



WiFi

使用指南

文档版本 05

发布日期 2018-02-10

版权所有 © 深圳市海思半导体有限公司 2014-2018。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HISILICON、海思和其他海思商标均为深圳市海思半导体有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

深圳市海思半导体有限公司

地址：深圳市龙岗区坂田华为基地华为电气生产中心 邮编：518129

网址：<http://www.hisilicon.com>

客户服务电话：+86-755-28788858

客户服务传真：+86-755-28357515

客户服务邮箱：support@hisilicon.com



前言

概述

本文档主要介绍 Wi-Fi 需要使用到的配置，基本操作、调试方法，使用注意事项和常见问题处理。



说明

- 本文未做特殊说明，Hi3516D 与 Hi3516A 完全一致。
- 未做特殊说明，Hi3518EV201、Hi3516CV200 与 Hi3518EV200 完全一致。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
Hi3516A	V100
Hi3516D	V100
Hi3518E	V200
Hi3518E	V201
Hi3516C	V200
Hi3559A	V100
Hi3559C	V100
Hi3519A	V100

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师



修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修订日期	版本	修订说明
2018-02-10	05	添加 Hi3559AV100 和 Hi3559CV100 相关内容 1.2 小节涉及修改 3.1.1、3.1.2、3.2.4 和 3.3.1 小节涉及修改
2018-01-15	04	第四次正式发布，删除 Hi3518EV20X 相关内容 每个章节均涉及修改，并新增第 4 章“测试”。
2015-09-25	03	添加 Hi3518EV200, Hi3518EV201 和 Hi3516CV200 的相关内容。
2015-02-10	02	3.3.3 节，将原来的步骤 3 和步骤 4 合并。
2014-12-30	01	添加 Hi3516D 的相关内容。 2.1 节中新增“配置 Wireless Extension”，删除“配置 Netlink”和“配置 Wireless Extension”，新增“配置 IPv6”一节。 2.2 删除“wifi 驱动配置”和“wifi 无线管理工具”。 第 3 章 新增“载入驱动文件”、“载入 firmware 文件”、“卸载驱动”和“适配国家或区域”等小节。
2014-11-09	00B01	第 1 次临时版本发布。



目 录

前 言.....	i
1 概述.....	1
1.1 背景.....	1
1.2 目录说明.....	1
2 配置说明.....	3
2.1 内核配置.....	3
2.1.1 配置 CFG80211.....	3
2.1.2 配置 Wireless Extension.....	3
2.1.3 配置 USB 和 SDIO.....	4
2.1.4 配置 IPv6.....	4
2.1.5 SDIO 中断配置.....	5
2.1.6 配置通用 GPIO.....	5
2.2 配置 wifi_project 开发包.....	6
3 WiFi 基本操作.....	7
3.1 载入文件.....	7
3.1.1 载入驱动文件.....	7
3.1.2 载入 firmware 文件.....	7
3.1.3 载入工具.....	8
3.1.4 wpa_supplicant.conf 文件.....	8
3.1.5 hostapd.conf 文件.....	8
3.1.6 udhcpd.conf 文件.....	8
3.1.7 entropy.bin 文件.....	8
3.2 STA 模式基本操作示例.....	9
3.2.1 加载驱动.....	9
3.2.2 扫描 AP.....	10
3.2.3 连接 AP.....	11
3.2.4 卸载驱动.....	13
3.3 SoftAP 模式基本操作示例.....	14
3.3.1 检查 WiFi 设备、加载驱动.....	14



3.3.2 hostapd 配置和启动 SoftAP	14
3.3.3 开启 udhcpd.....	15
3.3.4 卸载驱动	15
3.4 适配国家或区域.....	15
4 测试.....	17
4.1 吞吐量测试.....	17
4.1.1 TCP 发送吞吐量测试	17
4.1.2 TCP 接收吞吐量测试	18
4.1.3 UDP 发送吞吐量测试.....	19
4.1.4 UDP 接收吞吐量测试.....	19
4.2 射频指标测试.....	19



插图目录

图 2-1 CFG80211 配置	3
图 2-2 Wireless Extension 配置	4
图 2-3 Ipv6 配置	5
图 2-4 通用 GPIO 配置	5
图 2-5 通用 GPIO 配置 2	6
图 3-1 iwconfig 执行结果	10
图 3-2 扫描 AP 执行结果	10
图 3-3 wpa_cli 扫描 AP 结果	12
图 3-4 连接 AP	12
图 4-1 吞吐量测试组网环境	17
图 4-2 发送吞吐量测试示例	18
图 4-3 接收吞吐量测试示例	18



