

Android

Quick Start Guide

TCC892x-STB-Android-IceCreamSandwich-V1.04E-Quick Start Guide

June 25, 2012

TeleChips

Preliminary

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

This product may include technology owned by Microsoft Corporation and in this case it cannot be used or distributed without a license from Microsoft Licensing, GP.

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:

"Supply of this implementation of DTS technology does not convey a license, exhaust DTS' rights in the implementation, or imply a right under any patent, or any other industrial or intellectual property right of DTS to use, offer for sale, sell, or import such implementation in any finished end-user or ready-to-use final product. Notice is hereby provided that a license from DTS is required prior to such use."

"This product made under license to U.S. Patents 5,451,942; 5,956,674; 5,974,380; 5,978,762; 6,487,535; 6,226,616 and/or foreign counterparts."

"© 1996 – 2010 DTS, Inc."

Preliminary

Revision History

Date	Version	Description
2012-01-27	1.00	Initial release
2012-03-02	1.01	Update for ICS v12.03_r1-tcc-android-4.0.3 official release
2012-04-20	1.02	Update for ICS v12.04_r1-tcc-android-4.0.4 official release
2012-05-21	1.03	Update for ICS v12.05_r1-tcc-android-4.0.4 official release
2012-06-25	1.04	Update for ICS v12.06_r1-tcc-android-4.0.4 official release

TABLE OF CONTENTS

Contents

Revision History	3
TABLE OF CONTENTS	4
Contents	4
Figures.....	4
1 Introduction	6
2 Download Android ICS SDK	6
3 Compile Bootloader	6
3.1 Compile Environment	6
3.1.1 Evaluation Board	6
3.1.2 HDB892S Board.....	6
3.1.3 HDB892F Board	6
3.1.4 HDMI Dongle Board	7
3.2 Compile	7
4 Compile Linux Kernel.....	9
4.1 Compile Environment	9
4.1.1 Evaluation Board	10
4.1.2 HDB892S Board.....	10
4.1.3 HDB892F Board	10
4.1.4 HDMI Dongle Board	11
4.2 Compile	11
5 Compile Android Framework	12
5.1 Compile Environment	12
5.1.1 Evaluation Board	12
5.1.2 HDB892S Board.....	12
5.1.3 HDB892F Board	12
5.1.4 HDMI Dongle Board	12
5.2 Setup the Type of NAND Flash Memory	15
5.3 Compile Frameworks	15
6 Download Bootloader	17
6.1 Change Boot Mode	17
6.1.1 Evaluation Board	17
6.1.2 HDB892S Board.....	17
6.1.3 HDB892F Board	17
6.1.4 HDMI Dongle Board	17
6.2 Install VTC driver	17
6.3 Download.....	17
6.3.1 Select a file to donwload	17
6.3.2 Make and Load an Image for NAND partiton	18
6.3.3 Load a file	21
6.3.4 Download a file	22
7 Download Other Images	23
7.1.1 Enter Fastboot Mode	23
7.1.2 Install Android USB Driver.....	23
7.1.3 Download.....	24
8 Download All Images by using FWDN_V7	26

Figures

Figure 1. Execute FWDN_V7	18
Figure 2. Select a file to download.....	18
Figure 3. Add a file to file list	18
Figure 4. Select NAND Data	19
Figure 5. Select a Path.....	19
Figure 6. Save a file	20

Figure 7. Create an Image	20
Figure 8. Add a file to file list.....	21
Figure 9. Device Probing window	21
Figure 10. NAND Erase Pop-up window	22
Figure 11. Activate Start icon	22
Figure 12. During downloading a file	22
Figure 13. After completing to download	22
Figure 14. Install Android USB driver.....	23
Figure 15. Area Map is activated	26

1 Introduction

This document provides guideline for users to start Android platform v4.0 (ICS, Ice Cream and Sand witch) for TCC892x set-top-box (or media box) quickly.

There are four types of target board for TCC892x STB solution below.

- Evaluation
- HDB892S
- HDB892F
- 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 ICS 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 ics” option to download Android ICS SDK.

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

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

3 Compile Bootloader

3.1 Compile Environment

3.1.1 Evaluation Board

There are no changed points to setup compile environment for evaluation board.

3.1.2 HDB892S Board

There are no changed points to setup compile environment for HDB892S board.

3.1.3 HDB892F Board

Set `HW_REV` to **0x2001**(DDR3 1CS 512MB) or **0x2002**(DDR3 2CS 1024MB) in “bootable/bootloader/lk/target/tcc8920st_evm/rules.mk”.

```
#-----  
# Define board revision  
# 0x1000 : TCC892X_STB_D2_08X4_SV01 - DDR2 512MB(32BIT)  
# 0x1001 : TCC892x_STB_D2_08X4_SV01 - DDR2 1024MB(32BIT)  
# 0x1002 : TCC892X_STB_D3_08X4_SV01 - DDR3 512MB(32BIT)  
# 0x1003 : TCC892X_STB_D3_08X4_SV01 - DDR3 1024MB(32BIT)  
# 0x1004 : TCC892X_STB_D3_08X2_SV01 - DDR3 512MB(16Bit)  
# 0x2000 : TCC8925_HDMI_DONGLE - DDR3 512MB(16Bit)
```

```
# 0x2001 : TCC8925_HDB892F_1CS_SV0.1 - DDR3 512MB(16Bit)
# 0x2002 : TCC8925_HDB892F_2CS_SV0.1 - DDR3 1024MB(16Bit)
# 0x2003 : TCC8925_ISDBT_MODULE - DDR3 512MB(16Bit)
# 0x2004 : TCC8925_HDMI_DONGLE_EXT_DCDC - DDR3 512MB(16Bit)

#HW_REV=0x1000
#HW_REV=0x1001
#HW_REV=0x1002
#HW_REV=0x1003
#HW_REV=0x1004
#HW_REV=0x2000
HW_REV=0x2001
#HW_REV=0x2002
#HW_REV=0x2003
#HW_REV=0x2004
```

3.1.4 HDMI Dongle Board

Set HW_REV to 0x2000 in “bootable/bootloader/lk/target/tcc8920st_evm/rules.mk”. If you use HDMI dongle board which version is “TCC8925_HDMI_DONGLE V0.6”, you should set HW_REV to 0x2003.

```
#-----
# Define board revision
# 0x1000 : TCC892X_STB_D2_08X4_SV01 - DDR2 512MB(32BIT)
# 0x1001 : TCC892X_STB_D2_08X4_SV01 - DDR2 1024MB(32BIT)
# 0x1002 : TCC892X_STB_D3_08X4_SV01 - DDR3 512MB(32BIT)
# 0x1003 : TCC892X_STB_D3_08X4_SV01 - DDR3 1024MB(32BIT)
# 0x1004 : TCC892X_STB_D3_08X2_SV01 - DDR3 512MB(16Bit)
# 0x2000 : TCC8925_HDMI_DONGLE - DDR3 512MB(16Bit)
# 0x2001 : TCC8925_HDB892F_1CS_SV0.1 - DDR3 512MB(16Bit)
# 0x2002 : TCC8925_HDB892F_2CS_SV0.1 - DDR3 1024MB(16Bit)
# 0x2003 : TCC8925_ISDBT_MODULE - DDR3 512MB(16Bit)
# 0x2004 : TCC8925_HDMI_DONGLE_EXT_DCDC - DDR3 512MB(16Bit)

#HW_REV=0x1000
#HW_REV=0x1001
#HW_REV=0x1002
#HW_REV=0x1003
#HW_REV=0x1004
HW_REV=0x2000
#HW_REV=0x2001
#HW_REV=0x2002
#HW_REV=0x2003
#HW_REV=0x2004
```

3.2 Compile

Move in “bootable/bootloader/lk” folder and execute “make tcc8920st_evm” for NAND boot or “make tcc8920st_evm_emmc” for eMMC(T-Flash/SD) boot.

```
$ cd ~/mydroid/android/bootable/bootloader/lk
$ make tcc8920st_evm
```

```
$ cd ~/mydroid/android/bootable/bootloader/lk
$ make tcc8920st_evm_emmc
```

Then "lk.rom" will be created in "bootable/bootloader/lk/build-tcc8920st_evm/" folder.

