BP15 系列 SDK
蓝牙应用开发说明文档
V1.0

版本记录

版本	作者	日期	修改日志
V1.0	KK	2024-2-5	1、增加蓝牙相关版本信息说明
V0.3	KK	2024-1-25	1、完善蓝牙参数说明
V0.2	KK	2024-1-24	1、完善蓝牙通话说明 2、修改描述信息
V0.1	KK	2024-1-22	初版发布

Note：此文档适用于 BP15 的 MVsB5_BT_Audio_SDK 版本指导说明；
其他版本 SDK 中有个别差异，如有疑问请联系 FAE；

目录

蓝牙应用开发说明文档 1
1. BP15 蓝牙协议版本说明 4
2. 蓝牙参数配置 4
3. 经典蓝牙地址获取方式 4
4. 蓝牙发射功率配置 5
5. 蓝牙相关数据保存位置 5
6. 系统参量化数据说明 5
6.1. 参量化数据修改举例 1：蓝牙名修改 7
6.2. 参量化数据修改举例 2：蓝牙重连设置 8
6.3. 参量化数据修改举例 3：蓝牙后台设置 9
6.4. 参量化数据支持例表（持续更新） 9
7. 蓝牙频偏值相关说明 10
8. 蓝牙名称说明 11
9. 经典蓝牙单次连接超时时间 12
10. 连接手机超时时间 12
11. BLE 地址、广播内容修改 12
12. BLE GATT 服务修改 13
13. BLE MTU 配置 14
14. 蓝牙 Profile 定义说明 14
15. 高级 AVRCP 功能说明 14
16. 获取歌曲歌词功能说明 15
17. 通话相关参数配置 15
18. HFP 电池电量同步功能 16
19. 蓝牙连接成功、断开连接，提示音导入 16

1. BP15 蓝牙协议版本说明

1) 蓝牙核心协议版本：V5.3
2) 蓝牙 Profile 版本支持如下：

Profile	Role	Version
A2DP	source	V1.3.2
	sink	V1.3.2
AVCTP	controller	V1.4
	target	V1.4
AVDTP	source	V1.3
	sink	V1.3
AVRCP	controller	V1.6.2
	target	V1.6.2
GAVDP	initiator	V1.3
	acceptor	V1.3
HFP	AG	V1.8
	HF	V1.8
HID	N/A	V1.1.1
HSP	AG	not support
	HS	V1.2
OPP	client	V1.2.1
	server	V1.2.1
PBAP	PCE	V1.2.3
	PSE	not support
RFCOMM	N/A	V1.2
SPP	N/A	V1.2

2. 蓝牙参数配置

1) 蓝牙功能开关 CFG_APP_BT_MODE_EN 在 app_config.h 文件中；
2) 基带 EM 空间的分配参数配置在 bt_em_config.h 文件中；
3) 蓝牙应用功能的配置，蓝牙相关参数配置数据在 bt_config.h 文件中；
4) 蓝牙常用功能参数，结构体定义在 sys_param.c 文件中，参数的定义在 bt_config.h 文件中；

3. 经典蓝牙地址获取方式

1) 上电后从 flash 中指定位置读取蓝牙地址信息（最后 4K）；
2) 如 flash 无数据，从芯片的 efuse 中读取 chip_id 生成蓝牙地址；

3) 如 efuse 数据为空，则通过内置 flash 芯片的 chip_id 生成蓝牙地址，然后保存到 flash；(该方式基本不会使用，只有在前期芯片没有烧录 efuse 信息的时候才有效，后续的芯片必然会烧录 efuse，存在 chip ID 信息)

