

SIM7X00 Linux Dial-up User Guide

V1.03

(For SIM7500, SIM7800, SIM7900)

Version History

Data	Version	Description of change	Author
2018-12-10	V1.00	1) Combine separate technical documents into one. 2) Refine commands by the sense of common user.	Yongfeng duan
2019-1-11	V1.01	Add more supports to RNDIS and NDIS	Yongfeng duan
2019-2-11	V1.02	Remove modemmanger under Ubuntu	Yongfeng duan
2019-4-24	V1.03	Add more supports to RNDIS	Yongfeng duan

Table of Contents

1	USB Device.....	3
1.1	Description.....	3
1.2	Methods of Dialing-up.....	3
1.3	Install USB Serial Driver.....	3
1.3.1	Linux kernel source code.....	3
1.3.2	Modify .config.....	4
1.3.3	Modify VID PID and interfaces.....	4
1.3.4	Patch to reset and resume.....	6
1.3.5	Compile and install.....	6
1.3.6	Verify.....	6
1.4	Trouble shooting.....	7
2	Ppp.....	9
2.1	Install device driver.....	9
2.2	Dial-up.....	9
2.2.1	prepare configure file.....	9
2.2.2	Prepare script file.....	10
2.2.3	Dial-up.....	10
2.2.4	Verify.....	11
2.2.5	Disconnect network.....	11
2.3	Trouble shooting.....	12
3	NDIS.....	13
3.1	Install USB Net Driver.....	13
3.2	Check.....	13
3.3	Dial up.....	14
3.4	Get IP.....	14
3.5	NDIS AT command.....	15
3.6	NDIS dial-up cases.....	16
	Public Network. (LTE, UMTS, GSM).....	16
	Private Network.....	16
3.7	Trouble shooting.....	17
4	Rndis.....	19
4.1	Dial-up.....	19
4.2	Support AT command for Linux.....	20
4.3	Support Android.....	20
4.4	Restore product ID to the default 9001.....	21

1 USB DEVICE

1.1 Description

For SIM7500_SIM7600 serial module, Vendor ID (VID) is 0x1E0E, default Product ID (PID) is 0x9001. For RNDIS, PID is 0x9011, For ECM, PID is 0x9018.

As a slave USB device, SIM7500_SIM7600 USB is enumerated as listed below.

Interface #	Linux name	Description
0	USB serial	Diagnostic port for debugging purpose
1	USB serial	NMEA port for GPS NMEA data output
2	USB serial	AT port Interface
3	USB serial	Modem port Interface for PPP dial-up
4	USB serial	USB Audio Interface
5	USB Net	wwan interface
6	USB adb	Android adb debug port

after Linux recognizes the modem, USB devices named /dev/ttyUSBx.

For these serial ports, default baud rate is 115200.

1.2 Methods of Dialing-up

The module provides several dialing-up methods: PPP, RNDIS, NDIS. They will be introduced in following chapters.

Before dialing-up, the first step is to update USB-Serial driver in kernel source code.

1.3 Install USB Serial Driver

1.3.1 Linux kernel source code

Linux kernel's source code can be downloaded from <https://www.kernel.org/>

If you plan to test under Ubuntu, source code can obtained by shell command either.

```
# apt-cache search linux-source
```

```
linux-source - Linux kernel source with Ubuntu patches
```

```
linux-source-4.4.0 - Linux kernel source for version 4.4.0 with Ubuntu patches
```

```
linux-source-4.10.0 - Linux kernel source for version 4.10.0 with Ubuntu patches
```

```
linux-source-4.11.0 - Linux kernel source for version 4.11.0 with Ubuntu patches
```

Then check the current Linux version

```
#uname -a
```

*Linux dawnu16 4.15.0-39-generic #42~16.04.1-Ubuntu SMP Wed Oct 24 17:05:15 UTC 2018
i686 i686 i686 GNU/Linux*

Select the working version to start downloading. The downloaded package locates at /usr/src/

```
# sudo apt-get install linux-source-4.15.0
```

The downloaded package locates at /usr/src/. Then extract the source package

```
# sudo tar -jxv -f linux-source-4.15.0.tar.bz2
```

1. 3. 2 Modify .config

To build the driver, please modify make configuration file and ensure:

```
CONFIG_USB_SERIAL=y  
CONFIG_USB_SERIAL_WWAN=y  
CONFIG_USB_SERIAL_OPTION=y
```

These options can be edited in .config file directly.

