

망고100 메뉴얼

개발환경부터 포팅까지

개발환경부터 U-boot 분석

리눅스 역사(history)

- Linux 역사
 - 핀란드 대학원생 Linus B. Torvalds
 - 1991 version 0.01 발표
 - 1994년에는 리눅스 커널 버전 1.0
 - 1996년 2.0
 - 1999년 2.2 버전의 발표로, 엔터프라이즈 환경에 진입할 수 있는 초석 마련
 - 최근 2.4 버전이 주로 사용되고 2.6 발표 상태
 - A. Tanenbaum 교수의 Minix 기반
(<http://www.cs.vu.nl/~ast/minix.html>)
 - Philosophy of COPYLEFT(open source)
 - GNU support
 - Various Distributions : Redhat, Debian, Slackware, Alzza, MontaVista, Lineo, Gmate, Zaurus, Samsung, IBM, ..

Linux/GNU를 만들어가는 사람들

리누스 토발즈(Linus Torvalds)

- 최초의 리눅스 커널을 만들
- 리눅스 소스코드를 GNU의 GPL에 따라 인터넷에 공개

리처드 스톨만(Richard Stallman)

- GNU 프로젝트의 리더로 FSF를 설립
- 카피레프트(CopyLeft) 주장

래리 월(Larry Wall)

- 펄의 제작자
- 오픈 소스 프로그램과 여러 가지 게임 만들

밥 영 (Bob Young)

- RedHat의 공동 창립자, 오픈소스 운동의 선구자

리눅스와 GNU

- GNU (GNU's not Unix)
 - 80년대 초반 리처드 스톨만(Richard Stallman)에 의하여 시작
 - GPL (GNU Public License)
 - GPL에 의거한 모든 소프트웨어는 무료
 - 변경 사항을 포함해서 재판매 하는 것은 허용하나 소스는 공개해야 함
 - 프로그래머는 자신의 소프트웨어로 발생하는 어떤 위험이나 손해에 대한 법률적 책임이 없음
 - Linux에 gcc, emacs 등을 이식
 - BSD의 많은 유용한 유틸리티를 포함하게 하는 계기가 됨
- 리눅스는 GPL에 의거하여 배포

리눅스 커널 버전의 선택

- 커널 버전
 - 역사 : <http://www.linux.org/dist/kernel.html>
 - 버전 숫자 : X.Y.ZZ
 - X : 커널의 버전
 - Y : 릴리즈 번호, 홀수->개발 중, 짝수->안정된 버전
 - ZZ : Modifications, 사소한 변화를 의미
 - 최신 버전
 - 새로운 다양한 기능이 이미 추가되어 있음
 - 크기가 매우 크다는 단점이 있음
- 커널 버전의 선택
 - 임베디드 시스템의 크기를 고려
 - 필요한 기능을 고려
 - 확장성을 고려

안드로이드란?

- 운영체제와 미들웨어 그리고 핵심 애플리케이션을 포함하고 있는 모바일 디바이스를 위한 소프트웨어 스택
- 안드로이드 SDK는 Java 프로그래밍 언어를 사용하여 안드로이드 플랫폼상의 어플리케이션을 개발하기 위해 필요한 도구들과 API를 제공

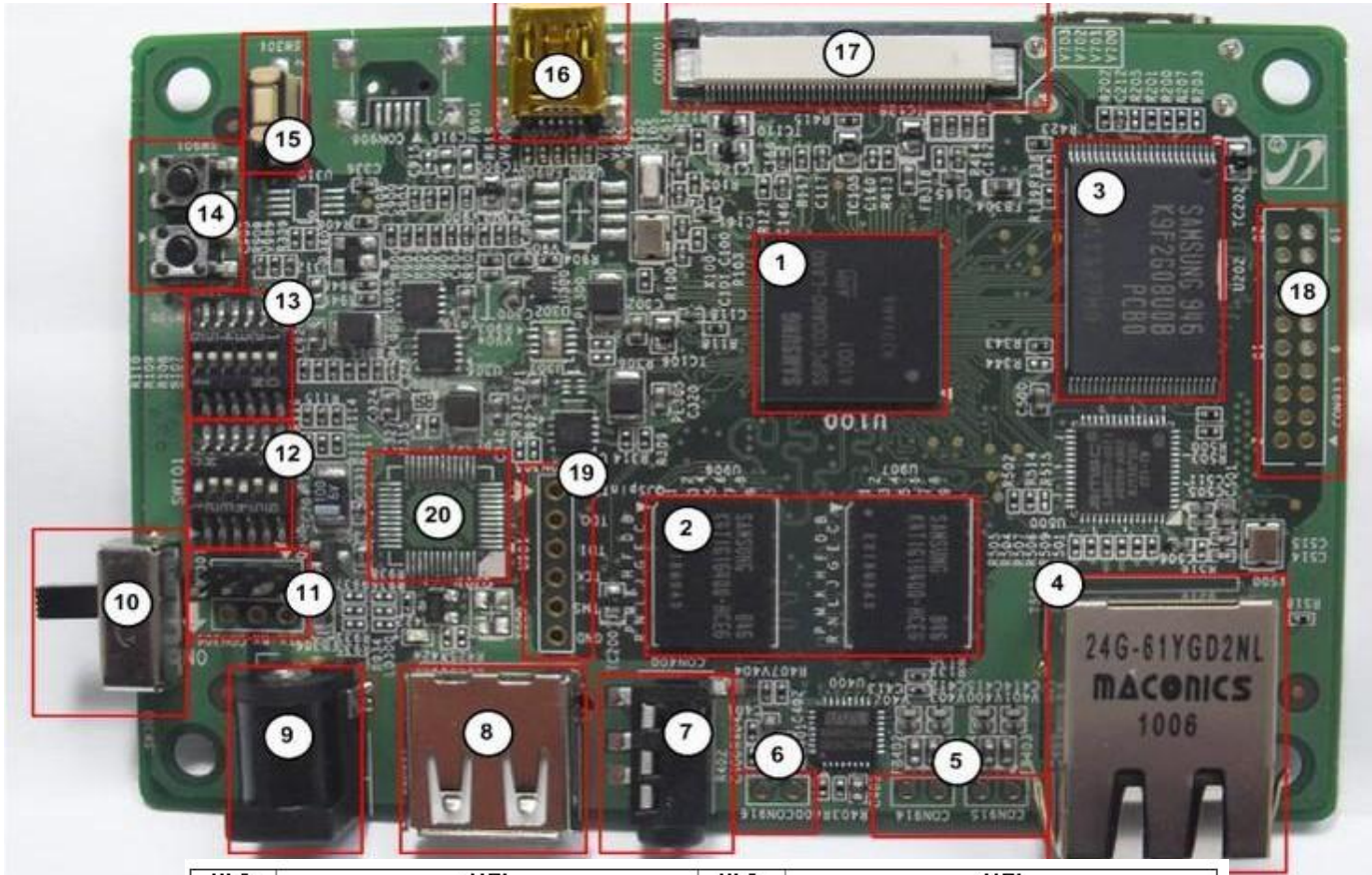
안드로이드 주요특징

- 어플리케이션 프레임 워크
- 달빅(Dalvik)가상머신
- 통합 브라우저 :오픈소스 웹킷 엔지 기반
- 최적화된 그래픽:3D 그래픽 강화
- SQLite:정형화된 데이터 저장공간을 위한 것
- 미디어 지원:MP4,H.264,MP3,AAC,AMR,JPG,PNG,GIF)
- GSM 테크놀러지,블루투스,EDGE,3G와WiFi:하드웨어 의존적
- 카메라,GPS:하드웨어 의존적
- 풍부한 개발환경

망고100 특징

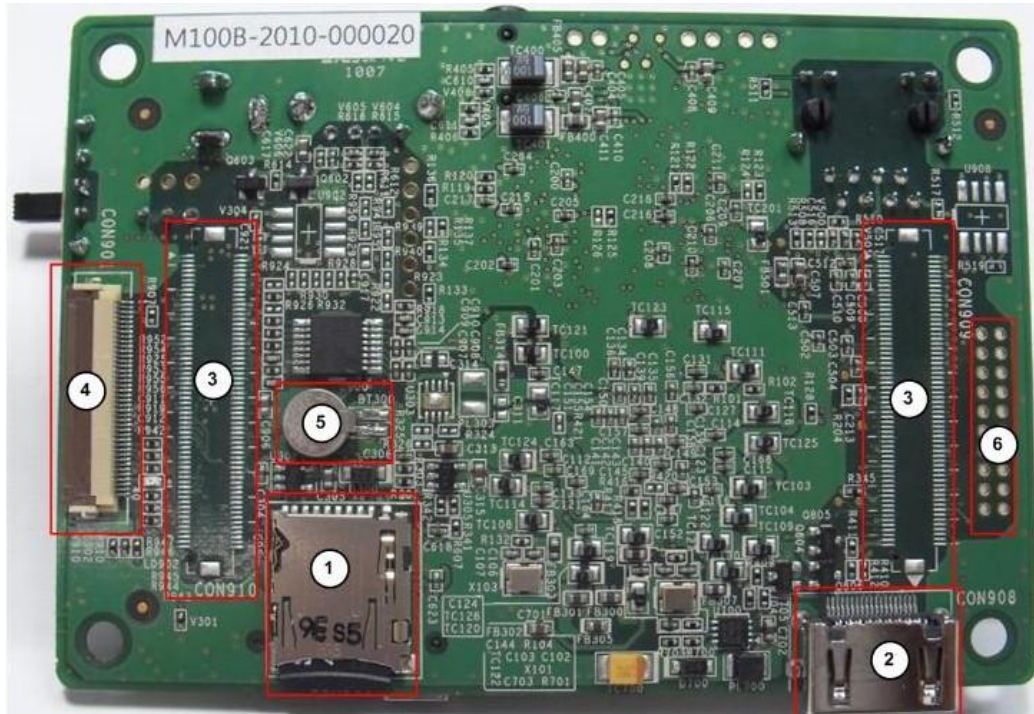
- Cortex-A8 Core 기반으로 삼성 CPU S5PC100 탑재 (667Mhz)
- 3D Hardware 가속기 CPU에 기본 내장
- Wince 6.0, Android, Embedded linux 체험 및 프로젝트에 적합한 보드
- 듀얼 카메라 지원
- HDMI ,TV out 기능 지원
- 가로100 mm,세로 70mm로 휴대 간편

망고100 하드웨어

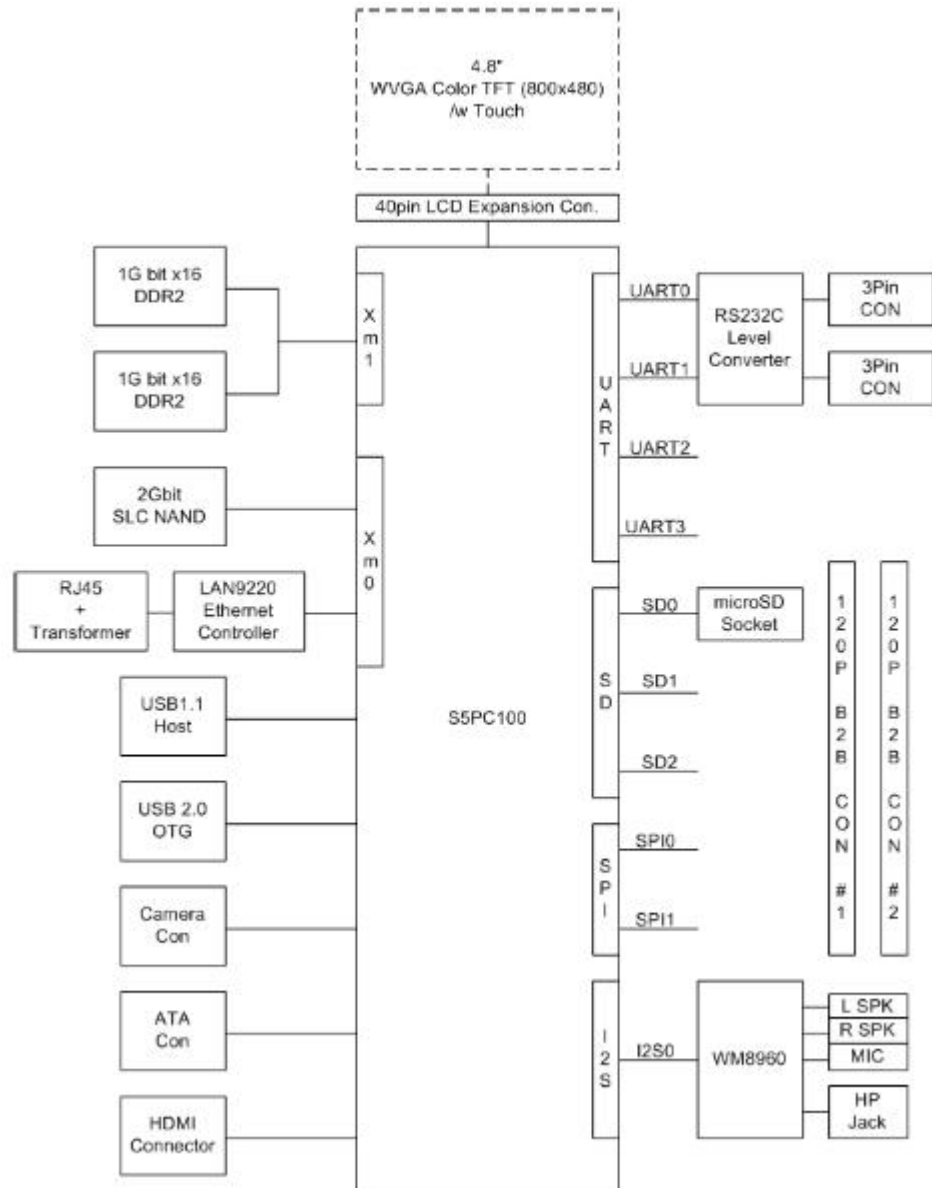


번호	설명	번호	설명
1	S5PC100 Application Processor	2	DDR2 SDRAM
3	SLC NAND Flash	4	10/100 Ethernet RJ45 Connector
5	Speaker Output	6	Microphone Input
7	Headphone Output	8	USB 1.1 Host Connector
9	DC +5V Power Input	10	Power On/Off Switch
11	RS232C Port from UART0 & UART1	12	Boot Mode Switch -1
13	Boot Mode Switch -2	14	USER Switch
15	Reset Switch	16	USB 2.0 OTG Connector
17	Samsung 4.8" WVGA TFT Connector	18	Camera Expansion Header
19	JTAG Connector	20	FT2232 USB/Serial/JTAG Controller

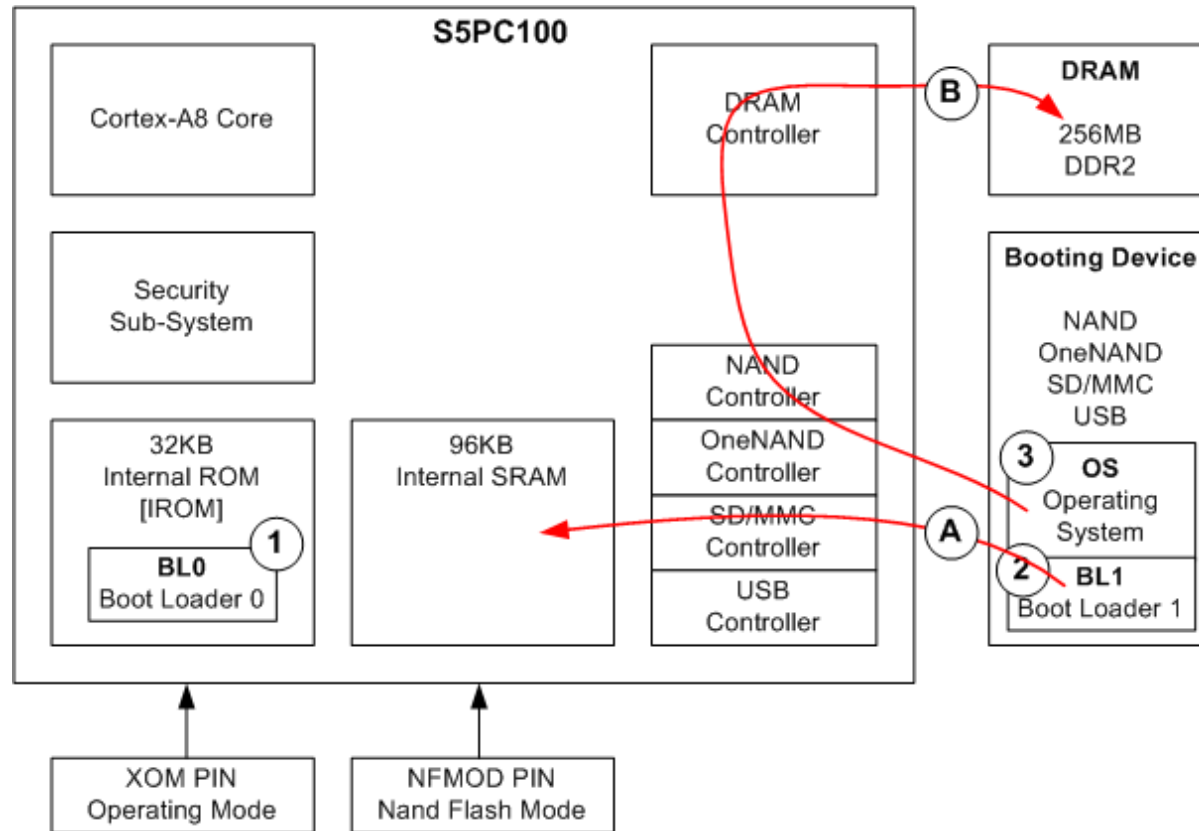
망고100 하드웨어



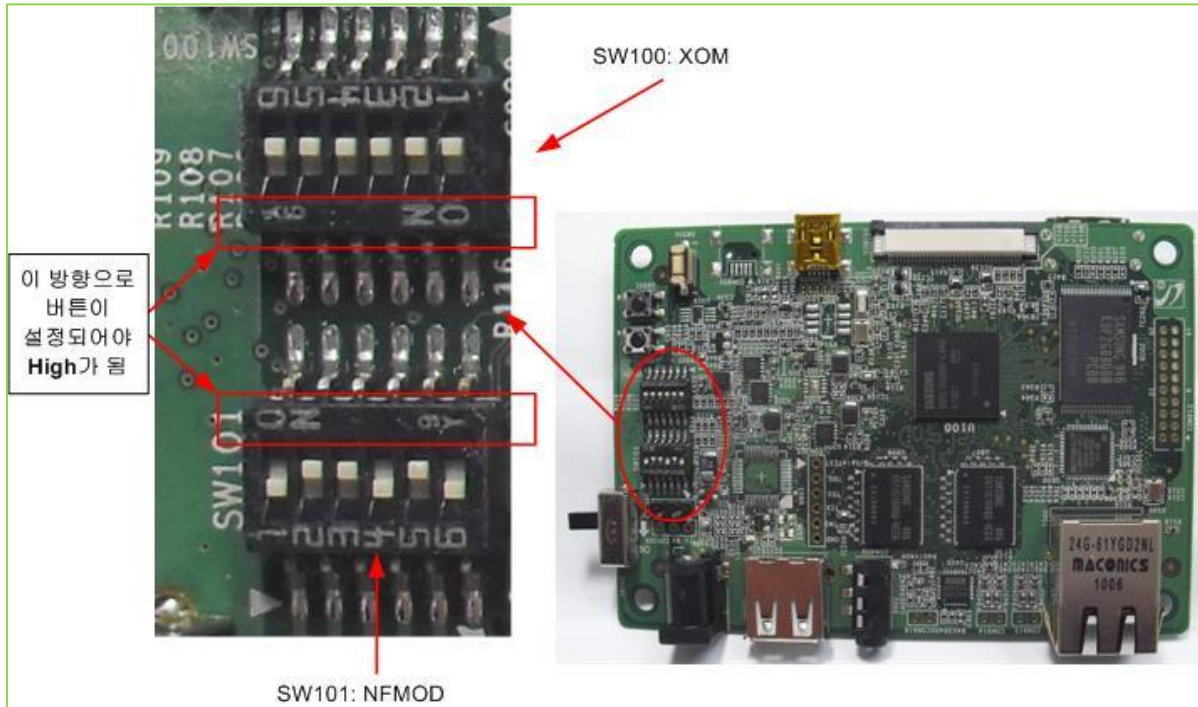
1	Micro-SD card Connector	2	HDMI connector
3	B2B Expansion Connector	4	40Pin ZIF ATA Connector
5	RTC Battery	6	20Pin Camera Expansion Connector



망고100 부팅과정



망고100 부팅 Mode



#	SW100	SW101
1	XOM0	NFMOD0
2	XOM1	NFMOD1
3	XOM2	NFMOD2
4	XOM3	NFMOD3
5	XOM4	NFMOD4
6	NC	NFMOD5

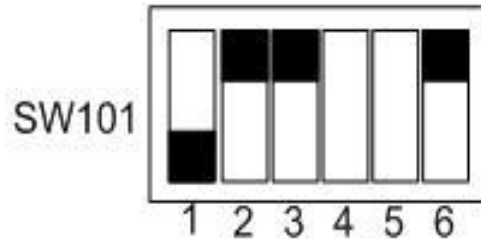
S5PC100 부팅 모드

Function Signal	I/O	Description	Pad	Type
OM[4:0]	Input	<p>OM[4]: 0 = normal mode, 1 = test mode</p> <p>OM[3]: 0 = 1st boot loader in iROM , 1 = reserved</p> <p>OM[2:1] : 00 = 2nd boot loader in NAND flash 01 = 2nd boot loader in OneNAND 10 = 2nd boot loader in MMC 11 = Reserved</p> <p>OM[0] : APLL/MPLL input selection 0 = XXTI 1 = XusbXTI</p>	XOM [4:0]	Dedicated
NFMODE [5:0]	Input	<p>NFMODE [5] : 0 = 2nd booting from the device selected by OM[2:1] 1 = 2nd booting from USB</p> <p>When OM[2:1] = 00 (2nd boot loader in NAND Flash),</p> <p>NFMODE [1:0]: 00 = Small Block (512page), 10 = Large Block (2048page), 11 = Large Block (4096page)</p> <p>NFMODE[2]: 0 = 3 address cycles (for small block) or 4 cycles (for large block) 1 = 4 address cycles (for small block) or 5 cycles (for large block)</p> <p>NFMODE[4]: 0 = 8bit ECC 1 = ECC off</p>	XNFMO D [5:0]	Muxed

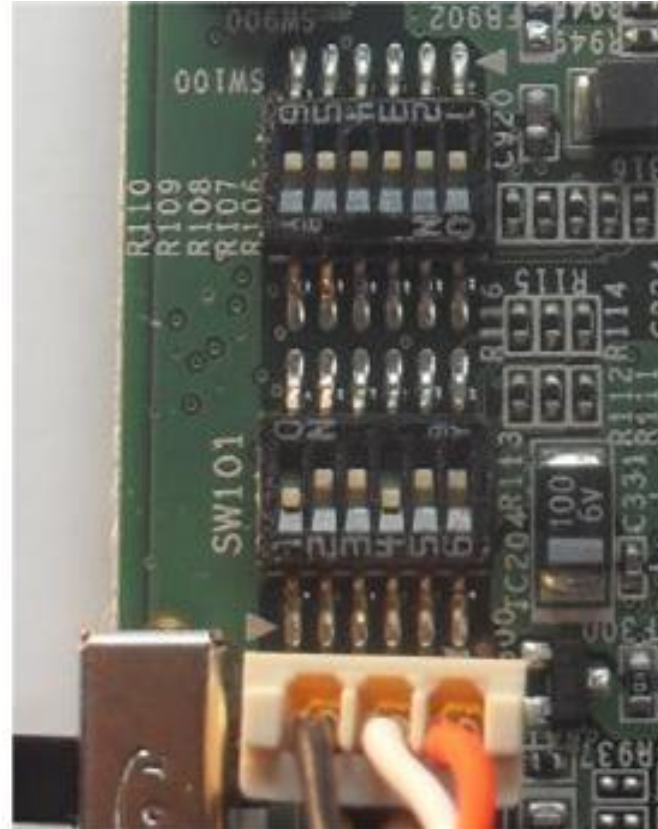
망고100 부팅모드

- USB Boot Mode
- Nand Boot Mode
- SD Boot Mode

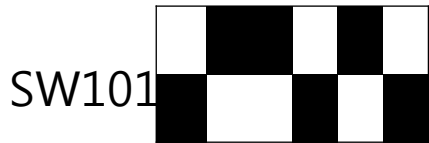
망고100 USB Mode



Button[6]=NFMOD[5]=1
Button[5:4]=NFMOD[4:3]=don't care
Button[3:1]=NFMOD[2:0]=NAND 설정
Large Block,
2048byte page,
4 addr-cycle



망고100 Nand Boot Mode

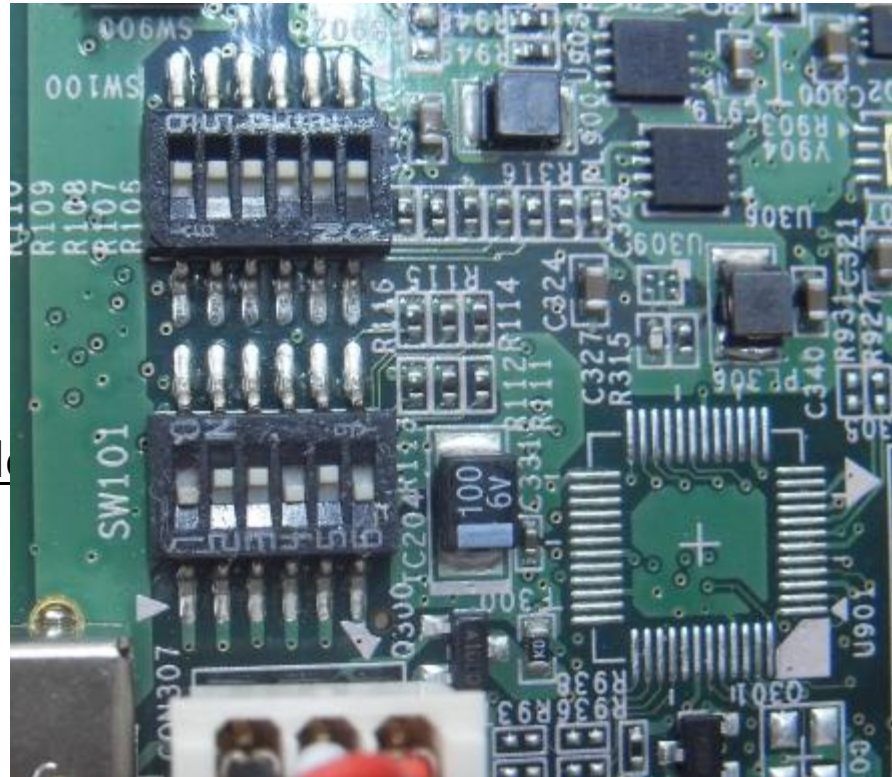


'OM[2:1]=0b00'

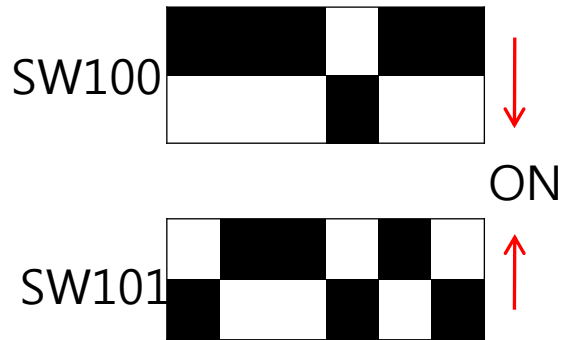
'NFMOD[2:0]=0b110'

Nand Large Block , 4 Addr Cycle

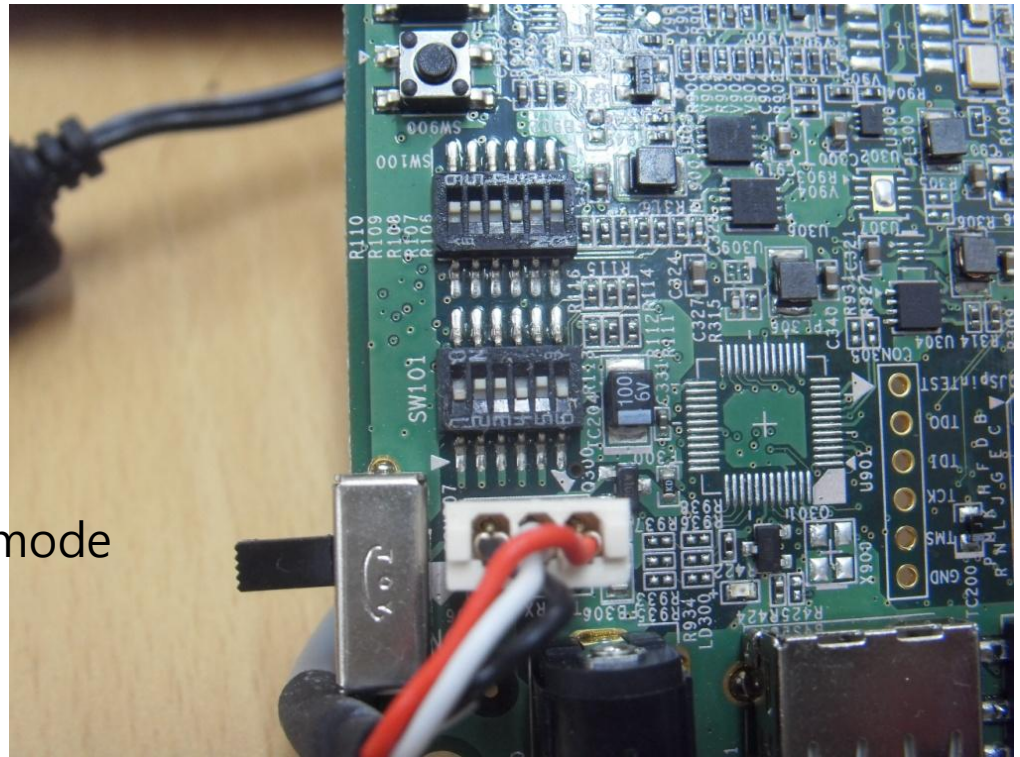
6번 0인 경우 Nand Booting



망고100 SD Boot Mode



OM[2:1]:0b10 MMC boot mode

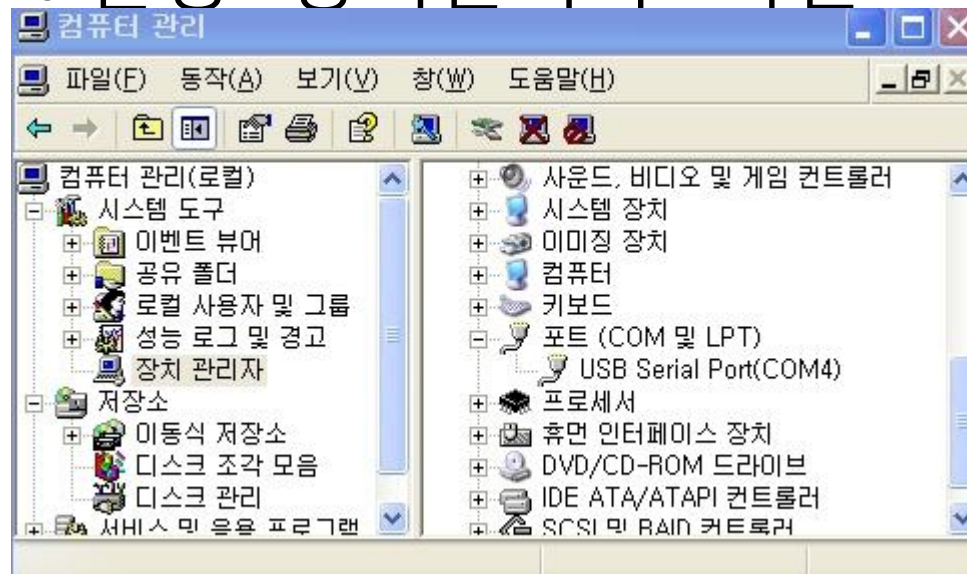


Boot Mode 실습

- Booting Mode 실습
- DNW Tool 다운받기

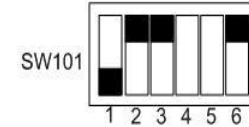
(<http://cafe.naver.com/embeddedcrazyboys/5701>)

- UART Port 설정 "장치관리자" 확인

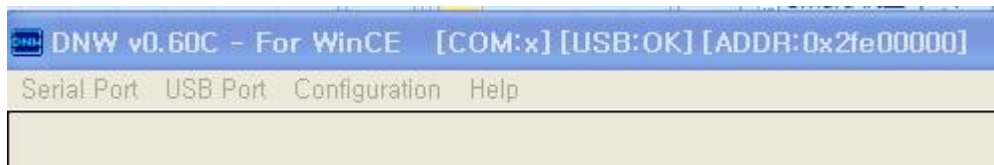
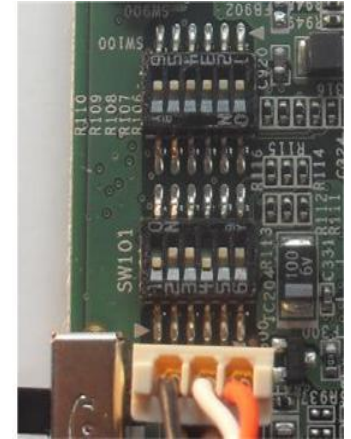


Mango100 USB Boot Mode 실습

- DNW Tool 실행
- 망고100 보드에 전원, UART, USB 연결
- SW101 Boot 스위치 조정
- 전원인가 후 아래와 같이 나오면 OK

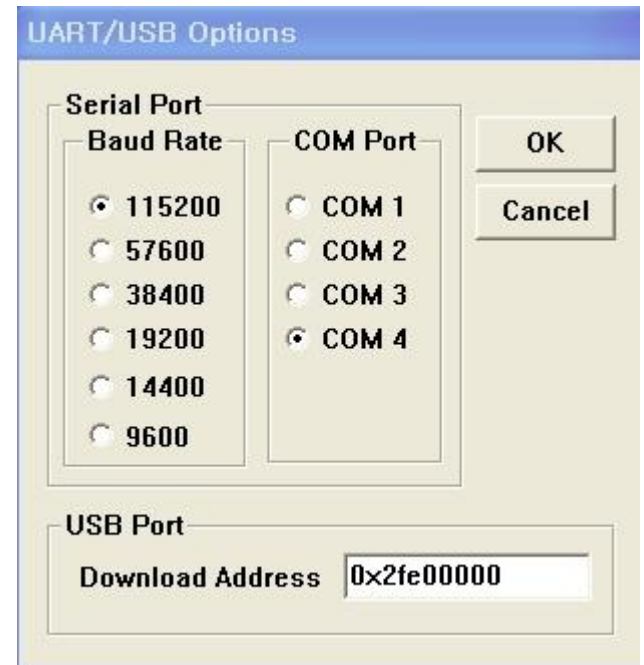


Button[6]=NFMOD[5]=1
Button[5:4]=NFMOD[4:3]=don't care
Button[3:1]=NFMOD[2:0]=NAND 설정
Large Block,
2048byte page,
4 addr-cycle



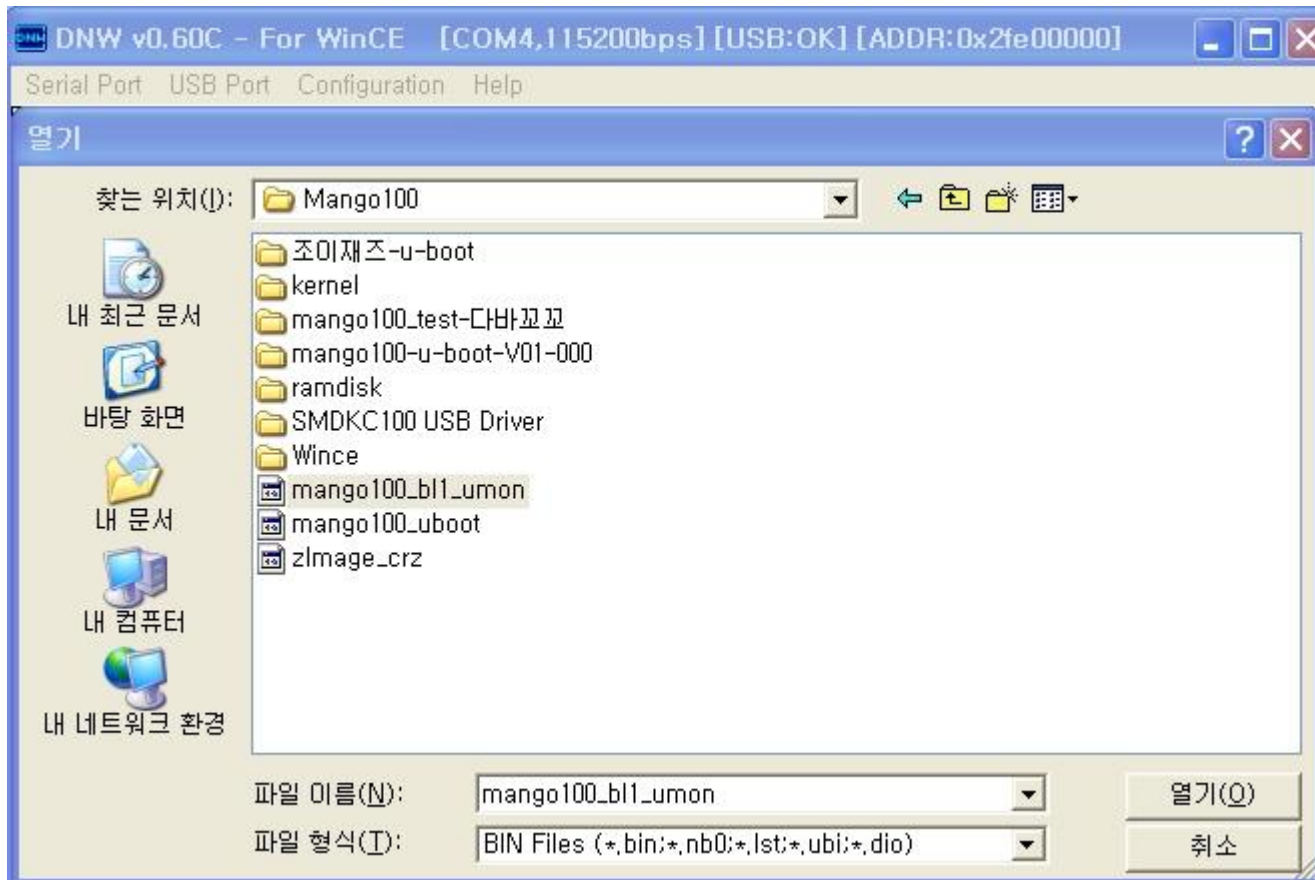
망고100 UART 연결

- “내 컴퓨터 ->장치관리자” PORT 확인
- DNW Tool Configuration 설정
- “Serial Port -> Connect ” 클릭



망고100 USB BootLoader 실행하기

- “USB Port -> Transmit -> Transmit” 선택
- “mango100_bl_umon.bin” 선택



망고100 Nand Mode boot 실습

- DNW 실행
- UART 연결 후 NAND Mode Booting
- Wince 구동 되는 모습

망고100 SD Boot 실행 모습

- SDHC Card를 보드에 Insert
- Default : GNOME File System
- Android 부팅 방법

```
(setenv bootargs 'root=/dev/mmcblk0p2  
rw rootfstype=ext3  
console=ttySAC1,115200 rootdelay=1')
```

개발환경설정

- Vmware 6.5 설치
- Fedora 12 사용
- Tool Chain 2009q3 버전 사용
- JAVA 1.5 버전

Vmware 설치-1

<http://www.vmware.com/products/workstation/index.html>

Why Workstation

What's New

Workstation in Action

Resources

Support

Why Choose VMware Workstation?

From the creator of PC virtualization comes the most reliable, secure way to run multiple operating systems at the same time. Winner of more than 50 industry awards, VMware Workstation transforms the way technical professionals develop, test, demo, and deploy software. VMware Workstation is an integral component of any serious technical professional's toolkit.

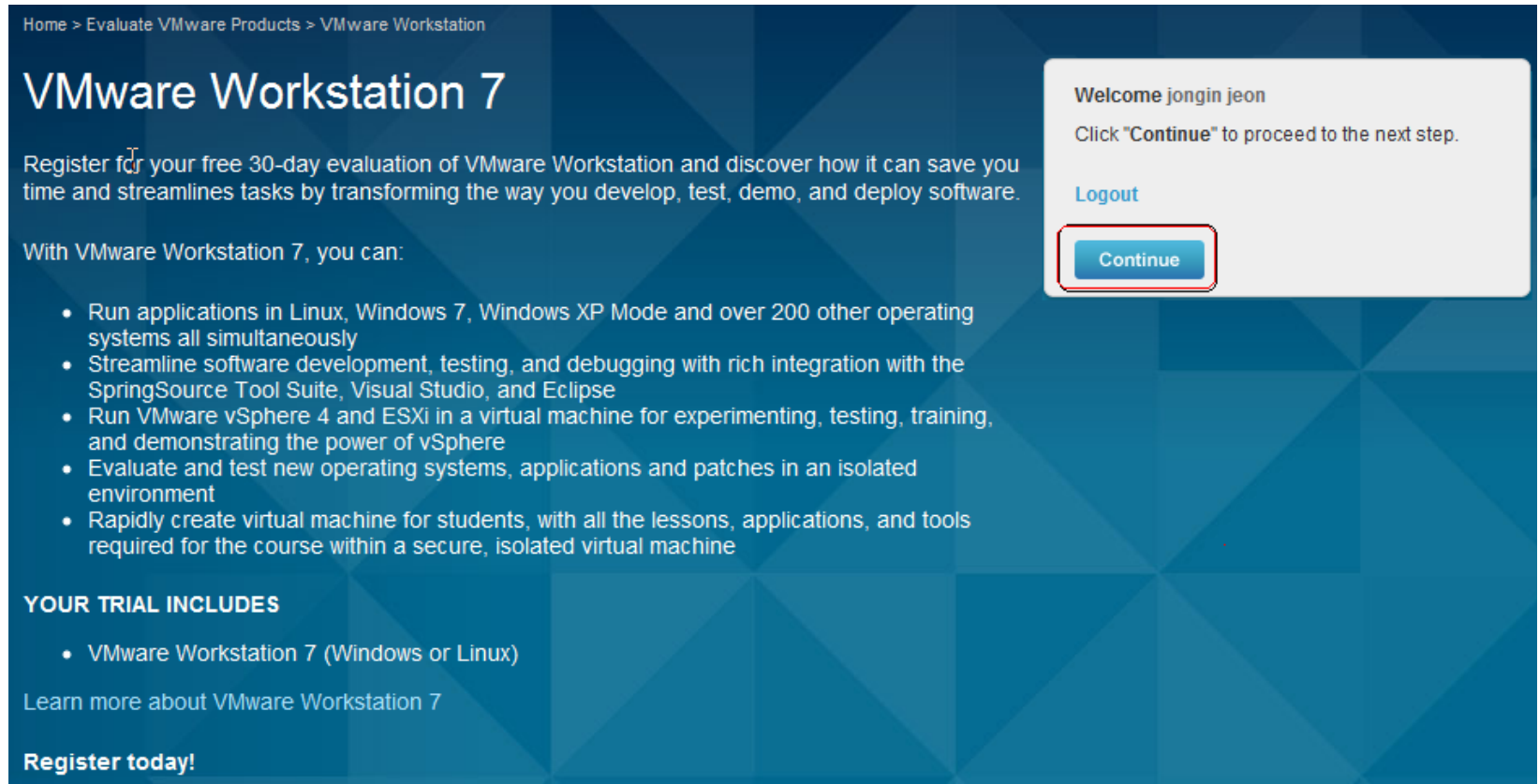
Get Maximum Performance for Windows 7

VMware Workstation 7 is optimized for maximum performance when running on 32-bit and 64-bit Windows 7 PCs. Supports Flip 3D and Aero Peek to show live thumbnails of your virtual machines. Run legacy application with 3D graphics, faster performance, and tighter desktop integration better than Windows XP Mode. Get shared folders and drag and drop convenience.



Vmware 설치 -2

로그인 후 계속 버튼 선택



Home > Evaluate VMware Products > VMware Workstation

VMware Workstation 7

Register for your free 30-day evaluation of VMware Workstation and discover how it can save you time and streamlines tasks by transforming the way you develop, test, demo, and deploy software.

With VMware Workstation 7, you can:

- Run applications in Linux, Windows 7, Windows XP Mode and over 200 other operating systems all simultaneously
- Streamline software development, testing, and debugging with rich integration with the SpringSource Tool Suite, Visual Studio, and Eclipse
- Run VMware vSphere 4 and ESXi in a virtual machine for experimenting, testing, training, and demonstrating the power of vSphere
- Evaluate and test new operating systems, applications and patches in an isolated environment
- Rapidly create virtual machine for students, with all the lessons, applications, and tools required for the course within a secure, isolated virtual machine

YOUR TRIAL INCLUDES

- VMware Workstation 7 (Windows or Linux)

Learn more about VMware Workstation 7

Register today!

Welcome jongin jeon
Click "Continue" to proceed to the next step.

[Logout](#)

[Continue](#)

VM ware 설치-3

Licensing Download Information

! Access Your Email to Activate and Access Your Download

- Please click the link in your evaluation activation email to confirm receipt and gain access to your download.
- Your (30) day evaluation has begun and will expire on 07/21/2010

Where is my Confirmation/Activation email?

- If you have not received your email within 5 minutes, please check your spam blocker or email filters. To ensure you receive the email, please set your email filter to allow mail from the vmware.com and connect.vmware.com domains.
- Please make sure the email address is a valid one.
- [Contact us](#) if you need additional help.

Need us to resend your activation email? To have another confirmation email sent to icanjji@gmail.com, [click here](#).

Binaries

Workstation for 32-bit and 64-bit
Windows
05/25/10 | 7.1 | 568 MB | Binary (.exe)

Start Download Manager ?

Manually Download

.exe installation file for 32-bit and 64-bit Windows with Tools

MD5SUM(*) c810e98030edcc8acc639aef9239f0ca

SHA1SUM(*) 5512cb520fc91b8c4ee9b0d6f80d1cfecb0fe50f

Vmware 설치 -4

Email 확인하면 Register 등록

Dear jongin,

Thank you for your interest in evaluating VMware Workstation. We are pleased to provide a 30 day license to support your evaluation, which will expire on 07/21/2010. Please use the link below to activate your evaluation license and access download information.

[Activate my VMware Workstation Evaluation](#)

Activate your evaluation license and access your download

Use the following link to access the license information and software necessary to complete your installation: [Activate My Evaluation!](#)

Home > Evaluate VMware Products > VMware Workstation > Activation

VMware Workstation 7 Product Evaluation License and Download

Thank you for trying VMware Workstation 7.

We hope you enjoy your free 30-day evaluation. Your license and download information can be found below. If you have questions or need support during your evaluation, visit the [VMware Technology Network](#) for product documentation, knowledge bases and other resources.

Licensing Download Information

Licensing

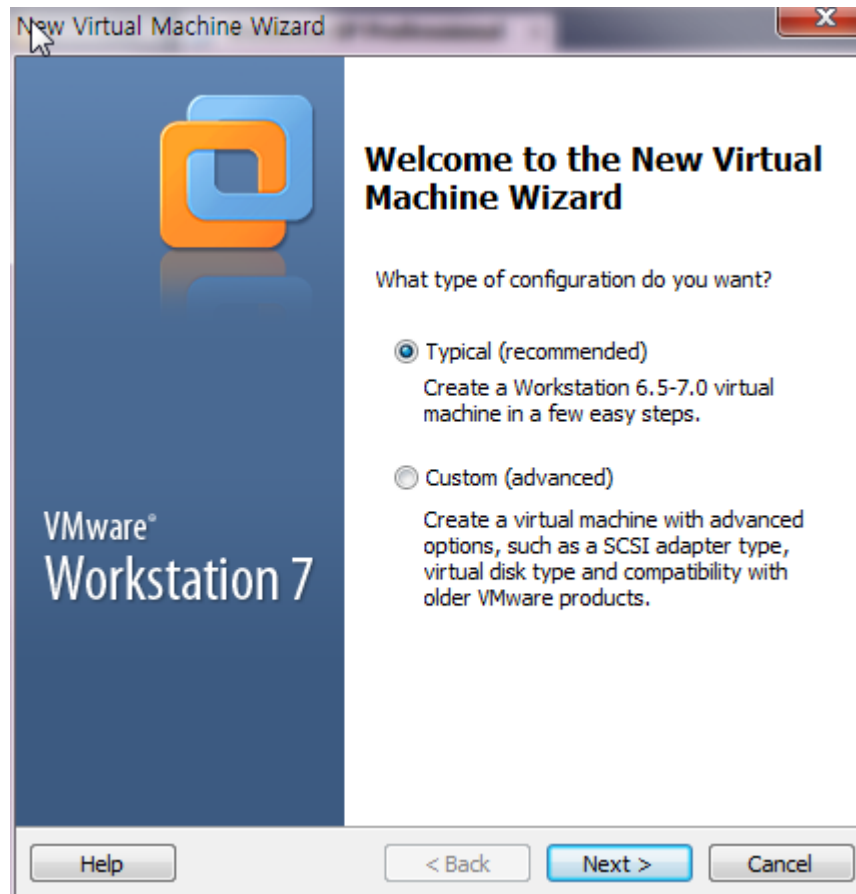
VMware Workstation 7.1

Expiration Date: Jul 21, 2010

M148Q-1V34L-08N3C-0V8K4-2NN0L

Fedora 12 설치-1

- Vmware에서 “File->New->Virtual Machine”선택



Fedora 12 설치-2

Fedora 12 설치 CD를 Insert

Guest Operating System Installation
A virtual machine is like a physical computer; it needs an operating system. How will you install the guest operating system?

Install from:

Installer disc:

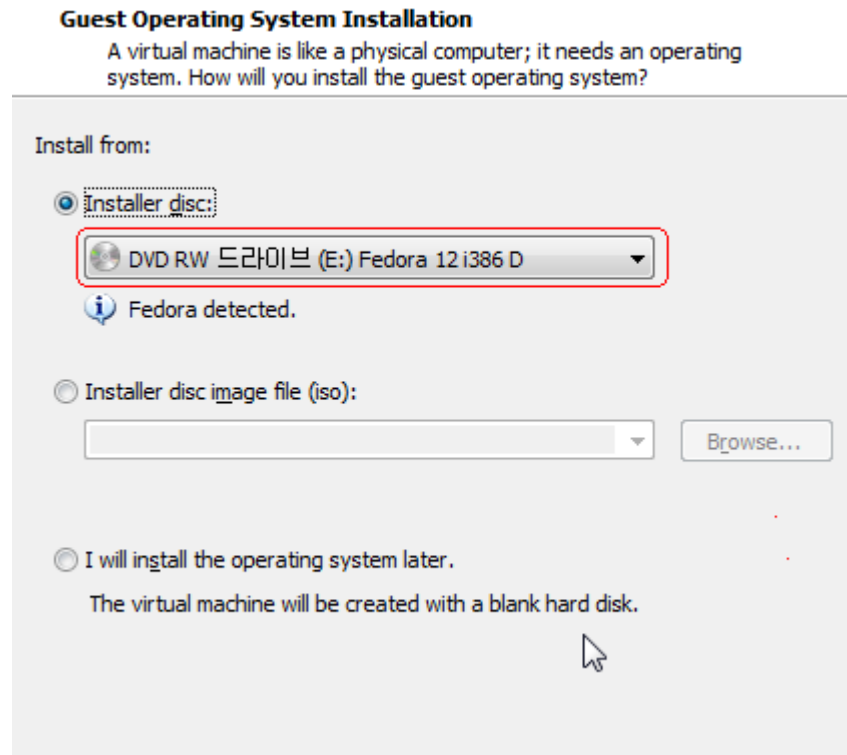
Installer disc image file (iso):

I will install the operating system later.

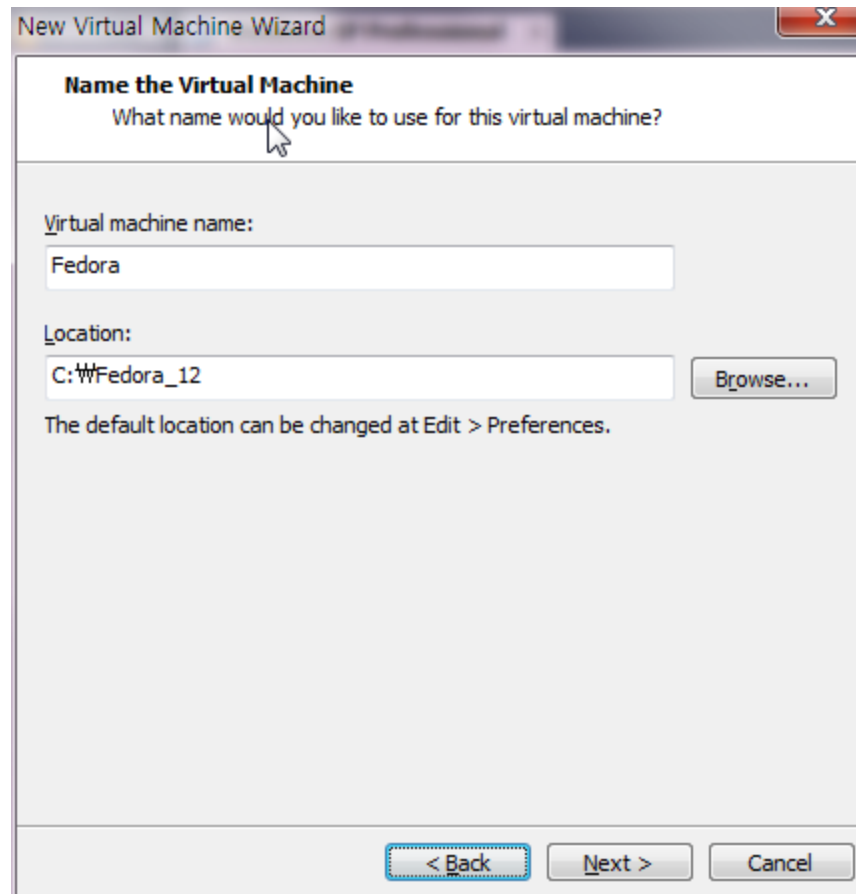
The virtual machine will be created with a blank hard disk.

DVD RW 드라이브 (E:) Fedora 12 i386 D

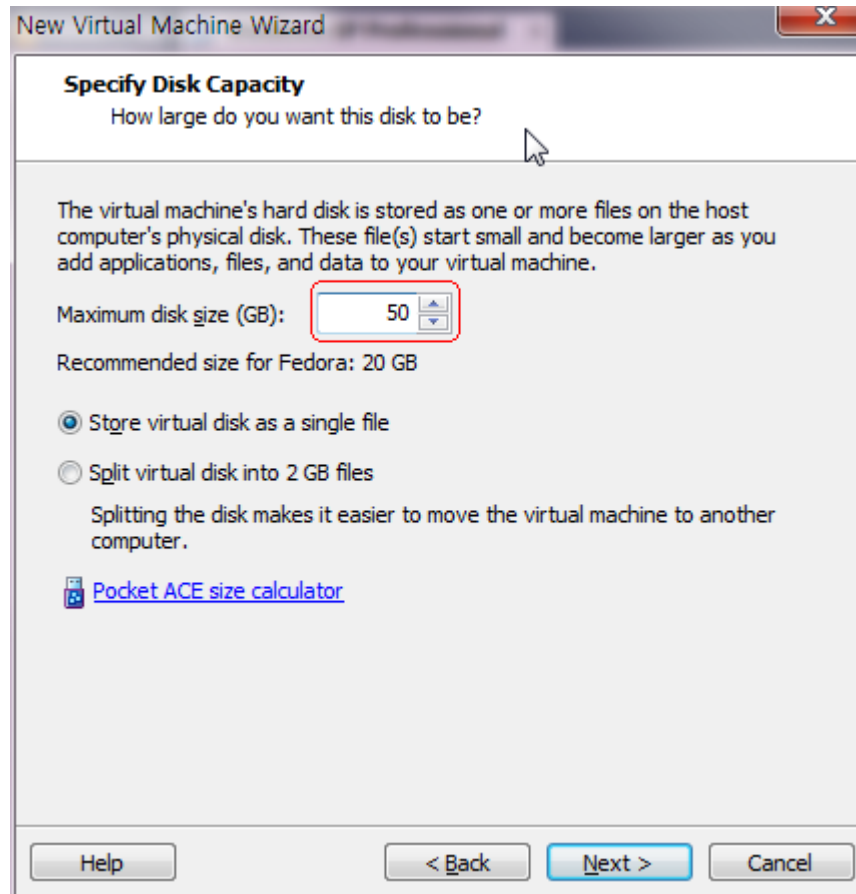
Browse...



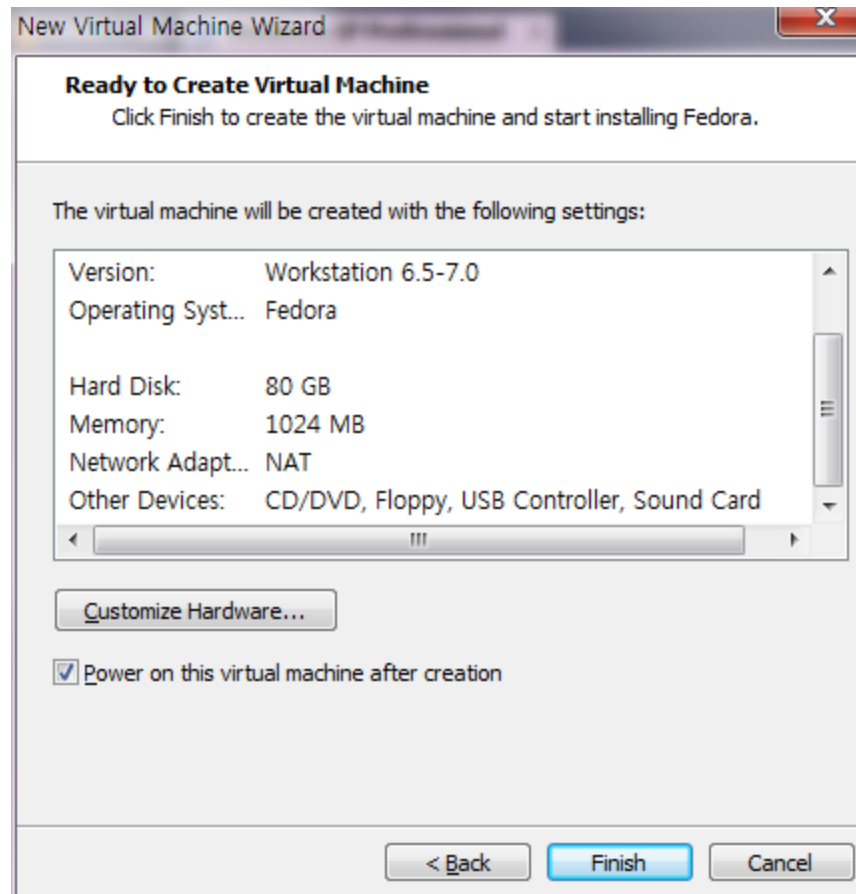
Fedora 12 설치-3



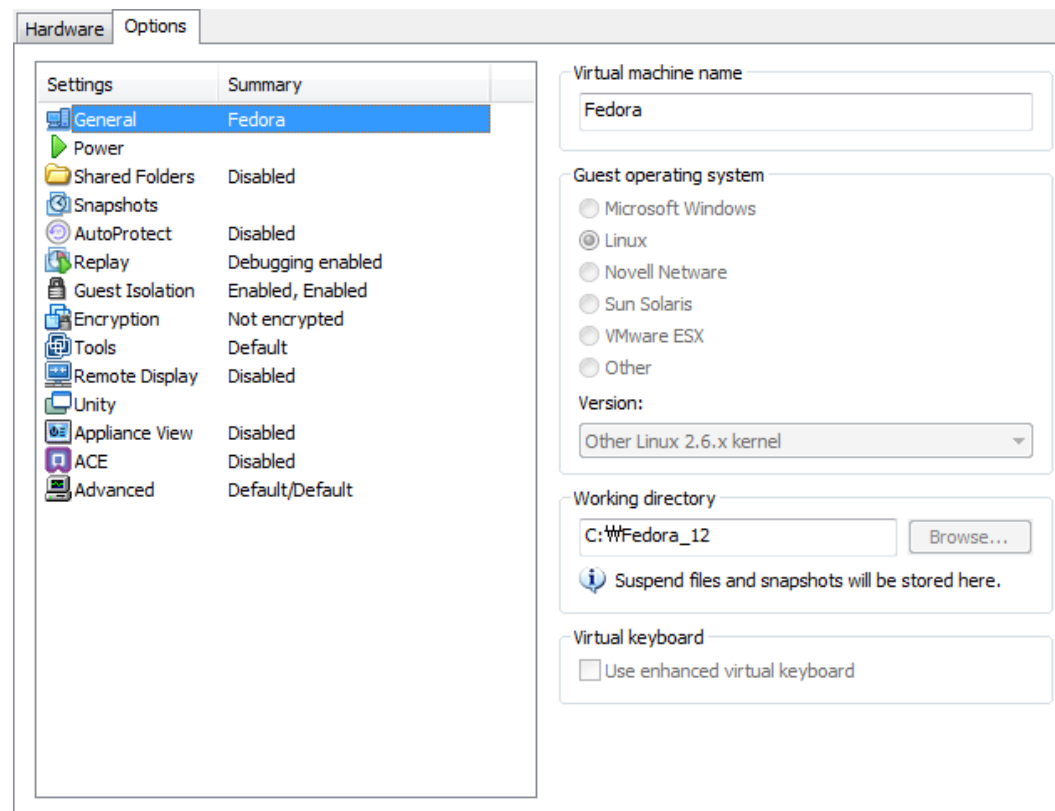
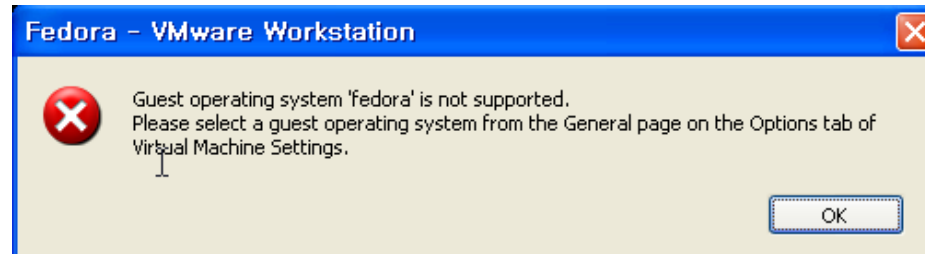
Fedora 설치-4



Fedora 설치-5



Fedora 설치-6



Fedora 설치-9



Fedora 설치-10

Fedora의 초기 설치는 일반적인 인터넷 사용에 맞는 소프트웨어의 모음을 포함하고 있습니다. 어떤 추가적인 임무가 시스템에서 지원되기를 원하십니까?

- 사무와 생산성
- 소프트웨어 개발
- 웹 서버

Please select any additional repositories that you want to use for software installation.

- Installation Repo
- Fedora 12 - i386
- Fedora 12 - i386 - Test Updates
- Fedora 12 - i386 - Updates

+ Add additional software repositories

Modify repository

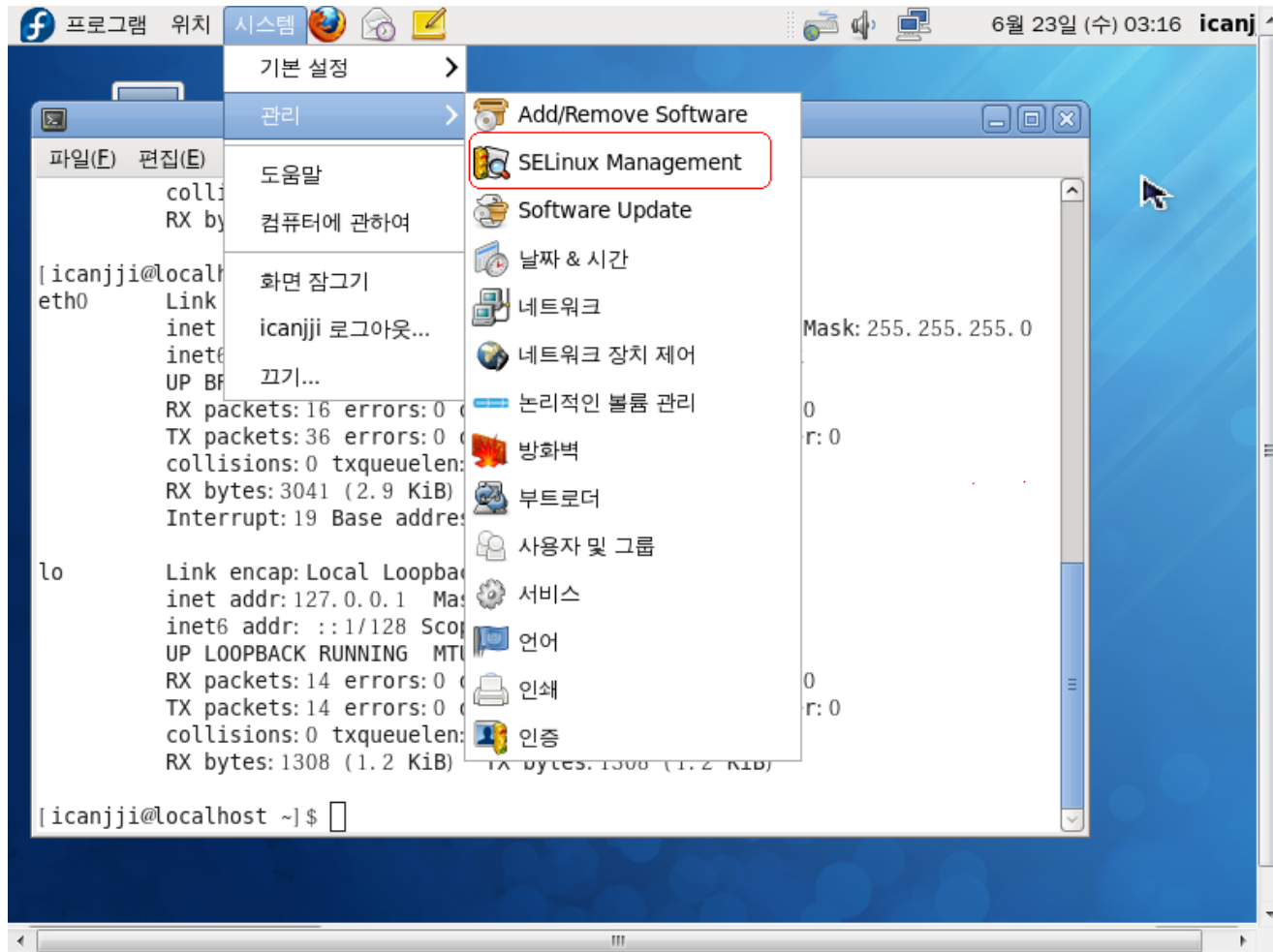
You can further customize the software selection now, or after install via the software management application.

