RM520N Ubuntu qmi datacall demo

Catalogue

Introduction 3
1. Hardware settings 4
2. Install drivers 5
2.1 Install Option USB Driver 5
2.2 Install QMI_WWAN network card driver 6
3. Install Quectel-CM tool 8
4. APN Setting & Register to Network 9
5. Dialing 10

Introduction

This article demonstrates the entire process of using the RM520N-GL module for QMI dialing in an Ubuntu environment.

Modules: RM520N-GL
Firmware: RM520NGLAAR01A08M4G_01.200.01.200
EVB: 5G-M2-EVB_V1.3
X86 PC: Ubuntu 16.04 LTS
Kernel ver.: 4.15.0-142-generic

1. Hardware settings

Install the module on the EVB, connect the antenna properly, plug in the sim card, and then connect to the PC via USB.

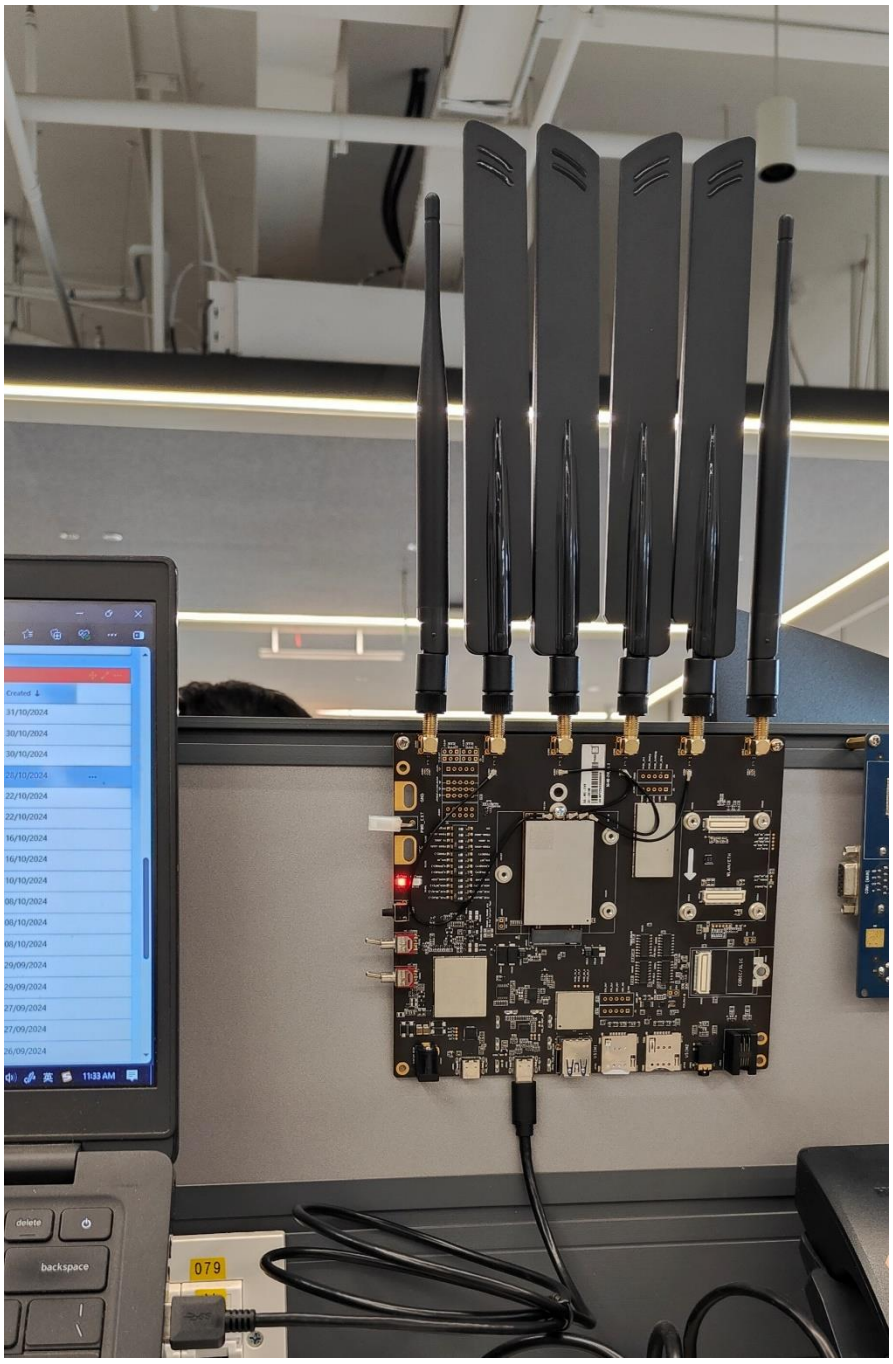


Figure1-1 hardware setting

2. Install drivers.

To perform QMI dialing on Linux, we need to install two drivers: the USB driver and the network card driver.

We install the USB driver "Quectel_Linux_USB_Serial_Option_Driver_20241012" and the network card driver "Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9".

Place the two driver packages on the PC.

