BPi R2 & OpenWrt 기반
Access Point Dev Guide

Slowboot(chunghan.yi@gmail.com)

05/01/2020

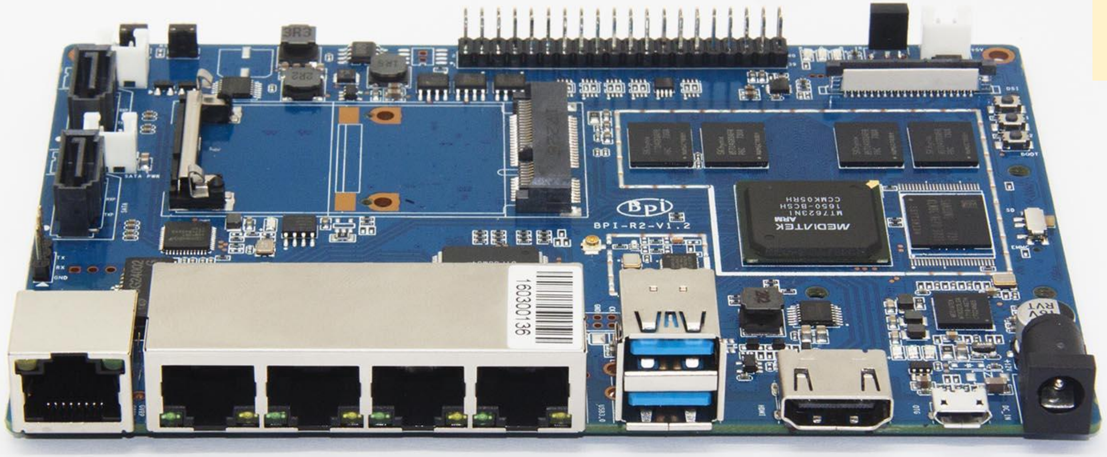
Contents

- 1. BPi R2
- 2. AP S/W Architecture
- 3. OpenWrt/LEDE
- 4. WireGuard 올리기
- 5. 802.11ac 무선랜 구동하기
- 6. OpenWrt/LEDE 참고 사항

1. BPi R2(1) – Target Board

Key Features

- MediaTek MT7623N, Quad-core ARM Cortex-A7
- Mali 450 MP4 GPU
- 2G DDR3 SDRAM
- Mini PCIE interface
- SATA interface
- 4x Gigabit LAN 1x Gigabit WAN



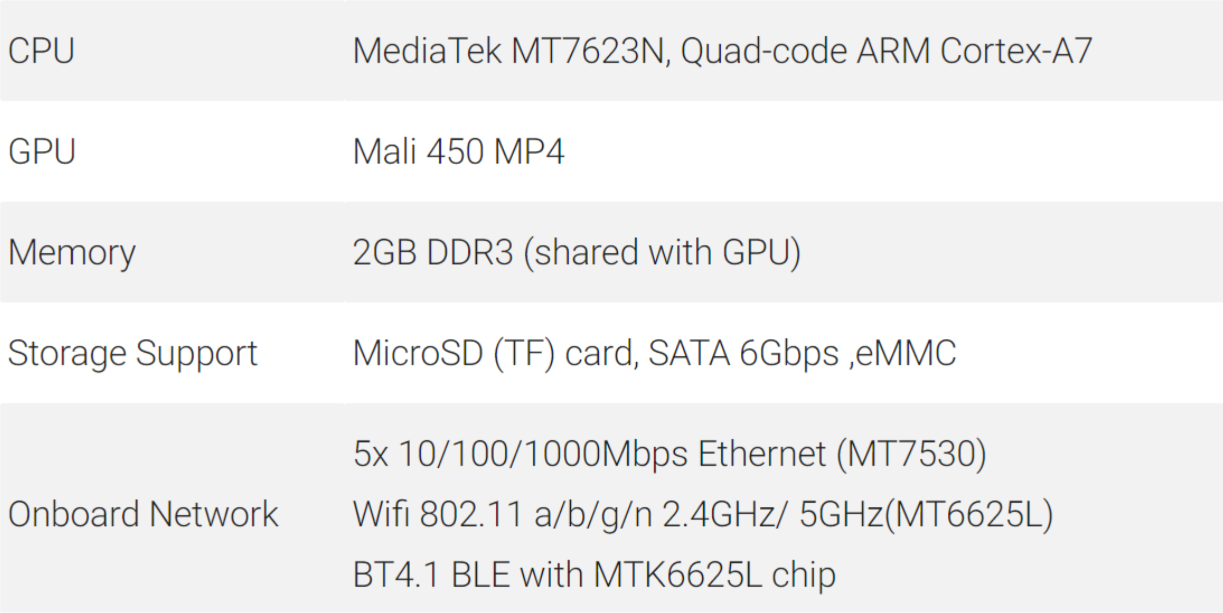
http://www.banana-pi.org/r2.html

802.11ac miniPCIe card를 별도로 장착해야 함.

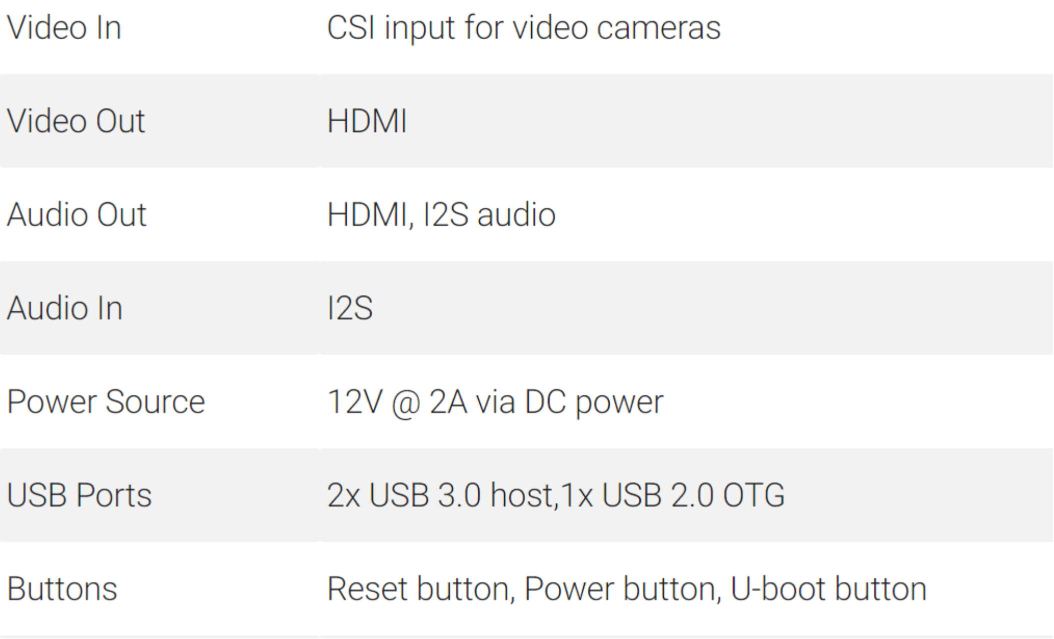
US $89.50

https://pt.aliexpress.com/store/product/Newest-arrive-Banana-PIBPI-R2-Opensource-Router/302756_32823351577.html?spm=a2g03.12010608.0.0.6e25663exMMEwO

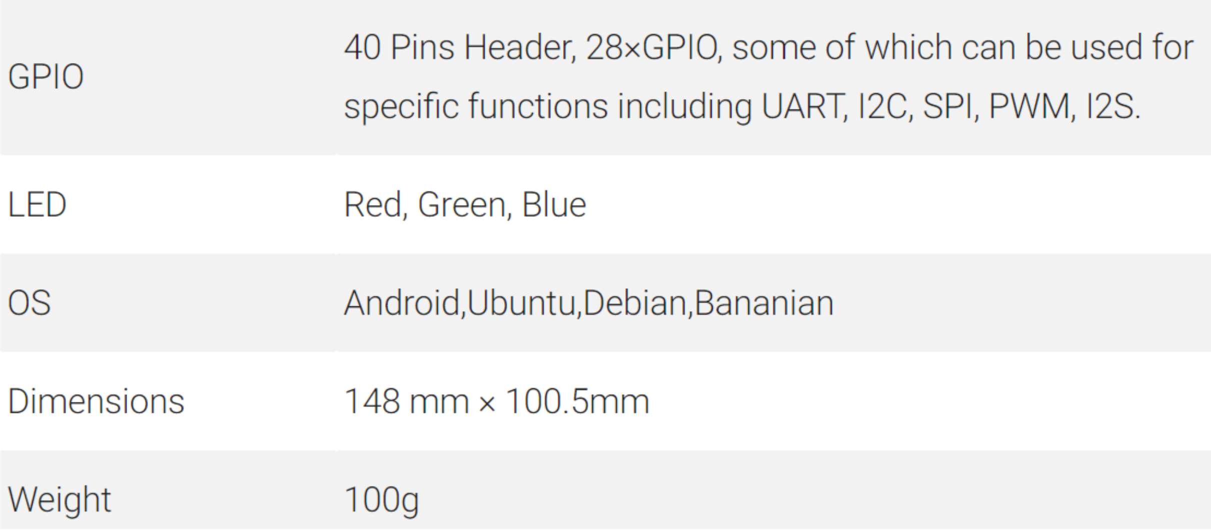
1. BPi R2(2) - Specification



CPU	MediaTek MT7623N, Quad-code ARM Cortex-A7
GPU	Mali 450 MP4
Memory	2GB DDR3 (shared with GPU)
Storage Support	MicroSD (TF) card, SATA 6Gbps ,eMMC
Onboard Network	5x 10/100/1000Mbps Ethernet (MT7530) Wifi 802.11 a/b/g/n 2.4GHz/ 5GHz(MT6625L) BT4.1 BLE with MTK6625L chip



Video In	CSI input for video cameras
Video Out	HDMI
Audio Out	HDMI, I2S audio
Audio In	I2S
Power Source	12V @ 2A via DC power
USB Ports	2x USB 3.0 host,1x USB 2.0 OTG
Buttons	Reset button, Power button, U-boot button

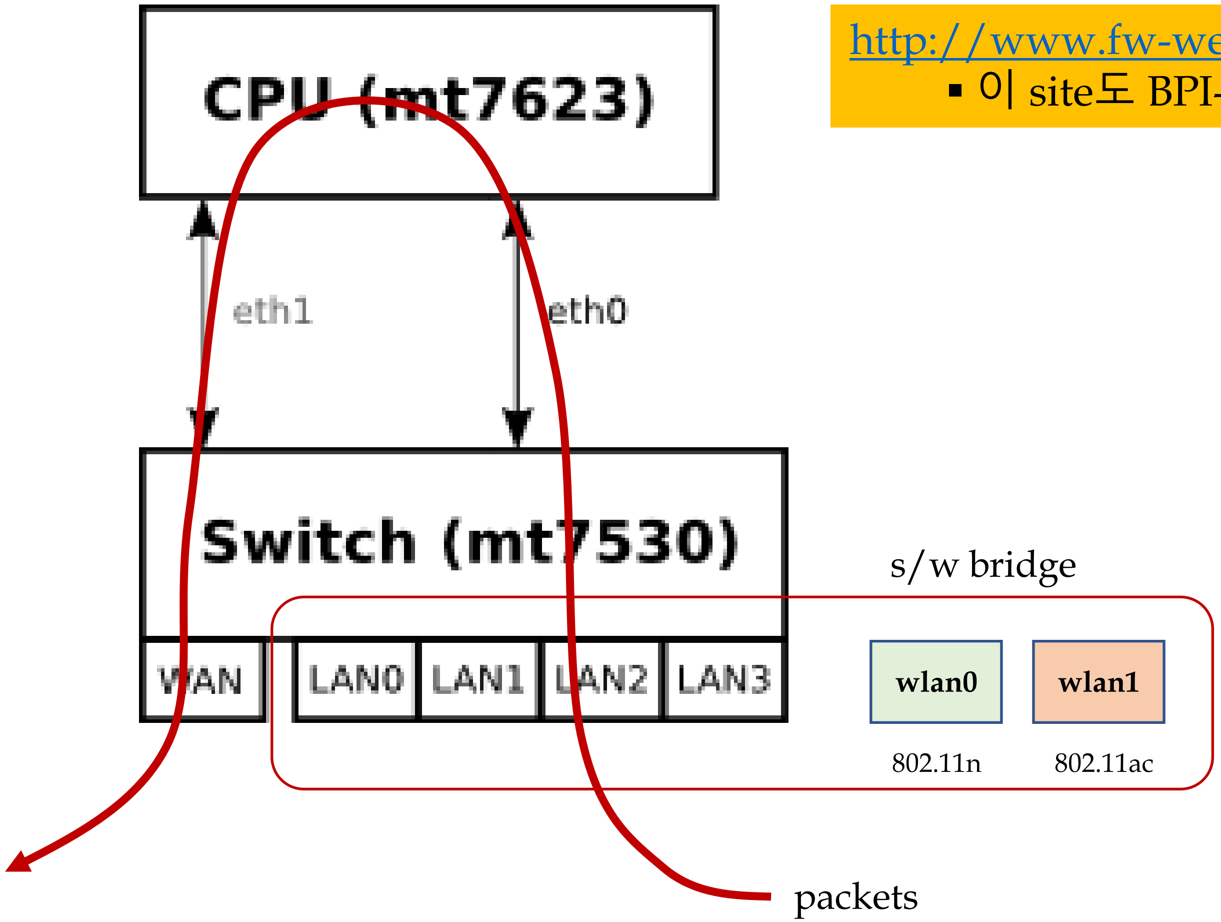


GPIO	40 Pins Header, 28×GPIO, some of which can be used for specific functions including UART, I2C, SPI, PWM, I2S.
LED	Red, Green, Blue
OS	Android,Ubuntu,Debian,Bananian
Dimensions	148 mm × 100.5mm
Weight	100g

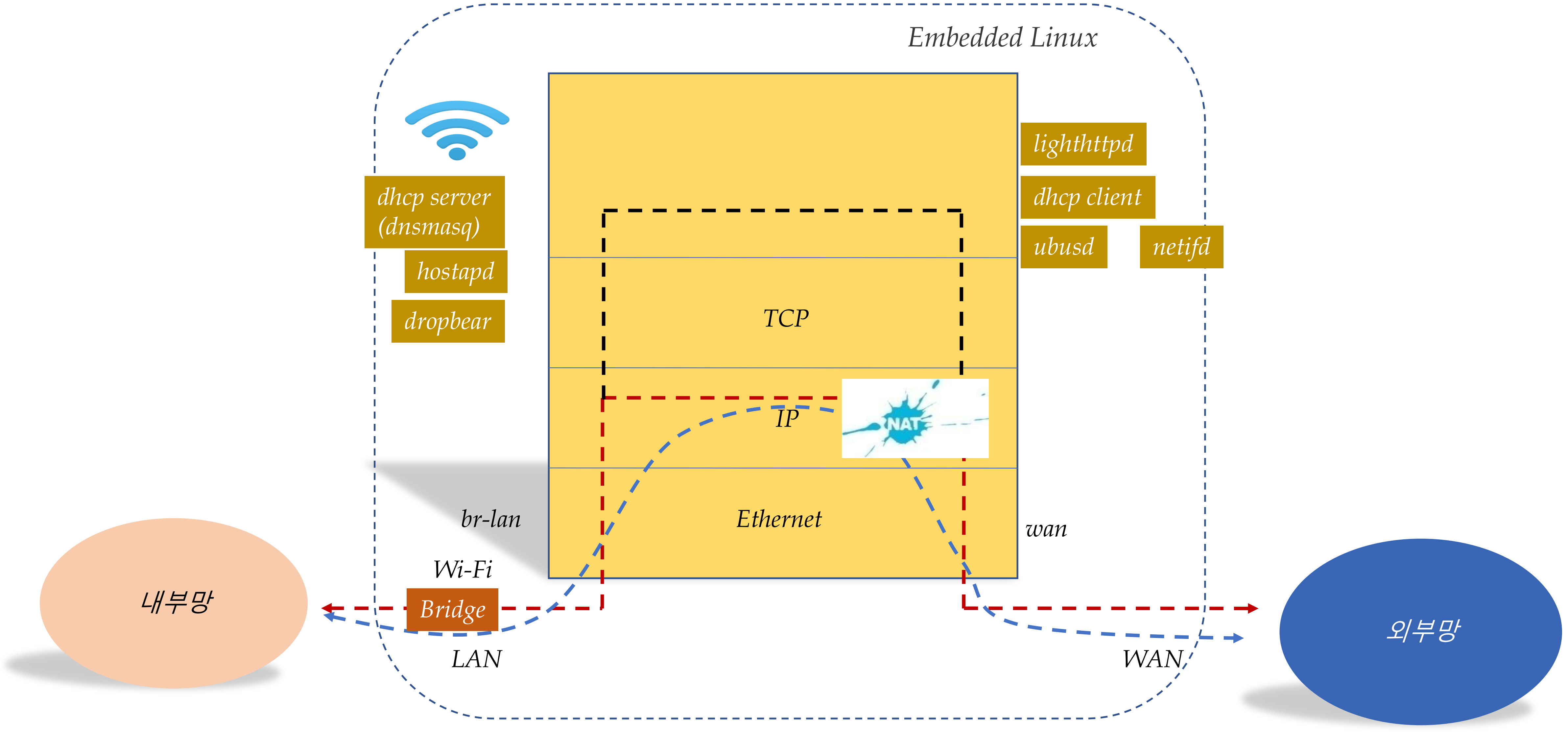
1. BPi R2(3) – H/W Architecture

http://www.fw-web.de/dokuwiki/doku.php?id=en:bpi-r2:start

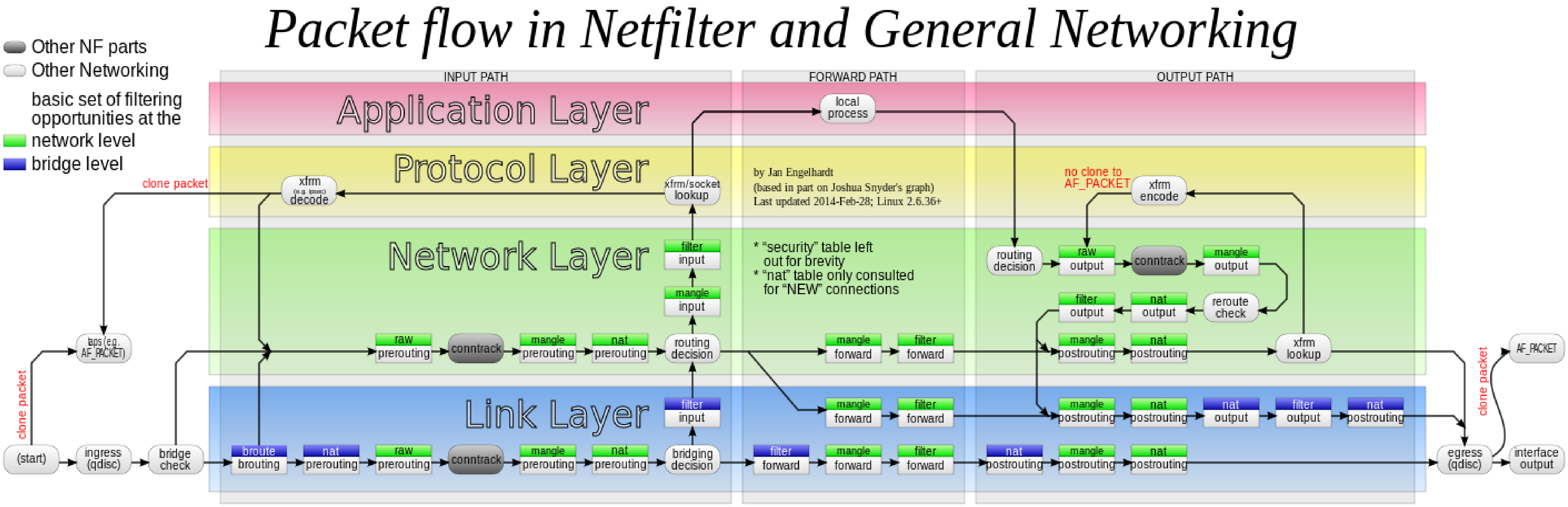
- 이 site도 BPI-R2 관련 재밌는 내용이 많이 있음.



2. AP S/W Architecture(1)



2. AP S/W Architecture(2) – Netfilter & L2 Bridge



Packet flow in Netfilter and General Networking

- Other NF parts
- Other Networking
- 🟩 network level — basic set of filtering opportunities at the network level
- 🟦 bridge level

INPUT PATH

FORWARD PATH

OUTPUT PATH

Application Layer

Protocol Layer

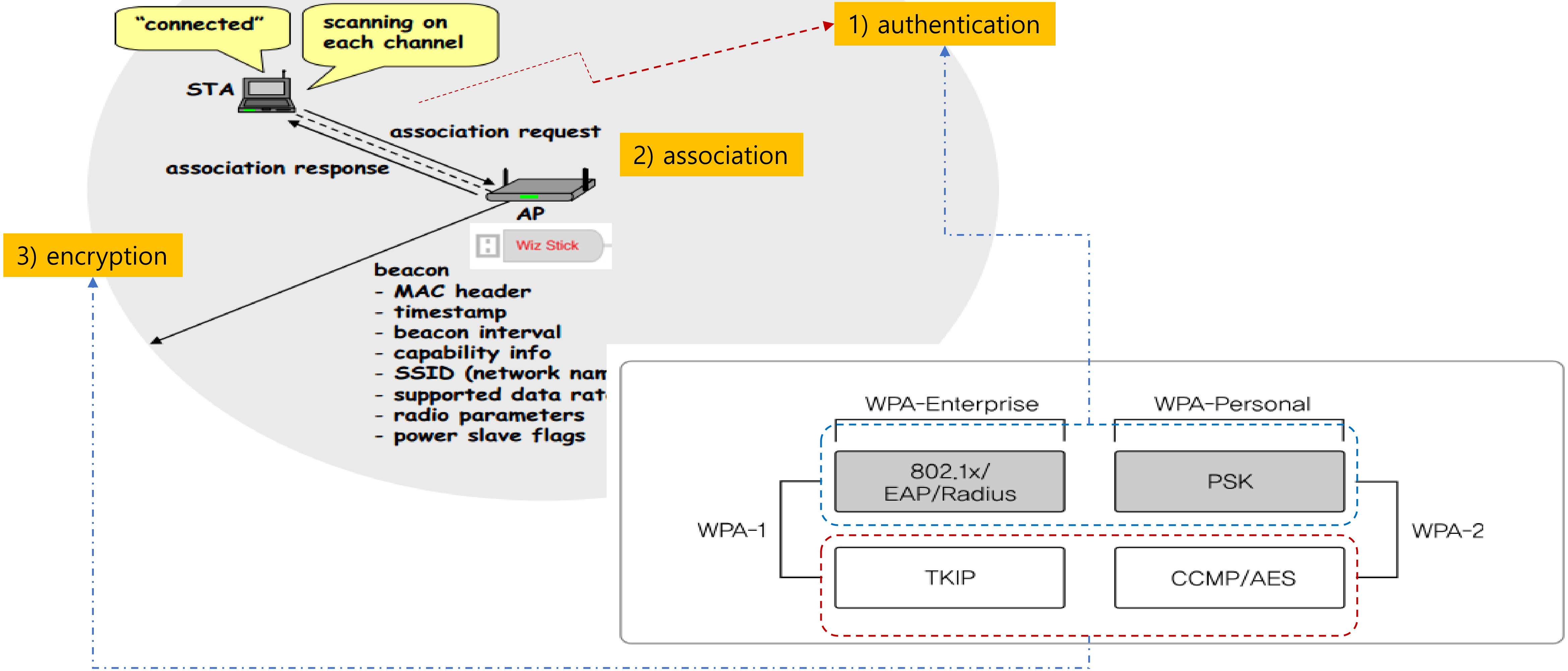
Network Layer

Link Layer

by Jan Engelhardt
(based in part on Joshua Snyder's graph)
Last updated 2014-Feb-28; Linux 2.6.36+

* "security" table left out for brevity
* "nat" table only consulted for "NEW" connections

