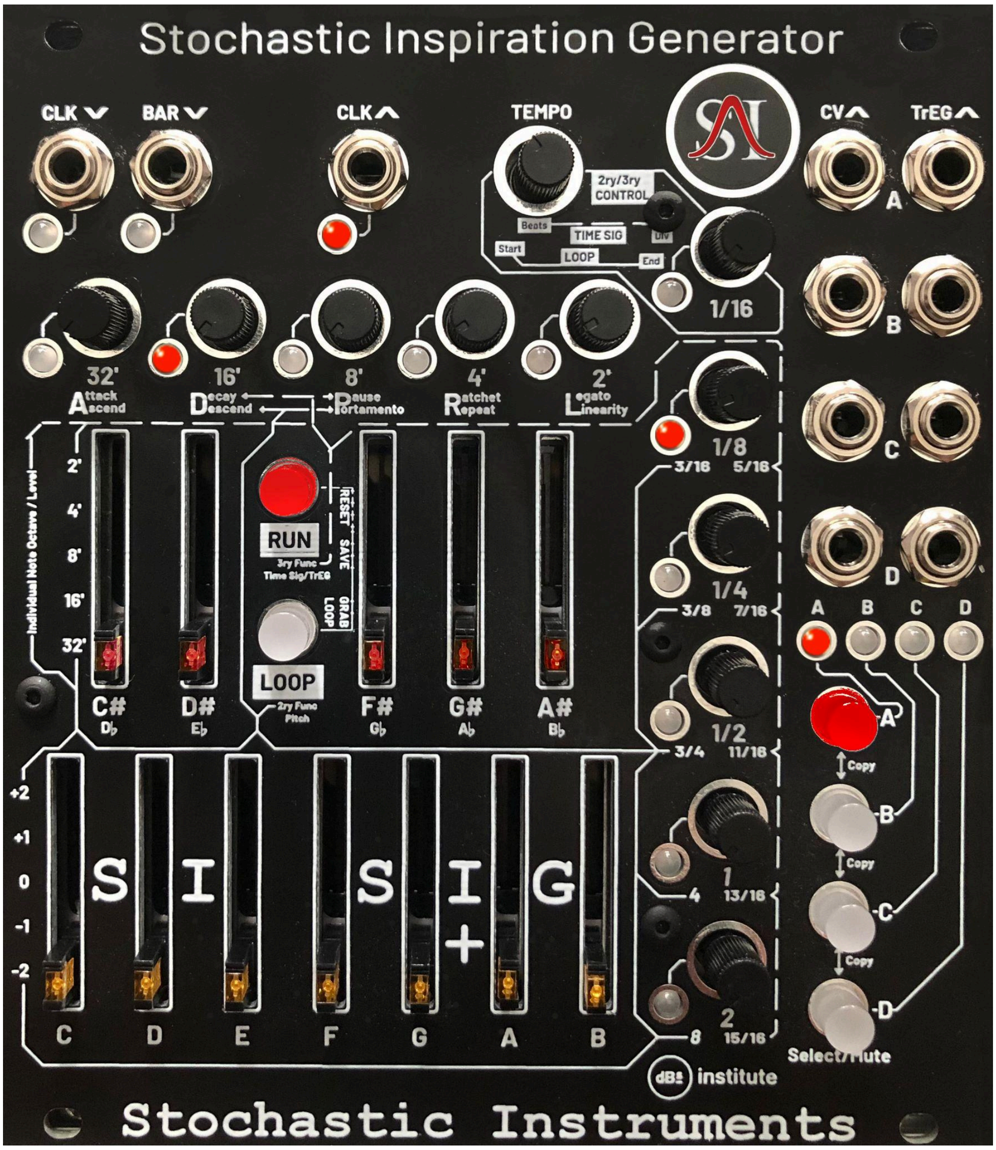
Stochastic Instruments
Rethink Random

Stochastic: An Inspiration Generator

Stochastic Instruments Ltd.

in association with

dBs institute



Rethink Random | ***Create*** Chaos | ***Perform*** Process

Installation and Safety

SIG & SIG+ use the standard Eurorack power connector with the red edge of the ribbon cable indicating -12V (supplied—however, always remember to check *any* power cable for faults).

Always ensure that the power is switched off before installing any unit into your rack or connecting/disconnecting the power cable. Also ensure you do not touch any electrical terminals when performing this process, and that your power supply is sufficiently current rated to serve all the modules connected to your bus board (use modulargrid.net to check this). Also ensure that you are statically discharged before installation, and avoid touching the rear circuitry.

Units sold before mid-2022 comprised a single channel Main Unit (MU) and optional 3-track Expander Module (EXP), until we consolidated these into the functionally identical 4-track SIG+ (see p.28). Users of the earlier MU&EXP version must ensure that the power is off while connecting or disconnecting the expander to the main board. **Hot-swapping Expanders can damage Eurorack board: *don't risk it!*** The EXP attaches to the main unit with one 20 pin ribbon (supplied) so it won't take up an additional power socket. Take great care attaching these, again, *with the power off.* Please also note, *old omsonic and new SI Expanders are not inter-compatible: omsonic boards employ two small ribbon connections and SI units use one large one so there is no way to cross-connect and thus no risk of accidental damage.*

Stochastic Instruments (SI) guarantees this product will be free of material defects and construction faults for a period of two years from the date of purchase if and only if bought direct from us. Malfunctions resulting from any misuse of the product, user modification of the circuitry, faceplate or firmware, the application of incorrect power supply voltages or connection of the power cable, design faults of third-party power cables or bus boards, or any other factors determined by SI to fall under the definition of 'user error' are not covered by this guarantee. SI still undertakes to service units under these conditions but at standard service rates. Your statutory rights are not affected.

For the period of the guarantee, defective units bought new from SI, whose fault is deemed by SI not to be caused by 'user error' will be repaired or replaced at our cost (including shipping to and from us), dependent upon presentation of valid proof of purchase from SI by the first owner. Units not bought from SI are not covered by this warranty. While we will undertake repair work, this will be charged, and the customer must cover postage to and from us.

SI implies and accepts no responsibility for harm to the user, or damage to equipment caused through any operation of this unit.

Background

Since the first Moog and Buchla models of the mid 1960s, sequencers have only been able to run linear series of steps in a loop, or address fixed steps in a random order. More modern Eurorack sequencers have embraced probability to a point, but these only go so far as controlling *if* an *already fixed* step in a linear sequence will fire. In September 2009, composer Phineas Head (HeadCell) turned this backwards and asked: **instead of fixing *steps* and randomising *progression,* what if we fixed *notes* and probabilized *selection*?**

By 2010 he had analogue circuit sketches but in the pre-boom Eurorack world affordable small scale PCB manufacture was not viable so it got shelved. By 2014 a functional MIDI version coded in Python existed, with a Doepfer Drehbank inputting control, and this worked extremely well. It was tested at the UK's prestigious dBs Institute music school, and used in some live work, but was again beyond what was feasibly manufacturable at the time as an integrated hardware unit.

A decade after its initial conception, modern rapid prototyping technology allowed Eurorack manufacturer omsonic to collaborate with dBs Institute's Modular Research Group to create an initial run of SIGs. When demand subsequently exceeded omsonic's manufacturing capacity, Stochastic Instruments was formed. The *Stochastic Inspiration Generator* is its maiden product.

What is Stochastic?

A stochastic process (/stəˈkastɪk/ from the Greek στόχος for '*aim'* or '*guess*') is one governed by controlled amounts of randomness. It's important then to differentiate SIG from other kinds of module: SIG is…

- **not a traditional sequencer** — it doesn't repeat itself (unless you tell it to!)
- **not a step sequencer with probability** — you don't program fixed steps that just vary
- **not a pure random source** — you can control what it will do (within limits)
- **not a chaotic source** — identical initial conditions don't produce identical output

…so we call it an *Inspiration Generator:* it responds to your probabilistic musical *direction* but you don't play 'on' it: you *duet with it* by starting with the fundamental musical building blocks and controlling how likely different note, duration, octave and transition events will be.

> *"The role of the composer is to prepare the elements which will permit the situation to become complex."*
>
> (John Cage, For the Birds: Sixth Interview)

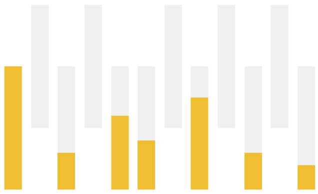
Overview

Congratulations on purchasing the Stochastic Instruments Stochastic Inspiration Generator!

With proper care, this unit will give you years of trouble-free aleatoricism.

Every event SIG generates has its note, octave, duration, onset, offset, portamento and ratchet conditions set by probability controls: you exert as much or as little control over what happens as you wish. This could be anything: from simple (set C, E, G and $^1/_{16}$ to 100% for a random arpeggiator), to complex (full 12-tone aleatoricism with 32640 possible outcomes for each new event) to highly tonal, melodic and performable (stepwise melody in C Natural Minor and real-time performable control over Looping and each note's octave and level).

