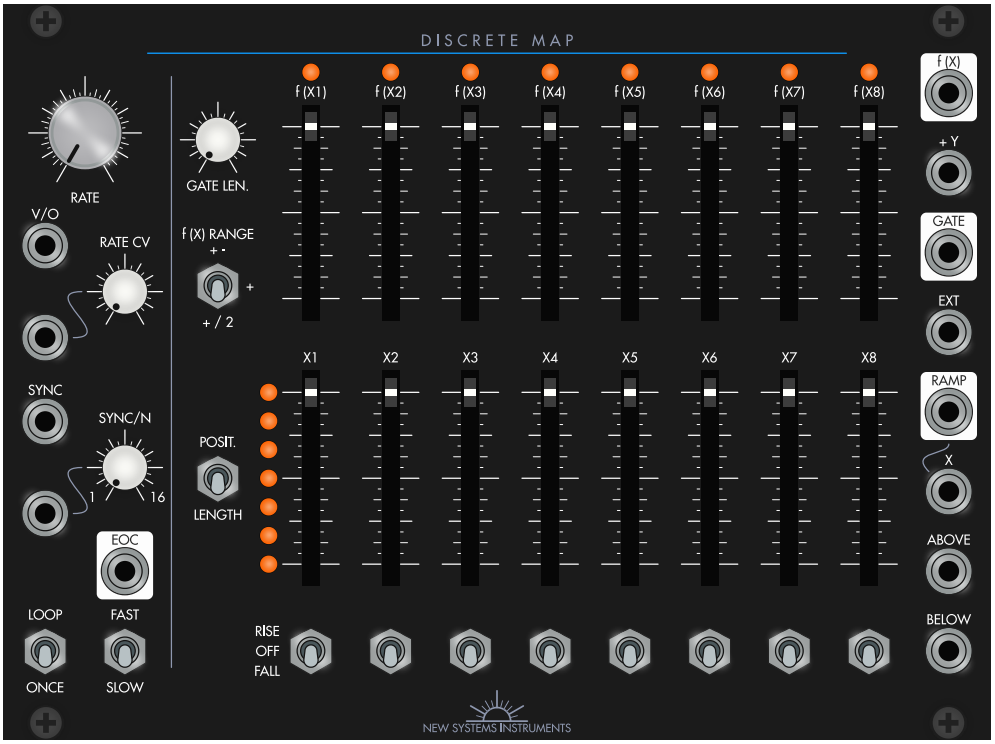
DISCRETE MAP AND A/B/C EXPANDER



Sequencer

Manual Revision 1.0

NEW SYSTEMS INSTRUMENTS

ACKNOWLEDGEMENTS

Like all good engineering, Discrete Map came about in a history of engineering that I didn't create, within a context of conversations and suggestions without which it wouldn't be the module that it is. The chief thanks go to Anton Riehl (Nireus), who provided many hours of testing, valuable feedback and ideas about usage, and with whom I had many conversations about possible directions. Thanks also go to Lachlan Fletcher, for many insights at the idea stage, as well as the core of the grouped expander idea. And thanks to everyone at Knobcon 2024 who tried out a prototype and gave me many suggestions about possible directions, especially Ben Wilson (DivKid) and Devin Weston (Weston Audio).

Clearly, inspiration for Discrete Map comes from Don Buchla's 248 Multiple Arbitrary Function Generator (MARF). Perhaps less clearly, considerable inspiration comes from Serge Tcherepnin's abstract approach to modular. While this module is unrelated to the Serge sequencer, it draws inspiration from the way sequencing is often done on non-sequencer modules in Serge systems.

SPECIFICATIONS

Discrete Map

Size	34HP
Depth	25mm
Clock Rate	4 minutes per cycle to 12kHz
Min. Ext. Step Interval	12.5µs
Tuning Accuracy	~7 octaves
Input Range	0 to 5V (SYNC, SYNC/N, EXT), −5V to 5V (ABOVE, BELOW), −9.5V to 9.5V (others)
Output Range	−5V to 5V, configurable
Gate	0–5V, 6µs to 2.8s
Power Consumption	+12V 180mA −12V 115mA
Input Impedance	Varies* (ABOVE, BELOW), 100kΩ (others)
Output Impedance	1kΩ
Output Drive	2kΩ (min), 20kΩ+ (ideal)

* In positional mode, ABOVE and BELOW have an input impedance of 12.5kΩ. In length mode, the module senses the total impedance, including the output impedance of any other stage. The impedance between ABOVE and BELOW will be the sum of 100kΩ times each slider position. Note in particular that when every slider is all the way down, the impedance between ABOVE and BELOW can be very low, and this might produce unexpected behavior. The safety of the connected devices is guaranteed by current limiting, which will dynamically change the impedance.

A / B / C Expander

Size	10HP
Depth	33mm
Input Range	−8V to 8V
Switching Impedance	240–360Ω
Switching Current Limit	6–10mA
Gate	0–5V
Power Consumption	+12V 35mA −12V 35mA
Input Impedance	100kΩ
Output Drive	2kΩ (min), 20kΩ+ (ideal)

INSTALLATION

Before installing the module, make sure the power is off. Attach the power cable to the module and to the bus. Double check the alignment of the red stripe with the markings on the module and the bus. The red stripe should correspond with −12V, as is standard in Eurorack. Screw the module to the rails of the case using the provided screws. (M2.5 and M3 size screws are provided.)

Please be careful and don't accidentally connect power to the expander input. New Systems Instruments uses only 10 pin power headers. Additionally, we will provide multicolor cables for expanders, and normal grey cables for power, but of course we can't guarantee other companies will do the same.

New Systems Instruments modules all have keyed headers and properly wired cables. But please remember to double check the other side of the cable for proper installation with the bus. Additionally, if using a different power cable, **note that not every company wires modular power cables such that the red stripe will align properly with a keyed header**. While our modules are reverse polarity protected as much as is practical, it is still possible that you could damage the module, your power supply, or another module by installing the power cable improperly.

Lastly, please fully screw down the module before powering on your case. The electronics are potentially sensitive to shorts, and if the module is not properly attached to a case, there is a small risk of contact with conductive or flammable matter.

INSTALLING EXPANDERS

Connecting an expander—use whichever end is most convenient

With a 16 pin expander port on the rear, each Discrete Map can connect to an essentially arbitrary number of expanders. To connect an expander, plug one end of the provided multicolor 16 pin ribbon cable into the expander port on Discrete Map, and plug the other end into the expander port on the expander module. To connect further expanders, plug each new expander into the 16 pin port on the expander cable.

OVERVIEW

Discrete Map provides the ability to control discrete musical things—notes, events, and rhythms—with continuous musical things—time durations and voltage contours. It can serve the same purpose as a sequencer, a chord generator, or a quantizer, as well as yield many new kinds of functions. At audio rate, it maps a continuous waveform to a stepped waveform, providing a unique kind of graphic VCO or waveshaper. Discrete Map runs directly on logic circuits, and does not contain a conventional DAC/ADC or microprocessor. Consequently all parameters operate at audio rate and do not have an underlying sample rate or bit depth.

