

# TCC892X STB Android SDK Quick Start Guide

TCC892X-STB- Android 4.4.2(Kitkat-mr1.1)-V1.00E-Quick Start Guide

Mar 31, 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 is 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. cannot 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. Trademarks 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
2014-02-13	0.01	Initial Release
2014-03-31	1.00	Android 4.4.2(Kitkat-mr1.1) Official version release – v14.04_tcc-android-4.4.2

## TABLE OF CONTENTS

### Contents

Revision History .....	3
TABLE OF CONTENTS .....	4
Contents .....	4
Figures.....	4
1 Introduction .....	5
2 Download Android Kitkat SDK .....	5
3 Setup compile environment .....	6
4 Compile Bootloader .....	8
4.1 Select board configuration.....	8
4.1.1 Set HW_REW for TCC892x .....	8
4.1.2 Set HW_REW for TCC893x .....	8
4.2 Compile .....	9
4.3 TCC8925 HDMI Dongle T-Flash Board.....	9
4.4 TCC8925 HDMI Dongle eMMC Board .....	9
4.5 TCC8925S HDMI Dongle Board .....	10
4.6 TCC8935 HDMI Dongle T-Flash Board.....	10
4.7 TCC8935(S) HDMI Dongle eMMC Board .....	10
4.8 TCC8935S HDMI Dongle Board .....	10
5 Compile Linux Kernel.....	11
5.1 eMMC/T-Flash Board .....	11
5.2 Compile .....	11
6 Compile Android Framework .....	12
6.1 Compile Environment .....	12
6.1.1 TCC892x Board.....	12
6.1.2 TCC893x Board.....	12
6.1.3 TCC8925S HDMI Dongle Board .....	12
6.1.4 TCC8935S HDMI Dongle Board .....	13
6.2 Compile Frameworks .....	13
6.3 Choose Wi-Fi Module .....	13
7 Download Images .....	14
7.1 Change Boot Mode .....	14
7.1.1 YJ8925/30/33/35T Board.....	14
7.1.2 HDMI Dongle Board .....	14
7.2 Install VTC driver .....	14
7.3 Partition Layout.....	14
7.3.1 Introduction.....	14
7.3.2 GPT Partition Layout .....	14
7.3.3 MBR Partition Layout .....	15
7.4 Download images with FWDN V7 .....	16
8 Download Images by using fastboot.....	19
8.1 Enter fastboot mode .....	19
8.2 Install Android USB Driver .....	19
8.3 Download.....	20
9 Summary of Compile command .....	21
9.1 TCC892x Board.....	21
9.2 TCC893x Board.....	23

### Figures

Figure 1. FWDN program in Windows PC .....	16
Figure 2. Low format option in FWDN.....	16
Figure 3. GPT Partition information for FWDN .....	17
Figure 4. MBR Partition information for FWDN.....	18
Figure 5. Install Android USB driver .....	19

## 1 Introduction

This document provides guideline for users to start Android platform v4.4 (Kitkat) for TCC892x and TCC893x STB (Set-Top-Box) solution more quickly.

There are four types of target boards for STB solution.

- STBM
- YJ8925T, YJ8935T
- uPC
- HDMI Dongle

You must setup compile environment according to target board differently.

You must compile *Bootloader*, *Kernel* and *Frameworks* respectively. *Bootloader* and *Kernel* must be compiled prior to compile *Android Frameworks*.

## 2 Download Android Kitkat SDK

Download SDK from Telechips Android git server (`ssh://android.telechips.com/androidce/`). Please contact agent to get information about accessing Telechips Android git server. You must to use “-b kitkat” option to download Android kitkat SDK.

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

After download SDK, you can see many folders and files from working folder. This document assumes working folder as “mydroid”.

### 3 Setup compile environment

Before you compile, `TARGET_PRODUCT` must be setup for proper board configuration by executing “`build/envsetup.sh`” shell-script. To execute shell-script, there is space between dot (‘.’) and other (“`build/envsetup.sh`”).

```
$ cd ~/mydroid

$ . 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
```

Please execute “`lunch`” and then you can see board lists.

```
$ 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
```

If you want to compile TCC892x STB solution, please select 'full\_tcc8920st-eng' and want to compile TCC893x STB solution, please select 'full\_tcc8930st-eng'.

Please execute 'choosevariant' and select 'user' or 'userdebug' for mass production.

```
$ choosevariant
Variant choices are:
  1. user
  2. userdebug
  3. eng
Which would you like? [eng]
```

## 4 Compile Bootloader

### 4.1 Select board configuration

Before compile bootloader, you must select proper board configuration.

#### 4.1.1 Set HW\_REW for TCC892x

You can see below from 'bootable/bootloader/lk/target/tcc8920st\_evm/rules.mk'.

```
#-----
# Define board revision
# 0x6000 : STB      /TCC8920 /DDR3 512MB (16BIT) /None
# 0x6020 : STB      /TCC8920 /DDR3 512MB (32BIT) /None
# 0x6030 : STB      /TCC8920 /DDR3 1024MB(32BIT) /None
# 0x6040 : STB      /TCC8920 /DDR2 512MB (32BIT) /None
# 0x6050 : STB      /TCC8920 /DDR2 1024MB(32BIT) /None
# 0x7100 : YAOJIN   /TCC8925 /DDR3 512MB (16BIT) /None(1CS)
# 0x7110 : YAOJIN   /TCC8925 /DDR3 1024MB(16BIT) /None(2CS)
# 0x8100 : UPC      /TCC8925 /DDR3 512MB (16BIT) /None
# 0x8110 : UPC      /TCC8925 /DDR3 1024MB(16BIT) /None
# 0x9100 : DONGLE   /TCC8925 /DDR3 512MB (16BIT) /None
# 0x9101 : DONGLE   /TCC8925 /DDR3 512MB (16BIT) /'S' version
# 0x9110 : DONGLE   /TCC8925 /DDR3 1024MB(16BIT) /None
# 0x9111 : DONGLE   /TCC8925 /DDR3 1024MB (16BIT) /'S' version
# 0x9112 : DONGLE   /TCC8925 /DDR3 1024MB (16BIT) /eMMC (BROADCOM)
HW_REV=0x6030
#HW_REV=0x7100
#HW_REV=0x8110
#HW_REV=0x9100
#HW_REV=0x9112
```

You must select HW\_REV for proper board.