SIG leverages tonal harmony by exploiting the statistical hierarchy of pitches: in any key some notes turn up a lot, some not so much, some not at all. Being 'in' C Major *means* having lots of Cs, Gs and Es, a few Fs, some Ds, As and Bs, but no C#s, D#s, F#s, G#s, A#s. A graph of the counts of each note in a piece in C Major would therefore look a lot like this:



Set SIG's note probabilities accordingly and it runs that logic in reverse by spontaneously jamming in C Major! Shift the probabilities and morph the tonality, key or mode they define into new tonal landscapes. Probablise both the *overall* octave range over which it improvises (to define register), and the fixed octave and volume offsets of *each* note (to define harmony). Bass notes provide root and inversion, so you can imply a change of C Root to C 1st Inversion by initially dropping only the Cs by 2 octaves and then only the Es, all in live performance. You can even set the likelihood that melodies will be more stepwise or more angular: *Stochastic thinks in music theory:* it knows the melodic and rhythmic parameters that shape a line and cut a groove.

SIG can control rhythm by setting the likelihood of different rhythmic values from an 8 bar drone to constant $^1/_{16}$ pulses, and also probablise up to 16 different note articulations. And if Stochastic strikes melody gold, capture your loop, set its start and end point and loop it, forwards or backwards, fully performably. Capture, edit and mangle, live, on the fly.

Finally, SIG can generate full 4 part probabilistic counterpoint in combination with its separate 4hp Expander, or with the later integrated SIG+ units.

Fully set across its 4 channels, after just 18 clocks, the possible range of material SIG could produce would exceed 3.85×10^{86}, 10000 times the number of atoms in the universe. *Boom*.

Operation Guide

Primary Functions

SIG's basic operation is very easy to grasp yet permits unlimited musical variations. Each event it generates is the combination of several probabilized parameters, and is indicated by a combination of **3 state LEDs** lighting to show the Note, Octave and Duration of that event.

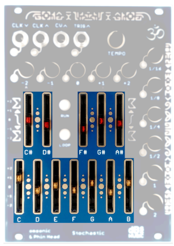
Tempo / Clock

SIG's internal clock runs 32–255bpm via the **Tempo** control and can also be clocked externally via **ClkIn** at 4PPQN (1 clk = 1 $^{1}/_{16}$)**. ClkOut** fires constant 5ms 5v triggers at $^{1}/_{16}$ notes while Run is on for both **IntClk** & **ExtClk**. **TrEG ^** can output **Tr**igs or **E**nvelope **G**enerator functions: *TrEGs*

Run Status	External Clock	Action
Stopped	Absent	Switches to **IntClk** after 0.5sec*, waits for Run
Stopped	Present	Switches to **ExtClk***,waits for Run
Running with **ExtClk**	Stops	Waits for next **ExtClk** indefinitely
Running with **IntClk**	Starts	Ignores, continues with **IntClk** until Stopped

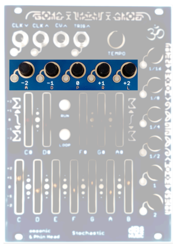
SIG can only switch clock when* ***stopped*. To detect and switch to ExtClk, clocks must occur at >2Hz (0.5sec = $^{1}/_{16}$ @ 30bpm). Slower clocks *whilst stopped* are not detectable so it will remain in IntClk mode. For slower ExtClks, multiply it *whilst stopped* and then divide it *once in Run*. **There is no clock limit in Run: Stochastic will just wait for the next one indefinitely.** If you stop, immediately remove the cable and press Run for IntClk in under 0.5sec, it remains in ExtClk mode and appears to freeze. Simply stop, wait 0.5sec and Run again to switch to IntClk operation.

Note Probability



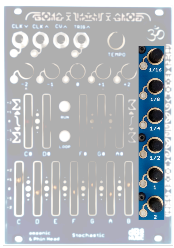
Note Probability is controlled by the 12 linear sliders arranged as a musical keyboard. The higher the slider, the greater the probability the next note will be the one designated. Pulling a slider fully back means that note will never be selected. Very low probabilities also affect whether a TrEG is sent (**Automatic Density**, below). Note output comes from the **CVout** jack, scaled to 1v/oct, ranging 0–7v and held constant until the next note is fired. See **Appendix 3** for typical scale key probabilities.

Octave Probability



Octave Probability is controlled by the horizontal row of 5 rotary pots marked -2, -1, 0, 1, 2. The more clockwise the pot, the greater the probability the next note, whatever it is, will be shifted up (+), or down (-) by 1 or 2 octaves, or remain central (0). In musical terms this is *15mb*/32', *8vb*/16', *loco*/8', *8va*/4', *15ma*/2'.

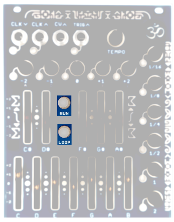
Duration Probability



Duration Probability is controlled by the vertical row of 6 rotary pots marked with note divisions. The further clockwise the pot, the greater the probability the next step will last for that duration. When a step fires Stochastic sends a trigger or envelope from **TrEG^** out.

Run & Initial State

Run toggles between **Stop** (LED off), **StandBy**, and **Play** (solid). On power up, Stochastic reads all the control positions but will only enter **Stochastic Play** when Run is pressed and at least one note probability is set greater than zero. It makes no sense for Stochastic to play if no notes are selected. With all Note faders at 0 and Run pressed, Stochastic is in **Standby**.



To Initialise, longhold Run-then-Loop. This zeros all notes, octaves, durations, loops and 2ry/3ry functions, and sets Octave and Duration defaults (0 and $^1/_{16}$). After Initialisation, Stochastic is in StandBy. As soon as you nudge a slider Stochastic Play will start in $^1/_{16}$ notes. This is closest to 'default' sequencer behaviour and the quickest way to reset and get started.

Control Sensing

Stochastic only senses control changes when Running so to 'drop in' with a complex randomisation press Run with all notes down, set your durations, octaves and Secondary / Tertiary (see below) functions and bring up the notes *last*: Stochastic will then begin to play with all your randomisations. Stochastic retains all settings between Primary, Secondary and Tertiary modes even though they use the same pots/sliders. New values are updated when returning to a previous mode when a control is nudged. Stochastic is about compositional spontaneity but can still save all settings past power down (see p24).

Default Octave and Duration

Octave and Duration controls work *relative to the default* of 8' Octave and $^1/_{16}$ notes, so on Initialise, every event initially generated has these values. Raising other Octave/Duration probabilities means these become more likely *but the default may still fire*—just less often. To completely defeat the default (e.g $^1/_{16}$) with an alternative (e.g ⅛) turn it up to 100%. Think of the pots representing their 'share' of the probability but the default initialises at 100%. Raising other pots distributes the share around proportionally so raising ⅛ to 50% shares it 50:50 with the default. Then, raising ½ to 25% shares it 25:50:25 and further raising it to 50% removes default $^1/_{16}$ altogether (0:50:50). *Manually* raising the default treats it like any other alternative.

Pauses / Rests

A rest occurs when the selected duration lasts without TrEG firing. Rests happen in two ways.

Automatic Density uses the note slider positions to infer note density. With all sliders at zero raising C slightly tells Stochastic you only want *some* Cs: a mixture of notes and rests. Raise C to maximum, you will get constant Cs with no rests. No matter how many notes you select, if they all have low probabilities, notes will be interspersed with rests. Higher probabilities result in fewer, and eventually no rests (= more, and eventually constant notes).

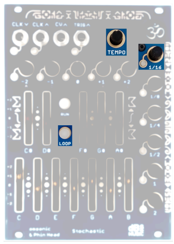
Pause (Run+Octave 0) sets the global probability of a pause / rest and overrides everything: if Pause is set non-zero, when it is randomly selected TrEG^ will not fire. At 100% Pause, Stochastic will be 'silent running' but is *still selecting notes and outputting CV* so you have a different behaviour to explore (by manually opening the VCF, say) than merely Stop.

Looping

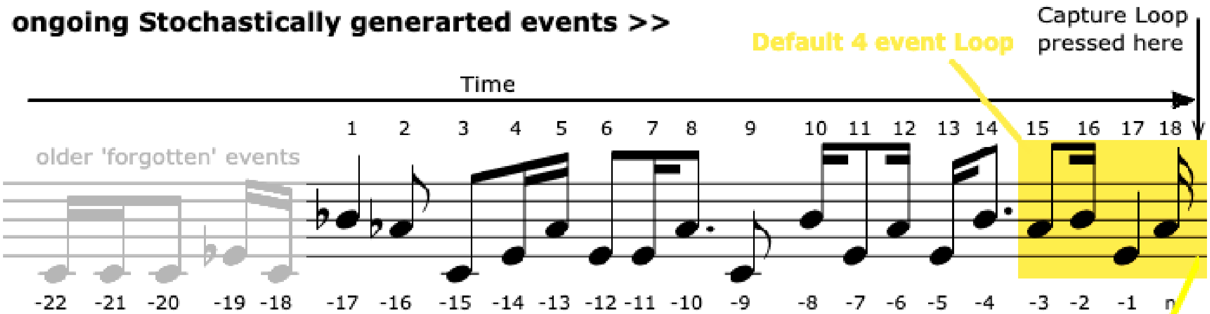
Stochastic can loop phrases like a traditional sequencer. However, *unlike* a traditional sequencer, Stochastic itself generates the music hence you don't pre-plan what is *going* to happen from a starting note *forwards*: you 'grab' a loop after hearing the *final* note which *completes* a musically useful generative phrase! This way of conceptualising composition comes quickly with practice.