1 概述

1.1 背景

由于 Wi-Fi 厂商提供的软件包编译时需要的命令、步骤各不相同，并且每款 Wi-Fi 支持的模式（一般有四种模式，分别是：SoftAP、STA、DIRECT、CONCURRENT）也不同，操作时所需要的工具、命令也存在差异。通过 wifi_project 开发包能够很好的解决上述问题，可以方便调试不同型号的 Wi-Fi。用户使用它不仅可以方便、快速的生成不同型号 Wi-Fi 所需要的驱动、工具等，而且还能够直接增加新的 Wi-Fi 驱动或删除已有的驱动。

通常 Wi-Fi 有四种模式，不同的 Wi-Fi 设备可能支持其中一种或几种模式，四种模式如下：

- SoftAP: access point，一种将无线设备连接到一个网络的设备，可以理解为无线路由。
- STA: station，无线设备客户端，必须连接上一个 AP 才能使用。
- DIRECT: Wi-Fi 直连，也称 p2p 模式。
- CONCURRENT: 同时支持 AP、STA 两种模式。

wifi_project 开发包中支持的 Wi-Fi 器件请参考主目录下的 Makefile。wifi_project 开发包仅支持 SoftAP 和 STA 模式。

1.2 目录说明

wifi_project

- ├─Makefile ----- wifi_project 编译脚本
- ├─docs ----- 各种类型 Wi-Fi 的说明、操作文档
- ├─firmware ----- 各种类型 Wi-Fi 使用的固件文件
- ├─tools ----- 操作、配置 Wi-Fi 的工具源码目录
 - ├─hostapd ----- hostapd 工具源码，负责 SoftAP 模式的操作
 - ├─wireless_tools ----- iwlist、iwpriv、iwconfig 等源码，用于调试



| |——wpa_supplicant ----- wpa_supplicant 工具源码，负责 STA 模式的操作
| |——libnl ----- libnl 源码，wpa_supplicant 和 hostapd 需要使用的库
| |——iperf-----iperf 源码
| |----wl -----Broadcom WiFi 使用的 wl 工具代码
| |——Makefile ----- 编译各工具所需的脚本
|----sample ----- 开启 WiFi 的样例脚本
|——drv ----- Wi-Fi 驱动源码
| |——Makefile ----- 编译各驱动所需的脚本
| |——usb_rtl8188ftv ----- Realtek rtl8188ftv 驱动源码
| |——sdio_ap6xxx ----- ap6181、ap6212、ap6255 等驱动源码
| |——usb_rtl8188eus ----- Realtek rtl8188eus 驱动源码
| |——usb_mt7601u ----- MediaTek mt7601u 驱动源码
| |——..... ----- 其他驱动源码



2 配置说明

2.1 内核配置

2.1.1 配置 CFG80211

CFG80211 是内核中 WiFi 驱动和用户态进程的标准接口，在 CFG80211 出现之前是 WEXT，现在越来越多的使用 CFG80211，WiFi Direct 功能只有 CFG80211 才支持。

进入 Network support->Wireless，设置 cfg80211 和 mac80211 为 M，如图 2-1 所示。

图2-1 CFG80211 配置

```
--- Wireless
<M> cfg80211 - wireless configuration API
[ ] nl80211 testmode command (NEW)
[ ] enable developer warnings (NEW)
[ ] cfg80211 regulatory debugging (NEW)
[*] enable powersave by default (NEW)
[ ] cfg80211 wireless extensions compatibility (NEW)
<M> Generic IEEE 802.11 Networking Stack (mac80211)
    Default rate control algorithm (Minstrel) --->
[ ] Enable mac80211 mesh networking (pre-802.11s) support (NEW)
[ ] Trace all mac80211 debug messages (NEW)
[ ] Select mac80211 debugging features (NEW) ----
```

2.1.2 配置 Wireless Extension

- WEXT 是内核中 WiFi 驱动和用户态进程的标准接口，调试工具 iwconfig、iwlist、iwpriv 需要使用该接口。如果没有配置改接口，有些驱动会有编译错误。
- WEXT 在内核配置中没有单独的配置项，只能通过打开依赖它的配置项来间接打开。配置了 CFG80211 后，进入 Device Drivers->Network device support->Wireless LAN，设置 USB ZD1201 based Wireless device support 为 M。如图 2-2 所示。



