Wi-Fi

# User Guide

**Issue** **05**

**Date** **2018-02-10**

**Copyright © HiSilicon Technologies Co., Ltd. 2014-2018. All rights reserved.**

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of HiSilicon Technologies Co., Ltd.

**Trademarks and Permissions**

 , HISILICON, and other HiSilicon icons are trademarks of HiSilicon Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

**Notice**

The purchased products, services and features are stipulated by the contract made between HiSilicon and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

## HiSilicon Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://www.hisilicon.com

Email: support@hisilicon.com

# About This Document

## Purpose

This document describes the configurations, basic operations, and debugging methods of the Wi-Fi drivers. It also describes the precautions to be taken and solutions to common problems.

**NOTE**

- Unless otherwise stated, Hi3516D and Hi3516A contents are consistent.
- Unless otherwise stated, Hi3518E V201, Hi3516C V200 and Hi3518E V200 contents are consistent.

## Related Version

The following table lists the product version related to this document.

| Product Name | Version |
| --- | --- |
| Hi3516A | V100 |
| Hi3516D | V100 |
| Hi3518E | V200 |
| Hi3518E | V201 |
| Hi3516C | V200 |
| Hi3559A | V100 |
| Hi3559C | V100 |
| Hi3519A | V100 |
| Hi3516E | V200 |
| Hi3516E | V300 |
| Hi3518E | V300 |


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

# Intended Audience

This document is intended for:

- Technical support engineers
- Software development engineers

# Change History

Changes between document issues are cumulative. Therefore, the latest document issue contains all changes made in previous issues.

## Issue 05 (2018-02-10)

This issue is the fifth official release, which incorporates the following changes:

The descriptions of theHi3559A V100 and Hi3559CV100 are added.

Section 1.2 is modified.

Sections 3.1.1, 3.1.2, 3.2.4, and 3.3.1 are modified.

## Issue 04 (2018-01-15)

This issue is the fourth official release, which incorporates the following changes:

The description of the Hi3518E V20X is removed.

The contents in the entire documents are updated.

Chapter 4 is added.

## Issue 03 (2015-09-25)

This issue is the third official release, which incorporates the following changes:

The contents related to the Hi3518EV 200, Hi3518E V201, and Hi3516C V200 are added.

## Issue 02 (2015-02-10)

This issue is the second official release, which incorporates the following changes:

In section 3.3.3, the original step 3 and step 4 are combined.

## Issue 01 (2014-12-30)

This issue is the first official release, which incorporates the following changes:

**Chapter 2 Configurations**

In section 2.1, the position of the section "Configuring Wireless Extension" is changed, the section "Configuring IPv6" is added, and the section "Configuring Netlink" is deleted.

In section 2.2, the sections "Configuring Wi-Fi Drivers" and "Wireless Manager" are deleted.

**Chapter 3 Basic Operations**

Section 3.1 and section 3.4 are added.


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

## Issue 00B01 (2014-11-09)

This issue is the first draft release.


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

# Contents




HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

# Figures




HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

# 1 Overview

## 1.1 Background

The commands and tools required for compiling software packets and compilation steps vary according to Wi-Fi vendors. The working modes supported by Wi-Fi devices are also different. The wifi_project development package allows you to debug the Wi-Fi devices of different models. By using the wifi_project, you can rapidly generate drivers and tools for various Wi-Fi devices, add new Wi-Fi drivers, or delete existing drivers.

Typically, a Wi-Fi device supports one or more working modes as follows:

- Soft access point (SoftAP): a device that is used to connect a wireless device to the network. An AP can be considered as a wireless route.
- Station (STA): a wireless device client. An STA is available only after an AP is connected.
- Direct: Wi-Fi direct connection mode. It is also called P2P mode.
- Concurrent: a mode in which AP and STA modes are supported simultaneously.

For details of the Wi-Fi devices supported by the wifi_project SDK, see **Makefile** in the root directory. The **wifi_project** SDK supports only the SoftAP and STA modes.

## 1.2 Directory

wifi_project

├─Makefile ---------- wifi_project compilation script

├─docs ----------Wi-Fi device description and usage guide documentation

├─firmware ---------- Firmware files for Wi-Fi devices

├─tools ---------- Directory of the source code for operating and configuring Wi-Fi tools

│ ├─hostapd ---------- Source code of the hostapd tool required for the operations in SoftAP mode

│ ├─wireless_tools ---------- Source code of iwlist, iwpriv, and iwconfig for debugging


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

```
│   ├─wpa_supplicant ---------- Source code of the wpa_supplicant tool required for the operations in STA mode
│   ├─libnl ---------- libnl source code and libraries required by wpa_supplicant and hostapd
│   ├─iperf---------- iperf source code
│   ├─----wl ---------- wl tool code used by Broadcom Wi-Fi device
│   └─Makefile ---------- Scripts required for compiling tools
│----sample ----------sample script for enabling the Wi-Fi network
├─drv ---------- Source code of Wi-Fi drivers
    ├─Makefile ---------- Scripts required for compiling drivers
    ├─usb_rtl8188ftv ---------- Source code of Realtek rtl8188ftv
    ├─sdio_ap6xxx ---------- Source code of drivers such as ap6181, ap6212, and ap6255
    ├─usb_rtl8188eus ---------- Source code of the Realtek rtl8188eus driver
    ├─usb_mt7601u ---------- Source code of the MediaTek mt7601u driver
    ├─…… ---------- Source code of other drivers
```


