GOWARRIOR

GoDroid

Application Notes

Customization of Boot Logo and Animation

v1.1

Copyright Statement

This document is provided "as is" without any warranty of any kind, either express or implied, statutory or otherwise; without limiting the foregoing, the warranties of satisfactory quality, fitness for a particular purpose or non-infringement are expressly excluded and under no circumstances will GoWarrior Community be liable for direct or indirect loss or damage of any kind, including loss of profit, revenue, goodwill or anticipated savings.

This document is intended only to assist the reader in the use of the product. GoWarrior Community makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, which is used at your own risk and should not be relied upon. The information could include technical inaccuracies or typographical errors. No license, whether express, implied, arising by estoppel or otherwise, to any intellectual property right is granted by this document.

The product described in this document is subject to continuous development and improvements. GoWarrior Community also reserves the right to make changes to the specifications and product description at any time without notice.

Third-party brands and names mentioned in this publication are for identification purpose only and may be the property of their respective owners.

Android™ is a registered trademark of Google Inc. Linux® is a registered trademark of Linus Torvalds. Microsoft® and Windows® are registered trademarks of Microsoft Corporation. Supply of this product does not convey a license nor imply a right under any patent, or any other industrial or intellectual property right of Google Inc., Linus Torvalds, and Microsoft Corporation 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 Google Inc., Linus Torvalds and Microsoft Corporation.

For the latest version of this document refer to:

www.gowarriorosh.com

Table of Contents

List of Tables

List of Figures

Preface

Overview

This manual mainly describes how to customize the boot Logo and boot animation of GoDroid v1.1. This manual is organized into the following chapters:

- **Chapter 1: Boot Logo**

This chapter provides details on how to create a boot Logo.

- **Chapter 2: Boot Animation**

This chapter provides information on how to customize boot animation.

Audience

This manual is written to provide complete guidance for those who wants to exploit GoWarrior TIGER Board, such as makers, tinkers, innovators, students, etc.

Applicable Products

This manual is applicable for the GoWarrior TIGER Board.

Reference Documents

N/A

Conventions

Typographical Conventions

Item	Format
codes, keyboard input commands, file names, equations, and math	Courier New, Size 10.5
Variables, code variables, and code comments	*Courier New, Size, Italic*
Menu item, buttons, tool names	**Ebrima, Size 10.5, Bold** e.g. Select **USB Debugging**
Screens, **windows**, dialog boxes, and tabs	**Ebrima, Size 10.5, Bold** **Enclosed in double quotation marks** e.g. Open the **"Debug Configuration"** dialog box

Table 1. Typographical Conventions

Symbol Conventions

Item	Description
⚠Caution	Indicates a potential hazard or unsafe practice that, if not avoided, could result in data loss, device performance degradation, or other unpredictable results.
✎Note	Indicates additional and supplemental information for the main contents.
💡Tip	Indicates a suggestion that may help you solve a problem or save your time.

Table 2. Symbol Conventions

How to Contact Us

Submit all your comments and error reports to the author at:

info@gowarriorosh.com

Tel: +886-2-8752-2000

Fax: +886-2-8751-1001

For questions regarding GoWarrior, contact our support team at the email listed below:

support@gowarriorosh.com

.

1 Boot Logo

The Android system booting takes relatively long time. During the system booting process, a static PNG image will be display first. Then subsequently, booting components like SurfaceFlinger, an animation composed of multiple PNG images will be display incessantly until completion of booting the entire Android system. Users can perform operations only once after the Android system enters the home page

This chapter provides a general overview and Procedure about how to customize the boot Logo.

After booting Android system, `/initlogo.rle` (a 565 rle compressed bitmap) will be read. If the system read done successfully, a Logo picture in `/dev/graphics/fb0` will be display. In case if the system read fails, it will set `/dev/tty0` as TEXT mode, open `/dev/tty0`, and output "A N D R O I D" characters.

1.1 Reference Code Files

Boot logo program includes the following files:

```
/system/core/init/init.c
```

```
/system/core/init/init.h
```

```
/system/core/init/logo.c
```

```
/system/core/rootdir/init.rc
```

```
/build/tools/rgb2565/ to565.c
```

1.2 Making initlogo.rle File

Procedure:

1. Make PNG picture

Create a picture with the resolution of 1280x720 by image processing software and save it as a file in PNG24 format, such as file name `android_logo.png`.

