



---

# **Porting Realtek Bluetooth into Android 6.0 Guide**

Date : 2015/12/29

Version : 3.0

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

**Realtek Bluetooth UART/USB Driver on Android 6.0 Porting Guide**

<b>Date</b>	<b>Version</b>	<b>Description</b>
2015/12/29	V3.0	1. The First version for android 6.0

# CONTENTS

<b>1</b>	<b>START PORTING CODE .....</b>	<b>4</b>
1.1	MODIFIED SDK INTRODUCTION .....	5
1.2	PLATFORM-DEPENDENT MODIFICATION .....	5
1.2.1	Build.....	6
1.2.2	Device .....	7
1.2	BLUETOOTH PATCHES .....	13
1.2.1	How to apply patch .....	13
1.2.2	Patches introduction.....	14
1.3	KERNEL.....	14
1.3.1	Change List .....	14
1.3.2	Rtk_btusb driver.....	15
1.3.3	Rtk_rfkill driver .....	15
1.3.4	TUN driver.....	16
1.3.5	UINPUT driver.....	16
1.3.6	HID driver .....	16
<b>2</b>	<b>BT FUNCTION CONFIGURATIN .....</b>	<b>17</b>
2.1	CONFIG SUPPORTED PROFILE.....	17
2.2	LOCAL NAME, COD AND HFP SUPPORT.....	17
2.3	SET BLUETOOTH ADDRESS IN HOST .....	18
<b>3</b>	<b>BASIC FUNCTION TEST AFTER PORTING FINISHED.....</b>	<b>20</b>
3.1	BT BASIC FUNCTION TEST.....	20
3.1.1	Basic function test.....	20
3.1.2	Debug with bluedroid Stack Log.....	20

## 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
```



## Realtek Bluetooth UART/USB Driver on Android 6.0 Porting Guide

```
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 take {product} platform of {vendor} as an example, you must modify files related to your platform to support Realtek BT Chip.*

#### 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 driver in android 5.0.

**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

New      device/{vendor}/{common}/rtkbt

New      device/{vendor}/{product}/rtkbt

## Realtek Bluetooth UART/USB Driver on Android 6.0 Porting Guide

Chg      device/{**vendor**}/{**product**}/{PRODUCT\_MAKEFILE}

Chg      device/{**vendor**}/{**product**}/init.{product}.rc

Chg      device/{**vendor**}/{**product**}/uevent.{product}.rc

### 3. device

New      system/**bt/hci/include/bt\_list.h** **bt\_skbuff.h** **rtk\_parse.h**

New      system/**bt/hci/include/bt\_list.c** **bt\_skbuff.c** **hci\_h5.c** **rtk\_parse.c**

Chg      system/bt/{**others**}

### 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. And delete or comment out other Macros such as **BOARD\_HAVE\_BLUETOOTH\_BCM**, **BLUETOOTH\_HCI\_USE\_MCT** in case of compiling conflict.

### 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) Copy directory **device/{vendor}/{common}/rtkbt** supplied to **device/{vendor}/{common}/**
- 2) Copy directory **device/{vendor}/{product}/rtkbt** realtek supplied to **device/{vendor}/{product}/**

**Customed** {vendor} and {product} according to your platform and produce in file

**device/vendor/product /rtkbt/conf/rtkbt.conf :**

```
.....  
  
#Indicate USB or UART driver bluetooth default uart ttyS1  
  
#For usb /dev/rtk_btusb // BT USB device node  
  
#For uart /dev/ttyS1 //change it to the actual BT UART ttySx device node  
  
BtDeviceNode=/dev/ttyS1 //change it to the actual BT UART ttySx device node  
  
.....
```

**In practice, adb pull system/etc/bluetooth/rtkbt.conf , push the modified rtk\_conf file according to the above and restart the Bt Settings to fulfill the automatical switch between UART and USB BT Interface.**

- 3) Please check **device/{vendor}/{product}/AndroidProducts.mk** first, find the correct **PRODUCT\_MAKEFILES**, for example:

```
.....  
  
PRODUCT_MAKEFILES := \  
  
$(LOCAL_DIR)/aosp_flo.mk  
  
.....
```



## Realtek Bluetooth UART/USB Driver on Android 6.0 Porting Guide

Add command to call **aosp\_flo.mk**(according to **code/device/vendor/product/rtkbt/aosp\_flo.mk**).

```
.....  
  
# Inherit from the common Open Source product configuration  
  
$(call inherit-product, $(SRC_TARGET_DIR)/product/aosp_base.mk)  
  
#rtkbt  
  
$(call inherit-product, device/{ vendor}/{ product}/rtkbt/rtkbt.mk)  
  
PRODUCT_NAME := aosp_flo  
  
PRODUCT_DEVICE := flo  
  
.....
```

- 4) Add commands to device/{**vendor**}/{**product**}/init.{**product**}.rc

**Attention:** Be sure these commands not in file init.{**product**}.rc. Delete if exists.

/dev/ttyS1 should be configured as the practical Uart interface.

sys/class/rfkill/rfkill0 should be configured as the practical Rfkill device node.

