

DEVELOPMENT GUIDE V1.0  
DS2-2GRam Android BSP  
V2.2.2

## Copyright

Copyright © 2013 VIA Technologies Incorporated. All rights reserved.

No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise without the prior written permission of VIA Technologies, Incorporated.

## Trademarks

All brands, product names, company names, trademarks and service marks are the property of their respective holders.

## Disclaimer

VIA Technologies makes no warranties, implied or otherwise, in regard to this document and to the products described in this document. The information provided in this document is believed to be accurate and reliable as of the publication date of this document. However, VIA Technologies assumes no responsibility for the use or misuse of the information in this document and for any patent infringements that may arise from the use of this document. The information and product specifications within this document are subject to change at any time, without notice and without obligation to notify any person of such change.

VIA Technologies, Inc. reserves the right to make changes to the products described in this manual at any time without prior notice.

## Revision History

Version	Date	Remarks
1.0	2015/5/19	Release for DS2-2GRam v2.2.2 Android BSP

## Table of Contents

<b>1.</b>	<b>Introduction .....</b>	<b>3</b>
1.1.	Overview .....	3
1.2.	Package Content .....	3
<b>2.</b>	<b>Building BSP.....</b>	<b>4</b>
2.1.	Setting Development Environment.....	4
2.2.	Prepare the BSP Source Tree.....	7
2.3.	Build and Install BSP .....	8
2.3.1.	Build U-Boot .....	8
2.3.2.	Build Kernel .....	8
2.3.3.	Build Android OS .....	8
2.3.4.	Build S3 Driver Package .....	9
2.3.5.	Build Android Image .....	9
<b>3.</b>	<b>Making System Booting Media .....</b>	<b>11</b>
3.1.	Update DS2-2GRAM Firmware .....	11
3.2.	Booting System to U-Boot Shell Environment.....	11
3.3.	Constructing the Bootable Media .....	12
3.3.1.	Flashing bootloader .....	12
3.3.2.	EMMC Booting .....	13
3.4.	Booting Android OS .....	14
3.4.1	Auto-Flash the bootloader .....	14
3.4.2	EMMC Booting .....	15
<b>4.</b>	<b>Functionality.....</b>	<b>16</b>
<b>5.</b>	<b>Debug Message.....</b>	<b>17</b>
5.1.	U-Boot Environment.....	17
5.2.	U-Boot Parameters Example.....	18
<b>6.</b>	<b>Clone-View Rotation.....</b>	<b>19</b>
6.1.	Enable Clone-View Dual Rotation Function .....	19
6.1.1	Replace the configuration files. ....	19
6.1.2	Rebuild the Graphic Driver.....	19
6.1.3	Rebuild the Android Image.....	20

7.	Change Display Mode .....	21
7.1.	Display Setting .....	21
7.2.	Display Setting Example .....	21
Appendix A: Definitions .....		22
Appendix B: BSP Source Reference .....		23
Appendix C: Notification .....		24
1.	Formatting SD-Card .....	24
2.	Development Option.....	24
3.	ADB function .....	26

# 1. Introduction

## 1.1. Overview

This Development Guide provides a practical introduction for the VIA Android<sup>TM</sup> DS2-2GRAM platform. It helps the developers to build the development environment as soon as possible. Developers can construct the framework and develop the application, as well as the tools for development, testing, and publishing software on the platform. In addition, this documentation helps the developers to understand the VIA DS2-2GRAM platform and it provides qualified Android BSP for system product.

## 1.2. Package Content

This BSP package includes three parts:

**BSP:** BSP source code. The source code package is composed of U-boot source code, Linux Kernel source code and Android source code.

**EVK:** Includes the Android evaluation image and the tools.

**Documents:** Includes evaluation guide, development guide (this document) and any other documents required for development.

## 2. Building BSP

The Android build system does not compile kernel on-fly. It only contains a prebuilt kernel binary which will be added to the target image. This approach may be good enough for the Arm Emulator target, but not suitable for DS2-2GRAM platforms. The DS2-2GRAM platforms have various hardware features. The kernel binary and its modules may need to be adjusted at compile time or runtime.

This section will describe the process of how to setup Building Environment for DS2-2GRAM Android BSP (Board Support Packages) and how to compile each component in the BSP.