注：用户可以不遵守如上的规则，自行定义蓝牙地址；

4) 蓝牙地址的顺序说明：

a. **客户烧录的蓝牙地址的顺序是： NAP-UAP-LAP**；即保存到 flash 的顺序；

b. flash 内保存的蓝牙地址顺序： NAP-UAP-LAP，同时手机和其他外设上看到顺序都是这个顺序；

c. 系统内部蓝牙在使用时，地址的顺序为：LAP-UAP-NAP，包含蓝牙连接手机成功后，回调反馈的蓝牙顺序，都是此顺序；

注：此顺序和 flash 保存的顺序是反序的；

4. 蓝牙发射功率配置

在 bt_config.h 中，配置 BT_TX_POWER_LEVEL 参数来调整发射功率；

1) 默认参数为 level22(+6db)；

2) 发射功率最大为 level 23(+8db)；按照 2db 递减；

3) 发射功率过大，可能会导致蓝牙回连会有干扰声的产生；

5. 蓝牙相关数据保存位置

在 bt_app_ddb_info.h 文件中，数据保存位置信息，具体说明如下：

1) 蓝牙地址，BLE 地址，频偏参数蓝牙相关配置参数保存在 flash 的最后 4K 位置，即：（flash_table_info.flash_capacity - BT_CONFIG_OFFSET）；该区域受 download 保护；

2) 蓝牙的配对记录区域地址获取方式：get_bp_data_addr ()； (共 8K)

3) 蓝牙的参数配置：如蓝牙名称、BLE 名称、蓝牙回连次数等信息，保存区域地址获取方式：get_bt_config_addr()；（共 4K）

4) 清除配对记录：调用 BtDdb_Erase 函数可以清除配对记录，并不会再发起回连；

6. 系统参量化数据说明

1) 参量化数据来源于两个地方：SDK 默认值和 flash 中保存的值。

- SDK 默认值来源： sys_param.c 文件 default_parameter 变量。

```
static const SYS_PARAMETER default_parameter =
{
    .bt_LocalDeviceName     = BT_NAME,
    .ble_LocalDeviceName    = BLE_NAME,
    .bt_TxPowerLevel        = BT_TX_POWER_LEVEL,
    .bt_PagePowerLevel      = BT_PAGE_TX_POWER_LEVEL,
    .BtTrim                 = BT_DEFAULT_TRIM,
    .TwsVolSyncEnable       = TRUE, //主从之间音量控制同步
    .bt_CallinRingType      = SYS_DEFAULT_RING_TYPE,
    .bt_BackgroundType      = SYS_BT_BACKGROUND_TYPE,
    .bt_SimplePairingEnable = BT_SIMPLEPAIRING_FLAG,
    .bt_PinCode             = BT_PINCODE,
    .bt_ReconnectionEnable          = TRUE,
    .bt_ReconnectionTryCounts       = 5,
    .bt_ReconnectionInternalTime    = 3,
    .bt_BBLostReconnectionEnable    = TRUE,
    .bt_BBLostTryCounts             = 90,
    .bt_BBLostInternalTime          = 5,
    .bt_TwsReconnectionEnable       = TRUE,
    .bt_TwsReconnectionTryCounts    = 3,
    .bt_TwsReconnectionInternalTime = 3,
    .bt_TwsBBLostReconnectionEnable = TRUE,
    .bt_TwsBBLostTryCounts          = 3,
    .bt_TwsBBLostInternalTime       = 5,
    .bt_TwsConnectedWhenActiveDisconSupport = FALSE,
    .bt_TwsPairingWhenPhoneConnectedSupport = TRUE,
};
```

- Flash 中保存的值：SDK 编译阶段由 flash_param.c 生成 flash_param.bin，并且打包到 MVA 文件里面。可以通过 PC 工具 OCPWorkBench 在线和离线修改。也可以在编译阶段直接在 flash_param.c 文件对 SysDefaultParm 进行修改。

```
const FLASH_PARAMETER SysDefaultParm =
{
    .SysVer = {SYS_PARA_VER_ID,sizeof(SysDefaultParm.SysVer.name),
            "Ver 1.0.0"},  //版本

    .BtName = {BT_PARA_BT_NAME_ID,sizeof(SysDefaultParm.BtName.name),
            BT_NAME},   //蓝牙名称
    .BleName = {BT_PARA_BLE_NAME_ID,sizeof(SysDefaultParm.BleName.name),
            BLE_NAME},  //BLE蓝牙名称
    .bt_TxPowerLevel = {BT_PARA_RF_TX_LEVEL_ID,sizeof(SysDefaultParm.bt_TxPowerLevel.para_val),
            BT_TX_POWER_LEVEL},//蓝牙正常工作时发射功率
    .bt_PagePowerLevel = {BT_PARA_RF_PAGE_LEVEL_ID,sizeof(SysDefaultParm.bt_PagePowerLevel.para_val),
            BT_PAGE_TX_POWER_LEVEL},//蓝牙回连发射功率
```