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

# 2 Configurations

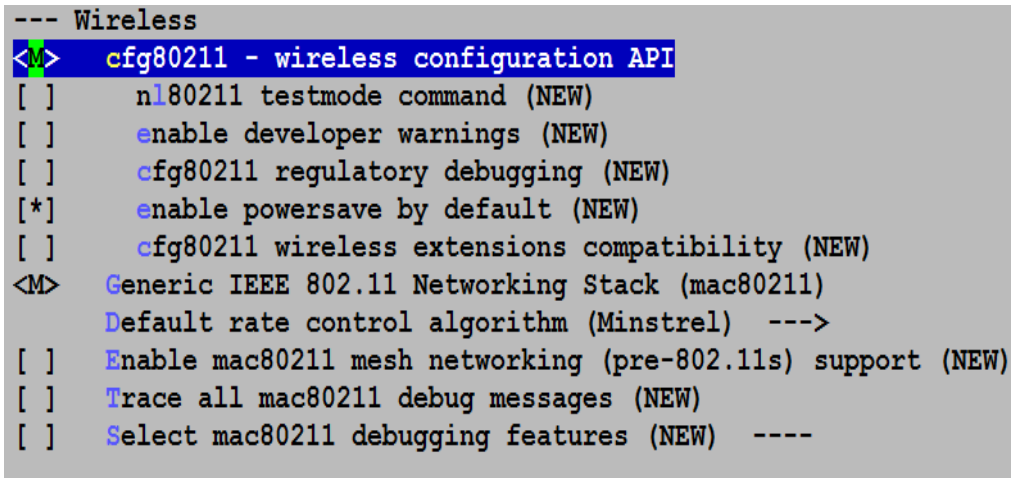
## 2.1 Configuring the Kernel

### 2.1.1 Configuring the CFG80211

CFG80211 is the standard interface for the Wi-Fi drivers in the kernel and user-mode processes. It becomes more popular than WEXT. Only CFG80211 supports the Wi-Fi direct function.

Choose **Network support** > **Wireless**, and set **cfg80211** and **mac80211** to **M**, as shown in Figure 2-1.

**Figure 2-1** CFG80211 configuration



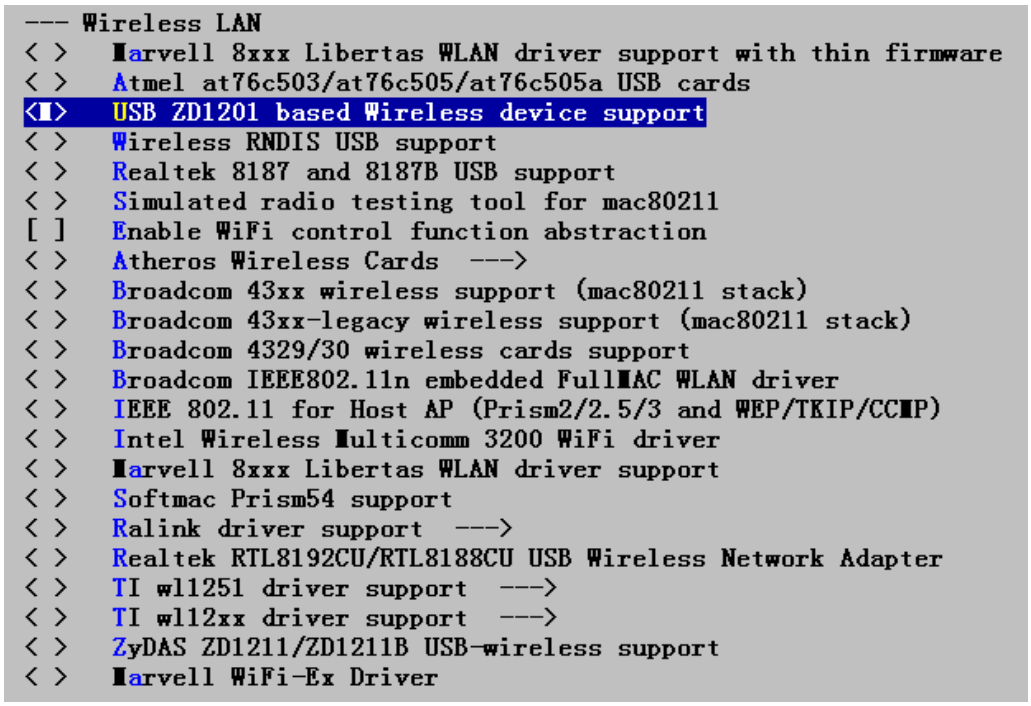

### 2.1.2 Configuring Wireless Extension

- Wireless extension (WEXT) is the standard interface for the Wi-Fi drivers in the kernel and user-mode processes. The debugging tools iwconfig, iwlist, and iwpriv need to use this interface. If this interface is not configured, errors occur when some drivers are compiled.
- Because there is no separate configuration item for the WEXT in the kernel configuration, you can configure the WEXT only by configuring the item that is


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

dependent on the WEXT. To be specific, choose **Device Drivers** > **Network device support** > **Wireless LAN**, and set **USB ZD1201 based Wireless device support** to **M**, as shown in Figure 2-2.

**Figure 2-2** Configuring wireless extension




If the configuration item cannot be found, configure the USB first.

## 2.1.3 Configuring the USB and SDIO

For details about the USB and SDIO operations, see the *Peripheral Driver Operation Guide*.

- The ap6181, ap6212, ap6212a, and ap6214a drivers use the SDIO 2.0 interface. For these drivers, you need to set the SDIO clock to approximately 50 MHz. The ap6255 driver uses the SDIO 3.0 interface. For this driver, you need to set the SDIO clock to approximately 150 MHz.
- The default I/O voltage of an SDIO 2.0 interface driver in the kernel is 3.3 V. If the I/O voltage provided to the Wi-Fi module is 1.8 V, the I/O voltage of the SDIO drivers needs to be changed into 1.8 V.

