

QMI\_WWAN驱动集成及拨号指南\_Linux（桌面）

V2.2

版权声明

版权所有©2021深圳市广和通无线股份有限公司。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。

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

G:\repos\cbb\zh\static\images\logo.png 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。

联系方式

公司网址： https://www.Fibocom.com/

总部地址：深圳市南山区西丽街道西丽社区打石一路深圳国际创新谷六栋A座10-14层

总机：+86 755-26733555

目录

[修订记录 2](#_Toc86855307)

[1 引言 3](#_Toc86855308)

[1.1 目的 3](#_Toc86855309)

[1.2 适用范围 3](#_Toc86855310)

[1.3 预置条件 3](#_Toc86855311)

[2 Ubuntu环境下Host侧驱动编译环境搭建 5](#_Toc86855312)

[3 Fedora环境下Host侧驱动编译环境搭建 7](#_Toc86855313)

[3.1 Fedora系统信息 7](#_Toc86855314)

[3.2 Fedora 29/30下编译环境搭建 7](#_Toc86855315)

[4 OpenSUSE环境下Host侧驱动编译环境搭建 10](#_Toc86855316)

[4.1 OpenSUSE维护支持系统信息 10](#_Toc86855317)

[4.2 OpenSUSE Leap 15.0/15.1下编译环境搭建 10](#_Toc86855318)

[5 集成Fibocom模块QMI\_WWAN驱动信息 12](#_Toc86855319)

[5.1 QMI\_WWAN驱动代码结构 12](#_Toc86855320)

[5.2 QMI\_WWAN驱动编译 12](#_Toc86855321)

[5.3 检测Fibocom模块设备加载 13](#_Toc86855322)

[6 Fibocom模块拨号 14](#_Toc86855323)

[6.1 QMI拨号流程 14](#_Toc86855324)

[6.2 QMI单路拨号 15](#_Toc86855325)

[6.3 QMI多路拨号 17](#_Toc86855326)

# 修订记录

|  |  |
| --- | --- |
| V2.2（2021-11-02） | 增加适配模块型号，FM101系列，FM130系列和FM160系列 |
| V2.1（2021-08-25） | 增加多路拨号描述 |
| V2.0（2021-06-30） | 对外发布文档采用新版本，版本号从三位数修改为两位数 |
| V 1.0.0（2021-02-05） | 初始版本 |

# 引言

桌面Linux发行版本众多，Fibocom模块Host侧驱动基于Ubuntu、OpenSUSE、Fedora三大主流系统进行编译使用。

## 目的

本文主要介绍桌面Linux下Host侧QMI\_WWAN驱动在Ubuntu、OpenSUSE、Fedora编译环境搭建方法，为Fibocom模块产品在桌面Linux上正常使用QMI\_WWAN驱动提供指导。

## 适用范围

适用于测试部/市场导入人员参考文档或指导客户搭建FIBOCOM 模块产品Host驱动编译环境使用。

QMI\_WWAN驱动目前适用的产品型号如下表所示：

表1. 适用型号

|  |  |  |
| --- | --- | --- |
| 序号 | 产品型号 | 说明 |
| 1 | NL95X系列 | M.2接口4G通信模组。 |
| 2 | FG150系列 | LGA封装，5G通信模组。 |
| 3 | FM150系列 | M.2接口5G通信模组。 |
| 4 | NL668系列 | NA |
| 5 | FM100系列 | M.2接口4G通信模组 |
| 6 | FG101系列 | LGA封装，5G通信模组。 |
| 7 | FM101系列 | M.2接口4G通信模组。 |
| 8 | FM130系列 | M.2接口5G通信模组。 |
| 9 | FM160系列 | M.2接口5G通信模组。 |

## 预置条件

1. PC需要安装下列的一种桌面Linux系统：
   * Ubuntu Desktop 16.04 LTS
   * Ubuntu Desktop 18.04 LTS
   * Ubuntu Desktop 19.04 LTS
   * OpenSUSE Leap 15.0
   * OpenSUSE Leap 15.1
   * Fedora 29 Workstation
   * Fedora 30 Workstation
2. PC已预置Fibocom模块
3. 用户具有root操作权限

# Ubuntu环境下Host侧驱动编译环境搭建

Ubuntu当前支持系统如下表所示，其中Ubuntu 16.04/19.04缺省情况下，默认安装了c/c++的编译环境，所以Ubuntu 16.04/19.04系统上，不需要安装额外编译环境，可以直接编译Fibocom模块Host侧驱动。

表2. Ubuntu系统版本信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ubuntu版本 | 发布日期 | 桌面版本支持结束时间 | 内核版本 | Make版本 | GCC版本 |
| 16.04 LTS | 2016-04-21 | 2021-04 | 4.4 | 4.1 | 5.4.0 |
| 18.04 LTS | 2018-04-26 | 2023-04 | 4.15 | NA | NA |
| 19.04 | 2019-04-18 | 2020-01 | 5.0 | 4.2.1 | 8.3.0 |