2) 参量化数据使用规则：上电以后会优先读取 flash 中的参数，如果没有找到对应参数会使用 SDK 中提供的默认值。sys_param.c 文件中 sys_parameter_init 完成了该操作。sys_param.c 中定义了 FlashParamReadMap 数组，通过 ID 从 flash 中读取数据，如果读取成功会从 flash 中复制 len 个字节到指定的地方。如果不成功，会从指定默认值的地方复制 len 个字节。

```
//flash中参数读取
//从flash中读取参数成功，将参数拷贝到dest_para
//读取不成功，拷贝默认参数default_para到dest_para
static const struct
{
    SYS_PARAMETER_ID    id;             //参数ID
    void *              dest_para;      //参数地址
    void *              default_para;   //默认参数地址
    uint8_t             len;            //参数长度
}FlashParamReadMap[] =
{
    {BT_PARA_BT_NAME_ID,            (void *)sys_parameter.bt_LocalDeviceName,       (void *)default_parameter.bt_LocalDeviceName,       BT_NAME_SIZE},
    {BT_PARA_BLE_NAME_ID,           (void *)sys_parameter.ble_LocalDeviceName,      (void *)default_parameter.ble_LocalDeviceName,      BLE_NAME_SIZE},
    {BT_PARA_RF_TX_LEVEL_ID,        (void *)&sys_parameter.bt_TxPowerLevel,         (void *)&default_parameter.bt_TxPowerLevel,         1},
    {BT_PARA_RF_PAGE_LEVEL_ID,      (void *)&sys_parameter.bt_PagePowerLevel,       (void *)&default_parameter.bt_PagePowerLevel,       1},
    {BT_PARA_TRIM_VAL_ID,           (void *)&sys_parameter.BtTrim,                  (void *)&default_parameter.BtTrim,                  1},
    {BT_PARA_CallinRingType_ID,     (void *)&sys_parameter.bt_CallinRingType,       (void *)&default_parameter.bt_CallinRingType,       1},
    {BT_PARA_BackgroundType_ID,     (void *)&sys_parameter.bt_BackgroundType,       (void *)&default_parameter.bt_BackgroundType,       1},
    {BT_PARA_SimplePairingEnable_ID,(void *)&sys_parameter.bt_SimplePairingEnable,  (void *)&default_parameter.bt_SimplePairingEnable,  1},
    {BT_PARA_PinCode_ID,            (void *)sys_parameter.bt_PinCode,               (void *)default_parameter.bt_PinCode,               BT_PIN_CODE_LEN},
    {BT_PARA_ReconnectionEnable_ID, (void *)&sys_parameter.bt_ReconnectionEnable,   (void *)&default_parameter.bt_ReconnectionEnable,   6},

    {TWS_PARA_TWS_VOL_SYNC_ID,      (void *)&sys_parameter.TwsVolSyncEnable,        (void *)&default_parameter.TwsVolSyncEnable,        1},
    {TWS_PARA_ReconnectionEnable_ID,(void *)&sys_parameter.bt_TwsReconnectionEnable,(void *)&default_parameter.bt_TwsReconnectionEnable,6},
    {TWS_PARA_TwsPairingWhenPhoneConnectedSupport_ID,(void *)&sys_parameter.bt_TwsPairingWhenPhoneConnectedSupport,(void *)&default_parameter.bt_TwsPairingWhenI
    {TWS_PARA_TwsConnectedWhenActiveDisconSupport_ID,(void *)&sys_parameter.bt_TwsConnectedWhenActiveDisconSupport,(void *)&default_parameter.bt_TwsConnectedWhe
};
```

