SIDSynth User Manual

Version 1.10.22 — October 25, 2026

Table of Contents

Introduction
Installation
Interface Overview
SIDSynth Advanced Menu
SIDSynth Parameter Reference
MIDI Mapping
NRPN Parameter Index
Notes

1. Introduction

1.1 What is SIDSynth?

SIDSynth is a modern software synthesizer inspired by the unmistakable character of the Commodore 64's SID chip. Rather than recreating the original hardware at the circuit level, SIDSynth uses contemporary digital signal processing and mathematically generated waveforms to capture the essence of SID-style sound while offering far more creative freedom.

This approach preserves the familiar SID flavor, but without the limitations of vintage hardware. It's a blend of retro charm and modern capability — a kind of musical *retro-futurism* that honors the past while embracing the possibilities of today.

1.2 Why SIDSynth Is Not a Hardware Emulation

Traditional hardware emulation aims to reproduce every analog quirk, timing irregularity, and electrical nuance of the original SID chip. While that can be authentic, it also means inheriting all the constraints of 1980s hardware.

SIDSynth intentionally takes a different direction:

- **More flexible** — modern polyphony, portamento, strobe effects, expressive MIDI control
- **More stable** — no chip-to-chip variation or unpredictable analog drift
- **Easier to extend** — designed for creative sound design, not historical replication

The tradeoff is simple: SIDSynth is not cycle-accurate, and it will not reproduce every SID artifact. In exchange, you get a stable, expressive instrument built for real-world music production.

1.3 SIDSynth vs. Full SID Emulation

Approach	Strengths	Trade-offs
SIDSynth	Modern features, consistent performance, creative flexibility	Not hardware-accurate-
Full SID Emulation	Authentic chip behavior	CPU-heavy, limited by original design-

---

2. Installation

2.1 Installing SIDSynth

To install SIDSynth, simply copy **SIDSynth.vst3** into your system's VST3 directory.

Windows default path:
C:\Program Files\Common Files\VST3

2.2 Host Requirements

SIDSynth requires:

- A **VST3-compatible host**
- **WebView-enabled interface support**

---

3. Interface Overview

3.1 SIDSynth Menu Bar



SIDSynth includes a floating menu bar that stays visible no matter where you are in the interface. It provides quick access to essential system functions and real-time information. The bar cannot be collapsed and always remains in view.

1 > — Hamburger Menu
Opens and closes the main menu system.

2 > — Voices Used (%)
Displays how many voice resources SIDSynth is currently using.
If this reaches 100%, the oldest active voices are automatically released to make room for new ones.
In practice, SIDSynth is lightweight enough that this rarely becomes a limitation — and you can always open multiple instances if needed.

3 > — Panic Button
Immediately sends “all notes off” in case something runs away from you.

Note:
SIDSynth is designed to be explored and experimented with. You can combine almost anything (aside from features like Hard Sync and Ring Modulation – there are requirements for these features), and even if you create something wild or unstable, you won’t break the synth.
That’s part of the fun — just like the original Commodore 64, experimentation often leads to surprising and delightful results.

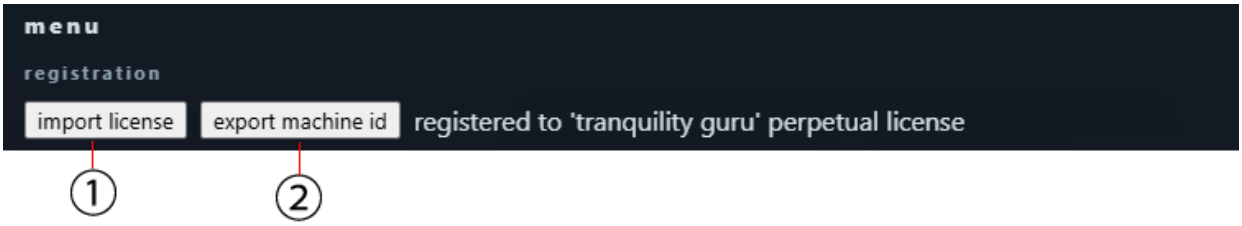
4 > — System Messages
Brief notifications appear here when loading, saving, or performing other system actions.

5 > — SIDSynth Advanced Menu Opens the advanced parameter menu.

6 > — MIDI indicator lamps The lamps light when oscillators receive MIDI data. The lamp colors match the oscillator’s theme color.

3.2 SIDSynth Hamburger Menu

Registration



Inside the hamburger menu, you'll find the registration section.
SIDSynth is fully functional in unregistered mode, with one exception: you cannot load or save patches or create/load .sidbank files. There are **no nags, no time limits, no noise bursts, and no mutes** — the goal is to let you evaluate the synth freely and comfortably.

When you register SIDSynth, the license becomes tied to the machine you register it on. The license is **perpetual**, meaning it's yours forever. If you replace or significantly change your hardware, simply contact support with your original registration information to receive a new key.

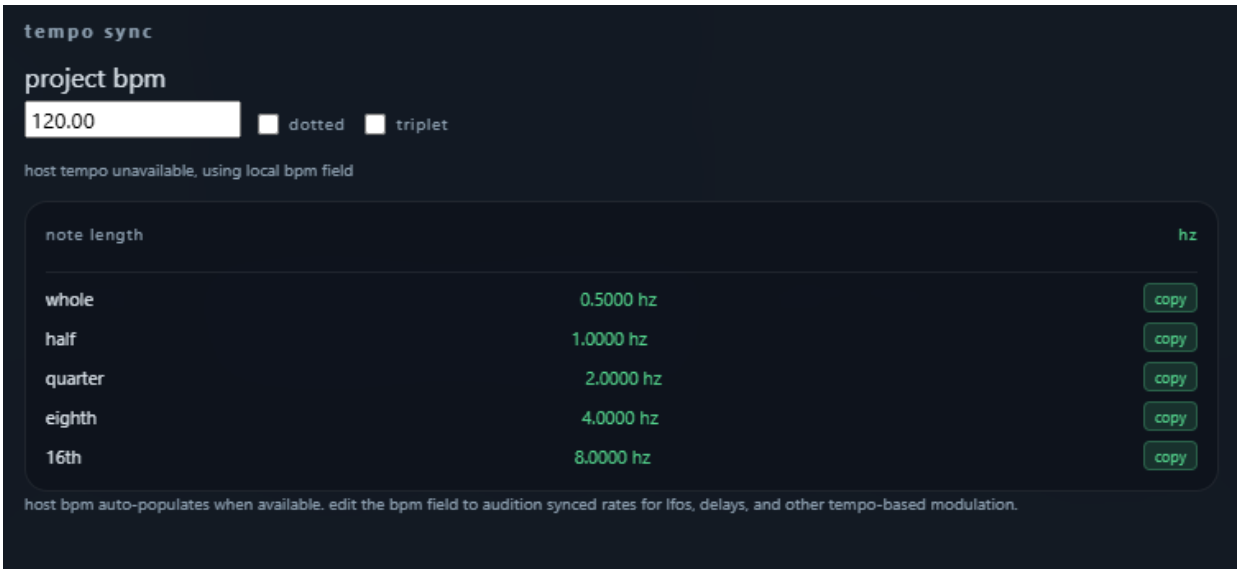
1 > — Import License
After purchasing, you'll receive a custom key via email. Save it to your system and import it here to unlock full functionality.

2 > — Export Machine ID
Generates a hardware fingerprint used to create your custom key.
This contains **no personal data** — only hardware identifiers.
Be sure to export the Machine ID **from the system you intend to register**, or the key will not work.

Keys are generated manually on an offline system for security, so delivery may take up to 48 hours.

Once you have your Machine ID, send it to 6581@sidsynth.com with your purchase receipt and the name you want on the registration. The key comes back by email.

Tempo Sync



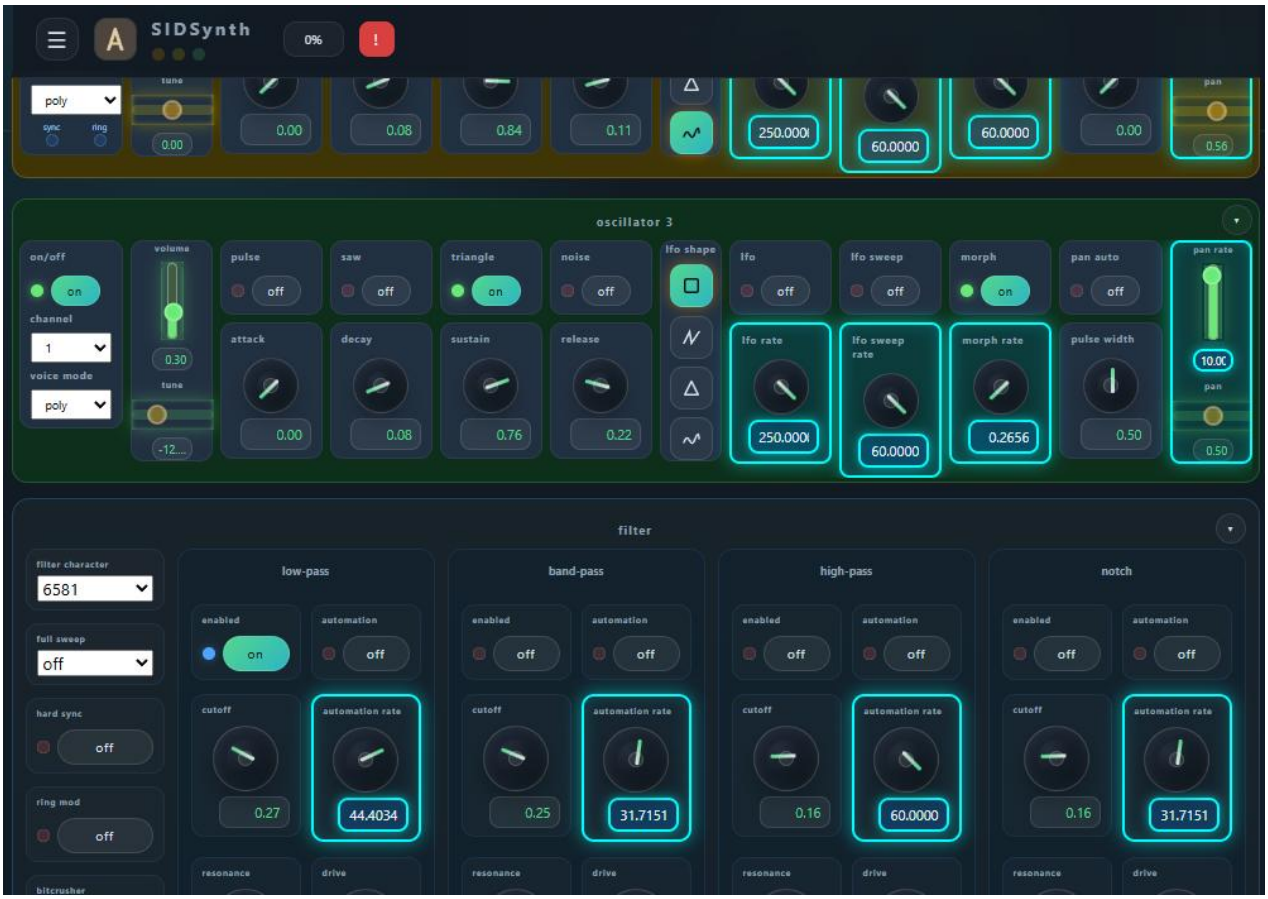
When working in a DAW, you'll often want SIDSynth's timing to match your project.
SIDSynth measures rhythmic functions in **Hz**, not BPM, which can be unintuitive — Tempo Sync bridges that gap.

