S5PV210 Android2.3

SMDKV210 Gingerbread

Revision 1.1
Feb 16, 2011

Installation Guide

© 2011 Samsung Electronics Co., Ltd. All rights reserved.

SAMSUNG ELECTRONICS

SAMSUNG

Important Notice

The information in this publication has been carefully checked and is believed to be entirely accurate at the time of publication. Samsung assumes no responsibility, however, for possible errors or omissions, or for any consequences resulting from the use of the information contained herein.

Samsung reserves the right to make changes in its products or product specifications with the intent to improve function or design at any time and without notice and is not required to update this documentation to reflect such changes.

This publication does not convey to a purchaser of semiconductor devices described herein any license under the patent rights of Samsung or others.

Samsung makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Samsung assume any liability arising out of the application or use of any product or circuit and specifically disclaims any and all liability, including without limitation any consequential or incidental damages.

"Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by the customer's technical experts.

Samsung products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, for other applications intended to support or sustain life, or for any other application in which the failure of the Samsung product could create a situation where personal injury or death may occur.

Should the Buyer purchase or use a Samsung product for any such unintended or unauthorized application, the Buyer shall indemnify and hold Samsung and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, expenses, and reasonable attorney fees arising out of, either directly or indirectly, any claim of personal injury or death that may be associated with such unintended or unauthorized use, even if such claim alleges that Samsung was negligent regarding the design or manufacture of said product.

S5PV210 Android2.3, SMDKV210 Gingerbread Installation Guide, Revision 1.1

Copyright © 2011 Samsung Electronics Co., Ltd.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electric or mechanical, by photocopying, recording, or otherwise, without the prior written consent of Samsung Electronics.

Samsung Electronics Co., Ltd.
San #24 Nongseo-Dong, Giheung-Gu
Yongin-City, Gyeonggi-Do, Korea 446-711

TEL: (82)-(31)-209-2254
FAX: (82)-(31)-209-1973

Home Page: http://www.samsungsemi.com

Printed in the Republic of Korea

SAMSUNG ELECTRONICS

SAMSUNG

Revision History

Revision No.	Date	Description	Refer to	Author(s)
0.01	Dec. 23, 2010	- Initial version	all	Minju Lee
0.01	Dec. 24.2010	- Added installing Bootloader interface driver	all	KangNam Park
0.02	Jan. 13. 2011	- Reviewed document for beta release.	all	JeongBae Seo
0.03	Jan 17, 2011	- Reviewed document for beta release	all	Hosung Kim
0.04	Jan 17, 2011	- Added descriptions of EXT4	all	Junseok Jung
1.0	Jan 24, 2011	- Release revision 1.0	all	Taekki Kim
1.1	Feb 16, 2011	- Modified SD/MMC device node name - Added build method as CPU board re-vision.	all	Taekki Kim

Table of Contents

List of Figures

List of Examples

List of Acronyms

Acronyms	Descriptions
ADB	Android Debug Bridge
ALSA	Advanced Linux Sound Architecture
APM	Advanced Power Management
ASL	Apache Software License
BSP	Board Support Package
EGL	Embedded Graphics Library
EXIF	Exchangeable Image File Format
FIFO	First In, First Out
FIMC	Fully Interactive Mobile Camera
FIMD	Fully Interactive Mobile Display
FIMG	Fully Interactive Mobile Graphic
GPL	General Public License
HDMI	High-Definition Multimedia Interface
HS-MMC	High Speed Multi Media Card
IP	Intellectual Property
MFC	Multi Format Codec
MIPI	Mobile Industry Processor Interface
MMC	Multi Media Card
MSB	Most-Significant Bit
MTD	Memory Technology Device
OpenGL	OpenGL(Open Graphic Library)
OpenGL ES	OpenGL(Open Graphic Library) for Embedded Systems
OS	Operating System
PC	Personal Computer
POP	Package on Package
SD	Secure Digital
SMDK	Samsung Mobile Development Kit
SPDIF	Sony Philips Digital Interconnect Format
TFTP	Trivial File Transfer Protocol
UART	Universal Asynchronous Receiver and Transmitter
U-boot	Universal Boot Loader
UMS	USB Mass-Storage
USB OTG	USB(Universal Serial Bus) On-The-Go

1 INTRODUCTION

This document is intent to provide detailed instruction on building the Android Gingerbread BSP on the SMDKV210 board.

Since the Android Gingerbread BSP supports Nand and Movi-Nand(SD/MMC) devices as booting devices, this document includes descriptions of how to get Nand and Movi-Nand(SD/MMC) for a booting and mass storage. If you are using Nand device as a booting device, it provides method to integrate MTD/Yaffs2 solution into the boot loader, Android kernel, and Android Gingerbread platform, and to write the binary images on Nand device. If you are using Movi-Nand(SD/MMC) as a booting device, it provides useful procedures to support ext4 file system and to write the binary images on Movi-Nand(SD/MMC) device.

Chapter 2 describes how to set up the development environment on the host PC. Chapter 3 explains the building guidance of the boot loader, Android kernel and Android Gingerbread platform. Chapter 4 provides flashing methods using boot loader commands and chapter 5 shows settings to select booting device (i.e Nand or Movi-Nand(SD/MMC))

NOTE: It has been created assuming that you are running an Ubuntu linux and should be familiar with knowledge of embedded linux.

2 SYSTEM REQUIREMENTS

The Android build is routinely tested on recent versions of Ubuntu (10.10 or later).

