CV180X & CV181X Wi-Fi User Guide

Version: 2.0.0

Release date: 2023-02-08

Contents

Revision History

Revision	Date	Description
1.0.0	2022/06/10	Initial version
2.0.0	2023/02/08	Compatible with cv180x/cv181x
2.0.1	2023/07/19	Compatible with dual_os

1 Disclaimer

Terms and Conditions

The document and all information contained herein remain the CVITEK Co., Ltd' s ("CVITEK") confidential information, and should not disclose to any third party or use it in any way without CVITEK' s prior written consent. User shall be liable for any damage and loss caused by unauthority use and disclosure.

CVITEK reserves the right to make changes to information contained in this document at any time and without notice.

All information contained herein is provided in "AS IS" basis, without warranties of any kind, expressed or implied, including without limitation mercantability, non-infringement and fitness for a particular purpose. In no event shall CVITEK be liable for any third party' s software provided herein, User shall only seek remedy against such third party. CVITEK especially claims that CVITEK shall have no liable for CVITEK' s work result based on Customer' s specification or published shandard.

Contact Us

Address Building 1, Yard 9, FengHao East Road, Haidian District, Beijing, 100094, China

Building T10, UpperCoast Park, Huizhanwan, Zhancheng Community, Fuhai Street, Baoan District, Shenzhen, 518100, China

Phone +86-10-57590723 +86-10-57590724

Website https://www.sophgo.com/

Forum https://developer.sophgo.com/forum/index.html

2 Overview

Wi-Fi is the trademark of Wi-Fi alliance.

It is a wireless LAN technology based on IEEE 802.11 standard.

Mobile terminals with Wi-Fi function can connect to the Internet within the signal coverage, so as to reduce the trouble of cable erection and improve the convenience of use.

At present, many processor manufacturers provide various types of Wi-Fi processor solutions with different drivers, but these drivers are not universal.

In addition, the functions and performance supported by different driver versions may vary.

We need to ask Wi-Fi solution providers to provide appropriate Linux Wi-Fi drivers for porting.

Linux platform has generality for different Wi-Fi processor driver and operation mode.

This document will respectively introduce how CV180X uses Realtek solution for driver porting and adaptation on different interfaces (such as USB or SDIO), as well as related operations.

The Wi-Fi module used in this document is

- AP6201BM (Broadcom bcm43013c1), supports SDIO interface.

3 Configuration Description

3.1 Kernel Configuration

Edit build/boards/{processor_name}/{board_name}/linux/ cvitek_{board_name}_defconfig,

Ex. build/boards/cv1801c_wevb_0009a_spinor/linux/cvitek_cv1801c_wevb_0009a_spinor_defconfig, enable Wifi-related Configuration (the red part is marked as the basic configuration that must be enabled, the other parts are enabled as needed).

```
#
# WiFi
#
CONFIG_WLAN=y
CONFIG_CFG80211=y
CONFIG_CFG80211_DEFAULT_PS=y
CONFIG_CFG80211_CRDA_SUPPORT=y
# CONFIG_CFG80211_WEXT is not set
# CONFIG_MAC80211 is not set
# CONFIG_MAC80211_HAS_RC is not set
# CONFIG_MAC80211_RC_MINSTRE is not set
# CONFIG_MAC80211_RC_MINSTREL_HT is not set
# CONFIG_MAC80211_RC_DEFAULT_MINSTREL is not set
# CONFIG_MAC80211_RC_DEFAULT="minstrel_ht"
CONFIG_CVI_WIFI_PIN=y
CONFIG_WIRELESS=y
CONFIG_WLAN_VENDOR_REALTEK=y
CONFIG_RTL8189FS=m    # This option selects the corresponding driver based on the wifi chip
used (adaptation is required)
# CONFIG_WEXT_CORE is not set
# CONFIG_WEXT_PROC is not set
```

Since the Wi-Fi interface is SDIO, it needs to be turned on Build/boards/cv180x/cv1801c_wevb_0009a_spinor/dts_riscv/{board_name}.dtsi

Confirm the wifisd node configuration as follows:

```
wifisd:wifi-sd@4320000 {
   compatible = "cvitek,cv181x-sdio";
```

