Erae

User Manual

Embodme

2026-10-29

Contents

1 Introduction **1**

1.1 What is Erae 2? . . . 1
1.2 Key Features . . . 2
1.3 What's New in This Beta . . . 2
1.4 Who This Manual is For . . . 3
1.5 What's in the Box . . . 3
1.6 Manual Conventions . . . 4

2 Getting Started **5**

2.1 Power On and Boot Sequence . . . 5
2.1.1 Low Power Mode . . . 5
2.2 USB Connection (Class-Compliant MIDI) . . . 6
2.2.1 Using Erae with Erae Sound in a DAW . . . 7
2.3 MIDI In and Out . . . 7
2.4 USB Host Port . . . 8
2.5 Your First Layout . . . 8
2.6 Standalone Mode vs. Lab-Connected Mode . . . 9
2.6.1 Standalone Mode . . . 9
2.6.2 Lab-Connected Mode . . . 10
2.7 Next Steps . . . 10

3 Chapter 3: Surface & Touch **11**

3.1 The Pad Grid . . . 11
3.2 XYZ Sensing . . . 12
3.3 10-Finger Multitouch . . . 12
3.4 Velocity Curve Setting . . . 13
3.5 Sensitivity . . . 13
3.6 Physical Care . . . 14

4 Layouts **15**

4.1 What Is a Layout? . . . 15
4.2 Layouts per Project . . . 15
4.3 Switching Layouts on the Erae 2 . . . 16
4.4 LED Indicators . . . 16
4.5 Editing Layouts in Erae Lab . . . 17

5 Elements **18**
5.1 5.1 Iso Keyboard . . . 19
5.2 5.2 Chroma Keyboard . . . 21
5.3 5.3 Drumpad . . . 22
5.4 5.4 Fader 1D . . . 24
5.5 5.5 Fader 2D . . . 25
5.6 5.6 Button {#56-button} . . . 27
5.7 5.7 Key . . . 29
5.8 5.8 Ableton Launchpad . . . 30
5.9 5.9 API Zone {#59-api-zone} . . . 31
5.10 5.10 Pedal . . . 31
5.10.1 Switch . . . 32
5.10.2 Kick . . . 32
5.10.3 Sustain (Binary) . . . 33
5.10.4 Expressive . . . 33
5.10.5 Sustain Continuous . . . 34

6 MIDI Configuration **35**
6.1 USB MIDI Ports . . . 35
6.1.1 Remapping a Third-Port Setup After the 2.x Update . . . 36
6.2 MIDI Channel and Group . . . 36
6.3 Per-Element Output Destination . . . 37
6.4 Key Element: Expressive Parameters . . . 38
6.4.1 Pressure . . . 38
6.4.2 Vibrato (Pitch Bend) . . . 39
6.4.3 Glissando . . . 39
6.4.4 Sound Brightness (CC 74) . . . 40
6.5 CC Mapping . . . 40
6.6 High-Resolution CC and NRPN . . . 41
6.7 MIDI 1.0 vs MIDI 2.0: Which Values You Actually Read . . . 41
6.8 MPE Configuration . . . 42
6.8.1 Getting Slide and Pressure Into Your DAW . . . 43
6.9 MIDI Routing Matrix . . . 44
6.10 Related Topics . . . 45

7 Scales & Tuning **46**
7.1 Built-In Scale Library . . . 46
7.2 Root Note . . . 47
7.3 Octave . . . 47
7.4 Show Offscale . . . 48
7.5 How Scales Affect Each Keyboard Type . . . 48
7.6 Global Scale . . . 49
7.7 Custom Scales . . . 49

7.8 Per-Element Scale Settings . . . 50

8 Arpeggiator & Looper 51
8.1 Arpeggiator . . . 51
8.1.1 Enabling the Arpeggiator . . . 51
8.1.2 Rate . . . 52
8.1.3 Style . . . 52
8.1.4 Octave Range . . . 52
8.1.5 Pressure to Velocity . . . 52
8.1.6 Quantize . . . 53
8.2 Looper . . . 53
8.2.1 Home Screen Looper Display . . . 53
8.2.2 Record and Playback . . . 54
8.2.3 Looper Controls . . . 54
8.2.4 Clock Sync . . . 55
8.2.5 Pattern Length . . . 55
8.2.6 Quantize Grid . . . 55
8.2.7 Tap Tempo . . . 55

9 Chapter 9: Step Sequencer 57
9.1 Entering and Leaving the Sequencer . . . 57
9.2 What You Can Sequence . . . 57
9.3 The Surface: Three Zones . . . 58
9.3.1 The Live Zone . . . 59
9.4 What a Cell Means: the Pitch Grid . . . 59
9.4.1 Scale Relative Recording . . . 59
9.5 Editing Steps on the Grid . . . 60
9.5.1 Selecting and Moving a Group of Notes . . . 61
9.6 Moving Around a Large Pattern . . . 61
9.7 Sequencer Button Shortcuts . . . 62
9.8 The Sequencer LCD . . . 63
9.8.1 Header: the Lane Selector . . . 63
9.8.2 Grid: the Map . . . 64
9.8.3 Footer: Division and Length . . . 64
9.8.4 Encoder and Return in This View . . . 65
9.9 Switching Patterns . . . 65
9.10 Sequencer Settings . . . 65
9.11 Auto-Save . . . 67
9.12 Clock . . . 67

10 CV Outputs 68
10.1 Hardware Overview . . . 68
10.2 Output Modes . . . 69
10.3 Per-Element CV Assignment . . . 69

10.4 Additional CV Outputs per Voice . . . 70
10.5 Eurorack Quickstart . . . 71
10.5.1 First Voice Setup (Pitch + Gate) . . . 71
10.5.2 USB Host MIDI to Eurorack . . . 71
10.5.3 CV Clock at 24 ppqn . . . 71
10.5.4 Suggested Eurorack Module Configurations . . . 71
10.6 Calibration . . . 72
10.7 CV Clock Output . . . 72
10.7.1 Accessing CV Clock Settings . . . 72
10.7.2 CV Clock Parameters . . . 72

11 Chapter 11: LCD Interface **74**
11.1 11.1 Encoder Navigation Model . . . 74
11.2 11.2 Home Screen . . . 75
11.3 11.3 Settings Menu {#113-settings-menu} . . . 76
11.4 11.4 Arpeggiator Settings Screen . . . 77
11.5 11.5 Scale / Key Selector {#115-scale–key-selector} . . . 77
11.6 11.6 Info Screen (Status Overlay) . . . 78
11.7 11.7 Routing Screen . . . 79
11.8 11.9 CV Clock Screen . . . 80
11.9 11.10 Pedal Input Configuration {#1110-pedal-input-configuration} . . . 81
11.9.1 11.10.1 Format SD Card Confirmation Dialog . . . 81
11.9.2 11.10.2 Pedal Input Calibration Wizard . . . 82
11.10 11.11 Velocity Curve Editor . . . 82
11.11 11.11b Sequencer Screens . . . 83
11.12 11.12 Element Routing Screen {#1112-element-routing-screen} . . . 84
11.13 11.13 Mapping Screen . . . 84
11.14 11.15 Save / Save As / Load Screens . . . 85
11.15 11.16 Custom Scales Screens {#1116-custom-scales-screens} . . . 86

12 Chapter 12: Switches & Encoder **88**
12.1 The Panel Switches . . . 88
12.1.1 Numbered Switches (N1 to N8) . . . 88
12.1.2 Function Switches . . . 88
12.2 The Encoder . . . 89
12.2.1 Scroll . . . 89
12.2.2 Click . . . 90
12.3 Return Switch . . . 90
12.4 Shift Combinations . . . 90

13 Chapter 13: LED Feedback **91**
13.1 The RGB Color System . . . 91
13.1.1 Brightness Control . . . 91

13.2 Per-Element Visualization . . . 91
13.2.1 Keys . . . 92
13.2.2 Buttons . . . 92
13.2.3 Faders (1D and 2D) . . . 92
13.2.4 API Zone . . . 93
13.3 Animations . . . 93
13.3.1 FingerGlow . . . 94
13.3.2 Ripple . . . 94
13.3.3 Halo . . . 94
13.3.4 ColorMorph . . . 94
13.4 Press Feedback . . . 95
13.5 Status Indicators . . . 95
13.5.1 Auto-Save Indicator . . . 95
13.5.2 Project Saved Confirmation . . . 95
13.5.3 Clock Sync Pulse . . . 95
13.5.4 Switch LEDs . . . 95

14 Chapter 14: Connecting to Erae Lab 97
14.1 USB Connection . . . 97
14.1.1 Auto-Detection . . . 97
14.2 Connection Indicators . . . 97
14.3 Project Sync . . . 98
14.3.1 Conflict Resolution . . . 98
14.4 Firmware Updates via Erae Lab . . . 98
14.5 What Requires Erae Lab vs. Standalone . . . 99

15 Chapter 15: Settings 101
15.1 Global Settings . . . 101
15.1.1 Brightness . . . 101
15.1.2 Sensitivity . . . 101
15.1.3 Velocity Curve . . . 102
15.2 Project Settings . . . 103
15.2.1 Tempo . . . 103
15.2.2 Clock Source . . . 103
15.2.3 Metronome . . . 103
15.2.4 CC on Layout Change . . . 104
15.3 Custom Scales . . . 104
15.4 MIDI Routing . . . 104
15.5 CV Clock Output . . . 105
15.6 Pedal Input Configuration . . . 106
15.6.1 Type . . . 106
15.6.2 Global Enable . . . 106
15.6.3 Routing . . . 106

15.6.4 Latched . . . 107
15.6.5 Channel . . . 107
15.7 Calibration Procedures . . . 107
15.7.1 CV Calibration . . . 107
15.7.2 Encoder Calibration . . . 107
15.7.3 Pedal Input Calibration . . . 107
15.8 Project Management . . . 108
15.8.1 MIDI 2.0 . . . 108
15.9 Storage States and Recovery . . . 108

16 Chapter 16: Troubleshooting 110
16.1 Connection Issues . . . 110
16.1.1 USB Not Recognized by Erae Lab . . . 110
16.1.2 MIDI Ports Not Appearing in DAW . . . 110
16.1.3 MPE Port Not Appearing on macOS (Stale Device Entry) . . . 111
16.2 MIDI Routing Problems . . . 111
16.2.1 Notes Not Sounding on Expected Output . . . 111
16.2.2 Unexpected Notes or Double Triggers . . . 111
16.3 Touch Feels Unresponsive . . . 112
16.4 Power Issues . . . 112
16.4.1 Low Power Message . . . 112
16.5 LED Not Responding . . . 112
16.5.1 Surface LEDs Are Dark or Partially Lit . . . 112
16.6 Factory, Backup, and Reset Procedures . . . 113
16.7 Bootloader Recovery . . . 113
16.8 Orphan Bootloader Detection . . . 114
16.9 SD Card Errors . . . 114
16.10 Contacting Support . . . 115

17 Chapter 17: Erae Compatibility 116
17.1 What Is the Erae Touch? . . . 116
17.2 Projects and Layouts . . . 116
17.2.1 Main/Alt Layout Pairs . . . 116
17.3 The 5-Button Navigation System . . . 117
17.4 Step Sequencer on the Erae Touch . . . 117
17.5 Velocity Curve on the Erae Touch . . . 118
17.6 Z-Only Pressure Sensing . . . 118
17.7 No LCD . . . 118
17.8 No CV Outputs . . . 119
17.9 No Looper, No Arpeggiator Screen . . . 119
17.10 No Encoder . . . 119
17.11 2-Stage Firmware Update . . . 119
17.12 Feature Comparison: Erae Touch vs Erae 2 . . . 120

18 Appendix A: Parameter Reference 122
18.1 Common Parameters . . . 122
18.2 Controller Slots . . . 122
18.3 Key . . . 123
18.3.1 Basic Parameters . . . 123
18.3.2 Pressure Tune . . . 124
18.3.3 Vibrato Tune (Pitch Bend) . . . 124
18.3.4 CC Assignments (Optional) . . . 124
18.3.5 Control Voltage . . . 125
18.4 Button . . . 126
18.4.1 Common Button Parameters . . . 126
18.4.2 Button: Note Type . . . 126
18.4.3 Button: Control Change Type . . . 126
18.4.4 Button: Program Change Type . . . 127
18.4.5 Button: Control Voltage Type . . . 127
18.4.6 Button: Tap Tempo Type . . . 127
18.5 Fader 1D . . . 127
18.6 Fader 2D . . . 128
18.7 Keyboard (Keyboard, Keygrid, Drumpad) . . . 129
18.7.1 Keyboard Layout Parameters . . . 129
18.7.2 Glissando Tune . . . 130
18.7.3 Sound Brightness (CC 74) . . . 131
18.7.4 Arpeggiator . . . 131
18.8 Ableton Launchpad (LivePad) . . . 132
18.9 API Zone . . . 132
18.10 Non-Visible Elements . . . 133
18.10.1 Pedal . . . 133
18.11 Project Settings . . . 134
18.11.1 CV Clock Output . . . 135
18.11.2 MIDI Routing . . . 135

19 Appendix B: MIDI Implementation Chart 137
19.1 Main Implementation Table . . . 137
19.2 MPE Detail . . . 139
19.3 Universal Identity Request . . . 140
19.4 SysEx Protocol . . . 140
19.4.1 API Zone Commands . . . 140
19.5 Default CC Mappings . . . 141

20 Appendix C: DAW Setup Guides 143
20.1 General MIDI Setup . . . 143
20.1.1 USB MIDI Ports . . . 143
20.1.2 Using Erae Sound in a DAW . . . 144

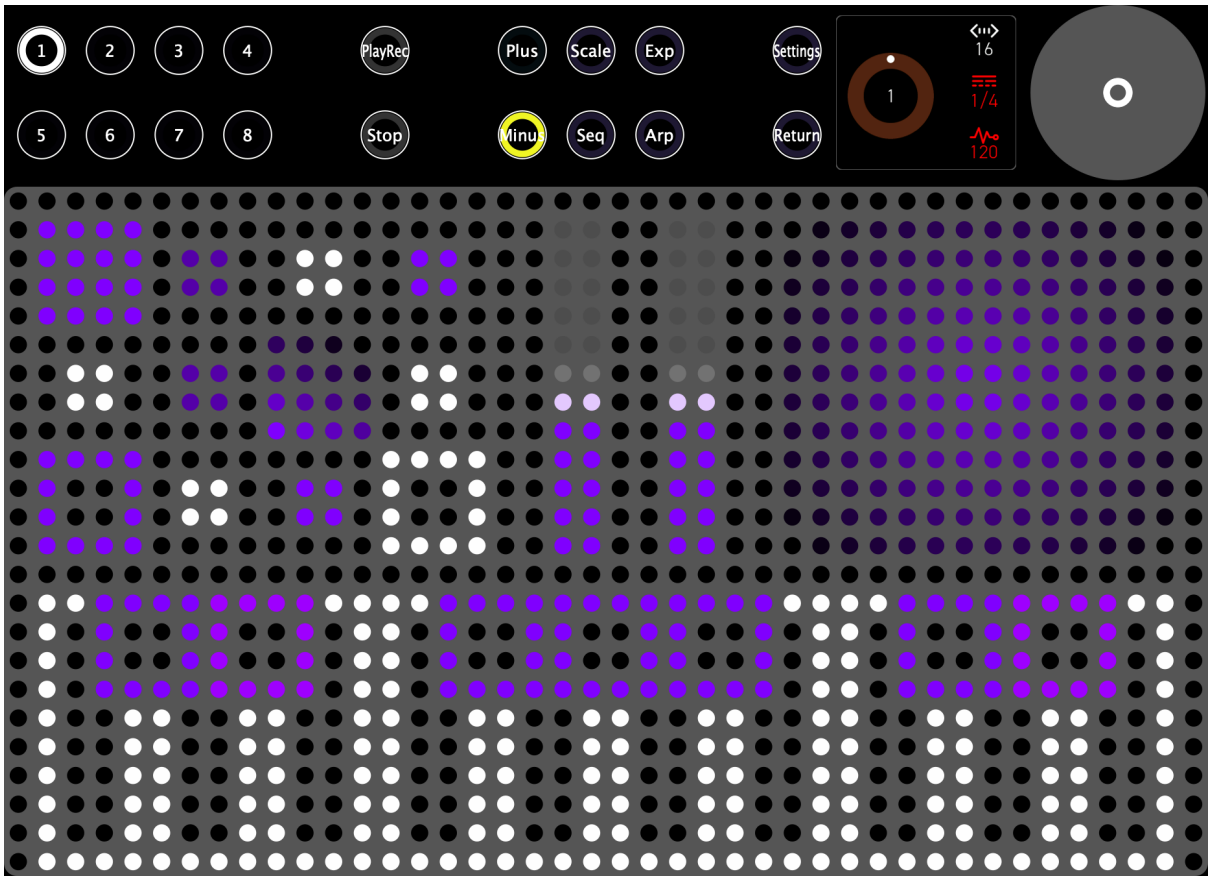
20.1.3 macOS: Removing Stale Device Entries . . . 145
20.1.4 Windows: Confirming Port Visibility . . . 145
20.2 Ableton Live . . . 145
20.2.1 Step 1: Enable the MIDI Ports . . . 145
20.2.2 Step 2: Set Up a Standard MIDI Instrument Track . . . 146
20.2.3 Step 3: Set Up an MPE Instrument Track . . . 146
20.2.4 Step 4: Install the MIDI Remote Script (LivePad Element) . . . 146
20.2.5 MIDI Clock Sync (Ableton -> Erae 2) . . . 147
20.3 Logic Pro . . . 147
20.3.1 Step 1: Enable MIDI Input in Audio MIDI Setup . . . 147
20.3.2 Step 2: Configure Logic's MIDI Preferences . . . 148
20.3.3 Step 3: Create a Standard MIDI Track . . . 148
20.3.4 Step 4: Enable MPE Per Track . . . 148
20.3.5 Step 5: Point the Track at the MPE Port . . . 148
20.3.6 Logic Pro: "Reset MIDI Drivers" Workaround . . . 148
20.3.7 MIDI Clock Sync (Logic -> Erae 2) . . . 149
20.4 FL Studio . . . 149
20.4.1 Step 1: Enable the MIDI Ports . . . 149
20.4.2 Step 2: Route MIDI to an Instrument . . . 150
20.4.3 Step 3: CC Mapping with Multi-Link . . . 150
20.4.4 Step 4: Per-Channel MPE Approximation in FL Studio . . . 150
20.4.5 MIDI Clock Sync (FL Studio -> Erae 2) . . . 151
20.5 Bitwig Studio . . . 151
20.5.1 Step 1: Configure the Controller . . . 151
20.5.2 Step 2: Enable MPE Input on an Instrument Track . . . 151
20.5.3 Step 3: Verify Note Expression . . . 151
20.5.4 Step 4: Map CC with Generic Flexi . . . 152
20.5.5 MIDI Clock Sync (Bitwig -> Erae 2) . . . 152
20.6 Cubase / Nuendo . . . 152
20.6.1 Step 1: Enable MIDI Ports . . . 152
20.6.2 Step 2: Create a Standard MIDI or Instrument Track . . . 153
20.6.3 Step 3: Declare Erae 2 as a Poly Expression Device (Cubase 12+) . . . 153
20.6.4 Step 4: Create an MPE Instrument Track . . . 153
20.6.5 Step 5: MPE Zone Configuration Per Layout . . . 153
20.6.6 MIDI Clock Sync (Cubase -> Erae 2) . . . 154
20.7 Reaper . . . 154
20.7.1 Step 1: Enable MIDI Devices . . . 154
20.7.2 Step 2: Create a MIDI Track for Standard Use . . . 154
20.7.3 Step 3: Enable MPE Input on a Track . . . 154
20.7.4 Step 4: MPE Pitch Bend Range Matching . . . 155
20.7.5 Step 5: MIDI Clock Sync (Reaper -> Erae 2) . . . 155
20.8 Connecting to Hardware Synthesizers . . . 155
20.8.1 TRS MIDI Output . . . 155

20.8.2 USB Host Port . . . 156
20.8.3 MIDI Routing Matrix for Hardware Setup . . . 156
20.9 Eurorack Integration . . . 157
20.9.1 Native CV Outputs . . . 157
20.9.2 USB Host to Eurorack MIDI-CV Modules . . . 159
20.9.3 Example Patch: Full Expressive Voice . . . 160
20.10 Related Topics . . . 161

21 Appendix D: Developer API 162
21.1 D.1 Transport . . . 162
21.2 D.2 SysEx Header . . . 163
21.3 D.3 Conventions . . . 164
21.3.1 D.3.1 Coordinate axes . . . 164
21.3.2 D.3.2 Color encoding . . . 164
21.3.3 D.3.3 Bitize-7 (image and finger data encoding) . . . 164
21.3.4 D.3.4 Streaming activation and reply framing . . . 165
21.4 D.4 Host -> Device Commands . . . 165
21.4.1 D.4.1 StartFingerDataStreaming (`0x01`) . . . 166
21.4.2 D.4.2 EndFingerDataStreaming (`0x02`) . . . 166
21.4.3 D.4.3 ZoneBoundaryRequest (`0x10`) . . . 166
21.4.4 D.4.4 Clear (`0x20`) . . . 166
21.4.5 D.4.5 SetPixel (`0x21`) . . . 166
21.4.6 D.4.6 DrawRectangle (`0x22`) . . . 167
21.4.7 D.4.7 DrawImage (`0x23`) . . . 167
21.4.8 D.4.8 VersionRequest (`0x7F`) . . . 168
21.5 D.5 Device -> Host Messages . . . 168
21.5.1 D.5.1 Zone Boundary Reply . . . 168
21.5.2 D.5.2 Version Reply . . . 169
21.5.3 D.5.3 Finger Stream . . . 169
21.5.4 D.5.4 Max Num Fingers and Finger Data Rate . . . 170
21.6 D.6 Worked Examples . . . 170
21.6.1 D.6.1 Light pixel `(3, 5)` red . . . 170
21.6.2 D.6.2 Clear zone . . . 170
21.6.3 D.6.3 Start finger streaming with prefix `7D 01` . . . 170
21.6.4 D.6.4 Drawing a 2x1 image of `[red, green]` at `(0, 0)` . . . 170
21.7 D.7 Error Handling and Limits . . . 171
21.8 D.8 Putting It Together . . . 172
21.9 D.9 Plugin Surface . . . 172
21.9.1 D.9.1 Session lifecycle . . . 173
21.9.2 D.9.2 When Show takes the screen . . . 173
21.9.3 D.9.3 Losing focus . . . 174
21.9.4 D.9.4 Host -> Device commands . . . 174
21.9.5 D.9.5 Device -> Host messages . . . 175

21.9.6 D.9.6 Worked examples 176

1 Introduction



Erae 2

1.1 What is Erae 2?

Erae 2 is a pressure-sensitive MIDI controller by Embodme. Its 42×24 LED grid is backed by a high-resolution Force-Sensitive Resistor matrix that captures X, Y, and Z (pressure) data from every finger simultaneously, turning the surface into a fully expressive instrument you design yourself. Whether you play it as a keyboard, a bank of faders, a live looper, a step sequencer, or something in between, Erae 2 adapts to your workflow.

Erae 2 is built for musicians who want more than velocity. It speaks MPE, sends CV, runs a built-in arpeggiator, looper and step sequencer, and holds eight independent layouts per project. Once configured, it plays without a computer.

1.2 Key Features

- **42×24 touch surface:** 1,008 LED cells sit above a high-resolution FSR matrix with 16 raw sensing sites per LED cell, for more than 16,000 pressure-sensing locations across the full surface.
- **XYZ expression per finger:** pressure, horizontal slide, and vertical slide are tracked independently for every simultaneous touch, up to 10 fingers.
- **10 element types:** Key, Keygrid, Keyboard, Drumpad, Button (Note, Control Change, Program Change, Control Voltage, Tap Tempo), Fader 1D, Fader 2D, Ableton Live session control, API Zone, and Pedal.
- **8 layouts per project:** store and switch between eight fully independent surface configurations. Projects are saved on the SD card.
- **MPE support:** per-note pitch bend, pressure, and slide over MIDI Polyphonic Expression for expressive polyphonic play.
- **CV outputs:** 24 control voltage outputs on 12 dual jacks for connecting to modular and analog synthesizers.
- **LCD interface:** a 280×240 colour display for adjusting scales, element mappings, the looper, the sequencer and settings without a computer.
- **Arpeggiator:** a configurable arpeggiator for any keyboard element in the active layout.
- **Looper:** a performance looper that records your touch gestures, with their pressure and slide, and replays them in sync with the project clock.
- **Step Sequencer:** draw, record and edit patterns directly on the surface, with eight patterns per layout.
- **Custom scales:** microtonal and alternative tunings for Keygrid elements, editable in Erae Lab or on the device.

> **Tip:** Erae 2 works standalone after configuration. Once your project is saved on the device, you do not need Erae Lab open during performance.

1.3 What's New in This Beta

This beta adds the following compared with the current public release.

Step Sequencer (see Step Sequencer)

- **Stay in the sequencer across layouts:** a short press on another layout button while you are in the sequencer takes you to that layout inside the sequencer, on the element you last sequenced there. Hold the button instead to preview the layout and pick a different element.
- **See what is playing:** each element with an active sequence shows a thin playhead that sweeps across it in time with the pattern. The layout buttons light up while a sequence or a loop plays on a layout you are not looking at.
- **Pitch lane:** the Pitch lane replaces the Glissando lane. A note that starts bent shows a marker at its starting pitch, linked to the note head.

- **Gesture lanes:** the X, Y and pressure gestures of a keyboard or solo key can be sequenced as their own lanes, one per element, and they are kept when you bounce a loop.
- **Drawn notes play with pressure:** a note you draw holds its velocity as a steady pressure, so it sounds on pressure sensitive synths even without a painted curve.
- **Plus and Minus on STEPS:** while the STEPS field is being edited, Plus and Minus double and halve the step count, from 1 to 256.
- **Arpeggiator on sequences:** when a keyboard's arpeggiator is on, the chords of its sequence play as arpeggios. Turn the arpeggiator off to hear the notes as drawn.
- **Bounce Loop follows Retrig:** a glissando in a bounced loop stays one note when Retrig is off on the keyboard.
- **Clear loops and sequences together:** hold one or more layout buttons, press **Return** and keep it held. After you confirm on the screen, the loops and all eight patterns of those layouts are cleared.

Scales and sounds

- **Custom scales are for Keygrid elements:** a Drumpad always keeps chromatic pads. See Scales & Tuning.
- Deleting a custom scale that a keyboard was using sets that keyboard to `Major`. Keys are coloured by the scale's own degrees when you switch to a scale with fewer or more notes.
- **Erae Sound slots:** Erae Sound presets are put into a slot from Erae Sound only. While you preview a preset, pressing a layout button takes you to that layout.
- Arpeggiator changes made on the Erae are saved with the layout and reach Erae Sound.

1.4 Who This Manual is For

This manual is written for musicians, producers, and experimenters who want to get the most out of Erae 2. It covers every aspect of the device, from placing your first element on a layout to routing CV to a modular rack. No prior experience with Erae 2 is required, but familiarity with basic MIDI concepts (channels, notes, control change) is helpful.

1.5 What's in the Box

- Erae 2 controller, with the fabric skin already mounted
- 65 W USB-C power supply
- 8 GB SD card
- Two 2 m USB-C cables
- Two MIDI TRS to DIN adapters

> **Tip:** Download Erae Lab from the Embodme website to design and manage layouts from your computer. Erae Lab is free and runs on macOS, Windows, and Linux.

1.6 Manual Conventions

Throughout this manual:

- Parameter names appear in **bold** (e.g., **Scale**, **PB Range**).
- Parameter values appear in `code formatting` (e.g., `Chromatic`, `50%`, `MPE`).
- Tips appear as blockquotes like the ones in this chapter.
- Screenshots are indicated by captions above each image.
- Cross-references link to other chapters (e.g., Layouts) or to specific sections (e.g., CV Clock Output).
- References to Erae Lab features are written as plain text: "See the Erae Lab User Manual, Chapter X."
- Note names follow the convention used on the device and in Erae Lab, where MIDI note 60 is `C3` and the lowest MIDI note (0) is `C-2`.

2 Getting Started

This chapter walks you through powering on Erae 2, making your first connections, and playing your first layout, whether you are working with a DAW over USB or performing standalone without a computer.

2.1 Power On and Boot Sequence

Erae 2 has three USB-C ports on the rear panel:

- **Power:** for the included 65 W power supply.
- **Computer:** the **USB Device** port, which connects Erae 2 to your computer as a MIDI controller.
- **Synth:** the **USB Host** port, which connects USB MIDI devices directly to Erae 2.

Insert the SD card, connect the included power supply to the **Power** port, and switch the device on. The **Computer** port can also power Erae 2 on its own, but a computer port cannot supply enough power for every feature (see Low Power Mode below).

On power-on, the following boot sequence occurs:

1. The LED surface lights up while the firmware initialises the touch hardware and loads your last-used project from the SD card. If no SD card is available, Erae 2 falls back to a backup copy stored in its internal memory.
2. The LCD display shows the boot splash, then the **Home** screen once the system is ready.
3. The layout buttons light up, indicating normal operation.

The device is ready to play as soon as the Home screen appears.

2.1.1 Low Power Mode

Erae 2 needs a power source that can deliver more than 15 W for full operation, such as the included 65 W supply. When it is powered from a source that cannot deliver that (for example a computer USB port), Erae 2 shows a **Low Power** message and runs with reduced features:

- LED brightness is reduced.
- CV outputs are disabled.

- MIDI output B is disabled.
- The metronome is disabled.
- The **Synth** (USB Host) port is not powered.

Connect the included power supply to the **Power** port to restore full operation.

> **Tip:** If the LED surface remains dark, check that the power supply is connected to the **Power** port and that the power switch is on.

2.2 USB Connection (Class-Compliant MIDI)

Erae 2 appears as a class-compliant USB MIDI device: no driver installation is required on macOS, Windows 10/11, or Linux. Connect the **Computer** (USB Device) port to your computer using a USB-C to USB-C or USB-C to USB-A cable.

In normal MIDI 1.0 mode, your DAW or MIDI software sees two user-facing USB MIDI ports:

Cable	Port name	Purpose
Main	`Erae 2 MIDI`	Standard MIDI: note output for non-MPE instruments and general DAW use
MPE	`Erae 2 MIDI (MPE)`	Per-note expression for MPE-compatible instruments (e.g. Equator2, Pigments, Omnisphere)

Select **`Erae 2 MIDI`** (Main) in your DAW's MIDI input preferences to receive note, velocity, pitch bend, and continuous controller messages from the surface. For MPE-capable instruments, point the instrument to **`Erae 2 MIDI (MPE)`** instead.

MIDI 2.0 is an alternate USB mode controlled by the **MIDI 2.0** row in **Settings**. Changing it asks for confirmation and reboots the device. Most DAW users should leave MIDI 2.0 off unless instructed by Embodme or by a specific Erae Sound or Erae Lab workflow.

> **Tip:** On macOS, open *Audio MIDI Setup -> MIDI Studio* to confirm the device is enumerated correctly. If the device appears with a numbered suffix (e.g. `Erae 2 2`) after a firmware update, open Audio MIDI Setup, select the old entry, and remove it. Reconnect USB to restore a clean enumeration.

> **Updating from firmware 1.x: port layout change.** The 2.x firmware removed the dedicated third "Lab/API" MIDI cable. Erae Lab now uses its own USB connection, which does not appear as a MIDI port in your DAW. Only two MIDI ports are present after the update:
>
> | Before 2.x (three ports) | After 2.x (two ports) |
> |---|---|
> | MIDI port 1: `Erae MIDI` (main) | `Erae 2 MIDI` (main, same role) |
> | MIDI port 2: `MIDI (Lab)` | *(removed)* |
> | MIDI port 3: `MIDI (MPE)` | `Erae 2 MIDI (MPE)` (was port 3, now port 2) |
>
> If your DAW or API setup referenced the old third port for MPE data, remap it to **`Erae 2 MIDI (MPE)`** (the second port). If you referenced the Lab/API SysEx port by number, use **`Erae 2 MIDI`** (the first port): API SysEx is now delivered on the main cable.

2.2.1 Using Erae with Erae Sound in a DAW

Erae Sound ships as a VST3/AU plugin. When you load it on a DAW instrument track:

- Fader gestures and basic keyboard note-on/off data arrive from the Erae on the **`Erae 2 MIDI`** port. Enable that port's input on the plugin track (most DAWs require you to set the track's MIDI input to this port explicitly).
- For full MPE expression (per-note pitch bend, pressure, and slide), enable **MPE mode** in the instrument track and point it at **`Erae 2 MIDI (MPE)`**. Slide (CC 74) requires the MPE port and requires the host to pass all 16 channels to the instrument.
- Do **not** run the Erae Sound engine inside Erae Lab at the same time as the plugin. Both instances listen to the same Erae MIDI input; notes and gestures reach both engines simultaneously, producing double triggering and audio bleed.

> **Silencing the built-in engine for DAW-only playback.** Erae Sound runs inside Erae Lab as a built-in audio engine. To route audio only through your DAW, click the power icon on the **Sound** tab in Erae Lab's top header bar to stop the engine. Layout editing and device sync stay active. Click the icon again to restart it. See the Erae Sound manual for details.

2.3 MIDI In and Out

Erae 2 has one MIDI input (**MIDI In**) and two MIDI outputs (**MIDI A** and **MIDI B**) on 3.5 mm TRS jacks, for connecting hardware synthesisers, drum machines, and effects units without a computer. MIDI B is disabled in Low Power Mode.

Use the included TRS to DIN adapters to connect to standard 5-pin DIN MIDI equipment: plug them into **MIDI In** and **MIDI A**, then use standard 5-pin MIDI cables to your gear.

Each element chooses which outputs carry its messages (see Per-Element Output Destination), and the routing matrix forwards incoming MIDI between ports (see MIDI Routing Matrix).

2.4 USB Host Port

The **Synth** (USB Host) port allows Erae 2 to act as a USB host, communicating with class-compliant USB MIDI devices without a computer: for example, a hardware synth with USB MIDI, a MIDI controller, or a USB-to-DIN adapter. The port supplies up to 5 W (5 V, 1 A) to the connected device, and only when the power supply is connected to the **Power** port.

Devices connected to the USB Host port appear as the **USB Host** destination in element routing and in the MIDI routing matrix. You can send touch events from the surface directly to a connected synth, or forward incoming MIDI from the synth to your computer.

> **Tip:** You can use the **Computer** and **Synth** ports at the same time.

2.5 Your First Layout

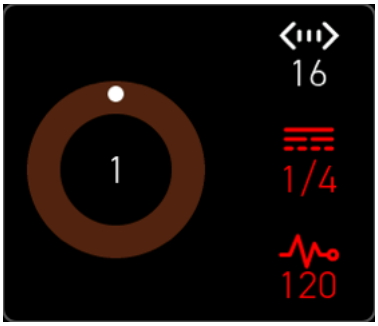
Erae 2 ships with a factory project that fills all eight layout slots. Each layout slot is selected with the numbered layout buttons **N1** to **N8**.

To select a layout, press one of the **N1** to **N8** buttons. The LED surface redraws to show the layout, and the button lights at full brightness to confirm the selection.

The factory layouts include:

- A combined layout with a drum pad, a piano-style keyboard, a large expressive key and faders
- Piano-style keyboards
- Keygrid (isomorphic) layouts, some of them set up for MPE
- A mixer-style layout with note buttons, faders and an XY pad
- A grid of individual expressive keys
- A large drum pad grid

The factory layouts use different MIDI channels, and the MPE layouts send on the `Erae 2 MIDI (MPE)` port. Each factory layout also asks Erae Sound to load a matching preset when you switch to it. You can explore, play, and perform on these layouts immediately without connecting to Erae Lab.



LCD Home Screen

The **Home** screen on the LCD shows the looper for the active layout: a circle with the layout number and the loop playback position, plus the loop **Length**, the **Quantize** grid and the **Tempo**. See Arpeggiator & Looper.

> **Tip:** Use the dedicated **N1** to **N8** layout buttons for predictable live switching.

2.6 Standalone Mode vs. Lab-Connected Mode

Erae 2 operates in two modes depending on whether Erae Lab is running and connected over USB.

2.6.1 Standalone Mode

When no computer is connected, or when Erae Lab is not open, Erae 2 runs in standalone mode. In this mode:

- The device plays the last-saved project from the SD card. Internal memory holds a backup copy and device settings, not the normal project library.
- All eight layouts are available and fully functional.
- Each element sends to the outputs selected in its routing (by default the USB Device port, `Erae 2 MIDI`).
- Edits you make on the device to a project stored on the SD card are saved automatically. The factory project and the backup project are not saved until you save them to the SD card (see Project Management).

Standalone mode is designed for live performance without a laptop. Power the device from the included power supply and Erae 2 operates entirely independently.

2.6.2 Lab-Connected Mode

When Erae Lab is open on your computer and detects Erae 2, the device enters Lab-connected mode. In this mode:

- Erae Lab can send new layouts and project configurations to the device.
- Edits made in Erae Lab are reflected on the surface immediately, without restarting.
- You can save projects from Erae Lab to the SD card for later standalone use.

