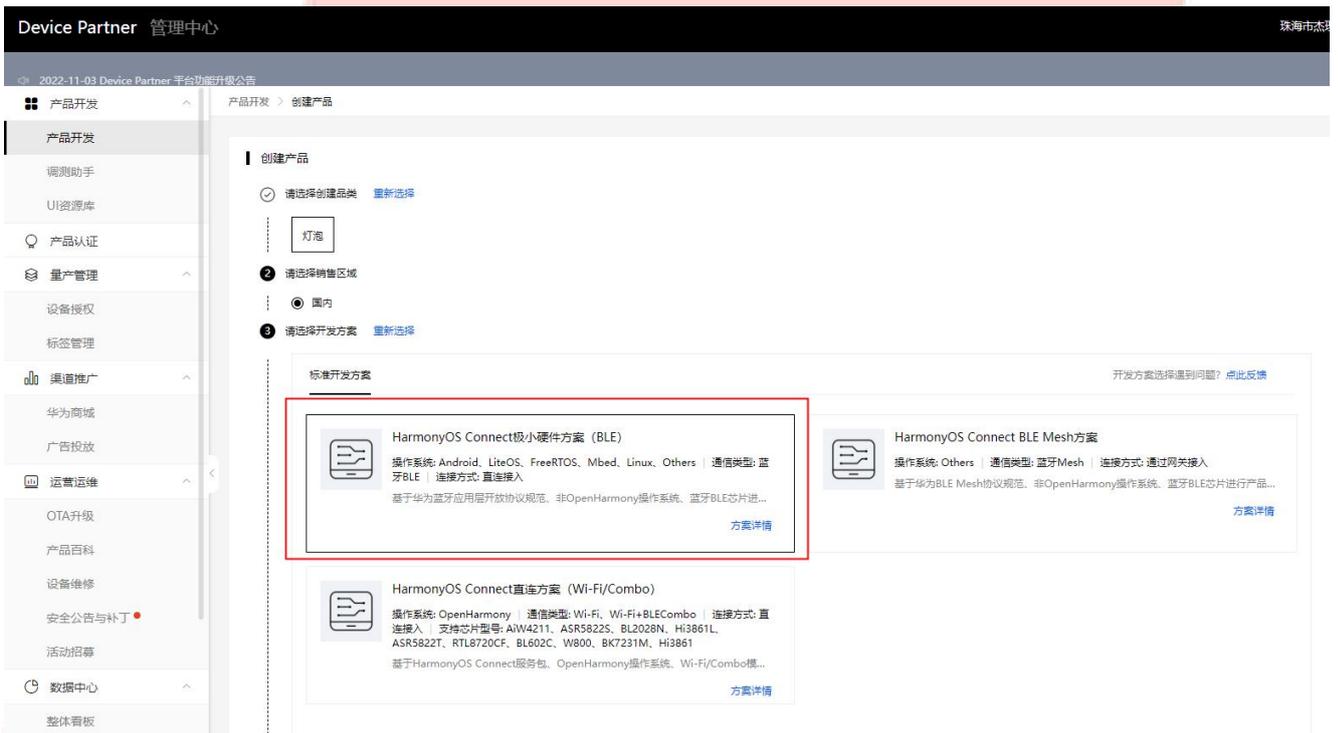


# HiLink 协议开发说明

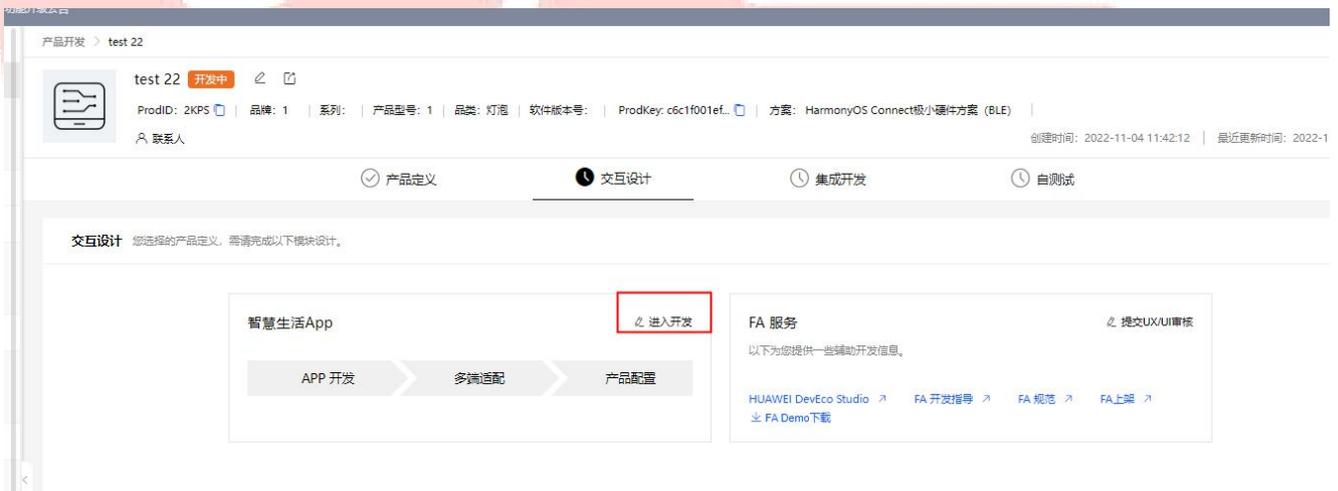
## 一、产品创建

(1) 登录华为 DP 平台 <https://devicepartner.huawei.com/console/>，申请账号后登录进入控制台。

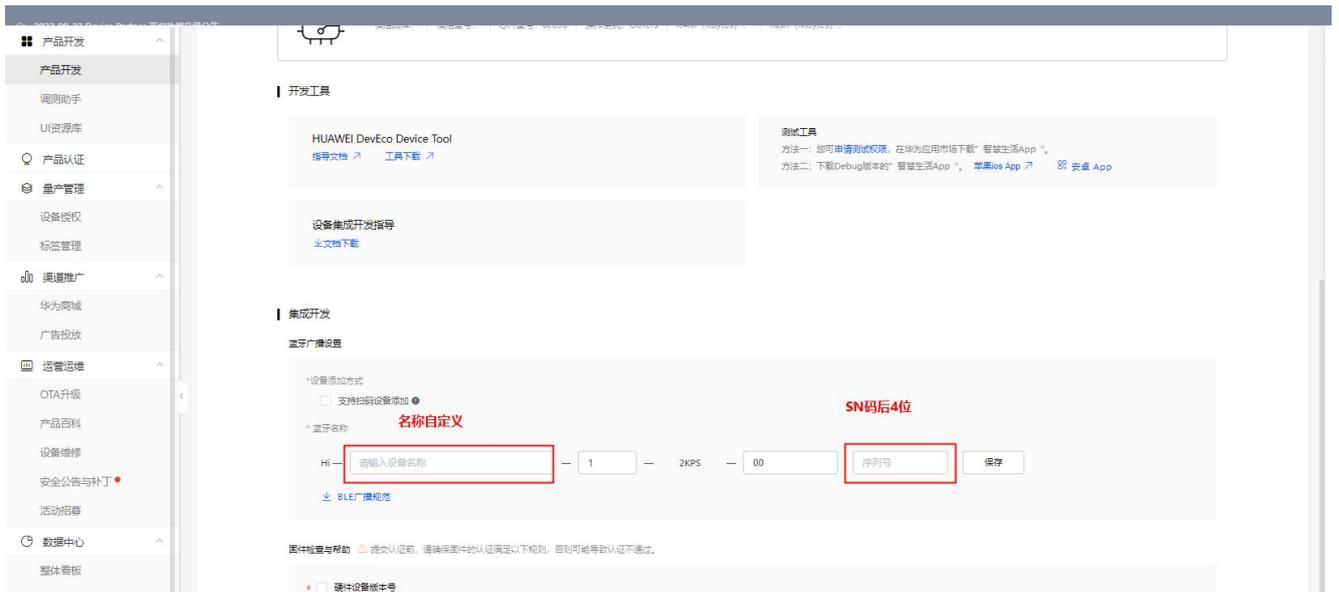
(2) 在产品开发界面创建一个产品选择 极小硬件方案并填写产品定义信息



(3) 在交互设计界面智慧生活 APP 部分开发 H5 界面

