SunVox User Manual

2021.03.02

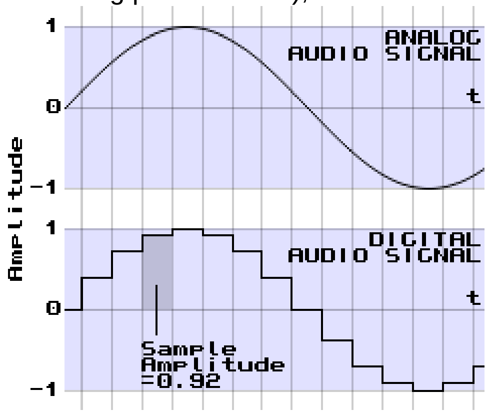
What is SunVox

SunVox is a small, fast and powerful modular synthesizer with pattern based sequencer (tracker). It is a tool for those people who like to compose music wherever they are, whenever they wish. On any device. On any system.

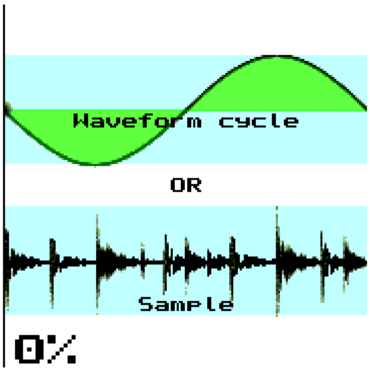
Official SunVox page

List of terms

- **Amplitude** - value (position on the Y-axis) of one sample of the digital audio signal, or the maximum value of the sound wave displacement (relative to zero) during the period (if we are talking about oscillations of the generator);
- **Audio signal** - representation of sound in the form of a graph (wave), process of changing the amplitude over time; in analog audio, amplitude is the voltage level; in digital audio, amplitude is a discrete value (usually a 32-bit floating point number);



- **Automation** - automatic change of project parameters during playback; for example, the program can smoothly decrease/increase the filter frequency, or change the tempo; the source of the automation commands is either a pattern, or one of the special modules (MultiCtl, Pitch2Ctl, Sound2Ctl, Velocity2Ctl);
- **Balance** - relative gain of the left and right channels; examples:
    - balance = -128: left channel volume = 100%; right channel volume = 0%;
    - balance = 0: left channel volume = 100%; right channel volume = 100%;
    - balance = 128: left channel volume = 0%; right channel volume = 100%.
- **BPM** - number of Beats Per Minute.
- **Envelope** - graph of the dependence of some parameter (volume, frequency, panning, etc.) on time; in general, any sound wave is an envelope - the audio signal can be converted into control commands using the Sound2Ctl module.
- **LFO** - Low Frequency Oscillator which is used to modulate some synth/effect parameters (frequency, volume, panning, etc.);
- **Panning** - the sound location in the left-right stereo space; if the sound source is mono, then this parameter is equivalent to Balance (described above).
- **Phase** - position of a point in time on a waveform cycle (in Generator) or on a sample. Usually this means the initial phase, not the current position.



- **Sample**:
    - one unit of sound data - represents an amplitude of the digital audio signal at a point in time;
    - portion of some sound in digital form: set of the values (amplitudes) described above; it can be very long (loop or the whole audio track) and very short (one note of some instrument).
- **Sample rate** - number of samples per second.
- **Sustain** - suspension point (until the Note OFF) on the envelope.
- **Tick** - smallest time interval in SunVox pattern. 1 beat = 24 ticks.
- **TPL** - number of Ticks Per Line.
- **Velocity** - universal parameter combining such types of expression as the velocity (how fast or hard the keys are pressed) and the aftertouch (amount of force on held-down key). It can be changed at any time in the life cycle of a single note.

Minimum system requirements

Windows: 2000 (x86); 7 (x86_64).

Linux: any modern distribution + libsdl2 (you can install this library using apt-get or some other package manager).

macOS: 10.9 (x86_64).

Android: 4.1.

iOS: 9.

Windows CE: 3.0 (ARM).

Installing

Windows:

1. Download SunVox for Windows (ZIP archive).
2. Unpack this archive to the user folder (with read and write permissions; for example, your own folder on the external flash drive).
3. You will see the new directory **sunvox** after unpacking - open it, and go to the **sunvox/windows_x86** (for 32-bit system) or **sunvox/windows_x86_64** (for 64-bit system) directory.
4. You will find several executable files in the last folder:
    - sunvox - version with 32-bit audio engine;
    - sunvox_lofi_console - version with 16-bit audio engine for slow devices.

Linux:

1. Download SunVox for Linux (ZIP archive).
2. Unpack this archive to the home user directory (/home/username).
3. You will see the new directory **sunvox** after unpacking - open it, and go to the **sunvox/linux**_X directory, where the X is the name of the architecture of your device.
4. You will find several executable files in the last folder:
    - sunvox - version with 32-bit audio engine;
    - sunvox_lofi - version with 16-bit audio engine for slow devices.
5. In case of some errors: open Terminal, go to the sunvox directory (use **cd** command), enter **chmod a+x sunvox** and then **./sunvox** ; if still not working - copy all error messages from the Terminal and send it to

nightradio@gmail.com.

Here is the instruction for Raspberry Pi users

macOS:

1. Download SunVox for macOS (ZIP archive).
2. You will see the new directory **sunvox** after unpacking; move it to some another user-created directory; don't use the Applications folder.
3. Go to the **sunvox/sunvox/macos** directory.
4. Read this important information for the macOS 10.12+ users
5. Launch SunVox.

Android:

- get SunVox on Google Play.

iOS:

- get SunVox on App Store.

Windows CE:

1. Download SunVox for Windows CE (ZIP archive).
2. Unpack it to any folder on your computer.
3. You will see the new directory **sunvox** after unpacking - open it, and go to the **sunvox/wince** directory.
4. You will find the sunvox.cab file in the **wince** folder - just copy and install this file on the Windows CE device.
5. Some old Windows CE devices has no GAPI pre-installed - this library is required for SunVox; you can try to get it here.

Comparison of different versions

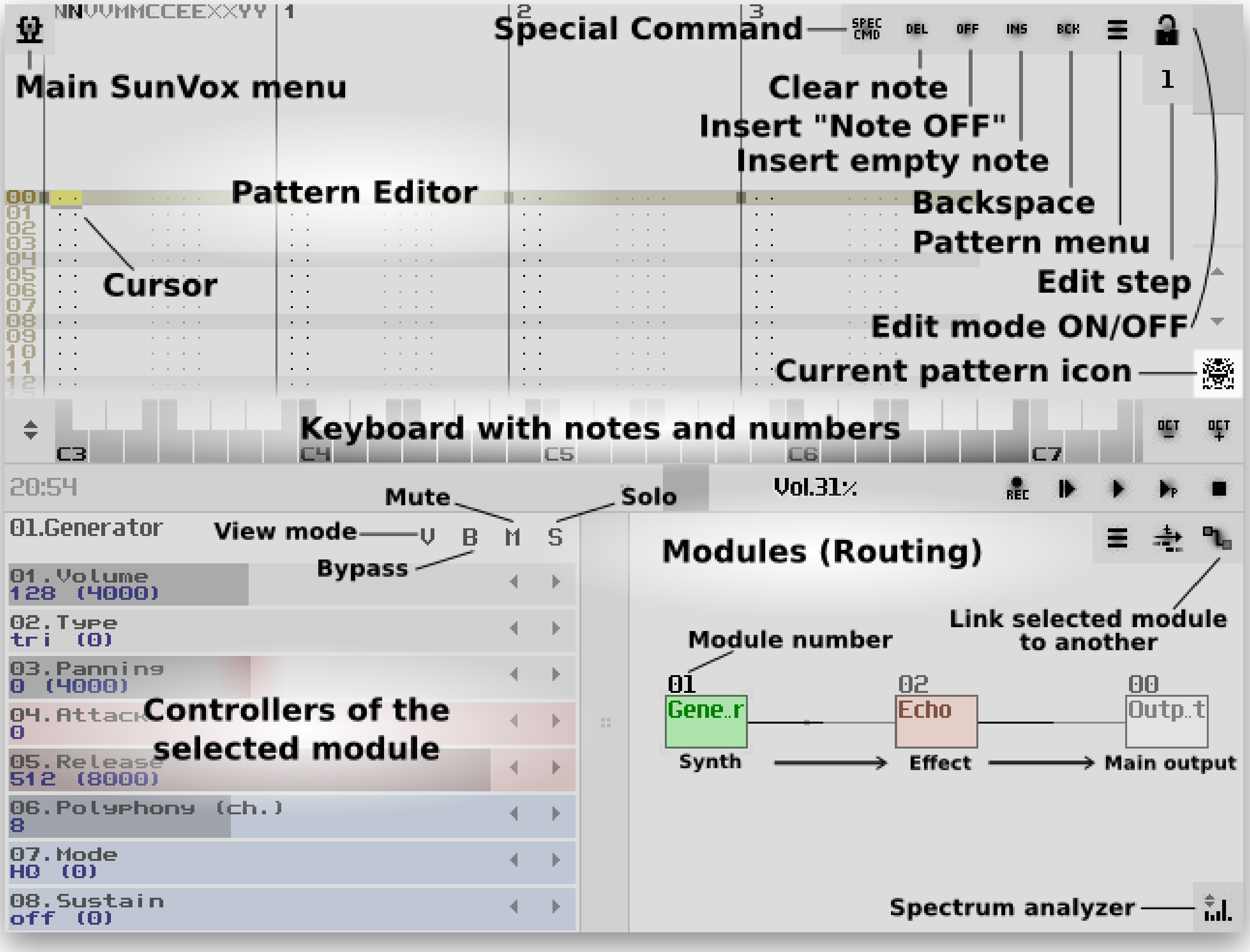
System	Supported architectures	Sound engine format	Multi-touch	GPIO
Linux	x86, x86_64, ARM (Maemo-based devices, Raspberry Pi, PocketCHIP, etc.), ARM64 (PINE64 etc.)	32-bit floating-point / 4,12 fixed-point	YES	YES
Windows	x86, x86_64	32-bit floating-point / 4,12 fixed-point	YES	
Windows CE	ARM	4,12 fixed-point		
macOS	x86_64	32-bit floating-point		
Android	ARM, ARM64, x86	32-bit floating-point	YES	YES
iOS	ARM, ARM64	32-bit floating-point	YES	

As you can see, there are two sound engine formats in SunVox. Let's look at them in detail.

- **32-bit floating-point** - using high quality 32-bit (and 64-bit in some cases) floating-point computations for the sound synthesis. Math coprocessor (FPU) is required. Can be used on most of modern computers.
- **4,12 fixed-point** - using low quality fixed-point (4 integer bits and 12 fractional bits) computations. Specially for the slow computers without FPU. Such version of the engine will be much faster than the 32-bit one, but a small amount of the quantization noise will be present in sound. In this case the 32-bit WAV export may be with better sound quality due to the absence of the signal clipping.

In some cases the sound engine format will be selected automatically. But in some cases (in Linux, for example) you can choose it manually: there are several versions of SunVox with different engines.
You can always find the information about the current engine format in the SunVox Main Menu -> About window.

Interface description



01.Generator

01.Volume
128 (4000)

02.Type
tri (0)

03.Panning
0 (4000)

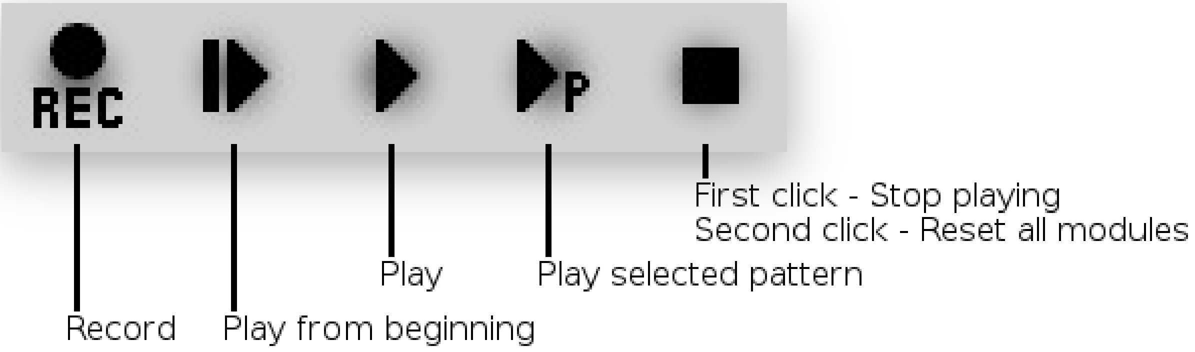
04.Attack
0

05.Release
512 (8000)

06.Polyphony (ch.)
8

07.Mode
HQ (0)

08.Sustain
off (0)



The SunVox interface is divided into the several rectangular areas (windows):

- pattern editor;
- musical keyboard;
- controllers of the selected module + modules (routing);
- timeline (select the pattern here and its content will be shown in the pattern editor at the top).

Each area has its own menu with different options and functions - use the button with three horizontal lines to open this menu. Click on some area sets the focus on this area.

First click on the Stop button - stop playing.Second click - reset all SunVox activity and switch the engine to standby mode.

Keyboard shortcuts

You can change it in the Main Menu -> Preferences -> Interface -> Shortcuts. Any keys and MIDI-buttons can be assigned to any SunVox action (from the list below).

