Bare and Non-Bare Chip Burning Upgrade

Operation Guide

Issue 00B06

Date 2019-03-05

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between HiSilicon and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

HiSilicon (Shanghai) Technologies Co., Ltd.

Address: New R&D Center, 49 Wuhe Road, Bantian,
Longgang District,
Shenzhen 518129 P. R. China

Website: http://www.hisilicon.com/en/

Email: support@hisilicon.com

About This Document

Purpose

This document provides the guidance for debugging bare and non-bare chip burning upgrade.

NOTE

- Unless otherwise stated, Hi3559C V100 and Hi3559A V100 contents are consistent.
- Hi3516E V200, Hi3516E V300, Hi3516D V200, and Hi3518E V300 do not support the SD card upgrade during MMC startup.

Related Versions

The following table lists the product versions related to this document.

Product Name	Version
Hi3559A	V100
Hi3559C	V100
Hi3519A	V100
Hi3556A	V100
Hi3516C	V500
Hi3516E	V200
Hi3516E	V300
Hi3518E	V300
Hi3516D	V200
Hi3516D	V300
Hi3516A	V300
Hi3559	V200
Hi3556	V200

Intended Audience

This document is intended for:

- Technical support engineers
- Software development engineers

Change History

Changes between document issues are cumulative. The latest document issue contains all changes made in previous issues.

Issue 00B06 (2019-03-05)

This issue is the sixth draft release, which incorporates the following changes:

Sections 1.3 and 2.3 are modified.

Issue 00B05 (2019-01-05)

This issue is the fifth draft release, which incorporates the following changes:

In section 2.3, Step 2 is modified.

Issue 00B04 (2018-10-26)

This issue is the fourth draft release, which incorporates the following changes:

Sections 1.3 and 2.3 are modified.

Issue 00B03 (2018-09-04)

This issue is the third draft release, which incorporates the following changes:

In section 2.3, Step 1 is modified.

Issue 00B02 (2018-05-15)

This issue is the second draft release, which incorporates the following changes:

Section 2.3 is modified.

Issue 00B01 (2018-02-10)

This issue is the first draft release, which incorporates the following changes:

The descriptions about the Hi3559 V100, Hi3556 V100, Hi3516C V300, and Hi3516E V100 are removed.

Sections 1.3 and 1.4 are modified.

Sections 2.3, 2.4, and 2.5 are modified.