### 2.1. Setting Development Environment

1. Operating System: Ubuntu 10.04 LTS 64-Bits.

2. Install Tool Chain (Sourcery CodeBench Lite 2011.09-70 for ARM GNU/Linux):

1. Unpack arm-2011.09-70-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2 to <TOOLPATH>
2. Add path "<TOOLPATH>/arm-2011.09/bin" in environment setting

Example:

```
$ mkdir tools  
$ tar -jxvf arm-2011.09-70-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2 -C tools  
$ vi .bashrc  
add: export PATH=/home/xxx/tools/arm-2011.09/bin:$PATH  
save and exit  
reboot system
```

3. Download Development Tools:

Step1. Download the JDK executable binary from Oracle official website.

```
http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javase6-  
419409.html
```

**Java SE Development Kit 6u38**

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux x86	65.43 MB	<a href="#">jdk-6u38-linux-i586.rpm.bin</a>
Linux x86	68.45 MB	<a href="#">jdk-6u38-linux-i586.bin</a>
Linux x64	65.66 MB	<a href="#">jdk-6u38-linux-x64-rpm.bin</a>
<b>Linux x64</b>	<b>68.72 MB</b>	<a href="#">jdk-6u38-linux-x64.bin</a>
Solaris x86	68.36 MB	<a href="#">jdk-6u38-solaris-i586.sh</a>
Solaris x86 (SVR4 package)	119.96 MB	<a href="#">jdk-6u38-solaris-i586.tar.Z</a>
Solaris SPARC	73.35 MB	<a href="#">jdk-6u38-solaris-sparc.sh</a>
Solaris SPARC (SVR4 package)	124.72 MB	<a href="#">jdk-6u38-solaris-sparc.tar.Z</a>
Solaris SPARC 64-bit	12.13 MB	<a href="#">jdk-6u38-solaris-sparcv9.sh</a>
Solaris SPARC 64-bit (SVR4 package)	15.41 MB	<a href="#">jdk-6u38-solaris-sparcv9.tar.Z</a>
Solaris x64	8.45 MB	<a href="#">jdk-6u38-solaris-x64.sh</a>
Solaris x64 (SVR4 package)	12.17 MB	<a href="#">jdk-6u38-solaris-x64.tar.Z</a>
Windows x86	69.74 MB	<a href="#">jdk-6u38-windows-i586.exe</a>
Windows x64	59.8 MB	<a href="#">jdk-6u38-windows-x64.exe</a>

[Back to top](#)

## Step2. Copy binary the library folder and execute the binary

```
# cp jdk-xxxx-linux-xxx.bin /usr/lib
# cd /usr/lib
# ./jdk-xxxx-linux-xxx.bin
```

jdk-xxxx-linux-xxx.bin is the JDK binary you download from Step1

```
jdk1.8.0_38/          jdk 6u38 linux x64 bin
root@bios-desktop:/usr/lib# ./jdk-6u38-linux-x64.bin
```

## Step3. Configure Environment Variable.

To edit the file `/root/.bashrc` and modify as follow:

```
export JAVA_HOME=/[java install directory]/jdkx.xx.x_xx
export CLASSPATH=.:$JAVA_HOME/lib
export PATH=$JAVA_HOME/bin:$PATH
```



### Note:

“`x.xx.x_xx`” is the version number of sun java JDK.  
java install directory: directory of JAVA

## Step4. Verify the JAVA version

```
# java version
```

```
root@bios-desktop:~# java -version
java version "1.6.0_38"
Java(TM) SE Runtime Environment (build 1.6.0_38-b05)
Java HotSpot(TM) 64-Bit Server VM (build 20.13-b02, mixed mode)
root@bios-desktop:~# █
```

#### 4. Install Dependent Library

```
Dependent library: git-core gnupg flex bison gperf build-essential zip curl zlib1g-dev libc6-dev
lib32ncurses5-dev ia32-libs x11proto-core-dev libx11-dev lib32readline5-dev lib32z-dev libgl1-mesa-dev
g++-multilib mingw32 tofrodos python-markdown libxml2-utils xsltproc u-boot-mkimage
```

## 2.2. Prepare the BSP Source Tree

The DS2-2GRAM Android BSP includes several Packages and all these Packages are packed as tgz files. To build the BSP, please unpack the sources of the BSP Packages by following steps:

