Xilinx Zynq FPGA, TI DSP, MCU 프로그래밍 및 회로 설계 전문가 과정

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com

AM5728 Based Wi-Fi AP Setting
(How to use WL1837 with AM5728)

작업 이전에 Wi-Fi Device Driver 와 관련한 작업이 모두 완료되어 있어야 한다.

초기에 /usr/sbin/wlconf 가 잡혀있는지 확인한다.
만약 없다면 디렉토리를 만들고 /usr/bin/wlconf 에 있는 모든 내용을 복사한다.

```
root@am57xx-evm:~# cd /usr/sbin/wlconf
-sh: cd: /usr/sbin/wlconf: No such file or directory
root@am57xx-evm:~# mkdir /usr/sbin/wlconf
root@am57xx-evm:~# cp -r /usr/bin/wlconf/
README                  example.conf            wl18xx-conf-default.bin
configure-device.sh     example.ini             wlconf
default.conf            official_inis/
dictionary.txt          struct.bin
root@am57xx-evm:~# cp -r /usr/bin/wlconf/* /usr/sbin/wlconf
root@am57xx-evm:~# ls
root@am57xx-evm:~# cd /usr/sbin/wlconf
root@am57xx-evm:/usr/sbin/wlconf# ls
README                  dictionary.txt          official_inis           wlconf
configure-device.sh     example.conf            struct.bin
default.conf            example.ini             wl18xx-conf-default.bin
root@am57xx-evm:/usr/sbin/wlconf#
```

안테나를 설정해준다.

```
root@am57xx-evm:/usr/sbin/wlconf# ./configure-device.sh

Please provide the following information.

Are you using a TI module? [y/n] : y
What is the chip flavor? [1801/1805/1807/1831/1835/1837 or 0 for unknown] : 1837
Should Japanese standards be applied? [y/n] : n
How many 2.4GHz antennas are fitted? [1/2] : 1
How many 5GHz antennas are fitted? [0/1/2] : 1
Should SISO40 support be applied? [y/n] : n

The device has been successfully configured.
TI Module: y
Chip Flavor: 1837
Number of 2.4GHz Antennas Fitted: 1
Number of 5GHz Antennas Fitted: 1
Diversity Support: y
SISO40 Support: n
Japanese Standards Applied: n

root@am57xx-evm:/usr/sbin/wlconf#
```

집이나 학원 공유기로 접속해본다.

```
root@am57xx-evm:/usr/sbin/wlconf# cd /usr/share/wl18xx/
root@am57xx-evm:/usr/share/wl18xx# ./load_wlcore.sh
root@am57xx-evm:/usr/share/wl18xx# ./sta_start.sh
root@am57xx-evm:/usr/share/wl18xx# Successfully initialized wpa_supplicant
Could not read interface p2p-dev-wlan0 flags: No such device
p2p-dev-wlan0: CTRL-EVENT-REGDOM-CHANGE init=USER type=COUNTRY alpha2=US

root@am57xx-evm:/usr/share/wl18xx# ./sta_connect-ex.sh SK_WiFiB01F
netid=0
========================
OK
OK
OK
root@am57xx-evm:/usr/share/wl18xx# udhcpc -i wlan0
udhcpc (v1.24.1) started
Sending discover...
Sending discover...
Sending discover...
^C
root@am57xx-evm:/usr/share/wl18xx#
```

비밀번호가 있다면 현재 이 방법으로 해야하고 없다면 위의 방법으로 접속된다.

```
root@am57xx-evm:/usr/share/wl18xx# ./sta_connect-ex.sh SK_wifi20_5 WPA-PSK [illegible]
netid=1
======================
OK
OK
OK
OK
root@am57xx-evm:/usr/share/wl18xx# wlan0: SME: Trying to authenticate with 00:30:0d:bf:b0:1e (SSI
D='SK_WiFiB01F' freq=2417 MHz)
wlan0: Trying to associate with 00:30:0d:bf:b0:1e (SSID='SK_WiFiB01F' freq=2417 MHz)
wlan0: Associated with 00:30:0d:bf:b0:1e
wlan0: WPA: Key negotiation completed with 00:30:0d:bf:b0:1e [PTK=CCMP GTK=TKIP]
wlan0: CTRL-EVENT-CONNECTED - Connection to 00:30:0d:bf:b0:1e completed [id=1 id_str=]

root@am57xx-evm:/usr/share/wl18xx# udhcpc -i wlan0
udhcpc (v1.24.1) started
Sending discover...
Sending select for 192.168.25.40...
Lease of 192.168.25.40 obtained, lease time 3600
/etc/udhcpc.d/50default: Adding DNS 210.220.163.82
/etc/udhcpc.d/50default: Adding DNS 219.250.36.130
root@am57xx-evm:/usr/share/wl18xx#
```

