

# TI MCU, DSP 및 Xilinx FPGA 프로그래밍 전문가 과정

Innova Lee(이상훈)  
[gcccompil3r@gmail.com](mailto:gcccompil3r@gmail.com)

# **How to make Secure AP**

이번에는 Wi-Fi AP 에 무차별 접속하는 사람들에 대항하기 위한 방법을 살펴보도록 한다.

[https://github.com/KOITT2/RC\\_Car/tree/master/dsp\\_proj/wl18xx\\_config](https://github.com/KOITT2/RC_Car/tree/master/dsp_proj/wl18xx_config)

위 링크에 있는 두 개의 Script 를 적절하게 활용해야 한다.

추가적으로 AP 를 활성화 하는데 있어서 IP 주소가 중복되면 충돌이 발생할 가능성이 존재한다.

그러므로 ap\_start.sh 부분의 IP Configuration 부분도 수정해줄 필요가 있다.

필자가 해당 작업을 진행하는데 참고한 링크들은 아래와 같다.

Secure AP 설정을 진행하는데 읽어 볼 필요는 없고

WL1837 Wi-Fi & Bluetooth Dual Band 모듈의 동작 과정을 좀 더 엄밀하게 살펴보고 싶을 경우 참고하도록 한다.

(추가적으로 Wi-Fi 를 사용하면서 Bluetooth 도 사용이 가능하기 때문에 추가 기능을 원한다면 살펴보길 바란다)

<http://processors.wiki.ti.com/index.php/WL18xx>

[http://processors.wiki.ti.com/index.php/WiLink8\\_Linux\\_Getting\\_Started\\_Guide](http://processors.wiki.ti.com/index.php/WiLink8_Linux_Getting_Started_Guide)

[http://processors.wiki.ti.com/index.php/WiLink8\\_WLAN\\_Demo:\\_Linux](http://processors.wiki.ti.com/index.php/WiLink8_WLAN_Demo:_Linux)

[http://processors.wiki.ti.com/index.php/WL18xx\\_Audio\\_Demo\\_Kit\\_QuickStart\\_Guide](http://processors.wiki.ti.com/index.php/WL18xx_Audio_Demo_Kit_QuickStart_Guide)

[http://processors.wiki.ti.com/index.php/WiLink8\\_WLAN\\_Advanced\\_Demos:\\_Linux](http://processors.wiki.ti.com/index.php/WiLink8_WLAN_Advanced_Demos:_Linux)

<http://www.ti.com/lit/an/swra503/swra503.pdf>

<http://www.ti.com/lit/ug/swru423a/swru423a.pdf>

<http://www.ti.com/lit/ug/swru437a/swru437a.pdf>

<http://www.ti.com/lit/ug/swru382a/swru382a.pdf>

<http://www.ti.com/lit/wp/swry024/swry024.pdf>

<http://www.ti.com/lit/wp/swry018/swry018.pdf>

<http://www.ti.com/lit/wp/spry256/spry256.pdf>

[http://processors.wiki.ti.com/index.php/WL18xx\\_Driver\\_Debug](http://processors.wiki.ti.com/index.php/WL18xx_Driver_Debug)

[http://processors.wiki.ti.com/index.php/WL18xx\\_Bluetopia\\_PM\\_Bluetooth\\_RF\\_Testing](http://processors.wiki.ti.com/index.php/WL18xx_Bluetopia_PM_Bluetooth_RF_Testing)

[http://processors.wiki.ti.com/index.php/WL18xx\\_Testing\\_Misc\\_Commands](http://processors.wiki.ti.com/index.php/WL18xx_Testing_Misc_Commands)

<https://www.linux.com/learn/create-secure-linux-based-wireless-access-point>

sdr@sdr-Z20NH-AS51B5U: ~

#!/bin/sh

지금 보는 Shell Script 는 /usr/share/wl18xx/ap\_start.sh 에 있다.

