



Hi3520 外围设备驱动

## 操作指南

文档版本	01
发布日期	2009-12-23
部件编码	N/A

深圳市海思半导体有限公司为客户提供全方位的技术支持，用户可与就近的海思办事处联系，也可直接与公司总部联系。

## 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](mailto:support@hisilicon.com)

**版权所有 © 深圳市海思半导体有限公司 2009。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

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。



## 目 录

前 言.....	1
1 PCI操作指南.....	1-1
1.1 操作准备 .....	1-1
1.2 操作过程 .....	1-1
1.3 操作示例 .....	1-2
1.4 操作中需要注意的问题 .....	1-2
2 SD/MMC卡操作指南.....	2-1
2.1 操作准备 .....	2-1
2.2 操作过程 .....	2-1
2.3 操作示例 .....	2-2
2.4 操作中需要注意的问题 .....	2-4
3 GMAC操作指南 .....	3-1
3.1 操作示例 .....	3-1
3.2 操作中需要注意的问题 .....	3-2
4 USB 2.0 操作指南 .....	4-1
4.1 操作准备 .....	4-1
4.2 操作过程 .....	4-1
4.3 操作示例 .....	4-2
4.3.1 U盘操作示例 .....	4-2
4.3.2 键盘操作示例 .....	4-3
4.3.3 鼠标操作示例 .....	4-3
4.4 操作中需要注意的问题 .....	4-4
5 附录.....	5-1
5.1 用fdisk工具分区 .....	5-1
5.1.1 查看当前状态 .....	5-1
5.1.2 创建新的分区 .....	5-1
5.1.3 保存分区信息 .....	5-3
5.2 用mkdosfs工具格式化.....	5-3



---

5.3 挂载目录.....	5-3
5.4 读写文件.....	5-3



## 插图目录

图 2-1 在控制台下实现读写SD卡的操作示例 .....	2-3
-------------------------------	-----



# 前言

## 概述

本文档主要是指导使用 PCI、SDIO、GMAC 以及 USB 2.0 Host 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
Hi3520 H.264 编解码处理器	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





## 约定

### 符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 危险	表示有高度潜在危险，如果不能避免，会导致人员死亡或严重伤害。



符号	说明
 <b>警告</b>	表示有中度或低度潜在危险，如果不能避免，可能导致人员轻微或中等伤害。
 <b>注意</b>	表示有潜在风险，如果忽视这些文本，可能导致设备损坏、数据丢失、设备性能降低或不可预知的结果。
 <b>窍门</b>	表示能帮助您解决某个问题或节省您的时间。
 <b>说明</b>	表示是正文的附加信息，是对正文的强调和补充。

## 通用格式约定

格式	说明
宋体	正文采用宋体表示。
黑体	一级、二级、三级标题采用黑体。
楷体	警告、提示等内容一律用楷体，并且在内容前后增加线条与正文隔离。
“Terminal Display” 格式	“Terminal Display” 格式表示屏幕输出信息。此外，屏幕输出信息中夹杂的用户从终端输入的信息采用加粗字体表示。
“ ”	用双引号表示文件路径。如 “C:\Program Files\Huawei”。

## 命令行格式约定

格式	意义
<b>粗体</b>	命令行关键字（命令中保持不变、必须照输的部分）采用加粗字体表示。
<i>斜体</i>	命令行参数（命令中必须由实际值进行替代的部分）采用斜体表示。
[ ]	表示用 “[ ]” 括起来的部分在命令配置时是可选的。
{ x   y   ... }	表示从两个或多个选项选取一个。
[ x   y   ... ]	表示从两个或多个选项选取一个或者不选。
{ x   y   ... } *	表示从两个或多个选项选取多个，最少选取一个，最多选取所有选项。



格式	意义
[ x   y   ... ] *	表示从两个或多个选项中选取多个或者不选。

鼠标操作约定

格式	意义
单击	快速按下并释放鼠标的一个按钮。
双击	连续两次快速按下并释放鼠标的一个按钮。
拖动	按住鼠标的一个按钮不放，移动鼠标。

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
2009-12-23	01	正式发布。
2009-10-30	00B20	修改 4.3.2 和 4.3.3 中鼠标和键盘的设备节点路径（涉及 dev/input 的目录和命令）。
2009-09-30	00B10	第一次发布。





