

How To Build Android for ARM Chip boards

Noritsuna Imamura



List of Chips

- RockChip
 - RK3188 (Cortex-A9 Quad)
 - MK802IV
- AllWinner
 - A31 (Cortex-A7 Quad)
 - Onda V973 (v2)
- MediaTek
 - MT6589 (Cortex-A7 Quad)
 - Acer Liquid E2
- Freescale
 - i.MX6Q (Cortex-A9 Quad)
 - i.MX6Q SABRA Lite
- TI
 - OMAP5432 (Cortex-A15 Dual)
 - OMAP5432-EVM
- Samsung
 - Exynos 5 (Cortex-A15 Quad and Cortex-A7 Quad)
 - ODROID-XU
- QUALCOMM
 - krait 400 (Like Cortex-A15 Quad and Cortex-A7 Quad)
 - New Nexus7



Feature Matrix

	AllWinner	RockChip	MediaTek	Freescale	TI	Samsung	Qualcomm
Chip Name	A31	RK3188	MT6589	i.MX6Q	OMAP5432	Exynos5	Krait400
CPU Type	A7 Quad	A9 Quad	A7 Quad	A9 Quad	A15 Dual	A15 Quad	A15 Quad
Device Type	Tablet	Mini PC	Smart Phone	EVB	EVB	EVB	Tablet
Used Area	Cheap Device	Cheap Device	Cheap Phone	N/A	Customized Device	N/A	3G/LTE Device
Official Source Code	Kernel	Kernel	Some Android Kernel	Full Android Kernel	Full Android Kernel	Full Android Kernel	Full Android Kernel
Support	✗	△	✗	○	◎	◎	○
Availability	○	○	?	○	○	?	✗
Promising	○	◎	○	✗	✗	○	△
Reason	Cheapest	Better chip	RF chip		Dev team is gone		Expensive

Common Setup

Building Environment

- CPU
 - Core i7
 - 4core – 8thread
- Memory
 - 8GB
- OS
 - Ubuntu12.04 64bits

Common Setup 1/2

● Setup Build Tools

- sudo apt-get install

- bison build-essential curl flex git-core gnupg gperf libesd0-dev libncurses5-dev libsdl1.2-dev libwxgtk2.8-dev libxml2 libxml2-utils lzop openjdk-6-jdk openjdk-6-jre pngcrush schedtool squashfs-tools xsltproc zip zlib1g-dev
- g++-multilib gcc-multilib lib32ncurses5-dev lib32readline-gplv2-dev lib32z1-dev

● Setup Build Tools for ARM

- sudo apt-get install

- uboot-mkimage

Common Setup 2/2

● Setup Build Tools for Android

- ```
> curl http://commondatastorage.googleapis.com/git-
repo-downloads/repo > ~/bin/repo
> chmod a+x ~/bin/repo
> export PATH=$PATH:~/bin/repo

> sudo add-apt-repository ppa:nilarimogard/webupd8
> sudo apt-get update
> sudo apt-get install android-tools-fastboot
```

- AllWinner – A31 (Cortex-A7 Quad)

# AllWinner – A31 (Cortex-A7 Quad)

- Target

- Tablet

- Onda V973 (v2)

- [http://www.onda.cn/Tablet/  
ProductInfo.aspx?ProductId=250](http://www.onda.cn/Tablet/ProductInfo.aspx?ProductId=250)

- OS

- Android

- UnOfficial CyanogenMod

- [http://www.slatedroid.com/  
topic/77481-rom-aokp-42-  
for-allwinner-a31-devices-  
sdk-32-2013-07-27-  
update/](http://www.slatedroid.com/topic/77481-rom-aokp-42-for-allwinner-a31-devices-sdk-32-2013-07-27-update/)



[rom] AOKP 4.2 for Allwinner A31 devices (SDK 3.2) - 2013-07-27 update

Christian Troy

Posted 11 June 2013 - 07:45 AM

If you appreciate my work and would like to support me with a small donation you can click on the [Donate](#) button.

If every happy user of this rom would just donate something I could invest more time in all these Android projects.

**Issues and possible workarounds**

\* After the setup wizard the launcher won't fit correctly the screen but after a reboot it will automatically fix by itself. If you don't like transparent status / system bar go into Settings -> ROM control and edit the transparency.

\* Accelerated video playback is supported by MX Player in HW+/HW mode and pretty much every file will play, try both playback method because it depends on the video codec. I noticed that HD videos plays better in HW+ mode while SD (XviD for example) play better in HW mode. For a better audio support (included DTS playback) have a look [here](#).

\* HDMI shows a picture but it doesn't fill the TV screen entirely.

\* This is not a broken functionality but a lack of this ROM that can only export one partition through [USB Mass Storage](#), if you wanna access to both memories you should use MTP instead of UMS, only CyanogenMod can export multiple LUNs.

\* Probably other things...

# AllWinner – A31 (Cortex-A7 Quad)



- Sorry, I couldn't get this tablet…

- RockChip – RK3188 (Cortex-A9 Quad)

# RockChip – RK3188 (Cortex-A9 Quad)

- Target

- Mini PC

- MK802IV

- <http://www.cloudsto.com/android-mini-pc-s/rikomagic-quad-core-mk802iv-8gb-flash-bluetooth-dhl-express-shipping-detail.html>

- OS

- Official Kernel

- (But No Android&Ubuntu)

- <http://www.rikomagic.co.uk/forum/viewtopic.php?f=6&t=4696>



**phpBB® Rikomagic MK802 Forums**  
Discussion forums for everything relating to the Rikomagic MK802 PC on a stick.

Board index < Rikomagic Forums < Rikomagic MK802 II Forum

FAQ Register Login

**MK802 IV QUAD CORE MINI PC**

NOW SHIPPING WORLDWIDE ORDER NOW

Linux Source Code for RK3188 (MK802IV) Now Available!

Moderators: tatubias, twjordc, KEgg

POSTREPLY Search this topic... Search

Linux Source Code for RK3188 (MK802IV) Now Available!

by andykirby Thu Jun 06, 2013 9:25 am

Dear Developer Community,

We have acquired the Linux Source Code for the RK3188 devices, we hope this will open up massive opportunities for customised firmware and Linux on all Rockchip 3188 devices.

The code will be uploaded to Github shortly.

Andy Kirby

andykirby Site Admin Posts: 273 Joined: Sun May 27, 2012 7:48 pm

232 posts • Page 1 of 24 • 1 2 3 4 5 ... 24

# RockChip – RK3188 (Cortex-A9 Quad)

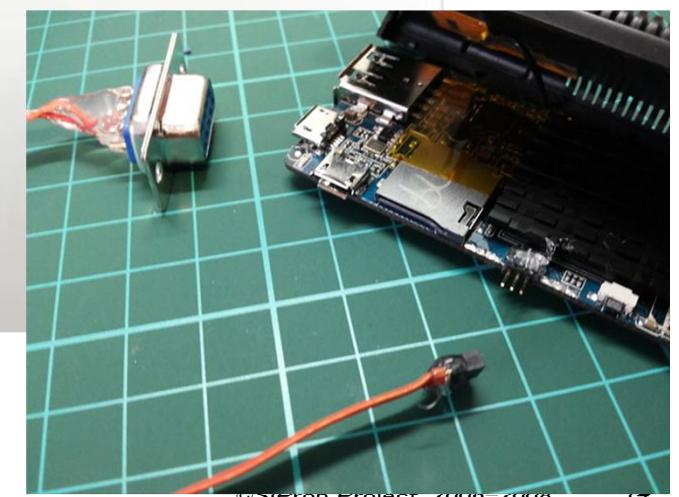
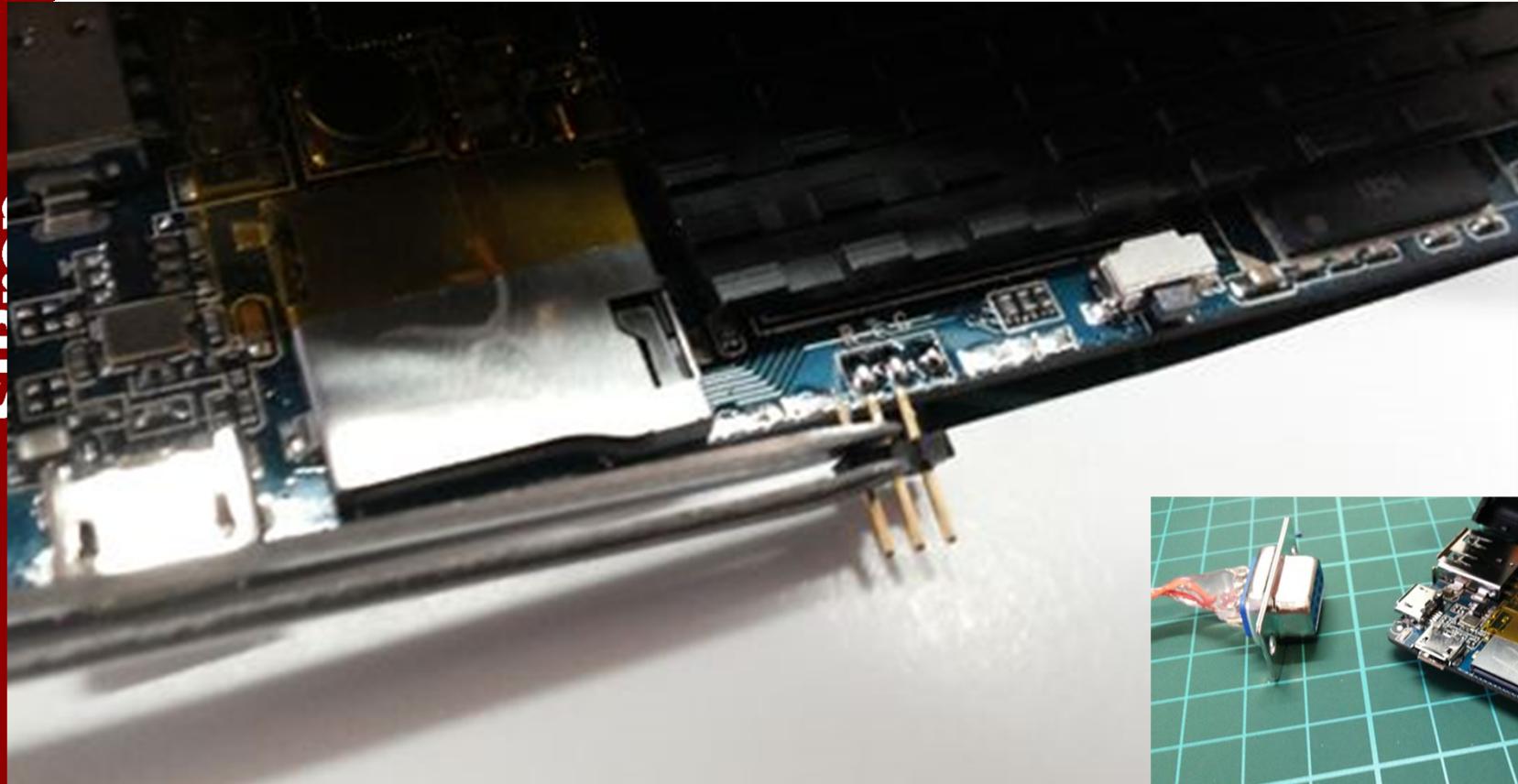


## ● Setup Process

1. Put Serial Connector on board.
2. Customize kernel
3. Make boot-image
4. Write new boot-image

# Put Serial Connector on board

- This connect is TTL level Serial.



