



# **SIM82XX\_SIM83XX Series \_Linux USB\_Application Note**

**5G Module**

**Shanghai SIMCom Wireless Solutions Ltd.**

SIMCom Headquarters Building, Building 3, No. 289  
Linhong Road, Changning District, Shanghai P.R. China  
Tel:86-21-31575100/31575200  
support@simcom.com  
www.simcom.com

<b>Document Title:</b>	SIM82XX_SIM83XX Series_Linux USB_Application Note
<b>Version:</b>	1.01
<b>Date:</b>	2021.11.25
<b>Status:</b>	Released

## GENERAL NOTES

SIMCOM OFFERS THIS INFORMATION AS A SERVICE TO ITS CUSTOMERS, TO SUPPORT APPLICATION AND ENGINEERING EFFORTS THAT USE THE PRODUCTS DESIGNED BY SIMCOM. THE INFORMATION PROVIDED IS BASED UPON REQUIREMENTS SPECIFICALLY PROVIDED TO SIMCOM BY THE CUSTOMERS. SIMCOM HAS NOT UNDERTAKEN ANY INDEPENDENT SEARCH FOR ADDITIONAL RELEVANT INFORMATION, INCLUDING ANY INFORMATION THAT MAY BE IN THE CUSTOMER'S POSSESSION. FURTHERMORE, SYSTEM VALIDATION OF THIS PRODUCT DESIGNED BY SIMCOM WITHIN A LARGER ELECTRONIC SYSTEM REMAINS THE RESPONSIBILITY OF THE CUSTOMER OR THE CUSTOMER'S SYSTEM INTEGRATOR. ALL SPECIFICATIONS SUPPLIED HEREIN ARE SUBJECT TO CHANGE.

## COPYRIGHT

THIS DOCUMENT CONTAINS PROPRIETARY TECHNICAL INFORMATION WHICH IS THE PROPERTY OF SIMCOM WIRELESS SOLUTIONS LIMITED. COPYING, TO OTHERS AND USING THIS DOCUMENT, ARE FORBIDDEN WITHOUT EXPRESS AUTHORITY BY SIMCOM. OFFENDERS ARE LIABLE TO THE PAYMENT OF INDEMNIFICATIONS. ALL RIGHTS RESERVED BY SIMCOM IN THE PROPRIETARY TECHNICAL INFORMATION, INCLUDING BUT NOT LIMITED TO REGISTRATION GRANTING OF A PATENT, A UTILITY MODEL OR DESIGN. ALL SPECIFICATION SUPPLIED HEREIN ARE SUBJECT TO CHANGE WITHOUT NOTICE AT ANY TIME.

### **SIMCom Wireless Solutions Limited**

SIMCom Headquarters Building, Building 3, No. 289 Linhong Road, Changning District, Shanghai P.R. China

Tel: +86 21 31575100

Email: [simcom@simcom.com](mailto:simcom@simcom.com)

### **For more information, please visit:**

<https://www.simcom.com/download/list-863-en.html>

### **For technical support, or to report documentation errors, please visit:**

<https://www.simcom.com/ask/> or email to: [support@simcom.com](mailto:support@simcom.com)

**Copyright © 2021 SIMCom Wireless Solutions Limited All Rights Reserved.**

# About Document

## Version History

Version	Date	Author	What is new
V1.00	2020.8.17	Xiaowei.Li	New version
V1.01	2021.11.25	Haotian.Wang	Update the format

## Scope

This document applies to the SIMCom SIM820X series, SIM821X series, SIM826X series and SIM83XX series.

# Contents

<b>About Document.....</b>	<b>2</b>
Version History.....	2
Scope.....	2
<b>Contents.....</b>	<b>3</b>
<b>1 Introduction.....</b>	<b>4</b>
1.1 Purpose of the document.....	4
1.2 Related Documents.....	4
1.3 Conventions and abbreviations.....	4
<b>2 Device Driver Installation.....</b>	<b>5</b>
2.1 USB Serial Option Driver.....	5
2.1.1 Add VID and PID.....	5
2.1.2 Compilation Configuration for USB Serial Option Driver.....	6
2.1.3 Building a Linux Driver Module.....	6
2.2 QMI WWAN Drivers.....	7
2.2.1 Source Code File Modification.....	7
2.2.2 Compilation Configuration for QMI WWAN Driver.....	8
2.2.3 Building a Linux Driver Module.....	8
2.3 Kernel Compilation Configuration.....	9
2.3.1 Compilation Configuration for PPP Driver.....	9
<b>3 Modem Usage.....</b>	<b>10</b>
3.1 Test AT Commands.....	10
3.2 Use PPP Data connection.....	10
3.2.1 How Does a PPP Dial-Up Connection Work?.....	10
3.3 Use NDIS Data connection.....	15
<b>4 Troubleshooting.....</b>	<b>17</b>
4.1 How to check whether the correct USB serial driver exists in the kernel?.....	17
4.2 What can I do if the port number does not start from ttyUSB0?.....	17
4.3 How to check whether the correct QMI WWAN driver integration in the kernel?.....	18
4.4 How do I provide required logs if the corresponding ttyUSB ports are not found?.....	19