- Customize later Customize now

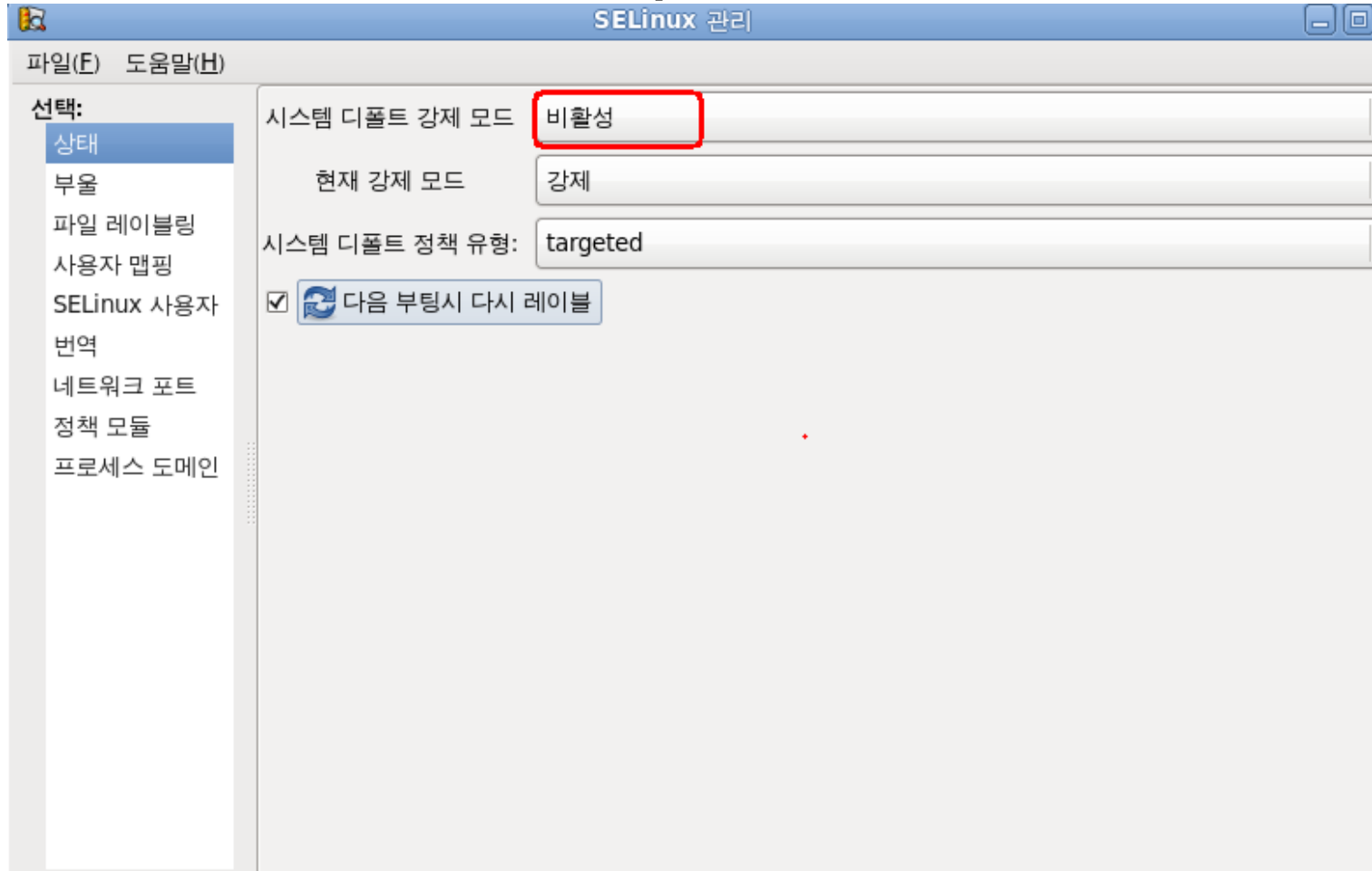
← 뒤로(B)

→ 다음(N)

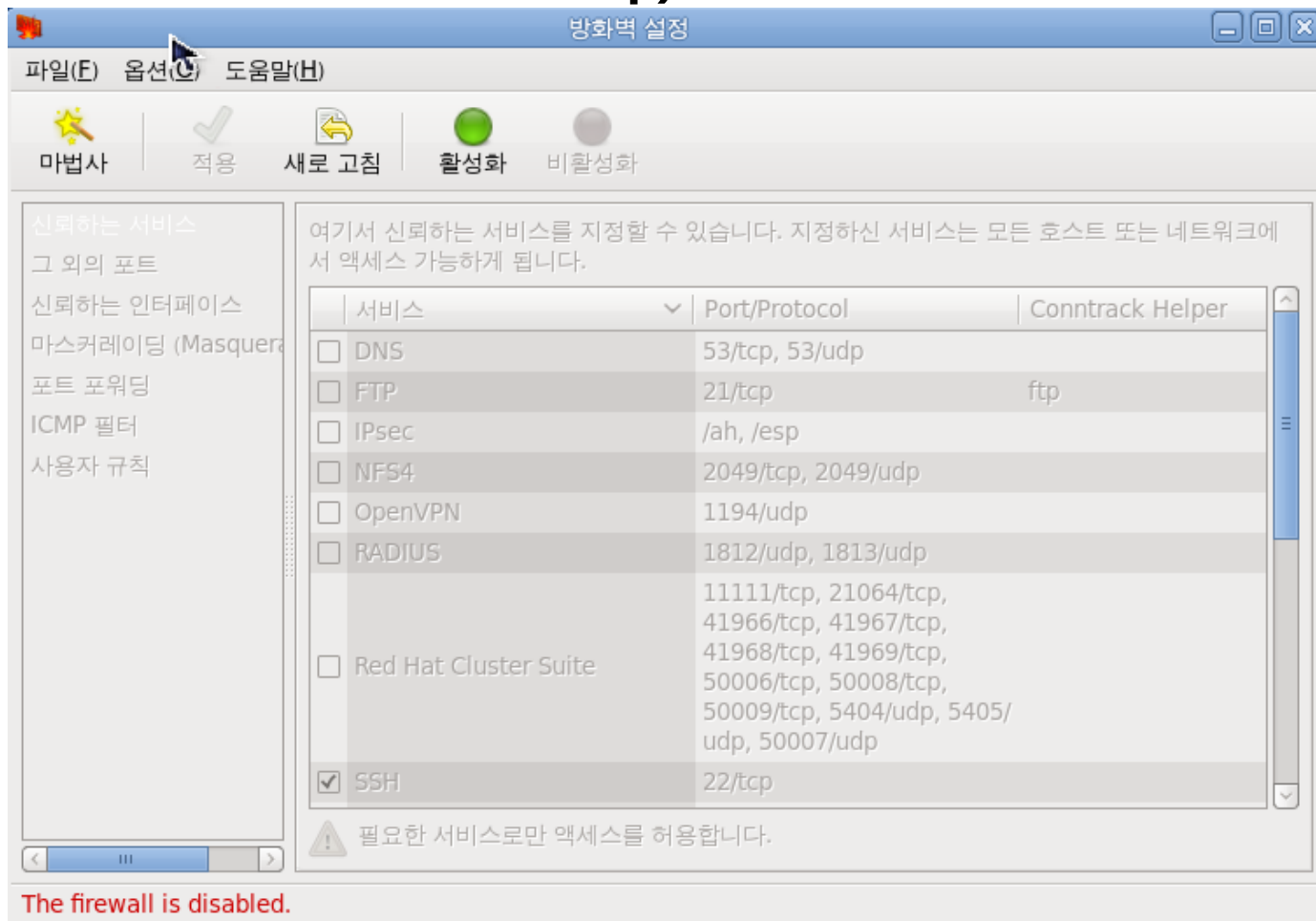
Fedora 환경설정



Fedora 환경 설정 (SELinux 비활성화)

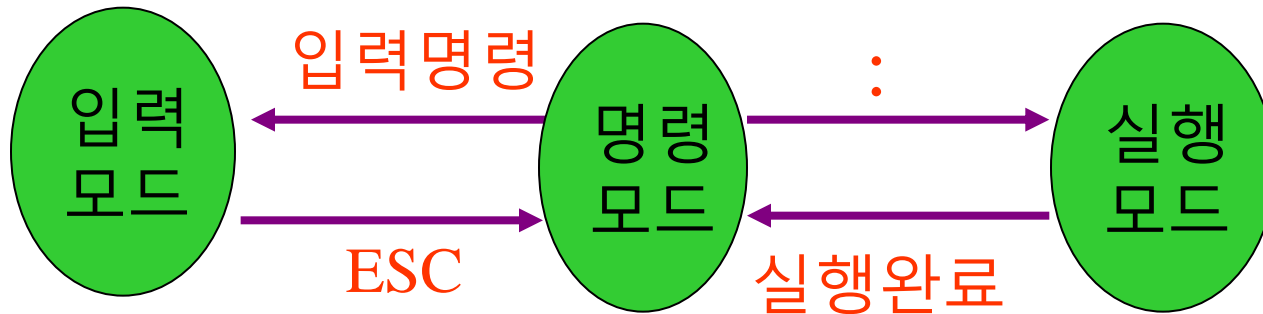


Fedora 환경 설정 (방화벽 비 활성화)



vi 편집기

- 실행 방법: vi 명령어 뒤에 파일 이름
- vi 실행 시 시작하는 모드는 명령 모드
- 편집기 모드
 - **입력모드 혹은 편집모드** - 글자를 입력할 수 있는 모드
 - 모드변환방법 - 명령모드에서 a,A,i,o,O를 입력 했을 때
 - **명령모드 혹은 ESC모드** - 커서이동 및 기타 명령어처리
 - 모드변환방법 - 실행모드 혹은 입력모드에서 ESC키를 눌렀을 때
 - **실행모드 혹은 콜론모드** - 내용바꾸기 및 기타
 - 모드변환방법 - 명령모드에서 콜론(:)을 입력했을 때



vi - 명령 모드

• 입력 모드 전환

a: 커서 위치의 다음 칸부터 입력하기(append)
A: 커서가 있는 줄의 끝부터 입력하기
i: 커서 위치부터 입력하기 (키보드의 Insert도 같은 기능을 합니다.)
I: 커서가 있는 줄의 맨 앞에서부터 입력하기
o: 커서 바로 아래에 줄을 만들고 입력하기(open line)
O: 커서 바로 위에 줄을 만들고 입력하기
s: 커서가 있는 단어를 지우고 입력하기
S: 커서가 있는 행을 지우고 입력하기

• 커서 이동

h: 왼쪽, **j**: 위로, **k**: 아래로, **l**: 오른쪽 (방향키 사용 가능) - 글자단위
w: 다음단어로, **b**: 이전단어로 - word 단위
^: 맨 왼쪽의 첫 글자, **\$**: 마지막글자의 끝 - 행 단위
^F: 한화면 아래로, **^B**:한 화면 위로, **^D**: 반 화면 아래로, **^U**: 반화면 위로

vi - 명령 모드(cont'd)

- 삭제 기능

x : 커서 위치의 글자 삭제
X : 커서 바로 앞의 글자 삭제
dw : 한 단어를 삭제
D : d\$ 커서 위치부터 줄의 끝까지 삭제
dd : 커서가 있는 줄을 삭제

- 복사 및 붙여넣기

yw : 커서 위치부터 단어의 끝까지 복사하기
y0 : 커서 위치부터 줄의 처음까지 복사하기
y\$: 커서 위치부터 줄의 끝까지 복사하기
yy : 커서가 있는 줄을 복사하기
yj : 커서가 있는 줄과 그 다음 줄을 복사하기
yk : 커서가 있는 줄과 그 앞줄을 복사하기
p : 커서의 다음 위치에 붙여 넣기
P : 커서가 있는 위치에 붙여 넣기

vi - 명령 모드(cont'd)

- 기타

u : 작업 취소하기 (undo)
U : 그 줄에 행해진 작업 모두 취소하기
. : 조금 전에 했던 명령을 반복하기
~ : 대소문자 전환
/검색어 : 아래 방향으로 찾기 (검색)
?검색어 : 위쪽 방향으로 찾기
n : 다음 찾기

vi - 실행 모드

- 치환관련 실행

```
:s/old/new/g - old를 new 로 치환  
:s/^old/new/g - 행의 첫 단어가 old 인 것을 new 로 치환  
:s/old$/new/g - 행의 끝 단어가 old 인 것을 new 로 치환  
:s/aaa//g - aaa를 삭제
```

- 파일 관련 실행

```
:w 파일명 "파일명"으로 저장  
:q 저장하지 않고 종료  
:q! 변경 사항을 버리고 종료  
:e 파일명 "파일명"의 파일을 불러들여 편집  
:r 파일명 "파일명"의 파일을 읽어서 삽입  
:!명령어 외부명령어 실행
```

Sudo 명령 사용하기

- #su
- #vim /etc/sudoers

```
## Allow root to run any commands anywhere
root    ALL=(ALL)    ALL
icanjji ALL=(ALL)    ALL
```

- #sudo 계정

Tool chain ?

- Tool chain 이란?
 - Target 시스템의 Software 개발을 진행하기 위해 필요한 host system의 cross compile(교차 컴파일)환경.
 - source code 을 compile하고 build하여 binary 실행 파일을 생성하는데 필요한 각종 Utility 및 Library의 모음.
 - 기본적으로 Assembler, Linker, C compiler, C library 등으로 구성되어 있다.
 - GNU에서 제공하는 Tool-chain을 사용.
 - GNU GCC compilers for C, C++
 - GNU binary utilities
 - assembler, linker various object file utilities
 - GNU C library

Toolchain 설치

- <http://crztech.iptime.org:8080/Release/Toolchain/cross-4.2.2-eabi.tar.bz2>
- #tar xvfz cross-4.2.2-eabi.tar.bz2
- # sudo mkdir /usr/local/arm
- #cd /usr/local/arm
- # sudo tar xvfj ~/cross-4.2.2-eabi.tar.bz2
- #vim ~/.bashrc
- #source ~/.bashrc
- #env
- #arm-linux-gcc -v
- 최신 Tool chain:<http://www.codesourcery.com/sgpp/lite/arm/portal/release1033>

```
export CROSS_COMPILE=/usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-  
export PATH=/usr/local/arm/4.2.2-eabi/usr/bin:$PATH
```

Toolchain

- Tool chain Test - Test 용 파일 생성

```
#> vi hello.c
#include<stdio.h>
int main(void) {
    printf("Hello Embedded");
    return 0;
}
#> gcc -o hello hello.c
#> arm-linux-gcc -o hello-arm hello.c
#> ls
#> file hello
# ./hello
#> file hello-arm
다음과 같은 결과가 나오면.. 크로스 개발 환경이 제대로 설치가 된 것이다.
```

VI 에디터를 사용해서 샘플 프로그램 생성

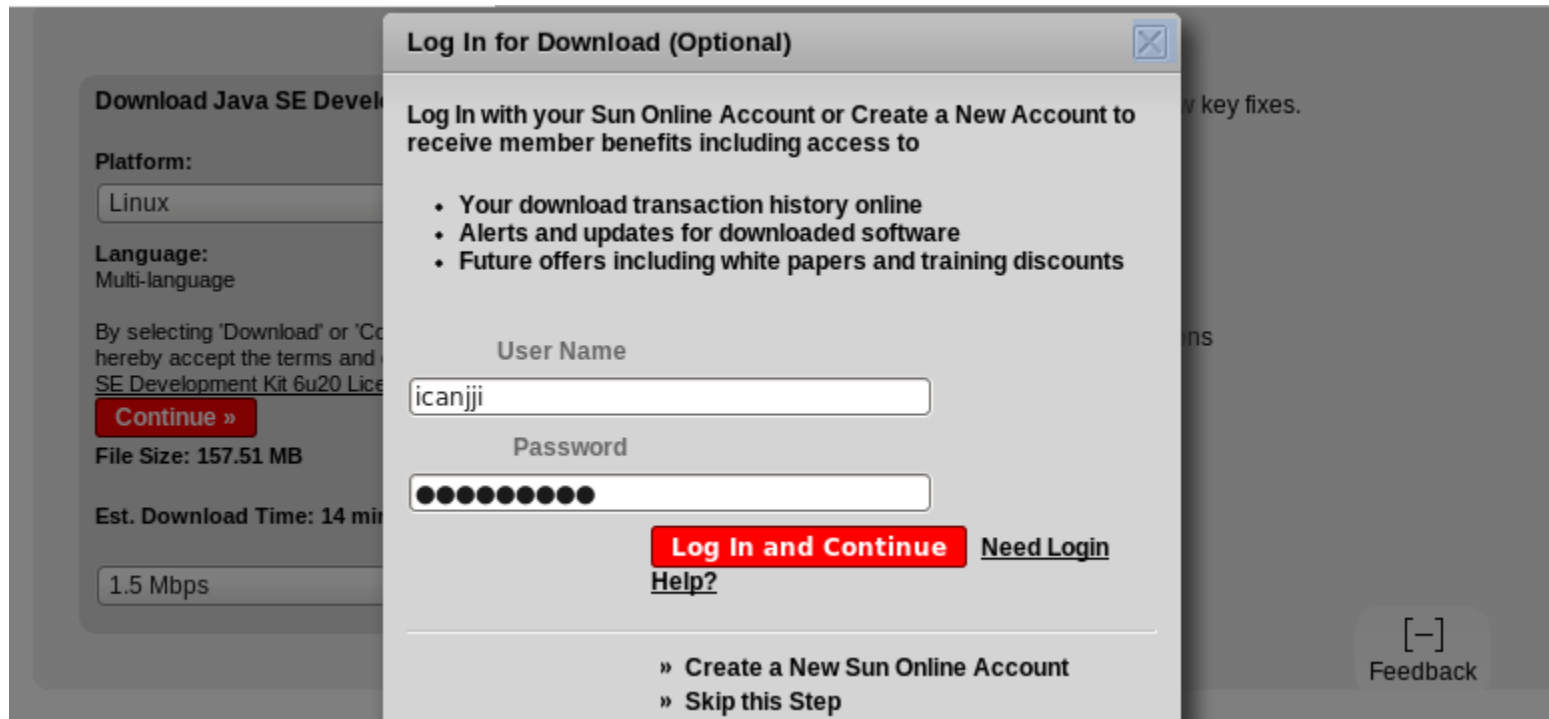
I386 용 프로그램 생성

ARM 용 프로그램 생성

```
[icanjji@localhost ~]$ file hello-arm
hello-arm: ELF 32-bit LSB executable, ARM, version 1 (SYSV), dynamically linked
(uses shared libs), for GNU/Linux 2.6.14, not stripped
```

JAVA JDK설치-1

- <http://java.sun.com/javase/downloads/widget/jdk6.jsp>



The screenshot shows a web page for downloading Java SE Development Kit 6u20. A modal dialog box titled "Log In for Download (Optional)" is overlaid on the page. The dialog box contains the following text and elements:

Log In for Download (Optional)

Log In with your Sun Online Account or Create a New Account to receive member benefits including access to

- Your download transaction history online
- Alerts and updates for downloaded software
- Future offers including white papers and training discounts

User Name

Password

Log In and Continue [Need Login Help?](#)

» Create a New Sun Online Account
» Skip this Step

[-] Feedback

Background page details:
Download Java SE Development Kit 6u20 License Agreement
Platform: Linux
Language: Multi-language
By selecting "Download" or "Continue" you hereby accept the terms and conditions of the Java SE Development Kit 6u20 License Agreement.
Continue »
File Size: 157.51 MB
Est. Download Time: 14 minutes
1.5 Mbps

JAVA JDK설치-2

- Binary Download

There is more information on the available files for download on the [Supported System Configurations](#) page.

Instructions: Click the file name to start the download.

Available Files

File Description and Name	Size
Java SE Development Kit 6u20 jdk-6u20-linux-i586-rpm.bin	76.64 MB
Java SE Development Kit 6u20 jdk-6u20-linux-i586.bin	80.87 MB

JAVA JDK설치-3

- `#sudo mkdir /usr/local/java`
- `#sudo cp jdk-6u20-linux-i586.bin /usr/local/java`
- `#sudo chmod 755 jdk-6u20-linux-i586.bin`
- `#sudo ./jdk-6u20-linux-i586.bin`

JAVA JDK설치-4

- #vim ~/.bashrc

```
export JAVA_HOME=/usr/local/java/jdk1.6.0_20
export CROSS_COMPILE=/usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-
export PATH=$JAVA_HOME/bin:/usr/local/arm/4.2.2-eabi/usr/bin:$PATH
```

- #source ~/.bashrc
- #which java

minicom 설정

- Minicom 설정
- >#yum install minicom
- >#ls /dev/ttyUSB*
- ># minicom -s

```
+-----[configuration]-----+
| Filenames and paths          |
| File transfer protocols     |
| Serial port setup          |
| Modem and dialing           |
| Screen and keyboard         |
| Save setup as dfl           |
| Save setup as..            |
| Exit                         |
| Exit from Minicom           |
+-----+-----+
```

Minicom 설정

- Serial Device : /dev/ttyUSB0
- Baudrate:115200
- Hardware Flow control: NO

```
+-----+
| A -   Serial Device       : /dev/ttyUSB0
| B - Lockfile Location    : /var/lock
| C -   Callin Program     :
| D -   Callout Program    :
| E -   Bps/Par/Bits       : 115200 8N1
| F - Hardware Flow Control : No
| G - Software Flow Control : No
|
| Change which setting? █
+-----+
```

```
+-----[ configuration]-----+
| Filenames and paths
| File transfer protocols
| Serial port setup
| Modem and dialing
| Screen and keyboard
| Save setup as dfl
| Save setup as..
| Exit
| Exit from Minicom
+-----+
```

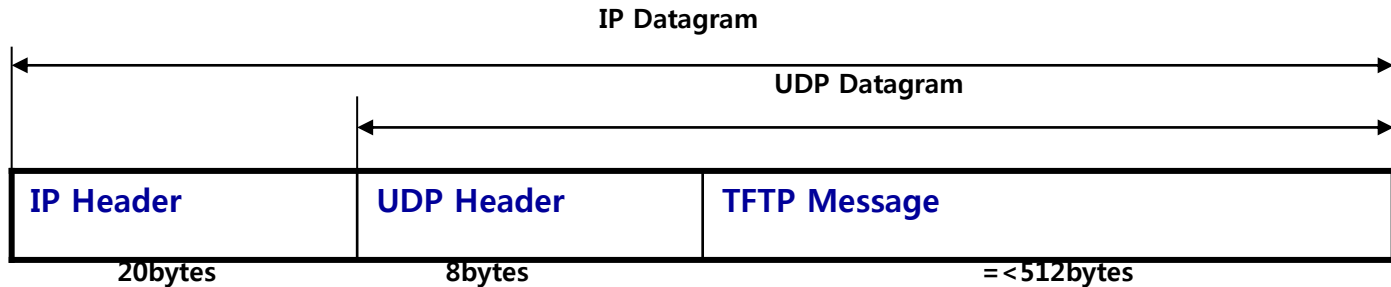
- Save setup as dfl선택

TFTP

- Server로 부터 필요한 File을 읽어와 자신의 Memory에 Load 시킬 때 필요한 Protocol
 - Flash Memory에 맞도록 설계된 단순한 Protocol
 - Disk 없는 System의 가동 프로세서에 이용
 - UDP를 이용하여 Client와 Server 사이의 통신
 - Booting에 필요한 간단한 Program과 BOOTP, RARP, TFTP만 탑재한 시스템에서 많이 이용되며 펌웨어 자동 업그레이드에 이용 가능
- 장점
 - 시스템이 단순하고 간단함
 - 어떤 형태의 전달 서비스상에서도 동작이 가능
 - 시스템 가격이 저렴
- 단점
 - 정보보호 기능이 없음
 - Data에 대한 보장성이 없음

TFTP

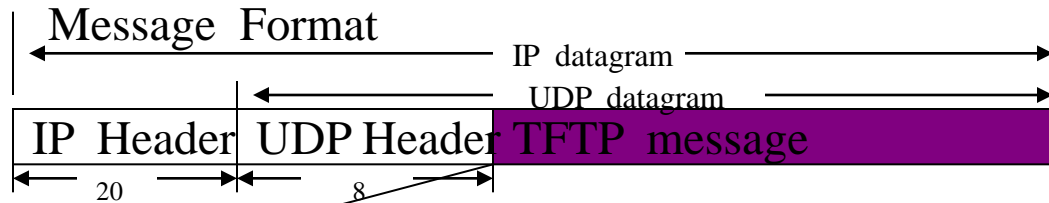
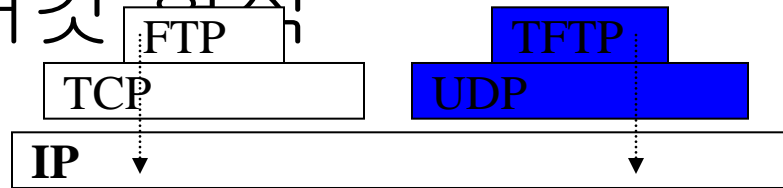
- Tftp 패킷 형식



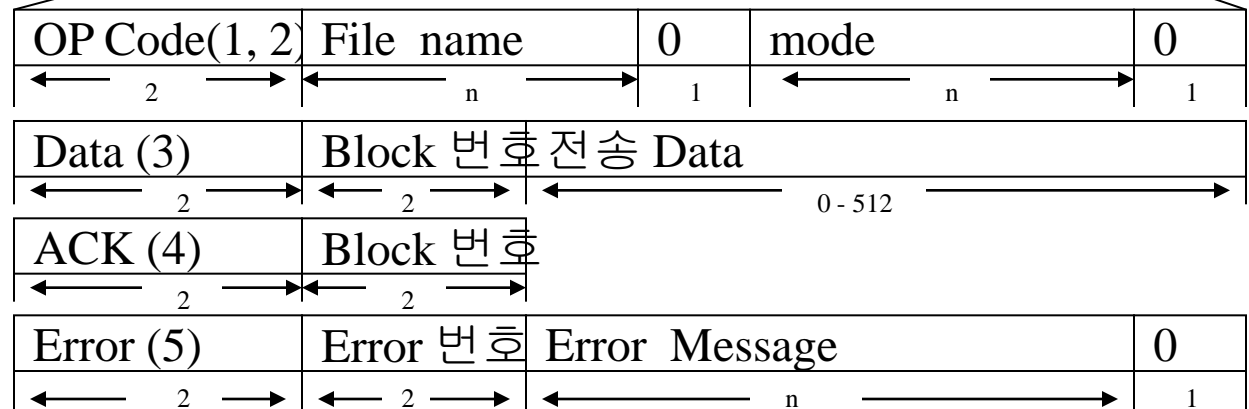
- 512 bytes의 Data Block을 전송한다.
- 각 Block은 아주 간단한 4bytes 헤더와 연결
- Block의 전송 번호는 항상 1부터 시작한다.
- ASCII 또는 Binary 전송을 지원한다.
- Remote File을 Read하거나 Write할 때 사용한다.
- No Checksum
- 높은 처리율보다는 간단함을 강조한 프로토콜이다.

TFTP(2)

- TFTP 패킷 형식



OP Code	Value
RRQ	1
WRQ	2
Data	3
ACK	4
Error	5



TFTP

- TFTP 설치 /미설치 확인 방법

```
#> rpm -qa | grep tftp
```

```
#> yum install tftp*
```

```
[root@localhost icanjji]# yum install tftp*
Loaded plugins: presto, refresh-packagekit
fedora/metalink | 12 kB 00:00
fedora | 4.2 kB 00:00
fedora/primary_db | 9.7 MB 00:08
updates/metalink | 3.7 kB 00:00
updates | 4.5 kB 00:00
updates/primary_db | 4.5 MB 00:03
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package tftp.i686 0:0.49-5.fc12 set to be updated
---> Package tftp-server.i686 0:0.49-5.fc12 set to be updated
--> Processing Dependency: xinetd for package: tftp-server-0.49-5.fc12.i686
--> Running transaction check
---> Package xinetd.i686 2:2.3.14-31.fc12 set to be updated
--> Finished Dependency Resolution
```

TFTP

- TFTP 환경설정

- Host의 /home 밑에 tftpboot라는 이름의 디렉토리를 만듦
- 만일 이 디렉토리를 변경하고 싶다면 위에서 환경설정 한 tftp 파일에 있는 server_args 의 디렉토리를 변경

```
service tftp
{
    disable = no
    socket_type = dgram
    protocol = udp
    wait = yes
    user = root
    server = /usr/sbin/in.tftpd
    server_args = -s /tftpboot
    per_source = 11
    cps = 100 2
    flags = IPv4
}
```

전송될 파일의 위치
디렉토리

TFTP

- 리눅스 부팅 시 자동 활성화 방법

```
#>/etc/init.d/xinetd restart  
#> chkconfig tftp on
```

```
xinetd 기반의 서비스:  
chargen-dgram:  해 제  
chargen-stream: 해 제  
cvs:           해 제  
daytime-dgram: 해 제  
daytime-stream: 해 제  
discard-dgram: 해 제  
discard-stream: 해 제  
echo-dgram:    해 제  
echo-stream:   해 제  
rsync:         해 제  
tcpmux-server: 해 제  
tftp:         활성화  
time-dgram:    해 제  
time-stream:   해 제
```

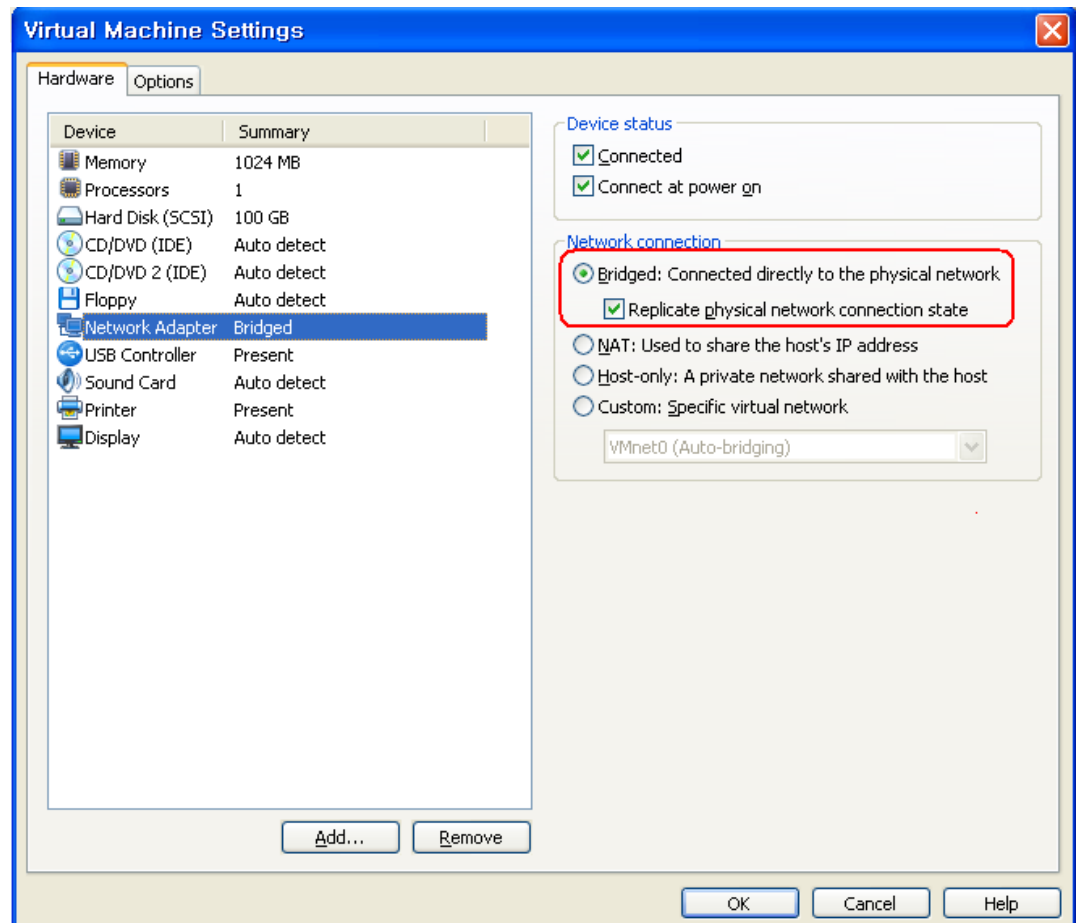
TFTP 실습

- #cd /home/tftpboot
- #touch test.tftp
- #cd /
- # ln -s /home/tftpboot /tftpboot
- #chmod -R 755 /home/tftpboot
- #tftp xxx.xxx.xxx.xxx
- >get test.tftp
- >quit
- #ls

```
[root@localhost ~]# tftp 192.168.1.2
tftp> get test.tftp
tftp> quit
[root@localhost ~]# ls
anaconda-ks.cfg  Downloads          Music              Templates  Videos
Desktop          install.log        Pictures           test
Documents        install.log.syslog Public             test.tftp
```

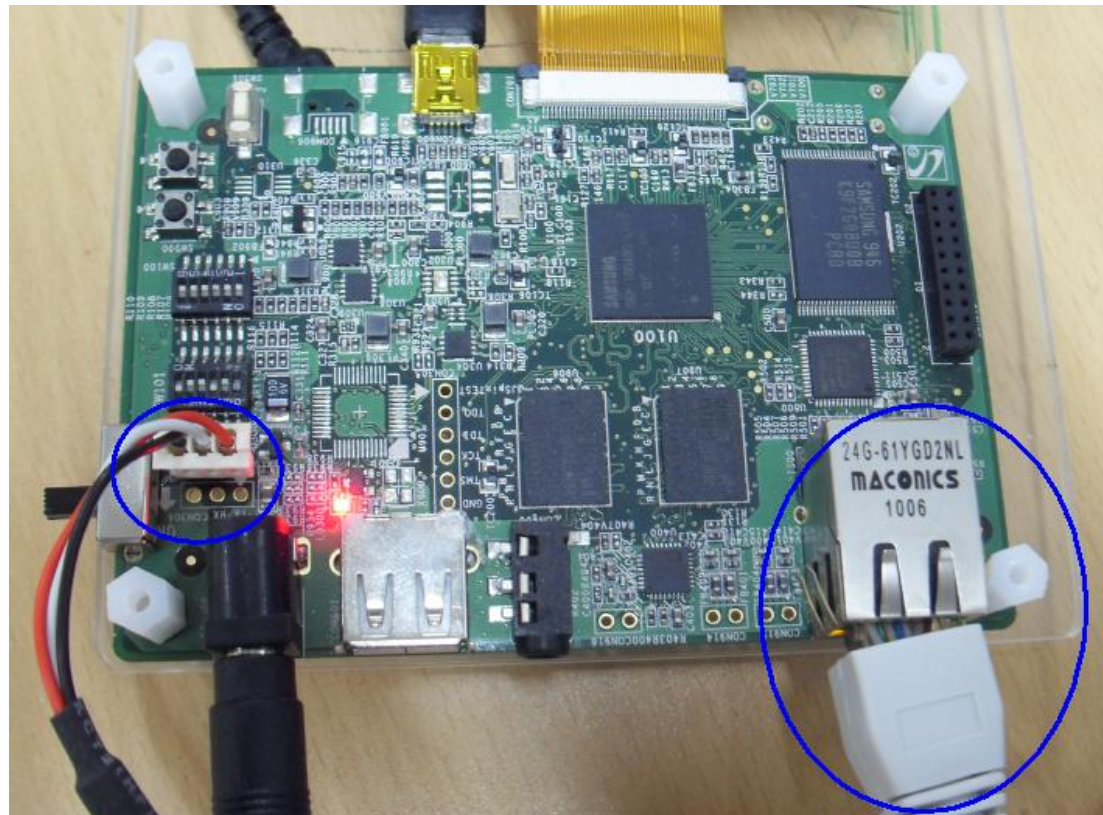
TFTP (VMWare 에서 설정)

- “VM->Setting”



TFTP (HOST PC)

- `#ifconfig eth0 xxx.xxx.xxx.xxx up`
- `#minicom`



TFTP 실습

Host PC 설정

Mango100 u-boot 설정

```
Desktop      install.log      Pictures tes
Documents    install.log.syslog Public tes
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:2A:
          inet addr:192.168.1.2  Bcast:192.168.1.2
          inet6 addr: fe80::20c:29ff:fe2a:d5e9/64
          UP BROADCAST RUNNING MULTICAST  MTU:1500
          RX packets:3580 errors:0 dropped:0 overru
          TX packets:555 errors:0 dropped:0 overru
          collisions:0 txqueuelen:1000
          RX bytes:861258 (841.0 KiB)  TX bytes:56
          Interrupt:19 Base address:0x2024

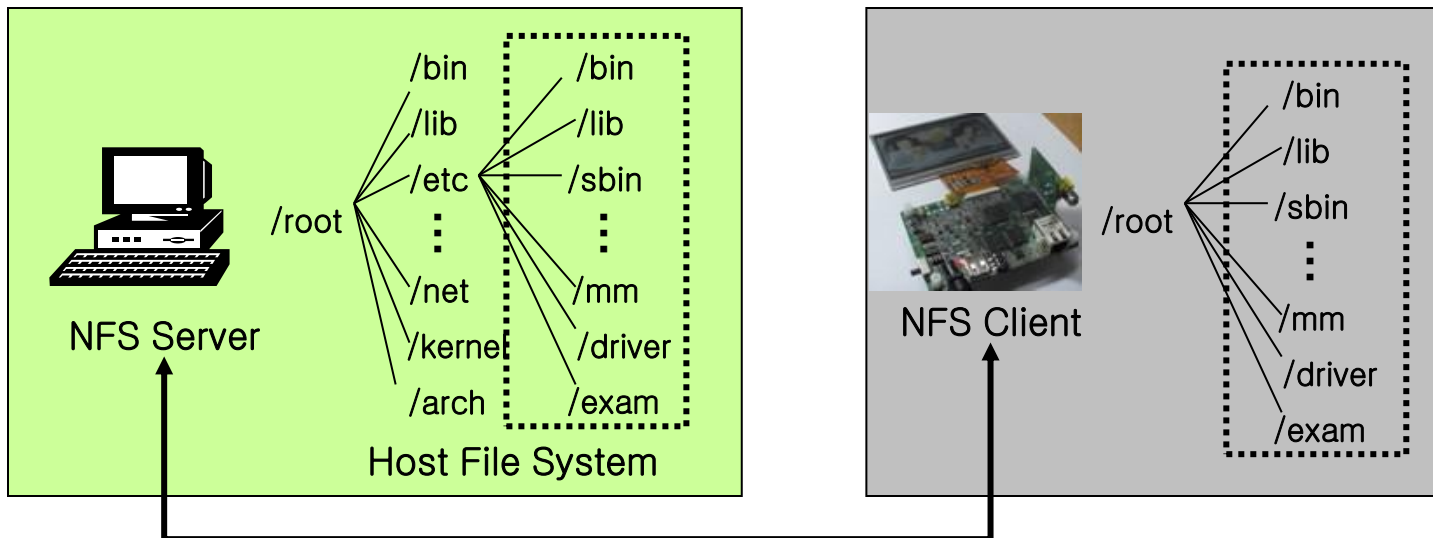
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:37 errors:0 dropped:0 overru
          TX packets:37 errors:0 dropped:0 overru
          collisions:0 txqueuelen:0
          RX bytes:1969 (1.9 KiB)  TX bytes:1969 (

[root@localhost ~]# ifconfig eth0 192.168.1.2 up
[root@localhost ~]#
```

```
MANGO100 # setenv gatewayip 192.168.1.1
MANGO100 # setenv serverip 192.168.1.2
MANGO100 # setenv ipaddr 192.168.1.20
MANGO100 # tftp 21000000 test
smc911x: initializing
smc911x: detected LAN9220 controller
smc911x: autonegotiation timed out
smc911x: MAC 00:40:5c:26:0a:5b
TFTP from server 192.168.1.2; our IP address i
Filename 'test'.
Load address: 0x21000000
Loading: T T T T T T T #
done
MANGO100 # tftp 21000000 test.tftp
smc911x: initializing
smc911x: detected LAN9220 controller
smc911x: autonegotiation timed out
smc911x: MAC 00:40:5c:26:0a:5b
TFTP from server 192.168.1.2; our IP address i
Filename 'test.tftp'.
Load address: 0x21000000
Loading: T T T T T T T #
done
MANGO100 # █
```

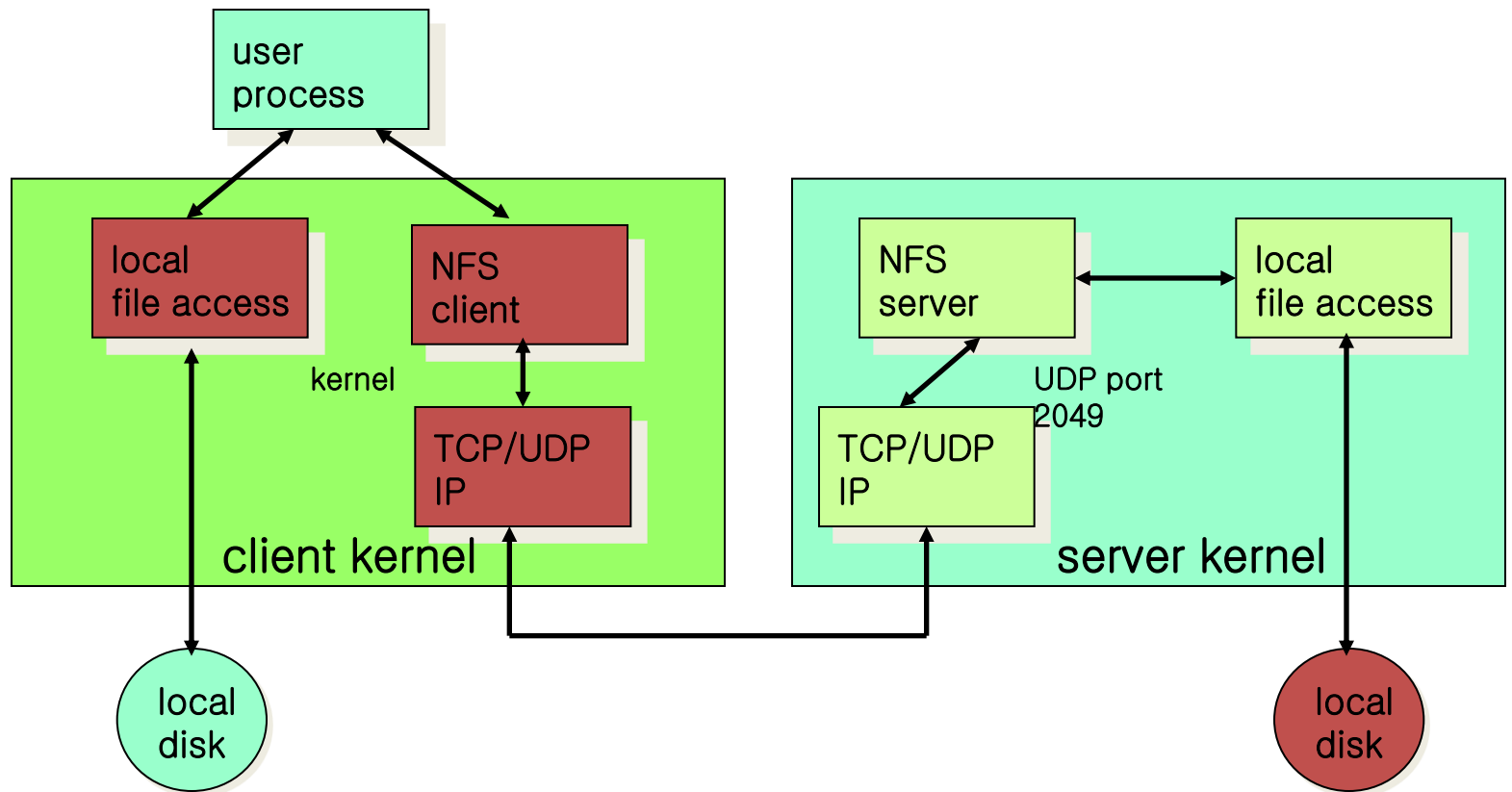
NFS(Network File System)

- NFS란?
 - SUN 사가 개발한 RPC(Remote Procedure Call) 기반 시스템
 - Remote Computer의 파일을 마치 자신의 컴퓨터에 있는 것처럼 이용
 - Server/Client 기반 응용 프로그램
 - FS이 존재하지 않는 Client 시스템에서 원격의 Host 시스템에서 설정된 일부 디렉터리를 이용
 - 임베디드 시스템 개발 시 많이 이용됨



NFS

- NFS 구성도



NFS

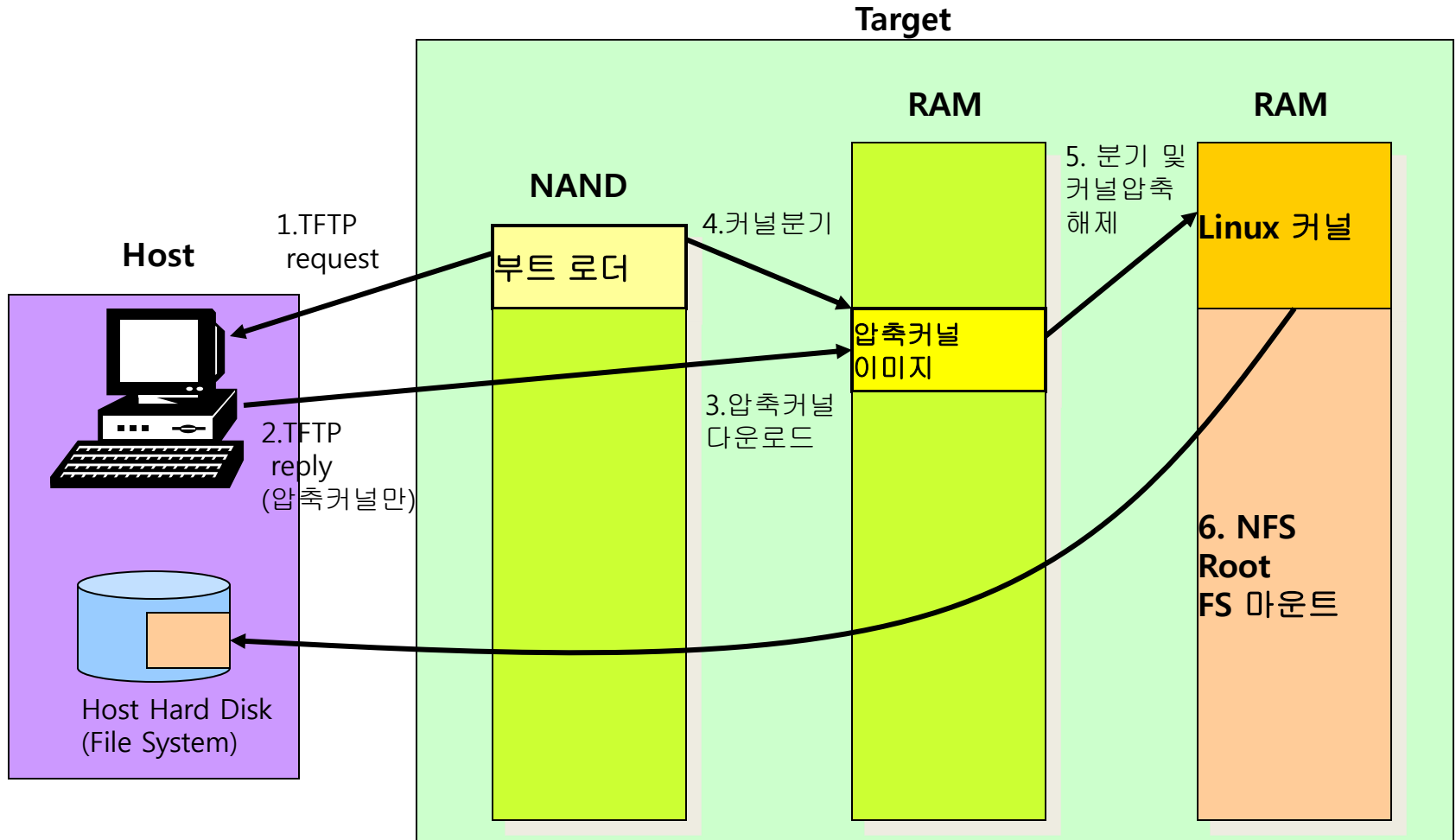
- 장점

- 개발 호스트에서 작업한 것을 NFS 시스템을 이용하여 Target Board의 리눅스 상에서 mount 시켜 사용하면 download 할 필요가 없음.
- 개발 호스트 상의 파일이 Target Board의 리눅스 파일시스템 위에서 접근이 가능하고 실행이 가능.
- 램 디스크 상에서 올리기에 너무 큰 파일도 NFS 상에서는 호스트의 기억 용량에 의존하기 때문에 쉽게 처리 가능.

- 단점

- 특수 파일은 NFS에 연결된 디렉토리에 만들 수 없음.
 - 예: 장치 파일
- 읽고 쓰는 속도가 빠른 파일로는 사용이 곤란.
 - 예: 멀티미디어 파일

TFTP/NFS를 이용한 부팅 시나리오



NFS 서버 설정

```
> # rpm -qa | grep nfs  
> # yum install nfs-*
```

- /etc/exports 파일의 내용에 추가할 사항
 - /home/nfs <tab>보드IP[또는 *](rw,no_root_squash)
예) /home/nfs *(rw, no_root_squash, no_all_squash) 또는
/home/nfs 203.247.100.101(rw, no_root_squash, no_all_squash)

no_root_squash : 원격 루트 사용자가 공유 파일 시스템에서 모든 파일을 변경

no_all_squash : uid,gid root권한으로 연결

rw: read/write 가능하도록 권한 부여

NFS 설정(예)

- NFS 서버측 설정

- NFS 디렉토리 생성 및 권한 설정

```
#>mkdir /home/계정/nfsroot  
#>chmod 777 nfsroot  
#>chgrp nobody nfsroot
```

- NFS 환경설정

```
#>vi /etc/exports  
...  
/home/계정/nfsroot *(rw,no_root_squash,no_all_squash)  
...  
#>
```

NFS 디렉토리, 타겟보드 IP

문장 사이 사이에 띄어쓰기는 없어야 한다

NFS 서버 설정(2)

- /usr/sbin/ntsysv
 - 여러 항목 리스트 중에서 nfs를 찾아서 선택(* 표시)
- NFS 서비스 start 시킴
 - Service nfs start 혹은 /etc/init.d/nfs restart

```
[root@localhost icanjji]# /etc/init.d/nfs restart
NFS mountd를 종료 중입니다: [실패]
NFS 데몬을 종료 중입니다: [실패]
NFS quotas를 종료 중입니다: [실패]
NFS 서비스를 종료 중입니다: [실패]
NFS 서비스를 시작하고 있습니다: [ OK ]
NFS 쿼터를 시작하고 있습니다: [ OK ]
NFS 데몬을 시작함: [ OK ]
NFS mountd를 시작하고 있습니다: [ OK ]
[root@localhost icanjji]# ps -aux | grep nfs
Warning: bad syntax, perhaps a bogus '-'? See /usr/share/doc/procps-3.2.8/FAQ
root      8269  0.0  0.0    0   0 ?        S<    00:13   0:00 [nfsd4]
root      8270  0.0  0.0    0   0 ?        S<    00:13   0:00 [nfsd]
root      8271  0.0  0.0    0   0 ?        S<    00:13   0:00 [nfsd]
-----
root      8272  0.0  0.0    0   0 ?        S<    00:13   0:00 [nfsd]
```

- 위와 같이 나오면 NFS 설정은 완료되었음

NFS/TFTP 실습

– <http://crztech.iptime.org:8080> 에서
Download

– Host PC

```
#>cd /home/계정/nfsroot
#>mv éclair_RFS /home/계정/nfsroot
#>chown -R root.root /home/계정
/nfsroot
#>chmod -R 777 /home/계정/nfsroot
```

– Target Board(u-boot 실행 후)

```
# setenv bootargs "root=/dev/nfs rw
nfsroot=192.168.0.10:/home/icanjji/nfsroot/root
fs
ip=192.168.0.20:192.168.0.10:192.168.0.1:255.25
5.255.0:::off init=/init console=ttySAC1,115200
mem=256M";tftpboot 21000000 zImage;bootm
21000000
```

VNC 서버 설정 (Tip)

- #rpm -qa | grep vnc-server
- # yum install vnc*
- #vi /etc/sysconfig/vncservers

```
VNCSERVERS="1:root"  
VNCSERVERARGS[1]="-geometry 800x600 -nolisten tcp -localhost"
```

- #vncserver :1
- 비밀번호 입력
- #service vncserver restart
- # vi /usr/bin/vncserver

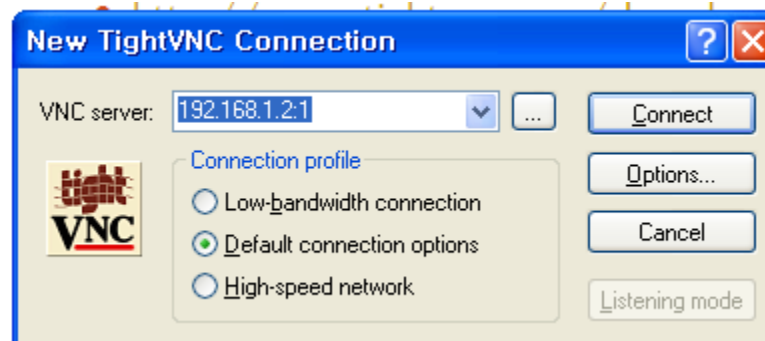
```
$geometry = "1024x768";  
#$depth = 16;
```

- #chkconfig vncserver on

VNC Client 설정

- <http://www.tightvnc.com/download.php>
Download Stable Version, TightVNC 1.3.10

Platform	Link	Description
Windows	download (1,421,291 bytes)	Self-installing package for Windows
	download (943,591 bytes)	Complete set of executables, no installer
	download (248,165 bytes)	Viewer executable, does not require installation



U-Boot란? & 특징

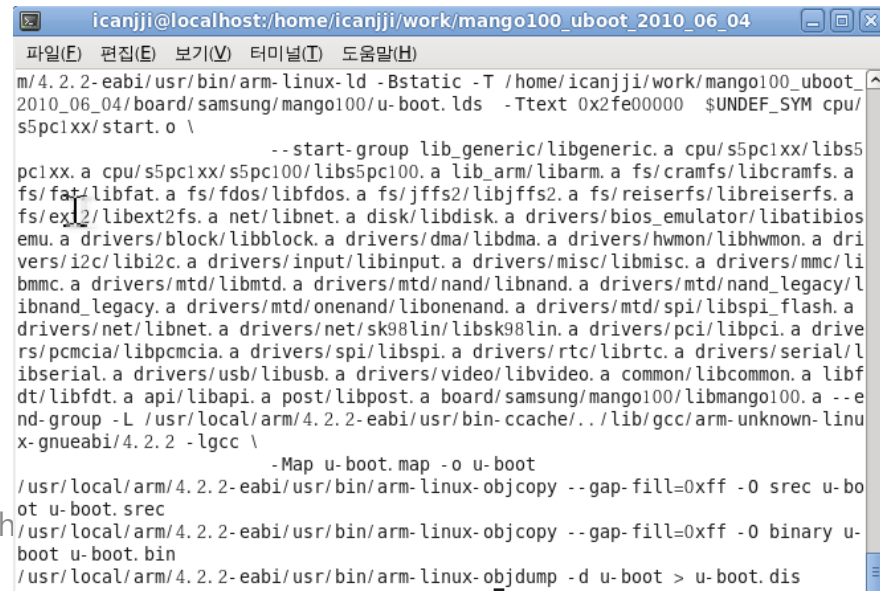
- 부트로더
 - 간단하게 특정 cpu에 OS 혹은 어떤 프로그램을 돌릴 수 있도록 cpu가 동작하는데 필요한 아주 기초적인 부분이나 ROM(or flash), RAM, UART등의 기본적인 디바이스들을 동작할 수 있게 만드는 프로그램.
- Universal Bootloader 의 약자로 간단하게 PowerPC 와 ARM에 기반을 둔 임베디드 보드를 위한 부트로더.
 - Ppcboot와 ARMboot를 썼던 사람들에게 익숙하다..
 - 발전을 해가면서 다양한 platform에 포팅(ppc, arm, mips, x86...)
 - 코드가 깔끔하고 구조가 좋음.
 - 굉장히 강력하고 그나마 쉬운(?) 환경 설정: 많이 쓰이는 cpu를 사용한 상용 evaluation board에 대한 기본 sample코드 존재(수정하여 사용가능)
 - 다른cpu에 대해 같은 명령어 체계를 사용하므로, 다른 platform에 적용을 하더라도 큰 어려움 없이 쉽게 접근가능

U-boot 최신 소스 얻는 법

- <http://www.denx.de/wiki/U-Boot>
- CPU 칩 Band 업체에서 얻는 것이 가장 좋다.
- [망고100 보드](http://www.mangoboard.com) : <http://crztech.iptime.org:8080> 최신 소스 사용

U-boot Download , Compile

- #mkdir ~/work
- #tar xvf mango100-uboot-2010-0629.tgz
- #cd mango100_uboot_2010_06_29
- #make mango100_config
- #make



```
icanjji@localhost:/home/icanjji/work/mango100_uboot_2010_06_04
파일(F) 편집(E) 보기(V) 터미널(T) 도움말(H)
m/4.2.2-eabi/usr/bin/arm-linux-ld -Bstatic -T /home/icanjji/work/mango100_uboot_
2010_06_04/board/samsung/mango100/u-boot.lds -Ttext 0x2fe00000 $UNDEF_SYM cpu/
s5pc1xx/start.o \
-- start-group lib_generic/libgeneric.a cpu/s5pc1xx/libs5
pc1xx.a cpu/s5pc1xx/s5pc100/libs5pc100.a lib_arm/libarm.a fs/cramfs/libcramfs.a
fs/fat/libfat.a fs/fdos/libfdos.a fs/jffs2/libjffs2.a fs/reiserfs/libreiserfs.a
fs/ext2/libext2fs.a net/libnet.a disk/libdisk.a drivers/bios_emulator/libatibios
emu.a drivers/block/libblock.a drivers/dma/libdma.a drivers/hwmon/libhwmon.a dri
vers/i2c/libi2c.a drivers/input/libinput.a drivers/misc/libmisc.a drivers/mmc/li
bmnc.a drivers/mtd/libmtd.a drivers/mtd/nand/libnand.a drivers/mtd/nand_legacy/l
ibnand_legacy.a drivers/mtd/onenand/libonenand.a drivers/mtd/spi/libspi_flash.a
drivers/net/libnet.a drivers/net/sk98lin/libsk98lin.a drivers/pci/libpci.a drive
rs/pcmcia/libpcmcia.a drivers/spi/libspi.a drivers rtc/librtc.a drivers/serial/l
ibserial.a drivers/usb/libusb.a drivers/video/libvideo.a common/libcommon.a libf
dt/libfdt.a api/libapi.a post/libpost.a board/samsung/mango100/libmango100.a --e
nd-group -L /usr/local/arm/4.2.2-eabi/usr/bin-ccache/./lib/gcc/arm-unknown-linu
x-gnueabi/4.2.2 -lgcc \
-Map u-boot.map -o u-boot
/usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-objcopy --gap-fill=0xff -O srec u-bo
ot u-boot.srec
/usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-objcopy --gap-fill=0xff -O binary u-
boot u-boot.bin
/usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-objdump -d u-boot > u-boot.dis
```

Custom Board용 u-boot설정

- 1) copy: board/samsung/mango100 =>
board/samsung/newmango
- 2) copy: include/configs/mango100.h =>
include/configs/newmango.h로 복사
- 3) vi Makefile
mango100_config : unconfig
@./mkconfig \$(@:_config=) arm s5pc1xx mango100
samsung s5pc100
=> 밑에 두줄 추가
newmango_config : unconfig
@./mkconfig \$(@:_config=) arm s5pc1xx newmango
samsung s5pc100
- 4) board/samsung/newmango/u-boot.lds
-> board/samsung/newmango/lowlevel_init.o 으로 수정
- 4) compile
make clobber
make newmango_config
make

Source tree 구조

- /board ---/mango100 : cpu 디렉토리에 있는 cpu들로 만들어진 보드에 대한코드
 - ... 보드에 밀접한 코드들로 보드 초기화 코드, memory bank 설정코드 flash코드, 부트로더가 dram에 위치해야하는 relocation address를 기록한 config.mk, 전체코드의 배치를 지정하는 u-boot.lds라는 링커 스크립트 파일
- /common** : 각종 명령어, user interface main routine
- /cpu ---/**cpu/s5pc1xx** : cpu에 대한 startup코드(cpu초기화)와 serial,
 - ... clock, timer등의 cpu specific한 코드
- /doc : 각종 readme file
- /drivers : 각종 network과 같은 driver
- /examples : 부트로더상에서 실행시키는 standalone app example
- /fs ---/jffs2 : OS(특히 linux)지원 file system
- /include ---/include/asm-arm : 해당 platform에 대한 코드는 include/asm-arm같은식으로 존재.
 - 중요한 파일로 u-boot.h에 board description structure가 존재함.
 - (ppc에서 매우 중요함, arm에서는 덜 중요)
 - ...
 - /include/configs** : 각 보드에 대한 설정파일들이 있습니다, [보드이름].h의 형태.
 - ...
- /lib_arm : u-boot의 arm쪽 C 메인코드들이 있습니다.
- /net : tftp등의 네트워크 코드
- /rtc : Real Time Clock driver
- /tools : mkimage등의 툴

Source Flow

- **startup(cpu/s5pc1xx/start.S)**
 - cpu초기화, dram초기화, 그 후에 부트로더의 ram으로의 relocation, C코드에서의 main인 start_armboot()를 호출함.
- **c쪽의 start코드(lib_arm/board.c)**
 - dram초기화 이후의 flash, uart등등의 주변 디바이스 초기화
- **메인루프(command line shell: common/main.c)**
 - 유닉스 shell과 같은 명령어 처리 루틴. autoboot delay동안 enter키 입력을 기다리는데 그동안 키 입력이 안되면 바로 boot command(설정파일을 보면 나옴)를 실행시킴.
- **app의 실행 혹은 리눅스등의 OS로 부팅(lib_arm/armlinux.c)**
 - 헤더가 붙은 커널의 헤더를 읽어서 압축을 풀고 리눅스로 제어권을 넘겨주는 코드

Mango100 tftp 로 u-boot write

- >#minicom
- 전원인가
- 설명:ipaddr :망고보드 ip, serverip :tftp host ip , gatewayip:Gatewayip

```
MANGO100 # setenv ipaddr 192.168.0.20; setenv gatewayip 192.168.0.1; setenv serverip 192.168.0.4
```

- ># tftp 21000000 u-boot.bin
- ># nand write 21000000 0 40000

망고100 USB OTG 이용 u-boot write

- <http://cafe.naver.com/embeddedcrazyboys/5484>

```
[root@localhost 다운로드]# mv s3c64xx-dltool.tar.gz ../
[root@localhost 다운로드]# cd ../
[root@localhost icanjji]# tar xvfz s3c64xx-dltool.tar.gz
dltool/
dltool/Makefile
dltool/smdk-usbd1
dltool/dltool.c
dltool/readme.txt
[root@localhost icanjji]# cd dltool/
[root@localhost dltool]# ls
Makefile dltool.c readme.txt smdk-usbd1
```

- >#mincom 실행
- >#dnw 21000000

```
[root@localhost dltool]# ./smdk-usbd1 -f /share/tftpboot/u-boot.bin -a 0x21000000
0
SMDK42XX, S3C64XX USB Download Tool
Version 0.20 (c) 2004, 2005, 2006 Ben Dooks <ben-linux@fluff.org>

S3C64XX Detected!
=> found device: bus 001, dev 002
=> loaded 0 bytes from /share/tftpboot/u-boot.bin
=> Downloading 10 bytes to 0x21000000
=> Data checksum 0010
=> usb_bulk_write() returned 10
```

망고100 SD booting 준비물

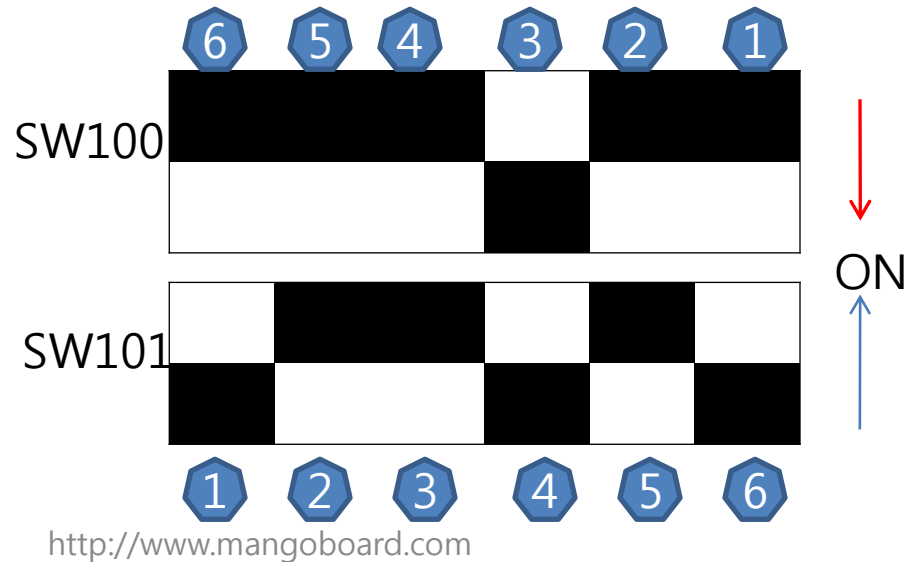
- SDHC 4G Card



- 리눅스 설치 된 PC
- SD boot 망고100 Image

망고100 SD booting Image Write

- MMC /SD Card 리눅스 Host 에 Insert
- SD boot Script 다운 받는곳(<http://crztech.iptime.org:8080>)
- `>#./sdwriter /dev/장치 mango-image`
- 완료가 되면, mmc card를 망고100에 삽입
- SD 부팅 모드로 변경



U-boot build 환경 분석

- Board/\$(Vendor)/\$(BOARD)/Config.mk 파일 분석

(U-boot 가 실행하는 주소 정의)

```
ifndef TEXT_BASE
```

```
//TEXT_BASE = 0xc7e00000 :MMU 사용
```

```
TEXT_BASE = 0x2fe00000 # Physical  
Address
```

```
endif
```

U-boot 1.3.4 Build 환경 분석

- #make mango100 config 실행 시 분석

\$(Top dir)/Makefile 에 아래 코드 실행

```
1
mango100_config :          unconfig
    @$(MKCONFIG) $(@:_config=) arm s5pc1xx mango100 samsung s5pc100
2 3 4 5 6 7

MKCONFIG := $(SRCTREE)/mkconfig
export MKCONFIG
```

- ① Configuration 할 보드 이름 정의
- ② mkconfig 로 \$1인자 의미
- ③ Architecture 인자 \$2인자
- ④ CPU :\$3인자
- ⑤ Board name :\$4
- ⑥ VENDOR :\$5
- ⑦ SOC :\$6

U-boot build 분석

```
mango100_config :      unconfig
    @$(MKCONFIG) $(@:_config=) arm s5pc1xx mango100 samsung s5pc100
```

실행 순서는

① unconfig 실행

② @\$(MKCONFIG) \$(@:_config=) arm s5pc1xx mango100 samsung s5pc100

```
unconfig:
    @rm -f $(obj)include/config.h $(obj)include/config.mk \
        $(obj)board/*/config.tmp $(obj)board/*/*/config.tmp \
        $(obj)include/autoconf.mk $(obj)include/autoconf.mk.dep
```

•include/config.h : include/configs/mango100.h 를 include,mkconfig 에서 create (0

```
echo "/* Automatically generated - do not edit */" >>config.h
echo "#include <configs/$1.h>" >>config.h
```

•Include/config.mk:

```
ARCH = arm
CPU = s5pc1xx
BOARD = mango100
VENDOR = samsung
SOC = s5pc100
```

내용

•board/*/config.tmp, board/*/*/config.tmp : 존재하면 삭제

•Include/autoconfig.mk :makefile 에서 생성

•Include/autoconf.mk.dep : makefile에서 생성

U-boot build 환경 분석

1

```
mango100_config :          unconfig
    @$(MKCONFIG) $(@:_config=) arm s5pc1xx mango100 samsung s5pc100
```

2

3

4

5

6

7

- \$(@:_config=) : @현재 target 값 , 즉 mango100_config, _config=는 null 변경하라는 의미
- 매크로 치환 (Macro substitution):<http://cafe.naver.com/embeddedcrazyboys/7000>
- <http://marvel.incheon.ac.kr/infomation/unix/makefile/GNU-Make-4.html>

```
[ "${BOARD_NAME}" || BOARD_NAME="$1"
[ $# -lt 4 ] && exit 1
[ $# -gt 6 ] && exit 1
echo "Configuring for ${BOARD_NAME} board..."
```

- "\$1"값은 mango100
- \${BOARD_NAME} : mango100 값을 가진다.
- \$# : 넘어오는 인자 수를 의미
- 출력은 "Configuring for mango100 board..."

U-boot build 환경 분석

1

```
mango100_config :          unconfig
    @$(MKCONFIG) $(@:_config=) arm s5pc1xx mango100 samsung s5pc100
```

2

3

4

5

6

7

```
if [ "$SRCTREE" != "$OBJTREE" ] ; then
    mkdir -p ${OBJTREE}/include
    mkdir -p ${OBJTREE}/include2
    cd ${OBJTREE}/include2
    rm -f asm
    ln -s ${SRCTREE}/include/asm-$2 asm
    LNPREFIX=" ../../include2/asm/"
    cd ../include
    rm -rf asm-$2
    rm -f asm
    mkdir asm-$2
    ln -s asm-$2 asm
else
    cd ../include
    rm -f asm
    ln -s asm-$2 asm
fi

rm -f asm-$2/arch

if [ -z "$6" -o "$6" = "NULL" ] ; then
    ln -s ${LNPREFIX}arch-$3 asm-$2/arch
else
    ln -s ${LNPREFIX}arch-$6 asm-$2/arch
fi
```

• SRCTREE와 OBJTREE값은 동일 ,else문 수행

• \$2 값은 arm

Include/asm → Include/asm-arm
Symbol link

[-z "\$6" : 파일이 존재하지 않으면 참
-o : 논리적 OR 연산

Include/asm-arm/arch → Include/arch-s5pc1xx
Symbol link

U-boot build 환경 분석

1

```
mango100_config :          unconfig
    @$(MKCONFIG) $(@:_config=) arm s5pc1xx mango100 samsung s5pc100
```

2

3

4

5

6

7

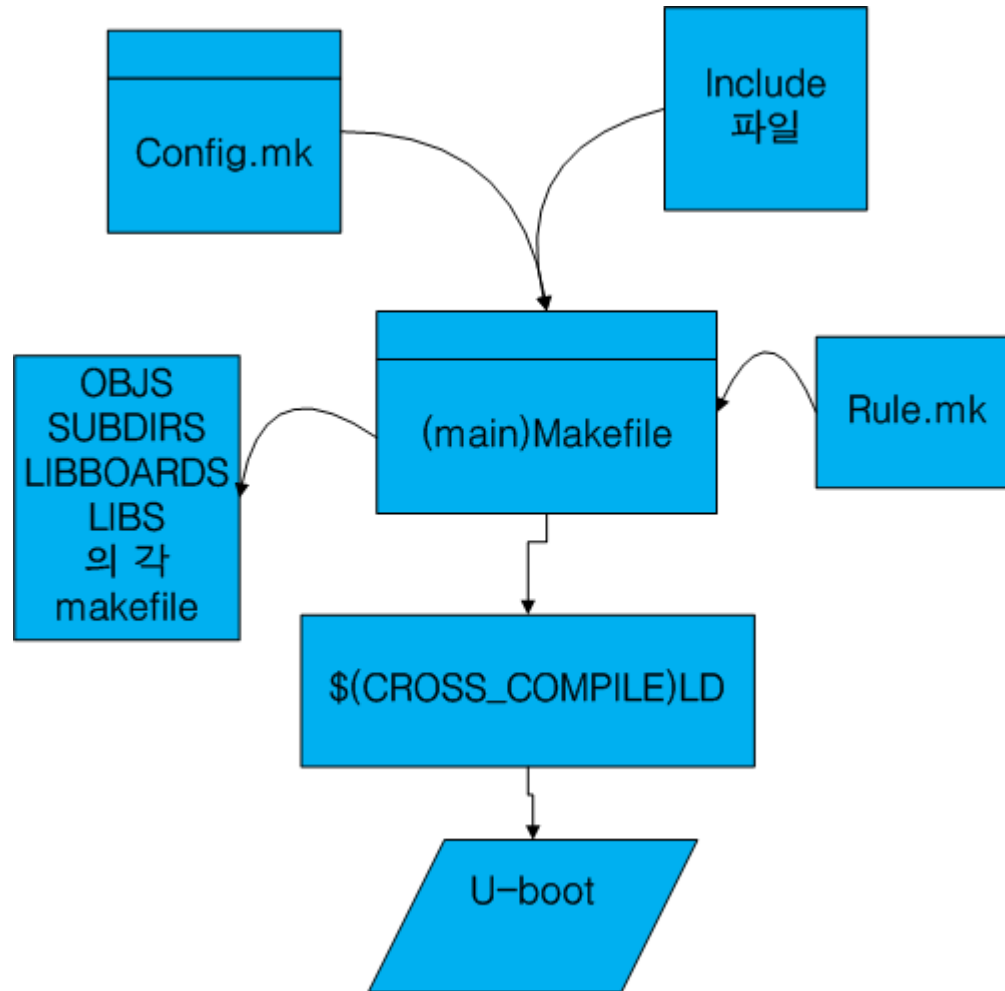
```
if [ "$APPEND" = "yes" ]      # Append to existing config file
then
    echo >> config.h
else
    > config.h                # Create new config file
fi
echo "/* Automatically generated - do not edit */" >>config.h
echo "#include <configs/$1.h>" >>config.h
exit 0
```

Config.h 를 만든다

Config.h 파일 내용은 "include <configs/mango100.h>" 이다

```
[icanjji@localhost include]$ cat config.h
/* Automatically generated - do not edit */
#include <configs/mango100.h>
```

U-boot Build 환경 구조도



리눅스 명령 find (TIP)

- #find . -name "*.ch]" -exec grep 문자열 } ₩; -print
- 예) find . -name "*.ch]" -exec grep mango100 } ₩; -print
- 스크립트 추가 방법

```
#!/bin/sh

if [ "$1" != "" ] && [ "$2" != "" ]; then
    find . -name "$1" -exec grep "$2" } ₩; -print
else
    echo "Usage: $0 *.* abcd"
    exit 1
fi
```

- /usr/bin 디렉토리 vi fin 추가
- #chmod 755 /usr/bin/fin

U-boot build 실행 분석

#make 명령 실행

\$(TOPDIR)/config.mk
\$(TOPDIR)/arm_config.mk
\$(TOPDIR)/rules.mk
...

```
include $(obj)include/autoconf.mk.dep
```

```
all: $(ALL)
```

```
ALL += $(obj)u-boot.srec $(obj)u-boot.bin $(obj)System.map $(U_BOOT_NAND) $(U_BOOT_ONENAND) $(obj)u-boot.dis  
ifeq ($(ARCH),blackfin)  
ALL += $(obj)u-boot.ldr  
endif
```

```
$(obj)u-boot.srec: $(obj)u-boot  
$(OBJCOPY) ${OBJCFLAGS} -O srec $< $@
```

```
$(obj)u-boot: depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)  
UNDEF_SYM=`$(OBJDUMP) -x $(LIBBOARD) $(LIBS) | \#  
sed -n -e 's/.*#$(SYM_PREFIX)__u_boot_cmd_.*#/-u#|/p'|sort|uniq`; \#  
cd $(LNDIR) && $(LD) $(LDFLAGS) $$UNDEF_SYM $(__OBJS) \#  
--start-group $(__LIBS) --end-group $(PLATFORM_LIBS) \#  
-Map u-boot.map -o u-boot
```

U-boot build 실행 분석

```
sinclude $(obj)include/autoconf.mk.dep
```

```
$(obj)include/autoconf.mk.dep: $(obj)include/config.h include/common.h
@$(XECHO) Generating $@ : \
set -e ; \
: Generate the dependancies ; \
$(CC) -x c -DDO_DEPS_ONLY -M $(HOST_CFLAGS) $(CPPFLAGS) \
-MQ $(obj)include/autoconf.mk include/common.h > $@
```

include/config.h == include/configs/mango100.h 내용동
Include/common.h 두개의 파일의 내용을 가지고 온다

\$(CC): \$(CROSS_COMPILE)gcc 값임
-M 옵션: 의존성있는 있는 파일을 구성해 준다.
-MQ (target) : target 이름을 정의 해 준다.

