

## iTOP-4412-驱动-usb 文档 12-usb3G 驱动移植

本文档将介绍如何移植华为 E261 WCDMA 12d1:1446USB 上网卡到 4412 开发板 QtE\Linux 系统。

移植 USB 的 3G 驱动，主要有以下几个步骤：

1 内核配置，3G 的驱动，包括 4G 的驱动，在内核中都自带，如果没有配置则需要配置下。

2 libusb 库的移植，运行 usb-modeswitch 的时候需要这个库的支持。

3 usb-modeswitch 库的移植，这个工具叫 usb-modeswitch，可以将 USB 设备从普通模式转化为调试解调器模式。

4 usb-modeswitch-data，这是个配置文件，用来匹配设备和 usb-modeswitch。

5 ppp 的移植，这是一个拨号工具，上网需要拨号，拨号的时候需要用到这个工具。

### 1 内核配置

在内核源码目录下，使用命令 “make menuconfig” 打开 menuconfig 配置。

配置内核以下选项，如下图示。

Device Drivers >

USB support

<\*>USB Mass Storage support

[\*]USB Mass Storage verbose debug

<\*>Realtek Card Reader support

<\*>Datafab Compact Flash Reader support

配置之后，如下图所示。

```
x lqqqq<F>qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
x x < > Inventra Highspeed Dual Role Controller (TI, ADI, ...) x x
x x *** USB Device Class drivers *** x x
x x <*> USB Modem (CDC ACM) support x x
x x < > USB Printer support x x
x x < > USB Wireless Device Management support x x
x x < > USB Test and Measurement Class support x x
x x *** NOTE: USB_STORAGE depends on SCSI but BLK_DEV_SD may ** x x
x x *** also be needed; see USB_STORAGE Help for more info *** x x
x x <*> USB Mass Storage support x x
x x [*] USB Mass Storage verbose debug x x
x x <*> Realtek Card Reader support x x
x x <*> Datafab Compact Flash Reader support x x
x x < > Freecom USB/ATAPI Bridge support x x
x x < > ISD-200 USB/ATA Bridge support x x
x x < > USBAT/USBAT02-based storage support x x
x x < > SanDisk SDDR-09 (and other SmartMedia, including DPCM) su x x
x x < > SanDisk SDDR-55 SmartMedia support x x
```

接着配置，如下所示。

DeviceDrivers-->

[\*]USB support-->

<\*>USB Serial Converter support-->

[\*]USB Generic Serial Driver

<\*>USB driver for GSM and CDMA modems

以及

Device Drivers >

Network device support>

PPP\*\*\*全选

配置之后，如下图所示。

```

.config - Linux/arm 3.0.15 Kernel Configuration
Network device support
Arrow keys navigate the menu. <Enter> selects submenus ---.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >
*** Enable WiMAX (Networking options) to see the WiMAX driver
USB Network Adapters ---
[ ] Wan interfaces support ---
*** CAIF transport drivers ***
PPP (point-to-point protocol) support
[*] PPP multilink support (EXPERIMENTAL)
[*] PPP filtering
<M> PPP support for async serial ports
<M> PPP support for sync tty ports
<M> PPP Deflate compression
<M> PPP BSD-Compress compression
<M> PPP MPPE compression (encryption) (EXPERIMENTAL)
<M> PPP over Ethernet (EXPERIMENTAL)
<M> PPP on L2TP Access Concentrator
<M> PPP on PPTP Network Server
< > SLIP (serial line) support
< > Network console logging support

```

然后重新编译内核，烧写到开发板，内核驱动部分就完成了。

## 2 工具的编译

在进行编译之前要先修改编译器为 4.3.2 版本，如何设置编译器参见手册 7.1 章节

“Qt/E4.7.1 编译器的安装”。

本次移植共需要四个工具，它们互相有依赖关系，为了方便编译，将它们拷贝到同一目录并解压，如下图所示。

在压缩包中都有下载好的源码，各个工具有不同官方网站，用户可以在管网上自行搜索下载，类似的资料超级多。

```
root@ubuntu:/home/frao/workplace/3GWDMA# ls
libusb-0.1.12      usb-modeswitch-1.2.6
libusb-0.1.12.tar.gz  usb-modeswitch-1.2.6.tar.bz2
ppp-2.4.4          usb-modeswitch-data-20140327
ppp-2.4.4.tar.gz   usb-modeswitch-data-20140327.tar.bz2
root@ubuntu:/home/frao/workplace/3GWDMA#
```