Contents

Figures

1 Image Burning Upgrade Without U-Boot Pre-Installed

1.1 Preparations

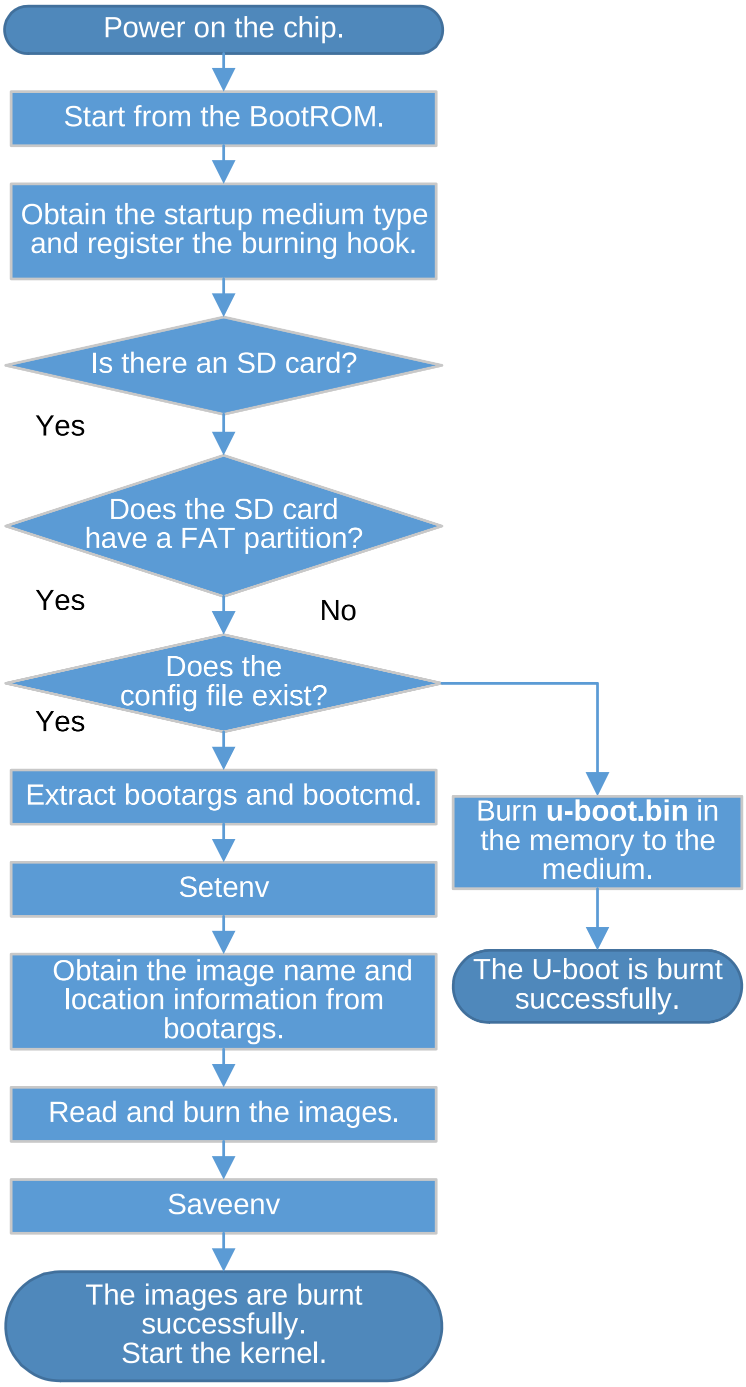
Prepare the following items:

- U-Boot, kernel, and rootfs images
- Upgrade package
- Storage medium (an SD card formatted with the FAT32 file system)

1.2 Process

Figure 1-1 shows the process of image burning upgrade without U-Boot pre-installed.

Figure 1-1 Process of image burning upgrade without U-Boot pre-installed


Power on the chip.
Start from the BootROM.
Obtain the startup medium type and register the burning hook.
Is there an SD card?
Yes
Does the SD card have a FAT partition?
Yes
No
Does the config file exist?
Yes
Extract bootargs and bootcmd.
Setenv
Obtain the image name and location information from bootargs.
Read and burn the images.
Saveenv
The images are burnt successfully. Start the kernel.
Burn **u-boot.bin** in the memory to the medium.
The U-boot is burnt successfully.

1.3 Procedure

Step 1 Compile the U-Boot, kernel, and rootfs images.

For the compiled U-Boot image, you need to enable the automatic upgrade macros. Go to the **u-boot-2016.11** directory and enable the macros in the **include/configs/hi35xx.h** file.

```
#define CONFIG_AUTO_UPDATE        1
#ifdef CONFIG_AUTO_UPDATE
   #define CONFIG_AUTO_UPDATE_ADAPTATION   1
   #define CONFIG_AUTO_SD_UPDATE    1
   #define CONFIG_AUTO_USB_UPDATE       1
   #define CONFIG_CMD_FAT        1
#endif
```

Step 2 Create the upgrade package.

Create the config file, and copy **bootargs** and **bootcmd** that match the image to the config file. The format is as follows:

- SPI NOR flash:

```
setenv bootargs 'mem=512M console=ttyAMA0,115200 clk_ignore_unused rw root=/dev/mtdblock2 rootfstype=jffs2 mtdparts=hi_sfc:1M(u-boot.bin),9M(kernel),16M(rootfs.jffs2) '
setenv bootcmd 'sf probe 0; sf read 4a000000 100000 900000; bootm 4a000000'
```

