

(IOTG200)

Hardware Test Guide

<http://www.mangoboard.com/>

<http://cafe.naver.com/embeddedcrazyboys>

Crazy Embedded Laboratory

Document History

Revision	Date	Change note
1.0	2015.01.13	Initial 오영환

목 차

1.	부팅용 SD 카드 제작.....	4
1.1.	U-Boot, Kernel 다운로드.....	4
1.2.	U-Boot 부팅 후 fdisk.....	4
1.3.	Format Partition.....	4
1.4.	파일시스템 복사.....	5
2.	부팅 Test.....	5
3.	Key Test.....	6
3.1.	Reset Button.....	6
3.2.	User Button.....	6
4.	Buzzer Test.....	7
5.	LED Test.....	7
6.	USB Host Test.....	7
7.	USB Device Test.....	8
8.	Ethernet Test.....	8
9.	RTC Test.....	10
10.	HDMI Test.....	11
11.	ZigBee Test.....	18
12.	Wi-Fi/BT Test.....	11
12.1.	Wi-Fi Test.....	11
12.2.	Wi-Fi AP 기능 Test.....	12
12.3.	Bluetooth Test.....	14
13.	LTE Module Test.....	17
14.	Sensor Connector (I2C2) Test.....	16
15.	MIPI CSI Camera Test.....	19
16.	Ext Connector (UART0, I2C0, SPI0, I2S0, XEINT3) Test.....	19

1. 부팅용 SD 카드 제작

1.1. U-Boot, Kernel 다운로드

사용할 SD 카드를 우분투 PC에서 인식 시킵니다.

```
$ df
/dev/sdb1      7814144      32  7814112    1% /media/9016-4EF8
```

/dev/sdb로 인식이 되었는데 sdc, sdd 등이 될 수도 있습니다.

```
$ sudo ./sdwriter sdb 220 bin
```

인식된 디바이스 이름인 sdb로 bin 옵션을 주어서 다운로드 합니다. 위 과정은 u-boot와 커널을 다운로드 해주는 것입니다.

1.2. U-Boot 부팅 후 fdisk

보드 Bottom 면의 Boot Switch를 모두 Off로 놓으면 SD 카드로 자동으로 부팅이 됩니다.

디버그 시리얼 포트의 설정은 아래와 같습니다.

(Baud Rate: 115200, Data Bits: 8, Parity: None, Stop Bits: 1, Flow Control: None)

```
MANGO220 # fdisk -c 0 2048 2048 1024
```

U-Boot까지만 부팅한 후 fdisk 수행, 위에 표시된 것 같이 SD 카드의 파티션을 나누게 됩니다. 뒤의 3개의 숫자는 파티션 2, 3, 4의 크기를 나타내는 것입니다. 사용하는 SD 카드의 크기에 따라서 적절한 값을 선택하면 됩니다.

1.3. Format Partition

이제 보드의 전원을 끄고, SD 카드를 다시 우분투에 인식 시킵니다.

```
yhoh@ubuntu:~$ sudo fdisk -l
```

sudo fdisk -l 을 수행해 보면 U-Boot에서 수행했던 내용을 확인할 수 있습니다.

```

yhoh@ubuntu:~$ sudo mkfs.vfat /dev/sdb1
yhoh@ubuntu:~$ sudo mkfs.ext4 -j /dev/sdb2
yhoh@ubuntu:~$ sudo mkfs.ext4 -j /dev/sdb3
yhoh@ubuntu:~$ sudo mkfs.ext4 -j /dev/sdb4

```

위 명령으로 모든 파티션을 포맷해 줍니다.