##### variables #####

WLAN=wlan1

WLAN2=wlan2

HOSTAPD\_PROC=/var/run/hostapd

HOSTAPD\_CONF=/usr/share/wl18xx/hostapd.conf

HOSTAPD\_BIN\_DIR=/usr/sbin

IP\_ADDR=192.168.43.1

IP\_ADDR2=192.168.33.1

DHCP\_CONF=udhcpd.conf

DHCP\_CONF2=udhcpd2.conf

DHCP\_CONF\_PROC=u[d]hcpd.conf

DHCP\_CONF\_PROC2=u[d]hcpd2.conf

##### body #####

### check for configuration file

if [ ! -f \$HOSTAPD\_CONF ]; then

if [ ! -f /etc/hostapd.conf ]

then

echo "error - no default hostapd.conf file"

exit 1

fi

cp /etc/hostapd.conf \$HOSTAPD\_CONF

chmod 777 \$HOSTAPD\_CONF

이 부분의 IP 값을 바꿔주면 AP 의 IP 를 바꿔 줄 수 있다.

192.168.43.1,

192.168.42.1,

192.168.41.1

위와 같이 서로 충돌되지 않게 한다.

또한 관례상 라우터(AP)의 주소는 1 로 고정하는 것이 좋다.



```
(7) Left Winker
buf = 13 1
(13) Specified Angle or PWM Duty(Servo)
buf = 13 4
(13) Specified Angle or PWM Duty(Servo)
buf = 13 2
(13) Specified Angle or PWM Duty(Servo)
buf = 7 0
```

이 부분은 팁인데 scp 명령어는 네트워크 상에서 cp(복사) 명령어를 사용하는 방법이다.  
아래 빨간 박스는 현재 DSP 상에서 wpa\_supplicant.conf 를  
192.168.0.52 의 계정 sdr 상의 디렉토리 /home/sdr/proj/RC\_Car/dsp\_proj/wl18xx\_config 로 복사하고자 함을 의미한다.

```
buf = 13 4
(13) Specified Angle or PWM Duty(Servo)
Client Disconnected!

Removed proc id: 1584
^C
root@am57xx-evm:~/workspace# ls
a.out          auto_wifi.sh      can_test          can_vcp_serial.c  dsp_wifi_can_serv.c  serial
ap_start.pl    can_serial_complete.c  can_test.c        dsp_central_serv.c  pwm_control.cpp       serial.cpp
root@am57xx-evm:~/workspace# cd /usr/share/wl18xx/
root@am57xx-evm:/usr/share/wl18xx# ls
ap_start.sh          load_wlcore.sh      mod_stop.sh         pre-push.sample     sta_connect-ex.sh
ap_stop.sh           mesh_bridge.sh       p2p_cli.sh          pre-rebase.sample    sta_start.sh
applypatch-msg.sample  mesh_join.sh         p2p_start.sh        prepare-commit-msg.sample  sta_stop.sh
calibrate.sh         mesh_start.sh        p2p_stop.sh         print_stat.sh        testing-boot.sh
commit-msg.sample     mesh_stop.sh         post-update.sample   ps_lock.sh           testing.ini
dynamic-debug.sh      mesh_supPLICANT.conf  pre-applypatch.sample  set_cmd_silence.sh    testing_set_wlcore.
hostapd.conf          mod_start.sh         pre-commit.sample    sta_connect-ex-dhcp.sh  udhcpd.conf
root@am57xx-evm:/usr/share/wl18xx# scp hostapd.conf sdr@192.168.0.52:/home/sdr/proj/RC_Car/dsp_proj/wl18xx_config/

Host '192.168.0.52' is not in the trusted hosts file.
(ssh-rsa fingerprint md5 ac:7e:6e:c2:42:a0:e7:93:92:ba:2c:5e:cb:05:4d:f1)
Do you want to continue connecting? (y/n) ^Croot@am57xx-evm:/usr/share/wl18xx# scp hostapd.conf sdr@192.168.0.56:/home/sdr/proj
sdr@192.168.0.56's password:
hostapd.conf
root@am57xx-evm:/usr/share/wl18xx# scp wpa_supplicant.conf sdr@192.168.0.56:/home/sdr/proj/RC_Car/dsp_proj/wl18xx_config/
sdr@192.168.0.56's password:
wpa_supplicant.conf
root@am57xx-evm:/usr/share/wl18xx#
```



```
sdr@sdr-Z20NH-AS51B5U:~/proj/RC_Car/dsp_proj/wl18xx_config$ git status
```

```
On branch master
```

```
Your branch is up-to-date with 'origin/master'.
```

```
nothing to commit, working directory clean
```

```
sdr@sdr-Z20NH-AS51B5U:~/proj/RC_Car/dsp_proj/wl18xx_config$ ls
```

```
ap_start.pl          can_test.c          dsp_wifi_can_serv.c  serial
auto_wifi.sh         can_vcp_serial.c    prepare.txt          serial
can_serial_complete.c dsp_central_serv.c  pwm_control.cpp      test.pl
```

```
sdr@sdr-Z20NH-AS51B5U:~/proj/RC_Car/dsp_proj$ mkdir wl18xx_config
```

```
sdr@sdr-Z20NH-AS51B5U:~/proj/RC_Car/dsp_proj$ cd wl18xx_config/
```

```
sdr@sdr-Z20NH-AS51B5U:~/proj/RC_Car/dsp_proj/wl18xx_config$ ls
```

```
sdr@sdr-Z20NH-AS51B5U:~/proj/RC_Car/dsp_proj/wl18xx_config$ ifconfig
```

```
enp1s0    Link encap:Ethernet  HWaddr 6c:62:6d:d3:d4:16
          inet addr:192.168.0.56  Bcast:192.168.0.255  Mask:255.255.255
          inet6 addr: fe80::a94b:1773:f163:2081/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10471 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:15531 (15.5 KB)  TX bytes:0 (0.0 KB)
```

