

# **MV 5422 Android 4.4.2**

## **Compiling**



**Microvision Co., Ltd**

## Document Information

Version	1.0
File Name	MV5422 Android Compilation.doc
Date	2014. 11. 06
Satus	Working

## Revision History

Date	Version	Update Descriptions	Editor
2014. 11. 06.	V1.0	First Edition	Microvision

## 1. Package for Development

The following packages are in the directory /SRC/Android in the CD:

파일	설명	버전
u-boot.tar.gz	Bootloader	
linux-3.10.9-kitkat.tar.gz	Kernel	3.10.9
Android-4.4.2.tar.gz	KitKat	4.4.2
arm-eabi-4.4.3.zip	arm-eabi-	Q3 67
arm-eabi-4.7	arm-eabi-	4.7

### Tool chain

- compilation of u-boot is to use the arm-eabi-4.4.3, and the compilation.
- Kernel compilation by using the Toolchain that are included in the Android, will compile.

## 2. Bootloader Setup

### 2.1. u-boot Environment Setup

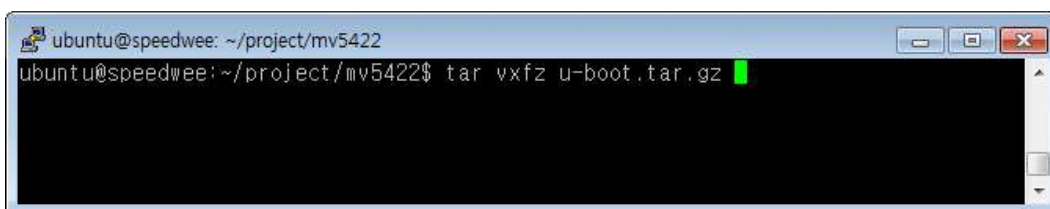
Generally, the Embedded Linux BSP is composed of 3 image files:

Embedded Linux BSP = Boot Loader + Kernel + File System

Boot Loader is the program necessary to load the kernel to the memory

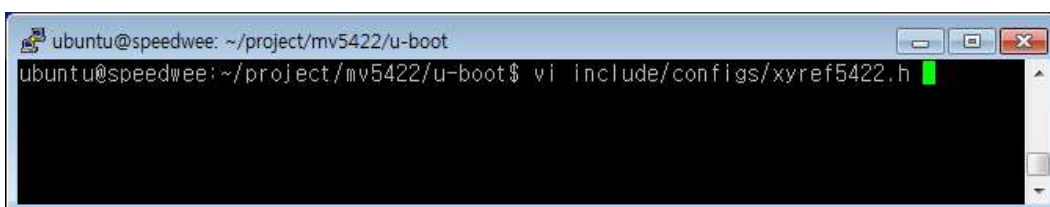
Enter in the following for file decompression:

```
# tar vxzf u-boot.tar.gz
```



As shown below using the “vi” editor, open the file “[xyref5422.h](#)” and you will find the basic environment at its default.(ex: TFTP, CPU clock, DDR Program Counter)

```
# vi include/configs/xyref5422.h
```



The prompt name on the mv5422 boot board after booting the new bootloader program:

```
#define CONFIG_SYS_PROMPT      "MV5422 # "
```

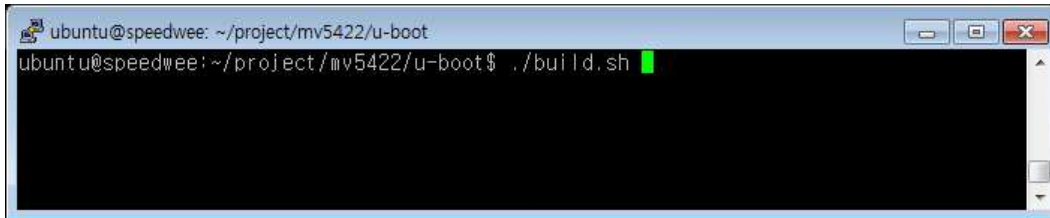
DRAM Program Counter address for Download

```
/* DRAM Base */
```

```
#define CONFIG_SYS_SDRAM_BASE      0x40000000
```

