Hi3518A/Hi3518C/Hi3518E/Hi3516C Peripheral Driver

Operation Guide

Issue 01

Date 2014-02-26

HiSilicon Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://www.hisilicon.com

Email: support@hisilicon.com

About This Document

Purpose

This document describes how to manage the peripherals connecting to the modules that have the secure digital input/output (SDIO), ETH, or universal serial bus (USB) 2.0 host driver installed. It covers the following topics, namely, preparations, operation procedures, precautions to be taken during operation, and operation instances.

Related Version

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

Product Name	Version
Hi3518A	V100
Hi3518C	V100
Hi3518E	V100
Hi3516C	V100

Intended Audience

This document is intended for:

- Technical support personnel
- Software development engineers

Change History

Changes between document issues are cumulative. Therefore, the latest document issue contains all changes made in previous issues.

Issue 01 (2014-02-26)

This issue is the first official release.

The descriptions of the Hi3518E are added.

Issue 00B20 (2012-12-26)

This issue is the second draft release.

The descriptions of the Hi3516C are added.

Issue 00B10 (2012-08-30)

This issue is the first draft release.

Contents

Figures

1 Operation Guide to the SD/MMC

NOTE

The principles and operations for the Hi3518A, Hi3518C, Hi3518E and Hi3516C are almost the same. The following uses the Hi3518A as an example to describe chip operations.

1.1 Preparations

Before using the secure digital (SD) card or multimedia card (MMC), ensure that the following items are available:

- Fastboot and Linux kernel released in the SDK
- Local file system yaffs2, jffs2, or SquashFS released in the SDK or mounted to the NFS
- .ko files

1.2 Procedure

Before you use the SD card or MMC, perform the following steps:

Step 1 Start the board, and load the local file system yaffs2, jffs2, or SquashFS or mount the local file system to the NFS.

Step 2 Load the kernel. By default, all drivers related to the SD card or MMC are compiled in the kernel. Therefore, you do not need to load the drivers. The drivers related to the SD card or MMC are as follows:

- Drivers related to the file system and storage devices
  - nls_base
  - nls_cp437
  - fat
  - vfat
  - msdos
  - nls_iso8859-1
  - nls_ascii
- SD or MMC drivers

- mmc_core
- himci
- mmc_block

Step 3 Insert the SD card or MMC. Then you can perform operations on the SD card or MMC. For details, see section 1.3 "Operation Instance."

----End

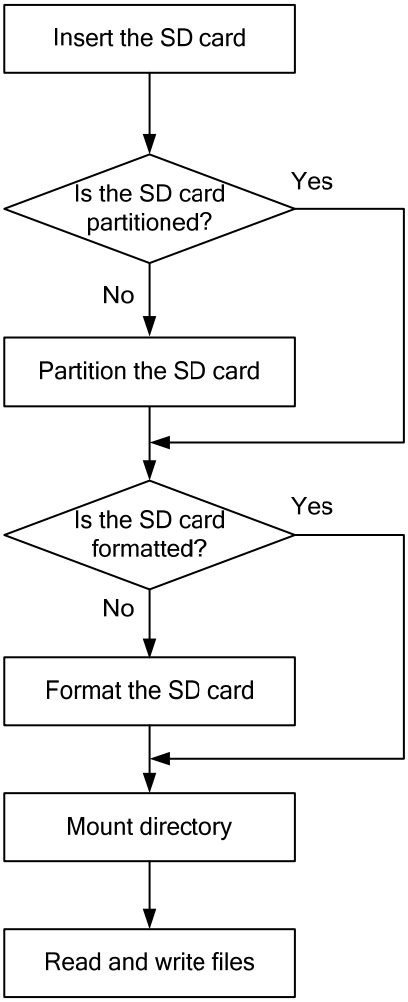
1.3 Operation Instance

The following instance describes how to read/write to an SD card by using the SDIO interface. The procedure for reading/writing to an MMC is similar that of an SD card and therefore not described in this document.

Flowchart

Figure 1-1 shows the read/write flowchart.

Figure 1-1 Read/write flowchart



Initialization

Perform the following initialization steps after an SD card or MMC is inserted:

NOTE
In the commands mentioned below, **X** indicates the partition number that is assigned when you partition the SD card or MMC by running the **fdisk** command.

Step 1 Check whether the SD or MMC is partitioned.

- If **p1** is not displayed, the SD card or MMC is not partitioned. In this case, run the **~ $ fdisk /dev/mmcblk0 command** to partition the SD card or MMC. For details, see section 4.1 "Partitioning a Storage Device."
- If **p1** is displayed, the SD card or MMC is detected and partitioned. In this case, go to Step 2.

