SIM7070_SIM7080_SIM7090 Series_Linux _Application Note

LPWA Module

SIMCom Wireless Solutions Limited
Building B, SIM Technology Building, No.633, Jinzhong Road
Changning District, Shanghai P.R. China
Tel: 86-21-31575100
support@simcom.com
www.simcom.com

Document Title:	SIM7070_SIM7080_SIM7090 Series_Linux_Application Note
Version:	1.02
Date:	2020.7.8
Status:	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
Building B, SIM Technology Building, No.633 Jinzhong Road, Changning District, Shanghai P.R. China
Tel: +86 21 31575100
Email: 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

About Document

Version History

Version	Date	Owner	What is new
V1.00	2020.02.26	Dong.Liu	First Release
V1.01	2020.03.31	Ping.Zhang	All
V1.02	2020.07.8	Xiannging.Meng	1. Add VID=0x1E0E, PID=0x9206. 2. How to enable ECM.

Scope

This document applies to the following products

Name	Type	Size(mm)	Comments
SIM7080G	CAT-M/NB	17.6*15.7*2.3	N/A
SIM7070G/SIM7070E	CAT-M/NB/GPRS	24*24*2.4	N/A
SIM7070G-NG	NB/GPRS	24*24*2.4	N/A
SIM7090G	CAT-M/NB	14.8*12.8*2.0	N/A

Contents

1 Introduction

1.1 Purpose of the document

This document will introduce how to install USB driver on Linux and how to run PPP application on SIM7070\7080\7090 series of module. Developers could understand and develop application quickly and efficiently based on this document.

Developers could understand and develop application quickly and efficiently based on this document.

1.2 Related documents

[1] SIM7070_SIM7080_SIM7090 Series_AT Command Manual

1.3 Conventions and abbreviations

In this document, the GSM engines are referred to as following term:

- ME (Mobile Equipment);
- MS (Mobile Station);
- TA (Terminal Adapter);
- DCE (Data Communication Equipment) or facsimile DCE (FAX modem, FAX board);

In application, controlling device controls the GSM engine by sending AT Command via its serial interface. The controlling device at the other end of the serial line is referred to as following term:

- TE (Terminal Equipment);
- DTE (Data Terminal Equipment) or plainly "the application" which is running on an embedded system;

2 USB Introduction

The USB (Universal Serial BUS) protocol states that all USB devices have a VID (Vendor ID) and a PID (Product ID). The VID is applied by the supplier to the USB-IF (Implementers Forum, Applicant Forum). The VID of each supplier is unique and the PID is at the discretion of the supplier. The host uses VID and PID to identify different USB devices. Depending on them (and the version number of the device), the corresponding driver can be loaded or installed on the device. Both VID and PID are two bytes in length.

For SIM7070/SIM7080/SIM7090 series of module, there are two types of VID and PID. One is VID=0x1E0E, and PID=0x9205 and another is VID=0x1E0E, and PID=0x9206. The two types can be switched by AT+CUSBSELNV. If AT+CUSBSELNV=1, VID=0x1E0E, and PID=0x9205. If AT+CUSBSELNV=86, VID=0x1E0E, and PID=0x9206. The default is VID=0x1E0E, and PID=0x9206.

As an USB device, SIM7070/SIM7080/SIM7090 USB is enumerated as listed below when VID=0x1E0E, and PID=0x9205.

Interface number	Endpoint Type	Function
0	USB serial	Diagnostic Interface
1	USB serial	GPS NMEA Interface
2	USB serial	AT port Interface
3	USB serial	Modem port Interface
4	USB ECM	ECM Interface

As an USB device, SIM7070/SIM7080/SIM7090 USB is enumerated as listed below when VID=0x1E0E, and PID=0x9206.

Interface number	Endpoint Type	Function
0	USB serial	Diagnostic Interface
1	USB serial	GPS NMEA Interface
2	USB serial	AT port Interface
3	USB serial	QFLOG Interface
4	USB serial	DAM Interface
5	USB serial	Modem port Interface

3 AT Commands for USB configuration

Command	Description
AT+CUSBSELNV	Select the USB configuration
AT+SECMEN	Enable ECM auto connecting.
AT+CREBOOT	Reset the module
AT+SECMAUTH	Set APN of ECM auto connecting

For detail information, please refer to “SIM7070_SIM7080_SIM7090 Series_AT Command Manual ”.

