META

Documentation

1. Introduction

META crossfades between its analog inputs according to the level of an internal contour generator. The contour generator fades from the A input to B input along an "attack slope" and then from B to A along a "release slope". A set of programming parameters allow the crossfade control signal to take on a wide span of behaviors ranging from a drum voice to a complex LFO.

This configuration allows for two equivalent approaches to patching with the module:

You can use the contour generator to set the level of the crossfader inputs.

You can use the crossfader inputs to set the level of the contour generator.

You can also do a little bit of both. If you want all the details, here is how it works. You can also find some patch examples here.

The crossfader inputs can be sampled for the duration of one or both slopes, creating resampling effects and isolating the crossfade contour from a moving input.

Control inputs connect the module to the rest of your system, allowing you to impose timing events on the contour generator and automate its frequency and shape. A trio of additional outputs offer signal sources to complement the main output.

2. Controls and IO

If you haven't yet, take a glance at this introduction of Via's controls, IO, and user interface.

Knobs

TIME 1 (knob 1) and **TIME 2** (knob 2) are two generalized time controls (or frequency, or speed; take your pick of which makes the most sense to you). The specific control response changes to suit the current operating frequency range. Details can be found in the description of the **FREQ** mode.

∿⋀⊿⊓ (knob 3) guides a smooth transition between the contour generator shapes in the currently selected **TABLE**.

CV

T1 (CV 1) and **T2** (CV 2) provide CV for the corresponding **TIME** controls. The CVs always match the behavior of the manual controls.

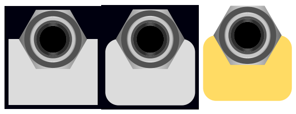
Likewise, ∿⋀⊿⊓ (CV 3) is combined with the manual control to the set shape of the contour generator.

Logic Input

TRIG starts or restarts the contour generator, with specific behavior set by the **TRIG** parameter.

FREEZE halts the progress of the contour generator, fixing its position between the **A** and **B** inputs while the logic level is high.

Outputs

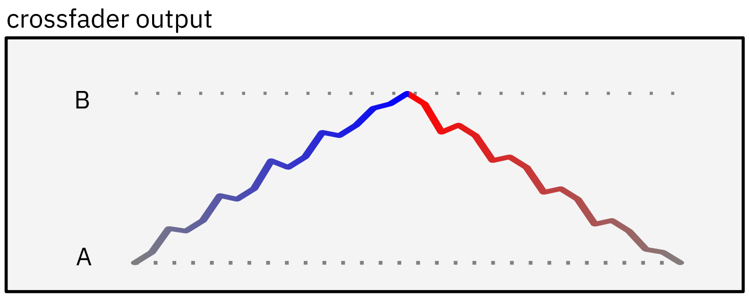


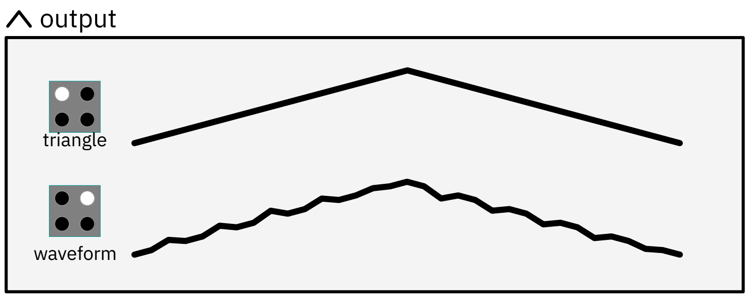
A X B is configured as a crossfader between the analog inputs, controlled by the contour generator.

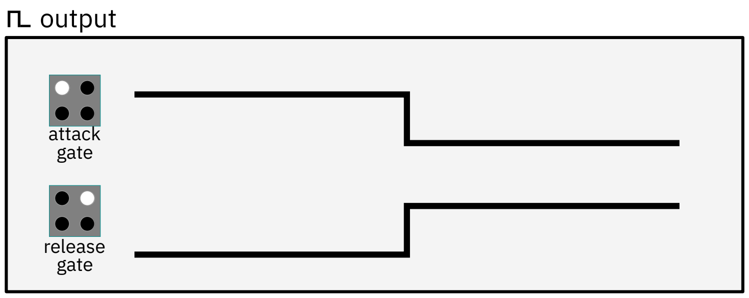
︿ offers an auxiliary signal output (independent of the main crossfader circuit), selected with the ︿ parameter.