Discrete Map transforms a continuously moving input voltage into a stepped output voltage. The bottom portion has eight sliders and switches, setting up to eight thresholds which activate the corresponding stage when the input voltage rises above or falls below that threshold, depending on the switch position. Expanders can modify these thresholds with CV. The top portion has eight corresponding sliders that set an output voltage corresponding to each stage. Expanders can do other things when a stage is activated, such as changing the position of a sequential switch. Discrete Map also contains a ramp oscillator optimized for use as a clock. When using this internal clock, during each cycle the voltage increases at a constant rate, and you can think of the bottom sliders as placing a note at a position in time within a repeating phrase.

By providing a simple interface between the continuous and the discrete, Discrete Map is able to construct musical notes and events in an unusually fluid and versatile way. It is designed to work with *modular time*, where rhythms often evolve organically from the interaction of different voltage contours, rather than coming from a clock/sequencer to which everything else is subordinate. Nevertheless, Discrete Map has a sophisticated sync interface that is able to work well with a conventional sequencer.

The A / B / C expander has a bank of switches to assign each of the eight stages into one of three groups: A, B, or C. The threshold for these three groups can be modified with CV. Additionally, the A / B / C expander provides a consolidated gate output for each group, as well as a switch that sequences the connection between the A, B, and C jacks and the COM jack.

HOW TO READ THIS MANUAL

This manual is intended to be an in-depth resource for exploration as you continue your journey through sound and synthesis. It provides a thorough analysis of how this module performs in a wide variety of contexts and applications. You are not at all required to read the whole manual before using the module. Read according to your own learning style. I recommend reading over the Overview and Interface section, then going through Quick Start and trying out some patches. From there, browse through the rest of the sections and read what interests you.

1–10. Clock Section, 11–17. Map Section, 18–26. Event Section

1–10. Internal Clock – These control the internal oscillator, which is output on **RAMP** (23). They have no effect on the sequence when an external signal is plugged into **X** (24), but the **RAMP** output can still be used as an independent LFO or VCO.

1. Rate – Controls the speed of the clock, from 4 minutes to 9Hz (**SLOW**) or 4Hz to 12kHz (**FAST**).

2. V/O Input – Controls the pitch of the internal oscillator. At lower frequencies, you can use this to precisely change the clock speed. Accurately tracks 7 octaves.

3. Rate Modulation Attenuator – Controls the depth of rate modulation.

4. Rate Modulation Input – Linearly modifies the rate of the clock. At audio rate, this is the same thing as frequency modulation.

5. Sync Input – When a pulse is received here (a transition from 1V or lower to 4V or higher), the clock immediately resets.

6. Sync Count – Sets how many pulses to count on the **SYNC/N** input before the clock resets.

7. Sync/N Input – Syncs the internal oscillator after receiving N pulses.

8. End of Cycle Output – When the clock resets or otherwise reaches the end of its cycle, this jack outputs a trigger.

9. Loop / One-shot Switch – When set to LOOP, the clock behaves normally. When set to ONCE, the clock will run from low to high, then wait for a sync from SYNC or SYNC/N to reset to low and begin the next cycle.

10. Rate Range Switch – Set the range of the internal clock to SLOW (4 minutes to 9Hz) or FAST (4Hz to 12kHz).

11–17. Map Section – This section controls the voltage that corresponds to each stage, as well as the gate output.

11. Gate Length – Sets the gate length from 6µs to 2.8s.

12. $f(X)$ Range Switch – Switches the range of the f (Xn) sliders, either −5 to +5V / 10 octaves (+ -), 0 to 5V / 5 octaves (+), or 0 to 2.5V / 2.5 octaves (+ / 2).

13. Stage Indicator – Shows which stage is currently activated. When no stage is activated (an EXT input has been received), all these LEDs are unlit.

14. $f(X_n)$ Sliders – Sets the f (X) voltage that will be output for the corresponding stage.

15. $f(X)$ Output – Outputs the stepped voltage for the current stage, plus the value of + Y.

16. $+Y$ Input – Precision adder input. This can be used to transpose the f (X) output, or to replace it when all stages are deactivated.

17. Gate Output – This output produces a 5V gate whenever a stage is activated or whenever a gate is received in EXT.

18–26. Event Section – This section activates stage 1–8 based on the voltage input to X (24).

18. Position / Length Mode Switch – Selects between positional mode, where each slider controls the exact position of the threshold, and length mode, where each slider controls the length of the interval between this stage's threshold and the next.

19. X Indicator – Lights according to the current value of X, where the top LED is +5V (or the value of ABOVE, if that is used), and the bottom LED is -5V (or the value of BELOW, if that is used).

20. Threshold Sliders – Sets the threshold for this stage. When the X input crosses this threshold, this stage will be activated.

21. Threshold Direction Switch – Depending on the setting of this switch, this stage will trigger when X rises above (RISE) or falls below (FALL) the threshold for this stage. When set to OFF, this stage will never be activated, but if in LENGTH mode, if the slider is not set to zero then the interval will still be present.

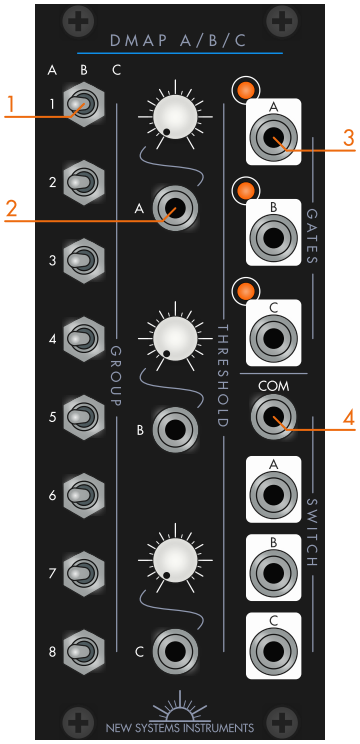
22. External Input – When a gate is received here, the current stage is cancelled and no stage is selected.

23. Ramp / Internal Clock Output – Outputs the internal clock. Normalized to the X input.

24. *X* Input – This input controls when each stage activates by crossing the X_n threshold. Normalized to the RAMP output.

25–26. Above and Below Inputs – Routing a voltage to the ABOVE input will set the uppermost value of the sliders. Routing a voltage to the BELOW input will set the lowermost value of the sliders. When using two or more Discrete Maps, plugging a cord from the ABOVE jack on the left/lower Discrete Map to the BELOW jack on the right/upper Discrete Map will join the two slider banks together. Note that the impedance will be very low when all the sliders are all the way down, and this might lower the voltage of the sources plugged into ABOVE and BELOW. Turning just one slider up a few percent should solve the problem.

A / B / C EXPANDER INTERFACE



1. Group Selection – These switches place stages 1–8 of the connected Discrete Map in groups A (left), B (center), or C (right).