# Customize kernel

- Almost Official Kernel doesn't work...

- naobsd released customized kernel.

- <https://github.com/linux-rockchip/rockchip-3.0/tree/reference/mk802iv-ap6210>

- Build this kernel

```
> git clone https://github.com/linux-rockchip/rockchip-3.0.git
> cd rockchip-3.0
> git checkout wip/rockchip-3.0-mk802iv-ap6210
> make rk3188_mk802iv_ap6210_defconfig
> make CROSS_COMPILE=../prebuilts/gcc/linux-x86/arm/arm-eabi-4.6/bin/arm-eabi- kernel.img
> make CROSS_COMPILE=../prebuilts/gcc/linux-x86/arm/arm-eabi-4.6/bin/arm-eabi- zImage
```

# Make boot-image 1/4

## ● Get Original Firmware

- <http://www.rikomagic.co.uk/support.html>

- This firmware has firmware image file & writing tool.

## ● Unpack Firmware

- <https://github.com/naobsd/rkutils>

```
> wget
https://raw.githubusercontent.com/naobsd/rkutils/master/rkunpack.c
> gcc -o rkunpack rkunpack.c
> ./rkunpack IV_AP6210_20130905_1080P.img
```

# Make boot-image 2/4

## ● Unpack boot-image

- [https://github.com/naobsd/cm\\_system\\_core/tree/ics/mkbootimg](https://github.com/naobsd/cm_system_core/tree/ics/mkbootimg)

```
> wget
 https://raw.github.com/naobsd/cm_system_core/ics/mkbootimg/unpackbootimg.c
> wget
 https://raw.github.com/naobsd/cm_system_core/ics/mkbootimg/bootimg.h
> gcc -o unpackbootimg unpackbootimg.c -lcrypto
> ./unpackbootimg -i boot.img
 ✖ boot.img-base, boot.img-cmdline, boot.img-
 pagesize, boot.img-ramdisk.gz, boot.img-zImage
```

# Make boot-image 3/4

- Unpack ramdisk
  - boot.img-ramdisk.gz

```
> mkdir ramdisk
> cd ramdisk
> gunzip -c ../boot.img-ramdisk.gz | cpio -I
```

- If your new kernel has new functions, please modify ramdisk files.
  - Ex. .ko file, new init file, etc.

# Make boot-image 4/4

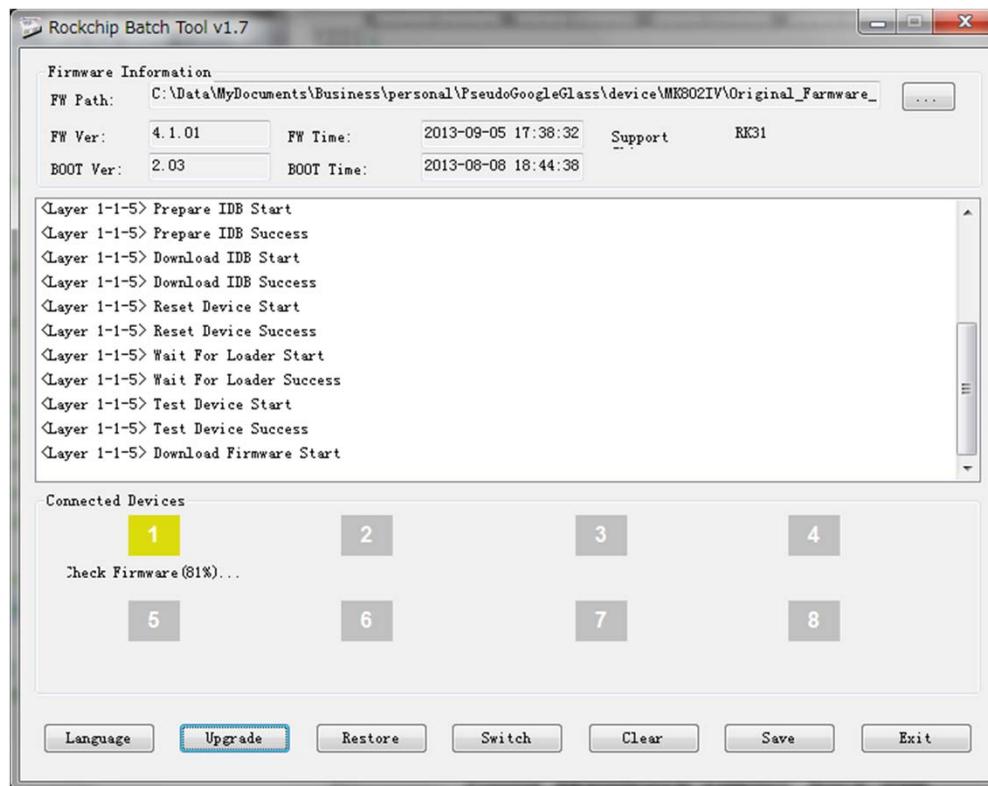
## ● Re-pack ramdisk & boot-image

- [https://github.com/naobsd/cm\\_system\\_core/tree/ics/mkbootimg](https://github.com/naobsd/cm_system_core/tree/ics/mkbootimg)

```
> find . | cpio -o -H newc |
 gzip .../newramdisk.cpio.gz
> cd ../
> wget
 https://raw.github.com/naobsd/cm_system_core/ics/mkbootimg/mkbootimg.c
> gcc -o mkbootimg mkbootimg.c -lcrypto
> ./mkbootimg --kernel zImage --ramdisk
 newramdisk.cpio.gz -o mynewboot.img
※zImage is your new kernel
```

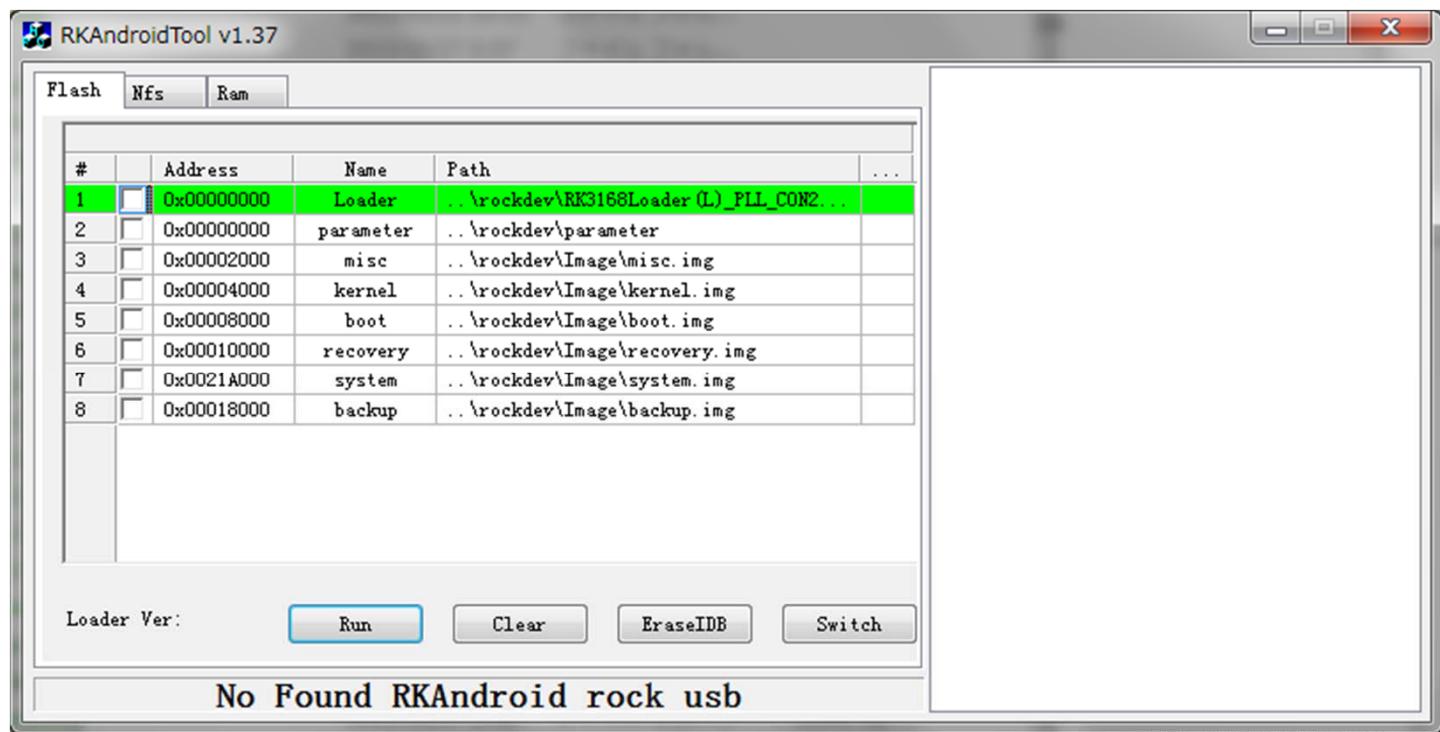
# Write new boot-image 1/4

- Setup driver for Windows
  - Execute “DriverInstall.exe”
- Update firmware to new firmware
  - Execute “RKBatchTool.exe”