1.4. 파일시스템 복사

SD 카드를 우분투에서 제거한 후에 다시 인식 시킵니다.

```

yhoh@ubuntu:~$ df
/dev/sdb3 295525 2062 277781 1% /media/81070050-44ca-4e64-a152-87ff8a493416
/dev/sdb1 6726488 4 6726484 1% /media/3455-B417
/dev/sdb2 295525 2062 277781 1% /media/409bf2ef-821f-4c66-86cf-1fdeb31cdded
/dev/sdb4 295525 2062 277781 1% /media/697c1f49-8fa1-446e-985c-16224cda3fa6
yhoh@ubuntu:~$ cd /media/409bf2ef-821f-4c66-86cf-1fdeb31cdded/

```

사용하는 것은 **/dev/sdb2** 입니다. 이곳으로 이동합니다.

```

$ sudo tar xvf ~/sd_fusing_image/keti-m220/custom_datafs.tar
$ sudo tar xvf ~/sd_fusing_image/keti-m220/rootfs.tar
$ sync

```

rootfs.tar, custom_datafs.tar의 압축을 풀고 sync를 수행합니다.

2. 부팅 Test

```

Welcome to Mango220
Mango220 login: root
Password:
[root@Mango220 ~]#

```

로그인 ID는 **root**, Password도 **root** 입니다. 정상 부팅되어 로그인이 되는 지 확인 합니다.

3. Key Test

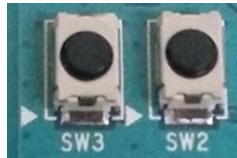
3.1. Reset Button

리셋 버튼을 눌러서 정상적으로 리셋이 되는지 확인 합니다.

3.2. User Button

```
# hexdump /dev/input/event0
00000000 03d0 0000 e56c 0003 0001 008b 0001 0000
00000010 03d0 0000 e585 0003 0000 0000 0000 0000
00000020 03d0 0000 df07 0006 0001 008b 0000 0000
00000030 03d0 0000 df13 0006 0000 0000 0000 0000
00000040 03d1 0000 9b85 000a 0001 009e 0001 0000
00000050 03d1 0000 9b97 000a 0000 0000 0000 0000
00000060 03d1 0000 9532 000d 0001 009e 0000 0000
00000070 03d1 0000 953e 000d 0000 0000 0000 0000
```

hexdump /dev/input/event0를 수행한 후에 버튼을 눌러보면 위와 같이 출력됩니다.



위 로그는 SW3, SW2를 차례로 눌렀다 떼는 동작시의 로그입니다.

/include/linux/input.h에 보면 아래와 같은 정의를 발견할 수 있습니다.

```
#define KEY_BACK      158      /* AC Back */
#define KEY_MENU      139      /* Menu (show menu) */
```

키를 누르면 value값은 1을 가집니다. 떼게 되면 0을 가지게 됩니다.

158은 Hex로 0x9E 값을 가집니다.

139은 Hex로 0x8B 값을 가집니다.

4. Buzzer Test

```
# ./buzzer_test.sh init
buzzer init
```

init을 수행하면 초기화가 진행됩니다.

```
# ./buzzer_test.sh on 2
buzzer On
time is 2 seconds.
```

on을 주면서 시간을 설정하면 그 시간동안 부저가 동작합니다.

5. LED Test

```
# ./led_test.sh init
# ./led_test.sh on 1
# ./led_test.sh on 2
# ./led_test.sh on 3
# ./led_test.sh off 1
# ./led_test.sh off 2
# ./led_test.sh off 3
```

led_test.sh 셸 스크립트를 만들어 두었습니다. led_test.sh init을 수행하면 초기화가 진행되고, on, off와 LED 번호를 주면 LED가 켜지거나 꺼지게 됩니다.

6. USB Host Test

USB Memory를 장착한다.

