

USB 3G Modem

From ArchWiki

A number of mobile telephone networks around the world offer mobile internet connections over UMTS (or EDGE or GSM) using a portable USB modem device.

Contents

- 1 Remove the PIN
- 2 Device identification
- 3 Mode switching
- 4 Connection
 - 4.1 Network Manager
 - 4.2 wvdial
 - 4.3 netctl
 - 4.4 sakis3g
 - 4.5 Low connection speed
 - 4.5.1 QoS parameter
 - 4.5.2 Baud parameter
 - 4.6 Monitor used bandwidth
- 5 Reading SMS
 - 5.1 command line script
- 6 Fix image quality

Related articles

wvdial

Direct Modem
Connection

3G and GPRS modems
with pppd

Category:Modems

Remove the PIN

First of all use your SIM card in a normal phone and disable the PIN request if present. If the SIM card asks the PIN wvdial will not work.

Device identification

Examine the output of

```
$ lsusb
```

which will show the vendor and product IDs of the device. Note that some devices will show *two* different product IDs at different times as explained below.

Mode switching

Often these devices will have two modes (1) USB flash memory storage (2) USB Modem. The first mode, sometimes known as ZeroCD, is often used to deliver an internet communications program for another operating system and is generally of no interest to Linux users. Additionally some have a slot

into which the user can insert an additional flash memory card.

A useful utility for switching these devices into modem mode is `usb_modeswitch` (https://www.archlinux.org/packages/?name=usb_modeswitch), available in the official repositories.

Udev rules are supplied with the package in `/lib/udev/rules.d/40-usb_modeswitch.rules`. This contains entries for many devices, which it will switch to modem mode upon insertion.

When a device is switched, its product ID may change to a different value. The vendor ID will remain unchanged. This can be seen in the output of `lsusb`.

Some devices are supported in the USB serial kernel module called "option" (after the Option devices, but not limited to just those) and may be used without `usb_modeswitch`.

Udev itself included a utility called `/lib/udev/modem-modeswitch`. In udev 157 this was renamed to `/lib/udev/mobile-action-modeswitch` and morphed into a tool that only switches Mobile Action cables. For other devices use `usb_modeswitch`.

Note: You can find an alternative way to do this base on eject command [here](#).

Connection

Network Manager

After installing `usbutils` (<https://www.archlinux.org/packages/?name=usbutils>) and `usb_modeswitch` (https://www.archlinux.org/packages/?name=usb_modeswitch), you just need to install `modemmanager` (<https://www.archlinux.org/packages/?name=modemmanager>) to make the modem work with NetworkManager.

Make sure both NetworkManager and ModemManager services are running, see `systemd#Using units` for details.

Make sure `mobile-broadband-provider-info` (<https://www.archlinux.org/packages/?name=mobile-broadband-provider-info>) and `nm-connection-editor` (<https://www.archlinux.org/packages/?name=nm-connection-editor>) are installed.

A system restart might be necessary for ModemManager to detect the USB modem. After you restart the NetworkManager-applet and plug the modem in again NetworkManager should recognize the modem in the menu without further configuration. Setting up the modem in NetworkManager is self-explanatory, you should only need the login-information provided by your network provider.

Whilst running ModemManager `gammu` will not work. SMS and Usd codes can be still used with `modem-manager-gui` (<https://www.archlinux.org/packages/?name=modem-manager-gui>).

wvdial

See main article: `wvdial`

netctl

Netctl can be used to establish a connection using a USB modem. An example configuration file provided by netctl (<https://www.archlinux.org/packages/?name=netctl>) is located at `/etc/netctl/examples/mobile_ppp`.

See the netctl article for more information.

sakis3g

There may be the chance that the modem stick is supported by sakis3g (<http://www.sakis3g.com/>) which is an all in one command line script and automates all the steps above. Install sakis3g (<https://aur.archlinux.org/packages/sakis3g/>)^{AUR} from the AUR.

Low connection speed

Someone claims that the connection speed under Linux is lower than Windows: [\[\[1\]\]](https://bbs.archlinux.org/viewtopic.php?id=111513) (<https://bbs.archlinux.org/viewtopic.php?id=111513>)

A short summary for possible solutions which are not fully verified. In most of conditions, the low speed is caused by bad receiver signals and too many people in cell. But you still could use the following method to try to improve the connection speed.