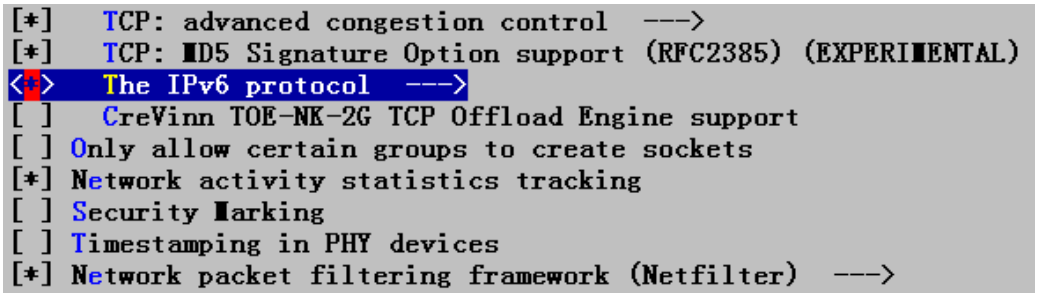
## 2.1.4 Configuring IPv6

IPv6 must be compiled for the ap6181, ap6212, ap6212a, ap6214a, and ap6255 drivers; otherwise, an error occurs when the ap6181 driver is running. However, IPv6 does not need to be configured for other Wi-Fi drivers.

To configure IPv6, choose **Network support** > **Networking options**, and set **The IPv6 protocol** to **y**, as shown in Figure 2-3.


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

**Figure 2-3** Configuring IPv6




## 2.1.5 Configuring the SDIO Interrupt

The SDIO interrupt is disabled by default in the kernel. The SDIO interrupt needs to be enabled if the Wi-Fi driver used does not support out-of-band (OOB) management. Enable it by adding **cap-sdio-irq** to the attribute of the SDIO port which is connected to the Wi-Fi in the **arch/arm/boot/dts/hi3516a.dtsi** configuration file.

**CAUTION**

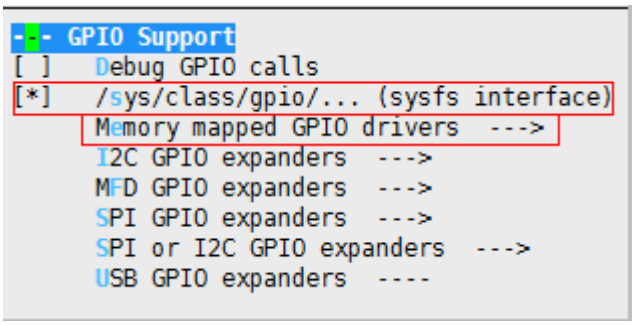
After the kernel is configured and compiled, the Wi-Fi driver must be recompiled based on the new kernel; otherwise, the pointer is null or the kernel symbol cannot be found when the driver is running.

## 2.1.6 Configuring the GPIO

**Step 1** Open the GPIO configuration window.

**Step 2** Go to **Device Drivers**, enable **GPIO Support**, and then go to **GPIO Support**, as shown in Figure 2-4.

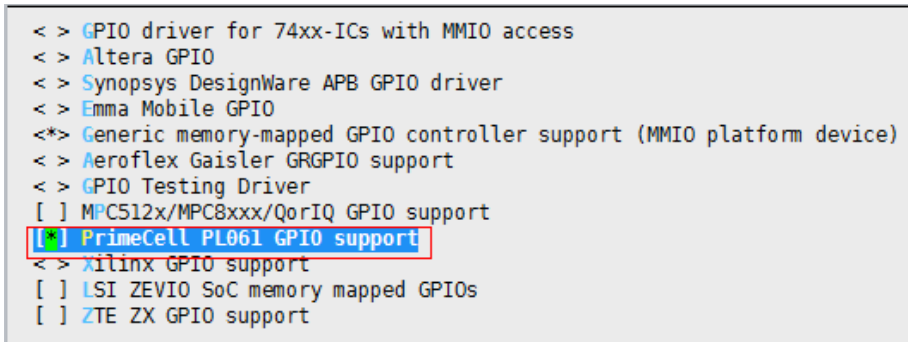
**Figure 2-4** Configuring the GPIO A




**Step 3** Go to the memory-mapped GPIO drivers and configure the GPIO as shown in Figure 2-5.


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

**Figure 2-5** Configuring the GPIO B




**----End**

# 2.2 Configuring the wifi_project

Before configuring the wifi_project, configure the cross compilation environment and modify **WIFI_DEVICE**, **CROSS_COMPILE**, and **KERNEL** in **Makefile** at the top layer.

- **CROSS_COMPILE**

  Cross compilation tool required for compilation, such as arm-hisiv500-linux- and arm-hisiv600-linux-.

  For example: ARCH := arm

  CROSS_COMPILE := arm-hisiv500-linux-

- **WIFI_DEVICE**

  Wi-Fi device to be compiled. For details about supported Wi-Fi devices, see the comments in **Makefile**.

  For example: WIFI_DEVICE := sdio_ap6xxx

  **WIFI_DEVICE** must be set to **sdio_ap6xxx** for the ap6181, ap6212, ap6212a, ap6214a, and ap6225 drivers.

- **KERNEL**

  Kernel path specified when you compile a Wi-Fi driver. Note that the kernel path must be specified when you compile the Wi-Fi driver and the kernel must have been compiled.

  For example: KERNEL:= /home/work/linux-3.18.y

After the preceding configurations, you can configure the wifi_project by running **make all** in the **wifi_project** directory. Then the drivers and tools are automatically compiled. Running **make driver** compiles only drivers, and running **make tools** compiles only tools.