The .config file is usually copied from the default configure file, which locates at kernel/arch/"arch-name"/configs/.

After .config is created under root directory of kernel, you can either editing related options by

```
# sudo make menuconfig
```

1. 3. 3 Modify VID PID and interfaces

To find the source code file *option.c* in kernel. (Usually, it is located in the path: *drivers/usb/serial/option.c*)

- If the kernel version is v3.2 or newer.

```
#define SIMCOM_VENDOR_ID          0x1E0E  
#define SIMCOM_SIM7600_PID        0x9001  
#define SIMCOM_PRODUCT_PID_X9011 0x9011  
#define SIMCOM_PRODUCT_PID_X9018 0x9018
```

Add into **option_ids** list

```
//for SIM7600 modem for NDIS
```

```
{ USB_DEVICE(SIMCOM_VENDOR_ID, SIMCOM_SIM7600_PID),
    .driver_info = RSVD(5) | RSVD(6) },
//for SIM7600 modem for RNDIS
{ USB_DEVICE(SIMCOM_VENDOR_ID, SIMCOM_PRODUCT_PID_X9011),
    .driver_info = RSVD(0) | RSVD(1) | RSVD(7) | RSVD(8) },
// for SIM7600 modem for ECM
{ USB_DEVICE(SIMCOM_VENDOR_ID, SIMCOM_SIM7600_PID),
    .driver_info = RSVD(5) | RSVD(6) | RSVD(7) },
```

● If the kernel version is below v3.2.

```
#define SIMCOM_SIM7600_VID        0x1E0E
#define SIMCOM_SIM7600_PID        0x9001
#define SIMCOM_PRODUCT_PID_X9011  0x9011
#define SIMCOM_PRODUCT_PID_X9018  0x9018
```

Add into `option_ids` list

```
{ USB_DEVICE(SIMCOM_SIM7600_VID, SIMCOM_SIM7600_PID)}, /*SIM7600 */
{ USB_DEVICE(SIMCOM_SIM7600_VID, SIMCOM_PRODUCT_PID_X9011)}, /*SIM7600 */
{ USB_DEVICE(SIMCOM_SIM7600_VID, SIMCOM_PRODUCT_PID_X9018)}, /*SIM7600 */
```

And skip interface #5 in `option_probe`.

```
/* SIM7600 */
if (serial->dev->descriptor.idVendor == SIMCOM_SIM7600_VID &&
    serial->dev->descriptor.idProduct == SIMCOM_SIM7600_PID &&
    (serial->interface->cur_altsetting->desc.bInterfaceNumber == 5 ||
    serial->interface->cur_altsetting->desc.bInterfaceNumber == 6) )
    return -ENODEV;

if (serial->dev->descriptor.idVendor == SIMCOM_SIM7600_VID &&
    serial->dev->descriptor.idProduct == SIMCOM_PRODUCT_PID_X9011 &&
    (serial->interface->cur_altsetting->desc.bInterfaceNumber == 0 ||
    serial->interface->cur_altsetting->desc.bInterfaceNumber == 1 ||
    serial->interface->cur_altsetting->desc.bInterfaceNumber == 7 ||
    serial->interface->cur_altsetting->desc.bInterfaceNumber == 8) )
    return -ENODEV;

if (serial->dev->descriptor.idVendor == SIMCOM_SIM7600_VID &&
    serial->dev->descriptor.idProduct == SIMCOM_PRODUCT_PID_X9018 &&
    (serial->interface->cur_altsetting->desc.bInterfaceNumber == 5 ||
    serial->interface->cur_altsetting->desc.bInterfaceNumber == 6 ||
    serial->interface->cur_altsetting->desc.bInterfaceNumber == 7) )
```

```
return -ENODEV;
```