3) 增加自己的参量化数据。

- flash_param.h 文件中 SYS_PARAMETER_ID 增加一个自定义参数的参数 ID，在 FLASH_PARAMETER 中增加参数的结构体类型（包含 ID+长度+参数值）。
- flash_param.c 文件中 SysDefaultParm，对这个参数赋默认值。
- sys_param.c 读取参数值，并且对参数做合法判断，非法的时候设置一个默认值。

6.1. 参量化数据修改举例 1：蓝牙名修改

1) 离线修改：PC 运行 OCPWorkBench_V1.0.4(2023.10.28)软件。
   a. 导入 MVA 文件。
   b. 修改蓝牙名称。
   c. 保存为新的 MVA 文件。
   d. 用新的 MVA 文件下载/烧录到目标板上既可以生效。

2) 在线修改：PC 用 USB 线连接目标设备，然后运行 OCPWorkBench_V1.0.4(2023.10.28)软件。
   a. 点击 Receive_data，读取设备中的参量化数据。
   b. 修改蓝牙名称。
   c. 点击 Send_data，写入数据到设备中。
   d. 重新断电上电设备既可以生效。

6.2. 参量化数据修改举例 2：蓝牙重连设置

1) 参考 6，蓝牙名称的修改步骤。同样支持在线/离线修改 2 种方式。
2) 参数说明（参考 SYS_PARAMETER 定义的结构体顺序）：

```
[1, 5, 3, 1, 90, 5]
参数 1 BT 自动重连(开机或者切换模式)----- 1  开启
参数 2  自动重连尝试次数  ----- 5 次
参数 3  自动重连每两次间隔时间(in seconds) -----  间隔 3S
参数 4 BB Lost 之后自动重连  1->  打开/0->关闭  ----- 1  打开
参数 5 BB Lost 尝试重连次数  ----- 90 次
参数 6 BB Lost 重连每两次间隔时间(in seconds) -----  间隔 5S
```

3) 点击参数修改的时候，在工具下方也会有对应的参数说明。

6.3. 参量化数据修改举例 3：蓝牙后台设置

1) 参考 6，蓝牙名称的修改步骤。同样支持在线/离线修改 2 种方式。
2) 参数说明：

BT 后台设置 0 -> BT 后台不能连接手机 1 -> BT 后台可以连接手机 2 -> 无后台

6.4. 参量化数据支持例表（持续更新）

ID	长度	默认参数	范围	说明
0x00	40	BP15_BT		字符串，蓝牙名称
0x01	40	BP15_BLE		字符串，BLE 蓝牙名称
0x02	2	23	0-23	蓝牙正常工作时发射功率
0x03	2	16	0-23	蓝牙回连发射功率
0x04	2	20	0-31	频偏 trim 值
0x05	2	2	0-3	bt 铃声设置 0 -> 不支持来电铃声 1 -> 来电报号和铃声 2 -> 使用手机铃声，若没有则播本地铃声 3 -> 强制使用本地铃声

0x06	2	0	0-2	BT 后台设置 0 -> BT 后台不能连接手机 1 -> BT 后台可以连接手机 2 -> 无后台
0x07	2	1	0-1	简易配对开启/关闭 0 -> 关闭 1 -> 开启
0x08	8	0000		字符串，配对 pin code
0x09	6	1,5,3,1,90,5		蓝牙回连设置 参数 1 BT 自动重连(开机或者切换模式) --- 1 开启 参数 2 自动重连尝试次数 --- 5 次 参数 3 自动重连每两次间隔时间(in seconds) --- 间隔 3S 参数 4 BB Lost 之后自动重连 1-> 打开/0->关闭 --- 1 打开 参数 5 BB Lost 尝试重连次数 --- 90 次 参数 6 BB Lost 重连每两次间隔时间(in seconds) --- 间隔 5S
0x200	40	Ver 1.0.0		参量化数据版本
0x1000				用户自定义数据

