

망고 220 1.3M 카메라 포팅 가이드

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

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

Crazy Embedded Laboratory

Document History

Revision	Date	Change note
Init	2015-03-30	전종인

1.	소스 다운로드 받기	4
2.	보드 준비 및 환경 설정.....	4
3.	커널 수정 및 컴파일 하기	4
3.1.	커널 수정 사항.....	5
4.	eMMC 에 이미지 Write 하기.....	6
4.1.	부팅 후 이미지 Write하기.....	8

1. 소스 다운로드 받기

<http://crztech.iptime.org:8080/Release/mango220-exynos4412/android-jellybean/2015-03-27/>

2. 보드 준비 및 환경 설정

망고220 보드와 10.1" LCD , 1.3M 카메라 , 5M 카메라를 준비를 했습니다.

카메라 Port가 2개가 있는데, CAM_A Port에는 5M 카메라를 CAM_B Port에는 1.3M를 연결을 했습니다.



3. 커널 수정 및 컴파일 하기

```
$ cd kernel-3.0.51/
```

eMMC 부팅을 할 것입니다.

아래와 같이 configuration을 합니다.

```
$ ./build_kernel defconfig  
mango220_android_emmc_10_1inch_1280_800_mt9p111_5M_CAM_A_defconfig
```

3.1. 커널 수정 사항

```
[icanjji@icanjji-Samsung-DeskTop-System kernel-3.0.51]$ ./build_kernel config
[sudo] password for icanjji:
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
제안하는 패키지:
  ncurses-doc
다음 패키지를 지울 것입니다:
  libncurses5-dev:i386
다음 새 패키지를 설치할 것입니다:
  libncurses5-dev
0개 업그레이드, 1개 새로 설치, 1개 제거 및 238개 업그레이드 안 함.
222 k바이트 아카이브를 받아야 합니다.
이 작업 후 174 k바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까 [Y/n]? y
```

"y"를 하고 진입합니다.

커널 configuration을 진입하기 위해서는 패키지가 설치가 되어 있어야 합니다.

```
sudo apt-get install libncurses5-dev
```

Configuration 화면으로 진입을 합니다.

```
Device Drivers --->
<*> Multimedia support --->
[*] Video capture adapters --->
Encoders, decoders, sensors and other helper chips --->
  <*> Aptina Sensor mt9p111 (YUV 5M) camera driver
  < > NOON130PC20 Camera Sensor (YUV 1.3M) camera driver
  <*> SR130PC10 Camera Sensor (YUV 1.3M) camera driver
  || Select ITU / MIPI Camera port (ITU Camera dual port) --->
  *** Video and audio decoders ***
```

커널 소스를 수정을 합니다.

"arch/arm/mach-exynos/mach-mango220.c" 파일에서

```
static struct s3c_platform_fimc fimc_plat = {
#ifdef CONFIG_VIDEO_MT9P111
    &mt9p111_a,
```

```

#ifdef CONFIG_ITU_DUAL
    // &mt9p111_b,
#endif
#endif
#ifdef CONFIG_VIDEO_SR130PC10
    //&sr130pc10,

#ifdef CONFIG_ITU_DUAL
    &sr130pc10_b,
#endif
#endif

```

위와 같이 수정을 합니다.

CAM_A Port에는 5M 카메라,

CAM_B Port에는 1.3M 카메라를 장착해서 사용하기 위함입니다.

커널을 컴파일 합니다.

```
$ ./build_kernel
```

4. eMMC 에 이미지 Write 하기

SD Card 8GB를 준비합니다.

리눅스 PC에 장착을 합니다.

```
$ cd ../image/
```

이미지 디렉토리로 이동을 합니다.