If you want to clean the compiled result, you can execute "make tcc8920st_evm clean".

4 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.

4.1 Compile Environment

Before you compile Linux kernel and Android frameworks, TARGET_PRODUCT must be setup for proper board configuration. Execute “. build/envsetup.sh”. There is space between dot (‘.’) and others (“build/envsetup.sh”) to execute shell-script. Then if you execute “lunch”, you can see lists below.

```
$ cd ~/mydroid/android

$ . build/envsetup.sh
including device/moto/stingray/vendorsetup.sh
including device/moto/wingray/vendorsetup.sh
including device/samsung/crespo4g/vendorsetup.sh
including device/samsung/crespo/vendorsetup.sh
including device/samsung/maguro/vendorsetup.sh
including device/samsung/torospr/vendorsetup.sh
including device/samsung/toro/vendorsetup.sh
including device/samsung/tuna/vendorsetup.sh
including device/telechips/m805_880x/vendorsetup.sh
including device/telechips/m805_892x/vendorsetup.sh
including device/telechips/tcc8800/vendorsetup.sh
including device/telechips/tcc8920st/vendorsetup.sh
including device/telechips/tcc8920/vendorsetup.sh
including device/ti/panda/vendorsetup.sh
including sdk/bash_completion/adb.bash

$ lunch

You're building on Linux

Lunch menu... pick a combo:
  1. full-eng
  2. full_x86-eng
  3. vbox_x86-eng
  4. full_stingray-userdebug
  5. full_wingray-userdebug
  6. full_crespo4g-userdebug
  7. full_crespo-userdebug
  8. full_maguro-userdebug
  9. full_torospr-userdebug
 10. full_toro-userdebug
 11. full_tuna-userdebug
 12. full_m805_880x-eng
 13. full_m805_880xv8-eng
 14. full_m805_892x-eng
 15. full_m805_892xv8-eng
 16. full_m805_892x_emmc-eng
 17. full_tcc8800-eng
 18. full_tcc8800v8-eng
 19. full_tcc8800emmc-eng
 20. full_tcc8920st_evm-eng
 21. full_tcc8920stv8-eng
 22. full_tcc8920st_emmc-eng
```

```
23. full_tcc8920-eng
24. full_tcc8920v8-eng
25. full_tcc8920emmc-eng
26. full_panda-eng
```

You can select 'full_tcc8920st_evm-eng' or 'full_tcc8920st_emmc-eng' for development of TCC892x STB solution, and 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] 1
```

4.1.1 Evaluation Board

Move in "kernel" folder and execute "make tcc8920st_defconfig" to setup default configuration for evaluation board.

```
$ cd ~/mydroid/android/kernel
$ make tcc8920st_defconfig
```

Configuration file(".config") will be created in "kernel" folder and you can modify a file ("kernel/arch/arm/configs/tcc8920st_defconfig") to change default configuration.

4.1.2 HDB892S Board

Move in "kernel" folder and execute "make tcc8920st_hdb892s_defconfig" to setup default configuration for HDB892S board.

```
$ cd ~/mydroid/android/kernel
$ make tcc8920st_hdb892s_defconfig
```

Configuration file(".config") will be created in "kernel" folder and you can modify a file ("kernel/arch/arm/configs/tcc8920st_hdb892s_defconfig") to change default configuration.

4.1.3 HDB892F Board

Move in "kernel" folder and execute "make tcc8925st_hdb892f_defconfig" to setup default configuration for HDB892F DDR3 2CS 1024MB board.

```
$ cd ~/mydroid/android/kernel
$ make tcc8925st_hdb892f_defconfig
```

Kernel configuration should be changed below for HDB892F DDR3 2CS 1024MB after setting-up default configuration.

```
$ make menuconfig
System Type --> DRAM Memory Size --> 1024MB
```

Configuration file(".config") will be created in "kernel" folder and you can modify a file ("kernel/arch/arm/configs/tcc8925st_hdb892f_defconfig") to change default configuration.

4.1.4 HDMI Dongle Board

Move in "kernel" folder and execute "make tcc8925st_dongle_defconfig" to setup default configuration of HDMI dongle board.

```
$ cd ~/mydroid/android/kernel
$ make tcc8925st_dongle_defconfig
```

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.

```
$ make menuconfig
System Type --> TCC892x HDMI Dongle WiFi Chip -->

( ) Artheros AR6102/6103
(X) Realtek RTL8188/8189
( ) Broadcom BCM4330
```

Configuration file(".config") will be created in "kernel" folder and you can modify a file ("kernel/arch/arm/configs/tcc8925st_dongle_defconfig") to change default configuration.

4.2 Compile

To compile Linux Kernel, just execute "make".

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

5 Compile Android Framework

5.1 Compile Environment

Please refer chapter 4.1 to setup TARGET_PRODUCT.

5.1.1 Evaluation Board

There are no changed points to setup compile environment for evaluation board.

5.1.2 HDB892S Board

There are no changed points to setup compile environment for HDB892S board.

5.1.3 HDB892F Board

There are no changed points to setup compile environment for HDB892F board.

5.1.4 HDMI Dongle Board

Define TARGET_BOARD_HDMI_DONGLE value as true for TCC8925 HDMI dongle board.

```
"device/telechips/tcc8920st/full_tcc8920st_evm.mk" --> NAND boot
"device/telechips/tcc8920st/full_tcc8920st_emmc.mk" --> eMMC boot
```

```
# Define if target board is HDMI Dongle
TARGET_BOARD_HDMI_DONGLE := true
```

You should change WIFI setting according to WIFI chipset that you use.

First, "device/telechips/tcc8920st/BoardConfig.mk" file should be changed.

For ARTEROS chipset

```
ifeq ($(PRODUCT_MODEL),TCC8925_HDMI_DONGLE)
# Wi-Fi defines
BOARD_USES_ATH_WIFI := true
BOARD_WIFI_MODULE := ar6003 # you can chose ar6002,ar6102,ar6003,ar6103,ar630
BOARD_USES_REALTEK_WIFI = false
BOARD_USES_BROADCOM_WIFI := false
BOARD_USES_WPS := true
else
...
endif
```

For REALTEK chipset

```
ifeq ($(PRODUCT_MODEL),TCC8925_HDMI_DONGLE)
# Wi-Fi defines
BOARD_USES_ATH_WIFI := false
BOARD_WIFI_MODULE := ar6003 # you can chose ar6002,ar6102,ar6003,ar6103,ar630
BOARD_USES_REALTEK_WIFI = true
BOARD_USES_BROADCOM_WIFI := false
BOARD_USES_WPS := true
else
...
endif
```

```
endif
```

Second, "device/telechips/tcc8920st/device_base.mk" file should be changed.

For ARTHEROS chipset

```
ifdef TARGET_BOARD_HDMI_DONGLE
# Atheros Wi-Fi module
$(call inherit-product, device/telechips/tcc892x-common/wifi/ar6003.mk)

# Realtek Wi-Fi module
$(call inherit-product, device/telechips/tcc892x-common/wifi/rt8189es.mk)

# Broadcom Wi-Fi module
$(call inherit-product-if-exists, hardware/broadcom/wlan/bcmdhd/firmware/bcm4330/device-bcm.mk)
$(call inherit-product, device/telechips/tcc892x-common/wifi/broadcom.mk)
else
...
endif
```

For REALTEK chipset

```
ifdef TARGET_BOARD_HDMI_DONGLE
# Atheros Wi-Fi module
$(call inherit-product, device/telechips/tcc892x-common/wifi/ar6003.mk)

