REALTEK

Porting Realtek Bluetooth USB driver into Android 5.0 Guide

Realtek Confidential

Date: 2015/01/06

Version: 1.1

version	date	author	description
1.0	2015/01/05	Mike	Basic architecture of USB driver and SDK (migration from Android 4.4)

Realtek Confidential

Content

1.Start Porting Code

1. All the code modified in this material follows two forms below:
   a. All the code modified are embraced by square box.
   b. Highlight the code modified in the box by color GREY.

2. Example

   Code below are raw code from SDK:

```
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
```

Our solution needs support H5, so we add code related H5 below. Code grey is modified or added by Realtek.

```
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
```

3. Porting demo

This material based on platform platformName of companyName. Different customers use different platforms, please pay attention to these files modified in different platforms when porting.

1.1 Porting introduction

In order to integrate WIFI/BT Combo Chip to customer's platform, we divided porting steps to two parts: modified SDK and patches supplied by Realtek.

1.2 Modified SDK

We modify or add these files to support Realtek WIFI/BT Combo Chip:

Chg: means modified under raw SDK.

New: means added file to raw SDK

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

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

3. **hardware**

New   hardware\realtek*

4. **kernel**

Chg   kernal\arch\arm\configs\ tegra11_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

modified 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 Realtek BT Chip macro definition supported.

1.2.2 device

This directory used to configure board, different child directory used for different hardware platform.

1.   Modified files

    a.   Copy file \bluetooth\bdroid_buildcfg.h to directory **device/{vendor}/{platform}/bluetooth/**

    b.   **Modified file** 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
```

Set BOARD_HAVE_BLUETOOTH_RTK true, for supporting Realtek BT chip.

c. **device\companyName\platformName\platformName.mk**

add commands in file platformName.mk used for copying firmware to destination directory.

```
#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)

#realtek add end
```

If the customer want support LE, make sure the configure below.

```
#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
```

Please make sure directory frameworks/native/data/etc has file android.hardware.bluetooth.xml and

android.hardware.bluetooth_le.xml, if not , please copy two files from

\device\companyName\platformName\.

d. **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
      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"
```

e. modified **ueventd.platformName.rc under directory device\companyName\platformName.**

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

f. modified **device\companyName\platformName\sepolicy\file_contexts**

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

1.2.3 hardware

copy directory /hardware/realtek from SVN to /hardware.

Add new file named vnd_{platform}.txt(see details below) under directory /hardware/realtek/bt/libbt-vendor/include/

```
BLUETOOTH_UART_DEVICE_PORT = "/dev/rtk_btusb"

BTVND_DBG = FALSE
```

1.3 Kernel

1.3.1 Add rtk_btusb driver

Copy the **rtk_btusb.h & rtk_btusb.c** file we supplied to the directory **kernel/drivers/bluetooth/**.

Then modify the "Kconfig" and "Makefile" files in the same directory as follows:

Add the following contents in **Kconfig**：

```
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

Add the following contents in **Makefile**:

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

1.3.2 Add uid driver (used by Bluedroid Stack for Bluetooth HID)

Copy the **uhid.c** file we supplied to the directory **kernel/drivers/hid/**

Copy the **uhid.h** file we supplied to the directory **kernel/include/linux/**

Then modify the "Kconfig" and "Makefile" files in the same directory as follows:

Add the following contents in **Kconfig**：

```
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
```

Add the following contents in **Makefile**:

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

1.3.3 Configuration

Modify the "platform configuration" file in the directory **kernel/arch/arm/configs/** to support Bluetooth USB. The "platform configuration" file is the file you can use to compile kernel. If there is no this file in your platform, you can modify the ".config" file directly. Using command "make menuconfig" to modify the options is fine too. You can choose any method you like to do this. No matter which method you would like to choose, the following options should be set as y.

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

To support more BT HID devices, we advice to open all "Special HID driver" options, the sample file is located in **kernel/hid_defconfig**

```
CONFIG_HID_A4TECH=y
CONFIG_HID_ACRUX=y
CONFIG_HID_ACRUX_FF=y
CONFIG_HID_APPLE=y
CONFIG_HID_BELKIN=y
…
```

1.3.4 PAN support

Modify the "platform configuration" file in the directory **kernel/arch/arm/configs/** to support PAN. The following options should be set as y.

