

ElkTracker

A tracker-style groovebox for handheld Linux devices

User Manual — v1.5.1

September 2026

Contents

1. Installation and Activation
2. Tracker Basics
3. Your First Beat
4. Controls
5. Navigation
6. Screens
7. Instrument Engines
8. LFOs
9. MOD Commands
10. Note OFF
11. Phrase FX Commands
12. Phrase Length
13. Labels
14. Quick Navigation
15. Copy, Paste & Undo
16. Song END Blocks
17. Choke Groups
18. Swing
19. Scale Mode
20. Slices
21. Sample Editor
22. Track Screen
23. Status Bar
24. Files and Data Layout
25. Gamepad Configuration
26. MIDI Out
27. Credits

Installation and Activation

ElkTracker installs like any other PortMaster-style port: unzip it onto your SD card and it appears in your launcher. You can open the app, explore every screen, and build a project without activating anything. Activation is only needed to save, load, or export your work.

Installing on device

1. **Download** the latest `ElkTracker.zip` from your dashboard or the releases page.
2. **Extract** — place the contents so the folder structure on your SD card is:

```
/roms/ports/ElkTracker.sh
/roms/ports/elktracker/elktracker
```

3. **Boot** — start your device. ElkTracker appears in the **Ports** section of your launcher.

On first launch the app creates its data directories (`projects/` , `samples/` , `wavetables/` , `exports/`) inside the port folder, writes a starter wavetable (`BASIC-MORPH.wav`), and generates a small demo project so there's something to play with immediately. If no gamepad mapping exists yet, the button-mapping wizard opens automatically before anything else.

What works before you activate

Everything **except saving and file export** is open from the first boot: browsing every screen, building a full song, auditioning instruments, editing samples. Unlicensed rows are marked `[UNLICENSED]` on the Project screen.

Autosave on quit (to `AUTOSAVED.elk`) also only runs when licensed. An unlicensed session's work doesn't persist between launches.

Activation

On every boot, ElkTracker writes a file called `elktracker.id` to your data root. This is your device's unique code. To unlock the full version:

1. Go to `elktracker.io/activate` on any device with a browser.
2. Log in with your purchase email.
3. Upload the `elktracker.id` file from your SD card (in the folder you installed to).
4. Download `elktracker.lic` and copy it into the same folder as the binary.
5. Reboot, or go to **Settings** → **ACTIVATE**, press A to open the Activate screen, then press A again on **CHECK FOR LICENSE FILE** to load it without rebooting.

The Activate screen shows your licence status (`LICENSED TO <name>` or `UNLICENSED`) and your device code, in case you need to copy it manually.

Tracker Basics

If you've never used a tracker before, a few conventions run through this entire manual. Ten minutes here will make everything else click faster!

Everything is hexadecimal

Trackers grew up on 8-bit computers, and ElkTracker keeps that tradition: almost every number you'll see or type is a **hex byte**, written as two digits `00` – `FF` instead of decimal `0` – `255`. The digits `0-9` count as usual, then `A B C D E F` continue for ten through fifteen. So one hex digit covers 0–15, and two digits cover the full byte range:

Hex	Decimal	Hex	Decimal
<code>00</code>	0	<code>80</code>	128
<code>0F</code>	15	<code>A0</code>	160
<code>10</code>	16	<code>F0</code>	240
<code>40</code>	64	<code>FF</code>	255

You'll see hex in row numbers (a 128-row song counts `00` – `7F`), FX parameters, and most instrument fields. The value editor is built around it too: holding **A** and pressing **left/right** nudges a value by 1, while **up/down** nudges by 16, or exactly one hex digit. Scrub the low digit with left/right, the high digit with up/down.

Nibble pairs. A handful of commands (like `ARP`) pack two independent 4-bit values into one byte, by using the high hex digit and the low hex digit separately. `ARP 37` means "high nibble 3, low nibble 7", not the number 55.

Notes

Notes are written `<letter><octave>`, e.g. `C-4`, `A#3`, `G-9`. Octave numbers follow MIDI convention starting at `-1`, so **C-4 is MIDI note 60**, or what we usually just call "middle C". The full playable range is `C-1` to `G9`.

The song hierarchy

ElkTracker organises music in four nested layers, each one a grid of references to the layer below:

SONG	128 rows x 8 channels, each cell holds a CHAIN number
CHAIN	16 slots, each holds a PHRASE number + transpose
PHRASE	up to 16 steps, each holds a NOTE + INSTRUMENT + FX
INSTRUMENT	the actual sound engine and its parameters

A phrase is the melodic/rhythmic pattern: a single drum pattern, a melody or chord sequence, a bassline. A chain arranges phrases into a longer part: verse, chorus, whatever your instrument plays across a section. The song grid then arranges chains across time, one per channel, building the whole track. **R (Dive)** steps one layer deeper into whatever's under the cursor, and **L (Back)** steps back out. See Navigation for more explanation on how this works.

Timing: rows, ticks, and speed

Every phrase advances **4 rows per beat** (so a 16-step phrase is one bar of 4/4). Within each row, the engine subdivides further into **ticks**. This is what the Project screen's **SPEED** field controls (1–16 ticks per row, default 16). Ticks are the resolution that per-row effects like **RTG** (retrigger), **KIL** (delayed release), and **ARP** operate at. For example, an **RTG 04** means "retrigger every 4 ticks within this row," which only makes sense relative to however many ticks the row actually has.

The two FX lanes

Every phrase step has **two independent FX slots**, each a **command** (three letters, e.g. **VOL**) plus a **hex parameter** (**xx**). Both slots can be used simultaneously and stack, so a step with **0FS 40** in FX1 and **RTG 02** in FX2 both apply on that row. See Phrase FX Commands for the full command list.

Reading this manual

- **A+D-pad** means hold the A button and press a direction.
- **Select+X** means hold Select and press X.
- Field names are written as they appear on screen, e.g. **CUTOFF**, **SLCMOD**.
- Hex bytes always appear as two digits (**00**, **4A**, **FF**); anything else (frame counts, percentages, semitones) is written as plain decimal.

With that all in place, the Quick Start walks through building an actual beat!

Your First Beat

The fastest way to learn the workflow is to build some music! This quick-start guide walks through a simple drum pattern plus one melodic layer, using nothing but the gamepad. If a term here is unfamiliar, Tracker Basics covers it.

A brand-new project starts empty on the **Song** screen at row 00. If you're just trying out ElkTracker, No license is needed for any of this. Only saving requires activation (see Installation & Activation).

1. Build a drum kit

1. From the Song screen, cursor on row 00, channel 1. **Tap A** in the empty cell to insert chain 00, then press **R (Dive)** to jump into it.
2. You're now on the **Chain** screen, slot 0, all empty. **Tap A** the slot to insert phrase 00, then **R (Dive)** into it.
3. You're on the **Phrase** screen, 16 empty steps. Move the cursor to step 00's **NOT** column and **tap A**. This inserts C-4 (MIDI 60, the editor's starting note) and automatically fills in instrument 00 too.
4. With the cursor still on step 00's **INSTR** column, press **R (Dive)** to open Instrument 00.
5. The cursor lands on the **TYPE** field. Hold **A** and press **→** repeatedly until it reads **DRUM**.
6. You'll see 8 pad rows, C-4 through G-4. Move to the C-4 row and press **R** (or A-tap) to open the sample browser. Use **↑ ↓** to find a kick sample, **R** to preview it, **A** to load it onto the pad. Repeat on the E-4 row with a hi-hat or snare sample.
7. Press **L (Back)** to return to the Phrase screen. Your kick note from step 3 is already sitting on step 00.

2. Write the beat

You now have a 16-step phrase and a drum kit on instrument 00. Enter notes on the **NOT** column: Tap A on an empty step to insert whatever note you last used, then nudge it with **A+↔** (±1 semitone) if you need a different one.

