



指纹模块 **USB** 驱动调试说明（**linux**）

杭州城章科技有限公司

2014 年 3 月

目录

目录.....	II
附图目录.....	III
1 系统及平台配置.....	1
1.1 内核版本及系统.....	1
1.2 编译工具.....	1
1.3 查看系统是否支持 usbfs 文件系统.....	1
2 代码修改.....	3
2.1 确保 PID,VID 正确.....	3
2.2 关闭 RESET 程序段.....	3
3 没有/proc/bus/usb 文件夹解决办法.....	4
3.1 下载源码树并解压.....	4
3.2 配置内核.....	4
3.3 编译内核.....	5
3.4 编译, 加载模块.....	5
3.5 加载内核.....	6
4 设备无法打开的情况.....	7

附图目录

图 1-1	bus 文件夹内容.....	1
图 1-2	usb 文件夹内容.....	1
图 1-3	devices 文件内容.....	2
图 2-1	vid 和 pid.....	3
图 2-2	屏蔽的 reset 代码段.....	3
图 3-1	源码树.....	4
图 3-2	源码文件.....	4
图 3-3	设备驱动选项.....	5
图 3-4	USB 支持选项.....	5
图 3-5	usb 文件系统选项.....	5

1 系统及平台配置

1.1 内核版本及系统

本驱动支持 linux 内核版本为 2.6 及以上版本。

使用 `#uname -a` 查看内核版本。

本驱动支持大部分 Linux 系统，如 ubuntu, redhat, centos, fedora, freeBSD 等。

1.2 编译工具

编译需要安装 g++ 编译工具。

使用 `#type g++` 查看是否已安装 g++ 编译工具。

如果没有，如下安装：

```
#yum install g++
```

1.3 查看系统是否支持 usbfs 文件系统

查看 `/proc/bus/` 下是否有 `usb` 文件夹，如下图所示。

```
[root@(none) bus]# ls
input  pci  usb
[root@(none) bus]# pwd
/proc/bus
```

图 1-1 bus 文件夹内容

如果有 `USB` 文件夹则证明在内核中是支持 `USBFS` 文件系统的，如果没有则查看标题 3。

再看一下 `/proc/bus/usb` 文件夹下是否为空，如果为空则输入以下命令挂载 `USBFS` 文件系统（注意要有 `ROOT` 权限）。

```
#sudo mount -t usbfs usbfs /proc/bus/usb
```

如果需要每次启动都进行此次挂载，则打开 `/etc/fstab` 文件（`root` 权限）加入下面一行：

```
usbfs /proc/bus/usb usbfs defaults 0 0
```

再查看 `/proc/bus/usb` 文件夹内容，应该如下：

```
[root@(none) bus]# cd usb/
[root@(none) usb]# ls
001 002 devices _
```

图 1-2 usb 文件夹内容

一般还有其他数字文件夹（如 `003,004`），里面都是挂载的 `USB` 设备。

如果插上 AS602 独立协议模块，等待几秒后查看 **devices** 文件。

```
#cat /proc/bus/usb/devices
```

可看到如下输出：

```
T: Bus=02 Lev=01 Prnt=01 Port=01 Cnt=02 Dev#= 3 Spd=
D: Ver= 1.10 Cls=09(hub ) Sub=00 Prot=00 MxPS= 8 #C
P: Vendor=0e0f ProdID=0002 Rev= 1.00
S: Product=VMware Virtual USB Hub
C:* #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr= 0mA
I:* If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00
E: Ad=81(I) Atr=03(Int.) MxPS= 1 IvL=255ms

T: Bus=02 Lev=02 Prnt=03 Port=00 Cnt=01 Dev#= 5 Spd=
D: Ver= 1.10 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #C
P: Vendor=2109 ProdID=7638 Rev= 1.00
S: Manufacturer=Finger Chip
S: Product=Finger Module
C:* #Ifs= 1 Cfg#= 1 Atr=80 MxPwr=300mA
I:* If#= 0 Alt= 0 #EPs= 2 Cls=08(stor.) Sub=06 Prot=50
E: Ad=81(I) Atr=02(Bulk) MxPS= 64 IvL=0ms
E: Ad=02(0) Atr=02(Bulk) MxPS= 64 IvL=0ms

T: Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=
B: Alloc= 0/800 us ( 0%), #Int= 0, #Iso= 0
D: Ver= 2.00 Cls=09(hub ) Sub=00 Prot=00 MxPS=64 #C
P: Vendor=1d6b ProdID=0002 Rev= 3.04
```

图 1-3 devices 文件内容

其中根据 **Vendor** 和 **prodID** 即可查看是否已经识别我们的设备，在我们提供的驱动中通过 **USB** 打开的 **VID** 为 **0453** **PID** 为 **9005**，如果通过 **UDISK** 打开的 **VID** 为 **2109** **PID** 为 **7638**。

注：这一步十分关键，有时在虚拟机上运行程序或者拔掉，重启设备经常使系统找不到设备，可在此查看。

2 代码修改

2.1 确保 PID,VID 正确

拿到代码首先查看 SYDevice.cpp 文件是否使用了正确的 vid 和 pid，正确设置如下图所示。

```
#define ST_VENDOR 0x0453
#define ST_PRODUCT 0x9005

#define SU_VENDOR 0x2109
#define SU_PRODUCT 0x7638
```