2. AP S/W Architecture(3) – Wi-Fi



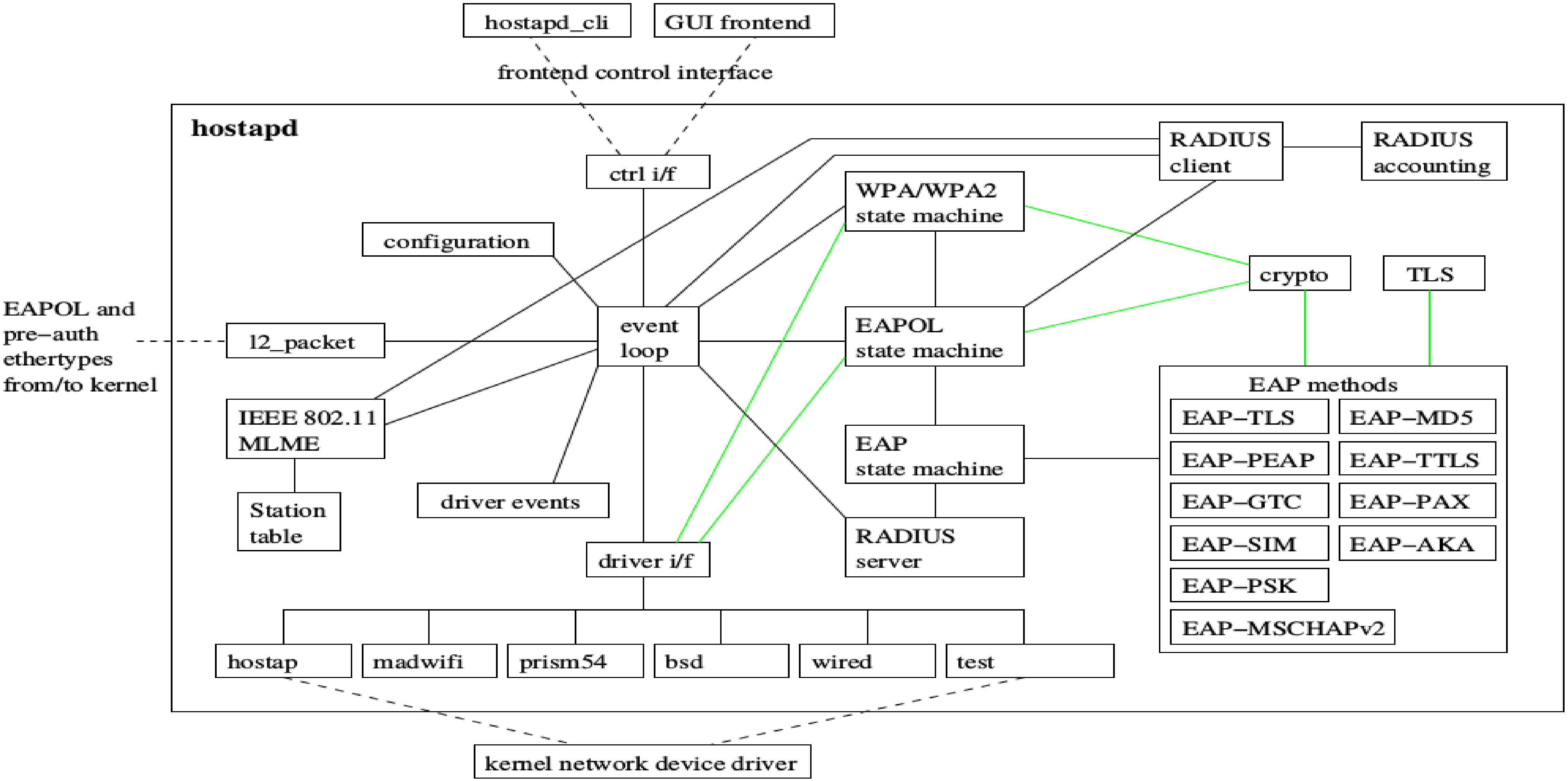
1) authentication

2) association

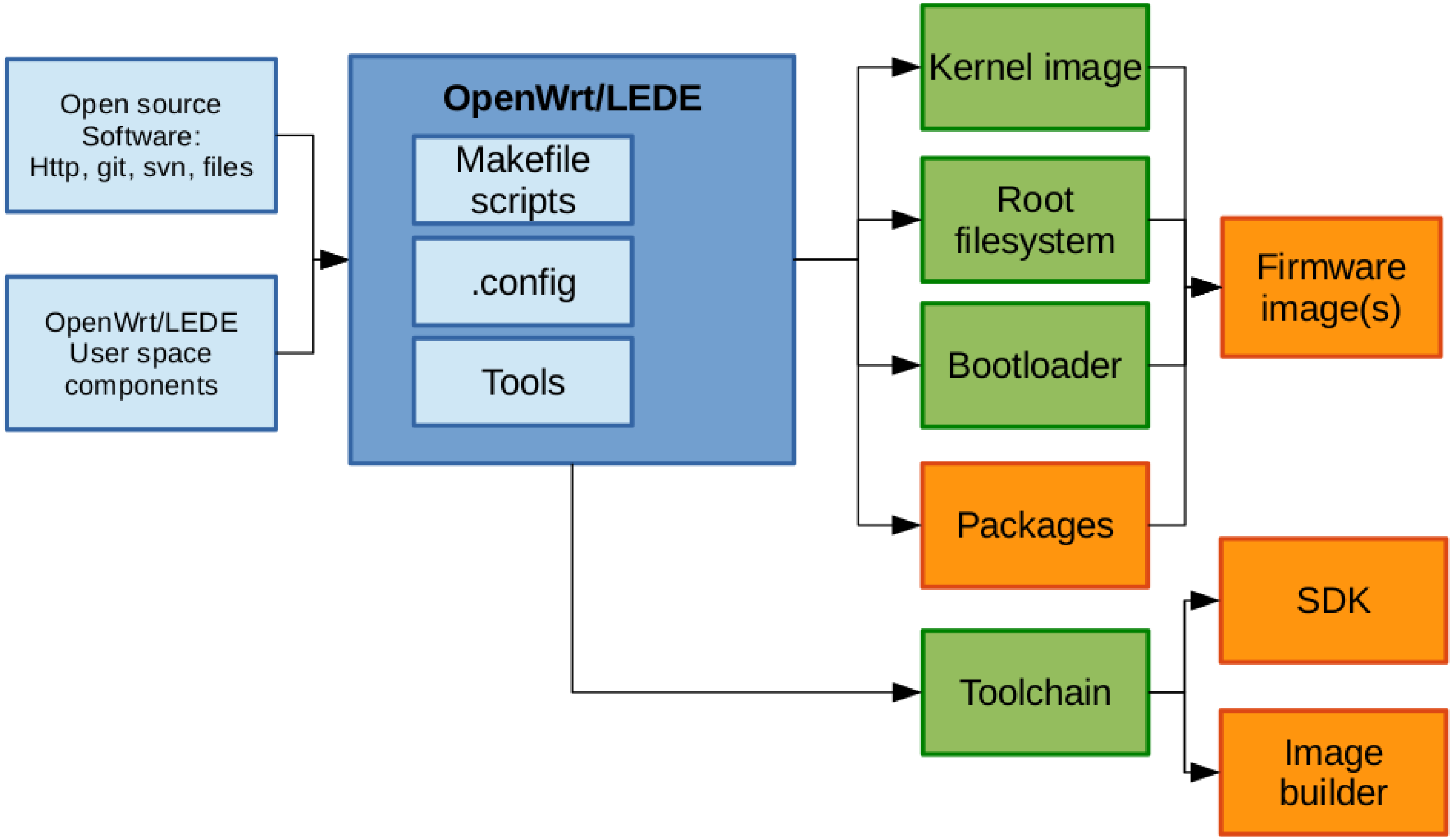
3) encryption

beacon
- MAC header
- timestamp
- beacon interval
- capability info
- SSID (network nam
- supported data rat
- radio parameters
- power slave flags

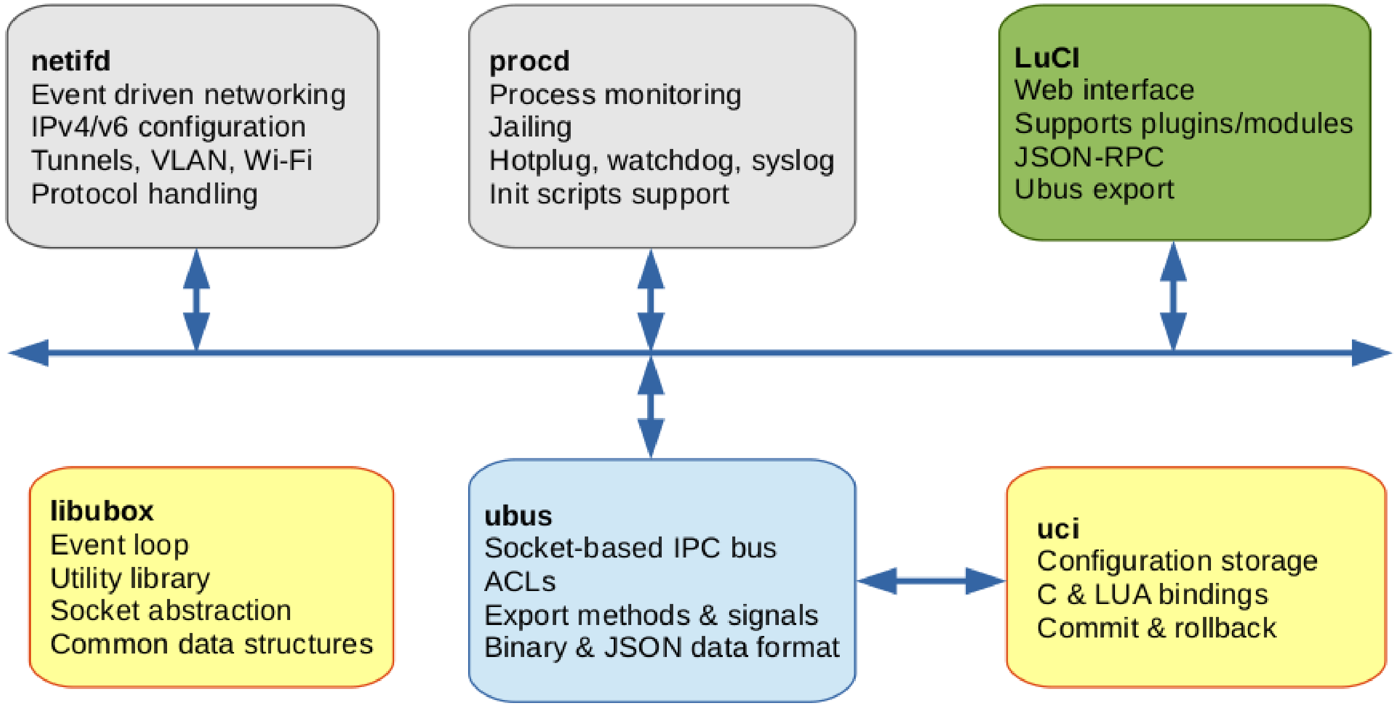
2. AP S/W Architecture(4) - hostapd



2. AP S/W Architecture(5) – OpenWrt/LEDE(1)



2. AP S/W Architecture(5) – OpenWrt/LEDE(2)



netifd
Event driven networking
IPv4/v6 configuration
Tunnels, VLAN, Wi-Fi
Protocol handling

procd
Process monitoring
Jailing
Hotplug, watchdog, syslog
Init scripts support

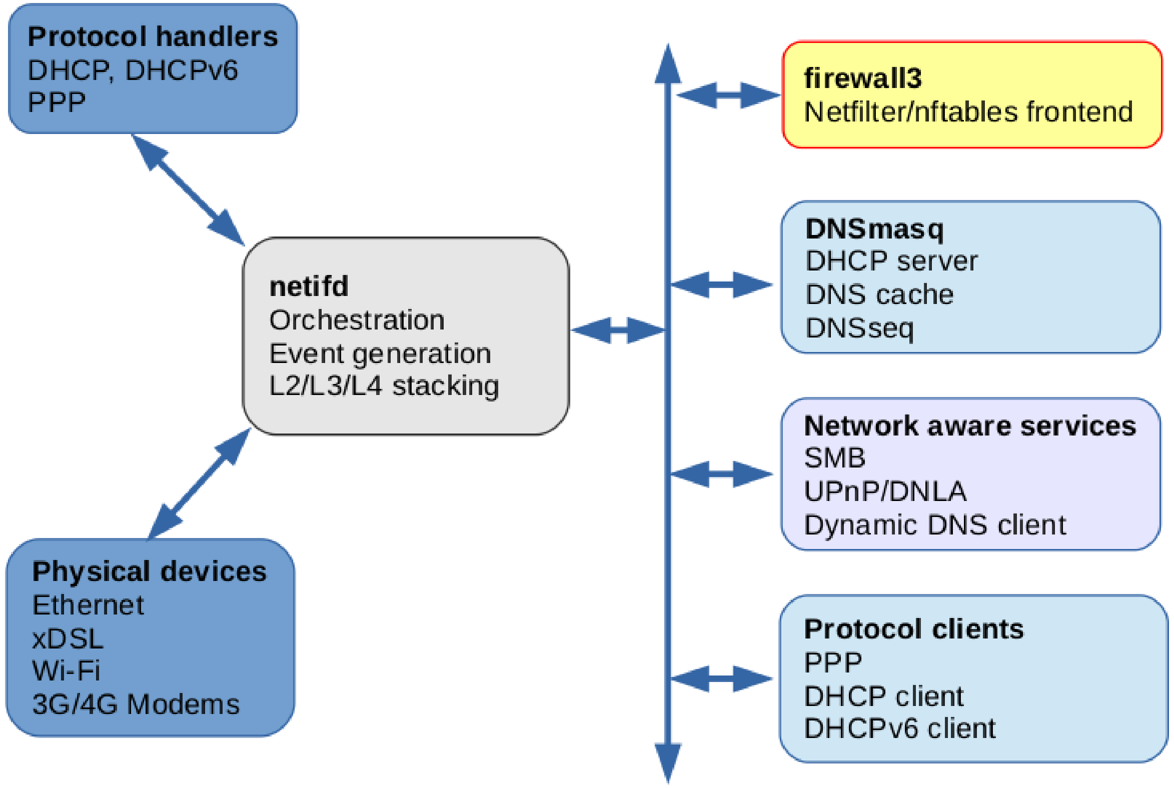
LuCI
Web interface
Supports plugins/modules
JSON-RPC
Ubus export

libubox
Event loop
Utility library
Socket abstraction
Common data structures

ubus
Socket-based IPC bus
ACLs
Export methods & signals
Binary & JSON data format

uci
Configuration storage
C & LUA bindings
Commit & rollback

2. AP S/W Architecture(5) – OpenWrt/LEDE(3)



3. OpenWrt/LEDE(1) – How to build(1)

- http://wiki.banana-pi.org/Getting_Started_with_R2#OpenWrt_On_SD_.26_Emmc_Steps
  - ✓ 전체 내용은 이 site를 참조하시기 바람.
  - ✓ 다음 단계로 진행하기 전에 아래 패키지를 미리 설치해 두어야 함.
  - ✓ gcc, binutils, bzip2, flex, python, perl, make, find, grep, diff, unzip, gawk, getopt, subversion, libz-dev and libc headers.

- $ **git clone https://github.com/garywangcn/bpi-r2_lede.git**
- $ **./scripts/feeds update –a**
- $ **./scripts/feeds install –a**
- $ **make menuconfig**
  - ✓ "Target System" to config as "MediaTek Ralink ARM“ 선택(별도로 화면 capture 안함)
  - ✓ 참고: menuconfig로 설정한 파일이 .config에 저장됨(이 파일을 수동으로 변경하는 것이 아님).
- $ **make -j1 V=s**
  - ✓ 전체 build를 수행함. 한참 걸릴 것임(최초 build 시 1시간 정도 걸린 듯).
  - ✓ V=99를 주면, compile 과정이 자세히 출력될 것임.

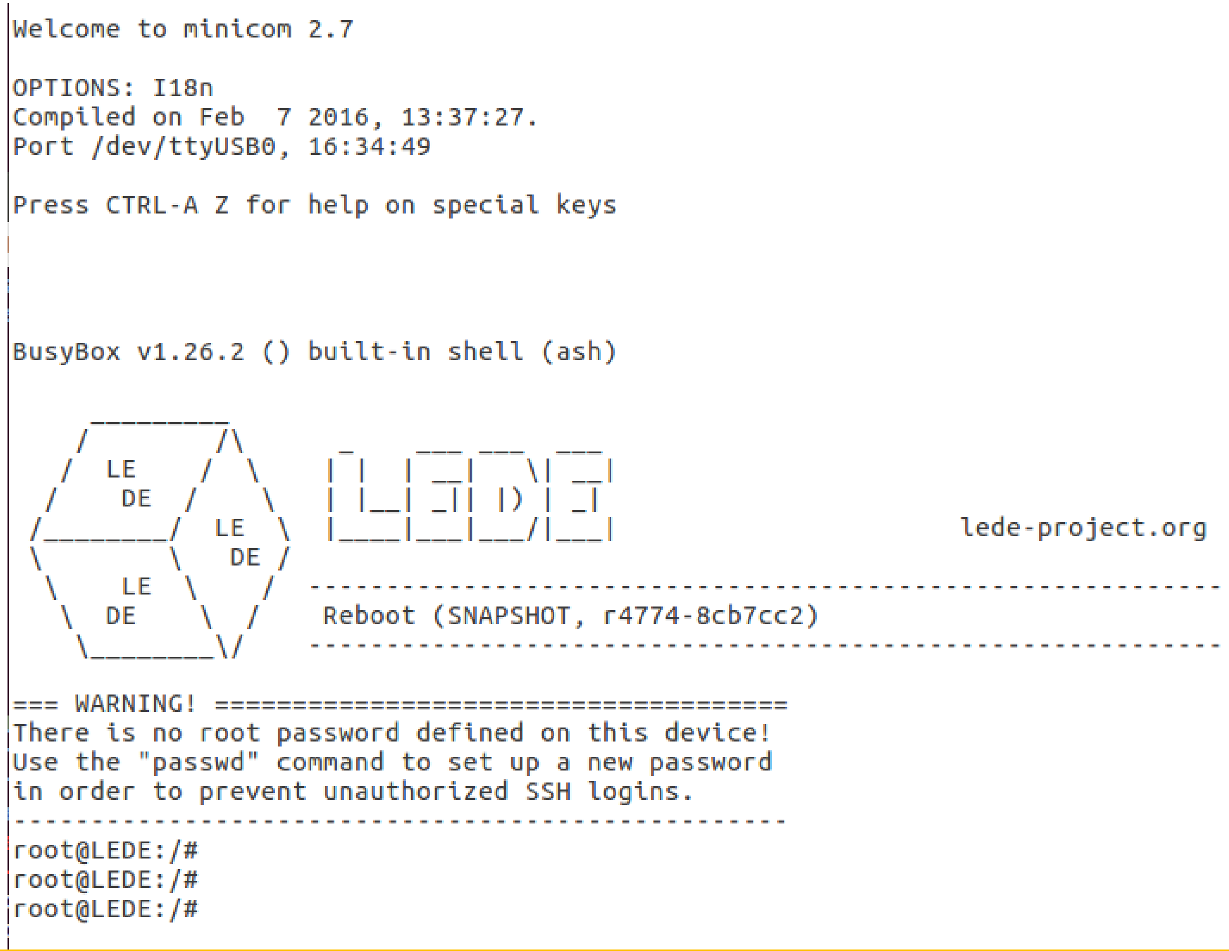
3. OpenWrt/LEDE(1) – How to build(2)

- $ **cd build_dir/target-arm_cortex-a7+neon-vfpv4_musl_eabi/linux-mediatek_32/**
- $ ls –l
  - mtk-bpi-r2-EMMC.img ← eMMC(on board)용 image 파일
  - mtk-bpi-r2-SD.img ← microSD용 image 파일
- $ **sudo dd if=./mtk-bpi-r2-SD.img of=/dev/sdc**
  - microSD를 PC에 insert 후, 위의 명령 수행
  - /dev/sdc는 실제 자신의 PC에서 인식하는 값으로 교체해야 함[주의].

- eMMC에 writing하는 방식은 reference site를 참조하시기 바람.

- **<부팅 테스트>**
  - microSD를 target board(BPI-R2)에 꽂고, 부팅 시작
  - Power button을 10초간 꾹 눌러 주어야 함(녹색 LED가 출력된 후, 2~3초 후에 power button을 떼면 됨)
  - Minicom : 115200, 8N1으로 설정하여 Console 메시지 확인해 보시기 바람.

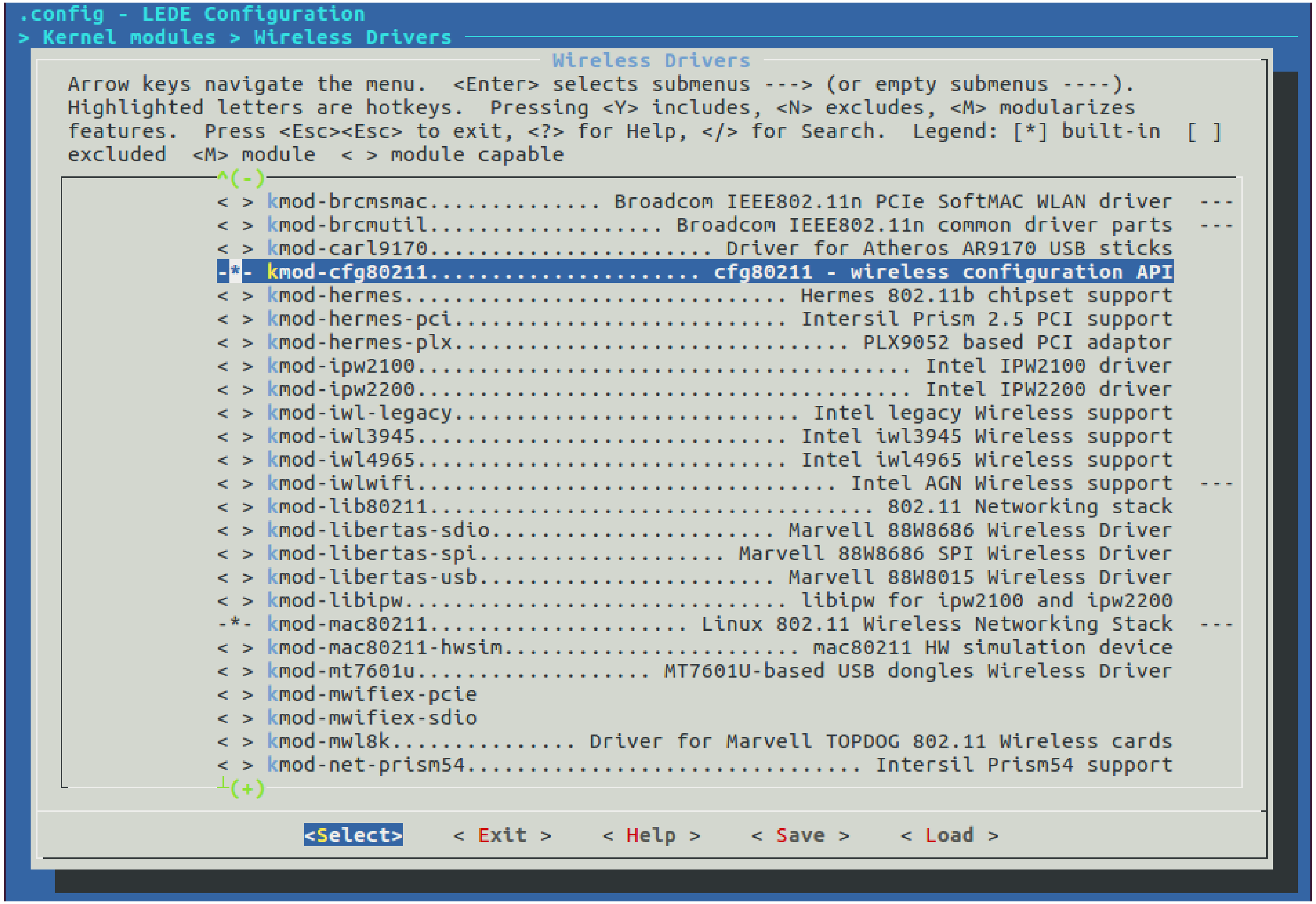
3. OpenWrt/LEDE(1) – How to build(3)



```
Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Feb  7 2016, 13:37:27.
Port /dev/ttyUSB0, 16:34:49

Press CTRL-A Z for help on special keys




BusyBox v1.26.2 () built-in shell (ash)

     _________
    /        /\      _    ___ ___  ___
   /  LE    /  \    | |  | __|   \| __|
  /    DE  /    \   | |__| _|| |) | _|
 /________/  LE  \  |____|___|___/|___|                      lede-project.org
 \        \   DE /
  \    LE  \    /  -----------------------------------------------------------
   \  DE    \  /    Reboot (SNAPSHOT, r4774-8cb7cc2)
    \________\/    -----------------------------------------------------------

=== WARNING! =====================================
There is no root password defined on this device!
Use the "passwd" command to set up a new password
in order to prevent unauthorized SSH logins.
--------------------------------------------------
root@LEDE:/#
root@LEDE:/#
root@LEDE:/#
```

이 상태에서 LAN, WAN에 ethernet cable을 연결하면 internet 될 것임.

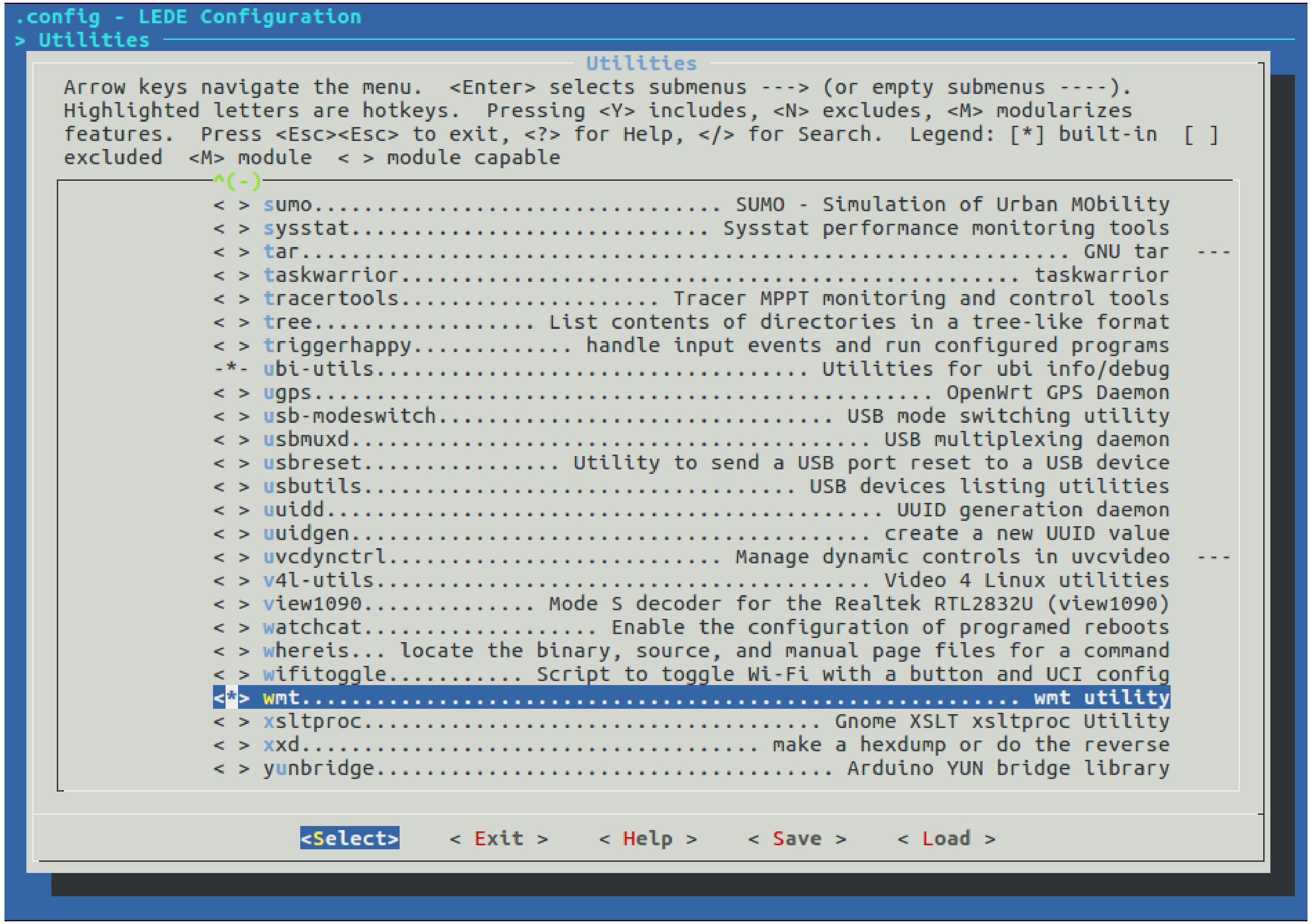
3. OpenWrt/LEDE(2) – 802.11n 무선랜 살리기(1)



```
.config - LEDE Configuration
> Kernel modules > Wireless Drivers ─────────────────────────────────────────────
                            Wireless Drivers
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).
  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

            ^(-)
            < > kmod-brcmsmac............. Broadcom IEEE802.11n PCIe SoftMAC WLAN driver  ---
            < > kmod-brcmutil.................. Broadcom IEEE802.11n common driver parts  ---
            < > kmod-carl9170...................... Driver for Atheros AR9170 USB sticks
            -*- kmod-cfg80211..................... cfg80211 - wireless configuration API
            < > kmod-hermes.............................. Hermes 802.11b chipset support
            < > kmod-hermes-pci.......................... Intersil Prism 2.5 PCI support
            < > kmod-hermes-plx............................... PLX9052 based PCI adaptor
            < > kmod-ipw2100....................................... Intel IPW2100 driver
            < > kmod-ipw2200....................................... Intel IPW2200 driver
            < > kmod-iwl-legacy............................ Intel legacy Wireless support
            < > kmod-iwl3945.............................. Intel iwl3945 Wireless support
            < > kmod-iwl4965.............................. Intel iwl4965 Wireless support
            < > kmod-iwlwifi.................................. Intel AGN Wireless support  ---
            < > kmod-lib80211........................................ 802.11 Networking stack
            < > kmod-libertas-sdio...................... Marvell 88W8686 Wireless Driver
            < > kmod-libertas-spi.................... Marvell 88W8686 SPI Wireless Driver
            < > kmod-libertas-usb....................... Marvell 88W8015 Wireless Driver
            < > kmod-libipw............................. libipw for ipw2100 and ipw2200
            -*- kmod-mac80211...................... Linux 802.11 Wireless Networking Stack  ---
            < > kmod-mac80211-hwsim......................... mac80211 HW simulation device
            < > kmod-mt7601u................... MT7601U-based USB dongles Wireless Driver
            < > kmod-mwifiex-pcie
            < > kmod-mwifiex-sdio
            < > kmod-mwl8k............... Driver for Marvell TOPDOG 802.11 Wireless cards
            < > kmod-net-prism54................................ Intersil Prism54 support
            ┴(+)

                <Select>    < Exit >    < Help >    < Save >    < Load >
```

Menuconfig(LEDE Configuration -> Kernel modules -> Wireless Drivers)에서 kmod-cfg80211, kmod-mac80211을 enable해 주어야 함.