(continues on next page)

(continued from previous page)

```
    bus-width = <4>;
    reg = <0x0 0x4320000 0x0 0x1000>;
    reg_names = "core_mem";
    src-frequency = <375000000>;
    min-frequency = <400000>;
    max-frequency = <50000000>;
    64_addressing;
    reset_tx_rx_phy;
    non-removable;
    pll_index = <0x7>;
    pll_reg = <0x300207C>;
    no-mmc;
    no-sd;
};
```

Also edit build/boards/default/dts/cv180x/{board_name}_{bga or qfn}.dtsi ,

ex. build/boards/default/dts/cv180x/cv180x_asic_bga.dtsi or the dtsi file of the corresponding project to confirm no-delete configuration of wifi-sd@5000000 node, example as follows:

```
/* /delete-node/ wifi-sd@5000000; */ /* comment or delete this row */
/delete-node/ i2c@04010000;
/delete-node/ i2c@04020000;
/delete-node/ ethernet@04520000;
/delete-node/ i2s@04120000;
...
```

3.2 Configure SDIO

Please refer to the relevant chapters of SDIO in <Peripheral Driver Operation Guide>. The SDIO IO voltage is 3.3V. Make sure the Wi-Fi module IO voltage is the same as the SDIO voltage.

3.3 Configure Pinmux

If the interface of Wi-Fi module is SDIO, the SDIO pinmux configuration can be set for the CV180X/1X by adding the required pinmux settings in the cvi_board_init.c file located at build/boards/{processor_name}/{board_name}/u-boot/, ex. (Following is the EVB configuration of 181xH, and the pin configured is the pin from SoC to the processor_en of WiFi module (according to the circuit diagram))

```
int cvi_board_init(void)
{
    ...
    //#########WIFI
    pinmux_config(PINMUX_SDIO1);
```

(continues on next page)

(continued from previous page)

```
    PINMUX_CONFIG(JTAG_CPU_TCK, XGPIOA_18);
    ...
    return 0;
}
```

For pin configuration details, please refer to u-boot-2021.10/board/cvitek/cv181x/board.c

3.4 Configure Wifi GPIO

Since the processor_en pin of the Wi-Fi module is controlled by a GPIO on the SOC, in order to operate this GPIO, we specially made a simple module and used the interface provided by the module to power up and down the wifi in the wifi driver. GPIO used by wifi can be specified through the device tree: (where the wakeup function is not used and can be removed; The poweron pin corresponds to the pinmux set in the previous section)

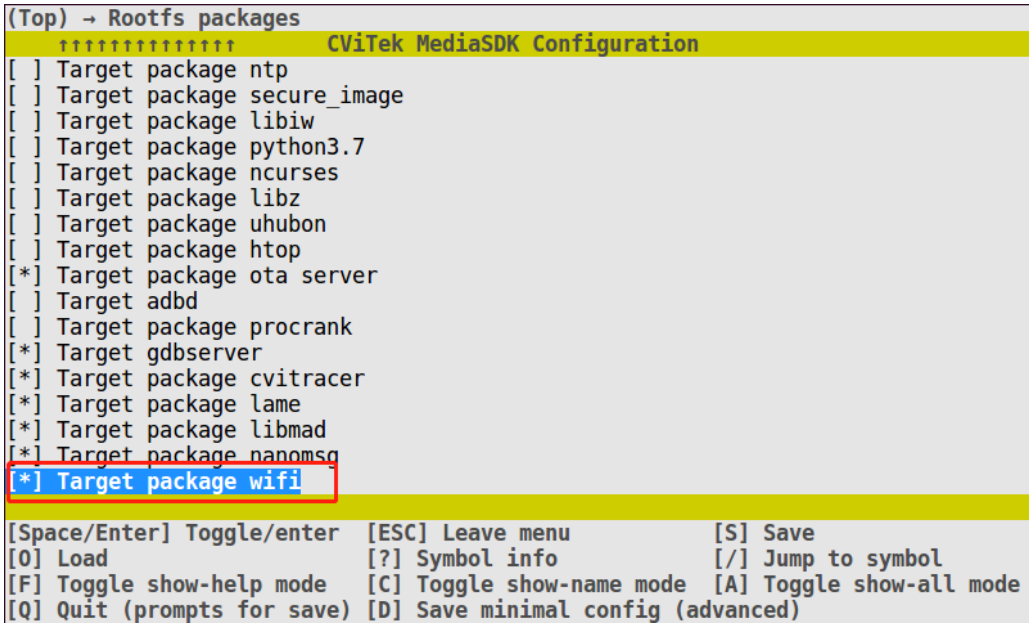
```
wifi_pin {
    compatible = "cvitek,wifi-pin";
    poweron-gpio = <&porta 18 GPIO_ACTIVE_HIGH>;
    wakeup-gpio = <&porte 7 GPIO_ACTIVE_HIGH>;
};
```

