

# **TCCXXXX**

# **HOW TO CHANGE LOGO**

## **User Guide**

**TCCxxxxAndroid-ALL-V0.01E- How to change log**

**Rev. 1.00**

**Feb 27, 2012**

***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

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."

### **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."

**Revision History**

Date	Version	Description
2012-02-27	1.00	Initial release

## TABLE OF CONTENTS

**Contents**

1 Introduction .....	1-1
1.1 Logo Change List.....	1-1
2 Bootloader Logo Change .....	2-2
2.1 Change Bootloader Logo by Changing Logo.h .....	2-2
2.1.1 Make *.img File to extract Hex Value .....	2-2
2.1.2 Extract Hex Value .....	2-2
2.1.3 How to Change and Apply Bootloader Logo Size .....	2-2
2.2 Change Bootloader Logo by Changing Splash.img .....	2-3
2.2.1 Make Splash.img .....	2-3
2.2.2 Change lk bootloader settings .....	2-3
2.2.3 Apply to Android System.....	2-3
3 Change Kernel Logo .....	3-4
3.1 Set Host Environment.....	3-4
3.2 Make Logo Image .....	3-4
3.2.1 Make *.ppm File.....	3-4
3.2.2 Make and Change *.c File .....	3-4
3.3 Change Linux Kernel Source .....	3-4
3.3.1 Change kernel/drivers/video/logo/Kconfig .....	3-4
3.3.2 Change kernel/drivers/video/logo/Makefile .....	3-5
3.3.3 Change kernel/include/linux/linux_logo.h.....	3-5
3.3.4 Change kernel/drivers/video/logo/logo.c .....	3-5
3.4 Kernel Configuration .....	3-5
4 Android init logo .....	4-7
4.1 Make Image (*.rle ) for Android init Logo .....	4-7
4.2 Set Android init Logo Path .....	4-7
4.3 Apply Android init Logo .....	4-7
5 Android animation logo .....	5-8
5.1 Change android animation logo by modify image which responsible for animation.....	5-8
5.1.1 Control Android Animation Logo .....	5-8
5.1.2 Image to Control Animation .....	5-8
5.1.3 Apply Android Animation Logo .....	5-8
5.2 Change android animation logo by bootanimation.zip .....	5-9

## 1 Introduction

This document is intended to help to easily and quickly change various image logos used for bootloader, kernel, Android init and Android animation of Telechips. This document is for all platforms supported by Telechips.

- ※ Note that TCC892X is used as standard.
- ※ bmp image file saved in this document is 24bit.

### 1.1 Logo Change List

- Bootloader logo (telechips logo)
- Kernel logo (linux penguin image)
- Android init logo ( androboy + telechips logo)
- Android animation logo

## 2 Bootloader Logo Change

### 2.1 Change Bootloader Logo by Changing Logo.h

If you want to change your logo by changing the alignment of telechips\_logo[] of Logo.h file, refer to the followings.

#### 2.1.1 Make \*.img File to extract Hex Value

Prepare 480x272 sized bmp image file in the following path, change it into \*.img(raw) format as follows.

```
$ cd ~/mydroid/android/bootloader/bootable/lk
$ convert -depth 8 telechips_logo.bmp rgb:boot_logo.img // bmp → img(raw)
$ rgb2565 <boot_logo.img> boot_logo_480X272.img // RGB888 → RGB565
```

#### 2.1.2 Extract Hex Value

In order to extract hex value from the \*.img(raw) made in 2.1.1, a command called **hexdump** is used.

```
$ hexdump -v -e '\1 "0x%02x,"' boot_logo_480X272.img > boot_logo.h
```