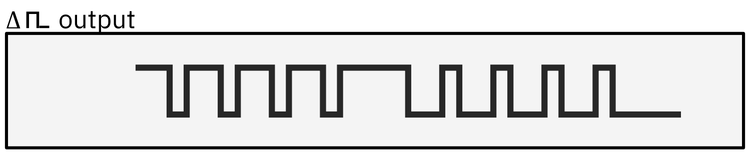
⎍ is low while the contour generator is on a path from **A** to **B** ("attack"), and high while traveling on the path from **B** to **A** ("release"). It can be thought of as an "end of attack" gate. The ⎍ parameter inverts this to a "end of release" gate, so the output is high on the **A** to **B** path and low on the **B** to **A** path.

Δ⎍ on the expander is high while the contour is moving towards **B** and low while moving towards **A**, tracking the changes in direction embedded in the contours themselves.

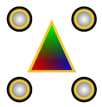








LED Display



When not actively setting a parameter with the buttons, the **white LEDs** connected to the A and B inputs show whether the input is held (LED is on) or tracking (LED is off). The LED connected to the main logic output is on whenever the logic output is high. The bottom right LED shows the state of the ∧ parameter.

The **RGB LED** represents the state of the contour generator. Like the **TIME** controls, its exact function changes to match the **FREQ** parameter.

The **bicolor LED** displays the analog out signal level. Off at 0V, increasing green intensity with positive voltages and increasing red intensity with negative voltages. When a bipolar audio-rate signal is present at the output a mixture of the two colors will be visible.

3. Parameters

FREQ and LOOP

FREQ can be set to (1) *audio,* (2) *envelope,* or (3) *sequence.*

LOOP can be (1) *on* or (2) *off.*

There are 6 possible combinations of **FREQ** and **LOOP**, and those generate the 6 primary behaviors of the contour generator. These parameters have the most dramatic effect on the overall function of the module. The relationship between **FREQ** and the available contour generator shapes is detailed in the **TABLE** section.

The module's programmable controls adapt to changes in the frequency parameter, forgoing the principle of knob-per-function in favor of control schemes that make the most sense for the type of signal being generated.

The *audio* mode uses **T1** for coarse tune and **T2** for fine tune with the **T1 CV** scaled for v/oct response. When **LOOP** is turned off, **T2** governs the release time of the drum voice.

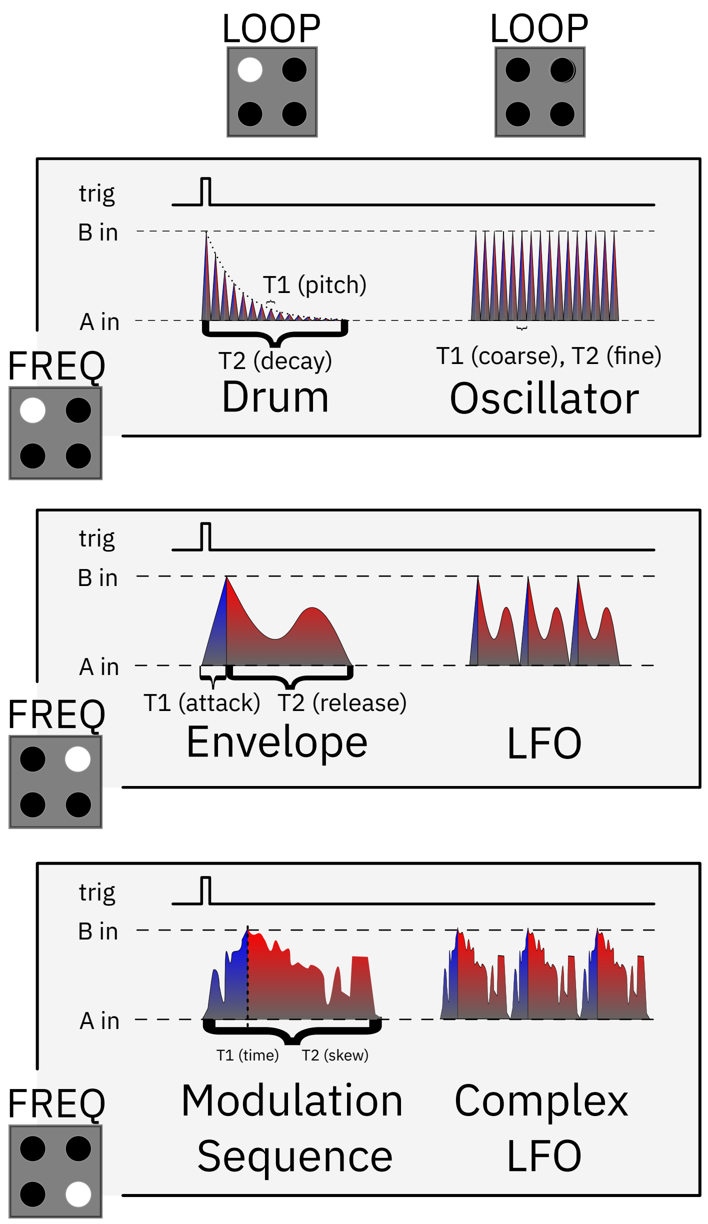
The *envelope* mode uses **T1** for coarse tune and **T2** for fine tune.