장착 후 디바이스 이름을 확인합니다.

```
$ dmesg | tail
```

```

[812355.413436] scsi 65:0:0:0: Direct-Access    Mango    File-CD Gadget  0000 PQ: 0 ANSI: 2
[812355.414657] sd 65:0:0:0: Attached scsi generic sg8 type 0
[812355.417856] sd 65:0:0:0: [sdh] Attached SCSI removable disk
[814920.337222] usb 2-1.3: USB disconnect, device number 25
[1034916.707458] sd 48:0:0:0: [sdg] 15523840 512-byte logical blocks: (7.94 GB/7.40 GiB)
[1034916.708988] sd 48:0:0:0: [sdg] No Caching mode page present
[1034916.708993] sd 48:0:0:0: [sdg] Assuming drive cache: write through
[1034916.711199] sd 48:0:0:0: [sdg] No Caching mode page present
[1034916.711201] sd 48:0:0:0: [sdg] Assuming drive cache: write through
[1034916.718128] sdg: sdg1

```

"sdg" 디바이스 이름을 확인 했습니다.

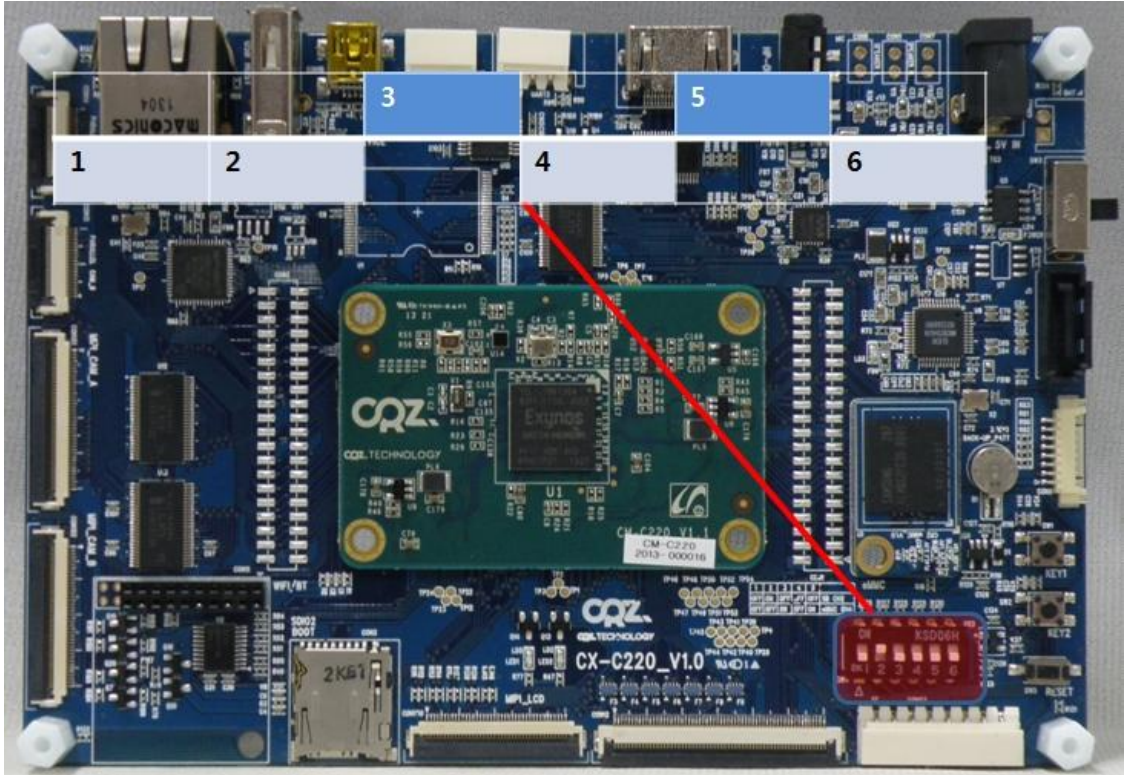
```
[icanjji@icanjji-Samsung-DeskTop-System image]$ sudo ./sdwriter sdg 220 bin
[sudo] password for icanjji:
Mango SD Writer V1.0

Unmount all : success
/dev/sdg reader is identified.
BL1 fusing
30+0 레코드 들어옴
30+0 레코드 나감
15360 바이트 (15 kB) 복사됨, 0.0723759 초, 212 kB/초
BL2 fusing
32+0 레코드 들어옴
32+0 레코드 나감
16384 바이트 (16 kB) 복사됨, 0.106233 초, 154 kB/초
u-boot fusing
573+1 레코드 들어옴
573+1 레코드 나감
293836 바이트 (294 kB) 복사됨, 0.941084 초, 312 kB/초
TrustZone S/W fusing
312+0 레코드 들어옴
312+0 레코드 나감
159744 바이트 (160 kB) 복사됨, 0.553996 초, 288 kB/초
U-boot image is fused successfully.
Eject SD card and insert it again.
success

Unmount all : success

Success
```

“bl1, bl2, TrustZone, u-boot” 이미지가 Write가 됩니다.
망고220 보드에 SD card Slot에 장착을 합니다.
부팅 모드는 3,5번 On을 합니다.