2. Threshold – A signal to one of these jacks, attenuated by the adjacent knob, will add to the thresholds (the lower sliders) of every stage in the corresponding stage group.

3. Gates – When a stage in the corresponding stage group is activated, a gate is sent to these outputs. The gate length is controlled by the gate length knob on the connected Discrete Map.

4. Sequential Switch – The COM (common) jack will be connected to the A, B, or C jack when a stage in the corresponding stage group is activated. The switch is bidirectional, so COM can be either output or input. When no stage is activated, the switch is disconnected. While this switch will accommodate most Eurorack signals, it has a little less headroom than many other modules (8V vs the more common 9.5V), so take care with peaking signals or 0–10V control signals.

QUICK START

Variable-Length Step Sequencer: Set the clock to LOOP and SLOW. Set f (X) RANGE to + / 2, LENGTH mode, first switch to FALL and the rest to RISE, and all the lower sliders halfway up. Adjust RATE to fit your goals. Take the pitch output from f (X), controlled by the top sliders. Take the gate output from GATE and adjust GATE LEN. as needed. Changing the relative position of the lower sliders will change the relative length of each step. Tip: if you need to manually start the sequencer at a specific time, you can set the clock mode switch to ONCE, then flip it to LOOP when you want it to start.

Variable-Position Step Sequencer: Set up as previous, but set mode to POSIT. and set the bottom sliders in a diagonal line from bottom left to upper right. You can now move any slider to change when that step happens, without affecting the timing of other steps. Steps don't have to happen in order, so you can slide one step past another step to change the order.

LFO or Envelope Synced Step Sequencer: Set up as previous, but plug an external control signal into the X input, instead of using the internal clock. Change the bottom slider positions and select RISE or FALL in order to make each stage happen when you want it to, tracking the timing of the input waveform. For interesting variations, you can use an LFO with skew control, a wavefolder, an envelope follower, or other complex waveforms.

Quantizer / Harmonizer: Set to positional mode and set f (X) RANGE to + / 2. Take the output from f (X). Plug a stepped input into X, such as a sample and hold, or a V/O signal from a keyboard or another sequencer. Adjust the bottom sliders and switches to set up where to map a note *from* and the upper sliders to determine where a note gets mapped *to*. With a V/O input, this can be used to create harmonies or countermelodies. With a sample and hold input, this will map a random signal into a set of chosen pitches. Adjusting the slider position with sample and hold can set up larger or smaller regions to make it more or less likely that a given note gets chosen.

Graphic VCO: Set clock to LOOP and FAST mode and plug a pitch signal to V/O. Set to length mode, set each step to trigger on RISE, and set the bottom sliders halfway up. Set f (X) RANGE to + - and draw a waveform using the upper sliders. Take the output from f (X).

Waveshaper / Bitcrusher: Set to positional mode, set the first four stages to trigger on RISE and the last four to trigger on FALL. Set up the sliders as follows:

X1, f (X1), X8, and f (X8): 20%,
X2, f (X2), X7, and f (X7): 40%,
X3, f (X3), X6, and f (X6): 60%,
X4, f (X4), X3, and f (X3): 80%.

Provide an audio input to X and take the output from f (X). This is a two-bit bitcrusher (four possible levels). By adjusting the top sliders, you can adjust the map between input and output, reversing the direction and getting a wavefolding type of effect, or limiting the amplitude for a saturation type of effect.

16 Step Sequencer: Set up two Discrete Maps as for a variable-length step sequencer, above. Connect ABOVE jack of the the lower/left Discrete Map to the BELOW jack of the upper/right Discrete Map. Connect the GATE output and f (X) output of the lower/left Discrete Map to the EXT input and + Y input of the upper/right Discrete Map, respectively. Lastly, use a stackable cable or mult the GATE output of the upper/right Discrete Map, and send one of the outputs back to the EXT input of the lower/left Discrete Map. Take the output from the upper/right f (X) and GATE.

QUICK START (EXPANDER)

Gate Sequencer: Set up the corresponding Discrete Map as a step sequencer, and use the switches to assign the steps into groups A, B, or C. Use the corresponding gates to trigger rhythmic sounds or events.

Phase Modulation Oscillator: Set up the corresponding Discrete Map as a Graphic VCO. With all switches in group A, you can control the phase of the entire waveform with the A jack in the threshold section. Note that unlike conventional phase modulation, when a section of the waveform is moved past ±5V, it will stop being triggered entirely. But this can be avoided through the careful use of attenuation (or limiting/waveshaping). By selecting different groups for different stages, the phases of different parts of the wave can be controlled independently.

Temporal Mixer: Set the corresponding Discrete Map to LOOP and FAST. Set the lower section to LENGTH mode, set the first three switches to RISE and turn the remaining switches off. Set the first three sliders halfway up and lower the rest to the bottom. Set the first three switches of the A / B / C expander to A, B, and C. Put three audio inputs into the A, B, and C jacks of the SWITCH, and take the output from COM. The first three sliders on Discrete Map now control how much relative time each signal gets. When each slider is raised, that signal gets more time. When the slider is lowered, that signal gets less time. CV to the A, B, or C thresholds will change the amount of time each signal is given. Experiment with different clock rates and different relative pitches of the signals in A, B, and C. Use additional stages to temporally mix one signal twice for each cycle. If you have more than one A / B / C expander, you can temporally mix two signals per expander plus one extra by chaining together COM with the switch for group C and setting up the groups accordingly.

QUANTITY AND QUALITY, CONTINUOUS AND DISCRETE

> In musical intervals, a harmonic ratio appears in the quantitative scale of progression from one quantity to the next without this quantity having for itself a different relationship to the preceding and following elements of the scale than these elements have to their preceding and following elements. While the rising tones are always themselves further from the root, while numbers seem to become only more different through arithmetic progression, nevertheless a sudden return takes place, a surprising consonance comes forth, which nothing *qualitative* in the immediately preceding progression had prepared, but which appears as an action at a distance, as a relationship with something far away. This succession of merely indifferent relations, which neither change the previous specific reality nor ever constituted this reality, suddenly interrupts itself; and while in a quantitative regard it continues the same way, here, with a leap, a specific interval breaks through.
>
> —G.W.F. Hegel, *Science of Logic*, 1828, "Nodal Lines in Measure Relations, Remark," my translation.

In the world of music, as in the universe we live in, we can describe two kinds of difference, or change. On the one side, there's *quantitative* difference and quantitative change, things measureable with numbers. Frequency, for example, is a quantitative measurement. Bass sounds and treble sounds are both sounds, both the same sort of thing. Only their frequencies differ. When quantitative values change, it happens gradually—at least, if we "zoom in" to a short enough time scale.* That is to say, quantitative changes are *continuous*, with no gaps. Some pitch moves around. Some filter opens up and closes. Some length of time passes. To be sure, these changes are often *important*, and certain kinds of drone music explore the possibility of making music with nothing but quantitative differences. But not every difference is quantitative.

