



Peripheral Driver

Operation Guide

Issue 04

Date 2013-06-21

Copyright © HiSilicon Technologies Co., Ltd. 2012-2013. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of HiSilicon Technologies Co., Ltd.

Trademarks and Permissions



HISILICON

, and other HiSilicon icons are trademarks of HiSilicon Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

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 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), gigabit media access control (GMAC), ETH, universal serial bus (USB) 2.0 Host, and (SATA) 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
Hi3521	V100
Hi3520A	V100
Hi3520D	V100
Hi3515A	V100
Hi3515C	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 04 (2013-06-21)

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

The descriptions related to the Hi3515C are added.

Issue 03 (2013-05-09)

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

Chapter 2 Operation Guide to the GMAC/ETH Module

The note "The FE PHY address for the Hi3520D must be set to 3" is added.

Issue 02 (2013-04-03)

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

The descriptions related to the Hi3515A are added.

Issue 01 (2013-02-28)

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

Chapter 2 Operation Guide to the GMAC/ETH Module

In section 2.1, the descriptions related to the Hi3520D are added.

Section 2.3 "IPv6" is added.

Issue 00B02 (2012-06-08)

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

Chapter 3 Operation Guide to the GMAC Module

The descriptions of the GMAC module and TOE function are added.

Issue 00B01 (2012-04-20)

This issue is the draft release.



Contents

About This Document.....	i
1 Operation Guide to the SD/MMC	1
1.1 Preparations	1
1.2 Procedure	1
1.3 Operation Instance	2
1.4 Precautions	4
2 Operation Guide to the GMAC/ETH Module.....	5
2.1 Operation Instance	5
2.2 Precautions	6
2.3 IPv6	6
3 Operation Guide to the USB 2.0 Module	8
3.1 Preparations	8
3.2 Procedure	8
3.3 Operation Instances	9
3.3.1 Operation Instance Related to the USB Flash Drive	9
3.3.2 Using the Keyboard	10
3.3.3 Using the Mouse	10
3.3.4 Using the USB WiFi Device	11
3.4 Precautions	12
4 Operation Guide to the SATA Module	13
4.1 Preparations	13
4.2 Procedure	13
4.3 Operation Instance	14
4.4 Precautions	14
5 Appendix	15
5.1 Partitioning a Storage Device	15
5.1.1 Checking the Current Partition Status of a Storage Device	15
5.1.2 Creating Partitions for a Storage Device	15
5.1.3 Saving the Partition Information	16
5.2 Formatting a Partition	17
5.3 Mounting a Directory	17



5.4 Reading/Writing to a File	17
-------------------------------------	----



Figures

Figure 1-1 Read/write flowchart	3
Figure 2-1 Configuration options of the IPv6 protocol	7



1 Operation Guide to the SD/MMC



CAUTION

The Hi3520D, Hi3515A or Hi3515C does not support the secure digital (SD) card and the multimedia card (MMC).

1.1 Preparations

Before using the SD card or MMC, ensure that the following items are available:

- Fastboot and Linux kernel released in the SDK
- Local file system Yaffs2, Jffs2, or Cramfs released in the SDK or mounted to the network file system (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 Cramfs 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

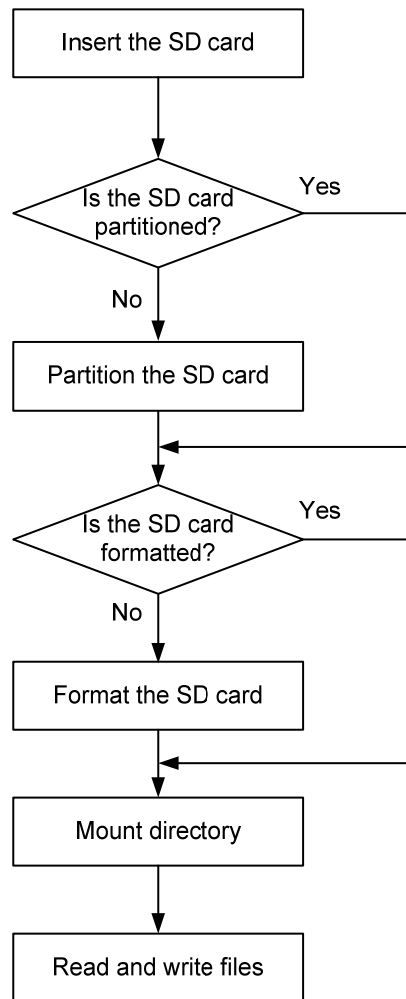
The following instance describes how to read/write to an SD card by using the SDIO interface. The procedure for reading/writing to an MMC is similar that of an SD card and therefore not described in this document.