STBM	<a href="#">HW_REV=0x6030</a>
YJ8925T	<a href="#">HW_REV=0x7100</a>
uPC	<a href="#">HW_REV=0x8110</a>
HDMI Dongle	<a href="#">HW_REV=0x9100, 0x9112</a>

#### 4.1.2 Set HW\_REW for TCC893x

You can see below from 'bootable/bootloader/lk/target/tcc8930st\_evm/rules.mk'.

```
#-----
# Define board revision
# 0x6230 : STBM      /TCC8930 /DDR3 1024MB(32BIT) /None
# 0x7230 : YAOJIN   /TCC8930 /DDR3 1024MB(32BIT) /Ex0.1
# 0x7231 : YAOJIN   /TCC8930 /DDR3 1024MB(32BIT) /Ex0.2
# 0x7430 : YAOJIN   /TCC8933 /DDR3 1024MB(32BIT) /None
# 0x7300 : YAOJIN   /TCC8935 /DDR3 512MB (16BIT) /None
# 0x7310 : YAOJIN   /TCC8935 /DDR3 1024MB(16BIT) /None
# 0x8310 : UPC      /TCC8935 /DDR3 1024MB(16BIT) /None
# 0x9300 : DONGLE   /TCC8935 /DDR3 512MB (16BIT) /None
# 0x9310 : DONGLE   /TCC8935 /DDR3 1024MB(16BIT) /None
# 0x9301 : DONGLE   /TCC8935 /DDR3 512MB (16BIT) /'S' version
```



```
# 0x9311 : DONGLE /TCC8935 /DDR3 1024MB(16BIT) /'S' version
# 0x9312 : DONGLE /TCC8935 /DDR3 1024MB(16BIT) /eMMC (BROADCOM)
# 0x9313 : DONGLE /TCC8935 /DDR3 1024MB(16BIT) /'S' version + eMMC
HW_REV=0x6230
#HW_REV=0x7230
#HW_REV=0x7231
#HW_REV=0x7430
#HW_REV=0x7300
#HW_REV=0x7310
#HW_REV=0x8310
#HW_REV=0x9300
#HW_REV=0x9310
#HW_REV=0x9312
#HW_REV=0x9313
```

You must select HW\_REV for proper board.

STBM	<a href="#">HW_REV=0x6230</a>
YJ893xT	<a href="#">HW_REV=0x7230, 0x7231, 0x7430, 0x7300, 0x7310</a>
uPC	<a href="#">HW_REV=0x8310</a>
HDMI Dongle	<a href="#">HW_REV=0x9300, 0x9310, 0x9312, 0x9313</a>

## 4.2 Compile

Move in “bootable/bootloader/lk” folder. Execute “make tcc8920st\_evm” for TCC892x STB solution and execute “make tcc8930st\_evm” for TCC893x STB solution. If you want to use eMMC(T-Flash/SD) boot, please add “\_emmc” suffix.

Board	Command	Output folder which include lk.rom
TCC892x NAND	<a href="#">make tcc8920st_evm</a>	build-tcc8920st_evm
TCC892x eMMC (T-Flash)	<a href="#">make tcc8920st_evm emmc</a>	build-tcc8920st_evm emmc
TCC893x NAND	<a href="#">make tcc8930st_evm</a>	build-tcc8930st_evm
TCC893x eMMC (T-Flash)	<a href="#">make tcc8930st_evm emmc</a>	build-tcc8930st_evm emmc

If you want to clean the compiled result, you can execute the command like “make tcc8920st\_evm clean” or “make tcc8930st\_evm clean”.

## 4.3 TCC8925 HDMI Dongle T-Flash Board

```
# 0x9100 : DONGLE /TCC8925 /DDR3 512MB (16BIT) /None
HW\_REV=0x9100

# 0x9110 : DONGLE /TCC8925 /DDR3 1024MB(16BIT) /None
HW\_REV=0x9110

$ make tcc8920st\_evm\_emmc
```

## 4.4 TCC8925 HDMI Dongle eMMC Board

```
# 0x9112 : DONGLE /TCC8925 /DDR3 1024MB (16BIT) /eMMC (BROADCOM)
HW\_REV=0x9112

$ make tcc8920st\_evm\_emmc
```

## 4.5 TCC8925S HDMI Dongle Board

```
# 0x9101 : DONGLE /TCC8925 /DDR3 512MB (16BIT) /'S' version
HW_REV=0x9101

# 0x9111 : DONGLE /TCC8925 /DDR3 1024MB (16BIT) /'S' version
HW_REV=0x9111

$ make tcc8920st_evm
```

## 4.6 TCC8935 HDMI Dongle T-Flash Board

```
# 0x9300 : DONGLE /TCC8935 /DDR3 512MB (16BIT) /None
HW_REV=0x9300

# 0x9310 : DONGLE /TCC8935 /DDR3 1024MB(16BIT) /None
HW_REV=0x9310

$ make tcc8930st_evm_emmc
```

## 4.7 TCC8935(S) HDMI Dongle eMMC Board

```
# 0x9312 : DONGLE /TCC8935 /DDR3 1024MB(16BIT) /eMMC (BROADCOM)
HW_REV=0x9312

# 0x9313 : DONGLE /TCC8935 /DDR3 1024MB(16BIT) /'S' version + eMMC
HW_REV=0x9313

$ make tcc8930st_evm_emmc
```

## 4.8 TCC8935S HDMI Dongle Board

```
# 0x9301 : DONGLE /TCC8935 /DDR3 512MB (16BIT) /'S' version
HW_REV=0x9301

# 0x9311 : DONGLE /TCC8935 /DDR3 1024MB(16BIT) /'S' version
HW_REV=0x9311

$ make tcc8930st_evm
```

## 5 Compile Linux Kernel

*Kernel must be compiled prior to compile Android Frameworks and lunch menu must be selected before compiling Kernel because lunch menu affect Linux Kernel Configuration.*

You must select *defconfig* according to proper board.