If the above procedure is executed, hex values are arrayed in boot\_logo.h file. Then, these values should be applied to the alignment of telechips\_logo[].

```
... /* omit */ ...

unsigned char telechips_logo[]={
0x77,0xde,0x57,0xd6,0x57,0xd6,0x57,0xde,0x57,0xde,0x57,0xde,0x77,0xde,
0x57,0xde,0x57,0xde,0x77,0xde,0x77,0xde,0x77,0xde,0x77,0xde,0x57,0xde,
0x77,0xde,0x77,0xde,0x77,0xde,0x77,0xde,0x57,0xde,0x77,0xde,0x77,0xde,
... /* omit */ ...
0x57,0xd6,0x57,0xd6,0x57,0xd6,0x77,0xde,0x77,0xde,0x77,0xde,0x77,0xde,0x57,0xde,
0x77,0xde,0x77,0xde,0x77,0xde,0x77,0xde,0x77,0xde,0x77,0xde,0x77,0xde,0x77,0xde,
0x77,0xde,0x77,0xde,0x77,0xde,0x77,0xde,0x57,0xd6,0x77,0xde,0x77,0xde,0x77,0xde,
}
... /* omit */ ...
```

※ You may delete boot\_logo.img, boot\_logo\_480X320.img, boot\_logo.h files made in the above step.

#### 2.1.3 How to Change and Apply Bootloader Logo Size

You can find that 480X272 is set as a default for a bootloader logo size. If you want larger size of logo, you may 800x480 sized bmp image in 2.1.1 and change 480 to 800 and 272 to 480 in the following variables.

```
$ cd ~/mydroid/android/bootloader/bootable/lk
$ vi platform/tcc892x/lcdc.c

#define LCDC_FB_WIDTH 800 //480
#define LCDC_FB_HEIGHT 480 //272
```

Then, if you have made a bootloader image and download it to a board, you can find the logo has been changed. For more details on bootloader image creation and download, refer to Quick Start Guide in the Manual path.

## 2.2 Change Bootloader Logo by Changing Splash.img

If you want to change your Bootloader logo by making splash.img, refer to the followings.

### 2.2.1 Make Splash.img

Prepare 800x480 sized bmp image files in the following path, change it into \*.img(raw) format as follows.

```
$ cd ~/mydroid/android/device/telechips/common  
  
$ convert -depth 8 telechips_logo.bmp rgb:boot_logo.img // bmp → img(raw)  
$ rgb2565 <boot_logo.img> splash_800x480.img // RGB888 → RGB565
```

### 2.2.2 Change lk bootloader settings

The following is how to set bootloader for a boot logo by using a splash image. The bootloader image made in the following procedure should be downloaded to a board.

```
$ cd ~/mydroid/android  
$ vi bootable/bootloader/lk/target/tcc8920_evm/rules.mk
```

```
# Define Default Splash  
ifeq ($(DISP_DEFINES), DISPLAY_DUAL)  
DEFINES += DISPLAY_SPLASH_SCREEN_DIRECT=1  
else  
DEFINES += DISPLAY_SPLASH_SCREEN=1  
#DEFINES += DISPLAY_SPLASH_SCREEN_DIRECT=0  
endif
```

### 2.2.3 Apply to Android System

Prepare binary images in order to be downloaded to a board.

```
tools> set ANDROID_PRODUCT_OUT= ~/mydroid/android/out/target/product/tcc8920
```

- ※ The path of tools is the one located within the folder where Android SDK is installed.  
(example : H:\android-sdk-windows\tools)

```
tools> fastboot flash splash ~/mydroid/android/device/telechips/common/splash_800X480.img  
      ( path of splash image )
```

By using “fastboot flash splash”, if you download a splash image (logo image) to a board, you can find the changed boot logo.

### 3 Change Kernel Logo

#### 3.1 Set Host Environment

If a bootup logo is intended to be used in Android booting, change to .ppm format is required.  
(For more information on .ppm extension, refer to <http://www.fileinfo.com/extension/ppm>.)