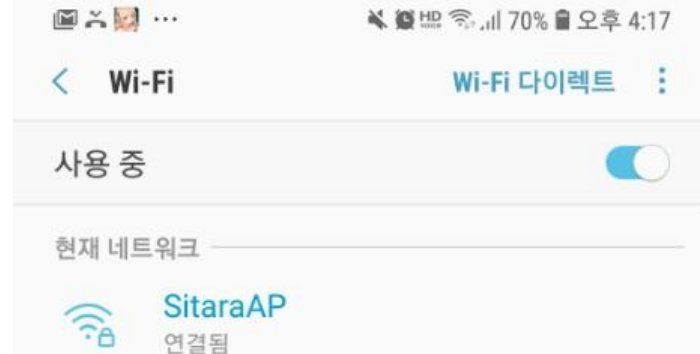
192.168.0.56 이 내가 사용하는 PC 의 주소로  
hostapd.conf 와 wpa\_supplicant.conf 로 잘 복사 되었음을 알 수 있다.  
현재 이 구성을 가지고 앞서 만들어놨던 auto\_wifi.sh 를 구동시키면  
Secure AP 즉 비밀 번호가 적용된 AP 를 만들 수 있게 된다.

```
lo        Link encap:Ethernet  HWaddr 00:00:00:00:00:00
          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:204 errors:0 dropped:0 overruns:0 frame:0
          TX packets:204 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15531 (15.5 KB)  TX bytes:15531 (15.5 KB)
```

```
sdr@sdr-Z20NH-AS51B5U:~/proj/RC_Car/dsp_proj/wl18xx_config$ ls
```

