Make Noise — MultiMod

- Manual PDF

Manual PDF

Make Noise MultiMod: using it for melodic patching

The attached manual is for the **Make Noise MultiMod**. It's a CV-processing module, not a sound source by itself, but it is extremely useful for building **melodic structure** from a single pitch or modulation source.

What MultiMod does in musical terms

MultiMod takes **one incoming control signal** and creates **8 related versions** of it at outputs 1–8. Those 8 versions can differ by:

- **phase**
- **speed**
- **time offset / delay**
- **read shape**
- **clock relationship** when synced to tempo

If nothing is patched to the input, it generates its own internal LFO/random shapes, which also makes it useful as a melodic animation source.

So for melody work, think of MultiMod as:

- an **8-way melodic variation generator**
- a **pitch canon / delay / phase tool**
- a **clocked CV reframer**
- a way to derive **counterpoint-like lines** from one sequence

---

The key melodic idea

Patch a melodic CV into **Signal In**, then send different outputs to different destinations such as:

- multiple oscillators' **1V/Oct**
- a quantizer input
- wavefolder or filter cutoff CV
- FM index CV
- sequencer transpose input
- sample & hold / track & hold destinations

Because each output is a related but altered version of the same source, you get melodies that feel **coherent**, not random.

Core controls and how they affect melody

1. Signal In

This is where you patch the source CV.

Best melodic sources:

- sequencer pitch row
- keyboard CV
- a slow envelope or function used as pitch material
- stepped random CV
- arpeggiator CV
- quantized modulation

If you patch a pitch sequence here, outputs 1–8 become variations of that phrase.

---

2. Time

Time controls the write/read speed and effectively the **length of the captured modulation phrase**.

For melody, this means:

- **short Time**: outputs closely track the incoming melody
- **long Time**: outputs become delayed, stretched, and more independent

Musically useful results:

- at short settings, you get tight melodic parallels
- at long settings, you get **melodic memory**, delayed imitation, and canon-like phrases
- when tempo is patched, Time becomes rhythmic subdivisions, which is very good for repeatable melodic relationships

A great use: - feed a sequencer into Signal In - clock MultiMod with the same master clock - use a longer Time setting - send outputs to multiple voices or transposition inputs

This creates staggered melodic echoes that remain tied to tempo.

---

3. Spread

Spread changes playback speed differently across the 8 outputs.

This is probably the most important control for melodic generation.

At **12:00**, all outputs are effectively same-speed copies. Moving away from noon causes some channels to run faster and others slower.

For melody, that gives:

- rhythmic displacement of notes
- pitch phrases arriving earlier/later
- different melodic contour rates from the same source
- evolving harmony if outputs drive several oscillators

Practical melodic effect: - channels near the center stay closer to the original phrase - outer channels diverge more dramatically

So if you patch outputs 3–6 to voices, you get subtler harmony. If you patch outputs 1 and 8, you get the wildest melodic deviations.

---

4. Phase

Phase offsets the copies relative to one another.

Melodically, this works like:

- **staggering note onset relationships**
- creating **rounds/canons**
- making one phrase feel like it chases another
- shifting identical pitch material into different harmonic moments

At Spread = noon, Phase is especially useful because all channels stay at the same speed, so the difference is mostly timing/offset rather than tempo drift.

This is ideal for: - multiple oscillators playing the same sequence with delayed entrances - one sequence feeding several quantizers or voices for phased harmony - generating “echoed” melodic entrances without using a delay effect

---

5. Shape

Shape determines how MultiMod reads the stored signal.

For melodic patching, each shape has a distinct compositional flavor:

- **Ramp / Forward Read**: most faithful melodic reproduction
- **Saw / Backward Read**: reverse melody
- **Triangle / Ping Pong**: melody moves forward then backward
- **Sine / Wow & Flutter Ping-Pong**: more elastic, less literal melodic motion

- **Square / Staircase**: stepped behavior, can become more gate/sequence-like
- **Stepped Random / Random Access**: reorders melodic values into controlled randomness
- **Smoothed Random Access**: gliding or semi-unpredictable motion, useful before quantization

For melodic clarity: - use **Ramp** first - then try **Saw** for reverse-countermelody - use **Orange** and **Yellow** with a **quantizer** after the outputs for generative pitch lines

---

6. Hold

Hold freezes current copies and loops them.

For melody this is excellent because it lets you:

- capture a phrase fragment
- loop that fragment as an ostinato
- create a temporary melodic vamp while the rest of the patch changes
- “sample” a good moment and keep it repeating

If your input melody is changing, Hold lets you grab a harmonic/melodic snapshot.