4 Install USB Serial Driver

Before install USB driver, please make sure module has been powered up and connected with Linux, developer can check the hardware connection by *lsusb* or *dmesg* log.

4.1 Precondition

Configure Linux kernel as following to support USB serial features.

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

4.2 Add VID and PID

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

If kernel version is V3.2 or newer

```
#define SIMCOM_SIM7080_VID                  0x1E0E
/*If you want to use VID=0x1E0E, and PID=0x9205.*/
#define SIMCOM_SIM7080_PID                  0x9205
/*If you want to use VID=0x1E0E, and PID=0x9206.*/
#define SIMCOM_SIM7080_PID                  0x9206

   static const struct option_blacklist_info simcom_SIM7080_blacklist = {
   };
```

Add into ***option_ids*** list

```
    … …

   { USB_DEVICE(SIMCOM_SIM7080_VID, SIMCOM_SIM7080_PID),
    .driver_info = (kernel_ulong_t)& simcom_SIM7080_blacklist t
   },
    … …
```

If kernel version is below V3.2

```
#define SIMCOM_SIM7080_VID                  0x1E0E
/* If you want to use VID=0x1E0E, and PID=0x9205.*/
#define SIMCOM_SIM7080_PID                  0x9205
```

```
/*If you want to use VID=0x1E0E, and PID=0x9206.*/
#define SIMCOM_SIM7080_PID                    0x9206
```

Add into ***option_ids*** list

```
static const struct usb_device_id option_ids[] = {

{ USB_DEVICE(SIMCOM_SIM7080_VID, SIMCOM_SIM7080_PID) },

};

static int option_probe(struct usb_serial *serial,
 const struct usb_device_id *id) {
......
 if (serial->dev->descriptor.idVendor == SIMCOM_SIM7080_VID &&
        serial->dev->descriptor.idProduct == SIMCOM_SIM7080_PID)
 return -ENODEV;
......
}
```

4.3 Kernel debug message

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.

When VID=0x1E0E, and PID=0x9205, the kernel debug message is as following:

```
cdc_ether 1-2:1.4 eth1: register 'cdc_ether' at usb-0000:00:14.0-2, CDC Ethernet Device, 00:a0:c6:cf:2a:f0
...
option 1-2:1.0: GSM modem (1-port) converter detected
usb 1-2: GSM modem (1-port) converter now attached to ttyUSB0
option 1-2:1.1: GSM modem (1-port) converter detected
usb 1-2: GSM modem (1-port) converter now attached to ttyUSB1
option 1-2:1.2: GSM modem (1-port) converter detected
usb 1-2: GSM modem (1-port) converter now attached to ttyUSB2
option 1-2:1.3: GSM modem (1-port) converter detected
usb 1-2: GSM modem (1-port) converter now attached to ttyUSB3
```

When VID=0x1E0E, and PID=0x9206, the kernel debug message is as following:

```
usbserial: USB Serial support registered for GSM modem (1-port)
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
```

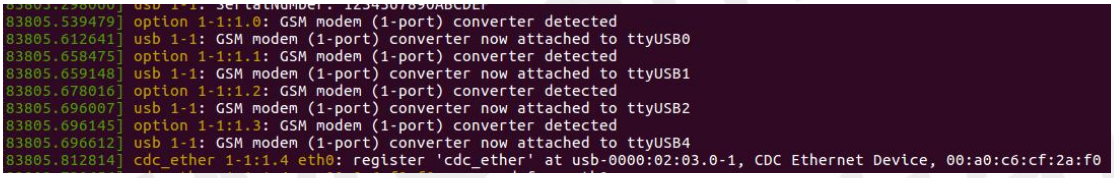
```
option 1-1:1.2: GSM modem (1-port) converter detected
usb 1-1: GSM modem (1-port) converter now attached to ttyUSB2
option 1-1:1.3: GSM modem (1-port) converter detected
usb 1-1: GSM modem (1-port) converter now attached to ttyUSB3
option 1-1:1.4: GSM modem (1-port) converter detected
usb 1-1: GSM modem (1-port) converter now attached to ttyUSB4
option 1-1:1.5: GSM modem (1-port) converter detected
usb 1-1: GSM modem (1-port) converter now attached to ttyUSB5
```

5 Verification

5.1 Verification of VID=0x1E0E and PID=0x9205

Now developer can verify if driver has been installed, when VID=0x1E0E, and PID=0x9205.

