



REALTEK

Porting Realtek Bluetooth USB driver into Android 5.0 Guide

Date : 2014/12/31

Version : 1.0

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

Realtek Bluetooth USB Driver on Android 5.0 Porting Guide

日期	版本	说明
2014/12/31	V1.0	1. The First version for android5.0

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	hardware	9
1.3	KERNEL 设置	9
1.3.1	rtk_btusb driver	9
1.3.2	uhid driver	10
1.3.3	platform configuration.....	11
1.3.4	uinput driver	12
1.3.5	tun driver	12
1.4	BLUETOOTH PATCHES	12
1.4.1	如何合入 patch.....	13
1.4.2	patch 文件说明	13
2	BT 功能配置（可选）	14
2.1	支持的 PROFILE 配置	14
2.2	本地设备名称，COD 以及 HFP 支持设定	14
2.3	配置 EXTRA CONFIG 文件.....	16
3	PORTING 结束后的基本测试.....	17
3.1	配置检查.....	17
3.2	ENABLE BLUEDROID BTSNOOP LOG(可选)	17

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

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

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

```
else
LOCAL_SRC_FILES += \
    src/hci_h4.c \
    src/serial.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。

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 2 个部分：平台相关修改说明以及 5.0_bluetooth_patches。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

New device\companyName\platformName\bluetooth\bdroid_buildcfg.h

Chg device\companyName\platformName\BoardConfig.mk

Chg device\companyName\platformName\platformName.mk

New device\companyName\platformName\android.hardware.bluetooth.xml

New device\companyName\platformName\android.hardware.bluetooth_le.xml

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

3. hardware

New hardware\realtek*

4. kernel

Chg kernal\arch\arm\configs\ **platformName**_android_defconfig

New kernal\driver\bluetooth\rtk_btusb.c

New kernal\driver\bluetooth\rtk_btusb.h

Chg kernal\driver\bluetooth\Kconfig

Chg kernal\driver\bluetooth\Makefile

New kernal\driver\hid\uhid.c

Chg kernal\driver\hid\Kconfig

Chg kernal\driver\hid\Makefile

New kernal\include\linux\uhid.h

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 \
    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。

1. 修改的文件

- 1) 将 Realtek 提供的文件\bluetooth\bdroid_buildcfg.h 拷贝到目录
device/{vendor}/{platform}/bluetooth/下（如果该目录已有此文件，可忽略此项）
- 2) 修改文件 **device\companyName\platformName\BoardConfig.mk**:

```
# OTA
TARGET_RECOVERY_UPDATER_LIBS += libnvrecoveryupdater

BOARD_BLUETOOTH_BDROID_BUILDCFG_INCLUDE_DIR ?=
device/{vendor}/{platform}/bluetooth
BOARD_HAVE_BLUETOOTH := true
#BOARD_HAVE_BLUETOOTH_BCM := true//commit by realtek
#BOARD_HAVE_BLUETOOTH_QCOM := true//commit by realtek
#BLUETOOTH_HCI_USE_MCT := true
#Realtek add start
BOARD_HAVE_BLUETOOTH_RTK := true
#Realtek add end

USE_CAMERA_STUB := false
```

设定 BOARD_HAVE_BLUETOOTH_RTK 为 true，表示支持 Realtek BT chip。

3) **device\companyName\platformName\platformName.mk**

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

- a. platformName.mk 文件最后添加调用的命令，用于将 Realtek 需要的 fw 文件拷贝到指定的目

录。

```
#Realtek add start
$(call inherit-product, hardware/realtek/bt/firmware/rtl8723a/device-rtl.mk)
$(call inherit-product, hardware/realtek/bt/firmware/rtl8723b/device-rtl.mk)
$(call inherit-product, hardware/realtek/bt/firmware/rtl8761a/device-rtl.mk)
$(call inherit-product, hardware/realtek/bt/firmware/rtl8821a/device-rtl.mk)
$(call inherit-product, hardware/realtek/bt/firmware/rtl8822b/device-rtl.mk)
$(call inherit-product, hardware/realtek/bt/firmware/rtl8723d/device-rtl.mk)
$(call inherit-product, hardware/realtek/bt/firmware/rtl8821c/device-rtl.mk)
#realtek add end
```

- b. 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
```

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 提供的 device\companyName\platformName\ 下对应的文件到 frameworks/native/data/etc/ 目录。

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

```
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

# USB device
insmod /system/lib/modules/rtk_btusb.ko （根据rtk_btusb.ko所在的平台目录修改此路径!）
```



```
chmod 0660 /dev/rtk_btusb  
chown bluetooth net_bt_stack /dev/rtk_btusb  
  
# 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"
```

- 5) 修改目录 **device\companyName\platformName** 下的文件 **ueventd.platformName.rc**，在文件末尾添加

```
/dev/rtk_btusb          0660    bluetooth    net_bt_stack
```

