

Mango24R2 Auto Write

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

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

Crazy Embedded Laboratory

Document History

Revision	Date	Change note

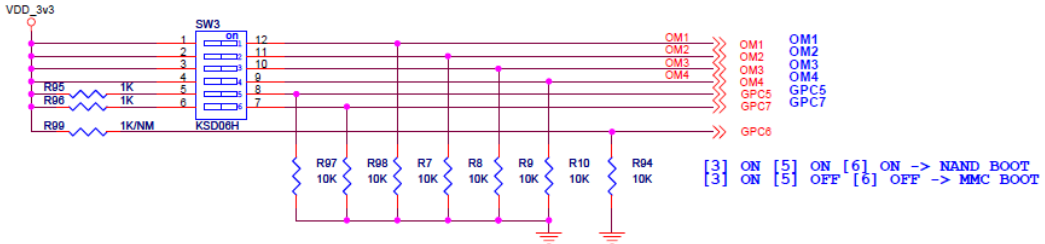
1. Auto write	4
1.1. uboot 수정	4
2. key1,2로 7 inch, 10.4 inch LCD선택하여 NAND flash write.....	9
3. LCD화면에 writing, write 완료 이미지 출력하기.....	16

1. Auto write

1.1. uboot 수정

uboot를 보면 아래와 같이 GPCDAT_REG 로 난드 모드인지 구별한다.

BOOT Option Switch



[4]		[5] : ON & [4] : OFF		OFF		
[3]		OneNAND/ROM		NAND/iROM/Security		
				OFF	ON	
				NAND	/iROM/Security	
				Advanced	Normal	
[2]	OFF	OFF	ON	page 4KB	ON	OFF
	ON	Muxed OneNAND	ROM / Demuxed OneNAND	page 2KB	NAND page 512B	iROM/Security
[1]	OFF		8 bit	ADDR4	ADDR3	OFF
	ON	16 bit	16 bit	ADDR5	ADDR4	iROM Security eFUSE

부트 스위치에서 5번과 6번에 의해서 난드 부팅과 MMC부팅으로 선택됩니다.

GPC5와 GPC7번을 확인합니다.

S3C2450매뉴얼을 보면

To use NAND Flash Device, The OM and the **GPC5/6/7** configuration should be set to use IROM boot and select proper nand device type. Nand Boot written below is boot device in IROM boot. Refer to IROM application Note for more information. S3C2450 supports nand boot by using IROM boot mode.

5 GPC5/6/7 PIN CONFIGURATION TABLE IN IROM BOOT MODE

	Page	Address Cycle	GPC7 [2]	GPC6 [1]	GPC5 [0]
MMC(MoviNAND/iNand)	-	-	0	0	0
Reserved	-	-	0	0	1
Nand	512	3	0	1	0
		4	0	1	1
	2048	4	1	0	0
		5	1	0	1
4096	5	1	1	0	

위와 같이 부트 스위치를 통해 NAND모드로 부팅합니다.

비트 분석기로 계산해보겠습니다.

The screenshot shows a bit analysis tool interface. At the top, a 32-bit register value is displayed as a sequence of bits: 00000000000000000000000011000000. The bits are grouped into nibbles (4 bits each) labeled 31, 24/23, 16/15, 8/7, and 0. The 8th and 7th bits are highlighted with red boxes and labeled 'A'. Below the bit display, the size information is shown as [160 Byte|0.16 KByte|0.00 MByte]. At the bottom, the result data is shown as HEX: 000000A0, DEC: 160, with TEST and Previous buttons.

“A0”입니다.

uboot\$ vi lib_arm/board.c

```
565 #if 1 /* Delete this code for NFS */
566     /* by CRZ, boot cmd, args setting */
567     if((GPCDAT_REG & 0xA0) == 0xA0)
568     {
569         setenv("bootcmd", CONFIG_NAND_BOOTCOMMAND);
570         setenv("bootargs", CONFIG_NAND_BOOTARGS);
571     }
572     else
573     {
574         setenv("bootcmd", CONFIG_MMC_BOOTCOMMAND);
575         setenv("bootargs", CONFIG_MMC_BOOTARGS);
576     }
577 #endif
```

if((GPCDAT_REG & 0xA0) == 0xA0) <<이부분이 AND연산하여 동일하다면 NAND부팅 모드입니다.
아래에서 각각 bootcmd를 설정합니다.