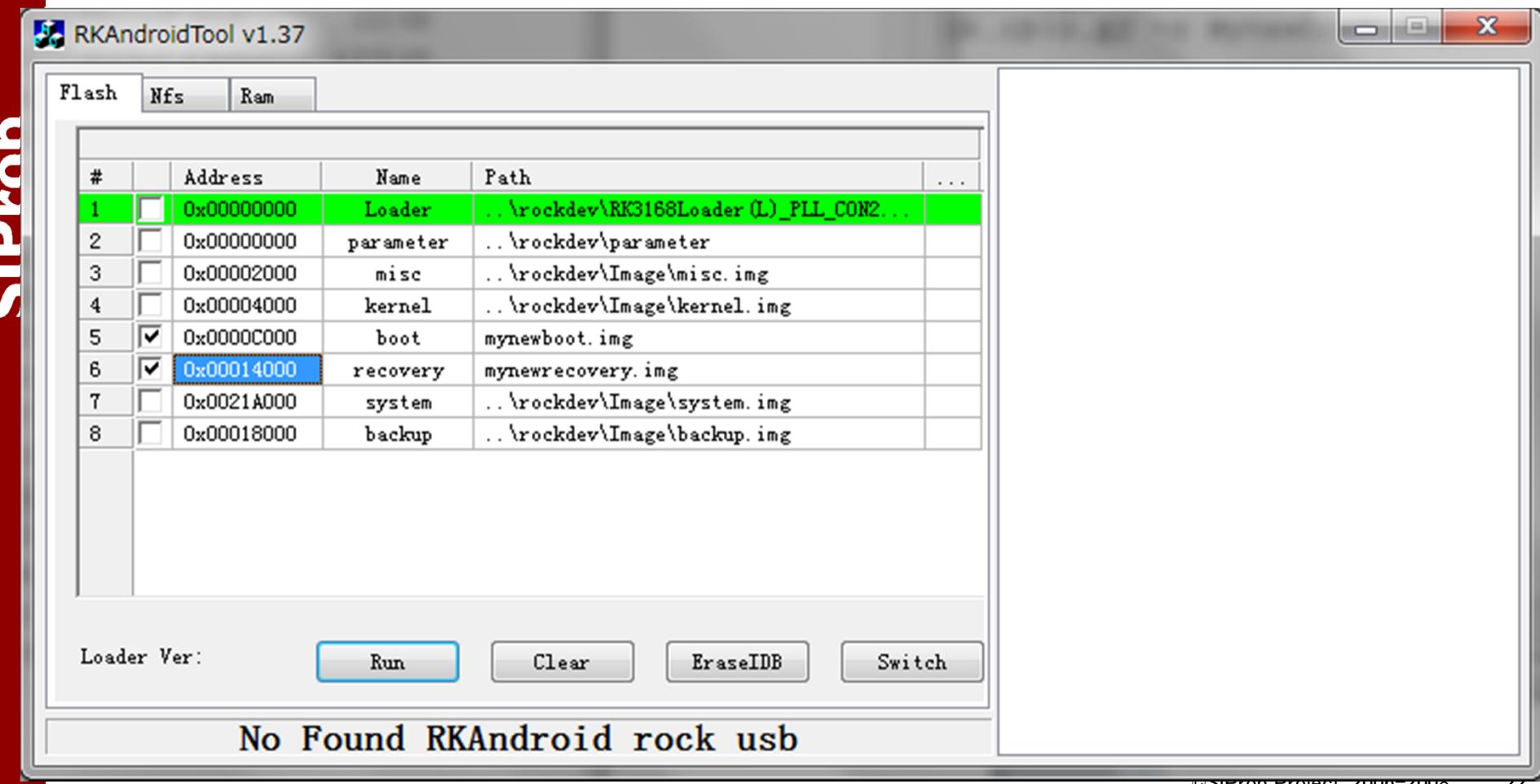
# Write new boot-image 2/4

- Use RockChip writing tool
  - Download writing tool
    - <http://www.filecrop.com/RKAndroidTool-1.37.html>
  - Execute “RKAndroidTool.exe”
    - Default Boot Address is “0x00008000”

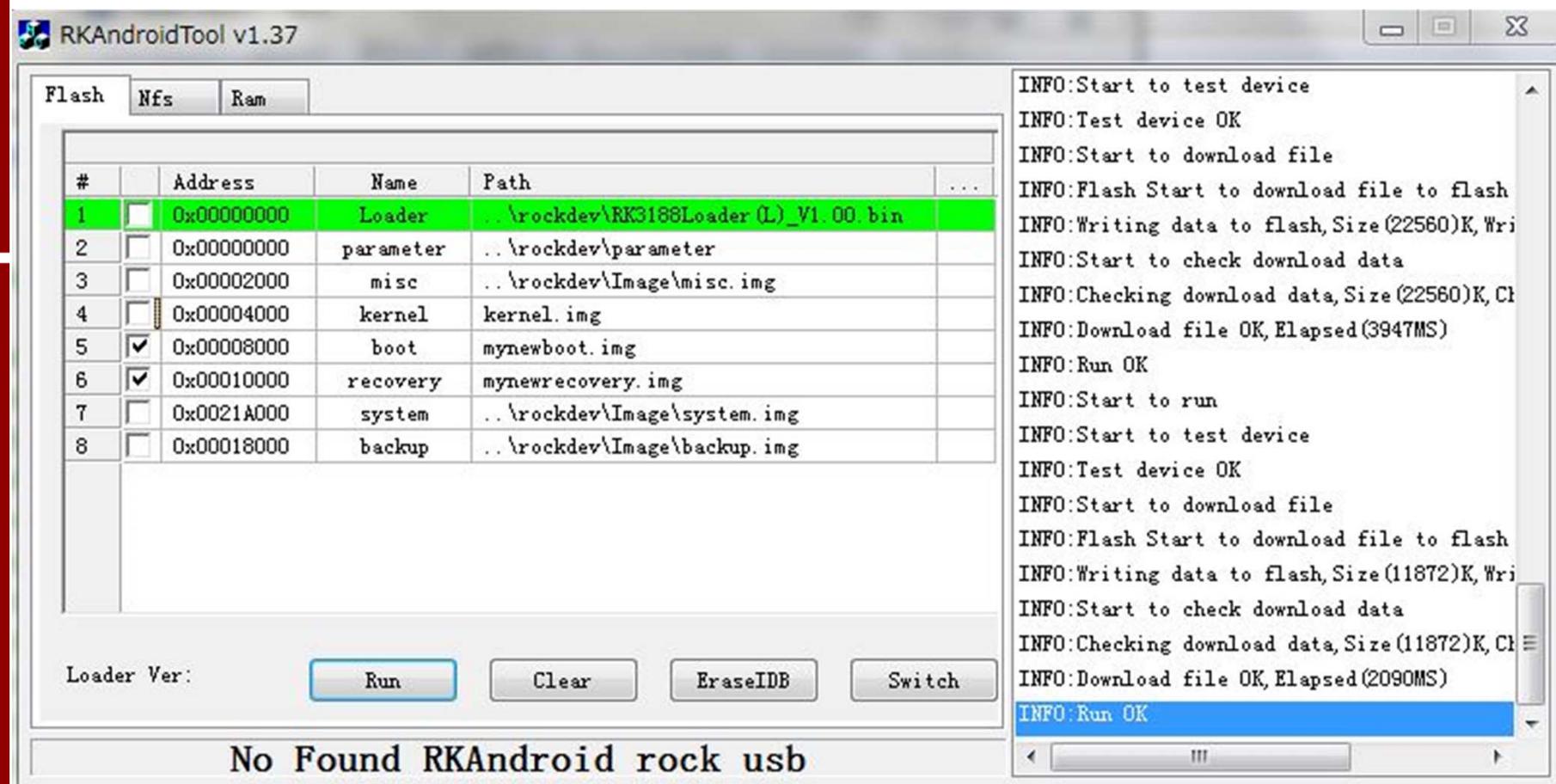


# Write new boot-image 3/4

- Execute “RKAndroidTool.exe”
- Change Boot Address to “0x0000C000”



# Write new boot-image 4/4



● MediaTek – MT6589 (Cortex-A7 Quad)

# MediaTek – MT6589 (Cortex-A7 Quad)



- Target

- Smart Phone

- Acer Liquid E2

- <http://www.acer.com.tw/ac/zh/TW/content/model/HM.HC5TA.001>



- OS

- Official Kernel & Android

- (But patch only)

- <http://us.acer.com/ac/en/US/content/drivers>

The screenshot shows a web page for the Acer V370 smartphone. At the top, there's a small image of the phone and a dropdown menu set to 'English'. Below that is a table with columns for Driver, Documents, BIOS, Patch, Application, and O.S. The first row of the table is highlighted in green and contains links for 'Download' under each category. Subsequent rows show user manuals for both Solo SIM and Duo SIM versions of the V370.

| Driver      | Documents | BIOS                   | Patch      | Application | O.S.                                |
|-------------|-----------|------------------------|------------|-------------|-------------------------------------|
| Category    | Vendor    | Description            | Version    | Size        | Date                                |
| App. Guide  | Acer      | Kernel Source Code     | 1          | 189.1 MB    | 2013/09/05 <a href="#">Download</a> |
| User Manual | Acer      | User Manual            | 1.0DUO SIM | 9.3 MB      | 2013/05/27 <a href="#">Download</a> |
| User Manual | Acer      | For Solo SIM card V370 | 1          | 8.5 MB      | 2013/05/13 <a href="#">Download</a> |
| User Manual | Acer      | For Duo SIM card V370  | 1          | 9.3 MB      | 2013/05/13 <a href="#">Download</a> |



## ● Setup Process

1. Get kernel source code & patch for Android-4.2.1
2. Merge AOSP & this patch
3. Build kernel
4. Copy drivers for Android
5. Modify Android source code
6. Build Android image
7. Write Android image to your phone

# Get kernel source code & patch for Android-4.2.1



- Download Acer's source code

- wget

- [http://global-download.acer.com/GDFiles/Document/App.%20Guide/App.%20Guide\\_Acer\\_1.0\\_A42J\\_A.zip?acerid=635013347354221361&Step1=Smartphone&Step2=Liquid%20E2&Step3=V370&OS=--&LC=en&BC=Acer&SC=EMEA\\_8](http://global-download.acer.com/GDFiles/Document/App.%20Guide/App.%20Guide_Acer_1.0_A42J_A.zip?acerid=635013347354221361&Step1=Smartphone&Step2=Liquid%20E2&Step3=V370&OS=--&LC=en&BC=Acer&SC=EMEA_8)

- Acer\_AV061\_V370\_1.007.00\_WW\_GEN1\_opensource.tar.gz

- [http://global-download.acer.com/GDFiles/Document/App.%20Guide/App.%20Guide\\_Acer\\_1.0\\_A\\_A.zip?acerid=635139433116342519&Step1=SMARTPHONE&Step2=LIQUID%20E2&Step3=V370&OS=ALL&LC=en&BC=ACER&SC=PA\\_6](http://global-download.acer.com/GDFiles/Document/App.%20Guide/App.%20Guide_Acer_1.0_A_A.zip?acerid=635139433116342519&Step1=SMARTPHONE&Step2=LIQUID%20E2&Step3=V370&OS=ALL&LC=en&BC=ACER&SC=PA_6)

- C11\_opensource\_20130903.tar

# Merge AOSP & this patch

- Download source tree of android JB from AOSP

```
> repo init -u
https://android.googlesource.com/platform/manifest
-b android-4.2.1_r1.2
> repo sync
```

- Merge source

- Over-write external,system,bootable,bionic dirs
  - from Acer\_AV061\_V370\_1.007.00\_WW\_GEN1\_opensource.tar.gz
- Add mediatek, kernel dirs
  - from C11\_opensource\_20130903.tar

# Build kernel

## ● Build kernel for MT6589

```
> export PATH=[your android dir for
MT6589]/prebuilts/gcc/linux-x86/arm/arm-linux-
androideabi-4.6/bin:$PATH
> cd [your android dir for MT6589]/kernel
> TARGET_PRODUCT=simcom89_wet_jb2
MTK_ROOT_CUSTOM=../mediatek/custom/ make
```

# Copy drivers for Android



- Some source code & driver aren't released.
  - Copy them from update image
    - wget [http://global-download.acer.com/GDFiles/OS/OS/OS\\_Acer\\_Acer\\_AV061\\_V370.1.023.00.WW.GEN1\\_A42J\\_A.zip?acerid=635158485778733078&Step1=SMARTPHONE&Step2=LIQUID%20E2&Step3=V370&OS=ALL&LC=en&BC=ACER&SC=PA\\_6](http://global-download.acer.com/GDFiles/OS/OS/OS_Acer_Acer_AV061_V370.1.023.00.WW.GEN1_A42J_A.zip?acerid=635158485778733078&Step1=SMARTPHONE&Step2=LIQUID%20E2&Step3=V370&OS=ALL&LC=en&BC=ACER&SC=PA_6)
    - OS\_Acer\_Acer\_AV061\_V370.1.023.00.WW.GEN1\_A42J\_A.zip
  - Copy system/vendor, system/lib dirs to Android device dir

| Category | Vendor | Description                                                           | Version  | Size       | Date                     |
|----------|--------|-----------------------------------------------------------------------|----------|------------|--------------------------|
| OS       | Acer   | OS update SOP. 0. Make sure th [...] Acer_AV061_V370.1.023.00.WW.GEN1 | 410.9 MB | 2013/09/27 | <a href="#">Download</a> |