3. OpenWrt/LEDE(2) – 802.11n 무선랜 살리기(2)



```
.config - LEDE Configuration
> Utilities ─────────────────────────────────────────────────────────────────
                              Utilities
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).
  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
              ^(-)
              < > sumo................................ SUMO - Simulation of Urban MObility
              < > sysstat............................ Sysstat performance monitoring tools
              < > tar................................................... GNU tar  ---
              < > taskwarrior........................................ taskwarrior
              < > tracertools.................... Tracer MPPT monitoring and control tools
              < > tree................. List contents of directories in a tree-like format
              < > triggerhappy............ handle input events and run configured programs
              -*- ubi-utils................................. Utilities for ubi info/debug
              < > ugps................................................ OpenWrt GPS Daemon
              < > usb-modeswitch................................ USB mode switching utility
              < > usbmuxd......................................... USB multiplexing daemon
              < > usbreset............... Utility to send a USB port reset to a USB device
              < > usbutils................................ USB devices listing utilities
              < > uuidd............................................ UUID generation daemon
              < > uuidgen........................................ create a new UUID value
              < > uvcdynctrl......................... Manage dynamic controls in uvcvideo  ---
              < > v4l-utils........................................ Video 4 Linux utilities
              < > view1090............. Mode S decoder for the Realtek RTL2832U (view1090)
              < > watchcat.................. Enable the configuration of programed reboots
              < > whereis... locate the binary, source, and manual page files for a command
              < > wifitoggle.......... Script to toggle Wi-Fi with a button and UCI config
              <*> wmt................................................... wmt utility
              < > xsltproc.................................... Gnome XSLT xsltproc Utility
              < > xxd.......................................... make a hexdump or do the reverse
              < > yunbridge..................................... Arduino YUN bridge library

              <Select>    < Exit >    < Help >    < Save >    < Load >
```

wmt_loader는 menuconfig에서 enable해 주어야 함.

3. OpenWrt/LEDE(2) – 802.11n 무선랜 살리기(3)

- $ **make clean**
  - Kernel module config 조정에 따라, 그냥 build 시 에러 발생하기 때문.
- $ **make -j1 V=99**

- 이후 다시, 아래 명령 수행 후, 부팅 시도
- $ **sudo dd if=./mtk-bpi-r2-SD.img of=/dev/sdc**

- 참고 사항
  - 부팅 전에 안테나를 연결해 주자.

3. OpenWrt/LEDE(2) – 802.11n 무선랜 살리기(4)

- https://www.cnblogs.com/topbin/p/9519881.html
- Target board에서 아래 내용 입력 후, 재 부팅하도록 하자.

<setup.sh 파일>

```
#!/bin/ash

wmt_loader &
sleep 3
stp_uart_launcher -p /etc/firmware &
sleep 5
echo A > /dev/wmtWifi
sleep 5
hostapd -d /etc/hostapd/hostapd.conf
```

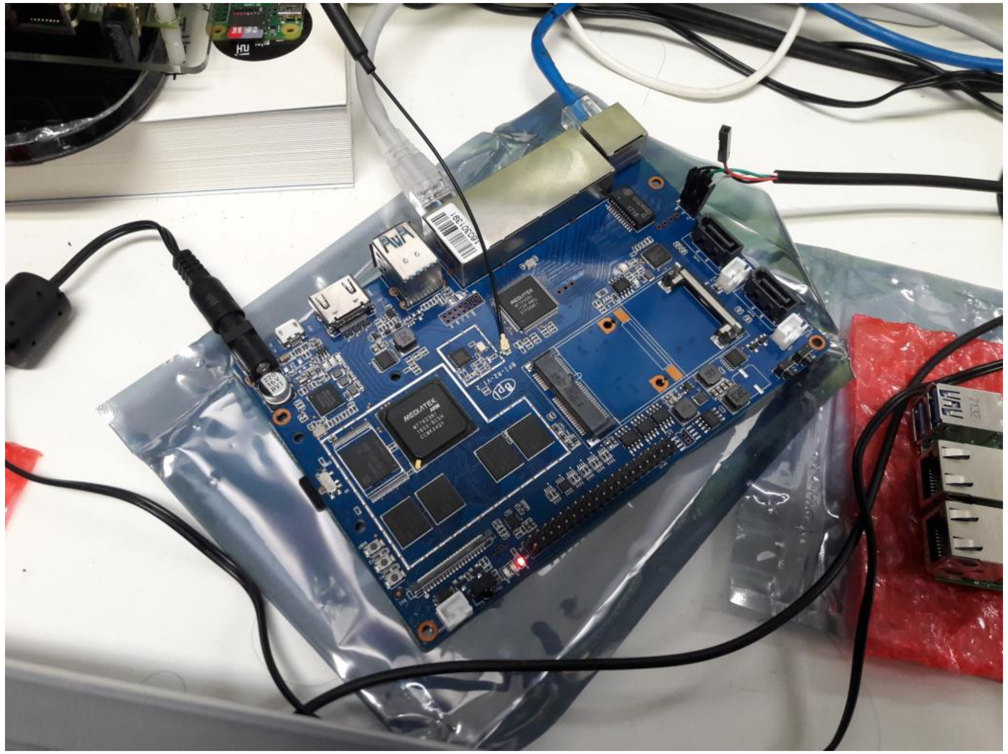
< /etc/hostapd/hostapd.conf 파일>

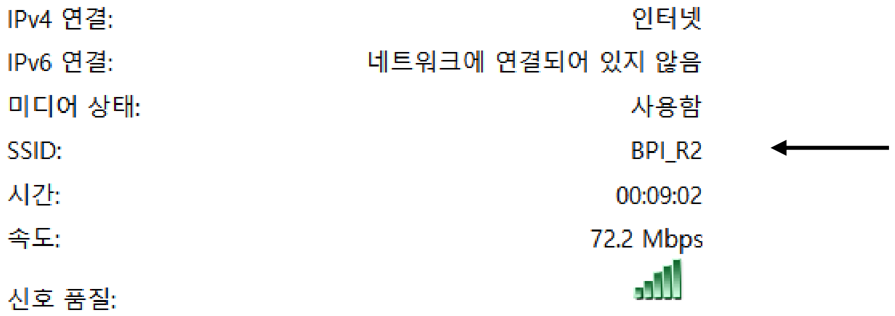
```
interface=ap0
bridge=br-lan
ssid=BPI_R2
driver=nl80211
country_code=CN
hw_mode=g
channel=1
max_num_sta=5
wpa=2
auth_algs=1
rsn_pairwise=CCMP
wpa_key_mgmt=WPA-PSK
wpa_passphrase=ledetest
logger_stdout=-1
logger_stdout_level=2
```

</etc/rc.local 파일 편집>

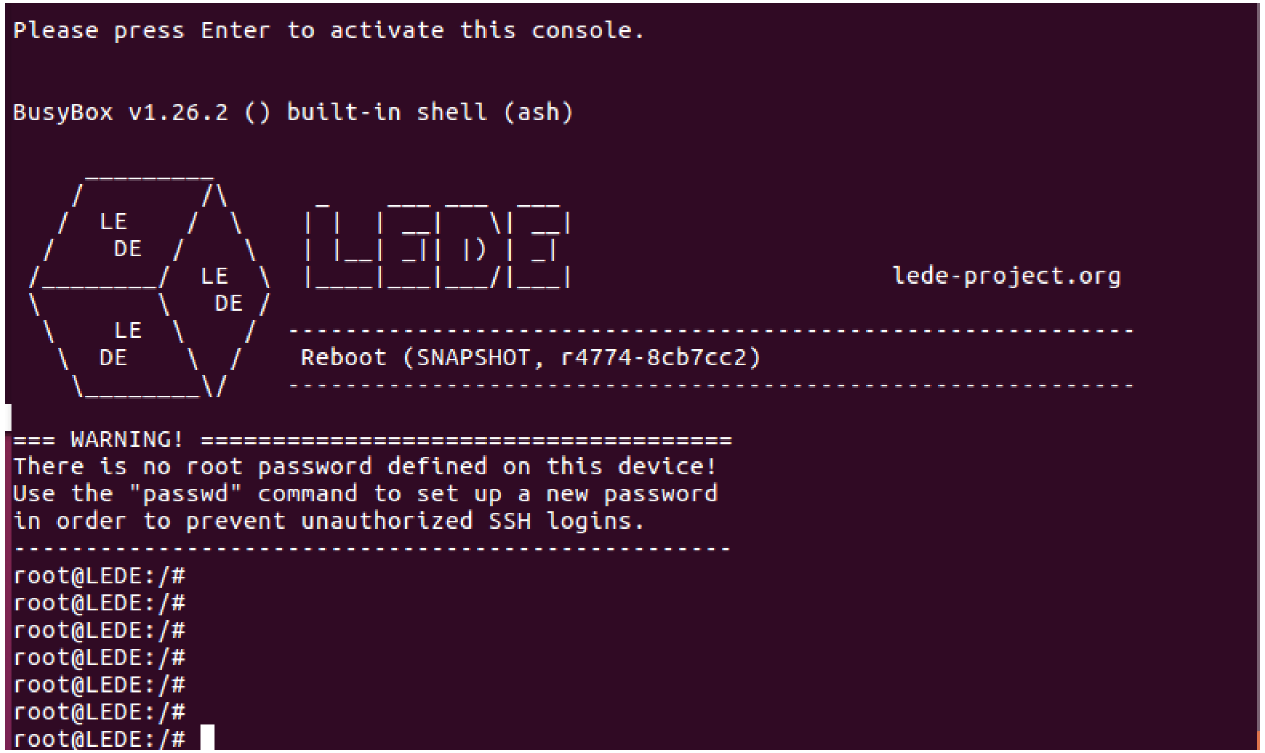
```

cd /root
sleep 6
./setup.sh &

exit 0
```

3. OpenWrt/LEDE(2) – 802.11n 무선랜 살리기(5)





IPv4 연결:	인터넷
IPv6 연결:	네트워크에 연결되어 있지 않음
미디어 상태:	사용함
SSID:	BPI_R2
시간:	00:09:02
속도:	72.2 Mbps
신호 품질:	📶

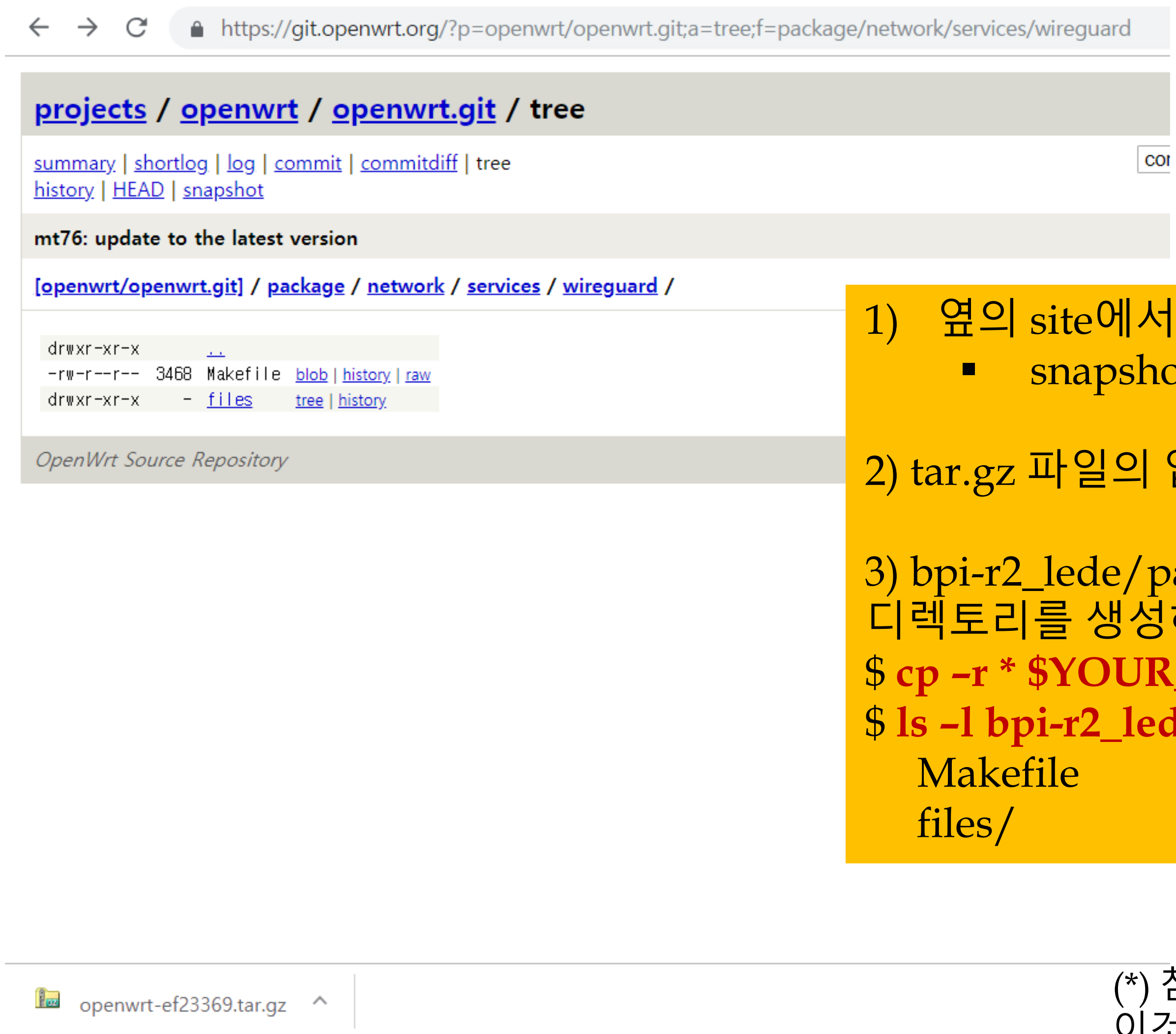


```
Please press Enter to activate this console.


BusyBox v1.26.2 () built-in shell (ash)

     _________
    /        /\      _    ___ ___  ___
   /  LE    /  \    | |  | __|   \| __|
  /    DE  /    \   | |__| _|| |) | _|
 /________/  LE  \  |____|___|___/|___|                      lede-project.org
 \        \   DE /
  \    LE  \    /  -----------------------------------------------------------
   \  DE    \  /    Reboot (SNAPSHOT, r4774-8cb7cc2)
    \________\/    -----------------------------------------------------------

=== WARNING! =====================================
There is no root password defined on this device!
Use the "passwd" command to set up a new password
in order to prevent unauthorized SSH logins.
--------------------------------------------------
root@LEDE:/#
root@LEDE:/#
root@LEDE:/#
root@LEDE:/#
root@LEDE:/#
root@LEDE:/#
root@LEDE:/#
```

3. OpenWrt/LEDE(2) – 802.11n 무선랜 살리기(6)

- <TODO>
  - 부팅 시 최대한 빨리 무선랜이 올라올 수 있도록 Tuning이 필요함.

4. WireGuard 올리기(1)



https://git.openwrt.org/?p=openwrt/openwrt.git;a=tree;f=package/network/services/wireguard

projects / openwrt / openwrt.git / tree

summary | shortlog | log | commit | commitdiff | tree
history | HEAD | snapshot

mt76: update to the latest version

[openwrt/openwrt.git] / package / network / services / wireguard /

```
drwxr-xr-x        ..
-rw-r--r--  3468  Makefile  blob | history | raw
drwxr-xr-x     -  files     tree | history
```

OpenWrt Source Repository

openwrt-ef23369.tar.gz

1) 옆의 site에서 openwrt용 wireguard package 파일을 download 받는다.
   - snapshot을 받으면 됨.

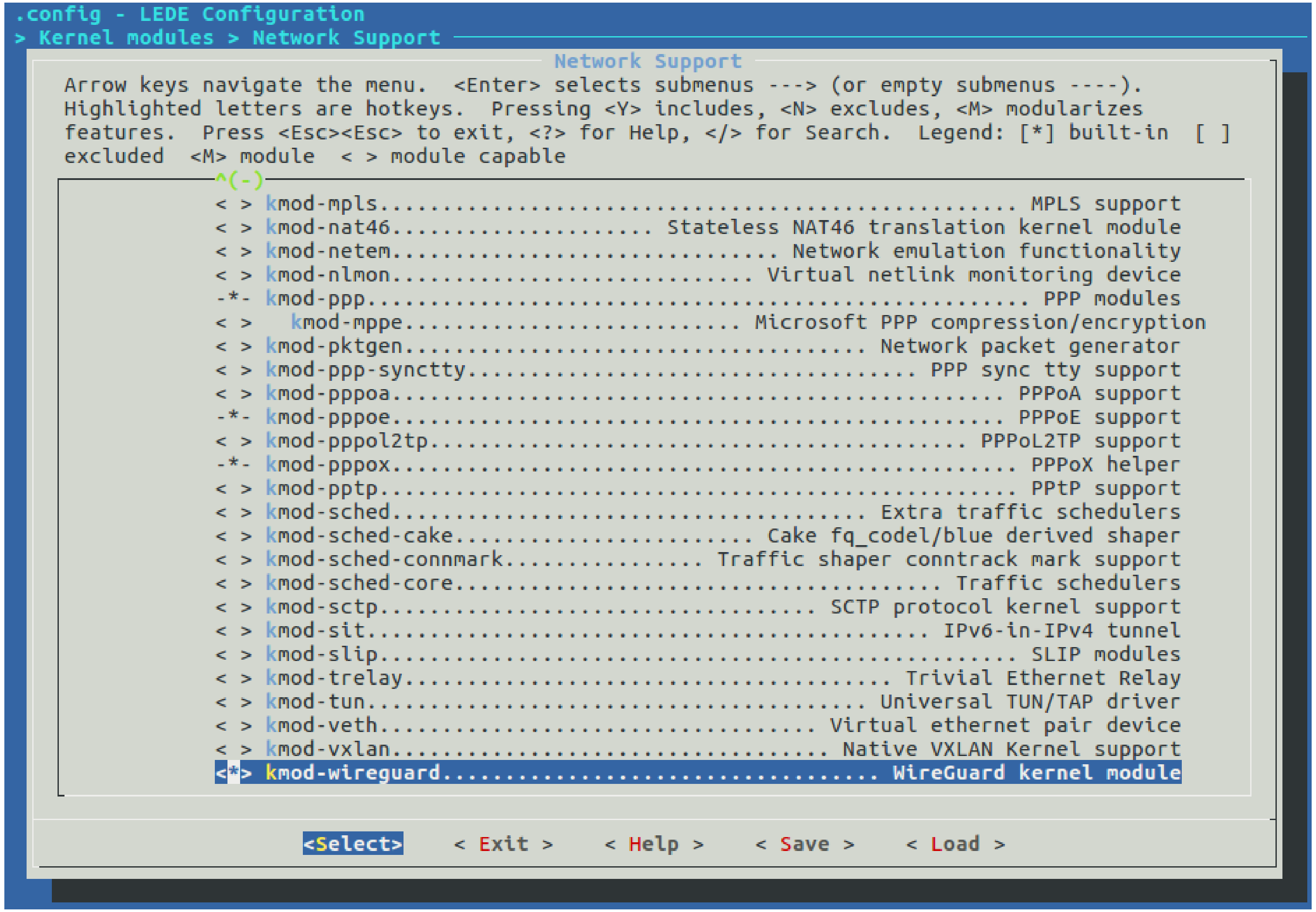
2) tar.gz 파일의 압축을 푼다.

3) bpi-r2_lede/package/network/services 아래에 wireguard 디렉토리를 생성하고, 여기에 압축을 푼 내용을 복사한다.

```
$ cp –r * $YOUR_PATH/bpi-r2_lede/package/network/services/wireguard
$ ls –l bpi-r2_lede/package/network/services/wireguard
    Makefile
    files/
```

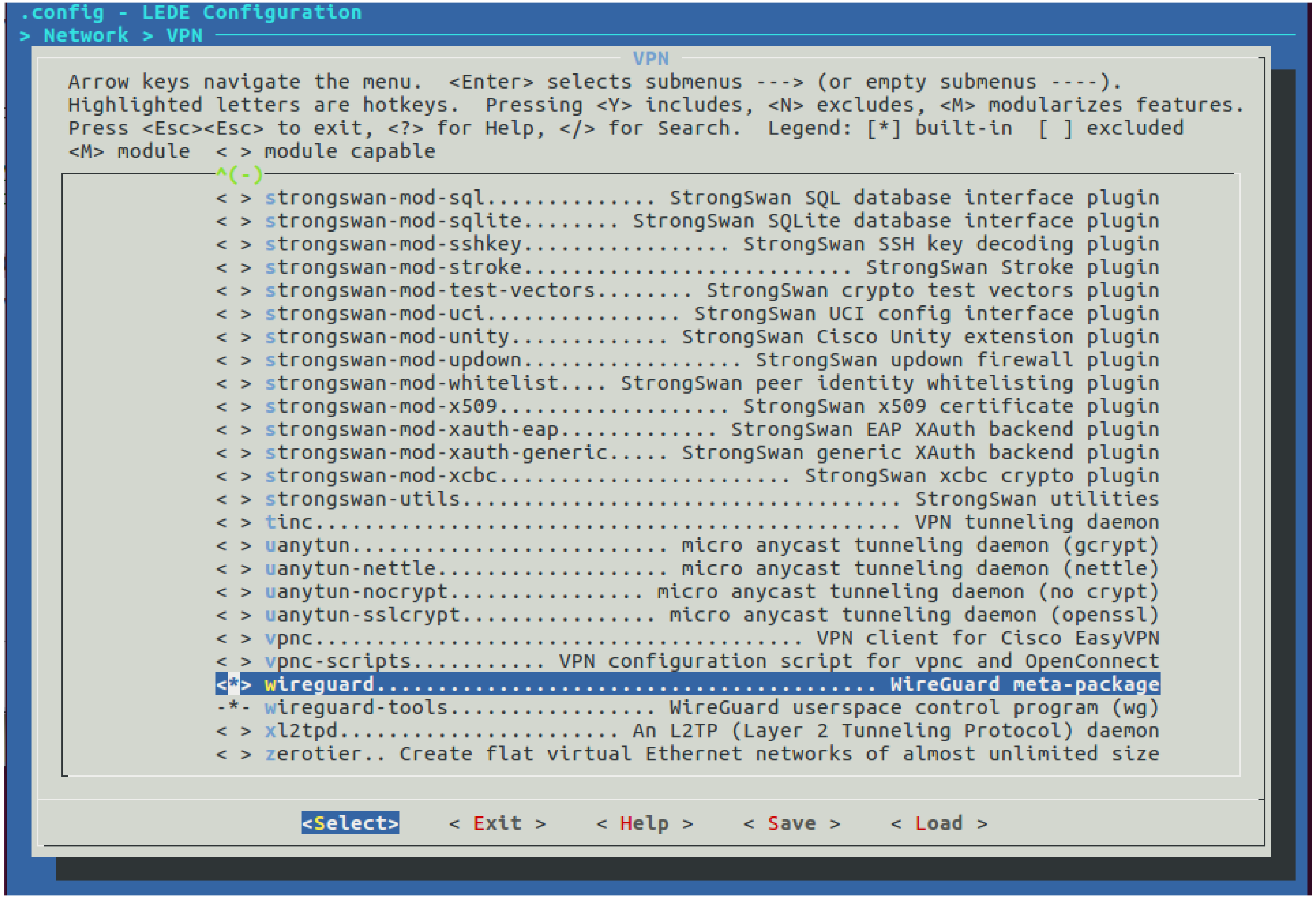
(*) 참고 사항: OpenWrt/LEDE에 hello_world를 하나 추가해 보라 !
이것을 해 보아야 OpenWrt 세상에 들어설 수 있다 ?!