### 1. Unpack U-Boot Source:

```
$ tar -zxf <<U-Boot_Source_Name>>.tgz -C << Uboot_DIR >>
```

P.S. <<U-Boot\_Source\_Name>>: U-Boot Reference Source Code (See Appendix B for more details)  
<<Uboot\_DIR>>: Directory of Uboot Source Code

### 2. Unpack Linux Kernel Source:

```
$ tar -zxf <<Kernel_Source_Name>>.tgz -C << Kernel _DIR >>
```

P.S. <<Kernel\_Source\_Name>>: Linux Kernel Reference Source (See Appendix B for more details)  
<<Kernel \_DIR>>: Directory of Linux Kernel Source Code

### 3. Unpack S3 Driver Package source.

```
$ tar -zxf <<S3Driver_Source_Name>>.tgz -C << S3Driver_DIR >>
```

P.S. <<S3Driver\_Source\_Name>>: S3 Driver Package Reference Source (See Appendix B for more details)  
<<S3Driver\_DIR>>: Directory of S3 Driver Package

### 4. Unpack Android Rootfs source.

```
$ tar -zxf <<Android_Source_Name>>.tgz -C << Android_DIR >>
```

P.S. <<Android\_Source\_Name>>: Android Rootfs Reference Source (See Appendix B for more details)  
<<Android\_DIR>>: Directory of Android Source Code

## 2.3. Build and Install BSP

This sub-section will describe how to build each component in DS2-2GRAM Android BSP. Please see Appendix B for the Reference Source Code of each component in current version of BSP.

### 2.3.1. Build U-Boot

The u-boot will initiate basic hardware and load kernel into memory.

```
$ cd <<Uboot_DIR>>/  
=====  
$ ./2g_build_elite1000_evm_emmc_config.sh
```

The U-Boot Binary (i.e., u-boot.bin) will be generated in <<Uboot\_DIR>>

P.S. <<Uboot\_DIR>>: Directory of Uboot Source Code

### 2.3.2. Build Kernel

The Android file system works based on Linux kernel. Following are the reference commands to build DS2-2GRAM Linux Kernel:

```
$ cd <<Kernel_DIR>>/  
=====  
$ ./build_emmc_dt.sh
```

The Linux Kernel Image will be generated in following folder:

<<Kernel\_DIR>>/arch/arm/boot

P.S. <<Kernel\_DIR>>: Directory of Linux Kernel Source Code

### 2.3.3. Build Android OS

The Android Image should be built according to the Booting Strategy. Following are the reference commands to build VT6080 Android Framework:

```
$ cd <<Android_DIR>>/  
=====  
$ source build/envsetup.sh  
$ lunch 13  
$ make
```

P.S. <<Android\_DIR>>: Directory of Android Source Code

### 2.3.4. Build S3 Driver Package

The S3 Driver Package includes GFX and Audio Driver for Android OS. Following are the reference commands to build S3G Driver Package for DS2-2GRAM Platform.

```
$ cd <<S3Driver_DIR>>/  
$ export S3ANDROID_DIR=<< S3Driver_DIR >>  
$ export LINUXDIR=<< Kernel_DIR >>  
$ ./build.sh
```

Following Driver Binary will be generated in <<S3Driver\_DIR>>:

```
S3_audio.ko  
S3g_core.ko  
S3g.ko
```

```
$ export ANDROID_SYS=<< Android_DIR >>/out/target/product/elite1000  
$ ./install.sh
```

P.S. <<S3Driver\_DIR>>: Directory of S3 Driver Package  
<<Kernel\_DIR>>: Directory of Linux Kernel Source Code  
<<Android\_DIR>>: Directory of Android Source Code