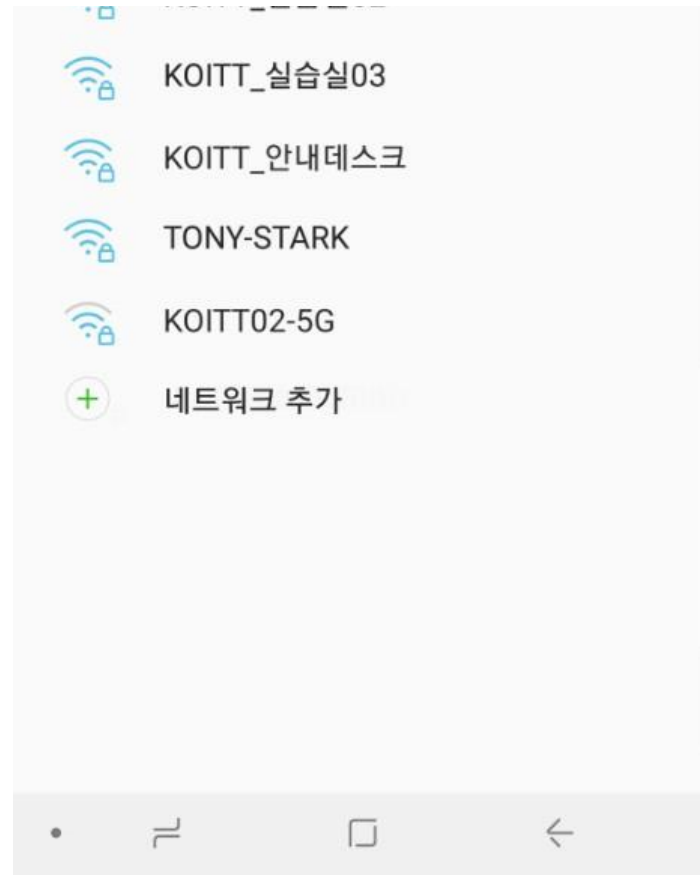
```
hostapd.conf  wpa_supplicant.conf
```

```
sdr@sdr-Z20NH-AS51B5U:~/proj/RC_Car/dsp_proj/wl18xx_config$
```



비밀 번호가 존재함을 나타내는 자물쇠가 생성된 것을 볼 수 있다.  
이제부터 다른 반이나 조종기의 제어권이 없는 사람이 함부로  
우리가 사용하는 DSP 에 접속하는 일은 없을 것이다.

실제로 확인을 하면 외부 접속이 없기 때문에 반응 속도가 매우 빨라졌음을 확인 할 수 있다.



## 최종적으로 정리를 하자면 절차는 아래와 같다.

1. 상호간 IP 충돌이 발생하지 않도록 회의를 하여 적절한 IP 를 고정하도록 한다.
2. 고정한 IP 를 `/usr/share/wl18xx/ap_start.sh` 에 설정하도록 한다.  
이 때 IP 의 마지막 주소는 반드시 1 로 끝나야하며  
맨 앞 두 자리는 192.168 을 반드시 유지하도록 한다.
3. GitHub 에 있는 `hostapd.conf` 와 `wpa_supplicant.conf` 를 DSP 로 복사하도록 한다.  
`scp` 명령어를 활용해야 하는데 아래와 같이 활용하면 된다.  
앞서 팁으로 보여줬듯이 활용해야 한다.

예) 현재 Wi-Fi AP(Router)의 IP 주소가 192.168.77.1 이라고 가정한다.

PC 에서 아래와 같이 입력하면 된다.

```
scp hostapd.conf root@192.168.77.1:/usr/share/wl18xx/  
wpa_supplicant.conf root@192.168.77.1:/usr/share/wl18xx/
```

4. 이제 `auto_wifi.sh` 를 실행하도록 한다.  
그러면 자물쇠 표시가 걸린 Wi-Fi AP 를 볼 수 있을 것이다.
5. TODO: DSP 를 부팅하면 나오는 Matrix App 을 만들어서  
이 App 을 누르면 자동으로 Wi-Fi 가 설정되도록 만들 필요가 있다.  
번거로운 작업 없이 터치 한 번으로 끝내버리자!