# 1 Introduction

## 1.1 Purpose of the document

This user guide serves the following purpose:

- Short introductions how to customize the USB driver in Linux OS
- Describes how software developers can use Linux devices for typical use cases.

## 1.2 Related Documents

[1] SIM82XX\_SIM83XX Series\_AT Command Manual

## 1.3 Conventions and abbreviations

Abbreviation	Description
USB	Universal Serial Bus
VID	Vendor ID
PID	Product ID
PPP	Point-to-Point Protocol
IPCP	IP Control Protocol
IP	Internet Protocol
NMEA	National Marine Electronics Association
DNS	Domain Name Server
NDIS	Network Driver Interface Specification
RNDIS	Remote Network Driver Interface Specification

Table 1: Terms and Abbreviations

## 2 Device Driver Installation

In order to recognize the modem, you must add VID and PID in Linux driver option, after the operating system recognizes the modem, devices named /dev/ttyUSBx are created, for example:

- dev/ttyUSB0 diag port for output developing messages
- dev/ttyUSB1 NMEA port for GPS NMEA data output
- dev/ttyUSB2 AT port for AT commands
- dev/ttyUSB3 Modem port for ppp-dial
- dev/ttyUSB4 audio port

### 2.1 USB Serial Option Driver

#### 2.1.1 Add VID and PID

In order to recognize the module, customers should add module VID and PID information as below:

Identify the kernel version in use, retrieve the source code and check in

[linux-src]/drivers/usb/serial/option.c

If one of the following statements exists, you can skip this section(2.1.1) and go directly to the next section.

```
{ USB_DEVICE(ALINK_VENDOR_ID, SIMCOM_PRODUCT_SIM7100E),  
  .driver_info = RSVD(5) | RSVD(6) },  
  
{ USB_DEVICE(ALINK_VENDOR_ID, SIMCOM_PRODUCT_SIM7100E),  
  .driver_info = (kernel_ulong_t)&simcom_sim7100e_blacklist },
```

Add the following statements to the ID list of **static const struct usb\_device\_id option\_ids[]** in the **option.c** :

File: [linux-src]/drivers/usb/serial/option.c

```
static const struct usb_device_id option_ids[] = {  
#if 1 //Added by Simcom
```

```
{ USB_DEVICE(0x1e0e, 0x9001) },  
#endif
```

Add the following statements to the function of **static int option\_probe** in the **option.c** :

File: [linux-src]/drivers/usb/serial/option.c

```
static int option_probe(struct usb_serial *serial,  
                        const struct usb_device_id *id)  
{  
.....  
#if 1 //Added by Simcom`  
if (serial->dev->descriptor.idVendor == cpu_to_le16(0x1E0E) && serial->dev->descriptor.idProduct ==  
cpu_to_le16(0x9001)  
    && serial->interface->cur_altsetting->desc.bInterfaceNumber >=5)  
    return -ENODEV;  
#endif  
.....
```

### 2.1.2 Compilation Configuration for USB Serial Option Driver

Configuration	Configuration(N/Y)
CONFIG_USB_SERIAL	Y
CONFIG_USB_SERIAL_WWAN	Y
CONFIG_USB_SERIAL_OPTION	Y

### 2.1.3 Building a Linux Driver Module

How to compile and install a kernel module in Linux. follow the steps below will guide you along in compiling and install your USB Serial Option driver On Ubuntu operating system.