### 2.3.5. Build Android Image

The Android File System should be re-built to Image format. Following are the reference commands to build the DS2-2GRAM Android EMMC Image:

```
$ cd <<Android_DIR>>/  
  
$ rm -f out/target/product/elite1000/kernel  
$ rm -f out/target/product/elite1000/boot.img  
$ rm -f out/target/product/elite1000/system.img  
$ rm -f out/target/product/elite1000/userdata.img  
$ rm -f out/target/product/elite1000/recovery.img  
$ rm -f out/target/product/elite1000/cache.img  
  
$ cp -f <<Kernel_DIR>>/arch/arm/boot/zImage device/s3graphics/elite1000/kernel  
$ cp -f <<S3Driver_DIR>>/s3g.ko device/s3graphics/elite1000/  
$ cp -f <<S3Driver_DIR>>/s3g_core.ko device/s3graphics/elite1000/  
$ cp -f <<S3Driver_DIR>>/s3_audio.ko device/s3graphics/elite1000/  
$ cp -f <<S3Driver_DIR>>/audio_firmware device/s3graphics/elite1000/  
$ source build/envsetup.sh  
$ lunch 13
```

```
$ make
```

```
$ cd <<SD-Card_DIR>>/  
$ cp <<Kernel _DIR>>/arch/arm/boot/elite1000-emmc.dtb ./  
$ cp <<Android_DIR>>/out/target/product/elite1000/boot.img ./  
$ cp <<Android_DIR>>/out/target/product/elite1000/system.img ./  
$ cp <<Android_DIR>>/out/target/product/elite1000/userdata.img ./  
$ cp <<Android_DIR>>/out/target/product/elite1000/recovery.img ./  
$ cp <<Android_DIR>>/out/target/product/elite1000/cache.img ./
```

Now the Android image is done. Please refer to next Section for more details of booting Android through EMMC

P.S. <<SD-Card\_DIR>>: Root Directory of SD Card

<<Android\_DIR>>: Directory of Android Source Code

<<S3Driver\_DIR>>: Directory of S3 Driver Package

<<Kernel \_DIR>>: Directory of Linux Kernel Source Code

# 3. Making System Booting Media

DS2-2GRAM can be booted through SD-Card. This Chapter will describe how to construct the System Booting Media for DS2-2GRAM Android OS.

## 3.1. Update DS2-2GRAM Firmware

The DS2-2GRAM System Firmware (including E-Loader and U-Boot) can be updated through the "Update Image" in the EVK Folder of BSP. Please copy the "vbsinst\_emmc.tgz" and decompress the files to the Root Folder of the EXT2-Formatted SD-Card, the system will flush the Update Firmware to SPI ROM of DS2-2GRAM Platform automatically.

Note: Please DO NOT put the SD-Card with "Update Image" into the system if you don't need to update the System Firmware.

## 3.2. Booting System to U-Boot Shell Environment

DS2-2GRAM supports SD-Card and EMMC Booting. To select the Boot Path, the related Boot-Scripts should be loaded to Memory in U-Boot Stage. To boot the system in U-Boot Shell Environment, please press any key during the Auto-Boot Count-Down Message shown up in Console:

```
148992      402653184(      384M)  cache
=====
No existing device info found.
Setting serial number from constant (no dieid info)
fastboot serial_number = 00123
Returning key pressed false
boot_method is 1

fbt preboot: request for a normal boot
Hit any key to stop autoboot: 0
S3 #
S3 #
```

Please refer Chapter 5 for more details of Debugging in U-Boot Environment.

## 3.3. Constructing the Bootable Media

### 3.3.1. Flashing bootloader

#### A. Preparation:

- (1) EXT2-formatted SD-Card (<<SD-Card\_DIR>>: SD-Card Root Directory)
- (2) Update-Package for Auto-Update Mechanism (refer to Document "Elite1000\_Auto-Update\_Tool\_User\_Guide\_v1.3.pdf")
- (3) Update Package for bootloader (vbspinst\_emmc\_bootloader.tgz)

#### B. Construction Steps: (Make sure SD-Card is cleaned before following steps)

```
tar zxvf vbspinst_emmc_bootloader.tgz -C <<SD-CARD_DIR>>
```

#### Note:

After you construct your bootable media. Your SD-Card should includes the component as follows.

**Binary:** uboot.bin, e-loader.bin, timing\_table.bin

**Uboot scripts:** scriptcmd, other\_env.uimg

**Bootloader flashing scripts:** bootloader\_setup.uimg

### 3.3.2. EMMC Booting

#### A. Preparation:

- (1) EXT2-formatted SD-Card includes Android EMMC image(<<SD-Card\_DIR>>; SD-Card Root Directory)
- (2) Update Script for Auto-Update Mechanism (refer to Document "Elite1000\_Auto-Update\_Tool\_User\_Guide\_v1.3.pdf")
- (3) Android Image (built by Chapter 2.)

