GOWARRIOR

GoDroid

Application Notes

Flash Partition

v1.0

Copyright Statement

For the latest version of this document refer to:

www.gowarriorosh.com

Table of Contents

List of Tables

Preface

Overview

This manual mainly describes GoWarrior TIGER Board Flash partitions, including how to customize the generated partition image files. This manual is organized into the following chapters:

- **Chapter 1: FLASH Partition Introduction**

This chapter gives general overview about all the partitions in FLASH.

- **Chapter 2: How to Customize FLASH Partitions**

The chapter provides information on how to customize flash partitions.

- **Chapter 3: How to Customize Partition Burning Images**

The chapter discusses how to customize and generate burning images of each partition.

- **Chapter 4: FAQ**

This chapter describes range of common questions and their corresponding solutions.

Audience

This manual is primarily 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 FLASH Partition Introduction

This chapter gives compact description on NAND Flash partitions of the GoWarrior TIGER Board. In this chapter we take 8k/page, 2M/block, and 4G NAND FLASH as example.

The flash is divided into 14 physical partitions as shown in the table below:

Partition	Description
Bootloader	The first bootarea, used for loading and jumping to main kernel or recovery.
bootloaderbak	The second bootarea. If OTP signature is enabled and the first bootarea verification fails, this partition will be enabled.
bootargs	Boot parameters partition, which includes partition table, cmdline, and memmapping. The partition must be located in 2048 Pages, which is accessed by bootloader.
deviceinfo	Device info partition, including serial number, MAC address, hardware version, and HDCP key
baseparams	The partition of system parameters, including AVInfo and software version
misc	The upgrade mark partition. Bootloader decides whether to start up recovery according to the partition contents.
Recovery	The recovery partition can be considered as an alternative boot partition that lets you boot the device into a recovery console for performing advanced recovery and maintenance operations on it.

Partition	Description
recoverybak	Recovery backup partition. If recovery is damaged, recoverybak will be enabled.
backup	System backup partitions used to realize system restore. This backup includes kernel and system partitions.
cache	Upgrade the buffer partition. Use cache to store upgrade command parameters, upgrade package, and upgrade log.
bootmedia	Bootloader bootlogo/bootmedia video
kernel	Main+ramdisk+see+ae
system	This partition basically contains the entire operating system (Android) (read only), other than the kernel and the bootloader.
data	The partition contains the user data (read and write) – this is where your contacts, messages, settings, and apps that you have installed go.

Table 3. Physical Partition of NAND FLASH

2 How to Customize FLASH Partitions

Flash partitions are defined by the xml file:

`./device/ali/m3932-demo/image/fs_ubi/Ali_nand_desc.xml`

In `Ali_nand_desc.xml`, the `<part_loop>` part is the related definition of flash partitions.

```
<part_loop flash_type="nand" flash_size="4G" page_size="8K" block_size="2M">
        <part name="boot" file="uboot_unify_1GB_training.abs.sign" local="fixed" package="bootloader.img">
                <size>0x800000</size>
                <version>ALI_BOOTLOADER_VERSION</version>
        </part>
        <part name="bootbak" file="uboot_unify_1GB_training.abs.sign" local="fixed">
                <size>0x800000</size>
        </part>
        <part name="bootargs" file="bootargs.abs" local="fixed" package="bootargs.img">
                <size>0x800000</size>
        </part>
        <part name="deviceinfo" file="deviceinfo.abs" local="fixed">
                <size>0x800000</size>
        </part>
        <part name="baseparams" file="baseparams.abs" local="fixed">
                <size>0x800000</size>
        </part>
        <part name="misc" file="" level="protected" local="fixed">
                <size>0x800000</size>
        </part>
        <part name="recovery" file="recovery.ubo" local="fixed" package="recovery.img">
                <size>0x2000000</size>
                <version>ALI_RECOVERY_VERSION</version>
        </part>
        <part name="recoverybak" file="recovery.ubo" local="fixed">
                <size>0x2000000</size>
        </part>
        <part name="backup" file="backup.abs" local="fixed">
                <size>0x14000000</size>
        </part>
        <part name="cache" file="cache.img" fs_type="ubifs" local="fixed">
                <size>0x1B800000</size>
        </part>
        <part name="bootmedia" file="bootmedia.ubo" local="unfixed" package="bootmedia.img">
                <size>0x1000000</size>
        </part>
        <part name="kernel" file="kernel.ubo" local="unfixed" package="kernel.img">
                <size>0x2000000</size>
        </part>
        <part name="system" file="system.img" fs_type="ubifs" local="unfixed" package="/system">
                <size>0x1D000000</size>
                <version>ALI_SYSTEM_VERSION</version>
        </part>
        <part name="data" file="userdata.img" fs_type="ubifs" local="fixed">
                <!-- the last part size will be auto computed -->
                <!-- lastpartsize = flash_size-lastpartoffset-block_size*16 -->
                <size>0x100000</size>
        </part>
</part_loop>
```

