



Perfect Wireless Experience
完美无线体验

FIBOCOM NL668

Application Guide _Linux ECM Dial

Version: V1.0.0

Date: 2018.04.13



Applicability type

S/N	Product Model	Description
1	NL668-CN-00	MCP is 4+2, supports MAIN_ANT、DIV_ANT、GNSS_ANT
2	NL668-CN-01	MCP is 2+1, supports MAIN_ANT
3	NL668-CN-02	MCP is 2+1, supports MAIN_ANT、DIV_ANT
4	NL668-EAU-00	MCP is 4+2, supports MAIN_ANT、DIV_ANT、GNSS_ANT

Copyright

Copyright ©2018 Fibocom Wireless Inc. All rights reserved.

Without the prior written permission of the copyright holder, any company or individual is prohibited to excerpt, copy any part of or the entire document, or transmit the document in any form.

Attention

The document is subject to update from time to time owing to the product version upgrade or other reasons. Unless otherwise specified, the document only serves as the user guide. All the statements, information and suggestions contained in the document do not constitute any explicit or implicit guarantee.

Trademark



The trademark is registered and owned by Fibocom Wireless Inc.

Versions

Version	Update Date	Description
V1.0.0	2018-04-13	Initial version

Contents

1	ECM Dial Principle.....	5
2	USB Port Information.....	5
3	USB Enumeration Mode Confirmation	5
4	Kernel Environment Confirmation	6
4.1	Kernel Compile Configuration Item	6
4.2	Kernel Configuration Steps	6
4.3	ECM Driver Loading Confirmation	7
4.4	USB Serial Port Driver Loading Confirmation.....	7
5	ECM Dial Process.....	8
5.1	Dial Command.....	8
5.2	Dial Command Usage Process.....	9
5.3	APN/Username/Password/Authentication Method Settings	10

1 ECM Dial Principle

This document describes ECM dial for the NL668-CN series wireless module. When the ECM dials, the NL668-CN module acts as a router. The internal AP invokes the router management process to implement WWAN dial. After the dial is successful, the DHCP server and other functions are activated inside the module. The client application system obtains the IP address allocated by the DHCP server of the module through the DHCP client.

2 USB Port Information

Vendor ID: 0x1508 Product ID:0x1001		
Interface Number	Interface Function	Interface Name
0	DEBUG	Device Diagnostic Interface
1	MODEM	Modem Connector
2	AT	Device Application Interface
3	PIPE	Device Pipe
4	ECM	CDC Control USB Device Interface
5	ECM	CDC Data USB Device Interface
6	ADB	Android Composite ADB Interface Ic

3 USB Enumeration Mode Confirmation

Use AT+GTUSBMODE? to query the current USB mode. (USBNET mode default is RMNET)

AT+GTUSBMODE?

+GTUSBMODE: 18

OK

If the return value is not 18, it indicates that the module has not entered the ECM mode. Please send AT+GTUSBMODE=18 to return, and restart the module to switch the module to ECM mode.

4 Kernel Environment Confirmation

The ECM is a standard NIC port. If you use ECM devices, make sure to load the CDCETHER driver in the system. The current KERENL default is opening the CDCETHER driver support. If it is not opened, please refer to the following steps.

4.1 Kernel Compile Configuration Item

CONFIG_USB_NET_CDCETHER=y

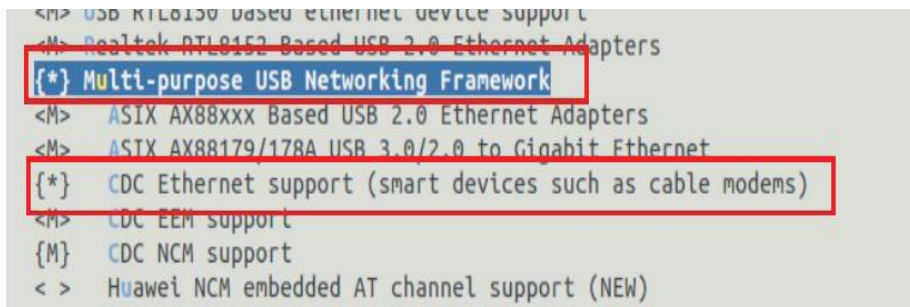
CONFIG_USB_USBNET=y

4.2 Kernel Configuration Steps

1: Enter the kernel directory (assuming "/home/ght/linux-3.08/") and make the <configuration> command (assuming standard make menuconfig).

2: Complete the ECM drive configuration according to the following steps:

Go to Device Drivers -> Network device support -> USB Network Adapters menu and select Multi-purpose USB Networking Framework and CDC Ethernet support (smart devices such as cable modems) options:



3: After configuration as above, exit the configuration interface step by step by selecting "<Exit>". Finally, in the save interface, select "<Yes>" and exit.