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 following commands, **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 5.1 "[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 5.2 "[Formatting a Partition](#)."



- If it is formatted, go to [Step 3](#).

Step 3 Check whether a directory is mounted. For details, see section [5.3 "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 [5.4 "Reading/Writing to a File."](#)

----End

1.4 Precautions

Note the following 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 GMAC/ETH Module



NOTE

- The Hi3521 and Hi3520A use the GMAC, whereas the Hi3520D, Hi3515A and Hi3515C uses the ETH.
- The following addresses are examples only. You need to configure the addresses as required.
- The address for the Hi3520D FE PHY must be set to **3** to ensure that the Ethernet indicator works properly, because the FE PHY address line is shared with the signal line of the Ethernet indicator.

2.1 Operation Instance



NOTE

The ETH drivers for the Hi3520D, Hi3515A and Hi3515C are compiled in the kernel by default. Therefore, you can configure IP addresses without loading ETH drivers.

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

- The GMAC driver **stmmac.ko** is saved in **/hitoe/**. You need to load it based on the application scenario.
 - To disable the TCP offload engine (TOE), run **insmod /hitoe/stmmac.ko**.
 - To enable the TOE, run **insmod /hitoe/stmmac.ko hitoe=1**.

The preceding command is compiled in the **S81toe** script that is saved in **/ect/init.d/**. You can modify the command in the **S81toe** script. The TOE is disabled by default. For details, see the script descriptions.

To enable the GMAC driver to be automatically loaded during startup, add the **S81toe** script in the **rcS** boot file in **/etc/init.d/**.

The following describes the TOE functions:

- The TOE enables the TCP, IP, UDP, or ICMP protocol stack to be processed on a high-speed device. The network adapter with the TCP/IP function can be developed by using the field-programmable gate array (FPGA) or application-specific integrated circuit (ASIC). This reduces the host CPU usage and increases the network transfer speed and host CPU efficiency.
- The Hi3521 uses part of TOE functions. That is, the Hi3531 uses the TOE function during data transfer based on the TCP. The UDP or ICMP protocol stack is still processed in the same way as standard protocol stacks.



- 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

When the Hi3521/Hi3520A GMAC module transmits data by using the TOE, a large memory with consecutive addresses is required. However, such memory is rare. Therefore, ensure that the physical memory is sufficient before data transmission. The following is an example only. The values vary depending on the application scenario.

```
echo 8192 > /proc/sys/vm/min_free_kbytes
echo 200 > /proc/sys/vm/vfs_cache_pressure
```

Vfs_cache_pressure indicates that the virtual memory for directory and i-node will be recycled. The default value is 100, and the recommended value is 200. The greater the value is, the larger the recycled memory. That is, a larger memory is provided for the TOE.

If the ETH port has insufficient memory space when the Hi3520D, Hi3515A or Hi3515C ETH is used, perform the following settings in the shell:

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

2.3 IPv6

The IPv6 function is disabled in the release package by default. If the IPv6 function is required, you need to modify the kernel options and recompile the kernel as follows:

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



NOTE

XXX_defconfig indicates godarm_defconfig (Hi3521), godcare_defconfig (Hi3520A), or hi3520d_full_defconfig (Hi3520D/Hi3515A/Hi3515C).

There are two cases for CROSS_COMPILE=arm-hisiXXX-linux-:

- Hi35xx_V100R001C01SPCxxx corresponds to uclibc. If the uclibc tool chain is used, CROSS_COMPILE=arm-hisiv100nptl-linux-.
- Hi35xx_V100R001C02SPCxxx corresponds to glibc. If the glibc tool chain is used, CROSS_COMPILE=arm-hisiv200-linux-.