4.1. 부팅 후 이미지 Write하기

망고보드에 DC전원 5V/2A , 3 Pin Serial Cable, USB Cable을 연결을 합니다.

부팅을 합니다.

아래와 같이 u-boot가 실행이 됩니다.

3pin Serial 케이블을 통하여 디버깅 메시지가 출력이 되는 것을 확인 할 수 있습니다.

OK

U-Boot 2010.12 (Mar 18 2015 - 16:40:03) for SMDK4412

CPU: S5PC220 [Samsung SOC on SMP Platform Base on ARM CortexA9]

APLL = 1000MHz, MPLL = 800MHz

DRAM: 1023 MiB

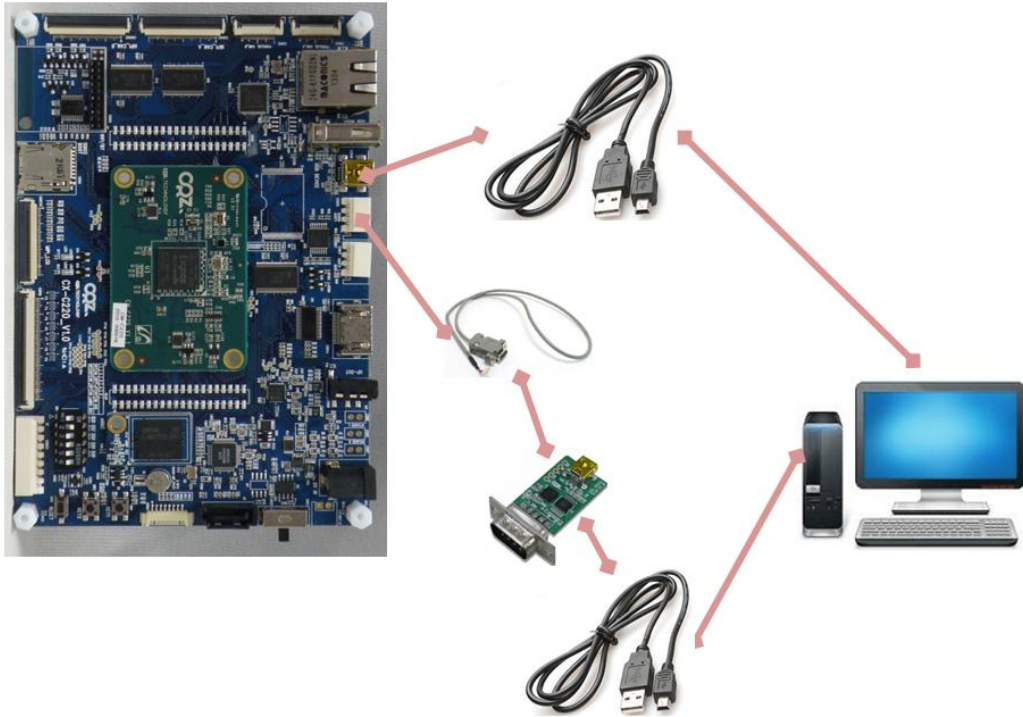
TrustZone Enabled BSP

BL1 version: 20121128


```
Checking Boot Mode ... EMMC4.41
CRZ wifi power down low GPIO_GetDataEach(eGPIO_X1, eGPIO_5)=0x0
  after WIFI_RTC_CLOCK_CTRL=0x200
setup_hsmmc_clock CLK_DIV_FSYS3=0x100
REVISION: 1.1
      EXT_CSD[162] = 0x1
NAME: S5P_MSHC4
EEEEEEEEEEEEEEEEEEEEEXT write OK!! : 0
MMC Device 0: 7456 MB
set_hsmmc_pre_ratio src_clock=800000000, real_clock=400000
set_hsmmc_pre_ratio src_clock=800000000, real_clock=400000
set_hsmmc_pre_ratio src_clock=800000000, real_clock=400000
set_hsmmc_pre_ratio src_clock=800000000, real_clock=400000
set_hsmmc_pre_ratio src_clock=800000000, real_clock=500000000
EEEEEEEEEEEEEEEEEEEEEXT write OK!! : 0
MMC Device 1: 7580 MB
MMC Device 2 not found
there are pending interrupts 0x00000001
*** Warning - using default environment

Net:   smc911x-0
Hit any key to stop autoboot:  0
SMDK4412 #
```

프롬프트 창에서 파티션과 fastboot 명령으로 Write를 합니다.



파티션을 나눕니다.

안드로이드 system, data,cache 파티션 영역을 만듭니다.

```
SMDK4412 # fdisk -c 0 1024 1024 300
Count: 10000
Count: 9999
Count: 9998
Count: 9997
Count: 9996
Count: 9995
Count: 9994
      EXT_CSD[162] = 0x1
NAME: S5P_MSHC4
EEEEEEEEEEEEEEEEEEEEEXT write OK!! : 0
fdisk is completed
```

partion #	size(MB)	block start #	block count	partition_Id
1	4956	4970691	10150360	0x0C
2	1027	134343	2104707	0x83

3	1027	2239050	2104707	0x83
4	306	4343757	626934	0x83

fat영역은 Partion에 1번 영역을 포맷을 합니다.