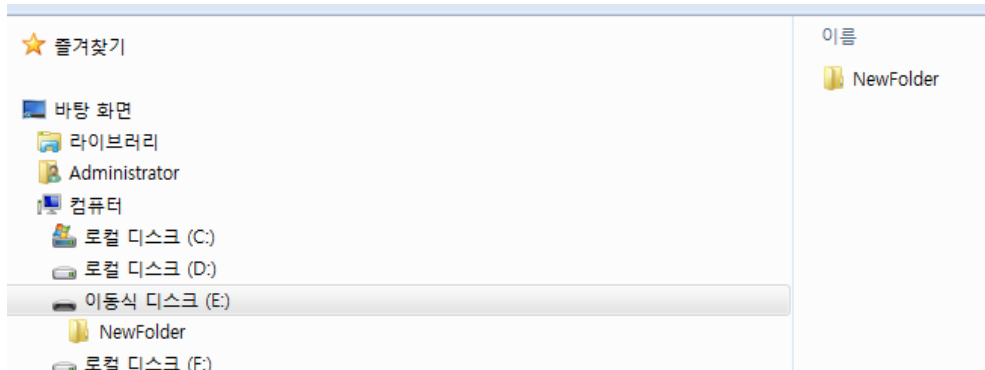
```
Success to mount ! /dev/sda1
```

위와 같이 정상적으로 마운트가 되고 동작도 문제가 없어야 합니다.

7. USB Device Test

```
# ./usb_dev_mass.sh
```

USB device cable을 연결하고 위 스크립트를 수행합니다.



PC에 이동식 디스크가 새로이 잡힙니다. 여기에 새 폴더를 만듭니다.

```
[root@Mango220 ~]# cd /media/mmcblk0p1
[root@Mango220 mmcblk0p1]# ls
/media/mmcblk0p1
total 4
drwxrwxrwx 2 root root 4096 Oct 9 2014 NewFolder/
[root@Mango220 mmcblk0p1]#
```

보드 상에서 PC에서 만든 폴더를 확인할 수 있습니다.

8. Ethernet Test

```
[root@Mango220 ~]# udhcpc -i eth0
[root@Mango220 ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:04:A3:E8:40:A2
          inet addr:192.168.57.155  Bcast:192.168.57.255  Mask:255.255.255.0
          inet6 addr: fe80::204:a3ff:fee8:40a2/64 Scope:Link
```

24AA02E48T MAC Control 확인

9. SSH 연결 시험

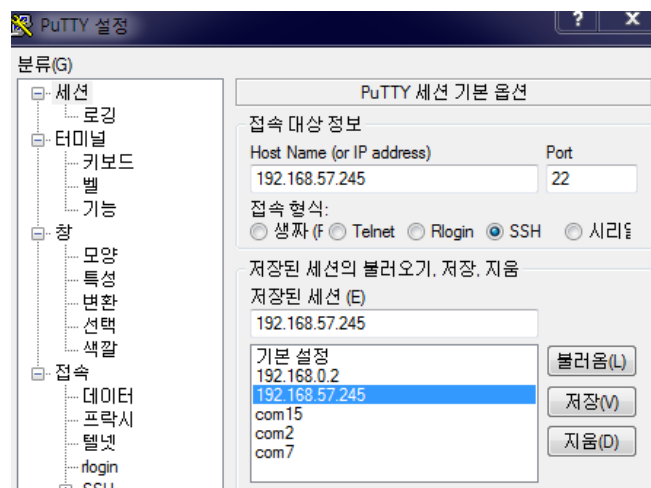
최초 부팅을 수행하면 아래와 같이 키 생성 작업을 수행하게 됩니다.

```
Generating RSA Key...
Generating public/private rsa1 key pair.
Your identification has been saved in /etc/ssh_host_key.
Your public key has been saved in /etc/ssh_host_key.pub.
The key fingerprint is:
50:7c:d7:cf:5c:7b:99:89:47:8f:9c:7e:3b:62:83:ee
The key's randomart image is:
+--[RSA1 2048]-----+
|      .. .      |
| oE. o .|
+-----+
... ..
```

