

## libgaudio

Written by Andreas Schiffler

Monday, 18 September 2006 16:37 - Last Updated Monday, 18 September 2006 19:46

---

Mixing engine for games as game audio library for multi-voice playback.

# libgaudio

Mixing engine for games as game audio library for multi-voice playback.

## DOWNLOAD

[\[ libgaudio-1.6.tar.gz \(596k\) \]](#)

[\[ libgaudio-1.6-samples.tar.gz \(5.95M\) \]](#)

## INTRO

Author:A. Schiffler, aschiffler [at] ferzkopp.net.  
Licenced under the GPL/MPL, see the file COPYING for details.

## README

libgaudio - Game-Audio Library

-----

(C) A. Schiffler, 1999-2001 under the dual MPL/GPL license.

Please see source code for details.  
Please see the file LICENSE for GPL info.

## libgaudio

Written by Andreas Schiffler

Monday, 18 September 2006 16:37 - Last Updated Monday, 18 September 2006 19:46

---

### ! Important Note !

Since Ver. 1.6 libgaudio uses 'libsoundfile' for all sample loading.  
For more info, see below.

### Introduction

=====

libgaudio is a library to facilitate easy incorporation of sound and sound effects in games. Samples are loaded into memory and playback is then triggered or stopped. The system mixes any number of concurrently playing samples together (up to a predefined maximum). Also possible is the inclusion of a background MP3 soundfile (external program `"mpg123"`; required). Since version 1.4 panning and volume control has been added. Since version 1.5 each playing sample can be controlled individually.

Currently this multithreaded library is set to mix a maximum of 12 concurrent 16bit signed stereo samples at 44100Hz. The included `wav2raw.sh` shell-script will take any file with the extension `.wav` and convert it into the proper `.raw` format used by libgaudio. The total number of samples that can be loaded is 256. The total size of samples depends on the main memory of the computer.

System load on my P3-450 is about 4% for the MP3-player and 0.5-4% for the `testaudio` application.

See the sample program `"testaudio.c"` for a demo implementation. Get the seperatly distributed file `"Sample.mp3"` (any `.mp3` will actually do, but it has to be called `Sample.mp3` for `testaudio.c` to use it) and activate its playback in the test application by editing the `Makefile`.

A second sample program called `"halloween.c"` shows how easy it can be to load multiple WAV files and play them back at random. This was a great effect during Halloween night. Start it up with the seperatly distributed `"ScarySounds"` WAV files for a cool effect. :-)

### Version History

=====

Ver. 1.0 - Wed Sep 22 14:15:01 CEST 1999

\* Initial release.

Ver. 1.1 - Wed Sep 29 16:41:01 CEST 1999

\* Changed audio format to 16bit/stereo/44.1KHz

## libgaudio

Written by Andreas Schiffler

Monday, 18 September 2006 16:37 - Last Updated Monday, 18 September 2006 19:46

---

\* Added MP3 background-playback function

Ver. 1.2 - Thu Mar 9 12:20:27 CET 2000

- \* Changed header to include C++ defines
- \* Moved Sample.mp3 out of the main distribution

Ver. 1.3 - Thu Apr 27 09:47:44 EDT 2000

- \* Thanks to Karl Robillard the following additions have been included in this version:
  - Added pauseAudio, unpauseAudio, stopAllSamples, and copySample functions.
  - Added audioValid and audioPaused query macros.
  - All functions handle a failed startAudio.
  - Added comments for each function in the source.
  - Added GAUDIO\_RATE to header.
  - Renamed tsample/tplaying structures to ga\_sample/ga\_playing to make the name more unique.
  - playBackground now specifies rate to mpg123 (it sometimes played too fast).
  - Re-wrote testaudio.c to take keyboard input.
- \* Updated README and testaudio.c

Ver. 1.4 - Sat Jul 8 22:08:19 EDT 2000

- \* Added stereo panning and volume control support.

Ver. 1.5 - Wed Jul 19 17:39:54 EDT 2000

- \* Changed licensing to a dual MPL/GPL license.
- \* Added function to load/copy only left or right channel of a sample (required for panning control).
- \* Added unique playing ID after each playSample() call for better control of individual playing samples.

Ver 1.6 - Tue Dec 12 14:47:11 EST 2000

- \* Added support for libsndfile (included).  
No more RAW audio loading required. Any WAV file can be loaded now.
- \* Added ScarySounds collection (distributed seperately) and halloween.c demo application.
- \* Added function to query MP3 stream playing time.

## Compilation

=====

Compile and install libsndfile.

```
cd libsndfile-0.0.22
./configure
make
make install
```

## libgaudio

Written by Andreas Schiffler

Monday, 18 September 2006 16:37 - Last Updated Monday, 18 September 2006 19:46

---

```
cd ..
```

Now, have a look at the comments in the libgaudio Makefile and maybe edit it.

To compile the library use  
make

When the library is to be linked into your program, add  
-lpthread -D\_REENTRANT\_  
for multithreading support plus the libsoundfile library  
-L/usr/local/lib -lsndfile

The MP3 functionality expects the mpg123 player at /usr/bin.  
To restart or queue background MP3s, the main program has to poll the status using  
checkBackground() and restart the MP3 player using playBackground(). libgaudio will  
not do this automatically.