TCC892x Board	command
TCC8920 STBM	<code>make tcc8920st_defconfig</code>
TCC8925 YJ8925T	<code>make tcc8925st_yj8925t_defconfig</code>
TCC8925 uPC	<code>make tcc8925st_upc_defconfig</code>
TCC8925 HDMI Dongle	<code>make tcc8925st_dongle_defconfig</code>
TCC8925S HDMI Dongle	<code>make dongle_tcc8925s_defconfig</code>

TCC893x Board	command
TCC8930 STBM	<code>make tcc8930st_defconfig</code>
TCC8930 YJ8930T	<code>make tcc8930st_yj8930t_defconfig</code>
TCC8933 YJ8933T	<code>make tcc8933st_yj8933t_defconfig</code>
TCC8935 YJ8935T	<code>make tcc8935st_yj8935t_defconfig</code>
TCC8935 uPC	<code>make tcc8935st_upc_defconfig</code>
TCC8935 HDMI Dongle	<code>make tcc8935st_dongle_defconfig</code>
TCC8935S HDMI Dongle	<code>make dongle_tcc8935s_defconfig</code>

### 5.1 eMMC/T-Flash Board

If target board uses eMMC boot, you should change kernel configuration.

```
$ make menuconfig
Device Drivers --> MMC/SD/SDIO card support --> [*] Support an eMMC
```

You should change kernel configuration according to WIFI chipset that you use. (Please refer chapter6.1)

### 5.2 Compile

To compile Linux Kernel, just execute “make”.

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

## 6 Compile Android Framework

### 6.1 Compile Environment

Please refer chapter 4.1 to setup `TARGET_PRODUCT`. Except STBM board, you need to change some configurations.

#### 6.1.1 TCC892x Board

Define the name of target board in “device/telechips/tcc8920st/full\_tcc8920st.mk”.

```
# Define the name of target board
TARGET_BOARD_8920_EV := true
#TARGET_BOARD_8925_YJ8925T := true
#TARGET_BOARD_8925_UPC := true
#TARGET_BOARD_8925_DONGLE := true
```

You should select the below define according to target board.

Target Board	Define
TCC8920 STBM Board	<a href="#">TARGET_BOARD_8920_EV</a>
TCC8925 YJ8925T Board	<a href="#">TARGET_BOARD_8925_YJ8925T</a>
TCC8925 uPC Board	<a href="#">TARGET_BOARD_8925_UPC</a>
TCC8925 HDMI Dongle Board	<a href="#">TARGET_BOARD_8925_DONGLE</a>

#### 6.1.2 TCC893x Board

Define the name of target board in “device/telechips/tcc8930st/full\_tcc8930st.mk”.

```
# Define the name of target board
TARGET_BOARD_8930_EV := true
#TARGET_BOARD_8930_YJ8930T := true
#TARGET_BOARD_8933_YJ8933T := true
#TARGET_BOARD_8935_YJ8935T := true
#TARGET_BOARD_8935_UPC := true
#TARGET_BOARD_8935_DONGLE := true
```

You should select the below define according to target board.

Target Board	Define
TCC8930 STBM Board	<a href="#">TARGET_BOARD_8930_EV</a>
TCC8930 YJ8930T Board	<a href="#">TARGET_BOARD_8930_YJ8930T</a>
TCC8933 YJ8933T Board	<a href="#">TARGET_BOARD_8933_YJ8933T</a>
TCC8935 YJ8935T Board	<a href="#">TARGET_BOARD_8935_YJ8935T</a>
TCC8935 uPC Board	<a href="#">TARGET_BOARD_8935_UPC</a>
TCC8935 HDMI Dongle Board	<a href="#">TARGET_BOARD_8935_DONGLE</a>

#### 6.1.3 TCC8925S HDMI Dongle Board

Define chipset feature in “device/telechips/tcc8920st/device.mk”.

```
#TARGET_BOARD_SOC := tcc892x
```

```
TARGET_BOARD_SOC := tcc892xS
```

#### 6.1.4 TCC8935S HDMI Dongle Board

Define chipset feature in “device/telechips/tcc8930st/device.mk”.

```
#TARGET_BOARD_SOC := tcc893x  
TARGET_BOARD_SOC := tcc893xS
```

### 6.2 Compile Frameworks

Just execute “make” command. You can build Android Frameworks.

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

### 6.3 Choose Wi-Fi Module

You can select Wi-Fi module by using the below command after finishing compiling *kernel* and *frameworks*.

```
$ choosewifi "vendor" "chipset"  
  
Ex)  
$ choosewifi atheros  
$ choosewifi realtek rt8188cu  
$ choosewifi realtek rt8189es  
$ choosewifi realtek rt8188eus  
$ choosewifi broadcom bcm4330  
$ choosewifi broadcom bcm4334  
$ choosewifi broadcom bcm4335
```

## 7 Download Images

You must use FWDN\_V7 program to download images to target board. FWDN\_V7 can be found in “vendor/telechips/tools/FWDN” folder. Please refer the description below to know how to download images by using FWDN\_V7. **The version of FWDN\_V7 must be higher than v2.44.**

### 7.1 Change Boot Mode

You must change boot mode of target board to “Boot from USB device - Firmware Download Mode” to download firmware. It is different to change boot mode according to target board. You can enter “Firmware Download Mode”, if you power off/on or reboot target board after changing boot mode.

#### 7.1.1 YJ8925/30/33/35T Board

You can see “USB” and “NAND” from JSW1 and connect the jumper to “USB” direction for changing boot mode to “Firmware Download Mode” (TCC8925/35T) or press push button and don’t release it before power on (TCC8933/35T).

