



外围设备驱动 操作指南

文档版本 04

发布日期 2013-06-21

版权所有 © 深圳市海思半导体有限公司 2013。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HISILICON、海思和其他海思商标均为深圳市海思半导体有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

深圳市海思半导体有限公司

地址： 深圳市龙岗区坂田华为基地华为电气生产中心 邮编：518129

网址： <http://www.hisilicon.com>

客户服务电话： +86-755-28788858

客户服务传真： +86-755-28357515

客户服务邮箱： support@hisilicon.com



前 言

概述

本文档主要是指导使用 SDIO、GMAC、ETH、USB 2.0 Host 和 SATA 等驱动模块的相关人员，通过一定的步骤和方法对和这些驱动模块相连的外围设备进行控制，主要包括操作准备、操作过程、操作中需要注意的问题以及操作示例。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
Hi3521 芯片	V100
Hi3520A 芯片	V100
Hi3520D 芯片	V100
Hi3515A 芯片	V100
Hi3515C 芯片	V100

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。



修订日期	版本	修订说明
2013-06-21	04	添加 Hi3515C 的网络使用方法。
2013-05-09	03	第 2 章 GMAC/ETH 操作指南 增加"Hi3520D 中 FE PHY 的地址需要设置为 3"的说明。
2013-04-03	02	添加 Hi3515A 的网络使用方法。
2013-02-05	01	添加 Hi3520D 的网络使用方法。
2012-06-08	00B02	第 2 章 GMAC 操作指南 修改如下内容： 1、在 GMAC 操作说明中添加 TOE 的相关说明； 2、在 GMAC 操作注意事项中添加使用 TOE 时相关的注意事项。
2012-04-20	00B01	第 1 次临时版本发布。



目 录

1 SD/MMC 卡操作	1
1.1 操作准备	1
1.2 操作过程	1
1.3 操作示例	2
1.4 操作中需要注意的问题	4
2 GMAC/ETH 操作指南	5
2.1 操作示例	5
2.2 操作中需要注意的问题	6
2.3 IPv6 说明	6
3 USB 2.0 操作指南	8
3.1 操作准备	8
3.2 操作过程	8
3.3 操作示例	9
3.3.1 U 盘操作示例	9
3.3.2 键盘操作示例	10
3.3.3 鼠标操作示例	10
3.3.4 USB-WiFi 操作示例	10
3.4 操作中需要注意的问题	12
4 SATA 操作指南	13
4.1 操作准备	13
4.2 操作过程	13
4.3 操作示例	14
4.4 操作中需要注意的问题	14
5 附录	15
5.1 用 fdisk 工具分区	15
5.1.1 查看当前状态	15
5.1.2 创建新的分区	15
5.1.3 保存分区信息	17
5.2 用 mkdosfs 工具格式化	17



5.3 挂载目录.....	17
5.4 读写文件.....	17



插图目录

图 1-1 在控制台下实现读写 SD 卡的操作示例.....	3
图 2-1 IPv6 Protocol 配置示意图.....	7



1 SD/MMC 卡操作



注意

Hi3520D/Hi3515A/Hi3515C 不支持 SD/MMC 卡

1.1 操作准备

SD/MMC 卡的操作准备如下：

- U-boot 和 Linux 内核使用 SDK 发布的 U-boot 和 kernel。
- 文件系统。
可以使用 SDK 发布的本地文件系统 yaffs2、jffs2 或 cramfs，也可以通过本地文件系统再挂载到 NFS。
- ko 文件。

1.2 操作过程

操作过程如下：

1. 启动单板，加载本地文件系统 yaffs2、jffs2 或 cramfs，也可以通过本地文件系统进一步挂载到 NFS。
2. 加载内核。默认 SD/MMC 相关模块已全部编入内核，不需要再执行加载命令。下面列出 SD/MMC 所有相关驱动：
 - 文件系统和存储设备相关模块
 - nls_base
 - nls_cp437
 - fat



