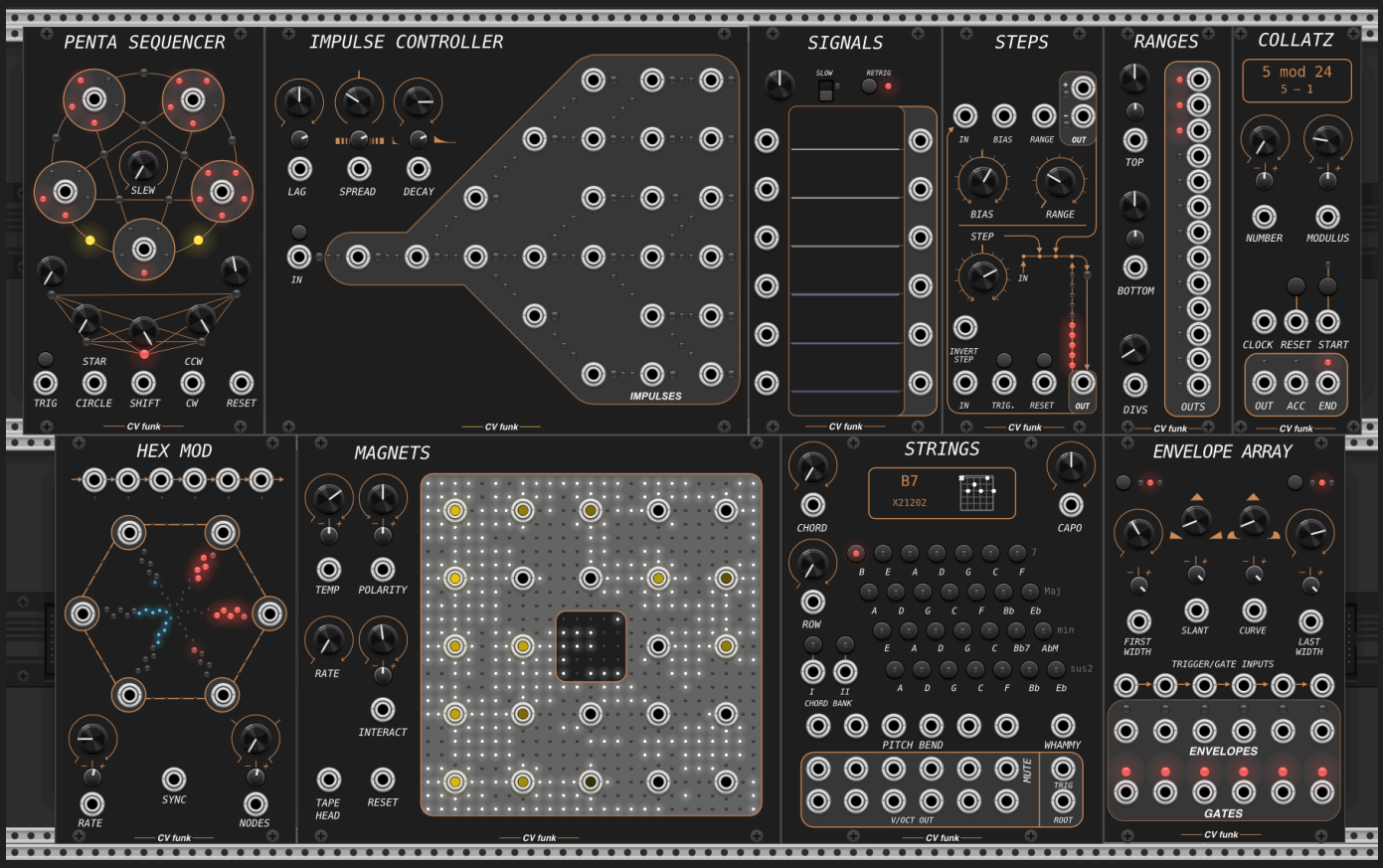
CV funk

CV funk Module Collection for VCV Rack



User Manual

CV funk

DONATE

All CV funk modules are made available for free. If you enjoy these modules and want to support their development you can make a donation at this link:

LICENSES

All source code for these plugins can be found on GitHub (https://github.com/codygeary/CVfunk-Modules).

All code and artwork for CV funk modules is copyright © 2024 Cody Geary / CV funk and licensed under MIT.

MIT License

Copyright © 2024 Cody Geary / CV funk

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CV funk Module Menu

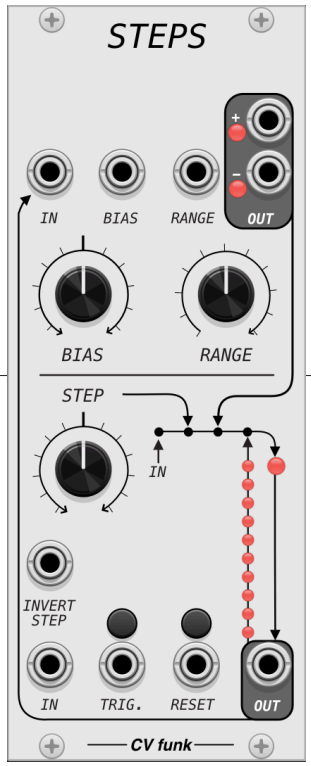
STEPS

Comparator Stage

IN: The signal to be compared. An input ***breaks all normals*** to stepper stage.

BIAS: Center of the window. Range ±5V

RANGE: Size of the window. Range 0–10V



OUT1: Two 5V gate outputs indicate if **IN** is above or below the window.

OUT is scaled by **RANGE** and normalled to the stepper circuit.

Stepper Stage

IN: Sums with the **STEP** knob to set the step size. Range ±1V

TRIG: Triggers the stepper.

RESET: Resets the stepper to **BIAS**. If the comparator normals are broken, resets to 0V.

INVERT STEP: Gate signal inverts the value of **STEP+IN**

MIXER: The mixer sums up:

IN + STEP + OUT1 + OUT2

Steps out of the window range result in ±RANGE from the OUT1 normal.

OUT2: Outputs the stepper value.

Normals to Comparator **IN**.

