

N720 User Manual

Linux

Version 1.0



Copyright © Neoway Technology Co., Ltd 2017. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Shenzhen Neoway Technology Co., Ltd.

Neoway[®] 有方 is the trademark of Neoway Technology Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

This document provides guide for users to use the **N720**.

This document is intended for **system engineers (SEs), development engineers, and test engineers**.

The information in this document is subject to change without notice due to product version update or other reasons.

Every effort has been made in preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Neoway provides customers complete technical support. If you have any question, please contact your account manager or email to the following email addresses:

Sales@neoway.com

Support@neoway.com

Website: <http://www.neoway.com>

Revision Record

Issue	Changes	Revised By	Date
V1.0	Initial draft	Luojian	2017-07

Contents

1 Port Mapping.....	2
2 Loading USB Driver.....	3
2.1 Querying Devices	3
2.2 Adding VID and PID	3
2.3 Configuring Kernel.....	3
2.4 Testing AT Commands.....	4
3 Establishing Connection Using pppd	8
3.1 Checking Device	8
3.2 Adding VID and PID	8
3.3 Adding pppd Driver to Kernel.....	8
3.4 Dialing up	9
3.5 Test and Disconnection.....	12
3.5.1 Testing the Connection.....	12
3.5.2 Disconnecting the Connection.....	12
4 Establishing Connection Using RMNET.....	14
4.1 Checking Device	14
4.2 Adding VID and PID	14
4.3 Adding GobiNet Driver to Kernel	14
4.4 Configuring Before Dialup.....	16
4.5 Dialing Up Using RMNET.....	17
4.6 Test and Disconnection.....	18
4.6.1 Testing the Connection.....	18
4.6.2 Disconnecting the Connection.....	19
5 Establishing Connection Using RNDIS.....	20
5.1 Checking Device	20
5.2 Adding VID and PID	20
5.3 Adding RNDIS Driver to Kernel	20
5.4 Dialing Up Using RNDIS.....	21
5.5 Test and Disconnection.....	22
5.5.1 Testing the Connection.....	22
5.5.2 Disconnecting the Connection.....	23
6 Establishing Connection Using ECM.....	24
6.1 Checking Device	24
6.2 Adding VID and PID	24

6.3 Adding ECM Driver to Kernel 24

6.4 Dialing Up Using RNDIS..... 26

6.5 Test and Disconnection..... 27

 6.5.1 Testing the Connection..... 27

 6.5.2 Disconnecting the Connection..... 27

7 FAQ..... 28

About This Document

This document provides guidance about how to add the USB driver of the N720 and how to share the Internet access of an N720 on Linux.

Neoway
Confidential

1 Port Mapping

N720 supports network share via pppd, RNDIS, ECM, and RMNET. Each mode corresponds to different VID and PID. Developers can switch the mode by sending AT command (AT+NETSHAREMODE). The mapping between the ports and share modes is shown as follows:

- VID: 0x2949 PID: 0x8241 Support pppd/RNDIS dialup
Default USB composition: RNDIS(00)+MODEM(02)+TTY(03 NMEA)+TTY(04 AT)+Diag(05)
- VID: 0x2949 PID: 0x8242 Support pppd/ECM dialup
USB composition (MI No.): ECM(00)+MODEM(02)+TTY(03 NMEA)+TTY(04 AT)+Diag(05)
- VID: 0x2949 PID: 0x8243 Support pppd/RMNET dialup
USB composition (MI No.): RMNET(00)+MODEM(01)+TTY(02 NMEA)+TTY(03 AT)+Diag(04)

Table 1-1 N720 port mapping

Model	VID	PID	Share Mode	Port Mapping	Function
N720	0x2949	0x8241	pppd/RNDIS	RNDIS (00)	RNDIS NIC port
				MODEM (02)	Private data service port
				TTY (03 NMEA)	GPS/BD data output port
				TTY (04 AT)	AT Command port
				Diag (05)	Diagnosis port
	0x2949	0x8242	pppd/ECM	ECM (00)	ECM NIC port
				MODEM (02)	Private data service port
				TTY (03 NMEA)	GPS/BD data output port
				TTY (04 AT)	AT command port
				Diag (05)	Diagnosis port
	0x2949	0x8243	pppd/GobiNet	RMNET(00)	GobiNet NIC port
				MODEM(01)	Private data service port
				TTY(02 NMEA)	GPS/BD data output pot
				TTY(03 AT)	AT command port
				Diag(04)	Diagnosis port

2 Loading USB Driver

2.1 Querying Devices

Execute **lsusb** to check whether USB devices are displayed. Ensure that N720 is identified.

```
support@neoway:~$ lsusb
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 008 Device 002: ID 046d:c05a Logitech, Inc. M90/M100 Optical Mouse
Bus 008 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 006 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 006: ID 2949:8241
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

In the figure above, the VID of N720 is 0x2949 and the PID of N720 is 0x8241.



If the host does not identify N720, check whether N720 is connected to the host, whether it is powered up and started, or whether it is connected to the host through USB.

2.2 Adding VID and PID

In `kernel/drivers/usb/serial/option.c`, add **VID:0x2949 PID:0x8241** to `option_ids`.

```
static const struct usb_device_id option_ids[] = {
    { USB_DEVICE_INTERFACE_PROTOCOL(0x2949,0x8241,0)}, //neoway n720 vid:0x2949 pid:0x8241
    { USB_DEVICE_INTERFACE_PROTOCOL(0x2949,0x8243,0)}, //neoway n720 vid:0x2949 pid:0x8243
}
```

2.3 Configuring Kernel

On Linux, the USB-to-Serial driver is widely used. Before adding the driver, configure the Linux kernel following steps below:

1. Navigate to the kernel directory.