uboot\$ vi include/configs/mango2450.h

```
363 #define CONFIG_NAND_BOOTCOMMAND "nand read 0xC0008000 0x80000 0x480000;bootm 0xC0008000"
364 #define CONFIG_NAND_BOOTARGS "noinitrd console=ttySAC1,115200 rw ubi.mtd=2 root=ubi0:rootfs rootfstype=ubifs rootwait"
365 //#elif defined(CONFIG_BOOT_MOVINAND)
366 //#define CFG_ENV_IS_IN_MOVINAND
367 //#define CONFIG_BOOTCOMMAND "sleep 1;nand scrub ;sleep 1;nand erase;sleep 1;movi read 3800000# 40000 c0000000 ;sleep 1;nand write c0000000 0 40000;sleep 1;movi read kernel c0000000 ; sleep 1;nand write c0000000 40000 200000;sleep 1;movi read 3600000# 3800000 c0000000;sleep 1;nand write.yaffs c0000000 400000 35db4c0"
```

```

368 #define CONFIG_MMC_BOOTCOMMAND "nand scrub; movi read u-boot C1000000; nand erase
0 40000; nand write C1000000 0 40000; movi read kernel C1000000; nand erase 80000 400000; nand
write C 1000000 80000 400000; movi read kernel 0xC0008000; bootm 0xC0008000"
369
370 #define CONFIG_MMC_BOOTARGS "noinitrd console=ttySAC1,115200 rw root=/dev/mmcbk0p2
rootfstype=ext3 rootwait"

```

nand scrub 입력 시 "y" 입력 안해도 되게 변경합니다.

```

418     if (scrub) {
419         skip = argc > 2 && !strcmp("y", argv[2]);
420         puts("Warning: "
421             "scrub option will erase all factory set "
422             "bad blocks!\n"
423             "          "
424             "There is no reliable way to recover them.\n"
425             "          "
426             "Use this command only for testing purposes "
427             "if you\n"
428             "          "
429             "are sure of what you are doing!\n"
430             "\nReally scrub this NAND flash? <y/N>\n");
431 #if 0
432         if (skip) {
433             opts.scrub = 1;
434         } else if (getc() == 'y' && getc() == '\nr') {
435             opts.scrub = 1;
436         } else {
437             puts("scrub aborted\n");
438             return -1;
439         }
440     }
441
442 #else
443     opts.scrub = 1;
444 #endif

```

변경 전

/uboot\$ vi include/configs/mango2450.h

```
#define CONFIG_NAND_BOOTCOMMAND "nand read 0xC0008000 0x80000 0x480000;bootm
0xC0008000"
363 #define CONFIG_NAND_BOOTARGS      "noinitrd console=ttySAC1,115200 rw ubi.mtd=2
root=ubi0:rootfs rootfstype=ubifs rootwait"
364 //#elif defined(CONFIG_BOOT_MOVINAND)
365 //#define CFG_ENV_IS_IN_MOVINAND
366 //#define CONFIG_BOOTCOMMAND      "sleep 1;nand scrub ;sleep 1;nand erase;sleep 1;movi
read 3800000# 40000 c0000000 ;sleep 1;nand write c0000000 0 40000;sleep 1;movi read kernel
c0000000;    sleep 1;nand write c0000000 40000 200000;sleep 1;movi read 3600000# 3800000
c0000000;sleep 1;nand write.yaffs c0000000 400000 35db4c0"
367 #define CONFIG_MMC_BOOTCOMMAND "movi read kernel 0xC0008000; bootm 0xC0008000"
368 #define CONFIG_MMC_BOOTARGS      "noinitrd console=ttySAC1,115200 rw root=/dev/mmcbk0p2
rootfstype=ext3 rootwait"
```