- a) If your product support both USB and UART Bluetooth, copy these commands below(according to **code/device/vendor/product/rtkbt /init.flo.rc**) to device/{**vendor**}/{**product**}/init.{**product**}.rc.

on boot

.....





## Realtek Bluetooth UART/USB Driver on Android 6.0 Porting Guide

```
#RTK BLUETOOTH START
```

```
# Bluetooth
```

```
change back to bluetooth from system
```

```
chown bluetooth net_bt_stack /data/misc/bluetooth
```

```
mkdir /data/misc/bluedroid 0770 bluetooth net_bt_stack
```

```
# 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"
```

```
# UART device
```

```
chmod 0660 /dev/ttyS1
```

```
chown bluetooth net_bt_stack /dev/ttyS1
```

```
# RFKILL
```

```
wait /sys/class/rfkill/rfkill0/state
```

```
chmod 0660 /sys/class/rfkill/rfkill0/state
```

```
wait /sys/class/rfkill/rfkill0/type
```

```
chmod 0660 /sys/class/rfkill/rfkill0/type
```



## Realtek Bluetooth UART/USB Driver on Android 6.0 Porting Guide

```
chown bluetooth net_bt_stack /sys/class/rfkill/rfkill0/state
```

```
chown bluetooth net_bt_stack /sys/class/rfkill/rfkill0/type
```

```
write /sys/class/rfkill/rfkill0/state 0
```

```
# USB device
```

```
insmod /system/lib/modules/rtk_btusb.ko
```

```
wait /dev/rtk_btusb
```

```
chmod 0660 /dev/rtk_btusb
```

```
chown bluetooth net_bt_stack /dev/rtk_btusb
```

```
#RTK BLUETOOTH END
```

- b) If your product using USB Bluetooth, copy these commands below(according to [code/device/vendor/product/rtkbt](#) [/init.flo.rc](#)) to device/{**vendor**}/{**product**}/init.{**product**}.rc

on boot

.....

```
#RTK BLUETOOTH START
```

```
# Bluetooth
```

```
change back to bluetooth from system
```

```
chown bluetooth net_bt_stack /data/misc/bluetooth
```

```
mkdir /data/misc/bluedroid 0770 bluetooth net_bt_stack
```



## Realtek Bluetooth UART/USB Driver on Android 6.0 Porting Guide

```
# 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"
```

```
# USB device
```

```
insmod /system/lib/modules/rtk_btusb.ko
```

```
wait /dev/rtk_btusb
```

```
chmod 0660 /dev/rtk_btusb
```

```
chown bluetooth net_bt_stack /dev/rtk_btusb
```

```
#RTK BLUETOOTH END
```

- c) If your product using UART Bluetooth, copy these commands below(according to [code/device/vendor/product/rtkbt/init.flo.rc](#)) to device/{**vendor**}/{**product**}/init.{**product**}.rc

```
on boot
```

```
.....
```

```
#RTK BLUETOOTH START
```

```
# Bluetooth
```

```
change back to bluetooth from system
```



## Realtek Bluetooth UART/USB Driver on Android 6.0 Porting Guide

```
chown bluetooth net_bt_stack /data/misc/bluetooth
```

```
mkdir /data/misc/bluedroid 0770 bluetooth net_bt_stack
```

```
# 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"
```

```
# UART device
```

```
chmod 0660 /dev/ttyS1
```

```
chown bluetooth net_bt_stack /dev/ttyS1
```

```
# RFKILL
```

```
wait /sys/class/rfkill/rfkill0/state
```

```
chmod 0660 /sys/class/rfkill/rfkill0/state
```

```
wait /sys/class/rfkill/rfkill0/type
```

```
chmod 0660 /sys/class/rfkill/rfkill0/type
```

```
chown bluetooth net_bt_stack /sys/class/rfkill/rfkill0/state
```

```
chown bluetooth net_bt_stack /sys/class/rfkill/rfkill0/type
```

```
write /sys/class/rfkill/rfkill0/state 0
```

```
#RTK BLUETOOTH END
```

## 5) **device/{vendor}/{product}/ueventd.{product}.rc**

Attention: **Only USB Bluetooth** can use it.