Step 1: Enter to kernel directory.

```
cd <your kernel directory>
```

Step 2: Build the driver.

```
sudo make -C /lib/modules/`uname -r`/build M=`pwd`/drivers/usb/serial obj-m=option.o modules
```

Step 3: Load the driver and reboot.

```
sudo cp drivers/usb/serial/option.ko /lib/modules/`uname -r`/kernel/drivers/usb/serial
sudo depmod
sudo reboot
```

## 2.2 QMI WWAN Drivers

### 2.2.1 Source Code File Modification

If the following statements are contained in the qmi\_wwan.c file under [linux-src]/drivers/net/usb/, remove them, as they will conflict with SIM8200's QMI WWAN driver.

#### NOTE

```
{QMI_QUIRK_SET_DTR(0x1e0e, 0x9001, 5)}
{QMI_FIXED_INTF(0x1e0e, 0x9001, 5)},
```

Simcom provides the source file qmi\_wwan\_simcom.c, which only be used for Simcom Module. Contact customer engineer of simcom to get the file.

Please modify the Makefile file under [linux-src]/drivers/net/usb/ and the lines in red are the newly added lines, like this:

```
obj-$(CONFIG_USB_USBNET) += usbnet.o qmi_wwan_simcom.o
```



## 2.2.2 Compilation Configuration for QMI WWAN Driver

Configuration	Configuration(N/Y)
CONFIG_USB_SERIAL	Y
CONFIG_USB_SERIAL_WWAN	Y
CONFIG_USB_SERIAL_OPTION	Y
CONFIG_USB_USBNET	Y
CONFIG_USB_WDM	Y

## 2.2.3 Building a Linux Driver Module

How to compile and install a kernel module in Linux, follow the steps below will guide you along in compiling and install your QMI WWAN driver On Ubuntu operating system.

Step 1: Enter to kernel directory.

```
cd <your kernel directory>
```

Step 2: Build the driver.

```
sudo make -C /lib/modules/`uname -r`/build M=`pwd`/drivers/net/usb  
obj-m=qmi_wwan_simcom.o modules  
sudo make -C /lib/modules/`uname -r`/build M=`pwd`/drivers/net/usb obj-m=qmi_wwan.o  
modules
```

Step 3: Load the driver and reboot.

```
sudo cp drivers/net/usb/qmi_wwan_simcom.ko /lib/modules/`uname  
-r`/kernel/drivers/net/usb  
sudo cp drivers/net/usb/qmi_wwan.ko /lib/modules/`uname -r`/kernel/drivers/net/usb  
sudo depmod
```

```
sudo reboot
```

## 2.3 Kernel Compilation Configuration

### 2.3.1 Compilation Configuration for PPP Driver

Configuration	Configuration(N/Y)
CONFIG_USB_SERIAL	Y
CONFIG_USB_SERIAL_WWAN	Y
CONFIG_USB_SERIAL_OPTION	Y
CONFIG_PPP	Y
CONFIG_PPP_FILTER	Y
CONFIG_PPP_MULTILINK	Y
CONFIG_PPP_BSDCOMP	Y
CONFIG_PPP_ASYNC	Y
CONFIG_PPP_SYNC_TTY	Y
CONFIG_PPP_DEFLATE	Y

## 3 Modem Usage

This chapter mainly introduces several commonly used HSUSB tethering methods and their general processes.

### 3.1 Test AT Commands

```
#cat /dev/ttyUSB2 &
#echo -e "at\r\n">/dev/ttyUSB2
#
OK
```

#### NOTE

Check the driver to make sure that Simcom VID and PID information modification has been added.

### 3.2 Use PPP Data connection

#### 3.2.1 How Does a PPP Dial-Up Connection Work?

You will need the right software and a couple of pieces of information before you start. First, check the pppd. If the programs do not exist, you can download the source code from <https://ppp.samba.org/download.html> and port them to your embedded development environment. Next you must write configuration file for pppd.