이와 같이 무선랜을 잡아오는 것을 볼 수 있다.
그러나 이것은 AP 가 되지 못하는 방식이다.

```
eth1      Link encap:Ethernet  HWaddr A0:F6:FD:AB:9E:AF
          UP 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)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1%132400/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:572 errors:0 dropped:0 overruns:0 frame:0
          TX packets:572 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:297947 (290.9 KiB)  TX bytes:297947 (290.9 KiB)

wlan0     Link encap:Ethernet  HWaddr 7C:EC:79:C8:23:73
          inet addr:192.168.25.40  Bcast:192.168.25.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:188 errors:0 dropped:0 overruns:0 frame:0
          TX packets:242 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:19487 (19.0 KiB)  TX bytes:36276 (35.4 KiB)
```

공유기에 ping 을 때려보면 아주 잘 가는 것을 볼 수 있다.

```
root@am57xx-evm:/usr/share/wl18xx# ping 192.168.25.1
PING 192.168.25.1 (192.168.25.1): 56 data bytes
64 bytes from 192.168.25.1: seq=0 ttl=64 time=0.931 ms
64 bytes from 192.168.25.1: seq=1 ttl=64 time=0.754 ms
64 bytes from 192.168.25.1: seq=2 ttl=64 time=0.749 ms
64 bytes from 192.168.25.1: seq=3 ttl=64 time=0.762 ms
64 bytes from 192.168.25.1: seq=4 ttl=64 time=0.762 ms
64 bytes from 192.168.25.1: seq=5 ttl=64 time=0.766 ms
64 bytes from 192.168.25.1: seq=6 ttl=64 time=0.775 ms
64 bytes from 192.168.25.1: seq=7 ttl=64 time=0.826 ms
64 bytes from 192.168.25.1: seq=8 ttl=64 time=0.767 ms
64 bytes from 192.168.25.1: seq=9 ttl=64 time=0.759 ms
64 bytes from 192.168.25.1: seq=10 ttl=64 time=0.759 ms
64 bytes from 192.168.25.1: seq=11 ttl=64 time=0.754 ms
64 bytes from 192.168.25.1: seq=12 ttl=64 time=0.752 ms
64 bytes from 192.168.25.1: seq=13 ttl=64 time=0.756 ms
64 bytes from 192.168.25.1: seq=14 ttl=64 time=0.773 ms
64 bytes from 192.168.25.1: seq=15 ttl=64 time=0.759 ms
64 bytes from 192.168.25.1: seq=16 ttl=64 time=0.768 ms
```

