

# TI MCU, DSP 및 Xilinx FPGA 프로그래밍 전문가 과정

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

# **DSP Integrated Test I**

이번 통합 테스트는 MCU 작업을 추가로 필요로 한다.

우선 DSP 에 꽃을 Wi-Fi 모듈이 필요하다.

<http://www.ti.com/tool/wl1837modcom8i>

\$49 짜리를 구하면 되고 이를 DSP 에 꽃는다.  
또한 MCU 프로젝트는 아래 링크를 참고하라.

[https://github.com/KOITT2/RC\\_Car/blob/master/experiment/doc/%ED%86%B5%ED%95%A9%ED%85%8C%EC%8A%A4%ED%8A%B8\\_MCU.pdf](https://github.com/KOITT2/RC_Car/blob/master/experiment/doc/%ED%86%B5%ED%95%A9%ED%85%8C%EC%8A%A4%ED%8A%B8_MCU.pdf)

문서 내용에서 소스 코드에 해당하는 부분을 약간 수정 했는데 해당 코드는 아래 링크를 통해 참조하라.

[https://github.com/KOITT2/RC\\_Car/tree/master/mcu\\_proj/Integration\\_Test\\_CAN\\_VCP](https://github.com/KOITT2/RC_Car/tree/master/mcu_proj/Integration_Test_CAN_VCP)

Git 에 올려놓은 프로젝트를 고스란히 올리고 가져오는 문서를 참조하면  
이 부부의 내용을 쉽게 프로젝트로 가져올 수 있을 것이다.













해당 문서는 아래를 참고하라(내용이 기니까 웹상에서 볼 것이라면 More Page 를 누르던 Download 를 해서 본다)

[https://github.com/KOITT2/RC\\_Car/blob/master/experiment/doc/How\\_to\\_Commit\\_All\\_Proj\\_on\\_Github.pdf](https://github.com/KOITT2/RC_Car/blob/master/experiment/doc/How_to_Commit_All_Proj_on_Github.pdf)

DSP 와 관련된 소스 코드들은 아래 링크를 참조하길 바란다.

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

효과적으로 DSP 의 코드를 설명하기 위해 각 코드들을 설명하도록 한다.

Branch: master ▾ RC_Car / dsp_proj /	
silenc3502 DSP CAN Integrated Test Source Code	
..	
 ap_start.pl	Wi-Fi AP 를 구동하기 위한 Perl Script
 auto_wifi.sh	Wi-Fi 설정을 자동화 시키는 Shell Script
 can_serial_complete.c	USB2CAN(FIFO) 버전에서 데이터를 1000 ~ 1900 까지 날려줌
 can_test.c	테스트용 코드이므로 신경쓰지 않아도 됨
 can_vcp_serial.c	USB2CAN(VCP) 버전에서 12345678 데이터를 지속적으로 날림
 dsp_central_serv.c	DSP 에서 Wi-Fi 테스트(서버 – DSP, 클라 – 휴대폰)
 dsp_wifi_can_serv.c	DSP 에서 Wi-Fi 와 MCU 간 CAN 통신 테스트(통합 테스트 코드)
 prepare.txt	쓸대 없는 것이라 조만간 지울 생각
 pwm_control.cpp	테스트용 코드이므로 신경쓰지 않아도 됨
 serial.cpp	테스트용 코드이므로 신경쓰지 않아도 됨
 serial_test.cpp	테스트용 코드이므로 신경쓰지 않아도 됨
 test.pl	Perl Script 의 동작 여부를 테스트 하는 코드임

우선 DSP 에 Wi-Fi PCB 안테나를 연결한다.

그리고 부팅시켜서 아래와 같이 창이 잘 나오는지 확인하도록 한다.



```
sdh@sdh-Z20NH-AS51B5U:~$ ssh root@192.168.0.52
root@am57xx-evm:~# ls
qttest      qttest2     qttest3     tibb         untitled     workspace
root@am57xx-evm:~# cd workspace/
root@am57xx-evm:~/workspace# ls
a.out          auto_wifi.sh      can_test      can_vcp_serial.c
ap_start.pl    can_serial_complete.c  can_test.c    dsp_central_serv.c
root@am57xx-evm:~/workspace# ./auto_wifi.sh
```

Please provide the following information.