#### 7.1.2 HDMI Dongle Board

Before power on HDMI Dongle, press the button and don’t release it. Then HDMI Dongle will enter “Firmware Download Mode”.

### 7.2 Install VTC driver

You need to install VTC driver to recognize target board on your PC with FWDN\_V7 program. Please execute “VTC Driver Installer....EXE” file in “vendor/telechips/tools/FWDN/vtcdrv” folder. **The version of VTC driver must be higher than v5.0.0.5 for TCC892x/3x STB solution.**

### 7.3 Partition Layout

#### 7.3.1 Introduction

Default partition layout scheme is GPT. MBR partition layout scheme can be also supported and it is used only for user firmware upgrade from old version (JellyBean) which use MBR scheme. So if you want to upgrade firmware from JellyBean to Kitkat, you should use MBR scheme. It is not compatible between GPT and MBR. You should understand how to configure GPT/MBR partition layout for Android System, before you download images

#### 7.3.2 GPT Partition Layout

Please refer to the following table to check default GPT partition layout.

Area	Name	Purpose	File System	Required
Boot	Boot Area	Kernel / RamDisk	RAW	Mandatory
System	Android System	Android System Area	EXT4	Mandatory
Cache	Android Cache	Android Cache Area	EXT4	Mandatory
Recovery	Android Recovery	Recovery Mode Boot Area Recovery Mode Kernel / Ram Disk	RAW	Mandatory
Kpanic	Kpanic	Kernel Panic Log	RAW	Mandatory
Splash	Splash	Boot Screen Image	RAW	Mandatory
Misc	Miscellaneous	Firmware Update Bootloader Flag	RAW	Mandatory
TCC	Telechips Only	Set-top Flag write	RAW	Optional
UserData	Android UserData	Android User Data Application / Database	EXT4	Mandatory

Please refer to the following table to set the size of each partition. The size of some partitions can't be changeable.

Area	Size	Partition	FileSystem
Boot	15MB	ndda1 / mmcblk0p1	RAW
System	650MB	ndda2 / mmcblk0p2	EXT4
Cache	150MB	ndda3 / mmcblk0p3	EXT4
Recovery	15MB	ndda4 / mmcblk0p4	RAW
Kpanic	5MB	ndda5 / mmcblk0p5	RAW
Splash	4MB	ndda6 / mmcblk0p6	RAW
Misc	1MB	ndda7 / mmcblk0p7	RAW
TCC	1MB	ndda8 / mmcblk0p8	RAW
UserData	Available Size	ndda9 / mmcblk0p9	EXT4

The additional partitions are not need for Android system. But you can add them for special purpose and source codes should be also changed to do it.

### 7.3.3 MBR Partition Layout

Please refer to the following table to check default MBR partition layout.

Area	Name	Purpose	FileSystem	Required
Boot	Boot Area	Kemel / Ram Disk	RAW	Mandatory
System	Android System	Android System Area	EXT4	Mandatory
UserData	Android UserData	Android User Data Application / Database	EXT4	Mandatory
Cache	Android Cache	Android Cache Area	EXT4	Mandatory
Recovery	Android Recovery	Recovery Mode Boot Area Recovery Mode Kemel / Ram Disk	RAW	Mandatory
Kpanic	Kpanic	Kemel Panic Log	RAW	Mandatory
Splash	Splash	Boot Screen Image	RAW	Mandatory
Misc	Miscellaneous	Firmware Update Bootloader Flag	RAW	Mandatory
TCC	Telechips Only	Set-top Flash write	RAW	Optional

Please refer to the following table to set the size of each partition. The size of some partitions can't be changeable.

Area	Size	Partition	FileSystem
Boot	15MB	Ndda1	RAW
System	650MB	Ndda2	EXT4
UserData	Available Size	Ndda3	EXT4
Extended	Extended	Ndda4	Extended
Cache	150MB	Ndda5	EXT4
Recovery	15MB	Ndda6	RAW
Kpanic	5MB	Ndda7	RAW
Splash	4MB	Ndda8	RAW
Misc	1MB	Ndda9	RAW
TCC	1MB	ndda10	RAW

Linux System can only support 4 primary partitions but we need more partitions. So we set an

extended partition (the forth partition) and it includes other logical partitions (6 ea). The additional partitions are not need for Android system. But you can add them for special purpose and source codes should be also changed to do it.

## 7.4 Download images with FWDN V7

Below steps can be used for not only NAND but also eMMC (T-Flash) boot.

**[Step1]** Load a bootloader image on FWDN and then connect target device (USB boot mode) with USB port of PC. If it succeeds in connection with FWDN, click “**NAND Data**” Button