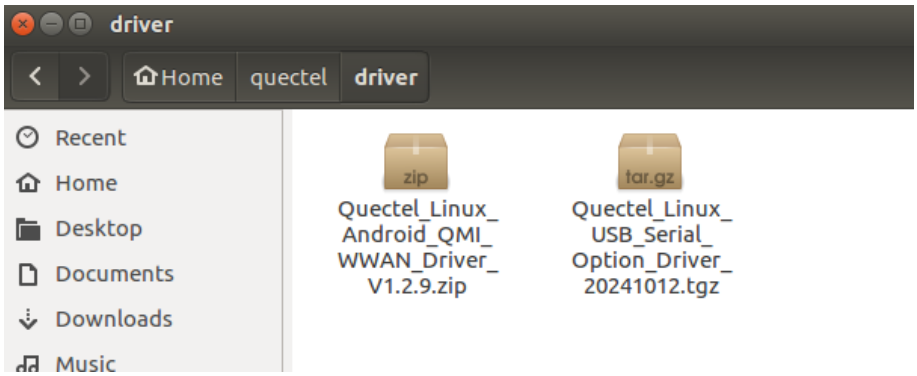


Figure2-1 drivers

2.1 Install Option USB Driver.

1. Extract the compressed packages.

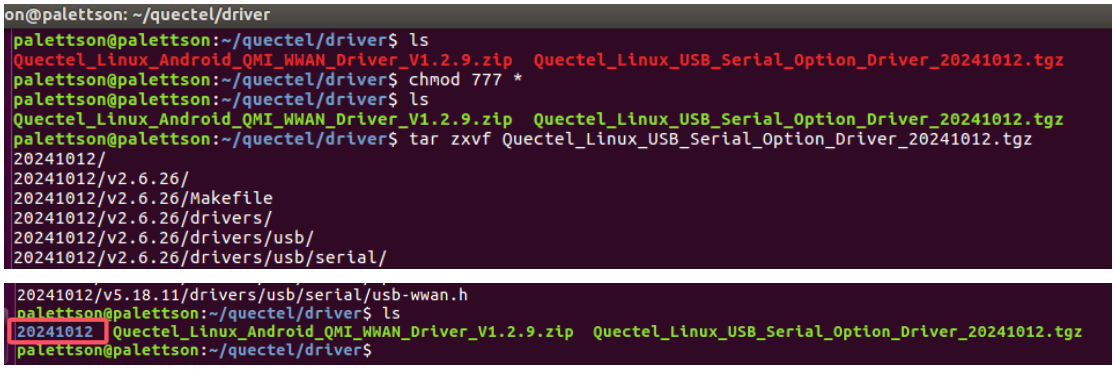


Figure2-2 Extract file

2. Check the kernel version, and find the most close one.

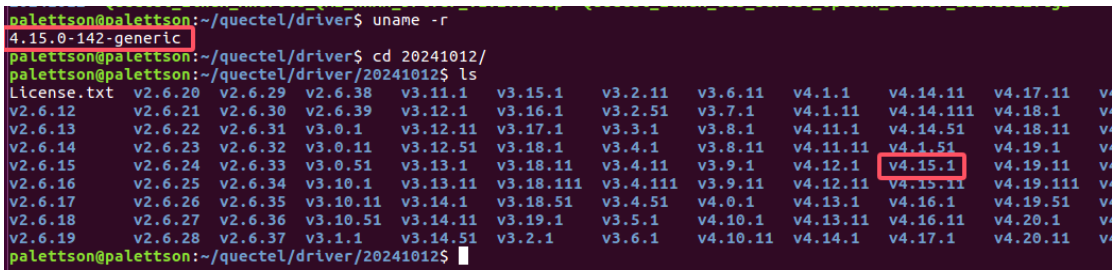


Figure2-3 kernel version

3. Get in the directory and compile the driver.

```
palettson@palettson:~/quectel/driver/20241012$ cd v4.15.1
palettson@palettson:~/quectel/driver/20241012/v4.15.1$ ls
drivers  Makefile
palettson@palettson:~/quectel/driver/20241012/v4.15.1$ sudo make install
[sudo] password for palettson:
make -C /lib/modules/4.15.0-142-generic/build M=/home/palettson/quectel/driver/20241012/v4.15.1 clean
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-142-generic'
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-142-generic'
make -C /lib/modules/4.15.0-142-generic/build M=/home/palettson/quectel/driver/20241012/v4.15.1 modules
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-142-generic'
  CC [M]  /home/palettson/quectel/driver/20241012/v4.15.1/./drivers/usb/serial/option.o
  CC [M]  /home/palettson/quectel/driver/20241012/v4.15.1/./drivers/usb/serial/usb_wwan.o
  CC [M]  /home/palettson/quectel/driver/20241012/v4.15.1/./drivers/usb/serial/qcserial.o
  Building modules, stage 2.
  MODPOST 3 modules
  CC      /home/palettson/quectel/driver/20241012/v4.15.1/./drivers/usb/serial/option.mod.o
  LD [M]  /home/palettson/quectel/driver/20241012/v4.15.1/./drivers/usb/serial/option.ko
  CC      /home/palettson/quectel/driver/20241012/v4.15.1/./drivers/usb/serial/qcserial.mod.o
  LD [M]  /home/palettson/quectel/driver/20241012/v4.15.1/./drivers/usb/serial/qcserial.ko
  CC      /home/palettson/quectel/driver/20241012/v4.15.1/./drivers/usb/serial/usb_wwan.mod.o
  LD [M]  /home/palettson/quectel/driver/20241012/v4.15.1/./drivers/usb/serial/usb_wwan.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-142-generic'
cp /home/palettson/quectel/driver/20241012/v4.15.1/drivers/usb/serial/*.ko /lib/modules/4.15.0-142-generic/kernel/drivers/usb/serial/
