

# TC4 ICS Installation Guide

---

## System Install

Rev. 1.0.2

Jun. 2012

SAMSUNG Confidential

si - plaza / Jeff\_cui at 15:49,2012.08.24

## TC4 ICS Installation Guide

SAMSUNG ELECTRONICS RESERVES THE RIGHT TO CHANGE PRODUCTS, INFORMATION AND SPECIFICATIONS WITHOUT NOTICE.

Products and specifications discussed herein are for reference purposes only. All information discussed herein is provided on an "AS IS" basis, without warranties of any kind.

This document and all information discussed herein remain the sole and exclusive property of Samsung Electronics. No license of any patent, copyright, mask work, trademark or any other intellectual property right is granted by one party to the other party under this document, by implication, estoppel or otherwise.

Samsung products are not intended for use in life support, critical care, medical, safety equipment, or similar applications where product failure could result in loss of life or personal or physical harm, or any military or defense application, or any governmental procurement to which special terms or provisions may apply.

For updates or additional information about Samsung products, contact your nearest Samsung office.

All brand names, trademarks and registered trademarks belong to their respective owners.

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

# Important Notice

Samsung Electronics Co. Ltd. ("Samsung") reserves the right to make changes to the information in this publication at any time without prior notice. All information provided is for reference purpose only. Samsung assumes no responsibility for possible errors or omissions, or for any consequences resulting from the use of the information contained herein.

This publication on its own does not convey any license, either express or implied, relating to any Samsung and/or third-party products, under the intellectual property rights of Samsung and/or any third parties.

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.

Customers are responsible for their own products and applications. "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 in applications intended to support or sustain life, or for any other application in which the failure of the Samsung product could reasonably be expected to create a situation where personal injury or death may occur. Customers acknowledge and agree that they are solely responsible to meet all other legal and regulatory requirements regarding their applications using Samsung products notwithstanding

any information provided in this publication. Customer 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 (including but not limited to personal injury or death) that may be associated with such unintended, unauthorized and/or illegal use.

**WARNING** 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. This publication is intended for use by designated recipients only. This publication contains confidential information (including trade secrets) of Samsung protected by Competition Law, Trade Secrets Protection Act and other related laws, and therefore may not be, in part or in whole, directly or indirectly publicized, distributed, photocopied or used (including in a posting on the Internet where unspecified access is possible) by any unauthorized third party. Samsung reserves its right to take any and all measures both in equity and law available to it and claim full damages against any party that misappropriates Samsung's trade secrets and/or confidential information.

**警 告** 本文件仅向经韩国三星电子株式会社授权的人员提供，其内容含有商业秘密保护相关法规规定并受其保护的三星电子株式会社商业秘密，任何直接或间接非法向第三人披露、传播、复制或允许第三人使用该文件全部或部分内容的行为（包括在互联网等公开媒介刊登该商业秘密而可能导致不确定第三人获取相关信息的行为）皆为法律严格禁止。此等违法行为一经发现，三星电子株式会社有权根据相关法规对其进行法律措施，包括但不限于提出损害赔偿请求。

**Copyright © 2011 Samsung Electronics Co., Ltd.**

Samsung Semiconductor (CHINA) R&D Co., Ltd. System solution Center

Address:

9-13F, Weiling Building, jiangnan Road 380#

Binjiang District, Hangzhou China (310052)  
Tel: (86571)8672-6288

Fax: (86571)8672-6280

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

**SAMSUNG ELECTRONICS**



## Trademarks

All brand names, trademarks and registered trademarks belong to their respective owners.

- Exynos, Exynos4212, FlexOneNAND, and OneNAND are trademarks of Samsung Electronics.
- ARM, Jazelle, TrustZone, and Thumb are registered trademarks of ARM Limited. Cortex, ETM, ETB, Coresight, ISA, and Neon are trademarks of ARM Limited.
- Java is a trademark of Sun Microsystems, Inc.
- SD is a registered trademark of Toshiba Corporation.
- MMC and eMMC are trademarks of MultiMediaCard Association.
- JTAG is a registered trademark of JTAG Technologies, Inc.
- Synopsys is a registered trademark of Synopsys, Inc.
- I2S is a trademark of Phillips Electronics.
- I2C is a trademark of Phillips Semiconductor Corp.
- MIPI and Slimbus are registered trademarks of the Mobile Industry Processor Interface (MIPI) Alliance.

All other trademarks used in this publication are the property of their respective owners.

## Chip Handling Guide

### Precaution against Electrostatic Discharge

When handling semiconductor devices, be sure that the environment is protected against static electricity.

1. Operators should wear anti-static clothing and use earth band.
2. All objects that come in direct contact with devices should be made of materials that do not produce static electricity that would cause damage.
3. Equipment and work table must be earthed.
4. Ionizer is recommended to remove electron charge.

### Contamination

Be sure to use semiconductor products in the environment that may not be exposed to dust or dirt adhesion.

### Temperature/Humidity

Semiconductor devices are sensitive to environment temperature and humidity. High temperature or humidity may deteriorate semiconductor device's characteristics. Therefore avoid storage or use in such conditions.

### Mechanical Shock

Care should be exercised not to apply excessive mechanical shock or force on semiconductor device.

## **Chemical**

Do not expose semiconductor device to chemical because reaction to chemical may cause deterioration of device characteristics.

