AB53XX BLE 开发说明

Application Note

Version： V1.1

DATE: 20181116

目录

一、 SDK 打开 BLE

AB53XX SDK V060 之后才支持 BLE，其他 SDK 需要替换 BLE 专用库才能支持。

V060 及以后 SDK 使能 BLE 步骤：

1、 在 app\platform\libs 下用 libbtstack_dm.a 替换掉 libbtstack.a；'

2、 config.h 下打开 BLE 掉宏定义：

```
#define BT_SCO_FAR_DELAY_EN             0   //通话是否增加延时，来解决某
#define BT_APP_EN                       0   //是否支持手机APP控制
#define LE_EN                           1   //是否打开BLE功能
#define LE_LIGHTING_EN                  0   //是否打开BLE灯光控制服务
#define LE_MUSIC_CTRL_EN                0   //是否打开BLE音乐控制服务
```

如果需要支持 AB Link APP，请打开：

```
#define BT_APP_EN                       1   //是否支持手机APP控制
```

3、 在 Downloader 里面打开 BLE 开关：

4、 需要注意的是：

- BLE 的库可能会使程序超过 492Kbyte；
- 使能 BLE 一般需要配合蓝牙后台使用；
- 注意备份 libbtstack.a；

二、 Profile 生成

AB53XX 的 BLE profile 可以使用工具 gatt.exe 生成。解压 gatt 工具包后修改 profile.gatt 文件，双击 gatt.exe 即可生成 profile.c。profile.gatt 格式如下所示：

PRIMARY_SERVICE, Service UUID

CHARACTERISTIC, Characteristc UUID, Characteristc 权限,

第一行 PRIMARY_SERVICE 表示服务的定义，第二行表示这个服务的特性(CHARACTERISTIC)。例如下面的例子表示有两个服务，其中第二个服务有两个特性：

PRIMARY_SERVICE, FF10

CHARACTERISTIC, FF11, READ | WRITE,

PRIMARY_SERVICE, FF12

CHARACTERISTIC, FF13, READ | WRITE | DYNAMIC,

CHARACTERISTIC, FF14, READ | WRITE | DYNAMIC,

UUID 是可以随便定义的，但是不能和已有的标准 profile UUID 冲突，标准 profile UUID 可以在官网或者 bluetooth_gatt.h 查询。

更详细的 profile 说明请参考附录。

三、 接口说明

定义好 BLE profile 表后要初始化 SDK 的 BLE 属性，在 ble_init_att()函数中对 profile 定义的 Characteristic Value 进行初始化：

```
const struct att_hdl_t att_hdl_tbl[BLE_IDX_END] = {
    [0]     = {ATT_CHARACTERISTIC_FF11_01_VALUE_HANDLE,  1},
    [1]     = {ATT_CHARACTERISTIC_FF13_01_VALUE_HANDLE,  1},
    [2]     = {ATT_CHARACTERISTIC_FF14_01_VALUE_HANDLE,  1},
};
```

初始化接口：void ble_init_att_do(u8 index, u16 handle, u8 config , u8* buf, u8 len);

- Index :记录 SDK 当前 Characteristic Value 的序号，后面读写 Value 都会使用；
- handle :Characteristic Value 的 handle，定义请查询 gatt.exe 生成的 profile.c；
- config：是否打开 Client configuration；

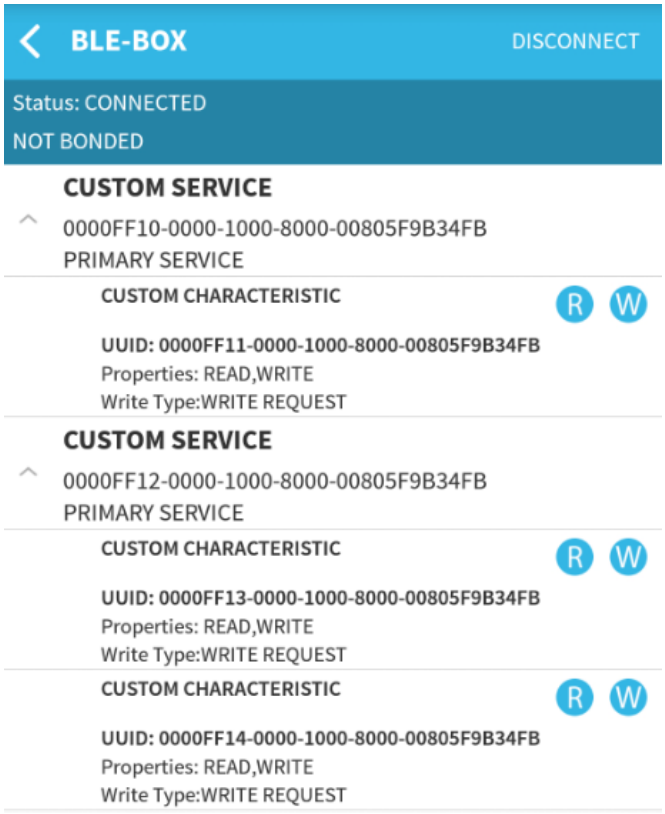
BLE Notify 接口：bool le_tx_notify(u8 index, u8* buf, u8 len);