General:

Keys	Description
CTRL + Z	undo
CTRL + Y or SHIFT + CTRL + Z	redo
CTRL + N	new project or object (module/pattern/...)
LEFT,RIGHT,UP,DOWN, PAGEUP,PAGEDOWN, HOME,END,TAB	navigation
SHIFT + UP/DOWN/LEFT/RIGHT	selection
CTRL + (	selection begin
CTRL +)	selection end
CTRL + A	select all
CTRL + X or SHIFT + DELETE	cut
CTRL + C	copy
CTRL + V or SHIFT + INSERT	paste
CTRL + D	duplicate / clone
INSERT (or Command+I on Mac)	insert an empty note and shift the pattern content down; or just insert something
BACKSPACE	delete previous note and shift the pattern content up; or just delete something
DELETE (or Fn+Backspace on Mac)	delete

Application:

Keys	Description
ESC	exit the application

Project:

Keys	Description
CTRL + SHIFT + N	new project
CTRL + O	load project
CTRL + S	save project

CTRL + B	save project to BACKUP.sunvox

Musical keyboard:

Keys	**Description**
Z,S,X,D,C,V,G,B,H,N,J,M,'<',L,'>',';','/' Q,2,W,3,E,R,5,T,6,Y,7,U,**I**,9,O,0,P,[,=,],ENTER	notes
F1...F8	set octave number
SHIFT +)	octave up
SHIFT + (	octave down

Pattern editor:

Keys	**Description**
SPACE	edit mode ON/OFF
CTRL + '='	increase the edit step
CTRL + '-'	decrease the edit step
CAPSLOCK or '~'	insert "Note OFF" (==)
K	insert a special command "Set Pitch" (SP)
SHIFT + K	insert a special command "Previous Track" (<<)
CTRL + M	paste and mix
CTRL + T	select track
CTRL + I	interpolate values
CTRL + U	interpolate velocity
SHIFT + '='	transpose up (+1 semitone)
SHIFT + '-'	transpose down (-1 semitone)
SHIFT + ']'	transpose octave up (+12 semitones)
SHIFT + '['	transpose octave down (-12 semitones)
CTRL + P	place selected events evenly
SHIFT + 7	cyclic shift up
SHIFT + 8	cyclic shift down
not assigned	remap

Modules (routing):

Keys	Description
CTRL + R	randomize module controllers
SHIFT + mouse movement	module link/unlink
SHIFT + controller value change	write the value to the pattern
SHIFT + >	next module
SHIFT + <	previous module
CTRL + >	next synth
CTRL + <	previous synth
CTRL + RIGHT	next module horizontally
CTRL + LEFT	previous module horizontally
CTRL + DOWN	next module vertically
CTRL + UP	previous module vertically
CTRL + 1	toggle mute
CTRL + 2	toggle solo
CTRL + 3	toggle bypass
CTRL + 4	unmute all modules
CTRL + F	find a module
SHIFT + scroll wheel	change the size of all modules
CTRL + scroll wheel	change the size of the selected module

Timeline:

Keys	Description
F9	play/stop
F10	play from beginning
F11	play pattern
F12	stop
SHIFT + F9	record start/stop
SHIFT + F10	play from the pattern cursor
SHIFT + F12	go to beginning
not assigned	slice

not assigned	toggle mute
not assigned	toggle solo
not assigned	unmute all patterns
not assigned	set restart position
not assigned	snap go grid ON/OFF

Touch Theremin:

Keys	**Description**
SHIFT + touch	write a microtone (Set Pitch XXYY command) to the pattern

Basic concepts

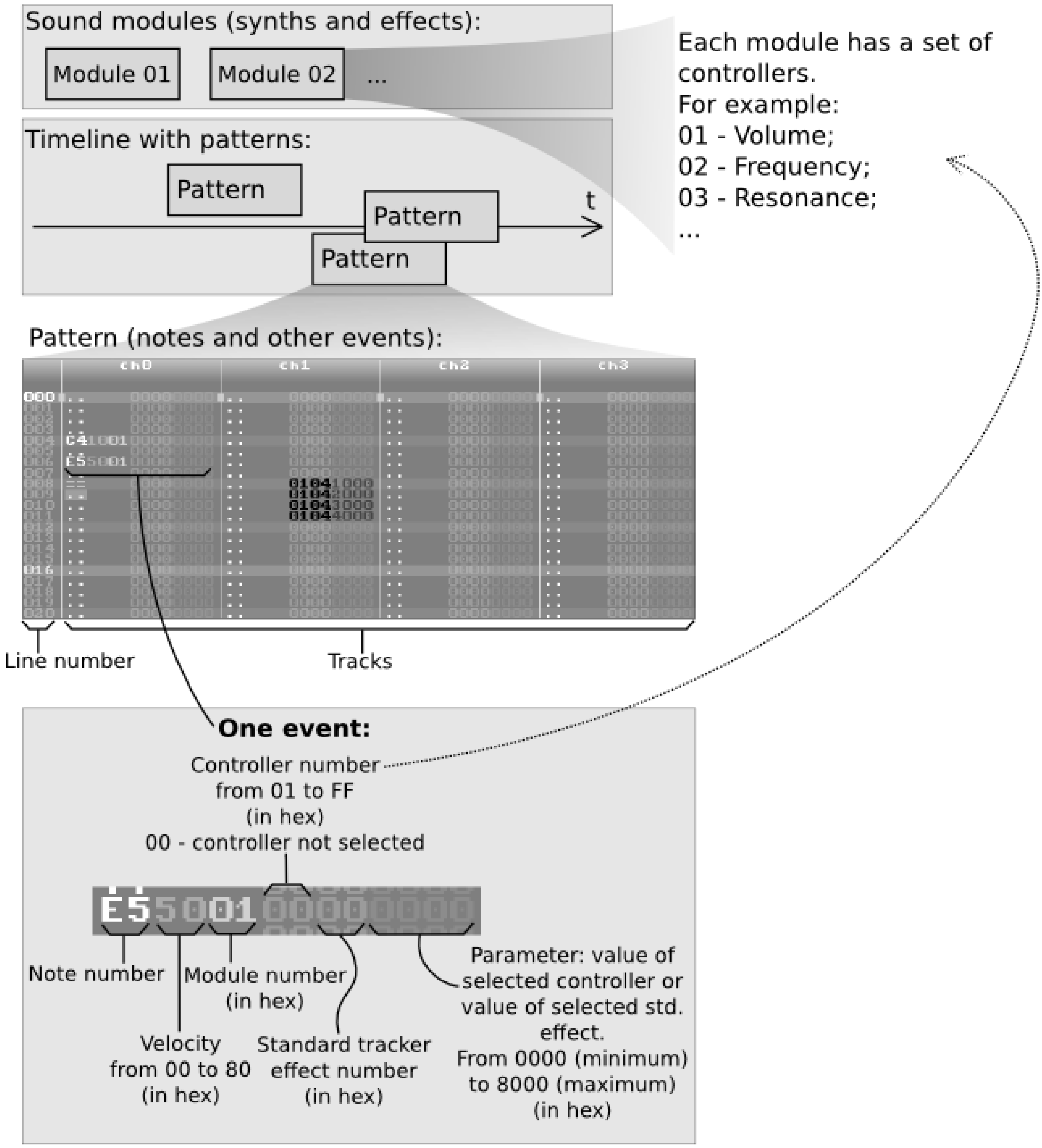
Project

The SunVox project consists of the following parts:

- **modules** generates the sound;
- **patterns** controls the modules (sends them notes and other commands);
- **timeline** determines the order of the patterns.

All the project parts are stored in the single *.sunvox file.The sound of the project should be the same (or close to the original) regardless of the device and system.

SunVox Song Structure:


Sound modules (synths and effects):
Module 01
Module 02
...
Each module has a set of controllers.
For example:
01 - Volume;
02 - Frequency;
03 - Resonance;
...
Timeline with patterns:
Pattern
Pattern
Pattern
t
Pattern (notes and other events):
ch0
ch1
ch2
ch3
Line number
Tracks
One event:
Controller number from 01 to FF (in hex)
00 - controller not selected
E5 5 0 01 00 00 0000
Note number
Module number (in hex)
Velocity from 00 to 80 (in hex)
Standard tracker effect number (in hex)
Parameter: value of selected controller or value of selected std. effect.
From 0000 (minimum) to 8000 (maximum) (in hex)

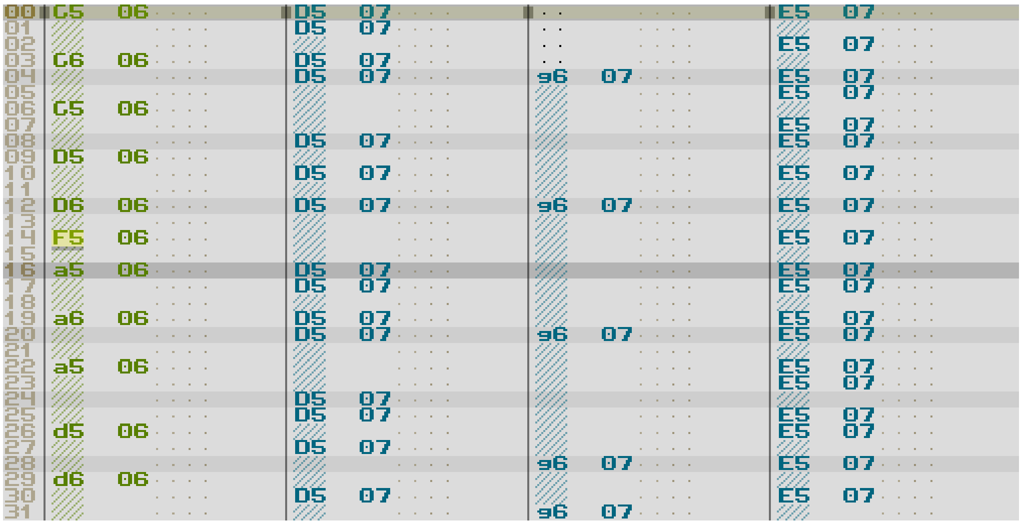
A few words about hex numbers. Memory locations and software in today's computers use the hexadecimal **(hex)** number system which comprises sixteen distinct symbols, digits 0 - 9 and letters A, B, C, D, E, F (or a - f) where the letters represent the numbers 10 - 15. For more detail on the hexadecimal system please visit: https://en.wikipedia.org/wiki/Hexadecimal

Examples of hexadecimal numbers and its decimal **(dec)** equivalents:

- 00 = 0;
- 01 = 1;
- ...
- 09 = 9;
- 0A = 10;
- 0B = 11;
- 0C = 12;
- 0D = 13;
- 0E = 14;
- 0F = 15;

- 10 = 16;
- 11 = 17;
- ...
- FF = 255

Pattern



```
00 C5  06 . . . .  D5  07 . . . .  . .        . . . .  E5  07 . . . .
01                 D5  07 . . . .  . .        . . . .              . . . .
02         . . . .            . . . .  . .        . . . .  E5  07 . . . .
03 C6  06 . . . .  D5  07 . . . .  . .        . . . .              . . . .
04         . . . .  D5  07 . . . .  g6  07 . . . .  E5  07 . . . .
05         . . . .            . . . .             . . . .  E5  07 . . . .
06 C5  06 . . . .            . . . .             . . . .              . . . .
07         . . . .            . . . .             . . . .  E5  07 . . . .
08         . . . .  D5  07 . . . .             . . . .  E5  07 . . . .
09 D5  06 . . . .            . . . .             . . . .              . . . .
10         . . . .  D5  07 . . . .             . . . .  E5  07 . . . .
11         . . . .            . . . .             . . . .              . . . .
12 D6  06 . . . .  D5  07 . . . .  g6  07 . . . .  E5  07 . . . .
13         . . . .            . . . .             . . . .              . . . .
14 F5  06 . . . .            . . . .             . . . .  E5  07 . . . .
15         . . . .            . . . .             . . . .              . . . .
16 a5  06 . . . .  D5  07 . . . .             . . . .  E5  07 . . . .
17         . . . .  D5  07 . . . .             . . . .  E5  07 . . . .
18         . . . .            . . . .             . . . .              . . . .
19 a6  06 . . . .  D5  07 . . . .             . . . .  E5  07 . . . .
20         . . . .  D5  07 . . . .  g6  07 . . . .  E5  07 . . . .
21         . . . .            . . . .             . . . .              . . . .
22 a5  06 . . . .            . . . .             . . . .  E5  07 . . . .
23         . . . .            . . . .             . . . .  E5  07 . . . .
24         . . . .  D5  07 . . . .             . . . .              . . . .
25         . . . .  D5  07 . . . .             . . . .  E5  07 . . . .
26 d5  06 . . . .            . . . .             . . . .  E5  07 . . . .
27         . . . .  D5  07 . . . .             . . . .              . . . .
28         . . . .            . . . .  g6  07 . . . .  E5  07 . . . .
29 d6  06 . . . .            . . . .             . . . .              . . . .
30         . . . .  D5  07 . . . .             . . . .  E5  07 . . . .
31         . . . .            . . . .  g6  07 . . . .              . . . .
```

Pattern is a group of simultaneously played tracks that represents some section of the SunVox project (score). It is divided into horizontal lines. Playing direction is top-down.

Any pattern in the SunVox looks like this:

line	track 0	track 1	track 2	...
000	event	event	event	...
001	event	event	event	...
xxx	...	...	...	...

event - some music event (something like a MIDI command).

Examples of events:

event of turning note C5 on:	C5 80 01 0000 0000
event of turning note A5 on:	A5 80 01 0000 0000
event of turning note off:	==

As you can see, an events contain several text/numeric columns (fields). **All numbers in the pattern are represented in HEXADECIMAL format**.

