Getting started with fluidsynth

Introduction

fluidsynth is a software music synthesizer that reads midi input events either from a midi piano keyboard or from a software application (e.g. midi sequencer) and then generates in realtime a musical audio output that corresponds to all the midi notes being played.

To work fluidsynth requires a Sound Font 2 file (.sf2 file) which contains all the audio waveforms for all of the different musical instruments sounds that can be produced by fluidsynth. The Sound Font files 'FluidR3_GM.sf2' and 'FluidR3_GS.sf2' work with fluidsynth and also have the advantage of having the Creative Commons License. These files are available in many Linux distributions in the packages 'fluid-soundfont-gm' and 'fluid-soundfont-gs' and they can also be downloaded from the net (try searching for 'FluidR3_GM.sf2'). The GM stands for General Midi which defines a standard mapping of MIDI patch numbers to musical instrument sounds.

Running fluidsynth

The easiest way test that fluidsynth is working correctly and to hear some midi music playing is to use the command line and to pass a Sound Font file and the midi file as the parameter. For example the following command line tests that fluidsynth is working on Ubuntu Linux.

```
fluidsynth /usr/share/sounds/sf2/FluidR3_GM.sf2 mymusicfile.mid
```

Normally you would not pass a midi file to fluidsynth but instead use another application to pass midi events to fluidsynth. In this case you would start fluidsynth the following parameters.

```
fluidsynth /usr/share/sounds/sf2/FluidR3_GM.sf2
```

For further examples on how to start fluid synth see ExampleCommandLines.

fluidsynth can also started using QSYNTH which provided a graphical user interface to the synth

Getting Help

For help with starting fluidsynth see the UserManual or type "man fluidsynth" on the command line. Typing "fluidsynth --help" lists all the command line options.

Once fluidsynth has started up type "help" to access the built in help. Type "settings" to show all the current settings. Type "info audio.driver" to list the available hardware devices.

User Manual

This document has the same layout as the manpage, but it contains links to other pages for more information: FluidFeatures and FluidSettings.

NAME

FluidSynth - a SoundFont synthesizer

SYNOPSIS

fluidsynth [options] [SoundFonts] [midifiles]

DESCRIPTION

FluidSynth is a real-time MIDI synthesizer based on the SoundFont(R) 2 specifications. It can be used to render MIDI input or MIDI files to audio. The MIDI events are read from a MIDI device. The sound is rendered in real-time to the sound output device. See FluidFeatures for a comprehensive list of features implemented and working.

The easiest way to start the synthesizer is to give it a SoundFont on the command line: 'fluidsynth soundfont.sf2'. fluidsynth will load the SoundFont and read MIDI events from the default MIDI device using the default MIDI driver. Once FluidSynth is running, it reads commands from the stdin. There are commands to send MIDI events manually, to load or unload SoundFonts, and so forth. All the available commands are discussed below.

FluidSynth can also be used to play a list of MIDI files. Simply run FluidSynth with the SoundFont and the list of MIDI files to play. In this case you might not want to open the MIDI device to read external events. Use the -n option to deactivate MIDI input. If you also want to deactivate the use of the shell, start FluidSynth with the -i option: 'fluidsynth -ni soundfont.sf2 midifile1.mid midifile2.mid'.

Run fluidsynth with the --help option to check for changes in the list of options.

OPTIONS

fluidsynth accepts the following options:

-a, --audio-driver=[label]

The audio driver [alsa,jack,oss,dsound,...]

-C, --chorus

Turn the chorus on or off [0|1|yes|no, default = on]

-c, --audio-bufcount=[count]

Number of audio buffers

-d, --dump

Dump incoming and outgoing MIDI events to stdout

-f, --load-config

Load command configuration file (shell commands)

-G, --audio-groups

Defines the number of LADSPA audio nodes

-g, --gain

Set the master gain [0 < gain < 10, default = 0.2]

-h, --help

Print out this help summary

-i, --no-shell

Don't read commands from the shell [default = yes]

-j, --connect-jack-outputs

Attempt to connect the jack outputs to the physical ports

-K, --midi-channels=[num]

The number of midi channels [default = 16]

-L, --audio-channels=[num]

The number of stereo audio channels [default = 1]

-l, --disable-lash

Don't connect to LASH server

-m, --midi-driver=[label]

The name of the midi driver to use [oss,alsa,alsa_seq,...]