# Modify Android source code 1/2



- Delete libcustom\_prop

- ./external/webkit/Android.mk

```
LOCAL_C_INCLUDES +=
 mediatek/frameworks/base/custom/inc
LOCAL_STATIC_LIBRARIES += libcustom_prop
```

- ./external/webkit/Source/WebCore/loader/FrameLoader.cpp

```
#include "custom_prop.h"
```

```
※ And "m_customXWapProfile" functions
```

# Modify Android source code 2/2



## ● Delete libcustom\_prop

- ./external/webkit/Source/WebCore/platform/network/android/SocketStreamHandle.h
- ./external/webkit/Source/WebCore/css/CSSComputedStyleDeclaration.cpp
- ./external/webkit/Source/WebCore/css/CSSParser.cpp
- ./external/webkit/Source/WebCore/loader/appcache/ApplicationCacheHost.cpp
- ./external/webkit/Source/WebCore/websockets/WebSocket.cpp
- ./external/webkit/Source/WebCore/xml/XMLHttpRequest.cpp
- ./external/webkit/Source/WebCore/websockets/WebSocketHandshake.cpp
- ./external/webkit/Source/WebKit/android/WebCoreSupport/WebViewClientError.cpp
- ./external/webkit/Source/WebCore/websockets/WebSocketChannel.cpp
- ./external/webkit/Source/WebKit/android/WebCoreSupport/SocketStreamHandleAndroid.cpp
- ./external/webkit/Source/WebKit/android/WebCoreSupport/SocketStreamHost.cpp

```
#include <cutils/xlog.h>
```

※ And “XLOG” functions

# Build Android image



```
> . build/envsetup.sh
```

```
> choosecombo
```

Build type choices are:

1. release

2. debug

Which would you like? [1] 1

Which product would you like? [full] full

Variant choices are:

1. user

2. userdebug

3. eng

Which would you like? [eng] 3

```
> make dist
```

out/dist/full-target\_files-eng.[your user name].zip

# Write Android image to your phone



1. adb push out/dist/full-target\_files-eng.[your user name].zip /sdcard/
2. Turn off your phone
3. Volume Up & Power
4. Wipe data/factory reset
5. Install zip from sdcard
6. Choose zip from sdcard
7. full-target\_files-eng.[your user name].zip
8. +++++Go Back++++
9. Reboot system now

- Freescale – i.MX6Q (Cortex-A9 Quad)

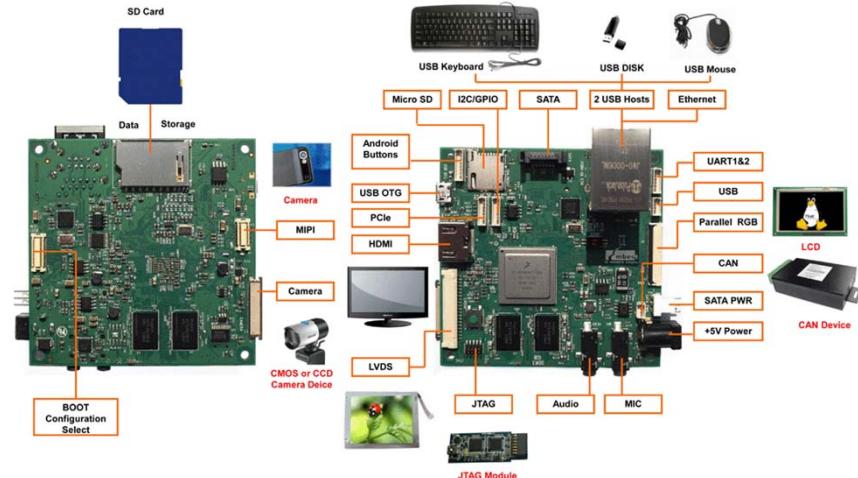
# Freescale – i.MX6Q (Cortex-A9 Quad)

## ● Target

### ● EVB

#### ● i.MX6Q SABRA Lite

- <http://www.cloudsto.com/android-mini-pc-s/rikomagic-quad-core-mk802iv-8gb-flash-bluetooth-dhl-express-shipping-detail.html>



The screenshot shows the Freescale website's search results page for the term "android". The search bar at the top contains "android". Below it, a table lists various software packages and resources related to the i.MX6Q. The table has columns for ID and Description, Type, Format, Size K, Rev #, Date Last Modified, Availability, and Favorite. Each row provides a link to download the specific file or resource. The results include patches for the Android kernel, support packages for the Linux GPU SDK, and examples like CRTC\_TOUCH\_DEMO\_ANDROID\_BSPIMX53QSBOARD.

| ID and Description                   | Type                        | Format | Size K  | Rev #         | Date Last Modified | Availability             | Favorite |
|--------------------------------------|-----------------------------|--------|---------|---------------|--------------------|--------------------------|----------|
| IMX6_JB422_100_AACPF_CODEC           | Codecs and other Algorithms | gz     | 84      | jb4.2.2_1.0.0 | 5/15/2013          | <a href="#">Download</a> | ☆        |
| IMX51_GPU_SDK_1.1                    | Board Support Packages      | tgz    | 169745  | 1.1           | 9/8/2010           | <a href="#">Download</a> | ☆        |
| IMX8_R13.4.1.04_ANDROID_PATCH        | Updates and Patches         | gz     | 120198  | r13.4.1.04    | 4/17/2013          | <a href="#">Download</a> | ☆        |
| IMX8_R13.4.03_ANDROID_PATCH          | Updates and Patches         | gz     | 120199  | r13.4-ga.03   | 4/17/2013          | <a href="#">Download</a> | ☆        |
| CRTOUCH_DEMO_ANDROID_BSPIMX53QSBOARD | Board Support Packages      | zip    | 366     | 1             | 6/28/2013          | <a href="#">Download</a> | ☆        |
| IMX51_R7_ANDROID_SOURCE_CODE         | Board Support Packages      | gz     | 131426  | r7            | 2/12/2010          | <a href="#">Download</a> | ☆        |
| IMX51_R7_ANDROID_DEMO_IMG            | Board Support Packages      | gz     | 1772972 | r7            | 2/13/2010          | <a href="#">Download</a> | ☆        |
| IMX51_R8_ANDROID_SOURCE_CODE         | Board Support Packages      | gz     | 159343  | R8            | 5/13/2010          | <a href="#">Download</a> | ☆        |

## ● OS

### ● Official Kernel & Android

#### ● With Setup Manual

- <http://www.freescale.com/webapp/search/Serp.jsp?QueryText=ndroid&assetIdResult=&attempt=0&lastQueryText=android&isFromFlex=false&getResultSet=false&assetLockedForNavigation=false&pageNum=2&iteration=2&fsrch=1&isComparison=false&showAllCategories=false&sessionChecker=cTgD1ZGfEqGCbd9IV29IKvJebizms2&getTree=false&fromTrng=false&fromPSP=false&getFilter=false&assetLocked=false&fromCust=false&showCustomCollateral=false&pageSize=25&RELEVANCE=true&SelectedAsset=Downloads&fromWebPages=false&fromASP=false&fromMobile=false&isAdvancedSearch=false&isTree=false&fromDAP=false&isResult=false>



## ● Setup Process

1. Download Source Code from Official Site
2. Patch to Android Source Code
3. Modify Android Source Code for SD Card
4. Fix 32bit files
5. Build Source Codes
  1. Android
  2. u-boot
  3. boot-image
6. Setup SD Card

# Download Source Code from Official Site



## ● Download Android

```
> repo init -u
https://android.googlesource.com/platform/
manifest -b android-4.2.2_r1
> repo sync
```

## ● Download Kernel

```
> git clone git://git.freescale.com/imx/linux-
2.6-imx.git kernel_imx
> cd kernel_imx
> git checkout jb4.2.2_1.1.0-ga
```

# Download Source Code from Official Site



## ● Download bootloader

```
> cd ../bootable
> cd bootloader
> git clone https://github.com/ldnunes/u-boot-imx.git uboot-imx
> cd uboot-imx
> git checkout jb4.2.2_1.1.0-ga
```

## ● Download Patch

- [https://www.freescale.com/webapp/Download?colCode=IMX6\\_JB422\\_110\\_ANDR\\_D\\_SRC\\_BSP&appType=license&location=null&fpfp=1&WT\\_TYPE=Board%20Support%20Packages&WT\\_VENDOR=FREESCALE&WT\\_FILE\\_FORMAT=gz&WT\\_ASSET=Downloads&sr=38&Parent\\_nodeId=from%20search&Parent\\_pageType=from%20search](https://www.freescale.com/webapp/Download?colCode=IMX6_JB422_110_ANDR_D_SRC_BSP&appType=license&location=null&fpfp=1&WT_TYPE=Board%20Support%20Packages&WT_VENDOR=FREESCALE&WT_FILE_FORMAT=gz&WT_ASSET=Downloads&sr=38&Parent_nodeId=from%20search&Parent_pageType=from%20search)
- android\_jb4.2.2\_1.1.0-ga.tar.gz

# Patch to Android Source Code



## ● Patch

```
> cd [your Android dir]
> tar zxfv android_jb4.2.2_1.1.0-ga.tar.gz
> tar zxfv android_jb4.2.2_1.1.0-ga_source.tar.gz
> cd android_jb4.2.2_1.1.0-ga_source/code/
> tar zxfv jb4.2.2_1.1.0-ga.tar.gz
> cd ../../
> source android_jb4.2.2_1.1.0-
 ga_source/code/jb4.2.2_1.1.0-
 ga/and_patch.sh
> c_patch android_jb4.2.2_1.1.0-
 ga_source/code/jb4.2.2_1.1.0-ga
 imx_jb4.2.2_1.1.0-ga
```

# Modify Android Source Code for SD Card



- device/fsl/imx6/etc/fstab.freescale

- Create New File(Please delete it, if exist.)

```
> /dev/block/mmcblk1p5 /system ext4 ro wait
> /dev/block/mmcblk1p4 /data ext4
 nosuid,nodev,nodiratime,noatime,nomblk_io_su
 bmit,noauto_da_alloc,errors=panic wait
> /dev/block/mmcblk1p6 /cache ext4
 nosuid,nodev,nomblk_io_submit wait
> /dev/block/mmcblk1p7 /device ext4
 ro,nosuid,nodev wait
```

# Modify Android Source Code for SD Card



- device/fsl/sabresd\_6dq/recovery.fstab

- Create New File(Please delete it, if exist.)

```
> /boot emmc /dev/block/mmcblk1p1
> /recovery emmc /dev/block/mmcblk1p2
> /system ext4 /dev/block/mmcblk1p5
> /cache ext4 /dev/block/mmcblk1p6
> /data ext4 /dev/block/mmcblk1p4
 reserved=32768
> /misc emmc /dev/block/mmcblk1p8
> /sdcard vfat /dev/block/mmcblk1p1
```