두개의 예제를 결과 비교 실습

```
#> vi hello.c
#include <stdio.h>
int main(void) {
    printf("Hello Mango");
    return 0;
}
#> arm-linux-gcc -M hello.c
```

```
#> vi hello.c
#include <stdio.h>
int main(void) {
    printf("Hello Mango");
    return 0;
}
#> arm-linux-gcc -M -MQ foo.o
hello.c
```


U-boot build 실행 분석

```
1 2 3 4 5 6  
$(obj)u-boot: depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)  
7 UNDEF_SYM=$(OBJDUMP) -x $(LIBBOARD) $(LIBS) | \\  
sed -n -e 's/.*\#($$SYMPREFIX) __u_boot_cmd_.*\#/-u\#1/p'|sort|uniq`;\#  
cd $(LNDIR) && $(LD) $(LDFLAGS) $$UNDEF_SYM $(__OBJS) \#  
--start-group $(__LIBS) --end-group $(PLATFORM_LIBS) \#  
-Map u-boot.map -o u-boot
```

```
depend dep: $(VERSION_FILE)  
for dir in $(SUBDIRS) ; do $(MAKE) -C $$dir _depend ; done
```

```
$(VERSION_FILE):  
@(\ printf '#define U_BOOT_VERSION "U-Boot %s%s"\n' "$(U_BOOT_VERSION)" \  
'$(shell $(CONFIG_SHELL) $(TOPDIR)/tools/setlocalversion $(TOPDIR))' \  
) > $@.tmp  
@cmp -s $@ $@.tmp && rm -f $@.tmp || mv -f $@.tmp $@
```

```
VERSION = 1  
PATCHLEVEL = 3  
SUBLEVEL = 4  
EXTRAVERSION =  
U_BOOT_VERSION = $(VERSION).$(PATCHLEVEL).$(SUBLEVEL)$(EXTRAVERSION)  
VERSION_FILE = $(obj)include/version_autogenerated.h
```

cmp -s file1 file2 : 비교하여 같으면 0
&& 연산자 : cmp 비교 결과 같으면 실행
|| 연산자: cmp 비교 결과 다르면 실행

U-boot build 실행 분석

```

1      2      3      4      5      6
$(obj)u-boot:      depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)
7      UNDEF_SYM=$(OBJDUMP) -x $(LIBBOARD) $(LIBS) | \
      sed -n -e 's/.*\#$(SYM_PREFIX)__u_boot_cmd_.*\#/-u\1/p'|sort|uniq;\#
      cd $(LNDIR) && $(LD) $(LDFLAGS) $$UNDEF_SYM $(__OBJS) \#
      --start-group $(__LIBS) --end-group $(PLATFORM_LIBS) \#
      -Map u-boot.map -o u-boot
  
```

```

depend dep:      $(VERSION_FILE)
                for dir in $(SUBDIRS) ; do $(MAKE) -C $$dir _depend ; done
  
```

```

SUBDIRS = tools \
          examples \
          api_examples
  
```

\$\$dir :tools, examples api_example 값을 가진다.

```

make -C tools _depend
make -C examples _depend
make -C api_example _depend
  
```

형식:make -C 실행디렉토리 타겟
 즉, -C 옵션에 정의된 디렉토리로 가서 Makefile을 수행하며
 target은 _depend 임

```

_depend:      $(obj).depend
$(obj).depend: $(src)Makefile $(TOPDIR)/config.mk $(SRCS)
                @rm -f $@
                @for f in $(SRCS); do \
                    g=`basename $$f | sed -e 's/\(.*\)\.o/'`; \
                    $(CC) -M $(HOST_CFLAGS) $(CPPFLAGS) -MQ $(obj)$$g $$f >> $@ ; \
                done
  
```

U-boot build 실행 분석

```

1      2      3      4      5      6
$(obj)u-boot:      depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)
7      UNDEF_SYM=$(OBJDUMP) -x $(LIBBOARD) $(LIBS) | \
      sed -n -e 's/.*\#($$SYM_PREFIX) __u_boot_cmd_.*\#)/-u\1/p'|sort|uniq`; \
      cd $(LNDIR) && $(LD) $(LDFLAGS) $$UNDEF_SYM $(__OBJS) \
      --start-group $(__LIBS) --end-group $(PLATFORM_LIBS) \
      -Map u-boot.map -o u-boot
  
```

```

$(SUBDIRS):      depend $(obj)include/autoconf.mk
      $(MAKE) -C $$@ all
  
```

Make -C \$SUBDIRS all 수행되면서 실제 빌드됨

```

$(obj)include/autoconf.mk: $(obj)include/config.h
$(XECHO) Generating $$@ : \
set -e ; \
: Extract the config macros ; \
$(CPP) $(CFLAGS) -DDO_DEPS_ONLY -dM include/common.h | \
sed -n -f tools/scripts/define2mk.sed > $$@
  
```

결과

```

CONFIG_BOOTP_BOOTPATH=y
CONFIG_DISPLAY_CPUINFO=y
CONFIG_SETUP_MEMORY_TAGS=y
CONFIG_SERIAL2=y
CONFIG_CMD_LOADB=y
CONFIG_CMD_LOADS=y
CONFIG_CMD_IMI=y
CONFIG_ARM=y
CONFIG_CMD_BDI=y
CONFIG_SERVERIP="192.168.1.2"
CONFIG_BOOTP_SUBNETMASK=y
CONFIG_CMD_REGINFO=y
CONFIG_S3C_USBD=y
CONFIG_S5PC1XX=y
"include/autoconf.mk" 77L, 1781C
  
```

U-boot build 실행 분석

```
1 2 3 4 5 6  
$(obj)u-boot:      depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)  
7 UNDEF_SYM=$(OBJDUMP) -x $(LIBBOARD) $(LIBS) | #  
  sed -n -e 's/.*#($$SYM_PREFIX) __u_boot_cmd_.*#/-u#1/p'|sort|uniq`;#  
  cd $(LNDIR) && $(LD) $(LDFLAGS) $$UNDEF_SYM $(__OBJS) #  
    --start-group $(__LIBS) --end-group $(PLATFORM_LIBS) #  
    -Map u-boot.map -o u-boot
```

```
$(OBJS):      depend $(obj)include/autoconf.mk  
              $(MAKE) -C cpu/$(CPU) $(if $(REMOTE_BUILD),,$@,$(notdir $@))
```

CPU 값은 s5pc1xx

\$(if \$(REMOTE_BUILD),,\$@,\$(notdir \$@)) 의미는
REMOTE_BUILD값이 있으면, OBJS값을 그대로 대입,
없으면, 디렉토리를 제외한 파일이름만 추출하라는 의미

make -C cpu/s5pc1xx start.o 이 수행이 됨

U-boot build 실행 분석(LIBBOARD)

```
1 2 3 4 5 6  
$(obj)u-boot:      depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)  
7  UNDEF_SYM=$(OBJDUMP) -x $(LIBBOARD) $(LIBS) | #  
   sed -n -e 's/.*#($$SYMPREFIX) __u_boot_cmd_.*#)/-u#1/p'|sort|uniq`:#  
   cd $(LNDIR) && $(LD) $(LDFLAGS) $$UNDEF_SYM $(__OBJS) #  
   --start-group $(__LIBS) --end-group $(PLATFORM_LIBS) #  
   -Map u-boot.map -o u-boot
```

```
$(LIBBOARD):      depend $(LIBS) $(obj)include/autoconf.mk  
                  $(MAKE) -C $(dir $(subst $(obj),,$@))
```

```
LIBBOARD = board/$(BOARDDIR)/lib$(BOARD).a  
LIBBOARD := $(addprefix $(obj),$(LIBBOARD))
```

```
BOARDDIR = $(VENDOR)/$(BOARD)  
VENDOR=samsung , BOARD=mango100 값  
따라서 BOARDDIR은 samsung/mango100  
LIBBOARD는  
$(obj)/board/samsung/mango100/libmango100.a
```

U-boot build 실행 분석(LIBBOARD)

```
1 2 3 4 5 6  
$(obj)u-boot: depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)  
7 UNDEF_SYM=$(OBJDUMP) -x $(LIBBOARD) $(LIBS) | \\  
sed -n -e 's/.*#$(SYM_PREFIX)__u_boot_cmd_.*#/-u#1/p'|sort|uniq`;\ \  
cd $(LNDIR) && $(LD) $(LDFLAGS) $$UNDEF_SYM $(__OBJS) \ \  
--start-group $(__LIBS) --end-group $(PLATFORM_LIBS) \ \  
-Map u-boot.map -o u-boot
```

```
$(LIBBOARD): depend $(LIBS) $(obj)include/autoconf.mk  
$(MAKE) -C $(dir $(subst $(obj),,$@))
```

```
depend dep: $(VERSION_FILE)  
for dir in $(SUBDIRS) ; do $(MAKE) -C $$dir _depend ; done
```

```
$(obj)include/autoconf.mk: $(obj)include/config.h  
$(XECHO) Generating $@ ; \ \  
set -e ; \ \  
: Extract the config macros ; \ \  
$(CPP) $(CFLAGS) -DDO_DEPS_ONLY -dM include/common.h | \ \  
sed -n -f tools/scripts/define2mk.sed > $@
```

```
$(LIBS): depend $(obj)include/autoconf.mk  
$(MAKE) -C $(dir $(subst $(obj),,$@))
```

```
$(MAKE) -C $(dir $(subst $(obj),,$@))
```

U-boot build 실행 분석

- 1
- 2
- 3
- 4
- 5
- 6

```
$(obj)u-boot:          depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)
                        UNDEF_SYM=$(OBJDUMP) -x $(LIBBOARD) $(LIBS) | \#
                        sed -n -e 's/.*\#$(SYM_PREFIX)__u_boot_cmd_.*\#/-u\#1/p'|sort|uniq;\#
7 cd $(LNDIR) && $(LD) $(LDFLAGS) $$UNDEF_SYM $(__OBJS) \#
                        --start-group $(__LIBS) --end-group $(PLATFORM_LIBS) \#
                        -Map u-boot.map -o u-boot
```

```
$(LDSCRIPT):          depend $(obj)include/autoconf.mk
                        $(MAKE) -C $(dir $@) $(notdir $@)
```

```
ifndef LDSCRIPT
#LDSCRIPT := $(TOPDIR)/board/$(BOARD)/u-boot.lds.debug
ifeq ($(CONFIG_NAND_U_BOOT),y)
LDSCRIPT := $(TOPDIR)/board/$(BOARD)/u-boot-nand.lds
else
LDSCRIPT := $(TOPDIR)/board/$(BOARD)/u-boot.lds
```

LDSCRIPT값은
/board/samsung/mango100/u-boot.lds
따라서,
\$(MAKE) -C /board/samsung/mango100 u-boot.lds
실행이 됨

U-boot.Ids 분석

```
OUTPUT_FORMAT("elf32-littlearm", "elf32-littlearm", "elf32-littlearm")
/*OUTPUT_FORMAT("elf32-arm", "elf32-arm", "elf32-arm")*/
OUTPUT_ARCH(arm)
ENTRY(_start)
SECTIONS
```

```
. = 0x00000000;
```

: 섹션의 시작 번지를 0x00000000로 설정. 실제로 컴파일된 실행코드인 text가 로딩되고 배치될 번지는 config.mk에 지정된 TEXT_BASE + 0x00000000이 된다.

```
. = ALIGN(4);
```

```
.text :
```

4byte align

```
{
    cpu/s5pc1xx/start.o (.text)
    cpu/s5pc1xx/s5pc100/cpu_init.o (.text)
    board/samsung/mango100/lowlevel_init.o (.text)
    cpu/s5pc1xx/nand_cp.o (.text)
    cpu/s5pc1xx/movi.o (.text)
    *(.text)
}
```

Text section의 맨앞에 cpu/\$cpu/start.o를 배열 (Start-up code)

```
. = ALIGN(4);
```

```
.rodata : { *(.rodata) }
```

```
. = ALIGN(4);
```

```
.data : { *(.data) }
```

```
. = ALIGN(4);
```

```
.got : { *(.got) }
```

```
__u_boot_cmd_start = ;
```

```
.u_boot_cmd : { *(.u_boot_cmd) }
```

```
__u_boot_cmd_end = ;
```

__u_boot_cmd_start와 __u_boot_cmd_end사이의 .u_boot_cmd는 u_boot에서 사용되는 user interface command structure를 이 사이에 배열

```
. = ALIGN(4);
```

```
.mmudata : { *(.mmudata) }
```

```
. = ALIGN(4);
```

```
__bss_start = ;
```

```
.bss : { *(.bss) }
```

```
_end = ;
```

__u_boot_cmd_start, __u_boot_cmd_end, __bss_start, __end는 나중에 C code, start-up code에서 주소계산을 위해 사용함.

U-boot build 실행 분석

```
1 2 3 4 5 6  
$(obj)u-boot: depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)  
7 UNDEF_SYM=`$(OBJDUMP) -x $(LIBBOARD) $(LIBS) |  
sed -n -e 's/.*W$(SYM_PREFIX)__u_boot_cmd_.*W)/-uW1/p'|sort|uniq`;#  
cd $(LNDIR) && $(LD) $(LDFLAGS) $$UNDEF_SYM $(__OBJS) #  
--start-group $(__LIBS) --end-group $(PLATFORM_LIBS) #  
-Map u-boot.map -o u-boot
```

```
UNDEF_SYM=`$(OBJDUMP) -x $(LIBBOARD) $(LIBS) #  
sed -n -e 's/.*W$(SYM_PREFIX)__u_boot_cmd_.*W)/-uW1/p'|sort|uniq`
```

```
UNDEF_SYM=`/usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-objdump -x  
board/samsung/mango100/libmango100.a lib_generic/libgeneric.a cpu/s5pc1xx/libs5pc1xx.a  
cpu/s5pc1xx/s5pc100/libs5pc100.a lib_arm/libarm.a fs/cramfs/libcramfs.a fs/fat/libfat.a fs/fdos/libfdos.a fs/jffs2/libjffs2.a  
fs/reiserfs/libreiserfs.a fs/ext2/libext2fs.a net/libnet.a disk/libdisk.a drivers/bios_emulator/libatibiosemu.a  
drivers/block/libblock.a drivers/dma/libdma.a drivers/hwmon/libhwmon.a drivers/i2c/libi2c.a drivers/input/libinput.a  
drivers/misc/libmisc.a drivers/mmc/libmmc.a drivers/mtd/libmtd.a drivers/mtd/nand/libnand.a  
drivers/mtd/nand_legacy/libnand_legacy.a drivers/mtd/onenand/libonenand.a drivers/mtd/spi/libspi_flash.a  
drivers/net/libnet.a drivers/net/sk98lin/libsk98lin.a drivers/pci/libpci.a drivers/pcmcia/libpcmcia.a drivers/spi/libspi.a  
drivers/rtc/librtc.a drivers/serial/libserial.a drivers/usb/libusb.a drivers/video/libvideo.a common/libcommon.a  
libfdt/libfdt.a api/libapi.a post/libpost.a | #  
sed -n -e 's/.*W(__u_boot_cmd_.*W)/-uW1/p'|sort|uniq`
```

```
실습  
#echo abcd123 | sed 's/W([a-z]*W).*/W1/'
```

U-boot build 실행 분석

```
1 2 3 4 5 6
$(obj)u-boot: depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)
7 UNDEF_SYM=$(OBJDUMP) -x $(LIBBOARD) $(LIBS) | \
  sed -n -e 's/.*\#$(SYM_PREFIX)__u_boot_cmd_.*\#/-u#1/p'|sort|uniq`;\#
  cd $(LNDIR) && $(LD) $(LDFLAGS) $$UNDEF_SYM $(__OBJS) \#
    --start-group $(__LIBS) --end-group $(PLATFORM_LIBS) \#
    -Map u-boot.map -o u-boot
```

```
cd $(LNDIR) && $(LD) $(LDFLAGS) $$UNDEF_SYM $(__OBJS) \#
  --start-group $(__LIBS) --end-group $(PLATFORM_LIBS) \#
  -Map u-boot.map -o u-boot
```

```
cd /home/icanjji/work/u-boot-work/mango100-uboot-2010-06-29 && /usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-ld
-Bstatic -l /home/icanjji/work/u-boot-work/mango100-uboot-2010-06-29/board/samsung/mango100/u-boot.lds -l text
$UNDEF_SYM cpu/s5pc1xx/start.o \#
```

```
--start-group lib_generic/libgeneric.a cpu/s5pc1xx/libs5pc1xx.a
cpu/s5pc1xx/s5pc100/libs5pc100.a lib_arm/libarm.a fs/cramfs/libcramfs.a
fs/fat/libfat.a fs/fdos/libfdos.a fs/jffs2/libjffs2.a fs/reiserfs/libreiserfs.a
fs/ext2/libext2fs.a net/libnet.a disk/libdisk.a drivers/bios_emulator/libatibiosemu.a
drivers/block/libblock.a drivers/dma/libdma.a drivers/hwmon/libhwmon.a drivers/i2c/libi2c.a
drivers/input/libinput.a drivers/misc/libmisc.a drivers/mmc/libmmc.a drivers/mtd/libmtd.a
drivers/mtd/nand/libnand.a drivers/mtd/nand_legacy/libnand_legacy.a drivers/mtd/onenand/libonenand
drivers/mtd/spi/libspi_flash.a drivers/net/libnet.a drivers/net/sk98lin/libsk98lin.a
drivers/pci/libpci.a drivers/pcmcia/libpcmcia.a drivers/spi/libspi.a drivers rtc/librtc.a
drivers/serial/libserial.a drivers/usb/libusb.a drivers/video/libvideo.a common/libcommon.a
libfdt/libfdt.a api/libapi.a post/libpost.a board/samsung/mango100/libmango100.a --end-group
-L /usr/local/arm/4.2.2-eabi/usr/bin-ccache/./lib/gcc/arm-unknown-linux-gnueabi/4.2.2 -lgcc \#
-Map u-boot.map -o u-boot
```

U-boot \$BOARDNAME.h 파일 분석

- #vi include/configs/ 에 위치
- #make mango100_config 명령 수행
mango100.h 이름

```
echo "/* Automatically generated - do not edit */" >>config.h  
echo "#include <configs/$1.h>" >>config.h
```

- Booting Device, 메모리, 프롬프트 이름 등,
u-boot 소스에서 가장 중요한 파일임
- Define한 Value값은 전체 시스템에 영향을 미침

U-boot \$BOARDNAME.h 파일 분석

```
#define CONFIG_S5PC100      1          /* in a SAMSUNG S5PC100 SoC */
#define CONFIG_S5PC1XX     1          /* in a SAMSUNG S5PC1XX Family */
#define CONFIG_MANGO100 1

//#define CONFIG_S5PC100_EVT1
#define CONFIG_S5PC100_EVT2

#define BOOT_ONENAND        0x1
#define BOOT_NAND           0x2
#define BOOT_MMCSDB        0x3
//S5PC100 CPU 는 Memory MAP에서 DRAM 영역 시작 정의
#define MEMORY_BASE_ADDRESS 0x20000000

#if defined(CONFIG_S5PC100_EVT2)
#define CONFIG_MEMORY_UPPER_CODE//u-boot code 위에 STACK,Malloc 메모리 사용
#define CONFIG_MMC // MMC Booting 지원
#define CONFIG_MOVINAND//
#endif

/* input clock of PLL */// 망고100 은 Crystal 12Mhz가 달려있습니다.
#define CONFIG_SYS_CLK_FREQ 12000000 /* the Mango100 has 12MHz input clock */
```

U-boot \$BOARDNAME.h 파일 분석

```
#undef CONFIG_ENABLE_MMU //MMU 미지원 Physical ADDR 사용
```

```
#ifdef CONFIG_ENABLE_MMU
#define virt_to_phys(x) virt_to_phy_smdkc100(x)
#else
#define virt_to_phys(x) (x)
#endif
```

```
#define CONFIG_MEMORY_UPPER_CODE
//#undef CONFIG_MEMORY_UPPER_CODE
```

```
#undef CONFIG_USE_IRQ /* we don't need IRQ/FIQ stuff */
```

```
#define CONFIG_INCLUDE_TEST
```

```
#define CONFIG_ZIMAGE_BOOT //커널 zImage 이미지 지원:압축
```

```
#define CONFIG_IMAGE_BOOT
```

```
#define BOARD_LATE_INIT
```

```
#define CONFIG_SETUP_MEMORY_TAGS
```

```
#define CONFIG_CMDLINE_TAG
```

```
#define CONFIG_INITRD_TAG
```

```
if (*(ulong *)(addr + 9*4) == LINUX_ZIMAGE_MAGIC) {
    printf("Boot with zImage\n");
    addr = virt_to_phys(addr);
    hdr = (image_header_t *)addr;
    hdr->ih_os = IH_OS_LINUX;
    hdr->ih_ep = ntohl(addr);
```

```
    memmove (&images.legacy_hdr_os_copy, hdr,
            sizeof(image_header_t));
```

U-boot \$BOARDNAME.h 파일 분석

```
/*
 * Architecture magic and machine type
 */
#define MACH_TYPE          1826//kernel machine type과 일치해야 함
#define UBOOT_MAGIC        (0x43090000 | MACH_TYPE)

/* Power Management is enabled
#define CONFIG_PM          smdkc100          MACH_SMDKC100          SMDKC100          1826
                          (kernel소스)/arch/arm/tools/mach-types에 위치함

#define CONFIG_DISPLAY_CPUINFO//CPU 정보 제공
#define CONFIG_DISPLAY_BOARDINFO//보드 정보 제공

#undef CONFIG_SKIP_RELOCATE_UBOOT
#undef CONFIG_USE_NOR_BOOT

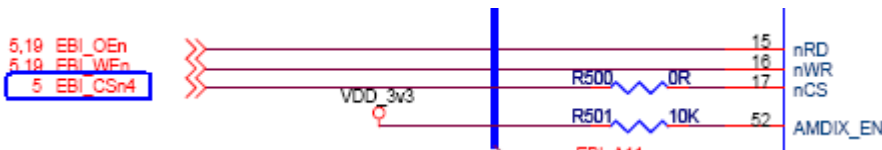
/*
 * Size of malloc() pool
 */
#define CFG_MALLOC_LEN      (CFG_ENV_SIZE + 1024*1024)//0x40000+1MB
#define CFG_GBL_DATA_SIZE  128 /* size in bytes reserved for initial data */

#define CFG_STACK_SIZE      512*1024/* STACK Size 정의 */
```

U-boot \$BOARDNAME.h 파일 분석

```
#define CONFIG_DRIVER_SMC911X 1 /* we have a SMSC9220 on-board */

#ifdef CONFIG_DRIVER_SMC911X
#define CONFIG_DRIVER_SMC911X_16_BIT
#undef CONFIG_DRIVER_CS8900
#define CONFIG_DRIVER_SMC911X_BASE 0xA0000300
#else
#define CONFIG_DRIVER_CS8900 1 /* we have a CS8900 on-board */
#define CS8900_BASE 0x18800300
#define CS8900_BUS16 1 /* the Linux driver does accesses as shorts */
#endif
```



이더넷이 CS Bank4에 연결되어 있으므로 0xA0000000 Address 값을 가진다. [7:0]bit값은 0값을 가져야 한다.

0x9800_0000	0xA000_0000	128MB	SMC Bank 3
0xA000_0000	0xA800_0000	128MB	SMC Bank 4
0xA800_0000	0xB000_0000	128MB	SMC Bank 5

U-boot \$BOARDNAME.h 파일 분석

```
#define CONFIG_SERIAL2      1      /*mango100 uart 1번이 DEBUG Port 임*/
#define CONFIG_USB_OHCI
#undef CONFIG_USB_STORAGE
#define CONFIG_S3C_USBD

//#define USBD_DOWN_ADDR      0xc0000000
#define USBD_DOWN_ADDR      0x20008000

/*****
 * RTC
 *****/

/* allow to overwrite serial and ethaddr */
#define CONFIG_ENV_OVERWRITE

#define CONFIG_BAUDRATE      115200 //Baurate 115200 설정
```

Usb otg 활성화하여 usb download 가능

U-boot \$BOARDNAME.h 파일 분석

```
#define CONFIG_BOOTDELAY      3/*u-boot가 실행 후 커널 이미지를 load하기 전에 3초 기다린다.*/

// MMC2 boot
#define CONFIG_BOOTARGS      "root=/dev/mmcbk0p3 rw rootfstype=ext3 console=ttySAC1,115200 rootdelay=1"

// gnome boot
//#define CONFIG_BOOTARGS      "ubi.mtd=2 root=ubi0:rootfs rootfstype=ubifs rw console=ttySAC1,115200"

// android boot
//#define CONFIG_BOOTARGS      "ubi.mtd=2 root=ubi0:rootfs rootfstype=ubifs rw init=/init console=ttySAC1,115200"

#define CONFIG_ETHADDR        00:40:5c:26:0a:5b
#define CONFIG_NETMASK        255.255.255.0
#define CONFIG_IPADDR         192.168.1.121 /* mango board IP 주소*/
#define CONFIG_SERVERIP       192.168.1.2 /* Host PC IP 주소*/
#define CONFIG_GATEWAYIP      192.168.1.1 /*Gateway IP 주소*/
```

Bootargs 에서
Root : 디바이스 이름
rootfstype: 사용할 root filesystem 타입(ubifs,yaffs2,jffs2,ext3,등)
Rw :Read , Write 가능 하도록
Console: uart 설정
Init: 커널에서 처음 실행하는 process
Ubi.mtd: MTD Partition Block Number

U-boot \$BOARDNAME.h 파일 분석

```
#define CFG_LONGHELP                /* undef to save memory */
#define CFG_PROMPT                  "MANGO100 # " /* Monitor Command Prompt */
#define CFG_CBSIZE                   256 /* Console I/O Buffer Size */
#define CFG_PBSIZE                   384 /* Print Buffer Size */
#define CFG_MAXARGS                  16 /* max number of command args */
#define CFG_BARGSIZE                 CFG_CBSIZE /* Boot Argument Buffer Size */

#define CFG_MEMTEST_START           MEMORY_BASE_ADDRESS /* memtest works on */
#define CFG_MEMTEST_END             MEMORY_BASE_ADDRESS + 0xfe00000 /* 256MB in DRAM */

#undef CFG_CLKS_IN_HZ               /* everything, incl board info, in Hz */

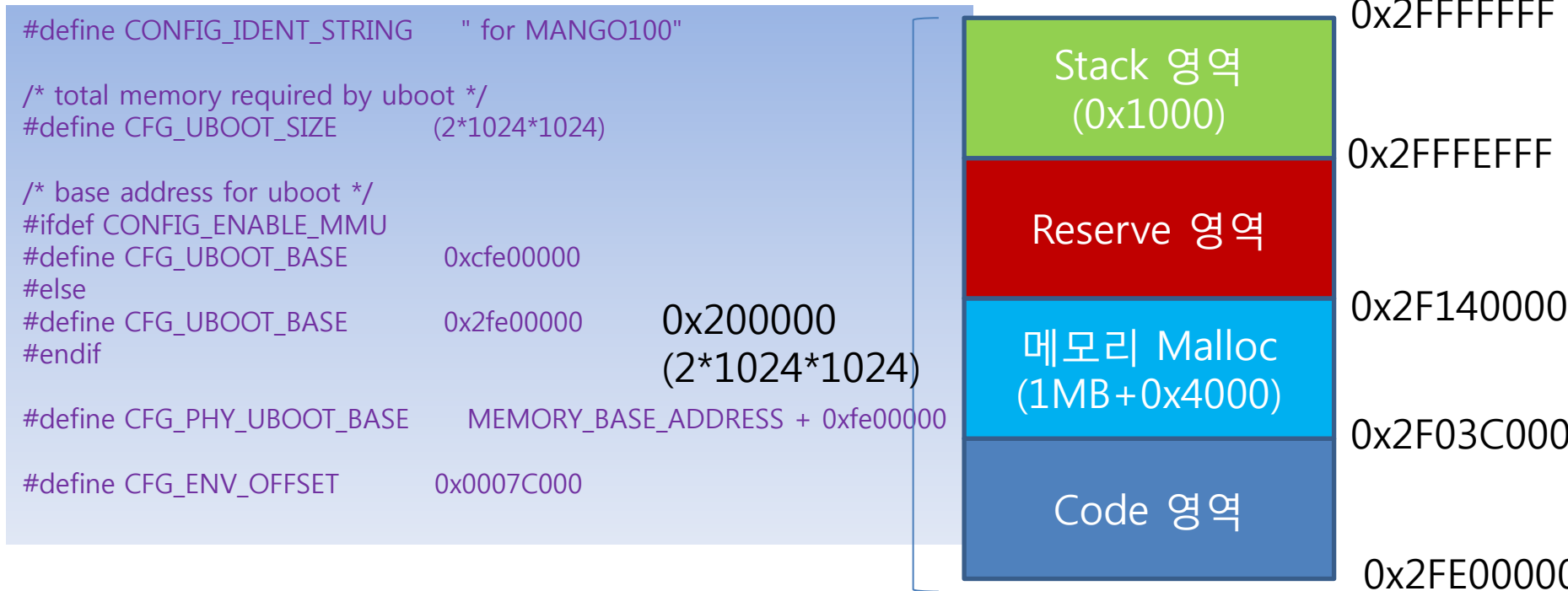
#define CFG_LOAD_ADDR                MEMORY_BASE_ADDRESS /* default load address */

/* the PWM Timer 4 uses a counter of 15625 for 10 ms, so we need */
/* it to wrap 100 times (total 1562500) to get 1 sec. */
#define CFG_HZ                       1562500 /* at PCLK 50MHz */

/* valid baudrates */
#define CFG_BAUDRATE_TABLE          { 9600, 19200, 38400, 57600, 115200 }

/*-----
 * Stack sizes
 *
 * The stack sizes are set up in start.S using the settings below
 */
#define CONFIG_STACKSIZE             0x40000 /* regular stack 256KB */
#ifdef CONFIG_USE_IRQ
#define CONFIG_STACKSIZE_IRQ         (4*1024) /* IRQ stack */
#define CONFIG_STACKSIZE_FIQ         (4*1024) /* FIQ stack */
#endif
```

U-boot \$BOARDNAME.h 파일 분석



U-boot \$BOARDNAME.h 파일 분석

```
/* NAND configuration */
#define CFG_MAX_NAND_DEVICE 1 /*mango100 Nand device 1개*/
#define CFG_NAND_BASE (0xE720000) /*Nand Control Register*/
#define NAND_MAX_CHIPS 1// Chip 갯수

#define NAND_DISABLE_CE() (NFCONT_REG |= (1 << 1))
#define NAND_ENABLE_CE() (NFCONT_REG &= ~(1 << 1))
#define NF_TRANSRnB() do { while(!(NFSTAT_REG & (1 << 0))); } while(0)

#define CFG_NAND_SKIP_BAD_DOT_I 1 /* ".i" read skips bad blocks */
#define CFG_NAND_WP 1
#define CFG_NAND_YAFFS_WRITE 1 /* support yaffs write */

#define CFG_NAND_HWECC //ECC 체크를 HW 처리
//#define CONFIG_NAND_BL1_8BIT_ECC
#undef CFG_NAND_FLASH_BBT

#define CONFIG_BOOTCOMMAND "movi read kernel 20008000;bootm 20008000"
//#define CONFIG_BOOTCOMMAND "nand read 20008000 80000 300000;bootm 20008000"

#define CONFIG_NAND
#define CONFIG_BOOT_NAND
#define CONFIG_BOOT_MOVINAND
#define CFG_ENV_IS_IN_AUTO
```

cpu/s5pc1xx/start.s – Jump vector table

```
/*  
*****  
*  
* Jump vector table as in table 3.1 in [1]  
*  
*****  
*/
```

```
.globl _start  
_start: b reset  
    ldr pc, _undefined_instruction  
    ldr pc, _software_interrupt  
    ldr pc, _prefetch_abort  
    ldr pc, _data_abort  
    ldr pc, _not_used  
    ldr pc, _irq  
    ldr pc, _fiq
```

ARM 이 exception 이 걸리면 각 예외에 따라 무조건 정해진 해당번지로 jump

```
_undefined_instruction: .word undefined_instruction  
_software_interrupt:   .word software_interrupt  
_prefetch_abort:      .word prefetch_abort  
_data_abort:          .word data_abort  
_not_used:            .word not_used  
_irq:                 .word irq  
_fiq:                 .word fiq
```

Exception vector의 위치를
word(4byte)로 정의

```
.balign 16, 0xdeadbeef
```

cpu/s5pc1xx/start.s – startup code (1)

```
_TEXT_BASE:  
    .word    TEXT_BASE
```

각 보드에 대한 config.mk 는 board/XXXXX/config.mk 에 선언되어 있다. 여기서는 TEXT_BASE = 0x33F80000

```
.globl _armboot_start  
_armboot_start:  
    .word    _start
```

_start 라는 위치로 _armboot_start 을 설정

```
/*  
 * These are defined in the board-specific linker script.  
 */
```

```
.globl _bss_start  
_bss_start:  
    .word    __bss_start
```

```
.globl _bss_end  
_bss_end:  
    .word    _end
```

```
#ifdef CONFIG_USE_IRQ  
/* IRQ stack memory (calculated at run-time) */  
.globl IRQ_STACK_START  
IRQ_STACK_START:  
    .word    0x0badc0de
```

```
/* IRQ stack memory (calculated at run-time) */  
.globl FIQ_STACK_START  
FIQ_STACK_START:  
    .word    0x0badc0de  
#endif
```

cpu/s5pc1xx/start.s – SVC32 mode

```
reset:
    /*
     * set the cpu to SVC32 mode and IRQ & FIQ disable
     */
    mrs    r0,cpsr
    bic    r0,r0,#0x1f
    orr    r0,r0,#0xd3
    msr    cpsr,r0
```

CPSR 을 R0로 읽어온다.

$R0 := R0 \& 0xFFFFFE0$ (하위 5 비트의 mode 비트를 clear 함)

$R0 := R0 | 0x11010011$ (supervisor 모드로 변경)

R0 의 값을 CPSR로 로딩

cpu/s5pc1xxx/start.s – cpu_init_crit

CP15 : system control processor
-cache, MMU, protection system,
clocking mode, big /litter endian
operation 과 같은 arm920t 의 다른
시스템 옵션들을 설정하고 제어함.
- MCR, MRC 명령으로 접근 가능

```
cpu_init_crit:
/*
 * Invalidate L1 I/D
 */
mov    r0, #0                @ set up for MCR
mcr    p15, 0, r0, c8, c7, 0 @ invalidate TLBs
mcr    p15, 0, r0, c7, c5, 0 @ invalidate icache

/*
 * disable MMU stuff and caches
 */
mrc    p15, 0, r0, c1, c0, 0
bic    r0, r0, #0x00002000    @ clear bits 13 (--V-)
bic    r0, r0, #0x00000007    @ clear bits 2:0 (-CAM)
orr    r0, r0, #0x00000002    @ set bit 1 (--A-) Align
orr    r0, r0, #0x00000800    @ set bit 12 (Z---) BTB
mcr    p15, 0, r0, c1, c0, 0
```

4.2 OPERATING MODE REGISTER (OMR, R, ADDRESS = 0XE000_0004)

Field	Bit	Description
Reserved	[31:18]	Read as zero
Reserved	[17]	Read as one
Reserved	[16]	Read as one
Reserved	[15:14]	Read as zero
<u>NFMOD</u>	<u>[13:8]</u>	<u>The value of the XNFMOD[5:0] pad</u>
Reserved	[7:5]	Read as zero
<u>OM</u>	<u>[4:0]</u>	<u>The value of the XOM[4:0] pad</u>

OM[0:4] Pin 정보를 읽어서
information REG 에 저장

```
/* Read booting information */
ldr    r0, =PRO_ID_BASE
ldr    r1, [r0, #OMR_OFFSET]
bic    r2, r1, #0xffffffff9
cmp    r2, #0x0
moveq  r3, #BOOT_NAND
cmp    r2, #0x2
moveq  r3, #BOOT_ONENAND
cmp    r2, #0x4
moveq  r3, #BOOT_MMCSO

ldr    r0, =INF_REG_BASE
str    r3, [r0, #INF_REG3_OFFSET]
```


cpu/s5pc1xxx/start.s – Boot mode

```
ldr    r0, =0xff000fff
bic    r1, pc, r0
ldr    r2, _TEXT_BASE
bic    r2, r2, r0
cmp    r1, r2
beq    after_copy

ldr    r0, =INF_REG_BASE
ldr    r1, [r0, #INF_REG3_OFFSET]
cmp    r1, #BOOT_NAND
beq    nand_boot
cmp    r1, #BOOT_ONENAND
beq    onenand_boot
cmp    r1, #BOOT_MMCSDBOOT
beq    mmcboot

nand_boot:
mov    r0, #0x1000
bl    copy_from_nand
b     after_copy
```

Info register에서 정보를 읽어서
Nand boot, mmcboot로 분기

Lowlevel_init ?

```
bl lowlevel_init /* go setup pll,mux,memory */
```

Cpu/s5pc1xx/start.s에서 call

```
lowlevel_init:
    ldr    sp, =0x37ff0 /* setup temp stack pointer */
    sub   sp, sp, #12
    push {lr}

    /* IO Retention release */
    ldr    r0, =(ELFIN_CLOCK_POWER_BASE + OTHERS_OFFSET)
    ldr    r1, [r0]
    ldr    r2, =IO_RET_REL
    orr   r1, r1, r2
    str   r1, [r0]

    /* Disable Watchdog */
    ldr    r0, =0xEA200000
    mov   r1, #0
    str   r1, [r0]

    /* CS0 - 16bit SRAM, Enable nBE */
    ldr    r0, =ELFIN_SROM_BASE
    mov   r1, #0x9
    str   r1, [r0]

    /* init system clock */
    bl    system_clock_init
    /* for UART */
    bl    uart_asm_init

    bl    dma_init
#ifdef CONFIG_NAND
    /* NAND GPIO PIN MUX */
    bl    nand_pin_mux

    /* simple init for NAND */
    bl    nand_asm_init
#endif
```

Watchdog, Clock, UART, Nand 을 초기화

cpu/s5pc1xx/start.s – setup the stack

```
stack_setup:
#if defined(CONFIG_MEMORY_UPPER_CODE)
    ldr    sp, =(CFG_UBOOT_BASE + CFG_UBOOT_SIZE - 0x1000)
#else
    ldr    r0, _TEXT_BASE          /* upper 128 KiB: relocated uboot
    sub    r0, r0, #CFG_MALLOC_LEN /* malloc area
    sub    r0, r0, #CFG_GBL_DATA_SIZE /* bdfinfo
#if defined(CONFIG_USE_IRQ)
    sub    r0, r0, #(CONFIG_STACKSIZE_IRQ+CONFIG_STACKSIZE_FIQ)
#endif
    sub    sp, r0, #12            /* leave 3 words for abort-stack
```

Stack address 를 지정하였으므로 드디어 RAM에서 코드가 동작할 수 있는 조건이 마련되었다.

Stack pointer의 최상위 3 개 word를 비워두는 것은 abort exception 발생하면, exception 발생하기 직전 PC 와 CPRS를 저장하여 debugging 정보로 이용하기 위해서이다.

```
clear_bss:
    ldr    r0, _bss_start
    ldr    r1, _bss_end
    mov    r2, #0x00000000

clbss_l:
    str    r2, [r0]
    add    r0, r0, #4
    cmp    r0, r1
    ble   clbss_l

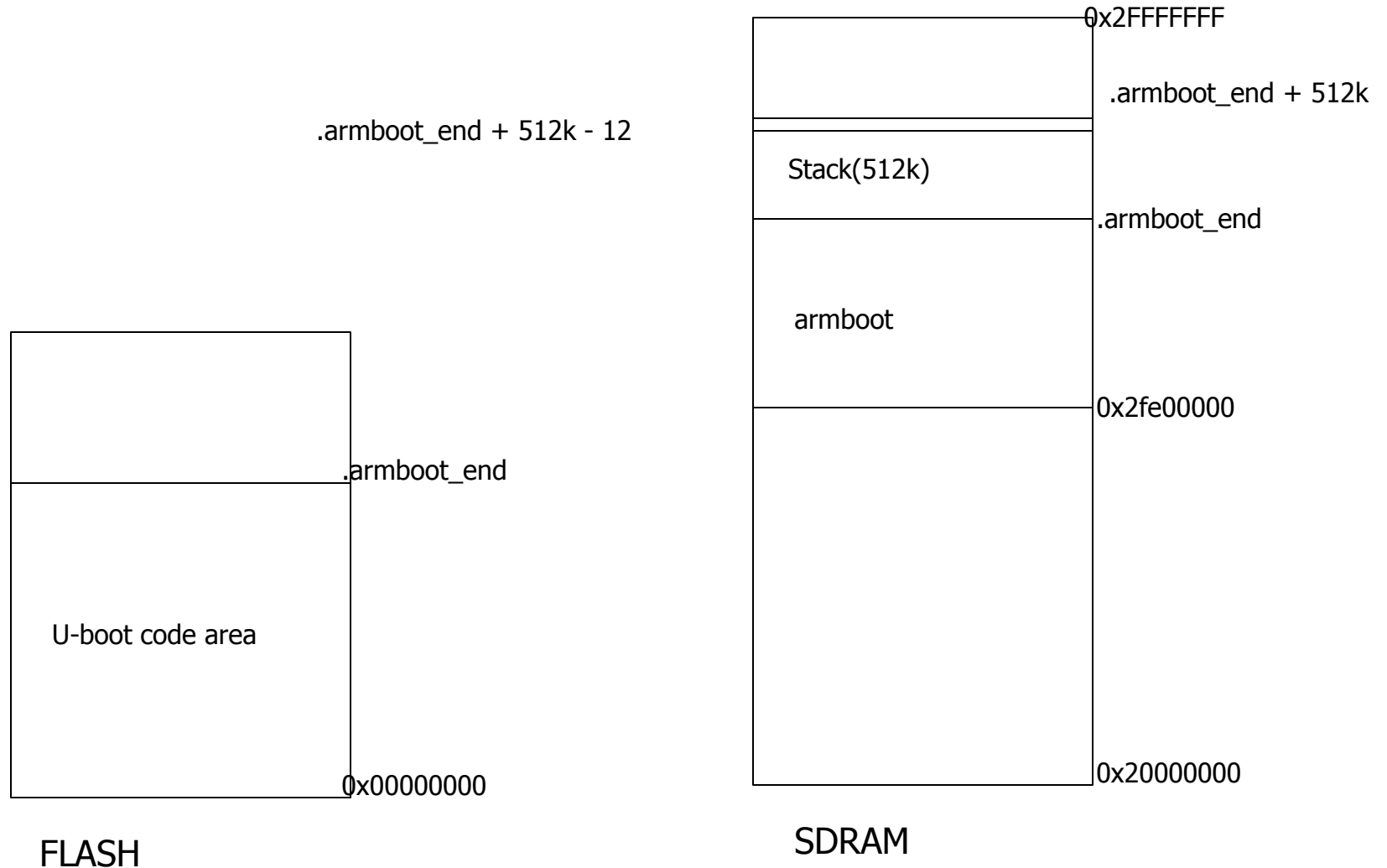
    ldr    pc, _start_armboot
```

Global 변수 들 0 으로 초기화

여기서 RAM의 _start_armboot 로 brach 하여 RAM에 있는 코드가 동작하게 된다.

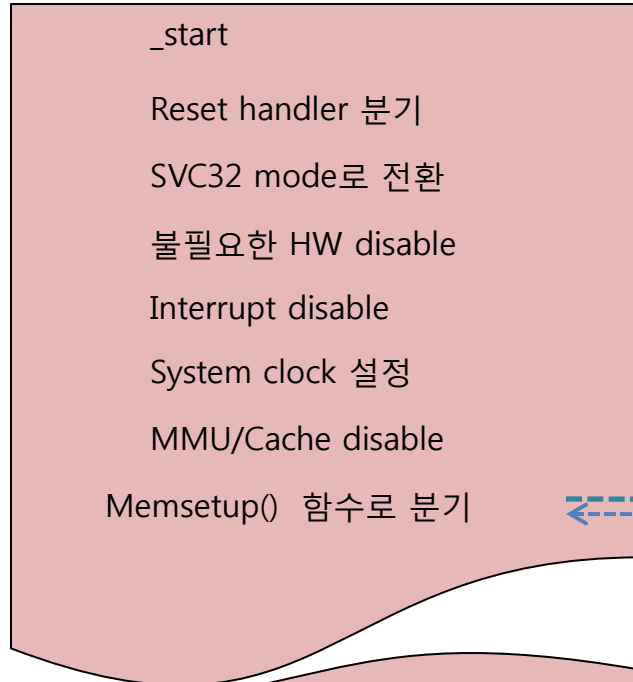
Lib_arm/board.c 의 Start_armboot() 로 JUMP

cpu/s5pc1xxx/start.s – setup the stack $\bar{\bar{}}$ memory map

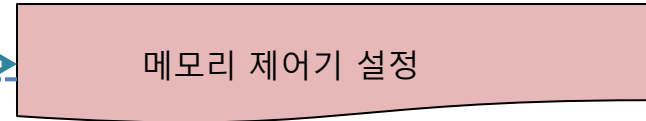


U-boot 초기화 Diagram

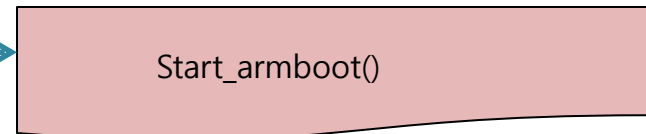
Cpu/s5pc1xx/start.s



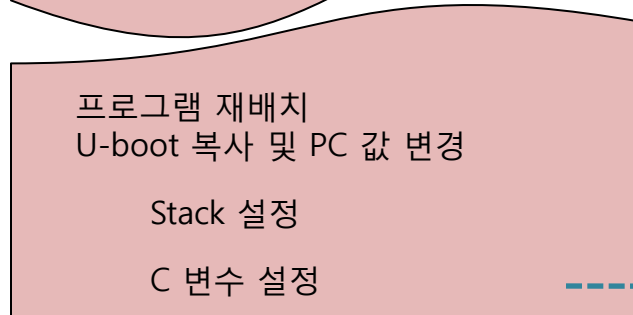
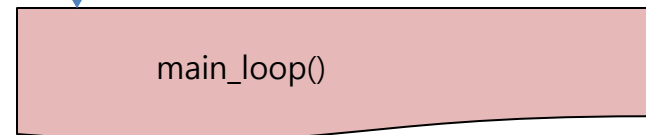
Cpu/s5pc1xx/s5pc100/cpu_init.S



Lib_arm/board.c



common/main.c



U-Boot 실행 순서 개요

[ASM] startup코드
(cpu/s5pc1xx/start.S)

CPU 초기화, SDRAM 초기화,
SDRAM으로 재배치,
start_armboot()호출

[C]코드 start_armboot()
(lib_arm/board.c)

NAND 플래시, 시리얼,
네트워크 카드 등 초기화
main_loop() 호출

main_loop()
(common/main.c)

명령어 처리 루틴, run_command()
자동 부팅

Lib_arm/board.c – start_armboot()

```
init_fnc_t *init_sequence[] = {
    cpu_init, /* basic cpu dependent setup */
#ifdef CONFIG_SKIP_RELOCATE_UBOOT
    reloc_init, /* Set the relocation done flag, must
                do this AFTER cpu_init(), but as soon
                as possible */
#endif
    board_init, /* basic board dependent setup */
    interrupt_init, /* set up exceptions */
    env_init, /* initialize environment */
    init_baudrate, /* initialize baudrate settings */
    serial_init, /* serial communications setup */
    console_init_f, /* stage 1 init of console */
    display_banner, /* say that we are here */
#ifdef CONFIG_DISPLAY_CPUINFO
    print_cpuinfo, /* display cpu info (and speed) */
#endif
#ifdef CONFIG_DISPLAY_BOARDINFO
    checkboard, /* display board info */
#endif
#ifdef CONFIG_HARD_I2C || defined(CONFIG_SOFT_I2C)
    init_func_i2c,
#endif
    dram_init, /* configure available RAM banks */
    display_dram_config,
    NULL,
};
```

Lib_arm/board.c – start_armboot()

Lib_arm/board.c

Mem_malloc_init()

Env_relocate()

Devices_init)

Jumtable_r)

Concole_init_r()

Enable_interrupt()

Main_loop()

Lib_arm/board.c

Common/env_common.c

Common/devices.c

Common/exports.c

Common/console.c

Cpu/s5pc1xxx/interrupts.c

common/main.c

Main_loop()

```
for (;;) {
#ifdef CONFIG_BOOT_RETRY_TIME
    if (rc >= 0) {
        /* Saw enough of a valid command to
         * restart the timeout.
         */
        reset_cmd_timeout();
    }
#endif

    len = readline (CFG_PROMPT);

    flag = 0; /* assume no special flags for now */
    if (len > 0)
        strcpy (lastcommand, console_buffer);
    else if (len == 0)
        flag |= CMD_FLAG_REPEAT;
#ifdef CONFIG_BOOT_RETRY_TIME
    else if (len == -2) {
        /* -2 means timed out, retry autoboot
         */
        puts ("Timed out waiting for command\n");
#ifdef CONFIG_RESET_TO_RETRY
        /* Reinit board to run initialization code again */
        do_reset (NULL, 0, 0, NULL);
#endif
        return; /* retry autoboot */
    }
#endif

    if (len == -1)
        puts ("<INTERRUPT>\n");
    else
        rc = run_command (lastcommand, flag);

    if (rc <= 0) {
        /* invalid command or not repeatable, forget it */
        lastcommand[0] = 0;
    }
}
} ? end for ; ; ?
```

Run_command()

```
/* find macros in this token and replace them */
process_macros (token, finaltoken);

/* Extract arguments */
argc = parse_line (finaltoken, argv);

/* Look up command in command table */
if ((cmdtp = find_cmd(argv[0])) == NULL) {
    printf ("Unknown command '%s' - try 'help'\n", argv[0]);
    rc = -1; /* give up after bad command */
    continue;
}

/* found - check max args */
if (argc > cmdtp->maxargs) {
    printf ("Usage: %s\n", cmdtp->usage);
    rc = -1;
    continue;
}

#if (CONFIG_COMMANDS & CFG_CMD_BOOTD)
/* avoid "bootd" recursion */
if (cmdtp->cmd == do_bootd) {
#ifdef DEBUG_PARSER
    printf ("[%s]\n", finaltoken);
#endif
    if (flag & CMD_FLAG_BOOTD) {
        puts ("'bootd' recursion detected\n");
        rc = -1;
        continue;
    }
    else
        flag |= CMD_FLAG_BOOTD;
}
#endif /* CFG_CMD_BOOTD */

/* OK - call function to do the command */
if ((cmdtp->cmd) (cmdtp, flag, argc, argv) != 0) {
    rc = -1;
}
```

Command의 인자를 추출...

```
cmd_tbl_t *find_cmd (const char *cmd)
{
    cmd_tbl_t *cmdtp;
    cmd_tbl_t *cmdtp_temp = &__u_boot_cmd_start; /*Init value */
    const char *p;
    int len;
    int n_found = 0;

    /*
     * Some commands allow length modifiers (like "cp.b");
     * compare command name only until first dot.
     */
    len = ((p = strchr(cmd, '.')) == NULL) ? strlen (cmd) : (p - cmd);

    for (cmdtp = &__u_boot_cmd_start;
         cmdtp != &__u_boot_cmd_end;
         cmdtp++) {
        if (strncmp (cmd, cmdtp->name,
                    (len == strlen (cmdtp->name)) ?
                    return cmdtp; /* full match */

        cmdtp_temp = cmdtp; /* a
        n_found++;
        return

    }
    if (n_found == 1) { /* exactly one match */
        return cmdtp_temp;
    }
    return NULL; /* not found or ambiguous command */
} ? end find_cmd ?
```

Command list 에서 Command 와 일치하는 것을 Search

Match된 명령을 처리할 handle 함수 포인터를 return

Command를 실행하기 위해 함수를 호출함

U_BOOT_CMD

```
/*
 * Monitor Command Table
 */

struct cmd_tbl_s {
    char    *name;        /* Command Name */
    int     maxargs; /* maximum number of arguments */
    int     repeatable; /* autorepeat allowed? */
                /* Implementation function */
    int     (*cmd)(struct cmd_tbl_s *, int, int, char *[]);
    char    *usage;      /* Usage message (short) */
#ifdef CFG_LONGHELP
    char    *help;      /* Help message (long) */
#endif
#ifdef CONFIG_AUTO_COMPLETE
    /* do auto completion on the arguments */
    int     (*complete)(int argc, char *argv[], char last_char, int maxv, char *cmdv[]);
#endif
};

typedef struct cmd_tbl_s  cmd_tbl_t;
```

Example

```
int do_nfs (cmd_tbl_t *cmdtp, int flag, int argc, char *argv[])
{
    return netboot_common(NFS, cmdtp, argc, argv);
}

U_BOOT_CMD(
    nfs, 3, 1, do_nfs,
    "nfs\t\t- boot image via network using NFS protocol\t\t",
    "[loadAddress] [host ip addr:bootfilename]\t\t")
);
```

U_BOOT_CMD

```
OUTPUT_FORMAT("elf32-littlearm", "elf32-  
littlearm", "elf32-littlearm")  
/*OUTPUT_FORMAT("elf32-arm", "elf32-arm",  
"elf32-arm")*/  
OUTPUT_ARCH(arm)  
ENTRY(_start)  
SECTIONS
```

```
{  
  
#define Struct_Section __attribute__((unused,section(".u_boot_cmd")))  
  
#ifndef CFG_LONGHELP  
  
#define U_BOOT_CMD(name,maxargs,rep,cmd,usage,help) ⚡  
cmd_tbl_t __u_boot_cmd_##name Struct_Section = {#name, maxargs, rep, cmd, usage, help}  
  
#else /* no long help info */  
  
#define U_BOOT_CMD(name,maxargs,rep,cmd,usage,help) ⚡  
cmd_tbl_t __u_boot_cmd_##name Struct_Section = {#name, maxargs, rep, cmd, usage}  
  
#endif /* CFG_LONGHELP */
```

_u_boot_cm
d_start

name
maxargs
repeatable
cmd
usage
help
name
maxargs
repeatable
cmd
usage
help
.....
name
maxargs
repeatable
cmd
usage
help

Uboot_cmd_
table

_u_boot_cm
d_end

Boot_os_Fcn

```
/*
 * Continue booting an OS image; caller already has:
 * - copied image header to global variable `header'
 * - checked header magic number, checksums (both header & image),
 * - verified image architecture (PPC) and type (KERNEL or MULTI),
 * - loaded (first part of) image to header load address,
 * - disabled interrupts.
 */
typedef void boot_os_Fcn (cmd_tbl_t *cmdtp, int flag,
                          int argc, char *argv[],
                          ulong addr, /* of image to boot */
                          ulong *len_ptr, /* multi-file image length table */
                          int verify); /* getenv("verify")[0] != 'n' */

#ifdef CONFIG_PPC
static boot_os_Fcn do_bootm_linux;
#else
extern boot_os_Fcn do_bootm_linux;
#endif
#ifdef CONFIG_SILENT_CONSOLE
static void fixup_silent_linux (void);
#endif
static boot_os_Fcn do_bootm_netbsd;
static boot_os_Fcn do_bootm_rtems;
#if (CONFIG_COMMANDS & CFG_CMD_ELF)
static boot_os_Fcn do_bootm_vxworks;
static boot_os_Fcn do_bootm_qnxelf;
int do_bootvx (cmd_tbl_t *cmdtp, int flag, int argc, char *argv[]);
int do_bootelf (cmd_tbl_t *cmdtp, int flag, int argc, char *argv[]);
#endif /* CFG_CMD_ELF */
#ifdef CONFIG_ARTOS
static boot_os_Fcn do_bootm_artos;
#endif
#ifdef CONFIG_LYNXKDI
static boot_os_Fcn do_bootm_lynxkdi;
extern void lynxkdi_boot (image_header_t *);
#endif
```

Lib_arm/armlinux.c

do_bootm() - 1

```
if (argc < 2) {
    addr = load_addr;
} else {
    addr = simple_strtoul(argv[1], NULL, 16);
}
```

Default Load address =0x20000000

Bootm 20008000 : loading 할 address 를 받아 command 로 받는다.

```
memcpy (&header, (char *)addr, sizeof(image_header_t));
```

zImage의 헤더를 읽어온다.

```
if (ntohl(hdr->ih_magic) != IH_MAGIC) {
    /* Check Magic Number */
}
```

```
data = (ulong)&header;
len = sizeof(image_header_t);
```

```
checksum = ntohl(hdr->ih_hcrc);
hdr->ih_hcrc = 0;
```

```
if (crc32 (0, (char *)data, len) != checksum) {
    /* Check CRC32 */
}
```

```
data = addr + sizeof(image_header_t);
len = ntohl(hdr->ih_size);
```

```
typedef struct image_header {
    uint32_t ih_magic; /* Image Header Magic Number */
    uint32_t ih_hcrc; /* Image Header CRC Checksum */
    uint32_t ih_time; /* Image Creation Timestamp */
    uint32_t ih_size; /* Image Data Size */
    uint32_t ih_load; /* Data Load Address */
    uint32_t ih_ep; /* Entry Point Address */
    uint32_t ih_dcrc; /* Image Data CRC Checksum */
    uint8_t ih_os; /* Operating System */
    uint8_t ih_arch; /* CPU architecture */
    uint8_t ih_type; /* Image Type */
    uint8_t ih_comp; /* Compression Type */
    uint8_t ih_name[IH_NMLEN]; /* Image Name */
} image_header_t;
```

Pointer를 header에서 실제 압축된 커널을 가르킨다.

do_bootm() - 2

```
switch (hdr->ih_type) {  
case IH_TYPE_STANDALONE:  
    name = "Standalone Application";  
    /* A second argument overwrites the load address */  
    if (argc > 2) {  
        hdr->ih_load = simple_strtoul(argv[2], NULL, 16);  
    }  
    break;  
case IH_TYPE_KERNEL:  
    name = "Kernel Image";  
    break;  
case IH_TYPE_MULTI:  
    name = "Multi-File Image";  
    len = ntohl(len_ptr[0]);  
    /* OS kernel is always the first image */  
    data += 8; /* kernel_len + terminator */  
    for (i=1; len_ptr[i]; ++i)  
        data += 4;  
    break;  
default: printf ("Wrong Image Type for %s command\n", cmdtp->name);  
    SHOW_BOOT_PROGRESS (-5);  
    return 1;  
} ? end switch hdr->ih_type ?
```

LINUX의 경우는 IH_TYPE_KERNEL

Kernel + ramdisk 이미지

do_bootm() - 3

```
switch (hdr->ih_comp) {  
case IH_COMP_NONE:  
    if(ntohl(hdr->ih_load) == addr) {  
        printf (" XIP %s ... ", name);  
    } else {  
        memmove ((void *) ntohl(hdr->ih_load), (uchar *)data, len);  
    }  
    break;
```

```
case IH_COMP_GZIP:  
    printf (" Uncompressing %s ... ", name);  
    if (gunzip ((void *)ntohl(hdr->ih_load), unc_len,  
              (uchar *)data, &len) != 0) {  
        puts ("GUNZIP ERROR - must RESET board to recover");  
        SHOW_BOOT_PROGRESS (-6);  
        do_reset (cmdtp, flag, argc, argv);  
    }  
    break;
```

압축된 커널을 압축 해제한다.

