



MeiG Linux PPP Dial-up Guide

Controlled file name: Meige Linux PPP dial-up guide

Controlled version number: V1.1

Issuer: Meige Intelligent Technology Co., Ltd.

Release date: November 19, 2016





Important Notice

Copyright Notice

Copyright: Meige Intelligent Technology Co., Ltd.

This document and all its contents are owned by Meige Intelligent Technology Co., Ltd. and are protected by Chinese laws and applicable international conventions on copyright. Without the written authorization of Fangge Precision Devices Co., Ltd., no one may copy, disseminate, distribute, modify or otherwise use part or all of the contents of this document in any form. Violators will be held accountable according to law.

Disclaimer

MeiG Intelligent Technology Co., Ltd. makes no representations or warranties, express or implied, regarding any content in this document and assumes no liability for merchantability or fitness for a particular purpose or any indirect, special or consequential damages.

Confidentiality Statement

The information contained in this document (including any attachments) is confidential information. The recipient understands that the document is confidential and cannot be used for any purpose other than the specified purpose, nor can it be disclosed to any third party.

Disclaimer

The company does not assume any liability for property or personal injury caused by abnormal operation of the customer. Please develop corresponding products according to the technical specifications and reference designs in the manual. Before the announcement, the company has the right to change the contents of this manual according to the needs of technological development, and the changed version will not be notified separately.



Table of contents

1. Introduction.....	4
2. Product Support.....	4
2.1 Supported Products.....	4
2.2 Supported Features.....	4
3. Application Mode.....	4
4. PPP connection establishment process.....	5
4.1 Establishing a PPP Connection.....	5
5. PPP related configuration and dial-up.....	6
5.1 PPP Configuration.....	6
5.2 USB-MODERM Configuration.....	7
5.3 Adding devices in the Linux kernel.....	8
5.4 Module Loading Check.....	8
5.5 PPP Dial-up Script.....	9
6. Serial port debugging at command under Linux.....	10

1. Introduction

This document briefly introduces the PPP (point to point protocol) function of MeiG's standard module.

It includes the PPP installation process, the PPP connection process, and a PPP dial-up example.

2. Product Support

2.1 Supported Products

product	Support
SML630	Yes
SML730	Yes
SML750	Yes

2.2 Support functions

Support functions	Support
Data Services	Yes
Language Business	Yes
SMS service	Yes

