ATOMATRIX 0PERAT0R

USER MANUAL

ATOMATRIX, part of ATOMOSYNTH
Copyright 2026 ATOMOSYNTH

Table of Contents

Before You Begin **6**
Safety notes 6
Warranty 6
Attribution 6

1 Introduction **7**

2 Hardware Overview **8**
2.1 Panel 8
2.2 Power 8
2.3 MIDI IO jacks 8
2.4 A/B switches 9
2.5 Display 9
2.6 SELECT and DATA 9
2.7 B button 9

3 Quickstart **10**
3.1 Home screen grid 10
3.2 Moving around 11
3.3 Making a MIDI thru box 11
3.4 Making a MIDI merge box 12
3.5 Routing insight 12
3.6 Never go out of scale 13

4 Getting Around **15**
4.1 Home screen 15
4.1.1 Cell activity 15
4.1.2 Routing view 16
4.1.3 Using the home screen 16
4.2 Status bar 17
4.3 Knobs and gestures 17
4.3.1 Input maps 18
4.4 Screen navigation 18
4.5 Screen reference 18
4.5.1 Menus 18
4.5.2 Multiple selection 19
4.5.3 Range selection 20
4.5.4 Name editor 20
4.5.5 Alerts 21

5 Routing **22**
5.1 Routing model 22
5.1.1 Overview 22

5.1.2 Message filtering reference . . . 22
5.1.3 Clock routing . . . 23
5.1.4 Merging multiple inputs . . . 23
5.2 Routing settings persistence . . . 23
5.3 Routing configuration example . . . 23
5.4 Output menu reference . . . 25
5.4.1 Output configuration screens reference . . . 25

6 Clock and Sync . . . 28
6.1 How clock is handled . . . 28
6.2 Master Clock . . . 28
6.3 Mode Clock . . . 29
6.4 Configuration screen reference . . . 29

7 Presets and settings persistence . . . 31
7.1 Device settings . . . 31
7.2 Modes . . . 31
7.3 Presets . . . 31
7.3.1 Saving presets . . . 31
7.3.2 Recalling presets . . . 32
7.3.3 Preset naming and deletion . . . 32
7.3.4 Preset screen reference . . . 32

8 MIDI Monitor . . . 33
8.1 Reading a line . . . 33
8.2 Clock and transport . . . 33
8.3 Freezing and scrolling . . . 33
8.4 Monitoring specific ports . . . 33
8.5 Screen reference . . . 34
8.5.1 MIDI Monitor . . . 34
8.5.2 Port abbreviations . . . 34
8.5.3 Message types and values . . . 34
8.5.4 Ports . . . 35

9 Modes . . . 36
9.1 Global and output modes . . . 36
9.2 Mode slots and pool size . . . 37
9.3 Loading a mode . . . 37
9.4 Editing parameters . . . 37
9.4.1 File parameters . . . 38
9.5 Modes with custom interfaces . . . 38
9.6 Unloading a mode . . . 38
9.7 High latency protection . . . 38
9.8 Authoring modes . . . 38
9.9 Screen reference . . . 39
9.9.1 Mode chain . . . 39

9.9.2 Mode browser . . . 39
9.9.3 Parameters page . . . 39
9.9.4 File picker . . . 40

10 Bundled Modes Reference . . . 41

10.1 Global modes reference . . . 41
10.1.1 Generative Sequencer . . . 41
10.1.2 Master Clock . . . 41
10.1.3 Note Probability . . . 41
10.1.4 User Scale . . . 41
10.2 Output modes reference . . . 42
10.2.1 Arpeggiator . . . 42
10.2.2 CC LFO . . . 42
10.2.3 Custom Scale Tuner . . . 42
10.2.4 Delay . . . 43
10.2.5 Euclidean Sequencer . . . 43
10.2.6 MIDI Player . . . 44
10.2.7 Quantizer . . . 44
10.2.8 User Scale . . . 44
10.2.9 Velocity Remap . . . 45

11 USB Modes . . . 46

11.1 Change USB mode . . . 46
11.2 Client . . . 46
11.3 Drive . . . 46
11.4 Screen reference . . . 47
11.4.1 The USB Mode screen . . . 47
11.4.2 Drive Mode screen . . . 47

12 Settings . . . 48

12.1 USB Mode . . . 48
12.2 Acceleration . . . 48
12.3 Input Map . . . 48
12.4 Status Bar . . . 48
12.5 Dim Timeout . . . 49
12.6 Developer Mode . . . 49
12.7 Reinitialize All . . . 49

13 Developer Mode . . . 50

13.1 Enabling Developer Mode . . . 50
13.2 Log Monitor . . . 50
13.2.1 Reading a line . . . 50
13.2.2 Scrolling and freezing . . . 51
13.2.3 Log levels filtering . . . 51
13.3 Mode Timing . . . 51
13.3.1 Reading the figure . . . 52
13.4 Turning Developer Mode off . . . 52

14 Updating Firmware 53
14.1 Update files 53
14.2 Flashing mode 53
14.3 Flashing the file 53
14.4 If something goes wrong 54

A MIDI Implementation Chart 55

B Specifications 56

C Troubleshooting 58

D Glossary 61

Before You Begin

Safety notes

Operator takes its power from the USB-C port. Connecting the cable starts the device, and disconnecting it shuts the device down.

Any standard USB power supply runs Operator. A supply that holds a steady voltage keeps the display and the timing steady with it.

Keep the device dry and use it in normal room conditions. It is built for the same temperature and humidity a person is comfortable in, so keep drinks away from the panel and away from the jacks.

Disconnect the USB cable before cleaning. Wipe the outside with a dry cloth. A non-conductive cleaning liquid such as isopropyl alcohol works for marks a dry cloth leaves behind, applied sparingly to the cloth rather than to the device.

Warranty

If Operator reaches you with a malfunction or a fabrication defect, let us know as soon as you notice it. Every case is handled individually.

Attribution

Operator is made by ATOMATRIX, part of ATOMOSYNTH.

Copyright 2026 ATOMOSYNTH.

1. Introduction

Thank you for choosing Operator. In your hands there's a device with many faces and endless possibilities. This manual describes what it does and how, but it is you who will give it its right place in your setup.

The best way to describe Operator is as a MIDI processor. MIDI arrives, the device decides what happens to it, and it leaves. It can route MIDI in many ways, which covers what every MIDI box does, but Operator is way more than that.

It has four inputs and eight outputs, and carries MIDI over the USB-C port in both directions as well, so MIDI can arrive in five places and leave from nine, in any combination. Put it in the middle of a setup and the small boxes that used to live there stop being necessary, and nothing gets rewired when you change your mind.

Playing with MIDI routes is where it starts to get interesting. Every output decides for itself what it accepts, so one synth takes the notes from your keyboard while another takes only the modulation wheel and the sustain pedal, and a third takes the bottom two octaves alone. One controller plays several instruments at once, each on its own slice of the keys, and suddenly you have a multitimbral sound none of them allowed you to perform on its own.

That is how Operator thinks: it works from the output end. Each output keeps its own list of the inputs it listens to, so routing means opening an output and choosing what it accepts. Once that idea lands, the menus need no explanation, because nearly every screen on the device is an output or something an output owns.