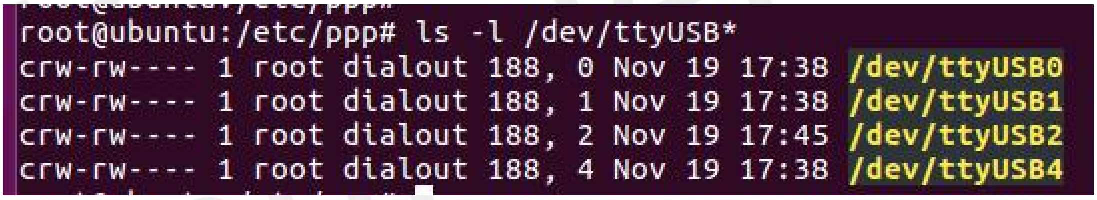
1) Connect physical USB interface of module and power on it.
2) Open Linux terminal and type the shell command “**dmesg”** to view kernel print information.



```
83805.539479] option 1-1:1.0: GSM modem (1-port) converter detected
83805.612641] usb 1-1: GSM modem (1-port) converter now attached to ttyUSB0
83805.658475] option 1-1:1.1: GSM modem (1-port) converter detected
83805.659148] usb 1-1: GSM modem (1-port) converter now attached to ttyUSB1
83805.678016] option 1-1:1.2: GSM modem (1-port) converter detected
83805.696007] usb 1-1: GSM modem (1-port) converter now attached to ttyUSB2
83805.696145] option 1-1:1.3: GSM modem (1-port) converter detected
83805.696612] usb 1-1: GSM modem (1-port) converter now attached to ttyUSB4
83805.812814] cdc_ether 1-1:1.4 eth0: register 'cdc_ether' at usb-0000:02:03.0-1, CDC Ethernet Device, 00:a0:c6:cf:2a:f0
```

3) List the ttyUSBx devices by “**ls -l /dev/ttyUSB***”.

The following files have been created in the /dev/ directory.



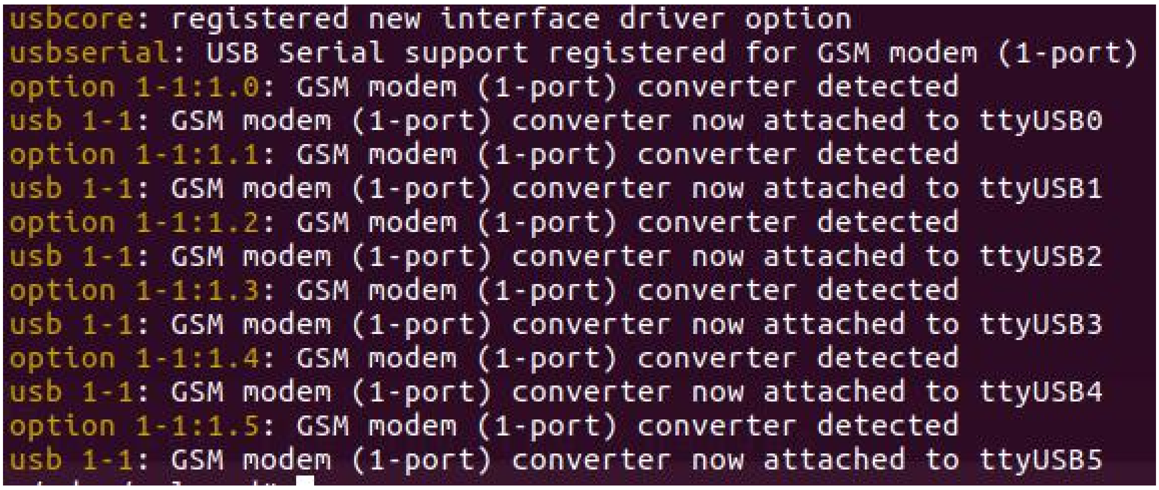
```
root@ubuntu:/etc/ppp# ls -l /dev/ttyUSB*
crw-rw---- 1 root dialout 188, 0 Nov 19 17:38 /dev/ttyUSB0
crw-rw---- 1 root dialout 188, 1 Nov 19 17:38 /dev/ttyUSB1
crw-rw---- 1 root dialout 188, 2 Nov 19 17:45 /dev/ttyUSB2
crw-rw---- 1 root dialout 188, 4 Nov 19 17:38 /dev/ttyUSB4
```

This information indicates device driver is installed successfully and the module was recognized by the PC device.

5.2 Verification of VID=0x1E0E and PID=0x9206

Now developer can verify if driver has been installed, when VID=0x1E0E, and PID=0x9206.