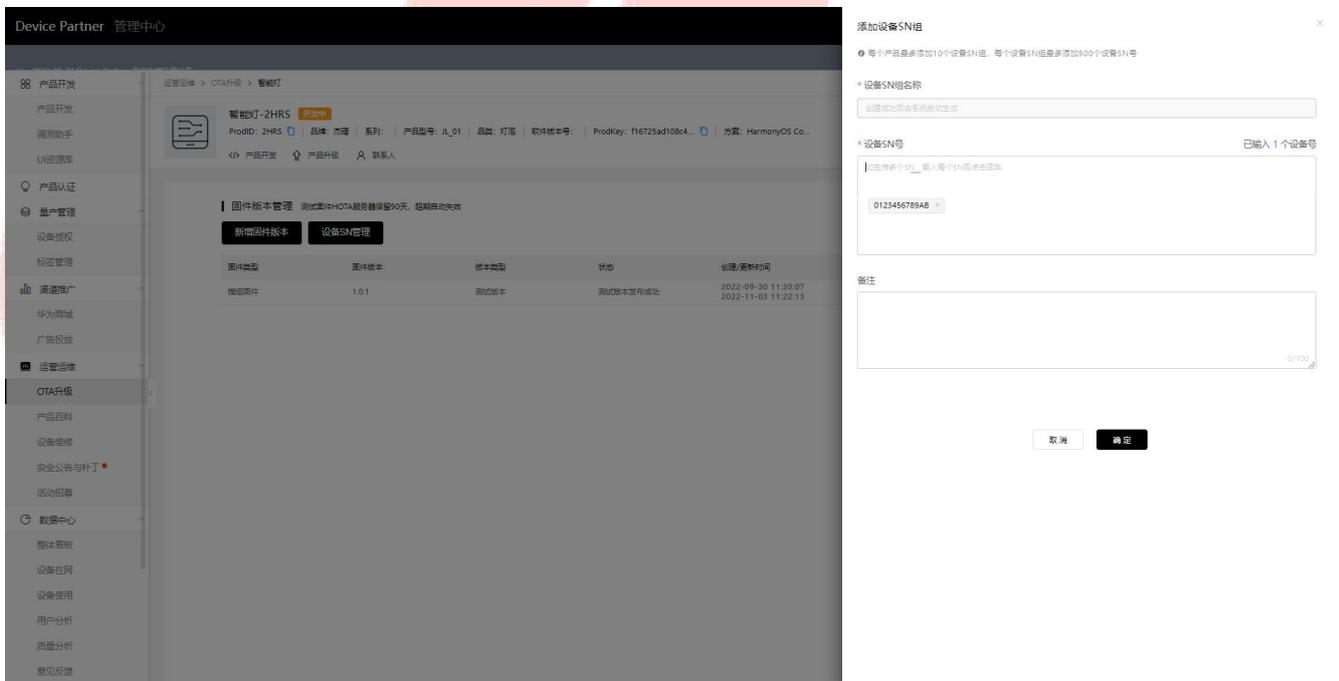


#### (4) 进入集成开发界面配置蓝牙名称

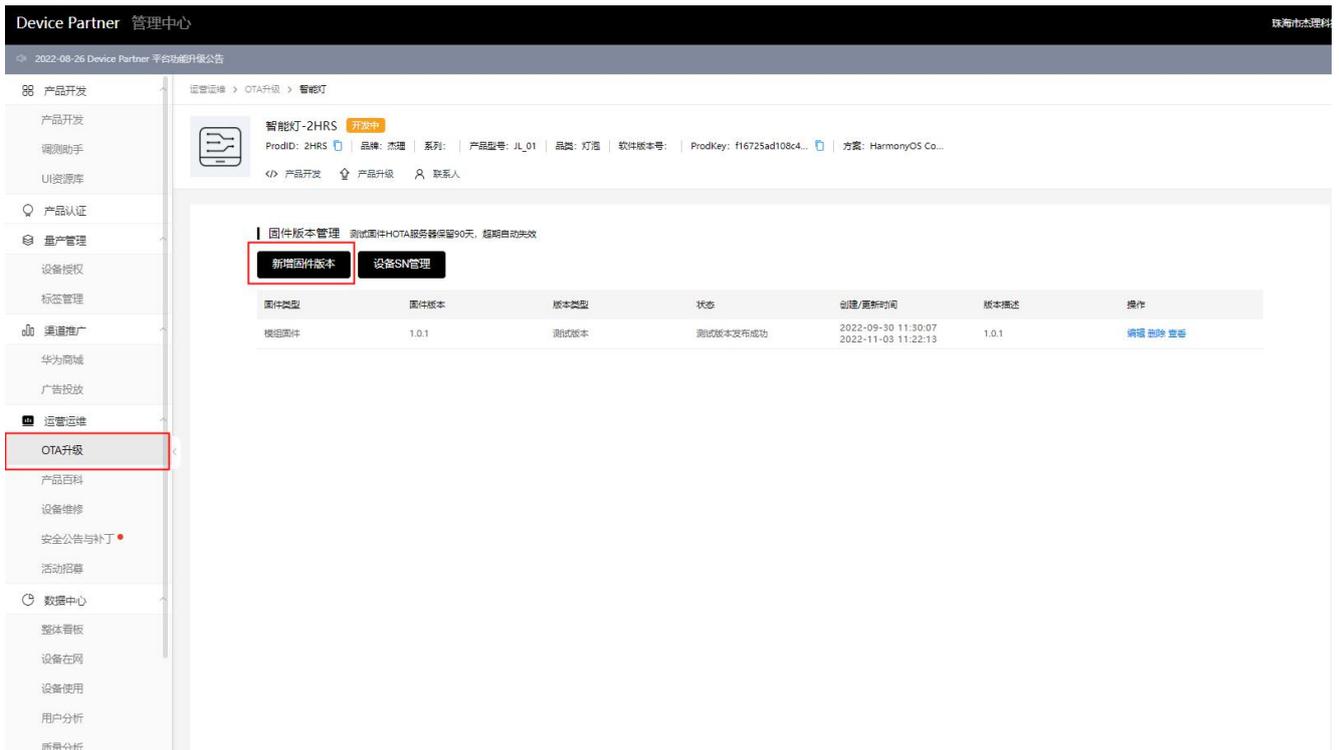


## 二、OTA 文件上传

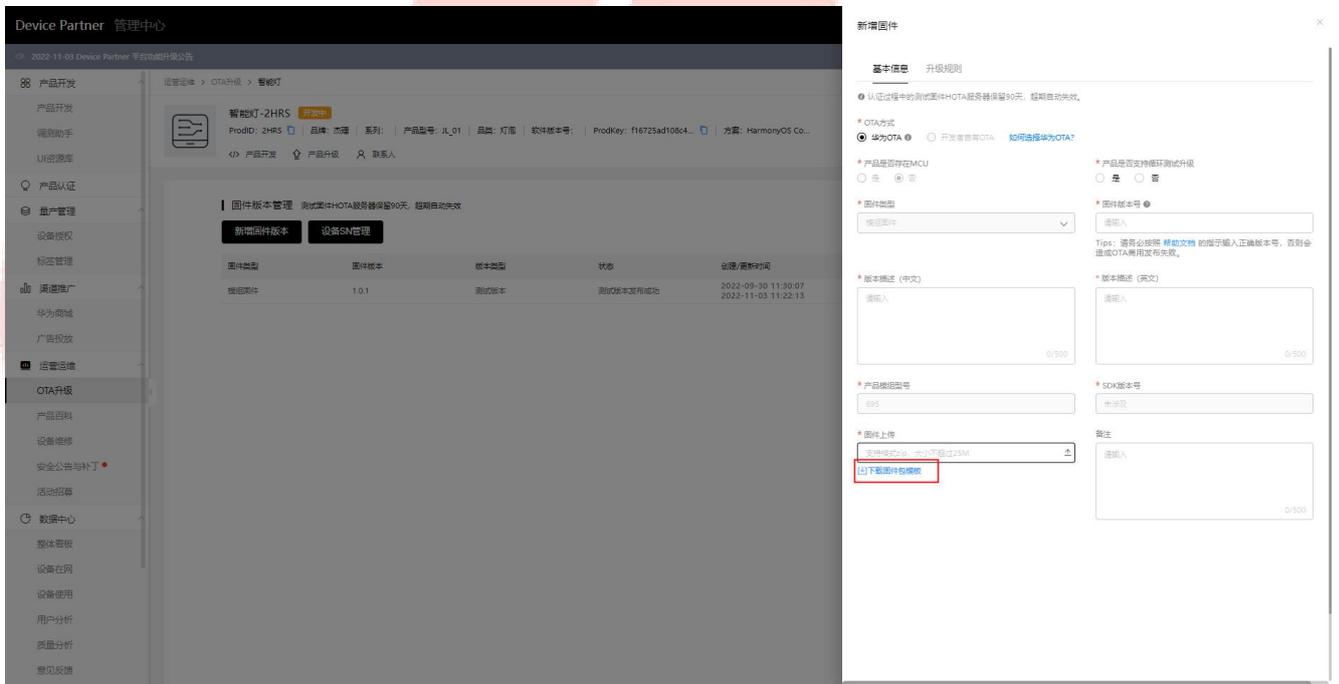