3. Application Mode

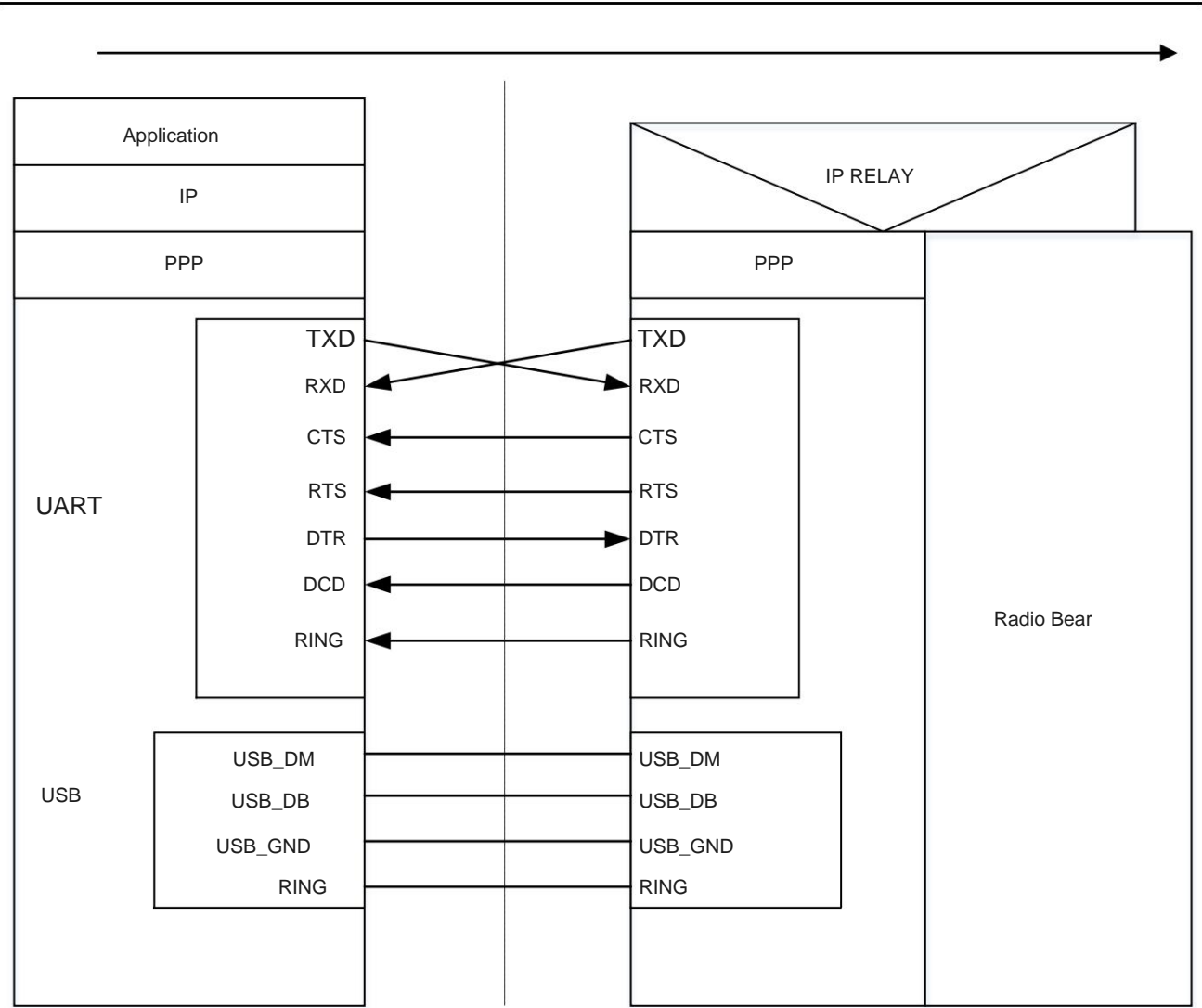


Figure 1 PPP application model

The above describes the application mode of PPP. For PPP connection, you can use the URAT interface or the USB interface.

The module side provides PPP Server function, the application side provides PPP client function, and the application side also needs to provide TCP/IP, Http protocol, etc. When the PPP connection is established, the IP data packets on the application side can be forwarded to the network through the module.

4. PPP connection establishment process

This section gives the recommended process for establishing a PPP connection for the MeiG module. If you need to develop a PPP application,

Please read this section before posting.