图2-2 Wireless Extension 配置

```
--- Wireless LAN
< > Marvell 8xxx Libertas WLAN driver support with thin firmware
< > Atmel at76c503/at76c505/at76c505a USB cards
< > USB ZD1201 based Wireless device support
< > Wireless RNDIS USB support
< > Realtek 8187 and 8187B USB support
< > Simulated radio testing tool for mac80211
[ ] Enable WiFi control function abstraction
< > Atheros Wireless Cards --->
< > Broadcom 43xx wireless support (mac80211 stack)
< > Broadcom 43xx-legacy wireless support (mac80211 stack)
< > Broadcom 4329/30 wireless cards support
< > Broadcom IEEE802.11n embedded FullMAC WLAN driver
< > IEEE 802.11 for Host AP (Prism2/2.5/3 and WEP/TKIP/CCMP)
< > Intel Wireless Multicom 3200 WiFi driver
< > Marvell 8xxx Libertas WLAN driver support
< > Softmac Prism54 support
< > Ralink driver support --->
< > Realtek RTL8192CU/RTL8188CU USB Wireless Network Adapter
< > TI wl1251 driver support --->
< > TI wl12xx driver support --->
< > ZyDAS ZD1211/ZD1211B USB-wireless support
< > Marvell WiFi-Ex Driver
```

如果找不到这项配置，需要先配置好 USB 再来配置该选项。

2.1.3 配置 USB 和 SDIO

请参考《外围设备驱动操作指南》中的 USB 和 SDIO 操作指南。

- ap6181、ap6212、ap6212a、ap6214a 的接口是 SDIO2.0，请设置 SDIO 时钟为 50MHz 左右，ap6255 的接口是 SDIO3.0，请设置 SDIO 时钟为 150MHz 左右。
- kernel 默认的 SDIO2.0 的 IO 电压为 3.3V，如果给 WiFi 模块的 IO 电压为 1.8V，需要修改 SDIO 驱动，将 IO 电压切换到 1.8V。

2.1.4 配置 IPv6

使用 ap6181、ap6212、ap6212a、ap6214a、ap6255 时，不配置 IPv6 会导致驱动运行错误，所以需要配置 IPv6。使用其他四款 WiFi 无需配置该项。

进入 Network support->Networking options，设置 The IPv6 protocol 为 y。如图 2-3 所示。



图2-3 Ipv6 配置

```
[*] TCP: advanced congestion control --->
[*] TCP: MD5 Signature Option support (RFC2385) (EXPERIMENTAL)
<+> The IPv6 protocol --->
[ ] CreVinn TOE-NK-2G TCP Offload Engine support
[ ] Only allow certain groups to create sockets
[*] Network activity statistics tracking
[ ] Security Marking
[ ] Timestamping in PHY devices
[*] Network packet filtering framework (Netfilter) --->
```

2.1.5 SDIO 中断配置

Kernel 默认没有开启 SDIO 中断，使用不支持 OOB 的 WiFi 时需要将 SDIO 中断打开，开启方法如：文件 arch/arm/boot/dts/hi3516a.dtsi，在 WiFi 对接的 SDIO 端口属性中增加：cap-sdio-irq。



注意

配置并编译完内核后，需要基于新的内核重新编译 WiFi 驱动，否则 WiFi 驱动在运行时会出现空指针或找不到内核符号错误。

2.1.6 配置通用 GPIO

打开通用 GPIO 配置，进入 Device Drivers，开启 GPIO Support，然后再进入 GPIO Support，按图 2-4 配置。

图2-4 通用 GPIO 配置

```
-- GPIO Support
[ ] Debug GPIO calls
[*] /sys/class/gpio/... (sysfs interface)
    Memory mapped GPIO drivers --->
        I2C GPIO expanders --->
        MFD GPIO expanders --->
        SPI GPIO expanders --->
        SPI or I2C GPIO expanders --->
        USB GPIO expanders ----
```

然后进入 Memory mapped GPIO drivers，按图 2-5 配置。



图2-5 通用 GPIO 配置 2

```
< > GPIO driver for 74xx-ICs with MMIO access
< > Altera GPIO
< > Synopsys DesignWare APB GPIO driver
< > Emma Mobile GPIO
< * > Generic memory-mapped GPIO controller support (MMIO platform device)
< > Aeroflex Gaisler GRGPIO support
< > GPIO Testing Driver
[ ] MPC512x/MPC8xxx/QorIQ GPIO support
[ * ] PrimeCell PL061 GPIO support
< > Xilinx GPIO support
[ ] LSI ZEVIO SoC memory mapped GPIOs
[ ] ZTE ZX GPIO support
```

