

PURA BELIA

# PORTADOR

---



USER  
MANUAL  
1.0

# Table of Contents

Welcome to Portador .....	1
Installation and Setup .....	2
Overview .....	5
Oscillators .....	7
Noise Generator .....	34
Gate Generator .....	36
Filters .....	53
Envelopes .....	56
LFOs .....	60
Logic (Function Generator) .....	65
Effects .....	71
Output Channels .....	77
Chord Engine and Arpeggiator .....	82
MIDI Controllers .....	115
Macros .....	116
MIDI Learn .....	118
Modulation Matrix .....	119
Random System .....	122
Presets .....	124
Settings .....	130
CPU Meter .....	136
Scope .....	137
Polyphony .....	141
Tips and Tricks .....	143
Technical Specifications .....	145
Keyboard Shortcuts .....	146

# Welcome to Portador

Portador is a modular software synthesizer built for those who want to go further. Whether you are sculpting a single oscillator into something alive or layering five engines through a 16x16 modulation matrix, Portador gives you a studio-grade instrument with no creative ceiling.

At its core, Portador combines the warmth of classic analog-style synthesis with the precision of wavetable, granular, and physical modeling engines. Five oscillators, each with up to 10 synthesis modes, feed into dual configurable filters, a chord engine that generates harmony in real time, and an arpeggiator with probability, swing, canon, and harmonics. Every parameter can modulate every other parameter through the matrix. The result is an instrument that can be as simple or as deep as you need it to be.

Portador includes 5 assignable macros, MIDI Learn, 34 musical scales, 32 microtuning systems, and a comprehensive preset browser. Two output channels with parametric EQ, distortion, and compression let you shape and finish your sound without leaving the plugin.

This manual covers every feature in detail, from basic oscillator setup to advanced modulation routing. If you are new to Portador, start with the [Overview](#) to understand the signal flow, then explore the sections that interest you. The [Tips and Tricks](#) section at the end offers practical starting points for common sound design goals.

# Installation and Setup

**Recommended installation (macOS):** Download the signed and notarized `.pkg` installer from your purchase page. Double-click the `.pkg` and follow the prompts — the installer places all plugin formats (VST3, AU, Standalone) in their system folders, along with the Factory preset library under `~/Documents/Portador Presets/`. You can then skip ahead to the **First Launch** section below.

The installer is signed with an Apple Developer ID and notarized by Apple, so macOS opens it without Gatekeeper warnings. If you prefer to install the individual plugin files manually, continue reading the sections below.

## Plugin Formats

Portador is available in three formats:

Format	Description	Typical Use
<b>VST3</b>	Industry-standard plugin format	Ableton Live, FL Studio, Bitwig, Cubase, Reaper, Studio One
<b>AU</b> (Audio Unit)	Apple's native plugin format	Logic Pro, GarageBand, MainStage
<b>Standalone</b>	Independent application	Practice, sound design without a DAW

## Installation Paths

After installation, the plugin files are placed in the standard system locations:

Format	Location
<b>VST3</b>	<code>~/Library/Audio/Plug-Ins/VST3/Portador.vst3</code>
<b>AU</b>	<code>~/Library/Audio/Plug-Ins/Components/Portador.component</code>
<b>Standalone</b>	<code>/Applications/Portador.app</code>

## First Launch

### In a DAW (VST3 / AU)

1. Open your DAW and create a new instrument track
2. Open the plugin browser and search for “Portador”
3. Insert Portador on the track
4. Play a MIDI note to verify audio output

If Portador does not appear in the plugin list, rescan your plugins from your DAW's preferences. Some DAWs require a restart after installing new plugins.

### Standalone

1. Open Portador from your Applications folder (or wherever you installed it)
2. Go to **Settings** (gear icon) and configure your audio device and MIDI input

3. Select your audio output device and sample rate
4. Play your MIDI controller or use the computer keyboard to trigger notes

## User Data Folder

Portador stores all user data in a single location:

```
~/Documents/Portador Presets/
```

Subfolder	Contents
(root)	Synth presets (.json)
Samples/Factory/	Factory samples in category folders (A-Z)
Samples/User/	User samples (WAV)
Wavetables/Factory/	Factory wavetables in category folders (A-G+)
Wavetables/User/	User wavetables (WAV)
Venv/Factory/	Factory VENV envelopes (10 banks A-J, 10 shapes each)
Venv/User/	User VENV envelope presets
GateGen/Factory/	Factory Gate Generator presets
GateGen/User/	User Gate Generator presets
ChordArp/Factory/	Factory Chord + Arp presets
ChordArp/User/	User Chord + Arp presets

This folder is created automatically on first launch. Factory wavetables and VENV presets are generated on the first run.

## Sample Folder Organization

Factory samples are organized in lettered category folders:

```
Samples/Factory/
├─ A - Analog & Electronic/
├─ B - Field Recordings - Keys/
├─ ...
├─ G - Strings/
│   ├── Solo Violin/           ← subcategories appear as submenus
│   └─ Viola Section/
│       ├── pizz/
│       └─ trem/
└─ Z - [any new category]/
```

## Category naming rules

- Folder names starting with a letter A-Z followed by -, \_ , or a space + uppercase letter are recognized as category prefixes
- Examples: A - Analog , A-Analog , A Drums are all valid
- Folders without a letter prefix are automatically assigned the next available letter
- Samples inside each category are auto-numbered: A01, A02, B01, B02, etc.

- Subcategories (subfolders within a category) appear as nested submenus in the sample browser

**Adding sample packs:** Simply drop a new folder (e.g., `K - Producer Pack/`) into the Factory folder. The plugin detects it automatically on next load.

**Capacity:** Up to 1024 factory samples and 256 user samples.

## Wavetable Folder Organization

Factory wavetables follow the same lettered category system as samples:

```

Wavetables/Factory/
├─ A - Classic Waves/
│   └─ A01.wav
│   └─ A10.wav
├─ B - Noise/
├─ C - Complex/
├─ D - Distorted/
├─ E - Random/
├─ F - Extra/
├─ G - Experimental/
└─ H - [any new category]/ ← add your own categories

```

On first launch, Portador extracts 70 factory wavetables (7 categories, 10 each) to this folder. All wavetables are 128-frame, 2048 samples per frame, 48kHz, 32-bit float WAV files.

**Adding wavetable packs:** Drop a new category folder (e.g., `H - AnaLog/`) with .wav wavetable files into `Wavetables/Factory/`. The plugin detects new categories automatically on next load, and they appear in all wavetable menus (Osc4/5, LFO, FreqShifter, Logic).

**User wavetables:** Load custom .wav files via the “Load from file...” option in the Osc4/5 wave menu, or drop them into `Wavetables/User/`. User wavetables are auto-numbered WT01, WT02, etc.

**Capacity:** Up to 512 factory wavetables and 256 user wavetables.

## Recommended Settings

Setting	Recommendation
<b>Sample Rate</b>	44.1 kHz or 48 kHz for most use cases
<b>Buffer Size</b>	256-512 samples for a good balance of latency and CPU
<b>Voices</b>	POLY 16 (default) — reduce to POLY 8 if CPU is high
<b>Oversampling</b>	Off for low CPU, 2x for cleaner FM/AM synthesis

# Overview

Portador is organized into modular sections that can be interconnected through the modulation matrix:

- **5 Oscillators** - Main sound sources
- **Noise Generator** - Colored noise with filtering
- **Gate Generator** - 5-track polymetric gate CV generator (Euclidean, Random, Manual modes)
- **2 Dual Filters** - Combined HPF + LPF (13 types including LPG and Ladder)
- **2 Envelopes** - Classic ADSR + Vector Envelope
- **4 LFOs** - With 106 waveforms, per-voice Poly and Release modes, Pair/Quad Link modes
- **Logic** - Versatile function generator
- **Granular** - Real-time granular processing
- **Frequency Shifter** - Frequency shifting
- **Delay** - Stereo delay effect with 3 modes
- **Resonator Bank** - Tuned comb filter resonances
- **Pitch Shimmer** - Crystalline pitch-shifted feedback
- **Random Gate** - Probabilistic rhythmic gate
- **Reverb** - Stereo reverb with 3 algorithms (Plate, FDN, Recursive)
- **2 Channels** - Mixing, parametric EQ, distortion, and compression
- **Chord Engine** - Automatic chord and arpeggio generation
- **5 Macros** - Simultaneous control of multiple parameters

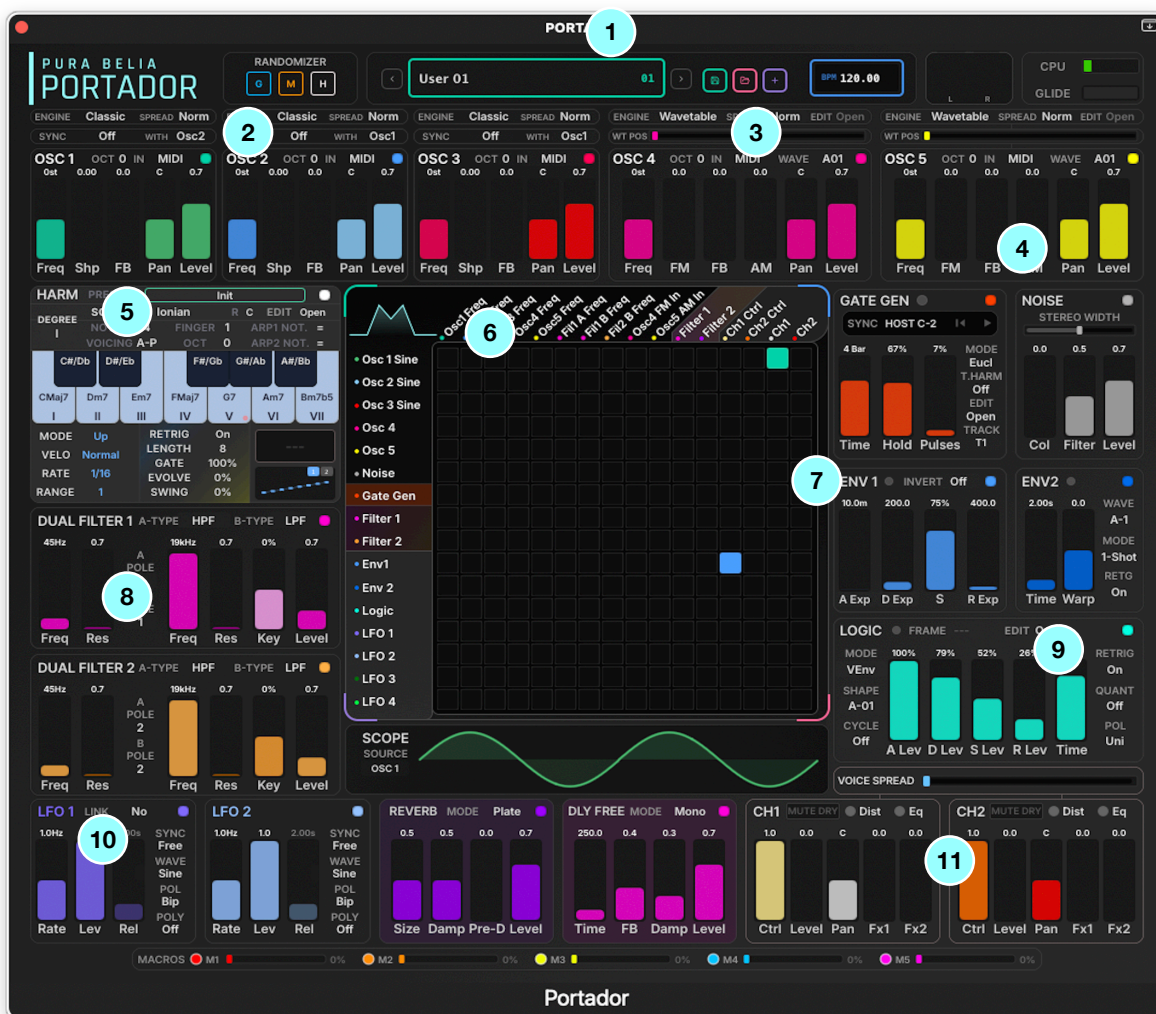


Figure 1: Portador main interface overview.

- |   |  |
|---|--|
| 1 Header bar — brand, randomizer, preset slot, BPM, CPU/GLIDE | 2 Oscillators 1, 2 and 3 (multi-engine)  |
| 3 Oscillators 4 and 5 (wavetable / sampler / granular)        | 4 Noise generator and Gate Generator     |
| 5 Chord Engine (HARM) and arpeggiator                         | 6 Modulation Matrix (16×16 routing grid) |
| 7 Envelopes (Env 1 ADSR + Env 2 Vector)                       | 8 Filter 1 and Filter 2 (dual HPF + LPF) |
| 9 Logic / VENV function generator                             | 10 LFOs (and FX strip: Reverb, Delay)    |
| 11 Channels 1 and 2 (output, EQ, distortion, compression)     |  |

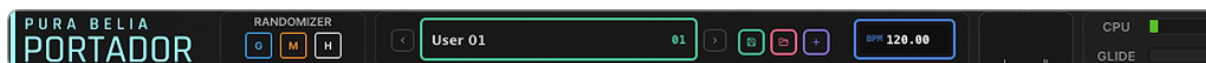


Figure 2: Header bar — left to right: brand (PURA BELIA / PORTADOR), Randomizer (C / M / H locks), preset slot with navigation, BPM control, CPU meter and GLIDE indicator.

# Oscillators

## Oscillators 1, 2 and 3 (Multi-Engine)



Figure 3: Oscillators 1, 2 and 3 — multi-engine layout. Each oscillator shows 5 faders (Freq, Shape, FB, Pan, Level) below the engine selector and source dropdown.

Oscillators 1, 2 and 3 share the same multi-engine architecture with 4 different synthesis engines:

### Synthesis Engines (8)

Engine	Description
<b>Classic</b>	Classic subtractive synthesis with standard waveforms
<b>Cloud</b>	Stereo detuned-voice synthesis with Motion crossfade — static wide cloud at Motion=0, animated breathing swarm at Motion=1
<b>Resonator</b>	Resonator bank for physical and metallic timbres
<b>Wavefolder</b>	Wave folding for rich harmonics and musical distortion
<b>Supersaw</b>	Multiple detuned saws JP-8000 style for massive sounds
<b>PhaseDist</b>	Phase distortion for unique digital timbres
<b>NoisePitch</b>	Noise with pitch tracking for tonal textures
<b>Drum</b>	Analog & organic drum synthesis with 15 drum types (submenu selection)

### Fader strip (5 slots, always visible)

Freq · Shape / Engine1 · FB / Engine2 · Pan · Level

Fader	Role	Notes
Freq (Tune)	±24 semitones	Always fine tuning.
Shape / Engine1	0-100% or engine-specific	In Classic/Drum = Shape; in Cloud = Voices; in Resonator = Body; in Wavefolder = Fold; in Supersaw = Voices; in PhaseDist = Dist; in NoisePitch = BW. Label follows engine.
FB / Engine2	0-100%	<b>Classic engine: DX7-style PM self-modulation</b> (clean at 0, warm growl ~0.3, classic DX7 bite ~0.7, screams at 1.0). In other engines = engine-specific warp fader (Motion / Bright / Sym / Detune / Res / Tone).
Pan	L-R	<b>Always active in every engine</b> (Classic, Cloud, Resonator, Wavefolder, Supersaw, PhaseDist, NoisePitch, Drum). Writes directly to <code>osc{N}_pan</code> regardless of which engine is loaded.
Level	0-100%	Oscillator volume.

### Other parameters (not in the fader strip)

Parameter	Range	Description
Wave	Sine, Saw, Triangle, Square, Pulse	Base waveform
Octave	-3 to +3	Octave transposition
Spread	Normal, Spread, Harmonic, Unison Wide, Unison Extra	Stereo spread mode
Source	MIDI, Chord Single (1-5), Chord All, Arp (Normal/Arp2/Fugue/Retrograde/Polytonal/Harmonics/NegHarm), Gate (T1-T5, Drum engine only), None	Note source

## Source Options

Source	Description
<b>MIDI</b>	Direct MIDI input - plays the note you press
<b>Chord Single (1-5)</b>	Notes from chord engine voices 1-5 (submenu). Route N oscillators to Chord1..ChordN to layer the full chord across oscillators with independent timbre, pan, and level per voice.
<b>Chord All</b>	Independent chord triggers. Each MIDI key triggers its own chord independently (polyphonic chords). Multiple chords can sound simultaneously, each with its own envelope. When <b>Glide &gt; 0</b> , switches to replace mode: the current chord voice-leads to the new chord with smooth pitch transitions. Best inversion is always used for musical voicing.
<b>Arp &gt; Normal</b>	Current arpeggiator note (quantized to arp clock)
<b>Arp 2</b>	Current note from the second arpeggiator (independent mode/rate/range)
<b>Arp 2 &gt; Fugue</b>	Diatonic canon from Arp 2's history (same params as Arp 1 Fugue)
<b>Arp 2 &gt; Retrograde</b>	Reversed Arp 2 sequence
<b>Arp 2 &gt; Polytonal</b>	Arp 2 note transposed by a fixed interval
<b>Arp 2 &gt; Harmonics</b>	Harmonic series indexed by Arp 2 step
<b>Arp 2 &gt; Neg. Harmony</b>	Negative Harmony of Arp 2 note
<b>Arp &gt; Fugue</b>	Diatonic canon — replays arp pattern delayed and transposed by scale degrees (see below)
<b>Arp &gt; Retrograde</b>	Plays the arp sequence in reverse order
<b>Arp &gt; Polytonal</b>	Arp note transposed by a fixed interval for polytonality
<b>Arp &gt; Harmonics</b>	Harmonic series based on the arpeggio root note
<b>Arp &gt; Neg. Harmony</b>	Negative Harmony — mirrors notes around the root/5th axis, scale-aware (snaps to nearest scale note)

*Gate source (visible only when the oscillator's engine is Drum):*

Source	Description
<b>Gate &gt; T1..T5</b>	Drum is triggered sample-accurately on every rising edge of the selected Gate Generator track. Replaces the old matrix "Trig In" destinations. The label and dot adopt the track's color (orange T1 / teal T2 / violet T3 / magenta T4 / lime T5). For non-drum engines use ENV1 Gate In instead — ENV1 already drives the voice's VCA.

**Source label colors with TRIG HARM:** when TRIG HARM is ON on track N, every osc whose source is chord- or arp-driven (Chord 1-5, Chord All, Arp, Fugue, Retrograde, Polytonal, Harmonics, Neg. Harmony, and the Arp 2 variants) paints its Source label in the **track's color** and appends a **T<N>** tag (e.g. **C.ALL T1** in orange, **Fug T2** in teal). At a glance you can tell which track is driving each osc. As Played / As Played 2 keep their green color — they are MIDI-driven and unaffected by TRIG HARM.

**Engine: Classic** Traditional subtractive synthesis. The Shape parameter applies: - Wave Folding on Sine - Wave Shaping on Saw - Pulse Width on Square/Pulse

The **FB** fader (fader 3) adds DX7-style PM self-modulation (operator feedback): previous output sample warps the phase lookup with a 0.5× scale factor. At 0 the oscillator is clean; around 0.3 it picks up a warm

analog growl; at 0.7 it hits the classic DX7 bite; at 1.0 it screams. Works across all base waveforms. FB is only audible in Classic — other engines reuse the same fader slot for their own engine-specific control.

**Engine: Cloud** Stereo detuned-voice synthesis. Stacks 2-16 voices with micro-detuning and cross-fades between two character extremes via the **Motion** fader: - **Voices** (2-16): Number of detuned voice layers with independent phase accumulators - **Motion** (0-100%): Crossfades character from static wide stereo cloud to animated breathing swarm - Motion=0: subtle 0.2 semitone base detune, full stereo pan spread, no drift — classic wide static pad - Motion=1: up to  $\pm 1.4$  semitones cluster detune + per-voice LFO drift with secondary beating interference + amplitude breathing + progressive wavefolding + 40% pan narrow — punchy animated swarm - In-between: smooth crossfade between the two extremes - Works with all base waveforms (Sine, Saw, Triangle, Square, Pulse)

**Engine: Resonator** Physical resonator bank: - Simulates resonances of acoustic bodies - Perfect for metallic, string, or percussive timbres

**Engine: Wavefolder** West coast style wave folding for complex harmonics: - Folds the waveform onto itself using sinusoidal folding with exponential gain curve - Smooth transition from clean to folded — gradual crossfade across the full Iterations range - **Iterations**: Fold amount (exponential curve for more usable range in lower settings) - **Warp**: Asymmetry (0 = symmetric folds, 1 = asymmetric for even harmonics)

**Engine: Supersaw** Multiple detuned voices (JP-8000 style) with aggressive character: - 2-16 detuned voices distributed across the stereo field ( $\pm 1$  semitone max spread) - Outer voices receive additional saturation for the characteristic “buzz” - **Iterations**: Voice count (2-16) - **Warp**: Detune spread amount — higher detune also increases output saturation for a harder sound - Works with all waveforms (not just saw), ideal for trance leads, massive pads, and thick chords

**Engine: PhaseDist** CZ-style phase distortion for digital timbres: - Distorts the phase trajectory of the waveform — accelerates first half, decelerates second half - Creates timbral morphing similar to filter cutoff sweeps but with a unique digital character - **Iterations**: Distortion amount (quadratic curve for extended sweet spot in the mid-range) - **Warp**: Resonance feedback — adds a resonant peak to the timbre like a filter Q - Pulse waveform produces CZ-style resonant wave ( $\cos \times \sin$  harmonics)

**Engine: NoisePitch** Pitched filtered noise with high-Q SVF bandpass: - Generates noise filtered through a proper SVF bandpass centered on the MIDI note frequency - Q range from 2 to 40 — at narrow bandwidth, produces clearly pitched tonal noise; at wide bandwidth, produces textured broadband noise - 5 noise types: White, Pink, Brown, Crackle, Digital - **Iterations**: Bandwidth (narrow = pitched, wide = noisy) - **Warp (Tone)**: Adds brightness (high-passed noise blended in)

**Engine: Drum** Drum synthesis engine with 15 types organized in two categories — **Classic** (traditional analog-modeled) and **Organic** (experimental synthesis): - Select the drum type from the **Engine menu submenu** — hovering over “Drum” opens a categorized submenu - The engine button displays the selected drum type name (e.g. “Kick I”) - **Decay** (Shape fader): Controls the envelope tail length (0 = short, 1 = long) - **Tone** (Warp fader): Controls timbre and brightness - **Tune/Octave**: Adjusts the drum pitch - Velocity modulates both volume and brightness (higher velocity = brighter transients)

## Classic Drum Types

Type	Description
Kick I	Classic analog kick — bridged-T resonator with pitch sweep and deep sub-bass
Kick II	Punchy attack kick — triangle VCO with sub oscillator and noise snap
Snare I	Dual-tone snare — two inharmonic bridged-T resonators + HP-filtered noise
Snare II	Tight noise snare — dual triangle VCOs with broadband noise
Hat I	Metallic hi-hat — 6 metallic square oscillators with 4-stage HP cascade
Hat II	Airy cymbal — ring modulation (inharmonic sine ratios) + filtered noise, stereo
Clap	Analog clap — multi-burst noise pattern with reverb tail (up to ~1.2s)
Tom	Pitched tom — bridged-T resonator with tunable pitch sweep
Cowbell	Dual-square cowbell — two detuned square oscillators with BP filter
Perc	Configurable percussion — 5 sub-types (Rimshot, Blip, Zap, Metallic, Noise)

## Organic Drum Types

Type	Description
Particle	Micro-grain cloud — 8 concurrent grains with randomized pitch, pan, and amplitude
Wavefold	Wavefolded transients — sine through multi-stage wavefolder with decaying fold depth
Metal	6-oscillator additive/FM — variable spread from harmonic (bells) to inharmonic (crashes)
Chaos	Lorenz attractor — chaotic oscillator producing organic, evolving, semi-unpredictable sounds
Feedback	Self-oscillating filter — SVF pushed past self-oscillation with cutoff sweep and feedback

**Note:** Organic types (Particle, Wavefold, Metal, Chaos, Feedback) are **tonal** — they track MIDI pitch by default (Pitch Lock is auto-disabled). Classic types auto-enable Pitch Lock for fixed-pitch behavior.

## Drum Variants

All 15 drum types offer 5 sub-variants selectable via the wave menu (Osc1-3) or VARIANT slider (Osc4-5):

Drum Type	Variants
Kick I	Sub, Punchy, Tight, Boom, Drive
Kick II	Classic, Hard, Soft, Long, Distorted
Snare I	Classic, Tight, Rimmy, Fat, Noisy
Snare II	Classic, Crack, Thick, Ghost, Trash
Hat I/II	Closed 1, Closed 2, Closed 3, Open 1, Open 2
Clap	Classic, Tight, Stack, Room, Snap
Tom	Low, Mid, High, Floor, Roto
Cowbell	Cowbell, Agogo, Clave, Block, Conga
Perc	Rimshot, Blip, Zap, Metallic, Noise
Particle	Dust, Cloud, Scatter, Swarm, Storm
Wavefold	Fold, Crush, Bite, Shred, Mangle
Metal	Bell, Strike, Clang, Crash, Shatter
Chaos	Orbit, Edge, Swarm, Fractal, Storm
Feedback	Sub, Ping, Growl, Scream, Acid

## Drum Character (Osc4-5 only)

The **Fader 6** (normally FB) becomes a per-drum-type shaping control in Drum mode. The label changes dynamically:

Drum Type	Label	Effect
Kick I	Sweep	Pitch sweep range (subtle → deep boom)
Kick II	T.Depth	Pitch sweep speed (fast/punchy → slow/boomy)
Snare I/II	Snappy	Body vs noise wire balance
Hat I/II	Bright	HP cutoff and resonant brightness
Clap	Room	Burst spacing and room size
Tom	Decay	Pitch sweep depth
Cowbell	Ring	Beating frequency between oscillators
Perc	Char	Varies per variant
Particle	Scatter	Pitch randomization between grains
Wavefold	Symm	Fold asymmetry (even vs odd harmonics)
Metal	Morph	Waveform morph (sine → triangle)
Chaos	Chaos	Lorenz rho parameter (periodic → chaotic)
Feedback	Filter	Filter type blend (LP → BP → HP)

**Drum + Gate In:** Gate In connections (Env1/Env2/VENV/H Gate In) trigger the Drum engine when the oscillator is in Drum mode, enabling rhythmic drum patterns without MIDI notes. Note: H Gate In no

longer auto-triggers envelopes — connect each envelope’s dedicated Gate In (Env1/Env2/VENV Gate In) in the Matrix for independent control.

### Internal Envelope and Gate source

Drum engines have their own internal amplitude envelope — they don’t need Env1 or Env2 to produce sound. By default, they trigger on every note-on (MIDI, arpeggio, or TRIG HARM — matching the oscillator’s Source setting). For **step-sequenced drum hits independent of your MIDI playing**, set the oscillator’s **Source to Gate > T1..T5**: the drum fires sample-accurately on every rising edge of the chosen Gate Generator track, and the Source label in the UI adopts that track’s color. You can still play MIDI fills on top — the Source governs which notes the oscillator responds to, and a Drum osc with `Source = Gate T_N` plus MIDI notes layers both streams (gate sequence + MIDI fills). To silence the MIDI fills, set Source to None and leave only the Gate trigger.

*Note (upgraders from earlier builds):* the old matrix “**Trig In**” destinations (one per oscillator) have been removed — they are replaced by the `Source = Gate T1..T5` option described above. The per-source approach prevents the “double trigger + TRIG HARM” trap (gate gen → Trig In + TRIG HARM on the same track used to fire the drum twice per step).

### Spread Modes (Osc1-3)

Mode	Description
Normal	Single voice, no additional stereo effect
Spread	Frequency band stereo redistribution (Logic Pro Stereo Spread style). Splits the signal into frequency bands and alternates them between L/R channels with configurable intensity
Harmonic	Harmonic series stereo spread — distributes odd harmonics to one channel and even harmonics to the other, creating a natural, phase-coherent stereo image
Unison Wide	4 detuned voices distributed in stereo
Unison Extra	8 detuned voices distributed in stereo

### Spread Parameters (accessible from the submenu when Spread is selected)

Parameter	Range	Description
Order	4-12	Number of frequency bands
Low Pan	0-100%	Stereo panning intensity in the low end
Hi Pan	0-100%	Stereo panning intensity in the high end
Lower Freq	20-2000 Hz	Lower frequency limit of the band range
Upper Freq	2k-20k Hz	Upper frequency limit of the band range

Osc1-3 Spread parameters are modulatable from the Modulation Matrix.

## Synchronization (OSC 1, 2 and 3)

All three oscillators (1, 2 and 3) include synchronization capability:

Sync Type	Description
Off	No synchronization
Hard	Hard reset - classic sync sound
Soft	Soft reset - more musical
Phase	Phase synchronization
Reverse	Reversal on each cycle

**Sync Target:** Can sync to any other oscillator (Osc1-5).

## Oscillators 4 and 5 (Wavetable Multi-Engine)



Figure 4: Oscillators 4 and 5 — wavetable multi-engine layout. Each shows 6 faders (Freq, FM/PM, FB or engine-specific, AM/RM, Pan, Level) plus the Wave selector and engine slot.

Wavetable synthesis engines with FM/PM/TZFM, AM/RM modulation, feedback, and 12 synthesis engines. FM/PM/TZFM and AM/RM work in **all engine modes**:

Parameter	Range	Description
Wave	Dynamic	Wavetable selection from factory categories (A-G+) and user tables
Position	0-100%	Position within wavetable (128 cycles)
Freq	±24 semitones	Tuning
FM/PM Index	0-100%	FM or PM modulation intensity (click label to toggle)
Fader 3	0-100%	Context-dependent per engine (see table below)
AM/RM Index	0-100%	AM or RM modulation intensity (click label to toggle)
Pan	L-R	Stereo position
Level	0-100%	Volume
Octave	-3 to +3	Transposition
Spread	Normal, Spread, Harmonic, Unison Wide, Unison Extra	Stereo spread mode

---

**6 faders per oscillator:** Freq, FM/PM, Fader 3, AM/RM, Pan, Level.

### **FM/PM/TZFM Toggle (Frequency / Phase / Through-Zero FM)**

Click the **FM** label to cycle through **PM** (Phase Modulation) and **TZFM** (Through-Zero FM). FM modulates the oscillator's frequency, creating classic FM synthesis timbres. PM modulates the phase directly, producing more metallic, bell-like tones. TZFM (Through-Zero FM) sweeps the oscillator's frequency through zero Hz and into negative frequencies, producing stable, alias-free FM tones that stay in tune regardless of modulation depth — ideal for deep bass and complex harmonic spectra. All three modes use the same FM Index fader and the same modulation source from the matrix. The FM/PM/TZFM toggle is available in **Wavetable and engine modes 1-7**. In Sampler/Granular, the fader becomes **P.Rat** (Pitch Ratio) — all three modes produce the same effect (playback speed modulation). In Drum mode, the fader is not visible on the UI but the parameter (**P.Mod** — Pitch Mod Depth) is accessible via the modulation matrix.

### **AM/RM Toggle (Amplitude vs Ring Modulation)**

Click the **AM** label to switch to **RM** (Ring Modulation). AM adds the modulator on top of the carrier signal (preserving the original). RM multiplies carrier  $\times$  modulator with no DC offset, creating sum and difference frequencies. RM crossfades smoothly from dry (fader at 0) to fully ring-modulated (fader at max) with gain compensation. AM/RM works in **all engine modes** including Drum — use the modulation matrix to set the AM Index and connect an audio source to the AM Audio In.

### **FB (Feedback)**

The oscillator's output feeds back into its own FM/PM input with a one-sample delay. Creates complex harmonics, metallic textures, and noise at high settings. Works in all engine modes (0-7).

## FM/AM per Engine Mode

Engine	FM fader	FM/PM/TZFM	AM/RM	Notes
Wavetable (0)	FM Index	Real FM/PM/TZFM	Yes	Full frequency/phase modulation
Cloud-NoisePitch (2-7)	FM Index	Real FM/PM/TZFM	Yes	Via matrix (faders show engine params). On Cloud, matrix slots relabel as Voices/Motion/FB for semantic clarity
Sampler (8)	P.Rat	Pitch ratio (all modes same)	Yes	Modulates playback speed
Granular (9)	P.Rat	Pitch ratio (all modes same)	Yes	+ per-grain FM/AM option
Drum (10)	P.Mod (matrix)	Pitch sweep depth	Yes	Deeper pitch sweep at higher values
Quantum (11)	Fuse Strength	Quantum coupling	No	Fuse mode couples two quantum oscillators

## Fader 3 — Per-Engine Control

Engine	Fader 3	Description
Wavetable	FB	FM/PM self-feedback
Cloud	FB	Per-voice DX7-style transient phase modulation feedback, decorrelated across voices via pseudo-random per-voice signs for stable character across the full FB range
Resonator	Pos	Excitation position — simulates plucking at different points along the string
Wavefolder	FB	FM/PM self-feedback
SuperWave	FB	FM/PM self-feedback
PhaseDist	Phase	Second phase distortion stage (independent from Amount)
NoisePitch	Reso	Extra resonance (Q boost) on top of the bandwidth filter
Sampler	FB	FM/PM self-feedback
Granular	Dir	Grain direction blend (reverse ↔ forward)
Drum	Decay	Drum envelope decay length
Quantum	Absorb	Energy absorption rate — how fast the waveform decays

## Synthesis Engines (11)

Engine	Description
<b>Wavetable</b>	Classic wavetable synthesis with no additional processing
<b>Cloud</b>	Stereo detuned wavetable voices with Motion crossfade from static cloud to breathing animated character (see Osc1-3 Cloud for full Motion fader behaviour)
<b>Resonator</b>	Karplus-Strong style resonant body over the wavetable
<b>Wavefolder</b>	Wave folding applied to the wavetable
<b>SuperWave</b>	Massive unison with up to 16 detuned micro-voices
<b>PhaseDist</b>	CZ-style phase distortion over the wavetable
<b>NoisePitch</b>	Pitched filtered noise modulated by the wavetable
<b>Sampler</b>	Sample playback with loop, crossfade and pitch-tracking
<b>Granular</b>	Granular synthesis with grain control, position and randomization
<b>Drum</b>	Analog & organic drum synthesis with 15 drum types (submenu selection)
<b>Quantum</b>	Schrödinger wave function synthesis — evolving wavetables via quantum simulation

### Engine: Quantum

Quantum is a unique synthesis engine that evolves wavetables in real-time using the Schrödinger equation (1D quantum mechanics simulation). Instead of reading a static wavetable, the waveform

lives and transforms according to quantum physics — dispersing, reflecting, interfering and reforming over time. This creates organic, constantly-evolving timbres impossible to achieve with conventional synthesis.

**How it works:** When a note is triggered, a wavetable frame is loaded as the initial quantum state (with subtle quantum fluctuations added to ensure the waveform always evolves, even from simple shapes like sine waves). The simulation then evolves this state at a rate controlled by RATE using the split-step Fourier method with sub-stepping for smooth, accurate evolution. The WIDTH parameter determines how concentrated the initial energy is — at high WIDTH values, the energy concentrates into a narrow packet that disperses dramatically (the classic quantum morphing effect where waveforms stretch from sine-like to saw-like and back). The WT POS slider works in real-time — moving it while a note is held instantly loads a new wavetable frame as the quantum state, allowing you to sweep through frames while the simulation continues evolving.

### Main Panel Controls (Quantum mode)

Control	Label	Description
WT POS slider	WT POS	Wavetable frame selection (same as other engines) — selects which frame becomes the quantum state
Fader 2	FM	FM Index (same as standard engines) — FM/PM/TZFM modulation works with quantum
Fader 3	AM	AM Index (same as standard engines) — AM/RM modulation works with quantum
Fader 6	Rate	Simulation speed — how fast the waveform evolves (0=static, max=fast morphing)
FM Mode	FM/PM/TZFM	Modulation type (Fuse is now a separate control in the Quantum Editor)

### Quantum Editor (EDIT button)

Opens a floating editor (731×511) with all quantum controls and a real-time waveform visualization showing the evolving wave function.