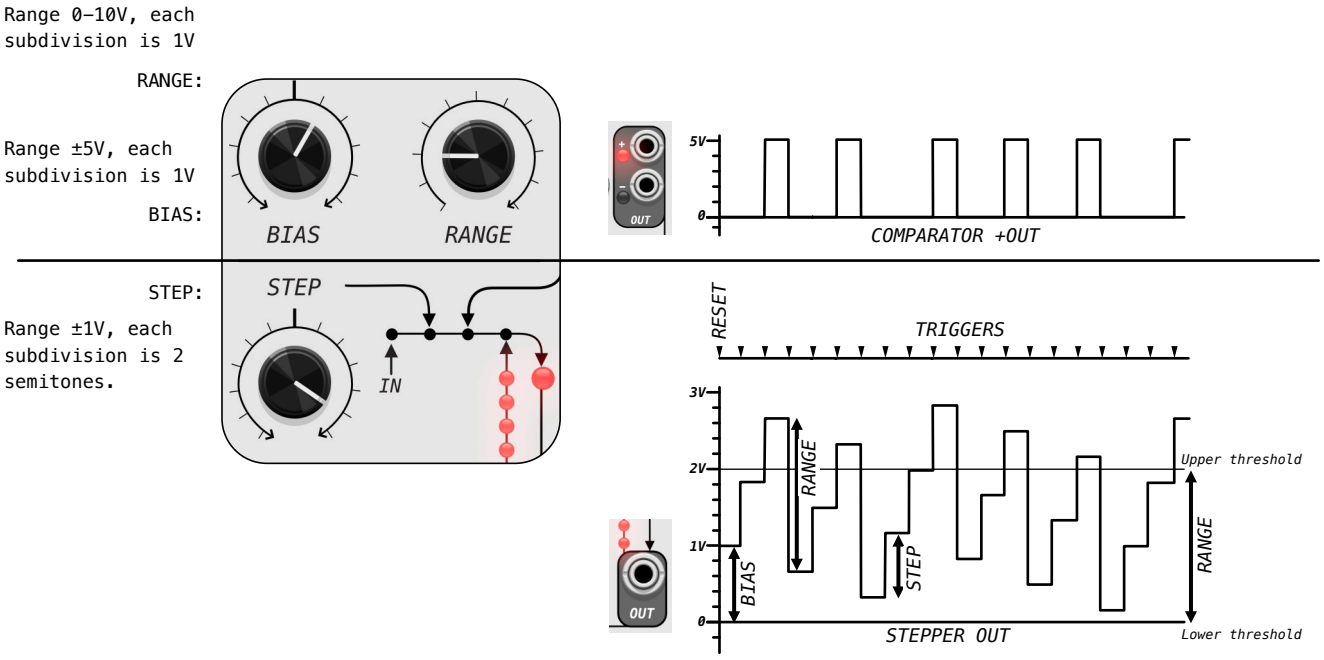
BUTTONS: Manual trigger inputs for **TRIG** and **RESET**

LIGHTS: The column of lights indicate the position of the stepper in the window range in 10% increments.

The single larger light indicate when a step has been triggered or reset.

STEPS combines comparator functionalities with a step sequencer to provide precise control over step signal generation. The stepper increments a voltage in a voltage window, and can be used to generate CV for fixed-interval scales of any width.

STEPS



Usage

1. Parameter Control: Adjust the **BIAS**, **RANGE**, and **STEP** parameters to set the initial conditions for comparison and step modulation.

2. Comparator Input: Feed the signal to be compared into the comparator stage **IN**. The module compares this input against the internally set threshold, influenced by **BIAS** and **RANGE** parameters.

3. Normalled Connections: Using the comparator input breaks the normalization to the stepper section, allowing both sections to be used independently. When normalled, the comparator provides feedback voltages to the stepper to bump it back into the window range after a step goes out of bounds. The lag of one step for corrections produces variation in the output.

4. Interval Sequencing: Utilize the **STEP** input to dynamically control the interval between steps. The **STEP** knob ranges from −1...1V, and is summed with the input. Inputs are scaled down 1/5, so that a −5...5V signal will sweep the full knob range for step size.

5. External Modulation: Patch signals into **BIAS** input, **RANGE** input, and **INVERT** input for dynamic control over the module's parameters.

6. Trigger and Reset: Use **TRIG** input to advance the step sequencer and **RESET** input to reset the sequence to its initial state.

Patch Suggestions

Dynamic Rhythms: Create dynamic rhythmic patterns by modulating **STEP** with LFOs or other modulation sources. Patch the comparator gate **OUTs** to obtain different clock divisions.

Melodic Sequencing: Use the stepper **OUT** to drive the pitch of an oscillator, crafting evolving melodic lines influenced by the comparator's operation. For more interesting patterns use a sequencer set to a slow clock division to program **STEP**, **RANGE** and **BIAS**, allowing you to sequence musical phrasing rather than individual notes.

Counting: Adjust the ratio between **RANGE** and **STEP** size to derive the number of desired steps. For example setting **RANGE**

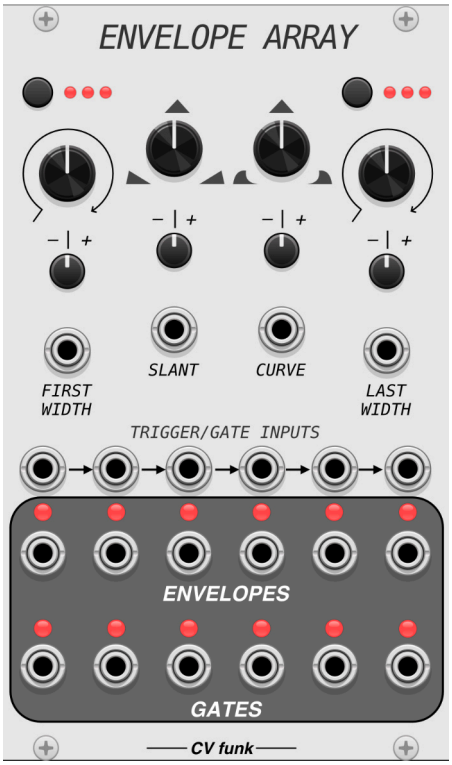
ENVELOPE ARRAY

RANGE: The **RANGE** button selects between three range settings: fast, seconds, minutes.

FIRST WIDTH: Sets the cycle time of the first channel. The knob ranges based on the range setting. CV can add or subtract to select widths even beyond the knob range.

LAST WIDTH: Sets the cycle time of the last channel. Intermediate channels will have times that span **FIRST** and **LAST**.

INPUTS: Each channel has its own input trigger. Triggers are normalized to the right by default.



SLANT: Sets the slant of all related envelopes.

CURVE: Adjust from Log to Linear to Exp and extremes.

Note: **SLANT** and **CURVE** will slightly influence the WIDTH.

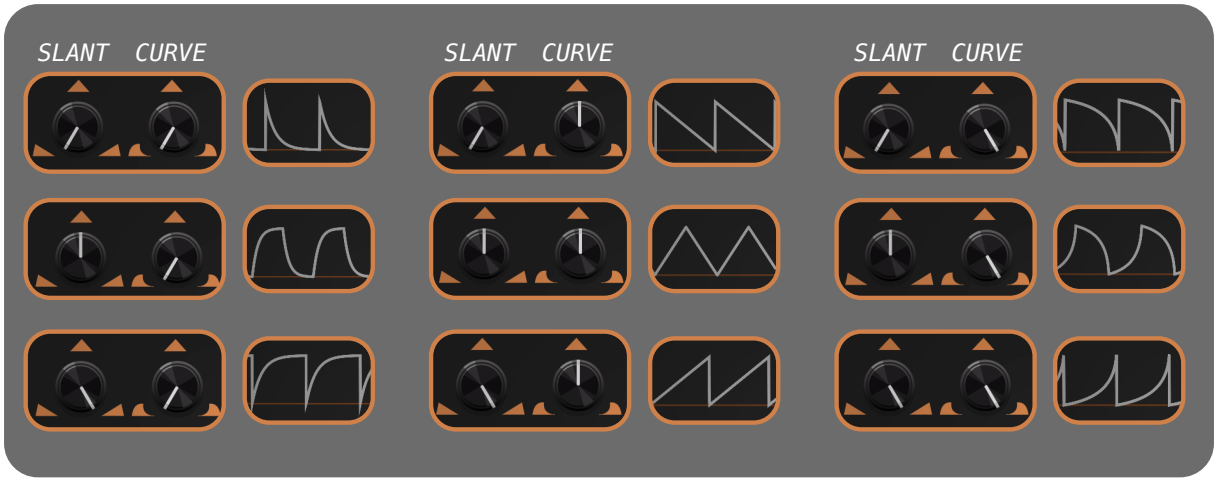
ENVELOPES: Six related envelope outputs.

