

MUSHIN

DUAL FILTER LFO MATRIX SYNTHESIZER

MODEL: MOD-02

An advanced sound processor translating the mathematics of information saturation, momentum kinetics, and trend exhaustion directly into audio processing.

MUSHIN AUDIO © 2026 // ALL RIGHTS RESERVED

Mushin - Feature & Operation User Manual *Model: MOD-02 Dual Filter LFO Matrix Synthesizer & Sound Processor*

1. Introduction & The Core Philosophy

Welcome to **Mushin**, a revolutionary hybrid C++/Web sound processor that bridges the volatile mathematics of financial markets with state-of-the-art digital signal processing (DSP).

In **Mushin**, we do not merely automate parameters using standard LFOs or envelopes. Instead, **market physics and trading concepts are mapped directly to DSP math**. The waveforms you feed into Mushin represent the market data, and the processing modules act as trading constraints, momentum chokes, and liquidity drains.

The Market-to-DSP Conceptual Map

* **Information Saturation (The S-Curve Waveshaper)**: As a market trend extends, information saturates and volume plateaus. In DSP, this is represented by a soft-clipping waveshaper using a hyperbolic tangent (\tanh) transfer function. Drive increases with trend momentum, rounding off the peaks of your audio. * **Trend Exhaustion (Hard Clipping)**: When statistical exhaustion is triggered (e.g., the last buyer enters), all dynamic range evaporates. The waveshaper instantaneously bypasses the soft curve and switches to a hard-clip threshold, squaring off the waveform and generating harsh, aggressive odd-order harmonics. * **Kinetic Choke (State-Variable Filter Modulation)**: Tracking the relationship between Potential Energy and Kinetic Energy (momentum), the dual filters act as dynamic energy chokes. As market momentum slows down, the filter's cutoff frequency drops, choking off high-frequency energy. Exhaustion events multiply the filter's Resonance (Q) exponentially to create a high-pitched resonant "scream" before the frequency is clamped down. * **Information Decay (Quantization Error / Bitcrusher)**: When statistical confidence drains, the resolution of information decays. This is modeled by real-time quantization noise and sample-rate reduction. At the peak of exhaustion, audio quality disintegrates under its own weight as bit-depth drops dynamically.

2. Technical Architecture & Build Specifications

Mushin is engineered with an advanced hybrid framework combining high-performance audio processing with a highly interactive, responsive web frontend.

* **DSP Core:** Written in highly optimized C++ utilizing the **JUCE 8.0.4** framework and JUCE DSP modules. * **User Interface:** A premium web UI built with modern HTML5, CSS3, and ES6 JavaScript, rendered via **Microsoft Edge/WebView2** on Windows. * **Asset Management:** Web assets (HTML, CSS, JS) are compiled directly into the plugin binary via CMake's `juce_add_binary_data`, ensuring zero external dependency latency. * **Event Bridge & Visualizer:** Pushes real-time 30Hz high-frequency waveform updates from the C++ process block directly to the WebView canvas using `emitEventIfBrowserIsVisible` to conserve CPU cycles when the UI is closed. * **Build Environment:** Structured for CMake with the MSBuild (Visual Studio) generator. The active development binary and installer output is compiled in the `build2` directory. * **Dependencies:** Runs standalone or as a VST3/CLAP/AU plugin. On Windows, it requires `WebView2Loader.dll` present in the binary output directory.

3. Module & Parameter Directory

Mushin features 8 high-performance DSP modules, fully automatable and exposed to the host DAW via the JUCE AudioProcessorValueTreeState (APVTS).



Module A: S-Curve Waveshaper (Saturation & Exhaustion)

The primary saturation stage, implementing information saturation.

Parameter ID	Parameter Name	Range	Default	Description
<code>drive</code>	Drive	1.0 to 20.0	1.0	Amplifies the input signal into the waveshaper to increase saturation harmonics.
<code>exhaustion</code>	Exhaustion	OFF, ON	OFF	Bypasses soft-clipping \tanh and enables hard-clip threshold clamping.
<code>threshold</code>	Threshold	0.0 to 1.0	1.0	The amplitude ceiling for hard-clipping when <code>exhaustion</code> is active.
<code>mix</code>	Dry/Wet Mix	0.0 to 1.0	1.0	Blends between processed (wet) and clean (dry) signals.
<code>gain</code>	Gain	0.0 to 2.0	1.0	Post-waveshaper output make-up gain.

