

蓝牙协议栈应用 **FAQ**

V1.0

1. 蓝牙协议栈应用 FAQ

Q: ***** 什 么 叫 协 议 栈 初 始 化 成 功

A: 协议栈初始化成功包括蓝牙 RF 初始化成功和协议栈初始化成功两个，需要两个初始化均成功蓝牙功能才能正常使用。

需要 log 打印出如下的信息才表示一切 OK，否则基本上就是 RF 初始化不成功。

"&&&&&&& WaitForPairing/Connection &&&&&&&"

Q: ***** 蓝 牙 回 连 异 常 ， 增 加 调 试 信 息 的 方 法

A: 1) 如果客户发现蓝牙设备开机回连异常，可在协议栈初始化前调用函数 EnableBtDbgInfo () 这样就会有更多的 log 给出，以方便问题原因的判断。

Q: ***** 蓝 牙 通 话 过 程 中 如 何 主 动 、 被 动 改 变 音 箱 音 量

A: 1) 蓝牙通话过程中，通话的音量需要根据当前协议栈传递的蓝牙音量值同步调整本地音频输出音量；

2) 蓝牙协议栈传递的音量取值范围为 0-15,0 表示 mute，15 表示最高音量，蓝牙音箱需要换算到实本地对应的音量；

3) 蓝牙音箱端试图调整音量需要调用函数 BTPhoneCallVolSyncSet (。。。) 同时修改本地音量的输出，手机端会有对应显示，但不会实际改变送出的信号；

4) 如果用户在手机端操作音量按键，通常会接收到事件【 BT_UI_CB_STATUS_HF_PHONE_SPEAKER_VOL 】，此时调用函数 BtGetRemoteSpeakerVol () 可获取当前被设置的音量值；

5) 接收到事件【BT_UI_CB_STATUS_HF_PHONE_SPEAKER_VOL】后，需要本地根据换算的结果调整本地音量。

6) 蓝牙通话的音量和播放音乐时的音量控制是不一样的，且不得交叉混用。

7) 实际应用时每次通话前可以将本地音量的值初始化为最大，如果接收到手机端控制事件再同步修正。

Q: ***** 电 话 号 码 本 功 能 使 用

A: 1) 电话号码簿功能的使用，请参考文档【PhoneBook 使用说明.doc】，SDK 中缺省不包含具体实现的代码，需要客户自行按需实现。

Q: *****AVRCP 获 取 ID3 信 息

A: 1) AVRCP 的 ID3 信息基本上只有苹果的设备支持，且苹果的 itunes 的播放器支持更完整，第三方播放器支持有可能支持部分功能。

2) 具体使用请参考文档【AVRCP 使用说明.doc】

Q: ***** 蓝牙非关闭电源的动态开关使用方法补充说明

A: 1) SDK 中如果打开宏 **【FAST_POWER_ON_OFF_BT】**，那么就会直接调用 **FastEnableDisableBT** (。。。) 进行快速开关蓝牙。

该函数实现的功能是：在非蓝牙模式下关闭蓝牙，蓝牙模式下开启蓝牙并重连。

2) 用户如果考虑其他的开关方式，那么可以调用下面的函数，需要用户做好相关的模式切换的状态保护（可参考函数 **FastEnableDisableBT** 的实现）。

FastDisableBT () 关闭蓝牙

FastEnableBT (reconnet_flag) 开启蓝牙，参数决定是否需要重连。

Q: ***** 蓝牙通过动态打开、关闭 RF 供电电源使用方法说明

A: 1) 在 SDK 中打开宏 **【HW_POWER_ON_OFF_BT】**，那么 SDK 会缺省直接调用 **void HwPowerOnOffBT(uint32_t CurModuleID);**

2) 如果使用硬件开关，那么需要用户按照实际板子情况实现对应的上下电的函数，如下面的两个（在文件 **bt_ui_callback.c** 中）

UserBtRfPowerPlugIn ()

UserBtRfPowerPlugOut ()

3) 用户需要保证宏 **【HW_POWER_ON_OFF_BT】** 与 **【FAST_POWER_ON_OFF_BT】** 的互斥唯一性。

Q: ***** 蓝牙地址及名称的确定

A: 1) 蓝牙地址的生成方式（标准 SDK 内的实现方式，用户可以根据需要调整），根据下面的顺序获取，直到获取成功：

a) 从 **const data** 区获取地址

b) OTP 内指定地址获取地址

c) 芯片序列号获取地址

d) 代码中指定的缺省地址 ***** 这个地址基本无效，因为芯片都是有序列号的。

注意：用户可以不遵守上述规则，直接知道某个地址。

2) 蓝牙名称的生成方式，标准 SDK 中根据下面的顺序获取，指导获取成功：

a) 最终用户自定义名称 ***** 该名称可以由用户在手机上通过蓝牙发送一个含有特定文件名的文件的方式，SDK 中已经做了处理。

该特定文件可以是空文件。具体实现由函数 **CheckValidatedBTUDIF** () 完成。

文件名的规则为： **@#**用户自定义的任意文字**@**

b) 从 **const data** 区获取

c) 代码中指定的缺省名称

Q: ***** 蓝牙相关变量、常量存取区域

A: 1) 蓝牙常量----蓝牙地址, 如果 **const** 数据区包含蓝牙地址信息, 那么就会在 **const** 区域保存在字段【LYDZ】中。

2) 蓝牙变量----蓝牙配对信息等, 存储在 **userdata** 区域, 区域【固定】占用【64KB】大小, 需确保 **flash** 对应区域有效, 且可以读、写、擦除等。