### 3.2.1.1 Chat Scription

```
#named simcom-connect-chat and place in /etc/ppp/peers
ABORT "BUSY"
ABORT "NO CARRIER"
ABORT "NO DIALTONE"
ABORT "ERROR"
ABORT "NO ANSWER"
TIMEOUT 30
"" AT
OK ATE0
OK ATI;+CSUB;+CSQ;+CPIN?;+COPS?;+CGREG?;&D2
# Insert the APN provided by your network operator, default apn is 3gnet
OK AT+CGDCONT=1,"IP","3gnet",,0,0
OK ATD*99#
CONNECT
```

```
#named simcom-disconnect-chat and place in /etc/ppp/peers
ABORT "ERROR"
ABORT "NO DIALTONE"
SAY "\nSending break to the modem\n"
"" +++
"" +++
"" +++
SAY "\nGoodbay\n"
```

### 3.2.1.2 Configure dialing and AT port

```
# named simcom-pppd and place in /etc/ppp/peers
/dev/ttyUSB3 115200
#Insert the username and password for authentication, default user and password are test
```

```
user "test" password "test"
# The chat script, customize your APN in this file
connect 'chat -s -v -f /etc/ppp/peers/simcom-connect-chat'
# The close script
disconnect 'chat -s -v -f /etc/ppp/peers/simcom-disconnect-chat'
# Hide password in debug messages
hide-password
# The phone is not required to authenticate
noauth
# Debug info from pppd
debug
# If you want to use the HSDPA link as your gateway
defaultroute
# pppd must not propose any IP address to the peer
noipdefault
# No ppp compression
novj
novjccomp
noccip
ipcp-accept-local
ipcp-accept-remote
local
# For sanity, keep a lock on the serial line
lock
modem
dump
nodetach
# Hardware flow control
nocrtscts
remotename 3gppp
ipparam 3gppp
ipcp-max-failure 30
# Ask the peer for up to 2 DNS server addresses
usepeerdns
```

### 3.2.1.3 Dial-Up Connection

```
# pppd call simcom-pppd &
```

When you see the output below, it shows that dial-up succeeded.

```
Connect: ppp0 <--> /dev/ttyUSB3
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x5107d141> <pcomp> <accomp>]
rcvd [LCP ConfReq id=0x0 <asyncmap 0x0> <auth chap MD5> <magic 0x9a5c1936> <pcomp>
<accomp>]
sent [LCP ConfAck id=0x0 <asyncmap 0x0> <auth chap MD5> <magic 0x9a5c1936> <pcomp>
<accomp>]
rcvd [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0x5107d141> <pcomp> <accomp>]
sent [LCP EchoReq id=0x0 magic=0x5107d141]
rcvd [LCP DiscReq id=0x1 magic=0x9a5c1936]
rcvd [CHAP Challenge id=0x1 <dd93b9f04d75e2bbba3786f6d24df3d7>, name =
"UMTS_CHAP_SRVR"]
sent [CHAP Response id=0x1 <498d4d7cf3b59dacfc07a45ce6eb7e26>, name = "test"]
rcvd [LCP EchoRep id=0x0 magic=0x9a5c1936 51 07 d1 41]
rcvd [CHAP Success id=0x1 ""]
CHAP authentication succeeded
CHAP authentication succeeded
sent [IPCP ConfReq id=0x1 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
rcvd [IPCP ConfReq id=0x0]
sent [IPCP ConfNak id=0x0 <addr 0.0.0.0>]
rcvd [IPCP ConfNak id=0x1 <addr 10.51.68.23> <ms-dns1 222.66.251.8> <ms-dns2 116.236.159.8>]
sent [IPCP ConfReq id=0x2 <addr 10.51.68.23> <ms-dns1 222.66.251.8> <ms-dns2 116.236.159.8>]
rcvd [IPCP ConfReq id=0x1]
sent [IPCP ConfAck id=0x1]
rcvd [IPCP ConfAck id=0x2 <addr 10.51.68.23> <ms-dns1 222.66.251.8> <ms-dns2 116.236.159.8>]
Could not determine remote IP address: defaulting to 10.64.64.64
local IP address 10.51.68.23
```

```
remote IP address 10.64.64.64
primary   DNS address 222.66.251.8
secondary DNS address 116.236.159.8
Script /etc/ppp/ip-up started (pid 6616)
Script /etc/ppp/ip-up finished (pid 6616), status = 0x0
```

Now PPP call is set up successfully. Please use following commands to check IP/DNS/Route.

```
# ifconfig ppp0
ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.216.159.39  P-t-P:10.64.64.64  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:9 errors:0 dropped:0 overruns:0 frame:0
          TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:362 (362.0 B)  TX bytes:316 (316.0 B)

# cat /etc/resolv.conf
nameserver 221.180.132.108

# route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
0.0.0.0          0.0.0.0         0.0.0.0          U        0      0      0 ppp0
10.64.64.64     0.0.0.0         255.255.255.255 UH       0      0      0 ppp0

# ping baidu.com
PING baidu.com (220.181.57.216) 56(84) bytes of data:
64 bytes from 220.181.57.216: icmp_seq=1 ttl=50 time=84.0 ms
64 bytes from 220.181.57.216: icmp_seq=2 ttl=50 time=34.2 ms
```

Following commands can be used to terminate PPPD process to disconnect a PPP call:

```
# killall pppd
```

### 3.3 Use NDIS Data connection

Make sure the <PID> has been switched to 9001.

