Fibocom

Perfect Wireless Experience
完美无线体验

L8 Family System Driver Integration and Application Guidance

Document version: V2.0.0

Update date: 11.24.2015

Applicability Table

No.	Product model	Description
1	L810-GL-00	NA
2	L810-GL-01	NA
3	L830-GL-00	NA
4	L830-GL-01	NA
5	L830-EA-00	NA
6	L831-EA-00	NA
7	L831-EA-01	NA

Copyright

Notice

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.

Version Record

Version	Update Date	Remarks
V1.0.0	10-17-2015	Initial version
V2.0.0	11-24-2015	Optimize document name and title; Optimize document structure and content; Optimize document figure;

Contents

1 Introduction

1.1 Purpose

- This article is the guidance for driver integration development activities for L8 series 4G module devices based on Android/Linux/Win8.1/Win10 systems. This document is mainly for driver developers for product developers based on the above systems.

1.2 Scope

The document applies to the following:

- Win8.1/Win10
- Android 4.0 and higher version.
- Linux2.6.22 and higher version.

2 Instructions for Linux system

2.1 Linux Kernel Device Driver Architecture

Linux kernel will load the USB driver according to the USB device interfaces reported by the 4G module. After the correct driver is loaded, the module can begin to work.

Linux kernel driver architecture of the Linux system is shown in Figure 2-1:

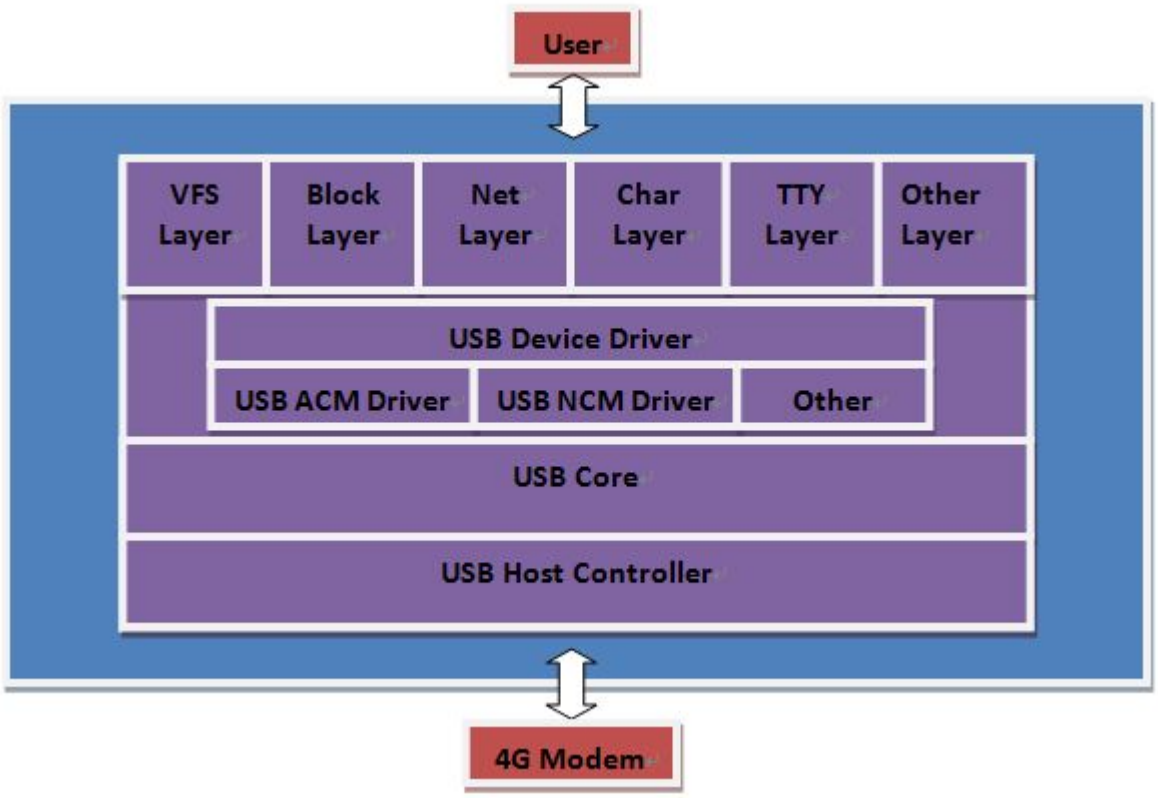


Figure 2-1 Driver Architecture

As is shown in Figure 2-1, driver modules related to 4G devices in the USB driver architecture of Linux system are: USB ACM and USB NCM driver modules.

Attention：

ACM Driver: USB ACM driver supports modem ports, AT ports and so on; the source code (cdc-acm.c) of the ACM driver is built-in in the Linux kernel.

NCM driver: The standard NCM network device drivers for USB, mainly for the transmission of network data.

2.2 Linux NCM Driver Integration

2.2.1 NCM Driver Porting

2.2.1.1 Linux 2.6.38 and Above Kernel Integration

For Linux 2.6.38 and above kernels, please add the configuration in accordance with Section 2.2.2.

2.2.1.2 Linux 2.6.32 to 38 Kernel Integration

Please refer to the following steps for integration:

1. Login https://www.kernel.org/pub/linux/kernel/v2.6/, and download source code for 2.6.38 above kernel
2. Decompress the downloaded kernel source code and copy the following files to the corresponding directory

   drivers/net/usb/cdc_ncm.c

   include/linux/usb/cdc.h

   include/linux/atomic.h
3. Modify Makefile and Kconfig files under the drivers/net/usb/ directory of source code

   Add obj-$(CONFIG_USB_NET_CDC_NCM) += cdc_ncm.oto the end of Makefile.

   Add the following statements to the end of Kconfig file.

   config USB_NET_CDC_NCM

   tristate "CDC NCM support"

   depends on USB_USBNET

   default y
4. Modify include/linux/usbnet.h

   Add #define FLAG_MULTI_PACKET 0x2000 and int (*manage_power)(struct usbnet *, int); add the above to the file, and the adding location can refer to usbnet.h in 2.6.38.

2.2.1.3 Linux 2.6.26 Kernel Integration

Please refer to the following steps for integration:

1. Login https://www.kernel.org/pub/linux/kernel/v2.6/, and download the source code for kernel 2.6.38
2. Decompress the downloaded kernel source code and copy the following files to the corresponding directory

   drivers/net/usb/cdc_ncm.c

   include/linux/usb/cdc.h

   include/linux/atomic.h
3. Modify drivers/net/usb/usbnet.c

   Delete static statement of usbnet_start_xmitfunction:

   int usbnet_start_xmit (struct sk_buff *skb, struct net_device *net)

   and add the following statements:

   EXPORT_SYMBOL_GPL(usbnet_start_xmit);
4. Modify the Makefile and Kconfig files under the drivers/net/usb/ directory of source code

   Add obj-$(CONFIG_USB_NET_CDC_NCM) += cdc_ncm.o to the end of Makefile.