4.1 Establishing a PPP Connection

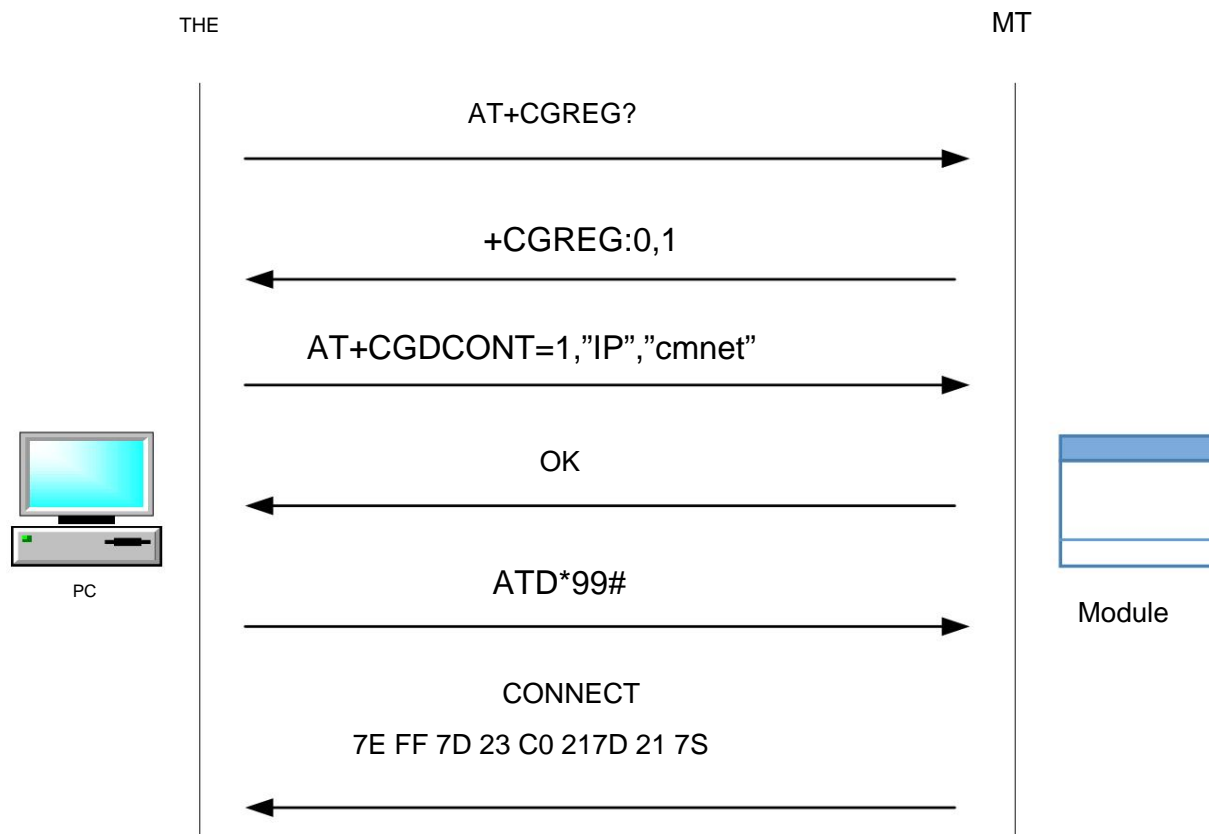


Figure 2 General process of establishing a PPP connection

After the module is registered to the GPRS network, configure the apn through `AT+CGDCONT` and use `ATD*99#` for PPP

Link is established. When `ATD*99#` is executed, the module enters the PPP data packet interaction process.

The packet exchange process is based on the standard Point to Point Protocol. More information about Point to Point Protocol can be found in

For more information, please refer to RFC 1661.

5. PPP related configuration and dial-up

5.1 PPP Configuration

The Linux kernel needs to be configured to support PPP. Follow the steps below to configure the PPP options.

first step:

`cd <kernel directory>`

Step 2:

make menuconfig

Step 3:

Device Drivers --->

[*] Network device support --->

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

[^{*}] PPP filtering

[^{*}] PPP multilink support

<*> PPP support for async serial ports

<*> PPP support for sync tty ports

<*> PPP Deflate compression

5.2 USB-MODERM Configuration

The Linux kernel needs to be configured to support USB-modem functionality. Follow the steps below to configure the usb-modem option:

first step:

cd <kernel directory>

Step 2:

make menuconfig

Step 3:

Device Drivers --->

[*] USB support --->

<*> USB Serial Converter support --->

[*] USB Generic Serial Driver

<*> USB drivers for GSM and CDMA modems

<*> USB Quatech Serial Driver for USB 2 devices

5.3 Adding devices to the LINUX kernel

Modify the drivers/usb/serial/option.c file and add the MeiG device's PID=0x9025, VID=0x0c56. Or

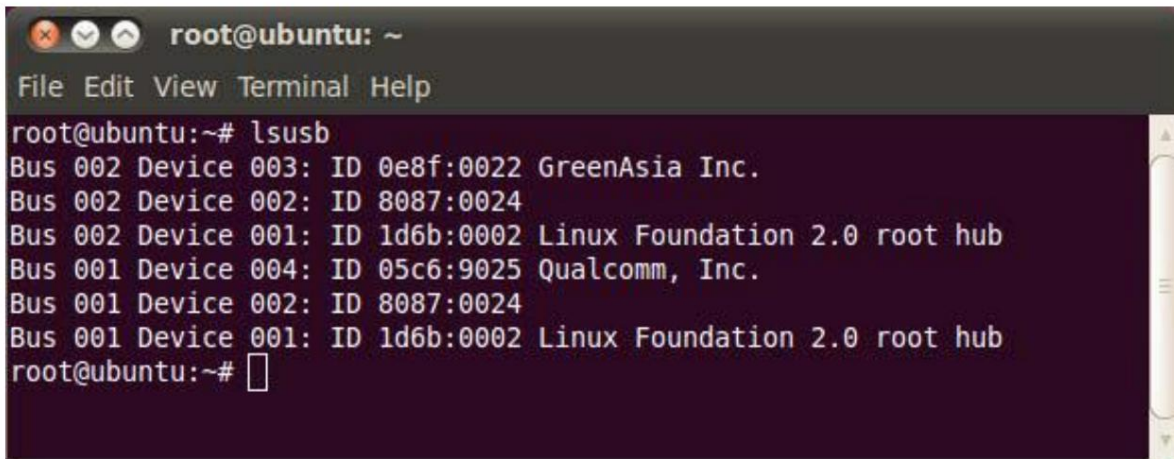
Person

PID=0xf601 VID=0x0c56

5.4 Module loading check

Turn on the host computer, power on the module, and check the device status through the following command:

root@ubuntu:~# lsusb



```
root@ubuntu: ~
File Edit View Terminal Help
root@ubuntu:~# lsusb
Bus 002 Device 003: ID 0e8f:0022 GreenAsia Inc.
Bus 002 Device 002: ID 8087:0024
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 004: ID 05c6:9025 Qualcomm, Inc.
Bus 001 Device 002: ID 8087:0024
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
root@ubuntu:~#
```