2.1 UBUNTU LINUX (64-BIT X86)

To set up your Linux development environment, make sure you have the following:

Required Packages:

1. Git 1.5.4 or newer and the GNU Privacy Guard.

2. JDK 6.0

3. flex, bison, gperf, libsdl-dev, libesd0-dev, libwxgtk2.6-dev (optional), build-essential, zip, curl.

 # sudo apt-get install git-core gnupg sun-java5-jdk flex bison gperf libsdl-dev libesd0-dev libwxgtk2.6-dev

build-essential zip curl libncurses5-dev zlib1g-dev

You might also want Valgrind, a tool that will help you find memory leaks, stack corruption, array bounds over-flows, etc.

 # sudo apt-get install valgrind

For more information regarding installation of them, refer to following web site.

- http://source.android.com/source/download.html

2.2 INSTALL TOOLCHAIN

Building the tool chain is not a trivial exercise and for most common situations pre-built tool chains already exists.

Unless you need to build your own, or you want to do it anyway to gain a deeper understanding, then simply

installing and using a suitable ready-made tool chain is strongly recommended.

- It will be provided toolchain file arm-2009q3.tar.gz. To build the Android BSP, you need to install the tool chain as follows.

Procedure:

- ***1. Get root permission and go to root directory***
  - ***#cd /***
- ***2. Extract toolchain file(arm-2009q3.tar.gz) at root directory.***
  - ***#gzip -d arm-2009q3.tar.gz; tar -xvf arm-2009q3.tar***
- ***3. Make sure that toolchain is installed at /opt/toolchains directory.***

2.3 PREPARATIONS FOR FASTBOOT

2.3.1 GETTING ANDROID SDK & INSTALLING ANDROID SDK

To download Android SDK, refer to following web site.

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

Required Packages

- Eclipse 3.4(Ganymede) or newer
- Eclipse JDT plugin(included in most Eclipse IDE package)
- JDK 6(JRE alone is not sufficient)
- Android Development Tools plugin

To install Android SDK, refer to following web site.

http://developer.android.com/sdk/installing.html

2.3.2 INSTALLING BOOTLOADER INSTERFACE DRIVER

Procedure:

- ***1. Execute the SDK Manager.***
- ***2. Set the proxy server (optional).***

- ***3. Check the Android SDK Platform-tools package & Uncheck “Display updates only”.***

- ***4. Check the Google USB Driver package & Click “Install Selected”.***

- ***5. Make sure the Package Description & License & Accept All & Click "Install".***

-
- ***5. Click "Close" after packages installed.***

- ***6. Add the following configurations at android_winusb.inf.***

```
…

[Google.NTx86]
; HTC Dream
```

```
%SingleAdbInterface%        = USB_Install, USB\VID_0BB4&PID_0C01
%CompositeAdbInterface%     = USB_Install, USB\VID_0BB4&PID_0C02&MI_01
%SingleBootLoaderInterface% = USB_Install, USB\VID_0BB4&PID_0FFF
; HTC Magic
%CompositeAdbInterface%     = USB_Install, USB\VID_0BB4&PID_0C03&MI_01
;
;Moto Sholes
%SingleAdbInterface%        = USB_Install, USB\VID_22B8&PID_41DB
%CompositeAdbInterface%     = USB_Install, USB\VID_22B8&PID_41DB&MI_01
;
;Google NexusOne
%SingleAdbInterface%        = USB_Install, USB\VID_18D1&PID_0D02
%CompositeAdbInterface%     = USB_Install, USB\VID_18D1&PID_0D02&MI_01
%SingleAdbInterface%        = USB_Install, USB\VID_18D1&PID_4E11
%CompositeAdbInterface%     = USB_Install, USB\VID_18D1&PID_4E12&MI_01
%CompositeAdbInterface%     = USB_Install, USB\VID_18D1&PID_4E22&MI_01
%SingleBootLoaderInterface% = USB_Install, USB\VID_18D1&PID_0002

[Google.NTamd64]
; HTC Dream
%SingleAdbInterface%        = USB_Install, USB\VID_0BB4&PID_0C01
%CompositeAdbInterface%     = USB_Install, USB\VID_0BB4&PID_0C02&MI_01
%SingleBootLoaderInterface% = USB_Install, USB\VID_0BB4&PID_0FFF
; HTC Magic
%CompositeAdbInterface%     = USB_Install, USB\VID_0BB4&PID_0C03&MI_01
;
;Moto Sholes
%SingleAdbInterface%        = USB_Install, USB\VID_22B8&PID_41DB
%CompositeAdbInterface%     = USB_Install, USB\VID_22B8&PID_41DB&MI_01
;
;Google NexusOne
%SingleAdbInterface%        = USB_Install, USB\VID_18D1&PID_0D02
%CompositeAdbInterface%     = USB_Install, USB\VID_18D1&PID_0D02&MI_01
%SingleAdbInterface%        = USB_Install, USB\VID_18D1&PID_4E11
%CompositeAdbInterface%     = USB_Install, USB\VID_18D1&PID_4E12&MI_01
%CompositeAdbInterface%     = USB_Install, USB\VID_18D1&PID_4E22&MI_01
%SingleBootLoaderInterface% = USB_Install, USB\VID_18D1&PID_0002
```

