

# Mango-AM335x 이더넷 Bonding 기능

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

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

Crazy Embedded Laboratory

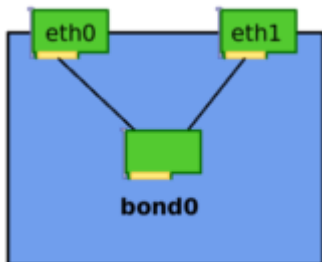
# Document History

Revision	Date	Change note
Init	2017-06-22	전종인

1. Bonding이란.....	4
2. 커널 Configuration .....	4
3. 파일 시스템 수정하기.....	4
4. 테스트 방법 .....	5
5. 레퍼런스 링크.....	8

# 1. Bonding이란

여러개의 네트워크 인터페이스를 하나로! Bonding 하기



bonding 기술에 대해서 다뤄보고자 한다. 이것은 여러개의 네트워크 인터페이스를 하나로 만들어 이더넷 카드 장애 또는 링크 Fail 에 대해 대비를 할 수 있다. 또한 로드밸런스를 구현할 수 있으며, 자연스럽게 Fault tolerance 를 보장해주어 고장 허용한계를 최소화 할 수도 있겠다. 스위치 장비 같은것에서 보면 "채널본딩", "트링킹" 같은 기술들이 이와 이론적으로는 유사하다고 할 수 있다.

## 2. 커널 Configuration

```
$ ./build_kernel config
```

```
Symbol: BONDING [=y]
| Type : tristate
| Prompt: Bonding driver support
| Location:
|   -> Device Drivers
|     -> Network device support (NETDEVICES [=y])
| (1)   -> Network core driver support (NET_CORE [=y])
```

```
CONFIG_BONDING=y
```

커널을 컴파일 한다.

컴파일 한 이미지를 sd에 Write한다.

## 3. 파일 시스템 수정하기

buildroot에서 ifenslave 명령이 필요하다.

```
$vi package/busybox/busybox-1.21.x.config 파일에
```

```
CONFIG_IFENSLAVE=y
```

추가한다.

컴파일을 한다.

output/images/rootfs.tar.gz 압축을 풀면 ifenslave 파일이 /sbin/ 디렉토리에 존재를 한다.  
해당 파일을 부팅 파일 시스템에 복사를 한다.

## 4. 테스트 방법

/etc/network/interface 에 내용은 모두 삭제한다.

부팅 후 아래와 같이 명령을 입력한다.

```
[root@localhost ~]# ifconfig bond0 192.168.100.100 netmask 255.255.255.0
[root@localhost ~]# ifenslave bond0 eth0 eth1
```

부팅 시 자동으로 실행하고 싶으면

```
[root@localhost ~]# vi /etc/init.d/S98bonding
```

파일에 아래와 같이 내용을 추가한다.

```
#!/bin/sh
ifconfig bond0 192.168.100.100 netmask 255.255.255.0 up
ifenslave bond0 eth0 eth1
```

```
[root@localhost ~]# chmod +x /etc/init.d/S98bonding
```

실행 결과

```
[root@localhost ~]# ifconfig bond0 192.168.100.100 netmask 255.255.255.0
[ 89.611815] IPv6: ADDRCONF(NETDEV_UP): bond0: link is not ready
[root@localhost ~]# ifenslave bond0 eth0 eth1
[ 94.509449] net eth0: initializing cpsw version 1.12 (0)
[ 94.593531] net eth0: phy found : id is : 0x221560
[ 94.610099] bond0: Enslaving eth0 as an active interface with an up link
[ 94.619283] IPv6: ADDRCONF(NETDEV_CHANGE): bond0: link becomes ready
[ 94.661503] net eth1: initializing cpsw version 1.12 (0)
[ 94.743545] net eth1: phy found : id is : 0x221560
[ 94.770812] bond0: Enslaving eth1 as an active interface with an up link
[root@localhost ~]# [ 97.593692] cpsw 4a100000.ethernet eth0: Link is Up - 100Mbps/Full - flow control rx/tx
```