1. 3. 4 Patch to reset and resume

Some USB host controller/USB hubs will lost power or be reset when MCU entering into suspend/sleep mode, and they cannot resume USB devices after MCU exits from suspend/sleep mode. Please add the following statements to enable reset-resume process.

```
static struct usb_serial_driver option_1port_device = {
.....
#ifdef CONFIG_PM
    .suspend = usb_wwan_suspend,
    .resume = usb_wwan_resume,
    #if 1 //Added by SIMCom. Supported by newer Linux kernel version
        .reset_resume = usb_wwan_resume,
    #endif
#endif
};
```

1. 3. 5 Compile and install

Please install necessary components for compiling

```
# sudo apt-get install libncurses5-dev
# sudo apt-get install libssl-dev
```

In Ubuntu, please follow next commands to compile and install USB-Serial module.

```
# cd kernel_src_dir
# sudo make -C /lib/modules/$(uname -r)/build M=$(pwd)/drivers/usb/serial modules
# sudo cp drivers/usb/serial/option.ko /lib/modules/$(uname -r)/kernel/drivers/usb/serial/
# sudo depmod
```

1. 3. 6 Verify

If USB serial driver is installed successfully, kernel will print below message automatically after module was re-started. And from this message, we could confirm if **dev/ttyUSB#** was enumerated successfully or not.

```
#dmesg
usb 1-1: new high speed USB device using rt3xxx-ehci and address 2
option 1-1:1.0: GSM modem (1-port) converter detected
usb 1-1: GSM modem (1-port) converter now attached to ttyUSB0
option 1-1:1.1: GSM modem (1-port) converter detected
```

```
usb 1-1: GSM modem (1-port) converter now attached to ttyUSB1
option 1-1:1.2: GSM modem (1-port) converter detected
usb 1-1: GSM modem (1-port) converter now attached to ttyUSB2
option 1-1:1.3: GSM modem (1-port) converter detected
usb 1-1: GSM modem (1-port) converter now attached to ttyUSB3
option 1-1:1.4: GSM modem (1-port) converter detected
usb 1-1: GSM modem (1-port) converter now attached to ttyUSB4
```

1. 3. 7 Send AT commands

Add current user to dialout group firstly

```
# sudo usermod -aG dialout ${USER}
```

If your are testing under Ubuntu desktop version, please remove modemmanager, and then reboot Linux.

```
# sudo apt-get remove modemmanager
```

busybox microcom is recommended to send AT commands to SIM7600 module. The default baud rate is 115200.

```
# busybox microcom -s 115200 /dev/ttyUSB2
AT
OK
ctrl+x to quit
```

If no busybox, an AT command can be sent via Linux Shell in following style

```
# sudo echo -en "AT\r\n" > /dev/ttyUSB2;cat /dev/ttyUSB2
OK
ctrl+c to quit
```

1. 4 Trouble shooting

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

```
# lsmod | grep option
option                45056  0
usb_wwan              20480  1 option
usbserial             40960  2 option,usb_wwan
```

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

```
#sudo modprobe option
```

Check dmesg output to see that the radio was detected:

```
# dmesg | grep option
```

```
[16.672] usbcore: registered new interface driver option
```

```
[16.672] option 2-1.2:1.0: GSM modem (1-port) converter detected
```

```
[16.672] option 2-1.2:1.1: GSM modem (1-port) converter detected
```

```
[16.672] option 2-1.2:1.2: GSM modem (1-port) converter detected
```

```
[16.672] option 2-1.2:1.3: GSM modem (1-port) converter detected
```

```
[16.672] option 2-1.2:1.4: GSM modem (1-port) converter detected
```


2 PPP

2.1 Install device driver

Check if ppp device has been included in your Linux system.

```
#ls /dev/ppp
```

If device does not exist, please enable ppp option during compiling.

```
#cd kernel_src_dir
#make menuconfig
```

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

```
[*] Network device support --->
```

```
<*> PPP (point-to-point protocol) support
```

```
[*] PPP multilink support (EXPERIMENTAL)
```

```
<*> PPP support for async serial ports
```

```
<*> PPP support for sync tty ports
```

```
<*> SLIP (serial line) support
```

```
[*] CSLIP compressed headers
```

If you don't have kernel source code, you can download the source code from <https://ppp.samba.org/download.html> and port it to your embedded system.