# Realtek Wi-Fi module
$(call inherit-product, device/telechips/tcc892x-common/wifi/rt8189es.mk)

# Broadcom Wi-Fi module
$(call inherit-product-if-exists, hardware/broadcom/wlan/bcmdhd/firmware/bcm4330/device-bcm.mk)
$(call inherit-product, device/telechips/tcc892x-common/wifi/broadcom.mk)
else
...
endif
```

Third, "device/telechips/tcc8920st/device_base.mk" file should be changed.

For ARTHEROS chipset

```
on boot
    setprop tcc.wifi.vendor atheros
    #setprop tcc.wifi.vendor realtek
    #setprop tcc.wifi.vendor broadcom

# for Atheros Wi-Fi -----
service wpa_supplicant /system/bin/wpa_supplicant -Dwext -iwlan0 -c/data/misc/wifi/...
    disabled
    oneshot

service hostapd /system/bin/hostapd_wps /data/misc/wifi/hostapd.conf
    socket hostapd_wlan0 dgram 660 root wifi
    user root
    group wifi
    oneshot
    disabled

# Special for the ugly AR6003 code
service abtfilter1 /system/bin/abtfilter -a -d -v -z -s
    disabled
    oneshot

service wlan_tool /system/bin/wlan_tool
    user root
    group root
```

```

disabled
oneshot

# -----

# for Realtek wifi -----
# service wpa_supplicant /system/bin/wpa_supplicant -Dnl80211 -iwlan0 -c/data/misc/wifi/...
#     socket wpa_wlan0 dgram 660 wifi wifi
#     group wifi inet
#     disabled
#     oneshot

# service hostapd /system/bin/hostapd -dddd /data/hostapd/hostapd.conf
#     class late_start
#     user root
#     group root
#     oneshot
#     disabled
# -----

...

```

For REALTEK chipset

```

on boot
#setprop tcc.wifi.vendor atheros
setprop tcc.wifi.vendor realtek
#setprop tcc.wifi.vendor broadcom

# for Atheros Wi-Fi -----
#service wpa_supplicant /system/bin/wpa_supplicant -Dwext -iwlan0 -c/data/misc/wifi/...
#     disabled
#     oneshot

#service hostapd /system/bin/hostapd_wps /data/misc/wifi/hostapd.conf
#     socket hostapd_wlan0 dgram 660 root wifi
#     user root
#     group wifi
#     oneshot
#     disabled

# Special for the ugly AR6003 code
#service abtflt1 /system/bin/abtflt -a -d -v -z -s
#     disabled
#     oneshot

#service wlan_tool /system/bin/wlan_tool
#     user root
#     group root
#     disabled
#     oneshot
# -----

# for Realtek wifi -----
service wpa_supplicant /system/bin/wpa_supplicant -Dnl80211 -iwlan0 -c/data/misc/wifi/...
    socket wpa_wlan0 dgram 660 wifi wifi
    group wifi inet
    disabled
    oneshot

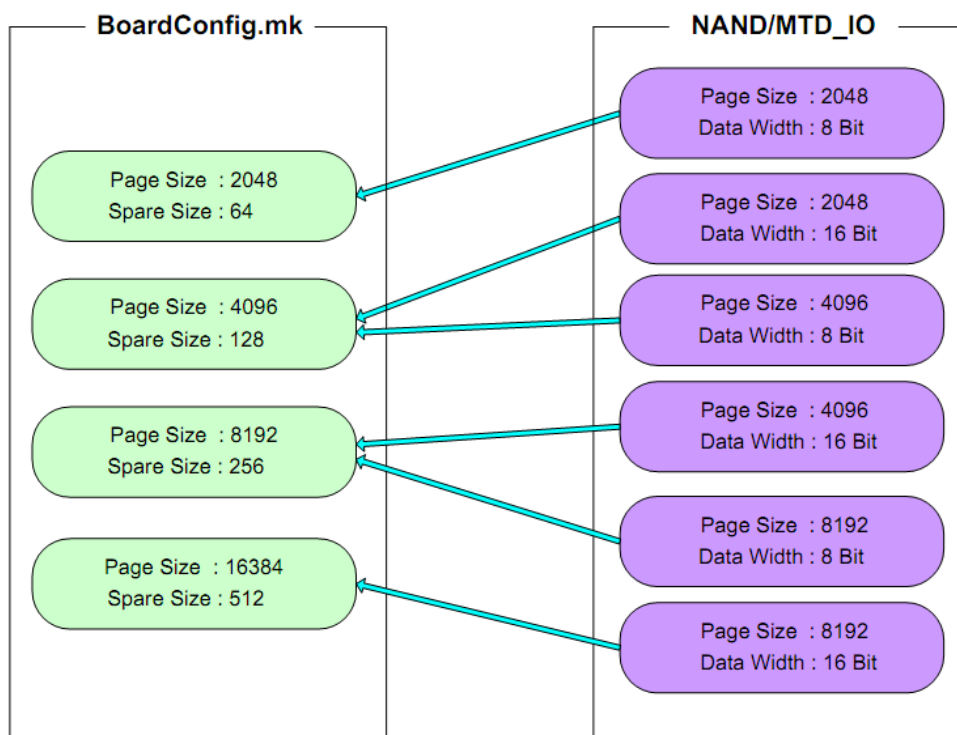
service hostapd /system/bin/hostapd -dddd /data/hostapd/hostapd.conf
    class late_start
    user root
    group root
    oneshot
    disabled
# -----

...

```

5.2 Setup the Type of NAND Flash Memory

Open and edit “device/telechips/tcc8920st/BoardConfig.mk” file for setting-up the type of NAND flash memory. The type of NAND flash memory can be checked in NAND available list or datasheet. You should setup page and spare size of NAND flash memory that you want to use.



```
$ cd ~/mydroid/android
$ vi device/telechips/tcc8920st/BoardConfig.mk
```

```
BOARD_NAND_PAGE_SIZE := 8192
BOARD_NAND_SPARE_SIZE := 256
BOARD_KERNEL_PAGESIZE := 8192
BOARD_FLASH_BLOCK_SIZE := 8192
```

5.3 Compile Frameworks

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

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

If you select proper TARGET_PRODUC, you can check it with below log. This log can be seen when you select “full_tcc8920st_evm-eng” or “full_tcc8920st_emmc-eng”.