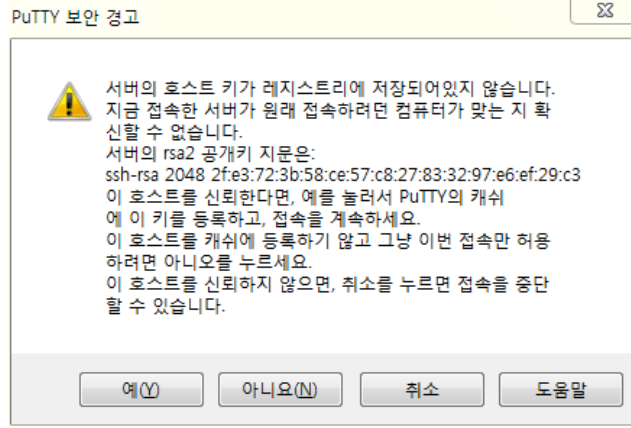
Starting sshd: OK

Starting sshd: OK가 수행되면 정상적으로 데몬이 떠 있는 것입니다.

보드의 ip 주소는 192.168.57.245 입니다.



PuTTY에서 보드로 접속을 시도합니다.



최초에는 위와 같이 경고 메시지가 보이게 됩니다. 예를 선택하면 됩니다.

```
yhoh@crz-ubuntu1204-02: ~/work/m2450_linux_120706_Pooh
login as: root
root@192.168.57.245's password:
[root@mangoC210 ~]# 1
/root
total 72
-rwxr-xr-x 1 default default 794 Oct 25 2013 buzzer_test.sh*
-rwxr-xr-x 1 default default 733 Jun 5 2013 led_test.sh*
drwxr-xr-x 2 default default 4096 Mar 5 2013 nfs_mount/
-rwxr-xr-x 1 default default 214 Mar 6 2013 nfs_mount.sh*
-rwxrwxr-x 1 default default 11868 Jun 5 2013 rs232_test*
-rwxrwxr-x 1 default default 20372 Aug 1 18:09 sensor_test_AL*
-rwxrwxr-x 1 default default 20539 Aug 4 19:50 sensor_test_HT*
[root@mangoC210 ~]#
```

root, root로 접속하면 터미널에서의 모습과 동일한 환경을 만날 수 있습니다.

10. RTC Test

```
# rdate -s time.bora.net
# hwclock -u --systohc
```

rdate를 이용해서 정확한 system 시간을 설정합니다.

hwclock 설정 시에 -u 옵션을 꼭 설정해야 합니다. 이는 UTC 시간으로 설정하는 것입니다.

이후 보드의 전원을 완전히 제거하고 다시 부팅하였을때 이전에 설정한 시간으로 다시 설정되어 있는 것을 확인할 수 있습니다.

```
# date
```

전원 케이블을 분리한 후 다시 부팅을 하였을 때 현재 시간이 정확히 출력되어야 합니다.

11. HDMI Test

HDMI 케이블을 연결하고 부팅을 수행하면 모니터의 화면이 아래와 같이 출력되어야 합니다.



12. Wi-Fi/BT Test

12.1. Wi-Fi Test

```
# iwlist wlan0 scanning | grep ESSID
ESSID:"crz_treego" [24]
```

```
# iwconfig wlan0 essid crz_treego
# iwconfig
wlan0 IEEE 802.11-DS ESSID:"crz_treego" [29] Nickname:""
Mode:Managed Frequency=2.462 GHz Access Point: 00:26:66:81:91:54
```

```
[root@Mango220 ~]# udhcpc -i wlan0
[root@Mango220 ~]# ping www.google.com
PING www.google.com (173.194.117.144): 56 data bytes
64 bytes from 173.194.117.144: seq=0 ttl=51 time=38.595 ms
```

12.2. Wi-Fi AP 기능 Test

```
[root@Mango220 ~]# ping www.google.com
PING www.google.com (173.194.117.145): 56 data bytes
64 bytes from 173.194.117.145: seq=0 ttl=51 time=40.367 ms
64 bytes from 173.194.117.145: seq=1 ttl=51 time=34.990 ms
```