In Ubuntu, you can install ppp component.

```
#sudo apt-get install ppp
```

2.2 Dial-up

2.2.1 prepare configure file

Configure file's path is /etc/ppp/peers/lte-dial
/dev/ttyUSB2 is SIM7600's AT port

your_apn is an operator's APN, which should match the USIM card and operator's business.

Content in this file:

```
hide-password
noauth
connect "/usr/sbin/chat -v -t 50 -f /etc/chatscripts/lte-chat"
debug
```

```

/dev/ttyUSB2
460800
defaultroute
noipdefault
local
lock
dump
nodetach
user " "
remotename your_apn
ipparam your_apn
usepeerdns

```

2. 2. 2 Prepare script file

The script file path is /etc/chatscripts/lte-chat

Content in this file:

```

#TIMEOUT 15
ABORT '\nBUSY\r'
ABORT '\nNO ANSWER\r'
ABORT '\nRINGING\r\n\r\nRINGING\r'
ABORT '\n+CME ERROR: 100\r'

#" AT
#'OK-+++ \c-OK' ATH0
#TIMEOUT 40
#" \rAT
" ATZ
#OK ATS0=0
#OK ATE0V1
OK AT+CGDCONT=1,"IP","your_apn"
OK AT+CGEQREQ=1,2,128,384,0,0,0,0,"0E0","0E0",,0,0
OK ATDT*98*1#
CONNECT

```

2. 2. 3 Dial-up

Before dialing-up, please check with AT commands to ensure the module SIM7600 has attached to wireless network successfully.

- 1) Check if network attached successfully by send an AT command

```
AT+CGATT?
```

```
+CGATT: 1  
OK
```

2) Start to Dial-up

```
# sudo pppd call lte-dial  
pppd options in effect:  
debug debug      # (from /etc/ppp/peers/lte-dial)  
...  
local IP address 172.17.93.102  
remote IP address 10.64.64.64  
Primary DNS address 112.65.184.255  
secondary DNS address 210.22.84.3  
Script /etc/ppp/ip-up started (pid 11642)  
Script /etc/ppp/ip-up finished (pid 11642), status = 0x0
```

When connect network successfully, local IP address will be assigned. With *ifconfig*, ppp0 interface is available.

2. 2. 4 Verify

Disable other existing connection except ppp0

```
# ifconfig  
...  
ppp0      Link encap:Point-to-Point Protocol  
          inet addr:100.97.213.248  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)  
  
# sudo ifconfig eth0 down  
# ping 114.114.114.114  
PING 114.114.114.114 (114.114.114.114) 56(84) bytes of data.  
64 bytes from 114.114.114.114: icmp_seq=1 ttl=64 time=9.67 ms  
64 bytes from 114.114.114.114: icmp_seq=2 ttl=90 time=11.0 ms
```

2. 2. 5 Disconnect network

The script file path is `/etc/chatscripts/disconnect-lte-chat`
Content in this file:

```
ABORT "ERROR"
```

```
ABORT "NO DIALTONE"
SAY "\nSending break to the modem\n"
"" +++
"" +++
"" +++
SAY "\nGoodbay\n"
```

2.3 Trouble shooting

- 1 Report error message when dialing up.
Please check your_apn, and if configure and script files have been put at proper location.
- 2 Can not obtain local address after dial-up
Please check if the module SIM7600 can attached wireless network successfully with AT command.
- 3 Have got local address successfully, but failed when verifying
If it is a permission issue.

```
# sudo usermod -a -G /dev/ppp $USER
```


If it is a route issue

```
# sudo route add default dev ppp0
```

3 NDIS

3.1 Install USB Net Driver

If linux kernel's version is greater than 3.4.1, needn't update usbnet driver, please skip this section.