4. WireGuard 올리기(2)



```
.config - LEDE Configuration
> Kernel modules > Network Support
                              Network Support
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).
  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

            ^(-)
            < > kmod-mpls.................................................. MPLS support
            < > kmod-nat46..................... Stateless NAT46 translation kernel module
            < > kmod-netem.............................. Network emulation functionality
            < > kmod-nlmon............................ Virtual netlink monitoring device
            -*- kmod-ppp................................................... PPP modules
            < >   kmod-mppe......................... Microsoft PPP compression/encryption
            < > kmod-pktgen.................................... Network packet generator
            < > kmod-ppp-synctty.................................. PPP sync tty support
            < > kmod-pppoa................................................ PPPoA support
            -*- kmod-pppoe................................................ PPPoE support
            < > kmod-pppol2tp........................................ PPPoL2TP support
            -*- kmod-pppox................................................. PPPoX helper
            < > kmod-pptp.................................................. PPtP support
            < > kmod-sched..................................... Extra traffic schedulers
            < > kmod-sched-cake........................ Cake fq_codel/blue derived shaper
            < > kmod-sched-connmark................. Traffic shaper conntrack mark support
            < > kmod-sched-core.......................................... Traffic schedulers
            < > kmod-sctp................................... SCTP protocol kernel support
            < > kmod-sit............................................. IPv6-in-IPv4 tunnel
            < > kmod-slip.................................................. SLIP modules
            < > kmod-trelay......................................... Trivial Ethernet Relay
            < > kmod-tun........................................ Universal TUN/TAP driver
            < > kmod-veth...................................... Virtual ethernet pair device
            < > kmod-vxlan...................................... Native VXLAN Kernel support
            <*> kmod-wireguard.................................. WireGuard kernel module

                  <Select>    < Exit >    < Help >    < Save >    < Load >
```

4. WireGuard 올리기(3)



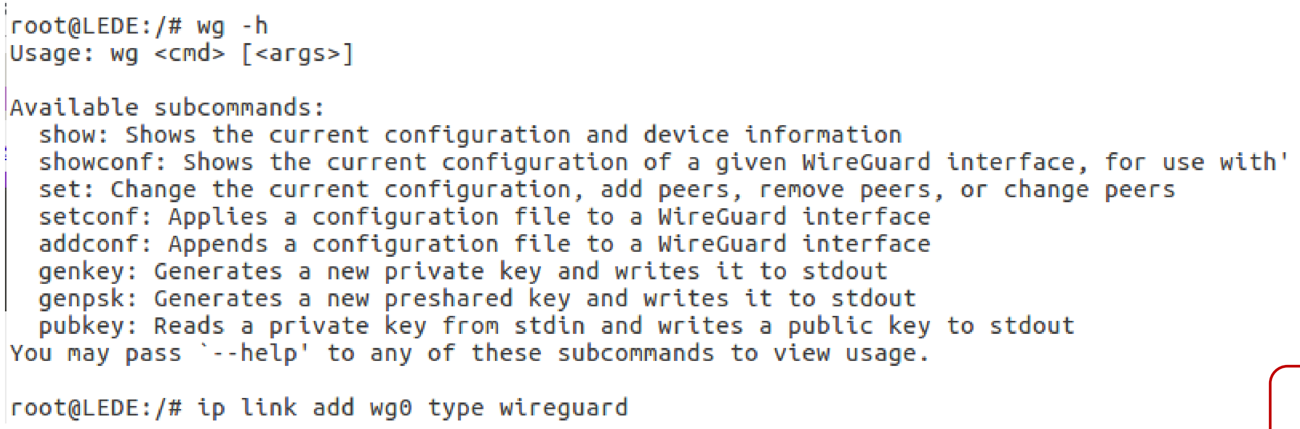
```
.config - LEDE Configuration
> Network > VPN ─────────────────────────────────────────────────────────────
                                   VPN
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).
  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.
  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded
  <M> module  < > module capable
              ^(-)
              < > strongswan-mod-sql............. StrongSwan SQL database interface plugin
              < > strongswan-mod-sqlite........ StrongSwan SQLite database interface plugin
              < > strongswan-mod-sshkey................. StrongSwan SSH key decoding plugin
              < > strongswan-mod-stroke........................... StrongSwan Stroke plugin
              < > strongswan-mod-test-vectors........ StrongSwan crypto test vectors plugin
              < > strongswan-mod-uci............... StrongSwan UCI config interface plugin
              < > strongswan-mod-unity............ StrongSwan Cisco Unity extension plugin
              < > strongswan-mod-updown.................. StrongSwan updown firewall plugin
              < > strongswan-mod-whitelist.... StrongSwan peer identity whitelisting plugin
              < > strongswan-mod-x509................. StrongSwan x509 certificate plugin
              < > strongswan-mod-xauth-eap............. StrongSwan EAP XAuth backend plugin
              < > strongswan-mod-xauth-generic..... StrongSwan generic XAuth backend plugin
              < > strongswan-mod-xcbc....................... StrongSwan xcbc crypto plugin
              < > strongswan-utils..................................... StrongSwan utilities
              < > tinc............................................... VPN tunneling daemon
              < > uanytun........................ micro anycast tunneling daemon (gcrypt)
              < > uanytun-nettle................. micro anycast tunneling daemon (nettle)
              < > uanytun-nocrypt.............. micro anycast tunneling daemon (no crypt)
              < > uanytun-sslcrypt............... micro anycast tunneling daemon (openssl)
              < > vpnc........................................ VPN client for Cisco EasyVPN
              < > vpnc-scripts.......... VPN configuration script for vpnc and OpenConnect
              <*> wireguard.......................................... WireGuard meta-package
              -*- wireguard-tools................ WireGuard userspace control program (wg)
              < > xl2tpd...................... An L2TP (Layer 2 Tunneling Protocol) daemon
              < > zerotier.. Create flat virtual Ethernet networks of almost unlimited size

              <Select>    < Exit >    < Help >    < Save >    < Load >
```

4. WireGuard 올리기(4)

- $ **make -j1 V=99**

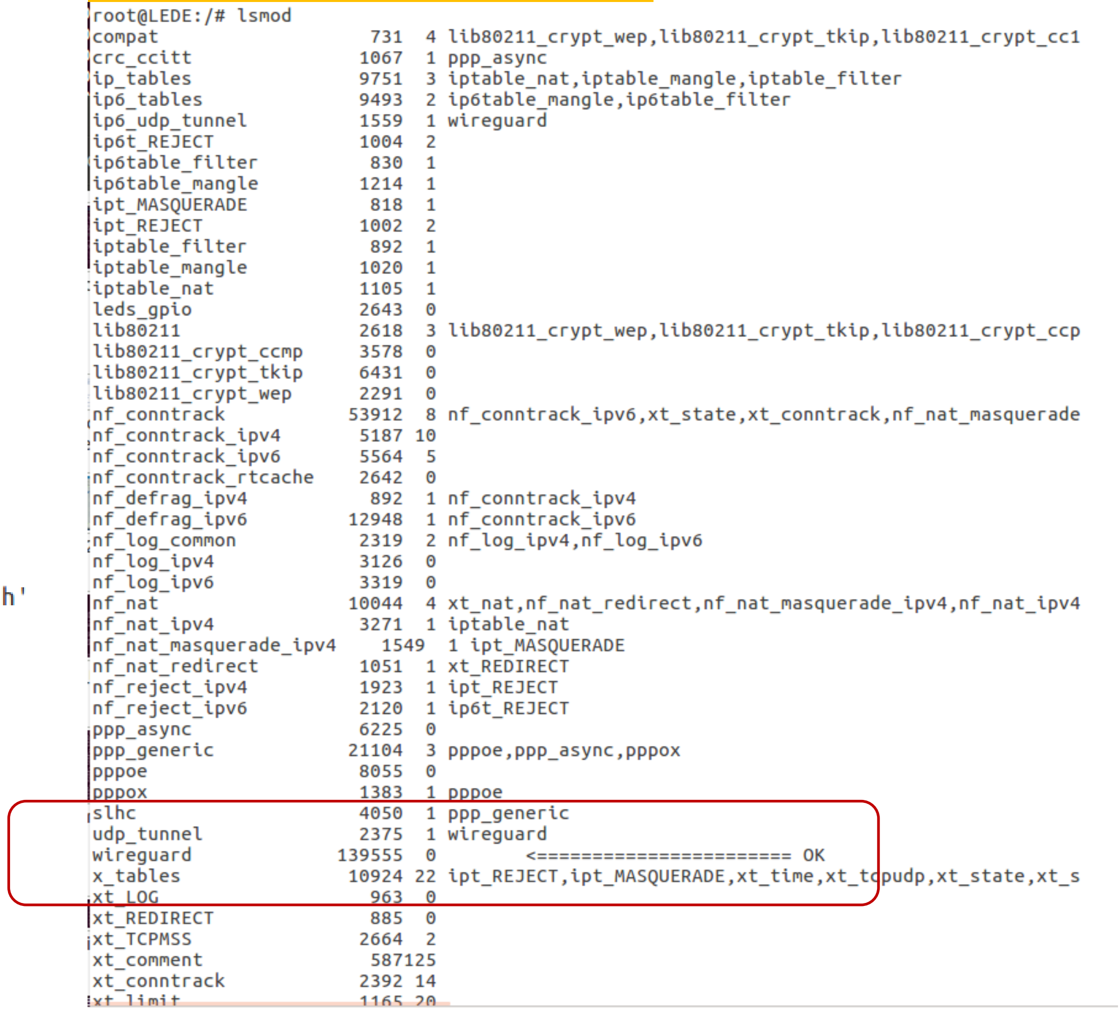
- 이후 다시, 아래 명령 수행 후, 부팅 시도
- $ **sudo dd if=./mtk-bpi-r2-SD.img of=/dev/sdc**

- **<부팅 후>**
- root@LEDE:1# **wg**



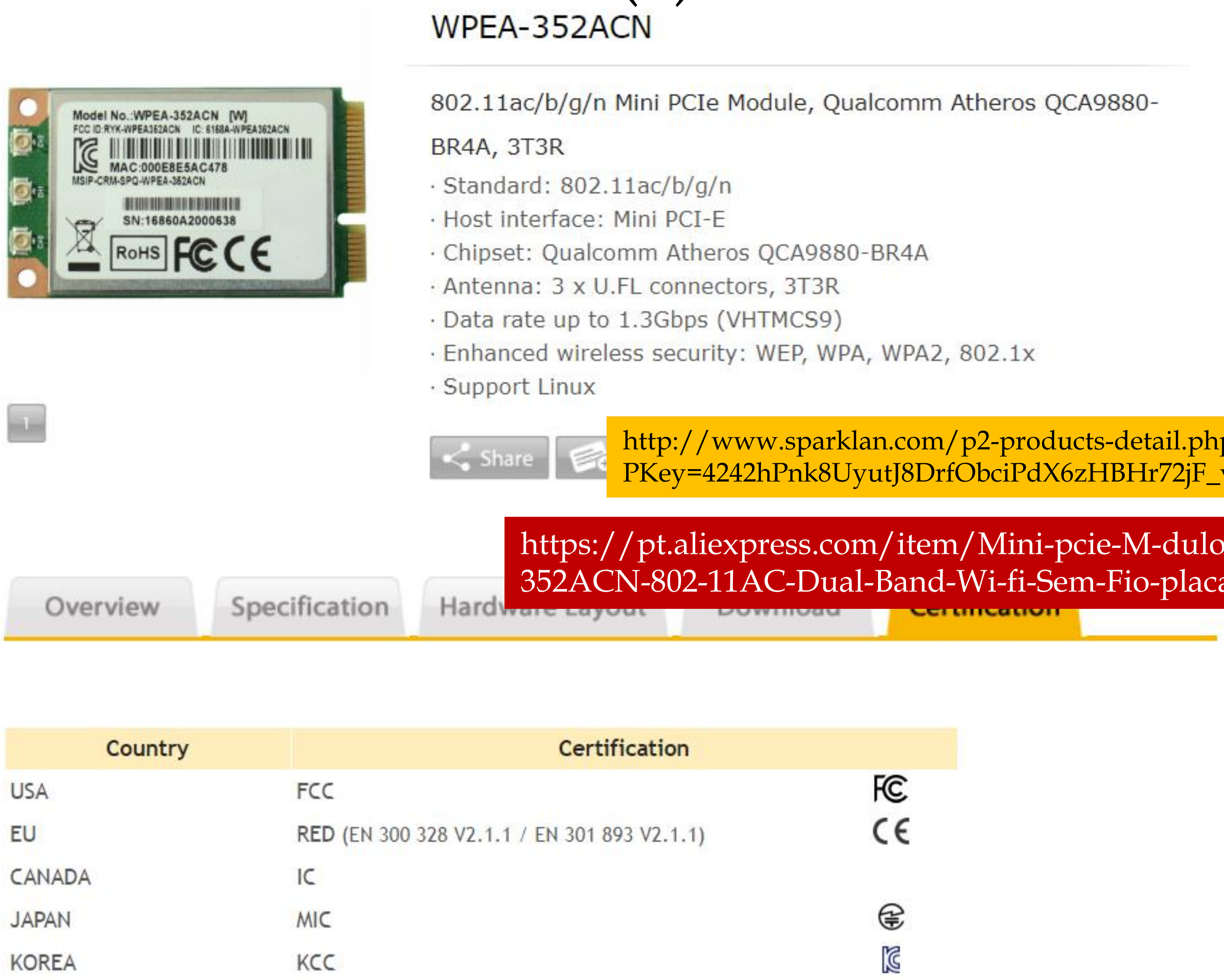
```
root@LEDE:/# wg -h
Usage: wg <cmd> [<args>]

Available subcommands:
  show: Shows the current configuration and device information
  showconf: Shows the current configuration of a given WireGuard interface, for use with'
  set: Change the current configuration, add peers, remove peers, or change peers
  setconf: Applies a configuration file to a WireGuard interface
  addconf: Appends a configuration file to a WireGuard interface
  genkey: Generates a new private key and writes it to stdout
  genpsk: Generates a new preshared key and writes it to stdout
  pubkey: Reads a private key from stdin and writes a public key to stdout
You may pass `--help' to any of these subcommands to view usage.

root@LEDE:/# ip link add wg0 type wireguard
```

<부팅 후, lsmod 명령 실행>



```
root@LEDE:/# lsmod
compat                   731  4 lib80211_crypt_wep,lib80211_crypt_tkip,lib80211_crypt_cc1
crc_ccitt               1067  1 ppp_async
ip_tables               9751  3 iptable_nat,iptable_mangle,iptable_filter
ip6_tables              9493  2 ip6table_mangle,ip6table_filter
ip6_udp_tunnel          1559  1 wireguard
ip6t_REJECT             1004  2
ip6table_filter          830  1
ip6table_mangle         1214  1
ipt_MASQUERADE           818  1
ipt_REJECT              1002  2
iptable_filter           892  1
iptable_mangle          1020  1
iptable_nat             1105  1
leds_gpio               2643  0
lib80211                2618  3 lib80211_crypt_wep,lib80211_crypt_tkip,lib80211_crypt_ccp
lib80211_crypt_ccmp     3578  0
lib80211_crypt_tkip     6431  0
lib80211_crypt_wep      2291  0
nf_conntrack           53912  8 nf_conntrack_ipv6,xt_state,xt_conntrack,nf_nat_masquerade
nf_conntrack_ipv4       5187 10
nf_conntrack_ipv6       5564  5
nf_conntrack_rtcache    2642  0
nf_defrag_ipv4           892  1 nf_conntrack_ipv4
nf_defrag_ipv6         12948  1 nf_conntrack_ipv6
nf_log_common           2319  2 nf_log_ipv4,nf_log_ipv6
nf_log_ipv4             3126  0
nf_log_ipv6             3319  0
nf_nat                 10044  4 xt_nat,nf_nat_redirect,nf_nat_masquerade_ipv4,nf_nat_ipv4
nf_nat_ipv4             3271  1 iptable_nat
nf_nat_masquerade_ipv4    1549  1 ipt_MASQUERADE
nf_nat_redirect         1051  1 xt_REDIRECT
nf_reject_ipv4          1923  1 ipt_REJECT
nf_reject_ipv6          2120  1 ip6t_REJECT
ppp_async               6225  0
ppp_generic            21104  3 pppoe,ppp_async,pppox
pppoe                   8055  0
pppox                   1383  1 pppoe
slhc                    4050  1 ppp_generic
udp_tunnel              2375  1 wireguard
wireguard             139555  0        <======================== OK
x_tables               10924 22 ipt_REJECT,ipt_MASQUERADE,xt_time,xt_tcpudp,xt_state,xt_s
xt_LOG                   963  0
xt_REDIRECT              885  0
xt_TCPMSS               2664  2
xt_comment               587125
xt_conntrack            2392 14
xt_limit                1165 20
```

5. 802.11ac 무선랜 구동하기(1)



WPEA-352ACN

802.11ac/b/g/n Mini PCIe Module, Qualcomm Atheros QCA9880-BR4A, 3T3R

- Standard: 802.11ac/b/g/n
- Host interface: Mini PCI-E
- Chipset: Qualcomm Atheros QCA9880-BR4A
- Antenna: 3 x U.FL connectors, 3T3R
- Data rate up to 1.3Gbps (VHTMCS9)
- Enhanced wireless security: WEP, WPA, WPA2, 802.1x
- Support Linux

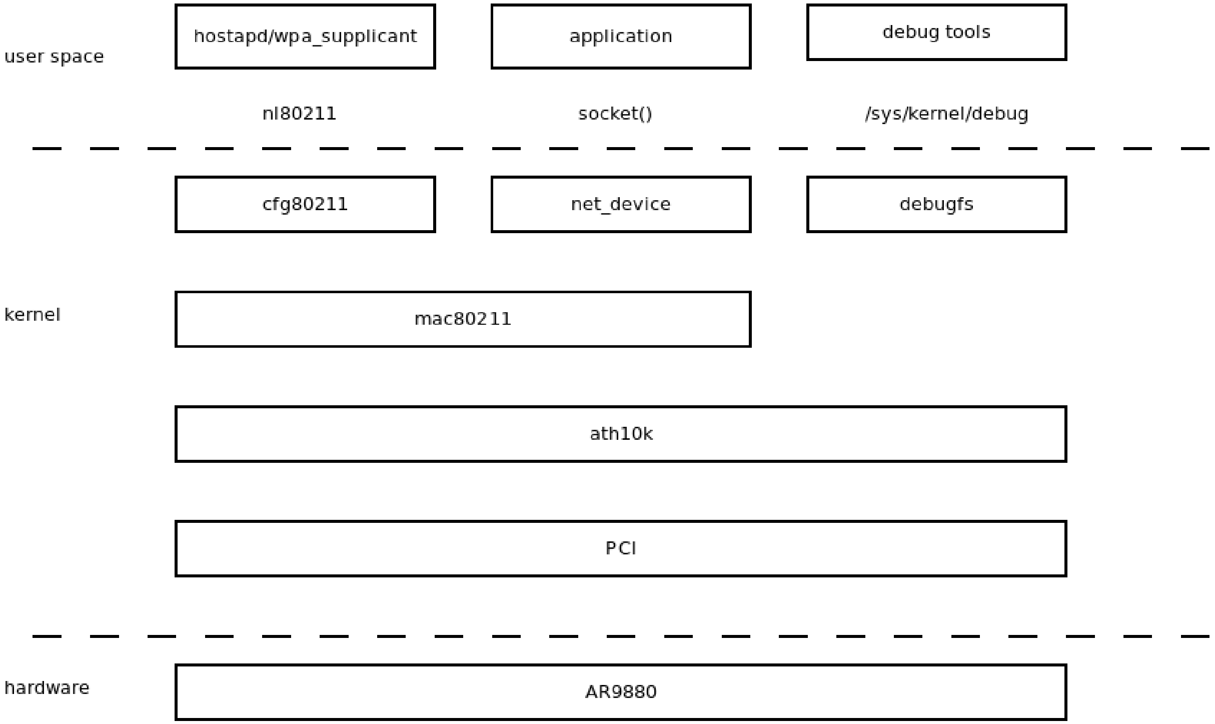
Country	Certification	
USA	FCC	FC
EU	RED (EN 300 328 V2.1.1 / EN 301 893 V2.1.1)	CE
CANADA	IC	
JAPAN	MIC	
KOREA	KCC	KC

http://www.sparklan.com/p2-products-detail.php?PKey=4242hPnk8UyutJ8DrfObciPdX6zHBHr72jF_vySd&WPEA-352ACN

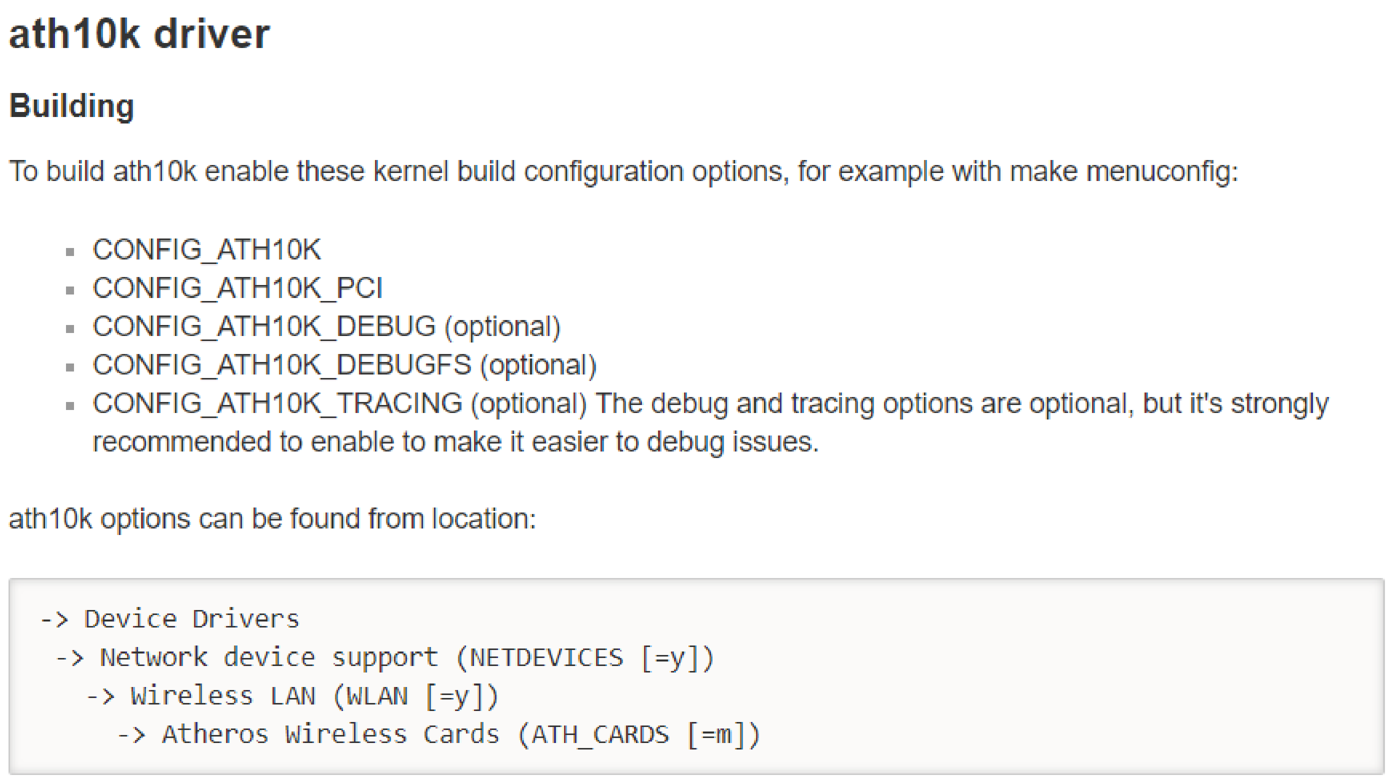
https://pt.aliexpress.com/item/Mini-pcie-M-dulo-QCA9880-WPEA-352ACN-802-11AC-Dual-Band-Wi-fi-Sem-Fio-placa/32844127884.html

5. 802.11ac 무선랜 구동하기(2)

- https://wireless.wiki.kernel.org/en/users/Drivers/ath10k
  - 이 site를 참고하면 다른 802.11ac 지원 miniPCIe module을 찾을 수 있음.
  - 뿐만 아니라, ath10k 관련 device driver 및 hostpad 등 관련 설정 내용을 파악할 수 있음.



5. 802.11ac 무선랜 구동하기(3)



ath10k driver

Building

To build ath10k enable these kernel build configuration options, for example with make menuconfig:

- CONFIG_ATH10K
- CONFIG_ATH10K_PCI
- CONFIG_ATH10K_DEBUG (optional)
- CONFIG_ATH10K_DEBUGFS (optional)
- CONFIG_ATH10K_TRACING (optional) The debug and tracing options are optional, but it's strongly recommended to enable to make it easier to debug issues.