If your DAW provides BPM information, SIDSynth picks it up automatically. If not, enter the tempo in Project BPM. SIDSynth converts BPM into Hz with four-decimal precision for common note lengths, including dotted and triplet values. The arpeggiator uses Project BPM when the host does not publish a tempo. The default is 120.

Clicking a "copy" button closes the menu and enters **Paste Mode**, highlighting all parameters that accept timing values. One click drops the value into place.

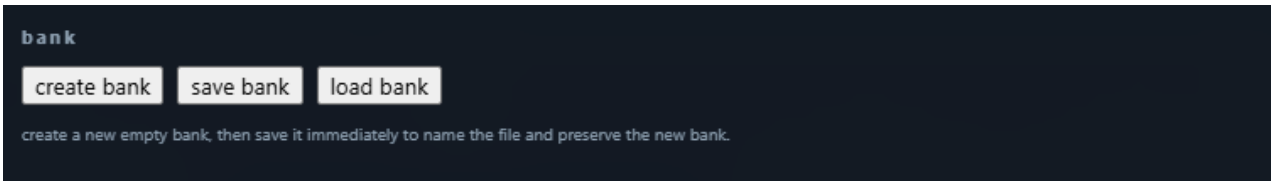
Why four decimal places?
Because BPM and Hz are different measurement systems, clean conversions don't always land neatly within one or two decimals. Four digits help prevent drift.

Paste Mode



Any parameter that can accept a timing value from Tempo Sync will highlight in cyan.
Click once to paste the copied value.

Bank Selection

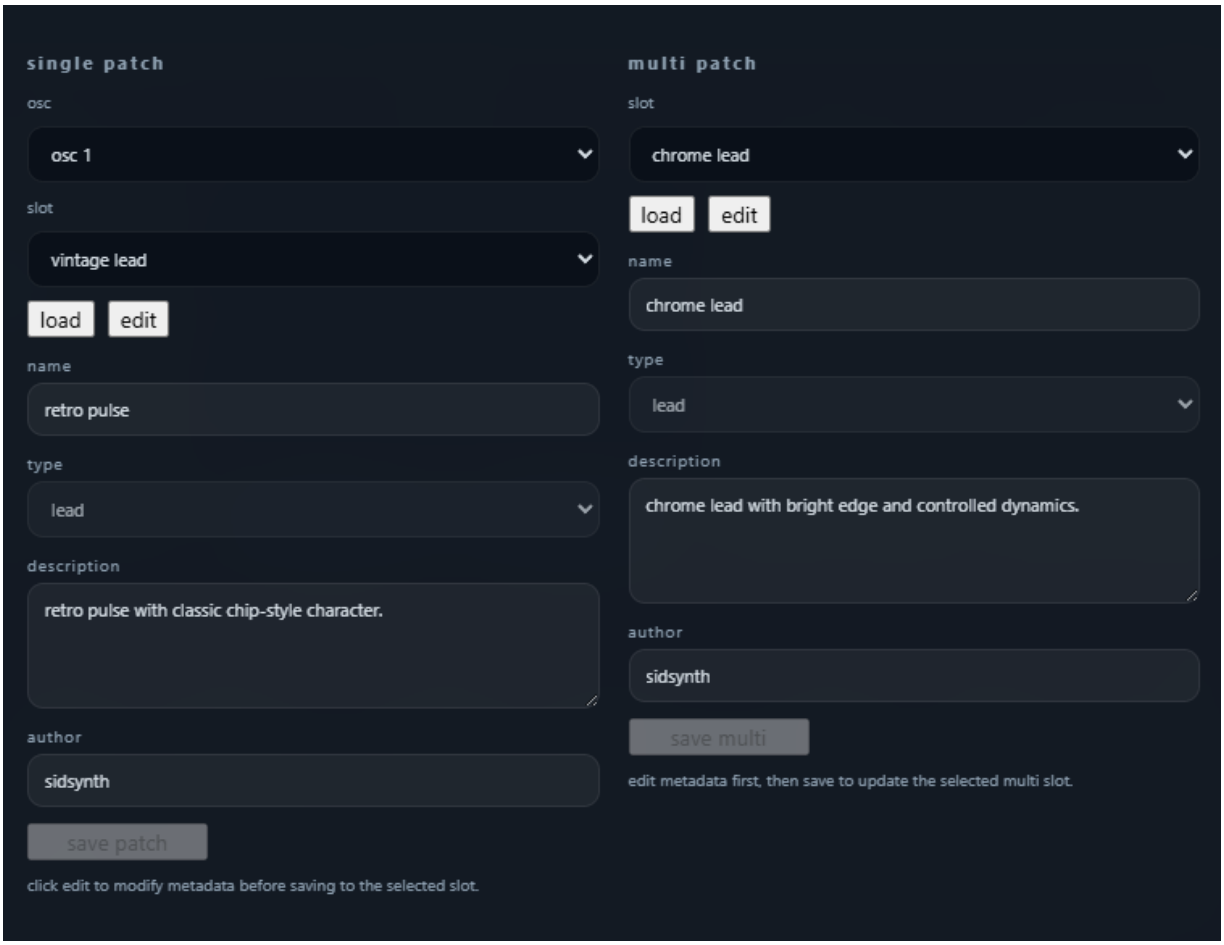


Create Bank
Creates a new, empty bank with 64 single slots and 64 multi slots.

Save Bank
Saves the entire bank — perfect for backups or sharing.

Load Bank
Loads an existing .sidbank file (e.g., factory.sidbank).

Using SIDBank patches

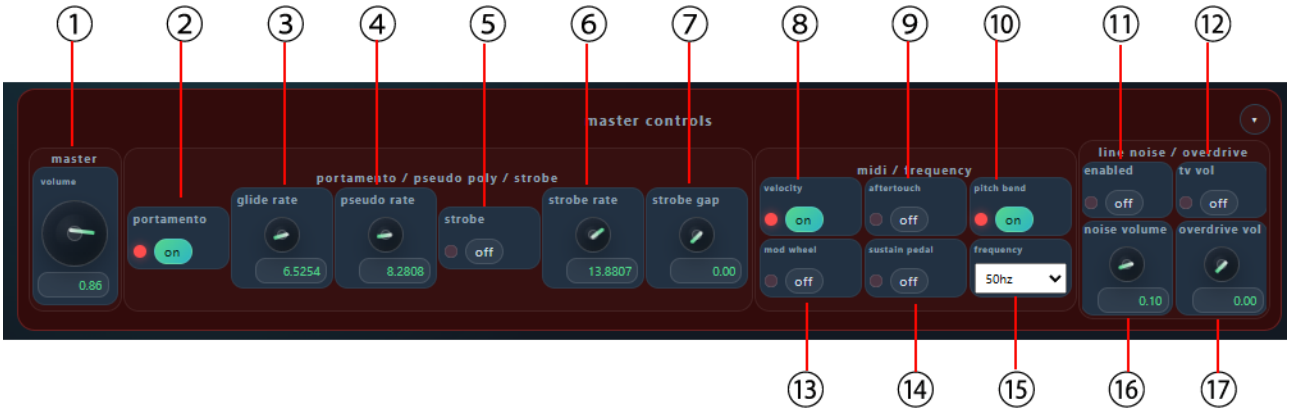


Inside the patch section, you can browse single and multi patches stored in the loaded .sidbank.

Single patches load into the oscillator you select.
You can edit the patch's name, type, description, and author.
When saving, make sure the correct oscillator is selected.

Multi patches save and load the *entire* SIDSynth configuration — oscillators, filters, effects, and master settings.

3.3 Master Controls



The Master Controls section affects **every part** of SIDSynth equally. These settings shape the overall behavior of the instrument, from performance response to global sound coloration.

1 > — Volume
Controls the overall output level of SIDSynth.

2 > — Portamento
A classic monosynth feature. When enabled, notes glide smoothly from one pitch to the next — a hallmark of many ’70s synth performances.

3 > — Glide Rate
Determines how quickly the pitch transitions during portamento.
Lower values = faster glide.

4 > — Pseudo Rate
The original SID chip had only three oscillators, each capable of playing a single note. To simulate chords, programmers rapidly cycled through notes, creating a shimmering “pseudo-polyphonic” effect.
Pseudo Rate (in Hz) controls how quickly SIDSynth cycles through notes when an oscillator is set to **PSEUDO POLY**.

5 > — Strobe
Cycles between oscillators in sequence, muting and unmuting them one at a time.
Unlike Pseudo Poly (which cycles notes), Strobe cycles **oscillators**, allowing each one to contribute its own character.

6 > — Strobe Rate
Controls how quickly oscillators switch during the strobe effect.

7 > — Strobe Gap
Sets the delay (in seconds) between one oscillator muting and the next one activating.

The Hidden Arpeggio Trick

Even though SIDSynth has a held note arpeggiator — if you wish, you can build one yourself:

Set all oscillators to **PSEUDO POLY** on **channel 1**.

