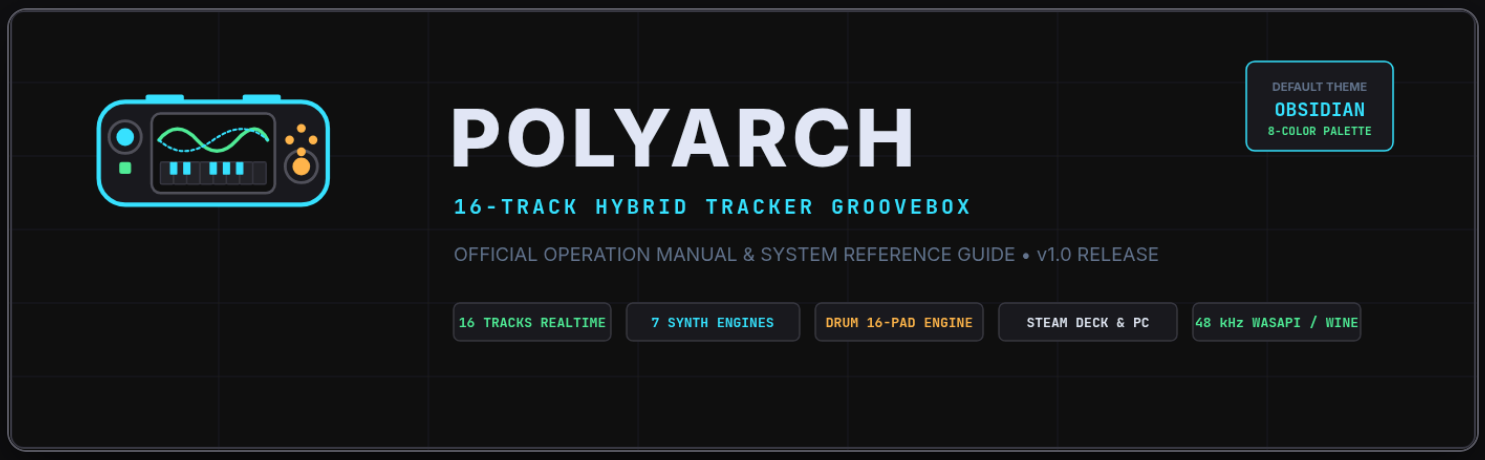
Polyarch — Operation Manual & Reference Guide



Version 1.0.0 (Beta.11 Certified)

Hardware & Desktop Reference for Windows PC, Steam Deck (SteamOS), and Handheld Consoles

> **TIP** **Dark-Themed Editions:** This manual is styled with a native dark aesthetic matching the Polyarch tracker console. You can view the full dark-themed HTML Manual or the 29-page publication-grade PDF User Guide.

Table of Contents

1. Operation Manual Conventions

2. Introduction & Overview

- 2.1 What is Polyarch?

- 2.2 Design Philosophy & Core Differences

- 2.3 System Requirements & Platform Support

3. Hardware Layout & Controller Mappings

- 3.1 The 4-Button Chorded Workflow

- 3.2 Steam Deck & Gamepad Layout

- 3.3 Desktop Keyboard Shortcuts & Direct Hex Entry

4. Getting Started & Core Concepts

- 4.1 Tracker Song Architecture

- 4.2 2D Spatial Screen Topology (The Minimap)

- 4.3 Loading the Demo Project

- 4.4 Hexadecimal Primer

5. Screen by Screen Walkthrough

- 5.1 Song View (16-Track Matrix)

- 5.2 Live Playback Mode

- 5.3 Chain View

- 5.4 Phrase View

- 5.5 Instrument View

- 5.6 Modulators View (Envelopes & LFOs)

- 5.7 Table View (Tick-Rate Micro-Sequencing)

- 5.8 Mixer View (16 Channel Strips & Peak VU Meters)

- 5.9 Track FX Insert Rack

- 5.10 Global FX Send Bus & Master Limiter

- 5.11 Project View (Tempo, Groove, and File Operations)

- 5.12 Theme & Mapping Views

6. Instrument Engines In-Depth

- 6.1 Engine 0: WAVE (Wavsynth)

- 6.2 Engine 1: FM (4-Operator FM Synth)

- 6.3 Engine 2: MACRO (Macrosynth)

- 6.4 Engine 3: DETUNE (Hypersynth)

- 6.5 Engine 4: SAMPLER (WAV Player & Live Microphone Recording)

- 6.6 Engine 5: MIDI OUT (Hardware & Virtual Routing)

- 6.7 Engine 6: DRUM (16-Pad Hybrid Machine & Choke Groups)

7. Hexadecimal Effect Command Reference

8. Audio Signal Flow & Realtime Architecture

9. Project Management & Offline Export

10. Appendix

- 10.1 Key & Controller Shortcut Cheat Sheet

- 10.2 Hexadecimal Conversion Table

- 10.3 Technical Specifications

---

1. Operation Manual Conventions

Throughout this manual, bracketed notation represents physical keys, controller buttons, or chorded combinations.

Notation	Meaning	Example
`[BUTTON]`	Press and release a single button.	`[EDIT]`, `[PLAY]`
`[BUTTON A]+[BUTTON B]`	Hold down Button A, then press Button B.	`[SHIFT]+[RIGHT]`, `[OPT]+[EDIT]`
`[BUTTON A, then BUTTON B]`	Press and release Button A, then press Button B.	`[SHIFT]+[OPT, then EDIT]`
`[DOUBLE TAP BUTTON]`	Rapidly press the button twice in succession.	`[DOUBLE TAP EDIT]`
`[DIRECTION]`	Any of the 4 directional inputs: Up, Down, Left, or Right.	`[UP]`, `[LEFT]`

---

2. Introduction & Overview

2.1 What is Polyarch?

Polyarch is a commercial 16-track hexadecimal music creation workstation and tracker groovebox engineered in Rust. Inspired by the rapid muscle-memory workflow of the **Dirtywave M8** and **LSDJ**, Polyarch expands the tracker paradigm to exploit the modern compute headroom and panoramic displays of PC handhelds (Steam Deck, ROG Ally, Legion Go) and desktop workstations.

Polyarch eliminates window management, floating modal dialogues, and mouse pointing. The entire creative workflow—from step sequencing, FM sound design, and sample slicing to live microphone recording, 16-channel mixing, and 3-band mastering—is navigated on a responsive vector monospace interface using an 8-button chorded controller or keyboard.

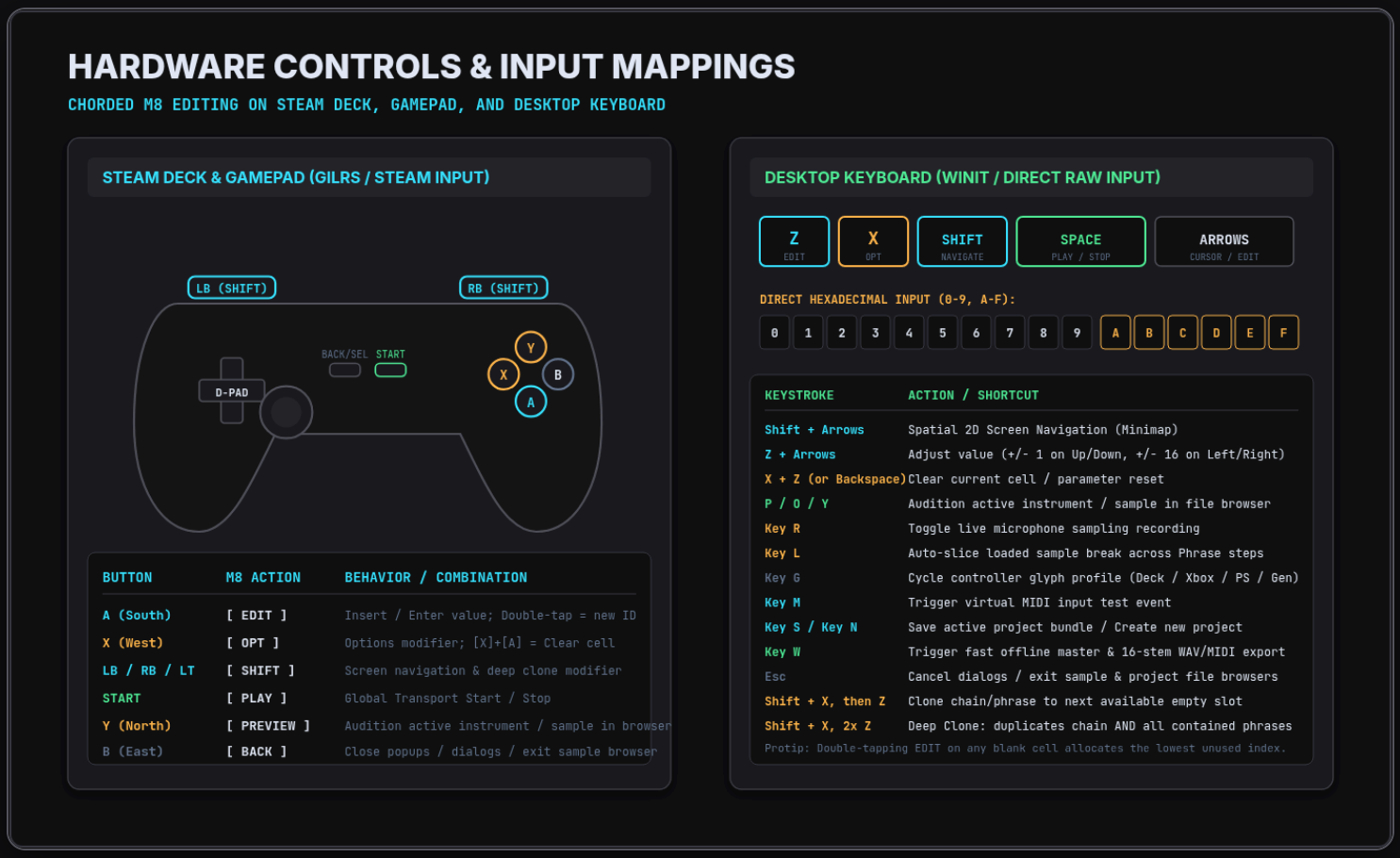
2.2 Design Philosophy & Core Differences

