

SBleepo2 VST Plugin

In this manual:

Helpful Safety Tips

presented by:



Loop Generator and Death Synth Emulator by dsperado.
Noah Vawter: Algorithm, perl code
Brian Whitman: C-code, VST Plug-In, GUI
<http://dsperado.com>

v1.10 3/21/2001 NTV

Table of Contents

Sound

- Oscillators 3
- Sequencers 3
- Combine Operations "Combyops" 4
- BPM/Measures 4

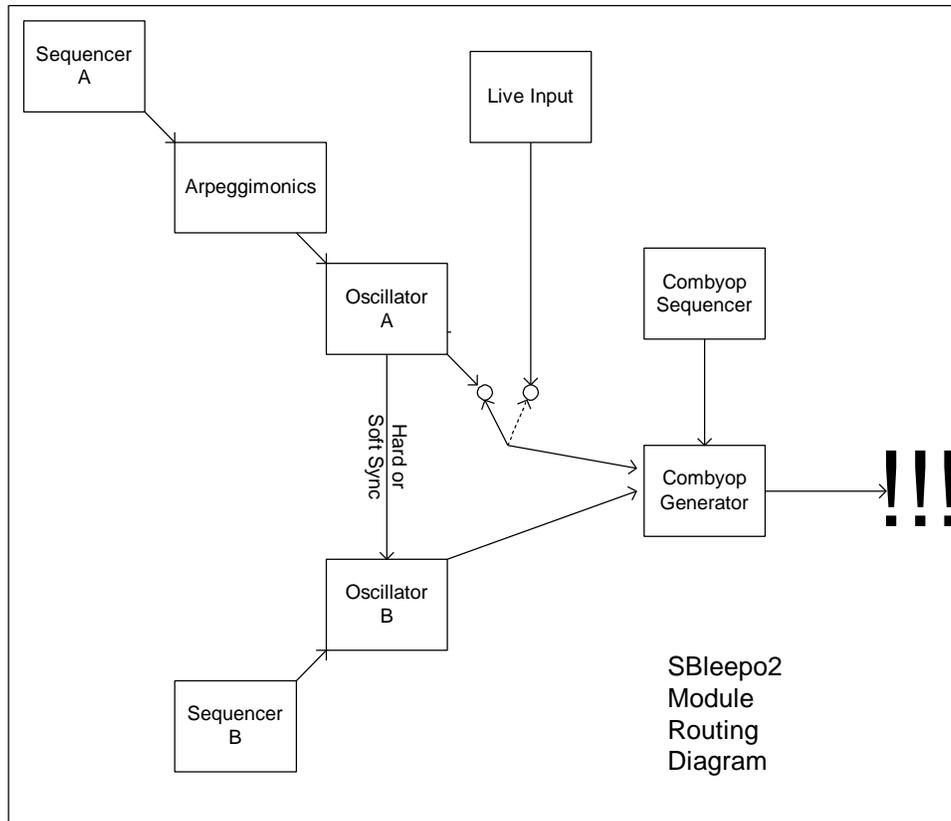
Effects

- Live Input 6
- Oscillator Synchronization 6
- Arpeggimomics 7

Reference

- Clocks <-> note length conversion 9
- Description of Combine Operations 10
- Theory of Operation 11

SoundSoundSoundSoundSoundSoundSoundSound



Osc. A and Osc. B

Use the sliders to pick the sound you really like

- 1) Mix between sawtooth waveform and pulse waveform
- 2) Duty cycle of the pulse waveform (0-100%)
- 3) Portamento
- 4) Frequency multiplier

Click on a number and type a new one in to control the oscillator frequency.

0 can be entered to stop the oscillator (make a rest).

0 can also be used with portamento to create kick drum type sounds.

The frequencies entered are into a linear scale, therefore, to create an octave series, use e.g. 10 20 40 80.

Sequencers

Each oscillator has a sequence of 1-10 frequencies, which are stepped through from left to right.

Change the number of "beats" to set the number of steps in a sequence.

Click on "beats" to select 1-10 beats.

When the sequencer reaches the last step, it restarts all the way from the left.

The number of "Clocks" controls the speed of the sequencers.
"Clocks" means "How many ticks until I move to the next step?"
Try beating the sequencers together: run one at 6 beats, the other at 5.
Or, beat the Clocks together, one at 6 Clocks, the other at 8 Clocks.
Even though there is no MIDI ability, it is modelled after MIDI clocks.
Therefore, 6 clocks means 1/16th notes, see Reference Section at end for more.