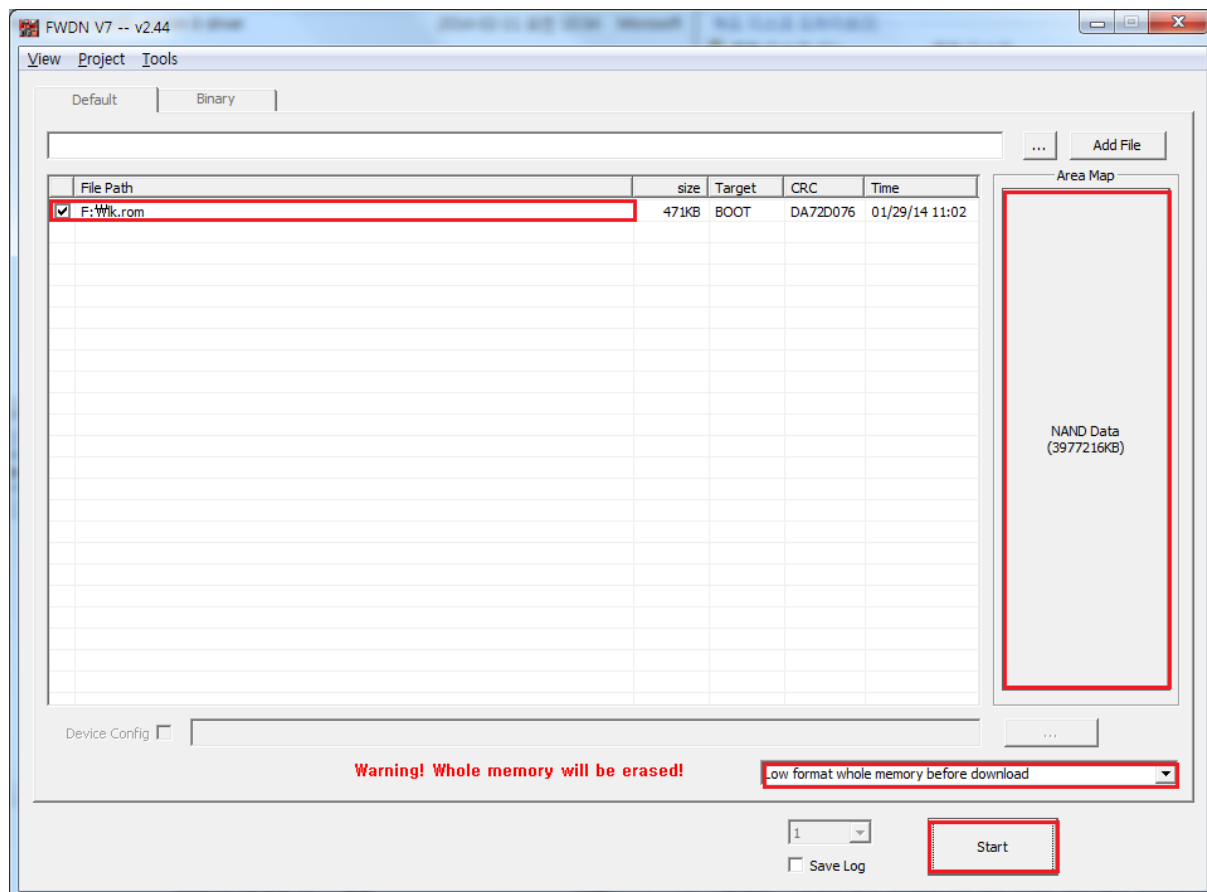


Figure 1. FWDN program in Windows PC

If you want to do low-level format before downloading, select “Low format” option.

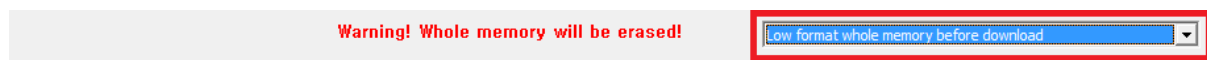


Figure 2. Low format option in FWDN

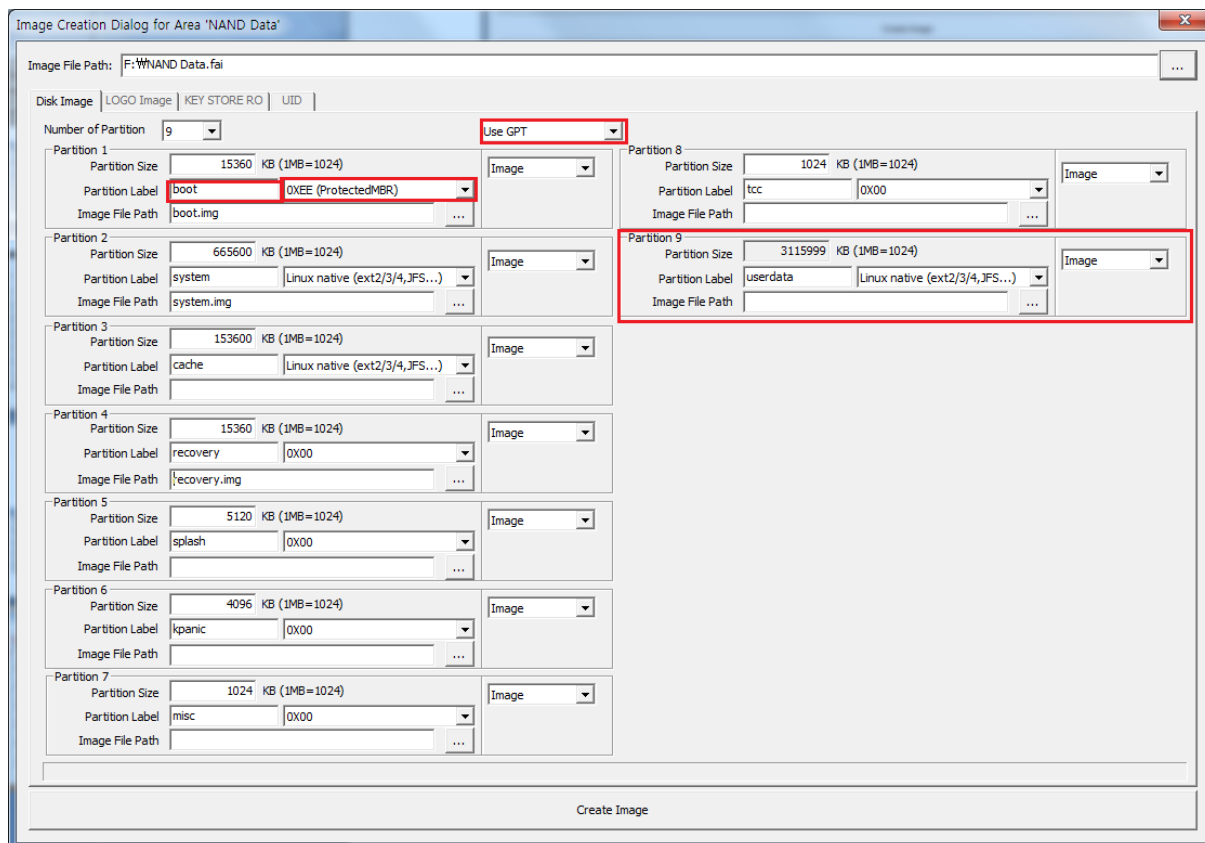


**[Step2]** Prepare the following images to download.

boot.img  
system.img  
recovery.img

The order of partitions should be same with the order of partition layout (section 7.3). Click **“Create Image”** button.