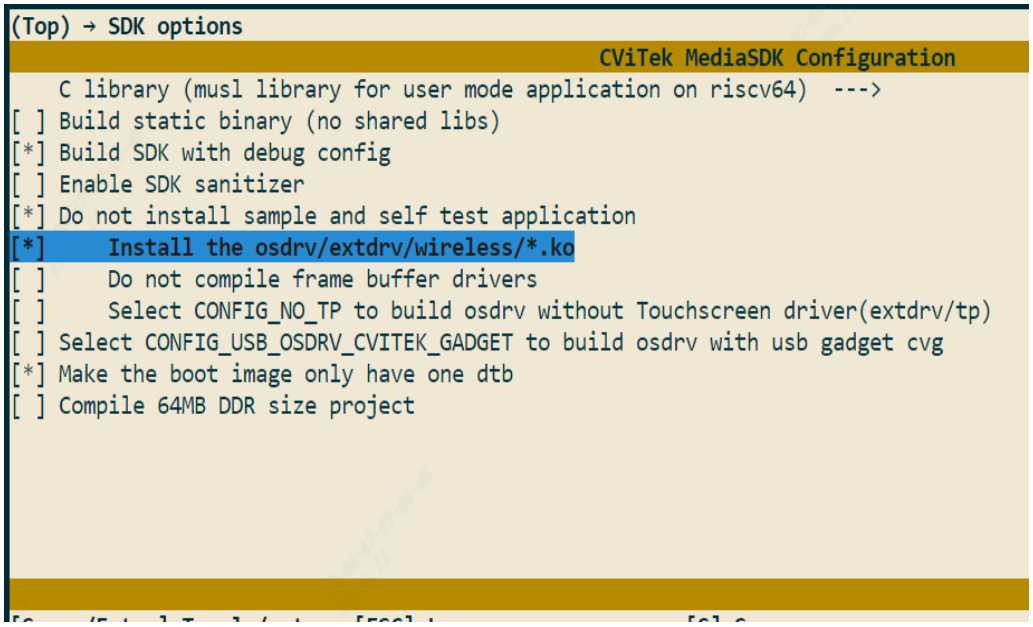
This configuration is in the file build/boards/default/{processor_name}/{processor_name}_base.dtsi

4 Wi-Fi Tools

wpa_supplicant, wpa_cli, hostapd and other open source tools are needed when users operate Wi-Fi.

- Select Rootfs packages -> Target package wifi and turn on the wireless through the menu mode, archive away and start compiling





- Or edit build/boards/{processor_name}/{board_name}/{board_name}_defconfig to enable the following options (as shown below), then execute defconfig $CHIP_$BOARD through command mode to automatically configure

```
#
# Rootfs packages
#
...
CONFIG_TARGET_PACKAGE_WIFI=y
CONFIG _CP_EXT_WIRELESS=y
# end of Rootfs packages
```

If users want to update to the latest version, please go to http://w1.fi/releases or http://www.linuxfromscratch.org/blfs/view/svn/basicnet/wireless_tools.html to obtain it, and install it to rootfs yourself.

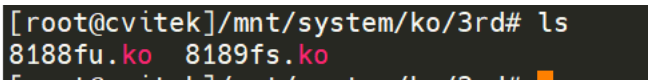
5 Wi-Fi Basic Operation

5.1 STA Mode Basic Operation

5.1.1 Loading Driver

Step 1. load the driver

check if the three files in the red box below are available under /mnt/system/ko/3rd



Note that since the wifi driver is not placed in the Linux source directory tree, it cannot be built-in, and can only be compiled into ko.

