Ripplemaker

Bram Bos

User Manual

version 1.0

Version history

Version history (this document's state reflects the latest available software version:
1.0 June 13, 2017 Initial publication

Down the rabbit hole

Welcome to the modular world of Ripplemaker. For some it's a first glimpse into a mysterious corner of the world of synthesizers, for others may be a gateway down the rabbit hole; eventually leading to a massive wired wall of blinking, bleeping modules somewhere in the studio. Either way, Ripplemaker offers a taste of the wonderful sound shaping freedom of modular synthesis.

When I set out to create Ripplemaker, my primary objective was to capture the fun of experimentation. Most of us already have dozens of synthesizer apps to efficiently deliver any sound needed in our productions and jam sessions. But for many musicians these days modular is an escape pod into a different dimension of musical expression. The appeal of modular is not necessarily in its production-efficiency, but in the way it invites exploration of the nature of sound, and how it brings out the different characters of the patches we create - as if they were sonic organisms.

Sometimes you just want to sit down with a handful of (virtual) wires and feel like a kid with a science kit; losing yourself in pulsating waves and wobbling modulations.

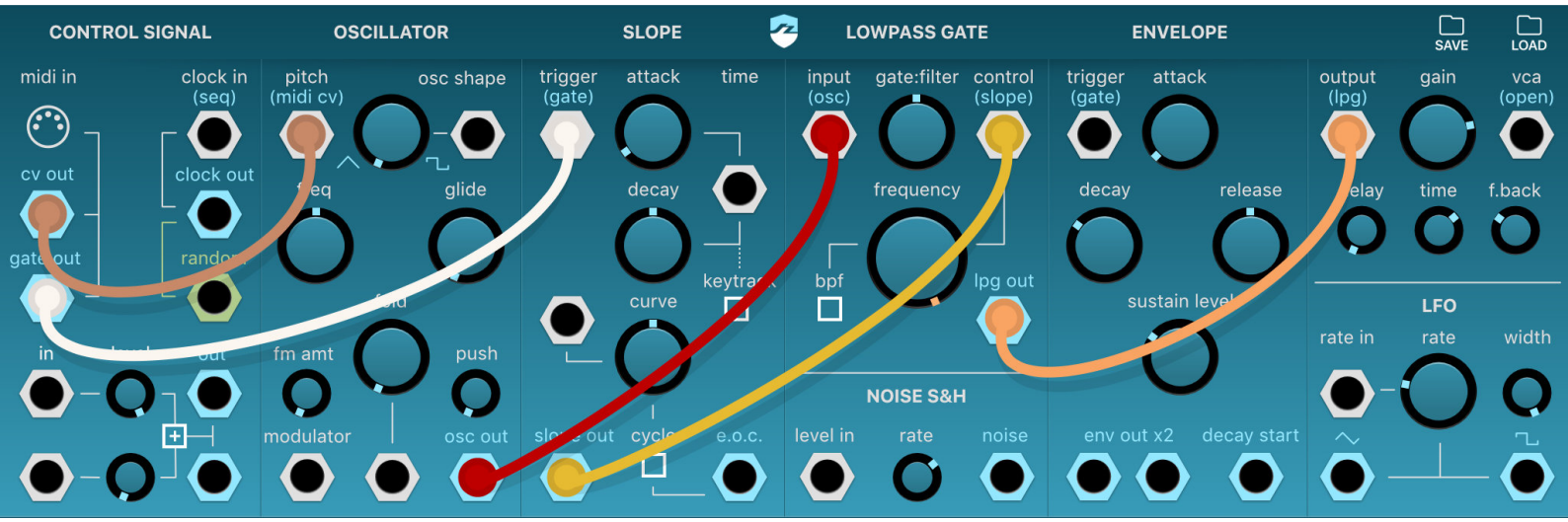
East Coast, West Coast

In modular synthesis, two dominant schools of thought are typically recognized; East Coast synthesis and West Coast synthesis. Many entertaining nights can be spent discussing the exact definitions of either, but for now a high level description will suffice. Generally, East Coast synthesis is associated with the synth designs of Bob Moog. His pursuits were focused on using (mainly subtractive) synthesis to convincingly recreate musical instruments for popular music genres (bass, guitar, brass, pianos, etc.). His world-famous ladder filter design is one of the mainstays of East Coast instrument architectures.

West Coast synthesis on the other hand evolved in California around synthesizer pioneers like Don Buchla and Serge Tcherepnin. Their approach was much more experimental; exploring the possibilities of sound synthesis with overtones and (in)harmonics through additive techniques. Where subtractive synthesis starts with a rich signal which is harnessed using filters and envelopes, Buchla's designs start with basic waves (sines, triangles) which are then stretched, folded and modulated into a colorful mosaic of complex overtones and morphing frequencies. Sound for sound's sake.

Although not nearly as complex as a Buchla Music Easel, Ripplemaker starts from these same West Coast principles. Hence you won't find a traditional voltage controlled filter, but instead a Buchlaesque 'lowpass gate' and a 'complex oscillator' section which starts with a triangle and lets you dial in the required complexity.

1. Modular, Semimodular, Normalized Connections



Ripplemaker is what is generally known as a *semimodular synth*. This basically means that the modules that make up the synth are already chosen for you, and -more interestingly- are already wired up for you.

> **Note:** all output jacks are blue, input jacks are white (only the random output is green to make it stand out). You can patch from input to output or output to input; the order does not matter.

Pre-wired connections are called *normalized connections*. If we were to make the connections ourselves, they would look like in the picture above: the Oscillator and the Slope generator are patched into the Lowpass Gate. The output of the Lowpass Gate goes into the Amplifier. And the MIDI input (from the sequencer) goes in the form of CV and Gate signals into the Oscillator (CV) and into the Slope and Envelope modules (Gate).

This default wiring gives you a pretty flexible monosynth, without needing any wire-patching.

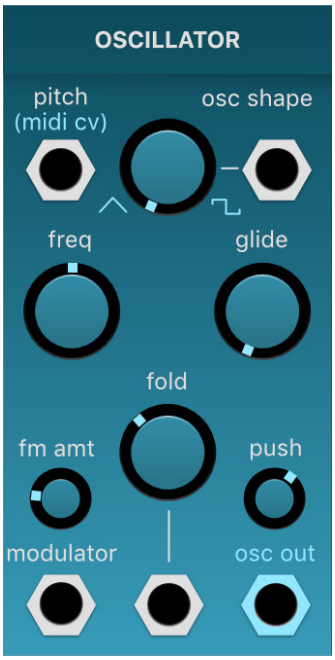
> **Note:** normalized connections are indicated between blue brackets on the frontpanel. You can override normalized connections by plugging something else into the white jacks.

When you plug something into a normalized input jack (white) it will override the normalized connection. For example: if you plug the LFO into the control-jack of the Lowpass Gate, it will break the normalized connection coming from the Slope.

On the other hand, normalized ouputs are not stolen when you use them for something else. For example: when you patch the Slope output into the Fold input, the normalized connection between Slope and Lowpass Gate will also still exist. That's neat; it effectively gives you two Slope outputs!

> **Note:** when the synth does something unexpected, it is usually because you forgot about a normalized connection somewhere. E.g. switching the Slope to oscillator mode will also make the Lowpass Gate open and close really fast. Unless you want to do filter FM you better override the LPG's normalized control input!

2. Oscillator Section



The Oscillator is the primary module responsible for generating sound. It is capable of creating waveforms of varying complexity across a broad range of frequencies.

Oscillator shapes
The simplest sound that can be made by the Oscillator is a triangle. It's the root of all other sounds that it produces. Hence this type of oscillator is typically called a "triangle core" oscillator.

Through a voltage controlled waveshaper, the triangle can be morphed into a square wave.

This shaper is controlled by a CV input jack and an attenuator knob. When nothing is plugged into the jack, the knob directly determines the shape of the wave. E.g. when it is set to 100% the wave will be a square wave, at 50% it will be something halfway between a square and a triangle wave.

When a cable is plugged into the osc shape jack (e.g. an LFO or an Envelope) the knob becomes an attenuator, scaling incoming voltages down to the maximum setting indicated by the knob. For example: if the knob is set to 50%, then the incoming values - e.g. from an LFO - are scaled to move up and down between 0% and 50%. If the knob is set to 100%, the LFO will go the full range from 0-100 per cent.

> **Note:** attenuator knobs scale incoming control voltages down. Input values are scaled (not clipped!) to fit between 0% and the knob's value. Most knobs in Ripplemaker work like this.

Pitch & Frequency
The base frequency of the oscillator is tuned with the freq[uency] knob. You can use this knob to tune your synth to other instruments or to pitch everything up or down by an octave or two. The base frequency can be further modulated by incoming pitchbend MIDI messages (+/- 2 semitones).

The actual pitch of the generated waveform is determined by the control voltage going into the pitch jack. By default this input jack is normalized to the MIDI CV source, coming from the sequencer.

However, you can plug anything into this pitch input, such as an LFO or a random voltage, allowing you to create some pretty wild non-musical (unscaled) frequencies if you want to.

The glide knob determines how fast the oscillator will sweep from the current frequency to the next. When glide is set to 0, the change will be as fast as the virtual analog components will allow the voltage to jump. When the values are higher, the change will take longer.

