망고 210 ICS mmc booting 메뉴얼 작성 및 patch

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

Document History

Revision	Date	Change note

1. android mmc build

android/device/crazyboys/mango210$ vi BoardConfig.mk

```
#false select nand, true select sd
TARGET_USERIMAGES_USE_EXT4 := true
#TARGET_USERIMAGES_USE_EXT4 := false

ifeq ($(TARGET_USERIMAGES_USE_EXT4),true)
TARGET_USERIMAGES_SPARSE_EXT_DISABLED := true
#TARGET_USERIMAGES_SPARSE_EXT_DISABLED := false
#BOARD_SYSTEMIMAGE_PARTITION_SIZE := 268435456
BOARD_SYSTEMIMAGE_PARTITION_SIZE := 241172480
#BOARD_USERDATAIMAGE_PARTITION_SIZE := 367001600
BOARD_USERDATAIMAGE_PARTITION_SIZE := 241172480
BOARD_FLASH_BLOCK_SIZE := 4096
endif

ifeq ($(TARGET_USERIMAGES_USE_EXT4),false)
#INTERNAL_USERIMAGES_USE_EXT := false
BOARD_NAND_PAGE_SIZE := 2048
BOARD_NAND_SPARE_SIZE := 64
endif
```

extended file system 4 , 확장된 파일 시스템 4를 사용하며 설정부분입니다.

android/device/crazyboys/mango210$ vi init.rc

```
on fs
# mount mtd partitions
    # Mount /system rw first to give the filesystem a chance to save a checkpoint
#    mount yaffs2 mtd@system /system
#    mount yaffs2 mtd@system /system rw remount
#    mount yaffs2 mtd@userdata /data   nosuid nodev
#    mount yaffs2 mtd@cache /cache   nosuid nodev
#     mount ubifs ubi0:system /system
#         mount ubifs ubi0:system /system rw remount
#         mount ubifs ubi2:userdata /data nosuid nodev
#         mount ubifs ubi1:cache /cache nosuid nodev
```

yaffs2는 낸드 사용시 마운트 합니다.

android/device/crazyboys/mango210$ vi init.mango210.rc

```
        mount ext4 partitions
        mount ext4 /dev/block/mmcblk0p2 /system wait rw
        mount ext4 /dev/block/mmcblk0p3 /data wait rw noatime nosuid nodev
        mount ext4 /dev/block/mmcblk0p4 /cache wait rw noatime nosuid nodev
         export EXTERNAL_STORAGE /mnt/sdcard
         export SECONDARY_STORAGE /mnt/ext_sd:/mnt/usb
```

위와 같이 변경 후 ./android_build.sh를 실행합니다.

2. kernel build

```
## 7inch 800x480 LCD(감압식)
$ ./build_kernel defconfig mango210_7inch_nand256MB_wifi8787_defconfig
$ ./build_kernel
```

3. sdwriter

현재 sd카드에 이미지를 라이팅 하기 위해서는

fastboot 를 이용해서 보드에 sd카드를 넣고 아래와 같이 라이팅 합니다.

```
sudo ./fastboot flash kernel zImage;
sudo ./fastboot flash system system.img;
sudo ./fastboot flash ramdisk ramdisk-uboot.img ;
sudo ./fastboot -w
```

위와 같은 불편함을 sdwriter명령어로만 바꾸기 위해 아래와 같이 변경합니다.

android₩android_build.sh

```
rm -rf rootfs
mkdir rootfs
cp -a out/target/product/$SEC_PRODUCT/root/* ./rootfs
cp -a out/target/product/$SEC_PRODUCT/data ./rootfs
cp -a out/target/product/$SEC_PRODUCT/system ./rootfs

##by crazboys 20140322
cd rootfs/system
```

```
tar zcvf ../../../image/mango210_system.tgz .
cd ..

cd data
tar zcvf ../../../image/mango210_data.tgz .
cd ../../
```

