



OM6681 蓝牙工程配置说明

Version 1.0

Confidential

Revision history

Version/Date	Modification
V1.0: 2023/12/28 Shangfeng	- 第一版本
	-

Confidential

目录

1	概述	4
2	信息配置	5
2.1	Bluetooth Classic / BLE public 地址设置	5
2.2	Bluetooth Classic 名称设置	5
2.3	BLE 名称设置	6
2.4	BLE 随机地址设置	7
3	功能配置	8
3.1	预设 GATT Profile 使能	8
3.2	Derivation of BR/EDR link key from LE LTK 使能	9

Confidential

1 概述

本文介绍了 OM6681 如何配置蓝牙信息以及蓝牙功能

Confidential

2 信息配置

2.1 Bluetooth Classic / BLE public 地址设置

Public 地址定义在 om_bluetooth.h 中，宏名称为 **USER_DEFAULT_BLUETOOTH_ADDR**，定义如下

```
#define USER_DEFAULT_BLUETOOTH_ADDR { \
    0x01, 0xA1, 0x81, 0x66, 0xBF, 0x01, /* BLE Public Address */ \
    0x01, 0xA1, 0x81, 0x66, 0xBF, 0x01 /* BT Address */ \
}
```

需要修改 public 地址时修改该宏定义即可

注意 BLE 只有使能了“Derivation of BR/EDR link key from LE LTK”功能时才使用 public 地址，否则 BLE 广播默认使用随机地址

2.2 Bluetooth Classic 名称设置

Bluetooth Classic 名称定义在 om_bluetooth.h 中，宏名称为 **USER_DEFAULT_CLASSIC_NAME** 和 **USER_DEFAULT_CLASSIC_EIR**

```
#define USER_DEFAULT_CLASSIC_NAME    { \
    'O', 'M', '6', '6', '8', 'U', 'S', 'R', '\0' \
}
#define USER_DEFAULT_CLASSIC_EIR    { \
    0x00, /* FEC is not required */ \
    0x09, 0x09, 'O', 'M', '6', '6', '8', 'U', 'S', 'R', /* classic_name */ \
    0x02, 0x0A, 0x00, /* txpower level */ \
    0x0D, 0x03, 0x0A, 0x11, 0x0C, 0x11, 0x0E, 0x11, 0x0B, 0x11, 0x1F, 0x11, 0x1E, 0x11, /* services uuid */ \
}
```

需要更改经典蓝牙名称时同时修改这两个宏定义即可

注意在修改 **USER_DEFAULT_CLASSIC_EIR** 时要符合《BLUETOOTH CORE SPECIFICATION Vol 3, Part C 11》中定义的 Advertising and Scan Response data format，即每个 AD structure 要符合格式

Length	AD type	AD Data
--------	---------	---------

比如要将经典蓝牙名称改为“OmBT”，则需要将这两个宏定义修改为

```

#define USER_DEFAULT_CLASSIC_NAME    { \
    'O', 'm', 'B', 'T', '\0' \
}
#define USER_DEFAULT_CLASSIC_EIR     { \
    0x00, /* FEC is not required */ \
    0x05, 0x09, 'O', 'm', 'B', 'T', /* classic_name */ \
    0x02, 0x0A, 0x00, /* txpower level */ \
    0x0D, 0x03, 0x0A, 0x11, 0x0C, 0x11, 0x0E, 0x11, 0x0B, 0x11, 0x1F, 0x11, 0x1E, 0x11, /* services uuid */ \
}

```

2.3 BLE 名称设置

BLE 名称定义在 om_bluetooth.h 中，宏名称为 **USER_DEFUALT_BLE_ADV_DATA** 和 **USER_DEFUALT_BLE_GAP_NAME**

```

#define USER_DEFUALT_BLE_GAP_NAME { \
    'O', 'M', '6', '6', '8', '1', '_', 'B', 'L', 'E', '\0' \
}
#define USER_DEFUALT_BLE_ADV_DATA { \
    0x02, 0x01, 0x06, \
    0x03, 0x03, 0x12, 0x18, \
    0x0B, 0x09, 'O', 'M', '6', '6', '8', '1', '_', 'B', 'L', 'E' \
}