## **Light Protection**

In case of non-EMC (Epoxy Molding Compound) package, do not expose semiconductor IC to strong light. It may cause device's malfunction. (But, some special products which utilize the light or have security function are excepted from this guide)

## **Radioactive, Cosmic and X-ray**

Semiconductor devices can be influenced by radioactive, cosmic ray or X-ray. Radioactive, cosmic and X-ray may cause soft error during device operation. Therefore semiconductor devices must be shielded under environment that may be exposed to radioactive, cosmic ray or X-ray.

## **EMS (Electromagnetic Susceptibility)**

Note that semiconductor device's characteristics may be affected by strong electromagnetic wave or magnetic field during operation under insufficient PCB circuit design for EMS.

**SAMSUNG Confidential**  
si - plaza / Jeff\_cui at 15:49,2012.08.24

## Revision History

| Revision No. | Date           | Description   | Author(s)    |
|--------------|----------------|---|--------------|
| 1.00         | MAR. 27, 2012  | <ul style="list-style-type: none"><li>Draft</li></ul>   | Ma Jun       |
| 1.01         | April 23, 2012 | <ul style="list-style-type: none"><li>Added appendix for secure booting and so on</li></ul>       | Zhang xiuwen |
| 1.02         | Jun 20, 2012   | <ul style="list-style-type: none"><li>Modified again for TC4 RTM version(Final version)</li></ul> | Zhang xiuwen |

SAMSUNG Confidential  
si - plaza / Jeff\_cui at 15:49,2012.08.24

## Table of Contents

|   |    |
|---|----|
| 3.1.1 Main features of U-BOOT .....         | 13 |
| 3.1.2 Compile Procedure: .....              | 13 |
| 4.1.1 JTAG.....                             | 16 |
| 4.1.2 Fastboot Tool (for windows).....      | 16 |
| 4.1.3 SD fuse tool.....                     | 16 |
| 4.2.1 bootloader .....                      | 16 |
| 4.3.1 Flash bootloader into SD/MMC .....    | 17 |
| 4.3.2 Flash bootloader into eMMC .....      | 19 |
| 4.4.1 Flash LOGO into SD Card .....         | 20 |
| 4.4.2 Flash LOGO into emmc Card.....        | 20 |
| 4.5.1 Flash OS into SD/MMC.....             | 21 |
| 4.5.2 Flash OS into eMMC .....              | 21 |
| 4.6.1 Updating automatically .....          | 22 |
| 4.6.2 Updating Manually.....                | 22 |
| 4.6.3 Updating Notice .....                 | 23 |
| 4.7.1 Compile the Recovery function .....   | 23 |
| 4.7.2 How to use the recovery function..... | 24 |
| 6.1.1 Trustzone & AP.....                   | 28 |
| 6.1.2 Trustzone & Software .....            | 28 |
| 6.3.1 Build Trustzone manner.....           | 28 |
| 6.3.2 Build Non-Trustzone manner .....      | 29 |

SAMSUNG Confidential  
si - plaza / Jeff\_cui at 15:49, 2012.08.24

## List of Figures

| Figure<br>Number | Title                        | Page<br>Number |
|------------------|------------------------------|----------------|
| Figure 1         | u-boot fusing configure..... | 18             |

**SAMSUNG Confidential**  
si - plaza / Jeff\_cui at 15:49,2012.08.24

# List of Conventions

## Register RW Access Type Conventions

| Type | Definition   | Description   |
|------|--------------|---|
| R    | Read Only    | The application has permission to read the Register field. Writes to read-only fields have no effect.   |
| W    | Write Only   | The application has permission to write in the Register field.  |
| RW   | Read & Write | The application has permission to read and writes in the Register field. The application sets this field by writing 1'b1 and clears it by writing 1'b0. |

## Register Value Conventions

| Expression       | Description  |
|------------------|--|
| x                | Undefined bit                                      |
| X                | Undefined multiple bits                            |
| ?                | Undefined, but depends on the device or pin status |
| Device dependent | The value depends on the device                    |
| Pin value        | The value depends on the pin status                |

## Reset Value Conventions

| Expression | Description               |
|------------|---------------------------|
| 0          | Clears the register field |
| 1          | Sets the register field   |
| x          | Don't care condition      |

**Warning:** Some bits of control registers are driven by hardware or write operation only. As a result the indicated reset value and the read value after reset might be different.

## List of Acronyms

| Acronyms | Descriptions  |
|----------|---------------|
| D/D      | Device driver |
| H/W      | Hardware      |
| S/W      | Software      |
|          |               |

**SAMSUNG Confidential**  
si - plaza / Jeff\_cui at 15:49,2012.08.24

# 1

## Introduction

This document is intended for providing detailed instructions on building and downloading the Android 4.0(ICS) for TC4 board.

Since the BSP supports SD/MMC and eMMC as the booting device, this document covers building, installing, and flashing images to those flash memory devices.

Chapter 2 describes how to set up the development environment on the host PC.

Chapter 3 describes how to configure and build the bootloader, Android kernel, and Android4.0(ICS) platform.

Chapter 4 describes how to download and fuse the images into the TC4 target board.

Appendix A describes the basic information and procedure of 4412/4212's secure booting

Appendix B describes how to compile 4212/4412's images which don't support TRUSTZONE

Appendix C list the build\_uboot.sh which we are using.

**NOTE:**

This document has been created assuming that the reader is familiar with Embedded Linux Development.

The manners decrypted below were verified in TC4 DVT board which contains AP V1.0/V1.1 and PMIC V2.0/V3.0/V4.0

# 2 System Requirements

To build Android 4.0, 32/64-bit Ubuntu version 10.10 or higher version is required, and tool chain (arm-2009q3) is required too.