> **Note:** the pitch glide is fixed-time, not fixed-rate. This means that it will always take the same amount of time to reach the new frequency, regardless of the magnitude of the change.

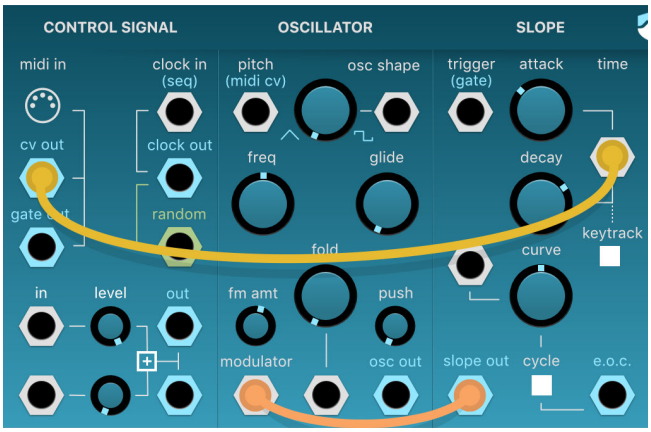
Fold & Push

To add some lovely/lively overtones we can send the oscillator through the wavefolder. A wavefolder is a type of waveshaper that clips an incoming wave and then folds the clipped part of the wave around into the other direction. This can be done multiple times to create gradually increasing harmonic complexity. It can sound a bit like FM, but the algorithm itself is quite different.

Before folding the wave, we can use the Push function to *push* the wave off-center (add a DC bias), creating an asymmetrical waveshape. This adds even more harmonic overtones to the resulting waveshape.

Frequency Modulation

FM, or frequency modulation, is a great technique for adding overtones to a simple sound. It requires two oscillators (typically referred to as *operators* in FM lingo) where one oscillator modulates the frequency of the other. The former is called the *modulator* (responsible for the overtones), the latter is called the *carrier* (responsible for the base frequency).

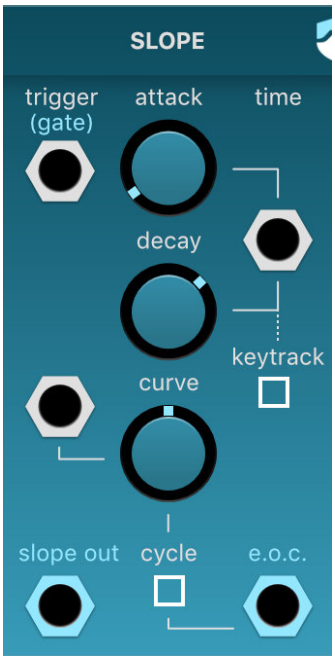


Our oscillator wave is the carrier, so we'll need another waveform to be modulator. Anything running at audio-rate can be used…

Possible candidates for Modulator duty are:

- the LFO (at a very high rate)
- the output of the Lowpass Gate (especially nice in bandpass mode)
- the Slope circuit when set up as a secondary oscillator (see example picture on the left)
- noise, from the S&H circuit

3. Slope Generator



The Slope module is a pretty versatile little unit. It has several uses:
- Attack-Decay Envelope Generator (AD EG)
- LFO
- Secondary oscillator

Attack-Decay Envelope
In its basic setup, the slope unit generates a simple percussive 2-stage envelope, sporting an attack and a decay stage. This type of EG is useful for percussive sounds, bass sounds, bell sounds and other sound types that typically don't need a sustain stage.

The attack and decay times can be set independently. The curve parameter determines the curvature of the slope. At high curve-values the slopes will be almost linear (for a smooth and gradual envelope behavior). At low values the curve will be much more percussive and *plucky.*

All knobs behave as attenuators when their respective CV inputs are used.

Note: the slope will trigger when input signal exceeds a certain voltage (roughly 50%). The envelope will then run its course, regardless of the duration of the trigger signal.

LFO Mode
One can never have too many LFOs. So it's good to know you can turn your Slope unit into an LFO should you need an extra modulation source. Simply enable *cycle mode* by tapping the checkbox labeled 'cycle'. This will make the slope retrigger itself indefinitely.

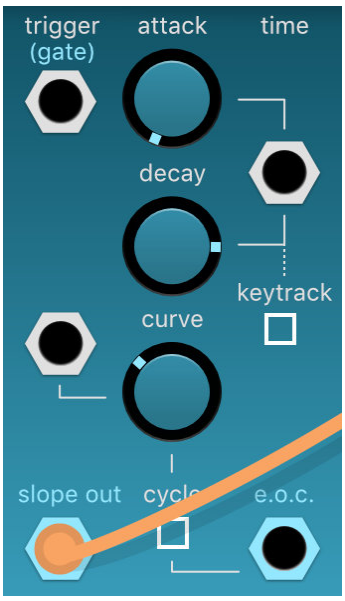
Additionally, the EOC feature will become active. EOC stands for *End Of Cycle.* It will send out a trigger pulse every time the Slope completes a full cycle and retriggers. This pulse can be used to generate an alternative clock source (patch it into the clock jack) or to trigger an envelope (patch it into the Envelope's trigger jack). This can be extremely useful, because it lets you synchronize the generation of a new random value with the start of a new slope cycle.

Oscillator Mode ('keytrack')
Since an LFO is just a wave, just like an oscillator, wouldn't it be awesome if we could speed up the Slope to become another oscillator? Yes, it would. So that's exactly what you can do if you enable the checkbox labeled 'keytrack'…

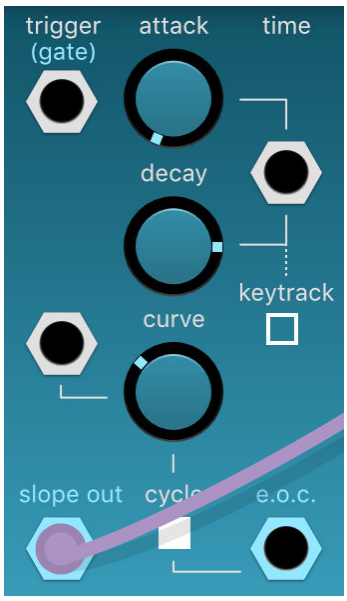
Note: don't forget that the Slope is normalized to the Lowpass Gate! If you mess around with the Slope, you may want to patch e.g. an envelope output into the LPG's control jack first!

Keytracking makes sure that the wavelength of the Slope's wave can be controlled by the same control voltage as our regular oscillator. Simply patch the CV output of the Control Signal module into the time jack of the Slope module.

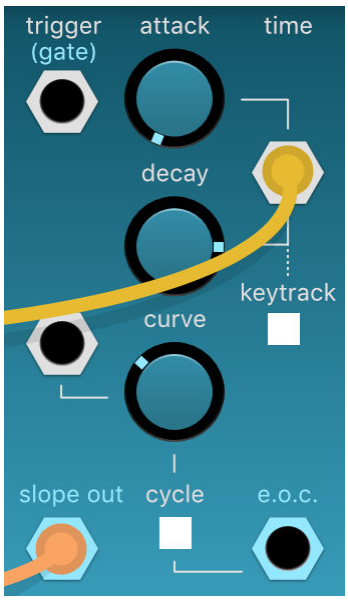
You can now finetune your Slope wave +/- 1 octave using the Curve button.



Mode 1:
Attack-Decay Envelope
(default setting)

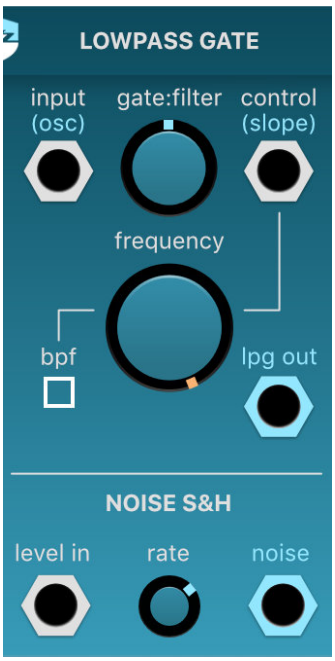


Mode 2:
LFO Mode
(cycle enabled)



Mode 3:
Oscillator
(cycle+keytrack enabled,
CV patched into time)

4. Lowpass Gate + Noise S&H



As mentioned in the introduction, West-Coast style modular setups often don't prominently feature a traditional filter. The Californian substitute for the subtractive filter is the Lowpass Gate (LPG).

Technically, an LPG also features a filter but conceptually it's a different type of module. In short, an LPG combines the function of a VCA and a low-resonant lowpass filter into one unit. The rationale behind such a design is that in nature sound behaves in a similar way. Hence, in an LPG, opening up the filter also makes the sound louder.

Originally, LPGs were designed around *vactrols*, photo-resistive opto-isolators. These components have a very specific non-linear response to incoming voltages that makes them suitable for percussive sounds. But due to manufacturing inconsistencies and tuning variability they have now been mostly replaced by other components.

Gate:filter
In Ripplemaker you can change the balance between the two functions of the LPG; gate (VCA) and filter (VCF). A bias towards the gate side will make the filter effect less strong (full gate setting means the LPG will behave like a pure VCA). A bias towards the filter side will make the gate effect weaker (full filter setting means the LPG will behave like a normal filter, no VCA effect will be applied).