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: y

Japanese Standards Applied: n

Class 2 Permissive Change (C2PC) Applied: n

/home/root/workspace

/usr/share/wl18xx

adding wlan1 interface

Configuration file: /usr/share/wl18xx/hostapd.conf

wlan1: interface state UNINITIALIZED->COUNTRY\_UPDATE

Using interface wlan1 with hwaddr 50:33:8b:64:97:d9 and ssid "SitaraAP"

wlan1: interface state COUNTRY\_UPDATE->ENABLED

wlan1: AP-ENABLED

root@am57xx-evm:~/workspace# wlan1: STA 94:76:b7:11:1f:91 IEEE 802.11: authenticated

wlan1: STA 94:76:b7:11:1f:91 IEEE 802.11: associated (aid 1)

wlan1: AP-STA-CONNECTED 94:76:b7:11:1f:91

auto\_wifi.sh 를 구동시키면

아래와 같이 안테나가 설정되고 wlan1 이 활성화된다.



osdr-Z20NH-AS51B5U: ~

```
root@am57xx-evm:~/workspace# ifconfig
eth0      Link encap:Ethernet  HWaddr FC:0F:4B:8C:10:42
          inet addr:192.168.0.52  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::fe0f:4bff:fe8c:1042%763860/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1708 errors:0 dropped:1 overruns:0 frame:0
          TX packets:319 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:213999 (208.9 KiB)  TX bytes:49151 (47.9 KiB)
          Interrupt:97

eth1      Link encap:Ethernet  HWaddr FC:0F:4B:8C:10:43
          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%763860/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:4385 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4385 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:339134 (331.1 KiB)  TX bytes:339134 (331.1 KiB)

wlan0     Link encap:Ethernet  HWaddr 50:33:8B:64:97:D8
          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)

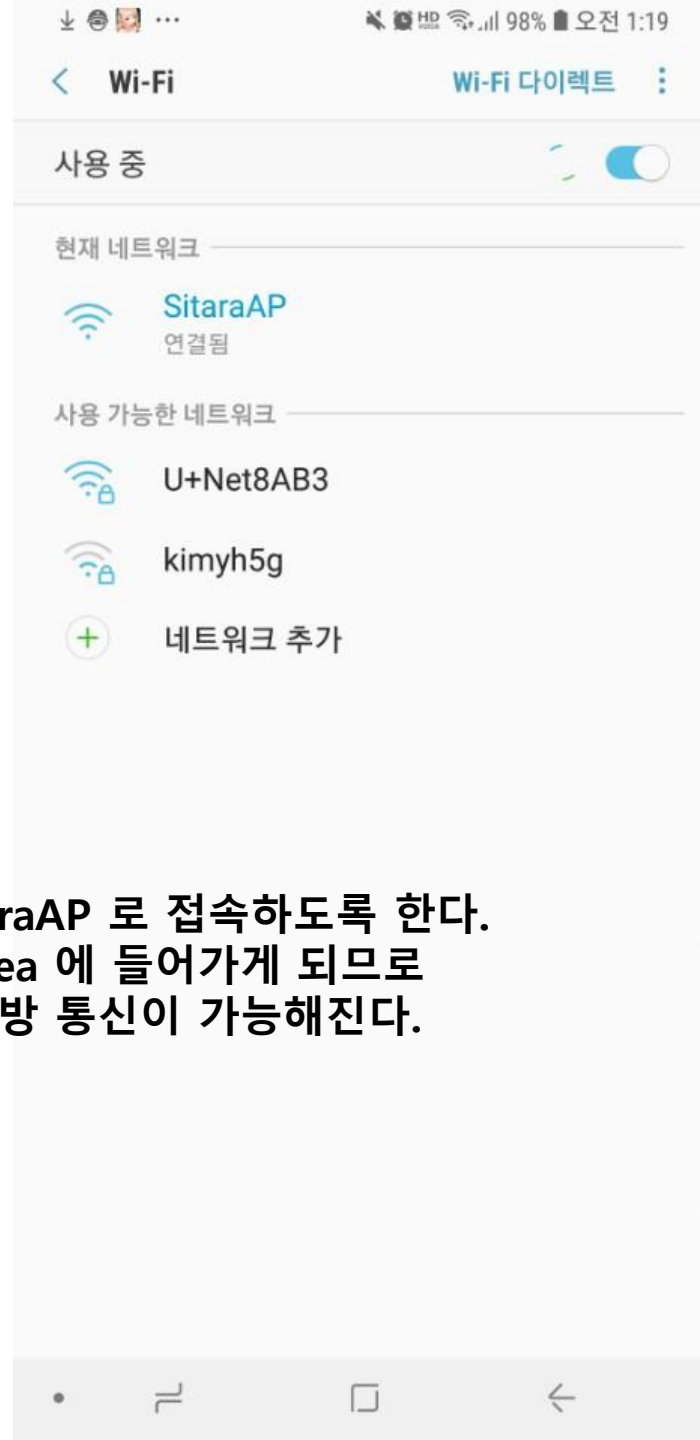
wlan1     Link encap:Ethernet  HWaddr 50:33:8B:64:97:D9
          inet addr:192.168.43.1  Bcast:192.168.43.255  Mask:255.255.255.0
          inet6 addr: fe80::5233:8bff:fe64:97d9%763860/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:9834 (9.6 KiB)
```