Event format: **NN VV MM CCEE XXYY**

Lets see detailed description of each column in the event:

- NN - some special command code or the note:
  - SP - Set Pitch XXYY (shortcut **K**), where 0000 - highest possible pitch, 7800 - lowest pitch (note C0); one semitone = 100 (256_{dec});
    formulas:
    from Pitch XXYY to Frequency: frequency (in Hz) = pow(2, (30720_{dec} - XXYY) / 3072_{dec}) * 16.3339;
    from Frequency to Pitch XXYY: XXYY = 30720_{dec} - log2(frequency / 16.3339) * 3072_{dec};

  - << - Previous Track (shortcut **SHIFT + K**); use it to apply effects to previous track; but the following secondary effects are not allowed: 2x, 40..5F, 1D;
  - == - Note OFF (shortcut **CAPSLOCK or '~'**);
  - C0 - C-0 (first note of the lowest octave);
  - c0 - C#0;
  - D0 - D-0;
  - d0 - D#0;
  - E0 - E-0;
  - F0 - F-0;
  - f0 - F#0;
  - G0 - G-0;
  - g0 - G#0;
  - A0 - A-0;
  - a0 - A#0;
  - B0 - B-0;
  - C1 - C-1 (first note of the octave 1);
  - ...
  - B9 - B-9 (last note of the highest octave);
- VV - velocity (00..80);
- MM - module number (00,01,02,03,04,05,06,07,08,09, ... FFFF);
- CC - the controller number (01,02,03, ... 7F) or the MIDI controller number (80,81,82, ... FF);
- EE - effect that will be applied to the last note activated on the current track; some effects (marked in bold) can use the last nonzero parameter when XXYY = 0:
  - **01** - slide up (portamento up; sliding speed = XXYY);
  - **02** - slide down (portamento down; sliding speed = XXYY);
  - **03** - slide to note (sliding speed = XXYY);
  - **04** - vibrato; (XX - frequency; YY - amplitude);
  - 05 - pitch bend up: pitch = original pitch (initialized when a note is activated) + XXYY; (one semitone = 100 (256_{dec}));
  - 06 - pitch bend down: pitch = original pitch (initialized when a note is activated) - XXYY; (one semitone = 100 (256_{dec}));
  - 07 - set phase (or sample offset) XXYY in percents (from 0000 (0%) to 8000 (100%));
  - 08 - arpeggio (XX - second note increment; YY - third note increment);
  - 09 - set phase (or sample offset) XXYY in samples*256_{dec};
  - **0A** - slide velocity up/down (XX - up speed; YY - down speed);
  - 0F - set playing speed (XXYY: 0001..001F - number of ticks per line; 0020..00FF - BPM) or time grids (XXYY: F001..F020 and F100..F120);
  - **11** - fineslide up;
  - **12** - fineslide down;
  - 13 - set Bypass/Solo/Mute (BSM = XYY) flags; example of mute: 13 0001; example of solo: 13 0010;
  - 14 - reset Bypass/Solo/Mute (BSM = XYY) flags; example of bypass reset: 14 0100;
  - 15 - change the relative note and finetune of the module; XX - relative note (00 - ignore; 01 = -127_{dec}; 80 = 0; FF = 127_{dec}); YY - finetune (00 - ignore; 01 = one semitone lower; 80 = 0; FF = one semitone higher);
  - 19 - retrigger note after XXYY ticks during the line;
  - 1C - cut note after XXYY ticks in the current line;
  - 1D - delays the start of note until tick XXYY in the current line;
  - 1F - set BPM;
  - 20 - note probability;
  - 21 - note probability with random velocity;
  - 22 - set controller value to the random number from 0000 to XXYY;
  - 23 - set controller value to the random number with range from XX (00..FF) to YY (00..FF);
  - 24 - take a note from line XXYY on the same track;
  - 25 - take a random note from the range of lines XX..YY (inclusive) on the same track;
  - 26 - take a note from track XXYY on the same line;
  - 27 - take a random note from the range of tracks XX..YY (inclusive) on the same line;
  - 28 - take a note from line XXYY on track 0;
  - 29 - take a random note from the range of lines XX..YY (inclusive) on track 0;
  - 30 - stop playing the song;
  - 31 - jump to line XXYY (address) right after the end of the current line;
  - 32 - set jump address mode YY: 0 - absolute address, relative to the start of the timeline (default); 1 - (pattern beginning + address); 2 - (pattern beginning - address); 3 - (next line + address); 4 - (next

    line - address);
  - destructive effects (irreversibly change the contents of the pattern):
    - 38 - delete an event on track XX with a probability of YY (00..FF(100%)); see simple_examples/pat effect 38 self destructive.sunvox
    - 39 - cyclic shift of track XX down by YY lines;
    - 3A - generate a new iteration of YY-line polyrhythm on track XX; see simple_examples/pat effect 3A polyrhythm.sunvox;
    - 3B - copy track XX to a pattern named YY;
  - 40...5F - delay an event for selected fraction of the line (from 40 (0%) to 5F (96.875%));
- XXYY - parameter of a selected effect/controller (0000 - min; 8000 - max).

Velocity can be changed dynamically.
Example:

```
NN VV MM CCEE XXYY
C3 10 03 0000 0000
-- 20 -- 0000 0000
-- 30 -- 0000 0000
-- 40 -- 0000 0000
-- 50 -- 0000 0000
-- 60 -- 0000 0000
```

In this example velocity changed from 10 to 60.

Example of changing cutoff frequency (global controller) of filter from min to max value:

```
NN VV MM CCEE XXYY
-- -- 01 0200 0000
-- -- 01 0200 1000
-- -- 01 0200 2000
-- -- 01 0200 3000
-- -- 01 0200 4000
-- -- 01 0200 5000
-- -- 01 0200 6000
-- -- 01 0200 7000
-- -- 01 0200 8000
```

The "Cutoff frequency" controller has a number 2 in the filter.

Some modules have the local controllers.
A local controller can be changed for a specific note on a specific track, regardless of other active notes in the same module.
In other words, the local controller only affects the specific polyphony channel of the module.

Example: (changing local controller 2 in the module 1)

```
NN VV MM CCEE XXYY
C3 80 01 0200 0000  //Set local controller 2 to value 0000. Separate sound begins playing here
-- -- -- 0200 0001  //Set local controller 2 to value 0001
-- -- -- 0200 0002  //Set local controller 2 to value 0002
-- -- -- 0200 0003  //Set local controller 2 to value 0003
-- -- -- 0200 0002  //...
-- -- -- 0200 0001
```

Pattern effects

01 - Slide up (Portamento Up). This effect will slide up the frequency of the synth being played on the track by XXYY (one semitone = 40) tones for every tick. If a slide rate is not specified (XXYY is zero) then the last slide rate used on the track is used again.

02 - Slide down (Portamento Down). This effect will slide down the frequency of the synth being played on the track by XXYY (one semitone = 40) tones for every tick. If a slide rate is not specified (XXYY is zero) then the last slide rate used on the track is used again.

03 - Slide to note. This effect will slide a note being played on a track to a specified note NN. The parameter XXYY (one semitone = 40) will states the speed at which a slide will occur. If a slide rate is not specified (XXYY is zero) then the last slide rate used on the track is used again. If NN is empty, the last NN (specified early for effect 03 in the same track) will be used.

04 - Vibrato with parameters XXYY, where the XX - frequency (examples of values: 20 - one period per line; 40 - two periods per line), YY - amplitude (one semitone = 40).

05 - Pitch bend up. Final pitch = original pitch (initialized when a note is activated) + XXYY. One semitone = 100 (256_{dec}).

06 - Pitch bend down. Final pitch = original pitch (initialized when a note is activated) - XXYY. One semitone = 100 (256_{dec}).

08 - Arpeggio. This effect is normally used to simulate chords. It can also be used to produce a heavy vibrato.
Example:
Note = C3; XXYY = 0407 (second note increment = 4; third note increment = 7)
this is an attempt to produce a C-major chord. At the beginning of a line the C3 note is played, then 1/3 into the line the sound is retriggered at E-3, at 2/3 is retriggered at G3, and at the beginning of the next line (if there are no new notes to be played on the track), it is retriggered at C3 again.
Video: Arpeggio Standard Effect

09 - Set sample offset. This functionality enables you to start a sample from a specified position rather than the normal beginning position. Multiply value XXYY * 256 to get the position in samples from the beginning of the sample where playback should start. If no sample is specified, the sample currently playing is retriggered to the offset specified.
Example: play instrument 2 at note C3, and XXYY = 0023. This would make playback of the sample start at offset 0023 * 0100 = 2300. This offset provides quite a wide range from where you can start playing the sample.

07 - Set sample offset (by percentage). XXYY from 0000 (sample start) to 8000 (sample end). Video: Sampler Retriggering / Scaling

0A - Slide velocity up/down. This effect will change the velocity of the synth being played on a track. The value XXYY the speed of the velocity change. If XX is nonzero the velocity is increased, and if YY is nonzero the velocity is decreased.
Example: set XXYY to 0003. This means that at the beginning of the line, the current velocity of the synth is decreased by 3 (because YY = 3). The velocity is decreased by 3 again for every tick on this line.

0F - Set playing speed. This effect changes the TPL (number of ticks per line). Valid values are 1 - 1F (1 - 31 in decimal). Values above 1F represent a modified BPM (beats per minute). A speed value of 2A (42 in decimal), equals 42 beats per minute.

11 - Fineslide up. This effect functions just like effect 1, except that the frequency of the note is only modified once (at the beginning of a line).

12 - Fineslide down. This effect functions just like effect 2, except that the frequency of the note is only modified once (at the beginning of a line).

13 - Set Bypass/Solo/Mute flags. BSM = XYY; Example of mute: 13 0001; example of solo: 13 0010.

14 - Reset Bypass/Solo/Mute flags. BSM = XYY. Example of bypass reset: 14 0100.

15 - Change the relative note XX and finetune YY of the module.

Relative note XX:

- 00: ignore;
- 01: -127_{dec};
- 02: -126_{dec};
- ...
- 80: 0 (no note offset);
- 81: +1;
- 82: +2;
- ...
- FF: $+127_{dec}$.

Finetune YY:

- 00: ignore;
- 01: one semitone lower;

- ...
- 80: 0 (no finetune offset);
- ...
- FF: one semitone higher.

19 - Retrigger. Effect 19 allows you to retrigger a note after XXYY ticks during the line. For example, let's specify note C3 and set XXYY = 2 when the speed (TPL) is currently 6. This starts the specified note at the beginning of the line, and after two ticks it is restarted. This continues until the beginning of the next line.

1C - Cut. This effect turns off the note after XXYY ticks in the current line.

1D - Delay. This effect delays the start of a note until tick XXYY in the current line. For example, if note C2 is played, with effect 1D and argument XXYY = 3 when the speed is 6. The note C2 will be triggered at the 3rd tick after the start of the line. Purpose: to delay the start of a sample for a VERY short amount of time.

1F - Set BPM to XXYY.

20 - Note probability. Set the probability XXYY (from 0 to 8000) of a note being triggered. This effect can be used for original unpredictable rhythms.

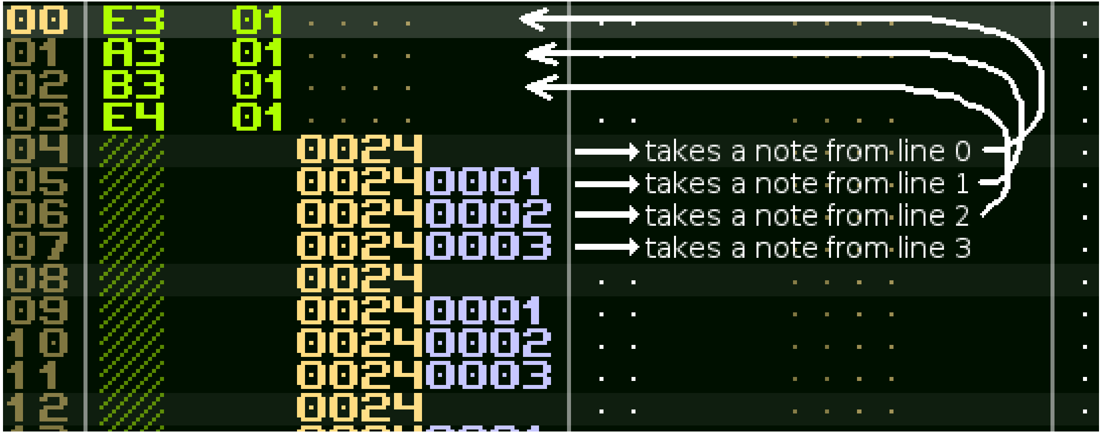
21 - Note probability with random velocity. Same as effect 20 but with random velocity.

22 - Write a random value 0...XXYY to the controller CC.

23 - Write a random value XX (00..FF) ... YY (00..FF) to the controller CC.

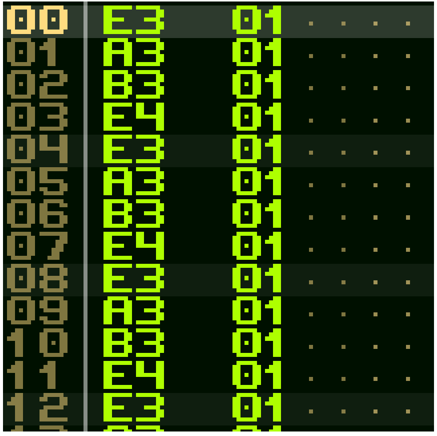
24 - Note from line XXYY. Take a note from line XXYY on the same track. If the note to the left of the effect is specified, it will be used as the transposition value (relative to C5: c5 - one semitone higher, C4 - one octave lower, etc.). If the module number is missing, then the module number from line XXYY will be used.

Example:



```
00  E3   01
01  A3   01
02  B3   01
03  E4   01
04            0024
05            00240001
06            00240002
07            00240003
08            0024
09            00240001
10            00240002
11            00240003
12            0024
```

There are four base notes (on top of the track).
The effect 24 helps you to repeat these notes.
Then you can change one of these notes, and the changes will immediately apply to the entire pattern. So you don't need to press DUPLICATE after each change.
Moreover, you can transpose these notes and change the order in XXYY column.
So the above picture is equivalent to this:



```
00  E3   01
01  A3   01
02  B3   01
03  E4   01
04  E3   01
05  A3   01
06  B3   01
07  E4   01
08  E3   01
09  A3   01
10  B3   01
11  E4   01
12  E3   01
```

25 - Random note from line XX...YY. Take a random note from the range of lines XX..YY (inclusive) on the same track.

26 - Note from track XXYY. Take a note from track XXYY on the same line.

27 - Random note from track XX...YY. Take a random note from the range of tracks XX..YY (inclusive) on the same line.

28 - Note from line XXYY on track 0. Take a note from line XXYY on track 0.

29 - Random note from line XX...YY on track 0. Take a random note from the range of lines XX...YY (inclusive) on track 0. See example "pat effect 29 random note from trk0".

30 - Stop playing the song.

31 - Jump to line XXYY (address) right after the end of the current line.

32 - Set jump address mode YY:

- 0: jump to absolute address, relative to the start of the timeline (default);
- 1: jump to (pattern beginning + address);
- 2: jump to (pattern beginning - address);
- 3: jump to (next line + address);
- 4: jump to (next line - address).

38 - Delete an event on track XX with a probability of YY (00..FF(100%)). See simple_examples/pat effect 38 self destructive.sunvox.

39 - Cyclic shift of track XX down by YY lines.

3A - Generate a new iteration of YY-line polyrhythm on track XX. See simple_examples/pat effect 3A polyrhythm.sunvox.

3B - Copy track XX to a pattern named YY.

40...5F - delay an event for selected fraction of the line (from 0% (40) to 96.875% (5F)).

Video: Standard Effects

MIDI Out

You can use controllers with number 80+X to send MIDI Control Change command. Where the X is MIDI Controller Number.