**In case of GPT Partition Layout,**



**Figure 3. GPT Partition information for FWDN**

**Notice:** To use GPT partition layout, you should select “Use GPT” like above image. You can see “Partition Label” is filled with the name of each partition. This name is very important because it is used to mount file system. So you should write correct name of partition to “Partition Label”.

The difference of MBR and GPT partition is the number of partitions. MBR has extended partition but GPT don’t have it.

**In case of MBR Partition Layout,**

Image Creation Dialog for Area 'NAND Data'

Image File Path: F:\NAND Data.fai

Disk Image | LOGO Image | KEY STORE RO | UID

Number of Partition: 9 (Use MBR)

Partition	Partition Size	Partition Label	Image File Path
Partition 1	15360 KB (1MB=1024)	0X00	boot.img
Partition 2	665600 KB (1MB=1024)	Linux native (ext2/3/4,JFS...)	system.img
Partition 3	3115920 KB (1MB=1024)	Linux native (ext2/3/4,JFS...)	
Partition 4	153600 KB (1MB=1024)	Linux native (ext2/3/4,JFS...)	
Partition 5	15360 KB (1MB=1024)	0X00	recovery.img
Partition 6	5120 KB (1MB=1024)	0X00	
Partition 7	4096 KB (1MB=1024)	0X00	
Partition 8	1024 KB (1MB=1024)	0X00	
Partition 9	1024 KB (1MB=1024)	0X00	

Create Image

**Figure 4. MBR Partition information for FWDN**

**Notice:** To use MBR partition layout, you should select “Use MBR” like above image. You can see that “Image File Path” of partition 3 (usderdata) and 4 (cache) is empty. From Android 4.1 (JellyBean), system will format these partitions and reboot automatically if they were not formatted during booting-up. So you don’t need to write any file path for these partitions.

Partition 3 (usderdata) is used for the shared storage through MTP. So the size of this partition should be the remaining size of NAND. If you set size of partition 3 to 1024, the size of partition 9 is automatically calculated as the remaining size of NAND. Then you can switch the size of partition 3 and 9 like above image.

**[Step3]** If it succeeds in creating an image, Click **“Start”** button to start downloading images to target board

## 8 Download Images by using fastboot

*Kernel* and *Framework* image can be downloaded with “fastboot” command. Generally “fastboot” is used for development.

### 8.1 Enter fastboot mode

If *bootloader* image was written to target board, you can enter “fastboot mode” by inputting ‘f’ character of keyboard on UART console window during booting-up. You can see messages below on console window if you entered “fastboot mode” normally.

```
[2720] fastboot_init()  
[2720] udc_start()
```

### 8.2 Install Android USB Driver

Connect target board with PC by using USB cable after entering “fastboot mode”. If target board is firstly connected with PC, you should install “Android USB driver” to use “fastboot” command.

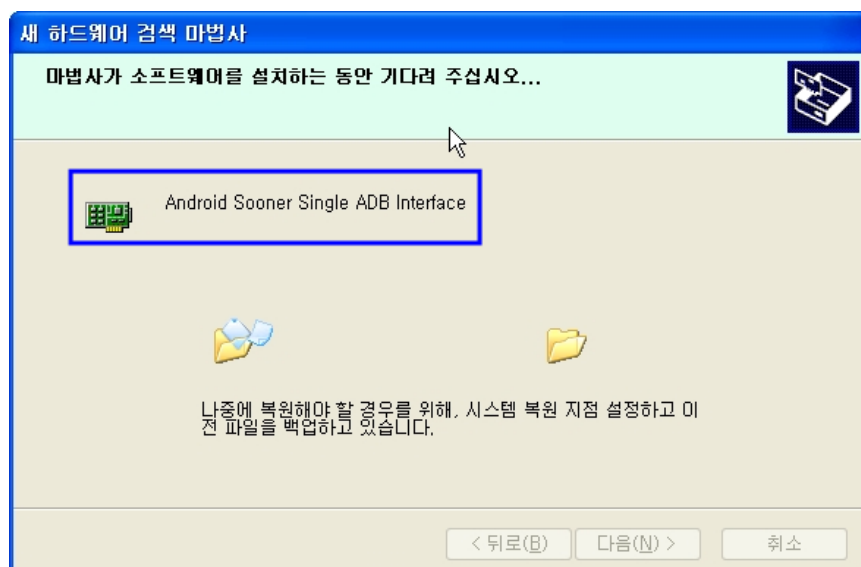


Figure 5. Install Android USB driver

If target board is normally connected with PC, you can see messages below on console window

```
[44040] USB Reset  
[44050] Connect Done[44050] : High Speed  
[44080] USB Reset  
[44090] Connect Done[44090] : High Speed  
[44180] fastboot: processing commands
```

To use “fastboot” command, you must have “fastboot.exe” which can be obtained from Android SDK. Android SDK can be downloaded from Google’s site. If you can use “adb.exe”, you can also use “fastboot.exe”. Please search and refer to various information about “adb” and “fastboot” from internet sites.

### 8.3 Download

Execute “cmd.exe” in your PC and move to folder which includes “fastboot.exe” and “adb.exe”.

```

C:\> 명령 프롬프트
D:\Android\Wtool>dir
D 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 5C96-7804

D:\Android\Wtool 디렉터리

2012-01-27 오후 06:00 <DIR>      .
2012-01-27 오후 06:00 <DIR>      ..
2009-06-30 오후 05:47          2,258,137 adb.exe
2009-06-30 오후 05:47          994,250 fastboot.exe
                2개 파일          3,252,387 바이트
                2개 디렉터리 39,548,002,304 바이트 남음

D:\Android\Wtool>
  
```

If you execute “fastboot devices” command, you can check list all connected devices. If there is no device, no messages will be shown.

```

D:\Android\Wtool>fastboot devices
?
fastboot

D:\Android\Wtool>
  
```

If images exist in “z:\android\out\target\product\tcc8930st”, you can set the path of images by using ANDROID\_PRODUCT\_OUT parameter like below in advance. So you don’t need to write full path when you use “fastboot” commands.

```

D:\Android\Wtool>set ANDROID_PRODUCT_OUT=Z:\android\out\target\product\tcc8920st
D:\Android\Wtool>
  
```

You can use the below commands to download images.