On the other side, there's *qualitative* difference and qualitative change, which is not itself numerically measureable. With the immense power of calculation in our world, modern thought has a hard time with this concept. Qualitative differences are either one thing or another, with nothing in between. For example, there is no clear way to measure the difference between a snare drum and a flute, there is no numerical scale between drumness and fluteness. Electronic music has often played with things like this, and nevertheless defined smooth curves between timbres. But each of these curves will be different, and none of them are really correct or incorrect. In music, qualitative differences appear as events, timbres, and so on. We can measure things *about* events and timbres, but we can't numerically measure those things themselves.

However, in the world, and in music, there are also relationships between these two kinds of differences. As I already mentioned, we can, in fact, measure things about qualitative differences—qualitative differences often imply quantitative differences. Similarly, a change in a quantitative difference can produce a qualitative difference. As Hegel points out in the quote above, maybe frequencies just keep going up or down, but there are points where one frequency is a major third above another frequency, and points where this interval is just not there. That is, because of the nature of physical reality, but mostly because of the nature of mathematics, there is a *character* or *quality* to certain kinds of quantitative relationships. This character appears at certain exact points in a continuous, quantitative progression. Thus, a change from one character to another involves an instantaneous leap from one number to another, with no continuous motion in between. These individual, special points are *quantized* or *discrete* quantities. In music, for example, we commonly use a

---

* Except, as far as we can tell, in the exchange of energy between electrons and light within an atom.

discrete set of frequencies—that is, a *scale* which is made up of *notes*—rather than the full range of continuous frequencies.

In fact, in the quantum physics of Louis de Broglie, we see quantized packets of energy (such as photons) for the same mathematical reason that musical ratios exist. Given a certain circular orbit and energy level, a electron matter wave can only occupy certain circular orbits without interfering with itself. The frequency of an electron can only increase by a major third, a fifth, etc., with nothing in between.* The universe is, in a sense, overblowing a trumpet, or using a synced oscillator.

The process of transforming a continuous quantity into a discrete quantity is known as *discretization* or *quantization*. This is the central task of Discrete Map. In order to implement it as flexibly as possible, quantization in Discrete Map takes place in two steps. First, the motion of a continuous quantity is transformed into the activation of a particular stage—quantity is transformed into quality. Second, that quality is transformed back into another quantity. The overall result is that an incoming, continuous signal is mapped to an outgoing, discrete signal.

In conventional Western music, instead of using this Hegelian perspective, people usually think about the relationship between quantity and quality in an older Aristotelian way, where instead of a real back and forth relationship (a *dialectical* relationship), there is only a hierarchy. This begins with the supposed genius composer, who generates ideas (qualities), which are transformed into a series of musical notes (the Western music notation paradigm), or even a sequence of musical phrases, each of which is comprised of a sequence of musical notes (the MPC/Ableton paradigm). These notes are assigned to preexisting voices (timbres), which then, finally, have some amount of quantitative control in the "expression" of the music.

This perspective has never fit well with modular synthesis, where generative and autonomous patches, sample and hold, and quantizers often create the notes and events themselves from the quantitative progressions of LFOs and random voltages. And yet, a large number of the modules we (ab)use to achieve these complex patches are still designed from a hierarchical musical perspective. Our gear is good at transforming quality into quantity, discrete into continuous, but it is not as good at the reverse. Sequencers and clocks, in particular, are difficult to control with modular events. While incredibly innovative sequencers exists, the original modular step sequencer paradigm colors their possibilities. In this paradigm, everything starts with an already-qualitative clock (the pulse is either there or it isn't), which qualitatively selects a stage, which finally produces a discrete pitch signal and a qualitative gate (event). There is no feedback from the rest of the system.

By transforming quantity into quality, continuous into discrete, Discrete Map closes the circle and allows a modular system to create music using a relational, Hegelian paradigm. Discrete Map addresses the whole problem at once, at the most abstract level possible. Any time you need a qualitative change or a discrete value—a note, a rhythm, an event, a timbre—Discrete Map is a tool to allow you to sequence that change as a precise response to any arbitrary quantitative change, whether that's the continuous passage of time, the motion of a random signal, the contour of another pitch signal, or any other signal you can extract from your modular system.

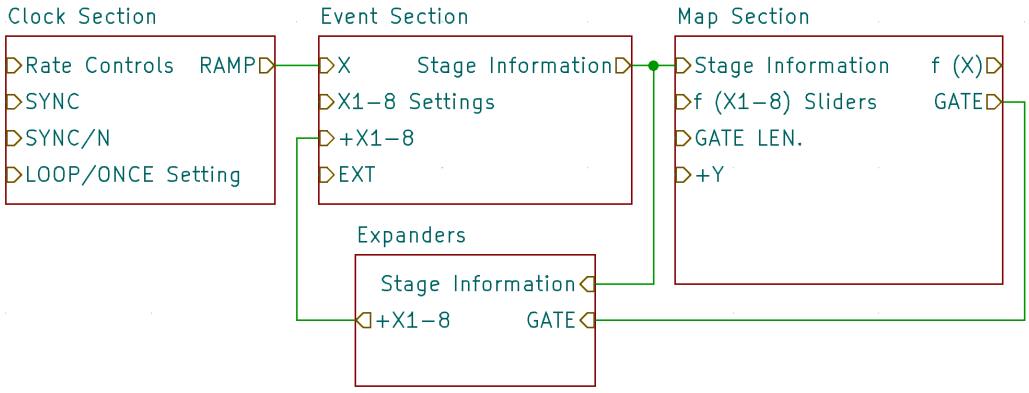
Discrete Map can certainly function as your only sequencer. However, Discrete Map can also bridge all existing discrete and continuous modules. Its internal clock is designed to produce a continuous quantitative signal that can correspond to a discrete series of events (such as a conventional

---

* This is currently beyond my pay grade, but I think the reality of De Broglian physics is a bit more complicated, as both the electron's speed and distance from the nucleus (hence the circumference of an orbit) as well as its frequency depend on its energy. Also, it should be mentioned that De Broglian physics is not the last word on the structure of an atom. Nevertheless, his theory heavily informs David Bohm's pilot wave theory, which is a current contender in interpretations of quantum physics.

clock) with the SYNC/N and SYNC controls. Its GATE output can be used as the clock that controls other sequencers. Rather than superseding existing sequencers and other modules, Discrete Map is designed to be the perfect tool to draw them into the relational sequencing paradigm and give them a new life there.

SIGNAL FLOW: THE THREE SECTIONS OF DISCRETE MAP



Signal Flow

Discrete Map is organized into three different sections: the clock section, the event section, and the map section.

The purpose of the clock section is to transform the passage of time into a signal that can be used to trigger particular stages. This signal is present on the RAMP output, normalled to the X input of the input section. Other than that normalled connection, the clock section is completely independent from the rest of the module.