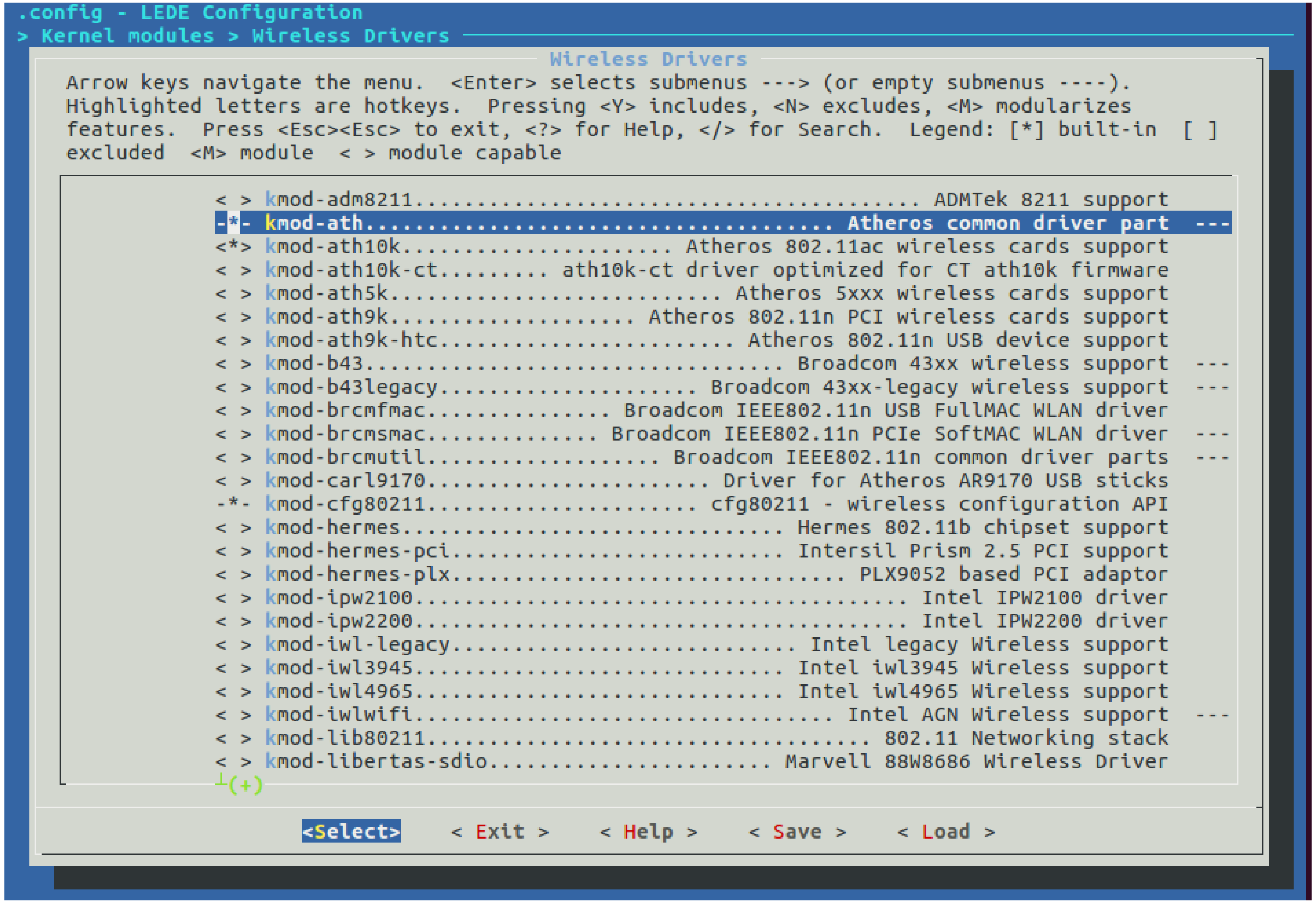
ath10k options can be found from location:

```
-> Device Drivers
 -> Network device support (NETDEVICES [=y])
   -> Wireless LAN (WLAN [=y])
     -> Atheros Wireless Cards (ATH_CARDS [=m])
```

https://wireless.wiki.kernel.org/en/users/drivers/ath10k/configuration

5. 802.11ac 무선랜 구동하기(4) – ath10k kernel module

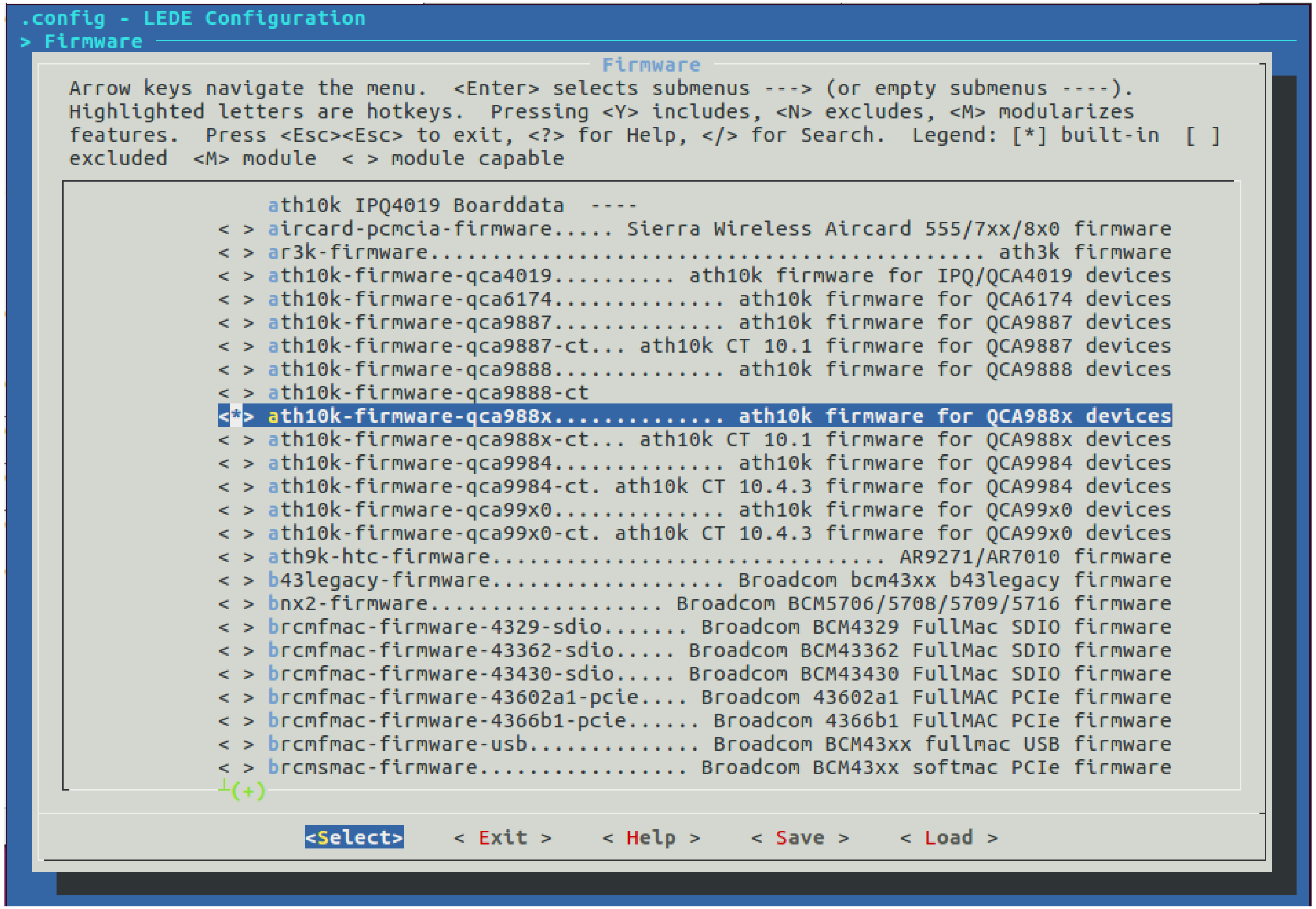
$ make menuconfig



```
.config - LEDE Configuration
> Kernel modules > Wireless Drivers ─────────────────────────────────────────
                          Wireless Drivers
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).
  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

            < > kmod-adm8211...................................... ADMTek 8211 support
            -*- kmod-ath................................. Atheros common driver part  ---
            <*> kmod-ath10k.................... Atheros 802.11ac wireless cards support
            < > kmod-ath10k-ct......... ath10k-ct driver optimized for CT ath10k firmware
            < > kmod-ath5k........................... Atheros 5xxx wireless cards support
            < > kmod-ath9k.................... Atheros 802.11n PCI wireless cards support
            < > kmod-ath9k-htc........................ Atheros 802.11n USB device support
            < > kmod-b43.............................. Broadcom 43xx wireless support  ---
            < > kmod-b43legacy..................... Broadcom 43xx-legacy wireless support  ---
            < > kmod-brcmfmac............... Broadcom IEEE802.11n USB FullMAC WLAN driver
            < > kmod-brcmsmac.............. Broadcom IEEE802.11n PCIe SoftMAC WLAN driver  ---
            < > kmod-brcmutil................... Broadcom IEEE802.11n common driver parts  ---
            < > kmod-carl9170........................ Driver for Atheros AR9170 USB sticks
            -*- kmod-cfg80211..................... cfg80211 - wireless configuration API
            < > kmod-hermes.............................. Hermes 802.11b chipset support
            < > kmod-hermes-pci.......................... Intersil Prism 2.5 PCI support
            < > kmod-hermes-plx.............................. PLX9052 based PCI adaptor
            < > kmod-ipw2100........................................ Intel IPW2100 driver
            < > kmod-ipw2200........................................ Intel IPW2200 driver
            < > kmod-iwl-legacy........................... Intel legacy Wireless support
            < > kmod-iwl3945.............................. Intel iwl3945 Wireless support
            < > kmod-iwl4965.............................. Intel iwl4965 Wireless support
            < > kmod-iwlwifi.................................. Intel AGN Wireless support  ---
            < > kmod-lib80211..................................... 802.11 Networking stack
            < > kmod-libertas-sdio...................... Marvell 88W8686 Wireless Driver
            └(+)

                <Select>    < Exit >    < Help >    < Save >    < Load >
```

5. 802.11ac 무선랜 구동하기(5) – ath10k firmware

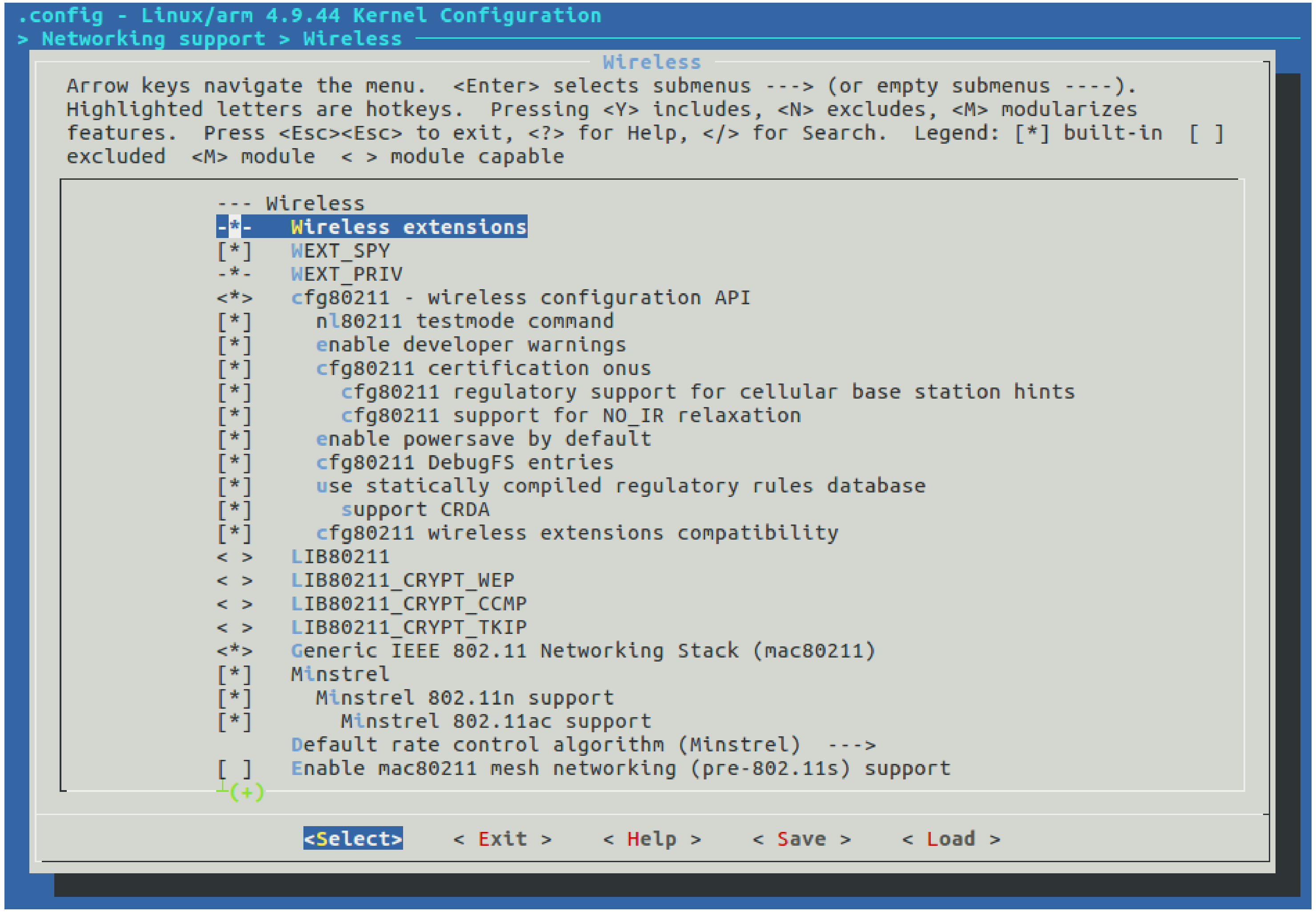
$ make menuconfig



```
.config - LEDE Configuration
> Firmware ─────────────────────────────────────────────────────────────────
                                 Firmware
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).
  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

              ath10k IPQ4019 Boarddata  ----
          < > aircard-pcmcia-firmware..... Sierra Wireless Aircard 555/7xx/8x0 firmware
          < > ar3k-firmware............................................. ath3k firmware
          < > ath10k-firmware-qca4019.......... ath10k firmware for IPQ/QCA4019 devices
          < > ath10k-firmware-qca6174.............. ath10k firmware for QCA6174 devices
          < > ath10k-firmware-qca9887.............. ath10k firmware for QCA9887 devices
          < > ath10k-firmware-qca9887-ct... ath10k CT 10.1 firmware for QCA9887 devices
          < > ath10k-firmware-qca9888.............. ath10k firmware for QCA9888 devices
          < > ath10k-firmware-qca9888-ct
          <*> ath10k-firmware-qca988x.............. ath10k firmware for QCA988x devices
          < > ath10k-firmware-qca988x-ct... ath10k CT 10.1 firmware for QCA988x devices
          < > ath10k-firmware-qca9984.............. ath10k firmware for QCA9984 devices
          < > ath10k-firmware-qca9984-ct. ath10k CT 10.4.3 firmware for QCA9984 devices
          < > ath10k-firmware-qca99x0.............. ath10k firmware for QCA99x0 devices
          < > ath10k-firmware-qca99x0-ct. ath10k CT 10.4.3 firmware for QCA99x0 devices
          < > ath9k-htc-firmware................................. AR9271/AR7010 firmware
          < > b43legacy-firmware................... Broadcom bcm43xx b43legacy firmware
          < > bnx2-firmware................... Broadcom BCM5706/5708/5709/5716 firmware
          < > brcmfmac-firmware-4329-sdio....... Broadcom BCM4329 FullMac SDIO firmware
          < > brcmfmac-firmware-43362-sdio..... Broadcom BCM43362 FullMac SDIO firmware
          < > brcmfmac-firmware-43430-sdio..... Broadcom BCM43430 FullMac SDIO firmware
          < > brcmfmac-firmware-43602a1-pcie.... Broadcom 43602a1 FullMAC PCIe firmware
          < > brcmfmac-firmware-4366b1-pcie...... Broadcom 4366b1 FullMAC PCIe firmware
          < > brcmfmac-firmware-usb............. Broadcom BCM43xx fullmac USB firmware
          < > brcmsmac-firmware................. Broadcom BCM43xx softmac PCIe firmware
          ┴(+)

           <Select>    < Exit >    < Help >    < Save >    < Load >
```

5. 802.11ac 무선랜 구동하기(6) – kernel config 조정(1)

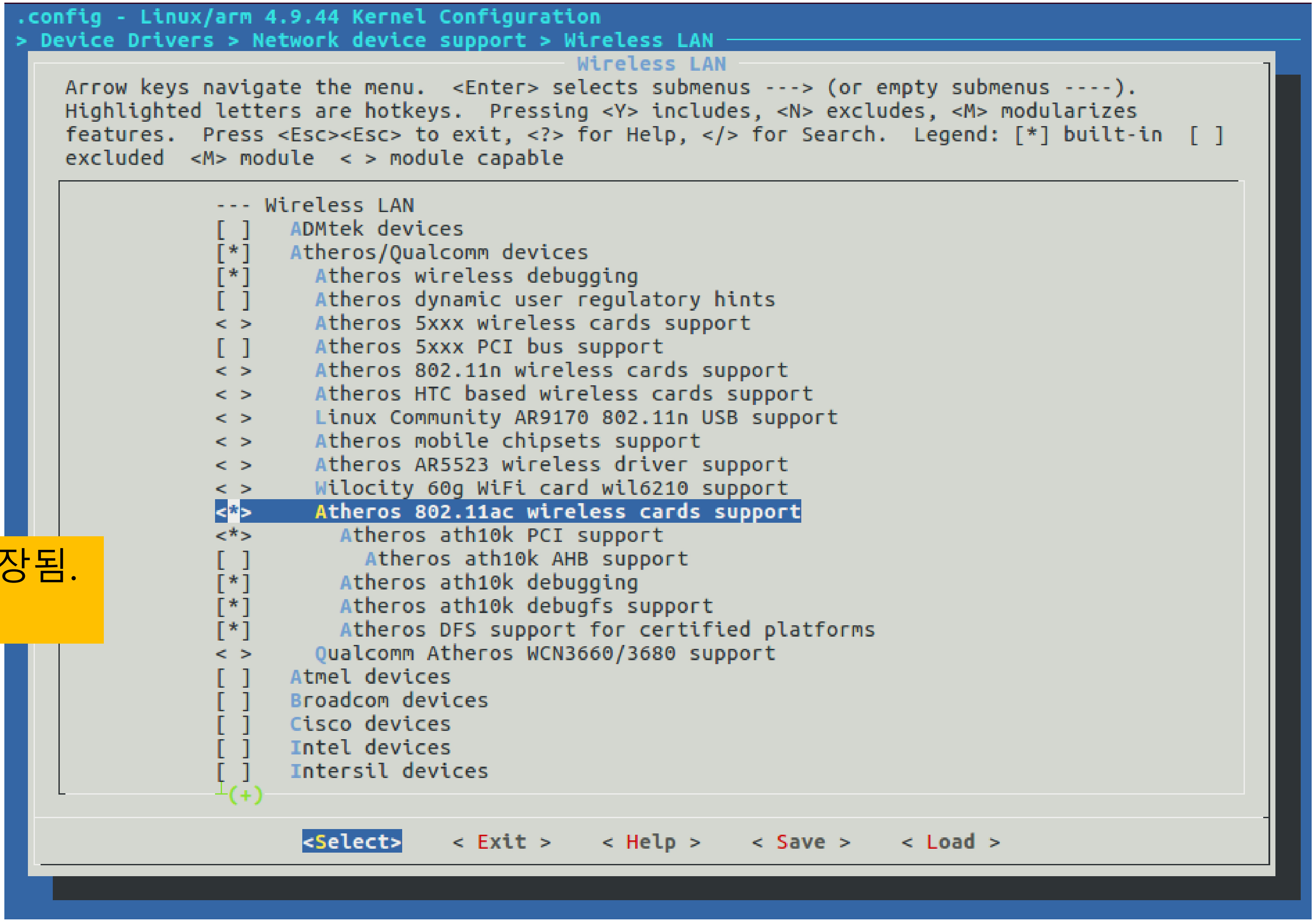
$ make kernel_menuconfig



```
.config - Linux/arm 4.9.44 Kernel Configuration
> Networking support > Wireless ─────────────────────────────────────────────
                                  Wireless
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).
  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
            -*-   Wireless extensions
            [*]   WEXT_SPY
            -*-   WEXT_PRIV
            <*>   cfg80211 - wireless configuration API
            [*]     nl80211 testmode command
            [*]     enable developer warnings
            [*]     cfg80211 certification onus
            [*]       cfg80211 regulatory support for cellular base station hints
            [*]       cfg80211 support for NO_IR relaxation
            [*]     enable powersave by default
            [*]     cfg80211 DebugFS entries
            [*]     use statically compiled regulatory rules database
            [*]       support CRDA
            [*]     cfg80211 wireless extensions compatibility
            < >   LIB80211
            < >   LIB80211_CRYPT_WEP
            < >   LIB80211_CRYPT_CCMP
            < >   LIB80211_CRYPT_TKIP
            <*>   Generic IEEE 802.11 Networking Stack (mac80211)
            [*]   Minstrel
            [*]     Minstrel 802.11n support
            [*]       Minstrel 802.11ac support
                  Default rate control algorithm (Minstrel)  --->
            [ ]   Enable mac80211 mesh networking (pre-802.11s) support
            └(+)

              <Select>    < Exit >    < Help >    < Save >    < Load >
```

5. 802.11ac 무선랜 구동하기(6) – kernel config 조정(2)

$ make kernel_menuconfig



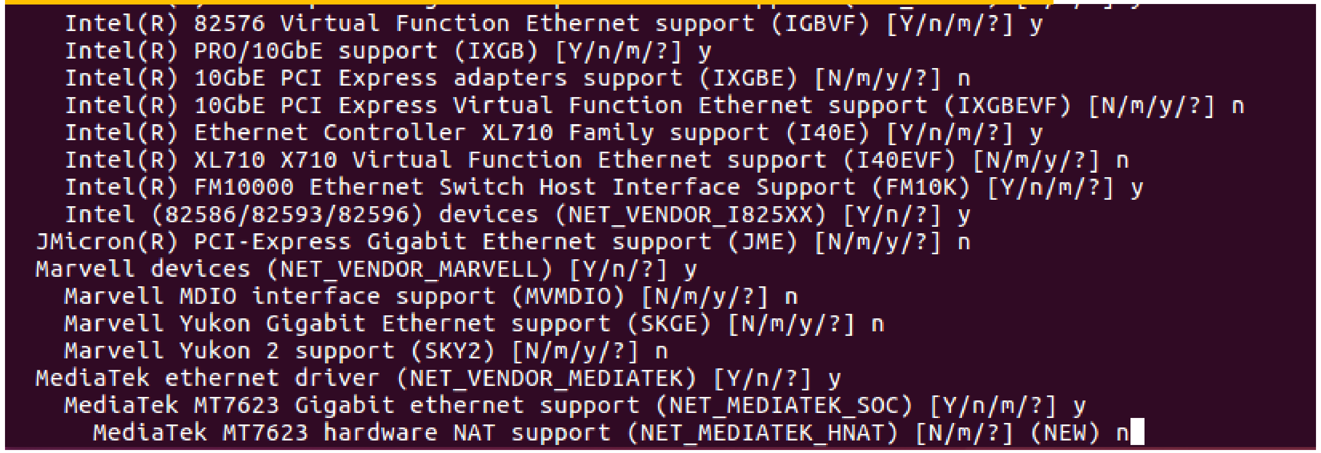
```
.config - Linux/arm 4.9.44 Kernel Configuration
> Device Drivers > Network device support > Wireless LAN ──────────────────────
                              Wireless LAN
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).
  Highlighted letters are hotkeys.  Pressing <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 LAN
            [ ]   ADMtek devices
            [*]   Atheros/Qualcomm devices
            [*]     Atheros wireless debugging
            [ ]     Atheros dynamic user regulatory hints
            < >     Atheros 5xxx wireless cards support
            [ ]     Atheros 5xxx PCI bus support
            < >     Atheros 802.11n wireless cards support
            < >     Atheros HTC based wireless cards support
            < >     Linux Community AR9170 802.11n USB support
            < >     Atheros mobile chipsets support
            < >     Atheros AR5523 wireless driver support
            < >     Wilocity 60g WiFi card wil6210 support
            <*>     Atheros 802.11ac wireless cards support
            <*>       Atheros ath10k PCI support
            [ ]         Atheros ath10k AHB support
            [*]       Atheros ath10k debugging
            [*]       Atheros ath10k debugfs support
            [*]       Atheros DFS support for certified platforms
            < >     Qualcomm Atheros WCN3660/3680 support
            [ ]   Atmel devices
            [ ]   Broadcom devices
            [ ]   Cisco devices
            [ ]   Intel devices
            [ ]   Intersil devices
            └(+)

       <Select>    < Exit >    < Help >    < Save >    < Load >
```

참고: 변경한 kernel config는 아래 위치에 저장됨.
✓target/linux/mediatek/config-4.9

5. 802.11ac 무선랜 구동하기(7) – config 조정 후 재compile

- $ make menuconfig
- $ make kernel_menuconfig
- $ make -j1 V=99
- or
- $ make menuconfig
- $ make kernel_menuconfig
- $ make target/linux/clean
- $ make -j1 V=99

<compile 중 아래에서 멈춤 – Enter or n 입력하면 됨>



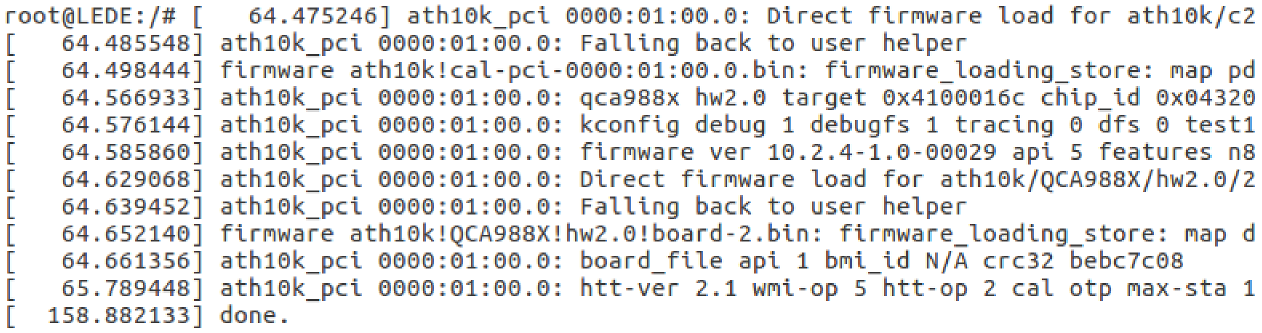
```
    Intel(R) 82576 Virtual Function Ethernet support (IGBVF) [Y/n/m/?] y
    Intel(R) PRO/10GbE support (IXGB) [Y/n/m/?] y
    Intel(R) 10GbE PCI Express adapters support (IXGBE) [N/m/y/?] n
    Intel(R) 10GbE PCI Express Virtual Function Ethernet support (IXGBEVF) [N/m/y/?] n
    Intel(R) Ethernet Controller XL710 Family support (I40E) [Y/n/m/?] y
    Intel(R) XL710 X710 Virtual Function Ethernet support (I40EVF) [N/m/y/?] n
    Intel(R) FM10000 Ethernet Switch Host Interface Support (FM10K) [Y/n/m/?] y
    Intel (82586/82593/82596) devices (NET_VENDOR_I825XX) [Y/n/?] y
  JMicron(R) PCI-Express Gigabit Ethernet support (JME) [N/m/y/?] n
  Marvell devices (NET_VENDOR_MARVELL) [Y/n/?] y
    Marvell MDIO interface support (MVMDIO) [N/m/y/?] n
    Marvell Yukon Gigabit Ethernet support (SKGE) [N/m/y/?] n
    Marvell Yukon 2 support (SKY2) [N/m/y/?] n
  MediaTek ethernet driver (NET_VENDOR_MEDIATEK) [Y/n/?] y
    MediaTek MT7623 Gigabit ethernet support (NET_MEDIATEK_SOC) [Y/n/m/?] y
      MediaTek MT7623 hardware NAT support (NET_MEDIATEK_HNAT) [N/m/?] (NEW) n
```

5. 802.11ac 무선랜 구동하기(8) – ath10k kernel message

이후 다시, 아래 명령 수행 후, 부팅 시도

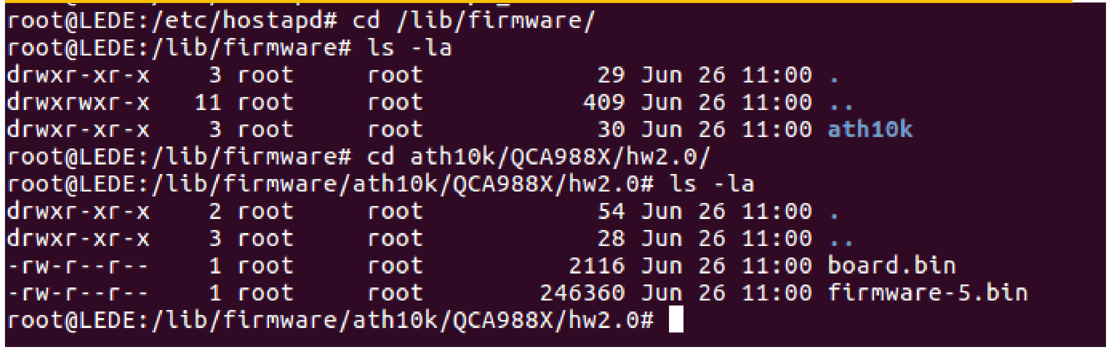
$ **sudo dd if=./mtk-bpi-r2-SD.img of=/dev/sdc**

<부팅 후, kernel message – 이 부분이 정상이 아닌 것 같군>



```
root@LEDE:/# [   64.475246] ath10k_pci 0000:01:00.0: Direct firmware load for ath10k/c2
[   64.485548] ath10k_pci 0000:01:00.0: Falling back to user helper
[   64.498444] firmware ath10k!cal-pci-0000:01:00.0.bin: firmware_loading_store: map pd
[   64.566933] ath10k_pci 0000:01:00.0: qca988x hw2.0 target 0x4100016c chip_id 0x04320
[   64.576144] ath10k_pci 0000:01:00.0: kconfig debug 1 debugfs 1 tracing 0 dfs 0 test1
[   64.585860] ath10k_pci 0000:01:00.0: firmware ver 10.2.4-1.0-00029 api 5 features n8
[   64.629068] ath10k_pci 0000:01:00.0: Direct firmware load for ath10k/QCA988X/hw2.0/2
[   64.639452] ath10k_pci 0000:01:00.0: Falling back to user helper
[   64.652140] firmware ath10k!QCA988X!hw2.0!board-2.bin: firmware_loading_store: map d
[   64.661356] ath10k_pci 0000:01:00.0: board_file api 1 bmi_id N/A crc32 bebc7c08
[   65.789448] ath10k_pci 0000:01:00.0: htt-ver 2.1 wmi-op 5 htt-op 2 cal otp max-sta 1
[  158.882133] done.
```

Ath10k용 firmware가 정상인지 여부도 확인이 필요함.