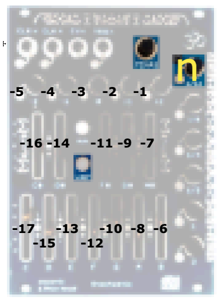
Quick-hold-Loop-then-Run captures the last 18 events into Stochastic's Loop Buffer. To edit the Loop, **Loop + Tempo sets Loop Start**, **Loop + $^{1}/_{16}$ sets Loop End**. Octave and Note LEDs light to represent looped events and **Longhold Loop latches Loop Edit** for single handed manipulation (Loop and Run flash). Press Loop again to exit latch: you can re-enter Loop Edit at any time. Setting **Loop End *before* Start plays the loop in reverse:** LEDs flash instead of light solid. **Single press Loop toggles between Loop Play and normal Stochastic Play.**

Loop-then-Run captures a new loop of stochastic events on the fly, whatever the current Play mode, to the current Loop Length/Start/End points. In Stochastic Play randomisation continues after capture: press Loop to hear the captured loop. In Loop Play, the new loop plays immediately on capture. Default *sounding* Loop Length is the last 4 events of the 18. You can then edit Loop Start/End to change the length but you must have captured a loop before you can edit these.

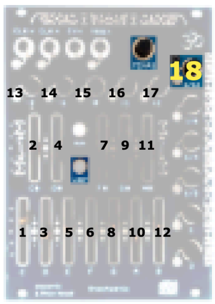


Events themselves are held as captured, so their parameters (e.g. Octave) are not editable.





Loop event numbering can be thought of as running *back* from the last event before Loop capture...



...or forward from the First event in the Loop buffer, 18 notes *ago*: whichever makes more sense to you!

Forward Loop:
(Octave 0, +1, +2 and $^{1}/_{16}$ LEDs light solid)

Reverse Loop:
(Octave 0, +1, +2 and $^{1}/_{16}$ LEDs flash)

Example: suppose Stochastic generates an 8-note phrase you want to loop and manipulate

1. Quickly press **Loop-then-Run** to capture the last 18 notes in memory. Stochastic continues playing as normal but Loop flashes to confirm the last 18 notes were captured.

2. **Longhold Loop to enter Loop Edit:** the default Loop Length of 4 LEDs displays. Turn Tempo fully *clockwise:* all LEDs switch off except $^{1}/_{16}$ indicating the most recent 'right most' **Loop End** event.

3. Turn Tempo *anticlockwise* to wipe *back* through the loop, right to left, from the last note *back* to the first (oldest) **Loop Start** note. Octave and Note LEDs light in turn indicating each previous event in the loop. All LEDs lit show the maximum 18 event Loop.

4. We want an 8 note loop: turn Tempo so 8 LEDs light from $^{1}/_{16}$ back to A#.

5. Turn $^{1}/_{16}$ to change the **Loop End** position: the 8 LEDs now shrink backwards. This is useful if you weren't quick enough to catch the loop in performance, but also acts as a powerful performance feature to live 'play' your loop in **Loop Play**.

6. Short press Loop **exits Loop Edit.**

7. Short press Loop again to toggle between Loop and Stochastic Play.

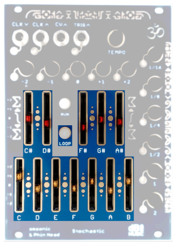
8. In Loop Play, live edit Start/Stop to chop and reverse the loop

9. Start/End and **Force Barline** (see below) interact very musically with the Loop, varying its rhythm but not notes, to provide 54,560 live-performable variations on your loop.

    - A Loop's implied time signature depends on the sum of all its durations: a 4 event loop is not necessarily in 4/4.

10. Hold Run+Tempo / Run+$^{1}/_{16}$ (or longhold Run to Latch) **ForceBarline** Edit and live remix the rhythm of your Loop in LoopPlay (see below).

    - Achieve 'beat repeat' effects by using 1/2, 1/4, 1/8, 1/16



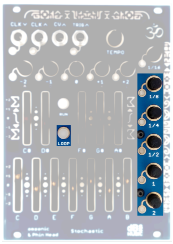
Secondary Functions: ***S***cale/Pitch Functions with Loop

To access Secondary Functions **hold or latch Loop and edit controls.**

Individual Note Octaves *(**Notes** + Loop)*

Holding Loop and changing a note slider transposes *only* that note by +/- 2 octaves (see the -2, -1, 0, 1, 2 left of the sliders). This is useful to imply harmonisation since lower notes carry greater harmonic weight. E.g. With all white notes up Stochastic is 'in' C Ionian (=Major) simply because C is the lowest note on the panel. Dropping D an octave changes the harmonic implication to D Dorian; dropping E would imply E Phrygian and so on. This is great for 'playing' Stochastic, reharmonizing its riffs with different bass notes. Remember that a fixed -2 on a note could still be 'cancelled out' by a probabilised +2 Octave but *on average* will be -2 Octaves low.

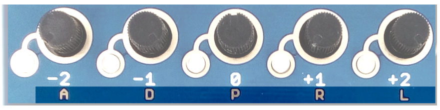
Dotted/Extended Durations *(**Durations** + Loop)*



$^1/_{16}$ Not used for duration (see Loop: Loop End Point)

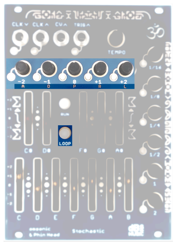
⅛ → dotted ⅛ = $^3/_{16}$
¼ → dotted ¼ = $^6/_{16}$
½ → dotted ½ = $^{12}/_{16}$
1 → 4 whole notes = $^{64}/_{16}$
2 → 8 whole notes = $^{128}/_{16}$

Achieve triplets by using dotted notes as straight eg. use dotted ⅛ (Loop+⅛) as straight ⅛ and straight $^1/_{16}$ will sound as triplets.



<u>A</u>scend *(**Octave-2** + Loop)*

Progressively skews Note probabilities so each event is more likely to be higher in pitch than the one that precedes it, causing the overall melodic contour to rise.



<u>D</u>escend *(**Octave-1** + Loop)*

Progressively skews Note probabilities so each event is more likely to be lower in pitch than the one that precedes it, causing the overall melodic contour to fall.

<u>P</u>ortamento *(**Octave** $_{3}0_{3}$ + Loop)*

Controls the probability that notes glide from the last pitch rather than step. CVOut ramps from its current value over the initial half of the new note, modelled on the classic TB303 slide.

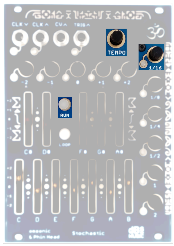
<u>R</u>epeat *(**Octave+1** +Loop)*

Controls the probability that the last event will repeat in all of its *probabilistic* aspects. 100% Repeat will 'freeze' on the note and 50–75% will 'cluster' durations, like live human players play.

<u>L</u>inearity *(**Octave+2** +Loop)*

Progressively morphs Note probabilities so successive notes are closer in pitch. Without Linearity note events are independent. Partial Linearity clusters them within octaves but still permits octave leaps. Maximum Linearity forces completely stepwise motion: no octave leaps occur but the melodic contour can still *traverse through* octaves, in stepwise motion.

Tertiary Functions: ***T***rEGger/***T***iming Functions with Run



To access Tertiary Functions, **hold or latch Run and edit controls.**

Force Barline *(**Tempo** + Run | **1/16** + Run)*

Force Barline Mode forces a new event to occur after a given **Time Signature**'s worth of clocks. Holding Run and turning Tempo will light 0 to 16 lights, left to right, bottom to top, across notes and then octaves. No lights is the default **Free Run Mode**, ranging up to 16 lights (the 5 octaves and 12 notes) and sets the note count (Time Signature top number). Run + $^{1}/_{16}$ sets the beat type (Time Sig bottom number) indicated by the respective duration. 4/4 would thus be Note LEDs lit to D# (4 top: 4 notes) and Duration LEDs lit at ¼ (4 bottom); 14/8 would be Notes and Octaves lit across all the notes and up to -1 Octave, duration at ⅛; and 3/2 Notes to D, duration at ½.

In **Free Run Mode** each new event will last its entire duration. This fluidity is part and parcel of Stochastic's character and perfectly compatible with both 'standard' and more experimental music, particularly if accompanied by a regular drum pattern. However, some genres consider a regular downbeat more of a necessity than others. If Stochastic randomised a $^{1}/_{16}$ as its first event on Run, followed by a series of whole notes, it would now effectively be displaced one $^{1}/_{16}$ ahead of the beat, not ideal for EDM with a heavy kick on every 4/4 downbeat:

```
Bars          |________________|________________|________________|__       etc
Clk           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^       etc
TrEG^ out     ||_______________|________________|________________|__       etc
```

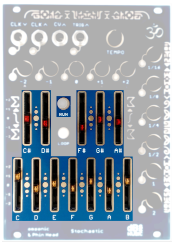
Force Barline Mode makes Stochastic re-randomise after a given number of clocks. If we selected Force Barline at 4/4 (= 16x $^{1}/_{16}$ clocks), it will cut short any event which runs over the bar, and will force a *new* event to fire to imply the barline e.g.

```
Bars          |________________|________________|________________|__       etc
Clk           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^       etc
ForceBar      ^                ^                ^                ^         etc
TrEG^ out     ||_______________|________________|________________|__       etc
```

The first $^{1}/_{16}$ still fires as does the following whole note ($^{16}/_{16}$) *except* Force Barline cuts it short so it only actually lasts $^{15}/_{16}$. The new event is thus heard as the downbeat of the next bar. **Longhold Run** displays current Time Sig and latches you in Barline Edit until you press Run again, permitting one-handed manipulation useful in Loop Play. Time Sigs are also independent per Expander track for your PolyMetric pleasure. **Appendix 1** shows common examples.