## 2.1 编译 libusb-0.1.12

解压 “libusb-0.1.12.tar.gz”，进入文件夹 libusb-0.1.12，执行以下配置命令。

```
./configure --host=arm-none-linux-gnueabi --prefix=$(pwd)/../_install
```

它将设置使用的编译器，并且指定 make install 后生成目录的路径为上级目录下的

\_\_install 文件夹，结果如下图所示，

```
root@ubuntu:/home/frao/workplace/3GWDMA/libusb-0.1.12# ./configure --host=arm-none-linux-gnueabi --prefix=$(pwd)/../__install
configure: WARNING: If you wanted to set the --build type, don't use --host.
        If a cross compiler is detected then cross compile mode will be used.
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
```

然后执行 “make” 以及 “make install” ，可以看到在上级目录产生了 “\_\_install” 文件夹，如下所示。

```
make[2]: Leaving directory /home/frao/workplace/3GWDMA/libusb-0.1.12/doc
make[1]: Leaving directory /home/frao/workplace/3GWDMA/libusb-0.1.12/doc
root@ubuntu:/home/frao/workplace/3GWDMA/libusb-0.1.12#
root@ubuntu:/home/frao/workplace/3GWDMA/libusb-0.1.12#
root@ubuntu:/home/frao/workplace/3GWDMA/libusb-0.1.12# cd ../__install/
root@ubuntu:/home/frao/workplace/3GWDMA/__install# ls
bin  include  lib
```

运行 modeswitch 工具的时候需要用到编译出的库，\_\_install/lib 目录下编译出的 libusb.so，libusb-0.1.so.4，libusb-0.1.so.4.4.4 拷贝到开发板的/lib 目录下。

编译 modeswitch 的时候也需要用到这个库，后面指定路径即可。

## 2.2 编译 usb-modeswitch-1.2.6

编译该工具需要使用上一步生成的库文件以及头文件，即\_\_install/include 以及 \_\_install/lib 文件夹，进入源码文件夹 usb-modeswitch-1.2.6，将 Makefile 改成以下所示内容。

```
PROG      = usb_modeswitch
VERS      = 1.2.6
CC        = arm-none-linux-gnueabi-gcc
CFLAGS    += -Wall $(pwd)/../__install/include/
#LIBS     = libusb.a
LIBS      = -l usb -L$(pwd)/../__install/lib/
RM        = /bin/rm -f
OBJS      = usb_modeswitch.c
```

以及



```
script: $(PROG) dispatcher-script
shared: $(PROG) dispatcher-shared
static: $(PROG) dispatcher-static
$(PROG): $(OBJS) usb_modeswitch.h
        $(CC) -o $(PROG) $(OBJS) $(CFLAGS) $(LIBS) $(LDFLAGS) -static
```

保存并编译，输入 make 指令，如下图所示。

```
root@ubuntu:/home/frao/workplace/3GWDMA/usb-modeswitch-1.2.6# make
No installed jimsh or tclsh, building local bootstrap jimsh0
arm-none-linux-gnueabi-gcc -o usb_modeswitch usb_modeswitch.c -Wall -I ../__install/include/ -L ../__install/lib/ -l usb -static
sed 's!/usr/bin/tclsh!'"/usr/bin/tclsh"'_ ' < usb_modeswitch.tcl > usb_modeswitch_dispatcher
root@ubuntu:/home/frao/workplace/3GWDMA/usb-modeswitch-1.2.6#
```

在源码目录下，编译得到的 usb\_modeswitch 文件，需要将其拷贝到开发板的 usr/sbin/ 目录。

## 2.3 编译 usb-modeswitch-data-20140327

进入该源码文件夹的 usb\_modeswitch.d 目录，如下图所示。

```
root@ubuntu:/home/frao/workplace/3GWDMA# cd usb-modeswitch-data-20140327/
root@ubuntu:/home/frao/workplace/3GWDMA/usb-modeswitch-data-20140327# ls
40-usb_modeswitch.rules  COPYING          Makefile  usb_modeswitch.d
ChangeLog                gen-rules.tcl   README
root@ubuntu:/home/frao/workplace/3GWDMA/usb-modeswitch-data-20140327# cd usb_modeswitch.d/
root@ubuntu:/home/frao/workplace/3GWDMA/usb-modeswitch-data-20140327/usb_modeswitch.d# ls
03f0:002a      0af0:8201      12d1:1520      19d2:1219
0408:ea17      0af0:8300      12d1:1521      19d2:1224
0408:ea25      0af0:8302      12d1:1523      19d2:1225
0408:ea43      0af0:8304      12d1:1526      19d2:1227
```