> **Tip:** Lab-connected mode activates automatically when Erae Lab detects the device. Open Erae Lab and connect the USB-C cable to the **Computer** port.

See Connecting to Erae Lab for full details on the Lab workflow, project transfer, and firmware updates.

2.7 Next Steps

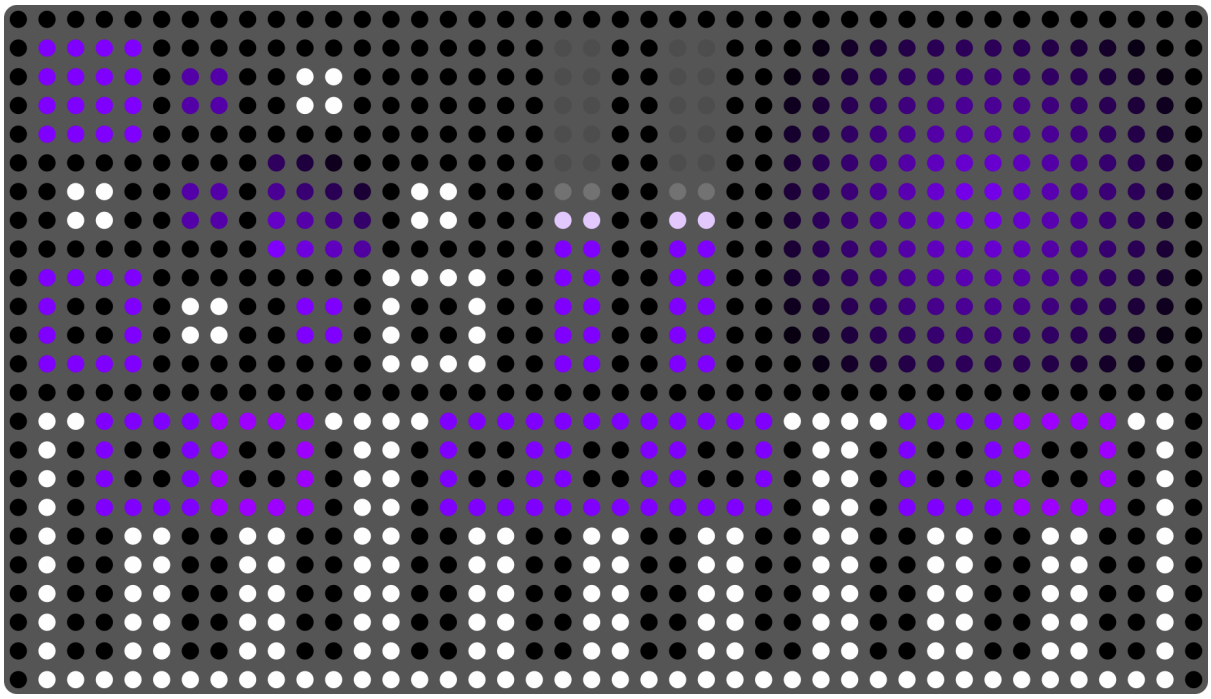
Once you have powered on the device and confirmed MIDI output from one of the factory layouts, you are ready to explore the surface in depth. Continue to Surface and Touch to learn how pressure, position, and polyphonic touch translate into expressive MIDI data.

3 Chapter 3: Surface & Touch

Erae 2 is built around a large pressure-sensitive pad that captures every nuance of your playing. This chapter explains how the surface works, what data it produces, and how to tune its behaviour for your technique.

3.1 The Pad Grid

The playing surface is organised as a **42 × 24** LED grid (42 columns across and 24 rows top to bottom), backed by a high-resolution Force-Sensitive Resistor (FSR) matrix. Each LED cell is supported by 16 raw sensing sites, giving more than 16,000 pressure-sensing locations across the playing area while keeping visual feedback aligned with the physical surface.



LED Touch Feedback

The physical grid is continuous: pad zones, faders, and keys can span any number of cells, so layouts are not constrained to a fixed button pitch. When a finger lands between cells the firmware interpolates position to sub-cell accuracy, giving smooth, high-resolution control.

> **Tip:** Wider elements (spanning more cells) give you more room to slide and produce smoother X/Y data. Use narrow elements only when you need many zones in a small space.

3.2 XYZ Sensing

Each touch reports three independent dimensions simultaneously:

- **X Position:** horizontal position of the finger within its element, normalised `0.0` to `1.0` (left to right). Sliding left or right generates X movement.
- **Y Position:** vertical position within the element, normalised `0.0` to `1.0` (bottom to top). Sliding up or down generates Y movement.
- **Z Pressure:** force applied to the surface, normalised `0.0` to `1.0`. Pressing harder increases Z.

The touch engine also derives **Motion Speed** from continuous finger movement across the surface. Motion Speed ranges from `0` to `100 cm/s` and can drive CCs.

Z is scaled so that a firm press reaches `1.0` well within the comfortable playing range.

> **Tip:** Use **X Position** and **Y Position** on Key elements to drive pitch bend and modulation simultaneously: this is the core of expressive MPE playing. See Chapter 6 for MPE configuration.

The surface is scanned **1,600 times per second** and turned into stable, expressive X/Y/Z finger data.

3.3 10-Finger Multitouch

Erae 2 tracks up to **10 simultaneous fingers** across the entire surface. Each finger is independently identified and assigned a stable tracking ID for the duration of its contact, so dense chords and two-handed gestures receive independent continuous X, Y, and Z streams.

The surface calibrates itself continuously, so there is no warm-up to wait for. Brief dips in pressure do not release a finger, which prevents accidental note-offs.

Two fingers placed very close together (closer than about one and a half cells) are treated as one touch until they move apart.

> **Tip:** Lay both hands across the surface for cluster chords: up to 10 fingers can be tracked independently as long as touches are not physically overlapping.

3.4 Velocity Curve Setting

The on-device **Velocity Curve** editor in **Settings** shapes how touch dynamics become MIDI note velocity. It is a global playing-response setting, not an element-by-element pressure curve.

- **Threshold** sets how much contact is required before velocity begins to rise.
- **Drive** changes how quickly the middle of the curve reaches higher velocities.
- **Compand** compresses or expands the response shape.
- **Range** sets the overall sensitivity: higher values need less pressure to reach full velocity. It is not a cap on the output value.

The curve preview on the LCD updates while you adjust these values. Use it to match Erae 2 to your playing style: lighter settings for subtle finger playing, firmer settings when you want more resistance before high velocities.

> **Tip:** If notes jump to maximum velocity too easily, lower **Range** or raise **Threshold**. If notes feel too soft, raise **Range** or **Drive**.

3.5 Sensitivity

The **Sensitivity** setting (found in **Settings**) controls how readily the surface registers a touch across the entire pad. Four modes are available:

Mode	Description
XSensitive	Most responsive: reacts to the lightest contact. Best for players with a very light touch. May produce false triggers if the surface is touched accidentally.
Sensitive	Responsive: lower activation threshold than the default.
Safe	Default setting: balanced threshold suitable for most playing styles and environments.
XSafe	Least responsive: requires firmer contact to register. Reduces false triggers from clothing, cables, or light accidental contact.

> **Tip:** If you experience phantom notes or false triggers during performance, switch from `Safe` to `XSafe`. If the surface feels unresponsive to light playing, try `Sensitive` or `XSensitive`. Sensitivity is the single most effective tuning adjustment for players new to FSR-based surfaces.

3.6 Physical Care

Erae 2 can be fitted with different playing skins. Treat the skin as part of the instrument, not as a replaceable drum head.

- **Fabric skin:** Designed for subtle finger performance and expressive surface control. Use clean hands where possible, avoid scraping or hard impacts, and clean stains carefully with isopropyl alcohol on a soft lint-free cloth. Do not soak the surface.
- **Drum / black skin:** Wipe with a slightly damp lint-free cloth, then dry it before storage. This skin can be played with sticks, but Erae 2 is still an electronic controller rather than an acoustic drum. Heavy drumming, sharp stick angles, or repeated high-force impacts can mark the skin and may shorten the product's life expectancy.
- **All skins:** Keep sharp objects, rings, picks, and abrasive materials away from the surface. Avoid excessive heat and prolonged direct sunlight. Store the instrument in a case when not in use.
- **Before playing:** Remove cables, tools, and other objects resting on the surface so the touch engine starts from a clean contact state.

4 Layouts

A layout is the fundamental building block of a project. It defines what the Erae 2 surface does: which elements are placed where, how they respond to touch, and what MIDI messages they send. Switching layouts lets you change the entire surface configuration in an instant, from a chromatic keyboard in one layout to a set of pressure-sensitive pads in the next.

4.1 What Is a Layout?

Each layout is an independent snapshot of the surface. It stores:

- Every **element** placed on the surface (keyboards, faders, buttons, and more)
- Each element's **position** and **size**
- All per-element **MIDI parameters** (channel, note range, CC assignments, MPE settings, etc.)
- The **scale** and base note of each keyboard element
- The **LED colors** and animations of each element
- The **pedal** configuration for the two pedal inputs
- Its own looper loop and step sequencer patterns

When you switch layouts, none of the element data from the previous layout carries over. Each layout is fully self-contained.

> **Tip:** Think of layouts as "scenes": you can dedicate one layout to melodic performance, another to rhythm pads, and a third to macro controls, then move between them without stopping your music.

4.2 Layouts per Project

Every project contains exactly **8 layouts**, numbered 1 through 8. All 8 slots are always present; any slot that has not been configured yet is empty (the surface is blank and produces no output).

You do not need to fill all 8 slots. An empty layout slot is a valid, playable state, useful as a momentary mute or a performance pause.

> **Tip:** Use an empty layout slot as a deliberate "silence" scene. Switching to it cuts all active touch events cleanly.

4.3 Switching Layouts on the Erae 2

Use the dedicated **N1** to **N8** panel buttons to switch layouts. The layout changes when you release the button. Empty slots are valid playable states and can be used as silence or mute scenes.

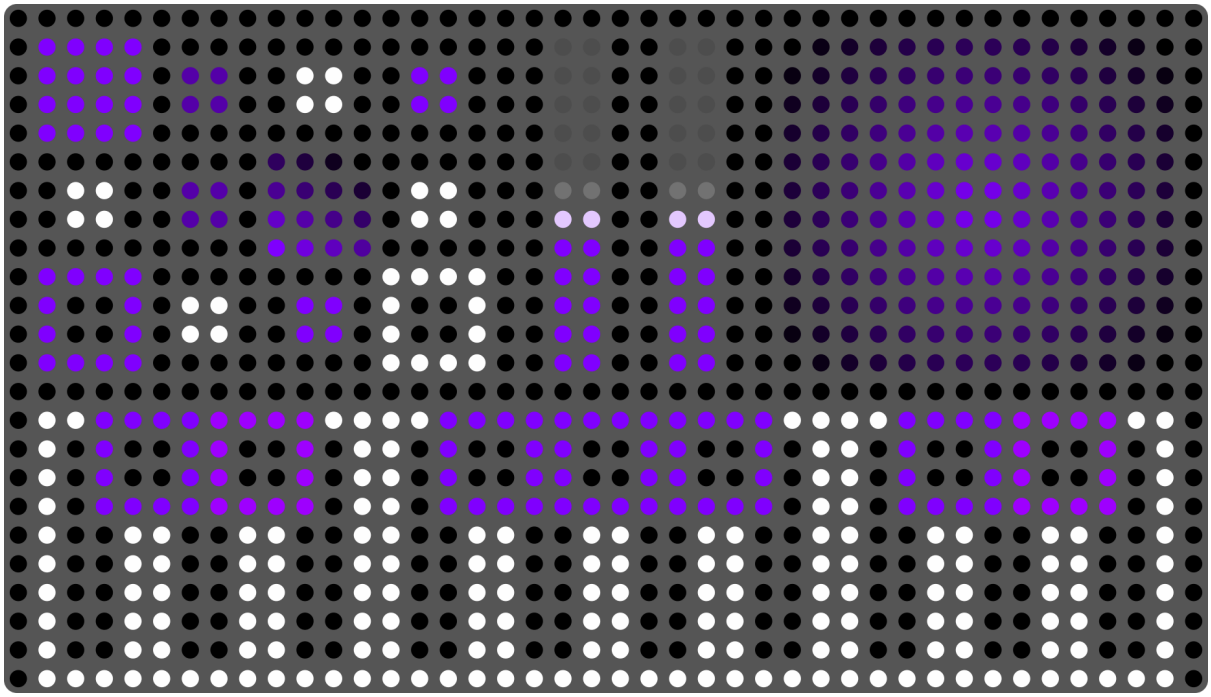
The Home screen shows the active layout number inside the looper circle. There is no separate LCD layout selector or Next/Prev layout workflow on Erae 2.

Switching layouts while fingers are held releases touches from the old layout and reintroduces held fingers to the new layout. This prevents stuck notes and phantom touches during performance layout changes.

4.4 LED Indicators

The **N1** to **N8** layout buttons show the state of each layout slot:

- The **active layout** button is lit at full brightness.
- The other layout buttons glow dimly, and briefly brighten when an element on that layout triggers a note (for example during looper or sequencer playback).
- A layout whose loop is muted is shown in red (see Arpeggiator & Looper).



Mixed Layout LEDs

This lets you see your current position and which layouts are playing at a glance, even in a dark performance environment.

4.5 Editing Layouts in Erae Lab

Layouts are created and edited in **Erae Lab** on your computer. From within Erae Lab you can:

- Drag elements onto the surface canvas and position them freely
- Configure every element's MIDI output, scale, and visual color
- Name each layout for easy identification
- Copy, paste, and reorder layouts within a project
- Preview layout LED appearance before sending to the device

When Erae Lab is connected, layout edits autosave and flow to the device through the background sync link. You normally do not need to run a manual push or pull after every edit. Wait for transfer activity to finish before disconnecting USB.

On the device itself you can adjust the scale and base note of a keyboard (see Scales & Tuning), element settings on the Mapping screen, and the arpeggiator, looper and sequencer. These edits are saved with the project.

5 Elements

Elements are the building blocks of a layout. Each element occupies a rectangular region of the touch surface, defines what MIDI (or CV) data it generates, and determines how the LEDs beneath it are rendered. A layout can contain any combination of elements placed side by side, giving you complete flexibility over how your performance surface is organized.

Every element has three shared properties:

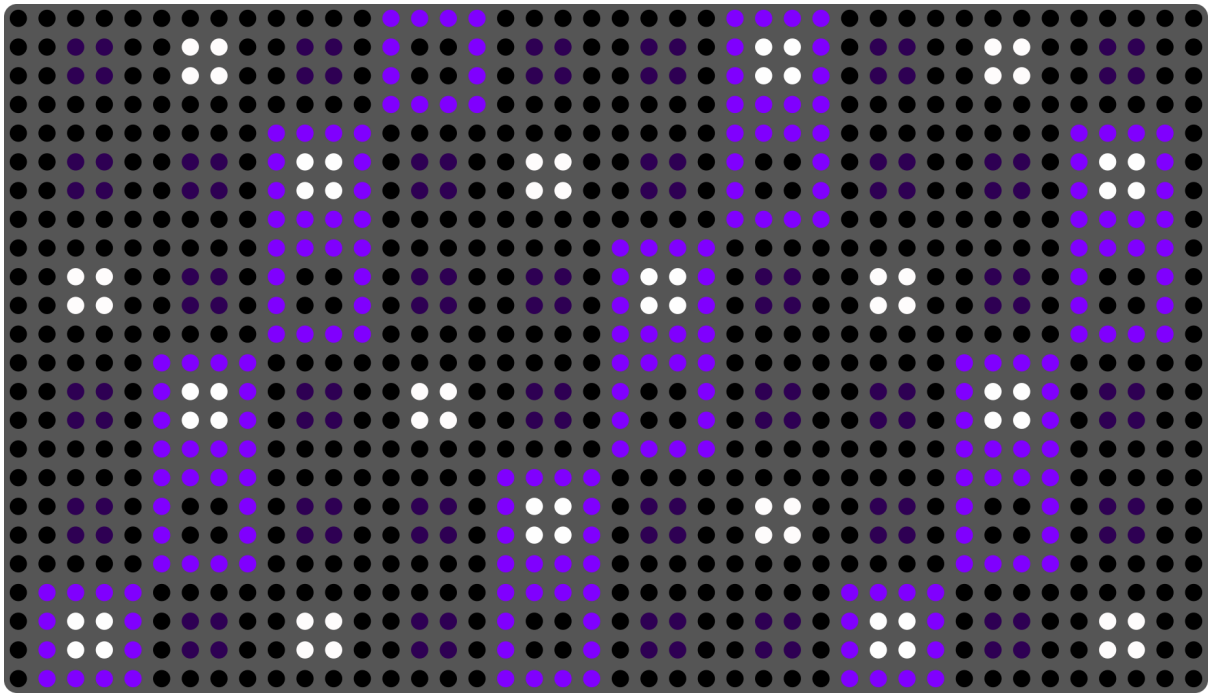
- **Geometry:** position and size on the surface grid (x, y, width, height)
- **Style:** LED color, gradient, or behavior-driven visualization
- **Animations:** optional Finger Glow, Ripple, Halo or Color Morph effects triggered when you touch, slide or release

Element-specific parameters (MIDI channel, CC numbers, note assignments, expressivity settings) are configured per element and do not affect other elements in the same layout. Every MIDI-generating element also chooses its output ports (see Per-Element Output Destination).

Elements are created and edited in Erae Lab, where they appear in the element palette as **Key**, **Keygrid**, **Keyboard**, **Drumpad**, **Button**, **Fader 1D**, **Fader 2D**, **LivePad** and **API zone**. The pedal inputs are configured separately for each layout. See Chapter 4 for how layouts work, Chapter 14 for the Lab connection, and the Erae Lab User Manual, Chapter 7 for a full guide to configuring each element type in Erae Lab.

Note names in this chapter follow the Erae convention: MIDI note 60 is C3.

5.1 5.1 Iso Keyboard



Iso Keyboard

The Iso Keyboard (called **Keygrid** in Erae Lab) is an isomorphic grid instrument: the same interval relationship exists between any two adjacent pads, regardless of where you are on the surface. Any fingering or chord shape works identically in every key, making transposition trivial and scale patterns immediately transferable.

Notes are arranged in a rectangular grid of pads, each capable of full per-note expression: pressure, horizontal slide (pitch bend and glissando), vertical slide, and CC and CV outputs.

Layout and tuning. Moving one pad to the right plays the next note. Moving up one row jumps by the line offset. With **Show Offscale** on, every chromatic semitone gets a pad and the row jump is **Semitones** (default 5, a fourth). With **Show Offscale** off, only scale notes get a pad and the row jump is **Degrees** (default 3 scale degrees). Changing one of the two offsets in Erae Lab switches **Show Offscale** to match. The base note of the bottom-left pad is set by the root note and octave (default C2).

Key Width / Key Height control how many surface cells each pad occupies. Larger pads give you more pressure-sensitive area per note.

MPE. Enable **MPE** to give each simultaneous touch its own MIDI channel, so per-note pitch bend, pressure, and slide reach MPE-capable synthesizers without crosstalk. The **MPE Master** channel can be Channel 1 or Channel 16. With MPE on, the **Channel** setting becomes **Nb channel**, the number of member channels (1 to 15).

Custom scales. A Keygrid with MPE enabled can use a custom scale. See Scales & Tuning.

Arpeggiator. The Iso Keyboard has a built-in arpeggiator. See Chapter 8 for full details.

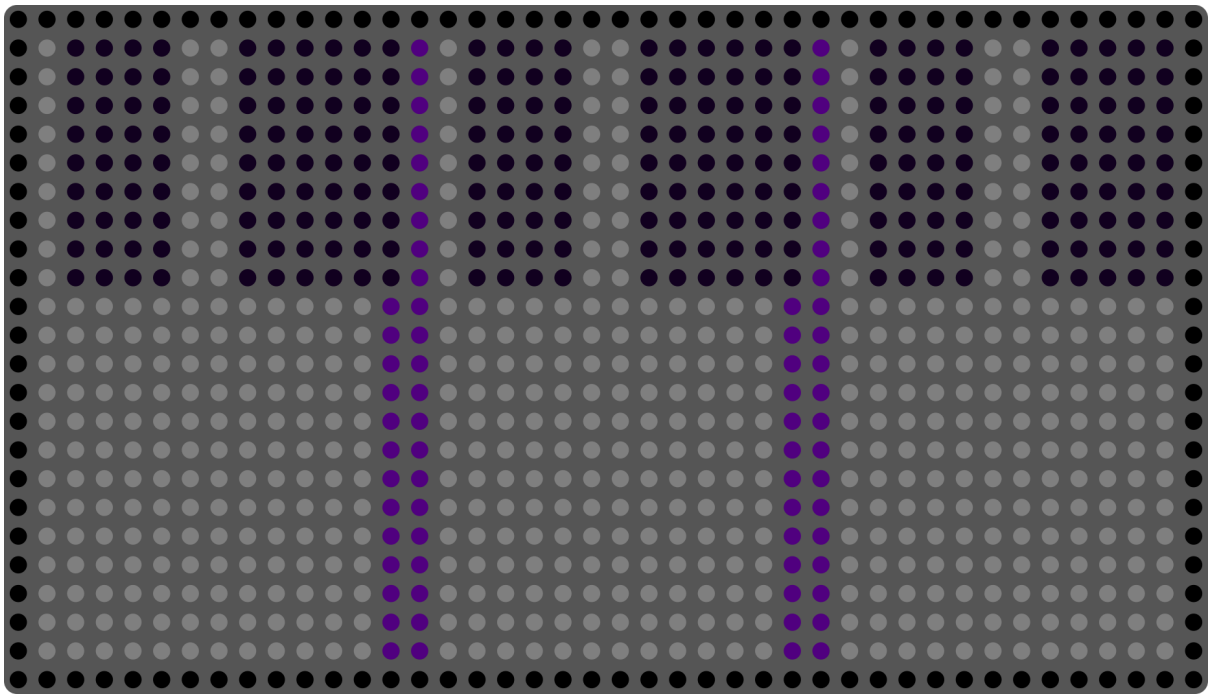
LED visualization. Scale-degree pads receive their color from the scale degree styles (one style per degree). Off-scale pads use the off-scale style. When a pad is touched, its animation plays based on the element's Animation settings.

Parameter	Description	Range	Default
Root note / Octave	Pitch of the bottom-left pad	`C-2` to `G8`	`C2`
Scale	Active musical scale	14 factory scales, or a custom scale with MPE on	`Major`
Show Offscale	Show every semitone, or only scale notes	On / Off	Off
Semitones	Row jump when Show Offscale is on	1 to 48	5
Degrees	Row jump when Show Offscale is off	1 to 48	3
Key Width	Surface cells wide per pad	cells	4
Key Height	Surface cells tall per pad	cells	4
Use global scale	Follow the project-wide scale	On / Off	Off
PB Range	Pitch bend range in semitones	1 to 96	12 (48 with MPE)
MPE	Enable MPE per-note channels	On / Off	Off
MPE Master	MPE zone master channel	`Channel 1` / `Channel 16`	`Channel 1`
Channel	MIDI channel (Nb channel with MPE on)	1 to 16	1

> **Tip:** In an isomorphic layout, the same chord shape (say, a major triad) works the same way no matter where you place your fingers. Practice chord voicings in one position and then shift them freely.

> **Tip:** Keep **Show Offscale** off and set **Scale** to `Blues` or `Japanese`. Every cell plays an in-scale note, which is ideal for improvisation without wrong notes.

5.2 5.2 Chroma Keyboard



Chroma Keyboard

The Chroma Keyboard (called **Keyboard** in Erae Lab) renders a traditional piano-style layout on the touch surface, with white and black keys. It is immediately recognizable to pianists and useful when you want to think in standard keyboard terms.

Unlike a physical piano, every key has full pressure sensitivity, and sliding along a key produces pitch bend and slide gestures on that note.

Key Width sets how many surface cells each key occupies. The number of visible keys scales with the element's width. **Line offset** sets the pitch jump between stacked rows of keys.

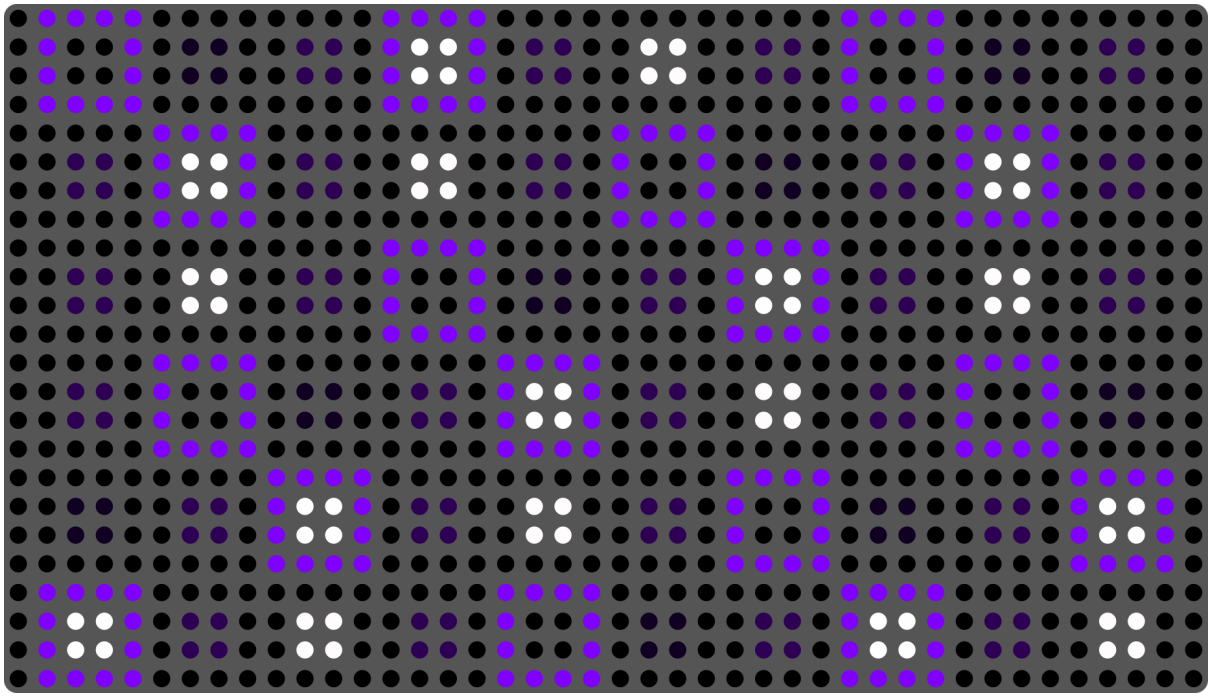
The **Scale** setting only colors the keys: root, in-scale and off-scale keys receive their own styles, but every semitone stays playable. The Chroma Keyboard always uses 12-tone equal temperament, so it cannot use custom scales or the global scale.

Glissando controls how pitch follows a finger sliding across keys. The **Sound Brightness (CC 74)** section maps a gesture (pressure, X or Y position, relative X or Y, or motion speed) to CC 74. See MIDI Configuration.

Parameter	Description	Range	Default
Octave	Octave of the lowest key	-2 to 8	2 (lowest key C2)
Scale	Scale used for coloring	15 factory scales	`Chromatic`
Key Width	Surface cells per key	cells	4
Line offset	Pitch jump between rows	1 to 48	
PB Range	Pitch bend range in semitones	1 to 96	12 (48 with MPE)
MPE	Enable MPE per-note channels	On / Off	Off
Channel	MIDI channel (Nb channel with MPE on)	1 to 16	1
Y rel CC	Relative vertical slide as a CC	CC or disabled	CC 74

> **Tip:** Place a Chroma Keyboard at the bottom of your layout spanning the full width for a performance keyboard, then add faders or buttons in the top portion for modulation controls, all within a single layout.

5.3 5.3 Drumpad



Drumpad

The Drumpad is a keyboard variant optimized for percussion. It uses the same grid structure as the Iso Keyboard but is always chromatic: each pad plays the next MIDI note, without scale filtering. This makes it straightforward to map pads to drum machine sounds, where each note corresponds to one instrument.

Arrangement. Pads are arranged left to right, bottom to top. The first pad (bottom-left) plays the base note and each pad to its right plays the next semitone. Each row up jumps by the line offset (default 4, so a 4-pad-wide grid plays consecutive notes). General MIDI drum maps start at note 36 (`C1`), which is the default base note.

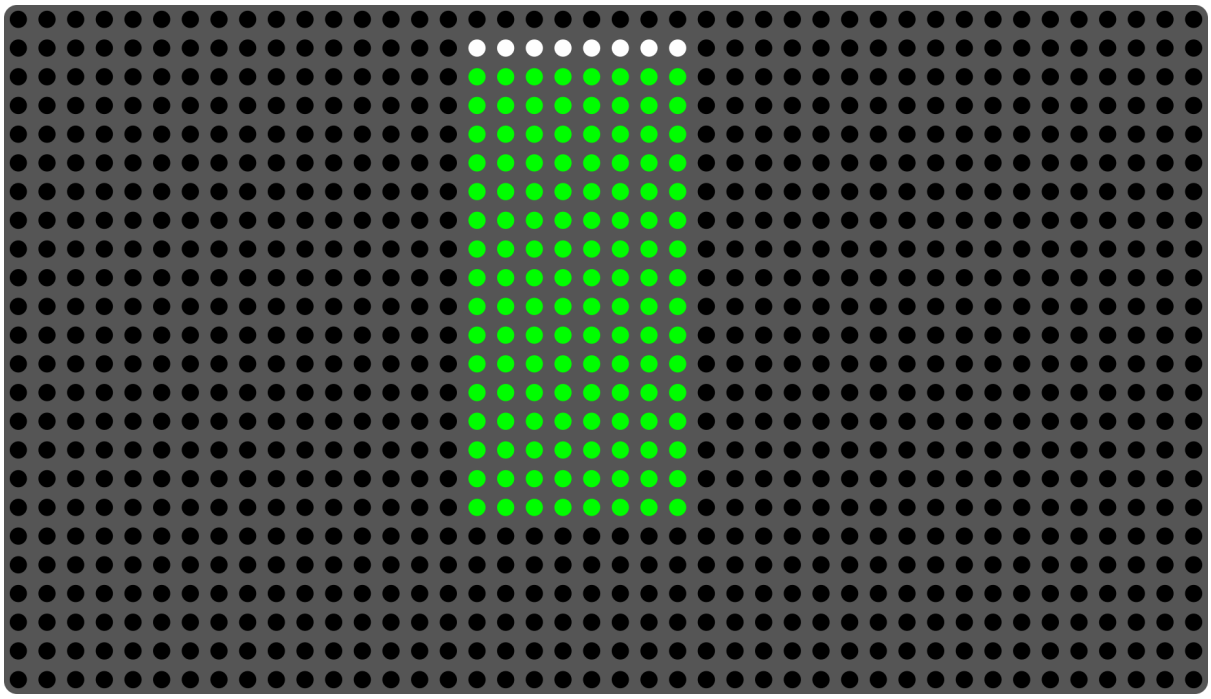
LED visualization. Each pad receives its color from a style keyed to its chromatic position (C to B), allowing you to color-code related sounds.

The Drumpad can follow the project-wide scale setting (**Use global scale**), but it always stays chromatic; custom scales do not apply.

Parameter	Description	Range	Default
Root note / Octave	Note of the first (bottom-left) pad	`C-2` to `G8`	`C1`
Key Width	Surface cells wide per pad	cells	4
Key Height	Surface cells tall per pad	cells	4
Channel	MIDI channel for all pads	1 to 16	1

> **Tip:** For a classic 4×4 drum pad grid, make the element 8×8 cells and set **Key Width** and **Key Height** to 2. You get 16 pads, each with a generous two-by-two cell footprint.

5.4 5.4 Fader 1D



Fader 1D

The Fader 1D is a single-axis continuous controller that tracks vertical finger position within the element's bounds. As you slide your finger from bottom to top, it sends a CC value from `0` to 127. The LED fill shows the current value at all times, even when not being touched.

Absolute position. The fader outputs an absolute Y position: the value corresponds directly to where your finger is on the surface, not to how far it has moved. Placing your finger at a new position jumps the value to that position.

Pressure and motion. Optional secondary CCs can follow finger pressure (**Pressure**) and finger movement speed (**Motion**), allowing simultaneous value and expression output from the same fader.

CV output. CV outputs can follow the Y axis (**Y abs**) and pressure (**Pressure**), making the Fader 1D usable in modular contexts without a MIDI-to-CV converter. See Chapter 10.

Default value. Default Y sets the fader's starting value when the layout is loaded. The default is 63 (center).

Center value. Center Y sets the visual zero point used by the LED fill. The fill is drawn between the center value and the current value, so a center of `0` looks like a traditional fader that fills from the bottom, while a center of 63 creates a bipolar fader that fills upward or downward from the middle.

When you create a fader in Erae Lab, its CC is set automatically to a controller number not yet used on its channel.

Mapping a DAW parameter. On the Erae 2 Mapping screen the `Y AXIS` section starts with a **Y Abs** switch. With your DAW in learn mode, toggle it on: the fader sends its **Y CC** at the maximum then the minimum value so the DAW learns it. Toggle it off and on again to send it once more. While **Y Abs** is off the fader sends no Y CC and its CC, Min and Max rows are hidden. Spring Fader 1D has the same switch.

Parameter	Description	Range	Default
Y Abs	Sends the Y CC; turning it on sends a test CC for DAW learn	on / off	on
Y abs CC	CC for vertical position	CC, 14-bit CC, RPN or NRPN	first unused CC
Default Y	Starting value on layout load	`0` to 127	`63`
Center Y	Visual zero point for the LED fill	`0` to 127	`0`
Pressure CC	Optional CC for finger pressure	controller or disabled	Disabled
Motion CC	Optional CC for finger movement speed	controller or disabled	Disabled
Y abs CV	CV output for vertical position	output or disabled	Disabled
Pressure CV	CV output for pressure	output or disabled	Disabled
Channel	MIDI channel	1 to 16	1

> **Tip:** Place several Fader 1D elements side by side, each on a different CC, to build a bank of mixer faders or send levels.

5.5 5.5 Fader 2D

The Fader 2D tracks both horizontal (X) and vertical (Y) finger position as independent absolute CC values, making it an XY pad. Touching anywhere within the element sets both axes to that position. The LED visualization indicates the current X/Y position and can render from a configurable center point.

Dual CC output. **X abs** and **Y abs** are assigned independently. A common assignment is filter cutoff on X and resonance on Y.

Mapping a DAW parameter. On the Erae 2 Mapping screen the `X AXIS` and `Y AXIS` sections each start with a switch, **X Abs** and **Y Abs**. With your DAW in learn mode, toggle one on: the pad sends that axis CC at the maximum then the minimum value so the DAW learns it, and only that axis, so X and Y map to two different parameters. While a switch is off its axis sends no CC and its CC, Min and Max rows are hidden. Spring Fader 2D has the same switches.

Pressure and motion. Optional **Pressure** and **Motion** CCs add further dimensions of control.

CV outputs. Both axes and pressure have CV outputs (**X abs**, **Y abs**, **Pressure**), allowing three-axis CV control from a single element.

Default position. **Default X** and **Default Y** set the starting coordinates when the layout is loaded (default 63, center).

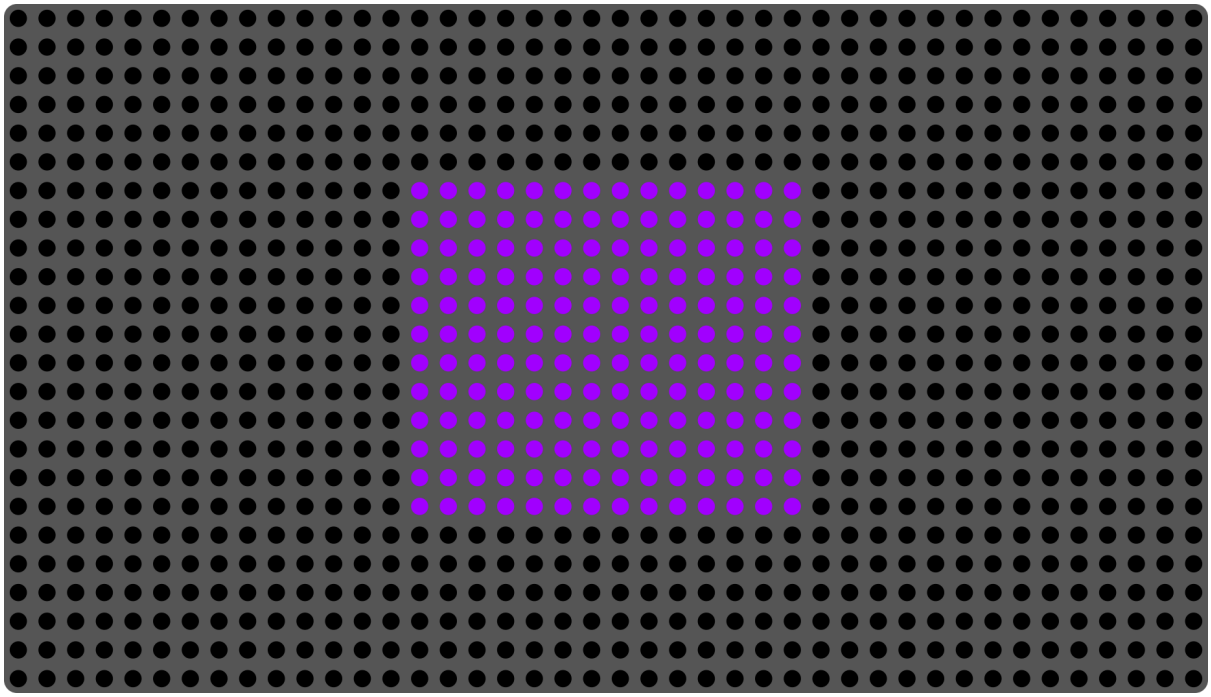
Center position. **Center X** and **Center Y** set the visual zero point used by the XY fill. The active area is drawn between the center point and the current position. With the default center of `0, 0`, the pad fills from the bottom-left corner. Setting the center to `63, 63` creates a center-detent XY pad: moving right and up fills one quadrant, moving left and down fills the opposite quadrant, and a dim center crosshair stays visible as a reference.