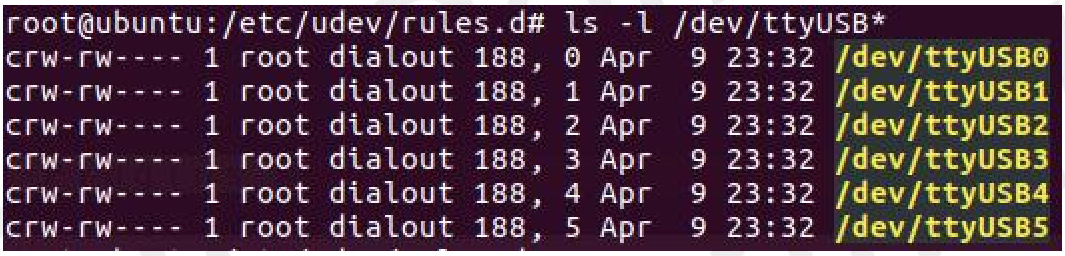
1) Connect physical USB interface of module and power on it.
2) Open Linux terminal and type the shell command “**dmesg”** to view kernel print information.



```
usbcore: registered new interface driver option
usbserial: USB Serial support registered for GSM modem (1-port)
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
option 1-1:1.5: GSM modem (1-port) converter detected
usb 1-1: GSM modem (1-port) converter now attached to ttyUSB5
```

3) List the ttyUSBx devices by “**ls -l /dev/ttyUSB***”.

The following files have been created in the /dev/ directory.



```
root@ubuntu:/etc/udev/rules.d# ls -l /dev/ttyUSB*
crw-rw---- 1 root dialout 188, 0 Apr  9 23:32 /dev/ttyUSB0
crw-rw---- 1 root dialout 188, 1 Apr  9 23:32 /dev/ttyUSB1
crw-rw---- 1 root dialout 188, 2 Apr  9 23:32 /dev/ttyUSB2
crw-rw---- 1 root dialout 188, 3 Apr  9 23:32 /dev/ttyUSB3
crw-rw---- 1 root dialout 188, 4 Apr  9 23:32 /dev/ttyUSB4
crw-rw---- 1 root dialout 188, 5 Apr  9 23:32 /dev/ttyUSB5
```

This information indicates device driver is installed successfully and the module was recognized by the PC device.

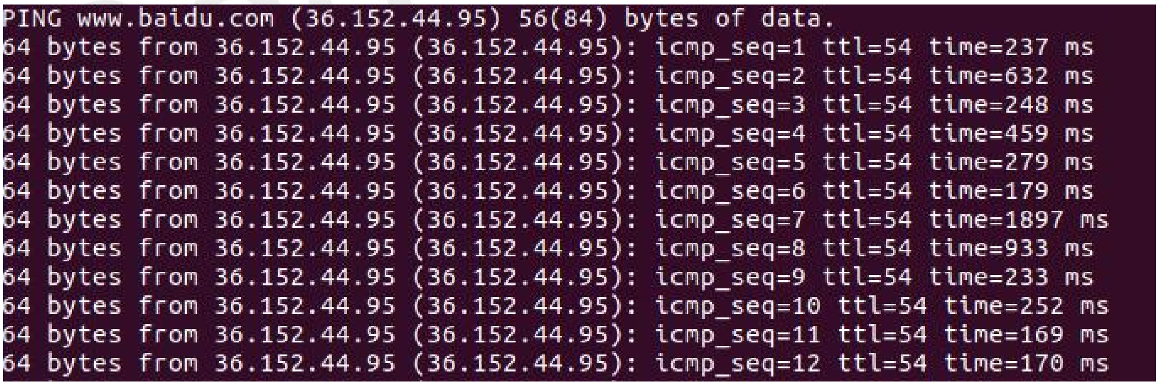
6 Usage of ECM

The MDM9205 only supports ECM. When ECM is used, VID should be 0x1E0E and PID should be 0x9205. The following way is to enable ECM.

```
//Example of enable ECM

AT+CUSBSELNV=1                              //Configure VID=0x1E0E, and PID=0x9205
OK
AT+SECMEN=1                                 //Enable ECM auto connecting.
OK
AT+SECMAUTH=<apn>[,<authType>,<userna       //set APN and auth
me>,<password>]                             authType(0-3)
OK                                          0  noauth
                                            1   pap
                                            2   chap
                                            3    both_pap_chap
AT+CREBOOT                                  //Reboot the module
OK
```

