Fibocom

完 美 无 线 体 验

适用型号

序号	产品型号	说明
1	FG650&FM650 系列	NA

版权声明

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。

商标申明

为深圳市广和通无线股份有限公司的注册商标，由所有人拥有。

版本记录

文档版本	编写人	主审人	批准人	更新日期	说明
V1.0.1	段栋	王斌		2020-08-15	修改格式,添加 ubuntu 挂载说明
V1.0.0	段栋	陈勇		2020-08-07	初始版本

目录

1 前言

本文阐述 FG650&FM650 系列模块在 Linux 系统的 NCM ECM RNDIS 拨号方法（含驱动移植及应用）。本文主要面向 Linux 驱动及应用开发人员。

本文适用于：Linux2.6.22 及以上内核版本。

2 拨号模式简介

ECM（Ethernet Networking Control Model 以太网控制模型）用于在设备和主机之间传输以太网数据包。在操作系统看来，CDC ECM 设备就是一个虚拟以太网卡，包含标准网卡需要的 MAC 地址和 IP 地址。CDC ECM 设备通常是一个以太网卡，用于连接 LAN 或者是 WLAN。当客户主机发起 ECM 拨号的请求时， FG650&FM650 相当于一个路由器，模块内部会调用相应的服务实现 WWAN 拨号。在拨号成功后，模块内部会启动 DHCP server 等功能。客户端应用系统通过 DHCP client 服务，获取模块 DHCP server 分配的 IP。客户主机调用 DHCP 等脚本实现把模块从网络侧获取到的 IP 和 DNS 配置到本机，实现上网。

NCM （Network Control Model） 是 ECM 协议的改进版。将多个以太网包组成一个 NTB 包在 USB 总线上传输，提高了带宽利用率。

RNDIS（Remote Network Driver Interface Specification），既是远程网络驱动接口规范，基于 USB 实现 RNDIS 实际上就是 TCP/IP over USB，就是在 USB 设备上跑 TCP/IP，让 USB 设备看上去像一块网卡.

3 拨号模式选择

FG650&FM650 支持 ECM/NCM/RNDIS 三种模式，三种模式的主要特性如下表

	ECM	NCM	RNDIS
支持 Linux	Y	Y	Y
支持 Windows	N	N	Y
支持硬件加速	N	Y	Y

客户可以根据自己的需求，选择对应的端口模式。推荐使用 RNDIS 或 NCM 方式拨号。模组默认是 NCM 模式，如果想切换到另外一种模式，请参看第 4,5 章节。

4 USB 端口信息

表 4-1 USB 端口信息

USB Compositions		
USB Mode	**VID/PID**	**Compositions**
34	0x2CB7/0x0A04	ECM + AT + DIAG + LOG
35	0x2CB7/0x0A04	ECM + AT + DIAG + LOG +ADB
36	0x2CB7/0x0A05	NCM + AT + MODEM + DIAG + LOG
37	0x2CB7/0x0A05	NCM + AT + MODEM + DIAG + LOG + ADB

38	0x2CB7/0x0A06	RNDIS + AT + MODEM + DIAG + LOG
39	0x2CB7/0x0A06	RNDIS + AT + MODEM + DIAG + LOG +ADB

表 4-2 ECM 模式端口枚举顺序

Vendor ID: 0X2CB7 Product ID:0x0A04 (USBMODE 34/35)		
Interface Number	**Interface Function**	**Interface Name**
0-1	USB NET	CDC ECM
2	USB Serial	Fibocom AT
3	USB Serial	Fibocom Diag
4	USB Serial	Fibocom Log

表 4-3 NCM 模式端口枚举顺序

Vendor ID: 0X2CB7 Product ID:0x0A05(USBMODE 36/37)		
Interface Number	**Interface Function**	**Interface Name**
0-1	USB NET	CDC NCM
2	USB Serial	Fibocom AT
3	USB Serial	Fibocom Modem
4	USB Serial	Fibocom Diag
5	USB Serial	Fibocom Log

表 4-4 RNDIS 模式端口枚举顺序

Vendor ID: 0X2CB7 Product ID:0x0A06(USBMODE 38/39)		
Interface Number	**Interface Function**	**Interface Name**
0-1	USB NET	RNDIS
2	USB Serial	Fibocom AT
3	USB Serial	Fibocom Modem
4	USB Serial	Fibocom Diag
5	USB Serial	Fibocom Log

说明：

[1] 模块的 USB 支持多种模式，可以向模块发送 AT+GTUSBMODE?，查询当前处于什么模式， 也可以用 AT+GTUSBMODE=<mode>，调整 usb 模式，模块回复 OK 后重启模块生效。

[2] 如果模块枚举前已经存在同名设备，模块端口号会顺延。部分系统可能会默认修改设备名。

5 USB 枚举模式确认和修改

采用 AT+GTUSBMODE?查询当前 USB 模式。

AT+GTUSBMODE?