Parameter	Description	Range	Default
X Abs / **Y Abs**	Send the X or Y CC; turning one on sends a test CC for DAW learn	on / off	on
X abs CC	CC for horizontal position	CC, 14-bit CC, RPN or NRPN	first unused CC
Y abs CC	CC for vertical position	CC, 14-bit CC, RPN or NRPN	first unused CC
Default X	Starting X on layout load	`0` to 127	`63`
Default Y	Starting Y on layout load	`0` to 127	`63`
Center X	Visual zero point for X rendering	`0` to 127	`0`
Center Y	Visual zero point for Y rendering	`0` to 127	`0`
Pressure CC	Optional CC for finger pressure	controller or disabled	Disabled

Parameter	Description	Range	Default
Motion CC	Optional CC for finger movement speed	controller or disabled	Disabled
X abs / **Y abs** / **Pressure** CV	CV outputs	output or disabled	Disabled
Channel	MIDI channel	1 to 16	1

> **Tip:** Use the Fader 2D as a performance XY controller for a synthesizer's filter (X for cutoff, Y for resonance) while the pressure dimension modulates drive or saturation.

5.6 5.6 Button {#56-button}



Button

The Button element is a discrete trigger that sends a defined MIDI (or CV) message when touched. It has five types: **Note**, **Control Change**, **Program Change**, **Control Voltage**, and **Tap Tempo**. The **Latched** option makes the button toggle between presses instead of acting as a momentary trigger.

Momentary vs. Latched. In momentary mode (default), pressing sends the “on” message and releasing sends the “off” message. In latched mode, the first press sends the “on” message and the button holds its state; the next press sends the “off” message.

Note. Sends a Note On on press and a Note Off on release, at a fixed medium velocity. A CV output (gate and 1 V/octave pitch) can be enabled alongside the MIDI note.

Control Change. Sends the **On** controller and value on press, and the **Off** controller and value on release (or on the second press when latched). Each of the two can be enabled separately, and each can use a CC, 14-bit CC, RPN or NRPN controller.

Program Change. Sends a program change, with optional bank select (MSB and LSB), for the on state. In latched mode, a second program change with its own optional bank select is sent for the off state.

Control Voltage. Outputs a configurable on and off voltage on a CV output, with no MIDI message. Use it to trigger modular envelopes, clocks, or logic gates directly from the touch surface.

Tap Tempo. Each press taps the project tempo. Erae 2 measures the interval between successive taps and updates the tempo accordingly. No MIDI settings are needed.

LED visualization. Buttons show a rest state and an active state, with styles such as Dual Intensity, Dual Color and Center Fill.

Parameter	Description	Range	Default
Type	Button message type	Note / Control Change / Program Change / Control Voltage / Tap Tempo	Note
Latched	Toggle vs. momentary	On / Off	Off
Note (Note)	MIDI note	`C-2` to `G8`	`C2`
On controller and value (CC)	Message for the “on” state	controller or disabled, value `0` to `127`	
Off controller and value (CC)	Message for the “off” state	controller or disabled, value `0` to `127`	
Bank / **Program** (PC)	Bank select MSB/LSB and program for on and off states	`0` to `127` each, or disabled	
CV On/Off (Control Voltage)	CV output with on and off voltages	output or disabled	Disabled

Parameter	Description	Range	Default
Channel	MIDI channel	1 to 16	1

> **Tip:** Use a latched Control Change button to switch an effect on and off: press once to engage, press again to disengage, with the LED confirming the state.

> **Tip:** In Program Change mode with latching enabled, a single button can alternate between two presets: the on program on the first press, the off program on the second.

5.7 5.7 Key

The Key element is a single expressive pad: one key from a keyboard, placed as a standalone element. It sends a Note On when touched and a Note Off when released, and supports the full set of per-note expression.

Expressivity. Each dimension of touch is mapped independently:

- **Pressure:** channel pressure or polyphonic aftertouch, with its own curve
- **Vibrato:** horizontal finger movement mapped to pitch bend
- **Pressure / X abs / Y abs / Motion** CCs: pressure, absolute position and movement speed as CCs
- **X rel / Y rel** CCs: movement relative to where the finger landed, as CCs
- **CV:** gate, 1 V/octave pitch and expression outputs for modular use

The Key element is a good choice when you want a single large performance pad with full expressivity for one note: a bass drone pad, a hi-hat trigger with pressure, or a macro modulation surface.

Parameter	Description	Range	Default
Note	MIDI note to send	`C-2` to `G8`	`C3`
Pressure	Aftertouch type and curve	Channel / Poly	Channel, enabled
Vibrato	Pitch bend from horizontal movement	style, strength, return speed	enabled
PB Range	Pitch bend range in semitones	1 to 96	12

Parameter	Description	Range	Default
Pressure / X abs / Y abs / Motion CC	Optional CCs	controller or disabled	Disabled
X rel / Y rel CC	Optional relative CCs	controller or disabled	Disabled
CV	CV outputs	output or disabled	Disabled
Channel	MIDI channel	1 to 16	1

5.8 5.8 Ableton Launchpad

The Ableton Launchpad element (called **LivePad** in Erae Lab) turns Erae 2 into an Ableton Live session controller: launch clips and scenes, and control track volume, mute, solo and sends. LED colors and clip state follow Ableton Live in real time. It requires the Erae 2 MIDI Remote Script in Ableton Live; see Appendix C: DAW Setup Guides.

Position. The element occupies the **Full Width** of the surface (all 42 columns), the **Left Half**, or the **Right Half**. Splitting lets you combine a session grid on one half with other elements (faders, keyboards, buttons) on the other.

Zoom. The **Zoom** setting chooses between **Small** cells (more clips at once) and **Large** cells (easier to hit). On the device, **Plus** and **Minus** change the zoom while the element is selected.

Modes. Press **Arp** while the element is selected to cycle between the Launcher, Mixer and Split views.

Parameter	Description	Range	Default
Position	Element placement on surface	Full Width / Left Half / Right Half	set by where you place it
Zoom	Grid zoom	Small / Large	Large

> **Tip:** Combine the Ableton Launchpad element on the left half of the surface with a bank of Fader 1D elements on the right to have both session clip control and extra faders in a single layout.

5.9 5.9 API Zone {#59-api-zone}

The API Zone element reserves a region of the touch surface for direct programmatic control through the Erae 2 developer API. Instead of translating touches into MIDI, the API Zone sends finger tracking data (position and pressure) to a connected host application.

This is intended for advanced integrations: custom Max patches, TouchDesigner setups, bespoke software instruments, or any application that wants direct access to the multi-touch data.

Zone Index. Multiple API Zones can coexist in a layout, each identified by its **Zone Index**. The host application reads the zone index to know which region the data comes from.

Max Num Fingers and **Finger Data Rate** are saved with the layout but not applied yet: the zone reports every touch at the device's own scan rate. See Appendix D: Developer API.

Parameter	Description	Range	Default
Zone Index	Zone identifier for the host API	`0` to `127`	`0`
Max Num Fingers	Saved, not applied yet	`0` to `10`	`10`
Finger Data Rate	Saved, not applied yet	`0` to `255`	`0`

> **Developer API.** The full SysEx wire protocol for finger streaming, LED drawing (`SetPixel`, `DrawRectangle`, `DrawImage`), zone boundary queries, and version negotiation is documented in Appendix D: Developer API. Read the Y-axis section before correlating finger reports with drawing commands: finger Y is bottom-origin, drawing Y is top-origin.

5.10 5.10 Pedal

The Pedal element configures one of the two pedal inputs (**Pedal A** or **Pedal B**) for a layout. It has no area on the touch surface; it defines how a connected pedal is interpreted and what MIDI or CV messages it generates. You set it up in Erae Lab or on the device under **Settings > Pedal A** and **Pedal B** (see Settings).

The two pedal inputs share jacks 11 and 12 with the CV outputs. A pedal type is chosen from **Switch**, **Kick**, **Sustain (Binary)**, **Sustain (Continuous)** and **Expressive**. On the device, the two sustain types appear as **Sustain** with a **Binary** or **Continuous** mode.

Output routing. Like every MIDI-generating element, the pedal has its own output routing (USB Device, USB Host, MIDI A, MIDI B) and MIDI channel.

5.10.1 Switch

A momentary or latching footswitch. Sends a Note, Control Change, Program Change, or Tap Tempo message. In momentary mode, the message is sent on press and a complementary message on release; in latched mode, each press toggles between on and off.

Message types: - **Note:** sends Note On on press, Note Off on release - **Control Change:** sends the **On** controller and value on press, and the **Off** controller and value on release (each can be enabled separately) - **Program Change:** sends a program change (with optional bank select) for the on and off states - **Tap Tempo:** each press taps the project tempo

A **CV On/Off** output can be enabled alongside the message for simultaneous modular triggering.

Parameter	Description	Range	Default
Latched	Toggle vs. momentary	On / Off	Off
Message Type	Note / Control Change / Program Change / Tap Tempo		Note
Note (Note)	MIDI note	`C-2` to `G8`	`C4`
On / **Off** controller and value (CC)	CC messages for press and release	controller or disabled, value `0` to `127`	
CV On/Off	CV output with on and off voltages	output or disabled	Disabled

5.10.2 Kick

Designed for kick drum pedals. The Kick type detects the sudden force of a kick pedal strike, measures its strength, and sends a Note On with a matching velocity. Note Off is sent after a fixed duration or when the pedal is released.

Parameter	Description	Range	Default
Note	Drum note to trigger	`C-2` to `G8`	`C0`
Velocity Sensitivity %	Impact velocity curve strength	`0` to `100`	`100`
Impact Threshold %	Minimum strike force to trigger	`1` to `100`	`10`
Duration Mode	Note Off timing	Until Release / Fixed	Until Release

Parameter	Description	Range	Default
Fixed Duration (ms)	Note length in Fixed mode	1 to 5000	100
CV Gate	Gate CV output	output or disabled	Disabled
CV Velocity	Velocity CV output	output or disabled	Disabled

5.10.3 Sustain (Binary)

A standard on/off sustain pedal. Sends a CC (default CC 64, Sustain) with the on value when pressed and the off value when released. Use a continuous type if you need to invert the pedal direction.

Parameter	Description	Range	Default
Controller	CC number	0 to 127	64 (Sustain)
On Value	CC value when pressed	0 to 127	127
Off Value	CC value when released	0 to 127	0
Latched	Toggle mode	On / Off	Off
CV On/Off	CV output with on and off voltages	output or disabled	Disabled

---

5.10.4 Expressive

A continuous expression pedal that maps pedal position to a CC value. The full pedal travel maps to the full CC range (0 to 127). **CC 11** (Expression) is the default. **Invert** reverses the mapping for pedals with reversed polarity.

A **Pressure** CV output provides a voltage proportional to pedal position.

Parameter	Description	Range	Default
Controller	CC number	0 to 127	11 (Expression)
Invert	Reverse pedal direction	On / Off	Off
CV	Continuous CV output	output or disabled	Disabled

> **Tip:** Use an expression pedal with CC 11 routed to a synth's volume for a natural swell control, while keeping your hands free on the touch surface.

5.10.5 Sustain Continuous

Works like the Expressive type but defaults to **CC 64** (Sustain). It is intended for half-damper sustain pedals that report continuous position rather than on/off, for progressive sustain depth with a compatible piano sound.

Parameter	Description	Range	Default
Controller	CC number	`0` to 127	64 (Sustain)
Invert	Reverse pedal direction	On / Off	Off
CV	Continuous CV output	output or disabled	Disabled

> **Tip:** For electronic drums, set Pedal A to Kick and use a Drumpad on the touch surface for snare, toms, and cymbals.

> **Tip:** The pedal's routing is independent of the touch surface elements. You can send kick drum notes over MIDI A to a drum machine while everything else goes over USB to your DAW.

CV Clock Output, the project-level clock signal sent to a CV output, is covered in Chapter 10, as it is a project-wide setting rather than a per-element parameter.

6 MIDI Configuration

Erae 2 is a full-featured MIDI controller that generates standard MIDI 1.0 messages and supports MIDI 2.0 internally. Every element on the surface (keys, buttons, faders) carries its own MIDI channel and output destination, giving you complete per-element control over where messages go and how they are expressed.

This chapter covers USB MIDI port layout, channel assignment, expressive parameter mapping, CC routing, MPE, high-resolution CC and NRPN, and the MIDI routing matrix.

6.1 USB MIDI Ports

When Erae 2 is connected over USB in normal MIDI 1.0 mode, the host computer sees two user-facing MIDI cables inside a single USB MIDI device:

Port	Name	Purpose
Main	`Erae 2 MIDI`	Standard MIDI output: use this for most DAW and instrument routing
MPE	`Erae 2 MIDI (MPE)`	MPE output: select this cable in your DAW for full expressive play; MPE messages always go to this cable

> **Tip:** When setting up an MPE instrument track in your DAW, point it at the `Erae 2 MIDI (MPE)` cable. Standard non-MPE tracks should receive from `Erae 2 MIDI`.

MIDI 2.0 is an alternate USB mode controlled by the **MIDI 2.0** row in **Settings**. Changing it asks for confirmation and reboots the device. Do not route a MIDI 2.0 cable manually in your DAW unless a specific workflow tells you to do so.

> **Updating from firmware 1.x.** The 2.x firmware reduced the port count from three to two. The dedicated Lab/API SysEx cable (formerly "MIDI (Lab)", port 2) was removed: Erae Lab now uses its own USB connection, which does not appear in your DAW. MPE data, previously on port 3 ("MIDI (MPE)"), is now on port 2 ("Erae 2 MIDI (MPE)"). API SysEx host software

> that targeted the old port 2 should be remapped to `Erae 2 MIDI` (port 1). DAW setups that targeted port 3 for MPE should be remapped to `Erae 2 MIDI (MPE)` (port 2). See the Getting Started chapter for a full port-mapping table.

6.1.1 Remapping a Third-Port Setup After the 2.x Update

If you used Erae before the 2.x firmware, a DAW project, MIDI utility, or custom API host may still be pointing at a MIDI port that no longer exists. The dedicated third MIDI port (the one Erae Lab and SysEx tools used to talk to the device) was removed in 2.x. Erae Lab now reaches the device over its own USB connection, which is not a MIDI port and does not appear in any DAW or MIDI-utility port list.

What to do:

- **DAW or instrument that received notes/CC from the old main port:** switch it to **`Erae 2 MIDI`**. This carries everything the old main port did.
- **SysEx or API host that targeted the old Lab port by number or name:** switch it to **`Erae 2 MIDI`** (the main port). API SysEx now travels on the main cable: see Appendix D: Developer API.
- **Anything that targeted the third port for MPE data:** switch it to **`Erae 2 MIDI (MPE)`**.

If a tool shows an empty or greyed-out port where the third port used to be, that entry is stale. Remove or reselect it and point the tool at one of the two ports above. You do not need the old third port for anything in 2.x.

6.2 MIDI Channel and Group

Every element that generates MIDI output has a MIDI channel:

Parameter	Range	Default	Description
Channel	1 to 16	1	MIDI channel for this element's output

Channel determines which of the 16 standard MIDI channels carries the element's Note On/Off, CC, and Program Change messages. Each element on the surface can use a different channel, so a single layout can drive multiple instruments or parts in your DAW at the same time. On a keyboard with MPE enabled, this setting becomes **Nb channel** (see MPE Configuration).

Elements also carry a MIDI 2.0 group number. Erae Lab does not expose it: elements use the first group, and in MIDI 1.0 mode the group is not transmitted.

> **Tip:** Put elements on the same channel when you want them to play one instrument, or on different channels to drive several parts from one layout.

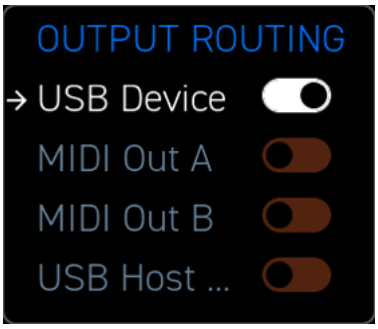
6.3 Per-Element Output Destination

Each element also selects which physical ports transmit its messages. You can enable any combination of the four ports:

Output	Port
USB Device	USB Device (the **Computer** port, to your computer)
USB Host	USB Host (the **Synth** port, to external USB gear)
MIDI A	MIDI output A
MIDI B	MIDI output B (disabled in Low Power Mode)

New elements send to **USB Device** only. Enabling additional destinations does not change the channel: the same message is sent to every selected port.

You set the outputs in Erae Lab, or on the device: open the element on the **Mapping** screen and choose **Routing** to reach the **OUTPUT ROUTING** screen, with **USB Device Out**, **MIDI Out A**, **MIDI Out B** and **USB Host Out** switches.



Element Routing

> **Tip:** Send a bass keyboard to MIDI A for a hardware synth and keep everything else on USB Device for your DAW.

6.4 Key Element: Expressive Parameters

Keys and keyboards are the primary expressive touch zones. A key sends Note On at contact, tracks pressure and position while held, and sends Note Off on release. Note velocity comes from how you strike the surface, shaped by the device-wide **Velocity Curve** in Settings (see Surface & Touch). In Erae Lab, the expressive parameters are grouped under **Pressure**, **Glissando** (keyboards only) and **Vibrato**.

6.4.1 Pressure

Pressure is the continuous force applied after note-on. Erae 2 can send pressure as polyphonic aftertouch (per note) or channel pressure (per channel):

Parameter	Range	Default	Description
Enable	on / off	on	Sends pressure messages
Type	`Channel` / `Poly`	`Channel`	Message type used for pressure output
Tracking	`None` / `LastPlayed` / `Highest` / `Lowest`	`None`	For channel pressure: which finger drives the value when several fingers are held
Filter	`Exponential` / `Linear`	`Exponential`	Response curve
Min/Max	`0` to `127`	`0` / `127`	Output range
Intensity	`0` to `255`	`127`	Sensitivity
Smoothing	`0` to `255`	`0`	Low-pass smoothing applied to pressure readings

> **Tip:** For MIDI 1.0 MPE instruments, channel pressure is usually the safest choice because each finger already has its own member channel. For traditional non-MPE synthesizers with a single aftertouch input, use `Channel` with **Tracking** set to `Highest`.

> **Warning:** In non-MPE mode, keys held at the same time share one MIDI channel. With `Poly`, each note's pressure message carries its note number, but many instruments map all poly pressure to a single value anyway. With `Channel`, only one pressure value is sent per channel, so several fingers compete for that stream. Enable MPE for true per-note pressure independence.

6.4.2 Vibrato (Pitch Bend)

The **Vibrato** block controls how horizontal finger movement on a key maps to pitch bend. The pitch bend range (**PB Range**) must match the setting in your synthesizer for accurate semitone tracking.

Parameter	Range	Default	Description
Enable	on / off	on	Sends pitch bend from finger movement
Style	`Linear` / `Quadratic` / `Far-Quadratic`	`Linear`	Mapping curve from position to pitch bend
Strength	slider		Amount of pitch bend for a given movement
Return Speed	`None` to `Max`		How quickly the bend returns to center
PB Range	1 to 96 semitones	12 (48 with MPE)	Semitone range of the pitch bend message

Turning MPE on sets **PB Range** to 48; turning it off resets it to 12.

6.4.3 Glissando

The **Glissando** block (keyboards only) controls pitch while a finger slides from key to key:

Parameter	Range	Default	Description
Enable	on / off	on	Enables glissando
Tune Loc	`Pad` / `Finger`	`Pad`	Reference point for zero pitch bend: the center of the key, or where the finger landed
In Tune Width	`0` to `100` %	`50` %	Width of the in-tune zone as a percentage of key width
Retrigger	on / off	off	Sends a new Note On when crossing to a new key while sliding

Parameter	Range	Default	Description
Vertical Gliss.	on / off	on	Lets vertical movement between rows contribute to the glide
Smoothing	`Low` to `High`		Smoothing of the glide

> **Tip:** Set **In Tune Width** to `100 %` for clean semitones without microtonal drift inside a key.

6.4.4 Sound Brightness (CC 74)

Keyboards have a **Sound Brightness (CC 74)** section that sends CC 74 from a chosen **Gesture**: `None`, `Pressure`, `X Absolute`, `Y Absolute`, `X Relative`, `Y Relative` or `Motion Speed`. It also has an **Initial value**, **Min/Max**, **Intensity**, **Smoothing**, **Tracking** and **Filter**. It is off (`None`) on new keyboards.

6.5 CC Mapping

Keys, keyboards and faders can send Continuous Controller messages from touch position and pressure. The available CC slots are:

CC Slot	Axis	Element Types
Pressure	Contact force (Z)	Key, keyboards, Fader 1D, Fader 2D
X abs	Horizontal position within the key or element	Key, keyboards, Fader 2D
Y abs	Vertical position within the key or element	Key, keyboards, Fader 1D, Fader 2D
X rel	Horizontal movement from where the finger landed	Key, keyboards
Y rel	Vertical movement from where the finger landed	Key, keyboards

CC Slot	Axis	Element Types
Motion	Finger movement speed	Key, keyboards, Fader 1D, Fader 2D

Each slot has an **Enable** switch, a controller **Type** (see below), a controller number, **Min/Max** output range (`0` to 127), **Intensity**, **Smoothing**, **Tracking** and **Filter**. Disabled slots send no data. Absolute slots follow the finger's position across the key or element. Relative slots start from an **Initial value** (default 64) when the finger lands and move with its displacement. Motion follows the finger's movement speed, smoothed and normalized from `0` to `100 cm/s`.

Button elements of the Control Change type send an **On** message and an **Off** message, each with its own controller and value, and each can be disabled.

> **Tip:** Map **Y abs** on a tall key to filter cutoff to create a ribbon-style expression strip within a single pad.

6.6 High-Resolution CC and NRPN

Every controller slot has a **Type**:

- **CC:** standard 7-bit Control Change on controller `0` to 127.
- **CC 14-bit:** high-resolution Control Change on controller `0` to 31. Erae 2 sends the value as an MSB on the chosen controller and an LSB on the controller plus 32, following the MIDI specification.
- **RPN** and **NRPN:** Registered and Non-Registered Parameter Numbers, addressed with a **Bank** and an **Index** (`0` to 127 each).

The receiving software must support 14-bit CC, RPN or NRPN to use the extra resolution.

> **Tip:** To send a 14-bit value, set the slot's **Type** to `CC 14-bit` and pick a controller between `0` and 31.

6.7 MIDI 1.0 vs MIDI 2.0: Which Values You Actually Read

Erae 2 captures and processes touch at high internal resolution, but what arrives in your software depends on which protocol the connection is actually using, and that is decided between your computer and the device, not by the device alone.

- **MIDI 1.0 (the default).** The two USB MIDI cables (`Erae 2 MIDI` and `Erae 2 MIDI (MPE)`) carry standard MIDI 1.0. A MIDI 1.0 receiver sees 7-bit Control Change values (`0` to `127`). To get more than 7 bits over MIDI 1.0, use the `CC 14-bit`, RPN or NRPN controller types described above: the receiver must understand them to reconstruct the high-resolution value.
- **MIDI 2.0 (alternate USB mode).** Turning on **MIDI 2.0** in **Settings** (the device reboots) puts the device into a USB MIDI 2.0 mode that exposes the Universal MIDI Packet (UMP) stream, which can carry full high-resolution values natively.

> **Reading high-resolution values requires both ends to agree.** Enabling MIDI 2.0 on the device is only half of it. Your host or operating system must negotiate the MIDI 2.0 protocol with the device and your software must read from the MIDI-2.0 endpoint/stream. If the host stays on MIDI 1.0, or your software reads the device through a MIDI-1.0 port, you will still see 7-bit CC values even though MIDI 2.0 is switched on. If you expected high-resolution data and only see `0` to 127, confirm that (1) MIDI 2.0 is on and the device has rebooted, (2) your OS/host reports the device as a MIDI 2.0 endpoint, and (3) the track or tool is reading the MIDI-2.0 stream rather than a legacy MIDI-1.0 view of the same device.

6.8 MPE Configuration

MPE (MIDI Polyphonic Expression) allows each finger to have independent pitch bend, pressure, and slide on its own MIDI channel, enabling per-note expression across polyphonic patches.

Erae 2 implements two MPE zones defined by the MIDI MPE specification. The zone is set **per keyboard element** with the **MPE Master** setting:

MPE Master	MPE Zone	Member Channels
Channel 1 (default)	Lower Zone	Channels 2 and up (allocated dynamically)
Channel 16	Upper Zone	Channels 15 and down (allocated dynamically)

When MPE is enabled, the keyboard's **Channel** setting becomes **Nb channel**: the number of member channels the keyboard uses (1 to 15). Turning MPE on also sets **PB Range** to 48.

MPE messages are always sent on the `Erae 2 MIDI (MPE)` port. See the USB MIDI Ports section at the top of this chapter.

MPE-relevant settings per keyboard element:

Setting	Recommended MPE Value
MPE	on
MPE Master	`Channel 1` (Lower Zone) or `Channel 16` (Upper Zone)
PB Range	48 semitones
Pressure **Type**	`Channel` for MIDI 1.0 MPE; `Poly` where MIDI 2.0 per-note pressure is used
Slide	CC 74, from the **Y rel** CC slot or the **Sound Brightness (CC 74)** section

> **Tip:** Most MPE synthesizers expect pitch bend range to be set identically on both the controller and the instrument. Set **PB Range** to 48 in the element editor and use the instrument's own MPE setup page to match.

6.8.1 Getting Slide and Pressure Into Your DAW

Slide and pressure are per-note expression, so they only survive the trip into your instrument if the whole chain is in MPE mode. If notes play but slide and pressure do nothing, the cause is almost always a non-MPE link in the chain, not a setting on the device. Check each step:

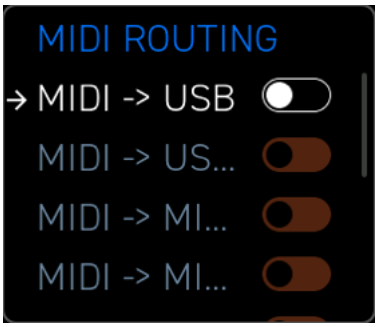
1. **Read from the MPE cable.** Point the DAW track's MIDI input at **`Erae 2 MIDI (MPE)`**, not the main port. Per-note expression is only sent on the MPE cable.
2. **Pass all 16 channels.** Set the track's input channel to **All Channels** (or **Any**). MPE spreads each finger across its own member channel; if the DAW folds incoming MIDI to a single channel, every note's expression collapses onto one.
3. **Enable MPE on the track and in the instrument.** Switch the track to MPE mode (where the DAW has an explicit toggle) and enable MPE / Per-Note / Poly Expression in the instrument itself.
4. **Map the slide axis to the instrument's slide.** Slide is normally CC 74 (Timbre). Confirm the keyboard sends CC 74 for vertical slide (see the MPE settings table above) and that the instrument listens to CC 74 as its slide/timbre input.
5. **Match the pitch bend range** on both sides (48 semitones by default) so pitch tracks correctly while sliding.

For per-DAW, step-by-step instructions covering Ableton Live, Logic, Cubase, Bitwig, Reaper, and FL Studio, see Appendix C: DAW Setup Guides.

> **Pitfall:** Most DAWs collapse incoming MIDI to a single channel before handing it to a non-MPE track, which silently discards slide and pressure. If you receive on the main port instead of `Erae 2 MIDI (MPE)`, or leave the track in standard (non-MPE) mode, you will hear notes but no expression.

6.9 MIDI Routing Matrix

The routing matrix controls which physical ports relay MIDI **input** messages to other ports, independently of element output destinations. This allows Erae 2 to act as a MIDI merge and thru box.



MIDI Routing

Open it from **Settings > Routing**; the screen is titled **MIDI ROUTING**. Each row is a toggle switch:

Switch	Source -> Destination	Effect
MIDI -> USB Host	MIDI In -> USB Host Out	Forward hardware MIDI In to connected USB devices
MIDI -> USB Device	MIDI In -> USB Device Out	Forward hardware MIDI In to the host computer
MIDI -> MIDI A	MIDI In -> MIDI A Out	MIDI Thru to MIDI A
MIDI -> MIDI B	MIDI In -> MIDI B Out	MIDI Thru to MIDI B
USB Device -> USB Host	USB Device In -> USB Host Out	Route host computer MIDI to connected USB device
USB Device -> MIDI A	USB Device In -> MIDI A Out	Route host computer MIDI to hardware gear on MIDI A

Switch	Source -> Destination	Effect
USB Device -> MIDI B	USB Device In -> MIDI B Out	Route host computer MIDI to hardware gear on MIDI B
USB Host -> USB Device	USB Host In -> USB Device Out	Route connected USB device MIDI to host computer
USB Host -> MIDI A	USB Host In -> MIDI A Out	Route connected USB device to hardware gear on MIDI A
USB Host -> MIDI B	USB Host In -> MIDI B Out	Route connected USB device to hardware gear on MIDI B

Turn a switch **on** to enable that routing path. All routing switches are independent: multiple sources can feed the same destination.

> **Tip:** To use Erae 2 as a simple 2-port MIDI interface, enable **MIDI -> USB Device** and **USB Device -> MIDI A**. Your DAW can then send to and receive from hardware synths through Erae 2 without any additional interface.

Routing settings are saved with the **project**, so each project can carry its own merge configuration.

6.10 Related Topics

- Elements: per-element type configuration details including button CC and program change setup
- LCD Interface: navigating menus on the LCD display
- Settings: project-level settings including MIDI clock and sync
- MIDI Implementation: complete message reference table

7 Scales & Tuning

Scale settings are **per keyboard element**, not global (unless you opt in to the global scale, see below). Each keyboard element in a layout stores its own scale and base note (root and octave). The Scale screen edits the settings for the **selected element**.

The Scale screen controls how the selected keyboard element interprets touch coordinates as musical pitches.

Scale Selector

Press the **Scale** button on the front panel to open the Scale screen for the selected keyboard; press it again to close it. To pick a different keyboard, hold **Scale**: everything except the selected element dims, and touching a keyboard selects it. Release **Scale** and the Scale screen opens for that keyboard.

The title row shows **SCALE**. Below it are the scale list, the **ROOT** and **OCTAVE** values and, on a Keygrid, the **GLOBAL** and **OFFSCALE** switches. Turn the encoder to move the highlight in that order (scale, root, octave, **GLOBAL**, **OFFSCALE**). Click a value to edit it, turn to change it and click again to confirm; click a switch to flip it.

7.1 Built-In Scale Library

The scale list lists the factory scales. Factory scales are fixed and cannot be edited or deleted.

The library contains 15 factory scales:

1. `Chromatic`: all 12 semitones
2. `Major`: the standard diatonic major scale

3. `Minor`: natural minor (Aeolian mode)
4. `Melodic Minor`: raised sixth and seventh
5. `Harmonic Minor`: raised seventh degree
6. `Arabic`
7. `Dorian`
8. `Phrygian`
9. `Gypsy`
10. `Mixolydian`
11. `Romanian`
12. `Gypsy Minor`
13. `Japanese`
14. `Spanish`
15. `Blues`

A Keygrid (Iso Keyboard) lists the 14 scales without `Chromatic`: to play every semitone on a Keygrid, turn **Show Offscale** on instead. The Chroma Keyboard lists all 15.

> **Tip:** On a Chroma Keyboard, `Chromatic` means no scale coloring: every semitone looks like a plain piano key.

7.2 Root Note

The **ROOT** value sets the tonal center of the selected scale. Twelve values are available: `C`, `C#`, `D`, `D#`, `E`, `F`, `F#`, `G`, `G#`, `A`, `A#`, `B`.

The root note determines which pitch class is treated as the first degree (the tonic) and is the note of the bottom-left pad. Changing the root transposes the entire scale without altering its interval structure. The Drumpad and the Chroma Keyboard both show ROOT and OCTAVE.

A custom scale with a frequency root sets its own root: the Scale screen then shows that frequency instead of a root note (see Custom Scales).

> **Tip:** On a Keygrid, pads that fall on the root note use the first scale degree style, making the tonal center immediately visible on the surface.

7.3 Octave

The **OCTAVE** value sets the octave of the keyboard's base note, from -2 to 8. With root `C`, octave 3 puts middle C (MIDI note 60) on the bottom-left pad; octave -2 starts at MIDI note 0.

Outside the Scale screen, the **Plus** and **Minus** buttons also move the selected keyboard up or down by one octave.

> **Tip:** Use the octave buttons to bring a keyboard into the range your synthesizer expects while you play.

7.4 Show Offscale

The **OFFSCALE** switch (bottom row of the Scale screen, Keygrid only) is the Show Offscale setting. It controls whether off-scale semitones appear on a Keygrid.

- **On:** all 12 semitones are playable. Notes in the scale receive the scale-degree color styles; notes outside the scale receive the off-scale style.
- **Off:** only scale degrees are shown. The surface re-maps so that every physical cell plays a note that belongs to the selected scale; no off-scale semitones are accessible.

> **Tip:** Turning Show Offscale off makes it impossible to play a "wrong" note, which is ideal for live performance, improvisation, and beginners.

7.5 How Scales Affect Each Keyboard Type

Keyboard Type	Scale support	Show Offscale toggle?	Global scale?	Custom scales?
Iso Keyboard (Keygrid)	Full: filters and highlights scale degrees, sets row intervals	Yes	Yes	Yes, with MPE on
Chroma Keyboard	Coloring only: all 12 semitones stay playable	No	No	No
Drumpad	None: always chromatic; scale list hidden	No	Yes (stays chromatic)	No

Iso Keyboard has the richest scale interaction: the scale also controls the row intervals. When Show Offscale is off, each row step is counted in scale degrees rather than semitones, so the iso-

morphic geometry is preserved within the scale.

Chroma Keyboard uses the scale only for coloring. The piano layout is fixed and always uses 12-tone equal temperament.

Drumpad ignores the scale; its Scale screen shows only the ROOT and OCTAVE lists. Each pad maps to consecutive MIDI notes starting from the base note.

7.6 Global Scale

Normally every keyboard element carries its own scale. When you want one scale to rule the whole project, opt keyboards in to the **global scale**: turn on the **GLOBAL** switch on the Scale screen, or **Use global scale** in Erae Lab. Changing the scale of any opted-in keyboard, on any layout, changes it for all of them.

- Only the scale is shared. Root note and octave stay individual to each keyboard.
- Turning the switch off detaches the keyboard: it keeps its current scale and no longer follows later changes.
- The Chroma Keyboard cannot opt in. A Drumpad can opt in, but it always stays chromatic.

7.7 Custom Scales

Besides the factory scales, a Keygrid can use a **custom scale**: any number of notes per octave, tuned freely by ratio or as an equal division of the octave, including microtonal tunings. Custom scales are created and edited in Erae Lab or on the device under **Settings > Custom Scales**, and they are shared by all projects.

- Custom scales are offered only on a Keygrid with **MPE** enabled. Each microtonal note is played as a base note plus its own pitch bend on an MPE member channel, so the tuning stays correct in chords.
- On such a keyboard, the scale list shows the factory scales, then your custom scales, then **NEW SCALE**, which creates a scale and opens the on-device editor.
- While a custom scale is selected on the Scale screen, the **Settings** button LED pulses slowly: press **Settings** to edit that scale on the device. **Return** brings you back to the Scale screen.
- Turning MPE off on a keyboard that uses a custom scale returns it to a factory scale.
- A custom scale whose root is set as a **Frequency** sounds its root at that frequency, whatever the keyboard's root note. While such a scale is selected, the Scale screen shows the root frequency (for example 432 Hz) in place of the **ROOT** value: it cannot be edited there, and the encoder goes from the scale list straight to **OCTAVE**. **OCTAVE** still moves the keyboard by whole periods of the scale.

See the Erae Lab User Manual, Chapter 18 for the full guide to creating, importing and editing custom scales, including the on-device editor.

7.8 Per-Element Scale Settings

Each keyboard element stores its own scale assignment. The Scale screen always edits the scale of the **selected element**. To configure different scales on different elements, hold **Scale**, touch the next keyboard, and release.

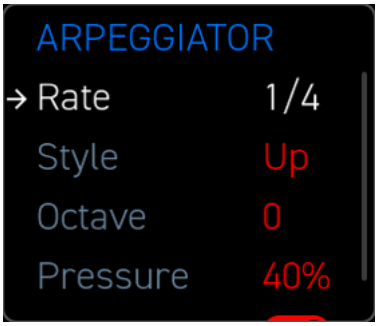
> **Tip:** You can create a layout where one element plays the Japanese scale while another plays Blues or Major, which is useful for layered instruments or split-surface setups.