Note:

It only supports pictures of 24 bit PNG.

2. Convert to RAW format

Use **ImageMagick**'s convert command in Linux to convert the picture format into **RAW**. The command is:

```
$ convert -depth 8 android_logo.png rgb:android_logo.raw
```

Ubuntu 12.04 has pre-installed **ImageMagick**. In case if this tool is not been installed on the system, please execute the following command to install it:

```
$ sudo apt-get install imagemagick
```

There is also an **ImageMagick** open source software suit in Windows. The same convert command and parameters can be used to convert.

3. Convert RAW format to rle file

rgb2565 tool needs to be use after compiling Android. The source codes are in `/build/tools/rgb2565/ to565.c`.After building Android, a binary executable file rgb2565 will be generated in /`out/host/linux-x86/bin`. The convert command is as follows.

```
$ rgb2565 -rle < android_logo.raw > initlogo.rle
```

If you wish to change the file name `initlogo.rle`, you need to modify the macro in `/system/core/init/init.h`:

```
#define INIT_IMAGE_FILE "/initlogo.rle"
```

4. Copy file

Copy the ready-made file `initlogo.rle` to the directory `/device/gowarrior/tigerboard`, modify `/device/gowarrior/tigerboard/device.mk`, and add the following codes:

```
PRODUCT_COPY_FILES +=

device/gowarrior/tigerboard/initlogo.rle:root/initlogo.rle
```

5. Rebuild and burn image

After rebuilding, the file `initlogo.rle` will be saved into `/out/target/product/tigerboard/ramdisk.img and then it` `will be merge into kernel.ubo`. After burning `kernel.ubo`, restart the system, and then the newly added `initlogo.rle` will be display once the Android starts.

2 Boot Animation

During the Android system booting, a static PNG image will be display first. Then subsequently the program booting components like SurfaceFlinger, an animation composed of multiple PNG images will be display incessantly until completion of booting the entire Android system. Users can perform operations only once after the Android system enters the home page.

2.1 Reference Code Files

Boot animation program is located in the directory below:

/frameworks/base/cmds/bootanimation

It includes the following files:

- Android.mk
- BootAnimation.cpp
- BootAnimation.h
- bootanimation_main.cpp

2.2 Flow Introduction

When booting the Android system, init.c parses init.rc, which defines a boot animation service:

```
“service bootanim /system/bin/bootanimation”,

service bootanim /system/bin/bootanimation

       class main

       user graphics

       group graphics

       disabled

       oneshot
```

Bootanimation service is executed by `SurfaceFlinger.readyToRun () (property_set ("ctl.start", "bootanim") ;)` to start the boot animation; Bootanimation service is executed by

`bootFinished () (property_set ("ctl.stop", "bootanim") ;)` to stop the boot animation.

2.3 Making bootanimation.zip File

GoDroid boot animation is mainly composed of a zip package `bootanimation.zip` stored in the directory `/system/media` or `/data/local`. In case if the both directories exist in system, then boot animation will be preferably stored in `/data/local`.

These both directories are defined in `/frameworks/base/cmds/bootanimation/BootAnimation.cpp:`

```
#define USER_BOOTANIMATION_FILE "/data/local/bootanimation.zip"

#define SYSTEM_BOOTANIMATION_FILE "/system/media/bootanimation.zip"
```

The package includes several PNG pictures and a text file named `desc.txt`. During booting process, the pictures will be appear continuously on the screen in the file name sequence specified in the `desc.txt`, just like playing tape movies and forming animation.

Procedure:

1. Make animation pictures

First, make several PNG pictures, and number the files in animation sequence. Then, place all the numbered pictures in a folder, such as `part0`, as shown below:

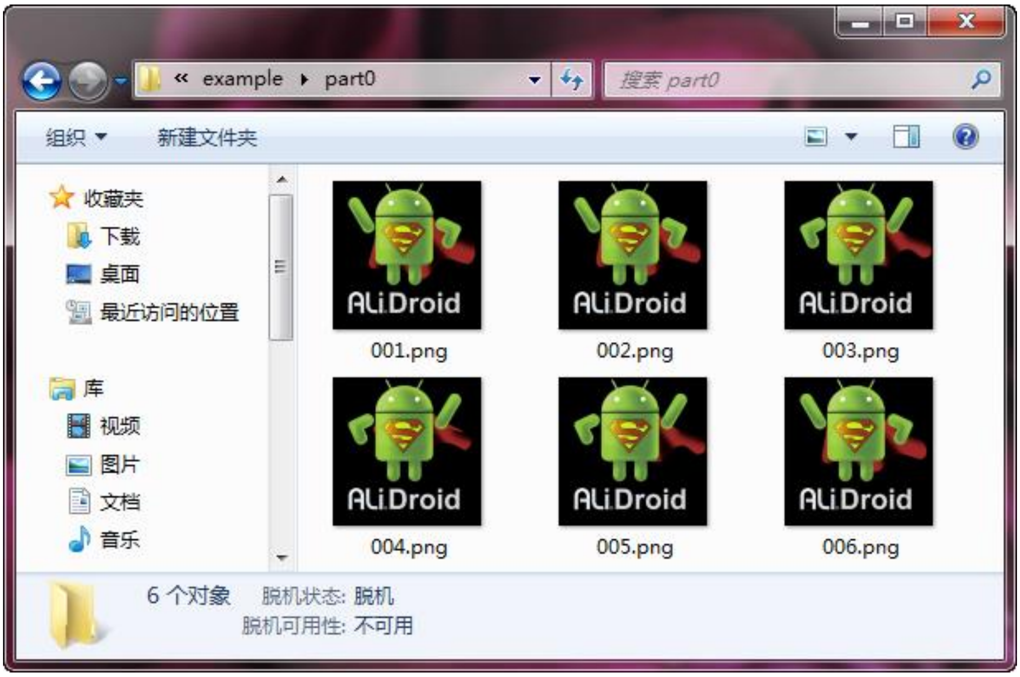


Figure 1. PNG Pictures for Boot Animation

Tip:

To prevent computer from consuming enormous amount of system memory, it is recommended to use smaller size of images.

2. Animation property description file

`desc.txt` is an ANSI file, used for setting animation pixel (size), frame numbers, flicker times, file folder name, etc. The content is as below:

```
336 336 4
p 1 0 part0
p 0 0 part1
```

- 336 336 4

Where the first 336 means the size (pixel) of picture width (Note: it must be an even number, multiples of 4 is the best); the second 336 means the size (pixel) of picture pixel height; 4 means the frame number.

- p 1 0 part0

Where p means identifier; 1 stands for cycle index; 0 means phase's interval time; part0 means the corresponding file folder name. It is the animation picture directory of Phase 1.

- p 0 0 part1

Here p stands for identifier; 0 stands for endless loop in this phase; 0 means phase's interval time; part1 means the corresponding file folder name. It is the animation pictures directory of Phase 2.

Phase switching interval: the unit is the time of a frame. For example, if the frame number is 30, then the frame duration time is 1s/30=33.3ms. During the phase switching interval, boot animation program enters into sleep status, leaving the CPU for system initialization. That is to say, the longer the sleep duration is, the faster the system is booted. But the animation effects will be compromised.