#### (1) 点击设备 SN 管理添加一个设备组并将要升级设备的 SN 码填写好点击确定



## (2) 在 OTA 升级页面新增一个固件版本



## (3) 下载固件模版包



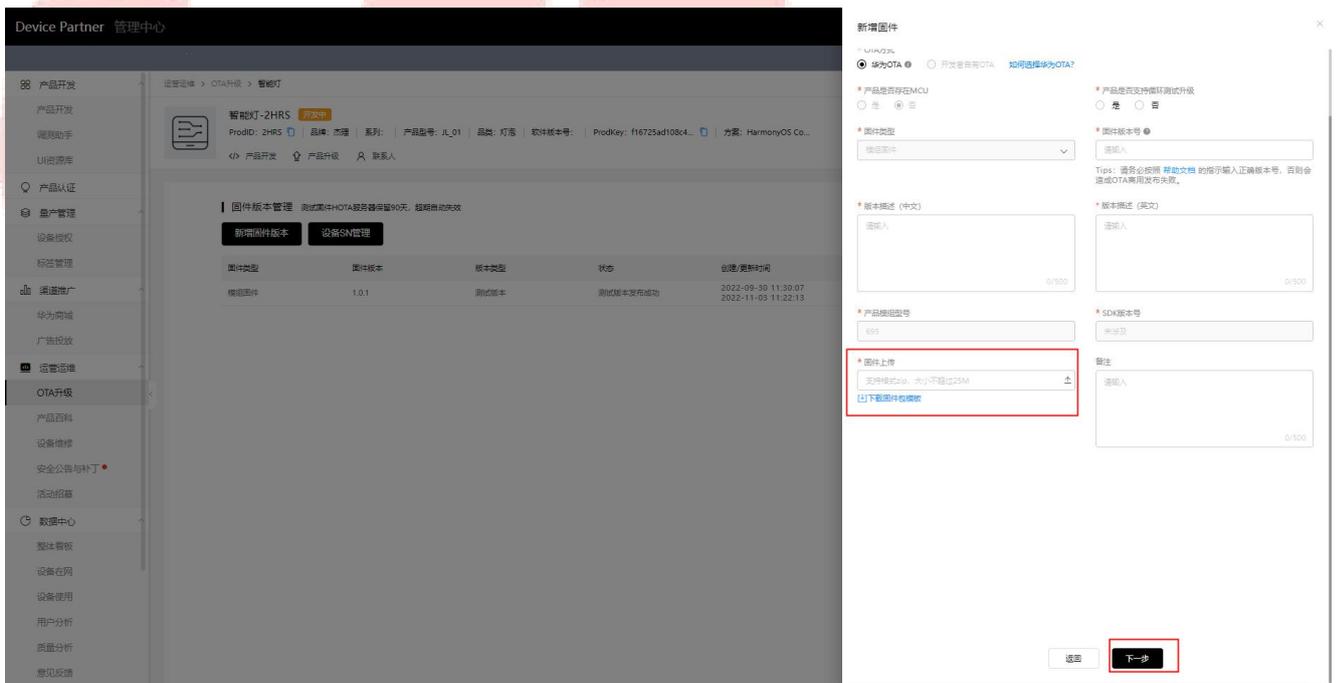
(4) 将下载目录下的 db\_update\_data.bin 使用 windows 自带的工具计算 sha256 值：  
打开 windows 的命令行，使用以下命令计算得到文件的 sha256 值

```
certutil -hashfile 文件名 SHA256
```