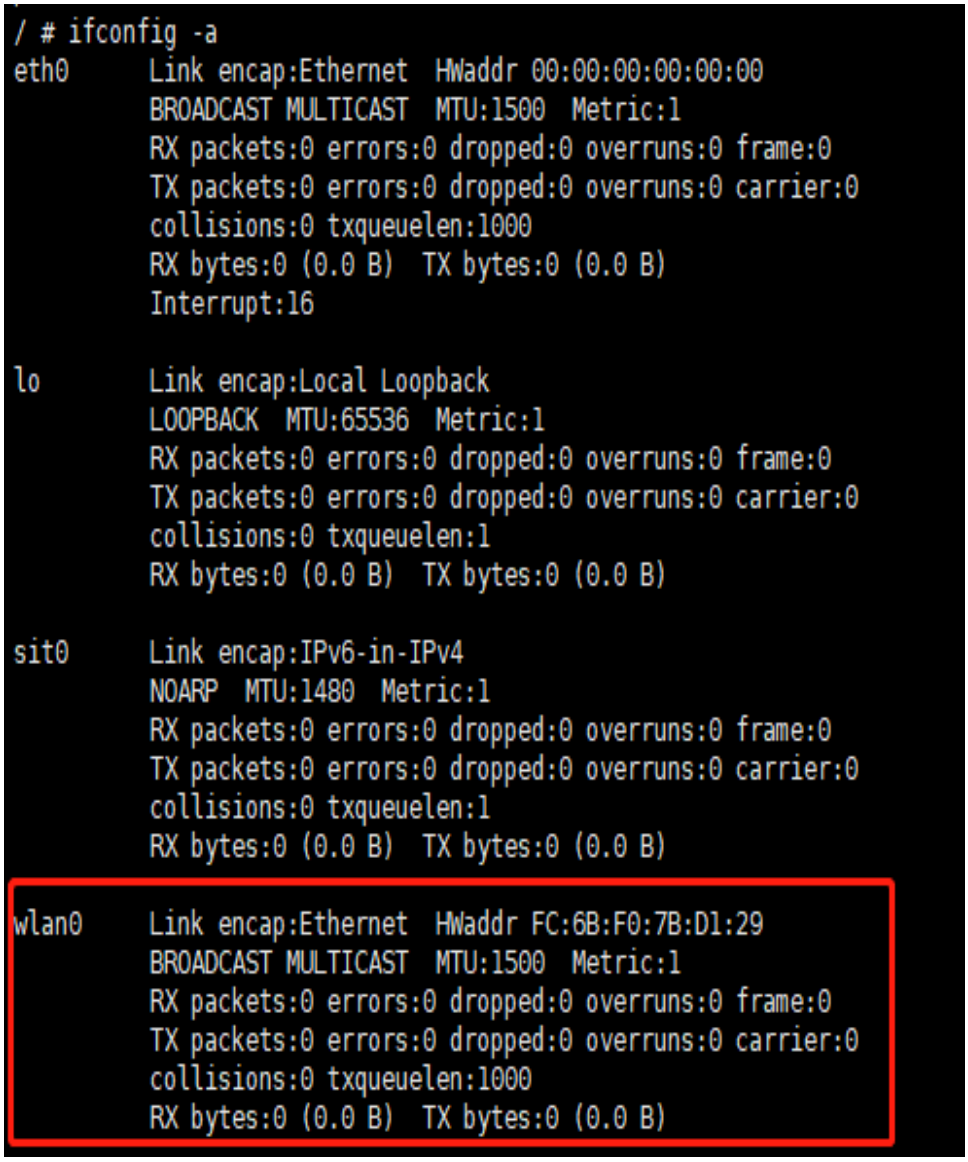
```
insmod /mnt/system/ko/3rd/8189fs.ko
```

Step 2. check whether the driver is loaded successfully

excute the shell command:

```
ifconfig -a
```

If the loading is successful, you can see the wlan0 interface after executing the shell command.