- 6) 修改目录 **device\companyName\platformName\sepolicy** 下的文件 **file_contexts**，在此文件中添加

```
/dev/rtk_btusb          u:object_r:hci_attach_dev:s0
```

- 7) 检查 **device\companyName\platformName\device-common.mk** 文件中的变量 **LOCAL_KERNEL** 的值，确保指向的是正确的 **Kernel** 的镜像。

1.2.3 hardware

将 Realtek 提供的 **/hardware/realtek/** 整个目录拷贝到 **/hardware/** 下。

1.3 Kernel 设置

1.3.1 rtk_btusb driver

1. 将 Realtek 提供的文件 **rtk_btusb.h** 和 **rtk_btusb.c** 拷贝到目录 **kernel/drivers/bluetooth/** 下；
2. 修改目录 **/kernel/drivers/bluetooth/** 下的文件“Kconfig” and “Makefile”

在 Kconfig 文件中增加 **rtu_btusb** 的选项：

```
config BT_HCIBTUSB
```

```
    tristate "HCI USB driver"
```

```
    depends on USB
```

```
    help
```

```
        Bluetooth HCI USB driver.
```

```
        This driver is required if you want to use Bluetooth devices with
        USB interface.
```

```
        Say Y here to compile support for Bluetooth USB devices into the
        kernel or say M to compile it as module (btusb).
```

```
config BT_RTKBTUSB
```

```
    tristate "RTK HCI USB driver"
```

```
    depends on USB
```

```
    help
```

```
        RTK Bluetooth HCI USB driver
```

在 Makefile 文件中添加目标文件 rtk_btusb.o:

```
obj-$(CONFIG_BT_HCIBTUSB) += btusb.o
```

```
obj-$(CONFIG_BT_RTKBTUSB) += rtk_btusb.o
```

1.3.2 uhid driver

如果 kernel 不支持 uhid driver，需要加入 uhid driver 以支持蓝牙 HID 以及 HOGP profile.

1. 将 Realtek 提供的文件 **uhid.c** 拷贝到目录 **/kernel/drivers/hid/**下
2. 将 Realtek 提供的文件 **uhid.h** 拷贝到目录 **/kernel/include/linux/**下
3. 修改目录 **/kernel/drivers/hid/**下的文件 “Kconfig” 和 “Makefile”

在 Kconfig 文件中增加 uhid 的选项:

```
config UHID
    tristate "User-space I/O driver support for HID subsystem"
    depends on HID
    default n
    ---help---
    Say Y here if you want to provide HID I/O Drivers from user-space.
    This allows to write I/O drivers in user-space and feed the data from
    the device into the kernel. The kernel parses the HID reports, loads the
    corresponding HID Device Driver or provides input devices on top of
    your user-space device.
    This driver cannot be used to parse HID-reports in user-space and write
    special HID-drivers. You should use hidraw for that.
    Instead, this driver allows to write the transport-layer driver in
    user-space like USB-HID and Bluetooth-HID do in kernel-space.
    If unsure, say N.
    To compile this driver as a module, choose M here: the
    module will be called uhid

source "drivers/hid/usbhid/Kconfig"

menu "Special HID drivers"
    depends on HID
```

在 Makefile 文件增加目标文件 uhid.o:

```
obj-$(CONFIG_HID)      += hid.o
obj-$(CONFIG_UHID)     += uhid.o
```

1.3.3 platform configuration

需要确保平台对应的 config 文件中已配置支持 rtk_butsb driver, 该配置文件一般位于

Realtek Bluetooth USB Driver on Android 5.0 Porting Guide

kernel/arch/arm/configs/目录下，用来配置编译 kernel 的选项，你可以直接修改 “.config” 文件或者用命令 “make menuconfig” 来修改 kernel 选项，无论选择哪种方法，最终都要确保有以下配置：

```
CONFIG_UHID=y
CONFIG_BT_RTKBTUSB=m
```

1.3.4 uinput driver

如果需要支持 AVRCP 功能，确保\kernel\arch\arm\configs\XXX_defconfig 文件中有以下配置：

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

1.3.5 tun driver

如果需要支持 Bluetooth PAN 功能，确保\kernel\arch\arm\configs\XXX_defconfig 文件中有以下配置：

```
CONFIG_TUN=y
```

1.4 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 patches 文件夹，其内容如下：

 external_bluetooth_bluedroid_patches	2016/8/22 15:55	文件夹
 framework_patches	2016/8/22 15:55	文件夹
 packages_apps_Bluetooth_patches	2016/8/22 15:55	文件夹

Realtek 的 patch 文件都放在上面的几个目录中，其中文件夹的命名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。

1.4.1 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_bluetooth_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\bluetooth 目录，然后在 external\bluetooth\bluetooth 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。

