Porting Realtek Bluetooth Uart H5 driver into Android 5.0 Guide

Date : 2014/11/28

Version : 1.0

This document is subject to change without notice. The document contains Realtek confidential information and must not be disclosed

日期	版本	说明
2014/11/28	V1.0	1. The First version for android 5.0

Realtek Confidential

目　录

1 开始 PORTING CODE 4
1.1 SDK 修改说明 5
1.2 平台相关修改说明 5
1.2.1 build 6
1.2.2 device 7
1.2.3 bluedroid 11
1.2.4 hardware 12
1.2.5 packages 13
1.3 BLUETOOTH PATCHES 14
1.3.1 如何合入 patch 14
1.3.2 patch 文件说明 14
1.4 KERNEL 设置 15
1.4.1 TUN driver 15
1.4.2 UINPUT driver 15
1.4.3 HID driver 15
2 BT 功能配置 16
2.1 支持的 PROFILE 配置 16
2.2 本地设备名称，COD 以及 HFP 支持设定 16
2.3 在 HOST 端配置蓝牙地址 17
3 PORTING 结束后的基本测试 19
3.1 配置检查 19
3.2 BT 测试 19
3.2.1 基本功能测试 19
3.2.2 打开 bluedroid Stack Log 打印方法 19
3.3 常见问题分析 21
3.3.1 BT 打开失败 21
3.3.2 HFP 没有声音或者都是噪音 21

Realtek Confidential

1 开始 Porting Code

1. 本文中所有 code 的修改都遵循如下格式：

a) 所有修改的 code 都使用方框括起来。

b) 方框中的 code 会使用 SDK 中没有颜色的原有 code 给出修改位置的信息。

c) 所有修改或添加的 code 均使用灰色突出显示。

2. 对 code 格式的来举例说明：

如下 code 是 SDK 中原有的 code：

```
ifeq ($(BLUETOOTH_HCI_USE_MCT),true)
LOCAL_CFLAGS := -DHCI_USE_MCT
LOCAL_SRC_FILES += \
        src/hci_mct.c \
        src/userial_mct.c
else
LOCAL_SRC_FILES += \
    src/hci_h4.c \
    src/userial.c
endif
```

Realtek 需要添加 H5 的支援，则在 porting guide 中会给出如下 code：

```
ifeq ($(BLUETOOTH_HCI_USE_MCT),true)
LOCAL_CFLAGS := -DHCI_USE_MCT
LOCAL_SRC_FILES += \
        src/hci_mct.c \
        src/userial_mct.c
else
ifeq ($(BLUETOOTH_HCI_USE_RTK_H5),true)
LOCAL_CFLAGS := -DHCI_USE_RTK_H5
LOCAL_SRC_FILES += \
    src/hci_h5.c \
    src/userial.c \
    src/bt_skbuff.c \
    src/bt_list.c
```

Realtek Confidential

```
else
LOCAL_SRC_FILES += \
        src/hci_h4.c \
        src/userial.c
endif
endif
```

- 灰色部分 code 是 Realtek 所添加以及修改的 code。

3. porting 说明

该文档 porting 说明中对应的厂商以及平台分别标识为 companyName 和 platformName，不同的客户会对应到不同的厂商以及平台，请客户在 porting 时注意修改到对应平台的文件。

Realtek Confidential

1.1 SDK 修改说明

为了方便客户整合 Realtek 的 WIFI/BT Combo Chip 到自己的平台中，Realtek 对于如何整合 BT 的 Driver 到客户平台分成了 3 个部分：平台相关修改说明，5.0_bluetooth_patches 以及 Kernel 设置。下面会分别说明。