```
=====
PLATFORM_VERSION_CODENAME=REL
PLATFORM_VERSION=4.0.4
TARGET_PRODUCT=full_tcc8920st_evm or full_tcc8920st_emmc
TARGET_BUILD_VARIANT=eng
TARGET_BUILD_TYPE=release
TARGET_BUILD_APPS=
TARGET_ARCH=arm
TARGET_ARCH_VARIANT=armv7-a-neon
HOST_ARCH=x86
HOST_OS=linux
HOST_BUILD_TYPE=release
BUILD_ID=IMM76D
=====
```


6 Download Bootloader

You must use FWDN_V7 program to download bootloader to target board. FWDN_V7 can be found in “vendor/telechips/tools/FWDN” folder. Please refer the description below to know how to download bootloader by using FWDN_V7. The version of FWDN_V7 must be higher than v2.22 for TCC892x STB solution.

If target board use eMMC boot, you should refer to “5 Prepare to Download With FWDN” in document below (vendor/telechips/documents/tcc892x/pdf/).

TCC892X-Android-IceCreamSandwich-V1.01E-eMMC Boot Qucik Start Guide.pdf

6.1 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.

6.1.1 Evaluation Board

You can see “USB” and “NAND” from JSW1 and switch it from “NAND” to “USB” to change boot mode of evaluation to “Firmware Download Mode”.

6.1.2 HDB892S Board

You can see “USB” and “NAND” from JSW1 and connect the jumper to “USB” direction for changing boot mode of HDB892S board to “Firmware Download Mode”.

6.1.3 HDB892F Board

You can see “USB” and “NAND” from JSW1 and connect the jumper to “USB” direction for changing boot mode of HDB892F board to “Firmware Download Mode”.

6.1.4 HDMI Dongle Board

You need to add hardware point to change boot mode of HDMI dongle board. You can discuss this point with hardware group of Telechips. So they will give some guide to you.

6.2 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 v5.0.0.3 for TCC892x STB solution.

6.3 Download

6.3.1 Select a file to donwload

You should compile bootloader before loading it.

1. Execute FWDN_V7.exe
2. “...” icon to select “lk.rom”.
3. Press “Add File” icon after selecting “lk.rom”.
4. “lk.rom” will be shown in file list.

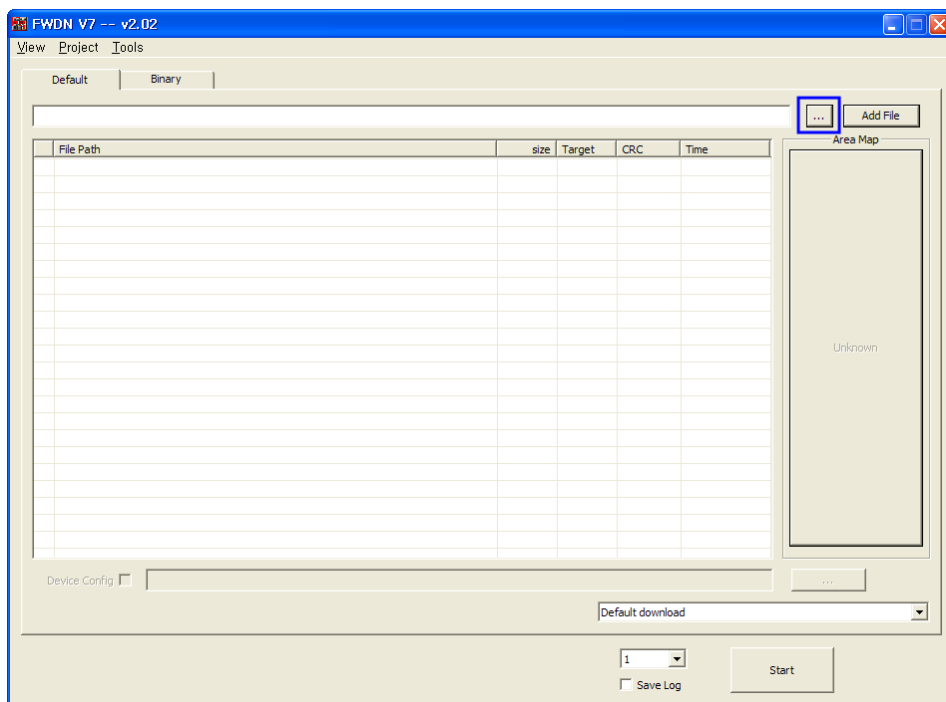


Figure 1. Execute FWDN_V7

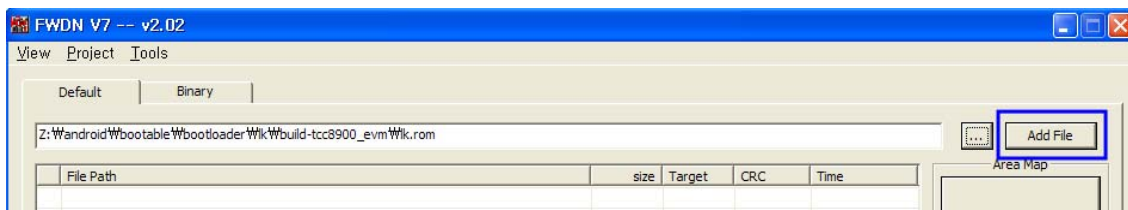


Figure 2. Select a file to download

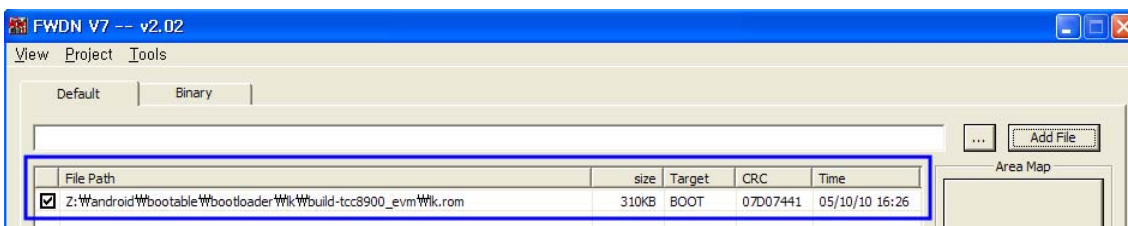


Figure 3. Add a file to file list

6.3.2 Make and Load an Image for NAND partiton