The event section does the central work of the module, transforming quantative motion into qualitative events. It sets a number of rising and falling thresholds, and then activates one of eight stages as the voltage to X passes the corresponding threshold. The information on which stage is activated is then sent to the map section as well as to all connected expander modules.

The map section takes the stage activation information from the input section and transforms it into a sequence of discrete voltages and gate signals.

Expanders produce other kinds of outputs based on stage and gate information provided by the module. They can also provide a set of signals to adjust the input thresholds.

CLOCK SECTION

The clock section produces a ramp wave that moves evenly with the passage of time. When the clock is active, this signal begins at −5V and gradually continues to +5V. When LOOP is set, the clock automatically resets to −5V as soon as it reaches +5V. Conceptually, this represents some repeated length of time, whether that is a measure or a phrase. When ONCE is set, the clock stops just above +5V, and waits for some kind of synchronization signal.

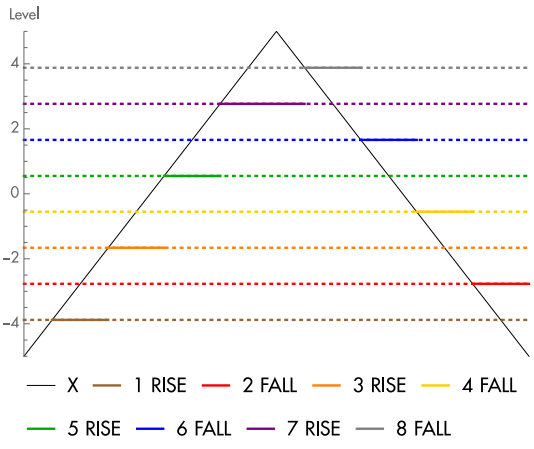
The clock can be synchronized in one of two ways. Any gate (a zero to 5V transition) on the SYNC jack will immediately reset the clock by setting it to −5V and starting the cycle. SYNC/N will count incoming gates, and reset the clock only when it reaches the count set by the corresponding knob. If both jacks are used, a signal to SYNC will also reset the count to zero.

The rate at which the ramp increases is controlled by the SLOW/FAST switch, the RATE knob, the RATE CV input and attenuator, and the V/O input. At audio (FAST) rate, this is the same as the familiar V/O, pitch, and linear FM controls on any other oscillator.

Interval	Closest Ratio
Minor 2nd	17/16
Major 2nd	9/8
Minor 3rd	6/5
Major 3rd	5/4
Perfect 4th	4/3
Tritone	17/12
Perfect 5th	3/2
Minor 6th	8/5
Major 6th	5/3
Minor 7th	16/9
Major 7th	15/8

At sequencer rates, note that the V/O input is useful as a way to precisely increase and decrease the speed of the clock. Any octave will raise or lower the speed of the clock by a factor of two. But as there is a relationship between musical intervals and mathematical ratios, pitch changes will have a predictable effect on the relative clock speed. With the usual equal-tempered tuning, the table above is accurate from 0.1% to 0.9%, depending on the interval. However, if just tuning is used, one can easily get exact ratios of any sort.

EVENT SECTION



Stage activations for a rising and falling input. Dotted lines indicate the thresholds for each stage, and solid lines indicate where those stages are active.

The event section sets up to 8 thresholds corresponding to the eight stages of Discrete Map. When the signal in X moves past one of these thresholds, the corresponding stage is activated and a gate signal is produced. These thresholds only work one-way, and will only be tripped by a rising or a falling signal moving past the threshold. When a signal moves past the threshold in the other direction, nothing happens.

There are two ways to set up the location of these eight thresholds: either by directly specifying their position, or by specifying the relative space between each threshold and the next.

Positional mode is activated by setting the mode switch to POSIT.. In positional mode, each threshold is set directly using the corresponding slider, with the lowest level at the bottom and the highest at the top. The slider position should roughly correspond to the LEDs on the left indic-

ating the current value of X. Note that there is no particular order to the sliders in positional mode, and stages are often activated out of order.

Setting the mode switch to LENGTH causes each slider to control the relative amount of space above each threshold. In length mode, the first threshold is always located at the bottom. The first slider then controls how much space is between the first and second threshold, the second slider controls how much space is between the second and third threshold, and so on, with the last slider controlling how much space is between the last threshold and the top.

In length mode, no matter what the positions of the sliders are, all eight lengths will always add up to the whole space. To put that another way, the sliders control *relative* length rather than *absolute* length. Sliders at the same position indicate the same length, while one slider that is proportionally higher than another will have proportionally that much more length. For example, if one slider is halfway up and another slider is all the way up, the slider that's all the way up will have twice the length as the other. The proportions between the two lengths would be the same if one slider were halfway up and the other a quarter way up, a third way up and a sixth way up, etc.

Turning a switch to OFF will keep that stage from activating, but in length mode the threshold and the length after it will still be present, effectively extending the length after the previous threshold. Setting the slider all the way down will set the length to zero and remove that stage from consideration.

The bottom and top of the space in which the threshold sliders operate is set by BELOW and ABOVE, respectively. When nothing is plugged into these, the bottom is −5V and the top is +5V. In positional mode, the bottom of the space and the top of the space correspond to the bottom of the sliders and the top of the sliders. In length mode, the bottom of the space corresponds to the first threshold, and the top of the space is at the top of the length after the last threshold.

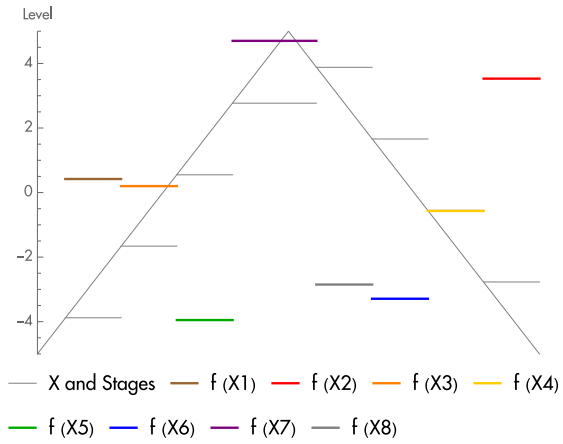
When in length mode, plugging ABOVE on one Discrete Map to BELOW on another one will join the two groups of sliders together, such that the relative position of all sixteen sliders add up to the total space. When in positional mode, connecting two or more Discrete Maps in this way evenly divides the space between them, with half going to each module (or a third if three are connected, and so on).

A signal to EXT cancels the currently active stage, such that no stage is active, and produces a gate signal. In order to keep EXT from being tripped by the gates produced by the module itself, EXT is disabled for a short time after any stage is activated, approximately 12.5µs. This limits the maximum frequency that EXT can handle at 80kHz. With 16 equal steps, this limits the maximum frequency of two interconnected Discrete Maps to around 5kHz. Note that the gate at EXT is reproduced, rather than mixed in with the other output gates. That means that it will be sized according to the GATE LEN. knob, but also that it will be delayed by a few microseconds. For particular audio rate applications, this might be undesirable, and you should instead use a gate mixer, such as Babel.