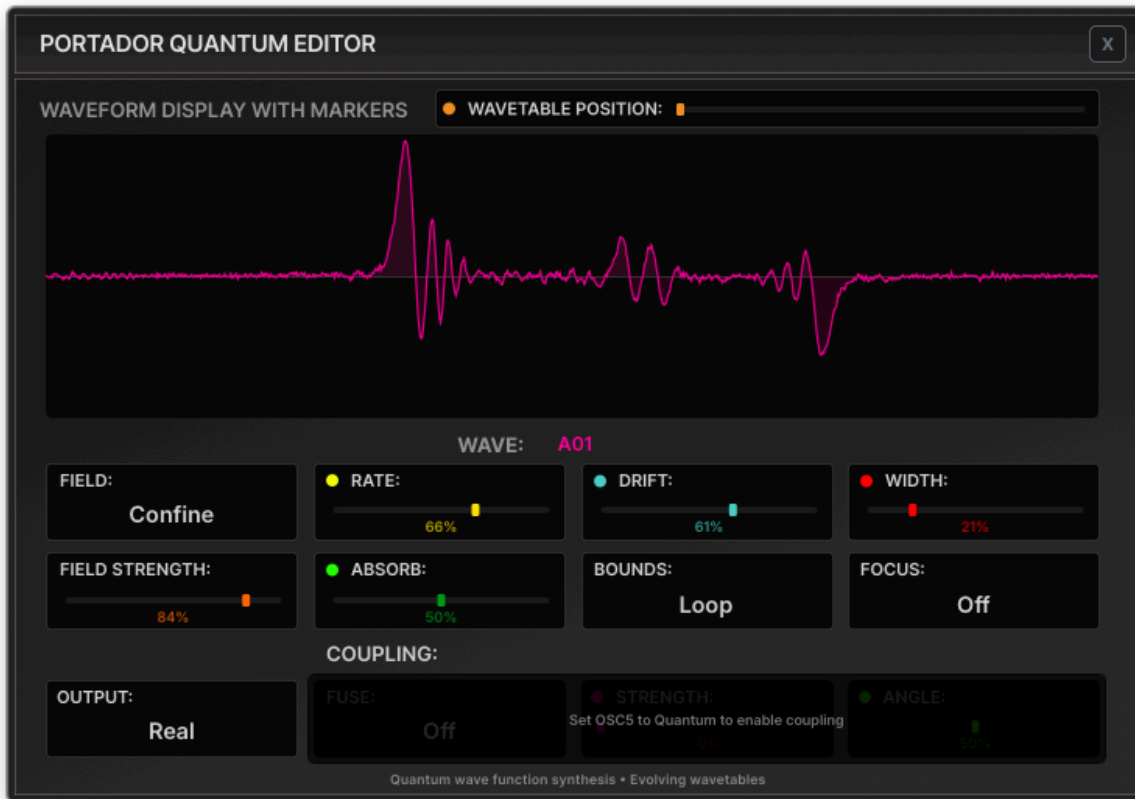


Figure 5: Quantum editor — real-time visualization of the wave function plus controls for evolution, dispersion, focus and fuse coupling.

Control	Description
WT POS	Selects which frame of the loaded wavetable becomes the quantum state (0-127). Works in real-time — moving it while a note is held loads the new frame instantly
RATE	Simulation speed (same as main panel). Slider sweeps respond in real-time (smoother is per-block-correct so the 50 ms ramp matches wall-clock time)
DRIFT	Directional momentum. <b>Unipolar: 0 = stationary, 1 = max forward drift.</b> Continuously injects positive phase gradient into the wavefunction so the spectrum slowly drifts upward over the held note. The slider is monotonic — moving it up always increases drift
WIDTH	Gaussian windowing applied <b>continuously</b> to the wavefunction. 0 = full wavetable, 1 = narrow packet. <b>High WIDTH is key for the classic quantum morphing effect.</b> The blend is now strong enough that the windowing converges in ~5 blocks instead of being washed out by the wavetable drive
FIELD (menu)	Energy field type: None, Confine (parabolic trap, default), Peak (central barrier), Valley (central well). Field strength is set via the FIELD STRENGTH slider (range maps from iterations 2-16 to potential strength 0 — 1)
OUTPUT (menu)	How audio is derived from the complex wavefunction: Real, Imaginary, Density (probability density mapped to full bipolar range), Stereo (L=Real, R=Imaginary)
ABSORB	Spectral damping. Higher harmonics are damped quadratically more than the fundamental, so high values produce a clearly audible darkening sweep. Pair with high RATE for “the lights going out” effects
BOUNDS (menu)	Boundary behavior: Loop (periodic), Reflect (mirror at edges), Decay (fade at edges)
FOCUS (menu)	Quantum focus: Off (free evolution), Spatial (concentrate on strongest region), Spectral (concentrate in frequency domain)

## Fuse — Quantum Coupling

When both OSC4 and OSC5 are set to Quantum engine, they can be quantum-coupled using Fuse. The coupling section in the Quantum Editor automatically activates when both oscillators are in Quantum mode. Toggle FUSE to “On” in the editor’s coupling section. Fuse is independent from FM — you can use FM/PM/TZFM modulation AND Fuse simultaneously (FM modulates the playback frequency, Fuse couples the quantum states — two independent layers).

Fuse creates a beam-splitter interaction where the two quantum wavefunctions influence each other mutually, plus audio cross-blend and ring modulation between the two outputs. This is fundamentally different from FM where only one oscillator modulates the other.

Control	Description
FUSE (toggle)	Quantum coupling On/Off (only available when both OSC4 and OSC5 are Quantum)
Strength	How strongly the two quantum states interact + audio cross-blend amount
Angle	Beam-splitter angle — controls the ring modulation intensity between the two outputs

**Visual Entanglement:** When Fuse is active, the Quantum Editor shows both waveforms overlaid — the current oscillator’s waveform in its own color (pink for OSC4, yellow for OSC5) and the other oscillator’s waveform in semi-transparent complementary color. This lets you see in real-time how the two quantum states influence each other — physics visualized.

When both oscillators are in Quantum mode, cyan borders and connector lines appear on the main panel, visually indicating the quantum link.

### Quantum in the Modulation Matrix

The Quantum module in the matrix provides modulatable destinations: - O4/O5 Output — modulate the output derivation mode - O4/O5 Field — modulate the energy field type - O4/O5 Focus — modulate the focus mode - O4/O5 Bounds — modulate the boundary behavior - O4/O5 Rate — modulate simulation speed - Fuse Str — modulate coupling strength - Fuse Angle — modulate beam-splitter angle - Fuse On — modulate coupling on/off

Additionally, the Osc 4/5 matrix sections include engine-specific parameters with dynamic labels: - Field (iterations) — potential field strength - Absorb (warp) — energy absorption/damping - Drift (feedback) — directional momentum - Width (excitation pos) — gaussian confinement

These labels change automatically based on the active engine mode (e.g., “Warp” in Wavetable, “Absorb” in Quantum).

### Quantum Sound Design — Quick Start

Quantum sounds great when you treat the parameters as a *system* — every control feeds back into the wavefunction, so small adjustments compound over a few seconds of held note. Five recipes to learn the engine:

1. **Pure Sine Evolution** — OSC4=Quantum, WT POS=0 (Sine bank), RATE=30%, WIDTH=40%, FIELD=50%, DRIFT=0, Potential=Confine, Output=Real, Bounds=Wrap. No FX, slight reverb only. Hold a note for 5+ seconds and listen as the pure sine slowly transforms into harmonically rich content. This is the cleanest way to “hear what Quantum does”.
2. **Entangled Pad** — Both OSC4 + OSC5 in Quantum. OSC4 WT POS=0 and OSC5 WT POS=0.6 (different starting frames so they begin divergent). Both Field=Confine, RATE=50% on both. **Fuse: ON, Strength=70%, Angle=50%**. Long attack/release envelope (3-5 s). The two oscillators now sound like ONE evolving instrument because the wavefunctions are coupled at the state level.
3. **Chaotic Texture** — RATE=90%, WIDTH=90%, FIELD=80%, Potential=Barrier, Focus=Spatial, Bounds=Decay. Heavy reverb + delay. The note becomes an evolving textural cloud — perfect for atmosphere and sound design beds.
4. **Harmonic Trap Pad** — WT POS=0.2 (saw-ish), RATE=50%, FIELD STRENGTH=80%, Potential=Confine, Focus=Off. Slow attack envelope. Sounds like a string/pad reluctant to leave its trapped state, with characteristic harmonics emerging as the wavefunction settles.

5. **Quantum FM** — OSC4 + OSC5 both in Quantum. FM Mode=FM, FM Index=40% (OSC5 frequency-modulates OSC4). **Fuse: ON, Strength=50%**. Both Potential=Confine. Demonstrates the unique combination only Portador offers: FM in the audio domain *and* Fuse in the quantum-state domain at the same time — two independent layers of interaction.

### Tips for taming Quantum

- DRIFT is unipolar — you can't go negative from the slider. If you want descending sweeps, use a bipolar matrix source on `osc4_quantum_rate` instead.
- WIDTH > 0 makes the wavefunction physically narrower, which sounds *brighter* (a narrow time-domain pulse has wide spectrum). Pair high WIDTH with low ABSORB for sharp percussive Quantum tones.
- ABSORB only acts on harmonics, not the fundamental. Use it for “lights going out” sweeps over sustained notes.
- The simulation is per-block (4 FFTs of 512 samples = ~2.7 ms at 44.1 kHz). RATE 0 freezes it; RATE 1 evolves it as fast as the kinetic clamp allows. Sweeping RATE in real time is now smooth (the per-block smoother was fixed).
- Two voices in unison genuinely diverge now (each has an independent perturbation seed), so chord stacks in Quantum don't sound like one robotic voice.

### Engine Parameters

Parameter	Range	Description
Iterations	2-16	Number of copies/voices/iterations (varies by engine)
Warp	0-100%	Displacement/spread/damping (varies by engine)

### Spread Modes

Mode	Description
Normal	Single voice, no additional stereo effect
Spread	Frequency band stereo redistribution (Logic Pro Stereo Spread style). Splits the signal into frequency bands and alternates them between L/R channels with configurable intensity
Harmonic	Harmonic series stereo spread — distributes odd harmonics to one channel and even harmonics to the other, creating a natural, phase-coherent stereo image
Unison Wide	4 detuned voices distributed in stereo
Unison Extra	8 detuned voices distributed in stereo

## Spread Parameters (accessible from the submenu when Spread is selected)

Parameter	Range	Description
Order	4-12	Number of frequency bands
Low Pan	0-100%	Stereo panning intensity in the low end
Hi Pan	0-100%	Stereo panning intensity in the high end
Lower Freq	20-2000 Hz	Lower frequency limit of the band range
Upper Freq	2k-20k Hz	Upper frequency limit of the band range

Spread parameters are modulatable from the Modulation Matrix (inside the Osc4/Osc5 module).

## Osc4 and Osc5 Wavetables (70 factory tables, A-G)

- Bank A (0-9): Classic Waves
- Bank B (10-19): Noise
- Bank C (20-29): Complex
- Bank D (30-39): Distorted
- Bank E (40-49): Random
- Bank F (50-59): Extra
- Bank G (60-69): Experimental

Both oscillators share the same wavetable library. All wavetables are 128 cycles with 2048 samples per cycle.

## User Wavetables (Osc4 and Osc5)

You can load your own WAV files as custom wavetables:

1. Open the wave selection menu for oscillator 4 or 5
2. In the **User** category, select “**Load...**”
3. Choose a WAV file from your disk
4. The wavetable is automatically split into 128 frames × 2048 samples

Up to 32 user wavetables can be loaded. The paths of loaded WAV files are saved and restored with the plugin state.

**Tip:** For best results, use WAV files with repeating wave cycles. The plugin automatically resamples any WAV file to the internal wavetable format.

## Sampler Engine (Mode 8)

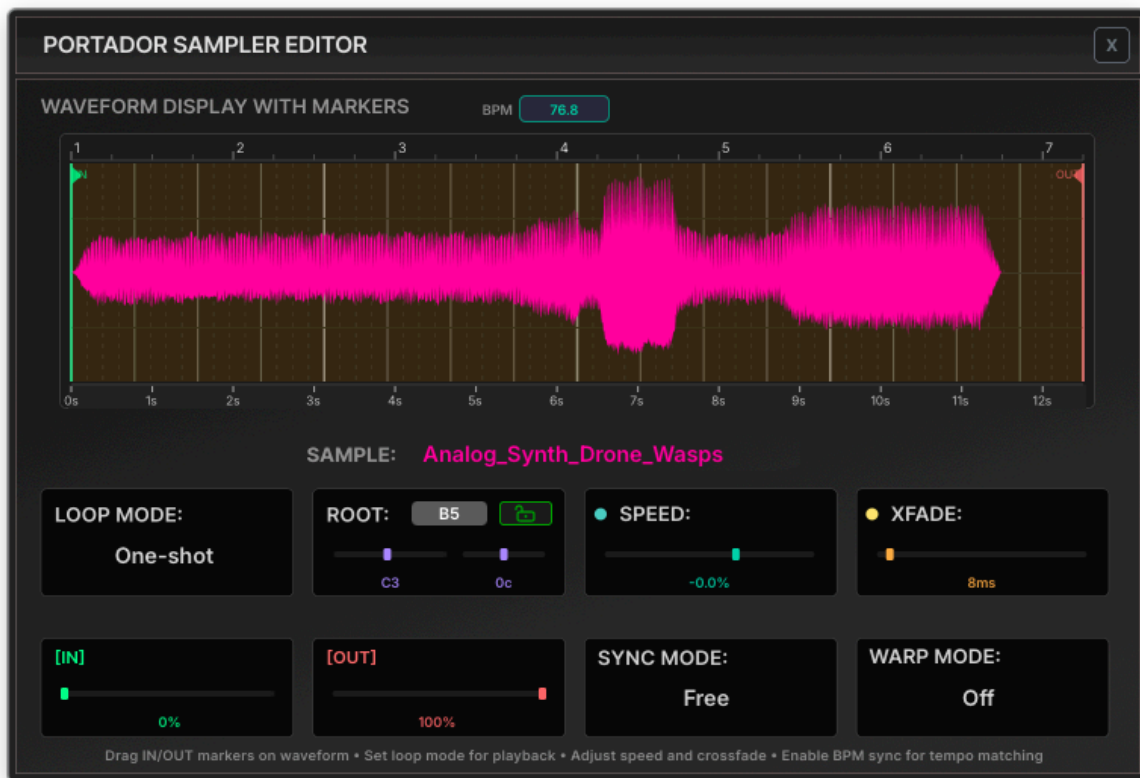


Figure 6: Sampler editor — waveform display with loop and crossfade controls.

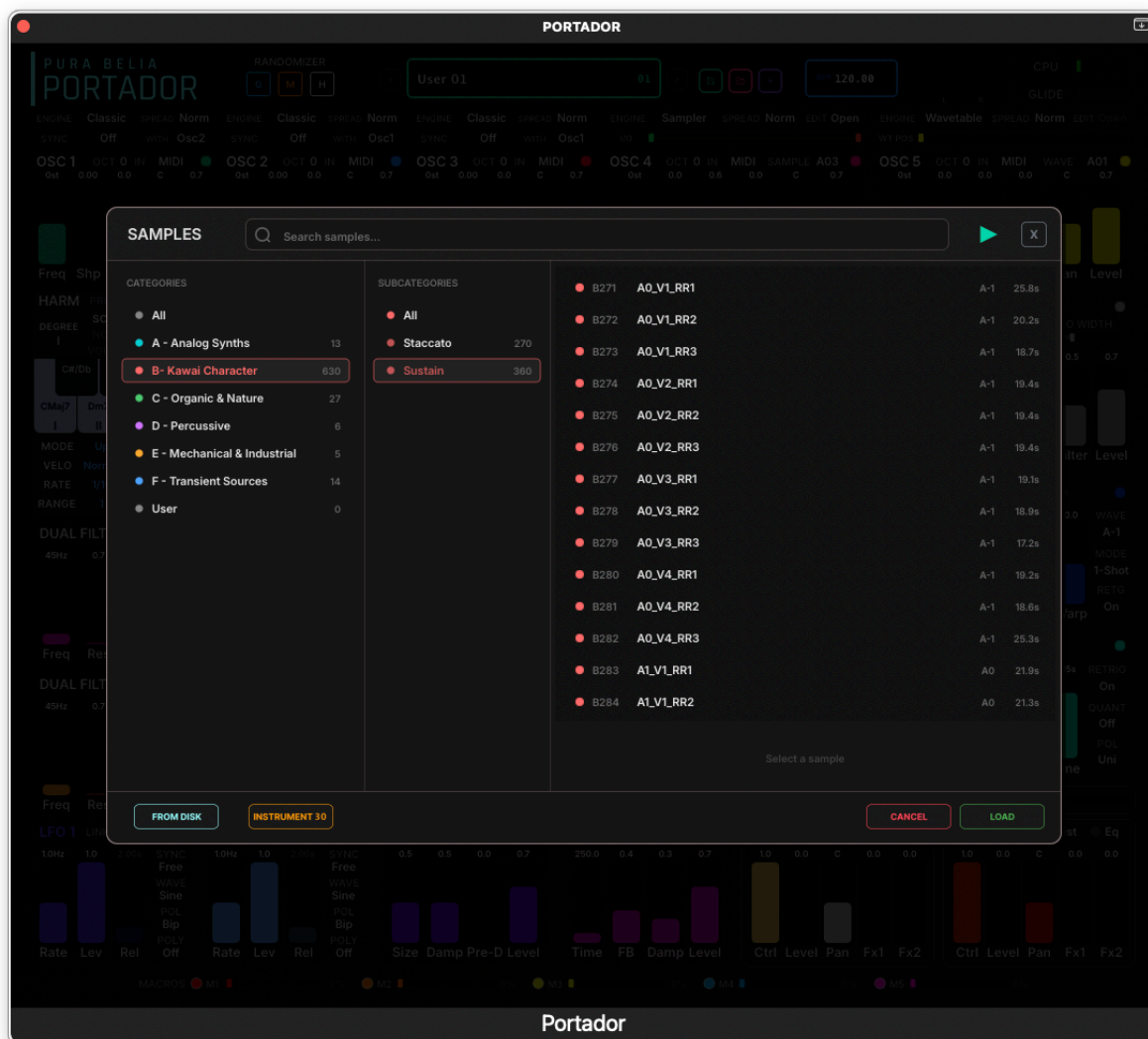


Figure 7: Sample browser — factory and user sample folders with category submenus.

The Sampler engine allows you to load and play audio files with loop control, crossfade and automatic pitch-tracking.

### Loading a Sample

1. Select the **Sampler** engine on OSC4 or OSC5
2. Load a sample using one of these methods:
  - Click the **SAMPLE** button on the main UI to open the **Sample Browser**
  - Drag an audio file (WAV, AIFF, FLAC, MP3, OGG) directly to the Sampler/Granular editor waveform area
3. The sample will load and display its waveform

## Sample Browser

Clicking the **SAMPLE** button opens a full-screen modal browser (similar to the Preset Browser) with the following layout:

Section	Description
<b>Search Bar</b>	Filter samples by name, short code, or category
<b>Categories</b> (left column)	All, factory categories (A-Z), User
<b>Subcategories</b> (middle column)	Subfolders within the selected category
<b>Folders</b> (right column, optional)	Deeper subfolders — only visible when the subcategory has children
<b>Sample Grid</b>	Scrollable list of samples showing colored dot, short code (A01, G15...), name, root note, and duration
<b>Preview Panel</b>	Shows selected sample details: name, root note, duration, sample rate

## Browser Controls

- **Single click** on a sample selects it and plays a preview (if autoplay is enabled)
- **Double click** loads the sample immediately and closes the browser
- **LOAD button** loads the selected sample and closes the browser
- **CANCEL button** closes without loading
- **FROM DISK button** opens a file chooser to import an external audio file — the file is copied to the User samples folder automatically
- **Autoplay toggle** (play icon in the header) enables/disables automatic preview on selection. Enabled by default.
- **Esc key** closes the browser
- Column widths are resizable by dragging the dividers, and widths are saved between sessions.

## User Sample Management

- **FROM DISK:** When you load a sample from disk, it is **copied** to `~/Documents/Portador Presets/Samples/User/` — the original file is never modified
- **Delete:** User samples show a delete button (X) on hover. Clicking it shows a confirmation dialog. Only the copy in the User folder is deleted, never the original file. Factory samples cannot be deleted.
- **Rename:** Double-click the name of a User sample in the browser to rename it. The short code (U01, U02...) is not affected.

Factory samples display a short code (e.g., **A01**, **G15**) on the UI button to prevent label truncation. The full name appears in the browser grid (e.g., `A01 - Analog_Synth_Drone`).

## Multi-Sample Instruments

Multi-sampling allows loading multiple samples of the same instrument recorded at different pitches (e.g., a violin sampled at C3, E3, G3, C4...). When playing chords, each note automatically selects the closest sample, eliminating the “chipmunk effect” that occurs when a single sample is pitch-shifted over large intervals.

## How It Works

- When you navigate to the **deepest subfolder** in the Sample Browser containing **2 or more samples with note names in their filenames** (e.g., `Violin_C3.wav`, `Violin_E3.wav`), the **INSTRUMENT**

button appears in the footer. If the current folder has subfolders, you must navigate into a specific subfolder first — the button won't appear at a parent level that groups multiple folders

- The button shows the number of qualifying samples (the count is displayed on the button, matching the button's text color)
- Clicking **INSTRUMENT** loads all qualifying samples as a single multi-sample instrument
- Each sample is automatically mapped to a zone covering the MIDI notes closest to its root note
- When you play a note, Portador selects the sample with the nearest root note and applies minimal pitch-shifting (typically  $\pm 1-2$  semitones — imperceptible)

## Filename Convention

Filename	Root Note	Round Robin
violin_C3.wav	C3	none
violin_C3_RR1.wav	C3	Round Robin 1
violin_C3_RR2.wav	C3	Round Robin 2
violin_F#3.wav	F# 3	none
pad_atmospheric.wav	Not detected	Single sample (full keyboard)

- **Note detection:** Portador recognizes note names (C, D, E, F, G, A, B) with optional sharp (#) or flat (b) and octave number (0-9) anywhere in the filename
- **Round Robin:** Add `_RR1`, `_RR2`, etc. to the filename. Samples with the same root note and different RR numbers will automatically alternate on each note-on, preventing the “machine gun effect” on repeated notes

## Zone Auto-Mapping

Zones are automatically calculated by splitting the MIDI range at the midpoint between adjacent root notes. For example, with samples at C3 (48), E3 (52), and G3 (55): - C3 zone covers MIDI notes 0-50 - E3 zone covers MIDI notes 51-53 - G3 zone covers MIDI notes 54-127

## Multi-Sample with Chords (SOURCE = CHORD ALL)

This is the primary use case for multi-sampling. When using chord generation: 1. Load a multi-sample instrument (e.g., orchestral strings)

2. Set SOURCE to CHORD ALL or any chord mode 3. Play a chord — each

voice independently selects the closest sample zone 4. Result: every note in the chord sounds natural, as if played by a real instrument

## Visual Indicators

- **Wave Button:** Shows `MS` followed by the zone count (e.g., `MS11` for 11 zones)
- **Sampler/Granular Editor:** Shows the instrument name with zone count (e.g., “Violin Sustain (11 zones)”) and a **zone indicator bar** below the waveform displaying each zone's position with dots at root notes and octave labels (C2-C6). The currently active zone is highlighted with a brighter color and an enlarged dot
- **Waveform:** Dynamically follows the currently playing note — when you play different notes, the waveform display updates to show the sample from the active zone. When no note is playing, it shows the first zone's waveform

## Recommended Sample Density

Density	Samples/Octave	Max Pitch Shift	Quality
Good	2/octave	±3 semitones	Minor artifacts
Great	3/octave	±2 semitones	Barely noticeable
Excellent	4/octave	±1.5 semitones	Transparent

3-4 samples per octave across 4-5 octaves (12-20 samples total) is the recommended sweet spot.

**Performance:** Multi-sampling has **zero CPU impact** — each voice still plays one sample at a time. The only additional cost is RAM (~2-4 MB per instrument, negligible).

**Returning to Single Sample Mode:** Loading a single sample from the browser (LOAD button or double-click) automatically deactivates multi-sample mode.

## Velocity Layers

Multi-sample instruments support automatic velocity layer detection. Add a velocity suffix to sample filenames:

Suffix	Layer	Default Range
_V1	Soft	0-42
_V2	Medium	43-85
_V3	Loud	86-127

Velocity ranges are auto-calculated based on the number of layers detected. Samples without a `_V` suffix use the full velocity range (0-127). Round-robin cycling works independently per velocity layer.

**Example:** `piano_C3_RR1_V1.wav`, `piano_C3_RR1_V2.wav`, `piano_C3_RR1_V3.wav` creates a C3 zone with 3 velocity layers and round-robin variant 1.

## Sampler Parameters

Parameter	Range	Description
Loop Mode	One-shot, Forward, Reverse, PingPong	Loop playback mode
Loop Start (IN)	0-100%	Loop start point
Loop End (OUT)	0-100%	Loop end point
Crossfade	0-128ms	Crossfade at loop points (anti-click)
Speed	-8x to +8x	Playback speed (center = 1x normal, right = up to 8x, left = reverse up to -8x)
Root Note	0-127 (C-1 to G9)	Root note for pitch-tracking
Fine Tune	-50 to +50 cents	Fine tuning adjustment

## Loop Modes

Mode	Description
<b>One-shot</b>	Play once and stop (no loop)
<b>Forward</b>	Loop forward continuously
<b>Reverse</b>	Loop backward continuously
<b>PingPong</b>	Alternate between forward and reverse

## Sampler Editor (popup window)

- **Waveform Display:** Sample visualization with zoom and scroll
  - Horizontal drag = Scroll (SCRUB)
  - Vertical drag = Zoom in/out
- **Tempo-synced Grid:** Top ruler with bars/beats synced to host BPM
- **Timeline:** Bottom ruler showing seconds
- **IN/OUT Markers:** Draggable directly on the waveform (green = IN, red = OUT)
- **Loop Region:** Yellow zone between markers when loop is active

## Quick Control from Main UI

The **WT Position** slider (above the OSC4/OSC5 module) shows two pointers when the engine is Sampler: - **Green pointer (IN):** Adjusts loop start point - **Red pointer (OUT):** Adjusts loop end point

This allows adjusting loop points without opening the editor.

## Scope Visualization

When OSC4/OSC5 is in Sampler mode, the Scope shows: - Loaded sample waveform (professional min/max style) - IN/OUT markers with colors - Yellow loop region (only when loop is active, not in One-shot)

## Speed Slider Behavior

The Speed slider provides intuitive control over playback direction and speed: - **Center position (1x):** Normal playback speed - **Moving right (1x to 8x):** Accelerated forward playback - **Moving left (-1x to -8x):** Reverse playback with increasing speed - When speed is negative, playback starts from the loop end and moves backward - Works correctly with all loop modes (One-shot, Forward, Reverse, PingPong)

## ROOT TUNE System

The Sampler includes a real-time tuning system to accurately calibrate sample pitch:

Control	Description
<b>Root Note Slider</b>	Adjusts the root note (0-127 MIDI) that corresponds to the sample's original pitch
<b>Fine Tune Slider</b>	Fine adjustment in cents (-50 to +50) for precise tuning
<b>Tuning Indicator</b>	Color-coded display showing tuning state

## Tuning Indicator Colors

Color	Meaning
Green	Perfect tuning - detected pitch matches expected note within 10 cents
Yellow/Orange	Note match - correct note but needs fine tune adjustment
Gray	Neutral - no pitch detection or no match

The tuning indicator displays the current root note name (e.g., “C4”) and cents offset when pitch detection is active. This helps you calibrate samples that don’t have pitch information in the filename.

## Pitch Tracking

The sampler detects the root note from the sample filename (e.g., “Piano\_C4.wav”) as a starting point. You can then manually adjust the Root Note and Fine Tune sliders while playing a reference note to achieve perfect tuning. The tuning indicator will turn green when the sample is correctly calibrated.

## FM/AM in Sampler

- **P.Rat** (Pitch Ratio): Modulates playback speed — replaces FM Index fader. FM/PM/TZFM mode toggle is hidden since all three produce the same effect for sample playback.
- **AM/RM**: Modulates output amplitude (AM) or ring-modulates the output (RM)

## Granular Engine (Mode 9)



Figure 8: Granular editor — waveform display with grain density, size, position and pitch quantization controls.

The Granular engine enables advanced granular synthesis over loaded samples.

## Loading a Sample

1. Select the **Granular** engine on OSC4 or OSC5
2. Load a sample using one of these methods:
  - Click the **SAMPLE** button on the main UI to open the **Sample Browser** (see Sampler section for full browser documentation)
  - Drag an audio file (WAV, AIFF, FLAC, MP3, OGG) to the Granular editor waveform area
- The sample is saved as a WAV file to `~/Documents/Portador Presets/Samples/`

## Granular Parameters

Parameter	Range	Description
Grain Size	1-500ms	Duration of each grain
Density	0.5-500 Hz	Grains generated per second
Position	0-100%	Read position in the sample
Pos Random	0-100%	Position randomization per grain
Pitch Random	0-24 st	Pitch randomization per grain
Pan Random	0-100%	Pan randomization per grain
Direction	-100% to +100%	Forward/reverse grain blend. -100% = all reverse, 0% = 50/50 random, +100% = all forward. Creates lush interference patterns with transient-rich samples
Rnd Tilt	0-100%	Per-grain random window asymmetry. 0% = symmetric windows, 100% = maximum random tilt. Each grain gets a different attack/decay character for organic timbral variation
Window	8 types	Grain window shape
Root Note	0-127 (C-1 to G9)	Root note for pitch-tracking
Fine Tune	-50 to +50 cents	Fine tuning adjustment

## Engine Specifications

- **128 simultaneous grains** for ultra-dense cloud textures
- **Density range 0.5-500 Hz:** from sparse, isolated grains to ultra-dense clouds
- **Grain stealing:** when all 128 slots are full, the grain closest to finishing is replaced — no grain loss
- **Direct spawn:** instant grain cloud on note-on with no gradual build-up delay
- **Independent head timers:** each of the 4 playheads spawns grains on staggered timers for organic textures
- **Multi-head volume preservation:** adding heads increases texture richness without reducing volume

## ROOT TUNE in Granular

The Granular engine includes the same ROOT TUNE system as the Sampler: - **Root Note Slider:** Sets which MIDI note corresponds to the sample's original pitch - **Fine Tune Slider:** Adjusts tuning in cents for precise calibration - **Tuning Indicator:** Shows tuning state with color coding (green = perfect, yellow = note match, gray = neutral)

This is especially useful for granular synthesis where pitch accuracy is important for melodic content.

## Window Types

Type	Description
Hann	Smooth cosine window - natural sound
Welch	Parabolic - smooth transitions
Gaussian	Bell curve shape - very smooth
Triangle	Linear - equal attack and decay
Trapezoid	Plateau with short attack/release
ExpDecay	Instant attack, exponential decay
ExpAttack	Exponential attack, instant release
Raised	Squared raised cosine - very smooth

## SYNC (Tempo-Synced Density/Size)

Two independent SYNC buttons lock grain density and/or size to the host tempo:

- **Density SYNC:** Quantizes grain density to musical beat divisions (1 Bar through 1/32)
- **Size SYNC:** Quantizes grain size to musical beat divisions

When SYNC is active, the slider display changes from Hz/ms to the beat division name. Each can be enabled independently — for example, sync density to tempo while keeping size free, or vice versa.

**Available beat divisions (12):** 1 Bar, 1/2, 1/2D, 1/4, 1/4D, 1/8, 1/8D, 1/8T, 1/16, 1/16D, 1/16T, 1/32.

## QUANT (Scale-Quantized Pitch)

The QUANT button quantizes all grain pitch randomization to the currently selected scale. When active, the Pitch Random parameter will only produce intervals that belong to the scale set in the Chord Generator, ensuring that all grains stay harmonically coherent.

## SCAN FREEZE (Lock Scan Position)

The LOCK button next to the SCAN slider freezes the scan position. When active, the scan playhead stops advancing through the sample — grains continue to spawn from the frozen position. The Position parameter still works, allowing you to manually move where grains are generated.

This is useful for isolating a specific moment in the sample and creating sustained textures from it, while still having full control over grain density, size, and randomization.

## Multi-Playhead (HEADS / Spread Heads)

The Granular engine supports up to 4 simultaneous read playheads, each generating independent grain streams:

Parameter	Range	Description
HEADS	1-4	Number of simultaneous read playheads
Spread Heads	0-100%	Spread distance between playheads across the sample

---

When HEADS > 1, multiple cursor lines appear in the waveform display. Each playhead generates its own grains from its position, effectively multiplying the grain density and creating richer, more complex textures. The Spread Heads parameter controls how far apart the playheads are distributed — at 0% they all read from the same position, at 100% they are maximally spread across the sample.

### **FM/AM in Granular**

- **P.Rat** (Pitch Ratio): Modulates playback speed — replaces FM Index fader. FM/PM/TZFM mode toggle is hidden since all three produce the same effect for sample playback.
- **FM on Grains**: When enabled, modulates each grain's pitch individually (separate from the global P.Rat)
- **AM on Grains**: When enabled, modulates each grain's amplitude individually
- **AM/RM**: Modulates output amplitude (AM) or ring-modulates the output (RM)

# Noise Generator



Figure 9: Noise generator module — stereo width control, color filter and level.

Colored noise source with integrated filtering.

Parameter	Range	Description
Colour	White → Pink → Brown	Noise type
Filter	Low-pass ↔ High-pass	Noise filtering
Level	0-100%	Volume
Stereo Width	0-200%	Stereo image width

## Noise Types

- **White:** Equal energy across all frequencies - bright
- **Pink:**  $-3\text{dB/octave}$  - natural, similar to many sounds in nature
- **Brown:**  $-6\text{dB/octave}$  - dark, like wind or ocean

## Stereo Width

Controls the stereo image of the noise signal using mid-side processing. The horizontal slider below the module label adjusts the width continuously.

- **0%** (left): Mono — both channels receive the same signal. Useful for bass noise layers or when mono compatibility is critical.
- **100%** (center, default): Normal stereo — fully decorrelated left/right channels as generated by the independent L/R noise engines.
- **200%** (right): Super-wide — exaggerated stereo difference. Creates an immersive, spacious sound but sacrifices mono compatibility.

Stereo Width is modulatable via the modulation matrix (destination: “Width” in the Noise submenu), enabling dynamic stereo movement driven by LFOs, envelopes, or macros.

## Tips

- Use 0% width for noise layered with bass to keep the low end centered
- Automate width with an LFO for pulsating stereo movement

- For pitched noise, use the NoisePitch engine (Osc1-5, engine type 7) which supports all Sources including arpeggiator

# Gate Generator



Figure 10: Gate Generator module — track selector, mode display, fader row and per-step controls.

Generates rhythmic gate CV patterns that can be routed through the modulation matrix to trigger envelopes, control the Harmonizer gate, or modulate any parameter. Three modes offer different approaches to gate pattern creation. Per-step velocity controls the amplitude of triggered drums and modulation depth.

## 5 Independent Tracks (Polymetric)

The Gate Generator has **5 independent tracks** (T1-T5), each with its own pattern, mode, time division, and parameters. This enables polymetric rhythms — for example, T1 running a 2-bar Euclidean pattern while T2 runs a 1-bar Random pattern, creating evolving rhythmic textures.