2.2 配置 wifi_project 开发包

wifi_project 在编译前需要配置交叉编译环境，修改顶层 Makefile 中的 WIFI_DEVICE、CROSS_COMPILE 和 KERNEL。

- 交叉编译工具链
编译时需要的交叉编译工具，例如 arm-hisiv500-linux-、arm-hisiv600-linux-等。
如：ARCH := arm
CROSS_COMPILE:= arm-hisiv500-linux -
- **WIFI_DEVICE**
希望编译的 Wi-Fi 设备名称，支持的 WiFi 设备见 Makefile 文件中的注释。
如：WIFI_DEVICE := sdio_ap6xxx
ap6181、ap6212、ap6212a、ap6214a、ap6225 都配置为 sdio_ap6xxx。
- **KERNEL**
编译 Wi-Fi 驱动时所指定的内核路径，在编译 Wi-Fi 驱动时，必须指定内核路径，注意内核一定要已经编译过的。
如：KERNEL := /home/work/linux-3.18.y

配置完后就可以进行编译了，在 wifi_project 目录下执行 make all，会自动编译驱动和工具，make driver 只编译驱动，make tools 只编译工具。

生成的驱动放在 wifi_project/out/kmod 目录下，工具放在 wifi_project/out/tools 目录下，主要包括 iwconfig、iwlist、iwpriv、wpa_cli、wpa_supplicant、hostapd 等，生成的库文件放在 wifi_project/out/lib 目录下。



3 WiFi 基本操作

3.1 载入文件

3.1.1 载入驱动文件

编译后驱动会生成在 `wifi_project/out/kmod` 目录下，拷贝所需的驱动到单板中。

各个 WiFi 的驱动文件如下：

- `mt7601u`
`cfg80211.ko`、`mtprealloc.ko`、`mt7601Usta.ko`
- `rtl8188ftv`
`cfg80211.ko`、`8188fu.ko`
- `rtl8188eus`
`cfg80211.ko`、`8188eu.ko`
- `rtl8189ftv`
`cfg80211.ko`、`8189fs.ko`
- `ap6181/ap6212/ap6212a/ap6214a/ap6255`
`cfg80211.ko`、`bcmdhd.ko`
`cfg80211.ko` 文件请在 `kernel` 的 `net/wireless` 目录下拷贝。

驱动在单板上的目录不重要，比如可以放在 `/kmod` 目录。

3.1.2 载入 firmware 文件

- 如果要使用 Broadcom 的 `ap6181` 芯片，请执行如下操作：
在单板上建立目录 `/etc/firmware`，将 `wifi_project/firmware/sdio_ap6181` 文件夹下的固件文件 `fw_bcm40181a2.bin`、`fw_bcm40181a2_apsta.bin` 和 `nvrn.txt` 下载到单板此目录下。`ap6212/ap6212a/ap6214a/ap6255` 请拷贝 `wifi_project/firmware` 对应目录的 `firmware` 和 `nvrn` 文件。
- 如果要使用 MediaTek 的 `mt7601u`，请执行如下操作：
在单板上建立目录 `/etc/Wireless/RT2870STA`，将 `wifi_project/firmware/usb_mt7601u` 文件夹下的固件 `MT7601USTA.dat` 下载到单板此目录下。



- RealTek 的芯片不需要额外加载固件。

3.1.3 载入工具

- 将 `wifi_project/out/lib` 目录下的 `libnl-genl.so.2.0.0` 和 `libnl.so.2.0.0` 拷贝到单板的 `/lib` 目录。进入单板 `/lib` 目录，创建这两个文件的软链接：

```
ln -s libnl-genl.so.2.0.0 libnl-genl.so.2
ln -s libnl.so.2.0.0 libnl.so.2
```
- 拷贝 `wifi_project/out/tools` 目录下的 `iwconfig`、`iwlist`、`iwpriv`、`iperf` 拷贝到单板的 `/sbin` 目录下。这几个是调试工具，实际使用时可以不用拷贝这几个文件。
- STA 模式需要将 `wifi_project/out/tools` 目录下的 `wpa_supplicant`、`wpa_cli` 拷贝到单板的 `/sbin` 目录下。
- AP 模式需要将 `wifi_project/out/tools` 目录下的 `hostapd` 拷贝到单板的 `/sbin` 目录下。
- Broadcom WiFi 除了使用 `wpa_supplicant` 和 `hostapd` 来配置外，还有一种方式是采用 `wl` 工具，这种情况下需要将 `wifi_project/out/tools` 目录下的 `wl` 拷贝到单板的 `/sbin` 目录下。