Technical note 1: In length mode, when all the threshold sliders are down, we end up with a semantic situation that doesn't really make sense: 8 sliders with a relative proportion of 0/0 somehow add up to the whole space. When this happens, ABOVE is essentially directly connected to BELOW and all the thresholds sit around the same spot in between the two. While there is current limiting to prevent damage even to poorly designed modules when ABOVE and BELOW are connected, the output voltages of any connected modules will likely droop, and may do other entirely unexpected things. Raising any slider just a little will make the interface semantics make sense again and provide plenty of impedance between any connected modules.

have a very low output impedance, essentially zero, and you don't need to think about this. The exception you are most likely to encounter is a passive attenuator. This will act like an extra stage between whatever is plugged into it and the top or bottom of the sliders.

MAP SECTION



The $f(X)$ sliders map each stage to a particular output value.

The map section works similarly to the sliders on a traditional sequencer. When a given stage is activated, a particular voltage is output on **f (X)**, with 0V output when no stage is activated (that is, after an **EXT** signal is received). The **+Y** input provides a precision adder to combine more than one signal into the **f (X)** input. The **RANGE** switch alters the range of the knobs between −5V to 5V, 0V to 5V, and 0V to 2.5V.

Every time a stage changes or an **EXT** signal is received, a gate is produced. The gate length is controllable by the corresponding knob and ranges from 6µs to 2.8S. If a new gate is triggered before the previous gate has stopped, the gate length timer is reset, and the gate continues over into the next step. This can be useful for constructing gates that are on for whole phrases, rather than individual notes.

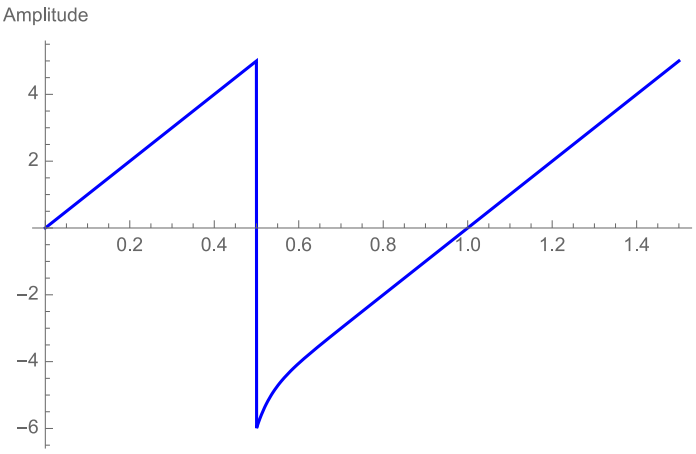
IMPERFECTIONS AND THE ANALOG IMPLEMENTATION

While the act of sequencing is in a sense inherently digital, Discrete Map contains no digital *values*, only digital *stages*. Before stage activation Discrete Map uses an analog oscillator to activate analog comparators, and after stage activation Discrete Map uses analog sliders to set analog levels. With analog, there is no such thing as perfectly equal values, and the slight differences between voltages at various points can lead to some results you might not immediately expect.

Imperfections in selecting analog output voltages will be familiar to anyone who has used any other sequencer with analog outputs. When dialing in an approximate output voltage, the visual slider position will approximately correspond, but if you need a more exact voltage, you are going to have to listen carefully and patiently nudge the slider to where you want it. Two different sliders will produce very close to the same output for the same position, but not quite exactly the same.

We are much less familiar with using sliders to control rhythm, but the same basic principle applies. Visual slider position is going to get you close, but the final adjustment should be done by ear. Our ear is almost as sensitive to timing as it is to pitch, and we can just barely make out the way two visual slider positions might not give the exact same time.

When we bring in the analog ramp wave as well, there's an additional inexactness to contend with: the mismatch between the top and bottom of the ramp wave and the top and bottom of the sliders. There is already an inherent ambiguity in setting a threshold exactly at the top or bottom. Should the wave activate at the top? That depends on whether you are thinking of thresholds as being activated when they are *reached* or when they are *exceeded*. Analog imperfections tend to turn this ambiguity into a dead zone near the top and bottom.



The ramp has a slight overshoot on fall, exagerrated here for clarity.

In length mode, where the first step is always anchored to the bottom, this dead zone would cause a problem. Consequently, when the Discrete Map clock resets, it briefly overshoots and extends below the bottom, ensuring that it is always activated. The timing as a whole remains accurate, but there is a barely noticeable delay between when a falling threshold and a rising threshold at the bottom would activate. There is no such overshoot at the top, and you may find a small dead zone there.

The short version: in positional mode, you may find a dead zone at the top. To get the value you're going for, you might want to have that stage activate on FALL or at the bottom on RISE. In length mode, having the first stage set to FALL rather than RISE produces the timing that most closely reflects the sliders. Nevertheless, you may find you need to sneak some sliders up or down from where they "should" be, particularly the last and the first.

There are a few intuitions you can develop that make the pitfalls of analog timing fairly easy to avoid. Try not to think of rhythm in classical Western terms—this mode of thought deals only with beats and divisions, and you'll end up thinking of the last beat as the "end." Discrete Map arranges rhythm in terms of position or duration within a total measure duration, and after each stage is activated it has to *stay* activated for some length of time in order to be useful. Additionally, the start of the measure is the same thing as the end of the measure, and this is generally best represented by the falling reset.

USING DISCRETE MAP WITH MIDI AND OTHER SEQUENCERS

MIDI provides several signals for sequencers to receive: START, CONTINUE, STOP, and CLOCK. Like most modular sequencers, Discrete Map is free running and so CONTINUE and STOP don't apply. However, START and CLOCK can be used to synchronize Discrete Map's clock system. Through a MIDI to CV interface, connect START to the SYNC input, and CLOCK to the SYNC/N input, and set the clock to ONCE. Usually, your MIDI to CV interface will have some way to set the number of pulses that correspond to the CLOCK output, and you can adjust the number of pulses that SYNC/N responds to accordingly. The module will stop when the MIDI transport has stopped sending a CLOCK signal, and start from a known state when START is sent. However, note that some transports send clock all the time, and Discrete Map will continue to respond to it as long as it is sent.