- index：初始化时对应的 index；

四、 BLE 测试

在定义和初始化完成 profile 后，可以利用手机对 AB53XX BLE 进行测试，安卓手机请下载“BLE 调试助手”或者其他的 BLE 测试软件，IOS 请下载 lightblue。

以安卓手机为例，可以看到连接之后显示了我们定义的两个 Service，分别为 0xFF10 和 0xFF12：



五、 AB Link 获取途径

AB Link 是配合 SDK 开发的手机 APP，现在还在继续完善中。

安卓市场：

- Google Play、小米商店请搜索“AB Link”；
- 应用宝请搜索“ABLink”；
- 其他商店正在上线中；

IOS 商店：

- 正在上线中；

附录、 profile 说明

SDK 的 BLE profile 主要在 bsp_ble.c 的 profile_data 表中定义，profile_data 的数据请用 gatt.exe 生成，不需要客户单独去修改 profile_data，以下说明仅供参考。

profile_data 的数据分为 4 种 Service、Characteristic、Value 和 Client configuration，定义如下：

注意，除了特别指定“Spec 定义”的数据不能修改，其他都可以自行修改：

Service				
长度(2Byte)	权限(2Byte)	Handle(2Byte)	Primary service UUID(2Byte)	Service UUID(2Byte)

0x0a, 0x00,	0x02, 0x00,	0x01, 0x00,	0x00, 0x28,	0x0f, 0x18,

长度：0x000a；

权限：0x0002(read)；

Handle：0x0001；

Primary service UUID：0x2800（固定）；

Service UUID：0x180f；

Characteristic						
长度(2Byte)	权限(2Byte)	Handle(2Byte)	Characteristic UUID(2Byte)	Characteristic Value 权限（1Byte）	Characteristic Value Handle（1Byte）	Characteristic UUID（1Byte）
0x0d, 0x00,	0x02, 0x00,	0x02, 0x00,	0x03, 0x28,	0x12,	0x03, 0x00,	0x19, 0x2a,

长度：0x000d；

权限：0x0002(read)；

Handle：0x0002；

Characteristic UUID：0x2803（Spec 定义）；

Characteristic Value 权限：0x12；

Characteristic Value Handle：0x0003；

Characteristic UUID：0x2a19;

Characteristic Value				
长度(2Byte)	权限(2Byte)	Handle(2Byte)	Characteristic UUID(2Byte)	Value
0x08, 0x00,	0x02, 0x01,	0x03, 0x00,	0x19, 0x2a,	0Byte 表示 Dynamic, N Byte 表示 Static

长度：0x0008；

权限：0x0102(read, dynamic)；

Handle：0x0003；

Characteristic UUID：0x2a19；

Value：0Byte 表示 Value 是动态不定长的，N Byte 表示 Value 固定为 NByte；

Client Configuration			
长度(2Byte)	权限(2Byte)	Handle(2Byte)	Client Configuration UUID(2Byte)

0x08, 0x00,	0x0a, 0x01,	0x04, 0x00,	0x02, 0x29,

长度：0x0008；

权限：0x010a(read, write, dynamic)；

Handle：0x0004；

Characteristic UUID：0x2902（Spec 定义）；

权限说明：

权限	值
BROADCAST	0x01
READ	0x02
WRITE WITHOUT RESPONSE	0x04
WRITE	0x08
NOTIFY	0x10
INDICATE	0x20
DYNAMIC	0x100
UUID128	0x200

例：可读可写 = 0x02 + 0x08 = 0x0a， 可读可写 NOTIFY = 0x02 + 0x08 + 0x10= 0x1a