Possible returns depend on the status of the client's USB device: Note that the red font below is the MeiG module:

ID 05c6:9025 Qualcomm, Inc. or ID 05c6:non-01 Qualcomm, Inc.

Use the following command to query the USB driver loading status:

root@ubuntu:/dev# ls ttyUSB*

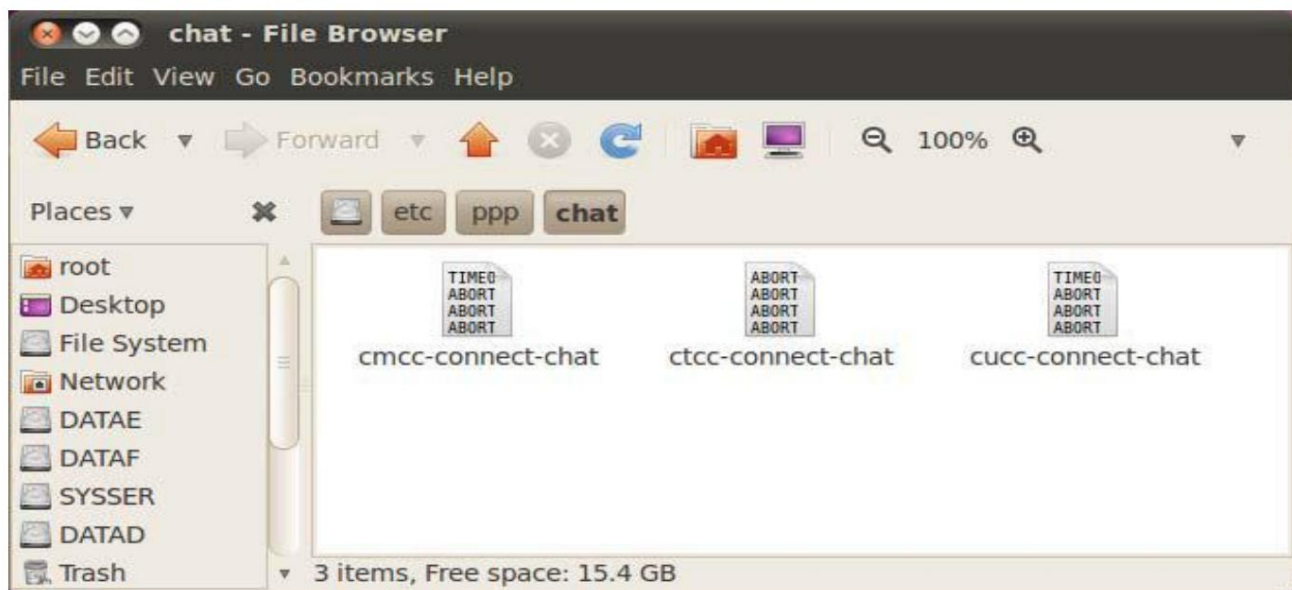

```
root@ubuntu: /dev
File Edit View Terminal Help
root@ubuntu:~# cd /dev
root@ubuntu:/dev# ls ttyUSB*
ls: cannot access ttyUSB*: No such file or directory
root@ubuntu:/dev# modprobe usbserial vendor=0x05c6 product=0x9025
root@ubuntu:/dev# ls ttyUSB*
ttyUSB0 ttyUSB1 ttyUSB2 ttyUSB3 ttyUSB4 ttyUSB5
root@ubuntu:/dev#
```

Return, there are 6 ttyUSB devices, ttyUSB1 is the MODEM port, ttyUSB2 is the AT port,

Customers do not need to pay attention to the remaining ports at present.