1.2 平台相关修改说明

为了支持 Realtek WIFI/BT Combo Chip， Realtek 对客户平台的如下文件做了修改或添加：

Chg： 表示 Realtek 在原始的 SDK 文件上做了修改。

New： 表示原始的 SDK 没有该文件，Realtek 添加了该文件。

1. build

Chg build\core\product.mk

2. device

Chg device**companyName\platformName**\BoardConfig.mk

Chg device\ **companyName\platformName\platformName.mk**

Chg device\ **companyName\platformName**\init.platformName.rc

Chg device**companyName\platformName**\overlay\frameworks\base\core\res\res\values\config.xml

3. external

Chg external\bluetooth\bluedroid\hci\Android.mk

Chg external\bluetooth\bluedroid\hci\src\bt_hci_bdroid.c

Chg external\bluetooth\bluedroid\hci\src\userial.c

New external\bluetooth\bluedroid\hci\src\hci_h5.c

New external\bluetooth\bluedroid\hci\src\bt_list.c

New external\bluetooth\bluedroid\hci\src\bt_skbuff.c

New external\bluetooth\bluedroid\hci\include\bt_list.h

New external\bluetooth\bluedroid\hci\include\bt_skbuff.h

4. hardware

New hardware\realtek\bt\libbt

New hardware\realtek\bt\firmware

New hardware\libhardware\include\hardware\ bluetoothmp.h

5. packages

New packages\apps\BluetoothMpTest

1.2.1 build

1. 修改的文件

1) build\core\product.mk

```
_product_stash_var_list += \
    BOARD_WPA_SUPPLICANT_DRIVER \
    BOARD_WLAN_DEVICE \
    BOARD_USES_GENERIC_AUDIO \
    BOARD_KERNEL_CMDLINE \
    BOARD_KERNEL_BASE \
    BOARD_HAVE_BLUETOOTH \
    BOARD_HAVE_BLUETOOTH_BCM \
    BOARD_HAVE_BLUETOOTH_QCOM \
    BOARD_HAVE_BLUETOOTH_RTK \
```

Realtek Confidential

```
    BOARD_VENDOR_QCOM_AMSS_VERSION \
    BOARD_VENDOR_USE_AKMD \
    BOARD_EGL_CFG \
    BOARD_BOOTIMAGE_PARTITION_SIZE \
        …
```

添加 Realtek BT Chip 支持的宏定义。

1.2.2 device

该目录主要用于设定不同硬件平台 board 配置信息。不同的子目录对应不同的硬件平台，需要根据具体硬件平台进行修改。

Realtek Confidential

1. 修改的文件

1) device**companyName****platformName**\BoardConfig.mk

```
# OTA
TARGET_RECOVERY_UPDATER_LIBS += libnvrecoveryupdater

BOARD_BLUETOOTH_BDROID_BUILDCFG_INCLUDE_DIR ?=
device/generic/common/bluetooth
BOARD_HAVE_BLUETOOTH := true
#BOARD_HAVE_BLUETOOTH_BCM := true//commit by realtek
#Realtek add start
BOARD_HAVE_BLUETOOTH_RTK := true
BLUETOOTH_HCI_USE_RTK_H5 := true
#Realtek add end

USE_CAMERA_STUB := false
```

设定 BOARD_HAVE_BLUETOOTH_RTK 为 true，表示支持 Realtek BT chip，设定 BLUETOOTH_HCI_USE_RTK_H5 为 true，表示 UART 传输采用 H5 协议。

2) device\companyName\platformName\device.mk

该部分主要有两处修改，分别用于拷贝 Realtek 的配置文件以及配置系统支持 Bluetooth 以及 Bluetooth LE。

在该文件最后添加调用 **rtlbtfw_cfg.mk** 的命令，用于将 Realtek 需要的配置文件拷贝到指定的目录。

```
#Realtek add start
$(call inherit-product, hardware/realtek/bt/firmware/rtlbtfw_cfg.mk)
#realtek add end
```

如果用户需要支持 Bluetooth 以及 Bluetooth LE，需要确保有如下配置，如果已经存在，则不需要配置。

```
#Realtek add start
PRODUCT_COPY_FILES += \
frameworks/native/data/etc/android.hardware.bluetooth.xml:system/etc/permissions/android.hardware.bluetooth.xml \
frameworks/native/data/etc/android.hardware.bluetooth_le.xml:system/etc/permissions/android.hardware.bluetooth_le.xml
#realtek add end
```