Each track has independent: - **Mode** (Euclidean, Random, Manual) - **Time** (cycle length — different time divisions per track create polymetric patterns) - **Resolution** (step density multiplier) - **Hold, Pulses/Density/Prob** (slot 3, mode-dependent), **Speed** (slot 4, all modes), **Swing, Rotate, Velocity - Evolve** + **Lock** (top-right of editor, mode-dependent — see below) - **Retrigger** (on/off per track — in editor)

### Track Colors

Track	Color
T1	Orange
T2	Yellow
T3	Pink
T4	Red
T5	Purple

## Controls

Control	Description
RUN	Starts/stops the gate generator. Blinks at quarter-note rate when active
LED	Shows real-time gate state (on = gate high)
TRACK	Selects the active track (T1-T5). All faders and controls switch to the selected track
MODE	Selects between Euclidean, Random, and Manual modes for the active track
FREE/HOST	Sync mode toggle. <b>HOST</b> : syncs to DAW transport (play/stop/position). <b>FREE</b> : runs independently with internal clock — useful for standalone or when you want the gate generator running without DAW playback
Rewind	Resets the free-run clock to the beginning (FREE mode only)
Play/Pause	Starts/pauses the internal clock (FREE mode only). Green = playing, gray = paused
TRIG HARM	Opens a menu to assign the active track as the chord sequencer trigger (see TRIG HARM section below). Shows the current mode (Linear, P.Pong, Random, Shuffle) or “Off”
OPEN	Opens the Gate Generator Editor — a circular step sequencer for visual pattern editing

## Fader Row (6 faders — slot 3 changes per mode, all others fixed)

The GatePadre editor now exposes a fixed 6-fader row plus a top-right container with the **Evolve** fader and the **Lock** button:

Slot	Fader	Manual	Random	Euclidean
1	<b>Length</b>	Cycle length	Cycle length	Cycle length
2	<b>Hold</b>	Default gate width	Center hold duration	Gate width % of step
3	<b>dynamic</b>	<b>Prob</b> — per-step probability	<b>Dens</b> — density per cycle	<b>Pulses</b> — Bresenham hit count
4	<b>Speed</b>	Speed Offset (bipolar)	Speed Offset (bipolar)	Speed Offset (bipolar)
5	<b>Swing</b>	Step shuffle	Step shuffle	Step shuffle
6	<b>Rotate</b>	Pattern rotate	Pattern rotate	Pattern rotate

## Top-right container (above the embedded scope)

Control	Active in	Description
<b>Evolve</b>	Manual + Random	Per-cycle mutation amount. In Manual it toggles steps + drifts vel/hold around the user pattern; in Random + Lock it morphs the locked snapshot. Dimmed in Euclidean.
<b>Lock</b>	Random only	Freezes the current Random pattern as the reference. Evolve then morphs away from the reference (lower Evolve = closer to the locked snapshot). Toggling Lock off resumes fresh-each-cycle behaviour. Dimmed in Manual + Euclidean.

All faders update their color to match the active track's color (orange for T1, yellow for T2, etc.).

## Speed Offset (slot 4, all modes)

Bipolar  $-100\%$  ..  $+100\%$ , default 0. Continuously distorts the cycle's internal rate so steps accelerate or decelerate within the bar. The underlying integration is a tempo multiplier  $2^{\text{speed}}$  (0.5x at  $-100\%$ , 2x at  $+100\%$ ) blended with a gentle within-cycle shape (notes rush toward the end at positive values, linger at the start at negative values). The warped phase wraps independently of the host bar so every warped cycle still fires every step exactly once, in order — no silenced steps, no merged gates, even at the extremes.

## Random Lock + Evolve (HBloom-style “random with sense”)

In Random mode, **Lock** captures the currently-playing pattern as a frozen reference. With **Evolve = 0** it replays the reference exactly each cycle. As you raise **Evolve**, each cycle the reference is copied and then perturbed by an XOR-mask scaled by the Evolve value:

- Active-flag flip probability:  $\text{Evolve} \times 0.30$
- Velocity drift:  $\pm \text{Evolve} \times 0.40$
- Hold drift:  $\pm \text{Evolve} \times 0.60$

Lower Evolve back toward 0 and the pattern returns close to the locked reference. Toggle Lock off and Random regenerates fresh each cycle as before. Changing **Length** or **Resolution** while Lock is on recaptures at the new step count.

**Why this matters:** lock a Random pattern you like, then automate Evolve from a slow LFO or macro fader for evolving-but-anchored rhythmic textures.

## Additional Parameters

Parameter	Description
<b>Prob</b>	Probability (0-100%). Each step has this chance of firing. At 100%, all steps play
<b>Rotate</b>	Shifts the pattern clockwise by N steps (0-63). Useful for creating variations
<b>Accent Level</b>	Global gain multiplier applied to all gate CV output

## Step Resolution

Each track has an independent **Resolution** setting that multiplies the number of steps within the TIME cycle:

Resolution	Multiplier	Steps per bar	Description
<b>x1</b>	1.0x	16	Standard 1/16 note grid
<b>x2</b>	2.0x	32	1/32 note grid — double density
<b>x3</b>	1.5x	24	Triplet grid — 1/16 triplets
<b>x4</b>	4.0x	64	1/64 note grid — maximum density

Resolution is selected from a popup menu (click the Resolution button in the editor center panel). The menu shows each option with a description. Higher resolutions create denser, more detailed patterns — useful for hi-hat rolls, ratchets, and granular rhythmic effects.

## Per-Step Velocity

Every step has an individual **velocity** value (0-100%) that controls: - **Drum trigger amplitude**: When Gate Gen triggers drums via Trig In routing, velocity directly scales the drum hit volume - **CV modulation depth**: Velocity scales the gate CV output for modulation destinations

Velocity replaces the old binary accent system with continuous control. Default velocity for new steps is 100%.

### Editing velocity

- **Double-click** a step to cycle through velocity presets: 100% → 70% → 40% → 100%
- **Scroll wheel** on the Step Display VEL area to fine-adjust in 5% increments
- In the editor ring, velocity is shown as the alpha/brightness of each step's gate arc — dimmer gates have lower velocity

## Euclidean Mode

Distributes a number of pulses evenly across the steps using the Bresenham algorithm — the same approach used in Portador's arpeggiator Euclidean mode. Steps are quantized to the current resolution grid within the TIME cycle.

- **Pulses**: Controlled by Fader 2, from 1 to total steps
- **Hold**: Controlled by Fader 3, from 1% to 100% of each step
- **Swing**: Shifts odd-numbered steps forward
- In the Editor: double-click an active pulse to cycle velocity, drag to extend per-step hold

## Random Mode (Musical Random)

Generates a new random pattern at the start of each cycle using **musically-weighted probability**. Instead of flat random distribution, steps on strong beat positions (downbeats, backbeats) have higher probability of being active, producing patterns that feel rhythmically grounded. Velocity is also weighted by metrical position — downbeats get higher velocity, off-beats get lower.

- **Dens** (Density): Higher values = more steps active per cycle. The algorithm guarantees a minimum of 2 steps and places beat boundaries first before filling remaining positions by metrical weight
- **Hold**: Controls average gate width, with random variation (50%-100% per step)

- Read-only in the Editor (pattern regenerates each cycle)

## Manual Mode (Step Editor)

Draw gate patterns directly in the circular step sequencer editor:

- **Click** an empty step to create a gate (renders as one cell wide regardless of the global Hold value, then click+drag to extend)
- **Click + drag** on a gate to extend its hold (arc grows clockwise, multi-step supported)
- **Double-click** a gate to cycle velocity: 100% → 70% → 40% → 100%
- **Right-click** a gate to delete it (or, on the TRIG HARM track, to open the Chord Assign / Skip context menu)
- **Clear** button removes all steps
- **Random** button generates a musically-weighted random pattern (beat-aware distribution, minimum 2 steps)
- **Evolve** (top-right fader, NOT in the main fader row): per-cycle mutation. At 0%, the pattern is static. Higher values introduce small random changes each loop — step toggles, hold variations, velocity drift — creating gradually evolving patterns. Lock is dimmed in Manual mode (Lock is Random-only).

## Gate Generator Editor (GatePadre)

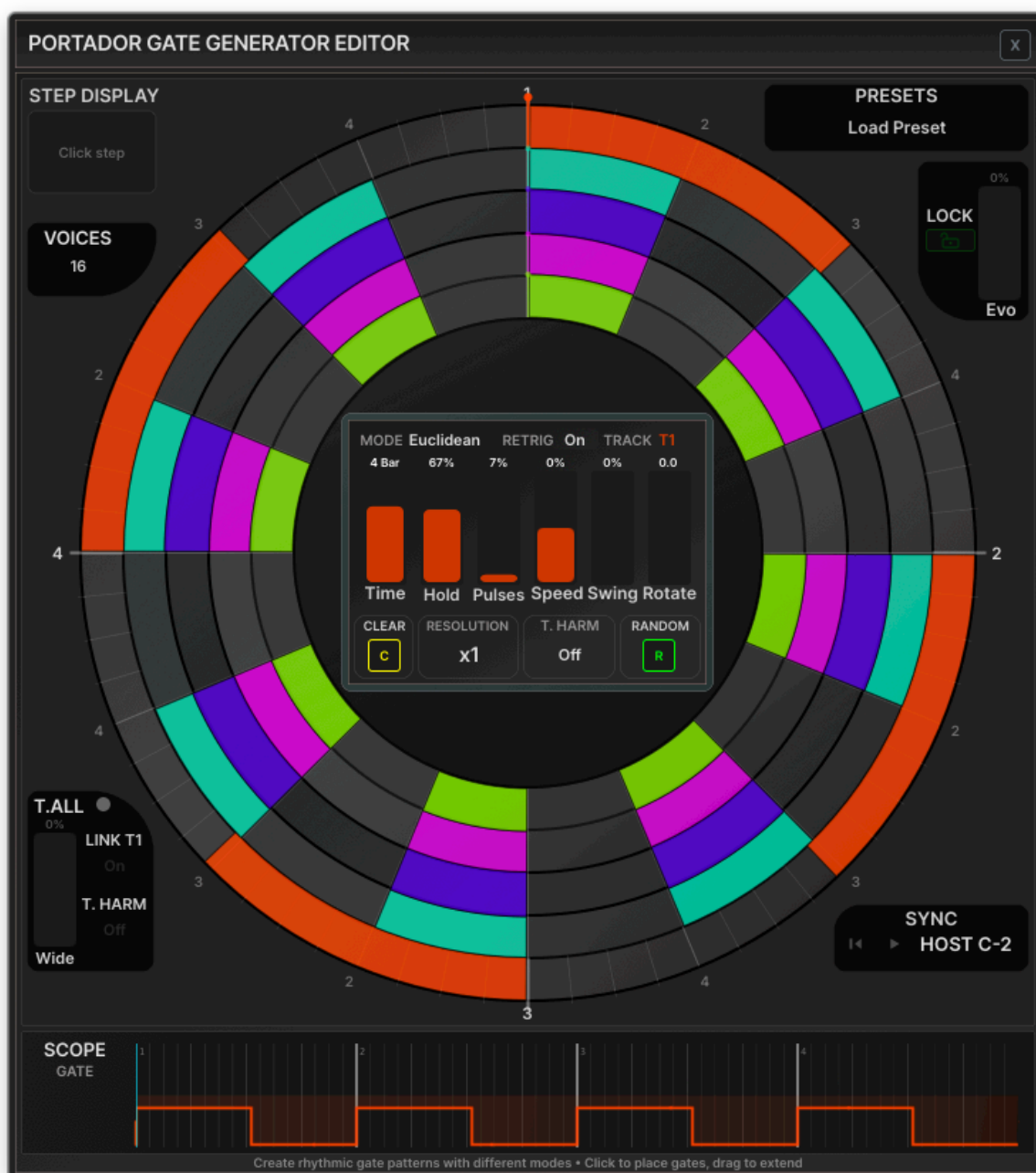


Figure 11: GatePadre — circular polymetric step sequencer with 5 concentric track rings, per-step velocity, per-track playheads and bar numbering.

Click **OPEN** to launch the Gate Generator Editor — a floating circular step sequencer window. The editor shows:

- **5 concentric rings:** Each ring represents one track (T1 outermost, T5 innermost). Click any ring to select that track — all faders and controls in the center panel switch accordingly.
- **Circular gate display:** Pie-slice gates around a clock face. Each track displays gates in its track color. Gate brightness/alpha reflects per-step velocity — brighter gates have higher velocity.
- **Per-ring playheads:** Each track now has its own mini playhead drawn inside that track's ring, in the track's color, advancing at the track's own cycle rate. This makes polymetric phasing immediately

visible — at a glance you can see how T1 (e.g. 2 bars) drifts against T2 (e.g. 1/2 bar). The selected track also gets a larger needle that runs from center to the outer circle.

- **Bar numbers:** Outside the outer circle at each bar boundary
- **Per-ring step grid:** Each ring now draws its OWN radial step grid sized to that track's step count — faint at every step, medium at beat boundaries, bright at bar boundaries. Different TIME / Resolution per track produce different grids in different rings simultaneously.
- **Step Display:** Shows the currently selected/hovered step's info: **STEP** number, **VEL** percentage, and **HOLD** percentage. Scroll wheel on the VEL area adjusts velocity in 5% increments; scroll on the HOLD area adjusts hold duration. Double-click VEL to cycle presets (100/70/40).
- **Center panel** (6-fader row): MODE, RETRIGGER, Resolution, **Length / Hold / dynamic (Pulses or Dens or Prob) / Speed / Swing / Rotate**, Clear/Random buttons — all controls apply to the selected track. The dynamic slot 3 swaps label per mode.
- **Top-right container** (above the embedded scope): **Evolve** fader and **Lock** button (see Evolve / Lock sections above for behavior; Lock is Random-only, Evolve is dimmed in Euclidean).
- **Resolution button:** Opens a popup menu to select step resolution (x1/x2/x3 Triplet/x4). Text color matches the selected track's color on hover.
- **Preset button** ("Load Preset"): Opens the preset menu for saving and loading gate patterns (see Presets section below)
- **Scope:** Embedded gate waveform scope at the bottom, displaying the selected track's output

The editor works as a floating window above the plugin, with drag-to-move on the title bar.

**Polymetric visualization:** When tracks have different TIME values, each ring shows a different number of steps. For example, T1 at 2 bars shows 32 steps in its ring while T2 at 1/2 bar shows 8 steps — both running simultaneously. Resolution further multiplies step count per ring.

## VOICES — Per-preset Voice Budget

In the top-left of the Gate Generator Editor, below the Step Display, the **VOICES** menu caps the number of simultaneous voices the gate dispatchers can allocate from the shared voice pool. This protects CPU under dense patterns with heavy per-voice engines (Cloud, Sampler, Granular) and lets each preset tune its density intent independently from the global POLY setting.

**Values:** 4 / 8 / 12 / 16 / 24 / 32 / 64 (default: 16).

### What it caps

- **CS / TRIG ALL allocations:** each ring's step-fire event checks the budget before allocating a voice. If the cap is reached, the pulse is dropped on that ring while the other rings still get their chance (fair per-ring scheduling).
- **TRIG HARM allocations** (per-track and global Linear / PingPong / Random / Shuffle): each rising edge checks the budget. If the cap is reached the entire rising edge is dropped cleanly — no chord advance, no envelope retrigger, no partial chord. Falling-edge release still fires normally when the gate closes, so existing voices release naturally.

**Effective cap:**  $\min(\text{VOICES}, \text{POLY})$ . Selecting 64 with POLY = 16 effectively caps at 16 — VOICES can never exceed your polyphony setting, so you can leave it high on a light preset without side effects.

### Choosing a value

- **4-8** — Ultra-tight. Single-voice arp feel, rhythmic and dry. Good for tight drum-and-bass style sequences with heavy engines.
- **12-16** — Balanced. The default 16 works for most presets.

- **24-32** — Dense clouds. Good for Cloud / Granular patches with long env R where you want audible overlap between gate hits.
- **64** — Uncapped (equivalent to no budget). Useful on powerful machines for extreme texture density, or on lightweight presets where gate gen is the only voice source.

**Why this exists:** dense Euclidean patterns on TRIG ALL / TRIG HARM with heavy-engine patches can saturate the 64-voice pool in seconds. Once saturated, new gate pulses either drop silently or trigger expensive voice-stealing cross-fades, both of which overload the audio thread. The budget gives each preset its own ceiling. Internally it is paired with envelope-inaudibility reclamation — voices whose envelope output stays below -60 dB for more than ~15 ms are recycled early so the pool rotates faster. Together these keep CPU bounded even on dense arps with long releases.

**Per-preset:** the VOICES value is saved in the preset. A cloud preset can ship with 24-32 while a percussion preset can leave it at 16, without the user having to tune it per session.

## Gate Gen Presets

The Gate Generator has its own preset system for saving and loading complete gate patterns across all 5 tracks.

**Preset menu** (click “Load Preset” in the editor): - **Factory:** Pre-built presets organized by category (submenu opens on hover). Categories include rhythmic styles like Foundations, Syncopated, Dense, Sparse, Euclidean, World, Polymetric, and Evolving. Each category has a colored dot (orange). - **User:** Your saved presets (blue dot). Hover to open the submenu listing all user presets. - **Save:** Overwrites the currently loaded preset (green dot) - **Save As...:** Opens a text input to name and save a new user preset (teal dot)

**What presets store:** All 5 tracks’ parameters (mode, time, hold, pulses, density, swing, prob, rotate, resolution, evolve, **speed**, **random\_lock**) plus every step’s state (on/off, hold, velocity, chord assign, skip count) for all tracks.

**Preset directory:** ~/Documents/Portador Presets/GateGen/ - **Factory/** — Built-in presets organized in category subdirectories - **User/** — User-created presets

## Matrix Routing

Gate Generator occupies **Row 7** in the modulation matrix with a distinctive color. Unlike other matrix rows, Gate Gen cells are **binary** — a single click toggles full on/off (no intermediate levels).

**Per-cell track assignment:** Right-click any Gate Gen cell in the matrix to assign it to a specific track (T1-T5). The cell’s color changes to match the assigned track color: - **Orange** = T1, **Teal** = T2, **Violet** = T3, **Magenta** = T4, **Lime** = T5

This means different matrix columns can receive gate CV from different tracks, enabling complex polymetric routing. For example, route T1 (Euclidean 2-bar) to Env1 Gate In and T2 (Random 1-bar) to Filter Cutoff simultaneously.

**Velocity and drum triggers:** When a Drum-engine oscillator has `Source = Gate T_N`, the per-step velocity of that track directly controls the drum hit amplitude — a step at 40% velocity produces a softer hit than a step at 100%. This uses the osc’s Source setting (no matrix wiring needed); see the Source section above for details.

### Typical matrix routing destinations

- **Env1 Gate In / Env2 Gate In / VENV Gate In** — Trigger envelopes rhythmically
- **H Gate In** — Trigger chord changes from gate pattern

- **Filter Cutoff** — Create rhythmic filter gating effects
- **Channel Pan / FX Send** — Rhythmic spatial effects
- (*Drum-trigger destinations “Osc1-5 Trig In” were removed — use `Source = Gate T_N` on the drum oscillator instead, see Generators section.*)

**Note:** Gate In destinations (Env1/Env2/VENV/H Gate In) require the CV signal to exceed 0.5 threshold. Gate Gen’s binary toggle ensures full-strength triggering.

## TRIG HARM — Gate-Triggered Chord Sequencer

TRIG HARM is one of Portador’s most powerful features: it turns the Gate Generator into an automatic chord sequencer. Each time a gate step fires, it advances to the next degree of the chord loaded in the Chord Engine, creating rhythmic harmonic progressions without playing any MIDI notes.

**Activating TRIG HARM:** In the GatePadre editor, click the **TRIG HARM** button to open a popup menu with 5 options:

- **Off:** Disables TRIG HARM. MIDI controls chords normally.
- **Linear:** Gate steps trigger chords sequentially (1→2→3→4→1→2→...).
- **Pingpong:** Gate steps trigger chords forward and backward (1→2→3→2→1→2→...), without repeating the boundary chord.
- **Random:** Each gate step triggers a random chord from the available slots.
- **Shuffle:** All available chords play once in random order before repeating (like shuffle on a playlist).

Only one track can have TRIG HARM active at a time. The button displays the active mode name (e.g., “Linear”, “Shuffle”).

### How it works

1. Load a chord or harmonic configuration in the Chord Engine (e.g., Chord All with a 4-note chord, or Free Chord with up to 12 chords).
2. Enable TRIG HARM on a Gate Generator track by selecting a mode (Linear, Pingpong, Random, or Shuffle).
3. Each time a gate step produces a rising edge (activates), the Chord Engine advances to the next chord degree according to the selected mode.
4. Each time the gate step produces a falling edge (deactivates), the notes are released.

**Per-Step Chord Assign:** In Manual mode, right-click on any step of the TRIG HARM track to open a context menu. Under **Chord Assign**, select a specific chord slot (shown by note name) or **Auto** to follow the current mode. When a step has a chord assigned, it always triggers that specific chord regardless of the mode — the mode’s sequence position is not advanced. This allows anchoring certain steps to specific chords while the rest follow the pattern. Steps with chord assign show the note name on the step arc.

**Per-Step Skip:** In the same right-click context menu, under **Skip**, select how many cycles to skip after the step plays (x1 to x7). The step always plays on the first cycle, then skips the configured number of cycles before playing again. For example, Skip x3 means: play, mute, mute, mute, play, mute, mute, mute... This is useful for creating variations where a chord appears only every few bars. Steps with skip show an amber “S{N}” indicator on the step arc. Select **Off** to disable skipping.

**Delete:** The right-click context menu also includes **Delete** to remove the step (same as right-clicking on non-TRIG HARM tracks).

## Envelope behavior

TRIG HARM behaves exactly like a MIDI note regarding envelopes: - **Without Gate In connected:** Global envelopes (Env1, Env2, VENV) trigger and release automatically with each gate step, just like pressing and releasing a MIDI key. - **With Gate In connected:** If Env1/Env2/VENV have their Gate In connected in the matrix, TRIG HARM does NOT trigger them. Envelopes are controlled independently via their dedicated Gate In. This allows decoupling the harmonic rhythm from the envelope rhythm.

## Voice allocation

TRIG HARM creates real voices in the synthesizer, just like playing MIDI: - With **source Chord All:** Multiple voices are created for each chord degree (one voice per chord note). - With **source Arp:** The arpeggiator receives the notes and processes them according to its configured pattern. - With **source MIDI or Chord1-5:** A single voice is created with the chord notes stored internally.

**CHORD REPEAT (Per-degree repetition):** Each chord degree can be configured to repeat a number of times (x1 to x8) before advancing to the next degree. Right-click a degree button to access the CHORD REPEAT section and select the count.

**REPEAT VARIATION:** When a degree has repeat count > 1, a REPEAT

VARIATION section appears in the same right-click menu with 6 modes (displayed in cyan):

Mode	Effect
<b>Static</b>	No variation — chord repeats identically (default)
<b>Inversion Cycle</b>	Each repetition rotates the chord inversion (root → 1st → 2nd → ...)
<b>Humanize Cresc</b>	Strum spread grows from tight (30%) to full across repetitions
<b>Velocity Up</b>	Crescendo — velocity ramps from 60% to 100%
<b>Velocity Down</b>	Diminuendo — velocity ramps from 100% to 60%
<b>PLR Transform</b>	Neo-Riemannian cycle: P→R→L→P→R→L (returns to origin after 6 transforms)

*PLR Transform* applies the three fundamental Neo-Riemannian operations (Parallel, Relative, Leading-tone exchange) to create smooth chromatic chord progressions. Each transform moves only one note by 1-2 semitones. Example with x7 starting from C: C → Cm → Eb → Gm → G → Em → C. For 4/5-note chords, PLR transforms the triad and recalculates the 7th (major→Maj7, minor→m7). PLR is only available for chords with a major or minor triad base — it is hidden for diminished, augmented, suspended, and quartal chords.

The **Chord Display** and **Piano Roll** in the HarmPadre editor update in real-time to reflect the varied chord during PLR and Inversion Cycle transformations.

**MIDI-Driven Repeat (DAW / HOST mode):** Chord Repeat and Repeat Variation also work without the gate sequencer. When you play or paint the **same MIDI note consecutively**, the repeat counter advances automatically:

- Play C C C with x3 PLR → CMaj7, Cm7, Gbmaj (P, then P+R transform)
- Play C D C → each note is the first hit of its degree, no variation applied
- Play C C C C → cycles back: original, P, P+R, P+R+L

This allows using PLR, Inversion Cycle, and other repeat variations directly from your DAW's piano roll or a MIDI controller, without needing SYNC FREE or TRIG HARM. The same repeat count and mode settings configured per-degree apply to both gate sequencer and MIDI input.

**PER-DEGREE ARP MODE:** Each chord degree can use a different arpeggio mode, overriding the global arp mode setting. Right-click a degree button and select **Arp Mode** to choose a specific mode for that degree, or select **Global** to follow the global setting.

When a degree has an arp override active, a colored amber line appears at the top of the degree button. The arp mode submenu uses the same category structure as the main arp mode menu (Directional, Random, Contour, Pattern, Algorithmic).

Only the **mode** changes per degree — all other arp parameters (rate, range, gate, steps, ratchet, slide, hold, evolve) remain global and apply uniformly.

**Example:** Set Degree I to Up (clean ascending), Degree IV to Additive (gradual expansion), Degree V to Euclidean (rhythmic), and Degree vi to Random (tension). This creates contrasting arpeggio characters for different harmonic areas within the same progression.

**PER-DEGREE ARP SPREAD:** Each chord degree can have its own

arpeggio spread setting, controlling how wide the notes are distributed across octaves. Right-click a degree button and select **Arp Spread** (or **Arp2 Spread** for the second arpeggiator) to choose from four levels:

Spread	Effect
<b>Close</b>	Notes stay in their original octave (default)
<b>Semi</b>	Some notes shifted +/-1 octave (half-octave max shift)
<b>Open</b>	Notes spread +/-2 octaves
<b>Wide</b>	Maximum spread +/-2-3 octaves

This simulates how a pianist plays arpeggios — close voicing (one hand position) versus open voicing (crossing fingers across octaves). Voice leading works naturally between spread levels: when transitioning from a Close degree to an Open degree, common tones are kept at the nearest octave for smooth transitions.

**Example:** Set Degree I to Close (stable, grounded), Degree IV to Semi (slightly open), and Degree V to Wide (maximum tension and movement). This creates dynamic register contrast across a chord progression.

The degree right-click menu (titled **PER DEGREE**) is organized into categories with submenus:

Category	Function
<b>Anchor</b>	Reset voice leading state when this degree plays
<b>Bass Note</b>	Select bass override (Root, C-B) for slash chords
<b>Tritone Sub</b>	Enable/disable tritone substitution (dominant chords only)
<b>Chord Repeat</b>	Set repeat count (x1-x8) and variation mode
<b>Arp Mode</b>	Per-degree arp mode override (Global or specific mode)
<b>Arp Spread</b>	Per-degree arp note spacing (Close, Semi, Open, Wide)
<b>Arp2 Mode</b>	Per-degree arp2 mode override (Global or specific mode)
<b>Arp2 Spread</b>	Per-degree arp2 note spacing (Close, Semi, Open, Wide)

**Voice Leading:** When voicing is set to one of the five **Auto** modes (Smooth, Classical, Jazz, Modern, Piano), Portador automatically applies voice-leading algorithms to create smooth chord transitions. **Auto Piano** is the default and models a real pianist — pedal bass point, anchored soprano, inner voices move by semitones. Auto Smooth minimizes total movement, Auto Classical follows SATB rules (no parallel 5ths/8ths), Auto Jazz prioritizes guide-tone voice leading, and Auto Modern keeps upper voices static for ambient textures. Each style also adapts the finger voicing to the chord type and register (see Adaptive Finger below).

**Combination with Arp:** TRIG HARM can coexist with the arpeggiator. If one oscillator uses source Arp and another uses source Chord All, both sound simultaneously — the arp receives notes from the chord engine while chord voices are generated in parallel.

### Automatic MIDI Split with TRIG HARM

When TRIG HARM is active, Portador automatically separates sound sources:

- **Oscillators with Chord/Arp source** (Chord All, Chord1-5, Arp, etc.): Sound ONLY with voices generated by TRIG HARM. MIDI keyboard notes do NOT generate new chords.
- **Oscillators with MIDI source:** Sound ONLY with notes you play on the keyboard. TRIG HARM voices do NOT play through these oscillators.

This enables **real-time improvisation over generative harmonies:** the Gate Generator controls the chord progression on some oscillators while you play freely on others, without the two sources interfering with each other.

MIDI notes are completely independent of the Chord Engine while TRIG HARM is active — they don't change the root, don't modify chord state, and don't interfere with the gate generator's chord sequence.

### Generative music + improvisation example

1. Set Osc1 source to "Chord All" (generative chords).
2. Set Osc2 source to "MIDI" (free melody).
3. Load a chord or scale in the Chord Engine.
4. Create a Euclidean pattern in Gate Generator Track 1.
5. Enable TRIG HARM on Track 1 with SYNC FREE.
6. Osc1 plays chords automatically following the gate pattern.
7. Play notes on the keyboard: only Osc2 responds, with its own envelope and timbre.

### Basic TRIG HARM example

1. Load a CMaj7 chord in Chord Engine (C-E-G-B).
2. Set Osc1 source to "Chord All".
3. In Gate Generator, create a Euclidean pattern on Track 5.
4. Enable TRIG HARM on Track 5 (select "Linear" mode).
5. Press Play: each gate hit advances through C→E→G→B→C→... creating an automatic rhythmic harmonic progression.

### Chord Assign example

1. Load 4 chords in Free Chord mode (C, Am, F, G).
2. Create a Manual pattern with 8 steps on the TRIG HARM track.
3. Right-click step 1 → Chord Assign → C. Right-click step 5 → Chord Assign → C.
4. Leave the other steps on Auto (they follow the mode).
5. With Linear mode: steps 1 and 5 always play C, while the rest cycle through Am, F, G linearly. This creates a progression where C is the anchor chord.

## Shuffle + Skip example

1. Load 3 chords in Free Chord mode.
2. Enable TRIG HARM in Shuffle mode on a Manual track.
3. Right-click step 3 → Skip → x2.
4. Step 3 plays on the first cycle, then is silent for 2 cycles, then plays again. This creates a pattern where one chord appears less frequently, adding variation to the harmonic rhythm.

**Right-click context menu in Euclidean mode:** Chord Assign and Skip also work on Euclidean tracks when TRIG HARM is enabled — right-click any algorithmic pulse to pin a specific chord on it or to make it skip N cycles. Unlike Manual mode, the right-click does NOT delete the step in Euclidean (steps are algorithmic and would regenerate on the next cycle anyway). This means you can build a Bjorklund rhythmic skeleton with Pulses + Hold + Speed + Swing, then surgically pin specific chords on key downbeats while letting the rest follow your TRIG HARM mode (Linear / P.Pong / Random / Shuffle).

## TRIG ALL — Chord Spread (Spiral Voice Allocation)

TRIG ALL turns the five Gate Generator tracks into the five voices of the currently active chord — Track 1 (outermost ring) plays voice 1 (root), Track 2 plays voice 2, and so on up to Track 5 plays voice 5. Each ring keeps its own gate pattern and timing, so when you apply **Wide** the voices spread in time around the cycle like a spiral arpeggio. Unlike per-track TRIG HARM (which drives chord advance from one track while the others do percussion), TRIG ALL is a **global** mode: all five tracks are committed to chord duty, and the result is a single coherent spiral of the chord degree.

**Activating TRIG ALL:** In the GatePadre editor the bottom-left panel has a container labeled **TRIG ALL**. Toggle the dot next to it to turn the mode on or off. When it turns on, any track currently in **Random** mode is automatically frozen to **Manual** using the random buffer as a starting pattern — this keeps the spiral coherent cycle to cycle (Random's per-cycle regeneration would otherwise break the voice alignment). Your other per-track settings are preserved. Turn it off and the tracks stay in whichever mode they ended up in; a global cleanup releases all Chord Spread voices cleanly.

**Chord advance (T. HARM menu inside TRIG ALL):** Below the toggle, the **T. HARM** menu controls how the chord degrees advance across gate hits — same menu as per-track TRIG HARM but scoped to the global CS dispatcher:

- **Off:** No auto-advance. TRIG ALL fires the voices of the currently-triggered chord on every gate hit. Good for held textures over a single degree.
- **Linear:** Degrees cycle 1→2→3→4→...→1. Step 1 is always degree 1 on the first cycle, so with *N* chords the pattern loops every *N* gate hits.
- **PingPong:** Degrees oscillate 1→2→3→2→1→2→... without repeating the boundary.
- **Random:** Each gate hit picks a random active degree.
- **Shuffle:** All degrees are played once in random order before the order reshuffles.

Per-step **Chord Assign** (right-click a step on Track 1) pins a specific degree to that step — the advance skips past it and uses the pinned slot. Any step that has an assign shows the note name on the outer ring along with the degree color (see below).

**Per-ring direction (Forward / Backward):** With TRIG ALL active the per-track TRIG HARM button on each ring's control panel is re-purposed to select the ring's playhead direction — **Forward** or **Backward**. The label above the button switches to **DIRECTION** to signal the change. A ring running Backward reads its gate pattern CCW, which reverses that voice's contribution to the spiral; combined with Wide it can produce descending arpeggios on that voice while the others ascend. The main-UI per-track Trig Harm button dims to signal it is inert while CS is on — direction lives on the ring control.

**Wide (spiral spread):** The **Wide** fader at the bottom of the TRIG ALL container angularly spreads each ring forward in the cycle by  $\text{trackIndex} \times \text{wide} / \text{kNumTracks}$ . Wide = 0 collapses the spiral to a single

chord strike (all rings fire together — a pad moment). `Wide = 1` spreads the outer ring almost a full cycle after the inner ring. Values in between produce anything from tight rolls to wide arpeggios. `Wide` is cyan-colored to signal it is a *global* CS parameter, not a Track 1 control. It is a full modulation destination in the matrix ( `chord_spread_wide` ), so LFOs, envelopes, macros, and CV columns can animate the spiral over time.

**LINK T1 (coherent vs polyrhythmic):** The **LINK** menu (above T. HARM, where the old HEADS button lived) has two states:

- **On** (default): Rings 2-5 inherit Track 1's **Time** and **Resolution**. The whole spiral shares one cycle length, pulses/hold/swing percentages translate to identical absolute positions across rings, and Sync Chord collisions stay predictable. Rings 2-5's Time fader and Resolution button are dimmed because they are inert in this state.
- **Off**: Each ring runs on its own Time and Resolution. You get polyrhythmic spirals — voices fire at independent rates but still play voice N of the *current* chord (chord advance is Track-1-only, so harmonic coherence survives regardless of how the ring timings drift).

Use LINK On for traditional arpeggio-like spirals where the spread is the interesting variable. Use LINK Off when you want cross-rhythmic textures — each voice with its own pulse count against a shared chord.

**Sync Chord (right-click a step → “Sync Chord”):** Right-click any step on any ring and pick **Sync Chord**. All five rings' chosen step re-align angularly to the clicked step's position. The anchor is the **exact cursor angle** — not a recomputed step start — so the sync lands where you clicked even when rotate, swing, or `Wide` would have placed the step somewhere slightly different. Each ring gets a per-step angular shift ( `anchorAngle - ringBaseAngle` ), normalised to the shortest rotation, so the voices all paint at the same position across rings regardless of per-ring rotate / swing / `Wide` asymmetry. Moving `Wide` away from the value it had at sync time re-spreads the rings naturally from the synced point. Moving `Wide` below the anchor produces a **reverse arpeggio** — outer rings now fire *before* inner rings from the sync point. This is the mechanism that lets you build chord hits inside a running pattern: sync at `Wide = 0.5`, then automate `Wide` down to 0 to morph from a pad strike into a reverse arpeggio and back.

Sync Chord is available on every step of every ring in TA mode. Non-sync'd steps use the natural spiral position (ring offset plus `Wide`). The engine's firing path is unified — every active step on every ring has a single effective fire-phase `stepPos + ringWide + shift / 2π`, so there are never double-fires when `Wide` changes.

**Overlapping chord steps — polyphonic voices:** When two chord-assigned steps on the same ring land at near-identical angular positions (typical after Sync Chord merges two voices onto one spot, or after rotate brings adjacent steps together), each step fires its **own voice with its own envelope**. The voices coexist for the visual length of their holds — the first chord doesn't cut off when the second fires, and each chord's release happens at its own step's endpoint, not at the next step's. This is the polyphonic CS+TA dispatcher: each `StepFireEvent` allocates an independent `SynthVoice` in the pool, tracked by its individual end-PPQ rather than a shared track gate. Useful for dense Bach-style passages, overlapping chord clusters, or any pattern where two chord hits visually stack.

**Paint-then-drag on track 1 (Manual mode):** Painting a new step on Track 1 in Manual mode with CS+LINK on propagates the step's hold, velocity and skip count to rings 2-5 **while you drag to set the hold** — the freshly-painted chord is born coherent across all voices. Once you release, rings 2-5 become independent: right-clicking and hold-dragging any specific ring changes only that ring's voice length, and a subsequent edit of Track 1's step keeps your per-ring customisations intact (only the active state is mirrored after the birth gesture).

**CS Wide modulation while paused:** When an LFO, macro or matrix source drives `chord_spread_wide`, the spiral follows the modulation visually in real time — even when the host transport is stopped. The

GatePadre repaint poll picks up >1 % deltas on the engine's modulated Wide value each tick so you can see modulation wobble on the step placements without needing the sequencer to be running.