GATES: Each gate goes high when the envelope is at 0V. Allows for extensive self-patching and chaining of multiple modules.

POLYPHONIC: 4-channel polyphony on the input triggers, envelope and gate outputs

ENVELOPE ARRAY is an advanced envelope generator module for VCV Rack, designed to provide a wide array of dynamic control over amplitude shaping with six stages and complex modulation capabilities. The module can be self-patched to create a wide variety of modulations and is capable of self-oscillation when self-patched.

ENVELOPE ARRAY



***SLANT** and **CURVE** combine to provide a wide range of possible envelope shapes.*

Usage

1. Set Envelope Stages: Adjust **FIRST** and **LAST** width knobs to define the width of initial and final stages of the envelope array, the other outputs will take intermediate values.

2. Shape Modulation: Use **SLANT** and **CURVE** knobs to shape the envelope's trajectory, creating everything from logarithmic to linear to exponential responses.

3. Dynamic Control: Patch CV signals into the respective attenuation inputs for real-time modulation of **SLANT**, **CURVE**, and **FIRST**/**LAST** width parameters.

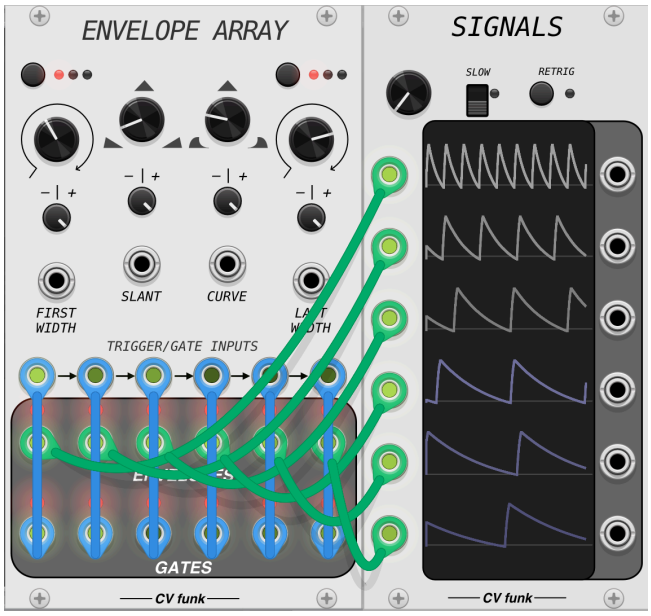
4. Monitor Envelopes: Observe the module's LED indicators for real-time visual feedback on the envelope stages and end-of-cycle signals.

5. Integrate with System: Utilize the end of function **GATE** outputs to trigger or synchronize with other modules, creating complex rhythmic patterns or sequences

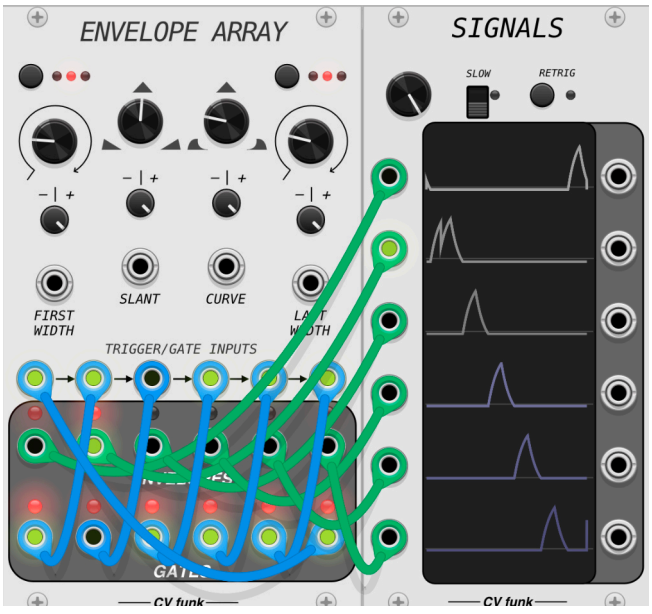
Patch Suggestions

Complex Modulation: Use the ***Envelope Array*** to modulate filter cutoffs, oscillator frequencies, or VCA amplitudes, creating evolving textures and timbres.

Rhythmic Sequencing: Connect the **GATE** outputs to sequencer reset inputs, using the envelope stages to define rhythmic patterns.

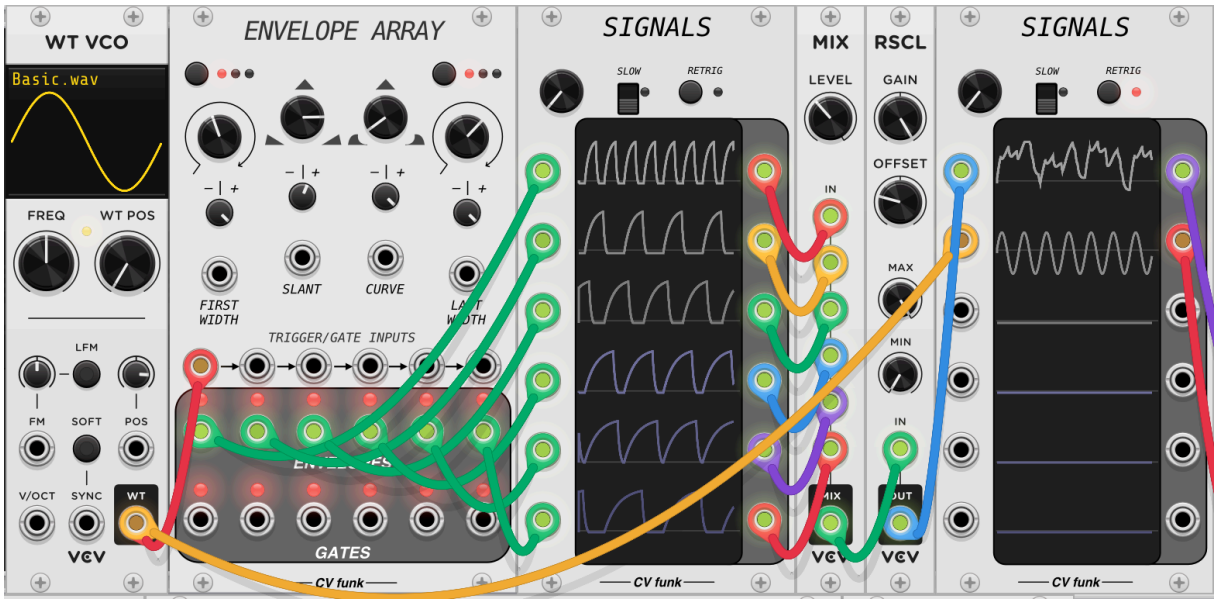


Six Oscillators: Self-patch each channels **GATE** to **IN.** The **GATE** will trigger a gate at the end of each cycle, starting a new cycle. The oscillators will not be in phase.



Chained Envelopes: Patch Channel 1 **GATE** into Channel 2 **IN,** Channel 2 **GATE** into Channel 3 **IN,** etc. When one envelope completes it triggers the next in the series. Patch Channel 6 **GATE** back to Channel 1 **IN** to daisy chain all of the envelopes. Use the envelope outs to self-modulate other parameters to generate more complex modulation patterns.