Set **Release** on all oscillators to **0.01**.

Tune Osc 1 to **+12**, Osc 2 to **–12**, Osc 3 to **0**.

Enable **Pseudo Rate** and **Strobe Rate** at low values.

Add a tiny **Strobe Gap**.

Play a chord.

You'll get a wonderfully complex, musical arpeggio reminiscent of classic Duran Duran textures — and every note stays in key.

Standard MIDI Controls

(8, 9, 10, 13, 14 >)

These switches enable or disable SIDSynth's response to common MIDI messages:

- **Velocity** — how hard a key is pressed
- **Aftertouch** — pressure applied after the initial strike
- **Pitch Bend** — smooth pitch shifting
- **Mod Wheel** — often used for vibrato, tremolo, or other modulation
- **Sustain Pedal** — holds notes after key release

These switches turn the responses on or off. Aftertouch amount, curve, and rate, and the pitch-bend range, live in Advanced controls.

Line Noise & TV Overdrive

15 > — Frequency (and 50/60 Hz hum)
The original SID behaved differently on PAL 50 Hz and NTSC 60 Hz power. SIDSynth defaults to 60 Hz. In that mode, several rate controls can run up to 60 Hz. At 50 Hz, those same controls are capped at 50 Hz, and a small indicator appears under each limited control. Unlike a real SID, this setting does not retune the voices. You can still add a steady 50/60 Hz hum for nostalgic authenticity.

Enable the hum with **11 >** and set its level with **16 > Noise Volume**.

12 > — TV Vol
Adds gritty, overdriven "old TV speaker" noise.
Its level is controlled by **17 > Overdrive Volume**.

Use these sparingly — unless you *want* to lean into the chaos. In that case, feel free to crank them.

3.4 The Oscillators



SIDSynth includes **three oscillators**, just like the original SID — but with far more flexibility and modern enhancements.

1 > — On/Off
Enables or disables the oscillator.

25 > — Channel
Sets the MIDI channel this oscillator responds to.

26 > — Voice Mode
Determines how many notes the oscillator can play at once:

- **POLY** — fully polyphonic
- **MONO** — classic one-voice behavior
- **PSEUDO POLY** — cycles through held notes to simulate polyphony, creating the signature SID shimmer

14 > — Sync & Ring Indicator Lamps
These light up when **Hard Sync** or **Ring Mod** is active (Osc 1 & 2 only).

2 > — Volume
Controls the oscillator's output level.

15 > — Tune
Adjusts pitch in semitones, from −24 to +24. For nostalgic SID detuning:

+0.63 ≈ PAL → NTSC pitch shift

–0.63 ≈ NTSC → PAL pitch shift

Waveforms (3, 4, 5, 6 >)

Each oscillator can use any combination of:

- Pulse (Square)
- Saw
- Triangle
- Noise

Selecting multiple waveforms blends them into a more complex tone.

Envelope (16, 17, 18, 19 >)

Each oscillator has its own ADSR envelope:

- **ATTACK** — time to reach full volume
- **DECAY** — time to fall to sustain level
- **SUSTAIN** — level held while the note is pressed
- **RELEASE** — time to fade out after release

LFO (7, 8, 20 >)

Each oscillator includes its own LFO:

7 > LFO Shape — Pulse, Saw, Triangle, Noise

8 > LFO On/Off

20 > LFO Rate — speed in Hz

LFO Sweep (9, 21 >)

LFO Sweep gradually changes the LFO's shape over time.

9 > Enable/Disable

21 > Sweep Rate

Morph (10, 22 >)

Morph cycles smoothly through selected waveforms, looping continuously for the duration of the note.

10 > Enable/Disable

22 > Morph Rate

Pulse Width (23 >)

Adjusts the shape of the Pulse waveform:

0.05 = very narrow (bright, nasal)

0.95 = very wide (hollow, scooped)

0.50 = symmetrical (classic square wave)

Pulse width affects any waveform combination that includes pulse. In Layer mode the control stays between 5% and 95%, and a small indicator appears under the knob. SID mode uses the full range.

Pan & Pan Automation (24, 11, 12 >)

24 > Pan — stereo position (0.0 left, 1.0 right)

11 > Pan Auto — enables automatic panning

12 > Pan Rate — speed of stereo movement

Collapse/Expand (13 >)

Each major section of SIDSynth can be collapsed or expanded to reduce scrolling and keep your workspace tidy.

3.5 Filters



SIDSynth includes four filter types, modeled after the SID's distinctive 6581 and 8580 variants. These are **mathematical models**, not hardware emulations — meaning they retain the SID's character while remaining more predictable and musically usable.

1, 2, 3, 4 > — Filter Types

- **Low-pass** — keeps low frequencies, cuts highs
- **High-pass** — keeps highs, cuts lows
- **Band-pass** — isolates a narrow frequency band
- **Notch** — removes a narrow band, keeps everything else

Filters can be combined freely.

Filter Controls

Cutoff — the frequency where filtering begins

Resonance — boosts frequencies around the cutoff

Drive — adds gain and harmonic distortion

5 > — Filter Character
Choose between:

- **6581** — gritty, unpredictable, full of personality
- **8580** — cleaner, more stable

6 > — Full Sweep
When automation is active, Full Sweep allows the cutoff to travel the entire knob range for dramatic sweeps.

7 > — Hard Sync
Recreates the SID's hard sync behavior using Osc 1 and 2.
Produces sharp, metallic, aggressive tones.

8 > — Ring Mod
Multiplies the waveforms of Osc 1 and 2 to create bell-like or metallic textures.
Automatically sets Osc 1 to Triangle.

9 > — Bitcrusher
Reduces bit depth from 12-bit down to 4-bit for classic lo-fi grit.



Above – A fully 'Bitcrushed' triangle wave, in its 'lo-fi', glitchy glory.

10 > — Model
SIDSynth has two modes: SID and Layer. SID mode stays closer to the original chip by limiting features the SID could not do natively:

- All four filter types can still be selected, but the sound passes through one shared filter setup: the low-pass cutoff, automation, resonance, and drive. The other filter setups are greyed out.
- Oscillator sustain snaps to 16 steps, and a small indicator appears under the sustain knobs. Layer mode keeps the fine value until you move the control.
- Oscillator morph and morph rate are unavailable, and those controls are greyed out.
- Per-waveform layers are unavailable, and that panel is greyed out. Volume stays at 100%, harmonics stay at 0, and noise color stays on the original SID noise.

In Layer mode, those expanded options come back, including the 5–95% pulse-width range.

What SID mode does not do

Polyphony, velocity, aftertouch, and many other modern features remain. SID mode limits the sound, not the whole instrument.

11 > — Route

Route chooses which oscillators pass through the filter. On filters the selected oscillator. Off bypasses it.

Filter Automation



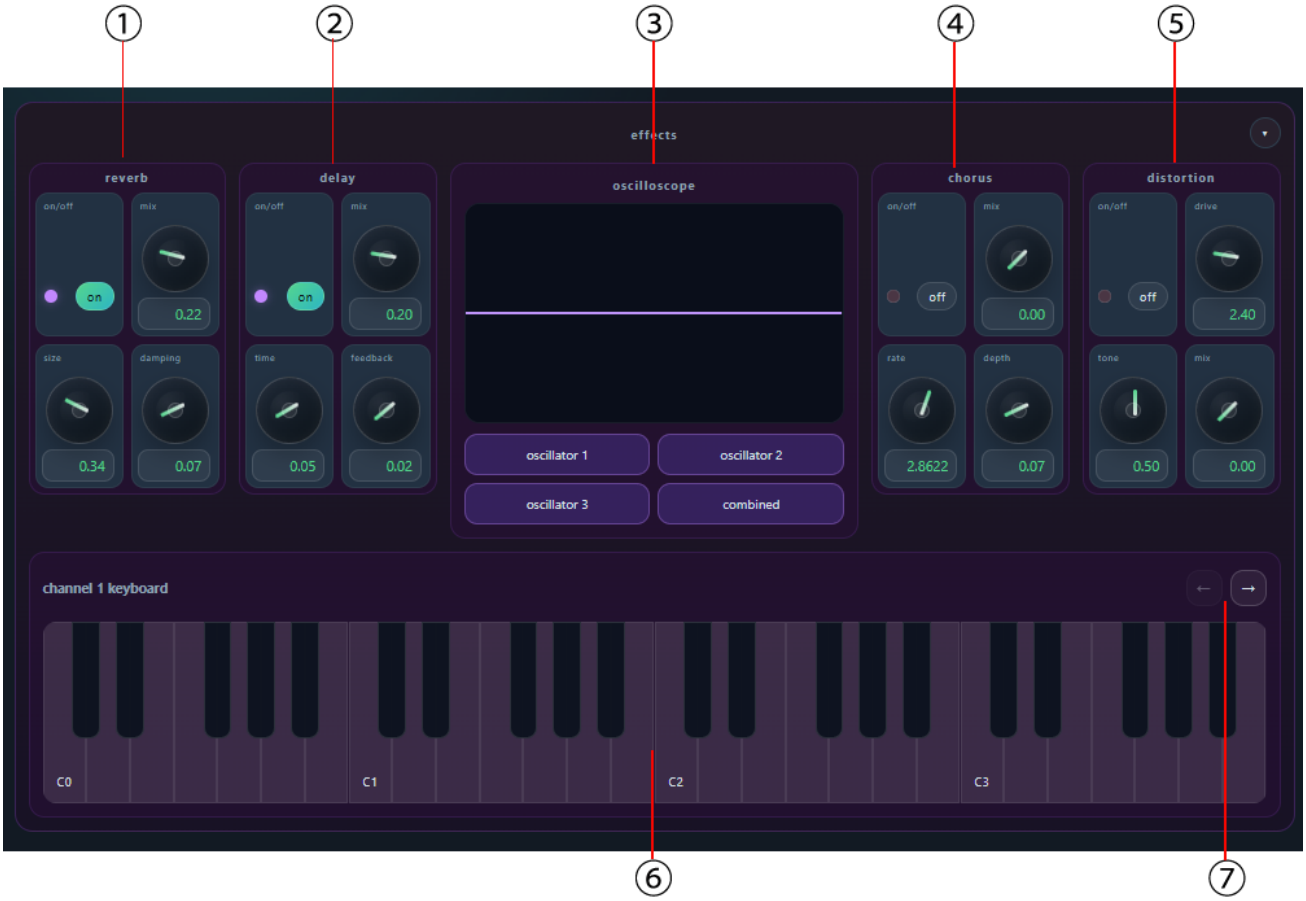
Filters can automatically sweep their cutoff frequency in a ping-pong motion.