TIP

To get warm, classic analog-like drive, keep `exhaustion` **OFF** and push `drive` between 3.0 and 8.0. For absolute industrial destruction, turn `exhaustion` **ON**, reduce the `threshold` to 0.4, and maximize `drive`.

Module B: Dual Filter System

Mushin contains two independent filters (A & B) that can be routed in Series or Parallel.

* **Serial Routing:** Audio -> Saturation -> Filter A -> Filter B -> Outputs. * **Parallel Routing:** Audio splits equally to Filter A and Filter B. Their processed outputs are summed ($0.5 \times A + 0.5 \times B$).

Filter Models

1. **Clean (TPT):** A modern, zero-delay feedback Topology Preserving Transform filter. Ultra-transparent.
2. **Vintage (Ladder):** A classic 4-pole transistor ladder model. Warms up when pushed.
3. **Acid (Diode):** A legendary 3-pole diode ladder filter with squelchy, raw resonance behavior.
4. **Digital (State Variable):** A crisp, versatile filter offering immediate high-precision cuts.

Filter Modes

* **Lowpass:** Attenuates frequencies above the cutoff point. * **Highpass:** Attenuates frequencies below the cutoff point. * **Bandpass:** Isolates a narrow frequency band around the cutoff point. * **Notch:** Attenuates a narrow frequency band around the cutoff point (creates phase cancellation).

Filter Parameters (Identical for A and B)

Parameter ID	Parameter Name	Range	Default	Description
<code>filter_a_cutoff</code> / <code>filter_b_cutoff</code>	Filter Cutoff	20.0 Hz to 20,000.0 Hz	20,000.0 Hz	Filter

cutoff frequency (skewed logarithmically for precise low-end control). | | / | Resonance | to | | Boosts the amplitude at the cutoff frequency. | | / | Filter Drive | to | | Pre-filter drive stage to warm up or saturate the signal before filtering. | | / | Filter Grit | to | | Post-filter clipping saturation, adding rasp and fuzz directly to the resonance peaks. | | / | Filter Type | , , , | | Selects the DSP filter algorithm. | | / | Filter Mode | , , , | | Selects the filter slope response. | | | Routing Mode | , | | Controls how the two filter blocks are structurally connected. |

Module C: LFO & Modulation Matrix

Two sample-accurate LFOs can modulate key filter settings through a bipolar 2x6 grid matrix.

LFO Controls

* **LFO 1 Freq** () / **LFO 2 Freq** (): to (Default). Skewed log-scale. * **LFO 1 Wave** () / **LFO 2 Wave** (): * **Sine**: Smooth cyclic modulation. * **Triangle**: Linear ramp up and down. * **Saw**: Ramp up with sharp instantaneous reset. * **Square**: Alternates between maximum positive and negative values. * **Random**: Sample & Hold random values, synchronized to LFO frequency rate.

Modulation Matrix Layout

The matrix coordinates assign **LFO 1** and **LFO 2** (Sources) to six primary targets (Targets 1–6) with a bipolar amount ranging from to (Default / No Modulation).

| Target ID | Modulation Target | Effect | | :--- | :--- | :--- | | **Target 1** | **Filter A Cutoff** | Modulates frequency by up to **+/- 5 octaves**. | | **Target 2** | **Filter A Resonance** | Modulates resonance depth by up to **+/- 1.0**. | | **Target 3** | **Filter A Grit** | Modulates post-filter grit saturation depth by up to **+/- 1.0**. | | **Target 4** | **Filter B Cutoff** | Modulates frequency by up to **+/- 5 octaves**. | | **Target 5** | **Filter B Resonance** | Modulates resonance depth by up to **+/- 1.0**. | | **Target 6** | **Filter B Grit** | Modulates post-filter grit saturation depth by up to **+/- 1.0**. |

Module D: Sidechain Processor (Envelope Follower)

A dynamic signal tracker that extracts the amplitude envelope of an incoming source and routes it to dynamically duck or pump key variables in Mushin.

