

Parallel Port Layer 3 Audio Decoder (lp3) Linux Driver Specification

Revision 2

January 5, 2000

© 1999 Nick Maddix

This document describes the interface between the lp3 Linux Device Driver and user-mode processes.

1.0 Introduction

The lp3 Linux Device Driver is a kernel module for Linux kernels that include “Parport” support. This character device driver provides effective sharing of the parallel port with other attached devices. Since lp3 has a “pass-through” port to which other parallel port devices can be attached, this sharing is essential to retaining full use of the parallel port used with lp3.

In addition, this driver provides support for more than one lp3 connected to the same computer (through the use of multiple parallel ports). Each lp3 can be accessed independently through it’s own device file (/dev/lp3-x) at the same time.

2.0 Parport Sharing

Parallel port sharing is accomplished through the use of the “Parport” functionality included with later Linux kernels. The Parport services provide drivers with a way to request and claim access to parallel port hardware for their own use. If the lp3 device driver needs to access a port, it requests it from the Parport driver and releases it when done so that other drivers can use the port.

Port requests (calls to `parport_claim()`) are made when the driver is opened. The port is released when the driver is closed.

Though this methodology may imply that the port is completely unavailable to other drivers while the lp3 driver is open, this is not the case. The port is also released whenever playback is explicitly paused (e.g. by a user pressing the pause button in a user-mode playback application). An additional “auto-pause” mode can be enabled (by `ioctl()` calls) that causes the driver to pause playback and release the port as soon as another driver wants to claim it.

A common example that illustrates this scheme is a situation in which a user is listening to music through lp3, but wants to print a document on a printer attached to the lp3. Starting the print job (using `lpr`, for example) will cause the lp3 driver to pause playback and release the port to the lp driver. When `lpr` has finished printing and releases the port, the lp3 driver reclaims it and resumes playback automatically.

3.0 Example Client Program

The lp3 device driver comes with an example client mp3 player called lp3client. Its source code (lp3client.c, .h) can be found with the driver source.

4.0 Building the Kernel Module

The lp3 device driver module is built using the supplied Makefile. Simply type 'make'.

5.0 Loading the Kernel Module

The lp3 device driver is loaded using the insmod command like any other module. The lp3 module requires the parport and parport_pc modules. Insmod will not load the lp3 module without these modules loaded.

After loading the lp3 module, it is wise to create device files based on the dynamically-allocated major number and the available lp3 minor numbers.

An example shell script, called 'loadlp3', is packaged with the driver source code. This script loads the module and takes care of setting up the device files properly.

6.0 /proc/lp3 File

When loaded, the lp3 kernel module maintains a file in the proc filesystem called 'lp3'. This file contains the device minor number to parport mapping of the detected lp3s on the system. It also shows the ID strings from each detected lp3. An example /proc/lp3 file is shown below:

```
0          parport0          lp3 v01.20 #000004C8
1          parport1          No lp3 detected
2          parport2          No lp3 detected
```

7.0 User-mode Interface

The interface to the lp3 device driver is that of a standard character device. The device file /dev/lp3-x (where x is a number corresponding to the parport port desired) is first opened with the open() system call. MPEG data is sent to the driver through the write() system call. Finally, the driver can be closed using the close() call.

7.1 open()

This call opens the lp3 driver, allocates buffer memory, looks for an attached lp3 device, and reads its ID string.

errno is set to the following codes upon error:

Code	Error
ENXIO	the minor number is out of range
EBUSY	the minor number is already open
EBUSY	the driver is unable to claim the parport
ENOMEM	the driver is unable to allocate FIFO memory

ENXIO	no lp3 is detected on the parport
-------	-----------------------------------

7.2 write()

This call writes MPEG data into the lp3 driver FIFO buffer. The data will be written out to the lp3 hardware FIFO as soon as it is ready to accept data.

errno is set to the following codes upon error:

Code	Error
EIO	A hardware error occurred
EAGAIN	the write would block

7.3 read()

This call reads the ID string of the attached lp3 device.

This call does not return an error code.

7.4 close()

This call frees all driver resources and closes the driver.

This call does not return an error code.

7.5 ioctl()

This call is used to configure and control the lp3 driver. See the next section for details.

errno is set to the following codes upon error:

Code	Error
EINVAL	the argument was invalid

7.6 select()

This call acts like the standard select() system call. The driver is always readable, and can only be written when its buffer is not full.