AP 가 되기 위해서는 해당 디렉토리 내부에 있는 hostapd.conf 를 살펴봐야한다.

```
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#       http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# AP netdevice name (without 'ap' postfix, i.e., wlan0 uses wlan0ap for
# management frames); ath0 for madwifi
interface=wlan1
```

```
# Driver interface type (hostap/wired/madwifi/test/none/nl80211/bsd);
# default: hostap). nl80211 is used with all Linux mac80211 drivers.
# Use driver=none if building hostapd as a standalone RADIUS server that does
# not control any wireless/wired driver.
driver=nl80211

# hostapd event logger configuration
#
# Two output method: syslog and stdout (only usable if not forking to
# background).
#
# Module bitfield (ORed bitfield of modules that will be logged; -1 = all
# modules):
```

```
##### IEEE 802.11 related configuration #######################################

# SSID to be used in IEEE 802.11 management frames
ssid=SitaraAP
# Alternative formats for configuring SSID
# (double quoted string, hexdump, printf-escaped string)
#ssid2="test"
#ssid2=74657374
#ssid2=P"hello\nthere"
```

```
# Enable IEEE 802.11h. This enables radar detection and DFS support if
# available. DFS support is required on outdoor 5 GHz channels in most countries
# of the world. This can be used only with ieee80211d=1.
# (default: 0 = disabled)
ieee80211h=1

# Operation mode (a = IEEE 802.11a, b = IEEE 802.11b, g = IEEE 802.11g,
# Default: IEEE 802.11b
hw_mode=g

# Channel number (IEEE 802.11)
# (default: 0, i.e., not set)
# Please note that some drivers do not use this value from hostapd and the
# channel will need to be configured separately with iwconfig.
# When set to 0, automatic channel selection will be engaged. A channel
# will be selected from the desired hw_mode.
channel=11
```

```
# Station MAC address -based authentication
# Please note that this kind of access control requires a driver that uses
# hostapd to take care of management frame processing and as such, this can be
# used with driver=hostap or driver=nl80211, but not with driver=madwifi.
# 0 = accept unless in deny list
# 1 = deny unless in accept list
# 2 = use external RADIUS server (accept/deny lists are searched first)
macaddr_acl=0

# Accept/deny lists are read from separate files (containing list of
# MAC addresses, one per line). Use absolute path name to make sure that the
# files can be read on SIGHUP configuration reloads.
#accept_mac_file=/etc/hostapd.accept
#deny_mac_file=/etc/hostapd.deny
```

```
# IEEE 802.11 specifies two authentication algorithms. hostapd can be
# configured to allow both of these or only one. Open system authentication
# should be used with IEEE 802.1X.
# Bit fields of allowed authentication algorithms:
# bit 0 = Open System Authentication
# bit 1 = Shared Key Authentication (requires WEP)
auth_algs=3
```

```
# Send empty SSID in beacons and ignore probe request frames that do not
# specify full SSID, i.e., require stations to know SSID.
# default: disabled (0)
# 1 = send empty (length=0) SSID in beacon and ignore probe request for
#     broadcast SSID
# 2 = clear SSID (ASCII 0), but keep the original length (this may be required
#     with some clients that do not support empty SSID) and ignore probe
#     requests for broadcast SSID
ignore_broadcast_ssid=0

# Additional vendor specfic elements for Beacon and Probe Response frames
# This parameter can be used to add additional vendor specific element(s) into
# the end of the Beacon and Probe Response frames. The format for these
# element(s) is a hexdump of the raw information elements (id+len+payload for
# one or more elements)
#vendor_elements=dd0411223301
```

AP 에 비밀번호를 지정하기 위해서는 이 부분을 살려야 한다.
우리의 Controller 는 이 부분을 살려야 한다.

```
# Enable WPA. Setting this variable configures the AP to require WPA (either
# WPA-PSK or WPA-RADIUS/EAP based on other configuration). For WPA-PSK, either
# wpa_psk or wpa_passphrase must be set and wpa_key_mgmt must include WPA-PSK.
# Instead of wpa_psk / wpa_passphrase, wpa_psk_radius might suffice.
# For WPA-RADIUS/EAP, ieee8021x must be set (but without dynamic WEP keys),
# RADIUS authentication server must be configured, and WPA-EAP must be included
# in wpa_key_mgmt.
# This field is a bit field that can be used to enable WPA (IEEE 802.11i/D3.0)
# and/or WPA2 (full IEEE 802.11i/RSN):
# bit0 = WPA
# bit1 = IEEE 802.11i/RSN (WPA2) (dot11RSNAEnabled)
#wpa=1
```