5.1.2 Start Wi-Fi and Connect AP

Step 1. Start wlan0

execute the shell command:

```
ifconfig wlan0 up
```

Step 2. Start wpa_supplication

excute the shell command:

```
echo "ctrl_interface=/var/run/wpa_supplicant" >/tmp/wpa_supplicant.
↪conf

wpa_supplicant -iwlan0 -Dnl80211 -c/tmp/wpa_supplicant.conf &
```

- -iwlan0 means to use wlan0 interface

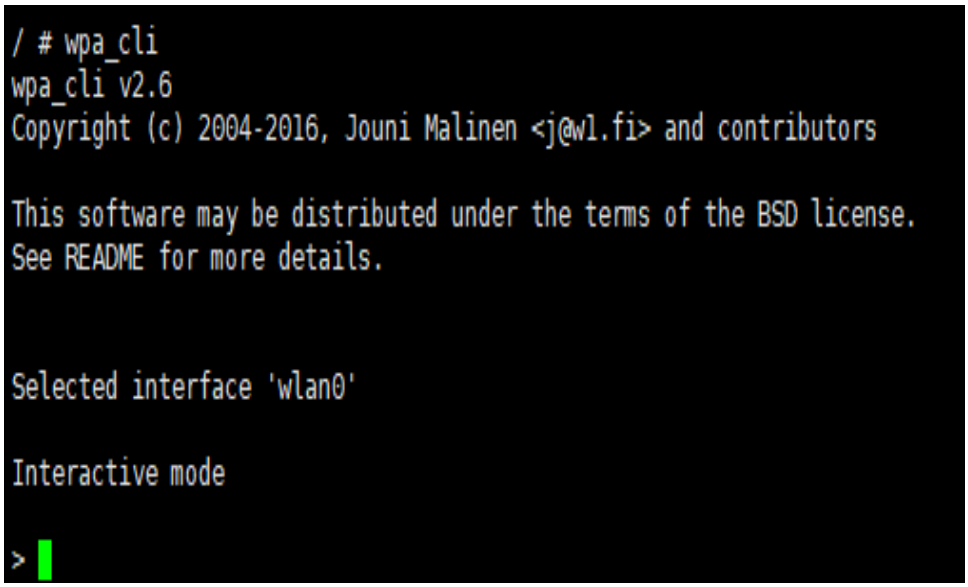
- -Dnl80211 means to use cfg80211 interface

Step 3. Start wpa_cli

excute the shell command:

```
wpa_cli -i wlan0
```

A ">" prompt will appear when the execution is successful.



Step 4. Scan nearby AP

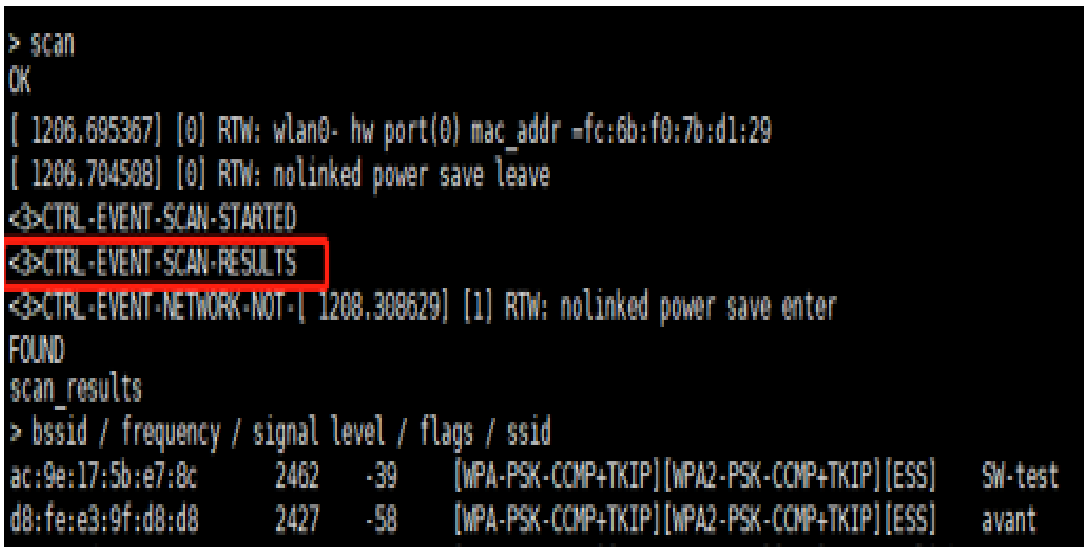
After the ">" prompt symbol, execute the following command:

```
scan
```

After "CTRL-EVENT-SCAN-RESULTS" appears, execute

```
scan_results
```

and then the scanning results can be obtained.