Please use the source code in the simcom-cm.zip package provided by our company. Adjust the Makefile file on the target host, and then compile them into the executable file or the library file if needed.

1) The Makefile in the source code applies to the Linux OS environment. After decompressing it in one Linux directory, enter the simcom-cm folder, and then execute the 'make' command to compile. If the source code is compiled successfully, the simcom-cm execution program will be generated. Connect the module to the Linux OS host through a USB cable. Please make sure the driver is installed successfully. Enter the directory which locates the simcom-cm execution program under root authority, and then execute './simcom-cm'.

```
# ./simcom-cm
[06-11_18:04:20:168] SIMCOM_CM START...
[06-11_18:04:20:169] ./simcom-cm profile[1] = (null)/(null)/(null)/0, pincode = (null)
[06-11_18:04:20:169] Find /sys/bus/usb/devices/1-5 idVendor=1e0e idProduct=9001
[06-11_18:04:20:169] Find /sys/bus/usb/devices/1-5:1.5/net/wwan0
[06-11_18:04:20:169] Find usbnet_adapter = wwan0
[06-11_18:04:20:169] Find /sys/bus/usb/devices/1-5:1.5/usbmisc/cdc-wdm0
[06-11_18:04:20:169] Find qmichannel = /dev/cdc-wdm0
[06-11_18:04:20:181] cdc_wdm_fd = 7
[06-11_18:04:20:274] Get clientWDS = 15
[06-11_18:04:20:306] Get clientDMS = 1
[06-11_18:04:20:338] Get clientNAS = 2
[06-11_18:04:20:370] Get clientUIM = 1
[06-11_18:04:20:402] Get clientWDA = 1
[06-11_18:04:20:434] requestBaseBandVersion MPSS.HI.2.0.5-00222.2-SDX55_CPEALL_PACK-1 1
[May 28 2020 17:00:00]
[06-11_18:04:20:498] requestGetSIMStatus SIMStatus: SIM_READY
[06-11_18:04:20:531] requestRegistrationState2 MCC: 460, MNC: 1, PS: Attached, DataCap: LTE
[06-11_18:04:20:562] requestQueryDataCall IPv4ConnectionStatus: DISCONNECTED
[06-11_18:04:20:626] requestRegistrationState2 MCC: 460, MNC: 1, PS: Attached, DataCap: LTE
```



```
[06-11_18:04:20:658] requestSetupDataCall WdsConnectionIPv4Handle: 0x1816e750
[06-11_18:04:20:722] requestQueryDataCall IPv4ConnectionStatus: CONNECTED
[06-11_18:04:20:754] ifconfig wwan0 up
[06-11_18:04:20:759] busybox udhcpc -f -n -q -t 5 -s /usr/share/udhcpc/default.script -i wwan0
[06-11_18:04:20:762] udhcpc (v1.22.1) started
[06-11_18:04:20:806] Sending discover...
[06-11_18:04:20:850] Sending select for 10.113.33.24...
[06-11_18:04:20:890] Lease of 10.113.33.24 obtained, lease time 7200
[06-11_18:04:20:897] /usr/share/udhcpc/default.script: Resetting default routes
SIOCDELRT: No such process
[06-11_18:04:20:902] /usr/share/udhcpc/default.script: Adding DNS 202.96.64.68
[06-11_18:04:20:902] /usr/share/udhcpc/default.script: Adding DNS 202.96.69.38
```