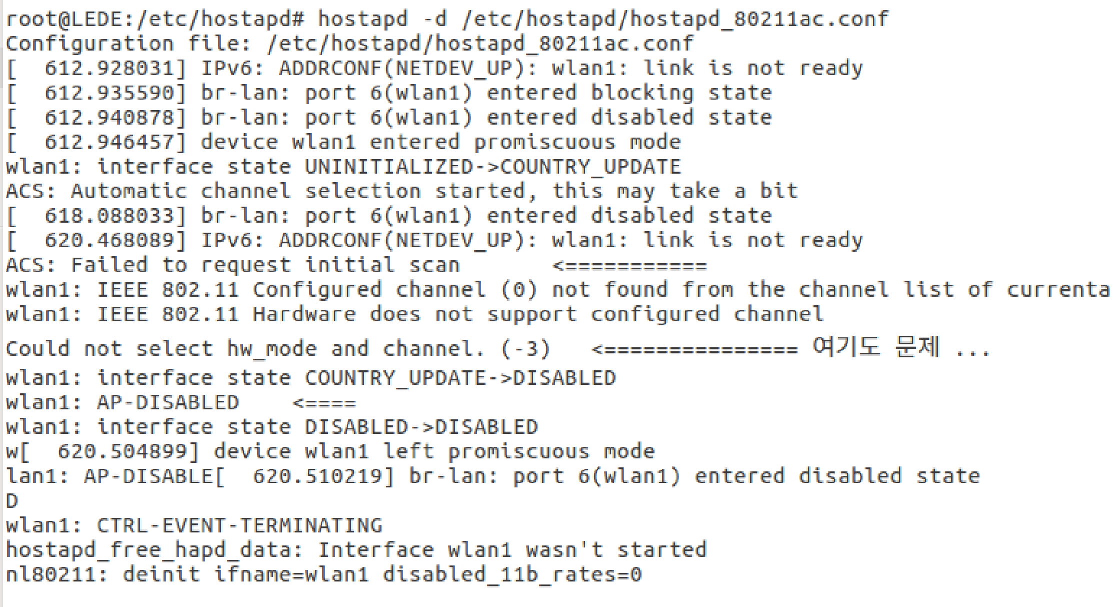


```
root@LEDE:/etc/hostapd# cd /lib/firmware/
root@LEDE:/lib/firmware# ls -la
drwxr-xr-x    3 root     root            29 Jun 26 11:00 .
drwxrwxr-x   11 root     root           409 Jun 26 11:00 ..
drwxr-xr-x    3 root     root            30 Jun 26 11:00 ath10k
root@LEDE:/lib/firmware# cd ath10k/QCA988X/hw2.0/
root@LEDE:/lib/firmware/ath10k/QCA988X/hw2.0# ls -la
drwxr-xr-x    2 root     root            54 Jun 26 11:00 .
drwxr-xr-x    3 root     root            28 Jun 26 11:00 ..
-rw-r--r--    1 root     root          2116 Jun 26 11:00 board.bin
-rw-r--r--    1 root     root        246360 Jun 26 11:00 firmware-5.bin
root@LEDE:/lib/firmware/ath10k/QCA988X/hw2.0# █
```

(*) 비정상 동작 이유가 firmware와 연관이 있을 것으로 예상하고
열심히 삽질해 봄 ☹ 지면 관계상 관련 내용을 모두 포함시키지는 않음.

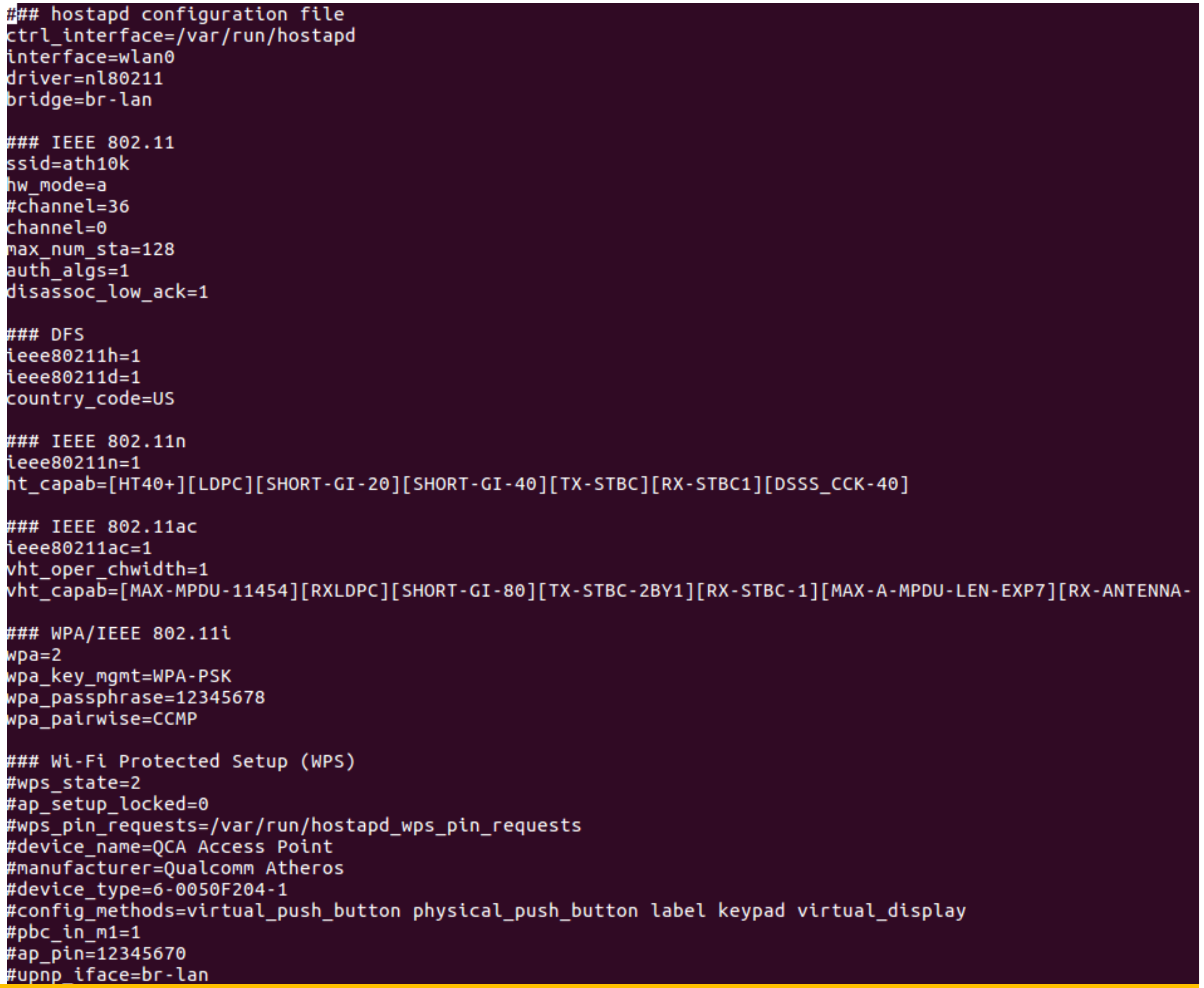
5. 802.11ac 무선랜 구동하기(9) – hostapd 구동(1)

예상대로 hostapd 실행 시, 에러 발생함.



```
root@LEDE:/etc/hostapd# hostapd -d /etc/hostapd/hostapd_80211ac.conf
Configuration file: /etc/hostapd/hostapd_80211ac.conf
[  612.928031] IPv6: ADDRCONF(NETDEV_UP): wlan1: link is not ready
[  612.935590] br-lan: port 6(wlan1) entered blocking state
[  612.940878] br-lan: port 6(wlan1) entered disabled state
[  612.946457] device wlan1 entered promiscuous mode
wlan1: interface state UNINITIALIZED->COUNTRY_UPDATE
ACS: Automatic channel selection started, this may take a bit
[  618.088033] br-lan: port 6(wlan1) entered disabled state
[  620.468089] IPv6: ADDRCONF(NETDEV_UP): wlan1: link is not ready
ACS: Failed to request initial scan       <===========
wlan1: IEEE 802.11 Configured channel (0) not found from the channel list of currenta
wlan1: IEEE 802.11 Hardware does not support configured channel
Could not select hw_mode and channel. (-3)   <================ 여기도 문제 ...
wlan1: interface state COUNTRY_UPDATE->DISABLED
wlan1: AP-DISABLED    <====
wlan1: interface state DISABLED->DISABLED
w[  620.504899] device wlan1 left promiscuous mode
lan1: AP-DISABLE[  620.510219] br-lan: port 6(wlan1) entered disabled state
D
wlan1: CTRL-EVENT-TERMINATING
hostapd_free_hapd_data: Interface wlan1 wasn't started
nl80211: deinit ifname=wlan1 disabled_11b_rates=0
```

/etc/hostapd/hostapd.conf 파일 생성

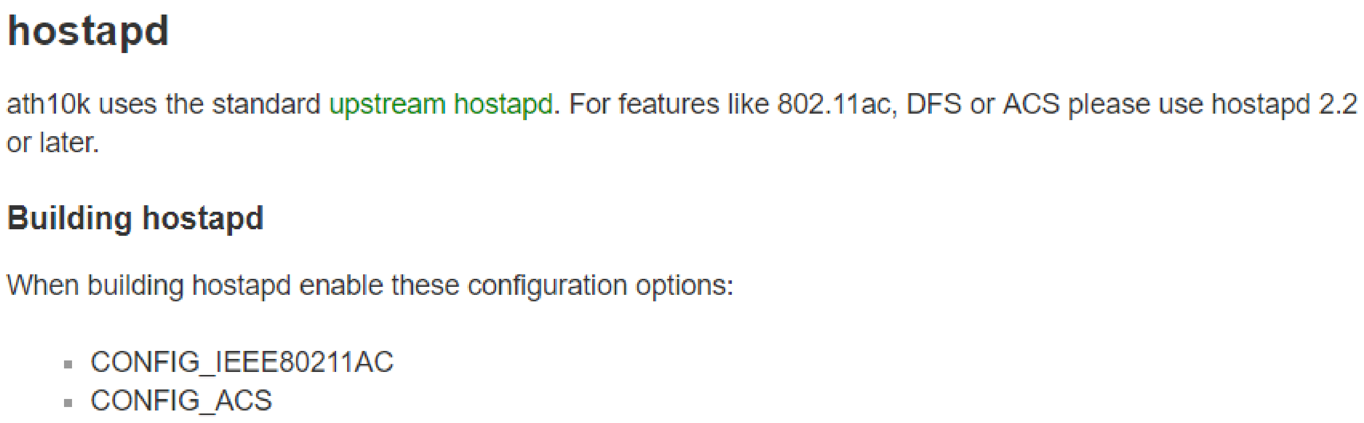


```
### hostapd configuration file
ctrl_interface=/var/run/hostapd
interface=wlan0
driver=nl80211
bridge=br-lan

### IEEE 802.11
ssid=ath10k
hw_mode=a
#channel=36
channel=0
max_num_sta=128
auth_algs=1
disassoc_low_ack=1

### DFS
ieee80211h=1
ieee80211d=1
country_code=US

### IEEE 802.11n
ieee80211n=1
ht_capab=[HT40+][LDPC][SHORT-GI-20][SHORT-GI-40][TX-STBC][RX-STBC1][DSSS_CCK-40]

### IEEE 802.11ac
ieee80211ac=1
vht_oper_chwidth=1
vht_capab=[MAX-MPDU-11454][RXLDPC][SHORT-GI-80][TX-STBC-2BY1][RX-STBC-1][MAX-A-MPDU-LEN-EXP7][RX-ANTENNA-

### WPA/IEEE 802.11i
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_passphrase=12345678
wpa_pairwise=CCMP

### Wi-Fi Protected Setup (WPS)
#wps_state=2
#ap_setup_locked=0
#wps_pin_requests=/var/run/hostapd_wps_pin_requests
#device_name=QCA Access Point
#manufacturer=Qualcomm Atheros
#device_type=6-0050F204-1
#config_methods=virtual_push_button physical_push_button label keypad virtual_display
#pbc_in_m1=1
#ap_pin=12345670
#upnp_iface=br-lan
```

https://wireless.wiki.kernel.org/en/users/drivers/ath10k/configuration

- 여기를 참조하여 작성함.

5. 802.11ac 무선랜 구동하기(9) – hostapd 구동(2)



hostapd

ath10k uses the standard upstream hostapd. For features like 802.11ac, DFS or ACS please use hostapd 2.2 or later.

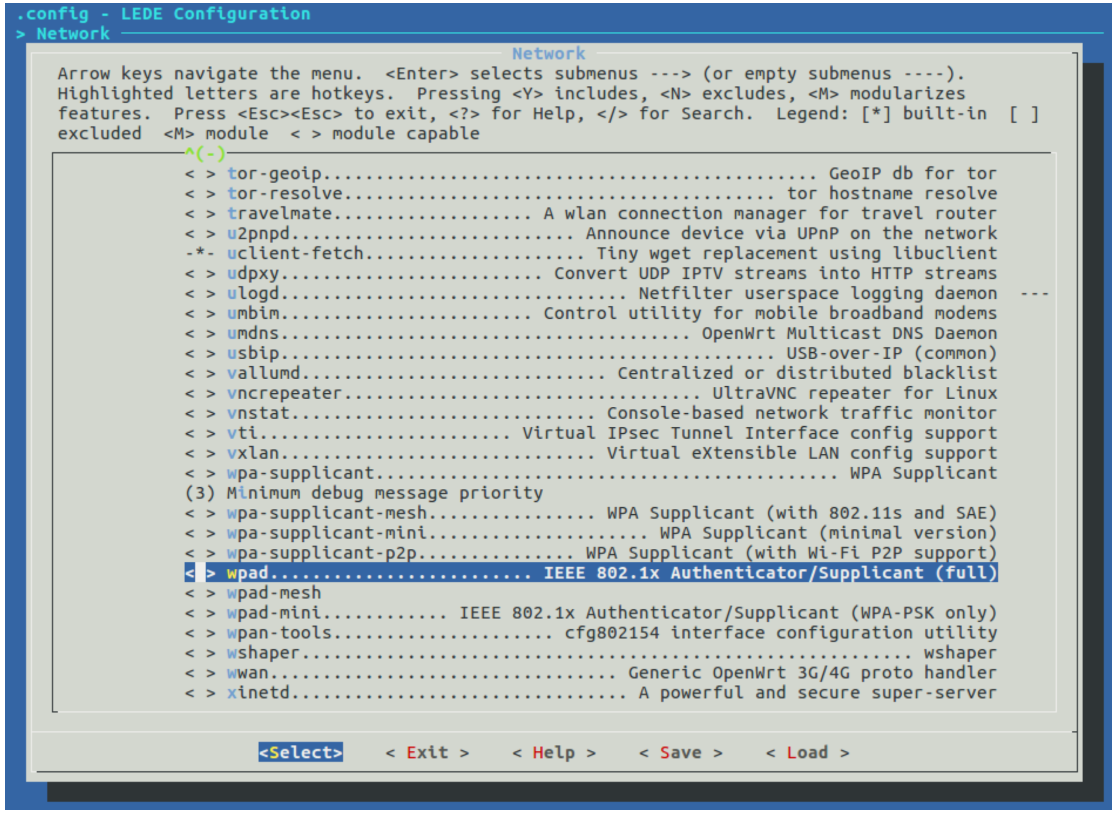
Building hostapd

When building hostapd enable these configuration options:

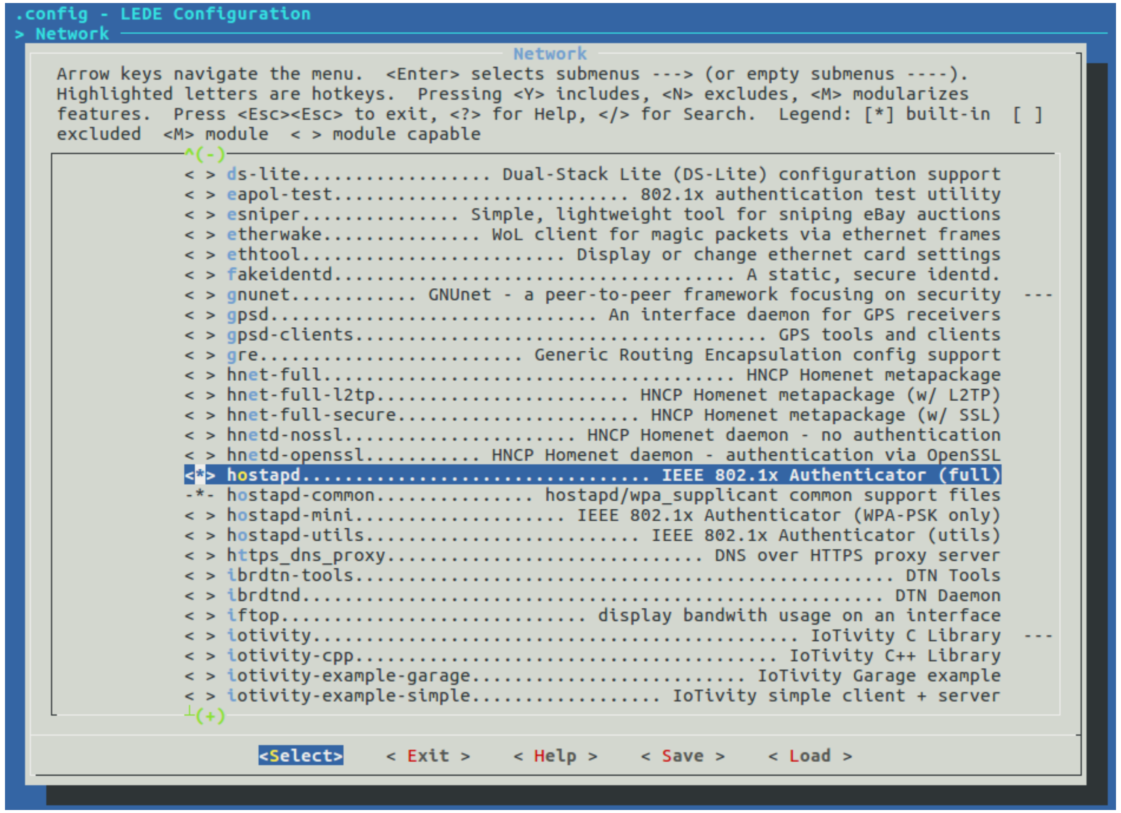
- CONFIG_IEEE80211AC
- CONFIG_ACS

(참고) hostapd의 주요 Config 내용은 package/network/services/hostapd/Makefile에 있음.

5. 802.11ac 무선랜 구동하기(9) – hostapd 구동(3)



```
.config - LEDE Configuration
> Network
                                    Network
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).
  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
              ^(-)
              < > tor-geoip............................................... GeoIP db for tor
              < > tor-resolve......................................... tor hostname resolve
              < > travelmate................... A wlan connection manager for travel router
              < > u2pnpd........................... Announce device via UPnP on the network
              -*- uclient-fetch..................... Tiny wget replacement using libuclient
              < > udpxy......................... Convert UDP IPTV streams into HTTP streams
              < > ulogd................................. Netfilter userspace logging daemon  ---
              < > umbim........................ Control utility for mobile broadband modems
              < > umdns....................................... OpenWrt Multicast DNS Daemon
              < > usbip............................................... USB-over-IP (common)
              < > vallumd............................. Centralized or distributed blacklist
              < > vncrepeater.................................. UltraVNC repeater for Linux
              < > vnstat............................. Console-based network traffic monitor
              < > vti........................ Virtual IPsec Tunnel Interface config support
              < > vxlan.............................. Virtual eXtensible LAN config support
              < > wpa-supplicant............................................ WPA Supplicant
              (3) Minimum debug message priority
              < > wpa-supplicant-mesh................ WPA Supplicant (with 802.11s and SAE)
              < > wpa-supplicant-mini..................... WPA Supplicant (minimal version)
              < > wpa-supplicant-p2p............... WPA Supplicant (with Wi-Fi P2P support)
              < > wpad......................... IEEE 802.1x Authenticator/Supplicant (full)
              < > wpad-mesh
              < > wpad-mini............ IEEE 802.1x Authenticator/Supplicant (WPA-PSK only)
              < > wpan-tools..................... cfg802154 interface configuration utility
              < > wshaper.......................................................... wshaper
              < > wwan................................. Generic OpenWrt 3G/4G proto handler
              < > xinetd................................ A powerful and secure super-server

                <Select>    < Exit >    < Help >    < Save >    < Load >
```



```
.config - LEDE Configuration
> Network
                                    Network
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).
  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
              ^(-)
              < > ds-lite.................. Dual-Stack Lite (DS-Lite) configuration support
              < > eapol-test............................ 802.1x authentication test utility
              < > esniper............... Simple, lightweight tool for sniping eBay auctions
              < > etherwake............... WoL client for magic packets via ethernet frames
              < > ethtool......................... Display or change ethernet card settings
              < > fakeidentd...................................... A static, secure identd.
              < > gnunet............ GNUnet - a peer-to-peer framework focusing on security  ---
              < > gpsd............................... An interface daemon for GPS receivers
              < > gpsd-clients....................................... GPS tools and clients
              < > gre......................... Generic Routing Encapsulation config support
              < > hnet-full....................................... HNCP Homenet metapackage
              < > hnet-full-l2tp........................ HNCP Homenet metapackage (w/ L2TP)
              < > hnet-full-secure....................... HNCP Homenet metapackage (w/ SSL)
              < > hnetd-nossl...................... HNCP Homenet daemon - no authentication
              < > hnetd-openssl........... HNCP Homenet daemon - authentication via OpenSSL
              <*> hostapd................................. IEEE 802.1x Authenticator (full)
              -*- hostapd-common............... hostapd/wpa_supplicant common support files
              < > hostapd-mini.................... IEEE 802.1x Authenticator (WPA-PSK only)
              < > hostapd-utils.......................... IEEE 802.1x Authenticator (utils)
              < > https_dns_proxy.............................. DNS over HTTPS proxy server
              < > ibrdtn-tools................................................... DTN Tools
              < > ibrdtnd.................................................... DTN Daemon
              < > iftop............................. display bandwith usage on an interface
              < > iotivity.............................................. IoTivity C Library  ---
              < > iotivity-cpp........................................ IoTivity C++ Library
              < > iotivity-example-garage.......................... IoTivity Garage example
              < > iotivity-example-simple.................. IoTivity simple client + server
              └(+)

                <Select>    < Exit >    < Help >    < Save >    < Load >
```

wpad를 내리고(disable), hostapd(full version)을 살려(enable) 보도록 하자.

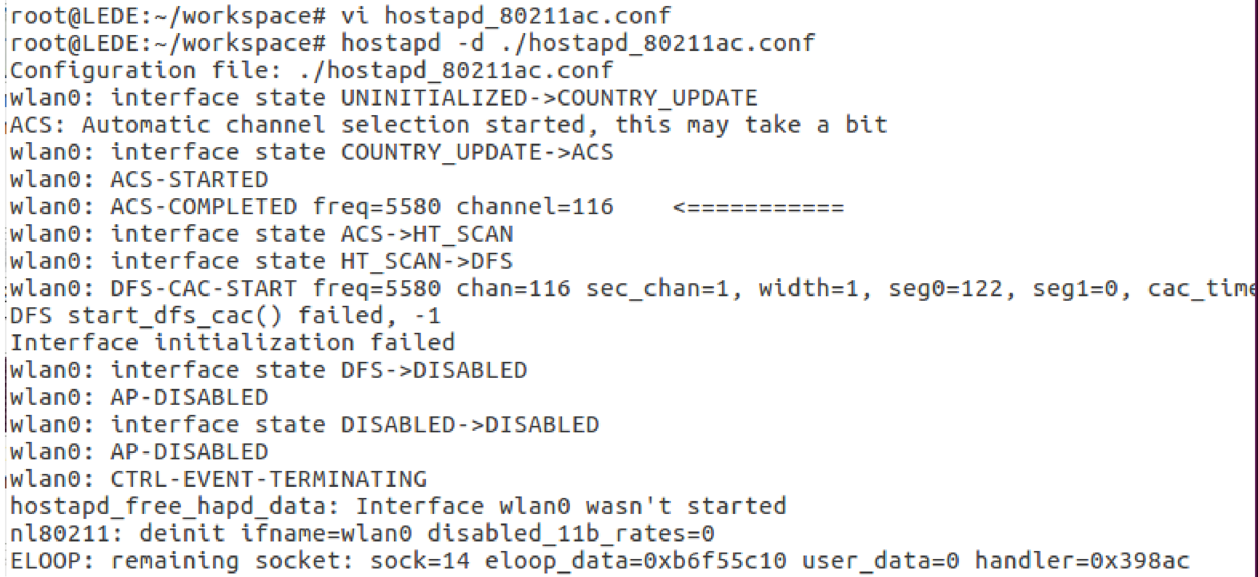
5. 802.11ac 무선랜 구동하기(9) – hostapd 구동(4)

- $ cd build_dir/target-arm_cortex-a7+neon-vfpv4_musl_eabi/hostapd-full/hostapd-2016-12-19-ad02e79d/hostapd
- $ vi .config

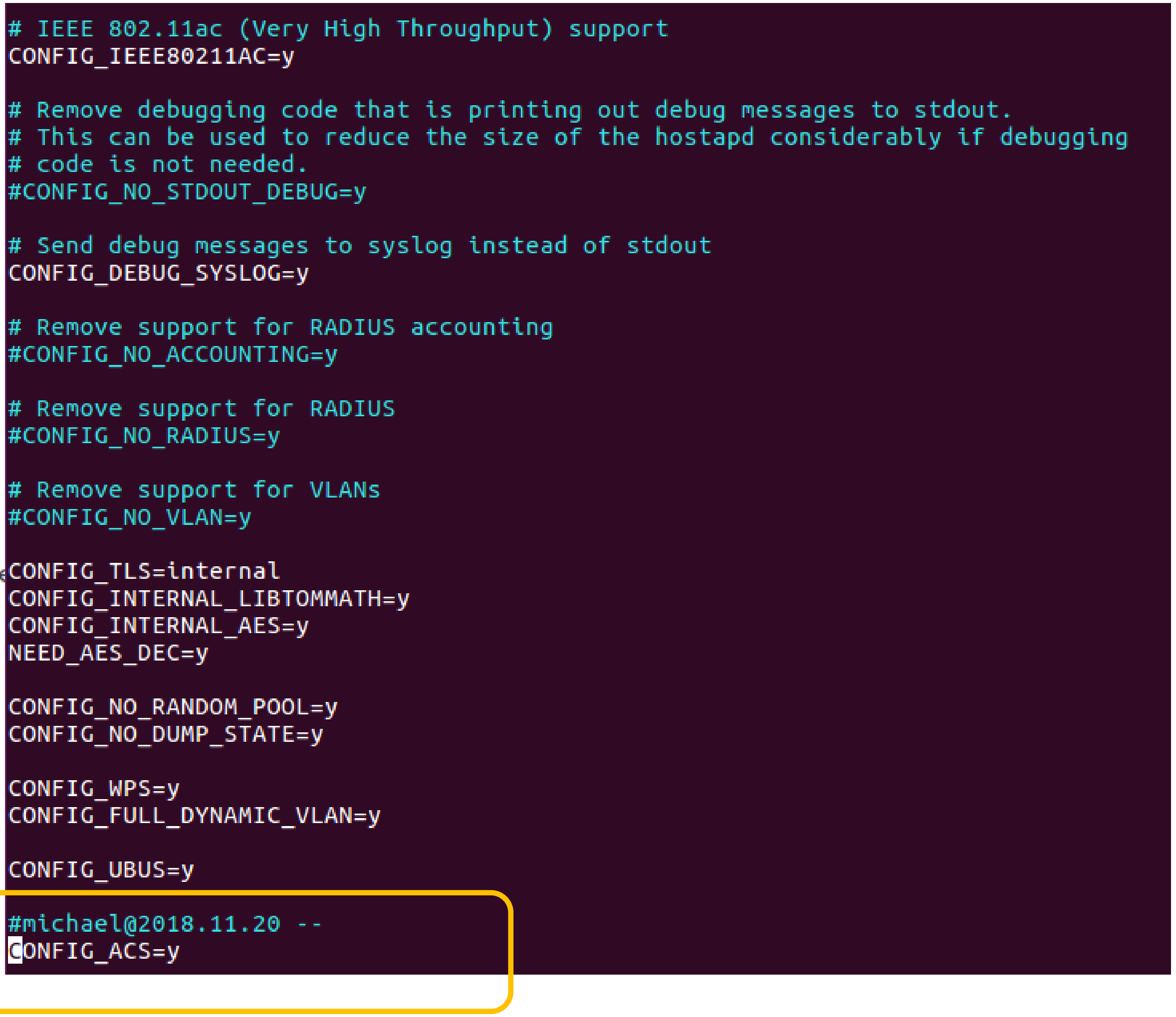
CONFIG_ACS=y 를 강제로 추가함.

