

GobiNet 驱动加载手册

手册说明:

本文档介绍了 NL650/NL660 系列模块在 Linux 系统下的驱动安装的方法, 以及驱动安装完成后测试 NDIS 拨号的方法。

驱动安装包括添加 USB 串口驱动以及集成 Nodecom 提供的 NDIS 口驱动。

1 USB 端口信息

NL650/NL660 模块 USB 会枚举出 6 个逻辑端口, 他们的枚举顺序, 端口功能, 端口名字如下表所示:

| Vendor ID:0x1508 Product ID:0x1001 | | |
|---------------------------------------|--------------------|---------------------------------------|
| Interface Number | Interface Function | Interface Name |
| 0 | Debug | Device Diagnostic Interface |
| 1 | Modem | Modem Connector |
| 2 | AT | Device Application Interface |
| 3 | Pipe | Device Pipe |
| 4 | NDIS | Wireless Data Device Ethernet Adapter |
| 5 | adb | Android Composite ADB Interface Ic |

2 USB 驱动添加系统组件

2.1 USB 串口驱动添加

普通的 3G 数据卡(modem)使用 USB 接口, 对上层的功能接口是串口, 在 Linux 系统中通常使用 USB 转串口的驱动。

驱动添加需要配置 Linux 内核, 配置方法如下:

```
cd kernel
```

```
make menuconfig
```

```
device drivers->usb support->usb serial converter support
```

选中如下组件:

USB driver for GSM and CDMA modems

选中后保存配置。

2.2 增加具体设备驱动

打开内核源码文件 option.c(路径一般为 drivers/usb/serial/option.c);

在源码中找到 option_ids 数组, 在数组中添加 Nodecom 产品的 VID(0x1508)和 PID(0x1001);

```
static const struct usb_device_id option_ids[] = {
#ifdef 1 //Added by Nodecom
    { USB_DEVICE(0x1508, 0x1001) }, /* Nodecom NL650/660 */
#endif
}
```

2.3 USB 串口驱动过滤 NDIS 接口

由于 USB 串口跟 NDIS 都属于非标准 CDC 设备，需要防止 NDIS 口被 USB 串口驱动加载而导致无法正常加载 NDIS 口驱动。有三种方式可以解决：

- 1) 比较新的 kernel 版本(3.8 以上)，在 option.c 中的 option_ids 中添加 blacklist，驱动在加载时会自动跳过 blacklist 指定的 interface；
设置 interface 4 不加载 option 驱动：

```
#ifdef 1 //Added by Nodecom
static const struct option_blacklist_info nodecom_nl650_blacklist = {
    .reserved = BIT(4),
};
#endif
```

添加 blacklist 到 option_ids 数组中：

```
#ifdef 1 //Added by Nodecom
    { USB_DEVICE(0x1508, 0x1001),
      .driver_info = (kernel_ulong_t)&nodecom_nl650_blacklist
    }
#endif
```

- 2) （推荐）对于之前的内核，不支持在 option_ids 数组中设置过 blacklist，要先增加 NL650/660 的 PID 和 VID：

```
static const struct usb_device_id option_ids[] = {
#ifdef 1 //Added by Nodecom
    { USB_DEVICE(0x1508, 0x1001) }, /* Nodecom NL650/660 */
#endif
}
```

在 probe 函数内判断当前 interface 号进行过滤：

```
#ifdef 1 //Added by Nodecom
    if (serial->dev->descriptor.idVendor == cpu_to_le16(0x1508) &&
        serial->dev->descriptor.idProduct == cpu_to_le16(0x1001) &&
        serial->interface->cur_altsetting->desc.bInterfaceNumber >= 4) {
        printk(KERN_INFO "Discover the 4th interface for NL650/660 NDIS\n");
        return -ENODEV;
    }
#endif
```

- 3) （不推荐）对于使用 usb-serial.ko 驱动的用户，需要在 usb-serial.c 文件中的 usb_serial_probe() 函数开始增加如下判断来过滤 NDIS 接口：

```
#ifdef 1 //Added by Nodecom
    if (interface->cur_altsetting->desc.bInterfaceNumber >= 4) {
        printk(KERN_INFO "Discover the 4th interface for NL650/660 NDIS\n");
        return -ENODEV;
    }
#endif
```

3 NDIS 驱动加载方法

3.1 NDIS 驱动添加系统组件

NDIS 驱动需要系统的 usbnet 驱动支持，因此需要配置 Linux 内核，配置方法如下：

```
cd kernel
```

```
make menuconfig
```

```
device drivers->Network device support->usb Network Adapters
```