Step 2 Check whether the SD card or MMC is formatted.

- If it is not formatted, run the **~ $ mkdosfs –F 32 /dev/mmcblk0pX command** to format it. For details, see section 4.2 "Formatting a Partition."
- If it is formatted, go to Step 3.

Step 3 Check whether a directory is mounted. For details, see section 4.3 "Mounting a Directory."

- If no directory is mounted, run the **~ $ mount -t vfat /dev/mmcblk0pX /mnt** command to mount a directory.
- If a directory is mounted, go to Step 4.

Step 4 Read/write to the SD card or MMC. For details, see section 4.4 "Reading/Writing to a File."

----End

1.4 Precautions

Note the following points during operation:

- Ensure that the metal sheet of the card is in good contact with the card slot; otherwise, the card fails to be detected or an error occurs in reading/writing to the card. When testing a thin MMC, hold the communication end of the slot if necessary.
- Before reading or writing to an SD card, ensure that the SD card is partitioned by running the **fdisk** command and formatted to the VFAT format by running the **mkdosfs** command. For details, see section 1.3 "Operation Instance."
- You need to run the **mount** command to mount an SD card to a file system so that you can read/write to the SD card. After you remove an SD card, no matter correctly or incorrectly, you need to release the mount node by running the **umount** command; otherwise, the partitions of the SD card fail to be detected after the card is inserted again. You are recommended to release the mount node before removing an SD card.
- Do not remove an SD card during read/write. Otherwise, exception information is displayed, and the files in the SD card or the file system may be damaged.
- The mount node cannot be released in the following cases:

- The current directory is the mounted directory, **mnt** for example. In this case, you need to go to another directory and run the **umount** command to release the mount node.
- The mounted directory read/write process is still running. In this case, end the process and then release the mount node.

Take the following measures if exceptions occur during operation:

- If a card is removed incorrectly during a cyclic test, press **Ctrl**+**C** to roll back to the shell; otherwise, misoperation information is reported repeatedly.
- If you reseat a card rapidly, the card may fail to be detected. This is because the registration and deregistration processes take a period of time.
- If a card is removed incorrectly, run the **umount** command to release the mount node. Otherwise, the mount node directory, **/mnt** for example, cannot be read or written and misoperation information is displayed.
- If an SD card has multiple partitions, you can run the **mount** command to switch to another partition for mounting. All mounted partitions are released only if the number of times of running the **umount** command equals that of running the **mount** command.
- If the file system is damaged due to read/write operations or other exceptions, the file system may be panic after you reinsert the card, mount a folder, and then read/write to the card. In this case, run the **umount** command, reseat the SD card, and then run the **mount** command.

2 Operation Guide to the ETH Module

NOTE
The following addresses are examples only. You need to configure the addresses as required.

2.1 Operation Instance

Note the following points when using the network interface under the kernel:

- By default, all drivers related to the ETH module are compiled in the kernel. Therefore, you do not need to load the drivers.
- You can configure the IP address and subnet mask by running the following command:
  ```
  ifconfig eth0 xxx.xxx.xxx.xxx netmask xxx.xxx.xxx.xxx up
  ```
- You can set the default gateway by running the following command:
  ```
  route add default gw xxx.xxx.xxx.xxx
  ```
- You can mount to the NFS by running the following command:
  ```
  mount -t nfs -o nolock xxx.xxx.xxx.xxx:/your/path /mount-dir
  ```
- You can upload or download files over TFTP in the shell.
  Ensure that the TFTP service software is running on the server.
  - To download a file, run the **tftp -r XX.file serverip –g** command.
    Where, **XX.file** is the file to be downloaded, and **serverip** is the IP address of the server where the file to be downloaded is located.
  - To upload a file, run the **tftp -l xx.file remoteip -p** command.
    Where, **xx.file** is the file to be uploaded, and **remoteip** is the IP address of the server that the file is uploaded to.