1 > Automation On/Off

2 > Cutoff knob glows green when automation is active

All filters can be automated simultaneously, each with its own rate.

3.6 Effects



SIDSynth includes several post-filter effects to enhance and shape your sound.

1 > — Reverb
Adds space and depth.

- **Mix** — wet/dry balance
- **Size** — room size
- **Damping** — high-frequency decay

2 > — Delay
Repeats the sound over time.

- **Time** — delay length
- **Feedback** — number of repeats
- **Mix** — wet/dry balance

3 > — Oscilloscope
Displays the waveform of each oscillator (pre-effects) and the combined output (post-effects).
Great for visualizing modulation, morphing, and filtering.

4 > — Chorus
Thickens the sound by simulating multiple slightly detuned voices.

- **Rate** — LFO speed
- **Depth** — modulation intensity
- **Mix** — wet/dry balance

5 > — Distortion
Adds harmonic saturation and clipping.

- **Drive** — distortion amount
- **Tone** — harmonic emphasis
- **Mix** — wet/dry balance

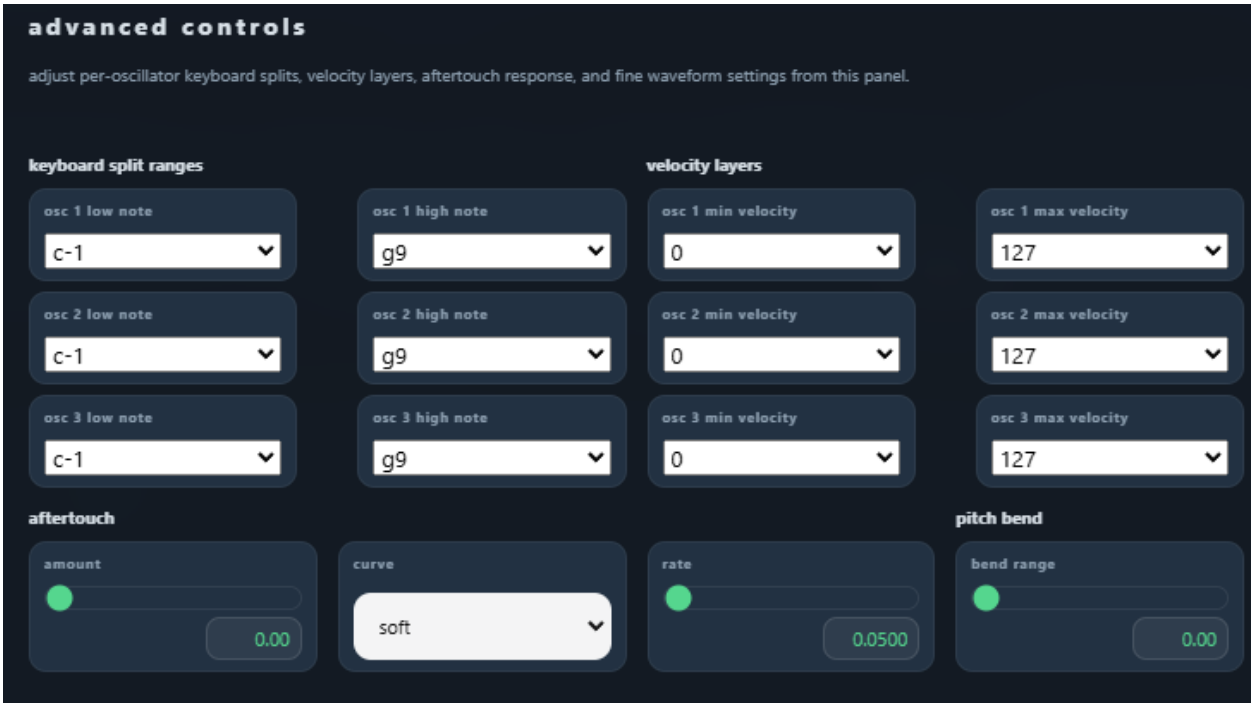
6 > — Integrated Keyboard
A built-in playable keyboard.
Oscillators must be set to **MIDI channel 1** to respond.
Use the arrows (7 >) to shift across the full 8-octave range.

Oscilloscope



Waveforms from oscillators 1, 2, and 3, each in that oscillator's theme color, combine into the purple trace after the filters and effects.

4. SIDSynth Advanced Menu



Advanced controls shape how SIDSynth responds to the keyboard, how layers interact, and how modulation behaves. Open the menu from the bar, then work through the sections below.

4.1 Keyboard Split Ranges

Keyboard Split Ranges allow you to define exactly which portion of the keyboard triggers each oscillator. This makes it possible to perform multiple roles simultaneously — for example:

- A heavy bass patch mapped to the lowest keys
- A bright, cutting lead mapped to the upper range
- Overlapping layers for stacked or blended timbres

Splits may be assigned anywhere across the full **127-note MIDI range (C-1 to G9)**. Each oscillator can have its own independent range, enabling complex multi-zone performance setups.

4.2 Velocity Layers

Velocity Layers determine how SIDSynth responds to playing dynamics — specifically, how hard the keys are struck. This allows different sounds or layers to appear based on performance intensity.

Velocity Layers can be freely combined with Keyboard Split Ranges, giving you full control over how your keyboard interacts with SIDSynth. Soft playing might trigger a mellow pad, while harder playing could introduce a sharper attack or an entirely different timbre.

4.3 Aftertouch

Aftertouch refers to pressure applied to a key *after* it has been pressed. In SIDSynth, Aftertouch is typically used to introduce vibrato or other expressive modulation.

Three parameters shape its behavior:

- **Amount** — Controls how strongly the effect is applied.
- **Curve** — Determines how much pressure is required to reach full modulation depth.
- **Rate** — Sets the vibrato speed, from 0.05 Hz to 20 Hz. The default is 6 Hz.

Curve options include:

- **Linear** — Standard aftertouch response.
- **Soft** — Requires more pressure to reach maximum vibrato; offers finer control.
- **Hard** — Reaches full vibrato quickly with minimal pressure.

These settings allow performers to tailor aftertouch response to their playing style.

4.4 Pitch Bend

Bend Range sets how far held notes can move, from 0 (no bend) to 24 semitones, two octaves in either direction. The default is 2 semitones. Pitch-bend enable stays on the main MIDI row.

4.5 Per Waveform Layers



Per-Waveform Layers provide additional control when blending multiple SID-style waveforms within a single oscillator. This feature allows you to sculpt complex timbres by adjusting the **relative volume**, **harmonic content**, and **noise coloration** of each waveform type.

Waveform Volume — [Waveform] vol

Sets the **relative volume** of the selected waveform. For example, adding **10–20% noise** to a waveform can introduce subtle grit or texture.

Relative volume means the value is scaled against the oscillator's own volume setting.

Waveform Harmonics — [Waveform] hmc

Adjusts the pitch of the selected waveform to match a harmonic interval chosen from the dropdown menu. A value of **0** preserves the oscillator's original pitch. Other values shift the waveform to one of the **six most common harmonics**, expressed in semitone format.

As with volume, harmonic selection is **relative to the oscillator's pitch**.

Noise Color

Noise does not respond to harmonic adjustments, since noise contains no pitched content. Instead, you may alter its **color**:

- **sid** — Classic SID-style noise
- **dark**
- **darker**
- **darkest**

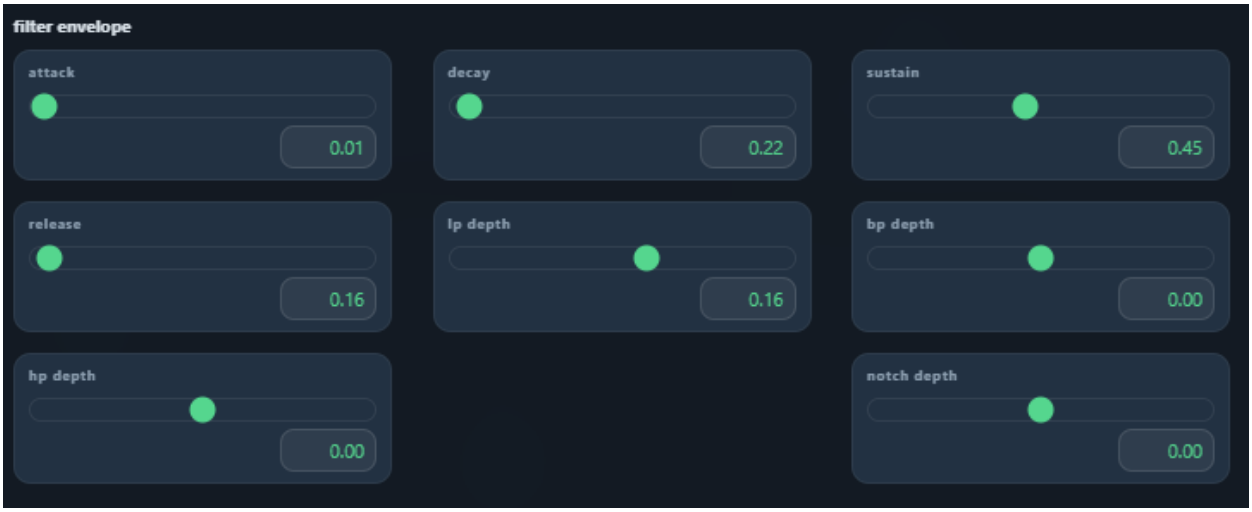
These darker variants provide alternative textures that may better suit certain sound-design goals.