工具拷贝到单板后，需要修改工具的可执行权限，如：

```
chmod a+x wpa_supplicant
```

3.1.4 wpa_supplicant.conf 文件

`wpa_supplicant.conf` 是启动 `wpa_supplicant` 进程时需要使用到的配置文件。可以在单板上新建一个，目录不重要，比如放在 `/etc/Wireless` 目录下。文件内容如下：

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

也可以将 `sample` 目录下的 `wpa_supplicant.conf` 拷贝到 `/etc/Wireless` 目录下。

3.1.5 hostapd.conf 文件

`hostapd.conf` 是启动 `hostapd` 进程时需要使用到的配置文件。可以在单板上新建一个，目录不重要，比如放在 `/etc/Wireless` 目录下。文件内容请参考 [hostapd 配置和启动 SoftAP](#)。也可以将 `sample` 目录下的 `hostapd.conf` 拷贝到 `/etc/Wireless` 目录下。

3.1.6 udhcpd.conf 文件

`udhcpd.conf` 文件是 SoftAP 模式下 DHCP server 需要使用到的配置文件。在 `wifi_project/sample` 目录下拷贝 `udhcpd.conf` 到单板目录下，目录不重要，比如放在 `/etc/Wireless` 目录下。

3.1.7 entropy.bin 文件

`entropy.bin` 文件是随机数种子文件，`hostapd` 进程配置带加密的 SoftAP 模式时需要使用。在 `wifi_project/sample` 目录下拷贝 `entropy.bin` 到单板目录下，目录不重要，比如放在 `/etc/Wireless` 目录下。



3.2 STA 模式基本操作示例

3.2.1 加载驱动

步骤 1. 加载驱动。

各个 WiFi 芯片请按以下对应关系加载相应的驱动模块。

- mt7601u
insmod cfg80211.ko
insmod mtprealloc.ko
insmod mt7601Usta.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
以 ap6181 为例：
insmod cfg80211.ko
insmod bcmdhd.ko firmware_path=/etc/firmware/fw_bcm40181a2.bin
nvram_path=/etc/firmware/nvram.txt dhd_oob_gpio_num=71
dhd_oob_gpio_num 为 OOB GPIO 号，计算方法为 GPIO 组×8 + GPIO 号，请参考
《外围设备驱动操作指南》；
加载完驱动后需要将 WL_REG_ON 拉低再拉高。

步骤 2. 查看驱动是否加载成功。

执行 shell 命令：

```
iwconfig
```

如果看到有一个 wlan0 网口，那说明驱动已经初始化成功，WiFi 设备可用。



图3-1 iwconfig 执行结果

```
# iwconfig
lo      no wireless extensions.

eth0    no wireless extensions.

wlan0   unassociated Nickname:"<WIFI@REALTEK>"
        Mode:Auto Frequency=2.412 GHz Access Point: Not-Associated
        Sensitivity:0/0
        Retry:off RTS thr:off Fragment thr:off
        Encryption key:off
        Power Management:off
        Link Quality:0 Signal level:0 Noise level:0
        Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
        Tx excessive retries:0 Invalid misc:0 Missed beacon:0
```

步骤 3. 启动 WiFi 网口。

执行 shell 命令：

```
ifconfig wlan0 up
```

执行完后，WiFi 是可用状态，可以进行扫描和连接操作了。

----结束

3.2.2 扫描 AP

执行 shell 命令：

```
iwlist wlan0 scan
```

图3-2 扫描 AP 执行结果

```
# iwlist wlan0 scan
wlan0    Scan completed :
          Cell 01 - Address: F4:EC:38:22:30:60
                   ESSID:"HiMMI"
                   Protocol:IEEE 802.11bg
                   Mode:Master
                   Frequency:2.412 GHz (Channel 1)
                   Encryption key:on
                   Bit Rates:54 Mb/s
                   Extra:wpa_ie=dd160050f20101000050f20401000050f20401000050f202
                   IE: WPA Version 1
                        Group Cipher : CCMP
                        Pairwise Ciphers (1) : CCMP
                        Authentication Suites (1) : PSK
                   Extra:rsn_ie=30140100000fac040100000fac040100000fac020100
                   IE: IEEE 802.11i/WPA2 Version 1
                        Group Cipher : CCMP
                        Pairwise Ciphers (1) : CCMP
                        Authentication Suites (1) : PSK
                        Preauthentication Supported
                   Quality=0/100 Signal level=42/100
```

