

TCC892X ANDROID SDK QUICK START GUIDE

TCC892x_Android 4.4.2(Kitkat-mr1.1)_v1.00E

Rev. 1.00

Apr. 1. 2014

Telechips

DISCLAIMER

All information and data contained in this material are without any commitment, are not to be considered as an offer for conclusion of a contract, nor shall they be construed as to create any liability. Any new issue of this material invalidates previous issues. Product availability and delivery are exclusively subject to our respective order confirmation form; the same applies to orders based on development samples delivered. By this publication, Telechips, Inc. does not assume responsibility for patent infringements or other rights of third parties that may result from its use.

Further, Telechips, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of Telechips, Inc.

This product is designed for general purpose, and accordingly customer be responsible for all or any of intellectual property licenses required for actual application. Telechips, Inc. does not provide any indemnification for any intellectual properties owned by third party.

Telechips, Inc. can not ensure that this application is the proper and sufficient one for any other purposes but the one explicitly expressed herein. Telechips, Inc. is not responsible for any special, indirect, incidental or consequential damage or loss whatsoever resulting from the use of this application for other purposes.

COPYRIGHT STATEMENT

Copyright in the material provided by Telechips, Inc. is owned by Telechips unless otherwise noted.

For reproduction or use of Telechips' copyright material, permission should be sought from Telechips. That permission, if given, will be subject to conditions that Telechips' name should be included and interest in the material should be acknowledged when the material is reproduced or quoted, either in whole or in part. You must not copy, adapt, publish, distribute or commercialize any contents contained in the material in any manner without the written permission of Telechips. Trade marks used in Telechips' copyright material are the property of Telechips.

Important Notice

For customers who use licensed Codec ICs and/or licensed codec firmware of mp3:

"Supply of this product does not convey a license nor imply any right to distribute content created with this product in revenue-generating broadcast systems (terrestrial, Satellite, cable and/or other distribution channels), streaming applications(via internet, intranets and/or other networks), other content distribution systems(pay-audio or audio-on-demand applications and the like) or on physical media(compact discs, digital versatile discs, semiconductor chips, hard drives, memory cards and the like). An independent license for such use is required. For details, please visit <http://mp3licensing.com>".

For customers who use other firmware of mp3:

"Supply of this product does not convey a license under the relevant intellectual property of Thomson and/or Fraunhofer Gesellschaft nor imply any right to use this product in any finished end user or ready-to-use final product. An independent license for such use is required. For details, please visit <http://mp3licensing.com>".

For customers who use Digital Wave DRA solution:

"Supply of this implementation of DRA technology does not convey a license nor imply any right to this implementation in any finished end-user or ready-to-use terminal product. An independent license for such use is required."

For customers who use DTS technology:

"This product made under license to certain U.S. patents and/or foreign counterparts."
"© 1996 – 2011 DTS, Inc. All rights reserved."

For customers who use Dolby technology:

"Supply of this Implementation of Dolby technology does not convey a license nor imply a right under any patent, or any other industrial or intellectual property right of Dolby Laboratories, to use this Implementation in any finished end-user or ready-to-use final product. It is hereby notified that a license for such use is required from Dolby Laboratories."

For customers who use MS technology:

"This product is subject to certain intellectual property rights of Microsoft and cannot be used or distributed further without the appropriate license(s) from Microsoft."

Revision History

Date	Version	Description
2014-04-01	1.00	Initial release.

TABLE OF CONTENTS

Contents

1 Introduction	1-5
2 Features.....	2-5
3 Download the SDK.....	3-5
4 Compile and build Android Framework	4-6
4.1 Setup compile environment	4-6
4.2 Compile Bootloader	4-7
4.2.1 TCC892x	4-7
4.2.1.1 TCC8925s	4-7
4.2.2 M805 Reference Board.....	4-8
4.3 Compile Linux Kernel.....	4-8
4.3.1 Set Kernel default setting.....	4-8
4.3.2 TCC8925s Setting	4-9
4.3.3 Set Kernel's configurations	4-9
4.3.3.1 Set TCC892x Chipset Type	4-9
4.3.4 Compile Kernel	4-9
4.4 Compile Frameworks	4-10
5 How to download Image	5-11
6 SD Slot Usage	6-12
6.1.1 Location of the SD slot at each EVB board.....	6-12
6.1.1.1 TCC8920 EVB Board	6-12
6.1.1.2 TCC8923 D3 CPU Board.....	6-13
6.1.1.3 TCC8925 D3 2CS CPU Board.....	6-14
6.1.2 Settings for High Speed SDHC Card	6-14
7 SD/MMC Booting Guide.....	7-15
7.1 Introduction	7-15
7.2 Prepare Download Image	7-15
7.2.1 How to configure Boot Mode (SD/MMC or eMMC) in Bootloader	7-15
7.2.2 Make Bootloader for SD/MMC or eMMC Booting	7-15
7.2.3 Kernel build for SD/MMC or eMMC booting.....	7-16
7.2.4 Android system build for SD/MMC or eMMC booting.....	7-16
7.3 PartitionLayout.....	7-16
7.3.1 GPT Partition Layout	7-16
7.3.2 Partition size definition.....	7-17
7.3.3 GPT Partition Layout	7-17
7.4 How to select Boot Mode	7-17
7.4.1 SD/MMC Boot Mode.....	7-17
7.4.2 eMMC Boot Mode.....	7-17
7.5 Prepare download with FWDN.....	7-18
7.5.1 GPT Partition Layout - eMMC Boot Mode(Hidden Area is NOT used).....	7-18
7.5.2 GPT Partition Layout – SD/MMC Boot Mode, eMMC Boot mode(Use Hidden Area).....	7-20
7.6 SD Slot Usages	7-22
7.6.1 Location of the SD slot at each EVB board.....	7-22
7.6.2 TCC8920 CPU Board	7-22
7.6.3 TCC8923 D3 CPU Board.....	7-23
7.6.4 TCC8925 D3 2CS CPU Board.....	7-24
7.7 Appendix	7-25

1 Introduction

This document provides guideline for users to start Android 4.4.2(Kitkat-mr1.1) platform for TCC892x quickly.

2 Features

Telechips Android 4.4.2(Kitkat-mr1.1) platform supports following features.

- Updated to Kernel-3.4.35

3 Download the SDK

Download SDK from 'kitkat' branch of Telechips Android GIT server.

```
$ mkdir mydroid
$ cd mydroid

$ repo init -u ssh://android.telechips.com/androidce/android/platform/manifest.git -
b kitkat-mr1.1

$ repo sync
```

You can download SDK using TAG or XML information.