4.6 Pulse Width Modulation



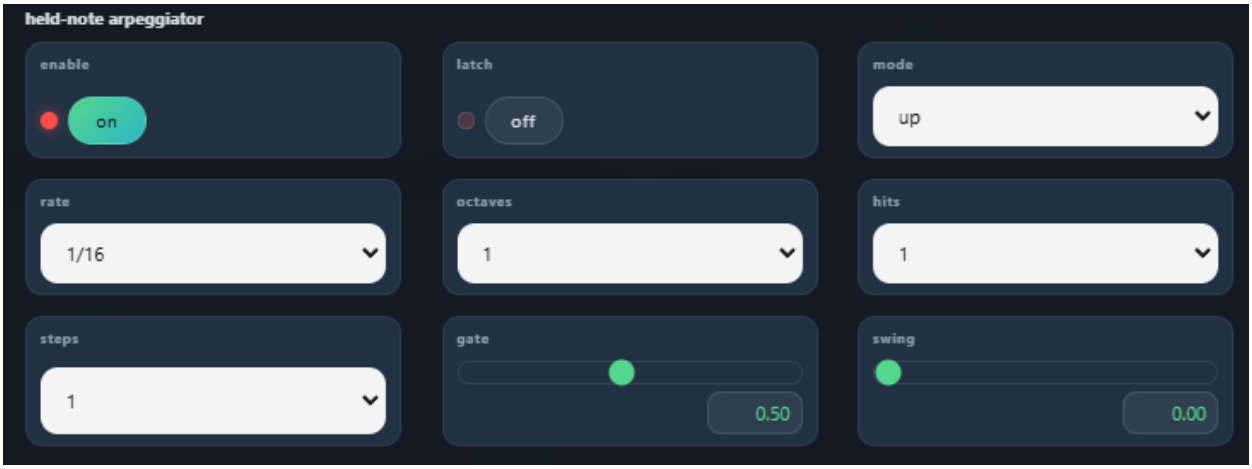
Each oscillator has its own pulse-width modulation, separate from the pulse-width knob on the main page. Raise that oscillator's PWM depth and the pulse width moves continuously. PWM rate sets the speed. Depth and rate are independent for each oscillator, and they do not follow the pitch LFO.

4.7 Filter Envelope



The filter envelope is a shared ADSR for the filters. It follows the newest note, but it moves filter cutoff instead of loudness. Use it for a short pop at the start of a note, a brass-like swell, or a quack bass. The ADSR shapes the contour. Each filter's depth control sets how strongly that filter follows it.

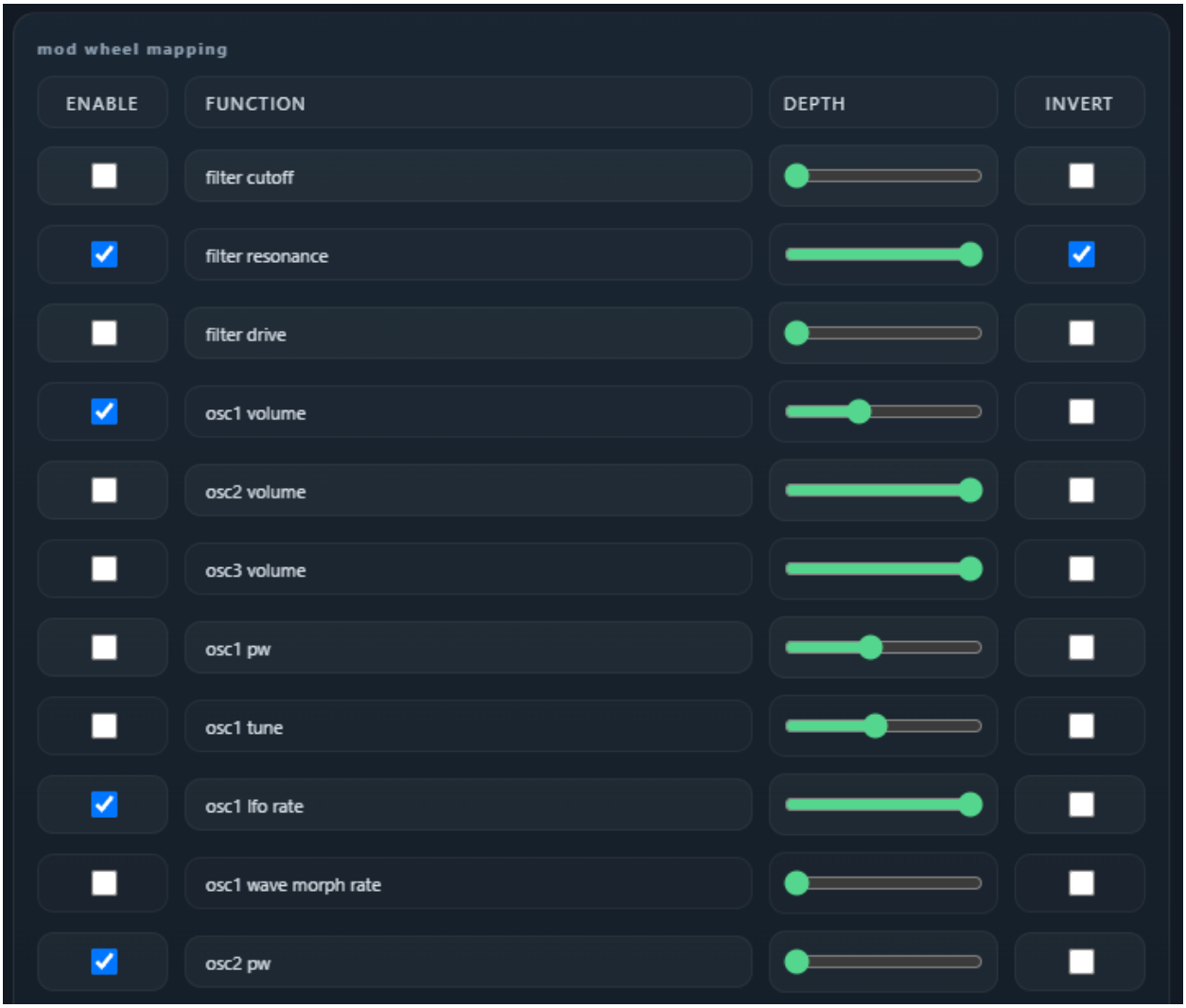
4.8 Held Note Arpeggiator



The held-note arpeggiator is in Advanced controls and defaults to off. When it is on, held notes are collected into a pattern of up to 16 notes instead of sounding together. Releasing the keys clears the pattern unless Latch is on.

- **Enable:** Turns the arpeggiator on or off. While it is on, the keyboard notes are played by the pattern.
- **Latch:** Keeps the current notes after you release them. The pattern continues until you play a new chord, which replaces it.
- **Mode:** Chooses the note order. Up and Down sort the held notes by pitch. Up-Down rises and falls. Played follows the order the notes were pressed. Random picks from the held notes. Root-Fifth-Octave builds a pattern from the lowest held note, a fifth, and the octave.
- **Rate:** Sets the step length to the host tempo. Choices run from a whole note to a 32nd note, including dotted and triplet values. The default is a 16th note. If the host does not publish a tempo, the rate follows Project BPM in the hamburger menu.
- **Octaves:** Repeats the pattern across 1, 2, or 3 octaves.
- **Steps:** Sets the pattern length from 1 to 16. The default is 16.
- **Hits:** Sets how many of those steps sound a note, from 1 to 16. Hits cannot be set above Steps. The sounding steps are spread across the pattern.
- **Gate:** Sets how long each hit is held, from a short blip to almost the full step. The default is halfway.
- **Swing:** Delays every other step. Leave it at zero for a straight pattern, and raise it for a looser feel.

4.9 Mod Wheel Mapping



Mod Wheel Mapping is one of SIDSynth's most expressive performance tools. Up to **24 different parameters** may be assigned to the mod wheel — *simultaneously* if desired — allowing sweeping, analog-style transformations with a single gesture.

Each mapping includes:

- **Enable** — Activates the modulation target.
- **Depth** — Sets the modulation amount, relative to the parameter's current value (similar to relative volume in waveform layers).
- **Invert** — Reverses the modulation direction.

This makes it possible to:

- Bend **osc1** tuning upward while bending **osc2** downward
- Fade out one oscillator while fading in another
- Increase filter resonance while decreasing cutoff
- Create evolving, layered motion with multiple simultaneous mappings

The number of possible combinations is enormous, making Mod Wheel Mapping a powerful tool for live performance and expressive sound design.

---

5. Numeric Parameter Reference

This section lists parameters that use a numeric value. Choice menus and on/off switches are left out, because they do not have a continuous range.

Oscillator 1

Parameter	Parameter ID	Range	Unit	Notes
Osc 1 Morph Rate	`osc1_wave_morph_rate`	0.0001 — 60.0	Hz	morph crossover speed between waveform components
Osc 1 Pan	`osc1_pan`	0.0 — 1.0	normalized	0 = left, 1 = right
Osc 1 Pan Automation Rate	`osc1_pan_automation_rate`	0.0001 — 10.0	Hz	pan automation speed
Osc 1 Volume	`osc1_volume`	0.0 — 1.0	normalized	amplitude gain
Osc 1 LFO Rate	`osc1_lfo_rate`	0.0001 — 250.0	Hz	LFO frequency
Osc 1 LFO Sweep Rate	`osc1_lfo_sweep_rate`	0.0001 — 60.0	Hz	LFO sweep speed
Osc 1 Pulse Width	`osc1_pw`	0.05 — 0.95	ratio	pulse width fraction
Osc 1 Tune	`osc1_tune`	-24.0 — 24.0	semitones	pitch offset

Oscillator 2

Parameter	Parameter ID	Range	Unit	Notes
Osc 2 Morph Rate	`osc2_wave_morph_rate`	0.0001 — 60.0	Hz	morph crossover speed
Osc 2 Pan	`osc2_pan`	0.0 — 1.0	normalized	0 = left, 1 = right
Osc 2 Tune	`osc2_tune`	-24.0 — 24.0	semitones	pitch offset
Osc 2 Pan Automation Rate	`osc2_pan_automation_rate`	0.0001 — 10.0	Hz	pan automation speed
Osc 2 Volume	`osc2_volume`	0.0 — 1.0	normalized	amplitude gain
Osc 2 LFO Rate	`osc2_lfo_rate`	0.0001 — 250.0	Hz	LFO frequency
Osc 2 LFO Sweep Rate	`osc2_lfo_sweep_rate`	0.0001 — 60.0	Hz	LFO sweep speed
Osc 2 Pulse Width	`osc2_pw`	0.05 — 0.95	ratio	pulse width fraction