2.2 Precautions

If the memory allocated for the network is sufficient, you can perform the following setting in the shell:

```
echo 3000 > /proc/sys/vm/min_free_kbytes
```

2.3 IPv6 Description

The IPv6 function in the release package is disabled by default. If the IPv6 is supported, you need to modify the kernel options and recompile the kernel. Perform the following operations:

```
hisilicon$cd osdrv/linux-3.0.y
hisilicon$cp arch/arm/configs/hi3518a_full_defconfig .config
hisilicon$make ARCH=arm CROSS_COMPILE=arm-hisiv100nptl-linux- menuconfig
```

NOTE

- Use the **hi3518c_full_defconfig** file to compile the Hi3518C kernel, the **hi3518e_full_defconfig** file to compile the Hi3518e kernel, and the **hi3516c_full_defconfig** file to compile the Hi3516C kernel.
- This document describes how to set **CROSS_COMPILE** by using the arm-hisiv100nptl-linux- (uClibc) tool chain as an example:

  arm-hisiv100nptl-linux- tool chain (uClibc): CROSS_COMPILE=arm-hisiv100nptl-linux-

  If you want to use the arm-hisiv200-linux- tool chain (glibc), set **CROSS_COMPILE** as follows: CROSS_COMPILE=arm-hisiv200-linux-
- The Uclibc tool chain supports both full version and tailored version, whereas the Glibc tool chain supports only the full version.

  Full version: hi3518X_full_defconfig

  Tailored version: hi3518X_mini_defconfig

  This document uses the full version hi3518X_full_defconfig as an example. To use the tailored version, change the parameter.

Go to the following directories and configure the options, as shown in Figure 2-1.

```
[*] Networking support  --->
    Networking options  --->
    <*>   The IPv6 protocol  --->
```

Figure 2-1 Configuration diagram of the IPv6 protocol

```
-*- The IPv6 protocol
[*]   IPv6: Privacy Extensions (RFC 3041) support
[*]   IPv6: Router Preference (RFC 4191) support
[ ]     IPv6: Route Information (RFC 4191) support (EXPERIMENTAL)
[ ]   IPv6: Enable RFC 4429 Optimistic DAD (EXPERIMENTAL)
<*>   IPv6: AH transformation
<*>   IPv6: ESP transformation
<*>   IPv6: IPComp transformation
< >   IPv6: Mobility (EXPERIMENTAL)
<*>   IPv6: IPsec transport mode
<*>   IPv6: IPsec tunnel mode
<*>   IPv6: IPsec BEET mode
< >   IPv6: MIPv6 route optimization mode (EXPERIMENTAL)
<*>   IPv6: IPv6-in-IPv4 tunnel (SIT driver)
[ ]     IPv6: IPv6 Rapid Deployment (6RD) (EXPERIMENTAL)
<*>   IPv6: IP-in-IPv6 tunnel (RFC2473)
[*]   IPv6: Multiple Routing Tables
[ ]     IPv6: source address based routing
[ ]   IPv6: multicast routing (EXPERIMENTAL)
```

The configurations of the IPv6 are as follows:

- Configure the IP address and subnet mask by running the following command:
  ```
  hisilicon$ ifconfig eth0 add <ipv6address>
  ```
- Configure the default gateway by running the following command:
  ```
  hisilicon$route -A inet6 add <ipv6network>/<prefixlength> gw
  ```
- Ping a website by running the following command:
  ```
  hisilicon$ ping6 -I eth0 <ipv6address>
  ```

3 Operation Guide to the USB 2.0 Module

3.1 Preparations

Before using the USB 2.0 module, ensure that the following items are available:

- Fastboot and Linux kernel released in the SDK
- NFS or the local file system yaffs2, jffs2, or SquashFS (jffs2 recommended)

3.2 Procedure

The operation procedure is as follows:

Step 1 Start the board, and load the NFS or the file system yaffs2, jffs2, or SquashFS.

By default, all drivers related to the USB 2.0 module are compiled in the kernel. Therefore, you do not need to load the drivers.

Step 2 Insert a USB device such as the USB flash drive, mouse, or keyboard, and then perform operations on the USB device. For details, see section 3.3 "Operation Instances."

The drivers related to USB are as follows:

- Drivers related to the file system and storage devices
  - vfat
  - scsi_mod
  - sd_mod
  - nls_ascii
  - nls_iso8859-1
- Drivers related to the keyboard
  - evdev
  - usbhid
- Drivers related to the mouse
  - mousedev
  - usbhid
  - evdev

- Drivers related to the USB protocol module
  - ohci-hcd
  - ehci-hcd
  - usb-storage
  - hiusb-3518

----End

3.3 Operation Instances

3.3.1 Operation Instance Related to the USB Flash Drive

Inserting and Detecting a USB Flash Drive

Insert a USB flash drive, and then check whether it can be detected.