A dedicated MIDI processor handles everything moving through, in real time and doing nothing else. Messages flow faster than MIDI itself can carry them, so the device stays invisible to whatever is connected to it, and a tightly timed, convoluted sequence never drifts.

Then there are modes, which are the reason Operator is not just a very good router. A mode is a small program that runs inside that processor, in the path a message takes between arriving and leaving, and it can shape what passes or invent something new. Your device already comes with a bunch of them: a master clock, a generative and a euclidean sequencer, an arpeggiator, a delay, a CC LFO, a quantizer, a user scale with multiple presets, a velocity remapper, a note probability filter, a microtonal scale tuner and a MIDI file player. Stack up to four of them in a chain for sounds and textures even you haven't heard before.

Modes are not proprietary or factory-defined. The authoring SDK is open source, so a mode can be written from scratch, altered until it does what you meant, or taken from whoever wrote it.

Operator is a USB MIDI class compliant device, so a computer, a tablet or a phone recognizes it as soon as it is plugged in, with no drivers to install. It doubles as an interface too: connected to a computer, the instruments on its jacks become playable and recordable from your DAW, useful if your interface has no MIDI ports.

We hope you enjoy it as much as we do, and can't wait to hear what lengths you take the device to.

2. Hardware Overview

2.1 Panel

On the panel you will find the following:

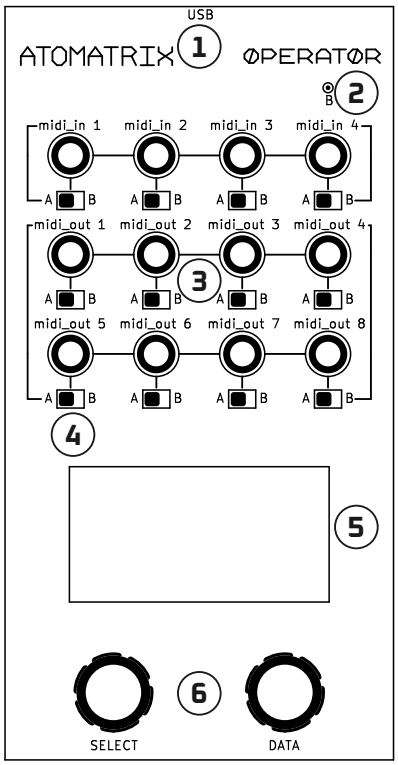

Figure 2.1: The Operator panel.

1. **USB-C port.** Used to power the device and can also be connected to a class compliant device like a computer.
2. **B button.** Used to put the device into flashing mode.
3. **MIDI IO jacks.** The top row is the inputs and the two rows under it are the outputs.
4. **A/B switches.** One next to every jack, each selecting the TRS MIDI type of its own jack.
5. **Display.** To see and set up the device.
6. **SELECT and DATA knobs.** Used to navigate and edit parameters.

2.2 Power

Operator takes its power from the USB-C port, which can be connected to a computer or a USB power supply. The screen coming to life is the sign that the device is powered. To power the device off, just unplug the cable.

The same port also carries MIDI to and from external software, and can be used as a drive for file transfer. Both are covered in Chapter 11.

2.3 MIDI IO jacks

Twelve 3.5mm TRS jacks drive the MIDI input and output ports in three rows of four, numbered left to right. The top row is for inputs, and the rest are all outputs. The device calls these jacks Input 1-4 and Output 1-8.

The USB-C port carries MIDI as well, so it joins the jacks wherever the device lists what it can route from and to. The device calls that pair Input USB and Output USB, and both appear while it is in a USB mode that carries MIDI. What the device does with the whole set is covered in Chapter 5.

2.4 A/B switches

Operator was designed to be as compact and portable as possible, but the industry did not agree on a uniform way to wire MIDI TRS jacks over the years, so there are currently two types of MIDI jacks: A and B. The MIDI Manufacturers Association settled on Type A in 2018, and gear built since then is usually (but not always) Type A.

This is why we included a small A/B switch next to every jack on the device, making the Operator any-manufacturer TRS MIDI capable.

To find out which type any device expects, you will have to look in its manual. The two wirings are identical from the outside, so that is the only reliable answer. Even 3.5mm TRS to five-pin DIN plug adapters have a specific type you will have to match with the jack you plug them into.

The difference between A and B is purely electrical, so a quick way to check if you've set the right type is to look at the home screen and send some MIDI through the device. What arrives at the Operator is either valid MIDI or nothing at all, so if you see any activity on the device, you are good to go.

2.5 Display

The OLED display switches itself off after a set time with no activity on the knobs, and comes straight back when either knob is turned or pressed. The device keeps routing and processing MIDI the whole time nonetheless, and the inactivity delay is set in Chapter 12.

2.6 SELECT and DATA

Both knobs can turn or be pressed to interact with the Operator. Turning SELECT moves through a list. Turning DATA changes the value the cursor sits on. What the knobs do when pressed is a choice covered in Chapter 12. Other gestures are taught in Chapter 4.

2.7 B button

A small hole marked B sits at the top right of the panel. When pressed with a straightened paper clip or a pin during power on, it puts the device into flashing mode, so firmware can be replaced. The update procedure is covered in Chapter 14.

3. Quickstart

Out of the box, the Operator has a passthrough routing configuration. Input 1 goes to Output 1, Input 2 to Output 2, and so on. This means that Outputs 1-4 pass all messages from their respective Inputs. Outputs 5 to 8 and Output USB are switched off.

3.1 Home screen grid

The home screen is a grid. Each row is an input, and each column is an output. Initially, every intersection represents one input into one output. There are two possible grid visualizations, and **turning SELECT** switches between them.

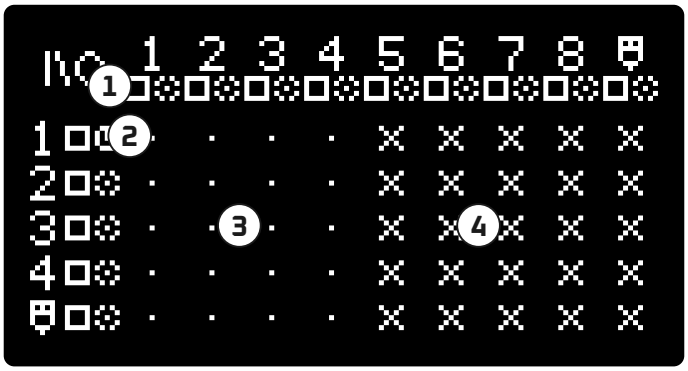

Figure 3.1: Home screen, on its activity view

Activity view shows MIDI as it flows through the device at any given moment:

1. **IO activity indicator.** Briefly fills when MIDI moves through an input or output. If the input square remains unfilled, nothing is either going in or coming out, so it is a quick way to tell a cable problem from a routing problem.
2. **Clock icon.** On an input row it marks the one input chosen as clock master. On an output column it becomes a full clock icon when clock data is being emitted.
3. **Grid dots.** The output is switched on and nothing has crossed this input-output pair recently. Once traffic moves, the dot briefly becomes a symbol for the most recent kind of message being routed.
4. **Grid crosses.** That whole output is disabled.