   Add the following statements to the end of Kconfig file.

   config USB_NET_CDC_NCM

   tristate "CDC NCM support"

   depends on USB_USBNET

   default y
5. Modify include/linux/usbnet.h

   Add #define FLAG_MULTI_PACKET 0x1000 and int (*manage_power)(struct usbnet *, int);add the above to the files, and the adding location can refer to usbnet.h in 2.6.38.

   And then add the following statements:

   extern int usbnet_start_xmit(struct sk_buff *skb,

   struct net_device *net);

2.2.1.4 Linux 2.6.22 Kernel Integration

Please refer to the following steps for integration:

1. Login https://www.kernel.org/pub/linux/kernel/v2.6/, and download the source codes of kernel 2.6.38 and 2.6.26
2. Decompress the downloaded 2.6.26 kernel source code, and copy the following files to the corresponding directory

   drivers/net/usb/usbnet.c

   include/linux/usb/usbnet.h

   Modify usbnet.c according to the following steps

   2.1 Search for key word “print_mac”, and give annotation for the printed statements that include printf_mac

   2.2 Give annotation for DECLARE_MAC_BUF(mac); function call

   2.3 Delete static statement for usbnet_start_xmit function:

   int usbnet_start_xmit (struct sk_buff *skb, struct net_device *net)

   and add the following statements:

   EXPORT_SYMBOL_GPL(usbnet_start_xmit);
3. Decompress the downloaded 2.6.38kernel source code, and copy the following files to the corresponding directory

   drivers/net/usb/cdc_ncm.c

   include/linux/usb/cdc.h

   include/linux/atomic.h
4. Modify the Makefile and Kconfig files under the drivers/net/usb/ directory of source code

   Add obj-$(CONFIG_USB_NET_CDC_NCM) += cdc_ncm.o to the end of Makefile.

   Add the following statements to the end of Kconfig file.

   config USB_NET_CDC_NCM

   tristate "CDC NCM support"

   depends on USB_USBNET

   default y
5. Modify include/linux/usbnet.h

   Add #define FLAG_MULTI_PACKET 0x1000 and int (*manage_power)(struct usbnet *, int);

   add the above to the file and the adding location can refer to usbnet.h in 2.6.38.

   And then add the following statements:

   extern int usbnet_start_xmit(struct sk_buff *skb,

   struct net_device *net);

2.2.1.5 Kernel below Linux 2.6.22

As for kernels below Linux 2.6.22, please refer to section 2.2.1.4 for integration, but there may be compatibility issues when integrating; please contact engineers from Fibocom Wireless Inc. for joint debugging.

2.2.2 NCM Driver Configuration

Step 1: Configuration confirmation

Modify the kernel compiling configuration (config files under the kernel root directory), and ensure that the following configuration items have been selected:

CONFIG_USB_USBNET=y

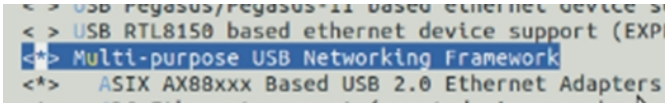
CONFIG_NETDEVICES=y

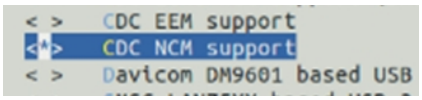
CONFIG_USB_NET_CDC_NCM=y

Step 2: Specific configuration operations

1. Open the Terminal tool, and enter the kernel directory (assumed to be "/home/ght /linux-3.0.8/"), perform make <configuration>command (assume apply the standard make menuconfig).
2. Complete NCM driver configuration according to the following statements:

   Enter **"Device Drivers"→"Network device support"→ "USB Network Adapters"** menu and select Multi-purpose USB Networking Framework and CDC NCM support items:





3. After the configuration, exit the configuration interface step by step by selecting "<Exit>"". And then select "<Yes>" and exit in the save interface.
4. After completing the configuration, run the make command and compile the modified kernel.

2.3 Linux ACM Driver Integration

2.3.1 ACM Driver Porting

1. Modification for driver code: As shown in Figure 2-2, add the codes in the red box to array "static const struct usb_device_idacm_ids[ ]" in file "drivers/usb/class/cdc-acm.c".

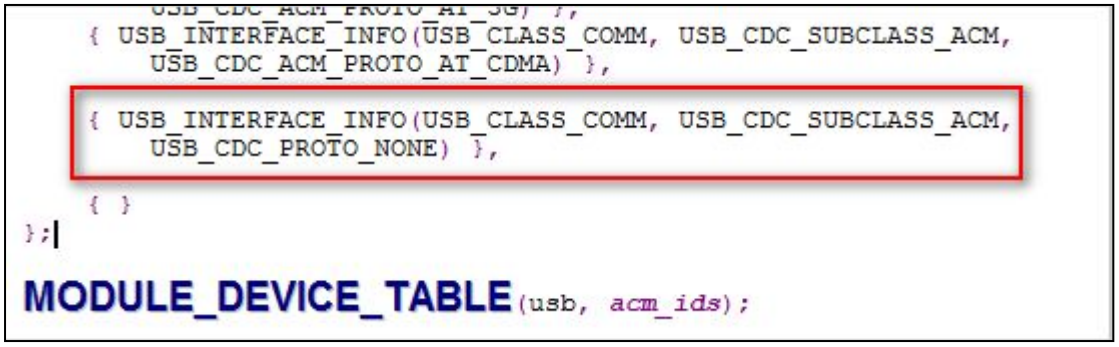


Figure 2-2 Codes of acm_ids

Codes in Figure 2-2 are as follows:

```
{ USB_INTERFACE_INFO(USB_CLASS_COMM, USB_CDC_SUBCLASS_ACM,
    USB_CDC_PROTO_NONE) },
```

2. Modify the compiled configuration of kernel (config files under the kernel directory) and ensure the following configuration items have been selected:

1) Related configuration items of PPP dial-up:

CONFIG_PPP=y

CONFIG_PPP_MULTILINK=y

CONFIG_PPP_FILTER=y

CONFIG_PPP_ASYNC=y

CONFIG_PPP_SYNC_TTY=y

CONFIG_PPP_DEFLATE=y

CONFIG_PPP_BSDCOMP=y

2) Related configuration items of USB ACM:

CONFIG_USB_ANNOUNCE_NEW_DEVICES=y (if such option exists, it's suggested to configure; if not, please ignore)

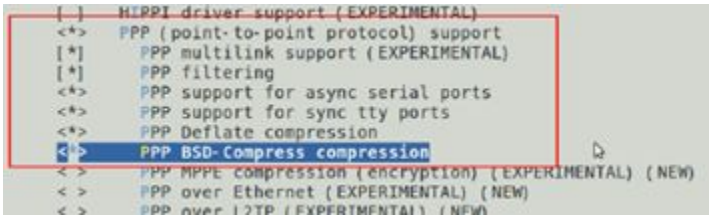
CONFIG_USB_ACM=y

2.3.2 Detailed Configuration Setup

1. Open the Terminal tool, enter the kernel directory (it is assumed to be "/linux-3.0.8/ /home/ght"), and execute the <configuration> make command (it's assumed to use standard menuconfig).

