The Linux Sound HOWTO

Jeff Tranter, jeff_tranter@mitel.com

v1.14, 10 June 1996

This document describes sound support for Linux. It lists the supported sound hardware, describes how to configure the kernel drivers, and answers frequently asked questions. The intent is to bring new users up to speed more quickly and reduce the amount of traffic in the Usenet news groups.

Contents

1 Introduction

This is the Linux Sound HOWTO document. It is intended as a quick reference covering everything you need to know to install and configure sound support under Linux. Frequently asked questions about sound under Linux are answered, and references are given to some other sources of information on a variety of topics related to computer generated sound and music.

The scope is limited to the aspects of sound cards pertaining to Linux. See the other documents listed in the *References* section for more general information on sound cards and computer sound and music generation.

1.1 Acknowledgments

Much of this information came from the `README` files provided with the sound driver source code, by Hannu Savolainen (hannu@voxware.pp.fi). Thanks go to Hannu and the many other people who developed the Linux kernel sound drivers and utilities.

Thanks to the Linuxdoc-SGML package, this HOWTO is available in several formats, all generated from a common source file.

1.2 Revision History

Version 1.1

first version; posted to SOUND channel of Linux activists mailing list only

Version 1.2

minor updates; first version available on archive sites

Version 1.3

converted to SGML; now available in several formats using Matt Welsh's Linuxdoc-SGML tools; appearance changed due to new format, only minor changes to content

Version 1.4

minor tweaking of SGML; added answer on PAS16 and Adaptec1542A SCSI adaptor incompatibilities

Version 1.5

2.5a sound driver is now in 1.1 kernel distribution; note on GUS-MAX support; other minor updates

Version 1.6

added info on "no space on device" error; added note that Hacker's Guide is in a "hidden" directory; added question on bidirectional mode; info on "device busy" errors; other minor changes

Version 1.7

added info on ASP and AWE32; VoxWare 2.9 is available; answer to question on using IRQ2; references to Sound and SCSI HOWTOs

Version 1.8

added question on errors under DOS; many minor things updated to match the version 2.90 sound driver; info on DOOM; answer on reducing noise

Version 1.9

questions on recording and clone cards

Version 1.10

mentioned that HOWTO is available on WWW, as printed copies, and translations; info on DMA conflict with QIC tape driver; info on Sound Galaxy NX Pro and Logitech BusMouse

Version 1.11

A long overdue update (I've been busy); document placed under GPL; brought up to date with version 3.0 sound driver; info on many new supported sound card drivers; more info on configuration and troubleshooting; lots of HTML links added; brought in line with format of CD-ROM HOWTO

Version 1.12

new sound drivers in 1.3.34 kernel; new sound device names; 1542 address is 334 not 333; clarify status of Creative Labs Emu and ASP; pointer to Creative Labs and MediaTrix Web sites

Version 1.13

note on the name VoxWare; updated to reflect latest supported sound cards and configuration options; question on Plug and Play support; question on block size problem; new xconfig and menuconfig options; modutils has sound device support; vger mailing list going away; emphasize author's Web site; other miscellaneous minor changes

Version 1.14

Audio Excell DSP16 is not currently supported (should be working again in a few months); changes to configure program; Italian version of HOWTO available; trick for setting mixer gains when loading sound module; latest stable kernel is now 2.0; new name for sound driver; question on root permissions on sound device files

1.3 New versions of this document

New versions of this document will be periodically posted to the comp.os.linux.answers newsgroup. They will also be uploaded to various anonymous ftp sites that archive such information including `<ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/>`.

Hypertext versions of this and other Linux HOWTOs are available on many World-Wide-Web sites, including `<http://sunsite.unc.edu/mdw/mdw.html>`. Most Linux CD-ROM distributions include the HOWTOs, often under the `</usr/doc/>` directory, and you can also buy printed copies from several vendors.

A French translation of this document is available at `<ftp://ftp.ibp.fr/pub2/linux/french/docs/HOWTO/>`.

A Japanese translation is available is available from `<http://yebisu.ics.es.osaka-u.ac.jp/linux/>`.

An Italian translation is available is available from `<http://www.psico.unipd.it/ildp/docs/HOWTO/Sound-HOWTO.html>`.

If you make a translation of this document into another language, let me know and I'll include a reference to it here.

1.4 Feedback

I rely on you, the users, to make this HOWTO useful. If you have any suggestions, corrections, or comments, please send them to me, jeff_tranter@mitel.com, and I will try to incorporate them in the next revision.

I am also willing to answer general questions on sound cards and Linux, as best I can. Before doing so, *please read all of the information in this HOWTO*, and *send me detailed information about the problem*.

If you publish this document on a CD-ROM or in hardcopy form, a complimentary copy would be appreciated. Mail me for my postal address. Also consider making a donation to the Linux Documentation Project to help support free documentation for Linux. Contact the Linux HOWTO coordinator, Greg Hankins `<mailto:gregh@sunsite.unc.edu>`, for more information.

1.5 Distribution Policy

2 Sound Card Technology

This section gives a *very* cursory overview of computer audio technology, in order to help you understand the concepts used later in the document. You should consult a book on digital audio or digital signal processing in order to learn more.

Sound is an *analog* property; it can take on any value over a continuous range. Computers are *digital*; they like to work with discrete values. Sound cards use a device known as an *Analog to Digital Converter* (A/D or ADC) to convert voltages corresponding to analog sound waves into digital or numeric values which can be stored in memory. Similarly, a *Digital to Analog Converter* (D/A or DAC) converts numeric values back to an analog voltage which can in turn drive a loudspeaker, producing sound.

The process of analog to digital conversion, known as sampling, introduces some error. Two factors are key in determining how well the sampled signal represents the original. *Sampling rate* is the number of samples made per unit of time (usually expresses as samples per second or Hertz). A low sampling rate will provide a less accurate representation of the analog signal. Sample size is the range of values used to represent each sample, usually expressed in bits. The larger the sample size, the more accurate the digitized signal will be.

Sound cards commonly use 8 or 16 bit samples at sampling rates from about 4000 to 44,000 samples per second. The samples may also be contain one channel (mono) or two (stereo).

FM Synthesis is an older technique for producing sound. It is based on combining different waveforms (e.g. sine, triangle, square). FM synthesis is simpler to implement in hardware that D/A conversion, but is more difficult to program and less flexible. Many sound cards provide FM synthesis for backward compatibility with older cards and software. Several independent sound generators or *voices* are usually provided.

Wavetable Synthesis combines the flexibility of D/A conversion with the multiple channel capability of FM synthesis. With this scheme digitized voices can be downloaded into dedicated memory, and then played, combined, and modified with little CPU overhead. State of the art sound cards all support wavetable synthesis.

Most sound cards provide the capability of *mixing*, combining signals from different input sources and controlling gain levels.

MIDI stands for Musical Instrument Digital Interface, and is a standard hardware and software protocol for allowing musical instruments to communicate with each other. The events sent over a MIDI bus can also be stored as MIDI files for later editing and playback. Many sound cards provide a MIDI interface. Those that do not can still play MIDI files using the on-board capabilities of the sound card.

MOD files are a common format for computer generated songs. As well as information about the musical notes to be played, the files contain digitized samples for the instruments (or voices). MOD files originated on the Amiga computer, but can be played on other systems, including Linux, with suitable software.