Example:

```
NN VV MM CCEE XXYY
-- -- 01 8100 0000 - set Modulation (MSB and LSB) to the min value
-- -- 01 8100 8000 - set Modulation (MSB and LSB) to the max value
MSB - Most Significant Byte;
LSB - Least Significant Byte.
```

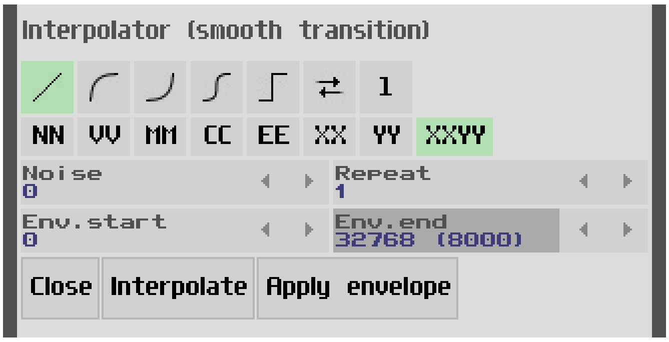
You can get MIDI Controller Number for CC column from the following table.

MIDI controller number (dec)	SunVox controller number (MIDI contr. + 80 (hex)); Use these values in the CC column of SunVox pattern	Controller name
0	80	Bank Select (MIDI Controller 32 more commonly used)
1	81	Modulation Wheel
2	82	Breath Contoller
3	83	Undefined

4	84	Foot Controller
5	85	Portamento Time
6	86	Data Entry MSB
7	87	Main Volume
8	88	Balance
9	89	Undefined
10	8A	Pan
11	8B	Expression
12	8C	Effect Control 1
13	8D	Effect Control 2
14-15	8E-8F	Undefined
16-19	90-93	General Purpose Controllers
20-31	94-9F	Undefined
32-63	A0-BF	LSB for MIDI Controllers 0 - 31
64	C0	Damper Pedal (Sustain)
65	C1	Portamento
66	C2	Sostenuto
67	C3	Soft Pedal
68	C4	Legato Footswitch
69	C5	Hold 2
70	C6	Sound Controller 1 (default: Sound Variation)
71	C7	Sound Controller 2 (default: Timbre/Harmonic Content)
72	C8	Sound Controller 3 (default: Release Time)
73	C9	Sound Controller 4 (default: Attack Time)
74	CA	Sound Controller 5 (default: Brightness)
75-79	CB-CF	Sound Controller 6-10 (no defaults)
80-83	D0-D3	General Purpose Controllers

84	D4	Portamento Control
85-90	D5-DA	Undefined
91	DB	Effects 1 Depth (previously External Effects Depth)
92	DC	Effects 2 Depth (previously Tremolo Depth)
93	DD	Effects 3 Depth (previously Chorus Depth)
94	DE	Effects 4 Depth (previously Detune Depth)
95	DF	Effects 5 Depth (previously Phaser Depth)
96	E0	Data Increment
97	E1	Data Decrement
98	E2	Non-Registered Parameter Number LSB
99	E3	Non-Registered Parameter Number LSB
100	E4	Registered Parameter Number LSB
101	E5	Registered Parameter Number MSB
102-120	E6-F8	Undefined
121	F9	Reset All Controllers
122	FA	Local Control
123	FB	All Notes Off
124	FC	Omni Off
125	FD	Omni On
126	FE	Mono On (Poly Off)
127	FF	Poly On (Mono Off)

Interpolator



Noise 0

Repeat 1

Env.start 0

Env.end 32768 (8000)

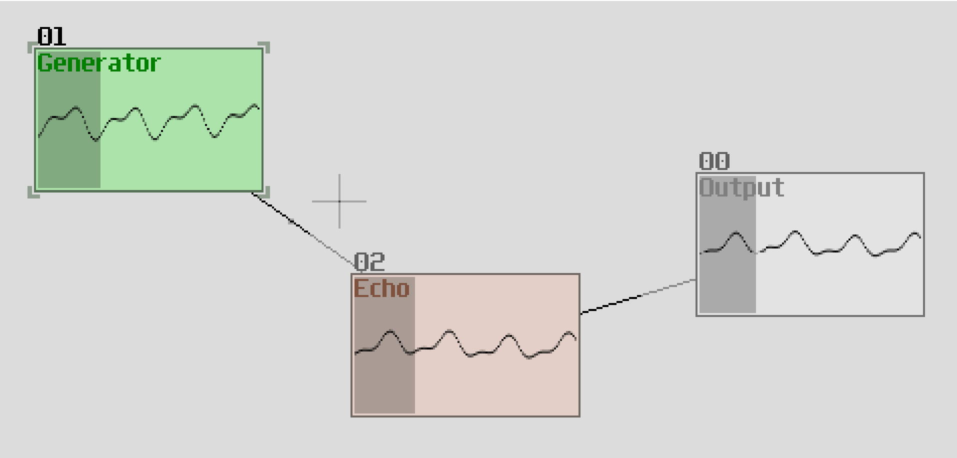
Interpolator is a tool that provides the ability to make a smooth transition between the values of the selected area of the pattern, with various parameters (like the curve type, noise, etc.).

Description of the buttons

- First row: type of the interpolation curve, ping-pong mode switch, interpolation step.
- Second row: column selection (use SHIFT key to select multiple columns).
- Third row: amount of the noise (random values) and the number of repeats.
- Fourth row: envelope begin/end values (are only relevant for the **Apply envelope** button).
- Fifth row:
  - **Interpolate**: New value = Smooth transition between the first and the last line of the selection. Video: Interpolation
  - **Apply envelope**: New value = Old value * Smooth transition between the**Env.Start** and the **Env.end** parameters. Video: How to Apply the Envelope

If you want the smooth transition between the first and the last line of your selection - use**Interpolate**.But if you want to change the existing values (fade them out for example) - use **Apply envelope**.

Module



Module is a basic element of the SunVox. There are several types of modules:

- synthesizer (generator) - can produce some sound;
- effect (filter) - can change the incoming sound;
- some other modules that can generate/change not only the sound but the music events (notes, parameters, etc.) too.

Lines between the modules define the direction of signal transmission: more intensive part of line must go**from the source**; less visible part must go **to the destination (reciever)**.

There is internal set of basic SunVox modules. Click twice on the empty area between the modules - and you will see the list of internal modules. You also can load some modules from the external storage medium by clicking the LOAD button under the list of internal modules. You can find several folders with additional modules (effects, instruments) in the SunVox directory.

There are three ways to connect one module to another:

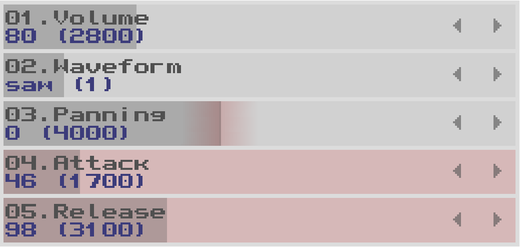
- Select the first module (source). Click on the button. Select the second module (destination).

- Select the first module (source). Drag it to destination with SHIFT key pressed.
- For multitouch devices: first touch on source, second touch on destination. Seethis simple video-tutorial.

Reconnection removes the previous connection.

Controllers

Every module has a set of parameters (**controllers**):



01.Volume
80 (2800)

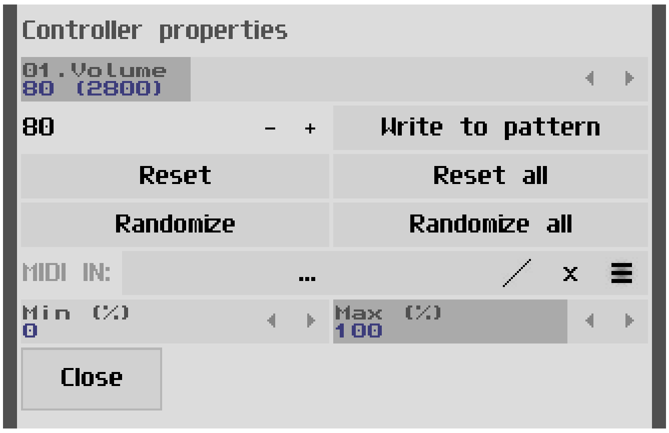
02.Waveform
saw (1)

03.Panning
0 (4000)

04.Attack
46 (1700)

05.Release
98 (3100)

Click twice (or right click) on the controller to open the**Controller Properties** window, where you can do the following operations: change, reset, randomize, write to pattern, assign MIDI IN:



01.Volume
80 (2800)

80

Min (%)
0

Max (%)
100

Controller automation

Automation of the module controllers - automatic change of controller values during playback. Sources of automation: special modules (MultiCtl, Pitch2Ctl, Sound2Ctl, Velocity2Ctl) or the patterns. In the latter case, it does not matter which pattern will send commands to the controller: in SunVox there is no strict binding of certain patterns to the modules and their parameters.
If you press SHIFT, and then on the controller, then its value will be written in a pattern (at the current cursor position).
If you need to write more complex automation, then there are three additional ways:

1. you can manually control the parameters as you want, and all these changes will be recorded in a special pattern in real time; to do this, simply press REC button;
2. in the pattern, you can specify the initial and final value of the controller (for example, at the beginning and at the end of the pattern), then select this region and use the Interpolator (in the pattern menu) to make a smooth transition;
3. fill in the columns MM (module number) and CC (controller number); turn on the "XXYY levels drawing" mode in the pattern menu; then you can draw a freehand automation curve directly in the pattern.

If automation is recorded in a pattern, then the changes do not occur smoothly, but in steps. Step length = 1 tick (after live recording), or 1 line (in other cases). To make changes smoother, automation commands can be sent to the MultiCtl module (in which the Response parameter is different from 1000), which will smooth the changes and send commands to other modules connected to it.

Video: Automation in SunVox: six ways

Modes

Some modules have a "Mode" parameter with the following possible values:

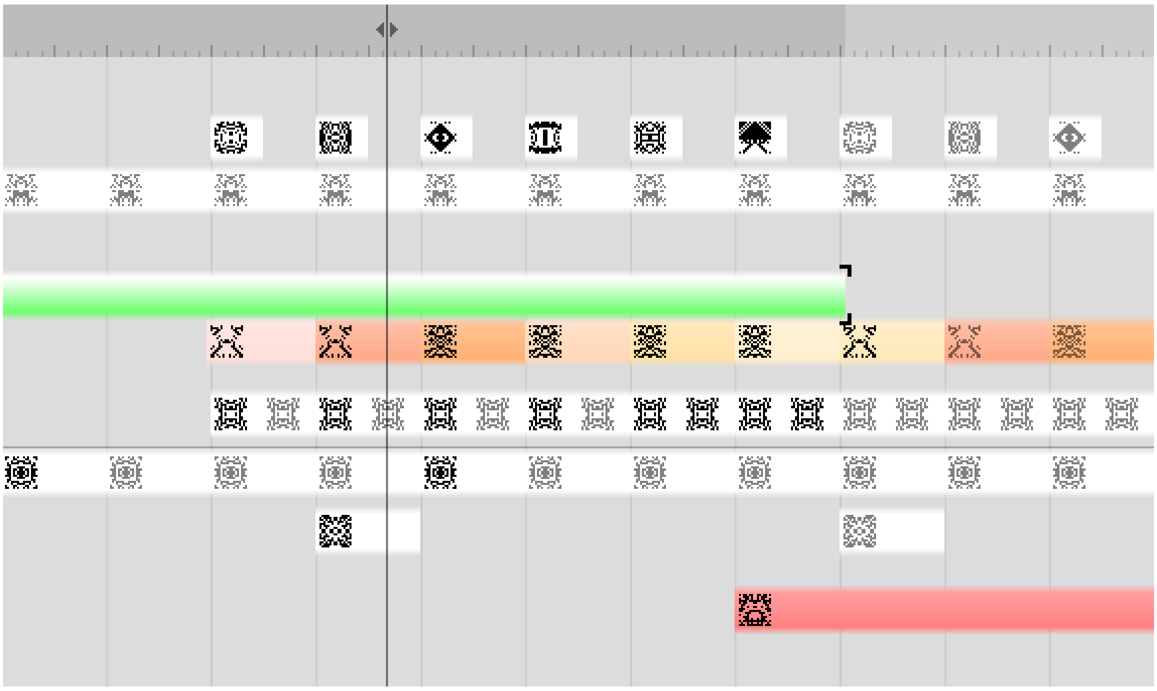
- HQ - High Quality (stereo);
- HQmono - High Quality (mono);

- LQ - Low Quality (stereo) for devices with slow CPU;
- LQmono - Low Quality (mono) for devices with slow CPU;
- HQspline - High Quality (stereo) with spline interpolation.

If the CPU of your device is too slow for some SunVox project (app stutters and hangs), you can use the following tips:

- use monophonic modes (Mono, HQmono or LQmono) for the modules;
- use low quality modes (LQ or LQmono) for the modules;
- use reduced polyphony for the generators;
- don't use the Reverb, or use it in LQ mode (Mode = LQ; All-pass filter = OFF);
- use the Filter instead of the Filter Pro;
- Sampler: turn off the interpolation (Sample interpolation = OFF; Volume interpolation = OFF);
- Analog Generator: don't enable the filter when the polyphony is greater than 2; instead, use the single Filter module connected to the output of the generator;
- Glide, Sound2Ctl, MultiCtl: don't use the Sample rate greater than 150 Hz.

Timeline



Timeline defines the layout of the SunVox project. It describes how to play the patterns. Patterns can be repeated across multiple orders to save tracking time and file space.

When you select a pattern, you will see its contents (tracks, lines, events) in the Pattern Editor. In the single pattern play mode, when you select another pattern, it will play after the end of the previous selected pattern.

Playback speed is defined by the following parameters:

- TPL (Speed) - Ticks Per Line. Tick is the smallest time interval in SunVox. Tick length in milliseconds = 60000 / BPM / 24.
- BPM - Beats Per Minute. Beat = 24 ticks. Number of lines per beat = 24 / TPL. So if TPL=6, Beat size is 4 lines. Or if TPL=3, Beat size is 8 lines.

Internal modules

Synths

Analog Generator

Generator with 32 double alias-free oscillators, 12/24dB filters, envelopes, and smooth change of parameters.
The sound quality of this module is better at a sample rate of 44100Hz.