3) **const** 区域起始地址在文件 **startup.c** 中 【CONSTDATA DCD 0xxxxxx】指定, 用户可以根据实际 **code** 大小按需指定位置。

4) **userdata** 区域其实地址在文件 **startup.c** 中 【UPDATEDATA DCD 0xxxxxx】指定,

用户可以根据实际 **const data** 大小按需指定位置, 本区域 **flash** 不得加锁, 程序运行过程中会有写、擦除等操作。

5) 通常 **flash** 从 **userdata** 区域开始为非加锁区域, 非 **userdata** 区域应该加锁保护。

Q: ***** 使用不同蓝牙芯片需要注意的事项

A: 1) 首先需要正确选择芯片的类型, 由【BT_DEVICE_TYPE】指定。

2) 如果使用 MTK 芯片, 需要选择 **BT_DEVICE_TYPE == > BTUartDeviceMTK662X**

3) 如果使用 **MT6626** 芯片, 需要在协议栈初始化前调用函数 **SetMTK6626RF()**, SDK 缺省使用 **MT6622**, 因此不会调用本函数。

4) 如果不知道 MTK 的具体型号是 **MT6622** 还是 **MT6626**, 那么请在协议栈初始化前调用函数 **EnableMtkRfAutoDetect()**, SDK 缺省不会调用本函数。

4) 注意 IO 复用关系是否正确。

Q: ***** 使用 MV 提供的蓝牙最终产品量产测试工具测试蓝牙

A: 1) 该量产测试工具的蓝牙名称为【MVBTTTester01】, 客户产品需要开启蓝牙开机回连功能, 需要对代码做适当的修改后方能使用。

2) 有两种方案可供客户选择:

方案 A--需要在 **bt_e2prom.c** 修改 **FlashGetAliveAddr()** 函数的获取方式, 用户选择一种触发条件执行修改的内容, 返回【MVBTTTester01】的蓝牙地址 (需要与测试设备一致)

方案 B--需要在 **bt_ui_callback.c** 中打开宏开关【**AUTO_MV_BT_TEST_FUNC**】, 用户选择一种触发条件执行修改变量 **MvBtAutoTestFlag=1;**

3) 优缺点:

方案 A--需要指定【MVBTTTester01】的蓝牙地址, 优点是速度快

方案 B--测试时间可能会增加, 优点是适合所有【MVBTTTester01】(可以是不同的地址)

4) 产品需要方案商自行设计实现一种触发条件执行上述的功能, 最终用户正常使用时需要保证不会执行上述的代码 (需要条件触发)。

5) 测试方法参考对应工具的说明文档。

Q: ***** 是否使用简单配对

A: 1) 协议栈缺省使用蓝牙 2.1 的简单配对工作模式
2) 如用户考虑使用非简单配对, 那么需要在协议栈初始化前调用函数 `BTDisableSimplePairing()`;
3) 使用非简单配对, SDK 缺省使用的 pin code 为 0000, 用户可以在协议栈初始化成功后 (在函数 `BTStackLoop` 之前), 调用函数 `BTStackSetPinCode` 改变该值。

Q: ***** 是否使用 Advanced AVRCP *****

A: 1) 协议栈缺省情况下该功能处于打开状态
2) 如果需要关闭该功能, 可调用函数 `DisableAdvancedAvrcpFeature()`
3) 协议会定时 (周期为 1 秒) 报告播放状态, 如果需要关闭该功能, 可调用函数 `DisableAdvancedAvrcpPlayStatusAutoReport()`

Q: ***** 是否使用 iAp 功能

A: 1) 协议栈缺省情况下该功能是关闭的, 如果需要打开该功能, 可调用函数 `EnableAppleIapFunction()`

Q: ***** 是否使用 HID 功能

A: 1) 协议缺省不支持 HID 功能, 如果需要可调用函数 `BtEnableHidFunc()`
2) HID 的相关使用方法请参考文档【HID Profile 使用说明.doc】

Q: ***** 如果使用的蓝牙 lib 为非 OS 相关版本

A: 1) 标准 SDK 蓝牙 lib 需要使用系统 OS 相关的接口函数才能正常使用, 也即标准 SDK 下无需做任何修改就能正常工作。
2) 针对某些特殊的客户, 不使用我们标准 SDK 的框架, 并且内存均由用户自行管理, 那么一定只能使用非 OS 相关的 lib (版本号后缀为 iOS)
3) 使用专用的非 OS 相关的 lib, 需要用户自行实现内存分配的功能, 协议栈不再管理内存 (包括对应内存的释放)
4) 内存分配采用注册内存分配钩子函数的形式使用, 对应注册函数为 `BTStackSetMemAllocHook()`
5) 钩子函数原型定义为: `typedef void* (*BTStackMemAllocHookFunc)(int32_t Size);`

Q: ***** 如何使用的蓝牙的 SPP 功能

A: 1) SPP 的是否使用取决在协议栈初始化函数 `BTStackRunInit()` 携带的参数。
2) SPP 的具体使用说明可参考文档【SPP Profile 使用说明.doc】

History:

yizhou @ 2014.05.29 initial version

yizhou @ 2014.09.04 add advAVRCP, iAp, hid function

yizhou @ 2014.09.15 add 内存分配钩子函数的用法、SPP 的用法