depmod
palettson@palettson:~/quectel/driver/20241012/v4.15.1$
```

Figure2-4 compile usb driver.

If the ttyUSB port can be identified, it indicates that the driver installation was successful. Refer to Figure 2-5.

```
palettson@palettson:~$ ls /dev/ttyUSB*
/dev/ttyUSB0  /dev/ttyUSB1  /dev/ttyUSB2  /dev/ttyUSB3
palettson@palettson:~$
```

Figure2-5 ttyUSB port

2.2 Install QMI_WWAN network card driver

1. Extract the compressed packages.

```
palettson@palettson:~/quectel/driver$ ls
20241012  Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9.zip  Quectel_Linux_USB_Serial_Option_Driver_20241012.tgz
palettson@palettson:~/quectel/driver$ sudo unzip
20241012/                                  Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9.zip
palettson@palettson:~/quectel/driver$ sudo unzip Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9.zip
[sudo] password for palettson:
Archive:  Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9.zip
   creating: Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9/
  inflating: Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9/License.txt
   creating: Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9/log/
  inflating: Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9/log/how_to_use_bridge.txt
  inflating: Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9/log/how_to_use_bridge_and_QMAP.txt
  inflating: Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9/Makefile
  inflating: Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9/qmi_wwan_q.c
  inflating: Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9/qmi_wwan_q.mod
  inflating: Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9/ReleaseNote.txt
  inflating: Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9/rmnet_nss.c
palettson@palettson:~/quectel/driver$ ls
20241012  Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9  Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9.zip  Quectel_Linux_USB_Serial_Option_Driver_20241012.tgz
palettson@palettson:~/quectel/driver$
```