8 Arpeggiator & Looper

Erae 2 provides two real-time performance tools: the **Arpeggiator**, which generates melodic patterns from held notes, and the **Looper**, which records and replays your playing as a repeating pattern synchronized to the project clock.

8.1 Arpeggiator

The Arpeggiator turns held notes into a rhythmic sequence. Each keyboard element (Keyboard, Keygrid or Drumpad) has its own arpeggiator.



Arpeggiator

Select a keyboard, then press the **Arp** button on the front panel. This turns the arpeggiator of that keyboard on (or off) and opens the **ARPEGGIATOR** screen. To open the screen without switching the arpeggiator on or off, hold **Settings** and press **Arp**.

8.1.1 Enabling the Arpeggiator

Each press of **Arp** switches the selected keyboard's arpeggiator on or off; the **Arp** button lights green while it is on. When the arpeggiator is off, notes play normally. When it is on, held notes are not sounded directly: the arpeggiator plays them one at a time instead. The on or off state is saved with the layout.

The arpeggiator also applies to notes played by the step sequencer on that keyboard: a chord drawn in the sequencer sounds as an arpeggio while the arpeggiator is on.

8.1.2 Rate

Rate sets how fast arpeggiated notes follow each other. Values:

`1/32`, `1/16`, `1/8`, `1/4`, `1/2`, `1/1`, `Pressure`

Rates are fractions of a beat (a quarter note) at the project tempo: `1/1` plays one note per beat, `1/4` plays four notes per beat (sixteenth notes), and `1/32` plays thirty-two notes per beat.

In `Pressure` mode, the rate follows how hard you press: light pressure plays about one note per beat, and pressing harder speeds the arpeggio up, to as fast as `1/32`.

> **Tip:** `1/4` gives a sixteenth-note arpeggio, the most common starting point. Use `Pressure` mode for an expressive arpeggio whose speed you control with your fingers.

8.1.3 Style

Style determines the order in which held notes are played:

- `Up`: notes are played from lowest to highest pitch, then repeat
- `Down`: notes are played from highest to lowest pitch, then repeat
- `UpDown`: notes ascend then descend before repeating; the turning note at the top and bottom is **not** played twice
- `UpAndDown`: notes ascend then descend before repeating; the turning note at the top and bottom **is** played twice (once on the way up, once on the way down)
- `Random`: each step picks a random note from the held set

> **Tip:** `UpDown` over a four-note chord produces a six-step pattern, which creates interesting phasing when the pattern length does not divide evenly into the bar. Use `UpAndDown` when you want the turning notes accented by their doubled appearance.

8.1.4 Octave Range

Octave sets how many extra octaves the arpeggiator spans above the played notes. Range: `0` to `8`.

At `0`, the arpeggio stays within the register of the held notes. At `1`, the arpeggiator plays the held notes, then repeats them an octave higher, before cycling back.

> **Tip:** A `Down` style with Octave 3 produces a cascading arpeggio that spans several octaves, a dramatic effect on pads and plucked sounds. Set Octave to `0` for tight, single-register arpeggiation.

8.1.5 Pressure to Velocity

Pressure sets how much the current finger pressure shapes the velocity of each arpeggiated note. Range: `0%` to `100%`, in steps of `10%`. Default: `40%`.

At 0%, arpeggiated notes use the velocity of your initial strike. At 100%, the velocity of each note follows how hard you are pressing at that moment. Values in between blend the two.

> **Tip:** Set Pressure to 60% to 80% for expressive playing where subtle changes in finger weight create natural accents, while keeping a stable overall dynamic.

8.1.6 Quantize

The **Quantize** toggle decides when the arpeggio starts. It is on by default.

- **On:** the arpeggio starts on the next step of the project clock grid, so it stays aligned with the tempo and with other parts. A note released before the next grid step does not sound.
- **Off:** the first note plays immediately when you touch, and the arpeggio runs from your touch rather than from the grid.

In both cases the rate follows the project tempo.

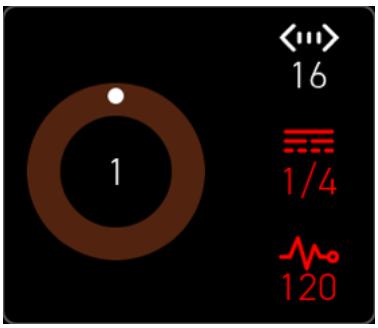
> **Tip:** Keep Quantize on when playing along with a DAW or the looper. Turn it off when you want very short touches, or short sequencer steps, to sound.

8.2 Looper

The Looper records your touch gestures on a layout, including pressure and slide, and replays them in a loop in sync with the project clock. Because it records gestures rather than MIDI, a replayed loop goes through the layout's elements again: change an element's settings and the loop follows.

Each layout has its own loop. The Home screen shows the loop of the layout you are on.

8.2.1 Home Screen Looper Display



LCD Home Screen

The **Home** screen shows:

- **Loop circle:** the layout number, with an arc showing the playback position in the loop. The arc turns a different color when the layout's loop is muted.
- **Length:** the loop length in steps (one step is a sixteenth note).
- **Quantize:** the quantize grid applied to recorded gestures.
- **Tempo:** the project tempo in BPM.

8.2.2 Record and Playback

The **Play/Rec** and **Stop** buttons control the transport:

- When the internal clock is stopped, **Play/Rec** starts it. The **Play/Rec** button lights green while the clock runs.
- While the clock runs (or when the project follows an external clock), **Play/Rec** arms or disarms recording. While armed, the button lights red, and everything you play on the current layout is recorded into its loop.
- **Stop** stops the clock. With the internal clock, stopping also disarms recording. The recorded loop is kept.

Recording always overdubs: while armed, new gestures are layered onto what the loop already holds. Each pass through the loop becomes a layer.

Undoing a layer. On the Home screen, press **Return** to remove the most recently recorded layer.

Clearing loops. Hold one or more layout buttons (**N1** to **N8**) and press **Return** to clear their loops. To also clear the sequences, keep **Return** held for about a second: the screen names the layouts and asks you to confirm, and **Yes** clears their loops and all eight sequencer patterns of each.

Muting loops. Hold **Stop** and press a layout button to mute that layout's loop; hold **Play/Rec** and press a layout button to unmute it. A muted layout's button is shown in red.

> **Tip:** Because recording always overdubs, short loops fill up quickly. Set the Length before you start recording.

Bouncing a loop into the Step Sequencer. A recorded loop can be converted into an editable sequencer pattern with the **Bounce Loop** row in Sequencer Settings. See Chapter 9.

8.2.3 Looper Controls

The looper controls are on the **Home** screen. Turn the encoder to highlight a control, click to edit it, turn to change the value, and click again to finish.

Length: the loop length in steps, from 1 to 256 (default 16). While editing Length, **Plus** doubles it and **Minus** halves it.

Quantize: the grid that recorded gestures snap to. See Quantize Grid below.

Tempo: the project tempo in BPM. While editing Tempo, **Plus** doubles it and **Minus** halves it. Tempo can only be changed here when the clock source is internal.

> **Tip:** Record with Quantize `Off` to keep natural expressive timing, or with `1/4` to tighten a rhythmic part to sixteenth notes as you record.

8.2.4 Clock Sync

The Looper is always locked to the project clock. When the project clock source is `INT` (internal), the Looper runs at the project tempo. When the clock source is `USB-dev`, `MIDI`, or `USB-host`, the Looper follows the incoming MIDI clock, allowing tight synchronization with a DAW or hardware clock source.

See Chapter 15: Settings for clock source configuration.

8.2.5 Pattern Length

Pattern length is set with the **Length** control on the Home screen, in steps of a sixteenth note, up to 256 steps (16 bars of 4/4).

8.2.6 Quantize Grid

The **Quantize** grid snaps the start of each recorded gesture to the nearest grid position when the gesture is recorded. Values:

`Off`, `1/16`, `1/12`, `1/8`, `1/6`, `1/4`, `1/3`, `3/8`, `1/2`, `3/4`, `1`, `5/4`, `2`, `3`, `4`

Values are fractions of a beat (a quarter note): `1/4` snaps to sixteenth notes, `1` to quarter notes, and `4` to whole bars of 4/4. The default is `1/4`.

Quantization is applied as you record. Changing the grid affects gestures you record afterwards, not the ones already in the loop.

> **Tip:** Only the start of each gesture is moved by the grid. Pressure, slide and other expression inside the gesture keep their natural timing.

8.2.7 Tap Tempo

Tap Tempo sets the project tempo by tapping in time. It works when the clock source is internal. To use Tap Tempo:

- Hold **Sequencer** and click the encoder once per beat.
- Configure a pedal (Switch type) with the **Tap Tempo** message type. Each press of the pedal taps the tempo.
- Use a **Button** element on the touch surface set to **Tap Tempo** (see Button).

Erae 2 measures the interval between successive taps and updates the project tempo. The Looper and Arpeggiator follow the new tempo immediately.

9 Chapter 9: Step Sequencer

The Step Sequencer is a pattern based MIDI sequencer built into the surface. Where the Looper captures a free performance, the Sequencer lets you build patterns two ways: draw and edit them step by step directly on the LED grid, or record them live into the pattern while the clock runs. You get independent control over pitch, velocity, gate length, and continuous expression for every step. Its state is saved with your project, per layout.

9.1 Entering and Leaving the Sequencer

Press the **Sequencer** button to enter sequencer mode. The touch surface switches to the three zone sequencer view (see below) and the LCD shows the Sequencer screen.

To choose which element to sequence as you enter, hold **Sequencer**: everything except the selected element dims, and touching an element selects it instead of playing it. Release **Sequencer** and sequencer mode opens on that element.

The button cycles through three cases:

- Sequencer mode is off: pressing **Sequencer** enters it.
- Sequencer mode is on and the LCD is showing the Sequencer or Sequencer Settings screen: pressing **Sequencer** exits sequencer mode and returns the LCD to Home.
- Sequencer mode is on but the LCD is showing another screen (Home, Scale, and so on): pressing **Sequencer** brings the Sequencer LCD screen back without leaving the mode.

If the active layout contains nothing the sequencer can record (for example a layout holding only API zones), the button does nothing: the sequencer needs at least one sequenceable element to open.

Leaving sequencer mode commits any note you were still drawing, so an edit finished right at the exit is never lost.

9.2 What You Can Sequence

The Sequencer does not only record keyboards. It captures four element kinds, and the lanes it shows change to match whichever element is focused:

Element	What a step records	Lanes offered
Keyboard (scale, chroma, drumpad, and the dedicated Sequencer Keyboard)	A note: pitch, velocity, gate length, and expression	Note, Pressure, Slide, Pitch, plus one lane per enabled key gesture (pressure CC, X abs, Y abs, X rel, Y rel)
Fader 2D (an XY pad)	X, Y, and Z (pressure) values	X, Y, Z
Fader 1D (a single fader)	One value automation	Value
Button	An on or off state per step (a press with duration)	State (shown as `steps`)

The sequencer focuses whichever layout element you touched last. If nothing suitable is focused when you enter, the first sequenceable element in the layout is picked automatically.

To change which element you are sequencing, hold that layout's button (**N1** to **N8**) on its own, without holding Sequencer. The surface temporarily shows that slot's normal full layout so you can tap the element you want. Release the button and the three zone sequencer view returns, bound to your choice. Note the difference between a hold and a tap: a quick tap and release of your current layout's own button leaves sequencer mode entirely. Pressing **Stop**, **Play/Rec** or **Return** while you hold a layout button mutes, unmutes or clears that layout's loop as usual, and the sequencer stays on the layout you were on.

Pressing the button of another layout while in sequencer mode switches to that layout and stays in sequencer mode, reopening on the element you last sequenced on that layout.

9.3 The Surface: Three Zones

While the Sequencer is active the touch surface is split left to right into three fixed zones, each spanning the full height:

- **Live**, the leftmost 7 columns: a live, playable mirror of the focused element. See The Live Zone below.
- **Grid**, the 32 columns in the middle: the step grid where you draw and edit.
- **Launcher**, the rightmost 3 columns: eight stacked pads, one per pattern slot, used to switch patterns. The top pad is pattern 1, the bottom pad is pattern 8.

With **Swap Sides** on (see Sequencer Settings) the Launcher moves to the left edge and Live moves to the right; the grid stays 32 columns wide in the middle.

Each finger's zone is fixed at the moment you touch down. A gesture that begins in the grid keeps editing even if the finger slides toward the launcher, so the zones never cross trigger mid gesture.

9.3.1 The Live Zone

The Live zone is a playable mirror of the focused element, and it records. Whether it records depends on whether recording is armed, which is the same **Play/Rec** arming used by the Looper:

- **Recording armed** (Play/Rec on): playing in the Live zone captures straight into the active pattern at the current playhead. A keyboard records notes with their pressure curve, release velocity, and true held length; a Fader 1D records value automation; a Fader 2D records X, Y, and Z; a Button records on steps at the playhead. Recording over a span replaces what was there.
- **Recording not armed**: the Live zone plays sound and MIDI without recording, so you can audition a part or perform along while a pattern runs.

Arming is read live, moment by moment, so you can punch in and out during a take: it is never latched to the start of a gesture.

For a keyboard the Live mirror is rotated a quarter turn compared with the normal layout: pitch runs vertically up the strip, so slide up for higher notes and down for lower ones. The mirror shares the grid's pitch scroll, so a lit playable row lines up with the same grid row. Fader and button mirrors render the normal way up and do not move with the pitch scroll.

9.4 What a Cell Means: the Pitch Grid

For keyboard patterns the grid is a chromatic piano roll. The vertical axis is one MIDI semitone per row, each row two LEDs tall, and rows are shaded like a piano: white key semitones glow dimly, black key semitones stay dark. Higher rows are higher pitches. A note is drawn as a small amber head plus a tail:

- **Horizontal position** is when the note starts.
- **Width** is the gate length, how long the note holds.
- **Vertical row** is the pitch.
- **Head brightness** is the velocity, how hard the note hits. The tail brightness follows the recorded pressure.

While the playhead crosses a note the note brightens.

9.4.1 Scale Relative Recording

How a note is stored depends on the keyboard:

- On a **scale (in-scale) keyboard**, a note stores a scale degree, not a fixed pitch. Degrees below the root are allowed. The actual pitch is worked out at playback against the element's

current scale and root, so if you later change the scale or transpose the root, the whole recorded pattern follows. A recorded degree sounds identical to playing it by hand.
- On a **chroma keyboard or drumpad**, a note stores an absolute pitch or pad index and stays fixed.

Whether a keyboard is in-scale or chromatic is set by the **Show Offscale** toggle on the keyboard element. Off means one key per scale degree; on means chromatic keys are shown and steps are stored as semitone positions.

Microtonal custom scales play back in tune, exactly as when you play them live.

9.5 Editing Steps on the Grid

The grid gesture depends on where you touch and which way you drag.

To draw a note, touch empty space and drag:

- The row under your finger sets the pitch.
- The pressure of your landing touch sets the velocity (press harder for a louder note). You can adjust it afterwards with a vertical drag from the note head.
- How far you drag horizontally sets the gate length, snapped to the quantize division. A plain tap with no drag makes a note at the minimum length.
- Vertical drift while you draw is recorded as a pitch bend curve on that note. Small wobble does not count: as long as your finger stays within one cell of where the note started, the note stays perfectly straight and no bend is recorded. Once you leave that cell the bend is captured from that point on, even if you drift back.
- Moving up or down on the start cell before dragging right sets the bend of the very first pulse, so a note can start away from its row pitch and slide into it. The head stays on its row; a smaller marker in the tail colour shows where the note starts, linked to the head by a dim bridge, and it follows your finger while you hold.
- Dragging right along the row and then moving up or down at the end gives a straight bend from the row pitch to where you stopped. Dragging back to the left erases the bend past your finger, so shortening and re-lengthening a note redraws it.
- A brief loss of contact while drawing still yields one note, not two: touching down again right away, close to where you lifted, continues the same note.

On an existing note head:

- A **tap** deletes the note.
- A **drag up or down** edits its velocity (bottom of the grid is 0, top is maximum).
- A **drag to the right** edits the focused expression curve for that note (which curve depends on the lane selected in the LCD header). On the Note lane this redraws the pitch bend and resizes the note with your drag, with the same one cell wobble tolerance as drawing.

On the Pitch lane, the grid edits only the pitch bend of existing notes, the way the Pressure and Slide lanes edit their curves:

- Touching empty space does nothing.
- A **tap** on a note head resets its bend curve. The note stays.
- Holding the note head and moving up or down sets the bend of the first pulse, one row per semitone, without changing the note length.
- Dragging along the note paints the bend where your finger goes and leaves the rest of the curve as it was. There is no wobble tolerance here: this lane is for setting exact values.

Each note carries several recorded curves at once: a pitch bend curve, a pressure curve, a slide curve, and one curve per enabled key gesture (pressure CC, X abs, Y abs, X rel, Y rel). Velocity, pressure, and curve values are stored at high resolution, finer than the 0 to 127 MIDI range, so expression is smooth.

While you hold a note to edit its curve, the curves of the other notes dim so the one you are working on stands out. They return to full brightness when you lift your finger.

Gesture lanes play back as the same CC messages the live keys send, on the same channel, using the gesture's own range, and they are kept when you bounce a loop into a sequence.

For a **Fader 1D** the grid records a value curve; for a **Fader 2D** it records X, Y, and Z; for a **Button** a tap toggles the on or off state of a step.

9.5.1 Selecting and Moving a Group of Notes

To move several notes at once, hold one finger still on empty grid space and touch a second finger: the two fingers define a box that selects every note inside it. Drag the second finger to move the whole selection, both in time and in pitch. If the box catches no notes, both fingers fall back to normal editing.

9.6 Moving Around a Large Pattern

The 32 column grid is a window onto the pattern, not the whole thing. A pattern can be up to 256 steps, so on anything longer than the window you see only part at a time and scroll or zoom to reach the rest.

- **Grab scroll:** hold the **Sequencer** button and drag a finger on the surface. The content follows your finger in both directions. This is the main way to move through a long pattern. Without holding Sequencer, a drag in the grid edits notes instead.
- **Octave scroll:** the **Plus** and **Minus** buttons jump the pitch window up (Plus) or down (Minus) by one octave per press. On a scale keyboard the window steps by scale degrees and lands on the next root. Octave scroll is ignored while the encoder switch is held; while you are editing the pattern Length in the footer, Plus and Minus change the length instead.

- **Zoom:** hold **Sequencer** and press **Plus** to zoom in (fewer bars visible) or **Minus** to zoom out (more bars). The widest zoom out always frames the entire pattern.
- **Encoder scroll:** hold **Sequencer** and turn the main encoder to nudge the window sideways, four columns per click. With Sequencer not held, turning the encoder moves the LCD focus and edits instead (see The Sequencer LCD).

When you raise the pattern Length, the window re-fits so the new steps are visible right away. If you had deliberately zoomed or scrolled to a region, your view is kept instead.

Follow Playhead (a Sequencer Setting, off by default) auto scrolls the grid to keep the moving playhead on screen, re-centering on it when it leaves the window. Leave it off to keep editing a fixed region while the pattern plays.

Fader and button patterns have no pitch axis, so for them only horizontal (time) scrolling applies.

9.7 Sequencer Button Shortcuts

Hold the **Sequencer** button and press another control for these performance shortcuts. They work alongside the scroll and zoom gestures above:

Hold Sequencer and press	Action
N1 to N8	In sequencer mode, switch the active pattern to that slot. Out of sequencer mode, launch that slot as a scene across layouts. This shortcut never bounces a loop: use the **Bounce Loop** row in Sequencer Settings for that.
Encoder switch	Tap tempo (internal clock only).
Play/Rec	Mute or unmute the active pattern. Double-click to unmute every pattern.
Stop	Mute or unmute the active pattern. Double-click to mute every pattern.
Plus or **Minus**	Zoom in or out (see Moving Around a Large Pattern).
Return	Clear the focused element's track (every pattern of that element). This is destructive, so take care not to hit it while reaching for scroll or undo.

Outside of the chords, the transport buttons keep their normal role in sequencer mode: **Play/Rec** starts the clock when it is stopped on the internal source, and otherwise toggles record arming; **Stop** stops the clock.

9.8 The Sequencer LCD

The LCD is a fixed three band screen: a Header on top, the Grid in the middle, a Footer along the bottom. Turn the main encoder (with the Sequencer button not held) to move the highlight across the three selectable controls, in the order Header, then Division, then Length, wrapping around. Click the encoder to start editing the highlighted control, and click again to stop. While a control is in edit, its value blinks and its icon stays lit.

9.8.1 Header: the Lane Selector

The header shows one expression lane at a time as an icon in a white disc plus a lowercase label. To change lane, highlight the header, click the encoder to enter edit, turn the encoder to cycle lanes, and click again to leave edit.

Only the lanes that make sense for the focused element are offered:

Element	Lane cycle	Labels
Keyboard	Note, Pressure, Slide, Pitch, then one lane per enabled key gesture (wraps)	note, pressure, slide, pitch, pressure CC, X abs, Y abs, X rel, Y rel
Fader 2D	X, Y, Z (wraps)	X, Y, Z
Fader 1D	Value only	value
Button	State only	steps

The **Note** lane shows notes as bars at their pitch rows. **Pressure**, **Slide**, and the gesture lanes draw the recorded curve on top of every note, and the note head dims so the curve reads as the foreground. **Pitch** shows each note's pitch bend as the shape of its tail (a note with no bend shows a flat bar) and lets the grid edit that bend without creating, deleting or resizing notes. The gesture lanes appear only for the gestures enabled on the focused element (in its MIDI settings); disabling a gesture removes its lane, and if that lane was selected the header falls back to Note. Changing lanes re-points what the LED grid lets you edit and relabels the header, but the LCD grid keeps showing all of the element's data.

The header also has two special states. If a pattern runs out of memory it shows **Pattern full** in red for a couple of seconds. If the element you were sequencing gets deleted it shows **Element**

deleted in grey, rather than an empty lane.

9.8.2 Grid: the Map

The LCD grid always draws the whole pattern edge to edge. A translucent outlined box marks the slice currently mapped onto the pad surface: horizontally it tracks your time position, and for keyboards it also shrinks vertically to the pitch band on the surface, so it shows which octave your fingers are on. Faders and buttons have no pitch axis, so their box spans the full height and only moves sideways.

Notes appear as white bars whose vertical position shows relative pitch; the pitch axis auto scales to fit whatever notes exist. Fader lanes appear as a stepped line and dot curve of the recorded value (on a 2D fader the X axis is white and the Y axis is red). Button patterns show their on steps as centred bars. Step and bar gridlines thin out automatically on long patterns so they stay readable.

A strip down the right edge shows all 8 pattern slots: the active slot is a red square with its number, the rest are grey dots. A moving white line marks the playhead while the pattern plays and disappears when it stops. The window box and the slot strip are read only feedback; you switch patterns from the Launcher on the surface.

9.8.3 Footer: Division and Length

The footer holds three items: the quantize **division** on the left, a live thumbnail of the layout in the centre, and the pattern **length** on the right. The division and the length are both selectable: highlight one, click to edit, then turn the encoder to change the value.

Control	Values	Default
Quantize division	Off, 1/16, 1/12, 1/8, 1/6, 1/4, 1/3, 3/8, 1/2, 3/4, 1, 5/4, 2, 3, 4	1/4
Pattern length	1 to 256 steps, one step per encoder click	16 steps

While you are editing the Length, **Plus** doubles the length and **Minus** halves it (within 1 to 256 steps); they return to octave scrolling once you leave the edit.

The centre thumbnail is a small live picture of the layout: the element you are sequencing is highlighted, and each element flickers when its steps trigger, so you can see which tracks are firing.

9.8.4 Encoder and Return in This View

- **Click the encoder** to toggle editing of the highlighted control (lane, division, or length).
- **Return** exits an edit if one is in progress; otherwise it undoes your last sequencer edit.

9.9 Switching Patterns

Each layout has 8 independent pattern slots, each with its own length, quantize division, and notes. Tap a pad in the Launcher zone (top pad is pattern 1) to switch. Pads respond on touch down only. How the switch lands depends on **Switch Mode**, set in Sequencer Settings:

- **Wait:** the new pattern is queued and takes over when the current loop ends. The pending launcher pad blinks on the surface until it takes over.
- **Jump:** the new pattern takes over immediately, continuing from the same position in the bar.
- **Restart:** the new pattern takes over immediately and restarts from its first step.

9.10 Sequencer Settings

Press **Settings** while the sequencer view is active to open Sequencer Settings; press **Settings** again to return, or **Return** to leave for the default view. Turn the encoder to move between items, click to act.

Item	Description
Clear Pattern	Removes all step data from the active pattern only. Confirms with a message and returns you to the sequencer.
Clear Element	Removes all step data from every pattern of the focused element. Confirms with a message and returns you to the sequencer.
Switch Mode	Wait, Jump, or Restart, as above.

Item	Description
Copy Pattern	Copies the active pattern into another slot. The screen asks you to tap a pattern pad in the Launcher; tap the destination and the copy happens. The Launcher pads show which slots are empty and which already hold notes. If the destination already has notes, or the source is longer than the destination, a confirmation lets you Replace, Extend, Trim, or Cancel. Tapping the source pad asks you to pick a different pattern; Return cancels silently.
Copy Element	Same as Copy Pattern, but copies only the focused element's notes into the same element on the destination pattern. Refused with a message when the element has nothing to copy.
Bounce Loop	Converts the loop recorded on this layout by the Looper into sequencer notes in the pattern you are on, replacing what that pattern held, and clears the loop. If that pattern already has notes, a confirmation names it and lists the patterns that are still empty, with **Bounce** and **Cancel**; cancelling changes nothing. Confirms with a message naming the pattern. Refused with a message when there is no loop to bounce (`No loop to bounce`) or when the loop is too big for a pattern (`Loop too big to bounce`). Works whether or not the clock is running. This row is the only way to bounce a loop.
Reset Pattern Settings	Restores the active pattern's length, quantize grid, and play style to their defaults. Notes are kept. If the pattern holds notes past the default length, a confirmation explains that those notes stay stored but stop playing.
Follow Playhead	Auto scrolls the grid to keep the playhead on screen. Off by default.

Item	Description
Swap Sides	Swaps the Live and Launcher zones left to right. The grid stays centred. This is a device-wide setting, not part of the project.

Changing the layout while a copy is waiting for a destination cancels the copy with a message.

The bounce replays your recorded gestures through the real layout elements, so the element's configuration and MIDI effects are baked into the resulting notes, and enabled gesture lanes are preserved.

9.11 Auto-Save

Editing that changes pattern content (drawing or recording notes, clearing a pattern or element, copying into a pattern, bouncing a loop, or resetting pattern settings) is saved automatically with the rest of your project. A brief save indicator appears once the save completes, so you never save sequencer content by hand.

9.12 Clock

The sequencer runs on the project clock, the same clock as the Looper. The project clock source is one of INT (internal), USB-dev, MIDI, or USB-host. On the internal clock the sequencer plays at the project tempo. When the project is slaved to an external clock it follows that clock and respects Start, Stop, and Continue, and will not run until a Start or Continue arrives.

10 CV Outputs

Erae 2 provides **24 CV output channels** on its rear panel. These outputs can carry pitch voltages, gate signals, or continuous control voltages, making Erae 2 a modular interface alongside its MIDI capabilities.

CV Clock

10.1 Hardware Overview

The 24 outputs are on 12 TRS mini-jacks, each carrying two outputs: jack n carries output n on the tip and output n + 12 on the ring. Use a jack directly for one output, or a TRS splitter cable to reach both.

- **Output range:** -5 V to +8 V
- **Pitch standard:** 1 V/octave
- **Gate level:** 0 V off, 5 V on

Jacks 11 and 12 also serve as the inputs for **Pedal A** and **Pedal B**, so outputs 11, 12, 23 and 24 are shared with the pedals.

> **Warning:** The CV outputs are disabled in Low Power Mode. Power Erae 2 from the included power supply to use them (see Getting Started).

10.2 Output Modes

Each CV output carries one kind of signal, set in the element configuration:

Pitch (1 V/oct): the output follows the 1 V/octave standard, with MIDI note 0 at `-3 V`. Middle C (MIDI note 60, `C3`) outputs `2 V`, and each semitone is `1/12 V` (about `83.3 mV`). Pitch bend from the key is included in the voltage. The full MIDI note range fits within the output range.

Gate: the output is `0 V` when no note is held and `5 V` while a note is held. The **Gate Mode** can be `Gate` (high for as long as the note is held), `Trig` (a short pulse at each note start, `25` to `250 ms`) or `Retrig` (like Gate, with a short drop at each new note so envelopes restart). An optional **Gate Delay** (`0` to `5 ms`) delays the gate so the pitch settles first.

Continuous control: the output carries a voltage proportional to a value such as fader position, finger pressure, or slide position. By default the value maps to `0 V` to `5 V`; you can set the minimum and maximum voltages within the output range.

> **Tip:** A Gate output followed by a Pitch output gives you a classic Gate plus V/oct pair to drive a monophonic analog synthesizer.

10.3 Per-Element CV Assignment

CV outputs are assigned in the element configuration in Erae Lab, in the **Control Voltage** section:

- **Enable:** turns CV on for the element.
- **Output:** the first CV output used by the element.
- **Num voice:** on keyboards, the number of simultaneous CV voices (polyphony), 1 to 15.

Each voice uses one output per enabled signal, in this order: **Gate**, **1V / Oct**, **CV Velocity**, **CV Pressure**, **CV X abs**, **CV Y abs**, **CV X rel**, **CV Y rel**. Voices follow each other: the second voice starts on the output after the last output of the first voice.

For example, with Gate and 1V / Oct enabled, **Output** 1 and **Num voice** 4, outputs 1 and 2 carry the gate and pitch of the first voice, outputs 3 and 4 the second voice, and so on up to output 8. Notes are assigned to free voices as you play.

> **Tip:** For a four-voice patch, enable Gate and 1V / Oct, set **Num voice** to 4, connect the four pitch outputs to four VCOs and the four gate outputs to four envelope generators.

10.4 Additional CV Outputs per Voice

Beyond the core Gate and Pitch pair, each keyboard voice can output expression as CV. Each enabled stream uses one more output per voice:

Output	Description
CV Velocity	Strike velocity of the note (default `0 V` to `5 V`)
CV Pressure	Continuous finger pressure (default `0 V` to `5 V`)
CV X abs	Horizontal finger position within the key (default `0 V` to `5 V`)
CV Y abs	Vertical finger position within the key (default `0 V` to `5 V`)
CV X rel	Horizontal movement from where the finger landed, starting from an initial value (default `2.5 V`)
CV Y rel	Vertical movement from where the finger landed, starting from an initial value (default `2.5 V`)

With all eight signals enabled, one voice uses 8 outputs, so the 24 outputs hold at most three such voices. Plan the allocation across all elements of a layout. Erae Lab warns you when outputs overlap, but does not prevent it.

Other elements offer these CV outputs:

- **Fader 1D: Pressure** and **Y abs**.
- **Fader 2D: Pressure**, **X abs** and **Y abs**.
- **Button (Note): Gate** and **1V / Oct**.
- **Button (Control Voltage):** one output with an on and an off voltage.
- **Pedals:** gate, velocity or continuous outputs depending on the pedal type.

> **Tip:** Use **CV Pressure** to drive a VCA for note-level dynamics entirely in the modular domain, and **CV X rel** to control vibrato depth.

10.5 Eurorack Quickstart

10.5.1 First Voice Setup (Pitch + Gate)

1. In Erae Lab, select a Keygrid element and enable **Control Voltage**, with **Gate** and **1V / Oct** on and **Num voice** set to 1.
2. Set **Output** to 1. Output 1 carries the gate and output 2 the pitch.
3. Connect output 1 to your envelope generator's Gate input.
4. Connect output 2 to your VCO's V/oct input.
5. Play a note: the VCO tracks pitch at 1 V/octave, and the envelope fires on each note.

Outputs 1 and 2 are on jacks 1 and 2 (tip). With a splitter cable, jack 1 also carries output 13 on its ring.

10.5.2 USB Host MIDI to Eurorack

Connect a USB MIDI to CV module (e.g., Expert Sleepers FH-2 or Intellijel uMIDI) to the Erae 2 **Synth** (USB Host) port and route elements to **USB Host**. Configure the module to receive MIDI from Erae 2 and convert it to CV. This path supplements Erae 2's native CV outputs for higher polyphony.

10.5.3 CV Clock at 24 ppqn

1. Open **Settings > CV Clock**.
2. Turn **CV Clock** on.
3. Set **Beat Division** to `24 ppqn`.
4. Set **Clock Output** to an unused output (e.g., output 3).
5. Connect output 3 to your clocked module or clock divider's clock input.
6. Connect the next output (here 4), the **Reset Output**, to the module's reset input.

10.5.4 Suggested Eurorack Module Configurations

Use Case	Suggested Module
MIDI to CV conversion (multi-voice)	Expert Sleepers FH-2
MIDI to CV (single voice, simple)	Intellijel uMIDI
High-polyphony V/oct + Gate	Mutable Instruments Yarns
Direct Erae 2 CV with no MIDI bridge	Native CV outputs (outputs 1 to 24)

10.6 Calibration

The CV outputs are factory-calibrated at Embodme. If you observe pitch tracking errors, you can run the calibration again:

1. Unplug all CV jacks and pedals.
2. Power Erae 2 from the included power supply.
3. In **Settings**, select **Run CV calib**.
4. The calibration takes several minutes, and Erae 2 reboots automatically when it is done.

After calibration you can fine-tune the pitch scaling with **CV Gain (cent/oct)** in **Settings**.

> **Tip:** Run calibration with the device at normal operating temperature, after some minutes of use, for the most accurate results.

10.7 CV Clock Output

The CV Clock Output sends a clock pulse on one of the CV outputs, synchronized to the project tempo. This is a project-level setting: it applies across all layouts rather than to a specific element.

10.7.1 Accessing CV Clock Settings

Press the **Settings** button on the front panel, then select **CV Clock**. The **CV CLOCK** screen opens.

10.7.2 CV Clock Parameters

CV Clock: on/off switch for the clock output. Off by default. When on, the clock and reset outputs are reserved for the clock.

Beat Division: the pulse rate relative to the project tempo. Available values:

Value	Description
1 ppqn	One pulse per quarter note (1 clock per beat)
2 ppqn	Two pulses per quarter note
4 ppqn	Four pulses per quarter note (one per sixteenth note), the default
8 ppqn	Eight pulses per quarter note
24 ppqn	Standard MIDI clock rate (24 per quarter note)
48 ppqn	High-resolution clock, 48 per quarter note

24 ppqn is the standard MIDI clock rate. Use 4 ppqn for sixteenth-note triggers or 1 ppqn for a beat-rate trigger.

> **Tip:** Many Eurorack modules expect a 24 ppqn or 48 ppqn clock. Connect the Erae 2 CV Clock output directly to the module's clock input: no clock divider module needed.

Clock Output: the CV output that carries the clock pulse. Range: 1 to 23.

Reset Output: read-only, always the output right after the Clock Output. It sends a reset pulse to keep modules aligned with the pattern start. It cannot be disabled or assigned separately.

> **Tip:** Connect Clock to the module's clock input and Reset to its reset input: Erae 2 keeps both aligned to its internal pattern start.

> **Tip:** The CV clock keeps running while the project clock runs, whichever layout is active, so you can switch layouts mid-performance without interrupting the modular clock.

11 Chapter 11: LCD Interface

Erae 2 includes a 280 × 240 px colour LCD. You navigate every screen with the rotary encoder and its push switch, together with the panel switches described in Chapter 12. At power on, the LCD briefly shows the firmware version before the Home screen appears.

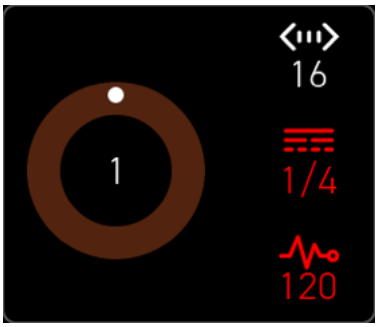
11.1 11.1 Encoder Navigation Model

Gesture	Action
Rotate	Move the focus highlight through the items on screen
Short press	Confirm a selection, open a sub-screen, or toggle a switch

On value rows (numbers and lists such as scale, rate or style), a short press enters edit mode: rotation then changes the value. A second short press confirms and leaves edit mode.

> **Tip:** Press **Return** to leave a screen. From a Settings sub-screen (Pedal, Routing, CV Clock, Velocity Curve, Save, Load) it goes back to Settings; from Settings, Mapping, Scale, Arpeggiator and Custom Scales it goes back to the Home screen.

11.2 11.2 Home Screen



Home Screen

The Home screen is the default LCD state. It shows the looper of the active layout:

- **Looper circle**: a circular arc showing the playback position within the loop. The number in the centre is the looper number, which is the number of the active layout (each layout has its own looper). The arc changes colour when that looper is muted.
- **Length**: the loop length in steps (sixteenth notes), from 1 to 256.
- **Quantize Grid**: the looper quantize grid (`Off`, `1/16`, `1/12`, `1/8`, and so on).
- **Tempo**: the project tempo in BPM, averaged to smooth out jitter from an external clock. Tempo can only be edited here when **Clock source** is `INT`.