# Modify Android Source Code for SD Card



- device/fsl/sabresd\_6dq/vold.fstab
- b/device/fsl/sabresd\_6dq/vold.fstab
- Modify it:

```
> @@ -26,7 +26,7 @@
> #used for all usb host
> dev_mount udisk /mnt/udisk auto /devices/platform/fsl-
ehci
> #mount SDHC4 SD card /mnt/sdcard as primary storage for
MX6Q SABER_LITE RevC
> -dev_mount extsd /mnt/extsd auto
/devices/platform/sdhci-esdhc-imx.2/mmc_host/mmc1
> +dev_mount extsd /mnt/extsd auto
/devices/platform/sdhci-esdhc-imx.1/mmc_host/mmc2
> #mount SDHC3 TF card to /mnt/extsd as external storage
forMX6Q SABER_LITE RevC
> #dev_mount sdcard /mnt/sdcard 4
/devices/platform/sdhci-esdhc-imx.3/mmc_host/mmc0
```

# Fix 32bit files 1/4

## ● Setup files

### ● sudo apt-get install

```
● uuid uuid-dev zlib1g-dev liblz-dev liblzo2-2 liblzo2-de
```

## ● Download 32bits files

```
> wget https://community.freescale.com/servlet/JiveServlet/download/312916-255326/liblzo2-2_2.05-1_i386.deb.zip
> wget https://community.freescale.com/servlet/JiveServlet/download/312916-255325/liblzo2-dev_2.05-1_i386.deb.zip
> wget https://community.freescale.com/servlet/JiveServlet/download/312916-255364/mtd-utils.tgz
> wget https://community.freescale.com/servlet/JiveServlet/download/312916-258740/uuid-dev_2.20.1-1ubuntu3_i386.deb.zip
```

# Fix 32bit files 2/4

- Setup 32bits files

- Install liblzo2 & uuid

```
> unzip liblzo2-2_2.05-1_i386.deb.zip
> unzip liblzo2-dev_2.05-1_i386.deb.zip
> unzip uuid-dev_2.20.1-1ubuntu3_i386.deb.zip
> sudo mkdir /usr/lib/liblzo2_i386
> sudo mkdir /usr/lib/uuid-dev_i386
> sudo dpkg -x liblzo2-dev_2.05-1_i386.deb
 /usr/lib/liblzo2_i386/
> sudo dpkg -x liblzo2-2_2.05-1_i386.deb
 /usr/lib/liblzo2_i386/
> sudo dpkg -x uuid-dev_2.20.1-
 1ubuntu3_i386.deb /usr/lib/uuid-dev_i386/
```

# Fix 32bit files 3/4

## ● Setup 32bits files

### ● Re-link liblzo2 & uuid

```
> sudo rm liblzo2.a
> sudo rm liblzo2.so
> sudo rm liblzo2.so.2
> sudo ln -s liblzo2_i386/usr/lib/liblzo2.a
 liblzo2.a
> sudo ln -s liblzo2_i386/usr/lib/liblzo2.so.2.0.0
 liblzo2.so
> sudo ln -s liblzo2_i386/usr/lib/liblzo2.so.2.0.0
 liblzo2.so.2
> sudo ln -s uuid-dev_i386/usr/lib/i386-linux-
 gnu/libuuid.a libuuid.a
> sudo ln -s uuid-dev_i386/usr/lib/i386-linux-
 gnu/libuuid.so libuuid.so
```

# Fix 32bit files 4/4



## ● Modify

### ● external/mtd-utils/mkfs.ubifs/Android.mk

```
> LOCAL_LDFLAGS += -L /usr/lib/
> LOCAL_C_INCLUDES += /usr/lib/uuid-
 dev/usr/include/
> LOCAL_C_INCLUDES
 += /usr/lib/liblzo2_i386/usr/include
```

# Build Source Codes 1/3



## ● Build Android

```
> cd [your Android dir]
> source build/envsetup.sh
> lunch sabresd_6dq-user
> make
```

# Build Source Codes 2/3



## ● Build u-boot

```
> cd [your Android
dir]/bootable/bootloader/uboot-imx
> export ARCH=arm
> export CROSS_COMPILE=[your Android
dir]/prebuilts/gcc/linux-x86/arm/arm-eabi-
4.6/bin/arm-eabi-
> make distclean
> make mx6q_sabresd_android_config
> make
```

# Build Source Codes 3/3

## ● Build kernel

- > export ARCH=arm
- > export CROSS\_COMPILE=[your Android dir]/prebuilts/gcc/linux-x86/arm/arm-eabi-4.6/bin/arm-eabi-
- > make imx6\_android\_defconfig
- > make uImage

## ● Build boot-image

- > cd [your Android dir]
- > source build/envsetup.sh
- > lunch sabresd\_6dq-user
- > make bootimage

# Setup SD Card



- If you want to boot from SD Card, you have to make a special sector for this board.
  - This script can make the special sector.

```
> cd [your Android dir]
> cp ./out/target/product/sabresd_6dq/*.img .
> chmod +x ./device/fsl/common/tools/fsl-
 sdcard-partition.sh
> sudo ./device/fsl/common/tools/fsl-sdcard-
 partition.sh /dev/sdX
※X is your SD Card device name
```

- TI – OMAP5432 (Cortex-A15 Dual)

# TI – OMAP5432 (Cortex-A15 Dual)

- Target

- EVB

- OMAP5432-EVM

- <http://www.ti.com/tool/omap5432-evm?DCMP=omap-5432evm-130521&HQS=omap-5432evm-b-sw>

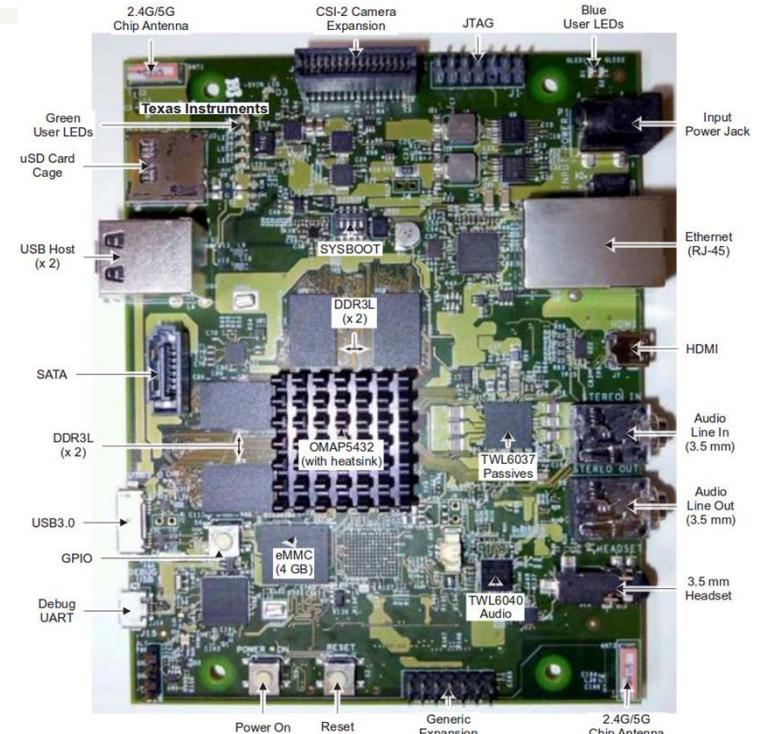
- OS

- Official Kernel & Android

- With Build Manual&Data Sheet

- <http://www.ti.com/tool/omap5432-evm-eval-sw>

- [http://omapedia.org/wiki/Panda5AJ.1.5\\_Release\\_Notes](http://omapedia.org/wiki/Panda5AJ.1.5_Release_Notes)



Software Development Kits (Evaluation-SW) for OMAP5432 processor-based EVM  
(ACTIVE) OMAP5432-EVM-EVAL-SW

| Description & Features | Technical Documents | Support & Community |
|------------------------|---------------------|---------------------|
|------------------------|---------------------|---------------------|

Order Now

| Part Number                                                                                  | Buy from Texas Instruments   | Alert Me                 | Status | OS                    | Current Version | Version Date | Core            |
|----------------------------------------------------------------------------------------------|------------------------------|--------------------------|--------|-----------------------|-----------------|--------------|-----------------|
| ANDROID-EVALUATION-SW-OMAP5432: Android Evaluation SW for OMAP5432-EVM                       | <a href="#">Get Software</a> | <a href="#">Alert Me</a> | ACTIVE | Android               | 5AJ.1.5         | May 10, 2013 | Dual Cortex A15 |
| GREENHILLS-EVALUATION-SW-OMAP5432-EVM: Green Hills Evaluation SW for OMAP5432-EVM            | <a href="#">Get Software</a> | <a href="#">Alert Me</a> | ACTIVE | Green Hills INTEGRITY | 11.0.4          | 13 May 2013  | Dual Cortex A15 |
| LINUX-UBUNTU-EVALUATION-SW-OMAP5432-EVM: Linux (Ubuntu based) Evaluation SW for OMAP5432 EVM | <a href="#">Get Software</a> | <a href="#">Alert Me</a> | ACTIVE | Linux                 | 6.00.00.07      | May 10, 2013 | Dual Cortex A15 |
| LINUX-YOCTO-EVALUATION-SW-OMAP5432-EVM: Linux (Yocto based) Evaluation SW for OMAP5432 EVM   | <a href="#">Get Software</a> | <a href="#">Alert Me</a> | ACTIVE | Linux                 | 6.01.00.01      | July 1, 2013 | Dual Cortex A15 |
| QNX-EVALUATION-BSP-OMAP5432-EVM: QNX Board Support Package for OMAP5432-EVM                  | <a href="#">Get Software</a> | <a href="#">Alert Me</a> | ACTIVE | QNX BSP               | 6.50 SP1        | May 10, 2013 | Dual Cortex A15 |



## ● Setup Process

1. Download Android Source Code
2. Build Android Source Code
3. Download kernel Source Code
4. Build kernel Source Code
5. Download u-boot Source Code
6. Build u-boot Source Code
7. Build fastboot image
8. Write fastboot images to eMMC

# Download Android Source Code



## ● Download Android

```
> cd [your Panda5 dir]
> mkdir -p mydroid
> cd mydroid
> export MYDROID=`pwd`
> repo init -u
 git://git omapzoom.org/platform/omapmanife
 st.git -b 5AJ.x -m RLS_panda5AJ.1.5.xml
> repo sync
```

# Build Android Source Code



## ● Build Android system

```
> . build/envsetup.sh
> lunch 25
> make clean
> make
```

# Download kernel Source Code



## ● Download kernel

```
> cd [your Panda5 dir]
> mkdir kernel
> git clone
 git://git.omapzoom.org/kernel/omap.git
 kernel/android-3.4
> cd kernel/android-3.4
> git checkout
 499e8d5a5aa8a271bf9c4ae404ad234b7fb22b9
 f
```

# Build kernel Source Code 1/2



## ● Build kernel

```
> export
PATH=$PATH:${MYDROID}/prebuilts/gcc/lin
ux-x86/arm/arm-eabi-4.6/bin/
> export
CROSS_COMPILE=${MYDROID}/prebuilts/gc
c/linux-x86/arm/arm-eabi-4.6/bin/arm-
eabi-
> cd [your Panda5 dir]/kernel/android-3.4
> make ARCH=arm mrproper
> make ARCH=arm android omap_defconfig
> make ARCH=arm uImage modules
```