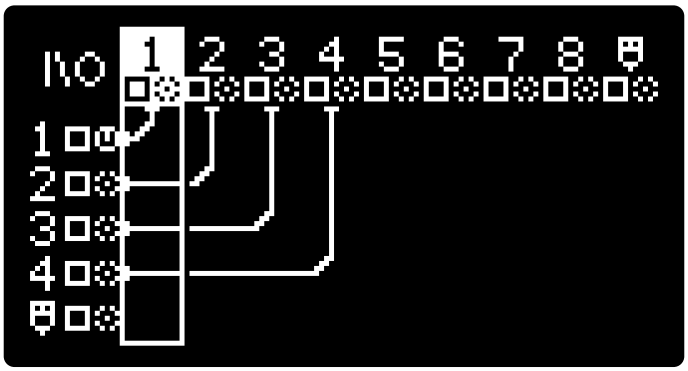

Figure 3.2: Home screen, on its routing view, with the cursor over Output 1

Routing view draws the current routing configuration as a wire diagram. Every input that reaches an output runs a line to it, making the whole arrangement legible at a glance.

3.2 Moving around

You can move around the home screen by:

- **Turning SELECT** to switch views.
- **Turning DATA** to enable a cursor across the output columns. The cursor clears itself after a few seconds of inactivity.
- **Clicking DATA** to open the configuration menu for an output under the cursor.
- **Clicking SELECT** to open the main menu.

By default, **clicking** DATA goes into whatever is selected and **clicking SELECT** goes back out. That is the factory Input Map, which can be changed as described in Chapter 4.

3.3 Making a MIDI thru box

A thru box takes one source and feeds it to several destinations. On the Operator, that means routing Input 1 into more than one output.

Turn DATA until the cursor sits over column 2, then **click DATA**.

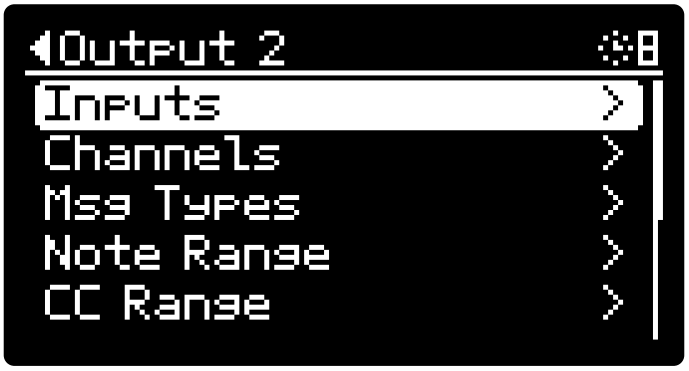

Figure 3.3: Output configuration menu

Here you can set everything an output does. You define a set of inputs first, then define what passes through it using the other menu entries: Channels, Msg Types, Note Range, CC Range and more. Inputs is already selected, so **click DATA** again.

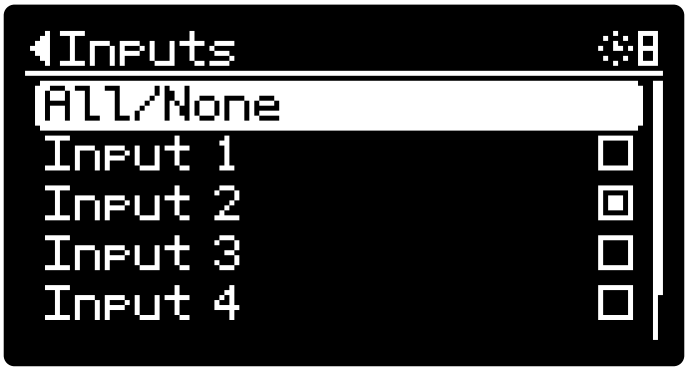

Figure 3.4: Input list for Output 2, which takes Input 2 by default

You will see all the available inputs this output can route from. **Turn SELECT** down to Input 1 and **click DATA** to check it, then to Input 2 and **click DATA** to clear it. Output 2 now takes the same source Output 1 does.

Click SELECT twice to go back all the way to the home screen, then do the same procedure on column 3. Switch to routing view and Input 1 now runs out to three outputs.

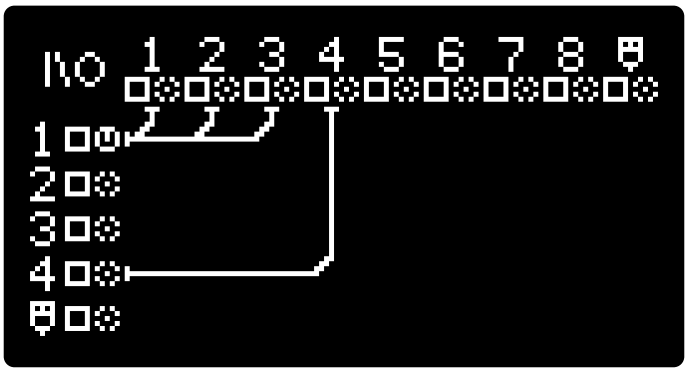

Figure 3.5: Routing view with Input 1 feeding Outputs 1, 2 and 3

3.4 Making a MIDI merge box

A merge box does the opposite of a thru box: several sources feed one destination. This means putting more than one input on the same output.

Put the cursor on column 1, **click** into it, **click** into Inputs, and **check** Input 2 and Input 3 alongside Input 1. Then **clear** the default input on Outputs 2 and 3, the same way you did on the way out of the thru box, so those two carry nothing. Anything arriving on any of the three inputs now leaves Output 1, merged into one stream.

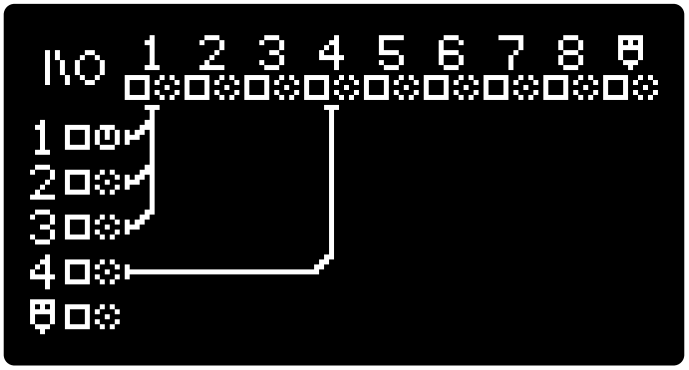

Figure 3.6: Routing view with Output 1 taking Inputs 1, 2 and 3

3.5 Routing insight

Every possible routing configuration has the same principles:

- An output owns a list of inputs.
- Several inputs on one output merge into it.
- One input on several outputs is copied to each of them.

Any other output configuration refines over these principles and the rest of this manual goes into detail on how they work.

3.6 Never go out of scale

Thanks to modes, all sorts of MIDI effects and generators can also be used to infinitely enhance your setup. A mode is a small program the device loads from its storage, and it comes in one of two kinds: a global mode works on everything arriving at the device, while an output mode belongs to a single output and works only on what leaves through it.

User Scale is one of the output modes the device ships with. It snaps every note to a scale, so whatever reaches an output running it comes out in key.