Force Barline with Loop: Loops are *n events* not *n clocks* long so a 4 event Loop of $^{3}/_{16}$ ¼ ½ ⅛ is $^{17}/_{16}$ and wouldn't fit in 4/4. **Force Barline** forces individual *events* to the bar so manipulating it live in **Loop Play** with **Loop Length** offers incredible scope for loop manipulation. Force Barlines to 1/4, 1/8, and 1/16 for Beat Repeat: notes remain fixed but their rhythms change!!

Force Barline can also be triggered ad hoc (SIG+ has a dedicated **Bar** input where 'original' SIG used a **TRS mini-jack** for ClkIn>TIP/ForceBarline>RNG). A trig forces a new event, permitting free external control over time signature (NB only works with an External Clock). See §Addenda for more details.

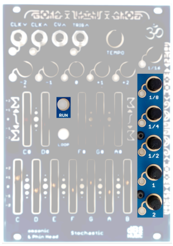


Individual Note Volumes *(**Notes** + Run)*

Holding Run and changing a note slider sets the voltage level of the TrEG^s for *only* that note, permitting accents on specific notes. Response is linear.

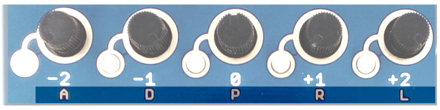
Irregular Durations *(**Durations** + Run)*

Prime number durations sound irregular in **Free Run Mode** and syncopated in **Force Barline**.



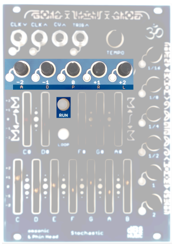
$^{1}/_{16}$ Not used for duration (see Force Barline: Time Signature Division)

⅛ → $^{5}/_{16}$	= ¼ + $^{1}/_{16}$ note
¼ → $^{7}/_{16}$	= double dotted ¼ note
½ → $^{11}/_{16}$	= ½ + dotted ⅛ note
1 → $^{13}/_{16}$	= dotted ½ + $^{1}/_{16}$ note
2 → $^{15}/_{16}$	= dotted ½ + dotted ⅛ note



Attack *(**Octave-2** + Run)*

Controls the probability that the next note event will have a slow attack. TrEG^ ramps up from 0v over the event, automatically scaled to duration. If this opens a VCF/VCA and the note sounds late, add DC offset or open the VCF/VCA slightly.



Decay *(**Octave-1** + Run)*

Controls the probability that the next note event will have a slow decay.
TrEG^ ramps down to 0v over the event, automatically scaled to duration.

Pause *(**Octave 0** + Run)*

Controls the probability that the next note event will be a rest (i.e. no TrEG^ will fire). This is useful to insert rests that override high note probabilities. Pitch CV will still be randomised.

Ratchet *(**Octave+1** + Run)*

Controls the probability that the next event will fire repeated $^{1}/_{16}$ notes over its duration. e.g. a ¼ note fires 4x $^{1}/_{16}$ notes at the same pitch, a dotted ⅛ fires 3x $^{1}/_{16}$ notes. Go Berlin School!

Legato *(**Octave+2** + Run)*

Controls the probability that the next TrEG^ will be a constant 5v rather than a trigger.

Attack/Decay/Ratchet/Legato combine multiply and independently to provide 16 complex articulations if all 4 controls are dialled in (Appendix 2 p20). Pause overrides all.

We have designed in *maximal versatility*: TrEG^ will trigger 'digital trig' modules (with an on/off response to Triggers/Ratchet), 'analogue trig' modules (vactrol LPGs that respond to Trigger voltage with A/D/L contoured Ratchets) and 'CV in' modules (VCAs/VCFs with the full range of A/D/R/L functions). This is why we often refer to ***TrEG***s rather than trigs. The raw Triggers themselves are actually 125ms linear A/D envelopes so Stochastic can mimic LPG functionality with VCAs/VCFs (see Specifications). If you need shorter square triggers, just externally trigger process the TrEG^ output. See Addenda for further info.

4 Track Expander/SIG+ (see also p28)

NEVER HOT-SWAP EURORACK EXPANDERS!

Stochastic is a completely new concept in sequencing so we've worked hard to cram maximum functionality into the minimum ergonomically performable footprint, for the lowest price. We want you to retain as much space in your rack (and wallet!) for other devices that play nicely with it. Equally, you expect multiple tracks from sequencing hardware which unavoidably increases price and footprint. We believe users should not be bound to large footprint/high cost units just to incorporate new concepts into their rig, so to balance these competing concerns, we've included the option to achieve full 4–track probabilistic counterpoint by adding the low-cost 4hp Stochastic Expander Module (EXP) powered form the main unit.

The SEM adds 3 further independent tracks, for a total of 4 tracks with the main unit. Tracks are selected for editing with the **Track Select** buttons on the expander.

Adding Tracks with Track Select

Tracks B, C and D can be 'overdubbed' at any time whilst running by simply pressing the new Track Select button. Stochastic switches to the new track while continuing to play any existing tracks. On the first press, the new track will be selected but Muted and in Standby (default Octave 0 and $^{1}/_{16}$ Duration are set, with no notes pre-selected). Set the track's notes, octaves, durations and $2^{ry}/3^{ry}$ parameters as desired and then press the track select again to unmute.

Confirm the track is playing by checking the corresponding TrEG^ LED. Tracks run to the common Clk but their settings, including Loop and Time Signature, are fully independent.

Full Initialise (Run+Loop) vs Track Initialise (Run+Loop+TrackSelect)

Longhold Run-then-Loop Initialises *all* tracks. **Shorthold Run-then-Loop-then-TrackSelect initialises a single track** (tracks B–D are initialised on powerup unless anything is in Memory).

Mute: Toggle Track Select

Press TrackSelect to toggle Mute Track. No TrEG^ will fire on the track until unmuted. Pressing Track Select before pressing Run can make the track or unit *appear* to be unresponsive, but it's merely Muted: repress and press Run. Red TrEG^ LEDs indicate track mute status.

Copy Track

Hold SourceTrack-then-DestinationTrack to Copy: e.g. hold A-then-B copies Track A to B. If source track is empty, nothing happens. Tracks run in Sync but have independent TimeSigs and downbeats so you can *perform* your drop-ins and create 4 Track PolyMetric madness.

Track Indicators

Orange Track Select buttons are blank if the track is empty, flash if programmed but unselected, and light solid if selected for editing. Again, check red TrigOut LEDs indicate for mute status.

Function Summary

	Primary Function	Secondary Function (+Loop)	Tertiary Function (+Run)
	Run **LED Off:** Stopped **LED Solid:** Running	**Initialise All** (*longhold* Run *then* Loop: all LEDs light) Initialises pr() on all Tracks to default	**Toggle Run** Press to toggle Stop or Standby/Run Longhold Latches Force Barline Edit
	Loop **LED Off:** Loop empty **LED Flash:** Loop loaded/*Stochastic Play* **LED Solid:** *Loop Play*	**Toggle Loop** Press to toggle Loop or Stochastic Play Longhold Latches Loop Edit (press Loop again to exit)	**Capture Loop** (quick hold Loop *then* quick press Run while Running) Overwrites existing Loop with new stochastic material
	Tempo Internal tempo 32–255 bpm	**Set Loop Start Point**	**Force Barline Beats** e.g **7**/4
Durations	**$^{1}/_{16}$ note**	**Set Loop End Point**	**Force Barline Div** e.g 7/**4**
Durations	$^{2}/_{16}$ = **⅛ note**	$^{3}/_{16}$ = **dotted ⅛ note**	**$^{5}/_{16}$ = ¼ + $^{1}/_{16}$ note**
Durations	$^{4}/_{16}$ = **¼ note**	$^{6}/_{16}$ = ⅜ = **dotted ¼ note**	**$^{7}/_{16}$ = double dot ¼ note**
Durations	$^{8}/_{16}$ = **½ note**	$^{12}/_{16}$ = ¾ = **dotted ½ note**	**$^{11}/_{16}$ = ½ + dot ⅛ note**
Durations	$^{16}/_{16}$ = **1 whole note**	$^{64}/_{16}$ = **4 whole notes**	**$^{13}/_{16}$ = dot ½ + $^{1}/_{16}$ note**
Durations	$^{32}/_{16}$ = **2 whole notes**	$^{128}/_{16}$ = **8 whole notes**	**$^{15}/_{16}$ = dot ½ + dot ⅛ note**
Notes	**Individual Note Probabilities**	**Individual Note Octaves** Set fixed octave offset per note	**Individual Note Volumes** Set fixed TrEG^ voltage per note
Octaves	**-2 Octave** *32' / 15mb*	**Ascend** Increases *pr*(next note higher than last)	**Attack** Sets *pr*(ramp up EG)
Octaves	**-1 Octave** *16' / 8vb*	**Descend** Increases *pr*(next note higher than last)	**Decay** Sets *pr*(ramp down EG)
Octaves	**0 Octave** *8' / loco*	**Portamento** Sets *pr*(next note glides from last)	**Pause** Sets *pr*(noTrEG^) = rest
Octaves	**+1 Octave** *4' / 8va*	**Repeat** Sets *pr*(next note repeats last)	**Ratchet** Sets *pr*(ratchet TrEG^)
Octaves	**+2 Octave** *2' / 15ma*	**Linearity** Increases *pr*(next note close to last)	**Legato** Sets *pr*(constant 5v sustain EG)
Expander	**Track Select / Mute** First press selects Track for editing Subsequent presses toggle mute Hold X then Y copies X to Y **LED Off**: Unselected & empty (default) **LED Flash:**Unselected but populated **LED Solid:** Track selected	**Initialise Selected Track Only** (*shorthold* Run *then* Loop *then* Track Select) Initialises all pr() on selected Track to default Track light goes out	