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 options 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 IPv6 configurations 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 Cramfs (Jffs2 is 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 Cramfs.

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-godeyes

----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 5.1 "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 5.2 "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 5.3 "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 5.4 "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 WiFi 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 Cramfs 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 inherent in the kernel 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 **wlan0** 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 wlan0 192.168.1.1 netmask 255.255.254.0 up /*Start the network*/  
iwlist wlan0 scanning //Search for a wireless network if the network name is  
unknown.*/  
iwconfig wlan0 essid dlink /*Set the wireless network name.*/  
iwconfig wlan0 key 1234567890 /*Set the access password.*/  
iwconfig wlan0 mode Managed /*Set a mode.*/  
iwconfig wlan0 channel 6 /*Specify a channel.*/
```



----End

3.4 Precautions

Note the following 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 Operation Guide to the SATA Module

4.1 Preparations

Before using the serial advanced technology attachment (SATA) module, ensure that the following items are available:

- Standard SATA hard disk
- U-boot and Linux kernel released in the SDK
- Local file system Yaffs2 or Cramfs released in the SDK or mounted to the NFS

4.2 Procedure

Before you can read/write to a SATA hard disk, you need to start the board, and load the local file system Jffs2 or Cramfs or mount the local file system to the NFS.

By default, all drivers related to the SATA module are compiled in the kernel, you do not need to run the command for loading drivers. For details, see section [4.3 "Operation Instance."](#)

The involved drivers are as follows:

- Drivers related to the file system and storage devices
 - nls_base
 - nls_cp437
 - fat
 - vfat
 - msdos
 - nls_iso8859-1
 - nls_ascii
 - scsi_mod
 - sd_mod
- Drivers related to hard disks
 - libata
 - ahci



4.3 Operation Instance

Perform the following operations:



NOTE

In the following commands, X indicates the partition number that is assigned when you partition a hard disk by running the **fdisk** command.

- Change the path in which the **fdisk** command is executed to **fdisk /dev/sda**.
- Change the path in which the formatting command **mkdosfs** is executed to **~ \$ mkdosfs -F 32 /dev/sdaX**.
- Change the mount path to **~ \$ mount -t vfat /dev/sdaX /mnt**.

Step 1 Check whether the SATA hard disk is partitioned.

- If **sda1** is not displayed, the hard disk is not partitioned. In this case, run the **~ \$ fdisk /dev/sda** command to partition it. For details, see section 5.1 "[Partitioning a Storage Device](#)", and then go to [Step 2](#).
- If **sda1** is displayed, the hard disk is detected and partitioned. In this case, go to [Step 2](#).

Step 2 Check whether the hard disk is formatted.

- If the hard disk is not formatted, run the **~ \$ mkdosfs -F 32 /dev/sdaX** command to format it. For details, see section 5.2 "[Formatting a Partition](#)."
- If it is formatted, go to [Step 3](#).

Step 3 Mount a directory. For details, see section 5.3 "[Mounting a Directory](#)."

Step 4 Read and write to the hard disk. For details, see section 5.4 "[Reading/Writing to a File](#)."

----End

4.4 Precautions

The Hi3521 SATA drivers support hot plugging. After removing a SATA hard disk, you need to unmount the device nodes mounted to the hard disk. Otherwise, the nodes of the SATA disk change after the SATA hard disk is reinserted.



5 Appendix

5.1 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 the preceding chapters.

Step 2 Press **Enter**.

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

----End

5.1.1 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 [5.1.2 "Creating Partitions for a Storage Device"](#), and save the partition information by following the description in section [5.1.3 "Saving the Partition Information."](#)

5.1.2 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

5.1.3 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.  
.....  
~ $
```

5.2 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 the following information is displayed on the console, a partition is formatted successfully:

```
mkdosfs 2.11 (12 Mar 2005)  
~ $
```

5.3 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.

5.4 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
```