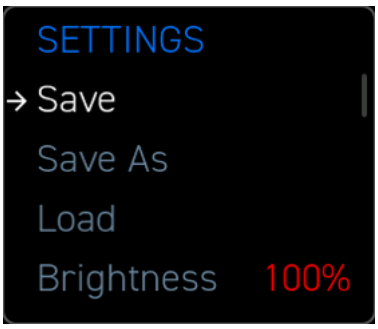
Rotate the encoder to highlight Length, Quantize Grid or Tempo, and press it to edit the highlighted value. While Length is being edited, **Plus** doubles it and **Minus** halves it. While Tempo is being edited, **Plus** doubles the tempo and **Minus** halves it.

Pressing **Return** on the Home screen removes the most recent looper layer.

Layouts are switched with the **N1** to **N8** panel switches. There is no LCD layout selector screen.

A small orange badge with an SD card icon fades in and out in the top left corner each time the device saves your project automatically.

11.3 11.3 Settings Menu {#113-settings-menu}



Settings Menu

Press the **Settings** switch to open the Settings menu, a vertically scrolling list. Press **Settings** again, or **Return**, to go back to the Home screen. Rotate to highlight a row, short press to activate it. The rows, in order:

Project: Save (shows the current project name), **Save As**, **Load**.

Global: Brightness (`5%` to `100%`), **Sensitivity** (list).

Project settings: Tempo (`1` to `999` BPM), **Clock source** (`INT`, `USB-dev`, `MIDI`, `USB-host`), **Metronome** (switch), **CC on layout change** (switch).

Sub-screens: Velocity Curve, **Pedal A**, **Pedal B**, **Custom Scales**, **Routing**, **CV Clock**.

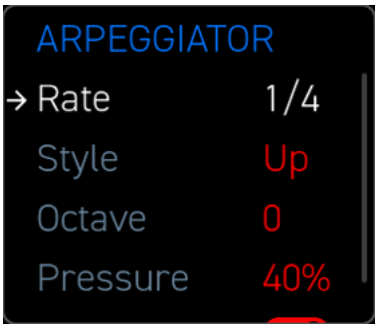
Backup and factory: Save Backup Project, **Load Backup Project**, **Load Factory Project**.

Calibration and system: encoder calibration status (read only), CV calibration status (read only), **CV Gain (cent/oct)** (`-50` to `+50`), **Run CV calib**, **MIDI 2.0: ON/OFF**, **Format SD Card & Reboot**.

Save, **Save As** and **Load** are greyed out when no SD card is mounted. In demo mode the title reads `DEMO MODE` and the list is reduced to **Reset Demo**, **Brightness**, **Sensitivity**, **Tempo**, **Clock source** and **Metronome**.

Every setting is described in Chapter 15.

11.4 11.4 Arpeggiator Settings Screen



Arpeggiator Settings

Select a keyboard element, then press **Arp**: the arpeggiator of that element is switched on or off and this screen opens. To open the screen without switching the arpeggiator on or off, hold **Settings** and press **Arp**.

Parameter	Type	Values
Rate	List	`1/32`, `1/16`, `1/8`, `1/4`, `1/2`, `1/1`, `Pressure`
Style	List	`Up`, `Down`, `UpDown`, `UpAndDown`, `Random`
Octave	Number	`0` to 8
Pressure	Number	`0%` to `100%`, in steps of 10 (finger pressure to velocity)
Quantize	Switch	Locks the arpeggiator to the project clock

Changes take effect immediately. See Chapter 8 for how each parameter behaves.

11.5 11.5 Scale / Key Selector {#115-scale–key-selector}

Scale Selector

The Scale screen edits the scale of one keyboard element. Hold **Scale**: everything except the focused element dims, and touching an element selects it instead of playing it. Release **Scale** to open the screen for the selected element. Press **Scale** again to close it.

The title row shows only **SCALE**. Below it the screen has:

- **Scale list** (full width): the factory scales, followed by your custom scales and a **NEW SCALE** entry when the keyboard can take a custom scale. Select it, then turn the encoder to browse.
- **ROOT** and **OCTAVE** values: root note C to B, octave -2 to 8. With a custom scale whose root is a frequency, **ROOT** shows that frequency (for example 432 Hz) and cannot be edited; the encoder skips it.
- **GLOBAL** and **OFFSCALE** switches (keygrids): **OFFSCALE** shows or hides the semitones outside the scale. When **GLOBAL** is on, the keyboard follows the project's global scale: changing the scale on any keyboard that has **GLOBAL** on changes it on all of them. Root note and octave always stay per element.

Turn the encoder to move the highlight in this order: scale, root, octave, **GLOBAL**, **OFFSCALE**. Click a value to edit it, turn to change it, and click again to confirm. Click a switch to flip it.

Element-specific behaviour: - **Iso Keyboard**: all controls are available. Custom scales appear in the list only when the keyboard has MPE enabled. - **Chroma Keyboard**: scale list, **ROOT** and **OCTAVE** only. The scale changes key colouring; the chromatic layout stays. There is no **OFFSCALE** or **GLOBAL** control. - **Drumpad**: **ROOT** and **OCTAVE** lists side by side; there is no scale list and no **GLOBAL** switch, because drumpads always play chromatic notes.

Selecting **NEW SCALE** and pressing the encoder opens the custom scale editor (see 11.16). When you press **Return** in the editor, the Scale screen comes back with the new scale selected on the keyboard.

While a custom scale is selected in the scale list, the **Settings** button LED pulses slowly: press **Settings** to edit that scale. Edits apply at once, so the keyboard plays the changed scale while you turn the encoder. Press **Return** in the editor to come back to the Scale screen, with the same keyboard and the scale still selected. On a factory scale, **Settings** opens the Settings screen as usual.

> **Tip:** Use the **ROOT** and **OCTAVE** values to transpose an Iso Keyboard quickly without editing the layout in Erae Lab.

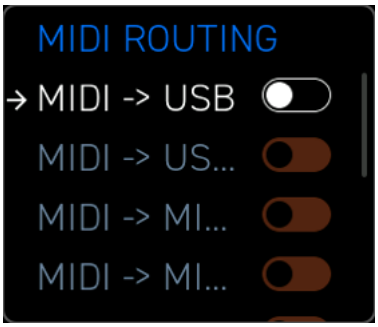
11.6 11.6 Info Screen (Status Overlay)

A message screen that appears automatically to report a condition. **Project Saved** disappears after two seconds; the other messages stay until you press any panel switch or the encoder. The messages are:

LCD message	Meaning
`Please insert microSD card and reboot`	No usable SD card was found at startup
`Project Saved`	A save to the backup slot completed
`Low Power (Low brightness, CV, MIDI B, Metronome disabled)`	The power supply is not strong enough for full operation
`SD Read/Write error`	The SD card itself failed
`Project corrupted (Loaded factory default)`	A project could not be read and the factory project was loaded instead
`Project read failed (File may be corrupted)`	A project file could not be read
`Project save failed (Please try again)`	A save did not complete
`Project file not found`	The requested project is missing from the SD card
`Legacy project Please connect ERAE Lab`	Projects from older firmware were found; Erae Lab converts them. The message closes by itself when Erae Lab starts sending the converted project

When the SD card has no readable filesystem, a confirmation screen asks `No filesystem found Format SD card?`. Choosing **Yes** formats the card and reboots Erae.

11.7 11.7 Routing Screen



MIDI Routing

Reached via **Settings -> Routing**. A list of on/off switches that forward incoming MIDI from one port to another:

MIDI -> USB Host, USB Device, MIDI A, MIDI B **USB Device ->** USB Host, MIDI A, MIDI B **USB Host ->** USB Device, MIDI A, MIDI B

Short press a switch to toggle it. Routing is saved with the project.

> **Tip:** Enabling **MIDI -> USB Device** lets Erae 2 act as a MIDI merger between a hardware synth and your DAW without additional software.

11.8 11.9 CV Clock Screen

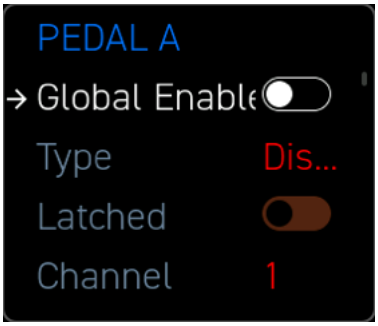
CV Clock

Reached via **Settings -> CV Clock**.

Parameter	Type	Description
CV Clock	Switch	Turns the CV clock output on or off
Beat Division	List	1 ppqn, 2 ppqn, 4 ppqn, 8 ppqn, 24 ppqn, 48 ppqn
Clock Output	Number	CV output carrying the clock pulses (1 to 23)
Reset Output	Read only	Always the output right after Clock Output (2 to 24)

> **Tip:** Set **Beat Division** to 24 ppqn to clock a Eurorack module that expects standard 24 ppqn clock.

11.9 11.10 Pedal Input Configuration {#1110-pedal-input-configuration}



Pedal Input Config

Reached via **Settings -> Pedal A** or **Pedal B**. The screen title is `PEDAL A` or `PEDAL B`.

Global Enable (switch) is always shown. When it is on, **Type** selects the pedal mode: `Disabled`, `Switch`, `Expressive`, `Sustain`, `Kick`. The other rows appear according to the type:

- **Switch**: **Out Routing**, **Latched**, **Channel**, **Message** (`Note`, `CC`, `PC`, `Tap Tempo`), then the note, CC on and off values, or program on and off values for that message, and **CV Enable** with its CV output and on and off voltages.
- **Expressive**: **Out Routing**, **Calibrate**, **Channel**, **CC Type** (`CC`, `RPN`, `NRPN`, `CC14bit`), **CC Number**, **Invert**, **CV Enable**.
- **Sustain**: **Mode** (`Binary` or `Continuous`). Binary shows **Out Routing**, **Latched**, **Channel**, **CC Type**, **CC Number**, **On Value**, **Off Value**, **CV Enable**. Continuous shows **Out Routing**, **Calibrate**, **Channel**, **CC Type**, **CC Number**, **Invert**, **CV Enable**.
- **Kick**: **Out Routing**, **Calibrate**, **Channel**, **Note**, **Velocity Sens**, **Note Duration** (`25ms`, `50ms`, `100ms`, `200ms`), **Impact Thresh**, **CV Gate** (gate output and on and off voltages) and **CV Velocity** (velocity output and minimum and maximum voltages).

Out Routing opens the Element Routing screen for the pedal (see 11.12). **Calibrate** opens the calibration wizard.

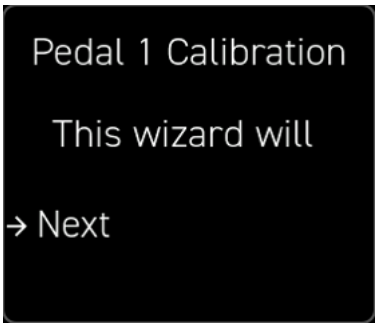
11.9.1 11.10.1 Format SD Card Confirmation Dialog

When you select **Format SD Card & Reboot** in the Settings menu, a confirmation screen asks `Format SD card? (Erae will reboot)`:

- **Yes**: formats the SD card and reboots the device.
- **No**: cancels and returns to the Settings menu.

The **MIDI 2.0** row uses the same kind of confirmation before it reboots Erae.

11.9.2 11.10.2 Pedal Input Calibration Wizard



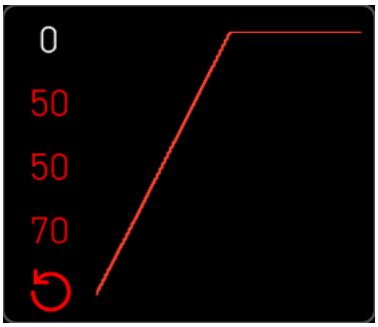
Pedal Input Calibration

A step by step wizard, reached with **Calibrate** on the pedal screen:

- **Expressive and Sustain Continuous:** Welcome, then **Capture Min** (pedal fully released, heel position), then **Capture Max** (pedal fully down, toe position). If the range is too small, the wizard reports a failure and offers **Retry**.
- **Kick:** Welcome, then **Capture** with the pedal at rest, then at least 3 light taps, then at least 3 strong kicks. If the strong kicks are not clearly harder than the light taps, the wizard reports a failure and offers **Retry**.

When calibration completes, the result is saved on the device.

11.10 11.11 Velocity Curve Editor



Velocity Curve

Reached via **Settings -> Velocity Curve**. A live curve graph shows how finger pressure maps to MIDI velocity. Four numeric buttons run down the left edge; the screen shows the values only, without labels, so remember their order from top to bottom:

Position	Parameter	Description
1 (top)	**Threshold**	How much contact is required before velocity starts to rise
2	**Drive**	Bends the curve: above 50 softer touches reach higher velocities, below 50 you must press harder
3	**Compand**	Expands or compresses the middle of the dynamic range (50 is neutral)
4	**Range**	Overall sensitivity: higher values need less pressure for full velocity. It is not a velocity ceiling

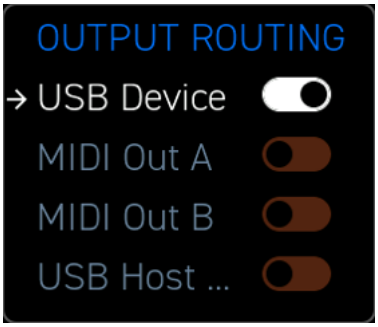
Turn the encoder to move the highlight across the four values and the reset arrow; click to edit, then each encoder click adds or subtracts 5 (values run 0 to 95). The **Reset** arrow restores the factory defaults (0, 50, 50, 70). The graph redraws live as parameters change; its horizontal axis is zoomed for readability, so do not read it as absolute pressure. **Return** goes back to Settings.

> **Tip:** High **Threshold** combined with a low **Range** makes the surface feel more like a traditional keyboard: intentional pressure required before notes register, and a firmer press for high velocities.

11.11 11.11b Sequencer Screens

While the Step Sequencer is active, the LCD shows the sequencer screen (lane header, full-pattern map, and division/length footer), and the **Settings** switch opens the Sequencer Settings menu instead of the global Settings. Both screens are described in the Step Sequencer chapter.

11.12 11.12 Element Routing Screen {#1112-element-routing-screen}

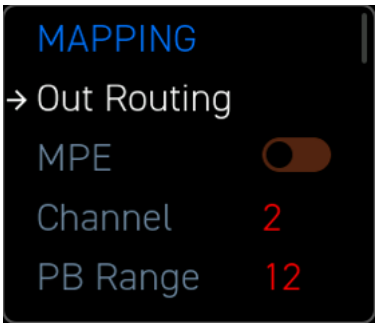


Element Routing

The `OUTPUT ROUTING` screen chooses which MIDI outputs receive data from the focused element or pedal. Reach it with **Out Routing** on the Mapping screen or on a pedal screen.

USB Device Out, **MIDI Out A**, **MIDI Out B**, **USB Host Out**: any combination can be on at the same time. Changes apply immediately.

11.13 11.13 Mapping Screen



Mapping

The Mapping screen edits the MIDI and CV parameters of one element. Hold **Mapping**: everything except the focused element dims, and touching an element selects it instead of playing it. Release **Mapping** to open the screen for the selected element. Press **Mapping** again to close it. Parameters depend on the element type:

- **Button**: button type (note, control change, program change, tap tempo, CV), channel, note, latch, CC and program values with optional bank, CV output settings.
- **Fader 1D / Fader 2D** (and their spring versions): for each axis an **X Abs** or **Y Abs** switch, then CC, default, centre, minimum and maximum (2D adds the X axis), optional pressure CC, CV outputs. The spring versions add a **SPRING PHYSICS** section (Drag, Max Speed, Stiffness,

Bounce). On a Spring Fader 2D it starts with a **Link X and Y** switch: while it is on, one set of physics drives both axes; turn it off to give each axis its own, then pick the axis to edit with the **Axis** row (X or Y). Turning the link back on copies the X values onto Y.

- **Key**: X and Y absolute and relative CCs, pitch bend range, pressure settings (**Pres Tune**) and pressure CC (**Pressure**), CV outputs.
- **Keyboard**: channel, MPE, pitch bend range, glissando, slide (CC74), pressure settings (**Pres Tune**), pressure CC (**Pressure**), X and Y CCs, line offset, and the CV section (pitch, gate and optional expression outputs).

Mapping a CC in your DAW. Put the DAW in learn mode and toggle the switch of the CC you want on (**Pressure**, **Motion Speed**, **X Abs**, **Y Abs**, **X Rel**, **Y Rel**): Erae 2 sends that CC at its maximum then its minimum value and the DAW learns it. If the switch is already on, toggle it off and on again. **Pres Tune** turns aftertouch on or off and sends no test CC.

Every element type also shows its transfer group. An **Out Routing** button at the bottom opens the Element Routing screen for that element.

11.14 11.15 Save / Save As / Load Screens

Save Project

Erae saves your work automatically: shortly after you change something, the current project is written to the SD card (the SD badge appears in the top left corner). The project entries in Settings cover the rest of the project lifecycle:

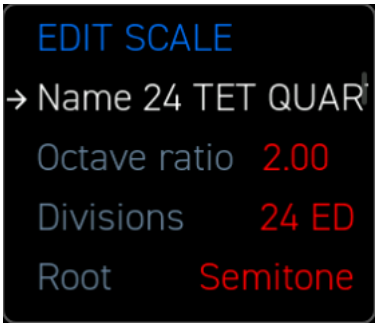
- **Save**: writes the current project to its SD card entry. If the project has no SD entry yet (a backup or factory project), it opens Save As instead.
- **Save As**: opens the `SAVE AS` list: **+ CREATE NEW** followed by the projects on the SD card, the current one marked with *. Pick an entry to name the project: a letter keyboard appears on the playing surface, prefilled with the name you picked. Type on the surface (the check mark key at the end of the last row also confirms), turn the encoder backward to delete a character or forward to add a space, press the encoder to confirm, or press **Return** to cancel. The screen shows the name and, under it, two reminders: `Type on the surface` and `Press encoder: save`.

- **Load**: opens the `LOAD PROJECT` list. Short press a project to load it; it replaces the current project.
- **Save Backup Project**: writes the current project to the internal backup slot. `Project Saved` confirms it.
- **Load Backup Project**: loads the backup slot. When an SD card is mounted, it is also saved to the SD card as `Backup`, `Backup_2`, and so on.
- **Load Factory Project**: loads the factory project as a temporary project with no SD entry, until you save it.

Backup and factory projects have no SD entry until you save them, so they are not auto-saved or synced with Erae Lab until you use **Save** or **Save As**.

> **Tip:** Before loading the factory project or the backup, check that the project you were playing has an SD entry (its name appears next to **Save**), so it is already safe on the card.

11.15 11.16 Custom Scales Screens {#1116-custom-scales-screens}



Custom Scale Editor

Settings -> Custom Scales lists your custom scales, followed by **+ Create scale** (while there is room), **Save scales** and **Load scales**. Select a scale to edit it, or **+ Create scale** to start a new one. The editor also opens from the Scale screen (see 11.5).

Return in the editor goes back to where you opened it: the Custom Scales list, or the Scale screen.

The editor (`NEW SCALE` or `EDIT SCALE`) lists these rows from the top:

- **Name**: opens the letter keyboard on the playing surface, as for a project name (see **Save As**): type on the surface, turn the encoder backward to delete a character or forward to add a space, press the encoder to save the name, or press **Return** to keep the old one. The screen shows the name and, under it, `Type on the surface` and `Press encoder: save`.
- **Octave ratio**: the interval at which the scale repeats (`2.00` is an octave).

- **Divisions**: `Free` for arbitrary ratios, or an equal division of the octave ratio (for example `12 ED`). While you turn the encoder, the degree list and the keyboards playing the scale follow at once; click to keep the value.
- **Root**: `Semitone` or `Frequency`, which decides the next row. This also takes effect when you click to leave the edit.
- **Root Offset** (semitone offset from the element's root note) or **Root Freq** (absolute frequency in Hz).
- A **DEGREES, CENTS ABOVE ROOT** section with one row per degree, each showing its pitch in cents above the root. With a **Frequency** root, the section is **DEGREES, FREQUENCY**: each row shows the degree's frequency in Hz (root frequency times the degree ratio, for example `486.0 Hz` for 9/8 above 432 Hz), with its cents above the root in smaller type below.
- **+ Add degree** (hidden on equal-division scales), **Load Template** (built-in tunings such as Just Intonation Major, Pythagorean or Bohlen-Pierce) and **Delete** (shown as `In use` and disabled while a keyboard uses the scale).

Click a degree to open its page: **Ratio**, **Semitone**, **Cents** (`0` to `99` within the semitone) and **Delete degree**. With a **Frequency** root, a **Freq** line on top shows the degree's frequency and its cents above the root while you edit; it is for reading only. On an equal-division scale, editing a degree turns the scale into a free scale. **Return** goes back to the editor with that degree's row highlighted.

Scales save automatically about 2 seconds after you stop editing, and at once when you leave the editor.

A custom scale can be assigned only to an Iso Keyboard with MPE enabled. Building and managing custom scales in Erae Lab, including Scala file import, is covered in the Custom Scales chapter of the Erae Lab manual.

12 Chapter 12: Switches & Encoder

Erae 2 gives you direct physical control through 18 panel switches with RGB LED indicators and a rotary encoder with a push switch, all arranged around the LCD. These controls let you switch layouts, drive the looper and sequencer, and navigate the LCD menus without touching the performance surface.

12.1 The Panel Switches

The panel carries 18 switches, each with its own LED: eight numbered switches and ten function switches.

12.1.1 Numbered Switches (N1 to N8)

Eight switches labelled **N1** to **N8** select the eight layouts of the current project. Press one to switch the whole playing surface to that layout. Each layout also has its own looper, and the numbered switches double as looper controls in combination with other switches (see Switch Combinations).

> **Tip:** Keep the layouts you switch between most often next to each other, so a single thumb movement changes your whole playing surface.

12.1.2 Function Switches

Switch	Function
Play/Rec	When the clock is stopped on internal clock, starts the clock. Otherwise arms or disarms looper recording (in sequencer mode, recording into the sequencer)
Stop	Stops the clock and playback
Plus	Shifts the selected element up one octave. On the Home screen, doubles the loop length or tempo you are editing

Switch	Function
Minus	Shifts the selected element down one octave. On the Home screen, halves the loop length or tempo you are editing
Scale	Hold to select an element by touching it, release to open the Scale screen for it
Sequencer	Enters or leaves the Step Sequencer; also the hold modifier for sequencer shortcuts (see Chapter 9)
Mapping	Hold to select an element by touching it, release to open the Mapping screen for it
Arp	Switches the arpeggiator of the selected keyboard on or off and opens the Arpeggiator screen
Settings	Opens or closes the Settings menu
Return	Leaves the current screen. On the Home screen, removes the most recent looper layer

Play/Rec and Stop also bring the LCD back to the Home screen (or the sequencer screen). On a LivePad element (Ableton Live clip control), **Plus** and **Minus** zoom and **Arp** cycles the LivePad mode.

12.2 The Encoder

Erae 2 has a single rotary encoder beside the LCD. It is the main control for navigating menus and entering values.

12.2.1 Scroll

Rotate the encoder to move through list items, change values, or step between items on screen. Each detent moves exactly one step; the encoder does not accelerate.

> **Tip:** For large tempo changes, edit **Tempo** on the Home screen: **Plus** doubles it and **Minus** halves it, then fine-tune with the encoder.

12.2.2 Click

Press the encoder to confirm the highlighted item: it opens a sub-screen, toggles a switch, or enters and leaves edit mode on a value.

12.3 Return Switch

Return leaves the current screen. From a Settings sub-screen (Pedal, Routing, CV Clock, Velocity Curve, Save, Load) it goes back to the Settings menu. From the Settings menu, the Mapping, Scale and Arpeggiator screens, it goes back to the Home screen, which shows the looper of the active layout and the current tempo.

12.4 Shift Combinations

Hold one switch and press another to reach these functions:

Combination	Action
Return + N1 to N8	Clear the loops of those layouts
Return (held) + N1 to N8	After you confirm, clear the loop and all eight sequencer patterns of those layouts
Play/Rec + N1 to N8	Unmute the looper of that layout
Stop + N1 to N8	Mute the looper of that layout
Settings + Arp	Open the Arpeggiator screen without switching the arpeggiator on or off

The **Sequencer** switch has its own set of combinations (pattern switching, tap tempo, zoom, clearing a track), described in Chapter 9.

13 Chapter 13: LED Feedback

The Erae 2 surface is covered by a 42 × 24 grid of full-colour RGB LEDs: 1,008 individually addressable pixels that bring your layouts to life. Every element type has its own visual language, and touch interactions produce immediate feedback so you always know the surface is responding.

13.1 The RGB Color System

Each LED shows 8-bit colour per channel (red, green, blue). The firmware applies a per-channel colour calibration so that the colours you pick in Erae Lab look natural and consistent on the surface, and caps the overall LED intensity to protect the surface from overheating during long sessions.

When the power supply cannot deliver enough power, Erae 2 lowers the LED intensity automatically and shows a Low Power message (see Troubleshooting).

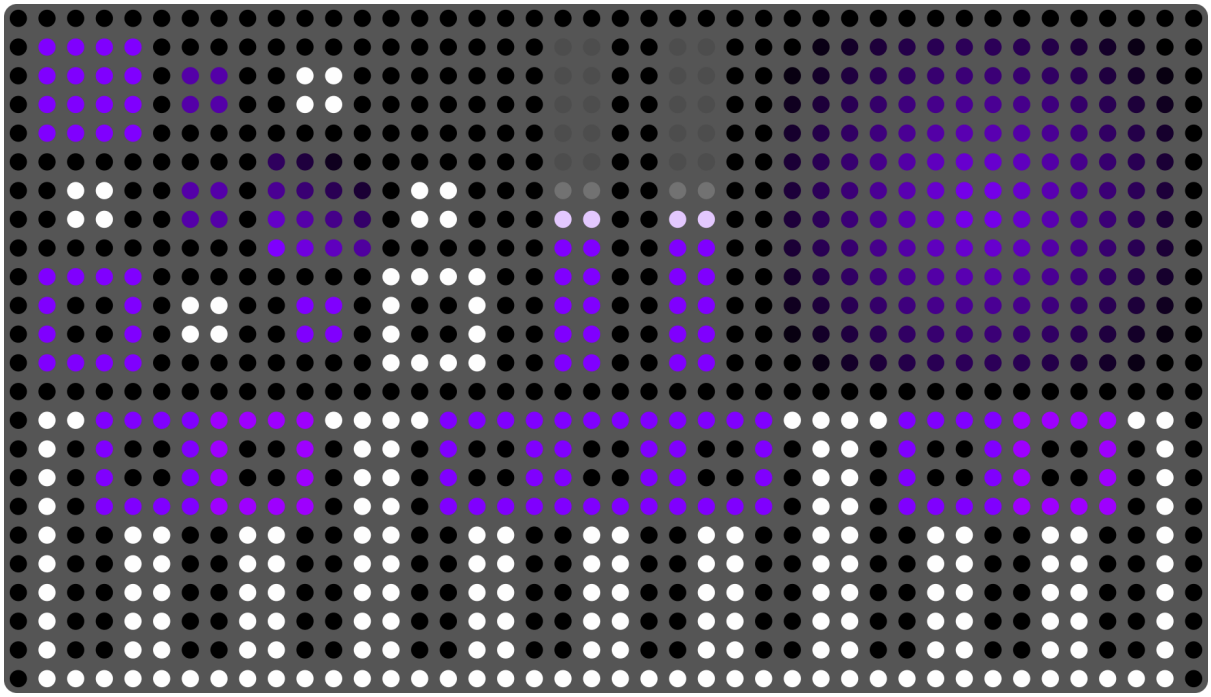
> **Tip:** Colours you set in Erae Lab are what you see on the hardware. The calibration is built into the firmware; you do not need to adjust anything.

13.1.1 Brightness Control

Global brightness is set with **Brightness** in the Settings menu. Lowering it is useful in dark studios and reduces power draw.

13.2 Per-Element Visualization

Each element type uses a distinct visual style. Styles are assigned in Erae Lab and stored in the project.



LED Touch Feedback

13.2.1 Keys

Key elements display a solid colour or gradient that covers their area. Scale-aware keyboards colour each key by its scale degree, so root notes and in-scale notes stand out from out-of-scale notes.

Style options for keys: - **Solid Color**: uniform fill across the entire key area - **Linear Gradient**: colour fades from one hue to another across the key - **Circular Gradient**: colour radiates outward from the centre of the key - **Rectangular Gradient**: colour fades from the centre to the edges - **Edge Color**: distinct border colour with a contrasting fill

13.2.2 Buttons

Buttons use the **Solid Color**, **Dual Intensity**, **Dual Color** or **Center Fill** style to show their on and off state. With Dual Intensity the off state is a dimmer version of the colour; with Dual Color the off state uses its own colour. Latched buttons hold their lit state between presses.

13.2.3 Faders (1D and 2D)

Fader elements display a bar or active area of light that tracks your finger position. The fill starts at the fader's **Center Value** and extends to the current value. Newly created faders use a center value of `0`, which matches the traditional from-minimum rendering; setting the center to 63 creates a center-detent visual response.

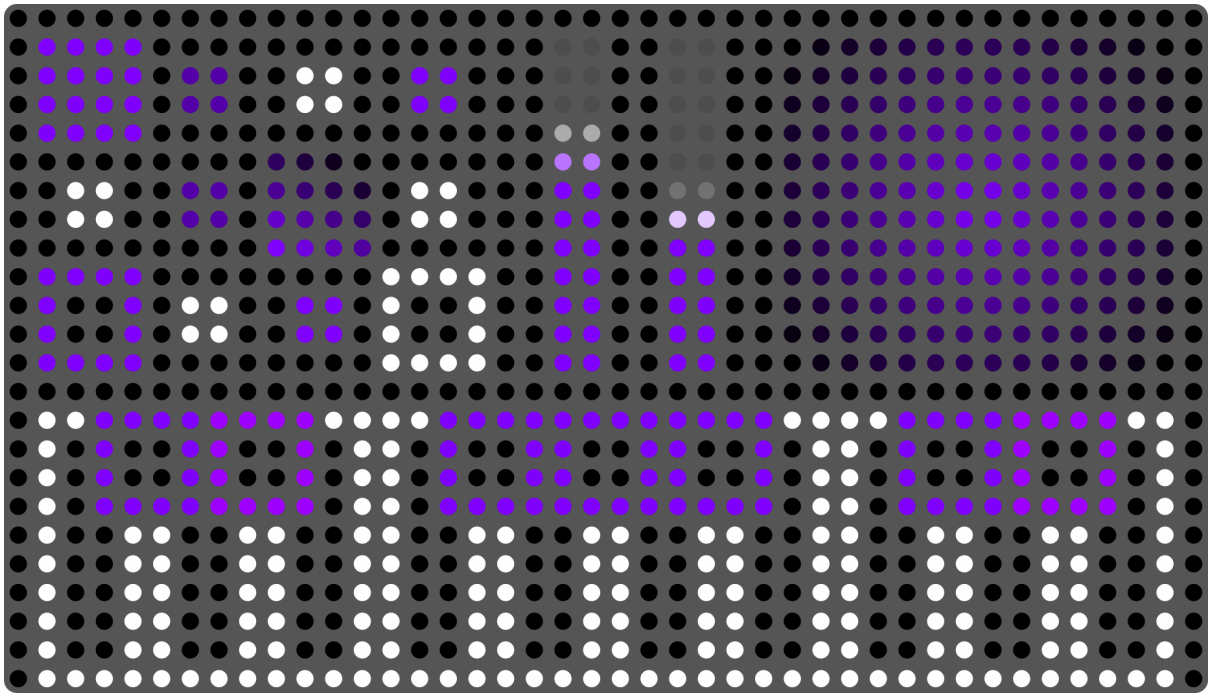
- **Fader 1D**: a vertical or horizontal bar fills from the center value toward your touch point. With center `0`, it fills from the bottom or minimum end. With center `63`, it fills away from the middle in either direction.
- **Fader 2D**: a crosshair or dot marks your X/Y position within the element area. The active area is drawn between the center point and the current position, and a dim center crosshair can remain visible as a reference for center-detent XY layouts.

13.2.4 API Zone

API zones have no built-in visual style: the colour of each pixel is driven entirely by your software through the Developer API, allowing fully custom visualizations.

13.3 Animations

Animations layer dynamic visual effects on top of the base element style. They are configured per element in Erae Lab and play in response to touch events.



LED Animation

Three animation trigger points are available:

Trigger	When it fires
Click	Immediately when the finger first contacts the surface
Slide	Continuously while the finger moves across the surface
Release	When the finger lifts off

Each animation has a **colour mode** (the element's own colour, or a colour you set), a **speed**, and a **colour**. ColorMorph also has an **intensity**.

13.3.1 FingerGlow

A bloom of light follows the moving finger across the element, leaving a brief luminous trail that fades naturally.

> **Tip:** A fast click FingerGlow with a contrasting colour gives playing keys a piano-like hammer effect, subtle but noticeable in live performance.

13.3.2 Ripple

A ring of light expands outward from the touch point, fading as it travels away from the contact. Ripples give percussion and rhythm elements a physical, drum-pad feel.

13.3.3 Halo

A glow appears at the contact point and stays there for the duration of the touch; it does not trail the finger. The Halo is useful for sustaining keys, where you want a persistent glow under each held note rather than a moving effect.

13.3.4 ColorMorph

The element transitions toward the configured colour. **Intensity** controls how much the colour changes; **Speed** controls the transition time. Assign ColorMorph to the Click trigger, the Release trigger, or both.

> **Tip:** Use ColorMorph on a fader element with a warm colour on Click and a cool colour on Release, making the active fader visually distinct at a glance.

13.4 Press Feedback

When you touch a keyboard key, the LED responds immediately in two ways:

1. **Intensity shift**: the key brightens or dims according to its intensity change setting in Erae Lab. A positive value makes the key brighter under your finger; a negative value dims it.
2. **Animation trigger**: if the element has a click animation, it fires at the same moment.

> **Tip:** Press feedback comes from touch detection, not from MIDI output, so you see it even when the element's MIDI outputs are all turned off.

13.5 Status Indicators

The LCD and switch LEDs carry additional status information that does not appear on the main surface grid.

13.5.1 Auto-Save Indicator

Each time Erae saves your project automatically to the SD card, a small orange badge with an SD card icon fades in and out in the top left corner of the LCD.

13.5.2 Project Saved Confirmation

When you save to the backup slot with **Save Backup Project**, the LCD shows **Project Saved** for two seconds.

13.5.3 Clock Sync Pulse

When an external MIDI clock source is selected (`USB-dev`, `MIDI`, or `USB-host`), the Home screen tempo shows the incoming tempo, averaged to smooth out jitter.

> **Tip:** If the tempo display fluctuates rapidly, the incoming MIDI clock may be jittery. Switch **Clock source** to `INT` to stabilize tempo while troubleshooting.

13.5.4 Switch LEDs

All 18 panel switches have an RGB LED:

- **N1 to N8**: the active layout is lit white at full brightness. The other layouts glow softly and pulse brighter when they play notes. A layout whose looper is muted is lit red.

- **Play/Rec**: green while the clock runs, red while recording, pulsing red when recording is armed with the clock stopped.
- **Plus / Minus**: show the octave of the selected element. At the centre octave both are lit; above it **Plus** takes a colour that changes with each octave, below it **Minus** does.
- **Scale, Mapping, Sequencer, Settings**: brighter while their screen is shown.
- **Arp**: lit in colour while the arpeggiator of the selected keyboard is on.

> **Tip:** The switch LEDs follow the global brightness: lowering **Brightness** in Settings dims both the surface and the switch panel.

14 Chapter 14: Connecting to Erae Lab

Erae Lab is the companion desktop application for designing, editing, and managing your Erae 2 projects. Connecting the hardware to Erae Lab unlocks layout editing, firmware updates, and background project sync.

14.1 USB Connection

Connect Erae 2 to your computer with a USB cable. Erae Lab does not use the MIDI ports, so you do not need to configure any MIDI port to use it, and your DAW can keep using them at the same time.

No driver installation is required on macOS or Windows. On Linux, Erae Lab shows a **Fix Now** notification the first time it needs USB permission; click it and enter your admin password.

14.1.1 Auto-Detection

Erae Lab looks for connected devices every few seconds, so you can plug Erae 2 in before or after launching Erae Lab. When the device is found, Erae Lab reads its firmware version and status, then compares the projects stored on the device with the Erae Lab library. This takes a moment on a healthy connection.

> **Tip:** If Erae Lab cannot reach the device, it shows a message with a **Rescan** button. Check the cable, power-cycle Erae 2 if the message says the device is not responding, then press **Rescan**.

14.2 Connection Indicators

- **Erae Lab**: the Erae logo at the right of the top bar lights up in colour when a device is connected, and is greyed out when none is.
- **Erae Lab notifications**: progress messages appear during sync, firmware updates and file transfers.

- **Erae 2 LCD**: the orange SD badge in the top left corner fades in each time a project change is saved on the card.