Oscillator 3

Parameter	Parameter ID	Range	Unit	Notes
Osc 3 Morph Rate	`osc3_wave_morph_rate`	0.0001 — 60.0	Hz	morph crossover speed
Osc 3 Pan	`osc3_pan`	0.0 — 1.0	normalized	0 = left, 1 = right
Osc 3 Tune	`osc3_tune`	-24.0 — 24.0	semitones	pitch offset
Osc 3 Pan Automation Rate	`osc3_pan_automation_rate`	0.0001 — 10.0	Hz	pan automation speed
Osc 3 Volume	`osc3_volume`	0.0 — 1.0	normalized	amplitude gain
Osc 3 LFO Rate	`osc3_lfo_rate`	0.0001 — 250.0	Hz	LFO frequency
Osc 3 LFO Sweep Rate	`osc3_lfo_sweep_rate`	0.0001 — 60.0	Hz	LFO sweep speed
Osc 3 Pulse Width	`osc3_pw`	0.05 — 0.95	ratio	pulse width fraction

Filter controls

Parameter	Parameter ID	Range	Unit	Notes
Low-pass Cutoff	`filter_lp_cutoff`	0.0 — 1.0	normalized	normalized cutoff control
Low-pass Resonance	`filter_lp_resonance`	0.1 — 1.0	normalized	resonance/Q control
Low-pass Drive	`filter_lp_drive`	0.0 — 1.0	normalized	drive amount
Low-pass Automation Rate	`filter_lp_automation_rate`	0.0001 — 60.0	Hz	automation speed
Band-pass Cutoff	`filter_bp_cutoff`	0.0 — 1.0	normalized	normalized cutoff
Band-pass Resonance	`filter_bp_resonance`	0.1 — 1.0	normalized	resonance/Q
Band-pass Drive	`filter_bp_drive`	0.0 — 1.0	normalized	drive amount
Band-pass Automation Rate	`filter_bp_automation_rate`	0.0001 — 60.0	Hz	automation speed
High-pass Cutoff	`filter_hp_cutoff`	0.0 — 1.0	normalized	normalized cutoff
High-pass Resonance	`filter_hp_resonance`	0.1 — 1.0	normalized	resonance/Q
High-pass Drive	`filter_hp_drive`	0.0 — 1.0	normalized	drive amount
High-pass Automation Rate	`filter_hp_automation_rate`	0.0001 — 60.0	Hz	automation speed
Notch Cutoff	`filter_notch_cutoff`	0.0 — 1.0	normalized	normalized cutoff
Notch Resonance	`filter_notch_resonance`	0.1 — 1.0	normalized	resonance/Q
Notch Drive	`filter_notch_drive`	0.0 — 1.0	normalized	drive amount
Notch Automation Rate	`filter_notch_automation_rate`	0.0001 — 60.0	Hz	automation speed

Envelope controls

Parameter	Parameter ID	Range	Unit	Notes
Osc 1 Attack	`osc1_attack`	0.001 — 5.0	seconds	attack time
Osc 1 Decay	`osc1_decay`	0.01 — 10.0	seconds	decay time
Osc 1 Sustain	`osc1_sustain`	0.0 — 1.0	normalized	sustain level
Osc 1 Release	`osc1_release`	0.01 — 10.0	seconds	release time
Osc 2 Attack	`osc2_attack`	0.001 — 5.0	seconds	attack time
Osc 2 Decay	`osc2_decay`	0.01 — 10.0	seconds	decay time
Osc 2 Sustain	`osc2_sustain`	0.0 — 1.0	normalized	sustain level
Osc 2 Release	`osc2_release`	0.01 — 10.0	seconds	release time
Osc 3 Attack	`osc3_attack`	0.001 — 5.0	seconds	attack time
Osc 3 Decay	`osc3_decay`	0.01 — 10.0	seconds	decay time
Osc 3 Sustain	`osc3_sustain`	0.0 — 1.0	normalized	sustain level
Osc 3 Release	`osc3_release`	0.01 — 10.0	seconds	release time

Bitcrusher

Parameter	Parameter ID	Range	Unit	Notes
Bitcrusher	`bitcrusher_amount`	0.0 — 1.0	normalized / bits	maps internally to ~4–12 bits

MIDI & performance

Parameter	Parameter ID	Range	Unit	Notes
Aftertouch Amount	`midi_aftertouch_amount`	0.0 — 1.0	normalized	aftertouch depth
Aftertouch Rate	`midi_aftertouch_rate`	0.05 — 20.0	Hz	skew 0.35, step 0.01, default 6 Hz

Modwheel mapping depths

Parameter	Parameter ID	Range	Unit	Notes
Modwheel Filter Cutoff Depth	modwheel_filter_cutoff_depth	0.0 — 1.0	normalized	filter cutoff modulation amount
Modwheel Filter Resonance Depth	modwheel_filter_resonance_depth	0.0 — 1.0	normalized	resonance modulation amount
Modwheel Filter Drive Depth	modwheel_filter_drive_depth	0.0 — 1.0	normalized	drive modulation amount
Modwheel Osc1 Volume Depth	modwheel_osc1_volume_depth	0.0 — 1.0	normalized	Osc 1 volume modulation amount
Modwheel Osc2 Volume Depth	modwheel_osc2_volume_depth	0.0 — 1.0	normalized	Osc 2 volume modulation amount
Modwheel Osc3 Volume Depth	modwheel_osc3_volume_depth	0.0 — 1.0	normalized	Osc 3 volume modulation amount
Modwheel Osc1 PW Depth	modwheel_osc1_pw_depth	0.0 — 1.0	normalized	Osc 1 pulse width modulation amount
Modwheel Osc1 Tune Depth	modwheel_osc1_tune_depth	0.0 — 1.0	normalized	Osc 1 tune modulation amount
Modwheel Osc1 LFO Rate Depth	modwheel_osc1_lfo_rate_depth	0.0 — 1.0	normalized	Osc 1 LFO rate modulation amount
Modwheel Osc1 Wave Morph Rate Depth	modwheel_osc1_wave_morph_rate_depth	0.0 — 1.0	normalized	Osc 1 morph rate modulation amount
Modwheel Osc2 PW Depth	modwheel_osc2_pw_depth	0.0 — 1.0	normalized	Osc 2 pulse width modulation amount
Modwheel Osc2 Tune Depth	modwheel_osc2_tune_depth	0.0 — 1.0	normalized	Osc 2 tune modulation amount
Modwheel Osc2 LFO Rate Depth	modwheel_osc2_lfo_rate_depth	0.0 — 1.0	normalized	Osc 2 LFO rate modulation amount
Modwheel Osc2 Wave Morph Rate Depth	modwheel_osc2_wave_morph_rate_depth	0.0 — 1.0	normalized	Osc 2 morph rate modulation amount
Modwheel Osc3 PW Depth	modwheel_osc3_pw_depth	0.0 — 1.0	normalized	Osc 3 pulse width modulation amount
Modwheel Osc3 Tune Depth	modwheel_osc3_tune_depth	0.0 — 1.0	normalized	Osc 3 tune modulation amount
Modwheel Osc3 LFO Rate Depth	modwheel_osc3_lfo_rate_depth	0.0 — 1.0	normalized	Osc 3 LFO rate modulation amount
Modwheel Osc3 Wave Morph Rate Depth	modwheel_osc3_wave_morph_rate_depth	0.0 — 1.0	normalized	Osc 3 morph rate modulation amount
Modwheel Bitcrusher Amount Depth	modwheel_bitcrusher_amount_depth	0.0 — 1.0	normalized	bitcrusher modulation amount
Modwheel Distortion Drive Depth	modwheel_distortion_drive_depth	0.0 — 1.0	normalized	distortion drive modulation amount
Modwheel Distortion Mix Depth	modwheel_distortion_mix_depth	0.0 — 1.0	normalized	distortion mix modulation amount
Modwheel Delay Mix Depth	modwheel_delay_mix_depth	0.0 — 1.0	normalized	delay mix modulation amount
Modwheel Chorus Depth Depth	modwheel_chorus_depth_depth	0.0 — 1.0	normalized	chorus depth modulation amount
Modwheel Reverb Mix Depth	modwheel_reverb_mix_depth	0.0 — 1.0	normalized	reverb mix modulation amount

Master & effect controls

Parameter	Parameter ID	Range	Unit	Notes
Master Volume	`master_volume`	0.0 — 1.0	normalized	output gain
Master Portamento Rate	`master_portamento_rate`	0.0001 — 60.0	rate / time-scale	glide smoothing speed
Master Paraphony Rate	`master_paraphony_rate`	0.0001 — 60.0	Hz	paraphony modulation rate
Master Strobe Rate	`master_strobe_rate`	0.0001 — 20.0	Hz	strobe frequency
Master Strobe Gap	`master_strobe_gap`	0.0 — 1.0	normalized	strobe gap amount
Line Noise Volume	`line_noise_volume`	0.0 — 1.0	normalized	line noise level
Speaker Overdrive Volume	`speaker_overdrive_volume`	0.0 — 1.0	normalized	overdrive gain
Distortion Drive	`distortion_drive`	1.0 — 8.0	gain factor	distortion amount
Distortion Tone	`distortion_tone`	0.0 — 1.0	normalized	tone control
Distortion Mix	`distortion_mix`	0.0 — 1.0	normalized	wet/dry mix
Delay Mix	`delay_mix`	0.0 — 1.0	normalized	delay wet mix
Delay Time	`delay_time`	0.01 — 0.8	seconds	delay length
Delay Feedback	`delay_feedback`	0.0 — 0.95	normalized	feedback amount
Chorus Mix	`chorus_mix`	0.0 — 1.0	normalized	wet mix
Chorus Rate	`chorus_rate`	0.0001 — 5.0	Hz	chorus LFO rate
Chorus Depth	`chorus_depth`	0.0 — 1.0	normalized	chorus depth
Reverb Mix	`reverb_mix`	0.0 — 1.0	normalized	reverb wet mix
Reverb Size	`reverb_size`	0.1 — 1.0	normalized	room size
Reverb Damping	`reverb_damping`	0.0 — 1.0	normalized	damping amount

Notes on units

- A range of 0.0–1.0 is a normalized control, unless the name shows frequency, time, or semitones.
- Osc Tune is in semitones.
- Bitcrusher is a normalized control that maps to a fixed bit reduction internally.
- Filter cutoff controls are normalized in the interface. The heard frequency comes from the selected filter character.