Figure2-6 Extract file

2. Compile the qmi driver.

```
palettson@palettson:~/quectel/driver$ cd Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9/
palettson@palettson:~/quectel/driver/Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9$ ls
License.txt  log  Makefile  qmi_wwan_q.c  qmi_wwan_q.mod  ReleaseNote.txt  rmnet_nss.c
palettson@palettson:~/quectel/driver/Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9$ sudo make install
make ARCH=x86_64 CROSS_COMPILE= -C /lib/modules/4.15.0-142-generic/build M=/home/palettson/quectel/driver/Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9 modules
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-142-generic'
  CC [M]  /home/palettson/quectel/driver/Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9/qmi_wwan_q.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/palettson/quectel/driver/Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9/qmi_wwan_q.mod.o
  LD [M]  /home/palettson/quectel/driver/Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9/qmi_wwan_q.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-142-generic'
cp /home/palettson/quectel/driver/Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9/qmi_wwan_q.ko /lib/modules/4.15.0-142-generic/kernel/drivers/net/usb/
depmod
palettson@palettson:~/quectel/driver/Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9$ ls
License.txt  Makefile      Module.symvers  qmi_wwan_q.ko   qmi_wwan_q.mod.c  qmi_wwan_q.o         ReleaseNote.txt
log          modules.order  qmi_wwan_q.c    qmi_wwan_q.mod  qmi_wwan_q.mod.o  qmi_wwan_q.o.ur-safe  rmnet_nss.c
palettson@palettson:~/quectel/driver/Quectel_Linux_Android_QMI_WWAN_Driver_V1.2.9$
```

Figure2-7 compile qmi driver

3, Confirm whether the file has been installed

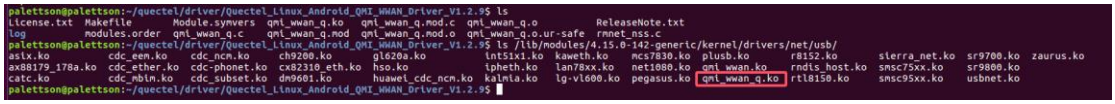


Figure2-8 Confirm driver

4, Confirm the identification of the qmi_wwan_q device. If it matches the image below, everything is working correctly.

(If you find that the default qmi_wwan driver is loaded on Ubuntu, you need to first unload the existing qmi_wwan driver and then load the qmi_wwan_q driver.)

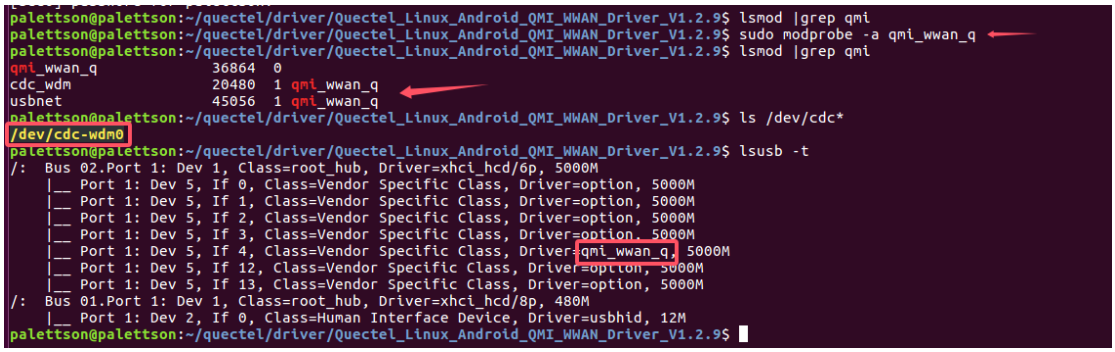


Figure2-9 Confirm qmi_wwan_q device

Note: To identify the QMI device, the parameters for the following two AT commands need to be set to their default values (for USB dial-up cases):

1) AT+QCFG=”usbnet”,0 //Set NIC data call method to RMNET

2) AT+QCFG="data_interface",0,0 //Set the network port communication via USB interface.