root@am57xx-evm:~/workspace#

위의 스크립트 동작 이후에  
wlan1 이 잘 잡히는 것을 볼 수 있을 것이다.

이 Network 의 Subnet Mask 를 보면  
255.255.255.0 이므로  
여러 Wi-Fi AP 를 구성하게 되면  
192.168.43.1, 192.168.42.1, 192.168.41.1  
로 각각의 Subnet Mask 를 가진  
AP 로 구성해야한다.

즉 조별로 Wi-Fi Router 의 주소를  
별로 지정해줘야 한다는 의미다.



휴대폰에서 Wi-Fi 를 SitaraAP 로 접속하도록 한다.  
그러면 해당 Network Area 에 들어가게 되므로  
이제 DSP 와 휴대폰이 쌍방 통신이 가능해진다.



```
root@am57xx-evm:~/network# gcc dsp_central_serv.c
root@am57xx-evm:~/network# ./a.out
use: ./a.out <port>
root@am57xx-evm:~/network# ./a.out 7777
wlan1: STA 50:77:05:ae:0b:b4 IEEE 802.11: authenticated
wlan1: STA 50:77:05:ae:0b:b4 IEEE 802.11: associated (aid 1)
wlan1: AP-STA-CONNECTED 50:77:05:ae:0b:b4

New Client Connected!

Client Disconnected!

Removed proc id: 1359
█
```

이후에 아래와 같이  
dsp\_central\_serv.c 를 컴파일 하여 실행한다.  
포트를 7777 로 잡았으므로 동일하게 포트를 맞춰줘야 한다.

SAMSUNG

12:00 98% 오전 1:20

서버 목록 추가

서버 이름

dsp

IP

192.168.43.1

Port

7777

추가 취소

1 2 3 <X>

4 5 6 완료

7 8 9 .-

0

12:00 98% 오전 1:20

12:00 98% 오전 1:20

192.168.43.1:7777

Connected

잘 연결이 되면 위와 같이  
Connected 메시지가 출력된다.  
(글자가 좀 작아서 잘 봐야함)