(5) 将模版文件夹的 image2\_all\_ota1.bin 和 image2\_all\_ota2.bin 都替换为 db\_update\_data.bin，  
并且名称都改为 image2\_all\_ota1.bin 和 image2\_all\_ota2.bin。打开模版中的 filelist.json  
文件把刚刚计算到的 sha256 值填入相应位置



(6) 将 package 文件夹压缩成 zip 格式。回到网页上传文件并且填写剩余信息后点下一步



完成固件部署后点击测试发布



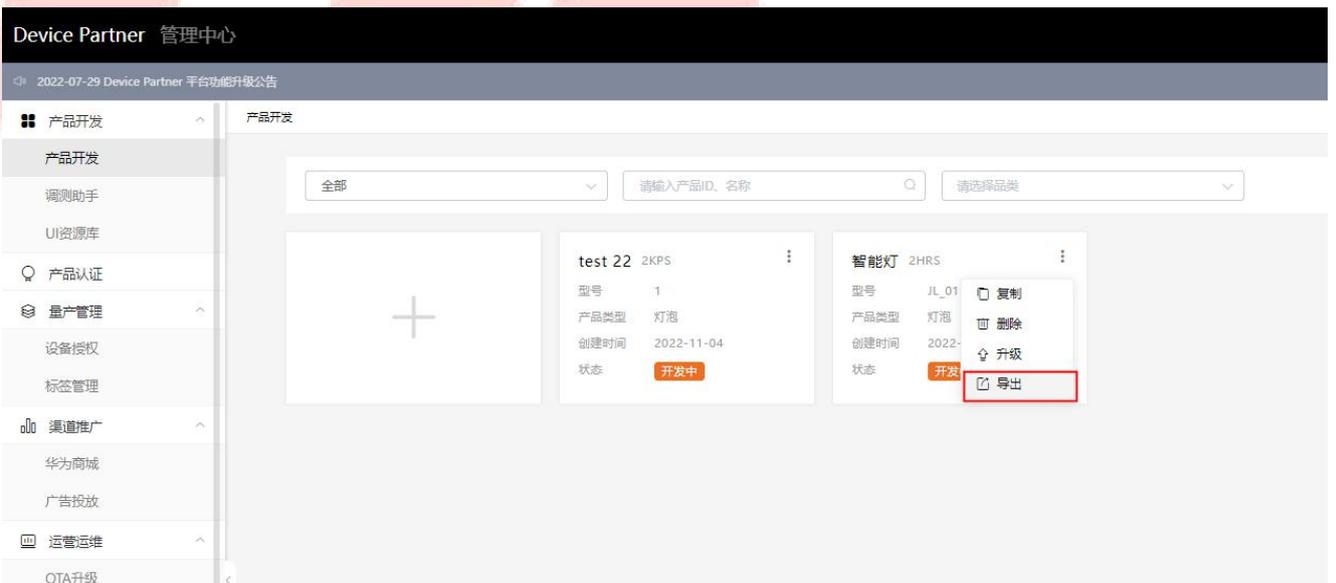
等待版本发布成功即可进行 OTA 升级



### 三、模块开发说明

#### (1) HiLink 厂商信息设置

在 DP 平台的产品开发界面导出设备信息



导出设备信息后填入到 hilink\_demo.c 文件的 hilink\_product\_info 结构体中。

结构体中的 prodID, model, dev\_t, prot\_t 分别对应导出的产品信息 json 文件中的 prodId, deviceModel, deviceTypeId, protocolType(ble 即为 4)

其他的参数自行定义

```
01hilink_demo.c
95 // 设置的hilink断连超时时间MPP字段,单位:秒
96 #define HILINK_HEARTBEAT_TIME 0x000a
97
98 uint16_t hilink_mtu = ATT_LOCAL_MTU_SIZE - 3;
99 static uint8_t hilink_ble_conn_state = 0;
100 static uint16_t hilink_timer = 0;
101 static uint8_t hilink_conn_finish = 0;
102 static uint8_t hilink_close_adv_flag = 0;
103
104 // hilink 相关信息配置
105 static dev_info_t hilink_product_info = {
106     .prodId = "2HRS",
107     .sn = "1234567890AB",
108     .dev_id = "",
109     .model = "JL_01",
110     .dev_t = "119",
111     .manu = "JL",
112     .hiv = "1.0.0",
113     .fwv = "1.0.0",
114     .hwv = "1.0.0",
115     .swv = "1.0.0",
116     .prot_t = "4",
117 };
118
```

```
//hilink私有消息