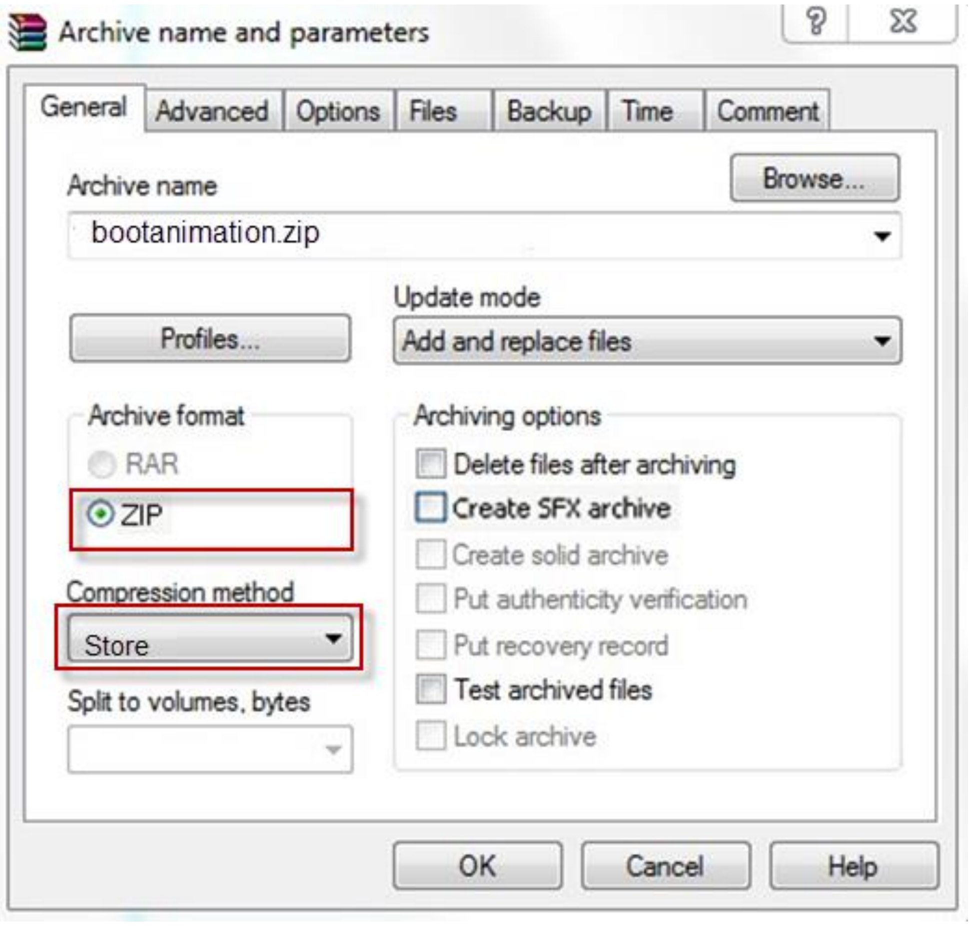
The `part0` and `part1` folders include the series pictures of two animations. The series pictures are in PNG format. They are sorted by the file name after loading and refreshing.

Note:

It is important to notice that there should be a line break (Pressing ENTER to jump to next line) after completing property description, to ensure the instruction is completed.

3. Package as bootanimation.zip after preparing pictures and desc.txt file

Select the picture folder `part0` and `desc.txt` (Reminder: please delete `Thumbs.db` from the picture folder before packaging), then right click to select "**Add to archive**" to enter "**Archive Name and parameters**" interface. Select "**ZIP**" as "**Archive format**" and "**Store**" as "**Compression method**"**.** Then click "**OK**".



Archive name and parameters

Archive name: bootanimation.zip

Update mode: Add and replace files

Archive format

○ RAR

● ZIP

Compression method: Store

Split to volumes, bytes

Archiving options

□ Delete files after archiving

□ Create SFX archive

□ Create solid archive

□ Put authenticity verification

□ Put recovery record

□ Test archived files

□ Lock archive

Figure 2. WINRAR Package Parameter Selection Interface

Note:

Set Compression method to Store.

4. Finally copy file bootanimation.zip and make image

Copy `bootanimation.zip` into `/device/gowarrior/tigerboard/system/media` or `/device/gowarrior/tigerboard/data/local`. Android system will select the animation in directory `/data/local` in preference, but it will be cleared during recovery.

Modify `/device/gowarrior/tigerboard/device.mk` simultaneously, and copy into directory `/system/media` or`/data/local` when building:

```
PRODUCT_COPY_FILES                                                     +=
device/gowarrior/tigerboard/system/media/bootanimation.zip:/system/me
dia/bootanimation.zip
```

After rebuilding, `bootanimation.zip` will be packaged into `system.img` or `userdata.img` in the directory `/out/target/product/tigerboard`.

Appendix: Glossary

Abbr.	Full Name
GoDroid	GoWarrior Android Development Kit
PNG	Portable Network Graphics
SurfaceFlinger	Graphics system for Android

Table 3. List of Abbreviations

Revision History

Document Change History

Revision	Changes	Date
v1.1	Updated document version to 1.1.	February 29, 2016
v1.0	Initial Release	September 07,2015

Table 4. Document Change History

Software Change History

Revision	Changes	Date
v1.1	Install and start GoDroid with a MicroSD card.	February 29, 2016
v1.0	Initial Release	September 07,2015

Table 5. Software Change History

www.gowarriorosh.com

Headquarters

Tel: +886-2-8752-2000

Fax: +886-2-8751-1001