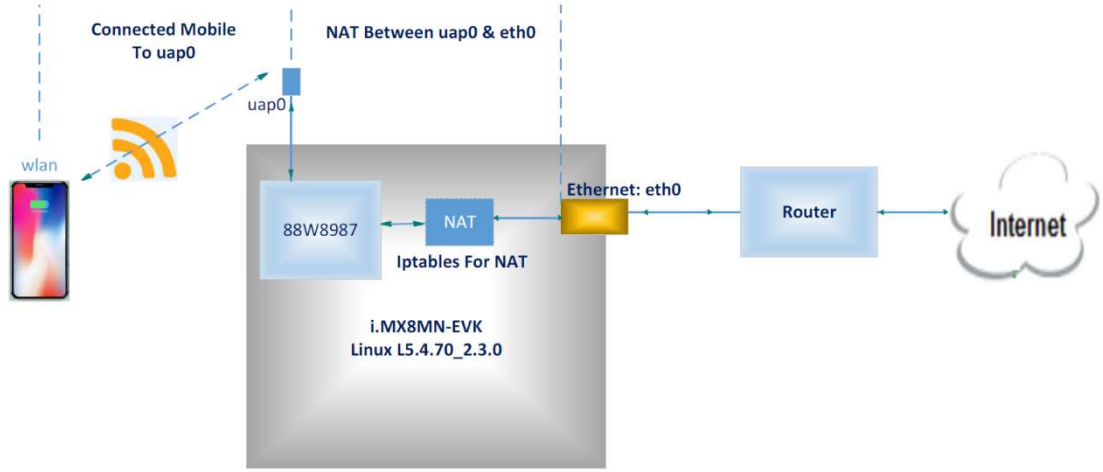
AP Configrations Based On 88W8987 of iMX8MN-EVK And L5.4.70_2.3.0

The article will describe how to configure Access Point Based On NXP platform and WIFI chipset step by step. Users can easily make her AP based on NXP WIFI module work normally by following steps in the article.

1. Environment for the validation

- Hardware Platform
  i.MX8MN-EVK
- Software
  - Kernel version: L5.4.70_2.3.0
  - rootfs : imx-image-multimedia
- WiFi module
  AW-CM358SM: NXP 88W8987 chipset

2. Diagram for Connections



[Note]

(1) **Arrow** : directions of data streaming

(2) **Mobile**: used to connect uap0 of 88W8987

(3) **NAT**: created by iptables , used to share internet access between uap0 and eth0

(4) **eth0**: ethernet on i.MX8MN-EVK, it's IP address will be obtained from Router.

Before configuring AP, we must ensure eth0 can access internet. Users can use ping commands to test it.

3. Preparations

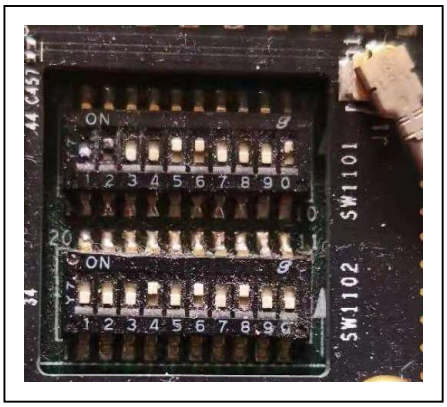
(1) Downloading demo image for i.mx8mn-evk from nxp official website
https://www.nxp.com/design/software/embedded-software/i-mx-software/embedded-linux-for-i-mx-applications-processors:IMXLINUX?tab=Design_Tools_Tab
Download it : L5.4.70_2.3.0_MX8MN

(2) Downloading uuu tool
https://github.com/NXPmicro/mfgtools/releases
uuu tool supports 2 versions, one is for linux, the other is for windows 10.
We will uuu for windows.

(3) Downloading L5.4.70_2.3.0 image to i.MX8MN-EVK board

- Decompressing demo image to a directory.

- Copy uuu.exe to the directory.
- Open windows command line, and enter into the directory by DOS command
- Connecting i.MX8MN-EVK USB Type-C port to PC USB
- Configuring board to be USB download mode



[Note]
1. Normal boot mode
   See SW1101, current setting is normal boot mode. PIN1=OFF & PIN2=ON
2. USB download mode
   Only need to set PIN1=ON and PIN2=OFF
Board will work in USB download mode.