```
CONFIG_TUN=y
```

1.3.5 AVRCP support

Modify the "platform configuration" file in the directory **kernel/arch/arm/configs/** to support AVRCP. The following options should be set as y.

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

At this time, you can make the Linux kernel.

1.3.6 Selective suspend support

This feature can be used on Linux kernel version later than 2.6.33   .It is disabled in driver by default. Make sure the OS support this function.

Change BTUSB_RPM defines in **kernel/drivers/bluetooth/rtk_btusb.h** to enable this function.

```
#define BTUSB_RPM          1 * USB_RPM
```

1.4 Bluetooth patches

In order to support Realtek BT Chip works well in android5.0, you should integrate these patches.

Patches are under directory 4.4_bluetooth_usb_patches.

external_bluetooth_bluedroid_patches	2014/1/3 16:52	文件夹
packages_apps_Bluetooth_patches	2014/1/3 16:52	文件夹
packages_apps_Settings_patches	2014/1/3 16:52	文件夹

external_bluetooth_bluedroid_patches means these patcher under this directory should integrate to directory *external/bluetooth/bluedroid.*

1.4.1 how to integrate patches

For example, all patches under external_bluetooth_bluedroid_patches, first copy to external\bluetooth\bluedroid, then use command **git am *.patch** to integrate all patches.

0002-Realtek-BT-Modify-PRELOAD_MAX_RETRY_ATTEMPTS-para...	2013/12/23 19:18	PATCH 文件	1 KB
0003-Realtek-BT-BT-open-fail-workaround.patch	2013/12/23 19:18	PATCH 文件	1 KB
0004-Realtek-BT-Modify-BTM_EXT_RMT_NAME_TIMEOUT.patch	2013/12/23 19:18	PATCH 文件	1 KB
0005-Realtek-BT-Fix-inquiry-too-long.patch	2013/12/23 19:18	PATCH 文件	3 KB
0006-Realtek-BT-Restart-BT-when-patch-download-fail.patch	2013/12/23 19:18	PATCH 文件	2 KB
0007-Realtek-BT-Fix-pairing-hid-hang-bug-when-connect-le-.patch	2013/12/23 19:18	PATCH 文件	3 KB
0008-Realtek-BT-Save-device-name-change-information.patch	2013/12/23 19:18	PATCH 文件	1 KB
0009-Realtek-BT-Disable-inquiry-when-pair-is-on-going.patch	2013/12/23 19:18	PATCH 文件	2 KB
0010-Realtek-BT-Pair-close-hid-fail.patch	2013/12/23 19:18	PATCH 文件	4 KB
0011-Realtek-BT-specific_uuidList_for_MSI_FS300.patch	2013/12/23 19:18	PATCH 文件	3 KB
0012-Realtek-BT-Add-protection-to-gki-module.patch	2013/12/23 19:18	PATCH 文件	8 KB
0013-Realtek-BT-Add-LE-HID-HOGP-support-for-bluedroid.patch	2013/12/23 19:18	PATCH 文件	61 KB

1.4.2 patch information

For example, patch 0013-Add-LE-HID-HOGP-support-for-bluedroid.patch is used for bug in Bluedroid LE HOGP. The rest patches in the directory is to fix other bugs of bluedroid. You can check by command **git commit log**.

2. BT function configure(Optional)

2.1 profile configure

Some platform need not support PBAP, HFP, HSP, you can choose to shut them down.

Modify file packages/apps/Bluetooth/res/values/config.xml

```
<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>
```

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:

   #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
```

3. Basic test after porting

3.1 BT basic function test

Notes: This is a fast Bluetooth function test to verify Realtek BT_USB 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

1. adb pull system/etc/bluetooth/bt_stack.conf

2. modify bt_stack.conf, change Debug Level from 2 to 5, set BtSnoopLogOutput true.

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


# 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
```

3. adb push bt_stack.conf system/etc/bluetooth/

4. through UI(Setting/bluetooth/) on bluetooth