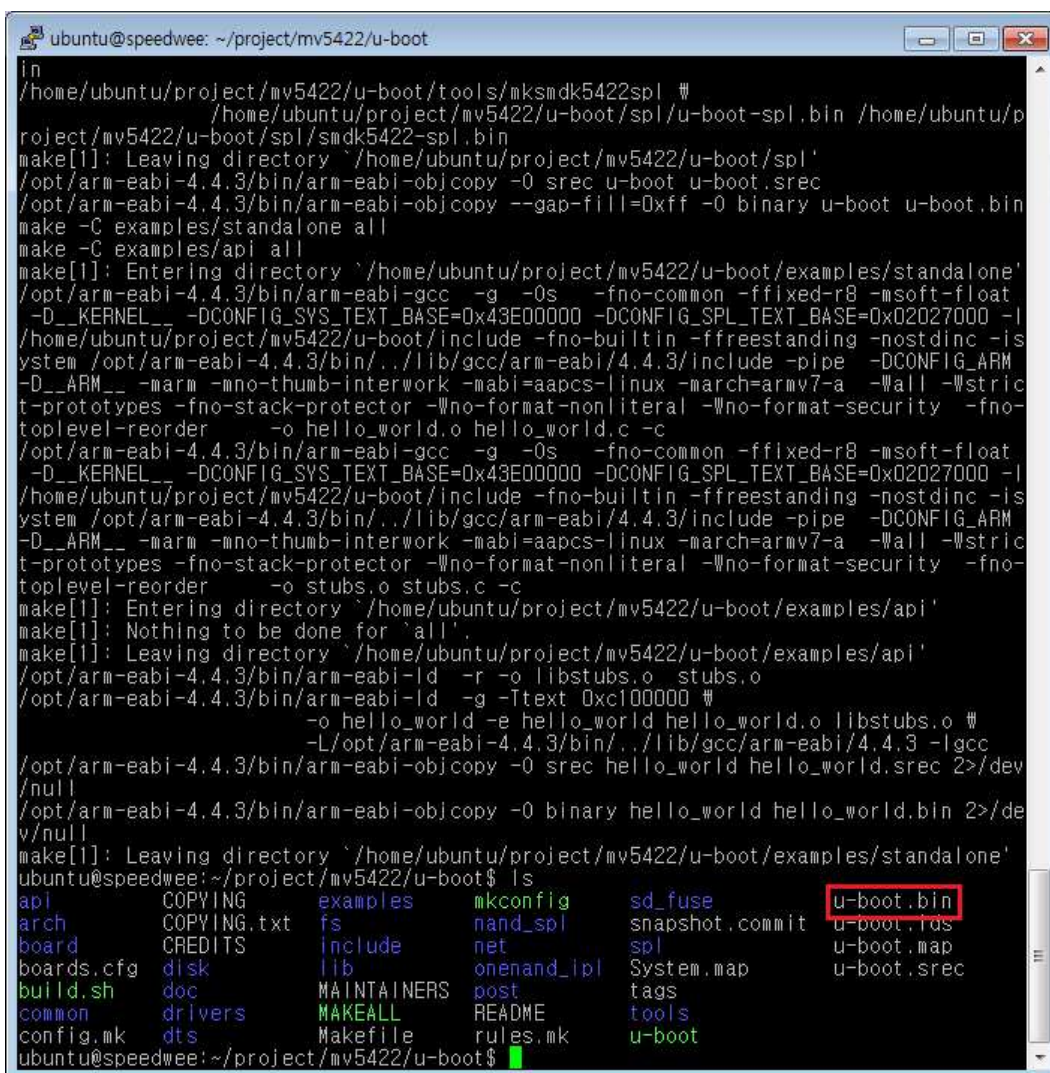


```
# ./build.sh
```



```
ubuntu@speedwee: ~/project/mv5422/u-boot
ubuntu@speedwee:~/project/mv5422/u-boot$ ./build.sh
```

Compile complete look and “u-boot.bin” You can see that the file is created.