5.5 PPP dial-up script

ÿ Copy the contents of the attached PPP directory into the etc\ppp directory of the Linux system





• PPP dial-up command

Command: `pppd call cmcc`

Note: Take China Mobile's SIM card as an example:

• Verify dialing success

1) Check if the `ppp0` device is assigned an IP address through `ifconfig`

2) Command: `root@ubuntu:~ ping www.baidu.com`

3) Access the network through a browser

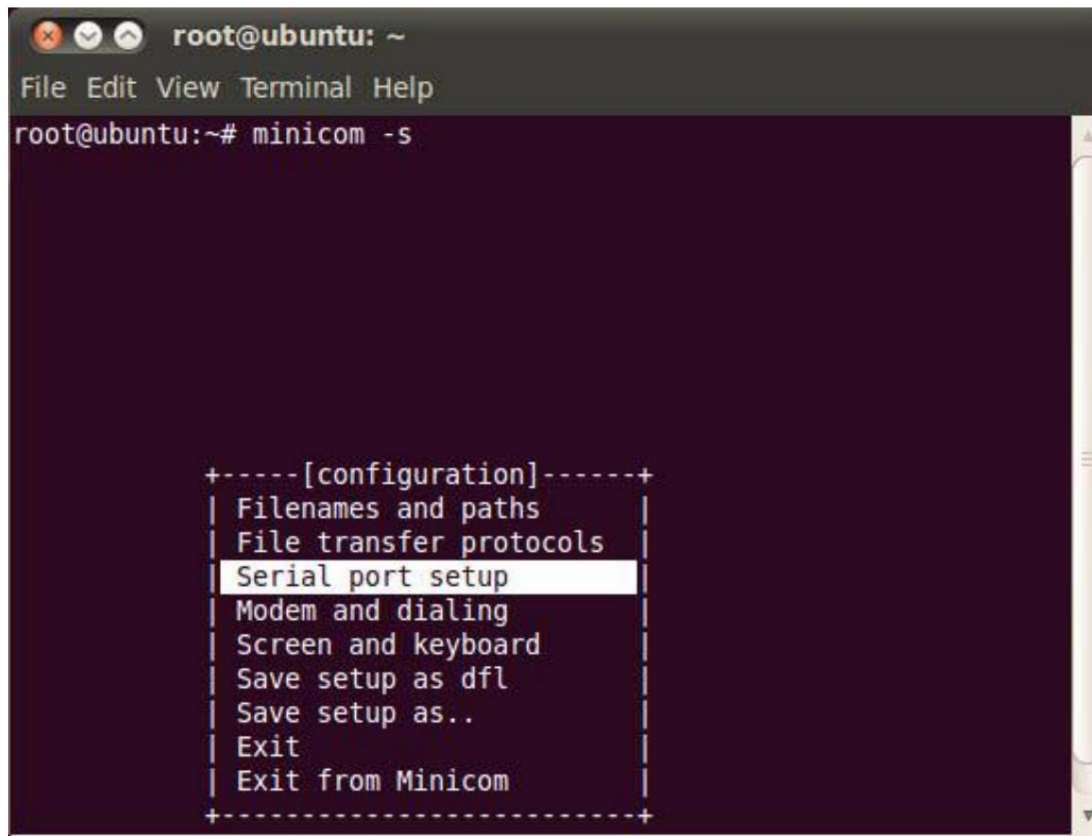
• Hang up the PPP connection

Command: `root@ubuntu:~ killall pppd`

6. Serial port debugging AT commands under LINUX

Configure the serial port through the following command:

Command: `root@ubuntu:~#minicom -s`

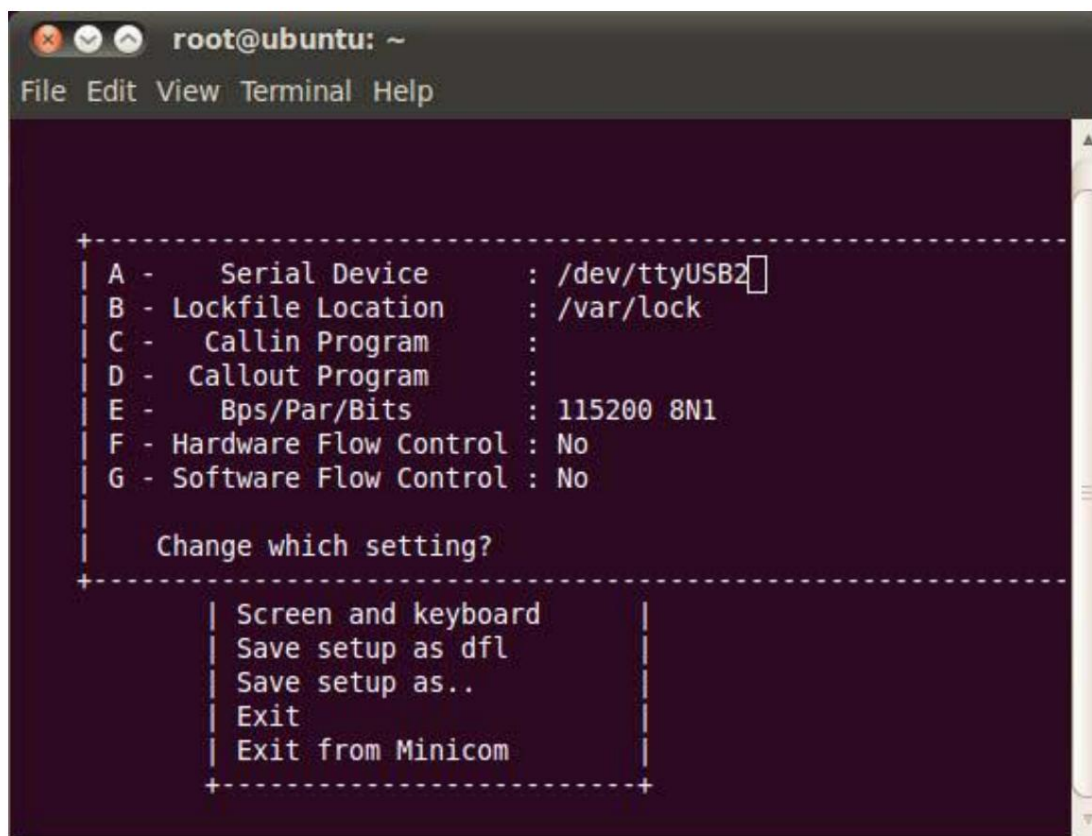


```
root@ubuntu: ~
File Edit View Terminal Help
root@ubuntu:~# minicom -s

+-----[configuration]-----+
| Filenames and paths          |
| File transfer protocols      |
| Serial port setup            |
| Modem and dialing            |
| Screen and keyboard          |
| Save setup as dfl            |
| Save setup as..              |
| Exit                         |
| Exit from Minicom            |
+-----+