```
support@neoway:~/luo/kernel/linux-4.1.38$ ls
arch      dmesg.log  include    kernel      modules.builtin  REPORTING-BUGS  signing_key.x509  virt
block     Documentation  init       lib         modules.order    samples          sound             vmlinux
COPYING   drivers     ipc        MAINTAINERS  Module.symvers   scripts          System.map        vmlinux-gdb.py
CREDITS   firmware    Kbuild     Makefile     net              security         tools             vmlinux.o
crypto    fs          Kconfig    mm           README           signing_key.priv  usr              x509.genkey
support@neoway:~/luo/kernel/linux-4.1.38$
```

2. Execute **make menuconfig**.
The kernel configuration is displayed.

```
support@neoway:~/luo/kernel/linux-4.1.38$ make menuconfig
scripts/kconfig/mconf Kconfig
```

3. Configure kernel.

```
Device Drivers --->
  [*] USB support --->
    <*> USB Serial Converter support --->
    <*> USB driver for GSM and CDMA modems (CONFIG_USB_SERIAL_OPTION=v)
```

4. Click **Save** and **Exit**.
5. Compile kernel again and burn the compiled kernel files to the host.
6. Power up the host again and execute **ls /dev/ttyUSB***.

```
support@neoway:~$ ls /dev/ttyUSB*
/dev/ttyUSB0 /dev/ttyUSB1 /dev/ttyUSB2
support@neoway:~$ █
```

If N720 is loaded successfully, ttyUSB0 to ttyUSBx are listed under **/dev**. Two ttyUSB ports can be used to send AT commands. One is used as modem port and the other one is used as AT port.

For example, ttyUSB0 and ttyUSB2 in the above figure can be used to send AT commands. ttyUSB0 is the modem port and ttyUSB2 is the AT port.

If ttyUSB0 to ttyUSB4 are displayed and ttyUSB0 and ttyUSB2 can be used to send AT commands, ttyUSB0 is the modem port and ttyUSB2 is the AT port. If ttyUSB0 cannot be used to send AT commands, ttyUSB2 and ttyUSB4 can be used to send AT commands. ttyUSB2 is the modem port and ttyUSB4 is the AT port.

2.4 Testing AT Commands

1. Power up the host.
2. Execute **ls /dev/ttyUSB*** to check whether N720 is identified and whether ttyUSB are listed.

```
support@neoway:~$ lsusb
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 008 Device 002: ID 046d:c05a Logitech, Inc. M90/M100 Optical Mouse
Bus 008 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 006 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 007: ID 2949:8241
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
support@neoway:~$ ls /dev/ttyUSB*
/dev/ttyUSB0 /dev/ttyUSB1 /dev/ttyUSB2
support@neoway:~$
```

3. Test AT commands.

- Using a serial port debugging tool
Open the UART debugging tool and configure the UART port.
minicom -s

```
+-----+
| A -   Serial Device       : /dev/ttyUSB2
| B - Lockfile Location    : /var/lock
| C -   Callin Program     :
| D -   Callout Program    :
| E -   Bps/Par/Bits       : 115200 8N1
| F - Hardware Flow Control : No
| G - Software Flow Control : No
|
|   Change which setting?
+-----+
```

Configure the setting as shown in the above figure.

Send AT commands.

```
Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Jan  1 2014, 17:13:19.
Port /dev/ttyUSB2, 14:47:50

Press CTRL-A Z for help on special keys

ATE
OK
AT+CGMR
+CGMR: N720_D0B0CM_BZ_V003
OK
```

ATE //Enable terminal display.

OK

AT+CGMR //Query software version.

+CGMR: N720_D0B0CM_BZ_V003

AT+CPIN? //Query PIN code.

+CPIN: READY

AT+CSQ //Query signal.

+CSQ: 31,99 //RSSI: 31; bit error rate: 99

AT+CREG? //Query the registration status of the CS domain (SMS and voice call service)

+CREG: 0,1

AT+CGREG? ///Query the registration status of the PS domain (data service)

+CGREG: 0,1

AT\$MYSYSINFO //Query the mode of the network registered.

\$MYSYSINFO: 4,03 //Registered China Telecom 4G network.

AT+COPS? //Query carrier's information.

+COPS: 0,0,"CHINA TELECOM",7 //China Telecom

- Testing AT commands in CLI

```
echo -e "AT+CSQ\r\n" > /dev/ttyUSB2;cat /dev/ttyUSB2
```



- Only modem port or AT port can be used to send AT commands.
- If any messages are displayed to ask for permission, modify the permission for the ttyUSB port.

```
support@neoway:/dev$ echo -e "AT+CSQ\r\n" > /dev/ttyUSB2;cat /dev/ttyUSB2
-bash: /dev/ttyUSB2: Permission denied
cat: /dev/ttyUSB2: Permission denied
support@neoway:/dev$
```

```
support@neoway:/dev$ echo -e "AT+CSQ\r\n" > /dev/ttyUSB2;cat /dev/ttyUSB2
+CSQ: 22,99
OK
```

- If the AT command contains any special character, add \ before it.