QoS parameter

The `AT+CGEQMIN` and `AT+CGEQRQ` commands can be used to set the QoS. It is also possible to decrease and limit the connect speed. Add the following `Init` command in `/etc/wvdial.conf`:

```
Init6 = AT+CGEQMIN=1,4,64,640,64,640
Init7 = AT+CGEQRQ=1,4,64,640,64,640
```

Baud parameter

Baud parameter in `wvdial.conf` could be used to increase the connection speed.

```
Baud = 460800
```

But the official Huawei E261 windows application set the Baud=9600 under Windows Vista. More verifications are needed to double check this point.

Monitor used bandwidth

Frequently a 3G connection obtained via a mobile phone operator comes with restricted bandwidth, so that you are only allowed to use a certain bandwidth per time (e.g. 1GB per month). While it is quite straight-forward to know which type of network applications are pretty bandwidth extensive (e.g. video streaming, gaming, torrent, etc.), it may be difficult to keep an overview about overall consumed bandwidth.

A number of tools are available to help with that. Two console tools are `vnstat` (<https://www.archlinux.org/packages/?name=vnstat>), which allows to keep track of bandwidth over time, and `iftop` (<https://www.archlinux.org/packages/?name=iftop>) to monitor bandwidth of

individual sessions. If you are a KDE user, knemo (<https://www.archlinux.org/packages/?name=knemo>) might help. All are available in the official repositories.

Reading SMS

This was tested on a Huawei EM770W (GTM382E) 3g card integrated into an Acer Aspire AS3810TG laptop.

```
$ pacman -S gnokii
$ mkdir -p $XDG_CONFIG_HOME/gnokii
```

Usually the configuration directory is `~/.config/gnokii`.

```
$ cp /etc/gnokiirc ~/.config/gnokii/config
```

edit `~/.config/gnokii/config` as follows:

```
port = /dev/ttyUSB0
```

You may have to use a different port depending on your configuration, for example `/dev/ttyUSB1` or something else:

```
model = AT
connection = serial
```

You need to be part of the `uucp` group to use `/dev/ttyUSB0`, for example if your user is called "x":

```
# gpasswd -a x uucp
$ newgrp uucp
```

The `newgrp` command allows you to take advantage of the new group assignment immediately without having to logout/login.

Then launch `gnokii`:

```
$ xgnokii
```

Click on the "SMS" icon button, a window opens up. Then click: "messages->activate sms reading". Your messages will show up in the window.

command line script

A small command line script using `gnokii` to read SMS on your SIM card (not phone memory) without having to start a GUI:

```
$ gnokii --getsms SM 0 end 2>&1|grep Text -A1 -B3|grep -v Text
```

What it does:

```
gnokii # invoke gnokii
--getsms SM 0 end # read SMS from SM-memory location (=SIM card) starting at 0 and reading all occupied memory loca
2>&1 # connect STDERR to STDOUT to make sure the output from the --getsms command can be piped to grep
|grep Text # pipe output from gnokii to grep, anchoring at output containing "Text"
-A1 -B3 # print one line after the matched pattern and three lines before the matched pattern
|grep -v Text # grep result to another grep to exclude the "Text" line (-v for inverting the pattern)
```

Granted this does not work very well if your SMS contains the word "Text", but you may adapt the script to your liking.

Fix image quality

If you are getting low quality images while browsing the web over a mobile broadband connection with the hints "shift+r improves the quality of this image" and "shift+a improves the quality of all images on this page", follow these instructions:

Install `tinypoxy` (<https://www.archlinux.org/packages/?name=tinypoxy>), available in the official repositories.

Edit `/etc/tinypoxy/tinypoxy.conf` and insert the following two lines:

```
AddHeader "Pragma" "No-Cache"
AddHeader "Cache-Control" "No-Cache"
```

Start `tinypoxy`:

```
systemctl start tinypoxy
```

Configure your browser to use `localhost:8888` as a proxy server and you are all done. This is especially useful if you are using, for example, Google Chrome which, unlike Firefox, does not allow you to modify the `Pragma` and `Cache-Control` headers.

Retrieved from "https://wiki.archlinux.org/index.php?title=USB_3G_Modem&oldid=412194"

Category: Modems

-
- This page was last modified on 13 December 2015, at 21:22.
 - Content is available under GNU Free Documentation License 1.3 or later unless otherwise noted.