Realtek Confidential

同时需要确保 frameworks/native/data/etc/目录有对应的文件 android.hardware.bluetooth.xml 以及 android.hardware.bluetooth_le.xml。如果没有请拷贝 Realtek SDK device\companyName\platformName\下对应的文件到 frameworks/native/data/etc/目录。

为了生成 Bluetooth.apk，还需要在 device.mk 中将 Bluetooth 加入 PRODUCT_PACKAGES。

```
PRODUCT_PACKAGES += Bluetooth \
                    bluetooth.mapsapi
```

3) device\companyName\platformName\init.platformName.rc

在 on boot 下面添加的代码,其中”ttyS2”是 platformName 平台的 Bluetooth UART 口,请修改此处的 ttyS2 为客户平台的 Bluetooth UART 口。例如您的平台使用 ttyS0，那么您需要将此处的”ttyS2”修改成 ttyS0。

```
on boot

# bluetooth
```

```
    # change back to bluetooth from system
    chown bluetooth net_bt_stack /data/misc/Bluetooth
    mkdir /data/misc/bluedroid 0770 bluetooth net_bt_stack

    # UART device
    chmod 0660 /dev/ttyS2
    chown bluetooth net_bt_stack /dev/ttyS2

    # power up/down interface
    chmod 0660 /sys/class/rfkill/rfkill0/state
    chmod 0660 /sys/class/rfkill/rfkill0/type
    write /sys/class/rfkill/rfkill0/state 0
    chown bluetooth net_bt_stack /sys/class/rfkill/rfkill0/state
    chown bluetooth net_bt_stack /sys/class/rfkill/rfkill0/type

    # bluetooth MAC address programming
    chown bluetooth net_bt_stack ro.bt.bdaddr_path
    chown bluetooth net_bt_stack /system/etc/bluetooth
    chown bluetooth net_bt_stack /data/misc/bluetooth
    setprop ro.bt.bdaddr_path "/data/misc/bluetooth/bdaddr"
```

Realtek Confidential

如果需要支持 Bluetooth PAN,需要添加 bluetooth 的配置

```
service dhcpcd_bt-pan /system/bin/dhcpcd -ABKL
    class main
    disabled
    oneshot

service iprenew_bt-pan /system/bin/dhcpcd -n
    class main
    disabled
    oneshot
```

4) device\companyName\platformName\overlay\frameworks\base\core\res\res\values\config.xml

如果需要支持 Bluetooth PAN,需要添加 bluetooth 的配置。

```
<string-array translatable="false" name="config_tether_bluetooth_regexs">
        <item>"bt-pan"</item>
</string-array>

<string-array translatable="false" name="networkAttributes">
        <item>"wifi,1,1,1,-1,true"</item>
        <!--item>"mobile,0,0,0,-1,true"</item>
        <item>"mobile_mms,2,0,2,60000,true"</item>
        <item>"mobile_supl,3,0,2,60000,true"</item>
        <item>"mobile_hipri,5,0,3,60000,true"</item>
        <item>"bluetooth,7,7,3,-1,true"</item>
        <item>"mobile_fota,10,0,2,60000,true"</item>
        <item>"mobile_ims,11,0,2,60000,true"</item>
        <item>"mobile_cbs,12,0,2,60000,true"</item-->
        <item>"wifi_p2p,13,1,0,-1,true"</item>
        <item>"eth,9,9,4,60000,true"</item>
    </string-array>

    <!-- An Array of "[ConnectivityManager connectionType],
                    [# simultaneous connection types]"    -->
    <string-array translatable="false" name="radioAttributes">
        <item>"1,1"</item>
        <!--item>"0,1"</item-->
        <item>"7,1"</item>
        <item>"9,1"</item>
    </string-array>
```