For example,

```
$repo init -u ssh://android.telechips.com/androidce/android/platform/manifest.git -b kitkat -m "XML NAME.xml".
```

or

```
$repo forall -c git checkout "TAG_NAME".
```

4 Compile and build Android Framework

4.1 Setup compile environment

Before compile operation, TARGET_PRODUCT must be set for the proper board configuration. Execute the commands below. There is space between *dot(.)* and '*build/envsetup.sh*' to execute shell scripter.

```
$ cd ~/mydroid/android
$ . build/envsetup.sh
including device/asus/deb/vendorsetup.sh
including device/asus/flo/vendorsetup.sh
including device/asus/grouper/vendorsetup.sh
including device/asus/tilapia/vendorsetup.sh
including device/generic/armv7-a-neon/vendorsetup.sh
including device/generic/mips/vendorsetup.sh
including device/generic/x86/vendorsetup.sh
including device/lge/hammerhead/vendorsetup.sh
including device/lge/mako/vendorsetup.sh
including device/samsung/manta/vendorsetup.sh
including device/telechips/m805_892x/vendorsetup.sh
including device/telechips/m805_893x/vendorsetup.sh
including device/telechips/tcc8920st/vendorsetup.sh
including device/telechips/tcc8920/vendorsetup.sh
including device/telechips/tcc8930st/vendorsetup.sh
including device/telechips/tcc893x/vendorsetup.sh
including sdk/bash_completion/adb.bash
$ lunch
```

You're building on Linux

Lunch menu... pick a combo:

1. aosp_arm-eng
2. aosp_x86-eng
3. aosp_mips-eng
4. vbox_x86-eng
5. aosp_deb-userdebug
6. aosp_flo-userdebug
7. aosp_grouper-userdebug
8. aosp_tilapia-userdebug
9. mini_armv7a_neon-userdebug
10. mini_mips-userdebug
11. mini_x86-userdebug
12. aosp_hammerhead-userdebug
13. aosp_mako-userdebug
14. aosp_manta-userdebug
15. full_m805_892x-eng
16. full_m805_893x-eng
17. full_tcc8920st-eng
18. full_tcc8920-eng
19. full_tcc8930st-eng
20. full_tcc893x-eng

Which would you like? [full-eng]

If you want to use TCC8920, please select "full_tcc8920-eng" with input 18.

※ How to set "user mode"

If you want to set "user mode", you should use "choosecombo" command.

※ This selection should be done before compiling kernel because it affects the kernel configuration of architecture.

4.2 Compile Bootloader

You must **compile bootloader, kernel and frameworks respectively** and compile bootloader and kernel firstly.

"bootable/bootloader/lk"

4.2.1 TCC892x

To compile for TCC892x, "make tcc892x_evm" must be executed.

```
$ cd ~/mydroid/android/bootable/bootloader/lk
$ make tcc892x_evm
```

Then "lk.rom" is created at *"bootable/bootloader/lk/build-tcc892x_evm/lk.rom"*

In addition,

You have to select H/W Revision of EVB Board that you are using.

Change *"bootable/bootloader/lk/target/tcc892x_evm/rules.mk"* for H/W Revision.

```
#-----
# Define board revision
# 0x1000 : TCC8920_D2_08X4_SV01 - DDR2 512MB(32Bit)
# 0x1001 : TCC8920_D2_08X4_SV01 - DDR2 1024MB(32Bit)
# 0x1002 : TCC8920_D3_08X4_SV01 - DDR3 512MB(32Bit)
# 0x1003 : TCC8920_D3_08X4_SV01 - DDR3 1024MB(32Bit)
# 0x1004 : TCC8920_D3_08X4_SV01 - DDR3 512MB(16Bit)
# 0x1005 : TCC8920_D3_16X4_2CS_SV01 - DDR3 1024MB (32Bit)
# 0x1006 : TCC8925_D3_08X4_2CS_SV01 - DDR3 1024MB (16Bit)
# 0x1007 : TCC8920_D3_08X4_SV60 - DDR3 1024MB(32Bit)
# 0x1008 : TCC8923_D3_08X4_SV01 - DDR3 1024MB(32Bit)
#HW_REV=0x1000
#HW_REV=0x1001
#HW_REV=0x1002
#HW_REV=0x1003
#HW_REV=0x1004
#HW_REV=0x1005
#HW_REV=0x1006
HW_REV=0x1007
#HW_REV=0x1008
```

4.2.1.1 TCC8925s

To compile for TCC8925s, "make tcc892x_evm" must be executed.

```
$ cd ~/mydroid/android/bootable/bootloader/lk
$ make tcc892x_evm
```

Then "lk.rom" is created at *"bootable/bootloader/lk/build-tcc892x_evm/lk.rom"*

You have to select TCC8925s chipset at

"bootable/bootloader/lk/target/tcc892x_evm/rules.mk"

```
#-----
# CHIPSET TYPE
#-----
DEFINES += CONFIG_CHIP_TCC8920
#DEFINES += CONFIG_CHIP_TCC8925
#DEFINES += CONFIG_CHIP_TCC8925S
```

4.2.2 M805 Reference Board

To compile for M805, “make m805_892x_evm” must be executed.

```
$ cd ~/mydroid/android/bootable/bootloader/lk
$ make m805_892x_evm
```

“lk.rom” is created at “bootable/bootloader/lk/build-m805_892x_evm/” folder.

In addition,
You have to select H/W Revision of M805 Board that you use.

Change “bootable/bootloader/lk/target/m805_892x_evm/rules.mk” for H/W Revision

```
#-----
# Define board revision
# 0x2000 = TCC8803_M805S_D3_08x4_V0.1(Change to TCC89 - DDR3_512MB (32Bit)
# 0x2001 = TCC8803/8923_M805S_D3_08x4_V1.1 - DDR3_1024MB (32Bit)
# 0x2002 = TCC8925_M805S_D3_08x4_2CS_V0.1 - DDR3_1024MB (16Bit)
# 0x2003 = TCC8925_M805S_D3_08x4_2CS_V1.2 2012.03.15 - DDR3_1024MB (16Bit)
# 0x2004 = TCC8925_M805S_D3_08x2_4L_V0.1A - DDR3_512MB (16Bit)
# 0x2005 = TCC8925_M805S_D3_08x4_2CS_V1.4A 2012.05.08 - DDR3_1024MB (16Bit)
# 0x2006 = TCC8925_M805S_D3_08x2_4L_V1.0 2012.06.26 - DDR3_512MB (16Bit)
# 0x2007 = TCC8925_M805S_D3_08x4(2CS)_4L_V0.1 2012.7.3- DDR3_1024MB (16Bit)
# 0x2008 = TCC8925X_35_M805S_D3_08x4_V0.1 2012.8.17 - DDR3_1024MB (16Bit)
# 0x2009 = TCC8925X_35_M805S_D3_08x4_V0.11 2012.9.20 - DDR3_1024MB (16Bit)
# 0x2009 = TCC8925X_35_M805S_D3_08x4_V1.2 2013.03.26 - DDR3_1024MB (16Bit)
#HW_REV=0x2000
#HW_REV=0x2001
#HW_REV=0x2002
#HW_REV=0x2003
#HW_REV=0x2004
#HW_REV=0x2005
#HW_REV=0x2006
#HW_REV=0x2007
#HW_REV=0x2008
#HW_REV=0x2009
```

4.3 Compile Linux Kernel

Kernel must be compiled firstly. After compile kernel, you must compile boot.img and system.img. If you change Kernel, boot.img must be compiled again.

4.3.1 Set Kernel default setting

Lunch menu must be selected before setting linux kernel configuration, because lunch menu affect the kernel configuration of architecture in Android 4.4.2 release.

The above configuration files locate in *kernel/arch/arm/configs/*

For TCC892x EVM, kernel configuration will be set by tcc892x_defconfig. In addition, you must set correct module in menuconfig.

And for M805, kernel configuration will be set by m805_892x_defconfig.

Move to Linux kernel directory and execute the command below to set default kernel configuration for TCC892x EVM Board.

.config file will be created at kernel directory.

```
$ cd ~/mydroid/android/kernel
$ make tcc892x_defconfig
```


For M805 Board, execute the command below.

```
$ cd ~/mydroid/android/kernel
$ make m805_892x_defconfig
```

4.3.2 TCC8925s Setting

TCC8925S option can be set in “menuconfig”.