+GTUSBMODE: 39

OK

采用 AT+GTUSBMODE=<usbmode> 指令可以修改 USB 模式。

AT+GTUSBMODE?

+GTUSBMODE: 39

OK

AT+GTUSBMODE=37

OK

AT+CFUN=15 //重启模组生效

6 USB 串口驱动集成

6.1 USB 串口驱动添加系统组件

6.1.1 USB 串口驱动添加需要配置 Linux 内核，配置方法如下：

1) cd kernel 进入到内核根目录：

```
root@ubuntu:~/testSrc/android5.x_new# cd kernel/
root@ubuntu:~/testSrc/android5.x_new/kernel#
```

2) 输入“make menuconfig”后回车：

```
root@ubuntu:~/testSrc/android5.x_new/kernel# make menuconfig
```

3) 在弹出的界面中依次选择：

device drivers ->

```
    Power management options  --->
[*] Networking support  --->
    Device Drivers  --->
    File systems  --->
```

usb support - >

```
    <*> Sound card support  --->
    HID support  --->
[*] USB support  --->
    <*> MMC/SD/SDIO card support  --->
```

usb serial converter support - >

```
    *** USB port drivers ***
<*> USB Serial Converter support  --->
    *** USB Miscellaneous drivers ***
```

选中如下 USB driver for GSM and CDMA modems 组件，选中后保存退出。

```
< >  USB Xircom / Entregra Single Port Serial Driver
<*>  USB driver for GSM and CDMA modems
< >  USB ZyXEL omni.net LCD Plus Driver
```

6.1.2 增加设备支持（以 FG650 为例）

1: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 FG650 的 PID/VID。 注意 ECM/NCM/RNDIS 等不同模式下的 PID 是不一样的。需要根据模组的模式添加相应的 PID/VID（请参考 USB 端口章节），或者可以把 FG650 支持的所有模式 ID 都添加上去。

#define FIBOCOM_ VENDOR_ID 0x2CB7

#define FIBOCOM_PRODUCT_FG650_ECM 0x0A04

#define FIBOCOM_ PRODUCT_FG650_NCM 0x0A05

```
#define FIBOCOM_ PRODUCT_FG650_RNDIS 0x0A06
static const struct usb_device_id option_ids[] = {
… …
#if 1
{ USB_DEVICE(FIBOCOM_ VENDOR_ID, FIBOCOM_PRODUCT_FG650_ECM) },
{ USB_DEVICE(FIBOCOM_ VENDOR_ID, FIBOCOM_ PRODUCT_FG650_NCM) },
{ USB_DEVICE(FIBOCOM_ VENDOR_ID, FIBOCOM_ PRODUCT_FG650_RNDIS) },

#endif
… …
```

2:在 USB 串口驱动中，过滤 ECM/NCM/RNDIS 接口。由于 USB 串口跟 ECM/NCM/RNDIS 都属于非标准 CDC 设备，需要防止 ECM/NCM/RNDIS 口被 USB 串口驱动加载而导致无法正常加载 ECM/NCM/RNDIS 口驱动。有三种方式可以解决：

A：比较新的 kernel 版本(3.8 以上)，在 option.c 中的 opiton_ids 中添加 blacklist，驱动在加载时会自动跳过 blacklist 指定的 interface，具体代码如下：

```
#if 1
static const struct option_blacklist_info fibocom_FG650_blacklist_ECM = {
    .reserved = BIT(0) | BIT(1)| BIT(5),
};
static const struct option_blacklist_info fibocom_FG650_blacklist_NCM = {
    .reserved = BIT(0) | BIT(1)| BIT(6),
};
static const struct option_blacklist_info fibocom_ FG650_blacklist_RNDIS = {
    .reserved = BIT(0) | BIT(1) | BIT(6),
};


#endif
```

添加 blacklist 到 option_ids 数组中，具体代码如下：

```
#if 1
{ USB_DEVICE(FIBOCOM_ VENDOR_ID, FIBOCOM_PRODUCT_FG650_ECM),
    .driver_info = (kernel_ulong_t)&fibocom_blacklist_ECM
}
{ USB_DEVICE(FIBOCOM_ VENDOR_ID, FIBOCOM_ PRODUCT_FG650_NCM),
```

```
        .driver_info = (kernel_ulong_t)&fibocom_blacklist_NCM
    }
    { USB_DEVICE(FIBOCOM_ VENDOR_ID, FIBOCOM_ PRODUCT_FG650_RNDIS),
        .driver_info = (kernel_ulong_t)&fibocom_blacklist_RNDIS
    }

    #endif
```

B:(推荐)对于之前的内核，不支持在 option_ids 数组中设置 blacklist，要先增加 FG650 系列的 PID/VID，具体如下：

```
    static const struct usb_device_id option_ids[] = {
    #if 1
    { USB_DEVICE(FIBOCOM_ VENDOR_ID, FIBOCOM_PRODUCT_FG650_ECM)},
    { USB_DEVICE(FIBOCOM_ VENDOR_ID, FIBOCOM_ PRODUCT_FG650_NCM)},
    { USB_DEVICE(FIBOCOM_ VENDOR_ID, FIBOCOM_ PRODUCT_FG650_RNDIS)},
    #endif
```