Realtek Confidential

5）\packages\apps\Bluetooth\Android.mk

请检查这个 Android.mk 文件，确保 **include $(BUILD_PACKAGE)** 是打开没有被注释掉。

```
LOCAL_PROGUARD_ENABLED := disabled
include $(BUILD_PACKAGE)
include $(call all-makefiles-under,$(LOCAL_PATH))
```

1.2.3 bluedroid

只是修改 **external\bluetooth\bluedroid\hci** 目录，添加了对 realtek H5 的支持。

1. 修改的文件

1. external\bluetooth\bluedroid\hci\Android.mk

```
ifeq ($(BLUETOOTH_HCI_USE_MCT),true)
LOCAL_CFLAGS := -DHCI_USE_MCT
LOCAL_SRC_FILES += \
        src/hci_mct.c \
        src/userial_mct.c
else
ifeq ($(BLUETOOTH_HCI_USE_RTK_H5),true)
LOCAL_CFLAGS := -DHCI_USE_RTK_H5
LOCAL_SRC_FILES += \
        src/hci_h5.c \
        src/userial.c \
 src/bt_skbuff.c \
 src/bt_list.c
else
LOCAL_SRC_FILES += \
        src/hci_h4.c \
        src/userial.c
endif
endif
```

设定当 BLUETOOTH_HCI_USE_RTK_H5 为 true 的情况下，编译时应当包含的源文件列表以及宏定义。

2. external\bluetooth\bluedroid\hci\src\bt_hci_bdroid.c

```
static int init(const bt_hc_callbacks_t* p_cb, unsigned char *local_bdaddr)
{
    #ifdef HCI_USE_MCT
        extern tHCI_IF hci_mct_func_table;
        p_hci_if = &hci_mct_func_table;
    #elif defined HCI_USE_RTK_H5
```

Realtek Confidential

```
        extern tHCI_IF hci_h5_func_table;
        p_hci_if = &hci_h5_func_table;
    #else
        extern tHCI_IF hci_h4_func_table;
        p_hci_if = &hci_h4_func_table;
    #endif
    …
    …
    …

}
```

如果定义了宏 HCI_USE_RTK_H5,则使用 H5 的接口注册到 p_hci_if ，而不是采用默认的 H4。

Realtek Confidential

2. 添加的文件

下面是添加的文件列表，直接拷贝到对应目录即可，主要是为了支持 Realtek UART H5。

New	external\bluetooth\bluedroid\hci\src\hci_h5.c
New	external\bluetooth\bluedroid\hci\src\bt_list.c
New	external\bluetooth\bluedroid\hci\src\bt_skbuff.c
New	external\bluetooth\bluedroid\hci\include\bt_list.h
New	external\bluetooth\bluedroid\hci\include\bt_skbuff.h

1.2.4 hardware

主要是为 Realtek 的 BT Chip 添加 libbt-vendor 模块以及固件配置文件。

1. 添加的文件

1) hardware\realtek\bt\libbt

此处存放 Realtek Vendor lib 的源代码，直接拷贝到对应目录即可,

hardware\realtek\bt\libbt\include\目录下会有一个 vnd_platformName.txt 文件，其中

BLUETOOTH_UART_DEVICE_PORT = "/dev/ttyS2"

会指定该平台对应的用于 Bluetooth 的 UART 是哪一个，如果该文件夹下没有对应平台的 txt 档，则需要添加 vnd_platformName.txt 文件（其中 platformName 为平台名称），并且需要正确配置 BLUETOOTH_UART_DEVICE_PORT 指定在该平台下哪一个 UART 用于 bluetooth。