-n, --no-midi-in

Don't create a midi driver to read MIDI input events [default = yes]

-o

Define a setting, -o name=value ("-o help" to dump current values). See FluidSettings for details

-p, --portname=[label]

Set alsa sequencer and coremidi port name

-R, --reverb

Turn the reverb on or off [0|1|yes|no, default = on]

-r, --sample-rate

Set the sample rate

-s, --server

Start FluidSynth as a server process

-V, --version

Show version of program

-v, --verbose

Print out verbose messages about midi events

-z, --audio-bufsize=[size]

Size of each audio buffer

SHELL COMMANDS

GENERAL

help

Prints out a summary of the main commands

help help

Prints out list of other help topics (type "help <topic>")

quit

Quit the synthesizer

SOUNDFONTS

load filename

Load a SoundFont

unload number

Unload a SoundFont. The number is the index of the SoundFont on the stack.

fonts

Lists the current SoundFonts on the stack

inst number

Print out the available instruments for the SoundFont.

MIDI MESSAGES

noteon channel key velocity

Send a note-on event

noteoff channel key

Send a note-off event

cc channel ctrl value

Send a control change event

prog chan num

Send program-change message

select chan sfont bank prog

Combination of bank-select and program-change

channels

Print out the presets of all channels.

AUDIO SYNTHESIS

gain value

Set the master gain (0 < gain < 5)

interp num

Choose interpolation method for all channels

interpc chan num

Choose interpolation method for one channel

REVERB

reverb [0|1|on|off]

Turn the reverb on or off

rev_preset num

Load preset num into the reverb unit

rev_setroomsize num

Change reverb room size

rev_setdamp num

Change reverb damping

rev_setwidth num

Change reverb width

rev_setlevel num

Change reverb level

CHORUS

chorus [0|1|on|off]

Turn the chorus on or off

cho_set_nr n

Use n delay lines (default 3)

cho_set_level num

Set output level of each chorus line to num

cho_set_speed num

Set mod speed of chorus to num (Hz)

cho_set_depth num

Set chorus modulation depth to num (ms)

MIDI ROUTER

router_default

Reloads the default MIDI routing rules (input channels are mapped 1:1 to the synth)

router_clear

Deletes all MIDI routing rules.

router_begin [note|cc|prog|pbend|cpress|kpress]

Starts a new routing rule for events of the given type.

router_chan min max mul add

Limits the rule for events on min <= chan <= max. If the channel falls into the window, it is multiplied by 'mul', then 'add' is added.

router_par1 min max mul add

Limits parameter 1 (for example note number in a note events). Similar to router_chan.

router_par2 min max mul add

Limits parameter 2 (for example velocity in a note event). Similar to router_chan.

router_end

Finishes the current rule and adds it to the router.

Router examples

router_clear

router_begin note

router_chan 0 7 0 15

```
router_end
```

Will accept only note events from the lower 8 MIDI channels. Regardless of the channel, the synthesizer plays the note on ch 15 (synthchannel=midichannel*0+15).

```
router_begin cc

router_chan 0 7 0 15

router_par1 1 1 0 64

router_end
```

Configures the modulation wheel to act as sustain pedal (transforms CC 1 to CC 64 on the lower 8 MIDI channels, routes to ch 15).

LADSPA

LADSPA processing is currently disabled because its implementation has stability issues.

ladspa_clear

> Resets LADSPA effect unit to bypass state

ladspa_add lib plugin n1 <- p1 n2 -> p2 ...

> Loads and connects LADSPA plugin

ladspa_start

> Starts LADSPA effect unit

ladspa_declnode node value

> Declares control node `node' with value 'value'

ladspa_setnode node value

> Assigns 'value' to 'node'

AUTHORS

Peter Hanappe <hanappe@...>

Markus Nentwig <nentwig@...>

Antoine Schmitt <as@...>

Josh Green <jgreen@...>

Stephane Letz <letz@...>

Please check the AUTHORS and THANKS files for all credits

DISCLAIMER

SoundFont(R) is a registered trademark of E-mu Systems, Inc.