3. Install Quectel-CM tool

Quectel-CM is a tool recommended by Quectel for dialing, supporting a rich set of features and the advantage of not occupying the AT port during dialing.

1. Place the Quectel-CM packages on the PC.

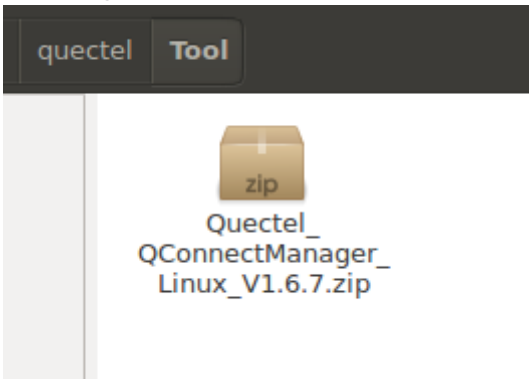


Figure3-1 Quectel-CM

2, Extract the compressed packages.

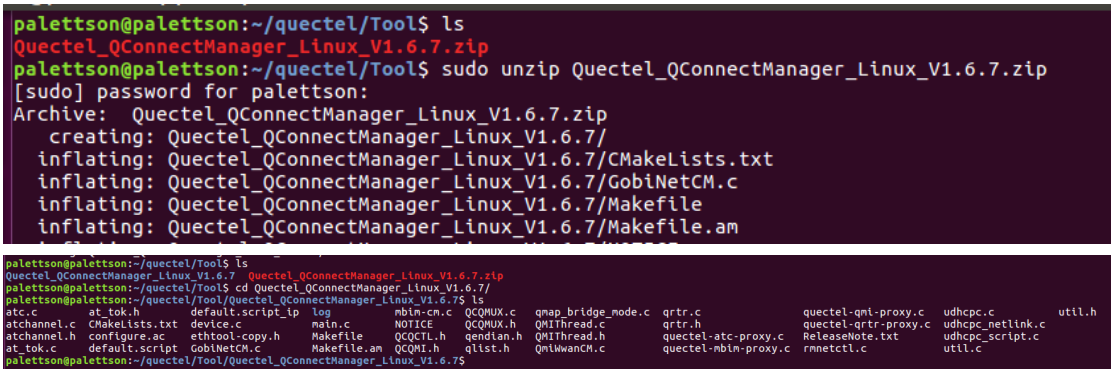


Figure3-2 Extract file

3, Compile the Quectel-CM tool.



Figure3-3 Quectel-CM tool

4, ModemManager and Quectel-CM can conflict. Ensure that ModemManager is uninstalled or disabled before dialing.

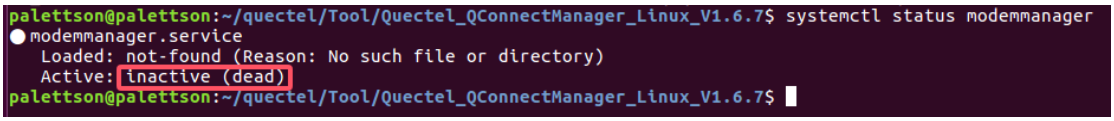


Figure3-4 ModemManager off

4. APN Setting & Register to Network

To dial, the module must first register to the network. To achieve this, we need to execute AT commands to configure the APN.

Note: Use any serial port tool to send AT commands to configure the APN, so that the module can register to the network.