# 1 PCI 操作指南

## 1.1 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）。

## 1.2 操作过程



### 说明

根据实际应用的需要选择以下操作。

PCI-SATA 硬盘测试步骤如下：

**步骤 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。

**步骤 2** 按以下顺序依次插入 ko 文件，由于部分模块已经编入内核了，以下可能有部分命令冗余：

- 加载文件系统和存储设备相关模块

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



- 加载硬盘相关模块  

```
modprobe libata  
modprobe sata_sil
```

**说明**

调试时如果对驱动有改动，重新生成了部分模块。则将上面的对应的 modprobe 换成 insmod 相应的模块即可。

----结束

## 1.3 操作示例

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

PCI-SATA 的操作步骤如下：

步骤 1 查看分区信息。

- 若没有分区，请参见“[5.1 用 fdisk 工具分区](#)”进行分区后，进入[步骤 2](#)。
- 若有分区信息，则硬盘已经检测到，进入[步骤 2](#)。

步骤 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[步骤 3](#)。
- 若已格式化，进入[步骤 3](#)。

步骤 3 挂载目录，请参见“[5.3 挂载目录](#)”。

步骤 4 对硬盘进行读写操作，请参见“[5.4 读写文件](#)”。

----结束

## 1.4 操作中需要注意的问题

- 由于 H3520 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。
- PCI-SATA 只能在主设备上才能进行操作。



# 2 SD/MMC 卡操作指南

## 2.1 操作准备

- 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 jffs2 或 cramfs，也可以通过本地文件系统再挂载到 NFS。
- ko 文件。



### 说明

所有提及的 ko 文件都可以从 SDK 中获取。

- SDIO 所需 ko 文件为：  
mmc\_core.ko, hi\_mci.ko, mmc\_block.ko
- 文件系统所需 ko 文件为：  
nls\_base.ko、nls\_cp437.ko、fat.ko、vfat.ko、nls\_iso8859-1.ko、nls\_ascii.ko、msdos.ko
- DMA 控制器所需 ko 文件为：  
hidmac.ko

## 2.2 操作过程

操作过程如下：

- 步骤 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。
- 步骤 2 按以下顺序依次插入 ko 文件，由于部分模块已经编入内核了，以下可能有部分命令冗余：
- 加载文件系统和存储设备相关模块  
modprobe nls\_base  
modprobe nls\_cp437  
modprobe fat



```
modprobe vfat
modprobe msdos
modprobe nls_iso8859-1
modprobe nls_ascii
```

- 加载 sdio 相关模块

```
modprobe hidmac
modprobe mmc_core
modprobe hi_mci
modprobe mmc_block
```

步骤 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“[2.3 操作示例](#)”。

----结束

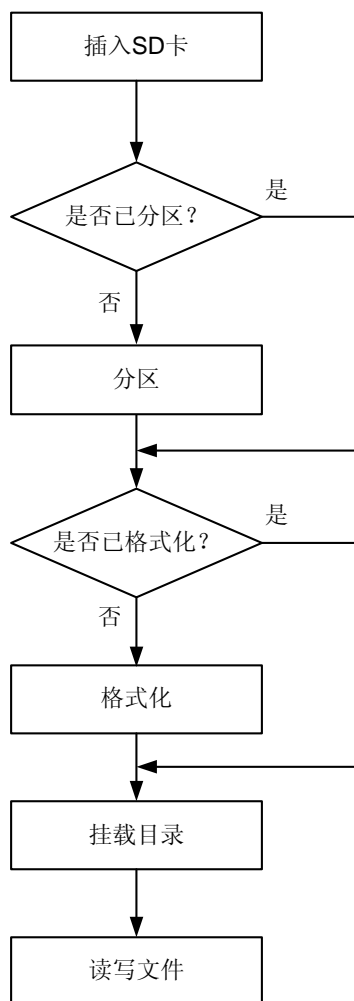
## 2.3 操作示例

此操作示例通过 SDIO 接口实现 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。

在控制台实现读写 SD 卡的操作示例如[图 2-1](#)所示。



图2-1 在控制台下实现读写 SD 卡的操作示例



## 插入 SD 卡

当内核 SD 卡的模块重新编译过，就不能用 `modprobe` 来插入模块了，而要在控制台上手动输入以下命令，需要注意的是，这里列出来的是可能需要手动插入的所有模块，具体环境中由于部分模块已经被编入内核，不需要手动插入，只需要其中的部分模块即可，重新编译的这些模块的路径要根据当前的环境进行设置。

```
insmod nls_base.ko
insmod nls_cp437.ko
insmod fat.ko
insmod vfat.ko
insmod msdos.ko
insmod nls_iso8859-1.ko
insmod nls_ascii.ko
insmod hidmac.ko
insmod mmc_core.ko
insmod hi_mci.ko
```



```
insmod mmc_block.ko
```

## 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/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

### 步骤 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[步骤 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[步骤 2](#)。

