

**Realtek Wi-Fi SDK for Android KK 4.4**  
**ver. 1.1.0**

**Contents**

<b>Release History .....</b>	<b>2</b>
<b>Introduction .....</b>	<b>3</b>
<b>1. Copy Necessary Files into SDK .....</b>	<b>4</b>
<b>2. Platform Related Files .....</b>	<b>4</b>
<b>2.1. BoardConfig.mk .....</b>	<b>4</b>
<b>2.2. init.xxx.rc .....</b>	<b>6</b>
<b>2.3. Others .....</b>	<b>8</b>
<b>3. System Resource Configurations .....</b>	<b>9</b>
<b>4. libhardware_legacy .....</b>	<b>10</b>
<b>5. wpa_supplicant_8 .....</b>	<b>11</b>
<b>6. Driver Configurations for Android KK .....</b>	<b>12</b>
<b>7. FAQ .....</b>	<b>14</b>
<b>7.1. Wi-Fi (STA mode) .....</b>	<b>14</b>
<b>7.1.1. Why Wi-Fi can't enable? .....</b>	<b>14</b>
<b>7.2. Portable Wi-Fi hotspot (AP mode) .....</b>	<b>14</b>
<b>7.2.1. Why Portable Wi-Fi hotspot can't enable? .....</b>	<b>14</b>
<b>7.3. Wi-Fi Direct (P2P mode) .....</b>	<b>15</b>
<b>7.3.1. There is no Wi-Fi Direct UI shown? .....</b>	<b>15</b>
<b>7.3.2. Wi-Fi Direct can't scan any peer? .....</b>	<b>15</b>

## Release History

0.0.1	2013/11/22	1. Beta release 1.1. realtek_wifi_SDK_for_android_KK_4.4_20131122.tar.gz 1.2. wpa_supplicant_8_kk_4.4_rtw_r9640.20131122.tar.gz
1.0.0	2014/01/17	1. First formal release 1.1. realtek_wifi_SDK_for_android_KK_4.4_20140117.tar.gz
1.1.0	2014/02/20	1. wpa_supplicant_8_kk_4.4_rtw_r10450.20140220.tar.gz

## SDK packages

- hardware/realtek/\*

Folder to store config files, private code from Realtek.

- hardware/libhardware\_legacy/wifi/Android.mk

Reference codes for applying wifi\_realtek.c

※ For wpa\_supplicant\_8\_kk\_4.4\_rtw\_r10450.20140220.tar.gz or newer version, see wpa\_supplicant\_hostapd folder of our SW release package or consult our contact window.

## Introduction

This document provides a simple guide to help engineers to apply Realtek Wi-Fi solution onto their Android KK 4.4 system. For now, we have supported the following two scenarios:

- **STA/AP** – Switch between STA mode and AP mode
- **(STA+P2P)/AP** – Switch between STA+P2P(Wi-Fi Direct) concurrent mode and AP mode

To port Realtek Wi-Fi driver onto Android 4.4 platform, you can go through the following guide with reference codes within our driver package's realtek\_wifi\_SDK\_for\_android\_KK\_4.4\_20140117.tar.gz.

Because Android's SDK may differ from platform to platform, our reference codes may not be applied on every platform without modifications. You should check if our reference code is suitable for you to use.

In this document, ANDROID\_SDK is the path of top folder of the target Android SDK; this term is used in the following paragraphs.

## **1. Copy Necessary Files into SDK**

After unzipping `realtek_wifi_SDK_for_android_KK_4.4_20140117.tar.gz`, copy the following folder into `ANDROID_SDK/hardware/` folder:

- `hardware/realtek`

## **2. Platform Related Files**

### **2.1. BoardConfig.mk**

To apply Realtek Wi-Fi solution onto your Android KK system, you need to define some compile-time variables in `BoardConfig.mk` of your platform. In general, the `BoardConfig.mk` file is located in:

`ANDROID_SDK /device/<soc_vendor_name>/<board_name>/`

Take TI panda board for example:

`ANDROID_SDK /device/ti/panda/ BoardConfig.mk`

Please define the following compile-time variables below:

```

BOARD_WIFI_VENDOR := realtek
ifeq ($(BOARD_WIFI_VENDOR), realtek)
    WPA_SUPPLICANT_VERSION := VER_0_8_X
    BOARD_WPA_SUPPLICANT_DRIVER := NL80211
    CONFIG_DRIVER_WEXT :=y
    BOARD_WPA_SUPPLICANT_PRIVATE_LIB := lib_driver_cmd_rtl
    BOARD_HOSTAPD_DRIVER      := NL80211
    BOARD_HOSTAPD_PRIVATE_LIB := lib_driver_cmd_rtl

    BOARD_WLAN_DEVICE := rtl8192cu
    #BOARD_WLAN_DEVICE := rtl8192du
    #BOARD_WLAN_DEVICE := rtl8192ce
    #BOARD_WLAN_DEVICE := rtl8192de
    #BOARD_WLAN_DEVICE := rtl8723as
    #BOARD_WLAN_DEVICE := rtl8723au
    #BOARD_WLAN_DEVICE := rtl8189es
    #BOARD_WLAN_DEVICE := rtl8723bs
    #BOARD_WLAN_DEVICE := rtl8723bu

    WIFI_DRIVER_MODULE_NAME   := "wlan"
    WIFI_DRIVER_MODULE_PATH   := "/system/lib/modules/wlan.ko"
    WIFI_DRIVER_MODULE_ARG    := "ifname=wlan0 if2name=p2p0"

    WIFI_FIRMWARE_LOADER      := ""
    WIFI_DRIVER_FW_PATH_STA   := ""
    WIFI_DRIVER_FW_PATH_AP    := ""
    WIFI_DRIVER_FW_PATH_P2P   := ""
    WIFI_DRIVER_FW_PATH_PARAM := ""
endif

```

- **BOARD\_WIFI\_VENDOR := realtek**

To distinguish the platform Wi-Fi device from products of other vendors, we define variable BOARD\_WIFI\_VENDOR as realtek. This is for compile-time choices to be applied for Realtek Wi-Fi solutions.

- **WPA\_SUPPLICANT\_VERSION := VER\_0\_8\_X**

For Android KK, please set WPA\_SUPPLICANT\_VERSION as VER\_0\_8\_X to

use wpa\_supplicant\_8.

- **BOARD\_WPA\_SUPPLICANT\_DRIVER := NL80211**
- **BOARD\_WPA\_SUPPLICANT\_PRIVATE\_LIB := lib\_driver\_cmd\_rtl**
- **BOARD\_HOSTAPD\_DRIVER := NL80211**
- **BOARD\_HOSTAPD\_PRIVATE\_LIB := lib\_driver\_cmd\_rtl**

We use NL80211 as the driver interface for wpa\_supplicant and hostapd to communicate with driver and provide lib\_driver\_cmd\_rtl as the private library.

- **BOARD\_WLAN\_DEVICE**

Realtek provide a variety of Wi-Fi solutions to choose. For now, BOARD\_WLAN\_DEVICE is not used for any purpose but we suggest setting this variable for your Wi-Fi solution you used.

- **WIFI\_DRIVER\_MODULE\_NAME**
- **WIFI\_DRIVER\_MODULE\_PATH**
- **WIFI\_DRIVER\_MODULE\_ARG**

These three variables will be used in libhardware\_legacy (wifi.c/wifi\_realtek.c) to do insmod and remmod. The value of WIFI\_DRIVER\_MODULE\_NAME should match the value of MODULE\_NAME specified in our driver's Makefile at compile-time. Please refer to "Platform Setting Section in Detail" of:

document/Quick\_Start\_Guide\_for\_Driver\_Compilation\_and\_Installation.pdf

- **WIFI\_FIRMWARE\_LOADER := ""**
- **WIFI\_DRIVER\_FW\_PATH\_STA := ""**
- **WIFI\_DRIVER\_FW\_PATH\_AP := ""**
- **WIFI\_DRIVER\_FW\_PATH\_P2P := ""**
- **WIFI\_DRIVER\_FW\_PATH\_PARAM := ""**

Because our driver has FW embedded inside, and will automatically load FW at NIC initialization process, there is no need to set these 5 variables, just keep them empty.

## 2.2. init.xxx.rc

For Wi-Fi to operate properly, we need some daemons to be defined as service inside init.xxx.rc. In general, the init.xxx.rc file is located in:

ANDROID\_SDK/device/<soc\_vendor\_name>/<board\_name>/

Take TI panda board for example:

ANDROID\_SDK/device/ti/panda/init.omap4pandaboard.rc.