변경 후

/uboot\$ vi include/configs/mango2450.h

```
377 #define CONFIG_NAND_BOOTCOMMAND "nand read 0xC0008000 0x80000 0x480000;bootm
0xC0008000"
378 #define CONFIG_NAND_BOOTARGS      "noinitrd console=ttySAC1,115200 rw ubi.mtd=2
root=ubi0:rootfs rootfstype=ubifs rootwait"
379 //#elif defined(CONFIG_BOOT_MOVINAND)
380 //#define CFG_ENV_IS_IN_MOVINAND
381 //#define CONFIG_BOOTCOMMAND      "sleep 1;nand scrub ;sleep 1;nand erase;sleep 1;movi
read 3800000# 40000 c0000000 ;sleep 1;nand write c0000000 0 40000;sleep 1;movi read kernel
c0000000    ;    sleep 1;nand write c0000000 40000 200000;sleep 1;movi read 3600000# 3800000
c0000000;sleep 1;nand write.yaffs c0000000 400000 35db4c0"
382 #define CONFIG_MMC_BOOTCOMMAND "nand scrub; movi read u-boot C1000000; nand erase
0 40000; nand write C1000000 0 40000; movi read kernel C1000000; nand erase 80000 400000; nand
write C    1000000 80000 400000; movi read kernel 0xC0008000; bootm 0xC0008000"
383
384 #define CONFIG_MMC_BOOTARGS      "noinitrd console=ttySAC1,115200 rw root=/dev/mmcbk0p2
rootfstype=ext3 rootwait"
```

S99_build_system.sh를 추가하였습니다.

```
/image/rootfs/etc/init.d$ vi S99_build_system.sh
```

```
1 #!/bin/sh
2
3 echo "Start nand boot ubifs filesystem ..."
4 mkdir -p /mnt/nand
5
6 flash_erase /dev/mtd2 0 0
7 echo "[Step 1] mtd2 flash_erase done ..."
8
9 ubiattach /dev/ubi_ctrl -m 2
10 echo "[Step 2] mtd2 ubiattach done ..."
11
12 ubimkvol /dev/ubi0 -N rootfs -m
13 echo "[Step 3] mtd2 ubimkvol done ..."
14
15 mount -t ubifs ubi0:rootfs /mnt/nand
16 echo "[Step 4] mount done ..."
17
18 tar xvf /root/ /rootfs.tar -C /mnt/nand
19 echo "[Step 5] file copy done ..."
20
21 sleep 2
22
23 umount /mnt/nand
24 echo "[Step 6] umount done ..."
25
26 echo "Complete nand boot ubifs filesystem ..."
27
28 sleep 1
```

```
rootfs/etc/init.d $ cd ../../
rootfs $ tar cf ../rootfs.tar *
```

sd 카드를 컴퓨터에 연결 후 dmesg|tail 명령어로 sd카드 인식 확인

```
image$ ./sdwriter_sdhc sdb 24
```

sd 카드에 이미지를 라이트 합니다.

```
image$ cd rootfs/etc/init.d/
```



```
init.d$ rm -rf S99_build_system.sh
image/rootfs/etc/init.d$ cd ../..
rootfs $ tar cf ../rootfs2.tar *
```

Mango24R2의 DRAM이 64MB입니다.
yaffs2의 용량을 확인하면 64MB를 훌쩍 넘어서
UBIFS를 사용했습니다.

mango24R2에 삽입되어있던 sd카드를 빼서
sd카드를 리눅스 pc에 삽입합니다.
fdisk -l 을 입력합니다.
개인에 맞게 sdc1 sdc2 가 보입니다.
sdc1은 이전에 sdwrie할때 사용한 것으로 sdc2를 사용합니다.

```
mkdir /home/usb
sudo chmod 755 /home/usb
```

마운트할 곳을 만들고 권한 설정해줍니다.

```
sudo mount /dev/sdc2 /home/usb
```

마운트 해주고

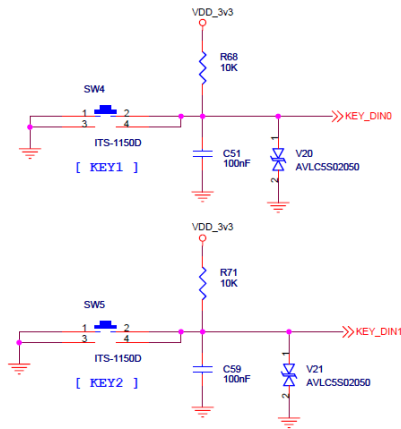
```
sudo cp rootfs2.tar /home/usb/root/rootfs.tar
sudo umount /home/usb
```

sd카드를 빼서 보드에 삽입 후 sd부트로 바꿔서 부팅 (3 ON)
난드 부팅 (3, 5, 6 ON)합니다.