3 Supported Hardware

This section lists the sound cards and interfaces that are currently supported under Linux. The information here is based on the latest Linux kernels, at time of writing.

The sound driver has its own version numbering. The latest stable Linux kernel release was version 2.0, using sound driver version 3.5.2-960330.

The author of the sound driver, Hannu Savolainen, typically also makes available newer beta releases of the sound driver as kernel patches before they are included as part of the standard Linux kernel distribution.

This information is only valid for Linux on the Intel platform. Some information may be applicable to other processor architectures, but I have no first hand experience or information.

3.1 Sound Cards

The following sound cards are supported by the Linux kernel sound driver:

- ATI Stereo F/X
- AdLib
- Ensoniq SoundScape (and compatibles made by Reveal and Spea)
- Gravis Ultrasound
- Gravis Ultrasound ACE
- Gravis Ultrasound Max
- Gravis Ultrasound with 16 bit sampling option
- Logitech Sound Man 16
- Logitech SoundMan Games
- Logitech SoundMan Wave
- MAD16 Pro (OPTi 82C929)
- Media Vision Jazz16
- MediaTriX AudioTriX Pro
- Microsoft Windows Sound System (MSS/WSS)
- Mozart (OAK OTI-601)
- Orchid SW32
- Personal Sound System (PSS)
- Pro Audio Spectrum 16
- Pro Audio Studio 16
- Pro Sonic 16
- Roland MPU-401 MIDI interface

- Sound Blaster 1.0
- Sound Blaster 16
- Sound Blaster 16ASP
- Sound Blaster 2.0
- Sound Blaster AWE32
- Sound Blaster Pro
- TI TM4000M notebook
- ThunderBoard
- Turtle Beach Tropez
- Yamaha FM synthesizers (OPL2, OPL3 and OPL4)
- 6850 UART MIDI Interface

The following cards are *not* supported, either because they are obsolete or because the vendor will not release the programming information needed to write a driver:

- Pro Audio Spectrum (original)
- Pro Audio Spectrum+
- older (Sierra Aria based) soundcards made by Diamond

Cards using the Plug and Play protocol are not currently supported but should be in the future.

Other sound cards that are claimed to be compatible with one of the supported sound cards *may* work if they are hardware (i.e. register level) compatible. Some cards described as "100% SoundBlaster compatible" are *not* register compatible. Some hints for getting compatible cards to work are given later in this document. If you have a card that works and is not listed here, let me know and I will add it to the list.

The Linux kernel supports the SCSI port provided on some sound cards (e.g. ProAudioSpectrum 16) and the proprietary interface for some CD-ROM drives (e.g. Soundblaster Pro). See the Linux SCSI HOWTO and CDROM HOWTO documents for more information.

A loadable kernel module to support joystick ports, including those provided on some sound cards, is also available.

Note that the kernel SCSI, CD-ROM, joystick, and sound drivers are completely independent of each other.

For the latest information on the sound card driver check Hannu Savolainen's World-Wide Web site listed in the References section.

3.2 Alternate Sound Drivers

There are some "unofficial" sound drivers available, not included in the standard Linux kernel distribution, and used in place of the standard sound driver.

Markus Mummert (mum@mmk.e-technik.tu-muenchen.de) has written a driver package for the Turtle Beach MultiSound (classic), Tahiti, and Monterey soundcards. The documentation states:

> "It is designed for high quality hard disk recording/playback without losing sync even on a busy system. Other features such as wave synthesis, MIDI and digital signal processor (DSP) cannot be used. Also, recording and playback at the same time is not possible. It currently replaces VoxWare and was tested on several kernel versions ranging from 1.0.9 to 1.2.1. Also, it is installable on UN*X SysV386R3.2 systems."

It can be found at `<http://www.cs.colorado.edu/ mccreary/tbeach>`.

Kim Burgaard (burgaard@daimi.aau.dk) has written a device driver and utilities for the Roland MPU-401 MIDI interface. The Linux software map entry gives this description:

> "A device driver for true Roland MPU-401 compatible MIDI interfaces (including Roland SCC-1 and RAP-10/ATW-10). Comes with a useful collection of utilities including a Standard MIDI File player and recorder.
>
> Numerous improvements have been made since version 0.11a. Among other things, the driver now features IRQ sharing policy and complies with the new kernel module interface. Metronome functionality, possibility for synchronizing e.g. graphics on a per beat basis without losing precision, advanced replay/record/overdub interface and much, much more."

It can be found at `<ftp://sunsite.unc.edu/pub/Linux/kernel/sound/mpu401-0.2.tar.gz>`.

3.3 PC Speaker

An alternate sound driver is available that requires no additional sound hardware; it uses the internal PC speaker. It is mostly software compatible with the sound card driver, but, as might be expected, provides much lower quality output and has much more CPU overhead. The results seem to vary, being dependent on the characteristics of the individual loudspeaker. For more information, see the documentation provided with the release.

The current version is 0.9b, and can be found at `<ftp://ftp.informatik.hu-berlin.de/pub/os/linux/hu-sound/>`

3.4 Parallel Port

Another option is to build a digital to analog converter using a parallel printer port and some additional components. This provides better sound quality than the PC speaker but still has a lot of CPU overhead. The PC sound driver package mentioned above supports this, and includes instructions for building the necessary hardware.

4 Installation

Configuring Linux to support sound involves the following steps:

1. Installing the sound card.
2. Configuring and building the kernel for sound support.
3. Creating the device files.
4. Booting the Linux kernel and testing the installation.

The next sections will cover each of these steps in detail.

4.1 Installing the Sound Card

Follow the manufacturer's instructions for installing the hardware or have your dealer perform the installation.

Older sound cards usually have switch or jumper settings for IRQ, DMA channel, etc; note down the values used. If you are unsure, use the factory defaults. Try to avoid conflicts with other devices (e.g. ethernet cards, SCSI host adaptors, serial and parallel ports) if possible.

4.2 Configuring the Kernel

When initially installing Linux you likely used a precompiled kernel. These kernels usually do not provide sound support. It is best to recompile the kernel yourself with the drivers you need. You may also want to recompile the kernel in order to upgrade to a newer version or to free up memory resources by minimizing the size of the kernel.

The *Linux Kernel HOWTO* `<http://sunsite.unc.edu/mdw/HOWTO/Kernel-HOWTO.html>` should be consulted for the details of building a kernel. I will just mention here some issues that are specific to sound cards.

If you have never configured the kernel for sound support before it is a good idea to read *all* of the Readme files included with the kernel sound drivers, particularly information specific to your card type. The following documentation files can be found in the kernel sound driver directory, usually installed in `</usr/src/linux/drivers/sound>`:

```
CHANGELOG         - description of changes in each release
COPYING           - copying and copyright restrictions
Readme            - latest and most important news
Readme.aedsp16    - information about Audio Excel DSP 16 sound card
Readme.cards      - notes on configuring specific cards
Readme.linux      - notes on installing separately release sound drivers
Readme.modules    - how to build driver as a loadable kernel module
```

```
Readme.v30        - new features in version 3.0 sound driver
experimental.txt  - notes on experimental features
```

Follow the usual procedure for building the kernel. There are currently three ways to run the configuration process. A graphical user interface that runs under X11 can be invoked using "make xconfig". A menu-based system that only requires text displays is available as "make menuconfig". I will assume that you use the traditional command line configuration process invoked using "make config", although the process is similar in each case.