A simple four-on-the-floor with off-beat hats:

Step	Note	Meaning
00, 04, 08, 0C	C-4	Kick on every beat
02, 06, 0A, 0E	E-4	Hi-hat on the off-beats

The **beat stripe** (a subtle background tint every 4th row) marks 00 04 08 0C at a glance. Since every step already has instrument 00 filled in from step 3 above, you only need to touch the NOT column.

3. Hear it

Press **L (Back)** twice to return to the Song screen (Phrase → Chain → Song). Press **Start** and playback begins from the cursor row and loops, since it's the only content in the song so far.

4. Add a melodic layer

1. Move the cursor to row 00, channel 2. **Double-tap A** quickly.
2. This finds the next *unused* chain automatically and fills it in, rather than reusing chain 00. A status message confirms which slot it picked (chain 01). Dive into it with **R**.

3. On the empty Chain **01**, **double-tap A** on slot **0** the same way to grab the next unused phrase. Dive **R** into it.
4. On the new Phrase, enter a note on step **00**'s NOT column (A-tap, then nudge with A+↑↓ for octave jumps or A+↔ for semitones. Try **C-3** for a bass note). Move to the **INSTR** column of that same step and **double-tap A** again to grab the next unused instrument (**01**). Dive (**R**) into it.
5. On Instrument **01**, hold A and press → once on the TYPE field to change it from Sampler to **Sub**. Sub is a basic 3-oscillator synth. The default saw wave is audible, but feel free to play with the parameters here by moving the cursor to them, holding A and pressing ↑↓ for large increments, ↔ for small increments. The info bar at the bottom will describe each parameter's function as you move the cursor to it.
6. Press **L (Back)** three times to return to the Song screen. Press **Start**. Both channels now play together, each advancing through their own chain independently.

5. Save it

Saving needs an activated license. Hold **Select** and press ← twice to reach the **Project** screen, move down to **NAME** to rename your project (A+↔ moves the cursor, A+↑↓ cycles characters), then down to **SAVE** and press A.

From here, Screens covers every screen in full, and Phrase FX Commands covers everything you can do to make a beat more interesting than four quarter notes.

Controls

Every screen in ElkTracker uses the same small vocabulary of button combinations. Screen-specific exceptions (a field that steps by 12 instead of 16, for example) are called out in that screen's own chapter.

Console	Keyboard	Action
D-pad	Arrow keys	Move cursor
A (hold) + D-pad	Z + arrows	Edit value ($\leftrightarrow \pm 1$, $\updownarrow \pm 16$ / $\pm$ octave)
A (tap)	Z	Insert last value in empty cell
B	X	Clear cell
R shoulder	Space	Dive in (song $\rightarrow$ chain $\rightarrow$ phrase $\rightarrow$ instrument)
L shoulder	Backspace	Back out
Select + $\rightarrow$ (Song/Chain/Phrase)	—	Dive into item at cursor (like R shoulder)
Select + $\leftarrow$ (Chain/Phrase/Instr)	—	Back out (like L shoulder)
Select + $\leftrightarrow$ (other screens)	1-8	Switch screen
Select + R (Dive)	—	Next chain / phrase / instrument (see Quick Nav)
Select + L (Back)	—	Previous chain / phrase / instrument
Start	Enter	Play / stop
Select + A	Ctrl+C	Copy cell (or set range anchor)
Select + B	Ctrl+X	Cut cell / cut range
R2	Ctrl+V	Paste
L2	Ctrl+Z	Undo (64 levels)
Select + L2	Ctrl+Shift+Z	Redo
Select + R2	—	Duplicate chain / phrase / instrument
Select + Start	Ctrl+P	PANIC — hard-stop all sound (while playing)
Select + Start (stopped)	—	Isolation play — see below
R (sample browser)	Space	Preview highlighted wav
$\leftrightarrow$ (sample browser)	—	Jump one full page up / down the list
R (waveform editor)	Space	Preview marked start..end region
—	Ctrl+S	Save project

Editing values: hold A and press a direction to change a value. Left/right adjust by 1, up/down adjust by 16 on most numeric fields (one full hex digit), though several screens use a different step for up/down where appropriate (chain transpose steps by 12 semitones, for instance). Holding a direction auto-repeats. A-tap on an empty cell inserts the last-used

value for that column.

Start plays from the cursor row when pressed on the **Song** screen, from every other screen it starts the song from row 00.

Isolation play (Select+Start when stopped): behaviour depends on the current screen:

Screen	Result	Status bar indicator
Song	Loops the currently selected row — all 8 channels play the chains on that row and restart them when done	↻ (circular arrow icon)
Chain	Plays the current chain on its own (channel 0, looping)	C
Phrase	Plays the current phrase on its own (channel 0, looping)	P
Any other screen	PANIC	—

Press Start alone to stop isolation playback. Select+Start *while playing* always triggers PANIC, regardless of screen.

Double-tap to create in next free slot: tap A twice quickly on the same empty cell to find the next unused chain (Song screen), phrase (Chain screen), or instrument (Phrase screen, instrument column) and fill the cell with it. The first tap still inserts the last-used value as usual; the second tap replaces it with the first unused slot. A status message confirms the slot number.

Navigation

ElkTracker uses a dive-down structure: **Song → Chain → Phrase → Instrument**, plus a few dive targets that open a dedicated editor screen directly. Two buttons carry almost all of it:

- **R (Dive)** goes deeper. From Song, it opens the chain under the cursor (an empty cell, or an END block, does nothing). From Chain, it opens the phrase in the current slot. From Phrase, it opens the instrument for the current step (if the step's INSTR column is empty, it opens whichever instrument you last had open). From Instrument: on a Sampler, R on SAMPLE opens the file browser and R on EDIT opens the waveform editor; on a Wavetable, R on WTABLE opens the wavetable browser; on a Drum kit, R on any pad row opens the sample browser. On the Project screen, R on LOAD opens the project browser. On Settings, R on any action row triggers it (including PANIC).
- **L (Back)** goes up one level. From Track, Mixer, Project, or Settings it returns to Song. From Help or Activate it returns to Settings. From the Sample (waveform editor) screen it returns to Instrument.

Select + → on the Song, Chain, or Phrase screen dives into the item under the cursor. If the cell is empty and no dive is possible, it falls back to switching to the next screen. **Select + ←** on the Chain, Phrase, or Instrument screen backs out one level (the same as pressing L). On any other screen, **Select + ↔** cycles through the 8 main screens. The Sample waveform editor, Help, and Activate screens are entered by diving, not by the screen switcher.

Select + R on the Chain, Phrase, or Instrument screen jumps to the next item in the parent list without going back up: next chain in the same song column, next phrase slot in the current chain, or next populated instrument. **Select + L** jumps to the previous item. See Quick Navigation for the wrap and selection behaviour, which differs slightly between chains/phrases and instruments.

Screens

ElkTracker has 8 main screens, cycled with Select+←→: Song, Chain, Phrase, Instrument, Track, Mixer, Project, Settings.

1 • Song

128 rows × 8 channels. Each cell holds a chain number (00 – 7F), an **END block** (EN), or is empty. Scrolling a cell's value up past 7F also lands on an END block, as an alternative to pressing B.

Beat stripes: every 4th row (rows 00 , 04 , 08 , 0C , ...) has a subtle background tint to help you count bars at a glance.

Header: the project name is shown at the top-left. A small accent-coloured dot appears to the right of the name when there are unsaved changes. When the cursor rests on a cell that holds a chain with a label, the label appears at the top-right.

Independent channel advancement (M8-style): each channel moves through its column on its own; they do not share a row position. When a channel finishes a chain it scans forward for the next populated cell in its column, skipping empty rows. If no more filled cells exist it wraps back to the first filled row and loops. Channels with different numbers of entries therefore cycle at different rates, creating evolving polyrhythm with no extra setup.

A small dot appears to the left of each cell to show that channel's current play position. Because channels advance independently, the dots scatter across different rows as the song plays.