Controllers:

- Volume;
- Waveform:
  - 0 - triangle;
  - 1 - saw;
  - 2 - square;
  - 3 - noise (sampler);
  - 4 - hand drawn waveform;
  - 5 - sine;
  - 6 - half-sine;
  - 7 - abs-sine;
  - 8 - hand drawn waveform with spline interpolation;
  - 9 - noise (sampler) with spline interpolation;
  - A - white noise;
  - B - pink noise;
  - C - red noise;
  - D - blue noise;
  - E - violet noise;
  - F - grey noise;
  - 10 - hand drawn harmonic distribution; remember, if you turn on all 32 harmonics, then the module's performance will slow down 32 times compared to a pure sine generator; reduced polyphony and LQmono mode will slightly reduce the CPU load;
- Panning;
- Attack - attack time (volume envelope);
- Release - release time (volume envelope);
- Sustain (off/on);
- Exponential envelope (off/on);
- Duty cycle (only for square waveform);
- Freq2 - frequency deviation of the additional generator (off in the zero position); semitone = 64;
- Filter:
  - off;
  - LP (Low-pass) 12dB;
  - HP (High-pass) 12dB;
  - BP (Band-pass) 12dB;
  - BR (Band-rejection) 12dB;
  - LP (Low-pass) 24dB;
  - HP (High-pass) 24dB;
  - BP (Band-pass) 24dB;
  - BR (Band-rejection) 24dB;
- Filter frequency (Hz);
- Filter resonance;
- Filter exponential frequency (off/on);
- Filter attack - filter envelope attack time;
- Filter release - filter envelope release time;
- Filter envelope: off, sustainOFF (without sustain), sustainON (with sustain);
- Polyphony - number of notes that the module can play simultaneously;
- Mode: HQ, HQmono, LQ, LQmono;
- Noise - additional amount of white noise.

Local controllers: Volume, Waveform, Panning, Duty cycle, Filter type, Filter freq, Filter resonance.

Options:

- 7F - Volume envelope time scaling per key;
- 7E - Filter envelope time scaling per key;
- 7D - Volume scaling per key;
- 7C - Filter frequency scaling per key;
- 78 - Filter frequency scaling per key (reverse);
- 75 - Filter frequency = note frequency;
- 7B - Velocity-dependent filter frequency;
- 74 - Velocity-dependent filter resonance;
- 7A - Frequency / 2;
- 79 - Smooth frequency change;
- 77 - Retain phase;

- 76 - Random phase;
- 73 - True zero attack/release.

Video: Virtual Analog Generator

DrumSynth

Drum synthesizer with variety of predefined sounds.
The sound quality of this module is better at a sample rate of 44100Hz.

Distribution of sounds:

- notes C, C#, D, D# - for bass drum;
- notes E, F, F# - for hihat;
- notes G, G#, A, A#, B - for snare drum.

Controllers:

- Volume - main synth volume;
- Panning;
- Polyphony - number of notes that the module can play simultaneously;
- Bass volume - bass drum volume;
- Bass power - bass drum power;
- Bass tone - bass drum tone;
- Bass length - bass drum length;
- Hihat volume;
- Hihat length;
- Snare volume - snare drum volume;
- Snare tone - snare drum tone;
- Snare length - snare drum length.

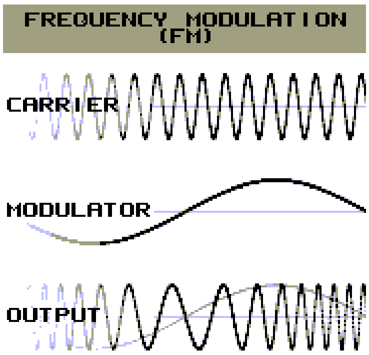
Local controllers: Panning.

Videos:

- DrumSynth
- DrumSynth and Patterns

FM

FM Synthesizer (Frequency Modulation) with ADSR envelope.
C - carrier (first operator);
M - modulator (second operator).



The sound quality of this module is better at a sample rate of 44100Hz.

Controllers:

- C.Volume - carrier volume;
- M.Volume - modulator volume;
- Panning;
- C.Freq mul - carrier frequency multiplier;
- M.Freq mul - modulator frequency multiplier;

- M.Feedback - modulator feedback;
- C.Attack - carrier attack time;
- C.Decay - carrier decay time;
- C.Sustain - carrier sustain level;
- C.Release - carrier release time;
- M.Attack - modulator attack time;
- M.Decay - modulator decay time;
- M.Sustain - modulator sustain level;
- M.Release - modulator release time;
- M.Scaling - how much the modulator volume depends on the pitch;
- Polyphony - number of notes that the module can play simultaneously;
- Mode:
  - 0 - HQ: High Quality; stereo; linear interpolation of envelopes;
  - 1 - HQmono: High Quality; mono; linear interpolation of envelopes;
  - 2 - LQ: Low Quality; stereo; no interpolation;
  - 3 - LQmono: Low Quality; mono; no interpolation;

Videos:

- FM Synth
- FM synth. Another example

Generator

Basic generator of different types of periodic signal waveforms with the volume envelope.

This module can receive the incoming signal and use it for the frequency modulation. To do this, connect any other audio source to the input of the Generator, and that source will affect the frequency of the Generator. And don't forget to set a "Frequency modulation on input" controller to some nonzero value.

Controllers:

- Volume;
- Waveform:
  - 0 - triangle;
  - 1 - saw;
  - 2 - square;
  - 3 - white noise;
  - 4 - hand drawn waveform;
  - 5 - sine;
  - 6 - half-sine;
  - 7 - abs-sine;
  - 8 - pulse-sine with duty cycle.
- Panning;
- Attack - attack time;
- Release - release time;
- Polyphony - number of notes that the module can play simultaneously;
- Mode: 0 - stereo; 1 - mono;
- Sustain: 0 - off; 1 - on;
- Frequency modulation on input: 0 - off ... 256 - use input signal for frequency modulation;
- Duty cycle (only for square waveform).

Local controllers: Type, Pan.

Videos:

- Generator Module
- Drawing your own waveform in Generator
- Drawing your own waveform in Generator #2

Input

Audio input from Microphone/Line-in.

Compatibility: implemented in the iOS and Android versions only. Other systems will be supported in future SunVox updates.

Kicker

Controllers:

- Volume;
- Type - type of waveform: 0 - triangle; 1 - square; 2 - sine;
- Panning;
- Attack;
- Release;
- Boost (additional volume);
- Env. Accel - volume & frequency envelope acceleration;
- Polyphony - number of notes that the module can play simultaneously;
- Anticlick: 0 - OFF; 1 - ON.

Local controllers: Waveform, Panning.

Video: Kicker, Compressor, and Ducking

Vorbis player

OGG Vorbis player.

Sampler

Sampler can play and record audio files.
Supported file formats: WAV (uncompressed), XI, AIFF, RAW. 8 / 16 / 24 / 32 bits. In Linux version all FFmpeg/AVConv formats are supported too.
The Sampler can load JPEG image as a sample: the brightness of the pixels will be converted into a sound wave (pixel scan line by line, left to right, top to bottom).
If the file format is not recognized, the Sampler will load it as raw data: you will be prompted to manually select the data format (number of bits, channels, etc.).

Buttons description:

- Load - load some audio file; this file will be stored in the module;
- Rec - start/stop recording of received audio signal; for example, you can connect Generator to Sampler, and Sampler will record the signal from the Generator;
- Edit - show/hide sample editor.

Controllers:

- Volume;
- Panning;
- Sample interpolation: 0 - OFF; 1 - Linear;
- Volume interpolation: 0 - OFF; 1 - Linear;
- Polyphony - number of notes that the module can play simultaneously;
- Rec threshold - the level at which the Sampler should start recording.

Local controllers: Panning.

Videos:

- Sampler Modules
- How to make the voice with loop in Sampler
- Sampler eats JPEG images
- Sampler Retriggering / Scaling
- How to keep the length on different notes
- Input Module and Sampler Recording
- Sampler Envelopes

SpectraVoice

SpectraVoice synthesizes sound with a complex spectrum.

The spectrum is a graph where the X-axis is the frequency and the Y-axis is the amplitude (loudness). You can place 16 harmonics on this graph, specifying the position, amplitude, shape and width for each.
To hear exactly the specified frequencies (without shifting up or down, depending on the note) - play the note c6.

Controllers:

- Volume;
- Panning;
- Attack (sec/256);
- Release (sec/256);
- Polyphony - number of notes that the module can play simultaneously;
- Mode: HQ, HQmono, LQ, LQmono, HQspline;
- Sustain (off/on);
- Spectrum resolution:
    - 0 - 4096 samples;
    - 1 - 8192 samples;
    - 2 - 16384 samples;
    - 3 - 32768 samples;
    - 4 - 65536 samples;
    - 5 - 131072 samples;
- Harmonic - current harmonic number to edit;
- H.freq - harmonic frequency;
- H.volume - harmonic amplitude;
- H.width - harmonic width;
- H.type - harmonic type:
    - 0 - half sine;
    - 1 - square;
    - 2 - organ1;
    - 3 - organ2;
    - 4 - organ3;
    - 5 - organ4;
    - 6 - full sine;
    - 7 - random;
    - 8 - triangle1;
    - 9 - triangle2;
    - A - overtones1;
    - B - overtones2;
    - C - overtones3;
    - D - overtones4;
    - E - overtones1+;
    - F - overtones2+;
    - 10 - overtones3+;
    - 11 - overtones4+;
    - 12 - metal (random seed is set by H.width);

Local controllers: Panning.

Video: SpectraVoice: four harmonics

Effects

Amplifier

Signal amplifier with various settings.

For example you can split a stereo audio input into the left and right channels with this module:

1. filter the desired channel using the Balance controller: -128 for the left; 128 for the right;
2. move this channel to the center by reducing the Stereo Width to 0.

Controllers:

- Volume;
- Stereo balance;

- DC offset;
- Inverse amplitude: 0 - off; 1 - on;
- Width - stereo width: 0 - mono; 128 - normal; 256 - extended;
- Absolute - enable it to apply "absolute value" filter to the signal;
- Fine volume;
- Gain - high amplification (up to 5000 times).

Compressor

Side chain compressor

Controllers:

- Volume;
- Threshold: 256 - 0 dB;
- Slope (%);
- Attack (ms);
- Release (ms);
- Mode: 0 - peak; 1 - RMS sensing; 2 - peak with zero latency for compression/limiting without delay (modes "peak" and "RMS" operate with a delay of 1 ms);
- Side chain input - side chain input number.

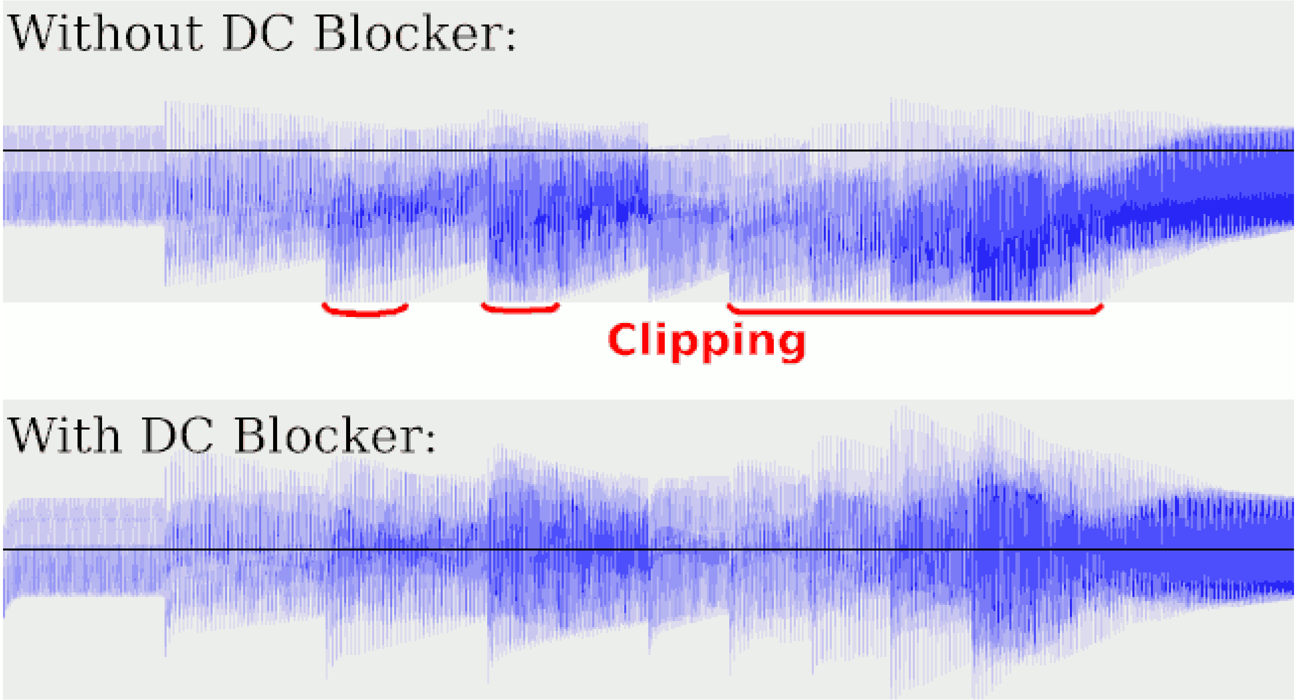
Videos:

- Side Chain Compressor
- Kicker, Compressor, and Ducking

Explanation of compressor settings on the forum

DC Blocker

DC blocking filter.



Controllers:

- Mono: 0 - stereo; 1 - mono.

Delay

This module delays the sound and the incoming events (note, pitch, phase, velocity).Max delay length = 4 seconds.