---

7. Reset

Reset re-aligns the channels.

This matters a lot for melody because with Spread active, channels drift relative to one another. Reset gives you:

- repeatable phrase starts
- barline alignment
- predictable harmonic downbeats
- controlled re-synchronization after divergence

In a musical patch, send a bar-reset or phrase-reset pulse to Reset.

Example: - master clock into Tempo In - every 16 steps send a reset trigger to MultiMod Reset - now your melodic divergences recur in structured phrases

That turns chaos into composition.

---

8. Tempo In

This is essential if you want melodic material to stay musically locked.

With a clock patched here:

- Time becomes quantized to clock divisions
- Spread relationships become grid-based
- Phase becomes tempo-grid aware
- Channel Index updates on each tempo clock

This makes MultiMod much more useful for **intentional melodic writing**, especially in techno, IDM, ambient sequence music, and polyrhythmic patches.

Best ways to use MultiMod for melody

A. One sequence, many voices

Patch:

- sequencer pitch CV → **Signal In**
- MultiMod outputs 1–4 → **1V/Oct** of four oscillators
- same gate pattern, or variations of it, to envelopes/VCAs
- master clock → **Tempo In**
- periodic reset pulse → **Reset**

What happens: - all voices derive from one melodic source - Spread and Phase create related but offset pitch lines - the result feels like composed ensemble writing

This is one of the strongest uses of the module.

Tip: - keep Spread low for consonant voice-leading - increase Phase for canon-like entries - use Time to set phrase length

---

B. Sequence to quantizer for generative counterpoint

Patch:

- smooth or stepped CV source → **Signal In**
- outputs 1–8 → multiple channels of quantizer, or one selected output to a quantizer
- quantizer outputs → oscillator pitch inputs

Why this works: - MultiMod creates related CV contours - quantization turns them into scale-constrained melodies - different channels produce different scale-degree paths

Best shapes: - Ramp for coherent variants - Saw for inversion-like reversal feeling - Orange/Yellow for more generative melodic behavior

This is a very strong ambient/generative technique.

C. Canon / round machine

Patch:

- pitch sequencer → **Signal In**
- Spread at **12:00**
- Phase increased from minimum
- outputs to several oscillators or separate voices
- common clock to Tempo In

- reset every 8 or 16 steps

At Spread noon, the channels are same-speed copies; Phase offsets them in time. This creates: - rounds - delayed entrances - melodic overlap - self-harmonizing lines

This is probably the cleanest “melodic composition” use in the manual’s feature set.

D. Melodic transposition network

Instead of patching outputs directly to oscillator pitch, patch them to:

- transpose input of a sequencer
- quantizer root/shift input
- oscillator FM amount
- filter tracking CV
- wavefolder symmetry
- chord voicing CV

Now the same melodic source produces higher-level changes: - phrase transposition - changing modal center - chord motion - timbral melody

This is often more musical than sending all outputs directly to 1V/Oct.

E. Use internal LFO/random mode as pitch source

With nothing in Signal In, MultiMod generates internal shapes.

For melody: - set Shape to **Stepped Random** or **Smooth Random** - patch outputs to a quantizer - quantizer to oscillator pitch - sync with Tempo In

Now MultiMod becomes a multi-channel melodic generator.

Great for: - generative lines - semi-related bass + lead + countermelody - evolving tonal clusters

Shape suggestions: - **Pink** less for pitch, more for clocks/gates - **Orange** excellent for stepped melodic material - **Yellow** excellent for gliding pre-quantized movement - **Sine/Triangle** useful for arpeggio-like scalar motion once quantized

---

F. Triggered envelope as melodic source

A very musical trick from the manual's logic:

Patch: - function generator / envelope output → **Signal In** - the trigger that fires that envelope → **Reset** - MultiMod outputs → quantizer or oscillator pitch CV destinations

Why this is good: - every triggered event starts from a known point - the contour of the envelope becomes pitch material - outputs become coordinated melodic gestures - repeatability is much better than free-running modulation

This is a strong way to make "played" melodies from envelopes rather than note sequencers.

G. Shift-register-like pitch spreading

The manual explicitly suggests a "Shift Register" style patch.

Patch: - pitch CV to Signal In - Spread at noon - Phase fully CCW then increase it - outputs to several oscillators' 1V/Oct

This creates spaced versions of the same melodic information across multiple voices.