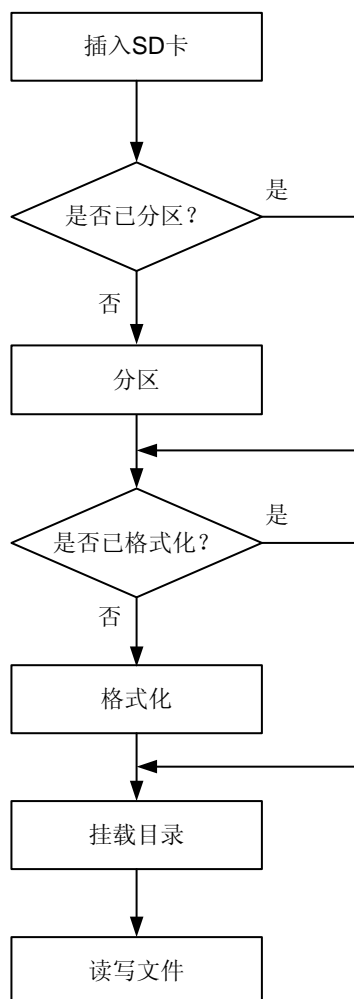
- vfat
 - msdos
 - nls_iso8859-1
 - nls_ascii
 - SD/MMC 相关模块
 - mmc_core
 - himci
 - mmc_block
3. 插入 SD/MMC 卡，就可以对 SD/MMC 卡进行相关的操作。具体操作请参见“[1.3 操作示例](#)”。
- 结束

1.3 操作示例

此操作示例通过 SDIO 接口实现对 SD 卡的读写操作，MMC 卡的读写操作和 SD 卡类似，这里不再举例。在控制台下实现读写 SD 卡的操作示例如[图 1-1](#) 所示。



图1-1 在控制台下实现读写 SD 卡的操作示例



初始化及应用，待 SD/MMC 卡插入后，进行如下操作：



说明

其中 X 为分区号，由 fdisk 工具分区时决定。

- 命令 fdisk 操作的具体目录需改为：~\$ fdisk /dev/mmcblk0
- 用 mkdosfs 工具格式化的具体目录需改为：~\$ mkdosfs -F 32 /dev/mmcblk0pX
- 挂载的具体目录需改为：~\$ mount -t vfat /dev/mmcblk0pX /mnt

1. 查看分区信息。

- 若没有显示出 p1，表示还没有分区，请参见“5.1 用 fdisk 工具分区”进行分区后，进入 2。
- 若有分区信息 p1，则 SD/MMC 卡已经检测到，并已经进行分区，进入 2。

2. 查看格式化信息。

- 若没有格式化，请参见“5.2 用 mkdosfs 工具格式化”进行格式化后，进入 3。
- 若已格式化，进入 3。



3. 挂载目录，请参见“[5.3 挂载目录](#)”。
4. 对 SD/MMC 卡进行读写操作，请参见“[5.4 读写文件](#)”。

----结束

1.4 操作中需要注意的问题

在正常操作过程中需要遵守的事项：

- 保证卡的金属片与卡槽硬件接触充分良好（如果接触不好，会出现检测错误或读写数据错误），测试薄的 MMC 卡，必要时可以用手按住卡槽的通讯端测试。
- 每次需要读写 SD 卡时，必须确保 SD 卡已经创建分区，并将该分区格式化为 vfat 文件系统（通过 fdisk 和 mkdosfs 命令，具体过程参见 [1.3 操作示例](#)）。
- 每次插入 SD 卡后，需要做一次 mount 操作挂载文件系统，才能读写 SD 卡；如果 SD 卡已经挂载到文件系统，拔卡后，必须做一次 umount 操作，否则，再次插入卡时就会找不到 SD 卡的分区。
- 正常拔卡后需要 umount 挂载点（建议正常的操作顺序是先 umount，再拔卡），异常拔卡后，也需要 umount 挂载点，否则再次插卡时就会找不到 SD 卡的分区。

在正常操作过程中不能进行的操作：

- 读写 SD 卡时不要拔卡，否则会打印一些异常信息，并且可能会导致卡中文件或文件系统被破坏。
- 当前目录是挂载目录如/mnt 时，不能 umount 操作，必须转到其它目录下才能 umount 操作。
- 系统中读写挂载目录的进程没有完全退出时，不能 umount 操作，必须完全结束操作挂载目录的任务才能正常 umount 操作。