### 步骤 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[步骤 3](#)。
- 若已格式化，进入[步骤 3](#)。

### 步骤 3 挂载目录，请参见“[5.3 挂载目录](#)”。

### 步骤 4 对 SD/MMC 卡进行读写操作，请参见“[5.4 读写文件](#)”。

-----结束

## 2.4 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 [2.3 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。







# 3 GMAC 操作指南



## 说明

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自己使用的地址来设置。

## 3.1 操作示例

内核下使用网口的操作涉及到以下几个方面：

- 网口模块参数说明

**port\_mode:** 当硬件是用 rgmii 接口连接时，需要设置该参数为 1，否则可不设置，缺省是 mii 接口。

- 加载模块

```
modprobe h2gether                /* MII接口*/  
modprobe h2gether port_mode=1    /*rgmii接口*/
```

- 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
```

- 加载 nfs 模块

```
modprobe nfs
```

- 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 //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。



## 3.2 操作中需要注意的问题

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_kbytes
```



# 4 USB 2.0 操作指南

## 4.1 操作准备

- 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 jffs2 或 cramfs，也可以使用 NFS，建议使用 jffs2。
- ko 文件  
所有提及的 ko 文件都可以从 SDK 中获取。
  - Hi3520 USB2.0 操作所需公共 ko 文件已编入内核
  - U 盘存储设备所需 ko 文件为：  
fat.ko、vfat.ko、scsi\_mod.ko、sd\_mod.ko、nls\_cp437.ko、nls\_iso8859-2.ko、usb-storage.ko。
  - 键盘所需 ko 文件为：  
usbhid.ko、evdev.ko。
  - 鼠标所需 ko 文件为：  
mousedev.ko、usbhid.ko、evdev.ko。

## 4.2 操作过程

操作过程如下：

步骤 1 设置 U-boot（本地文件系统为 jffs2）：

```
setenv bootargs 'mem=32M console=ttyAMA0,115200 root=1f01  
rootfstype=jffs2 mtdparts=phys_mapped_flash:2M(boot),12M(rootfs)'
```

步骤 2 启动单板，加载 jffs2 或 cramfs 文件系统。

步骤 3 按以下顺序加载相关模块，配置相关内容：

- 加载文件系统和存储设备相关模块  
modprobe vfat  
modprobe scsi\_mod



```
modprobe sd_mod
modprobe nls_ascii
modprobe nls_iso8859-2
```

- 加载键盘相关模块

```
modprobe evdev
modprobe usbhid
```

- 加载鼠标相关模块

```
modprobe mousedev
modprobe usbhid
modprobe evdev
```

- 加载 USB2.0 模块

```
modprobe usb-storage
```

步骤 4 加载完以上模块，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“4.3 操作示例”。

----结束

## 4.3 操作示例

### 4.3.1 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 2-2: new high speed USB device using h2v100-ehci and address 2
scsi0 : SCSI emulation for USB Mass Storage devices
Vendor: Generic   Model: USB Flash Disk   Rev: 0.00
Type:   Direct-Access                ANSI SCSI revision: 02
SCSI device sda: 32243711 512-byte hdwr sectors (16509 MB)
sda: Write Protect is off
sda: assuming drive cache: write through
SCSI device sda: 32243711 512-byte hdwr sectors (16509 MB)
sda: Write Protect is off
sda: assuming drive cache: write through
sda: sda1
Attached scsi removable disk sda at scsi0, channel 0, id 0, lun 0
```

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

步骤 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[步骤 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[步骤 2](#)。

步骤 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 [2.3 步骤 3](#)。
- 若已格式化，进入[步骤 3](#)。

步骤 3 挂载目录，请参见“[5.3 挂载目录](#)”。

步骤 4 对硬盘进行读写操作，请参见“[5.4 读写文件](#)”。

----结束

## 4.3.2 键盘操作示例

键盘操作过程如下：

步骤 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。

步骤 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

## 4.3.3 鼠标操作示例

鼠标操作过程如下：

步骤 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。

步骤 2 运行 gpm 中提供的标准测试程序（建议使用 mev）。



步骤 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

## 4.4 操作中需要注意的问题

- 对于 USB 设备的操作请参见地址：<http://www.usb.org/developers/compliance/>。
- 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。



# 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：

步骤 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)
```

**步骤 2 建立主分区。**

输入命令 **p**，选择主分区：

```
p
```

**步骤 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):
```

**步骤 4 选择起始柱面。**

本例选择默认值 1，直接回车：

```
Using default value 1
```

**步骤 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
```

**步骤 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)
```

**步骤 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
```