The *sequence* mode uses **T1** for cycle time and **T2** for skew.

When **FREQ** is set to *audio* and **LOOP** is set to *off*, the module enters a special "drum mode" that combines elements of both the frequency modes. An internal decay envelope is generated and mapped to the parameters of an audio oscillator according to the **DRUM** mode. **TIME 1** controls the oscillator's frequency

In the *envelope* or *sequence* modes, the **RGB LED** shows the phase of the contour generator (blue for attack, red for release) and the output level (LED gets brighter as crossfader moves closer to B and dimmer as it moves closer to A). In the audio rate, the LED fades from red (slower) to blue (faster). The green level represents the control.

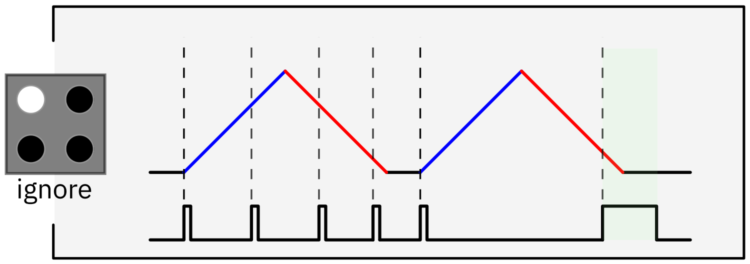
The following figure depicts the contour generator in each of the 6 behaviors along with the function of the TIME 1 and TIME 2 controls.



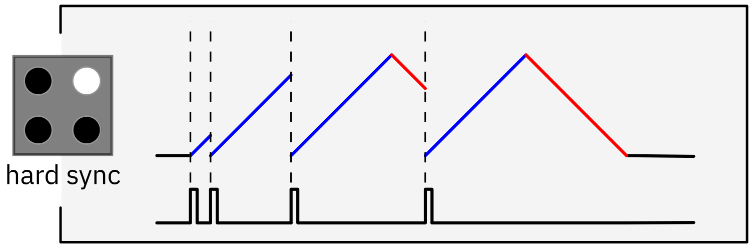
TRIG

The **TRIG** parameter controls how the module reacts to the **TRIG** input while the contour generator is in motion. As such, you will only notice the effect of the parameter when a signal is patched into the main logic input (gate sequences and square waves are ideal). You can start a patch by setting up **FREQ** and **LOOP** appropriately and then tweaking **TRIG** to taste (if using that input).

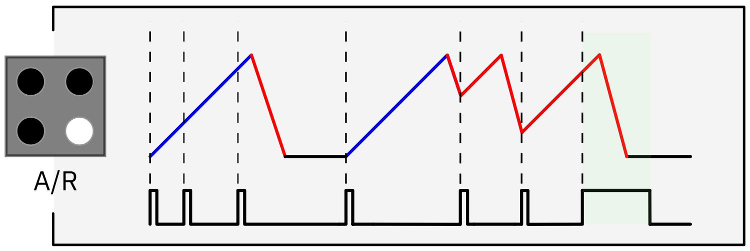
There are 5 possible modes when the module is not in drum mode: (1) *ignore,* (2) *hard sync,* (3) *A/R model,* (4) *gated A/R model,* and (5) *pendulum*. Note that the behavior of mode 5 is slightly different when **LOOP** is enabled vs. disabled.