…

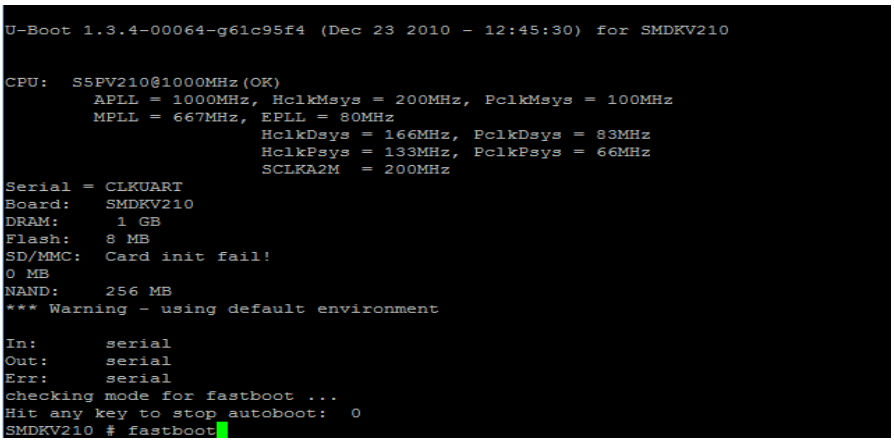
Figure 2-1 Android_winusb.inf

This file is used to install ADB USB driver at Window PC. If you already get the modified inf file, you do not need this step. Also, if you are developing at Linux machine, you do not need inf file.

- ***7. Power on.***

- ***8. Hit any key when displaying "Hit any key to stop autoboot: #".***

- ***9. [SMDKV210 board side] Execute u-boot command "fastboot".***



10. Install Android Bootloader Interface Drive

3 BUILDING ANDROID PACKAGE

3.1 GETTING ANDROID 2.3 PACKAGE

- Uboot : android_uboot_smdkv210.tar.bz2
- Kernel : android_kernel_2.6.35_smdkv210.tar.bz2
- Gingerbread : android_gingerbread_smdkv210.tar.bz2

3.2 CREATING NAND IMAGE

3.2.1 HARDWARE CONFIGURATION

The first step for creating Nand images is to configure SMDKV210 board for Nand.

- CPU BOARD
  - Configure Nand: CFG4[6:1] = OFF OFF OFF OFF ON X
    - CFG3[6:1] = OFF OFF ON ON OFF OFF
- BASE BOARD
  - Configure UART: CFGB13[4:1] = OFF OFF OFF ON
    - Turn on just CFGB13[1]
  - Configure CFGB11[6:1] = OFF OFF OFF OFF OFF ON

3.2.2 BUILDING U-BOOT

U-Boot is a primary boot loader for Android BSP on SMDKV210. This can load a kernel image and a ramdisk image from Nand device.

Procedure:

- ***1. Go top directory of U-boot.***
- ***2. Check Makefile.***
  - ***Set the path of cross compiler. This version of u-boot has to be compiled using tool chain 2009q3.***

Example 3-1 Makefile

```
143     ifeq ($(ARCH),arm)
144     CROSS_COMPILE = /opt/toolchains/arm-2009q3/bin/arm-none-linux-gnueabi-
145     endif
```

- ***3. Modify include/configs/smdkv210single.h***
  - ***Set the CFG_FASTBOOT_NANDBSP option(by removing comment sign "//")***
    - ***Disable CFG_FASTBOOT_ONENANDBSP option(by adding comment sign "//")***
    - ***Disable CFG_FASTBOOT_SDMMCBSP option(by adding comment sign "//")***

Example 3-2 smdkv210single.h

```
595 /* Just one BSP type should be defined. */
596 //#define CFG_FASTBOOT_ONENANDBSP
597 #define CFG_FASTBOOT_NANDBSP
598 //#define CFG_FASTBOOT_SDMMCBSP
```

  - ***If CPU board revision number is 0.2 and PMIC type is max8998, set the CONFIG_SMDKV210_REV02 option(by removing comment sign "//"). If PMIC type is not max8998, disable the CONFIG_SMDKV210_REV02 option(by comment sign "//").***

Example 3-3 smdkv210single.h

```
42 #define CONFIG_MCP_SINGLE          1
43 #define CONFIG_EVT1                1       /* EVT1 */
44 #define CONFIG_SMDKV210_REV02      1       /* Rev 0.2 and PMIC Type is MAX8998 */
```

- ***4. # make smdkv210single_config***
- ***5. # make***
- ***6. Verify that u-boot.bin is created.***

3.2.3 BUILDING KENEL 2.6.35

Kernel is configured to use MTD/Yaffs2.

Procedure:

1. Go to the top directory of Android kernel.

2. modify ~/.bashrc

- Add 2 code lines

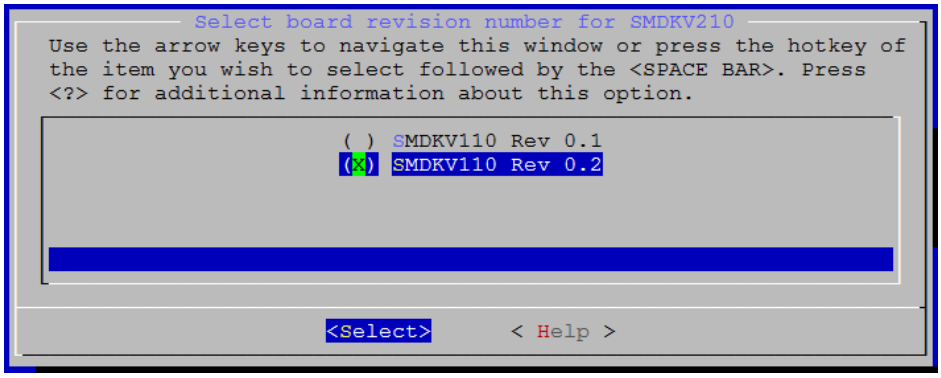
export ARCH=arm

export CROSSCOMPILE=/opt/toolchains/arm-2009q3/bin/arm-none-linux-gnueabi-

source ~/.bashrc

3. # make smdkv210_android_defconfig

- ***If CPU board revision number is 0.2 and PMIC type is max8998, change kernel option. If PMIC type is not max8998, kernel option is not change.***
  - ***# make menuconfig***
  - ***Go to "System Type → Select board revision number for SMDKV210".***
  - ***Set the "SMDKV210 Rev 0.2".***



