



Porting Realtek Bluetooth Uart H5 driver into Android 4.3.x-4.4.x Guide

Date : 2014/04/08

Version : 3.3

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

Date	Version	Description
2013/12/16	V3.0	1. The First version for android4.4
2014/02/17	V3.1	1.Add mp tools
2014/03/31	V3.2	1.Support bt address setted by host
2014/04/08	V3.3	1.Fix use host bdaddress patch problem which caused host can not free config_buf
2015/09/30	V3.1	1.Remove mp tools

CONTENTS

1	START PORTING CODE	4
1.1	MODIFIED SDK INTRODUCTION	5
1.2	PLATFORM-DEPENDENT MODIFICATION	5
1.2.1	build	6
1.2.2	device	7
1.2.3	bluedroid	10
1.2.4	hardware	12
1.3	BLUETOOTH PATCHES	12
1.3.1	How to apply patch	13
1.3.2	Patches introduction.....	13
1.4	KERNEL.....	13
1.4.1	TUN driver	13
1.4.2	UINPUT driver	14
1.4.3	HID driver	14
2	BT FUNCTION CONFIGURATIN	15
2.1	CONFIG SUPPORTED PROFILE	15
2.2	LOCALNAME, COD AND HFP SUPPORT.....	15
2.3	SET BLUETOOTH ADDRESS IN HOST	16
2.4	CONFIGURE EXTRA CONFIG	17
3	BASIC FUNCTION TEST AFTER PORTING FINISHED.....	18
3.1	BT BASIC FUNCTION TEST.....	18
3.1.1	Basic function test.....	18
3.1.2	How to capture logs	18
3.1.3	Debug with bluedroid Stack Log	19

1 Start Porting Code

1. Conventions used in this document

- a) All code modified or added by realtek are highlighted in boxes.
- b) All code modified or added by realtek are highlighted in gray.
- c) All code in original SDK use normal color in boxes.

2. Porting example:

Original code in SDK:

```
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
```

Code modified for support Realtek UART H5:

```
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
```

Code modified by realtek is highlighted in gray.

3. Porting Notes

We only take cardhu platform of nvidia as an example, if you use another different platform, you must modify files related to your platform to support Realtek H5 UART driver.

1.1 Modified SDK Introduction

In order to integrate Wifi/BT combo Chip of Realtek into your platform, we provide guides for customers to merge BT driver into their SDK.

1.2 Platform-dependent modification

You need to add or modify files list below to support Realtek H5 UART driver in android 4.4.x.

Chg: Which indicates the file has been modified from its original SDK by realtek to support Realtek BT Chip.

New: Which indicates the file is a new file added by realtek to support Realtek BT chip.

1. build

Chg build\core\product.mk

2. device

Chg device\nvidia\cardhu\BoardConfig.mk

Chg device\nvidia\cardhu\carhu.mk

Chg device\nvidia\cardhu\init.cardhu.rc



Chg device\realtek\cardhu\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\serial.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. Modified files

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 \  
...
```

Add macro *BOARD_HAVE_BLUETOOTH_RTK* to support Realtek BT Chip.

1.2.2 device

This directory is used to set board configuration for different hardware platforms. Different directories map to different hardware platforms. You should modify according to your platform settings.

1. Modified files

1) device\nvidia\cardhu\BoardConfig.mk

```
# OTA  
TARGET_RECOVERY_UPDATER_LIBS += libnvrecoveryupdater  
  
BOARD_BLUETOOTH_BDROID_BUILDCFG_INCLUDE_DIR ?=  
device/nvidia/cardhu/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
```

Set “*BOARD_HAVE_BLUETOOTH_RTK*” true to support Realtek BT chip, set

“*BLUETOOTH_HCI_USE_RTK_H5*” true to support H5 UART driver.

2) device\nvidia\cardhu\carhu.mk



Realtek Bluetooth UART H5 Driver on Android 4.3.x-4.4.x Porting Guide

Add command to call **rtlbtfw_cfg.mk**, which is used to copy configuration files of Realtek to specified directory.

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

If Bluetooth function not enabled in setting page, please add `android.hardware.bluetooth.xml` in your makefile. And if you want to support Bluetooth LE, you should also add `android.hardware.bluetooth_le.xml` in your makefile