Feature	Classic Handheld Trackers	Polyarch
Track Count	8 concurrent tracks	**16 concurrent tracks** rendered simultaneously without horizontal scrolling
Screen Resolution	320×240 / 480×320 fixed display	**Resolution-independent vector canvas** (dynamically scales from 1280×800 up to 4K)
GPU Rendering	Raw software blitting	**Native Vulkan backend** via `wgpu`, bypassing DirectX translation layers
Audio Engine	Fixed hardware DSP chips	**48 kHz 32-bit floating-point audio engine** via algebraic `fundsp` combinator graphs
Mixer & Inserts	Basic volume, pan, and global send	**16 vertical channel strips** with dedicated 3-band EQ, drive/crush, compressor/gate, and VU meters
Drum Engine	Single sample per track	**16-pad hybrid polyphonic drum machine** (8 voices, hardware choke groups, physical synthesis + sample pool)
Sampling	File import only	**Live microphone recording** with threshold gating, auto-trim, and automatic slice mapping
Stem Rendering	Realtime audio pass-through	**Deterministic offline bounce**: stereo master `.wav`, 16 isolated stem `.wav`s, and standard `.mid` file

2.3 System Requirements & Platform Support

- **Operating Systems:** Windows 10/11 (64-bit native), SteamOS 3.0+ (via Proton), Linux (Wine/Proton/Native), ArmadaOS (ARM64 via FEX-Emu).
- **Audio Output:** WASAPI Shared Mode at 48 kHz (Windows), WinePulse / PipeWire (SteamOS/Linux). Buffer sizes: 128 to 1024 frames (default: 256 frames, 5.3 ms latency).
- **Graphics:** Vulkan 1.2+ capable GPU (Intel Iris Xe, AMD RDNA2, NVIDIA GeForce GTX 900 series or newer).
- **Storage:** Minimal, offline-first. Portable zero-install directory (`polyarch.exe` + `steam_api64.dll`).

3. Hardware Layout & Controller Mappings



3.1 The 4-Button Chorded Workflow

Polyarch is operated entirely through four primary action buttons paired with directional inputs:

1. `[EDIT]` **(Value Insertion & Step Entry):** Enters edit mode, inserts values, confirms dialogs, and advances selections. Double-tapping `[EDIT]` on an empty slot allocates the next available unused Chain, Phrase, or Instrument.

2. `[OPT]` **(Options & Modulation Modifier):** Modifies input values in increments of 16, provides contextual menus, and acts as the clear/cut button when chorded with `[EDIT]`.

3. `[SHIFT]` **(Spatial Navigation & Deep Operations):** Switches directional inputs from moving the cursor to navigating the 2D screen map. Combined with `[OPT]` and `[EDIT]`, executes deep cloning of patterns.

4. `[PLAY]` **(Transport & Audition):** Starts and stops the global tracker transport. In Live Mode, cues and isolates individual tracks.

3.2 Steam Deck & Gamepad Layout

Polyarch features first-class native gamepad support via `gilrs` and Steamworks `ISteamInput`:

Gamepad Button	M8 Tracker Equivalent	Primary Function
A (South)	`[EDIT]`	Insert value, confirm action, double-tap for new ID
X (West)	`[OPT]`	Options modifier; hold `[X]` + tap `[A]` to delete/clear cell
LB / RB / LT	`[SHIFT]`	Hold to navigate screens via D-Pad; clone modifier
START	`[PLAY]`	Toggle global playback transport start/stop
Y (North)	`[PREVIEW]`	Audition current instrument or preview sample in browser
B (East)	`[BACK]`	Cancel dialog, close sample browser, step preview shift
D-Pad / Left Stick	`[DIRECTION]`	Move cursor across rows and columns

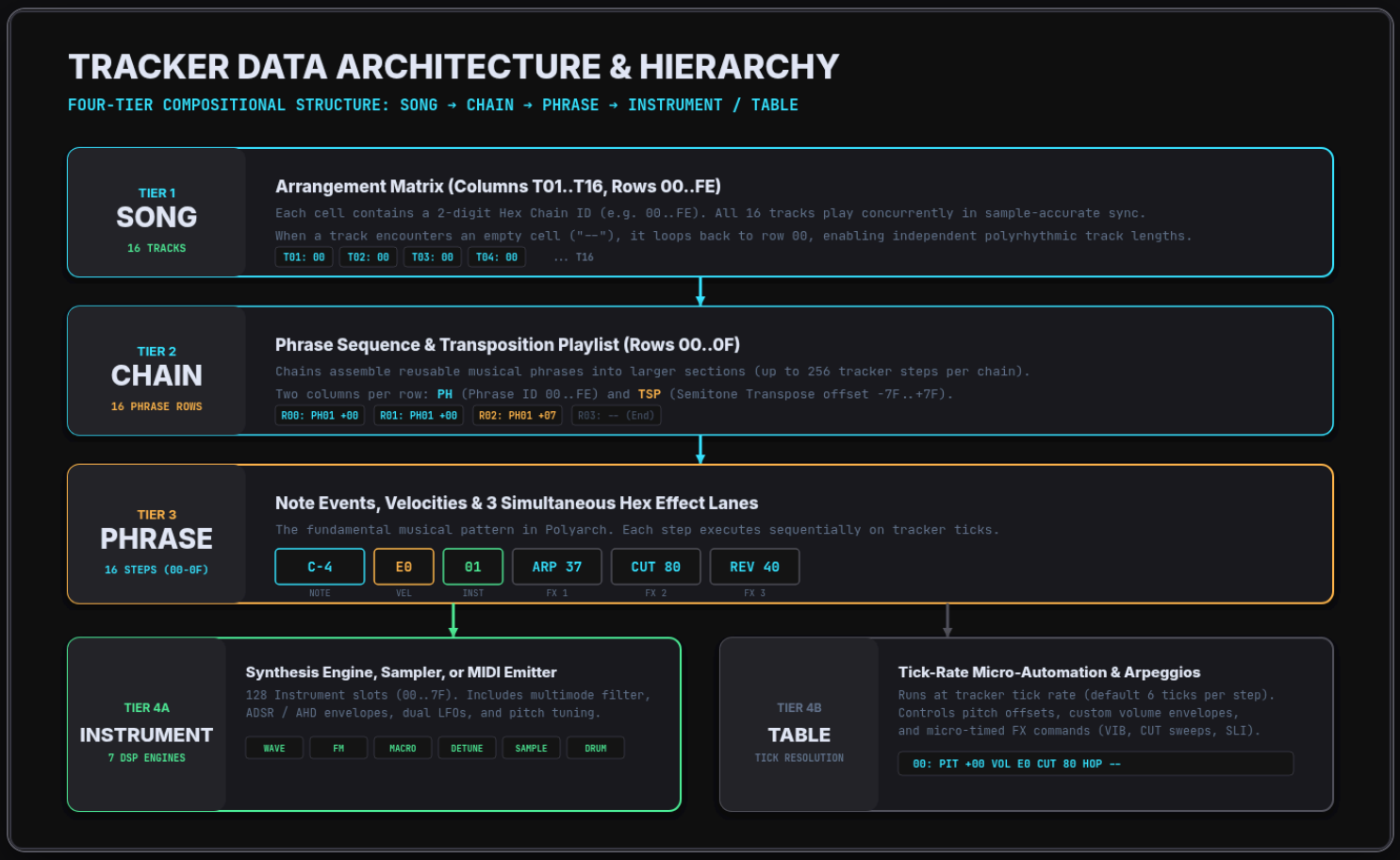
3.3 Desktop Keyboard Shortcuts & Direct Hex Entry

