Linmodem−HOWTO

Table of Contents

Table of Contents

Linmodem−HOWTO

Sean Walbran `sean(at)walbran.org` and Marvin Stodolsky `stodolsk(at)erols.com`

v0.43, 24 February 2001

This document describes Linmodem (winmodem hardware) support under Linux. While such support is limited (almost exclusively in the form of manufacturer−created, but unsupported, binary kernel modules), the number of chipsets with some form of support is growing rapidly.

---

1. Introduction

This is the Linux Linmodem HOWTO document. It is intended as a quick reference to help you find out if there is a way to get your (so−called) winmodem working under Linux, and, if so, how to do it. You should understand from the outset that there may well be *no* support for your winmodem: there is limited support for such modems, often in the form of vendor−created but vendor−unsupported, binary−only kernel modules (though a small number of open−source projects exist).

To emphasize: your best bet under Linux is *certainly* to get a true hardware modem. However, if you're stuck with a winmodem, perhaps this document can help.

For the most up−to−date information about available Linmodem drivers, visit Rob Clark's site, our small resources page, and the Linmodems.org mailing list archives. General modem issues, such as IRQ settings and dialup scripts, are dealt with much more thoroughly in the more general Modem−HOWTO, Serial−HOWTO, PPP−HOWTO, and other related HOWTOs available at the Linux Documentation Project site and elsewhere.

1.1 Copyright

distribute it provided that you:

* Send your derivative work (in the most suitable format such as sgml) to the LDP (Linux Documentation Project) or the like for posting on the Internet. If not the LDP, then let the LDP know where it is available.
* License the derivative work with this same license or use GPL. Include a copyright notice and at least a pointer to the license used.
* Give due credit to previous authors and major contributors.

If you're considering making a derived work other than a translation, it's requested that you discuss your plans with the current maintainer.

1.2 Disclaimer

Use the information in this document at your own risk. We disavow any potential liability for the contents of this document. Use of the concepts, examples, and/or other content of this document is entirely at your own risk.

All copyrights are owned by their owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark. In particular, since the term "Winmodem" is a trademark of US Robotics/3Com, we use the term "winmodem" here as does Rob Clark: to be read as "Winmodems(tm), host−based modems, HCF−modems, HSP−modems, and all similar modem−like hardware." Linux is a trademark of Linus Torvalds.

Naming of particular products or brands should not be seen as endorsements.

It are strongly recommended to make a backup of important and/or relevant files before any installation procedure.

1.3 Credits

Most individual credits are given in the body of the text where appropriate.

A large amount of information contained in this document comes a variety of great sources such as Rob Clark's site, the Linmodems.org, mailing lists, and Werner Heuser's Mobilix pages.

Special thanks to Mark Spieth (mark(at)digivation.com.au) for discussions, advice, and multiple and ongoing contributions.

This document itself was created using the SGML HOWTO template created by Stein Gojen, as described in the HOWTO−HOWTO. site.

1.4 Updates and Corrections

The most recent HTML version of this document is available at http://walbran.org/sean/linux/linmodem−howto.html, as a single HTML file at http://walbran.org/sean/linux/linmodem−howto−all.html, with source SGML at http://walbran.org/sean/linux/linmodem−howto.sgml.

The old version of this document was getting huge and unwieldy, so this rewrite seemed necessary. To avoid the total loss of that information, a copy of the old version is maintained at http://walbran.org/sean/linux/linmodem−howto−old.html.

Do you have a Linmodem which works, but is not described here? Are you developing a driver? Do you think something in this document is incorrect or misleading? Do you think that your or someone else's work has been used here but not appropriately credited? Please don't hesitate to email me at `sean(at)walbran.org` with corrections and suggestions.

---

2. Linmodems

2.1 What is a Linmodem?

A Linmodem is the Linux implementation of a "winmodem" (see disclaimer). These devices are 'less than' a modem in the sense that they depend on software to perform, to a greater or lesser extent, the functions traditionally handled by modem hardware. The rationale for this is, of course, that software is cheaper than hardware, and can be upgraded/expanded/improved without the use of screwdrivers (usually); however, for the modem to function at all, one requires software that can run on one's preferred operating system.

2.2 Which Linmodem hardware is supported?

An ever−growing number of winmodems will work under Linux. Each chipset for which a driver is known to exist has a section in this document, below, describing its installation. Any other chipset has *no known support* under Linux (at least, not known to us).

