Robaux — DCSN-3

- Manual PDF

Manual PDF

Robaux Decision Tree — using it for melodic components

The attached manual pages describe the **Robaux Decision Tree**, a trigger/gate routing and clock-divider module. It is **not a pitch generator by itself**, but it is very useful for building **melodic structure** when paired with oscillators, quantizers, envelopes, sequencers, shift registers, sample & hold, or precision adders.

What the module does

At its core, the Decision Tree:

- takes a **gate/trigger** at **input a**
- routes that trigger **randomly** to:
- one of the three main outputs **b / c / d**
- then to one of the related suboutputs:
    - **e f g**
    - **h i j**
    - **k l m**

This creates branching rhythmic paths.

It can operate in: - **Mono/Mono** - **Poly/Mono** - **Poly/Poly** - and their **Latch** versions

It also has: - a **random ↔ repeating 16-step pattern** control - a **new random sequence** button - an **auto-reset / hidden reset** behavior - **clock divider modes**: - Classic Divider - 2/3/5 Divider - Spread

Why this matters for melody

Melody in Eurorack is often built from several layers:

- **pitch source**: sequencer, random voltage, S&H, Turing machine, keyboard, quantizer
- **timing**: clocks, triggers, rhythmic gates
- **articulation**: envelopes, accent, ratchets, note length
- **structure**: variation, phrase resets, call-and-response

The Decision Tree is strongest in the last three categories: - it decides **when notes happen** - it creates **branching phrase structures** - it can generate **different melodic lanes** - it can create **repeating or semi-random phrase skeletons** - it can divide one clock into multiple related time layers

So, together with other melodic modules, it becomes a very powerful **melodic organizer**.

Best uses for melodic patching

1. Branch one pitch stream into multiple melodic voices

Patch idea

Use one pitch source and let the Decision Tree decide which voice gets played.

Patch: - clock or trigger sequencer → **input a** - same pitch CV source → several oscillators or voices - outputs **b/c/d** → different envelope generators or LPGs for 3 different voices

Result

Each incoming note event is sent to a different voice. If each voice has: - a different timbre - different octave - different quantizer scale - different filter setting

then a single pitch line becomes a **distributed melody**.

Musical effect

- call-and-response
- hocketing melodies
- pseudo-polyphony from a single source
- generative ensemble lines

This is especially effective in **Mono/Mono** mode if you want one clear note event per step.

---

2. Create melodic branches with separate pitch processors

Instead of only routing triggers to voices, route them to **different pitch-generation chains**.

Patch idea

- master clock → **a**
- outputs **b/c/d** trigger:
- sample & hold 1
- sample & hold 2
- sequencer advance
- suboutputs **e–m** trigger:
- quantizer change
- transposition source
- octave switch
- glide envelope
- accent envelope

Result

Different branches of the tree can produce: - different notes - different octave ranges - different scales - different note lengths

This turns the module into a **melodic decision engine**.

Example

- **b** triggers a bass-note generator
- **c** triggers a midrange melody
- **d** triggers a high-answer phrase
- lower outputs **e–m** trigger ornaments, grace notes, transpositions, or alternate quantizer inputs

---

3. Use repeat/random morphing for phrase evolution

The manual says knob **n** smoothly adjusts between: - **completely random** - and a **repeating 16-step pattern**

This is excellent for melody because it gives you a continuum between: - generative unpredictability - stable motif repetition

Musical application

Patch a regular clock into **a**, and let the outputs trigger: - a quantized random CV source - a step sequencer reset/advance - envelope triggers for note articulation

Then use knob **n** to move between: - evolving melodic fragments - recognizable looping phrases

Why this is powerful

A lot of modular melodies fail by being either: - too random to remember - too repetitive to stay interesting

The Decision Tree is designed to sit between those extremes.

---

4. Generate melody plus ornament layers

Because there are 12 outputs arranged in a hierarchy, you can assign different outputs to different melodic functions.