Ubuntu 18.04缺省情况下，未提供c/c++的编译环境，如下图示：

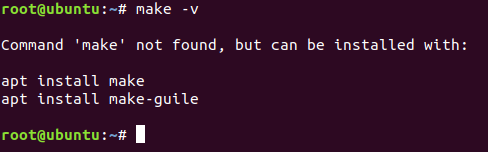


图1. 未提供make的编译环境

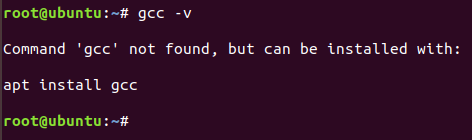


图2. 未提供gcc的编译环境

安装make：

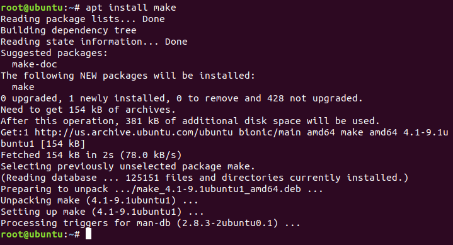


图3. 安装make

安装gcc工具：

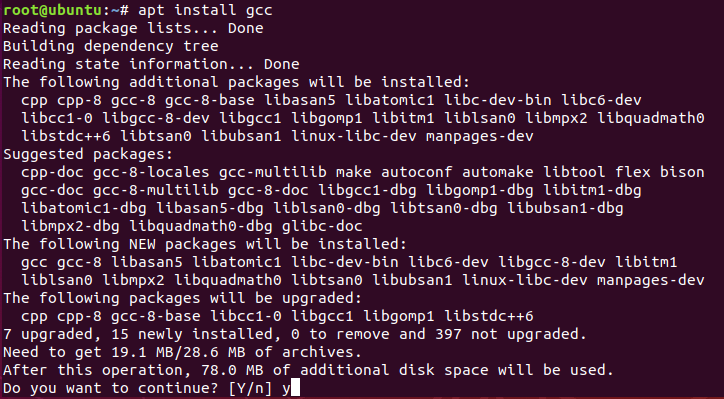


图4. 安装gcc工具

在安装gcc时，出现“Do you want to continue”提示后，输入y继续安装即可。

# Fedora环境下Host侧驱动编译环境搭建

## Fedora系统信息

Fedora仍被支持系统如下表所示，缺省情况下，未安装c/c++的编译环境，所以在Fedora系统上，需要安装额外编译环境后，再进行Fibocom模块Host侧驱动编译工作。

表3. Fedora系统版本信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fedora版本 | 发布日期 | 内核版本 | Make版本 | GCC版本 |
| Fedora 29 | 2018-10-30 | 4.19 | 8.2.1 | NA |
| Fedora 30 | 2019-04-29 | 5.0 | NA | NA |

## Fedora 29/30下编译环境搭建

安装kernel开发包：

1. 输入“**sudo -s**”切换到root模式：
2. 输入“**yum install kernel-devel-$(uname -r)**”命令，安装kernel开发包，如下图示：

出现“**Is this ok[y/N]**”时，输入“**y**”回车继续。

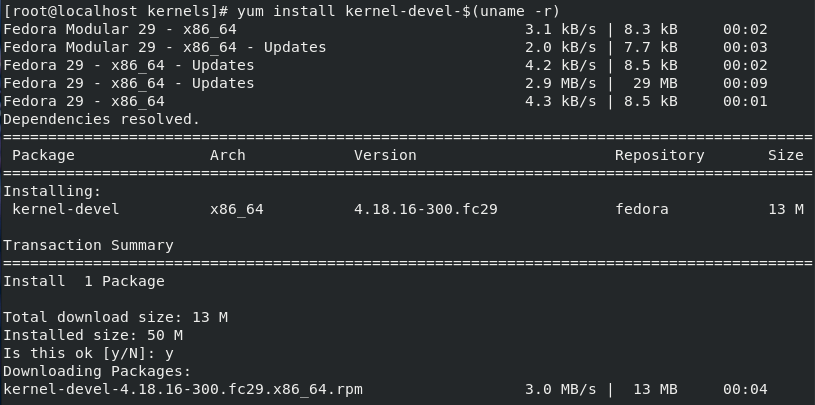


图5. 安装kernel开发包

|  |  |  |  |
| --- | --- | --- | --- |
|  | C:\Users\Administrator\Desktop\图层 4.png |  | 本文档基于Fedora 29搭建，用户请参照实际运行Fedora系统环境提示进行。 |

安装elfutils-libelf-devel工具：

1. 输入“**sudo -s**”切换到root模式：
2. 输入“**yum install elfutils-libelf-devel**”命令，安装elfutils-libelf-devel，如下图示：

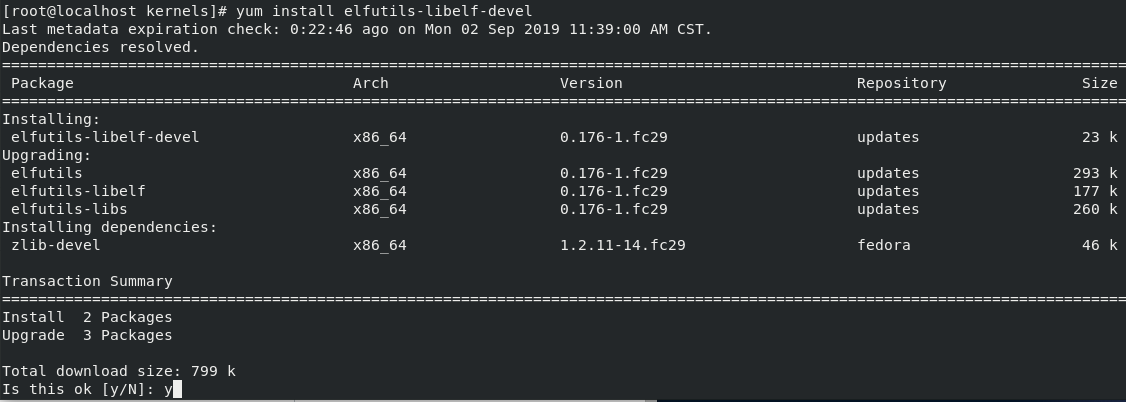


图6. 安装elfutils-libelf-devel

出现“**Is this ok[y/N]**”时，输入“**y**”回车继续。

安装make工具：

输入“**yum install make**”命令，安装make工具，如下图示：

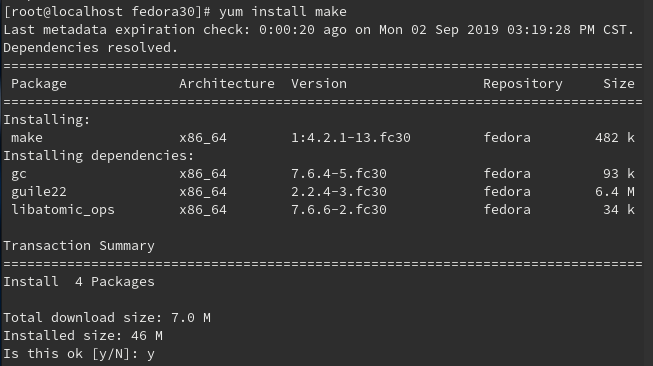


图7. 安装make工具

在出现“**Is this ok[y/N]**”时，输入“**y**”继续make工具的安装。

|  |  |  |  |
| --- | --- | --- | --- |
|  | C:\Users\Administrator\Desktop\图层 4.png |  | 上述工具安装完成后，请重启系统。 |