Controllers:

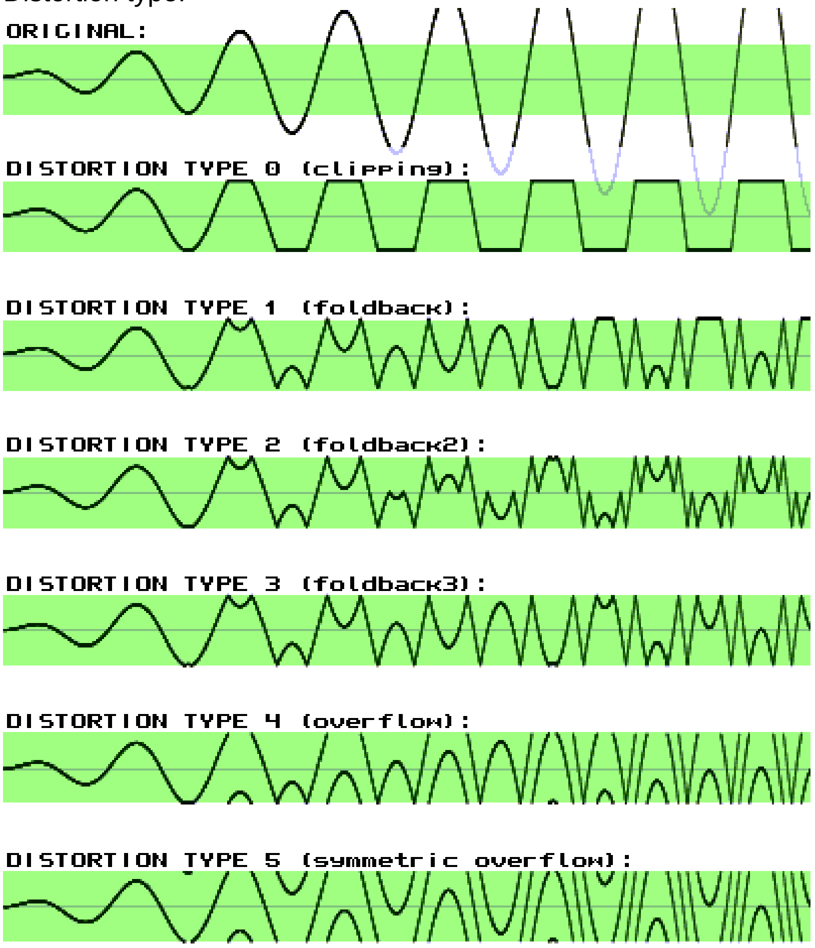
- Dry - amount of original signal;
- Wet - amount of processed signal;
- Delay L - left channel delay length, or event delay length;
- Delay R - right channel delay length;
- Volume L - left channel volume;
- Volume R - right channel volume;
- Mono: 0 - stereo; 1 - mono (use left channel only);
- Inverse amplitude (off/on);
- Delay unit for "Delay L" and "Delay R".

Distortion

This module adds various types of distortion to the sound.

Controllers:

- Volume;
- Distortion type:



- Power - amount of distortion of the type selected above;
- Bit depth - change it to reduce the resolution of the incoming sound; default value 16 means that the bit depth remains unchanged;
- Frequency (Hz) - change it to reduce the sample rate of the incoming sound; default value 44100 means that the sample rate remains unchanged;
- Noise - amount of noise.

Echo

Maximal delay length: 4 seconds.

Controllers:

- Dry - amount of original signal;
- Wet - amount of processed signal;
- Feedback;

- Delay - delay length; maximal delay - one second;
- Stereo: 0 - mono; 1 - stereo;
- Delay units.

EQ

3Band equalizer.

Controllers:

- Low;
- Middle;
- High;
- Mono: 0 - stereo; 1 - mono.

Filter

IIR Filter that can remove some unwanted frequency ranges.

In HQ mode the Filter is double-sampled.In LQ mode the Filter is 2x faster, but with the following consequences:

- frequency range is reduced by half (to 7000 Hz maximum); so if you want to set cut-off frequency 2000 Hz - you should set it to 4000;
- LQ algorithm is very fast (specially for slow devices), but the cost of this speed is the instability in some cases (Filter in LQ mode with high Freq values is unstable on some input signals with a predominant high-frequency component).

Controllers:

- Volume;
- Frequency (Hz);
- Resonance;
- Type:
    - 0 - low-pass;
    - 1 - high-pass;
    - 2 - band-pass;
    - 3 - notch (band-stop);
- Response - rate of change of the frequency, resonance and volume;
- Mode:
    - 0 - double sampled stereo;
    - 1 - double sampled mono;
    - 2 - low quality stereo;
    - 3 - low quality mono;
- Impulse - change it to make the frequency impulse (burst): from the Impulse value ... to the previous frequency value; impulse speed = Response;
- Mix - mix between the input (min) and the output (max) signals;
- LFO frequency;
- LFO amplitude;
- Set LFO phase - initial phase (after the module reset or after receiving the note (Note ON)) or the command to immediately set the phase to a new value;
- Exponential freq;
- Roll-off (dB per octave);
- LFO frequency unit;
- LFO waveform.

Videos:

- Acid Bass
- Filter LFO + MultiSynth
- Changing Filter frequency from the pattern

Filter Pro

High-quality 64-bit IIR Filter that can amplify, pass or attenuate some frequency ranges. This module is slower than the Filter.

Controllers:

- Volume;
- Type:
  - 0 - low-pass;
  - 1 - high-pass;
  - 2 - band-pass;
  - 3 - band-pass with constant peak gain;
  - 4 - notch (band-stop);
  - 5 - all-pass;
  - 6 - peaking (use the "Gain" controller to amplify or attenuate the peak at the specified frequency);
  - 7 - low shelf (use the "Gain" controller to amplify or attenuate the frequencies below the specified frequency);
  - 8 - high shelf (use the "Gain" controller to amplify or attenuate the frequencies above the specified frequency);
- Frequency (Hz);
- Frequency finetune (Hz/1000);
- Frequency scale (%);
- Exponential freq - enable it if you want to use the exponential frequency scale;
- Q - quality factor (resonance);
- Gain - additional parameter for the filter types 6-8;
- Roll-off (dB per octave);
- Response - rate of change of the volume, frequency, Q, gain and mix;
- Mode: 0 - stereo; 1 - mono;
- Mix - mix between the input (min) and the output (max) signals;
- LFO frequency;
- LFO amplitude;
- LFO waveform;
- Set LFO phase - initial phase (after the module reset or after receiving the note (Note ON)) or the command to immediately set the phase to a new value;
- LFO frequency unit.

Flanger

Flanger effect.

Controllers:

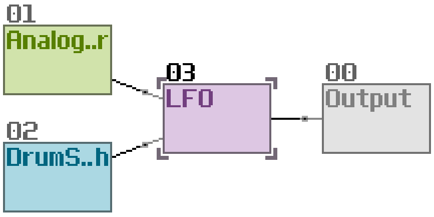
- Dry - amount of original signal;
- Wet - amount of processed signal;
- Feedback;
- Delay - delay length; maximum = 1/64 second;
- Response - rate of change of the delay length;
- LFO frequency;
- LFO amplitude;
- LFO waveform: 0 - half of sine; 1 - sine;
- Set LFO phase - initial phase (after the module reset or after receiving the note (Note ON)) or the command to immediately set the phase to a new value;
- LFO frequency unit.

LFO

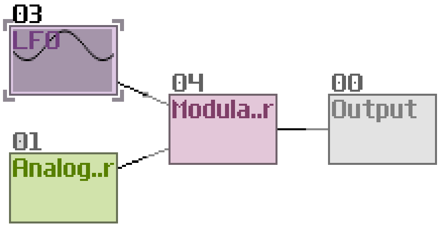
LFO - Low Frequency Oscillator.

Here are three ways to use this module.

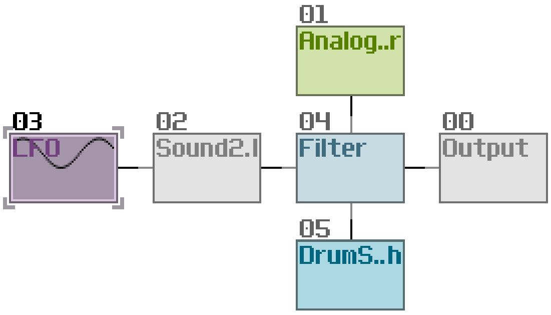
1. Generator=OFF; some signal is sent to the input; at the output we get the same signal, but its volume changes in accordance with the oscillations of the built-in oscillator.



2. Generator=ON; the input is empty; at the output we get the wave of the built-in oscillator; this output signal can be used for something else, for example to modulate the volume or frequency through the Modulator.

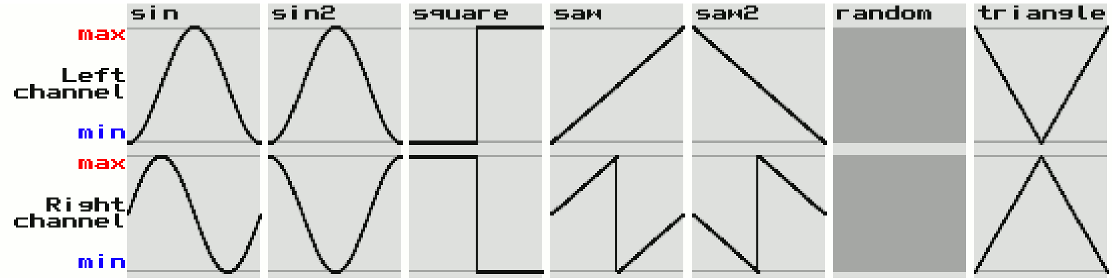


3. Generator=ON; the input is empty; output is connected to Sound2Ctl; thus, the LFO can change the values of the controllers of other modules.



Controllers:

- Volume;
- Type: 0 - amplitude modulation (tremolo); 1 - stereo balance modulation;
- Amplitude;
- Frequency;



- Waveform:
- Set phase - initial phase (after the module reset or after receiving the note (Note ON)) or the command to immediately set the phase to a new value;
- Channels: 0 - stereo; 1 - mono;
- Frequency unit;
- Duty cycle (only for square waveform);
- Generator:
  - 0 (off) - LFO works as a modulator (changes the volume of the incoming signal);
  - 1 (on) - LFO works as a generator, producing oscillation at the output, independent of the input signal;
- Frequency scale (%) - use it to slow or accelerate the oscillations;
- Smooth transitions - disable it to get hard transitions inside the waveform.

Loop

This module repeats a fragment of the incoming sound a specified number of times.To reset the loop: either change the "Repeats" controller, or send some note to this module.

Controllers:

- Volume;
- Delay - repeat length;

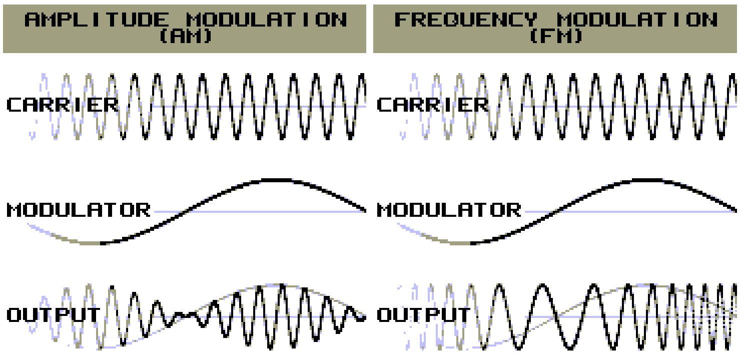
- Stereo: 0 - mono; 1 - stereo;
- Repeats - how many times to repeat a fragment before proceeding to capture the next;
- Mode: 0 - normal; 1 - ping-pong loop playback.

Video: Loop Module

Modulator

Amplitude or Phase modulator. First input = Carrier. Other inputs = Modulators (will be mixed into a single signal).

In phase modulation mode this module is simply a variable delay controlled by the amplitude of the modulating signal.



Controllers:

- Volume;
- Modulation type:
  - 0 - amplitude modulation;
  - 1 - phase modulation; default signal delay = 20ms; delay range: 40ms (mod.amp 1) ... 20ms (mod.amp 0) ... 0ms (mod.amp -1);
  - 2 - absolute phase modulation (only absolute amplitude values will be used); default signal delay = 0ms; delay range: 40ms (mod.amp 1) ... 0ms (mod.amp 0) ... 40ms (mod.amp -1);
- Channels: 0 - stereo; 1 - mono.

Pitch Shifter

Pitch shifter is a module for changing the pitch of any sound in real time.
The signal at the output of the module is always slightly delayed. Delay length is proportional to the "Grain size". The module will work without delay only if the following conditions are met: Pitch = 0; Feedback = 0; "Bypass if pitch=0" = slow/fast transition; the input signal is silence.

Controllers:

- Volume;
- Pitch - how much to lower or raise the pitch (semitone = 10; octave = 120); if 0, then the module will not change the signal; incoming notes (relative to C5) change the value of this controller;
- Pitch scale: if less than 100%, the pitch change is reduced; if more than 100%, the pitch change is increased;
- Feedback - amount of the output signal that will be mixed into the input of the module;
- Grain size: if rhythm prevails in the original signal, use small values; if the original signal is rather monotonic, use higher values; max = 1/4 sec;
- Mode;
- Bypass if pitch=0 (off / slow transition / fast transition) - play the original signal when the module does not change the pitch; it works only with zero feedback;

Video: Pitch Shifter

Reverb

Reverb is a module that simulates the reverberation effect (echo with numerous reflections to make the sound more natural).

Controllers:

- Dry - amount of original signal;

- Wet - amount of processed signal;
- Feedback;
- Damp - signal attenuation through a low-pass filter;
- Width;
- Freeze: 0 - none; 1 - freeze mode;
- Mode;
- All-pass filter (off/on);
- Room size;
- Random seed - random combination of partitions inside the room.

Vocal Filter

This filter was designed to simulate the vocal tract of a human.

Controllers:

- Volume;
- Bandwidth - bandwidth of formants;
- Amp. add - additional value for amplitude of formants;
- Formants - number of voice formants; maximum = 5 (best result);
- Vowel: A...E...I...O...U (minimal value - A; maximal - U);
- Character:
    - 0 - soprano;
    - 1 - alto;
    - 2 - tenor;
    - 3 - bass;
- Mono: 0 - Stereo; 1 - Mono.

Video: How to make a beautiful synthetic choir

Vibrato

Vibrato effect.

Controllers:

- Volume;
- Amplitude;
- Frequency;
- Channels: 0 - stereo; 1 - mono;
- Set phase - initial phase (after the module reset or after receiving the note (Note ON)) or the command to immediately set the phase to a new value;
- Frequency unit;
- Exponential amplitude (off/on).

WaveShaper

In simple terms, WaveShaper allows you to change the shape of the input signal. Default shape is linear (no changes; play the original signal).

In math terms, WaveShaper is the expression y = f(x); where y - output; x - input; f - function with graph which you can see in the WaveShaper interface.The horizontal axis of the graph is Input.The vertical axis of the graph is Output.

Controllers:

- Input volume;
- Mix: minimum - output=input; maximum - output=filtered input;
- Output volume;
- Symmetric;
- Mode;
- DC filter (DC Blocking Filter).

Misc

ADSR

ADSR envelope generator. The module can be started either by notes at the input (reacts to note ON/OFF), or by setting the "State" controller to the "start" value.