2.3 How can I find out if my GeeWhiz 9.8.7 Modem / Laptop has one of these chipsets?

Information from the system

The information about installed hardware using commands such as:

- PCI: `cat /proc/pci` and `lspci`
- ISA: `pnpdump` and `isapnp`
- Internal PCMCIA: `cardctl ident`
- General: `dmesg | more` and `cat /proc/interrupts`

MarvS notes that the Device Manager under Windows can provide similar information, but it should be noted that a manufacturer will often simply put its brand name on a built−in modem, so this information may not be as useful as you might hope (e.g., what chipset does a "Compaq Internal 56k" modem have?). Additional information may sometimes be obtained by making a modem log, implemented under MS Windows as a check box option within the Dial Up Networking menus. The file produced is `C:\WINDOWS\MODEM.LOG`. It will contain the modem initialization strings, and perhaps also the name of the modem configuration file, which may also contain other useful information.

Modem names and identification numbers

If you know the precise name of your modem, you can try searching the large Linux Modem Compatibility Database at Rob Clark's site. The color/letter code on the left side of the table will indicate if your modem is known to function or not under Linux. The code "LM" indicates a Linmodem, and the modem notes should indicate which driver you need. A "WM" means it's a winmodem, but no Linux support is known to exist. Be careful not to assume that modems with similar names will contain the same chipsets, or will necessarily behave similarly whatsoever! Your WhizBang LX56 and your friend's WhizBang GT56 could have entirely different innards.

If you do not know the precise name of your modem, you can search based on the identification number of the modem (on every modem there must be printed a registration number, which may either be the board producer's designation, or, alternatively, an FCC registration number. An example photo of such an ID number on a modem board can be found at http://www.idir.net/~gromitkc/fcc1.jpg on Rob Clark's site.) Use your web browser's "Find in Page" to search his table of modems and FCC ID's to obtain chipset/driver information. Alternatively, you can directly search the US Federal Communications Commission (FCC) database at http://www.fcc.gov/oet/fccid/. Read the directions carefully, and be careful not to confuse O (the letter) with 0 (the number), and other possible mixups.

Laptops with internal modems

You may not be able to obtain the FCC ID number if you have a laptop which you prefer not to open up, or are looking to buy a particular machine and the vendor has not been polite enough to provide you with the information nor a sample box for you to take apart and play with. In these cases, you might try:

- Kenneth Harker's Linux on Laptops site indexes a large number of user−created sites describing their experiences with Linux on particular laptop models.
- Werner Heuser's Mobilix: Linux Modems and Mobilix: Linux Mini−PCI pages include lists of specifications for laptops with internal modems and NIC's, as well as useful tips for obtaining more information in case the model is not listed there.
- The computer vendor's manual, web site, or (horrors!) technical support.
- http://www.google.com

---

3. General Setup and Kernel Module Issues

3.1 Kernel Module Support

All of the kernel drivers listed here are released as kernel modules; therefore, you must be sure to have a kernel which supports modules. In addition, "module version" support should be enabled to aid the use of kernels and modules which are not version matched, as described further below. If you use a kernel from a reasonably recent Linux distribution, such module support is most likely already enabled. If you're compiling the kernel yourself, then you should already be aware of how to enable modules, via the Kernel HOWTO. In any case, you can check to make sure that the following settings exist in your kernel configuration file (which is usually found under `/usr/src/linux`):

```
CONFIG_MODULES=y
CONFIG_MODVERSIONS=y
```

3.2 ISA Plug−n−Play

If you have an ISA Plug−n−Play modem, you will most likely need to use isapnptools to allocate resources to the modem card. For this, you need to have isapnptools installed and have an entry in the `/etc/isapnp.conf` file for the modem. You should read the manual pages and the Plug−and−Play−HOWTO, but if you have no other ISA devices you're concerned about, basically all you need to do is:

1. If possible, configure your BIOS to "Non−PNP OS."
2. As root, run `pnpdump` to generate a prototype isapnp.conf file based on probed cards and your system's current resource usage.
3. Look for your modem in this output, and uncomment the lines corresponding to the (otherwise unused) IRQ you wish to use for the modem. For example, Sean's `isapnp.conf` for a Thinkpad i1411 with a Lucent LT modem includes:

   ```
   (CONFIGURE ACRd119/1 (LD 0
     (INT 0 (IRQ 11 (MODE +E)))
     (IO 1 (SIZE 8) (BASE 0x0100) (CHECK))
     (NAME "ACRd119/1[0]{LT Win Modem        }")
   # (ACT Y)
   ))
   ```

   Strangely, in this case at least, it was necessary to leave the `#(ACT Y)` commented out. If it doesn't work for you one way, try it the other.
4. Copy the file to `/etc/isapnp.conf`
5. Reboot. You should see a message along the lines of 'Initializing ISA PNP devices...OK' on booting. If it fails, you have probably selected an IRQ/DMA setting which is already in use; try another of the options given in the `pnpdump` output. (Note that it is probably not necessary to reboot, if you run `isapnp` with the right flags. However, it's easiest for the beginner to simply reboot at this point.)

3.3 PCI Modems

If you wish to know more about your PCI modem than `cat /proc/pci` gives, utilities within the software package `pciutils` are useful, such as `scanpci` and `lspci`. In particular, `lspci −vv` gives lots of nice, useful information.

3.4 Module tools

The following commands are useful when dealing with modules. Many require root priveleges. See the manual pages (e.g., `man insmod`) for more detailed information on these commands.

insmod, insmod −f, and modprobe

A version−matched kernel module should usually be inserted using the command `modprobe` *module_name*; `modprobe` will try to insert any other modules on which your module depends (as determined by `depmod`, described below).

A single module can be inserted (without those modules on which it depends) using the command `insmod` *module_name*. If the module were compiled under a different kernel than the current one, `insmod` would report the version mismatch and refuse load the module. One can, however, pass a flag to force the module to load despite the mismatch: `insmod −f` *module_name*. If the kernel interface the module uses did not actually change with the kernel version, the module will be inserted and could be to some degree functional.

This is the case with, for example, the ESS modem module `esscom.o` which, while compiled under 2.2.12, can be forcibly inserted with later kernels and will function to a greater or lesser extent up through kernel version 2.2.14 without further changes; beyond 2.2.15, the patch to `tty.h` described below is required. However, even forcing insertion fails for kernels from the 2.4 series.

rmmod

A module can be unloaded (removed from the kernel) after use by issuing the command `rmmod`.

depmod

The `depmod` commands analyzes module dependencies. The compatility of precompiled modules with a running kernel can be checked with a command like:

```
depmod −e ltmodem.o
```

For the specific example of the ltmodem.o module compiled under kernel 2.2.12 with a running kernel 2.2.17, the returned information includes:

- depmod: *** Unresolved symbols in ltmodem.o
- depmod: bh_mask
- depmod: schedule_timeout
- depmod: request_region
- depmod: pcibios_read_co
- and many others.

Using a module with unresolved symbols can be a dangerous thing, as described below.

---

4. Tips and Tricks for Precompiled Modules with different Kernel Versions

Many of the linmodem drivers are only available as precompiled, binary kernel modules. Generally, modules/binaries transparently function only with the kernel against which they were co−compiled. Therefore, getting a precompiled linmodem driver to work with your particular kernel could be a challenge.

Since the Linux kernel is a dynamically changing beast, it is very unfortunate that many modem/chip vendors have not yet chosen to release source−code versions of their drivers, which would ensure your and our ability to modify these drivers appropriately as kernel source code evolves. Some of the binary modules have been coaxed to function under some later kernel versions using various tricks, as described below; however, even though a module may be rendered functional, it is advisable to use them minimally. Quoting an email from Mark Spieth,

"A driver can never work properly if there are unresolved symbols, as it means something is not going to work. Furthermore, it means that that something that would have been called will call something else in the kernel and this could be anything. This is *very* bad."

Therefore, you should be careful in using binary modules with a kernel of a different version; proceed at your own risk. If you require above all that your modem function, consider downgrading your kernel to match the module − this is by no means a ridiculous prospect. Despite these warnings, however, many others have used mismatched binary modules and kernels with only minor annoyances (such as the occasional kernel panic) using tricks and tools such as the following.

4.1 Fixscripting

Mark Spieth has contributed a progressively improved series of "fixscripts" for editing a binary module so that version mismatch warnings are eliminated. Insertion of the "fixed" module then proceeds without the forcing flag, i.e. simply `insmod` *module_name*. Later versions also rename module symbols to match those exported by the kernel, so that "Unresolved symbols" errors are not returned by the test `depmod -e`. It must be emphasized that this change is almost entirely cosmetic − it is still recommended that the module be used minimally.