ENVELOPE ARRAY



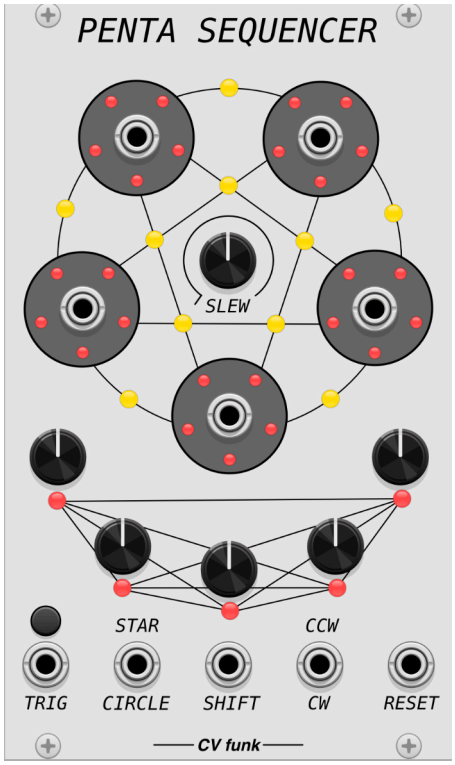
Sub Harmonics: Set both ranges to audio rate. Patch the output from another oscillator into Channel 1 **IN**, and adjust the **FIRST** range to be shorter than the wavelength (use *SIGNALS* to help you adjust the range). Patch all 6 envelope outputs to a mixer. Use *VCV RSCL* to DC offset the envelopes by −5V. Since each channel cannot reset until it is at the end of its cycle, the channels will be retriggered in overlapping ways but synchronized with the root oscillator, resulting in rich harmonics.

PENTA SEQUENCER

OUTS A–E: Five outs, A,B,C,D,E each output one of the five notes. LEDs around the OUT indicate which stage I–V they are outputting.

SLEW: Sets the slew limit based on note timing and interval. Set to the left left there is is no slew. At full right it slews based on the last trigger interval and the distance to the next note on the current track.

KNOBS I–V: Sets the voltage at each stage I, II, III, IV, V.



TRIG: Trigger input, cycles the output mappings on the current path, either STAR or CIRCLE.

SHAPE: (STAR/CIRCLE) Defaults to CIRCLE (I→II→III→IV→V→I), a gate input engages STAR mode (I→III→V→II→IV→I)

DIR: (CW/CCW) Defaults to CW (clockwise). A gate input changes direction to CCW (counterclockwise)

SHIFT: Functions as a transpose input. Sums with all stages I–V.

RESET: Trigger to reset the sequencer to A→I.

PENTA SEQUENCER is a versatile 5-step sequencer module, offering unique sequencing capabilities in a small footprint. The sequencer outputs all five of its notes simultaneously, can switch between any permutation of outputs, and has a computed slew that adjusts dynamically to the trigger input.

PENTA SEQUENCER

Usage

1. Knob Assignment: *Each of the five knobs (**I, II, III, IV, V**) is mapped to one of the five outputs. Advancing the sequencer rotates the mapping between output and knob, indicated by lights around each output. The knob corresponding to the bottom-most active output is indicated by a light.*

2. Mode Selection: *A gate to the **CIRC** input will switch between Circular (<=1V) and Star (>1V) mappings when held. Changing modes will change the mapping of outputs between star and circular relative to the output indicating a 1.*

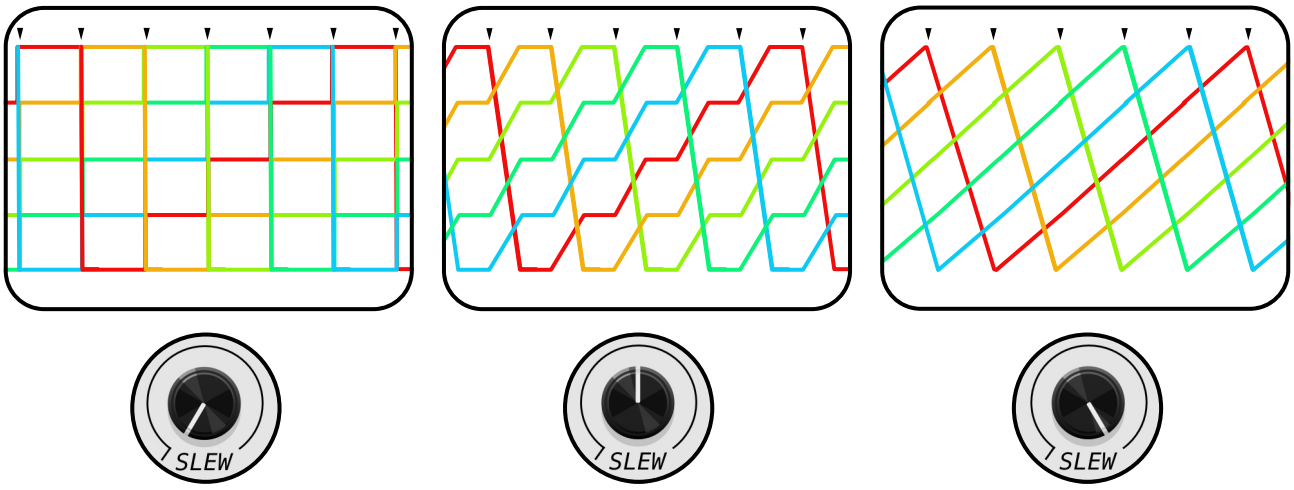
3. Direction Control: *A gate to the **CCW** input will switch between Clockwise (<=1V) and Counterclockwise (>1V) mappings when held. Changing direction will exchange the left and right outputs.*

4. Advancing the Sequence: *Send trigger signals or button press to the **TRIG** input to advance through the sequence according to the selected mode and direction.*

5. Resetting the Sequence: *Send a trigger signal to the **RESET** input to return the sequencer to the first step.*

6. Adjusting Slew: *Use the **SLEW** knob to control the transition smoothness between steps, the slew is synchronized to both the trigger timing and interval size, this means bigger octave jumps slew faster.*

(I=0V, II=1.25V, III=2.5V, IV=3.75V, V=5V)



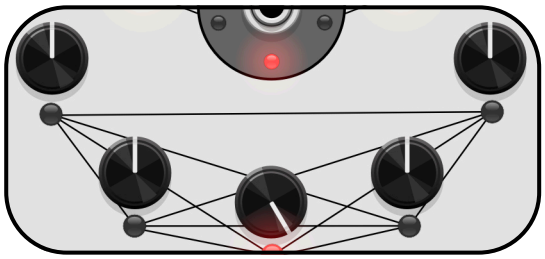
SLEW *allows continuous morphing between stepped and sloped Cvs.*

Patch Suggestions

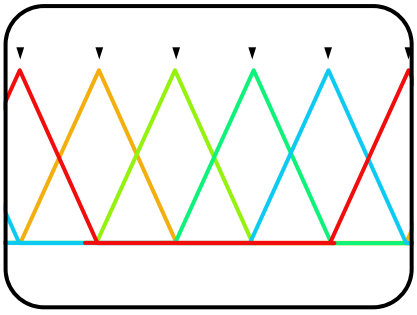
Slewed Octaves: Set each of the five knobs to −2, −1, 0, 1, 2. Set the slew to 0.25 for smooth slewing. Patch each of the five outs to a WT-VCO, adjust each to have a slightly different timbre. With a second copy of the *PENTA SEQUENCER*, set five different notes, set the slew of this one to 0.0, and ideally quantize the outputs before patching to the transpose input of the other sequencer. Now the slew will be perfectly in tune.

