|  |  |  |
| --- | --- | --- |
| Bochs User Manual | | |
| [Prev](http://docs.google.com/using-save-restore.html) | Chapter 5. Using Bochs | [Next](http://docs.google.com/common-problems.html) |

5.6. Using sound

5.6.1. Sound basics

Bochs supports two types of soundcards (SB16 ISA and ES1370 PCI) and the legacy PC speaker. The lowlevel sound interface provides a set of functions for wave (PCM) input / output and MIDI output using the host sound system. A driver must not implement all of these functions. The following table shows the supported drivers and their features.

**Table 5-3. Sound lowlevel modules**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Wave input | MIDI output |
| alsa | use the default ALSA PCM device and the default ALSA sequencer on Linux | Yes | Yes |
| oss | use the Open Sound System on Linux and FreeBSD | Yes | Yes |
| osx | for output on Mac OS 9 and Mac OSX | No | Yes |
| sdl | for wave output on platforms supported by SDL (1.2.x or 2.x) | No | No |
| win | for output to the midi and wave mapper of Windows | Yes | Yes |

These special values are also valid for the sound driver:

* default - select platform-default sound driver.
* file - wave and MIDI output to file(s)
* dummy - no output at all

When compiling Bochs, the lowlevel sound support is activated if one of the soundcards is enabled (--enable-sb16 or --enable-es1370). The configure script detects the available drivers and sets up a platform-default one.

At runtime the lowlevel sound module will be loaded automatically if one of the sound devices is enabled in the bochsrc. The drivers and devices for wave input / output and MIDI output must be set up with the [sound](http://docs.google.com/bochsrc.html#BOCHSOPT-SOUND) option.

5.6.2. The PC speaker

The PC speaker is connected to the PIT 8254 timer #2 and generates a square wave beep with a frequency depending on the timer value. In Bochs the speaker is a separate plugin that is controlled by the two events "beep on" and "beep off". The "beep on" event is also called when the frequency changes while turned on.

The destination for the speaker output can be selected with the mode parameter of the [speaker](http://docs.google.com/bochsrc.html#BOCHSOPT-SPEAKER) option. Three choices are available:

* sound : the beep is generated by the square wave generator which is a part of the lowlevel sound support.
* system : only available on Linux and Windows. On Linux /dev/console is used for output and on Windows the Beep() function.
* gui : forwards the beep to the related gui methods (currently only used by the Carbon gui).

5.6.3. SB16 runtime configuration

The SB16 configuration parameters loglevel and dmatimer are available in the runtime configuration menu or dialog. In addition to this, there is a small program called **SB16CTRL** to change emulation settings from inside the simulation.

5.6.3.1. Runtime options loglevel and dmatimer

Unlike other devices, the SB16 emulation has it's own logfile and a loglevel parameter to control what should be printed there. See the [sb16](http://docs.google.com/bochsrc.html#BOCHSOPT-SB16) bochsrc option for details.

The dmatimer parameter controls the DMA timing for wave (PCM) input and output. When you get non-continuous sound this value can be ajusted to fix this. This needs a reasonably correct setting for the [cpu: ips](http://docs.google.com/bochsrc.html#BOCHSOPT-CPU-IPS) option.

5.6.3.2. SB16CTRL

The source for the SB16CTRL program that is used to modify the runtime behavior of the SB16 emulation is included in misc/sb16/. It is a C program that can be run from inside the emulation.

It currently supports the following commands:

**Table 5-4. Supported options for sb16ctl**

|  |  |
| --- | --- |
| Option | Description |
| -i *number* | Show the selected emulator info string, e.g. **sb16ctrl -i 3** to show how many patch translations are active. |
| -t *six numbers* | Load a translation into the translation table. The numbers are: "OldBankMSB,OldBankLSB,OldProgram,NewBankMSB,NewBankLSB,NewProgram". All values can be 0..127 or 255. 255 for "Old" values means *match any* and for "New" values means *don't change*, e.g. **sb16ctrl -t 255,255,0,255,255,32** to change patch 0 (Piano) to patch 32 (Acoustic Bass). |
| -r | Reset the patch translation table e.g. **sb16ctrl -r**. |
| -m *some numbers* | Upload the given numbers to the midi output device. Note that it should be a complete midi message, and also that it is subject to patch translation, e.g. **sb16ctrl -m 0x80,64,0** to send a note-off message to channel 0. |
| -f *filename* | Read in a file and execute the commands in it. These have the same format as the above commands, except that they don't have the dash "-" in front of them. Comment lines are supported and start with a hash sign "#". |
| -h | Show a brief summary of the commands. |

All numbers can be valid parameters to the strtol() function, so hex and octal notation is fine. They have to be delimited by either commas "," or slashes "/", spaces are not allowed.

The command line can have any number of commands. However, if none are given, "-f -" is assumed, which means commands are taken from stdin.

|  |  |  |
| --- | --- | --- |
| [Prev](http://docs.google.com/using-save-restore.html) | [Home](http://docs.google.com/index.html) | [Next](http://docs.google.com/common-problems.html) |
| Save and restore simulation | [Up](http://docs.google.com/using-bochs.html) | Common problems and what to do about them (Troubleshooting) |