Step 5. Connect AP

- Connect the AP configured as WPA-PSK/WPA2-PSK authentication and encryption type.

  1. After the ">" prompt symbol, execute the following command to obtain the network ID (0 in this example):

```
add_network
```

  2. Configure the SSID of the network (the SSID in this example is "SW-test" , obtained from step 4)

```
set_network 0 ssid "SW-test"
```

  3. Configure the network encryption method and password (assuming that the SW-test password is 012345678)

```
set_network 0 psk "012345678"
```

  4. Start the network

```
select_network 0
```

  5. Observe whether you have received CTRL-EVENT-CONNECTED. If so, it means the connection is established. BTW, "status" command can be used to query the connection status.

```
> add_network
0
> set_network 0 ssid "SW-test"
OK
> set_network 0 psk "012345678"
OK
> select_network 0
OK
<3>CTRL-EVENT-SCAN-STARTED
<3>CTRL-EVENT-SCAN-RESULTS
wlan0: Trying to associate with ac:9e:17:5b:e7:8c (SSID='SW-test' freq=2462 MHz)
<3>Trying to associate with ac:9e:17:5b:e7:8c (SSID='SW-test' freq=2462 MHz)
[  171.729764] [0] IPv6: ADDRCONF(NETDEV_CHANGE): wlan0: link becomes ready
wlan0: Associated with ac:9e:17:5b:e7:8c
wlan0: CTRL-EVENT-SUBNET-STATUS-UPDATE status=0
<3>Associated with ac:9e:17:5b:e7:8c
<3>CTRL-EVENT-SUBNET-STATUS-UPDATE status=0
wlan0: WPA: Key negotiation completed with ac:9e:17:5b:e7:8c [PTK=CCMP GTK=TKIP]
wlan0: CTRL-EVENT-CONNECTED - Connection to ac:9e:17:5b:e7:8c completed [id=0 id_str=]
<3>WPA: Key negotiation completed with ac:9e:17:5b:e7:8c [PTK=CCMP GTK=TKIP]
<3>CTRL-EVENT-CONNECTED - Connection to ac:9e:17:5b:e7:8c completed [id=0 id_str=]

> > status
bssid=ac:9e:17:5b:e7:8c
freq=2462
ssid=SW-test
id=0
mode=station
pairwise_cipher=CCMP
group_cipher=TKIP
key_mgmt=WPA2-PSK
wpa_state=COMPLETED
address=fc:6b:f0:7b:d1:29
```

  6. Enter "quit" to exit wpa_cli. To get the dynamic IP address, execute the shell command as follows

```
udhcpc -b -i wlan0 -R &
```

7. Execute the ping command to check whether the network is operating normally

ex.

```
ping 8.8.8.8
```

- Connect the AP configured as open system

  The steps and configuration are the same for WPA-PSK/WPA2-PSK authentication and encryption type. Only when configuring the network encryption mode, you need to input the following command:

```
set_network 0 key_mgmt NONE
```

5.1.3 Turn off Wi Fi and Unload Driver

Step 1. Execute the shell command as follows:

```
ifconfig wlan0 down
```

Step 2. Execute the shell command as follows:

```
rmmod 8189fs.ko
```

5.2 SoftAP Mode Basic Operation

5.2.1 Loading Driver

The same as STA mode. Please refer to *5.1.1 loading driver* .

5.2.2 hostapd Configuration, udhcpd Configuration and Starting SoftAP

To start SoftAP, you need to start hostapd first. Similar to wpa_supplicant, hostpad can be used to configure various authentication protocols and connection processes of AP.

Step 1. Start the hostpad.