LFO: Set a slow clock to the trigger input, set the slew to 1.0. You will have five slow LFOs. The slew interpolates using the time between the last two inputs (trigger/shape/direction) to compute the ramp.

Fixed Root Note: Reset the sequencer so that the bottom output is set to 1. Send gates to switch between **STAR/CIRCLE/CCW/CW**, the bottom output will always stay the same.



Five Channel Fader – Set the first knob I to +5, set all the other positions to 0. Patch each output to a channel of a mixer. Set the slew to 1. A slow trigger to the input will result in smooth transitioning between all five channels on the mixer. Useful to pan between different soundscapes.

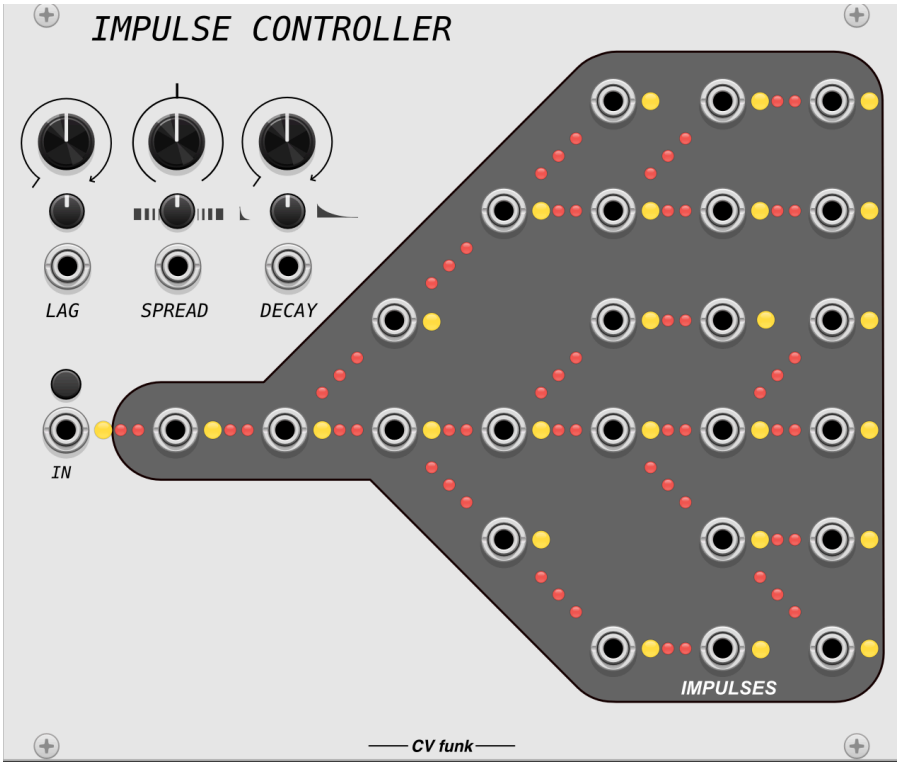


IMPULSE CONTROLLER

LAG: Determines the delay time for signals to pass between nodes.

SPREAD: Determines the synchronicity of the outputs, turning the knob left brings the nodes into unison, and right apart.

DECAY: Determines the decay time for nodes once activated.

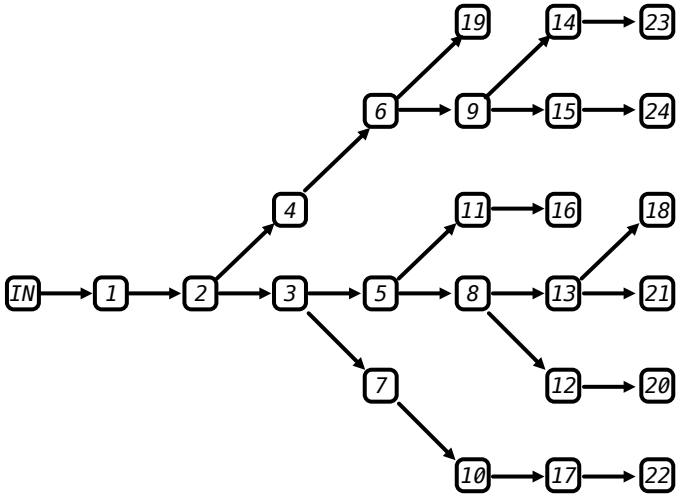


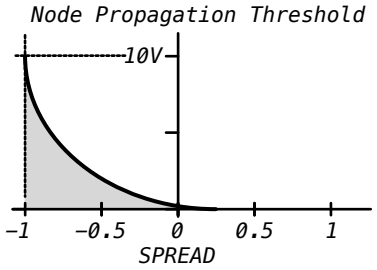
IN: A gate or trigger stimulates the first node, which propagates along the red lighted network.

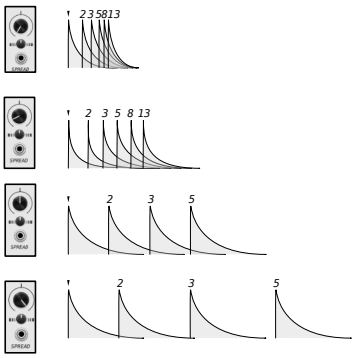
IMPULSES: The yellow light to the right of each output indicates the signal level. 24 outputs produce related decay envelopes in rhythm with the input signal.

IMPULSE CONTROLLER simulates the propagation of waves through a medium, offering unique visual and auditory feedback based on the interactions within a network of nodes. Each of the 24 nodes can generate a decay envelope when stimulated, allowing for long decay tails and rhythmic synchronized pulsations.

IMPULSE CONTROLLER







Usage

1. Initial Setup: Adjust the **SPAN, SPREAD,** and **DECAY** parameters to define the base characteristics of the wave propagation.

2. Signal Input: Introduce signals into the IN to initiate wave propagation across the network.

3. Button Interface: Manually trigger wave impulses with the button, useful for setup and tuning parameters.

4. Modulation: Utilize the dedicated **SPAN, SPREAD,** and **DECAY** inputs for real-time modulation of wave characteristics.

5. Output Utilization: Patch outputs from the 24 nodes into various destinations to explore spatial audio effects and dynamic modulation.

6. Self-Patching: Attenuverters on the inputs expand the self-patching possibilites with this module, and with 24 outputs, you have lots of options.

7. Lag: Adjust the timing for each node to awaken its child nodes.

8. Spread: Sets the voltage to which nodes must decay before they can propagate a signal. Setting the spread to the far left makes nodes fire in unison. **SPREAD** also changes the scaling of each nodes timing, negative spread makes the timing get shorter for each subsequent node activation, positive spread scales the timing from 1-2x longer at each node.

9. Decay: Adjust the decay rate for all nodes.

Patch Suggestions

– Spatial Audio Design: Use outputs to create immersive soundscapes, with *IMPULSE CONTROLLER* simulating movement across the stereo or surround field.

– Dynamic Modulation Source: Employ the module as a complex, evolving modulation source, with each node modulating different parameters in your patch.

– Networks: Patch multiple Wave Propagation modules together. By patching inter-patching the outputs from different Wave modules to their inputs, and parameters, it is possible to build complex rhythmic networks that can be used to drive expressiveness in your patches. The manual trigger button will jump start the self-patched network, now watch the signal propagate in fascinating ways.