압축이 풀리는 위치는 (void *)ntohl(hdr_ih_load)이다

```
#ifndef CONFIG_BZIP2
```

```
case IH_COMP_BZIP2:  
    printf (" Uncompressing %s ... ", name);  
    /*  
    * If we've got less than 4 MB of malloc() space,  
    * use slower decompression algorithm which requires  
    * at most 2300 KB of memory.  
    */  
    i = BZ2_bzBuffToBuffDecompress ((char*)ntohl(hdr->ih_load),  
                                   &unc_len, (char *)data, len,  
                                   CFG_MALLOC_LEN < (4096 * 1024), 0);  
    if (i != BZ_OK) {  
        printf ("BUNZIP2 ERROR %d - must RESET board to recover", i);  
        SHOW_BOOT_PROGRESS (-6);  
        udelay(100000);  
        do_reset (cmdtp, flag, argc, argv);  
    }  
    break;
```

```
#endif /* CONFIG_BZIP2 */
```

```
default:  
    if (iflag)  
        enable_interrupts();  
    printf ("Unimplemented compression type %d", hdr->ih_comp);  
    SHOW_BOOT_PROGRESS (-7);  
    return 1;
```

```
} ? end switch hdr->ih_comp ?
```


do_bootm() - 4

```
switch (hdr->ih_os) {
default: /* handled by (original) Linux case */
case IH_OS_LINUX:
#ifdef CONFIG_SILENT_CONSOLE
    fixup_silent_linux();
#endif
    do_bootm_linux (cmdtp, flag, argc, argv,
                    addr, len_ptr, verify);
    break;
case IH_OS_NETBSD:
    do_bootm_netbsd (cmdtp, flag, argc, argv,
                    addr, len_ptr, verify);
    break;

#ifdef CONFIG_LYNXKDI
case IH_OS_LYNXOS:
    do_bootm_lynxkdi (cmdtp, flag, argc, argv,
                    addr, len_ptr, verify);
    break;
#endif

case IH_OS_RTEMS:
    do_bootm_rtems (cmdtp, flag, argc, argv,
                    addr, len_ptr, verify);
    break;

#if (CONFIG_COMMANDS & CFG_CMD_ELF)
case IH_OS_VXWORKS:
    do_bootm_vxworks (cmdtp, flag, argc, argv,
                    addr, len_ptr, verify);
    break;
case IH_OS_QNX:
    do_bootm_qnxelf (cmdtp, flag, argc, argv,
                    addr, len_ptr, verify);
    break;
#endif /* CFG_CMD_ELF */
#ifdef CONFIG_ARTOS
case IH_OS_ARTOS:
    do_bootm_artos (cmdtp, flag, argc, argv,
                    addr, len_ptr, verify);
    break;
#endif
} ? end switch hdr->ih_os ?
```

드디어 Linux kernel 로 진입한
다...!!!

do_bootm_linux()

```
void do_bootm_linux (cmd_tbl_t *cmdtp, int flag, int argc, char *argv[],
                    bootm_headers_t *images)
{
    ulong    initrd_start, initrd_end;
    ulong    ep = 0;
    bd_t     *bd = gd->bd;
    char     *s;
    int      machid = bd->bi_arch_number;
    void     (*theKernel)(int zero, int arch, uint params);
    int      ret;

#ifdef CONFIG_CMDLINE_TAG
    char *commandline = getenv ("bootargs");
#endif

```

Bootargs 를 가지고 온다

```
theKernel = (void (*)(int, int, uint))ep;
s = getenv ("machid");
if (s) {
    machid = simple_strtoul (s, NULL, 16);
    printf ("Using machid 0x%x from environment\n", machid);
}

ret = boot_get_ramdisk (argc, argv, images, IH_ARCH_ARM,
                        &initrd_start, &initrd_end);
if (ret)
    goto error;

```

Kernel에게 이미지 entry poiboot_get_ramdisk

Machin id를 가지고 온다

Ram disk가 있는지 없는지 확인

do_bootm_linux()

```
cleanup_before_linux ();
```

```
theKernel (0, machid, bd->bi_boot_params);  
/* does not return */  
return;
```

Kernel image로 제어권을 넘겨준다.

```
int cleanup_before_linux (void)  
{  
    /*  
     * this function is called just before we call linux  
     * it prepares the processor for linux  
     *  
     * we turn off caches etc ...  
     */  
  
    unsigned long i;  
  
    disable_interrupts ();  
  
    /* turn off I/D-cache */  
    asm ("mrc p15, 0, %0, c1, c0, 0" : "=r" (i));  
    i &= ~(C1_DC | C1_IC);  
    asm ("mcr p15, 0, %0, c1, c0, 0" : "=r" (i));  
  
    /* invalidate l-cache */  
    arm_cache_flush();  
  
    i = 0;  
    /* mem barrier to sync up things */  
    asm ("mcr p15, 0, %0, c7, c10, 4" : "=r" (i));  
  
    return(0);  
}
```

커널 빌드 및 간단 드라이버

안드로이드 커널 Open Git 서버

- <http://android.git.kernel.org>

ANDROID

open source project

To clone one of these trees, install [git](#), and run:

```
git clone git://android.git.kernel.org/ + project path.
```

To clone the entire platform, install [repo](#), and run:

```
mkdir mydroid  
cd mydroid  
repo init -u git://android.git.kernel.org/platform/manifest.git  
repo sync
```

안드로이드 커널 다운받기(git)

- #git clone git://android.git.kernel.org/kernel/common.git
- # git checkout --track -b android-2.6.29 origin/android-2.6.29
- #git branch
- 현재까지 2.6.32 버전까지 릴리즈 됨

Linux 커널 공식 사이트

- <http://www.kernel.org/>

The Linux Kernel Archives

Welcome to the Linux Kernel Archives. This is the primary site for the Linux kernel source, but it has much more than just Linux kernel [Frequently Asked Questions](#)

Protocol	Location
HTTP	http://www.kernel.org/pub/
FTP	ftp://ftp.kernel.org/pub/
RSYNC	rsync://rsync.kernel.org/pub/

Latest Stable Kernel:



[2.6.34.1](#)

linux-next:	next-20100706	2010-07-06	[Patch]	[View Patch]	[Gitweb]			
mainline:	2.6.35-rc4	2010-07-05	[Full Source]	[Patch]	[View Patch]	[View Inc.]	[Gitweb]	[Changelog]
snapshot:	2.6.35-rc3-git8	2010-07-03	[Patch]	[View Patch]				
stable:	2.6.34.1	2010-07-05	[Full Source]	[Patch]	[View Patch]	[Gitweb]	[Changelog]	
stable:	2.6.33.6	2010-07-05	[Full Source]	[Patch]	[View Patch]	[View Inc.]	[Gitweb]	[Changelog]
stable:	2.6.32.16	2010-07-05	[Full Source]	[Patch]	[View Patch]	[View Inc.]	[Gitweb]	[Changelog]
stable:	2.6.31.14	2010-07-05	[Full Source]	[Patch]	[View Patch]	[View Inc.]	[Gitweb]	[Changelog]
stable:	2.6.27.48	2010-07-05	[Full Source]	[Patch]	[View Patch]	[View Inc.]	[Gitweb]	[Changelog]
stable:	2.4.37.9	2010-02-01	[Full Source]	[Patch]	[View Patch]	[Gitweb]	[Changelog]	

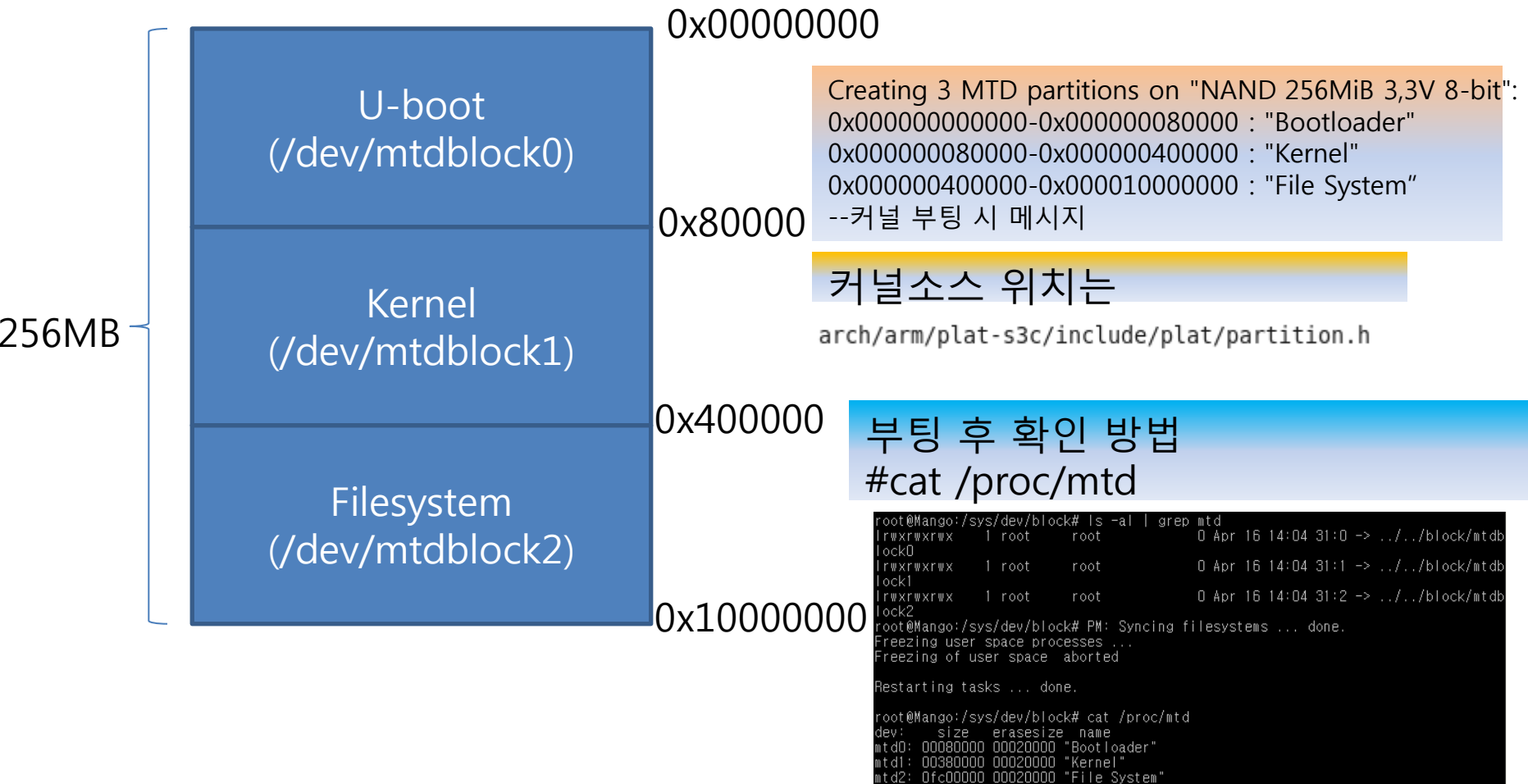
삼성 커널 다운받기

- #git clone
git://git.kernel.org/pub/scm/linux/kernel/
git/kki_ap/linux-2.6-samsung.git
- #cd linux-2.6-samsung
- #git fetch
- #git fetch --tag
- #git checkout 2.6.29-samsung

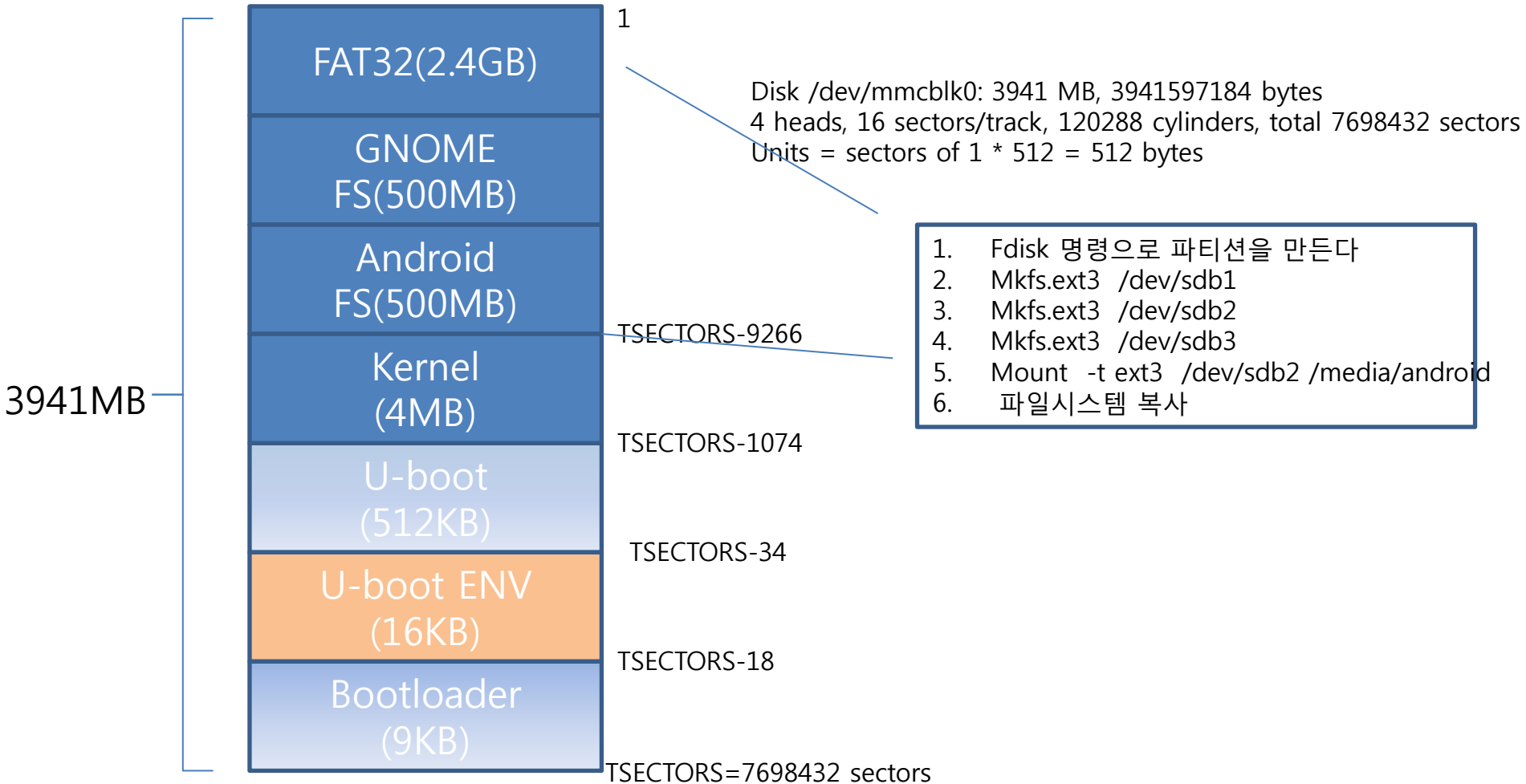
망고100 커널 소스 받기

- #wget
http://crztech.iptime.org:8080/Release/mango100/kernel/mango100_kernel_2010_06_30.tgz
- #tar xvf
mango100_kernel_2010_06_30.tgz

망고100 NAND Partition 영역



SD /MMC Partition 구성



커널 build 환경 분석

```
VERSION = 2  
PATCHLEVEL = 6  
SUBLEVEL = 29  
EXTRAVERSION =  
NAME = Temporary Tasmanian Devil
```

>(Top Dir)/Makefile 첫번째 라인에 위치
>커널 2.6.29 버전 사용

```
ARCH           ?= arm  
CROSS_COMPILE := $(shell if [ -f .cross_compile ]; then \  
                cat .cross_compile: \  
                fi)  
# Architecture as present in compile.b
```

>ARCH?=arm 의미는 ARCH 의 값으로 arm있느냐 묻고, 없으면, arm 문자를 대입
>.cross_compile 이 있으면, .cross_compile 내용을 CROSS_COMPILE 로 사용

커널 build 환경 분석(Config)

- #make mango100_android_defconfig 실행 시

```
%config: scripts_basic outputmakefile FORCE
$(Q)mkdir -p include/linux include/config
$(Q)$(MAKE) $(build)=scripts/kconfig $@
```

Scripts/kconfig/Makefile에서 아래 코드 수행

```
_%defconfig: $(obj)/conf
```

```
$(Q)$< -D arch/$(SRCARCH)/configs/$@ $(Kconfig)
```

→make -D arch/arm/configs/mango100_android_defconfig



.config 파일생성

```
[icanjji@localhost mango100_kernel_2010_06_30]$ ls -al arch/arm/configs/mango100*
-rw-rw-r-- 1 icanjji icanjji 45298 2010-06-30 02:51 arch/arm/configs/mango100_android_defconfig
-rw-rw-r-- 1 icanjji icanjji 46308 2010-06-30 02:51 arch/arm/configs/mango100_defconfig
-rw-rw-r-- 1 icanjji icanjji 50889 2010-06-30 02:51 arch/arm/configs/mango100_wifi_android_defconfig
```

커널 build 환경 분석(Conf)

- #make menu_config 명령 실행 시

```
%config: scripts_basic outputmakefile FORCE
$(Q)mkdir -p include/linux include/config
$(Q)$(MAKE) $(build)=scripts/kconfig $@
```

#make scripts/kconfig menuconfig 이 실행 됨

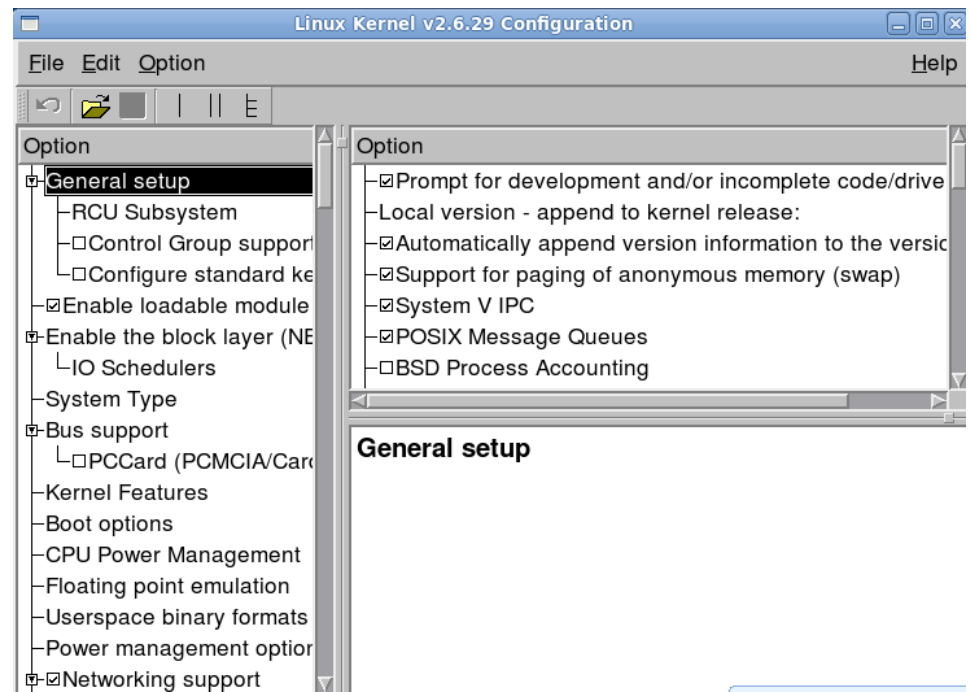
menuconfig: \$(obj)/mconf
\$< \$(Kconfig)

```
.config - Linux Kernel v2.6.29 Configuration
Linux Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---. Highlighted letters are
hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded x
<M> module < > module capable
General setup ---
[*] Enable loadable module support ---
-- Enable the block layer ---
System Type ---
Bus support ---
Kernel Features ---
Boot options ---
CPU Power Management ---
Floating point emulation ---
Userspace binary formats ---
Power management options ---
[*] Networking support ---
Device Drivers ---
File systems ---
Kernel hacking ---
Security options ---
-- Cryptographic API ---
Library routines ---
<Select> < Exit > < Help >
```

.config 저장

커널 build 환경 분석(Conf)

- #make xconfig (QT3 Package 필요)
- #yum install qt* 명령으로 설치
- "cannot find -lXi 에러 발생 시 #yum install libXi 실행



커널 build 실행 분석

- \$(TOP)/Makefile – 최상위 Makefile
 - `vmlinux`와 `modules` 생성
- `.config` – 커널 설정 파일
 - `make [config | menuconfig | xconfig]` 를 통해 생성.
- `arch/$(ARCH)/Makefile` - 아키텍처별 makefile
- `scripts/Makefile.*` - 모든 kbuild Makefile에 사용되는 규칙이 들어있는 파일
- kbuild Makefiles – 약 500개 정도가 있다.

커널 build 실행 분석

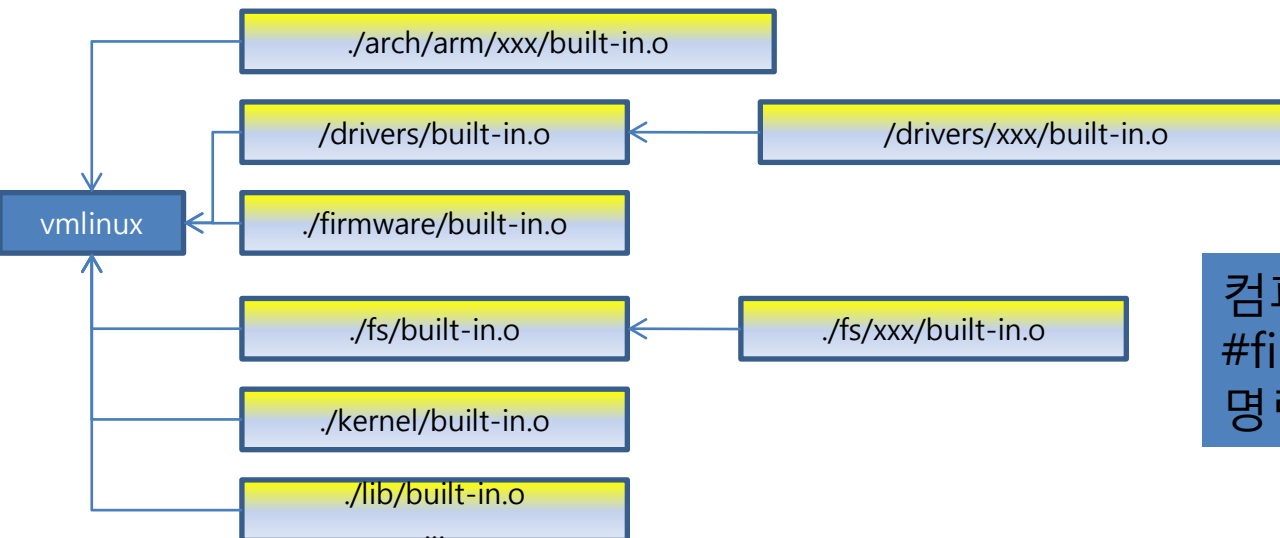
1. 커널 설정 (make config|menuconfig|xconfig)
.config를 만듦
2. 커널 버전을 include/linux/version.h 에 저장
3. include/asm-\$(ARCH)에 대한 심볼릭 링크 만듦
4. arch/\$(ARCH)/Makefile 에서 정의된, 그외의 타겟 빌딩을 위한 모든 종속 리스트를 준비
5. init-*, core-*, driver-*, net-* 등의 타겟 등을 만듦
6. 모든 오브젝트들이 링크되고, 소스 트리의 루트 디렉토리에 vmlinux를 만듦.
7. 최종 부트이미지(zImage)를 만들기 위한 아키텍처에 따른 부분이 실행됨

Built-in object goals (obj-y)

- specifying object files for vmlinux
- “\$(LD) -r” : to merge \$(obj-y) files into one `built-in.o` file

```
5 obj-y = sched.o fork.o exec_domain.o panic.o printk.o profile.o \
6       exit.o itimer.o time.o softirq.o resource.o \
7       sysctl.o capability.o ptrace.o timer.o user.o \
8       signal.o sys.o kmod.o workqueue.o pid.o \
9       rcupdate.o extable.o params.o posix-timers.o \
10      kthread.o wait.o kfifo.o sys_ni.o posix-cpu-timers.o mutex.o \
11      hrtimer.o rwsem.o latency.o nsproxy.o srcu.o
```

ex. \$(TOP)/kernel/Makefile



컴파일 후
#find . -name built-in.o
명령으로 확인

Built-in object goals (obj-y)

echo "(patsubst pattern,replacement,text)"

```
vmlinux-dirs := $(patsubst %/,%,$(filter %/, $(init-y) $(init-m) \W  
$(core-y) $(core-m) $(drivers-y) $(drivers-m) \W  
$(net-y) $(net-m) $(libs-y) $(libs-m)))
```

```
vmlinux-alldirs := $(sort $(vmlinux-dirs) $(patsubst %/,%,$(filter %/, \W  
$(init-n) $(init-) \W  
$(core-n) $(core-) $(drivers-n) $(drivers-) \W  
$(net-n) $(net-) $(libs-n) $(libs-))))
```

```
init-y := $(patsubst %/, %/built-in.o, $(init-y))  
core-y := $(patsubst %/, %/built-in.o, $(core-y))  
drivers-y := $(patsubst %/, %/built-in.o, $(drivers-y))  
net-y := $(patsubst %/, %/built-in.o, $(net-y))  
libs-y1 := $(patsubst %/, %/lib.a, $(libs-y))  
libs-y2 := $(patsubst %/, %/built-in.o, $(libs-y))  
libs-y := $(libs-y1) $(libs-y2)
```

Loadable module goals – (obj-m)

- object files which are built as loadable kernel modules.

```
# This goes last, so that "real" scsi devices probe early
obj-$(CONFIG_SCSI_DEBUG) += scsi_debug.o

obj-$(CONFIG_SCSI_WAIT_SCAN) += scsi_wait_scan.o

scsi_mod-y += scsi.o hosts.o scsi_cmnd.o scsi_error.o
            scsicam.o scsi_err.o

scsi_mod-$(CONFIG_SCSI_DMA) += scsi_lib_dma.o
scsi_mod-y += scsi_scan.o scsi_target.o
scsi_mod-$(CONFIG_SCSI_NETLINK) += scsi_netlink.o
scsi_mod-$(CONFIG_SCSI_SYSCtl) += scsi_sysctl.o
scsi_mod-$(CONFIG_SCSI_PROC_FS) += scsi_proc.o
```

ex. \$(TOP)/driver/scsi/Makefile

```
# CONFIG_CHR_DEV_ST is not set
# CONFIG_CHR_DEV_OSST is not set
# CONFIG_BLK_DEV_SR is not set
# CONFIG_CHR_DEV_SG is not set
# CONFIG_CHR_DEV_SCH is not set
#
# Some SCSI devices (e.g. CD jukebox) support multiple LUNs
# CONFIG_SCSI_MULTI_LUN is not set
# CONFIG_SCSI_CONSTANTS is not set
# CONFIG_SCSI_LOGGING is not set
# CONFIG_SCSI_SCAN_ASYNC is not set
CONFIG_SCSI_WAIT_SCAN=m
#
# SCSI Transports
#
# CONFIG_SCSI_SPI_ATTRS is not set
```

ex. \$(TOP)/.config

```
@$(kecho) ' Building modules, stage 2.';
$(Q)$MAKE -f $(srctree)/scripts/Makefile.modpost
$(Q)$MAKE -f $(srctree)/scripts/Makefile.fwinst obj=firmware __fw_modbuild
```

- 참고!! 커널에 포함되지 않은 external module을 컴파일 하기 위해서는 Documentation/kbuild/modules.txt를 참조
 - 디바이스 드라이버 개발

Environment Variables

variable	value	Description
V	0	빌드시에, 현재 컴파일되는 파일명만을 보여줌. (default)
V	1	빌드시에 실행되는 모든 명령 및 메시지를 보여 줌.
O	dir	컴파일 되는 모든 output file들을 dir에 저장되게 지정
C	1	빌드과정에서 sparse tool이 컴파일된 파일을 체 크하게끔 한다. sparse은 커널 소스 파일의 프로 그래밍 에러를 찾는 툴이다.
C	2	sparse tool은 컴파일에 관계없이 모든 파일을 체크하게끔 한다.

Example :# make V=1 ARCH=arm

- 커널 빌드시, 명령 및 메시지 출력 옵션

```
38 ifdef V
39     ifeq ("$(origin V)", "command line")
40         KBUILD_VERBOSE = $(V)
41     endif
42 endif
43 ifndef KBUILD_VERBOSE
44     KBUILD_VERBOSE = 0
45 endif
```

```
288 ifeq ($(KBUILD_VERBOSE),1)
289     quiet =
290     Q =
291 else
292     quiet=quiet_
293     Q = @
294 endif
```

- 커널 빌드시, 소스 코드 체크 옵션

```
57 ifdef C
58     ifeq ("$(origin C)", "command line")
59         KBUILD_CHECKSRC = $(C)
60     endif
61 endif
62 ifndef KBUILD_CHECKSRC
63     KBUILD_CHECKSRC = 0
64 endif
```

- 빌드된 파일의 출력 디렉토리 지정

```
101 ifdef O
102     ifeq ("$(origin O)", "command line")
103         KBUILD_OUTPUT := $(O)
104     endif
105 endif
```

\$(top)/Makefile

How to build vmlinux ??

-Makefile : 아키텍처 독립적인 부분

```
837 # vmlinux image - including updated kernel symbols
838 vmlinux: $(vmlinux-lds) $(vmlinux-init) $(vmlinux-main) vmlinux.o $(kallsyms.o) FORCE
839 ifdef CONFIG_HEADERS_CHECK
840     $(Q)$(MAKE) -f $(srctree)/Makefile headers_check
841 endif
842 ifdef CONFIG_SAMPLES
843     $(Q)$(MAKE) $(build)=samples
844 endif
845 ifdef CONFIG_BUILD_DOCSRC
846     $(Q)$(MAKE) $(build)=Documentation
847 endif
848     $(call vmlinux-modpost)
849     $(call if_changed_rule,vmlinux__)
850     $(Q)rm -f .old_version
```

```
688 vmlinux-init := $(head-y) $(init-y)
689 vmlinux-main := $(core-y) $(libs-y) $(drivers-y) $(net-y)
690 vmlinux-all := $(vmlinux-init) $(vmlinux-main)
691 vmlinux-lds := arch/$(SRCARCH)/kernel/vmlinux.lds
```

```
# Objects we will link into vmlinux / subdirs
init-y      := init/
drivers-y   := drivers/ sound/ firmware/
net-y      := net/
libs-y     := lib/
core-y     := usr/
```

```
core-y      += kernel/ mm/ fs/ ipc/ security/ crypto/ block/
```

- arch/arm/Makefile : 아키텍처 종속적인 부분

```
190 core-y      += arch/arm/kernel/ arch/arm/mm/ arch/arm/common/
191 core-y      += $(machdirs) $(platdirs)
192 core-$(CONFIG_FPE_NWFPE) += arch/arm/nwfpe/
193 core-$(CONFIG_FPE_FASTFPE) += $(FASTFPE_OBJ)
194 core-$(CONFIG_VFP) += arch/arm/vfp/
195
196 drivers-$(CONFIG_OPROFILE) += arch/arm/oprofile/
197
198 libs-y      := arch/arm/lib/ $(libs-y)
```

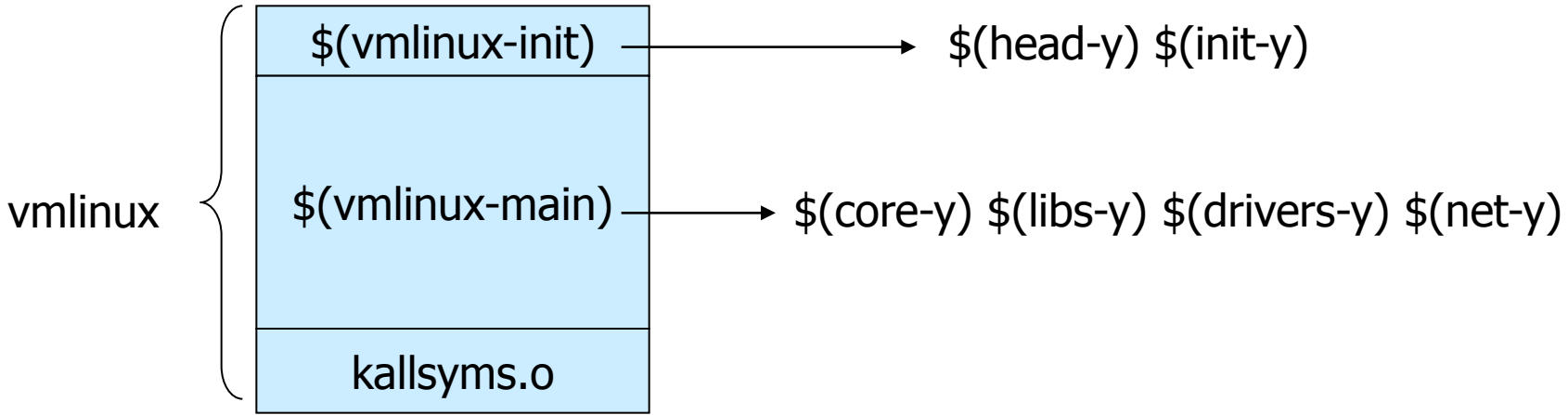

Building vmlinux

- vmlinux는 \$(vmlinux-init)와 \$(vmlinux-main)에서 정의된 오브젝트로 만들어진다.
- 각 오브젝트의 linking 순서가 중요.

링커스크립트 지정

```
arm-linux-ld -EL -p --no-undefined -X -o vmlinux -T arch/arm/kernel/vmlinux.lds arch/arm/kernel/head.o arch/arm/kernel/init_task.o init/built-in.o --start-group usr/built-in.o arch/arm/kernel/built-in.o arch/arm/mm/built-in.o arch/arm/common/built-in.o arch/arm/mach-s5pc100/built-in.o arch/arm/plat-s5pc1xx/built-in.o arch/arm/plat-s3c/built-in.o arch/arm/nwfpe/built-in.o arch/arm/vfp/built-in.o kernel/built-in.o mm/built-in.o fs/built-in.o ipc/built-in.o security/built-in.o crypto/built-in.o block/built-in.o arch/arm/lib/lib.a lib/lib.a arch/arm/lib/built-in.o lib/built-in.o drivers/built-in.o sound/built-in.o firmware/built-in.o net/built-in.o --end-group .tmp_kallsyms2.o
```

ld [옵션] 오브젝트파일 ..
-m : 어떤 포맷으로 출력물을 만들 것인가??
-T: 링커 스크립트 지정
--start-group ~ --end-group : ~에 지정된 오브젝트들 서로간에 변수나 함수 참조를 가능하게 함.
-o : 출력 파일명 지정



Building zImage (1/3)

1. arch/arm/boot/compressed/Image

\$(topdir)/vmlinux에서 .note 섹션, .comment 섹션 및 모든 심볼들과 재배치 정보들을 제거한 후, 인스 트럭션 데이터만을 뽑아, arch/arm/boot/compressed/Image 라는 바이너리 파일을 만든다.

```
arm-linux-objcopy -O binary -R .note -R .note.gnu.build-id -R .comment -S vmlinux  
arch/arm/boot/Image
```

objcopy [옵션] 입력파일 [출력파일]

-O 오브젝트형식: 어떤 오브젝트 형식으로 출력 파일을 만들 것인지 지정 (예: elf32-i386, binary)

-R 섹션: 출력 파일에서 해당 섹션을 지운다.

-S: 입력 파일의 재배치 정보와 심볼 정보를 출력 파일에 복사하지 않는다.

2. arch/arm/boot/compressed/piggy.gz

1단계에서 만든, Image을 가장 압축률이 좋은 방법으로 압축해서(-9), piggy.gz을 만듦

```
gzip -f -9 < arch/arm/boot/compressed/../Image  
> arch/arm/boot/compressed/piggy.gz
```

Building zImage (2/3)

3. arch/arm/boot/compressed/piggy.o

```
arm-linux-gcc -Wp,-MD,arch/arm/boot/compressed/.piggy.o.d -nostdinc -isystem
/usr/local/arm/4.2.2-eabi/usr/bin-ccache/./lib/gcc/arm-unknown-linux-gnueabi/4.2.2/include
-Iinclude -I/home/icanjji/work/mango100/mango100_kernel_2010_06_30/arch/arm/include
-include include/linux/autoconf.h -D__KERNEL__ -mlittle-endian
-Iarch/arm/mach-s5pc100/include -Iarch/arm/plat-s5pc1xx/include
-Iarch/arm/plat-s3c/include -D__ASSEMBLY__ -mabi=aapcs-linux -mno-thumb-interwork
-D__LINUX_ARM_ARCH__=7 -march=armv5t -Wa,-march=armv7-a -msoft-float -gdwarf-2
-Wa,-march=all -c -o arch/arm/boot/compressed/piggy.o arch/arm/boot/compressed/piggy.S
```

4. arch/arm/boot/compressed/vmlinux

head.o + misc.o + piggy.o 를 링킹해서, vmlinux를 만들, 이 때 .text 섹션은 0x20008000 위치부터, 엔트리 포인트는 startup 32로 지정한다.

```
arm-linux-ld -EL --defsym zreladdr=0x20008000 --defsym params_phys=0x20000100 -p --no-undefined -X /usr/local/arm/4.2.2-eabi/usr/bin-ccache/./lib/gcc/arm-unknown-linux-gnueabi/4.2.2/libgcc.a -T arch/arm/boot/compressed/vmlinux.lds arch/arm/boot/compressed/head.o arch/arm/boot/compressed/piggy.o arch/arm/boot/compressed/misc.o -o arch/arm/boot/compressed/vmlinux
```

ld [옵션] 오브젝트파일 ..

- -m *emulation* : 링커에게 해당 타겟 emulation에 맞는 정보를 제공 (예. 링커 스크립트 등)
- -r : 재할당 가능한 출력 파일을 생성. 즉, ld의 입력 오브젝트로 쓰일 수 있는 출력 파일을 생성. (실제로 piggy.o는 ld로 다시 링킹됨)
- --format *input-format* : 입력 오브젝트 파일의 형식 지정
- --oformat *output-format* : 출력 오브젝트 파일의 형식 지정.
- -o : 출력 파일명 지정
- -Ttext *org* : text 섹션의 시작주소를 org로 지정
- -e *entry* : 엔트리 포인트를 지정한다.

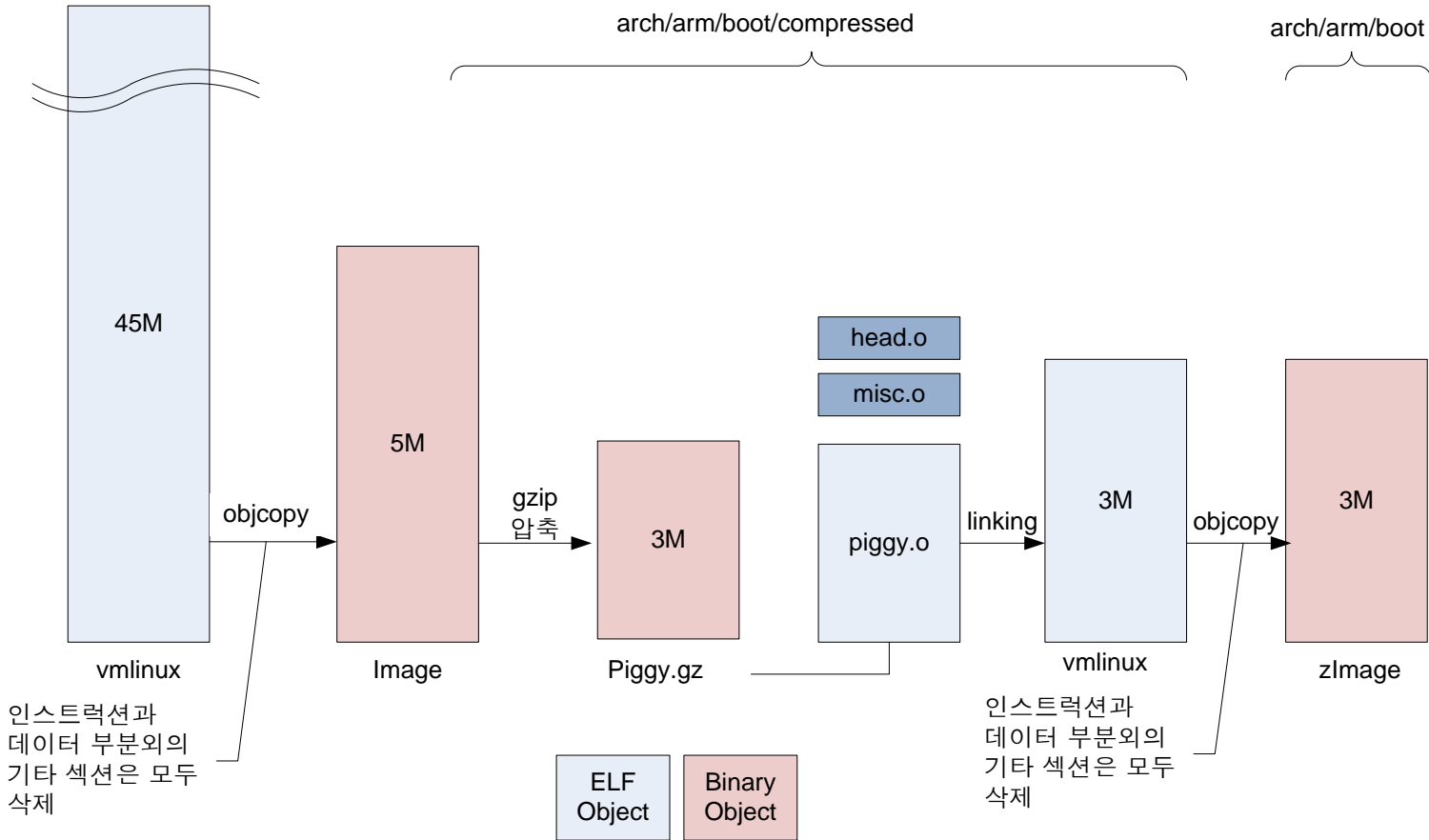
Building zImage (3/3)

5. arch/arm/boot/zImage

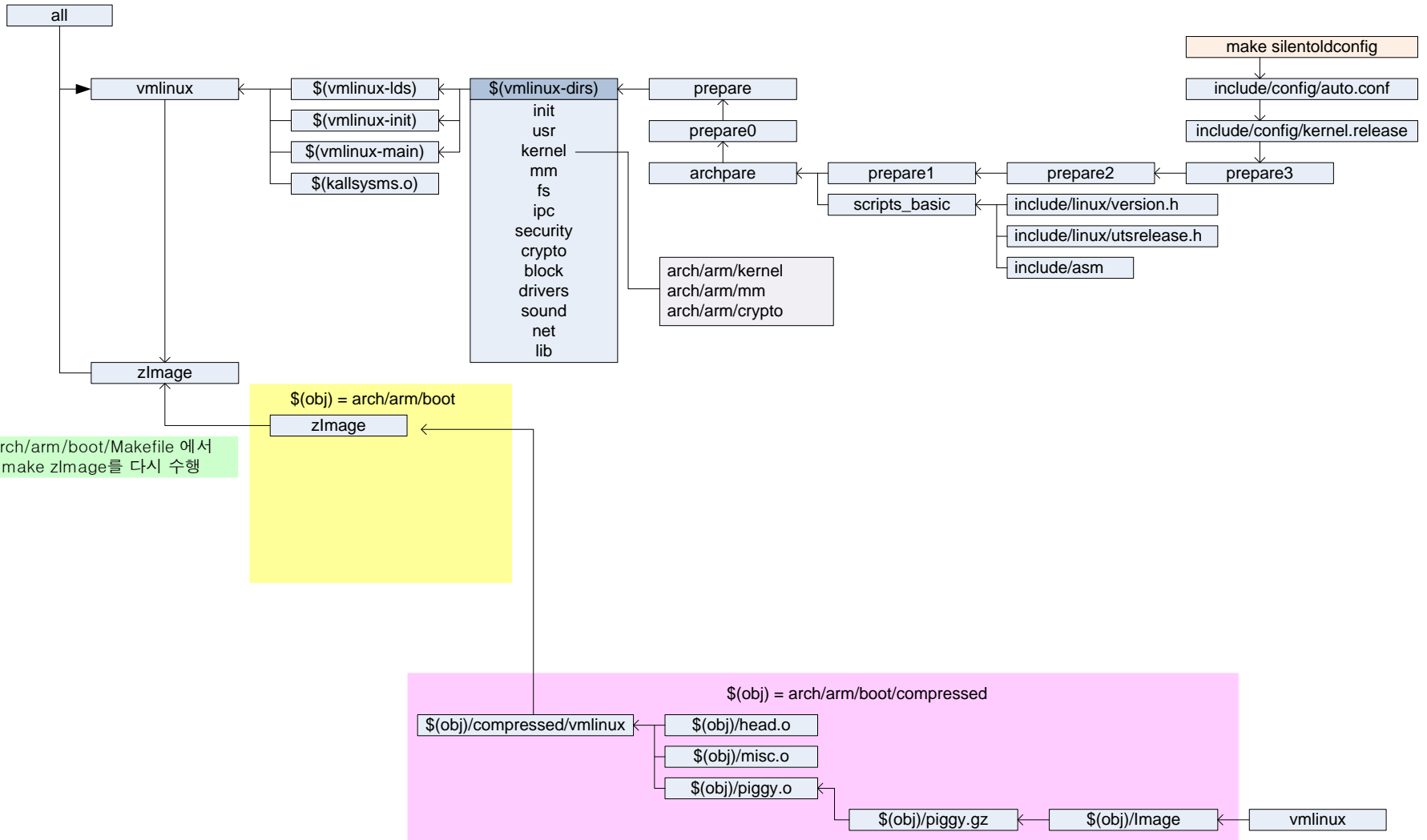
4단계에서 만든 vmlinux에서 .note 섹션, .comment 섹션 및 모든 심볼들과 재배치 정보들을 제거한 후, 인스트럭션 데이터만을 뽑아, arch/arm/boot/zImage 라는 바이너리 파일을 만든다.

```
arm-linux-objcopy -O binary -R .note -R .note.gnu.build-id -R .comment -S  
arch/arm/boot/compressed/vmlinux arch/arm/boot/zImage
```

결론 - Kernel Build Process



참고 : kernel Makefile 계층도



arch/arm/boot/Makefile 에서 make zImage를 다시 수행

여기에서 vmlinux는 커널 소스 최상위 디렉토리에서 만들어진 vmlinux를 말한다.

vmlinux.lds 첫 부분.

```
OUTPUT_ARCH(arm)
ENTRY(_start)
SECTIONS
{
    . = 0;
    _text = .;

    .text : {
        _start = .;
        *(.start)
        *(.text)
        *(.text.*)
        *(.fixup)
        *(.gnu.warning)
        *(.rodata)
        *(.rodata.*)
        *(.glue_7)
        *(.glue_7t)
        *(.piggydata)
        . = ALIGN(4);
    }
}
```

Start entry 포인
트는 _start

(예제) readelf -l arch/arm/boot/compressed/vmlinux

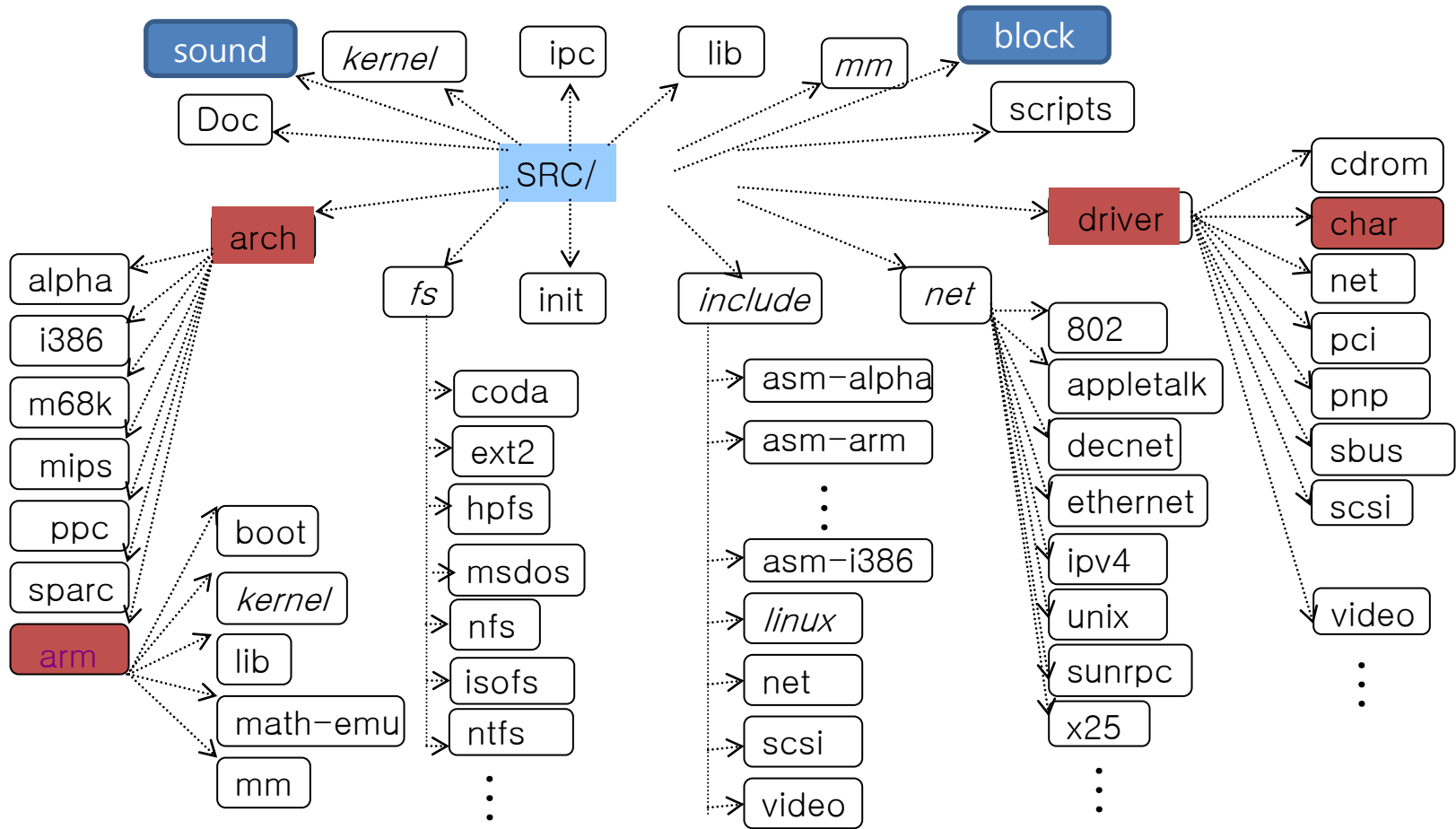
```
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00
  Class:                   ELF32
  Data:                     2's complement, little endian
  Version:                  1 (current)
  OS/ABI:                   UNIX - System V
  ABI Version:              0
  Type:                     EXEC (Executable file)
  Machine:                  ARM
  Version:                  0x1
  Entry point address:      0x0
  Start of program headers: 52 (bytes into file)
  Start of section headers: 2408208 (bytes into file)
  Flags:                    0x4000000, Version4 EABI
  Size of this header:      52 (bytes)
  Size of program headers:  32 (bytes)
  Number of program headers: 1
  Size of section headers:  40 (bytes)
  Number of section headers: 20
  Section header string table index: 17

Section Headers:
 [Nr] Name                Type          Addr          Off          Size   ES Flg Lk  Inf Al
 [ 0] NULL                  NULL          00000000     000000     000000  00  00  0  0  0
 [ 1] .text                 PROGBITS     00000000     008000     23e254  00  AX  0  0 32
 [ 2] .got                 PROGBITS     0023e254     246254     000070  00  WA  0  0  4
 [ 3] .got.plt             PROGBITS     0023e2c4     2462c4     00000c  04  WA  0  0  4
 [ 4] .bss                 NOBITS      0023e2d0     2462d0     008448  00  WA  0  0  4
 [ 5] .stack               PROGBITS     00246718     2462d0     001000  00   W  0  0  1
 [ 6] .comment             PROGBITS     00000000     2472d0     000012  00   0  0  1
 [ 7] .ARM.attributes     ARM_ATTRIBUTES 00000000     2472e2     00001b  00   0  0  1
```


참고 : 커널 빌드시, Log 남기기

- `make V=1 ARCH=arm 2>&1 | tee log-kernel.txt`

Linux 커널 소스 트리 구조



각 디렉토리 설명

- arch/
 - CPU 종속적인 부분, 각 처리기 마다 하위 디렉토리로 구성됨 (arch/arm, arch/i386, arch/alpha ...)
 - arch/arm/boot/
 - 부트스트래핑 코드
 - arch/arm/kernel/
 - 하드웨어 종속적인(hardware dependent) 커널 관리 루틴
 - 트랩, 인터럽트 처리 루틴
 - 문맥 교환 루틴
 - 장치 구성, 초기화 루틴
 - arch/arm/mm/
 - 하드웨어 종속적인 메모리 관리 루틴
- init/
 - 하드웨어 독립적인 커널 초기화 루틴 (start_kernel)
 - 태스크 0 (init_task or task[0]) 생성
 - 태스크 1, 2, 3 등 데몬 프로세스 생성

각 디렉토리 설명 (2)

- kernel/
 - 리눅스 커널의 가장 중심적인 디렉토리 (central section of the kernel)
 - 하드웨어 독립적인 커널 관리 루틴 (하드웨어 종속적인 커널 관리 루틴은 arch/arm/kernel 디렉토리에 존재)
 - fork, exit 등 태스크 관련 시스템 호출 처리 루틴
 - 스케줄러 (scheduler) 루틴
 - 시그널 처리 (signal handling)/시간 관리 (time management) 루틴
- mm/
 - 하드웨어 독립적인 메모리 관리 루틴 (하드웨어 종속적인 메모리 관리 루틴은 arch/arm/mm 디렉토리에 존재)
 - 가상 메모리 관리, 페이징 (paging), 스와핑 (swapping)
- fs/
 - 가상 파일 시스템 (virtual file system) 관리 루틴
 - open, read 등 태스크 관련 시스템 호출 처리 루틴
 - 특정 파일 시스템 관리 루틴은 하위 디렉토리에 존재 (ext2, ext3, ramfs, minix, jffs2, proc, nfs, msdos, coda, ..)

각 디렉토리 설명 (3)

- drivers/
 - 개개의 장치를 제어하기 위한 장치 드라이버 루틴
 - 디바이스 드라이버는 크게 문자/블록/네트워크 드라이버로 구분
 - drivers/block/ : 블록 장치 드라이버. 예를 들어 IDE 디스크 (hd)
 - drivers/char/ : 문자 장치 드라이버. serial ports, tty, modem, ..)
 - drivers/net : 네트워크 장치 드라이버. 예를 들어 3C509, ...
 - drivers/pci/ : PCI bus 제어
 - drivers/cdrom/ : CD-ROM 드라이버
 - drivers/scsi/ : SCSI 인터페이스 관리
- Sound :sound card 드라이버
- ipc/
 - 프로세스간 통신을 지원하기 위한 루틴
 - 세마포어(semaphores), 공유 메모리(shared memory), 메시지 큐(message queues)

각 디렉토리 설명 (4)

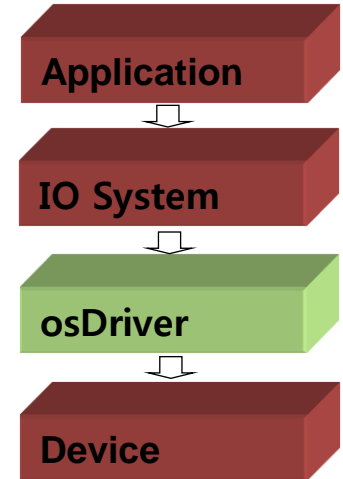
- net/
 - TCP/IP, ARP 등 네트워크 통신 프로토콜 (네트워크 장치 드라이버는 drivers/net에 존재)
 - 소켓 인터페이스
- include/
 - 커널 헤더
 - 하드웨어 독립적인 헤더 : include/linux/
 - 하드웨어 종속적인 헤더 : include/asm-*/ (예를 들어 ARM CPU 종속적인 헤더는 include/asm-arm/ 디렉토리에 존재)
- lib/
 - 커널 라이브러리 루틴
- doc/ or Documentation/
 - 커널 문서 디렉토리.

디바이스 드라이버 개요

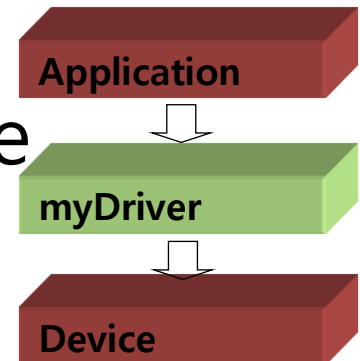
- 디바이스(Device)
 - 네트워크 어댑터, LCD 디스플레이, PCMCIA, Audio, 터미널, 키보드, 하드디스크, 플로피디스크, 프린터 등과 같은 주변 장치들을 말함
 - 디바이스의 구동에 필요한 프로그램, 즉 디바이스 드라이버가 필수적으로 요구됨
- Device Driver
 - 실제 장치 부분을 추상화 시켜 사용자 프로그램이 정형화된 인터페이스를 통해 디바이스를 접근할 수 있도록 해주는 프로그램
 - 디바이스 관리에 필요한 정형화된 인터페이스 구현에 요구되는 함수와 자료구조의 집합체
 - 표준적으로 동일 서비스 제공을 목적으로 커널의 일부분으로 내장
 - 응용프로그램이 H/W를 제어할 수 있도록 인터페이스 제공
 - 하드웨어 독립적인 프로그램을 작성을 가능하게 함

디바이스 드라이버 형태

- Device Driver Interface
 - Standard Device Driver Interface
 - UNIX compatible I/O system interface
: `open()`, `close()`, `read()`, `write()`, `ioctl()`

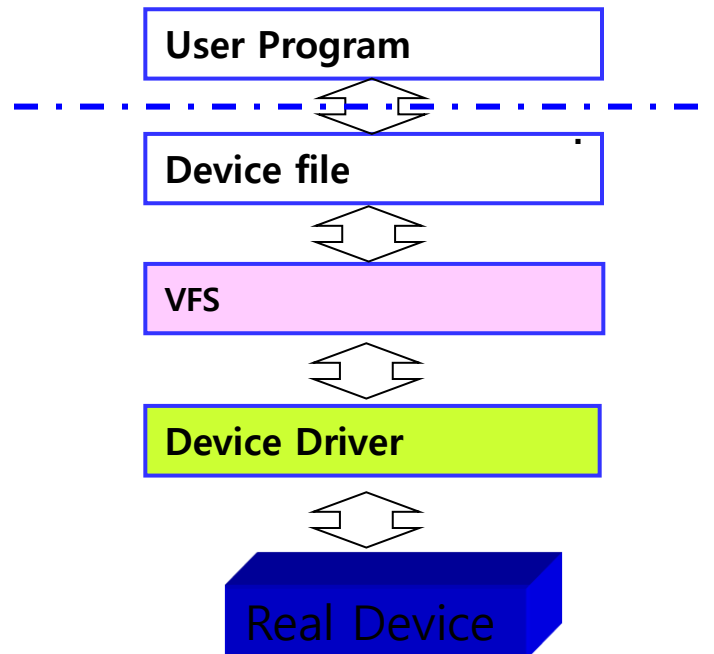


- Non-standard Device Driver Interface
 - Completely user-defined
 - Custom interface
 - May be more appropriate for some hardware



리눅스 디바이스 드라이버

- 사용자 관점에서의 디바이스 드라이버
 - 사용자는 디바이스 자체에 대한 자세한 정보를 알 필요 없음
 - Device는 하나의 파일로 인식됨
 - file에 대한 접근을 통하여 real device에 접근 가능



리눅스 디바이스 드라이버(2)

- 리눅스에서의 디바이스
 - Linux에서 Device는 특별한 하나의 파일처럼 취급되고, access가 가능함.
 - 사용자는 File operation을 적용할 수 있음
 - 각 디바이스는 Major number와 Minor number를 갖음
- Device Driver의 종류
 - 문자 디바이스 드라이버
 - 블록 디바이스 드라이버
 - 네트워크 디바이스 드라이버

Char Device(문자 디바이스)

- 문자 디바이스의 특징
 - 자료의 순차성을 지닌 장치
 - 버퍼 캐쉬를 사용하지 않음
 - 장치의 raw data를 사용자에게 제공
 - Terminal, Serial/Parallel, Keyboard, Sound Card, Scanner, Printer

- 리눅스에서의 문자 디바이스

null : black hole
tty* : virtual console
pt* : pseudo-terminal

c rw--w--w-	0	root	root	5,	1	Oct	1	1998	console
c rw-rw-rw-	1	root	root	1,	3	May	6	1998	null
c rw-----	1	root	root	4,	0	May	6	1998	tty
c rw-rw----	1	root	disk	96,	0	Dec	10	1998	pt0
c rw-----	1	root	root	5,	64	May	6	1998	cua0

파일 관련 정보 중 첫 문자인 C는 char device를 의미

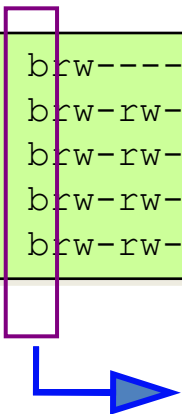
Block Device(블록 디바이스)

- Block device 특징
 - random access 가능
 - 블록 단위의 입출력이 가능한 장치
 - 버퍼캐쉬에 의한 내부 장치 표현
 - 파일 시스템에 의해 mount 되어 관리되는 장치
 - 디스크, Ram Disk, CD-ROM 등

- 리눅스에서의 Block device

fd* : Floppy disk
Hd* : Hard disk
sda : SCSI disk

brw-----	1	root	floppy	2,	0	May	6	1998	fd0
brw-rw----	1	root	disk	3,	0	May	6	1998	hda
brw-rw----	1	root	disk	3,	1	May	6	1998	hda1
brw-rw----	1	root	disk	8,	0	May	6	1998	sda
brw-rw----	1	root	disk	8,	1	May	6	1998	sda1



파일 관련 정보 중 첫 문자인 b는 block device를 의미

Network Device(네트워크 디바이스)

- Network device 특징
 - 대응하는 장치파일이 없음
 - 네트워크 통신을 통해 패킷을 송수신할 수 있는 장치
 - 응용프로그램과의 통신은 표준 파일 시스템관련 콜 대신 socket(), bind() 등의 시스템 콜 사용
 - Ethernet, PPP, ATM, ISDN 등이 있음

Major & Minor Number

- Major number(주번호)
 - 커널에서 디바이스 드라이버를 구분/연결하는데 사용
 - 같은 Device의 종류를 지칭, 1Byte (0~255사이의 값)
- Minor number(부번호)
 - 디바이스 드라이버 내에서 장치를 구분하기 위해 사용
 - 각 Device의 추가적인 정보를 나타냄, 2Byte (부번호)
 - 하나의 디바이스 드라이버가 여러 개의 디바이스 제어 가능
- \$ ls -al /dev/hda*

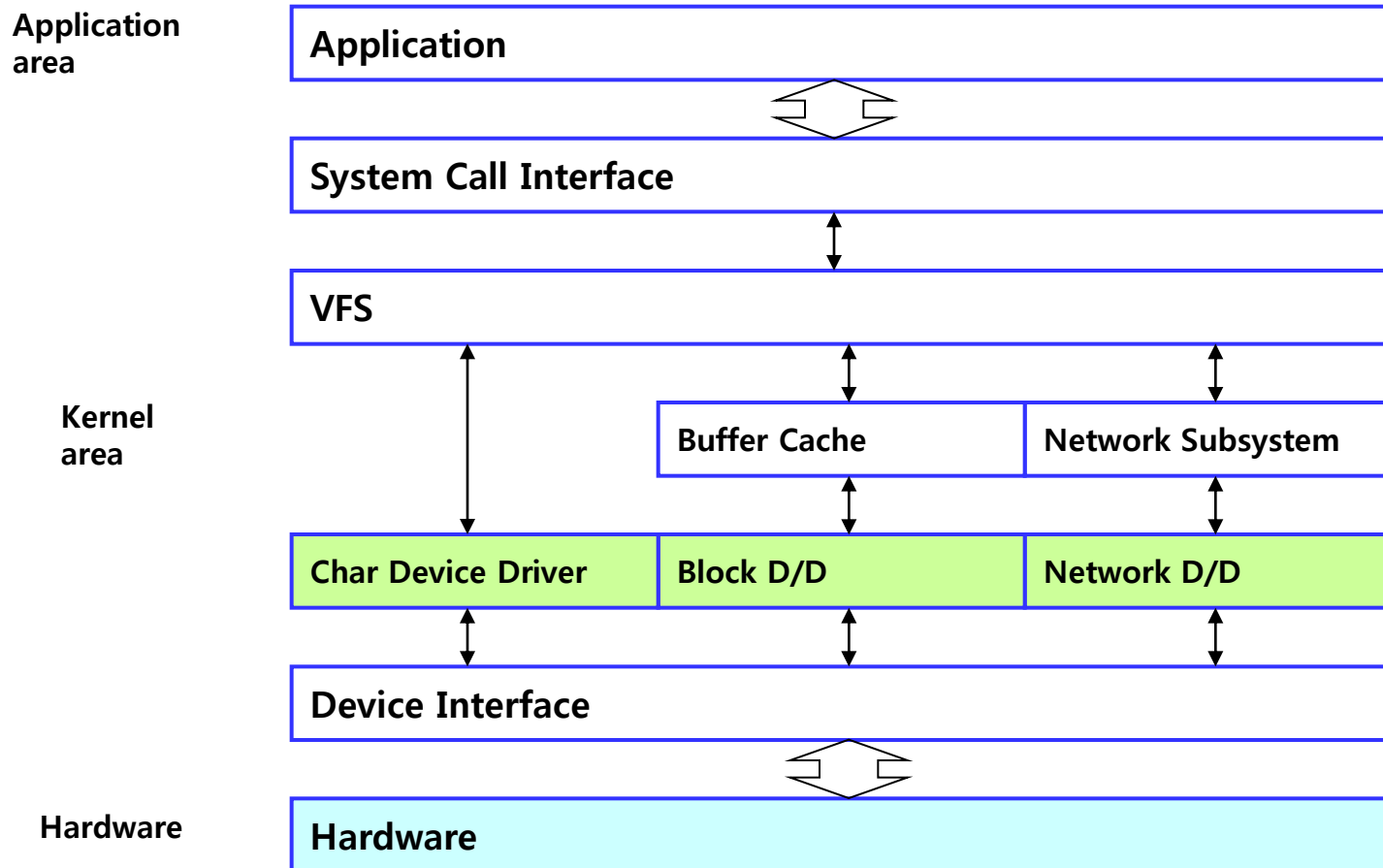
brw-rw----	1	root	disk	1,	0	May	6	1998	hda
brw-rw----	1	root	disk	1,	1	May	6	1998	hda1
brw-rw----	1	root	disk	1,	2	May	6	1998	hda2
brw-rw----	1	root	disk	1,	3	May	6	1998	hda3

주번호

부번호

디바이스 드라이버 구조

- 리눅스 시스템 구조 상의 디바이스 드라이버



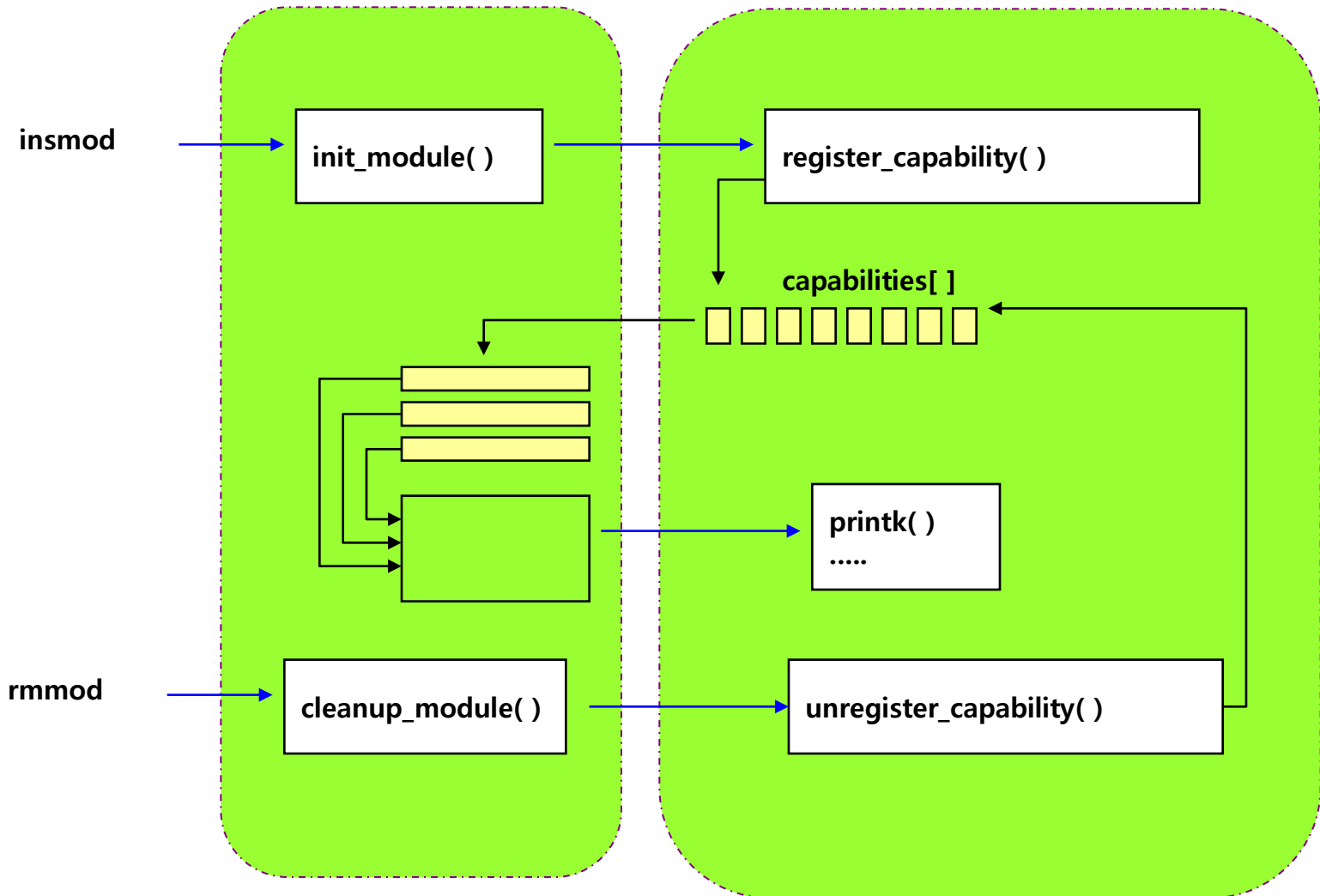
커널 모듈(kernel Module)

- 시스템 부팅 후에 동적으로 loading 할 수 있는 커널 구성요소
- 커널을 다시 컴파일 하거나 시스템 리부팅 할 필요 없이 커널의 일부분을 교체하는 것이 가능
- 디바이스 드라이버, 파일 시스템, 네트워크 프로토콜 등이 모듈로 제공됨
- 컴파일한 커널 버전 정보가 들어가야 하고, 현재 실행되고 있는 커널 버전과 일치해야 함
 - <linux/module.h>에 정의되어 있음
 - 모듈 정보는 전체 모듈에서 하나만 존재해야 함
- 일반 응용 프로그램과의 차이점
 - main() 함수가 없음
 - 커널에 로딩 및 제거 될 때 불러지는 함수가 존재
 - Loading 시 - int init_module(void) 함수 호출
 - Unloading 시 - void cleanup_module() 함수 호출

Linux Device Driver 특성

- Linux device driver의 공통적 특성
 - 커널 코드
 - 디바이스 드라이버는 커널의 한 부분이므로, 커널의 다른 코드와 마찬가지로 잘못되면 시스템에 치명적인 피해를 줄 수 있다
 - 커널 인터페이스
 - 디바이스 드라이버는 리눅스 커널이나 자신이 속한 서브시스템에 표준 인터페이스를 제공해야 한다.
 - 커널 메커니즘과 서비스
 - 디바이스 드라이버는 메모리 할당, 인터럽트 전달, wait queue같은 표준 커널 서비스를 사용할 수 있다.
 - Loadable
 - 대부분의 리눅스 디바이스 드라이버는 커널 모듈로서, 필요할 때 로드하고 더 이상 필요하지 않을 때 언로드 할 수 있다.
 - 설정가능(Configurable)
 - 리눅스 디바이스 드라이버를 커널에 포함하여 컴파일 할 수 있다. 어떤 장치를 넣을 것인지는 커널을 compile 할 때 설정할 수 있다

커널과 모듈의 링크 개념도



커널 모듈의 작성

- 예제 프로그램
 - 커널에 모듈이 로딩될 때 “hello mango world”를 출력
 - 모듈이 제거될 때 “Good Bye”를 출력
 - Source file : hello.c

```
/* hello.c */

#include <linux/module.h> /* 모든 모듈에 필요 */
#include <linux/kernel.h> /* printk() 등에 필요 */

static int __init hello_init(void) { // 모듈이 로딩될 때 호
출
    printk ("hello mango world\n");
        // from a text console, not X-terminal.
    return 0;
    // 0: success , 기타 - fail
}

static void __exit hello_exit(void)
{
    printk ("KERN_ALERT "Goodbye world");
}

module_init(hello_init);
Module_exit(hello_exit);
MODULE_LICENSE("GPL");
```

커널 모듈의 컴파일(커널에 포함)

- 커널 소스 퓌 디렉토리로 이동
- #cp hello.c arch/arm/mach-s5pc100/
- #vi arch/arm/mach-s5pc100/Makefile 수정

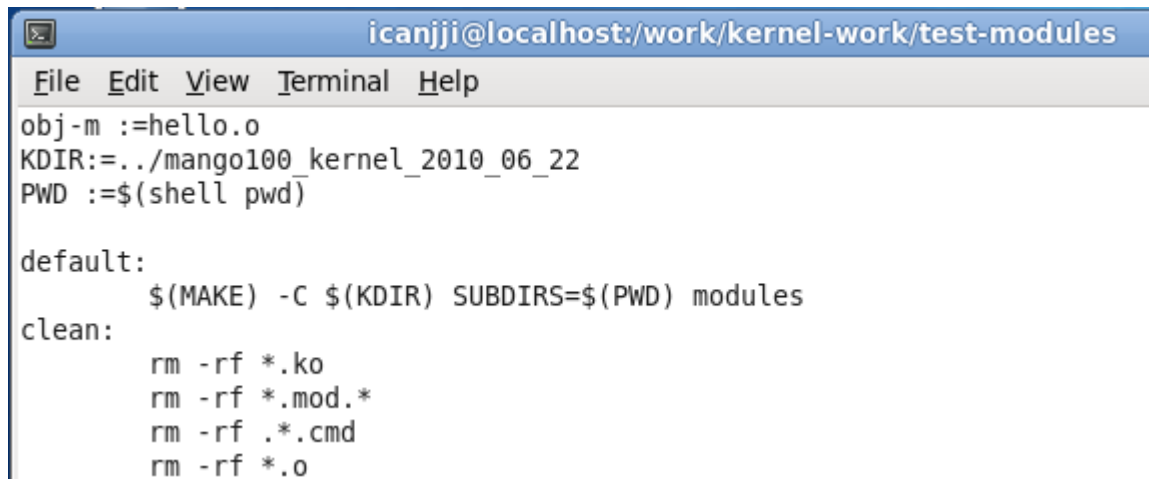
```
# Button EINT supporty  
obj-$(CONFIG_MACH_SMDKC100) += s5pc100-button.o  
obj-m +=hello.o
```

- #make modules

```
CC      arch/arm/mach-s5pc100/hello.mod.o  
LD [M]  arch/arm/mach-s5pc100/hello.ko
```

커널 모듈의 컴파일 (Makefile 작성)

- 커널 소스 퓌 디렉토리로 이동
- #mkdir test-module
- #cp hello.c test-module
- #vi Makefile

A terminal window titled 'icanjji@localhost:/work/kernel-work/test-modules' showing the content of a Makefile. The terminal has a menu bar with 'File', 'Edit', 'View', 'Terminal', and 'Help'. The Makefile content is as follows:

```
obj-m :=hello.o
KDIR:=../mangol00_kernel_2010_06_22
PWD :=$(shell pwd)

default:
    $(MAKE) -C $(KDIR) SUBDIRS=$(PWD) modules

clean:
    rm -rf *.ko
    rm -rf *.mod.*
    rm -rf *.cmd
    rm -rf *.o
```

- #make

module 파일시스템에 포함방법

- NFS 로 Mount 하는 방법
- 저장장치를 이용하는 방법
- 파일시스템에 포함하는 방법
- Etc...

module파일시스템에 포함방법 (NFS 로 Mount 하는 방법)

Mango(타겟) board console창에서 입력

```
#mkdir /mnt/nfs
##echo "/sbin/mount -t nfs -o nolock $(Serverip):$(nfs 디렉토리)
$(dir)
# /sbin/mount -t nfs -o nolock 192.168.0.10:/nfsroot /mnt/nfs
#df
```

```
root@Mango:/# df
Filesystem            1K-blocks    Used Available Use% Mounted on
ubi0:rootfs           233700      168688   59952    74% /
none                  64           52        12    81% /dev
tmpfs                  64           52        12    81% /dev
tmpfs                  100156      472      99684     0% /var/volatile
tmpfs                  100156      0      100156     0% /dev/shm
tmpfs                  100156      0      100156     0% /media/ram
192.168.0.2:/nfsroot  85084704  16757376  64005216  21% /mnt/nfs
```

```
# cd /mnt/nfs
#insmod hello.ko
#lsmod
#rmmod hello.ko
```

Host linux PC 명령 순서

```
#cp hello.ko /nfsroot
#ps -aux | grep nfs
[root@localhost mango100 kernel_2010_06_22]# ps -aux | grep nfs
Warning: bad syntax, perhaps a bogus '-'? See /usr/share/doc/procps-3.2.8,
root      4251  0.0  0.0    0   0 ?        S<    21:20   0:00 [nfsiod]
root      4591  0.0  0.0    0   0 ?        S<    21:54   0:00 [nfsd4]
root      4592  0.0  0.0    0   0 ?        S<    21:54   0:00 [nfsd]
#rpm -qa nfs
#yum install nfs*
#mkdir -p /home/nfsroot
#ln -s /home/nfsroot /nfsroot
#vi /etc/exports 에 아래 추가

/nfsroot *(rw,no_root_squash,no_all_squash)
#ifconfig eth0 192.168.0.10 up
```

```
root@Mango:/mnt/nfs# insmod hello.ko
Hello Mango world
root@Mango:/mnt/nfs# lsmod
Module                Size  Used by
hello                  1084  0
root@Mango:/mnt/nfs# rmmod hello.ko
Good bye Mango
```

디바이스 드라이버의 작성방법

- 커널 모듈의 형태로 디바이스 드라이버 함수 작성
 - struct file_operations 정의 및 함수 구현
 - init_module, module_exit 정의 및 함수 구현
- 커널에 디바이스 드라이버 등록
 - register_chrdev(), register_blkdev(), register_netdev()
- 컴파일/로딩
 - Insmod
- Make special file
 - Mknod
- 드라이버를 활용하는 응용프로그램 작성 및 테스트

설정 - 드라이버 적재 및 삭제

- 노드 생성 - 노드(파일)를 통해서 입출력 수행
 - `mknod /dev/파일이름 드라이버타입 주번호 부번호`
 - 예) `mknod /dev/keydd c 125 0`
 - 생성 후 속성변경 : `chmod ug+w /dev/keydd`
- 디바이스 드라이버 적재
 - `insmod 드라이버명.ko`
 - 예) `insmod keydd.ko`
- 디바이스 드라이버 삭제
 - `rmmmod 드라이버명.ko`
 - 예) `rmmmod keydd .ko` (주의 .ko 붙지 않아도 됨)
- 드라이버의 적재 여부
 - `lsmod`

디바이스 드라이버 - Etc.