并在 probe 函数内判断当前的 interface num 进行过滤，具体如下：

```
    #if 1
        if (serial->dev->descriptor.idVendor == cpu_to_le16(FIBOCOM_ VENDOR_ID) &&
        serial->dev->descriptor.idProduct == cpu_to_le16(FIBOCOM_PRODUCT_FG650_ECM) &&
        serial->interface->cur_altsetting->desc.bInterfaceNumber <= 1) {
        printk(KERN_INFO “Discover the 4th interface for fibocom\n”);
        return –ENODEV;

        if (serial->dev->descriptor.idVendor == cpu_to_le16(FIBOCOM_ VENDOR_ID) &&
        serial->dev->descriptor.idProduct == cpu_to_le16(FIBOCOM_ PRODUCT_FG650_NCM) &&
        serial->interface->cur_altsetting->desc.bInterfaceNumber <= 1) {
        printk(KERN_INFO “Discover the 4th interface for fibocom\n”);
        return –ENODEV;

        if (serial->dev->descriptor.idVendor == cpu_to_le16(FIBOCOM_ VENDOR_ID) &&
        serial->dev->descriptor.idProduct == cpu_to_le16(FIBOCOM_ PRODUCT_FG650_RNDIS) &&
        serial->interface->cur_altsetting->desc.bInterfaceNumber <= 1) {
        printk(KERN_INFO “Discover the 4th interface for fibocom\n”);
        return –ENODEV;
```

}

#endif

C: (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()函数中开始处，增加如下判断来过滤 ECM/NCM/RNDIS 接口，具体如下：

```
#if 1
    if (serial->cur_altsetting->desc.bInterfaceNumber <= 1) {
    printk(KERN_INFO  “Discover the 4th interface for fibocom\n”);
    return  -ENODEV;}

#endif
```

6.2 USB 串口驱动加载确认

执行 lsusb，检查是否检测到模组 USB，如果找到对应模式的 VID/PID，表示 USB 已正常检测到再通过 ls /dev/ttyUSB* 检查 USB 串口驱动是否正常加载。

```
root@ubuntu:/# lsusb
Bus 001 Device 005: ID 2cb7:0a04
Bus 002 Device 002: ID 0e0f:0003 VMware, Inc. Virtual Mouse
Bus 002 Device 003: ID 0e0f:0002 VMware, Inc. Virtual USB Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 004: ID 0e0f:0008 VMware, Inc.
root@ubuntu:/# ls /dev/ttyUSB*
/dev/ttyUSB0       /dev/ttyUSB1    /dev/ttyUSB2
```

7 内核环境确认

ECM/NCM 是一种标准的设备间以太网帧数据交换协议，系统中需要加载 CDC ETHER 驱动，RNDIS 需要配置 CONFIG_USB_NET_RNDIS_HOST 请参考如下步骤。

7.1 内核编译配置项

CONFIG_USB_NET_CDCETHER=y

CONFIG_USB_USBNET=y

CONFIG_USB_NET_RNDIS_HOST = y

7.2 内核编译配置步骤

1: 进入 kernel 目录（假定为"/home/ght/linux-3.08/"）,make <configuration>命令（假定使用标准 make menuconfig）。

2: 按照下述指引完成 ECM 驱动配置：

进入 **Device Drivers –> Network device support -> USB Network Adapters** 菜单后选择 **Multi-purpose USB Networking Framework**：