The generated drivers are stored in **wifi_project/out/kmod**. The tools including iwconfig, iwlist, iwpriv, wpa_cli, wpa_supplicant, and hostapd are stored in **wifi_project/out/tools**. The generated library file is stored in the **wifi_project/out/lib** directory.


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

# 3 Basic Operations

## 3.1 Loading Files

### 3.1.1 Loading Driver Files

After compilation, drivers are generated in **wifi_project/out/kmod**. Copy the required drivers to the board.

The Wi-Fi driver files include:

- mt7601u

  **cfg80211.ko**, **mtprealloc.ko**, and **mt7601Usta.ko**
- rtl8188ftv

  **cfg80211.ko** and **8188fu.ko**
- rtl8188eus

  **cfg80211.ko** and **8188eu.ko**
- rtl8189ftv

  **cfg80211.ko** and **8189fs.ko**
- ap6181/ap6212/ap6212a/ap6214a/ap6255

  **cfg80211.ko** and **bcmdhd.ko**

The **cfg80211.ko** file is in **net/wireless** of the kernel.

**NOTE**

The driver files can be copied to any directory of the board (such as **/kmod**).

### 3.1.2 Loading Firmware Files

- To use Broadcom ap6181, create the **/etc/firmware**/sdio_ap6181 directory on the board, and copy **fw_bcm40181a2.bin**, **fw_bcm40181a2_apsta.bin** and **nvram.txt** in **/wifi_project/firmware** to this directory. For the ap6212/ap6212a/ap6214a/ap6255 driver, copy the firmware and nvram files in the **wifi_project/firmware** directory.
- To use MediaTek mt7601u, create the **/etc/Wireless/RT2870STA** directory on the board, and copy **MT7601USTA.dat** in **/wifi_project/firmware/usb_mt7601u** to this directory.
- For RealTek, no additional firmware needs to be loaded.


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

## 3.1.3 Loading Tools

- Copy the **libnl-genl.so.2.0.0** and **libnl.so.2.0.0** files in **wifi_project/out/lib** to the **/lib** directory of the board, and create the soft links to the two files in the **/lib** directory:

  ```
  ln -s libnl-genl.so.2.0.0 libnl-genl.so.2
  ln -s libnl.so.2.0.0 libnl.so.2
  ```

- Copy iwconfig, iwlist, iwpriv, and iperf in **wifi_project/out/tools** to the **/sbin** directory of the board. These are debugging tools. You do not have to copy them.
- In STA mode, you need to copy wpa_supplicant and wpa_cli in **wifi_project/out/tools** to the **/sbin** directory of the board. In AP mode, you need to copy hostapd in **wifi_project/out/tools** to the **/sbin** directory of the board.
- Except using the wpa_supplicant and hostapd to configure the Broadcom Wi-Fi, you can use the wl tool for configuration. In this case, the **wl** file in the **wifi_project/out/tools** directory needs to be copied to the **/sbin** directory of the board.

After the tool is copied to the board, you need to modify the executable permission of the tool. For example:

```
chmod a+x wpa_supplicant
```

## 3.1.4 wpa_supplicant.conf File

**wpa_supplicant.conf** is a configuration file required when the wpa_supplicant process is started. You can create **wpa_supplicant.conf** on the board and store it in any directory (such as **/etc/Wireless**). The file content is as follows:

```
ctrl_interface=/var/wpa_supplicant
```

You can also copy the **wpa_supplicant.conf** file in the **sample** directory to the **/etc/Wireless** directory.

## 3.1.5 hostapd.conf File

**hostapd.conf** is a configuration file required when the hostapd process is started. You can create **hostapd.conf** on the board and store it in any directory (such as **/etc/Wireless**). For details about the file content, see section 3.3.2 "Configuring and Enabling the SoftAP by Using the hostapd Process."

You can also copy the **hostapd.conf** file in the **sample** directory to the **/etc/Wireless** directory.

## 3.1.6 udhcpd.conf File

**udhcpd.conf** is a configuration file required by the DHCP server in SoftAP mode. You can copy the **udhcpd.conf** file in the **wifi_project/sample** directory to any directory of the board (such as **/etc/Wireless**).

## 3.1.7 entropy.bin File

**entropy.bin** is a random number seed file which is needed when the encrypted SoftAP mode is configured in the hostapd process.You can copy the **entropy.bin** file in the **ifi_project/sample** directory to any directory of the board (such as **/etc/Wireless**).


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

# 3.2 Operation Examples for the STA Mode

## 3.2.1 Loading Drivers

Perform the following steps:

**Step 1** Load driver files by running the following commands based on the Wi-Fi drivers used:

- mt7601u

  **insmod cfg80211.ko**

  **insmod mtprealloc.ko**

  **insmod mtnet7601Usta.ko**
- rtl8188ftv

  **insmod cfg80211.ko**

  **insmod 8188fu.ko**
- rtl8188eus

  **insmod cfg80211.ko**

  **insmod 8188eu.ko**
- rtl8189ftv

  **insmod cfg80211.ko**

  **insmod 8189fs.ko**
- ap6181/ap6212/ap6212a/ap6214a/ap6255

  The following uses the ap6181 driver as an example:

  **insmod cfg80211.ko**

  **insmod bcmdhd.ko firmware_path=/etc/firmware/fw_bcm40181a2.bin nvram_path=/etc/firmware/nvram.txt dhd_oob_gpio_base=0x20140000 dhd_oob_gpio_num=71**

  **dhd_oob_ gpio_num** is the OOB GPIO number. The calculation method is as follows: GPIO group number x 8 + GPIO number. For details, see the *Peripheral Driver Operation Guide*.