It is also possible to build the sound driver as a kernel loadable module. I recommend initially building the driver into the kernel. Once it is tested and working you can explore using the kernel module option.

When you run `make config`, enable sound support by answering "y" to the question

```
Sound card support (CONFIG_SOUND) [M/n/y/?]
```

At the end of the configuration questions a sound configuration program will be compiled, run, and will then ask you what sound card options you want. Be careful when answering these questions since answering a question incorrectly may prevent some later ones from being asked. For example, don't answer "yes" to the first question (PAS16) if you don't really have a PAS16. Don't enable more cards than you really need, since they just consume memory. Also some drivers (like MPU401) may conflict with your SCSI controller and prevent the kernel from booting.

I list here a brief description of each of the configuration dialog options. Answer "y" (yes) or "n" (no) to each question. The default answer is shown so that "*Y/n/?*" means "y" by default and "*N/y/?*" means the default is "n". To use the default value, just hit Enter, but remember that the default value isn't necessarily correct.

Entering a question mark ("?") will produce a short descriptive message describing that configuration option.

Note also that all questions may not be asked. The configuration program may disable some questions depending on the earlier choices. It may also select some options automatically as well.

Old configuration exists in /etc/soundconf. Use it *Y/n/?*

If you have previously compiled the kernel for sound support, then the previous configuration can be saved. If you want to use the previous setup, answer "y". If you are trying a different configuration or have upgraded to a newer kernel, you should answer "n" and go through the configuration process.

ProAudioSpectrum 16 support *Y/n/?*

Answer "y" *only* if you have a Pro Audio Spectrum *16*, ProAudio Studio 16 or Logitech SoundMan 16. Don't answer 'y' if you have some other card made by Media Vision or Logitech since they are not PAS16 compatible.

SoundBlaster support *Y/n/?*

Answer "y" if you have an original SoundBlaster card made by Creative Labs or a 100% hardware compatible clone (like the Thunderboard or SM Games). If your card was in the list of supported cards look at the card specific instructions in the `Readme.cards` file before answering this question. For an unknown card you may answer "y"'if the card claims to be SoundBlaster compatible.

Gravis Ultrasound support *Y/n/?*

Answer "y" if you have a GUS or GUS MAX. Answer "n" if you don't have a GUS since the driver consumes a lot of memory.

MPU-401 support (NOT for SB16) *Y/n/?*

Be careful with this question. The MPU401 interface is supported by almost all soundcards. However, some natively supported cards have their own driver for MPU401. Enabling the MPU401 option with these cards will cause a conflict. Also enabling MPU401 on a system that doesn't really have a MPU401 could cause some trouble. If your card was in the list of supported cards, look at the card specific instructions in the `Readme.cards` file. It's safe to answer "y" if you have a true MPU401 MIDI interface card.

6850 UART Midi support *Y/n/?*

It's safe to answer "n" to this question in all cases. The 6850 UART interface is very rarely used.

PSS (ECHO-ADI2111) support *Y/n/?*

Answer "y" only if you have Orchid SW32, Cardinal DSP16 or some other card based on the PSS chipset (AD1848 codec + ADSP-2115 DSP chip + Echo ESC614 ASIC CHIP).

16 bit sampling option of GUS (*not* GUS MAX) *Y/n/?*

Answer "y" if you have installed the 16 bit sampling daughtercard on your GUS. Answer "n" if you have a GUS MAX. Enabling this option disables GUS MAX support.

GUS MAX support *Y/n/?*

Answer "y" only if you have a GUS MAX.

Microsoft Sound System support *Y/n/?*

Again think carefully before answering "y" to this question. It's safe to answer "y" if you have the original Windows Sound System card made by Microsoft or Aztech SG 16 Pro (or NX16 Pro). Also you may answer "y" in case your card was not listed earlier in this file. For cards having native support in VoxWare, consult the card specific instructions in `Readme.cards`. Some drivers have their own MSS support and enabling this option will cause a conflict.

Ensoniq Soundscape support *Y/n/?*

Answer "y" if you have a soundcard based on the Ensoniq SoundScape chipset. Such cards are being manufactured at least by Ensoniq, Spea and Reveal (Reveal makes other cards also).

MediaTriX AudioTriX Pro support *Y/n/?*

Answer "y" if you have the AudioTriX Pro.

Support for MAD16 and/or Mozart based cards?

Answer "y" if your card has a Mozart (OAK OTI-601) or MAD16 (OPTi 82C928 or 82C929) audio interface chip. These chips are currently quite common so it's possible that many no-name cards have one of them. In addition the MAD16 chip is used in some cards made by known manufacturers such as Turtle Beach (Tropez), Reveal (some models) and Diamond (latest ones).

Support for Crystal CS4232 based (PnP) cards *Y/n/?*

Answer "y" if you have a card based on the Crystal CS4232 chip set.

Support for Turtle Beach Wave Front (Maui, Tropez) synthesizers *Y/n/?*

Answer "y" if you have any of these cards.

SoundBlaster Pro support *Y/n/?*

Enable this option if your card is a SoundBlaster Pro or SoundBlaster 16. Enable it also with any SoundBlaster Pro clones. Answering "n" saves some memory but "y" is the safe alternative.

SoundBlaster 16 support *Y/n/?*

Enable if you have a SoundBlaster 16 (including the AWE32).

Audio Excel DSP 16 initialization support *Y/n/?*

Enable this if you have an Audio Excel DSP16 card. See the file `Readme.aedsp16` for more information.

The configuration program then asks some questions about the higher level services. It's recommended to answer "y" to each of these questions. Answer "n" only if you know you will not need the option.

/dev/dsp and /dev/audio support (usually required) *Y/n/?*

Answering "n" disables `/dev/dsp` and `/dev/audio`, the A/D and D/A converter devices. Answer "y".

MIDI interface support *Y/n/?*

Answering "n" disables `/dev/midixx` devices and access to any MIDI ports using `/dev/sequencer` and `/dev/music`. This option also affects any MPU401 and/or General MIDI compatible devices.

FM synthesizer (YM3812/OPL-3) support *Y/n/?*

Answer "y" here.

/dev/sequencer support *Y/n/?*

Answering "n" disables `/dev/sequencer` and `/dev/music`

Do you want support for the mixer of SG NX Pro ?

Answer "y" if you have a Sound Galaxy NX Pro sound card and want support for its extended mixer functions.

Do you want support for the MV Jazz16 (ProSonic etc.) ?

Answer "y" if you have an MV Jazz16 sound card.

Do you have a Logitech SoundMan Games *Y/n/?*

Answer "y" if you have a Logitech SoundMan Games sound card.

After the above questions the configuration program prompts for the card specific configuration information. Usually just a set of I/O address, IRQ and DMA numbers are asked. With some cards the program asks for some files to be used during initialization of the card. These are used by cards which have a DSP chip or microprocessor which must be initialized by downloading a program (microcode) file to the card. In some cases this file is written to a .h file by the config program and then included to the driver during compile. Again, read the information in the file `Readme.cards` pertaining to your card type.