Note that the samples have to be using the same rate as libgaudio (default 44.1kHz).

### Functions and Procedures

=====

```
int startAudio (struct gameaudio *ga, const char *device_name);
```

\* Start gameaudio system: open device (usually /dev/dsp), setup device parameters, initialize tables, start thread. Returns -1 on error.

```
int stopAudio (struct gameaudio *ga);
```

\* Stop gameaudio system: turn of all playing sounds, kill thread, close device. Returns -1 on error.

```
void pauseAudio( struct gameaudio* ga );
```

\* Mutes all audio playback. No return code.  
Check pause-state with the macro audioPaused(ga).

```
void unpauseAudio( struct gameaudio* ga );
```

\* Turn off muting of audio playback set by pauseAudio(). No return code.  
Check pause-state with the macro audioPaused(ga).

## libgaudio

Written by Andreas Schiffler

Monday, 18 September 2006 16:37 - Last Updated Monday, 18 September 2006 19:46

---

```
int loadSample (struct gameaudio *ga, const char *filename);
int loadSampleLeft (struct gameaudio *ga, const char *filename);
int loadSampleRight (struct gameaudio *ga, const char *filename);
```

\* Load a (signed 16bit stereo 44100Hz) sample into memory for use with the gameaudio system. Returns sample id or -1 on error.

The alternative functions loadSampleLeft() and loadSampleRight() will only load the left or the right channel of a stereo file. This function should help in creating sample sets from mono wave files that can be panned using the ctrlSample() function.

Since version 1.6 gaudio uses libsoundfile for sample loading. Several file formats are supported. For more information on the available formats refer to the documentation for libsoundfile. The only restriction to the sample is, that the sample-rate must correspond to libgaudio's rate (since no resampling is done) - which defaults to 44.1kHz. It is possible to load 8bit or mono samples.

```
int copySample( struct gameaudio* ga, void* data, int bytelen );
int copySampleLeft( struct gameaudio* ga, void* data, int bytelen );
int copySampleRight( struct gameaudio* ga, void* data, int bytelen );
```

\* Copy a (signed 16bit stereo 44100Hz) sample from a memory location &quot;data&quot; for use with the gameaudio system. Returns sample id or -1 on error.

The alternative functions copySampleLeft() and copySampleRight() will only copy the left or the right channel of a stereo set. This function should help in creating sample sets from sets that have equal volume on each channel to ones that can be panned using the ctrlSample() function.

```
int playSample (struct gameaudio *ga, int sample_id, int repeat_count);
```

\* Play sample referenced by id 'repeat\_count' times. if the repeat\_count is 0 playing will loop forever. The return code is a negative number corresponding to a unique playing ID or 0 on error. The playing id can be used to control this instance of the playing sample using ctrlSample() or stopSample().

```
void ctrlSample (struct gameaudio *ga, int sample_id, int volume, int pan);
```

\* Change playback parameters of sample with the give id (sample or playing id). Change volume of playing sound by adjusting volume parameter in the range 0-256. Adjust stereo channel panning using the pan variable in the range 0-256. At a value of 0 (default) the left channel plays left and the right channel

## libgaudio

Written by Andreas Schiffler

Monday, 18 September 2006 16:37 - Last Updated Monday, 18 September 2006 19:46

---

right. At a value of 128 we have center (mono) playback. At a value of 256 the left channel plays right and the right channel plays left.

```
void stopSample (struct gameaudio *ga, int sample_id);
```

\* Stop all playing samples with the given id (sample or playing id). No return code.

```
void stopAllSamples( struct gameaudio* ga );
```

\* Stop all playing samples (regardless of id). Background audio continues to play. No return code.

```
void playBackground(struct gameaudio *ga, const char *filename);
```

\* Start background playing of a MP3 file given in filename.

```
void stopBackground(struct gameaudio *ga);
```

\* Stop currently playing MP3 stream.

```
int checkBackground(struct gameaudio *ga);
```

\* Check if the background MP3 stream is playing (1=playing, 0=not playing).

```
float timeBackground(struct gameaudio *ga);
```

\* If the background stream is playing, this function will return the time of playback since start (in seconds) or 0.0 if the stream is not playing.

### NOTE:

There are two kinds of ID number used in the library. The `sample_id` is generated during the loading (or copying) of a sample into the libgaudio library. It uniquely identifies a specific sample and is always a positive number. The `playing_id` is a unique negative number that identifies an instance of a playing sample. Since the same sample can be triggered for playback several times, each time the `playSample()` function will return a different number. Both forms of id can be used to control a playing sample (as in `ctrlSample()` or `stopSample()`). Only the `sample_id` can be used to start a sample for playback (as in `playSample()`).

## **libgaudio**

Written by Andreas Schiffler

Monday, 18 September 2006 16:37 - Last Updated Monday, 18 September 2006 19:46

---

### **NOTE:**

To use panning effects, either create stereo wave files with only one channel used for the samples or load using the `loadSampleLeft()/loadSampleRight()` functions. Then control the pan value using `ctrlSample()`.

### **NOTE:**

See also the comments in the source code (`gaudio.c`) for even more gory details on how its done.