Mango-AM335x-ST NAND 부팅 하기

http://www.mangoboard.com/
http://cafe.naver.com/embeddedcrazyboys
Crazy Embedded Laboratory

Document History

Revision	Date	Change note
Init	2015-07-31	전종인

1. NAND Test 4
2. NAND U-Boot 작업 5
3. File System 작업 7
4. Kernel에서 NAND 이미지 작업 10

1. NAND Test

[root@mangoboard ~]# nandtest /dev/mtd0
ECC corrections: 0
ECC failures : 0
Bad blocks : 0
BBT blocks : 0
00000000: checking...
Finished pass 1 successfully
[root@mangoboard ~]# nandtest /dev/mtd1
ECC corrections: 0
ECC failures : 0
Bad blocks : 0
BBT blocks : 0
00000000: checking...
Finished pass 1 successfully
[root@mangoboard ~]# nandtest /dev/mtd2
ECC corrections: 0
ECC failures : 0
Bad blocks : 0
BBT blocks : 0
00000000: checking...
Finished pass 1 successfully
[root@mangoboard ~]# nandtest /dev/mtd3
ECC corrections: 0
ECC failures : 0
Bad blocks : 0
BBT blocks : 0
00000000: checking...
Finished pass 1 successfully
[root@mangoboard ~]# nandtest /dev/mtd4
ECC corrections: 0
ECC failures : 0
Bad blocks : 0
001a0000: checking...
001c0000: checking...

```
Finished pass 1 successfully
[root@mangoboard ~]# nandtest /dev/mtd5
ECC corrections: 0
ECC failures     : 0
Bad blocks       : 0
BBT blocks       : 0
00000000: checking...
Finished pass 1 successfully
[root@mangoboard ~]# nandtest /dev/mtd6
ECC corrections: 0
ECC failures     : 0
Bad blocks       : 0
BBT blocks       : 0
004e0000: checking...
Finished pass 1 successfully
[root@mangoboard ~]# nandtest /dev/mtd7
ECC corrections: 0
ECC failures     : 0
Bad blocks       : 9
BBT blocks       : 0
Bad block at 0x01660000
Bad block at 0x01880000
Bad block at 0x03460000
Bad block at 0x06160000
Bad block at 0x07a40000
Bad block at 0x095a0000
0c980000: reading...
 1 bit(s) ECC corrected at 0c980000
Bad block at 0x0cec0000
Bad block at 0x0e200000
Bad block at 0x0e680000
0f860000: checking...
Finished pass 1 successfully
```

2. NAND U-Boot 작업

```
U-Boot# mtdparts
```

```
device nand0 <omap2-nand.0>, # parts = 8
 #: name                 size            offset          mask_flags
 0: SPL                  0x00020000      0x00000000      0
 1: SPL.backup1          0x00020000      0x00020000      0
 2: SPL.backup2          0x00020000      0x00040000      0
 3: SPL.backup3          0x00020000      0x00060000      0
 4: u-boot               0x001e0000      0x00080000      0
 5: u-boot-env           0x00020000      0x00260000      0
 6: kernel               0x00500000      0x00280000      0
 7: rootfs               0x0f880000      0x00780000      0
active partition: nand0,0 - (SPL) 0x00020000 @ 0x00000000
defaults:
mtdids  : nand0=omap2-nand.0
mtdparts: mtdparts=omap2-
nand.0:128k(SPL),128k(SPL.backup1),128k(SPL.backup2),128k(SPL.backup3),1920k(u-boot),128k(u-boot-
env),5m(kernel),-(rootfs)
U-Boot#

Creating 8 MTD partitions on "omap2-nand.0":
0x000000000000-0x000000020000 : "SPL"
0x000000020000-0x000000040000 : "SPL.backup1"
0x000000040000-0x000000060000 : "SPL.backup2"
0x000000060000-0x000000080000 : "SPL.backup3"
0x000000080000-0x000000260000 : "U-Boot"
0x000000260000-0x000000280000 : "U-Boot Env"
0x000000280000-0x000000780000 : "Kernel"
0x000000780000-0x000010000000 : "File System"

-->0x00000000-> SPL start (SPL copy on 1st block)
| |-->0x0001FFFF-> SPL end
| |-->0x00020000-> SPL.backup1 start (SPL copy on 2nd block)
| |-->0x0003FFFF-> SPL.backup1 end
| |-->0x00040000-> SPL.backup2 start (SPL copy on 3rd block)
| |-->0x0005FFFF-> SPL.backup2 end
| |-->0x00060000-> SPL.backup3 start (SPL copy on 4th block)
| |-->0x0007FFFF-> SPL.backup3 end
| |-->0x00080000-> U-Boot start
| |-->0x002BFFFF-> U-Boot end
```