Note: a *gate:filter* bias towards filter means that the gate will be permanently open to some degree. You may want to use the VCA in the output section to silence the output when no note is playing!

Control and frequency
Any signal you send into the control CV jack will let you control how far the LGP will 'open'. Typically you will use an envelope or LFO for this. The Slope's output is normalized to this input.

The frequency knob acts as an attenuator for the control signal. It will determine how far the filter will open itself when the incoming control signal reaches its maximum value. At maximum setting virtually any audible frequency will pass through the filter, at minimum setting barely anything will escape the filter. Set the gate:filter bias fully to gate mode to completely 'disable' the filter.

BPF mode will switch the filter from a lowpass filter into a bandpass filter. In bandpass mode the filter will only let through a narrow frequency band around the cutoff frequency. This sounds distinctly different from lowpass mode, where all frequencies below the cutoff frequency are allowed to pass through.

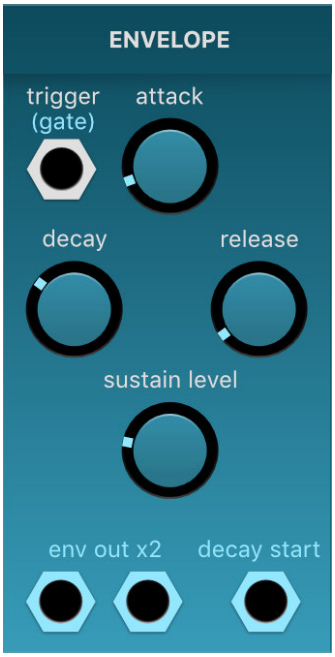
Note: the filter design is modelled after the 12dB/Oct Steiner-Parker filter. A minimal amount of resonance is added to keep the signal juicy, but without losing the typical character of an LPG.

Noise S&H
This little unit generates random values at timed intervals. During these intervals the same value is held, (hence it's called a *Sample and Hold* (S&H) module). If you speed up the rates, new values will be generated faster. At maximum rate, you will get white noise. You can use this module as a source for random voltages, or as a noise audio signal.

The level CV jack lets you control the volume (or maximum value) of the outgoing noise/random signal. You can e.g. use this to create puffs of noise (by patching an envelope into it), or to scale the outgoing random values (by patching a constant voltage from the Utility unit into it).

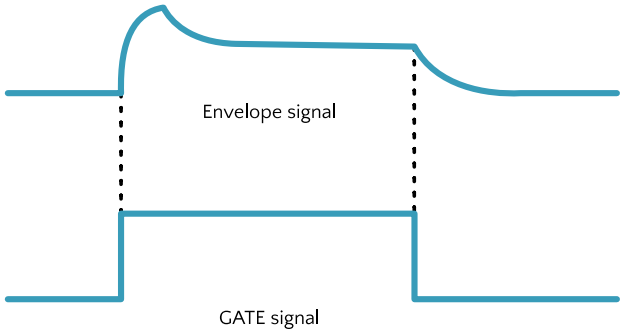
Note: at low rates the S&H unit functions as a random voltage generator. At high rates it becomes a white noise source.

5. Envelope Module



The Envelope module is a fairly straightforward envelope generator (EG). It features the typical 4 ADSR stages: attack, decay, sustain and release.

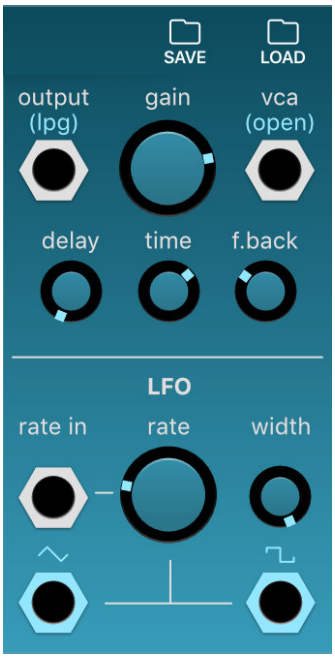
The EG can be actuated using a gate signal patched into the trigger jack (by default this is normalized to the MIDI Gate output). When the envelope reaches the sustain stage, the sound will remain sustaining until the gate signal closes (usually when the note off is received via MIDI).



The two Envelope outputs are identical so you can apply the same envelope to two different destinations.

The *decay start* output sends out a trigger pulse as soon as the envelope has left the attack stage and is about to enter the decay stage.

6. Amplifier + LFO



LFO
The Low Frequency Oscillator is an extremely useful utility for bringing your sounds to life.

The settings of the knob and the CV input are summed to calculate the desired frequency. Both have a range of 0-130.82Hz (C3). So you can go from glacial speeds to high into audio rate territory. This lets you create slowly evolving sounds, but also makes the LFO a useful input for the FM section or for ring modulation.

Setting both the knob and the CV input to 0Hz to allows you to temporarily stop the LFO.

Both LFO outputs can be used simultaneously.

LFO Width
The amplitude of the LFO can be set using the width knob. This is essentially a sort of attenuator letting you scale the LFO output to a maximum value. E.g. when set to 50%, the LFO will cycle back and forth between 0% and 50%.

Amplifier & Delay section
This is the output module of Ripplemaker's synth engine, followed by the 'external' Delay effect. What ever you want to ouput needs to be plugged into the *output* jack. This jack is normalized to the Lowpass Gate by default. If you want to bypass the Lowpass Gate, or chain something else in between, you'll need to manually patch it into this jack.

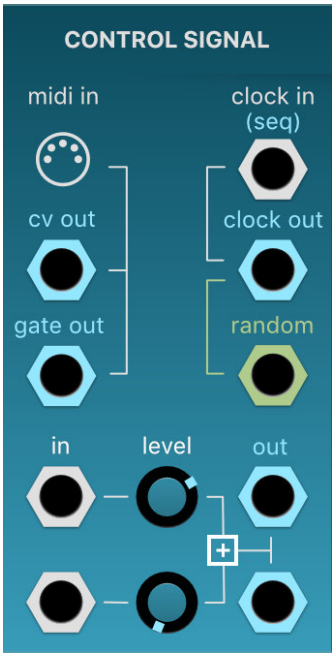
The VCA can be used to dynamically modulate the amplifier section. E.g. by patching an Envelope output into the VCA jack you can apply the EG to the sound. By default the VCA is wide open to make it easy to create drones. For example, if you patch the output of the Oscillator directly into the output jack, you will hear the oscillator drone on continuously because the VCA is fully open.

The Gain knob lets you alter the overall signal level of the output.

Delay Effect
The delay (echo) effect lets you adjust the effect strength (delay), the delay time and the delay feedback. The effect is linked to the sequencer/host's tempo and automatically changes along with tempo changes in the sequencer.

The delay time is set in 16th steps. Values > 50% change the delay time in full steps. Time values <50% let you set delay times in fractions of a 16th step (for creating oldschool metallic faux-reverb/feedback effects).

7. Control Signal + Utility module



Control section
I've saved this module for last, as it can be one of the most daunting parts of the Ripplemaker synth. The control section basically manages the input we receive from the outside world, such as MIDI, and translates it into signals we can use in our other modules.

MIDI Input
We typically receive MIDI input from the sequencer or from the AU Host which hosts our plugin. Since our plugin is fully based on analog models, we can't exactly use digital MIDI commands without converting them into something more 'analog'. Hence we translate incoming MIDI note messages into two distinct signals: CV and Gate.

CV: we use this to communicate the desired sound frequency
Gate: we use this to indicate whether sound is currently on (high signal) or off (no signal)

For instance, we patch CV into the oscillator to tell it at what frequency it should generate sound. Gate is patched into the envelope and slope sections to trigger them when we receive a MIDI note on message (and we release the envelope when we receive MIDI note off messages).

Clock section

When the synth is connected to a sequencer or AU plugin host, it will receive a clock signal. Every quarter note it will generate a pulse which will then be sent out again through the *clock out* jack. This pulse, like any other pulse can be used as a trigger source for things like the envelope or slope.

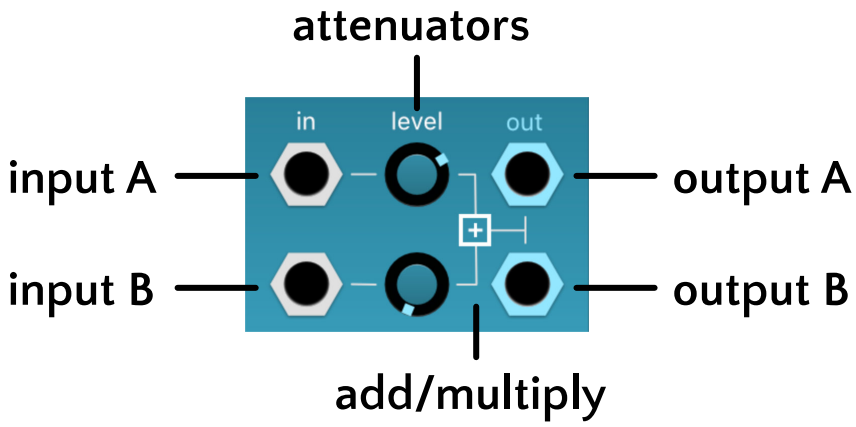
Every clock pulse, a new random value is generated and sent to the *random* output jack. That's why this source is referred to as a 'clocked random source'. This is extremely useful, since random values are changed in a rhythmically meaningful way and can be easily aligned with the start of a new note.