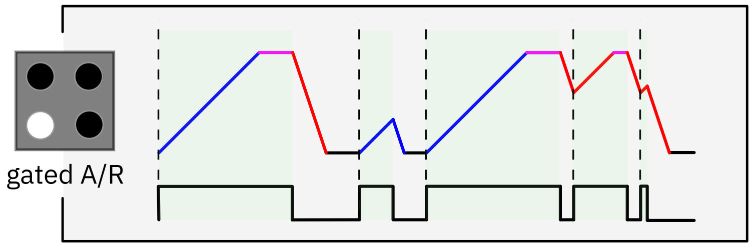
The contour generator does not respond to triggers while active.



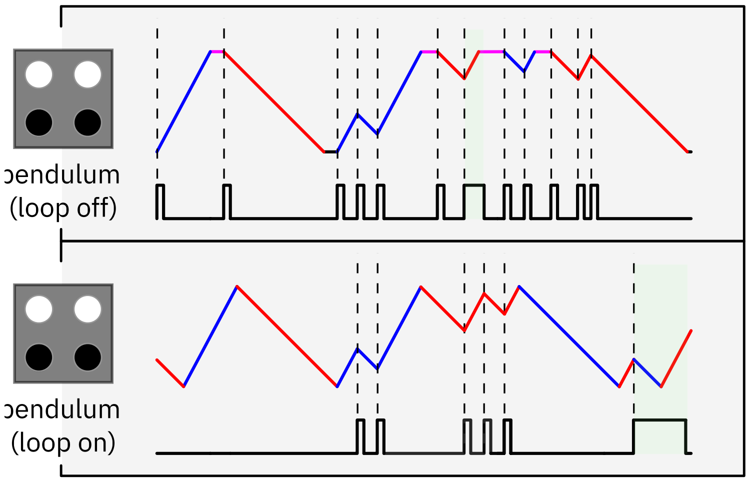
Every trigger resets the contour.



Ignore triggers during attack. Return to B using the attack time parameter if a trigger is received during release. This models a traditional analog A/R envelope.



Similar to A/R, but the contour generator remains at B while the input is high. It releases back to A when the gate goes low, even if the contour is still in the attack stage.



Pendulum changes slightly to suit the loop mode, but as a general rule, a trigger reverses the direction of travel of the contour generator. When loop is off, the contour generator "sticks" to A and B, latching at the ends of the slopes until new trigger restarts travel.

Drum Models

When **FREQ** is set to *audio* and **LOOP** is disabled, **TRIG** sets the response to the **TRIG** input by selecting 1 of 4 possible "drum models":

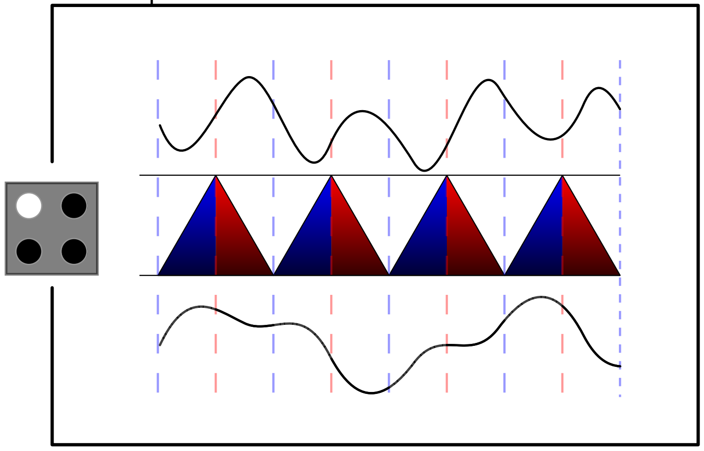
(1) *Kick*: A strong frequency transient and gradual pitch decay
(2) *Tom*: Pronounced pitch envelope
(3) *Pluck*: No pitch decay, but a frequency transient and a timbral decay
(4) *Tone:* No pitch or timbre modulation envelope, just amplitude

Note that in drum mode, the options for the ∧ parameter are augmented with (3) the *drum envelope* and (4) a *noise source*.