When operating Polyarch on a PC keyboard, inputs are handled with zero latency through raw window events:

Keystroke	Function
`Shift` + `Arrow Keys`	2D Spatial Screen Navigation (Minimap jump)
`Z` (or `Enter`)	`[EDIT]` key (tap to toggle/enter; hold `Z` + Arrows to edit values)
`X` (or `Alt`)	`[OPT]` key (hold `X` + Arrows for coarse adjustments of ±16)
`Space`	`[PLAY]` key (toggle global transport playback)
`Backspace` / `Delete`	Clear active cell (`--`)
`0` – `9`, `A` – `F`	Direct hexadecimal entry (immediately programs cell nibble)
`P` / `O` / `Y`	Audition instrument note / preview sample in browser
`Key R`	Toggle live microphone sampling recording
`Key L`	Auto-slice loaded sample break across Phrase steps
`Key S`	Instant project bundle save (`song.pgb` + `project.json`)
`Key N`	Create new blank project
`Key W`	Execute deterministic offline master, stem, and MIDI export
`Key G`	Cycle controller glyph display profile (Steam Deck / Xbox / PS / Generic)
`Key M`	Trigger virtual MIDI input test event
`Esc`	Close sample browser, cancel delete prompt, dismiss popups

4. Getting Started & Core Concepts

4.1 Tracker Song Architecture



Polyarch structures music into a strict, four-tier hierarchical data model:

```
[ SONG ] ──► 16 Tracks (T01..T16) × 256 Rows of Chains
   │
   └──► [ CHAIN ] ──► 16 Rows of Phrases + Semitone Transposition (TSP)
          │
          └──► [ PHRASE ] ──► 16 Steps: NOTE | VEL | INST | FX1 | FX2 | FX3
                 │
                 ├──► [ INSTRUMENT ] ──► 7 DSP Engines, Filter, Envelopes & LFOs
                 │
                 └──► [ TABLE ] ──────► 16 Tick-Rate Micro-Sequencing Steps
```

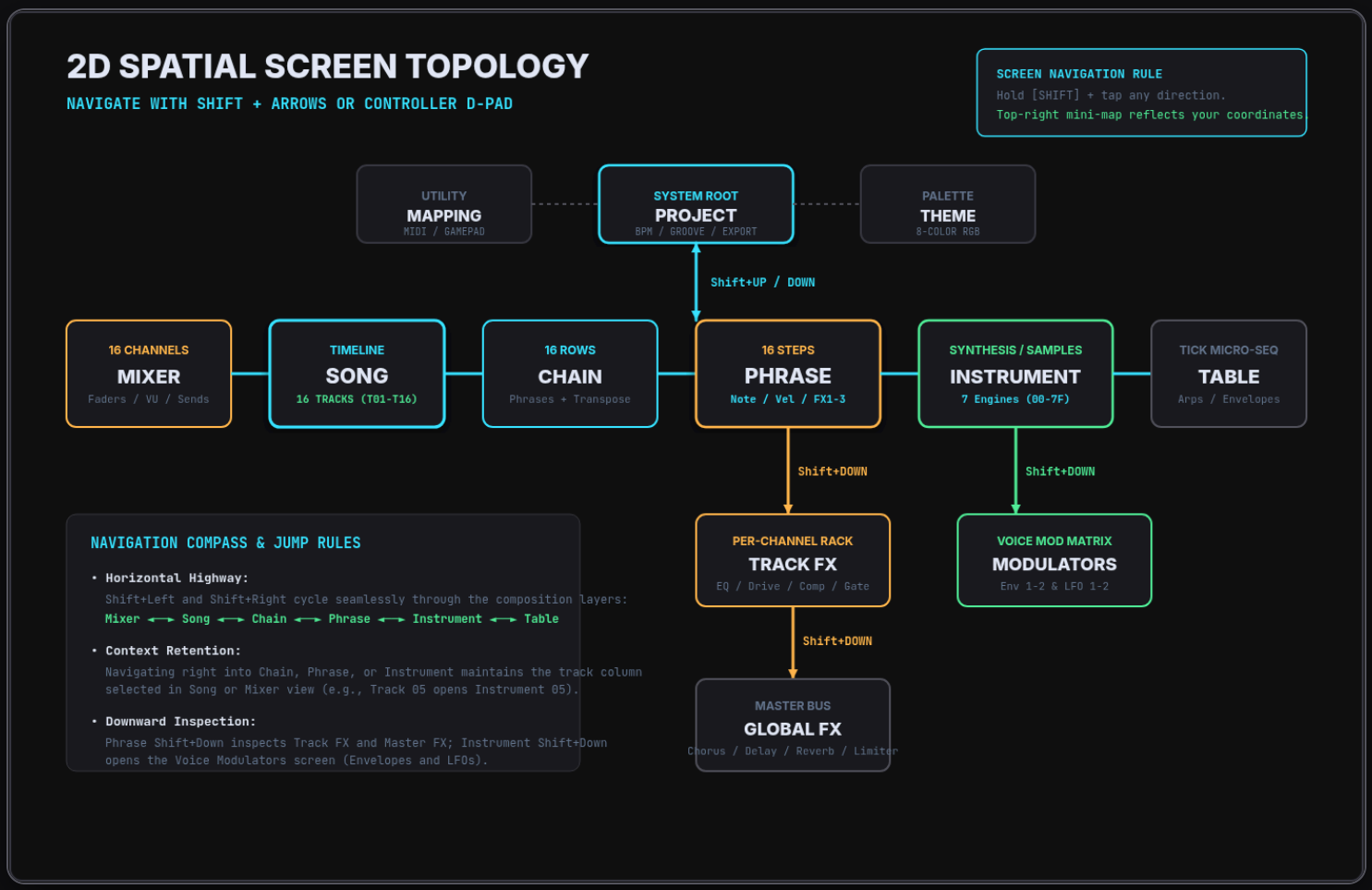
1. **Song:** The bird's-eye arrangement. Contains 16 vertical track columns (`T01` to `T16`) running up to 256 rows deep. Each cell references a **Chain ID** (`00` to `FE`).

2. **Chain:** A 16-row playlist of phrases. Each row pairs a **Phrase ID** (`00` to `FE`) with a **Transpose** offset (`TSP`), enabling complex melodic variations without copying pattern data. An empty cell (`--`) signals the end of the chain.

3. **Phrase:** A 16-step musical pattern. Each step contains six fields: `NOTE` (musical pitch or `OFF`), `VEL` (hex velocity `00` – `FF`), `INST` (instrument slot `00` – `7F`), and three independent effect command lanes (`FX1` , `FX2` , `FX3`).

4. **Instrument & Table:** The sound-generation core. Instruments define which engine (WAVE, FM, MACRO, DETUNE, SAMPLER, MIDI OUT, or DRUM) produces the voice, its filter cutoff, and envelope settings. Tables provide sub-step automation executed at tracker tick resolution.

4.2 2D Spatial Screen Topology (The Minimap)



Polyarch dispenses with tab bars and menus in favor of a 2D spatial screen map. The interface occupies one full-screen view at any moment. Holding `[SHIFT]` and pressing any direction navigates to adjacent screens:

- **Horizontal Main Highway:**

`[Mixer] ⟷ [Song] ⟷ [Chain] ⟷ [Phrase] ⟷ [Instrument] ⟷ [Table]`

- **Upward Utility Highway:**

Pressing `[SHIFT]+[UP]` from `Song` or `Chain` accesses `[Project]`, `[Theme]`, and `[Mapping]`.

- **Downward Sound Design Branches:**

- Pressing `[SHIFT]+[DOWN]` from `Phrase` opens `[Track FX]` (per-track EQ, drive, compression) and then `[Global FX]` (master chorus, delay, reverb, limiter).

- Pressing `[SHIFT]+[DOWN]` from `Instrument` opens `[Modulators]` (envelopes and LFO routing matrix).

> **TIP** The top-right corner of the interface always displays a real-time minimap indicator highlighting your current coordinates within the 2D spatial topology.

4.3 Loading the Demo Project

Polyarch includes a showcase project named `DEMO` that demonstrates all 7 synthesizer engines, drum pads, send effects, and 16-track arrangement techniques.

1. From `Song View`, hold `[SHIFT]` and press `[UP]` to enter `Project View`.

2. Navigate the cursor to highlight `LOAD` and press `[EDIT]`.

3. In the project browser, select `DEMO` and press `[EDIT]`.

4. Press `[SPACE]` (or `[START]` on gamepad) to start playback.

5. Hold `[SHIFT]` and press `[RIGHT]` to navigate through `Song`, `Chain`, `Phrase`, and `Instrument` to see the live playhead track data.