Suggested structure

- **b/c/d** = main note lanes
- **e/h/k** = accents
- **f/i/l** = grace notes or ratchets
- **g/j/m** = transposition, reset, or harmony events

Result

You can create melodies where: - some events trigger the main pitch - some add ornament notes - some trigger higher-octave doubles - some trigger secondary voices

This is a good way to create **musically legible complexity** instead of pure randomness.

---

5. Use latch modes as a melodic switch

The latch modes hold the signal until a new input appears. The manual specifically notes this makes the module good as a **random switch**.

For melody, this is extremely useful.

Patch idea

Use the module to hold one selected route active, so one melodic branch stays selected until the next trigger changes it.

Patch examples: - send one CV source into several switched destinations - hold one drone note active - keep one transposition lane enabled - select one of several sequencers for a phrase

Melodic effect

In **Latch Mono/Mono** or **Latch Poly/Mono**: - a route stays active longer - phrases feel more intentional - melodic fragments persist before changing

This can create: - sustained pedal tones - held harmonic states - phrase-based melodic switching

---

6. Use Poly modes for chords or layered melodic bursts

In: - **Poly/Mono**, one or more main outputs can fire, but each selected branch still picks one suboutput - **Poly/Poly**, multiple outputs can fire across the whole tree

Melodic use

These modes are ideal when your system has: - multiple oscillators - a chord voice - several quantized pitch sources - multiple envelope/VCA paths

Patch idea

- one clock into **a**
- several outputs trigger separate envelopes opening different tuned oscillators
- each oscillator is tuned to a scale degree or interval

Result

The tree produces: - dyads - triad fragments - staggered chord tones - clustered melodic bursts

In latch versions, these can become sustained harmonic textures.

---

7. Build a self-playing melodic ecosystem

The Decision Tree excels in generative patches where one timing source branches into many related events.

Example full melodic ecosystem

- clock → **a**
- **b** advances sequencer 1
- **c** samples random voltage into quantizer 2
- **d** triggers bass envelope
- **e/f/g** trigger:
- ornament envelope
- octave shift
- glide
- **h/i/j** trigger:
- alternate scale selection
- precision adder offset
- sequence reset
- **k/l/m** trigger:
- second quantizer
- harmonic voice
- phrase reset

Result

A single incoming pulse creates: - main melody - variation - harmony - resets and phrase structure

This is one of the strongest uses of the module in melodic music.

---

8. Use the clock divider modes as melodic timing sources

The divider section is very valuable for melody because it creates multiple related trigger streams at different rates.

Classic Divider mode

According to the manual: - **b** = /2 - **e** = /4 - **f** = /8 - **g** = /16

And the lower outputs: - **c h i j k l m d** act like an **8-step clock sequencer**, cycling one after the other.

Melodic uses

- Use /2, /4, /8, /16 to trigger:
- bass notes
- melody notes
- phrase changes
- chord changes
- Use the 8-step cycling outputs to:
- step through 8 note sources
- trigger 8 stored voltages
- create an addressable melody from static CVs

Especially useful for

- techno arps
- clocked Berlin-style sequences
- phrase resets and structured melodic repetition

---

9. Use 2/3/5 Divider mode for polyrhythmic melody

The manual specifies: - **b e f g** = /2 /4 /8 /16 - **c h i j** = /3 /6 /12 /24 - **d k l m** = /5 /10 /20 /40

This is excellent for creating **multi-timescale melodies**.

Patch idea

- /2 group → bassline triggers
- /3 group → lead melody
- /5 group → accent or transposition events

Result

Even with simple pitch material, the differing divisions create: - phase relationships - evolving counterpoint - non-repeating melodic interplay

Why this works musically

Pitch content can remain simple, while timing complexity creates the sense of intelligent melodic motion.

This mode is especially good with: - quantized random voltages - shift-register melodies - fixed 8-step sequencers - transposition inputs

---

10. Use Spread mode for harmonic and overtone-based melody

The manual says Spread mode creates successive divisions: - 1/2, 1/3, 1/4, 1/5, 1/6, 1/7, etc.

It also explicitly suggests using it at **audio rate for polyphonic chords using one square VCO**.