> **Tip:** Wait for Erae Lab's progress messages to finish before disconnecting, especially during a firmware update.

14.3 Project Sync

Erae Lab and Erae 2 stay in sync in the background. Layout edits in Erae Lab are saved automatically and sent to the device shortly after you stop editing. Changes you make on the device (for example on the Scale or Mapping screens) are saved on the SD card and reported back to Erae Lab. You do not need a manual push or pull step after each edit.

Projects that have no SD card entry yet (the factory project, or a backup loaded without an SD card) are not synced until you save them with **Save** or **Save As** on the device.

> **Note:** Brightness, Sensitivity, Velocity Curve and CV Gain are device-wide settings stored on Erae 2 itself. They are not part of any project and are not synced.

14.3.1 Conflict Resolution

If a project was changed both in Erae Lab and on the device since the last sync, Erae Lab asks which version to keep:

- **Keep Device**: replaces the Erae Lab project with the version currently on Erae.
- **Keep Both**: renames the Erae version, imports it into Erae Lab as a separate project, then syncs both projects back to Erae.
- **Keep Lab**: sends the Erae Lab version to Erae and discards the conflicting changes on the device. Closing the message also chooses Keep Lab.

> **Tip:** To avoid conflicts, connect Erae 2 before you start editing in Erae Lab.

14.4 Firmware Updates via Erae Lab

Firmware updates are delivered through **Erae Lab**, which includes the matching firmware. It can update a running device or recover a device that is already in bootloader mode.

To update:

1. Connect Erae 2 to an external power supply, then to your computer, and open Erae Lab. Erae Lab refuses to start an update when the device reports insufficient power.
2. When newer firmware is available, Erae Lab shows **Firmware update ready**. Click **Start Update**.
3. The device restarts into its bootloader and Erae Lab transfers the firmware. Do not disconnect the USB cable.
4. The device restarts into the new firmware, then finishes installing for about a minute. Do not switch it off during that time.

If the device firmware is too old for your version of Erae Lab, Erae Lab tells you so and asks you to update before it can connect.

If Erae Lab finds a device waiting in its bootloader, it shows **Bootloader detected** with three choices: **Install latest firmware**, **Select file** (a firmware `.syx` file on your computer), or **Dismiss**.

> **Tip:** Firmware updates do not erase your projects. Projects are stored on the SD card, separately from the firmware.

14.5 What Requires Erae Lab vs. Standalone

Erae 2 is fully functional without Erae Lab connected. The following table summarises which features require Erae Lab:

Feature	Standalone	Requires Erae Lab
Playing layouts	Yes	No
Switching layouts (N1 to N8)	Yes	No
Adjusting tempo and clock	Yes	No
Looper, arpeggiator and step sequencer	Yes	No
Pedal input and CV clock configuration	Yes (LCD Settings)	No
Editing element scale, MIDI and CV mapping	Yes (LCD Scale and Mapping screens)	For advanced editing
Creating and editing custom scales	Yes (LCD Custom Scales)	For Scala file import
Moving, resizing, styling, or changing element types	No	Yes
Creating new layouts	No	Yes
Firmware updates	No	Yes

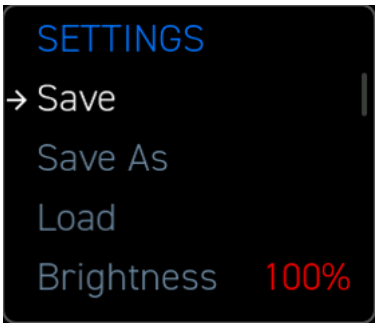
Feature	Standalone	Requires Erae Lab
Exporting and importing project files	No	Yes

> **Tip:** Changes you make on the device are saved to the SD card automatically, even when Erae Lab is not connected.

See the Device Sync and Firmware Updates chapters of the Erae Lab manual for details.

15 Chapter 15: Settings

The Settings menu is your on-device control centre for global preferences, project settings, saving and loading, MIDI routing, pedals, CV clock and calibration. Press the **Settings** switch to open it; press it again, or press **Return**, to go back to the Home screen. The full list of rows is given in Chapter 11.



Settings

15.1 Global Settings

Global settings are stored on Erae 2 itself. They apply to every project and are not synced with Erae Lab.

15.1.1 Brightness

Brightness controls the LED intensity of the surface and the switch panel together, from `5%` to `100%` (default `100%`). Lower values reduce power draw and are easier on the eyes in dark environments.

> **Tip:** A brightness around `60%` to `70%` is comfortable for most live performance environments. Reserve maximum brightness for installations or well-lit stages.

15.1.2 Sensitivity

Sensitivity sets how responsive the surface is to touch. Four preset options are available:

Option	Description
`XSensitive`	Most responsive: ideal for studio use or players with a very light touch
`Sensitive`	Responsive, with some protection against accidental triggers
`Safe`	Default: balanced for most playing styles and environments
`XSafe`	Least responsive: reduces false triggers in high-vibration environments

> **Tip:** If the surface feels unresponsive or requires excessive pressure, switch to `XSensitive`. If ghost notes appear without being touched, switch to `Safe` or `XSafe`.

15.1.3 Velocity Curve

Press **Velocity Curve** to open the velocity curve editor. It shapes how finger pressure becomes note velocity for the whole device. Four parameters shape the response, applied in this order:

- **Threshold**: how much contact is required before velocity starts to rise. Anything below the threshold produces no velocity; above it, the remaining pressure range is rescaled. Raise this to reduce accidental triggers on a light touch.
- **Compand**: expands or compresses the middle of the range, making dynamics easier or harder to control. 50 is neutral (linear).
- **Drive**: bends the curve. Values above 50 make the curve rise faster early (softer touches reach higher velocities), values below 50 make it rise later (you must press harder for the same velocity).
- **Range**: the overall sensitivity of the curve, not a velocity ceiling. Higher values mean less pressure is needed to reach full velocity. It does not limit the maximum velocity.

Each parameter runs from 0 to 95 and changes in steps of 5. The factory defaults are Threshold 0, Drive 50, Compand 50, Range 70.

A live curve preview renders on the LCD as you adjust each parameter. Changes apply to your playing immediately and are saved shortly after. Press the reset button to return all four values to factory defaults.

The velocity curve is a device-wide setting stored on the device itself, separate from the per-element velocity parameters (such as a key's Velocity Tune) described in Elements: the global curve shapes the pressure reading itself, and the element parameters then shape the note that is sent.

> **Tip:** If your playing feels too dynamic and notes are always at maximum velocity, lower **Range** and raise **Threshold** slightly. If all notes feel soft, raise **Range** or **Drive**.

15.2 Project Settings

These settings are saved with the current project.

15.2.1 Tempo

Tempo sets the internal tempo for the looper, arpeggiator, sequencer, metronome and CV clock. Range: 1 to 999 BPM. The change takes effect immediately.

15.2.2 Clock Source

Clock source selects the timing reference:

Value	Description
`INT`	Internal clock: Erae 2 generates its own tempo and sends MIDI clock, Start and Stop to USB Device, USB Host, MIDI A and MIDI B
`USB-dev`	Follows MIDI clock received on the USB Device port (your computer)
`MIDI`	Follows MIDI clock received on the TRS MIDI input
`USB-host`	Follows MIDI clock from a device connected to the USB Host port

With an external source, Erae 2 also follows incoming Start, Stop and Continue messages.

> **Tip:** When using a DAW as the clock master, set **Clock source** to `USB-dev`. Erae 2 then locks its looper, arpeggiator and sequencer to the DAW's transport.

15.2.3 Metronome

When **Metronome** is on, Erae 2 marks each beat with a short vibration pulse while the clock runs, with a longer pulse on the first beat of each bar. The metronome does not send any MIDI.

15.2.4 CC on Layout Change

When **CC on layout change** is on (the default), switching layouts sends Control Change messages on MIDI channel 16 to the USB Device port: CC `102` to `109` for layouts N1 to N8, value `127` for the layout you enter and `0` for the layout you leave. Faders also resend their current values when their layout is entered. This lets external software follow layout changes. Turn it off to keep layout switches silent.

15.3 Custom Scales

Press **Custom Scales** to list the custom scales stored on the device, create a new one, edit an existing one, or save and load the scale set with **Save Scales to Project** and **Load Scales from Project**. The editor is described in Chapter 11.

The project's global scale is not set here: turn on **GLOBAL** on the Scale screen of each keyboard that should follow it (see Chapter 11).

15.4 MIDI Routing

Press **Routing** to open the routing screen. Each switch forwards incoming MIDI from one port to another:

Route	Description
MIDI -> USB Host	Forward the TRS MIDI input to the USB Host port
MIDI -> USB Device	Forward the TRS MIDI input to your computer
MIDI -> MIDI A	Thru: TRS MIDI input to MIDI Out A
MIDI -> MIDI B	Thru: TRS MIDI input to MIDI Out B
USB Device -> USB Host	Forward MIDI from your computer to the USB Host port
USB Device -> MIDI A	Forward MIDI from your computer to MIDI Out A
USB Device -> MIDI B	Forward MIDI from your computer to MIDI Out B

Route	Description
USB Host -> USB Device	Forward MIDI from the USB Host device to your computer
USB Host -> MIDI A	Forward MIDI from the USB Host device to MIDI Out A
USB Host -> MIDI B	Forward MIDI from the USB Host device to MIDI Out B

> **Tip:** To use Erae 2 as a MIDI thru box, enable **MIDI -> MIDI A**, connect your upstream controller to the MIDI input and your synth to MIDI Out A.

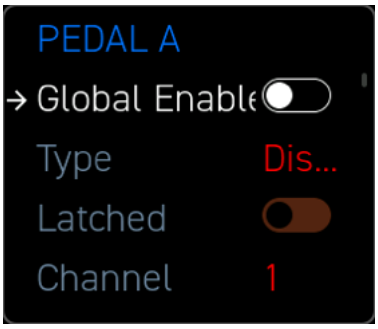
15.5 CV Clock Output

CV Clock sends clock pulses on a CV output, locked to the project tempo. Turn **CV Clock** on, choose the **Clock Output** (1 to 23) and set the **Beat Division** (default 4 ppqn):

Division	Description
1 ppqn	One pulse per quarter note
2 ppqn	Two pulses per quarter note
4 ppqn	Four pulses per quarter note (sixteenth notes)
8 ppqn	Eight pulses per quarter note
24 ppqn	Standard MIDI clock rate
48 ppqn	Double the MIDI clock rate

The **Reset Output** is read only: it is always the output right after the clock output, and it cannot be disabled or assigned separately. See CV Outputs for details.

15.6 Pedal Input Configuration



Pedal Input

Erae 2 has two pedal inputs, **Pedal A** and **Pedal B**. Press the corresponding row in Settings to open its configuration screen. The rows shown for each type are listed in Chapter 11.

15.6.1 Type

Type selects the pedal behaviour:

Type	Best for
`Disabled`	Input ignored
`Switch`	On/off switch pedal: sends a Note, CC, Program Change or Tap Tempo on press and release
`Expressive`	Continuous expression pedal, with an **Invert** option
`Sustain`	Sustain pedal. **Mode** `Binary` sends on and off values; `Continuous` follows the pedal position
`Kick`	Velocity-sensitive kick trigger

15.6.2 Global Enable

Toggle **Global Enable** to activate or bypass the pedal input without losing its configuration.

15.6.3 Routing

Out Routing opens the output routing screen, where you choose which MIDI outputs receive the pedal messages.

15.6.4 Latched

When **Latched** is on (Switch and Sustain Binary types), a single press toggles the output state instead of requiring the pedal to be held.

15.6.5 Channel

Channel sets the MIDI channel (1 to 16) for the pedal messages.

> **Tip:** Set Pedal A to sustain and Pedal B to expression for a keyboard-like playing setup.

15.7 Calibration Procedures

15.7.1 CV Calibration

The status row shows `CV calibrated` or `CV default calibration`. Press **Run CV calib** to run the automatic CV calibration: Erae 2 measures each of its 24 CV outputs against internal reference voltages and stores the result. **CV Gain (cent/oct)** (`-50` to `+50`) fine-tunes the pitch scaling of the outputs if octaves track slightly wide or narrow on your oscillator.

15.7.2 Encoder Calibration

The status row shows `Encoder calibrated` or `Encoder default calibration`. It is information only; there is no user action.

15.7.3 Pedal Input Calibration

For Expressive, Sustain Continuous and Kick pedals, press **Calibrate** on the pedal screen to launch the calibration wizard.

For Expressive and Sustain Continuous pedals:

1. **Welcome**: press **Next**.
2. **Capture Min**: release the pedal completely (heel position), then press **Capture Min**.
3. **Capture Max**: press the pedal fully down (toe position), then press **Capture Max**.

For **Kick** pedals:

1. **Capture**: leave the pedal at rest, then press **Capture**.
2. **Light taps**: tap the pedal lightly at least three times, then press **Continue**.
3. **Strong kicks**: kick the pedal hard at least three times, then press **Continue**.

The wizard then saves the calibration. If the pedal range is too small, or the strong kicks are not clearly harder than the light taps, it reports a failure and offers **Retry**.

> **Tip:** Calibrate each new pedal before your first performance. Pedals vary in their travel range, and an uncalibrated pedal may not reach its full range.

15.8 Project Management

- **Save**: writes the current project to its SD card entry. For a backup or factory project, opens Save As.
- **Save As**: saves under a new name or over a selected project.
- **Load**: loads a project from the SD card.
- **Save Backup Project**: writes the current project to the internal backup slot.
- **Load Backup Project**: loads the backup slot. With an SD card mounted, it is also saved to the card as `Backup`, `Backup_2`, and so on.
- **Load Factory Project**: loads the factory project as a temporary project until you save it.
- **Format SD Card & Reboot**: erases and reformats the SD card, then reboots. Use it only to recover from SD errors.
- **Reset Demo**: restores the demo content (demo mode only).

Erae also saves your changes to the SD card automatically, shortly after each change, as long as the project has an SD card entry. Factory and backup projects have no SD card entry until **Save** or **Save As** gives them one, so they are neither auto-saved nor synced with Erae Lab. Renaming and deleting projects is done in Erae Lab.

15.8.1 MIDI 2.0

The **MIDI 2.0: ON/OFF** row switches the USB MIDI mode. It is off by default (MIDI 1.0). Selecting it asks for confirmation, then Erae 2 reboots in the new mode.

15.9 Storage States and Recovery

Erae 2 saved projects normally live on the SD card. The internal flash holds the backup slot, calibration and device settings.

- **SD card mounted**: normal project save, load and Erae Lab sync are available.
- **No SD card**: the device can use the factory or backup project, but cannot save to or load from the SD project library.

- **No filesystem**: the device asks whether to format the card before it can be used.
- **Read or write failure**: saving or loading may fail; format or replace the SD card.

Project writes are protected against power loss: if power is interrupted during a write, the device recovers the last valid project data at the next start. Formatting the SD card erases the projects on it and reboots the device.

16 Chapter 16: Troubleshooting

This chapter covers the most common issues you may encounter with Erae 2 and how to resolve them.

16.1 Connection Issues

16.1.1 USB Not Recognized by Erae Lab

Symptoms: Erae Lab shows no connected device; the Erae logo in the top bar stays grey.

Steps:

1. Check that the USB cable supports data transfer (not a charge-only cable). Try a different cable.
2. Make sure the cable is in the Erae 2 USB port meant for the computer, not the power port or the port for USB MIDI devices, and connect directly to the computer rather than through a hub.
3. If Erae Lab shows a **Rescan** button, press it. If its message says the device is not responding, power-cycle Erae 2 first.
4. On Linux, Erae Lab needs USB permission. If a **Fix Now** notification appears, click it and enter your admin password, then reconnect the device.
5. Quit Erae Lab, unplug the device, wait five seconds, plug it back in, then relaunch Erae Lab.

16.1.2 MIDI Ports Not Appearing in DAW

Symptoms: Erae 2 is connected but its MIDI ports do not show up in your DAW's MIDI device list.

Steps:

1. Confirm the device is connected: its LCD is on and Erae Lab, if open, shows it as connected. The MIDI ports appear as soon as the device is connected; Erae Lab does not need to be running.
2. On macOS, check **Audio MIDI Setup** (Applications -> Utilities): Erae 2 should appear in the MIDI Studio window.

3. On Windows, check **Device Manager** for the MIDI device listing. Confirm no yellow warning icons are present.
4. Restart your DAW with Erae 2 already connected. Some DAWs only scan for MIDI ports at startup.
5. Check the **MIDI 2.0: ON/OFF** row in Settings. It is off by default; if you turned it on and your DAW does not support MIDI 2.0 devices, turn it off (Erae 2 reboots).

> **Tip:** In the default MIDI 1.0 mode, Erae 2 exposes `Erae 2 MIDI` (standard) and `Erae 2 MIDI (MPE)` (expressive play).

16.1.3 MPE Port Not Appearing on macOS (Stale Device Entry)

Symptoms: The `Erae 2 MIDI (MPE)` port does not appear in your DAW on macOS, or it appears with wrong port numbering, after a firmware update or reconnect.

Steps:

1. Open **Audio MIDI Setup** (Applications -> Utilities -> Audio MIDI Setup).
2. Click **Window -> Show MIDI Studio** if the MIDI Studio window is not visible.
3. Locate any stale Erae 2 device entry (it may appear greyed out or show an older name).
4. Right-click the stale entry and choose **Remove Device**.
5. Disconnect and reconnect the USB cable. The device re-enumerates and the expected MIDI ports appear.
6. Restart your DAW to pick up the refreshed port list.

16.2 MIDI Routing Problems

16.2.1 Notes Not Sounding on Expected Output

Steps:

1. Check the element's MIDI output destinations in Erae Lab, or with **Out Routing** on the device's Mapping screen (USB Device, MIDI A, MIDI B, USB Host).
2. Check the routing switches in Settings -> Routing for an unintended loop.
3. Confirm the MIDI channel matches your synth or DAW track.
4. For MPE instruments, make sure MPE is enabled on the keyboard element and on the receiving instrument, and point your DAW track at the `Erae 2 MIDI (MPE)` port.

16.2.2 Unexpected Notes or Double Triggers

Symptoms: Notes sound twice, or notes appear without touching the surface.

Steps:

1. Check **Routing** for enabled thru routes (for example **MIDI -> USB Device**). A thru route combined with a DAW echo setting can double notes.
2. If ghost triggers occur, check **Sensitivity** in Settings. Switch to `Safe` or `XSafe` to reduce false triggers.

16.3 Touch Feels Unresponsive

Symptoms: Notes require heavy pressure to trigger, or velocity values are consistently low.

Steps:

1. In Settings, set **Sensitivity** to `XSensitive` for maximum surface responsiveness.
2. Open **Velocity Curve** and raise **Range** (higher values need less pressure to reach full velocity); lower **Threshold** if notes require too much pressure to trigger at all.
3. Inspect the surface for debris or condensation. Clean it with a dry lint-free cloth.
4. Let the device run for a couple of minutes before playing: sensor readings settle slightly as it warms up.

16.4 Power Issues

16.4.1 Low Power Message

Symptoms: The LCD shows `Low Power (Low brightness, CV, MIDI B, Metronome disabled)`.

The power supply does not provide enough power for full operation, so Erae 2 lowers the LED brightness and turns off the CV outputs, MIDI Out B and the metronome. Connect Erae 2 to an external power supply that can deliver more power, then restart it. Erae Lab also refuses to update the firmware while the device reports low power.

16.5 LED Not Responding

16.5.1 Surface LEDs Are Dark or Partially Lit

Steps:

1. Check **Brightness** in Settings; it may be set to a low value (minimum is 5%).
2. Check for a Low Power message: with limited power the LEDs are dimmed.
3. Verify the active layout has elements with non-black colours assigned. An empty layout shows a dark surface.
4. If a region of the grid stays dark while the surrounding areas are lit, power-cycle the device (disconnect power, wait five seconds, reconnect). If it persists, contact support.

16.6 Factory, Backup, and Reset Procedures

Load Factory Project loads the factory project as a temporary project. Your projects on the SD card are not erased, and the factory project does not become part of the SD library until you save it.

1. Make sure the project you were playing is safe: if its name appears next to **Save** in Settings, it is already saved on the SD card.
2. In Settings, press **Load Factory Project**. It loads immediately, without a confirmation.

Save Backup Project writes the current project to the internal backup slot. **Load Backup Project** restores that backup; with an SD card mounted, it is also saved as `Backup`, `Backup_2`, and so on.

Reset Demo restores the demo content when demo mode is active. Use **Format SD Card & Reboot** only when you intentionally want to erase the SD card and recover from storage problems.

16.7 Bootloader Recovery

If the device does not start normally after an interrupted update, use bootloader recovery to reinstall the firmware:

1. Connect the device to an external power supply and to the computer, then open Erae Lab.
2. When Erae Lab shows **Bootloader detected**, choose **Install latest firmware**, or **Select file** to use a firmware `.syx` file saved on your computer.
3. Do not disconnect USB during the transfer.
4. The device restarts into the new firmware and finishes installing for about a minute. Do not switch it off during that time.

If the update keeps failing, Erae Lab asks you to unplug the device, plug it back in and try again. If the problem persists, contact support.

> **Tip:** Keep a copy of the latest firmware file on your computer, so you can use **Select file** if needed.

16.8 Orphan Bootloader Detection

If power is interrupted during an update, the device may restart into its bootloader and stay there. Erae Lab detects this automatically and shows **Bootloader detected** with the choices **Install latest firmware**, **Select file** and **Dismiss**. Every unit has a recovery bootloader, so the procedure above restores it.

> **Tip:** Use a good USB cable, keep the device on external power, and do not power it off during a firmware update or during the minute after it.

16.9 SD Card Errors

FAT32 is the recommended filesystem for the SD card. If errors persist after reinserting the card, use **Format SD Card & Reboot** in Settings to reformat it. If problems continue, try a different microSD card.

Saved projects normally live on the SD card. Without an SD card, the device can use the factory or backup project but cannot save to or load from the project library, and those projects are not synced with Erae Lab until they are saved to the card.

The LCD may display one of the following messages:

Message	Meaning	Action
`Please insert microSD card and reboot`	No usable SD card at startup	Insert a microSD card (FAT32, up to 32 GB) and restart
`No filesystem found Format SD card?`	The card has no readable filesystem	Choose **Yes** to format it (Erae reboots), or format it on a computer
`SD Read/Write error`	The SD card failed	Try **Format SD Card & Reboot**; if it persists, replace the card
`Project corrupted (Loaded factory default)`	A project could not be read	Load another project, or **Load Backup Project**

Message	Meaning	Action
`Project read failed (File may be corrupted)`	A project file could not be read	Load another project, or **Load Backup Project**
`Project save failed (Please try again)`	A save did not complete	Save again; if it persists, check or replace the card
`Project file not found`	The project is missing from the card	Load another project
`Legacy project Please connect ERAE Lab`	Projects from older firmware were found	Connect Erae Lab; it converts them

> **Tip:** If the SD card is repeatedly not detected after reinsertion, power-cycle the device.

16.10 Contacting Support

If you have followed the steps above and the problem persists, contact Embodme support:

Email: support@embodme.com

Please include your firmware version (shown on the LCD at startup), a description of the issue, and the steps you have already tried.

17 Chapter 17: Erae Compatibility

This chapter is a reference for users of the original Erae, the Erae Touch. If you own an Erae 2, most of this chapter does not apply to you: see the table at the end for a side-by-side comparison. The Erae Touch also has its own dedicated manual (the Erae Beta User Manual) with full chapters on its buttons, sequencer and velocity curve.

17.1 What Is the Erae Touch?

The Erae Touch is the first-generation hardware. It runs the same core surface and MIDI engine as Erae 2 but differs in several important ways: it has no LCD screen, no encoder, no CV outputs, no pedal inputs and no TRS MIDI input, and it is powered from its own adapter on a barrel jack rather than over USB. Navigation is entirely through five dedicated physical buttons and the LED surface. The Erae Touch firmware receives the same layout, scale, sequencer, and MIDI features as Erae 2, within the constraints of its hardware.

17.2 Projects and Layouts

The Erae Touch stores **four projects**, named A to D, each holding **eight layouts**. Pressing **Home** opens the layout selector on the LED surface: a row of project cells A to D above eight numbered layout cells. All edits are saved automatically to the active project about two seconds after your last change, while the **Alt** LED pulses orange; there is no manual save step on the device.

Erae Lab keeps the four projects in sync with its own copies and shows them pinned at the top of its project list.

> **Tip:** Use Erae Lab to back up your projects before making large layout changes.

17.2.1 Main/Alt Layout Pairs

The **Alt** button pairs the eight layouts two by two: layout 1 with 5, 2 with 6, 3 with 7, and 4 with 8. A click of **Alt** toggles the surface between the two layouts of the current pair, so you can, for

example, keep a chromatic keyboard on layout 1 and a drum pad on layout 5 and flip between them with a single press.

17.3 The 5-Button Navigation System

The Erae Touch has five dedicated physical buttons: **Home**, **Alt**, **Scale**, **Plus**, and **Minus**. There is no touchscreen, no encoder, and no soft buttons.

- **Home**: click to open or close the layout selector. Hold to enter sleep mode.
- **Alt**: click to toggle between the Main and Alt layout of the current pair, or to close any open overlay. Hold, pick a keyboard, and release to open the **Step Sequencer** on it (see below).
- **Scale**: click to open the scale selector for the focused keyboard. Hold to open CC mapping for the focused element.
- **Plus / Minus**: octave up and down on the focused keyboard.
- **Alt + Scale**: open the on-device **Velocity Curve** editor (see below).

17.4 Step Sequencer on the Erae Touch

The Erae Touch firmware includes the same sequencer engine as Erae 2, presented entirely on the LED surface: hold **Alt** on a layout with a keyboard element, then release to enter; click **Alt** to leave. The surface splits into a playable Live strip, a 32-column piano roll, and a launcher strip whose pads select patterns and control transport, undo/clear, and an on-surface settings page (play mode, clock source, quantize, tempo, and length).

Compared with Erae 2, the Erae Touch sequencer:

- sequences **keyboard and solo key elements only** (no faders or buttons)
- offers **four pattern slots** per layout instead of eight
- records **notes and velocities only**: per-note expression curves cannot be recorded or edited on the device, though patterns containing them play back correctly
- sets pattern length up to **64 steps** (Erae 2: 256)
- has **no looper bounce**

The Erae Beta User Manual covers it in full.

17.5 Velocity Curve on the Erae Touch

The Erae Touch has the same global velocity curve as Erae 2 (Threshold, Drive, Compand, Range, identical math), edited **on the device**: press **Alt + Scale** in the layout view to open the LED editor, tap the left or right half of a parameter cell to change its value, and watch the live curve plot on the right of the surface. The setting is stored in the device's global settings; Erae Lab does not modify it. The two devices' sensors are tuned differently, so the same numbers can feel different on an Erae Touch and an Erae 2.

17.6 Z-Only Pressure Sensing

The Erae Touch sensor reads **Z only**: the pressure at each cell, with no per-cell X or Y reading as on Erae 2.

Finger position (X and Y on the surface) is worked out from how the pressure spreads across neighboring cells. This approach gives accurate X and Y coordinates for finger tracking, but **XY resolution depends on the pressure pattern spreading across multiple cells**.

Practical implications:

- **Velocity** comes from how firmly your finger lands in the first moments of contact, giving a natural velocity response comparable to Erae 2.
- **Aftertouch** (channel pressure or poly pressure) works as expected and is fully functional.
- **Pitch bend and slide** work by tracking the computed X and Y position from the pressure distribution. Response is accurate but may feel slightly different from Erae 2 under very light touch, where the pressure spread across fewer cells reduces positional accuracy.

> **Tip:** For the most accurate slide and pitch bend response, use a confident, full-contact finger press. Very light or fingertip-only touches may reduce X and Y tracking precision.

17.7 No LCD

The Erae Touch has no LCD screen. Status information is conveyed through LEDs: the **Alt** LED pulses orange while changes are being saved, the **Plus** and **Minus** LEDs show the octave of the focused keyboard, and framed messages fill the surface for storage problems.

On-device editing screens (the layout selector, the sequencer and its settings page, the velocity curve editor, the scale selector, CC mapping) are drawn on the LED surface itself.

> **Tip:** If you need to confirm the firmware version or check device status, connect to Erae Lab.

17.8 No CV Outputs

The Erae Touch has no CV outputs. Erae 2's 24 CV channels, pitch and gate routing, and CV calibration system are entirely absent. If your setup requires CV and gate output, Erae 2 is required.

MIDI output from the Erae Touch is available on:

- **USB Device (Main)**: standard MIDI
- **USB Device (MPE)**: MPE MIDI
- **TRS MIDI out (port A)**: hardware DIN-style output

MIDI input is available on USB Device (Main) only. The Erae Touch does not have a TRS MIDI input; clock and control signals from external gear must arrive over USB.

17.9 No Looper, No Arpeggiator Screen

The Erae 2 looper is not exposed as a user-facing feature on the Erae Touch, and the arpeggiator settings screen has no equivalent on its LED-only interface. The sequencer (above) is the pattern tool on the Erae Touch.

17.10 No Encoder

The Erae Touch has no rotary encoder. Features that use the encoder on Erae 2, such as navigating LCD menus and fine-tuning values, are replaced on the Erae Touch by touch controls on the LED surface and the Plus and Minus buttons.

17.11 2-Stage Firmware Update

The Erae Touch uses a two-stage firmware update process. This is a key difference from Erae 2, which uses a single-stage update.

Update sequence:

1. Erae Lab restarts the device into its update mode.
2. Lab loads a small update program onto the device. The surface shows USB while the program waits for Lab.
3. Lab sends the new firmware to the update program, which shows a progress bar with a percentage while it receives and writes it.
4. The device restarts into the new firmware and reconnects to Lab.

The whole update can take several minutes. If something goes wrong, the surface shows an error code (E followed by two digits) in a blinking border.

> **Tip:** Do not disconnect the USB cable or switch the device off during a firmware update. If the update is interrupted, reconnect the device to Lab: Lab resumes the transfer when the device comes back in update mode, and offers **Install latest firmware** when it comes back in its recovery bootloader.

17.12 Feature Comparison: Erae Touch vs Erae 2

Feature	Erae Touch	Erae 2
Touch sensing	Z only (X and Y computed)	XYZ
Projects / Layouts	4 projects of 8 layouts (Alt pairs)	8 layouts per project
LCD	No	Yes
Step Sequencer	Yes: keyboards only, 4 patterns, notes and velocity, up to 64 steps	Yes: keyboards, faders, buttons; 8 patterns; expression lanes; up to 256 steps
Velocity Curve	Yes, edited on the LED surface (Alt + Scale)	Yes, edited on the LCD (Settings)
Looper	No	Yes
CV outputs	No	24 channels
Encoder	No	Yes
Buttons	5 dedicated	18 panel switches plus Calibrate
Pedal Inputs	No	2
TRS MIDI input	No	Yes
Power	Own adapter (barrel jack)	USB-C Power port

Feature	Erae Touch	Erae 2
Firmware update	2-stage	Single-stage
Project storage	Internal memory	SD card

18 Appendix A: Parameter Reference

This appendix lists the configurable parameters of each element type, organized by element category, with the names used in Erae Lab and on the device. Values are shown as you see them in Erae Lab: MIDI channels `1` to `16`, CC values `0` to `127`, and note names where MIDI note 60 is `C3`.

18.1 Common Parameters

These parameters appear on every MIDI-generating element.

Parameter	Description	Range	Default
Channel	MIDI channel for outgoing messages	1 to 16	1
Midi Output	Which physical outputs carry the element's messages	USB Device, USB Host, MIDI A, MIDI B (any combination)	USB Device only
Trig on entry	Which values the element sends when you switch to its layout	`Off` / `Default` / `All`	`Default`

18.2 Controller Slots

Every CC slot (Pressure, X abs, Y abs, X rel, Y rel, Motion, Sound Brightness) shares these settings.

Parameter	Description	Range	Default
Enable	Sends this controller	on / off	off

Parameter	Description	Range	Default
Type	Controller type	`CC` / `CC 14-bit` / `RPN` / `NRPN`	`CC`
Controller	Controller number (`CC 14-bit`: `0` to 31, LSB on controller plus 32)	`0` to 127	depends on slot
Bank / **Index**	Parameter address (RPN and NRPN)	`0` to 127 each	`0`
Min/Max	Output range	`0` to 127	`0` / 127
Intensity	Sensitivity scaling	`0` to 255	127
Smoothing	Low-pass smoothing	`0` to 255	`0`
Tracking	Multi-finger tracking	`None` / `LastPlayed` / `Highest` / `Lowest`	`None`
Filter	Response curve	`Exponential` / `Linear`	`Exponential`
Initial value	Starting value (relative slots and Sound Brightness)	`0` to 127	64

18.3 Key

Keys are the primary expressive building blocks. A Key element maps a touch zone to a single note with full per-finger expression. Keyboards use the same expression settings for all their keys.

18.3.1 Basic Parameters

Parameter	Description	Range	Default
Note	MIDI note	`C-2` to `G8`	`C3`
PB Range	Pitch bend range in semitones	1 to 96	12 (48 with MPE)

Note velocity follows the device-wide **Velocity Curve** in Settings.

18.3.2 Pressure Tune

Controls how finger pressure generates aftertouch (channel or polyphonic).

Parameter	Description	Range	Default
Enable	Sends pressure	on / off	on
Type	`Channel` (channel pressure) or `Poly` (polyphonic aftertouch)	enum	`Channel`
Tracking	Multi-finger tracking	`None` / `LastPlayed` / `Highest` / `Lowest`	`None`
Filter	Response curve	`Exponential` / `Linear`	`Exponential`
Min/Max	Output range	`0` to 127	`0` / 127
Intensity	Pressure sensitivity	`0` to 255	127
Smoothing	Low-pass smoothing	`0` to 255	`0`

18.3.3 Vibrato Tune (Pitch Bend)

Controls horizontal finger movement mapped to pitch bend.

Parameter	Description	Range	Default
Enable	Sends pitch bend	on / off	on
Style	Response curve	`Linear` / `Quadratic` / `FarQuadratic`	`Linear`
Strength	Pitch bend sensitivity	slider	
Return Speed	How quickly the bend returns to center	`None` to `Max`	

18.3.4 CC Assignments (Optional)

Each CC slot can be enabled independently (see Controller Slots).

Slot	Description	Default
Pressure	Pressure as a CC (in addition to aftertouch)	Disabled
X abs	Absolute horizontal position	Disabled
Y abs	Absolute vertical position	Disabled

Slot	Description	Default
X rel	Horizontal movement from where the finger landed	Disabled
Y rel	Vertical movement from where the finger landed	Disabled
Motion	Finger movement speed	Disabled

Motion speed is smoothed and normalized from `0` to `100 cm/s` before it is mapped to the controller range.

> **Tip:** When MPE is active, pitch bend and pressure travel on per-note channels automatically. The CC slots remain available for additional expressive axes.

18.3.5 Control Voltage

Parameter	Description	Range	Default
Enable	Sends CV	on / off	off
Output	First CV output used	`1` to `24`	
Gate / 1V / Oct	Gate and pitch outputs	on / off	on
Gate Mode	`Gate`, `Trig` or `Retrig`	enum	`Gate`
Trig Duration (ms)	Pulse length in Trig mode	`25` to `250`	`25`
Gate Delay (ms)	Delay before the gate opens	`0` to `5`	`0`
CV Velocity / CV Pressure / CV X abs / CV Y abs	Continuous outputs with a voltage range	`-5 V` to `+8 V`	off, `0 V` to `5 V`
CV X rel / CV Y rel	Relative outputs with a range and an initial value	`-5 V` to `+8 V`	off, initial `2.5 V`

See CV Outputs for the output order.

18.4 Button

Buttons send a discrete message on press and, depending on the type and the Latched option, a second message on release or on the next press.

18.4.1 Common Button Parameters

Parameter	Description	Range	Default
Channel	MIDI channel	1 to 16	1
Latched	Toggle mode: press sends the on message, next press sends the off message	on / off	off
Type	`Note`, `Control Change`, `Program Change`, `Control Voltage`, `Tap Tempo`	enum	`Note`

18.4.2 Button: Note Type

Parameter	Description	Range	Default
Note	MIDI note (sent at a fixed medium velocity)	`C-2` to `G8`	`C2`
CV	Optional gate and 1 V/oct outputs	output or disabled	Disabled

18.4.3 Button: Control Change Type

Parameter	Description	Range	Default
Enable On / controller	Controller sent on press	controller slot	CC 63, disabled
Value (on)	Value sent on press	`0` to 127	127
Enable Off / controller	Controller sent on release or on the next press	controller slot	CC 63, disabled

Parameter	Description	Range	Default
Value (off)	Value sent on release	`0` to `127`	`0`

18.4.4 Button: Program Change Type

Parameter	Description	Range	Default
Bank (on)	Send bank select with the on message	on / off	
Bank MSB / **Bank LSB** (on)	Bank Select MSB (CC 0) and LSB (CC 32)	`0` to `127`	
Program (on)	Program Change sent with the on message	on / off, `0` to `127`	
Bank / **Bank MSB** / **Bank LSB** (off)	Bank select for the off message	on / off, `0` to `127`	
Program (off)	Program Change sent with the off message	on / off, `0` to `127`	

18.4.5 Button: Control Voltage Type

Parameter	Description	Range	Default
Output	CV output	1 to 24	
On/Off	On and off voltages	`-5 V` to `+8 V`	`5 V` / `0 V`

18.4.6 Button: Tap Tempo Type

No parameters: each press taps the project tempo (internal clock only).