Please use following commands to check IP/DNS/Route.

```
# ifconfig wwan0
wwan0      Link encap:Ethernet  HWaddr c6:63:61:c4:bf:10
            inet addr:10.113.33.24  Bcast:10.113.33.31  Mask:255.255.255.240
            inet6 addr: fe80::c463:61ff:fec4:bf10/64 Scope:Link
            UP BROADCAST RUNNING NOARP MULTICAST  MTU:1500  Metric:1

# cat /etc/resolv.conf
nameserver 202.96.64.68
nameserver 202.96.69.38

# ip route show
default via 10.113.33.25 dev wwan0
10.113.33.16/28 dev wwan0  proto kernel  scope link  src 10.113.33.24

# ping baidu.com
PING baidu.com (123.125.114.144) 56(84) bytes of data.
64 bytes from 123.125.114.144: icmp_seq=1 ttl=56 time=114 ms
64 bytes from 123.125.114.144: icmp_seq=2 ttl=56 time=58.6 ms
64 bytes from 123.125.114.144: icmp_seq=3 ttl=56 time=45.1 ms
```

## 4 Troubleshooting

### 4.1 How to check whether the correct USB serial driver exists in the kernel?

If Linux does not create devices, check for the kernel module:

```
# lsmod | grep option
```

If entries aren't found, load the kernel module with root privileges:

```
# modprobe option
```

Check dmesg output to see that the radio was detected:

```
# dmesg | grep option
```

Check dmesg output to see that the radio was detected:

```
# dmesg | grep option
```

```
[ 16.672003] usbcore: registered new interface driver option
[ 16.672105] option 2-1.2:1.0: GSM modem (1-port) converter detected
[ 16.672216] option 2-1.2:1.1: GSM modem (1-port) converter detected
[ 16.672292] option 2-1.2:1.2: GSM modem (1-port) converter detected
[ 16.672365] option 2-1.2:1.3: GSM modem (1-port) converter detected
[ 16.672438] option 2-1.2:1.4: GSM modem (1-port) converter detected
```

### 4.2 What can I do if the port number does not start from ttyUSB0?

Check the ttyUSB port usage. Check whether the ttyUSB port is released when the module is disconnected.

### 4.3 How to check whether the correct QMI WWAN driver integration in the kernel?

```
[88843.080420] option 1-5:1.0: GSM modem (1-port) converter detected
[88843.080671] usb 1-5: GSM modem (1-port) converter now attached to ttyUSB0
[88843.080891] option 1-5:1.1: GSM modem (1-port) converter detected
[88843.081086] usb 1-5: GSM modem (1-port) converter now attached to ttyUSB1
[88843.081332] option 1-5:1.2: GSM modem (1-port) converter detected
[88843.082825] usb 1-5: GSM modem (1-port) converter now attached to ttyUSB2
[88843.083040] option 1-5:1.3: GSM modem (1-port) converter detected
[88843.083213] usb 1-5: GSM modem (1-port) converter now attached to ttyUSB3
[88843.083445] option 1-5:1.4: GSM modem (1-port) converter detected
[88843.083541] usb 1-5: GSM modem (1-port) converter now attached to ttyUSB4
[88843.648962] usbcore: registered new interface driver cdc_wdm
[88843.650716] qmi_wwan_simcom 1-5:1.5: cdc-wdm0: USB WDM device
[88843.650718] qmi_wwan_simcom 1-5:1.5: SIMCom 8200 work on RawIP mode
[88843.651108] qmi_wwan_simcom 1-5:1.5 wwan0: register 'qmi_wwan_simcom' at usb-0000:00:14.0-5,
WWAN/QMI device, aa:39:fd:18:de:95
[88843.651173] usbcore: registered new interface driver qmi_wwan_simcom
```

Check interface wwan0

```
wwan0      Link encap:Ethernet  HWaddr aa:39:fd:18:de:95
           inet6 addr: fe80::a839:fdff:fe18:de95/64 Scope:Link
           UP BROADCAST RUNNING 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)
```

## 4.4 How do I provide required logs if the corresponding ttyUSB ports are not found?

- Step 1 Open **Terminal**, run the command **dmesg**, and save the output to the **dmesg.txt** file.
- Step 2 Run the command **ls -l /sys/bus/usb/drivers/** and save the output to the **logcat.txt** file. Make sure the option folder is available under **/sys/bus/usb/drivers/** where the **logcat.txt** file is placed.
- Step 3 Run the command **ls -l /sys/bus/usb/drivers/option/\*\*/** and save the output to the **logcat.txt** file.
- Step 4 Run the command **cat /sys/bus/usb/drivers/option/\*\*/bInterface\*** and save the output to the **logcat.txt** file.
- Step 5 Run command the **cat /proc/bus/usb/devices** and save the output to the **logcat.txt** file. If the command cannot be executed, skip this step.