#### 3.2 Make Logo Image

##### 3.2.1 Make \*.ppm File

Prepare 800x480 bmp image in the following path. For easy understanding, telechip\_logo.bmp image will be used.  
Then, the bmp image should be changed to \*.ppm format as can be seen in the following procedure.

```
$ cd ~/mydroid/android/  
$ cd kernel/drivers/video/logo  
  
# bmtopnm telechips_logo.bmp | pnmtoplainpnm > image800x480.ppm  
# pnmquant -fs 223 image800x480.ppm > image800x480_256.ppm  
# pnmnoraw image800x480_256.ppm > logo_telechips_clut224.ppm
```

##### 3.2.2 Make and Change \*.c File

Make logo\_telechips\_clut224.c corresponding to logo\_telechips\_clut224.ppm file made in 3.2.1.

```
$ cd ~/mydroid/android/  
$ cd kernel/drivers/video/logo  
$ cp logo_linux_clut224.c logo_telechips_clut224.c
```

The file should be changed as follows.

```
$ cd ~/mydroid/android/  
$ vi kernel/drivers/video/logo/logo_telechips_clut224.c  
  
static unsigned char logo_telechips_clut224_data[] __initdata = {  
... /* omit */ ...  
static unsigned char logo_telechips_clut224_clut[] __initdata = {  
... /* omit */ ...  
const struct linux_logo logo_telechips_clut224 __initconst = {  
    .type          = LINUX_LOGO_CLUT224,  
    .width         = 800,  
    .height        = 480,  
    .clutsize      = 160,  
    .clut          = logo_telechips_clut224_clut,  
    .data          = logo_telechips_clut224_data  
};
```

#### 3.3 Change Linux Kernel Source

##### 3.3.1 Change kernel/drivers/video/logo/Kconfig

In order to make options selected by the user in “make menuconfig”, move to the following path.

```
$ cd ~/mydroid/android/kernel/drivers/video/logo  
$ vi Kconfig
```



Then, add the following procedure.

```
# add this code
config LOGO_TELECHIPS_CLUT224
bool "Telechips 224-color logo"
default y
```

### 3.3.2 Change kernel/drivers/video/logo/Makefile

```
obj-$(CONFIG_LOGO) += logo.o
obj-$(CONFIG_LOGO_LINUX_MONO) += logo_linux_mono.o
obj-$(CONFIG_LOGO_LINUX_VGA16) += logo_linux_vga16.o
obj-$(CONFIG_LOGO_LINUX_CLUT224) += logo_linux_clut224.o
obj-$(CONFIG_LOGO_TELECHIPS_CLUT224) += logo_telechips_clut224.o
```

### 3.3.3 Change kernel/include/linux/linux\_logo.h

```
$ cd ~/mydroid/android/
$ vi kernel/include/linux/linux_logo.h
```

```
extern const struct linux_logo logo_linux_clut224;
extern const struct linux_logo logo_telechips_clut224;
```

### 3.3.4 Change kernel/drivers/video/logo/logo.c

```
$ cd ~/mydroid/android/
$ cd kernel/drivers/video/
$ vi logo.c
```

```
#ifdef CONFIG_LOGO_LINUX_CLUT224
/* Generic Linux logo */
logo = &logo_linux_clut224;
#endif

/* add */
#ifdef CONFIG_LOGO_GPH_CLUT224
/* Telechips Linux logo */
logo = &logo_telechips_clut224;
#endif
```

## 3.4 Kernel Configuration

By executing "make menuconfig", execute kernel configuration.

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

Move to Device Drivers->Graphic support -> bootup logo and check as follows.

```
[ ] FB limit clock high
  Support for LCD panels  --->
[ ] Backlight & LCD device support  --->
  Display device support  --->
  Console display driver support  --->
[*] Bootup logo  --->
```

```
--- Bootup logo
[ ] Standard black and white Linux logo
[ ] Standard 16-color Linux logo
[ ] Standard 224-color Linux logo
[*] Telechips 244-color logo
```

Then, execute save & exit and rebuild a kernel. If you execute booting with this kernel image, you can check the kernel logo which has been set previously.