在操作过程中出现异常时的操作：

- 如果在循环测试过程中异常拔卡，需要按 ctrl+c 回退出到 shell 下，否则会一直不停地打印异常操作信息。
- 拔卡后，再极其快速地再次插入卡时可能会出现检测不到卡的现象，因为卡的检测注册/注销过程需要一定的时间。
- 异常拔卡后，必须执行 umount 操作，否则不能读写挂载点目录如/mnt，并会打印异常信息。
- SD 有多分区时，可以通过 mount 操作切换挂载不同的分区，但最后 umount 操作次数与 mount 操作次数相等时，才会完全 umount 所有的挂载分区。
- 如果由于读写数据或其它异常原因，导致文件系统破坏，重新插卡并挂载，读写卡时可能会出现文件系统 panic，这时，需要 umount 操作，拔卡，再次插卡并 mount，才能正常读写 SD 卡。



2 GMAC/ETH 操作指南



说明

- Hi3521/Hi3520A 使用 GMAC；Hi3520D/Hi3515A/Hi3515C 使用 ETH。
- 以下设置的地址只是一个举例说明，具体的地址设置要根据具体使用的地址来设置。
- Hi3520D 中 FE PHY 的地址需要设置为 3（FE PHY 地址与网口灯的信号线有共用关系，设置为 3 才能保证网口灯的正确）。

2.1 操作示例



说明

Hi3520D/Hi3515A/Hi3515C 默认相关 ETH 模块已全部编入内核，不需要执行加载操作，请直接跳至配置 IP 地址步骤。

内核下使用网口的操作涉及到以下几个方面：

- GMAC 相关模块编译后存放在文件系统中，路径为/hitoe/stmmac.ko，需要执行以下命令加载模块
 - 不使能 TOE：insmod /hitoe/stmmac.ko
 - 使能 TOE：insmod /hitoe/stmmac.ko hitoe=1

上述命令已经写进启动脚本 S81toe，路径为/ect/init.d/S81toe，使用时可在此文件中修改，默认不使用 TOE 功能。详细操作请参考脚本中的相关说明。

若想开机自动加载 GMAC 模块，只需在引导文件/etc/init.d/rcS 中加入运行上述脚本的命令即可。

TOE（TCP Offload Engine-TCP 卸载引擎）功能简介：

- 分担 CPU 对 TCP/IP 协议栈的处理，将协议处理过程放到高速设备上（包括 TCP、IP、UDP、ICMP 等）完成，即使用 FPGA、ASIC 等器件研制带有 TCP/IP 功能的网卡，从而将主机 CPU 解放出来，既可提高网络传输速度，又可提高主机 CPU 的工作效率。
 - Hi3521 芯片使用到的是部分 TOE 加速，即只针对 TCP 协议上的数据传输进行处理，UDP 或者 ICMP 协议仍然使用标准协议栈处理。
- 配置 ip 地址和子网掩码

```
ifconfig eth0 xxx.xxx.xxx.xxx netmask xxx.xxx.xxx.xxx up
```



- 设置缺省网关
`route add default gw xxx.xxx.xxx.xxx`
- mount nfs
`mount -t nfs -o nolock xxx.xxx.xxx.xxx:/your/path /mount-dir`
- shell 下使用 tftp 上传下载文件
前提是在 server 端有 tftp 服务软件在运行。
 - 下载文件: `tftp -r XX.file serverip -g`
其中: XX.file:需要下载的文件, serverip 需要下载的文件所在的 server 的 ip 地址。
 - 上传文件: `tftp -l xx.file remoteip -p //xx.file:需要上传的文件, remoteip 文件需要上传到的 server 的 ip 地址。`

2.2 操作中需要注意的问题

Hi3521/Hi3520A GMAC 每次使用 TOE 发送数据时, 需要大块连续的物理内存, 但是内核中大块连续内存的数量非常少。因此在使用前需要做如下处理, 确保能够分配到足够多的物理内存。下述处理只是简单的举例, 具体的数值需要根据业务场景使用的连接数来决定。

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

vfs_cache_pressure 表示设置了虚拟内存回收 directory 和 i-node 缓冲的倾向, 缺省值为 100, 推荐设置为 200, 因为该参数的值越大, 回收的倾向越严重。这样 TOE 就会有更多的物理内存使用。Hi3520D/Hi3515A/Hi3515C ETH 使用时如果网口出现内存分配不足的情况下可以在 shell 下进行如下设置: `echo 3000 > /proc/sys/vm/min_free_kbyte`

2.3 IPv6 说明

发布包中默认关闭 IPv6 功能。如果要支持 IPv6, 需要修改内核选项, 并重新编译内核。具体操作如下:

```
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
```