Please add the service definitions below:

- **wpa\_supplicant**

```
service rtw_suppl_con /system/bin/wpa_supplicant \  
-ip2p0 -Dnl80211 -c/data/misc/wifi/p2p_supplicant.conf \  
-e/data/misc/wifi/entropy.bin -N \  
-iwlan0 -Dnl80211 -c/data/misc/wifi/wpa_supplicant.conf \  
-O/data/misc/wifi/sockets \  
-g@android:wpa_wlan0  
class main  
socket wpa_wlan0 dgram 660 wifi wifi  
disabled  
oneshot  
  
service rtw_suppl /system/bin/wpa_supplicant \  
-iwlan0 -Dnl80211 -c/data/misc/wifi/wpa_supplicant.conf \  
-O/data/misc/wifi/sockets \  
-e/data/misc/wifi/entropy.bin  
-g@android:wpa_wlan0  
class main  
socket wpa_wlan0 dgram 660 wifi wifi  
disabled  
oneshot
```

- **dhcpcd**

```
service dhcpcd_wlan0 /system/bin/dhcpcd -aABDKL
    class main
    disabled
    oneshot

service dhcpcd_p2p /system/bin/dhcpcd -aABKL
    class main
    disabled
    oneshot

service iprenw_wlan0 /system/bin/dhcpcd -n
    class main
    disabled
    oneshot

service iprenw_p2p /system/bin/dhcpcd -n
    class main
    disabled
    oneshot
```

### 2.3. Others

For topics mentioned here, you can add the following code segments in any .mk file which your platform will use. Take TI panda board for example:

ANDROID\_SDK /device/ ti/panda/device.mk.

- **Add android.hardware.wifi.xml**

To claim Wi-Fi support for you device, please add the rule in the PRODUCT\_COPY\_FILES variable to copy the permission definition file of Wi-Fi to the /system/etc/permissions/ folder of your system image.

```
PRODUCT_COPY_FILES += \
frameworks/native/data/etc/android.hardware.wifi.xml:system/etc/permissions/android.hardware.
wifi.xml
```

- **Add android.hardware.wifi.direct.xml**

To claim Wi-Fi Direct (P2P) support for your device, please add the rule in the

PRODUCT\_COPY\_FILES variable to copy the permission definition file of Wi-Fi Direct to the /system/etc/permissions/ folder of your system image.

```
PRODUCT_COPY_FILES += \
frameworks/native/data/etc/android.hardware.wifi.direct.xml:system/etc/permissions/android.hardware.wifi.direct.xml
```

Make sure your driver is configured for STA+P2P concurrent mode or you may encounter error when you open the Wi-Fi. Please refer to **“7. Driver Configurations for Android KK”**

- **Set wifi.interface**

To specify the wifi interface name in Android, a system property named “wifi.interface” is used. For Realtek Wi-Fi driver, Wi-Fi interface name is assigned with “wlan%d”. In general, you should set wifi.interface as “wlan0”.

```
PRODUCT_PROPERTY_OVERRIDES += \
wifi.interface=wlan0
```

### 3. System Resource Configurations

You should set the following four resource configurations for your platform to configure the network function and enable the corresponding UI interface. In general, you can set the following configurations in your platform dependent config.xml file. Take TI panda board for example:

ANDROID\_SDK/device/ti/panda/overlay/frameworks/base/core/res/res/values/config.xml

Or the global config.xml file:

ANDROID\_SDK/frameworks/base/core/res/res/values/config.xml

- **networkAttributes**

To define the system’s available network interfaces, make sure the wifi and wifi\_p2p interface items is defined in the networkAttributes resource configuration in the config.xml. For example:

```
<string-array translatable="false" name="networkAttributes">
  <item>"wifi,1,1,1,-1,true"</item>
  <item>"bluetooth,7,7,0,-1,true"</item>
  <item>"ethernet,9,9,2,-1,true"</item>
</string-array>
```

- **radioAttributes**

To define the system's available network interfaces, we need to define interface items for wifi in the radioAttributes resource configuration. For example:

```
<string-array translatable="false" name="radioAttributes">
    <item>"1,1"</item>
    <item>"7,1"</item>
    <item>"9,1"</item>
</string-array>
```

- **config\_tether\_wifi\_regexs**

The interfaces set here are tetherable Wi-Fi interfaces which will be used as interfaces for Wi-Fi LAN port. We use 'wlan0' by default when our Wi-Fi is set as softap mode. So it needs to set 'wlan0' here. For example:

```
<string-array translatable="false" name="config_tether_wifi_regexs">
    <item>"wlan0"</item>
</string-array>
```

- **config\_tether\_upstream\_types**

The connection types set here are used as the interfaces for WAN port to connect to internet. For example, adding Wi-Fi and Ethernet:

```
<integer-array translatable="false" name="config_tether_upstream_types">
    <item>1</item>
    <item>9</item>
</integer-array>
```

At least one item should be declared here to enable the “Tethering&portable hotspot” option of WirelessSettings in Settings.apk.