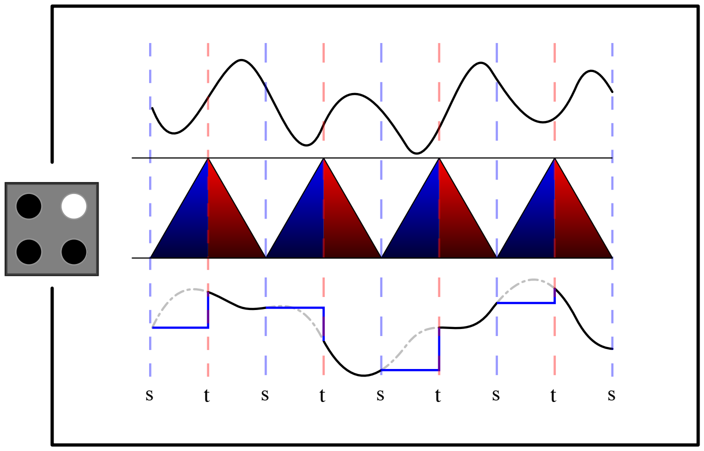
S+H

The **S+H** processing blocks are introduced in detail in the description of the crossfader configuration. Each **sample and hold** can either *track, hold and track,* or *resample*. In every mode but (6), the A input is not sampled when the contour generator is off, assuring that this input is always "passed through". All possible combinations are available, yielding the 6 modes:

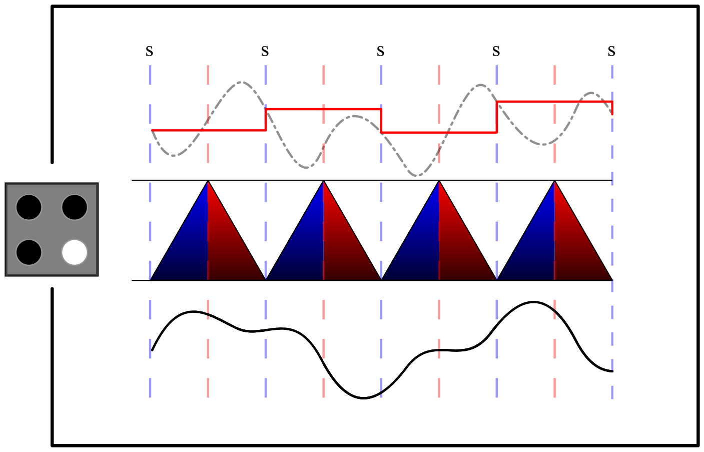
no sample and hold



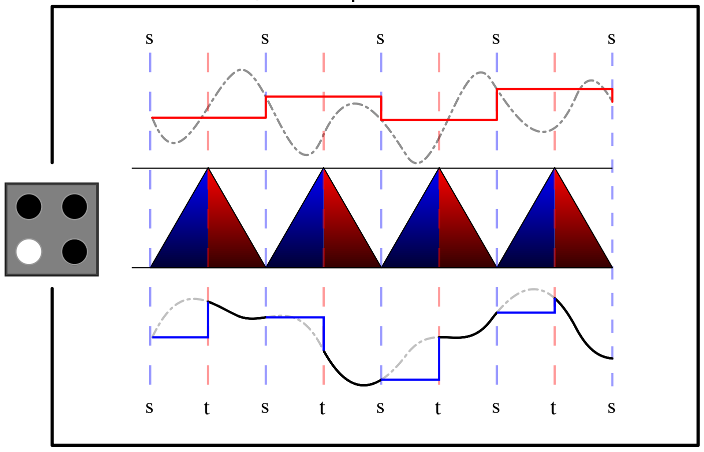
track and hold A



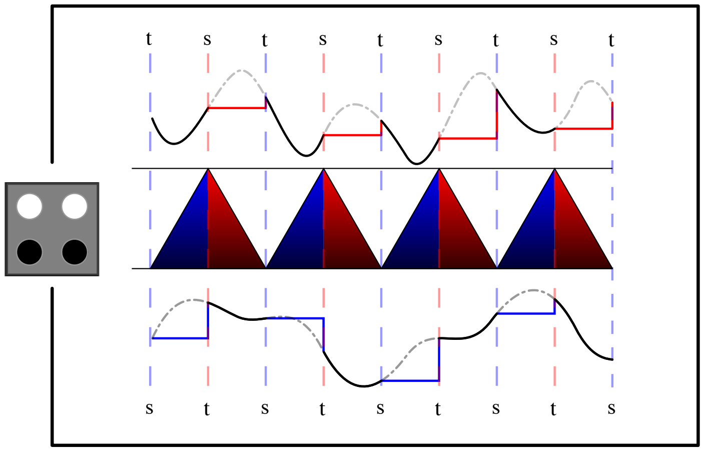
resample B



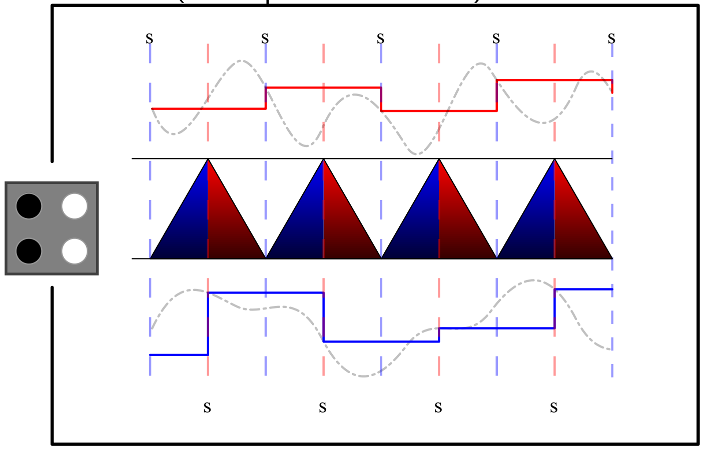
track and hold A, resample B



track and hold both A and B



decimate (resample both A and B)



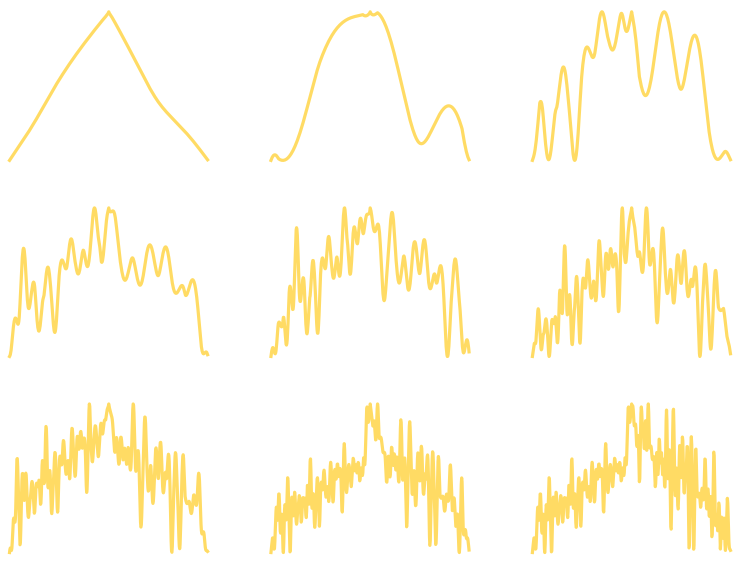
⎍ and ⋀

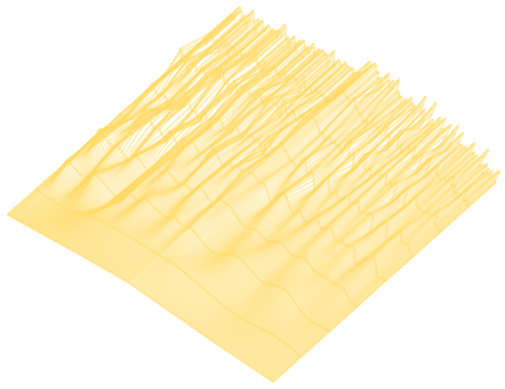
The ⎍ parameter toggles the mode of the ⎍ output between (1) *release gate* and (2) *attack gate.* You can change the mode by holding the **SHIFT** (**TABLE DOWN**) button and tapping the **FREQ** button.

Likewise, holding **SHIFT** (**TABLE DOWN**) and tapping **LOOP** toggles the ⋀ output between (1) *triangle* and (2) *contour*. When the module is in drum mode, there are 4 possible options: (1) *triangle,* (2) *contour,* (3) *drum envelope,* and (4) *noise.*

TABLE

The **TABLE UP** and **TABLE DOWN** arrows page through a set of 8 wavetables. Each **FREQ** mode includes a relevant set, with interesting timbres for audio, smooth slopes for envelope, and musical shapes for sequence. A table is comprised of a set of waveforms. The ∿∿⎍ control morphs smoothly through the set. The following images offers a pair of perspectives on a table object. First you see the individual waveforms, then you see the continuous space of possibilities created through the morph process:





4. Table Directory

Audio

1. **Impulse Train** - Impulse waveforms with increasing overtones resembling a filter sweep without resonance.
2. **Additive Pairs** - Emerging pairs of odd overtones.
3. **Linear Folds** - Mimics the effect of a triangle wave through a wavefolder.
4. **Skip Saw** - Imposes sawtooth ridges on a triangle wave to create a subdued supersaw waveform.
5. **Perlin Noise** - One dimensional slices from a spatial noise algorithm, increasing the frequency of noise per waveform.
6. **Synthesized Vowels*** - Morphs through renders of a modeled vocal tract using an impulse train and a filter bank tuned to the resonances of different vowels.
7. **Sampled Vowels*** - Morphs through waveforms re-synthesized from studio acapellas.
8. **Sampled Train Whistles*** - Morphs throug h waveforms re-synthesized from train whistles.