The *clock in* source can be overridden with anything that generates a pulse-like signal (like the *End Of Cycle* pulse or the square LFO wave). Whenever a pulse is received, it is sent out again via the *clock out* port, and a new random value is generated.

Note: by patching the LFO into the *clock in* jack, you can generate random values and clock pulses at the rate of the LFO. This lets you override the tempo of the sequencer with your own rate.

Utility Section

The utility section may note look like much, but it's one of the most useful modules of Ripplemaker. This unassuming cluster of controls is a true Swiss Army Knife for modular synthesis. It consists of two inputs, two outputs, two linear attenuators and a mathematical mixer.



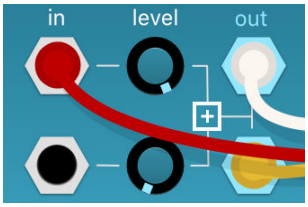
The Utility module takes both inputs, attenuates them according to their level settings, then blends both signals (either through summing or through multiplication) and sends the signal out of both its outputs.

Note: when nothing is patched into an input jack, its attenuator generates a constant voltage. This voltage will then be added or multiplied with the other signal (this can also be a constant voltage).

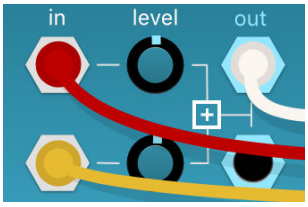
The inputs take CV or an audio signal, or a combination of both. Tapping the “+” button toggles between summing and multiplication mode.

> **Note:** the attenuator knobs are linear; so a knob setting of 50% means any incoming signal is scaled by half and a setting of 100% means the signal is passed through unchanged.

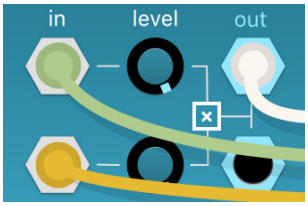
Here are some of the ways you can use the Utility Modules:



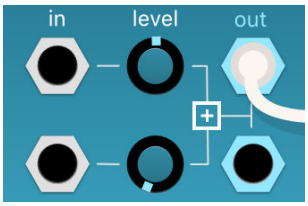
Signal Splitter (duplication)
In this example the incoming red signal is copied to both outputs unchanged. This is useful if you want to make copies of an LFO or your random source for two destinations. Make sure the second attenuator is set to 0 or its constant voltage will be added to the red signal!



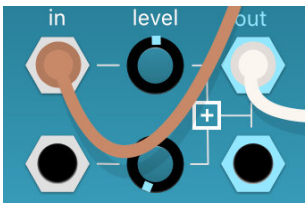
Summing
Here the red signal and the yellow signal are added together and sent out to both outputs. Note that both attenuators are set to 50% to ensure that the resulting signal will not be clipped. The incoming signals could be sound or CV or both.



Multiplication
By multiplying two CV signals (ENV, LFO, Slope, random) you can create complex modulation sources. Multiplying a CV signal with an audio source can be used for e.g. applying an EG to a sound source. Multiplying two audio sources is a clever trick for creating a *ringmod* timbre.



Constant Voltage
If nothing is plugged into the input jacks, the attenuators generate a constant voltage. In this example a voltage of 50% is sent out of the outputs.

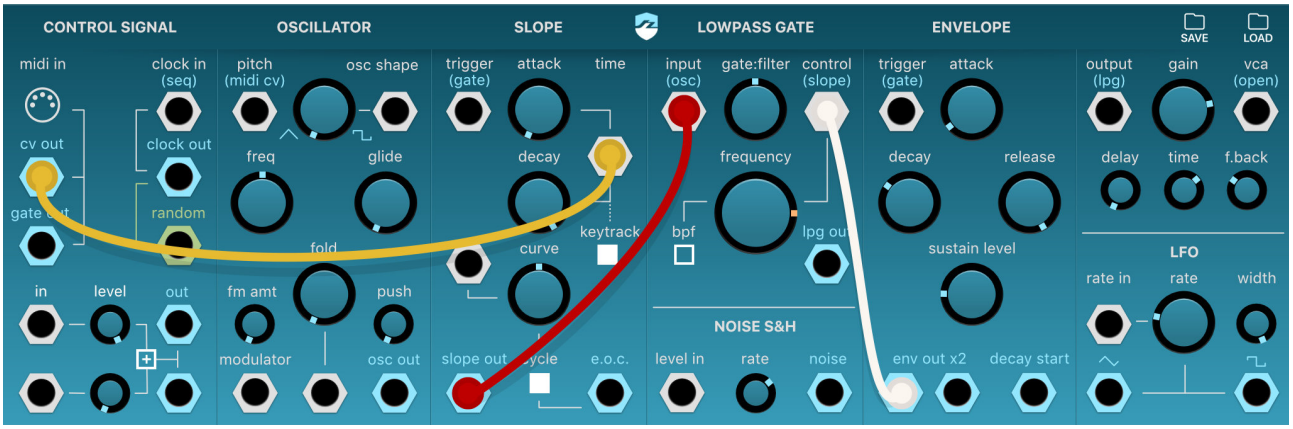


Scaling & fading
The attenuators help scaling an incoming signal down. E.g. if you don’t want your random values to exceed 50% or if you want to fade an audio signal down to half its volume you could use the setting in this example. Also if you want values not to start at 0 you can use the other attenuator to add a base-value to the scaled signal.

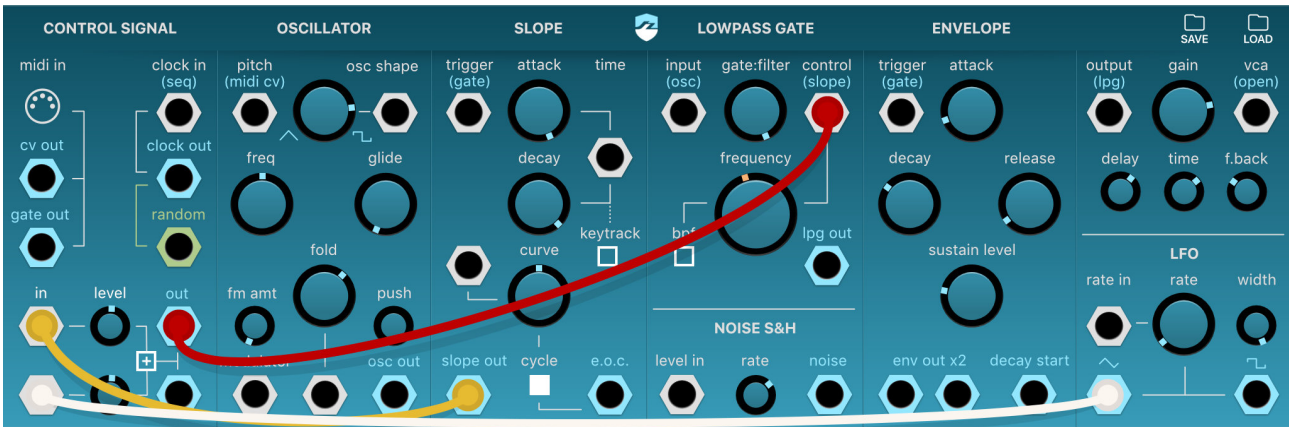
8. Advanced tips & tricks

The following example patches show some advanced techniques to get interesting sonic results. Hopefully these will inspire you to go off the beaten path and explore Ripplemaker beyond the obvious use cases.

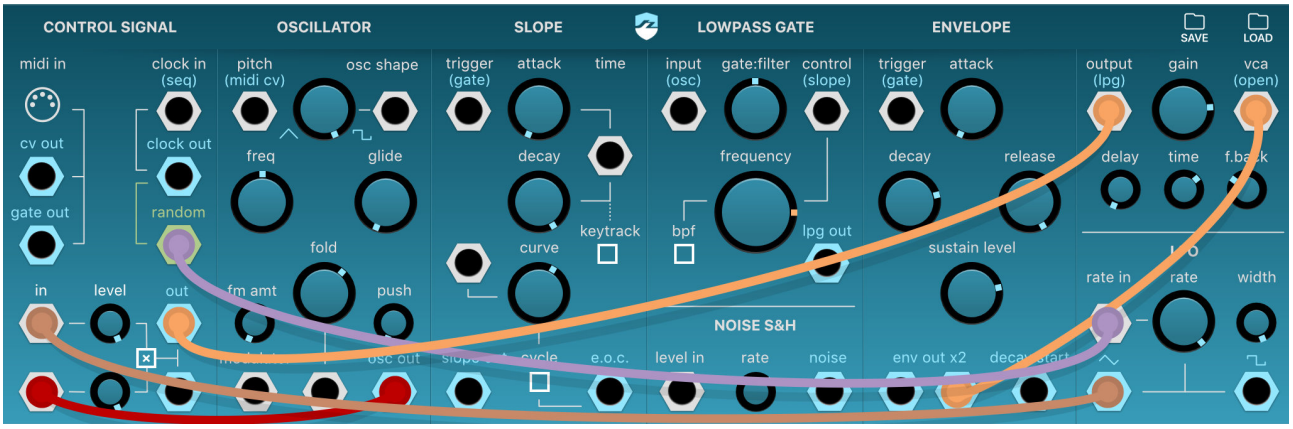
Sawtooth oscillator
Ripplemaker's primary oscillator is triangle-core design and does not offer a sawtooth. However, you can transform the Slope section into a fat sawtooth osc. By putting the module into oscillator mode and setting the slope to minimum attack and maximum decay (or vice versa for an upward ramp) you get all the sawtooth waves you need. The curve knob lets you tune the osc, but you can still modulate the saw's slope curve via the curve CV input jack.