# Build kernel Source Code 2/2



## ● Build Wi-Fi Driver

```
> export KERNEL_DIR=[your Panda5
dir]/kernel/android-3.4
> export KLIB=${KERNEL_DIR}
> export KLIB_BUILD=${KERNEL_DIR}
> cd
 ${MYDROID}/hardware/ti/wlan/mac80211/c
 ompat_wl18xx/
> make ARCH=arm
```

# Download u-boot Source Code



## ● Download u-boot

```
> cd [your Panda5 dir]
> git clone
git://git omapzoom.org/repo/omapboot.git
usbboot
> cd usbboot
> git checkout
33af7cb409b603cf7988306ab2ea70f052a9a02
b
```

# Build u-boot Source Code



## ● Build u-boot

```
> cd [your Panda5 dir]/usbboot
> make MACH=omap5 BOARD=omap5uevm
 clean
> make MACH=omap5 BOARD=omap5uevm
```

# Build fastboot image 1/

## ● Copy kernel to OTA image dir

```
> cd $MYDROID
> cp -v [your Panda5 dir]/kernel/android-
3.4/arch/arm/boot/zImage device/ti/panda5/kernel
```

## ● Copy Wi-Fi Drivers to OTA image dir

```
> cd $MYDROID/out/target/product/panda5
> mkdir -p system/lib/modules; cd system/lib/modules
> cp -fp ${MYDROID}/hardware/ti/wlan/mac80211/compat_wl18xx/compat/compat.ko .
> cp -fp
 ${MYDROID}/hardware/ti/wlan/mac80211/compat_wl18xx/net/wireless/cfg80211.ko .
> cp -fp
 ${MYDROID}/hardware/ti/wlan/mac80211/compat_wl18xx/net/mac80211/mac80211.ko .
> cp -fp
 ${MYDROID}/hardware/ti/wlan/mac80211/compat_wl18xx/drivers/net/wireless/ti/wl12xx/
wl12xx.ko .
> cp -fp
 ${MYDROID}/hardware/ti/wlan/mac80211/compat_wl18xx/drivers/net/wireless/ti/wl18xx/
wl18xx.ko .
> cp -fp
 ${MYDROID}/hardware/ti/wlan/mac80211/compat_wl18xx/drivers/net/wireless/ti/wlcore/
wlcore*.ko .
```

# Build fastboot image 2/



SIProp

## ● ReBuild Android System

```
> cd $MYDROID
> . build/envsetup.sh
> lunch 25
> rm out/target/product/panda5/*img
> rm
 out/target/product/panda5/obj/PACKAGIN
 G/systemimage_intermediates/system.img
> make
```

# Write fastboot images to eMMC 1/6



## ● Copy fastboot images to working dir

```
> cd [your Panda5 dir]
> mkdir emmc_files
> cp -v ${MYDROID}/out/target/product/panda5/*img
 emmc_files
> cp -v ${MYDROID}/device/ti/omap5sevm/boot/fastboot.sh
 emmc_files
> cp -v ${MYDROID}/out/host/linux-
 x86/bin/{simg2img,make_ext4fs,mkbootimg,fastboot,adb}
 emmc_files
> cp -v [your Panda5 dir]/kernel/android-
 3.4/arch/arm/boot/zImage emmc_files/kernel
> cp -v [your Panda5
 dir]/usbboot/out/omap5uevm/omap5uevm_GP_ES2.0_MLO
 emmc_files
> cp -v [your Panda5 dir]/usbboot/out/omap5uevm/usbboot
 emmc_files
```

# Write fastboot images to eMMC 2/6



- Register your board to udev.
  - Add the following line to ‘/etc/udev/rules.d/51-android.rules’

```
SUBSYSTEM=="usb",
ATTRS{idVendor}=="0451",
ATTRS{idProduct}=="d011", MODE="0666"
```

- Restart ‘udev’ service

# Write fastboot images to eMMC 3/6



- Connect 2 Cables to PC

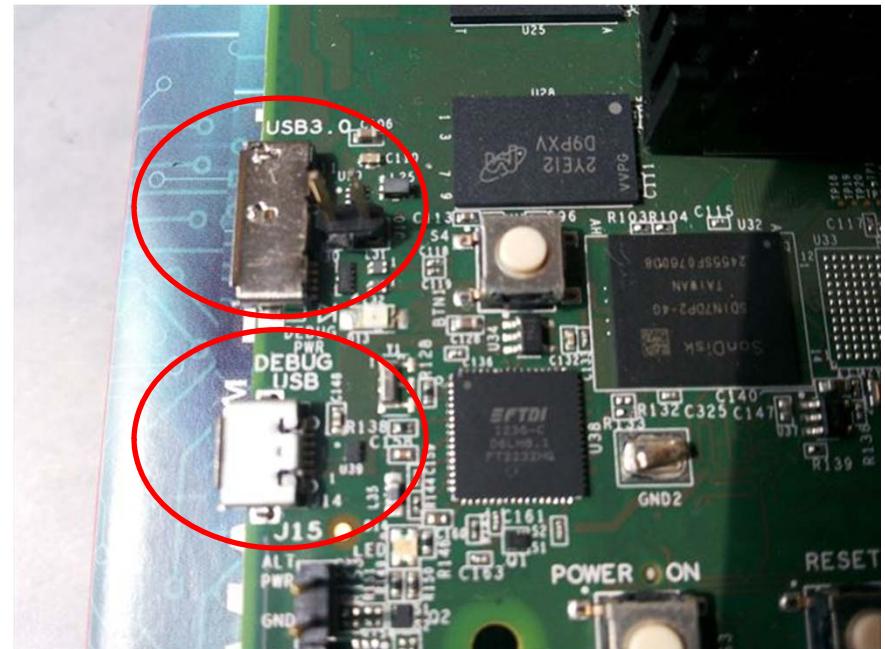
- Serial Connector

- This connector is MicroUSB.

- USB Connector for ADB

- This port is USB**3.0** connector.

- Connect **USB2.0** Cable.



# Write fastboot images to eMMC 4/6



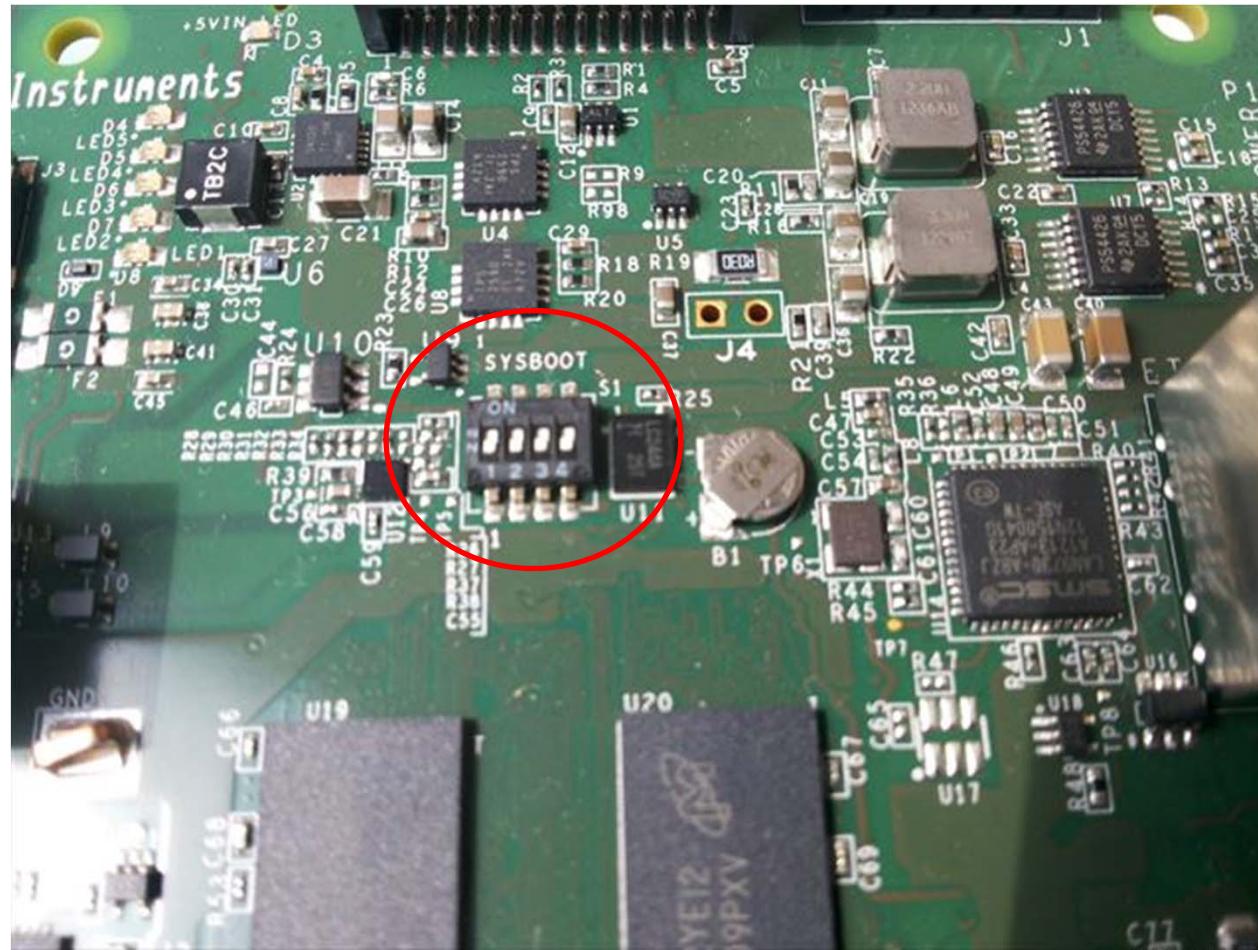
- Run fastboot program on your PC

```
> cd [your Panda5 dir]/emmc_files
> sudo ./usbbboot -f
"waiting for device.."
```

# Write fastboot images to eMMC 5/6



- Boot-up your board from eMMC
- Change boot-up dip switch to SW1–4=OFF



# Write fastboot images to eMMC 6/6



## ● Write All Images to eMMC

```
> sudo ./fastboot.sh
```

- Samsung – Exynos5 (Cortex-A15 Quad and Cortex-A7 Quad)