After setting the AT command and waiting for the module registering the network, you can use ping to check the connection.



```
PING www.baidu.com (36.152.44.95) 56(84) bytes of data.
64 bytes from 36.152.44.95 (36.152.44.95): icmp_seq=1 ttl=54 time=237 ms
64 bytes from 36.152.44.95 (36.152.44.95): icmp_seq=2 ttl=54 time=632 ms
64 bytes from 36.152.44.95 (36.152.44.95): icmp_seq=3 ttl=54 time=248 ms
64 bytes from 36.152.44.95 (36.152.44.95): icmp_seq=4 ttl=54 time=459 ms
64 bytes from 36.152.44.95 (36.152.44.95): icmp_seq=5 ttl=54 time=279 ms
64 bytes from 36.152.44.95 (36.152.44.95): icmp_seq=6 ttl=54 time=179 ms
64 bytes from 36.152.44.95 (36.152.44.95): icmp_seq=7 ttl=54 time=1897 ms
64 bytes from 36.152.44.95 (36.152.44.95): icmp_seq=8 ttl=54 time=933 ms
64 bytes from 36.152.44.95 (36.152.44.95): icmp_seq=9 ttl=54 time=233 ms
64 bytes from 36.152.44.95 (36.152.44.95): icmp_seq=10 ttl=54 time=252 ms
64 bytes from 36.152.44.95 (36.152.44.95): icmp_seq=11 ttl=54 time=169 ms
64 bytes from 36.152.44.95 (36.152.44.95): icmp_seq=12 ttl=54 time=170 ms
```

7 PPPD Call

7.1 Configure PPP protocol

Since PPP dialing requires the chat and PPPD command, you need to download the PPP protocol. Confirm that the following files already exist in Linux file system. If not, you need to apt-get install PPP.

```
/etc/ppp/chap-secrets
/etc/ppp/pap-secrets
/etc/ppp/ip-up
/etc/ppp/ip-down
/etc/ppp/peer/
```

7.2 Configure PPPD

Create 2 files under the file system /etc/ppp directory as follows: sim7080option and sim7080-chat.dat

7.2.1 Create option file

Create the new file for PPPD tty option **/etc/ppp/peer/sim7080option**.
/dev/ttyUSB3 will be changeable according your module ports.

The sim7080option file contents are as follows：

```
#/etc/ppp/peers/ sim7080option
# This is pppd script for China Mobile, used SIMCOM Module
/dev/ttyUSB3
#/dev/pts/11
115200
nocrtscts
noauth
connect '/usr/sbin/chat -v -s -f /etc/ppp/sim7080-chat.dat'
disconnect '/usr/sbin/chat -e -v "" +++ath'
debug
ipcp-accept-local
```

```
ipcp-accept-remote
usepeerdns
defaultroute
lcp-echo-failure 3
lcp-echo-interval 2
#asyncmap ffffffff
#idle 480
```

7.2.2 Create chat file

Create the chat file for PPPD chat **/etc/ppp/sim7080-chat.dat** .
The Chat program is used to establish a connection between the local PPPD and the remote PPPD program.

AT+CGDCONT sets the APN of the module, ibox.tim.it will be changeable according your modem APN.
The sim7080-chat.dat file contents are as follows：

```
#/etc/ppp/sim7080-chat.dat
ABORT           'NO CARRIER'
ABORT           'NO DIALTONE'
ABORT           'ERROR'
ABORT           'NO ANSWER'
ABORT           'BUSY'
TIMEOUT          120
''              AT
OK              ATE1
OK              AT+CGDCONT=1,"IPV4V6","ibox.tim.it"
OK              ATD*99#
CONNECT           ''
```