In case of *HDMI Dongle Board* you must make partitions of internal NAND flash memory. You can make NAND partitions by using a partition image and load it before downloading a bootloader image. You should download a bootloader and partition image together to *HDMI Dongle Board*.

Refer the description below for making and loading a partition image.

1. Select “NAND Data” area to make partitions.

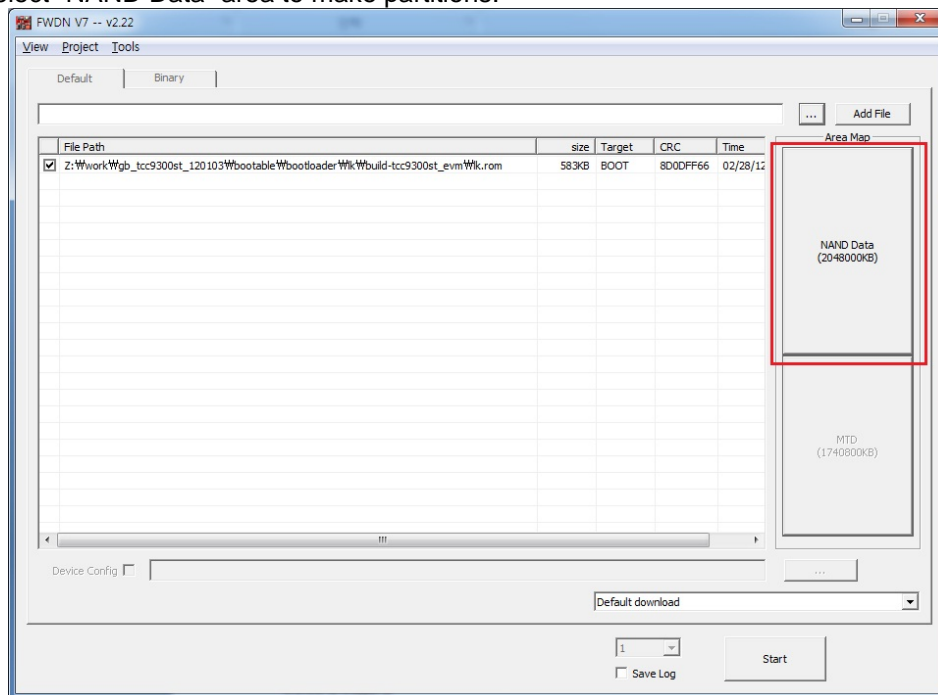


Figure 4. Select NAND Data

2. Select “...” to select a path to save “NAND Data.fai”

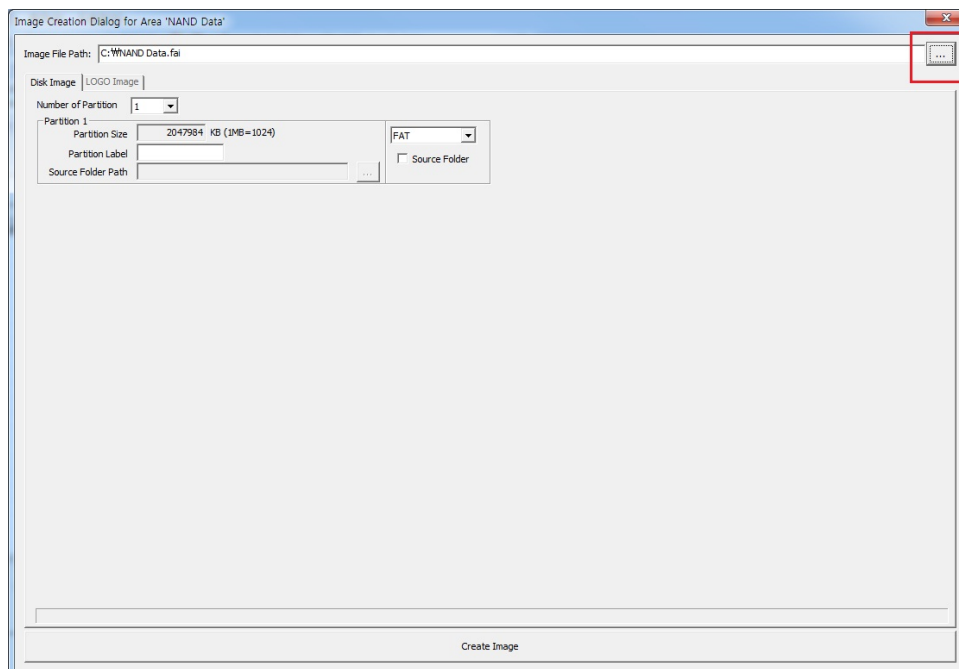


Figure 5. Select a Path

3. Save “NAND Data.fai” to a path that you select.

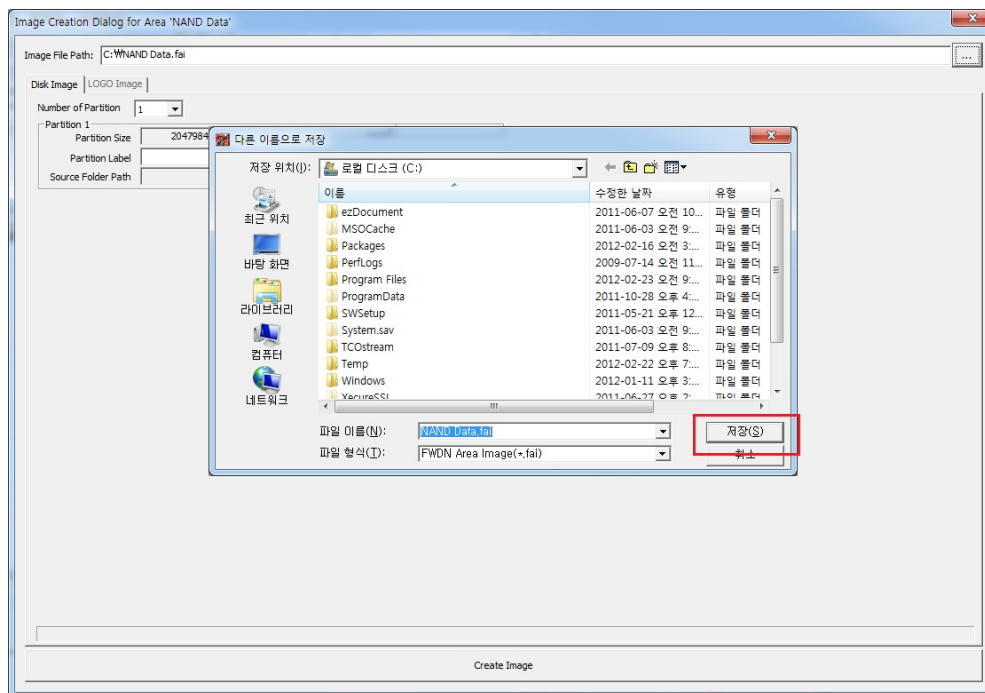


Figure 6. Save a file

4. Create an image to make NAND partitions.

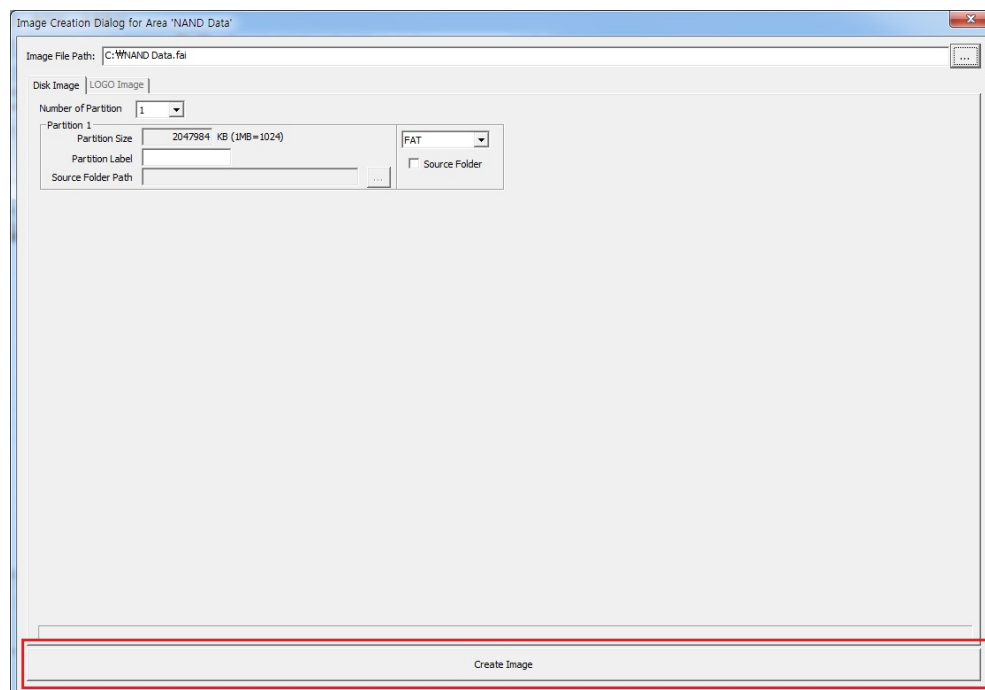


Figure 7. Create an Image

5. You can see "NAND Data.fai" to file list

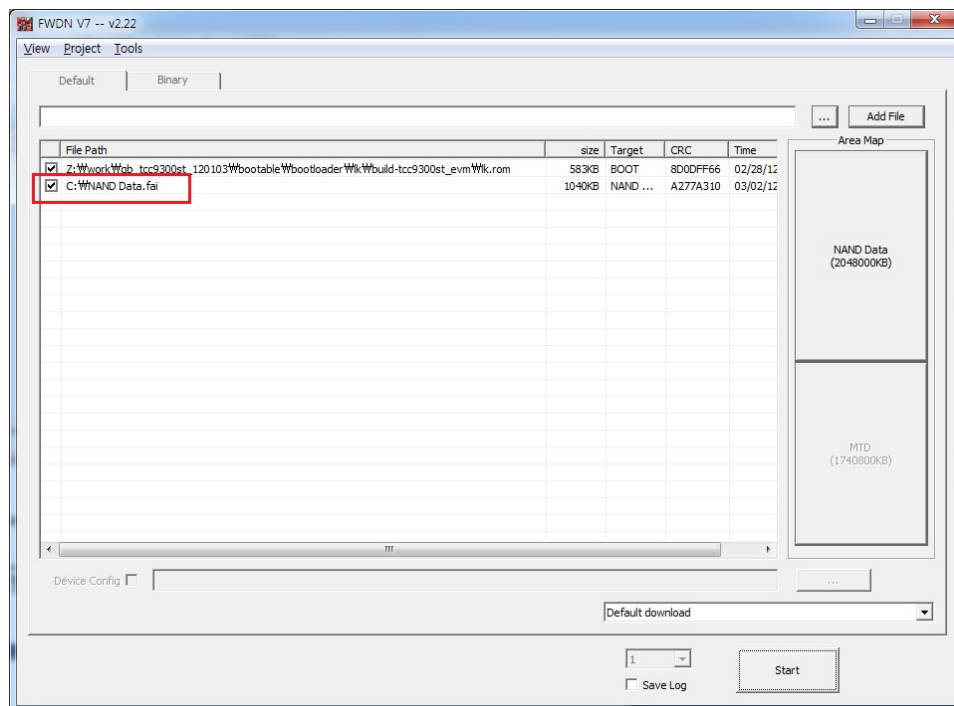


Figure 8. Add a file to file list

6.3.3 Load a file

Enter boot mode of target board in “Firmware Download Mode” and connect target board with PC by using USB cable. And then “Device Probing” window will appear and “lk.rom” will be loaded.

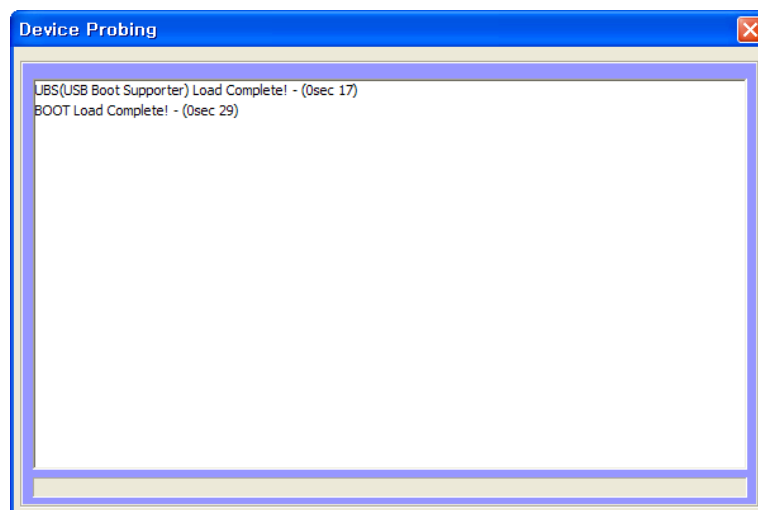


Figure 9. Device Probing window

“NAND Erase” pop-up window will be shown below if NAND partition was changed.

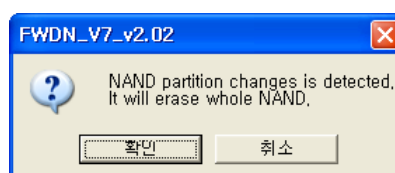


Figure 10. NAND Erase Pop-up window

If you press “Confirm” button, whole NAND will be erased before downloading firmware.

6.3.4 Download a file

“Start” icon will be activated after completing to load “lk.rom”.

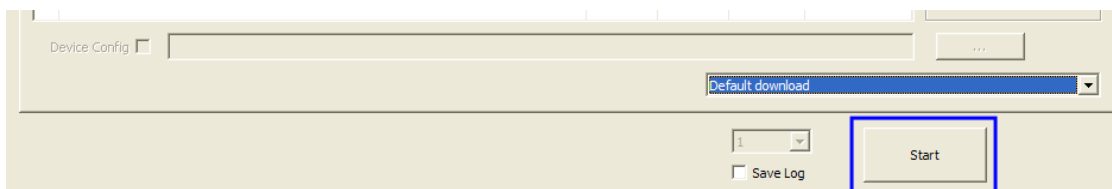


Figure 11. Activate Start icon

If you press “Start” icon, “Download” window will appear and start downloading “lk.rom”. If download is success, the color of window board will be changed to green color.

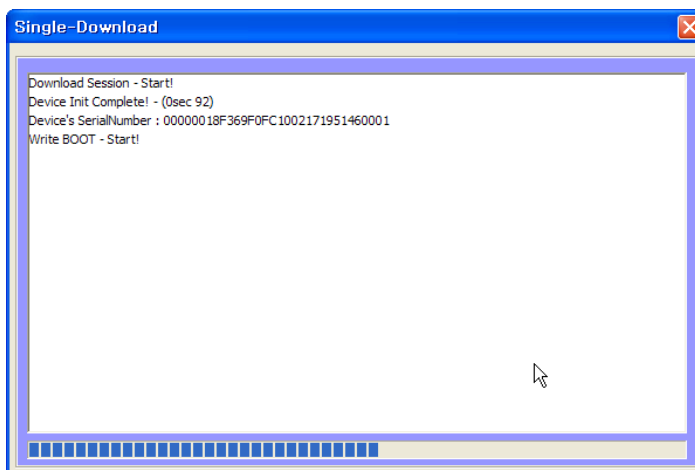


Figure 12. During downloading a file

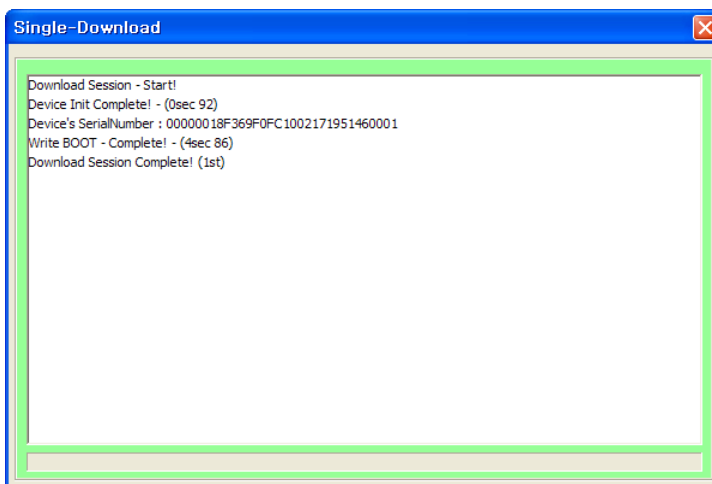


Figure 13. After completing to download

7 Download Other Images

Kernel and Framework image can be downloaded with “fastboot” command.

7.1.1 Enter Fastboot Mode

Change boot mode of target board to “NAND Boot Mode” after downloading bootloader and power off/on or reset target board. You can enter “Fastboot Mode” if you input ‘f’ character of keyboard on UART console window during boot-up. You can see messages below on console window if you enter “Fastboot Mode” normally.