图 2-1 vid 和 pid

其中 ST_XXXXX 为 USB 连接，SU_XXXXX 为 UDISK 连接。

这时候#./SYDemo 运行程序。

2.2 关闭 RESET 程序段

如果出现程序等待或者返回错误信息：ioctl:devices or resource busy。

因为此设备与主机一直在发送握手协议，导致设备读写 busy。则进行如下操作：

- 打开 SYDevice.cpp 文件。
- 找到 BOOL OpenUDisk()函数。
- 屏蔽掉 3.Reset 代码段，如图所示。

```
        perror("USBDEVFS_CLAIMINTERFACE");^M
    }^M
    ^M
    //3.Reset^M
    *
    ret = ioctl(fdusb->fd, USBDEVFS_RESET);^M
    perror("USBDEVFS_RESET");^M
    printf("openudisk function is success,here\n");
    /
    M
    return TRUE;^M
```

图 2-2 屏蔽的 reset 代码段

- 然后重新编译，#make

再运行#./SYDemo

3 没有/proc/bus/usb 文件夹解决办法

因为 linux 内核在 2.6.31 之后的版本遗弃了 USBFS 文件系统，所以在高内核版本的系统中找不到此文件夹，可通过编译内核加载内核解决此问题。

3.1 下载源码树并解压

登陆网站 kernel.org 点击你想编译内核的版本下载。

mainline:	3.9-rc6	2013-04-08	[tar.xz]	[pgp]	[patch]	[view patch]	[gitweb]
stable:	3.8.6	2013-04-05	[tar.xz]	[pgp]	[patch]	[view patch]	[view inc] [gitweb] [changelog]
stable:	3.7.10 [EOL]	2013-02-27	[tar.xz]	[pgp]	[patch]	[view patch]	[view inc] [gitweb] [changelog]
stable:	3.6.11 [EOL]	2012-12-17	[tar.xz]	[pgp]	[patch]	[view patch]	[view inc] [gitweb] [changelog]
longterm:	3.4.39	2013-04-05	[tar.xz]	[pgp]	[patch]	[view patch]	[view inc] [gitweb] [changelog]
longterm:	3.2.43	2013-04-10	[tar.xz]	[pgp]	[patch]	[view patch]	[view inc] [gitweb] [changelog]
longterm:	3.0.72	2013-04-05	[tar.xz]	[pgp]	[patch]	[view patch]	[view inc] [gitweb] [changelog]
longterm:	2.6.34.14	2013-01-16	[tar.xz]	[pgp]	[patch]	[view patch]	[view inc] [gitweb] [changelog]
linux-next:	next-20130409	2013-04-09					[gitweb]

Other resources

Social

图 3-1 源码树

把下载好的文件放入/usr/src 下。

如果是 tar.xz 压缩包 则可使用#tar Jxvf xxx.tar.xz 来压缩文件，此步骤需要一些时间。

进入压缩好的文件夹，如下图所示。

```
[root@(none) linux-3.4.39]# ls
arch          firmware     kernel       Module.symvers  sound
block         fs           lib           net              System.map
COPYING       include     MAINTAINERS  README           tools
CREDITS       init        Makefile     REPORTING-BUGS  usr
crypto        ipc         mm           samples          virt
Documentation Kbuild     modules.builtin scripts          vmlinux
drivers       Kconfig    modules.order security         vmlinux.o
```

图 3-2 源码文件

3.2 配置内核

输入命令

```
#make menuconfig
```

这时可能需要安装 **ncurse** 包，安装即可。

则进入内核配置选项：

我们选择

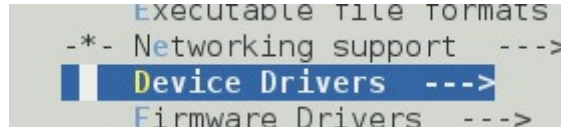


图 3-3 设备驱动选项

回车，然后选择

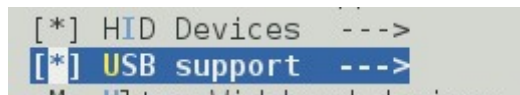


图 3-4 USB 支持选项

回车，然后选择

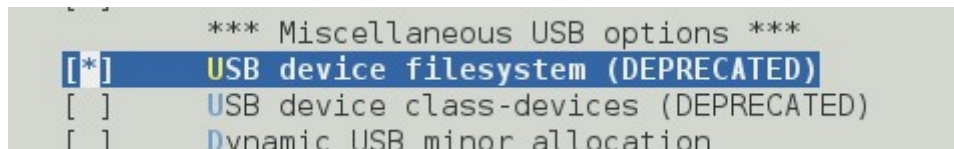


图 3-5 usb 文件系统选项

按空格是此项选中，即内核支持 **USBFS** 文件系统。保存退出即可。

3.3 编译内核

输入以下命令

```
#make dep
```

此命令确定编译依赖性，大概 1 分钟以内执行完成。

输入：

```
#make clean
```

大概一分钟

输入：

```
#make bzImage
```

编译内核，这部需要等很久，大概 15 分钟到一个小时

3.4 编译，加载模块

输入：

```
#make modules
```

等待很久。

输入:

```
#make modules_install
```

等待几分钟。

3.5 加载内核

输入:

```
#make install
```

重启即可。

重启后可以看到 `/proc/bus` 下有 `usb` 文件夹。

4 设备无法打开的情况

如果出现了设备无法开打的情况则需要检查用户权限。可以给设备更广泛的权限，首先要找到你的设备，在我们实验的机子上是 `/proc/bus/usb/002/004`（有些机子上在 `005` 文件夹下，此文件名称不固定变动较大）这里比如是 `004` 则给相应的权限，如：

```
#chmod 777 004
```

再执行程序 `./SYDemo`