- 1) To build the driver, please modify make configuration and ensure:

```
CONFIG_USB_USBNET=y
```

More information, please refer to [Modify .config](#)

- 2) Put `sim7500_sim7600_wwan.c` into the path `drivers/net/usb`, then modify `Makefile` under this directory.

```
obj-$(CONFIG_USB_USBNET) += usbnet.o sim7500_sim7600_wwan.o
```

(Note: `sim7500_sim7600_wwan.c` is provided by SIMCom)

- 3) Then start to compile and install the modules:

```
#cd kernel_src_dir
#sudo make -C /lib/modules/$(uname -r)/build M=$(pwd)/drivers/net/usb modules
#sudo cp ./drivers/net/usb/sim7500_sim7600_wwan.ko /lib/modules/$(uname -r)/kernel/drivers/net/usb/
#sudo depmod
```

- 4) If installed successfully, kernel will print below message automatically after module was re-started.

```
# dmesg | grep sim_wwan
sim_wwan 1-1:1.5 wwan0: register 'sim_wwan' at usb-0000:02:03.0-1, SIMCOM wwan/QMI device, 8a:de:f6:67:ce:1b
usbcore: registered new interface driver simcom_wwan
usbcore: registered new interface driver qmi_wwan
```

3.2 Check

"ifconfig -a" command can be used for checking NIC (Network Interface Card) information. (The status is "down" by default)

```
wwan0 Link encap:Ethernet HWaddr d2:7c:5c:66:09:05
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)
RX bytes:4881 (4.8 KB) TX bytes:8608 (8.6 KB)
```

3.3 Dial up

SIM7600 module starts to dial up via NIC by using below AT command.

3) Check if network attached successfully

```
AT+CGATT?
+CGATT: 1
OK
```

4) Start to dial up

```
AT$QCRMCALL=1,1
$QCRMCALL: 1, V4
OK
```

If this command is sent by shell, please add slash char before \$, e.g,

```
# sudo echo -en "AT\$QCRMCALL=1,1\r\n" > /dev/ttyUSB2; cat /dev/ttyUSB2
$QCRMCALL: 1, V4
OK
```

Note: For more information, please refer to [NDIS AT command](#).

3.4 Get IP

After dial up via NIC, customer need get NIC IP and DNS via DHCP client.

If the host environment is Linux OS on PC (such as Ubuntu), just change NIC from “down” to “up”, and then PC will get IP and DNS via DHCP program which included in NetManager automatically.

If the host environment is embedded Linux OS, normally should use *udhcpc* command below.

```
# sudo udhcpc -i wwan0
udhcpc (v1.22.1) started
Sending discover...
```

Sending select for 10.49.221.76...
Lease of 10.49.221.76 obtained, lease time 7200

After that, NIC information will be refreshed as below.

wwan0 Link encap:Ethernet HWaddr d2:7c:5c:66:09:05
inet addr:10.49.221.76 Bcast:10.49.221.79 Mask:255.255.255.248
inet6 addr: fe80::d07c:5cff:fe66:905/64 Scope:Link
UP BROADCAST RUNNING NOARP MULTICAST MTU:1500 Metric:1
RX packets:47 errors:0 dropped:0 overruns:0 frame:0
TX packets:96 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000

To confirm if connection is OK, please close other connections, then test with ping command.

ping 114.114.114.114
PING 114.114.114.114 (114.114.114.114) 56(84) bytes of data.
64 bytes from 114.114.114.114: icmp_seq=1 ttl=64 time=9.67 ms
64 bytes from 114.114.114.114: icmp_seq=2 ttl=90 time=11.0 ms

3. 5 NDIS AT command

AT+QCRMCall=<action>, <instance> [,<IP type> [,<Tech pref> [, <umts_profile> [,<cdma_profile> [,<APN> [,<username>,<password>[,<auth pref>]]]]]]]

<action>

Stop

1 - Start

<instance>

Rmnet instance

<Ip type>

IPV4

IPV6

IPV4V6

<Tech pref>

1-3GPP2

2-3GPP

<umts profile>

1-16

<cdma profile>

1-200

<APN>

String of APN

<username>

Username for authentication

<password>

Password for authentication

<auth pref>

NONE

PAP

CHAP

PAP or CHAP

3. 6 NDIS dial-up cases

Firstly, please to be cleared that **Dial up** and **Attach network** are two conceptions. Attach work is the precondition of dial up, without network attached, dial up must be failed.