如果是 ECM 模式，选择：

CDC Ethernet support（smart devices such as cable modems）

如果是 NCM 模式，选择：

CDC Ethernet support（smart devices such as cable modems）

CDC NCM Support

如果是 RNDIS 模式，选择：

Host for RNDIS and ActiveSync devices

注：可以把所有选项都选中，支持所有模式。

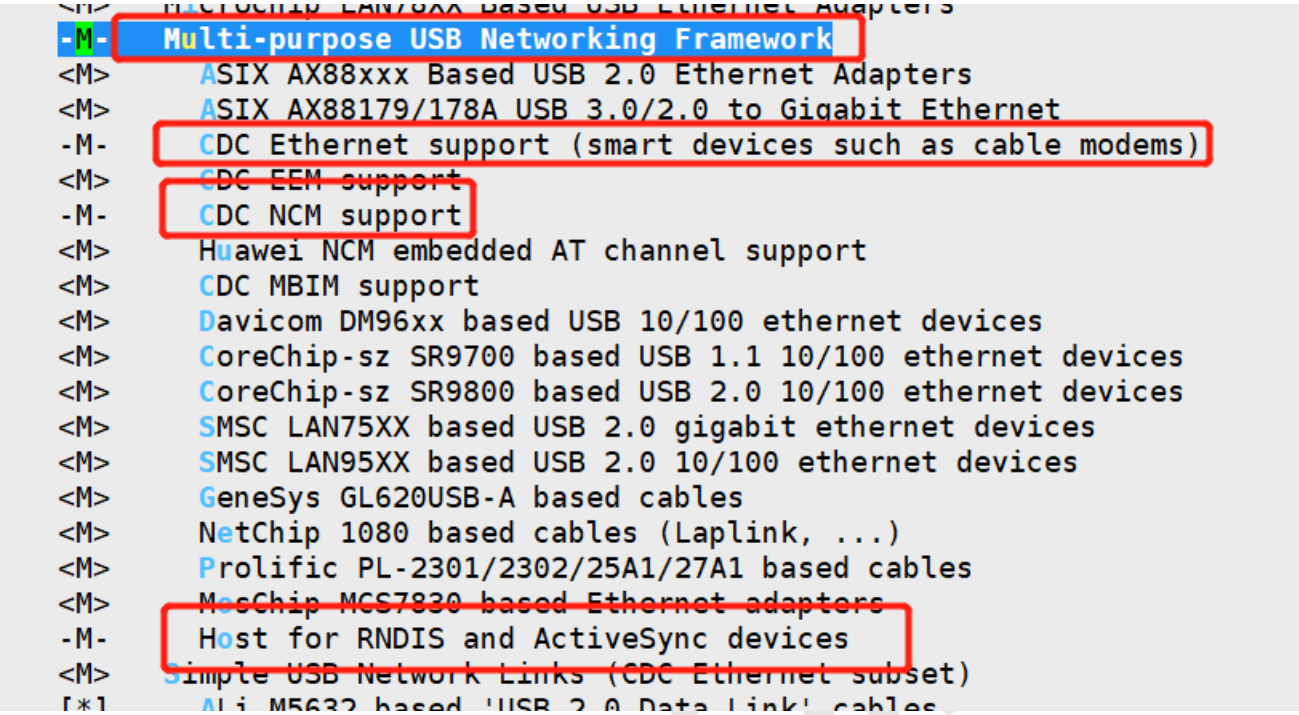


图 6-1 ECM 驱动配置

3: 如上配置后，通过选择“<Exit>”逐级退出配置界面。最后在保存界面中，选择“<Yes>”并退出。

4: 完成配置后，运行 make 命令，编译修改后的内核。

注意：

使用 ECM 模式，需要特别注意 Linux 系统相关部分是否被客户修改过。特别是 /kernel/drivers/net/usb/cdc_ether.c 中下不要删除数组 products[]中的如下项：