SIGNALS

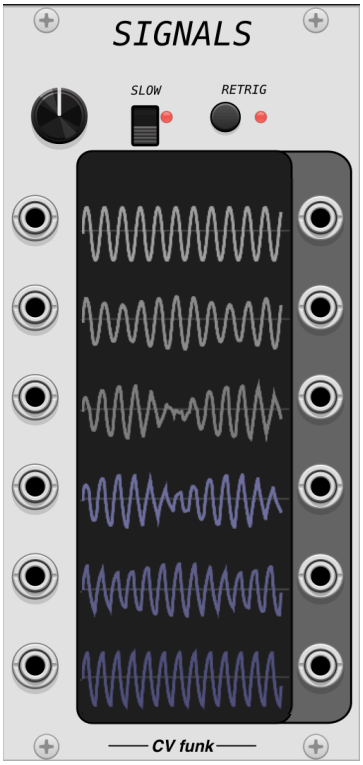
SPREAD: Determines the synchronicity of the outputs, turning the knob left brings the nodes into unison, and right apart.

DECAY: Determines the decay time for nodes once activated.

RANGE: Knob sets the time range. In normal operation it scales up to 1 sec. In 'SLOW' mode it scales to 10sec.

SLOW: Switches between 1sec and 10sec scales.

IN: Six inputs and six signals are displayed



RETRIG: Retriggering mode aligns signals based on 1V rising slope detection.

OUT: Six passthrough outputs simplifies using *SIGNALS* in patches.

SIGNALS is handy to tune up your envelope and LFO shapes. Simultaneous viewing of six channels in a small module makes it possible to fit these into your signal chain without getting in the way. Range knob and trigger latch make it useful for observing envelopes or free flowing signals.

RANGES

TOP: Sets the voltage of the top output. Sums with the input/ attenuverter.

BOTTOM: Sets the voltage of the bottom active output. Sums with the input/ attenuverter.

DIVS: Sets the number of subdivisions (and active ports). New outputs become linearly interpolated between **TOP** and **BOTTOM.**

RANGES generates a range of voltages from TOP to BOTTOM, in 0–12 divisions (DIVS). The inputs have CV control, allowing you to patch them with other modules, and for the module to be used in many applications, such as attenuator, gate generator, or crossfader.

Patch Suggestions

12-tone Scale: Set **TOP** to 0. Set **BOTTOM** to 1. Set **DIVS** to 12. Each output is now a note on the 12-tone scale.

4ths Interval Scale: Set **TOP** to 0. Set **BOTTOM** to 5. Set **DIVS** to 12. Now the outputs correspond to a scale of perfect fifths.

Signal Blending: Path two different signals to **TOP** and **BOTTOM**, set the attenuverters both to 1. Set **DIVS** according to how many crossfading outputs you require. Patch each output into a Signals module to visualize the different crossmixes. Adjust the attenuverters to change the ratio of mixing.

Dynamic Panning: Set **TOP** and **BOTTOM** to 0, and set their attenuverters to 1. Patch -5V and +5V to a fader module such as the *VCV FADE*. Patch fader outputs to the **TOP** and **BOTTOM** inputs. Set **DIVS** to the number of mixer channels you have, and patch the outputs to corresponding mixer channels. Crossfading the FADE module will pan all the channels at once.

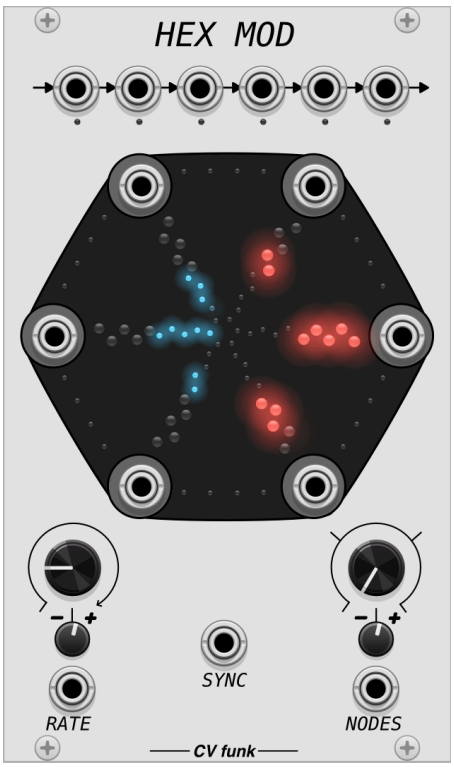
Gate Sequencer: Set **TOP** to 5. Set **BOTTOM** to 5. Set **DIVS** to 0. Patch the outputs numbers 3+ to gate inputs of other modules. A 5V sweeping voltage into **DIVS** will send 5V to all the outputs in sequential order.

HEX MOD

RATE: Sets the rate of the six oscillators. Ranges from 0.02Hz to 10Hz.

SYNC: By default an input to **SYNC** disables the **RATE** inputs. **SYNC** will synchronize the LFOs to an external clock or oscillator.

In the context menu an option is available to multiple **RATE** by **SYNC** to allow for more modulation options. In this mode **RATE**=1 will sync with the signal, **RATE**=2 will double the **SYNC** frequency.



INPUTS: Triggers reset an input phase and value to 0. By default triggers normalize to the right, in a loop.

NODES: Determines the phase alignment of the outputs. 0=hexagonal, 1=unison, 2=binodal, 3=trinodal

LIGHTS: Each output has 10 lights, that indicate the value of the output. The red lights each represent +1V, and the blue lights are each −1V. *You can disable the lights in the context menu.*

HEX MOD consists six low frequency oscillators, and produces dynamic, phase-related evolving patterns. Its unique NODE clustering control morphs between distinct phase relationships—hexagonal, unison, bimodal, and trimodal distributions—enabling a spectrum of CV from harmonious alignment to complex polyrhythms. Envelope inputs per oscillator allow for expressive modulation, resetting phases at peak points for synchronized or staggered pulsations.

Usage

1. Rate and Clustering: Begin by setting the **RATE** knob to control the base frequency of the oscillators, use the **NODE** knob to define the initial phase relationship and clustering pattern among the six oscillators.

2. Trigger Inputs: Introduce envelope signals into the inputs along the top of the module to dynamically influence the phase reset and modulation depth of each oscillator, tailoring the rhythmic interplay and synchronization. Trigger inputs normalize to their right, in a loop.

3. Rate CV and Node CV: Utilize **RATE** and **NODE** CV inputs for real-time modulation of the oscillators' frequency and clustering behavior, enabling responsive, evolving textures.

4. Output Exploration: Patch the LFO outputs to various destinations within your rack to modulate parameters with the evolving, complex rhythms generated by the *HEX MOD*.

5. Disable Light Blinking: Access the context menu to enable or disable the module's lights for performance or personal preference.

6. Sync Lock: By default the **SYNC** input locks the rate and disabled the **RATE** knob and CV. In the context menu, you can enable multiplying the synchronized rate by the **RATE** setting, allowing for CV control of FM of the synced rate or to multiply incoming clocks.

Patch Suggestions

Phased Panning: Control the panning of six channels in a mixer. Set the **RATE** or optionally use the **SYNC** input to tempo-lock the LFO. **NODES** set to 0 will evenly space the channels. A trigger to any of the top inputs will cause the sounds group in space and then slowly spread out again. Experiment with different settings of **NODES** to group the channels in different arrangements.