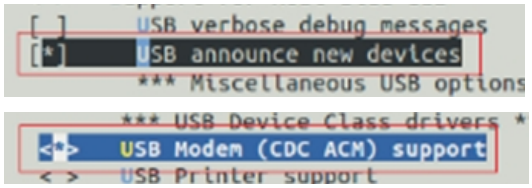
2. Complete configurations of PPP dial-up as the following guidelines:

   Enter **"Device Drivers"→"Network device support"** menu and select all items in the red border



3. Complete configurations of ACM driver as the following guidelines:

   Enter "**Device Drivers"**→"**USB support**" menu and select USB announce new devices and USB Modem (CDC ACM) support items:



4. After the configuration, exit the configuration interface step by step by selecting "<Exit>". And then select "<Yes>" and exit the save interface.

5. After completing configurations, run the make command to edit the modified kernel.

2.4 NCM/ACM Driver Configuration Confirmation

When the system starts up, execute the dmesg command and check the kernel messages. The information as shown in the red box in Figure 2-3 indicate that the NCM driver in the system has been successfully configured.

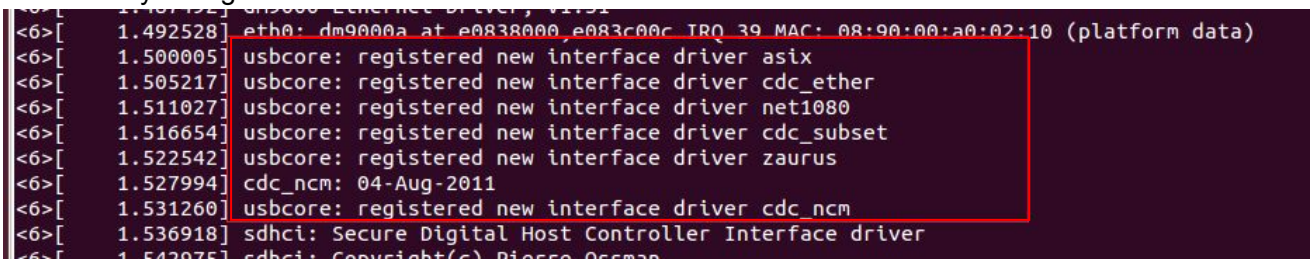


Figure 2-3 NCM Configuration

After the system is started and the 4G module is powered up, execute the dmesg command to check the kernel messages; the information as shown in the red box of in Figure 2-4 indicate that NCM driver has been successfully loaded, NCM network device nodes such as usb0, usb1, usb2 are created. For Linux 3.10 and above kernels, the network device nodes for NCM will be wwan0, wwan1, and wwan2 and so on. When loading drivers, the device nodes can be verified as shown in the red box in Figure 2-4. The NCM device nodes are usb0 ~ usbx for USB type, wwan0 ~ wwanx for wwan type.

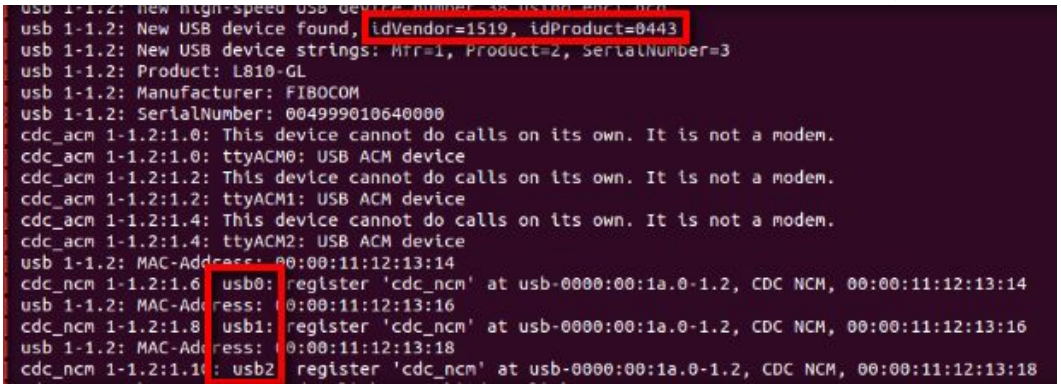


Figure 2-4 Driver Loading

Execute netcfg command to inquire such NCM network device nodes as usb0, usb1, usb2 and so on.

Execute ls –al /dev/ttyACM* command to inquire ttyACM0, ttyACM1 and ttyACM2.

2.5 Port Form Description

No.	Port name	Port form	Remarks
1	ttyACM0	Modem Port	For PPP data traffic, or for sending and receiving AT command under the non-data mode
2	ttyACM1	Trace Port	For capturing module debug information
3	ttyACM2	At Port	For AT communications, namely, sending and receiving AT commands
4	usb0(wwan0)usb1(wwan1) usb2(wwan2)usb3(wwan3)	NCM Port	Standard NCM network device driver for USB, mainly used for transporting network data. There are generally 3 nodes mapped, and the actual number of nodes depends on the actual situation.

2.6 Port Testing

2.6.1 Command Line Testing

1. Open the terminal.
2. Execute echo -e "ate0\r\n"> /dev/ttyACM2 **(**Execute this command before any other command or the cat command will be abnormal.**)**
3. Execute echo -e "at+cgmr\r\n"> /dev/ttyACM2 to inquire the software version.
4. Execute cat /dev/ttyACM2 to read the result.
5. Executer echo -e "at+cgdcont=1,\"ip\",\"3gnet\"\r\n"> /dev/ttyACM2 to set up APN.
6. Execute cat /dev/ttyACM2 to read settings and return result

If AT command contains double quotation marks, ESC"\"should be added, please refer to Step 5 for the format.

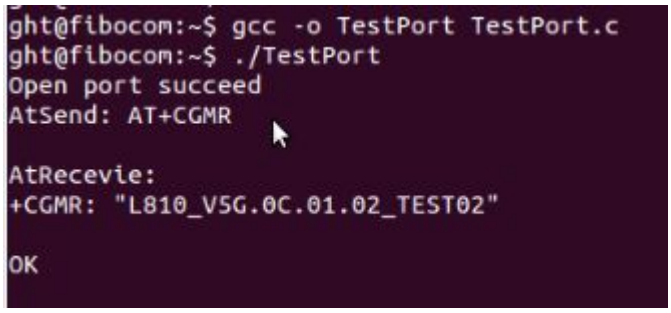
2.6.2 Program Testing

The C program below can be used to test send and receive of AT commands. The program opens the /dev/ttyACM2 device node, and calls the write and read function to send AT commands and receive the reply.