Controllers:

- Volume;
- Attack (ms) - the time from the beginning of the envelope (zero amplitude) to the maximum amplitude;
- Decay (ms) - the time from the point with the maximum amplitude to the sustain level;
- Sustain level;
- Release (ms) - the time from the sustain level to the end point of the envelope (zero amplitude);
- Attack curve - attack curve type (linear, smoother, sharper, etc.);
- Decay curve - decay curve type;
- Release curve - release curve type;
- Sustain (off/on) - suspension point (until the Note OFF) on the envelope: 0 (disabled) or 1 (enabled);
- Sustain pedal (off/on);
- State (stop/start);
- On NoteON:
  - do nothing;
  - start on first note;
  - start;
- On NoteOFF:
  - do nothing;
  - stop on last note;
  - stop;
- Mode:
  - generator - the envelope curve appears at the output of the module as an audio signal; useSound2Ctl to convert this signal into control commands (controllers automation);
  - amplitude modulator mono - the module works as a modulator: the amplitudes of the input signal and the envelope are multiplied;
  - amplitude modulator stereo - the same as the previous mode, but in stereo;
- Smooth transitions:
  - off;
  - rastart and volume change - soft (no clicks) restart of the envelope (when the previous envelope is still playing, but a new one is already running), smooth volume transitions (when changing the "Volume" controller);
  - restart (smoother) and volume change - same as the previous mode, but the envelope restart is even smoother;

Ctl2Note

Ctl2Note converts the value of the "Pitch" controller to a note (Note ON/OFF commands at the output).

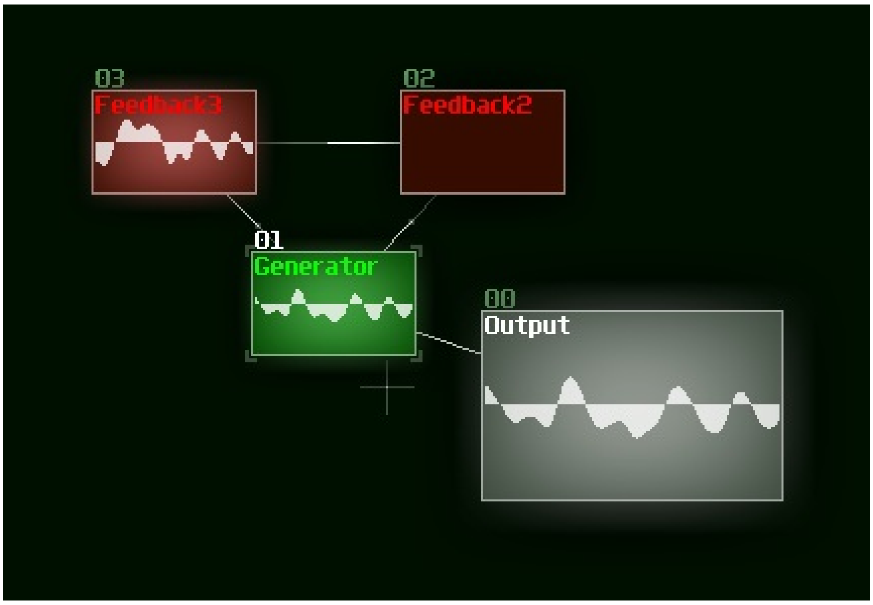
Controllers:

- Pitch (range: 0...32768);
- First note - note number corresponding to the zero value of the pitch;
- Range (number of semitones) - pitch range is divided into this number of semitones;
- Transpose;
- Finetune;
- Velocity;
- State (off/on);
- NoteON:
  - never;
  - on pitch change;
- NoteOFF:
  - never;
  - on min pitch;
  - on max pitch;
- Record notes (off/on): if this option is enabled and recording is in progress (the REC button is active), then all notes from the output of this module will be recorded into a new pattern.

Feedback

Generally the feedback is not allowed in SunVox you can't create an endless loop between the modules. But you can do it by placing two Feedback modules inside of the loop (see screenshot).

Internal Feedback delay = 20 ms.



Glide

Glide is similar to the MultiSynth (which sends the input events to the connected output modules), but it also adds the commands of smooth transition between the notes.

GPIO

With this module you can use the General-Purpose Input/Output (GPIO) pins of the device board. For example, you can send some signals to the LEDs, or receive the ON/OFF (1/0) messages from the buttons.

Requirements: Linux with GPIO Sysfs Interface enabled.

Platform-dependent GPIO numbering (not physical numbering) is used to access GPIO pins. here you can read about the GPIO numbering on Raspberry Pi.

Controllers:

- Out - if enabled, the module will send 0 or 1 to the output pin;
- Out pin - output pin number;
- Out threshold - if the signal level at the module input exceeds this value, the output pin will switch to 1;
- In - if enabled, the module will receive 0 or 1 from the input pin;
- In pin - input pin number;
- In note - if 1 is read from the input pin, a note with the specified number (except zero) will be sent to all receiver modules (connected to the output of GPIO module);
- In amplitude - if 1 is read from the input pin, the specified amplitude (signal level) be set at the output of the module.

Example in simple_examples folder: raspberry_pi_gpio.

Video: Using Raspberry Pi GPIO in SunVox

MetaModule

MetaModule is a full-featured copy of SunVox in a single module. So you can include one SunVox-project into another recursively. Some complex synth or effect (with large amount of the modules) may be hidden in the MetaModule. You also can use it as arpeggiator.

There are two ways to play some generator in the MetaModule:

1. set Input Module controller to play some selected module inside of the MetaModule; polyphony is possible in the this way, because it can be handled by the selected module directly;
2. set Play Patterns controller and play the whole SunVox project (with patterns) inside of the MetaModule;

polyphony is not possible in this way, because we need a separate copy of the MetaModule for each polyphony channel, but this limitation may be removed in future SunVox updates.

Read more...

Video: Five ways to use MetaModule

MultiCtl

With this module you can control multiple controllers at once.

Controllers:

- Value - input value;
- Gain (normal = 256);
- Quantization - max number of output levels;
- OUT offset;
- Response: 1000 - immediate change; 0...1000 - smooth change;
- Sample rate; optimal value = 150Hz, please avoid higher values without the urgent need.

Video: MultiCtl Module

MultiSynth

This module sends the incoming events (notes, pitch change, phase change) to any number of connected modules (receivers). So you can play a large number of synths simultaneously by pressing one note only. The MultiSynth can also modify incoming events according to the parameters described below.

Controllers:

- Transpose (number of semitones up or down);
- Random pitch (one semitone = 256);
- Velocity;
- Finetune (+256 = +1 semitone; -256 = -1 semitone);
- Random phase;
- Random velocity;
- Phase - the initial position of the sample or the initial phase of some Generator;
- Curve2 influence - mix between the linear (default) and the user-drawn curve2 (*output velocity = curve2(input velocity)*).

Options:

- 7F - Use static note C5 - replace incoming notes by C5;
- 7E - Ignore notes with zero velocity;
- 7C - Trigger: Note OFF commands will be ignored; the new note turns off the old note with the same pitch;
- 7B - Generate missed Note OFF cmds: if two notes follow each other on the same track - there are no Note OFF commands between them (one note simply replaces another); sometimes we need to insert these missing commands, to avoid notes hanging; read more;
- 7A - Round note X (X-axis on curve3) to the nearest integer (remove microtones);
- 79 - Round pitch Y (Y-axis on curve3) to the closest;
- 78 - Record notes to the scale (curve3) with automatic copying for 10 octaves; see example "dynamic scale";
- 76 - Output note = Output note - Input note + C5; (difference between output and input pitch, can be used in Pitch Shifter);
- 75 - Output port number = note % number of connected outputs; (% is the modulo (remainder) operation); see example "drumkit 3 (multisynth)";
- 74 - Output port number = polyphonic channel % number of connected outputs;
- 7D - Show Curve1: *output velocity = curve1(input note)*;
- 7D - Show Curve2: *output velocity = curve2(input velocity)*;
- 7D - Show Curve3: *output pitch = curve3(input pitch)*.

How to use option 78 to dynamically change the scale (curve3) from any pattern:

1. write 0001 (ON) to controller 78 of the MultiSynth;
2. send some notes to the MultiSynth - the new scale will be created from these notes;
3. write 0000 (OFF) to controller 78 to stop recording notes.

Videos:

- MultiSynth
- MultiSynth #2

Pitch2Ctl

This module converts the incoming notes to the controller values (in some another connected module).

Controllers:

- Mode:
  - frequency - use frequency (in Hz) of the incoming notes;
  - pitch - use linear pitch (**first note ... first note + number of semitones**) of the incoming notes;
- Note OFF action - action which will be performed after the note release:
  - do nothing;
  - pitch down (to the **OUT min**);
  - pitch up (to the **OUT max**);
- First note (only for the mode **pitch**); corresponds to the **OUT min** value of the controller;
- Number of semitones (only for the mode **pitch**) between the **OUT min** and **OUT max**;
- OUT min;
- OUT max;
- OUT controller.

Pitch Detector

Pitch Detector tries to detect the pitch of the incoming audio signal. The frequency and note will be displayed. Notes will be sent to the module output - synths connected to the Pitch Detector output will play in unison with the original sound.

Controllers:

- Algorithm (selected depending on the characteristics of the original signal):
  - 0 - fast (zero crossing): for basic periodic waveforms (sine, triangle, saw, square); the fastest and with the lowest latency;
  - 1 - autocorrelation: for sounds with a complex spectrum;
  - 2 - cepstrum: for sounds with a complex spectrum; compared to autocorrelation, it performs worse at low frequencies and better at high frequencies;
- Threshold - the detector will only accept a signal with an amplitude greater than this value;
- Gain - output note gain;
- Microtones: 0 (off) - detect only whole semitones (snap to grid of the 12-tone equal temperament (12-TET)); 1 (on): detect microtones (notes may not fit into 12-TET grid);
- Detector finetune - fine tuning of the detector base frequencies: -256 (semitone lower) ... 0 ... 256 (semitone higher);
- LP filter freq (Hz) - cutoff frequency of the low-pass filter at the module input; frequencies above this value will be suppressed - this can help remove noise interfering with the detector; 0 - filter off;
- LP filter roll-off (dB per octave);
- Alg1-2 Sample rate (Hz) - sampling rate of algorithms 1 and 2: a lower value will reduce the CPU load, but also reduce the quality of detection;
- Alg1-2 Buffer (ms) - buffer size of algorithms 1 and 2: less buffer size = less latency; larger buffer size = better detection quality;
- Alg1-2 Buffer overlap (%): less overlap = less CPU load; the higher the overlap, the more detailed changes in sound volume and frequency are detected;
- Alg1 Sensitivity (absolute threshold) - influence of neighboring harmonics on the result of algorithm 1; the exact value is selected experimentally, depending on the characteristics of the original signal; the typical value is 10;
- Record notes (off/on): if this option is enabled and recording is in progress (the REC button is active), then all notes from the output of this module will be recorded into a new pattern.

Example (sunvox/examples/simple_examples): pitch correction (automatic tuning)

Video: Pitch correction (automatic tuning)

Sound2Ctl

This module converts the audio signal to the numeric value of any selected controller.

Input port of the Sound2Ctl is for receiving the audio signal. Output port of the Sound2Ctl is for transmitting the commands like "set controller X to the value Y".

If you want to disable this module: just mute it, or set OUT Controller to 0.

Video: Sound2Ctl Module

Velocity2Ctl

This module converts the velocity parameter of the incoming notes to the controller values (in some another connected module).

Preferences

Go to the main menu -> Preferences, and you will see the window with the main SunVox settings (not for the separate project, but for the whole SunVox app).

- Main
- Modules
- Timeline
- Interface
- Audio
- Recording
    - Metronome ON/OFF.
    - Fit pattern to grid: when enabled - the start position of the new recorded pattern will be multiple of 32.
    - Note quantization: when enabled - the Delay effects (40…5F) will not be used for the new recorded notes.
    - Automation quantization: when enabled - the Delay effects (40…5F) will not be used for the automation (controller changing) commands, and the pattern size will be much more compact (one track per controller) and readable.
- MIDI

Configuration file

In most cases, you should not change this file manually, because the main settings are available in the SunVox Preferences window. But if you want to use some specific options, you can create your own configuration file based on this one (description of all options).

Links

- Forum
- Video tutorials
- Modules
- SunVox music on SoundCloud
- SunVox group on FB
- SunVox library for developers
- JavaScript SunVox Player

Frequently asked questions (FAQ)

Interface

Interface is lagging very much

Try to disable some module visualization options: Main Menu -> Preferences -> Modules.

I noticed the effects parameters on the right of the note, velocity etc at the top seem to be

missing

These columns are hidden by default. But you can enable them - just go to the Main menu and select "Show additional columns".

Modules, samples, effects

How to connect one module to another?

The answer is here.

Filter effect's impulse value doesn't change. When i change impulse value, it return 0

It is normal behaviour. There is no need to remember the "impulse" value. I'll try to describe the algorithm.

1. Frequency = 1000 (for example).
2. You set "Impulse" to 3000.
3. Filter sets its internal frequency to 3000.
4. Filter resets "Impulse" to 0. (0 = no more commands from user).
5. Frequency changes smoothly from 3000 to 1000. Speed depends on the "Response" value.

Is there a particular bit rate/sample rate/# channels that the samples need to be in order to work?

WAV samples must be in following format:8 / 16 / 24 / 32 (float) bits; mono / stereo;any sample rate; only uncompressed PCM data. Supported file formats: WAV, AIFF, XI, JPEG. As for JPEG, brightness of the pixels will be converted into the sound wave (pixel by pixel).

How to open the sample editor?

Please check this video-tutorial.

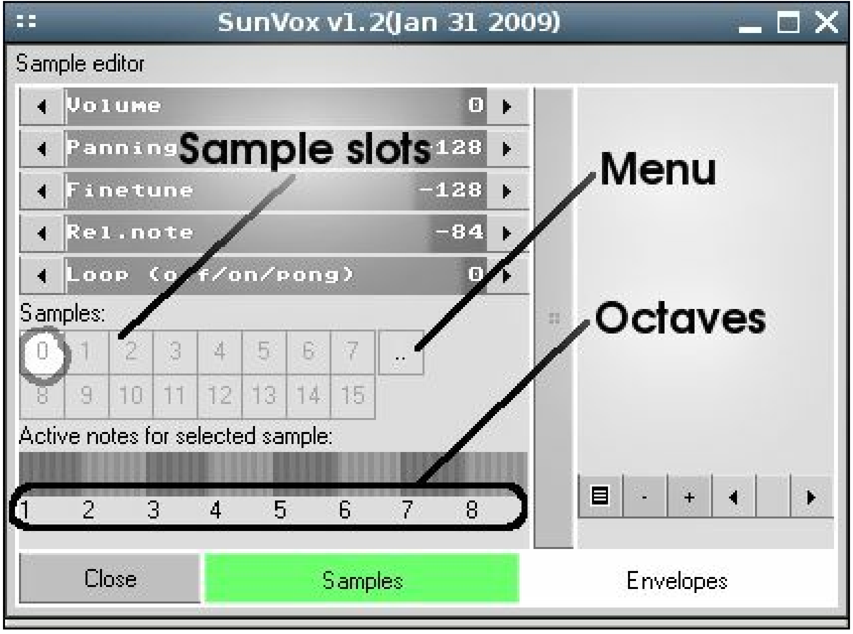
For multisamples, how do I set the root note of my sample?