이더넷을 통해서 구글에 ping이 접속되는지를 확인합니다.

```
# cd /lib/firmware/mrvl/
```

/lib/firmware/mrvl/ 폴더로 이동합니다.

```
[root@Mango220 mrvl]# cp config/80211d_domain.conf .
```

config 폴더의 80211d_domain.conf를 현재 폴더에 복사합니다.

```
[root@Mango220 mrvl]# ./uaputl.exe bss_start
BSS start successful!
```

./uaputl.exe bss_start를 수행합니다.
위와 같이 성공 메시지가 출력되어야 합니다.

```
[root@mangoC210 mrvl]# ifconfig uap0 192.168.100.1 up
```

ifconfig를 사용해서 uap0에 대한 IP를 할당합니다.

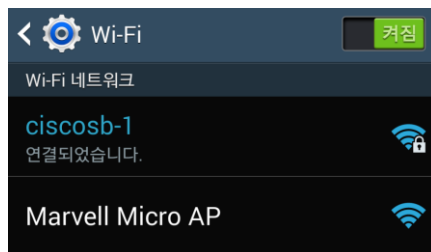
```
[root@mangoC210 mrvl]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:04:A3:E8:40:A2
          inet addr:192.168.57.155  Bcast:192.168.57.255  Mask:255.255.255.0
          inet6 addr: fe80::204:a3ff:fee8:40a2/64 Scope:Link
          ... ..
uap0      Link encap:Ethernet  HWaddr AC:3F:A4:04:D2:72
          inet addr:192.168.100.1  Bcast:192.168.100.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
```

AP 모드로 동작하는 상황에서 ping 접속만 수행하는 것이 아니라 eth0를 통해서 인터넷 접속까지 가능하려면 아래의 MASQUERADE 설정을 수행해 주어야 합니다.

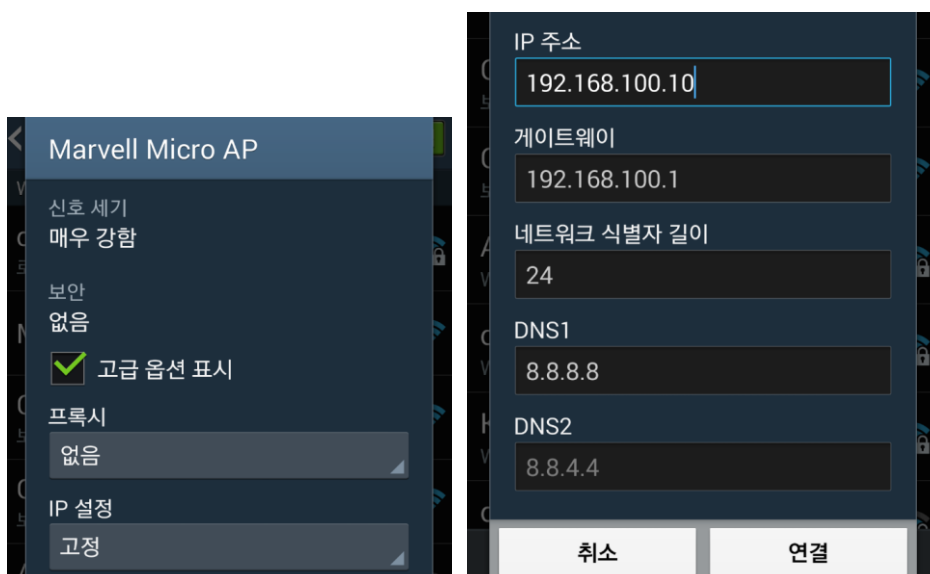
```
iptables --flush;
iptables --table nat --flush;
iptables --delete-chain;
iptables --table nat --append POSTROUTING --out-interface eth0 -j MASQUERADE;
iptables --append FORWARD --in-interface uap0 -j ACCEPT;
echo 0 > /proc/sys/net/ipv4/ip_forward;
echo 1 > /proc/sys/net/ipv4/ip_forward;
```

위의 모든 내용을 차례로 명령창에 입력하고 수행시킵니다.