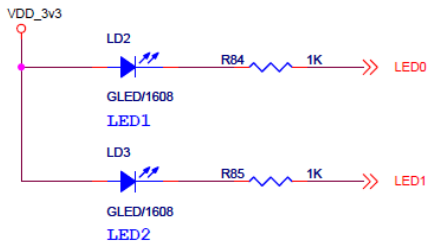
2. key1,2로 7 inch, 10.4 inch LCD선택하여 NAND flash write

key1을 누르고 부팅시 7인치 LCD용 커널 이미지를 라이트 하도록 하겠습니다.

KEYS



LEDS



매뉴얼을 보면

3.6 PORT F CONTROL REGISTERS (GPFCON, GPFDAT, GPFUDP)

If GPF0 – GPF7 will be used for wake-up signals from Sleep/Stop/Deep Stop mode, the ports will be set in EINT.

Register	Address	R/W	Description	Reset Value
GPFCON	0x56000050	R/W	Configures the pins of port F	0x0
GPFDAT	0x56000054	R/W	The data register for port F	0x0
GPFUDP	0x56000058	R/W	Pull-up/down control register for port F	0x5555
Reserved	0x5600005c	–	–	–

GPFCON	Bit	Description	
Reserved	[31:16]	Reserved	
GPF7	[15:14]	00 = Input 10 = EINT[7]	01 = Output 11 = Reserved
GPF6	[13:12]	00 = Input 10 = EINT[6]	01 = Output 11 = Reserved
GPF5	[11:10]	00 = Input 10 = EINT[5]	01 = Output 11 = Reserved
GPF4	[9:8]	00 = Input 10 = EINT[4]	01 = Output 11 = Reserved
GPF3	[7:6]	00 = Input 10 = EINT[3]	01 = Output 11 = Reserved
GPF2	[5:4]	00 = Input 10 = EINT[2]	01 = Output 11 = Reserved
GPF1	[3:2]	00 = Input 10 = EINT[1]	01 = Output 11 = Reserved
GPF0	[1:0]	00 = Input 10 = EINT[0]	01 = Output 11 = Reserved

GPFCON 으로 Input, Output, EINT, Reserved 모드를 결정합니다.

```

195 #define GPFCON_OFFSET    0x50
196 #define GPFDAT_OFFSET    0x54
197 #define GPFPU_OFFSET     0x58

268 #define GPFCON          (0x56000050)
269 #define GPFDAT          (0x56000054)
270 #define GP              (0x56000058)

341 #define GPFCON_REG      __REG(0x56000050)
342 #define GPFDAT_REG      __REG(0x56000054)
343 #define GPFPU_REG       __REG(0x56000058)

```

아래와 같이 키테스트를 합니다. key1을 누르고 부팅시 key1이 출력되고 key2를 누르고 부팅시 key2가 출력됩니다.

```
27 #define KEY01 0x2
28 #define KEY02 0x1
29
30 int key_test(void)
.....
70 int board_init(void)
71 {
72     DECLARE_GLOBAL_DATA_PTR;
73
74     usb_pre_init();
75     smsc9220_pre_init();
76
77     gd->bd->bi_arch_number = MACH_TYPE;
78     gd->bd->bi_boot_params = (PHYS_SDRAM_1+0x100);
79     key_test();
80     return 0;
81 }
.....
98 #if 1//key test
99 int key_test(void)
100 {
101     GPFCON_REG &=0x00;
102
103
104     if((GPFDAT_REG&0x3)==KEY01)
105     {
106         printf("key1");
107     }
108     else if((GPFDAT_REG &0x3)==KEY02)
109     {
110         printf("key2");
111     }
112
113 }
114 #endif
```