2.1 FLASH Information

The division of partition size is related to the flash type. The following information on NAND FLASH is defined in the <part_loop> attribute.

Attribute	Description
Flash_type	FLASH type : "nand"： NAND FLASH "nor"： NOR FLASH "emmc"： eMMC FLASH
Flash_size	FLASH size
Page_size	FLASH read and write unit size
Block_size:	FLASH erase unit size

Table 4. FLASH Attribute Definition

2.2 Partition Definition

Each <part> describes a partition. The sequence of partition description decides the burning sequence in FLASH.

Attribute	Description
Name	Partition name, is the unique identification for distinguishing and searching corresponding partition.
File	Partition burning files, will be located in *./image/* after compiling.
Fs_type	filesystem type "" – raw "initrd" – initrd fs "yaffs2" – yaffs2 fs

Attribute	Description
	"cramfs" – cramfs fs
Package	Files or directories corresponding to the full amount of upgrade package, are used when making the upgrade package. If one partition needs to be put into the full amount of upgrade package, the package should be set. If upgrade is based on the raw partition, you can write like this: `package="<partname>.img"` If upgrade is based on the file system. It starts with the symbol "/". You can write like this: `package="/<partname>"`.
Local	Whether the partition location is fixed or not. It is not used temporarily now. It is reserved for use when making an upgrade package.
Offset	Partition offset address. It can also be not defined. The address is calculated by size.
Size	Partition size

Table 5. Partition Definition

3 How to Customize Partition Burning Images

There are special documents for detailed introduction on Bootloader, kernel, system, and recovery. This chapter mainly discusses several other remaining partitions.

3.1 Bootargs Partition

Bootargs partition is used to store the boot parameters needed by U-Boot when booting up, including cmdline, partition table, and so on.

3.1.1 Cmdline

Cmdline is the command line that is passed by U-Boot to kernel. It contains the rootfs information, partition table, and so on.

Format is as shown below:

```
rootfstype=initrd          init=/init          androidboot.console=ttyS0
mtdparts=ali_nand:8M@0M(boot),8M@8M(bootbak),8M@16M(bootargs),8M@24M(
deviceinfo),8M@32M(baseparams),8M@40M(misc),32M@48M(recovery),32M@80M
(recoverybak),320M@112M(backup),440M@432M(cache),8M@872M(bootmedia),3
2M@880M(kernel),464M@912M(system),2688M@1376M(data)          mem=1024M
pm_counter=8824
```

```
<!-- more cmdline define here -->
<cmdline>init=/init androidboot.console=ttyS0</cmdline>
<!--recovery.param(1:enable, 0:disable) -->
<!--bit0: reset system -->
<!--bit1: USB/SD upgrade -->
<!--bit2: OTA upgrade -->
<!--bit3: NET upgrade -->
<recovery_cmdline>init=/init androidboot.console=ttyS0 recovery.param=7</recovery_cmdline>
```

In *<cmdline>*, the defined character string will be used into the cmdline of starting kernel.

In*<recovery_cmdline>*, the defined character string will be used into the cmdline of starting recovery kernel.

3.1.2 Partition Table

Whenever the partition name, number, sequence, or size has new updates, the partition table will be updated accordingly.