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <termios.h>
#define ATPORT "/dev/ttyACM2"
#define BUFSIZE 1000
#define BAUDRATE B115200
int open_port(char *port)
{
    struct termios options;
    int fd;
    fd = open(port, O_RDWR | O_NOCTTY | O_NDELAY);
    if (fd == -1) {
        printf("%s: Unable to open the port - \r\n",__func__);
    } else {
        fcntl(fd, F_SETFL, FNDELAY);
        tcgetattr(fd, &options );
        cfsetispeed(&options, BAUDRATE );
        cfsetospeed(&options, BAUDRATE );
        options.c_cflag |= ( CLOCAL | CREAD);
        options.c_cflag&= ~(CSIZE | PARENB | CSTOPB | CSIZE);
```

```
            options.c_cflag |= CS8;
            options.c_cflag&= ~CRTSCTS;
            options.c_lflag&= ~(ICANON | ECHO | ECHOE | ISIG);
            options.c_iflag&= ~(IXON | IXOFF | IXANY | ICRNL | INLCR | IGNCR);
            options.c_oflag&= ~OPOST;
            if ( tcsetattr( fd, TCSANOW, &options ) == -1 ) {
                printf ("Error with tcsetattr = %s\r\n", strerror ( errno ) );
            } else {
                printf ( "Open port succeed\r\n");
            }
        }
        return (fd);
}
int main()
{
        int fd = open_port(ATPORT);
        char at_cmd_ch[50]="AT+CGMR\r\n";
        char buf[BUFSIZE];
        memset(buf,0,BUFSIZE);
        printf("AtSend: %s\r\n", at_cmd_ch);
        write(fd, at_cmd_ch , strlen(at_cmd_ch));
        sleep(1);
        read(fd, buf, BUFSIZE );
        printf("AtRecevie: %s\r\n", buf);
        close(fd);
        return 0;
}
```

Save the above code in the TestPort.c text, execute the o - TestPortTestPort.c GCC command to compile the TestPort program, and then execute the compiled program to see the returned results.



Because the 4G module needs time to process after sending out the "AT+CGMR" command, it is necessary to delay at least 500ms before reading. Sleep (1) in the demo code is only for reference.

Attention:

sleep(1) means delaying 1 second.

2.7 Connect Internet via NCM by AT Commands

2.7.1 Query Signal, SIM Card and Network State

The range of returned parameter 1 of AT+CSQ is 0-31 or 99; the value 99 means no signal, and it's necessary to check antenna in this case.

AT+CSQ

+CSQ: 23,99

OK

AT+CPIN? Check SIM card state, and return READY, which means SIM card is available; if SIM PIN is returned, please use AT+CPIN="correct PIN" to decode PIN.

AT+CPIN?

+CPIN: READY

OK

AT+CPIN?

+CPIN: SIM PIN

OK

AT+CPIN="<correct PIN>"

OK

AT+COPS? Inquire operator selection and network state; if there is only one returned parameter, please check whether antenna and SIM card are normal.

```
AT+COPS?
+COPS: 0,0,"CHINA MOBILE",7
OK
```

Parameter 1 of the returned value means the registration mode; 0 represents the automatic operation, and 1 represents the manual operation;

Parameter 2 of the returned value indicates the display format; 0 represents the long character string format, 1 represents the short character string format, and 2 represents the number of words;

Parameter 3 of the returned value means that the operator's name can be displayed based on the parameter 2, and CHINA MOBILE is China Mobile;

Parameter 4 of the returned value means the network state, 7 represents LTE network, 2 represents UMTS, 0 represents GSM;

2.7.2 Dial-up AT Commands

In the case that RF signal, SIM card status and operator registry network are normal, it's necessary to send a dial-up AT command (please select ttyACM2 port for AT command). To register 2G or 3G network, please start from the step 3; to register 4G network, please follow the steps

1) AT+CGDCONT? Check if the 4th parameter (IP address) of the returned value is empty, if yes, execute in sequence; if there is an IP address, jump to Step 5 directly.

2) "AT+CGDCONT=1", "IP", "cmnet" defines the PDP context, the parameter 3 means the APN type, the China Unicom SIM card should be set to 3gnet.

3) AT+XDNS=1,1 using DNS address to inquire command.

4) AT+MGAUTH=1,1, "cmnet", "1234" set user name and password, parameter 2 means authentication type, 1 represents PAP, 2 represents CHAP, 0 represents NONE, and parameters 3 and 4 represent user name and password respectively.

5) AT+CGACT=1,1 activate the PDP context

6) "AT+XDATACHANNEL=1,1", "/USBCDC/2", "/USBHS/NCM/0", 0 configure data channels, parameter 3 represents AT channel, and parameter 4 represents data channel.

7) AT+CGDATA= "M-RAW_IP", 1 activate data state; after the command is sent successfully, the module triggers the host to initiate the DHCP process. Now the dial-up commands are sent.

Attention:

The return of the AT+CGACT command takes longer time; in case that there is no need to set up user name and password, please skip Step 4.

2.7.3 USBCDC and ACM Corresponding Table

No.	Port	Descriptor	Remarks
1	ttyACM0	/USBCDC/0	This table lists the correspondence between parameter 3 and parameter 4 of XDATACHANNEL in Step 6 of Section 2.7.2 and driver device node; if the ttyACM0 is selected to issue all AT Commands under Section 2.7.2, parameter 3 in the Step 6 shall be modified to /USBCDC/0
2	ttyACM1	/USBCDC/1	
3	ttyACM2	/USBCDC/2	
4	usb0(wwan0)	/USBHS/NCM/0	
5	usb1(wwan1)	/USBHS/NCM/1	
6	usb2(wwan2)	/USBHS/NCM/2	

2.7.4 IP and DNS Address Query

AT+CGPADDR=1 parameter 2 will return ipv4 address; returning 0.0.0.0 represents the failure of dialing in Step2, and it's necessary to restart Step 2.

AT+CGPADDR=1

+CGPADDR: 1,"10.47.67.169"

OK

AT+XDNS? parameter 2 and parameter 3 are primary DNS address and secondary DNS address respectively.

AT+XDNS?

+XDNS:1,"221,179,38,7","120.196.165.7"

OK

2.8 NCM Network Configuration

2.8.1 DNS Address Configuration

Red font is the DNS address acquired based on AT+XDNS?, which shall be written into /etc/resolv.conf file according to the actual results acquired, and the format is as follows:

nameserver 221.179.38.7

nameserver 120.196.165.7

2.8.2 IP Address and Route Configuration

Set the host IP and route according to the IP address inquired by sending AT+CGDCONT?, where $1 is the IP address, and $2 is the gateway address (gateway address can be the same as the IP address, and the last byte of the IP address can also be set to 1).

ifconfig usb0 $1 netmask 255.255.255.255 -arp

ip r add $2 dev usb0

ip r add 0.0.0.0/0 via $2 dev usb0

Attention:

If the NCM network device node name is wwan, please replace the red usb0 bywwan0

After setting up, please ping www.google.com to check whether the host network is able to work.

2.9 Connect Internet via PPP Dial-up

In application scenarios where NCM driver can't be supported, it's necessary to use PPP dial-up.

There are a total of three script files for PPP dial-up: chat-wcdma-connect, chat-wcdma-disconnect and wcdma, and the content for scripts is as shown in 2.10.

1. Put the above three script files in the /etc/ppp/peers/ directory, and use Chmod 777 XXX command to give the file read and execute permissions. Input the following in the command line:

   PPPD call <dial-up script >

   For example, if file name of the dial-up script is "wcdma", the command is as follows:

   pppd call wcdma

2. After successful dial-up, execute the ifconfig command to inquire IP address.

Figure 2-5 shows the query results after executing ifconfig command after successful ppp dial-up.

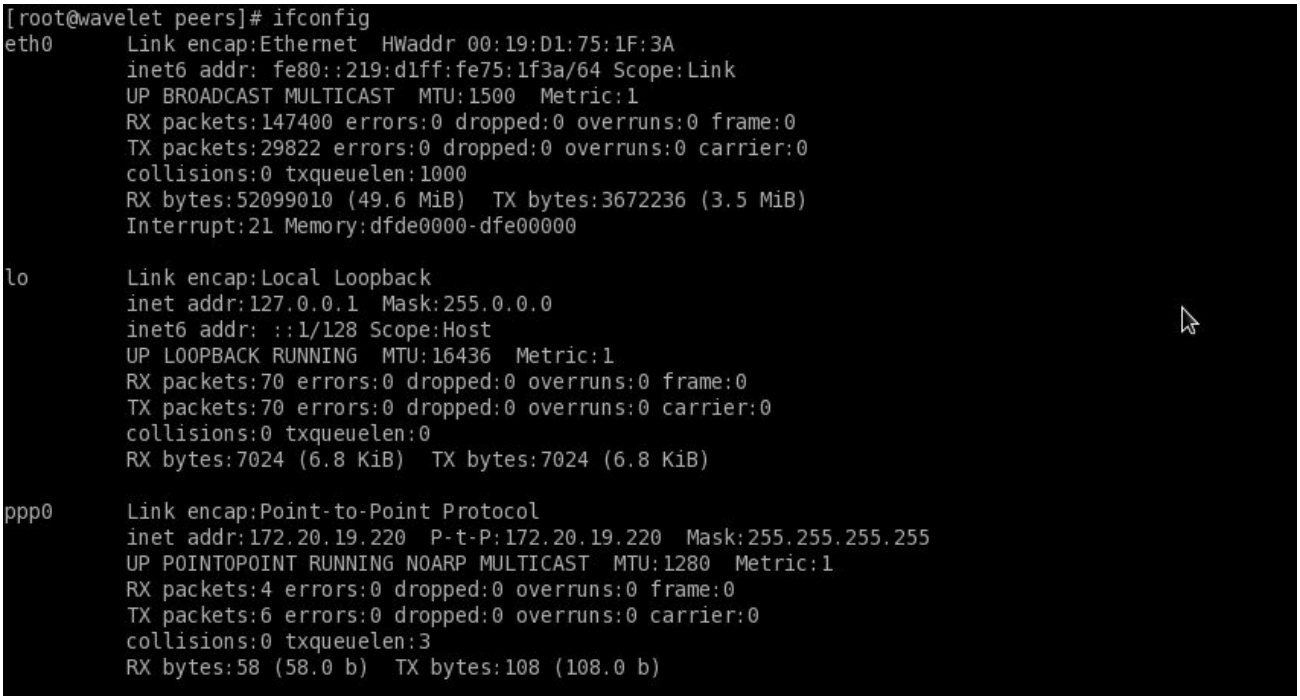


Figure 2-5 ifconfig query

Attention:

Precondition for dial-up:

- A valid SIM card is inserted.
- Module is powered up and running.
- Module can register network.

2.10 PPP Dial-up Script Description

Example of wcdma script:

```
nodetach
lock
/dev/ttyACM0
115200
crtscts
debug
#logfile /data/logfile
modem
hide-password
usepeerdns
noauth
noipdefault
novj
novjccomp
noccp
defaultroute
ipcp-accept-local
ipcp-accept-remote
connect 'chat -s -v -f /etc/ppp/peers/chat-wcdma-connect'
disconnect 'chat -s -v -f /etc/ppp/peers/chat-wcdma-disconnect'
```

Attention:

/dev/ttyACM0 assigns the port for dial-up; if it's necessary to use ACM2 port for dial-up, just modify ttyACM0 into ttyACM2.

Example of chat-wcdma-connect script:

```
" AT
OK "
ABORT 'NO CARRIER'
ABORT 'ERROR'
ABORT 'NO DIALTONE'
ABORT 'BUSY'
ABORT 'NO ANSWER'
" AT
OK ATZ
OK AT+GTRAT?
OK AT+CMEE=2
OK AT+CSQ
OK AT+CPIN?
OK AT+COPS?
OK AT+CGACT=0,1
OK AT+CGDCONT=1,\"IP\",\"cmnet\",,0,0
OK ATDT*99#
CONNECT "
```

Attention:

AT+CGDCONT=1,\"IP\",\"cmnet\",,0,0 (cmnet represents APN for China Mobile and APN for China Unicom is 3gnet)

Example of chat-wcdma-disconnect script:

```
ABORT   OK
ABORT   BUSY
ABORT   DELAYED
ABORT   "NO ANSWER"
ABORT   "NO CARRIER"
ABORT   "NO DIALTONE"
ABORT   VOICE
ABORT   ERROR
ABORT   RINGING
TIMEOUT   12
""   \K
""   \K
""   \K
""   +++ATH
""   +++ATH
""   +++ATH
""   ATZ
SAY "\nGoodbay\n"
```

3 Android System Instructions

3.1 Android Kernel Driver Porting and Loading

Android kernel is based on the Linux kernel; as for the configuration of Android kernel, please refer to sections 2.2, 2.3 and 2.4.

3.2 System Integration and Debugging

3.2.1 Communication between System and Module Ports

The communication function of Android system is realized by the interaction of AT commands between RIL and the module. The system shall have the hardware interface for data communication, such as USB and UART. If the system uses the USB port to send and receive AT command, the corresponding USB driver shall be loaded in the Android system kernel; please refer to Section 3.1 for detailed upload methods.

RIL (Interface Layer Radio) abstracts the various functions used by users to corresponding requests and unsolicited commands, which are eventually converted into 3GPP standard AT commands and sent to the 4G module, and the processing results will be returned to UI.

3.2.2 RIL Integration

1. Modify ril-daemon service in init.rc files:

```
#begin
serviceril-daemon /system/bin/rild -l /system/lib/libreference-ril.so -- -d    /dev/ttyACM2
class main
socketrild stream 660 root radio
socketrild-debug stream 660 radio system
user root
group radio cache inetmisc audio sdcard_rw log
#end
```

The parameter after –d is the actual port mapped by USB (AT communication port). In general, ttyACM2 is the AT communication port for 4G module.

2. Modify android/hardware/ril/rild/rild.c file, and mark the switchUser() function call.