Put the cursor on column 1 and **click DATA**, then **turn SELECT** down to Modes and **click DATA** again. Every output has four mode slots and they run in order. **Click DATA** on the first one.

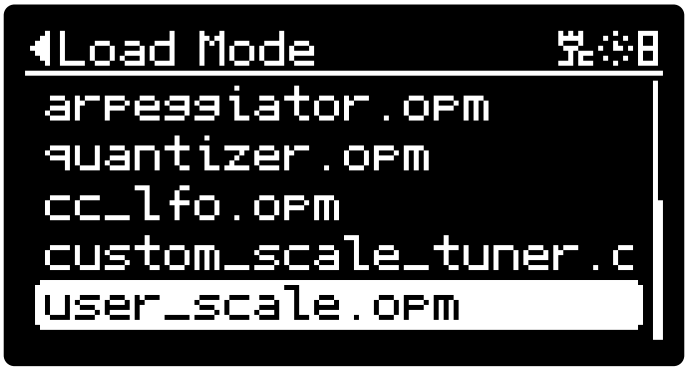

Figure 3.7: The browser an output slot opens, listing the output modes

Turn SELECT down to user_scale and **click DATA** to load it. The browser closes on its own, leaving the name in the slot. **Click DATA** on that slot to enter the mode.

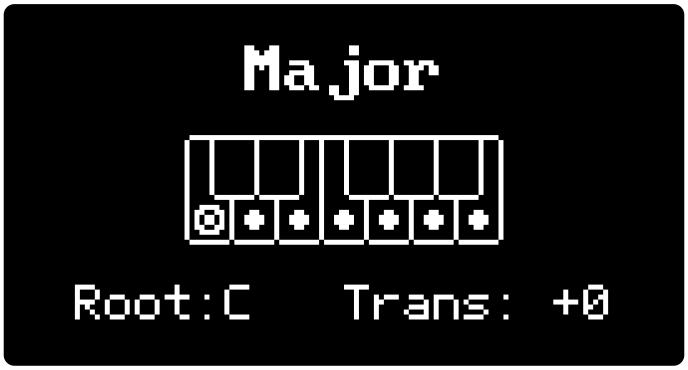

Figure 3.8: User Scale custom interface

Some modes draw a custom interface, like this one: the scale name, a one-octave keyboard marking the notes that are in the scale, and the root / transposition values below. You can also **use both SELECT and DATA** knobs to move things around on this interface.

From this view, a parameters list can also be accessed by **long-pressing DATA**. Likewise, you can go back to the modes list by **long-pressing SELECT** while in the mode's interface.

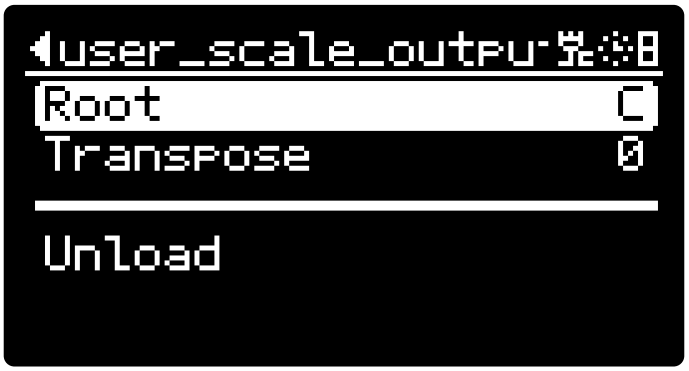

Figure 3.9: User Scale parameters, with Unload on its last row

Root is the note the scale starts on and Transpose shifts the result in semitones. Unload sits under the divider line and, as its name suggests, unloads the current mode from the output.

Until you unload it, Output 1 stays in key through a power cycle as well, because the device remembers what each chain is carrying. The rest of what modes can do is in Chapter 9.

4. Getting Around

With the display and the two knobs, everything on the device can be set, and almost every screen is navigated and operated the same way.

4.1 Home screen

The home screen is where the device starts and where it returns to. It is a grid of every input against every output, so each cell (an intersection) is one input into one output.

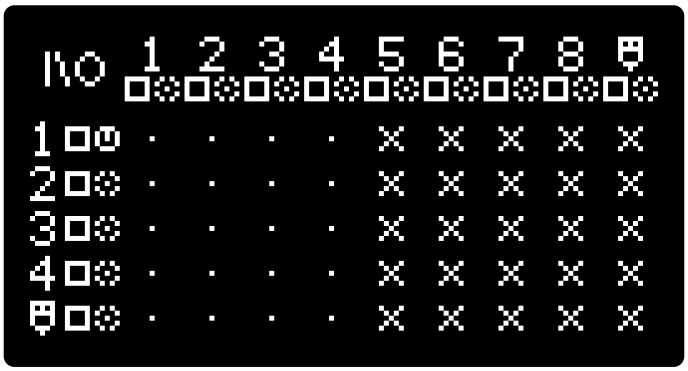

Figure 4.1: Home screen, activity view

Rows are inputs and columns are outputs. Input 1 to Input 4 are always present. Input USB and Output USB join them while USB MIDI is running, drawn with a USB connector symbol instead of a number.

Each input row carries three things: its number, a small square activity indicator that fills while MIDI arrives on that input, and a clock marker. The clock marker on an input row does not report traffic, but marks the one input chosen as the device clock master, which is covered in Chapter 6.

Likewise, each output column header carries its number, the activity indicator, and a clock marker that closes into a full circle while clock is going out of it.

4.1.1 Cell activity

A cell (the point where an input and an output intersect) reports the traffic from that input, showing one of the following:

Icon	What it reports
	The resting state. The output is switched on and nothing has crossed this pair recently.
	A note. This one holds for as long as the note is held rather than fading on its own.
	A control change.
	A program change.
	Pitch bend.

Icon	What it reports
	Aftertouch.
	Start, stop or continue.
	A system exclusive message.
	Any other system message.
	The output is switched off, so the cross fills its whole column.
	The pair runs over USB and is routed, but the USB connection is not live.

Clock is not drawn in cells, because the clock marker at the head of each output column reports it instead.

4.1.2 Routing view

Turning SELECT anywhere on the home screen switches to a diagram of the routing state on the device.

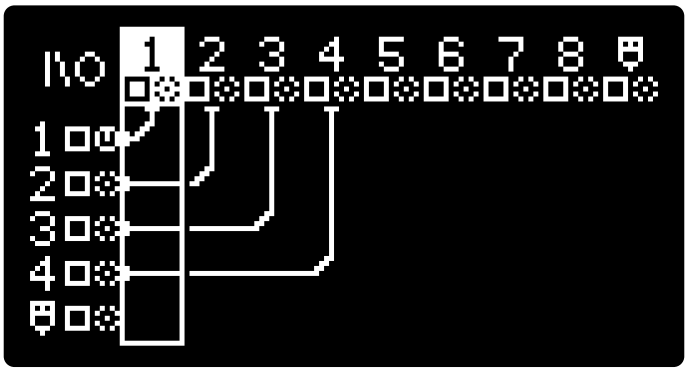

Figure 4.2: Home screen, routing view, with active cursor

Every output that carries at least one input shows a line from the top of the grid down to the lowest input row it takes, and every input that reaches at least one output runs a line out to the rightmost output it feeds. An output that carries inputs but is switched off draws its line as a dotted one.