7.3 PPPD call

1. Before dial-up, please shut down the eth0 .
2. Script file gives execute permission and executes with root privileges
3. Dial by the following command: **pppd call sim7080option**
4. Execution results, you can view file /tmp/ppp.log.

```
root@ubuntu:/# pppd call sim7080option debug logfile /tmp/ppp.log
abort on (NO CARRIER)
abort on (NO DIALTONE)
abort on (ERROR)
abort on (NO ANSWER)
abort on (BUSY)
timeout set to 120 seconds
send (AT^M)
expect (OK)
^M
OK
  -- got it

send (ATE1^M)
expect (OK)
^M
^M
OK
  -- got it

send (AT+CGDCONT=1,"IPV4V6","ibox.tim.it"^M)
expect (OK)
^M
AT+CGDCONT=1,"IPV4V6","ibox.tim.it"^M^M
OK
  -- got it

send (ATD*99#^M)
expect (CONNECT)
^M
ATD*99#^M^M
CONNECT
  -- got it

send (^M)
Script /usr/sbin/chat -v -s -f /etc/ppp/sim7000-chat.dat finished (pid 12254), status = 0x0
Serial connection established.
using channel 5
Using interface ppp0
Connect: ppp0 <--> /dev/ttyUSB4
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0xfa37b19e> <pcomp> <accomp>]
rcvd [LCP ConfReq id=0x0 <asyncmap 0x0> <auth chap MD5> <magic 0xc5bc7416> <pcomp> <accomp>]
sent [LCP ConfNak id=0x0 <auth pap>]
rcvd [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0xfa37b19e> <pcomp> <accomp>]
rcvd [LCP ConfReq id=0x1 <asyncmap 0x0> <auth pap> <magic 0xc5bc7416> <pcomp> <accomp>]
sent [LCP ConfAck id=0x1 <asyncmap 0x0> <auth pap> <magic 0xc5bc7416> <pcomp> <accomp>]
sent [LCP EchoReq id=0x0 magic=0xfa37b19e]
sent [PAP AuthReq id=0x1 user="ubuntu" password=<hidden>]
rcvd [LCP DiscReq id=0x2 magic=0xc5bc7416]
rcvd [LCP EchoRep id=0x0 magic=0xc5bc7416 fa 37 b1 9e]
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=0x3 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
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 [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 100.70.197.238> <ms-dns1 211.136.17.107> <ms-dns2 211.136.20.203>]
sent [IPCP ConfReq id=0x3 <addr 100.70.197.238> <ms-dns1 211.136.17.107> <ms-dns2 211.136.20.203>]
rcvd [IPCP ConfAck id=0x3 <addr 100.70.197.238> <ms-dns1 211.136.17.107> <ms-dns2 211.136.20.203>]
Could not determine remote IP address: defaulting to 10.64.64.64
not replacing default route to ens33 [172.21.254.254]
local   IP address 100.70.197.238
remote IP address 10.64.64.64
primary    DNS address 211.136.17.107
secondary DNS address 211.136.20.203
Script /etc/ppp/ip-up started (pid 12270)
Script /etc/ppp/ip-up finished (pid 12270), status = 0x0
rcvd [IPCP ConfReq id=0x2]
Connect time 0.1 minutes.
Sent 0 bytes, received 14 bytes.
Script /etc/ppp/ip-down started (pid 12315)
sent [IPCP ConfReq id=0x4 <compress VJ 0f 01> <addr 100.70.197.238> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
sent [IPCP ConfNak id=0x2 <addr 0.0.0.0>]
rcvd [IPCP ConfRej id=0x1 <compress VJ 0f 01>]
rcvd [IPCP ConfRej id=0x4 <compress VJ 0f 01>]
sent [IPCP ConfReq id=0x5 <addr 100.70.197.238> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
rcvd [IPCP ConfReq id=0x3]
sent [IPCP ConfAck id=0x3]
rcvd [IPCP ConfNak id=0x5 <ms-dns1 211.136.17.107> <ms-dns2 211.136.20.203>]
sent [IPCP ConfReq id=0x6 <addr 100.70.197.238> <ms-dns1 211.136.17.107> <ms-dns2 211.136.20.203>]
rcvd [IPCP ConfAck id=0x6 <addr 100.70.197.238> <ms-dns1 211.136.17.107> <ms-dns2 211.136.20.203>]
Could not determine remote IP address: defaulting to 10.64.64.64
not replacing default route to ens33 [172.21.254.254]
local   IP address 100.70.197.238
remote IP address 10.64.64.64
primary    DNS address 211.136.17.107
secondary DNS address 211.136.20.203
Script /etc/ppp/ip-down finished (pid 12315), status = 0x0
Script /etc/ppp/ip-up started (pid 12361)
Script /etc/ppp/ip-up finished (pid 12361), status = 0x0
sent [LCP EchoReq id=0x1 magic=0xfa37b19e]
rcvd [LCP EchoRep id=0x1 magic=0xc5bc7416 fa 37 b1 9e]
sent [LCP EchoReq id=0x2 magic=0xfa37b19e]
rcvd [LCP EchoRep id=0x2 magic=0xc5bc7416 fa 37 b1 9e]
```