Function Index

Example Tutorial

1. Connect Stochastic's CVOut to VCO v/Oct CVin and TrEG^ to VCF CVin
2. Slave Stochastic's ClkIn to to a clock source or standard sequencer's ClkOut
3. Set up a 4-to-the-floor kick with the standard sequencer to act as a reference grid
4. **Initialise** Stochastic (longhold Run-then-Loop) and manually zero all controls
5. Raise $^1/_{16}$ Duration and 8' (or 0) Octave pots
6. Start Stochastic by pressing **Run**
7. Slowly raise **Note** C and notice how the $^1/_{16}$ become more dense as the fader rises
8. With C about ⅓ the way up, raise D# and G to the same for a random arpeggiator
9. Now bring in D, F, G# and A# to make a scale of C Natural Minor (albeit spelled wrong!)
10. Raise them all to maximum and note how there are no longer any gaps in the triggers
11. Imply a chord change to G# (= A♭) — hold Loop & drop G# down to -2 **Note Octave**
    a. Stochastic remembers the original position of G# so it still fires with fader down
    b. Notice how the drop in octave implies the harmony
    c. Bring G# back up to normal 8' Octave pitch (Loop+G# to centre) and drop A#
    d. Bring A# back up to normal 8' Octave pitch (Loop+A# to centre) and drop C
12. Now play with **Note Volumes**: pull down D, F, A# (Run+D etc.)
    a. Notice now how those notes recede into the background
13. Open out all the **Octaves**
    a. Notice how the 'melody' is now much more 'jumpy' and less linear
14. Make it more linear by upping **Linearity** (Loop & 2' Octave)
    a. Notice how the melody will be more stepwise again, while still spread out
15. Try 'steering' the melody with **Ascend** and **Descend** (Loop & 32' or 16' Octave)
16. Make some notes glide by adding **Portamento** (Loop & 8' Octave)
17. Bring in some longer durations
    a. Notice how now, against the 4-to-the-floor, events sometimes don't line up
    b. **Force Barline** to 16 (= 4/4) and notice how they now do line up
    c. Patch **Bar** in (SIG+) or TRS Ring of Clk (SIG) to force 4/4 from another sequencer
18. Bring in some **alternative durations**
    a. Bring in some **dotted** notes (Loop & ⅛ or ¼) and notice the triplication
    b. Bring in some **irrational** durations (Run & ⅛ or ¼) and notice the syncopation
19. Vary note transitions by altering **Envelope Function** (best with longer durations)
    a. Try just **Attack**: Run & 32' Octave to max
    b. Zero Attack and try **Decay**: Run & 16' Octave to max
    c. Zero Decay and try **Legato**: Run & 2' Octave to max
    d. Zero Legato and try **Pause**: Run & 8' Octave to max
    e. Zero Pause and try **Ratchet**: Run & 4' Octave to max
    f. Now set all Envelope Functions to 50% and hear how they **combine**!
20. Dial in 100% **Repeat** (Loop & +1)
    a. Notice how the note sticks infinitely
    b. Play with its Note Octave and Volume (Loop + Note or Run + Note)
    c. Bring Repeat back to around 25%
21. **Loop** (Loop-then-Run, set Length Loop+Tempo/Loop+$^1/_{16}$ and try reverse)
22. Beat Repeat in Loop by setting Force Barline to 1/1, 1/2, 1/4, 1/8, and 1/16
23. Capture Loops on the fly in Loop with successive presses of Loop+Run

Advanced Stochastic Cookbook

In addition to its primary role, Stochastic has many additional tricks up its sleeve.

- Using TrEG^ as *note* CV Individual Note Volume allows for **Arbitrary Voltage Mode** (e.g. microtonal CV values!). These could even be arbitrary pitches from a riff, represented by each 'semitone' and so permitting stochastically controlled melodic re-organisation!

    - Sending the TrEG^ through a quantizer to a second VCO creates an incredible **additional 'Pseudo-Track' for each track**. Each track's Attack/Decay sweeps then become scales or arpeggios. Alternatively, by setting all events to Legato, Individual Note Volumes become a second line of independent homophonic pitches, held for duration!
    - In combination with the Expander this creates the potential for 8 lines of music (as 4 homophonic pairs)!

- In Demand Mode, a single note varied from 0-100% probability can 'pass through' an External Trigger to TrEG^ like a **Probabilistic Trigger Skipper** or **Bernoulli Gate**.
    - Extending this idea, by using other durations you now have a **Probabilistic Clock Divider**.

- Stochastic can be used as a **Probabilistic Transposer** in combination with a conventional external sequence (in octaves, fifths or any semitone amount you wish via the note probabilities) by summing its output with the source sequencer through a unity gain mixer (we recommend the Tubbutec *Sumtiple* or Shakmat *SumDif*).

    - Combining a conventional external sequence's trigger (only) with Stochastic's Attack/Decay control gives you a **Probabilistic Envelope Generator** to apply to your otherwise conventional sequence.
    - Setting a faster stochastic melody and/or Loop on track 1, long drones on track 2 *and then unity summing these* lets SIG stochastically transpose itself across phrases of (repeating?) melody.

- Sending just TrEG^ to a drum module gives **probabilistic drums** with Kick, Snare, Open and Closed Hats on Expander ChA, B, C and D. Using CV to select sample gives incredible glitch drum effects!

    - Using **Force Barline** divisions in **Loop Mode** is an ideal way to achieve beat repeat effects, since each division is double or half its neighbour!
    - In your sampler, map sets of 5 samples in 'groups' based on octaves of pitch-classes (e.g. octaves of Cs, C#s, Ds etc.). You can then use Stochastic's semitones to select a sample group and the octaves to set the relative probability of each member of that group firing.

- Controlling other (non-pitch) functions with CVOut submits *any* voltage controllable function to probabilistic control. For example, the notes from C–B can become **Pan Location Probabilities** from left to right (C=hard left, F#=centre, B=hard right) if you feed the pan CV input of a mixer (e.g. WMD Performance Mixer).

- Stochastic follows any rhythm sent from another conventional sequencer on its ClkIn.

  - These do not have to be regular so Stochastic will shuffle and swing if you send it shuffled or swung clocks.
  - In addition, setting Force Barline to a non-multiple of an input *rhythm* (effectively counting *events*, not $^{1}/_{16}$) will cause it to reset **Iso-Rhythmically**. Ars Nova? *Techne Stochastica!*
  - While externally clocking, mult Stochastic's CVOut to your VCO's pitch and your External Clock's rate for some Cowell/Theramin **Rhythmicon** action!

- Externally clock Stochastic at *audio rates,* and treat its CVOut as *audio* for a gnarly **Probabilistic Noise Oscillator**!

  - Looping makes a more pitched sound: experiment with the effect of semitone, octaves, durations and Loop Start/Stop

- Flicking up and back single Note sliders 'plays' Stochastic like a **conventional keyboard.**

  - In keeping with our philosophy of a 'sequencer you can direct and play', you will also get used to 'steering' the melodic lines Stochastic produces by careful manipulation of the Ascend, Descend and Linearity functions.
  - Setting a single note to max will repeat the pitch indefinitely, but its Individual Note Octaves and Volumes remain playable in real time, permitting manual octaving, fades and 'freezes'.

- Because Stochastic always passes on its clock to ClkOut (be it Internal or External) several Stochastics can be daisy chained together.

  - Even when not in main use, Stochastic is a useful **Trigger Processor**, able to convert any leading edge into a 5ms 5v trigger from ClkOut.

- Setting max(Linearity) and max(Portamento) with all octaves and notes at 100% creates a Random Walk LFO module!

- Extend p9's triplet trick to quintuplets (Run+⅛), septuplets (Run+¼), undectuplets (Run+½), tredectuplets (Run+1) and quindectuplates (Run+2). Spicy.

- Experiment with 'pseudo Euclidean Rhythm' e.g. set 6/16 TimeSig with 100% ¼ durations to get a constant swing pattern. Which other combinations work?

- Duplicating 1 track to all 4 creates an ensemble of 4 'players' riffing together with the same constraints but constantly differing variations. Think Gaelic Psalmody at Berghain.