To use the fixscript on, for example, the (now−deprecated) binary Lucent module ltmodem.o, make a working directory such as `/root/modem`. Obtain the latest fixscript from http://www.test.dclabs.com.au/linmodem/fixscript. Save the file as `fixscript`. View it with `less` or your favorite text editor to check that DOS hard stops were not accidentally acquired. They look like bold M, underlined M, or ^M depending upon your viewer/editor. NOTE: the viewer `more` does NOT display these DOS newlines.

Make the file executable with `chmod +x fixscript`. Generate a "fixed" module with, i.e.,

```
./fixscript ltmodem.o ltmodem2217.o
```

No errors should be generated by testing the module dependencies with

```
depmod −e ltmodem2217.o
```

and insertion should succeed with a simple, non−forced,

```
insmod ltmodem2217.o
```

The "source code" supplied with some PCTel modules (a small C file) performs similar masquerading when compiled and linked with the binary libraries in those packages; unlike the partially−open−source Lucent driver, it does *not* compensate for any actual changes to the kernel interface.

4.2 Patching `tty.h`

In his quest to get the original, binary−only Lucent LT modem driver (version 5.68) working with kernels later than 2.2.14, Mark Spieth noticed that one simple change in the Linux kernel source fixed the major incompatibilities incurred between the 2.2.14 to 2.2.16 kernel versions. This patch is no longer necessary when using the partial source/binary Lucent driver (version 5.78), but it remains useful for those with other modems whose drivers are compiled against pre−2.2.15 kernels.

The patched 2.2.17 `tty.h` and some 2.2.17 kernel packages compiled with this patch are available from http://walbran.org/sean/linux/stodolsk/. If you want to do the edit yourself, the line to shift is in the structure `tty_struct` within `include/linux/tty.h`; it has an extra member `poll_wait` in later kernels. Move this member to the bottom of the structure, so that the remaining offsets will then be the same as those in versions earlier than 2.2.15, and thus be compatible with the precompiled kernel module. You will need to recompile your kernel and modules after making this change to the source.

4.3 Using a `ppp.o` from Kernel 2.2.14

A trick exists for using the binary modules with kernels later than 2.2.15 which does not require kernel recompilation; however, following the discovery of the `tty.h` patch described above, this trick is no longer necessary nor recommended. The trick is to replace the `/lib/modules/net/ppp.o` module with one from kernel 2.2.14. Christoph Hebeisen (cth(at)sfu.ca) reported that the use of ppp.o version 2.2.14 rather than that of version 2.2.16 with the Lucent module provided functionality under 2.2.16 kernels. Willie Green (willjr(at)lcc.net) confirmed that this trick works also with the ESS module. After simple insertion of a supporting version−matched module:

```
insmod slhc
```

the mismatched ppp.o from 2.2.14 source is inserted

```
insmod −f ppp.o
```

We wish to emphasize that this trick with forced insertion is less stable than the easy and more effective change to the kernel source file `tty.h`, as described above.

---

5. Specific Chipsets and Their Drivers

5.1 IBM Mwave (Thinkpad 600E)

IBM has a completely open−source (GPL'ed) driver for the software modem in their Thinkpad 600E's available here.

5.2 Lucent LT

Overview

This modem enjoys the most support under Linux, in that there exist three different driver packages:

- There exists a manufacturer−unsupported, half−binary/half−open−sourced kernel module, originally designed for Red Hat 6.2's 2.2.14−5 kernel, but substantially reworked by Mark Spieth and others to function with 2.2.x and 2.4.x kernels. This is driver version 5.78(c,d,e,...), and is the driver you are most likely to have success using.
- There exists a manufacturer−unsupported, binary−only kernel module, compiled under Red Hat 6.0's 2.2.12−20 kernel. This is driver version 5.68.
- Some open source tools for use with Lucent modems are available at http://www.close.u−net.com/ltmodem.html. Pavel Machek writes that "It is not too useful, however: it is a hardware driver, and without a v.34 protocol stack, you can't connect to your ISP. It is enough to turn your Lucent winmodem into an answering machine, however."