# Samsung – Exynos5 (Cortex-A15 Quad and Cortex-A7 Quad)

- Target

- EVB

- ODROID-XU

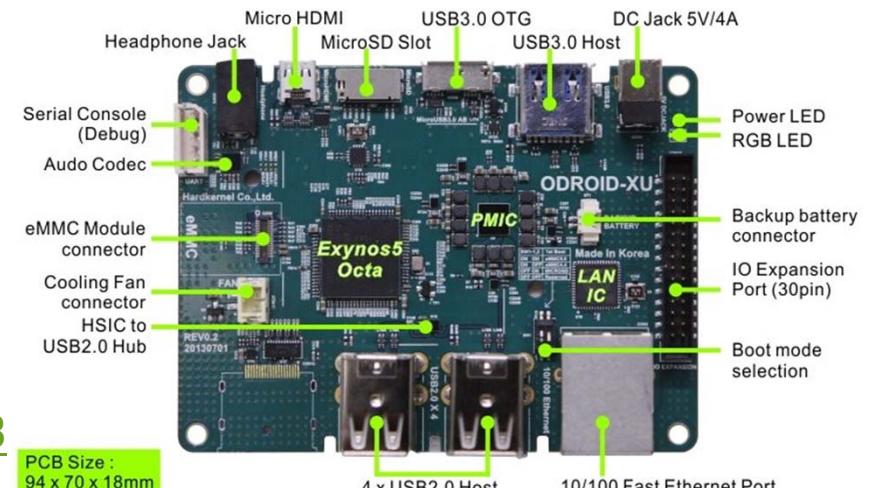
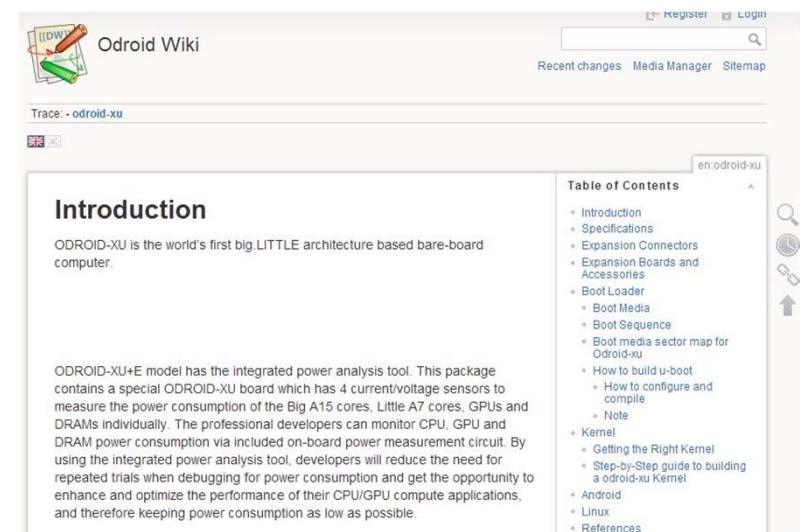
- [http://www.hardkernel.com/renewal\\_2011/products/prdt\\_info.php?g\\_code=G137510300620](http://www.hardkernel.com/renewal_2011/products/prdt_info.php?g_code=G137510300620)

- OS

- Official Kernel & Android

- With Build Manual&Data Sheet

- <http://odroid.com/dokuwiki/doku.php?id=en:odroid-xu>

The screenshot shows the Odroid Wiki page for the ODROID-XU model. The page includes:

- Introduction**: A brief description of the ODROID-XU as the world's first big.LITTLE architecture based bare-board computer.
- Table of Contents**:
  - Introduction
  - Specifications
  - Expansion Connectors
  - Expansion Boards and Accessories
  - Boot Loader
    - Boot Media
    - Boot Sequence
    - Boot media sector map for Odroid-xu
  - How to build u-boot
    - How to configure and compile
    - Note
  - Kernel
    - Getting the Right Kernel
    - Step-by-Step guide to building a odroid-xu Kernel
  - Android
  - Linux
  - References

# Samsung – Exynos5 (Cortex-A15 Quad and Cortex-A7 Quad)

## ● Setup Process

1. Build u-boot
2. Build kernel
3. Build Android
4. Setup eMMC
5. Write fastboot images to eMMC

# Build u-boot 1/2

## ● Setup Toolchains

- > wget http://dn.odroid.com/ODROID-XU/compiler/arm-eabi-4.4.3.tar.gz
- > mkdir [your toolchain dir]
- > cp arm-eabi-4.4.3.tar.gz [your toolchain dir]
- > cd [your toolchain dir]
- > tar zxvf arm-eabi-4.4.3.tar.gz

## ● Setup Paths

- > export PATH=\${PATH}:[your toolchain dir]/arm-eabi-4.4.3/bin
- > export CROSS\_COMPILE=arm-eabi-

# Build u-boot 2/2

## ● Get Source Code with kernel & Android

- [http://dn.odroid.com/ODROID-XU/Android\\_bsp/](http://dn.odroid.com/ODROID-XU/Android_bsp/)

● Please get latest Source Code from there.

### Index of /ODROID-XU/Android\_bsp/

| <a>Name</a>                                                                                                 | <a>Last modified</a> | <a>Size</a> | <a>Description</a> |
|-------------------------------------------------------------------------------------------------------------|----------------------|-------------|--------------------|
|  <a>Parent Directory</a> |                      | -           |                    |
|  <a>android.tar.gz</a>   | 21-Oct-2013 12:35    | 2.4G        |                    |
|  <a>kernel.tar.gz</a>    | 21-Oct-2013 12:27    | 104M        |                    |

*Apache/2.2.22 (Ubuntu) Server at dn.odroid.com Port 80*

## ● Build u-boot

```
> cd uboot
> make distclean
> make smdk5410_config
> make
```

# Build kernel

- Setup Toolschain & Get Source Code
  - Same with u-boot. Please see u-boot section.
- Build kernel

```
> cd kernel
> make distclean
> make odroidxu_android_defconfig
> make
```

# Build Android



- Run Build Script

```
> ./build.sh odroidxu platform
```

- If you get compile error, please do the following:
  - sudo ln -s /usr/lib/i386-linux-gnu/mesa/libGL.so.1  
/usr/lib/i386-linux-gnu/libGL.so

# Setup eMMC 1/4

- This board doesn't have boot loader on eMMC.
  - Setup boot loader to eMMC from SD Card.
- Setup Steps
  1. Download eMMC setup image for SD Card.
    1. It's disk image file for SD Card.
  2. Write this image to SD Card.
  3. Boot-up your board from SD Card.
  4. Setup boot loader to eMMC automatically.

# Setup eMMC 2/4

Download eMMC setup image for SD Card.

1. It's disk image file for SD Card.

Write this image to SD Card.

Boot-up your board from SD Card.

Setup boot loader to eMMC automatically.

# Setup eMMC 3/4



- Download eMMC setup image for SD Card

```
> wget http://dn.odroid.com/ODROID-
XU/Recovery_0.4/emmc_installer.zip
```

- And unzip.

- Write this image to SD Card

```
> dd if=./emmc_installer.img of=/dev/sdX
```

- `/dev/sdX` ‘s X is your SD Card device name.

# Setup eMMC 4/4

- Boot-up your board from SD Card
  - Change boot-up dip switch to SW1=OFF, SW2=ON.

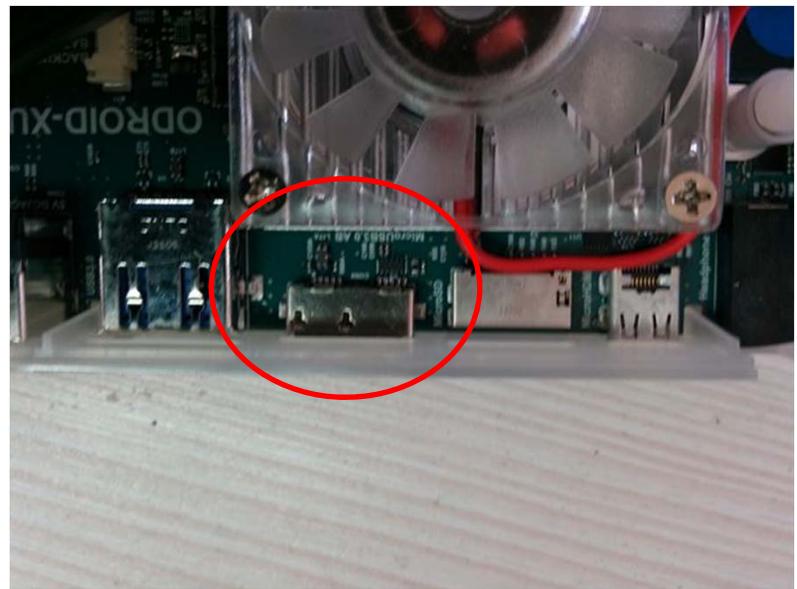
| SW1-1,2 | 1st Boot media |
|---------|----------------|
| ON ON   | eMMC5.0        |
| ON OFF  | eMMC4.4        |
| OFF ON  | MicroSD card   |
| OFF OFF | Reserved       |

- Setup boot loader to eMMC automatically
  - Put SD Card on your board.
  - Turn on power.
  - Wait a minute.

# Write fastboot images to eMMC 1/5



- Connect 2 Cables to PC
  - Serial Connector
    - This cable is optional parts.
    - If you want it, you should buy with board.
  - USB Connecter for ADB
    - This port is **USB3.0** connecter.
    - Connect **USB2.0** Cable.



# Write fastboot images to eMMC 2/5



- Change fastboot mode.
  - Show “Press ‘Enter’ or ‘Space’ to stop auto boot”
  - Press Enter’ or ‘Space’

```
U-Boot 2012.07 (Sep 23 2013 - 16:28:46) for Exynos5410
CPU: Exynos5410 Rev2.3 [Samsung SOC on SMP Platform Base on ARM CortexA15]
APLL = 900MHz, KPLL = 600MHz
MPPLL = 532MHz, BPPLL = 800MHz
DRAM: 2 GiB
WARNING: Caches not enabled

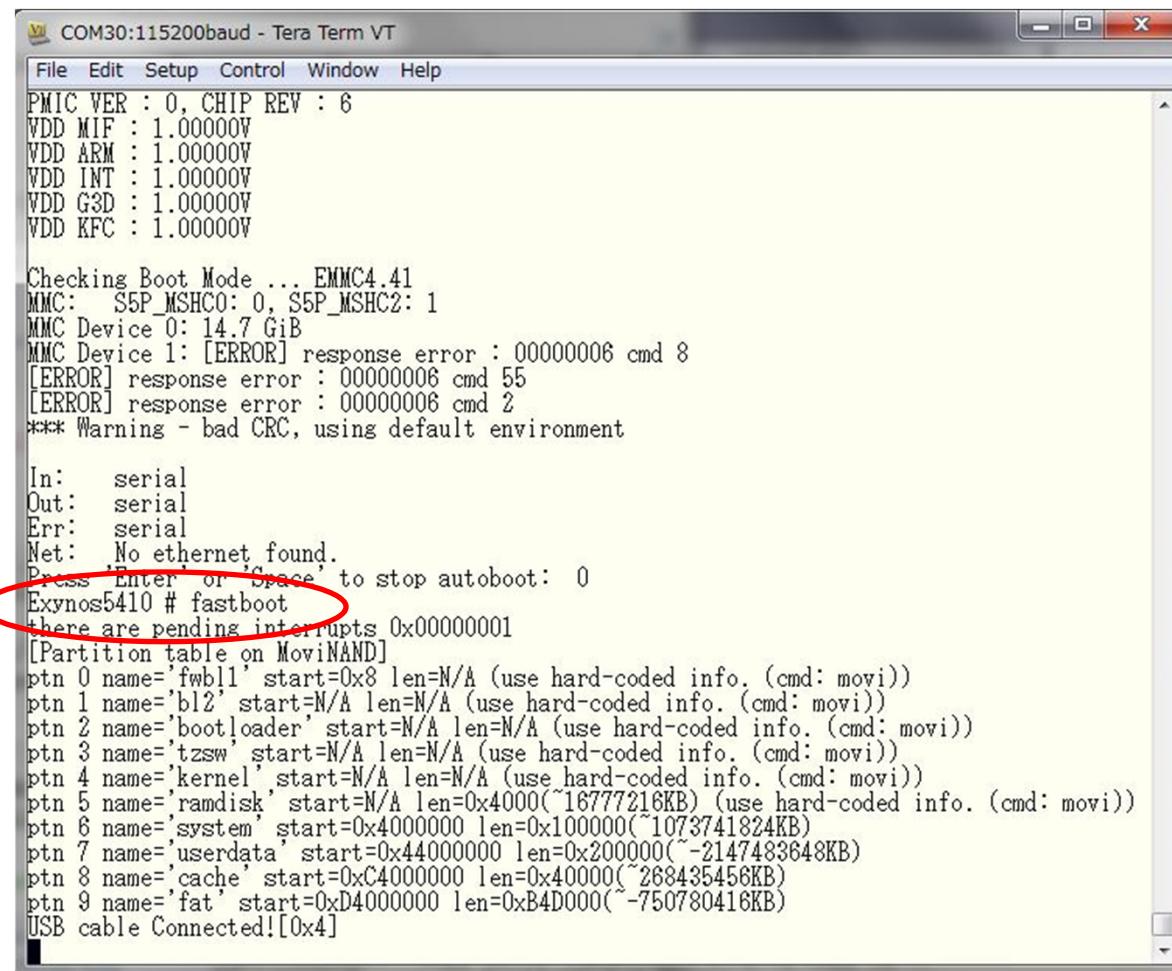
TrustZone Enabled BSP
BL1 version:
PMIC VER : 0, CHIP REV : 6
VDD MIF : 1.00000V
VDD ARM : 1.00000V
VDD INT : 1.00000V
VDD G3D : 1.00000V
VDD KFC : 1.00000V

Checking Boot Mode ... EMMC4.41
MMC: S5P_MSHC0: 0, S5P_MSHC2: 1
MMC Device 0: 14.7 GiB
MMC Device 1: [ERROR] response error : 00000006 cmd 8
[ERROR] response error : 00000006 cmd 55
[ERROR] response error : 00000006 cmd 2
*** Warning - bad CRC, using default environment

In: serial
Out: serial
Err: serial
Net: No ethernet found.
Press 'Enter' or 'Space' to stop autoboot: 0
Exynos5410 #
```