7. 蓝牙频偏值相关说明

1) 上电时，系统会从 flash 的指定区域读取频偏值，并使用；如果 flash 中没有保存频偏参数，则会使用默认频偏值 BT_DEFAULT_TRIM (bt_config.h)；
2) 可以通过串口日志信息观察当前板级的频偏值，蓝牙测试盒(v0.2.7 之后的固件版本)上也会显示校准后的参数；
串口日志表现如下：Freq trim:0x13
3) 在使用蓝牙测试盒的时候，可以根据实际情况选择操作：
   a. 频偏校准：将测试盒档位打到校准频偏模式(000)，在测试盒连接成功被测设备后，会自动将频偏值校准，并保存到 flash 中；测试盒上显示校准后的频率偏差值；
   b. 不需要频偏校准：将测试盒档位达到不校准频偏模式(001)，在测试盒连接成功被测设备后，不校准频偏，只是将当前被测设备的频率偏差值显示在屏幕上；
   c. 在频偏校准完成后，蓝牙测试盒上会显示校准后的频偏值；如下图，0x13 是校准值；4Khz 为校准后的偏差频率；

d. 若生产环节不需要每台机器做频偏校准，可使用试产的产品多个使用测试盒进行频率的校准后，确认测试盒显示的校准值是否保持接近，然后将频偏值写到 SDK 中；

e. 如果蓝牙测试盒的频偏校正后面未显示参数，而是“---”，则说明频偏值超过一定范围，测试盒无法校准；

解决方案：

a. 更换晶体；

b. 加大晶体谐振电流值；

例如：在 SystemClockInit() 函数内，系统默认是 Clock_HOSCCurrentSet(9); //晶体谐振电流配置为 9.

4) 目前蓝牙测试盒是支持 BT 和 BLE 进行频偏的校准；（固件版本：v0.2.11 之后）

a. 基于经典蓝牙进行频偏校准，需要开启 A2DP 协议；同时兼顾经典蓝牙的相关功能测试。

b. 基于 BLE 进行频偏校准，测试盒扫描特定的名称，需要按照测试盒的规范进行广播：（广播名称：MV_BLE）

8. 蓝牙名称说明

1) 在 bt_config.h 头文件中配置蓝牙名称 BT_NAME

2) 中文名称，可以直接在 bt_name.h 中，定义中文名称，但是需要在 bt_config.h 中，屏蔽 BT_NAME 定义，并打开 bt_name.h 的调用；

```
35
36 /*****************************************************************************
37  *
38  * 蓝牙名称注意事项:
39  * 1.蓝牙名称支持中文,需要使用URL编码
40  * 2.BLE的名称修改在ble广播数据中体现(ble_app_func.c)
41  *****************************************************************************/
42 #include "bt_name.h"
43 //#define BT_NAME                       "BP15_BT"
44 #define BLE_NAME                    "BP15_BLE"
```

注：bt_name.h 中中文名称的修改，需要保证修改的文件的格式为 UTF-8，避免由于文本格式错误，导致修改异常；

3) 如蓝牙名称为中文，或者其他语言，请将内容进行 URL 编码，再写入 bt_LocalDeviceName；

9. 经典蓝牙单次连接超时时间

在 bt_config.h 中，BT_LSTO_DFT 可以配置蓝牙连接成功后，通讯异常，超时断开的时间；只有在 BP10 和手机连接时，角色为 Master 的情况下才有用；
注意：配置参数和实际参数的转换关系；

10.连接手机超时时间

在 bt_config.h 内，BT_PAGE_TIMEOUT 可以设置每次 page 的超时时间；默认时间：5s

11.BLE 地址、广播内容修改

1) BLE 的地址获取方式：先从 flash 的指定位置读取；若无保存的地址信息，则通过 BT 的地址来生成，并保存到 flash；
2) BLE 广播内容：
遵循 BLE 广播规范 LTV 数据格式：length + type + data；

具体修改如下：
1) SDK 默认名称修改可在 bt_config.h 文件夹如下：

```
#define BLE_NAME                "BP15_BLE"
#define BLE_DFLT_DEVICE_NAME    (sys_parameter.ble_LocalDeviceName) //BLE 名称
```

2) 可直接修改 0x09 后的内容(注意修改后需要同时更改长度)

```
uint8_t ble_app_adv_data[30] = {
      //length + type + data
    // Flags general discoverable, BR/EDR not supported
    2, 0x01, 0x06,
    // Name
    9, 0x09, 'B','P','1','0','-','B','L','E',
};
```

