



# 外围设备驱动 操作指南

文档版本 00B05

发布日期 2013-12-25

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



# 前 言

## 概述

本文档主要是指导使用 GMAC、ETH、USB 2.0 Host/USB 3.0 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
Hi3535 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
2013-12-25	00B05	第 1 章 GMAC 操作指南 1.1 操作示例中，修改默认使用 TOE 功能的描述。 1.2 操作中需要注意的问题中，增加 2 个注意事项。



修订日期	版本	修订说明
2013-11-18	00B04	第 1 章 GMAC 操作指南 1.3 IPv6 配置的相关描述有更新。 第 3 章 SATA 操作指南 3.4 操作中需要注意的问题中，更新使用 SATA 的 port2 时的注意事项。
2013-10-24	00B03	第 3 章 SATA 操作指南 3.4 操作中需要注意的问题新增使用 SATA 的 port2 时的注意事项。
2013-10-08	00B02	第 2 次临时版本发布。
2013-08-31	00B01	第 1 次临时版本发布。



# 目 录

<b>1 GMAC 操作指南 .....</b>	<b>1</b>
1.1 操作示例.....	1
1.2 操作中需要注意的问题.....	2
1.3 IPv6 说明 .....	2
<b>2 USB 2.0 Host/ USB 3.0 Host 操作指南.....</b>	<b>5</b>
2.1 操作准备.....	5
2.2 操作过程.....	5
2.3 操作示例.....	6
2.3.1 U 盘操作示例.....	6
2.3.2 键盘操作示例.....	8
2.3.3 鼠标操作示例.....	8
2.4 操作中需要注意的问题.....	8
<b>3 SATA 操作指南 .....</b>	<b>9</b>
3.1 操作准备.....	9
3.2 操作过程.....	9
3.3 操作示例.....	10
3.4 操作中需要注意的问题.....	10
<b>4 附录.....</b>	<b>11</b>
4.1 使用分区工具进行分区 .....	11
4.1.1 用 fdisk 工具分区.....	11
4.1.2 用 parted 工具分区.....	13
4.2 用 mkdosfs 工具格式化 .....	14
4.3 挂载目录.....	14
4.4 读写文件.....	15



## 插图目录

图 1-1 IPv6 Protocol 配置示意图.....	3
--------------------------------	---



# 1 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具体使用的地址来设置。

## 1.1 操作示例



## 说明

Hi3535 默认相关 GMAC 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 tnk\_threshold=0

上述命令已经写进启动脚本 S81toe，路径为/etc/init.d/S81toe，使用时可在此文件中修改，默认使用 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。
- Hi3535 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。`

## 1.2 操作中需要注意的问题

- Hi3535 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`  
`min_free_kbytes` 表示最小保留内存的大小, 缺省值为 8192, `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。Hi3535 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 16384 > /proc/sys/vm/min_free_kbyte`, 增大最小保留内存。
- TOE 不支持选择性重传功能, 而内核默认打开了网络模块的选择性重传, 为了 Hi3535 在 TOE 模式下能与对端设备协商一致, 需要关闭选择性重传:  
`echo 0 > /proc/sys/net/ipv4/tcp_sack`
- TOE 接收描述子个数可以配置, 默认 4096, 消耗 8MB 内存 (每个描述子 2K)。  
补充说明: 个数可以根据内存大小或应用场景配置。CPU 压力越大, 网络压力越大, 则配置描述子大可以提高性能, 其他情况下可以减少配置, 一般简单场景 256 即可, 可以节省内存, 配置范围 256~8192。  
配置方法举例:  
`insmod stmmac.ko hitoe=1 tnk_rx_fifo=4096`
- 增加 `socket` 选项, 可以通过 `setsocketopt` 来实现某个 `socket` 是否使用 TOE 功能, 目前默认每个 `socket` 都使能 TOE。

实现方法:

在具体的业务 `socket` 层代码中定义如下宏:

```
#define SO_TOE_ENABLE 44
```

在创建 `socket` 连接的代码实现中添加 TOE 使能开关:

```
int val = 0; /* DISABLE TOE : val = 0    ENABLE TOE: val = 1*/
```





```
int len = sizeof(int);
setsockopt(sockfd, SOL_SOCKET, SO_TOE_ENABLE, &val, len);
```