이후 다시 테스트해 보았으나 동일한 문제 발생.

- 즉, hostapd는 동작하나, wlan이 살아나지 않음.



```
root@LEDE:~/workspace# vi hostapd_80211ac.conf
root@LEDE:~/workspace# hostapd -d ./hostapd_80211ac.conf
Configuration file: ./hostapd_80211ac.conf
wlan0: interface state UNINITIALIZED->COUNTRY_UPDATE
ACS: Automatic channel selection started, this may take a bit
wlan0: interface state COUNTRY_UPDATE->ACS
wlan0: ACS-STARTED
wlan0: ACS-COMPLETED freq=5580 channel=116    <===========
wlan0: interface state ACS->HT_SCAN
wlan0: interface state HT_SCAN->DFS
wlan0: DFS-CAC-START freq=5580 chan=116 sec_chan=1, width=1, seg0=122, seg1=0, cac_tim
DFS start_dfs_cac() failed, -1
Interface initialization failed
wlan0: interface state DFS->DISABLED
wlan0: AP-DISABLED
wlan0: interface state DISABLED->DISABLED
wlan0: AP-DISABLED
wlan0: CTRL-EVENT-TERMINATING
hostapd_free_hapd_data: Interface wlan0 wasn't started
nl80211: deinit ifname=wlan0 disabled_11b_rates=0
ELOOP: remaining socket: sock=14 eloop_data=0xb6f55c10 user_data=0 handler=0x398ac
```



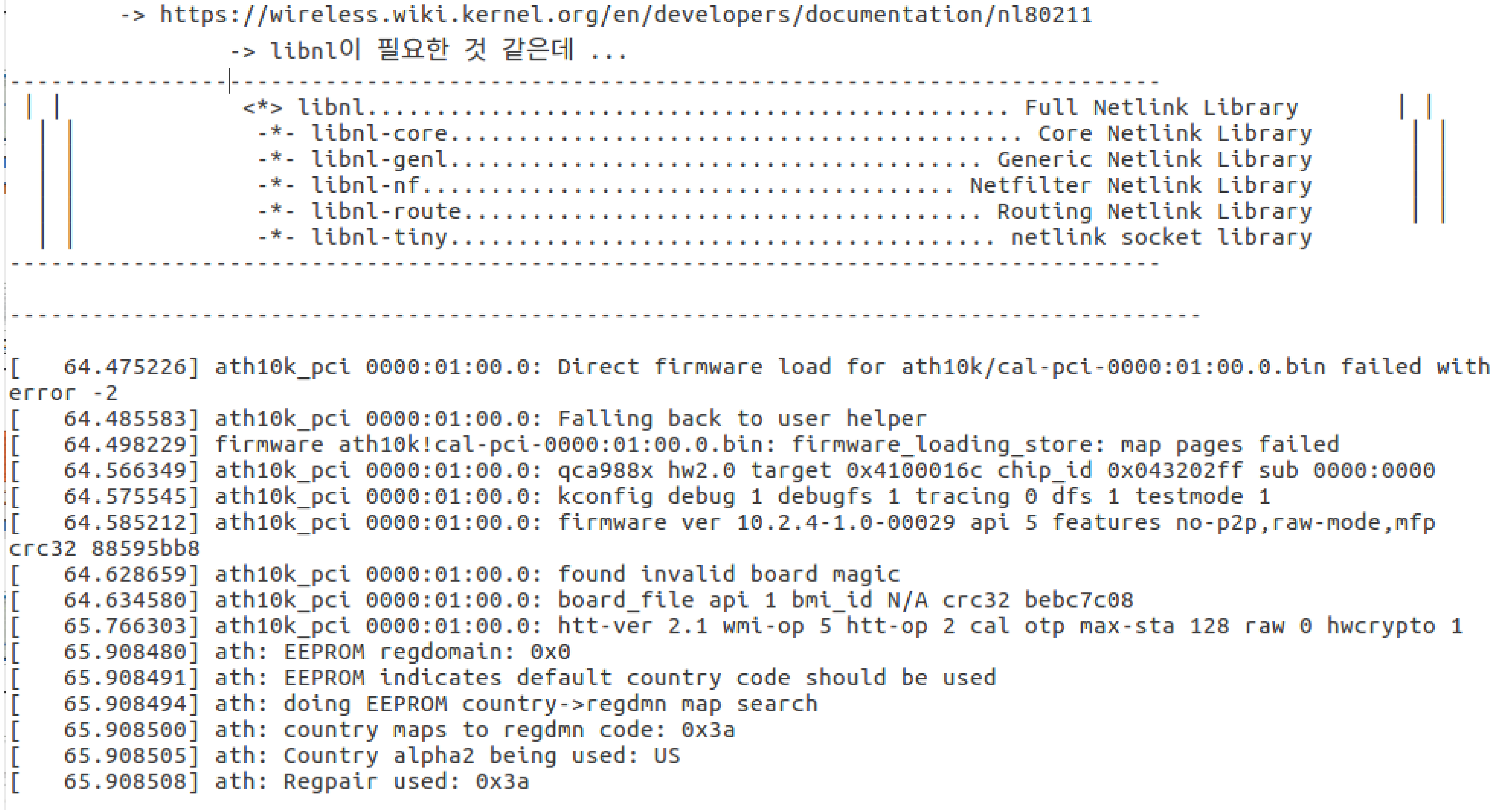
```
# IEEE 802.11ac (Very High Throughput) support
CONFIG_IEEE80211AC=y

# Remove debugging code that is printing out debug messages to stdout.
# This can be used to reduce the size of the hostapd considerably if debugging
# code is not needed.
#CONFIG_NO_STDOUT_DEBUG=y

# Send debug messages to syslog instead of stdout
CONFIG_DEBUG_SYSLOG=y

# Remove support for RADIUS accounting
#CONFIG_NO_ACCOUNTING=y

# Remove support for RADIUS
#CONFIG_NO_RADIUS=y

# Remove support for VLANs
#CONFIG_NO_VLAN=y

CONFIG_TLS=internal
CONFIG_INTERNAL_LIBTOMMATH=y
CONFIG_INTERNAL_AES=y
NEED_AES_DEC=y

CONFIG_NO_RANDOM_POOL=y
CONFIG_NO_DUMP_STATE=y

CONFIG_WPS=y
CONFIG_FULL_DYNAMIC_VLAN=y

CONFIG_UBUS=y

#michael@2018.11.20 --
CONFIG_ACS=y
```

5. 802.11ac 무선랜 구동하기(9) – hostapd 구동(5)

libnl library를 enable 시킴.

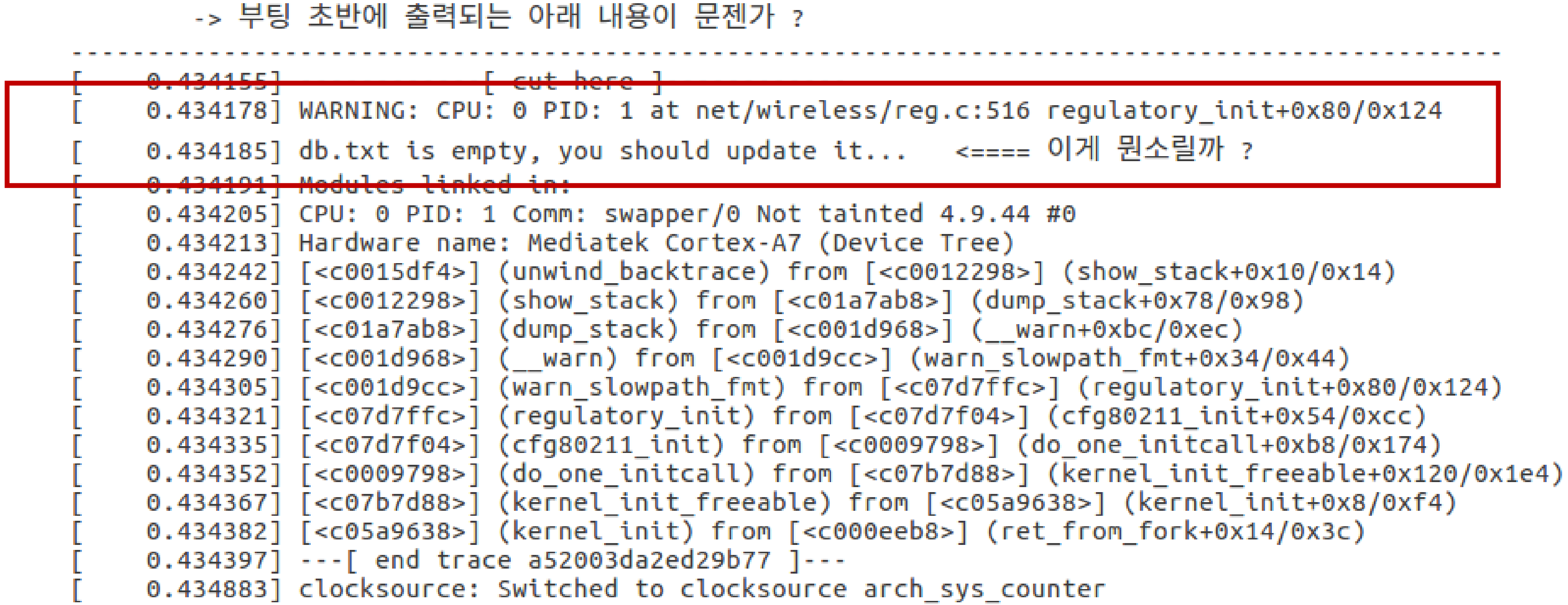
이후 다시 테스트해 보았으나 동일한 문제 발생.

- 즉, hostapd는 동작하나, wlan이 살아나지 않음.



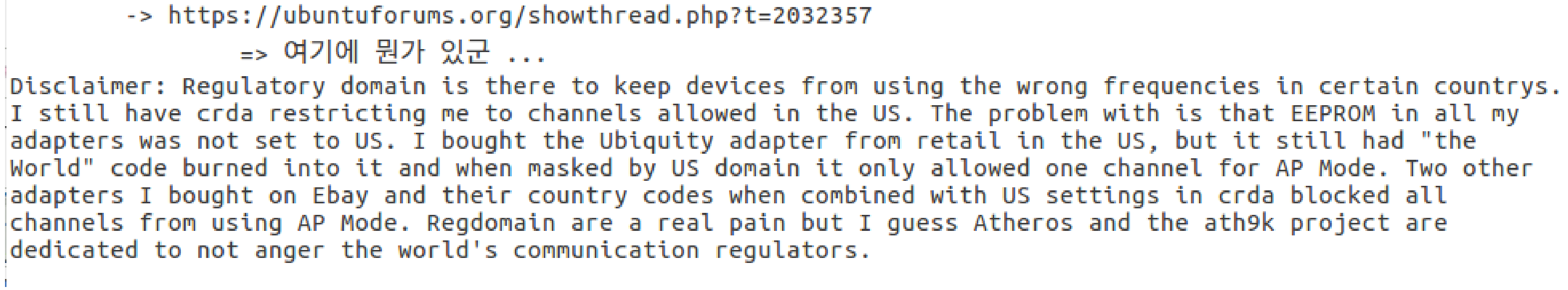
```
        -> https://wireless.wiki.kernel.org/en/developers/documentation/nl80211
                -> libnl이 필요한 것 같은데 ...
----------------|-----------------------------------------------------------------
 | |            <*> libnl...................................... Full Netlink Library       | |
  | |            -*- libnl-core................................. Core Netlink Library       | |
  | |            -*- libnl-genl.............................. Generic Netlink Library       | |
  | |            -*- libnl-nf............................. Netfilter Netlink Library       | |
  | |            -*- libnl-route.............................. Routing Netlink Library
  | |            -*- libnl-tiny................................ netlink socket library
----------------------------------------------------------------------------------

-------------------------------------------------------------------------------------

[   64.475226] ath10k_pci 0000:01:00.0: Direct firmware load for ath10k/cal-pci-0000:01:00.0.bin failed with
error -2
[   64.485583] ath10k_pci 0000:01:00.0: Falling back to user helper
[   64.498229] firmware ath10k!cal-pci-0000:01:00.0.bin: firmware_loading_store: map pages failed
[   64.566349] ath10k_pci 0000:01:00.0: qca988x hw2.0 target 0x4100016c chip_id 0x043202ff sub 0000:0000
[   64.575545] ath10k_pci 0000:01:00.0: kconfig debug 1 debugfs 1 tracing 0 dfs 1 testmode 1
[   64.585212] ath10k_pci 0000:01:00.0: firmware ver 10.2.4-1.0-00029 api 5 features no-p2p,raw-mode,mfp
crc32 88595bb8
[   64.628659] ath10k_pci 0000:01:00.0: found invalid board magic
[   64.634580] ath10k_pci 0000:01:00.0: board_file api 1 bmi_id N/A crc32 bebc7c08
[   65.766303] ath10k_pci 0000:01:00.0: htt-ver 2.1 wmi-op 5 htt-op 2 cal otp max-sta 128 raw 0 hwcrypto 1
[   65.908480] ath: EEPROM regdomain: 0x0
[   65.908491] ath: EEPROM indicates default country code should be used
[   65.908494] ath: doing EEPROM country->regdmn map search
[   65.908500] ath: country maps to regdmn code: 0x3a
[   65.908505] ath: Country alpha2 being used: US
[   65.908508] ath: Regpair used: 0x3a
```

5. 802.11ac 무선랜 구동하기(9) – hostapd 구동(6)



```
        -> 부팅 초반에 출력되는 아래 내용이 문젠가 ?
-------------------------------------------------------------------------------------------
[    0.434155]               [ cut here ]
[    0.434178] WARNING: CPU: 0 PID: 1 at net/wireless/reg.c:516 regulatory_init+0x80/0x124
[    0.434185] db.txt is empty, you should update it...   <==== 이게 뭔소릴까 ?
[    0.434191] Modules linked in:
[    0.434205] CPU: 0 PID: 1 Comm: swapper/0 Not tainted 4.9.44 #0
[    0.434213] Hardware name: Mediatek Cortex-A7 (Device Tree)
[    0.434242] [<c0015df4>] (unwind_backtrace) from [<c0012298>] (show_stack+0x10/0x14)
[    0.434260] [<c0012298>] (show_stack) from [<c01a7ab8>] (dump_stack+0x78/0x98)
[    0.434276] [<c01a7ab8>] (dump_stack) from [<c001d968>] (__warn+0xbc/0xec)
[    0.434290] [<c001d968>] (__warn) from [<c001d9cc>] (warn_slowpath_fmt+0x34/0x44)
[    0.434305] [<c001d9cc>] (warn_slowpath_fmt) from [<c07d7ffc>] (regulatory_init+0x80/0x124)
[    0.434321] [<c07d7ffc>] (regulatory_init) from [<c07d7f04>] (cfg80211_init+0x54/0xcc)
[    0.434335] [<c07d7f04>] (cfg80211_init) from [<c0009798>] (do_one_initcall+0xb8/0x174)
[    0.434352] [<c0009798>] (do_one_initcall) from [<c07b7d88>] (kernel_init_freeable+0x120/0x1e4)
[    0.434367] [<c07b7d88>] (kernel_init_freeable) from [<c05a9638>] (kernel_init+0x8/0xf4)
[    0.434382] [<c05a9638>] (kernel_init) from [<c000eeb8>] (ret_from_fork+0x14/0x3c)
[    0.434397] ---[ end trace a52003da2ed29b77 ]---
[    0.434883] clocksource: Switched to clocksource arch_sys_counter
```

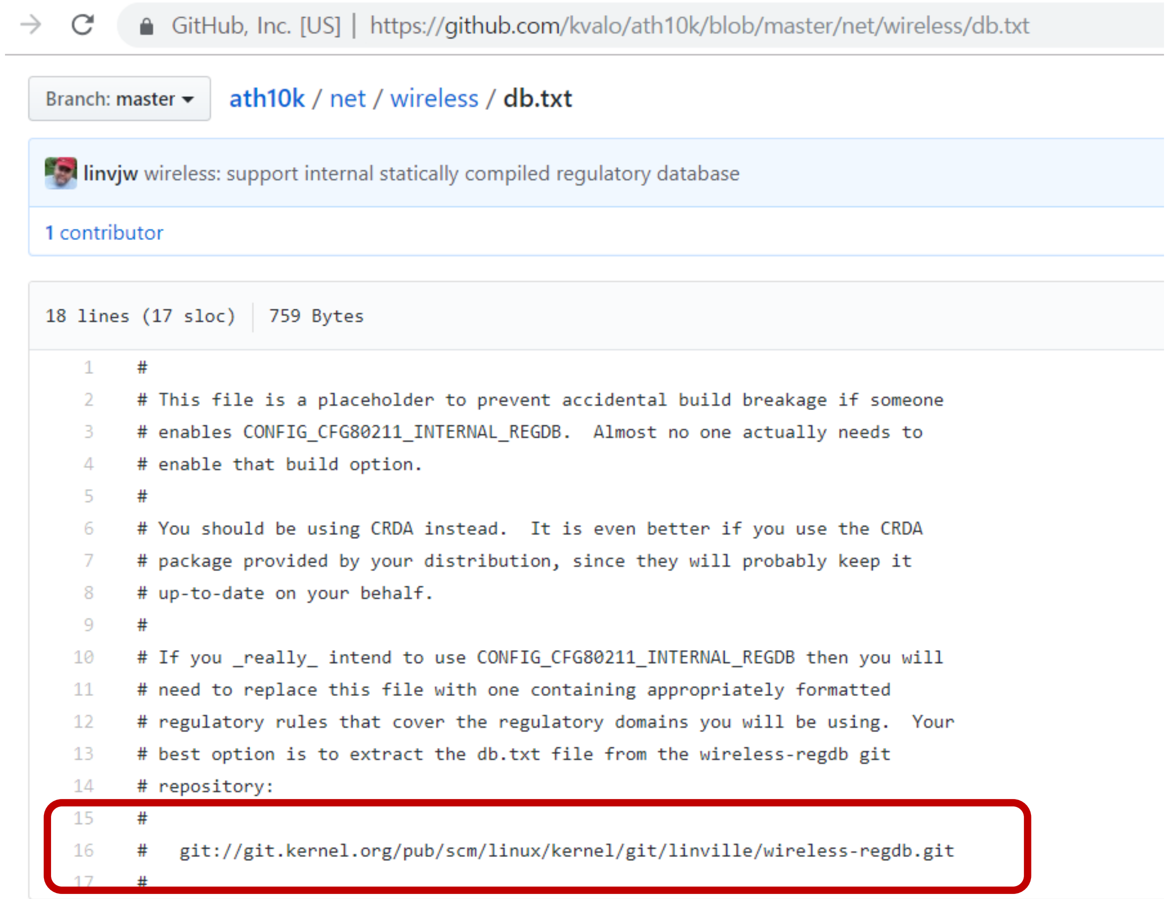
net/wireless/db.txt 파일을 확인해 보니, 내용이 비어 있음.
이 파일의 용도가 뭘까 ?



```
       -> https://ubuntuforums.org/showthread.php?t=2032357
               => 여기에 뭔가 있군 ...
Disclaimer: Regulatory domain is there to keep devices from using the wrong frequencies in certain countrys.
I still have crda restricting me to channels allowed in the US. The problem with is that EEPROM in all my
adapters was not set to US. I bought the Ubiquity adapter from retail in the US, but it still had "the
World" code burned into it and when masked by US domain it only allowed one channel for AP Mode. Two other
adapters I bought on Ebay and their country codes when combined with US settings in crda blocked all
channels from using AP Mode. Regdomain are a real pain but I guess Atheros and the ath9k project are
dedicated to not anger the world's communication regulators.
```

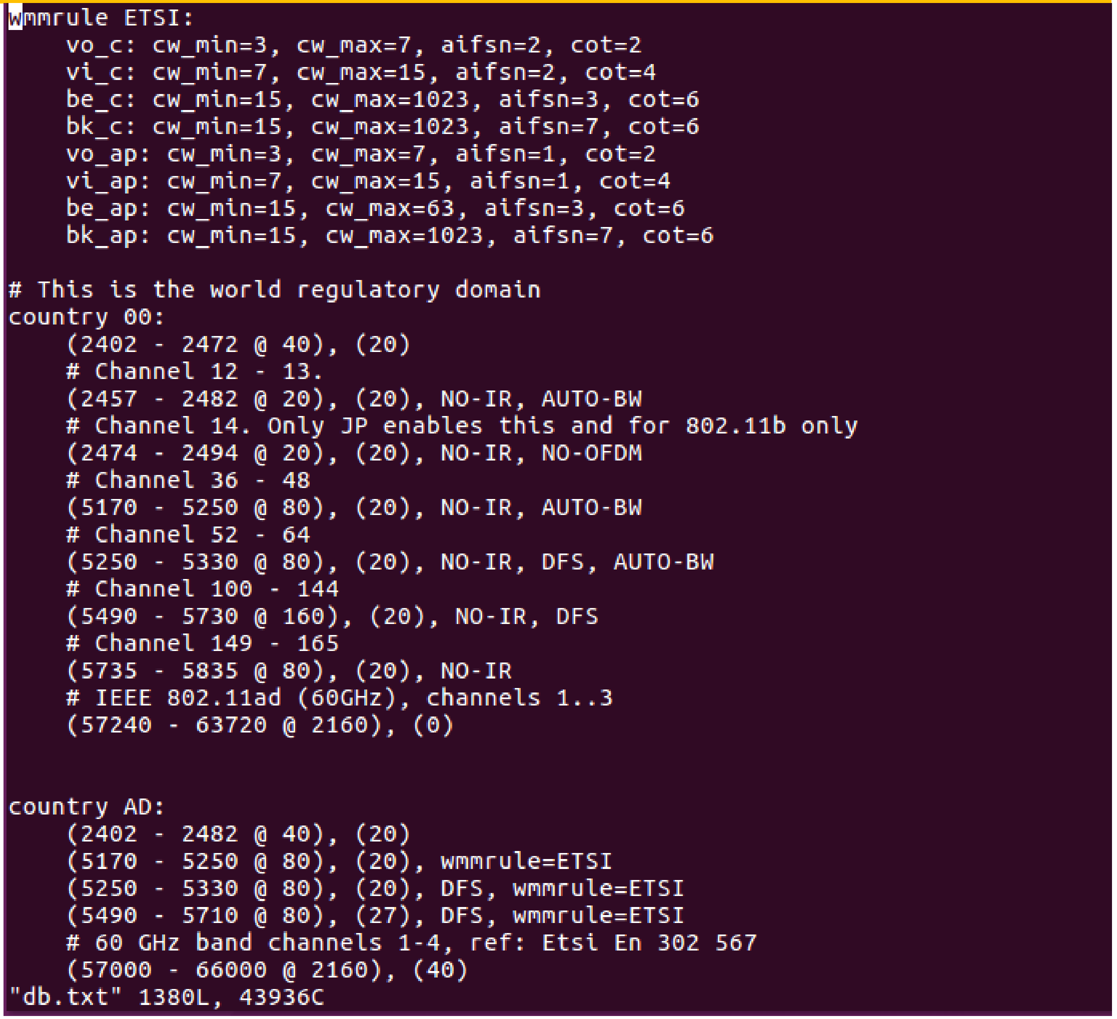
5. 802.11ac 무선랜 구동하기(9) – hostapd 구동(7)

여기서 파일을 download 받아, linux-4.9.44/net/wireless 아래 파일에 복사하자.