If the USB flash drive is detected, the following information is displayed:

```
~ $ usb 1-1: new high speed USB device using hiusb-ehci and address 2
scsi0 : usb-storage 1-1:1.0
scsi 0:0:0:0: Direct-Access     Kingston DT 101 G2       1.00 PQ: 0 ANSI:
2
sd 0:0:0:0: [sda] 62545024 512-byte logical blocks: (32.0 GB/29.8 GiB)
sd 0:0:0:0: Attached scsi generic sg0 type 0
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Assuming drive cache: write through
sd 0:0:0:0: [sda] Assuming drive cache: write through
 sda: sda1
sd 0:0:0:0: [sda] Assuming drive cache: write through
sd 0:0:0:0: [sda] Attached SCSI removable disk
sd 0:0:0:0: [sda] Assuming drive cache: write through
 sda:
sd 0:0:0:0: [sda] Assuming drive cache: write through
 sda: sda1
```

Where, **sda1** is the first partition of the USB flash drive or the portable drive. If there are multiple partitions, information such as sda1, sda2, sda3, … and sdaN is displayed.

Using the USB Flash Drive

Perform the following initialization steps after the related drivers are loaded:

NOTE

In the commands mentioned below, **X** indicates the partition number that is assigned when you partition a USB flash drive by running the **fdisk** command.

Step 1 Check whether the USB flash drive is partitioned.

- If **sda1** is not displayed, the USB flash drive is not partitioned. In this case, run the **~ $ fdisk /dev/sda** command to partition it. For details, see section 4.1 "Partitioning a Storage Device", and then go to Step 2.
- If **sda1** is displayed, the USB flash drive is detected and partitioned. In this case, go to Step 2.

Step 2 Check whether the USB flash drive is formatted.

- If it is not formatted, run the **~ $ mkdosfs –F 32 /dev/sdaX command** to format it. For details, see section 4.2 "Formatting a Partition."
- If it is formatted, go to Step 3.

Step 3 Check whether a directory is mounted.

- If no directory is mounted, run the **~ $ mount -t vfat /dev/sdaX /mnt** command to mount a directory. For details, see section 4.3 "Mounting a Directory."
- If a directory is mounted, go to Step 4.

Step 4 Read/write to the USB flash drive. For details, see section 4.4 "Reading/Writing to a File."

----End

3.3.2 Using the Keyboard

Before you can use the keyboard, you need to perform the following steps:

Step 1 Load the drivers related to the keyboard.

After the drivers related to the keyboard are loaded, the **event0** node is generated in **/dev/**.

Step 2 Receive the keyboard inputs by running the following command:

```
cat  /dev/ event0
```

Step 3 Press any keys on the keyboard.

If no error occurs, the content that you entered is displayed on the screen.

----End

3.3.3 Using the Mouse

Before you can use the mouse, perform the following steps:

Step 1 Load the drivers related to the mouse.

After the drivers related to the mouse are loaded, the **mouse0** node is generated in **/dev/**.

Step 2 Run a standard test program (mev recommended) of the gpm tool.

Step 3 Click randomly on the screen or move the pointer.

If the mouse functions properly, the corresponding code value is displayed.

----End

3.3.4 Using the USB Wi-Fi Device

Before you can use the USB WiFi device, perform the following steps:

Step 1 Start the board, and load the local file system yaffs2, jffs2, or SquashFS or mount the local file system to the NFS.

Step 2 Compile the driver of the WiFi device, and load the driver by running **insmod**. The drivers related to USB and WiFi protocol stack are compiled to the kernel. Therefore, you do not need to load the drivers.

CAUTION

You need to provide the driver of the WiFi device. In this example, the RT2870 USB WiFi driver provided by the Wi-Fi vendor is used.

Step 3 Configure the firmware.

You need to store the firmware files related to Wi-Fi in the corresponding directories of the file system.

```
cp  rt2870.bin  /lib/firmware/
cp  RT2870STA.dat  /etc/Wireless/RT2870STA/
```

Where, **rt2860.bin** and **RT2870STA.dat** are provided by the WiFi device vendor.

After the preceding operations, the USB WiFi device can be detected by the system. If the **ra0** device is displayed after you run the shell command **ifconfig –a**, the RT2870 USB-WiFi device is detected successfully.

CAUTION

The storage paths of the firmware files vary according to the WiFi devices provided by various vendors. For details, see the description of the WiFi driver.

Step 4 Use the wireless_tools.

After you configure the wireless_tools, APs can be used and communication over the wireless network can be implemented. In the following commands, **iwconfig** is used to configure the wireless network adapter, and **iwlist** is used to search for a wireless network.