## 1.3 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$cd kernel/linux-3.4.y
hisilicon$cp arch/arm/configs/hi3535_full_defconfig .config
hisilicon$make ARCH=arm CROSS_COMPILE=arm-hisivXXX-linux- menuconfig
```



说明

CROSS\_COMPILE=arm-hisiXXX-linux-中 XXX 表示两种情况。

- Hi3535\_V100R001C01SPCxxx 对应 uclibc，使用 uclibc 工具链时，CROSS\_COMPILE=arm-hisiv100nptl-linux-。
- Hi3535\_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 1-1 所示。

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

图1-1 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)
```



## 注意

TOE 不支持 IPV6 功能，如果要支持 IPv6，请加载 GMAC 模块时选择不使能 TOE 模式。

IPv6 环境配置如下：

- 配置 ip 地址及缺省网关

```
hisilicon$ ip -6 addr add <ipv6address>/<ipv6_prefixlen> dev <port>
```

示例：ip -6 addr add 2001:da8:207::9402/64 dev eth0

- Ping 某个网址

```
hisilicon$ ping -6 <ipv6address>
```

示例：ping -6 2001:da8:207::9403



# 2 USB 2.0 Host/ USB 3.0 Host 操作指南



## 说明

Hi3535 单板自带一个 USB 2.0 Host 接口和一个 USB 3.0 Host 接口，两者差异部分单独阐述。

## 2.1 操作准备

USB 2.0 Host/ USB 3.0 Host 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。

## 2.2 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“[2.3 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-hi3535
- USB3.0 模块
  - ohci-hcd
  - ehci-hcd
  - xhci-hcd
  - usb-storage
  - hiusb3.0

----结束

## 2.3 操作示例

### 2.3.1 U 盘操作示例

#### 插入检测

直接插入 U 盘，观察是否枚举成功。

USB 2.0 Host 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
```



USB 3.0 Host 正常情况下串口打印为:

```
usb 4-1: new SuperSpeed USB device number 2 using xhci-hcd
scsil : usb-storage 4-1:1.0
usbdev42 -> /dev/usbdev4.2
scsi 1:0:0:0: Direct-Access      SanDisk Extreme          0001 PQ: 0 ANSI: 6
sd 1:0:0:0: [sda] 31277232 512-byte logical blocks: (16.0 GB/14.9 GiB)
sd 1:0:0:0: Attached scsi generic sg0 type 0
sd 1:0:0:0: [sda] Write Protect is off
sd 1:0:0:0: [sda] Mode Sense: 33 00 00 08
sd 1:0:0:0: [sda] Write cache: enabled, read cache: enabled, doesn't
support DPO or FUA
sda: sda1
sd 1:0:0:0: [sda] Attached SCSI disk
udisk4110 -> /dev/sda
udisk4110p1 -> /dev/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 等字样。

## 初始化及应用

模块插入完成后, 进行如下操作:



说明

其中 X 代表磁盘号, Y 代表分区号, 请根据具体系统环境进行修改。

- 分区命令操作的具体设备节点为 sdX, 示例: \$ fdisk /dev/sda 或者 \$ parted /dev/sda
- 用 mkdosfs 工具格式化的具体分区为 sdXY: ~ \$ mkdosfs -F 32 /dev/sda1
- 挂载的具体分区为 sdXY: ~ \$ mount -t vfat /dev/sda1 /mnt

### 1. 查看分区信息。

- 运行命令 “ls /dev” 查看系统设备文件, 若没有分区信息 sdXY, 表示还没有分区, 请参见 “[4.1 使用分区工具进行分区](#)” 进行分区后, 进入 2。
- 若有分区信息 sdXY, 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 “[4.2 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 “[4.3 挂载目录](#)”。