Secondly, for some LTE network, especially for some private network, it may ask for auth type, user name and password. So customer should set them correctly before dial up.

Public Network. (LTE, UMTS, GSM)

1. Make sure that SIM7500_SIM7600 already register to the network;
(please use AT+CREG?, AT+CGREG?, AT+CPSI?)
2. Start to dual up by using AT\$QCRMCall=1,1
(BTW, AT\$QCRMCall=0,1 used for disconnecting, AT\$QCRMCall? used for checking the status.)
3. Get IP: udhcpc -i wwan0
4. Confirm network card information: ifconfig.

Private Network.

Private Network. (UMTS, GSM)

1. Make sure that SIM7500_SIM7600 already register to the network;
(please use AT+CREG?, AT+CGREG?, AT+CPSI?)
2. Start to dual up by using
AT\$QCRMCall=1,1,,,,,"apn","user name","password","auth pref"
3. Get IP: udhcpc -i wwan0

4. Confirm network card information: ifconfig.

Private Network. (LTE)

Mode #1: First register to public network, and then dial up to private network.

1. Make sure that SIM7500_SIM7600 already register to the network; (please use AT+CREG?, AT+CGREG?, AT+CPSI?)
2. Start to dual up by using
AT\$QCRMCall=1,1,,,,,"apn","user name","password","auth pref"
3. Get IP: udhcpc -i wwan0
4. Confirm network card information: ifconfig.

Mode #2: First register to private network, and then dial up.

1. Set APN by using AT+CGDCONT=1,"IP","apn"
2. Set authentication type, user name and password by using
AT+CGAUTH=1,3,"password","user name"
3. Make sure that SIM7500_SIM7600 already register to the network; (please use AT+CREG?, AT+CGREG?, AT+CPSI?)
4. Start to dual up by using
AT\$QCRMCall=1,1
5. Get IP: udhcpc -i wwan0
6. Confirm network card information: ifconfig.

CDMA/EVDO Network.

1. Make sure that SIM7500_SIM7600 already register to the network; (please use AT+CREG?, AT+CGREG?, AT+CPSI?.)
2. Start to dual up by using AT\$QCRMCall; (Refer to notes below)
3. Get IP: udhcpc -i wwan0
4. Confirm network card information: ifconfig.

Note: Normally, there is no need to input user name and pass word during attached network process; But during dial up process, network ask for them. So,

a) If under CDMA/EVDO public network;

Using AT\$QCRMCall=1,1,,,,,"ctwap@mycdma.cn","vnet.mobi"

b) If under CDMA/EVDO private network:

Using AT\$QCRMCall=1,1,,,,,"user name","password"

3. 7 Trouble shooting

- 1 If the module repose 'No Carrier'
 - a) Please try disconnecting firstly by AT\$QCRMCall=0,1

- b) Please add following lines to `/etc/network/interfaces`, then reboot Ubuntu

```
wwan0
iface wwan0 inet manual
iface wwan0 inet6 manual
nameserver 114.114.114.114
```

- c) If your are testing under Ubuntu desktop version, please remove modemmanager, and then reboot Linux.
- d) If your are testing under Ubuntu desktop version, please disable or kill network manager, and then reboot Linux.
- e) make sure these modules are not loaded: qmi_wwan, cdc_wdm.
- 2 If resulting 'Sending discover...' during dialing up with udhcpc,
- a) If kernel's version is older (≤ 3.1), replace `kernel/net/usb/usbnet.c` with provided file.
- b) the interface failed to obtain IP address, please disable or kill network manager, and then reboot Linux.
- 3 If testing with a Virtual machine, the wwan0 maybe renamed, like wwp0s11u1i5.
- 4 Set no AT echo mode by ATE0.