6. MIDI Mapping

SIDSynth supports both **NRPN control** and **standard MIDI messages**, giving you flexible options for automation, hardware integration, and expressive performance.

6.1 NRPN Control

SIDSynth exposes its parameters through NRPN (Non-Registered Parameter Number) messages. NRPN gives you high-resolution control of every parameter in the index, from 0 through 272.

The NRPN message sequence is:

- **CC 99** — NRPN MSB
- **CC 98** — NRPN LSB
- **CC 6** — Data Entry MSB
- **CC 38** — Data Entry LSB
- **CC 96** — Increment
- **CC 97** — Decrement

This means any hardware controller or DAW that supports NRPN can automate SIDSynth with precision.

6.2 Standard MIDI

SIDSynth also responds to common MIDI performance controls:

- **CC 1 — Mod Wheel**
- **CC 64 — Sustain Pedal**
- **Pitch Bend**
- **Aftertouch**

These controls allow expressive playing and integrate seamlessly with most MIDI keyboards and controllers.

---

7. NRPN Parameter Index

NRPN parameter list

This list maps each NRPN index to its SIDSynth parameter.

Use NRPN MSB = CC 99 and LSB = CC 98 to select the 14-bit parameter index.

Use CC 6/38 or 96/97 for data entry.

NRPN Index	Parameter Constant	Parameter ID
0	osc1_wave_pulse	osc1_wave_pulse
1	osc1_wave_saw	osc1_wave_saw
2	osc1_wave_triangle	osc1_wave_triangle
3	osc1_wave_noise	osc1_wave_noise
4	osc1_pulse_volume	osc1_pulse_volume
5	osc1_pulse_octave	osc1_pulse_octave
6	osc1_saw_volume	osc1_saw_volume
7	osc1_saw_octave	osc1_saw_octave
8	osc1_triangle_volume	osc1_triangle_volume
9	osc1_triangle_octave	osc1_triangle_octave
10	osc1_noise_volume	osc1_noise_volume
11	osc1_noise_octave	osc1_noise_octave
12	osc1_wave_morph_enable	osc1_wave_morph_enable
13	osc1_wave_morph_rate	osc1_wave_morph_rate
14	osc1_enable	osc1_enable
15	osc1_voice_mode	osc1_voice_mode
16	osc1_pan	osc1_pan
17	osc1_pan_automation_enable	osc1_pan_automation_enable
18	osc1_pan_automation_rate	osc1_pan_automation_rate
19	osc1_volume	osc1_volume
20	osc1_midi_channel	osc1_midi_channel
21	osc1_low_note	osc1_low_note
22	osc1_high_note	osc1_high_note
23	osc1_min_velocity	osc1_min_velocity
24	osc1_max_velocity	osc1_max_velocity
25	osc1_lfo_enable	osc1_lfo_enable
26	osc1_lfo_rate	osc1_lfo_rate
27	osc1_lfo_shape	osc1_lfo_shape
28	osc1_lfo_sweep_enable	osc1_lfo_sweep_enable
29	osc1_lfo_sweep_rate	osc1_lfo_sweep_rate
30	osc1_pw	osc1_pw
31	osc1_tune	osc1_tune
32	osc2_wave_pulse	osc2_wave_pulse

NRPN Index	Parameter Constant	Parameter ID
33	osc2_wave_saw	osc2_wave_saw
34	osc2_wave_triangle	osc2_wave_triangle
35	osc2_wave_noise	osc2_wave_noise
36	osc2_pulse_volume	osc2_pulse_volume
37	osc2_pulse_octave	osc2_pulse_octave
38	osc2_saw_volume	osc2_saw_volume
39	osc2_saw_octave	osc2_saw_octave
40	osc2_triangle_volume	osc2_triangle_volume
41	osc2_triangle_octave	osc2_triangle_octave
42	osc2_noise_volume	osc2_noise_volume
43	osc2_noise_octave	osc2_noise_octave
44	osc2_wave_morph_enable	osc2_wave_morph_enable
45	osc2_wave_morph_rate	osc2_wave_morph_rate
46	osc2_enable	osc2_enable
47	osc2_voice_mode	osc2_voice_mode
48	osc2_pan	osc2_pan
49	osc2_tune	osc2_tune
50	osc2_pan_automation_enable	osc2_pan_automation_enable
51	osc2_pan_automation_rate	osc2_pan_automation_rate
52	osc2_volume	osc2_volume
53	osc2_midi_channel	osc2_midi_channel
54	osc2_low_note	osc2_low_note
55	osc2_high_note	osc2_high_note
56	osc2_min_velocity	osc2_min_velocity
57	osc2_max_velocity	osc2_max_velocity
58	osc2_lfo_enable	osc2_lfo_enable
59	osc2_lfo_rate	osc2_lfo_rate
60	osc2_lfo_shape	osc2_lfo_shape
61	osc2_lfo_sweep_enable	osc2_lfo_sweep_enable
62	osc2_lfo_sweep_rate	osc2_lfo_sweep_rate
63	osc2_pw	osc2_pw
64	osc3_wave_pulse	osc3_wave_pulse
65	osc3_wave_saw	osc3_wave_saw
66	osc3_wave_triangle	osc3_wave_triangle
67	osc3_wave_noise	osc3_wave_noise
68	osc3_pulse_volume	osc3_pulse_volume
69	osc3_pulse_octave	osc3_pulse_octave
70	osc3_saw_volume	osc3_saw_volume
71	osc3_saw_octave	osc3_saw_octave
72	osc3_triangle_volume	osc3_triangle_volume
73	osc3_triangle_octave	osc3_triangle_octave
74	osc3_noise_volume	osc3_noise_volume

NRPN Index	Parameter Constant	Parameter ID
75	osc3_noise_octave	osc3_noise_octave
76	osc3_wave_morph_enable	osc3_wave_morph_enable
77	osc3_wave_morph_rate	osc3_wave_morph_rate
78	osc3_enable	osc3_enable
79	osc3_voice_mode	osc3_voice_mode
80	osc3_pan	osc3_pan
81	osc3_tune	osc3_tune
82	osc3_pan_automation_enable	osc3_pan_automation_enable
83	osc3_pan_automation_rate	osc3_pan_automation_rate
84	osc3_volume	osc3_volume
85	osc3_midi_channel	osc3_midi_channel
86	osc3_low_note	osc3_low_note
87	osc3_high_note	osc3_high_note
88	osc3_min_velocity	osc3_min_velocity
89	osc3_max_velocity	osc3_max_velocity
90	osc3_lfo_enable	osc3_lfo_enable
91	osc3_lfo_rate	osc3_lfo_rate
92	osc3_lfo_shape	osc3_lfo_shape
93	osc3_lfo_sweep_enable	osc3_lfo_sweep_enable
94	osc3_lfo_sweep_rate	osc3_lfo_sweep_rate
95	osc3_pw	osc3_pw
96	sync_enable	sync_enable
97	ring_mod	ring_mod
98	filter_character	filter_character
99	filter_full_sweep	filter_full_sweep
100	filter_lp_enable	filter_lp_enable
101	filter_lp_cutoff	filter_lp_cutoff
102	filter_lp_resonance	filter_lp_resonance
103	filter_lp_drive	filter_lp_drive
104	filter_lp_automation_enable	filter_lp_automation_enable
105	filter_lp_automation_rate	filter_lp_automation_rate
106	filter_bp_enable	filter_bp_enable
107	filter_bp_cutoff	filter_bp_cutoff
108	filter_bp_resonance	filter_bp_resonance
109	filter_bp_drive	filter_bp_drive
110	filter_bp_automation_enable	filter_bp_automation_enable
111	filter_bp_automation_rate	filter_bp_automation_rate
112	filter_hp_enable	filter_hp_enable
113	filter_hp_cutoff	filter_hp_cutoff
114	filter_hp_resonance	filter_hp_resonance
115	filter_hp_drive	filter_hp_drive
116	filter_hp_automation_enable	filter_hp_automation_enable

NRPN Index	Parameter Constant	Parameter ID
117	filter_hp_automation_rate	filter_hp_automation_rate
118	filter_notch_enable	filter_notch_enable
119	filter_notch_cutoff	filter_notch_cutoff
120	filter_notch_resonance	filter_notch_resonance
121	filter_notch_drive	filter_notch_drive
122	filter_notch_automation_enable	filter_notch_automation_enable
123	filter_notch_automation_rate	filter_notch_automation_rate
124	osc1_attack	osc1_attack
125	osc1_decay	osc1_decay
126	osc1_sustain	osc1_sustain
127	osc1_release	osc1_release
128	osc2_attack	osc2_attack
129	osc2_decay	osc2_decay
130	osc2_sustain	osc2_sustain
131	osc2_release	osc2_release
132	osc3_attack	osc3_attack
133	osc3_decay	osc3_decay
134	osc3_sustain	osc3_sustain
135	osc3_release	osc3_release
136	bitcrusher_amount	bitcrusher_amount
137	midi_velocity_enable	midi_velocity_enable
138	midi_aftertouch_enable	midi_aftertouch_enable
139	midi_aftertouch_amount	midi_aftertouch_amount
140	midi_aftertouch_curve	midi_aftertouch_curve
141	midi_pitch_bend_enable	midi_pitch_bend_enable
142	midi_mod_wheel_enable	midi_mod_wheel_enable
143	midi_sustain_pedal_enable	midi_sustain_pedal_enable
144	modwheel_filter_cutoff_enable	modwheel_filter_cutoff_enable
145	modwheel_filter_cutoff_depth	modwheel_filter_cutoff_depth
146	modwheel_filter_cutoff_invert	modwheel_filter_cutoff_invert
147	modwheel_filter_resonance_enable	modwheel_filter_resonance_enable
148	modwheel_filter_resonance_depth	modwheel_filter_resonance_depth
149	modwheel_filter_resonance_invert	modwheel_filter_resonance_invert
150	modwheel_filter_drive_enable	modwheel_filter_drive_enable
151	modwheel_filter_drive_depth	modwheel_filter_drive_depth
152	modwheel_filter_drive_invert	modwheel_filter_drive_invert
153	modwheel_osc1_volume_enable	modwheel_osc1_volume_enable
154	modwheel_osc1_volume_depth	modwheel_osc1_volume_depth
155	modwheel_osc1_volume_invert	modwheel_osc1_volume_invert
156	modwheel_osc2_volume_enable	modwheel_osc2_volume_enable
157	modwheel_osc2_volume_depth	modwheel_osc2_volume_depth
158	modwheel_osc2_volume_invert	modwheel_osc2_volume_invert