아래와 같이 7inch LCD용과 10.4inch LCD용 bootcommand를 정의합니다.

```
uboot$ vi include/configs/mango2450.h
```

```
363 #define CONFIG_NAND_BOOTCOMMAND "nand read 0xC0008000 0x80000 0x500000;bootm
0xC0008000"
364 #define CONFIG_NAND_BOOTARGS      "noinitrd console=ttySAC1,115200 rw ubi.mtd=2
root=ubi0:rootfs rootfstype=ubifs rootwait"
365 		#elif defined(CONFIG_BOOT_MOVINAND)
366 		#define CFG_ENV_IS_IN_MOVINAND
367 		#define CONFIG_BOOTCOMMAND      "sleep 1;nand scrub ;sleep 1;nand erase;sleep 1;movi
read 3800000# 40000 c0000000 ;sleep 1;nand write c0000000 0 40000;sleep 1;movi read kernel
c0000000 ; sleep 1;nand write c0000000 40000 200000;sleep 1;movi read 3600000# 3800000
c0000000;sleep 1;nand write.yaffs c0000000 400000 35db4c0"
368 #define CONFIG_MMC_BOOTCOMMAND " movi read kernel 0xC0008000; bootm 0xC0008000"
369
370 #define CONFIG_MMC_BOOTARGS "noinitrd console=ttySAC1,115200 rw root=/dev/mmcbk0p2
rootfstype=ext3 rootwait"
371
372 #define CONFIG_MMC_7inch_LCD_BOOTCOMMAND "nand scrub; movi read u-boot C1000000;
nand erase 0 40000; nand write C1000000 0 40000; nand erase 80000 500000; movi read 15622094
4FFFFFF c0000000; c0 000000; nand write c0000000 80000 500000; movi read kernel 0xC0008000;
bootm 0xC0008000"
373
374 #define CONFIG_MMC_10_4inch_LCD_BOOTCOMMAND "nand scrub; movi read u-boot
C1000000; nand erase 0 40000; nand write C1000000 0 40000; movi read kernel C1000000; nand
erase 80000 400000; nand write C 1000000 80000 400000; movi read kernel 0xC0008000;
bootm 0xC0008000"
```

위에서 두껍게 표시한 "15622094" 이 값은 변할 수도 있습니다. 동일한 8GB SD카드는 상관없으나 다른 SD카드에서는 값이 변경됩니다. 아래에서 SD Write할 때 설명하겠습니다.

```
/uboot$ vi lib_arm/board.c
```

```
63 #define KEY01 0x2
64 #define KEY02 0x1
.....
565 #if 1 /* Delete this code for NFS */
566     /* by CRZ, boot cmd, args setting */
567     GPFCON_REG &=0x00;
568
569     if((GPCDAT_REG & 0xA0) == 0xA0)
```

```

570  {
571      setenv("bootcmd", CONFIG_NAND_BOOTCOMMAND);
572      setenv("bootargs", CONFIG_NAND_BOOTARGS);
573  }
574  else
575  {
576      if((GPFDAT_REG&0x3)==KEY01)
577      {
578          printf("push key1\Wn7inch LCD image write \Wn");
579          setenv("bootcmd", CONFIG_MMC_7inch_LCD_BOOTCOMMAND);
580          setenv("bootargs", CONFIG_MMC_BOOTARGS);
581      }
582      else if((GPFDAT_REG&0x3)==KEY02)
583      {
584          printf("push key2\Wn10.4inch LCD image write \Wn")
585          setenv("bootcmd", CONFIG_MMC_10_4inch_LCD_BOOTCOMMAND);
586          setenv("bootargs", CONFIG_MMC_BOOTARGS);
587      }
588      else
589      {
590          setenv("bootcmd", CONFIG_MMC_BOOTCOMMAND);
591          setenv("bootargs", CONFIG_MMC_BOOTARGS);
592      }
593  }
594 #endif

```

zImage가 기본이 10.4인치 LCD용 이미지입니다.

image\$ vi sdwriter_sdhc

```

12 KERNEL_NAME=zImage
13 KERNEL_7_NAME=zImage_7inch
.....
74     OFFSET_KERNEL7=$((OFFSET_KERNEL-$SIZEOF_KERNEL))
.....
170     echo -n "OFFSET_KERNEL7 : $OFFSET_KERNEL7 "
171     dd bs=512 seek=$OFFSET_KERNEL7 if=$KERNEL_7_NAME of=$FLASH > /dev/null
2>&1
172     print_success "$?"