END blocks: press **B on an empty cell** to place an END block (EN). When the sequencer reaches an END block, that channel stops permanently until Play is pressed again. B on an END block clears it back to empty. See Song END Blocks.

Song-row loop (Select+Start when stopped): on the Song screen, hold Select and press Start while playback is stopped to lock all 8 channels to the currently selected row and loop it continuously. Each channel plays the chain assigned to that row. When a chain finishes, it restarts from the same row rather than advancing. A small circular-arrow icon (↻) appears next to the play indicator in the status bar. Press Start alone to stop, or Select+Start while playing to trigger a PANIC (hard stop).

Right panel — live channel table: the right side of the Song screen shows a live CH / IN / NOTE readout for all 8 channels, updating in real time as the sequencer runs. Row brightness and background colour fade with the channel's current output level.

Right panel — waveform grid: below the channel table, a 3×3 grid of mini waveform displays shows roughly the last third of a second of audio for each channel (1–8) and the master output (M). The grid updates continuously during playback.

2 • Chain

16 slots. Each slot holds a phrase number (00 – FE) and a **transpose** value in semitones (–48 to +48; A+←→ steps by 1, A+↑↓ by 12 — a full octave). A chain plays its filled slots in order and loops back to slot 0 when it reaches the **first empty slot**, not the last filled one, so a gap partway through (slot 0 filled, slot 1 empty, slot 2 filled) makes slot 2 unreachable during playback. Keep chain slots contiguous from slot 0.

Beat stripes: like in the Song screen, every 4th slot (rows 0, 4, 8, C) is tinted to mark bars.

Chains can have an **8-character label**. navigate up from slot 0 to reach the name row and edit it with A+D-pad. The label appears at the top-right of the Song screen when the cursor sits on that chain, and next to the title on the Chain screen itself. See Labels for more details.

When the cursor is on a phrase slot that contains a phrase with a label, the phrase label appears at the top-right of the Chain screen.

Chain isolation (Select+Start when stopped): hold Select and press Start while stopped to play the current chain on its own (channel 0 only, looping). A **C** label appears next to the play indicator in the status bar. Press Start alone to stop.

3 • Phrase

16 steps (or fewer: see Phrase Length). Each step has:

- **NOT:** a note (C-1 – G9), OFF (note off), or --- (empty)
- **IN:** instrument index, or empty (uses whichever instrument last played on this channel)
- **FX1 / P1:** first FX command and its hex parameter
- **FX2 / P2:** second FX command and its hex parameter

Beat stripes: every 4th step (rows 0, 4, 8, C) is tinted to mark bar subdivisions and help align notes to the beat.

Navigate up from row 0 to reach the **LEN** header to set the phrase length. Navigate up once more to reach the phrase **name row** (same label editing as chains). The label appears at the top-right of the Chain screen when the cursor is on that phrase slot, and next to the title on the Phrase screen itself.

When the cursor is on the INSTR column of a step that has an instrument assigned, the instrument's name appears at the top-right. When the cursor is on an FX column with a command set, the full FX command name appears in the status bar hint area.

If a scale is active, entering a note snaps to the nearest in-scale note and steps between in-scale notes rather than plain semitones. Notes already on the grid are never altered by turning a scale on or off.

Phrase isolation (Select+Start when stopped): hold Select and press Start while stopped to play the current phrase on its own (channel 0 only, looping). A **P** label appears next to the play indicator in the status bar. Press Start alone to stop.

4 • Instrument

Fields depend on the instrument type (see Instrument Engines). TYPE is always the first field. Type-specific parameters appear in the left column. Most instruments share these parameters in the right column: VOLUME, PAN, CUTOFF, RES, SEND A, SEND B, SEMI (coarse transpose, ±24 semitones), SCALE, ROOT, CHOKE. Some example exceptions: **Drum** has no CUTOFF or RES (pads always play at full brightness), and **FINE** (sub-semitone tuning) only appears on Sampler and Wavetable.

Navigate up from the first field to reach the instrument **name row**. Edit with A+D-pad. The name is shown on all screens that reference the instrument. When the instrument is unused, the screen shows (EMPTY - EDIT TO CREATE) .

The instrument screen has **two pages**, shown as 1/2 / 2/2 at top right. Page 1 shows all type-specific and common parameters arranged in two columns. Navigate past the bottom of either column (or up past the name row) to flip to **Page 2**, which contains the **LFO 1**, **LFO 2**, and **MOD** sections (see LFOs and MOD Commands). A ~ to the left of a parameter means it is targeted by an LFO or MOD slot.

B on a parameter field resets it to its default value, with two exceptions: B on **TYPE** does nothing, and B on the **name row** clears the whole name rather than resetting a value. **Tapping A on the SAMPLE** (Sampler) or **WTABLE** (Wavetable) field opens the file browser directly, the same as pressing R/Dive on those fields. Tapping A on a **Drum** pad row does the same for that pad's sample.

In the file browser: ↑ ↓ moves one item at a time; ↔ jumps a page at a time. In the **sample** picker, R previews the highlighted file and pressing it again stops the preview, and A loads it. In the project, wavetable, data-root, and duplicate pickers, R confirms the selection instead of previewing. B or L backs up one folder, or closes the picker entirely if you're already at the top level.

The **status bar** at the bottom of the screen shows a short description of the currently highlighted parameter while on the Instrument screen.

5 • Track

Per-channel insert FX: one effect per channel with two parameters (P1/P2). Below that, the two send buses — Bus A (reverb) has two parameters. Bus B (delay) has three, including a FREE/SYNC toggle for the delay time. See **Track Screen** for the full effect list, parameter ranges, and sync options.

6 • Mixer

8 VU meters with per-channel volume faders. A+↔ nudges by 1, A+↑ ↓ by 16. Mute (A-tap on M) and solo (A-tap on S) per channel. B resets a fader to full, and also clears mute on the M row or solo on the S row. Fader levels are saved with the project; **mute and solo are not** — they reset every time you reopen the project.

7 • Project

- **NAME:** 8 characters. A+↔ moves the cursor, A+↑ ↓ cycles through space, A-Z, a-z, 0-9, and ! - _.
- **BPM:** 40-300 (A+↔ ±1, A+↑ ↓ ±10).
- **SPEED:** ticks per row, 1-16 (see **Tracker Basics**).
- **SWING:** shown as OFF or 1% - 49%. See **Swing**.
- **ROOT / SCALE:** the project's key and scale. See **Scale Mode**.
- **SAVE:** saves to the current filename. If the file already exists, an **OVERWRITE?** dialog appears.
- **SAVE NEW:** saves as a new file, appending -1, -2, ... up to -99 to the project name.
- **LOAD:** opens the project file browser.
- **NEW:** resets to a blank project.
- **EXPORT WAV:** renders the full mix to `exports/<name>.wav`.
- **STEMS:** renders each channel separately to `exports/<name>_ch1.wav` etc.
- **COLLECT:** copies every sample and wavetable actually referenced by an instrument in the song into `exports/<name>/`, along with a copy of the project file, making that folder self-contained for sharing.
- **QUIT:** exits to the OS. If you're licensed, it first autosaves to `AUTOSAVED.elk`.

SAVE, SAVE NEW, LOAD, EXPORT WAV, STEMS, and COLLECT all require an activated license — see **Installation & Activation**. Unlicensed rows are marked **[UNLICENSED]**.

8 • Settings

- **THEME:** 5 colour palettes: NIGHT, FOREST, AMBER, ICE, NEON (the default).
- **DATA ROOT:** where `projects/`, `samples/`, `wavetables/`, and `exports/` live. Changing it browses the whole filesystem; selecting a folder creates the data directories there if missing, writes the starter wavetable/sample if they aren't already present, and reloads the current project's assets from the new location.
- **PANIC:** hard-stop all sound.