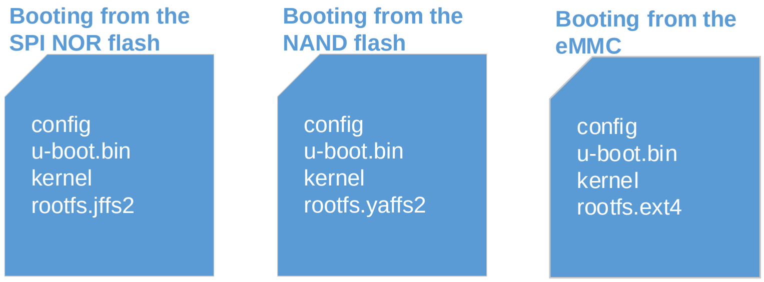
- NAND flash:

```
setenv bootargs 'mem=512M console=ttyAMA0,115200 clk_ignore_unused rw root=/dev/mtdblock2 rootfstype=yaffs2 mtdparts=hinand:1M(u-boot.bin),9M(kernel),32M(rootfs.yaffs2) '
setenv bootcmd ' nand read 4a000000 100000 900000; bootm 4a000000'
```

- eMMC:

```
setenv bootargs 'mem=512M console=ttyAMA0,115200 clk_ignore_unused rw rootwait root=/dev/mmcblk0p3 rootfstype= ext4 blkdevparts=mmcblk0:1M(u-boot.bin),9M(kernel),96M(rootfs.ext4) '
setenv bootcmd ' mmc read 0 4a000000 800 6000; bootm 4a000000'
```

The following is an example of the upgrade package:



Step 3 Insert the FAT32-formatted SD card that stores the upgrade package to SDIO 0, press the **UPDATE** button to start upgrading the board.

----End

NOTE

- Format the SD card with the FAT32 file system before saving the upgrade package onto it.
- The name of the U-Boot image in the upgrade package must be **u-boot.bin**, and the names of images must be consistent with those in **bootargs** of the config file. For example, the kernel image must be named **sample.bin** because it is **9M(sample.bin)** in **bootargs**.
- You can burn multiple file system images.
- The name of the yaffs image file must contain the yasffs character string, and that of any other image file cannot contain the yasffs character string.
- The name of the ext4 image file must contain the ext4 character string, and that of any other image file cannot contain the ext4 character string
- The **bootcmd** settings of Huawei LiteOS are different from those of Linux. You can run the following commands to configure **bootcmd** on Huawei LiteOS:

  SPI NOR flash: **setenv bootcmd 'sf probe 0;sf read 0x40000000 0x100000 0x400000;go 40000000'**

  SPI NAND flash: **setenv bootcmd 'nand read 0x40000000 0x100000 0x400000;go 40000000'**

1.4 Example

- Format the SD card as the FAT32 format.
- Copy the upgrade package created in section 1.3 "Procedure" to the formatted SD card. For details about how to format the SD card, see the *Peripheral Driver Operation Guide*.
- The following is an example of information displayed during the burning process (using SPI NOR flash as an example):

```
// Read u-boot.bin from BootROM to the memory and execute the U-Boot.
// Read the configuration file.
reading config
[0]=u-boot.bin    start=0x00000000 end=0x000fffff size=0x00100000
[1]=kernel        start=0x00100000 end=0x009fffff size=0x00900000
[2]=rootfs.jffs2  start=0x00a00000 end=0x01afffff size=0x01000000
// Read and burn u-boot.bin.
reading u-boot.bin
spinor erase...
spinor write...
```

```
// Read and burn the kernel.
reading kernel
spinor erase...
spinor write...
// Read and burn rootfs.jffs2.
reading rootfs.jffs2
spinor erase...
spinor write...
// Save the environment variables.
Erasing SPI flash, offset 0x00080000 size 256K ...done
Writing to SPI flash, offset 0x00080000 size 256K ...done
// The new system is started automatically.
```

1.5 Precautions

- The SD card must be formatted as the FAT32 format.
- If the SD card has multiple partitions, the upgrade package must be stored in the first partition; otherwise, it cannot be detected.
- The name of the U-Boot image must be **u-boot.bin**.
- The **bootargs** and **bootcmd** environment variables in the config file are automatically saved during the burning process.
- If no configuration file is provided, only the U-Boot image read by the BootROM is burnt.