# OpenSUSE环境下Host侧驱动编译环境搭建

## OpenSUSE维护支持系统信息

OpenSUSE仍被维护支持的系统如下表所示，缺省情况下，未安装c/c++的编译环境，在OpenSUSE系统上，也需要安装额外编译环境后，才能进行Fibocom模块Host侧驱动编译。

表4. OpenSUSE维护支持系统版本信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OpenSUSE版本 | 发布日期 | 内核版本 | Make版本 | GCC版本 |
| OpenSUSE Leap 15.0 | 2018-05-25 | 4.12 | 4.2.1 | NA |
| OpenSUSE Leap 15.1 | 2019-05-22 | 4.12 | 4.2.1 | NA |

## OpenSUSE Leap 15.0/15.1下编译环境搭建

安装kernel开发包：

1. 输入“sudo -s”切换到root模式：
2. 输入“zypper install kernel-default-devel”命令，安装kernel开发包，如下图示：

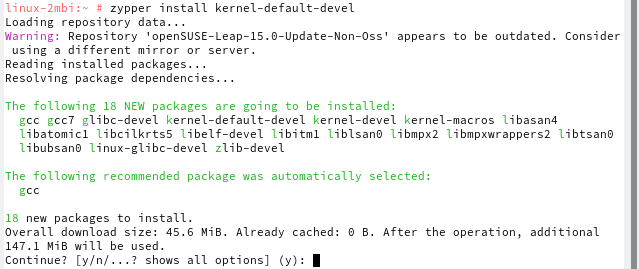


图8. 安装kernel开发包

出现“Continue?[y/n/…]”后输入“y”继续安装kernel module。

OpenSUSE Leap 15.0下安装完kernel开发包后，请确认/usr/src下安装的kernel开发包版本与系统内核版本是否一致。

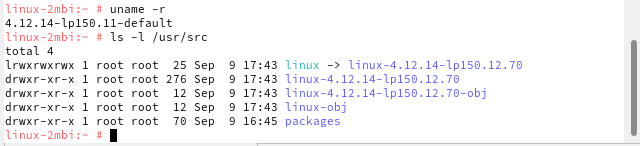


图9. 确认版本一致

如果不一致请使用“zypper update”命令同步进行系统更新，避免出现安装的kenel开发包与内核默认版本不匹配而出现编译失败问题。

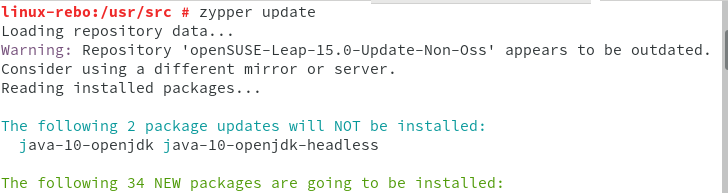


图10. 同步更新系统

|  |  |  |  |
| --- | --- | --- | --- |
|  | C:\Users\Administrator\Desktop\图层 4.png |  | 上述工具安装完成后，请重启系统。 |

# 集成Fibocom模块QMI\_WWAN驱动信息

## QMI\_WWAN驱动代码结构

如下所示，驱动以源代码的形式提供，由用户在自己的系统编译。

qmi\_wwan\_f/

├── qmi\_wwan\_f.c

├── Makefile

├── README

## QMI\_WWAN驱动编译

1. 进入qmi\_wwan\_f驱动目录，输入“make install”命令编译/安装qmi\_wwan\_f驱动，如下图示。

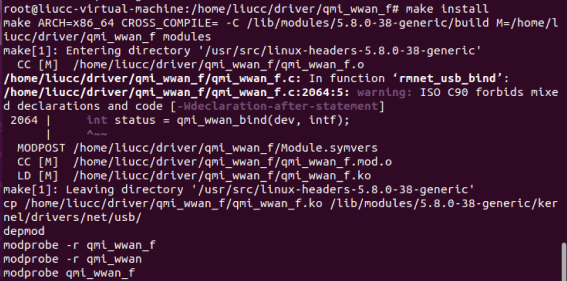


图11. qmi\_wwan\_f驱动加载

1. 输入“lsmod | grep qmi\_wwan\_f”，查看qmi\_wwan\_f驱动加载，驱动加载成功如下图示：

C:\Users\Administrator\AppData\Roaming\DingTalk\9077886_v2\ImageFiles\1612429022019_0526E748-8166-44b4-A6B1-E3A7E07D8B43.png

图12. 检测qmi\_wwan\_f驱动加载

## 检测Fibocom模块设备加载

驱动加载成功后，插入模块，使用“ls –l /dev/ttyUSB\*”命令查看模块端口信息，若设备正常挂载，将会有如下内容返回：

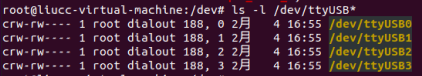


图13. 查看模块端口信息

表5. USB默认端口功能说明

|  |  |  |
| --- | --- | --- |
| 设备名 | 作用 | 备注 |
| ttyUSB0 | DIAG | 获取modem log端口 |
| ttyUSB1 | MODEM | PPP拨号端口 |
| ttyUSB2 | AT | RIL程序发送AT命令请求和接收命令响应的端口。 |
| ttyUSB3 | NMEA | 获取GPS信息的端口 |