At the end you will be prompted:

```
The sound driver is now configured.
Save copy of this configuration to /etc/soundconf [Y/n/?]
```

Normally you would enter "y" so that if you later need to recompile the kernel you have the option of using the same sound driver configuration.

If you are upgrading from an older sound driver, make sure that the files `/usr/include/sys/soundcard.h` and `/usr/include/sys/ultrasound.h` are symbolic links to the corresponding files in `/usr/include/linux`, or that they simply contain the lines `#include <linux/soundcard.h>` and `#include <linux/ultrasound.h>`, respectively.

You are now ready to compile and install the new kernel.

4.3 Creating the Device Files

The first time the kernel sound driver is configured you need to create the sound device files. The easiest way to do this is to cut the short shell script from the end of the file `Readme.linux` in the directory `/usr/src/linux/drivers/sound`, and run it as user `root`.

If your device entries already exist, you might want to ensure they are correct. If they are not, or if you are in doubt, run the above script and it will replace any existing entries with correct ones.

Some older Linux distributions provided install scripts which created incorrect sound device files. You may also have a `/dev/MAKEDEV` script for creating device files. Using the script included with the kernel sound driver is preferred since it should always be up to date with the latest supported sound devices.

After running the script your sound device files should look something like this:

```
lrwxrwxrwx   1 root        11 Aug 22 00:01 audio -> /dev/audio0
crw-rw-rw-   1 root   14,   4 Aug 22 00:01 audio0
crw-rw-rw-   1 root   14,  20 Aug 22 00:01 audio1
lrwxrwxrwx   1 root         9 Aug 22 00:01 dsp -> /dev/dsp0
crw-rw-rw-   1 root   14,   3 Aug 22 00:01 dsp0
crw-rw-rw-   1 root   14,  19 Aug 22 00:01 dsp1
crw-rw-rw-   1 root   14,   2 Aug 22 00:01 midi00
crw-rw-rw-   1 root   14,  18 Aug 22 00:01 midi01
crw-rw-rw-   1 root   14,  34 Aug 22 00:01 midi02
crw-rw-rw-   1 root   14,  50 Aug 22 00:01 midi03
crw-rw-rw-   1 root   14,   0 Aug 22 00:01 mixer
```

```
crw-rw-rw-   1 root   14,  16 Aug 22 00:01 mixer1
crw-rw-rw-   1 root   14,   8 Aug 22 00:01 music
crw-rw-rw-   1 root   14,  17 Aug 22 00:01 patmgr0
crw-rw-rw-   1 root   14,  33 Aug 22 00:01 patmgr1
crw-rw-rw-   1 root   14,   1 Aug 22 00:01 sequencer
lrwxrwxrwx   1 root        10 Aug 22 00:01 sequencer2 -> /dev/music
crw-rw-rw-   1 root   14,   6 Aug 22 00:01 sndstat
```

Pay particular attention to the device names and the major and minor device numbers listed in the fourth and fifth columns.

If you are using the PC speaker sound driver, read the documentation that came with the package to determine what device files to create.

Normally the configuration you used when building the kernel will be acceptable to the sound card driver. It is also possible to pass parameters on the kernel command line (e.g. from LILO) to configure the sound driver. These are defined in the file `Readme.linux`. It should rarely be necessary to use these. They are mainly intended for developers of Linux boot disks to create a kernel that supports multiple types of sound cards.

4.4 Booting Linux and Testing the Installation

You should now be ready to boot the new kernel and test the sound drivers. Follow your usual procedure for installing and rebooting the new kernel (keep the old kernel around in case of problems, of course).

During booting, check for a message such as the following on powerup (if they scroll by too quickly to read, you may be able to retrieve them with the "dmesg" command):

```
snd2 <SoundBlaster Pro 3.2> at 0x220 irq 5 drq 1
snd1 <Yamaha OPL-3 FM> at 0x388 irq 0 drq 0
```

This should match your sound card type and jumper settings (if any).

The driver may also display some error messages and warnings during boot. Watch for these when booting the first time after configuring the sound driver.

Next you should check the device file `/dev/sndstat`. Reading the sound driver status device file should provide additional information on whether the sound card driver initialized properly. Sample output should look something like this:

```
% cat /dev/sndstat
Sound Driver:3.5.2-960330 (Tue  4 16:31:10 EDT 1996 root,
Linux fizzbin 2.0.0 #1 Mon Jun 3 16:59:37 EDT 1996 i386)
Kernel: Linux fizzbin 2.0.0 #1 Tue Jun 4 16:57:55 EDT 1996 i386
Config options: a80002

Installed drivers:
Type 1: OPL-2/OPL-3 FM
```

```
Type 2: SoundBlaster
Type 6: SoundBlaster 16bit
Type 7: SB MPU

Card config:
SoundBlaster at 0x220 irq 5 drq 1,5
OPL-2/OPL-3 FM at 0x388 irq 0 drq 0

Audio devices:
0: SoundBlaster Pro 3.2

Synth devices:
0: Yamaha OPL-3

Midi devices:
0: SoundBlaster

Timers:
0: System clock

Mixers:
0: SoundBlaster
```

Now you should be ready to play a simple sound file. Get hold of a sound sample file, and send it to the sound device as a basic check of sound output, e.g.

```
% cat endoftheworld >/dev/dsp
% cat crash.au >/dev/audio
```

(Make sure you don't omit the ">" in the commands above).

Some sample sound files can be obtained from <ftp://tsx-11.mit.edu/pub/linux/packages/sound/snd-data-0.1.tar.Z>

Now you can verify sound recording. If you have sound input capability, you can do a quick test of this using commands such as the following:

```
# record 4 seconds of audio from microphone
EDT% dd bs=8k count=4 </dev/audio >sample.au
4+0 records in
4+0 records out
# play back sound
% cat sample.au >/dev/audio
```

Obviously for this to work you need a microphone connected to the sound card and you should speak into it. You may also need to obtain a mixer program to set the microphone as the input device and adjust the recording gain level.

If these tests pass, you can be reasonably confident that the sound D/A and A/D hardware and software are working. If you experience problems, refer to the next section of this document.

4.5 Troubleshooting

If you still encounter problems after following the instructions in the HOWTO, here are some things to check. The checks are listed in increasing order of complexity. If a check fails, solve the problem before moving to the next stage.

4.5.1 Step 1: Make sure you are really running the kernel you compiled.

You can check the date stamp on the kernel to see if you are running the one that you compiled with sound support. You can do this with the `uname` command:

```
% uname -a
Linux fizzbin 2.0.0 #1 Tue Jun 4 16:57:55 EDT 1996 i386
```

or by displaying the file `/proc/version`:

```
% cat /proc/version
Linux version 2.0.0 (root@fizzbin) (gcc version 2.7.0) #1 Tue Jun 4 16:57:55 EDT 1996
```

If the date stamp doesn't seem to match when you compiled the kernel, then you are running an old kernel. Did you really reboot? If you use LILO, did you re-install it (typically by running `/etc/lilo/install`)? If booting from floppy, did you create a new boot floppy and use it when booting?