- **HELP:** the full in-app manual, a condensed on-device reference.
- **ACTIVATE:** shows your licence status and device code; see Installation & Activation.
- **MAP BUTTONS:** the gamepad mapping wizard; see Gamepad Configuration.
- **MIDI OUT:** selects which detected MIDI device receives output from `MIDI` instruments (or `NONE`); see MIDI Out.
- **DIAGNOSTICS:** dive in to run a MIDI subsystem test and confirm your interface is detected. For more info on MIDI, see MIDI Out.

A DSP load meter at the bottom shows how much of your CPU budget the current project is using (green/amber/red).

Instrument Engines

Each of the 64 instrument slots can use one sound engine, chosen with the TYPE field. Nine engines are available: Sampler, Sub, Wavetable, Macro, Multi, Reso, Chip, Drum, and MIDI. The available instruments range from sample playback to professional synthesis DSP, plus output to external MIDI hardware.

How the sound engine works

Each channel has one voice, plus 3 extra "chord" voices for paraphonic CHD chords, and there's a shared pool of 16 voices used by all Drum instruments across every channel — so up to 16 drum hits can ring at once regardless of which channels trigger them.

Every voice shares one lowpass filter and one ADSR envelope shape:

- **Filter:** a single resonant lowpass, roughly 20 Hz–14 kHz. It's bypassed entirely when CUTOFF is at its maximum (`FF`) and RES is at `00` ; adjusting either parameter engages it. Touching the filter on a **Sampler or Drum** instrument also sums the output to mono, since the filter itself is mono, so keep CUTOFF at `FF` and RES at `00` on stereo samples you want to keep stereo.
- **Envelope:** attack, decay, and release each run 0–4 seconds depending on the ADSR value; not every engine uses it (see the per-engine notes below).
- **Panning** is equal-power: `00` = hard left, `80` = centre, `FF` = hard right.

A master limiter sits on the final mix to catch peaks, so you generally don't need to worry about clipping from stacking several loud instruments.

Shared parameters

Sampler, Sub, Wavetable, Macro, Multi, Reso, and Chip all have: VOLUME, PAN, CUTOFF, RES, SEND A, SEND B, SEMI (coarse transpose, ± 24 semitones), SCALE, ROOT, CHOKE.

Drum is the exception: no CUTOFF, RES or SEMI. **FINE** (sub-semitone tuning) only exists on Sampler and Wavetable.

MIDI has none of these as it has no local voice to apply them to (see MIDI Out).

Every instrument also has two **LFOs** and four **MOD slots** on page 2 of the Instrument screen. See LFOs and MOD Commands for more information on these parameters.

Sampler

Plays WAV files. Fields: SAMPLE (file slot), EDIT (opens the waveform editor — dive in with R), LOOP (`NONE` / `FWD` / `PING` / `ONCE` , default `ONCE`), START / LOOPST / END (frame offsets — END at `00` means "play to the end of the file"), SLCMOD / SLCRT (slice mode — see Slices), FINE (sub-semitone tuning, ± 100 cents). `NONE` and `ONCE` behave identically — both stop at the end marker.

Sampler has no amplitude envelope: a note-OFF or the end of a one-shot cuts the voice immediately rather than fading, in every loop mode.

Sub

Simple 3-oscillator subtractive synthesiser. Per oscillator: WAVE (Saw / Square / Tri / Sine / Noise / Off), SEMI (± 48), FINE, LEVEL. Saw and Square are band-limited; Tri and Noise are not. Shared filter (CUTOFF/RES) and ADSR envelope.

Wavetable

Scans 2048-sample single-cycle frames, up to 256 per file (longer files are truncated to whole frames and summed to mono on load). WT POS selects the frame (0-255, interpolated between neighbouring frames as it moves).

Slots 0-4 are built-in sine/square/saw/tri/pulse tables, and each of those is a **single frame**, so WT POS has no audible effect on them. Load your own multi-frame table to hear it scan.

Wavetable has a similar ADSR envelope to Sub. FINE tunes the pitch by ± 100 cents. R on the WTABLE field opens the wavetable browser.

Macro

48 algorithmic oscillator shapes (SHAPE field), including everything from classic waveshapes through FM, physical models, and granular textures. TIMBRE and COLOR morph each shape. The same ADSR envelope and filter as the other synths. Built on top of a modern classic DSP algorithm. Yeah, you know the one.

Multi

24 synthesiser models in one engine, selected by MODEL and shaped with TIMBRE, HARM, and MORPH (in that order on screen): two virtual-analog models, phase distortion, three 6-op FM variants, one 2-op FM, wave terrain, string machine, chiptune, waveshaping, grains, additive, wavetable, chords, speech, swarm, noise, particles, strings, modal, bass drum, snare, and hi-hat. On the percussive models, ENV D (decay) also drives the model's own internal decay time as well as the amplitude envelope. More classic DSP from the same place as Macro.

Reso

Modal / string resonator. MODEL selects one of 6 physical models (MODAL , SYMPS , STRNG , FMVOX , SYMPQ , STRRV). Parameters: STRUCT (harmonics), BRIGHT (brightness), DAMP (damping), POS (excitation position). Monophonic. You guessed it - more modern classic DSP.

Chip

4-channel chiptune oscillator for you video game nerds. Each channel (A, B, C, D) plays the instrument pitch with an independent waveform:

Wave	Description
SQR	Square wave, duty cycle set by PW (12%-50%, default ~31%)
P25	25% pulse (fixed duty: PW has no effect)
P12	12.5% pulse (fixed duty: PW has no effect)
TRI	Triangle
NES	NES-style staircase triangle

Wave	Description
SIN	Sine
NSE	Pitched LFSR noise
OFF	Channel silent

Channels B, C, and D also have a **SEMI** offset (± 48 semitones relative to channel A) and an independent **V** (volume) level. Channel A always plays at the root pitch. All four channels share a single **ADSR** amplitude envelope and the common CUTOFF, VOLUME, PAN, SEND A, SEND B, SEMI, SCALE, CHOKE parameters. it's worth noting that Chip's envelope decays and releases considerably more slowly than the shared ADSR used elsewhere for the same A/D/S/R values.

Drum

A polyphonic drum sampler with 8 independent pads, drawn from the shared 16-voice pool described toward the top of this page. The left column shows the 8 pads (notes C-4 through G-4) and the right column holds shared parameters (VOLUME, PAN, SEND A, SEND B, SCALE, ROOT, CHOKE).

Each pad has:

- **Sample:** the WAV file assigned to this pad. A+↔ scrolls through available samples. R (or A+tap) opens the sample browser. The pad name scrolls when the cursor is on that row.
- **Choke group:** A+↑↓ sets the choke group (0 = none, 1-8). Pads in the same group cut each other when triggered. Choke groups are per-pad, independent of the instrument-level CHOKE.

Pads always play at their sample's natural pitch. Chain transpose and phrase transpose have no effect, and notes outside C-4 – G-4 simply don't trigger.

Because Drum pulls from a separate voice pool rather than the per-channel voice, most phrase FX commands (PIT , OFS , REV , SLC , GLI , CHD) have no effect on it. This means pads are fixed-pitch, one-shot triggers by design. VOL , PAN , SDA , SDB , RTG , KIL , and note- OFF do work, releasing or adjusting every pad voice currently ringing on that channel.

Midi

Sends notes to external MIDI hardware. Fields:

- **CHAN:** MIDI output channel, 1-16.
- **BANK:** bank select MSB (CC 0), 1-128, or --- to skip.
- **PROG:** program change, 1-128, or --- to skip. Allows different phrase steps to trigger different presets on the receiving synth.
- **SCALE / ROOT:** scale quantize mode and root, same as other engines. Controls which notes the phrase editor steps through.

Chain-level transpose, the CHD chord FX command, and VOL all work: transpose shifts the outgoing MIDI note, CHD sends additional Note Ons for each chord tone, and VOL sets the Note On velocity (0x00-0xFF maps to MIDI velocity 0-127). Everything else (PAN, filter, sends, and most other phrase FX) has no effect. See MIDI Out for full setup, bank/program selection, and other info on MIDI functionality.