- Device의 정보를 가지는 File들
 - /proc/devices
 - 현재 System에 장착되어 있는 Device들의 정보
 - ./Documentation/devices.txt
 - 현재 Linux System에서 정의되어 있는 Device들의 Major, Minor Number들에 대한 정보
 - ./include/linux/major.h
 - Major Number를 define한 header

문자형 디바이스 드라이버 골격

<pre>#include <linux/kernel.h> #include <linux/module.h> #include <linux/init.h></pre>	Header Files
<pre>int device_open(...) { ... } int device_release(...) { ... } ssize_t device_write(...) { ... } ssize_t device_read(...) { ... }</pre>	Function Prototypes
<pre>static struct file_operations device_fops = { ... ssize_t (*read) (...); ssize_t (*write) (...); ... int (*open) (...); int (*release) (...); ... };</pre>	File Operation
<pre>int init_module(void) { ... }</pre>	모듈 설치 시 초기화 수행
<pre>Void module_exit(void) { ... }</pre>	모듈 제거 시 반환 작업수행

Device Driver 작성(1)

- Device structure

- 디바이스 구조체 : 2 개의 필드로 구성된 구조체

- Name field
 - file_operation files

```
static struct char_device_struct {
    struct char_device_struct *next;
    unsigned int major;
    unsigned int baseminor;
    int minorct;
    char name[64];
    struct cdev *cdev;          /* will die */
} *chrdevs[CHRDEV_MAJOR_HASH_SIZE];
```

```
struct file_operations { /* include/linux/fs.h */
```

```
    lseek;
    read;
    write;
    readdir;
    poll;
    ioctl;
    mmap;
    open;
    flush;
    release;
```

```
    ...
};
```

```
struct file_operations {
    struct module *owner;
    loff_t (*llseek) (struct file *, loff_t, int);
    ssize_t (*read) (struct file *, char __user *, size_t, loff_t *);
    ssize_t (*write) (struct file *, const char __user *, size_t, loff_t *);
    ssize_t (*aio_read) (struct kiocb *, const struct iovec *, unsigned long, loff_t);
    ssize_t (*aio_write) (struct kiocb *, const struct iovec *, unsigned long, loff_t);
    int (*readdir) (struct file *, void *, filldir_t);
    unsigned int (*poll) (struct file *, struct poll_table_struct *);
    int (*ioctl) (struct inode *, struct file *, unsigned int, unsigned long);
    long (*unlocked_ioctl) (struct file *, unsigned int, unsigned long);
    long (*compat_ioctl) (struct file *, unsigned int, unsigned long);
    int (*mmap) (struct file *, struct vm_area_struct *);
    int (*open) (struct inode *, struct file *);
    int (*flush) (struct file *, filp_owner_t id);
    int (*release) (struct inode *, struct file *);
    int (*fsync) (struct file *, struct dentry *, int datasync);
    int (*aio_fsync) (struct kiocb *, int datasync);
    int (*fsync) (int, struct file *, int);
};
```

Device Driver 작성(2)

- 디바이스 드라이버 등록
 - 드라이버를 커널에 등록하고, 파일 연산을 정의하는 등의 초기화 작업 수행이 필요
 - 모듈의 형태에서는 `init_module()` 함수에서 초기화 수행
 - 드라이버의 등록 함수

```
int register_chrdev( unsigned int major,  
                    const * name,  
                    struct file_operations * fops);
```

- 커널에 지정되어 있는 `chrdevs` 구조에 새로운 char device 등록
 - major number : 주번호, 0을 주면 사용하지 않는 값을 반환
 - name : 디바이스 이름으로 `/proc/devices`에 나타남
 - fops: 디바이스와 연관된 파일 연산 구조체 포인터
- 음수가 반환되면 오류가 발생했음을 나타냄

Device Driver 작성(3)

- 디바이스 드라이버 제거
 - 더 이상 사용되지 않는 드라이버의 제거
 - Rmmod하는 명령을 이용하여 제거하며, 이때 드라이버 내의 `cleanup_module`이 호출되는데, 이 루틴 안에서 다음의 시스템 콜을 호출
 - 드라이버 제거 함수

```
int unregister_chrdev( unsigned int major, const * name);
```

Device Driver 작성(4)

- 파일 연산
 - 디바이스 드라이버를 일반적인 파일과 유사한 인터페이스를 이용하여 관리
 - 각 디바이스는 파일 형태로 존재하고, 커널은 파일 연산을 이용하여 I/O 연산을 수행하도록 인터페이스 구성
 - 디바이스 드라이버를 구현한다는 것은 상당부분 파일연산 구조체에서 요구되는 기능들을 프로그래밍 한다는 것을 의미
 - 가상의 dummy character device 구현예제에서의 파일연산 구조체 정의 예

Device Driver 작성(5)

- 파일 연산 구조체의 전체 구조

```
struct file_operations { /* <linux/fs.h> */
    struct module *owner;
    loff_t (*llseek) (struct file *, loff_t, int);
    ssize_t (*read) (struct file *, char *, size_t, loff_t *);
    ssize_t (*write) (struct file *, const char *, size_t, loff_t *);
    int (*readdir) (struct file *, void *, filldir_t);
    unsigned int (*poll) (struct file *, struct poll_table_struct *);
    int (*ioctl) (struct inode *, struct file *, unsigned int, unsigned long);
    int (*mmap) (struct file *, struct vm_area_struct *);
    int (*open) (struct inode *, struct file *);
    int (*flush) (struct file *);
    int (*release) (struct inode *, struct file *);
    int (*fsync) (struct file *, struct dentry *, int datasync);
    int (*fasync) (int, struct file *, int);
    int (*lock) (struct file *, int, struct file_lock *);
    ssize_t (*readv) (struct file *, const struct iovec *, unsigned long, loff_t *);
    ssize_t (*writev) (struct file *, const struct iovec *, unsigned long, loff_t *);
};
```


Device Driver 작성(6)

- File operations

```
loff_t (*llseek)(struct file *, loff_t, int);
```

→현재의 read/write 위치를 옮긴다.

```
ssize_t (*read)(struct file *, char *, size_t, loff_t *);
```

→디바이스에서 데이터를 가져오기 위해서 사용

```
ssize_t (*write)(struct file*, const char*, size_t, loff_t*);
```

→디바이스에 데이터를 쓰기 위해서 사용

```
int (*readdir)(struct file *, void *, filldir_t);
```

→디렉토리를 다룰 때 사용

```
unsigned int (*poll)(struct file*, struct poll_table_struct*);
```

→현재 프로세스를 대기 큐에 넣기 위해서 사용

Device Driver 작성(7)

- File operations

int (***ioctl**) (struct inode *, struct file *, unsigned int, unsigned long);

→디바이스에 종속적인 명령을 만들기 위해 사용

int (***mmap**) (struct file *, struct vm_area_struct *);

→디바이스 메모리를 프로세스 메모리에 매핑

int (***open**) (struct inode *, struct file *);

→디바이스 노드에 의해 수행되는 첫번째 동작

int (***flush**) (struct file *);

→디바이스를 닫기 전에 모든 데이터를 쓴다.

int (***release**) (struct inode *, struct file *);

→디바이스를 닫을 때 수행

Device Driver 작성(8)

- File operations

```
int (*fsync) (struct file *, struct dentry *);
```

→버퍼에 있는 데이터를 모두 디바이스에 쓴다

```
int (*fasync) (int, struct file *, int);
```

```
int (*check_media_change) (kdev_t dev);
```

→블록 디바이스에서 사용, 제거 가능한 미디어에 사용

```
int (*revalidate) (kdev_t dev);
```

→블록 디바이스에서 사용, 버퍼 캐시의 관리와 상관

```
int (*lock) (struct file *, int, struct file_lock *);
```

→파일에 lock을 걸기 위해서 사용

Device Driver 작성(9)

- Dummy Character device 드라이버 소스코드

```
#include <linux/init.h>
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/fs.h>
#include <linux/errno.h>
#include <linux/types.h>
#include <linux/fcntl.h>

#define DUMMY_MAJOR_NUMBER 254 /* dummy.c */
#define DEV_NAME "dummy-device"
int dummy_open(struct inode *inode,struct file *filp)
{
    int num=MINOR(inode->i_rdev);
    printk(" Open call for Dummy Char Device  call open ->minor:%d\n",num);
    return 0;
}
loff_t dummy_llseek(struct file *filp,loff_t off,int whence)
{
    printk("call llseek->off:%08x, whenec:%08x\n",off,whence);
    return 0x23;
}
```

Device Driver 작성(10)

- Dummy Character device 드라이버 소스코드

```
ssize_t dummy_read(struct file *filp,char *buf, size_t count,loff_t *f_pos)
{
    printk("call read ->buf:%08x, count:%08x\n",buf,count);
    return 0x33;
}
ssize_t dummy_write(struct file *filp,const char *buf, size_t count,loff_t *f_pos)
{
    printk("call write->buf:%08x, count:%08x\n",buf,count);
    return 0x43;
}
int dummy_ioctl(struct inode *inode, struct file *filp,unsigned int cmd,unsigned long arg)
{
    printk("call ioctl->cmd:%08,arg:%08x\n",cmd,arg);
    return 0x53;
}
int dummy_release(struct inode *inode, struct file *filp)
{
    printk(" Release call for Dummy Char Device \n");
    return 0;
}
```

Device Driver 작성(11)

- Dummy Character device 드라이버 소스코드

```
struct file_operations dummy_fops =
{
    .owner = THIS_MODULE,
    .llseek = dummy_llseek,
    .open = dummy_open,
    .read = dummy_read,
    .write = dummy_write,
    .ioctl = dummy_ioctl,
    .release = dummy_release,
};
int dummy_init(void)
{
    int result;
    printk("call dummy_init\n");
    result = register_chrdev(DUMMY_MAJOR_NUMBER, DEV_NAME, &dummy_fops);
    if (result < 0) return result;
    return 0;
}
void dummy_exit(void)
{
    printk("call dummy_exit\n");
    unregister_chrdev(DUMMY_MAJOR_NUMBER, DEV_NAME);
}
module_init(dummy_init);
module_exit(dummy_exit);

MODULE_LICENSE("GPL");
```

Device Driver 작성(12)

- System call : dummy_open
 - File_operations 구조체에서 open operation 구현
 - Application program에서 'open' 에 의해서 불러짐

```
int dummy_open(struct inode *inode, struct file *file)
{
    printk("Open call for Dummy Char Device %n");
    return 0;
}
```

- System call : dummy_release
 - File_operations 구조체에서 release operation 구현
 - Application program에서 'close' 에 의해서 불러짐

```
int dummy_release(struct inode *inode, struct file *file)
{
    printk("Release call for Dummy Char Device %n");
    return 0;
}
```

Device Driver 작성(11)

- System call : dummy_read
 - File_operations 구조체에서 read operation 구현
 - Application program에서 'read' 에 의해서 불러짐

```
ssize_t dummy_read(struct file *file, char *buffer, size_t length, loff_t *offset)
{
    printk("Read Call for Dummy Device %n");
    buffer[0] = 0x34;    return 0;
}
```

- System call : dummy_write
 - File_operations 구조체에서 write operation 구현
 - Application program에서 'write' 에 의해서 불러짐

```
ssize_t dummy_write(struct file *file, const char *buffer, size_t length, loff_t *offset)
{
    printk("Write Call for Dummy Device : [%x]%n ", buffer[0]);
    return 0;
}
```


드라이버 컴파일/로딩/노드 생성

- 디바이스 드라이버 컴파일
- Makefile 작성

```
obj-m:=dummy-driver.o
KDIR:=../mango100_kernel_2010_06_30
PWD:=$(shell pwd)

default:
    $(MAKE) -C $(KDIR) SUBDIRS=$(PWD)
modules
clean:
    rm -rf *.ko
    rm -rf *.mod.*
    rm -rf *.cmd
    rm -rf *.o
```

- #make

module 파일시스템에 포함방법 (저장장치를 이용하는 방법)

- Usb stick, MMC 등 저장장치를 HOST PC에 삽입
- #df
- #cp dummy-driver.ko /mount 디렉토리
- HOST PC 저장장치 분리, 망고 보드에 삽입
- #insmod dummy-driver.ko

```
root@Mango:~# insmod dummy-driver.ko
call dummy_init
insmod: error inserting 'dummy-driver.ko': -1 Device or resource busy
```

에러 발생 시 #ls /sys/dev/char |grep 주번호, 확인 후 소스에서 major 번호 수정

```
root@Mango:/sys/dev/char# ls -al | grep 254
lrwxrwxrwx 1 root root 0 Apr 17 00:49 10:254 -> ../../class/misc/s3c-jpg
lrwxrwxrwx 1 root root 0 Apr 17 00:49 254:0 -> ../../class/rtdr/rtdr0
lrwxrwxrwx 1 root root 0 Apr 17 00:49 2:254 -> ../../class/tty/tty
lrwxrwxrwx 1 root root 0 Apr 17 00:49 3:254 -> ../../class/tty/tty
```

- #cat /proc/modules

장치파일 등록(1/2)

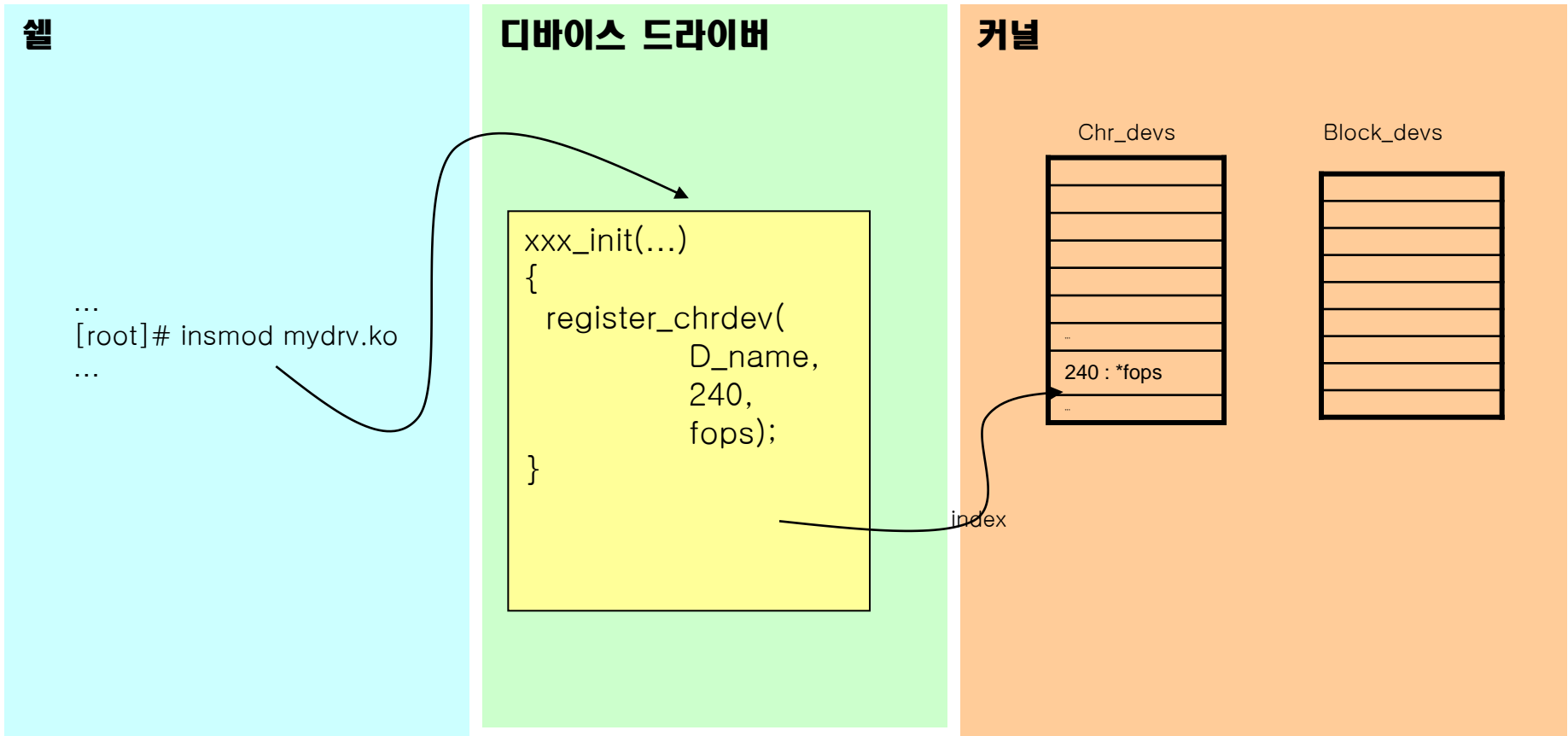
- 장치 파일 등록

```
[root]# insmod mydrv_dev.ko
```

```
int xxx_init(void)
{
    int result;
    result = register_chrdev(RDWR_DEV_MAJOR, RDWR_DEV_NAME, &xxx_fops);
    ....
    return 0;
}
module_init(xxx_init);
```

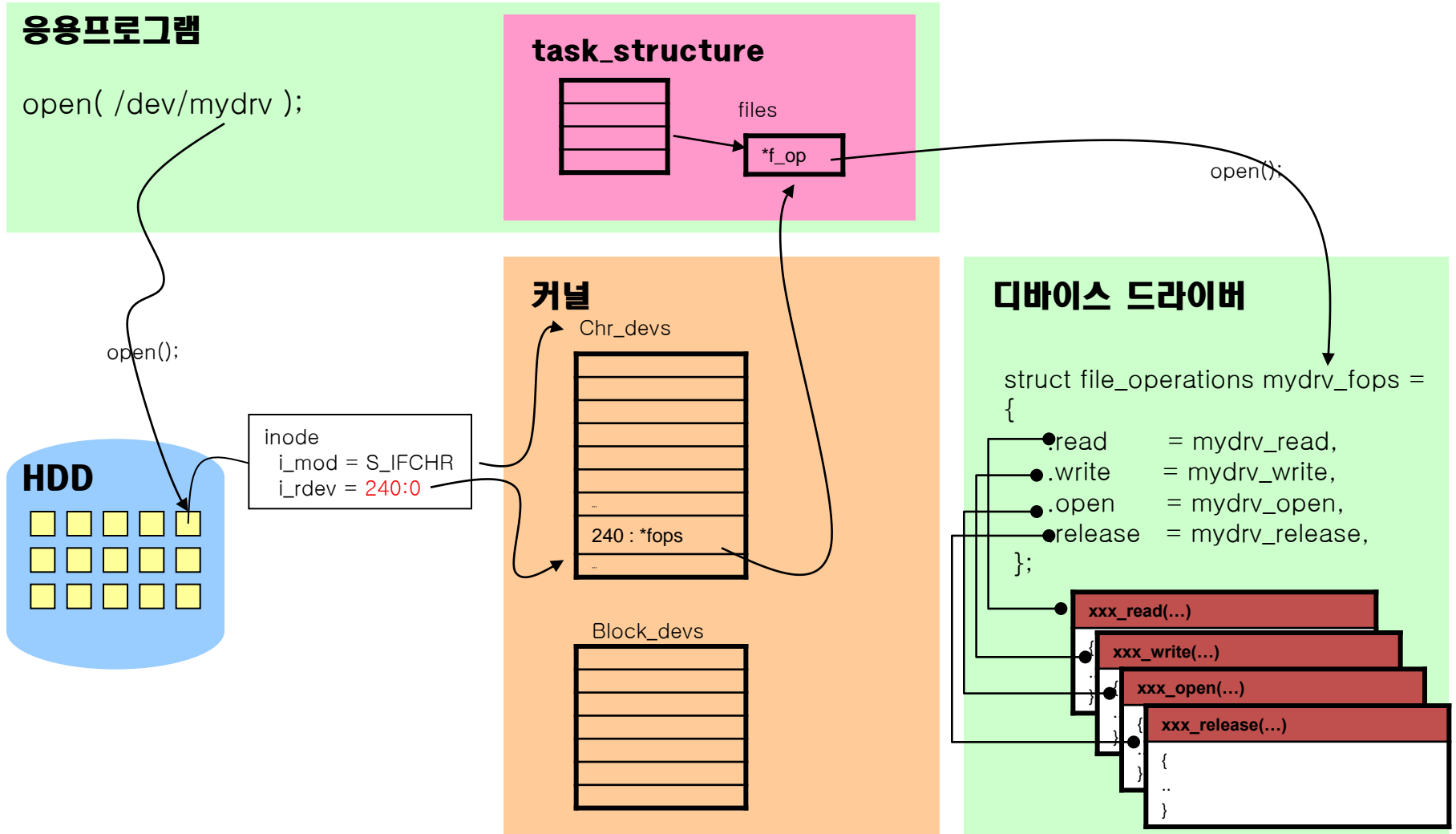
장치파일 등록(2/2)

- register_chrdev 의 기능



Open 의 이해

- Open 과정



Application Program 작성

- Read / write application program 작성
 - 작성한 dummy character device를 테스트
 - Dummy-device를 열고 문자열을 read /write 함
 - Console의 커널 내부 정보를 통해 실제 dummy_device를 통해서 read / write가 이루어졌는지 확인

Application Program 작성

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/ioctl.h>
#include <fcntl.h>
#include <unistd.h>
#define DEVICE_FILENAME "/dev/dummy-driver"
int main(){
    int dev;
    char buff[128];
    int ret;

    printf("1) device file open\n");

    dev = open(DEVICE_FILENAME, O_RDWR|O_NDELAY);
```

```
    if(dev >= 0)
    {
        printf("2) seek function call\n");

        ret = lseek(dev, 0x20, SEEK_SET);
        printf("ret = %08X\n", ret);

        printf("3) read function call\n");

        ret = read(dev, 0x30, 0x31);
        printf("ret = %08X\n", ret);

        printf("4) write function call\n");
        ret = write(dev, 0x40, 0x41);
        printf("ret = %08X\n", ret);
        printf("5) ioctl function call\n");
        ret = ioctl(dev, 0x51, 0x52);
        printf("ret = %08X\n", ret);

        printf("6) device file close\n");
        ret = close(dev);
        printf("ret = %08X\n", ret);

    }
    return 0;
}
```

Application Program 컴파일/적재

- `#arm-linux-gcc -o dummy-app.o dummy-app.c`
- Dummy-app.o파일을 망고보드 파일 시스템에 복사
- `#cp dummy-app.o /nfsroot`
- `#!/sbin/mount -t nfs -o nolock 192.168.0.10:/nfsroot /mnt/nfs`
- `#cp /mnt/nfs/dummy-app.o ~`

Application �행 방법

- #mknod /dev/dummy-driver c 240 32
- #./dummy-app.o

```
root@Mango:/sys/module# mknod /dev/dummy-driver c 240 32
root@Mango:/sys/module# cd /mnt/nfs
root@Mango:/mnt/nfs# ./dummy-app.o
1) device file open call open ->minor:32
call llseek->off:00000000, whenec:00000020
call read ->buf:00000030, count:00000031
call write->buf:00000040, count:00000041
call ioctl->cmd:%,arg:00000051
call release

2) seek function call
ret = 00000023
3) read function call
ret = 00000033
4) write function call
ret = 00000043
5) ioctl function call
ret = 00000053
6) device file close
ret = 00000000
```

커널 포팅,안드로이드 키처리

삼성 커널 다운받기

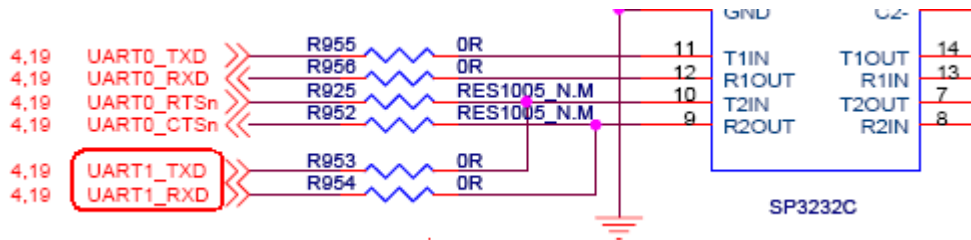
- #git clone
git://git.kernel.org/pub/scm/linux/kernel/
git/kki_ap/linux-2.6-samsung.git
- #cd linux-2.6-samsung
- #git fetch
- #git fetch --tag
- #git checkout 2.6.29-samsung

망고100 커널 소스 받기

- #wget
http://crztech.iptime.org:8080/Release/mango100/kernel/mango100_kernel_2010_06_30.tgz
- #tar xvf
mango100_kernel_2010_06_30.tgz

DEBUG PORT 포팅

```
$(topdir)/config 파일에서  
CONFIG_S3C_LOWLEVEL_UART_PORT=1  
# CONFIG_SPLIT_ROOT_FILESYSTEM is not set
```



Kernel configuration에서
"kernel hacking-> S3C UART to use for low-level debug"
에서 1 선택

```
# cat /proc/devices  
Character devices:  
1 mem  
2 tty  
3 ttyp  
4 /dev/vc/0  
4 tty  
4 ttyS  
5 /dev/tty  
5 /dev/console  
5 /dev/ptmx  
7 vcs  
10 misc  
13 input  
14 sound  
29 fb  
81 video4linux  
89 i2c  
90 mtd  
116 alsa  
128 ptm  
136 pts  
180 usb  
189 usb_device  
204 s3c2410_serial
```

./arch/arm/plat-s3c/include/plat/uncompress.h 소스에 포함
#define uart_base S3C_PA_UART + (S3C_UART_OFFSET * CONFIG_S3C_LOWLEVEL_UART_PORT)

.dev_name = "s3c2410_serial",
./drivers/serial/samsung.c 에서 디바이스 이름을 수정 할 수 있습니다.

```
# cat /proc/tty/drivers  
/dev/tty /dev/tty 5 0 system:/dev/tty  
/dev/console /dev/console 5 1 system:console  
/dev/ptmx /dev/ptmx 5 2 system  
/dev/vc/0 /dev/vc/0 4 0 system:vtmaster  
ttySAC /dev/s3c2410_serial 204 64-67 serial  
serial /dev/ttyS 4 64-67 serial  
pty_slave /dev/pts 136 0-1048575 pty:slave  
pty_master /dev/ptm 128 0-1048575 pty:master  
pty_slave /dev/ttyp 3 0-255 pty:slave  
pty_master /dev/pty 2 0-255 pty:master  
unknown /dev/tty 4 1-63 console
```

LCD 드라이버 수정-1

#make menuconfig

"Device Drivers->Graphic Support

->Support for frame buffer devices

->S3C Framebuffer support 선택

```
.config - Linux Kernel v2.6.29 Configuration
----- Support for frame buffer devices -----
Arrow keys navigate the menu.  <Enter> selects submenus --->.
Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes,
<M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </>
for Search.  Legend: [*] built-in [ ] excluded <M> module < >

(-)
[ ] Framebuffer foreign endianness support --->
[*] Enable Video Mode Handling Helpers
[ ] Enable Tile Blitting Support
*** Frame buffer hardware drivers ***
<+> S3C Framebuffer support
[ ] S3C Framebuffer debug messages
[ ] S3C Framebuffer FIFO underrun trace
(0) Default Window (0-4)
(1) Vertical Panning Step (0-2)
[ ] Select LCD Type (LTE480WV) --->

<Select>  < Exit >  < Help >
```

LCD 드라이버 수정-2



```
void lcd_power_ctrl(s32 value)
{
    int err;

    if (value) {
        if (gpio_is_valid(S5PC1XX_GPH0(2))) {
            err = gpio_request(S5PC1XX_GPH0(2), "GPH0");

            if (err) {
                printk(KERN_ERR "failed to request GPH0 for "
                    "lcd reset control\n");
            }
            gpio_direction_output(S5PC1XX_GPH0(2), 1);
        }
    }
    else {
        if (gpio_is_valid(S5PC1XX_GPH0(2))) {
            err = gpio_request(S5PC1XX_GPH0(2), "GPH0");

            if (err) {
                printk(KERN_ERR "failed to request GPH0 for "
                    "lcd reset control\n");
            }
            gpio_direction_output(S5PC1XX_GPH0(2), 0);
        }
    }
    gpio_free(S5PC1XX_GPH0(2));
    lcd_power = value;
}
```

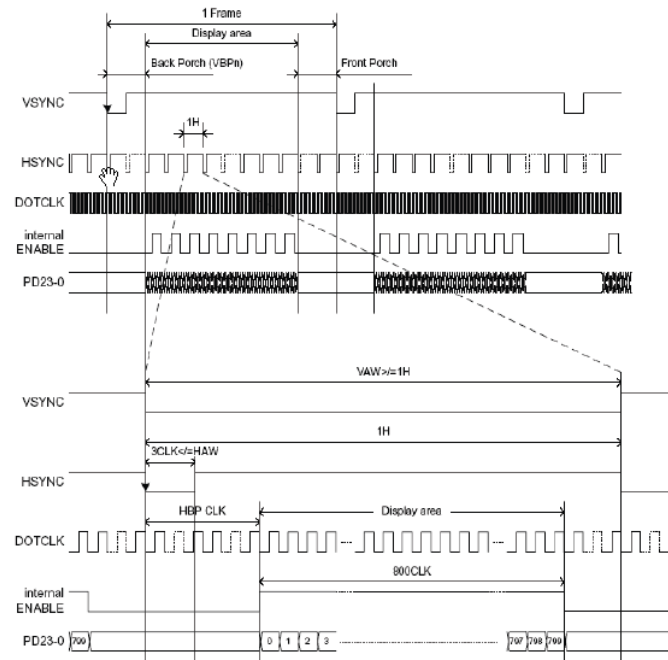
./drivers/video/samsung/s3cfb_lte480wv.c
에서 lcd_power_ctrl함수를 수정
LCD 에 전원을 인가한다.

LCD 드라이버 수정-3

```
static struct s3cfb_lcd lte480wv = {
    .width = 800,
    .height = 480,
    .bpp = 24,
    .freq = 60,

    .timing = {
        .h_fp = 8,
        .h_bp = 13,
        .h_sw = 3,
        .v_fp = 5,
        .v_fpe = 1,
        .v_bp = 7,
        .v_bpe = 1,
        .v_sw = 1,
    },

    .polarity = {
        .rise_vclk = 0,
        .inv_hsync = 1,
        .inv_vsync = 1,
        .inv_vden = 0,
    },
};
```



7-1. Vertical timing

Signal	Symbol	Min.	Typ.	Max.	Unit	Note
Frame Frequency	fFRM	-	60	-	Hz	
Vertical Back porch	VBP	-	8	-	H	*Note
Vertical Front porch	VFP	-	5	-	H	*Note

7-2. Horizontal timing

Signal	Symbol	Min.	Typ.	Max.	Unit	Note
Horizontal Back porch	HBP	-	16	-	DOTCLK	*Note
Horizontal Front porch	HFP	-	8	-	DOTCLK	*Note
DOTCLK Frequency	fDOTCLK	-	24.5	-	MHz	@fFRM=60Hz

drivers/video/samsung/s3cfb_lte480wv.c
에서 LCD datasheet 타이밍만 맞추면 됨

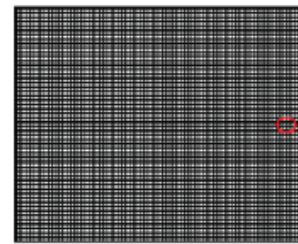
LCD 드라이버 수정-4

```
static int s3cfb_init_global(void)
{
    fbdev->output = OUTPUT_RGB;
    fbdev->rgb_mode = MODE_BGR_P;

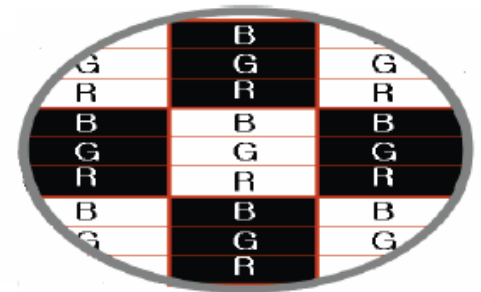
    fbdev->wq_count = 0;
    init_waitqueue_head(&fbdev->wq);
    mutex_init(&fbdev->lock);

    s3cfb_set_output(fbdev);
    s3cfb_set_display_mode(fbdev);
    s3cfb_set_polarity(fbdev);
    s3cfb_set_timing(fbdev);
    s3cfb_set_lcd_size(fbdev);

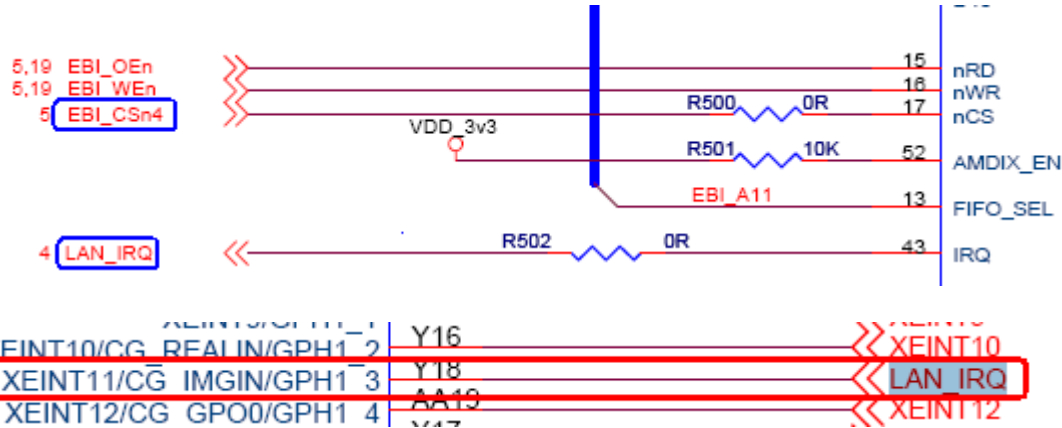
    return 0;
}
```



▶ 1 Dot pattern



이더넷 드라이브 포팅-1



```
static struct resource s3c_sm911x_resources[] = {
    [0] = {
        .start = S5PC1XX_PA_SM9220,
        .end   = S5PC1XX_PA_SM9220 + SZ_1M - 1,
        .flags = IORESOURCE_MEM,
    },
    [1] = {
        .start = IRQ_EINT11,
        .end   = IRQ_EINT11,
        .flags = IORESOURCE_IRQ,
    },
};
```

arch/arm/plat-s5pc1xx/devs.c 에서 수정

```
#define S5PC1XX_PA_SM9220 (0xA0000000)
/arch/arm/mach-s5pc100/include/mach/map.h
```

0xA000_0000	0xA800_0000	128MB	SMC Bank 4
-------------	-------------	-------	------------

이더넷 드라이버 -3

```
static void __init mango100_smc911x_set(void)
{
    unsigned int tmp;

    tmp = __raw_readl(S5PC1XX_GPKDCON);
    tmp &=~S5PC1XX_GPKD_4_MASK;
    tmp |= (S5PC1XX_GPKD_4_SROM_CSn4);
    __raw_writel(tmp, S5PC1XX_GPKDCON);

    tmp = __raw_readl(S5PC1XX_SROM_BW);
    tmp &= ~(S5PC1XX_SROM_BW_BYTE_ENABLE4_MASK | S5PC1XX_SROM_BW_WAIT_ENABLE4_M
ASK |
        S5PC1XX_SROM_BW_ADDR_MODE4_MASK | S5PC1XX_SROM_BW_DATA_WIDTH4_MASK);
    tmp |= S5PC1XX_SROM_BW_DATA_WIDTH4_16BIT;
    tmp |= S5PC1XX_SROM_BW_ADDR_MODE4_BYTE_ADDR;

    __raw_writel(tmp, S5PC1XX_SROM_BW);

    __raw_writel((0x0<<28)|(0x4<<24)|(0xd<<16)|(0x1<<12)|(0x4<<8)|(0x6<<4)|(0x0
<<0), S5PC1XX_SROM_BC4);
}
```

```
static struct platform_device *mango100_devices[] __initdata = {
#ifdef CONFIG_FB_S3C
    &s3c_device_fb,
#endif
    &s3c_device_rtc,
    &s3c_device_i2c0,
    &s3c_device_nand,
#ifdef CONFIG_TOUCHSCREEN_S3C
    &s3c_device_ts,
#endif
    &s3c_device_smc911x,
    &mango_battery,
}
```

./arch/arm/mach-s5pc100/mach-mango100.c
디바이스 등록 소스는
./arch/arm/plat-s5pc1xx/devs.c
에 위치

이더넷 디바이스 확인 방법

```
# cat /proc/iomem
20000000-2fffffff : System RAM
2002f000-20477fff : Kernel text
20478000-205526eb : Kernel data
a0000000-a00fffff : smsc911x
a0000000-a00ffffe : smsc911x
```

```
# ls /sys/devices/platform/smsc911x/
driver      mdio_bus    net          subsystem
ffffffff:01 modalias     power        uevent
# ls -al /sys/devices/platform/smsc911x/
drwxr-xr-x  6 0      0          0 Apr 21 19:56 .
drwxr-xr-x 40 0      0          0 Apr 21 19:56 ..
lrwxrwxrwx  1 0      0          0 Apr 22 01:06 driver -> ../../../../bus/
platform/drivers/smsc911x
drwxr-xr-x  3 0      0          0 Apr 21 19:56 ffffffff:01
drwxr-xr-x  3 0      0          0 Apr 21 19:56 mdio_bus
-r--r--r--  1 0      0          4096 Apr 22 01:06 modalias
drwxr-xr-x  3 0      0          0 Apr 21 19:56 net
drwxr-xr-x  2 0      0          0 Apr 21 19:56 power
lrwxrwxrwx  1 0      0          0 Apr 22 01:06 subsystem -> ../../../../b
us/platform
-rw-r--r--  1 0      0          4096 Apr 21 19:56 uevent
```

```
# cat /proc/net/dev
Inter-|   Receive                       |   Transmit
face |bytes  packets errs drop fifo frame compressed multicast|bytes  packe
ts errs drop fifo colls carrier compressed
lo:   0      0      0      0      0      0      0      0      0      0
eth0: 763466 11632  0      0      0      0      0      0      11623 2052
16:   0      0      0      0      0      0      0      0      0
```

이더넷 드라이버 확인 방법

```
# ls -al /sys/devices/platform/msmc911x/driver/
drwxr-xr-x  2 0      0      0 Apr 22 01:06 .
drwxr-xr-x 38 0      0      0 Apr 22 01:06 ..
--w-----  1 0      0      0      4096 Apr 22 03:17 bind
lrwxrwxrwx  1 0      0      0      0 Apr 22 03:17 smsc911x -> ../../../../
/devices/platform/msmc911x
--w-----  1 0      0      0      4096 Apr 22 03:17 uevent
--w-----  1 0      0      0      4096 Apr 22 03:17 unbind
```

이더넷 드라이버 irq 등록확인

```
# cat /proc/interrupts
CPU0
20:      694      s3c-uart  s5pc100-uart
22:     1631      s3c-uart  s5pc100-uart
37:         0  s3c_vic_eint  s5p-tvout
43:    23801  s3c_vic_eint  eth0
44:         0          VIC  I2S PCM Stereo out
```

S5pc100 user manual에는 11번째에 EINT11이다, 근데 등록은 32를 더한 값인 43번째 등록이 되어 있다.

이유는 커널이 32번까지 **spurious** 인터럽트로 등록하여 사용한다.

11

EINT11

EXT_INT[11]

```
#define IRQ_EINT11          S5PC1XX_IRQ_VIC0(11)
./arch/arm/plat-s5pc1xx/include/plat/irqs.h
```

Keypad driver

```

Keyboard configuration menu
-----
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < > module

Keyboard configuration menu
-----
--- Keyboards
  < > AT keyboard
  < > Sun Type 4 and Type 5 keyboard
  < > DECstation/VAXstation LK201/LK401 keyboard
  < > XT keyboard
  < > Newton keyboard
  < > Stowaway keyboard
  < > GPIO Buttons
  [*] Mango-100 board keypad support
  [*] Linux Output Event
  < > S3C keypad support
  Select keypad port

  <Select> < Exit > < Help >

```

./driver/input/keyboard/Kconfig 에 추가
.config 파일에
CONFIG_KEYPAD_MANGO100_LINUX_EVENT=y

```

config KEYPAD_MANGO100
    tristate "Mango-100 board keypad support"
    depends on (CPU_S3C6400 || CPU_S3C6410 || CPU_S5PC100)
    default n
    help
        Say Y here if you want to use the Mango100 keypad.
        To compile this driver as a module, choose M here: the
        module will be called mangoc100-keypad.

config KEYPAD_MANGO100_LINUX_EVENT
    bool "Linux Output Event"
    depends on KEYPAD_MANGO100
    default n
    help
        Say Y here if you want to use the Mango100 keypad for Linux Event.
        To compile this driver as a module, choose M here: the
        module will be called mangoc100-keypad.

```


Keypad driver

```
[icanjji@localhost kernel]$ ls drivers/input/keyboard/mango+  
drivers/input/keyboard/mango100_keycode.h  
drivers/input/keyboard/mango100_keypad.c  
drivers/input/keyboard/mango100_keypad.h  
drivers/input/keyboard/mango100_keypad.o  
drivers/input/keyboard/mango100_keypad_sysfs.c  
drivers/input/keyboard/mango100_keypad_sysfs.h  
drivers/input/keyboard/mango100_keypad_sysfs.o
```

./driver/input/keyboard/ 파일을 추가한다.

```
obj-$(CONFIG_KEYPAD_S3C) += s3c-keypad.o  
obj-$(CONFIG_KEYPAD_MANGO100) += mango100_keypad.o mango100_keypad_sysfs.o
```

./driver/input/keyboard/Makefile 에 추가

Keypad driver

```
static int __devinit mango100_keypad_probe(struct platform_device *pdev)
{
    int rc;
    int key, code;

    // struct init
    memset(&mango100_keypad, 0x00, sizeof(mango100_keypad_t));

    // create sysfs
    if((rc = mango100_keypad_sysfs_create(pdev))) {
        printk("%s : sysfs_create_group fail!!\n", __FUNCTION__);
        return rc;
    }
}
```

Driver/input/keyboard/mango100-keypad.c
등록확인 은 아래와 같이 확인 할 수 있다.

```
# ls -al /sys/devices/platform/mango100-keypad.0
drwxr-xr-x  3 0      0          0 Apr 21 19:56 .
drwxr-xr-x 40 0      0          0 Apr 21 19:56 ..
lrwxrwxrwx  1 0      0          0 Apr 22 02:25 driver -> ../../../../bus/
platform/drivers/mango100-keypad
-r--r--r--  1 0      0          4096 Apr 22 02:25 hdmi_connect_state
-r--r--r--  1 0      0          4096 Apr 22 02:25 modalias
drwxr-xr-x  2 0      0          0 Apr 21 19:56 power
lrwxrwxrwx  1 0      0          0 Apr 22 02:25 subsystem -> ../../../../b
us/platform
-rw-r--r--  1 0      0          4096 Apr 21 19:56 uevent
```

Keypad driver

```
struct input_event {  
    struct timeval time;  
    __u16 type;  
    __u16 code;  
    __s32 value;  
};
```

./include/linux/input.h에 정의

```
#define EV_KEY 0x01
```

```
#define KEY_BACK 158
```

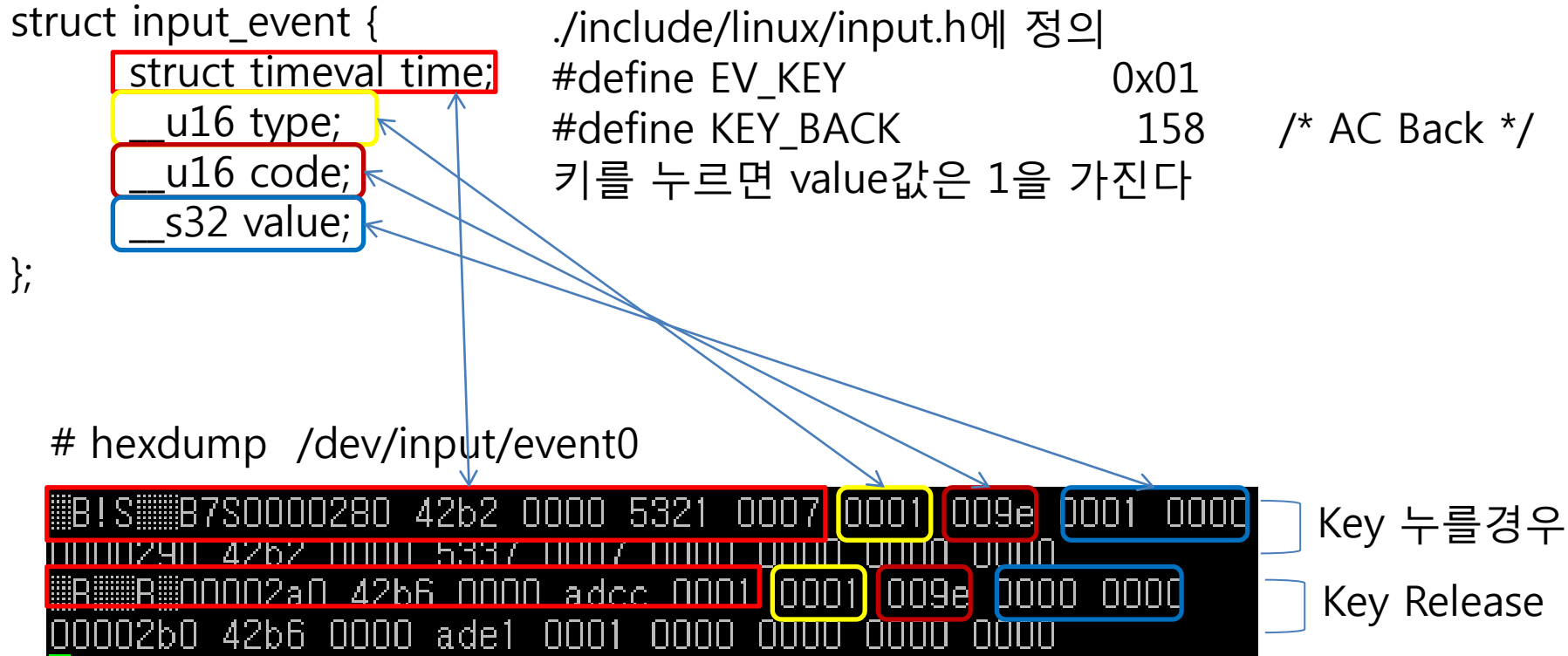
키를 누르면 value값은 1을 가진다

/* AC Back */

```
# hexdump /dev/input/event0
```

```
00000280 42b2 0000 5321 0007 0001 009e 0001 0000  
00000290 42b2 0000 5337 0007 0000 0000 0000 0000  
000002a0 42b6 0000 adcc 0001 0001 009e 0000 0000  
000002b0 42b6 0000 ade1 0001 0000 0000 0000 0000
```

Key 누를 경우
Key Release



Keypad driver

```
mango100_keypad.driver->name    = DEVICE_NAME;
mango100_keypad.driver->phys     = "mango100-keypad/input0";
mango100_keypad.driver->open     = mango100_keypad_open;
mango100_keypad.driver->close    = mango100_keypad_close;

mango100_keypad.driver->id.bustype    = BUS_HOST;
mango100_keypad.driver->id.vendor     = 0x16B4;
mango100_keypad.driver->id.product    = 0x0701;
mango100_keypad.driver->id.version    = 0x0001;
```

```
#define BUS_HOST
0x19
./include/linux/input.h
```

./drivers/input/keyboard/mango100_keypad.c 소스에 mango100_keypad_prob함수에 정의

```
# cat /proc/bus/input/devices
I: Bus=0019 Vendor=16b4 Product=0701 Version=0001
N: Name="mango100-keypad"
P: Phys=mango100-keypad/input0
S: Sysfs=/class/input/input0
U: Uniq=
H: Handlers=kbd event0
B: EV=3
B: KEY=40000800 0 0 0 0
```

Keypad driver

```
|aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa Kernel hacking aa
x Arrow keys navigate the menu. <Enter> select
x Highlighted letters are hotkeys. Pressing <
x <M> modularizes features. Press <Esc><Esc>
x for Search. Legend: [*] built-in [ ] exclu
x lqqqq^(-)aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
x x [*] Collect scheduler debugging info
x x [ ] Collect scheduler statistics
x x [*] Collect kernel timers statistics
x x [ ] Debug object operations
```

“Kernel hacking ” 설정
CONFIG_TIMER_STATS=y 로 설정

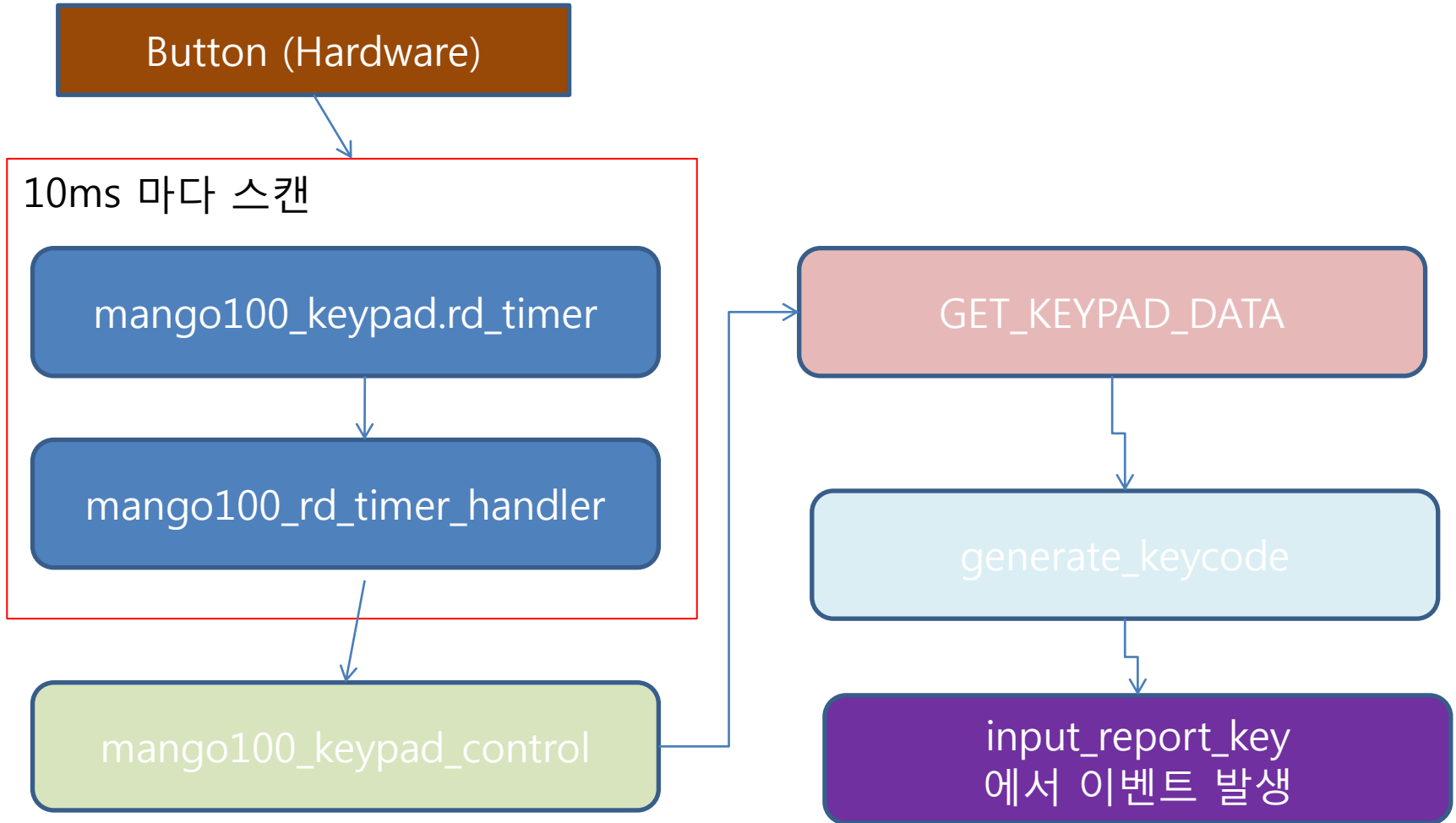
To activate the collection of stats

```
# echo 1 > /proc/timer_stats
# cat /proc/timer_stats
Timer Stats Version: v0.2
Sample period: 2.911 s
 291,   1 swapper      mango100_keypad_config (mango100_rd_timer_handler)
   2,   1 swapper      schedule_delayed_work_on (delayed_work_timer_fn)
   8, 1867 er.ServerThread futex_wait (hrtimer_wakeup)
   3,   1 swapper      phy_start_machine (phy_timer)
   3, 1832 zygote      queue_delayed_work (delayed_work_timer_fn)
   1,   0 swapper      page_writeback_init (wb_timer_fn)
308 total events, 105.805 events/sec
```

```
# echo 0 > /proc/timer_stats
```

To stop collecting stats

Keypad driver flow



안드로이드 Key event 처리

Key button Map 정의는 vendor\sec\mango100\mango100-keypad.kl frameworks\base\libs\EventHub.cpp에 scan_dir,open_device 정의

KeyInputQueue(1853): InputDeviceReader.run()

```
static const char *device_path = "/dev/input";  
bool EventHub::openPlatformInput(void)  
{  
..  
    res = scan_dir(device_path);  
..  
}
```

```
int EventHub::scan_dir(const char *dirname)  
{  
    while((de = readdir(dir))){  
        strcpy(filename, de->d_name);  
        open_device(devname);  
    }  
}
```

```
int EventHub::open_device(const char *deviceName)
```

안드로이드 Key event 처리

export ANDROID_ROOT /system
/init.rc에 정의

```
int EventHub::open_device(const char *deviceName)
{
...
const char* root = getenv("ANDROID_ROOT");
    property_get("persist.sys.keylayout", keylayout, "qwerty");
    snprintf(keylayoutFilename, sizeof(keylayoutFilename),
        "%s/usr/keylayout/%s.kl", root, keylayout);
    strcpy(devname, keylayout);
    bool defaultKeymap = access(keylayoutFilename, R_OK);
    if (defaultKeymap) {
        snprintf(keylayoutFilename, sizeof(keylayoutFilename),
            "%s/usr/keylayout/%s.kl", root, "qwerty");
        strcpy(devname, "qwerty");
    }
    LOGI("2:devname = %s, keylayout =%s, keylayoutFilename = %s",
        devname, keylayout, keylayoutFilename);
    device->layoutMap->load(keylayoutFilename);

...
}
```

```
I/EventHub( 1853): 2:devname = qwerty,
keylayout =qwerty,
keylayoutFilename =
/system/usr/keylayout/qwerty.kl
-#logcat 명령으로 디버깅 메시지 출력
```

frameworks/base/libs/ui/EventHub.cpp파일
EventHub::open_device함수에 정의

안드로이드 Key event 처리

```
int EventHub::open_device(const char *deviceName)
{
...
LOGI("New keyboard: publicID=%d device->id=0x%x devname='%s' propName='%s'
keylayout='%s'\n",
      publicID, device->id, name, propName, keylayoutFilename);
}

LOGI("New device: path=%s name=%s id=0x%x (of 0x%x) index=%d fd=%d classes=0x%x\n",
      deviceName, name, device->id, mNumDevicesById, mFDCount, fd, device->classes);

LOGV("Adding device %s %p at %d, id = %d, classes = 0x%x\n",
      deviceName, device, mFDCount, devid, device->classes);
}
```

frameworks/base/libs/ui/EventHub.cpp 에 open_device 정의

#logcat 명령으로 출력 결과

I/EventHub(1853): New keyboard: publicID=65537 device->id=0x10001

devname='mango100-keypad' propName='hw.keyboards.65537.devname' keylayout='/system/usr/keylayout/qwerty.kl'

I/EventHub(1853): New device: path=/dev/input/event0 name=mango100-keypad id=0x10001 (of 0x2) index=2 fd=58 classes=0x1

안드로이드 Key event 처리

```
Thread mThread = new Thread("InputDeviceReader") {
    public void run() {
        if (DEBUG) Log.v(TAG, "InputDeviceReader.run()");
        android.os.Process.setThreadPriority(
            android.os.Process.THREAD_PRIORITY_URGENT_DISPLAY);

        RawInputEvent ev = new RawInputEvent(),
        while (true) {
            try {
                InputDevice di;

                // block, doesn't release the monitor
                readEvent(ev);

                boolean send = false;
                boolean configChanged = false;

                if (true) {
                    Log.i(TAG, "Input event: dev=0x"
                        + Integer.toHexString(ev.deviceId)
                        + " type=0x" + Integer.toHexString(ev.type)
                        + " scancode=" + ev.scancode
                        + " keycode=" + ev.keycode
                        + " value=" + ev.value);
                }
            }
        }
    }
}
```

Key Event 처리
Thread

```
#logcat 메시지
I/KeyInputQueue( 1853): Input
event:
dev=0x10001 type=0x1
scancode=158
keycode=4 value=1
```

Mango100-keypad.kl과 scancode값
일치
BACK 키 수행이 됨
./frameworks/base/services/java/
com/android/server/KeyInputQueue
.java
파일에 정의

안드로이드 Key event 처리

frameworks/base/services/java/com/android/server/KeyInputQueue.java

```
Thread mThread = new Thread("InputDeviceReader") {  
    ....  
    while (true) {  
        try {  
            InputDevice di;  
            // block, doesn't release the monitor  
            readEvent(ev);  
            .....  
        }  
    }  
}
```

frameworks/base/services/jni/
com_android_server_KeyInputQueue.cpp

```
static JNINativeMethod gInputMethods[] = {  
    /* name, signature, funcPtr */  
    { "readEvent", "(Landroid/view/KeyEvent;)Z",  
      (void*) android_server_KeyInputQueue_readEvent },  
    ....  
};
```

```
static jboolean  
android_server_KeyInputQueue_readEvent(JNIEnv* env, jobject clazz,  
                                       jobject event)  
{  
    gLock.lock();  
    sp<EventHub> hub = gHub;  
    if (hub == NULL) {  
        hub = new EventHub;  
        gHub = hub;  
    }  
    gLock.unlock();  
  
    int32_t deviceId;  
    int32_t type;  
    int32_t scancode, keycode;  
    uint32_t flags;  
    int32_t value;  
    nsecs_t when;  
    bool res = hub->getEvent(&deviceId, &type, &scancode, &keycode,  
                           &flags, &value, &when);  
  
    env->SetIntField(event, gInputOffsets.mDeviceId, (jint)deviceId);  
    env->SetIntField(event, gInputOffsets.mType, (jint)type);  
    env->SetIntField(event, gInputOffsets.mScancode, (jint)scancode);  
    env->SetIntField(event, gInputOffsets.mKeycode, (jint)keycode);  
    env->SetIntField(event, gInputOffsets.mFlags, (jint)flags);  
    env->SetIntField(event, gInputOffsets.mValue, value);  
    env->SetLongField(event, gInputOffsets.mWhen,  
                     (jlong)(nanoseconds_to_milliseconds(when)));  
  
    return res;  
}
```

안드로이드 Key event 처리

```
public final boolean dispatch(Callback receiver, DispatcherState state,
    Object target) {
    switch (mAction) {
    ....
    case ACTION_UP:
        if (DEBUG) Log.v(TAG, "Key up to " + target + " in " + st
            + ": " + this);
        if (state != null) {
            state.handleUpEvent(this);
        }
        return receiver.onKeyUp(mKeyCode, this);
```

```
public void handleUpEvent(KeyEvent event) {
    final int keyCode = event.getKeyCode();
    if (DEBUG) Log.v(TAG, "Handle key up " + event + ": " +
```

./frameworks/base/core/java/android/view/KeyEvent.java
에서 dispatch를 한다.