使用命令 “vim 12d1:1446” 打开文件，并修改成以下内容。

```
# Huawei, newer modems
DefaultVendor= 0x12d1
DefaultProduct= 0x1446

TargetVendor=0x12d1
TargetProductList="1001,1406,140b,140c,1412,141b,1432,1433,1436,14ac"
CheckSuccess=20

MessageContent="555342431234567800000000000000011062000000101000100000000000000"
```

```
DisableSwitching=0
EnableLogging=0
```

如下图所示。

```
# Huawei, newer modems
DefaultVendor= 0x12d1
DefaultProduct= 0x1446

TargetVendor=0x12d1
TargetProductList="1001, 1406, 140b, 140c, 1412, 141b, 1432, 1433, 1436, 14ac"
CheckSuccess=20

MessageContent="5553424312345678000000000000000011062000000101000100000000000000"

DisableSwitching=0
EnableLogging=0
```

上面这段代码最重要的是 DefaultVendor= 0x12d1 和 DefaultProduct= 0x1446，它制定了 USB 设备的 VID 和 PID。这个 VID 和 PID 是在把 3G 上网卡插到开发板上之后，使用命令 lsusb 得到的信息。但是不在开发板，而是在 PC 的 Ubuntu 上的话，由于 Ubuntu 上自带了切换工具，所以查不到需要配置的信息，它会将 0x1446 自动切换为 0x1436，0x1436 是正确切换之后的 ID，我们需要的是切换之前的信息。

我们更改的这个文件如果要拷贝到 U 盘上则需要改名为 Windows 允许的名称，后面测试的时候再介绍如何改名。

## 2.4 编译 ppp-2.4.4

进入 ppp 源码目录，运行以下命令编译

```
./configure
make CC=arm-none-linux-gnueabi-gcc
```

编译到此完成，可以看到在 pppd 文件夹生成了 pppd 工具，在 chat 文件夹生成了 chat 工具，如下图所示。

```
-rw-r--r--  1 root root  58200 Jan 31 19:06 tdb.o
-rw-r--r--  1 root root   7092 Jan 31 19:06 spinlock.o
drwxr-xr-x  3 frao frao   4096 Jan 31 19:06 ./
-rwxr-xr-x  1 root root 626765 Jan 31 19:06 pppd*
root@ubuntu: /home/frao/workplace/3GWDMA/ppp-2.4.4/pppd#
```

```
-rw-r--r--  1 root root 46728 Jan 31 19:05 chat.o
drwxr-xr-x  2 frao frao  4096 Jan 31 19:05 ./
-rwxr-xr-x  1 root root 44853 Jan 31 19:05 chat*
root@ubuntu:/home/frao/workplace/3GWDMA/ppp-2.4.4/chat#
```

编译出来的 pppd 和 chat 工具拷贝到开发板/usr/sbin/目录。

### 3 测试

编译工作完成之后，接下来分别将每一步编译得到的工具拷贝到开发板。

(1) 将 2.1 中\_install/lib 目录下的 libusb.so，libusb-0.1.so.4，libusb-0.1.so.4.4.4 拷贝到开发板的/lib 文件夹。

(2) 将 2.2 中编译得到的 usb\_modeswitch 文件拷贝到开发板的/usr/sbin/目录。

(3) 将 2.3 中修改的文件 12d1:1446 拷贝到开发板的/etc/usb\_modeswitch.d/目录下。

注意到 Windows 的文件名中不允许有 “:”，所以使用较老版本的 ssh 软件传输该文件到 Windows 上的话会报错，如果这样那用户需要先重命名再传输。但是最新版本的 ssh 软件如 Winscp 在传输文件会自动将文件重命名为 Windows 允许的名称，如下图。所以用户需要在把文件拷贝到开发板上后要再改回原名。重命名操作如下图所示。

```
~ # ls
12d1%3A1446  home      opt
bin          lib        proc
dev          linuxrc   root
etc          mnt       sbin
~ # mv 12d1\%3A1446 12d1\:1446
~ # ls
12d1:1446  etc      linuxrc  proc
```