1.6 eMMC Extended CSD Registers

If the burning is implemented via an SD card, U-Boot will use its own eMMC driver to configure the eMMC extended CSD registers. No extra configuration is needed. If burning is implemented via a burner, the burner needs to configure the eMMC extended CSD registers. This section describes how to configure relevant eMMC registers on the burner GUI.

The eMMC component contains the BOOT1, BOOT2 and USER DATA partitions and supports the n_RST pin and power-off reset. When booting from the USER DATA partition. All image data is burnt to the USER DATA partition. In addition, the eMMC can only be reset by using the n_RST pin. Therefore, register values must be configured according to the following table. Otherwise, the board cannot be started.

Register No.	Register Value	Description
179	0x38	This register is used to configure the boot partition. Boot from the USER DATA partition is used by default.
177	0x2	This register is used to configure the bus width of the eMMC in boot mode. You need to set the value according to the bus width used in the hardware design (0x1: 4-bit, 0x2: 8-bit).

Register No.	Register Value	Description
167	0x1f	This register is used to configure the write reliability of the eMMC component. The value of this register must be set to **0x1f**.
162	0x1	This register is used to configure whether the n_RST pin is valid for the eMMC. The n_RST pin is used by default, and the value of this register must be set to **0x1**.

NOTE

- The eMMC extended registers must be configured before burning.
- Some burners may not support the configuration of extended CSD registers. You need to ask the burner vendor for help.

The setting varies according to the eMMC burner. For details, see the burner manual.

2 Image Burning Upgrade With U-Boot Pre-Installed

2.1 Preparations

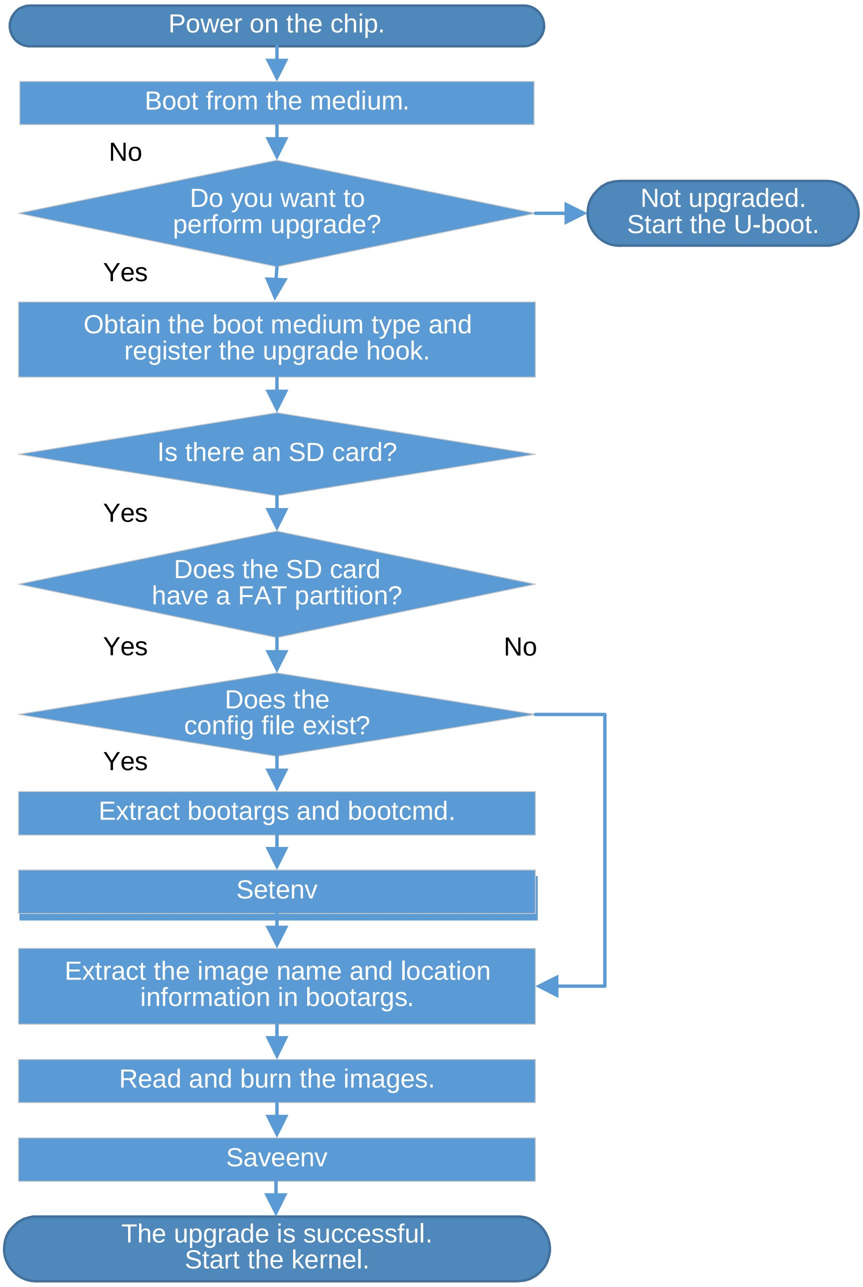
Prepare the following:

- U-Boot that supports upgrading, kernel image, and rootfs image
- Upgrade package
- Storage medium (an SD card or USB flash drive formatted with the FAT32 file system)

2.2 Process

Figure 2-1 shows the process of image burning upgrade with U-Boot pre-installed.

Figure 2-1 Process of image burning upgrade with U-Boot pre-installed


Power on the chip.
Boot from the medium.
No
Do you want to perform upgrade?
Not upgraded. Start the U-boot.
Yes
Obtain the boot medium type and register the upgrade hook.
Is there an SD card?
Yes
Does the SD card have a FAT partition?
Yes
No
Does the config file exist?
Yes
Extract bootargs and bootcmd.
Setenv
Extract the image name and location information in bootargs.
Read and burn the images.
Saveenv
The upgrade is successful. Start the kernel.

2.3 Procedure

Step 1 Compile the U-Boot image that supports upgrade, as well as the kernel and rootfs images.

For the compiled U-Boot image, you need to enable the automatic upgrade macros. Go to **u-boot-2016.11** and enable the macros in the **include/configs/hi35xx.h** file.

```
#define CONFIG_AUTO_UPDATE        1
#ifdef CONFIG_AUTO_UPDATE
```

```
   #define CONFIG_AUTO_UPDATE_ADAPTATION   1
   #define CONFIG_AUTO_SD_UPDATE     1
   #define CONFIG_AUTO_USB_UPDATE       1
   #define CONFIG_CMD_FAT         1
#endif
```

Step 2 Create the upgrade package.

Create the config file, and copy **bootargs** and **bootcmd** that match the upgrade package to the config file. The format examples are as follows:

- SPI NOR flash:

```
setenv bootargs 'mem=512M console=ttyAMA0,115200 clk_ignore_unused rw
root=/dev/mtdblock2 rootfstype=jffs2 mtdparts=hi_sfc:1M(u-
boot.bin),9M(kernel),16M(rootfs.jffs2) '
setenv bootcmd 'sf probe 0; sf read 4a000000 100000 900000; bootm 4a000000'
```

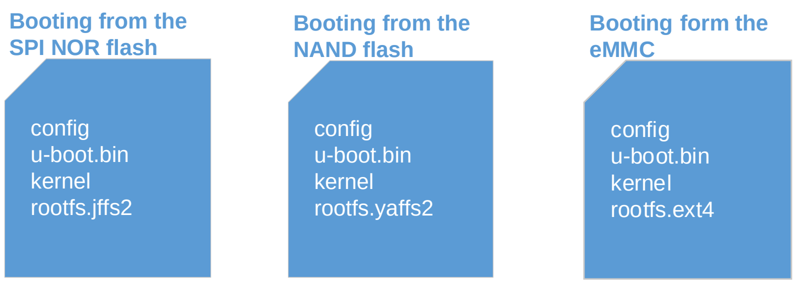
- NAND flash:

```
setenv bootargs 'mem=512M console=ttyAMA0,115200 clk_ignore_unused rw
root=/dev/mtdblock2 rootfstype=yaffs2 mtdparts=hinand:1M(u-boot.bin), 9M(kernel),
32M(rootfs.yaffs2) '
setenv bootcmd ' nand read 4a000000 100000 900000; bootm 4a000000'
```