  - ***Go to "Device Drivers → Multifunction device drivers".***
  - ***Disable the "Maxim Semiconductor MAX8698 PMIC Support".***
  - ***Set the "Maxim Semiconductor MAX8998 PMIC Support".***
  - ***Go to "Device Drivers → Voltage and Current Regulator Support".***
  - ***Disable the "Maxim 8698 voltage regulator".***
  - ***Set the "Maxim 8998 voltage regulator ".***

4. # make

5. Verify that arch/arm/boot/zImage is created.

3.2.4 BUILDING GINGERBREAD

Android Gingerbread platform requires four file systems so that it can operate properly: ramdisk, cache, system and data. Ramdisk is a small sized image for root file system. The others are for directories, /cache, /system, and /data respectively.

In order to get images for Android Gingerbread platform, you firstly build Android Gingerbread platform.

Procedure:

1. Go to the top directory of Android kernel.

2. modify ~/.bash_profile

- Add JDK file and path

```
JDK=/java/jdk1.6.0_21/bin

PATH=$GIT_PATH:$JDK:$UBOOT_COMPILER:$PATH
```

3. source ~/.bash_profile

4. Execute the following command to setup the product name.

- ***#export SEC_PRODUCT=smdkv210***

5. Execute the following command to setup the kernel directory.

- ***#export KERNEL_DIR=…***
  - ***(Ex. #export KERNEL_DIR=/public/android_kernel_2.6.35)***
- ***If you don't setup kernel directory, "boot.img" file is not created.***

6. Execute the following command

- ***#./build_android.sh***

7. Verify ramdisk-uboot.img, boot.img, system.img and userdata.img have been created at out/target/product/smdkv210.

If you are working on 32bit machine, you need to set up as follows :

1. Modify /build/core/main.mk

-ifneq (64, $(findstring 64, $(build_arch)))

+ifneq(i686, $(findstring i686, $(build_arch)))

2. Modify 4 files

./external/clearsilver/cgi/Android.mk

./external/clearsilver/java-jni/Android.mk

./external/clearsilver/util/Android.mk

./external/clearsilver/cs/Android.mk

Disable 2 options(by adding comment sign "#")

 # LOCAL_CFLAGS += -m64

 # LOCAL_LDFLAGS += -m64

3.3 CREATING MOVI-NAND(SD/MMC) IMAGE

3.3.1 HARDWARE CONFIGURATION

The first step for creating Movi-Nand(SD/MMC) images is to configure SMDKV210 board for SD/MMC.

- CPU BOARD
  - Configure SD/MMC: CFG4[6:1] = OFF OFF ON ON OFF X
- BASE BOARD
  - Configure UART: CFGB13[4:1] = OFF OFF OFF ON
    - Turn on just CFGB13[1]
  - Configure Audio: Remove JP2 under the CPU board
-

3.3.2 BUILDING U-BOOT

U-Boot is a primary boot loader for Android BSP on SMDKV210. This can load a kernel image and a ramdisk image from Movi-Nand(SD/MMC) device.

Procedure:

1. Go to top directory of U-boot.

2. Check Makefile.

- ***Set the path of cross compiler. This version of u-boot has to be compiled using tool chain 2009q3. If you already added the path of cross compiler line to bash profile or bashrc file, you do not need this step.***

Example 3-4 Makefile

```
143     ifeq ($(ARCH),arm)
144     CROSS_COMPILE = /opt/toolchains/arm-2009q3/bin/arm-none-linux-gnueabi-
145     endif
```

3. Modify include/configs/smdkv210single.h

- ***Set the CFG_FASTBOOT_SDMMCNANDBSP option(by removing comment sign "//")***
  - ***Disable CFG_FASTBOOT_ONENANDBSP option(by adding comment sign "//")***
  - ***Disable CFG_FASTBOOT_NANDBSP option(by adding comment sign "//")***

Example 3-5 smdkv210single.h

```
595 /* Just one BSP type should be defined. */
596 //#define CFG_FASTBOOT_ONENANDBSP
```

```
597 //#define CFG_FASTBOOT_NANDBSP
598 #define CFG_FASTBOOT_SDMMCBSP
```

- ***If CPU board revision number is 0.2 and PMIC type is max8998, set the CONFIG_SMDKV210_REV02 option(by removing comment sign "//"). If PMIC type is not max8998, disable the CONFIG_SMDKV210_REV02 option(by comment sign "//").***

Example 3-6 smdkv210single.h

```
42 #define CONFIG_MCP_SINGLE          1
43 #define CONFIG_EVT1                1       /* EVT1 */
44 #define CONFIG_SMDKV210_REV02      1       /* Rev 0.2 and PMIC Type is MAX8998 */
```