18.5 Fader 1D

A one-dimensional fader that tracks the Y position of a finger within its zone.

Parameter	Description	Range	Default
Channel	MIDI channel	1 to 16	1
Y abs CC	Controller for vertical position	controller slot	first unused CC on the channel
Default Y	Value sent on layout load before any touch	0 to 127	63
Center Y	Visual zero point for the fader fill	0 to 127	0
Pressure CC	Optional controller for pressure	controller slot	Disabled
Motion CC	Optional controller for finger movement speed	controller slot	Disabled
Pressure / **Y abs** CV	Optional CV outputs	output, voltage range	Disabled

> **Tip:** Set **Center Y** to 63 for a bipolar fader that fills away from the midpoint.

18.6 Fader 2D

A two-dimensional XY pad that tracks both axes simultaneously.

Parameter	Description	Range	Default
Channel	MIDI channel	1 to 16	1
X abs CC	Controller for horizontal position	controller slot	first unused CC on the channel
Y abs CC	Controller for vertical position	controller slot	first unused CC on the channel
Default X	Horizontal value on load	0 to 127	63
Default Y	Vertical value on load	0 to 127	63
Center X	Visual zero point for X rendering	0 to 127	0

Parameter	Description	Range	Default
Center Y	Visual zero point for Y rendering	`0` to `127`	`0`
Pressure CC	Optional controller for pressure	controller slot	Disabled
Motion CC	Optional controller for finger movement speed	controller slot	Disabled
Pressure / **X abs** / **Y abs** CV	Optional CV outputs	output, voltage range	Disabled

18.7 Keyboard (Keyboard, Keygrid, Drumpad)

Keyboard elements span multiple keys and share a set of layout-level attributes. Every key uses the expression settings described under Key above.

18.7.1 Keyboard Layout Parameters

Parameter	Description	Range	Default
Root note / **Octave**	Base note (Keyboard: Octave only)	`C-2` to `G8`	Keygrid and Keyboard `C2`, Drumpad `C1`
Scale	Scale applied to the keyboard	Keyboard: 15 factory scales; Keygrid: 14 factory scales, or custom scales with MPE on	Keyboard `Chromatic`, Keygrid `Major`
Use global scale	Follow the project-wide scale (Keygrid, Drumpad)	on / off	off
Show Offscale	Show every semitone (Keygrid)	on / off	off
Semitones	Row jump in semitones when Show Offscale is on (Keygrid, Drumpad)	1 to 48	Keygrid 5, Drumpad 4

Parameter	Description	Range	Default
Degrees	Row jump in scale degrees when Show Offscale is off (Keygrid)	1 to 48	3
Line offset	Row jump (Keyboard)	1 to 48	
Key Width / Key Height	Size of each key in grid cells	cells	4 / 4
MPE	Enable MIDI Polyphonic Expression	on / off	off
MPE Master	MPE zone: `Channel 1` (Lower Zone) or `Channel 16` (Upper Zone)	enum	`Channel 1`
Nb channel	Number of MPE member channels (replaces Channel when MPE is on)	1 to 15	
Num voice	Number of CV voices	1 to 15	1

18.7.2 Glissando Tune

Controls pitch behavior when a finger slides across keys.

Parameter	Description	Range	Default
Enable	Enables glissando	on / off	on
Vertical Gliss.	Vertical movement contributes to the glide	on / off	on
Retrigger	Retrigger the note when crossing a key boundary	on / off	off
Tune Loc	Reference point: `Pad` (key center) or `Finger` (initial touch position)	enum	`Pad`

Parameter	Description	Range	Default
Smoothing	Glide smoothing	`0` to `255`	`63`
In Tune Width	Size of the in-tune zone as a percentage of key width	`0` to `100`	`50`

18.7.3 Sound Brightness (CC 74)

Maps a gesture axis within a key to CC 74 (Brightness / Timbre).

Parameter	Description	Range	Default
Gesture	Which axis drives the CC: `None`, `Pressure`, `X Absolute`, `Y Absolute`, `X Relative`, `Y Relative`, `Motion Speed`	enum	`None`
Initial value	Value sent before any touch	`0` to `127`	`64`
Other settings	**Min/Max**, **Intensity**, **Smoothing**, **Tracking**, **Filter**	see Controller Slots	

18.7.4 Arpeggiator

The arpeggiator is set on the device (**Arp** button); see Arpeggiator & Looper.

Parameter	Description	Range	Default
On / off	Toggled with the **Arp** button	on / off	off
Quantize	Start the arpeggio on the clock grid	on / off	on
Octave	Number of extra octaves	`0` to `8`	`0`

Parameter	Description	Range	Default
Rate	Step rate, in fractions of a beat: `1/32`, `1/16`, `1/8`, `1/4`, `1/2`, `1/1`, `Pressure`	enum	`1/4`
Style	`Up`, `Down`, `UpDown`, `UpAndDown`, `Random`	enum	`Up`
Pressure	How much finger pressure shapes note velocity	`0%` to `100%`	`40%`

18.8 Ableton Launchpad (LivePad)

Parameter	Description	Range	Default
Position	Full Width, Left Half or Right Half	enum	set by placement
Zoom	Grid cell size	`Small` / `Large`	`Large`

18.9 API Zone

The API Zone exposes finger data to external applications via SysEx streaming.

Parameter	Description	Range	Default
Zone Index	Identifier for this zone in the data stream	`0` to `127`	`0`
Max Num Fingers	Saved, not applied yet	`0` to `10`	`10`
Finger Data Rate	Saved, not applied yet	`0` to `255`	`0`

18.10 Non-Visible Elements

Non-visible elements are not displayed on the surface but respond to external inputs.

18.10.1 Pedal

Each layout configures the two pedal inputs, **Pedal A** and **Pedal B**.

Parameter	Description	Range	Default
Type	`Disable`, `Switch`, `Kick`, `Sustain (Binary)`, `Sustain (Continuous)`, `Expressive`	enum	`Disable`
Channel	MIDI channel	1 to 16	1
Midi Output	Output ports	USB Device, USB Host, MIDI A, MIDI B	USB Device

18.10.1.1 Pedal: Switch

Parameter	Description	Range	Default
Latched	Toggle on each press (rather than press and release)	on / off	off
Message Type	`Note`, `Control Change`, `Program Change`, `Tap Tempo`	enum	`Note`
Note	Note for the Note type	`C-2` to `G8`	`C4`

Control Change and Program Change settings mirror those of the Button element. With **Tap Tempo**, each press taps the project tempo.

18.10.1.2 Pedal: Sustain (Binary)

Parameter	Description	Range	Default
Controller	Controller for sustain	controller slot	CC 64
On Value	Value when the pedal is pressed	`0` to `127`	`127`
Off Value	Value when the pedal is released	`0` to `127`	`0`

Parameter	Description	Range	Default
Latched	Toggle on each press	on / off	off

18.10.1.3 Pedal: Continuous (Expressive / Sustain Continuous)

Parameter	Description	Range	Default
Controller	Controller for continuous output	controller slot	Expressive CC 11, Sustain (Continuous) CC 64
Invert	Reverse the pedal direction	on / off	off

18.10.1.4 Pedal: Kick

Parameter	Description	Range	Default
Note	MIDI note triggered on impact	`C-2` to `G8`	`C0`
Velocity Sensitivity %	Velocity response scaling	`0` to `100`	`100`
Impact Threshold %	Minimum strike force to trigger	`1` to `100`	`10`
Duration Mode	`Until Release` or `Fixed`	enum	`Until Release`
Fixed Duration (ms)	Note duration in Fixed mode	`1` to `5000`	`100`

18.11 Project Settings

Project-level settings apply globally across all layouts.

Parameter	Description	Range	Def
Clock source	`INT`, `USB-dev`, `MIDI`, `USB-host`	enum	`INT`
Tempo	Internal tempo	1 to 999 BPM	`120`
Metronome	Enable click output	on / off	off
Pedal A / **Pedal B** Global Enable	Activate the pedal inputs	on / off	off
CC on layout change	Send element values when switching layouts	on / off	on

Parameter	Description	Range	Def
Switch Mode	Sequencer pattern switch mode: `Wait`, `Jump`, `Restart`	enum	`Wai`

MIDI 2.0 (alternate USB mode, applied after a reboot) is a device setting in the same menu, off by default.

18.11.1 CV Clock Output

Parameter	Description	Range	Default
CV Clock	Enable CV clock output	on / off	off
Beat Division	Clock rate: `1 ppqn`, `2 ppqn`, `4 ppqn`, `8 ppqn`, `24 ppqn`, `48 ppqn`	enum	`4 ppqn`
Clock Output	CV output that carries the clock	1 to 23	1
Reset Output	Read-only, the output after the clock output	Clock Output plus 1	2

18.11.2 MIDI Routing

Each routing switch forwards incoming MIDI from one port to another.

Parameter	Description	Default
MIDI -> USB Host	Route MIDI In to USB Host	off
MIDI -> USB Device	Route MIDI In to USB Device	off
MIDI -> MIDI A	Route MIDI In to MIDI Output A	off
MIDI -> MIDI B	Route MIDI In to MIDI Output B	off
USB Device -> USB Host	Route USB Device input to USB Host	off
USB Device -> MIDI A	Route USB Device input to MIDI Output A	off
USB Device -> MIDI B	Route USB Device input to MIDI Output B	off
USB Host -> USB Device	Route USB Host input to USB Device	off
USB Host -> MIDI A	Route USB Host input to MIDI Output A	off

Parameter	Description	Default
USB Host -> MIDI B	Route USB Host input to MIDI Output B	off

> **Tip:** MIDI routing is useful when using Erae 2 as a USB-to-MIDI interface. Enable **MIDI -> USB Device** to merge an external keyboard with the surface output seen by your DAW.

19 Appendix B: MIDI Implementation Chart

Erae 2 transmits MIDI over USB Device (class compliant), USB Host, MIDI Out A and MIDI Out B. Each element selects its output destinations independently. Incoming MIDI is received on USB Device, USB Host and the TRS MIDI input, and can be forwarded between ports with the project routing switches (**Settings -> Routing**).

In the table, **Recognized** means Erae 2 acts on the message itself. Incoming channel messages it does not act on can still be forwarded by the routing switches.

19.1 Main Implementation Table

Function	Transmitted	Recognized	Remarks
Basic Channel	Ch 1 to 16 per element		Each element has its own channel
Note On	Yes	No	Keys, Buttons (note type), keyboard elements, Switch and Kick pedals
Note Off	Yes	No	Sent on finger lift; Lift Tune sets the release velocity
Polyphonic Aftertouch	Yes (optional)	No	Key and keyboard elements when Pressure Type is `PolyPressure`
Channel Aftertouch	Yes (optional)	No	Key and keyboard elements when Pressure Type is `ChannelPressure`
Control Change	Yes	No	See the CC table below

Function	Transmitted	Recognized	Remarks
Program Change	Yes	No	Buttons (program change type) and Switch pedals, with optional Bank Select
Pitch Bend	Yes	No	Key and keyboard elements; range 1 to 96 semitones (default 12, 48 with MPE)
14-bit CC	Yes	No	Controller type `CC14bit`: MSB on the CC number, LSB on the CC number + 32
RPN	Yes	No	Controller type `RPN` on elements and pedals that offer a CC Type
NRPN	Yes	No	Controller type `NRPN` on elements and pedals that offer a CC Type
MPE	Yes	No	Enabled per keyboard; MPE output always uses the `Erae 2 MIDI (MPE)` port
System Exclusive	Yes	Yes	Universal Identity Request, and the Embodme protocol (see below)
MIDI Clock (0xF8)	Yes	Yes	Sent when **Clock source** is `INT`; followed when an external source is selected

Function	Transmitted	Recognized	Remarks
MIDI Start (0xFA)	Yes	Yes	Sent when the internal clock starts; followed with an external source
MIDI Stop (0xFC)	Yes	Yes	Sent when the internal clock stops; followed with an external source
MIDI Continue (0xFB)	No	Yes	With an external source, starts playback if stopped
Active Sensing (0xFE)	No	No	Not used

With the internal clock, Clock, Start and Stop are sent to USB Device, USB Host, MIDI A and MIDI B.

> **Tip:** Erae 2 does not send an MPE configuration or pitch bend range message. Set your instrument's MPE pitch bend range to match the keyboard's pitch bend range (48 semitones by default).

19.2 MPE Detail

When **MPE** is enabled on a keyboard element, Erae 2 sends as an MPE Lower Zone (master channel 1) or Upper Zone (master channel 16), chosen by the keyboard's **MPE Master Channel** setting.

MPE Parameter	Value
Master Channel	Ch 1: Lower Zone (default); Ch 16: Upper Zone
Member Channels (Lower Zone)	Ch 2 upward, allocated dynamically
Member Channels (Upper Zone)	Ch 15 downward, allocated dynamically
Per-Note Pitch Bend	Sent on the member channel
Per-Note Pressure	Sent on the member channel

MPE Parameter	Value
Per-Note Timbre	CC74 on the member channel (when the keyboard's slide output is enabled)
Pitch Bend Range (member)	Keyboard pitch bend range, 48 semitones by default
USB Port	`Erae 2 MIDI (MPE)`

19.3 Universal Identity Request

Erae 2 answers the standard Identity Request (`F0 7E <id> 06 01 F7`) with device ID `0x01` or `0x7F`, on the USB port it came from:

```
F0 7E 01 06 02 00 21 50 00 01 00 02 <channel> <major> <minor> <patch> F7
```

`00 21 50` is the Embodme manufacturer ID, `00 01` the Erae family and `00 02` the Erae 2 model. The last four data bytes give the firmware version.

19.4 SysEx Protocol

The Embodme SysEx messages an external program can use on Erae 2 are the **Developer API** commands, which read touches from API Zone elements and paint their LEDs. They start with this header:

```
F0 00 21 50 00 01 00 02 01 01 04 <sub-service> <payload...> F7
```

`00 21 50` is the Embodme manufacturer ID, `00 01 00 02` identifies Erae 2, `01` is the device ID, and `01 04` selects the API service. Messages addressed to another device ID, including the broadcast ID `0x7F`, are ignored.

19.4.1 API Zone Commands

Sub-service	Direction	Description
StartFingerDataStreaming (`0x01`)	Host -> Device	Start sending finger data from API zones
EndFingerDataStreaming (`0x02`)	Host -> Device	Stop finger data
ZoneBoundaryRequest (`0x10`)	Host -> Device	Ask for the size of an API zone
Clear (`0x20`)	Host -> Device	Clear the LEDs of an API zone
SetPixel (`0x21`)	Host -> Device	Set one LED
DrawRectangle (`0x22`)	Host -> Device	Fill a rectangle
DrawImage (`0x23`)	Host -> Device	Draw an image
VersionRequest (`0x7F`)	Host -> Device	Ask for the API protocol version

Byte layouts, coordinates, colour and image encoding, and worked examples are in Appendix D: Developer API. Erae Lab and the Ableton Live Remote Script use their own messages; you do not need to send those yourself.

19.5 Default CC Mappings

The following CC numbers are used by default when the relevant features are enabled. Element and pedal CC assignments can be changed in Erae Lab or on the device.

CC Number	Name	Used By	Notes
CC0 / CC32	Bank Select MSB / LSB	Buttons (program change) and Switch pedals	When a bank is enabled
CC11	Expression	Expressive pedal	Default CC number
CC64	Sustain	Sustain pedal	Binary: on and off values; Continuous: follows the pedal
CC74	Timbre	Keyboard slide output	Y position within a key; standard MPE timbre axis

CC Number	Name	Used By	Notes
CC102 to CC109	Layout change	**CC on layout change**	Channel 16, USB Device only: 127 when a layout is entered, `0` when it is left
User defined	Pressure	Key and fader pressure CC	Optional
User defined	X Position	Key and fader X absolute CC	Absolute horizontal position
User defined	Y Position	Key and fader Y absolute CC	Absolute vertical position
User defined	X Relative	Key and keyboard X relative CC	Movement since the touch started
User defined	Y Relative	Key and keyboard Y relative CC	Movement since the touch started
User defined	Motion Speed	Key and fader motion speed CC	Finger movement speed, smoothed and normalized from 0 to 100 cm/s

20 Appendix C: DAW Setup Guides

This appendix provides step-by-step connection guides for the most common DAW and hardware configurations. Whether you are routing standard MIDI, enabling per-note MPE expression, or connecting to hardware synthesizers and Eurorack modules, the relevant procedure is covered here.

20.1 General MIDI Setup

20.1.1 USB MIDI Ports

When Erae 2 is connected over USB in normal MIDI 1.0 mode, your DAW sees two user-facing MIDI cables inside a single class-compliant USB MIDI device. No driver installation is required on macOS, Windows 10/11, or Linux.

Cable	Port Name	Use
Cable 0	`Erae 2 MIDI`	Standard MIDI in both directions: notes, velocity, pitch bend, CC, clock. Use this for most DAW instrument tracks, for clock sent to Erae 2, and for general routing.

Cable	Port Name	Use
Cable 1	`Erae 2 MIDI (MPE)`	MPE output only: all per-note expression (pitch bend, pressure, slide) for MPE-compatible instruments. MPE messages always go to this cable; this routing is not configurable. Erae 2 does not receive MIDI on this cable.

Quick rule of thumb: route non-MPE instrument tracks to `Erae 2 MIDI` and MPE instrument tracks to `Erae 2 MIDI (MPE)`. MIDI 2.0 is an alternate USB mode controlled by **Settings -> MIDI 2.0: ON/OFF** and requires rebooting the device.

> **Pitfall:** Most DAWs collapse incoming MIDI to a single channel (typically channel 1) before delivering it to a VST or AU instrument unless the track is explicitly configured for MPE. When designing a layout intended for use with a plugin (such as Erae Sound), keep all elements that send CCs on the **same MIDI channel**, channel 1 is the safest default. Layouts that spread macro CCs across multiple channels will lose those CCs in non-MPE DAW tracks. This restriction does not apply to hardware routing, MPE keyboards on the MPE port, or DAW tracks with MPE explicitly enabled.

20.1.2 Using Erae Sound in a DAW

When Erae Sound is loaded as a VST3 or AU plugin on a DAW instrument track, the Erae surface sends its data over the standard USB MIDI ports, no proprietary driver is involved.

Fader gestures and keyboard notes (non-MPE) arrive on **`Erae 2 MIDI`**. Enable that port as the MIDI input on the instrument track and set the channel to match your layout (default: channel 1).

MPE slide and per-note expression arrive on **`Erae 2 MIDI (MPE)`**. For full MPE playback, enable MPE mode on the instrument track, point its input at `Erae 2 MIDI (MPE)`, and set the channel to **All Channels** so all 16 member channels reach the instrument.

> **Do not run the Erae Sound standalone and the plugin at the same time.** Both instances listen to the same Erae MIDI input. Notes and gestures will reach both engines simultaneously, causing double triggering and audio from both outputs. If you hear audio from Erae

> Lab while playing through the plugin, click the power icon on the **Sound** tab in Erae Lab's top header bar to disable Erae Sound, this stops its audio engine while keeping the device connected for layout editing and sync.

20.1.3 macOS: Removing Stale Device Entries

After a firmware update, macOS sometimes retains the old USB device entry alongside the new one. This can cause duplicate or numbered port names (e.g., `Erae 2 2`).

To clean up:

1. Open **Audio MIDI Setup** (in `/Applications/Utilities/`).
2. Select **Window -> Show MIDI Studio** (or press **Command-2**).
3. Locate any old `Erae 2` entry with a warning icon or a numbered suffix.
4. Right-click it and choose **Remove Device**.
5. Disconnect and reconnect the USB-C cable. The device re-enumerates cleanly with a single entry.

> **Tip:** After removing the stale entry, restart any open DAW so it re-reads the updated port list.

20.1.4 Windows: Confirming Port Visibility

1. Open **Device Manager** and expand the **Sound, video and game controllers** or **Universal Serial Bus controllers** section.
2. Confirm Erae 2 appears without a yellow warning icon.
3. In your DAW, trigger a MIDI device rescan or restart the application. Windows MIDI ports are registered at driver level: they do not hot-reload in all hosts.

> **Tip:** If the device appears but MIDI data does not flow, check that your DAW is not set to "exclusive mode" on the MIDI port, which prevents sharing with other applications.

20.2 Ableton Live

Ableton Live 11 and 12 are fully compatible with Erae 2 for both standard MIDI and MPE. A MIDI Remote Script is available from Embodme for the LivePad clip and scene control element.

20.2.1 Step 1: Enable the MIDI Ports

1. Open **Ableton Live -> Preferences** (macOS: **Live -> Settings**, Windows: **Options -> Preferences**).

2. Click the **Link, Tempo & MIDI** tab (Live 11: **Link/Tempo/MIDI**).
3. In the **MIDI Ports** section, locate the Input row for **`Erae 2 MIDI`**.
    - Set **Track** to **On**: this allows MIDI data from the port to be received on instrument tracks.
    - Set **Remote** to **On**: this allows the port to be used by a Control Surface script for clip launching and parameter mapping.
4. Locate the Input row for **`Erae 2 MIDI (MPE)`**.
    - Set **Track** to **On**.
    - Leave **Sync** and **Remote** off for this port (they are not needed for the MPE port).
5. Optionally enable the Output row for `Erae 2 MIDI` if you want Live to send MIDI back to Erae 2 (for example clock sync or LivePad LED feedback). The MPE port has no output to Erae 2.

> **Pitfall:** If you only enable `Erae 2 MIDI` and not `Erae 2 MIDI (MPE)`, MPE instruments will not receive expressive data. Both ports must have Track enabled for a full expressive setup.

20.2.2 Step 2: Set Up a Standard MIDI Instrument Track

1. Create an **Audio** or **MIDI** track with your target instrument.
2. In the track's **MIDI From** selector (the top input selector on the track), choose **`Erae 2 MIDI`**.
3. Set the channel to **All Channels** or the specific channel matching your Erae 2 element's MIDI channel assignment.
4. Arm the track for recording or set the monitoring to **In** to hear it in real time.

20.2.3 Step 3: Set Up an MPE Instrument Track

1. Create a MIDI track with an MPE-compatible instrument (Ableton's **Drift**, **Meld**, or any MPE plugin).
2. In **MIDI From**, choose **`Erae 2 MIDI (MPE)`**.
3. Set the channel to **All Channels**: MPE requires all 16 channels to be received simultaneously.
4. Open the instrument's plugin settings and enable **MPE mode** if the instrument has an explicit toggle.

> **Tip:** Ableton's own MPE instruments (Drift, Meld) enable MPE automatically when they detect a multi-channel input. Third-party MPE plugins may require a manual toggle in their settings panel.

20.2.4 Step 4: Install the MIDI Remote Script (LivePad Element)

The Erae 2 MIDI Remote Script enables the LivePad element to launch clips, fire scenes, and navigate the session grid in Ableton Live.

1. Download the **Erae 2 Remote Script** package from the Embodme support site.
2. Copy the script folder (named `Erae2`) into the Ableton User Library MIDI Remote Scripts folder:
   - **macOS:** `~/Library/Preferences/Ableton/Live x.x.x/User Remote Scripts/`
   - **Windows:** `C:\Users\<username>\Documents\Ableton\User Library\Remote Scripts\`
3. Restart Ableton Live.
4. Open **Preferences -> Link, Tempo & MIDI**.
5. In the **Control Surface** section, click an empty slot and select **Erae2** from the dropdown.
6. Set the **Input** to **`Erae 2 MIDI`** and the **Output** to **`Erae 2 MIDI`**.

The LivePad element on the surface will now reflect clip state in real time (playing, stopped, queued) using LED color feedback.

> **Pitfall:** If the Control Surface dropdown shows Erae2 but clip colours do not update, confirm the Remote Script's **Output** port is set to `Erae 2 MIDI` rather than the MPE port. The LED feedback goes over the Main cable.

20.2.5 MIDI Clock Sync (Ableton -> Erae 2)

To synchronise Erae 2's arpeggiator and looper to Ableton's tempo:

1. In **Preferences -> Link, Tempo & MIDI**, find the Output row for **`Erae 2 MIDI`**.
2. Set **Sync** to **On**.
3. On Erae 2, open **Settings** and set **Clock source** to `USB-dev`.
4. Press Play in Ableton: Erae 2 will lock to Ableton's tempo automatically.

20.3 Logic Pro

Logic Pro X (10.5+) and Logic Pro for Mac support MPE natively. Each MPE-capable track uses multi-channel recording, and per-channel expression is preserved in the MIDI region.

20.3.1 Step 1: Enable MIDI Input in Audio MIDI Setup

1. Open **Audio MIDI Setup** and confirm Erae 2 appears in the **MIDI Studio** window with the standard and MPE ports visible.
2. If no ports appear, click the **Rescan MIDI** button (the curved arrow icon at the top of the MIDI Studio window).

20.3.2 Step 2: Configure Logic's MIDI Preferences

1. Open **Logic Pro -> Settings -> MIDI** (macOS Ventura+) or **Logic Pro -> Preferences -> MIDI** (earlier versions).
2. In the **General** tab, confirm **Auto-demix by channel if multi-channel recording** is enabled. This ensures MPE data on different channels is not collapsed onto a single channel during recording.
3. Leave the default port assignments unchanged: Logic scans and exposes all available MIDI inputs automatically.

20.3.3 Step 3: Create a Standard MIDI Track

1. In the **Tracks** area, create a new Software Instrument track.
2. Open the track's **Region Inspector** or the **Channel Strip** and confirm the MIDI Input is set to **All** or to the specific channel matching your Erae 2 layout.
3. Logic automatically receives from all connected MIDI devices. If you want to restrict input to Erae 2 only, use the **External Instrument** plugin in the instrument slot and assign the input to `Erae 2 MIDI`.

20.3.4 Step 4: Enable MPE Per Track

1. Select the Software Instrument track you want to play expressively.
2. In the **Track Inspector** (left panel), click the **i** (Information) button to expand track settings.
3. Locate the **MIDI Channel** parameter and set it to **All**: required for MPE multi-channel recording.
4. Click the **MPE** toggle in the Track Inspector to enable per-note expression. When enabled, Logic records each note's pitch bend, pressure, and slide independently.

> **Pitfall:** If MPE is not enabled per track, Logic merges all channels onto channel 1 and per-note expression collapses. The MPE toggle must be active on each track that receives from `Erae 2 MIDI (MPE)`.

20.3.5 Step 5: Point the Track at the MPE Port

1. In the track's **Channel Strip**, click the **MIDI In** section.
2. From the input source dropdown, select **`Erae 2 MIDI (MPE)`**.
3. Set the channel to **All**.

20.3.6 Logic Pro: "Reset MIDI Drivers" Workaround

On some macOS versions, Logic caches MIDI device state at launch. If Erae 2 ports appear in Audio MIDI Setup but are missing from Logic's port selector:

1. In Logic, open **Window -> Show MIDI Environment** (or press **Command-0**).
2. From the **Midi Environment** menu bar, choose **Special -> Reset All MIDI Drivers**.
3. Logic rescans all connected MIDI devices without requiring a restart.
4. Close the MIDI Environment window and return to the Tracks area: the ports should now appear in the input selectors.

> **Tip:** If the MIDI Environment feels unfamiliar, a simpler workaround is to quit Logic, disconnect and reconnect Erae 2, and relaunch Logic. The MIDI driver cache is cleared during application startup.

20.3.7 MIDI Clock Sync (Logic -> Erae 2)

1. In **Logic Pro -> Settings -> MIDI**, enable **Transmit MIDI Clock**.
2. Tick the checkbox next to `Erae 2 MIDI` in the clock destination list.
3. On Erae 2, open **Settings** and set **Clock source** to `USB-dev`.

20.4 FL Studio

FL Studio (version 20+) does not implement native MPE in the same way as the DAWs above. Per-note expression requires manual per-channel routing or the use of Patcher for multi-instance instrument setups. For standard polyphonic use, FL Studio works well with Erae 2's `Erae 2 MIDI` port.

20.4.1 Step 1: Enable the MIDI Ports

1. Open **Options -> MIDI Settings**.
2. In the **Input** section, find **`Erae 2 MIDI`** in the device list.
3. Click it to highlight it, then check **Enable** to activate the port.
4. Assign it to an available **Port** number (e.g., Port 0). Note this number: you will use it to route MIDI to specific instruments.
5. Repeat for **`Erae 2 MIDI (MPE)`** if you plan to use per-channel routing, assigning it a different port number (e.g., Port 1).
6. Enable **Send master sync** if you want FL Studio's transport to sync Erae 2's clock.

> **Pitfall:** FL Studio's MIDI port enable is not persistent across sessions by default. If the port appears unchecked after a restart, re-enable it in MIDI Settings and save your FL Studio project immediately.

20.4.2 Step 2: Route MIDI to an Instrument

1. In the **Channel Rack**, right-click the instrument you want to receive from Erae 2.
2. Select **Receive notes from** and choose the Port number you assigned to `Erae 2 MIDI`.
3. Set the **MIDI channel** in the Receive Notes dialog to match the channel assigned in your Erae 2 layout (default: channel 1).
4. Play the surface: notes should now trigger the instrument.

> **Tip:** If you want multiple instruments to receive from Erae 2 simultaneously on different channels, assign each instrument a different MIDI channel in the Channel Rack's MIDI settings and match it to the per-element channel assignments in Erae 2.

20.4.3 Step 3: CC Mapping with Multi-Link

FL Studio's **Multi-Link to Controllers** feature is the fastest way to map Erae 2's CC outputs to instrument parameters.

1. Open the **Mixer** or the instrument plugin you want to control.
2. From the menu bar, choose **Tools -> Multi-link to controllers** (or right-click any knob and select **Link to controller**).
3. Move the target knob or fader in FL Studio.
4. Touch the corresponding surface element on Erae 2 to send the CC: FL Studio captures the CC number automatically and creates the mapping.
5. Click **Accept** to confirm.

> **Tip:** Erae 2's fader elements send CC on the axes you configure in Erae Lab. A vertical 1D fader sends CC Y Absolute by default; a 2D fader can send both X and Y simultaneously. Map each axis to a separate parameter for two-dimensional expression from a single touch zone.

20.4.4 Step 4: Per-Channel MPE Approximation in FL Studio

FL Studio does not natively handle MPE messages as a unified instrument concept. The practical approach for expressive playing is to route each MIDI channel to a separate instrument instance:

1. Assign Erae 2's MPE keyboard element to Lower Zone (Master Channel 1, member channels 2-N).
2. In FL Studio, add N instances of your target instrument to the Channel Rack.
3. Set each instance to **Receive notes from Port 1** (the port assigned to `Erae 2 MIDI (MPE)`), with channels 1, 2, 3… respectively.
4. Connect all instances to the same Mixer track if you want unified audio output.

For instruments that support Patcher (FL's modular environment), a single Patcher instance can host multiple sub-instruments and route each incoming channel to its own sub-instrument automatically, providing a more manageable setup.

20.4.5 MIDI Clock Sync (FL Studio -> Erae 2)

1. In **Options -> MIDI Settings**, select `Erae 2 MIDI` in the output section, enable it, and assign it a port.
2. Enable **Send master sync** for that port.
3. On Erae 2, open **Settings** and set **Clock source** to `USB-dev`.

20.5 Bitwig Studio

Bitwig Studio (version 3.2+) has among the most comprehensive MPE support available in any DAW. Note Expression mode lets you view and edit per-note pitch bend, pressure, and timbre in the Piano Roll as individual automation lanes per note.

20.5.1 Step 1: Configure the Controller

1. Open **Bitwig Studio -> Settings -> Controllers**.
2. Click the **+** button to add a controller.
3. Choose **Generic -> Generic Flexi** from the manufacturer/script list (or, for full MPE integration, choose **ROLI -> Seaboard Rise**: this script works correctly with any MPE device).
4. Set the **Input** to **`Erae 2 MIDI`** and the **Output** to **`Erae 2 MIDI`**.

> **Tip:** The Generic Flexi script allows you to map any CC or Note message to any Bitwig parameter. If you only need notes and CC without ROLI-specific features, Flexi is simpler to configure. If you want full MPE note expression in Bitwig's Note Expression lanes, use the ROLI Seaboard Rise script: it functions identically for any MPE transmitter.

20.5.2 Step 2: Enable MPE Input on an Instrument Track

1. Create an **Instrument** track with an MPE-capable plugin or Bitwig's own **Polysynth** or **Phase-4**.
2. In the track header, click the **MIDI Input** selector.
3. Choose **`Erae 2 MIDI (MPE)`** as the input port.
4. Set the channel to **All**: Bitwig needs to see all 16 channels to reconstruct per-note expression data.

20.5.3 Step 3: Verify Note Expression

1. Record a short passage of expressive playing.
2. Open the **Piano Roll** for the recorded region.

3. Click any note: you should see per-note expression lanes below the note for **Pitch**, **Pressure**, and **Timbre (CC#74)**. Each lane shows the continuous data recorded for that note individually.

> **Pitfall:** If all notes show identical expression data rather than per-note data, confirm the track input is set to `Erae 2 MIDI (MPE)` and not `Erae 2 MIDI`. Standard MIDI channel pressure is a per-channel message; per-note expression requires the MPE cable.

20.5.4 Step 4: Map CC with Generic Flexi

1. In **Settings -> Controllers**, click the Flexi script entry, then open its settings panel.
2. Click **Learn** next to a binding slot.
3. Move a fader or CC source on Erae 2: Bitwig captures the CC number automatically.
4. Assign it to any Bitwig device parameter using the parameter browser in the same panel.

20.5.5 MIDI Clock Sync (Bitwig -> Erae 2)

Bitwig sends MIDI clock to any output port that has Sync enabled:

1. In **Settings -> Controllers**, select the Erae 2 controller entry.
2. Enable **Send Clock** in the controller options.
3. On Erae 2, open **Settings** and set **Clock source** to `USB-dev`.

20.6 Cubase / Nuendo

Cubase Pro (version 10.5+) and Nuendo support per-note MIDI expression via the **Expression Map** system and the **MIDI Polyphonic Expression** device declaration. Cubase 12+ adds dedicated MPE track support.

20.6.1 Step 1: Enable MIDI Ports

1. Open **Studio -> Studio Setup** (Cubase) or **Devices -> Device Setup** (older versions).
2. Select **MIDI Port Setup** in the left panel.
3. Find **`Erae 2 MIDI`** in the list. Ensure the **In** (input) column is checked (visible/active).
4. Find **`Erae 2 MIDI (MPE)`** and check its **In** column as well.
5. Click **OK** to apply.

> **Pitfall:** Cubase has separate **Visible** and **Active** columns. A port must be marked **Active** to

> appear in track input selectors. A port marked **Visible** but not **Active** appears in the list but does not pass data.

20.6.2 Step 2: Create a Standard MIDI or Instrument Track

1. Create a new **Instrument Track** (or **MIDI Track** routed to a rack instrument).
2. In the track's **Inspector** (left panel), set **MIDI Input** to **`Erae 2 MIDI`**.
3. Set the **Channel** to **Any** or to the specific channel your Erae 2 element uses.

20.6.3 Step 3: Declare Erae 2 as a Poly Expression Device (Cubase 12+)

Cubase 12 introduced explicit MPE support. To enable it:

1. Open **Studio -> Studio Setup**.
2. Under **MIDI Port Setup**, locate `Erae 2 MIDI (MPE)` and click the **Edit** (pencil) icon.
3. Set the **Device Type** to **MPE Instrument**.
4. Set **Lower Zone Master Channel** to **1** (matching Erae 2's default MPE configuration).
5. Set **Member Channels** to the number of simultaneous voices you need (default: 15 for Lower Zone, using channels 2-16).
6. Click **OK**.

20.6.4 Step 4: Create an MPE Instrument Track

1. Create a new **Instrument Track** with an MPE-capable VST3 instrument.
2. In the Inspector, set **MIDI Input** to **`Erae 2 MIDI (MPE)`**.
3. Set Channel to **Any**.
4. Open the instrument's own settings and activate its MPE mode (typically labelled **Poly Expression**, **MPE**, or **Per-Note Modulation**).

> **Tip:** Not all VST3 instruments support the Note Expression API. Instruments that do (Komplete Kontrol, Equator2, Pigments, Serum 2) will show per-note modulation lanes in the Key Editor when MPE input is active.

20.6.5 Step 5: MPE Zone Configuration Per Layout

If your Erae 2 layouts use different MPE zone configurations (some layouts use Lower Zone on Ch 1, others might use Upper Zone on Ch 16), you can save matching MIDI Device setups in Cubase's Studio Setup panel and switch between them as needed. Most users only ever need the Lower Zone default.

20.6.6 MIDI Clock Sync (Cubase -> Erae 2)

1. Open **Transport -> Project Synchronization Setup**.
2. Ensure **MIDI Clock** is enabled.
3. In the **MIDI Clock Destinations** list, tick **`Erae 2 MIDI`**.
4. On Erae 2, open **Settings** and set **Clock source** to `USB-dev`.