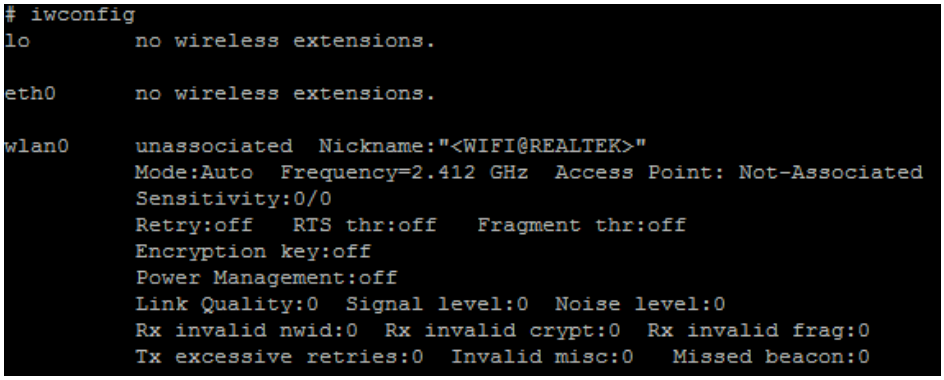
  After a driver is loaded, the WL_REG_ON needs to be pulled down and then pulled up.

**Step 2** Check whether the driver is successfully loaded by running the shell command **iwconfig**.

If the wlan0 network port exists, the driver is successfully initialized, and the Wi-Fi device is available.


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

**Figure 3-1** Execution result of iwconfig




**Step 3** Enable the Wi-Fi network port by running the shell command **ifconfig wlan0 up**.

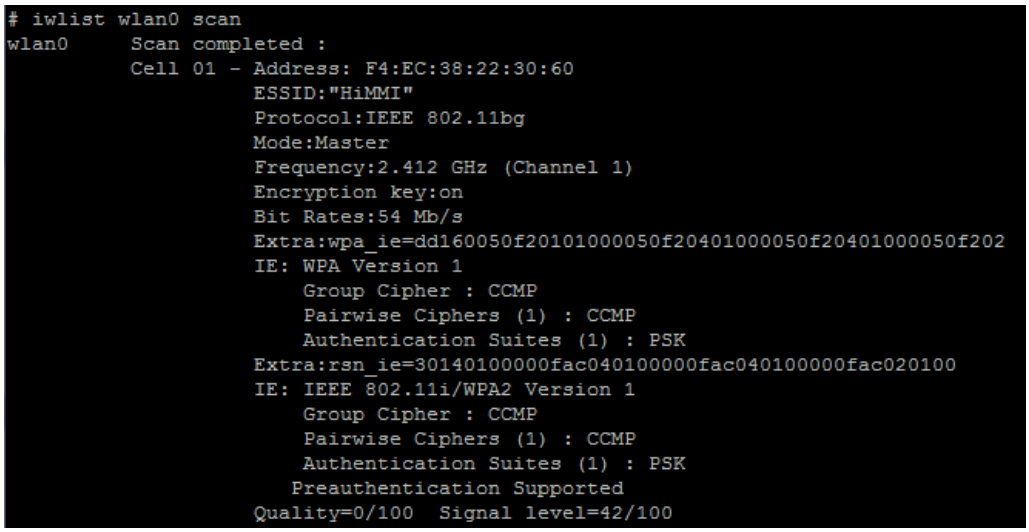
After the preceding command is executed, the Wi-Fi is available, and you can perform the scan and connect operations.

**-----End**

## 3.2.2 Scanning for APs

To scan for APs, run the shell command **iwlist wlan0 scan**.

**Figure 3-2** AP scanning result




The detected APs are displayed in the format of Cell *xx*, and each AP corresponds to a Cell *xx*.

The AP information includes:

- **Address**: MAC address
- **ESSID**: AP name, that is, SSID
- **Protocol**: IEEE80211 protocol, 11b/g/n
- **Frequency**: frequency


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

- **Encryption key** (authentication encryption information): WEP, WPA-PSK, WPA2-PSK, WPA, and WPA2
- **Quality**: signal quality. This data is sometimes inaccurate and can be ignored.
- **Signal Level**: signal strength. A larger value indicates greater signal strength. The display mode of the signal level varies with the Wi-Fi driver, for example, *xx*/100, or *xx* dBm.

The display format of the preceding information varies with the Wi-Fi driver.

**CAUTION**

When you scan for APs by running the **iwlist** command, the scan result is returned not necessarily after all frequencies are scanned. Therefore, some APs cannot be detected, especially for MT7601U. MT7601U scans each frequency for a long time period, and therefore only APs at one or two frequencies can be detected during the first scan.

## 3.2.3 Connecting to an AP

The wpa_supplicant process is used to connect the Wi-Fi device to an AP. wpa_supplicant is an open-source code which is used on Linux and Android to implement the Wi-Fi connection process. It includes protocols such as WEP, WPA/WPA2, WPA-PSK/WPA2-PSK, WAPI, WPS, P2P, and EAP.

**Step 1** Start the wpa_supplicant process by running the following shell command:

**wpa_supplicant –iwlan0 –Dnl80211 –c/etc/Wireless/wpa_supplicant.conf&**

- **-iwlan0** indicates that the wlan0 network port is used.
- **-Dnl80211** indicates that the cfg80211 interface is used (libnl for user-mode interfaces and cfg80211 for kernel-mode interfaces).
- /**etc/Wireless/wpa_supplicant.conf** indicates the configuration file of wpa_supplicant. Ensure that the file exists.