## 2.1 Ubuntu Linux (32/64-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 1.6
3. flex, bison, gperf, libsdl-dev, libesd0-dev, libwxgtk2.6-dev (optional), build-essential, zip, curl.

```
# sudo apt-get install xinetd build-essential nfs-kernel-server apache2 samba git-core gnupg flex bison gperf  
libsdl-dev libesd0-dev libwxgtk2.6-dev build-essential zip curl libncurses5-dev zlib1g-dev cscope u-boot-mkimage
```

For more information, refer to following web site.

<http://source.android.com/source/download.html>

si - plaza / Jeff\_cui at 15:49, 2012.08.24

## 2.2 ToolChain Installation

To build the Android BSP, you need to install the tool chain arm-2009q3

Extract arm-2009q3.tar.bz2 to /usr/local/arm

```
#cd /home/YOURNAME/
```

```
#vim .bashrc (attach following line to the ending)
```

---

```
-----
```

```
export PATH=$PATH:/usr/local/arm/arm-2009q3/bin
```

---

```
-----
```

```
#source .bashrc
```

Please find the toolchain(IA32 GNU/Linux TAR) at <http://www.codesourcery.com/sgpp/lite/arm/portal/release1600>

**SAMSUNG Confidential**  
si - plaza / Jeff\_cui at 15:49,2012.08.24

# 3 Build Images

Assume that you have gotten TC4's images (named like: TC4.uboot.dvt.01.ics.rtm1.0.0.tar.bz2, TC4.kernel.dvt.01.ics. rtm1.0.0.tar.bz2 and TC4.android.dvt.01.ics. rtm1.0.0.tar.bz2)

Extract three sources' files into the Ubuntu server. It is recommended to extract UBOOT, Kernel and Android under the same directory(Otherwise, you have to modify the Kernel\_DIR context of TC4\_ICS/build\_android.sh). And extract files name like TC4\_uboot, TC4\_Kernel\_3.0 and TC4\_ICS

## 3.1 Build Uboot

U-BOOT is the boot loader of Android BSP for TC4 board.

### 3.1.1 Main features of U-BOOT

Support fusing bootloader, kernel, RAMDISK, system image into eMMC or SD/TF card

Support loading kernel image and RAMDISK image from eMMC or SD/TF

### 3.1.2 Compile Procedure:

To Compile the U-boot, besides the u-boot source code, there is a zip file named "CodeSign4SecureBoot.rar" is needed. Actually, it contains securing booting/trustzone files which are related with Samsung's Pegasus chip secure booting, you have to get it from Samsung.[Detailed information please see Appendix A & Appendix B which descript trustzone and secure booting]

After you got the securing booting and trustzone files from Samsung, you should make a directory called CodeSign4SecureBoot and put the files into this directory. Please pay attention, you should make the CodeSign4SecureBoot in the same directory with TC4\_uboot, otherwise, you have to modify the build\_uboot.sh which is in TC4\_uboot

After finished upper steps, you can start the u-boot compile.

[1]: enter the directory TC4\_uboot

[2]: execute the command below for u-boot compile

For TC4 EVT/DVT board

```
./build_uboot.sh
```

For TC4Plus board

```
./build_uboot.sh tc4_plus
```

After this command is executed, there are several bin files are generated in this directory.

u-boot-exynos4212-evt0-nonfused.bin : the image used for TC4 EVT(dual core) board.

u-boot-exynos4212-evt1-efused.bin: the images used for TC4 DVT(dual core) board

u-boot-exynos4212-evt1-efused-tz.bin: the images used for TC4 DVT(dual core) board supporting Trustzone

u-boot-exynos4412-evt0-nonfused.bin: the image used for TC4 EVT(quad core) board

u-boot-exynos4412-evt1-efused.bin: the image used for TC4 DVT(Quad core) board

u-boot-exynos4412-evt1-efused-tz.bin: the image used for TC4 DVT(Quad core) board supporting Trustzone

### 3.2 Build Kernel

[1]: Change to kernel(TC4\_Kernel\_3.0) directory

[2]: run commands listed below, after that, the zImage will be generated

```
#make distclean  
si - plaza / Jeff_cui at 15:49,2012.08.24  
#make tc4_dvt_ap11_icecream_nfc_defconfig //Support NFC & Trustzone  
  
#make
```

#### NOTE:

1. tc4\_evt\_icecream\_defconfig is for TC4 EVT board
2. tc4\_dvt\_icecream\_defconfig is for TC4 DVT AP1.0 board which does not support NFC function
3. tc4\_dvt\_ap11\_icecream\_nfc\_defconfig is for TC4 DVT AP1.1 board supporting Trustzone and NFC function
4. tc4\_dvt\_ap11\_icecream\_nfc\_defconfig is for TC4 DVT AP1.1 board supporting Trustzone but not NFC function
5. Please use the corresponding config file for the tc4 board

### 3.3 Build Android

[1]: Change to Android(TC4\_ICS) directory

[2]: Run command listed below, after that, ramdisk.img, system.img & update.zip will be generated

```
#make clobber  
#./build_android.sh
```

**NOTE:**

RAMDISK(ramdisk-uboot.img) : The root file system

System(system.img) : /system

Userdata(userdata.img) : /data

Cache : /cache

In addition, a special package file, "update.zip" has the kernel image, the RAMDISK image, and the system image.