2) hardware\realtek\bt\firmware

此处存放 realtek 的配置文件，该文件夹直接拷贝即可。

3) hardware\libhardware\include\hardware\bluetoothmp.h

该头文件用于 MP tool。

1.2.5 packages

Realtek Confidential

主要是为 Realtek 的 BT Chip 添加 MP Tool 模块。

1. 添加的文件

1) packages\apps\BluetoothMpTest

该文件夹直接拷贝即可

2. 修改的文件

2) packages\apps\BluetoothMpTest\jni\bluedroid\libbt\include\bt_vnd_rtk.h

由于 MP tool 测试 Bluetooth RF 功能时，也是需要使用 UART 口的，所以此处需要修改该文件来配置 Bluetooth UART 口。在 49-51 行有如下的定义，客户必须修改 BLUETOOTH_UART_DEVICE_PORT 为平台对应的用于 Bluetooth 的 UART 口。

```
/* Device port name where Bluetooth controller attached */
#ifndef BLUETOOTH_UART_DEVICE_PORT
#define BLUETOOTH_UART_DEVICE_PORT      "/dev/ttyS2"
#endif
```

1.3 Bluetooth patches

为了使 Realtek BT chip 在 android5.0 上能够正常工作，还需要合入下面的 patch 文件。需要合入的 bluetooth patches 文件都在 5.0_bluetooth_patches 文件夹下。其中文件夹的命令方式是以目录的方式来命名的，比如：**external_bluetooth_bluedroid**_patches 表示这里面的 patch 都是合入到 ***external/bluetooth/bluedroid/*** 目录，一般情况下这些目录都会有 git project，这时客户只需要将该目录下的 patch 文件全部合入（git am *.patch）到客户的 SDK 中即可。

如果该目录没有 git project 的话，那么需要客户自己添加这部分到客户的 SDK 中，方便后期维护。比如 hardware_realtek_bt_patches 这个目录代表 *hardware/realtek/bt*，这个是 Realtek 提供的一个 BT 模块在客户的 SDK 中应该是没有 git project 的，那么就需要客户把这部分添加进去。

Realtek Confidential

1.3.1 如何合入 patch

客户需要首先拷贝 patch 文件到指定的目录，比如对于 external_bluetooth_bluedroid_patches 目录下的 patch 文件，首先需要将这下面的 patch 文件全部拷贝到 external\bluetooth\bluedroid 目录，然后在 external\bluetooth\bluedroid 目录执行下面的命令打入所有的 patch。

```
git am *.patch
```

当然，合入 patch 的方法还有很多种，客户也可以采用自己熟悉的方式。合入 patch 时有可能会出现冲突，如果出现冲突了，就需要客户手动来修改解决了。

1.3.2 patch 文件说明

1）external_bluetooth_bluedroid_patches

该目录下的 patch 用于修正 bluedroid 的一些 bug。

2）packages_apps_Bluetooth_patches

该目录下的 patch 用于修正 bluetooth java 层的一些 bug。

3）packages_apps_Settings_patches

该目录下的 patch 用于修正 bluetooth java 层的一些 bug。

1.4 Kernel 设置

1.4.1 TUN driver

如果需要支持 Bluetooth PAN， 确保 TUN Driver 已经编译到 kernel，

kernel\arch\arm\configs\XXX_defconfig

```
CONFIG_TUN=y
```

Realtek Confidential

1.4.2 UINPUT driver

如果需要支持 AVRCP 功能，确认以下打开配置，

```
CONFIG_INPUT_UINPUT=y      # User level driver support
CONFIG_INPUT_MISC=y
```

1.4.3 HID driver

如果需要使用 Bluetooth HID，必须支持 uhid driver；确认打开需要支持的 HID 配置

```
CONFIG_UHID=y
CONFIG_HID_xxx=y
```