After the command is executed, run the **ps** command to check whether the wpa_supplicant process exists. If yes, it works properly. If no, raise the wpa_supplicant print level and find out the cause from the logs.

**wpa_supplicant –iwlan0 –Dnl80211 –c/etc/Wireless/wpa_supplicant.conf –ddd &**

**Step 2** Start the wpa_cli process by running the following shell command:

**wpa_cli –iwlan0–p/var/wpa_supplicant**

If the preceding command is successfully executed, the symbol ">" is displayed.

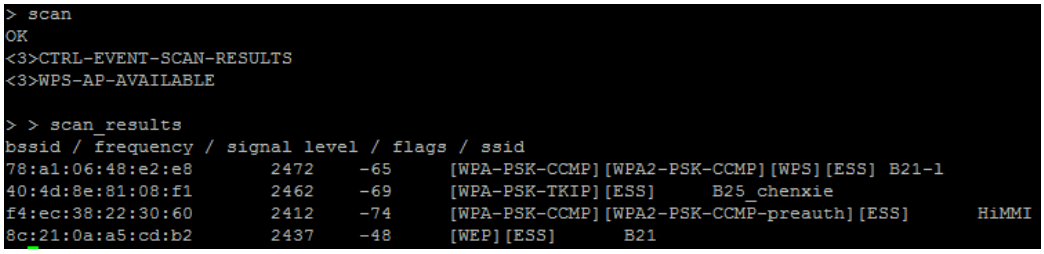
If "Could not connect to wpa_supplicant - re-trying" is displayed, the socket connection cannot be set up between wpa_cli and wpa_supplicant. In this case, check whether the wpa_supplicant process, **/var/wpa_supplicant/wlan0**, and **ctrl_interface=/var/wpa_supplicant** in the **wpa_supplicant.conf** file exists.

**Step 3** Scan for APs.

Run the **scan** command after the symbol ">", and run **scan_results** after **CTRL-EVENT-SCAN-RESULTS** is received. The scan result is displayed.


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

**Figure 3-3** AP scanning result of wpa_cli

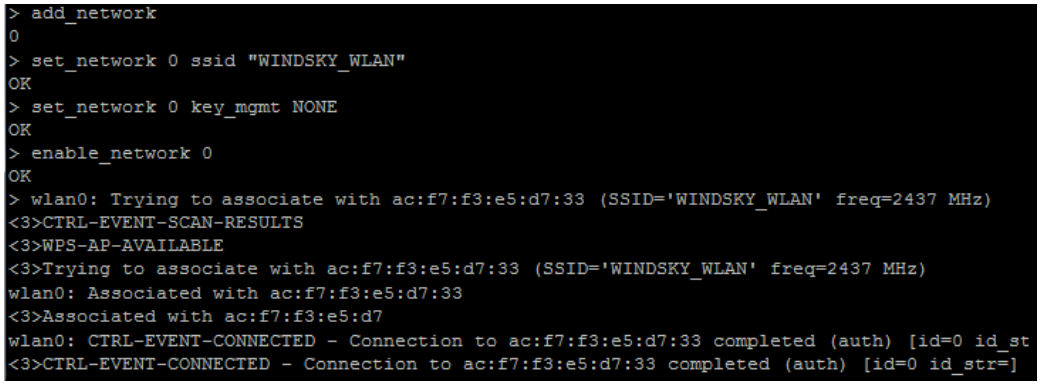



**Step 4** Connect an AP in any of the following ways as required:

- Connect an AP in open mode.
  1. Run **add_network** after the symbol ">". Assume that the returned network ID is 0.
  2. Configure the network SSID by running **set_network 0 ssid** (SSID of the AP).
  3. Configure the network encryption mode by running **set_network 0 key_mgmt NONE**.
  4. Enable the network by running **select_network 0**.

     If **CTRL-EVENT-CONNECTED** is received, the AP is successfully connected.

**Figure 3-4** Connecting to an AP




- Connect an AP in WPA-PSK/WPA2-PSK mode.
  1. Run **add_network** after the symbol ">". Assume that the returned network ID is 0.
  2. Configure the network SSID by running **set_network 0 ssid** (SSID of the AP).
  3. Configure the network encryption mode by running **set_network 0 psk** "*AP password*".
  4. Enable the network by running **select_network 0**.
  5. If **CTRL-EVENT-CONNECTED** is received, the AP is successfully connected.

The Broadcom Wi-Fi can be scanned and connected with the wl tool. For details about the connection method, see the help information of the sample and wl files.

**Step 5** Obtain the IP address.

Enter **q** to exit wpa_cli, and run the shell command **udhcpc –i wlan0**.

After the IP address is configured, run the **ping** command to check whether it can be pinged.


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

**----End**

### 3.2.4 Uninstalling Drivers

Uninstall drivers by running the following commands based on the used Wi-Fi drivers:

- mt7601u
  **ifconfig wlan0 down**
  **rmmod mt7601Usta.ko**
  **rmmod mtprealloc.ko**
  **rmmod cfg80211.ko**
- rtl8188ftv
  **ifconfig wlan0 down**
  **rmmod 8188fu.ko**
  **rmmod cfg80211.ko**
- rtl8188eus
  **ifconfig wlan0 down**
  **rmmod 8188eu.ko**
  **rmmod cfg80211.ko**
- rtl8189ftv
  **ifconfig wlan0 down**
  **rmmod 8189fs.ko**
  **rmmod cfg80211.ko**
- ap6181/ap6212/ap6212a/ap6214a/ap6255
  **ifconfig wlan0 down**
  **rmmod bcmdhd.ko**
  **rmmod cfg80211.ko**

## 3.3 Operation Examples for the SoftAP Mode

### 3.3.1 Checking Wi-Fi Devices and Loading Drivers

The mt7601u, rtl8188ftv, rtl8188eus, and rtl8189ftv drivers are loaded the same way as in STA mode. The firmware used by the ap6181, ap6212, ap6212a, ap6214a, or ap6255 driver is different as that in STA mode.

```
insmod cfg80211.ko
insmod bcmdhd.ko firmware_path=/etc/firmware/fw_bcm40181a2_apsta.bin
nvram_path=/etc/firmware/nvram.txt dhd_oob_gpio_num=71
```

After a driver is loaded, the WL_REG_ON needs to be pulled down and then pulled up.