GitHub, Inc. [US] | https://github.com/kvalo/ath10k/blob/master/net/wireless/db.txt
Branch: master
ath10k / net / wireless / db.txt
linvjw wireless: support internal statically compiled regulatory database
1 contributor
18 lines (17 sloc) | 759 Bytes

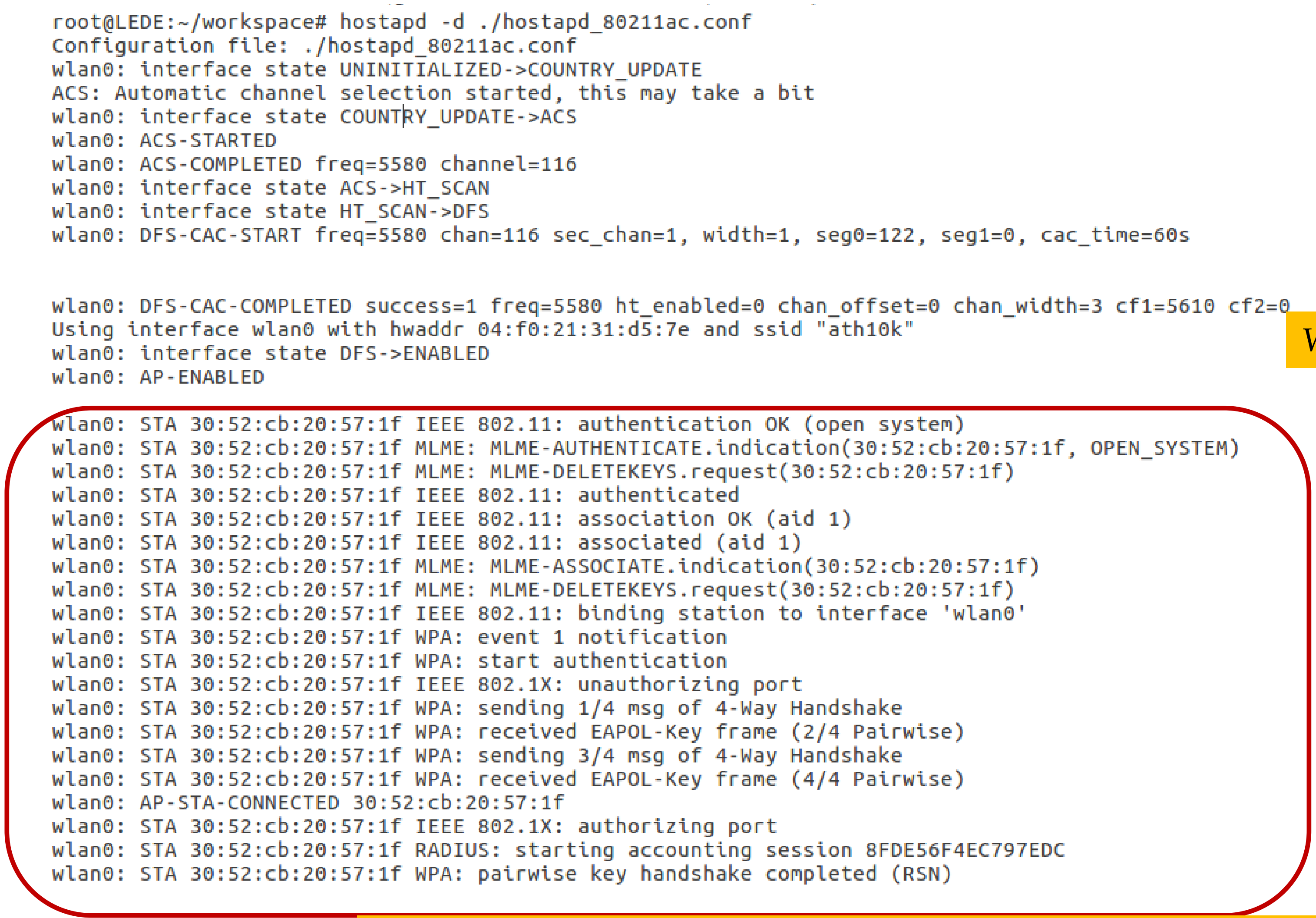
```
#
# This file is a placeholder to prevent accidental build breakage if someone
# enables CONFIG_CFG80211_INTERNAL_REGDB.  Almost no one actually needs to
# enable that build option.
#
# You should be using CRDA instead.  It is even better if you use the CRDA
# package provided by your distribution, since they will probably keep it
# up-to-date on your behalf.
#
# If you _really_ intend to use CONFIG_CFG80211_INTERNAL_REGDB then you will
# need to replace this file with one containing appropriately formatted
# regulatory rules that cover the regulatory domains you will be using.  Your
# best option is to extract the db.txt file from the wireless-regdb git
# repository:
#
#   git://git.kernel.org/pub/scm/linux/kernel/git/linville/wireless-regdb.git
#
```

국가별 채널 주파수를 구분해 주는 내용인 듯 한데 …

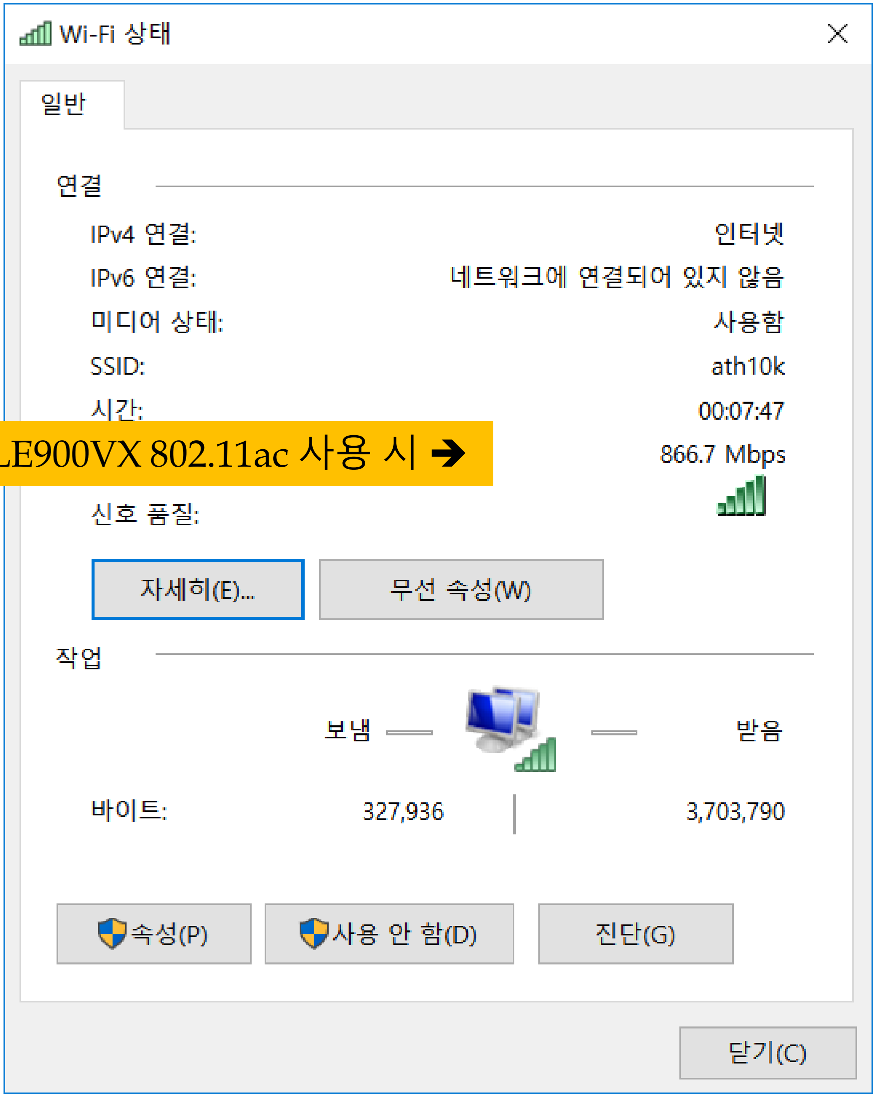


```
wmmrule ETSI:
    vo_c: cw_min=3, cw_max=7, aifsn=2, cot=2
    vi_c: cw_min=7, cw_max=15, aifsn=2, cot=4
    be_c: cw_min=15, cw_max=1023, aifsn=3, cot=6
    bk_c: cw_min=15, cw_max=1023, aifsn=7, cot=6
    vo_ap: cw_min=3, cw_max=7, aifsn=1, cot=2
    vi_ap: cw_min=7, cw_max=15, aifsn=1, cot=4
    be_ap: cw_min=15, cw_max=63, aifsn=3, cot=6
    bk_ap: cw_min=15, cw_max=1023, aifsn=7, cot=6

# This is the world regulatory domain
country 00:
    (2402 - 2472 @ 40), (20)
    # Channel 12 - 13.
    (2457 - 2482 @ 20), (20), NO-IR, AUTO-BW
    # Channel 14. Only JP enables this and for 802.11b only
    (2474 - 2494 @ 20), (20), NO-IR, NO-OFDM
    # Channel 36 - 48
    (5170 - 5250 @ 80), (20), NO-IR, AUTO-BW
    # Channel 52 - 64
    (5250 - 5330 @ 80), (20), NO-IR, DFS, AUTO-BW
    # Channel 100 - 144
    (5490 - 5730 @ 160), (20), NO-IR, DFS
    # Channel 149 - 165
    (5735 - 5835 @ 80), (20), NO-IR
    # IEEE 802.11ad (60GHz), channels 1..3
    (57240 - 63720 @ 2160), (0)


country AD:
    (2402 - 2482 @ 40), (20)
    (5170 - 5250 @ 80), (20), wmmrule=ETSI
    (5250 - 5330 @ 80), (20), DFS, wmmrule=ETSI
    (5490 - 5710 @ 80), (27), DFS, wmmrule=ETSI
    # 60 GHz band channels 1-4, ref: Etsi En 302 567
    (57000 - 66000 @ 2160), (40)
"db.txt" 1380L, 43936C
```

5. 802.11ac 무선랜 구동하기(9) – hostapd 구동(8)



```
root@LEDE:~/workspace# hostapd -d ./hostapd_80211ac.conf
Configuration file: ./hostapd_80211ac.conf
wlan0: interface state UNINITIALIZED->COUNTRY_UPDATE
ACS: Automatic channel selection started, this may take a bit
wlan0: interface state COUNTRY_UPDATE->ACS
wlan0: ACS-STARTED
wlan0: ACS-COMPLETED freq=5580 channel=116
wlan0: interface state ACS->HT_SCAN
wlan0: interface state HT_SCAN->DFS
wlan0: DFS-CAC-START freq=5580 chan=116 sec_chan=1, width=1, seg0=122, seg1=0, cac_time=60s


wlan0: DFS-CAC-COMPLETED success=1 freq=5580 ht_enabled=0 chan_offset=0 chan_width=3 cf1=5610 cf2=0
Using interface wlan0 with hwaddr 04:f0:21:31:d5:7e and ssid "ath10k"
wlan0: interface state DFS->ENABLED
wlan0: AP-ENABLED

wlan0: STA 30:52:cb:20:57:1f IEEE 802.11: authentication OK (open system)
wlan0: STA 30:52:cb:20:57:1f MLME: MLME-AUTHENTICATE.indication(30:52:cb:20:57:1f, OPEN_SYSTEM)
wlan0: STA 30:52:cb:20:57:1f MLME: MLME-DELETEKEYS.request(30:52:cb:20:57:1f)
wlan0: STA 30:52:cb:20:57:1f IEEE 802.11: authenticated
wlan0: STA 30:52:cb:20:57:1f IEEE 802.11: association OK (aid 1)
wlan0: STA 30:52:cb:20:57:1f IEEE 802.11: associated (aid 1)
wlan0: STA 30:52:cb:20:57:1f MLME: MLME-ASSOCIATE.indication(30:52:cb:20:57:1f)
wlan0: STA 30:52:cb:20:57:1f MLME: MLME-DELETEKEYS.request(30:52:cb:20:57:1f)
wlan0: STA 30:52:cb:20:57:1f IEEE 802.11: binding station to interface 'wlan0'
wlan0: STA 30:52:cb:20:57:1f WPA: event 1 notification
wlan0: STA 30:52:cb:20:57:1f WPA: start authentication
wlan0: STA 30:52:cb:20:57:1f IEEE 802.1X: unauthorizing port
wlan0: STA 30:52:cb:20:57:1f WPA: sending 1/4 msg of 4-Way Handshake
wlan0: STA 30:52:cb:20:57:1f WPA: received EAPOL-Key frame (2/4 Pairwise)
wlan0: STA 30:52:cb:20:57:1f WPA: sending 3/4 msg of 4-Way Handshake
wlan0: STA 30:52:cb:20:57:1f WPA: received EAPOL-Key frame (4/4 Pairwise)
wlan0: AP-STA-CONNECTED 30:52:cb:20:57:1f
wlan0: STA 30:52:cb:20:57:1f IEEE 802.1X: authorizing port
wlan0: STA 30:52:cb:20:57:1f RADIUS: starting accounting session 8FDE56F4EC797EDC
wlan0: STA 30:52:cb:20:57:1f WPA: pairwise key handshake completed (RSN)
```



Wi-Fi 상태

일반

연결

IPv4 연결:	인터넷
IPv6 연결:	네트워크에 연결되어 있지 않음
미디어 상태:	사용함
SSID:	ath10k
시간:	00:07:47
	866.7 Mbps
신호 품질:	

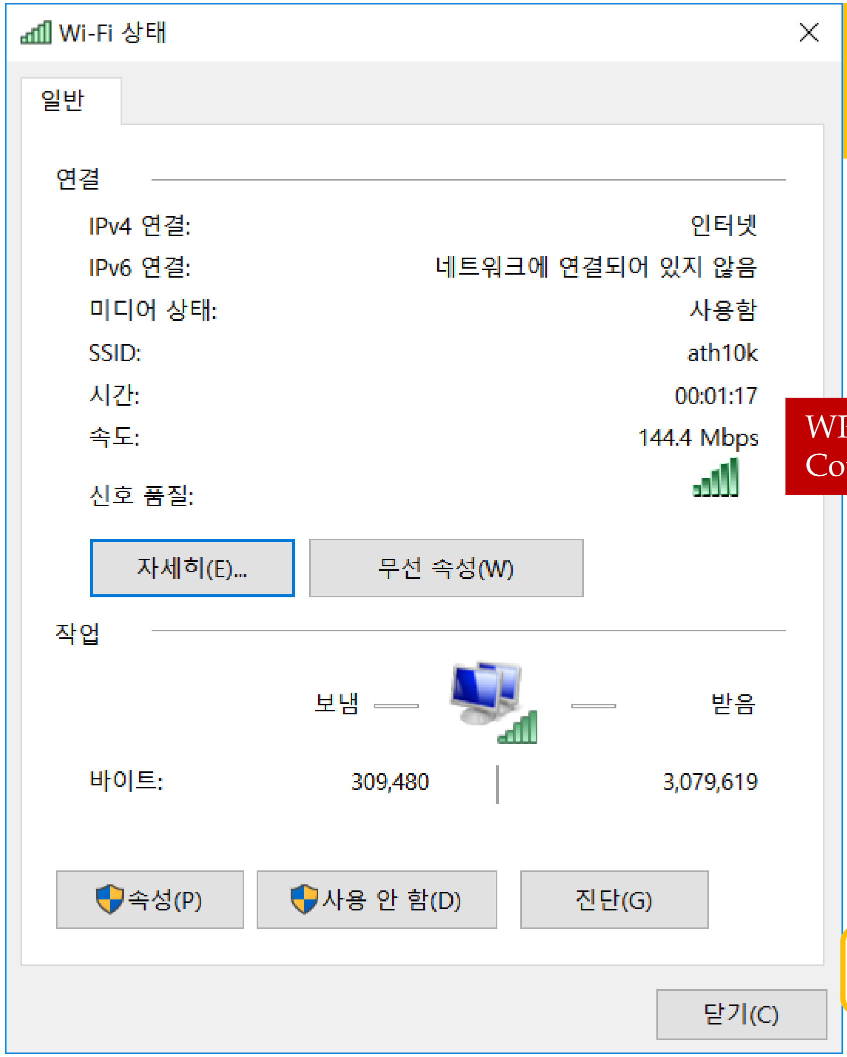
작업

	보냄	받음
바이트:	327,936	3,703,790

WLE900VX 802.11ac 사용 시 ➔

드디어 동작한다 ☺ 근데, 동작까지의 과정이 매우 느리다. 부팅 후 몇분을 기다려야 한다.[개선 Point]

5. 802.11ac 무선랜 구동하기(9) – hostapd 구동(9)



Wi-Fi 상태

연결

IPv4 연결: 인터넷

IPv6 연결: 네트워크에 연결되어 있지 않음

미디어 상태: 사용함

SSID: ath10k

시간: 00:01:17

속도: 144.4 Mbps

신호 품질:

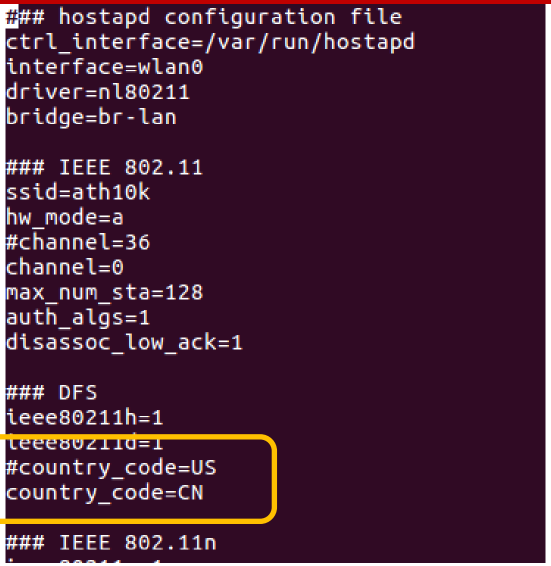
작업

보냄 받음

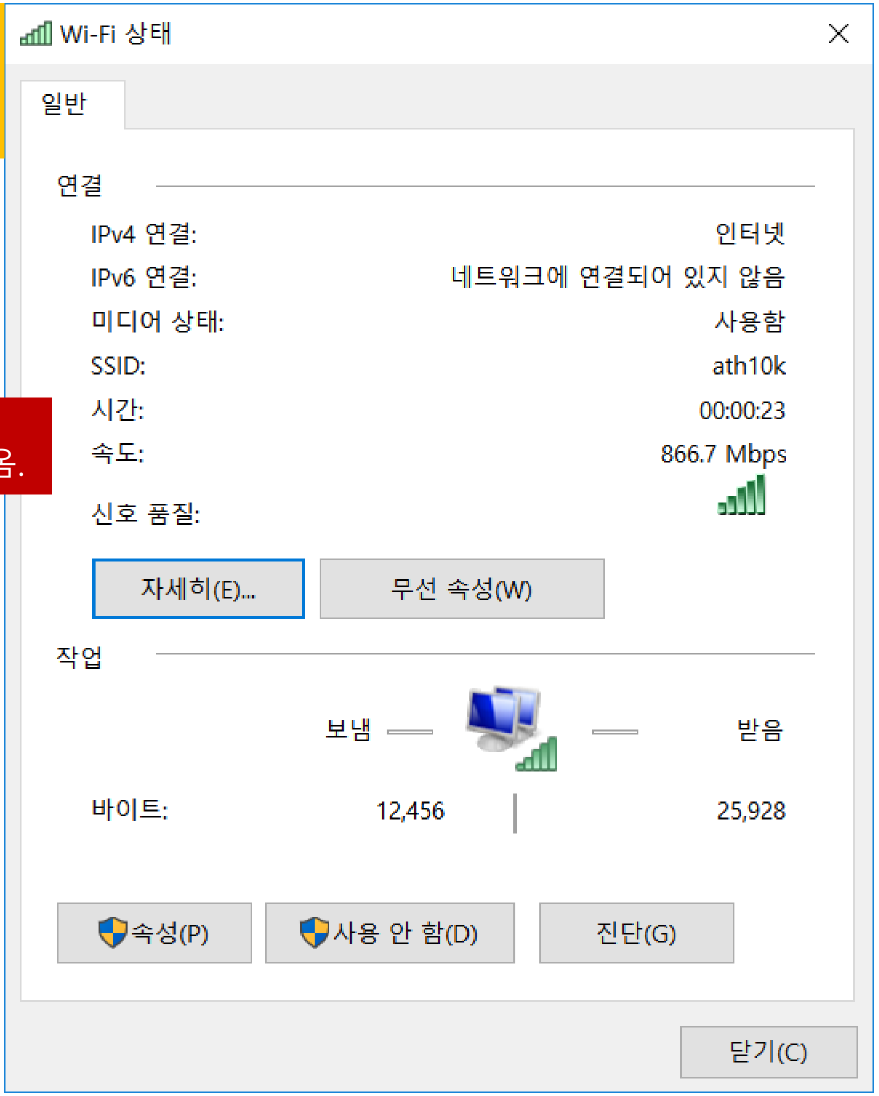
바이트: 309,480 3,079,619

←WPEA-352ACN 사용시
어라, 왜 이리 속도가 안 나올까 ?
Firmware 문제인가 ?! [Tuning Point]

WPEA-352ACN은 중국에서 구입한 것임.
Country를 US => CN으로 교체 후, 정상 속도 나옴.



```
### hostapd configuration file
ctrl_interface=/var/run/hostapd
interface=wlan0
driver=nl80211
bridge=br-lan

### IEEE 802.11
ssid=ath10k
hw_mode=a
#channel=36
channel=0
max_num_sta=128
auth_algs=1
disassoc_low_ack=1

### DFS
ieee80211h=1
ieee80211d=1
#country_code=US
country_code=CN

### IEEE 802.11n
```



Wi-Fi 상태

연결

IPv4 연결: 인터넷

IPv6 연결: 네트워크에 연결되어 있지 않음

미디어 상태: 사용함

SSID: ath10k

시간: 00:00:23

속도: 866.7 Mbps

신호 품질:

작업

보냄 받음

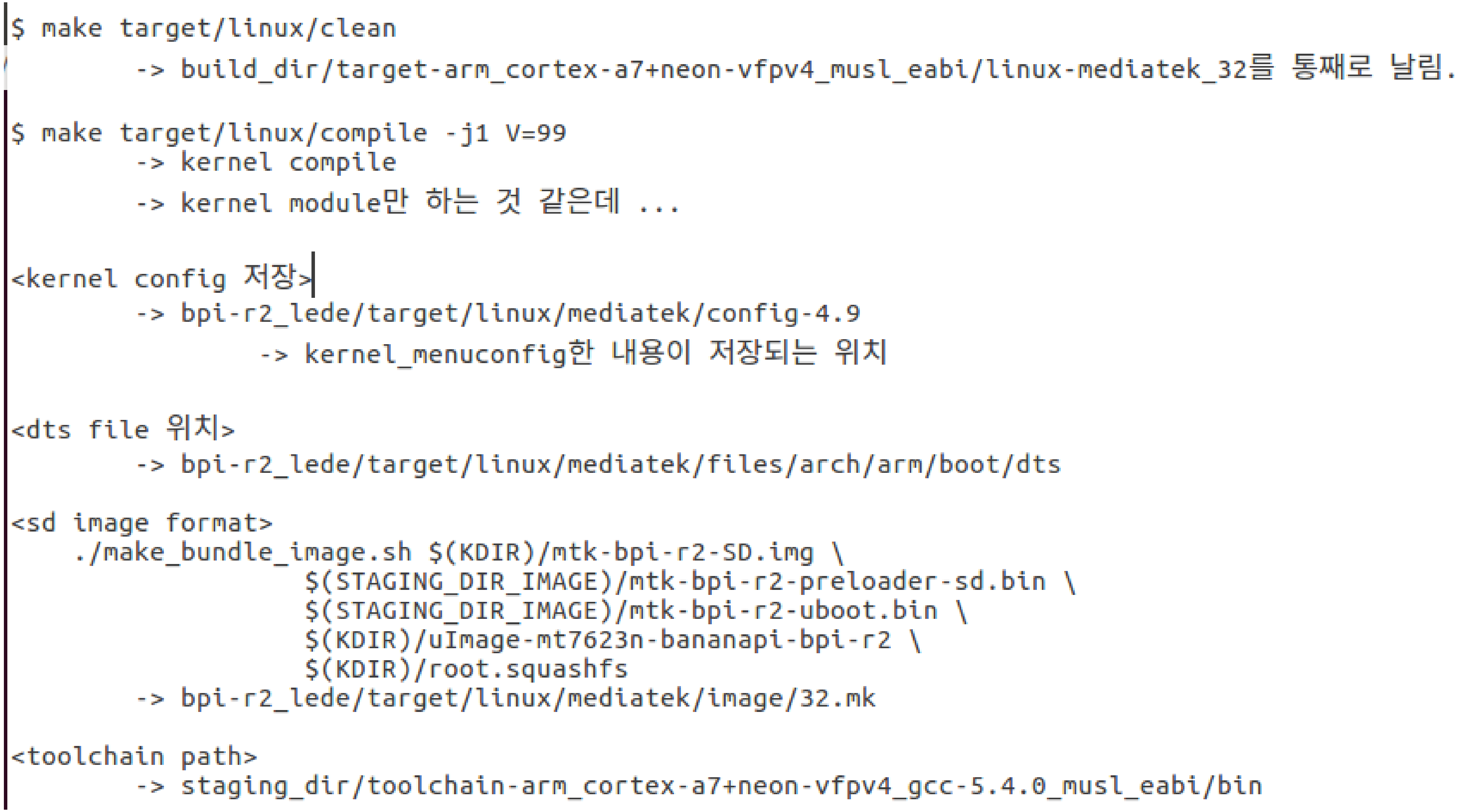
바이트: 12,456 25,928

5. 802.11ac 무선랜 구동하기(10)

5. 802.11ac 무선랜 구동하기(11) – TODO

- 부팅 시 802.11ac WLAN driver의 인식 속도가 매우 느리다(2~3분 걸림). 이 점을 개선해야 한다.
- 802.11n(예: wlan0), 802.11ac(예: wlan1) 두개의 wlan interface를 hostapd로 함께 구동하는 방법 테스트해 보아야 한다.

6. OpenWrt/LEDE 참고 사항(1)



```
$ make target/linux/clean
        -> build_dir/target-arm_cortex-a7+neon-vfpv4_musl_eabi/linux-mediatek_32를 통째로 날림.

$ make target/linux/compile -j1 V=99
        -> kernel compile
        -> kernel module만 하는 것 같은데 ...

<kernel config 저장>
        -> bpi-r2_lede/target/linux/mediatek/config-4.9
                -> kernel_menuconfig한 내용이 저장되는 위치

<dts file 위치>
        -> bpi-r2_lede/target/linux/mediatek/files/arch/arm/boot/dts

<sd image format>
    ./make_bundle_image.sh $(KDIR)/mtk-bpi-r2-SD.img \
                   $(STAGING_DIR_IMAGE)/mtk-bpi-r2-preloader-sd.bin \
                   $(STAGING_DIR_IMAGE)/mtk-bpi-r2-uboot.bin \
                   $(KDIR)/uImage-mt7623n-bananapi-bpi-r2 \
                   $(KDIR)/root.squashfs
        -> bpi-r2_lede/target/linux/mediatek/image/32.mk

<toolchain path>
        -> staging_dir/toolchain-arm_cortex-a7+neon-vfpv4_gcc-5.4.0_musl_eabi/bin
```

6. OpenWrt/LEDE 참고 사항(2)

- 1) http://www.banana-pi.org/r2.html
- 2) http://wiki.banana-pi.org/Getting_Started_with_R2
- 3) http://www.fw-web.de/dokuwiki/doku.php?id=en:bpi-r2:start
- 4) https://wireless.wiki.kernel.org/en/users/drivers/ath10k