“System Type → Select TCC8925S Sub SoC Type”

(If you want to use TCC8925S chip, you should select this feature)

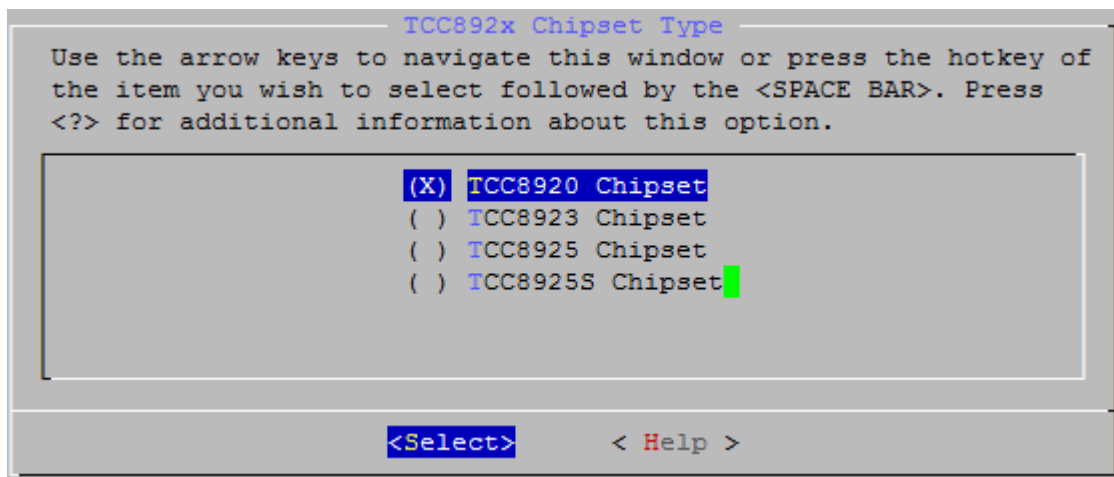
4.3.3 Set Kernel's configurations

After executing the command to set default configuration of the Kernel, you must change some options as below.

4.3.3.1 Set TCC892x Chipset Type

Select “System Type → TCC892x Chipset Type”.

Then, select the chipset type which you use on board as follows.



4.3.4 Compile Kernel

In kernel directory, just execute “make”.

```
$ cd ~/mydroid/android/kernel
$ make
```

4.4 Compile Frameworks

For TCC8925S

You have to change target board at
device/telechips/tcc892x/device.mk

```
#TARGET_BOARD_SOC := tcc892x  
TARGET_BOARD_SOC := tcc892xs
```

Just execute “make” command, you can build Android frameworks. It takes time.

```
$ cd ~/mydroid/android  
$ make
```

If you select proper *TARGET_PRODUCT* with “Setup compile environment” chapter, you can check it with the log below. This log can be seen when you select “full_tcc892x-eng”.

```
=====
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=4.4
TARGET_PRODUCT=full_tcc8920
TARGET_BUILD_VARIANT=eng
TARGET_BUILD_TYPE=release
TARGET_BUILD_APPS=
TARGET_ARCH=arm
TARGET_ARCH_VARIANT=armv7-a-neon
TARGET_CPU_VARIANT=generic
HOST_ARCH=x86
HOST_OS=linux
HOST_OS_EXTRA=Linux-2.6.33.3-85.fc13.x86_64-x86_64-with-fedora-13-Goddard
HOST_BUILD_TYPE=release
BUILD_ID=KRT16M
OUT_DIR=out=====
```

5 How to download Image

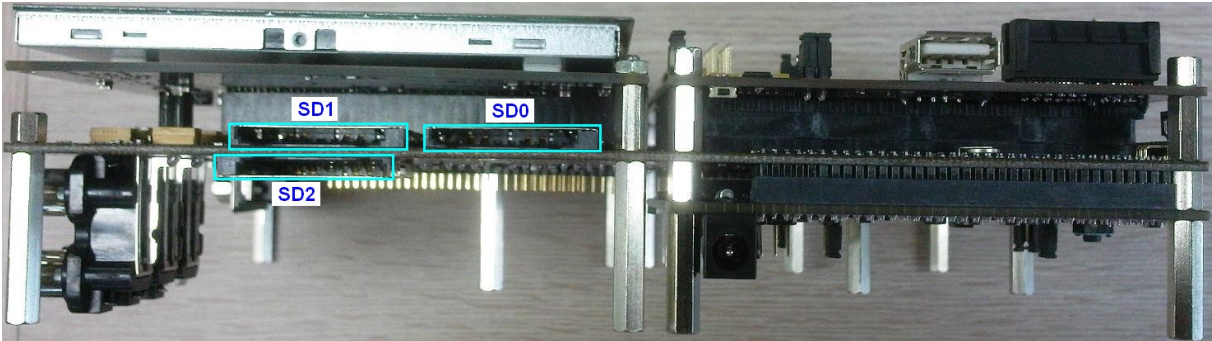
In case of Android 4.4.2(Kitkat-mr1.1) version, the partition layout uses GPT.
FWDN has been supporting GPT since v2.44 so you have to use **FWDN-v2.44 or Higher version** for it.

For more information, please refer to “TCCxxx-Android-ALL-V1.0-Partition Layout Guide.pdf” document in “vendor/telechips/documents/common/pdf”.

6 SD Slot Usage

6.1.1 Location of the SD slot at each EVB board

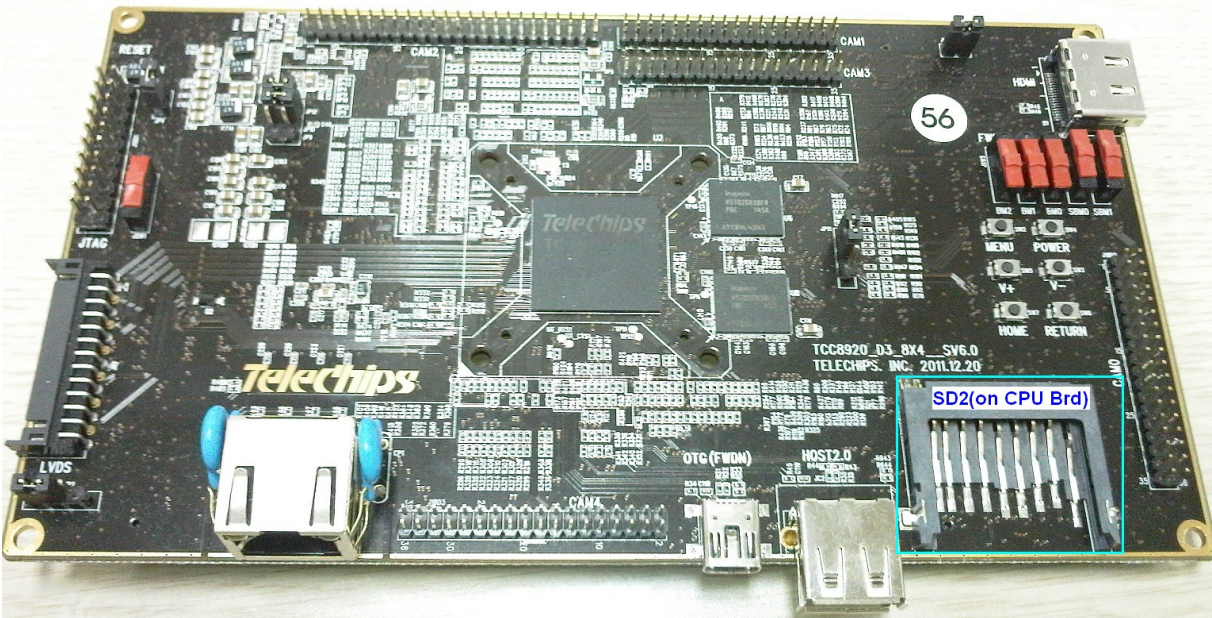
In the TCC892x evm board, its SD2 slot is not used.