2 BT 功能配置

2.1 支持的 Profile 配置

对于有些平板应用不需要支持 PBAP，HFP 以及 HSP，可以按照下面的配置来关闭这几个 Profile，如果用户需要支持，只需把对应值设置为 true 即可。

packages/apps/Bluetooth/res/values/config.xml 文件如下

```
<resources>
    <bool name="profile_supported_a2dp">true</bool>
    <bool name="profile_supported_a2dp_sink">fasle</bool>
    <bool name="profile_supported_hdp"> false </bool>
    <bool name="profile_supported_hs_hfp"> false </bool>
    <bool name="profile_supported_hfpclient">fasle</bool>
    <bool name="profile_supported_hid">true</bool>
    <bool name="profile_supported_opp">true</bool>
    <bool name="profile_supported_pan">true</bool>
    <bool name="profile_supported_pbap"> false </bool>
    <bool name="profile_supported_gatt">true</bool>
    <bool name="pbap_include_photos_in_vcard"> false </bool>
    <bool name="pbap_use_profile_for_owner_vcard"> false </bool>
    <bool name="profile_supported_map"> false </bool>
    <bool name="profile_supported_avrcp_controller">false</bool>
</resources>
```

Realtek Confidential

2.2 本地设备名称，COD 以及 HFP 支持设定

修改 BTM_DEF_LOCAL_NAME 为平台需要的名称。

修改 BTA_DM_COD 为平台需要的 COD。

默认情况下，在 Android4.2 的 SDK 中，只支持 HSP，不支持 HFP，用户需要支持 HFP，请

修改 BTIF_HF_SERVICES 以及 BTIF_HF_SERVICE_NAMES。

#define BTIF_HF_SERVICES (BTA_HSP_SERVICE_MASK| BTA_HFP_SERVICE_MASK)

#define BTIF_HF_SERVICE_NAMES { BTIF_HSAG_SERVICE_NAME,BTIF_HFAG_SERVICE_NAME}

```
#ifndef _BDROID_BUILDCFG_H
#define _BDROID_BUILDCFG_H

#define BTM_DEF_LOCAL_NAME "Realtek Tablet"
// SERVICE_CLASS:0x5A (Bit17 -Networking,Bit19 - Capturing,Bit20 -Object Transfer,Bit22
-Telephony)
// MAJOR CLASS: COMPUTER
// MINOR CLASS: TABLET
#define BTA_DM_COD {0x5A, 0x01, 0x1C}

#define BTIF_HF_SERVICES (BTA_HSP_SERVICE_MASK)
#define BTIF_HF_SERVICE_NAMES { BTIF_HSAG_SERVICE_NAME }

#endif
```

Realtek Confidential

2.3 在 host 端配置蓝牙地址

Bluetolth controller 内部有默认设置蓝牙地址，host 也可以修改通过一下方法修改蓝牙地址

1）在 hardware/realtek/bt/libbt/include/vnd_xxx.txt 文件中添加定义

USE_CONTROLLER_BDADDR = FALSE

2）在 init.xxx.rc 中修改制定 bt address 的文件路径

默认设置为：setprop ro.bt.bdaddr_path "/data/misc/bluetooth/bdaddr"

stack 中获取 address 的地方在 btif_core 中 btif_fetch_local_bdaddr 函数中，

获取位置优先级："ro.bt.bdaddr_path" > "persist.service.bdroid.bdaddr" > rand

所以需要修改"ro.bt.bdaddr_path"为指定 bt address 文件路径

3）关于 bdaddr 文件中 BT address 格式应该为 00:00:00:AA:BB:CC

(中间必须以非字符隔开）

Realtek Confidential

3 Porting 结束后的基本测试

3.1 配置检查

为了进一步确保 porting 没有问题，在测试之前先确认 fw 以及 config 文件是否存在。

1) adb shell 到测试平台的根目录，检查测试平台的 system/etc/firmware/目录中 rtlxxxx_fw 以及 rtlxxxx_config 文件是否存在(xxxx 为 BT Chip 型号)。

3.2 BT 测试

注意：本测试是 porting 结束后对 BT 基本和常用功能的一个快速测试，旨在快速验证一些基本问题，不代表 BT 完整的测试，测试结果也非正式 test report。如果使用的是非 Realtek BT chip，该项测试可能没有意义。