### Visual cues on the spiral

- **Per-step chord color:** In T. HARM mode, each painted step on Track 1 is colored by the chromatic degree that will fire there (12-tone HSV palette). The color is predicted from the current advance mode (Linear / PingPong / per-step Chord Assign) and updates live as you edit degrees or change mode. Rings 2-5 inherit the color so each voice of a chord reads in the same hue.
- **Chord name label:** The outer ring shows the chord name (e.g. *CMaj7*, *G7*) rotated to follow the arc of each step. Selecting a step keeps the white selection border without hiding the name beneath it.
- **Dimmed non-active voices:** If the current chord has fewer than 5 voices (most chords do), the rings beyond the chord's voice count fade to near-background alpha. This is per-step — step 1 may play a 3-note chord (rings 3-4 dim) while step 2 plays a 5-note chord (all rings visible).
- **Selected track highlight:** The selected ring's step gets the usual white bordered selection; in CS mode the chord color remains readable on top.

**Humanize bypass:** The per-chord **Humanize** roll-up / roll-down / natural modes are DSP-bypassed while TRIG ALL is active — the spiral owns per-voice onset timing, and layering a humanize delay on top would desync the rings. The APVTS values are preserved; turning CS off restores your humanize exactly.

**MIDI drag-export:** When you drag the *Chord* stream of the MIDI capture into your DAW while TRIG ALL is playing, the notes are placed at their **true per-ring onsets** — the spiral you see is the spiral you hear and the spiral you export. Each ring's noteOn is pushed at its sample-accurate rising edge (Wide and Sync Chord shifts applied); each noteOff at its falling edge. If you pause or deactivate CS mid-note, the dangling note gets a clean noteOff at the stop moment.

**Randomizer (R button) scope:** When TRIG ALL is on, pressing the R (randomize) button at the bottom of the center panel randomizes Track 1's algorithmic params + step pattern *plus* the global **Wide** value. Rings 2-5 are intentionally left alone — they mirror Track 1 (or run independent cycles if LINK is off) so per-ring randomization on them would be inert or fight the mirror.

**Reset Voices:** In TRIG ALL mode the top-right preset menu's **Init** entry becomes **Reset Voices** — clearing every ring's per-step angular shift (undoing Sync Chord) and resetting the spiral to the pure Wide-derived positions. Undo (Cmd+Z) remains available for fine-grained rollback.

### Example: ambient pad → reverse arpeggio

1. Load a 4-chord progression in Free Chord mode (e.g. C, Am, F, G) and set Osc 1 source to Chord All.
2. Enable TRIG ALL. Set T. HARM to Linear. Leave LINK On.
3. On Track 1 paint a 4-pulse Euclidean pattern over 4 bars. Set Hold high enough that each step lasts most of its bar.
4. Start with Wide = 0 — you hear a pad-like chord strike on each of the 4 beats.
5. Right-click step 1 on any inner ring → Sync Chord. Automate Wide from 0 to 0.4: the voices spread forward in time into a soft rolling arpeggio.
6. Automate Wide *below* 0 (down toward the negative side of the anchor): the spiral reverses — voice 5 hits before voice 4, producing a descending arpeggio that returns to a pad at the anchor.

### Example: polyrhythmic spiral

1. Same chord setup as above.
2. Enable TRIG ALL. Set LINK to Off.
3. Track 1: 3 pulses over 2 bars. Track 2: 4 pulses over 2 bars. Track 3: 5 pulses over 2 bars. Track 4: 7 pulses over 2 bars.

- Each ring fires its voice of the current chord at its own rate. Wide at 0 keeps the rings in phase per chord advance; Wide at 0.3 adds a spiral offset inside each chord while the polyrhythm keeps the voices from cluster-firing.

## SYNC — FREE / HOST Mode

The **SYNC** button at the bottom-right of the GatePadre editor (and duplicated in the main UI next to the Gate Generator block) controls the Gate Generator's clock source:

- **HOST** (default): Gate Generator syncs to the DAW transport and is **armed by a dedicated MIDI trigger note**. The button displays `HOST C-2` (or the currently assigned note) in white.
- **FREE**: Gate Generator runs on an independent internal clock. Works without the DAW transport being active. Button text turns green. Useful for:
  - Real-time sound design without pressing Play
  - Creating rhythmic textures independent of project tempo
  - Standalone mode where there is no host
- **LEARN**: a transient assignment mode (see below). The button blinks the word `LEARN` until a note is received or the 10-second timeout expires.

Clicking the SYNC button opens a three-item popup — **HOST**, **FREE**, and **LEARN** — each showing the active state.

### HOST Trigger Note

In HOST mode, the Gate Generator does **not** start just because MIDI notes are being played. Instead, one dedicated MIDI note (default **C-2**) acts as the exclusive on/off switch:

- **Note-on on the trigger note** → Gate Generator starts running, aligned to the host transport
- **Note-off on the trigger note** → Gate Generator stops
- The trigger note is filtered out of the MIDI stream before it reaches the VoiceManager, so it never creates a voice, never triggers envelopes, and never sounds through any oscillator
- Any other MIDI note is routed normally to the synth without affecting the Gate Generator

This is what allows **TRIG HARM to coexist with HOST mode**: the keyboard freely drives chord/arp voices while the gate generator is controlled by its own dedicated note (for example a low C-2 from a MIDI clip that defines the rhythmic grid).

### Assigning a new trigger note (LEARN)





1. Click the SYNC button and choose **LEARN**. The button starts blinking and the assignment window opens.
2. Send a MIDI note-on from your keyboard or DAW within 10 seconds. The first note received becomes the new trigger note.
3. If no note arrives before the timeout, the previous note is kept.

The trigger note is saved with presets and with the DAW session.

**Standalone note**: in Standalone mode the gate generator falls back to FREE automatically and the trigger note is ignored.

## Transport Controls — Play/Pause and Rewind

Two transport buttons appear to the left of the SYNC button:

-  **Rewind**: Resets the sequencer to step 1. Also resets the chord sequence when TRIG HARM is active. Works while playing or while paused.
- **Play /  Pause**: Toggles the sequencer on and off.
  - In **FREE mode**: Fully interactive. When switching to FREE, the sequencer starts **paused** — this gives you time to set up TRIG HARM, paint steps, and configure parameters before starting playback. Click Play () to start; the button turns into a green Pause (). Click Pause to freeze the sequencer at its current position (the clock does not advance). Click Play again to resume from where you left off.
  - In **HOST mode**: The buttons are dimmed and non-interactive. They reflect the DAW transport state (playing or stopped) as a visual indicator.

## Standalone Mode

In standalone mode, Gate Generator uses the internal BPM clock (adjustable in Settings). FREE mode is automatically enabled allowing patterns to run independently.

# Filters

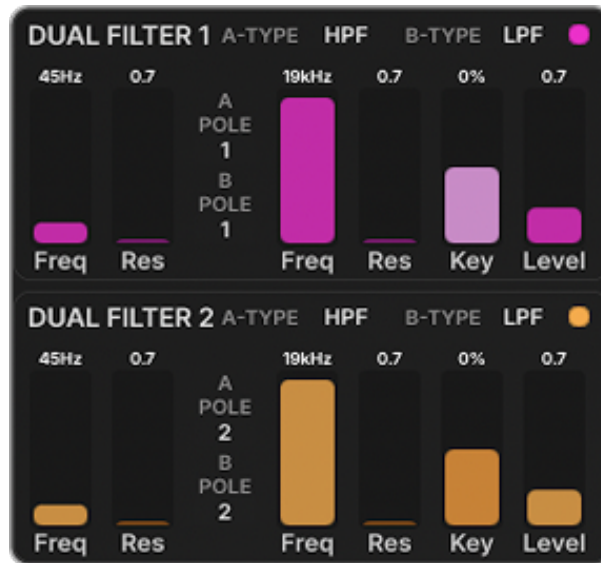


Figure 12: Dual Filter 1 and Dual Filter 2 — each combines an HPF and LPF with selectable type, frequency, resonance and keytracking. 13 filter types share the same fader strip.

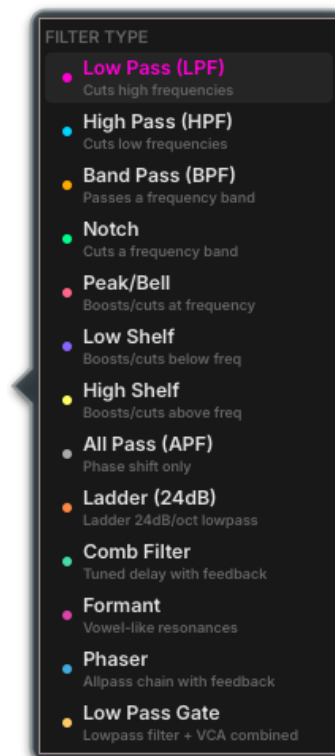


Figure 13: Filter type selection menu — 13 types including LPF/HPF/BPF/Notch/Peak/LShelf/HShelf/APF/Ladder/Comb/Formant/Phaser/LPG.

Portador includes two identical and fully independent dual filter sections. Each filter has two stages (A and B) that can be configured with any of the 13 available filter types, allowing hundreds of possible combinations per filter.

## Filter 1 and Filter 2

Each dual filter has two configurable stages:

### Available Filter Types (13)

Type	Name	Description
LPF	Low Pass	Cuts high frequencies - warm and smooth sound
HPF	High Pass	Cuts low frequencies - thin and airy sound
BPF	Band Pass	Allows only a band of frequencies
NOTCH	Notch	Cuts a specific band of frequencies
PEAK	Peak/Bell	Boosts or attenuates at a specific frequency
L.SH	Low Shelf	Boosts or attenuates everything below the frequency
H.SH	High Shelf	Boosts or attenuates everything above the frequency
APF	All Pass	Only changes phase, without affecting amplitude
LADDER	Ladder	4-pole ladder filter (Huovilainen model) with self-oscillation at high resonance
COMB	Comb	Resonant comb filter — generates pitch based on cutoff. Ideal for metallic effects and Karplus-Strong
FORMANT	Formant	Morphs between 5 vowels (A-E-I-O-U) using 3 BPF filters. Cutoff controls morph position
PHASER	Phaser	Cascaded allpass stages (1-8) with feedback. Creates phase notches that sweep with cutoff
LPG	Low Pass Gate	West Coast filter + VCA combo. Vactrol-smoothed cutoff (10 ms tau) drives both lowpass AND internal VCA simultaneously — when cutoff goes low, level drops too. Morphs from pure VCA (Poles=1) through 2-pole LPF+VCA (Poles=2) to 3-pole LPF+VCA (Poles=3). Route an envelope to cutoff via the matrix to get plucky organic character — the internal VCA follows automatically without needing a separate amp envelope on Channel CTRL

### Parameters per stage (A and B)

Parameter	Range	Description
Type	13 types	Filter type (see table above)
Cutoff	20Hz - 20kHz	Cutoff frequency
Resonance	0.1 - 10.0	Q factor / Resonance (LPG mode capped internally at Q=4 — no self-oscillation)
Poles	1-4	Slope: 6/12/18/24 dB/oct (LPG mode clamps 1-3: Pole 1 = pure VCA, Pole 2 = classic LPG, Pole 3 = 3-pole LPG with more filter bite)

## Global Parameters

Parameter	Range	Description
Keytracking	-100% to +100%	Keyboard tracking
Level/Drive	0-300%	Gain with soft saturation
Stereo Cutoff Offset	-48 to +48 semitones	Offsets the cutoff frequency between left and right channels. Positive values raise the right channel cutoff and lower the left; negative values do the opposite. Creates stereo filter movement and wide, spacious timbres

**Tip:** Combine an LPF with an HPF to create a custom band-pass filter. Use NOTCH to eliminate problematic frequencies. Shelf filters are ideal for tonal equalization.

## Filter Feedback Routing

Since both filters are fully routable via the Matrix, you can create feedback loops between them:

Route	Matrix Cell	Effect
Filter1 → Filter2	Row F1, Col F2 In	Serial filtering (Filter1 output feeds Filter2 input)
Filter2 → Filter1	Row F2, Col F1 In	Cross-feedback loop — creates resonant textures
Filter1 → Filter1	Row F1, Col F1 In	Self-feedback — pushes the filter into self-resonance territory
Filter2 → Filter2	Row F2, Col F2 In	Self-feedback — same effect on Filter2

- All feedback routes use a 1-block delay (~1.5ms at 44.1kHz), which creates the resonant character without instability.
- Per-voice feedback path: each voice has its own independent feedback signal.
- **Warning:** High feedback levels combined with high resonance can produce very loud signals. Start with low matrix levels (10-30%) and increase gradually.
- Combine with LFO modulation on the matrix level for evolving feedback textures.

# Envelopes

## Envelope 1 (Classic ADSR)



Figure 14: Envelope 1 — classic ADSR with attack, decay, sustain and release stages plus invert toggle. Traditional 4-stage envelope with selectable curve types and output inversion.

Parameter	Range	Description
Attack	0.1ms - 10s	Attack time
Decay	3ms - 2s	Decay time
Sustain	0-100%	Sustain level
Release	3ms - 10s	Release time
Attack Curve	Linear, Exp, Log, S-Curve	Attack curve shape
Decay Curve	Linear, Exp, Log, S-Curve	Decay curve shape
Release Curve	Linear, Exp, Log, S-Curve	Release curve shape
<b>Invert</b>	Off / On	Inverts envelope output

## Invert

The **Invert** toggle flips the envelope output **at the source**: it outputs `1.0 - level` instead of the normal level. This means:

- **Attack** goes from 1.0 down to 0.0 (ducking effect at note start)
- **Sustain** holds at `1.0 - sustain level` (inverted sustain)
- **Release** rises from the inverted sustain back up to 1.0

**Important — scope of the inversion:** this toggle flips Env1 **before anything reads it**. It therefore affects simultaneously:

1. The **direct voice VCA** — the non-matrix path that uses Env1 as the main amplitude envelope. This is the key use case: it turns the AMP envelope into a “reverse envelope” (slow fade-in instead of fast attack).
2. **Every matrix cell** that reads the Env 1 row — all routings receive the inverted version at once.

If you only want to invert **a single matrix cell** (without touching the voice VCA or the other cells), use per-cell invert instead: **right-click** the cell in the matrix (red border + ± glyph). See “Per-Cell Invert” in the Modulation Matrix section.

This is useful for: - **Reverse AMP envelope**: Turn Invert On with A=2s → the voice starts silent and fades in over 2 seconds - **Global ducking / sidechain**: With Invert On, Env1 in row 10 is a descending unipolar signal — route it to CH1 Level for per-note ducking - **Filter opening on release**: Route inverted Env1 to filter cutoff — filter opens as the note releases - **Reverse dynamics**: Create sounds that get quieter when you play and louder when you release

## Curve Types

Type	Description
<b>Linear</b>	Straight line - consistent change rate
<b>Exponential</b>	Fast start, slows towards end - natural, musical (default)
<b>Logarithmic</b>	Slow start, accelerates towards end - punchy attacks
<b>S-Curve</b>	Smooth sigmoid curve - gradual transitions

**Tip:** Exponential curves sound most natural for typical synth sounds. Use Logarithmic for punchy attacks or S-Curve for very smooth transitions.

The LED indicates when the envelope is active.

## Gate In (CV-Triggered Envelopes)

Each envelope (ENV1, ENV2, VENV) can be triggered by modulation sources instead of MIDI notes using the **Gate In** matrix destinations.

### How it works

1. In the modulation matrix, select “**Gate In**” as the column destination for the desired envelope (found in the Env 1, Env 2, or Logic submenu)
2. Connect a modulation source (e.g., LFO square wave) to the same column
3. When the signal crosses above 0.5, the envelope triggers. When it falls below 0.5, the envelope releases

### Key behavior

- When Gate In is connected, the envelope **stops responding to MIDI** and only responds to the CV source
- When nothing is connected to Gate In, the envelope works with MIDI normally
- Works perfectly in **drone mode** (no MIDI notes) — oscillators run continuously and Gate In creates rhythmic gating patterns
- Each envelope can have a different trigger source: for example, ENV1 triggered by MIDI, ENV2 and VENV triggered by LFOs
- **Drum trigger**: Gate In also triggers the Drum engine on any oscillator that is in Drum mode — enabling rhythmic drum patterns from LFOs, Gate Generator, or any CV source without MIDI notes

### Creative uses

- **Rhythmic drones**: Connect an LFO square to ENV1 Gate In for tempo-synced drone gating
- **Chained envelopes**: Route ENV1 output to ENV2 Gate In — ENV2 fires when ENV1 reaches its peak
- **Generative patterns**: Use different LFO rates on ENV1 and ENV2 Gate In for polyrhythmic envelope patterns

- **Drum machine from Gate Gen:** Set oscillators to different Drum types, route Gate Gen to Env1 Gate In — instant rhythmic drum patterns

## Envelope Output Modulation (Mod)

Each envelope and Logic module output can be **amplitude-modulated** by any matrix source using the **Mod** matrix destinations. This multiplies the envelope's output signal, creating effects like tremolo, rhythmic gating, or complex waveshaping on the envelope itself.

### How it works

1. In the modulation matrix, select "**Mod**" as the column destination (found in the Env 1, Env 2, or Logic submenu)
2. Connect a modulation source (e.g., LFO, Gate Generator, another envelope, Macro) to the same column
3. The envelope output is multiplied by  $(1 + CV \times \text{depth})$  — at zero depth the envelope is unchanged, positive CV boosts, negative CV attenuates

### Available destinations

Destination	Label	Description
Env 1 Mod	E1 Mod	Modulates ENV1 output amplitude
Env 2 Mod	E2 Mod	Modulates ENV2 output amplitude
Logic Mod	VE Mod	Modulates Logic output amplitude (all modes: VENV, Function, Random, S&H)

### Key behavior

- The modulation is applied **after** the envelope generates its output but **before** it reaches the VCA and other downstream destinations
- The matrix source signals (rows) see the **unmodulated** envelope, preventing feedback loops
- In POLY mode, the modulation is applied **per-voice** — each voice's envelope is modulated independently
- Output is clamped to 0-1 range to prevent clipping
- Works with any matrix source: LFOs, envelopes, Gate Generator, Macros, Evolve

### Creative uses

- **Tremolo on envelope:** Connect a sine LFO → E1 Mod for amplitude modulation that follows the envelope shape
- **Rhythmic chopping:** Connect a square LFO or Gate Generator → E1 Mod for tempo-synced gating of the envelope
- **Envelope-on-envelope:** Route ENV2 → E1 Mod to reshape ENV1's curve with a second envelope
- **Evolving textures:** Connect Evolve → VE Mod for slowly mutating envelope shapes over time
- **Macro control:** Assign a Macro to the modulation depth for performance control over the effect intensity

**Scope visualization:** When Mod is active, the scope switches from parametric display to **real-time captured audio** — you see the actual modulated envelope output as it plays. The display uses progressive decimation to show the full envelope duration regardless of length: short notes show fine detail, long sustains automatically zoom out to fit.

## Envelope 2 (Vector)



Figure 15: Envelope 2 — vector envelope module with time, warp and shape selection. Wavetable-based envelope with 100 predefined shapes (10 banks × 10 shapes).

Parameter	Range	Description
Time	1ms - 16s	Total duration
Warp	-100% to +100%	Time distortion (speeds up/slows down)
Bank	A-J	Shape bank
Shape	0-9	Shape within the bank
Mode	One-shot, Loop	Playback mode
Retrig	Off, On	Reset with each note

### Envelope Banks

- **A-B**: Classic shapes (exponential, linear)
- **C-D**: Complex multi-stage shapes
- **E-F**: Organic curves
- **G-H**: Rhythmic shapes
- **I-J**: Special effects

### User Envelopes

The envelope selector also includes a **USER** category below the factory banks. User envelopes are custom VENV shapes saved via “Save As...” in the Logic VENV editor. When loading a user envelope in ENV2, the per-section fader gains are baked into the waveform for accurate reproduction.

# LFOs



Figure 16: LFO 1 and LFO 2 — first pair, with rate, level, release, sync mode, waveform, polarity and poly options.



Figure 17: LFO 3 and LFO 4 — second pair. Use Pair/Quad Link to chain LFOs into longer modulation phrases.

Four identical LFOs (LFO1-4) with 106 waveforms each. The LFOs are organized in two UI slots:

- **Slot A** — toggles between LFO1 and LFO3 (selector at the top of the module)
- **Slot B** — toggles between LFO2 and LFO4

All parameters of each LFO persist regardless of which one is currently visible, and all four run in audio simultaneously. The slot selector only changes which LFO is displayed.

Parameter	Range	Description
Rate (was “Speed”)	0.01Hz - 100Hz (Free) or divisions (Host)	LFO frequency. The fader was renamed from “Speed” to “Rate” so the label fits the slot in narrow LFO views.
Level	0-100%	Modulation depth
Wave	0-106	Waveform
Sync	Free, Host	Synchronization
Polarity	Unipolar, Bipolar	Output range
Poly	Off, On	Per-voice mode — each voice resets the LFO phase on note-on
Release	0-10s	Exponential decay on note-off (only active with Poly On)

## Waveform Categories

1. **Classic** (0-3): Sine, Triangle, Saw, Pulse
2. **LFO Shapes** (4-19): 16 specialized modulation shapes

3. **Special LFO** (20-27): 8 unique and experimental shapes
4. **Wavetables A-G** (28-97): 70 wavetable shapes
5. **Random** (98-106): 9 random and chaos modes

## Random LFO Modes

The Random category provides 9 specialized modes for generative modulation:

Index	Mode	Description
98	<b>Note Random</b>	Generates a fixed random value on each note-on, held for the duration of the note
99	<b>Scale Note Random</b>	Same as Note Random but quantized to the active scale
100	<b>Stepped Random</b>	Classic Sample & Hold — generates a new random value at each LFO rate cycle
101	<b>Scale Stepped Random</b>	Stepped Random quantized to the active scale
102	<b>CV Random</b>	Smooth, continuously interpolated random drift (organic movement)
103	<b>CV Scale Random</b>	CV Random quantized to the active scale
104	<b>Chaos 1 (Lorenz)</b>	Lorenz attractor — produces complex, non-repeating organic curves
105	<b>Chaos 2 (Rössler)</b>	Rössler attractor — produces asymmetric, non-repeating organic curves
106	<b>Organic (Perlin)</b>	Perlin noise — smooth, naturally flowing random movement with no sudden jumps. Produces organic, breath-like modulation ideal for subtle parameter drift and evolving textures

**Scale-quantized modes** (99, 101, 103) use the scale currently set in the Chord Generator, ensuring that modulation values always land on harmonically coherent intervals.

**Chaos modes** (104-105) are deterministic systems that produce complex, never-repeating patterns. They are excellent for evolving textures and organic parameter movement that avoids the mechanical feel of periodic LFOs.

## Link Mode

LFO1 and LFO3 include a Link selector. LFO1 can chain to LFO2 (Pair modes) or drive LFO2+LFO3+LFO4 simultaneously (Quad modes). LFO3 only offers Pair modes over LFO4. LFO2 and LFO4 have no selector of their own — they receive Link state from their master.

**Pair modes** (LFO1→LFO2, LFO3→LFO4):

Mode	Description
No	Independent LFOs — each with its own speed and waveform
Dual	Slave runs at master's frequency with 90° phase offset. Slave Speed fader is disabled
Phase	Slave runs at master's frequency with a configurable phase offset. Slave Speed fader controls offset (0-360°). Label changes to "Phase"
Divide	Slave runs at a fraction of master's frequency. Slave Speed fader controls division ratio (÷1 to ÷16). Label changes to "Divide"
Sync	Slave resets its phase together with master. Slave Speed fader is disabled

**Quad modes** (LFO1 only — drives LFO2, LFO3 and LFO4 simultaneously):

Mode	Description
Q.Dual	All three slaves run at LFO1's frequency with 90°, 180° and 270° phase offsets respectively
Q.Phase	All three slaves run at LFO1's frequency. Each slave's Speed fader controls its own phase offset
Q.Div	Each slave runs at an independent division of LFO1 (per-slave Speed fader controls its ratio)
Q.Sync	All three slaves reset their phase together with LFO1

When LFO1 is in a Quad mode, LFO3's Link selector is disabled (LFO3 is already slaved to LFO1).

**Pendulum mode** (LFO1 only — transforms all 4 LFOs into a harmonic polyrhythm):

Inspired by the Harvard physics demonstration where 15 pendulums of increasing length produce a traveling-wave illusion before reconverging. In Pendulum, each LFO is locked to its own integer **Harmonic** of a shared **Cycle Time**, so the four LFOs reconverge to phase 0 simultaneously at every Cycle boundary — guaranteed by construction, with zero drift across long cycles.

When LFO1 selects Pendulum from the LINK dropdown, the UI reconfigures:

Control	Meaning in Pendulum
Rate fader (each LFO)	Becomes <b>Harm</b> — integer 1..16 displayed as "N cyc". Each LFO completes N cycles per shared Cycle Time. Defaults 3 / 5 / 7 / 11 (coprime primes, so the four only reconverge at the full Cycle boundary).
Sync button (each LFO)	Becomes <b>Cycle</b> — the shared Cycle Time selector, backed by one APVTS parameter so all four buttons always show the same value.
LFO3 Link selector	Dimmed — Pendulum owns all four LFOs.

**Cycle Time choices** (13 values, grouped by a separator in the dropdown):

- Free seconds (no DAW transport needed): 5s, 10s, 20s, 30s, 60s, 120s, 180s
- Host bars (locked to DAW tempo): 1, 2, 4, 8, 16, 32 bars

In Host mode the reconvergence lands exactly on the bar downbeat — every 4 bars at Cycle = "4 bars", every 16 at "16 bars", etc. When the transport is stopped the LFOs free-run at the equivalent tempo-derived rate so you hear movement while auditioning MIDI without playback; the phase snaps back to the grid the moment the transport starts.

**Phase math:**  $\text{phase}(t) = \text{fmod}(\text{Harm} \times t / \text{Cycle}, 1)$ , derived per-sample — never accumulated — so long Free cycles (180 s) land on phase 0 within a fraction of a sample.

**Right-click to reset:** right-click on the LINK button (when Pendulum is active) or on any of the four Cycle dropdowns realigns the system — all four LFO phases snap to 0 and the shared elapsed counter zeros. The next reconvergence is one full Cycle Time from the click. The matrix LFO Reset destinations do the same when driven from a CV source (Gate Generator, Env, another LFO's pulse).

**Scope source “LINK”:** a unified LFO-link visualizer that adapts its render to whichever link mode is currently active. A small tag in the top-right corner of the scope (DUAL / PHASE / DIVIDE / SYNC / QUAD or QUAD+submode / PEND) identifies the current family, so the picture stays self-describing as you sweep the LINK selector.

- **Off:** “OFFLINE” placeholder — LFOs run independently and nothing is linked.
- **Pair modes (Dual / Phase / Divide / Sync):** LFO1 and LFO2 render as a real-time scrolling waveform overlay. Each LFO drawn in its fader colour (purple / light blue) at 80% alpha so their crossings stay readable. The traces are fed each timer tick from the engine's *current effective output* (`engine->getPhase()` × the active waveform), so Phase shifts appear as a horizontal offset between the two lines, Divide as a visibly slower cycle on the slave, and Sync as reset glitches on the slave whenever the master crosses zero. Wavetables show their real evolving shape — the overlay captures whatever the audio path is doing, not a sine approximation.
- **Quad modes:** the viewport splits horizontally — top panel shows pair LFO1+LFO2, bottom panel shows LFO3+LFO4, separated by a thin divider. Both pairs follow the same Dual-family submode but are independent of each other, so you see two simultaneous pair interactions at a glance.
- **Pendulum:** the 16-ball Harvard-style rig — 16 virtual bolas arranged horizontally, each tracing its harmonic's current phase as a vertical sine projection. The four bolas at the active `Lfo{N}_harm` positions are painted in their LFO's colour; the other eleven are dim ghosts that reveal the underlying traveling-wave pattern. A thin stroke connecting all sixteen shows the instantaneous wave-shape across the rig. When two or more LFOs share a harmonic the ball renders as a segmented ring so overlap reads at a glance.

The scrolling history (~4 seconds at 60 Hz) is used only in Pair / Quad modes — Pendulum keeps its time-origin ball physics unchanged.

**Waveform notes (Pendulum only):** inside Pendulum the scope renders sine-projected phase (it visualises the harmonic phase clock, not the per-LFO wave). The audio path still uses whatever wave each LFO is set to — Saw, Pulse, Wavetable, Special, etc. all work and produce their own emergent polyrhythm with the same reconvergence point. Non-periodic waves (Chaos 104-105, Organic 106) lose the reconvergence guarantee since they don't cycle — using them in Pendulum is valid but the result is “chaos modulated by a shared cycle”, not Pendulum Wave proper. **Dual / Quad modes do NOT share this limitation** — they render the LFO's actual effective output, so wavetables, chaos, and any other waveform show their true shape in real time.

**Example routings** — Pendulum is most audible when all four LFOs drive **different destinations**. One LFO modulating a single parameter sounds like a sine wave because a single pendulum IS a sine wave; the pendulum-wave effect only emerges when you hear four rates interacting. Try:

- LFO1→Osc1 tune, LFO2→Osc2 tune, LFO3→Osc3 tune, LFO4→Osc4 tune — unison pitch drift that reconverges on the downbeat
- LFO1→Filter1 cutoff, LFO2→Filter2 cutoff, LFO3→Ch1 pan, LFO4→Ch2 pan — evolving stereo timbre with bar-locked reconvergence
- LFO1/2/3/4→CH1 CTRL at different depths — overlapping tremolos that breathe together every cycle

## Poly Mode and Release

Each LFO has a **Poly** button that switches it from global (a single shared phase across all voices) to per-voice (each voice keeps its own phase). With Poly On, the LFO phase resets on every note-on, enabling tremolos and attack-coupled movements that start in sync with the note.

The **Release** fader (0-10s, exponential decay with  $\tau = \text{release}/5$ ) controls how the LFO modulation fades out on note-off. Only active when Poly is On.

- **Release = 0**: modulation active as long as the voice exists, abrupt cutoff on note-off
- **Release = 2s**: ~2 seconds to full silence after release
- **Release = 10s**: very long tails for evolving pads

With Poly Off the Release fader is dimmed and has no effect.

## LFO Reset (matrix destination)

Each LFO exposes a **Reset** destination in the matrix. A rising-edge CV signal (e.g. Gate Generator, another LFO in Pulse shape, or a fast envelope) instantly resets the LFO phase. Useful for syncing the LFO to external rhythmic patterns without relying on Host Sync.

### Examples

- Gate Gen → LF01 Reset — the LFO retriggers with every Gate Generator pulse
- LF02 (Pulse) → LF01 Reset — LFO1 syncs to LFO2 divisions
- Env2 → LF03 Reset — each note-on re-aligns LFO3 with the envelope

# Logic (Function Generator)

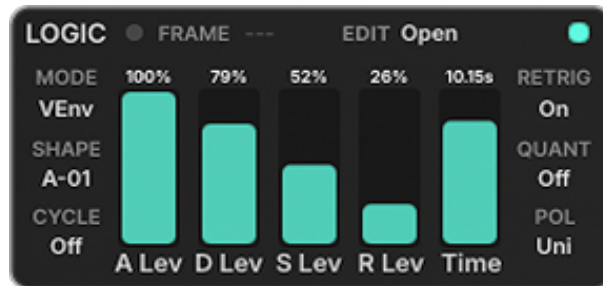


Figure 18: Logic module — multi-mode function generator (VENV / FUNC / RAND / S&H) with shape, length, attack/release and quantization controls.

A versatile module that can function as a vector envelope, function generator, random, or sample & hold.

## Parameters by Mode

### VENV Mode

Parameter	Description
Fader 1	Attack Gain - gain of red zone (0 → Attack BP)
Fader 2	Decay Gain - gain of cyan zone (Attack BP → Decay BP)
Fader 3	Sustain Gain - gain of yellow zone (Decay BP → Sustain BP)
Fader 4	Release Gain - gain of purple zone (Sustain BP → end)
Fader 5	TIME - total envelope duration (100ms to 30s)

### Function Mode

Parameter	Description
Fader 1	Rise - rise time
Fader 2	Fall - fall time
Fader 3	Curve - curvature
Fader 4	Smooth - smoothing
Fader 5	Level - output level

## Random/S&H Mode

Parameter	Description
Fader 1	Rate - change speed
Fader 2	Spread - variation range
Fader 3	Curve - distribution
Fader 4	Smooth - smoothing
Fader 5	Level - output level

## Audio In (S&H External Input)

In **Sample & Hold mode**, the Logic module can receive an external signal from the modulation matrix instead of its internal random/waveform source.

### Setup

1. Open the modulation matrix
2. Select a source row (e.g., LFO1, Env1, Gate Gen)
3. Set the column destination to **“Logic > In”** (found at the end of the Logic submenu)
4. The S&H module now samples from the incoming CV signal

### Behavior

- **Cycle ON:** The external signal is sampled at each cycle trigger (classic S&H behavior). The Spread fader still scales around center, and Smooth applies slew limiting.
- **Cycle OFF:** The external signal is followed continuously (envelope follower behavior), with smoothing applied by the Smooth fader.
- **WAVE button:** When an external input is connected, the WAVE selector button is dimmed since the internal waveform source is bypassed.
- **Disconnecting:** Remove the matrix connection and the WAVE button re-enables; Logic returns to its normal internal source.

### Practical examples

- Route an LFO to Logic In → rhythmic modulation synchronized to LFO shape
- Route Gate Gen to Logic In → binary gate patterns converted to S&H steps
- Route Env1 to Logic In → envelope-driven modulation with quantization (via Spread)

**Note:** Audio In only works in S&H mode. In VENV and Function modes, the Logic In matrix destination has no effect.

## Operation Modes

Mode	Description
<b>VENV</b>	Vector Envelope - customizable envelope with visual editor. <b>In POLY mode, VENV is per-voice</b> (each voice has its own independent instance)
<b>Function</b>	Function generator with rise/fall (global)
<b>Random</b>	Random values with rate and spread (global)
<b>Sample &amp; Hold</b>	Samples input on each trigger (global)

## VENV - Vector Envelope

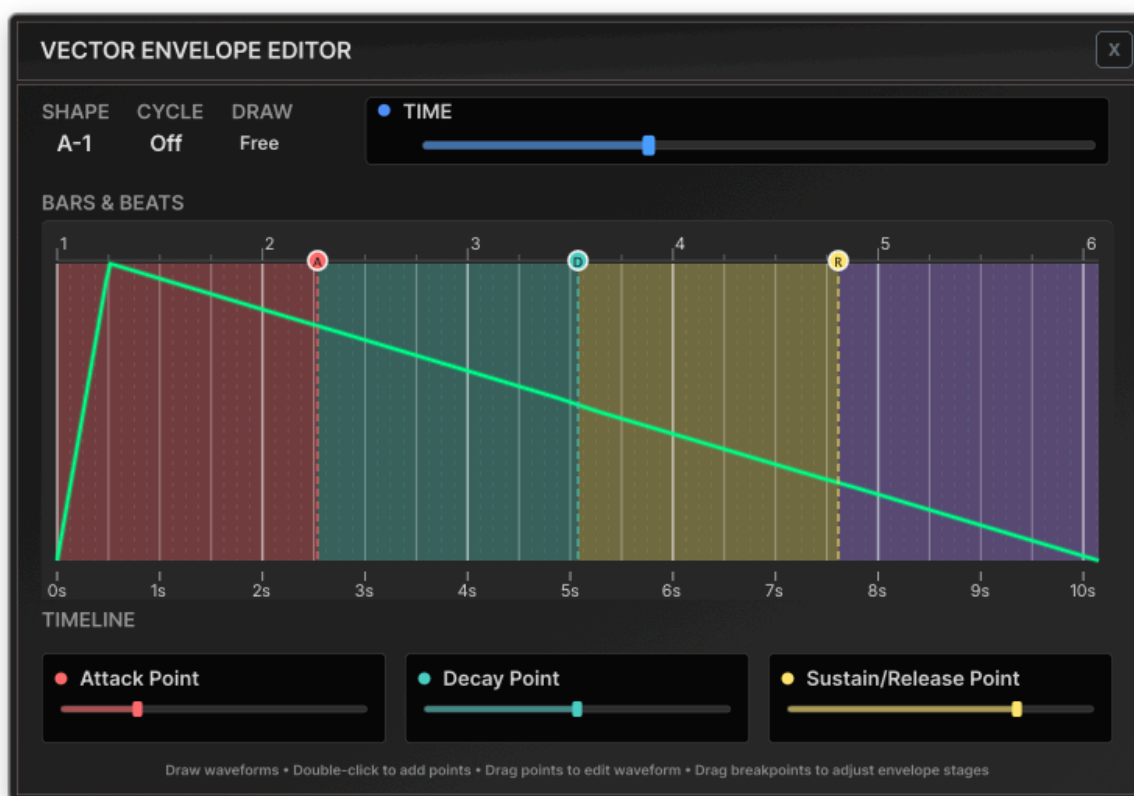


Figure 19: VENV editor — visual vector wave editor with section gain controls, loop region and preset selector.