FluidSynth Settings

The actual list of available settings, their options and min, max and default values will depend on build settings.

AUDIO DRIVER SETTINGS

audio.driver

String. ['pulseaudio', 'alsa', 'oss', 'coreaudio', 'dsound', 'portaudio', 'sndman', 'jack', 'file']. Name of the audio driver to use. Defaults to first available (built-in) driver from the list.

audio.periods

Integer. [min=2, max=64]. Number of periods tried by the alsa, oss, dsound and jack audio drivers. A warning will be shown when the actual value is different than the requested one. Default value is 8 for Windows and MacOS9, 16 for all other builds.

audio.period-size

Integer. [min=64, max=8192]. Period size tried by the pulseaudio, alsa, oss, dsound and jack audio drivers. A warning will be shown when the actual value is different than the requested one. Default value is 512 for Windows, 64 for all other builds.

audio.sample-format

String. ['16bits', 'float']. Sample format tried by the alsa, oss, dsound and jack audio drivers. Default value: '16bits'.

audio.input-channels

Integer. [min=0, max=2, def=0]. Obsolete.

audio.output-channels

Integer. [min=2, max=32, def=2]. Obsolete.

ALSA

audio.alsa.device

String. ALSA audio device name. Default value: 'default'.

JACK

audio.jack.autoconnect

Integer. [0, 1]. If equal to 1 then auto-connect audio output ports to the first physical input ports found. Default value: 0.

audio.jack.id

String. JACK identifier to use. Default value: 'fluidsynth'.

audio.jack.multi

String. ['no', 'yes']. If 'yes' then create separate output ports for reverb and chorus effects. There will be as many left/right port pairs as defined by "synth.audio-channels" + one left/right pair for reverb + one left/right pair for chorus effect. Default value: 'no'.

DSOUND

audio.dsound.device

String. DSound device name. Default value: 'default'.

OSS

audio.oss.device

String. OSS device name. Default value: '/dev/dsp'.

PULSEAUDIO

audio.pulseaudio.server

String. PulseAudio server. Default value: 'default'.

audio.pulseaudio.device

String. PulseAudio device name. Default value: 'default'.

PORTAUDIO

audio.portaudio.device

String. PortAudio device name. Default value: 'PortAudio Default'.

FILE

audio.file.name

String. File name where data will be written to. Default value: 'fluidsynth.raw'.

MIDI SETTINGS

midi.driver

String. ['alsa_seq', 'alsa_raw', 'oss', 'winmidi', 'midishare', 'coremidi']. Name of the MIDI driver to use. Defaults to first available (built-in) driver from the list excluding alsa_raw.

ALSA_SEQ

midi.alsa_seq.id

String. Client identifier, used to generate client and port names automatically. Default value: pid.

midi.alsa_seq.device

String. ALSA sequencer device name. Default value: 'default'.

midi.portname

String. MIDI client and port name. Default value: *(automatically generated).*

ALSA_RAW

midi.alsa.device

String. MIDI device name. Default value: 'default'.

OSS

midi.oss.device

String. OSS MIDI device name. Default value: '/dev/midi'.

WINMIDI

midi.winmidi.device

String. WinMIDI device name. Default value: 'default'.

COREMIDI

midi.coremidi.id

String. Client identifier, used to generate a distinguished client name automatically. Default value: pid.

midi.portname

String. MIDI port name. Default value: *(automatically generated)*.