```
OpenLib:
#endif
//    switchUser();

    dlHandle = dlopen(rilLibPath, RTLD_NOW);
```

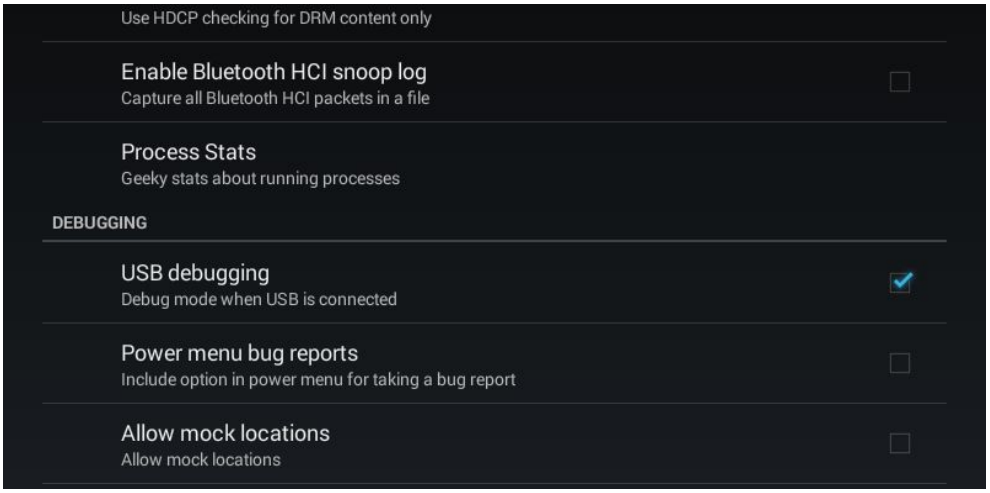
During compiling, pack libreference-ril.so provided by Fibocom into the system mirror image directory /system/lib/, and then burn the new system image.

3.2.3 adb Tool Installation

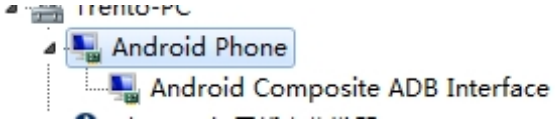
To see if the RIL is normally loaded, it's necessary to see RIL operation LOG using adb tool.

1. adb tool installation and instructions

After starting up, check "developer option"->>"USB debugging" option in the Android system settings to use adb debugging features.



Connect Android device to PC, and install adb driver (which can be installed via the commonly used Android mobile assistant tool). After successful installation, an adb device is found in the device management.



Open a cmd window in the windows system, enter the directory of adb tool and input "adb shell" command to enter the command line terminal of Android equipment for all kinds of debugging work.

3.2.4 RIL Library Replacing

In the debugging process, it's necessary to update the RIL library from time to time, and the following steps can be performed in order to update the RIL library.

1. adb devices --- inquire whether the device is identified by adb tool

```
C:\Users\Administrator>adb devices
List of devices attached
0123456789ABCDEF        device
```

2. adb root --- switch to root for adb terminal

```
C:\Users\Administrator>adb root
adbd is already running as root
```

3. adb remount --- give /system/directory read-write permission

```
C:\Users\Administrator>adb remount
remount succeeded
```

4. adb shell ls init*.rc --- inquire how many init.xx.rc files are in android system

```
C:\Users\Administrator>adb shell ls init*.rc
```

```
E:\intel-update\Intel_update>adb shell ls init*.rc
init.am335xevm.rc
init.am335xevm.usb.rc
init.goldfish.rc
init.rc
init.trace.rc
init.usb.rc
```

5. adb shell cat init.xx.rc – inquire the init.xx.rc file one by one, until the /system/bin/rild keyword is found; use the key word rild to search the RIL library name in use: lib reference-ril.so (Intel platform rild keyword is generally in the init.modem.rc file)

```
service ril-daemon /system/bin/rild -l /system/lib/libreference-ril.so -- -d /dev/ttyACM2
    class main
    socket rild stream 660 root radio
    socket rild-debug stream 660 radio system
    user root
    group radio cache inet misc audio sdcard_rw log
```

6. adb shell push xxx\libght-ril.so /system/lib/libreference-ril.so --- replace ril library

```
C:\Users\Administrator>adb push E:\libreference-ril.so /system/lib/libreference-ril.so
```

Attention:

The first parameter of push is the ril library path in the local directory of the computer, and the second parameter is the ril library path used in android system. The ril library name given from the second parameter must be the same as the ril library name inquired in Step 5, or ril can't run normally.

7. adb shell stop ril-daemon ---suspend ril service
8. adb shell logcat –b radio –c --- clear up ril log
9. adb shell start ril-daemon --- start up ril service
10. adb shell logcat –b radio –v time > D:/radio.txt --- capture ril log and confirm version number

Use ctrl+c to terminate log capturing, and open radio.txt files under the root directory in D:\, then search for key words RIL Daemon version to inquire the RIL version number and confirm whether RIL library is updated successfully.

```
C:\Users\Administrator>adb shell stop ril-daemon

C:\Users\Administrator>adb shell logcat -b radio -c

C:\Users\Administrator>adb shell start ril-daemon

C:\Users\Administrator>adb shell logcat -b radio -v time > D:/radio.txt
^C
C:\Users\Administrator>
```

3.2.4.1 Check for RIL Driver Loading

Open adb shell and input logcat - b radio; check the AT command interaction through the output log, and the following figure is an example of log with normal RIL initialization:

```
D/AT      ( 1484): AT> ATE0Q0V1
D/AT      ( 1484): AT< ATE0Q0V1
D/AT      ( 1484): AT< OK
D/AT      ( 1484): AT> ATE0Q0V1
D/AT      ( 1484): AT< OK
D/AT      ( 1484): AT> ATS0=0
D/AT      ( 1484): AT< OK
D/AT      ( 1484): AT> AT+CMEE=1
D/AT      ( 1484): AT< OK
D/AT      ( 1484): AT> AT+CREG=2
D/AT      ( 1484): AT< OK
D/AT      ( 1484): AT> AT+CGREG=1
D/AT      ( 1484): AT< OK
D/AT      ( 1484): AT> AT+CCWA=1
D/AT      ( 1484): AT< OK
D/AT      ( 1484): AT> AT+CMOD=0
```

3.2.4.2 Check for RIL Version Number

Open adb shell and input logcat –b radio; check RIL version number through the output log, and search for the key word “RIL Daemon version“, of which RIL_V4x.00.02 is the version number of RIL.