LFOs

Every instrument has **two LFOs** (LFO 1 and LFO 2), on **page 2** of the Instrument screen alongside the MOD slots. Each LFO has:

Field	Description
ASSIGN	Which parameter the LFO modulates: VOLUME, PAN, PITCH, CUTOFF, RES, WT POS, SEND A, SEND B, or OFF
SYNC	FREE : rate set by a hex byte, exponentially from 0.05 Hz (00) to 20 Hz (FF). SYNC : rate locked to the project BPM as a fraction
RATE	Free mode: hex speed (00 - FF). Sync mode: N / D — A+↔ changes N (1-8), A+↑↓ changes D (1, 2, 4, 8, 16, 32). E.g. 1 / 4 = one full LFO cycle per bar
DEPTH	Modulation amount (00 - FF)
RETRIG	ON (default) restarts the LFO phase on every new note trigger. OFF lets it run free.

The LFO waveform is a sine only. Parameters targeted by any LFO show a ~ on the left of the field label. B on an LFO field resets that whole LFO to its defaults.

DEPTH doesn't sweep the full range of every target; full depth (FF) means:

Target	Swing at DEPTH FF
VOLUME	±50% around the current value
PAN	±half the pan range
CUTOFF / RES	±half the filter range
WT POS	±half the wavetable's frame range
SEND A / SEND B	±half the send range
PITCH	±2 semitones — a vibrato, not a full sweep

If both LFOs target the same parameter, their effects **add together** — two full-depth LFOs on VOLUME can swing twice as far as one. LFO values are snapshotted at the moment a note triggers, so changing an LFO's settings mid-note doesn't affect the voice already ringing, only the next one.

MOD Commands

Each instrument has four **MOD slots** (MOD 1-4), on page 2 of the Instrument screen alongside the LFOs. Each slot is assigned to one parameter (VOLUME, PAN, PITCH, CUTOFF, RES, WT POS, SEND A, SEND B).

Use the M01 - M04 FX commands in a phrase to set the assigned parameter to a specific value:

Command	Parameter	Description
M01	xx	Write xx to the parameter assigned to MOD slot 1
M02	xx	Write xx to MOD slot 2's target
M03	xx	Write xx to MOD slot 3's target
M04	xx	Write xx to MOD slot 4's target

Example: Set MOD 1 → CUTOFF on an instrument, then use M01 40 in the phrase FX lane to instantly set the filter cutoff to 40. The change applies to the playing voice immediately and is also written back to the Instrument screen. This means **this is a real, saved edit to the instrument**, not a temporary override.

When the target is **PITCH**, the parameter is a signed byte like PIT (80 = -128, FF = -1, 01 = +1), unclamped by the SEMI field's usual ±24 range, so a stray M01 40 on a PITCH slot can jump a voice by 64 semitones!

MOD only reaches the instrument's main channel voice. It doesn't affect that channel's chord voices (CHD) or, on a Drum instrument, the shared pool of pad voices.

Parameters targeted by a MOD slot also display the ~ marker.

MOD and LFO can both target the same parameter simultaneously; in this instance the LFO modulates around whatever value the MOD command last wrote.

Note OFF

In the phrase note column, pressing **B on an empty cell** (---) inserts OFF . An OFF step releases the playing voice on that channel: it starts the release stage of the ADSR envelope on engines that have one (Sub, Wavetable, Macro, Multi, Chip, Reso), and stops immediately on Sampler and Drum. OFF also releases any active CHD chord voices and, on a Drum instrument, every pad voice currently ringing on that channel.

- B on an OFF cell clears it back to ---
- A-tap on an OFF cell replaces it with the last-used note
- A PRB command on the same step also gates the OFF — a failed probability roll skips the release just like it would skip a note.

Phrase FX Commands

Two FX slots per step, each a three-letter command plus a hex parameter (`00` - `FF`) (see Tracker Basics if hex bytes, signed bytes, and nibble pairs aren't familiar yet). Most commands act on the channel's currently-playing voice, so they need a note somewhere on that channel to have anything to affect - either on the same row or an earlier one. **On a Drum instrument**, most of these are inert, as pads use a separate voice pool instead of the channel voice. Only `VOL`, `PAN`, `SDA`, `SDB`, `RTG`, and `KIL` reach Drum pads. For more information on this, see the Drum section of Instrument Engines.

Command	Parameter	Description
<code>VOL</code>	<code>xx</code>	Set voice volume (<code>00</code> = silent, <code>FF</code> = full)
<code>PIT</code>	<code>xx</code>	Pitch offset in semitones (signed byte: <code>80</code> = -128, <code>FF</code> = -1, <code>01</code> = +1)
<code>KIL</code>	<code>xx</code>	Release the voice after <code>xx</code> ticks into the row. Only meaningful for <code>xx</code> less than the project's SPEED (ticks per row). Larger values never fire.
<code>HOP</code>	<code>xx</code>	Jump to phrase row <code>xx</code> (wraps to the phrase length). Using HOP on a row also cancels that row's normal end-of-chain advance
<code>TPO</code>	<code>xx</code>	Set the project's BPM directly from a hex value (<code>78</code> = 120, <code>A8</code> = 168). Values below <code>28</code> (40 decimal) are ignored. This permanently changes and saves the project's tempo , it isn't a temporary per-row override.
<code>RTG</code>	<code>xx</code>	Retrigger the voice every <code>xx</code> ticks. <code>00</code> retriggers every tick; values at or beyond SPEED are clamped down to one retrigger per row.
<code>PAN</code>	<code>xx</code>	Pan: <code>00</code> = hard left, <code>80</code> = centre, <code>FF</code> = hard right.
<code>SDA</code>	<code>xx</code>	Set send level to reverb bus.
<code>SDB</code>	<code>xx</code>	Set send level to delay bus.
<code>OFS</code>	<code>xx</code>	Sampler only: start playback at <code>xx</code> /256 of the start..end region. The break-chopping command!
<code>REV</code>	—	Sampler only: play backwards from the current position. Stack after OFS to reverse from a specific point.
<code>SLC</code>	<code>xx</code>	Sampler only: jump to slice point <code>xx</code> (0-based, works regardless of the instrument's SLCMOD setting). Playback ends at the next slice or the END marker. Combine with RTG + OFS to ratchet within a slice.
<code>ARP</code>	<code>xy</code>	Arpeggio: cycles root → root+x → root+y → root → ... as the row's ticks pass. <code>x</code> is the high nibble, <code>y</code> the low nibble, both unsigned 0-15. Set <code>y=0</code> for a 2-note arp. Mutually exclusive with GLI on the same step . If both are present, GLI wins and ARP is ignored.
<code>PRB</code>	<code>xx</code>	Probability gate: the whole step (note, OFF, HOP, TPO — everything on that row) only fires if a fresh roll lands below <code>xx</code> . <code>00</code> = never, <code>FF</code> = always, otherwise roughly <code>xx</code> /256 chance (<code>80</code> ≈ 50%). Two <code>PRB</code> commands on one step roll independently and both must pass.
<code>GLI</code>	<code>xx</code>	Glide / portamento: slide pitch from the previous note to this one over <code>xx</code> ticks. Smooth on Sampler, Sub, Wavetable, and Chip; pitch-quantised (semitone steps) on Macro, Multi, and Reso.
<code>M01</code> - <code>M04</code>	<code>xx</code>	Write <code>xx</code> to the parameter assigned to the matching MOD slot. See MOD Commands.

Command	Parameter	Description
CHD	xx	Play a paraphonic chord alongside the row's note. Up to 3 extra voices through the same instrument, sharing its filter and envelope. 00 = off (root only). See the chord table below. Works on every engine except Drum (including Sampler, which stacks extra pitched copies of the sample).

CHD tones glide along with GLI when both are used together, and are stopped by choke and by CHOKE groups just like the main voice (but they don't count toward a choke group themselves).