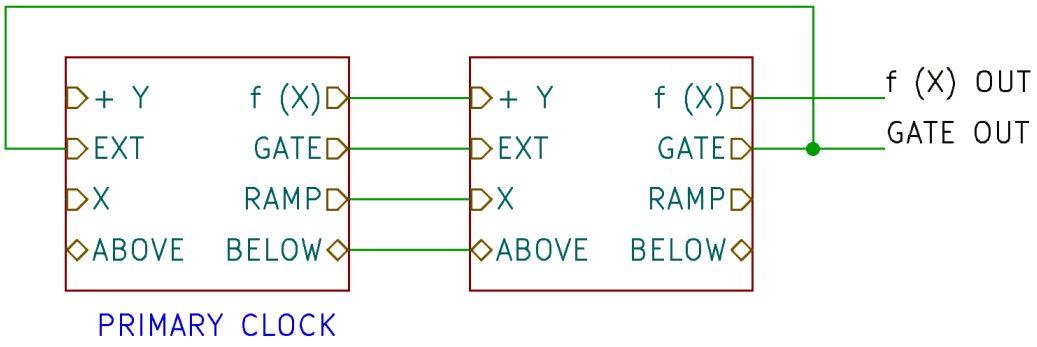
You can connect other sequencers in one of two ways, either by synchronizing the Discrete Map clock with the sequencer, or by using the Discrete Map gate output as the clock signal for the other sequencer.

To synchronize the Discrete Map clock with a traditional clock signal, you can connect the clock to the SYNC/N input and adjust accordingly. If you ever need to reset the count of the SYNC/N input (to line up the first stages, for example), you can send a gate to the SYNC input. If the other seq-

uencer provides individual outs or a start signal, you can omit the SYNC/N input and just use these. Note that using any of these clock synchronization schemes doesn't affect the rate of the clock itself, just the point at which the clock resets.

Instead of synchronizing the clock, you could use Discrete Map as the source for a clock that other modules will use. While a few modules might expect a regular clock signal, most of them will work fine with any irregular rhythm. The A / B / C expander makes this even more flexible, as you can use the gate from only select stages as a clock, ignoring the rest.

USING MORE THAN ONE DISCRETE MAP



Connecting two Discrete Maps to function like a single 16 step sequencer

Discrete Map is fitted with a number of features to make it easy to integrate into a full sequencing system of two or more interpatched Discrete Maps. Just as much, Discrete Map is designed with the flexibility to do this in more than one way. Discrete Maps can share an X input, combine their stages together, combine their f (X) outputs, and combine their threshold sliders. Additionally, they can trigger each other in the same way that they can be triggered by a different sequencer (see the previous section). One of these connections can be set up, all of them can be set up, or anything in between.

To share clocks, patch one RAMP output to the X input of the other Discrete Map. This sets them up as multiple independent mappings of the same basic time structure.

To combine the stages of several Discrete Maps into a single block, all the EXT and GATE outputs have to be connected together. When this is done, only one stage will be active on all connected modules. With two Discrete Maps, the easiest way to make this connection is to patch each GATE output to the EXT input on the other module. For three or more modules, use a gate mixer / logical OR (such as Babel) to combine all gates, then a mult to return the mixed gates to all EXT jacks.

To combine f (X) outputs, patch one Discrete Map's f (X) to the other's +Y. This can be continued for several Discrete Maps if required. When used in conjunction with stage combination, only one stage will be active and the voltage for that stage will be available on the unpatched f (X). Otherwise, the active stages will add together to produce new intervals. With clever timing, two Discrete Maps can produce up to 64 different voltages (80 voltages if EXT is also used).

Lastly, to combine threshold sliders in LENGTH mode, patch ABOVE on one Discrete Map to BELOW on the other. This can be continued for three or more Discrete Maps. In POSIT. mode, this patch is unnecessary, as all sliders in position mode already cover the whole range. However, if you want to split up the full range and make each Discrete Map responsible for a proportional fraction, you can make the connection from ABOVE to BELOW. For two Discrete Maps, for example, this

would divide the space in two, setting the range of the sliders of the first Discrete Map from −5V to 0V and the second from 0V to 5V.

As an alternate method, you can connect any number of Discrete Maps in a loop using the ONCE setting and the EOC jack. The EOC jack creates a pulse when a cycle is completed. By connecting the EOC jack of each Discrete Map to the SYNC jack of the next, you create a sequence of sequencers, with each one activating and running in turn.

A / B / C EXPANDER

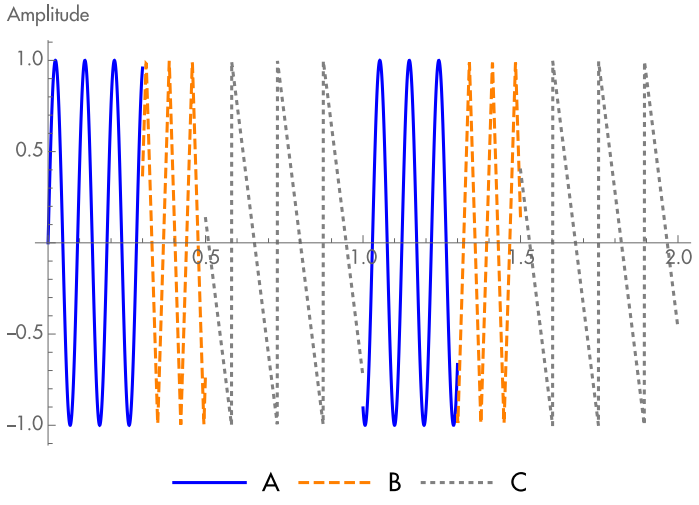
The A / B / C Expander module allows control over and response to a wide variety of events using a limited number of inputs and outputs. It groups each of the eight stages of the connected Discrete Map into one of three groups, A, B, or C. These stages can then be controlled and responded to as a single unit.

Patching a signal to one of the THRESHOLD inputs will add the incoming signal to the threshold control for all stages in that group. When more than one expander is connected, these threshold adjustments all add together. This adjustment is not affected by the LENGTH/POSIT. switch, and so steps can move past each other in either mode. When the corresponding Discrete Map is running off a clock, this has the effect of moving the corresponding step backward and forward in time. When operating at audio rate, this phase modulates the corresponding part of the wave. When the rest of the wave remains fixed, this has an effect similar to pulse width modulation. Note that moving a given step past the end of the RAMP voltage will effectively shut that step off entirely. As such, there is more space to modulate steps in the center of the wave than steps near the end, and it can make sense to construct your three groups accordingly.

All the gates for each group will be reproduced on the corresponding GATE output of each module. At audio rate, these produce a sort of irregular frequency division, which can be used as a signal or as an interesting sync input to another oscillator.

The switch section can be used to sequence the patching of your modular system. It is fully bidirectional and low impedance, so you can just as easily switch one output to three different inputs as switch one input between three different outputs. Using external voltage sources as the inputs effectively gives your sequencer a second f (X) out.