{USB_INTERFACE_INFO(USB_CLASS_COMM, USB_CDC_SUBCLASS_ETHERNET, USB_CDC_PROTO_NONE),

.driver_info = (unsigned long) &cdc_info,}

7.3 ECM/NCM/RNDIS 驱动加载确认

1: 确认 cdc_ether.ko 或者 cdc_ether.o 已经在内核中。如果*.ko 没有加载，用以下方法加载：

root@ubuntu:/# insmod cdc_ether.ko

或者

root@ubuntu:/#

cdc_ether

2: 加载后，执行如下确认是否加载成功：

root@ubuntu:/# lsmod | grep cdc_ether //确认驱动加载是否成功

cdc_ether 13502 0

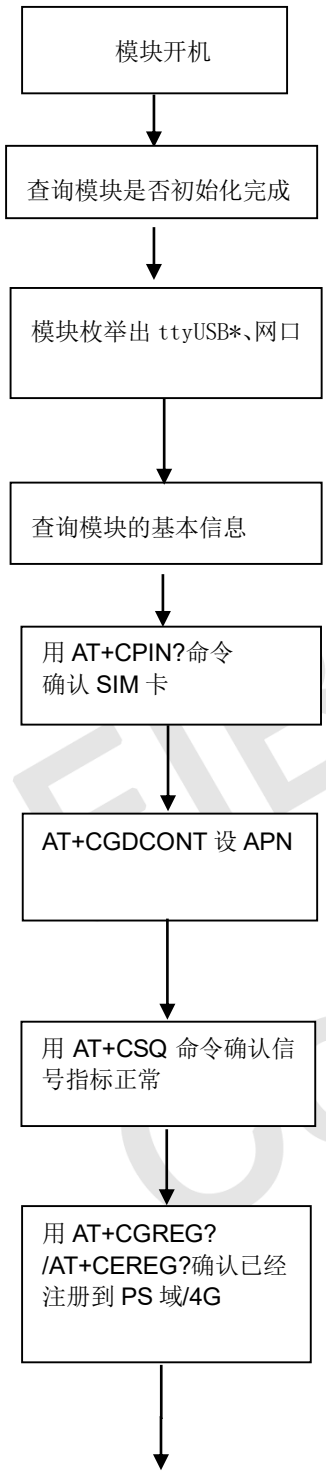
usbnet 31972 1 cdc_ether

使用 ifconfig 命令参查询到以下设备（不同系统不同版本内核下面这个名字可能不一样，视具体情况为准）：

```
usb0      Link encap:Ethernet  HWaddr ae:2a:e8:41:31:1f
          inet6 addr: fe80::ac2a:e8ff:fe41:311f/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

8 拨号流程

8.1 拨号参考流程图



模块开机： 模块的供电稳定后。
模块关机

使用 lsusb 查询模块是否初始化完成，如果完成可以看到模块的 VID/PID

最少等 90 秒。仍然没有端口则认为本次开机失败。重新进入模块开机动作。
或者延时 5 秒后连续给模块发 AT，等模块回复 OK。最长等 90 秒。

AT+CGSN? 查询 IMEI，每个模块唯一
AT+CFSN? 查询批次号 S/N
AT+CFUN? 回复+CFUN: 1 确认模块正常工作模式。不是 1 则设置成 1
AT+CGMR? 查询模块当前的固件版本号

回复+CPIN: READY 说明识别到 SIM 卡。
假如最少连续查询 90 秒仍然找不到 SIM 卡，复位模块。