Chord shapes (CHD parameter)

The parameter is a hex byte, not a decimal count, so shapes past the 9th sit at 0A - 10, not 10 - 16 :

xx (hex)	Name	Intervals from root
01	MAJ	+4, +7
02	MIN	+3, +7
03	5TH	+7
04	MAJ7	+4, +7, +11
05	MIN7	+3, +7, +10
06	DOM7	+4, +7, +10
07	SUS2	+2, +7
08	SUS4	+5, +7
09	AUG	+4, +8
0A	DIM	+3, +6
0B	DIM7	+3, +6, +9
0C	MMAJ7	+3, +7, +11
0D	MAJ6	+4, +7, +9
0E	MIN6	+3, +7, +9
0F	ADD9	+4, +7, +14
10	MADD9	+3, +7, +14

CHD is a row-level command; add it to every row where the chord should play, same as ARP ; a note without CHD on the following row releases the previous chord's extra voices as usual.

Some cool FX command usage examples

Break-chopping a drum loop: on a Sampler with a breakbeat loaded, 0FS 40 starts playback a quarter of the way into the region; stack RTG 04 on the same step to ratchet that starting point every 4 ticks for a classic chopped-break stutter.

Probability hats: put a hi-hat note on every 8th-note step with PRB A0 in FX1 — roughly 60% of hits play, giving a human, non-mechanical feel without hand-editing a pattern.

Arpeggiated chord: CHD 01 (major) on a synth note plus ARP 04 on the same step arpeggiates through root, root+4, root, root+4 while the chord's own extra voices ring underneath — a quick way to get a lot of movement out of one note.

Phrase Length

Each phrase has its own step count (1–16). Navigate **up from row 0** to reach the **LEN** header; edit with A+↑↓ or A+←→ (both step by 1); B resets to 16. Steps beyond LEN never play; their row-number gutter dims to show they're out of range, the note/instrument/FX content underneath stays visible at full brightness if it's already there from a longer setting.

The LEN row sits outside the step grid, so copy, cut, paste, and delete all skip over it.

Polyrhythm tip: set channel 1's phrase to LEN 8 and channel 2's to LEN 12 for a 2:3 feel. All channels run independently (see Screens), so any combination works.

Labels

Chains, phrases, and instruments can each have an **8-character name**, accessible by navigating right to the top of the page on Chain, Phrase or Instrument.

While on the name row: **A+←→** moves the character cursor (wrapping across the 8 characters); **A+↑↓** cycles through the character set. There's no lowercase here, unlike the Project screen's NAME field. **B clears the whole name.**

The name appears at the **top-right** of the parent screen when the cursor is on that item - e.g. the chain label shows on the Song screen when you hover over that chain's cell, and the phrase label shows on the Chain screen when the cursor is on that phrase slot, and next to the title on the item's own screen too.

Quick Navigation

On the **Chain**, **Phrase**, or **Instrument** screen, hold **Select** and press **R** (Dive) or **L** (Back) to jump to the next or previous item in the parent list without going back up a level:

Screen	Select+R	Select+L
Chain	Next chain in the same song column	Previous chain
Phrase	Next phrase slot in the current chain	Previous phrase slot
Instrument	Next populated instrument	Previous populated instrument

This lets you audition or edit a sequence of chains or phrases quickly without navigating back to the Song or Chain screen each time.

Copy, Paste & Undo

Editing a beat almost always means repeating something with small changes: duplicating a fill, copying a chain to build a variation, moving a run of steps. ElkTracker's clipboard works cell-by-cell or as a selected range, on the **Song**, **Chain**, and **Phrase** screens.

Single cell

Combo	Action
Select+A	Copy the cell under the cursor
Select+B	Cut the cell under the cursor
R2	Paste

On Song this is one chain number; on Chain it's the whole step (phrase + transpose); on Phrase it's the whole step (note, instrument, both FX slots).

Selecting a range

1. Select+A on a cell sets the **anchor** there and shows MARK SET – MOVE TO SELECT, A TO COPY .
2. Move the cursor with the D-pad up or down to grow the range from the anchor to the selected row.
3. Select+A again copies the range; Select+B cuts it.

A range always copies **whole rows**. On Phrase, that's all six columns (note, instrument, both FX slots) regardless of which column your cursor was on; on Chain it's both columns (phrase + transpose).

Pasting a range starts at the cursor and simply stops if it would run past the bottom of the grid.

Plain **B** (without Select) cancels an active selection. The status bar confirms with SELECTION CLEARED .

Duplicating a whole slot

On the **Chain**, **Phrase**, or **Instrument** screen, Select+R2 opens a picker listing every *unused* slot of that type. Confirming copies the current chain/phrase/instrument into it and switches your view to the new copy (handy for "same drum pattern, now change the fill" without hand re-entering every step).

Undo / redo

Combo	Action
L2	Undo
Select+L2	Redo

Up to **64 steps** of history. Rapid edits (holding a value while scrubbing with A+D-pad, for example) coalesce into a single undo step rather than one per tick, so undo rewinds by *edit*, not by keystroke. History is cleared whenever you load or start a new project.

Song END Blocks

Press **B on an empty song cell** to place an **END block** (displayed as `EN`; scrolling a cell's value up past chain `7F` reaches the same thing).

When the sequencer reaches an END block on a channel, that channel stops **permanently**; it will not restart until Play is pressed again. Pressing B on an END block clears it back to empty.

END blocks are useful for stopping individual channels before others: e.g. place `EN` on channels 4 and 5 in a later row to let drums and bass loop while keys and arp go silent.

Two things worth knowing: an END block only stops a channel when playback **reaches** it while running. If you start playback directly on a row that already has `EN` in it, that channel doesn't stop, it just sits idle for 16 rows before scanning forward as usual. And because of independent channel advancement, each channel only checks for an END block when it's actually its turn to advance, so the exact moment you'll hear it stop depends on that channel's own pacing, not the global row counter.

Choke Groups

Set **CHOKE 1-8** on instruments to link them: when any voice in a group triggers, all other voices in the same group (on any channel) stop immediately. CHOKE 0 disables choke (default).

Classic use: assign open hi-hat and closed hi-hat to CHOKE group 1, so the closed hat cuts the open hat, exactly like a real drummer would play.

Drum pads have their own, separate choke groups, set per-pad rather than per-instrument - see the Drum section of Instrument Engines. A Drum instrument's own top-level CHOKE field doesn't affect its pads at all. Paraphonic CHD chord voices are stopped along with the main voice they belong to.

Swing

Set on the **Project screen**. The field shows OFF or 1% - 49% — OFF means straight timing, and higher percentages push the shuffle further.

Swing lengthens the first row of every pair and shortens the second by the same amount, so the pair's total duration never changes, only the feel of the subdivision inside it does.

Swing scales *all* tick timing within a row, not just where the row itself falls, so RTG retriggers, ARP steps, and GLI glides on a swung row are compressed or stretched right along with it. Swing is global — all 8 channels share it.

Scale Mode

Project screen: set ROOT (C-B) and SCALE — 19 scales, or OFF for fully chromatic:

MAJOR , MINOR , HARM MIN , MEL MIN , DORIAN , PHRYGIAN , LYDIAN , MIXOLYD , LOCRIAN , MAJ PENT , MIN PENT , BLUES , WHOLE ,
DIM W-H , DIM H-W , HUNG MIN , PHRYG DM , HIRAJOSH , IN-SEN

When a scale is active, entering a note in the phrase editor snaps to the nearest in-scale note and steps between in-scale notes rather than plain semitones; up/down moves whole octaves, still snapped to the scale.

Per-instrument SCALE field:

- PROJECT — inherit the project scale
- OFF — all 128 notes available (use this for drum kits and slice samplers)
- Own scale + ROOT — instrument uses its own key, ignoring the project scale

Scale mode only guides how new notes are entered - it has **no effect on playback**. Already-entered notes are never changed by turning a scale on, off, or to something else.

Slices

Up to **16 slice points** per Sampler instrument. Each slice marks a frame offset within the sample. Slices are kept automatically sorted by position.