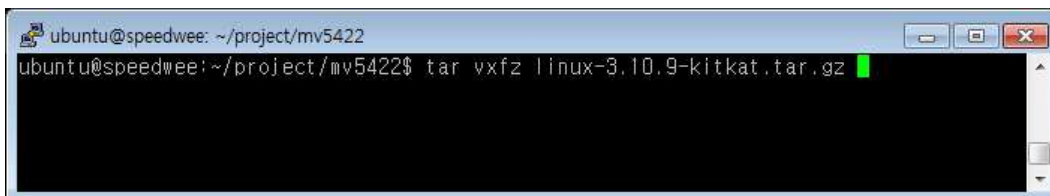
```
in
/home/ubuntu/project/mv5422/u-boot/tools/mksmdk5422spl #
/home/ubuntu/project/mv5422/u-boot/spl/u-boot-spl.bin /home/ubuntu/p
project/mv5422/u-boot/spl/smdk5422-spl.bin
make[1]: Leaving directory `/home/ubuntu/project/mv5422/u-boot/spl'
/opt/arm-eabi-4.4.3/bin/arm-eabi-objcopy -O srec u-boot u-boot.srec
/opt/arm-eabi-4.4.3/bin/arm-eabi-objcopy --gap-fill=0xff -O binary u-boot u-boot.bin
make -C examples/standalone all
make -C examples/api all
make[1]: Entering directory `/home/ubuntu/project/mv5422/u-boot/examples/standalone'
/opt/arm-eabi-4.4.3/bin/arm-eabi-gcc -g -Os -fno-common -ffixed-r8 -msoft-float
-D__KERNEL__ -DCONFIG_SYS_TEXT_BASE=0x43E00000 -DCONFIG_SPL_TEXT_BASE=0x02027000 -I
/home/ubuntu/project/mv5422/u-boot/include -fno-builtin -ffreestanding -nostdinc -is
ystem /opt/arm-eabi-4.4.3/bin/./lib/gcc/arm-eabi/4.4.3/include -pipe -DCONFIG_ARM
-D__ARM__ -marm -mno-thumb-interwork -mabi=aapcs-linux -march=armv7-a -Wall -Wstric
t-prototypes -fno-stack-protector -Wno-format-nonliteral -Wno-format-security -fno-
oplevel-reorder -o hello_world.o hello_world.c -c
/opt/arm-eabi-4.4.3/bin/arm-eabi-gcc -g -Os -fno-common -ffixed-r8 -msoft-float
-D__KERNEL__ -DCONFIG_SYS_TEXT_BASE=0x43E00000 -DCONFIG_SPL_TEXT_BASE=0x02027000 -I
/home/ubuntu/project/mv5422/u-boot/include -fno-builtin -ffreestanding -nostdinc -is
ystem /opt/arm-eabi-4.4.3/bin/./lib/gcc/arm-eabi/4.4.3/include -pipe -DCONFIG_ARM
-D__ARM__ -marm -mno-thumb-interwork -mabi=aapcs-linux -march=armv7-a -Wall -Wstric
t-prototypes -fno-stack-protector -Wno-format-nonliteral -Wno-format-security -fno-
oplevel-reorder -o stubs.o stubs.c -c
make[1]: Entering directory `/home/ubuntu/project/mv5422/u-boot/examples/api'
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/home/ubuntu/project/mv5422/u-boot/examples/api'
/opt/arm-eabi-4.4.3/bin/arm-eabi-ld -r -o libstubs.o stubs.o
/opt/arm-eabi-4.4.3/bin/arm-eabi-ld -g -Ttext 0xc100000 #
-o hello_world -e hello_world hello_world.o libstubs.o #
-L/opt/arm-eabi-4.4.3/bin/./lib/gcc/arm-eabi/4.4.3 -lgcc
/opt/arm-eabi-4.4.3/bin/arm-eabi-objcopy -O srec hello_world hello_world.srec 2>/dev
/null
/opt/arm-eabi-4.4.3/bin/arm-eabi-objcopy -O binary hello_world hello_world.bin 2>/de
v/null
make[1]: Leaving directory `/home/ubuntu/project/mv5422/u-boot/examples/standalone'
ubuntu@speedwee:~/project/mv5422/u-boot$ ls
api          COPYING      examples     mkconfig    sd_fuse     u-boot.bin
arch        COPYING.txt  fs           nand_spl    snapshot.commit  u-boot.tds
board       CREDITS     include      net         spl         u-boot.map
boards.cfg  disk        lib          onenand_lpl System.map  u-boot.srec
build.sh    doc         MAINTAINERS post         tags
common     drivers     MAKEALL     README     tools
config.mk   dts        Makefile    rules.mk    u-boot
ubuntu@speedwee:~/project/mv5422/u-boot$
```