如果需要 PIN 码则先用 AT+CPIN 设置 PIN 码

APN 是模块即将选用的运营商网络的标识。建议联系 SIM 供应商确认设置对应的 APN。如果有多种 SIM 卡，可以在注册 PS 域后，设置 AT+COPS=3,2 后根据 AT+COPS? 查询结果设置。
专网下用户名和密码一般由运营商提供，如果没有提供，电信卡请以‘card’‘card’，进行配置用户名以及密码。鉴权类型也由运营商确定，一般是 PAP、CHAP 或者两者都有，默认情况下或者是运营商没有提供的话，请使用 PAP 或者 PAP&CHAP。

返回值+CSQ: <rssi>,<ber> 第一个参数 <rssi>大于 0 并且不是 99。
假如连续查 90 秒返回值不正确，则复位模块。

查询结果是 1 或者 5 表示注册上 PS 域，连续查询超过 90 秒仍然注册不上 PS 域，复位模块。[2]

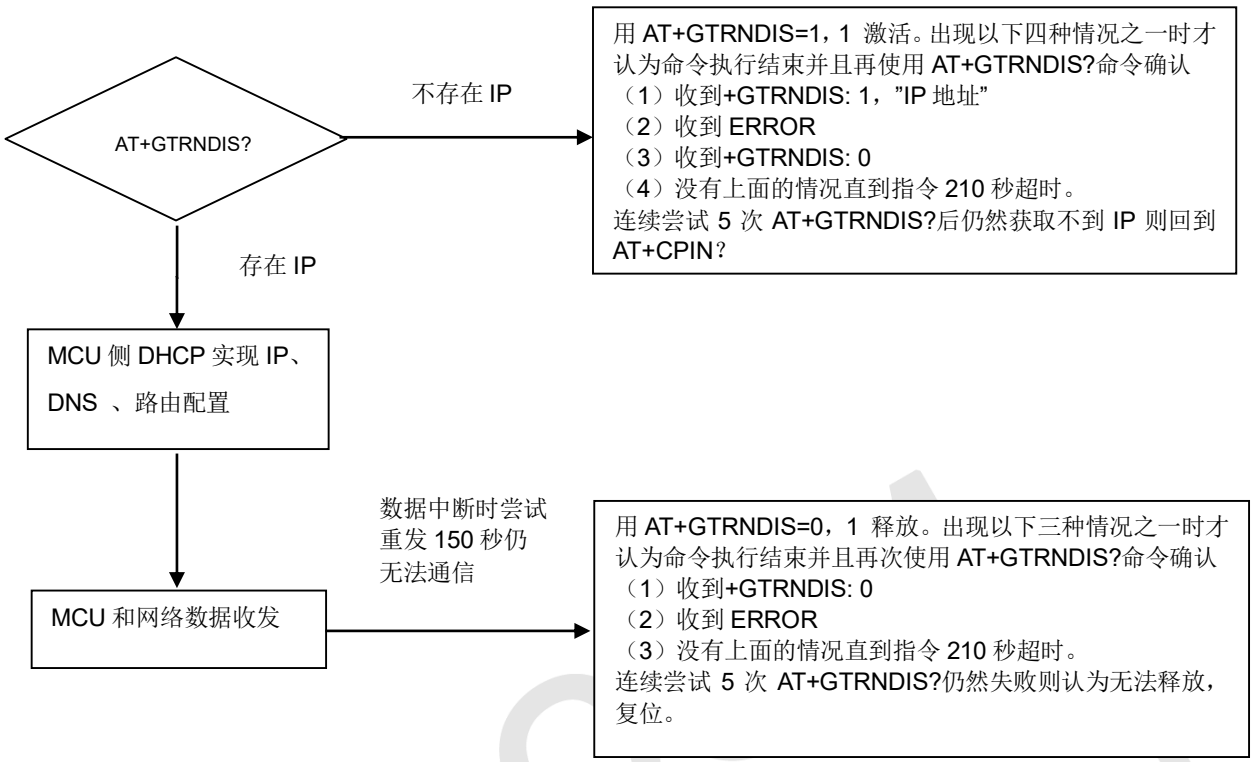


图 8-1 拨号流程

8.2 关键 AT 命令说明

AT+GTRNDIS 命令格式如下表 7-1：

表 7-1 AT+GTRNDIS 命令格式

Command	Possible response(s)
+GTRNDIS=<state>,<cid>	OK or ERROR
+GTRNDIS?	+GTRNDIS: <state>,<cid>,<ip>,<pdns>,<sdns> OK or +GTRNDIS: 0
AT+ GTRNDIS=?	+GTRNDIS =(list of supported <state>s), (list of supported <cid>s) OK

<state>: integer type

0 deactivate RNDIS. Default value.

1 active RNDIS

<cid>: int 型数据，用于 ECM 拨号的 profile id ，FG650&FM650 支持范围是（1-6）。

<ip>: string 类型;PDN 激活后网络分配给 ECM 的 IP 地址。

<pdns>: string 类型; PDP 激活网络分配的主 DNS。

<sdns>: string 类型; PDP 激活网络分配的辅 DNS。

8.3 相关 AT log 和说明

