

SigmaStar

WiFi 使用指南

文档版本 01

目录

1	内核配置说明.....	3
1.1	配置 WEXT.....	3
1.1	配置 CFG80211.....	3
2	WiFi 基本操作.....	4
2.1	STA 模式操作.....	4
2.1.1	检查 WiFi 设备.....	4
2.1.2	加载驱动.....	4
2.1.3	连接 AP.....	5
2.2	AP 模式操作.....	6
2.2.1	检查 WiFi 设备.....	6
2.2.2	加载驱动.....	6
2.2.3	Hostapd 配置和启动 AP.....	6
2.2.4	开启 udhcpd.....	7
2.3	monitor 模式操作.....	7
2.3.1	开启 monitor.....	7
2.3.2	切换信道.....	7
2.3.3	停止 monitor.....	7
3	吞吐量测试.....	8
3.1	发送吞吐量测试.....	8
3.2	接收吞吐量测试.....	8
4	常见问题及解决办法.....	8
4.1	工具使用.....	8
4.1.1	iw tools.....	8
4.1.2	WiFi 分析仪.....	8
4.1.3	OmniPeek.....	9

1 内核配置说明

1.1 配置 WEXT

Wext 是 Wireless Extension,是内核中 WIFI 驱动和用户态进程的标准接口,iwpriv 命令依赖这个配置

```
.config - Linux/x86 3.18.30 Kernel Configuration
Networking support Wireless
Wireless
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenu ----).
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features.
Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded
<M> module < > module capable
[ ] nl80211 testmode command
[ ] enable developer warnings
[ ] cfg80211 regulatory debugging
[ ] cfg80211 certification onus
[*] enable powersave by default
[ ] cfg80211 DebugFS entries
[ ] use statically compiled regulatory rules database
[ ] cfg80211 wireless extensions compatibility
< > Generic IEEE 802.11 Networking Stack (mac80211)
<Select> < Exit > < Help > < Save > < Load >
```

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

```
.config - Linux/x86 3.18.30 Kernel Configuration
Networking support Wireless
Wireless
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenu ----).
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features.
Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded
<M> module < > module capable
--- Wireless
< > cfg80211 - wireless configuration API
[ ] nl80211 testmode command
[ ] enable developer warnings
[ ] cfg80211 regulatory debugging
[ ] cfg80211 certification onus
[*] enable powersave by default
[ ] cfg80211 DebugFS entries
[ ] use statically compiled regulatory rules database
+ (+)
<Select> < Exit > < Help > < Save > < Load >
```

2 WIFI 基本操作

2.1 STA 模式操作

2.1.1 检查 WiFi 设备

开启 WiFi 前，检查一下 WiFi 设备是否正常，可以通过看系统开机 log 和 lsusb 命令来确认，如果找到说明已经识别到 WiFi 设备，如果没有识别到则 USB 驱动没加载或 WiFi 未插入，没上电或者已经损坏。

```
BC disable
[USB] soc:Mstar-ehci-1 irq --> 95
soc:Mstar-ehci-1 soc:Mstar-ehci-1: EHCI Host Controller
soc:Mstar-ehci-1 soc:Mstar-ehci-1: new USB bus registered, assigned bus number 2
soc:Mstar-ehci-1 soc:Mstar-ehci-1: irq 95, io mem 0xfd284800
usb usb2: New USB device found, idVendor=1d6b, idProduct=0002
usb usb2: New USB device strings: Mfr=3, Product=2, SerialNumber=1
usb usb2: Product: EHCI Host Controller
usb usb2: Manufacturer: Linux 3.18.30 ehci_hcd
usb usb2: SerialNumber: mstar
hub 2-0:1.0: USB hub found
hub 2-0:1.0: 1 port detected
usbcore: registered new interface driver usb-storage
ifconfig: SIOCSIFHWADDR: Cannot assign requested address
==20150512==> hub_port_init 1 #0
Plug in USB Port1
usb 2-1: new high-speed USB device number 2 using soc:Mstar-ehci-1
usb 2-1: New USB device found, idVendor=1b20, idProduct=8888
usb 2-1: New USB device strings: Mfr=16, Product=32, SerialNumber=0
usb 2-1: Product: SigmaStarWIFI
usb 2-1: Manufacturer: SigmaStar inc
```

```
[/ext/demo]##
[/ext/demo]## lsusb
Bus 002 Device 002: ID 1b20:8888
Bus 001 Device 001: ID 1d6b:0002
Bus 002 Device 001: ID 1d6b:0002
[/ext/demo]##
```

2.1.2 加载驱动

步骤 1 加载 cfg80211 驱动,SSW101B 驱动使用了 cfg80211,cfg80211 是编译成 ko 文件，需要先加载 cfg80211.ko,进入 ko 存放目录，执行命令：

```
insmod cfg80211.ko
```