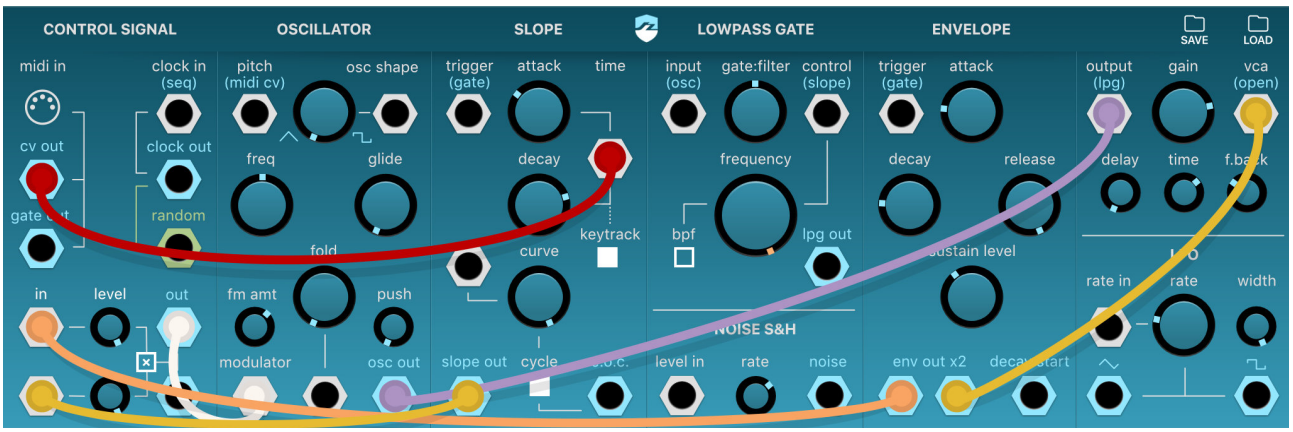
Complex LFOs
The Utility section lets us take two modulation sources as input and combine them into one - much more complex - modulator. In the example below, we take two LFOs and sum them to achieve a less predictable LFO signal. Experiment with different modulation sources and summing or multiplying them.



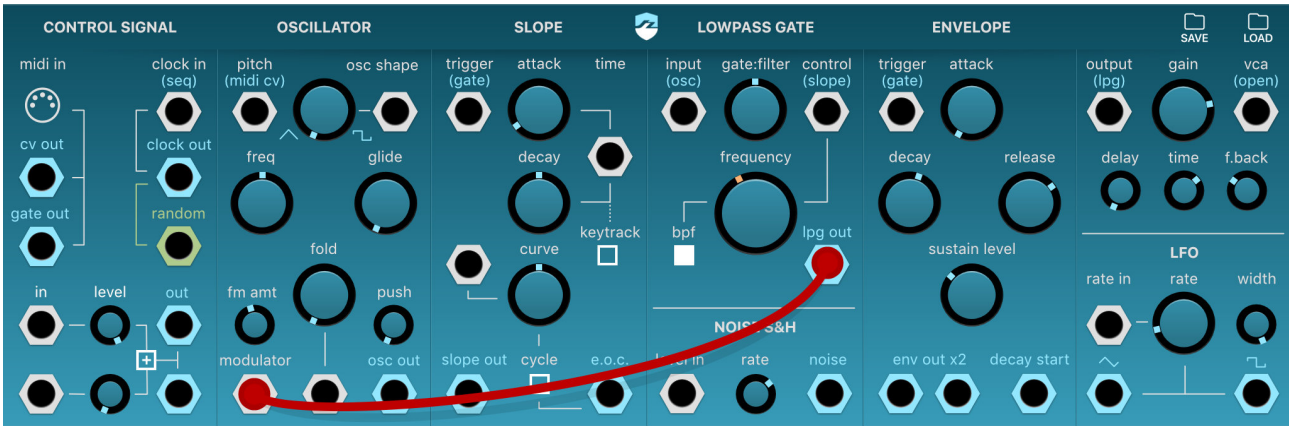
Ringmodulator
A ringmodulator multiplies two audio signals and spits out the sum and difference of the incoming frequencies. So simply multiplying two audio streams in the Utility section lets you easily generate complex inharmonic frequencies, e.g. for use in bell sounds.



FM with dynamic modulator
Frequency modulation is even nicer when an envelope is applied to the modulator. We can do this in Ripplemaker by sending the slope oscillator and an envelope signal into the Utility section, multiplying them and then sending the resulting signal back into the FM modulator input.

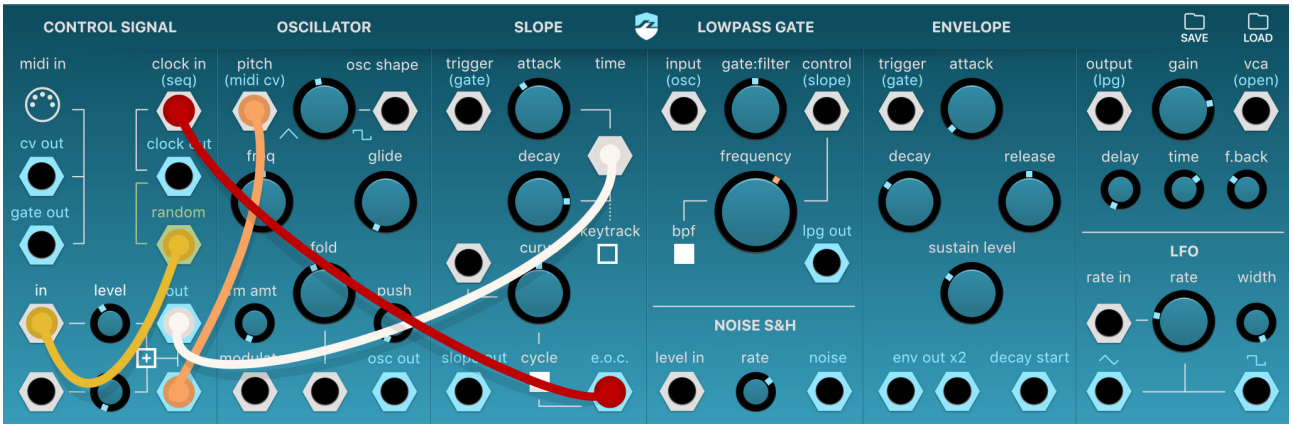


Instant squelch
Set the Lowpass Gate to bandpass mode, feed the output back into the oscillator's FM input. Aciiiid!

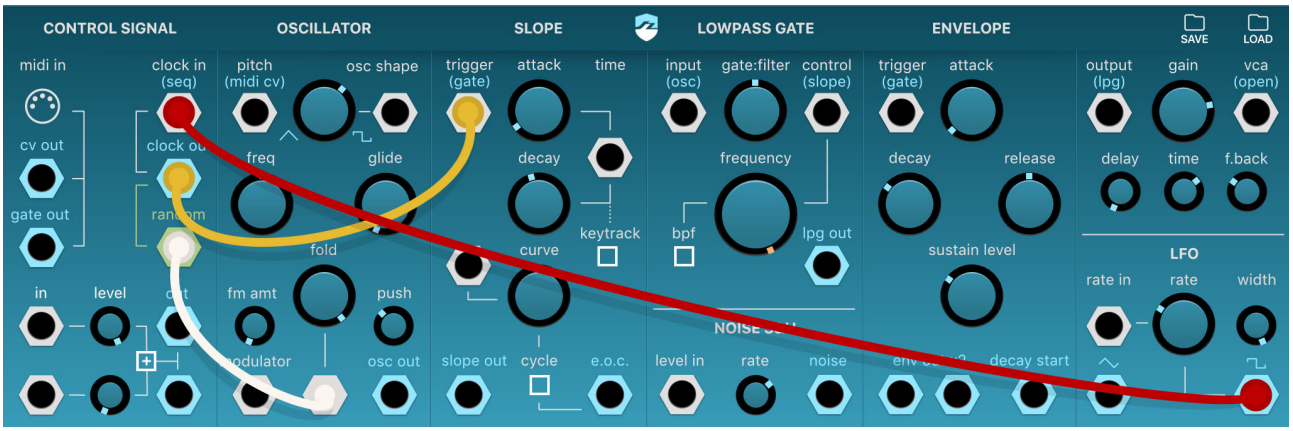


Base Recipe for a Krell Patch
Krell Patches are a typical style of generative patches often made on modular synths. Some characteristics are that they re-trigger themself and every tone has a different duration. Shorter notes have a higher pitch than long notes. The setup below is a good starting point for creating your own Krell variations.