### 4. 对硬盘进行读写操作, 请参见 “[4.4 读写文件](#)”。

----结束



## 2.3.2 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

## 2.3.3 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

## 2.4 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。



# 3 SATA 操作指南

## 3.1 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）。

## 3.2 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“[3.3 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

### 3.3 操作示例

对 SATA 硬盘进行如下操作：



说明

其中 X 代表磁盘号，Y 代表分区号，请根据具体系统环境进行修改。

- 分区命令操作的具体设备节点为 sdX，示例：\$ fdisk /dev/sda 或者 \$ parted /dev/sda
- 用 mkdosfs 工具格式化的具体分区为 sdXY：~ \$ mkdosfs -F 32 /dev/sda1
- 挂载的具体分区为 sdXY，示例：\$ mount -t vfat /dev/sda1 /mnt

SATA 的操作步骤如下：

1. 查看分区信息：
  - 运行命令“ls /dev”查看系统设备文件，若没有分区信息 sdXY，表示还没有分区，请参见“[4.1 使用分区工具进行分区](#)”进行分区后，进入 2。
  - 若有分区信息 sdXY，则已经检测到硬盘，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“[4.2 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“[4.3 挂载目录](#)”。
4. 对硬盘进行读写操作，请参见“[4.4 读写文件](#)”。

----结束

### 3.4 操作中需要注意的问题

- 由于 Hi3535 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。
- Hi3535 支持 3 个 SATA 端口（port0、port1 以及 port2），port2 默认不使能。需要使用 port2 时，执行如下配置步骤修改配置表格：
  - 配置 CRG 寄存器 0x200300AC（PERI\_CRG43）的 bit[6]为 0x0，设置 CMB PHY 为 SATA 模式。
  - 配置外设控制寄存器 0x20120004（MISC\_CTRL1）的 bit[12]为 0x0，设置 CMB PHY 的 SATA 功能使能。
  - 配置 CRG 寄存器 0x200300AC（PERI\_CRG43）的 bit[5]为 0x0，选择 CMB PHY 的 refclk 为 100MHz。
  - 配置 CRG 寄存器 0x200300AC（PERI\_CRG43）的 bit[4]为 0x1，打开 CMB PHY 的 refclk 的时钟门控。





# 4 附录

## 4.1 使用分区工具进行分区

fdisk 工具采用传统的 MBR(Master Boot Record)分区方式，适合对不超过 2TB 的硬盘进行分区（或单个分区超过 2TB），超过 2TB 请使用 parted 工具进行分区（参考“[4.1.2 用 parted 工具分区](#)”），否则建议使用 fdisk 工具（参考“[4.1.1 用 fdisk 工具分区](#)”）。

### 4.1.1 用 fdisk 工具分区

通过[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“[4.2 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。

### 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[创建新的分区](#)和[保存分区信息](#)的描述对设备进行分区。

### 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

## 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.
.....
~ $
```

## 4.1.2 用 parted 工具分区

通过[查看分区信息](#)，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“[4.2 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 `parted`，具体格式如下：  
~ \$ `parted` 设备节点

回车后，输入命令 `help`，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。

## 查看分区信息

在控制台的提示符下，输入命令 `print`，查看当前分区状态：

```
(parted) print
```

控制台显示出分区状态信息：

Number	Start	End	Size	File system	Name	Flags
1	1049kB	2003GB	2002GB			

## 创建分区标签

在控制台提示符下输入 `mklabel` 创建分区标签：

```
(parted) mklabel
```

```
New disk label type? gpt
```

```
gpt
```

```
Warning: The existing disk label on /dev/sda will be destroyed and all
```



```
data on
this disk will be lost. Do you want to continue?
Yes/No? Yes
(parted)
```

## 创建分区

1. 输入 `mkpart` 命令开始分区

```
(parted) mkpart
mkpart
Partition name? []?          /* 设置分区名称, 此处为空 */
File system type? [ext2]? ext2 /* 设置分区格式 */
```

2. 选择分区起始、结束位置

分区的起始大小可以用4GB或者10%表示:

```
Start? 0%          /*这里起始位置用0% */
End? 100%          /*结束位置用100% */
```

----结束

## 保存分区信息

直接输入 `quit` 就可以把当前的分区信息保存到分区表:

```
(parted) quit
```

## 4.2 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“[4.3 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)
~ $
```

## 4.3 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。

## 4.4 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
```