If the source is a clocked sequencer, also: - patch related tempo to Tempo In - keep Time short for more precision

Result: - stacked melodic delay lines - pseudo shift-register harmony - clustered canon textures

---

H. Use Channel Index Out for melodic selection logic

Channel Index Out outputs 1–8V depending on which channel currently has highest amplitude.

This is not directly a melody output, but it can drive:

- a sequential switch
- address input on a selector
- logic/comparator network
- quantizer preset selection if supported
- octave switching or register changes

That means you can derive a second layer of structure from MultiMod: - one set of outputs creates melody - Channel Index chooses which melody/timbre/voice is active

This can create evolving melodic form from a single source.

Specific melodic patch recipes

1. Self-harmonizing lead

Patch: - 8-step pitch sequence → Signal In - master clock → Tempo In - outputs 2, 4, 6 → precision adders or oscillators - Shape = Ramp - Spread slightly off noon - Phase moderate - Reset every 8 steps

Result: - three voices with related but offset pitch contours - good for Berlin-school and generative polyphony

---

2. Bassline plus countermelody

Patch: - sequencer CV → Signal In - out 4 → bass oscillator 1V/Oct - out 7 → lead oscillator 1V/Oct - out 2 → filter cutoff or wavefolder on lead - Tempo In clocked - Shape = Saw or Triangle

Result: - bass follows source more closely - lead feels like a reversed or mirrored relative of the bassline - timbral CV stays compositionally linked

---

3. Quantized random ensemble

Patch: - no input to Signal In - Shape = Orange or Yellow - Tempo In from master clock - outputs 1–4 → quantizer inputs - quantizer outputs → four oscillators or one oscillator via switch

Result: - related generative melodies - each voice has its own timing/ speed/phase behavior - great for ambient, aleatoric, or modular minimalism

---

4. Canonic chord bloom

Patch: - keyboard CV or sequencer CV → Signal In - Spread at noon - Phase turned up - outputs to three oscillators tuned similarly - different envelope timings for each voice - common quantizer if needed

Result: - one melody blooms into overlapping entries - rich suspended harmonies emerge from repeated scale tones

---

5. Melodic ratchet/transposition helper

Patch: - main sequence CV → oscillator directly - also mult that CV into MultiMod Signal In - one MultiMod output → quantizer → transpose input of sequencer or precision adder - another output → envelope decay or filter CV - reset each bar

Result: - the “main melody” remains recognizable - MultiMod adds phrase-level transposition and ornamentation around it

---

Important limitations for pitch use

Because MultiMod is a DSP CV-copying/reshaping tool, it is not necessarily a strict precision pitch processor in the same way as a dedicated precision adder or buffered mult. So for best melodic results:

- use a **quantizer after MultiMod** if you want exact notes
- use **Tempo In** for repeatable relationships
- use **Reset** often for phrase structure
- keep **Spread modest** when you want tonal coherence
- use **Ramp** shape for the most faithful melodic copying

If you want experimental pitch behavior, then loosen all of those rules.

Best companion modules for melodic use

MultiMod pairs especially well with:

- **sequencers**: provide source pitch material
- **quantizers**: make outputs scale-accurate
- **precision adders**: combine outputs with fixed intervals
- **multiple oscillators**: create harmony and canon
- **function generators**: envelopes as pitch contours
- **clock dividers/multipliers**: structured resets and tempo sync
- **sequential switches**: select between output channels for phrase variation
- **sample & hold / T&H**: capture moments from drifting outputs

---

Best overall melodic strategies

If your goal is melody rather than abstract modulation, the most reliable workflows are:

1. **Input a sequenced or shaped CV**
2. **Clock MultiMod with Tempo In**
3. **Use Ramp shape first**

4. **Quantize the outputs**
5. **Reset on phrase boundaries**
6. **Use Spread sparingly at first**
7. **Use Phase for canon-like separation**
8. **Increase Time to create delayed melodic memory**

That turns MultiMod from “complex modulation gadget” into a very powerful melodic composition tool.

Summary

The Make Noise MultiMod is excellent for melodic work because it can derive **8 musically related CV lines** from a single source. Its strongest melodic uses are:

- **canon / round generation**
- **multi-voice harmonization**
- **clocked melodic delay**
- **related generative melodies through quantization**
- **phrase looping with Hold**
- **bar-structured resynchronization with Reset**
- **internal stepped/smooth random pitch generation**

In short: if you feed MultiMod one good melodic idea, it can turn that into an entire family of lines for bass, lead, harmony, transposition, and timbral motion.

Generated With Eurorack Processor