4. # make smdkv210single_config

5. # make

6. Verify that u-boot.bin is created.

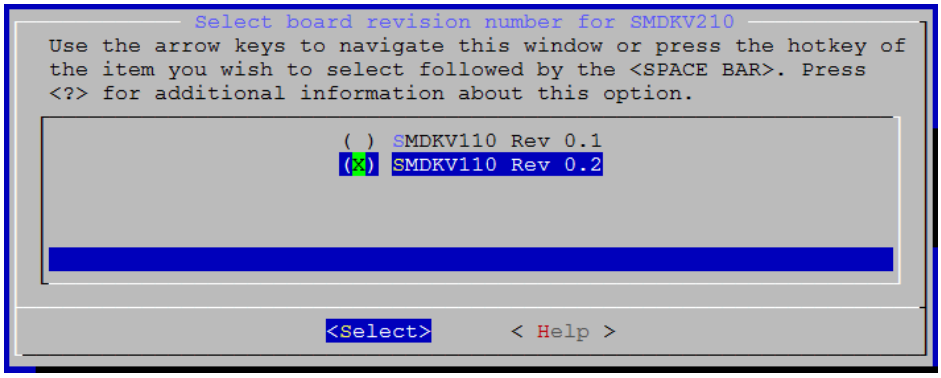
3.3.3 BUILDING KENEL 2.6.35

Procedure:

1. Go to the top directory of Android kernel

2. # make smdkv210_android_defconfig

- ***Check Ext4 filesystem on menuconfig.***
  - ***Go to File Systems → File systems***
  - ***Set <*> The Extended 4 (ext4) filesystem***
  - ***Set [*] Use ext4 for ext2/ext3 file systems***
  - ***Save & Exit***
- ***If CPU board revision number is 0.2 and PMIC type is max8998, change kernel option. If PMIC type is not max8998, kernel option is not change.***
  - ***# make menuconfig***
  - ***Go to "System Type → Select board revision number for SMDKV210".***
  - ***Set the "SMDKV210 Rev 0.2".***



- ***Go to "Device Drivers → Multifunction device drivers".***
- ***Disable the "Maxim Semiconductor MAX8698 PMIC Support".***
- ***Set the "Maxim Semiconductor MAX8998 PMIC Support".***
- ***Go to "Device Drivers → Voltage and Current Regulator Support".***
- ***Disable the "Maxim 8698 voltage regulator".***
- ***Set the "Maxim 8998 voltage regulator ".***

3. # make

4. Verify that arch/arm/boot/zImage is created.

3.3.4 BUILDING GINGERBREAD

Android Gingerbread platform requires four file systems so that it can operate properly: ramdisk, cache, system and data. Ramdisk is a small sized image for root file system. The others are for directories, /cache, /system, and /data respectively.

In order to get images for Android Gingerbread platform, you firstly build Android Gingerbread platform.

Procedure:

1. Execute the following command to setup the product name.

- ***#export SEC_PRODUCT=smdkv210***

2. Execute the following command to setup the kernel directory.

- ***#export KERNEL_DIR=…***
  - ***(Ex. #export KERNEL_DIR=/public/android_kernel_2.6.35)***
- ***If you don't setup kernel directory, "boot.img" file is not created.***

3. Modify device/samsung/smdkv210/BoardConfig.mk.

Example 3-7 BoardConfig.mk

```
77 BOARD_SDMMC_BSP := true
78
79 ifeq ($(BOARD_SDMMC_BSP),true)
80 TARGET_USERIMAGES_USE_EXT4 := true
81 BOARD_SYSTEMIMAGE_PARTITION_SIZE := 125829120
82 BOARD_USERDATAIMAGE_PARTITION_SIZE := 370147328
83 BOARD_FLASH_BLOCK_SIZE := 4096
84 endif
```

4. Modify system/extras/ext4_utils/mkuserimg.sh.

Example 3-8 mkuserimg.sh

```
44 echo "make_ext4fs -l $SIZE -a $MOUNT_POINT $OUTPUT_FILE $SRC_DIR"
45 make_ext4fs -l $SIZE -a $MOUNT_POINT $OUTPUT_FILE $SRC_DIR
```

```
46 if [ $? -ne 0 ]; then
47   exit 4
48 fi
```

- ***Remove -s option***

5. Execute the following command

- ***#./build_android.sh***

6. Verify ramdisk-uboot.img, system.img and userdata.img have been created at out/target/product/smdkv210.

4 FLASHING SMDKV210

4.1 MAKING BOOTABLE SDCARD

First of all, it is very important that SD card reader is identified in Linux PC. When the SD card reader is connected Linux PC, it is automatically created device file corresponding to the SD card reader in /dev directory (typically, /dev/sdb). Check whether /dev/sdb device file is generated.

When the SD card reader is NOT connected Linux PC

```
# ls -l /dev/sd?
brw-rw---- 1 root disk 8, 0 2010-03-30 11:37 /dev/sda
#
```

Figure 4-1 device files without sdcard device

When the SD card reader is connected Linux PC

```
# ls -l /dev/sd?
brw-rw---- 1 root disk 8,  0 2010-03-30 11:37 /dev/sda
brw-rw---- 1 root disk 8, 16 2010-03-30 12:57 /dev/sdb
#
```

Figure 4-2 device files with sdcard device

To fuse u-boot image, insert SD card and execute “sd_fusing.sh” shell script. Before executing sd_fusing.sh shell script, should obtain root permission. The "sd_fusing.sh" shell script performs sd card partitioning, vfat formatting, and bl1, uboot image fusing.