4.4 Hexadecimal Primer

Polyarch uses standard two-digit hexadecimal notation (`00` to `FF`) for values, step positions, and parameters:

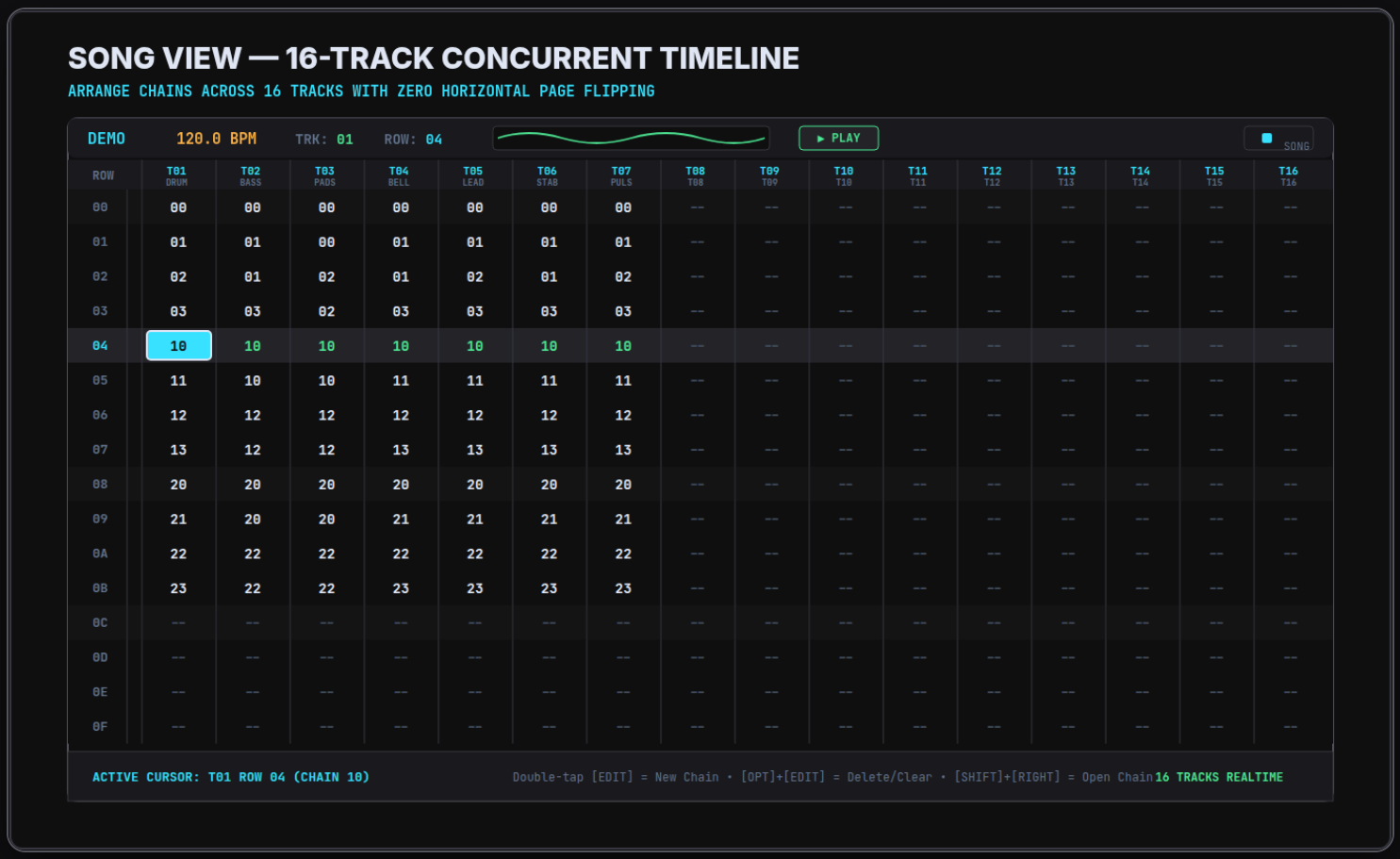
- `00` **in decimal is** `0` (minimum value / off / left).
- `80` **in decimal is** `128` (exact center value for Pan, Pitch offset `+00`, and balance).
- `FF` **in decimal is** `255` (maximum value / full volume / right).

In `Phrase` and `Chain` editing, direct digit entry (`0` – `9` , `A` – `F`) modifies the highlighted nibble immediately without opening numeric text fields.

---

5. Screen by Screen Walkthrough

5.1 Song View (16-Track Matrix)



The Song View displays all 16 tracks (`T01` to `T16`) side-by-side in high information density.

- **Columns:** 16 vertical channels labeled `T01` through `T16` with custom track name badges (e.g. `DRUM` , `BASS` , `LEAD`).
- **Rows:** Sequencer rows numbered `00` to `FE` in hex.
- **Cells:** Display the 2-digit Hex Chain ID (`00` .. `FE`) playing on that track. Empty cells display `--` .
- **Top Bar:** Shows the active project name, current BPM, active track number, row number, live audio oscilloscope tap, transport status, and navigation minimap.
- **Track Looping:** When a track encounters an empty cell (`--`), it automatically loops back to row `00` . Tracks with different chain lengths loop independently, allowing polyrhythmic arrangements.

Song View Shortcuts

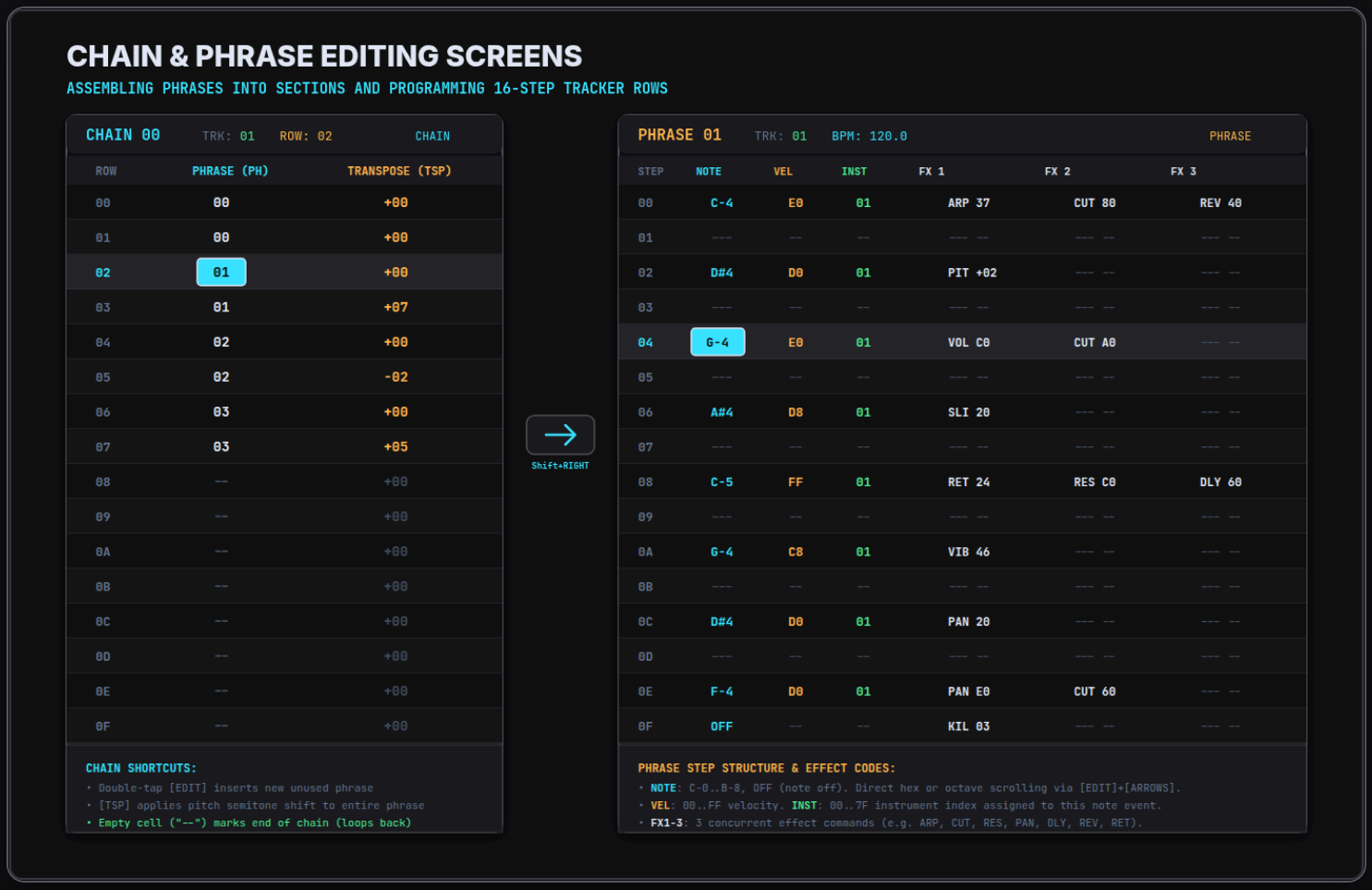
Input	Action
`[DIRECTION]`	Move cursor across tracks and rows
`[OPT]+[UP / DOWN]`	Jump cursor 16 rows up or down
`[EDIT]` on `--`	Insert chain ID (defaults to last edited chain)
`[DOUBLE TAP EDIT]`	Allocate and insert the next unused empty chain
`[OPT]+[EDIT]`	Delete chain / clear cell
`[SHIFT]+[OPT, then EDIT]`	Clone chain contents into a new chain ID
`[SHIFT]+[OPT, then 2x EDIT]`	**Deep Clone**: duplicate chain AND all referenced phrases into new IDs
`[OPT]+[SHIFT]`	Toggle mute on the current track
`[OPT]+[PLAY]`	Toggle solo on the current track

5.2 Live Playback Mode

Pressing `[SHIFT]+[LEFT]` from `Song View` enters **Live Mode**. In Live Mode:

- Tracks can be cued, isolated, or stopped on-the-fly during live performance.
- Pressing `[PLAY]` on a song cell cues that chain to start when the currently playing chain completes (quantized to chain boundary).
- Pressing `[SHIFT]+[PLAY]` stops playback on that track at the next boundary.