Image	fastboot command
boot.img	<code>fastboot flash boot</code>
recovery.img	<code>fastboot flash recovery</code>
system.img	<code>fastboot flash system</code>

## 9 Summary of Compile command

### 9.1 TCC892x Board

Board	Build Item	Command
TCC8925 HDMI Dongle	Compile Environment	<pre>\$ . build/envsetup.sh \$ lunch --&gt; Select "17. full_tcc8920st-eng"</pre>
	Bootloader	<pre>\$ cd bootable/bootloader/lk \$ vi target/tcc8920st_evm/rules.mk --&gt; HW_REV=0x9100 \$ make tcc8920st_evm</pre>
	Kernel	<pre>\$ cd kernel \$ make tcc8925st_dongle_defconfig \$ make</pre>
	Frameworks	<pre>\$ vi device/telechips/tcc8920st/full_tcc8920st.mk --&gt; TARGET_BOARD_8925_DONGLE := true \$ make</pre>
TCC8925 uPC	Compile Environment	<pre>\$ . build/envsetup.sh \$ lunch --&gt; Select "17. full_tcc8920st-eng"</pre>
	Bootloader	<pre>\$ cd bootable/bootloader/lk \$ vi target/tcc8920st_evm/rules.mk --&gt; HW_REV=0x8100 \$ make tcc8920st_evm</pre>
	Kernel	<pre>\$ cd kernel \$ make tcc8925st_upc_defconfig \$ make</pre>
	Frameworks	<pre>\$ vi device/telechips/tcc8920st/full_tcc8920st.mk --&gt; TARGET_BOARD_8925_UPC := true \$ make</pre>
TCC8925 YJ8925T	Compile Environment	<pre>\$ . build/envsetup.sh \$ lunch --&gt; Select "17. full_tcc8920st-eng"</pre>
	Bootloader	<pre>\$ cd bootable/bootloader/lk \$ vi target/tcc8920st_evm/rules.mk --&gt; HW_REV=0x7100 \$ make tcc8920st_evm</pre>
	Kernel	<pre>\$ cd kernel \$ make tcc8925st_yj8925t_defconfig \$ make</pre>
	Frameworks	<pre>\$ vi device/telechips/tcc8920st/full_tcc8920st.mk --&gt; TARGET_BOARD_8925_YJ8925T := true \$ make</pre>

Board	Build Item	Command
TCC8925S HDMI Dongle	Compile Environment	<pre>\$ . build/envsetup.sh \$ lunch --&gt; Select "17. full_tcc8920st-eng"</pre>
	Bootloader	<pre>\$ cd bootable/bootloader/lk \$ vi target/tcc8920st_evm/rules.mk --&gt; HW_REV=0x9101 \$ make tcc8920st_evm</pre>
	Kernel	<pre>\$ cd kernel \$ make dongle_tcc8925s_defconfig \$ make</pre>
	Frameworks	<pre>\$ vi device/telechips/tcc8920st/full_tcc8920st.mk --&gt; TARGET_BOARD_8925_DONGLE := true \$ vi device/telechips/tcc8920st/device.mk --&gt; TARGET_BOARD_SOC := tcc892xs \$ make</pre>
TCC8920 STBM	Compile Environment	<pre>\$ . build/envsetup.sh \$ lunch --&gt; Select "17. full_tcc8920st-eng"</pre>
	Bootloader	<pre>\$ cd bootable/bootloader/lk \$ vi target/tcc8920st_evm/rules.mk --&gt; HW_REV=0x6030 \$ make tcc8920st_evm</pre>
	Kernel	<pre>\$ cd kernel \$ make tcc8920st_defconfig \$ make</pre>
	Frameworks	<pre>\$ vi device/telechips/tcc8920st/full_tcc8920st.mk --&gt; TARGET_BOARD_8920_EV := true \$ make</pre>

## 9.2 TCC893x Board

Board	Build Item	Command
TCC8935 HDMI Dongle  (512MB)	Compile Environment	\$ . build/envsetup.sh \$ lunch --> Select "19. full_tcc8930st-eng"
	Bootloader	\$ cd bootable/bootloader/lk \$ vi target/tcc8930st_evm/rules.mk --> HW_REV=0x9300 \$ make tcc8930st_evm
	Kernel	\$ cd kernel \$ make tcc8935st_dongle_defconfig \$ make menuconfig System Type -> DRAM Memory Size -> 512MB System Type -> DDR settings -> Hynix DDR3_H5TQ2G83BFR_PBC \$ make
	Frameworks	\$ vi device/telechips/tcc8930st/full_tcc8930st.mk --> TARGET_BOARD_8935_DONGLE := true \$ make
TCC8935 HDMI Dongle  (1024MB)	Compile Environment	\$ . build/envsetup.sh \$ lunch --> Select "19. full_tcc8930st-eng"
	Bootloader	\$ cd bootable/bootloader/lk \$ vi target/tcc8930st_evm/rules.mk --> HW_REV=0x9310 \$ make tcc8930st_evm
	Kernel	\$ cd kernel \$ make tcc8935st_dongle_defconfig \$ make menuconfig System Type -> DRAM Memory Size -> 1024MB System Type -> DDR settings -> Hynix DDR3_H5TQ4G83AFR_PBC \$ make
	Frameworks	\$ vi device/telechips/tcc8930st/full_tcc8930st.mk --> TARGET_BOARD_8935_DONGLE := true \$ make
TCC8935S HDMI Dongle eMMC  (1024MB)	Compile Environment	\$ . build/envsetup.sh \$ lunch --> Select "19. full_tcc8930st-eng"
	Bootloader	\$ cd bootable/bootloader/lk \$ vi target/tcc8930st_evm/rules.mk --> HW_REV=0x9312 \$ make tcc8930st_evm_emmc
	Kernel	\$ cd kernel \$ make tcc8935st_dongle_defconfig \$ make menuconfig System Type -> DRAM Memory Size -> 1024MB System Type -> DDR settings -> Hynix DDR3_H5TQ4G83AFR_PBC Device Drivers -> MMC/SD/SDIO card support -> [*] Support an eMMC \$ make
	Frameworks	\$ vi device/telechips/tcc8930st/full_tcc8930st.mk --> TARGET_BOARD_8935_DONGLE := true \$ make