VENV mode is a fully customizable envelope. It includes a visual wave editor that allows you to design unique envelope shapes.

### Features

- **182 predefined shapes** organized in banks
- **Visual wave editor** with free draw and line modes
- **4 colored sections:** Attack (red), Decay (cyan), Sustain (yellow), Release (purple)
- **Tempo synchronization** to DAW or internal clock
- **Per-voice in POLY mode:** each voice has its own independent instance

### Per-Section Gain System

Each envelope section has its own gain (Faders 1-4) that multiplies the waveform in that zone:

Section	Color	Fader	Effect
Attack	Red	Fader 1	Controls the initial rise amplitude
Decay	Cyan	Fader 2	Controls the fall amplitude
Sustain	Yellow	Fader 3	Controls the sustain amplitude
Release	Purple	Fader 4	Controls the release amplitude

## Automatic interpolation

To avoid clicks when gains differ between sections, the system smoothly interpolates (~5ms) at boundaries: - If Attack=100% and Decay=50%, the transition will be smooth - If you release the note during Attack or Decay, it will jump to Release without clicking - Release always ends at 0 regardless of its gain value

## Practical examples

- For a soft “fade in”: Attack=0%, then gradually rises to Decay=100%
- To attenuate the sustain: Sustain=50% while keeping others at 100%
- For a softer release: Release=75%

## Vector Wave Editor

Click the **EDIT** button next to the shape selector to open the Vector Wave Editor.

### Editor Window

- **Display area:** Shows the waveform with 4 colored sections
- **Top ruler:** Shows bars/beats synchronized to tempo
- **Bottom timeline:** Shows time in seconds/milliseconds
- **3 draggable breakpoints:** Define boundaries between Attack/Decay/Sustain/Release

### Draw Modes

Mode	Description
<b>FREE</b>	Freehand drawing - drag the mouse to paint the shape directly
<b>LINES</b>	Control point mode - add and connect points with Bezier curves

### FREE Mode

- Drag the mouse over the drawing area to paint the shape
- Ideal for organic and irregular shapes
- Stroke follows mouse movement in real-time

### LINES Mode

- **Double-click** to add control points
- **Drag** points to move them
- **Shift + drag** to adjust curve tension/curvature
- Points automatically connect with smooth interpolation
- Ideal for precise and geometric shapes

### Breakpoints

The 3 vertical breakpoints divide the envelope into 4 sections: 1. **Attack** (0 → BP1): Attack phase - time to reach maximum 2. **Decay** (BP1 → BP2): Decay phase - transition to sustain level 3. **Sustain** (BP2 → BP3): Sustain phase - while note is held 4. **Release** (BP3 → end): Release phase - after note release

Drag breakpoints horizontally to adjust each section’s duration.

## Editor Controls

Control	Function
TIME	Total envelope duration (side fader)
Grid	Enable/disable tempo-synced grid
Division	Grid resolution (1/4, 1/8, 1/16, etc.)

**Tip:** Use LINES mode to create envelopes with precise transitions (like steps or ramps), and FREE mode for more organic and chaotic textures.

## VENV Loop

When **Cycle** is enabled in VENV mode, the envelope performs a ping-pong loop during the sustain phase:

- The loop plays between the **Decay breakpoint** and the **Sustain breakpoint**
- It first traverses the shape forward, then backward, and repeats
- The loop continues while the note is held
- When the note is released, the envelope jumps directly to the Release phase

This allows creating rhythmic modulations and evolving textures synchronized with the total envelope duration.

## VENV User Presets

You can save custom envelope shapes as user presets and reload them later:

### Saving

1. Design your envelope in the VENV editor (using predefined shapes, FREE draw, or LINES mode)
2. Click the **envelope selector** (shape menu) at the top of the editor
3. Select **Save As...** at the bottom of the menu
4. Enter a name in the dialog and click **Save**

### Loading

1. Click the envelope selector in the VENV editor
2. Select the **USER** category at the bottom of the bank list
3. Click a saved preset name to load it

### Data saved in a VENV preset

- Complete waveform shape
- Control points (for LINES mode envelopes)
- Draw mode (FREE/LINES)
- Breakpoint positions (Attack/Decay/Sustain)
- Fader values (per-section gains)
- Cycle, Retrig, Quant, Bipolar settings

**File location:** ~/Documents/Portador Presets/Venv/User/

---

## Additional Logic Options

---

<b>Option</b>	<b>Description</b>
<b>Shape</b>	182 predefined shapes organized in banks
<b>Cycle</b>	ON = ping-pong loop during sustain (VENV) or auto-repeat (other modes), OFF = one-shot
<b>Retrig</b>	Off = no retrigger, Note = retrigger on note, Always = continuous retrigger
<b>Quant</b>	Quantization: Off, Scale (to current scale), 2-12 steps
<b>Bipolar</b>	Unipolar (0 to +1) or Bipolar (-1 to +1)

---

## Effects

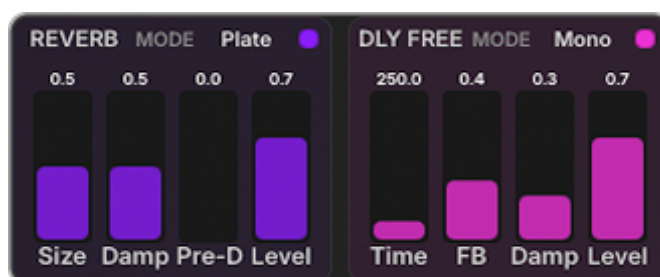


Figure 20: FX Slots — two effect slots with mode selector and parameter faders. Each channel sends independently to FX 1 and FX 2.

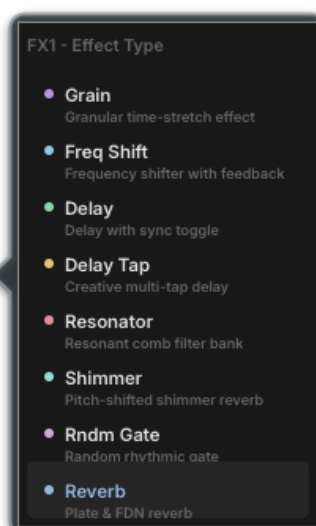


Figure 21: FX type selection menu — 8 effects: Granular, Frequency Shifter, Delay (Free / Host), Resonator Bank, Pitch Shimmer, Random Gate and Reverb.

Portador includes 8 audio effects that can be loaded into the two FX Slots. Each channel can send audio to FX1 and FX2 independently.

### Granular Processor

Breaks audio into small “grains” to create unique textures. Supports up to 128 simultaneous grains with near-instant startup (~50-500ms response time). When all grain slots are full, the grain closest to finishing is replaced automatically (grain stealing).

Parameter	Range	Description
Size	1-500ms	Size of each grain
Density	1-50	Number of simultaneous grains
Pitch	±24 semitones	Grain transposition
Level	0-100%	Output volume

## Modes

Mode	Description
Follow	Grains follow incoming audio
Freeze	Freezes a moment of audio and repeats it
Reverse	Plays grains backwards
Scatter	Random dispersion of grain position

## Frequency Shifter

Shifts all frequencies by a fixed amount (doesn't transpose, shifts).

Parameter	Range	Description
Shift	-1000 to +1000 Hz	Shift amount
Shape	0-100%	Waveform position
Feedback	-100% to +100%	Feedback
Level	0-100%	Volume
Wave	0-97	Modulator waveform

**Tip:** Small shift values (1-10 Hz) create phasing and chorus effects. Larger values create metallic and bell-like effects.

## Delay

Stereo delay effect with tempo synchronization and tone control in the feedback path. Free and Host sync modes are unified into a single effect — toggle between them via right-click on the FX name.

Parameter	Range	Description
Time	1ms - 2000ms	Delay time (Free mode)
Div	1/64T - 4 Bar	Musical division (Host Sync mode)
FB	0-99%	Feedback amount
Damp	0-100%	Filtering in the feedback loop (attenuates highs)
Level	0-100%	Delay output volume

## Stereo Modes (3)

Mode	Description
<b>Mono</b>	Single delay line, both channels delayed identically
<b>Ping Pong</b>	Sound bounces between left and right channels
<b>Wide</b>	Independent delay processing per channel for stereo widening

## Synchronization (right-click menu)

- **Free** (display: DLY FREE): Manual delay time control (1-2000ms)

- **Host** (display: DLY HOST): Synced to DAW tempo with musical divisions (25 options from 1/64T to 4 Bar)

Toggle sync mode by right-clicking the effect name — the menu shows a Sync section with Free/Host options. The display name and first fader update automatically.

**Tip:** Use Ping Pong mode with high Feedback and moderate Damping to create rhythmic delays that progressively fade. Wide mode is ideal for adding stereo depth to leads and pads.

## Delay Tap

A creative multi-tap delay with 8 internal taps. Unlike the standard Delay, Delay Tap generates complex rhythmic and textural patterns from a single input. The **Scatter** fader is a macro control that simultaneously affects tap timing, stereo width, per-tap darkening, tape-like wobble modulation, and pitch shift intensity.

Parameter	Range	Description
Time	1ms - 2000ms	Base delay time (Free mode)
Div	1/64T - 4 Bar	Base division (Host Sync mode)
Scat	0-100%	Complexity macro — controls tap spread, stereo, darkening, wobble, and pitch depth
FB	0-99%	Global feedback — with pitch modes, creates cascading shimmer effects
Level	0-100%	Wet output volume

### Scatter Macro — what it controls

Scatter = 0%	Scatter = 100%
Nearly single-tap, clean delay	8 taps fully spread, maximum character
Centered stereo	Wide ping-pong stereo field
All taps bright	Later taps progressively darker (LP filter)
No wobble	Tape-like time modulation per tap (0.6-1.65 Hz)
No pitch shift (Rise/Fall)	Full pitch shift (up to $\pm 7$ semitones)

### Modes (3)

Mode	Description
<b>Cluster</b>	Taps bunched symmetrically around the base time. No pitch shift. Dense, chorus-like slapback character.
<b>Rise</b>	Taps spread forward with exponential spacing. Each tap pitched UP progressively (up to +7 semitones at full scatter). With feedback, creates ascending shimmer cascades.
<b>Fall</b>	Taps arranged from long to short delay (reverse feel). Each tap pitched DOWN progressively (up to -7 semitones). Creates darkening, descending textures with a build-up character.

## Synchronization (right-click menu)

Same as Delay — right-click the effect name to toggle between Free (DLY TAP) and Host (TAP HOST) sync modes.

**Pitch Shift:** In Rise and Fall modes, each tap reads the delay buffer at a slightly different speed, creating real pitch transposition. The amount scales with Scatter — at Scatter 0% there is no pitch shift. With feedback, the pitch effect compounds across repetitions: Rise creates ever-ascending shimmer tails, Fall creates descending spiral textures.

**Darkening:** Later taps pass through a progressive low-pass filter (controlled by Scatter). This only affects the audio output, not the feedback path — so repeats maintain their energy while the output gets progressively warmer.

**Wobble:** Each tap has its own slow LFO (0.6-1.65 Hz, staggered phases) that subtly modulates its delay time, adding analog tape-like character. The depth increases with Scatter.

**Signal Stability:** High feedback values sustain properly without signal collapse. An internal DC blocker prevents low-frequency buildup, and a smooth saturation curve keeps the output musical even at extreme feedback settings.

**Tip:** Try Rise mode with Scatter ~50% and FB ~60% for lush shimmer delays. Fall mode with high Scatter creates reverse-like ambient textures. Cluster at low Scatter with high FB produces thick, modulated slapback effects. Modulate Scatter via the Matrix (LFO or Envelope) for evolving delay textures.

## Resonator Bank

A bank of 8 tuned comb filters that resonate at harmonics of the current MIDI note. The input audio excites the resonances, creating pitched metallic, string, or bell-like timbres.

Parameter	Range	Description
Decay	0-100%	Resonance duration (maps to feedback 0.8-0.999)
Damping	0-100%	High-frequency absorption in feedback (0=bright, 100%=dark)
Stretch	-24 to +24 st	Progressive inharmonic detuning: higher harmonics get proportionally more detune (0 = pure harmonic stack, ±24 = bell/metallic inharmonic spread)
Level	0-100%	Output volume

## Modes (4)

Mode	Description
<b>Harmonic</b>	Harmonics 1, 2, 3, 4, 5, 6, 7, 8 — natural harmonic series
<b>Odd</b>	Harmonics 1, 3, 5, 7, 9, 11, 13, 15 — hollow, clarinet-like
<b>Struck</b>	Inharmonic ratios (1, 2.76, 5.4, 8.93...) — metallic, bell-like
<b>String</b>	Karplus-Strong style — noise burst excites the comb filters for plucked string sounds

**Tuning:** The resonator automatically tunes to the last played MIDI note. All 8 comb filters are tuned to harmonics of that frequency based on the selected mode.

**Tip:** Use Harmonic mode with low Damping for bright, shimmering resonances. Struck mode creates metallic bell tones. String mode with high Decay produces self-sustaining plucked sounds even without audio input.

## Pitch Shimmer

A pitch-shifting delay effect with feedback. Each repetition is transposed up or down, creating crystalline ascending or descending tails.

Parameter	Range	Description
Shift	-12 to +12 semitones	Pitch shift per repetition
Feedback	0-95%	Amount of shifted signal fed back into the delay
Size	10-500ms	Delay/grain size
Level	0-100%	Output volume

## Modes (3)

Mode	Description
<b>Up</b>	Pitch shifts upward only
<b>Down</b>	Pitch shifts downward only
<b>Dual</b>	Stereo split — left channel shifts up, right shifts down

**Tip:** Set Shift to +12 (octave up) with moderate Feedback for classic shimmer reverb tails. Use Dual mode for wide, ethereal stereo effects. Negative shifts create dark, descending textures.

**Note on Shift = 0:** Pitch Shimmer is a pitch-shifter at heart, not a transparent delay. With **Shift = 0** and longer Size values ( $\geq 200$ ms), the internal feedback path interacts with the soft-saturation stage to emphasize the second harmonic of the input — meaning Shift = 0 is NOT a clean passthrough but a colored, second-harmonic-rich texture. This is intentional. For a transparent send, bypass Pitch Shimmer or set Feedback to 0 with very small Size. For clean delay, use the dedicated Delay effect instead.

## Random Gate

A probabilistic rhythmic gate that cuts the audio with controllable randomness, creating dynamic rhythmic patterns.

Parameter	Range	Description
Density	0-100%	Probability that each step will be open (higher = more audio passes through)
Rate	0-100%	Gate speed (0.5Hz - 50Hz)
Smooth	0-100%	Attack/release of the gate envelope (0=hard chop, 100%=smooth tremolo)
Level	0-100%	Output volume

## Step Modes (4)

Mode	Description
4	4 steps per cycle
8	8 steps per cycle
16	16 steps per cycle
32	32 steps per cycle

**Tip:** Low Density with 16 steps creates sparse, glitchy patterns. High Density with Smooth at 100% produces a subtle tremolo effect. Combine with delay for rhythmic tape-stop effects.

## Reverb

Stereo reverb effect available as an FX Slot type, with three selectable algorithms.

Parameter	Range	Description
Size/Time	0-100%	Reverb size (0.1s to 10s)
Damping	0-100%	High-frequency absorption in the reverb tail
Pre-delay	0-100%	Pre-delay before reverb onset (0-200ms)
Level	0-100%	Wet/dry mix

## Algorithms (3)

Mode	Description
<b>Plate</b>	Plate reverb (Dattorro algorithm) — 4 input diffusion allpasses, 2 parallel decay tanks, stereo taps. Classic, dense, and bright sound
<b>FDN</b>	Feedback Delay Network — 8 delay lines, Hadamard 8×8 mixing matrix, per-line lowpass damping. Spacious and natural sound
<b>Recursive</b>	Recursive reverb — nested allpass and comb filter network with modulated delay times. Produces lush, organic tails with natural density build-up

**Modulation:** All 4 reverb parameters (Size, Damping, Pre-delay, Level) are available as modulation destinations in the matrix when Reverb is loaded in an FX slot.

**Tip:** Use Plate mode with low Size for short, bright room reverbs. FDN mode with high Size creates wide, enveloping spaces. Recursive mode excels at lush, evolving tails with natural density. Pre-delay adds definition to the dry signal before the reverb tail.

# Output Channels



Figure 22: Channels 1 and 2 — output mixing strip with CTRL, Level, Pan, FX 1/2 sends, plus per-channel EQ and Distortion access. Voice Spread panel on the right.

Two mixing channels with integrated compression, parametric EQ, and distortion.

Parameter	Description
Ctrl	VCA modulation control
Level	Channel volume
Pan	Stereo position
FX1 Send	Send level to FX Slot 1 (Granular, FreqShifter, Delay, Delay Tap, Resonator, Shimmer, Random Gate, or Reverb)
FX2 Send	Send level to FX Slot 2 (Granular, FreqShifter, Delay, Delay Tap, Resonator, Shimmer, Random Gate, or Reverb)

## Integrated Compressors

- **Channel 1:** Optical-type compressor (smooth, musical)
- **Channel 2:** VCA-type compressor (fast, precise)

## Parametric EQ (per channel)

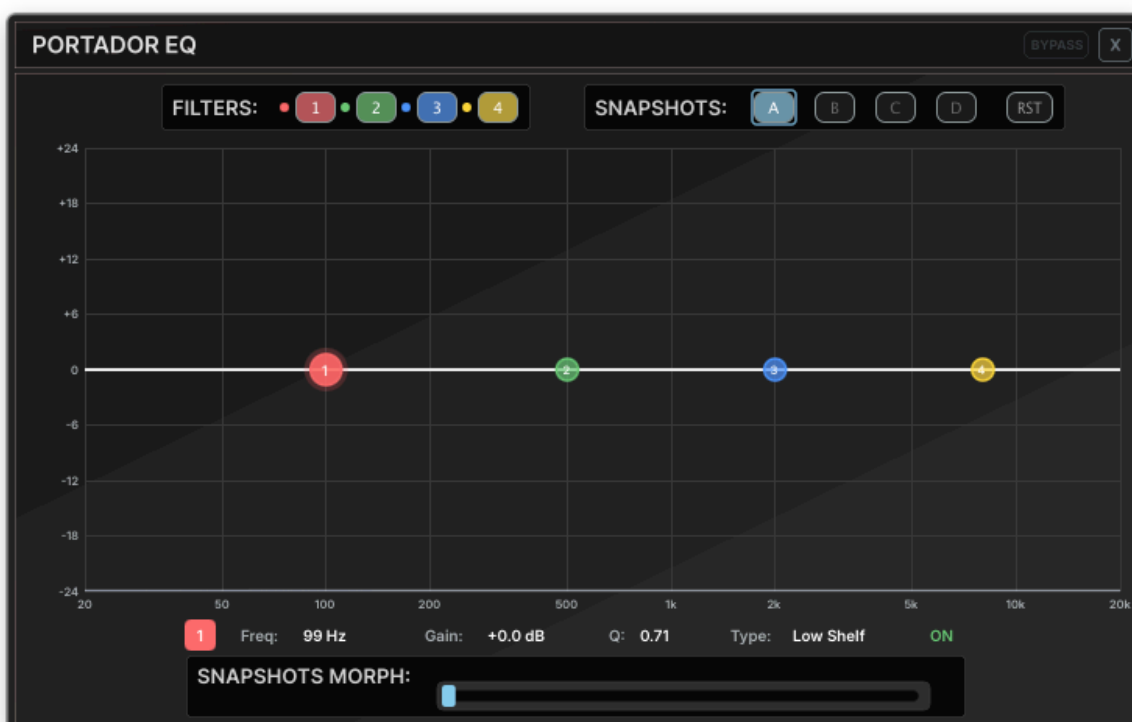


Figure 23: Parametric EQ — 4 colored draggable bands with frequency response curve, A/B/C/D snapshots and morph control.

Each channel includes a 4-band parametric EQ accessible via the **EQ** button above the channel faders. Clicking opens a popup window with a Pro-Q3-style visual editor.

### Features

- 4 colored draggable nodes on the frequency response curve
- Real-time frequency response visualization with POST-EQ spectrum analyzer
- Biquad cascade DSP (4 biquads in series)
- ON/OFF button to enable/disable the EQ
- Bottom bar values (Freq, Gain, Q) support both mouse wheel scroll and click+drag interaction

### Parameters per band

Parameter	Range	Description
Freq	20Hz - 20kHz	Center frequency (logarithmic scale)
Gain	±24 dB	Band gain
Q	0.1 - 10.0	Q factor / Bandwidth
Type	Bell, LS, HS, LP, HP	Band type (bell, low shelf, high shelf, low pass, high pass)

## Real-Time Spectrum Analyzer

The EQ editor displays a real-time POST-EQ spectrum analyzer behind the frequency response curves. This shows how the audio signal looks after the EQ is applied, allowing you to see the effect of your EQ adjustments in real time.

## Bottom Bar Interaction

The bottom bar displays the selected band's Freq, Gain, and Q values. You can adjust these values by:

- **Mouse wheel:** Scroll up/down over any value to increase/decrease it
- **Click + drag:** Click on a value and drag up/down to adjust it

**Modulation:** All EQ band parameters (Freq, Gain, Q) are available as modulation destinations in the matrix when the channel's EQ is enabled. Frequency modulation uses exponential scaling ( $\pm 4$  octaves at full depth), gain modulation is  $\pm 24$ dB, and Q modulation is  $\pm 5.0$ .

**Tip:** Drag nodes horizontally to change frequency and vertically to change gain. The EQ activates automatically when opening the window.

## EQ Morph (Snapshots)

The EQ editor includes a **Snapshot Morph** system that lets you store up to 4 EQ states (A, B, C, D) and smoothly interpolate between them with a single control.

## Interface

The EQ window (730x460) is organized into zones:

- **Header:** "PORTADOR EQ" title + BYPASS and close buttons
- **Top containers:** FILTERS (buttons 1-4 for activating/selecting bands) and SNAPSHOTS (buttons A-D for managing snapshots)
- **EQ Graph:** Frequency response curves + draggable nodes + spectrum analyzer
- **Info bar:** Freq, Gain, Q, Type and ON for the selected band
- **Morph bar:** Horizontal slider with wavetable-style pointer for controlling the morph position

## How Snapshots Work

- **Snapshot A** is active by default with the current EQ state
- **Activating a snapshot** (e.g. click B): Copies the current editing snapshot's state and selects it for editing. The morph slider automatically jumps to that snapshot's position so you immediately hear what you're editing
- **Editing:** Simply drag the nodes — changes are saved automatically to the selected snapshot and you hear the result in real time
- **Switching snapshots:** Click on another active snapshot to edit it. The morph slider jumps to that position
- **RST button:** Resets the entire EQ to flat defaults (all bands at 0dB) and clears all snapshots except A

## Morph Slider

Parameter	Range	Description
Morph	0.0 - 1.0	Interpolation position between active snapshots

- With 2 active snapshots (A and B): Morph 0.0 = A, Morph 1.0 = B
- With 3 active snapshots (A, B, D): Morph 0.0 = A, 0.5 = B, 1.0 = D
- With 4 active snapshots: evenly distributed across the 0-1 range

## Interpolation

- **Frequency:** Log-space interpolation (perceptually linear)
- **Gain:** Linear interpolation in dB
- **Q:** Log-space interpolation
- **Type:** Snaps to the nearest snapshot's type at the 50% crossover point
- **Disabled bands:** Stored with Gain=0dB (transparent) and smoothly fade in/out during morph

## Visual Feedback

- **White curve** = the snapshot you're currently editing (matches the draggable nodes)
- **Colored curves** = other active snapshots shown as reference (A=ice blue, B=lilac, C=orange, D=mint)
- **During morph interaction:** When you move the morph slider (manually or via macro/modulation), the white curve switches to show the real interpolated output — you see the EQ move between snapshots in real time. All snapshots appear in their colors as reference. When you stop interacting with the slider or click a snapshot, the white curve returns to the editing snapshot

**Modulation:** The Morph parameter is available as a modulation destination in the matrix (under Channel 1 / Channel 2 → "EQ Morph"). This allows animating the EQ with LFOs, envelopes, macros, VENV, or the gate generator. Right-click the Morph slider to assign a macro. The macro range indicator updates in real time when adjusting via Macro Settings.

**Persistence:** Snapshots are saved both in the DAW project state (close/reopen session) and in plugin presets.

**Tip:** Assign an LFO to EQ Morph to create organic EQ movements that "breathe" between different states. Set snapshot A as a neutral EQ and snapshot B with a high-frequency boost — the LFO will make the brightness rise and fall rhythmically.

## Distortion (per channel)

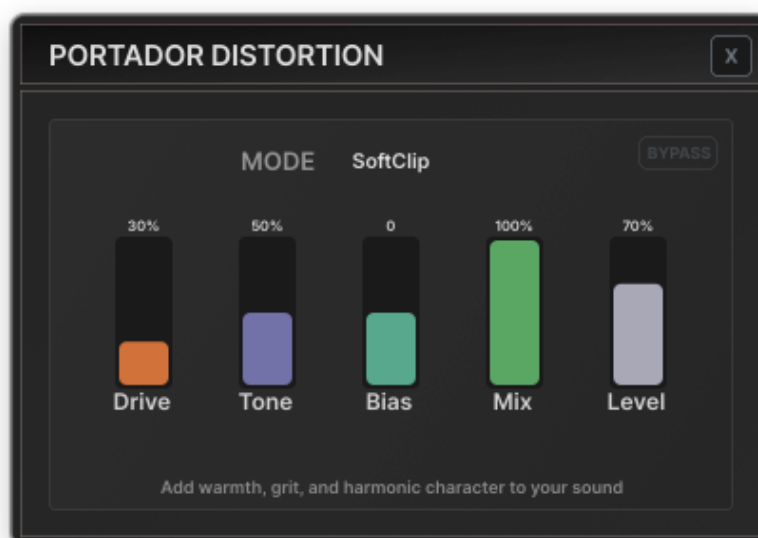


Figure 24: Distortion editor — 5 algorithms (Soft Clip, Hard Clip, Foldback, Wave-shaper, Bit Crush) with drive, tone, bias, mix and output.

Each channel includes a distortion module accessible via the **DST** button above the channel faders. Clicking opens a popup window with 5 distortion algorithms.

## Distortion Modes (5)

Mode	Description
<b>Soft Clip</b>	Soft tanh saturation — warm and musical
<b>Hard Clip</b>	Soft-knee clipping — aggressive with odd harmonics. Quadratic soft knee in the clipping region eliminates harsh digital artifacts while preserving the hard-clip character
<b>Wavefolder</b>	Wave folding — complex metallic harmonics. Includes a dedicated 2-pole anti-aliasing filter (5 kHz SVF) that tames the harshest folding products
<b>Bitcrush</b>	Sample rate and bit depth reduction — lo-fi and digital. Pre-decimation low-pass filter automatically tracks the target sample rate to reduce aliasing
<b>Tube</b>	Asymmetric tube-style saturation — warm with even harmonics. Smooth crossfade between plate saturation (positive) and grid cutoff (negative) eliminates zero-crossing artifacts

## Parameters

Parameter	Range	Description
Drive	0-100%	Distortion intensity
Tone	0-100%	Post-distortion tilt EQ (dark ↔ bright)
Bias	-100% to +100%	Post-distortion DC bias offset (adds even harmonics). Applied after the distortion curve to avoid deforming the saturation shape
Mix	0-100%	Dry/wet mix
Output	0-200%	Output level (makeup gain). Allows boosting above unity to compensate for level loss at high drive settings

- Distortion includes internal 2x oversampling to reduce aliasing, plus mode-specific anti-aliasing filters for Wavefolder and Bitcrush.
- All parameters are smoothed (20ms ramp) for click-free automation and modulation.
- ON/OFF button to enable/disable the distortion module.
- **Signal chain:** Distortion is applied before the EQ in the channel strip.
- **Modulation:** All 5 distortion parameters (Drive, Tone, Bias, Mix, Output) are available as modulation destinations in the matrix when the channel's distortion is enabled.

**Tip:** Use Soft Clip with moderate Drive to add analog warmth to pads. Bitcrush is ideal for lo-fi effects and retro textures. Increase Output above 100% to add makeup gain when using heavy distortion.

# Chord Engine and Arpeggiator

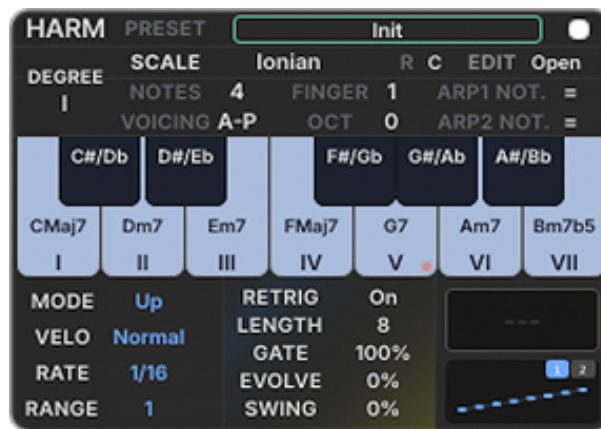


Figure 25: Chord Engine module (HARM) on the main UI — degree, scale and chord selectors plus arp parameters.

## Chord Types (21)

Chord	Notes
Maj7	1-3-5-7
m7	1-b3-5-b7
7	1-3-5-b7
m7b5	1-b3-b5-b7
dim7	1-b3-b5-bb7
mMaj7	1-b3-5-7
augMaj7	1-3-#5-7
aug7	1-3-#5-b7
6	1-3-5-6
m6	1-b3-5-6
sus4	1-4-5-7
sus2	1-2-5-7
Quartal	1-4-b7-b3' (P4-P4-P4 stack)
Quartal+	1-4-b7-3' (P4-P4-M3 "So What")
UST LydDom	Lydian Dominant upper structure
UST Alt	Altered scale upper structure
UST Lyd	Lydian upper structure
UST LocMaj	Locrian Major upper structure
UST Lyd#9	Lydian #9 upper structure
UST Lyd13	Lydian 13 upper structure
Maj7sus4	1-4-5-7 (Maj7 with suspended 4th — {0,5,7,11})

**Quartal Chords:** Unlike traditional tertian harmony (built on 3rds), quartal chords are built by stacking perfect 4ths. **Quartal** stacks three perfect 4ths (P4-P4-P4), creating an open, ambiguous sound used extensively in modern jazz and film scoring. **Quartal+** replaces the top perfect 4th with a major 3rd (P4-P4-M3), producing the famous "So What" voicing from Miles Davis's *Kind of Blue*. These chord types work particularly well with modal scales (Dorian, Mixolydian) and create a sound that avoids the strong major/minor identity of traditional chords.

**Upper Structure Triads (UST):** Upper structure triads are an advanced jazz voicing technique where a triad is played over an unrelated bass note or chord, creating complex extended harmony. Each UST type corresponds to a specific modal scale and produces rich tensions. These chord types are especially effective on dominant chords and work well with the Tritone Substitution feature (see below).

## Arpeggio Modes (22)

Mode	Description
Up	Ascending
Down	Descending
UpAndDown	Pendulum: ascending then descending. With 2 notes [A,C]: A-C-A-C-A-C (extremes repeat for rhythmic pulse). With 3+ notes [A,C,E]: A-C-E-C-A-C-E-C... (standard pendulum — no double notes at extremes)
DownAndUp	Pendulum (inverted): descending then ascending. With 2 notes [A,C]: C-A-C-A-C-A. With 3+ notes [A,C,E]: E-C-A-C-E-C-A-C... (standard pendulum — no double notes at extremes)
Random	Random order
InsideOut	From center outward
OutsideIn	From outside to center
InOutOutIn	Alternating I&O pattern
OutInInOut	Alternating O&I pattern
TonalRandom	Random within scale
AtonalRandom	Chromatic random
Chord	All notes at once
RepeatX2	Repeat each note 2 times
RepeatX4	Repeat each note 4 times
StairsUp	2 up, 1 down
StairsDown	2 down, 1 up
Converge	Inward zigzag: lowest, highest, 2nd lowest, 2nd highest...
Diverge	Outward zigzag: middle, adjacent out, further out...
Euclidean	Euclidean rhythm distribution
Additive	Philip Glass-style grow/shrink pattern
UpAndDown+	Like UpAndDown but repeats both extremes (Omnisphere-style) — emphasized bounce at each end of the pendulum
DownAndUp+	Like DownAndUp but repeats both extremes — mirror of UpAndDown+

### Euclidean Mode

The **Euclidean** mode distributes a number of pulses evenly across the arpeggio steps using the Bresenham algorithm. This creates classic Euclidean rhythms used in many world music traditions.

- **Pulses** (1-16): Set via the RANGE control, which changes to show “PULSES” in Euclidean mode
- **Steps**: Determined by the LENGTH parameter
- The algorithm distributes pulses as evenly as possible: e.g., 3 pulses in 8 steps = X . . X . . X .

### Examples

- 3/8: West African bell pattern
- 5/8: Common in Middle Eastern music
- 7/16: Complex polyrhythmic pattern

## Converge and Diverge Modes

**Converge** and **Diverge** create zigzag melodic contours by alternating between the extremes and center of the chord:

- **Converge** (inward): Plays the lowest note, then the highest, then the 2nd lowest, then the 2nd highest, and so on – converging toward the center. Creates a dramatic narrowing motion
- **Diverge** (outward): Starts from the center note(s) and expands outward – plays the middle note, then the notes adjacent to it, then further out. Creates an expanding, opening motion

These modes work especially well with wide chord voicings (5-note chords, open finger modes) where the pitch range between the lowest and highest notes is large.

## Additive Mode

The **Additive** mode creates a Philip Glass-inspired pattern where the arpeggio gradually grows from a single note to the full chord, then contracts back. The pattern cycles:

- **Growing:** C → C-E → C-E-G → C-E-G-B (full chord)
- **Shrinking:** C-E-G → C-E → C → (repeats)

The window size grows and shrinks based on the **natural chord note count** (not the step count), so a 4-note chord cycles through window sizes 1-2-3-4-3-2-1 (16 steps total per cycle). This creates an organic, meditative quality characteristic of minimalist composition.

Additive mode works well with slow rates and long step counts, allowing the gradual expansion to unfold naturally.

## Evolve

The **EVOLVE** control introduces gradual pitch mutations to the arpeggio sequence over successive cycles. It replaces the former R.GATE parameter.

Parameter	Range	Description
Evolve	0-100%	Mutation probability per step per cycle

- **0%:** Pattern lock mode – for **random arp modes** (Random, TonalRandom, AtonalRandom), the random sequence is auto-captured after the first complete cycle and then replayed identically on every subsequent cycle. This turns a random sequence into a repeatable loop. For standard modes, no mutation occurs
- **Low values (5-20%):** Slow, subtle pitch drift – patterns evolve organically over many cycles
- **High values (50-100%):** Rapid mutations – patterns change significantly each cycle

## Behavior by mode

- **Standard modes** (Up, Down, etc.): Per-step pitch offsets accumulate and mutate each cycle. At 0%, no offsets are applied
  - **Random modes** (Random, TonalRandom, AtonalRandom): At 0%, the random sequence auto-captures after the first cycle completes and locks into a repeatable loop. At >0%, the captured sequence mutates gradually each cycle
  - **Euclidean:** Pitch mutations apply while the underlying rhythmic pattern (pulse distribution) stays fixed
- EVOLVE can also be modulated via the matrix (Chord/Arp > Evolve destination).

## Random Sequence Lock (RND SEQ)

The **Random Sequence Lock** allows you to capture a random arpeggio sequence and preserve it across note retriggering. Without lock, each new note-on generates a fresh random sequence. With lock enabled, the captured sequence persists until you manually unlock it.