扫描到的 AP 会以“Cell xx”的形式显示，一个 AP 对应一个“Cell xx”。



每个 AP 的信息包括：

- Address: MAC 地址。
- ESSID: AP 的名称，即 SSID。
- Protocol: IEEE80211 协议，11b/g/n。
- Frequency: 信道。
- 认证加密信息: WEP、WPA-PSK、WPA2-PSK、WPA、WPA2。
- Quality: 信号质量，该数据有些 WiFi 显示得不准确，可以忽略。
- Signal Level: 信号强度，数字越大，信号强度越高，WiFi 芯片不同，显示的方式有些区别，有的是以 xx/100 类型显示，有的是以 xx dBm 显示。

上述信息并不是所有 WiFi 都是以这种格式显示，WiFi 不同显示的格式也不一样。



注意

使用 iwlist 进行扫描时，iwlist 不会等驱动扫描完所有信道才返回扫描结果，所以经常会出现有些 AP 没有搜出来的情况，尤其是 MT7601U，由于在每个信道上停留的时间较长，所以第一次扫描时，只能搜到 1~2 个信道里的 AP。

3.2.3 连接 AP

连接 AP 是通过 wpa_supplicant 进程进行的。wpa_supplicant 是开源代码，Linux、Android 都是采用它负责 WiFi 的连接过程，它包含了 WEP、WPA/WPA2、WPA-PSK/WPA2-PSK、WAPI、WPS、P2P、EAP 等协议。

步骤 1. 启动 wpa_supplicant 进程。

执行 shell 命令：

```
wpa_supplicant -iwlan0 -Dnl80211 -c/etc/Wireless/wpa_supplicant.conf&
```

- -iwlan0 表示使用 wlan0 网口；
- -Dnl80211 表示使用 cfg80211 接口（用户态的接口是 libnl，内核中是 cfg80211）；
- /etc/Wireless/wpa_supplicant.conf 是 wpa_supplicant 的配置文件，要保证该文件已经存在。

执行完后，用 ps 命令查看一下 wpa_supplicant 进程是否存在，存在表示工作正常。如果没有 wpa_supplicant 进程，可以增加 wpa_supplicant 的打印级别，从 log 看出什么问题，如：

```
wpa_supplicant -iwlan0 -Dnl80211 -c/etc/Wireless/wpa_supplicant.conf -ddd  
&
```

步骤 2. 启动 wpa_cli 进程。

执行 shell 命令：

```
wpa_cli -iwlan0-p/var/wpa_supplicant
```



执行成功会出现“>”符号。

如果出现“Could not connect to wpa_supplicant - re-trying”，那表示 wpa_cli 不能和 wpa_supplicant 建立 socket 连接，这时要检查 wpa_supplicant 进程是否还在，再看是否有/var/wpa_supplicant/wlan0，然后检查 wpa_supplicant.conf 文件中是否是 ctrl_interface=/var/wpa_supplicant。

步骤 3. 扫描。

在“>”后执行“scan”命令，收到“CTRL-EVENT-SCAN-RESULTS”后再执行“scan_results”，会获得扫描结果。

图3-3 wpa_cli 扫描 AP 结果

```
> scan
OK
<3>CTRL-EVENT-SCAN-RESULTS
<3>WPS-AP-AVAILABLE

> scan_results
ssid / frequency / signal level / flags / ssid
78:a1:06:48:e2:e8      2472      -65      [WPA-PSK-CCMP][WPA2-PSK-CCMP][WPS][ESS] B21-1
40:4d:8e:81:08:f1      2462      -69      [WPA-PSK-TKIP][ESS] B25_chenxie
f4:ec:38:22:30:60      2412      -74      [WPA-PSK-CCMP][WPA2-PSK-CCMP-preauth][ESS] HiMMI
8c:21:0a:a5:cd:b2      2437      -48      [WEP][ESS] B21
```

步骤 4. 连接。

连接 OPEN 方式的 AP:

- 在“>”后执行 add_network，假如返回网络 ID 为 0。
- 配置网络的 SSID，执行 set_network 0 ssid “AP 的 SSID”。
- 配置网络的加密方式，执行 set_network 0 key_mgmt NONE。
- 启动网络，执行 select_network 0。
- 收到 CTRL-EVENT-CONNECTED 表示连接成功。