Manual slice entry

1. Open the waveform editor (EDIT on a Sampler instrument)
2. Adjust the **START** marker to the desired slice position
3. Navigate to the **SLICES** row; press **A** to add a slice at the current START position
4. A+←→ selects a slice; A+↑↓ nudges the selected slice by the same fine/coarse step as the other markers. B removes it.

Autoslice

Navigate to **AUTOSLICE >** in the right column of the waveform editor and press A (or R) to open the modal:

Setting	Description
MODE: TRANSIENT	Detects drum hits by energy onset. Lower THRESH = more slices; higher = only sharp transients.
MODE: GRID (default)	Splits the start..end region into 4, 8, or 16 equal chunks.
APPLY	Runs the slicer and writes results (clears existing slices first).

Autoslice places the first slice at the START marker and works within the START..END region. It pushes to undo, so it's reversible.

Slice playback modes (SLCMOD on instrument)

Mode	Behaviour
OFF	Slices exist, but nothing auto-selects one. Trigger the instrument normally and it plays from the sample's usual START
NOTE	Incoming note selects the slice: the root note (SLCRT, default C-4) plays slice 0, root+1 plays slice 1, and so on. Notes below the root all play slice 0; notes past the last slice all play the last one. Voice always plays at neutral pitch regardless of which note was used. Great for mapping a drum kit across a keyboard.
CMD	Use SLC xx in phrases to select slices explicitly

SLC xx works regardless of SLCMOD. Even with SLCMOD set to OFF or NOTE , an explicit SLC command in a phrase still jumps to that slice. SLCMOD only controls what happens automatically when a note triggers the instrument.

Sample Editor

The waveform editor is where you trim, reverse, normalise, time-stretch, and slice a sample. Open it from a Sampler instrument's **EDIT** field (press R or A).

Waveform display (top): S = Start E = End (L = Loop start, only shown when the instrument's LOOP mode isn't NONE)

Slice markers shown as vertical lines, labelled 0-F

Markers are shown as **percentages** of the sample (the Instrument screen shows the same values as raw frame counts). START/LOOPST/END step by a small amount with A+↔ and a larger amount with A+↑↓, scaled to the sample's length.

Left column (navigate with ↑↓):

Row	A+↔ / A+↑↓	A-tap / B
START	Move start marker (fine/coarse)	— / reset to 0
LOOPST	Move loop-start marker	— / reset to 0
END	Move end marker	— / reset to 0
SLICES	A+↔ selects a slice, A+↑↓ nudges it	A = add at START / B = remove
TRIM	Any direction with A held runs it	A = trim sample to START..END, write new file
REVERSE	Any direction with A held runs it	A = reverse sample, write new file
NORMALZ	Any direction with A held runs it	A = normalise sample, write new file

Right column (↔ hops between columns):

Row	A+↔	A+↑↓	A-tap
FIT xBAR	Choose 1/2/4/8/16 bars	— (no effect)	Apply pitch-preserving stretch to fit that many bars at the project BPM
STRETCH %	Adjust percent by 1	Adjust percent by 10	Apply stretch (25–400%)
PAUL Xn	Adjust factor by 1	Adjust factor by 4	Apply paulstretch (2–32×)
AUTOSLICE >	—	—	Open the autoslice modal (also reachable with R/Dive)

R toggles preview of the start..end region. Press it again to stop early.

Trim, Reverse, Normalize, Stretch

All four write a **new** file to `samples/edited/` and repoint the instrument at it. **Original samples are never modified**, so you can always go back. Output is always 16-bit stereo at the sample's original rate, regardless of the source format.

Trim and all three **stretch** operations reset START/LOOPST/END to 0, since the old offsets no longer correspond to anything meaningful in the new file; existing slices are **not** remapped, so re-slice afterwards if you had any. Reverse and Normalize

don't change the sample's length or offsets, so your markers stay put.

FIT xBAR and **STRETCH %** use the vendored Signalsmith Stretch algorithm (transparent, pitch-preserving). **PAUL** uses an in-house paulstretch reimplementation, tuned for the smeary, ambient character paulstretch is known for rather than transparency. A status message reads `STRETCHING...` while a stretch runs; **FIT** computes its ratio from the sample's length and the project BPM, and shows `FIT RATIO OUT OF RANGE` instead of stretching if that computed ratio is too extreme for the algorithm (very short one-shots stretched to many bars, for instance).

Track Screen

Each of the 8 channels has one **insert effect** with two hex parameters (P1/P2), applied after the channel's instrument and before the mix. Below that sit the two global **send buses**, reverb and delay, shared by every instrument via their SEND A / SEND B levels.

Insert FX

Code	Effect	P1	P2
DRIV	Drive/clip	Gain, ×1-12	Output level, 25%-100%
CRSH	Bitcrush	Bit depth, 16 down to 1 — higher P1 means fewer bits , the opposite of most controls	Rate divisor, 1-64
CHOR	Chorus	Rate, 0.1-4 Hz	Depth (tap spacing, 8-30 ms)
DELY	Delay	Time, 10 ms-1 s	Feedback, 0-90%
FLNG	Flanger	Rate, 0.05-2 Hz	Feedback, 0-85%
PHAS	Phaser	Rate, 0.05-3 Hz	Feedback, 0-80%
LPF	Low-pass filter	Cutoff, 20 Hz-14 kHz	Resonance
HPF	High-pass filter	Cutoff, 20 Hz-14 kHz	Resonance
SAT	Tape saturation	Drive amount	Tone (HF rolloff, 2-16 kHz)

Default parameters are 80 / 40 for any newly-selected effect. **B resets the whole row** (effect type and both parameters) back to ---- and defaults, not just the highlighted field!

Send buses

- **Bus A (Diffusion reverb):** TIME, DAMP.
- **Bus B (Delay):** feedback (FDBK, up to 92%), plus a **FREE / SYNC** toggle for the delay time:
 - **FREE:** TIME is a free-running knob (00 - FF , roughly 20 ms-1.2 s).
 - **SYNC:** delay time is set as a musical fraction of a bar, **N/D**, where N is the numerator (1-8, adjusted with A+↔→) and D cycles through 32, 16, 16·, 8, 8·, 4, 4·, 2, 2·, 1 with A+↑↓. The · suffix is a dotted value (1.5× duration).
Example: 1/4 = one beat; 1/8· = a dotted eighth. Delay time updates in real time when BPM changes.

The delay bus feeds back into itself with some cross-channel mixing for a wide, "ping-pong-ish" spread rather than a strict hard-left/hard-right bounce.

Per-instrument send levels are set in the SEND A / SEND B fields on the Instrument screen, or "live" via the SDA / SDB phrase commands.

Status Bar

The narrow strip at the bottom of every screen shows:

- **Screen name** and current **BPM**
- **Transport symbol**: a filled triangle (▶) when playing, a filled square (■) when stopped. A small label appears next to the triangle to show the current isolation mode: **C** for chain loop (Select+Start on Chain screen), **P** for phrase loop (Select+Start on Phrase screen), or a **circular arrow icon** (↻) for song-row loop (Select+Start on Song screen when stopped).
- **Context hint**: a short description of the currently highlighted field. This shows on the Phrase screen (full FX command name when an FX column is selected), the Instrument screen (parameter description), and the Track screen (insert FX parameter name).
- **Battery indicator**: a small battery icon appears at the right end of the status bar when the device reports power state (checked roughly every 10 seconds). The icon turns accent-coloured when charging or when the battery is below 20%.

Toast messages: confirmations like `SAVED`, `UNDO`, or `SLICE ADDED` appear briefly in a small box near the bottom-right, floating above everything else on screen (including modals), and clear themselves after about 3 seconds.