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

# 3.3.2 Configuring and Enabling the SoftAP by Using the hostapd Process

The hostapd process is used to configure the SoftAP. The hostapd process is similar to wpa_supplicant. It contains various authentication protocols and connection processes of the AP end, whereas wpa_supplicant belongs to the STA end.

**Step 1** Modify **hostapd.conf**.

The hostapd process requires the **hostapd.conf** configuration file. You can set the SSID, frequency, and encryption mode in the configuration file. The following shows the examples of configuration files:

- OPEN

```
interface=wlan0
driver=nl80211
ctrl_interface=/var/hostapd
ssid=HisiAP
channel=6
hw_mode=g
ieee80211n=1
ht_capab=[SHORT-GI-20][SHORT-GI-40][HT40-]
```

- WPA2-PSK

```
interface=wlan0
driver=nl80211
ctrl_interface=/var/hostapd
ssid=HisiAP
channel=6
hw_mode=g
ieee80211n=1
ht_capab=[SHORT-GI-20][SHORT-GI-40][HT40-]
wpa=3
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP CCMP
wpa_passphrase=12345678
```

The hostapd code is open source. For details about parameters in the configuration file, search for network resources.

**ht_capab** can be configured to support or not support the 40 MHz bandwidth. When **[SHORT-GI-40][HT40-]** or **[SHORT-GI-40][HT40+]** is added after the **ht_capab=[SHORT-GI-20]**, the 40 MHz bandwidth is supported. When the value of the **channel** is less than 6, **[SHORT-GI-40][HT40+]** is added after the **ht_capab=[SHORT-GI-20]**. When the value of the **channel** is greater than or equal to 6, **[SHORT-GI-40][HT40-]** is added after the **ht_capab=[SHORT-GI-20]**.

The mt7601u, ap6181, ap6212, ap6212a, ap6214a, and ap6255 drivers do not support the 40 MHz bandwidth in SoftAP mode. Therefore, configure **ht_capab** as follows:

**ht_capab=[SHORT-GI-20]**


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

**Step 2** Start the hostapd process by running the following shell command:

**hostapd -e /etc/Wireless/entropy.bin /etc/Wireless/hostapd.conf &**

After the command is executed, run the **ps** command to check whether the hostapd process exists. If yes, it works properly, and the SoftAP can be detected by the STA device. If no, raise the hostapd print level and find out the cause from the logs. For example:

**hostapd -e /etc/Wireless/entropy.bin –ddd /etc/Wireless/hostapd.conf &**

**----End**

The Broadcom Wi-Fi can be configured with the wl tool. For details about the connection method, see the help information of the sample and wl files.

### 3.3.3 Enabling udhcpd

Enable udhcpd by running the following shell commands:

**ifconfig wlan0 192.168.1.1**

**busybox udhcpd –fS /etc/Wireless/udhcpd.conf**

Ensure that **/etc/Wireless/udhcpd.conf** exists, and the configured network segment is 192.168.1.*x*. After the preceding commands are executed, the SoftAP can be scanned and connected by using the STA device. If the SoftAP is successfully connected and network gateway can be pinged, the AP is successfully configured.

### 3.3.4 Uninstalling Drivers

The driver uninstalling process is the same as in STA mode.

## 3.4 Configuring the Country or Region

The frequency range varies according to the country or region. For example, for the 2.4 GHz frequency band, the USA supports channels 1 to 11, China and Europe support channels 1 to 13, and Japan supports channels 1 to 14. The situation is similar for the 5 GHz frequency band. The Wi-Fi device needs to be configured based on the country or region in which the device is to be launched.

The configuration method varies according to the Wi-Fi device. For example, to use RTL8188ftv in the USA, add the parameter **rtw_channel_plan=0x22** as follows when loading the driver:

**insmod rtl8188fu.ko rtw_channel_plan=0x22**

This document does not provide the configurations of all Wi-Fi drivers for various countries or regions. For details, consult module vendors or Wi-Fi driver vendors.


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

# 4 Tests

## 4.1 Throughput Test

The throughput tests show the Wi-Fi performance and are widely used and proved by chip vendors, module vendors, and Wi-Fi device vendors. The most frequently used throughput testing tool is iperf.

The test environment is that a PC connects to the AP with a cable, a board connects to the AP by using the Wi-Fi, and the PC and the board can ping each other successfully. The iperf tool is installed on both the PC and the board. Assume that the IP address for the PC is 192.168.1.100, and that for the board is 192.168.1.101.