Realtek Confidential

3.2.1 基本功能测试

1) 打开/关闭 BT 无失败现象。
2) 能够搜索到近处 BT 设备。
3) 和搜索到的蓝牙耳机或其他设备配对。
4) 连接上蓝牙耳机，使用 BT A2DP 听音乐(sdcard 确保存在)。
5) 连接上蓝牙耳机，使用 BT HFP/HSP 打电话(确保不保用蓝牙时能够正常通话)。
6) 传输文件到远端支持蓝牙 OPP Server 的设备，从远端支持蓝牙 OPP client 的设备传送文件到本地(sdcard 确保存在)。
7) 连接上蓝牙键盘，打开需要输入的应用，通过蓝牙键盘输入。

3.2.2 打开 bluedroid Stack Log 打印方法

修改 system/etc/Bluetooth/bt_stack.conf 檔，把 Debug Level 從 2 調成 5， 并打开 BtSnoop 的 LOG，增加 H5 log 输出选项 H5LogOutput，如果需要测试 H5 数据发送/接收，设置 H5LogOutput 为 ture.

```
# Enable BtSnoop logging function
# valid value : true, false
BtSnoopLogOutput=true

# valid value : true, false
H5LogOutput= false

# BtSnoop log output file
BtSnoopFileName=/sdcard/btsnoop_hci.log

# Enable trace level reconfiguration function
# Must be present before any TRC_ trace level settings
TraceConf=true

# Trace level configuration
#   BT_TRACE_LEVEL_NONE    0    ( No trace messages to be generated )
#   BT_TRACE_LEVEL_ERROR   1    ( Error condition trace messages )
#   BT_TRACE_LEVEL_WARNING 2    ( Warning condition trace messages )
#   BT_TRACE_LEVEL_API     3    ( API traces )
#   BT_TRACE_LEVEL_EVENT   4    ( Debug messages for events )
#   BT_TRACE_LEVEL_DEBUG   5    ( Full debug messages )
TRC_BTM=5
TRC_HCI=5
TRC_L2CAP=5
TRC_RFCOMM=5
TRC_OBEX=5
TRC_AVCT=5
TRC_AVDT=5
TRC_AVRC=5
TRC_AVDT_SCB=5
TRC_AVDT_CCB=5
TRC_A2D=5
TRC_SDP=5
TRC_GATT=5
TRC_SMP=5
TRC_BTAPP=5
TRC_PAN=5
TRC_BNEP=5
```

Realtek Confidential

3.3 常见问题分析

3.3.1 BT 打开失败

打开 H5 UART Driver Log， 使用 logcat 抓取 log，看 H5 SYNC 过程时候成功，如果 H5 SYNC 失败，那么需要首先检查硬件电路是否正确（Power Supply， BT Reset PIN， UART TX/RX，CTS/RTS），然后检查卡片 efuse，用示波器量测 UART 波形，看 Host 是否把数据正确的发送到 Controller。

如果 H5 SYNC 成功，那么下一步就是 Change Baudrate，判断 Change Baudrate 是否成功。如果 Change Baudrate 失败，那么需要确定 Host 是否支持该波特率，config 文件是否正确设定了波特率。

如果 Change Baudrate 成功，下一步是下载 fw 以及 config 文件，如果下载完毕之后，收不到 Controller 回复的 Command Complete Event，那么需要检查 fw 以及 config 文件是否正确，BT Reset PIN 是否为高电平。

如果下载 fw 以及 config 文件成功，那么下一步就是根据 config 文件的设定修改 HW Flowcontrol 的设置。设置成功之后，bluedroid stack 会下第一个 HCI Comamnd。

如果第一个 HCI Command 一直 H5 重传，那么说明可能 HW flowcontrol 有问题，需要检查 Host 的 UART driver 是否支持 HW Flowcontrol。

3.3.2 HFP 没有声音或者都是噪音

首先需要检查，Audio 模块是否把声音切换到 BT，其次需要检查 BT config 文件中的 PCM 设定是否与平台的 PCM 设定匹配。

Realtek Confidential