**SAMSUNG Confidential**  
si - plaza / Jeff\_cui at 15:49,2012.08.24

# 4 Flashing Images

## 4.1 Requirements & Tools

### 4.1.1 JTAG

Trace32 or Codeviser

TC4 CMM file

### 4.1.2 Fastboot Tool (for windows)

Fastboot.exe

Usb device driver

Adb.exe

Ums driver

### 4.1.3 SD fuse tool

SDMMC\_Fusing\_tool\_20100115.exe(for windows)

sd\_fusing\_exynos4412.sh(for Linux) (which is in TC4\_uboot\sdifuse\_q\)

## 4.2 Board Booting Sequence

### 4.2.1 bootloader

In the TC4 board, by default, the system will boot from internal eMMC, except below two conditions:

1, eMMC is empty

2, or the context in eMMC is invalid

Then the iRom will try to load bootloader from SD/MMC card.

And at anytime, we can use trace32 with tc4\_cmm.cmm file(which is in TC4\_uboot\ ) to boot the board too.

## **4.3 Flash bootloader into booting device**

### **4.3.1 Flash bootloader into SD/MMC**

There are five manners to fuse bootloader into SD/MMC.

#### **4.3.1.1 Linux sheet(recommended)**

Assume that now the eMMC is empty(or invalid), and SD/MMC is empty too.

[1]: insert SD/MMC card to Linux PC, in /dev directory, it will generated /dev/sdb(or sdc...) node

[2]: change to TC4\_uboot directory

[3]: in the Terminal of ubuntu input the corresponding commands as below:

For TC4 EVT board

`./mkuboot 4212 /dev/sdb`

For TC4(DVT)/TC4Plus board

`./mkuboot 4412tz /dev/sdb`

[4]: after this , the corresponding uboot bin file will be flash into SD/MMC card

[5]: plug out the SD/MMC and insert into TC4 board

[6]: Power on and it will boot from SD/MMC card and it will enter into uboot mode

#### **4.3.1.2 Windows fusing tool**

[1]:Run SDMMC\_Fusing\_tool\_20100115.exe in windows

[2]:Setting like the following for u-boot.

错误！未找到引用源。 shows an example of u-boot image fusing for TC4(DVT)/TC4Plus board.

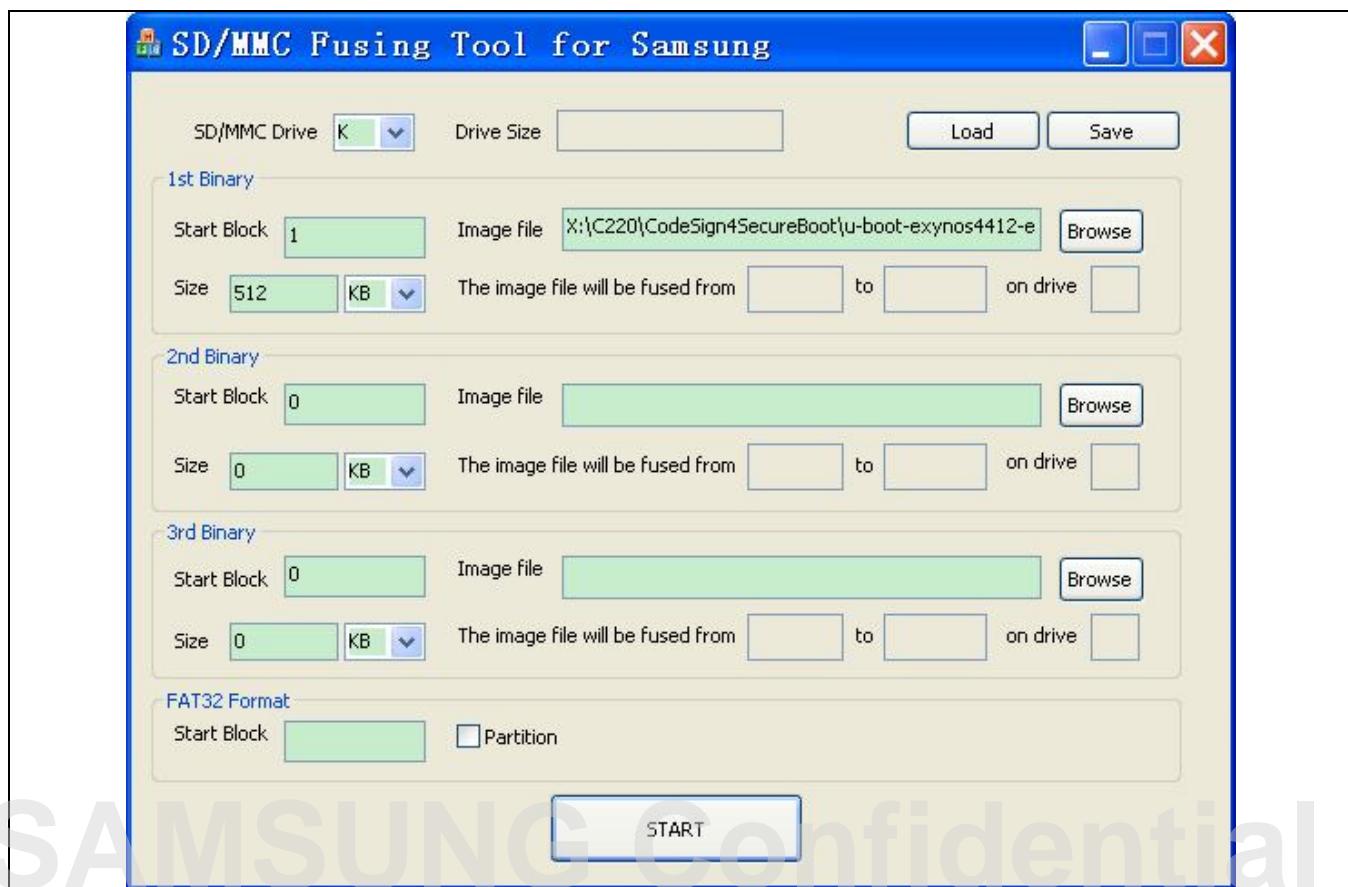


Figure 1 u-boot fusing configure

#### **4.3.1.3 Trace32**

Assume that now the eMMC is empty(or invalid), and SD/MMC is empty too.