选中如下组件

Multi-purpose USB Networking Framework

选中后保存配置，重新编译内核。

3.2 代码结构

如下图所示，驱动以源代码的形式提供，由用户在自己的系统编译。

```
driver/
├── GobiUSBNet.c
├── Makefile
├── QMI.c
├── QMIDevice.c
├── QMIDevice.h
├── QMI.h
└── Structs.h
```

3.3 驱动编译

用户可以单独编译，也可以将代码放入内核中一起编译。

1) 单独编译

修改 Makefile 中 KDIR 的值为 kernel 的编译路径；

在 driver 目录下执行 make 命令，编译成功就可以生成 GobiNet.ko 驱动；

2) 与内核一起编译

将代码文件复制到用户自己的 kernel 代码的 drivers/net/usb 目录下；

在 drivers/net/usb/Makefile 中增加以下内容：

```
obj-m += GobiNet.o
```

```
GobiNet-objs := GobiUSBNet.o QMIDevice.o QMI.o
```

之后每次编译内核都会自动编译 NDIS 驱动。

3.4 驱动的加载

1) 加载 USB 串口驱动

```
sudo modprobe option
```

使用 dmesg 命令查看系统 log，如下图所示，除了 NDIS 口，其他的 5 个端口都加载上 USB 串口驱动。

```

: Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#= 14 Spd=480 MxCh= 0
: Ver= 2.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs= 1
: Vendor=1508 ProdID=1001 Rev= 3.18
: Manufacturer=Nodecom NL660 Modem
: Product=Nodecom NL660-752E Modem
: SerialNumber=1be278
: * If#= 6 Cfg#= 1 Atr=a0 MxPwr=500mA
: * If#= 0 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=ff Prot=ff Driver=usbserial_generic
: Ad=81(I) Atr=02(Bulk) MxPS= 512 IvL=0ms
: Ad=01(O) Atr=02(Bulk) MxPS= 512 IvL=0ms
: * If#= 1 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=00 Prot=00 Driver=usbserial_generic
: Ad=83(I) Atr=03(Int.) MxPS= 10 IvL=32ms
: Ad=82(I) Atr=02(Bulk) MxPS= 512 IvL=0ms
: Ad=02(O) Atr=02(Bulk) MxPS= 512 IvL=0ms
: * If#= 2 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=00 Prot=00 Driver=usbserial_generic
: Ad=85(I) Atr=03(Int.) MxPS= 10 IvL=32ms
: Ad=84(I) Atr=02(Bulk) MxPS= 512 IvL=0ms
: Ad=03(O) Atr=02(Bulk) MxPS= 512 IvL=0ms
: * If#= 3 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=00 Prot=00 Driver=usbserial_generic
: Ad=87(I) Atr=03(Int.) MxPS= 10 IvL=32ms
: Ad=86(I) Atr=02(Bulk) MxPS= 512 IvL=0ms
: Ad=04(O) Atr=02(Bulk) MxPS= 512 IvL=0ms
: * If#= 4 Alt= 0 #EPs= 3 Cls=ff(vend.) Sub=ff Prot=ff Driver=GobiNet
: Ad=89(I) Atr=03(Int.) MxPS= 8 IvL=32ms
: Ad=88(I) Atr=02(Bulk) MxPS= 512 IvL=0ms
: Ad=85(O) Atr=02(Bulk) MxPS= 512 IvL=0ms
: * If#= 5 Alt= 0 #EPs= 2 Cls=ff(vend.) Sub=42 Prot=01 Driver=usbserial_generic
: Ad=06(O) Atr=02(Bulk) MxPS= 512 IvL=0ms
: Ad=8a(I) Atr=02(Bulk) MxPS= 512 IvL=0ms

```

2) 加载 NDIS 驱动

通过 insmod 命令加载 NDIS 驱动:

```
sudo insmod GobiNet.ko
```

驱动默认 debug 关闭, 可以采用如下方式开启:

```
ehco 1 > /sys/module/GobiNet/parameters/debug
```

通过 modprobe 命令加载 NDIS 驱动:

在 driver 目录执行 make install 命令:

```
sudo make install
```

会将驱动安装到系统的 module 目录中, 并且分析依赖关系, 将相关模块同时加载。

使用 ifconfig 命令查看网卡信息, 如果出现 usb0 表示驱动加载成功

```

usb0      Link encap:Ethernet  HWaddr 52:f4:dc:d6:ce:0a
          inet6 addr: fe80::50f4:dcff:fed6:ce0a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:670 (670.0 B)