# Fibocom模块拨号

## QMI拨号流程

模块开机： 模块的供电稳定后，FG150/FM150/NL952系列模块上电后开机；

模块关机：模块断电。推荐断电和再次上电间隔12秒以上（视断电时电容放电时间而定）建议相邻两次开机间隔大于90秒，相邻两次断电大于300秒。

模块开机

使用lsusb查询模块是否初始化完成。

查询模块是否初始化完成

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。

模块枚举出ttyUSB\*、网口

进入/drivers/qmi\_wwan\_f目录；

执行make install命令；

加载qmi\_wwan\_f驱动；

usb-devices查看驱动加载成功。

加载qmi\_wwan\_f驱动

成功获取IPV4地址说明单路拨号成功；

假如最少连续拨号5次仍然获取不到IPV4地址，复位模块。

如果需要PIN码则先用AT+CPIN按AT手册设置PIN码。

执行./Fibocom-dial拨号

ifconfig –a查看wwan0是否获取到ip

验证wwan0是否获取到IP；

假如连续查5次，wwan0都未获得IP，则复位模块；

假如连续查90秒返回值不正确，则复位模块。

查询结果一直连续加载数据包收发则表明成功ping通，如果连续ping超过5次仍然ping不通，复位模块。

ping www.baidu.com查看其是否能ping通网

实网测试是否能正常应用，如果加载不出网页，复位模块。

实网测试

## QMI单路拨号

1. QMI\_WWAN驱动加载完毕，即可进入/fibocom-dial目录（根据个人目录指定），首先执行make命令进行编译，生成fibocom-dial二进制文件，其次执行./fibocom-dial进行数据拨号，成功获取到IPV4地址，如下所示。

root@ght-Lenovo-V130-14IKB:~/lcc/tool/Fibocom-dial\_Linux\_Tool\_V2.0.8/src# ./fibocom-dial

[07-19\_14:28:57:848] Fibocom-dial\_Linux\_Tool\_V2.0.8

[07-19\_14:28:57:848] ./fibocom-dial profile[1] = (null)/(null)/(null)/0, pincode = (null)

[07-19\_14:28:57:849] socket[3] successfuly!

[07-19\_14:28:57:849] Waiting client to connect.....

[07-19\_14:28:57:851] Find /sys/bus/usb/devices/1-1 idVendor=2cb7 idProduct=0104

[07-19\_14:28:57:851] Find /sys/bus/usb/devices/1-1:1.4/net/wwan0

[07-19\_14:28:57:852] Find usbnet\_adapter = wwan0

[07-19\_14:28:57:852] Find /sys/bus/usb/devices/1-1:1.4/usbmisc/cdc-wdm0

[07-19\_14:28:57:852] Find qmichannel = /dev/cdc-wdm0

[07-19\_14:28:57:852] qmichannel(/dev/cdc-wdm0) usbnet\_adapter(wwan0)

[07-19\_14:28:57:852] pcie mode

[07-19\_14:28:57:852] ioctl(0x89f3, qmap\_settings) failed: Operation not supported, rc=-1

[07-19\_14:28:57:852] access /sys/class/net/wwan0/qmap\_mode

[07-19\_14:28:57:853] qmap\_mode = 1, muxid = 0x81, qmap\_netcard = wwan0

[07-19\_14:28:57:853] ioctl(0x89f3, qmap\_settings) failed: Operation not supported, rc=-1

[07-19\_14:28:57:853] access /sys/class/net/wwan0/qmap\_mode

[07-19\_14:28:57:853] qmap\_mode = 1, muxid = 0x81, qmap\_netcard = wwan0

[07-19\_14:28:57:853] qmap\_mode=1

[07-19\_14:28:57:865] cdc\_wdm\_fd = 9

[07-19\_14:28:57:865] write triger\_event: 4098 to qmidevice\_control\_fd

[07-19\_14:28:57:955] QmiWwanGetClientID: QMIType = 1 clientid 14

[07-19\_14:28:57:955] Get clientWDS = 14

[07-19\_14:28:57:987] QmiWwanGetClientID: QMIType = 2 clientid 4

[07-19\_14:28:57:987] Get clientDMS = 4

[07-19\_14:28:58:019] QmiWwanGetClientID: QMIType = 3 clientid 2

[07-19\_14:28:58:019] Get clientNAS = 2

[07-19\_14:28:58:051] QmiWwanGetClientID: QMIType = 11 clientid 2

[07-19\_14:28:58:051] Get clientUIM = 2

[07-19\_14:28:58:083] QmiWwanGetClientID: QMIType = 26 clientid 1

[07-19\_14:28:58:083] Get clientWDA = 1

[07-19\_14:28:58:115] requestBaseBandVersion 89602.1000.00.04.08.21

[07-19\_14:28:58:147] qmap\_settings.rx\_urb\_size = 4096

[07-19\_14:28:58:211] sim\_select = 0

[07-19\_14:28:58:243] curr\_ints\_sim1

[07-19\_14:28:58:243] curr\_ints->CardState is 1

[07-19\_14:28:58:243] curr\_ints->NumApp is 1

[07-19\_14:28:58:243] AppType = 2

[07-19\_14:28:58:243] requestGetSIMStatus SIMStatus: SIM\_READY

[07-19\_14:28:58:275] requestGetICCID DeviceICCID: 898600F0261831632928

[07-19\_14:28:58:307] requestGetIMSI DeviceIMSI: 460026092319783

[07-19\_14:28:58:339] requestGetProfile[1] ///0

[07-19\_14:28:58:371] requestRegistrationState2 MCC: 460, MNC: 0, PS: Detached, DataCap: UNKNOW