typedef struct {
    const char *prodId;          /**<设备HiLink认证号,长度范围 (0,5]*/
    const char *sn;             /**<设备唯一标识,比如sn号,长度范围 (0,40]*/
    const char *dev_id;
    const char *model;         /**<设备型号,长度范围 (0,32]*/
    const char *dev_t;         /**<设备类型,长度范围 (0,4]*/
    const char *manu;          /**<设备制造商,长度范围 (0,4]*/
    const char mac[18];        /**<设备MAC地址,字符串形式固定18字节*/
    const char *hiv;           /**<设备HiLink协议版本,长度范围 (0,32]*/
    const char *fwv;           /**<设备固件版本,长度范围 [0,64]*/
    const char *hwv;           /**<设备硬件版本,长度范围 [0,64]*/
    const char *swv;           /**<设备软件版本,长度范围 [0,64]*/
    const char *prot_t;        /**<设备协议类型,取值范围 [1,3]*/
} dev_info_t;
```

(2) 修改蓝牙 ble 名称

gap\_name\_unbound 和 gap\_name\_bound 区别在于名字开头的 Hi 和 HI。

```
124
125 // ble广播名称
126 static char gap_name_unbound[] = "Hi-jieli-12HRS0090AB";
127 static char gap_name_bound[] = "HI-jieli-12HRS0090AB";
128
```

(3) 修改广播内容

红框 1 处修改为 ProdID 的 HEX 形式，实例的 ProdID 为 2HRS，即为 0x32,0x48,0x52,0x53  
红框 2 处修改为 SN 码的后两位，实例的 SN 码后两位为 AB，即为 0x41,0x42

```
3
4 // 广播最后两位要与sn/mac最后两位相同
5 static u8 hilink_service_data[20] = {
6     0xEE, 0xFD, 0x01, 0x01, 0x07, 0x04, 0x00, 0x11, 0xE2, 0x12, 0x32, 0x48, 0x52, 0x53, 0xFF, 0x00,
7     0x15, 0x02, 0x41, 0x42
8 };
9
```