3) 默认初始化配置接口代码如下：

```
le_user_config.adv_data.adv_data = (uint8_t *)ble_app_adv_data;
le_user_config.adv_data.adv_len = adv_len;

// 广播回复数据需要时填写,不需要时填 NULL
```

```
le_user_config.rsp_data.adv_rsp_data = (uint8_t *)ble_app_rsp_adv_data;
le_user_config.rsp_data.adv_rsp_len = rsp_adv_len;
```

其中两个 BUFF 加起来总共可扩充 31+31 =62 个广播字节长度；

4) 动态广播数据更新 API，进行动态更新广播内容。

```
void app_set_adv_data(uint8_t *data, uint16_t len); //填充普通广播数据
void app_set_scan_rsp_data(uint8_t *data, uint16_t len); //塞入广播回复内容
void app_start_advertising(void); //使能广播
```

12.BLE GATT 服务修改

1) 16bitUUID 配置

2) 16bitUUID 需填充 ble_gatt_att16_desc_t 结构体模型，参考配置如下：

```
static ble_gatt_att16_desc_t att16_db[UDSS_IDX_NB] = {
  // ATT UUID
  // | Permission | EXT PERM | MAX ATT SIZE
  // User Data Service Service Declaration
  [0] = {GATT_DECL_PRIMARY_SERVICE, RD_P, 0}, // 0x2800

// Characteristic1
//  DataBase Index Increment Characteristic Declaration
[1] = {GATT_DECL_CHARACTERISTIC, WR_P | NTF_P,
       0},  // 0x2803
            //  DataBase Index Increment Characteristic Value
[2] = {0xff11, WC_P, LE_VAL_MAX_LEN},

// Characteristic2
//  DataBase Index Increment Characteristic Declaration
[3] = {GATT_DECL_CHARACTERISTIC, NTF_P,
       0},  // 0x2803
            //  DataBase Index Increment Characteristic Value
[4] = {0xff12, NTF_P, LE_VAL_MAX_LEN},
// Client Characteristic Configuration Descriptor
[5] = {GATT_DESC_CLIENT_CHAR_CFG, RD_P | WR_P, OPT(NO_OFFSET)},  // RW

// Characteristic3
//  DataBase Index Increment Characteristic Declaration
[6] = {GATT_DECL_CHARACTERISTIC, NTF_P | WR_P,
       0},  // 0x2803
            //  DataBase Index Increment Characteristic Value
[7] = {0xff13, IND_P, LE_VAL_MAX_LEN},
//          // Client Characteristic Configuration Descriptor
[8] = {GATT_DESC_CLIENT_CHAR_CFG, RD_P | WR_P, OPT(NO_OFFSET)},  // RW

// Characteristic4
//  DataBase Index Increment Characteristic Declaration
```

```
[9] = {GATT_DECL_CHARACTERISTIC, WR_P,
       0},  // 0x2803
            //  DataBase Index Increment Characteristic Value
[10] = {0xff14, WR_P, LE_VAL_MAX_LEN},

};
```

注意事项：
a. 横排配置遵循 16BIT_UUID+属性权限+扩展信息（限定属性发送长度）
b. 竖排遵循服务+服务配置或特性+特性内容配置；

3) 16BIT-UUID：填充好描述服务后将其配置给如下变量：

```
le_user_config.profile_uuid16
le_user_config.ble_uuid16_service
```

4) 128BIT-UUID：同配置给如下变量：

```
le_user_config.profile_uuid128
le_user_config.ble_uuid128_service
```

13.BLE MTU 配置

在 ble_app_func.c 中，BLE 初始化函数：LeInitConfigParams ()内，
le_user_config.att_default_mtu = DEFAULT_MTU_SIZE;（默认大小：250 字节）

14.蓝牙 Profile 定义说明

3) BT_A2DP_SUPPORT 宏定义开关，包含了 A2DP 和 AVRCP 2 个协议的开关，由于考虑到 SDK 的使用情况，默认都是支持蓝牙音乐播放，所以必须要开启 A2DP；
4) BLE、HFP、SPP 根据实际的需要进行开启和关闭；
5) 支持不同的 profile，使用的 ram 资源是有区别的，请查看 bt_stack_memory.h 文件中 BT_STACK_MEM_SIZE 的相关说明；