```

```
/image$ sudo ./sdwriter_sdhc sdb 24
```

```
[sudo] password for ubuntu:
no talloc stackframe at ../source3/param/loadparm.c:4864, leaking memory
WARNING: Ignoring invalid value 'share' for parameter 'security'
Ignoring unknown parameter "display charset"
Mango SD Writer V1.0
  TFLASH_SECTORS: 15644672
START_ROOTFS=13508606

Unmount all : success

Partition Create : success

Write Mango24R2 Signature OFFSET_SIGNATURE_24: 15643646 success
  OFFSET_BL1: 15643630 Write Mango24R2 BL1 : success
  OFFSET_BL2: 15642574 Write Mango24R2 Uboot : success
  OFFSET_KERNEL: 15632334 Write Mango24 Kernel : success
OFFSET_KERNEL7 : 15622094 success

Linux Filesystem Create : success

Unmount all : success

Success
```

이위에서 SD카드에 따라서 변경될 수도 있는 값을 여기서 확인합니다.

위에서 두껍게 표시한 OFFSET_KERNEL7 : **15622094** success 이값을 사용하여
" uboot\$ vi include/configs/mango2450.h" <<이부분의 값을 변경하면 됩니다.

rootfs.tar를 sd카드에 복사해야합니다.

rootfs.tar와 rootfs2.tar의 차이는 /etc/init.d에 "S99_build_system.sh"가 있고 없고의 차이입니다.

sdwriter_sdhc에서는 rootfs.tar를 사용하여 "S99_build_system.sh"스크립트를 사용합니다.

NAND 부팅을 하기 위해 NAND flash에 ubifs 를 사용하여 filesystem을 write합니다.

스크립트에서 사용하는 filesystem은 rootfs2.tar입니다.

rootfs2.tar는 /etc/init.d에 "S99_build_system.sh" 스크립트를 제거하였습니다.
sd write 후 아래와 같이 명령어를 입력합니다.

```
image$ sudo mount /dev/sdb2 /home/usb
image$ sudo cp rootfs2.tar /home/usb/root/rootfs.tar
image$ sudo umount /home/usb
```

이 부분을 sdwriter_sdhc에 추가하였습니다.

```
199 nand_boot_ubifs_filesystem()
200 {
201     echo -n "Start nand_boot_ubifs_filesystem "
202     sudo mkdir /home/usb
203     sudo chmod 755 /home/usb
204     sudo mount "$TFLASH"2 /home/usb
205     sudo cp rootfs2.tar /home/usb/root/rootfs.tar
206     sudo umount /home/usb
207 }
```

이제 위에 명령어를 입력 안 하고 "sudo ./sdwriter_sdhc sdb 24" 이렇게 입력하여 sdcard를 write하면 됩니다.

3. LCD화면에 writing, write 완료 이미지 출력하기

NAND FLASH에 라이트 후 이미지를 띄우기 위해 rootfs 수정

아래와 같이 추가

```
/etc/init.d$ vi S99_build_system.sh
```

```
/bin/sh /root/bmp_output.sh
```

```
/root$ vi bmp_output.sh
```

```
1 #!/bin/sh
2 export PATH=/usr/bin/./sbin:$PATH
3
4 /root/bmp_reader i /root/bmp/all_ok.bmp
```

bmp_reader를 파일시스템에 root안에 복사합니다.

```
cp bmp_reader root/
```

```
/etc/init.d$ vi S01logging
```

```
/bin/sh /root/bmp_output_writing.sh
```



```
root$ vi bmp_output_writing.sh
```

```
1 #!/bin/sh
2 export PATH=/usr/bin/./sbin:$PATH
3
4 /root/bmp_reader i /root/bmp/file_ing.bmp
```

권한을 설정합니다.

```
sudo chmod 755 bmp_output_writing.sh bmp_output.sh
```

NAND flash에 라이팅 중에는 LCD화면에 아래 이미지 출력하였습니다.

Filesystem Writing...



NAND flash에 write후에는 LCD화면에 아래 이미지 출력하였습니다.

ALL TEST Complete



4. 추가 변경 사항

기본 부팅을 10.4inch LCD에서 7inch LCD로 변경하였습니다.