```
getc: f
Diplay initialized
fastboot_init()
ept1 in @0x8219294c/0x3c max=512 bit=20000
ept1 out @0x8219296c/0x28 max=512 bit=2
udc_start()
```

7.1.2 Install Android USB Driver

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

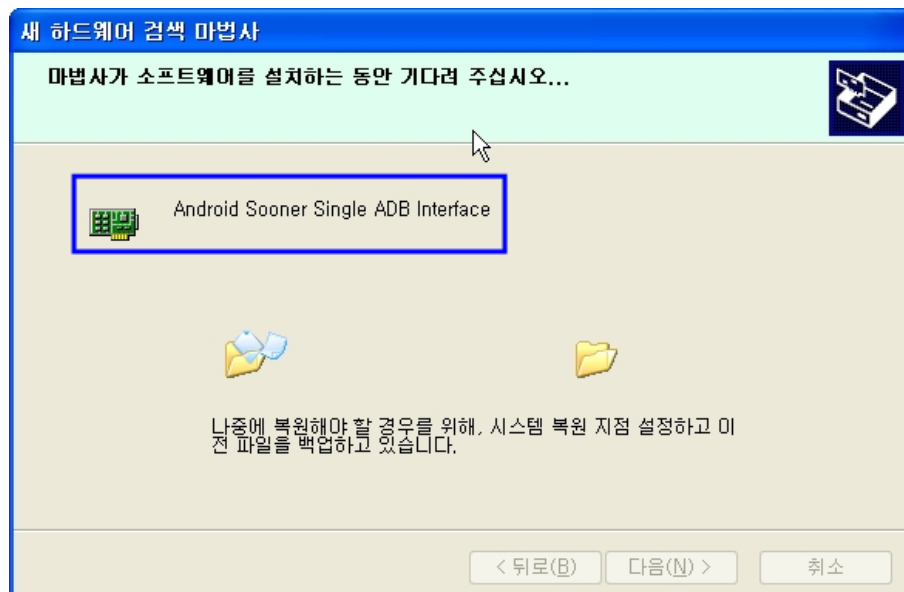


Figure 14. Install Android USB driver

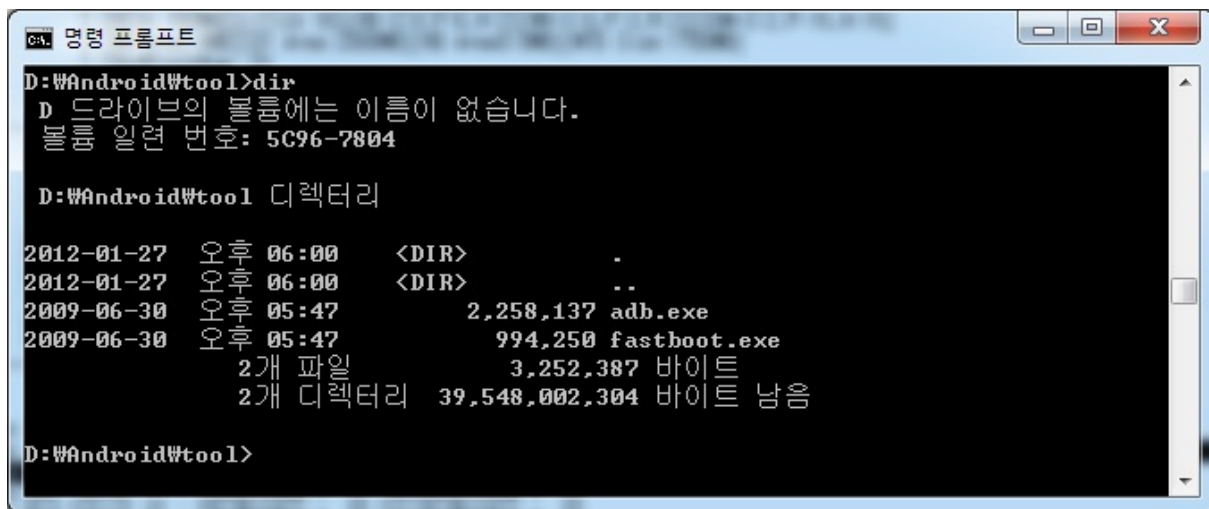
If target board is connected with PC, you can see messages below on console window