## 3. Kernel setup

### 3.1. How to compile

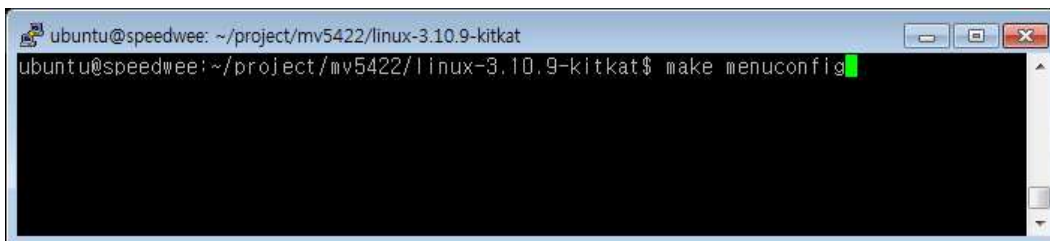
Enter in the following for file decompression

```
# tar vxzf linux-3.10.9-kitkat.tar.gz
```

A terminal window titled 'ubuntu@speedwee: ~/project/mv5422'. The prompt is 'ubuntu@speedwee:~/project/mv5422\$' and the command 'tar vxzf linux-3.10.9-kitkat.tar.gz' is entered. A green cursor is at the end of the command line.

Put in the following commands for compilation to execute the kernel environment setup:

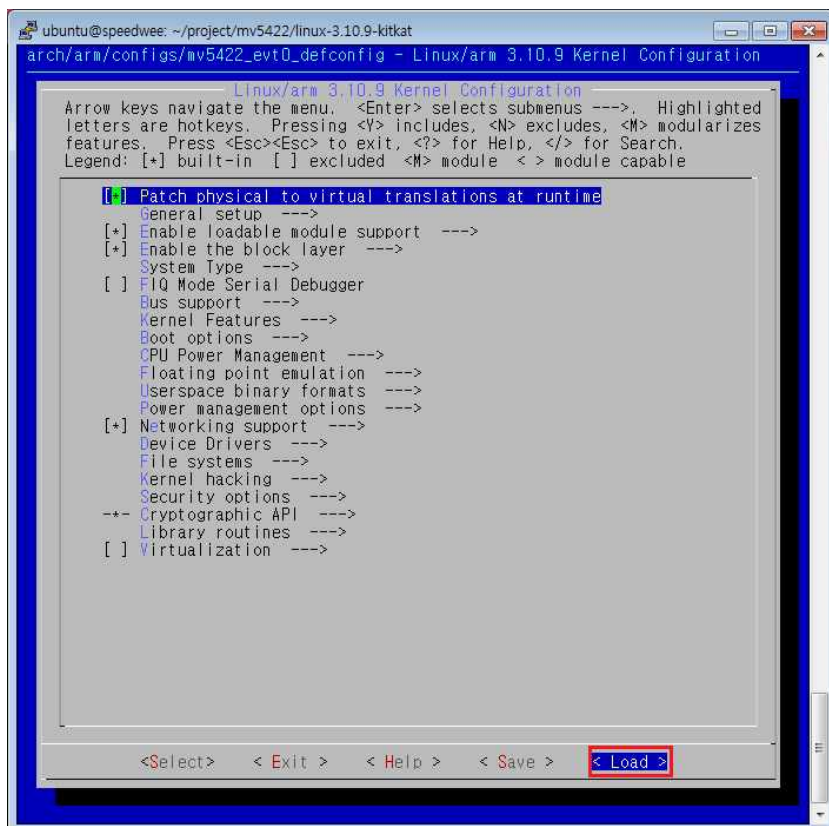
```
# make menuconfig
```

A terminal window titled 'ubuntu@speedwee: ~/project/mv5422/linux-3.10.9-kitkat'. The prompt is 'ubuntu@speedwee:~/project/mv5422/linux-3.10.9-kitkat\$' and the command 'make menuconfig' is entered. A green cursor is at the end of the command line.