```
#logcat 수행 결과
KeyEvent( 1910): Key up to android.inputmethodservice.InputMethodService$InputMethodSessionImpl@43da0b40
in
android.view.KeyEvent$DispatcherState@43d92cf8: KeyEvent{action=1 code=4 repeat=0 meta=0 scancode=158
mFlags=8}
V/KeyEvent( 1910): Handle key up KeyEvent{action=1 code=4 repeat=0 meta=0 scancode=158 mFlags=8}: a
ndroid.view.KeyEvent$DispatcherState@43d92cf8
D/MountListener( 1853): handleEvent ums_connected
```

안드로이드 Key Event 처리

```
Window Manager  
distpatchKey(KeyEvent..)
```

```
frameworks/base/services/java/com/android/server/WindowManagerService.java
```

```
Thread mThread = new Thread("InputDeviceReader")  
readEvent(ev);  
di = newInputDevice(ev.deviceId);
```

```
frameworks/base/services/java/com/android/server/KeyInputQueue.java
```

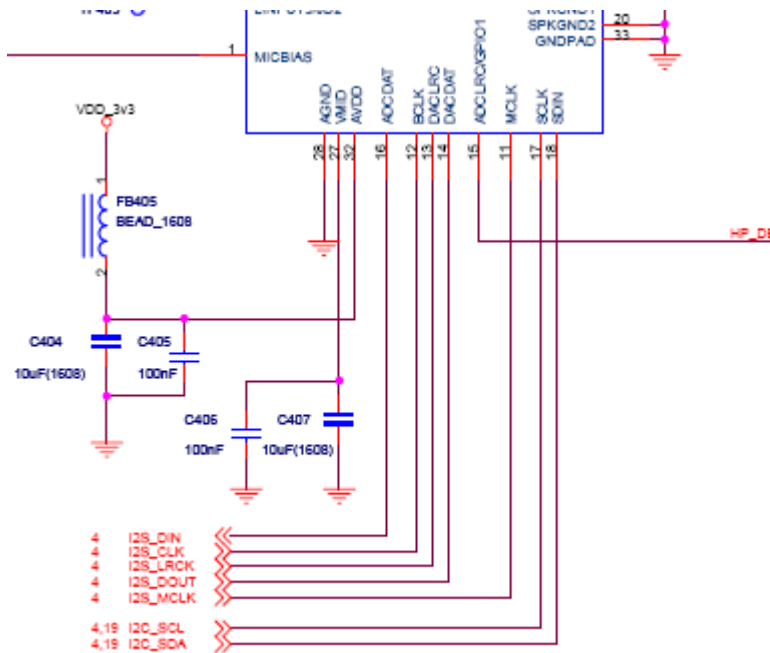
```
static JNINativeMethod gInputMethods[] = {  
    /* name, signature, funcPtr */  
    { "readEvent",      "(Landroid/view/RawInputEvent;)Z",  
      (void*) android_server_KeyInputQueue_readEvent },
```

```
frameworks/base/services/jni/com_android_server_KeyInputQueue.cpp
```

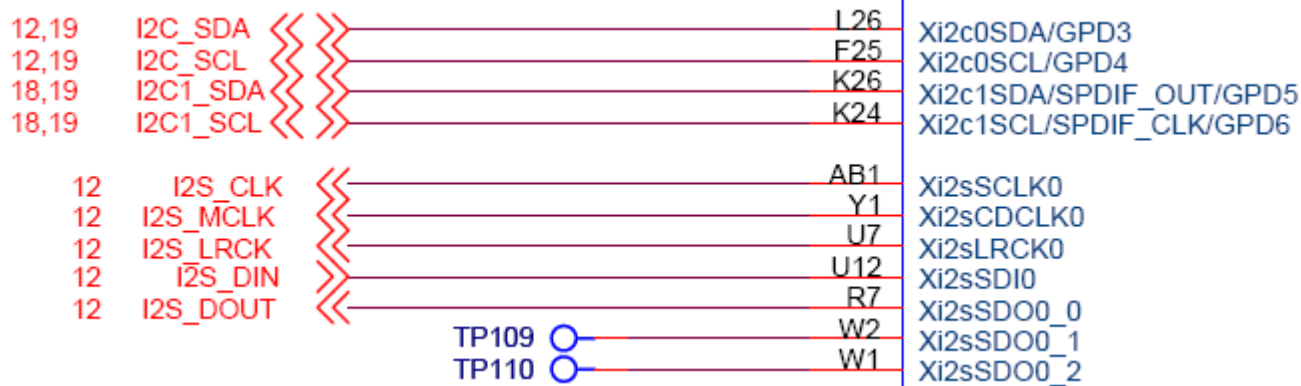
```
static const char *device_path = "/dev/input";  
bool EventHub::getEvent(int32_t* outDeviceId, int32_t* outType,  
bool EventHub::openPlatformInput(void)
```

```
frameworks/base/libs/ui/EventHub.cpp
```

codec driver(wm8960)



WM8960 디바이스를
I2C 0번 채널로 디바이스를 컨트롤하고
Data는 I2S0번 채널로 전송토록 설계



codec driver(wm8960)

```
Device Drivers
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for
Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module <> module capable

Generic Driver Options --->
<> Connector - unified userspace <> kernel-space linker --->
<> Memory Technology Device (MTD) support --->
<> Parallel port support --->
[*] Block devices --->
[*] Misc devices --->
<> ATA/ATAPI/MFM/RLL support --->
SCSI device support --->
<> Serial ATA (prod) and Parallel ATA (experimental) drivers --->
[ ] Multiple devices driver support (RAID and LVM) --->
[*] Network device support --->
[ ] ISDN support --->
Input device support --->
Character devices --->
[*] I2C support --->
[ ] SPI support --->
--> GPIO Support --->
<> Dallas's 1-wire support --->
<*> Power supply class support --->
<> Hardware Monitoring support --->
<> Generic Thermal sysfs driver --->
v(1)
<Select> <Exit> <Help>
```

if

```
config SND_SOC_ALL_CODECS
tristate "Build all ASoC CODEC drivers"
select SND_SOC_AC97_CODEC if SND_SOC_AC97_BUS
select SND_SOC_AD1980 if SND_SOC_AC97_BUS
select SND_SOC_AD73311 if I2C
select SND_SOC_AK4535 if I2C
select SND_SOC_CS4270 if I2C
select SND_SOC_PCM3008
select SND_SOC_SSM2602 if I2C
select SND_SOC_TLV320AIC23 if I2C
select SND_SOC_TLV320AIC26 if SPI_MASTER
select SND_SOC_TLV320AIC3X if I2C
select SND_SOC_TWL4030 if TWL4030_CORE
select SND_SOC_UDA134X
select SND_SOC_UDA1380 if I2C
select SND_SOC_WM8350 if MFD_WM8350
select SND_SOC_WM8510 if SND_SOC_I2C_AND_SPI
select SND_SOC_WM8580 if I2C
select SND_SOC_WM8728 if SND_SOC_I2C_AND_SPI
select SND_SOC_WM8731 if SND_SOC_I2C_AND_SPI
select SND_SOC_WM8750 if SND_SOC_I2C_AND_SPI
select SND_SOC_WM8753 if SND_SOC_I2C_AND_SPI
select SND_SOC_WM8900 if I2C
select SND_SOC_WM8903 if I2C
select SND_SOC_WM8960 if I2C
```

CONFIG_I2C = y 로 설정이 되어 있어야 SND_SOC_WM8960 이 활성화 됨
./ sound/soc/codecs/Kconfig 에 추가

codec driver(wm8960)

```
snd-soc-wm8900-objs := wm8900.o
snd-soc-wm8903-objs := wm8903.o
snd-soc-wm8960-objs := wm8960.o
snd-soc-wm8971-objs := wm8971.o
snd-soc-wm8990-objs := wm8990.o
snd-soc-wm9712-objs := wm9712.o
snd-soc-wm9713-objs := wm9713.o

obj-$(CONFIG_SND_SOC_AC97_CODEC) += snd-soc-ac
obj-$(CONFIG_SND_SOC_AD1980) += snd-soc-ad1980.o
obj-$(CONFIG_SND_SOC_AD73311) += snd-soc-ad73311.o
obj-$(CONFIG_SND_SOC_AK4535) += snd-soc-ak4535.o
obj-$(CONFIG_SND_SOC_CS4270) += snd-soc-cs4270.o
"sound/soc/codecs/Makefile" 57L, 2335C
obj-$(CONFIG_SND_SOC_L3) += snd-soc-l3.o
obj-$(CONFIG_SND_SOC_PCM3008) += snd-soc-pcm3008.o
obj-$(CONFIG_SND_SOC_SSM2602) += snd-soc-ssm2602.o
obj-$(CONFIG_SND_SOC_TLV320A1C23) += snd-soc-tl
obj-$(CONFIG_SND_SOC_TLV320A1C26) += snd-soc-tl
obj-$(CONFIG_SND_SOC_TLV320A1C3X) += snd-soc-tl
obj-$(CONFIG_SND_SOC_TWL4030) += snd-soc-twl4030.o
obj-$(CONFIG_SND_SOC_UDA134X) += snd-soc-uda134x.o
obj-$(CONFIG_SND_SOC_UDA1380) += snd-soc-uda1380.o
obj-$(CONFIG_SND_SOC_WM8350) += snd-soc-wm8350.o
obj-$(CONFIG_SND_SOC_WM8510) += snd-soc-wm8510.o
obj-$(CONFIG_SND_SOC_WM8580) += snd-soc-wm8580.o
obj-$(CONFIG_SND_SOC_WM8728) += snd-soc-wm8728.o
obj-$(CONFIG_SND_SOC_WM8731) += snd-soc-wm8731.o
obj-$(CONFIG_SND_SOC_WM8750) += snd-soc-wm8750.o
obj-$(CONFIG_SND_SOC_WM8753) += snd-soc-wm8753.o
obj-$(CONFIG_SND_SOC_WM8900) += snd-soc-wm8900.o
obj-$(CONFIG_SND_SOC_WM8903) += snd-soc-wm8903.o
obj-$(CONFIG_SND_SOC_WM8960) += snd-soc-wm8960.o
obj-$(CONFIG_SND_SOC_WM8971) += snd-soc-wm8971.o
```

snd-soc-wm8960.o==wm8960.o 동일
이유는 사운드 드라이버 소스가 2개이상 인
경우를 대비

./ sound/soc/codecs/Makefile 에 추가
Wm8960.c 드라이버 파일을
./sound/soc/codecs/에 추가

codec driver(wm8960)

```
config SND_S5P_MANG0100
    tristate "Soc Audio support Mango100"
    depends on SND_SAMSUNG_SOC && (MACH_MANG0100)
    select SND_S3C24XX_SOC
    help
        Say Y if you want to add support for SoC audio on the MANG0100.

choice
    prompt "Select MAnGo100 Audio Port Type"
    depends on SND_S5P_MANG0100

config SND_MANG0100_WM8960
    bool "WM8960 Driver"
    select SND_SOC_WM8960
    select SND_S5P_SOC_I2S
    select SND_S3C_I2SV2_SOC

config SND_MANG0100_HDMI_SPDIF
    bool "HDMI SPDIF Driver"
    select SND_S5P_SPDIF
```

sound/soc/s3c24xx/Kconfig 파일에 위의 내용 추가

codec driver(wm8960)

```

aaaaaaaaaaaaaaaaaaaaaaaaaaaa ALSA for SoC audio support aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < > module
l
--- ALSA for SoC audio support
x <+> Samsung SoC Audio Drivers
x <+> Soc Audio support Mango100
x Select MAngo100 Audio Port Type (WM8960 Driver) --->
x < > SoC AC97 Audio support for LN2440SBC - ALC650
x < > SoC I2S Audio support UDA134X wired to a S3C24XX
x < > HDMI-SPDIF on SMDKS5P
x < > Build all ASoC CODEC drivers
x
x
x
x
x
x
x
x
x
x
x
x
x
x
x
x
x
x
x
x
x
x
m
<Select> < Exit > < Help >

```

#make menuconfig 명령으로 확인 할 있음

codec driver(wm8960)

```
obj-$(CONFIG_SND_S3C24XX_SOC) += snd-soc-s3c24xx.o
obj-$(CONFIG_SND_LPAM_SOC) += snd-soc-lpam.o
obj-$(CONFIG_SND_S3C24XX_SOC_I2S) += snd-soc-s3c24xx-i2s.o
obj-$(CONFIG_SND_S3C2443_SOC_AC97) += snd-soc-s3c2443-ac97.o
obj-$(CONFIG_SND_S3C2412_SOC_I2S) += snd-soc-s3c2412-i2s.o
obj-$(CONFIG_SND_S5P_SOC_AC97) += snd-soc-s5p-ac97.o
obj-$(CONFIG_SND_S5P_SOC_I2S) += snd-soc-s5p-i2s.o
obj-$(CONFIG_SND_S5P_SOC_I2S_LP) += snd-soc-s5p-i2s-lp.o
obj-$(CONFIG_SND_S3C_I2SV2_SOC) += snd-soc-s3c-i2s-v2.o
obj-$(CONFIG_SND_S3C_SOC_PCM) += snd-soc-s3c-pcm.o
obj-$(CONFIG_SND_S5P_SPDIF) += snd-soc-s5p-spdif.o

# S3C24XX Machine Support
snd-soc-neo1973-wm8753-objs := neo1973_wm8753.o
snd-soc-smdk2443-wm9710-objs := smdk2443_wm9710.o
snd-soc-ln2440sbc-alc650-objs := ln2440sbc_alc650.o
snd-soc-s3c24xx-uda134x-objs := s3c24xx_uda134x.o
snd-soc-smdks5p-wm9713-objs := smdks5p_wm9713.o
snd-soc-smdks5p-wm8580-objs := smdks5p_wm8580.o
snd-soc-smdks5plp-wm8580-objs := smdks5plp_wm8580.o
snd-soc-mango100-wm8960-objs := mango100_wm8960.o
snd-soc-universal-ak4671-objs := universal_ak4671.o
snd-soc-smdks5p-spdif-objs := smdks5p_hdmi_spdif.o
snd-soc-mango100-spdif-objs := mango100_hdmi_spdif.o

obj-$(CONFIG_SND_S3C24XX_SOC_NEO1973_WM8753) += snd-soc-neo1973-wm8753.o
obj-$(CONFIG_SND_S3C24XX_SOC_SMDK2443_WM9710) += snd-soc-smdk2443-wm9710.o
obj-$(CONFIG_SND_S3C24XX_SOC_LN2440SBC_ALC650) += snd-soc-ln2440sbc-alc650.o
obj-$(CONFIG_SND_S3C24XX_SOC_S3C24XX_UDA134X) += snd-soc-s3c24xx-uda134x.o
obj-$(CONFIG_SND_S5P_SOC_SMDK_WM9713) += snd-soc-smdks5p-wm9713.o
obj-$(CONFIG_SND_S5P_SOC_WM8580) += snd-soc-smdks5p-wm8580.o
obj-$(CONFIG_SND_S5P_SOC_WM8580_LP) += snd-soc-smdks5plp-wm8580.o
obj-$(CONFIG_SND_S5P_SOC_UNIVERSAL_AK4671) += snd-soc-universal-ak4671.o
obj-$(CONFIG_SND_SMDKS5P_HDMI_SPDIF) += snd-soc-smdks5p-spdif.o
obj-$(CONFIG_SND_MANGO100_WM8960) += snd-soc-mango100-wm8960.o
obj-$(CONFIG_SND_MANGO100_HDMI_SPDIF) += snd-soc-mango100-spdif.o
```

sound/soc/s3c24xx/Makefile을 위와 같이 수정
sound/soc/s3c24xx/mango100_hdmi_spdif.c
sound/soc/s3c24xx/mango100_wm8960.c
을 sound/soc/s3c24xx 디렉토리에 파일을 만들어서 추가

codec driver(wm8960)

```
static struct snd_soc_dai_link smdk_dai[] = {
{
    .name = "WM8960 I2S",
    .stream_name = "Tx/Rx",
    .cpu_dai = &s5p_i2s_dai[0],
    .codec_dai = &wm8960_dai,
    .init = smdk_wm8960_init,
    .ops = &smdk_i2s_ops,
},
};

static struct snd_soc_card smdk = {
    .name = "smdk",
    .platform = &s3c24xx_soc_platform,
    .dai_link = smdk_dai,
    .num_links = ARRAY_SIZE(smdk_dai),
};

static struct wm8960_setup_data smdk_wm8960_setup = {
    .i2c_bus = 0,
    .i2c_address = 0x1a,
};

static struct snd_soc_device smdk_snd_devdata = {
    .card = &smdk,
    .codec_dev = &soc_codec_dev_wm8960,
    .codec_data = &smdk_wm8960_setup,
};
```

```
# cat pcm
00-00: Tx/Rx WM8960-0 : : playback 1 : capture 1
# pwd
/proc/asound
```

```
drwxr-xr-x  3 0      0      0 Apr 22 17:52 a
drwxr-xr-x  3 0      0      0 Apr 22 17:52 c
lrwxrwxrwx  1 0      0      0 Apr 22 17:54 d
audio
drwxr-xr-x  3 0      0      0 Apr 22 17:52 d
-rw-r--r--  1 0      0      0 Apr 22 17:54 i
drwxr-xr-x  3 0      0      0 Apr 22 17:52 m
-r--r--r--  1 0      0      0 Apr 22 17:54 n
drwxr-xr-x  3 0      0      0 Apr 22 17:52 p
drwxr-xr-x  3 0      0      0 Apr 22 17:52 p
drwxr-xr-x  2 0      0      0 Apr 22 17:52 p
lrwxrwxrwx  1 0      0      0 Apr 22 17:54 s
./../class/sound
-rw-r--r--  1 0      0      0 Apr 22 17:52 u
# pwd
/sys/devices/platform/soc-audio/driver/soc-audio/sound/car
```

```
/sys/bus/i2c/devices/0-001a/driver
# ls -al
drwxr-xr-x  2 0      0      0 Apr 22 17:38 .
drwxr-xr-x  7 0      0      0 Apr 22 17:37 ..
lrwxrwxrwx  1 0      0      0 Apr 22 17:39 0-001a ->
evices/platform/s3c2410-i2c.0/i2c-adapter/i2c-0/0-001a
--w-----  1 0      0      0 Apr 22 17:39 bind
--w-----  1 0      0      0 Apr 22 17:39 uevent
--w-----  1 0      0      0 Apr 22 17:39 unbind
```

codec driver(wm8960)

```
static const struct i2c_device_id wm8960_i2c_id[] = {
    { "wm8960", 0 },
    { }
};
MODULE_DEVICE_TABLE(i2c, wm8960_i2c_id);

static struct i2c_driver wm8960_i2c_driver = {
    .driver = {
        .name = "WM8960 I2C Codec",
        .owner = THIS_MODULE,
    },
    .probe =    wm8960_i2c_probe,
    .remove =   __devexit_p(wm8960_i2c_remove),
    .id_table = wm8960_i2c_id,
};
```

Sound/soc/codecs/wm8960.c 파일 참조

```
drwxr-xr-x  3 0      0      0 Apr 22 18:26 .
drwxr-xr-x  5 0      0      0 Apr 22 18:26 ..
lrwxrwxrwx  1 0      0      0 Apr 22 18:39 bus -> ../../../../bus/
i2c
lrwxrwxrwx  1 0      0      0 Apr 22 18:39 driver -> ../../../../b
us/i2c/drivers/WM8960 I2C Codec
-r--r--r--  1 0      0      0 Apr 22 18:39 modalias
-r--r--r--  1 0      0      0 Apr 22 18:39 name
drwxr-xr-x  2 0      0      0 Apr 22 18:26 power
lrwxrwxrwx  1 0      0      0 Apr 22 18:39 subsystem -> ../../../../
./bus/i2c
-rw-r--r--  1 0      0      0 Apr 22 18:26 uevent
# pwd
/sys/class/i2c-adapter/i2c-0/0-001a/driver/0-001a
```

codec driver(wm8960)

The WM8960 is controlled by writing to registers through a 2-wire serial control interface. A control word consists of 16 bits. The first 7 bits (B15 to B9) are address bits that select which control register is accessed. The remaining 9 bits (B8 to B0) are data bits, corresponding to the 9 bits in each control register. Many devices can be controlled by the same bus, and each device has a unique 7-bit address (this is not the same as the 7-bit address of each register in the WM8960).

The device address is 0011010 (0x34h).

```
static struct wm8960_setup_data smdk_wm8960_setup = {
    .i2c_bus = 0,
    .i2c_address = 0x1a,
};
```

sound/soc/s3c24xx/mango100_wm8960.c에 정의

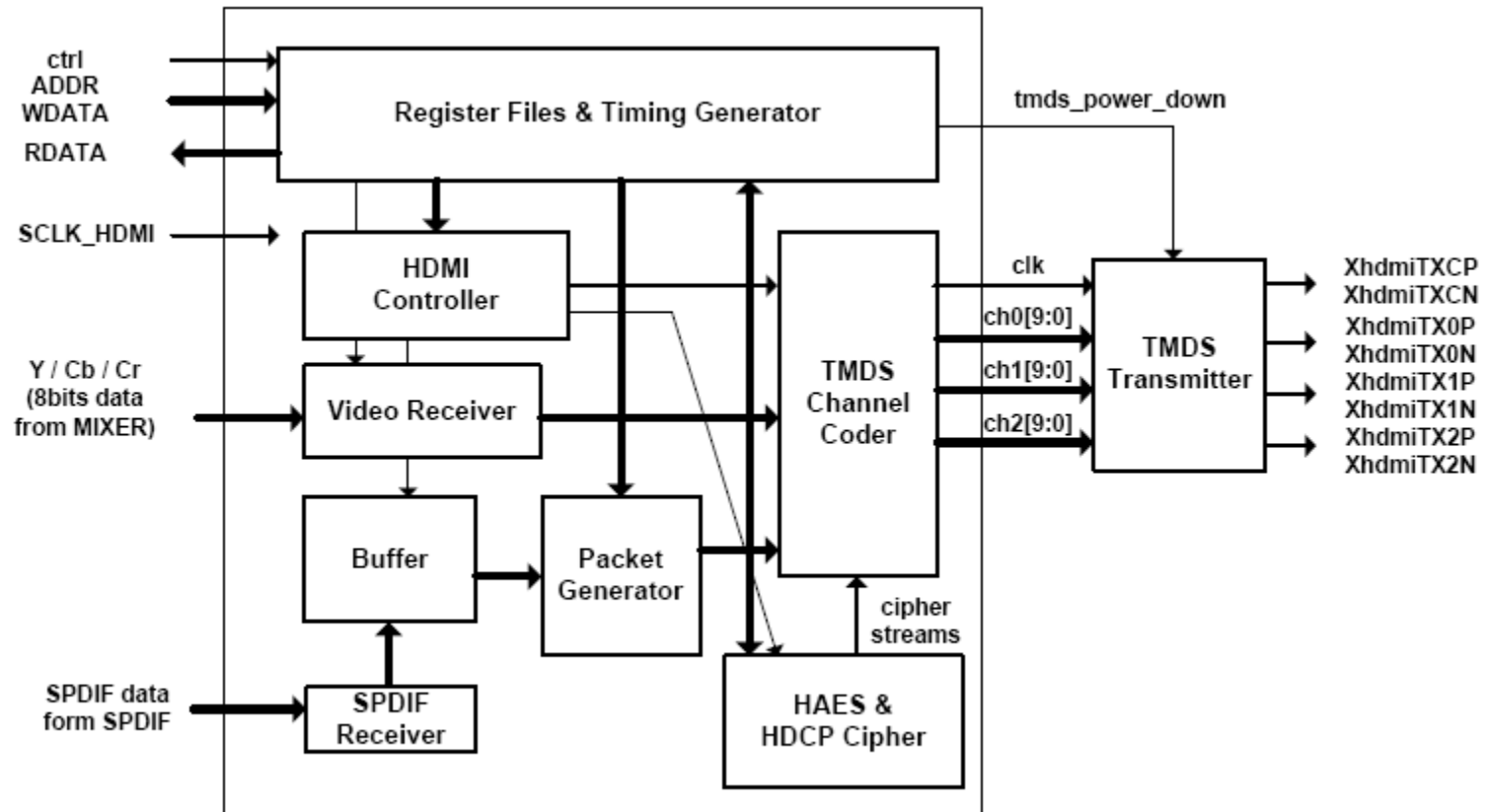
```
static struct i2c_board_info i2c_devs0[] __initdata = {
    { I2C_BOARD_INFO("wm8960", 0x1a), }, // WM8960
};
```

../arch/arm/mach-s5pc100/mach-mango100.c 에 정의

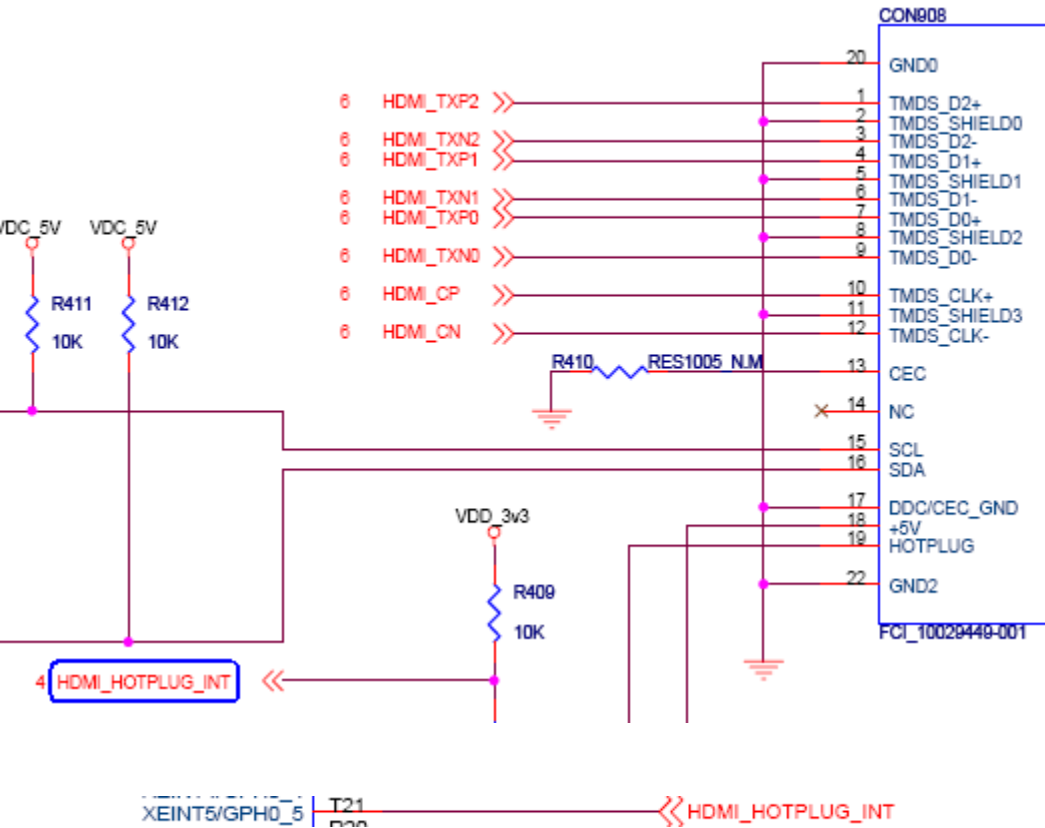
```
static void s3c24xx_i2c_message_start(struct s3c24xx_i2c *i2c,
                                     struct i2c_msg *msg)
{
    unsigned int addr = (msg->addr & 0x7f) << 1;
```

./drivers/i2c/busses/i2c-s3c2410.c 에서 0x1a<<1 로 shift

HDMI Driver



HDMI Driver



```
static struct resource s5p_tvout_resources[] = {
..
[8] = {
    .start = IRQ_EINT5,
    .end   = IRQ_EINT5,
    .flags = IORESOURCE_IRQ
}
./arch/arm/plat-s5pc1xx/devs.c에 수정,
드라이버 소스는
drivers/media/video/samsung/tv20/s5pc100/hdmi_s5pc100.c
```


HDMI S/PDIF Driver

S/PDIF의 약자는 Sony/Philips Digital InterFace의 약자입니다.
소니와 필립스사에서 디지털 오디오 전송을 위해서 만든 표준 인터페이스로써,
신호선 1개와 그라운드선 1개 이렇게 2가닥을 사용하는 방식입니다.

기존의 디지털 오디오 신호의 경우 동기(싱크신호)신호를 넣어야 했고
또 신호의 간섭이 심하여 일정 이상의 거리를 전송하기 어려웠던 반면에
S/PDIF는 약 20 미터 정도까지 일정하게 데이터를 전달할 수 있어
CDP, 사운드 카드, CD-ROM drive, A/V 리시버에 다양하게 사용되고 있습니다.

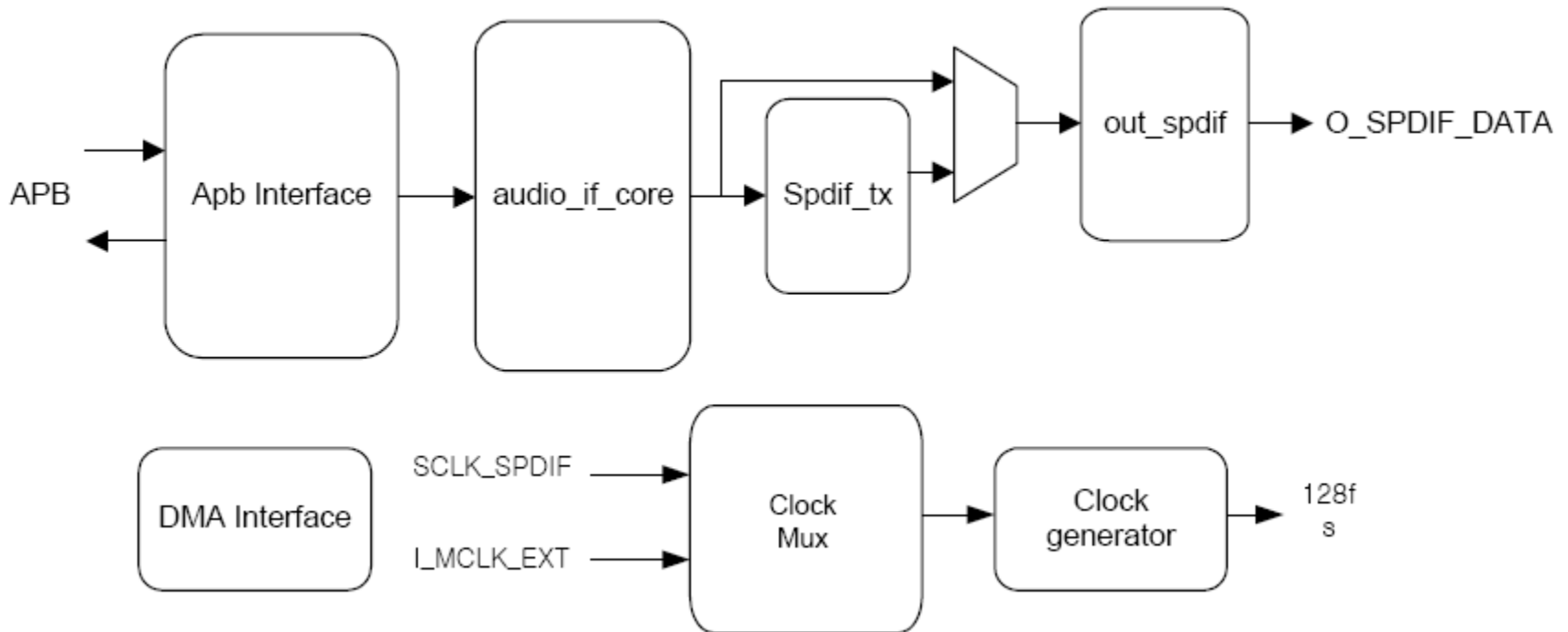
사람이 들을 수 있는 신호는 모두 아날로그이며, 컴퓨터는 궁극적으로
디지털 신호만을 입출력합니다. 컴퓨터의 경우도 아날로그 출력은
디지털 신호를 아날로그로 컨버트(Digital to Analog Converter)하여
출력하는 것이고, 반대로 아날로그 신호를 저장하려면
ADC(Analog to Digital Converter)를 사용해야 합니다.

예를 들면 컴퓨터의 MP3 데이터(디지털)을 MD(디지털)로 저장하려면
디지털 신호를 사용하면 디지털 -> 디지털로 전달하면 되지만
아날로그 신호의 경우는 사운드 카드에서
디지털 -> DAC -> ADC -> 디지털로 받게 되어 음질이 많이 손실됩니다.
때문에 디지털 인터페이스를 사용하는 것입니다.

일반적으로 디지털 기기간 원본의 손실없이 전달하기 위해
아날로그 신호전송 보다 디지털 전송을 선호하고 있습니다.
참고로 S/PDIF의 케이블간의 저항의 권장치는 75오옴입니다.

HDMI S/PDIF Driver

S5pc100 cpu가 기본 제공



HDMI S/PDIF Driver

```
config SND_S5P_MANG0100
    tristate "Soc Audio support Mango100"
    depends on SND_SAMSUNG_SOC && (MACH_MANG0100)
    select SND_S3C24XX_SOC
    help
        Say Y if you want to add support for SoC audio on the MANG0100.

choice
    prompt "Select MAnGo100 Audio Port Type"
    depends on SND_S5P_MANG0100

config SND_MANG0100_WM8960
    bool "WM8960 Driver"
    select SND_SOC_WM8960
    select SND_S5P_SOC_I2S
    select SND_S3C_I2SV2_SOC

config SND_MANG0100_HDMI_SPDIF
    bool "HDMI SPDIF Driver"
    select SND_S5P_SPDIF
```

sound/soc/s3c24xx/Kconfig 파일에 위의 내용 추가

HDMI S/PDIF Driver

```
lcccccccccccccccccccc Select MAngo100 Audio Port Type cccccccccccccccccck
x Use the arrow keys to navigate this window or press the hotkey of x
x the item you wish to select followed by the <SPACE BAR>. Press x
x <?> for additional information about this option. x
x lccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccck x
x x ( ) ||M8960 Driver x x
x x (X) HDMI SPDIF Driver x x
x x x x x x
x x x x x x
x x x x x x
x x x x x x
x x x x x x
x ||ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccj x
tccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccL
x <Select> < Help > x
||ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccj
```

"Device driver -> Sound card support->
Advanced Linux Sound Architecture-> ALSA for Soc audio support
->Soc Audio support Mango100->
Select Mango100 Audio Port Type에서 HDMI SPDIF Driver선택

HDMI S/PDIF Driver

ALSA device list:

#0: smdks5p (HDMI-SPDIF)

커널 로그에서 위와 같이 메시지가 출력

```
static struct snd_soc_dai_link smdks5p_dai[] = {
{
.name = "HDMI-SPDIF",
.stream_name = "HDMI-SPDIF Playback",
.cpu_dai = &s5p_spdif_dai,
.codec_dai = &s5p_hdmi_spdif_dai[0],
.init = smdks5p_spdif_init,
.ops = &smdks5p_spdif_ops,
},
};
```

```
static struct snd_soc_card smdks5p = {
.name = "smdks5p",
.platform = &s3c24xx_soc_platform,
.dai_link = smdks5p_dai,
.num_links = ARRAY_SIZE(smdks5p_dai),
};
```

```
# ls -al
dr-xr-xr-x  4 0          0          0 Apr 22 21:14 .
dr-xr-xr-x 63 0          0          0 Jan  1  1970 ..
dr-xr-xr-x  4 0          0          0 Apr 22 21:14 card0
-r--r--r--  1 0          0          0 Apr 22 21:14 cards
-r--r--r--  1 0          0          0 Apr 22 21:14 devices
dr-xr-xr-x  2 0          0          0 Apr 22 21:14 oss
-r--r--r--  1 0          0          0 Apr 22 21:14 pcm
lrwxrwxrwx  1 0          0          0 Apr 22 21:14 smdks5p -> card0
-r--r--r--  1 0          0          0 Apr 22 21:14 timers
-r--r--r--  1 0          0          0 Apr 22 21:14 version
# cat cards
0 [smdks5p          ]: HDMI-SPDIF - smdks5p
smdks5p (HDMI-SPDIF)
```

```
# pwd
/proc/asound
# cat devices
2:          : timer
3: [ 0- 0]: digital audio playback
4: [ 0- 0]: digital audio capture
5: [ 0]   : control
# cat pcm
00-00: HDMI-SPDIF Playback HDMI-SPDIF Codec-0 : : playback 1 : capture 1
```

```
struct snd_soc_dai s5p_hdmi_spdif_dai[] = {
{
.name = "HDMI-SPDIF Codec",
.id = 0,
.playback = {
.stream_name = "Playback",
.channels_min = 1,
.channels_max = 2,
.rates = S5P_HDMI_SPDIF_RATES,
.formats = S5P_HDMI_SPDIF_FORMATS,
},
.capture = {
.stream_name = "Capture",
.channels_min = 2,
.channels_max = 2,
.rates = S5P_HDMI_SPDIF_RATES,
.formats = S5P_HDMI_SPDIF_FORMATS,
},
.ops = {
.hw_params = NULL,
.set_fmt = NULL,
.set_clkdiv = NULL,
.set_pll = NULL,
},
},
};
```

Battery Driver

```
config BATTERY_MANGO_DUMMY
    tristate "Mango Dummy Battery"
    help
        Mango Dummy Battery Driver
```

```
----- Power supply class support -----
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for
Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

--- Power supply class support
[ ] Power supply debug
< > Generic PDA/phone power driver
< > Battery driver (S3C)
< > APM emulation for class batteries
< > DS2760 battery driver (HP IPAQ & others)
< > BQ27200 battery driver
< > Maxim MAX17040 Fuel Gauge
[*] Mango Dummy Battery
```

drivers/power/Kconfig 에 dummy battery 를 추가

Battery Driver

```
ifeq ($(CONFIG_SYSFS),y)
power_supply-objs += power_supply_sysfs.o
endif

ifeq ($(CONFIG_LEDS_TRIGGERS),y)
power_supply-objs += power_supply_leds.o
endif

ifeq ($(CONFIG_POWER_SUPPLY_DEBUG),y)
EXTRA_CFLAGS += -DDEBUG
endif

obj-$(CONFIG_POWER_SUPPLY) += power_supply.o

obj-$(CONFIG_PDA_POWER) += pda_power.o
obj-$(CONFIG_BATTERY_S3C) += s3c_fake_battery.o
obj-$(CONFIG_APM_POWER) += apm_power.o
obj-$(CONFIG_WM8350_POWER) += wm8350_power.o

obj-$(CONFIG_BATTERY_DS2760) += ds2760_battery.o
obj-$(CONFIG_BATTERY_PMU) += pmu_battery.o
obj-$(CONFIG_BATTERY_OLPC) += olpc_battery.o
obj-$(CONFIG_BATTERY_TOSA) += tosa_battery.o
obj-$(CONFIG_BATTERY_WM97XX) += wm97xx_battery.o
obj-$(CONFIG_BATTERY_BQ27X00) += bq27x00_battery.o
obj-$(CONFIG_BATTERY_DA9030) += da9030_battery.o
obj-$(CONFIG_CHARGER_PCF50633) += pcf50633-charger.o
obj-$(CONFIG_BATTERY_MAX17040) += max17040_battery.o
obj-$(CONFIG_BATTERY_MANGO_DUMMY) += mango_dummy_battery.o
```

drivers/power/Makefile 에 추가

Battery Driver

```
static struct platform_driver dummy_battery_device = {
    .probe           = dummy_battery_probe,
    .remove          = dummy_battery_remove,
    .driver = {
        .name = "dummy-battery"
    }
};
```

```
drwxr-xr-x  4 0      0      0 Apr 22 21:11 .
drwxr-xr-x 39 0      0      0 Apr 22 21:11 ..
lrwxrwxrwx  1 0      0      0 Apr 23 01:36 driver -> ../../../../bus/platf
orm/drivers/dummy-battery
-r--r--r--  1 0      0      0 Apr 23 01:36 modalias
drwxr-xr-x  2 0      0      0 Apr 22 21:11 power
drwxr-xr-x  5 0      0      0 Apr 22 21:11 power_supply
lrwxrwxrwx  1 0      0      0 Apr 23 01:36 subsystem -> ../../../../bus/pl
atform
-rw-r--r--  1 0      0      0 Apr 22 21:11 uevent
# cat modalias
platform:dummy-battery
# ls power
wakeup
# ls power_supply/
ac      battery  usb
```

drivers/power/mango_dummy_battery.c 드라이버 소스 생성 추가

Battery Driver

```
#define POWER_SUPPLY_PATH "/sys/class/power_supply"
```

실행 결과

```
# ls /sys/class/power_supply/  
ac      battery  usb
```

Éclair 소스에서 framework/base/services/jni/com_android_server_BatteryService.cpp
에 POWER_SUPPLY_PATH 정의

```
struct dummy_battery_data {  
    struct power_supply  battery;  
    struct power_supply  ac;  
    struct power_supply  usb;  
  
#if defined(CONFIG_HAS_WAKELOCK)  
    int                  locked;  
#endif  
  
    int                 usb_online;  
    int                 ac_online;  
} *dummy_data;
```

커널 .drivers/power/mango_dummy_battery.c

```
struct Fields {  
    // members  
    jfieldID mAcOnline;  
    jfieldID mUsbOnline;  
    jfieldID mBatteryStatus;  
    jfieldID mBatteryHealth;  
    jfieldID mBatteryPresent;  
    jfieldID mBatteryLevel;  
    jfieldID mBatteryVoltage;  
    jfieldID mBatteryTemperature;  
    jfieldID mBatteryTechnology;  
};
```

Éclair 소스에서 framework/base/services/jni/com_android_server_BatteryService.cpp

Battery Driver

```
int register_android_server_BatteryService(JNIEnv* env)
{
    DIR* dir = opendir(POWER_SUPPLY_PATH);
    while ((entry = readdir(dir)) {
        const char* name = entry->d_name;

        char buf[20];
        // Look for "type" file in each subdirectory
        snprintf(path, sizeof(path), "%s/%s/type", POWER_SUPPLY_PATH, name);
        int length = readFromFile(path, buf, sizeof(buf));
        if (length > 0) {
            if (strcmp(buf, "Mains") == 0) {
                snprintf(path, sizeof(path), "%s/%s/online", POWER_SUPPLY_PATH, name);
                if (access(path, R_OK) == 0)
                    gPaths.acOnlinePath = strdup(path);
                # cat /sys/class/power_supply/Mains/online
                1
            }

            else if (strcmp(buf, "USB") == 0) {
                snprintf(path, sizeof(path), "%s/%s/online", POWER_SUPPLY_PATH, name);
                if (access(path, R_OK) == 0)
                    gPaths.usbOnlinePath = strdup(path);
                # cat /sys/class/power_supply/usb/online
                0
            }

            else if (strcmp(buf, "Battery") == 0) {
                snprintf(path, sizeof(path), "%s/%s/status", POWER_SUPPLY_PATH, name);
                if (access(path, R_OK) == 0)
                    gPaths.batteryStatusPath = strdup(path);
                # cat /sys/class/power_supply/battery/status
                Charging
            }
        }
    }
}
```

Battery Driver

```
    snprintf(path, sizeof(path), "%s/%s/health", POWER_SUPPLY_PATH, name);
if (access(path, R_OK) == 0)
    gPaths.batteryHealthPath = strdup(path);
snprintf(path, sizeof(path), "%s/%s/present", POWER_SUPPLY_PATH, name);
if (access(path, R_OK) == 0)
    gPaths.batteryPresentPath = strdup(path);
snprintf(path, sizeof(path), "%s/%s/capacity", POWER_SUPPLY_PATH, name);
if (access(path, R_OK) == 0)
    gPaths.batteryCapacityPath = strdup(path);

snprintf(path, sizeof(path), "%s/%s/voltage_now", POWER_SUPPLY_PATH, name);
if (access(path, R_OK) == 0) {
    gPaths.batteryVoltagePath = strdup(path);
    // voltage_now is in microvolts, not millivolts
    gVoltageDivisor = 1000;
} else {
    snprintf(path, sizeof(path), "%s/%s/batt_vol", POWER_SUPPLY_PATH, name);
    if (access(path, R_OK) == 0)
        gPaths.batteryVoltagePath = strdup(path);
}

snprintf(path, sizeof(path), "%s/%s/temp", POWER_SUPPLY_PATH, name);
if (access(path, R_OK) == 0) {
    gPaths.batteryTemperaturePath = strdup(path);
} else {
    snprintf(path, sizeof(path), "%s/%s/batt_temp", POWER_SUPPLY_PATH, name);
    if (access(path, R_OK) == 0)
        gPaths.batteryTemperaturePath = strdup(path);
}
```

```
/sys/class/power_supply/battery/capacity
/sys/class/power_supply/battery/device/
/sys/class/power_supply/battery/health
/sys/class/power_supply/battery/power/
/sys/class/power_supply/battery/present
/sys/class/power_supply/battery/status
/sys/class/power_supply/battery/subsystem/
/sys/class/power_supply/battery/technology
/sys/class/power_supply/battery/temp
/sys/class/power_supply/battery/type
/sys/class/power_supply/battery/uevent
/sys/class/power_supply/battery/voltage_now
```

Battery Driver

```
data->battery.properties = dummy_battery_props;
data->battery.num_properties = ARRAY_SIZE(dummy_battery_props);
data->battery.get_property = dummy_battery_get_property;
data->battery.name = "Battery";
data->battery.type = POWER_SUPPLY_TYPE_BATTERY;

data->ac.properties = dummy_ac_props;
data->ac.num_properties = ARRAY_SIZE(dummy_ac_props);
data->ac.get_property = dummy_ac_get_property;
data->ac.name = "Mains";
data->ac.type = POWER_SUPPLY_TYPE_MAINS;

data->usb.properties = dummy_usb_props;
data->usb.num_properties = ARRAY_SIZE(dummy_usb_props);
data->usb.get_property = dummy_usb_get_property;
data->usb.name = "USB";
data->usb.type = POWER_SUPPLY_TYPE_USB;

data->ac_online = 1;
data->usb_online = 0;
```

drivers/power/mango_dummy_battery.c에서
dummy_battery_probe함수에 Name을 지정하고, 속성을 지정

```
# ls /sys/class/power_supply/
Battery Mains USB
```

안드로이드 PowerServiceManager

```
private void setTimeoutLocked(long now, int
nextState)
{
    if ( mDoneBooting) {
```

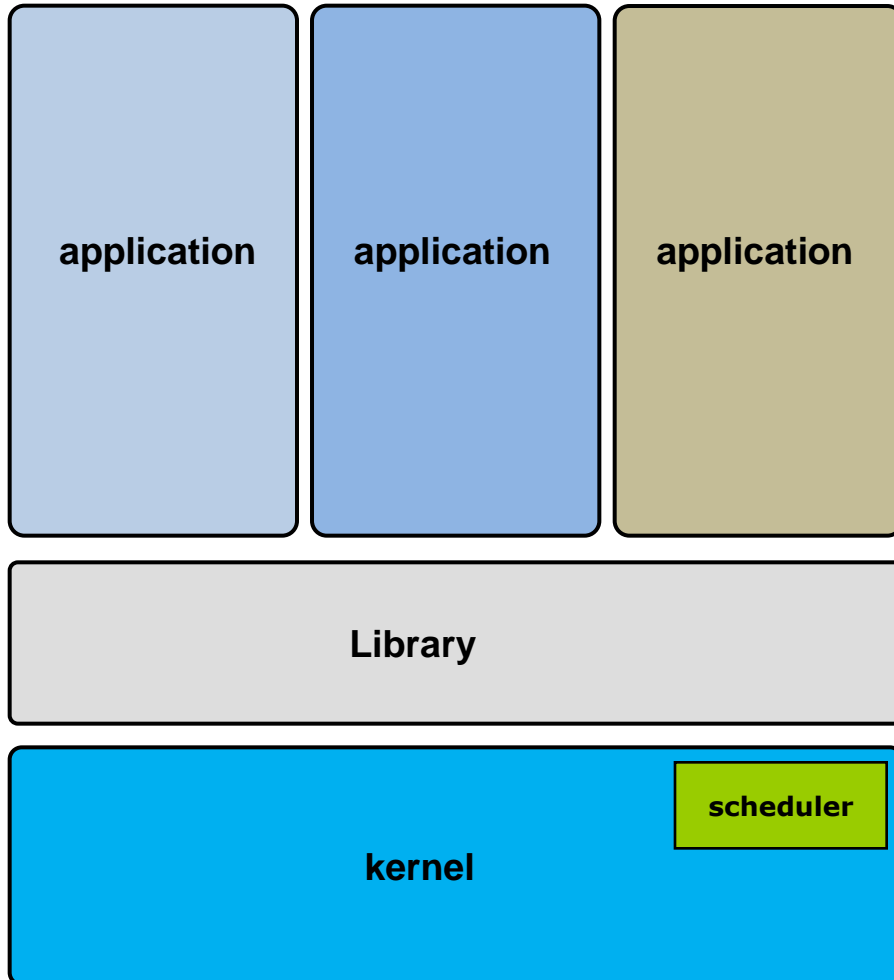


```
private void setTimeoutLocked(long now, int nextState)
{
    if ( false&mDoneBooting) {
```

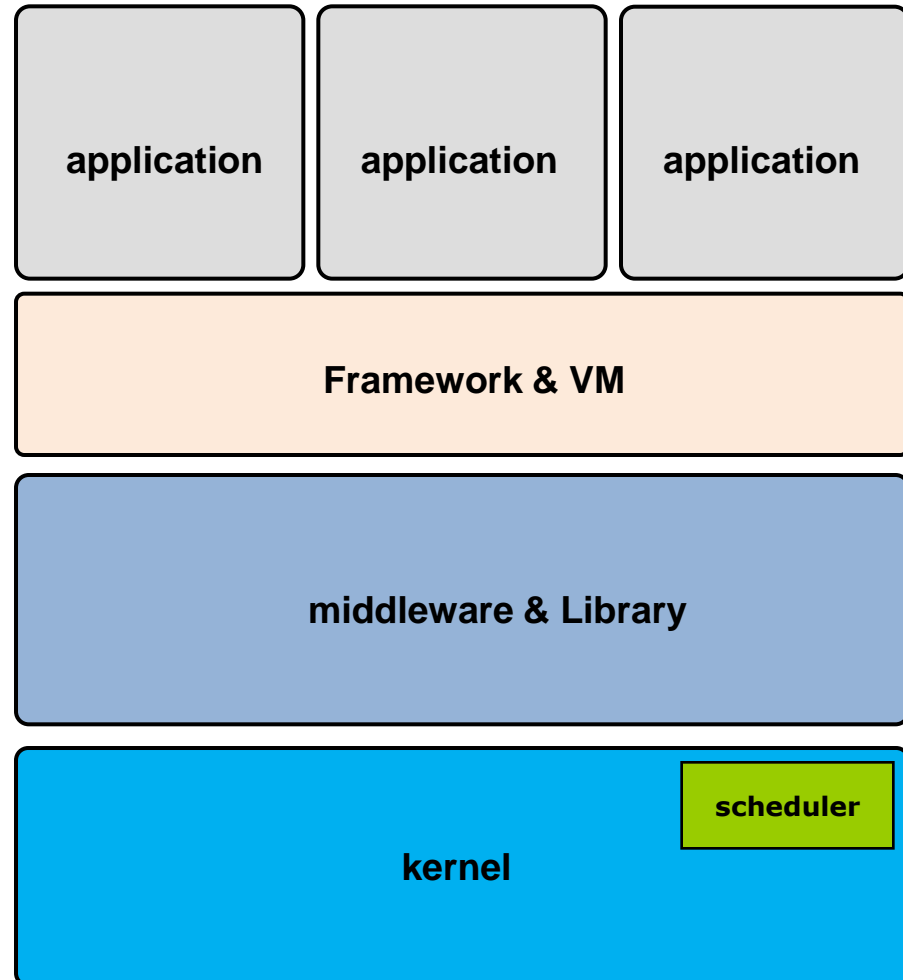
frameworks/base/services/java/com/android/server/PowerManagerService.java파일에서
안드로이드 부팅 후 Sleep으로 진입을 하지 않음

Android /Linux 파일 시스템 차이점

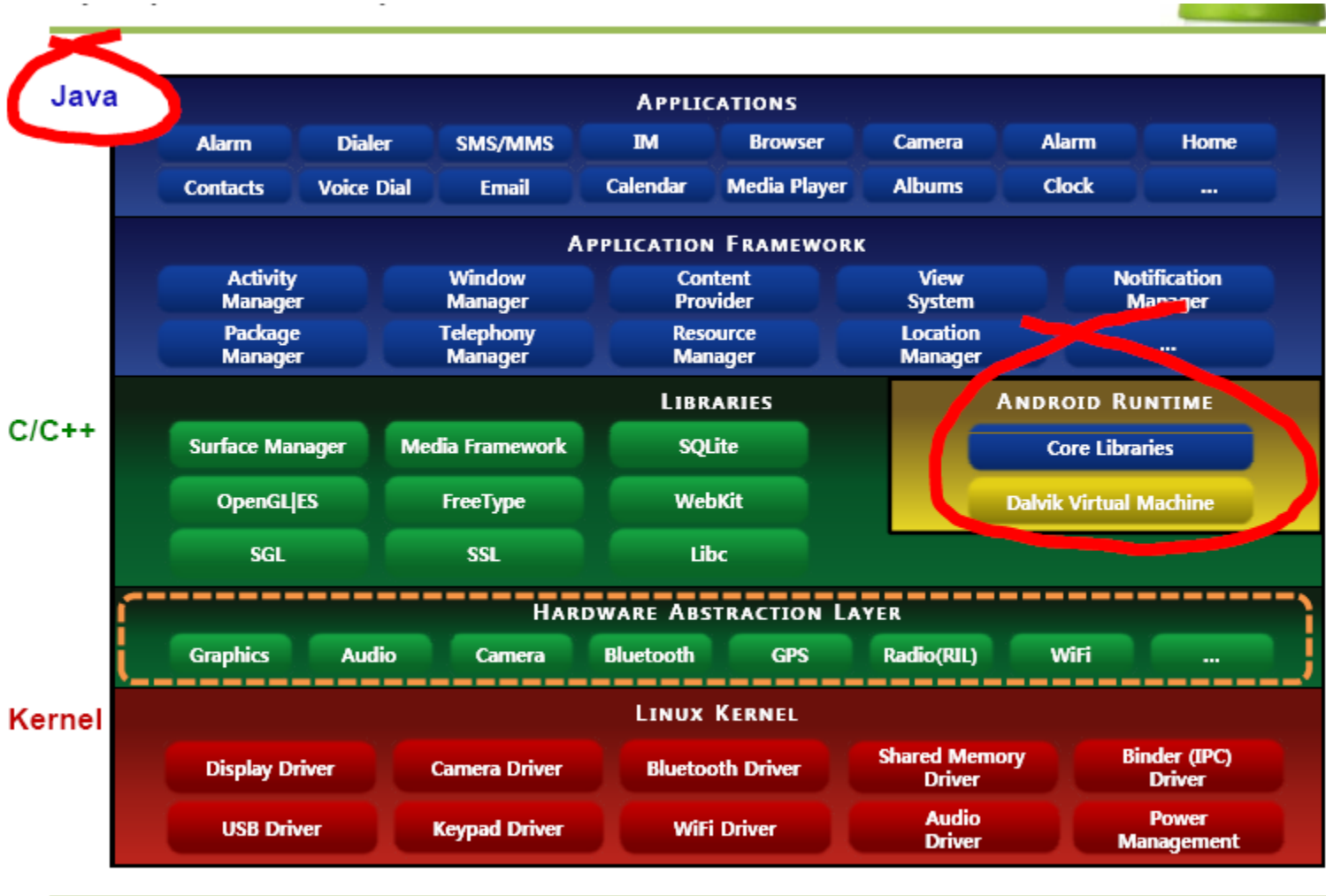
Multi-Process_(linux)



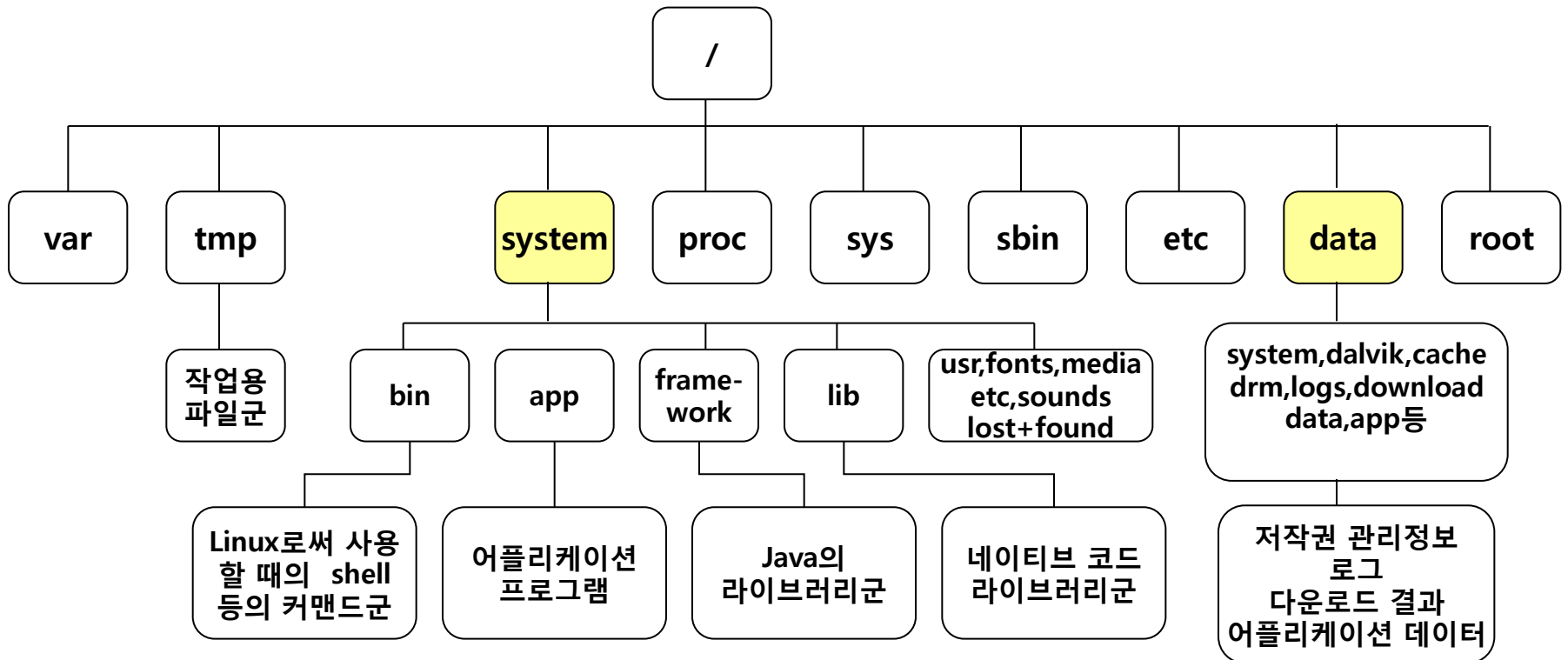
Multi-Process_(android)



안드로이드 구조



안드로이드 파일 시스템



안드로이드 코드 검색

<http://www.google.com/codesearch>

 [고급 코드 검색](#)

공개 소스 코드 검색

구문 및 예 ([regexp 구문에 대한 자세한 정보](#))

<code>regexp</code>	정규식 검색 go(2)gle hello.\world ^int printk
<code>"exact string"</code>	정확히 일치하는 문자열 검색 "compiler happy"
<code>class:regexp function:regexp</code>	<code>regexp</code> 와 이름이 일치하는 클래스 또는 함수 검색 New! class:BTREE function:laugh class:hash.*multimap function:.*range
<code>file:regexp</code>	<code>regexp</code> 와 일치하는 파일 또는 디렉토리에서만 검색 file:\js\$ XMLHttpRequest file:include/ ioctl file:/usr/sys/ken/slp.c "You are not expected to understand this."
<code>package:regexp</code>	<code>regexp</code> 와 이름이 일치하는 패키지 검색 (패키지 이름은 해당 URL 또는 CVS 서버 정보입니다.) package:perl.*tar.gz Frodo package:linux-2.6 int\ printk

안드로이드 메뉴생성 및 코덱/SPDIF 포팅

안드로이드 소스 다운 받기

- # yum install flex bison gperf libSDL-dev libesd0-dev libwxgtk2.6-dev build-essential zip curl wget
- #yum install valgrind
- #yum install python*
- # yum install zlib-devel
- #yum install libX11-devel.i686
- #yum install glibc*
- yum install git git-*
- \$ cd ~
- \$ mkdir bin
- \$ echo \$PATH
- \$ curl http://android.git.kernel.org/repo >~/bin/repo
- \$ chmod a+x ~/bin/repo
- \$ mkdir eclair
- \$ cd eclair
- \$ repo init -u git://android.git.kernel.org/platform/manifest.git -b eclair
- \$ repo sync

Mango100 소스 받기

- <http://crztech.iptime.org> 에서 최신 소스를 다운 받으시면 됩니다.

Mango 100 Source And Image

- SD booting Image (uboot, kernel, gnome, android)

- 1) [sdboot_2010_06_04](#)
- 2) [sdboot_2010_06_16](#)
- 3) [sdboot_2010_06_22](#)
- 4) [sdboot_2010_06_22\(4GB Image\)](#)
- 5) [sdboot_2010_06_30\(4GB Image\)](#)
- 6) [sdboot_2010_07_15\(4GB Image\)](#)

- Windows CE (v6.0)

- 1) [Mango100-Wince \(v6.0\)](#)

- U-boot (v1.3.4)

- 1) [mango100_uboot_2010_06_04](#)
- 2) [mango100_uboot_2010_06_29](#)

- Linux Kernel (v2.6.29)

- 1) [mango100_kernel_2010_06_04](#)
- 2) [mango100_kernel_2010_06_11](#)
- 3) [mango100_kernel_2010_06_16](#)
- 4) [mango100_kernel_2010_06_22](#)
- 5) [mango100_kernel_2010_06_30](#)
- 6) [mango100_kernel_2010_07_15](#)

- Android (Eclair, v2.1)

- 1) [mango100_eclair_2010_06_04](#)
- 2) [mango100_eclair_2010_06_11](#)
- 3) [mango100_eclair_2010_06_22](#)
- 4) [mango100_eclair_2010_06_30](#)
- 5) [mango100_eclair_2010_07_15](#)

Android Patch 및 build 방법

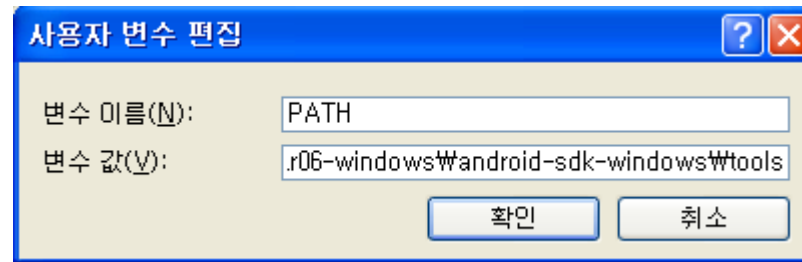
- ./frameworks/base/api/current.xml은 make update-api 실행으로 생성 됨
- ./frameworks/base/api/current.xml 복구 방법은 ./frameworks/base/api/7.xml을 copy를 해서 current.xml을 만든 후 make update-api를 실행
- #make update-api
- #./mango100_build.sh

Android SDK설치

- <http://developer.android.com/sdk/index.html>

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r06-windows.zip	23293160 bytes	7c7fcec3c6b5c7c3df6ae654b27effb5
Mac OS X (intel)	android-sdk_r06-mac_86.zip	19108077 bytes	c92abf66a82c7a3f2b8493ebe025dd22
Linux (i386)	android-sdk_r06-linux_86.tgz	16971139 bytes	848371e4bf068dbb582b709f4e56d903


- PATH 설정



Android SDK설치

- JDK설치(Window 용)

Download Information and Files

 **Get the latest Java Runtime Environment to use Sun Download Manager**


Internet Explorer Users: Check the top of this page for a "Java(TM) Web Start ActiveX Control" message in the information bar. If it appears, click it to finish detecting your Java version.

We were unable to detect a recent version of Java Runtime Environment (JRE) on your system. With the latest JRE, you can automatically download, install, and run Sun Download Manager (SDM) directly from this page. We highly recommend SDM to easily manage your downloads (pause, resume, restart, verify, and more). Visit java.com for the latest JRE.

There is more information on the available files for download on the **Supported System Configurations** page.

Instructions: Click the file name to start the download.

Your download should start automatically.
If not, click the file link below.

File Description and Name	Size
Java SE Development Kit 6u21  jdk-6u21-windows-i586.exe	75.91 MB

Android SDK설치

- Eclipse 다운로드 설치
- Eclipse 3.4또는 3.5 Download

Eclipse downloads - mirror selection

All downloads are provided under the terms and conditions of the [Eclipse Foundation Software User Agreement](#) unless otherwise specified.

Download eclipse-SDK-3.5-win32.zip from:



[\[Korea, Republic Of\] Amazon AWS \(http\)](#)

Checksums: [\[MD5\]](#) [\[SHA1\]](#)

...or pick a mirror site below.

- 적당한 곳에 압축 풀기

Android SDK설치

Eclipse 3.5 (Galileo)

1. Start Eclipse, then select **Help > Install New Software**.
2. In the Available Software dialog, click **Add...**
3. In the Add Site dialog that appears, enter a name for the remote site (for example, "Android Plugin") in the "Name" field.

In the "Location" field, enter this URL:

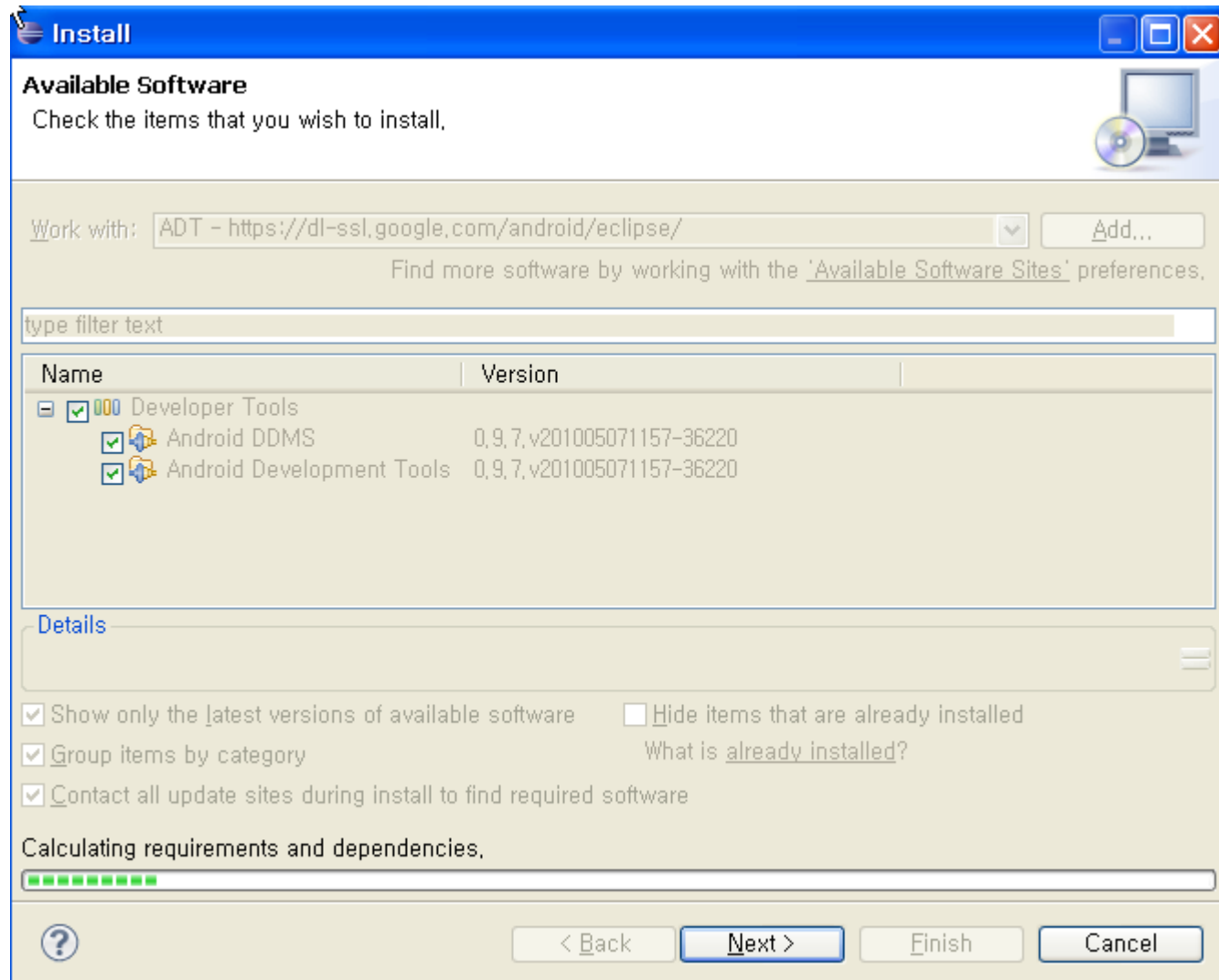
```
https://dl-ssl.google.com/android/eclipse/
```

Note: If you have trouble acquiring the plugin, you can try using "http" in the URL, instead of "https" (https is preferred for security reasons).

Click **OK**.

4. Back in the Available Software view, you should now see "Developer Tools" added to the list. Select the checkbox next to Developer Tools, which will automatically select the nested tools Android DDMS and Android Development Tools. Click **Next**.
5. In the resulting Install Details dialog, the Android DDMS and Android Development Tools features are listed. Click **Next** to read and accept the license agreement and install any dependencies, then click **Finish**.
6. Restart Eclipse.

Android SDK설치



Configuring the ADT Plugin

- <http://developer.android.com/sdk/eclipse-adt.html>

Configuring the ADT Plugin

Once you've successfully downloaded ADT as described above, the next step is to modify your ADT preferences in Eclipse to point to the Android SDK directory:

1. Select **Window** > **Preferences...** to open the Preferences panel (Mac OS X: **Eclipse** > **Preferences**).
2. Select **Android** from the left panel.
3. For the *SDK Location* in the main panel, click **Browse...** and locate your downloaded SDK directory.
4. Click **Apply**, then **OK**.

Updating the ADT Plugin

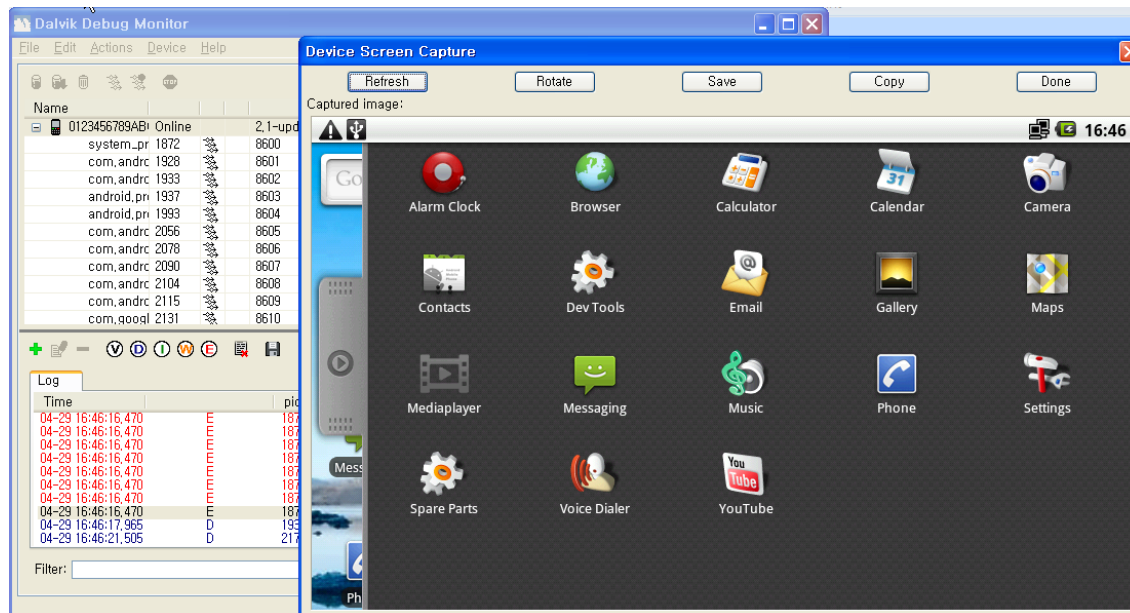
- <http://developer.android.com/sdk/eclipse-adt.html>

Eclipse 3.5 (Galileo)

1. Select **Help > Check for Updates**.
2. In the resulting Available Updates dialog, locate the Android DDMS and Android Development Tools features in the list and ensure that the checkboxes next to them are selected. Click **Next**.
If the Available Updates dialog does not list Android DDMS and Android Development tools, make sure that you have set up a remote update site for them, as described in [Installing the ADT Plugin](#).
3. In the Update Details dialog, click **Next**.
4. Read and accept the license agreement and then click **Finish**. This will download and install the latest version of Android DDMS and Android Development Tools.
5. Restart Eclipse.

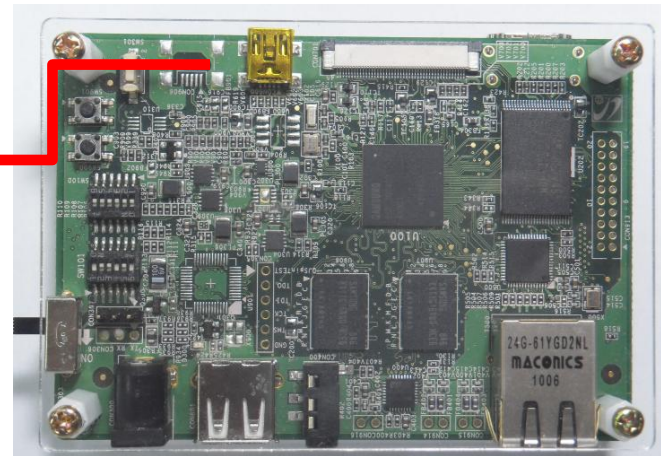
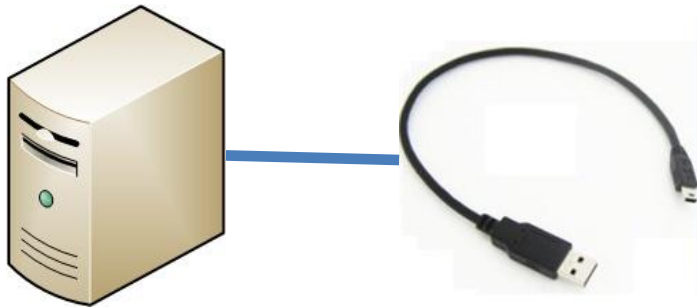
DDMS사용하기

- E:\Utils-tools\android-sdk_r06-windows\android-sdk-windows\tools>ddms.bat



ADB 연결하기

- PC와 망고보드를 usb cable로 연결



- >adb devices

```
Y:\work\busybox-work>adb devices
* daemon not running. starting it now *
* daemon started successfully *
List of devices attached
0123456789ABCDEF device
```

Busybox install

- >adb push busybox /
- >adb shell
- >ls

```
Y:#work#busybox-work>adb push busybox /
3024 KB/s <0 bytes in 1838856.000s>

Y:#work#busybox-work>adb shell
# ls
ls
data
d
default.prop
dev
init.rc
etc
proc
config
sbin
sdcard
init.goldfish.rc
init
init.mango100.rc
busybox
```

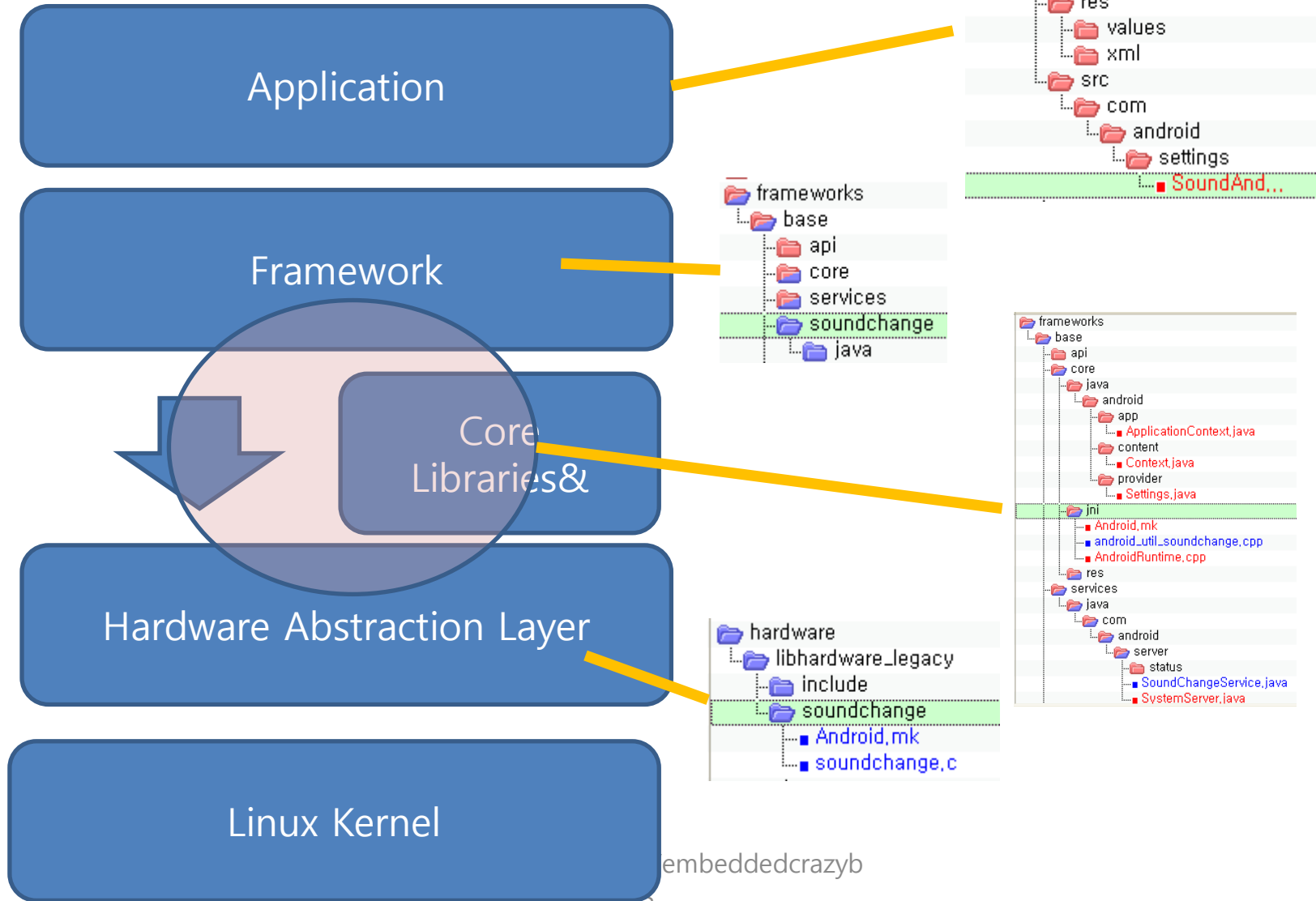
Busybox install

- # mkdir bin
- # chmod 777 busybox
- # ./busybox --install
- # cd bin
- # ./vi ../init.rc 수정

```
export PATH /bin:/sbin:/system/sbin:/system/bin:/system/sbin  
service console /bin/sh
```

- /bin/sh 또는 리부팅

Android 구조



Android 메뉴 생성(커널 수정) (WM8960,SPDIF 드라이버 포함)

sound\soc\3c24xx\Kconfig 파일에 아래와 같이 수정

```
config SND_MANGO100_WM8960
bool "WM8960 Driver"
depends on SND_S5P_MANGO100
select SND_SOC_WM8960
select SND_S5P_SOC_I2S
select SND_S3C_I2SV2_SOC
```