说明

XXX_defconfig 表示 godarm_defconfig(Hi3521)、godcare_defconfig(Hi3520A) hi3520d_full_defconfig(3520D/Hi3515A/Hi3515C)中的某一种。

CROSS_COMPILE=arm-hisiXXX-linux- 中 XXX 表示两种情况。

- Hi35xx_V100R001C01SPCxxx 对应 uclibc，使用 uclibc 工具链时，CROSS_COMPILE=arm-hisiv100nptl-linux-。
- Hi35xx_V100R001C02SPCxxx 对应 glibc，使用 glibc 工具链时，CROSS_COMPILE=arm-hisiv200-linux-。

进入如下目录，将该页面选项配置如图 2-1 所示。

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

图2-1 IPv6 Protocol 配置示意图

```
-- 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)
```

IPv6 环境配置如下：

- 配置 ip 地址和子网掩码
hisilicon\$ ifconfig eth0 add <ipv6address>
- 设置缺省网关
hisilicon\$route -A inet6 add <ipv6network>/<prefixlength> gw
- Ping 某个网址
hisilicon\$ ping6 -I eth0 <ipv6address>



3 USB 2.0 操作指南

3.1 操作准备

USB2.0 的操作准备如下：

- U-boot 和 Linux 内核使用 SDK 发布的 U-boot 和 kernel
- 文件系统
可以使用本地文件系统 yaffs2、jffs2 或 cramfs，也可以使用 NFS，建议使用 jffs2。

3.2 操作过程

操作过程如下：

1. 启动单板，加载 yaffs2、jffs2 或 cramfs 文件系统，也可以使用 NFS。
2. 默认 USB 相关模块已经全部编入内核，不需要再执行加载命令，就可以对 U 盘、鼠标或者键盘进行相关的操作了。具体操作请参见“[3.3 操作示例](#)”。下面列出所有 USB 相关驱动：
 - 文件系统和存储设备相关模块
 - vfat
 - scsi_mod
 - sd_mod
 - nls_ascii
 - nls_iso8859-1
 - 键盘相关模块
 - evdev
 - usbhid
 - 鼠标相关模块
 - mousedev
 - usbhid
 - evdev



- USB2.0 模块
 - ohci-hcd
 - ehci-hcd
 - usb-storage
 - hiusb-godeyes

----结束

3.3 操作示例

3.3.1 U 盘操作示例

插入检测

直接插入 U 盘，观察是否枚举成功。

正常情况下串口打印为：

```
~ $ 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
```

其中：sda1 表示 U 盘或移动硬盘上的第一个分区，当存在多个分区时，会出现 sda1、sda2、sda3 的字样。

初始化及应用

模块插入完成后，进行如下操作：



说明

其中 X 为分区号，由 fdisk 工具分区时决定。

- 命令 fdisk 操作的具体目录需改为：~\$ fdisk /dev/sda
- 用 mkdosfs 工具格式化的具体目录需改为：~\$ mkdosfs -F 32 /dev/sdaX



- 挂载的具体目录需改为：`~$ mount -t vfat /dev/sdaX /mnt`

1. 查看分区信息。

- 若没有分区信息 `sda1`，表示还没有分区，请参见“[5.1 用 fdisk 工具分区](#)”进行分区后，进入 [2](#)。
- 若有分区信息 `sda1`，则已经检测到 U 盘，并已经进行分区，进入 [2](#)。

2. 查看格式化信息。

- 若没有格式化，请参见“[5.2 用 mkdosfs 工具格式化](#)”进行格式化后，进入 [1.3 3](#)。
- 若已格式化，进入 [3](#)。

3. 挂载目录，请参见“[5.3 挂载目录](#)”。

4. 对硬盘进行读写操作，请参见“[5.4 读写文件](#)”。

----结束

3.3.2 键盘操作示例

键盘操作过程如下：

1. 插入模块。

插入键盘相关模块后，键盘会在 `/dev/` 目录下生成 `event0` 节点。

2. 接收键盘输入。

执行命令：`cat /dev/event0`

然后在 USB 键盘上敲击，可以看到屏幕有输出。

----结束

3.3.3 鼠标操作示例

鼠标操作过程如下：

1. 插入模块。

插入鼠标相关模块后，鼠标会在 `/dev/` 目录下生成 `mouse0` 节点。

2. 运行 `gpm` 中提供的标准测试程序（建议使用 `mev`）。

3. 进行鼠标操作（点击、滑动等），可以看到串口打印出相应码值。

----结束

3.3.4 USB-WiFi 操作示例

USB-WiFi 操作过程如下：



1. 启动单板，加载本地文件系统 yaffs2、jffs2 或 cramfs，也可以通过本地文件系统进一步挂载到 NFS。
2. 插入 WiFi 设备驱动。除 WiFi 设备的驱动，USB 以及 WiFi 协议栈相关模块已全部编入内核，不需要再执行加载命令。