SYNTH SETTINGS

synth.reverb.active

String. ['yes', 'no']. When set to "yes" the reverb effects module is activated. Otherwise, no reverb will be added to the output signal. Note that when the reverb module is active, the amount of signal send to the reverb module depends on the "reverb send" generator defined in the SoundFont. Default value: 'yes'.

synth.chorus.active

String. ['yes', 'no']. When set to "yes" the chorus effects module is activated. Otherwise, no chorus will be added to the output signal. Note that when the reverb module is active, the amount of signal send to the chorus module depends on the "chorus send" generator defined in the SoundFont. Default value: 'yes'.

synth.ladspa.active

String. ['no', 'yes']. When set to "yes" the LADSPA subsystem will be called. This subsystem allows to load and interconnect LADSPA plugins. The output of the synthesizer is processed by the LADSPA subsystem. Note that the synthesizer has to be compiled with LADSPA support. Currently disabled because of some stability issues. Default value: 'no'.

synth.midi-channels

Integer. [min=16, max=256]. This setting defines the number of MIDI channels of the synthesizer. The MIDI standard defines 16 channels, so most hardware keyboards are limited to 16. If you plan to use the synthesizer as a plugin in an application, it might be interesting to set the number of channels to a larger value. In this case you can program a greater number of presets. One MIDI port is created for every 16 channels. Default value: 16.

synth.midi-bank-select

String. ['gm', 'gs', 'xg', 'mma']. This setting defines the MIDI bank select style (GM, GS, XG or MMA. Default=GS). This setting is used only for synth initialization. The behavior of each option is the following:

- GM ignores CC0 and CC32 msgs
- GS CC0 becomes the channel bank, CC32 is ignored
- XG CC32 becomes the channel bank, CC0 is ignored
- MMA bank = CC0*128+CC32

synth.polyphony

Integer. [min=16, max=4096]. The polyphony defines how many voices can be played in parallel. The number of voices is not necessarily equivalent to the number of notes played simultaniously. Indeed, when a note is struck on a specific MIDI channel, the preset on that channel may created several voices, for example, one for the left audio channel and one for the right audio channels. The number of voices activated depends on the number of instrument zones that fall in the correspond to the velocity and key of the played note. Default value: 256.

synth.sample-rate

Float. [min=22050.000, max=96000.000]. The sample rate of the audio generated by the synthesizer. The audio driver will try to use the same sample rate, but when not possible it will just show a warning and continue with not matching sample rates. Default value: 44100.000.

synth.verbose

String. ['no', 'yes']. When set to "yes" the synthesizer will print out information about the received MIDI events to the stdout. This can be helpful for debugging. This setting can not be changed after the synthesizer has started. Default value: 'no'.

synth.gain

Float. [min=0.000, max=10.000]. The gain is applied to the final or master output of the synthesizer. It is set to a low value by default to avoid the saturation of the output when random MIDI files are played. Default value: 0.200.

synth.audio-channels

Integer. [min=1, max=256]. By default, the synthesizer outputs a single stereo signal. Using this option, the synthesizer can output multichannel audio. Number of left/right output pairs when "audio.jack.multi" is enabled. There's a practical limit of 128 channels in the source code (FIXME). Default value: 1.

synth.audio-groups

Integer. [min=1, max=256]. By default, the synthesizer outputs a single stereo signal. Using this option, the synthesizer can output multichannel audio. This value should be the same as "synth.audio-channels" unless LADSPA is enabled, then it will be the number of inputs to the LADSPA subsystem. There's a practical limit of 128 groups in the source code (FIXME). Default value: 1.

synth.effects-channels

Integer. [min=2, max=2]. Obsolete.

synth.dump

String. ['no', 'yes']. Obsolete.

SHELL SETTINGS

shell.port

Integer. [min=1, max=65535]. Port used in server mode (not working for Windows, see ~~#20~~). Default value: 9800.

shell.prompt

String. Prompt to use by the shell. Default value: .

Example Command Lines to start fluid synth

Introduction

This page gives examples of how to start fluidsynth from the command line with different configurations. Please feel free to add the configuration that works best for you.

fluidsynth with JACK

To be completed.

fluidsynth on NetBooks and low performance computers

First you need to reduce the CPU usage which helps reduce the chances of data under-run which causes the audio cut out. This can be done by turning off the Reverb and Chorus with the flags '-C0 -R0' and also by halving the sample rate with the flag '-r22050'. Changing the sample rate does not work with the alsa hw layer so use the plug layer instead. For example this command line works quite well on a eeePC 901 NetBook .

```
fluidsynth -C0 -R0 -r22050 -l -a alsa -o audio.alsa.device=plughw:0
```

If you get problems with unsteady playback or the audio cutting out then try closing all other programs,

turning off your wireless network and unpluging any network cable.