```
support@neoway:~$ echo -e "AT$MYGPSPWR=1\r\n" >/dev/ttyUSB2 ; cat /dev/ttyUSB2
AT=1
ERROR
^C
support@neoway:~$ echo -e "AT\$MYGPSPWR=1\r\n" >/dev/ttyUSB2 ; cat /dev/ttyUSB2
AT$MYGPSPWR=1
OK
```

3 Establishing Connection Using pppd

All three configuration modes of N720 VID:0x2949 PID:0x8241, VID:0x2949 PID:0x8242, and VID:0x2949 PID:0x8243 support network dialup using pppd. The following sections describe how to dial up and establish a PPP connection using pppd.

3.1 Checking Device

Execute **lsusb** to check if N720 is identified.

```
support@neoway:~$ lsusb
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 008 Device 002: ID 046d:c05a Logitech, Inc. M90/M100 Optical Mouse
Bus 008 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 006 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 006: ID 2949:8241
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

N720 has been identified with a VID of 0x2949 and a PID of 0x8241.

3.2 Adding VID and PID

In **kernel/drivers/usb/serial/option.c**, add **VID:0x2949 PID:0x8241** to **option_ids**.

```
static const struct usb_device_id option_ids[] = {
    { USB_DEVICE_INTERFACE_PROTOCOL(0x2949,0x8241,0)}, //neoway n720 vid:0x2949 pid:0x8241
    { USB_DEVICE_INTERFACE_PROTOCOL(0x2949,0x8243,0)}, //neoway n720 vid:0x2949 pid:0x8243
}
```

3.3 Adding pppd Driver to Kernel

1. Navigate to the kernel directory and execute **make menuconfig**.
The kernel configuration GUI is displayed.
2. Configure kernel.
Select **Device Drivers** > **Network device support** > *ppp options*.

```

[*]    PPP (point-to-point protocol) support
<M>    PPP BSD-Compress compression
<M>    PPP Deflate compression
[*]    PPP filtering
<M>    PPP MPPE compression (encryption)
[*]    PPP multilink support
<M>    PPP over ATM
<M>    PPP over Ethernet
<M>    PPP over IPv4 (PPTP)
<M>    PPP over L2TP
<M>    PPP support for async serial ports
<M>    PPP support for sync tty ports
<M>    SLIP (serial line) support
[*]    CSLIP compressed headers

```

3. Click **Save** and **Exit**.
4. Compile kernel again and burn the compiled kernel files to the host.

3.4 Dialing up

1. Repeat the processes of USB loading and AT interaction.
Ensure that the module registers a network and the returned RSSI value is greater than 13.
2. Check whether the Linux OS embeds the pppd and chat tools.
If the system does not contain pppd, install kppp, which contains pppd program.
pppd 2.4.5 and pppd 2.4.7 are recommended.
3. Check if the following scripts are installed.

```

/etc/ppp/:
chap-secrets  ioptions      options  resolv.conf  connect-errors  ip-down
pap-secrets   ip-up         peers
/etc/ppp/peers:

```

ip-up: a script that configures settings after a connection is set up.

ip-down: a script that is used after the connection is disconnected.

pppd call gpra-dial: executed to dial up to set up a connection.

gprs-chat: contains the AT commands and response to them between chat and the GPRS module.

4. Check the carrier information and network mode, and the set APN.

```
AT+CGDCONT=1,"IP","APN_XXX"
```

OK

XXX indicates the carrier APN. For example,

2G, 3G, and 4G of China Mobile: CMNET

2G, 3G, and 4G of China Unicom: 3GNET

2G and 3G of China Telecom: CTNET

4G of China Telecom: CTLTE

For other networks, consult your carriers.

 NOTE

- The module will set APN based on current network mode if developers do not set it before PPPD dialup. If the APN is set before pppd dialup, the module will dial up the network using the set APN.
 - If developers use a private network SIM card, obtain the private network APN from carrier.
5. Execute *99# to dial up network.

 NOTE

If developers use a private network SIM card or other SIM cards, obtain user and password from your carriers when necessary.

6. Create a pppd script and a chat script.
- /etc/ppp/peers/gprs-dial

```
/dev/ttyUSB0
115200
hide-password
noauth
```

```
noipdefault
local
lock
dump
nodetach
user "card"
password "card"
remotename cmnet
ipparam cmnet
```

 NOTE

Set the modem port as the dialup port. Otherwise, the module cannot dial up a network successfully.

- /etc/ppp/gprs-chat

```

TIMEOUT 5
ABORT "BUSY"
ABORT "DELAYED"
ABORT "ERROR"
ABORT "NODIALTONE"
ABORT "NOCARRIER"
TIMEOUT 5

"AT
OK AT+CSQ