3.1.3 How to Quickly Generate a Partition Burning Image

Command line: `$ build bootargs.abs`

Output: `./image/bootargs.abs`

3.2 Deviceinfo Partition

Deviceinfo partition includes information, such as (sequence number, MAC address, hardware version, and HDCP key). Currently the information can't be upgraded; this issue is being well taken care of.

For further information please refer to the following file:

`./device/ali/m39n32-demo/image/xml/deviceinfo.xml`

```
<deviceinfo>
        <magic type="string" size="16">deviceinfo</magic>
        <hdmi>
                <hdcp type="file" size="288"></hdcp>
        </hdmi>
        <firmware>
                <sn type="string" size="64">AAAABBCCCDDEE0112233445566</sn>
                <mac1 type="mac">10:20:30:01:70:00</mac1>
                <mac2 type="mac">12:22:33:44:55:66</mac2>
                <mac3 type="mac">14:22:33:44:55:66</mac3>
                <mac4 type="mac">16:22:33:44:55:66</mac4>
                <oui type="uint">0x1008</oui>
                <hw_ver type="string" size="128">00000001</hw_ver>
                <rsv type="ubyte" size="1820"></rsv>
        </firmware>
        <bootkey>
                <bootkey_desc type="string" size="16">bootkey</bootkey_desc>

                <ir_enable type="uint">1</ir_enable>
                <ir_timeout type="uint">5000</ir_timeout>

                <panel_enable type="uint">1</panel_enable>
                <panel_count type="uint">2</panel_count>
                <panel_timeout type="uint" size="2">500,400</panel_timeout>
                <panel_keys type="uint" size="2">0xffff0001,0xffff0002</panel_keys>

                <gpio_enable type="uint">0</gpio_enable>
                <gpio_polar type="uint">0</gpio_polar>
                <gpio_position type="uint">5</gpio_position>
        </bootkey>
        <bootconfig>
                <bootconfig_desc type="string" size="16">bootconfig</bootconfig_desc>
                <bootserailno_enable type="ubyte">0</bootserailno_enable>
                <bootblverison_enable type="ubyte">0</bootblverison_enable>
                <bootmode_enable type="ubyte">0</bootmode_enable>
                <boothwversion_enable type="ubyte">0</boothwversion_enable>
                <status type="uint">1</status>
                <rsv type="ubyte" size="56"></rsv>
        </bootconfig>
</deviceinfo>
```

3.2.1 HDCP

If you need to define the HDCP key, please point it to a valid HDCP key file. The length of file is 288 bytes.

For example: `<hdcp type="file" size="288">hdcp.key</hdcp>`

3.2.2 Sequence Number

`<sn>`: stands for Sequence number string

3.2.3 MAC Address

`<mac1>`: stands for MAC address. We only use mac1 currently.

3.2.4 OUI

<oui>: stands for organizationally unique identifier (OUI) .

3.2.5 Hardware Version

<hw_ver>: stands for hardware version, which is stored in a string.

3.2.6 Forced to Enter Recovery Configuration

Keeping pressing any IR key during power-up will force the system into recovery.

Related configurations:

<ir_enable>: 1 –Enable the IR function to force the system into recovery;

0 – Disable this function.

<ir_timeout>: The key press duration required to force the system into recovery. Its unit is ms.

3.2.7 Configurations for Triggering System Restore

During power-up, if the specified GPIO is detected, the system restore configuration will be triggered. Related configurations:

<gpio_enable>: 1 –Enable the system restore detection function; 0 – Disable this function

<gpio_polar>: Define GPIO polarity, active high or low effective

<gpio_position>: GPIO number

3.2.8 Androidboot Parameter Configuration

You can add a series of androidboot commands in cmdline to pass properties to Android system. These properties can be parsed by the init in Android.

- <bootserailno_enable>

1: In cmdline, add “androidboot.serialno=XXXXX” to pass the sequence number.

0: No additional display

- <bootblverison_enable>

1: In cmdline, add “androidboot.bootloader=XXXXX” to pass bootloader version.

0: No additional display

- <bootmode_enable>

1: Besides <bootmode_enable> is “1”and bit0 in <status> is “1”, in cmdline, “androidboot.mode=factory” will be added.