It should be noted that the binary−only driver module (from "linux568.zip") contains code from the GPL'ed Linux `serial.c` driver, so, since the source code for the modem driver is not available, trafficking in this driver is apparently in violation of the GPL. Distributing the partially open source driver ("i56lvp578.zip") may or may not be technically legal, since the GPL'ed code, though not yet linked with the closed−source code, is certainly intended to be so. See this Kernel Traffic issue and a Linux−Kernel mailing list archive for the week including Dec. 3rd, 2000, for more details.

Driver v5.78(c,d,e,...) − Installation

You should obtain the most recent package for your kernel from http://walbran.org/sean/linux/stodolsk/ and follow the up−to−date instructions given there.

Manufacturer's driver version 5.68 − Installation

This driver is superseded by version 5.78, described above; however, it may still be of some use in special cases.

1. Obtain the package for your kernel:
    1. 2.2.12 to 2.2.15 http://linmodems.org/linux568.zip
    2. 2.2.15 and above : same URL, but "tty.h" patch is required; see "Tips and Tricks...", above.
2. `unzip linux568.zip`
3. `su` (enter root password when prompted)
4. `./ltinst` (a 'file not found' error will be issued due to a flaw in the installation script; ignore this error.)

Your modem should now be accessible as the device `/dev/modem` or `/dev/ttyS14`.

Open Source Tools − Installation

See the documentation with the source for instructions.

5.3 ESS

Overview

Binary−only drivers for ES56T−PI (PCI) and ES56V−I (ISA), compiled under RedHat 6.0's kernel 2.2.12−20, are available. The driver has been used via forced insertion up through kernel 2.2.15, and up through 2.2.17 using the "tty.h" patch described in the "Tips and Tricks..." section, above.

Installation

1. Obtain the package for your modem:
    - ISA: ftp://ftp.esstech.com/pub/modem/isa/unsupported/56v−i/linux/kernel61/linux111.zip or
    - PCI: ftp://ftp.esstech.com/pub/modem/pci/unsupported/56t−pi/linux/Kernel61/111.zip.
2. For kernel 2.2.15 and later, apply tty.h patch (See "Tips And Tricks...," above); recompile kernel and modules.
3. Unpack the package with: `unzip` *package_name*
4. Change to the root user: `su` (enter root password when prompted)
5. Create the device file: `mknod /dev/esscom c 127 1`
6. Make convenience device: `ln −s /dev/esscom /dev/modem`

7. Make convenience device: `ln −s /dev/esscom /dev/ttyS15`
8. Set device ownership: `chgrp uucp /dev/esscom`
9. Set device permissions: `chmod 666 /dev/esscom`
10. Masquerade module version (See "Tips And Tricks...," above): `./fixscript essmodem.o essmodem.fix.o`
11. Install module file: `cp essmodem.fix.o /lib/modules/`uname −r`/misc/essmodem.o`
12. Insert module in kernel: `insmod −f essmodem`
13. (Optional) Provide for automatic module loading: add a line "alias char−major−127 essmodem" to the file `/etc/modules.conf` or `/etc/conf.modules`

5.4 PCTel

Overview

Binary drivers can be found at http://www.idir.net/~gromitkc/winmodem.html#drivers.

A Debian−style installation package for kernel 2.2.16 was made available by Corel at ftp://ftp.corel.com/pub/linux/CorelLinux/dists/corellinux−1.2/corel/binary−i386/utils/pctel−kernel−2.2.16−driver−cdl−v1.0_1.0.deb, though this link appears to now be incorrect. A gzipped/tarred package derived from the .deb is available here. In addition, a driver for kernel 2.4 was contributed by Thomas Wright, and is also available here. Other packages, requiring the superficial compilation described below, are also known to exist.

Installation

There are apparently two types of PCTel module package around.

1. A package (rpm or deb) which installs two module files, pctel_hsp.o and pctel_pci.o, in `/lib/modules/2.2.16`.

   With such a package, if you are running a kernel more recent than 2.2.16, you will need to use forced insertion (insmod −f), and if you are not successful, might try the "fixscript" method used with the Lucent 5.68 and ESS modules above − but, note that this has not, to my knowledge, been tried out yet. If you are running a kernel older than 2.2.16, you should consider upgrading your kernel, or else try the fixscripting as well (this is also not guaranteed to work). Please send me a report if you get these to work.