```
//下面是相关指令的参考 log 和相关注释
//如果需要上网，拨号流程推荐（以电信卡为例），具体如下：
AT+CPIN?
+CPIN: READY                                    //确保 SIM 卡就绪

OK
AT+CSQ
+CSQ: 21,99                                     //确保模块能接收到信号

OK
AT+CGDCONT=1,"IP","ctlte"                       //设置 APN 接入点信息
OK

AT+COPS?
+COPS: 0,0,"CHN-TELECOM",7                      //确保模块注册上运营商网络

OK

AT+CGREG?
+CGREG: 0,1                                     //模块附着数据网络

OK
AT+CEREG?                                       //模块附着 LTE 网络
+CEREG: 0,1

OK

AT+CGDCONT?
```

+CGDCONT: 1,"IP","ctlte","0.0.0.0",0,0,0,0 //查询是否设置成功

OK

AT+GTRNDIS=1,1

OK

……

AT+GTRNDIS? //查询拨号状态，查看模块是否已经获取到 IP。

+GTRNDIS: 1,1,"100.85.126.41","202.101.172.37", "202.101.173.157"

OK

……

//AT+GTRNDIS 返回 OK 后，下发 AT+GTRNDIS?，查询拨号是否成功？命令返回 IP 后，客户端需要

//启动 DHCP client 获取 IP 。（例如输入 udhcpc –i usb0 & 或 dhclient usb0，后台运行 dhcp 服务。）

//dhcp 成功获取 IP 地址打印信息如下：

Sending select for 192.168.0.2...

Lease of 100.85.126.41 obtained, lease time 7200

/usr/share/udhcpc/default.script: Resetting default routes

SIOCDELRT: No such process

/usr/share/udhcpc/default.script: Adding DNS 192.168.0.1

//如果需要断开拨号，先 kill 掉 dhcp client 相关进程后，并下发如下命令释放 IP：

AT+GTRNDIS=0,1

OK

AT+GTRNDIS?

+GTRNDIS: 0

OK

9 常见 FAQ

9.1 如何把模块挂载到 Windows 下的 Ubuntu 虚拟机

将模块 USB 连接到 PC 上，启动 Ubuntu 虚拟机后找到菜单栏，再按如下方式找到模块连接和断开的选项：

虚拟机⟶可移动设备⟶ FG650&FM650 Module ⟶ 连接（断开与主机的连接），即可将模块挂载到 ubuntu 上

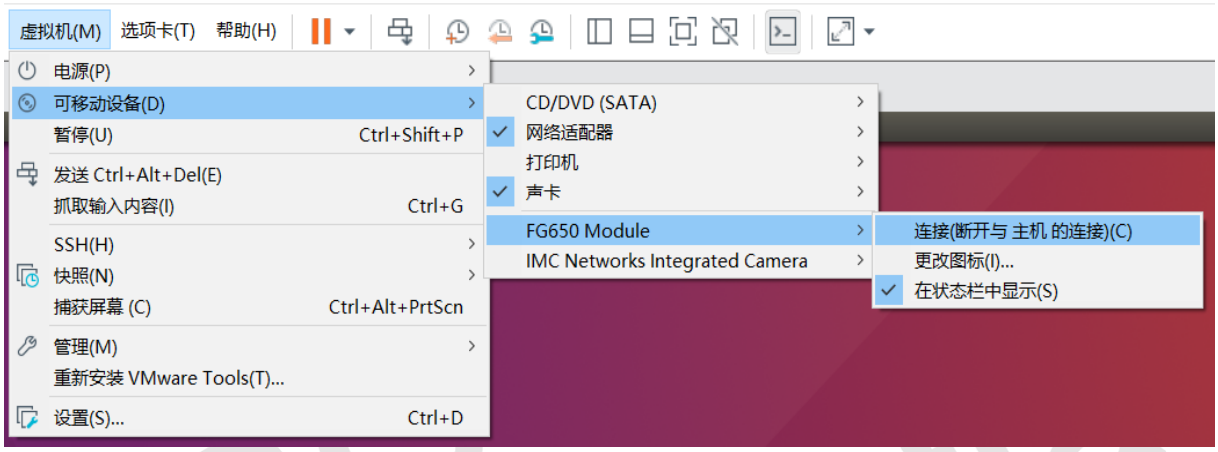


图 9-1 Ubuntu 虚拟机连接模块

9.2 执行 lsusb 和 dmesg 后没有看到模块的 PID 和 VID

执行 lsusb 活 dmesg，如果没有 usb 端口章节中的表 1 中的 ID，那么可能是硬件问题或者还没有枚举完成。如果已经找到 2cb7 的 vid，只是 pid 不对，那么可能是 USB 设置不对，请参考表 4-1，通过 AT+GTUSBMODE 查询和设置相应的模式。

（1）当模块 GTUSBMODE 是 39 时，如下图所示。

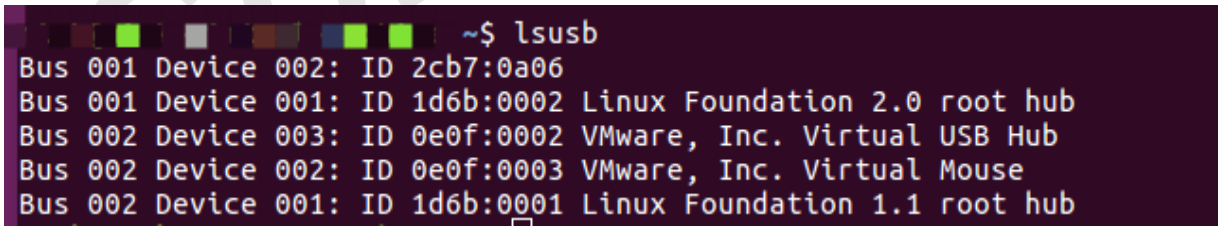


图 9-2 lsusb 打印

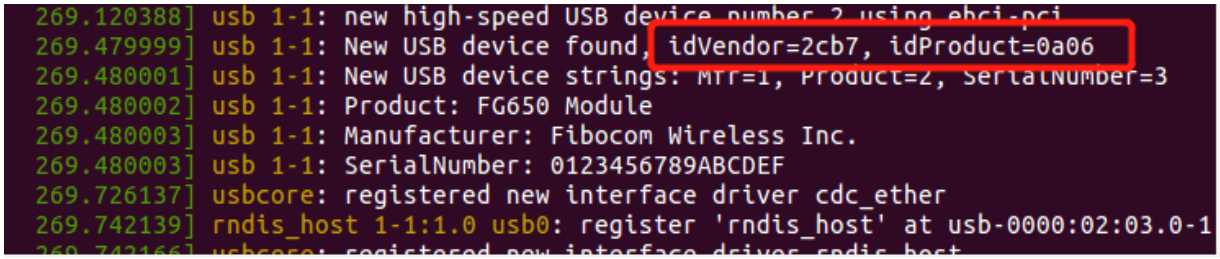


图 9-3 dmesg 打印

切换 USB 模式的方法： 模块的 USB 口接 PC，并且模块正常开机。在 Windows 下用串口工具或者 Linux 用 minicom 等工具发送 AT+GTUSBMODE=<mode>，再重启模块即可切换模式。

```
AT
OK
AT+GTUSBMODE=35
OK
AT+GTUSBMODE?
+GTUSBMODE: 35

OK
```

图 9-4 设置 USBMODE

（2）当 GTUSBMODE 是 34-37 时，USB 处于 ECM 或 NCM 模式，Windows 默认并不支持 ECM 和 NCM 所以 Winwos 会显示驱动为安装。 如下 u

在 Windows 下没有装驱动，设备管理器提示如图 9-5：

图 9-5 win 下 ECM 网卡

9.3 如何加载模块端口（以 FG650 为例）

如果是 Ubuntu 下，当模块正确开机并且 USB 挂载到 Ubuntu 系统后，dmesg 能看到上面的 PID 和 VID。

把以下内容保存成 FG650.sh 然后 chmod 777 FG650.sh 并./ FG650.sh，能看到模块端口。

KVERSION=`uname -r`

echo "################ insmod FG650 drivers #################\r\n"