NRPN Index	Parameter Constant	Parameter ID
159	modwheel_osc3_volume_enable	modwheel_osc3_volume_enable
160	modwheel_osc3_volume_depth	modwheel_osc3_volume_depth
161	modwheel_osc3_volume_invert	modwheel_osc3_volume_invert
162	modwheel_osc1_pw_enable	modwheel_osc1_pw_enable
163	modwheel_osc1_pw_depth	modwheel_osc1_pw_depth
164	modwheel_osc1_pw_invert	modwheel_osc1_pw_invert
165	modwheel_osc1_tune_enable	modwheel_osc1_tune_enable
166	modwheel_osc1_tune_depth	modwheel_osc1_tune_depth
167	modwheel_osc1_tune_invert	modwheel_osc1_tune_invert
168	modwheel_osc1_lfo_rate_enable	modwheel_osc1_lfo_rate_enable
169	modwheel_osc1_lfo_rate_depth	modwheel_osc1_lfo_rate_depth
170	modwheel_osc1_lfo_rate_invert	modwheel_osc1_lfo_rate_invert
171	modwheel_osc1_wave_morph_rate_enable	modwheel_osc1_wave_morph_rate_enable
172	modwheel_osc1_wave_morph_rate_depth	modwheel_osc1_wave_morph_rate_depth
173	modwheel_osc1_wave_morph_rate_invert	modwheel_osc1_wave_morph_rate_invert
174	modwheel_osc2_pw_enable	modwheel_osc2_pw_enable
175	modwheel_osc2_pw_depth	modwheel_osc2_pw_depth
176	modwheel_osc2_pw_invert	modwheel_osc2_pw_invert
177	modwheel_osc2_tune_enable	modwheel_osc2_tune_enable
178	modwheel_osc2_tune_depth	modwheel_osc2_tune_depth
179	modwheel_osc2_tune_invert	modwheel_osc2_tune_invert
180	modwheel_osc2_lfo_rate_enable	modwheel_osc2_lfo_rate_enable
181	modwheel_osc2_lfo_rate_depth	modwheel_osc2_lfo_rate_depth
182	modwheel_osc2_lfo_rate_invert	modwheel_osc2_lfo_rate_invert
183	modwheel_osc2_wave_morph_rate_enable	modwheel_osc2_wave_morph_rate_enable
184	modwheel_osc2_wave_morph_rate_depth	modwheel_osc2_wave_morph_rate_depth
185	modwheel_osc2_wave_morph_rate_invert	modwheel_osc2_wave_morph_rate_invert
186	modwheel_osc3_pw_enable	modwheel_osc3_pw_enable
187	modwheel_osc3_pw_depth	modwheel_osc3_pw_depth
188	modwheel_osc3_pw_invert	modwheel_osc3_pw_invert
189	modwheel_osc3_tune_enable	modwheel_osc3_tune_enable
190	modwheel_osc3_tune_depth	modwheel_osc3_tune_depth
191	modwheel_osc3_tune_invert	modwheel_osc3_tune_invert
192	modwheel_osc3_lfo_rate_enable	modwheel_osc3_lfo_rate_enable
193	modwheel_osc3_lfo_rate_depth	modwheel_osc3_lfo_rate_depth
194	modwheel_osc3_lfo_rate_invert	modwheel_osc3_lfo_rate_invert
195	modwheel_osc3_wave_morph_rate_enable	modwheel_osc3_wave_morph_rate_enable
196	modwheel_osc3_wave_morph_rate_depth	modwheel_osc3_wave_morph_rate_depth
197	modwheel_osc3_wave_morph_rate_invert	modwheel_osc3_wave_morph_rate_invert
198	modwheel_bitcrusher_amount_enable	modwheel_bitcrusher_amount_enable
199	modwheel_bitcrusher_amount_depth	modwheel_bitcrusher_amount_depth
200	modwheel_bitcrusher_amount_invert	modwheel_bitcrusher_amount_invert

NRPN Index	Parameter Constant	Parameter ID
201	modwheel_distortion_drive_enable	modwheel_distortion_drive_enable
202	modwheel_distortion_drive_depth	modwheel_distortion_drive_depth
203	modwheel_distortion_drive_invert	modwheel_distortion_drive_invert
204	modwheel_distortion_mix_enable	modwheel_distortion_mix_enable
205	modwheel_distortion_mix_depth	modwheel_distortion_mix_depth
206	modwheel_distortion_mix_invert	modwheel_distortion_mix_invert
207	modwheel_delay_mix_enable	modwheel_delay_mix_enable
208	modwheel_delay_mix_depth	modwheel_delay_mix_depth
209	modwheel_delay_mix_invert	modwheel_delay_mix_invert
210	modwheel_chorus_depth_enable	modwheel_chorus_depth_enable
211	modwheel_chorus_depth_depth	modwheel_chorus_depth_depth
212	modwheel_chorus_depth_invert	modwheel_chorus_depth_invert
213	modwheel_reverb_mix_enable	modwheel_reverb_mix_enable
214	modwheel_reverb_mix_depth	modwheel_reverb_mix_depth
215	modwheel_reverb_mix_invert	modwheel_reverb_mix_invert
216	master_volume	master_volume
217	master_portamento_enable	master_portamento_enable
218	master_portamento_rate	master_portamento_rate
219	master_paraphony_rate	master_paraphony_rate
220	master_strobe_enable	master_strobe_enable
221	master_strobe_rate	master_strobe_rate
222	master_strobe_gap	master_strobe_gap
223	line_noise_enable	line_noise_enable
224	line_noise_freq	line_noise_freq
225	line_noise_volume	line_noise_volume
226	speaker_overdrive_enable	speaker_overdrive_enable
227	speaker_overdrive_volume	speaker_overdrive_volume
228	distortion_enable	distortion_enable
229	distortion_drive	distortion_drive
230	distortion_tone	distortion_tone
231	distortion_mix	distortion_mix
232	delay_enable	delay_enable
233	delay_mix	delay_mix
234	delay_time	delay_time
235	delay_feedback	delay_feedback
236	chorus_enable	chorus_enable
237	chorus_mix	chorus_mix
238	chorus_rate	chorus_rate
239	chorus_depth	chorus_depth
240	reverb_enable	reverb_enable
241	reverb_mix	reverb_mix
242	reverb_size	reverb_size

NRPN Index	Parameter Constant	Parameter ID
243	reverb_damping	reverb_damping
244	voice_model	voice_model
245	osc1_filter_route	osc1_filter_route
246	osc2_filter_route	osc2_filter_route
247	osc3_filter_route	osc3_filter_route
248	osc1_pwm_depth	osc1_pwm_depth
249	osc2_pwm_depth	osc2_pwm_depth
250	osc3_pwm_depth	osc3_pwm_depth
251	filter_env_attack	filter_env_attack
252	filter_env_decay	filter_env_decay
253	filter_env_sustain	filter_env_sustain
254	filter_env_release	filter_env_release
255	filter_env_lp_depth	filter_env_lp_depth
256	filter_env_bp_depth	filter_env_bp_depth
257	filter_env_hp_depth	filter_env_hp_depth
258	filter_env_notch_depth	filter_env_notch_depth
259	arp_enable	arp_enable
260	arp_mode	arp_mode
261	arp_rate	arp_rate
262	arp_octaves	arp_octaves
263	arp_gate	arp_gate
264	arp_swing	arp_swing
265	arp_latch	arp_latch
266	arp_hits	arp_hits
267	arp_steps	arp_steps
268	midi_pitch_bend_range	midi_pitch_bend_range
269	osc1_pwm_rate	osc1_pwm_rate
270	osc2_pwm_rate	osc2_pwm_rate
271	osc3_pwm_rate	osc3_pwm_rate
272	midi_aftertouch_rate	midi_aftertouch_rate

Notes

Pitch bend range is 0-24 semitones, default 2. PWM rate is an independent sine, 0.01-20 Hz. Aftertouch rate is 0.05-20 Hz, default 6 Hz, which matches the previous fixed vibrato.

Choice values are stored as normalized 14-bit NRPN data, the same way .sidbank files store them. Two-state choices use 0 and 16383. Longer choices use the choice index, normalized across the full 14-bit range.

Parameter ID	Choice index	Meaning
voice_model	0	SID
voice_model	1	Layer
osc1_filter_route, osc2_filter_route, osc3_filter_route	0	Off
osc1_filter_route, osc2_filter_route, osc3_filter_route	1	On
arp_enable, arp_latch	0	Off
arp_enable, arp_latch	1	On
arp_mode	0	Up
arp_mode	1	Down
arp_mode	2	Up-Down
arp_mode	3	Played
arp_mode	4	Random
arp_mode	5	Root-Fifth-Octave
arp_rate	0	1/1
arp_rate	1	1/2
arp_rate	2	1/4
arp_rate	3	1/8
arp_rate	4	1/16
arp_rate	5	1/32
arp_rate	6	1/4 dotted
arp_rate	7	1/8 dotted
arp_rate	8	1/16 dotted
arp_rate	9	1/8 triplet
arp_rate	10	1/16 triplet
arp_octaves	0	1
arp_octaves	1	2
arp_octaves	2	3
arp_hits, arp_steps	0	1
arp_hits, arp_steps	15	16

For arp hits and arp steps, choice index 0 is 1 and choice index 15 is 16. Hits cannot be set above steps.

8. Notes

- SIDSynth uses a WebView-based interface.
- It requires a VST3-compatible host.
- A standalone version is included, so you can play without a DAW. An ASIO-compatible sound card is recommended.
- Linux and macOS versions are planned.