```

7. Execute `pppd call gprs-dial &`.

```

Serial connection established.
using channel 1
Using interface ppp0
Connect: ppp0 <-> /dev/ttyUSB0
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x2e36ba1d> <pcomp> <accomp>]
rcvd [LCP ConfReq id=0x0 <asyncmap 0x0> <auth pap> <magic 0x77f80375> <pcomp> <accomp>]
sent [LCP ConfAck id=0x0 <asyncmap 0x0> <auth pap> <magic 0x77f80375> <pcomp> <accomp>]
rcvd [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0x2e36ba1d> <pcomp> <accomp>]
sent [PAP AuthReq id=0x1 user="card" password=<hidden>]
rcvd [LCP DiscReq id=0x1 magic=0x77f80375]
rcvd [PAP AuthAck id=0x1 ""]
PAP authentication succeeded
sent [CCP ConfReq id=0x1 <deflate 15> <deflate(old#) 15> <bsd v1 15>]
sent [IPCP ConfReq id=0x1 <compress VJ 0f 01> <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
rcvd [LCP ProtRej id=0x2 80 fd 01 01 00 0f 1a 04 78 00 18 04 78 00 15 03 2f]
Protocol-Reject for 'Compression Control Protocol' (0x80fd) received
rcvd [IPCP ConfReq id=0x0]
sent [IPCP ConfNak id=0x0 <addr 0.0.0.0>]
rcvd [IPCP ConfRej id=0x1 <compress VJ 0f 01>]
sent [IPCP ConfReq id=0x2 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
rcvd [IPCP ConfReq id=0x1]
sent [IPCP ConfAck id=0x1]
rcvd [IPCP ConfNak id=0x2 <addr 10.171.109.99> <ms-dns1 202.96.128.86> <ms-dns2 202.96.134.133>]
sent [IPCP ConfReq id=0x3 <addr 10.171.109.99> <ms-dns1 202.96.128.86> <ms-dns2 202.96.134.133>]
rcvd [IPCP ConfAck id=0x3 <addr 10.171.109.99> <ms-dns1 202.96.128.86> <ms-dns2 202.96.134.133>]
Could not determine remote IP address: defaulting to 10.64.64.64
not replacing existing default route via 192.168.12.254
Cannot determine ethernet address for proxy ARP
local IP address 10.171.109.99
remote IP address 10.64.64.64
primary DNS address 202.96.128.86
secondary DNS address 202.96.134.133
Script /etc/ppp/ip-up started (pid 3355)
Script /etc/ppp/ip-up finished (pid 3355), status = 0x0

root@neoway:/home/support#
root@neoway:/home/support# pppd call gprs-dial &

```

3.5 Test and Disconnection

3.5.1 Testing the Connection

Input **ifconfig** to check the IP address of ppp0.

```
ppp0      Link encap:Point-to-Point Protocol  
          inet addr:10.171.109.99 P-t-P:10.64.64.64  Mask:255.255.255.255  
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1  
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:3  
          RX bytes:62 (62.0 B)  TX bytes:101 (101.0 B)
```

Ping a valid IP address to check if the host is connected to the Internet.

```
root@neoway:/home/support# ping 115.239.210.27  
PING 115.239.210.27 (115.239.210.27) 56(84) bytes of data.  
64 bytes from 115.239.210.27: icmp_seq=1 ttl=53 time=31.2 ms  
64 bytes from 115.239.210.27: icmp_seq=2 ttl=53 time=30.6 ms  
64 bytes from 115.239.210.27: icmp_seq=3 ttl=53 time=30.4 ms  
64 bytes from 115.239.210.27: icmp_seq=5 ttl=53 time=30.0 ms  
64 bytes from 115.239.210.27: icmp_seq=6 ttl=53 time=31.7 ms
```

Ping a valid domain name.

```
root@neoway:/home/support# ping www.baidu.com  
PING www.a.shifen.com (14.215.177.38) 56(84) bytes of data.  
64 bytes from 14.215.177.38: icmp_seq=1 ttl=54 time=6.05 ms  
64 bytes from 14.215.177.38: icmp_seq=2 ttl=54 time=6.97 ms  
64 bytes from 14.215.177.38: icmp_seq=3 ttl=54 time=7.51 ms
```

If a website can be reached through IP address but cannot be reached through its domain name, add DNS(114.114.114.114) to **/etc/resolv.conf**.

3.5.2 Disconnecting the Connection

1. Execute the **ppp-off** script.

```
root@neoway:/home/support/ppp# ./ppp-off
PPP link to ppp0 terminated.
Terminating on signal 2
Connect time 0.3 minutes.
Sent 0 bytes, received 0 bytes.
Script /etc/ppp/ip-down started (pid 3605)
root@neoway:/home/support/ppp# sent [LCP TermReq id=0x2 "User request"]
rcvd [LCP TermAck id=0x2]
Connection terminated.
Waiting for 1 child processes...
  script /etc/ppp/ip-down, pid 3605
Script /etc/ppp/ip-down finished (pid 3605), status = 0x0

[1]+  Exit 5                  pppd call gprs-dial
root@neoway:/home/support/ppp# █
```

2. Run the # **killall pppd** command.

```
root@neoway:/home/support/ppp# killall pppd
Terminating on signal 15
Connect time 31.7 minutes.
Sent 0 bytes, received 0 bytes.
Script /etc/ppp/ip-down started (pid 3504)
sent [LCP TermReq id=0x2 "User request"]
root@neoway:/home/support/ppp# rcvd [LCP TermAck id=0x2]
Connection terminated.
Waiting for 1 child processes...
  script /etc/ppp/ip-down, pid 3504
Script /etc/ppp/ip-down finished (pid 3504), status = 0x0
```

4 Establishing Connection Using RMNET

N720 supports dialup and connection setup using RMNET with a VID of 0x2949 and a PID of 0x8243.

GobitNet driver is used to interact with N720 on the Linux for network dialup through RMNET. GobitNet will create a network device called ethX and a /dev/qcqmIX character device. After the two devices are created and N720 can register a network, developers can dial up the network using the dialup program.



NOTE

Send **AT+NETSHAREMODE=2** to N720 to switch the configuration if the PID is not 0x8243.