```

到这里, 驱动的加载就全部结束。

4 NDIS 拨号测试

Nodecom 提供两种拨号方式:

- 1、快速拨号
- 2、QMI 拨号

4.1 快速拨号

4.1.1 设置 APN

通过 AT+CGDCONT 配置 APN，详细请参考 Nodecom 通信模块 AT 手册

4.1.2 拨号

通过 AT\$QCRMCall=1,1 进行拨号，详细请参考 Nodecom 通信模块 AT 手册

4.1.3 断网

通过 AT\$QCRMCall=0,1 进行断网，详细请参考 Nodecom 通信模块 AT 手册

示例：

```
ate0
```

```
OK
```

```
at+cops?
```

```
+COPS: 0,0,"CHN-UNICOM",7
```

```
OK
```

```
at$qcrmcalls=1,1,1,2,1
```

```
$QCRMCall:1, V4
```

```
OK
```

```
# udhcpc -i usb0 -s /usr/share/udhcpc/usb0.sh
```

```
/usr/share/udhcpc/usb0.sh: exec: line 3:
```

```
/usr/share/udhcpc/usb0.deconfig: not found
```

```
*****
```

```
sysconf conn dhcpc usb0 10.60.236.82 255.255.255.252 10.60.236.81
```

```
221.11.1.67 221.11.1.68
```

```
*****
```

```
Read hw setting header failed!
```

```
Invalid hw setting signature [sig=]!
```

```
Initialize AP MIB failed !
```

```
/bin/sh: can't create /proc/usb0/up_event: nonexistent directory
```


4.2 QMI 拨号

为了支持 QMI 拨号，且避免客户 QMI TLV 组包的困扰，Nodecom 提供一组 QMI SDK 和一个 QMI 拨号 Demo。完成拨号功能和其他信息的获取。（QMI 拨号完成后，需要启动 dhcpc 获取 IP）

4.2.1 运行测试程序

使用管理员权限运行 dial_demo，进行 NDIS 拨号，输入如下命令：

```
sudo apt-get install udhcpc
```

```
sudo ./dial_demo
```

初始化完成后输入 24 可以进行拨号，输入 25 断开网络连接，-1 退出程序。

拨号成功后手动启用 dhcp client 获取 IP 地址和 DNS 地址等，如下所示：

1)在 Ubuntu PC 上，执行以下命令：

```
# dhclient usb0
```

```
root@longsung:~/gobinet/version1.0/qmi_pc/nasclient# dhclient usb0
Internet Systems Consortium DHCP Client V3.1.3
Copyright 2004-2009 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/usb0/52:f4:dc:d6:ce:0a
Sending on LPF/usb0/52:f4:dc:d6:ce:0a
Sending on Socket/fallback
DHCPDISCOVER on usb0 to 255.255.255.255 port 67 interval 8
DHCPOFFER of 10.10.29.194 from 10.10.29.193
DHCPREQUEST of 10.10.29.194 on usb0 to 255.255.255.255 port 67
DHCPACK of 10.10.29.194 from 10.10.29.193
bound to 10.10.29.194 -- renewal in 2840 seconds.
```

2) 如果在嵌入式系统上，需要配置自己的 kernel 支持 dhcp client 功能，并且配置 busybox 支持

udhcpc 命令，由于篇幅限制，本文不再详述，方法请参考如下附件：

```
#Udhcpc -i usb0 -s /var/udhcpc.script
```

```
# route add default dev usb0
```

```
# iptables -t nat -A POSTROUTING -o usb0 -j MASQUERADE
```

```
#echo 1 > /proc/sys/net/ipv4/ip_forward
```



安装DHCP
client.docx

4.4 查看 IP 地址

输入 ifconfig 命令查看 usb0 的 IP 地址，如下图所示：

```
usb0    Link encap:Ethernet  Hwaddr 52:f4:dc:d6:ce:0a
        inet addr:10.10.29.194  Bcast:10.10.29.195  Mask:255.255.255.252
        inet6 addr: fe80::50f4:dcff:fed6:ce0a/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:43 errors:0 dropped:0 overruns:0 frame:0
        TX packets:68 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:7170 (7.1 KB)  TX bytes:12201 (12.2 KB)
```

4.5 测试网络链接

输入 `ping www.baidu.com` 命令，测试是否可以 `ping` 通网站；或者打开浏览器上网进行测试。