```
I/RILJ    (  505): Connected to 'rild' socket
I/RILC    ( 1264): libril: new connection
I/RILC    ( 1264): RIL Daemon version: RIL_V4x.00.02
D/RILJ    (  505): [UNSL]< UNSOL_RIL_CONNECTED (7)
D/RILJ    (  505): [0001]> RADIO_POWER off
D/RIL     ( 1264): onRequest: RADIO_POWER
D/RIL     ( 1264): mly1:onRequest: RADIO_POWER
```

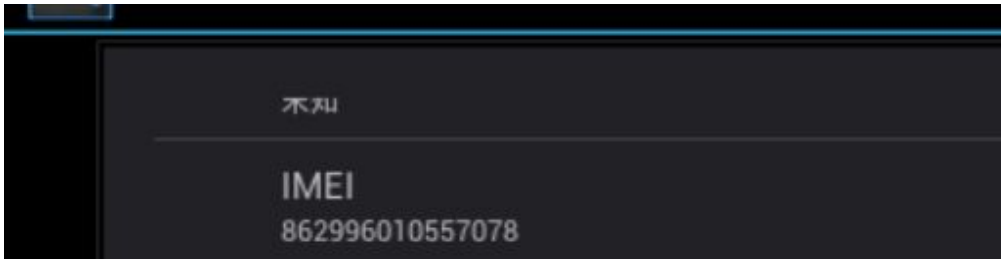
3.2.5 Debug for Signal Quality, Telephone, SMS

3.2.5.1 Query for Module IMEI

After the completion of the initialization, the upper-level application will query the IMEI number of the module, and the following is the normal query results of IMEI number in RIL log.

```
D/RIL     ( 1264): onRequest: GET_IMEI
D/RIL     ( 1264): mly1:onRequest: GET_IMEI
D/RIL     ( 1264): mly2:onRequest: 38
D/RIL     ( 1264): mly3:sState: 4
D/AT      ( 1264): AT> AT+CGSN?
```

Enter the "Settings" - > "panel information" - > "state information" to query IMEI. If it's "Unkown", the interaction between the RIL and the module is abnormal.



3.2.5.2 Signal Strength Query

After the successful operation of RIL, the upper level can query the signal strength regularly, and the following is the normal query results of signal intensity in log RIL:

```
D/RIL     ( 1505): onRequest: SIGNAL_STRENGTH
D/RIL     ( 1505): mly1:onRequest: SIGNAL_STRENGTH
D/RIL     ( 1505): mly2:onRequest: 19
D/RIL     ( 1505): mly3:sState: 2
D/AT      ( 1505): AT> AT+CSQ
D/AT      ( 1505): AT< +CSQ: 13,99
D/AT      ( 1505): AT< OK
D/RILJ    (  505): [0097]< SIGNAL_STRENGTH SignalStrength: 13
```

Signal strength displayed on the interface:

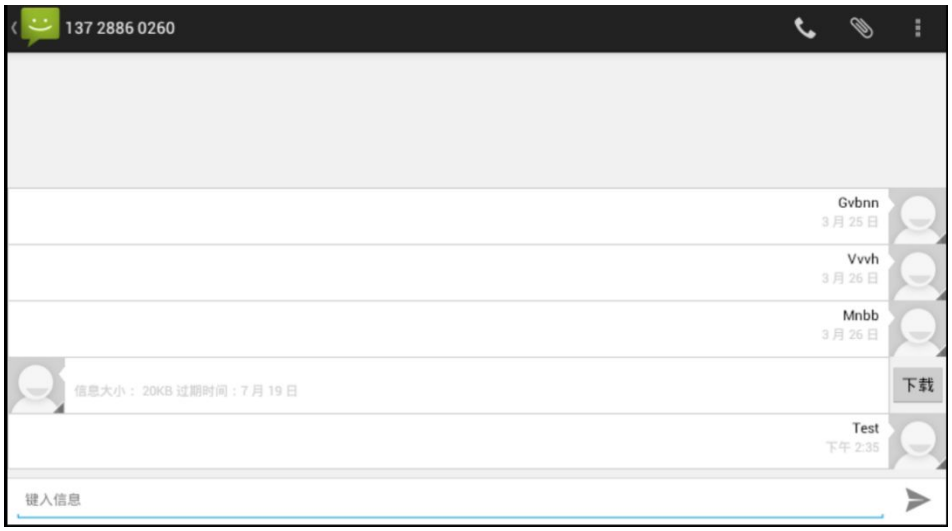
3.2.5.3 Voice and SMS Detection

If the above debug and test results are normal, then we can proceed the voice call as well as sending and receiving SMS. Check whether the corresponding commands are sent via adb by logcat –b radio command during corresponding operations.

Dial 10086, and RIL information are displayed as follows:

```
D/use-Rlog/RLOG-AT(  227): AT> AT+CMUT=0
D/use-Rlog/RLOG-AT(  227): AT< OK
D/use-Rlog/RLOG-RIL(  227): onRequest: DIAL
D/use-Rlog/RLOG-AT(  227): AT> ATD10086;
D/RILJ    (  797): [4989]< SET_MUTE
D/use-Rlog/RLOG-AT(  227): AT< +STKCTRLIND: 0,0,,"10086",129
D/use-Rlog/RLOG-AT(  227): AT< OK
D/use-Rlog/RLOG-AT(  227): AT< +CLCC: 1,0,2,0,0,"10086",129
```

Send "test" to the other side through SMS:

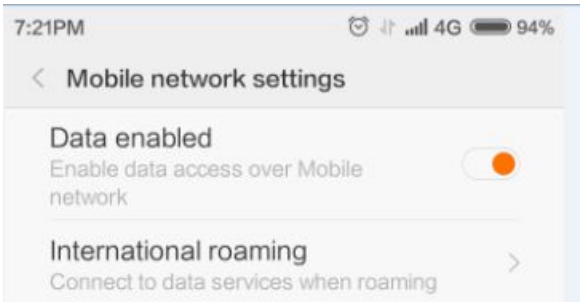


Corresponding log of RIL:

```
D/RILJ    ( 2033): [0058]> SEND_SMS
D/RIL     ( 1484): onRequest: SEND_SMS
D/AT      ( 1484): AT> AT+CMGS=17
D/AT      ( 1484): AT< >
D/AT      ( 1484): AT> 0001000b813127880662f0000004d4f29c0e^Z
D/AT      ( 1484): AT< +CMGS: 24
D/AT      ( 1484): AT< OK
D/RILJ    ( 2033): [0058]< SEND_SMS { messageRef = 0, errorCode
11}
```

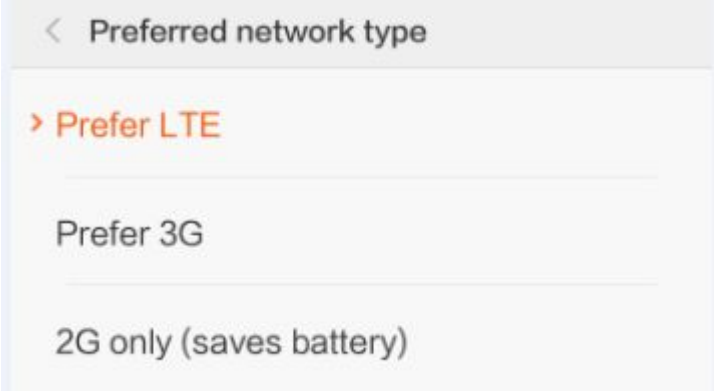
3.2.5.4 Enabling Data Network

Enter “settings”->“mobile network”->“enable data network”, and click to enable, the upper left corner of the signal bar will display the icon of the current network type:



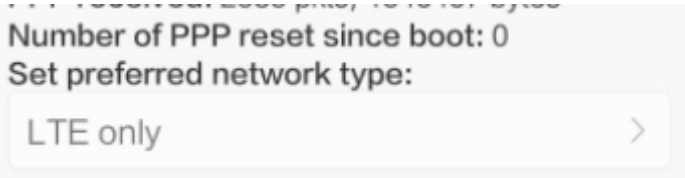
3.2.5.5 Preferred Network

Enter “settings”->“mobile network”->“preferred network”, and click the corresponding menu to switch to the corresponding network



If there is no “Prefer 4G” in the menu, the customer needs to modify the UI layer to add the option.

If there is no “Prefer 4G” in the menu, customers can enter the engineering mode for network switching: Open the Phone app - > input *#*#4636#*#* - >select “Tablet information "- >pull down “Set preferred network type” menu, and select LTE Only to switch to 4G network. This method is only used for testing, and the final solution is still adding the “Prefer 4G” menu in the UI layer.



3.2.5.6 Debug Audio Channel Switch and Volume Adjustment

Android engineers add contents to telephone service layer, HAL layer or the driver layer for corresponding platforms to implement audio channel switch and volume adjustment functions by AT commands.

4 Win8.1/Win10 System Instructions

The 4G module supports Win8.1/Win10 and the applications on win8.1/win10 system are similar. The article below, we use Win10 as the example to illustrate the operation.

4.1 Introduction

4G module in the Win10 system will map the MBIM (Mobile Broadband Interface Model) port. Accompanied with the mobile broadband, MIBM is widely used for mobile devices such as notebook/Ultrabook, Tablet, Pad. The interface standard is presented by Intel, Microsoft and the other USB/IF members. Specific standard can be downloaded at the official website of USB IF. It has unified the interface standards of mobile broadband devices (USB data card/CDMA, NGFF data card, etc.) with the PC terminal.

Attention:

Modem manufacturers don't have to provide driver, which is the in-box driver of Win8.1 and Win10.

4.2 Application Note

4.2.1 MBIM Driver Upload

MBIM driver comes with Win8.1 and Win10 system; it can be automatically identified when 4G module is inserted. Check the network adapter layer of the device manager; if Fibocom xxxx is displayed, the device is loaded successfully. The example is as follows:

4.2.2 Application Procedures

4.2.2.1 Data Network

Click the icon in the lower right corner to display all of the mobile network connections, click:

Enter the mobile network setting menu, as shown in figure 4-1:

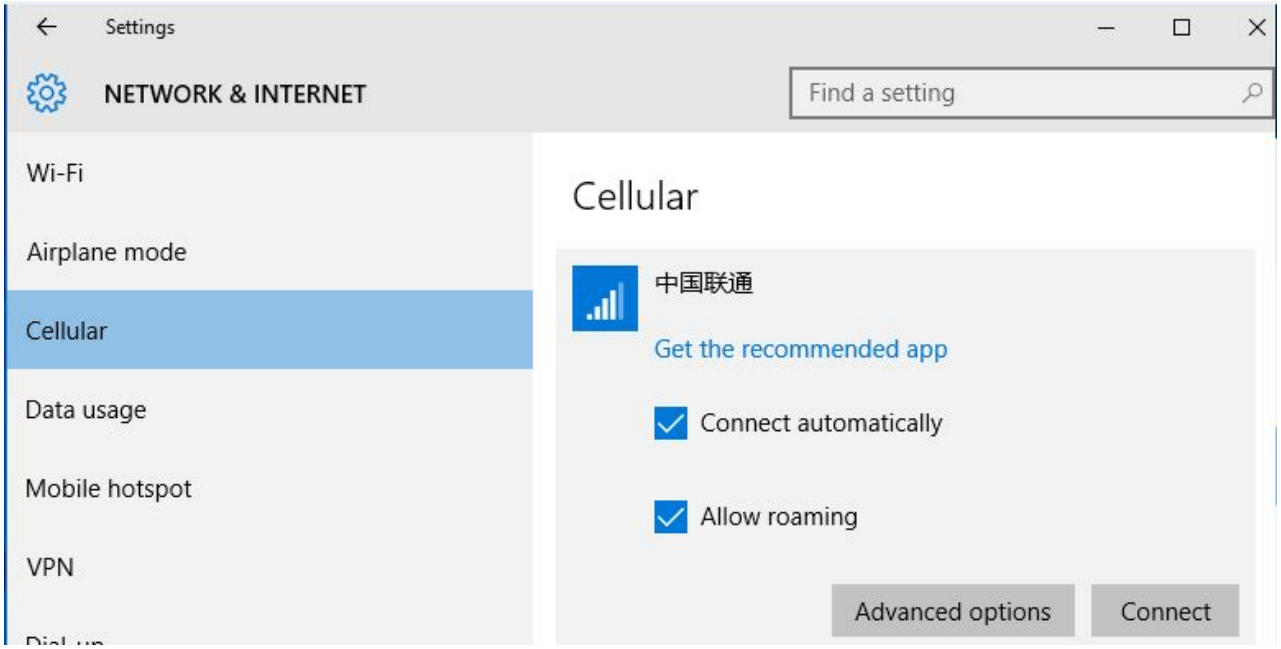


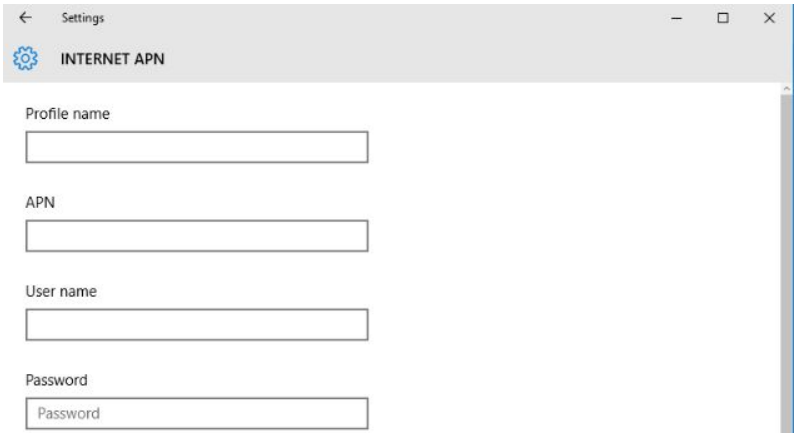
Figure 4-1 Mobile Network Menu

Click the Connect button to connect the data; after the successful connection, Connected will be shown, and then you can browse the web.

4.2.2.2 APN Settings

Click: Advanced options; to find the APN Internet option, click Add an Internet APN,

Fill in Profile Name and APN, and click Save. Fill in User Name and Password



4.2.2.3 SIM PIN Settings

Click: Advanced options; to find the Use SIM PIN option, input pin code in the pop-up menu, and click OK.