2. A package which, when unpacked, gives a set of libraries (hsp.a, etc...) and a small C source file (ptmodule.c), which should be in directories like lib/ and src/module/. If there are no directories, create them and arrange the files with:

   ```
   mkdir lib
   mkdir src
   mkdir src/module
   mv *.a lib/
   mv Makefile *.c src/module
   ```

   Now go to the directory src/module and type `make`. This should generate the module file `pctel.o`, which will appear back up in the directory `lib`. (The driver module is *not* the object file `ptmodule.o` in `src/module`!)

   The apparent version of the module generated in this way will match your current kernel version.

With the modules in hand, proceed to install as follows:

1. Change to the root user: `su` (enter root password when prompted)
2. Create the device file: `mknod /dev/pctel c 62 79`
3. Make convenience device: `ln −s /dev/pctel /dev/modem`
4. Make convenience device: `ln −s /dev/pctel /dev/ttyS15`
5. Set device ownership: `chgrp uucp /dev/pctel`
6. Set device permissions: `chmod 666 /dev/pctel`
7. Install module file (only for package type 2, above): `cp pctel.o /lib/modules/`uname −r`/misc/`
8. Insert module(s) in kernel with `insmod −f` *modulename*

5.5 Conexant/Rockwell HSF

There exist drivers for kernels 2.2.14, 2.2.16, and 2.2.17 at http://www.olitec.com/pci56kv2.html The page is in French, but the installation commands are given on the page in boldface red text (you can also use the babel fish). Essentially, download the appropriate package, unpack it with `tar −zxvf`, and run the installation script `ins_all`.

This driver is a bit finicky (with the most common symptom of failure being the "NO DIALTONE" response), but a number of people have been able to get it to work, usually by inserting their modem's vendor ID in the modem's .inf file, perhaps along with a change of the device major number from 254 to 253. See the Linmodems.org mailing list archives for details; one example of many is given here.

5.6 Intel (formerly Ambient Technology, formerly Cirrus Logic)

Mikhail Moreyra has written a GPL'ed driver for the CL−MD5620DT chipset which can do up to 33.6 kbps; however, this is alpha software and should be treated with due care. The driver can be obtained at http://linmodems.org/CLModem−0.3.0.tar.gz. Gabriel Gambetta (ggambett(at)internet.com.uy) issued a patched version of the driver to allow standard AT modem commands; you can get this version at Rob Clark's site here.

A driver for the HaM modem was beta−tested in early 2001 and is expected to be released quite soon.

5.7 3Com

56k

An rpm package with a driver for the 3Com MDP3900V−U modem (apparently found in the Dell Dimension L733r) was posted to the Linmodems.org mailing list (click here for more information), and is mirrored here.

Mini−PCI

A request for comments was posted by a 3Com official about the possible demand for a binary−only driver for their miniPCI combination NIC/winmodem here on the Linodems.org mailing list; please respond to the address given, linmodem@new−n−used.com, and not to the mailing list. Though to my knowledge no driver has yet been released, Werner Heuser's miniPCI page has more information and links.

5.8 AMR

Ian Stewart reports that he is working on a "mid−level driver" for the AC97 codec.

6. Troubleshooting

So you've read through this document, the Modem−HOWTO, and the PPP Howto, are pretty sure that your modem matches one of the drivers available, but it still doesn't work? There are a number of points in the process at which something could break down.

Linux generally maintains records of networking connections which are very useful in troubleshooting problems. Their particular filenames vary with both the Linux distribution and Dial−in software, but the system log files `/var/log/messages`, `/var/log/syslog`, etcetera, should provide at least some information.

For both your own trouble shooting and queries for help to a list, it will be useful if you accumulate the information requested below. As root, change to the directory in which the modem install scripts are located, and start a script record as shown below. After this script is terminated with "exit," copy it out of your Linux partition for transmission to the list which may aid you.