To know the definition and set other upstream connection types, please refer to ANDROID\_SDK/frameworks/base/core/java/android/net/ConnectivityManager.java.

#### 4. **libhardware\_legacy**

The libhardware\_legacy library includes functionality for Wi-Fi to operate. We have made modifications and extensions for our Wi-Fi solutions. To apply this, please modify ANDROID\_SDK/hardware/libhardware\_legacy/wifi/Android.mk to include

wifi\_realtek.c instead of wifi.c into LOCAL\_SRC\_FILES. For example:

```
ifeq ($(BOARD_WIFI_VENDOR), realtek)
LOCAL_SRC_FILES += ../realtek/wlan/libhardware_legacy/wifi/wifi_realtek.c
else
LOCAL_SRC_FILES += wifi/wifi.c
endif
```

## 5. wpa\_supplicant\_8

We provide wpa\_supplicant\_8\_kk\_4.4\_rtw\_r10450.20140220.tar.gz or newer version in the wpa\_supplicant\_hostapd/ of our SW release package. You can:

- **Compare and merge with your own wpa\_supplicant\_8**

Compare and merge from wpa\_supplicant\_8\_kk\_4.4\_rtw by your own. For both ANDROID\_SDK/external/wpa\_supplicant\_8/wpa\_supplicant/Android.mk and ANDROID\_SDK/external/wpa\_supplicant\_8/hostapd/Android.mk, you should notice that the two macros REALTEK\_WIFI\_VENDOR and ANDROID\_P2P should be added into L\_CFLAGS. For example:

```
ifeq ($(BOARD_WLAN_DEVICE), bcmhdhd)
L_CFLAGS += -DANDROID_P2P
endif

ifeq ($(BOARD_WIFI_VENDOR), realtek)
L_CFLAGS += -DREALTEK_WIFI_VENDOR
L_CFLAGS += -DANDROID_P2P
Endif

# Use Android specific directory for control interface sockets
```

Here is the description of the specific macros:

MACRO	Description	
ANDROID_P2P	Android's wpa_supplicant_8 patch.	Must
REALTEK_WIFI_VENDOR	General purpose patch made by Realtek.	Must

- **Use the wpa\_supplicant\_8\_kk\_4.4\_rtw instead of the original**

A. Backup and remove the original external/wpa\_supplicant\_8/ folder

- B. Extract and copy the wpa\_supplicant\_8\_kk\_4.4\_rtw tar file to the external/ folder of your Android SDK.
- C. Rename wpa\_supplicant\_8\_kk\_4.4\_rtw as wpa\_supplicant\_8.

※ We have enabled the two macros ANDROID\_P2P and REALTEK\_WIFI\_VENDOR by default.

## 6. Driver Configurations for Android KK

Android KK support two scenarios for Wi-Fi solution:

- **STA/AP – Switch between STA and AP mode**
- **(STA+P2P)/AP – Switch between STA+P2P concurrent and AP mode**

The configuration of driver to fit the requirement of each scenario, see the following table:

MACRO	STA/AP	(STA+P2P)/AP	Kernel ver.
CONFIG_IOCTL_CFG80211	Defined	Defined	ver. >= 2.6.35
RTW_USE_CFG80211_STA_EVENT	Defined	Defined	ver. >= 3.2.0
CONFIG_CONCURRENT_MODE	Undefined	Defined	-
CONFIG_WFD	Don't Care	Must for Miracast	-
RTW_ENABLE_WIFI_CONTROL_FUNC	Must for platform device/driver mechanism		

- **RTW\_USE\_CFG80211\_STA\_EVENT** is used for driver to indicate new cfg80211 STA event which is required by wpa\_supplicant\_8 of Android 4.4. Linux kernel supports this feature after kernel 3.2. For kernel version between 3.0 and 3.2, please refer to the patch file:

linux-3.0.42\_STATION\_INFO\_ASSOC\_REQ\_IES.diff

- **RTW\_ENABLE\_WIFI\_CONTROL\_FUNC** is used to register platform driver callbacks. If your platform needs those callbacks, please define this macro to register platform driver callback functions. For example, these functions include:

```
static struct platform_driver wifi_device = {
    .probe          = wifi_probe,
    .remove         = wifi_remove,
```

By default, the probe callback is used to set up Wi-Fi power and remove callback is used to close Wi-Fi power.