注意

由于 WiFi 设备由用户选型，因此需要用户提供具体驱动。编译好相关驱动后，直接通过 insmod 命令加载即可。本例使用内核自带的 RT2870 USB-WiFi 驱动。

3. Firmware 配置

需要将 WiFi 的 firmware 文件存放在文件系统相应目录：

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

其中 rt2870.bin 和 RT2870STA.dat 均由 WiFi 厂商提供。

完成上述操作后，插入 USB-WiFi 设备，系统即可正常识别。在 shell 下输入命令“ifconfig -a”，出现一个名为“wlan0”的设备，表示 RT2870 USB-WiFi 设备已枚举成功。



注意

不同厂商的 WiFi 设备，要求存放 firmware 的目录可能不一样，一般在厂商提供的 WiFi 驱动里有说明。

4. 无线管理工具的使用

通过无线管理工具 wireless_tools 进行配置，实现无线 AP 的链接以及无线网络的通信。其中，iwconfig 用来配置无线网卡，iwlist 用来搜索无线网络。

说明

wireless_tools 需要用户从 WiFi 厂商获取。

```
ifconfig wlan0 192.168.1.1 netmask 255.255.254.0 up /*打开网络*/  
iwlist wlan0 scanning /*不知道无线网络名时搜索网络名*/  
iwconfig wlan0 essid dlink /*指定无线网络名*/  
iwconfig wlan0 key 1234567890 /*指定访问密码*/  
iwconfig wlan0 mode Managed /*指定模式*/  
iwconfig wlan0 channel 6 /*指定信道*/  
此时完成所有网络配置。
```

----结束



3.4 操作中需要注意的问题

操作中需要注意的问题如下：

- 在操作时请尽量按照完整的操作顺序进行操作（mount→操作文件→umount），以免造成文件系统的异常。
- 目前键盘和鼠标的驱动要和上层结合使用，比如鼠标事件要和上层的 GUI 结合。对键盘的操作只需要对/dev 下的 event 节点读取即可，而鼠标则需要标准的库支持。
- 在 Linux 系统中提供了一套标准的鼠标应用接口 libgpm，如果需要是用鼠标客户可自行编译此库。在使用时建议使用内核标准接口 gpm。

已测试通过的标准接口版本：gpm-1.20.5。

另外在 gpm 中还提供了一整套的测试工具源码（如：mev 等），用户可根据这些测试程序进行编码等操作，降低开发难度。



4 SATA 操作指南

4.1 操作准备

硬件环境：SATA 测试使用标准的 SATA 硬盘。

- U-boot 和 Linux 内核使用 SDK（Software Development Kit）发布的 U-boot 和 kernel。
- 文件系统

可以使用 SDK 发布的本地文件系统 jffs2 或 cramfs，也可以通过本地文件系统再挂载到 NFS（Network File System）。

4.2 操作过程

SATA 硬盘测试步骤如下：

1. 启动单板，加载本地文件系统 jffs2 或 cramfs，也可以通过本地文件系统进一步挂载到 NFS。
2. 默认 SATA 的相关模块已经全部编入内核，不需要再执行加载命令，就可以对 SATA 硬盘进行相关的操作。具体操作请参见“4.3 操作示例”。

- 文件系统和存储设备相关模块

- nls_base
- nls_cp437
- fat
- vfat
- msdos
- nls_iso8859-1
- nls_ascii
- scsi_mod
- sd_mod

- 硬盘相关模块

- libata



- ahci
----结束

4.3 操作示例

对 SATA 硬盘进行如下操作：



说明

其中 X 为分区号，由 fdisk 工具分区时决定。

- 命令 fdisk 操作的具体目录需改为：fdisk /dev/sda。
- 用 mkdosfs 工具格式化的具体目录需改为：~\$ mkdosfs -F 32 /dev/sdaX
- 挂载的具体目录需改为：~\$ mount -t vfat /dev/sdaX /mnt