15.高级 AVRCP 功能说明

默认开启高级 AVRCP 功能：BT_AVRCP_ADVANCED
1) 通过高级 AVRCP 功能能及时更新当前的播放状态；
2) 开启 BT_AVRCP_VOLUME_SYNC，支持蓝牙音量同步功能，主要适用于苹果手机的播放音乐音量同步；
3) 开启 BT_AVRCP_SONG_PLAY_STATE，实时的刷新歌曲的播放时间；
4) 开启 BT_AVRCP_SONG_TRACK_INFOR，在歌曲开始播放、歌曲切换，获取歌曲的 ID3

信息；

16.获取歌曲歌词功能说明

手机部分音乐播放 APP(酷狗)支持歌曲歌词的获取，代码层面需要修改如下：

1) 在 bt_config.h 文件中，需要开启如下宏定义：

```
BT_AVRCP_PLAYER_SETTING          ENABLE
BT_AVRCP_SONG_PLAY_STATE         ENABLE
BT_AVRCP_SONG_TRACK_INFOR        ENABLE
```

2) 调用函数 BTCtrlGetMediaInfor 来获取歌曲信息；

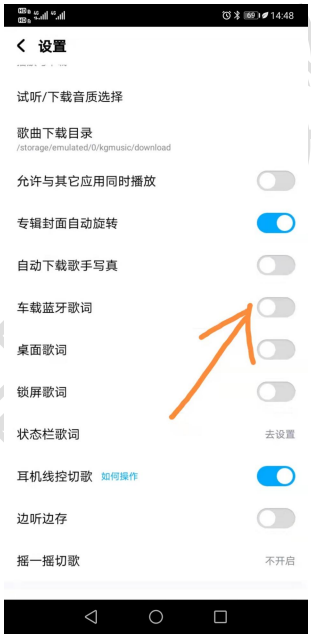
然后通过回调事件 BT_STACK_EVENT_AVRCP_ADV_MEDIA_INFO 反馈歌曲的信息；

3) 歌词信息会在 Title of the media 中显示；

4) 由于歌曲信息是实时刷新的，所以在应用层，需要设定间隔时间，定时来获取(例如 1s)；

5) 每次只能获取到 1 行歌词，此功能是由手机 APP 来决定；

6) 手机端 APP 需要开启 车载蓝牙歌词 功能，可以参考下图：



7) 在使用时，需要将歌词信息保存下来，不要直接打印，转到其他任务中再次处理，否则可能会导致蓝牙任务阻塞而出现异常；

17.通话相关参数配置

1) 参数配置在 bt_config.h 中：

MIC 增益根据 MIC 的实际的灵敏度来进行调整：

BT_HFP_MIC_PGA_GAIN：默认值为 level 15(1.5db 左右)

BT_HFP_MIC_DIGIT_GAIN：默认值为 4095(0db)

BT_HFP_INPUT_DIGIT_GAIN：默认值为 4095(0db)，用于调小 DAC 输出的通话声音大小

2) AEC 音效的参数配置，都集成在了音效框图内
3) SDK 默认关闭手机的 AEC 功能，即开启宏定义：BT_REMOTE_AEC_DISABLE

18.HFP 电池电量同步功能

在 bt_config.h 文件内，BT_HFP_BATTERY_SYNC 宏定义开关用于开启电池电量同步功能。
1) 在 HFP 协议连接成功后，自动同步当前设备的电量到手机端显示；
2) 需要配合 CFG_FUNC_POWER_MONITOR_EN 功能一起使用；

19.蓝牙连接成功、断开连接，提示音导入

蓝牙协议栈会在蓝牙连接成功和断开连接的时候，将 MSG_BT_STATE_CONNECTED 和 MSG_BT_STATE_DISCONNECT 2 个消息发送到 main_task.c 中，可在消息处理中加入相关的提示音；

在使用过程中，如有疑问请联系 FAE。