图3-4 连接 AP

```
> add_network
0
> set_network 0 ssid "WINDSKY_WLAN"
OK
> set_network 0 key_mgmt NONE
OK
> enable_network 0
OK
> wlan0: Trying to associate with ac:f7:f3:e5:d7:33 (SSID='WINDSKY_WLAN' freq=2437 MHz)
<3>CTRL-EVENT-SCAN-RESULTS
<3>WPS-AP-AVAILABLE
<3>Trying to associate with ac:f7:f3:e5:d7:33 (SSID='WINDSKY_WLAN' freq=2437 MHz)
wlan0: Associated with ac:f7:f3:e5:d7:33
<3>Associated with ac:f7:f3:e5:d7
wlan0: CTRL-EVENT-CONNECTED - Connection to ac:f7:f3:e5:d7:33 completed (auth) [id=0 id_str=]
<3>CTRL-EVENT-CONNECTED - Connection to ac:f7:f3:e5:d7:33 completed (auth) [id=0 id_str=]
```

连接 WPA-PSK/WPA2-PSK 方式的 AP:

- 在“>”后执行 add_network，假如返回网络 ID 为 0。



- 配置网络的 SSID，执行 `set_network 0 ssid “AP 的 SSID”`。
- 配置网络的加密方式，执行 `set_network 0 psk “AP 的密码”`。
- 启动网络，执行 `select_network 0`。
- 收到 CTRL-EVENT-CONNECTED 表示连接成功。

Broadcom 的 WiFi 还可以通过 `wl` 工具进行扫描和连接，连接方法请参考 `sample` 和 `wl` 的 `help` 信息。

步骤 5. 获取 IP 地址。

输入 `q` 退出 `wpa_cli`，执行 `shell` 命令：`udhcpc -i wlan0`

配置了 IP 地址后，可以 `ping` 网关看是否能 `ping` 通。

----结束

3.2.4 卸载驱动

各个 WiFi 芯片请按以下对应关系卸载相应的驱动模块。

- `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 SoftAP 模式基本操作示例

3.3.1 检查 WiFi 设备、加载驱动

mt7601u、rtl8188ftv、rtl8188eus、rtl8189ftv 加载驱动的方法和 STA 模式一样，ap6181/ap6212/ap6212a/ap6214a/ap6255 用的 firmware 和 STA 模式不一样，如下：

```
insmod cfg80211.ko
```

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

加载完驱动后将 WL_REG_ON 拉低再拉高。

3.3.2 hostapd 配置和启动 SoftAP

配置 SoftAP 是通过 hostapd 进行的。hostapd 和 wpa_supplicant 类似，它包含了 AP 端的各种认证协议、连接流程，wpa_supplicant 是 STA 端的。

步骤 1. 修改 hostapd.conf 文件。

hostapd 进程需要使用 hostapd.conf 配置文件，在配置文件里设置 SSID、信道、加密方式等。配置文件的内容举例如下：

- 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
```

hostapd 是开源代码，配置文件中的参数可以参考网络资源。



ht_capab 可以配置是否支持 40M 带宽，带有[SHORT-GI-40][HT40-]或者[SHORT-GI-40][HT40+]即支持 40M 带宽。当 channel 小于 6 时，用[SHORT-GI-40][HT40+]，当 channel 大于等于 6 时，用[SHORT-GI-40][HT40-]。

mt7601u、ap6181、ap6212、ap6212a、ap6214a、ap6255 的 SoftAP 模式不支持 40M 带宽，所以 ht_capab 设置如下：

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

步骤 2. 启动 hostapd 进程。

执行 Shell 命令：

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

执行完后，用 ps 命令查看一下 hostapd 进程是否存在，存在表示工作正常，用 STA 设备可以搜索到 SoftAP。如果没有，可以增加 hostapd 的打印级别，从 log 看出现什么问题，如：

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

----结束

Broadcom 的 WiFi 还可以通过 wl 工具进行配置，配置方法请参考 sample 和 wl 的 help 信息。

3.3.3 开启 udhcpd

执行 Shell 命令：