5.3 Chain View



The Chain View arranges phrases into musical movements. Each chain consists of 16 rows:

- `PH` **(Phrase):** Phrase index (`00` to `FE`) to trigger.
- `TSP` **(Transpose):** Pitch transposition in signed semitones (e.g. `+00` , `+07` , `-02`). Transposition shifts all note data inside the phrase without modifying the underlying pattern.
- **End of Chain:** Playback advances down rows `00` through `0F` . The first row containing `--` terminates the chain and advances the song playhead to the next song row.

5.4 Phrase View

The Phrase View is where note data, velocities, and effects are composed. Each phrase contains 16 tracker steps (`00` to `0F`) divided into six fields:

1. `NOTE` : Musical note name across 8 octaves (`C-0` to `B-8`), empty step (`---`), or note kill (`OFF`).
2. `VEL` : Note velocity in hex (`00` to `FF`). Defaults to `E0` .
3. `INST` : Instrument index (`00` to `7F`) assigned to generate the sound.

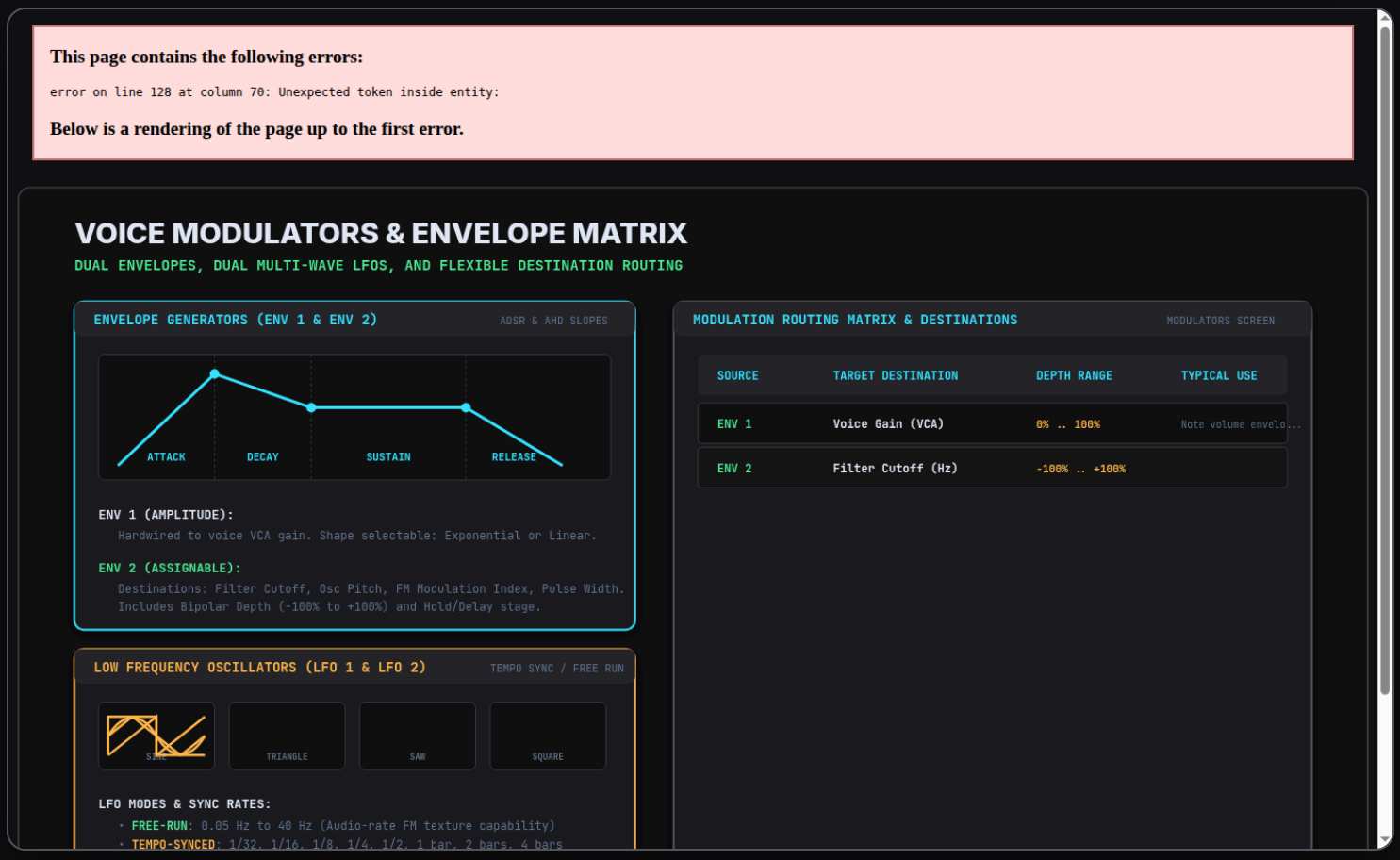
4. `FX1` / `FX2` / `FX3`: Three parallel effect lanes per step. Each lane consists of a 3-letter command mnemonic and a 2-digit hex operand (e.g., `ARP 37`, `CUT 80`, `REV 40`).

5.5 Instrument View

The Instrument View configures synthesis algorithms, filter types, tuning, and output routing for the 128 available instrument slots (`00` to `7F`).

- `ENGINE`: Selects from one of 7 internal engines: `WAVE`, `FM`, `MACRO`, `DETUNE`, `SAMPLER`, `MIDI OUT`, or `DRUM`.
- `FILTER`: State-Variable Filter (SVF) mode: Off, Low-Pass (LP), High-Pass (HP), Band-Pass (BP), or Notch.
- `CUTOFF` & `RES`: Filter cutoff frequency (`00`–`FF`) and resonance peak (`00`–`FF`).
- `AMP` & `PAN`: Output volume gain and stereo panning (`80` = center).
- `TABLE`: Assigns an automated micro-sequencing Table (`00`–`7F`) triggered on note-on.

5.6 Modulators View (Envelopes & LFOs)



Navigating `[SHIFT]+[DOWN]` from the Instrument screen reveals the Voice Modulators matrix:

- `ENV 1` **(Amplitude):** ADSR envelope controlling voice VCA volume.