4.5.2 Step 2: Make sure the kernel sound drivers are compiled in.

You can see what drivers are compiled in by looking at `/proc/devices`:

```
% cat /proc/devices
Character devices:
 1 mem
 4 tty
 5 cua
 6 lp
14 sound
15 Joystick

Block devices:
 2 fd
 3 hd
25 sbpcd
```

What we are looking for here is character device 14, labeled "sound". If the sound device is not listed then something went wrong with the kernel configuration or build. Start the installation process again, beginning with configuration and building of the kernel.

4.5.3 Step 3: Did the kernel detect your sound card during booting?

Make sure that the sound card was detected when the kernel booted. You should have seen a message on bootup. If the messages scrolled off the screen, you can usually recall them using the `dmesg` command:

```
% dmesg
```

or

```
% tail /var/adm/messages
```

If your sound card was not found then something is wrong. Make sure it really is installed. If the sound card works under DOS then you can be reasonably confident that the hardware is working, so it is likely a problem with the kernel configuration. Either you configured your sound card as the wrong type or wrong parameters, or your sound card is not compatible with any of the Linux kernel sound card drivers.

One possibility is that your sound card is one of the "compatible" type that requires initialization by the DOS driver. Try booting DOS and loading the vendor supplied sound card driver. Then soft boot Linux using `Control-Alt-Delete`. Make sure that card I/O address, DMA, and IRQ settings for Linux are the same as used under DOS. Read the `Readme.cards` file from the sound driver source distribution for hints on configuring your card type.

If your sound card is not listed in this document, it is possible that the Linux drivers do not support it. You can check with some of the references listed at the end of this document for assistance.

4.5.4 Step 4: Can you read data from the dsp device?

Try reading from the `/dev/audio` device using the `dd` command listed earlier in this document. The command should run without errors.

If this does not work, then a possible cause is the device file. Make sure than the device files in the `/dev` directory has the correct major and minor numbers as listed previously. Check that the permissions on the device file allow reading and writing.

A remote possibility is a hardware problem. Try testing the drive under DOS, if possible, to determine if this could be the case.

4.5.5 When All Else Fails

If you still have problems, here are some final suggestions for things to try:

- carefully re-read this HOWTO document
- read the references listed at the end of this document, especially the relevant kernel source Readme files
- post a question to one of the `comp.os.linux` or other usenet newsgroups

- send a question to the Sound channel of the Linux mailing list
- try using the latest Linux kernel
- contact your computer dealer
- contact the sound card manufacturer
- send mail to the author of the sound driver
- send mail to the author of the Sound HOWTO
- fire up emacs and type `Esc-x doctor`

5 Applications Supporting Sound

I give here a sample of the types of applications that you likely want if you have a sound card under Linux. You can check the Linux Software Map, Internet archive sites, and/or files on your Linux CD-ROM for more up to date information.

As a minimum, you will likely want to obtain the following sound applications:

- audio file format conversion utility (e.g. `Sox`)
- mixer utility (e.g. `aumix` or `xmix`)
- digitized file player/recorder (e.g. `play` or `wavplay`)
- MOD file player (e.g. `tracker`)
- MIDI file player (e.g. `mp`)

There are text-based as well as GUI-based versions of most of these tools. There are also some more esoteric applications (e.g. speech synthesis and recognition) that you may wish to try.

6 Answers To Frequently Asked Questions

This section answers some of the questions that have been commonly asked on the Usenet news groups and mailing lists.

6.1 What are the various sound device files?

These are the most "standard" device file names, some Linux distributions may use slightly different names.

/dev/audio

normally a link to `/dev/audio0`

/dev/audio0

Sun workstation compatible audio device (only a partial implementation, does not support Sun ioctl interface, just u-law encoding)

/dev/audio1

second audio device (if supported by sound card)

/dev/dsp

normally a link to `/dev/dsp0`

/dev/dsp0

first digital sampling device

/dev/dsp1

second digital sampling device

/dev/mixer

sound mixer

/dev/mixer1

second sound mixer

/dev/music

high-level sequencer interface

/dev/patmgr0

Patch Manager

/dev/patmgr1

Patch Manager

/dev/sequencer

low level MIDI, FM, and GUS access

/dev/sequencer2

normally a link to `/dev/music`

/dev/midi00

1st raw MIDI port

/dev/midi01

2nd raw MIDI port

/dev/midi02

3rd raw MIDI port

/dev/midi03

4th raw MIDI port

/dev/sndstat

displays sound driver status when read

The PC speaker driver provides the following devices:

/dev/pcaudio

equivalent to /dev/audio

/dev/pcsp

equivalent to /dev/dsp

/dev/pcmixer

equivalent to /dev/mixer

6.2 How can I play a sound sample?

Sun workstation (.au) sound files can be played by sending them to the `/dev/audio` device. Raw samples can be sent to `/dev/dsp`. Using a program such as `play` is preferable, as it will recognize most file types and set the sound card to the correct sampling rate, etc.

6.3 How can I record a sample?

Reading `/dev/audio` or `/dev/dsp` will return sampled data that can be redirected to a file. A program such as `vrec` makes it easier to control the sampling rate, duration, etc. You may also need a mixer program to select the appropriate input device.

6.4 Can I have more than one sound card?

Up to two sound cards is supported. It's possible to install a Gravis UltraSound or MPU-401 with a SoundBlaster, SoundBlaster Pro, SoundBlaster16 or ProAudioSpectrum16. It's *not* possible to have a ProAudioSpectrum16 and SoundBlaster at the same time (the PAS16 has an SB emulator in it). It's also not possible to have more than one card of the same type at the same time – for example, a GUS + GUS combination is not possible.

You can change the sound card configuration parameters at boot time using command line options from a boot loader such as LILO. See the kernel sound driver file `Readme.linux` for details.

6.5 Error: `No such file or directory` for sound devices

You need to create the sound driver device files. See the section on creating device files. If you do have the device files, ensure that they have the correct major and minor device numbers (some older CD-ROM distributions of Linux may not create the correct device files during installation).

6.6 Error: `No such device` for sound devices

You have not booted with a kernel containing the sound driver or the I/O address configuration doesn't match your hardware. Check that you are running the newly compiled kernel and verify that the settings entered when configuring the sound driver match your hardware setup.

6.7 Error: `No space left on device` for sound devices

This can happen if you tried to record data to /dev/audio or /dev/dsp without creating the necessary device file. The sound device is now a regular file, and has filled up your disk partition. You need to run the script described in the *Creating the Device Files* section of this document.

6.8 Error: `device busy` for sound devices

Only one process can open a given sound device at one time. Most likely some other process is using the device in question. One way to determine this is to use the fuser command:

```
% fuser -v /dev/dsp
/dev/dsp:            USER       PID ACCESS COMMAND
                     tranter    265 f....  tracker
```

In the above example, the fuser command showed that process 265 had the device open. Waiting for the process to complete or killing it will allow the sound device to be accessed once again.

6.9 I still get `device busy` errors!

According to Brian Gough, for the SoundBlaster cards which use DMA channel 1 there is a potential conflict with the QIC-02 tape driver, which also uses DMA 1, causing "device busy" errors. If you are using FTAPE, you may have this driver enabled. According to the FTAPE-HOWTO the QIC-02 driver is not essential for the use of FTAPE; only the QIC-117 driver is required. Reconfiguring the kernel to use QIC-117 but not QIC-02 allows FTAPE and the sound-driver to coexist.