1.4.2 patch 文件说明

1) external_bluetooth_bluetooth_patches

该目录下目前有如下 patch，都是修改 bluetooth 的其他问题，客户可以从 git commit log 中查看。

0001-Realtek-BT-BT-open-fail-workaround.patch	2014/10/17 14:28	PATCH 文件
0002-Realtek-BT-Save-device-name-change-information.patch	2014/10/17 14:28	PATCH 文件
0003-Realtek-BT-Disable-inquiry-when-pair-is-on-going.patch	2014/12/10 11:50	PATCH 文件
0004-Realtek-BT-Add-pair-retry.patch	2014/12/10 16:10	PATCH 文件
0005-Realtek-BT-Fix-hid-could-not-pair.patch	2014/12/10 16:17	PATCH 文件
0006-Realtek-BT-Set-default-scan-mode-standard.patch	2014/10/17 14:28	PATCH 文件
0007-Extend-RFC-timeout-from-3-to-30.patch	2014/10/17 14:28	PATCH 文件
0008-Realtek-BT-disable-LPM.patch	2014/10/17 14:28	PATCH 文件
0009-Realtek-BT-Add-protection-to-gki-module.patch	2014/12/10 16:45	PATCH 文件

2) framework_patches

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

3) packages_apps_Bluetooth_patches

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

2 BT 功能配置（可选）

2.1 支持的 Profile 配置

对于有些平台应用不需要支持 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">false</bool>
    <bool name="profile_supported_hdp"> false </bool>
    <bool name="profile_supported_hs_hfp"> false </bool>
    <bool name="profile_supported_hfpclient">false</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>
```

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（具体请参考 SIG 官方网站 COD 部分

<https://www.bluetooth.org/zh-cn/specification/assigned-numbers/baseband>), 可以根据网站上 COD 的值来确定您需要的 COD。Realtek 默认的值是 #define BTA_DM_COD {0x5A, 0x01, 0x1C}。

Bluedroid 可以设定 HSP/HFP 的不同支持, 客户可根据需求设定:

Realtek Bluetooth USB Driver on Android 5.0 Porting Guide

1.同时支持 HSP/HFP, bluedroid 会默认优先使用 HFP, 则在 bdroid_buildcfg.h 中定义

```
#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 }
```

2. 只支持 HSP, 则在 bdroid_buildcfg.h 中定义

```
#define BTIF_HF_SERVICES (BTA_HSP_SERVICE_MASK)
```

```
#define BTIF_HF_SERVICE_NAMES { BTIF_HSAG_SERVICE_NAME, NULL }
```

```
#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, NULL }

#endif
```

2.3 配置 Extra Config 文件

对于新卡片支持 Extra Config 文件配置 config，如需使用 Extra Config 文件配置 config，可在/data/misc/bluetooth/目录下新建 rtk_btconfig.txt 文档，修改文件权限为 644。具体配置方式以 8723bs 为例说明：

```
rtl8723bs_config
0x5b 0x01 0x04 0x21 0x22 0x22 0x21
0xe6 0x01 0x01 0x20
#0xbb 0x01 0x01 0x3c
#0xed 0x00 0x01 0x00
~
```

- 1) 文件首行必须为需要配置的蓝牙芯片的 config 文件名，如这里对 8723bs config 进行配置，则首行内容必须与 8723bs config 文件名一致，即“rtl8723bs_config”
- 2) 从文件次行开始进行 config offset 及 value 值配置。格式为两字节 offset + 一字节 length + length 字节 value 值，小端模式，以 16 进制设置，以空格分隔每个字节，每行设置一个 offset 及 value。如图中次行设置即为：offset:0x015b、length:0x04 value:0x21222221
- 3) 支持用“#”进行单行注释
- 4) 不支持对 MAC address 配置

注：请不要随意使用 Extra Config 文件进行 config 内容配置，如有需要，请一定请 FAE 对配置内容进行 review，否则容易由于异常配置导致各种问题。

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 目录下是否有类似于 rtl87*_fw 文件是否存在。

3.2 Enable Bluedroid BtSnoop log(可选)

1. adb pull system/etc/bluetooth/bt_stack.conf
2. 修改 bt_stack.conf 文件，把 Debug Level 从 2 改成 6，并打开 BtSnoop 的 LOG。

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

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

# Preserve existing BtSnoop log before overwriting
BtSnoopSaveLog=false

# 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 )
# BT_TRACE_LEVEL_VERBOSE 6    ( Verbose messages ) - Currently supported for
TRC_BTAPP only.
TRC_BTM=6
TRC_HCI=6
TRC_L5CAP=6
TRC_RFCOMM=6
TRC_OBEX=6
TRC_AVCT=6
TRC_AVDT=6
TRC_AVRC=6
TRC_AVDT_SCB=6
TRC_AVDT_CCB=6
TRC_A5D=6
TRC_SDP=6
TRC_GATT=6
TRC_SMP=6
TRC_BTAPP=6
TRC_BTIF=6
```

3. adb push bt_stack.conf system/etc/bluetooth/
4. 通过 UI(Setting/bluetooth/)开关一次 bluetooth,使上述更改生效