Procedure:

1. Go to the top directory of U-boot.

2. # cd sd_fusing

3. # make

4. # bash ./sd_fusing.sh /dev/sdb

NOTE: In general, this step is only required to prepare a bootable sdcard. If you already a bootable and well-partitioned sdcard, you can skip this step.

4.2 FLASHING ANDROID PACKAGE TO NAND VIA USB

Once you prepared a bootable sdcard, the next step is to copy Android images into a directory pc-window. In order to use fastboot command, your pc-window is prepared fastboot command and connected to SMDKV210 board.

Procedure:

1. Set OM pin at CPU board to be sdmmc booting.

- ***sdmmc booting: CFG4[6:1] = OFF OFF ON ON OFF X***

2. Insert bootable sdcard into SLOT 0 at SMDK board.

3. Power on.

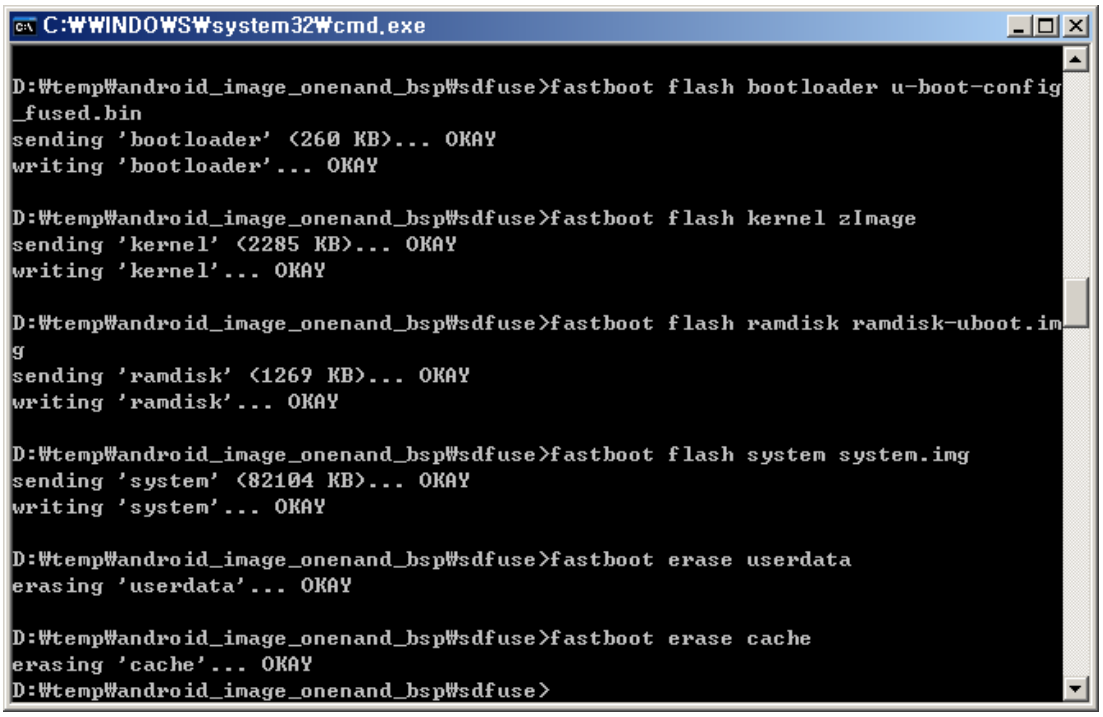
4. Hit any key when displaying "Hit any key to stop autoboot: #"

5. [SMDKV210 board side] Execute u-boot command "fastboot"

6. [Host-PC side] Execute the following commands on pc-window.

- ***"fastboot flash bootloader u-boot.bin"***
- ***"fastboot flash kernel zImage"***
- ***"fastboot flash ramdisk ramdisk-uboot.img"***
- ***"fastboot flash system system.img"***
- ***"fastboot erase userdata"***
- ***"fastboot erase cache"***

NOTE: u-boot.bin will be used when CONFIG_FUSED is not selected in smdkv210single.h



4.3 FLASHING ANDROID PACKAGE TO NAND VIA SDCARD

4.3.1 COPYING IMAGES

Once you prepared a bootable sdcard, the next step is to copy Android images into sdcard. In order to use sdfuse command, you should copy images into sdfuse directory.

Procedure:

1. Mount sdcard

2. Create a directory, sdfuse, at the root directory of sdcard.

3. Copy u-boot-config_fused.bin, boot.img, system.img to the sdfuse, if you want to write u-boot into Nand.

4. Umount sdcard

4.3.2 FLASHING IMAGES

Note: To give similar user experience, we made the usage of sdfuse to be looked like that of fastboot. The differences are 1) sdfuse works only at board-side, 2) sdfuse does not support update command.

An sdfuse command provides four functions: info, flashall, flash, and erase.

1. info: Print information related sdfuse.

2. flashall: Flash boot.img and system.img. Erase two partition userdata and cache. And, reboot.

3. flash: Write a file in sdcard to a specific partition.

4. erase: Erase (format) a specific partition.

NOTE: This document assumes that you will write images of u-boot.bin, boot.img and system.img. If you want other images, such as zImage, ramdisk-uboot.img, userdata.img, then use "sdfuse flash" command. The usage of this command is also similar to that of fastboot. The names of valid partitions are displayed when "sdfuse info" is executed at u-boot.