Technical Specifications

- Width: 18hp (unexpanded), 22hp (expanded)
- Height: 3U
- Depth: SIG+ 26 mm (with power cable inserted, measured from bottom of panel)
  - Original SI SIG seperate MU/EXP: 32mm

- Tracks: 1 + 1 psudotrack (unexpanded SIG), 4 + 4 psudotracks (expanded/SIG+)
- Internal tempo range: 32–255bpm
- External tempo range: arbitrary–500Hz (>2Hz trigIns required to initially lock to ExtClk)

- ClkIn TIP: accepts +/-12v, responds on leading edge 5v threshold
- ClkIn RING RevC SIG boards only: accept +/-12v, responds on leading edge 5v threshold
- ClkOut: 5v 5ms impulse
- CVout: 0–7v
- TrEGout: 5v 125ms exponentially decaying impulse
  - leading/trailing edge linear ramp, duration dependant if Attack/Decay set
  - full 5v DC if Legato set

- +12v current draw: 50 mA (unexpanded), 70 mA (expanded)
- -12v current draw: 30 mA (unexpanded), 45 mA (expanded)
- +5v current draw: 0 mA (unexpanded), 0 mA (expanded)

Credits and Acknowledgements

2009 conceptual design & software, 2019 functionality/panel layout: **Phineas Head**
Firmware design, 2019 hardware consultation, 2021 SI hardware design: **Stuart MacVeigh**

Development support: **dBs Institute Modular Research Group** www.dbsinstitute.ac.uk
2019 hardware & panel artwork: Fenn Marks / **omsonic** omsonic.co.uk

Manuals, Website & Project Management: Phineas Head
Initial 2009 analogue circuit consultation: Keith Wood
Early MCU consultation: Liam Grazier
Early Firmware consultation: Jake Shaw-Sutton

General Testing: dBs Institute undergraduates/postgraduates/staff
Beta Testing & Suggestion Credits: Matt Ward (ForceBarline refinement), Chris Pratt (Latching & ArbitraryVoltageMode), Perry Callaghan (manual refinement), Stuart MacVeigh (TrEG combinations)

UK production completion fabrication & testing: Dan Legg / Fully Wired Electronics

Stochastic shoutouts to the hero composers that inspired it: JSB WAM AFWS JMC KS SMR MLN MF BPGE
Phin shoutouts to Lucy, Madds, Marsha, Tony, Gizmo, Mr. Whizzah 🐾 [Save the World Entire: v.ht/vegan]

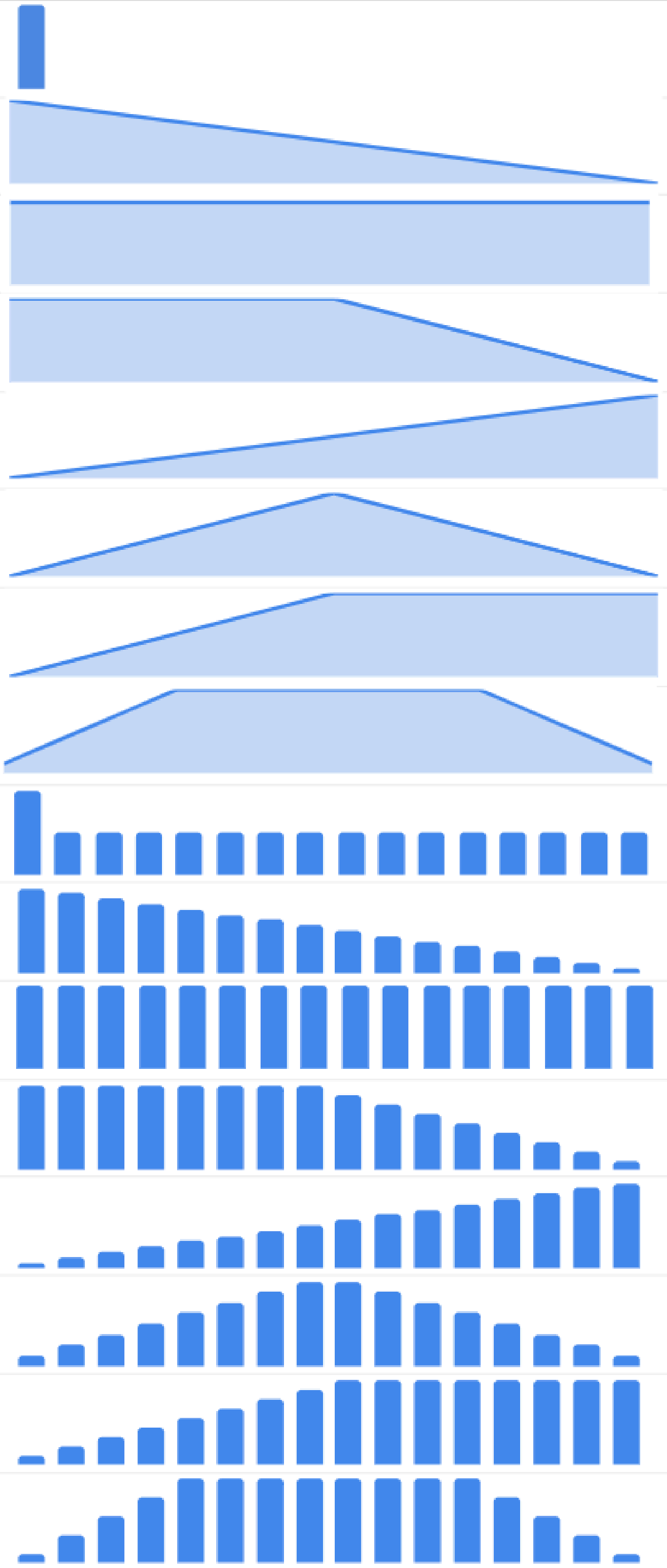
Special thanks to Olly Thomson & Nige Burt of dBs Institute for believing in the project.

Appendix 1: Some Common Example Time Signatures

Clock Count	Time Signature	Lights
0: Free Run Mode ***Default Power On Mode***	**None** **'Linear Meter'**	○ ○○○○○ ○ ○ ○○ ○○○ ○ ○○○○○○○ ○
1: Demand Mode ***Each ClkIn Forces New Event***	**1/16 or** **on ClkIn**	● ○○○○○ ○ ○ ○○ ○○○ ○ ●○○○○○○ ○
12	3/4	○ ○○○○○ ○ ● ●○ ○○○ ○ ●●○○○○○ ○
12	6/8	○ ○○○○○ ● ○ ●● ○○○ ○ ●●●●○○○ ○
14	7/8	○ ○○○○○ ● ○ ●● ●○○ ○ ●●●●○○○ ○
16	4/4	○ ○○○○○ ○ ● ●● ○○○ ○ ●●○○○○○ ○
18	9/8	○ ○○○○○ ● ○ ●● ●●○ ○ ●●●●●○○ ○
20	5/4	○ ○○○○○ ○ ● ●● ○○○ ○ ●●●○○○○ ○
24	14/8	○ ●●○○○ ● ○ ●● ●●● ○ ●●●●●●● ○
256	16/1	○ ●●●●○ ○ ○ ●● ●●● ● ●●●●●●● ○
512	'16/0.5' = 32/1 Longest available time sig, also useful if running Stochastic clock doubled (i.e. so $^1/_{16}$ sound as $^1/_{32}$)	○ ●●●●○ ○ ○ ●● ●●● ○ ●●●●●●● ●

Appendix 2: Ratchet/Attack/Legato/Decay *TrEG* Combinations

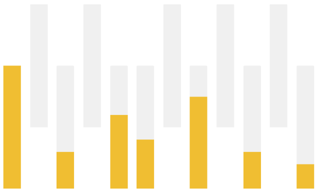
R Ratchet	A Attack	L Legato	D Decay	Envelope / Triggers
X	X	X	X	
X	X	X	√	
X	X	√	X	
X	X	√	√	
X	√	X	X	
X	√	X	√	
X	√	√	X	
X	√	√	√	
√	X	X	X	
√	X	X	√	
√	X	√	X	
√	X	√	√	
√	√	X	X	
√	√	X	√	
√	√	√	X	
√	√	√	√	

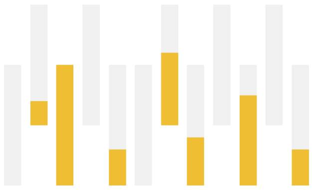


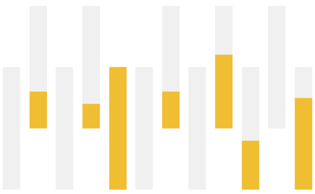
Appendix 3: Scale Probability Profiles

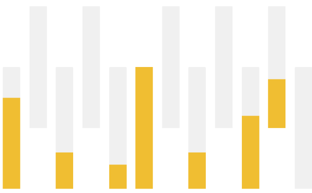
In all cases, transposing the root -1 or -2 octaves will add solidity to the feel of the mode.