SoundFont is a file format for sample based instrument sounds. You will need a SoundFont to use FluidSynth.

If you just need to play General Midi files, these SoundFonts are known to work well with FluidSynth:

- S. Christian Collins GeneralUser GS - 30 MB
- Fluid (R3) General MIDI SoundFont (GM) - 140 MB

More download sites for SoundFonts:

- Hammersound - A nice resource for downloading free SoundFont instrument files.
- ResonanceDB (Currently down, until a new hosting location is found)

Developer Resources

The SoundFont format was originally created by Creative Labs and EMU Systems and used in the SoundBlaster AWE 32 and later cards. There are now many other hardware platforms and software synthesizers supporting this format. SoundFont 2.0 and later are open formats and the specification is freely available.

- Wikipedia SoundFont page - Good overview of SoundFont format and other resources.
- SoundFont 2.4 specification - PDF document describing SoundFont format technical details.
- Creative Labs Developer Documentation - Specifications, docs, SF2 test files, etc.

Low latency tips

There are several FluidSynth options which affect audio latency. Many of these depend on what audio driver is being used. Additionally the operating system being used, how it is setup and what audio card you have will limit the lowest artifact free latency that can be achieved.

Specifying the audio driver

To specify the audio driver which FluidSynth will use, supply the -a DRIVER option, for example:

```
fluidsynth -a alsa
```

Audio buffer size and count

These options are used for the majority of audio drivers that FluidSynth supports. An exception is that the Jack driver does not use the audio buffer size or count options.

Command line switches and equivalent FluidSynth settings (in parenthesis):

- -r=RATE (synth.sample-rate): Sample rate
- -c=NUM (audio.periods): Set the number of audio buffers
- -z=SIZE (audio.period-size): Set audio buffer size

The sample rate sets the native sample rate that FluidSynth synthesizes to. Typical values include 44100 and 48000.

Note: Currently the actual sample rate being used by the audio driver may differ, in which case the synth will be out of tune. Will be fixed in a future version of FluidSynth. Ensure that the internal FluidSynth sample rate matches that of your driver for proper operation.

The audio buffer count and size sets the values of these parameters used by the audio driver. Total latency in samples is NUM * SIZE. To calculate latency in seconds use NUM * SIZE / RATE. For example: 2 * 256 / 48000 = ~10ms latency. Suggested values for NUM is 2 or 3. Suggested values for SIZE include: 64, 128, 256, 512, 1024. Non power of 2 values can also be used and sound cards have different limits on this value.

ALSA specific tips

ALSA is a pretty flexible audio system and is the de-facto standard on Linux systems.

Specifying what ALSA device to use to playback to, can have a huge affect on achievable low latency. The hardware device layer is the best, followed by the plug hardware layer. Using dmix is not recommended for live playback, since it typically has rather high latency response. (Unfortunately specifying the hardware layer may bypass all of the desktop volume controls.)

To use the hardware layer, specify the ALSA audio device in the form "hw:N" where N is the card number. For example:

```
fluidsynth -a alsa -o audio.alsa.device=hw:0
```

If FluidSynth fails to initialize your sound card, you may need to specify a different sample rate (some sound cards only operate at fixed rates). Use the -o synth.sample-rate=RATE or -r=RATE command line options. Playing with the audio buffer size and count may also be required. If your sound card does not support 16 bit audio (for example it is fixed to 24 bit), then you will need to use the plughw layer as FluidSynth currently does not support 24 bit directly.

To use the plug hardware layer, specify the ALSA audio device in the form "plughw:N" where N is the card number. For example:

```
fluidsynth -a alsa -o audio.alsa.device=plughw:0
```

The advantage to the plug hardware layer is that it will do sample conversion as necessary, though this will require additional CPU, so the hw layer is preferred if possible.

Linux kernel

The version of the Linux kernel being used and its configured build time configuration options can have a pretty dramatic effect on achievable artifact free low latency. This can be an art in and of itself. Some people use RT kernels which may or may not have a lowlatency patch applied and usually have full preemption enabled.

FIXME: Add more info and/or links to Linux kernel tuning.