[07-19\_14:28:58:371] write signo: 12 to signal\_control\_fd

[07-19\_14:28:58:371] epoll fd = 6, events = 0x0001

[07-19\_14:28:58:371] get signo: 12

[07-19\_14:28:58:403] requestQueryDataCall IPv4ConnectionStatus: DISCONNECTED

[07-19\_14:28:58:403] usbnet\_link\_change :link:0

[07-19\_14:28:58:403] enter udhcpc\_stop

[07-19\_14:28:58:403] enter fibo\_set\_driver\_link\_state

[07-19\_14:28:58:404] if\_link\_down wwan0

[07-19\_14:28:58:404] write signo: 10 to signal\_control\_fd

[07-19\_14:28:58:404] epoll fd = 7, events = 0x0000

[07-19\_14:28:58:404] epoll fd = 6, events = 0x0001

[07-19\_14:28:58:404] get signo: 10

[07-19\_14:28:58:404] usbnet\_link\_change :link:0

[07-19\_14:28:59:011] requestRegistrationState2 MCC: 460, MNC: 0, PS: Attached, DataCap: LTE

[07-19\_14:28:59:011] epoll fd = 6, events = 0x0000

[07-19\_14:28:59:011] epoll fd = 7, events = 0x0001

[07-19\_14:28:59:011] write signo: 12 to signal\_control\_fd

[07-19\_14:28:59:011] epoll fd = 6, events = 0x0001

[07-19\_14:28:59:011] get signo: 12

[07-19\_14:28:59:075] usbnet\_link\_change :link:1

[07-19\_14:28:59:107] enter fibo\_set\_driver\_link\_state

[07-19\_14:28:59:107] if\_link\_up wwan0

[07-19\_14:28:59:108] IPv4 MTU: 1500

[07-19\_14:28:59:108] IPv4 Address: 10.9.199.226

[07-19\_14:28:59:108] IPv4 Netmask: 30

[07-19\_14:28:59:108] IPv4 Gateway: 10.9.199.225

[07-19\_14:28:59:108] IPv4 DNS1: 211.137.130.2

[07-19\_14:28:59:108] IPv4 DNS2: 211.137.130.4

1. 拨号成功后，输入ifconfig –a命令，查询wwan0是否获取到IP，如下所示，IP地址与上面拨号成功后所获取到的IP地址相同：

wwan0: flags=193<UP,RUNNING,NOARP> mtu 1500

inet 10.9.199.226 netmask 255.255.255.252

inet6 fe80::844:72ff:fe90:32b4 prefixlen 64 scopeid 0x20<link>

ether 0a:44:72:90:32:b4 txqueuelen 1000 (以太网)

RX packets 2 bytes 217 (217.0 B)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 3 bytes 196 (196.0 B)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

1. ping包测试：

root@ght-Lenovo-V130-14IKB:~/lcc/qmi\_wwan\_f/Fibocom\_QMI\_WWAN\_Driver\_V1.0.3# ping 8.8.8.8

PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.

64 bytes from 8.8.8.8: icmp\_seq=1 ttl=52 time=105 ms

64 bytes from 8.8.8.8: icmp\_seq=2 ttl=52 time=75.9 ms

^C

--- 8.8.8.8 ping statistics ---

2 packets transmitted, 2 received, 0% packet loss, time 1002ms

## QMI多路拨号

1. 多路拨号需要在驱动加载前配置完成多PDN，打开源码qmi\_wwan\_f.c文件，将117行的qmap\_mode设置为多路，如下设置为4，表示当前可支持4路拨号（当前最大可支持4路拨号）。

static uint \_\_read\_mostly **qmap\_mode = 4;**

1. 进入qmi\_wwan\_f驱动源码所在的路径下，执行“make install”，安装qmi\_wwan\_f驱动，同理3.2.1安装驱动的操作。
2. 驱动安装完毕，通过“ifconfig –a”命令即可查看当前根据物理网卡wwan0映射出wwan0.1、wwan0.2、wwan0.3、wwan0.4四个虚拟网卡。
3. 进入拨号工具所在路径，执行make命令进行编译，生成fibo\_qmimsg\_server和multi-pdn-manager二进制文件。首先执行“./fibo\_qmimsg\_server &”，启动启动qmi消息服务进程：

root@ght-Lenovo-V130-14IKB:~/lcc/tool/Fibocom-dial\_Linux\_Tool\_V2.0.8/src# **./fibo\_qmimsg\_server &**

Find /sys/bus/usb/devices/1-1 idVendor=2cb7 idProduct=0104

Find /sys/bus/usb/devices/1-1:1.4/usbmisc/cdc-wdm0

Will use cdc-wdm /dev/cdc-wdm0

qmi\_proxy\_init enter

qmi\_proxy\_loop enter thread\_id 140040038012672

send\_qmi\_timeout ret=110

link\_prot 2

ul\_data\_aggregation\_protocol 5

dl\_data\_aggregation\_protocol 5

dl\_data\_aggregation\_max\_datagrams 63

dl\_data\_aggregation\_max\_size 16384

ul\_data\_aggregation\_max\_datagrams 1

ul\_data\_aggregation\_max\_size 4096

qmi\_proxy\_init finished, rx\_urb\_size is 16384

local server: fibo\_qmimsg\_server sockfd = 4

qmi\_start\_server: qmi\_proxy\_server\_fd = 4

1. 执行./multi-pdn-manager进行数据拨号，参考《Fibocom QMI拨号工具使用指南\_Linux》 2.4章节，首先用第1路虚拟网卡wwan0.1和第1路profile1进行拨号，如下所示：

root@ght-Lenovo-V130-14IKB:~/lcc/tool/Fibocom-dial\_Linux\_Tool\_V2.0.8/src# **./multi-pdn-manager**

[07-19\_11:29:01:466] Start Fibocom multi-pdn-manager!