(the following explanation was supplied by Harald Albrecht `albrecht@igpm.rwth-aachen.de`)

Some soundcards support using DMA channel 0. The sound driver configuration program allows this, and the kernel compiles properly, but accessing the sound device results in a "device busy" error message.

The reason is that the Linux kernel reserves DMA channel 0 for DRAM refresh. This is no longer true for modern 386/486 boards which use their own refresh logic. You can correct it by changing this line in the file /usr/src/linux/kernel/dma.c:

```
static volatile unsigned int dma_chan_busy[MAX_DMA_CHANNELS] = {
                1, 0, 0, 0, 1, 0, 0, 0
};
```

Replace the first 1 with a 0; this enables DMA channel 0. Don't do the same with DMA channel 4 as this is cascade and won't work! The code should now look like this:

```
static volatile unsigned int dma_chan_busy[MAX_DMA_CHANNELS] = {
                0, 0, 0, 0, 1, 0, 0, 0
};
```

Recompile and reboot with the new kernel.

6.10 Partial playback of digitized sound file

The symptom is usually that a sound sample plays for about a second and then stops completely or reports an error message about "missing IRQ" or "DMA timeout". Most likely you have incorrect IRQ or DMA channel settings. Verify that the kernel configuration matches the sound card jumper settings and that they do not conflict with some other card.

Another symptom is sound samples that "loop". This is usually caused by an IRQ conflict.

6.11 There are pauses when playing MOD files

Playing MOD files requires considerable CPU power. You may have too many processes running or your computer may be too slow to play in real time. Your options are to:

- try playing with a lower sampling rate or in mono mode
- eliminate other processes
- buy a faster computer
- buy a more powerful sound card (e.g. Gravis UltraSound)

If you have a Gravis UltraSound card, you should use one of the mod file players written specifically for the GUS (e.g. `gmod`).

6.12 Compile errors when compiling sound applications

The version 1.0c and earlier sound driver used a different and incompatible `ioctl()` scheme. Obtain newer source code or make the necessary changes to adapt it to the new sound driver. See the sound driver `Readme` file for details.

Also ensure that you have used the latest version of `soundcard.h` and `ultrasound.h` when compiling the application. See the installation instructions at beginning of this text.

6.13 SEGV when running sound binaries that worked previously

This is probably the same problem described in the previous question.

6.14 What known bugs or limitations are there in the sound driver?

See the `Readme` and `CHANGELOG` files included with the sound driver kernel source.

6.15 What do all the sound driver configuration options mean?

During configuration of the sound driver, a `configure` program is compiled and executed. This program asks you some questions and then generates the header file `local.h` that defines the sound card configuration.

The configuration file defines (or undefines) the following symbols:

```
Symbol                  Meaning
======                  =======
KERNEL_SOUNDCARD        enable/disable sound driver
EXCLUDE_PAS             ProAudioSpectrum support
EXCLUDE_SB              SoundBlaster support
EXCLUDE_ADLIB           AdLib support
EXCLUDE_GUS             Gravis UltraSound support
EXCLUDE_MPU401          MPU-401 MIDI interface support
EXCLUDE_UART6850        6850 MIDI UART support
EXCLUDE_PSS             Professional Sound System support
EXCLUDE_GUS16           Gravis UltraSound support
EXCLUDE_GUSMAX          Gravis UltraSound Max support
EXCLUDE_MSS             Microsoft Sound System support
EXCLUDE_SBPRO           SoundBlaster Pro support
EXCLUDE_SB16            SoundBlaster 16 support
EXCLUDE_AUDIO           Digitized voice support
EXCLUDE_MIDI            MIDI interface support
EXCLUDE_YM3812          FM synthesizer (YM3812/OPL-3) support
EXCLUDE_SEQUENCER       MIDI sequencer support
EXCLUDE_PRO_MIDI        SoundBlaster Pro MIDI support
```

```
EXCLUDE_CHIP_MIDI       MIDI on CHIP support
SBC_BASE 0x220          SoundBlaster I/O base address
SBC_IRQ                 SoundBlaster IRQ number
SBC_DMA                 SoundBlaster DMA channel
SB16_DMA                SoundBlaster 16 DMA channel
SB16_MIDI_BASE          base address of SoundBlaster 16 MIDI port
PAS_IRQ                 ProAudioSpectrum IRQ number
PAS_DMA                 ProAudioSpectrum DMA channel
GUS_IRQ                 Gravis UltraSound IRQ number
GUS_DMA                 Gravis UltraSound DMA channel
GUS_BASE                base address of Gravis UltraSound
MPU_IRQ                 MPU-401 IRQ number
MPU_BASE                base address of MPU-401 port
DSP_BUFFSIZE            DMA buffer size
```

Several other defines are also created, setting such things as the sound driver revision level and the time and date when configure was run.

There are other parameters that are not set by the configure program. If you need to change these, edit the file `sound_config.h`.

To disable the sound driver, run `make config` and answer "no" to the "Sound card support?" question.

6.16 Where are the sound driver ioctls() etc. documented?

These are partially documented in the *Hacker's Guide to VoxWare*, currently available in draft form. The latest version is draft 2, and can be found on `<ftp://nic.funet.fi/pub/OS/Linux/ALPHA/sound>`. Note that this directory is "hidden" and will not appear in directory listings. If you "cd" to the directory and use the FTP "dir" command, the files *are* there.

6.17 What CPU resources are needed to play or record without pauses?

There is no easy answer to this question, as it depends on:

- whether using PCM sampling or FM synthesis
- sampling rate and sample size
- which application is used to play or record
- Sound Card hardware
- disk I/O rate, CPU clock speed, cache size, etc.

In general, any 386 machine should be able to play samples or FM synthesized music on an 8 bit soundcard with ease.

Playing MOD files, however, requires considerable CPU resources. Some experimental measurements have shown that playing at 44kHz requires more than 40% of the speed of a 486/50 and a 386/25 can hardly play faster than 22 kHz (these are with an 8 bit card sound such as a SoundBlaster). A card such as the Gravis UltraSound card performs more functions in hardware, and will require less CPU resources.

These statements assume the computer is not performing any other CPU intensive tasks.

Converting sound files or adding effects using a utility such as Sox is also much faster if you have a math coprocessor. The kernel driver itself does not do any floating point calculations, though.

6.18 Problems with a PAS16 and an Adaptec 1542 SCSI host adaptor

(the following explanation was supplied by `seeker@indirect.com`)

Linux only recognizes the 1542 at address 330 (default) or 334, and the PAS only allows the MPU-401 emulation at 330. Even when you disable the MPU-401 under software, something still wants to conflict with the 1542 if it's at its preferred default address. Moving the 1542 to 334 makes everyone happy.

Additionally, both the 1542 and the PAS-16 do 16-bit DMA, so if you sample at 16-bit 44 KHz stereo and save the file to a SCSI drive hung on the 1542, you're about to have trouble. The DMAs overlap and there isn't enough time for RAM refresh, so you get the dread "PARITY ERROR - SYSTEM HALTED" message, with no clue to what caused it. It's made worse because a few second-party vendors with QIC-117 tape drives recommend setting the bus on/off times such that the 1542 is on even longer than normal. Get the SCSISEL.EXE program from Adaptec's BBS or several places on the internet, and reduce the BUS ON time or increase the BUS OFF time until the problem goes away, then move it one notch or more further. SCSISEL changes the EEPROM settings, so it's more permanent than a patch to the DOS driver line in CONFIG.SYS, and will work if you boot right into Linux (unlike the DOS patch). Next problem solved.