Board	Build Item	Command
TCC8935S HDMI Dongle	Compile Environment	<pre>\$ . build/envsetup.sh \$ lunch --&gt; Select "19. full_tcc8930st-eng"</pre>
	Bootloader	<pre>\$ cd bootable/bootloader/lk \$ vi target/tcc8930st_evm/rules.mk --&gt; HW_REV=0x9311 \$ make tcc8930st_evm</pre>
	Kernel	<pre>\$ cd kernel \$ make dongle_tcc8935s_defconfig \$ make</pre>
	Frameworks	<pre>\$ vi device/telechips/tcc8930st/full_tcc8930st.mk --&gt; TARGET_BOARD_8935_DONGLE := true \$ vi device/telechips/tcc8930st/device.mk --&gt; TARGET_BOARD_SOC := tcc893xs \$ make</pre>
TCC8935S HDMI Dongle eMMC	Compile Environment	<pre>\$ . build/envsetup.sh \$ lunch --&gt; Select "19. full_tcc8930st-eng"</pre>
	Bootloader	<pre>\$ cd bootable/bootloader/lk \$ vi target/tcc8930st_evm/rules.mk --&gt; HW_REV=0x9313 \$ make tcc8930st_evm_emmc</pre>
	Kernel	<pre>\$ cd kernel \$ make dongle_tcc8935s_defconfig \$ make menuconfig Device Drivers -&gt; MMC/SD/SDIO card support -&gt; [*] Support an eMMC \$ make</pre>
	Frameworks	<pre>\$ vi device/telechips/tcc8930st/full_tcc8930st.mk --&gt; TARGET_BOARD_8935_DONGLE := true \$ vi device/telechips/tcc8930st/device.mk --&gt; TARGET_BOARD_SOC := tcc893xs \$ make</pre>
TCC8935 uPC	Compile Environment	<pre>\$ . build/envsetup.sh \$ lunch --&gt; Select "19. full_tcc8930st-eng"</pre>
	Bootloader	<pre>\$ cd bootable/bootloader/lk \$ vi target/tcc8930st_evm/rules.mk --&gt; HW_REV=0x8310 \$ make tcc8930st_evm</pre>
	Kernel	<pre>\$ cd kernel \$ make tcc8935st_upc_defconfig \$ make</pre>
	Frameworks	<pre>\$ vi device/telechips/tcc8930st/full_tcc8930st.mk --&gt; TARGET_BOARD_8935_UPC := true \$ make</pre>



Board	Build Item	Command
TCC8935 YJ8935T	Compile Environment	<pre>\$ . build/envsetup.sh \$ lunch --&gt; Select "19. full_tcc8930st-eng"</pre>
	Bootloader	<pre>\$ cd bootable/bootloader/lk \$ vi target/tcc8930st_evm/rules.mk --&gt; HW_REV=0x7310 \$ make tcc8930st_evm</pre>
	Kernel	<pre>\$ cd kernel \$ make tcc8935st_yj8935t_defconfig \$ make</pre>
	Frameworks	<pre>\$ vi device/telechips/tcc8930st/full_tcc8930st.mk --&gt; TARGET_BOARD_8935_YJ8935T := true \$ make</pre>
TCC8933 YJ8933T	Compile Environment	<pre>\$ . build/envsetup.sh \$ lunch --&gt; Select "19. full_tcc8930st-eng"</pre>
	Bootloader	<pre>\$ cd bootable/bootloader/lk \$ vi target/tcc8930st_evm/rules.mk --&gt; HW_REV=0x7430 \$ make tcc8930st_evm</pre>
	Kernel	<pre>\$ cd kernel \$ make tcc8933st_yj8933t_defconfig \$ make</pre>
	Frameworks	<pre>\$ vi device/telechips/tcc8930st/full_tcc8930st.mk --&gt; TARGET_BOARD_8933_YJ8933T := true \$ make</pre>
TCC8930 YJ8930T	Compile Environment	<pre>\$ . build/envsetup.sh \$ lunch --&gt; Select "19. full_tcc8930st-eng"</pre>
	Bootloader	<pre>\$ cd bootable/bootloader/lk \$ vi target/tcc8930st_evm/rules.mk --&gt; HW_REV=0x7231 \$ make tcc8930st_evm</pre>
	Kernel	<pre>\$ cd kernel \$ make tcc8930st_yj8930t_defconfig \$ make</pre>
	Frameworks	<pre>\$ vi device/telechips/tcc8930st/full_tcc8930st.mk --&gt; TARGET_BOARD_8930_YJ8930T := true \$ make</pre>

Board	Build Item	Command
TCC8930 STBM	Compile Environment	<pre>\$ . build/envsetup.sh \$ lunch --&gt; Select "19. full_tcc8930st-eng"</pre>
	Bootloader	<pre>\$ cd bootable/bootloader/lk \$ vi target/tcc8930st_evm/rules.mk --&gt; HW_REV=0x6230 \$ make tcc8930st_evm</pre>
	Kernel	<pre>\$ cd kernel \$ make tcc8930st_defconfig \$ make</pre>
	Frameworks	<pre>\$ vi device/telechips/tcc8930st/full_tcc8930st.mk --&gt; TARGET_BOARD_8930_EV := true \$ make</pre>