4.1 Checking Device

Execute **lsusb** to check if N720 is identified.

```
support@neoway:~$ lsusb
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 008 Device 002: ID 046d:c05a Logitech, Inc. M90/M100 Optical Mouse
Bus 008 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 006 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 010: ID 2949:8243
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

N720 has been identified with a VID of 0x2949 and a PID of 0x8243.

4.2 Adding VID and PID

In **kernel/drivers/usb/serial/option.c**, add **VID:0x2949 PID:0x8243** to **option_ids**.

```
static const struct usb_device_id option_ids[] = {
    { USB_DEVICE_INTERFACE_PROTOCOL(0x2949,0x8241,0)}, //neoway n720 vid:0x2949 pid:0x8241
    { USB_DEVICE_INTERFACE_PROTOCOL(0x2949,0x8243,0)}, //neoway n720 vid:0x2949 pid:0x8243
}
```

4.3 Adding GobiNet Driver to Kernel

1. Copy the GobiNet driver code (NWY-GobiNet/src/) to **kernel/drivers/net/usb**.
2. Add config **USB_GOBI_NET** to **kernel/drivers/net/usb/Kconfig**.

```
config USB_GOBI_NET

tristate "Gobi USB Net driver for NWY module"

help

Support NWY module.

A modem manager with support for GobiNet is recommended.
```

3. Modify **kernel/drivers/net/usb/Makefile** to ensure that GobiNet module can be compiled.

```
obj-$(CONFIG_USB_GOBI_NET)      += GobiNet.o

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

4. Configure kernel. Select the GobiNet option and save the settings.

```
Device Drivers --->

[*]Network device support --->

    [*]USB Network Adapters --->

        [M]Gobi USB driver for NWY module
```

5. Compile the kernel again and burn the kernel to the host.
 6. Power up the host and check whether the driver is loaded successfully.
- Input **lsusb -t** to check whether the driver is identified.

```
support@neoway:~$ lsusb -t
/: Bus 08.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
   |__ Port 2: Dev 2, If 0, Class=Human Interface Device, Driver=usbhid, 1.5M
/: Bus 07.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
/: Bus 06.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
/: Bus 05.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
/: Bus 04.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
/: Bus 03.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
/: Bus 02.Port 1: Dev 1, Class=root_hub, Driver=ehci-pci/6p, 480M
/: Bus 01.Port 1: Dev 1, Class=root_hub, Driver=ehci-pci/6p, 480M
   |__ Port 5: Dev 11, If 0, Class=Vendor Specific Class, Driver=GobiNet, 480M
   |__ Port 5: Dev 11, If 1, Class=Vendor Specific Class, Driver=option, 480M
   |__ Port 5: Dev 11, If 2, Class=Vendor Specific Class, Driver=option, 480M
   |__ Port 5: Dev 11, If 3, Class=Vendor Specific Class, Driver=option, 480M
   |__ Port 5: Dev 11, If 4, Class=Vendor Specific Class, Driver=, 480M
support@neoway:~$
```

Input **ifconfig** to check NIC information.

If a new NIC, e.g. eth1, is displayed, the driver is loaded successfully.

```
eth1      Link encap:Ethernet  HWaddr ba:2b:25:c9:06:d3
          UP BROADCAST NOARP 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)
```

7. Check if **qcqmi1** is created successfully.

```
support@neoway:~$ ls /dev/qcqm*
/dev/qcqm1
support@neoway:~$
```

4.4 Configuring Before Dialup

To dial up and set a connection through RMNET on Linux, developers need to use Neoway dialup program.

Execute **make** under the **NWY-CM-code/src** directory to obtain the NWY-CM program. Assign the NWY-CM program execution permission.

```
root@neoway:/home/support/luo/rmnet/NWY-CM-code/src# make
rm -rf NWY-CM *~
gcc -Wall -s QmiWwanCM.c GobiNetCM.c main.c MPQMUX.c QMITHread.c util.c udhpc
.c -o NWY-CM -lpthread
main.c: In function 'usbnet_link_change':
main.c:42:24: warning: variable 'r' set but not used [-Wunused-but-set-variable]
    unsigned char *r;
                   ^
main.c:58:24: warning: unused variable 'gw' [-Wunused-variable]
    unsigned char *gw = (unsigned char *)&profile->ipv4.Gateway;
                   ^
util.c:47:21: warning: 'get_time' defined but not used [-Wunused-function]
    static const char * get_time(void) {
                       ^
```

Input **NWY-CM-h** to check **Help**.

```
support@neoway:~/luo/rmnet/NWY-CM-code/src$ ./NWY-CM -h
```

```
Usage: ./NWY-CM [-s [apn [user password auth]]] [-f logfilename]
```

```
-s [apn [user password auth]] Set apn/user/password/auth get from your network provider
```

```
-f logfilename                Save log message of this program to file
```

```
Example 1: ./NWY-CM
```

```
Example 2: ./NWY-CM -s 3gnet card pwd 0 -f log.txt
```

NWY-CM (Option) (Parameter)	Description		
-s [apn [user password auth]]	Set APN/user name/ password/ authentication method.	APN: Access point name	China Mobile: CMNET China Unicom: 3GNET 2G and 3G of China Telecom: CTNET 4G of China Telecom: CTLTE For other networks, consult your carriers.
		user: User name	Consult your carriers.
		password: password	
		auth: authentication method	0: No authentication 1: PAP 2: CHAP
-f logfilename	Save dialing log.		