[1]:use trace32(+cmm) to boot the board and enter into uboot mode

[2]:in the terminal of board input commands:

```
movi write u-boot 40000000
```

[3]:After finished write, reset the board

#### **4.3.1.4 Fastboot tool**

[1]: boot the board with trace32 or eMMC or SD

[2]: in the terminal of board input commands:

```
#fastboot 1
```

[3]: plug-in usb cable

[4]: Start windows pc command window, change to fastboot directory, input:

fastboot flash bootloader \path\ u-boot-exynos4x12-xvt1-efused.bin('x' depends on the corresponding target board)

[5]:Reset the board

### **4.3.2 Flash bootloader into eMMC**

There are three manners to fuse the bootloader into eMMC.

#### **4.3.2.1 Update from SD card**

Assume that now the eMMC is empty

[1]: Fuse the bootloader into SD/MMC(refer to “Section 4.3.1 Flash bootloader into SD/MMC”)

[2]: Copy the uboot/zImage/Android images into SD/MMC through linux pc or windows pc with FAT file system

[3]: Plug-in SD/MMC into TC4 board

[4]: Press “volume up key” and power on or use the command below at u-boot command mode:

```
sdfuse flashall
```

[5]: System will boot with SD and copy the uboot/zImage/Android images into eMMC automatically

[6]: After copy finished, reset the board

#### **4.3.2.2 Trace32**

[1]: use trace32(+cmm) to boot the board

[2]: in Terminal input commands

movi write emmc-uboot 40000000        (in the cmm file, the corresponding uboot file has been loaded into memory, pls check the cmm file and the corresponding uboot file)

[3]: reset the board

#### **4.3.2.3 Fastboot tool(recommended)**

[1]: boot the board with trace32 or eMMC or SD

[2]: in the terminal of board input commands:

#fastboot

[3]: plug-in usb cable

[4]: Start windows pc command window, change to fastboot directory, input:

fastboot flash bootloader \path\ u-boot-exynos4x12-xvt1-efused.bin('x' depends on the corresponding target board)

[5]: Reset the board

### **4.4 Flash LOGO Resource into device**

#### **4.4.1 Flash LOGO into SD Card**

[1]: In the terminal of board run command:

#fastboot 1

[2]: plug-in usb cable

[3]: Start windows pc command window, change to fastboot directory, input:

>fastboot flash logo \path\logo\_resource.bin

#### **4.4.2 Flash LOGO into emmc Card**

[1]: In the terminal of board run command:

#fastboot

[2]: plug-in usb cable

[3]: Start windows pc command window, change to fastboot directory, input:

```
>fastboot flash logo \path\logo_resource.bin
```

logo\_resource.bin is released with TC4\_uboot source code

## **4.5 Flash OS(Kernel/Android images) into device**

### **4.5.1 Flash OS into SD/MMC**

[1]: Boot the board and run the command listed below to format the SD card:

```
#fdisk -c 1  
#fatformat mmc 1:1  
#ext3format mmc 1:2  
#ext3format mmc 1:3  
#ext3format mmc 1:4
```

[2]: In the terminal of board run command:

```
#fastboot 1
```

[3]: plug-in usb cable

[4]: Start windows pc command window, change to fastboot directory, input:

```
>fastboot flash kernel \path\zImage (check the corresponding zImage with the EVT/DVT board)  
>fastboot flash ramdisk \path\ramdisk-uboot.img  
>fastboot flash system \path\system.img  
>fastboot -w
```

[5]: After transfer finished, Reset the board

### **4.5.2 Flash OS into eMMC**

[1]: Boot the board and run the command listed below to format the EMMC card:

```
#fdisk -c 0  
#fatformat mmc 0:1  
#ext3format mmc 0:2
```

```
#ext3format mmc 0:3
```

```
#ext3format mmc 0:4
```

[2]: In the terminal of board run command:

```
#fastboot
```

[3]: plug-in usb cable

[4]: Start windows pc command window, change to fastboot directory, input:

```
>fastboot flash kernel \path\zImage(check the corresponding zImage with the EVT/DVT board)
```

```
>fastboot flash ramdisk \path\ramdisk-uboot.img
```

```
>fastboot flash Recovery \path\ramdisk-recovery-uboot.img
```

```
>fastboot flash system \path\system.img
```

```
>fastboot -w
```

[5]: After transfer finished, reset the board

## 4.6 Update Images From SD Card

### 4.6.1 Updating automatically

[1]: Fuse the bootloader into SD or eMMC(refer to “Section 4.3.1 Flash bootloader into SD/MMC”)

[2]: Insert the SD card into Linux pc or windows pc with FAT file system and make a directory named “sdupdate”