4: After completing the configuration, run the make command to compile the modified kernel.



Note:

With ECM mode, you need to pay special attention to whether the relevant parts of the Linux system have been modified by the client. In particular, in /kernel/drivers/net/usb/cdc_ether.c, do not delete the following items in the array products[]:

{USB_INTERFACE_INFO(USB_CLASS_COMM, USB_CDC_SUBCLASS_ETHERNET,

```
USB_CDC_PROTO_NONE),
    .driver_info = (unsigned long) &cdc_info,}
```

4.3 ECM Driver Loading Confirmation

1: Confirm that cdc_ether.ko or cdc_ether.o is already in the kernel. If *.ko is not loaded, load it as follows:

```
root@ubuntu:/# insmod cdc_ether.ko
```

or

```
root@ubuntu:/# modprobe cdc_ether
```

2: After loading, execute the following to confirm whether the loading is successful:

```
root@ubuntu:/# lsmod | grep cdc_ether // whether the driver is loaded successfully
```

```
cdc_ether      13502      0
usbnet        31972      1    cdc_ether
```

Use the ifconfig command to query the following devices:

```
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)
```

4.4 USB Serial Port Driver Loading Confirmation

Use the lsusb command to query whether 1508:1001 device information exists. If so, it means that the NL668 USB enumeration has been identified.

If not, use modprobe usbserial vendor=0x1508 product=0x1001 to load the USB serial port driver. After loading is successful, using ls /dev/ttyUSB*, you can query the tty device of ttyUSB0~ttyUSB4, as shown in the following table:

```
root@ubuntu:/# lsusb
Bus 001 Device 005: ID 1508:1001
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:/# modprobe usbserial vendor=0x1508 product=0x1001
root@ubuntu:/# ls /dev/ttyUSB*
/dev/ttyUSB0    /dev/ttyUSB1  /dev/ttyUSB2  /dev/ttyUSB3  /dev/ttyUSB4

```



Note:

For the ECM mode, if you use the dynamic load driver, you must first load cdc_ether and then load the usbserial driver. Otherwise, the module network port will enumerate in the form of ttyUSB, and the USB0 network port cannot be generated.

5 ECM Dial Process

5.1 Dial Command

AT+GTRNDIS command format is as follows:

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>: integer type; profile id used by RNDIS and specified with AT+CGDCONT

<ip>: string type; IP address assigned by network to RNDIS device via PDP context activate accept

<pdns>: string type; primary DNS assigned by network via PDP context activate accept

<sdns>: string type; secondary DNS assigned by network via PDP context activate accept

5.2 Dial Command Usage Process

If you need to access the Internet, the dial process is recommended (take a Telecom card as an example), as follows:

```
AT+CPIN?  
+CPIN: READY //Make sure your SIM card is ready  
OK  
AT+CSQ  
+CSQ: 21,99 //Ensure that the module can receive signal  
OK  
AT+COPS?  
+COPS: 0,0,"CHN-TELECOM",7 //Ensure that the module register on carrier network  
OK  
AT+CGREG?  
+CGREG: 0,1 // Module attached to data network  
OK  
AT+CEREG? // Module attached to LTE network  
+CEREG: 0,1  
OK  
AT+PSRAT  
+PSRAT: SRLTE // Extended instruction, NL668 registered data network type query  
OK  
AT+CGDCONT=1,"IP","ctlte" // Set access point information  
OK  
AT+CGDCONT?  
+CGDCONT: 1,"IP","ctlte","0.0.0.0",0,0,0,0 // Query if it has set successfully  
OK  
AT+GTRNDIS=1,1  
OK  
.....
```

```
AT+GTRNDIS?
```

```
+GTRNDIS: 1,1,100.85.126.41,202.101.172.37, 202.101.173.157 // Check the dial status  
regularly to see if the module has obtained the IP.
```

```
OK
```

```
.....
```

AT + GTRNDIS return to OK, and issue AT + GTRNDIS?, query if dial is successful? After the command returns to IP, the client needs to start the DHCP client to obtain the IP. (Enter `udhcpc -i usb0` & or `dhclient usb0` to run the dhcp service in the background.)

Dhcp successfully obtains the IP address and prints information as follows:

```
Sending select for 100.85.126.41...  
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 202.101.172.37  
/usr/share/udhcpc/default.script: Adding DNS 202.101.173.157
```

If you need to disconnect the network, first kill the dhcp client related process, and issue the following:

```
AT+GTRNDIS=0,1  
OK  
AT+GTRNDIS?  
+GTRNDIS: 0,0,0,0,0  
OK
```

5.3 APN/Username/Password/Authentication Method Settings

See the *NL668 Application Business Process Manual*