Melodic/harmonic application 1: subharmonic melody

Feed an audio-rate square wave into the input and use the outputs as subharmonic tones.

This can create: - chordal intervals - pseudo-organ harmonies - subharmonic melodic structures

If you switch or sequence these outputs through VCAs, you can build melodies from one oscillator’s divided frequencies.

Melodic/harmonic application 2: trigger-derived note hierarchies

At clock rate, use the spread outputs to trigger different voices at increasingly slower rates: - frequent notes in one voice - slower harmony notes in another - very slow root movement in a third

This gives a natural melodic hierarchy: - ornament - phrase - harmonic anchor

---

11. Hidden reset input for phrase control

The hidden reset input uses **jack m** in a special mode, and is useful for returning to the first step in loop mode.

This matters for melody because reset is how you create: - downbeats - phrase boundaries - recurring motifs - synchronized loops with other sequencers

Use cases

- reset the module every 16 or 32 beats
- align it with a master sequencer
- force phrase restarts at section changes
- keep generative melodies from drifting too far

When activated, **m** is no longer available as an output, so it’s a tradeoff: - fewer melodic branches - more phrase control

Usually worth it if your patch needs stronger song structure.

---

12. Practical melodic patch recipes

Patch 1: Three-lane melody router

Goal: one pitch source, three melodic personalities

- clock → **a**
- quantized CV → 3 oscillators
- **b/c/d** → 3 envelopes/VCAs
- tune each oscillator differently:
- b = root register
- c = fifth above
- d = octave/high voice

Use **Mono/Mono** for clean hocketing.
Use **Poly/Poly** for chord fragments.

Patch 2: Generative melody with stable phrase memory

Goal: random notes that become recurring motifs

- trigger clock → **a**
- outputs trigger a sample & hold and quantizer setup
- set knob **n** midway toward repeating pattern
- press **o** to regenerate until the phrase feels musical

This gives a melody that sounds composed rather than chaotic.

Patch 3: Ornament engine

Goal: main notes plus decorative notes

- **b/c/d** trigger main melody voice events
- **e/f/g** trigger short envelope for upper-octave oscillator
- **h/i/j** trigger filter ping or FM accent

- **k/l/m** trigger transposition CV changes

Now the tree creates melodic detail around the core line.

Patch 4: Polyrhythmic melodic trio

Goal: three related melodies with different pulse rates

In **2/3/5 Divider** mode: - /2 outputs → bassline sequencer advance - /3 outputs → lead sequencer advance - /5 outputs → transposition or accent logic

This produces slowly evolving melodic counterpoint.

Patch 5: Subharmonic chord melody

Goal: melody/harmony from one oscillator

In **Spread** mode: - square VCO → input - outputs → separate VCAs/mixer/filter paths - use external gates to open selected divided outputs rhythmically

This creates dark, organ-like melodic harmonies from one source.

Strengths for melodic composition

The Robaux Decision Tree is especially strong for:

- **generative melodies**
- **branching phrase logic**
- **polyrhythmic note triggering**
- **hocketing between voices**
- **melody/harmony separation**
- **self-playing patches with structure**
- **subharmonic harmonic material in Spread mode**

Limitations

By itself, it does **not** generate pitch CV.
To make it fully melodic, pair it with: - quantizers - random CV sources - sequencers - sample & hold - oscillators - switch modules - precision adders - envelope/VCA voice chains

Think of it as a **melody traffic controller**, not the melody source itself.

Best partner modules

It pairs especially well with: - quantizers - Turing Machine / random CV modules - sequencers with reset/advance inputs - sample & hold - shift registers - envelope generators - logic modules - VCAs/LPGs - precision adders - multi-oscillator voice setups

Bottom line

The Decision Tree can be a powerful melodic module when used as a **rhythmic branching brain**. It shines when you want melodies that feel: - alive - structured - varied - pseudo-composed

Its strongest musical role is to distribute timing and phrase decisions across several pitch paths, turning ordinary melodic sources into evolving, layered musical systems.