4.5 Dialing Up Using RMNET

1. Run the following commands to confirm that N720 registered a network.

```
AT+CPIN?
```

```
AT+CSQ
```

```
AT+CREG?
```

```
AT+CGREG?
```

```
AT$MYSYSINFO
```

2. Execute NWY0CM to dial up and establish a network connection.

```

root@neoway:/home/support/luo/rmnet/NWY-CM-code/src# ./NWY-CM -s ctnet card card 1 &
[1] 4152
root@neoway:/home/support/luo/rmnet/NWY-CM-code/src# NWY-ConnetManager
./NWY-CM profile[1] = ctnet/card/card/1
Find qmichannel = /dev/qcqm1l
Find usbnet_adapter = eth1
Get clientWDS = 7
Get clientDMS = 8
Get clientNAS = 9
Get clientUIM = 10
Get clientWDA = 11
requestBaseBandVersion N720_D0B0CM_BZ_V003 1 [Aug 17 2017 01:00:00]
requestGetSIMStatus SIMStatus: SIM_READY
requestSetProfile[1] ctnet/card/card/1
requestGetProfile[1] ctnet/card/card/1
requestRegistrationState2 MCC: 460, MNC: 11, PS: Attached, DataCap: LTE
requestQueryDataCall ConnectionStatus: DISCONNECTED
requestGetIPAddress QMUXResult = 0x1, QMUXError = 0xf
busybox ifconfig eth1 0.0.0.0 netmask 0.0.0.0
ifconfig: SIOCSIFNETMASK: Cannot assign requested address
busybox ifconfig eth1 0.0.0.0 netmask 0.0.0.0
ifconfig: SIOCSIFNETMASK: Cannot assign requested address
busybox route add default dev eth1
Adding DNS 0.0.0.0 0.0.0.0
requestRegistrationState2 MCC: 460, MNC: 11, PS: Attached, DataCap: LTE
requestSetupDataCall WdsConnectionIPv4Handle: 0x872bce90
requestQueryDataCall ConnectionStatus: CONNECTED
busybox ifconfig eth1 10.177.202.227 netmask 255.255.255.248
busybox ifconfig eth1 10.177.202.227 netmask 255.255.255.248
busybox route add default dev eth1
route: SIOCADDRT: File exists
Adding DNS 202.96.128.86 202.96.134.133

```

4.6 Test and Disconnection

4.6.1 Testing the Connection

Input **ifconfig** to check the IP address of eth1.

```

eth1      Link encap:Ethernet  HWaddr ba:2b:25:c9:06:d3
          inet addr:10.177.202.227  Bcast:10.177.202.231  Mask:255.255.255.248
          inet6 addr: fe80::b82b:25ff:fec9:6d3/64 Scope:Link
          UP BROADCAST RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:23 errors:0 dropped:0 overruns:0 frame:0
          TX packets:52 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2060 (2.0 KB)  TX bytes:5505 (5.5 KB)

```

Ping a valid IP address to check if the host is connected to the Internet.

```
root@neoway:/home/support# ping 115.239.210.27
PING 115.239.210.27 (115.239.210.27) 56(84) bytes of data.
64 bytes from 115.239.210.27: icmp_seq=1 ttl=53 time=31.2 ms
64 bytes from 115.239.210.27: icmp_seq=2 ttl=53 time=30.6 ms
64 bytes from 115.239.210.27: icmp_seq=3 ttl=53 time=30.4 ms
64 bytes from 115.239.210.27: icmp_seq=5 ttl=53 time=30.0 ms
64 bytes from 115.239.210.27: icmp_seq=6 ttl=53 time=31.7 ms
```

Ping a valid domain name.

4.6.2 Disconnecting the Connection

Run the **killall NWY-CM** command.

```
root@neoway:/home/support/luo/rmnet/NWY-CM-code/src# killall NWY-CM
root@neoway:/home/support/luo/rmnet/NWY-CM-code/src# requestDeactivateDefaultPDP err = 0
busybox ifconfig eth1 10.177.202.227 netmask 255.255.255.248
busybox ifconfig eth1 10.177.202.227 netmask 255.255.255.248
busybox route add default dev eth1
route: SIOCADDRT: File exists
Adding DNS 202.96.128.86 202.96.134.133
GobiNetThread exit
main exit

[1]+  Done                  ./NWY-CM -s ctnet card card 1
```

5 Establishing Connection Using RNDIS

N720 supports dialup and connection setup using RMNET with a VID of 0x2949 and a PID of 0x8241.



Send **AT+NETSHAREMODE=2** to N720 to switch the configuration if the PID is not 0x8241.

5.1 Checking Device

Execute **lsusb** to check if N720 is identified.

```
support@neoway:~$ lsusb
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 008 Device 002: ID 046d:c05a Logitech, Inc. M90/M100 Optical Mouse
Bus 008 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 006 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 006: ID 2949:8241
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

N720 has been identified with a VID of 0x2949 and a PID of 0x8241.

5.2 Adding VID and PID

In **kernel/drivers/usb/serial/option.c**, add **VID:0x2949 PID:0x8241** to **option_ids**.

```
static const struct usb_device_id option_ids[] = {
    { USB_DEVICE_INTERFACE_PROTOCOL(0x2949,0x8241,0)}, //neoway n720 vid:0x2949 pid:0x8241
    { USB_DEVICE_INTERFACE_PROTOCOL(0x2949,0x8243,0)}, //neoway n720 vid:0x2949 pid:0x8243
}
```

5.3 Adding RNDIS Driver to Kernel

1. Navigate to **kernel** and execute **make menuconfig**.
2. Configure kernel. Select RNDIS option and save the settings.