```

Select Serial port setup



```
root@ubuntu: ~
File Edit View Terminal Help

+-----+
| A -   Serial Device           : /dev/ttyUSB2 |
| B -   Lockfile Location       : /var/lock    |
| C -   Callin Program          :              |
| D -   Callout Program         :              |
| E -   Bps/Par/Bits            : 115200 8N1   |
| F -   Hardware Flow Control   : No          |
| G -   Software Flow Control   : No          |
+-----+

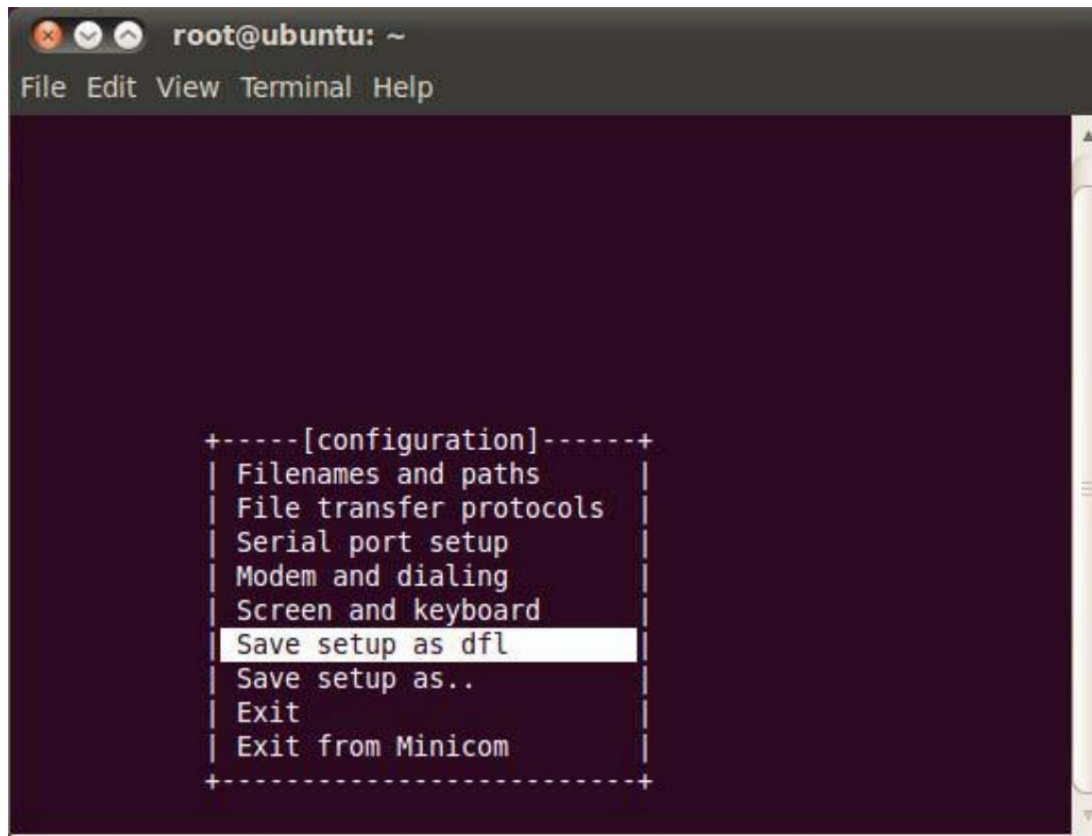
Change which setting?

+-----+
| Screen and keyboard          |
| Save setup as dfl            |
| Save setup as..              |
| Exit                         |
| Exit from Minicom            |
+-----+

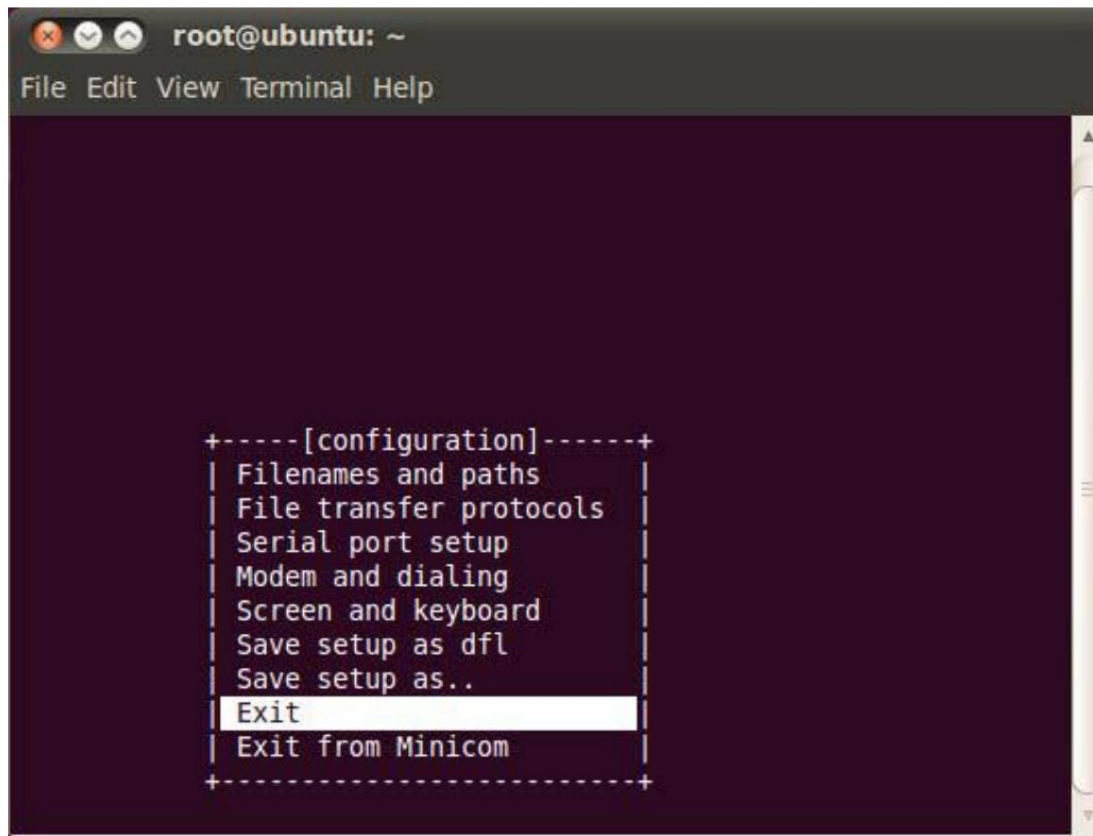