Add a command to this file(according to [code/device/vendor/product/rtkbt/ueventd.flo.rc](#)).

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

## 1.2 Bluetooth patches

You must apply all patched below to make Realtek BT chip work normally. All bluetooth patches needed to be applied are in directory “**system\_bt\_patches**”.

Patch directory is named by directory hierarchy in android6.0 SDK, for example: “**system\_bt\_patches**” represents all patches in this directory will be applied to directory “**system\_bt\_patches**” in android6.0 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 android6.0 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 android6.0 SDK, so you need to add it by git.

### 1.2.1 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: “**system\_bt\_patches**”, you should copy all patches to directory “**system\_bt\_patches**”, 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



## Realtek Bluetooth UART/USB Driver on Android 6.0 Porting Guide

merge the files to resolve conflicts. Pay attention that all the file as blow are the added files of realtek.

**system/bt/hci/include/bt\_list.h bt\_skbuff.h rtk\_parse.h**

**system/bt/hci/include/bt\_list.c bt\_skbuff.c hci\_h5.c rtk\_parse.c**

while others in system/bt/ dir as blow are the modified file of Android6.0,where the patches of realtek are seperated by the maros as BLUETOOTH\_RTK and BLUETOOTH\_RTK\_COEX :

system/bt/{others}

### 1.2.2 Patches introduction

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

#### 1) packages\_apps\_Bluetooth\_patches

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

#### 2) packages\_apps\_Settings\_patches

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

## 1.3 Kernel

### 1.3.1 Change List

Chg kernel/arch/arm/configs/xxx\_defconfig

Chg kernel/driver/bluetooth/Kconfig

Chg kernel/driver/bluetooth/Makefile

New kernel/driver/bluetooth/rtk\_btusb.c

New kernel/driver/bluetooth/tk\_btusb.h

### 1.3.2 Rtk\_btusb driver

1. Cp Realtek **rtk\_btusb.h** 和 **rtk\_btusb.c** to **kernel/drivers/bluetooth/**;
2. Modify **/kernel/drivers/bluetooth/** “Kconfig” and “Makefile”

Add BT\_RTKBTUSB in Kconfig :

```
config BT_RTKBTUSB

    tristate "RTK HCI USB driver"

    depends on USB

    help

        RTK Bluetooth HCI USB driver
```

Add rtk\_btusb.o in Makefile:

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

3. Select rtk\_btusb driver in make menuconfig;

### 1.3.3 Rtk\_rfkill driver

#### **Notes:**

**Generally ,the platform manufactures have already fulfilled the rfkill dirver, You may just apply for the RFKILL\_TYPE\_BLUETOOTH rfkill node for Bluetooth which configure the GPIO to the**

**RESET\_PIN of BT Module. Please kindly contact us if any question.**

#### **1.3.4 TUN driver**

kernel\arch\arm\configs\XXX\_defconfig

```
CONFIG_TUN=y
```

#### **1.3.5 UINPUT driver**

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

#### **1.3.6 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_a2dp_sink">true</bool>
    <bool name="profile_supported_hdp">true</bool>
    <bool name="profile_supported_hs_hfp">true</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">fasle</bool>
    <bool name="profile_supported_gatt">true</bool>
    <bool name="pbap_include_photos_in_vcard">true</bool>
    <bool name="pbap_use_profile_for_owner_vcard">true</bool>
    <bool name="profile_supported_map">true</bool>
    <bool name="profile_supported_avrcp_controller">>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.

Bluedroid can set different support for HSP/HFP, customers could set it by what you need.

1. Supportting HSP/HFP, bluedroid will use HFP as default. Defined in bdroid\_buildcfg.h:

## Realtek Bluetooth UART/USB Driver on Android 6.0 Porting Guide

```
#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. Only support HSP, define it in bdroid\_buildcfg.h:

```
#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 }

#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

**Realtek Bluetooth UART/USB Driver on Android 6.0 Porting Guide**

**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"**

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

### **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 Debug with bluebird Stack Log**

Modify “system/etc/bluetooth/bt\_stack.conf” , change Debug Level from 2 to 6, and set BtSnoopLogOutput,

If H5 UART Driver Log needed, set H5LogOutput.Setting BtSnoopFileName, TraceConf as follows:

```
# Enable BtSnoop logging function
# valid value : true, false
BtSnoopLogOutput=true
# valid value : true, false
H5LogOutput= false
```



## Realtek Bluetooth UART/USB Driver on Android 6.0 Porting Guide

```
# 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
```