dev: /dev/ttyUSB1

rate:115200

sendbuffer:at+gtpcie=3

at+gtpcie=3

+GTPCIE: RC

OK

[07-19\_11:29:01:509] access /sys/class/net/wwan0/qmap\_mode

[07-19\_11:29:01:509] access /sys/class/net/wwan0/qmap\_mode

[07-19\_11:29:01:511] Fibocom multi-pdn-manager 1.0.4

[07-19\_11:29:01:511] Fibocom manager current qmap\_num is 4

[07-19\_11:29:01:511] Fibocom manager sub-instance1 Disconnected

[07-19\_11:29:01:511] Fibocom manager sub-instance2 Disconnected

[07-19\_11:29:01:511] Fibocom manager sub-instance3 Disconnected

[07-19\_11:29:01:511] Fibocom manager sub-instance4 Disconnected

Please select an action[0-9]

1.show profile setting

2.set profile setting

3.connect to network

4.disconnect to network

5.show connect log

6.reload profile setting

7.save profile setting

8.show max instance number

9.set max instance number

0.exit

**input [0-9]: 3**

**connect visual net interface use:1**

**connect profile use:1**

[07-19\_11:29:04:555] Start connect network use instance 1 profile 1

fibocom-dial -N 4 -n 1 -m 1 -s 1234567890 qwertyuio asdfghjkl 2 -4 -f instance1.txt

Press any key to continus[07-19\_11:29:04:561] exec pid 11717

[07-19\_11:29:04:562] Fibocom-dial\_Linux\_Tool\_V2.0.8

[07-19\_11:29:04:562] fibocom-dial profile[1] = 1234567890/qwertyuio/asdfghjkl/2, pincode = (null)

[07-19\_11:29:04:562] socket[5] successfuly!

[07-19\_11:29:04:562] Waiting client to connect.....

[07-19\_11:29:04:563] Find /sys/bus/usb/devices/1-1 idVendor=2cb7 idProduct=0104

[07-19\_11:29:04:563] Find /sys/bus/usb/devices/1-1:1.4/net/wwan0

[07-19\_11:29:04:563] Find usbnet\_adapter = wwan0

[07-19\_11:29:04:563] Find /sys/bus/usb/devices/1-1:1.4/usbmisc/cdc-wdm0

[07-19\_11:29:04:563] Find qmichannel = /dev/cdc-wdm0

[07-19\_11:29:04:563] qmichannel(/dev/cdc-wdm0) usbnet\_adapter(wwan0)

[07-19\_11:29:04:563] pcie mode

[07-19\_11:29:04:563] ioctl(0x89f3, qmap\_settings) failed: Operation not supported, rc=-1

[07-19\_11:29:04:563] access /sys/class/net/wwan0/qmap\_mode

[07-19\_11:29:04:563] qmap\_mode = 4, muxid = 0x81, qmap\_netcard = **wwan0.1**

[07-19\_11:29:04:563] ioctl(0x89f3, qmap\_settings) failed: Operation not supported, rc=-1

[07-19\_11:29:04:563] access /sys/class/net/wwan0/qmap\_mode

[07-19\_11:29:04:563] qmap\_mode = 4, muxid = 0x81, qmap\_netcard = wwan0.1

[07-19\_11:29:04:563] connect to fibo\_qmimsg\_server sockfd = 11

[07-19\_11:29:04:563] cdc\_wdm\_fd = 11

[07-19\_11:29:04:563] write triger\_event: 4098 to qmidevice\_control\_fd

[07-19\_11:29:04:621] QmiWwanGetClientID: QMIType = 1 clientid 14

[07-19\_11:29:04:621] Get clientWDS = 14

[07-19\_11:29:04:652] QmiWwanGetClientID: QMIType = 2 clientid 1

[07-19\_11:29:04:652] Get clientDMS = 1

[07-19\_11:29:04:684] QmiWwanGetClientID: QMIType = 3 clientid 2

[07-19\_11:29:04:684] Get clientNAS = 2

[07-19\_11:29:04:716] QmiWwanGetClientID: QMIType = 11 clientid 2

[07-19\_11:29:04:716] Get clientUIM = 2

[07-19\_11:29:04:749] requestBaseBandVersion 89602.1000.00.04.08.21

[07-19\_11:29:04:812] sim\_select = 0

[07-19\_11:29:04:844] curr\_ints\_sim1

[07-19\_11:29:04:844] curr\_ints->CardState is 1

[07-19\_11:29:04:845] curr\_ints->NumApp is 1

[07-19\_11:29:04:845] AppType = 2

[07-19\_11:29:04:845] requestGetSIMStatus SIMStatus: SIM\_READY

[07-19\_11:29:04:875] requestGetICCID DeviceICCID: 898600F0261831632928

[07-19\_11:29:04:908] requestGetIMSI DeviceIMSI: 460026092319783

[07-19\_11:29:04:908] requestSetProfile[1] 1234567890/qwertyuio/asdfghjkl/2

[07-19\_11:29:04:972] requestGetProfile[1] 1234567890/qwertyuio/asdfghjkl/2

[07-19\_11:29:05:004] requestRegistrationState2 MCC: 460, MNC: 0, PS: Attached, DataCap: LTE

[07-19\_11:29:05:004] write signo: 12 to signal\_control\_fd

[07-19\_11:29:05:004] epoll fd = 8, events = 0x0001

[07-19\_11:29:05:004] get signo: 12

[07-19\_11:29:05:036] requestQueryDataCall IPv4ConnectionStatus: DISCONNECTED

[07-19\_11:29:05:036] usbnet\_link\_change :link:0

[07-19\_11:29:05:036] enter udhcpc\_stop

[07-19\_11:29:05:036] enter fibo\_set\_driver\_link\_state

[07-19\_11:29:05:037] if\_link\_down wwan0.1

[07-19\_11:29:05:037] write signo: 10 to signal\_control\_fd

[07-19\_11:29:05:037] epoll fd = 9, events = 0x0000

[07-19\_11:29:05:037] epoll fd = 8, events = 0x0001

[07-19\_11:29:05:037] get signo: 10

[07-19\_11:29:05:037] usbnet\_link\_change :link:0

[07-19\_11:29:05:069] requestRegistrationState2 MCC: 460, MNC: 0, PS: Attached, DataCap: LTE