```

Serial Device /dev/ttyUSB2 —This is the AT port

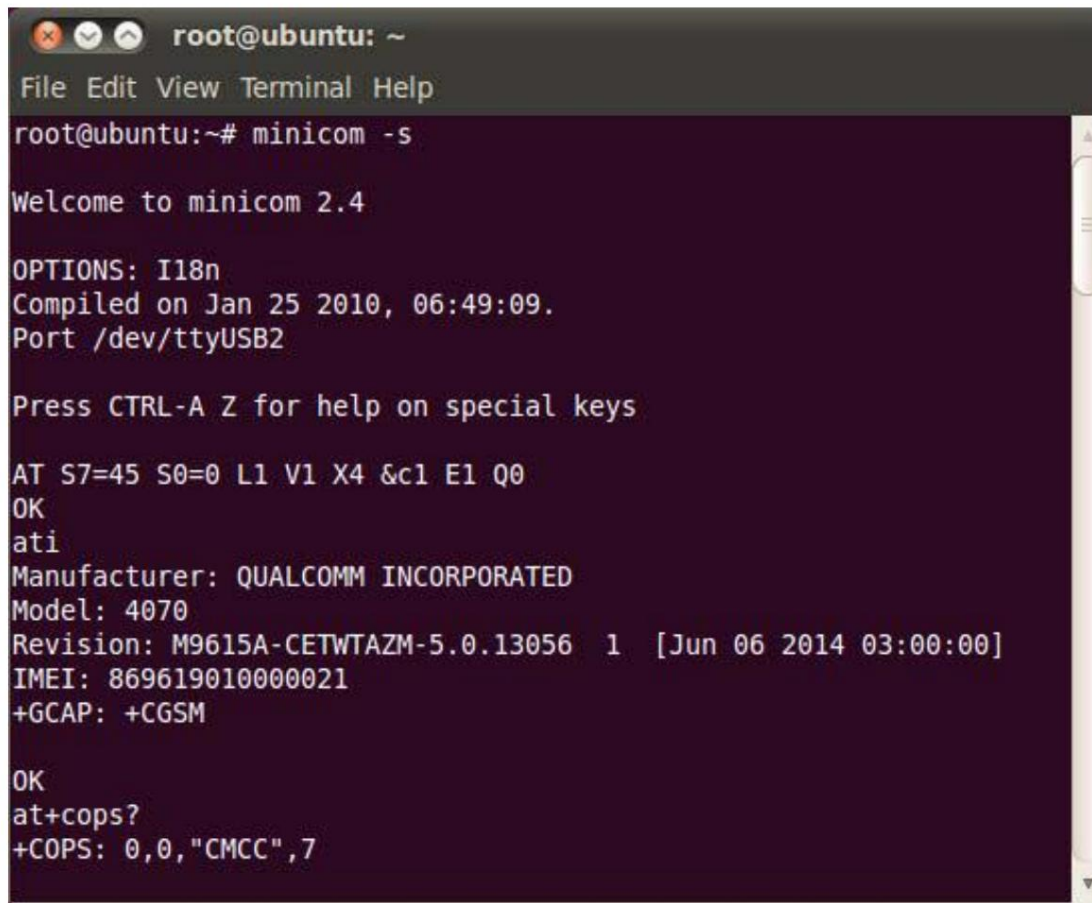
After the modification is completed, save it.



Exit



Now you can start AT debugging:

A terminal window titled 'root@ubuntu: ~' with a menu bar (File, Edit, View, Terminal, Help). The user has entered 'minicom -s'. The output shows 'Welcome to minicom 2.4', 'OPTIONS: I18n', 'Compiled on Jan 25 2010, 06:49:09.', 'Port /dev/ttyUSB2', and a prompt to 'Press CTRL-A Z for help on special keys'. The user enters 'AT S7=45 S0=0 L1 V1 X4 &c1 E1 Q0', receiving 'OK' and 'ati'. The user then enters 'at+cops?', receiving '+COPS: 0,0,"CMCC",7'.

```
root@ubuntu: ~
File Edit View Terminal Help
root@ubuntu:~# minicom -s

Welcome to minicom 2.4

OPTIONS: I18n
Compiled on Jan 25 2010, 06:49:09.
Port /dev/ttyUSB2

Press CTRL-A Z for help on special keys

AT S7=45 S0=0 L1 V1 X4 &c1 E1 Q0
OK
ati
Manufacturer: QUALCOMM INCORPORATED
Model: 4070
Revision: M9615A-CETWTAZM-5.0.13056 1 [Jun 06 2014 03:00:00]
IMEI: 869619010000021
+GCAP: +CGSM

OK
at+cops?
+COPS: 0,0,"CMCC",7
```