NOTE

You need to obtain wireless_tools from the WiFi device vendor.

```
ifconfig ra0 192.168.1.1 netmask 255.255.254.0 up        /*Start the network*/
iwlist ra0 scanning                    //Search for a wireless network if the network name is unknown.*/
iwconfig ra0 essid dlink               /*Set the wireless network name dlink.*/
iwconfig ra0 key 1234567890            /*Set the access password.*/
```

----End

3.4 Precautions

Note the following points when performing the operations related to the USB 2.0 module:

- You need to run the **mount** command, operate a file, and then run the **umount** command in sequence each time. This avoids exceptions in the file system.
- The drivers of the keyboard and mouse must work with the upper layer. For example, mouse events are displayed on the graphical user interface (GUI) of the upper layer. You only need to operate the keyboard by accessing the event node in **/dev**. However, standard libraries are required for mouse operations.
- Mouse application libgpm libraries are provided in Linux. If you need to use the mouse, these libraries must be compiled. You are recommended to use the standard kernel interface gpm-1.20.5 that has passed the test.

  In addition, a set of test programs (such as mev) are provided in the gpm tool. You can perform encoding by using the test programs, making the development easier.

4 Appendix

4.1 Partitioning a Storage Device

Perform the following steps:

Step 1 Type the following command at the command prompt of the console:

```
~ $ fdisk device node
```

NOTE

The device node depends on the type of the actual storage device. For details, see the operation instances in preceding chapters.

Step 2 Press **Enter**.

Step 3 Type **m** at the command prompt, and perform operations as prompted.

----End

4.1.1 Checking the Current Partition Status of a Storage Device

To check the current partition status of a storage device, enter the following command at the command prompt of the console:

```
Command (m for help): p
```

If the following information is displayed on the console, the storage device is not partitioned:

```
Disk /dev/mmc/blk1/disc: 127 MB, 127139840 bytes
8 heads, 32 sectors/track, 970 cylinders
Units = cylinders of 256 * 512 = 131072 bytes
Device Boot Start End Blocks Id System
```

In this case, you need to partition the device by following the description in section 4.1.2 "Creating Partitions for a Storage Device", and save the partition information by following the description in section 4.1.3 "Saving the Partition Information."

4.1.2 Creating Partitions for a Storage Device

Perform the following steps:

Step 1 Create partitions by entering the following command at the command prompt:

```
Command (m for help): n
```

The following information is displayed on the console:

```
Command action
e extended
p primary partition (1-4)
```

Step 2 Create the primary partition by running the following command:

```
p
```

Step 3 Set the number of partitions by entering a number (for example, **1**).

```
Partition number (1-4): 1
```

The following information is displayed on the console:

```
First cylinder (1-970, default 1):
```

Step 4 Select the start cylinder by typing a number, and then press **Enter**. In this example, the default value **1** is retained.

```
Using default value 1
```

Step 5 Select the end cylinder by typing a number, and then press **Enter**. In this example, the default value **970** is retained.

```
Last cylinder or +size or +sizeM or +sizeK (1-970, default 970):
Using default value 970
```

Step 6 Select a data format for the storage device.

The default file system is Linux. The Win95 FAT file system is to be used in this example. You need to run the following commands:

```
Command (m for help): t
Selected partition 1
```

Then run the following command:

```
Hex code (type L to list codes): b
```

To view the details about all partitions of the Win95 FAT file system, run the following command:

```
Changed system type of partition 1 to b (Win95 FAT32)
```

Step 7 Check the information about partitions by running the following command:

```
Command (m for help): p
```

If the partition information is displayed on the console, the partitioning is successful.

----End

4.1.3 Saving the Partition Information

To save the partition information, run the following command:

```
Command (m for help): w
```

If the following information is displayed on the console, the partition information is successfully saved:

```
The partition table has been altered!
Calling ioctl() to re-read partition table.
.............
~ $
```

4.2 Formatting a Partition

You can format a partition by running the following command:

```
~ $ mkdosfs -F 32 Partition name
```

NOTE

The name of a partition depends on the type of the actual storage device. For details, see the operation instances in the preceding chapters.

If no error message is displayed on the console, a partition is formatted successfully:

```
~ $
```

4.3 Mounting a Directory

Before reading/writing to a file, you must mount a partition to the **mnt** directory by running the following command:

```
~ $ mount -t vfat Partition name /mnt
```

NOTE

The name of a partition depends on the type of the actual storage device. For details, see the operation instances in the preceding chapters.

4.4 Reading/Writing to a File

You can read/write to a file in various manners. In this section, the **cp** command is used to read/write to a file.

To write to a file, copy the **test.txt** file in the current directory to the **mnt** directory of a storage device by running the following command:

```
~ $ cp ./test.txt /mnt
```