[Figure 8.1 Telechips EVM Main Board – Left Side]

6.1.1.1 TCC8920 EVB Board

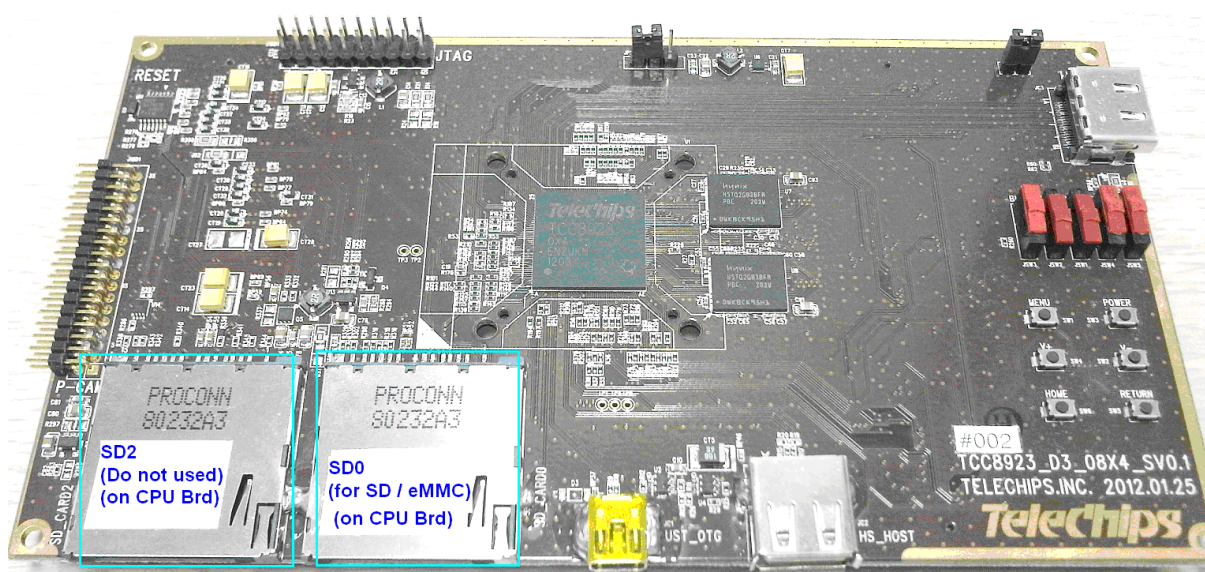
TCC8920 EVM	SD slot	SD + WiFi (Default Setting)	eMMC + SD + WiFi (Optional)
Main Board	SD0	-	eMMC
	SD1	WiFi	WiFi
	SD2	-	-
CPU Board	SD2	SD	SD



[Figure 8.2 TCC8920 D2 CPU Board]

6.1.1.2 TCC8923 D3 CPU Board

TCC8923 EVM	SD slot	SD + WiFi (Default Setting)	eMMC + SD + WiFi (Optional)
Main Board	SD0	-	-
	SD1	WiFi	SD or WiFi
	SD2	-	-
CPU Board	SD0	SD	eMMC
	SD2	-	-

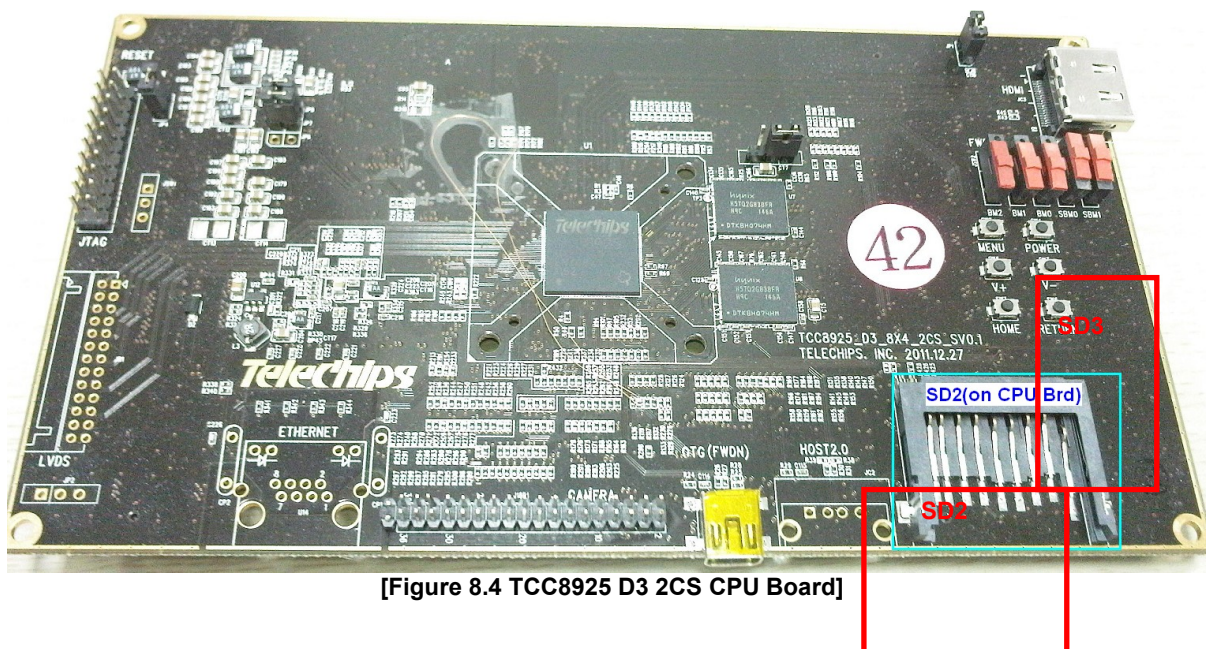


[Figure 8.4 TCC8923 D3 CPU Board]

6.1.1.3 TCC8925 D3 2CS CPU Board

TCC8925 EVM	SD slot	SD + WiFi (Default Setting)	eMMC + SD + WiFi (Optional)
Main Board	SD0	-	eMMC
	SD1	-	-
	SD2	-	-
CPU Board	SD2	SD	SD

(WiFi is using the USB Host Interface.)



[Figure 8.4 TCC8925 D3 2CS CPU Board]

6.1.2 Settings for High Speed SDHC Card

Some SDHC Card is able to occur that access error for SD High-Speed operation. Access error is according to external factors (hardware pattern, passive element value, etc), also EVM board.

If you want to change the SD High-Speed mode support or not, that is following.

```
$ cd ~/mydroid/android
$ vi kernel/arch/arm/mach-tcc892x/board-tcc8920-mmc.c
```

tcc8920_mmc_platform_data[].caps in the Board-tcc8920-mmc.c file,

- 1) Support the SD High-Speed mode
`.caps = MMC_CAP_SDIO_IRQ | MMC_CAP_4_BIT_DATA
| MMC_CAP_SD_HIGHSPEED | MMC_CAP_MMC_HIGHSPEED,`
- 2) Not support the SD High-Speed mode
`.caps = MMC_CAP_SDIO_IRQ | MMC_CAP_4_BIT_DATA
/*| MMC_CAP_SD_HIGHSPEED | MMC_CAP_MMC_HIGHSPEED*/,`

7 SD/MMC Booting Guide

7.1 Introduction

This document describes how to configure for SD/MMC booting in Telechips Android Platform.