```
config SND_MANGO100_HDMI_SPDIF
bool "HDMI SPDIF Driver"
depends on SND_S5P_MANGO100
select SND_S5P_SPDIF
```

```
static struct snd_soc_dai_link mango_dai[] = {
{
    .name = "WM8960 I2S",
    .stream_name = "Tx/Rx",
    .cpu_dai = &s5p_i2s_dai[0],
    .codec_dai = &wm8960_dai,
    .init = mango_wm8960_init,
    .ops = &mango_i2s_ops,
},
};

static struct snd_soc_card mango = {
    .name = "mango",
    .platform = &s3c24xx_soc_platform,
    .dai_link = mango_dai,
    .num_links = ARRAY_SIZE(mango_dai),
};

static struct wm8960_setup_data mango_wm8960_setup = {
    .i2c_bus = 0,
    .i2c_address = 0x1a,
};
```

```
static struct snd_soc_dai_link mango100_dai[] = {
{
    .name = "HDMI-SPDIF",
    .stream_name = "HDMI-SPDIF Playback",
    .cpu_dai = &s5p_spdif_dai,
    .codec_dai = &s5p_hdmi_spdif_dai[0],
    .init = mango100_spdif_init,
    .ops = &mango100_spdif_ops,
},
};

static struct snd_soc_card mango100 = {
    .name = "mango100",
    .platform = &s3c24xx_soc_platform,
    .dai_link = mango100_dai,
    .num_links = ARRAY_SIZE(mango100_dai),
};
```

Android 메뉴 생성(커널 수정) (WM8960,SPDIF 드라이버 포함)

```
# ls -al /proc/asound/
dr-xr-xr-x  5 0      0      0 Apr 29 15:06 .
dr-xr-xr-x 64 0      0      0 Jan  1  1970 ..
dr-xr-xr-x  4 0      0      0 Apr 29 15:06 card0
dr-xr-xr-x  4 0      0      0 Apr 29 15:06 card1
-r--r--r--  1 0      0      0 Apr 29 15:06 cards
-r--r--r--  1 0      0      0 Apr 29 15:06 devices
lrwxrwxrwx  1 0      0      0 Apr 29 15:06 mango -> card0
lrwxrwxrwx  1 0      0      0 Apr 29 15:06 mango100 -> card1
dr-xr-xr-x  2 0      0      0 Apr 29 15:06 oss
-r--r--r--  1 0      0      0 Apr 29 15:06 pcm
-r--r--r--  1 0      0      0 Apr 29 15:06 timers
-r--r--r--  1 0      0      0 Apr 29 15:06 version

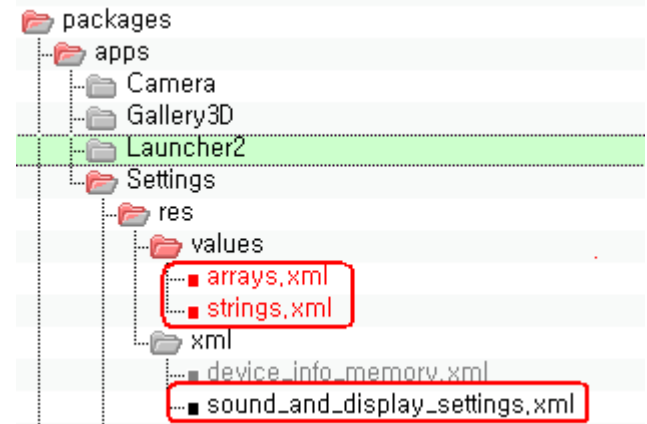
# ls /proc/asound/card0/
id      oss_mixer  pcm0c      pcm0p
# ls /proc/asound/card1
id      oss_mixer  pcm0c      pcm0p
# cat /proc/asound/card0/id
mango
# cat /proc/asound/devices
2:          : timer
3: [ 0- 0]: digital audio playback
4: [ 0- 0]: digital audio capture
5: [ 0]   : control
6: [ 1- 0]: digital audio playback
7: [ 1- 0]: digital audio capture
8: [ 1]   : control
# cat /proc/asound/cards
0 [mango          ]: WM8960 - mango
  mango (WM8960)
1 [mango100       ]: HDMI-SPDIF - mango100
  mango100 (HDMI-SPDIF)
```

<http://care.naver.com/embeddedcrazy>

Android 메뉴 생성

packages\apps\Settings\res\values\arrays.xml

```
<!-- Sound Change Setting. -->
<string-array name="sound_change_entries">
  <item>WM8960</item>
  <item>SPDIF</item>
</string-array>
<!-- Do not translate. -->
<string-array name="sound_change_values" translatable="false">
  <item>10</item>
  <item>20</item>
</string-array>
```



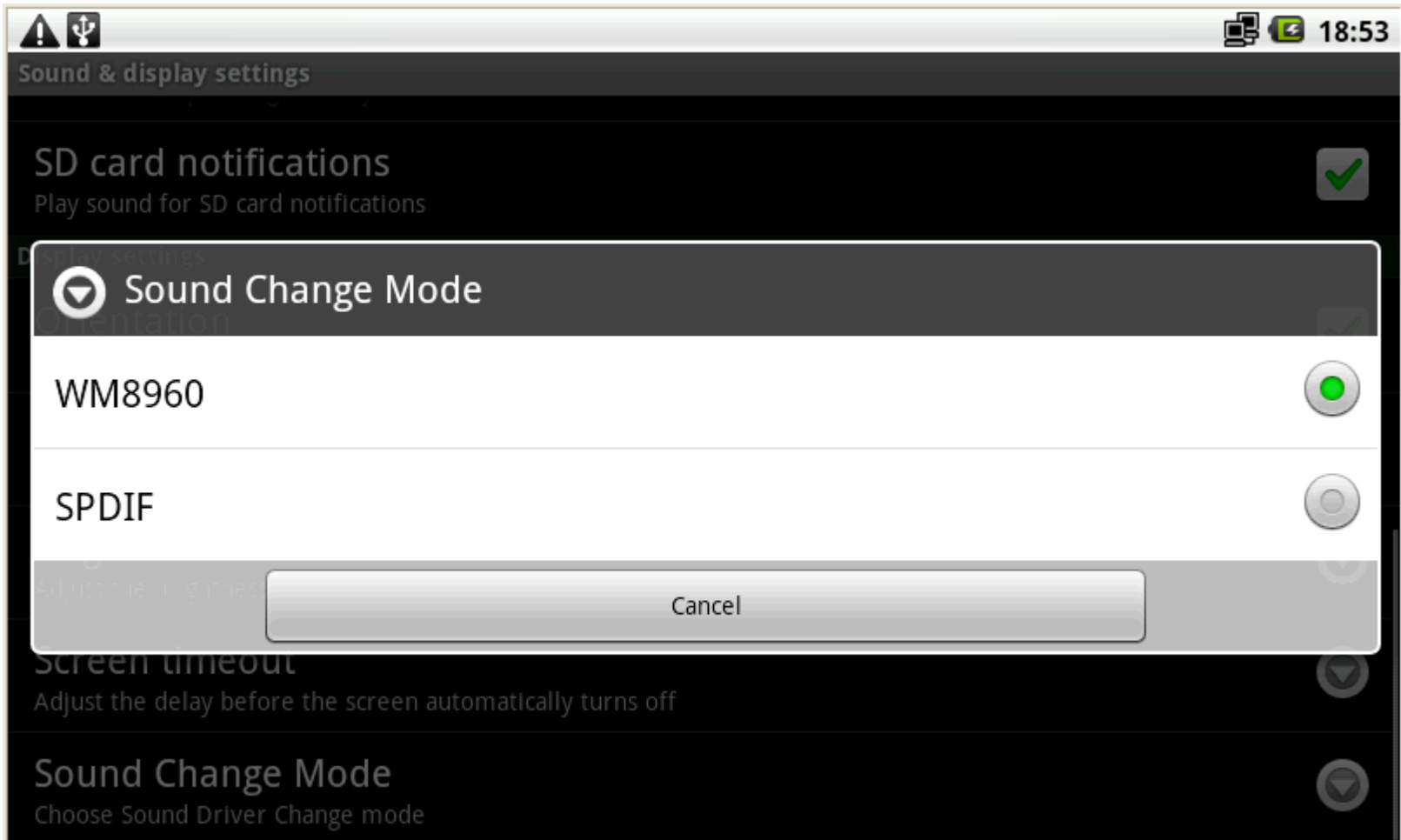
packages\apps\Settings\res\values\strings.xml

```
<string name="sound_change">Sound Change Mode</string>
<!-- Sound & display settings screen, setting option summary to change Sound Driver -->
<string name="sound_change_summary">Choose Sound Driver Change mode</string>
<!-- Sound & display settings screen, setting option name to change Sound Driver Mode -->
```

packages\apps\Settings\res\xml\sound_and_display_settings.xml

```
<ListPreference
  android:key="sound_change"
  android:title="@string/sound_change"
  android:summary="@string/sound_change_summary"
  android:persistent="false"
  android:entries="@array/sound_change_entries"
  android:entryValues="@array/sound_change_values" />
```

Android 메뉴 생성



Android 메뉴 생성

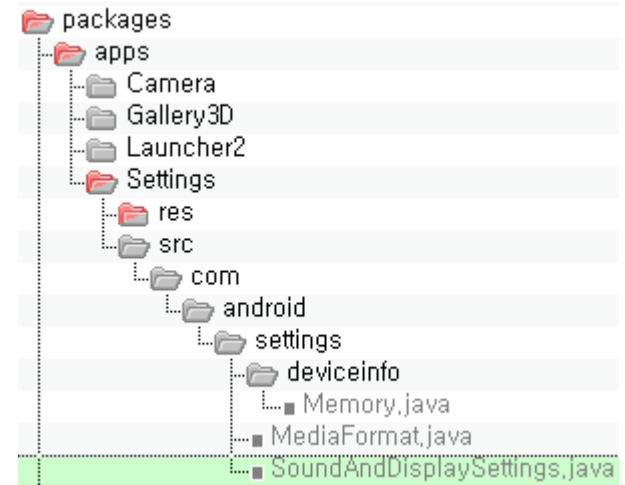
SoundAndDisplaySettings.java

```
import static android.provider.Settings.System.SOUND_CHANGE;
import android.util.soundchange.SoundChangeManager;
public class SoundAndDisplaySettings extends PreferenceActivity implements
    Preference.OnPreferenceChangeListener {
    private static final String TAG = "SoundAndDisplaySettings";

    /** If there is no setting in the provider, use this. */
    private static final int FALLBACK_SCREEN_TIMEOUT_VALUE = 30000;
    private static final int FALLBACK_SOUND_CHANGE_VALUE = 10;
    private static final int FALLBACK_EMERGENCY_TONE_VALUE = 0;

    private static final String KEY_SILENT = "silent";
    private static final String KEY_VIBRATE = "vibrate";
    private static final String KEY_SCREEN_TIMEOUT = "screen_timeout";
    private static final String KEY_SOUND_CHANGE = "sound_change";
```

```
private SoundChangeManager mSoundChangeManager;
ListPreference SoundChangePreference =
    (ListPreference) findPreference(KEY_SOUND_CHANGE);
    SoundChangePreference.setValue(String.valueOf(Settings.System.getInt(
        resolver, SOUND_CHANGE, FALLBACK_SOUND_CHANGE_VALUE)));
    SoundChangePreference.setOnPreferenceChangeListener(this);
    mSoundChangeManager = (SoundChangeManager) getSystemService(SOUNDCHANGE_SERVICE);
./packages/apps/Settings/src/com/android/settings/SoundAndDisplaySettings.java
```



```
public static final String SOUND_CHANGE = "sound_change";
./frameworks/base/core/java/android/provider/Settings.java
```

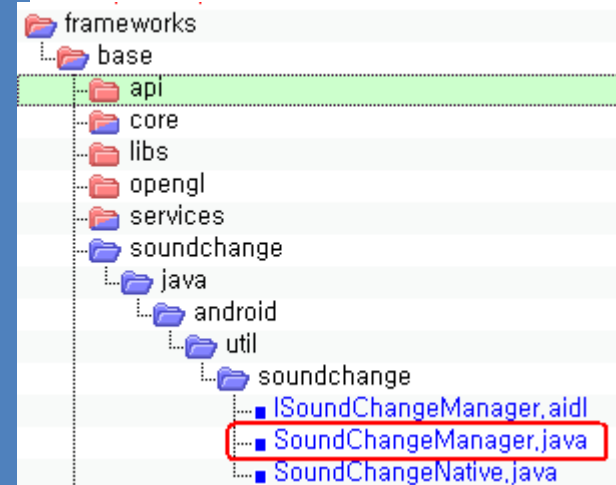
Android 메뉴 생성

SoundAndDisplaySettings.java

```
import android.util.soundchange.SoundChangeManager;  
./packages/apps/Settings/src/com/android/settings/SoundAndDisplaySettings.java
```

SoundChangeManager.java,ISoundChangeManager.aidl,SoundChangeNative.java
소스 추가

```
package android.util.soundchange;  
  
import java.util.List;  
  
import android.annotation.SdkConstant;  
import android.annotation.SdkConstant.SdkConstantType;  
import android.os.Handler;  
import android.os.RemoteException;  
import android.util.Log;  
  
public class SoundChangeManager {
```



Android 메뉴 생성

SoundAndDisplaySettings.java

```
private SoundChangeManager mSoundChangeManager;  
ListPreference SoundChangePreference =  
    (ListPreference) findPreference(KEY_SOUND_CHANGE);  
    SoundChangePreference.setValue(String.valueOf(Settings.System.getInt(  
        resolver, SOUND_CHANGE, FALLBACK_SOUND_CHANGE_VALUE)));  
    SoundChangePreference.setOnPreferenceChangeListener(this);  
    mSoundChangeManager = (SoundChangeManager) getSystemService(SOUNDCHANGE_SERVICE);  
./packages/apps/Settings/src/com/android/settings/SoundAndDisplaySettings.java
```

```
public Object getSystemService(String name) {  
    ..  
    }else if (SOUNDCHANGE_SERVICE.equals(name)) {  
        return getSoundChangeManager();  
    }  
./frameworks/base/core/java/android/app/ApplicationContext.java
```

```
private SoundChangeManager getSoundChangeManager()  
{  
    synchronized (sSync) {  
        if (sSoundChangeManager == null) {  
            IBinder b = ServiceManager.getService(SOUNDCHANGE_SERVICE);  
            ISoundChangeManager service = ISoundChangeManager.Stub.asInterface(b);  
            sSoundChangeManager = new SoundChangeManager(service);  
        }  
    }  
    return sSoundChangeManager;  
}./frameworks/base/core/java/android/app/ApplicationContext.java
```


Android 메뉴 생성

SoundAndDisplaySettings.java

APP

- `mSoundChangeManager = (SoundChangeManager) getSystemService(SOUNDCHANGE_SERVICE);`
- `./packages/apps/Settings/src/com/android/settings/SoundAndDisplaySettings.java`

FRAMEWORK

- `public Object getSystemService(String name) {..`
- `./frameworks/base/core/java/android/app/ApplicationContext.java`

FRAMEWORK

- `private SoundChangeManager getSoundChangeManager(){`
- `}/frameworks/base/core/java/android/app/ApplicationContext.java`

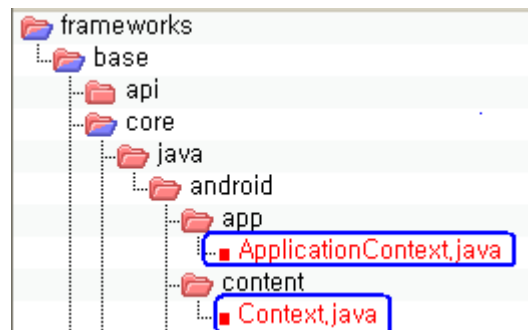
Android Framework

```
public Object getSystemService(String name) {  
    ..  
    }else if (SOUNDCHANGE_SERVICE.equals(name)) {  
        return getSoundChangeManager();  
    }  
}
```

./frameworks/base/core/java/android/app/ApplicationContext.java

```
public static final String SOUNDCHANGE_SERVICE = "soundchange";
```

./frameworks/base/core/java/android/content/Context.java



Android Framework

```
private SoundChangeManager getSoundChangeManager()
{
    synchronized (sSync) {
        if (sSoundChangeManager == null) {
            IBinder b = ServiceManager.getService(SOUNDCHANGE_SERVICE);
            ISoundChangeManager service = ISoundChangeManager.Stub.asInterface(b);
            sSoundChangeManager = new SoundChangeManager(service);
        }
    }
    return sSoundChangeManager;
}
./frameworks/base/core/java/android/app/ApplicationContext.java
```

```
soundchange = new SoundChangeService(context);
ServiceManager.addService(Context.SOUNDCHANGE_SERVICE, soundchange);
./frameworks/base/services/java/com/android/server/SystemServer.java
```

```
package com.android.server;
public class SoundChangeService <synchronized> extends ISoundChangeManager.Stub{
    private static final String TAG = "SoundChangeService";

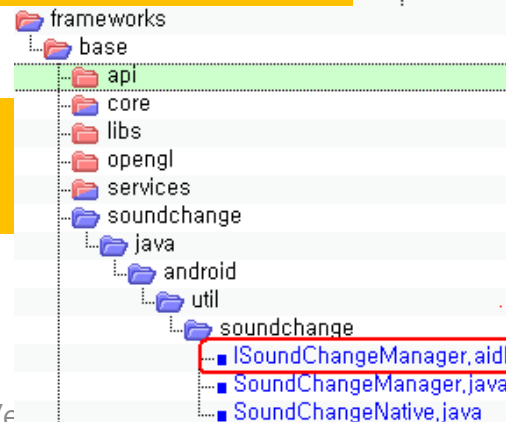
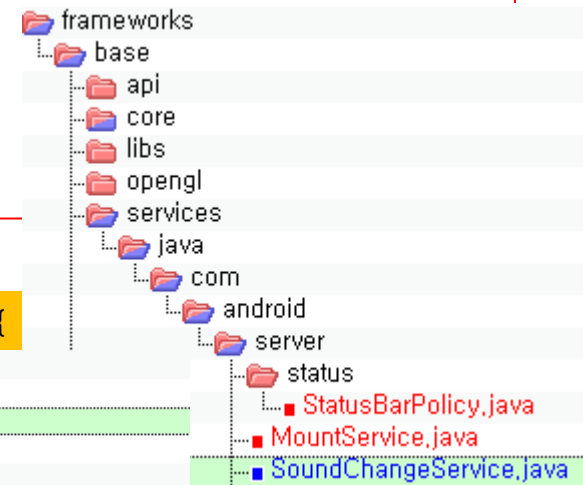
    public SoundChangeService(Context context){
        mContext = context;
        mPreSoundChangeState = getPersistedState();
        setSoundChangeMode(mPreSoundChangeState);
    }
}
frameworks/base/services/java/com/android/server/SoundChangeService.java
```

Android Framework(Binder)

```
private SoundChangeManager getSoundChangeManager()
{
    synchronized (sSync) {
        if (sSoundChangeManager == null) {
            IBinder b = ServiceManager.getService(SOUNDCHANGE_SERVICE);
            ISoundChangeManager service = ISoundChangeManager.Stub.asInterface(b);
            sSoundChangeManager = new SoundChangeManager(service);
        }
    }
    return sSoundChangeManager;
}
./frameworks/base/core/java/android/app/ApplicationContext.java
```

```
public class SoundChangeService <synchronized> extends ISoundChangeManager.Stub{
```

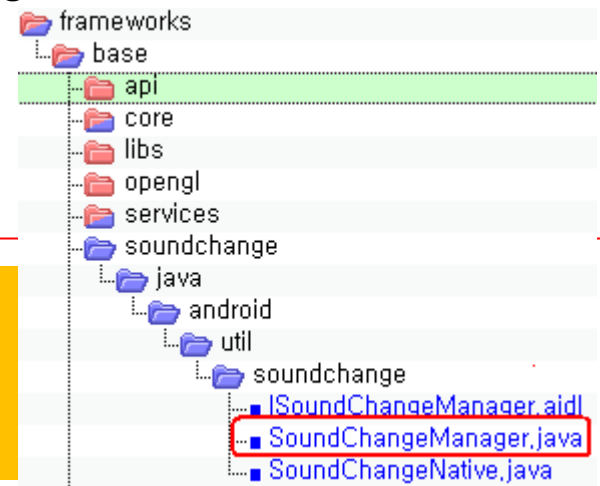
```
import android.util.soundchange.ISoundChangeManager;
./frameworks/base/core/java/android/app/ApplicationContext.java
```



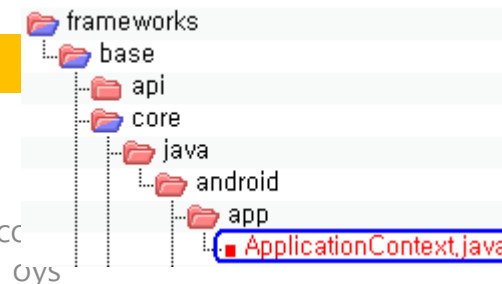
Android Framework(Binder)

```
private SoundChangeManager getSoundChangeManager()
{
    synchronized (sSync) {
        if (sSoundChangeManager == null) {
            IBinder b = ServiceManager.getService(SOUNDCHANGE_SERVICE);
            ISoundChangeManager service = ISoundChangeManager.Stub.asInterface(b);
            sSoundChangeManager = new SoundChangeManager(service);
        }
    }
    return sSoundChangeManager;
}
./frameworks/base/core/java/android/app/ApplicationContext.java
```

```
ISoundChangeManager mService;
public SoundChangeManager(ISoundChangeManager service) {
    Log.i(TAG, "Init SoundChange Manager");
    mService = service;
}
```



```
import android.util.soundchange.SoundChangeManager;
```



<http://cafe.naver.coys>

Android 메뉴 생성

SoundAndDisplaySettings.java

```
public boolean onPreferenceChange(Preference preference, Object objValue) {  
    ...  
} else if (KEY_SOUND_CHANGE.equals(key)) {  
    int value = Integer.parseInt((String) objValue);  
    try {  
        Settings.System.putInt(getContentResolver(),  
            SOUND_CHANGE, value);  
    } catch (NumberFormatException e) {  
        Log.e(TAG, "Failed to persist sound change mode setting", e);  
    }  
    // update manager  
    mSoundChangeManager.setSoundChangeMode(value);  
}. packagesWappsWSettingsWsrcWcomWandroidWsettingsWSoundAndDisplaySettings.java
```

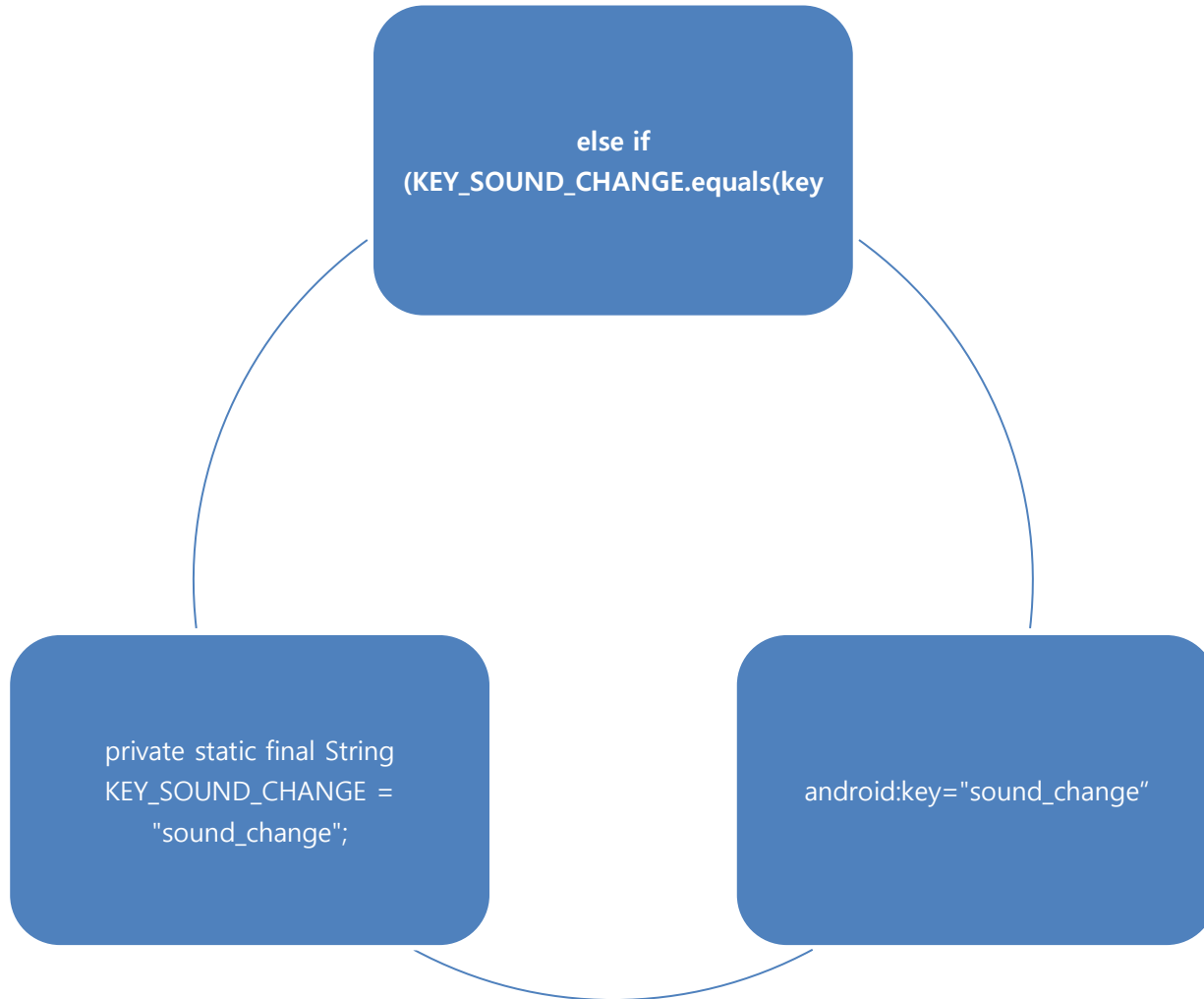
```
private static final String KEY_SOUND_CHANGE = "sound_change";  
. packagesWappsWSettingsWsrcWcomWandroidWsettingsWSoundAndDisplaySettings.java
```

<ListPreference

android:key="sound_change"

packagesWappsWSettingsWresWxmlWsound_and_display_settings.xml

Android 메뉴 생성 SoundAndDisplaySettings.java



Android 메뉴 생성(APP~JNI)

```
./packages/apps/Settings/src/com/android/settings/SoundAndDisplaySettings.java
```

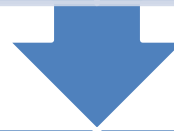
```
mSoundChangeManager.setSoundChangeMode(value  
);
```



```
./frameworks/base/services/java/com/android/server/SoundChangeService.java
```

```
setSoundChangeMode(mPreSoundChangeState);
```

```
SoundChangeNative.SoundDriverChange(mode);
```



```
./frameworks/base/soundchange/java/android/util/soundchange/SoundChangeNative.java
```

```
public native static void SoundDriverChange(int mode);}
```


Android Framework(JNI)

service

- `SoundChangeNative.SoundDriverChange(mode);`
- `./frameworks/base/services/java/com/android/server/SoundChangeService.java`

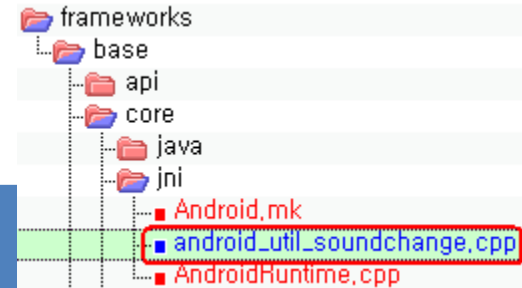
JNI

- `public native static void SoundDriverChange(int mode);`
- `./frameworks/base/soundchange/java/android/util/soundchange/SoundChangeNative.java`

JNI

- `{ "SoundDriverChange", "(I)V", (void*)android_util_soundchange_SoundChangeManager },`
- `./frameworks/base/core/jni/android_util_soundchange.cpp`

Android Framework(JNI)




```
return AndroidRuntime::registerNativeMethods(env,  
"android/util/soundchange/SoundChangeNative", g_methods, NELEM(g_methods));
```

```
public class SoundChangeNative {  
    public native static void SoundDriverChange(int mode);  
};  
./frameworks/base/soundchange/java/android/util/soundchange/SoundChangeNative.java
```

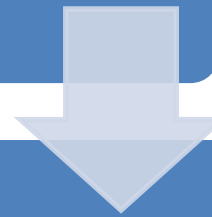
```
static JNINativeMethod g_methods[] = {  
    { "SoundDriverChange", "(I)V", (void*)android_util_soundchange_SoundChangeManager },  
};
```

Android Framework(JNI~HAL)

```
android_util_soundchange_SoundChangeManager()  
{::sound_change_service(mode);}  
./frameworks/base/core/jni/android_util_soundchange.cpp
```



```
int sound_change_service(int mode)  
hardware/libhardware_legacy/soundchange/soundchange.c
```



```
file_copy("/system/etc/asound_wm8960.conf","/system/etc/asound.conf");
```

Android Framework(LIB~HAL)

libandroid_runtime.so

./frameworks/base/core/jni/android_util_soundchange.cpp

```
::sound_change_  
service(mode);
```

```
./hardware/libhardware_legacy/soundchange/soundchange.  
c
```

```
int sound_change_service(int  
mode)
```

Libruntime.so등록하기 (core Libraries)

- `extern int register_android_util_soundchange_SoundChangeManager(JNIEnv* env);`
- `REG_JNI(register_android_util_soundchange_SoundChangeManager),`
- `frameworks\base\core\jni\AndroidRuntime.cpp` 에 추가



Android 메뉴 생성 (aidl등록)

```
LOCAL_SRC_FILES += ₩
```

```
soundchange/java/android/util/soundchange/ISoundChangeManager.aidl ₩
```

. Frameworks/base/Android.mk 파일에 추가

```
LOCAL_SRC_FILES := $(call find-other-java-files,$(FRAMEWORKS_BASE_SUBDIRS))
```



```
define find-other-java-files
    $(call find-subdir-files,$(1) -name "*.java" -and -not -name
    ".*")
Endef
./build/core/definitions.mk
```

Android (current.xml)

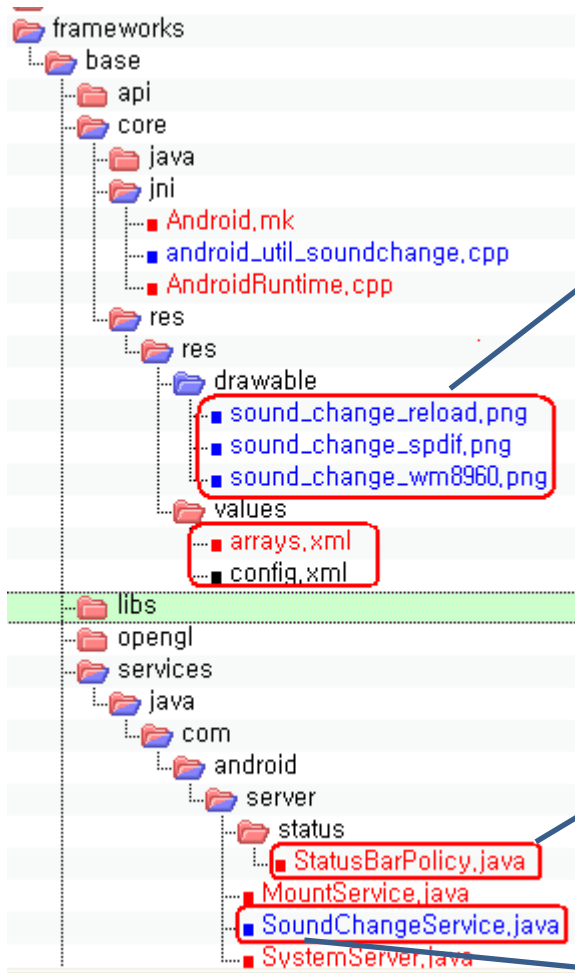
frameworks\base\api\current.xml

```
<field name="SOUNDCHANGE_SERVICE"  
  type="java.lang.String"  
  transient="false"  
  volatile="false"  
  value="&quot;soundchange&quot;"  
  static="true"  
  final="true"  
  deprecated="not deprecated"  
  visibility="public"  
>  
</field>
```

```
<field name="SOUND_CHANGE"  
  type="java.lang.String"  
  transient="false"  
  volatile="false"  
  value="&quot;sound_change&quot;"  
  static="true"  
  final="true"  
  deprecated="not deprecated"  
  visibility="public"  
>  
</field>
```

make update-api : 새로 추가된 java class 를 android platform 에 등록.
#make update-api를 실행하면 current.xml이 자동으로 등록 됨
#make
update-api: \$(INTERNAL_PLATFORM_API_FILE) | \$(ACP)
./build/core/tasks/apicheck.mk

Android 아이콘 생성



아이콘이미지

```
else if  
(action.equals(SoundChangeManager.SOUND_CHANGE_STATE_CHANGED_A  
CTION)){  
    updateSoundChange(intent);
```

Control
(SoundChangeManager.SOUND_CHANGE_STATE_CHANGED_ACTION)

Android 아이콘 생성

```
final Intent intent = new Intent(SoundChangeManager.SOUND_CHANGE_STATE_CHANGED_ACTION);  
mContext.sendStickyBroadcast(intent);  
./frameworks/base/services/java/com/android/server/SoundChangeService.java
```



```
filter.addAction(SoundChangeManager.SOUND_CHANGE_STATE_CHANGED_ACTION);
```

```
private BroadcastReceiver mIntentReceiver = new BroadcastReceiver() {  
else if (action.equals(SoundChangeManager.SOUND_CHANGE_STATE_CHANGED_ACTION)){  
    updateSoundChange(intent);  
./frameworks/base/services/java/com/android/server/status/StatusBarPolicy.java
```



```
updateSoundChange(intent);
```

Android 아이콘 생성



```
private final void updateSoundChange(Intent intent) {
    final int event = intent.getIntExtra(SoundChangeManager.EXTRA_SOUND_CHANGE_STATE, SoundChangeManager.SOUND_CHANGE_STATE_DEFAULT);
    int iconId;
    switch (event) {
        case SoundChangeManager.SOUND_CHANGE_STATE_WM8960:
        default:
            iconId = sSoundChangeImages[0];
            break;
        case SoundChangeManager.SOUND_CHANGE_STATE_SPDIF:
            iconId = sSoundChangeImages[1];
            break;
        case SoundChangeManager.SOUND_CHANGE_STATE_RELOAD:
            iconId = sSoundChangeImages[2];
            break;
    }
    mSoundChangeData.iconId = iconId;
    mService.updateIcon(mSoundChangeIcon, mSoundChangeData, null);
}
```

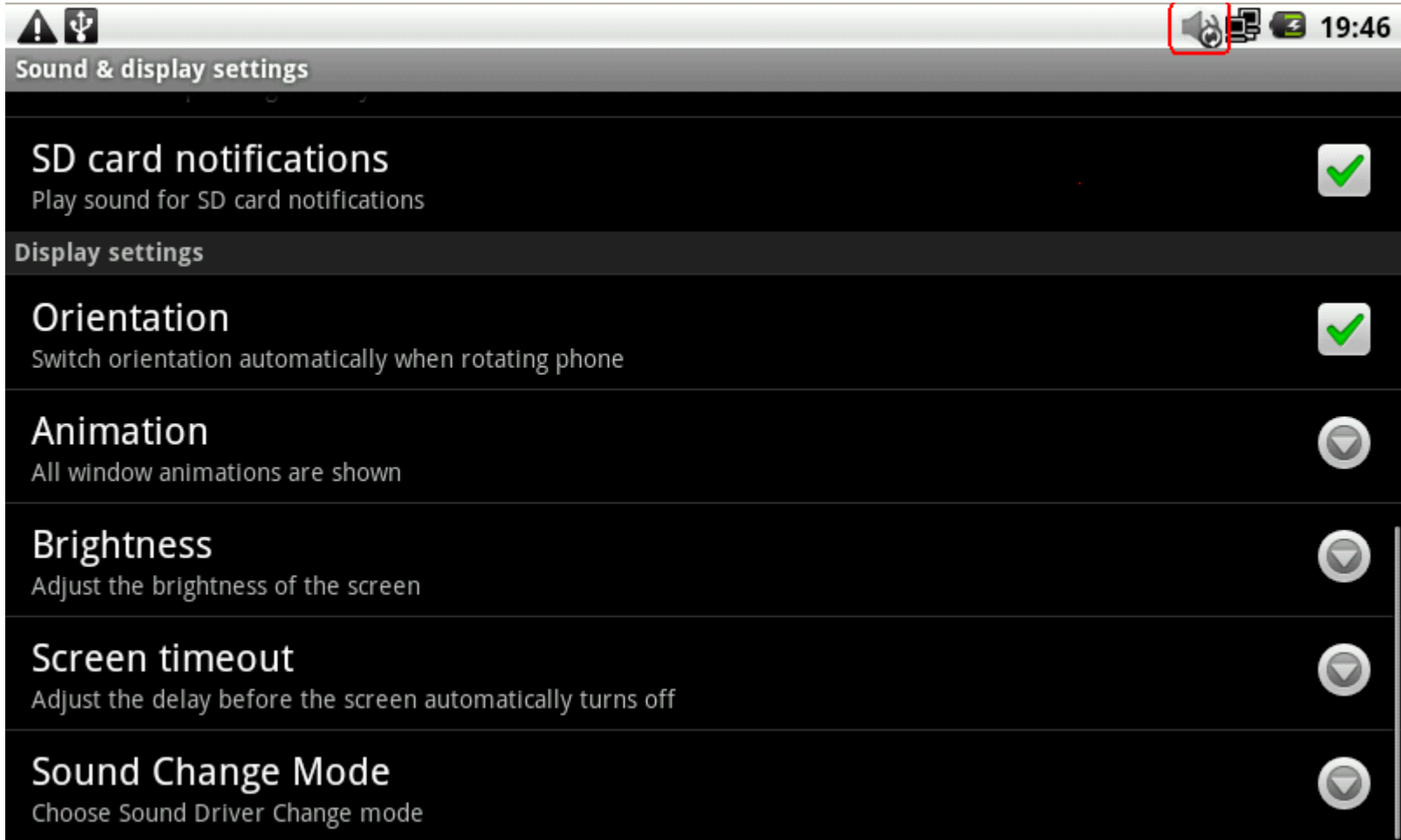
```
intent.putExtra(SoundChangeManager.EXTRA_SOUND_CHANGE_STATE, mod
```

```
private static final int[] sSoundChangeImages = new int[] {
    com.android.internal.R.drawable.sound_change_wm8960,
    com.android.internal.R.drawable.sound_change_spdif,
    com.android.internal.R.drawable.sound_change_reload
};
private IBinder mSoundChangeIcon;
private IconData mSoundChangeData;
```

```
public void updateIcon(IBinder key,
    String slot, String iconPackage, int iconId, int iconLevel) {
    enforceStatusBar();
    updateIcon(key, IconData.makeIcon(slot, iconPackage, iconId, iconLevel, 0), null);
}
```

./frameworks/base/services/java/com/android/server/status/StatusBarService.java

안드로이드 아이콘 생성 결과

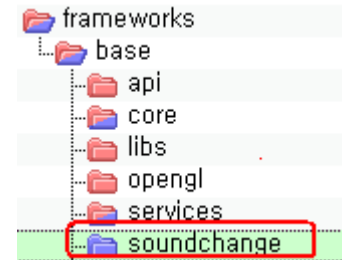


안드로이드 메뉴 생성(build)

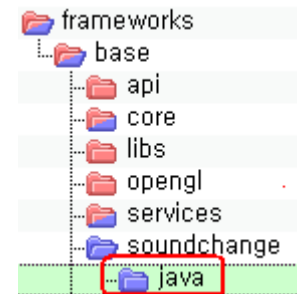
build\wcore\pathmap.mk 파일 수정

```
FRAMEWORKS_BASE_SUBDIRS := ₩  
$(addsuffix /java, ₩  
  core ₩  
  graphics ₩  
  location ₩  
  media ₩  
  opengl ₩  
  sax ₩  
  telephony ₩  
  ethernet ₩  
  wifi ₩  
  vpn ₩  
  keystore ₩  
  soundchange ₩  
  )
```

```
FRAMEWORKS_BASE_JAVA_SRC_DIRS := ₩  
$(addprefix frameworks/base/,$(FRAMEWORKS_BASE_SUBDIRS))
```



addsuffix /java ↓



안드로이드 메뉴 생성(aidl)

frameworks\base\Android.mk 파일 수정

```
# FRAMEWORKS_BASE_SUBDIRS comes from build/core/pathmap.mk
LOCAL_SRC_FILES := $(call find-other-java-files,$(FRAMEWORKS_BASE_SUBDIRS))

LOCAL_SRC_FILES := $(filter-out \#
                        org/mobilecontrol/% \#
                        ,$(LOCAL_SRC_FILES))
ifeq ($(TARGET_BUILD_TYPE),debug)
    LOCAL_SRC_FILES += $(call find-other-java-files,core/config/debug)
else
    LOCAL_SRC_FILES += $(call find-other-java-files,core/config/ndebug)
endif

LOCAL_SRC_FILES += \#

    ethernet/java/android/net/ethernet/IEthernetManager.aidl \#
    soundchange/java/android/util/soundchange/ISoundChangeManager.aidl \#
```

AIDL은 <http://cafe.naver.com/embeddedcrazyboys>의 약자

Android build (find-other-java-files)

```
LOCAL_SRC_FILES := $(call find-other-java-files,$(FRAMEWORKS_BASE_SUBDIRS))
```



```
define find-other-java-files
    $(call find-subdir-files,$(1) -name "*.java" -and -not -name
".*")
Endef
./build/core/definitions.mk
```

안드로이드 메뉴 생성

frameworks\base\core\java\android\app\ApplicationContext.java에 추가

```
import android.util.soundchange.ISoundChangeManager;  
import android.util.soundchange.SoundChangeManager;
```

frameworks\base\soundchange\java\android\util\soundchange\SoundChangeManager.java 를 포함

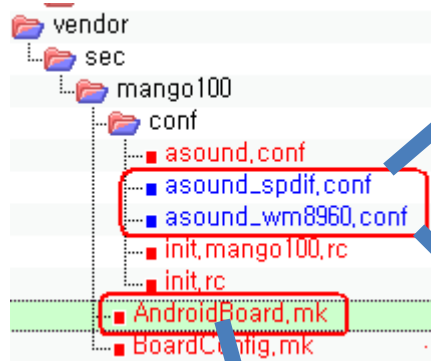
```
package android.util.soundchange;  
  
import java.util.List;  
  
import android.annotation.SdkConstant;
```

frameworks\base\soundchange\java\android\util\soundchange\ISoundChangeManager.aidl

```
package android.util.soundchange;  
  
interface ISoundChangeManager  
{  
    int getSoundChangeStatus();  
    int getPreSoundChangeStatus();  
    void setSoundChangeMode(int mode);  
}
```

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

Build 환경



```
ctl.AndroidPlayback {
    type hw
    card 1
}
```

```
# cat info
card: 1
device: 0
subdevice: 0
stream: PLAYBACK
id: HDMI-SPDIF Playback HDMI-SPDIF Codec-0
name:
subname: subdevice #0
class: 0
subclass: 0
subdevices_count: 1
subdevices_avail: 1
# pwd
/proc/asound/card1/pcm0p
```

```
ctl.AndroidPlayback {
    type hw
    card 0
}
```

```
# pwd
/proc/asound/card0/pcm0c
# cat info
card: 0
device: 0
subdevice: 0
stream: CAPTURE
id: Tx/Rx WM8960-0
name:
subname: subdevice #0
class: 0
subclass: 0
subdevices_count: 1
subdevices_avail: 1
```

PRODUCT_COPY_FILES += ₩

vendor/sec/mango100/conf/asound.conf:system/etc/asound.conf ₩

vendor/sec/mango100/conf/asound_wm8960.conf:system/etc/asound_wm8960.conf ₩

vendor/sec/mango100/conf/asound_spdif.conf:system/etc/asound_spdif.conf ₩

안드로이드 아이콘 생성

```
<item> <xliff:g id="id">soundchange</xliff:g> </item>
```

frameworks\base\core\res\res\values\arrays.xml 파일에 추가

```
<!-- Declared at res/drawable/sound_change_reload.png:0 -->  
<public type="drawable" name="sound_change_reload" id="0x01080271"  
>  
./out/target/common/obj/APPS/framework-  
res_intermediates/public_resources.xml
```

SD Card 인식(Froyo)

- 부팅 후 아래와 같이 수정
- `dev_mount sdcard /mnt/sdcard auto /devices/platform/s3c-sdhci.0/mmc_host/mmc0`
- 또는
- `vendor/sec/mango100/conf/vold.fstab` 파일을 아래와 같이 수정
- `dev_mount sdcard /mnt/sdcard auto /devices/platform/s3c-sdhci.0/mmc_host/mmc0`

ALSA Sound 소스 받기(froyo)

#vi .repo/manifest.xml 을 수정

```
<project path="external/alsa-lib" name="platform/external/alsa-lib" />
```

```
<project path="external/alsa-utils" name="platform/external/alsa-utils" />
```

```
<project path="hardware/alsa_sound" name="platform/hardware/alsa_sound" />
```

#repo sync

#ls external

```
[icanjji@localhost froyo]$ ls external/  
alsa-lib      dhcpd         fdlibm        iptables  
alsa-utils    dnsmasq      freetype      jdiff  
apache-http   dropbear     fsck_msdos    jhead  
astl          2f494949     2f494949     2f494949
```

#ls hardware

```
[icanjji@localhost froyo]$ ls hardware/  
alsa_sound    broadcom     libhardware   libhardware_legacy
```

ALSA Sound 설정(froyo)

```
build/target/board/generic/BoardConfig.mk  
을 추가  
BOARD_USES_GENERIC_AUDIO := false  
BUILD_WITH_ALSA_UTILS := true  
BOARD_USES_ALSA_AUDIO := true
```

```
-asound.conf 파일을 작성 파일 시스템에 복사  
# cp asound.conf system/usr/share/alsa_sound/
```

```
또는  
build/target/board/generic/AndroidBoard.mk 파일에 아래와 같이 수정  
PRODUCT_COPY_FILES += \
```

```
    build/target/board/generic/conf/asound.conf:system/etc/asound.conf
```

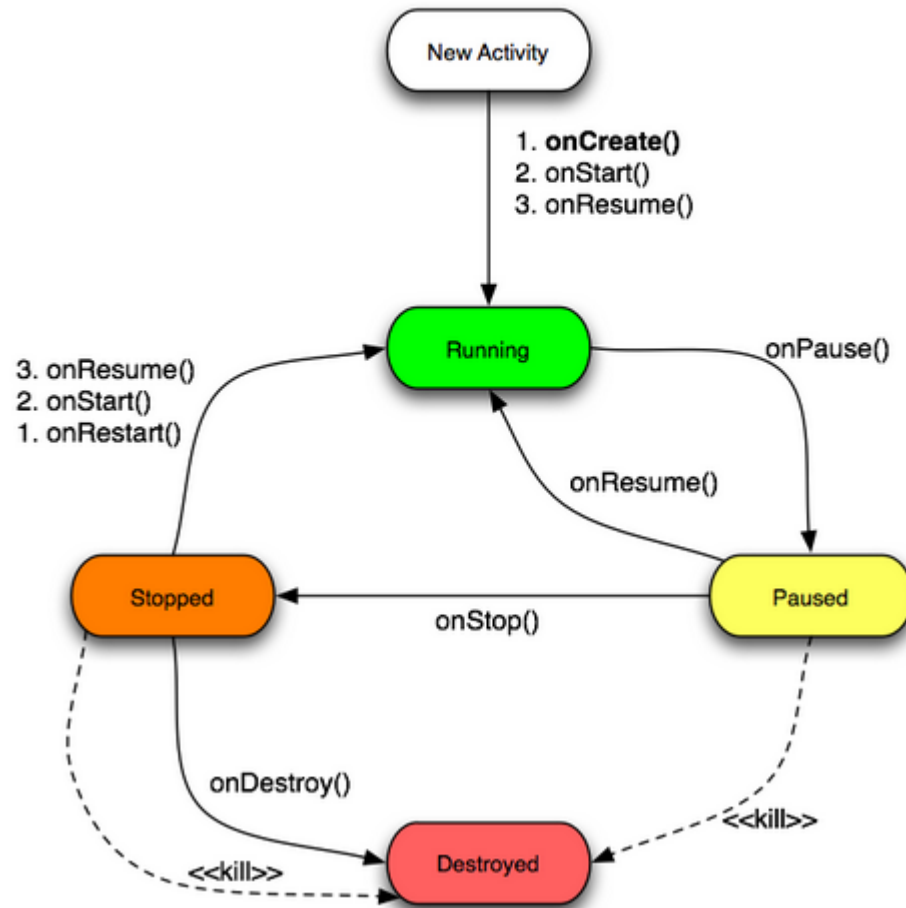
Patch 만들기 , 사용법

- `#diff -urN SRC DES소스 > xxx.diff`
- `#cat xxx.diff | patch -p1`
- <http://crztech.ipstime.org:8080/Release/mango100/eclair/android-menu-patch.tar> 다운받기
- `# cd mango100_eclair_2010_06_30`
- `# cat ../../mango100-eclair-menu.diff | patch -p1`
- `# cp sound_change_reload.png
frameworks/base/core/res/res/drawable/`
- `#cp sound_change_spdif.png
frameworks/base/core/res/res/drawable/`
- `# cp sound_change_wm8960.png
frameworks/base/core/res/res/drawable/`

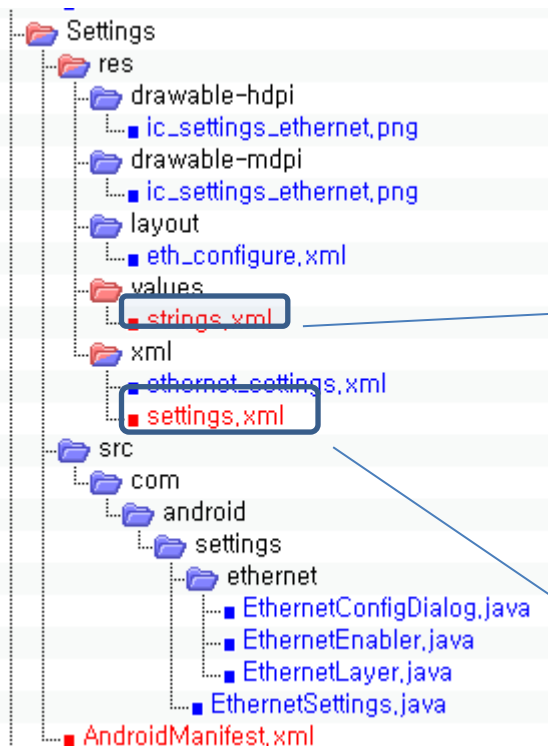
안드로이드 이더넷 메뉴추가

Android Activity Lifecycle

Activity Lifecycle



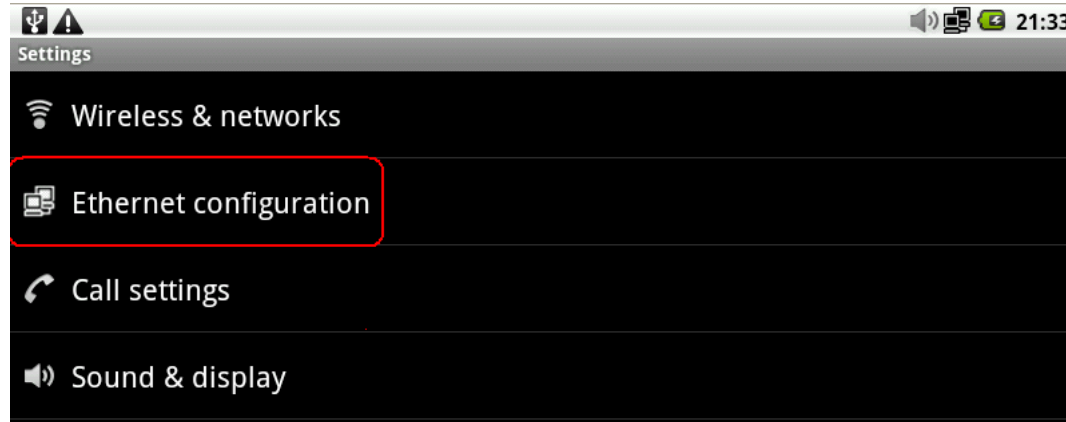
안드로이드 Ethernet Config 추가



```
<!-- Ethernet configuration dialog -->
<string name="eth_config_title">Configure Ethernet device</string>
<string name="eth_setting">Ethernet setting</string>
<string name="eth_dev_list">Ethernet Devices:</string>
<string name="eth_con_type">Connection Type</string>
<string name="eth_con_type_dhcp">Dhcp</string>
<string name="eth_con_type_manual">Static IP</string>
<string name="eth_dns">DNS address</string>
<string name="eth_gw">Default Router</string>
<string name="eth_ipaddr">IP address</string>
<string name="eth_quick_toggle_title">Ethernet</string>
<string name="eth_quick_toggle_summary">Turn on Ethernet</string>
<string name="eth_conf_save">Save</string>
<string name="eth_conf_cancel">Cancel</string>
<string name="eth_radio_ctrl_title">Ethernet configuration</string>
<string name="eth_radio_ctrl_summary">Configure Ethernet devices</string>
<string name="eth_conf_perf_title">Ethernet configuration</string>
<string name="eth_conf_summary">Configure Ethernet devices</string>
<string name="eth_mask">Netmask</string>
<string name="eth_toggle_summary_off">Turn off Ethernet</string>
<string name="eth_toggle_summary_on">Turn on Ethernet</string>
```

```
<com.android.settings.IconPreferenceScreen
    android:title="@string/eth_radio_ctrl_title"
    settings:icon="@drawable/ic_settings_ethernet">
    <intent
        android:action="android.intent.action.MAIN"
        android:targetPackage="com.android.settings"
        android:targetClass="com.android.settings.EthernetSettings" />
</com.android.settings.IconPreferenceScreen>
```


안드로이드 Ethernet Config 추가



```
<com.android.settings.IconPreferenceScreen
  android:title="@string/eth_radio_ctrl_title"
  settings:icon="@drawable/ic_settings_ethernet">
  <intent
    android:action="android.intent.action.MAIN"
    android:targetPackage="com.android.settings"
    android:targetClass="com.android.settings.EthernetSettings" />
</com.android.settings.IconPreferenceScreen>
```

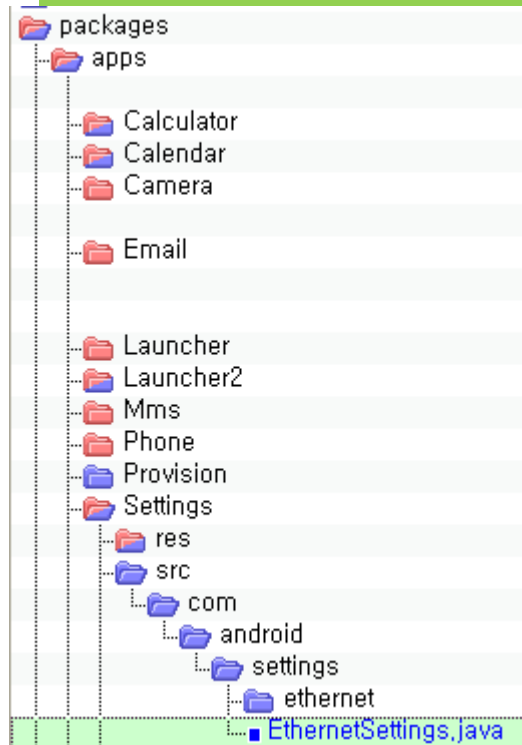
Setting.xml 과 IconPreferenceScreen

```
<com.android.settings.IconPreferenceScreen
    android:title="@string/eth_radio_ctrl_title"
    settings:icon="@drawable/ic_settings_ethernet">
    <intent
        android:action="android.intent.action.MAIN"
        android:targetPackage="com.android.settings"

android:targetClass="com.android.settings.EthernetSettings" />
    </com.android.settings.IconPreferenceScreen>
```

XML과 EthernetSettings class관계

```
<com.android.settings.IconPreferenceScreen
    android:title="@string/eth_radio_ctrl_title"
    settings:icon="@drawable/ic_settings_ethernet" >
<intent
    android:action="android.intent.action.MAIN"
    android:targetPackage="com.android.settings"
```




```
:"com.android.settings.EthernetSettings" />
settings.IconPreferenceScreen
```


```
public class EthernetSettings extends PreferenceActivity {
    private static final String KEY_TOGGLE_ETH = "toggle_eth";
    private static final String KEY_CONF_ETH = "eth_config";
    private EthernetEnabler mEthEnabler;
    private EthernetConfigDialog mEthConfigDialog;
    private Preference mEthConfigPref;
```

Ethernet Service 초기화

```
(EthernetManager) getSystemService(ETH_SERVICE),  
./packages/apps/Settings/src/com/android/settings/EthernetSettings.j  
ava
```



```
}else if (ETH_SERVICE.equals(name)) {  
    return getEthernetManager();  
./frameworks/base/core/java/android/app/ApplicationContext.java
```



```
{  
    synchronized (sSync) {  
        if (sEthManager == null) {  
            IBinder b = ServiceManager.getService(ETH_SERVICE);  
            IEthernetManager service =  
            IEthernetManager.Stub.asInterface(b);  
            sEthManager = new EthernetManager(service,  
            mMainThread.getHandler());
```

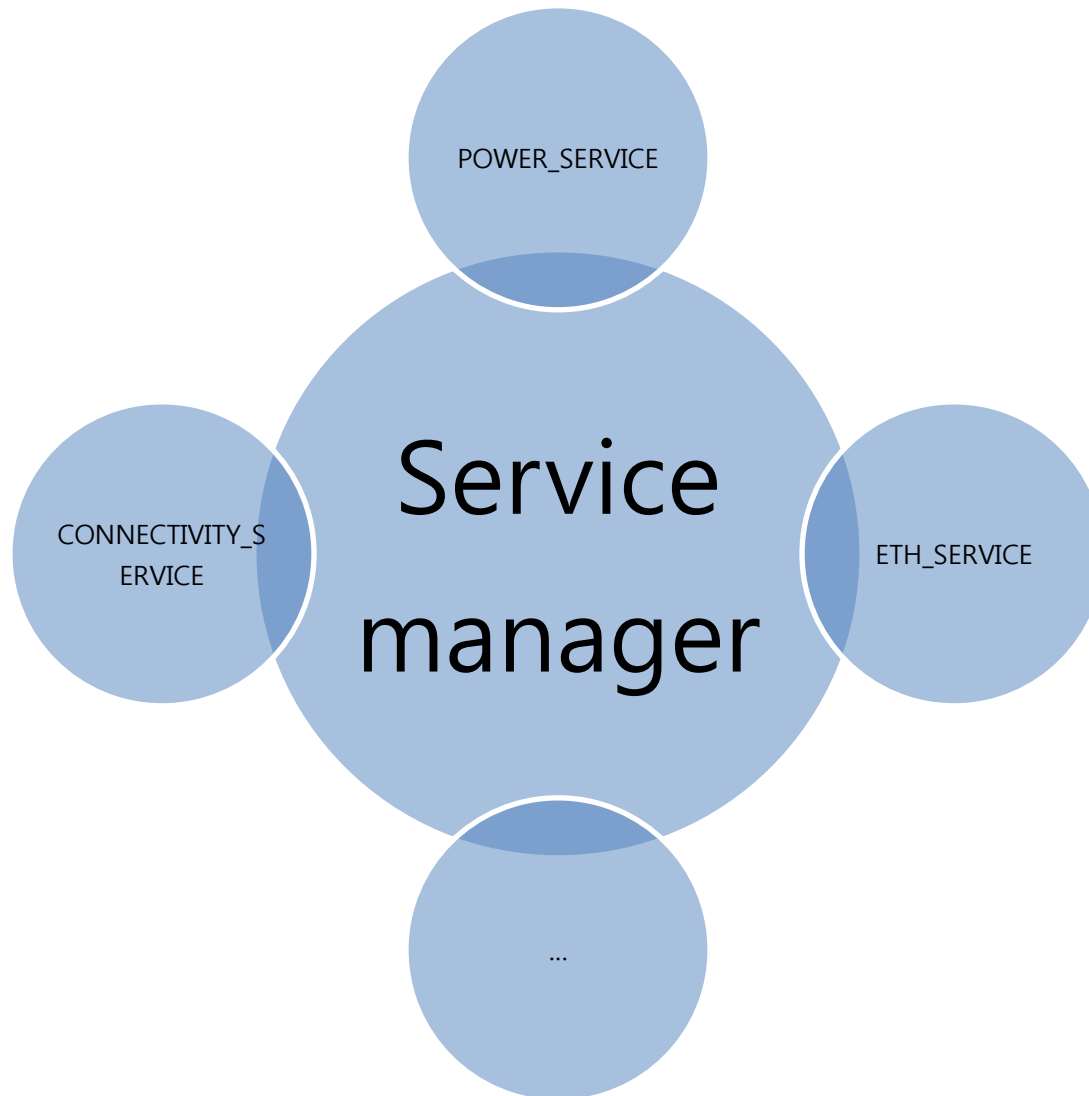
Service Manager Ethernet 등록

```
{
    synchronized (sSync) {
        if (sEthManager == null) {
            IBinder b = ServiceManager.getService(ETH_SERVICE);
            IEthernetManager service =
            IEthernetManager.Stub.asInterface(b);
            sEthManager = new EthernetManager(service,
            mMainThread.getHandler());
        }
    }
}
```

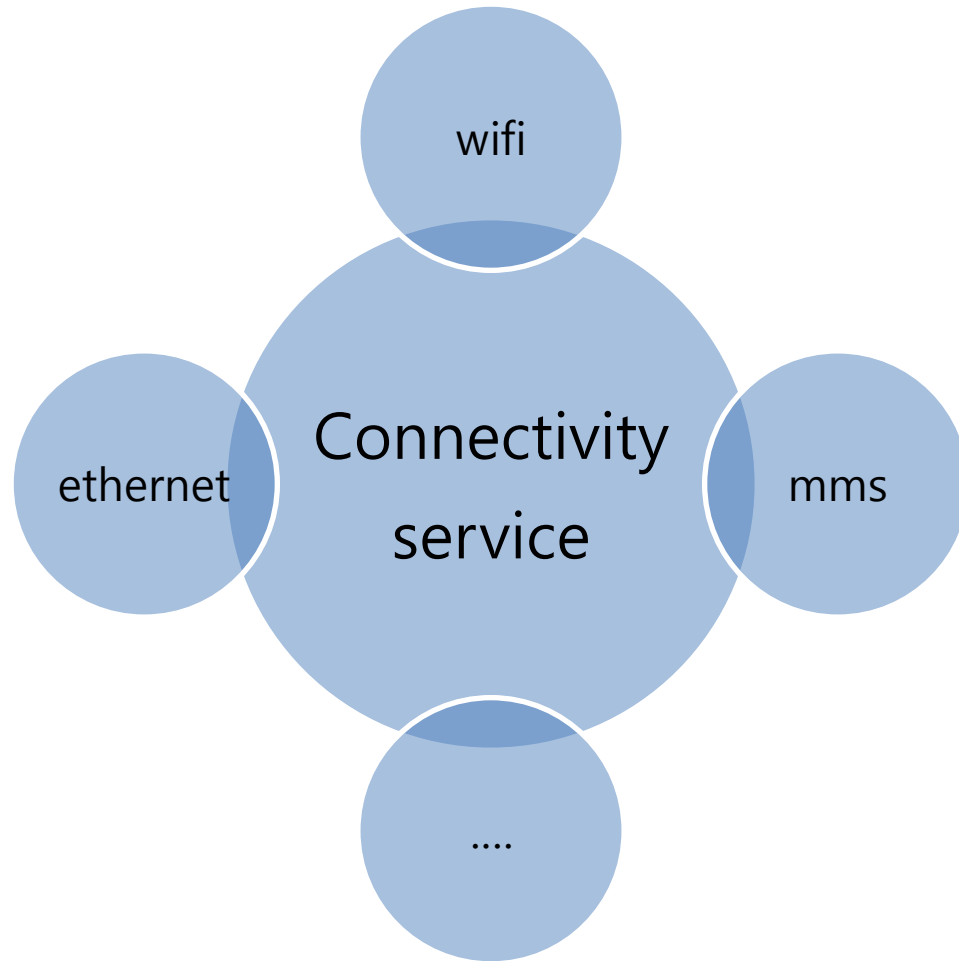
Connectivity service에서 등록

```
private ConnectivityService(Context context) {
    ....
    ServiceManager.addService(Context.ETH_SERVICE, ethService);
    ./frameworks/base/services/java/com/android/server/ConnectivityService.java
```

Service Manager 역할

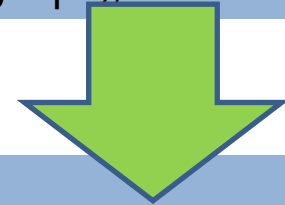


Connectivity Service



Ethernet service (부팅 시)

```
private ConnectivityService(Context context) {  
    if (DBG) Log.v(TAG, "ConnectivityService starting up");
```



```
if (DBG) Log.v(TAG, "Starting Ethernet Service");  
    mEthernetStateTracker = new EthernetStateTracker(context, mHandler);  
    EthernetService ethService = new EthernetService(context,  
                                                    mEthernetStateTracker);  
    ServiceManager.addService(Context.ETH_SERVICE, ethService);  
    mNetTrackers[ConnectivityManager.TYPE_ETH] = mEthernetStateTracker;
```

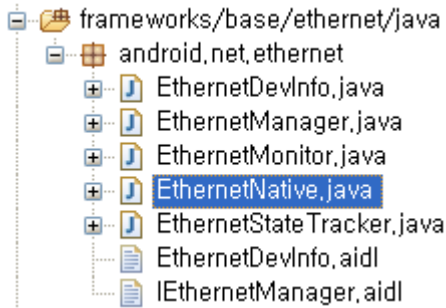


Ethernet service (부팅 시)

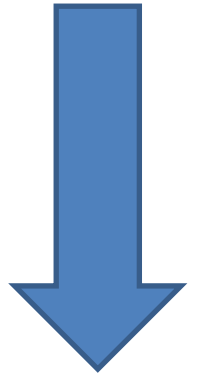


super는 자손클래스에서 조상클래스로부터 상속받은 멤버를 참조하는데 사용되는 참조변수

```
public EthernetStateTracker(Context context, Handler target) {  
    super(context, target, ConnectivityManager.TYPE_ETH, 0,  
        "ETH", "");  
    Log.i(TAG, "Starts...");  
    if(EthernetNative.initEthernetNative() != 0 )  
    {  
        Log.e(TAG, "Can not init ethernet device layers");  
        return;  
    }  
}
```



```
package android.net.ethernet;  
  
public class EthernetNative {  
    public native static String getInterfaceName(int i);  
    public native static int getInterfaceCnt();  
    public native static int initEthernetNative();  
    public native static String waitForEvent();  
}
```



Ethernet service (부팅시)



```
static JNINativeMethod gEthernetMethods[] = {  
    {"waitForEvent", "()Ljava/lang/String;", (void *)android_net_ethernet_waitForEvent},  
    {"getInterfaceName", "(I)Ljava/lang/String;",  
void )android_net_ethernet_getInterfaceName},  
    {"initEthernetNative", "()I", (void *)android_net_ethernet_initEthernetNative},  
    {"getInterfaceCnt", "()I", (void *)android_net_ethernet_getInterfaceCnt}  
};
```



```
static jint android_net_ethernet_initEthernetNative(JNIEnv *env,  
                                                    jobject clazz)  
{  
    if ((ret = netlink_init_interfaces_list()) < 0) {
```



Ethernet service (부팅 시)



```
static int netlink_init_interfaces_list(void) {
```

```
..  
if ((netdir = opendir(SYSFS_CLASS_NET)) != NULL) {  
while((de = readdir(netdir))!=NULL) {
```

```
# cd /sys/class/net/  
# ls  
eth0 lo
```

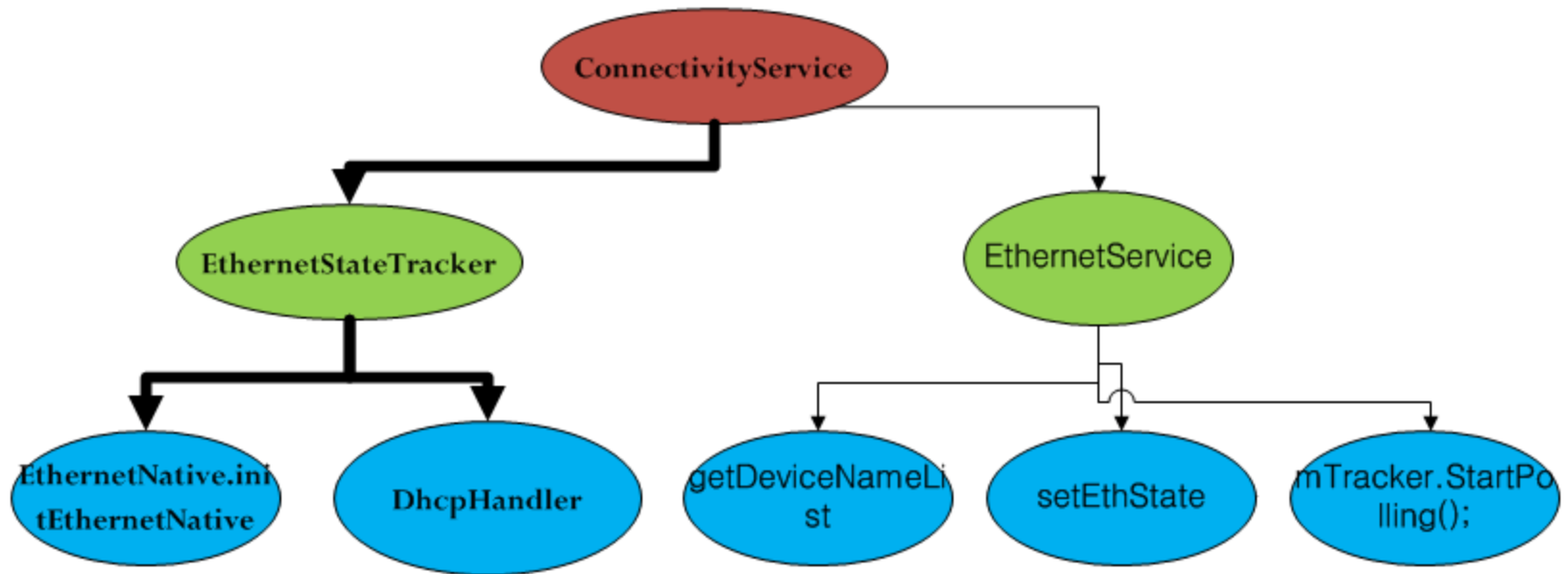
```
static const char SYSFS_CLASS_NET[] =  
"/sys/class/net";
```