0: No additional display

- <boothwversion_enable>

1: In cmdline, add “androidboot.hardware=XXXXX” to pass the information about hardware version.

0: No additional display

3.2.9 How to Quickly Generate a Partition Burning Image

Command line: $ build deviceinfo.abs

Output: ./image/deviceinfo.abs

3.3 Baseparams Partition

Partition includes avinfo and software version information and so on. For further information please refer to the following file:

./device/ali/m3932-demo/image/xml/baseparams.xml

```
<baseparams>
        <magic type="string" size="16">baseparams</magic>
        <avinfo>
                <tvSystem type="ubyte">LINE_720_25</tvSystem>
                <progressive type="ubyte">FALSE</progressive>
                <tv_ratio type="ubyte">TV_ASPECT_RATIO_AUTO</tv_ratio>
                <display_mode type="ubyte">DISPLAY_MODE_LETTERBOX</display_mode>
                <scart_out type="ubyte">SCART_YUV</scart_out>
                <vdac_out type="ubyte" size="6">VDAC_YUV_V,VDAC_YUV_U,VDAC_YUV_Y,VDAC_CVBS,VDAC_NULL,VDAC_NULL</vdac_out>
                <video_format type="ubyte">SYS_DIGITAL_FMT_RGB</video_format>
                <audio_output type="ubyte">SYS_DIGITAL_AUD_LPCM</audio_output>
                <brightness type="ubyte">50</brightness>
                <contrast type="ubyte">50</contrast>
                <saturation type="ubyte">50</saturation>
                <sharpness type="ubyte">5</sharpness>
                <hue type="ubyte">50</hue>
                <snd_mute_gpio type="ubyte">80</snd_mute_gpio>
                <snd_mute_polar type="ubyte">0</snd_mute_polar>
                <hdcp_disable type="ubyte">1</hdcp_disable>
                <rsv type="ubyte" size="7"></rsv>
        </avinfo>
        <sysinfo>
                <sw_ver type="string" size="128">ALI_SYSTEM_VERSION</sw_ver>
                <sw_md5 type="ubyte" size="128">1,2,3,4,5</sw_md5>
                <sw_private type="ubyte" size="6144">9,8,7,6,5,4</sw_private>
        </sysinfo>
        <rsv type="ubyte" size="1024"></rsv>
</baseparams>
```

3.3.1 Avinfo

<tvSystem>: Television system

PAL

NTSC

PAL_M

PAL_N

PAL_60

NTSC_443

SECAM

MAC

LINE_720_25

LINE_720_30

LINE_1080_25

LINE_1080_30

LINE_1080_50

LINE_1080_60

LINE_1080_24

LINE_1152_ASS

LINE_1080_ASS

PAL_NC

LINE_576P_50_VESA

LINE_720P_60_VESA

LINE_1080P_60_VESA

<progressive>

TRUE (true) – Progressive scanning

FALSE (false) – Interlaced scanning

<tv_radio>: Aspect ratio

TV_ASPECT_RATIO_43

TV_ASPECT_RATIO_169

TV_ASPECT_RATIO_AUTO

<display_mode>: Display mode

DISPLAY_MODE_NORMAL

DISPLAY_MODE_LETTERBOX

DISPLAY_MODE_PANSCAN

<scart_out>: Scart output

SCART_CVBS

SCART_RGB

SCART_SVIDEO

SCART_YUV

<vdac_out>: 6 VDAC configurations.

VDAC_CVBS

VDAC_SVIDEO_Y

VDAC_SVIDEO_C

VDAC_YUV_Y

VDAC_YUV_U

VDAC_YUV_V

VDAC_RGB_R

VDAC_RGB_G

VDAC_RGB_B

VDAC_SCVBS

VDAC_SSV_Y

VDAC_SSV_C

VDAC_NULL

<video_format>

SYS_DIGITAL_FMT_BY_EDID

SYS_DIGITAL_FMT_RGB

SYS_DIGITAL_FMT_RGB_EXPD

SYS_DIGITAL_FMT_YCBCR_444

SYS_DIGITAL_FMT_YCBCR_422

<audio_output>

SYS_DIGITAL_AUD_BS

SYS_DIGITAL_AUD_LPCM

SYS_DIGITAL_AUD_AUTO

<brightness>: Brightness. The value ranges from 0 to 100, with the default value of 50.