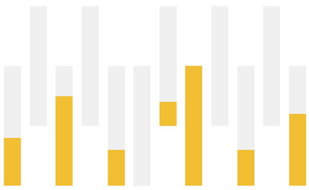
Major Scale Note Probabilities

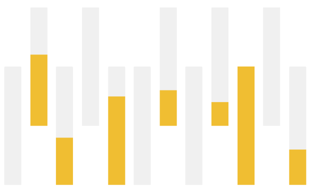

C

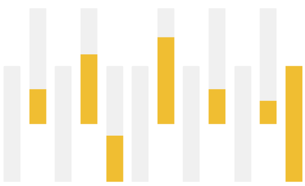

D

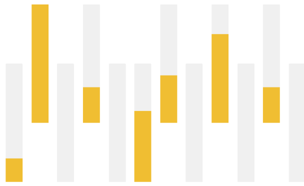

E

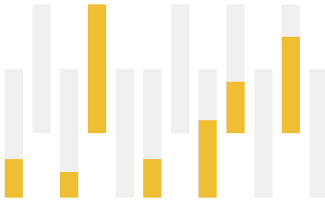

F

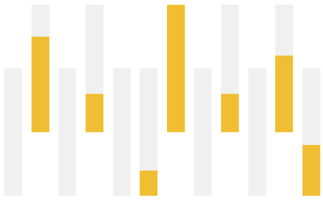

G

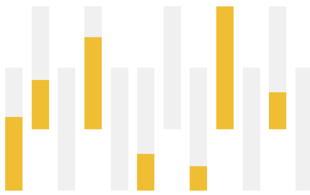

A

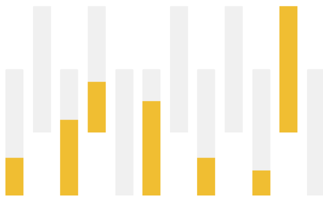

B


C# / D♭

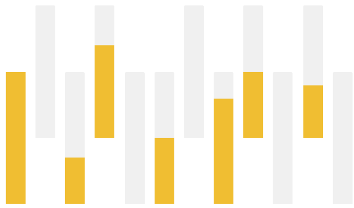

D# / E♭

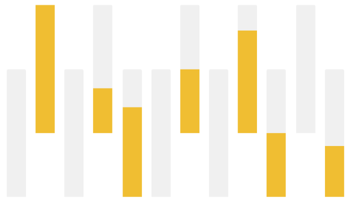

F# / G♭

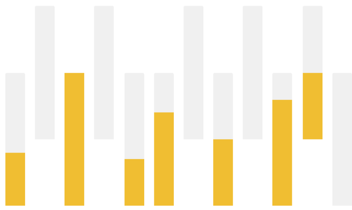

G# / A♭

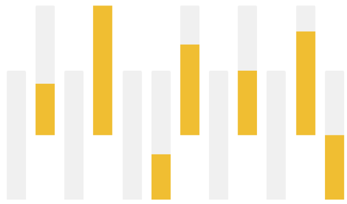

A# / B♭

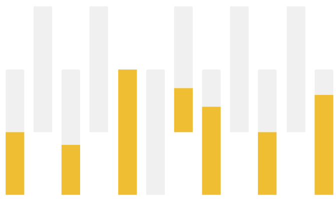
Natural Minor Scale Note Probabilities

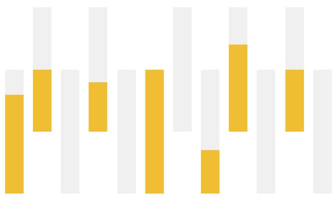

C

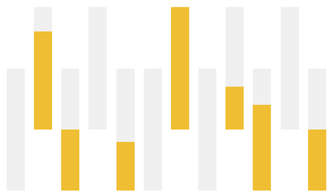

C# / D♭

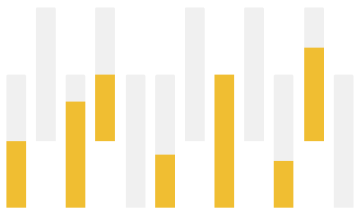

D

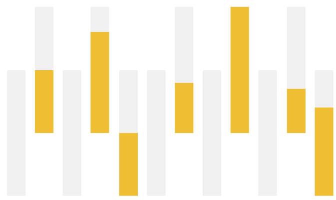

D# / E♭

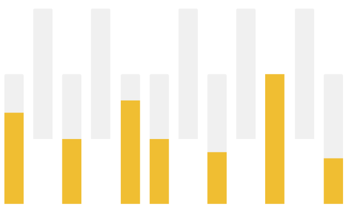

E

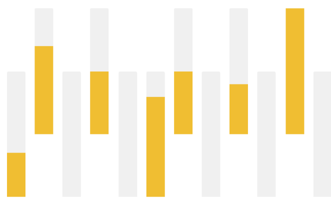

F

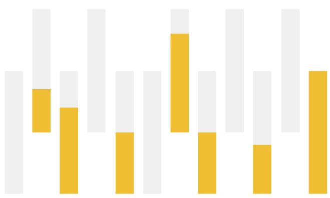

F# / G♭


G

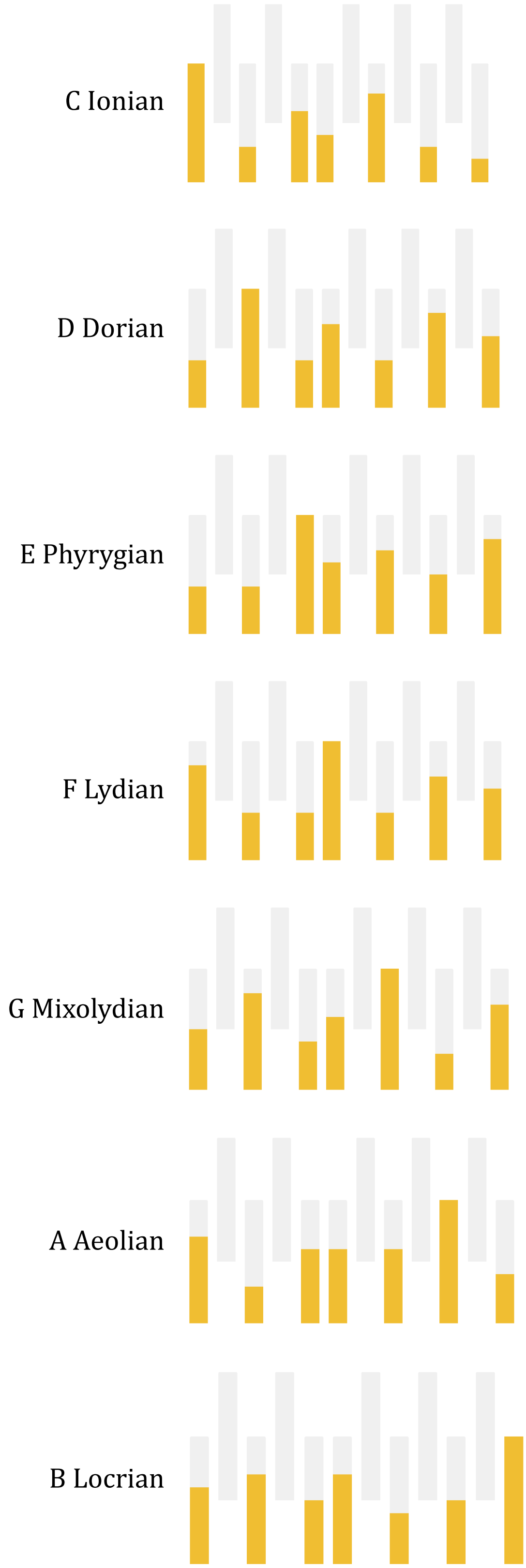

G# / A♭


A


A# / B♭


B

Suggested Modal Scale Probabilities



Addenda

Omsonic / Stochastic Instruments Ltd.

While omsonic were the original manufacturing partner for the Stochastic Inspiration Generator they were unable to meet the demand for the module. This precipitated the creation of the Stochastic Instruments Ltd. company, so from January 1st 2021 all Stochastic Inspiration Generators and their Expanders are made and supported by SI. In deference to omsonic's role in originally realising the project, we retain with slight updates his panel artwork under licence. omsonic continue to support all Stochastics sold by them before 1/1/21, while all SI Stochastics fall under SI's ongoing support: see https://www.stochasticinstruments.com/faq for details.

While SI Stochastics are basically functionally identical to the omsonic-made ones, our hardware redesign brings some important additions and upgrades: SI SIG/Expander units…

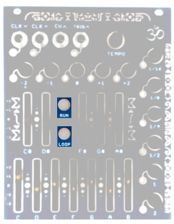
- have internal Flash Storage **Memory** to retain their settings after power-off
- have a TRS ClkIn jack to set an **Arbitrary Barline** on an external trigger
- are **32mm Skiff Friendly**
- use an **Analogue Noise Source** rather than a PRNG
- have **Centre-Dented Sliders** to better identify
    - 50% probability point
    - 'home' Individual Note Octave
- have full **16bit TrEG Outputs**

Store Settings to Flash Memory

SIG's design philosophy is first and foremost about *improvised performance and discovery* rather than traditional through-composed sequencing and pattern storage. However, we recognise that you will create amazing patterns with SIG that you might want to recall the next day. As such, SI SIGs allow the full storage of all the unit's settings and Loop material, on all tracks.

Remember that this is *not a performance feature*: SIG has been carefully and deliberately designed away from being a pattern sequencer so the concept is not to store and recall *in performance*. Rather, this is a way to *pick up your improvisation at a later date.* This means there is no 'recall' function per se: SIG simply returns to your last saved settings on power up.

To store all settings, longhold Loop-then-Run *while stopped.* Remember that doing this while running captures a new Loop so it's important to be stopped: again, this reinforces the idea that this isn't a performance feature.