4.1.3 Using the home screen

Four gestures work on the home screen:

- **Turning SELECT** switches views.
- **Turning DATA** moves a cursor across the output columns. The first turn always lands on column 1. The cursor clears itself after a few seconds.
- **Pressing DATA** opens the configuration for the output under the cursor.
- **Pressing SELECT** opens the main menu.

On every other screen, **pressing SELECT** goes back. The home screen is where all navigation starts, so this is the only place where going back moves you forward into the menu.

4.2 Status bar

Every menu screen carries a status bar:

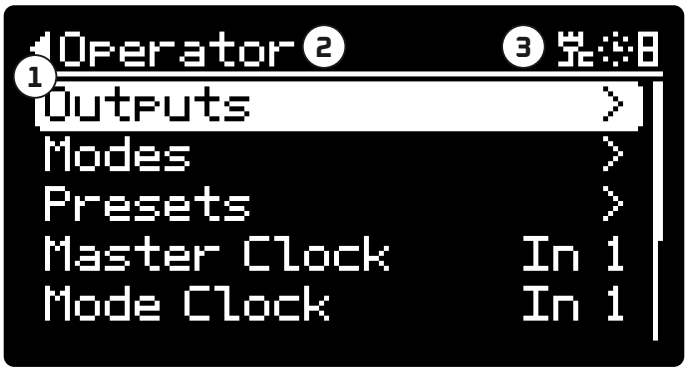

Figure 4.3: Status bar over the main menu

1. **Back arrow.** Present whenever another screen sits underneath the current one.
2. **Title.** The name of the current screen.
3. **Indicators.** A set of icons, drawn from left to right in the order listed below:

Icon	Meaning
	USB MIDI is running as a client.
	MIDI clock is arriving on an input.
	No clock is arriving on any input.
	No MIDI is moving.
	MIDI is arriving on an input.
	MIDI is leaving on an output.
	MIDI is arriving and leaving at once.

4.3 Knobs and gestures

The knobs are labeled SELECT and DATA. Depending on the current setting, turning them will produce a more accentuated response. Acceleration is defined in the Settings menu.

Behavior on the two knobs can also be defined in the device settings. Turning or clicking the knobs produces one of six **gestures**, and every screen in this manual is described in terms of those six:

- **Scroll** moves a cursor through a list or along a value.
- **Change** edits whatever the cursor is on.
- **Enter** activates the selected item, which may open a screen, toggle a box or fire an action.

- **Back** leaves the current screen and returns to the one underneath, and can also cancel the current action.
- **Parameters** is a long press, used only when opening the parameter page of a mode that has taken over the display.
- **Exit** is the other long press, reserved for leaving that display again.

A click counts as a short press when the knob is released within half a second. Held past that, the long press fires while the knob is still down, and the release that follows does nothing.

4.3.1 Input maps

In the Settings page, Input Map lets you pick between two choices, described in the table below. Turning is identical under both. SELECT always scrolls and DATA always changes a value under either map. Only the press roles swap.

Knob and action	Default	Organic
SELECT, turn	Scroll	Scroll
DATA, turn	Change	Change
SELECT, short press	Back	Enter
DATA, short press	Enter	Back
SELECT, long press	Exit	Parameters
DATA, long press	Parameters	Exit

4.4 Screen navigation

Screens sit on a **stack**, on top of each other. **Enter** puts a new one on top and **Back** takes it off again, which is what the arrow at the left of the status bar reports.

Values are edited in place, as they are turned. Leaving a screen with **Back** does not cancel edits. The one exception is a confirmation alert, where **Back** dismisses it without carrying out the action.

4.5 Screen reference

The screens below are the patterns the whole device is built from, so interacting with them is always the same no matter which screen you're on.

4.5.1 Menus

The most common pattern. Most lists of items in the device have this shape.

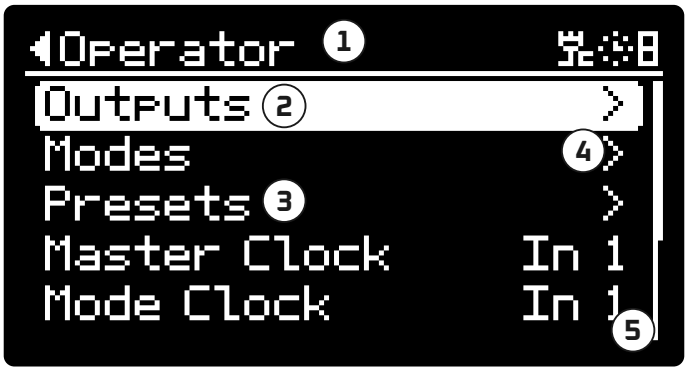

Figure 4.4: Main menu, cursor on the first row

1. **Status bar.** Explained above, shows the current view and device status.
2. **Selected row.** Shows the currently selected item with a filled bar.
3. **Row label.** A label that doesn't fit the screen scrolls while its row is selected.
4. **Row marker.** Shows whether the row opens another screen, carries a switch, or holds a value. A row with no marker fires an action.
5. **Scrollbar.** Shows if there's more content in the current screen and the current relative position inside it.

Scroll walks the rows. **Enter** acts upon the selected row: it opens a screen, flips a box, or fires an action. **Change** edits a value row, cycling a choice or stepping a number. **Back** leaves the current screen.

> **Quick edit on numeric values**
>
> **Enter** on a number row underlines its rightmost digit, and **Change** moves that digit. Each further **Enter** steps the underline one place to the left, so a large value is covered in tens or hundreds.

4.5.2 Multiple selection

Used when multiple things can be picked at once.

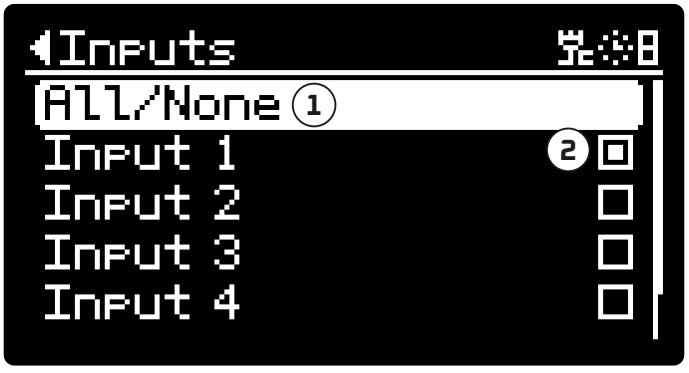

Figure 4.5: Input list for Output 1

1. **All/None row.** Sets every box at once, and clears them when all boxes are set.
2. **Toggle box.** Filled when the item is picked and empty when it is not.

Scroll walks the rows. **Enter** flips the box on the selected row. **Change** also sets it, clockwise to pick and counter-clockwise to clear. **Back** leaves.

4.5.3 Range selection

Used for certain values that require a low-high numeric pair.