<contrast>: Contrast. The value ranges from 0 to, with the default value of 50.

<saturation>: Saturation. The value ranges from 0 to 100, with the default value of 50.

<sharpness>: Sharpness. The value ranges from 0 to 100, with the default value of 50.

<hue>: Hue. The value ranges from 0 to 100, with the default value of 50.

3.3.2 Software Version Information

<sw_ver>: the string about system software version

3.3.3 How to Quickly Generate a Partition Burning Image

Command line: `$ build baseparam.abs`

Output: `./image/ baseparam.abs`

3.4 Backup Partition

Backup partition contains all the data stored on the partitions of the system. It includes all the partition contents that require data recovery during system restore.

3.4.1 Customizing Backup Contents

According to the plan of current partition, the backup partition includes the

backup of baseparams, kernel, and system partitions. For further information please refer to the following file

./device/ali/m3932-demo/image/xml/backup.xml

```
<backup>
        <magic type="string" size="16">backup</magic>
        <default_flag>COMPRESS_ZIP</default_flag>
        <bupart type="file" name="baseparams">image/baseparams.abs.zip</bupart>
        <bupart type="file" name="kernel">image/kernel.ubo.zip</bupart>
        <bupart type="file" name="system" fs_type="ubifs">image/system.img.zip</bupart>
</backup>
```

<default_flag>: The default flag of all backup partitions is included by Backup partition. If the flag is undefined in *<bupart…>*, the default default_flag is used. It can be filled like this:

"None" – Don't contain any flag information

"COMPRESS_GZIP" – Compress the backup partition using the gzip method

"COMPRESS_LZO" – Compress the backup partition using the lzo method

"COMPRESS_7ZIP" – Compress the backup partition using the 7zip method

"COMPRESS_ZIP" – Compress the backup partition using the zip method

Note:

Currently, only "None" and "COMPRESS_ZIP" are supported.

<bupart>: One bupart represents one partition that needs to be backed up. Up to 16 partitions can be backed up.

Name: The name of backup partitions, which correspond to the partition name described in Ali_nand_desc.xml.

Flag: Each backup partition can use a separate flag. It is defined in the same way as default_flag. If flag definition is not available, the default_flag definition will be applied.

nodeValue: Points to the image file of backup partition.

3.4.2 How to Quickly Generate a Burning Image in Backup Partition

Command Line: `$ build image`

Output: `./image/ backup.abs`

Note:

The command "build image" can generate other partition burning images at the same time.

3.5 Bootmedia Partition

Bootmedia partition includes the first logo or video displayed on screen during power-up.

It differs from native Android in that the boot logo and boot animation of native Android are displayed only during system boot.

The bootmedia partition can support pictures (in JPEG or MEPG2 format), and also support video (MKV format) which can be a piece of animation, or a combination of several pictures and animations.

3.5.1 Customizing Bootmedia

Bootmedia description file:

`./device/ali/m3932-demo/image/xml/bootmedia.xml`

Directory of resource files like pictures or videos:

`./device/ali/m3932-demo/image/res/`

To better illustrate bootmedia customization, we use a relatively complex instance:

```
<bootmedia>
        <head>
                <magic type="string" size="16">adfbootmedia</magic>
        </head>
        <body>
                <media>
                        <name type="file" media_type="jpeg">image/res/Logo1.jpg</name>
                        <duration>10000</duration>
                </media>
                <loop count="2">
                        <media>
                                <name type="file" media_type="jpeg">image/res/Logo2.jpg</name>
                                <duration>10000</duration>
                        </media>
                        <media>
                                <name type="file" media_type="mkv">image/res/video.mkv</name>
                                <duration>10000</duration>
                        </media>
                </loop>
                <media>
                        <name type="file" media_type="mpeg2">image/res/Logo3.m2v</name>
                        <duration>10000</duration>
                </media>
        </body>
</bootmedia>
```

`<media>`: Each one media indicates a picture or a video

`media_type`: Type of media file: `mpeg2/jpeg/mkv`

`nodeValue`: resource file like "`image/res/Logo1.jpg`", which points to `/device/ali/m3932-demo/image/res/Logo1.jpg`

`<duration>`: Media playback duration (unit: ms)

`<loop>`: One or more media can be defined in a loop and will be played sequentially and circularly. Count is used to define the times of loop. If count is equal to 0 or undefined, those media will be played in an infinite loop.