이 부분에 비밀번호등을 설정하게 된다.

```
# WPA pre-shared keys for WPA-PSK. This can be either entered as a 256-bit
# secret in hex format (64 hex digits), wpa_psk, or as an ASCII passphrase
# (8..63 characters) that will be converted to PSK. This conversion uses SSID
# so the PSK changes when ASCII passphrase is used and the SSID is changed.
# wpa_psk (dot11RSNAConfigPSKValue)
# wpa_passphrase (dot11RSNAConfigPSKPassPhrase)
#wpa_psk=0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef
#wpa_passphrase=secret passphrase
```

```
# Optionally, WPA passphrase can be received from RADIUS authentication server
# This requires macaddr_acl to be set to 2 (RADIUS)
# 0 = disabled (default)
# 1 = optional; use default passphrase/psk if RADIUS server does not include
#       Tunnel-Password
# 2 = required; reject authentication if RADIUS server does not include
#       Tunnel-Password
#wpa_psk_radius=0

# Set of accepted key management algorithms (WPA-PSK, WPA-EAP, or both). The
# entries are separated with a space. WPA-PSK-SHA256 and WPA-EAP-SHA256 can be
# added to enable SHA256-based stronger algorithms.
# (dot11RSNAConfigAuthenticationSuitesTable)
#wpa_key_mgmt=WPA-PSK WPA-EAP
```

```
# Set of accepted cipher suites (encryption algorithms) for pairwise keys
# (unicast packets). This is a space separated list of algorithms:
# CCMP = AES in Counter mode with CBC-MAC [RFC 3610, IEEE 802.11i/D7.0]
# TKIP = Temporal Key Integrity Protocol [IEEE 802.11i/D7.0]
# Group cipher suite (encryption algorithm for broadcast and multicast frames)
# is automatically selected based on this configuration. If only CCMP is
# allowed as the pairwise cipher, group cipher will also be CCMP. Otherwise,
# TKIP will be used as the group cipher.
# (dot11RSNAConfigPairwiseCiphersTable)
# Pairwise cipher for WPA (v1) (default: TKIP)
#wpa_pairwise=TKIP CCMP
# Pairwise cipher for RSN/WPA2 (default: use wpa_pairwise value)
#rsn_pairwise=CCMP
# Time interval for rekeying GTK (broadcast/multicast encryption keys) in
# seconds. (dot11RSNAConfigGroupRekeyTime)
wpa_group_rekey=0
# Rekey GTK when any STA that possesses the current GTK is leaving the BSS.
# (dot11RSNAConfigGroupRekeyStrict)
#wpa_strict_rekey=1
# Time interval for rekeying GMK (master key used internally to generate GTKs
# (in seconds).
wpa_gmk_rekey=0
```

AP 를 활성화시키고자 하였지만 스크립트가 꼬여있다.

```
root@am57xx-evm:/usr/share/wl18xx# vi hostapd.conf
root@am57xx-evm:/usr/share/wl18xx# ./ap_start.sh
adding wlan1 interface
./ap_start.sh: line 43: /usr/local/bin/hostapd: No such file or directory

root@am57xx-evm:/usr/share/wl18xx#
```

이 부분이 잘못된 것이다.

```
### start a hostapd interface, if not present
if [ ! -r $HOSTAPD_PROC ]
then
 $HOSTAPD_BIN_DIR/hostapd $HOSTAPD_CONF &
 sleep 1
fi
```

아래와 같이 수정한다.

```
HOSTAPD_CONF=/usr/share/wl18xx/hostapd.conf
HOSTAPD_BIN_DIR=/usr/local/bin
IP_ADDR=192.168.43.1
```

```
HOSTAPD_CONF=/usr/share/wl18xx/hostapd.conf
HOSTAPD_BIN_DIR=/usr/sbin
IP_ADDR=192.168.43.1
```