| Parameter ID | Parameter Name | Range | Default | Description | | :--- | :--- | :--- | :--- | :--- | | | SC Active | , | | Bypasses or enables the sidechain processor. | | | SC Source | , , | | Selection of detector input signal: Main feed,

DAW Aux Sidechain feed, or internal test cycle. | | `sc_threshold` | SC Threshold | `-60.0 dB` to `0.0 dB` | `-24.0 dB` | Level above which the envelope tracker begins generating control signals. | | `sc_attack` | SC Attack | `0.1 ms` to `500.0 ms` | `10.0 ms` | Lookahead envelope attack smoothing duration. | | `sc_release` | SC Release | `1.0 ms` to `2000.0 ms` | `100.0 ms` | Envelope release decay duration. | | `sc_mode` | SC Mode | `Peak`, `RMS` | `Peak` | Peak-level fast tracking vs. Root-Mean-Square average level tracking. | | `sc_amount` | SC Amount | `-100.0%` to `100.0%` | `0.0%` | Bipolar scaling of the sidechain control signal. Negative amounts duck, positive pump. | | `sc_target` | SC Target | `Drive`, `Cutoff`, `Gain` | `Gain` | Target to modulate: Saturation drive, Filter cutoffs, or Output Gain. | | `sc_hp_freq` | SC HPF | `20.0 Hz` to `2,000.0 Hz` | `20.0 Hz` | High-pass filter in the detector path to ignore sub frequencies. | | `sc_lp_freq` | SC LPF | `500.0 Hz` to `20,000.0 Hz` | `20,000.0 Hz` | Low-pass filter in the detector path to isolate low-end weight (e.g., ducking on kick drum). |

NOTE

The sidechain's `sc_target = Cutoff` modulates **both** Filter A and Filter B Cutoff parameters simultaneously based on the active `sc_amount` depth.

Module E: Noise Generator / Auxiliary Oscillator

Injects synthetic textures, raw noise, or primary synth tones into the processing signal path, allowing Mushin to act as an independent tone generator or an exciter.

| Parameter ID | Parameter Name | Range | Default | Description | | :--- | :--- | :--- | :--- | :--- | | `noise_active` | Noise Active | `OFF`, `ON` | `OFF` | Bypasses or enables the auxiliary generator. | | `noise_type` | Noise Type | `White Noise`, `Pink Noise`, `Sine Tone`, `Triangle Wave`, `Sawtooth Wave`, `Square Wave` | `White Noise` | Waveform signature of the generator. Includes noise models and classic oscillators. | | `noise_freq` | Noise Freq | `20.0 Hz` to `20,000.0 Hz` | `440.0 Hz` | Frequency of the synth waveforms (Sine, Triangle, Saw, Square). | | `noise_level` | Noise Level | `0.0` to `1.0` | `0.1` | Master gain level of the injected oscillator. | | `noise_routing` | Noise Routing | `Pre-Dist`, `Pre-Filter`, `Post-Filter` | `Pre-Dist` | Placement in signal path: Saturation input, Filter input, or Output summing stage. | | `noise_fm_mod` | Filter FM Mod | `0.0` to `1.0` | `0.0` | Direct FM modulation depth routing the noise oscillator directly into Filter A/B Cutoffs. |

Module F: Trance Gate

A rhythmic gate that modulates the output amplitude based on host tempo and pre-defined sequences.

* Gate Patterns:

1. `Straight 16th`: Steady rapid chops.

2. **Offbeat 16th**: Traditional offbeat pulsing syncopations.
3. **Classic 1** & **Classic 2**: Iconic trance gate chopping cycles.
4. **Four-On-Floor**: Rhythmic gating matching typical 4/4 kicks.
5. **Galop**: Bouncing triplets.
6. **Space Gate**: Ethereal long-duration gating patterns.
7. **Euclidean 5**: Mathematically spaced rhythmic pulse sequence.