| |-->0x00260000-> ENV start
| |-->0x0027FFFF-> ENV end
| |-->0x00280000-> Linux Kernel start
| |-->0x0077FFFF-> Linux Kernel end
| |-->0x00780000-> File system start
+------------+-->0x10000000-> NAND end (Free end)

nand scrub.chip
mmc rescan
fatload mmc 0 ${loadaddr} MLO
nand erase 0x0 0x20000
nand write ${loadaddr} 0x0 0x20000
fatload mmc 0 ${loadaddr} u-boot.img
nand erase 0x80000 0x1e00000
nand write ${loadaddr} 0x80000 0x1e0000

부트 스위치 #3, #4를 On하면 NAND mode 부팅이다.
일단 U-boot까지 부팅하는 것은 성공하였다.

mmc rescan
fatload mmc 0 ${loadaddr} uImage
nand erase 0x280000 0x500000
nand write ${loadaddr} 0x280000 0x500000

위 작업까지 수행한 후 Kernel까지 부팅하는 것은 성공하였다.

3. File System 작업

[root@mangoboard ~]# ./nfs_mount.sh
[root@mangoboard ~]# l nfs_mount
-rw-rw-r-- 1 1001 1001 80013 Feb 24 15:26 custom_datafs.tar.gz
-rw-rw-r-- 1 1001 1001 19781015 Feb 24 15:26 rootfs.tar.gz
[root@mangoboard ~]# **cp nfs_mount/custom_datafs.tar.gz .**
[root@mangoboard ~]# **cp nfs_mount/rootfs.tar.gz .**
[root@mangoboard ~]# gunzip custom_datafs.tar.gz
[root@mangoboard ~]# gunzip rootfs.tar.gz

```
[root@mangoboard ~]# sync

[root@mangoboard ~]# cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00020000 00020000 "SPL"
mtd1: 00020000 00020000 "SPL.backup1"
mtd2: 00020000 00020000 "SPL.backup2"
mtd3: 00020000 00020000 "SPL.backup3"
mtd4: 001e0000 00020000 "U-Boot"
mtd5: 00020000 00020000 "U-Boot Env"
mtd6: 00500000 00020000 "Kernel"
mtd7: 0f880000 00020000 "File System"

[root@mangoboard ~]# flash_erase /dev/mtd7 0 0
Erasing 128 Kibyte @ 1640000 --  8 % complete flash_erase: Skipping bad block at 01660000
Erasing 128 Kibyte @ 1860000 --  9 % complete flash_erase: Skipping bad block at 01880000
Erasing 128 Kibyte @ 3440000 -- 21 % complete flash_erase: Skipping bad block at 03460000
Erasing 128 Kibyte @ 6140000 -- 39 % complete flash_erase: Skipping bad block at 06160000
Erasing 128 Kibyte @ 7a20000 -- 49 % complete flash_erase: Skipping bad block at 07a40000
Erasing 128 Kibyte @ 9580000 -- 60 % complete flash_erase: Skipping bad block at 095a0000
Erasing 128 Kibyte @ cea0000 -- 83 % complete flash_erase: Skipping bad block at 0cec0000
Erasing 128 Kibyte @ e1e0000 -- 90 % complete flash_erase: Skipping bad block at 0e200000
Erasing 128 Kibyte @ e660000 -- 92 % complete flash_erase: Skipping bad block at 0e680000
Erasing 128 Kibyte @ f860000 -- 100 % complete

[root@mangoboard ~]# ubiattach /dev/ubi_ctrl -m 7
UBI: attaching mtd7 to ubi0
UBI: physical eraseblock size:    131072 bytes (128 KiB)
UBI: logical eraseblock size:     126976 bytes
UBI: smallest flash I/O unit:     2048
UBI: VID header offset:           2048 (aligned 2048)
UBI: data offset:                 4096
UBI: empty MTD device detected
UBI: max. sequence number:        0
UBI: create volume table (copy #1)
UBI: create volume table (copy #2)
UBI: attached mtd7 to ubi0
UBI: MTD device name:             "File System"
```