4.3.2.1 STEP 1: Writing Boot Loader

4.3.2.1.1 Starting U-boot

Procedure:

1. Set OM pin at CPU board to be sdmmc booting.

- ***sdmmc booting: CFG4[6:1] = OFF OFF ON ON OFF X***

2. Insert bootable sdcard into SLOT 0 at SMDK board.

3. Power on.

4. Hit any key when displaying "Hit any key to stop autoboot: #"

4.3.2.1.2 Executing sdfuse

Procedure:

1. #sdfuse flash bootloader u-boot.bin

2. Power off

3. Set OM pin at CPU board to be Nand booting

- ***Nand booting: CFG4[6:1] = OFF OFF OFF OFF ON X***

4. Power on

5. Verify that u-boot is started.

```
[Fusing Image from SD Card.]
Fastboot: employ default partition information
[Partition table on NAND]
ptn 0 name='bootloader' start=0x0 len=0x100000(~1024KB)
ptn 1 name='recovery' start=0x100000 len=0x500000(~5120KB)
ptn 2 name='kernel' start=0x600000 len=0x500000(~5120KB)
ptn 3 name='ramdisk' start=0xB00000 len=0x300000(~3072KB)
ptn 4 name='system' start=0xE00000 len=0x6E00000(~112640KB) (Yaffs)
ptn 5 name='cache' start=0x7C00000 len=0x5000000(~81920KB) (Yaffs)
ptn 6 name='userdata' start=0xCC00000 len=N/A (Yaffs)
Partition: bootloader, File: /sdfuse/u-boot.bin
Partition1: Start Address(0x5af0), Size(0x3bdaf8)
reading /sdfuse/u-boot.bin
262144 (0x00040000) bytes read
flashing 'bootloader'

NAND erase: device 0 offset 0x0, size 0x100000
Erasing at 0x100000 -- 100% complete.
OK

NAND write: device 0 offset 0x0, size 0x40000

Writing data at 0x100000 -- 100% complete.
 262144 bytes written: OK
partition 'bootloader' flashed
OK
SMDKV210 #
```

Figure 4-4 Result - sdfuse flash bootloader u-boot.bin

4.3.2.2 STEP 2: Writing All Images

In order to fuse Android images, you had better use "sdfuse flashall" command. This command writes boot.img (zImage + ramdisk-uboot.img) and system.img in a single command. Also it erases two MTD partitions, userdata and cache. Finally, it reboot.

Procedure:

1. Insert bootable sdcard into SLOT 0 at SMDK board.

2. Power on.

3. Hit any key when displaying "Hit any key to stop autoboot: #"

4. # sdfuse flashall

```
SMDKV210 # sdfuse flashall
[Fusing Image from SD Card.]
Fastboot: employ default partition information
[Partition table on NAND]
ptn 0 name='bootloader' start=0x0 len=0x100000(~1024KB)
ptn 1 name='recovery' start=0x100000 len=0x500000(~5120KB)
ptn 2 name='kernel' start=0x600000 len=0x500000(~5120KB)
ptn 3 name='ramdisk' start=0xB00000 len=0x300000(~3072KB)
ptn 4 name='system' start=0xE00000 len=0x6E00000(~112640KB) (Yaffs)
ptn 5 name='cache' start=0x7C00000 len=0x5000000(~81920KB) (Yaffs)
ptn 6 name='userdata' start=0xCC00000 len=N/A (Yaffs)
Partition: boot, File: /sdfuse/boot.img
Partition1: Start Address(0x5af0), Size(0x3bdaf8)
reading /sdfuse/boot.img
2856960 (0x002b9800) bytes read
Kernel size: 001d7034
Ramdisk size: 000e1798
flashing 'kernel'

NAND erase: offset 0x600000, size 0x500000
OK

NAND write: offset 0x600000, size 0x200000
```

```
Main area write (8 blocks):
 2097152 bytes written: OK
flashing 'ramdisk'

NAND erase: offset 0xb00000, size 0x300000
OK

NAND write: offset 0xb00000, size 0x100000
Main area write (4 blocks):
 1048576 bytes written: OK
partition 'kernel+ramdisk' flashed
Partition: system, File: /sdfuse/system.img
Partition1: Start Address(0x5af0), Size(0x3bdaf8)
reading /sdfuse/system.img
102359040 (0x0619e000) bytes read
Partition: userdata
erasing 'userdata'

NAND erase: offset 0xb800000, size 0x14800000
Skip erase bad block 802 at 0xc880000
Skip erase bad block 1665 at 0x1a040000
Skip erase bad block 1866 at 0x1d280000
Skip erase bad block 2025 at 0x1fa40000
OK
partition 'userdata' erased
Partition: cache
erasing 'cache'

NAND erase: offset 0x6800000, size 0x5000000
Skip erase bad block 459 at 0x72c0000
OK
partition 'cache' erased
reset...
```

Figure 4-5 Result - sdfuse flashall

4.4 FLASHING ANDROID PACKAGE TO MOVI-NAND(SD/MMC) VIA USB

Once you prepared a bootable sdcard, the next step is to copy Android images into a directory pc-window. In order to use fastboot command, your pc-window is prepared fastboot command and connected to SMDKV210 board.