전송

1 2 3 4 5 6 7 8 9 0

q w e r t y u i o p

a s d f g h j k l

↑ z x c v b n m <X>

!#1 한/영 , \_ . 완료

12:00 98% 오전 1:20



wlan1: AP-STA-CONNECTED 50:77:05:ae:00:04

root@am57xx-evm:~/workspace# ./a.out 7777

-sh: ./a.out: No such file or directory

root@am57xx-evm:~/workspace# ./a.out 7777

-sh: ./a.out: No such file or directory

root@am57xx-evm:~/workspace# ls

auto\_wifi.sh dsp\_central\_serv.c dsp\_wifi\_can\_serv.c serial

root@am57xx-evm:~/workspace# gcc dsp\_central\_serv.c

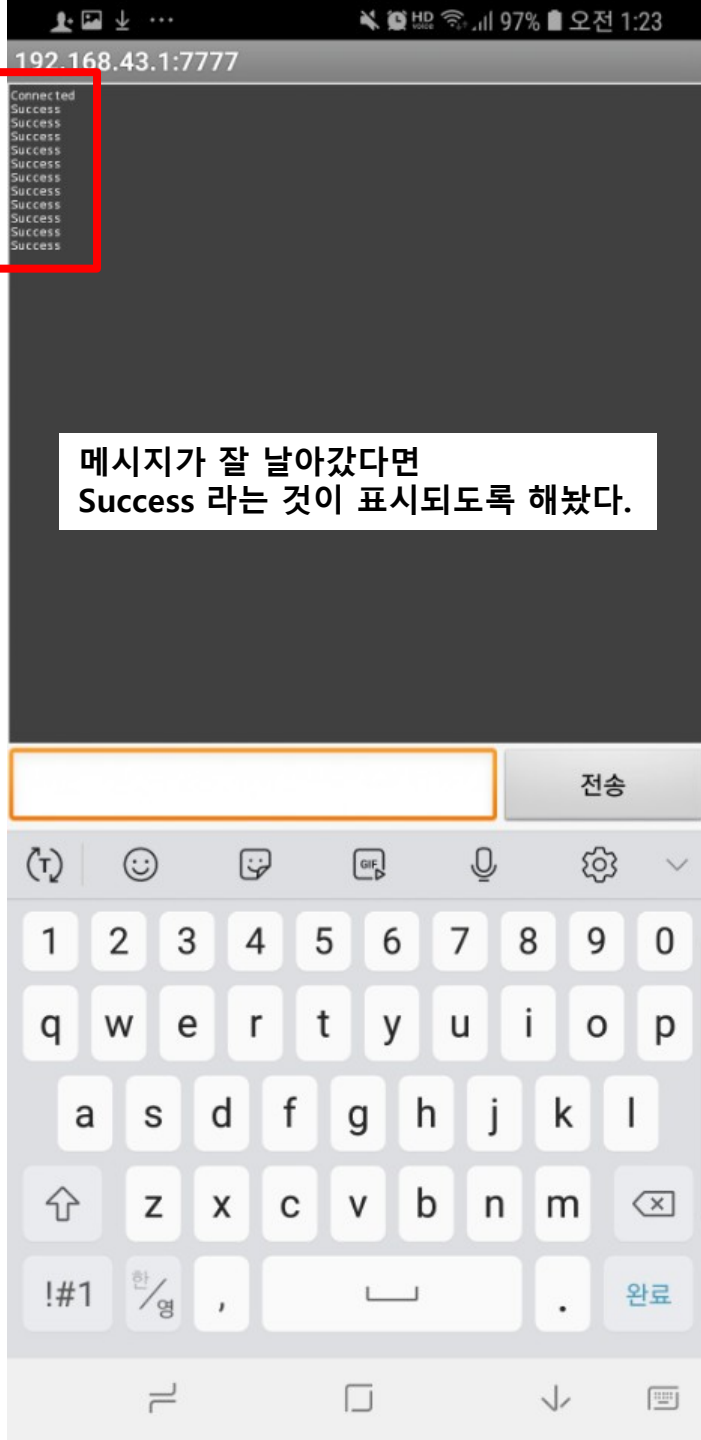
root@am57xx-evm:~/workspace# ./a.out 7777

New Client Connected!

- (1) Turn Left
  - (2) Turn Right
  - (3) Go Straight
  - (4) Go backward
  - (5) Stop
  - (6) Collision Warn
  - (7) Left Winker
  - (8) Right Winker
  - (9) Head Light
  - (10) DSP Image Meta Data
  - (11) FPGA Lidar Meta Data
  - (1) Turn Left
  - (12) Specified Velocity or PWM Duty
  - (13) Specified Angle or PWM Duty(Servo)
- Client Disconnected!

Removed proc id: 2951

휴대폰에서 값을 입력하면 다음과 같이  
입력한 값에 대한 적절한 메시지가 출력될 것이다.  
이를 확인했다면 dsp\_central\_serv.c 에 대한 테스트는 모두 완료된 것이다.  
즉 휴대폰과 DSP 가 Wi-Fi 로 잘 통신을 수행함을 의미한다.



다음으로 MCU 와 DSP 간의 CAN 통신을 테스트하도록 한다.  
이 경우에는 can\_serial\_complete.c 코드를 활용하여 테스트 하도록 한다.  
MCU 에서 Can Transmit 을 해주고 DSP 에서 해당 정보를 받아서  
패킷 형식은 유지한채로 데이터만 바꿔서 다시 MCU 로 쏘주는 것이다.