The Slope is set to cycle mode and each end-of-cycle a pulse is sent into the clock. This will trigger a new random value from the clocked random source. This value is scaled down to avoid extreme values. The resulting random signal is then sent to both the pitch and the Slope time.



Using the LFO as a Clock Source
The nice thing about control voltages is that anything can be used a trigger source and you can do many interesting things without the need for a sequencer per se. In this example we feed the square LFO into the clock input. The clock output pulse is subsequently used to trigger the Slope in the rhythm of the LFO. The clocked random source is fed into the fold input which makes each note sound different.



9. Automation Controllers

AU Parameters
Every knob and checkbox in the Ripplemaker plugin can be controlled and automated using AU parameters from the plugin's host application. AU Parameters are convenient because they have a high resolution (unlike MIDI CC, which only supports integer values from 0-127) and they have descriptive names (e.g. "FM Amount") rather than just an arbitrary number.

Ripplemaker even supports recording of knob-movements in its GUI when hosted in an AU compatible DAW. Unfortunately I am not aware of any current DAW app that has implemented this feature, but for future convenience: it is there should any of the DAW makers decide to support it.

MIDI CC
Until all AU hosts and DAWs have full support for AU Parameters, MIDI CC is also implemented as a fallback. As mentioned before, MIDI CC is suboptimal and is only offered as a fallback feature to give DAW makers time to fully implement AU Parameters. This is also the reason no MIDI learn is available; please contact your favorite AU host developers and request AU Parameter support. The world will be a better place.

The following controller numbers are for MIDI CC. The respective Audio Unit parameters use their corresponding names.

Utility Section

	Attenuator 1	Attenuator 2	Sum Mode			
CC#	10	11	12			

Oscillator Section

	Tune Freq	Osc Shape	Osc Glide	FM Amount	Fold Amount	Push Amount
CC#	13	17	18	19	20	21

Slope Section

	Attack	Decay	Curve	Keytrack	Cycle Mode	
CC#	22	23	24	26	25	

Lowpass Gate + Noise S&H Section

	Frequency	Gate:Filter	BPF Mode	S&H Rate		
CC#	14	15	16	33		

Envelope Section

	Attack	Decay	Sustain	Release		
CC#	27	28	29	30		

Amp + LFO Section

	Gain	Delay Amount	Delay Time	Delay Fdbk	LFO Rate	LFO Width
CC#	7	102	101	103	31	32

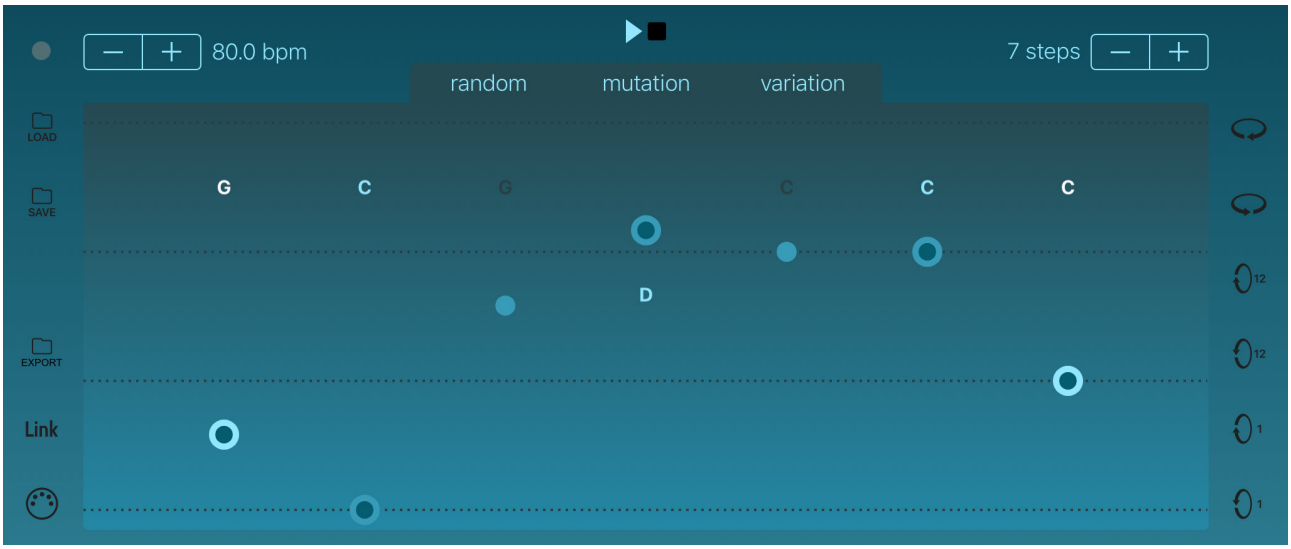
Note: MIDI CC uses value ranges from 0-127. All parameter values are mapped to this range. 50% = CC value 64; 100% = CC value 127.

Example: to change the Lowpass Gate frequency to maximum, send a MIDI CC command 14 with a value of 127. To set the LPG gate:filter balance to 50:50, send MIDI CC command 15 with a value 64.

10. The sequencer section

When using Ripplemaker as a standalone app, it lets you program synth patterns using its own built-in sequencer. Under the surface the sequencer is actually a normal MIDI sequencer sending bog-standard midi commands to the synth module.

Once you have designed (or randomly generated) a nice pattern, you can export your pattern as a standard MID file and send it directly into your DAW of choice to build a full track out of it (taking advantage of the Audio Unit Plugin format of Ripplemaker). The exported MIDI file should sound identical to the pattern played back in Ripplemaker's own sequencer.



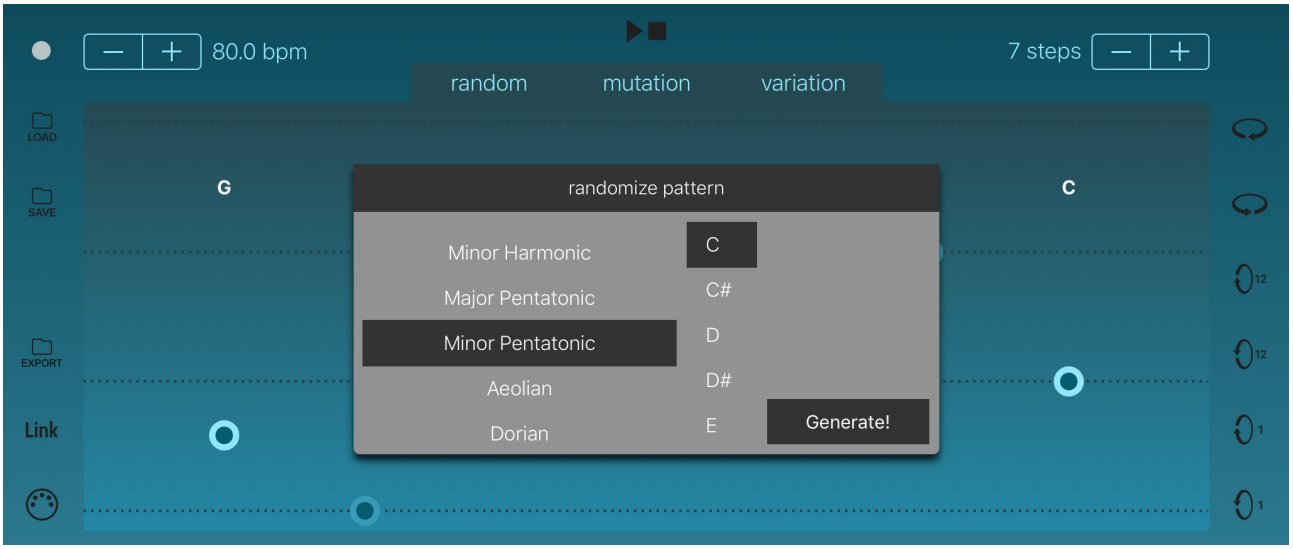
The piano roll
Well, it's not exactly a piano roll. But it was inspired by piano rolls. It covers the useful 3-octave range of C-2 to C-5. The dotted lines indicate where the C notes in their respective octaves are.

Fat blue circles are notes (the light variety are accented notes), dots are rests. Tap on notes to enable/ disable them.

Drag a note to change its pitch. While you're dragging a note, a subtle piano roll appears in the background showing the exact positions of each note.

Steps
Using the 'steps' control you can set the length of your pattern. The steps value is in 16th notes. You can set values of 1-16, 24, 32, 48 and 64 steps. When using values > 16 Ripplemaker assumes you're using more than one bar and additional vertical lines in the sequencer will be shown to indicate bar separators. Slide the screen horizontally to pan the sequencer view if needed.

Random
This button randomly generates a pattern for you. Most patterns will be 8 or 16 steps, but sometimes you'll get a more unusual number of steps. You can indicate which scale and rootnote you want the generated pattern to conform to. Choose 'chromatic' if you don't care and want a fully random pattern.



Mutation
This is the realtime mutation engine. While you're playing patterns, these settings determine the probability that the sequencer will actually do something else. These mutations are a realtime playback feature. They are not permanent and will not be saved with the pattern. When exporting MID or WAV files, the current mutation settings will be used for the exported output though.

Variation
Takes the current pattern and makes a variation out of it. It will keep the existing notes and rests and the current pattern length, but will randomize the octaves, accents and slides.