(4) 功能开发

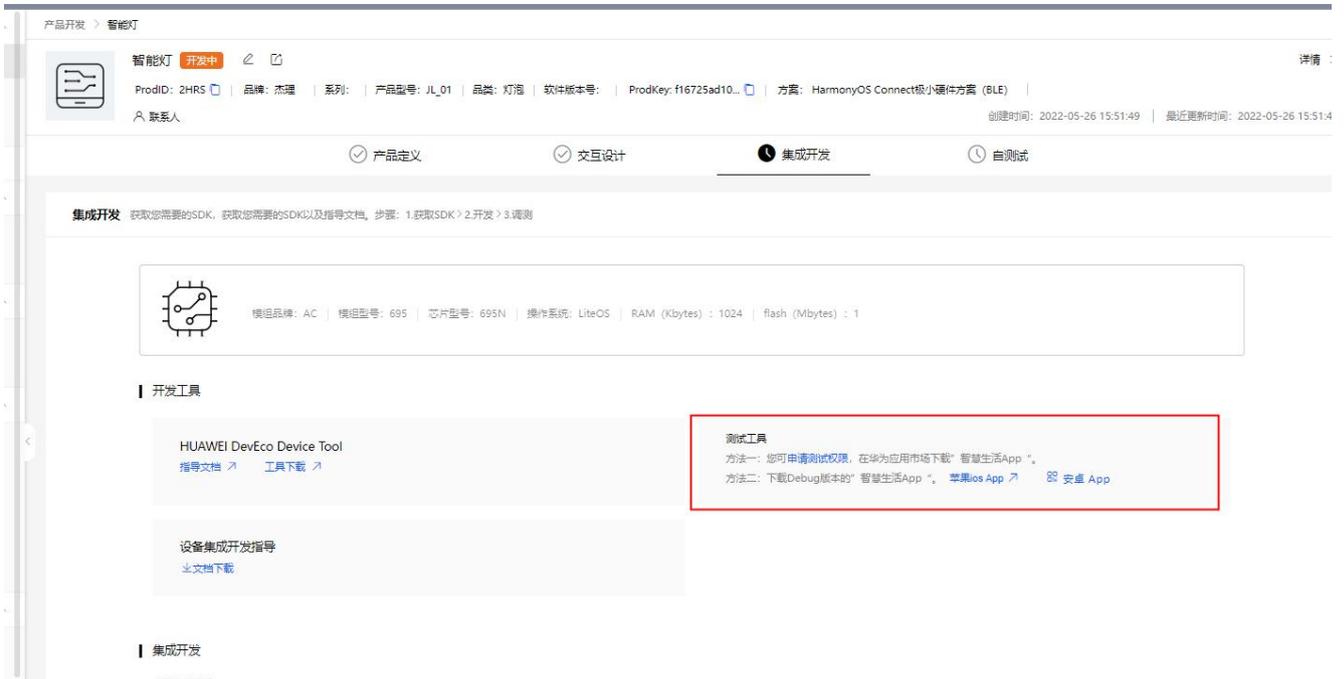
本实例已经完成配网鉴权流程与消息解析流程，用户需要在 hilink\_protocol.c 文件中的 int hilink\_cmd\_deal(uint8\_t \*buf)函数里根据申请的产品具体功能，添加功能点 sid 的判断，并根据传递下来的值去执行相应功能，并将改变后的产品状态进行上报。

例如本实例的灯控开关功能，将下发控制命令的 json 数据解析出功能点 sid 为 switch 后获取控制开关的值并将灯的开关状态做相应的改动。

```
if (memcmp(sid->valuestring, "switch", 6) == 0) {
    cJSON *state = cJSON_GetObjectItem(data, "on");
    hilink_attr.onoff = state->valueint;
    log_info("led set to %d", hilink_attr.onoff);
    hilink_led_set(hilink_attr.onoff);
    hilink_state_report();
    hilink_attr_info_store();
}
```

### 三、实际使用演示

(1) 进入产品集成开发页面下载测试用 app



(2) 打开 APP 后进入设置->关于->环境设置，选择认证沙箱后重启 app，进入设置->关于->环境设置，关闭设备升级白名单。

(3) 回到主界面点击右上角‘+’号或者添加设备来搜索连接设备