#### B. Construction Steps: (Make sure SD-Card includes all Android Image)

```
$ cd <<SD-CARD_DIR>>/  
$ cp <<Update_Package_DIR>>/scriptcmd ./  
$ cp <<Update_Package_DIR>>/img_setup.uimg ./  
$ cp <<Update_Package_DIR>>/other_env.uimg ./
```

#### Note:

After you construct your bootable media. Your SD-Card should includes the component as follows.

**Android Image:** boot.img, system.img, cache.img, userdata.img, recovery.img, elite1000-emmc.dtb

**Uboot scripts:** scriptcmd, other\_env.uimg

**Android Image Installation scripts:** img\_setup.uimg

## 3.4. Booting Android OS

### 3.4.1 Auto-Flash the bootloader

#### A. Preparation:

- (1) EXT2-formatted SD-Card
- (2) Update Package for bootloader (vbspinst\_emmc\_bootloader.tgz)

#### B. Installation Steps:

Step1. Format the SD Card to EXT2-format. (See Appendix C Formatting SD-Card Part)

Step2. Extract the compress file into SD card

```
tar zxvf vbspinst_emmc_bootloader.tgz -C <<SD-CARD_DIR>>
```

Step3. Eject SD card from PC side.

Step4. Insert SD card to DS2 platform which is connected with HDMI monitor and boot the MB.

Step5. Wait until the Console shows "**BSP updated done!! PLEASE REBOOT SYSTEM!!**"

(Please power-off system manually)

### 3.4.2 EMMC Booting

#### A. Preparation:

- (1) EXT2-formatted SD-Card
- (2) Update Package for Android Image (vbspinst\_emmc\_android\_4\_3.img.tgz)

Note.

If you want to install android image build by yourself. Please refer Section 2.3 Build and Install BSP to construct your android image.

#### B. Installation Steps:

Step1. Create Bootable SD Card

```
tar zxvf vbspinst_emmc_emmc_android_4_3_img.tgz -C <<SD-CARD_DIR>>
```

Step2. Eject SD Card from PC Side

Step3. Insert the Installation SD-Card in SD0 Slot

Step4. Power-on and boot System

Step5. Wait until the Console shows "BSP updated done!! PLEASE REBOOT  
SYSTEM!!"

(Please power-off system manually)

Step6. Power off the system and remove the Installation SD-Card

Step7. Power on the system and it should boot to target Android OS

## 4. Functionality

DS2-1GRAM is designed with enhanced features including NFS and Watchdog Timer support. These Functions can be controlled in SmartETK Tool under Android Environment. For more details of SmartETK, please refer to "API-ref.pdf" in the "Doc" Folder of BSP.

# 5. Debug Message

## 5.1. U-Boot Environment

DS2-2GRAM Platform can stop booting to enter U-Boot environment. The u-boot will initiate hardware at an earlier stage by specific parameters.

1. Connect debug port.

Use terminal application on PC site



Comm speed: 115200

Comm parity: None

Comm data: 8

Comm stopbits:1

## 2. Enter U-Boot.

The u-boot will wait 3 seconds to stop booting after power on by pressing any key. When booting is stopped, that prompt sign “S3 #” will show up on terminal screen.

U-Boot is like a tiny operation system that has its own commands. Here it describes some important commands and parameters.

### 5.2. U-Boot Parameters Example

- Print online help

```
S3 # help
```

- Save changed parameters

```
S3 # saveenv
```

# 6. Clone-View Rotation

To enable Clone-view Rotation function, please follow the following steps to rebuild the graphic driver and rebuild the Android Image.

Note:

When enabling this feature, please connect both HDMI outputs to make sure that boot procedure is correct. And please backup your current gfx configuration files which will be replaced in next step.

## 6.1. Enable Clone-View Dual Rotation Function

### 6.1.1 Replace the configuration files.

```
$cd <<S3Driver_DIR>>/  
$cp << S3Driver_DIR >>/clone_view/<<Resolution_Type>>/s3g_mode.cfg ./  
$cp <<S3Driver_DIR>>/clone_view/<<Resolution_Type>>/via_kernel_android.conf ./
```

P.S. <<S3Driver\_DIR>>: Directory of S3 Driver Package  
<<Resolution\_Type>>: Support Resolution Type for Clone-View Rotation.