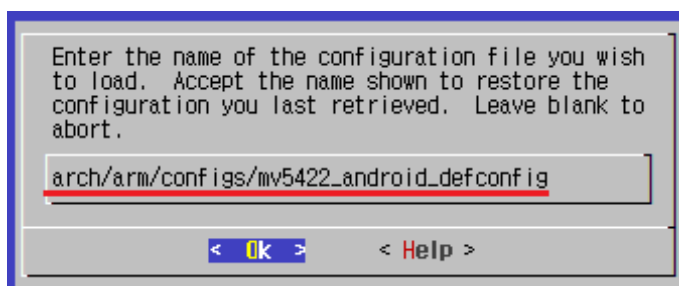
Besides make menuconfig there are kernel setting commands such as make config and make xconfig but the most popular one is the make menuconfig which is simple UI (User Interface) to use with the arrow keys known as the console (monitor) or telnet terminal is used for the Kernel Configuration.

If all the content of the setting menu is set, it doesn't have to be newly set in each time. So to save the previous configuration to a separate file, there is an option in the menu down below as "Save Configuration to an Alternate File". In opposite, previous setup configuration can be reloaded, Load and Kernel Configuration can be made by reading the file from "[mv5422\\_android\\_defconfig](#)" which is saved at arch/arm/configs/which is Kernel Source directory.

Therefore, make menuconfig "[<Load>](#)" on pages at the bottom of the screen, you can select the menu, enter the following screen.



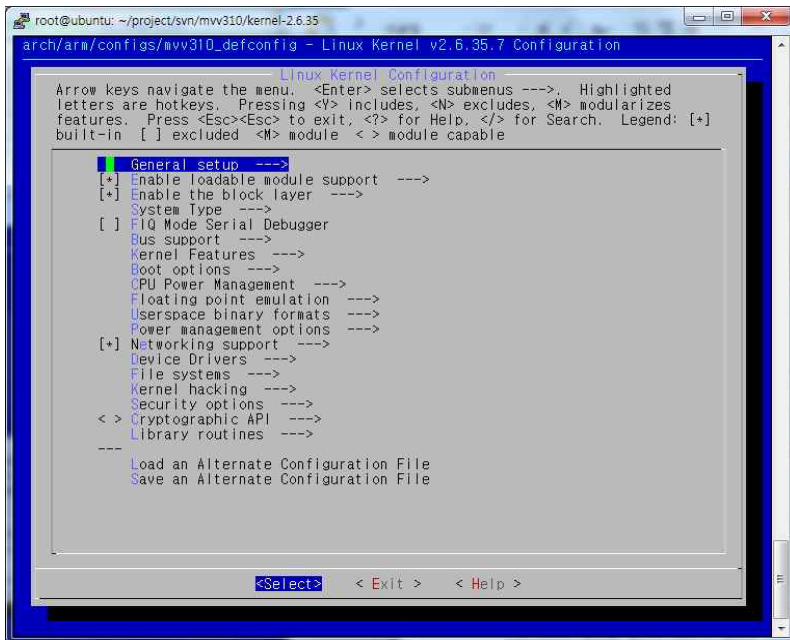
Load “arch/arm/configs/mv5422\_android\_defconfig”



The kernel configuration(make menuconfig) must be saved after the setup is complete. The kernel configuration is saved under the file name “.config” under the kernel source directory. The reason “.config” needs to be saved is that it will be checked during the “make dep” step, which is a crucial step for the compilation process. If a window asking to save pops up, make sure to answer “yes”