이제 휴대폰에서 먼저 Wi-Fi를 켜고 검색을 합니다.



"Marvell Micro AP"란 이름으로 검색이 되어야 합니다.



검색된 Marvell Micro AP 부분을 선택해서 고급 옵션 표시를 켜고 IP 설정을 고정으로 변경합니다. (혹은 폰의 버전에 따라서 네트워크 수정을 선택하고 고급 옵션 표시를 선택해야 할 수도 있습니다. Static(정적)으로 표시되는 경우도 있습니다.)

IP 주소를 적절히 설정합니다. 위 그림에서 192.168.100.10으로 설정하였습니다. Gateway는 192.168.100.1로 설정합니다. 연결을 눌러서 Marvell Micro AP에 연결합니다. (기기에 따라서는 저장을 하고 연결은 나중에 수행될 수도 있습니다.)

```
[root@mangoC210 mrvl]# ping 192.168.100.10
PING 192.168.100.10 (192.168.100.10): 56 data bytes
64 bytes from 192.168.100.10: seq=0 ttl=64 time=56.302 ms
64 bytes from 192.168.100.10: seq=1 ttl=64 time=79.585 ms
```

192.168.100.10 부분에 ping을 수행하면 정상적인 접속이 이루어져 있습니다.



이제 인터넷 접속을 수행해 보면 위와 같이 정상적으로 접속이 됨을 확인할 수 있습니다.

12.3. Bluetooth Test

```
[root@Mango220 ~]# hciconfig
hci0:  Type: BR/EDR  Bus: SDIO
        BD Address: AC:3F:A4:04:D2:79  ACL MTU: 1021:6  SCO MTU: 120:6
```

```

DOWN
RX bytes:571 acl:0 sco:0 events:18 errors:0
TX bytes:164 acl:0 sco:0 commands:19 errors:0
hci1: Type: BR/EDR Bus: UNKNOWN
BD Address: 78:D2:04:A4:3F:AC ACL MTU: 1492:4 SCO MTU: 0:0
DOWN
RX bytes:155 acl:0 sco:0 events:11 errors:0
TX bytes:87 acl:0 sco:0 commands:11 errors:0

```

```

[root@Mango220 ~]# hciconfig hci0 up
[root@Mango220 ~]# hcidtool dev
Devices:
      hci0      AC:3F:A4:04:D1:C5

```

내 폰을 BT가 검색되도록 설정한 후 inquiring을 수행시킨다.

```

[root@Mango220 ~]# hcidtool inq
Inquiring ...
      38:0B:40:81:49:91      clock offset: 0x1fcd      class: 0x5a020c
[root@Mango220 ~]# hcidtool info 38:0B:40:81:49:91
Requesting information ...
BD Address: 38:0B:40:81:49:91
Device Name: YHOH-SHV-E330
LMP Version: 4.0 (0x6) LMP Subversion: 0x4106
Manufacturer: Broadcom Corporation (15)
Features page 0: 0xbf 0xfe 0xcf 0xff 0xdf 0xff 0x7b 0x87
      <3-slot packets> <5-slot packets> <encryption> <slot offset>
      <timing accuracy> <role switch> <sniff mode> <RSSI>
      <channel quality> <SCO link> <HV2 packets> <HV3 packets>
      <u-law log> <A-law log> <CVSD> <paging scheme> <power control>
      <transparent SCO> <broadcast encrypt> <no. 24>
      <EDR ACL 2 Mbps> <EDR ACL 3 Mbps> <enhanced iscan>
      <interlaced iscan> <interlaced pscan> <inquiry with RSSI>
      <extended SCO> <EV4 packets> <EV5 packets> <no. 34>
      <AFH cap. slave> <AFH class. slave> <LE support>
      <3-slot EDR ACL> <5-slot EDR ACL> <sniff subrating>
      <pause encryption> <AFH cap. master> <AFH class. master>