7.2 Prepare Download Image

7.2.1 How to configure Boot Mode (SD/MMC or eMMC) in Bootloader

Support two kinds of boot mode: 1)SD/MMC boot mode and 2)eMMC¹ boot mode(eMMC Spec 4.4x²). The SD/MMC boot mode refers that bootloader is stored in normal data partition in which normal code and/or data such as boot(kernel) image, android system image, and user data are stored. The second type of boot mode, eMMC boot mode, uses extra boot partition additionally existed only on eMMC device above eMMC Spec 4.4x. For example, eMMC device(above eMMC Spec 4.4x) can be used for the all of boot mode because it has two kinds of partitions: normal partition and dedicated boot partition. On the contrary to the eMMC device, SD card has only a normal partition. So, if SD card is used as boot device, you have no choice to select boot mode; that is, you must select SD boot mode.

!Caution: When the eMMC boot mode is used, take notice the boot partition size of the eMMC device. If bootloader image size is greater than the boot partition size of the eMMC device, the bootloader image can't be stored on it. Depending on the eMMC manufacturers, the boot partition size differs³. We usually use eMMC device which boot partition size is set at least 1MByte to save bootloader safely. So, please check boot partition size of eMMC device before you decide it to use boot device.

SD/MMC boot mode is selected to default boot mode. To change the boot mode, SD/MMC or eMMC, please change build script in bootloader. If **FEATURE_SDMC_MMC43_BOOT** option is enabled in the bootloader, boot mode is set to eMMC boot mode. Otherwise, the boot mode is set to SD/MMC boot mode.

Location : bootable/bootloader/lk/platform/tcc892x/rules.mk

```
ifeq ($(EMMC_BOOT),1)
DEFINES += BOOTSD_KERNEL_INCLUDE
ifeq ($(filter $(DEFINES),TARGET_TCC8925_STB_DONGLE),)
    ifeq ($(filter $(HW_REV),0x9112),)
        DEFINES += FEATURE_SDMC_MMC43_BOOT
    endif
else
DEFINES += FEATURE_SDMC_MMC43_BOOT
endif
endif
"bootable/bootloader/lk/platform/tcc892x/rules.mk" 81 lines --30%-- 25,1 16%
```

7.2.2 Make Bootloader for SD/MMC or eMMC Booting

1. Change directory to LK bootloader
\$ cd bootable/bootloader/lk
2. Compile bootloader for TCC892x SD/MMC or eMMC Boot mode
\$ make tcc892x_evm_emmc

¹ Generally, MMC and eMMC are meant to same eMMC(embedded MMC) device. To classify different boot mode, however, the terms for SD/MMC boot mode and eMMC boot mode have being used only in Telechips in-evitably.

² Refer to the JEDEC standard document for eMMC, <http://www.jedec.org/>

³ Refer to the datasheet of eMMC manufacturers such as SanDisk, Samsung, Toshiba, and so on.

7.2.3 Kernel build for SD/MMC or eMMC booting

To use both SD/MMC and eMMC boot mode, you must identically configure the kernel as follows.

```
1) Kernel$ make menuconfig
   Device Drivers -> MMC/SD/SDIO card support ->
   [*] Telechips SD/MMC Host Controller Driver
   [*] Support an eMMC
   [*] Enable TCC SDHCO
   [*] Enable TCC SDHC1
   [*] Enable TCC SDHC2
   [ ] Enable TCC SDHC3
```

7.2.4 Android system build for SD/MMC or eMMC booting

- 1). Execute lunch command in android root
- 2). Select full_tcc8920-eng mode

When Android system is compiled, the all images are located in **out/target/product/tcc8920** directory.

7.3 PartitionLayout

You should understand how SD/MMC partition layout is configured in Android system before you download Android system images. See the following table of default SD/MMC partition layout.

From Android v4.4(Kitkat), GPT⁴ partition table is used as default configuration. But, MBR partition layout is also supported to maintain compatibility with Android v4.2(Jellybean). If you want to use MBR partition layout, execute following operation:

1. Move Target Device Directory in Android Kitkat
\$ cd device/telechips/<platform>
2. Overwrite Configuration Files for MBR Partition Layout
\$ cp -rf parttype_mbr/ ./

7.3.1 GPT Partition Layout

Area	Name	Purpose	File System Type	Mandatory
Boot	Boot Area	Kernel/Ram Disk	Raw	Y
System	Android System	Android System Area	Ext4	Y
Cache	Android Cache	Android Cache Area	Ext4	Y
Recovery	Android Recovery	Recovery Mode Boot Area Kernel/Ram Disk(Recovery Mode)	Raw	Y
Splash	Splash	Boot Screen Image	Raw	Y
Kpanic	Kpanic	Kernel Panic Log	Raw	Y
Misc	Miscellaneous	Recovery Mode related Command (FOTA, Factory Rest)	Raw	Y
Tcc	Telechips Only	Set-top Flash Write	Raw	N
Userdata	Android User Data	Android User Data Application / Database	Ext4	Y

⁴ http://en.wikipedia.org/wiki/GUID_Partition_Table

7.3.2 Partition size definition

This layout is the minimum mandatory partition needed in Android. The optional partitions are not necessarily needed for Android system, and it is also used only for special purpose.

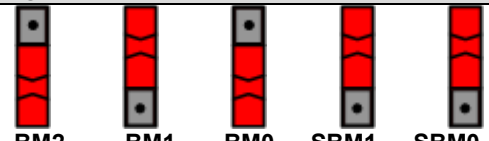
7.3.3 GPT Partition Layout

Area	Size	Partition	File System Type	Mandatory
Boot	15 MByte	mmcblk0p1	Raw	Y
System	650 MByte	mmcblk0p2	Ext4	Y
Cache	150 MByte	mmcblk0p3	Ext4	Y
Recovery	15 MByte	mmcblk0p4	Raw	Y
Splash	5 MByte	mmcblk0p5	Raw	Y
Kpanic	4 MByte	mmcblk0p6	Raw	Y
Misc	1 MByte	mmcblk0p7	Raw	Y
Tcc	1 MByte	mmcblk0p8	Raw	N
Userdata	Remaining Capacity	mmcblk0p9	Ext4	Y

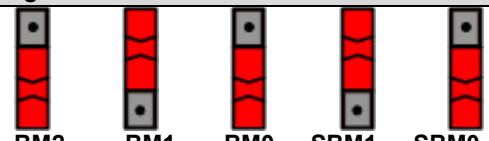
7.4 How to select Boot Mode

As explained previous section, there are two boot modes: SD/MMC and eMMC boot mode. You have to set correct boot mode switch for each boot mode. If you want to know detailed information about boot mode, refer to the boot mode section on tcc892x Specification Document.

7.4.1 SD/MMC Boot Mode

Pin	Value	Figure
BM2	High (0)	
BM1	Low (0)	
BM0	High (1)	
SBM1	Low (0)	
SBM0	Low (0)	