Trance Gate Parameters

Parameter ID	Parameter Name	Range	Default	Description
<code>tg_active</code>	Gate Active	<code>OFF</code> , <code>ON</code>	<code>OFF</code>	Enactivates the rhythmic gate sequence.
<code>tg_mix</code>	Gate Mix	<code>0.0</code> to <code>1.0</code>	<code>1.0</code>	Dry/Wet mix of the gated signal.
<code>tg_pattern</code>	Gate Pattern	*(8 Choices)* <code>Straight 16th</code>		
<code>tg_rate</code>	Gate Rate	<code>1/16</code> , <code>1/8</code> , <code>1/4</code> , <code>1/16</code>		Speed of the pattern step transitions relative to host tempo.
<code>tg_start</code>	Gate Start	<code>0.0%</code> to <code>100.0%</code>	<code>5.0%</code>	Rhythmic gate attack envelope ramp-up speed.
<code>tg_hold</code>	Gate Hold	<code>10.0%</code> to <code>100.0%</code>	<code>50.0%</code>	Duration that the gate remains wide open per active step.
<code>tg_end</code>	Gate End	<code>0.0%</code> to <code>200.0%</code>	<code>10.0%</code>	Rhythmic gate decay/release envelope ramp-down speed.
<code>tg_depth</code>	Gate Depth	<code>0.0%</code> to <code>100.0%</code>	<code>100.0%</code>	Depth of attenuation during closed steps (100% means total silence).

Module G: Info Decay (Quantization Error / Bitcrusher)

The information decay simulation block, acting as an advanced digital bitcrusher.

Parameter ID	Parameter Name	Range	Default	Description
<code>qe_active</code>	Info Decay Active	<code>OFF</code> , <code>ON</code>	<code>OFF</code>	Activates digital bitcrushing and downsampling.
<code>qe_depth</code>	Decayed Resolution	<code>2.0</code> to <code>16.0</code> Bits	<code>16.0</code> Bits	Continuous amplitude resolution reduction (Logarithmic sweet spot centered at <code>5.5</code> bits).
<code>qe_downsample</code>	Time Resolution	<code>1.0</code> to <code>32.0</code>	<code>1.0</code>	Sample rate divider (e.g. 2.0 downsamples a 44.1kHz stream to 22.05kHz).
<code>qe_mix</code>	Decay Mix	<code>0.0</code> to <code>1.0</code>	<code>1.0</code>	Blends crushed signal with pre-crushed clean audio.
<code>qe_link</code>	Exhaustion Link	<code>OFF</code> , <code>ON</code>	<code>OFF</code>	Dynamically overrides resolution depth and plunges it to a crushed <code>4.0</code> bits during exhaustion spikes!

WARNING

Reducing `qe_depth` to low numbers (e.g. `2.0` to `4.0` bits) will significantly amplify signal distortion and white noise. Reduce output monitor volume or utilize low wet/dry mixes (`qe_mix`) to bleed in textures subtly.

Module H: Delay Processor (Stereo / Ping-Pong)

A highly versatile stereo delay line optimized for both rhythmic delays and physical-modeling comb filtering.

Parameter ID	Parameter Name	Range	Default	Description
<code>delay_active</code>	Delay Active	OFF, ON	OFF	Bypasses or enables the stereo delay line.
<code>delay_time</code>	Delay Time	1.0 ms to 2000.0 ms	300.0 ms	Delay duration. Highly warped skew-factor curve: first 30% of travel targets 1ms to 17ms for metallic resonators.
<code>delay_feedback</code>	Delay Feedback	0.0 to 0.95	0.3	Controls decay duration. Clamped safely below 1.0 to prevent feedback explosions.
<code>delay_mix</code>	Delay Mix	0.0 to 1.0	0.3	Dry/Wet ratio of the delay tails.
<code>delay_pingpong</code>	Ping-Pong	OFF, ON	OFF	Bounces delay lines between the Left and Right channels for wide stereophony.
<code>delay_sync</code>	Delay Sync	OFF, ON	OFF	Syncs delay time directly to the DAW host tempo, ignoring the millisecond knob.
<code>delay_tempo</code>	Delay Tempo Sync	1/16, 1/8, 1/4, 1/2	1/4	Selected note value interval for tempo-sync delays.

4. UI Operation, Presets & Aesthetics

Mushin is designed to look premium, dynamic, and state-of-the-art.

The Dynamic Theme System

Mushin operates a **Theme-Injection Architecture** where visual styles are fed dynamically from C++ as a virtual stylesheet named `skin.css`. The UI updates instantly when themes are selected. Settings are persisted in `%APPDATA%\MushinPlugin\settings.ini` (Windows) or `~/Library/Application Support/MushinPlugin/settings.ini` (macOS).