### How to enable

- **Click the arp pattern display** (in main UI or HarmPadre editor) — the display shows a red border when locked
- **Click the LOCK button** in the HarmPadre editor (right of the Evolve fader, labeled “RND SEQ”)

### Behavior

- Only active in **random modes** (Random, Tonal Random, Atonal Random) — the button remains visible but dimmed in other modes
- The sequence auto-captures after the first complete cycle (same as Evolve at 0%)
- While locked, releasing and re-pressing a note replays the same sequence
- **Evolve continues to work** on a locked sequence — mutations accumulate but the base sequence survives retrigger
- Switching to a non-random arp mode automatically unlocks the sequence
- Lock state is saved with presets and DAW sessions

### Use cases

- Find a random pattern you like, lock it, then play different chords — same rhythmic pattern, different harmonies
- Lock a sequence, then gradually increase Evolve to let it drift from the original
- Use with Tonal Random for scale-constrained melodies that stay consistent

## RND MODE (Random Contour)

**RND MODE** allows you to apply a musical contour (ordering pattern) to the arpeggiator’s random sequences. Instead of random notes playing in an unstructured order, the contour organizes them into structured melodic patterns — combining the unpredictability of random note selection with the musicality of a defined contour.

Each arpeggiator has its own independent RND MODE control: - **RND1 MODE** (blue text): Contour for Arp1 - **RND2 MODE** (yellow text): Contour for Arp2

Both buttons are always visible in the HarmPadre editor (above and below the LOCK button), but only enabled when their corresponding arpeggiator is in a random mode (Random, Tonal Random, or Atonal Random).

## Contour options (15)

#	Label	Category	Effect
0	OFF	—	Random selection without contour (original behavior)
1	Up	Directional	Selected notes sorted ascending
2	Down	Directional	Selected notes sorted descending
3	U&D	Directional	Ascending then descending
4	D&U	Directional	Descending then ascending
5	I&O	Contour	Center outward
6	O&I	Contour	Edges inward (low-high zigzag)
7	I/O	Contour	I&O followed by O&I
8	O/I	Contour	O&I followed by I&O
9	x2	Pattern	Each selected note repeated twice
10	x4	Pattern	Each selected note repeated four times
11	St-Up	Pattern	Stairs ascending (2 up, 1 back)
12	St-Dn	Pattern	Stairs descending (2 down, 1 back)
13	Conv	Contour	Converge from edges
14	Div	Contour	Diverge from center

### How it works

- The note pool is built based on the active random mode:
  - **Random:** chord notes × octaves (range)
  - **Tonal Random:** all scale notes × octaves
  - **Atonal Random:** all 12 chromatic notes × octaves
- On key press, N random notes are selected from the pool (N = number of steps)
- The selected notes are ordered by the chosen contour
- The sequence plays linearly and repeats while the key is held
- On release and re-press (without Lock), a new random selection is generated with the same contour
- With Lock enabled, the same selection is preserved across retriggers

### Interaction with other controls

- **Lock:** Preserves the note selection across retriggers (same as random mode without contour)
- **Evolve:** Pitch mutations are applied on top of the contoured sequence
- **Step Sequencer:** TRANS, VELO, GATE, and PROB are applied on top of the contour
- **Range:** More range = larger pool = more variety in random selection

### Practical example

- Mode: Tonal Random + RND1 MODE: O&I + Scale: C Major + Range: 3
- Press a key → selects 8 random notes from the 21 scale notes (3 octaves)
- O&I contour orders them in a low-high zigzag: E3, C5, G3, A4, B3, F4, D4, A3
- Result: a melody that alternates between bass and treble — musical yet different every time

## Available Scales (34)

Portador includes an extensive collection of scales organized by categories:

### Principal Modes (7)

Scale	Description
Ionian (Major)	Major scale - bright and happy
Dorian	Minor with major 6th - jazz, funk
Phrygian	Minor with minor 2nd - flamenco, metal
Lydian	Major with augmented 4th - ethereal, cinematic
Mixolydian	Major with minor 7th - blues, rock
Aeolian (Minor)	Natural minor scale - melancholic
Locrian	Diminished - tense, unstable

### Minor Variants (3)

Scale	Description
Harmonic Minor	Minor with major 7th - classical, neoclassical metal
Melodic Minor	Minor with major 6th and 7th - jazz
Jazz Melodic Minor	Ascending melodic minor in both directions

### Symmetric Scales (2)

Scale	Description
Diminished (Whole-Half)	Alternating whole and half steps
Whole Tone	Only whole steps - impressionistic, dreamy

### Other Scales (5)

Scale	Description
Phrygian Major	Phrygian with major 3rd - flamenco
Lydian Augmented	Lydian with augmented 5th
Lydian Dominant	Lydian with minor 7th - fusion
Altered (Superlocrian)	All altered degrees - advanced jazz
Double Harmonic (Byzantine)	Arabic/Byzantine scales

### Experimental Scales (4)

Scale	Description
Enigmatic	Enigmatic scale - mysterious
Harmonic Major	Major with minor 6th
Chromatic	All 12 notes - atonal
Prometheus	Scriabin's scale - mystical

### Modern Modal Scales (4)

Scale	Description
Mixolydian b6	Mixolydian with minor 6th
Lydian #5	Lydian with augmented 5th
Lydian b7	Same as Lydian Dominant
Dorian #4	Dorian with augmented 4th

### Artificial Scales (3)

Scale	Description
Ladder Scale	Ladder scale - contemporary
Spanish Octatonic	Spanish 8-note scale
Major with #4 and #5	Major with augmented 4th and 5th

### Messiaen Modes (5)

Olivier Messiaen's modes of limited transposition — symmetric scales that repeat at intervals shorter than an octave, creating a distinctive floating, non-tonal quality. Widely used in film scoring and contemporary classical music.

Scale	Notes	Intervals	Character
Mode 3	9	2-1-1-2-1-1-2-1-1	Shimmering, iridescent — used by Messiaen for bird-song harmonies
Mode 4	8	1-1-3-1-1-1-3-1	Dark, mysterious — contains augmented triads
Mode 5	6	1-4-1-1-4-1	Sparse, angular — wide leaps create an alien quality
Mode 6	8	2-2-1-1-2-2-1-1	Rich, ambiguous — blends major and minor qualities
Mode 7	10	1-1-1-2-1-1-1-1-2-1	Dense, chromatic — nearly all 12 notes, very atonal

Note: Messiaen Mode 1 is the Whole Tone scale (already available under Symmetric Scales) and Mode 2 is the Diminished (Whole/Half) scale.

## Notes Modes (Chord Voicing Size)

The **Notes** parameter determines how many voices each chord uses. Each mode produces a different chord density:

Mode	Name	Notes Used	Description
1	Single	Root only	Monophonic – plays only the root note
2	Fifth	Root + 5th	Power chord – root and perfect fifth
3	Triad	Root + 3rd + 5th	Standard triad (major or minor depending on chord type)
4	Seventh	Root + 3rd + 5th + 7th	Full 7th chord (default). Click to open submenu with 4 voicing options
5	Ninth	Root + 3rd + 5th + 7th + extension	5-note chord with configurable extension (see 5th Chord Extension below)
<b>Shell</b>	Shell	Root + 3rd + 7th	3-note jazz voicing that omits the 5th. Common in jazz comping where the 5th is considered redundant. Shell voicings keep harmony clear and work especially well in dense arrangements or when combined with bass instruments

**Tip:** Shell voicings are ideal for jazz comping, lo-fi beats, and situations where you want harmonic clarity without the thickness of full 7th chords. They pair well with the Rootless finger modes for a complete jazz voicing toolkit.

**Tip – 1-voice mode (Single):** The 1-voice option turns the chord generator into a scale-aware monophonic melody tool. The output is just the degree's root note, but every other system on top still works (arp, TRIG HARM, voice leading, glide). This unlocks the Richter / Einaudi trick: route some oscillators to a 3-voice triad (Notes 3) and others to a 4-voice add9 arp (Notes 4 → Add9), and run a 1-voice (Notes 1) lead on top – three independent layers all driven by the same chord sequence.

## Notes 4 Sub-Modes (Tetrad Voicing Options)

When selecting **4** in the Notes menu, a submenu appears with 5 voicing options. These all produce 4-note chords but with different interval selections:

Sub-Mode	Intervals	Example (CMaj7)	Character
<b>Tetrad</b>	1-3-5-7	C-E-G-B	Standard 7th chord (default)
<b>Add9</b>	1-3-5-9	C-D-E-G	Cinematic – the 9th is placed <i>within</i> the octave (as a major 2nd above root), creating the “Arrival of the Birds” pattern when arpeggiated with Up&Down mode
<b>Open9</b>	1-5-7-9	C-D-G-B	Open voicing without the 3rd – ambiguous, ethereal quality
<b>Shell9</b>	1-3-7-9	C-D-E-B	Jazz shell with 9th extension – skip the 5th, add color
<b>Oct</b>	1-3-5-8	C-E-G-C'	Triad with octave doubling – the bass note is doubled one octave higher. Fuller than a simple triad while maintaining harmonic clarity

**Key design detail:** The 9th note is placed as a major 2nd above the root (not an octave higher). When the arpeggiator sorts by pitch, the sequence becomes Root-9th-3rd-5th. With Up&Down arp mode, this produces: C-D-E-G-E-D – exactly the cinematic ascending/descending pattern heard in pieces like “Arrival of the Birds”.

**Tip:** Combine **Add9** with arp mode **Up&Down** for instant cinematic arpeggios. Use **Shell9** for modern jazz voicings that combine shell economy with 9th color.

## 5th Chord Extension

When the **Notes** mode is set to **5**, a fifth voice is added to each chord. The 5th Chord Extension parameter determines which tension note is used for this extra voice.

### How to enable

1. Click the **Notes** menu in the Chord Engine section
2. Select “**5**” to activate 5-note chords
3. A submenu appears showing “**5TH NOTE**” with 7 extension options

## Extension Options

Extension	Name	Interval from Root	Character
<b>b9</b>	Minor 9th	+13 semitones	Dark, dissonant — dominant and blues chords
<b>9</b>	Major 9th (default)	+14 semitones	Bright, modern — the most common extension
<b>#9</b>	Augmented 9th	+15 semitones	Edgy, aggressive — the “Hendrix chord” sound
<b>11</b>	Perfect 11th	+17 semitones	Smooth, suspended — adds a 4th flavor
<b>#11</b>	Augmented 11th	+18 semitones	Tense, chromatic — tritone tension
<b>b13</b>	Minor 13th	+20 semitones	Jazz color — adds a minor 6th quality
<b>13</b>	Major 13th	+21 semitones	Warm, full — rich and sophisticated

### Example with C Major 7th (C-E-G-B)

- With **9**: C-E-G-B-D (adds major 2nd above the octave)
- With **b9**: C-E-G-B-Db (adds minor 2nd — dark tension)
- With **#11**: C-E-G-B-F# (adds tritone — Lydian color)
- With **13**: C-E-G-B-A (adds major 6th — warm and full)

**Note:** The extension note is placed one octave above the root to avoid clashing with the fundamental chord tones. Each scale degree can have a different extension configured independently.

**Tip:** Use **9** for modern pop and R&B voicings, **b9** for dominant tension in jazz, **#9** for funky or blues chords, and **13** for lush, orchestral-style harmony.

## Step Probability

Each arpeggiator step has an independent probability value that determines whether the step plays or is silenced.

Parameter	Range	Description
Probability	0-100%	Probability that the step will sound (default 100%)

- **100%:** Step always plays (normal behavior)
- **0%:** Step never plays (always silenced)
- **Intermediate values:** Step plays randomly based on probability

When a step is silenced by probability, a noteOff is sent for the previous note but no noteOn is emitted, creating a natural rhythmic silence.

**Interaction:** Drag vertically to adjust, scroll wheel when enabled, double-click to reset to 100%.

**Tip:** Use low probabilities (20-40%) combined with high Density for generative patterns that vary on each repetition.

## Gate (Note Duration)

Controls the duration of each arpeggiator note as a percentage of the step duration. Includes a global control and per-step override capability.

### Global Gate

Parameter	Range	Description
Gate	0-100%	Note duration relative to step (default 80%)

- **100%:** Note lasts the entire step (legato)
- **50%:** Note lasts half the step (moderate staccato)
- **10%:** Very short note (extreme staccato)

### Per-Step Gate

Parameter	Range	Description
Step Gate	-1 (global) or 0-100%	-1 = use global gate, 0-100% = individual override

**Interaction:** Drag vertically to adjust, scroll wheel when enabled, double-click to reset to -1 (global).

**Tip:** Combine short gates with reverb or delay for rhythmic textures with space. Use long gates (90-100%) for legato melodic lines.

## New Arpeggio Sources

### AS PLAYED

The **As Played** source (and **As Played 2** for the second arpeggiator) outputs the currently held MIDI notes in the exact order you played them, cycling at the current arp rate. Unlike the regular Arp source, the pitch comes straight from your keyboard — the chord engine does not transform it.

- **Monophonic cycling:** the source steps through held notes 1 → 2 → 3 → ... and wraps around
- **Release-safe:** if you lift one key while others are held, the remaining notes continue cycling without a pitch glitch
- **MIDI-like:** As Played behaves like a MIDI source in the UI — oscillators using it co-exist with keyboard-driven layers and are mutually exclusive with Chord/Arp sources (a confirmation dialog appears when you switch)
- **Degree LEDs:** the HarmPadre degree indicators do NOT illuminate for As Played (same as MIDI)
- **Trig Harm exception:** As Played works alongside Gate Generator Trig Harm — the gate triggers the chord/arp layer while As Played plays your held notes independently

**Use cases:** pluck-style sequences that follow your fingering, duophonic patterns by alternating two held notes, simple melodic loops that match the chord sequencer rhythmically without harmonising it.

## RETROGRADE

Plays the arpeggio sequence in reverse order. No parameters — simply reverses the direction of the arp pattern.

**Example:** If the arp plays C, E, G, B... Retrograde plays B, G, E, C...

## FUGUE

A diatonic canon — replays the arp pattern delayed and transposed by **scale degrees** (not chromatic semitones). This ensures the result always stays in key, unlike the old chromatic Canon.

Parameter	Range	Description
Delay	1-8 steps	How many arp steps to look back in the history
Interval	-Oct to +Oct	Transposition in scale degrees (diatonic)

### How it works

- A ring buffer stores the last 32 arpeggio notes
- Fugue reads back `delay` steps in the history
- The interval transposes by scale degrees, not semitones
- Example in C Major: a +3rd interval transposes C→E, D→F, E→G (all stay in scale)

**Tip:** Set one oscillator to Arp, another to Fugue with `delay=4` and `interval=+5th` for a classic fugal texture.

## ARP HARMONICS

Generates notes based on the natural harmonic series of the arpeggio's root note.

Parameter	Range	Description
Harmonics Count	4-16	Number of harmonics in the series (default 8)
Harmonics Odd Only	On/Off	Odd harmonics only (default Off)

### How it works

- Calculates the frequency of each harmonic: `freq × harmonic_number`
- Converts each frequency to a MIDI note
- The arpeggio step cyclically indexes the harmonic sequence
- The sequence rebuilds when the chord changes

**Odd Only:** When enabled, only odd harmonics (1, 3, 5, 7...) are used, producing a timbre similar to a square wave with a hollow character.

**Tip:** Use Harmonics with few harmonics (4-6) for subtle tonal effects, or many (12-16) for complex spectral textures.

### Negative Harmony

The **Neg. Harmony** source applies the Levy/Collier negative harmony transform to the arpeggio note. Each note is mirrored around the axis between the root and perfect 5th of the current key. **Now scale-aware:** the result is automatically snapped to the nearest scale note, preventing out-of-key results.

**Formula:** `negPC = (2 * rootPC + 7 - notePC) mod 12`

## Examples (key of C)

- C -> G, D -> F, E -> Eb, G -> C, A -> Bb
- A Cmaj chord becomes Fmin, G7 becomes Fm6

## Use cases

- Create unexpected but harmonically related melodic variations
- Generate mirror-image bass lines
- Explore the “dark side” of chord progressions

## Swing

Controls the rhythmic feel of the arpeggiator by offsetting even-numbered steps.

Parameter	Range	Description
Swing	0-100%	Swing amount (0% = straight, 100% = maximum shuffle)

## How it works

- At 0%, all steps have equal duration (straight feel)
- As swing increases, even steps are delayed and odd steps are shortened
- At 50%, you get a moderate shuffle feel (similar to triplet swing)
- At 100%, maximum swing creates an extreme shuffle

**Tip:** Use 30-50% swing for a natural groove feel. Higher values work well with slower arp rates for jazz or hip-hop feels.

## Groove Templates

The **Groove** menu provides 8 genre-specific timing and velocity templates that replace the standard swing with complex rhythmic feels. When a groove template is active, the Swing fader is dimmed (groove overrides swing).

Template	Style	Description
<b>Off</b>	—	No groove applied (use Swing for simple shuffle)
<b>Dilla</b>	Hip-hop	J Dilla-style lazy, behind-the-beat feel with heavy velocity accents
<b>Jazz</b>	Jazz	Subtle triplet swing with dynamic accent patterns
<b>Bossa</b>	Latin	Bossa nova rhythmic displacement with syncopated velocity
<b>DnB</b>	Electronic	Drum and bass breakbeat timing with sharp velocity contrasts
<b>Soul</b>	R&B	Relaxed soul/R&B pocket groove with warm dynamics
<b>House</b>	Dance	Four-on-the-floor house groove with offbeat emphasis
<b>Trap</b>	Hip-hop	Trap-style timing with hi-hat roll accents

### How it works

- Each template defines per-step timing offsets (how much each step is pushed or pulled relative to the grid) and velocity scaling (accent patterns)
- The groove applies to arpeggiator steps only, not chord mode
- Select a groove from the GROOVE menu in the HarmPadre editor
- Only one groove can be active at a time; selecting “Off” returns to standard timing

**Tip:** Grooves work especially well with longer arp sequences (16-32 steps) where the rhythmic variation becomes more apparent. Combine with Humanization for even more organic results.

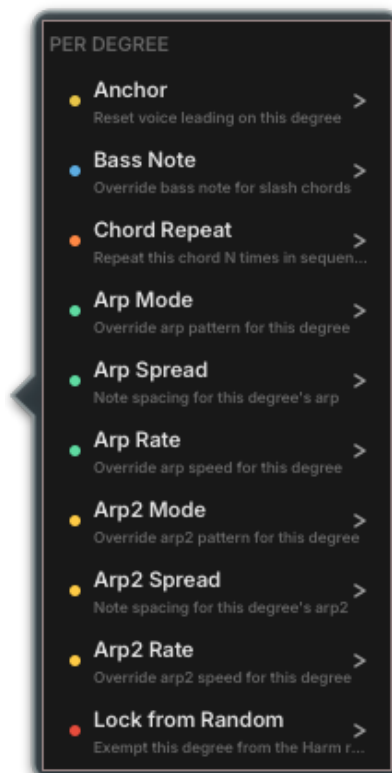


Figure 26: Per-degree right-click menu — chord type, arp mode override, tritone substitution, slash chord and other per-degree options.

## Tritone Substitution

**Tritone Substitution** is a jazz harmony technique where a dominant chord is replaced by the dominant chord a tritone (6 semitones) away. In Portador, this can be enabled per degree.

### How to enable

1. In the HarmPadre editor, click a degree button to open its menu
2. If the degree has a dominant-family chord type (7, aug7, or any UST type), a **Tritone Sub** option appears
3. Enable it to substitute the chord

### Behavior

- When enabled, the chord root is shifted up by 6 semitones and the chord type is forced to Dominant 7th
- A small colored dot appears on the degree button indicating tritone substitution is active
- The chord display shows the substituted chord name (e.g., degree V in C Major shows “Db7” instead of “G7”)
- **While tritone sub is active, the chord type is locked** — you cannot change the chord type until you disable tritone substitution first. This is because the tritone sub technique specifically requires a dominant chord

### Example (C Major)

- Degree V = G7 → with tritone sub → Db7

- The Db7 resolves to C with the same strong voice leading as G7 (the tritone interval F-B is present in both chords)

## Slash Chords (Bass Override)

**Slash chords** allow you to specify a bass note different from the chord root, creating chord inversions and polychord effects. This is configured per degree.

### How to enable

1. In the HarmPadre editor, click a degree button to open its menu
2. Select **Slash** and choose a bass note (C through B)
3. Select **“No Bass”** to remove the override

### Behavior

- The bass note is placed one octave below the chord root note (the MIDI key you press), ensuring predictable octave placement regardless of finger/voicing spread
- The degree button shows a “/” indicator when a slash chord is active
- The chord display shows the slash notation (e.g., “CMaj7/G”)
- Works in combination with Tritone Substitution

**Example:** Setting bass override to G on a CMaj7 chord creates CMaj7/G (second inversion with G in the bass).

## Voice Leading Anchor

When using Auto Voice Leading, the same chord can sound different depending on what was played before it. The **Anchor** feature locks a degree’s voicing so it always sounds the same, regardless of the previous chord.

### How to use

1. Play your chord progression until the anchored degree sounds the way you want
2. Right-click the degree key → **Anchor** → **On**
3. The exact voicing is captured and replayed every time that degree is triggered
4. Other (non-anchored) degrees continue to voice-lead normally from the anchor

**Visual indicator:** A golden left border appears on anchored degree keys.

**When to use:** Set the first chord of your progression as anchor. This ensures the entire progression sounds identical every time it loops, because all subsequent chords voice-lead from the same starting point.

The anchor works in all chord modes — both the standard chord generator (Chord Single 1-5) and the Chord All recognition path. The anchor voicing is saved with chord/arp presets and DAW state.

## Empty Per-Degree Slot Behavior

When using Chord/Arp source modes (Chord1..5, Arp, Fugue, Retrograde, Polytonal, NegHarmony, Chord All), playing a chromatic note that falls outside the current scale and has **no manual chord type assigned** at that degree produces **silence** for chord-source oscillators. This is by design: each per-degree slot represents an explicit “what should this note play” decision, and an empty slot means “do not play anything for this input note”.

If you want chromatic input notes to always sound, the alternative routing modes cover those cases:

- **Source = MIDI:** chromatic passthrough — every note plays the oscillator at its own pitch
- **Gate Generator with Trig Harm:** you can play chromatic notes on the keyboard while the chord pattern triggers harmonically — both layers coexist
- **Channel split** (Settings → MIDI): if Synth Ch and Harm Ch are set to different channels, MIDI-source and Chord/Arp-source oscillators can coexist on separate channels

**Tip:** If you want chord coverage on all 12 notes (e.g., for chromatic passages), enable Free Chord Mode (next section) and assign a chord type to every degree slot you want to play.

## Free Chord Mode

**Free Chord Mode** allows you to build a custom chord palette without being constrained by a scale. Each of the 12 chromatic positions can be independently assigned any root note and chord type.

### How to enable

1. In the Chord/Arp preset menu, select **“Free Chord”**
2. All 12 degree slots become empty (no chord assigned)
3. Click each degree button and assign a chord type from the menu
4. The root note for each slot corresponds to its chromatic position (C, C#, D, etc.)

### Behavior

- Scale and Root menus are dimmed (not applicable in free chord mode)
- Each slot starts empty — you populate only the chords you need
- All per-degree settings (Notes, Voicing, Finger, Octave, Humanization) work normally
- Tritone Substitution and Slash Chords work independently per slot
- Loading a scale preset exits free chord mode and returns to scale-based chord generation

### Use cases

- Building non-diatonic chord progressions (e.g., chromatic mediant, modal interchange)
- Creating custom chord palettes for specific songs
- Experimenting with unconventional harmonic relationships

## Chord All Mode

**Chord All** plays the full generated chord using multiple voices (one per chord note). It is only available in POLY mode.

**How it works:** When you press a key, Portador generates the chord for that degree and allocates one voice per chord note. Voice leading is always active — when you press a new key, the chord transitions smoothly by minimizing total voice movement.

**Last-note-wins:** Pressing C plays CMaj7. While holding C, pressing D voice-leads to Dmin7. Releasing D returns to CMaj7.

**Alternative:** Use Chord Single voices (1-5) instead — each oscillator plays one voice of the chord with its own timbre, pan, and level. Combine with MOSC (right-click on the OSC label) for monophonic chord-voice behavior.

### Creative combinations

- **Osc1: Chord All** (voice-led pad) + **Osc2: Arp** (rhythmic melody) — layered pad with arpeggio

- **Osc1: Chord All + Osc2: Chord 2** — reinforce the 3rd of the chord with a different timbre
- **Osc1: Chord All + Osc2: Arp > Fugue + Osc3: Arp > Neg. Harmony** — voice-led pad with canonic counterpoint

**Combining MIDI and Chord All sources:** When any oscillator has a chord or arp source, the keyboard switches to Chord Mode (see Keyboard Modes below). To play a free melody alongside generated chords, use TRIG HARM — it re-enables MIDI source for free play over sequenced chords.

**Tip:** Chord All works with all per-degree settings (Notes mode, Voicing, Finger, Humanization).

## Keyboard Modes

Portador operates in one of two keyboard modes:

### MIDI Mode (Default)

When all oscillators are set to **MIDI** or **None** source, the keyboard works normally. Each key plays the note you press. The chord recognizer is active for display only.

### Chord Mode

When any oscillator has a chord or arp source (**Chord All**, **Chord Single 1-5**, or any **Arp** variant), the keyboard switches to Chord Mode:

- The keyboard becomes a **chord slot selector**. Each key activates its corresponding chord slot.
- **MIDI source is automatically silenced** — oscillators set to MIDI switch to None.
- **Empty chord slots = complete silence**. No sound, no envelope trigger.
- **None source** remains silent. Use it to disable specific oscillators in Chord Mode.

**Exception — TRIG HARM:** When Gate Generator TRIG HARM is active, MIDI source becomes available again. The gate sequencer handles chord selection, freeing the keyboard for normal MIDI playing. This is the recommended way to combine free melody with generated chords. In HOST mode this works thanks to the dedicated Gate Trigger Note (see Gate Generator → SYNC).

**Source Mode Change dialog:** When you switch an oscillator between a MIDI-like source (MIDI, As Played, As Played 2) and a Chord/Arp source, if the change would force other oscillators to flip to **MIDI** or **None**, Portador shows a **Source Mode Change** confirmation dialog listing how many oscillators will be affected. Click **Accept** to apply, or dismiss to cancel the change.

## Global Configuration

Parameter	Description
Scale	Base scale
Scale Root	Root note (C to B)
Voicing	Root, Inv1, Inv2, Inv3, Auto Smooth, Auto Classical, Auto Jazz, Auto Modern, Auto Piano (default)
Finger	Close, Open, Spread, Tight, Wide, Drop2, Drop3, Drop2+4, Upper, Alt, Rootless A, Rootless B, Cluster, Auto, Piano
Octave	Octave transposition
Arp Range	Arpeggio range (1-9 octaves)
Arp Rate	Arpeggio speed
Arp Steps	Number of steps (1-32)
Arp Release	Release time for arpeggio notes (0-100%). Controls how long each arp step rings after its gate closes, from staccato (0%) to fully legato (100%)
Retrigger	Arpeggio reset

**Voice Leading Styles:** When voicing is set to an **Auto**

mode, Portador automatically applies voice leading algorithms to create smooth transitions between chords. There are five voice leading styles, each modeled after real musical traditions:

Style	Label	Description
<b>Auto Smooth</b>	A	Minimizes total voice movement by trying all inversions and picking the one with minimum semitone distance. Pure mathematical optimization — no style-specific rules. Creates generic smooth transitions
<b>Auto Classical</b>	C	Follows SATB (Soprano-Alto-Tenor-Bass) counterpoint rules. Checks parallel fifths and octaves between <b>all voice pairs</b> (not just outer voices). Rewards contrary motion between bass and soprano. Resolves 7ths downward by step. Penalizes leaps larger than a 3rd in any voice. The most rule-based mode
<b>Auto Jazz</b>	J	Guide-tone voice leading. The 3rd and 7th (the tones that define chord quality) are weighted 3x heavier — they should move by semitone or stay as common tones. Root is free to move (weight 0.7x), top voice has melodic freedom (0.5x). Penalizes parallel guide-tone movement (mechanical). Rewards 7th→3rd semitone resolution (the essence of ii-V-I). Works with triads too (3rd weighted 2.5x)
<b>Auto Modern</b>	M	Cinematic/ambient style (Hans Zimmer, Nils Frahm). Upper voices form a <b>static cluster</b> (soprano weight 5x, upper inner 3.5x) while the bass moves freely underneath (weight 0.3x). Rewards chromatic drift in upper voices and dramatic bass leaps (4ths/5ths). Creates the “floating pad” effect where harmony changes feel subtle
<b>Auto Piano</b>	P	<b>Default.</b> Models real two-hand piano playing. Bass prefers pedal point (common tone bonus -15) or stepwise motion (2nds/3rds bonus -6), penalizes leaps >5th (+8). Soprano anchored with stepwise melodic bonus. Inner voices fill by semitones. Right-hand cohesion check penalizes spread >9th. The most “human-feeling” mode

**Adaptive Finger:** Each voice leading style dynamically adapts

the finger voicing based on the chord type, register, and number of voices — just like a real musician:

<b>Style</b>	<b>Default Finger</b>	<b>Adaptations</b>
<b>Smooth</b>	Auto (octave-match)	Wide on first chord, then minimizes movement
<b>Classical</b>	Wide	Low register (< C3) → Open; high register (> C5) → Close; diminished → Close
<b>Jazz</b>	Drop 2	Dominant 7th → Rootless A (low root) or B (high root) by register; diminished → Close; low register → Drop 3
<b>Modern</b>	Spread	Diminished/augmented → Cluster; low register → Wide; 5 voices → Drop 2+4
<b>Piano</b>	Piano	Very low register → Open; very high → Close; dominant 7th → Drop 2

## Finger Modes (Voicing Style)

The **Finger** parameter controls the spatial distribution of chord notes across the keyboard, determining the voicing character:

Mode	Description
<b>Close</b>	Standard close voicing – all notes within one octave
<b>Open</b>	Open voicing – notes spread across a wider range
<b>Spread</b>	Wide spread – notes distributed across two or more octaves
<b>Tight</b>	Tightly packed voicing – minimal intervals between notes
<b>Wide</b>	Very wide distribution – maximum spacing between chord tones
<b>Drop 2</b>	Drop 2 voicing – the 2nd highest note is dropped one octave. Standard jazz guitar and big band voicing
<b>Drop 3</b>	Drop 3 voicing – the 3rd highest note is dropped one octave. Common in jazz piano
<b>Drop 2+4</b>	Drop 2+4 voicing – the 2nd and 4th highest notes are both dropped one octave. Open, orchestral character
<b>Upper</b>	Upper structure – keeps notes in the upper register
<b>Alt</b>	Alternating octave displacement for notes
<b>Rootless A</b>	Jazz rootless voicing type A: 3rd, 5th, 7th, 9th (no root). The bass player or context provides the root. Standard left-hand jazz voicing
<b>Rootless B</b>	Jazz rootless voicing type B: 7th, 9th, 3rd, 5th (no root, inverted order). Alternative left-hand jazz voicing, avoids voice crossing with Rootless A
<b>Cluster</b>	Cluster voicing – all chord notes compressed within a very small range (close voicing with minimal spacing). Creates dense, colorful harmonic clouds. Effective for impressionistic textures and modern jazz
<b>Auto</b>	Inherits the spread from the previous chord. Each note is octave-matched ( $\pm 24$ semitones) to minimize distance from the corresponding voice in the previous voicing. The first chord uses <b>Wide</b> spread for a pianist-like open voicing. Ideal for smooth voice-leading transitions. In <b>Chord All</b> mode, this is automatically applied to all degrees after the first chord
<b>Piano</b>	Classic pianist spread voicing. Root and 5th drop one octave while the 3rd stays in place. For Am: produces A2-E3-C4 – root in the bass, 5th in the middle register, 3rd on top. Creates the natural open voicing a pianist would use. Offsets: [-1, 0, -1, 0, +1]. Ideal as a starting voicing before Auto voice leading for smooth jazz/classical progressions

**Tip:** Rootless A and Rootless B are designed to be used together – assign Rootless A to one degree and Rootless B to the adjacent degree to create smooth voice leading without root notes, as in traditional jazz piano technique. Cluster voicing works especially well with dissonant chord types (dim7, altered) for dramatic textural effects.

## Humanization

Mode	Description
Off	No humanization
Tight	Minimal variation
Natural	Natural variation
Expressive	Expressive variation
MelodyLead	Emphasis on high note
BassLead	Emphasis on low note
RollUp	Upward strum
RollDown	Downward strum

## Velocity Modes

Normal, Humanize, Up, Down, Up&Down, Down&Up, Random, In&Out, Out&In

## POLY Mode and Chords

In **POLY mode**, the chord generator includes automatic chord recognition: the system automatically detects the chord you're playing with multiple keys, and the Chord 1-5 sources are available in all polyphony modes.

## Chord/Arp Gate In (CV-Triggered Chords)

The chord engine and arpeggiator can be triggered by modulation sources instead of MIDI notes using the **H Gate In** matrix destination.

### Setup

1. In the modulation matrix, select "**H Gate In**" as a column destination (found in the Harm submenu)
2. Connect a modulation source (LFO, envelope, etc.) to that column
3. The chord engine will trigger on rising edges (>50%) and release on falling edges

### How it works

- When the CV signal rises above 50%, a chord is generated and the arpeggiator starts
- When the CV drops below 50%, the arpeggiator stops and envelopes release
- The chord root note is derived from currently held MIDI notes, or from oscillator tune/octave params when no MIDI notes are held
- **Modular envelope control:** H Gate In does NOT auto-trigger envelopes. To have envelopes respond to H Gate In, connect each envelope individually to its own Gate In destination (Env1 Gate In, Env2 Gate In, or VENV Gate In) in the modulation matrix. This gives you independent control — for example, use ENV1 with Gate In for auto-chords on CH1 while keeping ENV2 MIDI-triggered for keyboard playing on CH2
- **Auto-trigger drums:** H Gate In also triggers the Drum engine on any oscillator in Drum mode (same as envelope Gate In)

---

## Harm Root (CV-Modulated Chord Root)

The **Harm Root** matrix destination allows modulating the chord root note in real-time via CV:

- The CV value is scaled to semitones ( $\pm 12$  range at full bipolar modulation)
- When a scale is active, the modulation is quantized to the nearest scale interval
- This allows creating melodic chord progressions driven by LFOs, envelopes, or other modulation sources

### Creative uses

- **Rhythmic chords:** Connect an LFO square to H Gate In for tempo-synced chord triggering without MIDI
- **Chord progressions from LFO:** Use Harm Root with a slow LFO to cycle through chord roots automatically
- **Drone chords:** Set oscillator tune to a root note, connect an LFO to H Gate In, and create rhythmic drone chords without any MIDI input
- **Polyrhythmic harmony:** Use different LFO rates for H Gate In and Harm Root to create evolving harmonic patterns

## HarmPadre Editor

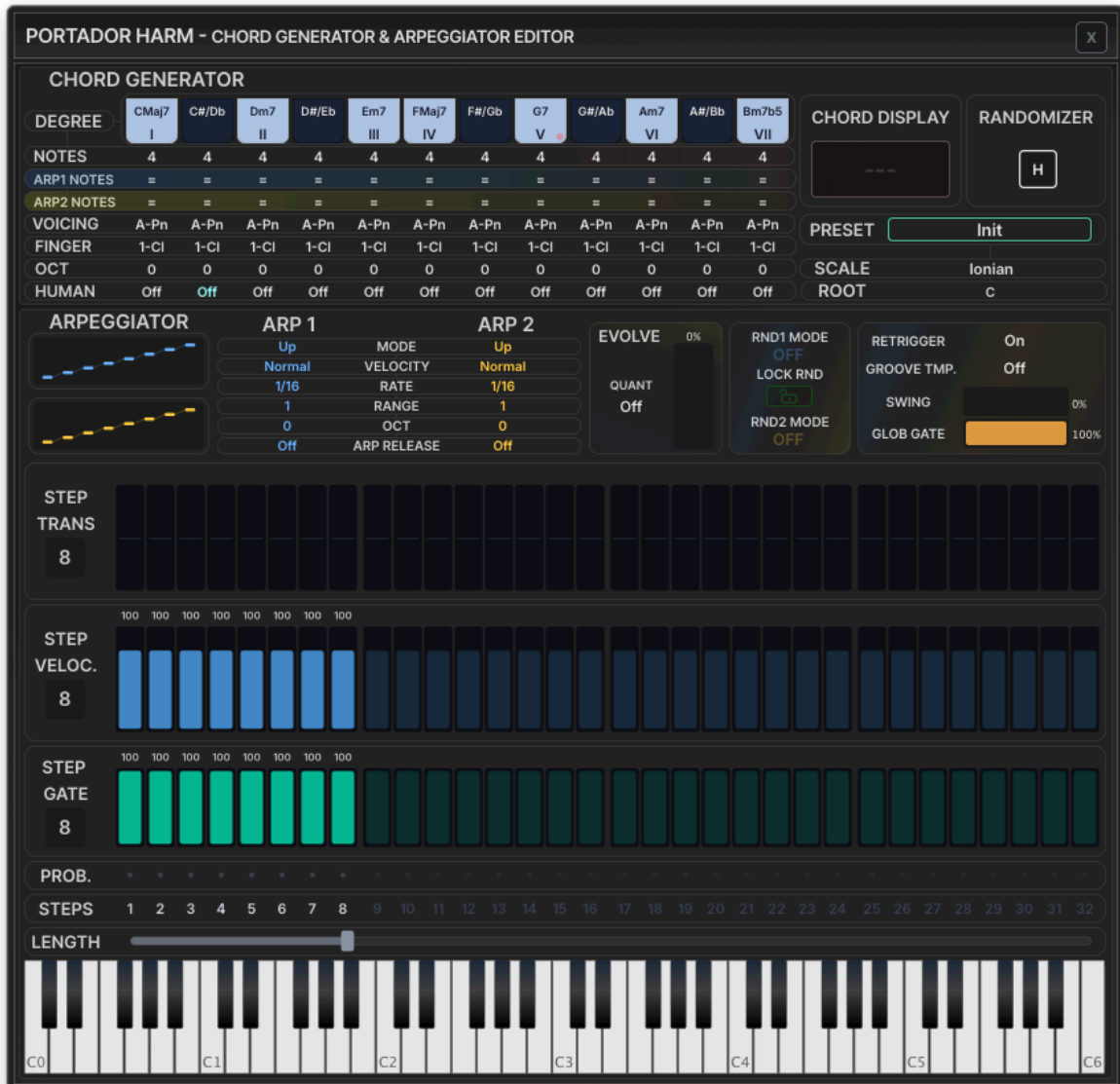


Figure 27: HarmPadre — chord generator and arpeggiator editor with per-degree configuration, dual-arp step grid and full piano roll.