```
SMDK4412 # fatformat mmc 0:1
Start format MMC&d partition&d ...
Partition1: Start Address(0x4bd8c3), Size(0x9ae1d8)
size checking ...
Under 8G
write FAT info: 32
Fat size : 0x26b8
Erase FAT region.....
Partition1 format complete.
```

“fastboot” 명령으로 입력합니다.

```
SMDK4412 # fastboot
[Partition table on MoviNAND]
ptn 0 name='fwbl1' start=0x1 len=N/A (use hard-coded info. (cmd: movi))
ptn 1 name='bl2' start=N/A len=N/A (use hard-coded info. (cmd: movi))
ptn 2 name='bootloader' start=N/A len=N/A (use hard-coded info. (cmd: movi))
ptn 3 name='tsw' start=N/A len=N/A (use hard-coded info. (cmd: movi))
ptn 4 name='kernel' start=N/A len=N/A (use hard-coded info. (cmd: movi))
ptn 5 name='ramdisk' start=N/A len=0x0(~27262976KB) (use hard-coded info. (cmd: movi))
ptn 6 name='system' start=0x1 len=0x0(~1077609984KB)
ptn 7 name='userdata' start=0x1 len=0x0(~1077609984KB)
ptn 8 name='cache' start=0x1 len=0x0(~320990208KB)
ptn 9 name='fat' start=0x1 len=0x0(~902017024KB)
OTG cable Connected!
```

리눅스 PC에서 아래와 같이 명령을 입력합니다.

```
$ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 002 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 001 Device 003: ID 05e3:0716 Genesys Logic, Inc. USB 2.0 Multislot Card Reader/Writer
Bus 002 Device 100: ID 058f:6366 Alcor Micro Corp. Multi Flash Reader
```

Bus 002 Device 026: ID 18d1:0002 Google Inc.

디바이스 인식 되었는지 확인을 합니다.

아래와 같이 "bl1, bl2, trust zone, u-boot, kernel, ramdisk,system" 이미지를 Write를 합니다.

```
sudo ./fastboot flash fwbl1 E4412_S.bl1.SMDK.MR3.bin
sudo ./fastboot flash bl2 E4412_S.bl2.SMDK.MR3.bin.signed
sudo ./fastboot flash bootloader u-boot.bin
sudo ./fastboot flash tzsw E4412_S.tzsw.SMDK.MR3.bin.signed
```

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

Write가 완료가 되고, reboot를 합니다.

카메라 테스트 결과