(Below, # are explanatory comments.)

```
# start the recording,
script ModemTest.txt
# type in as much info on your Modem card as you have
echo winmodem name, manufacturer, designation, and chip if possible
# this gives your current kernel version
uname −r
# this gives information on your serial ports
setserial −agv /dev/ttyS*
# this information on your interrupts (irq)
cat /proc/interrupts
# show the contents of your module installation script (insert script name):
cat ScriptName
# Check if your script is executable:
ls −l ScriptName
# a response is OK if it has "x" such as below:
# −rwxrw−rw−  1 root     root  654 Jan  6  2000 ltinst
# otherwise make it executable with:
chmod o+x ScriptName
# verify with
ls −l ScriptName
# if ScriptName has not been successfully run before under this kernel
# run it with:
./ScriptName
# what is the symbolic link /dev/modem set to:
ls −l /dev/modem
# What is the DeviceName specified in the ScriptName (/dev/ttyS14 or ...?)
echo DeviceName
# what is your modem driver name? Something like DriverName.o
# with the ".o" indicating it is a compiled binary
echo This is my DriverName.o
# if should have been inserted in the Modules Path
# Try to display it there with:
```

```
find /lib/modules | grep DriverName
# Is DriverName among the modules installed in the running kernel?
lsmod
# if not try a simple insertion:
insmod ./DriverName.o
# or if it was in the Modules Path, the following will suffice:
insmod DriverName
# check for insertion:
lsmod
# if not inserted, try forcing:
insmod -f ./DriverName
# list your inserted modules again.
lsmod
# If DriverName is NOT listed,
# their is an incompatibility between modem hardware, driver and kernel.
# Further effort will be of No use.
# If DriverName is listed, let's do a bit more information.
# You may first wish to rerun the configuration utility
# used to setup dial-in connections for your Linux installation.
# Remember to edit your PassWord from this record later.
# You will probably be queried for the following information
# which you should have ready:
#Port to be used (/dev/modem or /dev/ttySn),Dial-inNumber, UserName, PassWord.
# Run your configuration utility.
YourSetUpConf
# To stop recording
exit
```

If dialin was not successfull, append to this a record from your log file. As an example, a section of a /var/log/syslog from a Debian Linux system is below.

---

7. FAQ

7.1 I have a winmodem. Will it work under Linux?

Probably not. Please see the section "Which Linmodem hardware is supported?" above, and check the Linux Modem Compatibility database at Rob Clark's site.

7.2 I get "NO DIALTONE".

Try setting your BIOS option from "PNP OS" to "non−PNP OS", from "Windows" to "Other OS", or the equivalent.

Conexant users: See the Conexant section, above.

7.3 I get a "device or resource busy" error.

- If you have an ISA modem, did you use the isapnptools to allocate IRQ and DMA resources to the card? See "ISA Plug−n−Play", above, for more information.
- Double−check that you created the device file correctly, and try to eliminate any IRQ conflicts you might have. If all looks well, but it still doesn't work, check the Linmodems.org mailing list to see if someone else has (and has perhaps fixed) the same problem, or try to fix it yourself and inform others

of your results.

7.4 I get unresolved symbols when fixscripting/insmoding.

Unresolved symbols are a true danger of version mismatching and are, in general, bad, but are also almost inevitable with binary modules. If the fixscript reports unresolved symbols, or the module does not work despite the unresolved symbols, you may be out of luck with that kernel/module combination; however, a few common cases involve symbols like:

- `slhc_xxxx`: You probably need to insmod the slhc module before inserting the modem/ppp modules; using modprobe rather than insmod should also obviate this problem.
- `printk, jiffies`: Your kernel may be compiled with SMP enabled. None of the binary modules are known to be SMP−safe, and will probably only work on a single−processor machine with a single−processor kernel, i.e. SMP disabled. You should try recompiling your kernel or otherwise obtaining a version with SMP disabled. (Thanks to Tom Reinertson (treinertson(at)uswest.net))
- `tty_xxxx` with esscom.o: Earlier fixscripts were not able to handle the version−specific symbols in this module. More recent versions are available at http://www.test.dclabs.com.au/linmodem/fixscript) which should be able to fix this module as well.

If a module works in an unstable fashion, it could be that, under some circumstances, you are avoiding those symbols, while in others, you slam up against them. Try out different ppp dialup programs (wvdial, kppp), which call a different set of functions under similar conditions. It is also possible that the fixscript, which was designed for the lucent module, is not "fixing" the symbols used in your module. If you find no combination that works, consider "downgrading" to a kernel which has a closer version match to that of the module.

7.5 My PCTel modem doesn't work.

- Do you need to give the module a country code parameter? See the appendix.
- Are you using the right driver module? There are a few PCTel drivers around (see the section "Which Linmodem hardware is supported?" above). You might try one of the others and see if that helps.