步骤 2 加载 WiFi 驱动，进入 ko 存放目录，执行命令：

```
insmod ssw101_wifi_usb.ko
```

步骤 3 查看驱动是否加载成功，执行命令：

```
iwconfig
```

```

[/ext/demo/wifi]## iwconfig
[Sstar_log]:ieee80211_netdev_ioctl:cmd err
[Sstar_log]:ieee80211_netdev_ioctl:cmd err
[Sstar_log]:ieee80211_netdev_ioctl:cmd err
wlan0 IEEE 802.11abgn ESSID:off/any [Sstar_log]:ieee80211_netdev_ioctl:cmd err
:off/any
[Sstar_log]:ieee80211_netdev_ioctl:cmd err
Mode:Managed Access Point: Not-Associated Tx-Power=0 dBm
Retry short limit:7 RTS thr:off Fragment thr:off
Encryption key:off
Power Management:on

p2p0 IEEE 802.11abgn ESSID:off/any
Mode:Managed Access Point: Not-Associated Tx-Power=0 dBm
Retry short limit:7 RTS thr:off Fragment thr:off
Encryption key:off
Power Management:on

lo no wireless extensions.

eth0 no wireless extensions.

```

如果看到有一个 wlan0 网口，说明驱动已经初始化成功，WiFi 可以使用

步骤 4 启动 WiFi 网口，执行命令：

`ifconfig wlan0 up`

2.1.3 连接 AP

连接 AP 是通过 wpa_supplicant 进程进行的，wpa_supplicant 是开源代码，它包含了 WEP WPA/WPA2 WPA-PSK/WPA2-PSK 等协议。

步骤 1 启动 wpa_supplicant 进程，执行命令：

`wpa_supplicant -Dnl80211 -iwlan0 -c /ext/demo/wifi/wpa_supplicant.conf &`

参数说明：-iwlan0 表示使用 wlan0 网口

-Dnl80211 表示使用 cfg80211 接口

-c/xx/xx/xx/wpa_supplicant.conf 是 wpa_supplicant 的配置文件，要保证改文件已经存在。

执行完后，用 ps 命令看下 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 看出出现什么问题，命令如下：

`wpa_supplicant -Dnl80211 -iwlan0 -c /ext/demo/wifi/wpa_supplicant.conf -ddd &`

步骤 2 连接 AP

```

[/ext/demo/wifi]## wpa_cli -p /ext/demo/run/wpa_supplicant/ -i wlan0 add_network
0
[/ext/demo/wifi]## wpa_cli -p /ext/demo/run/wpa_supplicant/ -i wlan0 set_network
0 ssid "TP_GW" 要连接的AP SSID
OK
[/ext/demo/wifi]## wpa_cli -p /ext/demo/run/wpa_supplicant/ -i wlan0 set_network
0 psk "12345678" AP 的密码
OK
[/ext/demo/wifi]## wpa_cli -p /ext/demo/run/wpa_supplicant/ -i wlan0 enable_netw
ork 0

[Sstar_log]:Sstar_hw_scan:if_id(0)
OK

```

-p 后面的参数是 wpa_supplicant.conf 配置文件 ctrl_interface=/ext/demo/run/wpa_supplicant(路径可以自己修改)

步骤 3 判断成功连接

Log 可以看到成功连接信息

```

## [Sstar_log]:scan start band(0),(14)
[Sstar_log]:hw_priv->scan.status 0
[Sstar_log]:Sstar_scan_work:end(0)
[Sstar_log]:Sstar_hw_scan:if_id(0)
[Sstar_log]:Sstar_hw_scan:scan, delay suspend
[Sstar_log]:_ieee80211_scan_completed:489
[Sstar_log]:scan start band(1),(6)
cfg80211: Calling CRDA to update world regulatory domain
[Sstar_log]:hw_priv->scan.status 0
[Sstar_log]:Sstar_scan_work:end(0)
wlan0: SME: Trying to authenticate with 8c:a6:df:0c:7f:7f (SSID='TP_GW' freq=2462 MHz
)
[Sstar_log]:ieee80211_mgd_auth:(8c:a6:df:0c:7f:7f),ssid(TP_GW)
[Sstar_log]:Sstar_set_priv_queue_cap:[0],queue_cap[224]
[Sstar_log]:wlan0: authenticated
wlan0: Trying to[Sstar_log]:ieee80211_start_connecting_work:bssid(8c:a6:df:0c:7f:7f)
associate with 8c:a6:df:0c:7f:7f (SSID='TP_GW' freq=2462 MHz)
[Sstar_log]:wlan0: associated
[Sstar_log]:[8c:a6:df:0c:7f:7f]:20M channel
wlan0: Associated with 8c:a6:df:0c:7f:7f
wlan0: CTRL-EVENT-SUBNET-STATUS-UPDATE status=0
wlan0: WPA: Key negotiation completed with 8c:a6:df:0c:7f:7f [PTK=CCMP GTK=CCMP]
wlan0: CTRL-EVENT-CONNECTED - Connection to 8c:a6:df:0c:7f:7f completed [id=0 id_str=
]