[3]: Copy the uboot/ramdisk/recovery/kernel/Android images into SD card

[4]: Insert the SD card into the TC4 board

[5]: Keep the “volume up” key pressing down while power on until the information below is shows in your terminal.

```
$Checking Boot Mode ... EMMC4.41  
$SYSTEM ENTER Updating MODE1[0x0]
```

[6]: After copy finished, reset the board

### 4.6.2 Updating Manually

[1]: Fuse the bootloader into SD or eMMC(refer to “Section 4.3.1 Flash bootloader into SD/MMC”)

[2]: Insert the SD card into Linux pc or windows pc with FAT file system and make a directory named “sdupdate”

---

[3]: Copy the uboot/ramdisk/recovery/kernel/Android images into SD card

[4]: Insert the SD card into the TC4 board

[5]: Power on the board and make it enter u-boot mode(press any key before system enter kernel)

[6]: Input the command below in the terminal

```
$sdfuse flashall
```

[7]: After copy finished, reset the board

#### **4.6.3 Updating Notice**

Either automatically or manually updating can be used for whole system updating. There are several notices you need know.

1: During the updating process, all of the partition on your target disk (eMMC here) will be erased. So you need do the backup work firstly.

2: In default, the updating function will update all of the system including: bootloader, ramdisk, recovery, kernel, system.

3: Except system image, you can choose the any image you want to update. For example you just want to update the kernel image, you can only copy the kernel image into "sdupdate" directory in your SD card.

si - plaza / Jeff\_cui at 15:49, 2012.08.24

### **4.7 Recovery System**

The main function of recovery system is used for either system recovery or update. It mainly includes three functions:

1): wipe data : this function is used to erase the data partition in your system

2):wipe cache: this function is used to erase the cache partition in your system

3):update the system from external storage: this function is used for update the kernel,system form SD card.

Basically, this function is not included in the android system. To use this function , please compile the android one more time based on the instruction below.

#### **4.7.1 Compile the Recovery function**

Step1: please ensure you have compiled the android and generated the system.img and ramdisk-uboot.img

Step2: find the file in BoardConfig.mk in your device directory.

For TC4 system, this directory is

TC4\_ICS\device\samsung\smdk4x12\BoardConfig.mk

Step3:changed the option likes below

TARGET\_NO\_KERNEL := false

Step4: Compile the android system using the command below:

./build\_android.sh recovery

When the the compile is over, you will find below files generated in out directory.

1): full\_smdk4x12-ota-eng.xxx.zip (xxx: user name)

2): ramdisk-recovery.img

#### **4.7.2 How to use the recovery function**

Step1: Before using this function, please flash the file “ramdisk-recovery-uboot.img” into your emmc.

For the image flashing, please refer section “4.4.2.1 fastboot tool”

Step2: put the file “full\_smdk4x12-ota-eng.xxx.zip” in the SD card root directory and insert the SD card into the TC4 board.

si - plaza / Jeff\_cui at 15:49, 2012.08.24

Step3: Keeps the “volume down” key pressing down when power-up until the TC4 logo is displayed. Wait for a moment; you will see the update screen and menu.

Step4: You can choose the corresponding function you want.

# 5

## Appendix A: Secure Booting

### 5.1 Background

Secure booting is required by Pegasus4412/4212's iROM. So when you use 4412/4212, you have to sign your uboot file, otherwise, your uboot can NOT be run.

### 5.2 Requirement

Before beginning, you should check/get below files from samsung FAE or Agent.

After you have gotten all of these files, please create a directory called "CodeSign4SecureBoot" and put them into this directory.

For conveniently, we suggest that you should put CodeSign4SecureBoot directory with TC4\_uboot togther, otherwise, you need modify the build\_uboot.sh in TC4\_uboot

| File Name(example)   | Use                                     | Remark  |
|--|---|---|
| SEC_Customer_Service_CodeSigner_V2.1_Client_Guide_Document.pdf | Codesigner_V21's user manual            | C-Dance   |
| SEC_Customer_Service_CodeSigner_V2.1_Client.zip                | CodeSinger_V21.exe and chksum.exe tools | C-Dance   |
| all00_padding.bin  | used to fill uboot                      | C-Dance   |
| codesigner_v21   | sign tool(Linux)                        | C-Dance   |
| E4212_S.bl1.SSCR.EVT1.1.bin                                    | bl1 of Exynos4212                       | Samsung provide it based on customers' public key which generated by Codesigner_V21 and customers' uboot.bin file |
| E4412_bl2_tz.bin   | bl2 of Exynos4412 with trustzone        | C-Dance   |

|                      |                           |   |
|----------------------|---------------------------|---|
| E4412_S.SSCR.bl1.bin | bl1 of Exynos4412         | Samsung provide it based on customers' public key which generated by Codesigner_V21 and customers' uboot.bin file |
| E4412_tzsw.bin       | trustzone of Exynos4412   | C-Dance   |
| Exynos4212_V21.prv   | private key of Exynos4212 | Customers generate it with Codesign_v21 tool  |
| Exynos4212_V21.spk   | public key of Exynos4212  | Customers generate it with Codesign_v21 tool  |
| Exynos4412_V21.prv   | private key of Exynos4412 | Customers generate it with Codesign_v21 tool  |
| Exynos4412_V21.spk   | public key of Exynos4412  | Customers generate it with Codesign_v21 tool  |
| paddingaa            | used to fill uboot        | C-Dance   |