SATA 的操作步骤如下：

1. 查看分区信息：
 - 若没有分区信息 sda1，表示还没有分区，请参见“[5.1 用 fdisk 工具分区](#)”进行分区后，进入 [3.3.1 2](#)。
 - 若有分区信息 sda1，则已经检测到硬盘，并已经进行分区，进入 [3.3.1 2](#)。
2. 查看格式化信息。
 - 若没有格式化，请参见“[5.2 用 mkdosfs 工具格式化](#)”进行格式化后，进入 [1.3 3](#)。
 - 若已格式化，进入 [3.3.1 3](#)。
3. 挂载目录，请参见“[5.3 挂载目录](#)”。
4. 对硬盘进行读写操作，请参见“[5.4 读写文件](#)”。

----结束

4.4 操作中需要注意的问题

由于 Hi3521 的 SATA 驱动支持热插拔。在热拔掉后，需要 umount 硬盘所 mount 的节点。否则重新插上后，硬盘的设备节点会发生变化。



5 附录

5.1 用 fdisk 工具分区

通过 [5.1.1 查看当前状态](#)，对应以下情况选择操作：

- 若已有分区，本操作可以跳过，直接到“[5.2 用 mkdosfs 工具格式化](#)”。
- 若没有分区，则在控制台的提示符下，输入命令 `fdisk`，具体格式如下：

~ \$ `fdisk` 设备节点

回车后，输入命令 `m`，根据帮助信息继续进行以下的操作。

其中设备节点与实际接入的设备类型有关，具体名称在以上各章节的“操作示例”中均有说明。

5.1.1 查看当前状态

在控制台的提示符下，输入命令 `p`，查看当前分区状态：

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

控制台显示出分区状态信息：

```
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
```

上面信息表明设备没有分区，需要按照 [5.1.2 创建新的分区](#)和 [5.1.3 保存分区信息](#)的描述对设备进行分区。

5.1.2 创建新的分区

创建新的分区步骤如下：

1. 创建新的分区。

在提示符下输入命令 `n`，创建新的分区：

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



控制台显示出如下信息：

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

2. 建立主分区。

输入命令 **p**，选择主分区：

```
p
```

3. 选择分区数。

本例中选择为 1，输入数字 1：

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

控制台显示出如下信息：

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

4. 选择起始柱面。

本例选择默认值 1，直接回车：

```
Using default value 1
```

5. 选择结束柱面。

本例选择默认值 970，直接回车：

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

6. 选择系统格式。

由于系统默认为 Linux 格式，本例中选择 Win95 FAT 格式，输入命令 **t** 进行修改：

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

输入命令 **b**，选择 Win95 FAT 格式：

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

输入命令 **l**，可以查看 **fdisk** 所有分区的详细信息：

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

7. 查看分区状态。

输入命令 **p**，查看当前分区状态：

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

控制台显示出当前分区状态信息，表示成功分区。



----结束

5.1.3 保存分区信息

输入命令 `w`，写入并保存分区信息到设备：

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

控制台显示出当前设备信息，表示成功写入分区信息到设备：

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

.....
~ $
```

5.2 用 mkdosfs 工具格式化

存在以下情况选择操作：

- 若已格式化，本操作可以跳过，直接到“[5.3 挂载目录](#)”。
- 若没有格式化，则输入命令 `mkdosfs` 进行格式化：

```
~ $ mkdosfs -F 32 设备分区名
```

其中设备分区名与实际接入的设备类型有关，具体名称在以上各章节的“操作示例”中均有说明。

控制台显示出如下提示信息，表示成功格式化：

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

5.3 挂载目录

使用命令 `mount` 挂载到 `mnt` 目录下，就可以进行读写文件操作：

```
~ $ mount -t vfat 设备分区名 /mnt
```

其中设备分区名与实际接入的设备类型有关，具体名称在以上各章节的“操作示例”中均有说明。

5.4 读写文件

读写操作的具体情况很多，在本例中使用命令 `cp` 实现读写操作。

使用命令 `cp` 拷贝当前目录下的 `test.txt` 文件到 `mnt` 目录下，即拷贝至设备，实现写操作，如：

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