```

已经连接上 mac 为 8c:a6:df:0c:7f:7f AP

通过命令来确认是否连接上，执行: `iwconfig`

```

[/ext/demo/wifi]## iwconfig
[Sstar_log]:ieee80211_netdev_ioctl:cmd err
[Sstar_log]:ieee80211_netdev_ioctl:cmd err
[Sstar_log]:ieee80211_netdev_ioctl:cmd err
wlan0      IEEE 8[Sstar_log]:ieee80211_netdev_ioctl:cmd err
02.11abgn  ESSID[Sstar_log]:ieee80211_netdev_ioctl:cmd err
: "TP_GW"
Mode:Managed Frequency:2.462 GHz Access Point: 8C:A6:DF:0C:7F:7F
Bit Rate=1 Mb/s Tx-Power=30 dBm
Retry short limit:7 RTS thr:off Fragment thr:off
Encryption key:off
Power Management:on
Link Quality=42/70 Signal level=-68 dBm
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0

p2p0      IEEE 802.11abgn ESSID:off/any
Mode:Managed Access Point: Not-Associated Tx-Power=30 dBm
Retry short limit:7 RTS thr:off Fragment thr:off
Encryption key:off
Power Management:on

```

连接上的SSID

步骤 4 获取 IP 地址，执行命令:`udhcpc -i wlan0`

2.2 AP 模式操作

2.2.1 检查 WiFi 设备

跟 STA 模式一样

2.2.2 加载驱动

跟 STA 模式一样，执行命令:

`ifconfig p2p0 up`

2.2.3 Hostapd 配置和启动 AP

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 信道密码，配置文件如下：

```
interface=p2p0
#bridge=br0
ctrl_interface_group=0
driver=nl80211
ieee80211n=1
ssid=IPC1
hw_mode=g
channel=4
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=3
wpa_passphrase=12345678
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

步骤 2 启动 `hostapd` 进程，执行命令：

```
hostapd hostapd.conf &
```

执行完成后，用 `ps` 命令查看下 `hostapd` 进程是否存在，存在表示工作正常，用 STA 设备可以搜索到 AP，如果没有，可以增加 `hostapd` 的打印等级，从 `log` 看什么问题。如：

```
hostapd hostapd.conf -ddddd &
```

2.2.4 开启 udhcpd

执行命令：

```
udhcpd -fs udhcpd.conf
```

2.3 monitor 模式操作

2.3.1 开启 monitor

执行命令：

```
ifconfig wlan0 down
iwconfig wlan0 mode monitor
ifconfig wlan0 up
ifconfig wlan0 0.0.0.0
iwconfig wlan0 channel [1,13] 选择一个信道监听
```

2.3.2 切换信道

```
iwconfig wlan0 channel [1,13] //选择一个信道
```

2.3.3 停止 monitor

```
ifconfig wlan0 down
iwconfig wlan0 mode manager
```

3 吞吐量测试

吞吐量测试可以反映 WiFi 性能，具有很高的认可度，吞吐量测试最常用的工具是 iperf
测试环境为 PC 通过有线和 AP 连接，IPC 通过 WIFI 和 AP 连接，IPC 和 PC 可以互相 ping 通

3.1 发送吞吐量测试

发送吞吐量测试操作如下：

步骤 1 PC 上命令进入 iperf 目录执行：

```
iperf -s
```

步骤 2 IPC 进入 iperf 目录执行：

```
iperf -c 192.168.1.10 -t 120 -i 1
```

其中，iperf -s 表示启动的服务器，

```
iperf -c 192.168.1.10 -t 120 -i 1 //客户端命令  
-c 192.168.1.10 /连接服务器 192.168.1.10(PC 端 IP)  
-t 120 //表示测试 120 秒  
-i 1 //表示每隔 1 秒打印一次
```

3.2 接收吞吐量测试

步骤 1 IPC 进入 iperf 目录执行：

```
iperf -s
```

步骤 2 PC 命令进入 iperf 目录执行：

```
iperf -c 192.168.1.10 -t 120 -i 1
```

4 常见问题及解决办法

4.1 工具使用

4.1.1 iw tools

iwconfig iwlist iwpriv 命令

iwconfig 看 WiFi 驱动是否加载成功，有 wlan0 表示驱动加载成功，查看无线网络状态，包括 AP 的 SSID，AP 的 MAC 地址加密方式，信号质量

iwlist

扫描 AP，连接不上 AP 时可以扫描看 AP 是否存在

iwpriv

WiFi 需要开启 lmac debug 信息时要打开命令

执行：

```
iwpriv wlan0 fwcmd ccastart
```

```
iwpriv wlan0 fwcmd set_rate_print,1
```

4.1.2 WIFI 分析仪

WiFi 分析仪是安卓应用，可以统计 AP 的数量，SSID，采用的信道，信号强度，用来分析 WiFi 当前的干扰情况

4.1.3 OmniPeek

Omnipeek 是一款网络抓包工具，可以抓到 WiFi 的空中包，用来分析 WiFi 协议