### 5.3 Build Secure booting uboot bin files

enter into TC4\_Uboot directory, run "./build\_uboot.sh", (if customers has gotten bl1/bl2 and secure sign from samsung), it will generated below files

u-boot-exynos4212-evt0-nonfused.bin: the image used for TC4 EVT(dual core) board.

u-boot-exynos4212-evt1-efused.bin: the images used for TC4 DVT(dual core) board

u-boot-exynos4212-evt1-efused-tz.bin: the images used for TC4 DVT(dual core) board supporting trustzone function

u-boot-exynos4412-evt0-nonfused.bin: the image used for TC4 EVT(quad core) board

u-boot-exynos4412-evt1-efused.bin: the image used for TC4 DVT(Quad core) board

u-boot-exynos4412-evt1-efused-tz.bin: the image used for TC4 DVT(Quad core) board supporting Trustzone

**SAMSUNG Confidential**  
si - plaza / Jeff\_cui at 15:49,2012.08.24

# 6

## Appendix B: Trustzone & 4212/4412

### 6.1 Background

#### 6.1.1 Trustzone & AP

For Pegasus4412 V1.1, you have to enable Trustzone function; for Pegasus4412 V1.0/0.1, you can select to use or not use Trustzone function.

For Pegasus4212, you can select to use or not use Trustzone function.

#### 6.1.2 Trustzone & Software

If uboot enable trustzone, the kernel HAVE TO support trustzone function; if uboot disable trustzone function, the kernel HAVE TO disable it. Otherwise, if the kernel does not match with uboot, the kernel can NOT boot successfully.

### 6.2 Requirement

E4212.BL1.TZ.SSCR.EVT1.1.bin for 4212(name is example)

E4212.BL2.TZ.SSCR.EVT1.1.bin for 4212(name is example)

E4212.TZ.SSCR.EVT1.1.bin for 4212(name is example)

E4412.BL2.TZ.SSCR.EVT1.1.bin for 4412(name is example)

E4412.TZ.SSCR.EVT1.1.bin for 4412(name is example)

You should get upper trustzone bin files from Samsung FAE or agent, without these files, you can NOT generate the uboot bin file supporting trustzone function

### 6.3 Build Trustzone & Non-Trustzone manner

#### 6.3.1 Build Trustzone manner

By default, without any modification, you will generate the uboot bin /kernel zImage supporting for 4412 (we don't have 4212's trustzone files, so by now, we have not verified 4212's trustzone.)

1, uboot

---

enter into TC4\_Uboot directory, run "./build\_uboot.sh", (if customers has gotten bl1/bl2 and secure sign from samsung), it will generated below files

u-boot-exynos4212-evt0-nonfused.bin: the image used for TC4 EVT(dual core) board.

u-boot-exynos4212-evt1-efused.bin: the images used for TC4 DVT(dual core) board

u-boot-exynos4212-evt1-efused-tz.bin: the images used for TC4 DVT(dual core) board supporting Trustzone

u-boot-exynos4412-evt0-nonfused.bin: the image used for TC4 EVT(quad core) board

u-boot-exynos4412-evt1-efused.bin: the image used for TC4 DVT(Quad core) board

u-boot-exynos4412-evt1-efused-tz.bin: the image used for TC4 DVT(Quad core) board supporting Trustzone

## 2, Kernel

(by default, it is used trustzone)

make tc4\_dvt\_icecream\_defconfig/tc4\_dvt\_ap11\_icecream\_nfc\_defconfig/tc4\_dvt\_ap11\_icecream\_defconfig

make

--> it will generate zImage supporting trustzone function for 4412 & 4212

### 6.3.2 Build Non-Trustzone manner

1, uboot    si - plaza / Jeff\_cui at 15:49, 2012.08.24

enter into tc4\_uboot directory, run "./build\_uboot.sh", (if customers has gotten bl1/bl2 and secure sign from samsung), it will generated below files:

u-boot-exynos4212-evt0-nonfused.bin : the image used for TC4 EVT(dual core) board.

u-boot-exynos4212-evt1-efused.bin: the images used for TC4 DVT(dual core) board

u-boot-exynos4412-evt0-nonfused.bin: the image used for TC4 EVT(quad core) board

u-boot-exynos4412-evt1-efused.bin: the image used for TC4 DVT(Quad core) board

u-boot-exynos4412-evt1-efused-tz.bin: the image used for TC4 DVT(Quad core) board supporting Trustzone

## 2, Kernel:

(4412 & 4212 can use the same zImage file).

make tc4\_dvt\_icecream\_defconfig

make menuconfig

disable "support trustzone-enabled Trusted Execution Environment" in "system type" menu

---

0.23.61 (1) | 109.120.23.61 (2) | 109.120.23.61 (3)

### 3.0.15 Kernel Configuration

**System Type**

Navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys  
 - excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>  
 [ ] excluded <M> module < > module capable