**Figure 4-1** Networking for the throughput test


### 4.1.2 TCP TX Throughput Test

To test the (transmit) TX throughput, perform the following steps:

**Step 1** Run the **iperf –s** command to o to the directory of the iperf tool on the PC.

**Step 2** Go to the directory of the iperf tool using shell on the board by the following command:

**iperf –c 192.168.1.100 –t 10 –i 1**


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

**Figure 4-2** TX throughput test example

```
# iperf -c 192.168.1.100 -t 10 -i 1
Client connecting to 192.168.1.100, TCP port 5001
TCP window size:  512 KByte (default)
[  3] local 192.168.1.101 port 44753 connected with 192.168.1.100 port 5001
[  3]  0.0- 1.0 sec  8.40 MBytes  70.5 Mbits/sec
[  3]  1.0- 2.0 sec  8.57 MBytes  71.9 Mbits/sec
[  3]  2.0- 3.0 sec  8.65 MBytes  72.5 Mbits/sec
[  3]  3.0- 4.0 sec  8.52 MBytes  71.4 Mbits/sec
[  3]  4.0- 5.0 sec  8.57 MBytes  71.9 Mbits/sec
[  3]  5.0- 6.0 sec  8.52 MBytes  71.4 Mbits/sec
[  3]  6.0- 7.0 sec  8.59 MBytes  72.1 Mbits/sec
[  3]  7.0- 8.0 sec  8.52 MBytes  71.5 Mbits/sec
[  3]  8.0- 9.0 sec  8.72 MBytes  73.1 Mbits/sec
[  3]  9.0-10.0 sec  8.62 MBytes  72.4 Mbits/sec
[  3]  0.0-10.0 sec  85.7 MBytes  71.6 Mbits/sec
```

**iperf -s** indicates starting the server. **iperf -c 192.168.1.100** indicates starting the client and connecting to 192.168.1.100. **-t 10** indicates 10-second testing period. **-i 1** indicates that the result will be printed once every 1 second.

The displayed test result "0.0-10.0 sec  85.7 MBytes  71.6 Mbit/sec" in the last row indicates that the average throughput in 10 seconds is 71.6 Mbit/s.

**----End**

## 4.1.3 TCP RX Throughput Test

To test the receive (RX) throughput, perform the following steps:

**Step 1** Run the **iperf –s** command to go to the directory of the iperf tool using shell on the board.

**Step 2** Go to the directory of the iperf tool on the PC by the following command:

**iperf –c 192.168.1.101 –t 10 –i 1 –w 1M**

**Figure 4-3** RX throughput test example

```
# iperf -s -i 1
Server listening on TCP port 5001
TCP window size: 1.00 MByte (default)
GetDesiredTssiAndCurrentTssi: BBP TSSI INFO is not ready. (BbpR47 = 0x94)
RT5390_AsicTxAlcGetAutoAgcOffset: Incorrect desired TSSI or current TSSI
[  4] local 192.168.1.101 port 5001 connected with 192.168.1.100 port 59938
[  4]  0.0- 1.0 sec  10.1 MBytes  85.0 Mbits/sec
[  4]  1.0- 2.0 sec  10.3 MBytes  86.5 Mbits/sec
[  4]  2.0- 3.0 sec  10.1 MBytes  84.4 Mbits/sec
[  4]  3.0- 4.0 sec  9.86 MBytes  82.8 Mbits/sec
[  4]  4.0- 5.0 sec  9.83 MBytes  82.4 Mbits/sec
[  4]  5.0- 6.0 sec  9.92 MBytes  83.3 Mbits/sec
[  4]  6.0- 7.0 sec  9.33 MBytes  78.3 Mbits/sec
[  4]  7.0- 8.0 sec  9.99 MBytes  83.8 Mbits/sec
[  4]  8.0- 9.0 sec  9.70 MBytes  81.4 Mbits/sec
[  4]  9.0-10.0 sec  10.0 MBytes  84.2 Mbits/sec
[  4]  0.0-10.1 sec   100 MBytes  83.3 Mbits/sec
```

The iperf tool can also be used to perform the User Datagram Protocol (UDP) test. The speed of a single UDP thread is limited on some PCs, and therefore multiple threads are required.


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd

The throughput tests for the SoftAP are similar.

**CAUTION**

The speed of some PCs is affected by the installed software. Ensure that the PC speed is not affected. The 802.11n protocol cannot be used in WEP safe mode, and therefore the speed is low, typically over 20 Mbit/s.

**----End**

### 4.1.4 UDP TX Throughput Test

To test the TX throughput, perform the following steps:

**Step 1** Run the **iperf –s -u –l 32k** command to go to the directory of the iperf tool on the PC.

**Step 2** Go to the directory of the iperf tool using shell on the board by the following command:

**iperf –c 192.168.1.100 –u –t 10 –i 1 –l 32k –b 100M**

**----End**

### 4.1.5 UDP RX Throughput Test

To test the RX throughput, perform the following steps:

**Step 1** Run the **iperf –s -u** command to go to the directory of the iperf tool using shell on the board.

**Step 2** Go to the directory of the iperf tool on the PC by the following command:

**iperf –c 192.168.1.101 –u –t 10 –i 1 –l 32k –b 100M**

**----End**

## 4.2 RF Specification Test

The throughput tests reflect the Wi-Fi performance and are mandatory during product development. Some companies also conduct the radio frequency (RF) specifications test, which can accurately verify whether the Wi-Fi RF meets specifications. The RF specifications test is mandatory during module production. Therefore, if a Wi-Fi module is used, this test is optional. However, the Wi-Fi RF performance may be affected due to board interference and unclean GND traces during hardware design. Therefore, you are advised to conduct this test if possible.

The RF specifications include the following: receive sensitivity, transmit power, error tolerance of the transmit carrier frequency, packet loss rate, error vector magnitude (EVM), transmit adjacent channel power ratio (ACPR), receive ACPR, and so on.

The test tools include spectrum analyzer, power measurer, network analyzer and so on.

For details about the test methods, see the instructions of the test tools.


HiSilicon Proprietary and Confidential
Copyright © HiSilicon Technologies Co., Ltd