After loading is complete, exit

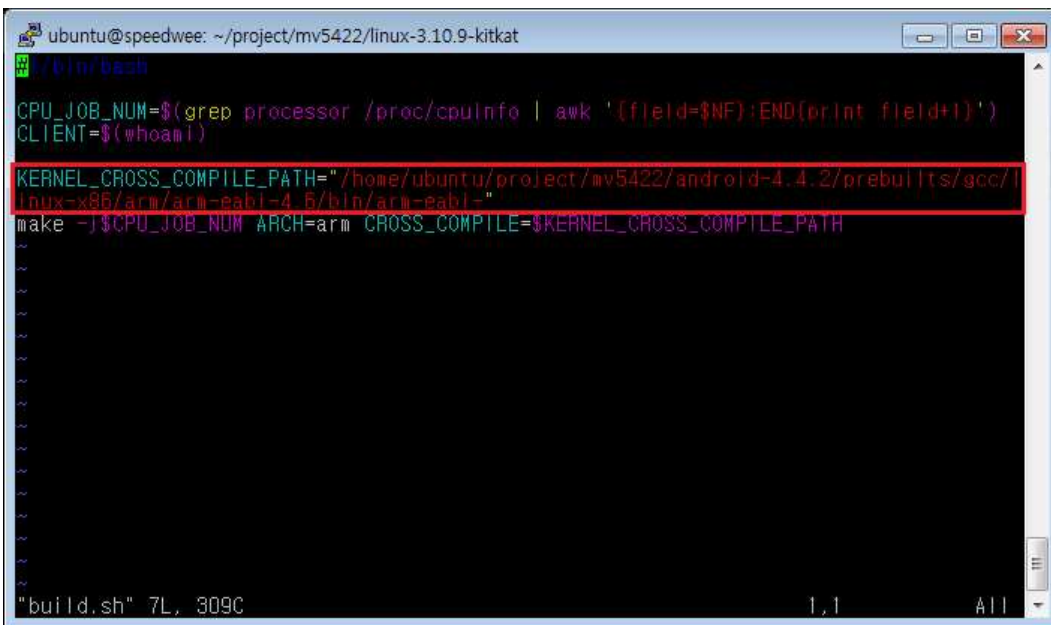
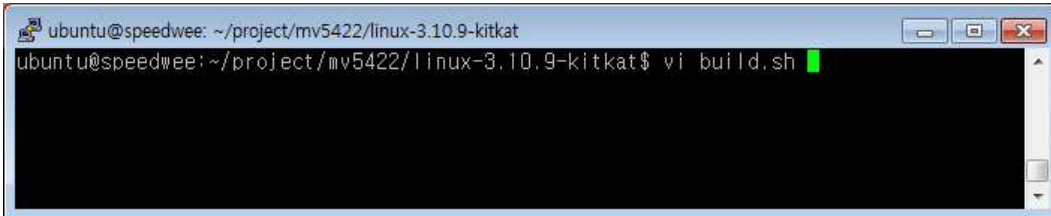




The Linux kernel image(zImage) making process is divided into compiling, linking, file type changing (ELF→BIN) by Binutil(objcopy), and file decompression (gzip). All of these combined make up the command “make” under Makefile.

For compiling kernel, you have to set up include in Android Toolchain.

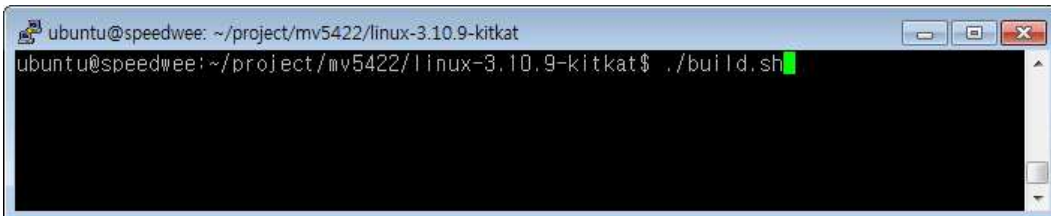
#vi build.sh



KERNEL\_CROSS\_COMPILE\_PATH="/home/ubuntu/project/mv5422/android-4.4.2/prebuilts/gcc/linux-x86/arm/arm-eabi-4.6/bin/arm-eabi-"

(Above path, you must do a set to the location of the folder where you unzipped the compression from customers.)