Last problem - the older Symphony chipsets drastically reduced the timing of the I/O cycles to speed up bus accesses. None of various boards I've played with had *any* problem with the reduced timing except for the PAS-16. Media Vision's BBS has SYMPFIX.EXE that's supposed to cure the problem by twiddling a diagnostic bit in Symphony's bus controller, but it's not a hard guarantee. You may need to:

- get the motherboard distributor to replace the older version bus chip,
- replace the motherboard, or
- buy a different brand of sound card.

Young Microsystems will upgrade the boards they import for around \$30 (US); other vendors may be similar if you can figure out who made or imported the motherboard (good luck). The problem is in ProAudio's bus interface chip as far as I'm concerned; *nobody* buys a \$120 sound card and sticks it in a 6MHz AT. Most of them wind up in 25-40MHz 386/486 boxes, and should be able to handle *at least* 12MHz bus rates if the chips are designed right. Exit soapbox (stage left).

The first problem depends on the chipset used on your motherboard, what bus speed and other BIOS settings, and the phase of the moon. The second problem depends on your refresh option setting (hidden or synchronous), the 1542 DMA rate and (possibly) the bus I/O rate. The third can be determined by calling

Media Vision and asking which flavor of Symphony chip is incompatible with their slow design. Be warned, though - 3 of 4 techs I talked to were brain damaged. I would be very leery of trusting *anything* they said about someone else's hardware, since they didn't even know their own very well.

6.19 Problems with the FM synthesizer on a SoundBlaster Pro 1

The newer SB Pro has an OPL-3 FM chip, but the older version 1 used the OPL-2. The sound driver assumed the presence of an OPL-3. Version 2.5 and later of the sound driver corrects this problem.

6.20 Is it possible to read and write samples simultaneously?

Due to hardware limitations, this is not possible with most sound cards. Some newer cards do support it. See the section on "bidirectional mode" in the *Hacker's Guide to Voxware* for more information.

6.21 My SB16 is set to IRQ 2, but configure does not allow this value.

On '286 and later machines, the IRQ 2 interrupt is cascaded to the second interrupt controller. It is equivalent to IRQ 9.

6.22 Are the SoundBlaster AWE32 or SoundBlaster16 ASP supported?

These cards offer special chips (ASP and Emu) that support additional features such as wavetable synthesis, however Creative Labs is not willing to release programming information. Unless they change their policy there can be no support for the special hardware under Linux. The cards *are* supported as regular SoundBlaster 16 cards under Linux.

The Gravis UltraSound card has capabilities similar to the AWE32, and is supported under Linux. Cards based on other DSPs such as the Analog Devices ADSP-21xx may be supported in the future.

6.23 If I run Linux, then boot DOS, I get errors and/or sound applications do not work properly.

This happens after a soft reboot to DOS. Sometimes the error message misleadingly refers to a bad CONFIG.SYS file.

Most of the current sound cards have software programmable IRQ and DMA settings. If you use different settings between Linux and MS-DOS/Windows, this may cause problems. Some sound cards don't accept new parameters without a complete reset (i.e. cycle the power or use the hardware reset button).

The quick solution to this problem it to perform a full reboot using the reset button or power cycle rather than a soft reboot (e.g. Ctrl-Alt-Del).

The correct solution is to ensure that you use the same IRQ and DMA settings with MS-DOS and Linux (or not to use DOS :-).

6.24 Problems running DOOM under Linux

Users of the port of ID software's game DOOM for Linux may be interested in these notes.

For correct sound output you need version 2.90 or later of the sound driver; it has support for the real-time "DOOM mode".

The sound samples are 16-bit. If you have an 8-bit sound card you can still get sound to work using one of several programs available in `<ftp://sunsite.unc/edu/pub/Linux/games/doom>`.

If performance of DOOM is poor on your system, disabling sound (by renaming the file `sndserver`) may improve it.

By default DOOM does not support music (as in the DOS version). The program `musserver` will add support for music to DOOM under Linux. It can be found at `<ftp://pandora.st.hmc.edu/pub/linux/musserver.tgz>`.

6.25 How can I reduce noise picked up by my soundcard?

Using good quality shielded cables and trying the sound card in different slots may help reduce noise. If the sound card has a volume control, you can try different settings (maximum is probably best).

Using a mixer program you can make sure that undesired inputs (e.g. microphone) are set to zero gain.

Some sound cards are simply not designed with good shielding and grounding and are prone to noise pickup.

Finally, on my system I found that the kernel command line option `no-hlt` reduces the noise level. This tells the kernel not to use the halt instruction when running the idle process loop. You can try this manually when booting, or set it up using the command `append = "no-hlt"` in your LILO configuration file.

6.26 I can play sounds, but not record.

If you can play sound but not record, try these steps:

- use a mixer program to select the appropriate device (e.g. microphone)
- use the mixer to set the input gains to maximum
- If you can, try to test sound card recording under MS-DOS to determine if there is a hardware problem

6.27 My "compatible" sound card only works if I first initialize under MS-DOS.

Some sound card clones are not 100% register compatible with the real thing; they sometimes contain extra circuitry such as mixers. You *may* be able to use these under Linux if you first initialize under MS-DOS, then soft boot Linux (i.e. Ctrl-Alt-Delete).

One user also reported that he had better results if he used LOADLIN rather than LILO to boot Linux after initializing his sound card under MS-DOS (this was with a Diamond sound card)., They may or may not function reliably. The real solution is to find out from the manufacturer what the differences are and have the support added to the sound driver. This has been done, for example, for the Sound Galaxy NX Pro.

6.28 My 16-bit SoundBlaster "compatible" sound card only works in 8-bit mode under Linux.

16-bit sound cards described as SoundBlaster compatible are really only compatible with the 8-bit SoundBlaster Pro. They typically have a 16-bit mode which is not compatible with the SoundBlaster 16 and not compatible with the Linux sound driver.

If your card is also listed as compatible with the Microsoft Windows Sound System, you may be able to get it to work in 16-bit mode if you enable support for the WSS in the Linux sound driver. You will also probably have to do the DOS initialization trick to get the card to work.

6.29 Where can I find sound applications for Linux?

Here are some good archive sites to search for Linux specific sound applications:

- `<ftp://sunsite.unc.edu:/pub/Linux/kernel/sound/>`
- `<ftp://sunsite.unc.edu:/pub/Linux/apps/sound/>`
- `<ftp://tsx-11.mit.edu:/pub/linux/packages/sound/>`
- `<ftp://nic.funet.fi:/pub/OS/Linux/util/sound/>`
- `<ftp://nic.funet.fi:/pub/OS/Linux/xtra/snd-kit/>`
- `<ftp://nic.funet.fi:/pub/OS/Linux/ALPHA/sound/>`

6.30 Can the sound driver be compiled as a loadable module?

With recent kernels the sound driver is supported as a kernel loadable module.

See the files `</usr/src/linux/drivers/sound/Readme.modules>` and `</usr/src/linux/Documentation/modules.txt>` (or `</usr/src/linux/README>`) for details.