Waveshaped Outputs: Patch one of the outputs into **RATE**, and set the attenuverter. Patch another output into **NODES**, adjust the attenuverter.

Audio Rate Fun: Create audio rate modulations by plugging an oscillator into the **SYNC** input, allowing you to sync with waveforms much faster than the normal range of this module. In the context menu you can turn on an addition option to multiply **SYNC** with the **RATE** setting. Set **RATE** to 1 to mirror the sync source, or set it to 2 to double the frequency. Self-patch outputs to the **RATE** and **NODE** inputs using attenuation to create interesting timbers.

COLLATZ

NUMBER: Sets the starting number for the Collatz sequence.

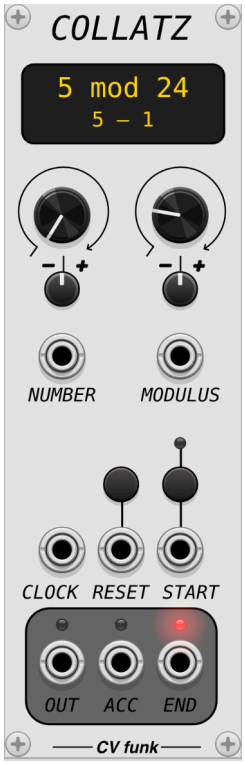
Even: /2

Odd: ∗3 + 2

Range 2–1000

CLOCK: The timing between pulses sets the clock interval for the polyrhythms. A **CLOCK** signal is needed for the module to output gates.

OUTS: The **OUT** and **ACC** outs are computed from the **MODULUS.** Each out is a gate signal that has a 50% pulse width.



MODULUS: Limits the number of pulses per clock interval. The remainder sets the main **OUT**, and the divisor sets the ACC out.

START: Triggers the start of the sequence. Once armed, the sequencer will wait for the next clock pulse before starting.

END: When the sequence reaches 1 it terminates and raises the **END** gate, rather than continue to 4–2–1 in a loop.

Patch **END** to **START** to make the sequencer cycle.

COLLATZ generates rhythmic triggers synced to a clock signal using numbers from Collatz sequences. You pick number, and if it's even divide by two, and if it's odd multiply by three and add one. This generates a sequence. According to the Collatz Conjecture, this series will eventually lead to one. When the module reaches 1, it terminates and outputs an end gate. Triggers are outputted as clock multiples of the number mod MODULUS, and the divisor is used to compute the accent output.

COLLATZ

Usage

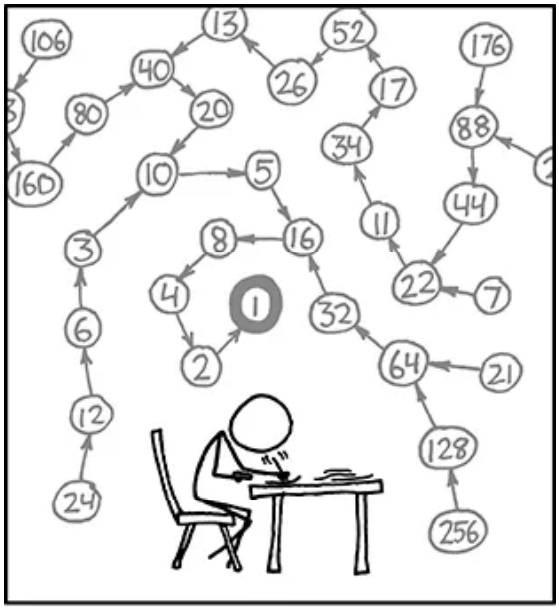
1. Number and Mod: Begin by setting **NUMBER** and **MODULUS.** The **MODULUS** sets the maximum number of pulses per clock interval, and is best set to a low number if you wish to use it for drum beats, but can potentially be faster up to audio rate depending on the setting.

2. Clock Input: This module requires a clock signal to work. If a clock is not patched, no output signals will go to **OUT** or **ACC.**

3. Start Input: Patch a trigger or gate to **START** to initiate the sequence. The value of **NUMBER** will be sampled and used as the starting number for the sequence, which will start on the next **CLOCK** pulse.

4. Reset: A trigger or gate to the **RESET** will stop the sequence and turn off outputs.

5. End Gate: When the sequence reaches 1, the END gate will go high. If you desire looping sequences, you can patch **END** into



THE COLLATZ CONJECTURE STATES THAT IF YOU PICK A NUMBER, AND IF IT'S EVEN DIVIDE IT BY TWO AND IF IT'S ODD MULTIPLY IT BY THREE AND ADD ONE, AND YOU REPEAT THIS PROCEDURE LONG ENOUGH, EVENTUALLY YOUR FRIENDS WILL STOP CALLING TO SEE IF YOU WANT TO HANG OUT.

Image Credit: XKCD comics https://xkcd.com/710/
Creative Commons Attribution-NonCommercial 2.5 License

Patch Suggestions

Drums: The snappy gates of this module can be used for snare drums or other percussive elements. Link **CLOCK** to your main clock, or a clock division, and patch **END** to **START** to cause the module to cycle.

Strumming: Use the clock multiplied outputs to activate the plucking of Karplus Strong strings.

Basic Rhythm Patterms: Set **NUMBER** to 4,8,16 or 32 to get a simple drum pattern similar to rotating clock divisions.

32→16→8→4→2→1

16→8→4→2→1

8→4→2→1

4→2→1

Patterns are often nested, so you can choose depending on the length you wish to use:

12→6→3→10→5→16→8→4→2→1

6→3→10→5→16→8→4→2→1

3→10→5→16→8→4→2→1

10→5→16→8→4→2→1

5→16→8→4→2→1

Low numbers can result in long patterns, so it's a bit hard to predict. But each **NUMBER** will always lead to the same sequence, leading eventually to 4→2→1.

9→28→14→7→22→11→34→17→52→26→13→40→20→10→5→16→8→4→2→1

The MODULUS input allows for a variety of different drum patterns to be produced from any Collatz sequence. Experiment with different combinations of **NUMBER** and **MODULUS**, and for sequences that go on for 100s of steps don't forget there is a **RESET** input! Explore and enjoy.

STRINGS

CHORD:

Selects the chord, spans 1–7.

ROW:

Selects the row of chords, spans 1–4.

CHORD BANK:

The two input combine to specify 1 of 4 chord banks. Manually select a bank with the buttons. Button latching inverts the gate operation, allowing you to pick which bank is selected by a high gate.

PITCH BEND:

Uses the absolute value of the input. A 10V envelope will result in a 2 semitone bend.

V/OCT OUT:

Six outputs, each correspond to one string. Muted strings output −10V, for logic uses in patching.

MUTE:

Outputs a gate when a string is muted in a chord. Useful for emulating the guitar by muting the outputs of oscillators.

TRIG:

Outputs a trigger signal anytime a **BUTTON** is pressed, or **CHORD**, **ROW**, **CHORD BANK** or **CAPO** setting is changed.

CAPO:

Transposes the chords. Inputs range from −12 to 12V, in 1 semitone intervals, or 12V/oct.

BUTTONS:

Manually trigger a new **CHORD**. The illuminated button indicates the **ROOT** note of the CHORD.

The chord type is indicated right of the buttons, and details of the chord are shown in the display.

WHAMMY:

Works like the **PITCH BEND** input, but applies to all inputs in addition to **PITCH BEND**.

ROOT:

Outputs the **ROOT** note of the current **CHORD**.