Combine Operations - "Combyops"

When you first turn on SBleepo, you will only be listening to Osc. A.
When you change the first A in the row to B, you will hear only Osc. B.
To change the A, click on it, then select B.
To hear A and B at the same time, change the B to "A+B".
To combine A and B in different ways besides addition, select other combinations such as A*B.
Every *combyop* creates a unique spectral distribution which varies wildly depending on the input waveforms shape and frequencies. See Theory of Operation section at end.

You should sequence the combine operations just like the two oscillators.
Click on "beats" to pick the length of the combyop sequence.
Modify the "Clock" value to change its speed.

Try sequencing the combyops at the same speed as one of the oscillators: use the same "Clock" value as one of the other oscillators.
Or, use a combyop "Clock" value such as 4 with a sequencer "Clock" of 12 to cycle through 3 combyops for every sequencer step.
Try "beating" the combyops sequencer against the other two sequencers: e.g. use a combyop "beats" value of 7 with a sequencer "beats" value of 8.

Tips:

Make rhythms using the combyop, such as A A A B A A A A*B

Percussive effects can be generated with fast "Clocks", such as 1, when using a sequence like:
C64 A*B -?A^B Max

BPM/Measures

In the bottom, right-hand corner of the screen, notice two windows "M" and "BPM".
Click on the number next to BPM, and type a new number to change it. This value is in Beats Per Minute, assuming 4/4 time. Example BPM settings are:

90 - reggae
105 - rap tempo
120-125 - video game music
125-133 - techno
180 - jungle tempo

No matter how crazy you make all of your settings, the loops generated by SBleepo will always perfectly match your BPM setting.

The "M" window represents a number of measures.
Normally, SBleepo resets itself after every measure, but when you make combinations of beating sequencers, you want the loop to continue playing until the two step lengths have caught up with

each other, so click on this number and select a new value to make the loops take longer before repeating.

EffectsEffectsEffectsEffectsEffectsEffectsEffects

Live Input

If you like, click on Squ-Saw to toggle “Live Input”.

In “Live Input” mode, Osc. A will not be used.

Instead, your signal input to super-bleepo will be used for all combinations with Osc. B.

For example, you can effect anything you input by choosing combyops and sequencing Osc. B.

Note: this version of SBleepo does *not* re-pitch the Live Input...

Osc Sync

This makes Osc B. react to Osc A in a way which creates a potentially beautiful and dazzling array of spectral content.

By default, this slider is set all the way to the right, which means no oscillator sync.

When you slide this toward the left, you enable oscillator sync and set the “hardness”.

Each time Osc A completes a full cycle, it sends a pulse to Osc. B.

When that pulse arrives, Osc. B resets its phase back to zero if oscillator sync is 100%

100% is also known as hard sync, however, you should also generate soft sync.

When soft-syncing, the phase of Osc. B is only reset by a certain amount.

For example, slide $\frac{1}{4}$ of the way to the left to create a value of 25% sync.

Osc. B's pitch will slightly track the pitch of Osc. A.

There are two special ways to use Osc. Sync. if you desire:

Tune the Freq Mult of Osc. B down very low.

Tune the Freq Mult of Osc. A to the upper half. Set Osc. Sync to 75-100%.

Set the combyop to Osc. B only.

Since Osc. A's output is at such high frequency, Osc. B will rarely, or never complete its cycle before being re-triggered. Therefore, the frequency of Osc. B serves merely as a *tone control* for the sound, while Osc. A drives the pitch!

Alternatively, you may follow the above instructions, but set Osc.A to a very low “Freq Mult” and Osc. B to a very high “Freq Mult”. Now set the “Portamento” of Osc. A to very long. The sound will most closely resemble Osc. B, yet Osc. A will affect it, adding a spectral sweep due to the portamento. Keep experimenting, you will undoubtedly create sounds no one has ever heard before.

Arpeggimionics

The special technique of arpeggimionics is devised to vary the harmonic structure of Osc. A It is invented by us, so SBleepo is the *original implementation* of the arpeggimonic sound. This trick rapidly changes the fundamental frequency of an oscillator like an arpeggiator. You provide a sequence of multiples, and SBleepo will precisely multiplex them.