20.7 Reaper

Reaper (version 6.0+) supports MPE natively. MPE data is stored per-note as separate pitch bend and CC envelopes in the MIDI editor, and per-note expression is preserved across recording, playback, and editing.

20.7.1 Step 1: Enable MIDI Devices

1. Open **Options -> Preferences -> Audio -> MIDI Devices**.
2. In the **MIDI Inputs** list, locate **`Erae 2 MIDI`** and double-click it to enable it (shown with a checkmark).
3. Locate **`Erae 2 MIDI (MPE)`** and double-click to enable it as well.
4. Click **Apply**.

> **Pitfall:** Reaper requires the device to be connected before it appears in the MIDI Devices list. If you plugged Erae 2 in after opening Preferences, click the **Reset** button to rescan.

20.7.2 Step 2: Create a MIDI Track for Standard Use

1. Insert a new track (**Track -> Insert New Track**).
2. Click the track's **ARM** button (red circle).
3. The track's input defaults to all MIDI. To restrict to Erae 2 specifically, click the track's input selector and choose **MIDI Input -> `Erae 2 MIDI` -> All Channels**.

20.7.3 Step 3: Enable MPE Input on a Track

1. Insert a track and arm it for recording.
2. Click the track input selector and choose **MIDI Input -> `Erae 2 MIDI (MPE)` -> All Channels**.
3. Insert a VST3 MPE-capable instrument on the track (e.g., Pigments, Equator2).
4. In the instrument's settings, enable its MPE mode.

Reaper records all 16 MIDI channels in a single take, preserving full per-note expression. In the MIDI editor, enable **View -> Show per-note pitch** to see individual pitch bend curves per note event.

20.7.4 Step 4: MPE Pitch Bend Range Matching

Erae 2's MPE keyboard element defaults to a pitch bend range of 48 semitones. Ensure your instrument matches:

1. In the MIDI editor for a recorded region, right-click the pitch bend lane.
2. Set the **Pitch Bend Range** to **±48 semitones** (or whatever value matches the Erae 2 element setting).
3. In the instrument plugin, set its pitch bend range to the same value.

> **Tip:** Mismatched pitch bend ranges are the most common cause of pitch tracking errors in MPE setups. If playing one octave up sounds like two octaves up, the ranges are different. Set both sides to 48 semitones.

20.7.5 Step 5: MIDI Clock Sync (Reaper -> Erae 2)

1. Open **Options -> Preferences -> MIDI Devices**.
2. Enable the `Erae 2 MIDI` output device.
3. Open **File -> Project Settings -> Notes, Media** and enable **Send MIDI clock to** the Erae 2 MIDI output.
4. On Erae 2, open **Settings** and set **Clock source** to `USB-dev`.

20.8 Connecting to Hardware Synthesizers

Erae 2 is designed for direct hardware integration. Two physical ports carry MIDI to synthesizers and drum machines without a computer in the signal path.

20.8.1 TRS MIDI Output

The rear panel includes a 3.5 mm TRS MIDI output. Two TRS-to-DIN 5-pin adapters are included in the box.

Wiring type: the Erae 2 TRS MIDI jacks use **Type A** wiring, and there is no setting to change it. Gear with a Type A TRS input connects with a plain TRS cable; for gear that expects Type B, use a Type A to Type B converter cable. Check your synthesizer's manual for its expected TRS type: a wiring mismatch is the most common cause of hardware MIDI not working over TRS.

Connection steps:

1. Connect the included TRS adapter to the rear MIDI Output A jack on Erae 2.
2. Connect a standard 5-pin DIN MIDI cable from the adapter to your synthesizer's MIDI In socket.
3. In Erae Lab (or the LCD mapping screen), open the element you want to route to the hardware synth.
4. In **MIDI Output Destination**, enable **MIDI A** (or **MIDI B** if using the second output).
5. Set the element's **MIDI Channel** to match the synthesizer's receive channel.
6. Play the surface: notes should sound through the hardware synthesizer.

> **Tip:** You can enable multiple destinations simultaneously. An element with both **USB Device** and **MIDI A** checked sends to your DAW and your hardware synth at the same time.

20.8.2 USB Host Port

The USB Host port allows Erae 2 to act as a USB host, powering and communicating with class-compliant USB MIDI devices without a computer.

Power budget: The USB Host port provides up to 500 mA. Devices that require more (e.g., some USB keyboards with backlighting) may need external power.

Connection steps:

1. Connect a USB MIDI synthesizer, drum machine, or USB-to-DIN adapter to the USB Host port on the rear panel.
2. The connected device appears as the **USB Host** port in Erae 2's MIDI routing.
3. In Erae Lab, open the element's settings and enable **USB Host** in the **MIDI Output Destination** bitmask.
4. Set the MIDI channel to match the synthesizer's receive channel.

> **Tip:** If the connected USB device is not recognized, confirm it is class-compliant (no driver required on macOS/Windows). Devices that require a proprietary driver are not compatible with USB Host mode.

20.8.3 MIDI Routing Matrix for Hardware Setup

The routing matrix (accessible at **Settings -> Routing** on the LCD) controls which ports relay incoming MIDI to other ports. For hardware setups:

Useful Route	Why
USB Device -> MIDI A	Send DAW MIDI through Erae 2 to hardware synth on MIDI A
USB Device -> MIDI B	Send DAW MIDI to a second hardware device on MIDI B

Useful Route	Why
MIDI -> USB Device	Forward hardware MIDI into your DAW for recording
USB Host -> MIDI A	Chain USB MIDI device output to TRS gear
MIDI -> MIDI A	Hardware MIDI thru without a computer

Worked example: Erae 2 as USB MIDI interface

To use Erae 2 as a two-way MIDI interface between your DAW and a hardware synth:

1. Enable **MIDI -> USB Device**: hardware synth MIDI output flows into your DAW.
2. Enable **USB Device -> MIDI A**: your DAW sends MIDI out through Erae 2 to the synth.
3. The synth is now available as a DAW-controllable instrument with no separate MIDI interface required.

Worked example: routing surface notes to a hardware synth

1. In Erae Lab, select the keyboard element on your layout.
2. In **MIDI Output Destination**, enable **MIDI A** (and disable USB Device if you do not want the notes to also go to the DAW).
3. Set the MIDI channel to the synth's receive channel (e.g., channel 1).
4. Connect a TRS adapter and cable from MIDI Output A to the synth's MIDI In.
5. Play the surface: notes route directly to the synth, with no computer involved.

> **Tip:** Routing settings are saved per Project. Create a dedicated "hardware" project with the routing matrix pre-configured for your synth setup, and a separate "DAW" project for studio use.

20.9 Eurorack Integration

Erae 2 connects to Eurorack modular synthesizers in three complementary ways: native CV outputs, MIDI over TRS to MIDI-CV modules, and USB Host to USB MIDI-CV converter modules.

20.9.1 Native CV Outputs

Erae 2 provides **24 independent CV output channels** on 3.5 mm jacks, directly compatible with Eurorack and the 1 V/octave pitch standard. Voltage ranges and resolution are listed in Chapter 10.

Detailed CV configuration is covered in Chapter 10: CV Outputs. A summary for Eurorack connections:

20.9.1.1 First Voice: Pitch and Gate

1. In Erae Lab, create or edit an Isomorphic Keyboard element.
2. Set **CV Num Voice** to 1.
3. Assign the base output channel (e.g., channel 1). Channel 1 carries Pitch (1 V/oct); channel 2 carries Gate (0 V / 5 V).
4. Connect channel 1 to your VCO's **V/oct** input.
5. Connect channel 2 to your envelope generator's **Gate** input.
6. Play a note: pitch tracks at 1 V/octave, and the envelope fires on each note-on.

> **Tip:** Middle C (MIDI note 60) outputs 2 V and the C one octave above outputs 3 V. If octaves track slightly wide or narrow on your VCO, adjust **CV Gain (cent/oct)** in **Settings**, or press **Run CV calib** to recalibrate the outputs.

20.9.1.2 Additional Per-Voice Expressions

Each voice can output supplementary expression streams as CV. These consume one CV channel each, per voice:

CV Output	Voltage Range	Eurorack Use
Velocity	0-5 V	Envelope initial level, VCA offset
Pressure	0-5 V	Filter cutoff, VCA gain, LFO depth CV
X Position	0-5 V	Any parameter responding to horizontal position
Y Position	0-5 V	Any parameter responding to vertical position
X Slide	centred at 2.5 V	Vibrato depth, pitch deviation via CV
Y Slide	centred at 2.5 V	Timbre shift, formant position

To enable these outputs:

1. Select the element in Erae Lab, or open its Mapping screen on the device.
2. In the CV section, enable the desired outputs (Velocity, Pressure, X Position, etc.).
3. Each enabled output consumes one CV channel per voice. Plan your channel allocation across all elements before enabling high voice counts.

> **Warning:** Assigning 4 voices with all six optional outputs enabled occupies 4 × 8 = 32 CV channels: more than Erae 2's 24 available outputs. Keep voice count and optional output count balanced.

20.9.1.3 CV Clock Output for Eurorack Clock Sync

Erae 2's CV Clock output sends a pulse signal to a CV channel at a configurable ppqn rate, locked to the project tempo.

1. Press the **Settings** button on the front panel.
2. Navigate to **CV Clock**.
3. Set **Enabled** to On.
4. Set **Beat Division** to **24 ppqn** (standard MIDI clock rate, compatible with most Eurorack clock dividers and clocked modules).
5. Set **Clock Output** to an unused channel (e.g., channel 3).
6. Connect channel 3 to your clocked module's **Clock In** or clock divider input.
7. Connect the adjacent reset output to the module's **Reset** input for sample-accurate start alignment.

Available ppqn rates: 1, 2, 4, 8, 24, 48 ppqn. Use 48 ppqn for high-resolution LFO sync with compatible modules.

20.9.2 USB Host to Eurorack MIDI-CV Modules

For higher polyphony than the native CV outputs allow, or for setups where you want MIDI-CV conversion handled by a dedicated Eurorack module, connect the module directly to Erae 2's USB Host port.

The module receives MIDI from Erae 2 (including MPE if the module supports it) and converts it to CV without a computer in the path.

Suggested modules:

Module	Best For
Expert Sleepers FH-2	High polyphony, MPE support, deeply configurable
Intellijel uMIDI 1U	Compact single-voice MIDI-to-CV with USB input
Mutable Instruments Yarns	Up to 4-voice polyphony, arpeggiator, MIDI clock
Endorphin.es Shuttle Control	16-voice USB MIDI, extensive CV mapping
Befaco MIDI Thing	Simple, affordable TRS MIDI to dual CV

> **Tip:** The Expert Sleepers FH-2 is the most capable option for Erae 2 integration. It accepts USB MIDI directly from the USB Host port, supports MPE input (assign each member channel to a separate CV output pair), and can be fully configured from its browser-based editor.

Connection steps for Expert Sleepers FH-2:

1. Connect a USB-A cable from Erae 2's USB Host port to the FH-2's USB port.
2. The FH-2 enumerates as a USB MIDI device on Erae 2's USB Host.
3. If you want MIDI from the FH-2 to reach your computer, enable **USB Host -> USB Device** in **Settings -> Routing**.
4. In Erae Lab, set your keyboard element's **MIDI Output Destination** to include **USB Host**.
5. Configure the FH-2 in its browser editor: assign MIDI channels to CV output pairs, set pitch bend range to 48 semitones for MPE voices.

20.9.3 Example Patch: Full Expressive Voice

This example routes Erae 2 through a complete Eurorack voice using native CV outputs and gives independent per-note expression.

Hardware: - Erae 2 (CV outputs via 3.5 mm jacks) - Any Eurorack VCO with V/oct and linear FM inputs - Any ADSR envelope generator - Any VCF with CV cutoff input - Any VCA with CV gain input

Patch:

Erae 2 Output	Eurorack Destination	Result
CV Ch 1 (Pitch)	VCO V/oct	Pitch tracks touch position
CV Ch 2 (Gate)	ADSR Gate In	Envelope fires on touch
CV Ch 3 (Velocity)	ADSR Initial Level	Attack level reflects strike velocity
CV Ch 4 (Pressure)	VCF Cutoff CV	Filter opens with increasing pressure
CV Ch 5 (Pressure)	VCA Gain CV	Volume swells with pressure
CV Ch 6 (X Slide)	VCO Linear FM	Horizontal slide = pitch deviation (vibrato)

Setup in Erae Lab:

1. Create an Isomorphic Keyboard element.
2. Set **CV Num Voice** to 1.
3. Set base channel to 1.
4. Enable **Velocity**, **Pressure**, and **X Slide** additional outputs.
5. Channels 1-6 are now assigned; connect as shown in the table.
6. Set **Pressure Type** to `PolyPressure` in the element settings: this maps individual finger pressure to the correct CV per note (for polyphonic setups).

> **Tip:** Start with a monophonic patch (CV Num Voice = 1) to verify all CV signals behave as expected before moving to a polyphonic configuration. Use a voltmeter on the pitch output to confirm 1 V/octave tracking before the VCO is patched in.

20.10 Related Topics

- MIDI Configuration: USB port layout, channel assignment, MPE zones, routing matrix
- CV Outputs: complete CV output reference, calibration, per-voice assignment
- Settings: clock source, MIDI routing, CV clock configuration
- Troubleshooting: MIDI port not appearing, notes not sounding, routing loops
- Appendix B: MIDI Implementation Chart: full message table, MPE detail, SysEx protocol

21 Appendix D: Developer API

Erae 2 exposes a programmable surface for advanced integrations: custom Max/MSP patches, TouchDesigner setups, web tools, bespoke software instruments. The API is delivered over USB MIDI System Exclusive (SysEx) and operates on **API Zone** elements placed in the layout. A host application can:

- Receive raw multi-touch data (X, Y, pressure) from one or more API zones, without the MIDI abstraction layer.
- Paint the LEDs of each API zone directly: clear, set pixels, fill rectangles, draw images.
- Query the size of each API zone and the API protocol version.

This appendix documents the wire protocol. For an overview of the API Zone element from a layout author's perspective, see Section 5.9: API Zone.

> **Compatibility.** Erae 2 answers only the Erae 2 SysEx header (family member code `00 02`) addressed to device ID `0x01`. Host software written for the Erae Touch, which uses the member code `00 01`, must change that code to talk to an Erae 2. The command IDs and payload layouts are the same on both.

21.1 D.1 Transport

All API messages travel over the main USB MIDI port.

Direction	USB MIDI Port
Device -> Host (finger streams, replies)	`Erae 2 MIDI`
Host -> Device (draw, clear, query)	`Erae 2 MIDI`

The `Erae 2 MIDI (MPE)` port carries only MPE output and is not used by the API.

> **Connection check.** From a WebMIDI host, open the input and output named `Erae 2 MIDI`, the one whose name does not contain `MPE`.

> **Linux.** When Erae 2 is in MIDI 2.0 mode, the Linux kernel's conversion from its legacy MIDI ports to MIDI 2.0 packets can corrupt SysEx messages longer than 6 data bytes, so the device receives garbled commands and does not answer. If your tool gets no reply over a legacy ALSA port (for example with `amidi -p hw:...`), send the MIDI 2.0 packets directly to the device's UMP port (`/dev/snd/umpC<card>D0`). This does not affect macOS or Windows.

21.2 D.2 SysEx Header

Every host to device API message begins with a fixed 11-byte header (including the `F0` start byte), then the sub-service byte, and ends with `F7`:

```
F0  00 21 50  00 01  00 02  01    01   04   <sub>  <payload...>  F7
^   ^^^^^^^^  ^^^^^  ^^^^^  ^^    ^^   ^^   ^^^^^
|   Embodme   Erae   Erae 2 ID    Svc  API  Sub-
SysEx  manuf  family model        grp  svc  service
start  ID
```

Bytes	Meaning
`F0`	SysEx start
`00 21 50`	Embodme manufacturer ID
`00 01`	Erae hardware family code
`00 02`	Erae 2 model code
`01`	Device ID. Erae 2 answers only `0x01`; the broadcast ID `0x7F` is ignored by the API
`01`	Services group
`04`	Service: API
`<sub>`	Sub-service ID (see D.4)
`<payload...>`	Sub-service specific bytes
`F7`	SysEx end

Device to host messages use a different framing, described in D.3.4.

21.3 D.3 Conventions

21.3.1 D.3.1 Coordinate axes

The Erae surface is a 42-wide by 24-tall LED grid (1008 LEDs). Within an API zone, coordinates are **zone-local**: `(0, 0)` is one corner of the zone and `(width-1, height-1)` the opposite corner. X increases from left to right.

> **One Y convention: bottom-origin, Y up.**
>
> Finger data and drawing commands share the same axes: the origin is the bottom-left corner of the zone and Y increases upward.
>
> | Stream | Y convention | `y = 0` means |
> |---|---|---|
> | Finger data (device -> host) | **Bottom-origin** | bottom row of the zone |
> | Drawing commands (host -> device) | **Bottom-origin** | bottom row of the zone |
>
> To light the LED under a finger reported at `(x, y)`, draw at `SetPixel(x, y, color)`, no flip. `DrawImage` follows the same axes: image row 0 is painted on row y of the zone, and each following image row one row higher.

21.3.2 D.3.2 Color encoding

`SetPixel` and `DrawRectangle` take colours as three 7-bit MIDI bytes (R, G, B). The device shifts each component one bit left:

```
display_R = sysex_R << 1   // 0..254 in steps of 2
display_G = sysex_G << 1
display_B = sysex_B << 1
```

A full white pixel is therefore `7F 7F 7F`; pure red is `7F 00 00`.

`DrawImage` is different: its pixels are full 8-bit values (`0x00` to `0xFF`), carried through MIDI with the bitize-7 packing below and used without any shift.

21.3.3 D.3.3 Bitize-7 (image and finger data encoding)

MIDI forbids data bytes with bit 7 set. Binary blocks are therefore sent with an Embodme 7-bit packing: every 7 source bytes become 8 MIDI bytes. The first MIDI byte of a group carries the most significant bit of each of the following 7 source bytes (source byte 0 in bit 6, down to source byte 6 in bit 0), and each following byte carries the lower 7 bits of one source byte.

Pseudo-code for the packer (host side):

```
for chunk of 7 source bytes:
   out.push(((src[0] >> 7) << 6) | ((src[1] >> 7) << 5) | ... | ((src[6] >> 7) << 0))
    for i in 0..7:
        out.push(src[i] & 0x7F)
# A trailing partial chunk emits its MSB byte first, then only the bytes present.
```

The encoded size of `N` source bytes is `ceil(N * 8 / 7)`. The **checksum** of a block is the XOR of every byte of the *encoded* (bitized) block, including each group's leading MSB byte. Because every encoded byte is 7-bit, the checksum is too.

> **Common mistake.** Do **not** checksum the decoded source bytes. It happens to match for images whose components are all `0x7F` or less, but not for brighter images, and the firmware silently drops a `DrawImage` whose checksum does not match. Checksum the bytes you put on the wire.

> **Reference implementation.** A public Python example of the API, including the bitize-7 packer, the checksum routine and image upload, is available in the `erae_api_sysex` repository (`run_erae_api.py`), together with the `Erae_API_V2.pdf` protocol specification.

21.3.4 D.3.4 Streaming activation and reply framing

Finger data and `ZoneBoundaryRequest` replies are only sent **after** the host has sent `StartFingerDataStreaming` (`0x01`). The streaming state is global to the device, not per zone.

Every device to host API message is framed as:

```
F0  <prefix data>  <body>  F7
```

There is no Embodme header in these messages: the **prefix data** supplied by the host (with `StartFingerDataStreaming`, or with `VersionRequest` for the version reply) is copied verbatim right after `F0`. Use it to make the replies valid SysEx for your host and to tell several Erae devices apart, for example by starting the prefix with a manufacturer ID byte such as `7D` (reserved for non-commercial use) followed by a session tag.

21.4 D.4 Host -> Device Commands

All values are hex unless noted. <header> stands for `F0 00 21 50 00 01 00 02 01 01 04`, as shown in D.2.

21.4.1 D.4.1 StartFingerDataStreaming (0x01)

Enables finger data and zone boundary replies. The optional <prefix data> (0 to 16 bytes) is copied at the start of every device to host message (see D.3.4). Sending it again replaces the prefix.

```
<header>  01  <prefix data 0..16 bytes>  F7
```

21.4.2 D.4.2 EndFingerDataStreaming (0x02)

Disables finger data and zone boundary replies and clears the prefix. No payload.

```
<header>  02  F7
```

21.4.3 D.4.3 ZoneBoundaryRequest (0x10)

Asks for the size of an API zone. The device replies with a Zone Boundary Reply. If streaming is disabled, no reply is sent.

```
<header>  10  <zoneIdx>  F7
```

Byte	Field	Range	Notes
0	zoneIdx	0x00 .. 0x7F	Zone index, as set in Erae Lab

21.4.4 D.4.4 Clear (0x20)

Clears the LEDs of one zone (fills it with black).

```
<header>  20  <zoneIdx>  F7
```

21.4.5 D.4.5 SetPixel (0x21)

Writes a single LED.

```
<header>  21  <zoneIdx>  <x>  <y>  <R>  <G>  <B>  F7
```

Byte	Field	Range	Notes
0	`zoneIdx`	`0x00 .. 0x7F`	Target zone
1	`x`	`0 .. width-1`	Out-of-range writes are silently dropped
2	`y`	`0 .. height-1`	**Bottom-origin** (see D.3.1)
3	`R`	`0x00 .. 0x7F`	Shifted left by 1 internally
4	`G`	`0x00 .. 0x7F`	
5	`B`	`0x00 .. 0x7F`	

21.4.6 D.4.6 DrawRectangle (`0x22`)

Fills a rectangle with one colour (7-bit components, as for `SetPixel`).

```
<header>  22  <zoneIdx>  <x>  <y>  <w>  <h>  <R>  <G>  <B>  F7
```

The rectangle covers cells `(x .. min(x+w, width) - 1, y .. min(y+h, height) - 1)`. If `x >= width` or `y >= height`, nothing is drawn. Otherwise the rectangle is clipped to the zone.

21.4.7 D.4.7 DrawImage (`0x23`)

Draws a `w x h` colour image. Pixels are read in **row-major** order: row 0 left to right, then row 1, and so on. Row 0 is the **bottom** row of the image (zone row y), row 1 the row above it. Each pixel is three 8-bit bytes (R, G, B, each `0x00` to `0xFF`), used as is.

```
<header> 23 <zoneIdx> <x> <y> <w> <h> <bitized RGB data...> <chksum> F7
```

Byte	Field	Notes
0	`zoneIdx`	Target zone
1	`x`	Bottom-left X of the image inside the zone
2	`y`	Bottom-left Y (bottom-origin)
3	`w`	Image width in pixels
4	`h`	Image height in pixels
5..N-1	`bitized RGB`	`w * h * 3` source bytes encoded as in D.3.3
N	`chksum`	XOR of every byte of the bitized data; see D.3.3

The firmware checks four things and silently drops the message if any fails:

1. The decoded size is a multiple of 3 (whole pixels only).
2. The decoded size equals exactly `w * h * 3`.
3. The image fits in the surface (`w * h <= 1008`).
4. The checksum matches the XOR of the bitized bytes.

The image is clipped to the zone's right and top edges (`xmax = min(x+w, width)`, `ymax = min(y+h, height)`). Pixels outside the zone are not drawn; pixels inside the zone come from the matching position in the source image. If x or y is outside the zone, nothing is drawn.

21.4.8 D.4.8 VersionRequest (`0x7F`)

Asks for the API protocol version. The device replies with a Version Reply, whether or not streaming is enabled.

```
<header>  7F  <prefix data>  F7
```

The `<prefix data>` of this request is copied into the reply (see D.3.4), so a host can pair the reply with its request even when several devices are present.

21.5 D.5 Device -> Host Messages

All device to host messages start with `F0` and the host prefix data, as described in D.3.4.

21.5.1 D.5.1 Zone Boundary Reply

Reply to ZoneBoundaryRequest.

```
F0  <prefix data>  7F  01  <zoneIdx>  <width>  <height>  F7
```

Byte	Field	Notes
0	`0x7F`	Marks a reply (not finger data)
1	`0x01`	Zone boundary reply
2	`zoneIdx`	Copied from the request
3	`width`	Zone width in pixels, or `0x7F` if no API zone uses this index
4	`height`	Zone height, or `0x7F` if none

21.5.2 D.5.2 Version Reply

Reply to VersionRequest.

```
F0  <prefix data>  7F  02  <apiVersion>  F7
```

<apiVersion> is 0x02. It identifies the device to host wire format, most importantly the Finger Stream body, and changes only if that format changes. A device that answers VersionRequest sends the Finger Stream body described below. Erae Touch firmware and early Erae 2 firmware do not answer this request.

21.5.3 D.5.3 Finger Stream

Sent for every touch event inside an API zone while streaming is enabled. All zones share the same stream and are told apart by zoneIdx.

```
F0 <prefix data> <action> <zoneIdx> <bitized fingerIdx (uint64)> <bitized X,Y,Z (3 x fl
```

Byte	Field	Notes
0	action	0x00 Click (finger down), 0x01 Slide (move), 0x02 Release (finger up)
1	zoneIdx	Source zone
2..11	fingerIdx	64-bit unsigned finger ID, bitize-7 encoded (8 bytes become 10 on the wire)
12..25	x, y, z	Three little-endian float32 values bitize-7 encoded as one 12-byte block (14 bytes on the wire): zone-local position in pixels (**Y bottom-origin**, see D.3.1) and normalized pressure z (about 0.0 to 1.0)
26	chksum	XOR of the 14 bitized bytes of the x, y, z block

Rebuild per-finger tracks by (zoneIdx, fingerIdx) and treat Release as the end of a track. Gestures replayed by the looper appear as fingers with their own IDs.

21.5.4 D.5.4 Max Num Fingers and Finger Data Rate

Erae Lab shows two more API Zone parameters, **Max Num Fingers** (`0` to `10`) and **Finger Data Rate**. They are saved in the layout, but the current Erae 2 firmware does not apply them: every touch in the zone is reported, and `Slide` messages follow the finger at the device's own scan rate. Size your host's track table for up to 10 simultaneous fingers per zone, and throttle on the host side if you need fewer updates.

21.6 D.6 Worked Examples

All examples target zone `0x00`, device ID `0x01`.

21.6.1 D.6.1 Light pixel `(3, 5)` red

`SetPixel`, R=`0x7F`, G=0, B=0:

```
F0 00 21 50 00 01 00 02 01 01 04 21  00 03 05 7F 00 00  F7
```

21.6.2 D.6.2 Clear zone

```
F0 00 21 50 00 01 00 02 01 01 04 20  00  F7
```

21.6.3 D.6.3 Start finger streaming with prefix `7D 01`

```
F0 00 21 50 00 01 00 02 01 01 04 01  7D 01  F7
```

Every message the device then sends starts with `F0 7D 01`. For example the reply to a `Version-Request` sent with the same prefix is `F0 7D 01 7F 02 02 F7`.

21.6.4 D.6.4 Drawing a 2x1 image of `[red, green]` at `(0, 0)`

Raw data (full 8-bit): `FF 00 00 00 FF 00` (6 bytes, one partial chunk).

```
MSB byte = bit7(FF)<<6 | bit7(00)<<5 | bit7(00)<<4 | bit7(00)<<3 | bit7(FF)<<2 | bit7(00)<<
payload  = 44 7F 00 00 00 7F 00
```

Checksum = XOR of the bitized bytes = `44 XOR 7F XOR 00 XOR 00 XOR 00 XOR 7F XOR 00 = 0x44`.

Full message:

```
F0 00 21 50 00 01 00 02 01 01 04 23  00 00 00 02 01  44 7F 00 00 00 7F 00  44  F7
                                     |  |  |  |  |  ----------------  |
                                    zo x  y  w  h  bitized payload    chksum
```

21.7 D.7 Error Handling and Limits

Condition	Behavior
Message addressed to a device ID other than `0x01`	Ignored
Zone index does not match any API zone in the active layout	Command silently dropped (no error reply)
x or y out of zone bounds	Command silently dropped (`SetPixel`) or clipped (`DrawRectangle`, `DrawImage`)
`DrawImage` checksum mismatch	Command silently dropped
`DrawImage` decoded size is not `w * h * 3`	Command silently dropped
`DrawImage` larger than the surface (1008 pixels)	Command silently dropped
`StartFingerDataStreaming` prefix longer than 16 bytes	Streaming is **not** enabled
API zone removed from the layout while streaming	Streaming continues; commands to that index are dropped until a zone with that index is present again
Several API zones with the same index	Only the first one receives draw commands; avoid duplicate indexes

Limits:

Limit	Value
Prefix data	16 bytes
`DrawImage` pixel count	1008 (the full surface)
Incoming SysEx message	8 KB. A message is kept while it arrives, however slowly; longer messages are discarded
Device to host message	128 bytes including the prefix

21.8 D.8 Putting It Together

A typical session looks like this:

1. Open the `Erae 2 MIDI` input and output.
2. Send `VersionRequest` with your prefix and check that the reply reports `0x02`.
3. Send `StartFingerDataStreaming` with the same prefix (for example a manufacturer ID byte and a session tag).
4. For each zone you use, send `ZoneBoundaryRequest` and keep the `(width, height)` reply.
5. Draw with `Clear`, `DrawImage`, `DrawRectangle` and `SetPixel`. A full-surface image is about 3.5 KB on the wire; pace large images so your host's MIDI output keeps up, or split them into smaller rectangles.
6. Sort incoming finger messages by `(zoneIdx, fingerIdx)`, using the same axes for touches and drawn pixels (no flip).
7. Send `EndFingerDataStreaming` when your host disconnects.

21.9 D.9 Plugin Surface

The plugin surface lets a host application (a plugin, a patch, a script) draw on the **whole** Erae surface and receive every touch on it, without asking the user to build a layout with an API zone. Erae keeps the surface itself, next to the project's layouts:

- Up to **4 sessions** can be open at once, one per plugin instance. Each session draws into its own full-screen buffer (42 x 24).
- Exactly one source is on the LEDs at a time: the user's layout, or one session.
- Only the session on screen receives touches. A hidden session can keep drawing; its latest frame appears at once when it is shown.
- The user always comes first: a layout button (N1 to N8) or Return leaves the plugin surface at once. On Erae Touch, Home and Alt do the same.
- A session that goes silent for 2 seconds is closed and the user's layout comes back, so a crashed plugin never leaves a frozen picture behind.

The plugin surface lives in RAM only. It is never saved, it is not part of a project, and switching projects does not close sessions. It uses the same SysEx service as the API zones (sub-services `30` to `33` and `40` to `43`), on the same `Erae 2 MIDI` port. API zones placed in layouts, `StartFingerDataStreaming` and `EndFingerDataStreaming` keep working exactly as described above, so a layout with a full-screen API zone of index 0 still works.

> **Detecting support.** Send `Open` and wait for an answer. A device that does not answer within 500 ms has no plugin surface. `VersionRequest` still reports `0x02`.

21.9.1 D.9.1 Session lifecycle

1. **Open** a session with your name and prefix. The reply gives you a session ID (7-bit), the surface size and the number of sessions the device supports.
2. **Draw** into the session's buffer with the session draw commands. While the session is hidden nothing changes on the LEDs.
3. **Show** the session. If the user is on a layout, it takes the screen at once and you get `Focus on`. From then on every touch on the surface comes to you.
4. Keep the session alive: **any message** from the session (a draw, a Show, a Heartbeat) renews its 2 second lease. When you have nothing to draw, send a **Heartbeat** every 500 ms.
5. **Close** the session when your plugin closes.

Session IDs are handed out in rotation (`00`, `01` up to `7F`, then `00` again), so an ID is not reused until many other sessions have been opened. Draws to an ID that is not open are ignored.

21.9.2 D.9.2 When Show takes the screen

Situation	What happens
The user is on a layout, no finger on the surface for 1 s	The session takes the screen at once: `Focus on, show`
A finger is on the surface, or was lifted less than 1 s ago	The Show **waits** until 1 s after the last finger lifts. Nothing ever switches under a finger
Several Shows wait	The most recent one wins; the one it replaces gets `Focus off, superseded`
The user presses a layout button, Return, previews a layout from Erae Lab or previews an Erae Sound preset	Any waiting Show is cancelled
The device shows Mapping, Scale, a sequencer page, the settings or a Lab preview	The Show is refused with `Focus off, busy`. It is not queued: show again later
The same session already sent a Show less than 2 s ago	The extra Show is ignored
Another session is on screen	It gets `Focus off, show` and yours takes its place

21.9.3 D.9.3 Losing focus

Your session loses the screen when the user presses a layout button or Return, opens a non-layout screen, previews a layout or preset, or when another session is shown. Before the `Focus off` reply, the device sends a finger **Release** (action `02`) for every finger of yours still on the surface, so your note or gesture tracking always ends cleanly. Those fingers are then ignored until they lift.

While the plugin surface is on screen, the layout's sequencer, looper and clock keep running exactly as when switching layouts. No layout button is lit at full brightness: the button of the layout that will come back glows dim. A layout pushed from Erae Lab meanwhile is stored and the plugin surface stays on screen.

A session closes when you send `Close`, when its lease runs out (2 s without a message), when the USB cable is disconnected (all sessions at once) or when the device restarts. If it was on screen, the user's last layout comes back.

21.9.4 D.9.4 Host -> Device commands

<header> is the same as in D.4. <s> is the session ID.

Sub	Command	Payload
`30`	Open	`<len> <name> <prefix...>`: `len` is `01` to `18` (1 to 24), the name is `len` printable ASCII characters (`20` to `7E`), the prefix is 0 to 16 bytes of 7-bit data copied at the head of every reply to this session
`31`	Show	`<s>`
`32`	Heartbeat	`<s> <prefix...>`: the same prefix as in `Open`. No reply while the session lives
`33`	Close	`<s>`
`40`	Session Clear	`<s>`
`41`	Session SetPixel	`<s> x y r g b`, exactly as D.4.5
`42`	Session DrawRectangle	`<s> x y w h r g b`, exactly as D.4.6

Sub	Command	Payload
43	Session DrawImage	`<s> x y w h <bitized pixels> <chksum>`, exactly as D.4.7

The session draw commands use the API zone payloads byte for byte, with the session ID in place of the zone index: 7-bit colours shifted left by one on the device, bitize-7 images with the XOR checksum, clipping at the surface edges, and drawing coordinates with `y = 0` at the **bottom** row, as for API zones and the finger stream (D.3.1). Porting a host from an API zone to a session changes the zone index byte only.

21.9.5 D.9.5 Device -> Host messages

Replies use the usual framing `F0 <prefix> 7F <code> ... F7`, next to the zone boundary reply (`01`) and the version reply (`02`):

Code	Message	Body
`03`	Opened	`<s> <width 2A> <height 18> <maxSessions> <lease in 100 ms units, 14>`
`04`	Refused	`<reason>`: `01` full (all sessions in use), `02` bad name, `03` busy (firmware update running)
`05`	Focus	`<s> <00 off, 01 on> <reason>`: `01` show, `02` user, `03` superseded, `04` busy, `05` preview
`06`	Closed	`<s> <reason>`: `01` closed, `02` expired, `03` unknown

- `Refused, bad name` is sent with an empty prefix (`F0 7F 04 02 F7`), since the prefix follows the name.
- A Heartbeat for a session the device does not know (it restarted, or the lease ran out) or with a different prefix gets `Closed, unknown`, framed with the Heartbeat's prefix. Open a new session and show it again only if the user asks for it.
- **Finger stream**: the D.5.3 body, unchanged, with the session ID in the `zoneIdx` byte, framed with the session's prefix and sent only to the session on screen. It does not need `StartFin-`

`gerDataStreaming`. Positions cover the whole surface, with `y = 0` at the **bottom** row as for API zones (D.3.1).

Parse tolerantly: ignore codes you do not know and bytes after the fields you expect. Later firmware may add codes and append fields, never change the existing ones.

21.9.6 D.9.6 Worked examples

Message	Bytes
Open “Synth” with prefix `7D 12 34 56`	`F0 00 21 50 00 01 00 02 01 01 04 30 05 53 79 6E 74 68 7D 12 34 56 F7`
Opened, session `03`	`F0 7D 12 34 56 7F 03 03 2A 18 04 14 F7`
Show session `03`	`F0 00 21 50 00 01 00 02 01 01 04 31 03 F7`
Focus on, session `03`	`F0 7D 12 34 56 7F 05 03 01 01 F7`
Session `03`: pixel `(3, 5)` red	`F0 00 21 50 00 01 00 02 01 01 04 41 03 03 05 7F 00 00 F7`
Session `03`: 2x1 image `[red, green]` at `(0, 0)`	`F0 00 21 50 00 01 00 02 01 01 04 43 03 00 00 02 01 44 7F 00 00 00 7F 00 44 F7`
Heartbeat session `03`	`F0 00 21 50 00 01 00 02 01 01 04 32 03 7D 12 34 56 F7`
Close session `03`	`F0 00 21 50 00 01 00 02 01 01 04 33 03 F7`

> **Reference client.** A Python client and an example game, Erae Rally (`mido`), ship as one folder, the plugin surface SDK (`tools/erae_plugin_surface/sdk/` in the firmware repository). The client keeps a shadow frame and sends only the changed rectangle, at most 30 times per second, heartbeats on its own thread, and reopens its session (hidden) after `Closed`.