4.1. 커널 수정

```
/image$ vi sdwriter_sdhc_test_touch_ethernet_usb
```

```

.....
KERNEL_10_4_NAME=zImage_10.4inch
KERNEL_7_NAME=zImage_7inch
ROOTFS_NAME=rootfs_test_touch_ethernet_usb.tar
.....
case "$BOARD" in
    24)
        SIZEOF_SIGNATURE=1026
        # SIZEOF_SIGNATURE=2
        OFFSET_SIGNATURE_24=$((TFLASH_SECTORS-$SIZEOF_SIGNATURE))
        SIZEOF_BL1=16
        OFFSET_BL1=$((OFFSET_SIGNATURE_24-$SIZEOF_BL1))
        #u-boot(1024x512) +env(32x512)
        SIZEOF_BL2=1056

        OFFSET_BL2=$((OFFSET_BL1-$SIZEOF_BL2))

        SIZEOF_KERNEL7=10240
        OFFSET_KERNEL7=$((OFFSET_BL2-$SIZEOF_KERNEL7))
        SIZE_FAT=$((TFLASH_SECTORS-$SIZE_BINARY-$SIZE_ROOTFS-2))

        OFFSET_KERNEL_10_4=$((OFFSET_KERNEL7-$SIZEOF_KERNEL7))
        START_FAT=2048
        START_ROOTFS=$((START_FAT+$SIZE_FAT))
        echo "START_ROOTFS=$START_ROOTFS"
        ;;
.....
write_bin()
{
    case "$BOARD" in
        24)
            echo
            echo -n "Write Mango24R2 Signature"
            echo -n " OFFSET_SIGNATURE_24: $OFFSET_SIGNATURE_24 "
            dd bs=512 seek=$OFFSET_SIGNATURE_24 if=$M24_SIGNATURE of=$TFLASH > /dev/null
2>&1
            print_success "$?"

```

```

echo -n " OFFSET_BL1: $OFFSET_BL1 "
echo -n "Write Mango24R2 BL1 : "
dd bs=512 seek=$OFFSET_BL1 if=$BL1_NAME of=$TFLASH > /dev/null 2>&1
print_success "$?"
echo -n " OFFSET_BL2: $OFFSET_BL2 "
echo -n "Write Mango24R2 Uboot : "
dd bs=512 seek=$OFFSET_BL2 if=$UBOOT_NAME of=$TFLASH > /dev/null 2>&1
print_success "$?"

echo -n "OFFSET_KERNEL7 : $OFFSET_KERNEL7 "
echo -n "Write Mango24 Kernel : "
dd bs=512 seek=$OFFSET_KERNEL7 if=$KERNEL_7_NAME of=$TFLASH > /dev/null
2>&1
print_success "$?"

echo -n " OFFSET_KERNEL_10_4: $OFFSET_KERNEL_10_4 "
dd bs=512 seek=$OFFSET_KERNEL_10_4 if=$KERNEL_10_4_NAME of=$TFLASH >
/dev/null 2>&1
print_success "$?"
;;
*)
echo "Make Bootloader and Kernel: Error !!!"
;;
esac
}

```

4.2. u-boot수정

```
#define CONFIG_MMC_BOOTCOMMAND " movi read kernel 0xC0008000; bootm 0xC0008000"
```

```
#define CONFIG_MMC_BOOTARGS "noinitrd console=ttySAC1,115200 rw root=/dev/mmcblk0p2  
rootfstype=ext3 rootwait"
```

```
#define CONFIG_MMC_10_4inch_LCD_BOOTCOMMAND "nand scrub; movi read u-boot  
C0000000; nand erase 0 40000; nand write C0000000 0 40000; nand erase 80000 500000; movi  
read 15501262 4FFFFFF c0000000; nand write c0000000 80000 500000; movi read kernel  
0xC0008000; bootm 0xC0008000"
```

```
#define CONFIG_MMC_7inch_LCD_BOOTCOMMAND "nand scrub; movi read u-boot C0000000;  
nand erase 0 40000; nand write C0000000 0 40000; movi read kernel C0000000; nand erase  
80000 400000; nand write C0000000 80000 400000; movi read kernel 0xC0008000; bootm  
0xC0008000"
```