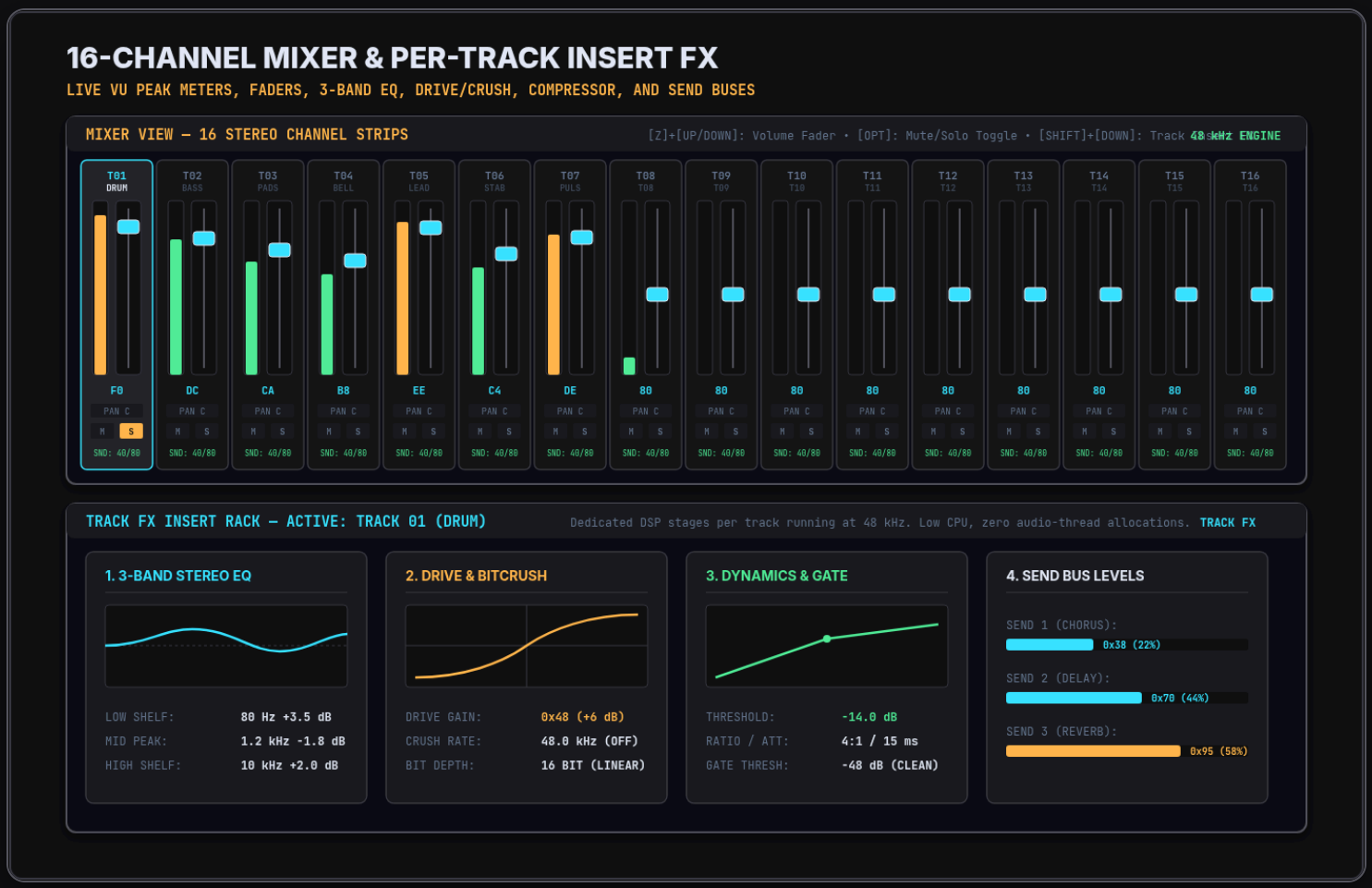
- `ENV 2` **(Assignable):** AHD/ADSR envelope routed to Filter Cutoff, Osc Pitch, FM Modulation Index, or Pulse Width with bipolar depth (`-100%` to `+100%`).
- `LFO 1` **&** `LFO 2` **:** Multi-waveform low-frequency oscillators (Sine, Triangle, Saw, Square, Sample & Hold) operating in either Free-Run (0.05 Hz–40 Hz) or Tempo-Synced (1/32 to 4 bars) modes.

5.7 Table View (Tick-Rate Micro-Sequencing)

Tables execute at tracker **tick resolution** (default 6 ticks per step, running at ~48 Hz to 96 Hz depending on tempo).

- Each table contains 16 rows with four columns: `PIT` (relative pitch offset), `VOL` (volume), and two effect lanes (`FX1` , `FX2`).
- Used to create hardware-accurate chiptune arpeggios, custom exponential envelope curves, pitch vibratos, and micro-timed filter sweeps.

5.8 Mixer View (16 Channel Strips & Peak VU Meters)



The Mixer View renders all 16 tracks simultaneously as vertical console channel strips:

- **Stereo VU Meters:** High-resolution peak meters with peak hold indicators updated at 60 Hz from real-time audio analysis.

- **Channel Faders:** Volume faders scaled from `-∞` up to `+6 dB` (hex `00` to `FF`, nominal unity at `E0`).
- **Pan Pots:** Stereo panning (`00` = Full Left, `80` = Center, `FF` = Full Right).
- **Mute / Solo Badges:** Quick toggles for stage monitoring.

5.9 Track FX Insert Rack

Pressing `[SHIFT]+[DOWN]` from the Phrase screen accesses the dedicated per-channel DSP rack for the active track:

1. **3-Band Stereo EQ:** Low Shelf (80 Hz), Parametric Mid Peak (1.2 kHz, Q=1.4), and High Shelf (10 kHz) with ±12 dB boost/cut.

2. **Drive & Bitcrush:** Analog soft-clip saturation stage paired with sample-rate decimation (crush) and bit-depth quantization.

3. **Dynamics & Noise Gate:** Downward expander gate and peak compressor with adjustable Threshold, Ratio (1:1 to 20:1), Attack, and Release.

4. **Send Levels:** Independent auxiliary send controls routing signal into Send 1 (Chorus), Send 2 (Delay), and Send 3 (Reverb).

5.10 Global FX Send Bus & Master Limiter

Pressing `[SHIFT]+[DOWN]` from Track FX opens the Master Bus screen:

- **Send 1 (Stereo Chorus):** Multi-tap modulated BBD delay line providing stereo width and lush detuning.
- **Send 2 (Ping-Pong Delay):** Tempo-synced cross-feedback delay with low-pass damping.
- **Send 3 (Algorithmic Reverb):** 8-delay feedback delay network (FDN) simulating plate, hall, and cavern acoustics.
- **Master Brickwall Limiter:** Zero-overshoot lookahead limiter placed immediately before hardware output. Ceiling fixed at `-0.1 dBFS`. Feeds the top-bar oscilloscope ring buffer.

5.11 Project View (Tempo, Groove, and File Operations)

Located above the Song View (`[SHIFT]+[UP]`):

- `BPM`: Master tempo (30.0 to 300.0 BPM in 0.1 BPM increments).
- `GROOVE`: Swing template adjusting ticks per step (e.g. alternating 6 and 7 ticks for MPC swing).
- `FILE`: New Project (`Key N`), Save (`Key S`), Load (`LOAD`), and Delete (`DELETE`).

- `EXPORT`: Triggers offline master, stem, and MIDI rendering (`Key W`).

5.12 Theme & Mapping Views

Theme View (8-Color RGB Palette)

The Theme View (`[SHIFT]+[UP]` from Project, then `[SHIFT]+[RIGHT]`) provides complete hexadecimal RGB control over Polyarch's color palette. Polyarch's default theme is **Obsidian**, designed for high legibility, low eye fatigue during long sessions, and instant recognition of cursor and beat positions:

Slot	Role Name	Default Hex	RGB Values	Role in UI & Sequencer
0	`BACKGROUND`	`#101012`	`[16, 16, 18]`	Deep canvas backdrop and master background
1	`PANEL / BAR`	`#1C1C20`	`[28, 28, 32]`	Top status header, bottom legend bar, and screen headers
2	`BEAT HIGHLIGHT`	`#26262C`	`[38, 38, 44]`	Every 4th sequencer step row background (bar/beat marker)
3	`TEXT PRIMARY`	`#E2EAF5`	`[226, 234, 245]`	Active note names, velocities, instrument numbers, and titles
4	`TEXT MUTED`	`#64768E`	`[100, 118, 142]`	Inactive labels, column headers, and secondary parameter values
5	`CURSOR / CYAN`	`#38E5FF`	`[56, 229, 255]`	Active selection cursor, highlighted cells, and primary indicators
6	`ACCENT AMBER`	`#FFB84D`	`[255, 184, 77]`	Phrase transpose, warning states, and secondary parameters
7	`SCOPE / METER`	`#50EB96`	`[80, 235, 150]`	Top-bar live oscilloscope wave tap, VU peak meters, and transport

Derived UI values automatically adapt to user palette adjustments: even row striping (`#141417`), subtle borders (`#3A3A42`), bright selection borders (`#53535C`), and empty slot indicators (`#424E5E`). Double-tapping `[EDIT]` on the Theme screen resets all channels back to the default Obsidian palette.

Mapping View

Assigns external USB MIDI Continuous Controllers (CC knobs, sliders) and gamepad triggers to tracker parameters. Highlight any parameter, move an external knob, and press `[EDIT]` to bind.

---

6. Instrument Engines In-Depth

Polyarch includes seven distinct sound engines behind its unified `InstrumentEngine` architecture:

6.1 Engine 0: WAVE (Wavsynth)

Anti-aliased analog synthesizer and wavetable oscillator engine:

- `SHAPE`: Pulse (with variable width), Sawtooth, Triangle, Noise, and Wavetable scan.
- `WARP`: Wavefolder and pulse-width modulation.
- `HARD SYNC`: Phase resets oscillator against an internal master frequency for aggressive sync sweeps.

6.2 Engine 1: FM (4-Operator FM Synth)

Classic 4-operator phase modulation synthesizer:

- `ALGORITHM`: 8 operator routing matrix topologies (series, parallel, dual carriers, branching).
- `RATIO 1..4`: Frequency multiplication ratios (0.5× to 16.0×).
- `MOD INDEX`: Modulation depth determining harmonic brightness and clangorous timbres.
- `FEEDBACK`: Self-modulation on Operator 4 for noise and saw-wave generation.