# ./build.sh



```

ubuntu@speedwee: ~/project/mv5422/linux-3.10.9-kitkat
HOSTCC  scripts/mod/file2alias.o
HOSTLD  scripts/mod/modpost
CALL    scripts/checksyscalls.sh
GEN     usr/initramfs_data.cpio
AS      usr/initramfs_data.o
LD      usr/built-in.o
CHK     include/generated/compile.h
CC      drivers/power/samsung_fake_battery.o
LD      drivers/power/built-in.o
LD      drivers/built-in.o
LINK    vmlinux
LD      vmlinux.o
MODPOST vmlinux.o
GEN     .version
CHK     include/generated/compile.h
UPD     include/generated/compile.h
CC      init/version.o
LD      init/built-in.o
KSYM    .tmp_kallsyms1.o
KSYM    .tmp_kallsyms2.o
LD      vmlinux
SORTEX  vmlinux
SYSMAP  System.map
Building modules, stage 2.
MODPOST 0 modules
OBJCOPY arch/arm/boot/Image
DTC      arch/arm/boot/dts/exynos5422_evt0-mv5422.dtb
Kernel: arch/arm/boot/Image is ready
GZIP    arch/arm/boot/compressed/piggy.gzip
AS      arch/arm/boot/compressed/piggy.gzip.o
LD      arch/arm/boot/compressed/vmlinux
OBJCOPY arch/arm/boot/zImage
Kernel: arch/arm/boot/zImage is ready
CAT     arch/arm/boot/zImage-dtb
Kernel: arch/arm/boot/zImage-dtb is ready
ubuntu@speedwee:~/project/mv5422/linux-3.10.9-kitkat$ █

```

When compilation is complete, zImage-dtb file is generated in arch/arm/boot

```

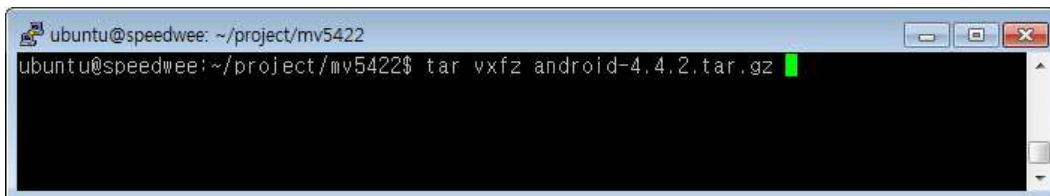
ubuntu@speedwee: ~/project/mv5422/linux-3.10.9-kitkat
UPD     include/generated/compile.h
CC      init/version.o
LD      init/built-in.o
KSYM    .tmp_kallsyms1.o
KSYM    .tmp_kallsyms2.o
LD      vmlinux
SORTEX  vmlinux
SYSMAP  System.map
Building modules, stage 2.
MODPOST 0 modules
OBJCOPY arch/arm/boot/Image
DTC      arch/arm/boot/dts/exynos5422_evt0-mv5422.dtb
Kernel: arch/arm/boot/Image is ready
GZIP    arch/arm/boot/compressed/piggy.gzip
AS      arch/arm/boot/compressed/piggy.gzip.o
LD      arch/arm/boot/compressed/vmlinux
OBJCOPY arch/arm/boot/zImage
Kernel: arch/arm/boot/zImage is ready
CAT     arch/arm/boot/zImage-dtb
Kernel: arch/arm/boot/zImage-dtb is ready
ubuntu@speedwee:~/project/mv5422/linux-3.10.9-kitkat$ ls arch/arm/boot/
bootp compressed dts Image install.sh Makefile zImage zImage-dtb
ubuntu@speedwee:~/project/mv5422/linux-3.10.9-kitkat$ █

```

## 4. KitKat Compilation

Enter in the following command:

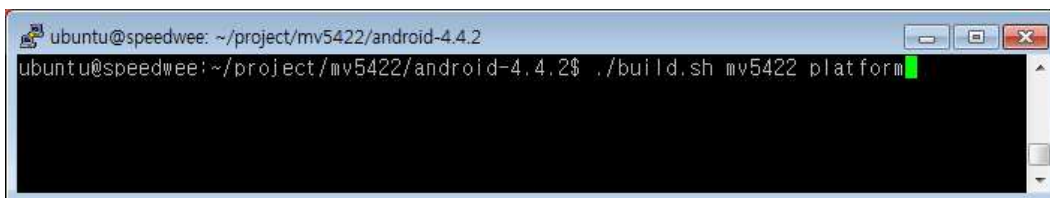
```
# tar vxzf android-4.4.2.tar.gz
```



```
ubuntu@speedwee: ~/project/mv5422
ubuntu@speedwee:~/project/mv5422$ tar vxzf android-4.4.2.tar.gz
```