Play/Stop
This button starts and stops the playback of the pattern. If Ableton Link is enabled there may be a brief count-in before the pattern starts playing to ensure your pattern will be in sync with the other Ableton Link enabled apps on the network.

Pattern rotation features
The buttons on the righthand side of the sequencer let you rotate the current pattern to the left, to the right, up and down (either one note or per octave). Notes that 'fall off the grid' will appear back on the other side.

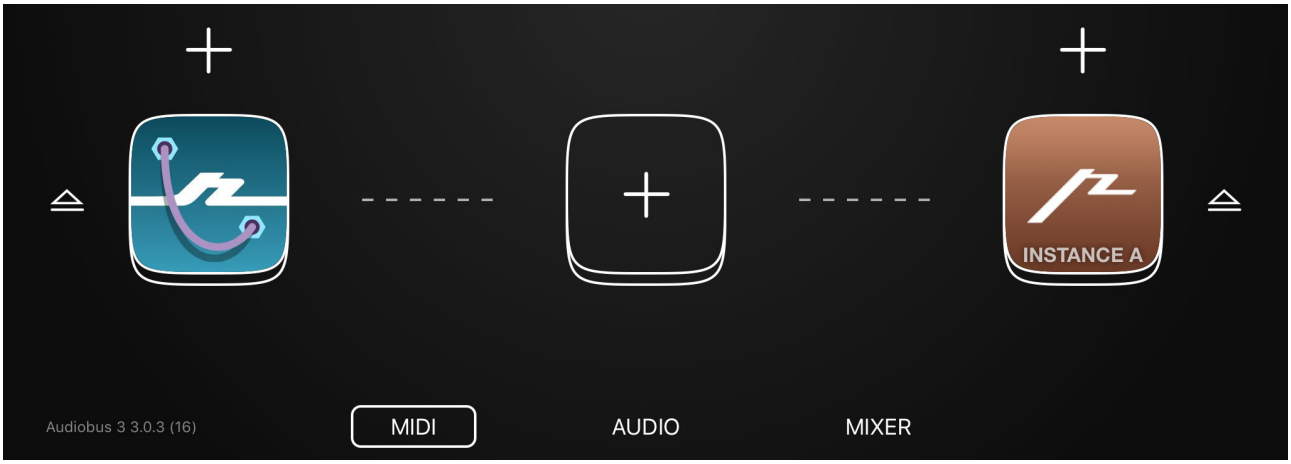
Link
This lets you control the Ableton Link feature. You can use this to sync the app with other Ableton Link compatible apps and devices which are on the same device or on the same network as Ripplemaker. Changing the tempo will then change the tempo in all linked apps and conversely Ripplemaker will also respond to tempo changes initiated by other apps.

> **Note:** when a Link session is active, Ripplemaker will respect its tempo rules. Opening a pattern with a different tempo will not change the active Link session's tempo.

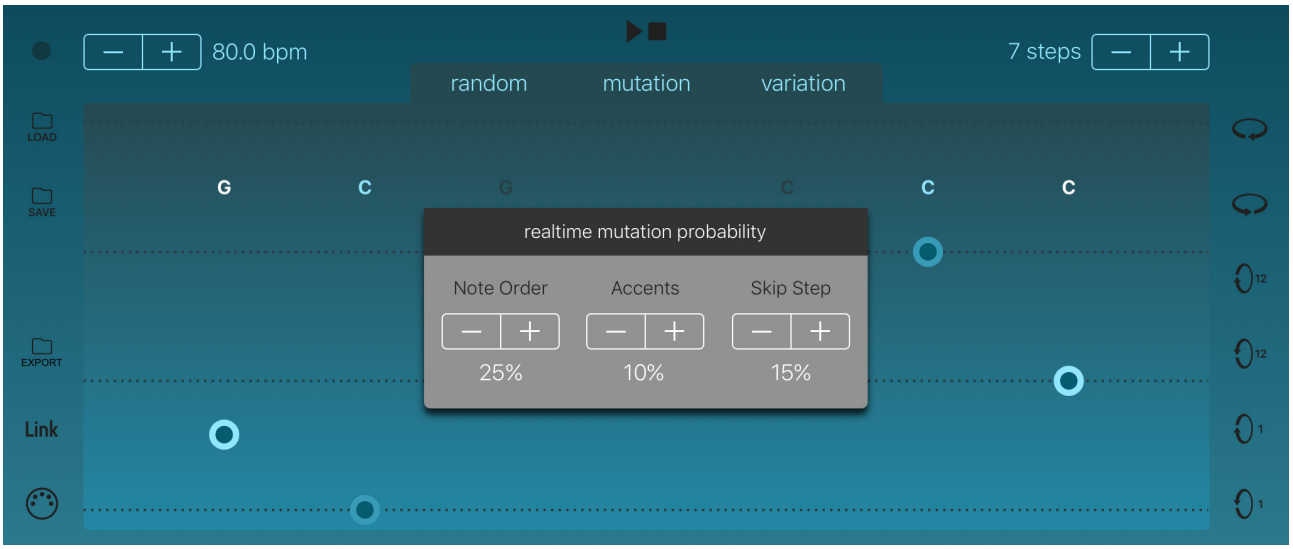
MIDI settings & Audiobus 3 MIDI
This lets you change basic settings for midi input (channel) and select Bluetooth MIDI devices. Ripplemaker is fully compatible with CoreMIDI, Virtual MIDI and BTLE MIDI on iOS and will respond to midi input while running in background mode.

Note: Audiobus 3 MIDI IN/OUT is also supported. No settings need to be made for it; they are enabled automatically when Ripplemaker is loaded into an AB3 MIDI slot.

When using Ripplemaker in an Audiobus 3 MIDI slot you can use the built-in sequencer to drive other Audiobus 3 compatible apps and Audio Unit Instruments. In this case Ripplemaker's internal synth engine is disabled and only MIDI notes are transmitted - not audio.



11. Using realtime mutations



The realtime mutation engine takes your pattern and introduces semi-predictable variations during playback. Four different parameters can be individually processed using the engine.

For each parameter, the selected percentage represents the probability that a variation will happen. So in the example shown above, for each note in the pattern (rests are ignored) there is:

Note Order: 25% chance that the note will be replaced by a random other note from the same pattern
Accents: 10% chance that a regular note will be played back as an accented note (and vice versa
Skip Step: 15% chance that a step in the pattern will be skipped, letting the pattern go “out of sync”

The higher the probability percentage, the higher the chance that the variation will happen. To disable a variation, set it to 0%

Probabilities can be changed (or disabled) in real-time using the following MIDI CC commands; the values from 0% to 100% are then mapped to the MIDI value range 0-127:

Setting	**CC code**	**Example**
Note Order	CC#40	CC#40 Value 0 disables the probability that a different note will trigger
Accents	CC#41	CC#41 Value 64 will set the probability of an accent variation to 50%
Skip Step	CC#43	CC#43 Value 32 will set the chance of skipping a step to 25%

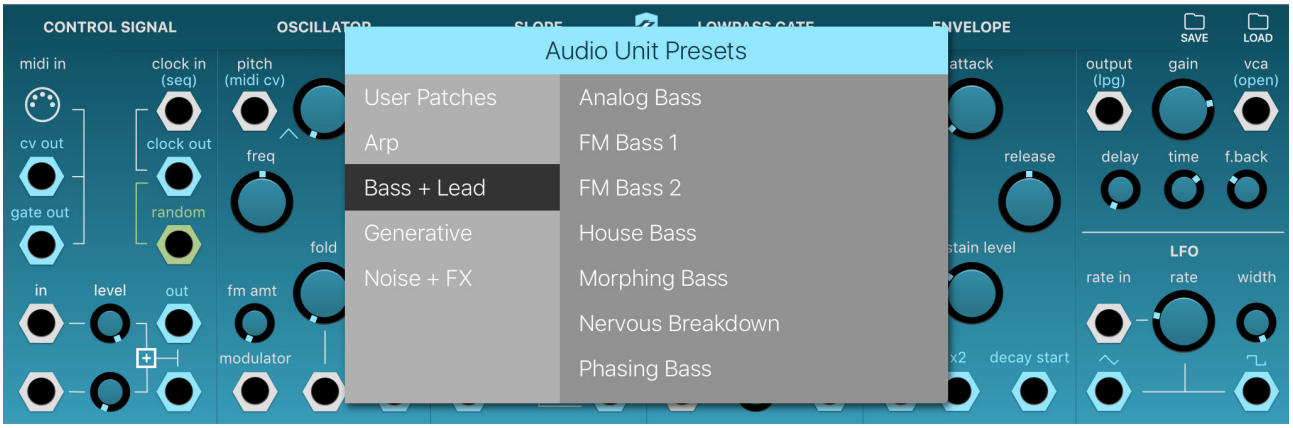
Mutation settings are a realtime playback feature only and do not permanently change your pattern, but the mutation settings are saved with the current pattern.

Transpose
Use MIDI CC#44 for realtime transposition of the current pattern. The range is -12 to +12 semitones.