- Run “uuu uuu.auto-imx8mnevk” command to begin downloading image.
- When it is done, power off board, and set PIN1&PIN2 of SW1101 to normal boot mode.
- Connecting Debug UART to PC USB, and running terminal software on windows.
  Debug UART on board is a microUSB connector, users should connect it to PC USB via a microUSB to USB TYPE-AM cable. For teminal software, user can select one she likes, like SecureCRT.
- Power On i.MX8MN-EVK board, logs will be showed in terminal software.
  After booting is done, input “root” to log in board.

4. Configurations For Access Point

No.1 Step: Configuring or Creating the following files

(1) /etc/hostapd.conf

```
# If we want to use 2.4G, it's contents should be :
Deleting original one , and Using “nano /etc/hostapd.conf” to create a New one.
root@imx8mnevk:~# nano /etc/hostapd.conf
------------------
interface=uap0
ssid=NXP-uap0
hw_mode=g
channel=1
own_ip_addr=192.168.2.2    # this is the uap0 ip address
------------------
# If we want to use 5G, it's contents should be :
------------------
interface=uap0
ssid=NXP-uap0
hw_mode=a
channel=40
ieee80211n=1
own_ip_addr=192.168.2.2    # this is the uap0 ip address
```

(2) /etc/systemd/network/hostapd.network

If the file can't be found in the directory, use touch command to create it.
add these lines to the file.

```
-------------
[Match]
Name=uap0
[Network]
Address=192.168.2.1/24
DHCPServer=yes
--------------
```

(3) /lib/systemd/system/hostapd.service

Use nano to open the file, and add these 3 lines to the end:

root@imx8mnevk:~# nano /lib/systemd/system/hostapd.service

BindsTo=sys-subsystem-net-devices-uap0.device

After=sys-subsystem-net-devices-uap0.device

After=enable-wifi.service

Save and Exit. Then run the command below:

root@imx8mnevk:~# systemctl --system daemon-reload

(4) /etc/systemd/system/enable-wifi.service

nano /etc/systemd/system/enable-wifi.service

```
[Unit]
Description=Enable wifi
BindsTo=sys-subsystem-net-devices-uap0.device
After=sys-subsystem-net-devices-uap0.device
[Service]
Type=oneshot
ExecStart=/usr/bin/connmanctl enable wifi
ExecStop=
RemainAfterExit=yes
[Install]
WantedBy=multi-user.target
```

(5) /etc/sysctl.conf

root@imx8mnevk:~# nano /etc/sysctl.conf

net.ipv4.ip_forward=1

net.ipv6.conf.all.forwarding=1

Save and exit

sysctl -p (enable the new configuration)

[note]

cat /proc/sys/net/ipv4/ip_forward (check if net.ipv4.ip_forward is valid)

echo 1 >/proc/sys/net/ipv4/ip_forward (enable it manually)

Or

```
# sysctl -w net.ipv4.ip_forward=1
```

(6) /etc/udhcpd.conf

root@imx8mnevk:~# nano /etc/udhcpd.conf

```
……
# The start and end of the IP lease block
start 192.168.2.1
end 192.168.2.24
interface uap0 #default: eth0
max_leases 20 #default: 254
remaining yes #default: yes
auto_time 7200 #default: 7200 (2 hours)
decline_time 3600 #default: 3600 (1 hour)
conflict_time 3600 #default: 3600 (1 hour)
offer_time 60 #default: 60 (1 minute)
min_lease 60 #defult: 60
lease_file /etc/udhcpd.leases
opt dns 192.168.0.1          #router IP
option subnet 255.255.255.0
opt router 192.168.2.1            # uap0 gw IP
option domain local
option lease 864000
……
```

Save and Exit

Then create /etc/udhcpd.leases file:

```
# touch /etc/udhcpd.leases
```

Step 2 Configurations For NAT between uap0 & eth0

```
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
# iptables -A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
# iptables -A FORWARD -i uap0 -o eth0 -j ACCEPT
# iptables-save > /etc/iptables/iptables.rules
```

[note]