아래와 같이 image/sdwriter을 변경했습니다.

```
#!/bin/bash
# Mango T-Flash Writer tool

LC_ALL="C"
TFLASH=/dev/$1
BOARD=$2
OPT=$3

BL1_210=mango210_bl1.bin
UBOOT_210=u-boot.bin
KERNEL_210=zImage
ANDROID_210=mango210_ics.tgz
SYSTEM_210=mango210_system.tgz
DATA_210=mango210_data.tgz
RAMDISK_210=ramdisk-uboot.img

UBOOT_100=mango100_uboot.bin
KERNEL_100=mango100_zImage
ANDROID_100=mango100_eclair.tgz
GNOME=mango_gnome.tgz

echo "Mango SD Writer V1.0"

print_success()
{
	if [ "$1" == 0 ]; then
		echo "success"
	else
		echo "failed"
		exit -1
```

```
	fi
}

print_usage()
{
	echo "Usage: $0 device board <options: bin|android|gnome|all>"
	echo "options:"
	echo "	bin: wirte bootloader and kernel"
	echo "	android: wirte android file system"
	echo "	gnome: write gnome file system "
	echo "	all or no option: wirte all"
	echo "ex) $0 sdb 210"
	echo "ex) $0 sdb 210 android"
	exit 1
}

# Check args
check_args()
{
if [ ! -b "$TFLASH" ]; then
	print_usage
fi

# Check TFlash Sectors
TFLASH_SECTORS=`fdisk -l -u $TFLASH | grep sectors | head -n 1 ₩
| cut -d',' -f4 | cut -d' ' -f3`

# Boot images and ANDROID, GNOME Partition Size
#	Boot images(bootloader + Kernel): 10 MB
#	GONME : 512 MB
#	Android : 512 MB
SIZE_BINARY=`expr 2048 ₩* 10`
SIZE_GNOME=`expr 2048 ₩* 512`
SIZE_ANDROID_100=`expr 2048 ₩* 512`
SIZE_ANDROID_210=`expr 2048 ₩* 512`

case "$BOARD" in
	100)
```

```
	SIZEOF_BL1_100=18
	OFFSET_BL1_100=$(($TFLASH_SECTORS-$SIZEOF_BL1_100))
	SIZEOF_BL2_100=1024
	OFFSET_BL2_100=$(($OFFSET_BL1_100-$SIZEOF_BL2_100))
	SIZEOF_KERNEL_100=`ls -s --block-size=512 $KERNEL_100 | cut -d' ' -f1`
	OFFSET_KERNEL_100=$(($OFFSET_BL2_100-$SIZEOF_KERNEL_100))
	SIZE_FAT=$(($TFLASH_SECTORS-$SIZE_BINARY-$SIZE_GNOME-$SIZE_ANDROID_100-2))

	START_FAT=12
	START_GNOME=$(($START_FAT+$SIZE_FAT))
	START_ANDROID_100=$(($START_GNOME+$SIZE_GNOME))
	;;

	210)
	OFFSET_BL1_210=1
	SIZEOF_BL1_210=48
	OFFSET_BL2_210=$(($OFFSET_BL1_210+$SIZEOF_BL1_210))
	SIZEOF_BL2_210=1024
	OFFSET_KERNEL_210=$(($OFFSET_BL2_210+$SIZEOF_BL2_210))
	SIZEOF_KERNEL_210=`ls -s --block-size=512 $KERNEL_210 | cut -d' ' -f1`
		OFFSET_RAMDISK_210=11313

	SIZE_FAT=$(($TFLASH_SECTORS-$SIZE_BINARY-$SIZE_GNOME-$SIZE_ANDROID_210-2))
	START_FAT=$SIZE_BINARY
	START_GNOME=$(($START_FAT+$SIZE_FAT))
	START_ANDROID_210=$(($START_GNOME+$SIZE_GNOME))
	;;

	*)
	print_usage
	;;
esac
}

make_bl1()
{
case "$BOARD" in
	100)
```

```
	./mkbl1 mango100_uboot.bin mango100_bl1.bin 8192
	;;

	210)
	./mkbl1 u-boot.bin mango210_bl1.bin 8192
	;;

	*)
	echo "Make BL1: Error !!!"
	;;
esac
}

partition_add()
{
	echo n
	echo p
	echo $1
	echo $2
	echo $3
}

sdcard_format()
{
	(

# Pre Partition Delete
		echo d
		echo 6
		echo d
		echo 5
		echo d
		echo 4
		echo d
		echo 3
		echo d
		echo 2
		echo d
```

```
# Partition Create

        partition_add 1 1438200 15523839
        partition_add 2 30600 504899
        partition_add 3 504900 1223999
        partition_add 4 1224000 1438199

        echo t
        echo 1
        echo c
        echo w
        echo q
    ) | fdisk -u $TFLASH    > /dev/null 2>&1
}

umount_all()
{
echo
echo -n "Unmount all : "
DEV_MOUNT=`df | grep $1 | awk '{ print $6 }'`
for i in $DEV_MOUNT; do
umount $i
done

if [ "$?" == 0 ]; then
        echo "success"
else
        echo "not Mounted"
fi
}

write_bin()
{
case "$BOARD" in
    100)
    echo
    echo -n "Write Mango100 BL1 : "
```

```
	dd bs=512 seek=$OFFSET_BL1_100 if=$BL1_100 of=$TFLASH > /dev/null 2>&1
	print_success "$?"
	echo -n "Write Mango100 Uboot : "
	dd bs=512 seek=$OFFSET_BL2_100 if=$UBOOT_100 of=$TFLASH > /dev/null 2>&1
	print_success "$?"
	echo -n "Write Mango100 Kernel : "
	dd bs=512 seek=$OFFSET_KERNEL_100 if=$KERNEL_100 of=$TFLASH > /dev/null 2>&1
	print_success "$?"
	;;

	210)
	echo
	echo -n "Write Mango210 BL1 : "
	dd bs=512 seek=$OFFSET_BL1_210 if=$BL1_210 of=$TFLASH > /dev/null 2>&1
	print_success "$?"
	echo -n "Write Mango210 Uboot : "
	dd bs=512 seek=$OFFSET_BL2_210 if=$UBOOT_210 of=$TFLASH > /dev/null 2>&1
	print_success "$?"
	echo -n "Write Mango210 Kernel : "
	dd bs=512 seek=$OFFSET_KERNEL_210 if=$KERNEL_210 of=$TFLASH > /dev/null 2>&1
	print_success "$?"
	echo -n "Write Mango210 Ramdisk : "
	dd bs=512 seek=$OFFSET_RAMDISK_210 if=$RAMDISK_210 of=$TFLASH > /dev/null 2>&1
	print_success "$?"
	;;

	*)
	echo "Make Bootloader and Kernel: Error !!!"
	;;
esac
}

write_gnome()
{
echo
echo -n "Gnome Filesystem Create : "
mkdir temp
mkfs.ext3 "$TFLASH"2 -L gnome > /dev/null 2>&1
```

```
mount "$TFLASH"2 temp
cd temp
tar zxvf ../$GNOME > /dev/null 2>&1
cd ..
sync
umount temp
rm -rf temp
print_success "$?"
}

#write_android()
#{
#mkdir temp4
#echo
#echo -n "Android Filesystem Create : "
#mkfs.ext3 "$TFLASH"3 -L android > /dev/null 2>&1
#mount "$TFLASH"3 temp4
#cd temp4
#if [ "$BOARD" == "100" ]; then
#tar zxvf ../$ANDROID_100 > /dev/null 2>&1
#else
#tar zxvf ../$ANDROID_210 > /dev/null 2>&1
#fi
#cd ..
#sync
#umount temp4
#rm -rf temp4
#print_success "$?"
#}

#by crazboys 20140322
write_android()
{
mkfs.vfat "$TFLASH"1 -L mango > /dev/null 2>&1

mkdir temp4
echo
echo -n "Android System Create : "
```

```
mkfs.ext4 "$TFLASH"2 -L system > /dev/null 2>&1
mount "$TFLASH"2 temp4
cd temp4
tar zxvf ../$SYSTEM_210 > /dev/null 2>&1
chmod 755 bin/*
chmod 755 xbin/*
chmod 755 vendor/bin/*
chown root:root *
cd ..
sync

umount temp4
rm -rf temp4
print_success "$?"

echo
echo -n "Android Data Create : "
mkdir temp
mkfs.ext4 "$TFLASH"3 -L data > /dev/null 2>&1
mount "$TFLASH"3 temp
cd temp
tar zxvf ../$DATA_210 > /dev/null 2>&1
cd ..
sync
umount temp
rm -rf temp
print_success "$?"

mkfs.ext4 "$TFLASH"4 -L cache > /dev/null 2>&1
sync
}

check_args
umount_all $1
case "$OPT" in
        bin)
        make_bl1
        write_bin
```

```
	;;

	android)
	write_android
	;;

	gnome)
	write_gnome
	;;

	format)
	sdcard_format
	;;

	all|*)
	echo
	echo -n "Partition Create : "
	sdcard_format
	print_success "$?"
	make_bl1
	write_bin
	mkfs.vfat "$TFLASH"1 -n mango > /dev/null 2>&1
	write_android
#	write_gnome
	;;
esac
umount_all $1

echo
echo "Success"
```