```
Device Drivers --->
*- Network device support --->
{M} USB Network Adapters --->
-M- Multi-purpose USB Networking Framework (CONFIG_USB_NET=m)
    -M- Host for RNDIS and ActiveSync devices (CONFIG_USB_NET_RNDIS_HOST=m)
```

3. Compile the kernel again and burn the kernel to the host.
4. Power up the host and check whether the usb0 is displayed.

```
usb0      Link encap:Ethernet  HWaddr 8e:0a:1c:dd:d9:c1
          inet addr:192.168.225.36  Bcast:192.168.225.255  Mask:255.255.255.0
          inet6 addr: fe80::8c0a:1cff:fedd:d9c1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:166 errors:0 dropped:0 overruns:0 frame:0
          TX packets:290 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6826 (6.8 KB)  TX bytes:41157 (41.1 KB)
```

5. Query whether the RNDIS driver is loaded successfully.

Input **lsusb -t**.

```
support@neoway:~$ lsusb -t
/: Bus 08.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
   |__ Port 2: Dev 2, If 0, Class=Human Interface Device, Driver=usbhid, 1.5M
/: Bus 07.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
/: Bus 06.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
/: Bus 05.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
/: Bus 04.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
/: Bus 03.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
/: Bus 02.Port 1: Dev 1, Class=root_hub, Driver=ehci-pci/6p, 480M
/: Bus 01.Port 1: Dev 1, Class=root_hub, Driver=ehci-pci/6p, 480M
   |__ Port 5: Dev 12, If 0, Class=Communications, Driver=rndis_host, 480M
   |__ Port 5: Dev 12, If 1, Class=CDC Data, Driver=rndis_host, 480M
   |__ Port 5: Dev 12, If 2, Class=Vendor Specific Class, Driver=option, 480M
   |__ Port 5: Dev 12, If 3, Class=Vendor Specific Class, Driver=option, 480M
   |__ Port 5: Dev 12, If 4, Class=Vendor Specific Class, Driver=option, 480M
   |__ Port 5: Dev 12, If 5, Class=Vendor Specific Class, Driver=, 480M
```

Input **lsmod | grep rndis**.

```
support@neoway:~$ lsmod | grep rndis
rndis_wlan          53248  0
rndis_host          16384  1 rndis_wlan
cfg80211            561152  1 rndis_wlan
cdc_ether            16384  1 rndis_host
usbnet               45056  4 rndis_host,rndis_wlan,GobiNet,cdc_ether
support@neoway:~$
```

5.4 Dialing Up Using RNDIS

1. Run the following commands to confirm that N720 registered a network.

AT+CPIN? //Query SIM card status.

AT+CSQ //Query signal.

AT+CREG?

AT+CGREG?

AT\$MYSYSINFO //Query network mode.

```
AT+CPIN?  
+CPIN: READY  
OK  
AT+CSQ  
+CSQ: 19,99  
OK  
AT+CREG?  
+CREG: 0,1  
OK  
AT+CGREG?  
+CGREG: 0,1  
OK  
AT$MYSYSINFO  
$MYSYSINFO: 4,03  
OK
```

2. Send AT commands to dial up.

```
AT+NETSHAREACT=?  
+NETSHAREACT: (0-11),(0-1),(0-1),"apn","user","passwd",(0-3)  
OK
```

```
AT+NETSHAREACT=1,1,0  
OK  
AT+NETSHAREACT?  
+NETSHAREACT: 1,0,,, "IPV4",RNDIS  
OK
```

NOTE

After dialing up and establishing a connection using RNDIS, the module cannot dial up or establish a connection using pppd anymore.

5.5 Test and Disconnection

5.5.1 Testing the Connection

Input **ifconfig** to check the IP address of usb0.

```
usb0      Link encap:Ethernet  HWaddr 8e:0a:1c:dd:d9:c1
          inet addr:192.168.225.36  Bcast:192.168.225.255  Mask:255.255.255.0
          inet6 addr: fe80::8c0a:1cff:fedd:d9c1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:199 errors:0 dropped:0 overruns:0 frame:0
          TX packets:329 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7942 (7.9 KB)  TX bytes:45316 (45.3 KB)
```

Ping a valid website to see if the connection works properly.

5.5.2 Disconnecting the Connection

Send AT+NETSHAREACT=1,0,0 to disconnect the connection.

```
OK
AT+NETSHAREACT=1,0,0
OK
AT+NETSHAREACT?
+NETSHAREACT: 0,0,,, "IPV4", RNDIS
OK
```

6 Establishing Connection Using ECM

N720 supports dialup and connection setup using ECM with a VID of 0x2949 and a PID of 0x8242.



Send **AT+NETSHAREMODE=2** to N720 to switch the configuration if the PID is not 0x8242.

6.1 Checking Device

Execute **lsusb** to check if N720 is identified.

```
support@neoway:~$ lsusb
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 008 Device 002: ID 046d:c05a Logitech, Inc. M90/M100 Optical Mouse
Bus 008 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 007 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 006 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 013: ID 2949:8242
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
support@neoway:~$
```

N720 has been identified with a VID of 0x2949 and a PID of 0x8242.

6.2 Adding VID and PID

In **kernel/drivers/usb/serial/option.c**, add **VID:0x2949 PID:0x8242** to **option_ids**.

