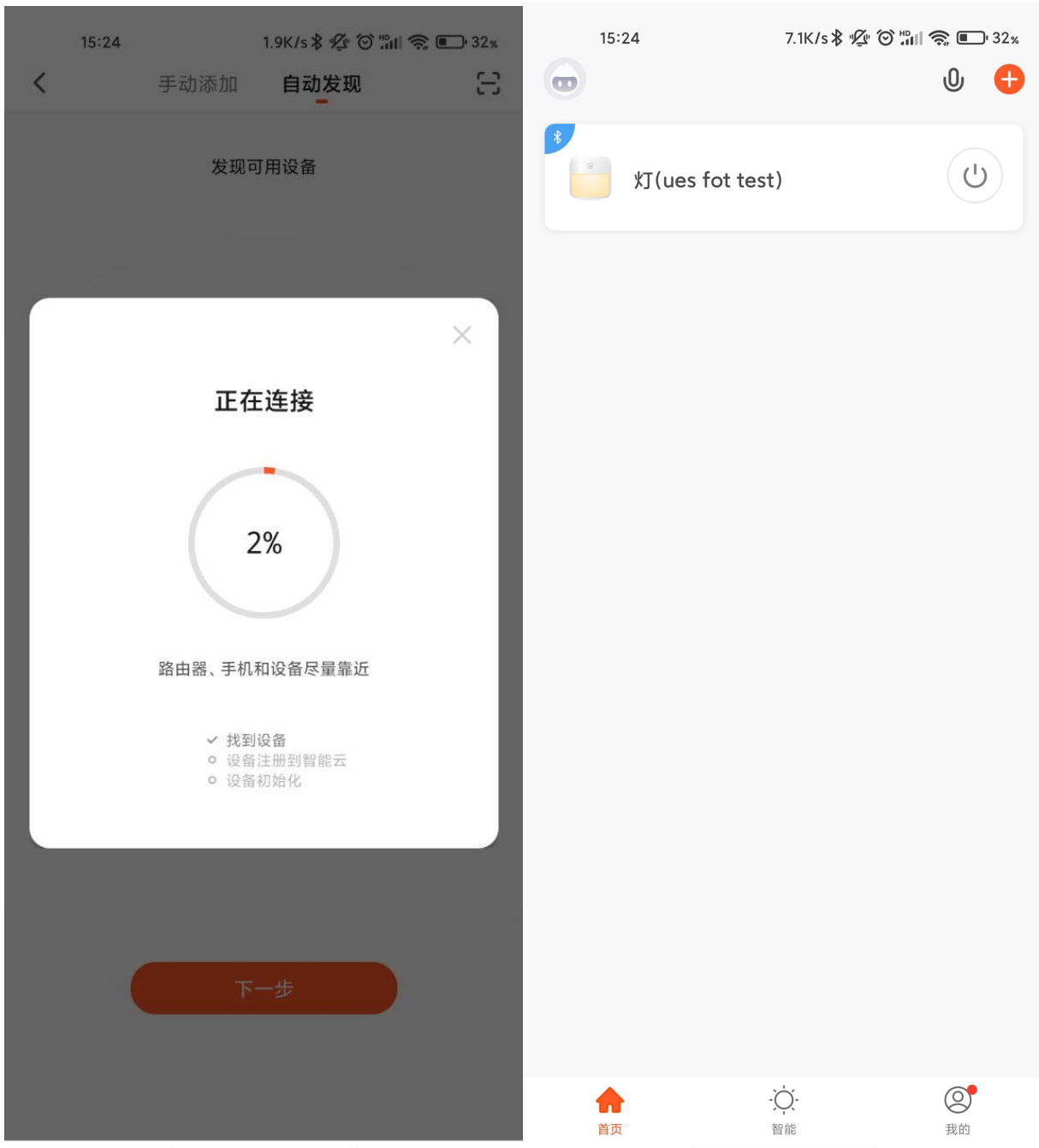
2. 点击添加设备，进入自动发现页面



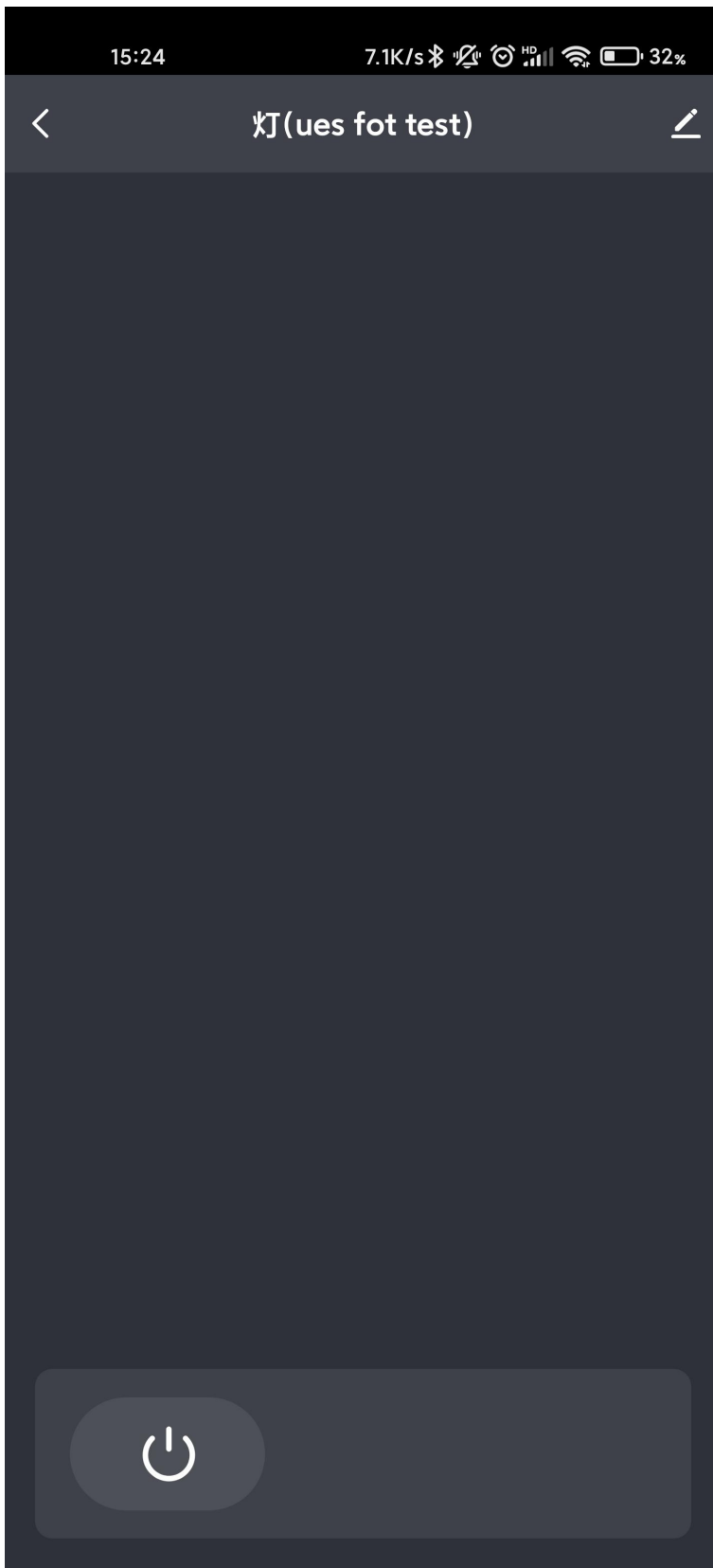
3. 发现到设备后点击下一步




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