UBI: MTD device size: 248 MiB
UBI: number of good PEBs: 1979
UBI: number of bad PEBs: 9
UBI: number of corrupted PEBs: 0
UBI: max. allowed volumes: 128
UBI: wear-leveling threshold: 4096
UBI: number of internal volumes: 1
UBI: number of user volumes: 0
UBI: available PEBs: 1956
UBI: total number of reserved PEBs: 23
UBI: number of PEBs reserved for bad PEB handling: 19
UBI: max/mean erase counter: 0/0
UBI: image sequence number: 1314679700
UBI: background thread "ubi_bgt0d" started, PID 959
UBI device number 0, total 1979 LEBs (251285504 bytes, 239.6 MiB), available 1956 LEBs (248365056 bytes, 236.9 MiB), LEB size 126976 bytes (124.0 KiB)

[root@mangoboard ~]# **ubimkvol /dev/ubi0 -N rootfs -m -n 0**
Set volume size to 248365056
Volume ID 0, size 1956 LEBs (248365056 bytes, 236.9 MiB), LEB size 126976 bytes (124.0 KiB), dynamic, name "rootfs", alignment 1

[root@mangoboard ~]# **mkdir -p /mnt/nand_rootfs**
[root@mangoboard ~]# **mount -t ubifs ubi0:rootfs /mnt/nand_rootfs**
UBIFS: default file-system created
UBIFS: mounted UBI device 0, volume 0, name "rootfs"
UBIFS: file system size: 246714368 bytes (240932 KiB, 235 MiB, 1943 LEBs)
UBIFS: journal size: 12316672 bytes (12028 KiB, 11 MiB, 97 LEBs)
UBIFS: media format: w4/r0 (latest is w4/r0)
UBIFS: default compressor: lzo
UBIFS: reserved for root: 4952683 bytes (4836 KiB)
[root@mangoboard ~]# **cd /mnt/nand_rootfs/**
[root@mangoboard nand_rootfs]# **tar xvf ~/custom_datafs.tar**
[root@mangoboard nand_rootfs]# **tar xvf ~/rootfs.tar**

```
flash_erase /dev/mtd7 0 0
ubiattach /dev/ubi_ctrl -m 7
ubimkvol /dev/ubi0 -N rootfs -m -n 0
```

```
mkdir -p /mnt/nand_rootfs
mount -t ubifs ubi0:rootfs /mnt/nand_rootfs
cd /mnt/nand_rootfs/
[root@mangoboard nand_rootfs]# tar xvf ~/custom_datafs.tar
[root@mangoboard nand_rootfs]# tar xvf ~/rootfs.tar
sync
```

위 과정을 통해서 파일시스템까지 NAND 부팅 완료

4. Kernel에서 NAND 이미지 작업

```
[root@localhost ~]# ./flash_eraseall /dev/mtd0
flash_eraseall has been replaced by `flash_erase <mtddev> 0 0`; please use it
Erasing 128 Kibyte @ 0 -- 100 % complete
[root@localhost ~]# ./flash_eraseall /dev/mtd1
[root@localhost ~]# ./flash_eraseall /dev/mtd2
[root@localhost ~]# ./flash_eraseall /dev/mtd3
[root@localhost ~]# ./flash_eraseall /dev/mtd4
[root@localhost ~]# ./flash_eraseall /dev/mtd5
[root@localhost ~]# ./flash_eraseall /dev/mtd6
[root@localhost ~]# ./flash_eraseall /dev/mtd7

[root@localhost ~]# mount /dev/mmcblk0p1 /root/boot_dir/
[root@localhost ~]# cd boot_dir/

[root@mangoboard ~]# cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00020000 00020000 "SPL"
mtd1: 00020000 00020000 "SPL.backup1"
mtd2: 00020000 00020000 "SPL.backup2"
mtd3: 00020000 00020000 "SPL.backup3"
mtd4: 001e0000 00020000 "U-Boot"
mtd5: 00020000 00020000 "U-Boot Env"
mtd6: 00500000 00020000 "Kernel"
mtd7: 0f880000 00020000 "File System"
```

```
nandwrite -p /dev/mtd0 MLO
nandwrite -p /dev/mtd4 u-boot.img
nandwrite -p /dev/mtd6 uImage

ubiattach /dev/ubi_ctrl -m 7
ubimkvol /dev/ubi0 -N rootfs -m -n 0
mkdir -p /mnt/nand_rootfs
mount -t ubifs ubi0:rootfs /mnt/nand_rootfs
cd /mnt/nand_rootfs/
[root@mangoboard nand_rootfs]# tar xvf ~/custom_datafs.tar
[root@mangoboard nand_rootfs]# tar xvf ~/rootfs.tar
sync
```

tar xvf /update_image/**custom_datafs.tar**

tar xvf /update_image/**rootfs.tar**

sync