The **HarmPadre Editor** is a dedicated floating window that provides a comprehensive view and control of the entire chord and arpeggiator system. Open it by clicking the **HARM** button in the Chord Engine section header.

**Window size:** 845 x 820 pixels (resizable with UI scaling).

### Chord Palette (Top Section)

- **12 Degree Buttons:** Shows the chord name for each chromatic degree (C through B). Click to toggle degrees on/off within the current scale. Active degrees show their chord type (Maj7, m7, etc.).

• **Per-Degree Configuration** (5 rows below each degree):

Row	Options	Description
<b>NOTES</b>	1, 2, 3, 4, 5, Shell	Number of voices per chord. Mode 4 has sub-modes: Tetrad, Add9, Open9, Shell9
<b>ARP NOTES</b>	= NOTES, 1, 2, 3, 4, 5, Shell	Independent note selection for the arpeggiator. Default “=” links to NOTES. Set independently for e.g. 2-note chord + 5-note arp
<b>VOICING</b>	Root, Inv1, Inv2, Inv3, Auto	Chord inversion (Auto=minimum voice movement)
<b>FINGER</b>	Close, Open, Spread, Tight, Wide, Drop2, Drop3, Drop2+4, Upper, Alt, Rootless A, Rootless B, Cluster	Voicing style (13 modes)
<b>OCT</b>	-2 to +2	Per-degree octave transposition
<b>HUMAN</b>	Off, Tight, Natural, Expressive, MelodyLead, BassLead, RollUp, RollDown	Per-degree humanization

- **Chord Display** (right panel): Source-dependent display:
  - **Source MIDI**: Shows only the recognized chord from MIDI input. Degree LEDs do not illuminate.
  - **Source Chord/Arp**: Shows only the generated chord name. Degree LEDs illuminate the active degree.
  - **Gate Gen TRIG HARM + MIDI**: Dual display – recognized MIDI chord on top (only when keys are held), generated chord on bottom.
  - In chord/arp modes, the mismatch indicator (red text) is disabled since the display only shows the generated chord.
- **Scale / Root / Preset menus** (right panel): Quick access to scale selection, root note, and chord/arp presets

### Arpeggiator Controls

Control	Description
<b>MODE</b>	Arpeggio mode selector (19 modes)
<b>VELOCITY</b>	Velocity mode (Normal, Humanize, Up, Down, etc.)
<b>RATE</b>	Arpeggio speed/rate
<b>RANGE</b>	Octave range (shows PULSES in Euclidean mode)
<b>OCT</b>	Global arp octave offset (-3 to +3). Shifts arp notes independently from the chord octave.
<b>RETRIGGER</b>	Retrigger behavior
<b>SWING</b>	Swing amount (horizontal fader)
<b>GATE</b>	Global gate duration (horizontal fader)
<b>EVOLVE</b>	Mutation rate (vertical fader)

### ARP NOTES (Independent Arp Note Selection)

The **ARP NOTES** parameter controls which chord notes the arpeggiator uses, independently from the **NOTES** parameter (which controls the chord voicing).

- **Default: “= NOTES” (LINK)** – the arpeggiator uses the same notes as the chord. Changing NOTES automatically changes the arp notes too. This is the traditional behavior.
- **Independent mode:** Click ARP NOTES and select any Notes mode (2, 3, 4, 5, or their sub-modes). The arp will use its own note selection from the full 4-note chord, regardless of what NOTES is set to.

### Use cases

- **2-note chord + 5-note arp:** Set NOTES=2 (power chord pad) and ARP NOTES=5 (full pentad arpeggio). The sustained chord plays only root+5th while the arpeggiator plays the complete extended chord.
- **4-note chord + 2-note arp:** Set NOTES=4 (full tetrad) and ARP NOTES=2 (root+3rd). The chord voicing uses all 4 notes but the arp only alternates between root and 3rd – perfect for the “Arrival of the Birds” pattern.
- **Add9 arp over simple chord:** Set NOTES=3 (triad) and ARP NOTES=4 Add9. The chord plays a simple triad while the arp adds the cinematic 9th.

**Per-degree:** ARP NOTES is configured independently for each degree, just like NOTES. Each degree in the scale can have its own arp note selection.

**Arp Voice Leading:** In both LINK and independent modes, common tones are kept at the same pitch across chord changes, and new notes are placed in the nearest register. This creates smooth arp transitions without octave jumps.

### Arp 2 (Second Independent Arpeggiator)

Portador features a **second arpeggiator (Arp 2)** that runs simultaneously with Arp 1. Each oscillator can be independently routed to Arp 1 or Arp 2 via the Source menu, allowing two different arpeggio patterns to play at the same time.

### Independent parameters (Arp 2 has its own)

Parameter	Description
<b>MODE</b>	Arpeggio mode (same 20 modes as Arp 1)
<b>VELOCITY</b>	Velocity distribution mode
<b>RATE</b>	Arpeggio speed (independent from Arp 1)
<b>RANGE</b>	Octave range
<b>OCT</b>	Global arp2 octave offset (-3 to +3)
<b>ARP RELEASE</b>	Note release behavior

### Shared parameters (inherited from Arp 1)

- All 32 steps (TRANS, VELO, GATE, PROB)
- EVOLVE, RETRIGGER, GROOVE, SWING, GATE
- Step LENGTH (polymetric lane lengths)
- Ratchet, Slide, Hold per-step settings

Both arpeggiators read the same step data but advance independently. If Arp 1 is at step 5 and Arp 2 is at step 12, they read their respective step’s transpose/velocity/gate/probability values.

**ARP2 NOTES:** Like ARP NOTES for Arp 1, the **ARP2 NOTES** parameter allows independent note selection for Arp 2. It follows a triple LINK chain: ARP2 NOTES → ARP NOTES → NOTES. When set to “=” (LINK), it inherits from ARP NOTES; if that is also LINK, it inherits from NOTES.

**Per-degree Arp2 Mode Override:** Right-click any degree button in the HarmPadre editor to access the **Arp2 Mode** submenu. This allows overriding the global Arp 2 mode for specific scale degrees, just like the Arp 1 override.

### Per-Degree Arp Rate Override

Each chromatic degree can also have its own independent arp rate, decoupled from the global **Arp Rate / Arp2 Rate** faders. Combined with the polyphonic arp (see below), this enables true polyrhythmic textures — for example, degree I at 1/8 notes against degree V at 1/16 triplets.

#### How to assign

1. Right-click any degree button (in the main UI or the HarmPadre editor) to open the per-degree context menu.
2. Under **Arp Rate** (green) or **Arp2 Rate** (gold), choose one of the 18 rate values (1/1 → 1/64T, same list as the global Rate fader), or **Global** to clear the override and follow the global rate again.

Degrees with any active override (mode, spread, or rate) show a coloured **right-side border** on the degree button — amber for arp1 overrides, gold for arp2 overrides. This makes it easy to see at a glance which degrees carry their own settings.

**Saved with presets:** per-degree rate overrides are stored in both the DAW session and in ChordArp presets (JSON fields `arpRate` and `arp2Rate`). Missing fields fall back to “Global” for backwards compatibility with older presets.

### Polyphonic Arp (Free Chord Mode)

When multiple MIDI notes are held simultaneously and at least one oscillator is set to an Arp source, Portador runs **independent arp streams per held key** instead of collapsing them into a single chord.

- The **first** note uses the main arp engine (full PPQ sync, grid alignment, ratchet, slide, hold, etc.)
- Additional notes (up to 4 simultaneous keys) get their own lightweight poly streams that advance with their own rate and per-degree settings
- The streams share the global arp mode, step grid, swing, gate, and evolve, but read their per-degree rate / mode / spread independently
- Voice-stealing: pressing a 5th key steals the oldest stream (a note-off is emitted so nothing gets stuck)
- Releasing a key cleanly stops that stream only; other streams keep playing

**Polyrhythmic setup example:** turn on **Free Chord Mode**, assign different chord types to a handful of degrees, then give each degree its own Arp Rate (e.g. C = 1/8, E = 1/16, G = 1/16T). Play a two- or three-note cluster on the keyboard — each key drives its own independent arp at its own rate, producing the kind of layered polyrhythmic sequences you hear in minimalist/ambient piano music (Richter, Einaudi, etc.).

**HarmPadre Editor:** The arp controls section shows Arp 1 on the left and Arp 2 on the right in a symmetric layout. Both arps share the step fader grid, evolve, swing, gate, and other shared controls.

**Main UI:** The ArpPatternDisplay includes small **“1” and “2” tab buttons** in the top-right corner. Clicking “2” switches the main arp controls (MODE, VELOCITY, RATE, RANGE) to show and edit Arp 2 parameters. The pattern display color changes from **blue** (Arp 1) to **yellow** (Arp 2) to indicate which arpeggiator is being viewed. Shared controls (RETRIGGER, LENGTH, GATE, EVOLVE, SWING) remain unchanged regardless of the selected arp.

**Retrigger (shared):** When Retrigger is ON, pressing a new note resets both Arp 1 and Arp 2 simultaneously — both restart from the first step of their respective sequences. When OFF (legato), both continue from their current position.

**ARP RELEASE (independent):** Each arp has its own ARP RELEASE toggle. When enabled, releasing all keys causes that arp to continue playing with a gradual velocity fade-out instead of stopping immediately. If both arps have ARP RELEASE enabled, each fades independently. The chord engine stays active until the last arp finishes its fade.

**Random Sequence Lock (RND SEQ):** The lock state is shared between both arpeggiators, but the visual indicator (red border) only appears on displays whose arp is currently in a random mode (Random, Tonal Random, or Atonal Random). Clicking a display to toggle lock only works if that display's arp is in a random mode. The lock button (RND SEQ LOCK) in the HarmPadre editor is always visible (dimmed when neither arp is in a random mode).

**RND MODE:** Each arpeggiator has an independent RND MODE control (RND1 MODE in blue, RND2 MODE in yellow) that applies a musical contour to the random sequence. This allows combining random modes with contours such as Up, I&O, Converge, etc. The controls only enable when their corresponding arpeggiator is in a random mode. See the “RND MODE (Random Contour)” section for full details.

**Source menu:** Select “**Arp 2**” in any oscillator’s Source menu to route that oscillator to the second arpeggiator. Arp 2 appears as its own category in the source menu with its own derived sources: **Fugue 2**, **Retrograde 2**, **Polytonal 2**, **Harmonics 2**, and **Neg. Harmony 2**. Each derived source operates on Arp 2’s note, sequence, and history (not Arp 1’s). Per-oscillator parameters (delay, interval, harmonics count, etc.) are shared between Arp 1 and Arp 2 derived sources. Example setup: - OSC1 → Arp 1 (bass line, 1/8 notes, Up mode) - OSC2 → Arp 2 Fugue (delayed canon of Arp 2’s melody) - OSC3 → Arp 2 (melody, 1/16 notes, Random mode) - OSC4 → MIDI (sustained pad, unaffected by either arp)

**Colors:** Arp 1 uses **blue** ( #6CB4FF ), Arp 2 uses **yellow** ( #FFCC44 ) throughout the UI for visual distinction.

### Step Fader Grid (32 Steps)

Four rows of per-step controls for detailed arpeggio programming:

Row	Range	Default	Description
<b>TRANS</b>	-24 to +24	0	Per-step pitch transposition (semitones)
<b>VELO</b>	0-127	100	Per-step velocity
<b>GATE</b>	-1 or 0-100%	-1 (global)	Per-step gate override (-1 = use global)
<b>PROB</b>	0-100%	100	Per-step probability

- Steps are grouped visually in blocks of 8
- Active steps are highlighted; inactive steps are dimmed
- **Scroll wheel:** Adjust values by scrolling (when enabled in Settings > Interface)
- **Double-click:** Reset to default value (TRANS=0, VELO=100, GATE=-1, PROB=100)
- **Drag:** Click and drag vertically to adjust values

### Step Context Menu (Ratchet, Slide, Hold)

Click on a **step number** (below the fader grid) to open the **Step Context Menu**, which provides three per-step articulation controls: Ratchet, Slide, and Hold. The menu appears as a dark popup matching Portador’s UI style.

### Ratchet (per-step 1-4x repeat)

Ratcheting subdivides a step into rapid-fire note repetitions, essential for Berlin school sequences and techno patterns.

Option	Description
<b>1x</b>	Normal playback (no repeat)
<b>2x</b>	Note repeats 2 times within the step
<b>3x</b>	Note repeats 3 times within the step
<b>4x</b>	Note repeats 4 times within the step

- Ratcheted steps display in **orange** with the repeat count (e.g., “2x”, “3x”, “4x”)
- Each repeat is equally spaced within the step duration
- Setting a ratchet value clears any Hold on the same step (mutually exclusive)

### Pitch Slide (TB-303 Style)

Per-step slide creates smooth pitch glides between consecutive notes, recreating the classic TB-303 acid bassline effect.

State	Description
<b>Slide OFF</b>	Normal note transition
<b>Slide ON</b>	Ties into next note with ~60ms pitch glide (ease-out curve)

- When Slide is ON, the gate is forced to 100% (tied to the next note)
- Slide-enabled steps display **“S” in green**
- Tight noteOff/noteOn at the same sample position (no gap). For portamento between slid notes, raise Glide.

### Hold (sustained across steps)

Hold ties a note across multiple consecutive steps, creating sustained legato phrases. The gate-off only fires at the end of the hold period.

Option	Description
<b>H2</b>	Note sustains across 2 steps
<b>H4</b>	Note sustains across 4 steps
<b>H8</b>	Note sustains across 8 steps
<b>H16</b>	Note sustains across 16 steps

- Hold steps display in **cyan** with the hold duration (e.g., “H2”, “H4”, “H8”, “H16”)
- During hold, no new noteOn is emitted for the held steps – the original note continues
- The noteOff fires at the END of the hold period
- Setting a hold value clears any Ratchet on the same step (mutually exclusive)

**Tip:** Combine Ratchet with short gates for machine-gun fills, Slide for acid-style melodic movement, and Hold for sustained melodic notes within an arpeggio pattern. All three articulations are visible on the step number row at a glance.

### LENGTH Slider

Horizontal slider at the bottom controlling the number of active arpeggio steps (1-32).

## Polymetric Step Counts

Each fader lane (Transpose, Velocity, Gate) can have its own independent step count, cycling at its own rate. This creates polyrhythmic and polymetric patterns where different aspects of the arpeggio evolve at different cycle lengths.

### How it works

- By default, all lanes follow the global LENGTH value and wrap together
- Each lane can be set to its own count (1-32), causing it to cycle independently of the others
- The playback indicator wraps per-row independently, so you can see each lane cycling at its own rate

### Accessing per-lane step counts

- Small step count displays appear in the **left margin** of the HarmPadre editor, below the STEP TRANS / STEP VELOC. / STEP GATE labels
- **Click + drag vertically** on a lane count to change its value
- **Right-click** or **double-click** to reset the lane to follow global LENGTH

### Visual feedback

Color	Meaning
White	Lane follows the global LENGTH
Cyan	Lane has an independent step count

### Interaction rules

- When a lane count is set higher than the current global LENGTH, the global LENGTH auto-increases to match
- When the global LENGTH decreases, all lane counts are capped to the new maximum
- Lanes that were “following global” get frozen to their current value when another lane bumps the global LENGTH

**Example:** Set Transpose to cycle every 5 steps, Velocity every 7 steps, and Gate every 3 steps with a global LENGTH of 16. The three dimensions will phase against each other, creating patterns that take many cycles to repeat (LCM of 5, 7, and 3 = 105 cycles before the exact pattern repeats).

### Piano Roll

A code-painted keyboard display (C1-C7, 73 notes) at the bottom of the editor. Highlights currently active chord notes in real time, updating at 30Hz.

### Randomizer

The **Harm Random** button generates musically coherent random variations: - Randomizes chord types, per-degree settings, arp mode, step values - Scale-aware: stays within the current scale context - Moderate parameter ranges to ensure musical results - Updates LENGTH display to match the randomized step count

### Bidirectional Sync

All changes in the HarmPadre editor sync bidirectionally with the main UI: - Chord type changes update the main UI chord degree buttons - Step fader changes reflect in both directions at 30Hz - Scale, root,

and preset changes sync instantly via APVTS parameter listeners - Evolve offsets update in real time during playback

## ArpPatternDisplay (Main UI)

The **ArpPatternDisplay** is a visual indicator in the main Chord Engine section that shows the current arp mode's pitch contour at a glance. It is located below the degree buttons in the main chord section.

### Display modes

- **Standard modes** (Up, Down, UpAndDown, etc.): Shows a pitch contour line with horizontal dashes representing relative note positions, giving an immediate visual impression of the melodic shape
- **Chord mode**: Displays a vertical cluster of dots, indicating that all notes are played simultaneously
- **Euclidean mode**: Shows dots arranged on a pentagon/circle shape, representing the Euclidean rhythm distribution
- **Additive mode**: Shows a grow/shrink contour representing the expanding and contracting pattern

**Arp 1 / Arp 2 Toggle (Main UI only):** Small **"1"** and **"2"** **tab buttons** in the top-right corner of the display allow switching between Arp 1 and Arp 2. When Arp 2 is selected, the display color changes to **yellow** and the main arp controls (MODE, VELOCITY, RATE, RANGE) switch to show Arp 2 parameters. Shared controls are unaffected.

**HarmPadre Editor:** Both Arp 1 and Arp 2 displays are visible simultaneously — Arp 1 on the left (blue) and Arp 2 on the right (yellow). Each shows its own mode pattern independently.

**Random Sequence Lock:** In random modes (Random, Tonal Random, Atonal Random), clicking a display toggles the **Random Sequence Lock**. The red border only appears on displays whose arp is in a random mode — if Arp 1 is Random and Arp 2 is Down, only the Arp 1 display shows the red border and accepts lock clicks. The lock state itself is shared between both arps. The RND SEQ LOCK button in the HarmPadre editor toggles the lock for both and is visible when either arp is in a random mode.

**Sync:** The ArpPatternDisplay syncs bidirectionally with the HarmPadre editor – changing the arp mode in either location updates the display immediately. Lock state is synced across all displays (main UI + both HarmPadre displays).

# MIDI Controllers

Portador responds to standard MIDI controller messages from your keyboard or DAW:

Controller	Function	Range
<b>Pitch Bend</b>	Bends all oscillator pitches	$\pm 2$ semitones
<b>Mod Wheel (CC1)</b>	Assignable via Macro MIDI Source	0-127
<b>Channel Aftertouch</b>	Assignable via Macro MIDI Source	0-127
<b>Sustain Pedal (CC64)</b>	Holds notes in POLY mode	On/Off

## Pitch Bend

The pitch bend wheel directly bends all 5 oscillators by  $\pm 2$  semitones (configurable in Settings → MIDI → PB Range).

## Mod Wheel & Aftertouch

These are not hardwired to any parameter. Instead, you can assign them to any of the 5 Macros via right-click on a Macro fader. This gives you full flexibility — for example, assign Mod Wheel to Macro 1, then assign Macro 1 to Filter Cutoff, LFO depth, or any combination of parameters.

## Sustain Pedal

In POLY mode, holding the sustain pedal keeps notes sounding even after you release the keys. When the pedal is released, all sustained notes enter their release phase.

# Macros

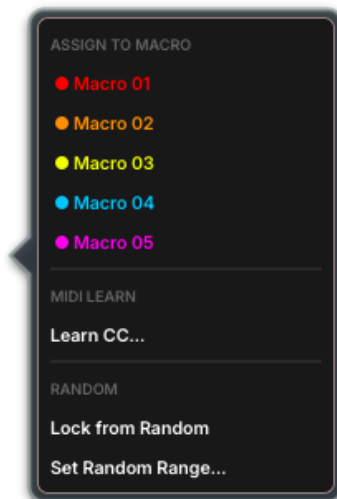


Figure 28: Macro assignment menu — right-click a Macro fader to assign destinations, set range, invert and configure curve.

Portador includes **5 Macro controls** that allow you to control multiple parameters simultaneously with a single fader. Macros are located horizontally below the modulation matrix.

## Macro Colors

Macro	Color
Macro 1	Red
Macro 2	Orange
Macro 3	Yellow
Macro 4	Cyan
Macro 5	Magenta

## Assigning Parameters

To assign a parameter to a macro:

1. **Right-click** any fader in the synthesizer
2. Select **“Add to Macro 01”** ... **“Add to Macro 05”** from the context menu
3. The fader will show a small **color dot** at the top indicating which macro it’s assigned to

A parameter can be assigned to multiple macros, and a macro can control multiple parameters simultaneously.

## Assignment Settings

Each parameter-to-macro assignment has individual settings:

Parameter	Range	Description
Min	0-100%	Minimum value of the control range
Max	0-100%	Maximum value of the control range
Direction	Normal, Inverted	Normal = rises with macro, Inverted = falls with macro
Curve	Linear, Exp, Log, S-Curve	Macro response curve

### Direction

- **Normal:** When the macro goes up, the parameter goes up (within its Min-Max range)
- **Inverted:** When the macro goes up, the parameter goes down

### Curves

Curve	Description
<b>Linear</b>	Direct linear response
<b>Exponential</b>	Fast change at start, slow at end
<b>Logarithmic</b>	Slow change at start, fast at end
<b>S-Curve</b>	Smooth transition with acceleration in the middle

### Usage

- **Drag** the macro fader to change its value (0-100%)
- **Click** on the color dot of an assigned fader to see the list of parameters assigned to that macro
- Assigned faders move automatically when you move the macro
- Macros are saved and restored with presets
- **Right-click** on a macro fader to assign a **MIDI Source**: None, Mod Wheel, or Aftertouch
- When a MIDI source is assigned, the macro value is driven by the MIDI controller instead of the fader
- A small **MW** (Mod Wheel) or **AT** (Aftertouch) label appears above the macro name when active
- MIDI source assignments are saved with presets

### Practical example

- Assign Filter1 Cutoff to Macro 1 with range 20%-80%
- Assign Filter2 Cutoff to Macro 1 with Inverted direction
- Now a single macro fader opens one filter while closing the other

### LFO Modulation via Macros

Macros can be modulated at audio rate via the modulation matrix. Route an LFO or other source to a Macro destination (Macro 1-5) in the matrix, and all parameters assigned to that macro will be modulated smoothly per-sample. This allows complex multi-parameter modulation from a single matrix connection. You can combine direct fader control with CV modulation — the macro value is the sum of the fader position and the CV signal.

---

# MIDI Learn

Portador includes a global **MIDI Learn** system that allows you to assign any MIDI CC controller to any synthesizer parameter.

## How to Use MIDI Learn

1. Click the **LEARN** button in the header (it will turn red)
2. Move any fader or parameter in the synthesizer that you want to control
3. Move the knob or fader on your MIDI controller
4. The MIDI CC is automatically mapped to the parameter — the LEARN button deactivates

## Features

- **Global:** MIDI Learn mappings are not per-preset — they persist regardless of the loaded preset
- **One CC per parameter:** Each parameter can have only one CC assigned, and each CC controls only one parameter
- **Min/Max Range:** Each mapping has a configurable range with minimum (0.0) and maximum (1.0) defaults
- **Persistence:** Mappings are saved and restored with the plugin state

## Mapping Management

- When creating a new mapping, any previous mapping for the same CC or the same parameter is automatically removed
- Mappings are stored as XML within the plugin state

**Tip:** MIDI Learn is ideal for mapping physical knobs on your controller to frequently used parameters like filter cutoff, oscillator level, or LFO depth.

# Modulation Matrix



Figure 29: Modulation Matrix — 16x16 grid. Rows are modulation sources (Osc 1-5, Noise, Gate Gen, Filters, Envs, LFOs, Logic). Columns are destinations (any modulatable parameter via right-click, plus fixed audio routes for filter/channel I/O).

The 16x16 matrix is the heart of Portador’s architecture. It allows connecting any modulation source to any destination parameter. Each intersection cell controls the modulation amount (0-100%) from the source (row) to the destination (column).

## Sources (Rows)

The matrix has 16 rows divided into audio sources (rows 1-9) and control sources (rows 10-16):

### Audio Sources

- **Row 1-3: Osc 1, Osc 2, Osc 3** — Audio output from oscillators 1-3. These rows include a **waveform selector** dropdown: click the row label to choose between Sine, Saw, Triangle, Square, and Pulse without leaving the matrix.
- **Row 4-5: Osc 4, Osc 5** — Audio output from wavetable oscillators.
- **Row 6: Noise** — Noise generator output.
- **Row 7: Gate Gen** — Gate Generator CV output (binary toggle). Cell color indicates assigned track: orange (T1), yellow (T2), pink (T3), red (T4), purple (T5). Right-click to assign track.
- **Row 8-9: Filter 1, Filter 2** — Filter output signals.

### Control Sources (CV)

- **Row 10: Env 1** — Envelope 1 (per-voice in POLY mode).
- **Row 11: Env 2** — Envelope 2 (per-voice in POLY mode).
- **Row 12: Logic** — Logic module output (per-voice in VENV mode, global otherwise).

- **Row 13: LFO 1** — LFO 1 output.
- **Row 14: LFO 2** — LFO 2 output.
- **Row 15: LFO 3** — LFO 3 output.
- **Row 16: LFO 4** — LFO 4 output.

**Note:** the former dedicated “LFO 1 Inv” and “Logic Inv” rows were removed in the LFO Quad refactor (2026-04-15). Inversion is now managed per-cell — see “Per-Cell Invert” below.

## Destinations (Columns)

**Columns 1-10** are freely assignable parameter destinations. Click the column header to open a hierarchical dropdown menu organized by module. You can assign any of the following parameters:

**Oscillators (Osc 1-5):** Freq, Shape, FB, Pan, Level. The FB slot modulates `osc{N}_feedback` in Classic engine (DX7 PM self-mod) or `osc{N}_warp` in non-Classic engines (Motion / Bright / Sym / Detune / Res / Tone depending on which engine is loaded — the label in the matrix column header tracks the main-UI fader). Osc 4 and 5 also include: FM/PM Index, AM/RM Index, FM/PM Audio In, AM/RM Audio In, WT Position. Labels update dynamically based on FM/PM and AM/RM mode.

**Filters:** Filter 1 and Filter 2, each with: A Freq, A Res, B Freq, B Res, Keytrack, Level.

**LFOs:** LFO 1-4: Speed, Level, Reset, Release. **Reset** accepts rising-edge CV to restart the LFO phase. **Release** (0-10s) is the exponential decay, only active in Poly mode. Modulating Release with another LFO creates modulation tails that stretch and shrink dynamically.

**Envelopes:** Env 1: Attack, Decay, Sustain, Release, Gate In. Env 2: Attack, Release, Gate In.

**Gate Gen:** Time, Hold, Pulses.

**Noise:** Colour, Level, Filter.

**Logic:** Parameters change based on mode (VENV: A Lev, D Lev, S Lev, R Lev, Time, Gate In. Function: Rise, Fall, Curve, Smooth, Level, Gate In. Random/S&H: Rate, Spread, Curve, Smooth, Level, Gate In).

**Channels:** CH1 and CH2: Pan, FX1 Send, FX2 Send.

**Macros:** Macro 1, Macro 2, Macro 3, Macro 4, Macro 5.

**Chord/Arp:** Arp Rate, Gate, Evolve, Swing, Range, Length, H Gate In, Harm Root, Scale Root.

**Scale Root** (Chord submenu): Modulates the scale root note from the matrix. CV value of  $\pm 1.0$  transposes through all 12 semitones. Useful for automated key changes synced to LFOs or envelopes.

**Spread (per oscillator):** Order, Low Pan, Hi Pan, Lower Freq, Upper Freq. These appear when the oscillator’s Spread mode is active.

**Sampler/Granular (Osc 4/5):** Speed, Position, Size, Density, Scan, Pitch, Direction, Random Position, Random Pitch, Random Pan, Rnd Tilt. These appear when Osc 4 or Osc 5 is in Sampler or Granular engine mode.

**Drum (all Osc):** When an oscillator is in Drum engine mode, the matrix column labels change dynamically: Shape becomes **Decay** and Warp becomes **Tone**. The available parameters are: Freq, Decay, Tone, Level (+ Spread if active).

**Columns 11-16** are fixed audio/control routing destinations:

- **Col 11-12: Filter 1 Audio In, Filter 2 Audio In** — Route audio sources into the filters.
- **Col 13-14: CH1 Ctrl, CH2 Ctrl** — VCA control for per-voice amplitude (connect envelopes here).
- **Col 15-16: CH1 Audio In, CH2 Audio In** — Route audio sources to the output channels.

## Dynamic Modules

Some destination modules appear conditionally in the column menu:

- **Distortion CH1/CH2:** Only visible when the channel's distortion is enabled.
- **EQ CH1/CH2:** Only visible when the channel's EQ is enabled (12 parameters: 4 bands x Freq, Gain, Q).
- **FX modules** (Delay, Delay Tap, Granular, Resonator, Shimmer, Random Gate, Reverb): Only visible when loaded in an FX slot.

## Per-Cell Invert

**Right-click** any active cell to toggle inversion. An inverted cell shows a **red border** and a  $\pm$  glyph over it. Inversion is applied to the source value before scaling by the cell level:

- **Bipolar sources** (LFO1-4, bipolar Env2, Logic in Function/Random modes): inversion returns  $-s$ , mirroring the wave around zero.
- **Unipolar sources** (Env1, unipolar Env2, Gate Gen seen as CV, Noise): inversion returns  $1 - s$  (complement), e.g. turning an attack into a release.

Row 7 (Gate Gen) does not support invert since it is a binary toggle that other rows can read as audio or CV depending on destination.

This system replaces the former dedicated "LFO 1 Inv" and "Logic Inv" rows: you can now invert any source at any destination independently, without consuming full matrix rows.

## Using the Matrix

1. **Click a column header** to open the destination selector and choose which parameter to modulate
2. Locate the source row you want to use as modulator
3. **Click the intersection cell** and drag up/down to adjust the modulation amount (0-100%)
4. **Right-click** a cell to invert the modulation (red border +  $\pm$  glyph)
5. Active connections are highlighted with a colored dot in the cell

**Tip:** The matrix can route over 360 parameters. Use it to build complex modulation chains — for example, route LFO1 to Filter Cutoff, Env2 to oscillator FM, and LFO2 to LFO1's speed (and its Release, and its Reset), all simultaneously.

# Random System

Portador includes three randomization buttons for different aspects of the sound:

## Gen Random (Cyan)

Randomizes generative parameters: - Oscillator waveforms - Tuning and shape - Pan and levels - Note sources - Noise parameters - Gate Generator

## Mod Random (Purple)

Randomizes modulators: - Envelope times - LFO speeds and shapes - Filter cutoffs and resonances - Granular parameters - Frequency shifter parameters

## Harm Random (Orange)

Randomizes harmony: - Chord types per degree - Voicings and inversions - Arpeggio modes - Step velocities - Step transpositions

## Gate Random (Yellow)

Randomizes gate generator parameters when enabled.

## Lock from Random (Per-Fader)

Right-click any fader to access the context menu. Under the **RANDOM** section:

- **Lock from Random:** Protects this specific parameter from all randomization buttons. A locked parameter will be skipped by Gen Random, Mod Random, Harm Random, and Gate Random. Toggle on/off.
- **Set Random Range...:** Opens a popup panel where you can set the **MIN** and **MAX** bounds for randomization. When a random range is set, randomization will only generate values within that range. The range is shown as visual indicators on the fader.
- **Remove Random Range:** Removes the custom range (only shown when a range is set).

Random ranges and locks are saved and restored with presets.

## Module Lock (Per-Module)

Right-click on a module name label (e.g., “OSC 1”, “FILTER 1”, “LFO 1”) to lock the entire module from randomization. When a module is locked, all its parameters are excluded from randomization — equivalent to locking every fader in that module individually.

**Available modules:** OSC 1-5, FILTER 1-2, NOISE, GATE GEN, LFO 1-2, ENV 1-2, LOGIC, FX 1-2.

Module locks are session preferences — they are preserved during the session but not saved in presets, so your lock configuration persists as you browse presets.

---

## Chord Parameter Context Menus

Right-click context menus are available on the following chord parameter controls in both the main UI and HarmPadre editor: **Notes, Voicing, Finger, Octave, Human, Scale Root, ARP1 Notes, and ARP2 Notes**. Each menu offers: - **Assign to Macro (1-5)** - **MIDI Learn** - **Lock from Random** - **Set Random Range**

# Presets

## Preset Browser

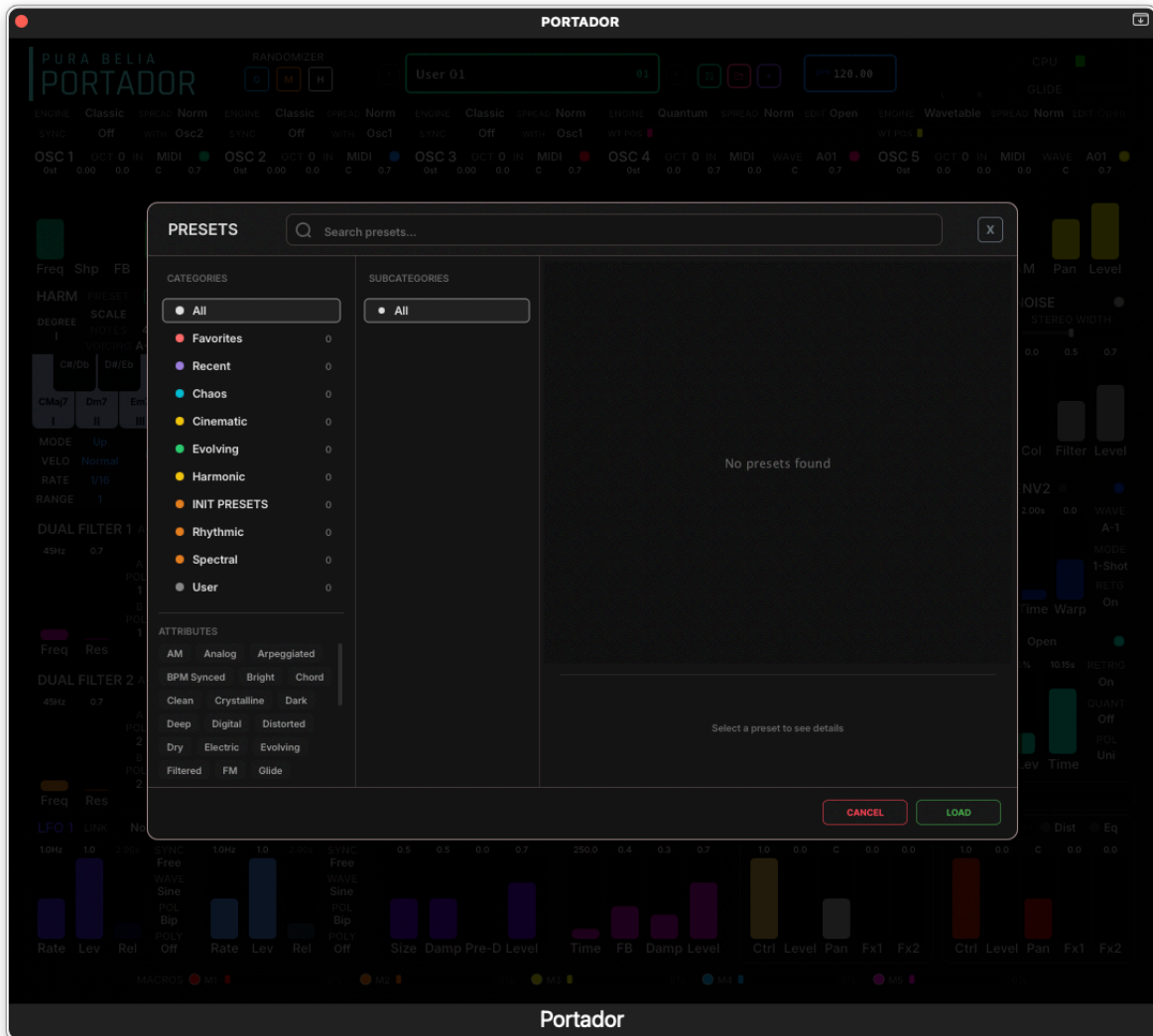


Figure 30: Preset Browser — categorized preset list with search, tags, favorites, author and audition controls.