```

```
<EDR eSCO 2 Mbps> <EDR eSCO 3 Mbps> <3-slot EDR eSCO>  
<extended inquiry> <LE and BR/EDR> <simple pairing>  
<encapsulated PDU> <err. data report> <non-flush flag> <LSTO>  
<inquiry TX power> <EPC> <extended features>  
Features page 1: 0x03 0x00 0x00 0x00 0x00 0x00 0x00 0x00
```

13. Sensor Connector (I2C2) Test

센서 연결 커넥터를 장착합니다. 조도 센서를 연결합니다. sensor_test_AL을 수행해서 조도 측정을 수행합니다.

```
[root@Mango220 ~]# ./sensor_test_AL  
[Test_AL] Start Ambient Light Sensor  
[Test_AL] i2c device open done  
[Test_AL] set_slave_addr done  
[Test_AL] read data: 0x0000, Lux: 0.000000  
[Test_AL] read data: 0x00CC, Lux: 170.000000  
[Test_AL] read data: 0x00CB, Lux: 169.166672  
[Test_AL] read data: 0x00CB, Lux: 169.166672  
[Test_AL] read data: 0x00C6, Lux: 165.000000  
[Test_AL] read data: 0x00AA, Lux: 141.666672  
[Test_AL] read data: 0x0061, Lux: 80.833336  
[Test_AL] read data: 0x000F, Lux: 12.500000  
[Test_AL] read data: 0x000B, Lux: 9.166667
```

온습도 센서를 연결합니다. sensor_test_HT를 수행해서 온습도 측정을 수행합니다.

```
[root@Mango220 ~]# ./sensor_test_HT  
[Test_HT] Start Temperature & Humidity Sensor  
[Test_HT] i2c device open done  
[Test_HT] set_slave_addr done  
[Test_HT] Temperature: 24.825829, Humidity: 16.693634  
[Test_HT] Temperature: 24.836554, Humidity: 16.693634  
[Test_HT] Temperature: 24.836554, Humidity: 16.693634  
[Test_HT] Temperature: 24.836554, Humidity: 16.701263  
[Test_HT] Temperature: 24.858004, Humidity: 16.693634
```


14. LTE Module Test

최초 동작시 USB0에 RNDIS를 통해서 LTE 모듈과 연결 됩니다.

```
[root@Mango220 ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:04:A3:E8:5A:62
          UP BROADCAST MULTICAST  MTU:1488  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: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  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:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

usb0      Link encap:Ethernet  HWaddr 1E:05:16:42:09:51
          inet addr:192.168.225.4  Bcast:192.168.225.255  Mask:255.255.255.0
          inet6 addr: fe80::1cc5:16ff:fe42:951/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:22 errors:0 dropped:0 overruns:0 frame:0
          TX packets:35 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2527 (2.4 KiB)  TX bytes:5207 (5.0 KiB)

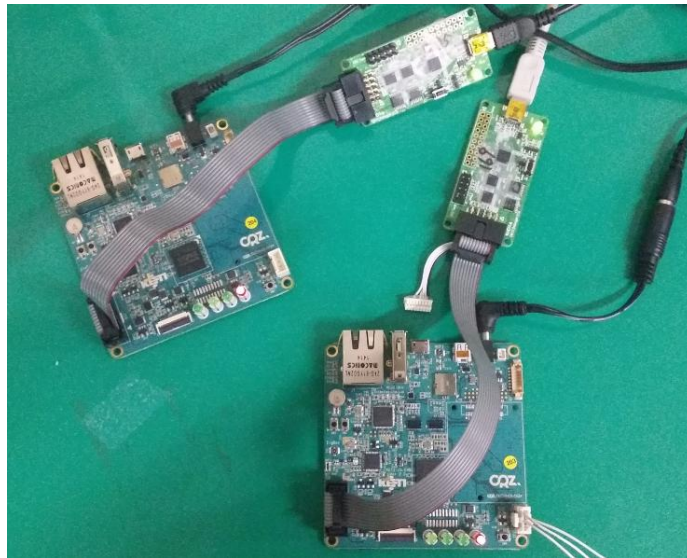
[root@Mango220 ~]# ping www.google.co.kr
PING www.google.co.kr (117.111.0.163): 56 data bytes
64 bytes from 117.111.0.163: seq=0 ttl=58 time=156.378 ms
64 bytes from 117.111.0.163: seq=1 ttl=58 time=28.122 ms
64 bytes from 117.111.0.163: seq=2 ttl=58 time=26.573 ms
64 bytes from 117.111.0.163: seq=3 ttl=58 time=26.193 ms
64 bytes from 117.111.0.163: seq=4 ttl=58 time=26.713 ms
64 bytes from 117.111.0.163: seq=5 ttl=58 time=26.117 ms
64 bytes from 117.111.0.163: seq=6 ttl=58 time=25.900 ms
█
```