Use the **Length** to describe the length of your sequence, and click on values to change them.

The **Period** field indicates how many cycles of each waveform to play before skipping ahead to the next step in the sequence. When the end of the sequence is reached, it starts again from the beginning.

Rate should usually be set to 1.00. However, if you desire, you can affect the period with the "Rate" parameter. A value greater than 1 causes the period to grow over time, making your arpeggimonic sequence slow down. There is no real limit to how slow it can go, however, eventually the period will become so long that you can no longer hear the sequence skip steps. Values less than 1 make the period grow short, which speeds up your arpeggimonic sequences.

Try values of 1.02 for a slow speed-up or 0.99 for a slow-down.

The sequencer resets the value of the period to its original value every time the number of measures in the Measure window is reached.

Example Parameters

You can use Arpeggimionics to create many styles of effects. For example:

Vibrato:

Using these series with a medium period like 10-40 will create a vibrato effect.

Vibrato1: 1.00 0.90 1.00 1.10

Vibrato 2: 1.00 0.95 1.00 1.05 (more subtle)

Vibrato 3: 1.00 1.02 1.03 1.01 1.04 1.02 (very subtle)

Chords:

Make an arpeggiated major chord with a sequence like: 1.00 1.25 1.50.

Minor chord: 1.00 1.20 1.50

Dominant 7th: 1.00 1.25 1.50 1.89

A series of octaves and fifths: 1.00 1.50 2.00 3.00 4.50 6.00

Or simply octaves: 8.00 4.00 2.00 1.00

Or simply triads: 1.00 1.25

Or the Diminished Fifth, a.k.a Demented Fifth, a.k.a. Tritone: 1.00 1.41

Sub-Oscillator:

These work best with ultra-short periods like 1-3.

Simple suboscillator: 1.00 0.50

Improved Suboscillator: 1.00 1.00 0.50 1.00 0.50 0.25

Non-Harmonic Suboscillator #1 1.00 0.50 1.00 0.75

Non-Harmonic Suboscillator #2 1.00 0.66 1.00 0.75 0.50 0.33

Bass Rolls:

1.00 0.90 0.70 0.50 0.30 0.10

1.00 0.75 0.66 0.50 0.33 0.25

Tube Sound:

Some people postulate the warm sound of tubes is due in part to their addition of even harmonics to a sound. Perhaps you might be able to make your SBleepo sound warm and fat with these arpeggimonic sequences:

Tube 1: 1.00 2.00

Tube 2: 1.00 4.00 2.00 4.00 2.00 4.00

Tube 3: 10.00 4.00 8.00 2.00 6.00 4.00

Randomize:

Use these with long periods 40-100 for sloppy, video game explosion sounds

Medium periods 10-40 for warbling sounds

Short periods 1-10 for spectral color

Video game explosion: 2.76 3.39 4.34 1.57 3.63 2.49

Subtle texture: 1.00 1.02 0.93 1.03 0.96 1.09

Very large values tend to have unusual effects. E.g. you can set the oscillator to spin beyond the Nyquist rate, which will generate unusual aliasing sounds like a spaceship.

Reference Section noitceS ecnerefeR

Clock <-> Note Length Conversion

3 Clocks = 1/32nd notes

6 Clocks = 1/16th notes

12 Clocks = 1/8th notes

24 Clocks = 1/4th notes

48 Clocks = 1/2 notes

96 Clocks = 1 Measure

1 Clock = 1/96th notes

9 Clocks = dotted 1/16th notes

18 Clocks = dotted 1/8th notes

36 Clocks = dotted 1/4 notes

72 Clocks = dotted 1/2 notes

32 Clocks = 1/3 notes (you will get 3 1/3rd notes per measure)

16 Clocks = 1/6 notes (you will get 6 1/6th notes per measure)

8 Clocks = 1/12 notes (you will get 12 1/12th notes per measure)

4 Clocks = 1/24 notes (you will get 24 1/24th notes per measure)

2 Clocks = 1/48 notes (you will get 48 1/48th notes per measure)

Combyop Descriptions

A+B Simply adds the two oscillators together, like common synthesizers.

A+B! Adds the two oscillators together without pre-attenuation. This provides the ability to generate clipping, adding a small number of overtones.

A*B Multiplies the two oscillators together, generating sum and difference frequencies. This is the same as a ring modulator.