6.3 Engine 2: MACRO (Macrosynth)

Multi-algorithm digital sound generator ported to Rust:

- **Algorithms:** Resonant Formant, Modal Resonator, Chord Generator, Two-Operator FM, Harmonic Oscillator, and Digital Noise.
- `TIMBRE` & `COLOR`: Morph the primary harmonic balance and filter structure of the chosen algorithm.

6.4 Engine 3: DETUNE (Hypersynth)

High-density polyphonic super-saw and stacked unison engine:

- `VOICING`: Single unison stack, Octave fifths, Minor 7th chord, or Suspended 4th chord.
- `DETUNE`: Spreads 7 phase-independent virtual oscillators across the stereo panorama.
- `SUB OSC`: Adds a dedicated -1 octave sub-bass foundation.

6.5 Engine 4: SAMPLER (WAV Player & Live Microphone Recording)

In-memory sample playback engine with cubic interpolation:

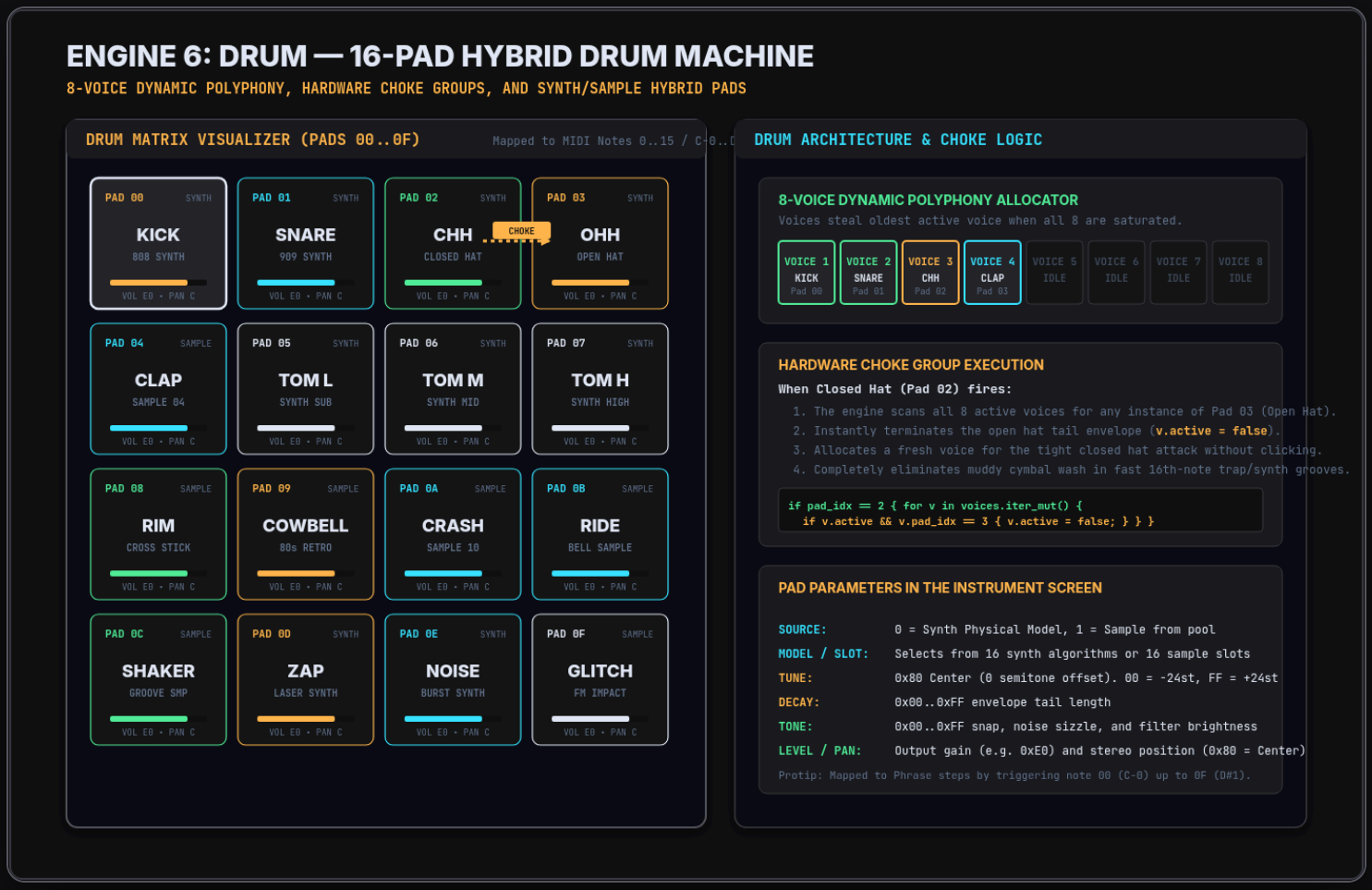
- **Supported Formats:** 16-bit, 24-bit, and 32-bit PCM `.wav` files (automatically resampled to 48 kHz).
- `PLAY MODE`: Forward, Reverse, Ping-Pong, and Sliced playback.
- **Auto-Slice (`Key L`):** Automatically divides drum breaks into 16 slices mapped across Phrase steps.
- **Live Mic Sampling (`Key R`):** Streams input from the system microphone into a lock-free buffer. Captures audio with threshold trigger, trims silence, and saves directly into the project's `samples/` directory.

6.6 Engine 5: MIDI OUT (Hardware & Virtual Routing)

Routes tracker steps out to physical USB MIDI synthesizers or software DAWs:

- `PORT`: Selects active USB MIDI hardware output port.
- `CHANNEL`: MIDI channel 1 through 16.
- `PROGRAM`: Program Change number (0–127).
- `CC A..D`: 4 assignable Continuous Controller lanes automated via FX commands.

6.7 Engine 6: DRUM (16-Pad Hybrid Machine & Choke Groups)



A self-contained 16-pad drum machine engine with an 8-voice dynamic polyphonic allocator and hardware choke groups:

- **16 Drum Pads (Pads 00..0F):** Triggered by MIDI notes `C-0` (00) through `D#1` (0F).
- **Hybrid Source per Pad:** Each pad can be independently configured as either an internal **physical modeled synthesizer** or a **sample** loaded from the 256-sample curated factory kit (`samples/drums/`).
- **Hardware Choke Group:** Pad 02 (Closed Hi-Hat) automatically chokes Pad 03 (Open Hi-Hat), instantly terminating cymbal ring-out for tight, realistic groove sequencing.
- **Per-Pad Parameters:**

- `TUNE`: Pitch offset (`80` = center unity).

- `DECAY`: Tail length (`00` to `FF`).

- `TONE`: Snap attack, noise sizzle, and filter balance.

- `LEVEL` & `PAN`: Pad output gain and stereo position.