1. Execute the AT+QICSGP command to configure the APN. Please refer to the steps in the Figure4-1.

After configuring correctly, the module should register to the network. As shown in Figure 4-1, the RM520N has registered to the 5G SA network.

```
[2024-11-01_18:41:35:404]AT+CFUN=0
[2024-11-01_18:41:36:055]OK
[2024-11-01_18:41:56:451]AT+QICSGP=1,1,"cmnet"
[2024-11-01_18:41:56:465]OK
[2024-11-01_18:42:03:989]AT+CFUN=1
[2024-11-01_18:42:04:250]OK
[2024-11-01_18:42:05:402]
[2024-11-01_18:42:05:402]+CPIN: READY

[2024-11-01_18:42:05:402]+QUSIM: 1
[2024-11-01_18:42:06:724]
[2024-11-01_18:42:06:724]+QIND: SMS DONE
[2024-11-01_18:42:11:434]
[2024-11-01_18:42:11:434]+QIND: PB DONE
[2024-11-01_18:42:16:188]AT+QENG="servingcell"
[2024-11-01_18:42:16:188]+QENG: "servingcell","NOCONN","NR5G-
SA","TDD",460,00,350452000,693,46550B,504990,41,12,-70,-11,25,1,-

[2024-11-01_18:42:16:188]OK
[2024-11-01_18:42:25:484]AT+C5GREG?
[2024-11-01_18:42:25:484]+C5GREG: 0,1

[2024-11-01_18:42:25:484]OK
```

Figure4-1 APN Setting& Register to network

5. Dialing

1. Run the Quectel-CM tool to perform dialing.

A successful dial will display as shown in the Figure5-1. In my demonstration, it connected to the 5G network, and you can see "5G_SA" displayed.