```
ifconfig wlan0 192.168.1.1  
busybox udhcpd -fS /etc/Wireless/udhcpd.conf
```

请确保/etc/Wireless/udhcpd.conf 文件存在，并且配置的网段为 192.168.1.x。执行完后，用 STA 设备可以扫描和连接该 SoftAP，如果能成功连接且 ping 通网关表示此 AP 配置成功。

3.3.4 卸载驱动

卸载驱动的方法和 STA 模式一样。

3.4 适配国家或区域

不同的国家或区域，采用的频率范围有些不同，比如 2.4GHz 频段，美国支持 1~11 信道，中国和欧洲支持 1~13 信道，日本支持 1~14 信道。5GHz 频段也类似。WiFi 需要根据产品上市的国家或区域做相应的配置，以适用于该国家的频率范围。

不同的 WiFi 配置方法不一样，比如 RTL8188ftv：rtl8188ftv 配置成美国，在加载驱动时带上 rtw_channel_plan=0x22 参数：

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



本文档无法列出所有的 WiFi 对应的国家配置，产品开发中请询问模组厂家或者 WiFi 芯片厂家。



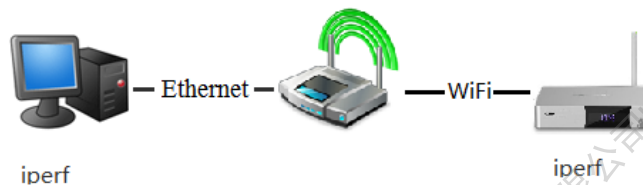
4 测试

4.1 吞吐量测试

吞吐量测试可以反映 WiFi 的性能，是目前芯片厂家、模组厂家、设备厂家普遍使用的测试方法，具有很高的认同度。吞吐量测试最常使用的工具是 iperf。

测试环境为 PC 机通过有线和 AP 连接，单板通过 WiFi 和 AP 连接，单板和 PC 机可以互相 ping 通。在 PC 机和单板上都有 iperf 工具。假设 PC 机的 IP 地址为 192.168.1.100，单板的 IP 地址为 192.168.1.101。

图4-1 吞吐量测试组网环境



4.1.1 TCP 发送吞吐量测试

发送吞吐量测试操作如下：

步骤 1 PC 机上命令行进入 iperf 工具目录，执行：

```
iperf -s
```

步骤 2 单板上通过 Shell 进入 iperf 工具目录，执行：

```
iperf -c 192.168.1.100 -t 10 -i 1
```




图4-2 发送吞吐量测试示例

```
# 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
```

其中，iperf -s 表示启动服务端，iperf -c 192.168.1.100 表示启动客户端，连接 192.168.1.100，-t 10 表示测试 10 秒钟，-i 1 表示每隔 1 秒钟打印一次结果。

最后打印的“0.0-10.0 sec 85.7 MBytes 71.6 Mbits/sec”表示这 10 秒钟的平均吞吐量为 71.6Mbps。

----结束

4.1.2 TCP 接收吞吐量测试

接收吞吐量测试操作如下：

步骤 1 单板上通过 Shell 进入 iperf 工具目录，执行：

```
iperf -s
```

步骤 2 PC 机上命令行进入 iperf 工具目录，执行：

```
iperf -c 192.168.1.101 -t 10 -i 1 -w 1M
```

图4-3 接收吞吐量测试示例

```
# 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
```



iperf 也可以进行 UDP 测试，在有些 PC 机上单个 UDP 线程进行了限速，因此需要开启多个线程。

SoftAP 的吞吐量测试类似。



注意

有些 PC 机，由于安装了一些软件，对速率会有影响，一定要确保 PC 机没有速率的瓶颈。WEP 安全模式不能使用 802.11n 协议，因此速率比较低，一般只有 20+Mbps。

----结束

4.1.3 UDP 发送吞吐量测试

发送吞吐量测试操作如下：

步骤 1 PC 机上命令行进入 iperf 工具目录，执行：

```
iperf -s -u -l 32k
```

步骤 2 单板上通过 Shell 进入 iperf 工具目录，执行：

```
iperf -c 192.168.1.100 -u -t 10 -i 1 -l 32k -b 100M
```

----结束

4.1.4 UDP 接收吞吐量测试

接收吞吐量测试操作如下：

步骤 1 单板上通过 Shell 进入 iperf 工具目录，执行：

```
iperf -s -u
```

步骤 2 PC 机上命令行进入 iperf 工具目录，执行：

```
iperf -c 192.168.1.101 -u -t 10 -i 1 -l 32k -b 100M
```

----结束

4.2 射频指标测试

吞吐量测试可以反映 WiFi 的性能，在产品开发中必须要做的工作。有些有条件的公司还会进行射频指标测试，它可以准确的验证 WiFi 射频是否达标。因为模组厂家在生产模组时是必须做的工作，所以如果采用的是模组，这项工作可选的。但由于硬件设计时，有可能地线不干净、板上干扰等原因，影响 WiFi 射频性能，因此建议有条件的公司要进行该项测试。



射频指标包括：接收灵敏度、信道功率抑制、发送功率、发送载频容差、丢包率、EVM、接收杂散、发送杂散等。

测试仪器包含：频谱分析仪、功率测量仪、网络分析仪等。

测试方法较复杂，可以参考测量仪器说明书。