8.0 ioctl() Commands

There are currently 13 ioctl() commands accepted by the lp3 device driver. These commands are used to control the immediate operation and mode of the driver. They are defined in lp3.h.

Name	Description
LP3_IOCTLRESET	Reset lp3 and the driver
LP3_IOCTLFLUSH	Flush both lp3 and driver buffers

LP3_IOCTLPAUSE	Pause or resume playback
LP3_IOCTLVOL	Set left channel volume
LP3_IOCTLRVOL	Set right channel volume
LP3_IOCTLBASS	Set bass level
LP3_IOCTLTREB	Set treble level
LP3_IOCGLLEN	Get the number of bytes in the driver buffer
LP3_IOCSTATE	Get the driver state
LP3_IOCTLAUTOP	Turn on/off auto-pause mode
LP3_IOCTLAUTOR	Turn on/off auto-resume mode
LP3_IOCTLBITRATE	Set the bitrate of the current stream
LP3_IOCTLPERFLVL	Set the driver perflevel

8.1 LP3_IOCRESET

This command resets the attached lp3 as well as the driver state. The lp3 and driver FIFOs will be flushed, the volume will be set to a default level of -18dB , and bass and treble levels will be set to flat (0dB). Any playback in progress will immediately stop. LP3_IOCRESET takes no argument.

8.2 LP3_IOCFLUSH

This command immediately flushes both the lp3 and driver FIFOs. Any playback in progress will immediately stop until more data is sent to the driver (by a write() call). This command is useful for implementing a “Stop” user interface command. LP3_IOCFLUSH takes no argument.

8.3 LP3_IOCTLPAUSE

This command pauses or resumes playback immediately. It takes a single argument whose value should be either 1 or 0. 1 pauses playback, while 0 resumes playback where it left off.

In addition, when playback is paused, ownership of the parallel port is released to other drivers that may want it. This allows a user to press pause and then print a document.

This command is useful for implementing a “Pause” user interface command.

8.4 LP3_IOCTLVOL and LP3_IOCTLRVOL

These commands immediately set the volume level of the left and right output channels respectively. Each takes a numeric argument between 0 and 63, inclusive. 0 corresponds to full mute while 63 corresponds to full volume (0dB). Each step is 1dB (e.g. 62 is -1dB , 61 is -2dB , etc.).

8.5 LP3_IOCTLBASS and LP3_IOCTLTREB

These commands immediately set the bass and treble levels of the outgoing audio. Each takes a numeric argument between 0 and 30,

inclusive. 0 corresponds to a –15dB cut, while 30 corresponds to a +15dB boost. 15 corresponds to a flat (0dB) response. Each step is 1dB.

8.6 LP3_IOCGLLEN

This command returns the number of bytes in the driver FIFO. Its argument is a pointer to an unsigned long variable (unsigned long*) that is to receive the length.

8.7 LP3_IOCSTATE

This command returns the state of the driver. Its argument is a pointer to an integer variable (int*) that is to receive the state word. The driver state word returned is a bitwise combination of the following states (defined in lp3.h):

State	Description
PAUSED	Playback is paused
AUTOPAUSED	Playback was paused by a port claim

8.8 LP3_IOCTAUTOP and LP3_IOCTAUTOR

These commands turn on and off the driver auto-pause and auto-resume modes respectively. They take a single numeric argument which should be either 0 or 1 (off and on, respectively).

With auto-pause mode on, the driver will pause playback and release the parallel port if another driver (e.g. the lp driver) requests access to the port.

With auto-resume mode on, the driver will re-claim the parallel port and resume playback automatically when the parallel port is next available.

8.9 LP3_IOCTBITRATE

This command advises the lp3 device driver of the bitrate of the current MPEG stream. Bitrate defaults to 192kbps when the driver is open()ed. The argument should be between 8000 and 448000, inclusive.

The driver should be advised of the bitrate (or average bitrate in the case of VBR files) anytime a new file is played. This will assure that the driver will send the MPEG data to the lp3 hardware at the correct speed.

8.10 LP3_IOCTPERFLVL

This command allows slight tweaking of the performance of the driver. It may be necessary to set the perflvel if using an excessively fast or slow computer. The value ranges between 0 and 3, inclusive, and defaults to 2.

A perflevel of 3 tells the driver to send data at the highest rate it possibly can. This setting might be used on very slow ('386) computers.

Other perflevels extend the data cycle time to the hardware depending on the bitrate of the MPEG stream. A perflevel of 0 results in the longest cycles.