If there are more than one media in bootmedia, they will be played in order. The playback time is defined by the `<duration>`. If `<loop>` is available, then the media will be played circularly within the loop defined.

As shown above example, the playback sequence:

Logo1 ->

Logo2 -> video -> // loop1

Logo2 -> video -> // loop2

Logo3

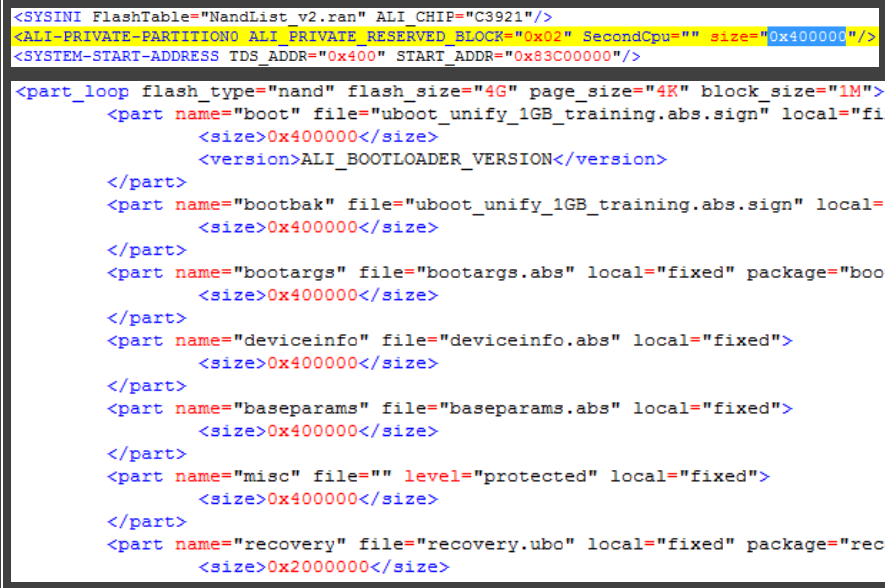
Please define bootmedia according to your actual requirement.

3.5.2 How to Quickly Generate a Partition Burning Image

Command line: `$ build bootmedia.ubo`

Output: `./image/ bootmedia.ubo`

4 FAQ

No.	Problem Description	Solution
1	How to define a partition table when using the NAND FLASH with 4k/page, 1M/block?	Modify several flash-related attributes in <part_loop>: flash_size, page_size, block_size Make sure that the starting address of the bootargs partition is the 2048th page. If the page size if 4k, then the start offset is 8M. The size of each partition should be the multiples of block size. If partition size of *boot* has been modified, the attribute size in <ALI-PRIVATE-PARTITION0> needs to be modified synchronously 
2	How to get rid of the low space warning when building a system image?	First, check that the partition size of *system* is enough to store system image. It is suggested that the partition size of *system* is two times the actual size of system image. The reserved space is for bad blocks and expansion in future.

No.	Problem Description	Solution
3	What is the actual size of Data partition?	Data partition is the last one in a partition table. The partition size of the last partition is automatically computed according to flash size and previous partitions. For more details, please refer to `AOSP/image/ALI.ini`.

Table 6. FAQ List

Appendix: Glossary

Abbr.	Full Name
GoDroid	GoWarrior Android Development Kit

Table 7. List of Abbreviations

Revision History

Document Change History

Revision	Changes	Date
v1.0	Initial Release	September 07,2015

Table 8. Document Change History

Software Changes

Revision	Changes	Date
v1.0	Initial Release	September 07,2015

Table 9. Software Change History

www.gowarriorosh.com

Headquarters

Tel: +886-2-8752-2000

Fax: +886-2-8751-1001