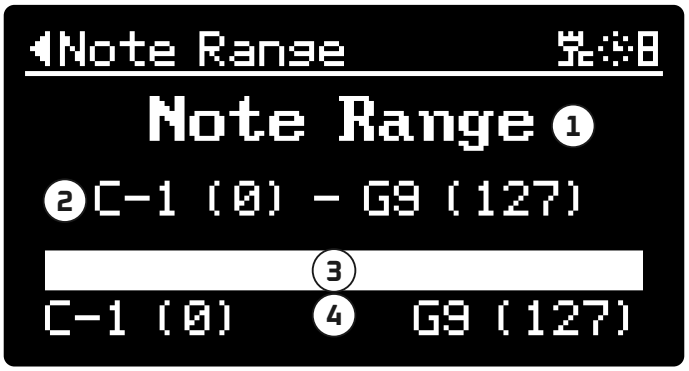

Figure 4.6: Note range for Output 1

1. **Title.** The value you are editing.
2. **Current range.** The current range being set.
3. **Bar.** A visual representation of the current range over its minimum and maximum.
4. **Range limits.** Lowest and highest possible values.

Turning the **left** knob moves the minimum value, and the **right** knob moves the maximum value. **Back** leaves the editor.

4.5.4 Name editor

Used in certain places that allow editing names.

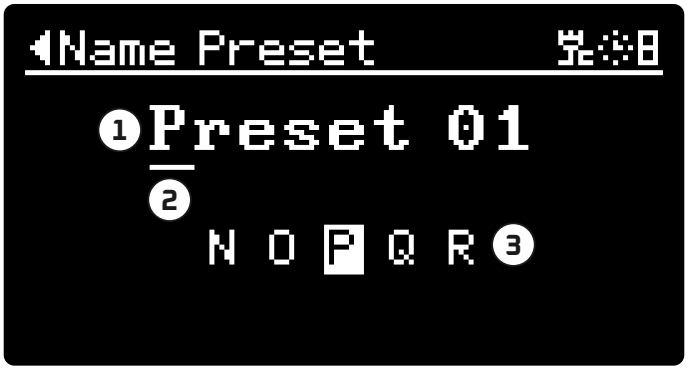

Figure 4.7: Preset naming screen

1. **Name.** Text being edited.
2. **Cursor.** The current character being edited.
3. **Character strip.** Lets you pick the character under the cursor.

Scroll moves the cursor and **Change** cycles the character under it. Moving past the last character lets you add more characters using **Change** when the cursor is after the current text. Either **Back** or **Enter** saves the current text.

Removing characters

Long-pressing DATA removes the character under the cursor.

4.5.5 Alerts

Actions that require confirmation will present an alert for you to either confirm or cancel them. Sometimes, it will be used without options to briefly inform you about something.

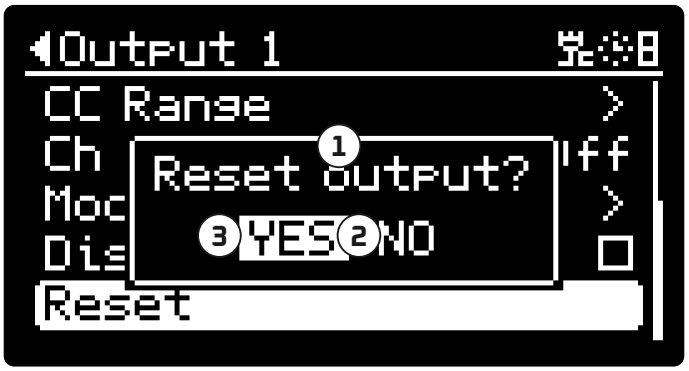

Figure 4.8: A confirmation over the output it would act on

1. **Text.** Lets you know what you need to confirm.
2. **Options.** Available options, when the alert carries any.
3. **Selected option.** The one **Enter** carries out.

Scroll lets you pick a button. **Enter** carries out the selected one. **Back** dismisses the alert without doing anything.

5. Routing

At its core, the Operator has a powerful routing engine that you can use to move MIDI around in any way you like.

5.1 Routing model

5.1.1 Overview

Routes are defined around outputs. Each physical output on the Operator decides which inputs it takes MIDI from. Before emitting anything, an output passes its incoming messages through a series of filters and then through the modes it has loaded. Modes are explained in detail in Chapter 9.

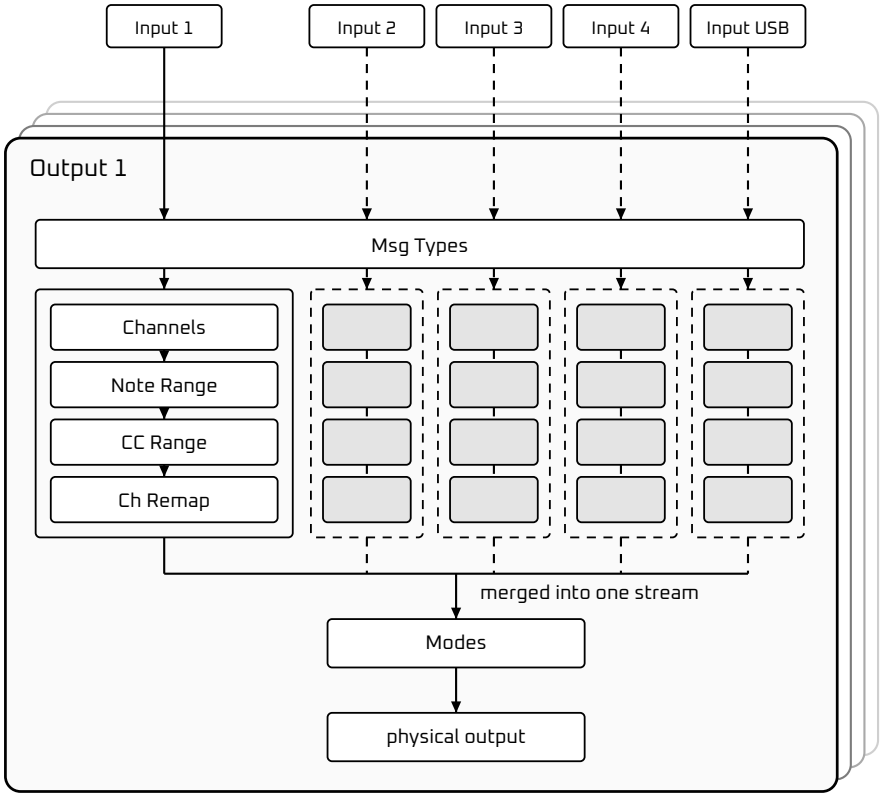

Figure 5.1: Output 1 default message flow where only Input 1 is enabled. Every output sets its inputs independently, parallel to each other.

In the diagram above, Output 1 has Input 1 enabled. All incoming messages are first filtered by their type (notes, CC and so on), after which four additional filters refine what reaches the next step.

Every input has its own stack of four filters, each with its own configuration. The four stacks drawn greyed and dashed are the ones Output 1 does not take in this configuration. Any incoming MIDI message that fails the current configuration for a given output is dropped.