```
snprintf(path, SYSFS_PATH_MAX, "%s/%s/ifindex", SYSFS_CLASS_NET, de->d_name);
```

```
if ((ifidx = fopen(path, "r")) != NULL ) {
```


```
/sys/class/net/eth0  
# cat ifindex  
2  
# cat dev_id  
0x0  
# cat uevent  
INTERFACE=eth0  
IFINDEX=2
```

Ethernet service (부팅 시)



Ethernet Service 등록 Flow

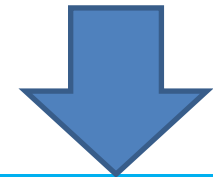
```
private ConnectivityService(Context context) {  
    ...  
    if (DBG) Log.v(TAG, "Starting Ethernet Service");  
        mEthernetStateTracker = new  
EthernetStateTracker(context,mHandler);  
        EthernetService ethService = new EthernetService(context,  
mEthernetStateTracker)
```



```
public EthernetService(Context context, EthernetStateTracker Tracker){  
    mTracker = Tracker;  
    mContext = context;  
    isEthEnabled = getPersistedState();  
    Log.i(TAG,"Ethernet dev enabled " + isEthEnabled );  
    getDeviceNameList();  
    setEthState(isEthEnabled);  
    Log.i(TAG, "Trigger the ethernet monitor");  
    mTracker.StartPolling();  
    }  
}
```

Ethernet Service 등록 Flow(계속)

```
public EthernetService(Context context, EthernetStateTracker Tracker){  
...  
    getDeviceNameList();  
}
```



```
public String[] getDeviceNameList() {  
    if (scanEthDevice() > 0 )  
        return DevName;  
    else  
        return null;  
}
```




Ethernet Service 등록 Flow(계속)

```
private int scanEthDevice() {
    int i = 0;
    if ((i = EthernetNative.getInterfaceCnt()) != 0) {
        Log.i(TAG, "total found "+i+ " net devices");
        DevName = new String[i];
    }
    else
        return i;


    for (j = 0; j < i; j++) {
        DevName[j] =
EthernetNative.getInterfaceName(j);
        if (DevName[j] == null)
            break;
        Log.i(TAG,"device
    }

    return i;
}
```



```
static JNINativeMethod gEthernetMethods[] = {
    {"waitForEvent", "()Ljava/lang/String;",
    (void *)android_net_ethernet_waitForEvent},
    {"getInterfaceName", "(I)Ljava/lang/String;",
    (void *)android_net_ethernet_getInterfaceName},
    {"initEthernetNative", "()I",
    (void *)android_net_ethernet_initEthernetNative},
    {"getInterfaceCnt", "()I",
    (void *)android_net_ethernet_getInterfaceCnt}
};
```

```
static JNINativeMethod gEthernetMethods[] = {
    {"waitForEvent", "()Ljava/lang/String;",
    (void *)android_net_ethernet_waitForEvent},
    {"getInterfaceName", "(I)Ljava/lang/String;",
    (void *)android_net_ethernet_getInterfaceName},
    {"initEthernetNative", "()I",
    (void *)android_net_ethernet_initEthernetNative},
    {"getInterfaceCnt", "()I",
    (void *)android_net_ethernet_getInterfaceCnt}
};
```



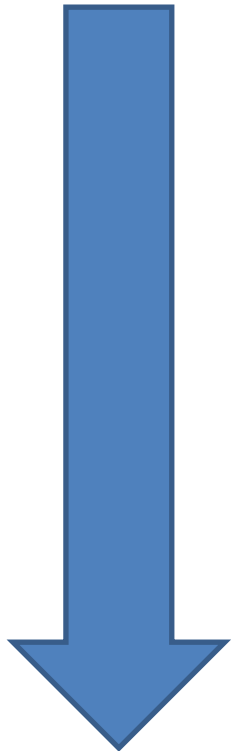
Ethernet Service 등록 Flow(계속)

```
static jint android_net_ethernet_getInterfaceCnt() {  
    return total_int;  
}
```

```
static jint  
android_net_ethernet_initEthernetNative(JNIEnv *env,  
                                           jobject  
                                           clazz)
```

netlink_init_interfaces_list()

```
LOGI("interface %s:%d found", intfinfo->  
      name, intfinfo->i);  
add_int_to_list(intfinfo); //여기서 total_int  
++
```



Ethernet Service 등록 Flow(계속)

```
static jstring android_net_ethernet_getInterfaceName(JNIEnv *env,
object clazz, jint index) {
    info = interfaces;
    if (total_int != 0 && index <= (total_int - 1)) {
        while (info != NULL) {
            if (index == i) {
                LOGI("Found :%s",info->name);
                return env->NewStringUTF(info->name);
            }
            info = info->next;
        }
    }
}
```

```
static jint
android_net_ethernet_initEthernetNative(JNIEnv *env,
object
clazz)
```

netlink_init_interfaces_list()

```
static void add_int_to_list(interface_info_t *node) {
    /*
    *Todo: Lock here!!!!
    */
    node->next = interfaces;
    interfaces = node;
    total_int ++;
}
```

```
LOGI("Interface %s:%d found",intfinfo->name,intfinfo->i);
add_int_to_list(intfinfo); //에서 Node 생
```

서

Ethernet Event 처리 쓰레드 생성

```
public EthernetService(Context context, EthernetStateTracker Tracker){
```

```
    Log.i(TAG, "Trigger the ethernet monitor");
```

```
    mTracker.StartPolling();
```

쓰레드 생성

```
class MonitorThread extends Thread {
```

```
    public MonitorThread() {  
        super("EthMonitor");  
    }
```

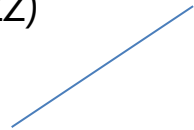
```
    public void run() {  
        int index;  
        int i;
```

```
        //noinspection InfiniteLoopStatement  
        for (;;) {  
            Log.i(TAG, "go poll events");
```

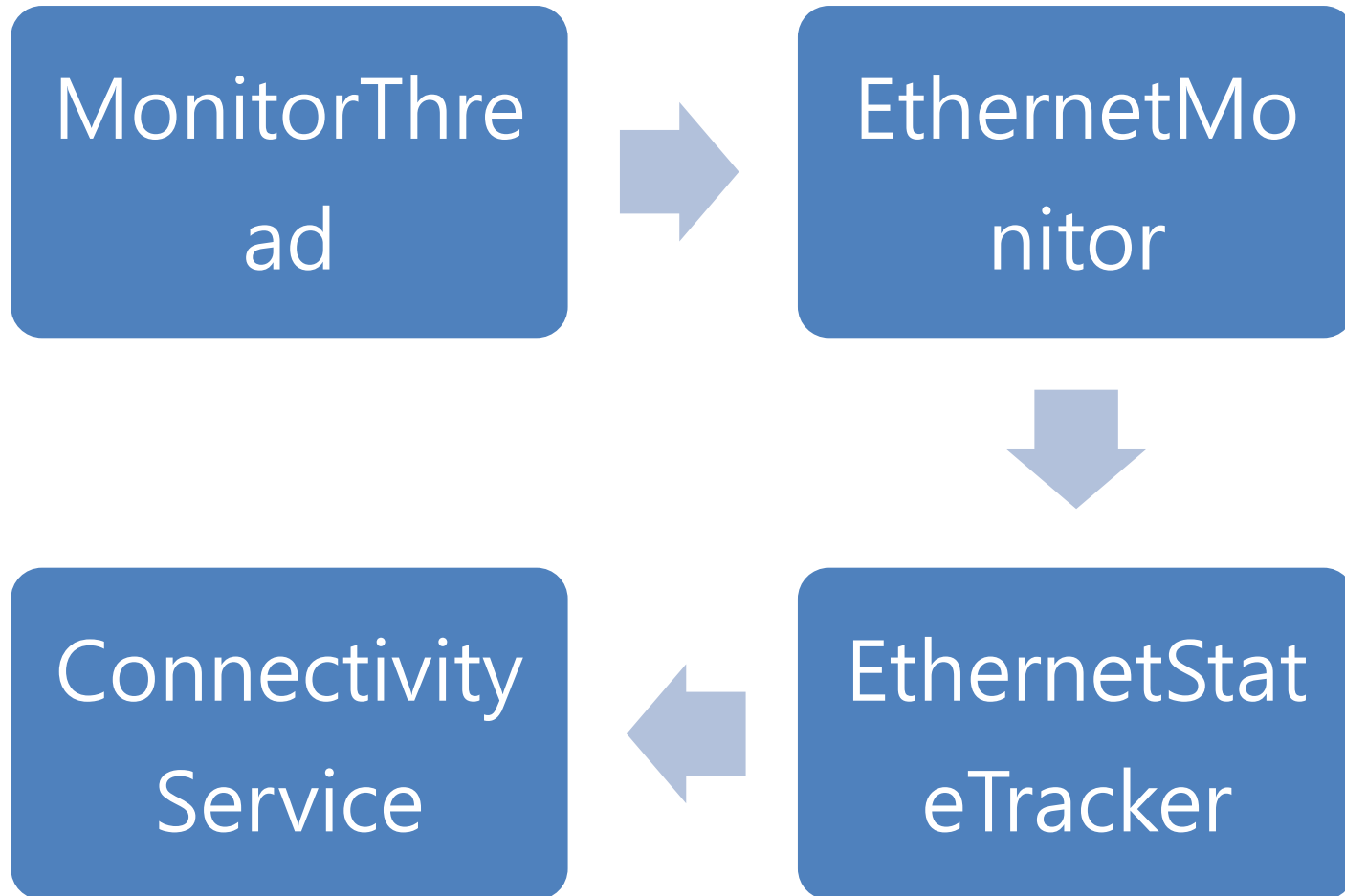
```
            String eventName = EthernetNative.waitForEvent();
```

```
static jstring  
android_net_ethernet_waitForEvent(JNIEnv  
*env,
```

object clazz)
{



이더넷이 연결된 경우(DHCP)



이더넷이 연결된 경우(DHCP)

```
static jstring android_net_ethernet_waitForEvent(JNIEnv *env,  
                                                jobject clazz)  
{  
    if((len = recvmsg(nl_socket_poll, &msg, 0)) >= 0) {
```

MonitorThread.run

```
.  
} else if (cmd == NEW_LINK) {  
    event = PHYUP;  
    handleEvent(events[i], event);
```

이더넷이 연결된 경우(DHCP)

```
void handleEvent(String ifname,int event) {  
    switch (event) {  
    case PHYUP:  
        mTracker.notifyPhyConnected(ifname);  
        break;  
    }
```

```
public void notifyPhyConnected(String ifname) {  
    if synchronized(this) {  
        this.sendMessage(EVENT_HW_PHYCONNECTED);  
    }
```

MonitorThread

이더넷이 연결된 경우(DHCP)

```
public void handleMessage(Message msg) {  
    ..  
    case EVENT_HW_PHYCONNECTED:  
    try {  
        configureInterface(info);  
    }  
}
```

```
private boolean configureInterface(EthernetDevInfo info) throws  
UnknownHostException {  
    if  
    (info.getConnectMode().equals(EthernetDevInfo.ETH_CONN_MODE  
_DHCP)) {  
        mDhcpTarget.sendMessage(EVENT_DHCP_START,
```

DHCP
Handler

이더넷이 연결된 경우(DHCP)

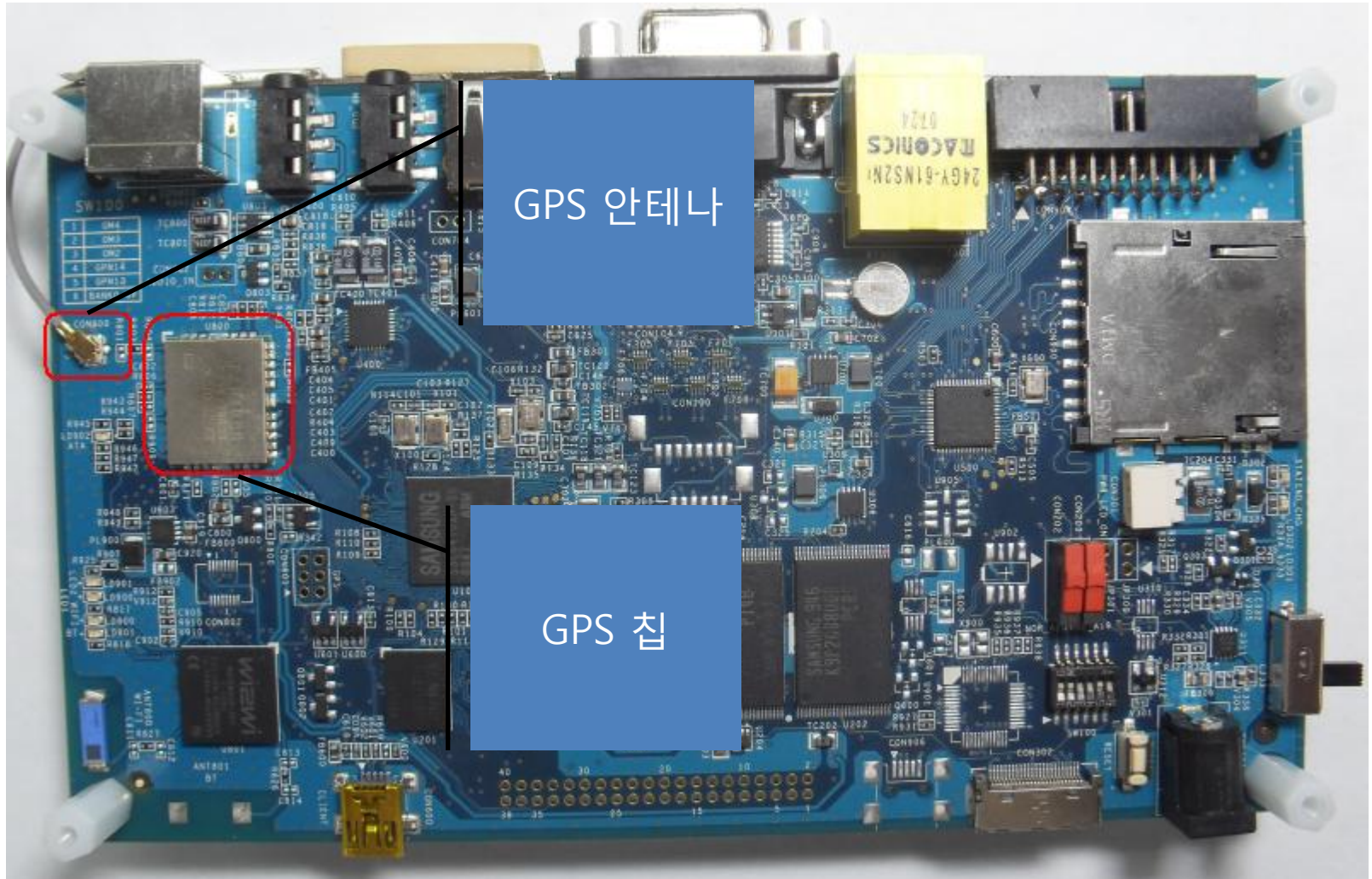
```
public void handleMessage(Message msg) {  
    ..  
    switch (msg.what) {  
    case EVENT_DHCP_START:  
        if (NetworkUtils.runDhcp(mInterfaceName, mDhcpInfo)) {}  
    }  
}
```

```
static JNINativeMethod gNetworkUtilMethods[] = {  
    /* name, signature, funcPtr */  
    { "enableInterface", "(Ljava/lang/String;)I", (void *)android_net_utils_enableInterface },  
    { "disableInterface", "(Ljava/lang/String;)I", (void *)android_net_utils_disableInterface },  
    { "addHostRoute", "(Ljava/lang/String;I)I", (void *)android_net_utils_addHostRoute },  
    { "removeHostRoutes", "(Ljava/lang/String;)I", (void *)android_net_utils_removeHostRoutes },  
    { "setDefaultRoute", "(Ljava/lang/String;I)I", (void *)android_net_utils_setDefaultRoute },  
    { "getDefaultRoute", "(Ljava/lang/String;)I", (void *)android_net_utils_getDefaultRoute },  
    { "removeDefaultRoute", "(Ljava/lang/String;)I", (void *)android_net_utils_removeDefaultRoute },  
    { "resetConnections", "(Ljava/lang/String;)I", (void *)android_net_utils_resetConnections },  
    { "runDhcp", "(Ljava/lang/String;Landroid/net/DhcpInfo;)Z", (void *)android_net_utils_runDhcp },  
    { "stopDhcp", "(Ljava/lang/String;)Z", (void *)android_net_utils_stopDhcp },  
    { "releaseDhcpLease", "(Ljava/lang/String;)Z", (void *)android_net_utils_releaseDhcpLease },  
    { "configureNative", "(Ljava/lang/String;IIII)Z", (void *)android_net_utils_configureInterface },  
    { "getDhcpError", "(Ljava/lang/String;)", (void *)android_net_utils_getDhcpError },  
};
```

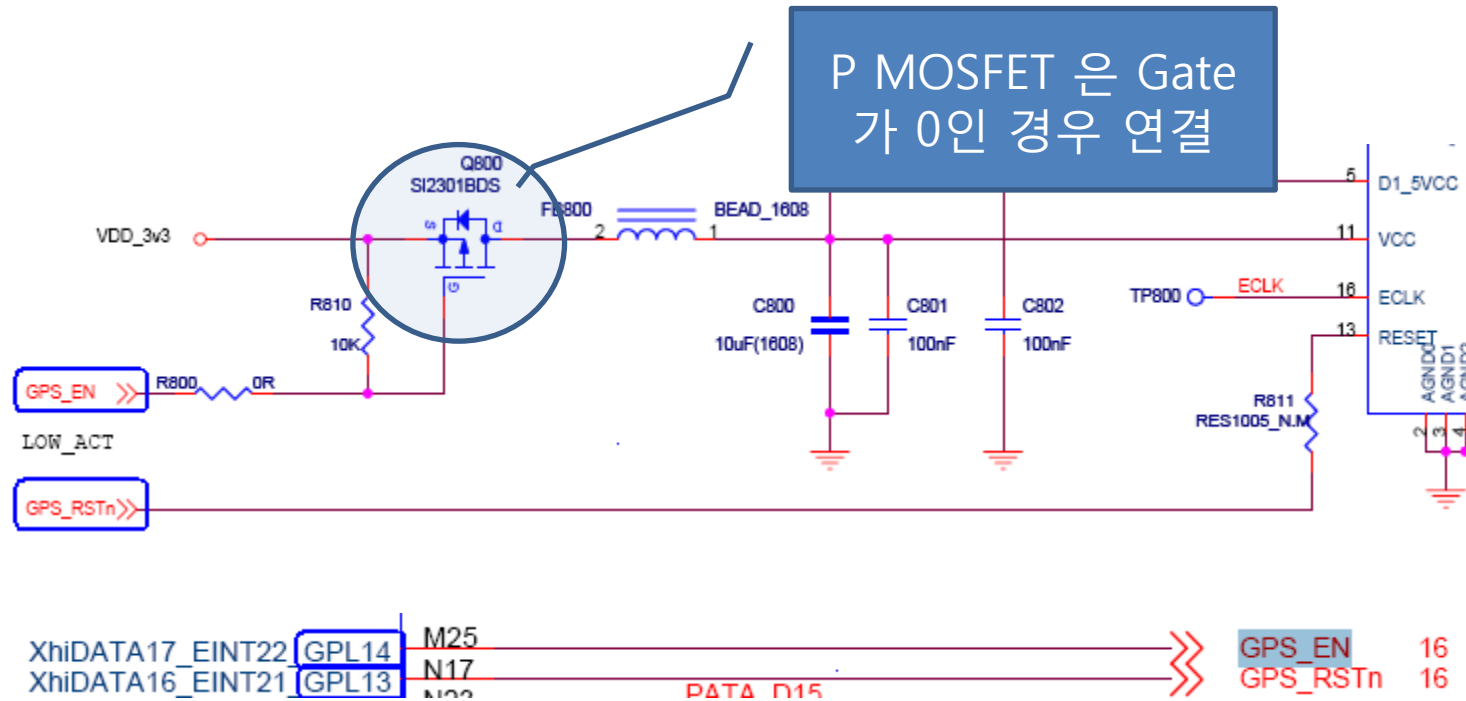
```
static jboolean android_net_utils_runDhcp(JNIEnv* env, jobject  
clazz, jstring ifname, jobject info)  
{  
    result = ::dhcp_do_request(nameStr, &ipaddr, &gateway,  
&mask,
```

```
&dns1, &dns2, &server, &lease):
```

안드로이드 GPS(Mango64)



안드로이드 GPS(Mango64)



팁: <http://ko.wikipedia.org/wiki/MOSFET>

안드로이드 GPS(Mango64)

```
.config - Linux Kernel v2.6.29 Configuration
----- MANGO6410 GPS setup -----
Arrow keys navigate the menu.  <Enter> selects submenus --->.
Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes,
<M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </>
for Search.  Legend: [+] built-in [ ] excluded <M> module < >

[+] Use GPS

<Select>  < Exit >  < Help >
```

“system type->MANGO6410 GPS Setup ”에서 설정

안드로이드 GPS(Mango64)

arch/arm/mach-s3c6410/Kconfig 파일에 아래 내용 추가

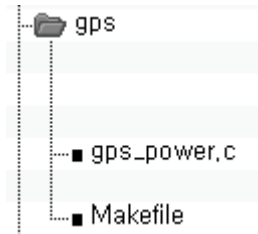
```
menu "MANGO6410 GPS setup"
    depends on MACH_MANGO6410

config MANGO6410_GPS
    bool "Use GPS"
    depends on MACH_MANGO6410
    help
        Use GPS Module

endmenu
```

안드로이드 GPS(Mango64)

커널 소스에
Driver폴더에 gps폴더추가 후 gps_power.c 파일,
Makefile 파일 추가



obj-\$(CONFIG_MANGO6410_GPS) += gps/
위의 내용을 driver 폴더에 Makefile에 추가

안드로이드 GPS(Mango64)

```
ssize_t gps_write(struct file *filp, const char *buf, size_t count, loff_t *off_what)
{
...
    if(*tmp == '1') { // power on
//      printk(KERN_WARNING "gps power on processing\n");
        gpio_direction_output(S3C64XX_GPL(14), 0);
        gpio_direction_output(S3C64XX_GPL(13), 1);
        gpio_set_value(S3C64XX_GPL(14), 0);
        gpio_set_value(S3C64XX_GPL(13), 0);
        udelay(10);
        gpio_set_value(S3C64XX_GPL(13), 1);
        udelay(10);
    } else { // power off
//      printk(KERN_WARNING "gps power off processing\n");
        gpio_set_value(S3C64XX_GPL(14), 1);
        gpio_set_value(S3C64XX_GPL(13), 0);
    }

    kfree(tmp);
    return count;
}
```

XhiDATA17_EINT22 GPL14
XhiDATA16_EINT21 GPL13

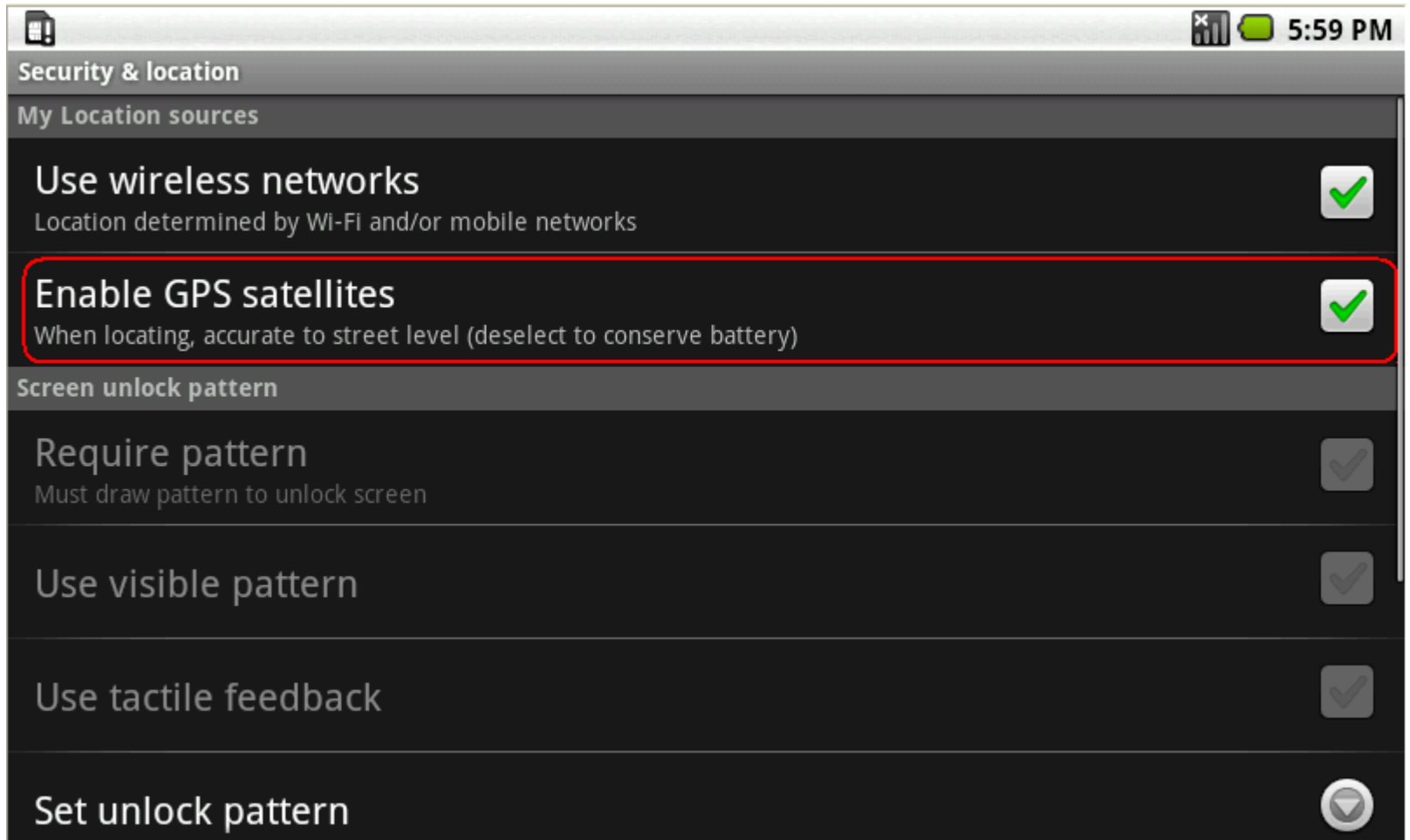
M25
N17
M22

DATA 015




GPS_EN 16
GPS_RSTn 16

안드로이드 GPS(Mango64)




안드로이드 GPS(Mango64)

```
mEnabled = native_init();  
private native boolean native_init();  
./frameworks/base/location/java/com/android/internal/location/GpsLocationProvider.  
java
```



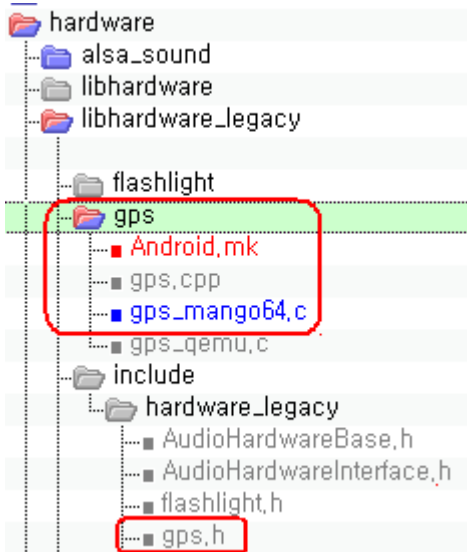
```
static jboolean android_location_GpsLocationProvider_init(JNIEnv* env, jobject obj)  
{  
    if (!sGpsInterface)  
        sGpsInterface = gps_get_interface();  
    return (sGpsInterface && sGpsInterface->init(&sGpsCallbacks) == 0);  
}  
frameworks/base/core/jni/android_location_GpsLocationProvider.cpp
```



```
const GpsInterface* gps_get_interface()  
{    if (sGpsInterface == NULL)        gps_find_hardware();    return sGpsInterface;}  
hardware/libhardware_legacy/gps/gps.cpp
```


안드로이드 GPS(Mango64)

•gps_mango64.c 파일 추가



```
BOARD_HAVE_GPS := true
HAVE_GPS_HARDWARE := true
build\target\board\generic\BoardConfig.mk 에
추가
```

```
# Use hardware GPS implementation if
USE_GPS_HARDWARE is set.
#
ifeq ($(HAVE_GPS_HARDWARE),true)
    LOCAL_CFLAGS += -
    DHAVE_GPS_HARDWARE
    LOCAL_SRC_FILES += gps/gps_mango64.c
endif
```

```
LOCAL_SRC_FILES += gps/gps.cpp
hardware\libhardware_legacy\gps\Android.mk
파일 수정
```

안드로이드 GPS(Mango64)

```
static struct perms_devperms[] = {
    { "/dev/null",      0666,  AID_ROOT,    AID_ROOT,    0 },
    { "/dev/zero",     0666,  AID_ROOT,    AID_ROOT,    0 },
    { "/dev/full",     0666,  AID_ROOT,    AID_ROOT,    0 },
    { "/dev/ptmx",     0666,  AID_ROOT,    AID_ROOT,    0 },
    { "/dev/tty",      0666,  AID_ROOT,    AID_ROOT,    0 },
    { "/dev/random",   0666,  AID_ROOT,    AID_ROOT,    0 },
    { "/dev/urandom",  0666,  AID_ROOT,    AID_ROOT,    0 },
    { "/dev/ashmem",   0666,  AID_ROOT,    AID_ROOT,    0 },
    { "/dev/binder",   0666,  AID_ROOT,    AID_ROOT,    0 },
    /* mango64 GPS permission crazyboy*/
    { "/dev/s3c2410_serial0", 0666,  AID_SYSTEM,  AID_SYSTEM,  0 },
    { "/dev/mango64_gps",    0666,  AID_SYSTEM,  AID_SYSTEM,  0 },
}
```

system/core/init/devices.c 파일 수정

```
struct {
    const char *prefix;
    unsigned int uid;
} property_perms[] = {
    { "service.adb.root",    AID_SHELL },
    { "persist.sys.",        AID_SYSTEM },
    { "persist.service.",    AID_SYSTEM },
    { "kernel.android.gps",  AID_SYSTEM },
    { "gps.power_on",        AID_SYSTEM },
    { NULL, 0 }
};
```

./system/core/init/ property_service.c 파일 수정

```
static int check_perms(const char *name, unsigned int uid)
{
    int i;
    if (uid == 0)
        return 1;

    if (!strncmp(name, "ro.", 3))
        name += 3;

    for (i = 0; property_perms[i].prefix; i++) {
        int tmp;
        if (strncmp(property_perms[i].prefix, name,
                    strlen(property_perms[i].prefix)) == 0) {
            if (property_perms[i].uid == uid) {
                return 1;
            }
        }
    }

    return 0;
}
```

안드로이드 GPS(Mango64)

```
int device_init(void)
{
// GPS Power Device Driver
    make_device("/dev/mango64_gps", 0, 235, 0);//Node 생성
}
./system/core/init/devices.c에서 Node생성
```

```
static void gps_dev_power(int state)
{
    char prop[PROPERTY_VALUE_MAX];
    int fd;
    char cmd = '0';
    int ret;
// look for a kernel-provided device name
    if(property_set("gps.power_on","/dev/mango64_gps")<0) //NODE 이용
    {
        D("Set mango64 power");
        //return;
    }
    if (property_get("gps.power_on",prop,GPS_POWER_IF) == 0) {
        LOGE("no gps power interface name");
        return;
    }
    ../hardware/libhardware_legacy/gps/gps_mango64.c 에서 컨트롤
```

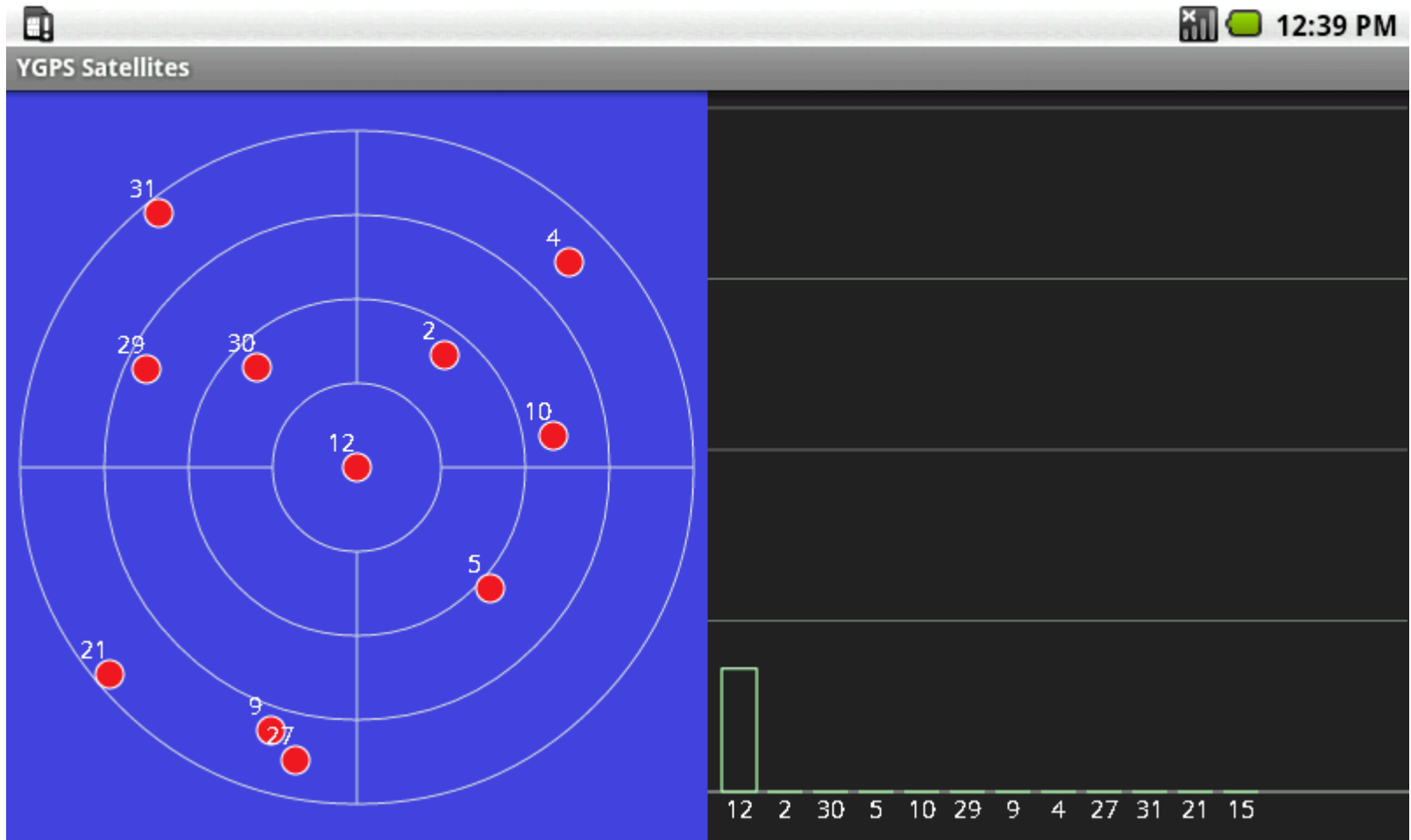
안드로이드 GPS(Mango64)

```
static void
gps_state_init( GpsState* state )

// look for a kernel-provided device name
if(property_set("ro.kernel.android.gps","s3c2410_serial0")<0)
/UART0
    {
        D("Set mango64 error");
        return;
    }
if (property_get("ro.kernel.android.gps",prop,"s3c2410_serial0")
== 0) {
    D("no kernel-provided gps device name");
    LOGE("ro.kernel.android.gps: '%s'", prop);
    return;
}
if ( snprintf(device, sizeof(device), "/dev/%s", prop) >=
(int)sizeof(device) ) {
    LOGE("gps serial device name too long: '%s'", prop);
    return;
}
```

NMEA 데이터를
/dev/s3c2410_serial0 통해서 받는다

안드로이드 GPS 실행 결과



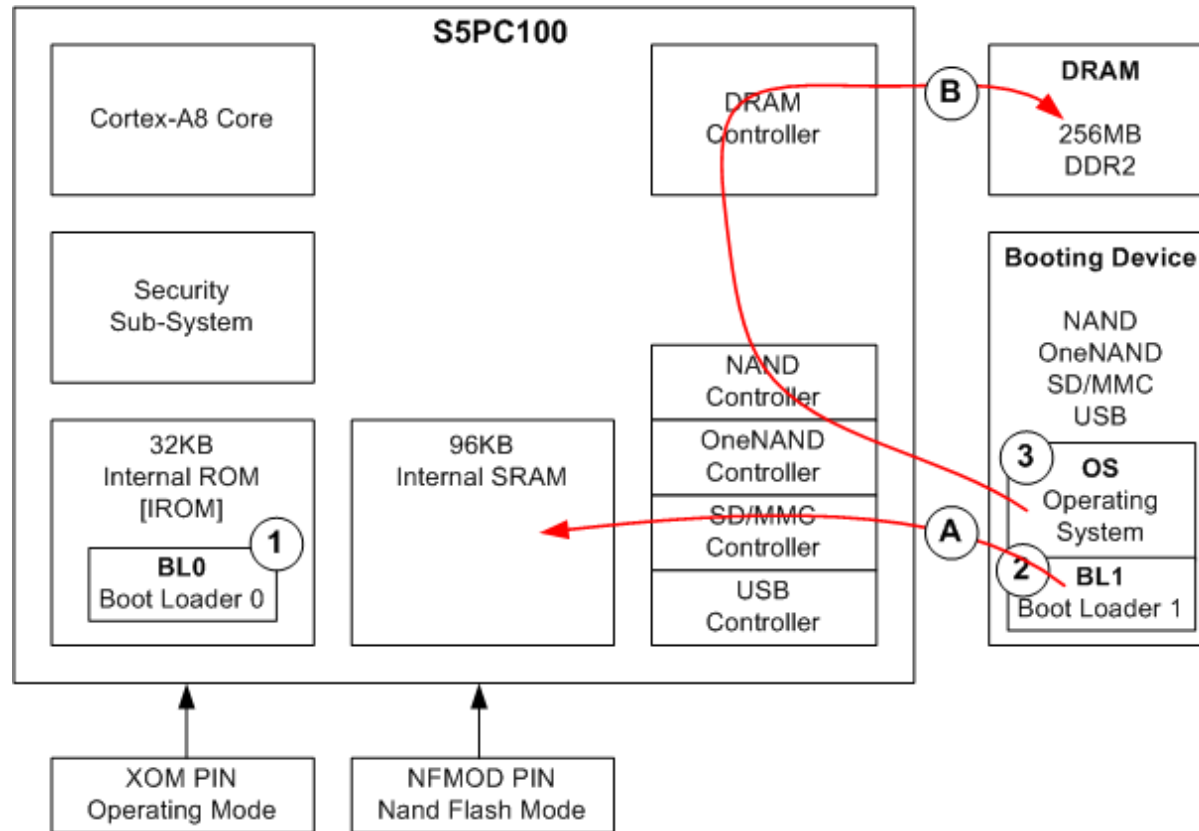
리눅스 커널 버전의 선택

- 커널 버전
 - 역사 : <http://www.linux.org/dist/kernel.html>
 - 버전 숫자 : X.Y.ZZ
 - X : 커널의 버전
 - Y : 릴리즈 번호, 홀수->개발 중, 짝수->안정된 버전
 - ZZ : Modifications, 사소한 변화를 의미
 - 최신 버전
 - 새로운 다양한 기능이 이미 추가되어 있음
 - 크기가 매우 크다는 단점이 있음
- 커널 버전의 선택
 - 임베디드 시스템의 크기를 고려
 - 필요한 기능을 고려
 - 확장성을 고려

안드로이드란?

- 운영체제와 미들웨어 그리고 핵심 애플리케이션을 포함하고 있는 모바일 디바이스를 위한 소프트웨어 스택
- 안드로이드 SDK는 Java 프로그래밍 언어를 사용하여 안드로이드 플랫폼상의 어플리케이션을 개발하기 위해 필요한 도구들과 API를 제공

망고100 부팅과정



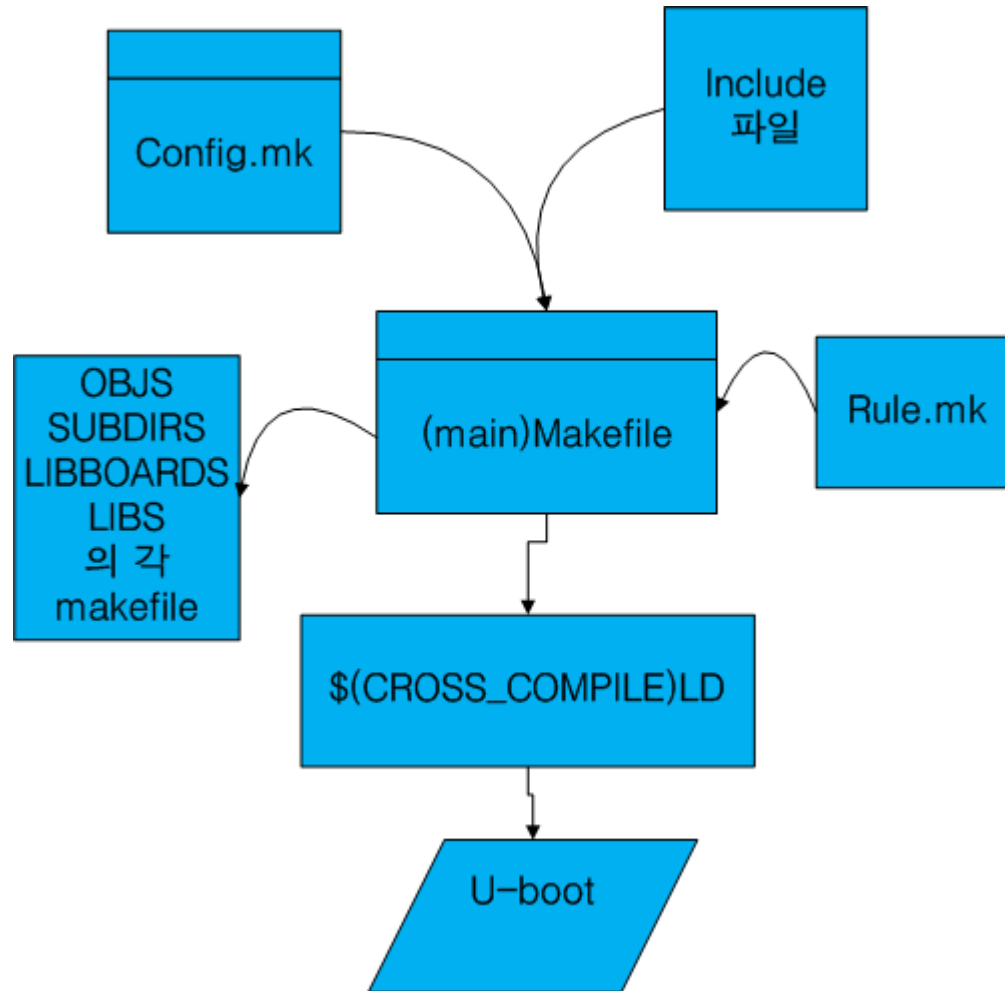
망고100 부팅모드

- USB Boot Mode
- Nand Boot Mode
- SD Boot Mode

개발 환경

- 리눅스 설치
- Toolchain 설치
- JDK설치
- TFTP 설정
- NFS 설정

U-boot Build 환경 구조도



U-boot build 실행 분석

#make 명령 실행

\$(TOPDIR)/config.mk
\$(TOPDIR)/arm_config.mk
\$(TOPDIR)/rules.mk
...

```
include $(obj)include/autoconf.mk.dep
```

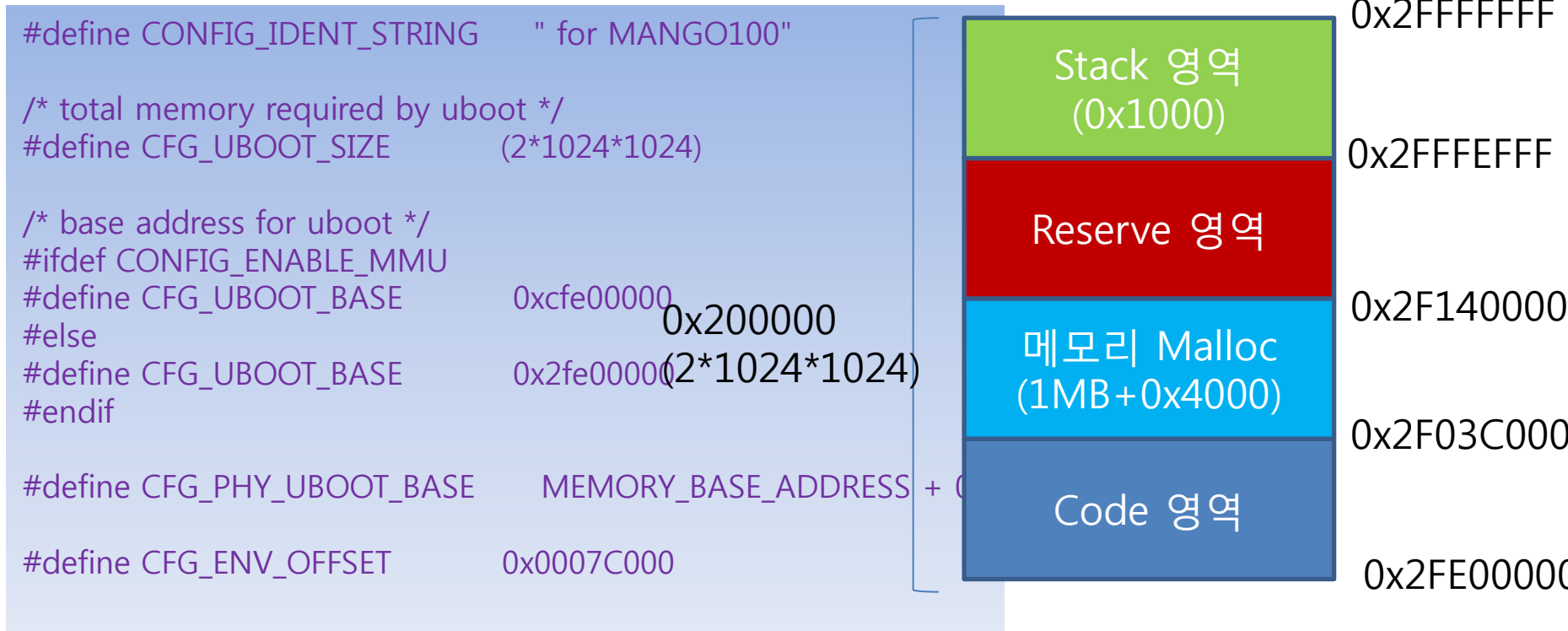
```
all: $(ALL)
```

```
ALL += $(obj)u-boot.srec $(obj)u-boot.bin $(obj)System.map $(U_BOOT_NAND) $(U_BOOT_ONENAND) $(obj)u-boot.dis  
ifeq ($(ARCH),blackfin)  
ALL += $(obj)u-boot.ldr  
endif
```

```
$(obj)u-boot.srec: $(obj)u-boot  
$(OBJCOPY) ${OBJCFLAGS} -O srec $< $@
```

```
$(obj)u-boot: depend $(SUBDIRS) $(OBJS) $(LIBBOARD) $(LIBS) $(LDSCRIPT)  
UNDEF_SYM=`$(OBJDUMP) -x $(LIBBOARD) $(LIBS) | \#  
sed -n -e 's/.*#$(SYM_PREFIX)__u_boot_cmd_.*#/-u#|/p'|sort|uniq`; \#  
cd $(LNDIR) && $(LD) $(LDFLAGS) $$UNDEF_SYM $(__OBJS) \#  
--start-group $(__LIBS) --end-group $(PLATFORM_LIBS) \#  
-Map u-boot.map -o u-boot
```

U-boot \$BOARDNAME.h 파일 분석



안드로이드 커널 Open Git 서버

- <http://android.git.kernel.org>

ANDROID

open source project

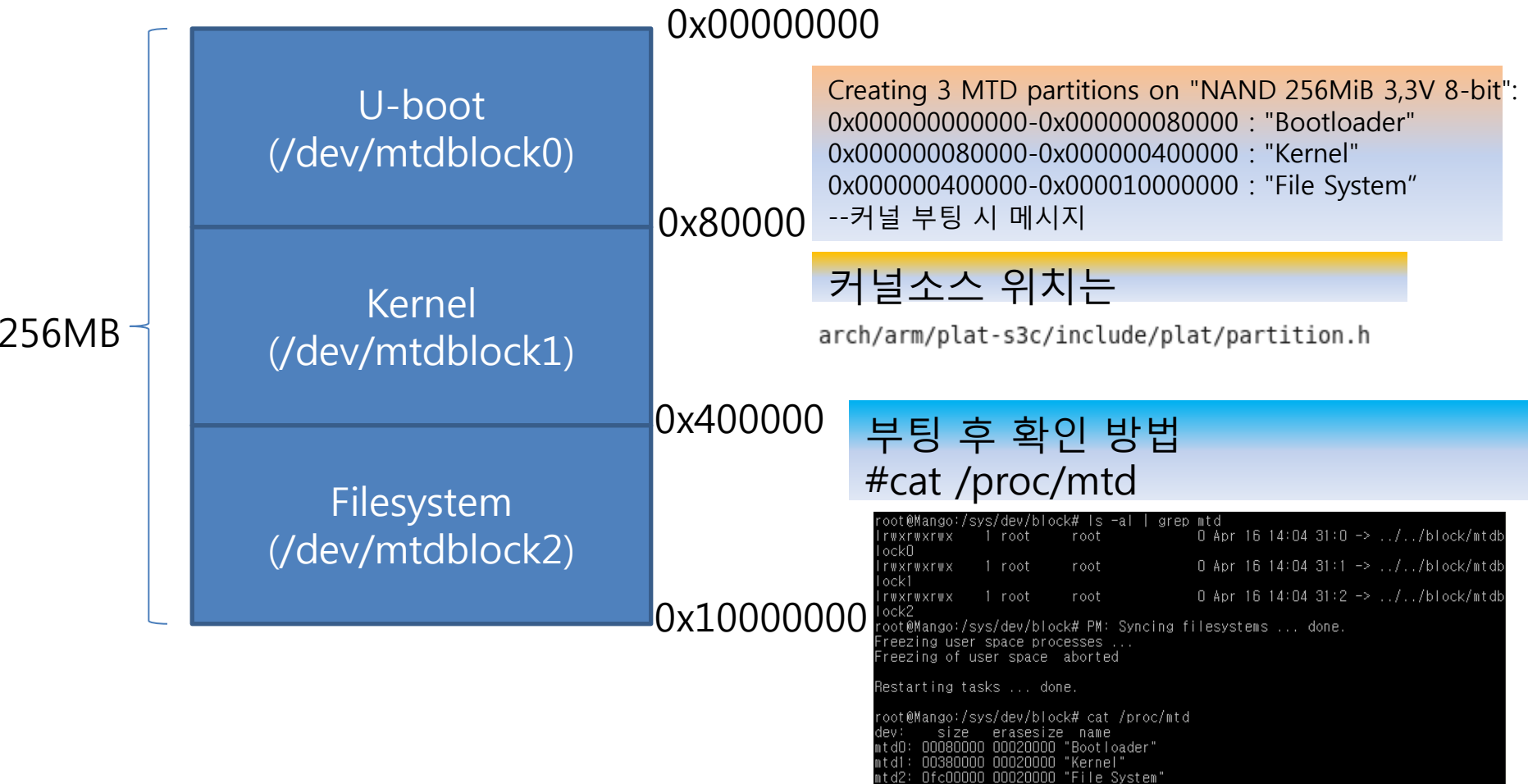
To clone one of these trees, install [git](#), and run:

```
git clone git://android.git.kernel.org/ + project path.
```

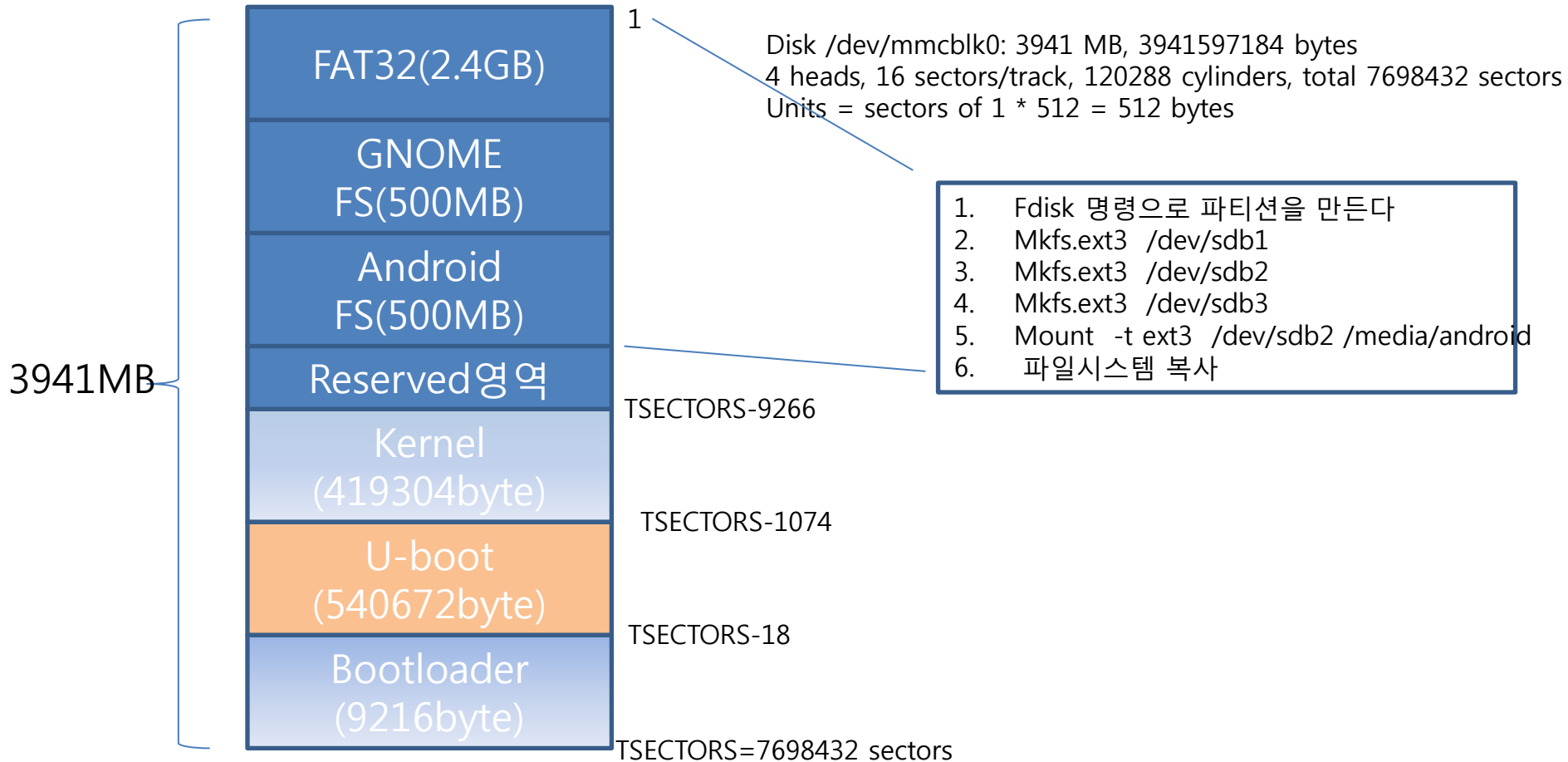
To clone the entire platform, install [repo](#), and run:

```
mkdir mydroid
cd mydroid
repo init -u git://android.git.kernel.org/platform/manifest.git
repo sync
```

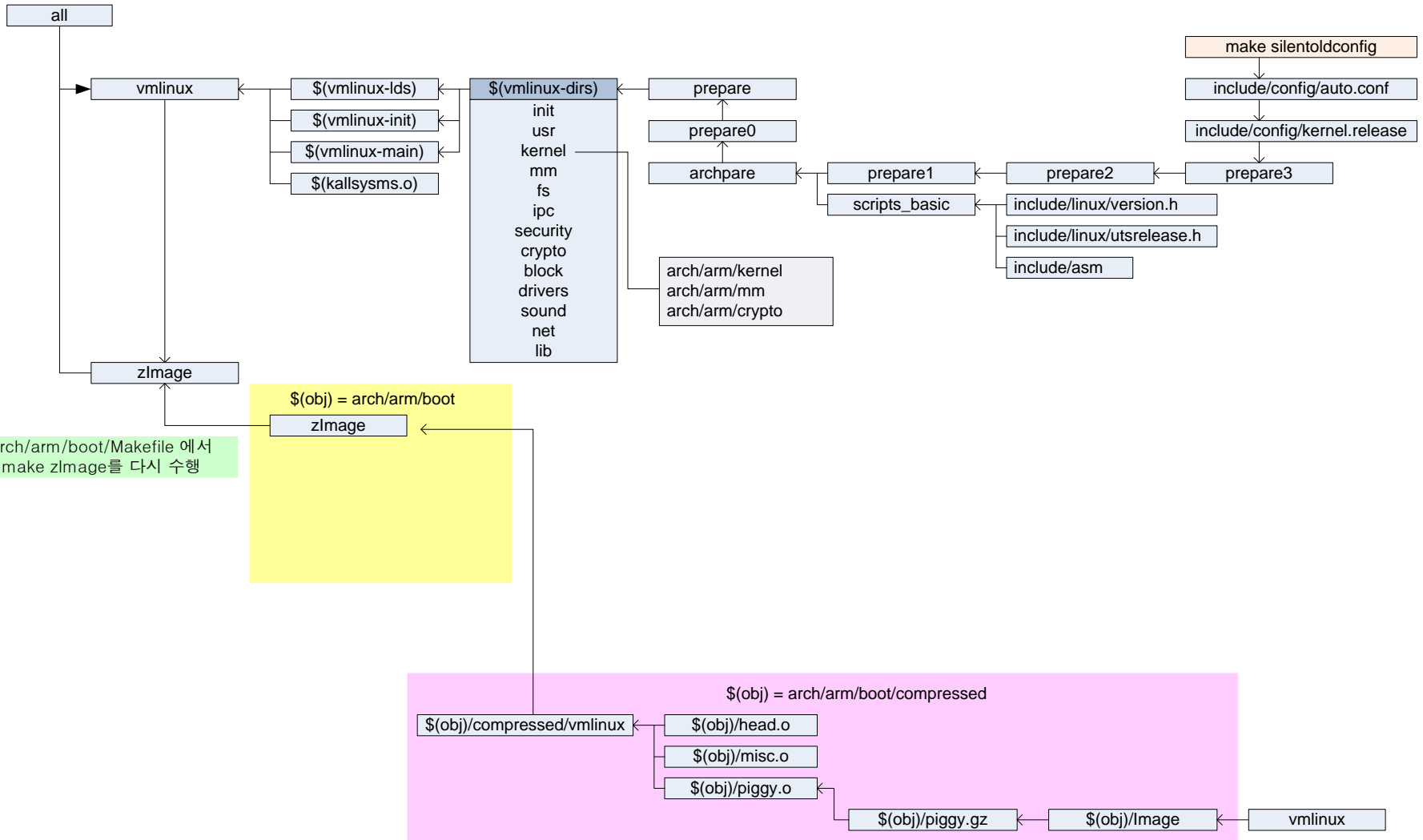
망고100 NAND Partition 영역



SD /MMC Partition 구성



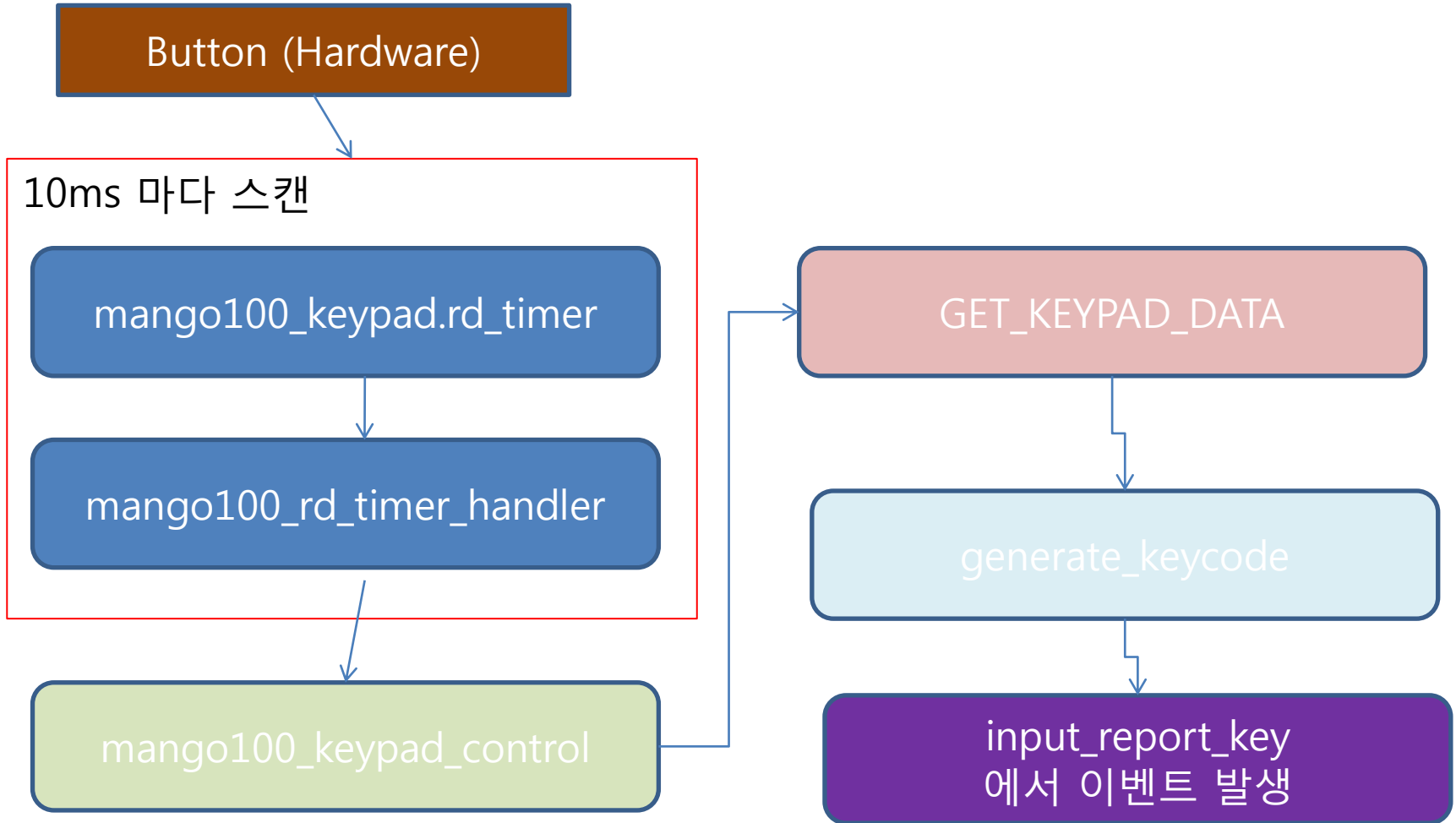
참고 : kernel Makefile 계층도



arch/arm/boot/Makefile 에서 make zImage를 다시 수행

여기에서 vmlinux는 커널 소스 최상위 디렉토리에서 만들어진 vmlinux를 말한다.

Keypad driver flow



안드로이드 Key event 처리

Key button Map 정의는 vendor/wsec/wmango100/wmango100-keypad.kl
frameworks/base/libs/EventHub.cpp에 scan_dir, open_device 정의

KeyInputQueue(1853): InputDeviceReader.run()

```
static const char *device_path = "/dev/input";  
bool EventHub::openPlatformInput(void)  
{  
    ..  
    res = scan_dir(device_path);  
    ..  
}
```

```
int EventHub::scan_dir(const char *dirname)  
{  
    while((de = readdir(dir))) {  
        strcpy(filename, de->d_name);  
        open_device(devname);  
    }  
}
```

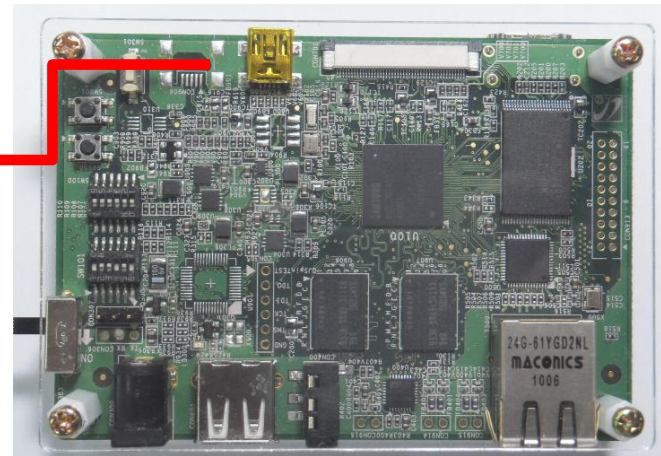
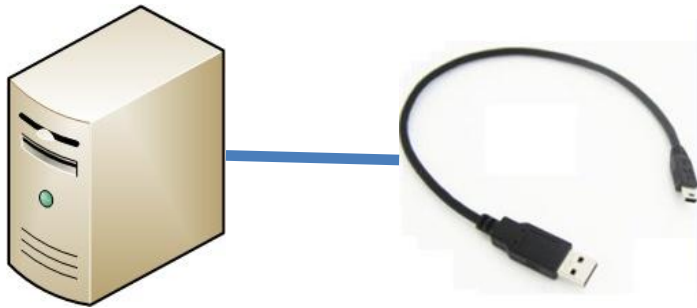
```
int EventHub::open_device(const char *deviceName)
```

안드로이드 구조



ADB 연결하기

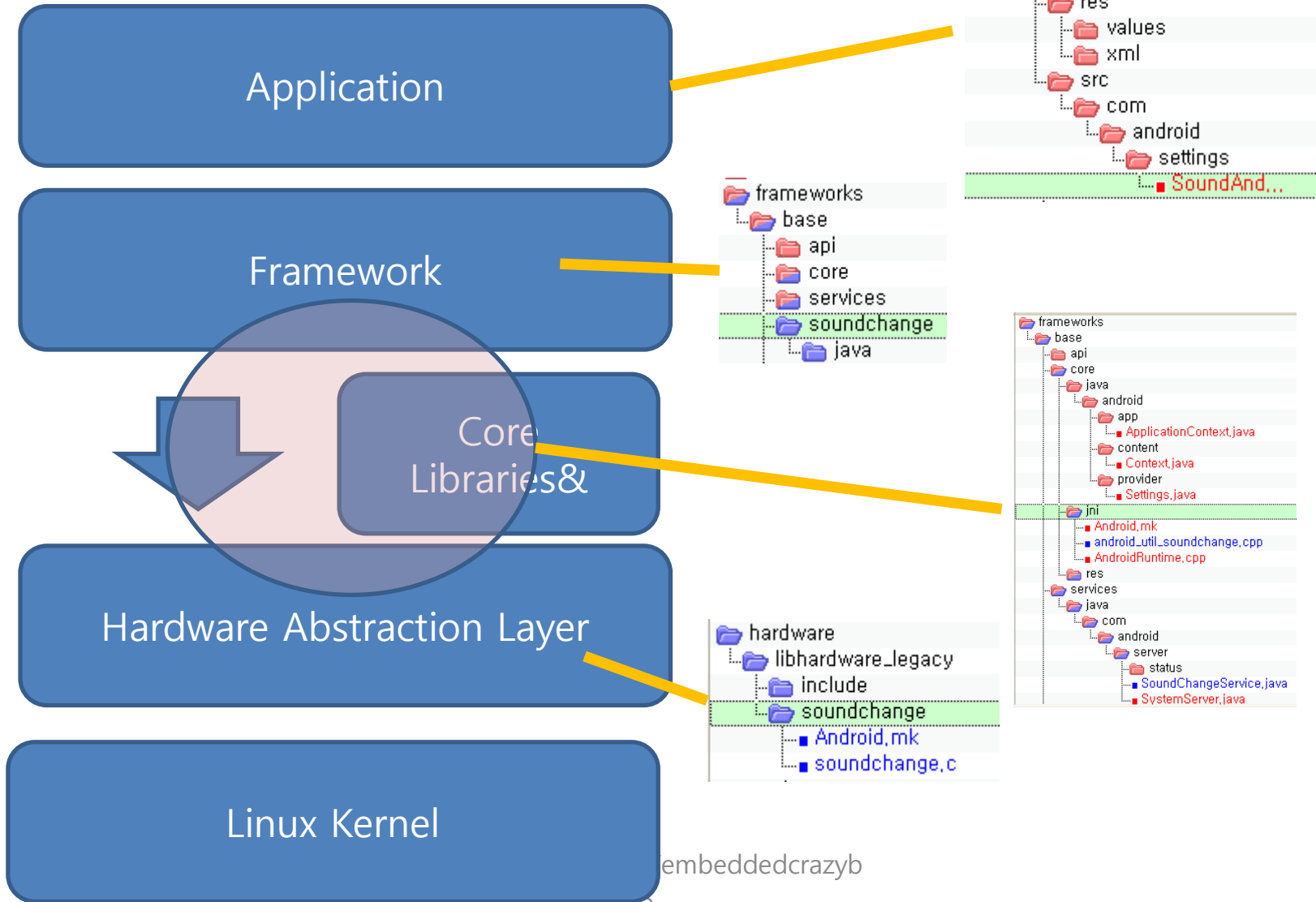
- PC와 망고보드를 usb cable로 연결



- >adb devices

```
Y:\work\busybox-work>adb devices
* daemon not running. starting it now *
* daemon started successfully *
List of devices attached
0123456789ABCDEF    device
```

Android 구조



Q&A