Procedure:

1. Set OM pin at CPU board to be sdmmc booting.

sdmmc booting: CFG4[6:1] = OFF OFF ON ON OFF X

2. Insert bootable sdcard into SLOT 1 at SMDK board.

3. Power on.

4. Hit any key when displaying "Hit any key to stop autoboot: #"

5. [SMDKV210 board side] Execute u-boot command "fdisk -c 0"

6. [SMDKV210 board side] eject sdcard from SMDKV210

7. [Host-PC side] Insert sdcard into Linux Host PC

8. [Host-PC side] format sdcard

- ***# mkfs.vfat /dev/sdb0***
- ***# mkfs.ext4 -j /dev/sdb1***
- ***# mkfs.ext4 -j /dev/sdb2***
- ***# mkfs.ext4 -j /dev/sdb3***

9. eject sdcard from Host PC

10. Insert bootable sdcard into SLOT 1 at SMDK board.

11. Power on.

12. Hit any key when displaying "Hit any key to stop autoboot: #"

13. [SMDKV210 board side] Execute u-boot command "fastboot"

14. [Host-PC side] Execute the following commands on pc-window.

- ***"fastboot flash bootloader u-boot.bin"***
- ***"fastboot flash kernel zImage"***
- ***"fastboot flash ramdisk ramdisk-uboot.img"***
- ***"fastboot flash system system.img"***

5 BOOTING SMDKV210

5.1 NAND BOOTING

Procedure:

1. Power off

2. Set OM pin at CPU board to be Nand booting

Nand booting: CFG4[6:1] = OFF OFF OFF OFF ON X

3. Power on

4. Make sure that Gingerbread initial home screen is shown on LCD.

5.2 MOVI-NAND(SD/MMC) BOOTING

Procedure:

1. Power off

2. Set OM pin at CPU board to be SD/MMC booting

sdmmc booting: CFG4[6:1] = OFF OFF ON ON OFF X

3. Power on

4. Make sure that Gingerbread initial home screen is shown on LCD.

6 APPENDIX

6.1 FLASHING ANDROID PACKAGE USING T32

6.1.1 FLASHING NAND

You can use Trace32, a JTAG debugger to load u-boot image in SMDKV210. You can be available the attached cmm file

V210_SMDK_Base_Ver1.cmm

Procedure:

1. Load u-boot.bin at 0x20008000 using Trace32

Example 6-1 Sample Booting Message from Trace32

```
U-Boot 1.3.4-dirty (Jan 20 2011 - 19:08:40) for SMDKV210



CPU:  S5PV210@800MHz(OK)
        APLL = 800MHz, HclkMsys = 200MHz, PclkMsys = 100MHz
        MPLL = 667MHz, EPLL = 80MHz
                       HclkDsys = 166MHz, PclkDsys = 83MHz
                       HclkPsys = 133MHz, PclkPsys = 66MHz
                       SCLKA2M  = 200MHz
Serial = CLKUART
Board:   SMDKV210
DRAM:     1 GB
Flash:   8 MB
SD/MMC:  Card init fail!
0 MB
```

```
NAND:    256 MB
*** Warning - using default environment


In:      serial
Out:     serial
Err:     serial
checking mode for fastboot ...
Hit any key to stop autoboot:  0
SMDKV210 # printenv
bootcmd=nand read C0008000 600000 400000; nand read 30A00000 B00000 180000; bootm C0008000 30A00000
mtdpart=80000 400000 3000000
bootdelay=3
baudrate=115200
ethaddr=00:40:5c:26:0a:5b
ipaddr=192.168.0.20
serverip=192.168.0.10
gatewayip=192.168.0.1
netmask=255.255.255.0


Environment size: 269/16380 bytes
SMDKV210 #
```

2. [SMDKV210 board side] Execute u-boot command "fastboot"

3. [Host-PC side] Execute the following commands on pc-window.

- ***fastboot flash bootloader u-boot.bin***
- ***fastboot flash kernel zImage***
- ***fastboot flash ramdisk ramdisk-uboot.img***
- ***fastboot flash system system.img***
- ***fastboot erase userdata***
- ***fastboot erase cache***

6.1.2 FLASHING MOVI-NAND(SD/MMC)

You can use Trace32, a JTAG debugger to load u-boot image in SMDKV210. You can be available the attached cmm file

1. Insert sdcard into SLOT 1 at SMDK board

2. Power on SMDK board.

3. Load u-boot.bin at 0x20008000 using Trace32

4. Hit any key when displaying "Hit any key to stop autoboot: #"

5. [SMDKV210 board side] Execute u-boot command "fdisk -c 0"

6. [SMDKV210 board side] Power Off SMDKV210 board

7. [SMDKV210 board side] eject sdcard from SMDKV210

8. [Host-PC side] Insert sdcard into Linux Host PC

9. [Host-PC side] format sdcard

- ***# mkfs.vfat /dev/sdb0***
- ***# mkfs.ext4 -j /dev/sdb1***
- ***# mkfs.ext4 -j /dev/sdb2***
- ***# mkfs.ext4 -j /dev/sdb3***

•

10. eject sdcard from Host PC

11. Insert sdcard into SLOT 1 at SMDK board.

12. Power on SMDK board.

13. Load u-boot.bin at 0x20008000 using Trace32

14. Hit any key when displaying "Hit any key to stop autoboot: #"

15.[SMDKV210 board side] Execute u-boot command "fastboot"

16. [Host-PC side] Execute the following commands on pc-window.

- ***fastboot flash bootloader u-boot.bin***
- ***fastboot flash kernel zImage***
- ***fastboot flash ramdisk ramdisk-uboot.img***
- ***fastboot flash system system.img***