```
static const struct option_blacklist_info n720_8242_blacklist = {
    .reserved = BIT(0) | BIT(1) | BIT(3) | BIT(5),
};

static const struct usb_device_id option_ids[] = {
    { USB_DEVICE_INTERFACE_PROTOCOL(0x2949, 0x8241, 0)}, // neoway n720 vid:0x2949 pid:0x8241
    { USB_DEVICE_INTERFACE_PROTOCOL(0x2949, 0x8243, 0)}, // neoway n720 vid:0x2949 pid:0x8243
    { USB_DEVICE(0x2949, 0x8242),
      .driver_info = (kernel_ulong_t)&n720_8242_blacklist },
}
```

6.3 Adding ECM Driver to Kernel

1. Navigate to **kernel** and execute **make menuconfig**.
2. Configure kernel. Select **RNDIIS** option and save the settings.

```
Device Drivers --->
```

```
-*- Network device support --->
```

```
{M} USB Network Adapters --->
```

```
-M- Multi-purpose USB Networking Framework (CONFIG_USB_NET=m)
```

```
-M- CDC Ethernet support (smart devices such as cable modems)
(CONFIG_USB_NET_CDCETHER=m)
```

3. Compile the kernel again and burn the kernel to the host.
4. Power up the host and check whether the usb0 is displayed.

```
usb0      Link encap:Ethernet  HWaddr f6:c1:7b:bf:be:11
          inet addr:192.168.225.48  Bcast:192.168.225.255  Mask:255.255.255.0
          inet6 addr: fe80::f4c1:7bff:febf:be11/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:20 errors:0 dropped:0 overruns:0 frame:0
          TX packets:52 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1606 (1.6 KB)  TX bytes:8652 (8.6 KB)
```

5. Query whether the ECM driver is loaded successfully.

Input **lsusb -t**.

```
support@neoway:~$ lsusb -t
/: Bus 08.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
   |__ Port 2: Dev 2, If 0, Class=Human Interface Device, Driver=usbhid, 1.5M
/: Bus 07.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
/: Bus 06.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
/: Bus 05.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
/: Bus 04.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
/: Bus 03.Port 1: Dev 1, Class=root_hub, Driver=uhci_hcd/2p, 12M
/: Bus 02.Port 1: Dev 1, Class=root_hub, Driver=ehci-pci/6p, 480M
/: Bus 01.Port 1: Dev 1, Class=root_hub, Driver=ehci-pci/6p, 480M
   |__ Port 5: Dev 13, If 0, Class=Communications, Driver=cdc_ether, 480M
   |__ Port 5: Dev 13, If 1, Class=CDC Data, Driver=cdc_ether, 480M
   |__ Port 5: Dev 13, If 2, Class=Vendor Specific Class, Driver=option, 480M
   |__ Port 5: Dev 13, If 3, Class=Vendor Specific Class, Driver=, 480M
   |__ Port 5: Dev 13, If 4, Class=Vendor Specific Class, Driver=option, 480M
   |__ Port 5: Dev 13, If 5, Class=Vendor Specific Class, Driver=, 480M
```

Input **lsmod | grep cdc**.

```
support@neoway:~$ lsmod | grep cdc
cdc_ether                16384  1 rndis_host
usbnet                   45056  4 rndis_host,rndis_wlan,GobiNet,cdc_ether
support@neoway:~$
```

6.4 Dialing Up Using RNDIS

1. Run the following commands to confirm that N720 registered a network.

```
AT+CPIN?    //Query SIM card status.
AT+CSQ      //Query signal.
AT+CREG?
AT+CGREG?
AT$MYSYSINFO //Query network mode.
```

```
AT+CPIN?
+CPIN: READY
OK
AT+CSQ
+CSQ: 19,99
OK
AT+CREG?
+CREG: 0,1
OK
AT+CGREG?
+CGREG: 0,1
OK
AT$MYSYSINFO
$MYSYSINFO: 4,03
OK
```

2. Send AT commands to dial up.

```
AT+NETSHAREACT=?
+NETSHAREACT: (0-11),(0-1),(0-1),"apn","user","passwd",(0-3)
OK
```

```
AT+NETSHAREACT=1,1,0
OK
AT+NETSHAREACT?
+NETSHAREACT: 1,0,,, "IPV4",ECM
OK
```

NOTE

After dialing up and establishing a connection using ECM, the module cannot dial up or establish a connection using pppd anymore.

6.5 Test and Disconnection

6.5.1 Testing the Connection

Input **ifconfig** to check the IP address of usb0.

```
usb0      Link encap:Ethernet  HWaddr f6:c1:7b:bf:be:11
          inet addr:192.168.225.48  Bcast:192.168.225.255  Mask:255.255.255.0
          inet6 addr: fe80::f4c1:7bff:febf:be11/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:55 errors:0 dropped:0 overruns:0 frame:0
          TX packets:143 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2874 (2.8 KB)  TX bytes:18102 (18.1 KB)
```

Ping a valid website to see if the connection works properly.

6.5.2 Disconnecting the Connection

Send AT+NETSHAREACT=1,0,0 to disconnect the connection.

```
AT+NETSHAREACT=1,0,0
OK
AT+NETSHAREACT?
+NETSHAREACT: 0,0,,,"IPV4",ECM
OK
```

7 FAQ

Q: Why I cannot find the USB ports after adding the VID and PID of the module to the kernel and executing **ls /dev/ttyUSB***?

A:

1. Check if the module is powered up and connected through USB.
Input **lsusb** or **dmesg** to check the USB device information.
2. Check the VID and PID are added correctly.
3. Check if the modifications are compiled by the system.

You can find the USB ports by inputting **ls /dev/ttyUSB*** if all the above are correct.