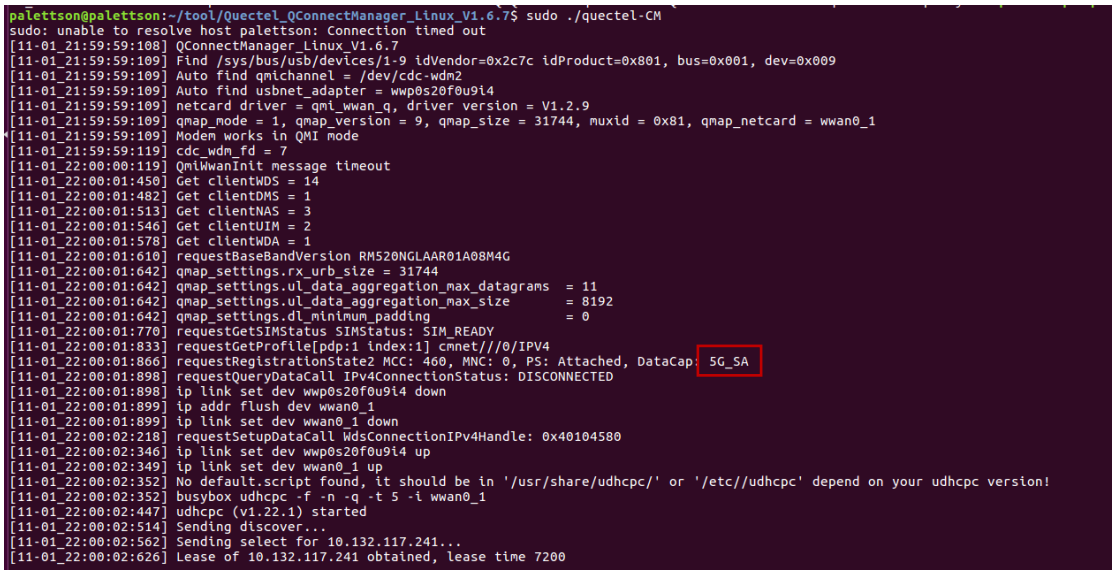


Figure5-1 QMI dialing

2. We can use the ifconfig to confirm the generated network interface.

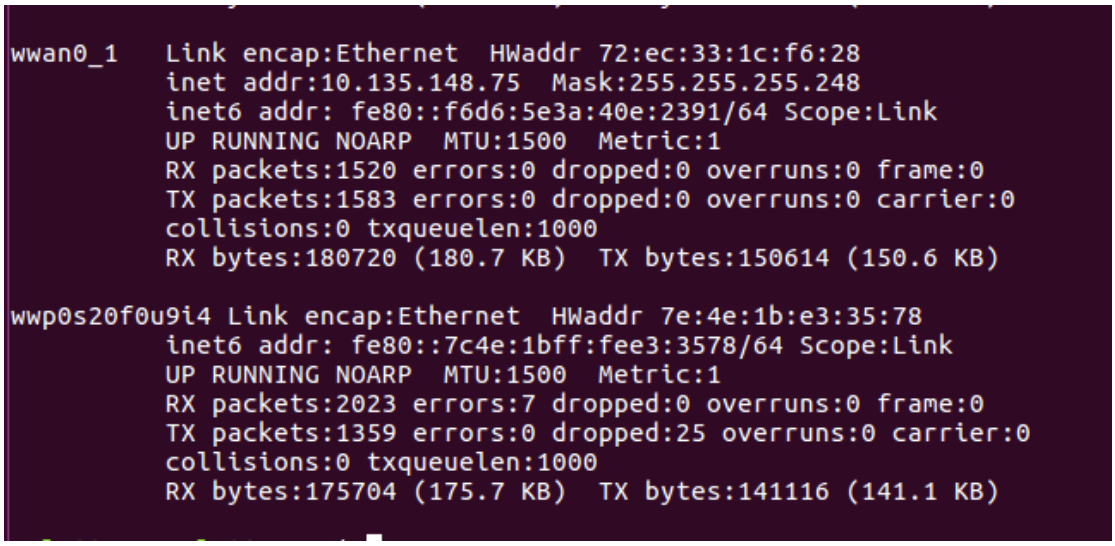


Figure5-2 network interface

3. And we should be able to successfully ping the network.
(If you are evaluating in the Ubuntu environment on a PC, you might encounter an issue where you cannot ping due to default route problems. In this case, you can first delete the default route and then perform the ping operation.)

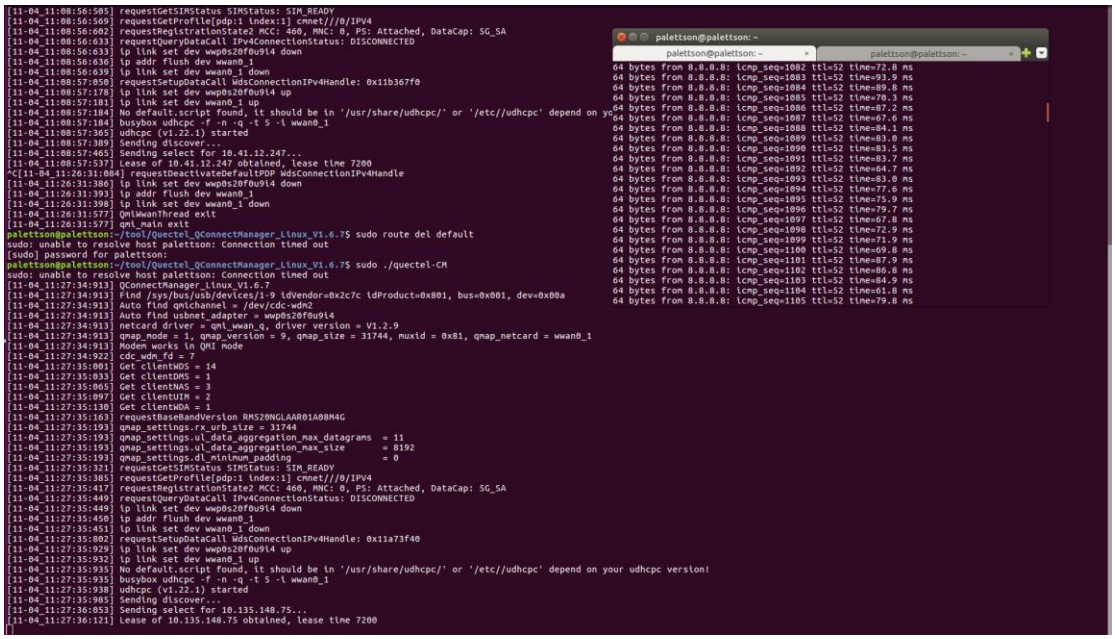


Figure5-3 ping 8.8.8.8