4 RNDIS

The Remote Network Driver Interface Specification (RNDIS) is a Microsoft proprietary protocol used mostly on top of USB. It provides a virtual Ethernet link to most versions of the Windows, Linux, and FreeBSD operating systems. Linux already implements RNDIS natively.

4.1 Dial-up

1) Switch to RNDIS mode

Ensure the related module is loaded

```
# sudo modprobe rndis_host
```

Send an AT command

```
AT+CUSBPIDSWITCH=9011,1,1
```

OK

The module SIM7600 will restart after receiving this command.

2) Check RNDIS interface

```
# dmesg | grep -i rndis
```

```
[ 35.786] rndis_host 2-1.2:1.0 usb0: register 'rndis_host' at usb- 0000:00:1d.0-1.2, RNDIS device, 16:43:65:ea:9c:ab
```

```
[ 35.786] usbcore: registered new interface driver rndis_host [ 4335.879128] usbcore: registered new interface driver rndis_wlan [ 4335.885735] rndis_host 2-1.2:1.0 enp0s29u1u2: renamed from usb0
```

```
[ 93.468] rndis_host 2-1.2:1.0 enp0s29u1u2: unregister 'rndis_host' usb- 0000:00:1d.0-1.2, RNDIS device
```

```
[ 102.490] rndis_host 2-1.2:1.0 usb0: register 'rndis_host' at usb- 0000:00:1d.0-1.2, RNDIS device, 16:43:65:ea:9c:ab
```

3) Check

If having no IP assigned, please apply for one manually.

```
# ifconfig -a
```

```
# sudo udhcpc -i usb0
```

4) Verify

```
# ping 114.114.114.114
```

```
PING 114.114.114.114 (114.114.114.114) 56(84) bytes of data.  
64 bytes from 114.114.114.114: icmp_seq=1 ttl=64 time=9.67 ms  
64 bytes from 114.114.114.114: icmp_seq=2 ttl=90 time=11.0 ms
```

4. 2 Support AT command for Linux

- 1) Enable usb0,
In .config file, check the following configuration ,if not y ,please set y
`CONFIG_USB_NET_RNDIS_HOST=y`
`CONFIG_USB_NET_CDC_SUBSET=y`

- 2) Then start to compile and install the modules:

```
# cd kernel_src_dir  
# sudo make -C /lib/modules/$(uname -r)/build M=$(pwd)/drivers/net/usb modules  
# sudo cp ./drivers/net/usb/*.ko /lib/modules/$(uname -r)/kernel/drivers/net/usb/  
# sudo depmod
```

- 3) Install serial driver

For a temporary test:

```
# sudo modprobe option  
# sudo su  
# echo "1e0e 9011" > /sys/bus/usb-serial/drivers/option1/new_id
```

For permanency usage, Please refer to section [1.3.3 Modify VID PID](#)

- 4) Verify:

```
# ls /dev/ttyU*  
/dev/ttyUSB0 /dev/ttyUSB1 /dev/ttyUSB2 /dev/ttyUSB3 /dev/ttyUSB4
```

Send an AT command

AT

OK

4. 3 Support Android

- 1) Enable usb0
check the following configuration
`CONFIG_USB_NET_RNDIS_HOST=y`
`CONFIG_USB_NET_CDC_SUBSET=y`

2) Close eth0

Modify CONFIG_ETHERNET=n in the config file in kernel/arch/arm(arm64). After this step, you can burn android to see if the usb0 port appears and there is no eth0 port.

3) Edit kernel/drivers/net/usb/usbnet.c, change *usb%d* to *eth%d* in statement
strcpy (net->name, "usb%d");

4) Compile the kernel, and make sure that the modified location is compiled.

4. 4 Restore product ID to the default 9001

If you want to restore to default product ID, please use following command. Where ttyUSB2 should be changed to your local AT interface.

Send an AT command

```
AT+CUSBPIDSWITCH=9001,1,1
```

OK

4. 5 Trouble shooting

1 Have got local address successfully, but failed when verifying

If it is a route issue

```
# sudo route add default dev usb0
```