```
#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
```

First, you need to make sure files `android.hardware.bluetooth.xml` and `android.hardware.bluetooth_le.xml` is located in directory “`frameworks/native/data/etc/`”. If not, please copy them from directory “`Realtek SDK device\nvidia\cardhu\`”.

3) `device\nvidia\cardhu\init.cardhu.rc`

You need to add settings as follows to configure Bluetooth UART and power up/down interface. You need to change “`ttyHS2`” to the UART port of Bluetooth on your own platform.

```
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/ttyHS2
chown bluetooth net_bt_stack /dev/ttyHS2

# 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"
```

You need to add settings as follows to configure Bluetooth PAN.

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

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

4) device\nvidia\cardhu\overlay\frameworks\base\core\res\res\values\config.xml

You need to add settings as follows to configure Bluetooth PAN.

```
<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>
```

1.2.3 bluedroid

We only modified some files in directory “*external\bluetooth\bluedroid\hci*” to support realtek H5 UART driver.

1. Modified files

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

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

When `BLUETOOTH_HCI_USE_RTK_H5` is set true, we should turn on macro `HCI_USE_RTK_H5`, and include necessary source files used for H5.

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

```
static int init(const bt_hci_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
        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
    ...
    ...
    ...
}
```

If defined macro `HCI_USE_RTK_H5`, register H5 interface to `p_hci_if` instead of H4.

2. Added files

Add files are as follows, you need to copy them to the corresponding directory, We only add these files to support H5 UART driver.



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

Add module ibbt-vendor to support Realtek BT Chip.

1. Added files

1) hardware\realtek\bt\libbt

Source code of Realtek Vendor lib is placed here, you need to copy them to the corresponding directory.

2) hardware\realtek\bt\firmware

Files of firmware and configuration of realtek are placed here. You only need to copy them to this directory.

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

Header file for MP tool.

1.3 Bluetooth patches

You must apply all patches below to make Realtek BT chip work normally. For android4.3, it does not support LE HID (HOGP), so realtek will use bluedroid source code of android4.4 to replace with android4.3. All bluetooth patches needed to be applied are in directory "4.4_bluetooth_patches".

Patch directory is named by directory hierarchy in android4.4 SDK, for example:

"external_bluetooth_bluedroid_patches" represents all patches in this directory will be applied to directory

"external/bluetooth/bluedroid/" in android4.4 SDK.

Usually, you will see a file named “.git” in this directory, You need to apply all patches to your SDK (git am *.patch).

If there is no “.git” file in your directory, you need to create git project and add it to your android4.4 SDK. For example: “hardware_realtek_bt_patches” represents all patches in this directory will be applied to directory “*hardware/realtek/bt*”.

This is a module provided by Realtek, there is no git project in your android4.4 SDK, so you need to add it by git.

1.3.1 How to apply patch

Before apply all patches to your SDK, you need to first copy all patches to specified directory, such as patches in directory:” external_bluetooth_bluedroid_patches”, you should copy all patches to directory “external\bluetooth\bluedroid”, and then apply all patches by commands as follows:

```
git am *.patch
```

Absolutely, you can also apply patches by other ways. if there are conflicts when you apply patches, you need to manually merge the files to resolve conflicts.

1.3.2 Patches introduction

1) external_bluetooth_bluedroid_patches

All other patches are used to fix other bugs on bluedroid, you can see details from git log.

2) packages_apps_Bluetooth_patches

Patches here is used to fix some bugs on Bluetooth UI.

3) packages_apps_Settings_patches

Patches here is used to fix some bugs on bluetooth Settings.

1.4 Kernel

1.4.1 TUN driver