Files and Data Layout

Everything lives under one **data root** (set in Settings; defaults to the current working directory).

```
projects/      .elk project files
samples/      WAV files for the sampler (subfolders supported, 4 levels deep)
wavetables/   WAV files for the wavetable synth (2048-sample frames, up to 256 per file)
exports/      rendered mixes, stems, and COLLECT output
samples/edited/ output from trim/stretch/normalize (auto-created)
```

User preferences (theme, data root, last project) are stored in `elktracker.cfg` in the data root's working directory.

Auto-save: if licensed, the project is saved to `AUTOSAVED.elk` on quit (not the current project filename, so your named saves are never silently overwritten) and `AUTOSAVED.elk` becomes the project that reopens next launch. If no project has ever been opened, the app generates a small demo project.

WAV import: 8/16/24/32-bit PCM and 32-bit float, mono or stereo, at any sample rate (no resampling happens on import — pitch is calculated relative to the file's own rate).

Export WAV: renders the full mix to `exports/<name>.wav`, always at 44.1 kHz / 16-bit / stereo regardless of your audio device's own rate. Length is estimated automatically (song length + 2 seconds) and capped at 20 minutes; a render that comes out completely silent is flagged with an `EXPORTED (SILENT?)` message.

Stems: renders each channel separately to `exports/<name>_ch1.wav` etc. Each stem still passes through the shared reverb/delay sends and the master limiter, so the eight stems won't sum back to an exact bit-match of the full mix.

COLLECT: copies every sample and wavetable actually reached by an instrument that's referenced from the song (not just sitting in an unused slot) into `exports/<name>/`, along with a copy of the project file, making that folder self-contained to hand to someone else or move to another machine.

MIDI.log: a diagnostic file written to the data root each time you run Settings → DIAGNOSTICS → MIDI TEST — see MIDI Out.

Gamepad Configuration

Button Mapping Wizard

ElkTracker opens the button mapping wizard automatically on first boot. You can also run it any time from **Settings → MAP BUTTONS**, which clears any existing mapping and starts over.

The wizard asks for 14 inputs, in this order:

1. **A** — Edit / Insert
2. **B** — Clear cell
3. **X** — a spare face button, defaulted to the same action as B (Clear)
4. **Y** — a spare face button, defaulted to the same action as A (Edit)
5. **L1** — Back out
6. **R1** — Dive in
7. **L2** — Undo
8. **R2** — Paste
9. **SELECT** — Switch screen
10. **START** — Play / stop 11–14. **UP, DOWN, LEFT, RIGHT** — d-pad directions

Press the corresponding physical button when prompted. If your device uses a **hat switch** for the d-pad, the wizard detects this automatically once it reaches the direction steps (11–14) and fills in all four at once (hat input is handled natively and doesn't need individual button mappings).

There's no cancel button mid-wizard; every gamepad press is recorded as whatever step you're on, so if you make a mistake, finish the wizard and run **MAP BUTTONS** again from Settings to redo it.

Once complete, the wizard writes `gamepad.cfg` to the data root as plain `key=index` lines (`a=0` , `b=1` , ..., or just 10 lines if a hat was detected), which you can also hand-edit directly if you know what you're doing and you'd rather skip the wizard.

MIDI Out

ElkTracker can drive an external synth or MIDI module by adding a **MIDI** instrument and sending its notes out over a class-compliant USB MIDI interface - useful for triggering hardware synths from your handheld.

Setup

1. Plug in a class-compliant USB MIDI interface. Most modern USB audio/MIDI interfaces need no extra driver on Linux.
2. **Settings** → **MIDI OUT** — A+↔ cycles through `NONE` and every detected port; pick yours. The selection is saved and reopens automatically the next time you launch.
3. **Settings** → **DIAGNOSTICS** — dive in, then A on `MIDI TEST >` to confirm ElkTracker can see your interface. A `KERNEL MODULES` section is shown for reference only — some devices build USB MIDI support directly into the kernel rather than as a loadable module, so those rows reading `n/a` is normal and doesn't mean anything's wrong. Trust the `MIDI PORTS` count and the verdict at the bottom (`MIDI READY` / `NO MIDI PORTS FOUND`) instead. The same report is written to `MIDI.log` in your data root for reference.

Creating a MIDI instrument

Open an instrument slot and cycle **TYPE** to `MIDI` (the last of the engine types). Five fields are available:

- **CHAN**: the MIDI output channel, 1-16.
- **BANK**: bank select MSB (CC 0), 1-128, or `---` to skip.
- **PROG**: program change (0xC0), 1-128, or `---` to skip.
- **SCALE**: scale quantize mode, same as other engines: see `Scale Mode`.
- **ROOT**: scale root note (C-B), active when **SCALE** is set to a specific scale (not `PROJECT` or `OFF`).

BANK and **PROG** are independent, so you can set either one without the other. Neither is sent if it is `---`.

Write notes into a phrase against this instrument exactly like any other engine. A note plus this instrument in the **INSTR** column sends a Note On on **CHAN**; a Note Off is sent first if the channel's previous MIDI note is still sounding.

Selecting a preset with **BANK** and **PROG**

Most synths that respond to bank select follow this sequence:

1. CC 0 value 0-127 (Bank Select MSB) - specifies bank group
2. CC 32 value 0 (Bank Select LSB) - ElkTracker always sends 0 here
3. Program Change value 0-127 - specifies preset within that bank

If your synth only has 128 presets in a single bank, set **BANK** to `---` and just use **PROG**. If it has more, check the synth manual for which Bank MSB value corresponds to the bank you want.

The bank/program change is sent on every note trigger, so you can switch presets mid-sequence by assigning different MIDI instruments to different phrase steps. Be careful when using this, though - spamming lots of fast program changes is a good way to lock up your MIDI interface! If this happens, just restart ElkTracker.

Usage example

1. Instrument 01 : TYPE → MIDI , CHAN → 1 , BANK → --- , PROG → 5 .
2. On channel 2 of a phrase, set INSTR to 01 and enter C-4 on row 00 .
3. Press Start; with an external synth listening on MIDI channel 1, it switches to preset 5 and plays the note in time with the rest of the project.

What works, and what doesn't

MIDI instruments have no built-in voice, so only the commands that make sense for triggering an external device do anything:

Works	No effect
Note trigger (NOTE column)	PAN, SDA, SDB
VOL → sets Note On velocity	PIT, OFS, REV, SLC
KIL (note off on tick)	GLI, ARP
RTG (retrigger)	LFO1, LFO2, MOD1-4
Note OFF	
CHD chord shapes	
Chain transpose	

A few things worth knowing:

- **VOL sets velocity.** Write a VOL FX command on the same row as a note to set the Note On velocity, and the 0x00-0xFF range maps to MIDI velocity 0-127. Without a VOL command, velocity defaults to 127. RTG retriggers on the same row also use the VOL value from that row.
- **Chain transpose** shifts the sent MIDI note number up or down by the chain step's transpose value, exactly like any other engine.
- **CHD** sends additional Note On messages for each chord tone on the same MIDI channel, and Note Offs when the note ends. The external synth needs to be polyphonic (or in poly mode) to sound all the voices.
- **SCALE / ROOT** control which notes the phrase editor steps through when entering notes, they don't re-quantize notes at playback time. Set to PROJECT to follow the song's global scale, or pick a specific scale and root for this instrument.

Panic

Select+Start (or **Settings → PANIC**) sends MIDI All Notes Off on every channel, alongside silencing local voices.

Credits

- **Macro/Multi/Reso engines + Bus A reverb:** DSP by Émilie Gillet (Mutable Instruments), MIT licence, vendored in `third_party/eurorack/`
- **Chip engine:** original DSP inspired by Mutable Instruments Edges
- **Timestretch:** Signalsmith Stretch by Geraint Luff (Signalsmith Audio), MIT licence, in `third_party/signalsmith-stretch/`
- **Paulstretch mode:** our own implementation of Paul Nasca's published algorithm
- **Bitmap font:** public domain (Daniel Hepper)

ElkTracker is an independent project — not affiliated with or endorsed by any of the above.

Big thanks to Nina for always being my inspiration.