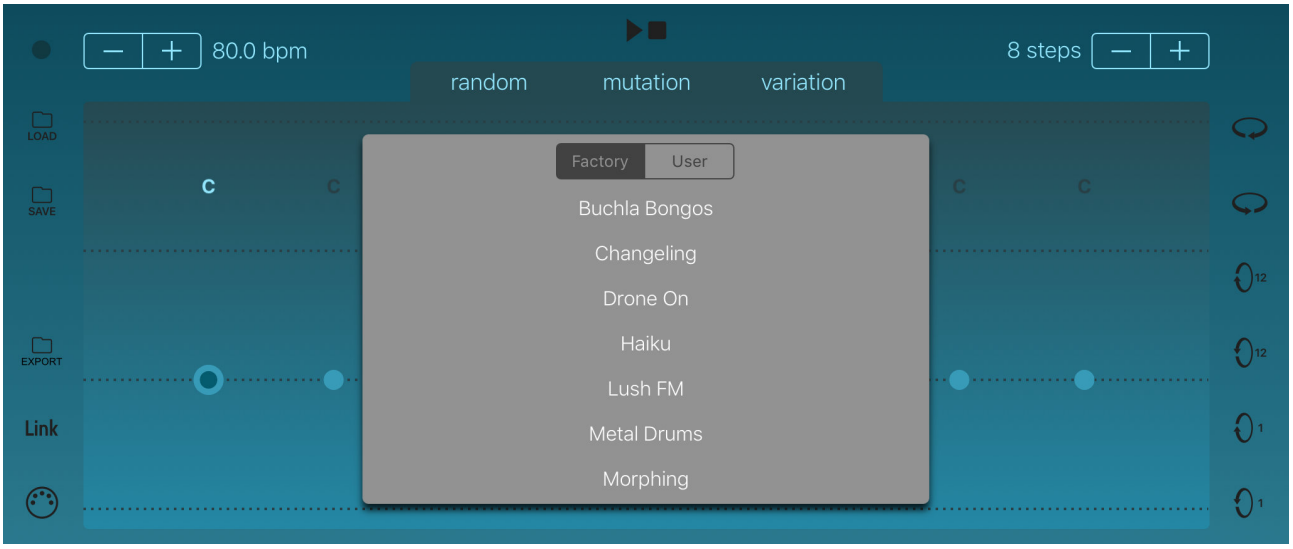
- There is no visual feedback on the screen of the current transposition. Use with care!
- Set MIDI CC#44 to value 64 (centre value) to reset the transpose value to 0
- Incoming MIDI notes are not transposed, only pattern notes
- Transpose is not applied to exported MIDI files, but will be applied to exported WAVs

12. Saving and Exporting

AU Presets
Synth settings can be saved and loaded from inside the Audio Unit plugin. This ensures that all your favorite presets and homemade patches are always available, across different AU hosts and the standalone mode of Ripplemaker. To access the AU Presets, tap the SAVE and LOAD buttons in the top-right corner of the plugin.

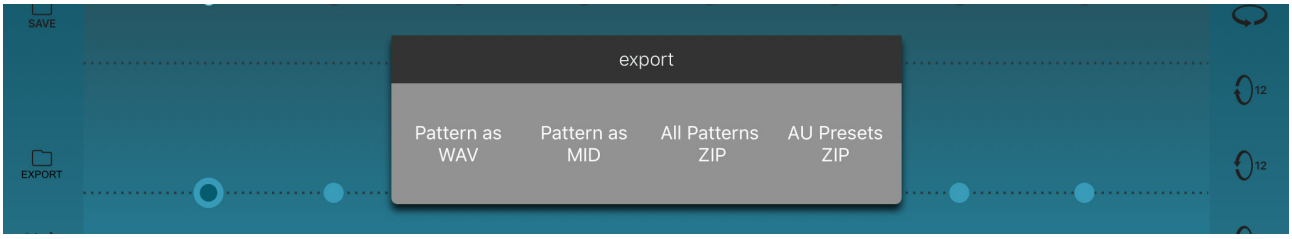


Patterns
A Ripplemaker pattern contains the sequencer note data, the current synth settings, the tempo and the current mutation settings. Patterns can be saved and loaded from the standalone sequencer version of Ripplemaker. They can be accessed from the SAVE and LOAD buttons in the left hand margin of the sequencer.



Note: you can delete user presets and patterns from the list by swiping-to-the-left and then confirming you want to delete them. Factory patterns/presets can't be deleted.

Exporting
Several things can be exported (and imported) from the standalone version of Ripplemaker. Tap the EXPORT button to bring up the export options window…



Pattern as WAV - Exports the 4 bars of the current sequencer pattern & synth settings as a WAVE file

Pattern as MID - Exports a standard MIDI file with the current sequencer pattern for use in your DAW

All Patterns ZIP - Packs all your user patterns into a Zip file and sends it to whereever you like

AU Presets ZIP - Packs all your homemade AU Presets and exports them to your preferred destination

Importing
Patterns and AU presets can be reimported as Zip files. To make sure they are valid and end up in the right place there is a special rule for how to name your Zip files:

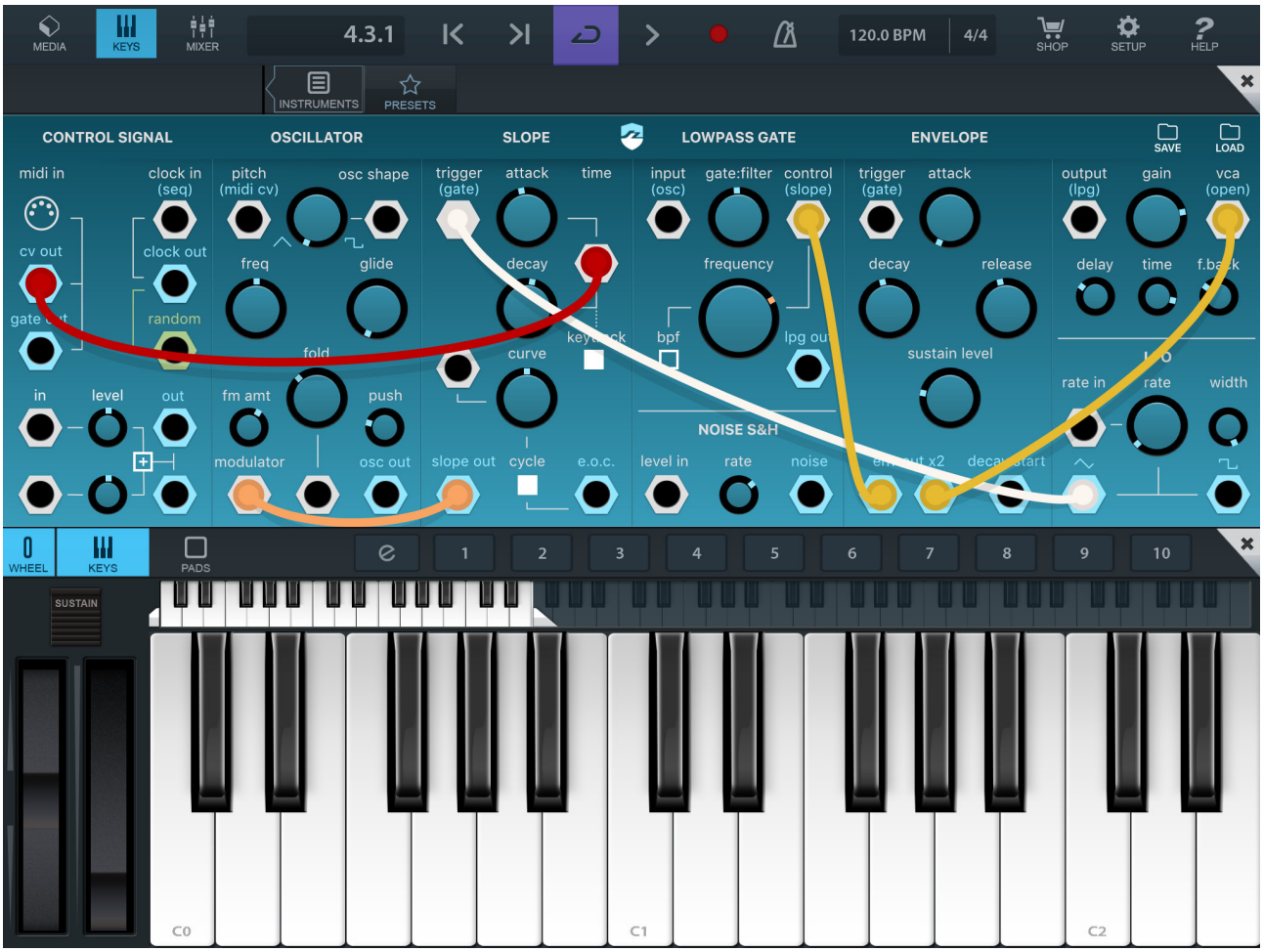
Pattern Zip file: [yourname]_patterns.zip e.g. *ripplemaker_patterns.zip*
AU Preset Zip file: [yourname]_presets.zip e.g. *ripplemaker_presets.zip*

Other filenames are ignored.

13. Using Ripplemaker in plugin mode

When using Ripplemaker as plugin inside third party hosts and DAWs it will behave as any regular MIDI instrument. You can run as many simultaneous instances as your device can handle.

The Delay section and the clock input will automatically synchronize with the AU host's tempo settings.



This document, Ripplemaker and the Ruismaker logo © Bram Bos 2017
www.ruismaker.com