```
kernel\arch\arm\configs\XXX_defconfig
```



```
CONFIG_TUN=y
```

1.4.2 UINPUT driver

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

```
CONFIG_INPUT_MISC=y
```

1.4.3 HID driver

```
CONFIG_UHID=y
```

```
CONFIG_HID_xxx=y
```

2 BT function configuratin

2.1 Config supported Profile

For customer do not support PBAP HFP and HSP, use the following configurate, set true to support, set false do not support.

packages/apps/Bluetooth/res/values/config.xml as follows:

```
<resources>
    <bool name="profile_supported_a2dp">true</bool>
    <bool name="profile_supported_hdp">false</bool>
    <bool name="profile_supported_hs_hfp">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="pbap_include_photos_in_vcard">false</bool>
    <bool name="pbap_use_profile_for_owner_vcard">false</bool>
</resources>
```

2.2 Local name, COD and HFP support

Modify BTM_DEF_LOCAL_NAME as platform name to be display.

Modify BTA_DM_COD as platform COD to be display。

In Android4.2 SDK, only support HSP by default, if customer wants to support HFP,

please Modify BTIF_HF_SERVICES and 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
```

2.3 Set bluetooth address in host

Bluetooth address is written in Bluetooth controller, host can also change it by the following steps:

1) add definitions in hardware/realtek/bt/libbt/include/vnd_xxx.txt

USE_CONTROLLER_BDADDR = FALSE

2) Modify init.xxx.rc to set file path which is used as Bluetooth address

Default: setprop ro.bt.bdaddr_path **"/data/misc/bluetooth/bdaddr"**

4) BT address format **00:00:00:AA:BB:CC**

2.4 Configure Extra Config

If you need configure extra config, you need new a file named “rtk_btconfig.txt” in /data/misc/bluetooth/, and modify file permissions to 644. **Please be sure to find FAE to review.** Take rtl8723bs as an example:

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

- 1) The first line must be the name of the config file to be configured.
- 2) Starting from the second line, each line configure one offset and values. Format is offset(2Bytes) + length(1byte) + value(lengthBytes); Little-endian, Separate each byte(Hex) with a space. The setting in the second line of the figure is: offset:0x015b length:0x04 value:0x21222221
- 3) Support for single-line comments with “#”
- 4) Not support configure MAC

3 Basic Function Test after porting finished

3.1 BT basic function test

Notes: This is a fast Bluetooth function test to verify Realtek H5 UART driver has been porting successfully into your platform. The test is only to verify some basic function. You should not take the test result as a formal test report. And if you don't use Realtek BT chip, the test procedure will be no meaningful.

3.1.1 Basic function test

- 1) Turn On/Off BT success.
- 2) Search nearby devices which are discoverable.
- 3) Pair and unpair with device successfully.
- 4) Connect to Bluetooth headset, listen music with A2DP profile.
- 5) Connect to Bluetooth headset, make a call and talk with Bluetooth HFP/HSP.
- 6) Transfer files to remote device which supports OPP server, and transfer files from remote device which supports OPP client to local device.
- 7) Connect Bluetooth HID device (Mouse or Keyboard), Mouse and keyboard can work successfully.

3.1.2 How to capture logs

If you find something abnormal for Bluetooth, you can modify code in file `external\bluetooth\bluedroid\hci\src\hci_h5.c` as follows to turn on bt log.

```
define H5_TRACE_DATA_ENABLE 1

#undef ALOGI
#define ALOGI
...
```

3.1.3 Debug with bluedroid Stack Log

Modify “system/etc/bluetooth/bt_stack.conf” , change Debug Level from 2 to 5, and set BtSnoopLogOutput,

BtSnoopFileName and TraceConf as follows:

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

# BtSnoop log output file
BtSnoopFileName=/sdcard/btsnoop_hci.log//you can also store data to other directory

# 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_AVRCP=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