Bonding 확인 방법

```
[root@localhost ~]# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.7.1 (April 27, 2011)
```

Bonding Mode: load balancing (round-robin)

MII Status: up

MII Polling Interval (ms): 0

Up Delay (ms): 0

Down Delay (ms): 0

**Slave Interface: eth0**

MII Status: up

Speed: 100 Mbps

Duplex: full

Link Failure Count: 0

Permanent HW addr: 60:64:05:28:f1:be

Slave queue ID: 0

**Slave Interface: eth1**

MII Status: up

Speed: 100 Mbps

Duplex: full

Link Failure Count: 0

Permanent HW addr: 60:64:05:28:f1:c0

Slave queue ID: 0

```
[root@localhost ~]# ifconfig -a
```

```
bond0    Link encap:Ethernet  HWaddr 60:64:05:28:FD:2C  
          inet addr:192.168.100.100  Bcast:192.168.100.255  Mask:255.255.255.0  
          inet6 addr: fe80::6264:5ff:fe28:fd2c/64 Scope:Link  
          UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500  Metric:1  
          RX packets:7 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:12 errors:0 dropped:6 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:1480 (1.4 KiB)  TX bytes:1910 (1.8 KiB)  
  
eth0     Link encap:Ethernet  HWaddr 60:64:05:28:FD:2C  
          UP BROADCAST SLAVE 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)
Interrupt:172
```

```
eth1    Link encap:Ethernet HWaddr 60:64:05:28:FD:2C
        inet addr:192.168.100.28 Bcast:192.168.100.255 Mask:255.255.255.0
        UP BROADCAST RUNNING SLAVE MULTICAST MTU:1500 Metric:1
        RX packets:18 errors:0 dropped:0 overruns:0 frame:0
        TX packets:39 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:3700 (3.6 KiB) TX bytes:5203 (5.0 KiB)
```

ping으로 테스트 방법

Linux PC에서 ping을 한다.

```
$ ping 192.168.100.100
```

보드에 이더넷을 ETH0 포트에 삽입한다.

Linux PC에서 ping 통신이 되는지 확인한다.

이더넷을 빼서 ETH1에 삽입한다.

Linux PC에서 IP 192.168.100.100으로 ping 통신이 되는지 확인한다.

```
[icanjji@icanjji-Samsung-DeskTop-System image]$ ping 192.168.100.100
PING 192.168.100.100 (192.168.100.100) 56(84) bytes of data:
64 bytes from 192.168.100.100: icmp_req=1 ttl=64 time=0.390 ms
64 bytes from 192.168.100.100: icmp_req=2 ttl=64 time=0.190 ms
64 bytes from 192.168.100.100: icmp_req=3 ttl=64 time=0.235 ms
64 bytes from 192.168.100.100: icmp_req=4 ttl=64 time=0.179 ms
64 bytes from 192.168.100.100: icmp_req=5 ttl=64 time=0.172 ms
64 bytes from 192.168.100.100: icmp_req=10 ttl=64 time=0.413 ms
64 bytes from 192.168.100.100: icmp_req=11 ttl=64 time=0.240 ms
64 bytes from 192.168.100.100: icmp_req=12 ttl=64 time=0.157 ms
64 bytes from 192.168.100.100: icmp_req=13 ttl=64 time=0.278 ms
64 bytes from 192.168.100.100: icmp_req=14 ttl=64 time=0.201 ms
64 bytes from 192.168.100.100: icmp_req=15 ttl=64 time=0.160 ms
64 bytes from 192.168.100.100: icmp_req=16 ttl=64 time=0.193 ms
64 bytes from 192.168.100.100: icmp_req=17 ttl=64 time=0.193 ms
64 bytes from 192.168.100.100: icmp_req=18 ttl=64 time=0.280 ms
```

## 5. 레퍼런스 링크

<http://www.packetinside.com/2012/07/bonding.html>