* This table will create discontinuities in the A input when S&H is activated

Envelope

1. **Asymmetric expo/log** - The attack slope morphs from a bowed-out logarithmic curve to a sharp exponential curve. The release slope morphs between the same shapes in the other direction.
2. **Symmetric expo/log** - Both slopes morph from an exponential curve to a logarithmic curve.
3. **Circular** - Akin to table 2, but elliptic arcs take the place of the expo/log curves. Intermediate shapes have flatted regions between curve slopes.
4. **Plateaus and cliffs** - The attack slope morphs from a shape with a flatted plateau to a shape with a vertical cliff. T he release slope morphs between the same shapes in the other direction.
5. **Moving lump** - The attack slope is always linear, morph moves a lump down the otherwise exponential release slope.
6. **Fixed lump** - The attack slope is always linear, morph increases the size of a lump in the middle of the otherwise exponential slope.
7. **Compressor** - Snapshots of a model of a compressor with increasing ratio, decreasing attack time, and decreasing threshold .
8. **Variable sustain** - Attack slopes are linear, release slopes mimic an ADSR with fixed decay and release time and increasing sustain level .

Sequence

1. **Ridges** - Evenly spaced ridges emerge from a smooth slope with increasing number of ridges.
2. **Euclidean Ridges** - Distributes ridges across the slopes according to the euclidean algorithm and morphs through underlying patterns.
3. **Bounce** - Morphs through snapshots of a bouncing ball simulation with increasing number of bounces.
4. **Spring** - Morphs through snapshots of a model of a vibrating spring.
5. **Ramps** - Each slope consists of an increasing number of ramps; existing ramp start and end points stay in place as new ramps are added.
6. **Sinusoids** - Each table is comprised of 16 half-sinusoids with varying start and end levels.
7. **Sequences** - Interpolate through a set of 16-step sequences, with 8 steps per slope.
8. **Steps** - The slopes are stepped, morph adds steps.

5. Presets

Meta ships with 6 presets covering the 6 fundamental behaviors of the **FREQ** and **LOOP** parameters. This allows you to focus on experimenting with the other parameters without totally changing the function of the module, and easily recall a specific behavior.

Drum (SH button)

FREQ: *audio*
LOOP: *off*
TRIG: *kick*
S+H: *off*
TABLE: *impulse*
⋀ **Out:** *drum envelope*
⎍ **Out:** *high during release*

Try an example patch

Oscillator (TABLE UP button)

FREQ: *audio*
LOOP: *on*
TRIG: *pendulum*
S+H: *track A, decimate B*
TABLE: *perlin*
⋀ **Out:** *triangle*
⎍ **Out:** *high during attack*

Try an example patch

AR Envelope (FREQ button)

FREQ: *envelope*
LOOP: *off*
TRIG: *gated AR*
S+H: *sample and track A*

TABLE: *fixed lump*
∧ **Out:** *contour*
⎍ **Out:** *high during release*

Try an example patch

Looping AR (TRIG button)

FREQ: *envelope*
LOOP: *on*
TRIG: *hard sync*
S+H: *sample and track A, decimate B*
TABLE: *plateaus and cliffs*
∧ **Out:** *triangle*
⎍ **Out:** *high during attack*

Try an example patch

Modulation sequence (TABLE DOWN button)

FREQ: *sequence*
LOOP: *off*
TRIG: *pendulum*
S+H: *sample and track A, sample and track B*
TABLE: ***bounce***
∧ **Out:** *contour*
⎍ **Out:** *high during release*

Try an example patch

Complex LFO (LOOP button)

FREQ: *sequence*
LOOP: *on*
TRIG: *hard sync*
S+H: *off*
TABLE: ***waves***
∧ **Out:** *triangle*
⎍ **Out:** *high during attack*

Try an example patch