# Write fastboot images to eMMC 3/5

- Change fastboot mode.
- Input “fastboot”



```
PMIC VER : 0, CHIP REV : 6
VDD MIF : 1.00000V
VDD ARM : 1.00000V
VDD INT : 1.00000V
VDD G3D : 1.00000V
VDD KFC : 1.00000V

Checking Boot Mode ... EMMC4.41
MMC: S5P_MSHC0: 0, S5P_MSHC2: 1
MMC Device 0: 14.7 GiB
MMC Device 1: [ERROR] response error : 00000006 cmd 8
[ERROR] response error : 00000006 cmd 55
[ERROR] response error : 00000006 cmd 2
*** Warning - bad CRC, using default environment

In: serial
Out: serial
Err: serial
Net: No ethernet found.
Press 'Enter' or 'Space' to stop autoboot: 0
Exynos5410 # fastboot
there are pending interrupts 0x00000001
[Partition table on MoviNAND]
ptn 0 name='fwbl1' start=0x8 len=N/A (use hard-coded info. (cmd: movi))
ptn 1 name='bl2' start=N/A len=N/A (use hard-coded info. (cmd: movi))
ptn 2 name='bootloader' start=N/A len=N/A (use hard-coded info. (cmd: movi))
ptn 3 name='tzsw' start=N/A len=N/A (use hard-coded info. (cmd: movi))
ptn 4 name='kernel' start=N/A len=N/A (use hard-coded info. (cmd: movi))
ptn 5 name='ramdisk' start=N/A len=0x4000(~16777216KB) (use hard-coded info. (cmd: movi))
ptn 6 name='system' start=0x4000000 len=0x100000(~1073741824KB)
ptn 7 name='userdata' start=0x44000000 len=0x200000(~-2147483648KB)
ptn 8 name='cache' start=0xC4000000 len=0x40000(~268435456KB)
ptn 9 name='fat' start=0xD4000000 len=0xB4D000(~-750780416KB)
USB cable Connected![0x4]
```

# Write fastboot images to eMMC 4/5



- Register your board to udev.
  - Add the following line to ‘/etc/udev/rules.d/51-android.rules’

```
SUBSYSTEM=="usb",
ATTRS{idVendor}=="18d1", MODE=="0666"
```

- Restart ‘udev’ service

# Write fastboot images to eMMC 5/5



## ● Put your systems to your board.

### ● Use fastboot

● If your system doesn't have fastboot, please do:

● `sudo apt-get install android-tools-fastboot`

```
> fastboot flash xloader uboot/MLO
> fastboot flash bootloader uboot/u-boot.bin
> fastboot flash kernel
 kenel/arch/arm/boot/zImage
 ☒ If you write
 boot partition, it doesn't need.
> fastboot flash boot
 out/target/product/odroidxu/boot.img
> fastboot flash system
 android/out/target/product/odroidxu/syste
 m.img
```

- Qualcomm – krait 400 (Like Cortex-A15 Quad and Cortex-A7 Quad)

# Qualcomm – krait 400 (Like Cortex-A15 Quad and Cortex-A7 Quad)

- Target

- Tablet

- Nexus 7(2013)

- <http://www.google.com/nexus/7/>



- OS

- Official Kernel & Android

- With Build Manual

- <http://www.rikomagic.co.uk/forum/viewtopic.php?f=6&t=4696>

**ANDROID** | Source Devices Accessories Compatibility

**Welcome to the Android Open Source Project!**

Android is an open-source software stack for a wide range of mobile devices and a corresponding open-source project led by Google. Here you can find the information and source code you need to learn more about the Android platform. From there you can create custom variants of the Android software stack, port devices and accessories to the Android platform, and ensure your devices are compatible with the Android compatibility definition.

**UPDATES**

**Source Code Available for Android**

Android is an open-source software stack for a wide array of mobile devices with different form factors. We created Android in response to our own experiences launching mobile apps. We wanted to make sure there was no central point of failure so no industry player can restrict or control the innovations of any other. That's why we created Android and made its source code open.

**GETTING STARTED**

**Explore the Source**

Get the complete Android platform and modify and build it to suit your needs. You can also contribute to the Open Source Project to make your changes available to everyone else in the Android ecosystem.

**Port Android to Devices**

Port the latest Android platform and create compelling devices that your customers want.

**Build Accessories**

Sometimes, a device can't do it all. Tap into Android's open accessory standard and build accessories to complement the wide variety of Android-powered devices.

**Get Compatible**

Being Android-compatible lets you offer custom features but still give users and developers a consistent and standard experience across all Android-powered devices. Android provides guidance and a test suite to verify your Android compatibility.

Except as noted, this content is licensed under Creative Commons Attribution 2.5. For details and restrictions, see the Content License.  
[About Android](#) | [Community](#) | [Legal](#)

2,094

Qualcomm – krait 400 (Like Cortex-A15 Quad and Cortex-A7 Quad)



# Target

## Tablet

## ● Nexus 7(2013)

- <http://www.google.com/nexus/7/>



 OS

JCROM

# With Build Manual

● [https://sites.google.com/  
site/jcromproject/](https://sites.google.com/site/jcromproject/)



# Qualcomm – krait 400 (Like Cortex-A15 Quad and Cortex-A7 Quad)



## ● Setup Process

1. Download Source Code
2. Download BSP from site & your real device
3. Download & Build kernel
4. Make OTA image
5. Write new OTA image

# Download Source Code



## ● Download from JCROM's repository

```
> repo init -u
https://bitbucket.org/sola/jcrom_manifest -
m jcrom_jb-4.3-master.xml
> repo sync
```

# Download BSP from site & your real device 1/2

- Register your board to udev.
  - Add the following line to ‘/etc/udev/rules.d/51-android.rules’

```
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1",
ATTR{idProduct}=="4e40", MODE="0666" # Bootloader
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1",
ATTR{idProduct}=="d001", MODE="0666" # Recovery
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1",
ATTR{idProduct}=="4e43", MODE="0666" # PTP media, USB debug off
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1",
ATTR{idProduct}=="4e44", MODE="0666" # PTP media, USB debug on
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1",
ATTR{idProduct}=="4e41", MODE="0666" # MTP media, USB debug off
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1",
ATTR{idProduct}=="4e42", MODE="0666" # MTP media, USB debug on
SUBSYSTEM=="usb", ATTR{idVendor}=="18d1",
ATTR{idProduct}=="4ee2", MODE="0666" # MTP media, USB debug on
```

- Restart ‘udev’ service

# Download BSP from site & your real device 2/2



## ● Download BSPs

### ● Use adb

- If your system doesn't have adb, please do:

- sudo apt-get install android-tools-fastboot

```
> cd [your JCROM dir]/android/device/asus/flo
> ./download-blobs.sh

> cd [your JCROM
dir]/jcrom/asus/flo/proprietary/
> ./extract-files.sh
```

# Download & Build kernel



## ● Download & Build kernel

```
> cd [your JCROM dir]
> export ARCH=arm
> export CROSS_COMPILE=[your JCROM
dir]/android/prebuilts/gcc/linux-x86/arm/arm-eabi-
4.6/bin/arm-eabi-
> git clone
https://android.googlesource.com/kernel/msm.git -b
android-msm-flo-3.4-jb-mr2 kernel/nexus72
> cd kernel/nexus72
> make flo_defconfig
> make

> cp arch/arm/boot/zImage [your JCROM
dir]/android/device/asus/flo-kernel/kernel
```

# Make OTA image



- Make OTA image

- This OTA image is update image for Android.

```
> cd [your JCROM dir]/android/
> source build/envsetup.sh
> lunch aosp_flo-user
> make otapackage
```

- This file name is

- aosp\_flo-ota-eng.[your Linux's User name].zip

# Write new OTA image 1/



- If your phone has OEM lock, please unlock it.

```
> adb reboot bootloader
> fastboot oem unlock
```

- Change Recovery image.

```
> wget
http://download2.clockworkmod.com/recover
ies/recovery-clockwork-6.0.4.3-flo.img
> fastboot flash recovery recovery-clockwork-
6.0.4.3-flo.img
> fastboot boot recovery-clockwork-6.0.4.3-
flo.img
```

# Write new OTA image 2/

- Push your new OTA image to your N7

```
> adb push [your JCROM
dir]/android/out/target/product/flo/aosp_flo
-ota-eng.[your Linux's User name].zip
/sdcard/
```

- If you want to use Google Apps

```
> wget http://goo.im/gapps/gapps-jb-
20130813-signed.zip
> adb push gapps-jb-20130813-signed.zip
/sdcard/
```

# Write new OTA image 3/

- Push your new OTA image to your N7

```
> adb push [your JCROM
dir]/android/out/target/product/flo/aosp_flo
-ota-eng.[your Linux's User name].zip
/sdcard/
```

- If you want to use Google Apps

```
> wget http://goo.im/gapps/gapps-jb-
20130813-signed.zip
> adb push gapps-jb-20130813-signed.zip
/sdcard/
```

# Write new OTA image 4/

1. Turn off your phone
2. Volume Up & Power
3. Wipe data/factory reset
4. Install zip from sdcard
5. Choose zip from sdcard
6. aosp\_flo-ota-eng.[your Linux's User name].zip
7. Choose zip from sdcard
8. gapps-jb-20130813-signed.zip
9. +++++Go Back++++
10. Reboot system now

● Enjoy your Original System!!!