4. sd 부팅 방법

SD부팅을 하기 위해 SD카드에 이미지를 퓨징 하겠습니다.

SD카드를 리더기에 삽입 후 linux pc에 삽입

#df
명령으로 디바이스 확인

```
# cd image

$ dmesg | tail
[12403.632015] usb 2-5: new high-speed USB device number 25 using ehci_hcd
[12403.856263] hub 2-0:1.0: unable to enumerate USB device on port 5
[12483.752014] usb 2-5: new high-speed USB device number 26 using ehci_hcd
[12483.976252] hub 2-0:1.0: unable to enumerate USB device on port 5
[12513.801490] sd 9:0:0:0: [sdd] 15644672 512-byte logical blocks: (8.01 GB/7.45 GiB)
[12513.802983] sd 9:0:0:0: [sdd] No Caching mode page present
[12513.802986] sd 9:0:0:0: [sdd] Assuming drive cache: write through
[12513.807109] sd 9:0:0:0: [sdd] No Caching mode page present
[12513.807113] sd 9:0:0:0: [sdd] Assuming drive cache: write through

[12513.808249]  sdd: sdd1 sdd2 sdd3 sdd4
```

위와 같이 dmesg | tail로 값을 확인 할 수 있습니다.
개별로 맞는 값을 확인해서 사용하시면 됩니다.

```
cd image
sudo ./sdwriter sdd 210
```

부트 이미지를 가진 sd카드를 SDIO 0/BOOT에 삽입합니다.
sd 부팅을 합니다.

```
mango210 보드 1, 3, 4 ON
CM-V210 보드 2, 3, 6 ON
```

usb otg, 3PIN 연결 후 전원을 인가합니다.

부팅 되는 것을 볼 수 있습니다.