All red accidentals will light (while the unit erases the last store), followed by all orange naturals (while the unit writes the new settings). The process takes about 2 seconds and you can resume normal operation immediately after. On power off all settings and Loops at the point you saved are retained, and immediately available when you power back on.

To perform an erase operation, first Reset (longhold Run-then-Loop) then Store.

CVOut Calibration (RevG & SIG+)

Board RevG adds a small blue multi-turn trimmer pot on the rear of the unit to calibrate CV out. RevG Expanders have 3 for their additional channels and SIG+ has 4, one each for its 4 channels. This is factory set so you shouldn't need to adjust but there is always variation in VCO input impedance so we added it to cover all eventualities. To adjust, Set 'C' ¼ to max with all Octaves at 0 except -2 and +2 at max. Attach a frequency counter to your VCO (or *listen!*) and adjust the micropot with a plastic or otherwise non-conductive screwdriver until you get a perfect 4 octave shift. Take care not to short out any of the circuit since the power will be on. We recommend sitting SIG upside down in a horizontal rack so there is no chance of slippage or shorting.

External Clock/Force Barline Trigger Timing

Some external clocking set ups can cause SIG to ignore External Force Barline triggers. Fortunately however there is a very simple trick to address this.

First, check your basic wiring setup is correct with these steps:

1. Power on
2. Reset (longhold Run-then-Loop) to clear any pre-existing settings
3. Plug a Y cable to ClkIn (SIG) or patch direct to Bar (SIG+) with a VCO/LPG from ChA CV/TrEG respectively
4. Send SIG an external clock
5. Press Run and dial in some notes
6. SIG should start to play 1/16s
7. Dial down all the durations
8. Turn up the '2' duration (the longest, bottom one) to max
9. SIG's events should immediately slow down to 2 bars of 4/4 (i.e. 32 clocks)
10. Now send SIG external Force Barline Trigs (SIG ring input of the Y-splitter, SIG+ Bar)

Result: the long events should cut short whenever you send the Force Barline trig. If they are not and you have a SIG (not SIG+) check is that you have the right kind of Y splitter— input 1 *only* goes to the tip and input 2 *only* goes to the ring. It won't work if both inputs go to both tip and ring (i.e. you need an 'insert splitter' or L/R splitter, not a 'headphone sharer').

If you are confident the connection is correct but SIG still ignores external barline trigs then the issue is easily addressed by checking the timing relationship between the two external triggers.

External Force Barline trigs need to fractionally lead the External Clock and cannot be faster than half the clock rate. The barline (or Clk ring on pre SIG+) input is sampled on each rising clock edge: lead time can be as little as 1ms and you shouldn't need more than 15ms. SIG samples the barline input within 1µs of the Clk so if the barline line is not asserted by then it may be missed. If the barline line is still asserted at the following clock edge, it will be processed then, so its duration can have an effect too. It's actually worth experimenting here because you can get some interesting effects! Most multi-channel clock modules offer per-channel delay so use one channel as clock and another as barline with a few ms 'positive delay' (i.e. phase lead). If you don't have a clock that can do that, you will need to apply a Trig Delay module to the *clock.*

Force Barline Overrides

SIG has been designed for maximal compositional and performative flexibility so there are some subtle but very musically important differences between how it treats *Internal* vs *External* Forced Barlines. Internal Barlines still work with both internal and external clocks, but External Barlines only work when SIG is clocked externally (otherwise SIG has no way to sync them to its internal clock).

Internal and External barlines are independent of each other so will both apply to your material, permitting more complex and interesting poly-metricism. However, External barlines take precedence and override everything. In addition, the (Internal) barlines of each track are also independent of each other (i.e. not sync'd to each other, although of course all sync'd to the clock), again permitting greater poly-metric complexity, but these can be 're-synced' with a single External barline trig.

One other point to consider is that Force Barline is designed to be more interesting / generative / Cagian / unique than just a hard 'reset'. It *will* act as a reset assuming there is no Internal Bar set and will 'sound' like a hard reset if the channels have a single Duration set (say, all 1/4). However, things get a bit more complex with a Loop: what it actually does is to fit the loop as it plays into the bar, on a *per event basis*. Because the Loop is made up of up to 18 *events* of *any* length, rather than beats, those events won't necessarily add up to a bar. Therefore, as it plays, if SIG gets a barline it only shortens the currently playing event in the loop and immediately fires the next event. In effect this is live duration remixing! If you really need the loop to repeat on the barline then you need to make sure it's exactly a bar long when you capture it. That is, if you want the Loop exactly as it is to Loop on the bar, it has to be the same length in clocks NOT events as the bar. In short, barline restricts the maximum length of loop *events*.

Finally, muting a track changes the metrical phase relationship between the muted track and the other tracks. For example, two track set to Internal 4/4 and firing ¼ notes could be mute-phased to be 1/16 out of sync. By carefully playing the mute, you can move the phase relationship of one to the other to line them up, or move them away…and then use a single External pulse to res-sync everything, thus 'Phase Muting' can become a performative control! Remember: SIG is an instrument!

Swing

While SIG does not have a dedicated internal swing control, swing is absolutely possible and there are three ways to achieve it. The simplest is to use SIG's External Clock's ability to just wait for the next Trig, which means you can send it a swung (or shuffled, or humanisaed…) clocks and SIG will just following exactly.

Second, SIG has two ways to achieve swing internally. You could set your durations to use only 1/16 and 3/16. While this won't reliably always alternate da-daa da-daa da-daa, it will have a kind of dotted swing 'feel' to it. For a more consistent da-daa da-daa da-daa result, you can set up two tracks in and set both Time Signatures to be identical, say 4/4. Provided you then only dial in one identical duration (say, 1/4 notes) on both channels, you can use Polymetric Phase Muting (see above) to time them so they are exactly 1/16 out!

Linearity Climbing

Linearity makes more melodic (as opposed to 'jumpy') steps likely. Depending on the pitches you select SIG can appear to prefer to naturally ascend, particularly if you have selected a spaced arpeggio rather than a scale. SIG will jump into the next octave up because the scale ‘wraps’ and so the nearest pitch may be in the next octave.

Suppose you had only C and G up. If SIG selects a G with high Linearity, the next C up is closer than the C below, so it will jump to the next octave. That only needs to happen a few times and the pitch will climb quite high. You work around that in three ways:

1. Ensure your VCO is tuned so an 8’ C from SIG results in an 8’ C from the VCO
2. Dial in more notes in SIG (the more notes, the less SIG will tend to climb)
3. Steer the pitch in performance using Ascend and Descend

Of these, 3) is probably the most important because again, SIG is designed to be *played.* Given the notes you have selected, you may want to dial in some constant A or D to compensate for any climbing (or falling) SIG is trying to do. Think of Ascend, Descend and Linearity as three interdependent but closely linked melodic performance controls.

VCA TrEG Onset

SIG’s TrEGs were tested across many different modules to match the ‘slew’ (the rate of change of voltage with respect to time) of the leading edge of SIG’s TrEG *output* when functioning as a standard ‘trig’ with whatever the *input* modules required to sound their best. We needed to find a slew rate that would trigger the digital Trigger In of a digital voice like Rings, the analogue Trigger/Gate inputs of a standard Envelope Generator, and 'raw' signal processors like VCFs/VCAs/LPGs. Because Eurorack has no ‘standard’ slew rate this is quite tricky to achieve.

Low Pass Gates typically have the ‘spongiest’ responses so too slow an input slew means they will lack their classic ‘pluck’ onset. Conversely, some solid state VCAs can respond so snapily to the same input slew rate that they can click. SIG’s TrEG *trig* slew characteristics were therefore carefully designed to accommodate all the modules we tested, deriving an optimal average behaviour fast enough to ‘pluck’ LPGs but not so fast that (our) VCAs clicked.

However, some VCAs just slew faster than others. The fastest will inevitably click if they are hit with a near-vertical trig and TrEG *trigs* may therefore open some VCAs quickly enough to manifest a click due to their particular internal electronic design, just as they would if controlled by a standard EG set to minimum Attack. In these cases, there are two simple solutions:

1. Run SIG’s TrEG output through a utility slew module just to take the edge off. That way, you can still use the probabalistic ASR characteristics but when SIG selects a 'hard' edge, it's not so hard that it clicks your VCA.
2. Check if your VCA can switch its response and test Linear vs Exponential vs Logarithmic

SIG+ (August 2022 onwards)

As discussed in the Expander section above, for the first 3 years of SIG's life we offered SIG separately from its Expander, not anticipating the near-complete take up of the full 4-track system, the numbers of people adopting it, and, latterly and less happily, the added manufacturing complexities the separate Expander necessitated in an increasingly challenging manufacturing climate.

We therefore took the decision to integrate the two units into a single combined module. Most important is to make clear that there is absolutely nothing SIG+ does that a standard SIG+EXP pair doesn't: it's running the identical Firmware on the identical chip, it's the same width (22hp = 18hp + 4hp) and we sell it for the same price as we would the old SIG & EXP.

SIG+ is identical in every way to a SIG & EXP pair:
firmware, form, functions and footprint remain fixed!

(To be absolutely accurate, there is one tiny difference: the external barline control has its own dedicated jack instead of being on the TRS ring of ClkIn, but the barline function is still 100% identical: you just no longer require a TRS Y-Splitter to access the input.)