## 4 Android init logo

### 4.1 Make Image (\*.rle ) for Android init Logo

In order to make an image(\*.rle) with Android init logo used, move to the following path.

```
$ cd ~/mydroid/android/  
$ cd device/telechips/common/
```

Prepare 800x480 sized bmp image file. then, change it to \*.img(raw) format with the following commands and then change the image into \*.rle file.

```
$ convert -depth 8 ci_256_800_480.bmp rgb:initlogo800x480.raw  
$ rgb2565 -rle <initlogo800x480.raw> initlogo800x480.rle
```

### 4.2 Set Android init Logo Path

In Android, You may set a path of Android init logo.

```
$ cd ~/mydroid/android/  
$ vi device/telechips/tcc8920/device.mk
```

You may select a certain file and this can be used for Android init logo.

```
PRODUCT_COPY_FILES += \  
    device/telechips/common/initlogo800x480.rle:root/initlogo.rle
```

### 4.3 Apply Android init Logo

According to 4.2, if you properly set a path of \*.rle file which has been made in 4.1, you can use your wanted Android init logo.

## 5 Android animation logo

### 5.1 Change android animation logo by modify image which responsible for animation.

#### 5.1.1 Control Android Animation Logo

The Android animation logo in this part has two image files and source code which handles control. Source code is composed of the part setting a path of image files and the part giving an effect to each image.

```
$ cd ~/mydroid/android/  
$ vi frameworks/base/cmds/bootanimation/BootAnimation.cpp
```

```
bool BootAnimation::android()  
{  
    initTexture(&mAndroid[0], mAssets, "images/android-logo-mask.png");    // main mask  
    initTexture(&mAndroid[1], mAssets, "images/android-logo-shine.png");  
    // logo which flows behind mask  
  
    ... omit ...  
}
```

#### 5.1.2 Image to Control Animation

```
$ cd ~/mydroid/android/  
$ find . -name android-logo-*.png  
  
./frameworks/base/core/res/assets/images/android-logo-mask.png  
./frameworks/base/core/res/assets/images/android-logo-shine.png
```

The Android animation logo to be changed has two png files and are configured as follows.



As can be seen above, the upper one is android-logo-mask.png and the lower one is android-logo-shine.png. In a Mask, letters are transparent. (Because it is a PNG file, alpha value exists.)

#### 5.1.3 Apply Android Animation Logo

```
$ cd ~/mydroid/android/  
$ cd frameworks/base/core/res/assets/images
```

As 5.2, if you make two images, save them to the above path and set a path of the images according to 5.1, you can use Android animation logo as you want.

## 5.2 Change android animation logo by bootanimation.zip

In 2.0(Éclair), the android boot animation has been changed to enable easy replacement or customization. There is a file called bootanimation.zip stored in [out/target/product/tcc8920/system/media/](#) in the root file system.

```
$ cd ~/mydroid/android/  
$ cd out/target/product/tcc8920/system/media
```

This file contains two things.

- 1) A description file (desc.txt) that outlines how the animation progresses, what images to use, image size etc.
- 2) Folder(s) that contain the images for the animation.

The basic structure of the bootanimation.zip file is as follows:

```
bootanimation.zip  
  desc.txt  
  android  
  part1
```

android and part1 are directories that contain a series of images for example, in android there is:

```
320_480_001.png  
320_480_002.png  
... omit ...  
320_480_023.png  
320_480_024.png
```

These images from the 'android' and 'part1' animations that are combined as outlined in the 'desc.txt' file to form the overall startup animation. The images are ordered by number and run in sequence.

```
240 320 10  
p 1 0 android  
p 0 0 part1
```

The 'desc.txt' file outlines how the animation progresses and a sample is as follows:

command	description
240	width of the animation
320	height of the animation
10	desired fps of the animation
p	defines a animation part
1	how many times this animation. Part loop
0	defines a pause ( max = 10)
android	the folder name where the animation images are
p	defined another animation part
0	defines that it loops forever ( until android start )
0	defines a pause
part1	folder for the second animation part

< table 1. Descript of command in the "desc.txt" >