Users can select from **7 premium visual modes** in the header dropdown:

1. **Industrial (Default):** A sleek, professional brushed metal chassis with electric blue (`#00bfff`) primary knobs and warm orange accents.
2. **Synthwave:** A dark magenta chassis (`#14091a`) styled in glowing hot-pink (`#ff00aa`) and neon cyan (`#00ffff`) laser lines.
3. **Acid:** A poisonous dark green chassis (`#091209`) with high-voltage glowing lime-green (`#bfff00`) and radio-active neon controls.
4. **Firepits:** A volcanic obsidian chassis (`#120906`) with flowing lava orange-red (`#ff3c00`) highlights and glowing parameters.
5. **Ocean Deep:** A deep oceanic blue-navy chassis (`#060b1a`) glowing in tropical neon cyan (`#00f3ff`) and marine accents.
6. **Ice World:** A frosted arctic grey-blue chassis (`#0f151c`) with sharp icy-blue (`#66e0ff`) highlights and pristine white accents.
7. **Dark Hellish:** A terrifying gothic blood-red and absolute black chassis (`#0e0606`) glowing in dark embers (`#ff2200`).

Preset Browser & Search

The header panel contains a robust XML-based preset manager linked directly to local system folders.

* **Windows Presets Path:** `%AppData%\Mushin\Presets\` * **macOS Presets Path:** `~/Library/Application Support/Mushin/Presets/` * **Controls:** * **Preset Selector:** Choose from factory default configurations or custom user settings. * **Search Box:** Interactive instant search filtering preset names on-the-fly. Press `Enter` to load the top match. * **Arrow Buttons (◀ / ▶):** Instantly cycle to the previous or next preset. * **Save Button:** Prompts the user to enter a name, serializing current settings into a custom XML preset file. * **Delete Button:** Deletes the active user preset file from the system folder.

5. Sound Design Recipes (Annex)

Here are five creative sound design setups to transform any input signal using Mushin's unique DSP mappings.

Recipe 1: The Acid Flashback (Squelchy Bassline Exciter)

* **Aesthetic Theme:** Acid * **Input Audio:** A steady rhythmic synth block or sub-bass. * **S-Curve**
Waveshaper: `drive` at `8.0`, `exhaustion` `OFF`, `mix` at `0.7`. * **Dual Filters:** Routed in **Serial**. * **Filter A:** Mode `Lowpass`, Type `Acid`, `cutoff` at `400 Hz`, `resonance` at `0.85`, `drive` at `3.0`, `grit` at `0.4`. * **Filter B:** Mode `Lowpass`, Type `Clean`, `cutoff` at `1.2 kHz`, `resonance` at `0.2`. * **LFO 1:** Wave `Saw`, frequency at `2.5 Hz`. * **Mod Matrix:** Mod LFO 1 to Target 1 (Filter A Cutoff) set to `+0.7`. *
Output: Creates a classic, highly resonant squelching acid filter sweep with rich harmonic saturation.

Recipe 2: The Vocaloid A-O (Dual Vowel Resonator)

* **Aesthetic Theme:** Synthwave * **Input Audio:** Sustained broad-spectrum chords or white noise. * **S-Curve**
Waveshaper: `drive` at `2.0`, `mix` at `0.3`. * **Dual Filters:** Routed in **Parallel**. * **Filter A:** Mode `Bandpass`, Type `Vintage`, `cutoff` at `800 Hz`, `resonance` at `0.9`. * **Filter B:** Mode `Bandpass`, Type `Vintage`, `cutoff` at `1800 Hz`, `resonance` at `0.9`. * **LFO 1:** Wave `Sine`, frequency at `0.4 Hz`. * **Mod Matrix:** * Mod LFO 1 to Target 1 (Filter A Cutoff) set to `+0.5`. * Mod LFO 1 to Target 4 (Filter B Cutoff) set to `-0.5` (Opposite phase). * **Output:** The bandpass filters sweep in opposite directions, carving out vowel formants that mimic vocal patterns ("Ah-Oh" sweeps).