The full-screen Preset Browser provides professional preset management:

### Opening the Browser

- Click the “PRESETS” button in the header

### Browser Layout

- **Left Sidebar:** Categories and Attributes filters
- **Center:** Preset list with favorites
- **Bottom:** Selected preset details and Init buttons

**Dynamic Category Scanning:** The category list is scanned at startup from the Factory and User preset folders. Create new sub-folders under `~/Documents/Portador Presets/User/` (e.g. `User/MyCustomCategory/`) and they will appear automatically in the browser and in the Save dialog dropdown after restart. Unknown category names get a deterministic color derived from the folder name.

## Default Categories

Category	Description
Arp	Arpeggiated sequences
Bass	Bass sounds
Drums	Drum and percussion
Effects	Sound effects
Keys	Keyboard sounds
Noisescapes	Noise-based atmospheres
Pads	Atmospheric pads
Percussive	Percussive hits
Sequence	Sequenced patterns
Synth Long	Long sustained synths
Synth Mono	Monophonic leads
Synth Poly	Polyphonic synths
Synth Hit	Short synth stabs
Textures	Textural sounds
User Presets	Your saved presets

## Attributes (Tags)

Filter presets by sonic characteristics: AM, Analog, Arpeggiated, BPM Synced, Bright, Chord, Clean, Dark, Deep, Digital, Distorted, Dry, Electric, Evolving, Filtered, FM, Glide, Granular, Hit, Long Release, Melodic, Metallic, Noisy, Monophonic, Percussive, Sequenced, Slow, Stab, Sub, Wet

## Actions

- **Single click:** Select and preview preset details
- **Double click:** Load preset immediately
- **Heart icon:** Toggle favorite
- **Search bar:** Filter by name
- **Init buttons:** Load category-specific init presets

## Saving Presets

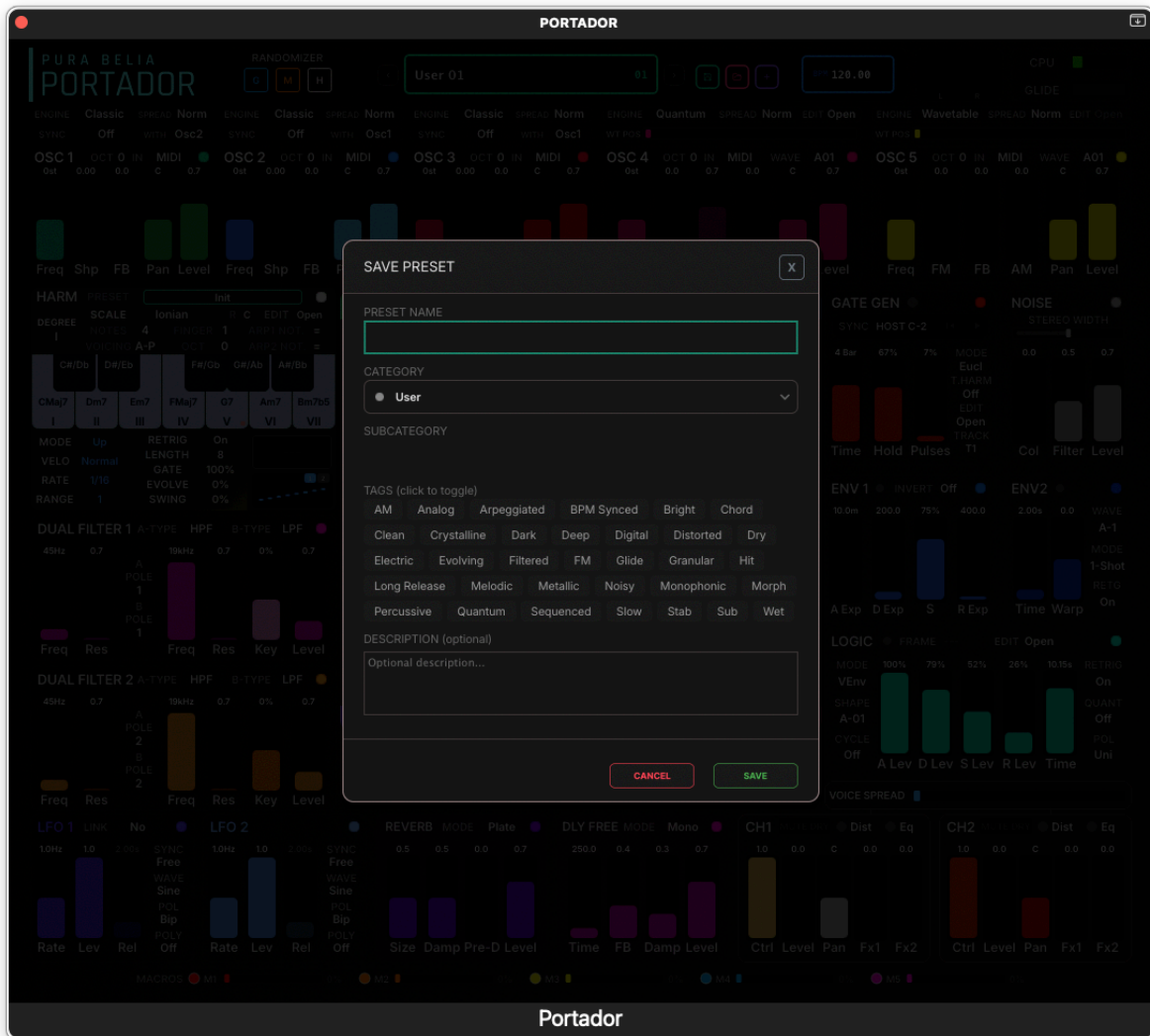


Figure 31: Save Preset dialog — name, author, category, tags and bank selection before writing the preset to disk.

### Save Button

- Overwrites current preset if not a factory preset
- Opens Save dialog for new presets

### Save As Dialog

1. Enter **Preset Name**
2. Select **Category** from dropdown
3. Toggle **Tags** to describe the sound
4. Add optional **Description**
5. Click **SAVE**

### Overwrite Confirmation

When saving with a name that already exists, you'll be asked to confirm before overwriting.

## Quick Navigation

Use the arrow buttons (◀ ▶) to navigate between presets in the current category.

## Preset Contents

Each Portador preset is a JSON file that saves the full state of the synthesizer. Beyond the standard audio parameters, presets also persist:

- **EQ Morph snapshots** — the A/B/C/D morph bank of each channel EQ (frequencies, gains, Q values, band types, and current snapshot state)
- **Matrix column destinations** — which parameter each dynamic matrix column (1-10) is routed to
- **Synth and Harm MIDI channels** (Settings → MIDI)
- **Microtuning selection**
- **User wavetable paths**
- **MIDI Learn mappings**
- **Random rule ranges** (Gen/Mod/Harm/Gate)
- **Gate Gen per-step chord assignments and skip counts**

When loading a preset, all of this preset-owned state is reset to defaults before the preset values are applied, so nothing leaks between presets — the preset you load is exactly what you get, regardless of previous edits or previously-loaded presets.

## Chord+Arp Presets

In addition to the main preset system, Portador includes an independent preset system specifically for the Chord Engine and Arpeggiator. This allows saving and loading chord/arp configurations independently from the synthesizer sound.

### File locations

- **Factory:** Portador Presets/ChordArp/Factory/
- **User:** Portador Presets/ChordArp/User/

## Factory categories (13)

Category	Description
<b>Ambient</b>	Slow evolving pads, atmospheric arpeggios, gentle textures
<b>Avant-Garde</b>	Schoenberg, Berg, Bartok, Messiaen, Stravinsky, Ligeti, Webern – atonal clusters, polytonal layers, serial-inspired patterns
<b>ChordAll</b>	Showcases Chord All combined with Fugue, Retrograde, NegHarmony, Polytonal
<b>Cinematic</b>	Film-inspired progressions: Interstellar, Blade Runner, Satie, Debussy, and more
<b>Classical</b>	Bach, Mozart, Beethoven, Chopin, Debussy – counterpoint canons, Alberti patterns, voice-led progressions
<b>Dark</b>	Diminished scales, tritone axes, atonal random, noise gates
<b>Diatonic</b>	All 7 modes (Ionian through Locrian) plus harmonic/melodic minor variations
<b>Film Scores</b>	Desplat, Zimmer, Greenwood, Richter, Johannsson, Reznor, Santaolalla
<b>Jazz</b>	Coltrane Changes, ii-V-I rootless, bebop, bossa nova, Monk clusters, quartal jazz
<b>Modal</b>	Exotic scales: Phrygian Dominant, Enigmatic, Hungarian, Persian, Prometheus, Whole Tone
<b>Neo-Soul</b>	D'Angelo, Erykah Badu, Robert Glasper, Thundercat – lush extended chords with swing
<b>Rhythmic</b>	Euclidean patterns, polymetric sequences, TB-303 acid, ratchet machines, step melodies
<b>Symmetric</b>	Diminished, augmented, whole tone, chromatic cluster patterns

## Data included in a Chord+Arp preset

- Scale and root note
- Configuration for all 12 degrees (chord type, voicing, etc.)
- Arpeggio mode, range, rate, steps, retrigger
- Velocity mode, time signature, humanization
- 5th chord extension, evolve, swing
- Global gate, gate release
- Oscillator sources (including Chord All)
- Step Velocities (32 steps)
- Step Transpose (32 steps)
- Step Probabilities (32 steps)
- Step Gates (32 steps)
- Step Ratchets, Slides, Holds (32 steps each)
- Free chord mode with 12-slot chord progression
- Fugue, Retrograde, Polytonal configuration

---

### **Saving Chord+Arp Presets**

1. Configure the Chord Engine and Arpeggiator to your liking
2. Click the **Save** button in the Chord+Arp preset section
3. Enter a name in the dialog and click **Save**
4. The preset is saved to `Portador Presets/ChordArp/User/`

### **Loading Chord+Arp Presets**

- Click the preset name to open the browser
- Factory presets and User presets are listed separately
- Click a preset name to load it immediately

**Tip:** Chord+Arp presets are ideal for reusing chord progressions and rhythmic patterns with different synthesizer sounds.

# Settings



Figure 32: Settings panel — UI size, interface, oversampling, voices, glide, tuning, analog drift, MIDI and presets configuration.

Access the Settings panel by clicking the gear icon in the header.

## UI Size (Display)

Adjust the plugin window scale:

Option	Description
75%	Compact size
90%	Small
100% (1:1)	Default
115%	Slightly larger
130%	Large
150%	Extra large
175%	Very large
200% (2:1)	Double size

## Interface

UI interaction settings.

**Scroll Wheel:** Toggle to enable or disable mouse wheel control on all faders and sliders globally. When OFF (default), faders and sliders only respond to click and drag. When ON, you can also adjust values using the mouse scroll wheel. This applies to all faders in the main UI, editor windows (Sampler, Granular, VENV, HarmPadre), and per-step arp controls (transpose, velocity, gate, probability).

**Scope Auto-Focus:** When enabled (OFF by default), the spectrum analyzer (Scope) automatically switches its display source to whichever filter or envelope you are directly adjusting. When you drag a Filter 1 or Filter 2 fader (cutoff, resonance, level), or an envelope fader (ENV1, ENV2, VENV), the scope temporarily switches to show that module's response. After ~2.5 seconds of inactivity, the scope automatically returns to the previously selected source. Auto-focus only activates on direct fader interaction — it does not trigger when parameters are changed via macros, LFO modulation, or other indirect sources.

State	Behavior
<b>OFF</b> (default)	Scope always shows the manually selected source
<b>ON</b>	Scope automatically follows the filter you're editing

**Tip:** This feature is very useful for sound design, as it lets you see the effect of your filter adjustments in real time without having to manually switch the scope source.

## Oversampling

Improves audio quality by reducing aliasing at the cost of CPU:

Option	Description
Off	No oversampling (lightest CPU)
2x	Double sample rate
4x	Quadruple sample rate (best quality)

**Scope:** Oversampling affects: - **Osc4 and Osc5:** FM/AM synthesis (external oversampling with anti-aliasing filter) - **Osc1, Osc2, and Osc3:** Non-Classic engines (Cloud, Wavefolder, Supersaw, PhaseDist, NoisePitch) — internal oversampling via sub-sample averaging

**Note:** The Classic engine in Osc1-3 uses PolyBLEP for anti-aliasing and doesn't need oversampling. The Resonator engine also doesn't need it as it's band-limited by the comb filter.

## Voices

Select the synthesizer's polyphony:

Option	Description
<b>POLY 8</b>	8 polyphonic voices - light CPU
<b>POLY 16</b>	16 polyphonic voices - balanced
<b>POLY 32</b>	32 polyphonic voices - high polyphony
<b>POLY 64</b>	64 polyphonic voices - maximum polyphony (default)

For monophonic behavior, see **MOSC (Mono per oscillator)** below — instead of a global mono mode, each oscillator can be individually toggled to mono via right-click on its OSC label. This lets you mix mono and poly oscillators in the same patch (e.g. mono lead on Osc1 + poly chord on Osc2-5).

**Tip:** Glide is global and works in any combination of MOSC and poly oscillators. Set Glide > 0 for portamento between notes.

## Glide Type

Select the portamento behavior between notes:

Type	Description
<b>Linear</b>	Uniform, constant pitch transition
<b>Exponential</b>	Natural transition, faster at the start
<b>Glissando</b>	Chromatic semitone steps
<b>Scale</b>	Transition following active scale notes

## Tuning

Microtuning system with 32 scales organized in 7 categories. Explore tonal systems from different cultures and eras.

## Available categories and scales

Category	Scales
<b>Western</b>	Equal Temperament 12-TET (default)
<b>Historical</b>	Just Intonation, Pythagorean, Meantone Quarter-Comma, Kirnberger III, Werkmeister III, Silbermann
<b>Arabic/Turkish</b>	Al-Farabi 22-tone, Sabbagh 7-tone, Turkish 24-tone
<b>African</b>	Amadinda 240-cent, Balafon 7-Tone, Charimba, Sanza 7-Tone, Matepe
<b>Gamelan</b>	Pelog 7-tone, Slendro 5-tone, Degung 5-tone
<b>Microtonal</b>	11-TET, 13-TET, 17-TET, 19-TET, 24-TET, 31-TET, 48-TET
<b>Modern</b>	Bohlen-Pierce 13, Carlos Alpha, Carlos Beta, Carlos Gamma, Carlos Harmonic, Partch 43, Carlos Super Just

### Usage

- Click a category to expand it and see its scales
- Select a scale to apply it to all oscillators
- Tuning is saved and restored with each preset
- Equal Temperament 12-TET is the standard Western tuning

**Note:** Microtuning scales affect all 5 oscillators simultaneously, including the chord generator and arpeggiator.

### Analog Drift

Adds subtle per-voice parameter variations on each note-on, simulating the behavior of analog synthesizers where no two notes sound exactly alike.

Parameter	Off	Subtle	Moderate	Heavy
<b>PITCH</b>	0 cents	±3 cents	±7 cents	±15 cents
<b>FILTER</b>	0%	±2%	±5%	±10%
<b>SHAPE</b>	0%	±1%	±3%	±6%
<b>LEVEL</b>	0 dB	±0.3 dB	±0.7 dB	±1.5 dB

Each voice receives unique random offsets when a note is triggered, creating natural variation across polyphonic chords. The offsets are held for the duration of the note.

**Default values:** PITCH = Off, FILTER = Subtle, SHAPE = Subtle, LEVEL = Subtle.

Click each parameter to cycle through Off → Subtle → Moderate → Heavy. Drift settings are saved with presets.

## MIDI

Configure global MIDI input channels and pitch bend range. Portador routes MIDI on separate channels for the synth engines and the chord/arp engine, which allows flexible source combinations.

**Synth Ch:** MIDI channel for oscillators using `Source = MIDI`. When an oscillator is set to MIDI, it responds only to notes on this channel. Set to **ALL** to accept any channel.

**Harm Ch:** MIDI channel for the chord/arp engine. Chord recognition, arpeggiator triggers, and all chord/arp sources (Chord1-5, Chord All, Arp, Fugue, Retrograde, Polytonal, Harmonics, NegHarm) react only to notes on this channel. Set to **ALL** to accept any channel.

**Osc1-5 Ch:** Per-oscillator channel override. Each oscillator can listen to a specific MIDI channel independently of the main Synth Ch. Useful for multi-timbral setups where different oscillators respond to different keyboards or sequencer tracks.

Option	Description
<b>ALL</b>	Receive on all 16 channels
<b>1-16</b>	Receive only on the selected channel

**Chord/Arp + MIDI coexistence (channel split):** Normally when an oscillator's source is set to `MIDI`, it mutually excludes chord/arp sources on other oscillators — the UI prevents you from combining them on the same channel. However, when **Synth Ch** and **Harm Ch** are set to *different* specific channels (e.g. Synth=Ch1, Harm=Ch2), the exclusion is lifted and both source types can coexist. This lets you play a melody on the Synth channel (with MIDI-source oscillators) while the chord/arp engine is driven independently on the Harm channel (with Chord All / Arp sources). Same behavior as Gate Generator TRIG HARM.

**PB Range:** Set the global pitch bend range. Use the left/right arrows to select.

Option	Description
<b>±1 st</b>	±1 semitone
<b>±2 st</b>	±2 semitones (default)
<b>±5 st</b>	±5 semitones
<b>±7 st</b>	±7 semitones (perfect fifth)
<b>±12 st</b>	±1 octave
<b>±24 st</b>	±2 octaves
<b>±48 st</b>	±4 octaves

## Presets Section

- **Reset to Default:** Resets all parameters to their initial values. This includes all oscillator settings, filters, envelopes, effects, modulation matrix, chord/arp configuration, and macros. All UI components update immediately to reflect the default state. Any open editor windows (VENV, Sampler, Granular) are closed, and custom waveforms are cleared.
- **Show User Folder:** Opens the preset folder in Finder (macOS) or Explorer (Windows).
- **Set Preset Folder...:** Opens a folder picker to choose a custom location for all presets, samples, wavetables, and user data. Useful if you want to store everything on an external drive or a shared location. The setting is saved in `~/Library/Application Support/Portador/settings.json` and persists across sessions.

- **Reset Preset Folder:** Only visible when a custom folder is set. Resets the preset folder to the default location ( `~/Documents/Portador Presets/` ). Note: existing files are not moved automatically — you should move them manually if needed.

The current preset folder path is displayed at the bottom of this section.

## Preset Folder Structure

By default, all data is stored under `~/Documents/Portador Presets/` :

Folder	Contents
<i>(root)</i>	Synth presets (.json)
<code>Samples/Factory/ + User/</code>	Factory and user samples
<code>Wavetables/Factory/ + User/</code>	Factory and user wavetables
<code>Venv/Factory/ + User/</code>	Factory and user VENV envelopes
<code>GateGen/Factory/ + User/</code>	Factory and user Gate Gen presets
<code>ChordArp/Factory/ + User/</code>	Factory and user Chord+Arp presets

Each subsystem follows the same Factory/User split. Factory content is generated on first launch; User content is created when you save presets. Adding files to any Factory subfolder makes them available in the corresponding menu on next load.

**Path Portability:** When saving DAW sessions or presets that reference samples or wavetables, Portador stores paths relative to the preset folder using the `$PRESETS/` prefix. This means if you move your preset folder to another location (or another computer), references will resolve correctly as long as the folder structure is preserved.

## About

Shows version information and credits.

---

# CPU Meter

In the upper right corner of the interface, next to the Glide fader, you'll find the CPU meter.

## Features

- Shows real-time audio processor load
- Horizontal bar indicating CPU usage percentage
- **Colors based on load:**
  - **Green** (0-50%): Low load - optimal operation
  - **Yellow** (~50%): Moderate load - attention
  - **Red** (50-100%): High load - possible audio glitches

## What affects CPU load

- Number of active voices
- Synthesis engines used (some are heavier)
- Oversampling enabled
- Granular processing
- Active modulations in matrix

**Tip:** If you notice the meter is frequently red, consider reducing the number of voices in Settings or disabling oversampling.

# Scope

The Scope is a real-time waveform and envelope visualizer located above the modulation matrix. It displays the output of oscillators and envelopes, helping you understand what each sound source is producing.

## Source Selection

Use the **SOURCE** dropdown button (left of the scope display) to select what to visualize:

Source	Description
<b>OSC 1</b>	Oscillator 1 waveform (green)
<b>OSC 2</b>	Oscillator 2 waveform (light blue)
<b>OSC 3</b>	Oscillator 3 waveform (red)
<b>OSC 4</b>	Oscillator 4 wavetable (magenta)
<b>OSC 5</b>	Oscillator 5 wavetable (yellow)
<b>GATE</b>	Gate Generator pattern (yellow/lime)
<b>F1 A</b>	Filter 1 High-Pass section output
<b>F1 B</b>	Filter 1 Low-Pass section output
<b>F1 A+B</b>	Filter 1 combined output (both sections)
<b>F2 A</b>	Filter 2 High-Pass section output
<b>F2 B</b>	Filter 2 Low-Pass section output
<b>F2 A+B</b>	Filter 2 combined output (both sections)
<b>LFO 1</b>	LFO 1 waveform (purple)
<b>LFO 2</b>	LFO 2 waveform (light blue)
<b>LFO 3</b>	LFO 3 waveform (dark green)
<b>LFO 4</b>	LFO 4 waveform (light green)
<b>LINK</b>	Unified LFO-link visualizer — scrolling overlay in Dual/Quad, 16-ball pendulum rig in Pend, OFFLINE in Off (see LFOs > Link Mode)
<b>ENV 1</b>	Envelope 1 ADSR curve (blue)
<b>ENV 2</b>	Envelope 2 vector shape (dark blue)
<b>VENV</b>	Logic VENV envelope (cyan)
<b>HARM</b>	Piano roll with chord / arp visualization (per-degree colours)
<b>STEREO</b>	Stereo field visualization (cyan-green)
<b>SPECTRUM</b>	Spectrum analyzer with peak hold (ice-blue)
<b>MIDI</b>	MIDI capture and drag-and-drop export (green)

## Display Modes

### OSC 1-3 (Classic Oscillators)

- Shows 2 cycles of the waveform
- Updates in real-time when you change Wave type or Shape parameter
- Displays the effect of wave folding, wave shaping, and pulse width modulation

### OSC 4-5 (Wavetable Oscillators)

- When playing: shows the **real audio output** including FM/PM, AM/RM, and feedback modulation effects
- When not playing: shows the current wavetable frame based on WT Position
- Updates in real-time to reflect all modulation including FM/PM In, AM/RM In, feedback, and engine modes

### GATE (Gate Generator)

- Shows the current gate pattern as rectangular pulses for the **selected track**
- When switching tracks in the main UI or GatePadre editor, the scope updates to show that track's pattern with its per-track parameters (time, hold, pulses, swing, mode)
- Euclidean mode: displays the Bresenham-distributed pulse pattern with hold width
- Random mode: shows a deterministic preview of the density/probability pattern
- Manual mode: displays a flat line until a pattern is recorded
- Updates in real-time when you adjust Time, Hold, Pulses, or Mode

### ENV 1 (ADSR Envelope)

- Shows the Attack, Decay, Sustain, and Release stages
- Displays the selected curve types (Linear, Exp, Log, S-Curve)
- Updates when you adjust any envelope parameter
- When **Mod** is connected in the matrix: switches to **real-time display** showing the actual modulated envelope output (e.g., LFO tremolo, rhythmic gating). The display auto-scales to fit any duration using progressive decimation

### ENV 2 (Vector Envelope)

- Shows the selected envelope shape from the bank
- Displays the warp effect applied to the shape
- When **Mod** is connected: switches to real-time modulated output display

### VENV (Logic Vector Envelope)

- Shows the current VENV waveform
- In **Unipolar** mode: displays as a standard envelope (0 to 1)
- In **Bipolar** mode: displays as an oscillator-style waveform (-1 to +1) with positive/negative fill
- When **Mod** is connected: switches to real-time modulated output display

### STEREO (Stereo Field)

- Displays a Lissajous XY visualization of the stereo output
- The horizontal axis represents the Mid signal (L+R), the vertical axis represents the Side signal (L-R)
- A centered mono signal appears as a vertical line; wide stereo content spreads horizontally

- Crosshair lines and L/R/M labels help orient the display
- The trace uses a connected line with a glow effect for visibility
- Continuously updates at the scope refresh rate

## MIDI (MIDI Capture & Drag-and-Drop)

The MIDI source turns Portador into a composition tool by letting you capture the MIDI output from the chord engine, arpeggiators, and evolve mutations, then drag it as a standard .mid file into your DAW or any other instrument.

### Normal view (scope area)

The normal scope view shows all controls in a compact toolbar: - **CAPTURE**: Freezes the current recording. Portador continuously records all chord and arp MIDI events in the background — you never need to “arm” or “start” recording. Just play, experiment, and when you like what you hear, hit CAPTURE. - **CLEAR**: Clears the current snapshot and resets the buffer. Next CAPTURE starts fresh. - **CHORD / ARP 1 / ARP 2 / ALL**: Drag zones. Click and drag from any zone to export that stream as a .mid file. Drop it directly into a Logic Pro X MIDI track (or any DAW that accepts MIDI files). - **ZOOM**: Opens the full piano roll display.

### Zoomed view (click ZOOM)

The zoomed MIDI view provides a full piano roll editor for reviewing captured notes:

- **Toolbar**: CAPTURE, CLEAR, duration (bars), BPM, and × close button. Always visible.
- **Timeline ruler**: Bar numbers (1, 2, 3...) and beat ticks, aligned with the note grid.
- **Piano keyboard**: Vertical keyboard on the left with note names on every C. White and black keys have different background shading (like Logic Pro).
- **Note display**: Captured notes shown as colored blocks on a time-vs-pitch grid.
  - **Velocity colors** (Logic Pro style): Purple (vel 1) → Blue (30) → Green (70) → Amber (100) → Red (127). Each note has a white velocity line inside — length proportional to velocity.
  - **Bar/beat grid**: Vertical lines aligned with the ruler. Brighter lines on bar boundaries, subtle lines on beats.
- **Navigation**: Click and drag in the note area:
  - **Drag left/right** → scroll horizontally through the recording
  - **Drag up** → zoom in (more detail)
  - **Drag down** → zoom out (see more)
  - When zoomed in, the duration label shows “visible/total bars” (e.g., “2.0/8.2 bars”).
- **Scroll wheel**: When enabled in Settings, scrolls horizontally through the recording.
- **Drag zones** (bottom strip): Same as normal view — CHORD, ARP 1, ARP 2, ALL.

### Smart capture — only captures what sounds

- Source = MIDI → captures the raw keyboard notes you play
- Source = Chord → captures the generated chord notes (including voice leading, inversions, etc.)
- Source = Arp → captures arpeggiator output with sample-accurate timing
- If no oscillator uses a source, that stream is not captured (no phantom notes)
- Humanization-aware: roll-up delays are preserved — each chord voice is captured at the exact moment it starts sounding

### Workflow example

1. Set up your chord progression, arpeggiator, and evolve settings
2. Select **MIDI** as the scope source

3. Play — all notes are captured in the background
4. Hit **CAPTURE** when you're happy with the performance
5. Drag **ARP 1** to a Logic MIDI track — done!
6. Click **ZOOM** to review the captured notes in the piano roll
7. Navigate with click-drag to zoom into specific sections
8. Use the exported MIDI in another Portador instance (Source = MIDI) or any other instrument

### Supported scenarios

- Works with DAW playing or stopped (uses internal clock when host is stopped)
- Works with Gate Generator in FREE mode + TRIG HARM
- Works in Standalone mode
- Captures velocities including humanization and timing offsets
- Mini-notes (< 10ms) are automatically filtered from the exported MIDI file

### Visual Features

- **Color coding:** Each source has a unique color for easy identification
- **Glow effect:** The waveform line has a subtle glow for better visibility
- **Fill regions:** Oscillator waveforms show positive (brighter) and negative (darker) fill areas
- **Dual cycle display:** All oscillators show 2 waveform cycles for better visibility

**Tip:** Use the Scope to visualize how modulation affects your sound. For example, watch OSC4 while an LFO modulates the wavetable position to see the waveform morphing in real-time.

# Polyphony

Access polyphony controls via the VOICES section in Settings.

## POLY (8 / 16 / 32 / 64 voices)

Multiple voice operation: - Up to 64 simultaneous voices (configurable in Settings) - **Per-voice envelope processing:** ENV1, ENV2, and VENV (Logic in VENV mode) have independent state for each voice - Voice stealing when limit is reached - Automatic level normalization - 5ms fade-in for click-free note starts

### Per-Voice Envelope System

When you connect ENV1, ENV2, or LOGIC (in VENV mode) to Ch1 Ctrl or Ch2 Ctrl via the matrix, each voice applies its own envelope independently. This means: - Chords playing simultaneously respect each note's attack - Transitions between chords without clicks - One voice's release doesn't affect other voices - VENV (Vector Envelope from Logic module) works per-voice only in VENV mode; other Logic modes (Random, Function, S&H) remain global

### Glide Control

The GLIDE fader sets portamento time for smooth pitch transitions between notes. Glide is per-voice — each voice has its own pitch trajectory, so chord transitions glide voice-by-voice in ChordAll mode and per-osc in MOSC mode.

## MOSC (Mono per oscillator)

Right-click on the OSC{N} label → “Set to Mono” toggles that oscillator into mono mode. The label updates to MOSC{N} so you see at a glance which oscillators are mono.

### How it works

- POLY remains active globally — voices coexist as usual.
- When a new voice arrives, any oscillator marked as MOSC fades out (~5 ms) on the previous voice that owned it. Other (non-MOSC) oscillators keep playing across all voices.
- Voice stealing is per-oscillator: each MOSC tracks its own ownership independently.

### Common use cases

- All 5 oscillators MOSC + Glide > 0 = classic mono synth lead behavior (legato glide via the per-voice glide system).
- 1 oscillator MOSC + 4 polyphonic = mono bass on Osc1 with polyphonic pad on Osc2-5.
- TrigHarm + Chord Spread + MOSC1234 src=Chord1-4 = arpeggiated chord spiral, each note in its own oscillator slot.
- TrigHarm + osc5 src=MIDI on top of chord-routed MOSC1234 = play melody freely over a gate-driven chord progression.

**Why per-oscillator instead of a global mono mode**

Older synths force a global mono switch. Portador exposes mono at the oscillator level so you can layer mono and poly within a single patch, and so chord-source oscillators (Chord1-5, ChordAll, Arp, etc) can each behave monophonically while the underlying polyphonic engine continues to schedule envelopes per voice.

---

# Tips and Tricks

## For bass sounds

- Use Osc1 or Osc2 with Saw or Square wave
- Octave at -1 or -2
- Filter1 LPF around 200-500Hz
- Env1 with short Attack, medium Decay, high Sustain

## For atmospheric pads

- Combine Osc4 and Osc5 with wavetables
- Spread on “Wide” or “Extra”
- Long Attack on Env1
- LFO modulating wavetable position
- Add granular with large Size and low Density

## For aggressive leads

- Osc2 or Osc3 with Hard Sync
- Move the sync oscillator's Tune with LFO or Env2
- Filter with high resonance
- Positive keytracking on filter

## For experimental textures

- Enable Cloud mode on Osc1 with Motion fader at 0.7-1.0
- Route Gate Gen to Env1 Gate In for rhythmic envelope triggering
- Frequency Shifter with small values and feedback
- Granular in Scatter or Reverse mode

## For generative rhythms

- Use Gate Gen in Euclidean mode routed to H Gate In for auto-chord progressions
- Combine Gate Gen → Env1 Gate In with arpeggiator for rhythmic gating
- Manual mode: draw a pattern in GatePadre, then increase Evolve for gradual mutations
- Route Gate Gen to Filter Cutoff for tremolo/gating effects
- Set oscillators to different Drum types and route Gate Gen → Env1 Gate In for instant drum machine patterns

## For polymetric drum machines

- Set Osc1-5 to different Drum types (e.g. 808 Kick, 909 Snare, 808 Hat, Clap, Tom)
- Route Gate Gen to Env1 Gate In
- In the matrix, right-click each Gate Gen cell to assign different tracks: T1 for kick, T2 for snare, T3 for hat, etc.
- Set different TIME values per track (e.g. T1 = 1 bar, T2 = 2 bars, T3 = 1/2 bar) for polymetric rhythms

- Use Euclidean mode on some tracks and Random on others for evolving patterns

### **For chord progressions**

- Configure the chord engine with your scale
- Use different voicings per degree
- Enable humanization for more naturalness
- Experiment with arpeggio modes

---

# Technical Specifications

- **Formats:** VST3, AU, Standalone (macOS)
- **Internal Resolution:** 32-bit float
- **Sample Rate:** Supports 44.1kHz - 192kHz
- **Oversampling:** Off/2x/4x on all oscillators (configurable)
- **Matrix:** 16x16 audio routing matrix
- **Polyphony:** POLY 8/16/32/64 (configurable) + MOSC (per-oscillator mono toggle)
- **Synthesis Engines Osc1-3:** 8 (Classic, Cloud, Resonator, Wavefolder, Supersaw, PhaseDist, NoisePitch, Drum)
- **Synthesis Engines Osc4-5:** 11 (Wavetable, Cloud, Resonator, Wavefolder, SuperWave, PhaseDist, NoisePitch, Sampler, Granular, Drum, Quantum)
- **Osc4/Osc5 Wavetables:** 70 factory tables (A-G) × 128 cycles + up to 32 user wavetables
- **Filters:** 13 types (LPF, HPF, BPF, Notch, Peak, Low Shelf, High Shelf, All Pass, Ladder, Comb, Formant, Phaser, LPG)
- **Per-Channel EQ:** 4-band parametric with visual editor
- **Per-Channel Distortion:** 5 modes (Soft Clip, Hard Clip, Wavefolder, Bitcrush, Tube)
- **Macros:** 5 macro controls with configurable ranges, direction, and curves
- **MIDI Learn:** Global CC-to-parameter assignment
- **Delay:** Stereo with 3 modes (Mono, Ping Pong, Wide)
- **Reverb:** 3 algorithms (Dattorro Plate, 8-line FDN, Recursive)
- **Resonator Bank:** 8 stereo comb filters with 4 harmonic modes
- **Pitch Shimmer:** Granular pitch shifting with feedback, 3 modes
- **Random Gate:** Probabilistic step sequencer gate, 4 step counts
- **Musical Scales:** 28
- **Vector Envelopes:** 100 predefined shapes + customizable visual editor
- **LFO Shapes:** 98
- **Logic Shapes:** 182
- **Preset Categories:** 15
- **Preset Attributes:** 30 tags

# Keyboard Shortcuts

---

Key	Action
Cmd+Z (Mac) / Ctrl+Z (PC)	Undo last parameter change
Cmd+Shift+Z (Mac) / Ctrl+Shift+Z (PC)	Redo last undone change
ESC	Close browser/dialog
Enter	Confirm/Load selected
← →	Navigate presets
Right-click any fader	Open context menu (Macro, MIDI Learn assignment)

---

**Note:** Undo/Redo tracks all parameter changes and VENV waveform edits. It does not track preset switching.

*Portador - Limitless Modular Synthesis*