Please modify both the include/autoconf.h and the specific autoconf file(needed

for compound driver release) for your Wi-Fi product.

Chip type	Autoconf file to modify
RTL8192CU-series	autoconf_rtl8192c_usb_linux.h
RTL8192CE-series	autoconf_rtl8192c_pci_linux.h
RTL8192DU-series	autoconf_rtl8192d_usb_linux.h
RTL8192DE-series	autoconf_rtl8192d_pci_linux.h
RTL8723AS-series	autoconf_rtl8723a_sdio_linux.h
RTL8723AU-series	autoconf_rtl8723a_usb_linux.h
RTL8723BS-series	autoconf_rtl8723b_sdio_linux.h
RTL8189ES-series	autoconf_rtl8189e_sdio_linux.h
RTL8188EU-series	autoconf_rtl8188e_usb_linux.h
RTL8723BS-series	autoconf_rtl8723b_sdio_linux.h
RTL8723BU-series	autoconf_rtl8723b_usb_linux.h

For example, if you want to configure RTL8192CU-series driver (ex: RTL8188CUS, RTL8192CU) to fit the scenario of (STA+P2P)/AP, make sure the macros: CONFIG\_IOCTL\_CFG80211, RTW\_USE\_CFG80211\_STA\_EVENT and CONFIG\_CONCURRENT\_MODE in both include/autoconf.h and autoconf\_rtl8192c\_usb\_linux.h (needed for compound driver release) are defined. As following:

```
#define CONFIG_IOCTL_CFG80211
#ifdef CONFIG_IOCTL_CFG80211
    #define RTW_USE_CFG80211_STA_EVENT
    //define CONFIG_CFG80211_FORCE_COMPATIBLE_2_6_37_UNDER
    //define CONFIG_DEBUG_CFG80211 1
#endif
...
#define CONFIG_CONCURRENT_MODE
...
```

To compile our Wi-Fi driver, please refer to the following document for more informations:

[document/Quick\\_Start\\_Guide\\_for\\_Driver\\_Compilation\\_and\\_Installation.pdf](#)

## 7. FAQ

### 7.1. Wi-Fi (STA mode)

#### 7.1.1. Why Wi-Fi can't enable?

The whole Wi-Fi enabling procedure includes the following three main check points. Please check in sequence:

- **Is network interface(s) created?**
  - insmod driver success
  - Wi-Fi device is recognized
  - wlan0 (and p2p0) is created
- **Does wpa\_supplicant run successfully?**
  - wpa\_supplicant.conf (and p2p\_supplicant.conf) exists and is correct
  - Service definition of wpa\_supplicant exists and is correct
  - Binary file wpa\_supplicant exists and is executable
- **Do connections of communication socket setup?**
  - Make sure the communication socket settings is matched below:
    - ◆ ctrl\_interface in:  
/data/misc/wifi/wpa\_supplicant.conf  
(and /data/misc/wifi/p2p\_supplicant.conf)
    - ◆ Service definition of wpa\_supplicant
    - ◆ Paths of communication socket in wifi.c (or wifi\_realtek.c)

### 7.2. Portable Wi-Fi hotspot (AP mode)

#### 7.2.1. Why Portable Wi-Fi hotspot can't enable?

The whole Portable Wi-Fi hotspot enabling procedure includes the following three main check points. Please check in sequence:

- **Is network interface created?**
  - insmod driver success
  - Wi-Fi device is recognized
  - wlan0 is created
- **Does netd and hostapd run successfully?**
  - /data/misc/wifi/hostapd.conf exists and is correct
  - Binary file netd and hostapd exist and are executable
- **Does dnsmasq run successfully?**
  - Binary file dnsmasq exist and are executable

### **7.3. Wi-Fi Direct (P2P mode)**

#### **7.3.1. There is no Wi-Fi Direct UI shown?**

Please refer to “**Add android.hardware.wifi.direct.xml**” in chapter **2.3. Others** to enable Wi-Fi Direct functionality of Android KK.

#### **7.3.2. Wi-Fi Direct can’t scan any peer?**

First, make sure you have workable Wi-Fi Direct device nearby. Make them into Wi-Fi Direct scanning state. Push “SEARCH FOR DEVICES” button also in our device and wait for a while.

If there is still no peer shown the problem is usually caused by wrong service definition of wpa\_supplicant services. Please refer to “**wpa\_supplicant**” in chapter **2.2. init.xxx.rc** to check your service definition of wpa\_supplicant.