-?A&B Quantizes the input signals to create 1-bit values, then logically AND's them together.

-?A^B Same as -?A&B, except Exclusive-OR'ing generates a different harmonic series.

-?A|B Same as -?A&B, except OR'ing generates a different harmonic series.

Max Picks whichever of the oscillators has the highest amplitude in the moment.

Cond secret

A<<B Quantizes Osc. B's output into 16 steps, then outputs the value of Osc A shifted by that number of bits. Positive values of B shift A to the left, Negative B values shift A to the right.

*,+ A mixture of multiplication and addition of the two oscillators. Addition is weighed slightly heavier, since multiplication generates higher frequencies with more spectral power than addition.

C64 Emulates the ring modulation algorithm of the SID chip in the Commodore 64. From an Andreas Varga interview with Bob Yannes found at <http://stud1.tuwien.ac.at/~e9426444/yannes.html> :

"Ring Modulation was accomplished by substituting the accumulator MSB of an oscillator in the EXOR function of the triangle waveform generator with the accumulator MSB of the previous oscillator."

BsA This functions somewhat like a Sample and Hold circuit. It samples A whenever B is positive and holds when B is negative.

S/R This emulates a Set/Reset Flip-Flop in the phase detector of a Phase Locked Loop circuit. The output is set to 1 when Osc. A transitions from 0 to 1, and goes to zero whenever Osc. B transitions from 1 to 0.

A&B This is the bitwise AND of the entire 32-bit value of Osc. A & Osc. B.

A|B This is identical to A&B, except A|B generates a different harmonic series.

A^B This is identical to A&B, except A^B generates a different harmonic series.

Theory of Operation

For the past 200 years, multi-oscillator synthesizers have used multiple oscillators to generate sounds with intricate overtones. Summing the output of 2-4 tuned oscillators in tune (or hundreds, in some cases) generates complex sounds which invite multiple listenings. The formula has been subjected to only slight variation in commercial applications. For example, many synthesizers are able to perform sync or ring modulation between two oscillators. SBleepo does sync, ring, and a dozen more non-linear operations!

For what feels like such a long time, non-linear combinations of two oscillators have been ignored. Perhaps this is due to their unpredictability and tendency to generate anharmonic overtones? What overtones DO they in fact generate? In general, it can be said that non-linear combine operations, such as those SBleepo serves up generously, create overtones which are harmonic series of the sum and difference frequencies. They create intricate sets of overtones, which one can more or less describe through this formula. (The “more or less” qualifier is because some of the functions SBleepo does have a *state* to them, which makes them even harder to characterize; they are not merely lookup tables):

$$\text{sum } [n=0..\text{Inf}, m=0..\text{Inf}] (n*(A+B)) + m*(A-B))$$

For example, multiplying a 1500 Hz and 4000Hz sine wave together generates the sum and difference frequencies with no overtones: 2500Hz and 5500Hz. That is a simple case. If you perform other nonlinear operations on the same two waveforms, you will generate frequencies which are integer-multiples of 2500Hz and 4500Hz, e.g. 2500, 5000, 7500, 10000, etc and 4500, 9000, 13500, 18000. Furthermore, the volumes of the frequencies in those overtones vary with the combyop used.

Still following along? Good, because this system is actually more fascinating. You see, when two *non-sinusoidal* waveforms non-linearly combine, essentially even more frequencies are injected into the system. And every single combination of frequencies generates a harmonic series.

Say, for example, you input a waveform with harmonics of 1500Hz, 3000Hz and 4500Hz. The other input to the system adds 4000Hz, 12000Hz, and 20000Hz. If you were to multiply those two signals together, the result would be a summation of signals with 18 components:
1500 +/- 4000, 1500 +/- 12000, 1500 +/- 20000
3000 +/- 4000, 3000 +/- 12000, 3000 +/- 20000
4500 +/- 4000, 4500 +/- 12000, 4500 +/- 20000

(nevermind the fact the you can't hear many of those for now).

Finally, if you were combine those two input signals with a more complex combyop than multiplication, the result would be 18 entire harmonic series, each series with a fundamental from the list above! So you can see that combining two oscillators with varying waveforms and non-linear combyops can generate a dazzling array of overtones. This is what SBleepo was designed to do!

Bugs:

When re-starting SBleepo, it sometimes takes a few measures to converge on the exact sound. We suggest you take advantageous of this.