STRINGS produces six V/oct signals representing the strings of a guitar. Buttons and inputs select between different chord banks and voicing options, which are indicated dynamically on the panel and display. Pitch bend, whammy bar and capo are implemented allowing guitar-like effects to be intuitively produced.

STRINGS

Usage

1. Chord and Row: Begin by setting **CHORD** and **ROW.** The **CHORD** addresses the column, while **ROW** sets the row on the button grid.

Row 1:	B	E	A	D	G	C	F
Row 2:	A	D	G	C	F	Bb	Eb
Row 3:	E	A	D	G	C	Bb7	Abmaj
Row 4:	A	D	G	C	F	Bb	Eb

The arrangement of notes on the button panel are

2. Chord Bank: The two inputs **I** and **II** combine to select one of four chord banks.

(I,II)

(0,0) Bank 1: Dominant 7th, Major, Minor, Suspended 2nd

(1,0) Bank 2: Dominant 7th Barre, Major Barre, Minor Barre, 6th

(0,1) Bank 3: Suspended 4th, Major 7th, Minor 7th, Dominant 7 Suspended 4th

(1,1) Bank 4: Add 9, Augmented, Minor 6th, Minor 9th

Bank 1 contains the most essential chord types. Chords in the rows are arranged so that the relative major and minor chords are juxtaposed, allowing for easy transitions.

Bank 2 contains Barre chords, but also B and E major in the top row with the other Barre chord 7th variants. Major and minor Barre chord variants are available. In the Minor chord row, two exceptions, Bb7 and AbMaj, add more flavor.

Bank 3 features Major and Minor 7ths, which provide tension and harmony to chord progressions. In Row 3, the Ab7 offers a contrasting chord.

Bank 4 features more colorful chords, especially the Minor 9th chords in Row 4.

3. Mutes: Many guitar chord fingerings contain muted strings. In this case, the string will output −10V. The mute gates allow you to easily patch mutes to connected oscillators when the chord voicing requires.

4. Pitch Bend and Whammy: PITCH BEND and **WHAMMY** inputs allow you to bend the notes positively, all inputs will be rectified before applying the bend, simulating string bending. A 10V input will result in a 2 semitone bend, making it easy to program bends with envelope outputs. **WHAMMY** applies the bend to all strings at once. **PITCH** and **WHAMMY** can be applied at the same time for extra bending.

5. Capo: The **CAPO** input simulates adding a capo to the guitar. Each 1V of input results in 1 semitone of offset. Try patching the VCV *OCT* module into **CAPO** for a nice interface. Negative values to **CAPO** will let you tune the guitar down.

6. Trigger Out: Every time the chord or fingering changes, or a **BUTTON** is pressed, a trigger is sent to the **TRIG** output. This allows you to trigger the strumming of the strings everytime a new chord is selected, for example by patching **TRIG** into the IN of a module like *IMPULSE CONTROLLER*.

7. Root Out: The ROOT output send an extra v/oct signal that is the root note of the chord. This is useful if you want to have another instrument accompany the guitar sounds.

Patch Suggestions

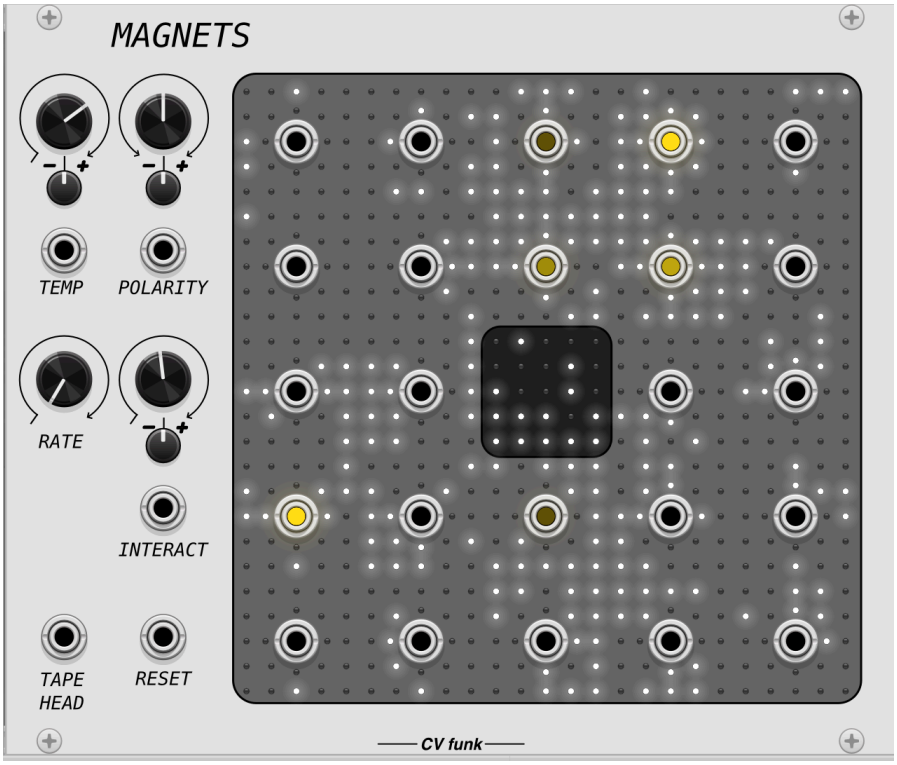
Guitar: Patch the six v/oct **OUT** signals into six Karplus Strong modules. Patch the outputs of each oscillator into a Bogaudio *MUTE8* module signal input, patch the outputs of the MUTE gates to the mute input of *MUTE8*, patch the outputs into your mixer. Patch the **TRIG output** into the IN of CV funk *IMPULSE CONTROLLER.* Patch different outputs into the plucking input of the Karplus Strong module, or patch the outputs to open a VCA if using an oscillator.

Note Bending: Patch the **TRIG** output into CV funk *ENVELOPE ARRAY* or other envelope generator. Patch an envelope output to the 5th string **PITCH BEND** input.

Neck Slide: Slide down the neck, patch a decay envelope from *ENVELOPE ARRAY* into the **CAPO** input. The quantizing will sound like sliding down the frets.

Glass Slide Effect: Patch Sine from VCV *LFO* to the **WHAMMY** input. Set the *LFO* module to 6Hz, and unipolar modulation. Patch **TRIG** out to the Reset of *LFO* to sync the oscillation to the chords.

MAGNETS



TEMP:

Temperature of the simulation.

POLARITY:

Changes the polarization of the array.

INTERACT:

Sets how much nodes interact with each other.

RATE:

Sets the delay between simulation updates. Faster updates requires more CPU.

TAPE HEAD:

Sets the up/down spins of the central square. Also sums with POLARITY to affect the whole array.

RESET:

Randomizes the spin state of all the nodes, based on the POLARITY.

Usage

1. TEMP: Set the temperature of the simulation. A low temp can freeze the simulation, while a hot temp adds lots of randomness.

2. POLARITY: The polarization input allows you to influence all of the nodes at once, making them more likely to polarize when updated in the simulation.

3. RATE: Sets the update timing for the Metropolis/Monte Carlo simulation. Faster rates require more CPU.

4. RESET: A trigger or gate to the **RESET** will randomize the spin states of all lattice points according to the **POLARITY** setting.

5. INTERACT Sets how much neighboring lattice nodes interact with each other. This setting can induce phase changes in the array. Lower **INTERACT** settings allow the array to behave more fluid, while higher settings can cause some states to be trapped.