```

需要更改 BLE 蓝牙名称时同时修改这两个宏定义即可

注意在修改 **USER_DEFUALT_BLE_ADV_DATA** 时要符合《BLUETOOTH CORE SPECIFICATION Vol 3, Part C 11》中定义的 Advertising and Scan Response data format，即每个 AD structure 要符合格式

Length	AD type	AD Data
--------	---------	---------

比如要将 BLE 蓝牙名称改为“OmBLE”，则需要将这两个宏定义修改为

```

#define USER_DEFUALT_BLE_GAP_NAME { \
    'O', 'm', 'B', 'L', 'E', '\0' \
}
#define USER_DEFUALT_BLE_ADV_DATA { \
    0x02, 0x01, 0x06, \
    0x03, 0x03, 0x12, 0x18, \
    0x06, 0x09, 'O', 'm', 'B', 'L', 'E' \
}

```

2.4 BLE 随机地址设置

BLE 的随机地址定义在 om_bluetooth.h 中，宏名称为 **USER_DEFAULT_BLE_RANDOM_ADDR**，定义如下

```
#define USER_DEFAULT_BLE_RANDOM_ADDR { \  
    0x01, 0xA1, 0x81, 0x66, 0xBF, 0xC1 \  
}
```

需要修改随机地址时修改该宏定义即可

注意 BLE 只有未使能 “Derivation of BR/EDR link key from LE LTK” 功能时才使用随机地址，否则 BLE 广播默认使用 public 地址

3 功能配置

3.1 预设 GATT Profile 使能

工程已集成的 GATT profile 有

- HID media
- OnmicroDFU
- TSPP
- ANCS
- Device Information Service
- Battery Service

这些功能在 om_bluetooth.h 中均有对应的宏定义开关，按需要配置这些宏定义即可

```
#define OM_BLE_ENABLE_OM_HID_MEDIA 1
#define OM_BLE_ENABLE_OM_DFU 1
#define OM_BLE_ENABLE_OM_TSPP 1
#define OM_BLE_ENABLE_OM_ANCS 0
#define OM_BLE_ENABLE_OM_DIS 1
#define OM_BLE_ENABLE_OM_BATT 1
```


3.2 Derivation of BR/EDR link key from LE LTK 使能

Derivation of BR/EDR link key from LE LTK 功能用于测试一键配对方案，该方案目的是在智能手表与手机绑定的场景下，手机只弹出一次配对对话框。

该功能的宏定义在 om_bluetooth.h 中，宏名称为 **OM_BLE_ENABLE_KEY_SHARE**，定义如下

```
/* Derivation of BR/EDR link key from LE LTK
 * *BLE: if set to 1, Public address will be used in advertising.
 */
#define OM_BLE_ENABLE_KEY_SHARE    0
```

该功能的测试流程大致如下：

1. 蓝牙设备开启 BLE 链路广播（经典蓝牙处于不可见状态），广播地址类型为 public，等待手机连接
2. 与手机连接成功后，若手机类型为安卓，则不进行配对；手机类型为苹果则执行安全配对流程
3. 手机端将手机类型信息通过蓝牙设备的 TSPP 服务传递给蓝牙设备
 - a) 若为苹果手机，则向 TSPP 写入 **<apple**
 - b) 若为安卓手机，则向 TSPP 写入 **<android**
4. 蓝牙设备根据手机类型设置 Classic 蓝牙地址
 - a) 若为苹果手机，则 Classic 地址与 BLE 的 public 地址相同
 - b) 若为安卓手机，则 Classic 地址与 BLE 的 public 地址不同
5. 蓝牙设备配置 Classic 蓝牙为可发现状态，等待手机连接 Classic 蓝牙链路
6. 手机与 Classic 蓝牙链路连接后，根据需要发起 Classic 蓝牙绑定流程
 - a) 若为苹果手机，则直接使用 BLE 链路的加密信息进行加密，无需再次确认配对
 - b) 若为安卓手机，则执行正常的绑定流程

综上，与安卓手机绑定时，只有步骤 6 会弹出绑定确认框；与苹果手机绑定时，只有步骤 2 会弹出绑定对话框

注意该功能依赖于 TSPP 服务，测试该功能时需保证 TSPP 已使能