MCU 코드는 아래를 링크를 참고하도록 하고

DSP 에서 단순히 can\_serial\_complete.c 를 컴파일 하여 구동시키면 된다.  
Network 프로그래밍이 아니므로 별도의 포트 지정은 필요 없다.



이제 MCU 와 DSP 그리고 휴대폰을 연동하여 통합 테스트를 수행하도록 한다.  
이번에는 dsp\_wifi\_can\_serv.c 를 활용하여 테스트 하도록 한다.  
Network 프로그래밍이 연동되므로 포트 설정이 필요하다.  
7777 로 해도 되고 원하는 값으로 해도 상관 없다.  
다만 상호간 포트 값은 맞춰줘야함을 잊지 말자!





```
root@am57xx-evm:~/workspace# ./a.out 7777
```

```
fd = 3
```

```
New Client Connected!
```

7 번이 방향 지시등을 제어하는 프로토콜이고

13 번은 PWM Duty 를 조정하는 프로토콜이다.

휴대폰에서 적절하게 신호를 보내서 이를 제어하도록 한다.

잘 동작한다면 휴대폰에서 Wi-Fi 로 DSP 에 가서 CAN 을 통해 MCU 로 정보가 전달 될 것이다.

```
temp = 1
```

```
(1) Turn Left
```

```
temp = 7
```

```
can buf = 2 0 8 40 1 0 0 0 7 1 0 0 0 0 0 0 51 3
```

```
(7) Left Winker
```

```
temp = 7
```

```
can buf = 2 0 8 40 1 0 0 0 7 0 0 0 0 0 0 0 50 3
```

```
(7) Left Winker
```

```
temp = 13
```

```
can buf = 2 0 8 40 1 0 0 0 d 1 1 0 0 0 0 0 58 3
```

```
(13) Specified Angle or PWM Duty(Servo)
```

```
temp = 13
```

```
can buf = 2 0 8 40 1 0 0 0 0 1 0 0 0 0 0 0 4a 3
```

```
(13) Specified Angle or PWM Duty(Servo)
```

```
temp = 13
```

```
can buf = 2 0 8 40 1 0 0 0 d 1 7 0 0 0 0 0 5e 3
```

```
(13) Specified Angle or PWM Duty(Servo)
```

```
temp = 13
```

```
can buf = 2 0 8 40 1 0 0 0 d 1 8 0 0 0 0 0 5f 3
```

```
(13) Specified Angle or PWM Duty(Servo)
```

```
Client Disconnected!
```

```
Removed proc id: 1257
```

모터가 꺼지고 방향 지시등이 켜지는 것을 확인하면 된다.

## MoonWalker MW USB2CAN(FIFO) v1.1

Windows10 사용가능, USB to CAN Converter, 데이터 전송률 최대 1M byte/second, 전용 모니터링 툴 / SDK 제공



▶ 상품코드	1323536
▶ 판매가격	72800 원 (부가
▶ 제조사	NTRexLAB
▶ 적립금	원
▶ 평균준비기간	1~2일
▶ 브랜드	MoonWalker <span>브랜</span>
▶ A/S정보	
▶ 최소주문수량	1 개
▶ 수량	<input type="text" value="1"/> <span>↕</span>

현재까지의 CAN 통신은 해당 모듈을 통해 수행되었다.  
Linux 에서 사용 불가라고 하는데 우리는 잘 쓰고 있다.



바로구매

장바

## MoonWalker MW USB2CAN(VCP) v1.1

Windows10사용가능, USB to CAN Converter, 데이터 전송률 최대 300K byte/second, 전용 모니터링 툴 / SDK 제공



▶ 상품코드	1323537
▶ 판매가격	58,800원 (부가
▶ 제조사	NTRexLAB
▶ 적립금	0원
▶ 평균준비기간	1~2일
▶ 브랜드	MoonWalker 브
▶ A/S정보	
▶ 최소주문수량	1 개
▶ 수량	<input type="text" value="1"/>

이 모듈을 사용할 경우에는 또 다른 사용 방식이 필요하다.

이를 위해 can\_vcp\_serial.c 라는 코드가 존재한다.

MCU 와 DSP 를 연결하고 구동시키면 잘 동작하는 것을 볼 수 있을 것이다.

랑구매 Call 히트

바로구매

장