```
getc: f
Diplay initialized
fastboot_init()
ept1 in @0x8219294c/0x3c max=512 bit=20000
ept1 out @0x8219296c/0x28 max=512 bit=2
udc_start()
fastboot: processing commands
```

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

7.1.3 Download

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



```

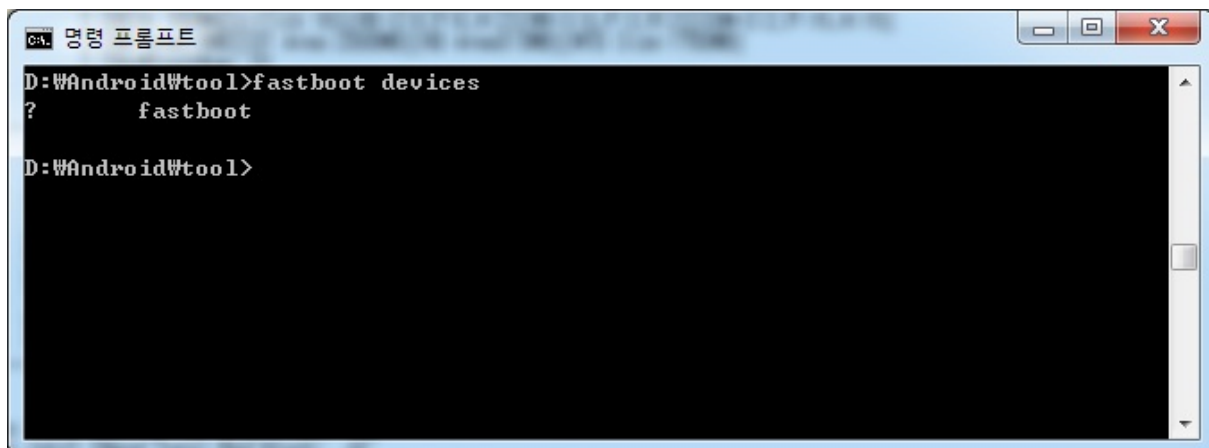
C:\> 명령 프롬프트
D:\AndroidWtool>dir
D 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 5C96-7804

D:\AndroidWtool 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:\AndroidWtool>
  
```

If you execute “fastboot devices” command, you can check list all connected devices.



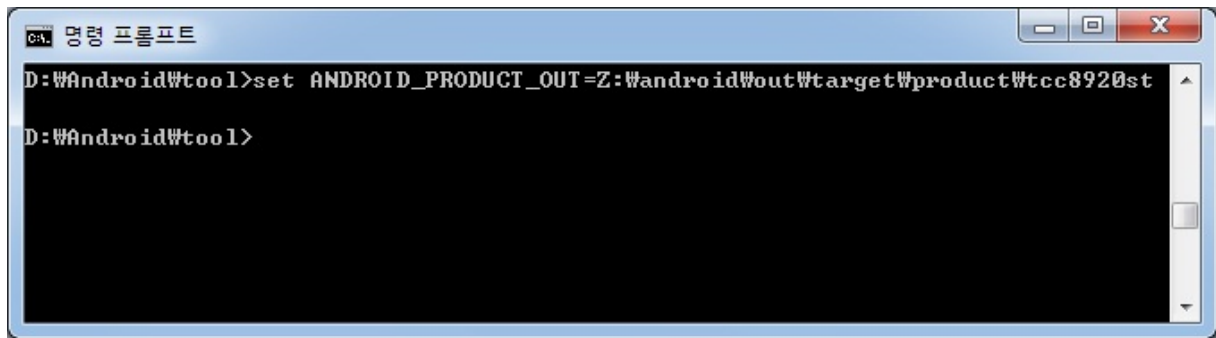
```

C:\> 명령 프롬프트
D:\AndroidWtool>fastboot devices
?      fastboot

D:\AndroidWtool>
  
```

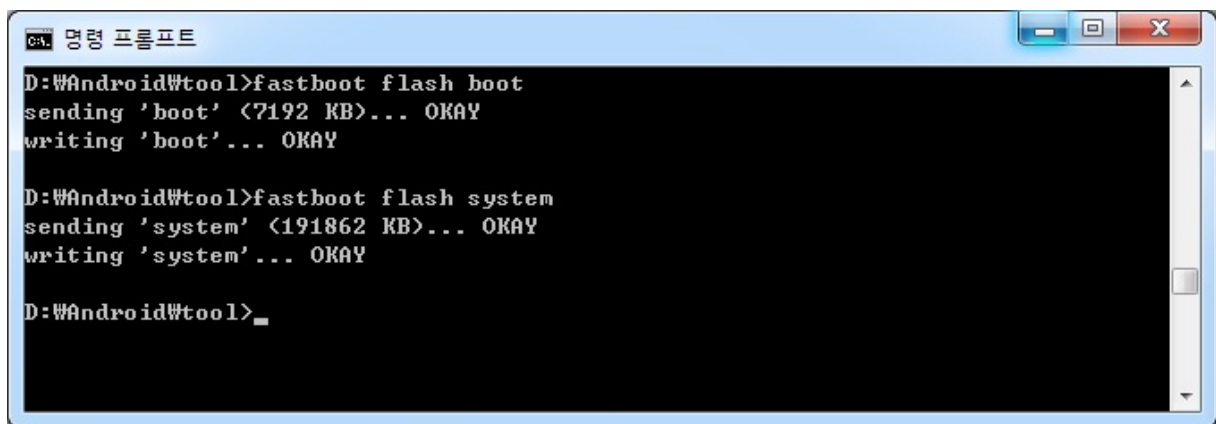
If there are no devices, no messages will be shown.

If compiled images exist in “z:\android\out\target\product\tcc8920st”, you must set the path with command below. So you don’t need to write full path to download images.