[07-19\_11:29:05:100] requestSetupDataCall WdsConnectionIPv4Handle: 0x360397e0

[07-19\_11:29:05:132] write triger\_event: 4101 to qmidevice\_control\_fd

[07-19\_11:29:05:164] epoll fd = 9, events = 0x0000

[07-19\_11:29:05:165] epoll fd = 8, events = 0x0000

[07-19\_11:29:05:165] epoll fd = 9, events = 0x0001

[07-19\_11:29:05:165] write signo: 12 to signal\_control\_fd

[07-19\_11:29:05:165] epoll fd = 8, events = 0x0001

[07-19\_11:29:05:165] get signo: 12

[07-19\_11:29:05:229] usbnet\_link\_change :link:1

[07-19\_11:29:05:261] enter fibo\_set\_driver\_link\_state

[07-19\_11:29:05:261] ifconfig wwan0 up

[07-19\_11:29:05:269] if\_link\_up wwan0.1

**[07-19\_11:29:05:270] IPv4 MTU: 1500**

**[07-19\_11:29:05:270] IPv4 Address: 10.84.69.148**

**[07-19\_11:29:05:270] IPv4 Netmask: 29**

**[07-19\_11:29:05:270] IPv4 Gateway: 10.84.69.149**

**[07-19\_11:29:05:271] IPv4 DNS1: 211.137.130.2**

**[07-19\_11:29:05:271] IPv4 DNS2: 211.137.130.4**

**[07-19\_11:29:05:271] if\_link\_up wwan0.1**

1. 从上述拨号log来看，使用第1路虚拟网卡wwan0.1拨号成功，代码段加粗字体可看到成功获取IP，我们再使用第2路虚拟网卡wwan0.2进行拨号，此处采用第2路profile2，拨号过程如下所示：

**input [0-9]: 3**

**connect visual net interface use:2**

**connect profile use:2**

[07-19\_11:49:09:141] Start connect network use instance 2 profile 2

fibocom-dial -N 4 -n 2 -m 2 -s ctnet -4 -f instance2.txt

Press any key to continus[07-19\_11:49:09:147] exec pid 12115

[07-19\_11:49:09:149] Fibocom-dial\_Linux\_Tool\_V2.0.8

[07-19\_11:49:09:149] fibocom-dial profile[2] = ctnet///0, pincode = (null)

[07-19\_11:49:09:150] socket[5] successfuly!

[07-19\_11:49:09:150] Waiting client to connect.....

[07-19\_11:49:09:152] Find /sys/bus/usb/devices/1-1 idVendor=2cb7 idProduct=0104

[07-19\_11:49:09:152] Find /sys/bus/usb/devices/1-1:1.4/net/wwan0

[07-19\_11:49:09:152] Find usbnet\_adapter = wwan0

[07-19\_11:49:09:152] Find /sys/bus/usb/devices/1-1:1.4/usbmisc/cdc-wdm0

[07-19\_11:49:09:152] Find qmichannel = /dev/cdc-wdm0

[07-19\_11:49:09:152] qmichannel(/dev/cdc-wdm0) usbnet\_adapter(wwan0)

[07-19\_11:49:09:152] pcie mode

[07-19\_11:49:09:153] ioctl(0x89f3, qmap\_settings) failed: Operation not supported, rc=-1

[07-19\_11:49:09:153] access /sys/class/net/wwan0/qmap\_mode

[07-19\_11:49:09:153] qmap\_mode = 4, muxid = 0x82, qmap\_netcard = **wwan0.2**

[07-19\_11:49:09:153] ioctl(0x89f3, qmap\_settings) failed: Operation not supported, rc=-1

[07-19\_11:49:09:153] access /sys/class/net/wwan0/qmap\_mode

[07-19\_11:49:09:153] qmap\_mode = 4, muxid = 0x82, qmap\_netcard = wwan0.2

[07-19\_11:49:09:154] connect to fibo\_qmimsg\_server sockfd = 11

[07-19\_11:49:09:154] cdc\_wdm\_fd = 11

[07-19\_11:49:09:154] write triger\_event: 4098 to qmidevice\_control\_fd

[07-19\_11:49:09:205] QmiWwanGetClientID: QMIType = 1 clientid 15

[07-19\_11:49:09:205] Get clientWDS = 15

[07-19\_11:49:09:237] QmiWwanGetClientID: QMIType = 2 clientid 2

[07-19\_11:49:09:237] Get clientDMS = 2

[07-19\_11:49:09:268] QmiWwanGetClientID: QMIType = 3 clientid 3

[07-19\_11:49:09:268] Get clientNAS = 3

[07-19\_11:49:09:301] QmiWwanGetClientID: QMIType = 11 clientid 3

[07-19\_11:49:09:301] Get clientUIM = 3

[07-19\_11:49:09:332] requestBaseBandVersion 89602.1000.00.04.08.21

[07-19\_11:49:09:397] sim\_select = 0

[07-19\_11:49:09:429] curr\_ints\_sim1

[07-19\_11:49:09:429] curr\_ints->CardState is 1

[07-19\_11:49:09:429] curr\_ints->NumApp is 1

[07-19\_11:49:09:429] AppType = 2

[07-19\_11:49:09:429] requestGetSIMStatus SIMStatus: SIM\_READY

[07-19\_11:49:09:461] requestGetICCID DeviceICCID: 898600F0261831632928

[07-19\_11:49:09:493] requestGetIMSI DeviceIMSI: 460026092319783

[07-19\_11:49:09:493] requestSetProfile[2] ctnet///0

[07-19\_11:49:09:557] requestGetProfile[2] ctnet///0

[07-19\_11:49:09:589] requestRegistrationState2 MCC: 460, MNC: 0, PS: Attached, DataCap: LTE