usb0의 IP 주소를 확인하고, ping test를 수행합니다.

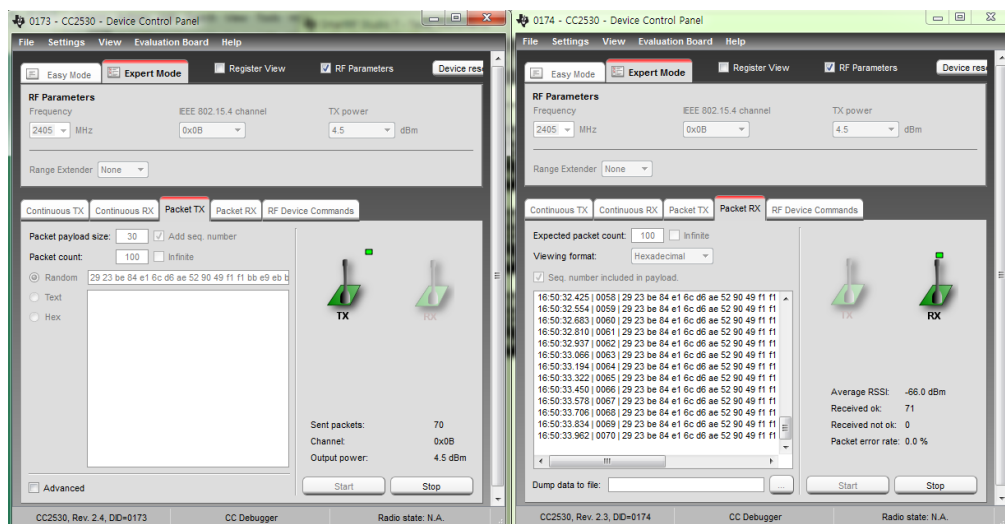
15. ZigBee Test

15.1. 디바이스 인식 시험

CC Debugger를 사진과 같이 연결한 후, CC Debugger의 리셋 버튼을 눌렀을 때 녹색 LED가 켜지면 정상적으로 디바이스가 인식된 것입니다.



15.2. SmartRF Studio 통신 시험



두 보드에 동일하게 연결하고 TI SmartRF Studio를 수행해서 한쪽은 TX, 다른 한쪽은 RX로 설정해서 통신이 원활하게 수행되는가를 검사합니다.

15.3. ZigBee Serial Test

Work\CC2530 Test Binary

한쪽은 z100_cordinator_2011.1227.2034.hex

다른 한쪽은 z100_device_2011.1227.2034.hex 를 각각 구웠습니다.

```
./rs232_test 115200 /dev/ttySAC3
```

위 명령으로 UART3와 통신을 할 수 있습니다. ZigBee CC2530과 연결된 것입니다.

rs232_test 소스는 app/rs232_test_app 부분에 존재합니다.

16. MIPI CSI Camera Test

17. Ext Connector (UART0, I2C0, SPI0, I2S0, XEINT3) Test