### 6.1.2 Rebuild the Graphic Driver

```
$ cd <<S3Driver_DIR>>/  
$ export S3GANDROID_DIR=<< S3Driver_DIR >>  
$ export LINUXDIR=<< Kernel _DIR >>  
$ ./build.sh
```

Following Driver Binary will be generated in <<S3Driver\_DIR>>:  
S3\_audio.ko  
S3g\_core.ko  
S3g.ko

```
$ export ANDROID_SYS=<< Android_DIR >>/out/target/product/elite1000  
$ ./install.sh
```

P.S. <<S3Driver\_DIR>>: Directory of S3 Driver Package  
<<Kernel\_DIR>>: Directory of Linux Kernel Source Code  
<<Android\_DIR>>: Directory of Android Source Code

### 6.1.3 Rebuild the Android Image

Please follow Section 2.3.5 to Rebuild Android Image. If you do not build the bsp now, please back to Chapter 2 to starting building bsp.

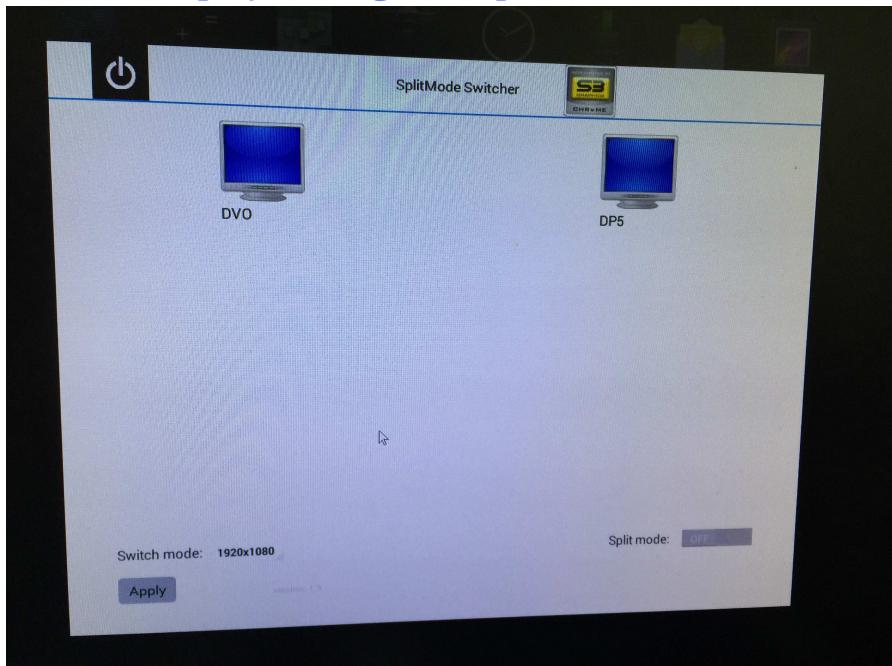
# 7. Change Display Mode

The display modes on DS2-2GRam platform are Normal Clone Mode and Extension Mode. The default mode is Normal Mode. If users want to enable the Extension Mode, please refer to the following steps to enable it.

## 7.1. Display Setting

The display mode can be changed through our sample ap behavior. Please get the sample ap “[SplitModeSwitcher\\_4.3\\_v1.3.apk](#)” from the Test\_Tool Folder which is located in EVK Folder of BSP. After that, please install SplitModeSwither application in your DS2 platfrom.

## 7.2. Display Setting Example



# Appendix A: Definitions

<b>Android</b>	Android is a trademark of Google Inc.
<b>ARM</b>	ARM is a trademark of ARM Inc.
<b>BSP</b>	Board Support Package
<b>HDMI</b>	High Definition Multimedia Interface
<b>SD</b>	Secure Digital Multimedia Card
<b>DS2-2GRAM</b>	The Target Product Name
<b>VIA</b>	VIA Technologies, Inc.