Recipe 3: Warm Tape Saturation (Vintage Tape Glue)

* **Aesthetic Theme:** Industrial * **Input Audio:** Master bus mix or drum loop. * **S-Curve**
Waveshaper: `drive` at `3.5`, `exhaustion` `OFF`, `mix` at `0.4`. * **Dual Filters:** Routed in **Serial**. * **Filter A:** Mode `Lowpass`, Type `Clean`, `cutoff` at `8,500 Hz` (rolls off digital high-end harshness), `resonance` at `0.0`. * **Filter B:** Bypassed (`cutoff` at `20k`, `resonance` at `0.0`). * **Sidechain:** Active `ON`, target `Gain`, source `Internal`, amount `-15%`, attack `20 ms`, release `150 ms`. * **Output:** Gentle high-end roll-off coupled with slight dynamic gain-ducking and hyperbolic soft-saturation, beautifully mimicking vintage tape glue.

Recipe 4: The Sub Destroyer (Harmonic Weight Generator)

* **Aesthetic Theme:** Dark Hellish * **Input Audio:** A clean 808 kick or sine sub-bass. * **S-Curve**
Waveshaper: `drive` at `15.0`, `exhaustion` `ON`, `threshold` at `0.5`, `mix` at `0.6`. * **Dual Filters:** Routed in **Serial**. * **Filter A:** Mode `Lowpass`, Type `Vintage`, `cutoff` at `120 Hz`, `resonance` at `0.4`

(locks onto sub frequencies). * **Filter B:** Bypassed. * **Output:** Squared clipping converts the pure sub sine wave into a roaring square-like wave, while the low-pass filter filters out high fizz, keeping the bass thick and highly saturated.

Recipe 5: Digital Info Decay (Destroyed Lo-Fi Ringer)

* **Aesthetic Theme:** Ice World * **Input Audio:** Clean electric guitar or synth pad. * **Info Decay**

(Bitcrusher): Active **ON**, `qe_depth` at `5.5 Bits` (The Sweet Spot), `qe_downsample` at `3.0x`, `qe_mix` at `0.5`. * **S-Curve Waveshaper:** `drive` at `10.0`, `exhaustion` **ON**, `threshold` at `0.9` (Linked via `qe_link` = **ON**). * **Delay:** Active **ON**, time `15 ms` (comb filtering), `feedback` at `0.7`, `mix` at `0.4`. *

Output: Creates an incredibly unique, ringing digital crunch where transients disintegrate into bit-depth decay while carrying a metallic, frozen comb-filter tail.

6. Underway & Future Roadmap

Mushin is constantly evolving. Below is a summary of the conceptual models and DSP blocks currently in development:

A. The Air Gap Suspension (Granular Looper)

* **Concept:** Represents the market "Air Gap" pause. Halts the write pointer of a circular buffer (50–100ms long) during extreme exhaustion events, while the read pointer loops continuously with an applied Hanning window to prevent clicks. * **Status:** *Underway (DSP conceptual design complete, UI toggle planned)*. * **Audible Result:** When exhaustion hits, forward time freezes into a static, granular drone or ambient stutter until the market trend reverses.

B. Final Stage Limiter / LMC (Listen Mic Compressor)

* **Concept:** A final-stage dynamic brickwall limiter / SSL character compressor to guarantee safety against feedback while providing aggressive NYC-style pumping. * **Parameters:** `limiter_active`, `limiter_mode` (Clean vs LMC), `limiter_drive`, `limiter_ceiling`, `limiter_release`, `limiter_mix`. * **Status:** *Underway (DSP algorithm defined in specs, awaiting APVTS parameter wiring)*.

C. LFO Phase Invert

* **Concept:** A simple checkbox or toggle in LFO 1 & LFO 2 sections to instantly flip the phase of the modulator wave ($\phi = -\phi$). * **Status:** *Planned*. * **Audible Result:** Allows instant inversion of sweeps (e.g., triangle sweep going up on Filter A while ramping down on Filter B with a single shared sync rate).

D. Vocal Tract Formant Filters

* **Concept:** An addition to the `filter_type` parameter to include a physical vocal formant model. * **Status:** *Planned*. * **Audible Result:** Allows users to choose specific vowel vowels ("A", "E", "I", "O", "U") as filter curves rather than needing parallel bandpass sweeps.