(4) 将 2.4 中得到的 pppd 和 chat 工具拷贝到开发板/usr/sbin/目录。

(5) 将压缩包提供的脚本 wcdma，wcdma-chat-connect，wcdma-chat-disconnect 拷贝到开发板的/etc/ppp/peers/目录下。在使用 usb\_modeswitch 拨号工具切换 USB-3G 的模式之后，需要通过 pppd 和 chat 来拨号上网，这几个脚本中，其中 wcdma，wcdma-

chat-connect , wcdma-chat-disconnect 这三个互相依赖 , pppd\_conf.sh 中会运行 wcdma 脚本 , 然后 wcdma 脚本中会在连接和断开连接会分别调用剩下的两个脚本 , 其中的核心是 pppd 和 chat 工具的使用。这两个工具使用 AT 指令集 , 即海斯指令集来给上网卡交流 , 如果想了解更多关于 AT 指令集 , 推荐去 github 找指导文档。如果单纯的用来拨号联网 , 这几个脚本就够了。

拷贝完成后 , 将上网卡 E261 插到开发板上 , 开发板弹出以下内容。

```
~ #
~ #
~ # [ 453.610365] usb 1-3.1: new high speed USB device number 6 using s5p-ehci
[ 453.723513] usb 1-3.1: New USB device found, idVendor=12d1, idProduct=1446, b
[ 453.730529] usb 1-3.1: New USB device strings: Mfr=3, Product=2, SerialNumber
[ 453.737711] usb 1-3.1: New USB device Class: Class=0, SubClass=0, Protocol=0
[ 453.744731] usb 1-3.1: Product: HUAWEI Mobile
[ 453.749067] usb 1-3.1: Manufacturer: HUAWEI Technology
[ 453.780713] scsi2 : usb-storage 1-3.1:1.0
[ 453.840191] scsi3 : usb-storage 1-3.1:1.1
[ 454.812257] scsi 2:0:0:0: CD-ROM HUAWEI Mass Storage 2.31 PQ
~ # [ 454.829804] scsi 2:0:0:0: Attached scsi generic sg0 type 5
[ 454.862191] scsi 3:0:0:0: Direct-Access HUAWEI SD Storage 2.31 PQ
[ 454.898697] sd 3:0:0:0: Attached scsi generic sg1 type 0
[ 454.911981] sd 3:0:0:0: [sda] Attached SCSI removable disk
```

等待上网卡闪烁蓝灯 , 然后输入以下指令。

```
usb_modeswitch -W -c /etc/usb_modeswitch.d/12d1:1446
```

之后运行压缩包中提供的脚本 pppd\_conf.sh , 运行耗时大约 20s。

```
~ #
~ # sh pppd_conf.sh
pppd options in effect:
debug          # (from /etc/ppp/peers/wcdma)
nodetach       # (from /etc/ppp/peers/wcdma)
dump           # (from /etc/ppp/peers/wcdma)
noauth         # (from /etc/ppp/peers/wcdma)
user card      # (from /etc/ppp/peers/wcdma)
password ????? # (from /etc/ppp/peers/wcdma)
remotename 3gppp # (from /etc/ppp/peers/wcdma)
```

设置成功应如下图所示。

```
rcvd [IPCP ConfAck id=0x4 <addr 10.27.207.13> <ms-dns1 202.99.160.68> <ms-dns3 202.99.166.4>]
Could not determine remote IP address: defaulting to 10.64.64.64
not replacing existing default route via 192.168.1.1
local IP address 10.27.207.13
remote IP address 10.64.64.64
primary DNS address 202.99.160.68
secondary DNS address 202.99.166.4

10.64.64.64
202.99.160.68
[root@iTOP-4412]#
```

现在就可以使用 3G 网络上网了。



至此，USB 的 3G 移植完成，整个过程可以看出，在 linux 中，一帮大牛们将复杂的 USB 和 3G 转化为简单的类似 linux 命令的操作，极大的降低了开发难度以及提升了开发效率。

## 联系方式

北京迅为电子有限公司致力于嵌入式软硬件设计，是高端开发平台以及移动设备方案提供商；基于多年的技术积累，在工控、仪表、教育、医疗、车载等领域通过 OEM/ODM 方式为客户创造价值。

iTOP-4412 开发板是迅为电子基于三星最新四核处理器 Exynos4412 研制的一款实验开发平台，可以通过该产品评估 Exynos 4412 处理器相关性能，并以此为基础开发出用户需要的特定产品。

本手册主要介绍 iTOP-4412 开发板的使用方法，旨在帮助用户快速掌握该产品的应用特点，通过对开发板进行后续软硬件开发，衍生出符合特定需求的应用系统。

如需平板电脑案支持，请访问迅为平板方案网“<http://www.topeet.com>”，我司将有能力为您提供全方位的技术服务，保证您产品设计无忧！

本手册将持续更新，并通过多种方式发布给新老用户，希望迅为电子的努力能给您的学习和开发带来帮助。

迅为电子

2018 年 2 月