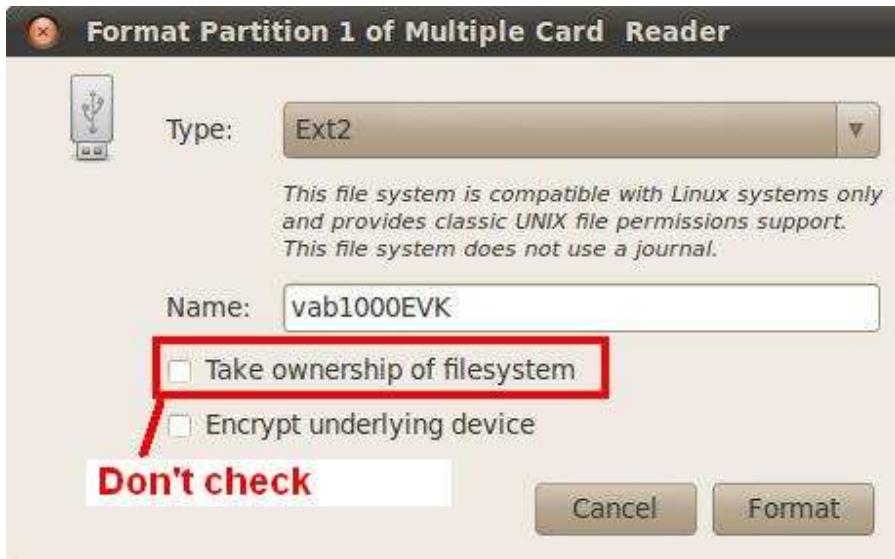
## Appendix B: BSP Source Reference

BSP Component	Source Code Reference
Version	2.2.2
U-Boot	u-boot-2013.07-VEPD-9_8fb9_52.02.02d.c1.tgz
Linux Kernel	elite-android-VEPD-9-d8f8_16.00.03f_520202d.c1.tgz
Busybox Rootfs	N/A
S3 Driver Package	s3g-android-52.02.02d-El.tgz
Android Rootfs	android_4_3-VEPD-9-832b_52.02.02d.c1.tgz

# Appendix C: Notification

## 1. Formatting SD-Card

If you format the SD-Card with <Ubuntu Disk Utility> program. You need to notice that options-Take ownership of filesystem cannot be chosen as Figure shows.



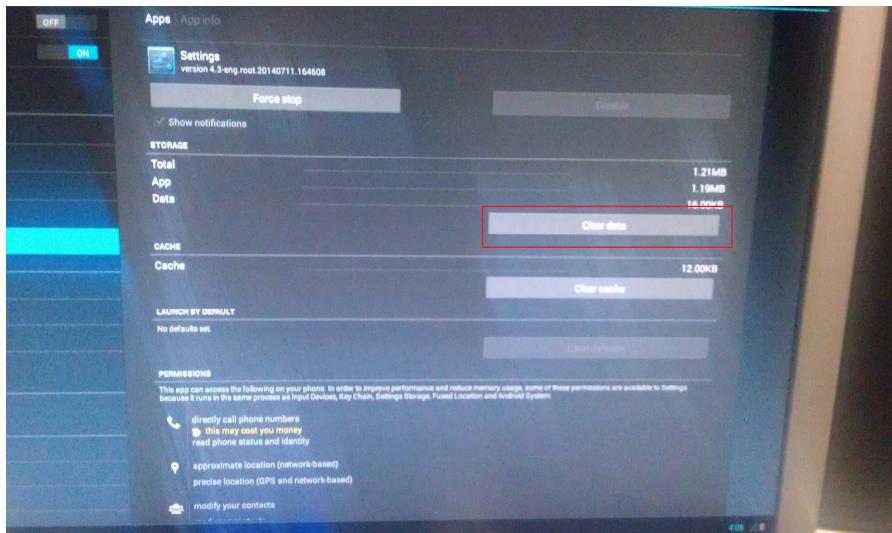
## 2. Development Option

Developer option is hidden by default on Android 4.3. If you want to enable the development option, you need follow the following procedure.

1. Run the setting application
2. Select the About phone option
3. tap build number several times
4. Development Option will show up

Also, if you want to re-hidden the Development Option, you can clear the setting data to re-hidden Development Option.

1. Run the Setting application
2. Select app option
3. Select the setting application
4. Enable the clear data option



### 3. ADB function

We close the ADB TCP function because of the security issue. To enable the ADB function, users need to enable the Development Options first. After that, users can see the “Start TCP ADB” option in Section Debugging. Users can choose if ADB function is enabled or not.