```
D:\WAndroid\Wtool>set ANDROID_PRODUCT_OUT=Z:\Wandroid\out\target\product\wtcc8920st
D:\WAndroid\Wtool>
```

You can download images by using “fastboot flash boot” and “fastboot flash system” commands.

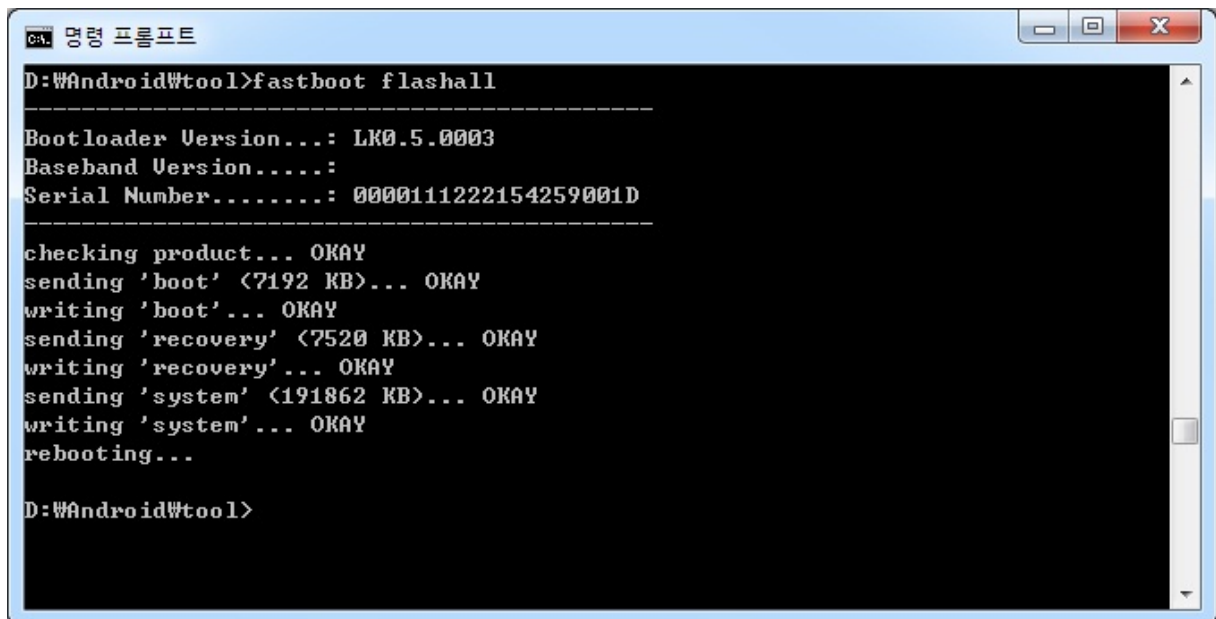


```
D:\WAndroid\Wtool>fastboot flash boot
sending 'boot' <7192 KB>... OKAY
writing 'boot'... OKAY

D:\WAndroid\Wtool>fastboot flash system
sending 'system' <191862 KB>... OKAY
writing 'system'... OKAY

D:\WAndroid\Wtool>
```

Otherwise, you can use “fastboot flashall” command to download all images.



```
D:\WAndroid\Wtool>fastboot flashall

-----
Bootloader Version...: LK0.5.0003
Baseband Version...:
Serial Number...: 0000111222154259001D
-----

checking product... OKAY
sending 'boot' <7192 KB>... OKAY
writing 'boot'... OKAY
sending 'recovery' <7520 KB>... OKAY
writing 'recovery'... OKAY
sending 'system' <191862 KB>... OKAY
writing 'system'... OKAY
rebooting...

D:\WAndroid\Wtool>
```

boot.img, recovery.img, and system.img are downloaded in the order. boot.img includes “kernel”, “ramdisk”, “command line parameter” and “base address”.

8 Download All Images by using FWDN_V7

We describes how to use FWDN_V7 for “lk.rom” and “fastboot” command for “boot.img” and “system.img” in previous chapters. Actually you can download all images by using only FWDN_V7. You should make an image including all img files by using “mkmtddimg” command below and you can download it with “lk.rom” by using FWDN_V7.

```
$ cd ~/mydroid/android/
$ out/host/linux-x86/bin/mkmtddimg --boot
out/target/product/tcc8920st/boot.img --system
out/target/product/tcc8920st/system.img --recovery
out/target/product/tcc8920st/recovery.img --output tcc8920st_mtd.img
```

You can add “tcc8920st_mtd.img” to file list by using same way to add “lk.rom”. Please refer

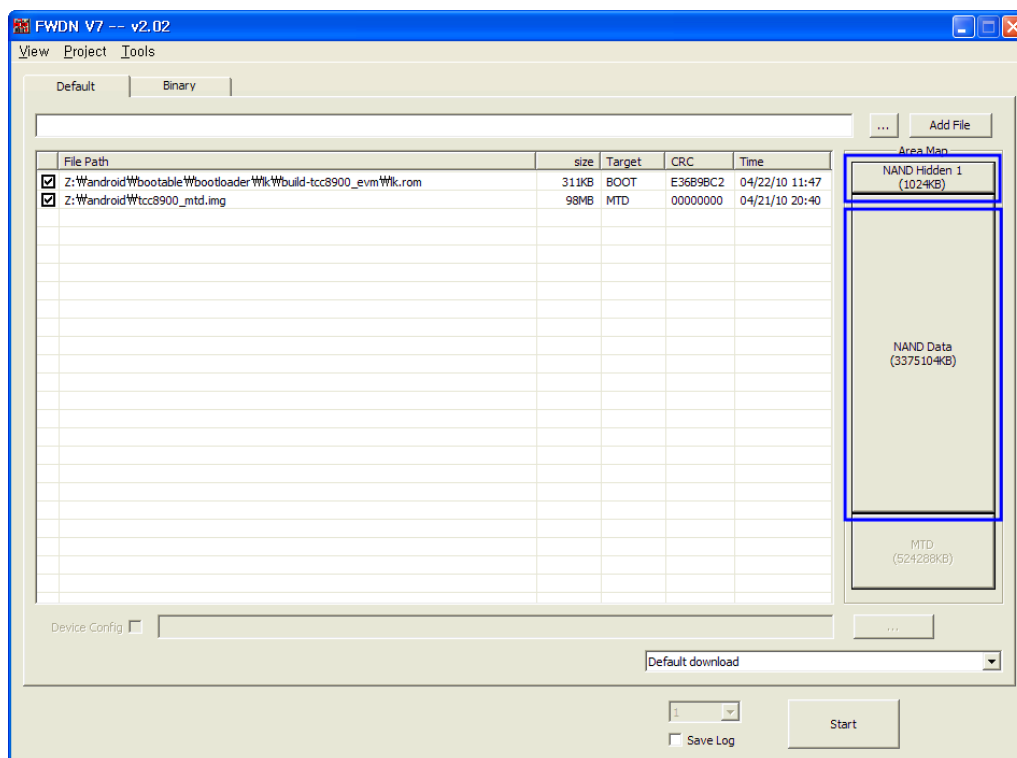


Figure 15. Area Map is activated

If your target board use eMMC boot, you should refer to “5 Prepare to Download With FWDN” in document below ([vendor/telechips/documents/tcc892x/pdf/](#)).

TCC892X-Android-IceCreamSandwich-V1.00E-eMMC Boot Qucik Start Guide.pdf