```
insmod /lib/modules/$KVERSION/kernel/drivers/usb/serial/usbserial.ko
insmod /lib/modules/$KVERSION/kernel/drivers/usb/serial/usb_wwan.ko
insmod /lib/modules/$KVERSION/kernel/drivers/usb/serial/option.ko
echo "############### load FG650 drivers #################\r\n"
echo "2cb7 0a04" > /sys/bus/usb-serial/drivers/option1/new_id
echo "2cb7 0a05" > /sys/bus/usb-serial/drivers/option1/new_id
echo "2cb7 0a06" > /sys/bus/usb-serial/drivers/option1/new_id
echo "############### List Serial Port And ECM Port #################\r\n"
ls -al /dev/ttyUSB*
ls /sys/class/net
echo "############### ifconfig   #################\r\n"
ifconfig
```

9.4 无法上网时如何诊断

假定当前模块的 USB 模式是 GTUSBMODE 39，枚举出的端口是 /dev/ttyUSB0、/dev/ttyUSB1、/dev/ttyUSB2、网口 USB0（有的是 ens35u1）。注意模块枚举前已经如果存在同名设备，模块端口号会顺延。

可以使用 minicom 工具下发 AT 的方式 通过 ttyUSB0 这个 AT 口来查看状态。

root 权限下运行命令：minicom –D /dev/ttyUSB0

打开 minicom。

（1）确认 SIM 卡

```
at+cpin?

+CPIN: READY

OK
at+csq?

+CSQ: 25,99

OK
at+creg?

+CREG: 0,1

OK
at+cops?

+COPS: 0,2,"46001",7

OK
```

图 9-6 注网查询

（2）不能识别 SIM 卡时查看错误原因

```
at+cpin?

ERROR
at+cmee=2

OK
at+cpin?

+CME ERROR: SIM not inserted
```

图 9-7 SIM 未识别

（3）尝试拨号

```
at+gtrndis=1,1

OK
at+gtrndis?

+GTRNDIS: 1,1,"10.126.191.25","0.0.0.0","0.0.0.0"

OK
```

图 9-8 发起 ECM 拨号

（4）ifconfig 中可以看到 ECM 网卡 usb0(或 ens35u1)获取到 IP 地址。

```
[redacted]~$ ifconfig usb0
usb0      Link encap:以太网  硬件地址 4a:cc:d9:c0:f0:02
          inet 地址:192.168.0.2  广播:192.168.0.255  掩码:255.255.255.0
          inet6 地址: 2408:84fb:121a:43dc:5d55:23bf:906d:de6e/64 Scope:Global
          inet6 地址: 2408:84fb:121a:43dc:d00a:43eb:f643:b915/64 Scope:Global
          inet6 地址: 2408:84fb:121a:43dc:4961:ba56:4594:39d6/64 Scope:Global
          inet6 地址: 2408:84fb:121a:43dc:98b4:8a7b:767c:2969/64 Scope:Global
          inet6 地址: fe80::4c8d:e77a:afa8:cae2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
          接收数据包:2678 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:2729 错误:1 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:3547845 (3.5 MB)  发送字节:262816 (262.8 KB)
```

图 9-9 ECM 网卡获取到 IP

（5）可以 ping 通 DNS 地址和外网 IP。

```
[redacted]:~$ ping www.baidu.com
PING www.baidu.com (61.135.169.125) 56(84) bytes of data.
64 bytes from 61.135.169.125: icmp_seq=1 ttl=53 time=128 ms
64 bytes from 61.135.169.125: icmp_seq=2 ttl=53 time=41.0 ms
64 bytes from 61.135.169.125: icmp_seq=3 ttl=53 time=53.7 ms
64 bytes from 61.135.169.125: icmp_seq=4 ttl=53 time=38.6 ms
64 bytes from 61.135.169.125: icmp_seq=5 ttl=53 time=44.3 ms
64 bytes from 61.135.169.125: icmp_seq=6 ttl=53 time=54.4 ms
64 bytes from 61.135.169.125: icmp_seq=7 ttl=53 time=41.3 ms
64 bytes from 61.135.169.125: icmp_seq=8 ttl=53 time=66.8 ms
```

图 9-10　IPV4 ping 包验证

```
[redacted]:~$ ping6 ipv6.sjtu.edu.cn
PING ipv6.sjtu.edu.cn(2001:da8:8000:1::80) 56 data bytes
64 bytes from 2001:da8:8000:1::80: icmp_seq=1 ttl=44 time=62.9 ms
64 bytes from 2001:da8:8000:1::80: icmp_seq=2 ttl=44 time=62.9 ms
64 bytes from 2001:da8:8000:1::80: icmp_seq=3 ttl=44 time=65.4 ms
64 bytes from 2001:da8:8000:1::80: icmp_seq=4 ttl=44 time=65.9 ms
64 bytes from 2001:da8:8000:1::80: icmp_seq=5 ttl=44 time=69.2 ms
64 bytes from 2001:da8:8000:1::80: icmp_seq=6 ttl=44 time=68.1 ms
64 bytes from 2001:da8:8000:1::80: icmp_seq=7 ttl=44 time=59.7 ms
```

图 9-11 IPV6 ping 包验证