- eMMC:

```
setenv bootargs 'mem=512M console=ttyAMA0,115200 clk_ignore_unused rw rootwait
root=/dev/mmcblk0p3 rootfstype= ext4 blkdevparts=mmcblk0:1M(u-
boot.bin),9M(kernel),96M(rootfs.ext4)'
setenv bootcmd ' mmc read 0 4a000000 800 6000; bootm 4a000000'
```

The following is an example of the upgrade package:



Step 3 Insert the FAT32 format SD card (SDIO0) or USB flash drive that stores the upgrade package, press the **UPDATE** button to start upgrading the board.

----End

NOTE

- The SD card or USB flash drive must be formatted with the FAT32 file system before saving the mass production package to it.
- The names of images in the upgrade package must be consistent with those in **bootargs**. For example, the kernel image must be named **sample.bin** because it is **9M(sample.bin)** in **bootargs**.
- If the image file does not exist, the corresponding item will not be upgraded.
- Multiple file system images can be upgraded.
- The config file is not required if you want to continue to use the environment variables **bootargs** and **bootcmd** in the U-Boot.
- The name of the yaffs image file must contain the yasffs character string, and that of any other image file cannot contain the yasffs character string.
- The name of the ext4 image file must contain the ext4 character string, and that of any other image file cannot contain the ext4 character string.
- The **bootcmd** settings of Huawei LiteOS are different from those of Linux. You can run the following commands to configure **bootcmd** settings on Huawei LiteOS:

  SPI NOR flash: **setenv bootcmd 'sf probe 0;sf read 0x40000000 0x100000 0x400000;go 40000000'**

  SPI NAND flash: **setenv bootcmd 'nand read 0x40000000 0x100000 0x400000;go 40000000'**

2.4 Example

- Burn the supported U-Boot image to the flash memory (SPI NOR flash or NAND flash) or directly download the image to the memory.
- Format the SD card or USB flash drive with the FAT32 file system.
- Copy the upgrade package created in section 2.3 "Procedure" to the formatted SD card or USB flash drive. For details about how to format the SD card or USB flash drive, see the *Peripheral Driver Operation Guide*.
- Power on the development boot. The U-Boot starts and begins automatic upgrade.
- The following is an example of information displayed during the upgrade process (using SPI NOR flash as an example):

```
// Read the upgrade configuration file.
reading config
[0]=u-boot.bin    start=0x00000000 end=0x000fffff size=0x00100000
[1]=kernel        start=0x00100000 end=0x009fffff size=0x00900000
[2]=rootfs.jffs2  start=0x00a00000 end=0x01afffff size=0x01000000
// Read and burn u-boot.bin.
reading u-boot.bin
spinor erase...
spinor write...
// Read and burn the kernel.
reading kernel
spinor erase...
spinor write...
// Read and burn rootfs.jffs2.
reading rootfs.jffs2
spinor erase...
spinor write...
// Save the environment variables.
Erasing SPI flash, offset 0x00080000 size 256K ...done
Writing to SPI flash, offset 0x00080000 size 256K ...done
// The new system is automatically started.
```

2.5 Precautions

- The SD card or USB flash drive must be formatted with the FAT32 file system.
- If the SD card or USB flash drive has multiple partitions, the upgrade package must be stored in the first partition; otherwise, it cannot be detected.
- The **bootargs** and **bootcmd** environment variables in the **config** file are automatically saved during the upgrade of **u-boot.bin**.