Execute shell command

```
ifconfig wlan0 192.168.1.1 up
hostapd /etc/network/hostapd.conf -B -i wlan0
```

Step 2. Start udhcpd to assign dynamic IP to the Wi-Fi device by executing shell command

```
udhcpd /etc/network/udhcpd.conf
```

Remark:

- Users can modify hostapd.conf to configure the ssid, channel, encryption and authentication mode of SoftAP. The document is located in /ramdisk/rootfs/overlay/{processor_name}/etc/network in the SDK package or /etc/network on the platform. For example, users can configure the AP name and login password by modifying ssid and wpa_passphrase.

  ```
  interface=wlan0
  ctrl_interface=/var/run/hostapd
  ssid=CV180X_EVB
  channel=6
  wpa=3
  wpa_passphrase=012345678
  ```

  The significance of other parameters can be referred to http://manpages.ubuntu.com/manpages/bionic/man5/udhcpd.conf.5.html

- Users can modify udhcpd.conf to configure the IP range provided by SoftAP. The document is located in /ramdisk/rootfs/overlay/{processor_name}/etc/network in the SDK package or /etc/network on the platform.

  ```
  # The start and end of the IP lease block
  start     192.168.1.10    #default: 192.168.0.20
  end       192.168.1.254    #default: 192.168.0.254
  ```

6 Tests

6.1 Throughput Test

The performance of Wifi can be observed and tuned through throughput test. The most commonly used tool for throughput testing is iperf3. The test environment is as follows:

PC is connected with wireless AP by wired Ethernet, while CVITEK platform is connected with wireless AP through Wi-Fi. Suppose in this example, the IP address of PC is 192.168.0.11, and that of CVITEK platform is 192.168.0.112. Both PC and CVITEK platform have iperf3 tools.

6.1.1 Sending Throughput Test

Step 1. Enter iperf3 tool directory on PC and execute the following command:

```
iperf3 -s
```

Step 2. The platform executes shell commands as follows:

- Test TCP protocol

  ```
  iperf3 -c 192.168.0.11 -t 10
  ```

- Test UDP protocol

```
iperf3 -c 192.168.0.11 -t 10 -u -b 100M -l 32k
```

```
/ # iperf3 -c 192.168.0.11 -t 10
Connecting to host 192.168.0.11, port 5201
[  5] local 192.168.0.112 port 50194 connected to 192.168.0.11 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  1.42 MBytes  11.9 Mbits/sec    0    138 KBytes
[  5]   1.00-2.00   sec   941 KBytes  7.71 Mbits/sec   75    114 KBytes
[  5]   2.00-3.00   sec   627 KBytes  5.14 Mbits/sec    0    130 KBytes
[  5]   3.00-4.00   sec   941 KBytes  7.71 Mbits/sec    0    137 KBytes
[  5]   4.00-5.00   sec   941 KBytes  7.71 Mbits/sec    0    138 KBytes
[  5]   5.00-6.00   sec  1.23 MBytes  10.3 Mbits/sec    0    138 KBytes
[  5]   6.00-7.00   sec   941 KBytes  7.71 Mbits/sec    0    140 KBytes
[  5]   7.00-8.00   sec  1.53 MBytes  12.8 Mbits/sec    0    147 KBytes
[  5]   8.00-9.00   sec   314 KBytes  2.57 Mbits/sec    1    158 KBytes
[  5]   9.00-10.00  sec   753 KBytes  6.17 Mbits/sec    0    178 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  9.51 MBytes  7.98 Mbits/sec   76             sender
[  5]   0.00-10.00  sec  8.89 MBytes  7.46 Mbits/sec                  receiver
```

The results can be obtained through iperf3 sending test, as shown in the figure above. The meaning of each parameter can be explained by executing "iperf3 -h".

It can be seen from the above figure that the average throughput of 10 seconds is 7.98 Mbps.

6.1.2 Receiving Throughput Test

Step 1. Execute the shell instruction on the platform as follows:

```
iperf3 -s
```

Step 2. Enter iperf3 tool directory on PC and execute the following commands:

- Test TCP protocol

```
iperf3 -c 192.168.0.112 -t 10
```

- Test UDP protocol

```
iperf3 -c 192.168.0.112 -t 10 -u -b 100M -l 32k
```