7.6 The modem dials and connects fine, but then it drops the connection.

This is an often−reported problem that may have a few, or no, solutions:

1. It is possible that the module is installed correctly and is working, but that you have a problem with your ppp configuration. In particular, if you find an error in the log along the lines of "peer is not authorized," try changing "auth" to "noauth" in /etc/ppp/options, and/or commenting out "auth" and "lock" (by placing a '#' at the beginning of the line). Corel has a FAQ entry at http://linux.corel.com/support/html/9314.htm about this.
2. It has been reported that, with some kernel/module mismatches, a program like `kppp` will give this error, while an alternative like `wvdial` does not, for the same modules and hardware. You may wish to try a different ppp dialer and see if that helps.
3. Lastly, there is the potential relationship with sound support. Comparing functionality of ltmodem.o with/without sound support in the kernels, dial−in is OK, but ppp is NOT achieved for the kernel without sound support.

Most Linux distributions do deposit a kernel configuration file along with the kernel. For Debian related distributions, it is the file

```
/boot/config-version
```

The positive choices can be quickly displayed with:

```
grep SOUND /boot/config-version |grep -v not
```

For the specific example of a 2.2.17 version:

```
# grep SOUND  /boot/config-2.2.17 |grep -v not
CONFIG_SOUND=m
CONFIG_SOUND_OSS=m
CONFIG_SOUND_SB=m
CONFIG_SOUND_MPU401=m
CONFIG_SOUND_YM3812=m
CONFIG_SOUND_VMIDI=m
CONFIG_SOUND_YMPCI=m
CONFIG_LOWLEVEL_SOUND=y
```

Either CONFIG_SOUND=m or CONFIG_SOUND=yes would show that the kernel has sound support (as would simple sound output).

If none of these helps, you may wish to consider trying to use a kernel version which is closer to the module. Otherwise, try the mailing list at Linmodems.org for help.

7.7 I get a kernel panic on closing the connection or unloading the driver.

There are a couple of possible solutions to this, neither of which may work:

- Try a different ppp dialer (wvdial, kppp).
- Configure the module so that it stays in the kernel, i.e. so that it is not unloaded.

7.8 Nothing seems to work. To whom can I turn for help?

- Double−check that the modem you have is actually supported by the module you have. See "Which Linmodem Hardware Is Supported?" above.
- Try to determine at which stage of the installation process things break down. Check the man pages on the commands used in that stage and see if you can determine the source of the problem.

If all seems lost, please see the section "Troubleshooting", below , and consider sending a message with the complete information described there to the mailing list at Linmodems.org.

7.9 Who wrote the driver for my winmodem, and how do I contact him/her?

If a contact address is not given above, you can in general assume that it was probably somebody on contract to the manufacturer, who probably does not have the authority the update/release/change the source code, and who probably doesn't have time to reply to your email in any case. See, for example, http://lwn.net/1999/1209/a/lucent.html

8. Appendix

8.1 PCTel Module Parameters: Country Code

The following is quoted from one of the PCTel readme files. Thus you can choose the appropriate country code by inserting the module with a parameter as:

```
insmod pctel.o country_code=7
```

(the "7" being replaced by your country code from the list below). Thanks to Jonathan Emery for pointing out the correct syntax.

```
Set and report country code.

This driver takes a module parameter to setup the correct country code
setting for various country's telephone networks and it also can report
back the country code been set.

Here are the two versions for country_code selection and reporting:

VERSION #1:

To set country code:
"country_sel_rep   sel   7" will sets the country code to 7.

To query the driver for the currently set country code:
"country_sel_rep   rep" returns the current country code as the exit code.

VERSION #2:

To set country code:
"country_sel   7" to set the country code to 7.

To query the driver for the currently set country code:
"country_rep" return the current country code as the exit code.

country_code                    country_name

    1                           USA
    2                           FRANCE
    3                           GERMANY
    4                           ITALY
    5                           SWEDEN
    6                           UK
    7                           JAPAN
    8                           AUSTRALIA
    9                           SPAIN
   10                           TAIWAN
   11                           SINGAPORE
   12                           KOREA
   13                           SWITZERLAND
   14                           NORWAY
   15                           NETHERLANDS
```

```
   16                            BELGIUM
   17                            CANADA
   18                            IRELAND
   19                            PORTUGAL
   20                            POLAND
   21                            HUNGARY
   22                            FINLAND
   23                            DENMARK
   24                            AUSTRIA
   25                            S.AFRICA
   26                            CTR21 COUNTRIES
   27                            CHINA
   28                            MALAYSIA
   29                            LUXUMBURG
   30                            GREECE
   31                            ICELAND
   32                            NEW ZEALAND
   33                            BRAZIL
```