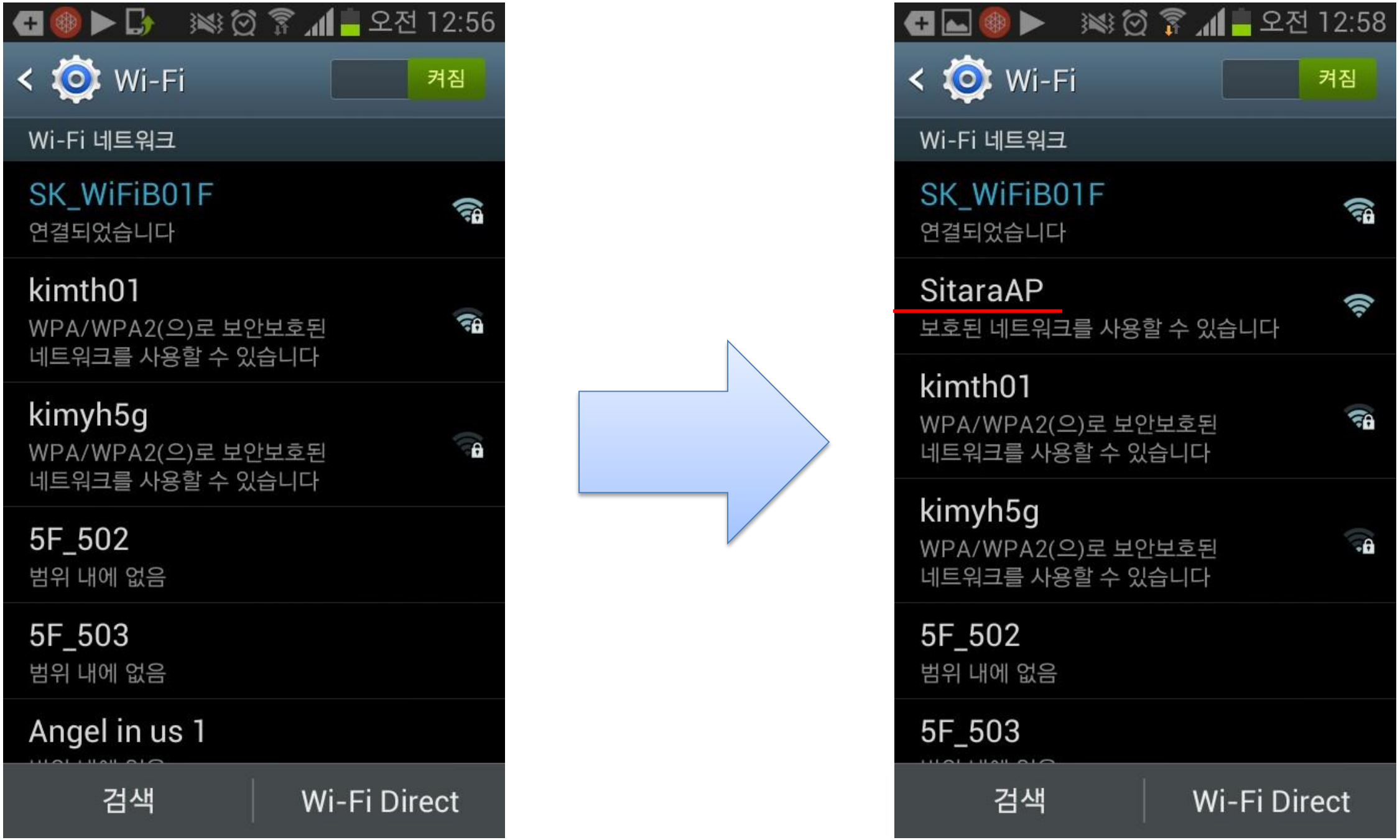
```
root@am57xx-evm:/usr/share/wl18xx# ./ap_start.sh
Configuration file: /usr/share/wl18xx/hostapd.conf
wlan1: interface state UNINITIALIZED->COUNTRY_UPDATE
Using interface wlan1 with hwaddr 7e:ec:79:c8:23:73 and ssid "SitaraAP"
wlan1: interface state COUNTRY_UPDATE->ENABLED
wlan1: AP-ENABLED

root@am57xx-evm:/usr/share/wl18xx#
```

```
wlan0     Link encap:Ethernet  HWaddr 7C:EC:79:C8:23:73
          inet addr:192.168.25.40  Bcast:192.168.25.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:283 errors:0 dropped:0 overruns:0 frame:0
          TX packets:270 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:31553 (30.8 KiB)  TX bytes:40130 (39.1 KiB)

wlan1     Link encap:Ethernet  HWaddr 7E:EC:79:C8:23:73
          inet addr:192.168.43.1  Bcast:192.168.43.255  Mask:255.255.255.0
          inet6 addr: fe80::7cec:79ff:fec8:2373%132112/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:49 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:9150 (8.9 KiB)

root@am57xx-evm:/usr/share/wl18xx#
```


오전 12:56
Wi-Fi
켜짐
Wi-Fi 네트워크
SK_WiFiB01F
연결되었습니다
kimth01
WPA/WPA2(으)로 보안보호된
네트워크를 사용할 수 있습니다
kimyh5g
WPA/WPA2(으)로 보안보호된
네트워크를 사용할 수 있습니다
5F_502
범위 내에 없음
5F_503
범위 내에 없음
Angel in us 1
검색
Wi-Fi Direct
오전 12:58
Wi-Fi
켜짐
Wi-Fi 네트워크
SK_WiFiB01F
연결되었습니다
SitaraAP
보호된 네트워크를 사용할 수 있습니다
kimth01
WPA/WPA2(으)로 보안보호된
네트워크를 사용할 수 있습니다
kimyh5g
WPA/WPA2(으)로 보안보호된
네트워크를 사용할 수 있습니다
5F_502
범위 내에 없음
5F_503
검색
Wi-Fi Direct