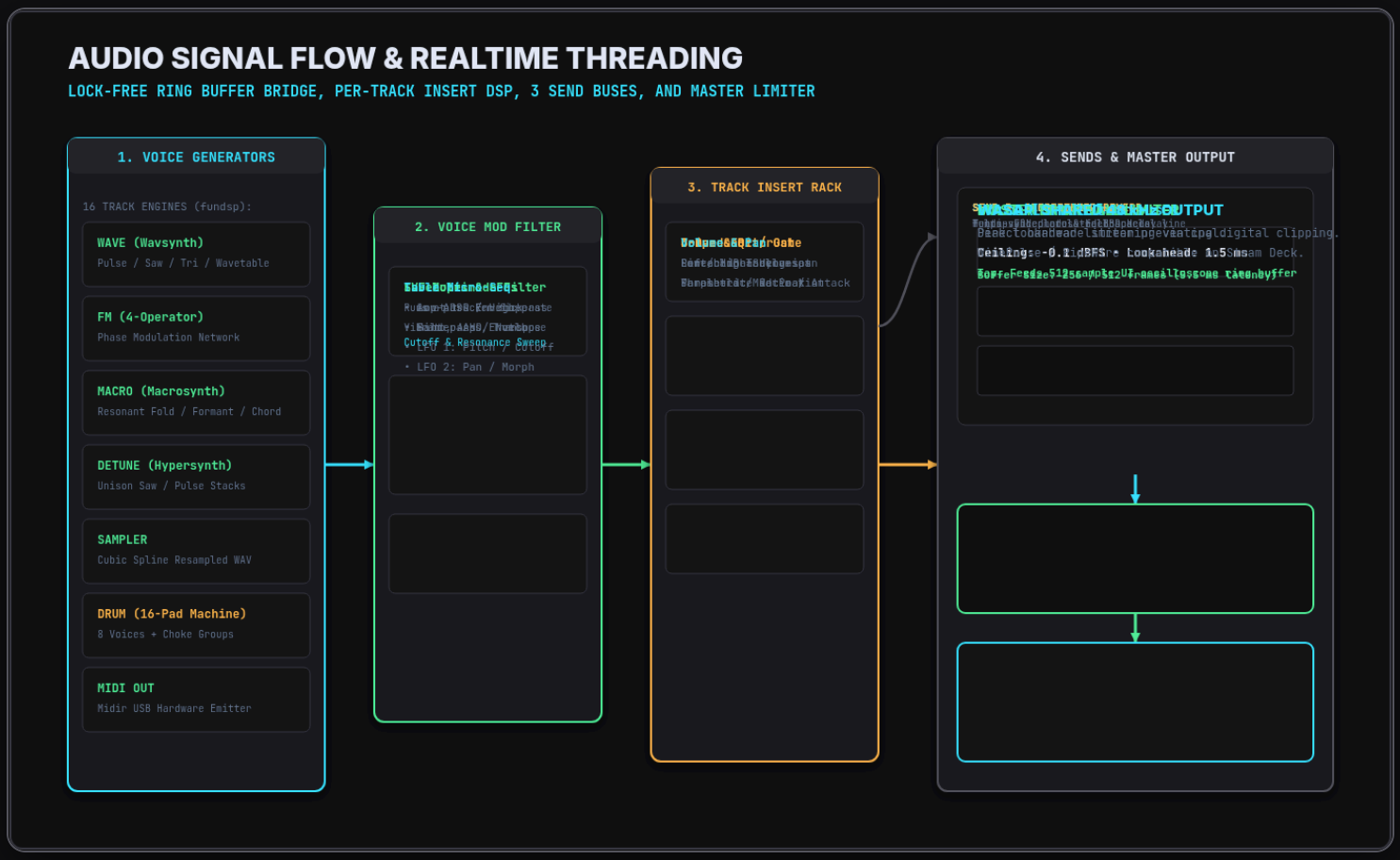
7. Hexadecimal Effect Command Reference

Polyarch supports 16 core hexadecimal effect commands across its three Phrase FX lanes and Table rows:

Code	Mnemonic	Parameter Range	Description & Usage
`00`	`ARP`	`00` – `FF`	**Arpeggiator:** High nibble = second semitone; Low nibble = third semitone. Cycles `0`, `hi`, `lo` every tick. (e.g. `ARP 37` plays minor triad, `ARP 47` plays major triad).
`01`	`PIT`	`00` – `FF`	**Pitch Offset:** Signed semitone offset (`80` = 0 offset, `<80` = pitch down, `>80` = pitch up).
`02`	`VOL`	`00` – `FF`	**Step Volume:** Overrides step velocity volume without retriggering envelopes.
`03`	`CUT`	`00` – `FF`	**Filter Cutoff:** Sweeps the SVF cutoff frequency from 20 Hz (`00`) to 20 kHz (`FF`).
`04`	`RES`	`00` – `FF`	**Filter Resonance:** Adjusts filter Q / resonance sharpness (`00` = flat, `FF` = self-oscillation).
`05`	`PAN`	`00` – `FF`	**Stereo Pan:** `00` = Hard Left, `80` = Dead Center, `FF` = Hard Right.
`06`	`RET`	`00` – `FF`	**Retrigger:** High nibble = tick spacing; Low nibble = retrigger count. Creates ratchets and stutter rolls.
`07`	`HOP`	`00` – `0F`	**Position Hop:** Jumps immediately to the specified step row, truncating phrase playback.
`08`	`SLI`	`00` – `FF`	**Portamento / Slide:** Glides pitch smoothly toward the step note at the specified rate.
`09`	`VIB`	`00` – `FF`	**Vibrato:** High nibble = vibrato speed; Low nibble = depth in cents.
`0A`	`AMP`	`00` – `FF`	**Drive / Distortion:** Adjusts the track insert soft-clip saturation gain.
`0B`	`LIM`	`00` – `FF`	**Master Limiter Ceiling:** Dynamically automates master limiter threshold.

Code	Mnemonic	Parameter Range	Description & Usage
`0C`	`CHO`	`00` – `FF`	**Chorus Send Level:** Controls auxiliary send to Send 1 (Stereo Chorus).
`0D`	`DLY`	`00` – `FF`	**Delay Send Level:** Controls auxiliary send to Send 2 (Ping-Pong Delay).
`0E`	`REV`	`00` – `FF`	**Reverb Send Level:** Controls auxiliary send to Send 3 (Algorithmic Reverb).
`0F`	`KIL`	`00` – `0F`	**Note Kill:** Abruptly cuts voice output after `N` ticks for tight rhythmic gating.

8. Audio Signal Flow & Realtime Architecture



Polyarch enforces strict real-time audio safety. The `cpal` audio callback thread **never allocates heap memory, acquires mutex locks, or performs file I/O**.

1. **Lock-Free Ring Buffers:** The UI thread and audio callback communicate exclusively through single-producer single-consumer (`rtrb`) ring buffers.

2. **DSP Voice Graphs:** Synthesizers compile down to stack-allocated `fundsp` audio combinator graphs running at 48 kHz 32-bit floating-point resolution.

3. **Insert Racks:** Each of the 16 tracks processes audio through an independent 3-band EQ, soft-clipper, and compressor before hitting the mixer fader.

4. **Master Limiter:** Audio sums into the master stereo bus, passes through a lookahead limiter at `-0.1 dBFS`, and copies a 512-sample buffer back to the UI oscilloscope visualizer.

---

9. Project Management & Offline Export

9.1 Self-Contained Project Bundles

Projects are saved in self-contained folders inside the `projects/` directory:

```
projects/MySong_120BPM/
├── song.pgb          # Bincode binary containing all 16 tracks, chains, phrases, inst
├── project.json      # Human-readable metadata (title, BPM, author, creation timestam
└── samples/          # Local copies of all WAV files and recorded mic takes used in t
```

When sharing a project, moving or zipping the single folder preserves all samples, instrument patches, and mix settings.

9.2 Offline Render Engine (`Key W`)

Polyarch features a fast, non-realtime offline export engine that bounces entire songs directly to disk in seconds:

1. **Stereo Master Mix (`exports/master_mix.wav`):** Uncompressed 48 kHz 24-bit stereo WAV with limiter processing.

2. **16-Track Isolated Stems (`exports/stems/01_DRUM.wav` .. `16_T16.wav`):** Perfect for importing into standard DAWs (Ableton, Reaper, Logic) for post-production.

3. **Standard MIDI File (`exports/master_score.mid`):** Type 1 16-channel MIDI file with notes, velocities, and CC lanes.

---

10. Appendix

10.1 Key & Controller Shortcut Cheat Sheet

Action	Desktop Keyboard	Steam Deck / Gamepad
Navigate Screens	`Shift` + `Arrow Keys`	`LB` / `RB` + `D-Pad`
Move Cursor	`Arrow Keys`	`D-Pad` or `Left Stick`
Edit / Insert Value	`Z` + `Up / Down` (±1), `Left / Right` (±16)	`A` + `D-Pad`
New Unused ID	Double-tap `Z`	Double-tap `A`
Delete / Clear Cell	`X` + `Z` (or `Backspace`)	Hold `X` + tap `A`
Clone Pattern	`Shift` + `X, then Z`	`LB` + `X, then A`
Deep Clone (Chain+Phrases)	`Shift` + `X, then 2x Z`	`LB` + `X, then 2x A`
Audition Instrument	`P` or `O`	`Y` (North)
Toggle Play / Stop	`Space`	`START`
Mute Track	`X` + `Shift`	`X` + `LB`
Solo Track	`X` + `Space`	`X` + `START`
Live Mic Record	`Key R`	Assigned via Mapping
Auto-Slice Sample	`Key L`	Assigned via Mapping
Save Project	`Key S`	Project Screen ➔ SAVE
Offline WAV Export	`Key W`	Project Screen ➔ RENDER

10.2 Hexadecimal Conversion Table

Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec
`00`	0	`40`	64	`80`	128	`C0`	192
`08`	8	`48`	72	`88`	136	`C8`	200
`10`	16	`50`	80	`90`	144	`D0`	208
`18`	24	`58`	88	`98`	152	`D8`	216
`20`	32	`60`	96	`A0`	160	`E0`	224
`28`	40	`68`	104	`A8`	168	`E8`	232
`30`	48	`70`	112	`B0`	176	`F0`	240
`38`	56	`78`	120	`B8`	184	`FF`	255

10.3 Technical Specifications

Parameter	Specification
Audio Engine	48 kHz, 32-bit floating-point stereo, real-time lock-free DSP
Polyphony	Up to 16 synth channels + 8 dynamic drum voices
Display Backend	Resolution-independent `wgpu` Vulkan backend (`Backends::VULKAN`)
Buffer Sizes	128, 256, 512, 1024 frames (WASAPI Shared / WinePulse)
Sample Playback	16/24/32-bit PCM `.wav` with cubic Hermite spline interpolation
Export Formats	48 kHz 24-bit stereo WAV, 16 individual stem WAVs, Type 1 Standard MIDI
Controller Input	`gilrs` + Steamworks `ISteamInput` with dynamic controller glyph profiles
Binary Footprint	Standalone zero-dependency executable (`polyarch.exe` + `steam_api64.dll`)

Polyarch User Guide © 2026. Engineered for producers who value speed, precision, and sonic depth.