5.1.2 Message filtering reference

An arriving message is checked against the following filters in order:

1. **Msg Types** asks what kind of message this is. Setting this affects every incoming message regardless of its input.
2. **Channels** asks which channel it arrived on. Notes, control changes, program changes, pitch bend and aftertouch carry a channel. Clock, transport, system exclusive and the system messages do not.
3. **Note Range** checks whether a note falls inside a specific range.
4. **CC Range** checks whether a CC falls inside a specific range.
5. **Ch Remap** is not a filter but a message rewrite. Every message that carries a channel leaves on the channel this row names.

A message that fails any of the checks above is dropped from the output.

5.1.3 Clock routing

Filtering clock messages works slightly differently from the rest. All outputs receive clock messages from the input set in the **Master Clock** setting in the Operator's main menu, so enabling or disabling the **Clock** message type affects the incoming clock from the **Master Clock** instead of whatever inputs are currently selected for an output. More details on how clock is handled can be found in Chapter 6.

5.1.4 Merging multiple inputs

After the filters, every message that passes is merged into a single stream. That stream runs through the output's chain of modes, and what leaves the chain goes out of the physical output. The merge is a plain interleave. Every message that passes an input's filters is written to the output in the order Operator took it off the wire.

System exclusive is the only exception to message merging. A system exclusive message takes over an output until it ends. Any other input's system exclusive message is dropped, and everything else waits for the message to end before being routed again.

5.2 Routing settings persistence

Input filter configuration is persisted and not cleared if an output decides to unset any given input. When an output sets that input again, all of its filter settings come back exactly as they were. The Reset row in the Output settings menu is what returns every one of an output's filters to the factory state at once.

With one input assigned, Channels, Note Range, CC Range and Ch Remap open their editors straight away. With multiple inputs, each of those rows opens a list of the assigned inputs first, to let you set each input separately.

Msg Types is the only exception. There's no input-selection submenu for it, and a change made there is applied to every input the output currently holds.

5.3 Routing configuration example

We'll set the routing configuration for the following scenario: a controller, a drum machine and a synth. The controller's notes should play the synth, its knobs should drive the drum machine, and the drum machine should keep the synth in time.

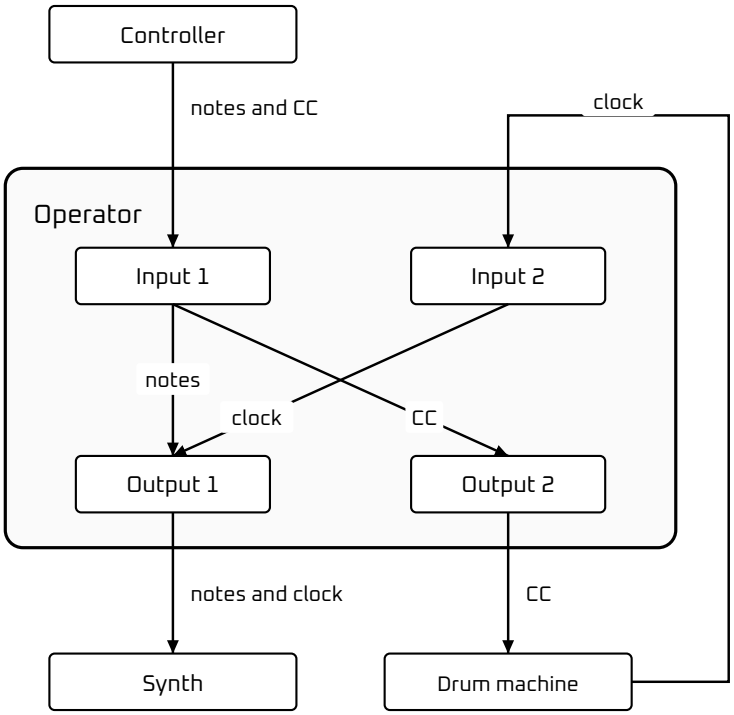

Figure 5.2: Example scenario where a MIDI controller drives two machines and one of them acts as a clock source

Connect the controller's MIDI output to Input 1, since everything it sends is routed on from there to both the synth and the drum machine. The drum machine is both played and drives the tempo for the whole setup, so its MIDI output goes to Input 2 and Output 2 comes back to its MIDI input. Output 1 goes to the synth.

Now to configure the routing:

1. From the home screen, **press SELECT** to open the main menu. **Scroll** to Master Clock and **Change** it to In 2. The drum machine is now the setup's master clock.
2. **Scroll** back to Outputs and **Enter** it, then **Enter** Output 1. **Enter** the Inputs menu and make sure only Input 1 is checked, clearing anything else with **Change**. Go **Back** to the Output 1 settings menu.
3. **Scroll** to Msg Types and **Enter**. Press **Enter** on All/None to clear any selection, then **Scroll** to Notes and **Change** to check it, doing the same for Clock. Now the synth receives the controller's notes and nothing else, plus the clock, which arrives from Input 2 even though Output 1 does not take that input. Go **Back** twice to get to the Outputs list.
4. **Enter** Output 2, then **Enter** Inputs. **Scroll** to Input 1 and **Change** to check it, then **Scroll** to Input 2 and **Change** to clear it, so only Input 1 is checked. Now the drum machine receives MIDI from the controller. Go **Back** to the Output 2 settings menu.
5. **Scroll** to Msg Types and **Enter**. Press **Enter** on All/None to clear all nine types, then **Scroll** to CC and **Change** to check it. Now the drum machine will only receive CC messages from Input 1.

> ⓘ **Note**
>
> If you are unsure what **Scroll**, **Change**, **Enter** and **Back** mean, they are set out in Section 4.3.1

The synth now plays what the controller plays, in time with the drum machine. The drum machine takes the controller's knobs and nothing else, because Output 2 passes only CC data, so the machine generating the clock never receives it back.

5.4 Output menu reference

Routing is set in each output's configuration screen. Entering Outputs in the main menu presents a list holding every output, and every row there opens the same screen, named for the output it was opened for. The home screen is the quicker way in: turn DATA to highlight an output column, then press DATA.

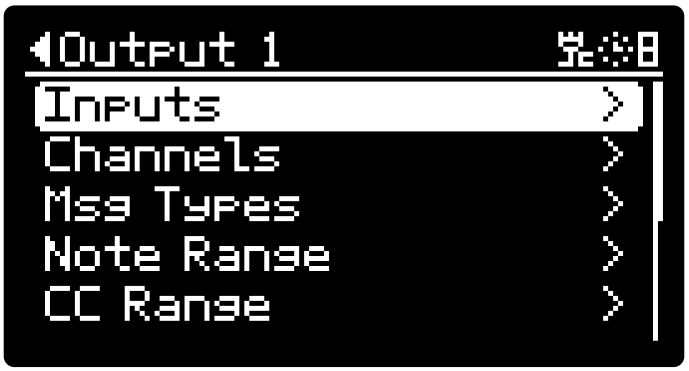

Figure 5.3: Output 1 configuration screen

If an output is disabled, it shows a Disable toggle and nothing else. Menu items appear and disappear as the configuration changes, always in the order below:

1. **Inputs.** Defines which inputs this output receives MIDI from.
2. **Channels.** Defines which MIDI channels pass, per input.
3. **Msg Types.** Defines the MIDI message types to allow.
4. **Note Range.** Passes notes only if they fall inside a specific range.
5. **CC Range.** Similar to Note Range, defines a range of allowed CC messages.
6. **Ch Remap.** Rewrites the channel of every message that carries one to the channel set here.
7. **Clock Source.** Defines an input to receive clock messages from. It only appears when there is no Master Clock set, the output takes more than one input, and Clock is checked in its Msg Types. More details on how clock is handled can be seen in Chapter 6.
8. **Modes.** The chain of modes this output runs. Chains are covered in Chapter 9. Factory-provided modes are covered in Chapter 10.
9. **Disable.** Turns off the output completely. A disabled output receives nothing at all, and its configuration screen shows this row alone to switch the output back on.
10. **Reset.** Returns this output to its factory state: input assignments, filters for every input, channel remap and clock source.