6.31 Can I use a soundcard to replace the system console beep?

Try the `oplbeep` program, found at `<ftp://sunsite.unc.edu/pub/Linux/apps/sound/oplbeep-alpha.tar.gz>`

Another variant is the `beep` program found at `<ftp://sunsite.unc.edu/pub/Linux/kernel/patches/misc/modreq_beep.tgz>`

The `modutils` package has an example program and kernel patch that supports calling an arbitrary external program to generate sounds when requested by the kernel.

Alternatively, with some sound cards you can connect the PC speaker output to the soundcard so that all sounds come from the sound card speakers.

6.32 What is VoxWare?

The kernel sound drivers support several different Intel-based Unix compatible operating systems, and can be obtained as a package separate from the Linux kernel. Up until February 1996 the author had called the software "VoxWare". Unfortunately this name has been registered by VoxWare Incorporated, and can not be used. On March 29th, 1996 Hannu Savolainen announced that the new name was Unix Sound System (USS) Lite.

The Unix Sound System (USS) is going to be a commercially available kernel sound driver for various Unix systems, sold by 4Front Technologies. A free version, known as USS/Lite will continue to be made freely available for Linux systems.

For more information see the 4Front Technologies Web page at `<http://www.4front-tech.com/>`.

6.33 Are Plug and Play sound card supported?

Linux does not yet fully support Plug and Play devices, but it is being worked on. In the mean time some users have reported success by either disabling the Plug and Play options in their BIOS, or by initializing the devices under Windows 95 before soft booting Linux.

6.34 Sox/Play/Vplay reports "invalid block size 1024"?

A change to the sound driver in version 1.3.67 broke some sound player programs which (incorrectly) checked that the result from the SNDCTL_DSP_GETBLKSIZE ioctl was greater than 4096. You should get a newer version of the program (if available) or fix it yourself. For the *Sox* `<ftp://sunsite.unc.edu/pub/Linux/kernel/sound/convert/Lsox-linux.tar.gz>` program, the following patch worked for me:

```
--- sbdsp.c.orig        Thu Feb 22 22:46:00 1996
+++ sbdsp.c     Thu Feb 22 22:51:18 1996
@@ -176,7 +176,7 @@
        }

        ioctl (dspfd, SNDCTL_DSP_GETBLKSIZE, \&abuf_size);
-       if (abuf_size < 4096 || abuf_size > 65536) {
+       if (abuf_size < 1) {
                if (abuf_size == -1)
                perror (dspname);
                else
```

6.35 Why does the sound driver have its own configuration program?

The sound driver supports many different configuration parameters. The `configure` program included with the sound driver checks for many dependencies between parameters. The tools used to configure the kernel don't support this level of functionality.

That said, the recent 1.3.x kernels do optionally allow using the standard kernel configuration tools with the sound driver. See the notes in the `CHANGELOG` file for the sound driver. This is still experimental and some options cannot be set this way.

6.36 The mixer settings are reset whenever I load the sound driver module!

You can build the sound driver as a loadable module and use `kerneld` to automatically load and unload it. This can present one problem - whenever the module is reloaded the mixer settings go back to their default values. For some sound cards this can be too loud (e.g. SB16) or too quiet. Markus Gutschke (`gutschk@uni-muenster.de`) found this solution. Use a line in your `/etc/conf.modules` file such as the following:

```
options sound dma_buffsize=65536 \&\& /usr/bin/setmixer igain 0 ogain 0 vol 75
```

This causes your mixer program (in this case `setmixer`) to be run immediately after the sound driver is loaded. The `dma_buffsize` parameter is just a dummy value needed because the option command requires a command line option. Change the line as needed to match your mixer program and gain settings.

If you have compiled the sound driver into your kernel and you want to set the mixer gains at boot time you can put a call to your mixer program in a system startup file such as `/etc/rc.d/rc.local`.

6.37 Only user root can record sound

By default the script in Readme.linux that creates the sound device files only allows the devices to be read by user `root`. This is to plug a potential security hole. In a networked environment, external users could conceivably log in remotely to a Linux PC with a sound card and microphone and eavesdrop. If you are not worried about this, you can change the permissions used in the script.

With the default setup, users can still play sound files. This is not a security risk but is a potential for nuisance.

7 References

If you have a sound card that supports a CD-ROM or SCSI interface, the Linux SCSI HOWTO and the Linux CD-ROM HOWTO have additional information that may be useful to you.

Hannu Savolainen has written a draft version of the *Hacker's Guide to VoxWare*. The latest version is draft 2, and can be found on `<ftp://nic.funet.fi/pub/OS/Linux/ALPHA/sound/>`.

The following FAQs are regularly posted to the usenet newsgroup news.announce as well as being archived at `<ftp://rtfm.mit.edu/pub/usenet/news.answers>`:

- PCsoundcards/generic-faq (Generic PC Soundcard FAQ)
- PCsoundcards/soundcard-faq (comp.sys.ibm.pc.soundcard FAQ)
- PCsoundcards/gravis-ultrasound/faq (Gravis UltraSound FAQ)
- audio-fmts/part1 (Audio file format descriptions)
- audio-fmts/part2 (Audio file format descriptions)

The FAQs also list several product specific mailing lists and archive sites. The following Usenet news groups discuss sound and/or music related issues:

- alt.binaries.sounds.* (various groups for posting sound files)
- alt.binaries.multimedia (for posting Multimedia files)
- alt.sb.programmer (Soundblaster programming topics)
- comp.multimedia (Multimedia topics)
- comp.music (Computer music theory and research)
- comp.sys.ibm.pc.soundcard.* (various IBM PC soundcard groups)

A Web site dedicated to multimedia can be found at `<http://viswiz.gmd.de/MultimediaInfo/>`. Creative Labs has a Web site at `<http://www.creaf.com/>`. MediaTrix has a Web site at `<http://www.mediatrix.com/>`.

The Linux mailing list has a number of "channels" dedicated to different topics, including sound. To find out how to join, send a mail message with the word "help" as the message body to majordomo@vger.rutgers.edu. (Note: at time of writing these mailing lists were severely overloaded and a replacement was being sought).

As mentioned several times before, the kernel sound driver includes a number of `Readme` files containing useful information about the sound card driver. These can typically be found in the directory `</usr/src/linux/drivers/sound>`.

The author of the kernel sound driver, Hannu Savolainen, can be contacted by email at hannu@voxware.pp.fi. He also has a World-Wide Web site at `<http://personal.eunet.fi/pp/voxware>`. The Web site is the best source for finding out the latest status of supported sound cards, known problems, and bug fixes.

The *Linux Software Map* (LSM) is an invaluable reference for locating Linux software. Searching the LSM for keywords such as *sound* is a good way to identify applications related to sound hardware. The LSM can be found on various anonymous FTP sites, including `<ftp://sunsite.unc.edu/pub/Linux/docs/LSM.gz>`.

The Linux Documentation Project has produced several books on Linux, including *Linux Installation and Getting Started*. These are freely available by anonymous FTP from major Linux archive sites or can be purchased in hardcopy format.

Finally, a shameless plug: If you want to learn a lot more about multimedia under Linux (especially CD-ROM and sound card applications and programming), check out my new book *Multimedia on Linux* published by O'Reilly and Associates (`<http://www.ora.com/>`). It should be available in the September 1996 time frame.