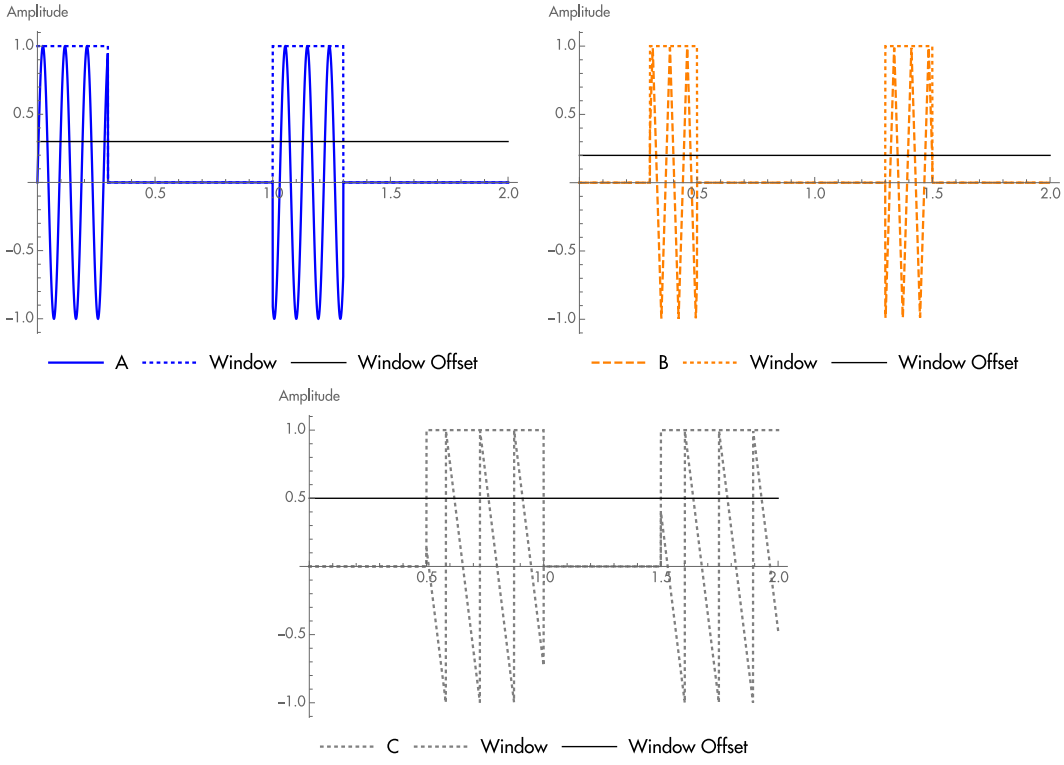
A / B / C: TEMPORAL MIXING



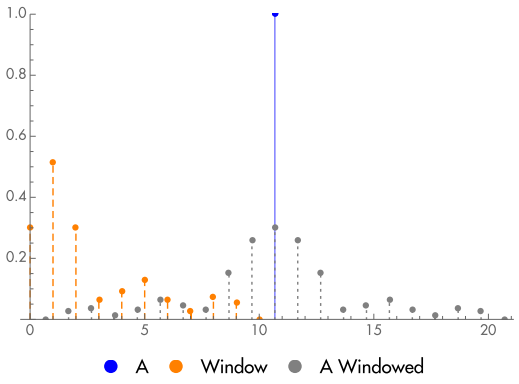
A sine, triangle, and saw wave mixed together via temporal mixing

When operated at audio rate with audio rate inputs to A, B and C, the A / B / C switch becomes a *temporal mixer*. Each period of the clock on the connected Discrete Map is divided into some number of sections, and each voice fed into the switch can be heard only during its section. Further, when the corresponding Discrete Map is in length mode, the sliders control how much time each voice gets, just like the sliders of a conventional mixer control how much amplitude each voice gets.

We can decompose a temporally mixed signal into a sum of several windowed signals.



The previous graph decomposed into three windowed signals.



The spectrum of the original A signal, the window, and the result when the window limits the time of the A signal.

Windowing is the same thing as amplitude modulation with a pulse wave that moves from zero to unity. That is, when the window is off, we multiply by zero, and when it is on, we multiply by one, and this is the same thing as multiplying by a pulse wave that varies between these two levels. This 0–1 pulse wave will have a DC offset proportional to the length of time it spends at 1. Consequently, the spectrum of the original waveform will be reproduced at a level that corresponds to the proportion of time for which it is present. However, as in amplitude modulation, additional harmonics will be created that correspond to the sum and difference frequencies between the harmonics of this pulse wave, running at the Discrete Map frequency, and the harmonics of the original signal, whatever they are. As with any kind of pulse width modulat-

ion, the spectrum changes symmetrically with the relative width of the pulse, such that a pulse that is nearly always on produces the same spectrum as one that is nearly always off. As this pulse becomes more narrow, the level of higher harmonics will tend to increase, while the lower harmonics will fall to a level closer in value to the higher. At the limit, these harmonics extend to infinity, but the value of each individual harmonic falls to zero.

TROUBLESHOOTING

The gate signal isn't triggering my envelope / sequencer.

There are two common reasons for this. First, if the gate length is longer than the length of any stage, the gate won't get a chance to go low before the next stage sets it high again. Just adjust the gate to be shorter. Second, the minimum gate length of 6μs is very small, as Discrete Map was designed to produce gates at audio rate. Some modules need as much as 10ms to respond. Simply adjust the gate upwards until your module consistently responds.

The $f(X)$ output isn't controlling my oscillator / quantizer.

While some modules will respond to any pitch signal, others will only respond to positive voltages, and some will have even more restrictive ranges. Select the + or + / 2 range.

The LEDs on the A / B / C expander don't blink.

The A / B / C expander uses the gate length on the connected Discrete Map to set the length of its own gate ouputs, including the time the corresponding LEDs are lit. At 6μs, the LEDs are energized so briefly that you won't get a chance to see them. Turn up the gate length.

I did a bunch of stuff with the gate signal then fed it back to EXT and now nothing works.

Discrete Map has no way to tell whether what it receives on EXT is someone else's gate or its own gate fed back to it. Consequently, it ignores a signal to EXT for about 12μs, then assumes any subsequent signal is from something else to which it will have to respond. When a sufficiently delayed copy of its own gate is fed into EXT, effectively every stage activation cancels itself. Your only choice is to reorganize the signal chain to shorten the delay or to remove the feedback.

When I string two Discrete Maps together, I get a big DC offset on the signal of the second.

This usually indicates that the EXT signal is not arriving. Double check that the gate times are sufficiently small for both modules to consistently produce gates, and check that they are correctly routed to each other's EXT input. If there are stages on the two modules that are activating simultaneously, this can also fool one Discrete Map into thiking the EXT signal is its own, and ignore it. Double check the lower sliders and switches, as well as the ABOVE and BELOW connections, if you are using them.

I can't get this em effing thing in time!

The burden of specification is always the price of freedom. Take a deep breath and proceed step by step. First, turn every stage to off. If you're using the internal clock, adjust it so it syncs at the time you want, and it runs long enough for you to have a good enough space to work in. ONCE mode can help, but you don't necessarily need to run the clock perfectly to the end. Now don't touch the clock. Next, turn on POSIT. mode and only one note. Adjust the position so that note happens at the right time. Continue, setting notes one by one. As a continuous-time sequencer, Discrete Map is always going to sound "organic," rather than mechanically perfect. This is usually good, but if your use case demands perfect timing, consider using another approach, such as freezing the Discrete Map output with a sample and hold.