7.4.2 eMMC Boot Mode

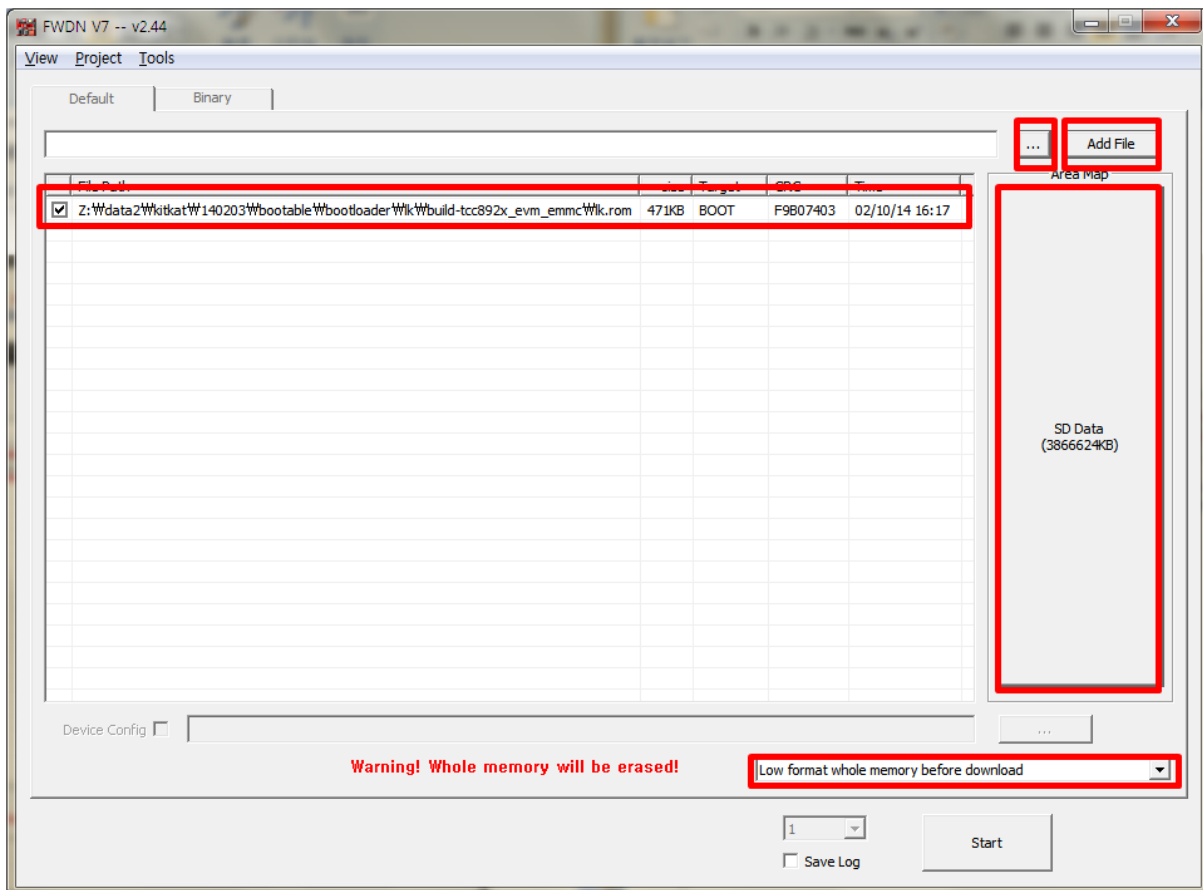
Pin	Value	Figure
BM2	High (1)	
BM1	Low (0)	
BM0	High (1)	
SBM1	Low (0)	
SBM0	High (1)	

7.5 Prepare download with FWDN

For downloading bootloader and Android system images to TCC892x, you have to use FWDN V2.44 or higher version. This section describes how to prepare and download the images.

7.5.1 GPT Partition Layout - eMMC Boot Mode(Hidden Area is NOT used)

*Step 1. Load bootloader image to FWDN, and then attach target devices to FWDN through usb boot mode. If the devices is attached to FWDN successfully, click **SD Data Button***



Step 2. Prepare download images.

Select number of partition, partition size, images and "Use GPT" as following figure. The partition order is same as partition layout described in the previous section. Finally, click **Create Image Button**.

Image Creation Dialog for Area 'SD Data'

Image File Path: SD Data.fai

Tab: Disk Image | LOGO Image | KEY STORE RO | UID

Number of Partition: 9 (highlighted in green box)

Use GPT (highlighted in green box with green arrow)

Partition 1: Partition Size 15360 KB (1MB=1024), Partition Label boot, Image File Path boot.img

Partition 2: Partition Size 665600 KB (1MB=1024), Partition Label system, Image File Path system.img

Partition 3: Partition Size 153600 KB (1MB=1024), Partition Label cache, Image File Path You can optionally generate cache.img (highlighted in blue box)

Partition 4: Partition Size 15360 KB (1MB=1024), Partition Label recovery, Image File Path recovery.img

Partition 5: Partition Size 5120 KB (1MB=1024), Partition Label splash, Image File Path

Partition 6: Partition Size 4096 KB (1MB=1024), Partition Label kpanic, Image File Path

Partition 7: Partition Size 1024 KB (1MB=1024), Partition Label misc, Image File Path

Partition 9: Partition Size 3005407 KB (1MB=1024), Partition Label userdata, Image File Path You can optionally generate userdata.img (highlighted in blue box)

Create Image

Notice :

Green Box - In this configuration, GPT partition table is stored both at Primary GPT and Secondary GPT area.

Blue Box - You can see partition 3 and partition 9 in the blue box: user data partition and cache partitions. Those image paths are empty. From Android 4.2(JellyBean), if user data and cache partitions are not formatted as ext4 file system format, the system will be operated abnormally. For this reason, if the partitions in the blue box are not formatted as ext4 type, system will format those partitions to the ext4 file system at first boot time. Therefore, system will be rebooted again for initializing Android systems. So FWDN does not write any data to that partition if image file path is empty.

Otherwise, if you want to protect formatting file system during initial boot sequence, add following parameter depending on your storage capacity. This configuration generates 1)userdata.img and 2)cache.img in your \$OUT/ directory. Then choose the images in FWDN tool.

```
BOARD_FLASH_BLOCK_SIZE := 4096
BOARD_SYSTEMIMAGE_PARTITION_SIZE := 681574400 #650MB System
BOARD_USERDATAIMAGE_PARTITION_SIZE := 3077536768

BOARD_CACHEIMAGE_PARTITION_SIZE := 157286400
BOARD_CACHEIMAGE_FILE_SYSTEM_TYPE := ext4

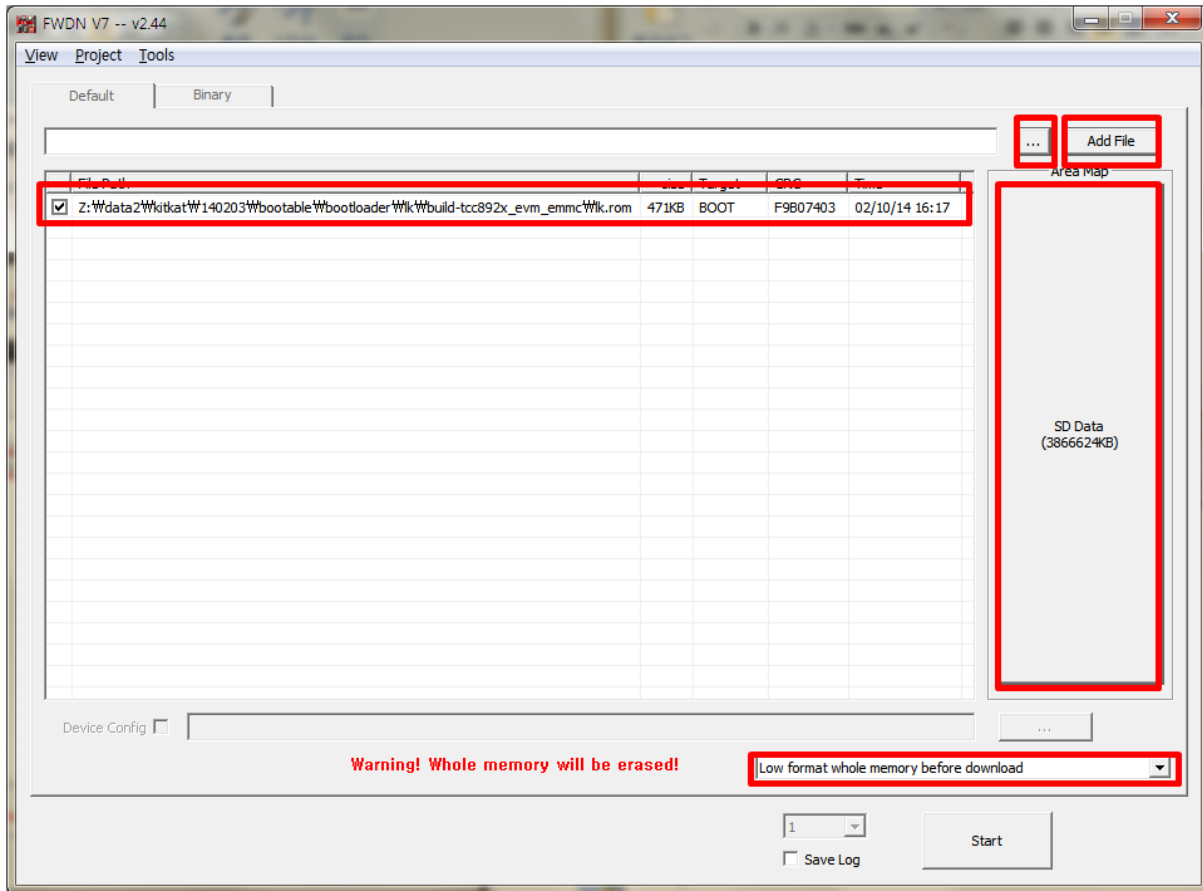
# Wi-Fi defines
BOARD_USES_REALTEK_WIFI := true
BOARD_USES_BROADCOM_WIFI := false

"device/telechips/tcc893x/BoardConfig.mk" [Modified] 147 lines --26%-- 39,0-1 25%
```