Use the "rel.note" (relative note) parameter for such purposes.

How do I adjust multisample instruments? I see slots for multiple samples and a zone managment section in the editor, but I can only load one sample at a time?

I hope this little tutorial will help you.

Step 1. Open new empty sampler:


SunVox v1.2(Jan 31 2009)

Sample editor

Volume 0

Panning 128

Finetune -128

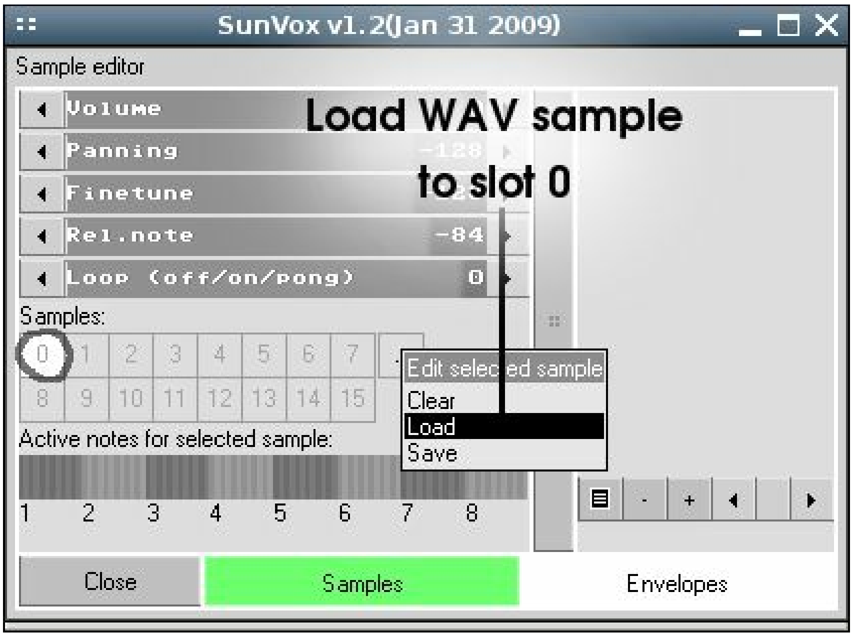
Rel.note -84

Loop (off/on/pong) 0

Samples:

Active notes for selected sample:

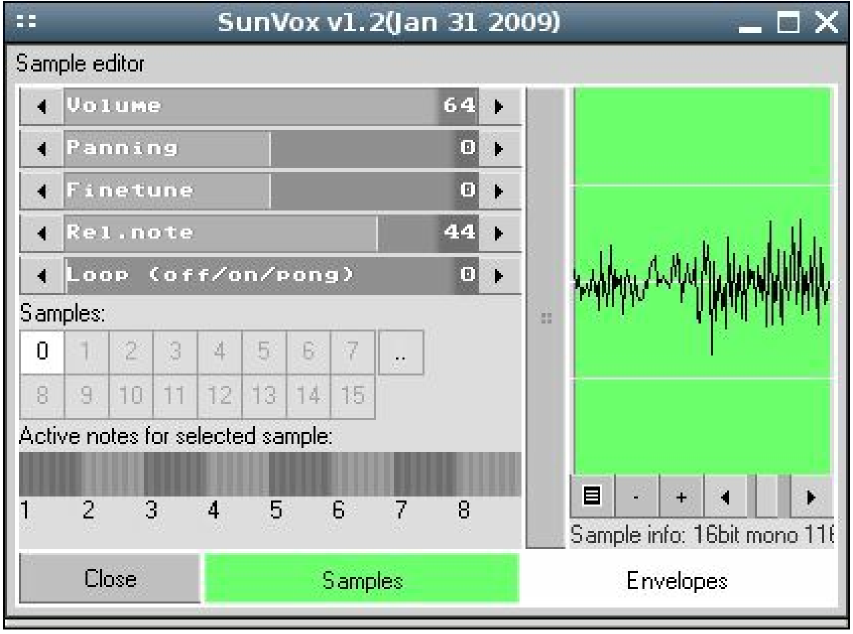
Step 2. Load sample 0:



Volume
Panning -128
Finetune
Rel.note -84
Loop (off/on/pong) 0

Load WAV sample to slot 0

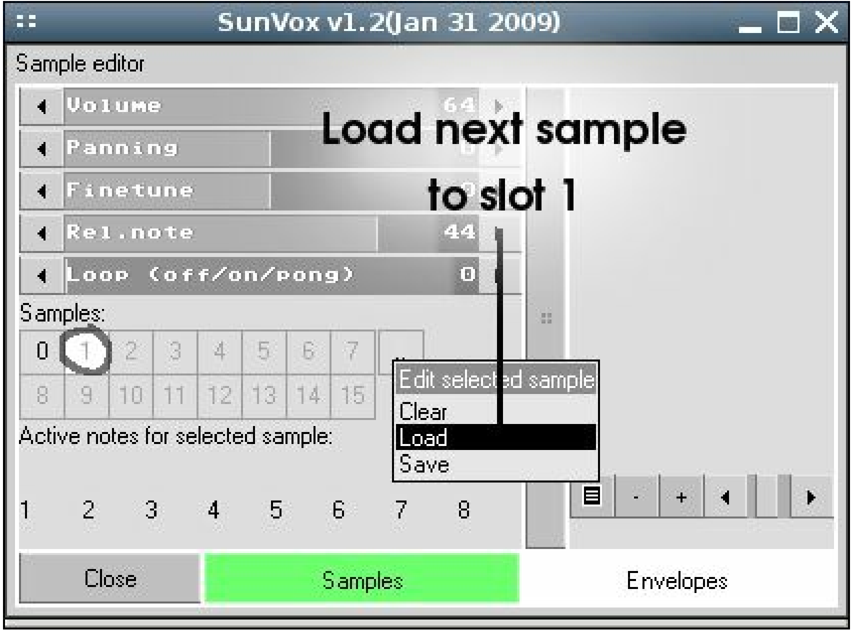
Step 3. Sample 0 loaded:



Volume 64
Panning 0
Finetune 0
Rel.note 44
Loop (off/on/pong) 0

Sample info: 16bit mono 11

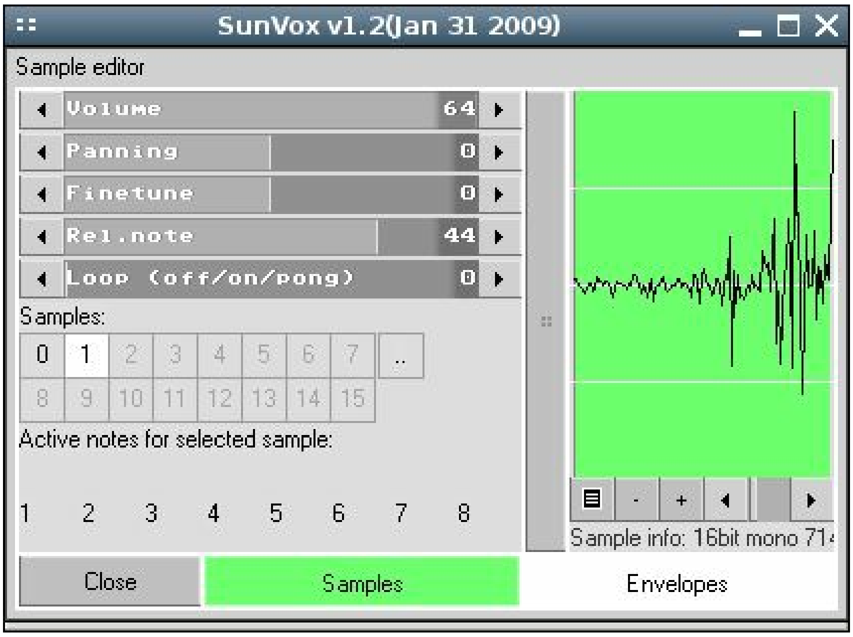
Step 4. Load sample 1:



Volume 64
Panning 0
Finetune 0
Rel.note 44
Loop (off/on/pong) 0

Load next sample to slot 1

Step 5. Sample 1 loaded:



SunVox v1.2(Jan 31 2009)

Sample editor

Volume 64

Panning 0

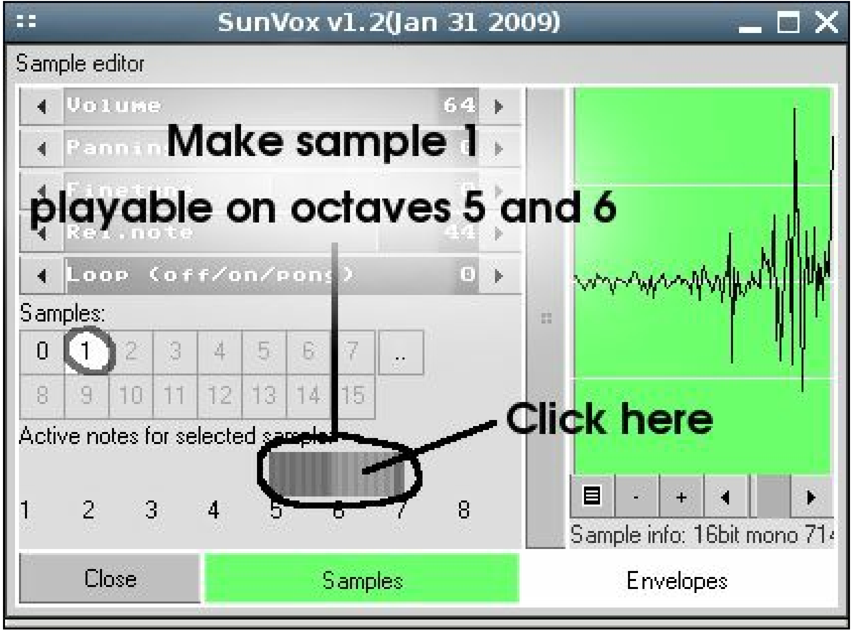
Finetune 0

Rel.note 44

Loop (off/on/pong) 0

Sample info: 16bit mono 71

Step 6. Make sample 1 playable on octaves 5 and 6:



SunVox v1.2(Jan 31 2009)

Sample editor

Make sample 1 playable on octaves 5 and 6

Click here

Sound

Sound stutters or (and) freezes. How to fix?

General tips

1. Make sure that the CPU load is not too high in your project. Check some lightweight project (for example, NightRadio - Tiny Tune).
2. Open SunVox Main menu -> Preferences -> Audio and change the Sample rate to some other value if you know the native frequency of your sound card.

Windows: Open SunVox Main menu -> Preferences -> Audio and set Driver to ASIO; if there are no ASIO drivers for your sound card, install a universal driver (for example, ASIO4ALL).

Android: Open SunVox Main menu -> Preferences -> Audio and try to change the Sample rate to 44100 (if it's 48000 by default).

Linux

1. Go to the SunVox Preferences -> Audio and check the Output field. If it is set to Auto - change it to some real audio output device from the drop-down list.
2. If this does not help, try to launch the SunVox with the following command: **pasuspender -- ./sunvox** . This will open an exclusive access to the sound card for the SunVox.

Windows CE: go to the SunVox Preferences -> Audio and change the Buffer to some value between 1024 and 4096; for example, 1536 is the best value for iPAQ devices.

MIDI

How to connect SunVox to another app via MIDI?

- Windows: use LoopMIDI to create the virtual MIDI cable between the apps; in SunVox this port can be used as MIDI input (SunVox -> Preferences -> MIDI) or MIDI output (SunVox Module Properties -> MIDI OUT).

How to split MIDI notes by channels, and send the notes from different channels to different modules?

Go to the properties of the module and set MIDI IN to "always", if this module should play always (even when not selected). Next, change the channel value to the right of MIDI IN.

Other

SunVox crashing for no apparent reason

Try to clean the settings of the SunVox. Launch it from console with the following command: sunvox clearall. If you are Windows user - just run the reset_sunvox.bat script.

iOS

How to import/export files from/to the SunVox internal storage?

There are several different ways

- Use the **Files** app.
- Use iTunes File Sharing to access the SunVox files. Launch the iTunes on your computer. Sync it with iOS device. Then find SunVox in the list of apps of your device. Select it and you will see the content of the internal SunVox storage.
- In some other app (for example, Mail) click on the audio file and select**Open in** -> **SunVox** (this function may be hidden in some menus).
- Use one of the many features presented in SunVox -> Main menu -> Export/import:
  - Wi-Fi export/import (can work even without Wi-Fi (locally) to access files from a browser on the same device);
  - Copy WAV file to system pasteboard;
  - Paste WAV file from system pasteboard;
  - Open in... (open SunVox file in some other app);
  - System export (copy file from SunVox to another place on your device);
  - System import (take SunVox-compatible file from another place on your device).

I cannot find files from the standalone app when using AUv3 version of SunVox

AU and standalone app store their files in different locations. You can use the following file transfer methods.

Method 1

1. Open SunVox AU.
2. Maximize the window with SunVox.
3. Go to the SunVox main menu -> Export/import -> System import -> iOS document picker will open.
4. Find the folder of the standalone version of SunVox -> pick some file -> selected file will be opened in the AU.

Method 2

1. Open SunVox AU.
2. Go to the SunVox main menu -> Export/import -> Wi-Fi export/import.
3. Open Web Browser on the same device and enter the following address: localhost:8080
4. You will see the file storage of the AU-version of SunVox.
5. Click the file select button at the top of the page -> iOS document picker will open.
6. Find the folder of the standalone version of SunVox.
7. Pick some file.

8. Press SEND.
9. File will be sent to the folder of the AU.
10. Go back to AU and close the Wi-Fi window.

Android

No signal from the microphone

Open the Android Settings -> Apps -> SunVox -> enable the "record audio" permission.

Known solutions for other problems

macOS

Copy/Paste don't work, the previous session is not saved

Please read this instruction