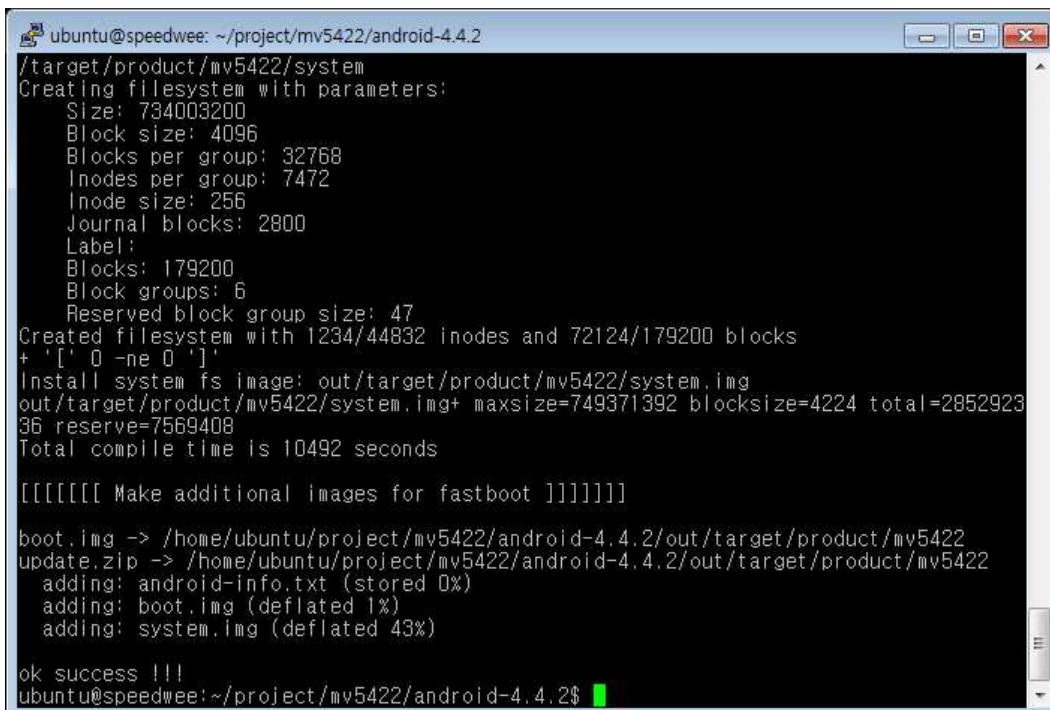
This is command for compilation.

```
# ./build.sh mv5422 platform
```



```
ubuntu@speedwee: ~/project/mv5422/android-4.4.2
ubuntu@speedwee:~/project/mv5422/android-4.4.2$ ./build.sh mv5422 platform
```

Image is in the folder `Android/out/target/product/mv5422`



```
ubuntu@speedwee: ~/project/mv5422/android-4.4.2
/target/product/mv5422/system
Creating filesystem with parameters:
  Size: 734003200
  Block size: 4096
  Blocks per group: 32768
  Inodes per group: 7472
  Inode size: 256
  Journal blocks: 2800
  Label:
  Blocks: 179200
  Block groups: 6
  Reserved block group size: 47
Created filesystem with 1234/44832 inodes and 72124/179200 blocks
+ '[' 0 -ne 0 ']'
Install system fs image: out/target/product/mv5422/system.img
out/target/product/mv5422/system.img+ maxsize=749371392 blocksize=4224 total=2852923
36 reserve=7569408
Total compile time is 10492 seconds

[[[[[[[ Make additional images for fastboot ]]]]]]

boot.img -> /home/ubuntu/project/mv5422/android-4.4.2/out/target/product/mv5422
update.zip -> /home/ubuntu/project/mv5422/android-4.4.2/out/target/product/mv5422
  adding: android-info.txt (stored 0%)
  adding: boot.img (deflated 1%)
  adding: system.img (deflated 43%)

ok success !!!
ubuntu@speedwee:~/project/mv5422/android-4.4.2$
```