[07-19\_11:49:09:589] write signo: 12 to signal\_control\_fd

[07-19\_11:49:09:589] epoll fd = 8, events = 0x0001

[07-19\_11:49:09:589] get signo: 12

[07-19\_11:49:09:621] requestQueryDataCall IPv4ConnectionStatus: DISCONNECTED

[07-19\_11:49:09:621] usbnet\_link\_change :link:0

[07-19\_11:49:09:621] enter udhcpc\_stop

[07-19\_11:49:09:621] enter fibo\_set\_driver\_link\_state

[07-19\_11:49:09:622] if\_link\_down wwan0.2

[07-19\_11:49:09:622] write signo: 10 to signal\_control\_fd

[07-19\_11:49:09:622] epoll fd = 9, events = 0x0000

[07-19\_11:49:09:622] epoll fd = 8, events = 0x0001

[07-19\_11:49:09:622] get signo: 10

[07-19\_11:49:09:622] usbnet\_link\_change :link:0

[07-19\_11:49:09:653] requestRegistrationState2 MCC: 460, MNC: 0, PS: Attached, DataCap: LTE

[07-19\_11:49:18:839] write signo: 12 to signal\_control\_fd

[07-19\_11:49:18:840] epoll fd = 8, events = 0x0001

[07-19\_11:49:18:840] get signo: 12

[07-19\_11:49:18:874] usbnet\_link\_change :link:1

[07-19\_11:49:18:874] epoll fd = 9, events = 0x0000

[07-19\_11:49:19:131] requestSetupDataCall WdsConnectionIPv4Handle: 0x361b1650

[07-19\_11:49:19:163] write triger\_event: 4101 to qmidevice\_control\_fd

[07-19\_11:49:19:195] epoll fd = 9, events = 0x0000

[07-19\_11:49:19:195] epoll fd = 8, events = 0x0000

[07-19\_11:49:19:195] epoll fd = 9, events = 0x0001

[07-19\_11:49:19:195] write signo: 12 to signal\_control\_fd

[07-19\_11:49:19:195] epoll fd = 8, events = 0x0001

[07-19\_11:49:19:195] get signo: 12

[07-19\_11:49:19:259] usbnet\_link\_change :link:1

[07-19\_11:49:19:291] enter fibo\_set\_driver\_link\_state

[07-19\_11:49:19:291] ifconfig wwan0 up

[07-19\_11:49:19:298] if\_link\_up wwan0.2

**[07-19\_11:49:19:298] IPv4 MTU: 1500**

**[07-19\_11:49:19:299] IPv4 Address: 10.82.57.76**

**[07-19\_11:49:19:299] IPv4 Netmask: 29**

**[07-19\_11:49:19:299] IPv4 Gateway: 10.82.57.77**

**[07-19\_11:49:19:299] IPv4 DNS1: 211.137.130.2**

**[07-19\_11:49:19:299] IPv4 DNS2: 211.137.130.4**

**[07-19\_11:49:19:299] if\_link\_up wwan0.2**

**[07-19\_11:49:21:301] epoll fd = 9, events = 0x0000**

1. 从上述拨号log来看，使用第二路虚拟网卡wwan0.2拨号成功，代码段加粗字体可看到成功获取IP地址。拨号成功后，另开一个窗口，通过“ifconfig –a”查看当前网卡状态，可以看到wwan0.1和wwan0.2均获取到对应的IP地址，如下所示：

wwan0.1: flags=193<UP,RUNNING,NOARP> mtu 1500

**inet 10.84.69.148** netmask 255.255.255.248

ether 0a:44:72:90:32:b4 txqueuelen 1000 (以太网)

RX packets 7 bytes 917 (917.0 B)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 7 bytes 488 (488.0 B)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wwan0.2: flags=193<UP,RUNNING,NOARP> mtu 1500

**inet 10.82.57.76** netmask 255.255.255.248

ether 0a:44:72:90:32:b4 txqueuelen 1000 (以太网)

RX packets 18 bytes 2474 (2.4 KB)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 18 bytes 1272 (1.2 KB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

1. 网卡获取到IP地址后，进行ping测试，如下所示，均可ping通：

root@ght-Lenovo-V130-14IKB:~/lcc/qmi\_wwan\_f/Fibocom\_QMI\_WWAN\_Driver\_V1.0.3# **ping 8.8.8.8 -I wwan0.1**

PING 8.8.8.8 (8.8.8.8) from 10.84.69.148 wwan0.1: 56(84) bytes of data.

64 bytes from 8.8.8.8: icmp\_seq=1 ttl=111 time=140 ms

64 bytes from 8.8.8.8: icmp\_seq=2 ttl=111 time=78.5 ms

64 bytes from 8.8.8.8: icmp\_seq=3 ttl=111 time=86.1 ms

64 bytes from 8.8.8.8: icmp\_seq=4 ttl=111 time=93.4 ms

root@ght-Lenovo-V130-14IKB:~/lcc/qmi\_wwan\_f/Fibocom\_QMI\_WWAN\_Driver\_V1.0.3# **ping 8.8.8.8 -I wwan0.2**

PING 8.8.8.8 (8.8.8.8) from 10.82.57.76 wwan0.2: 56(84) bytes of data.

64 bytes from 8.8.8.8: icmp\_seq=2 ttl=52 time=232 ms

64 bytes from 8.8.8.8: icmp\_seq=3 ttl=52 time=85.5 ms

64 bytes from 8.8.8.8: icmp\_seq=4 ttl=52 time=106 ms