Step 3. If creating image is succeeded, press start button to download the image to the target board.

7.5.2 GPT Partition Layout – SD/MMC Boot Mode, eMMC Boot mode(Use Hidden Area)

*Step 1. Load bootloader image to FWDN, and then attach target devices to FWDN through usb boot mode. If the devices is attached to FWDN successfully, click **SD Data Button***



Step 2. Prepare download images.

Select number of partition, partition size, images, and and "Use GPT(only Primary)" as following figure. The partition order is same as partition layout described in the previous section. Finally, click **Create Image Button**.

Image Creation Dialog for Area 'SD Data'

Image File Path: SD Data.fat

Tab: Disk Image | LOGO Image | KEY STORE RO | UID

Number of Partition: 9 (highlighted in red box)

Use GPT (only Primary) (highlighted in green box)

Partition 1: Partition Size 15360 KB (1MB=1024), Partition Label boot, Image File Path boot.img

Partition 2: Partition Size 665600 KB (1MB=1024), Partition Label system, Image File Path system.img

Partition 3: Partition Size 153600 KB (1MB=1024), Partition Label cache, Image File Path You can optionally generate cache.img (highlighted in blue box)

Partition 4: Partition Size 15360 KB (1MB=1024), Partition Label recovery, Image File Path recovery.img

Partition 5: Partition Size 5120 KB (1MB=1024), Partition Label splash, Image File Path

Partition 6: Partition Size 4096 KB (1MB=1024), Partition Label kpanic, Image File Path

Partition 7: Partition Size 1024 KB (1MB=1024), Partition Label misc, Image File Path

Partition 8: Partition Size 1024 KB (1MB=1024), Partition Label tcc, Image File Path

Partition 9: Partition Size 3005423 KB (1MB=1024), Partition Label userdata, Image File Path You can optionally generate userdata.img (highlighted in blue box)

Create Image

Notice :

Green Box - In this configuration, GPT partition table is stored only at Primary GPT because bootloader and/or hidden data are stored at Secondary GPT table area.

Blue Box - You can see partition 3 and partition 9 in the blue box: user data partition and cache partitions. Those image paths are empty. From Android 4.2(JellyBean), if user data and cache partitions are not formatted as ext4 file system format, the system will be operated abnormally. For this reason, if the partitions in the blue box are not formatted as ext4 type, system will format those partitions to the ext4 file system at first boot time. Therefore, system will be rebooted again for initializing Android systems. So FWDN does not write any data to that partition if image file path is empty.

Otherwise, if you want to protect formatting file system during initial boot sequence, add following parameter depending on your storage capacity. This configuration generates 1)userdata.img and 2)cache.img in your \$OUT/ directory. Then choose the images in FWDN tool.

```
BOARD_FLASH_BLOCK_SIZE := 4096
BOARD_SYSTEMIMAGE_PARTITION_SIZE := 681574400 #650MB System
BOARD_USERDATAIMAGE_PARTITION_SIZE := 3077553152

BOARD_CACHEIMAGE_PARTITION_SIZE := 157286400
BOARD_CACHEIMAGE_FILE_SYSTEM_TYPE := ext4

# Wi-Fi defines
BOARD_USES_REALTEK_WIFI := true
BOARD_USES_BROADCOM_WIFI := false

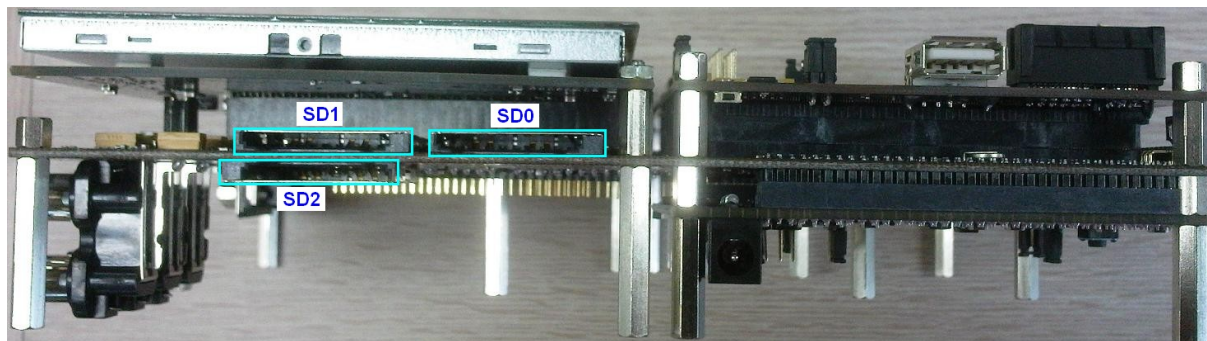
"device/telechips/tcc893x/BoardConfig.mk" 147 lines --26%-- 39,0-1 25%
```

Step 3. If creating image is succeeded, press start button to download the image to the target board.