The file /lib/systemd/system/iptables.service should be like below, if the file can't be find in the directory, create it, please! And add following lines to the file.

```
[Unit]
Description=Packet Filtering Framework
Before=network-pre.target
Wants=network-pre.target

[Service]
Type=oneshot
ExecStart=/usr/sbin/iptables-restore /etc/iptables/iptables.rules
ExecReload=/usr/sbin/iptables-restore /etc/iptables/iptables.rules
RemainAfterExit=yes
```

Step 3 Enabling iptables and Restart board

```
# systemctl --system daemon-reload
# systemctl enable iptables
# reboot
```

5. Loading WIFI driver

```
root@imx8mnevk:~# modprobe moal mod_para=nxp/wifi_mod_para.conf
[    61.616597] wlan: Loading MWLAN driver
[    61.719651] vendor=0x02DF device=0x9149 class=0 function=1
[    61.725238] Attach moal handle ops, card interface type: 0x105
[    61.732963] SD8987: init module param from usr cfg
[    61.737898] card_type: SD8987, config block: 0
[    61.742379] cfg80211_wext=0xf
[    61.745370] wfd_name=p2p
[    61.747904] max_vir_bss=1
[    61.750541] cal_data_cfg=none
[    61.753522] drv_mode = 7
[    61.756055] ps_mode = 2
[    61.758514] auto_ds = 2
[    61.760975] fw_name=nxp/sdiouart8987_combo_v0.bin
[    61.765733] SDIO: max_segs=128 max_seg_size=65535
[    61.770450] rx_work=1 cpu_num=4
[    61.773627] Attach mlan adapter operations.card_type is 0x105.
[    61.779854] wlan: Enable TX SG mode
[    61.783370] wlan: Enable RX SG mode
[    61.788508] Request firmware: nxp/sdiouart8987_combo_v0.bin
[    62.202434] Wlan: FW download over, firmwarelen=527468 downloaded 527468
[    63.181126] WLAN FW is active
[    63.184104] on_time is 63183695000
[    63.218170] fw_cap_info=0x181c3f03, dev_cap_mask=0xffffffff
[    63.223775] max_p2p_conn = 8, max_sta_conn = 8
[    63.259070] wlan: version = SD8987---16.92.10.p208-MXM5X16210-GPL-(FP92)
[    63.273048] wlan: Driver loaded successfully
```

```
root@imx8mnevk:~# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:04:9f:06:da:96
          inet addr:192.168.0.109  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::204:9fff:fe06:da96/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:103 errors:0 dropped:0 overruns:0 frame:0
          TX packets:68 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:13187 (12.8 KiB)  TX bytes:9342 (9.1 KiB)
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:94 errors:0 dropped:0 overruns:0 frame:0
          TX packets:94 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7402 (7.2 KiB)  TX bytes:7402 (7.2 KiB)
mlan0     Link encap:Ethernet  HWaddr 20:4e:f6:20:98:f1
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
p2p0      Link encap:Ethernet  HWaddr 22:4e:f6:20:98:f1
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

uap0      Link encap:Ethernet  HWaddr 20:4e:f6:20:99:f1
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
root@imx8mnevk:~#
```

6. Configuring uap0 IP/gateway and DHCP service

```
root@imx8mnevk:~# ifconfig uap0 192.168.2.2 netmask 255.255.255.0 up
root@imx8mnevk:~# route add default gw 192.168.2.1
```

7. Starting DHCP Service

```
root@imx8mnevk:~# udhcpd -S /etc/udhcpd.conf -f &
```

8. Starting relevant service

```
root@imx8mnevk:~# connmanctl enable wifi
root@imx8mnevk:~# systemctl enable hostapd
root@imx8mnevk:~# systemctl start hostapd
root@imx8mnevk:~# systemctl daemon-reload
root@imx8mnevk:~# systemctl enable enable-wifi.service
```

9. Test

Up to now, all configuraions have been done. Users can connect mobile to uap0 and check if it can access internet.

-Reference Materials

[1] https://developer.toradex.com/knowledge-base/wi-fi-access-point-mode
[2] NXP User manual，UM1152.pdf

NXP CAS-TIC Wirless MCU Team
Weidong Sun
04-16-2021