---

**EXYNOS system type (Samsung Exynos4) --->**

- [ ] Exynos4 Content Path Protection
- TC4 Android Version (TC4 Icecream Android4.0) --->
- TC4 Hardware Version (TC4 DVT Board) --->
- TC4 AP Version (TC4 AP 1.0) --->
- [\*] Exynos4 Power Management
- [\*] Exynos4 CPUIDLE Feature
- [\*] Use thermal management
  - Thermal management Debug
- [\*] ARM core clock down feature enable
  - Max CPU frequency (Max 1400MHZ CPUFREQ LEVEL) --->
  - Support dynamic CPU Hotplug --->
  - Busfreq Model (Busfreq with OPP) --->
- [\*] EXYNOS4 Machines --->
- [\*] GPIO\_AS\_I2C driver
- MMC/SD slot setup --->
  - \*\*\* Miscellaneous drivers \*\*\*
- [ ] Wakeup assist driver
  - \*\*\* Processor Type \*\*\*
  - \*\*\* Processor Features \*\*\*
- [\*] Support TrustZone-enabled Trusted Execution Environment
- [\*] Support Thumb user binaries
- [\*] Enable ThumbEE CPU extension
- [\*] Emulate SWP/SWPB instructions
- [\*] Disable I-Cache (I-bit)
- [\*] Disable D-Cache (C-bit)
- [\*] Disable branch prediction
- [\*] Enable the L2x0 outer cache controller
- [\*] ARM errata: Stale prediction on replaced interworking branch
- [\*] ARM errata: Processor deadlock when a false hazard is created

then save and exit

make

--> it will generate zImage does not supporting trustzone function

# 7

## Appendix C: build\_uboot.sh

```
#!/bin/sh

#./build_uboot.sh ---> build the uboot images for tc4

#./build_uboot.sh tc4_plus -->build the uboot images for tc4_plus

#./build_uboot.sh clean ----> clean the images

#./build_uboot.sh windows ----> encrypt the uboot image in window pc
```

```
option1="tc4_plus"
sec_path="../CodeSign4SecureBoot/"
CPU_JOB_NUM=$(grep processor /proc/cpuinfo | awk '{field=$NF};END{print field+1}')
ROOT_DIR=$(pwd)
CUR_DIR=${ROOT_DIR##*/}
```

```
case "$1" in
clean)
    echo make clean
    rm u-boot.bin
    rm u-boot-exynos4412-evt0-nonfused.bin
    rm u-boot-exynos4212-evt0-nonfused.bin
    rm u-boot-exynos4412-evt1-efused.bin
    make mrproper
;;

```

windows)

```
make tc4_android_config  
make -j$CPU_JOB_NUM  
echo "*****"  
echo "[NOTICE]please copy \"checksum_bl2_14k.bin\" and \"u-boot.bin\" to your"  
echo " windows pc for encryption and gernerate the final bin file for your using "  
echo "*****"  
;;
```

\*)

```
if [ ! -d $sec_path ]
```

```
then
```

```
echo "*****"  
echo "[ERR]please get the CodeSign4SecureBoot first"  
echo "*****"  
return
```

```
fi
```

```
if [ -z $1 ]
```

```
then
```

```
    make tc4_android_config
```

```
elif [ $1 = $option1 ]
```

```
then
```

```
    make tc4_plus_android_config
```

```
else
```

```
    echo please input right parameter.
```

```
    exit 0
```

```
fi
```

```
make -j$CPU_JOB_NUM
```

```
if [ ! -f checksum_BL2_14k.bin ]
```

```
then
```

```
    echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
```

```
    echo "There are some error(s) while building uboot, please use command make to check."
```

```
    echo "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
```

```
    exit 0
```

```
fi
```

```
cp -rf checksum_BL2_14k.bin $sec_path
```

```
cp -rf u-boot.bin $sec_path
```

```
rm checksum_BL2_14k.bin
```

```
cd $sec_path
```

```
./codesigner_v21 -v2.1 checksum_BL2_14k.bin BL2.bin.signed.4412 Exynos4412_V21.prv -STAGE2
```

```
./codesigner_v21 -v2.1 checksum_BL2_14k.bin BL2.bin.signed.4212 Exynos4212_V21.prv -STAGE2
```

SAMSUNG Confidential  
SI-plaza / Jeff\_cui at 15:49, 2012.08.24

```
cat E4412.S.BL1.SSCR.EVT1.1.bin BL2.bin.signed.4412 paddingaa u-boot.bin > u-boot-exynos4412-evt1-efused.bin
```

```
cat E4212.S.BL1.SSCR.EVT1.1.bin BL2.bin.signed.4212 paddingaa u-boot.bin > u-boot-exynos4212-evt1-efused.bin
```

```
# generate the uboot bin file support trust zone
```

```
cat E4412.S.BL1.SSCR.EVT1.1.bin E4412.BL2.TZ.SSCR.EVT1.1.bin all00_padding.bin u-boot.bin  
E4412.TZ.SSCR.EVT1.1.bin > u-boot-exynos4412-evt1-efused-tz.bin
```

```
cat E4212.BL1.TZ.SSCR.EVT1.1.bin E4212.BL2.TZ.SSCR.EVT1.1.bin u-boot.bin  
E4212.TZ.SSCR.EVT1.1.bin > u-boot-exynos4212-evt1-efused-tz.bin
```

```
mv u-boot-exynos4212-evt1-efused.bin $ROOT_DIR  
mv u-boot-exynos4412-evt1-efused.bin $ROOT_DIR  
mv u-boot-exynos4412-evt1-efused-tz.bin $ROOT_DIR  
mv u-boot-exynos4212-evt1-efused-tz.bin $ROOT_DIR
```

```
rm checksum_bl2_14k.bin  
rm BL2.bin.signed.4412  
rm BL2.bin.signed.4212  
rm u-boot.bin
```

echo u-boot-exynos4412-evt1-efused.bin generated for Quad core secure boot,use this one to fuse exynos 4412 evt1.

echo echo Please use sd\_fusing.sh in sdfuse\_q directory with su permission for programming the u-boot to SD card. Ex: ./sd\_fusing.sh /dev/sdb

```
echo  
echo  
;;
```

esac