7.6 SD Slot Usages

7.6.1 Location of the SD slot at each EVB board

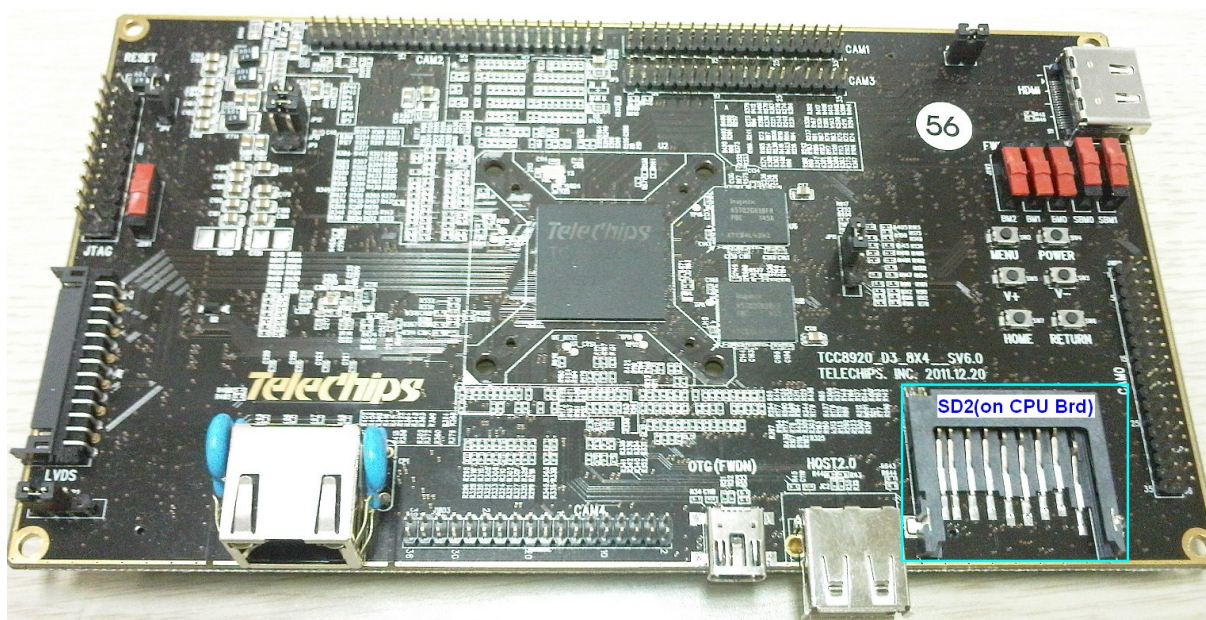
In the TCC892x EVM Board, SD2 slot shown in the figure below is not used.



[Figure 8.1 Telechips EVM Main Board – Left Side]

7.6.2 TCC8920 CPU Board

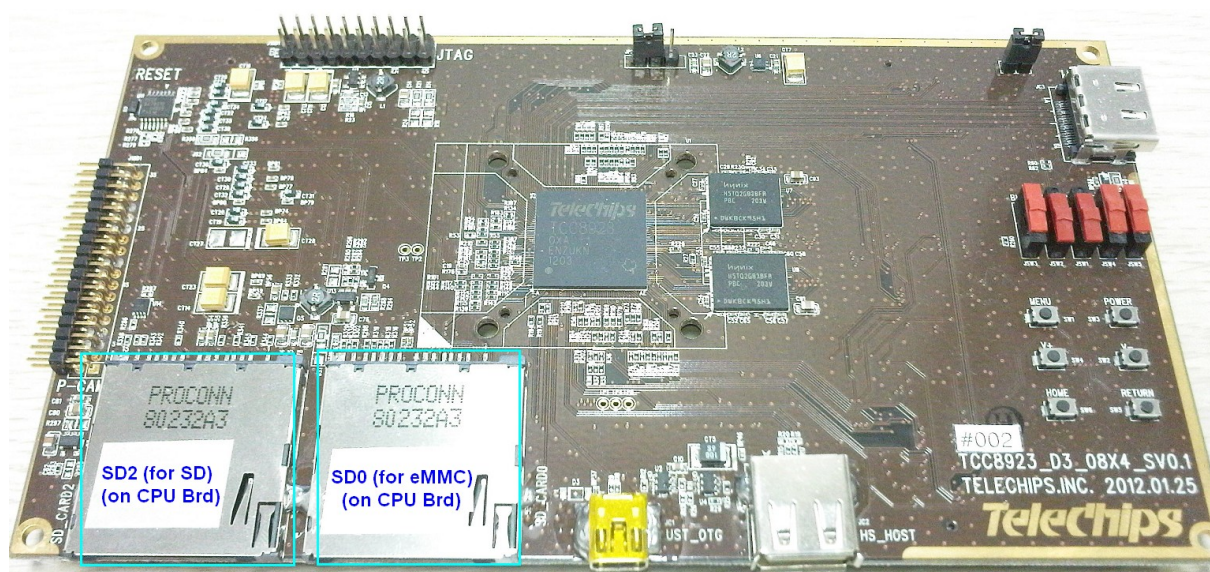
TCC8920 EVM	SD Slot	NAND Boot (Default Setting)	SD/MMC/eMMC boot (Optional)
Main Board	SD0	-	eMC
	SD1	WiFi	WiFi
	SD2	-	-
CPU Board	SD2	SD	SD



[Figure 8.2 TCC8920 D3 CPU Board]

7.6.3 TCC8923 D3 CPU Board

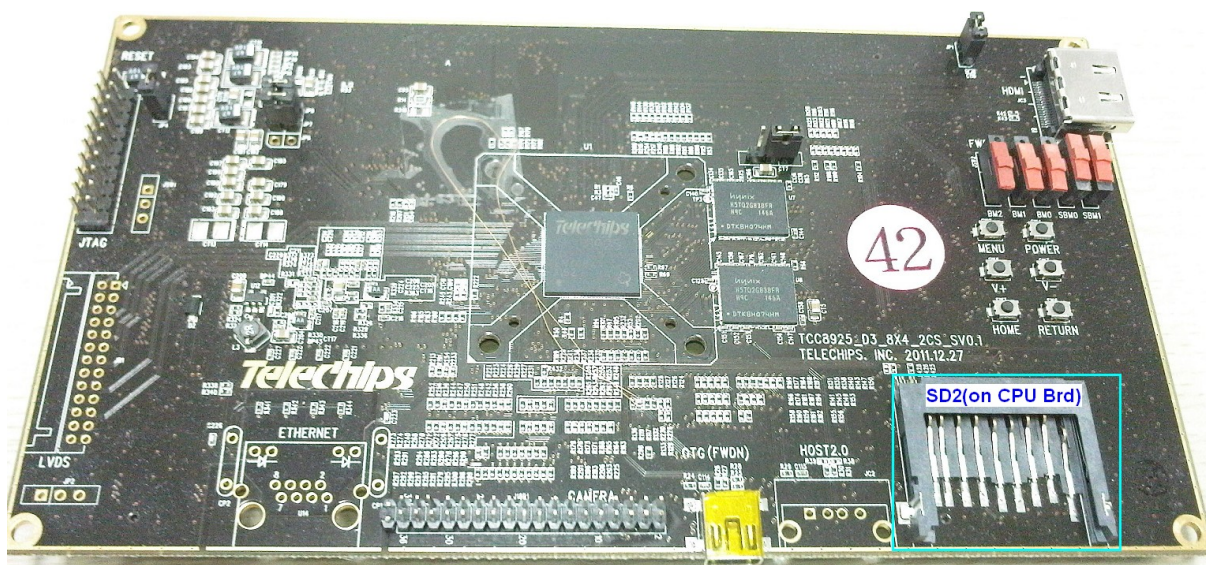
TCC8923 EVM	SD Slot	NAND Boot (Default Setting)	SD/MMC/eMMC boot (Optional)
Main Board	SD0	-	-
	SD1	WiFi	WiFi
	SD2	-	-
CPU Board	SD0	-	eMMC
	SD2	SD	SD



[Figure 8.3 TCC8923 D3 CPU Board]

7.6.4 TCC8925 D3 2CS CPU Board

TCC8925 EVM	SD Slot	NAND Boot (Default Setting)	SD/MMC/eMMC boot (Optional)
Main Board	SD0	-	eMMC
	SD1	-	-
	SD2	-	-
CPU Board	SD2	SD	SD



[Figure 8.4 TCC8925 D3 2CS CPU Board]

7.7 Appendix

FWDN v2.30 is changed for new TCC892X_AX or TCC892X Chipset. So, if you use TCC892X_XX, please set the TCC892x_XX Chipset like the image below.

Tools Menu -> etc tab you can see the check box for TCC892x XX Chipset