5.4.1 Output configuration screens reference

Every screen these rows open follows one of the patterns set out in Chapter 4.

Inputs, Channels and Msg Types are all multiple selection screens, with All/None always in their first row.

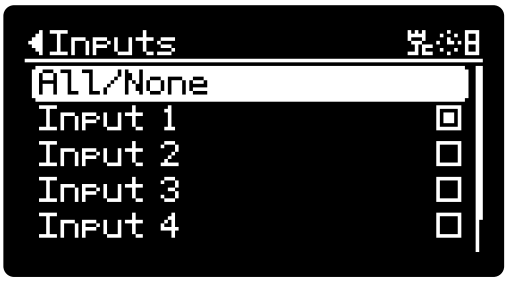

Figure 5.4: Inputs selection screen

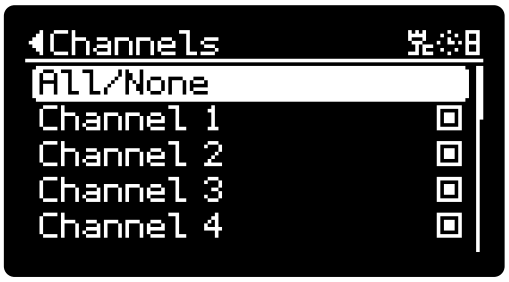

Figure 5.5: Channels selection screen

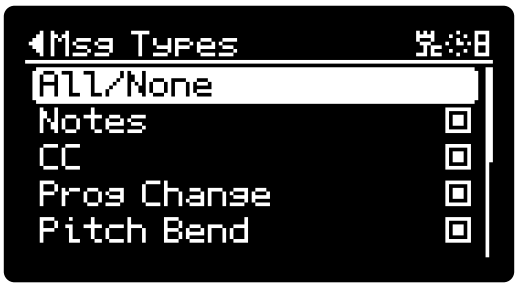

Figure 5.6: Msg Types selection screen

Note Range and CC Range are range selection screens. Full range is the factory default for both, which reads C-1 (0) to G9 (127) for notes and 0 to 127 for CC.

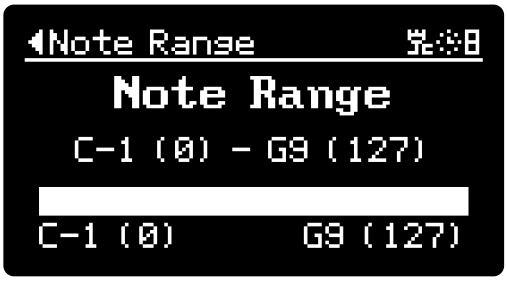

Figure 5.7: Note Range screen

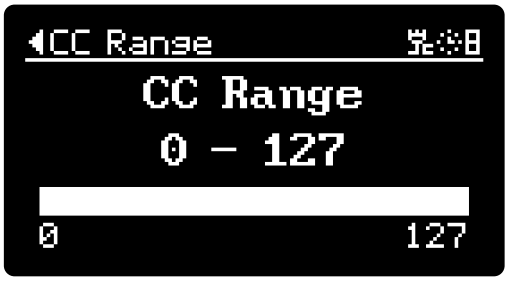

Figure 5.8: CC Range screen

Ch Remap and Clock Source are edited in place rather than in a different screen.

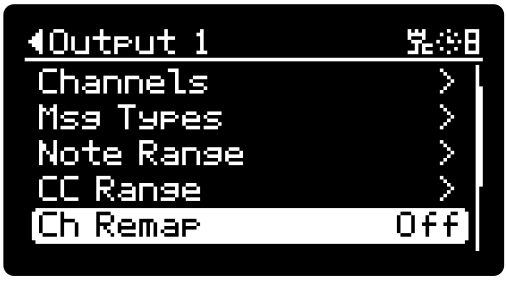

Figure 5.9: Ch Remap setting under an output configuration screen

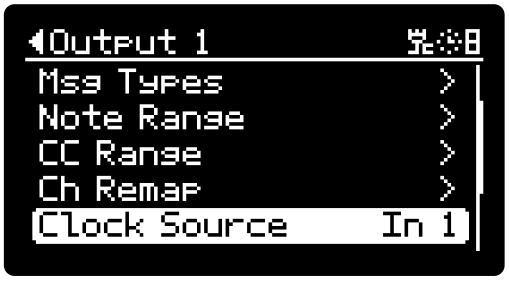

Figure 5.10: Clock Source setting under an output configuration screen

Disable switches the output off and back on. Reset asks for confirmation before reverting an output to its factory state.

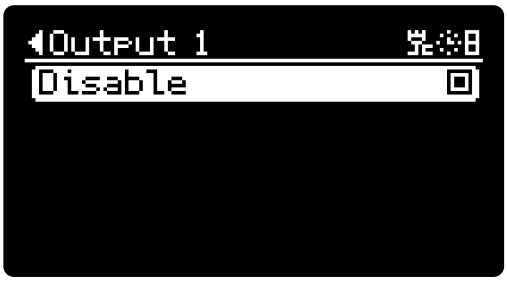

Figure 5.11: Disabled output configuration screen

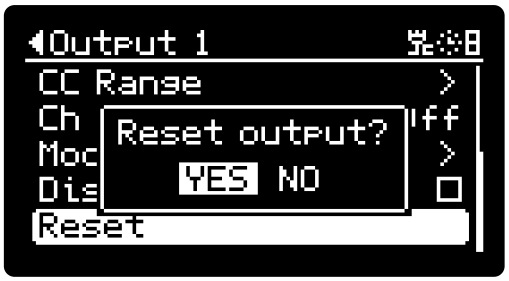

Figure 5.12: Reset confirmation alert

6. Clock and Sync

Three settings decide where Operator takes its clock from. Two of them are persisted along with presets and the other is saved with the global device settings.

Setting	Purpose	Persisted with
Master Clock	Input whose clock is routed to the outputs	Preset
Clock Source	The same choice, made one output at a time	Preset
Mode Clock	Input the modes take clock from	Device settings

6.1 How clock is handled

MIDI clock is a stream of individual pulses arriving at 24 per quarter note. The tempo is whatever rate those pulses arrive at, and a device that receives them follows the rate it is given.

Operator can receive two independent clock streams for different purposes: Master Clock is the input whose pulses are passed on to the outputs, and Mode Clock is the input whose clock the loaded modes follow.

Both reach every output. The Master Clock stream is checked against each output's Msg Types filter, which decides whether it passes. If allowed, it goes straight to the physical output. In contrast, Mode Clock is sent straight to each mode chain.

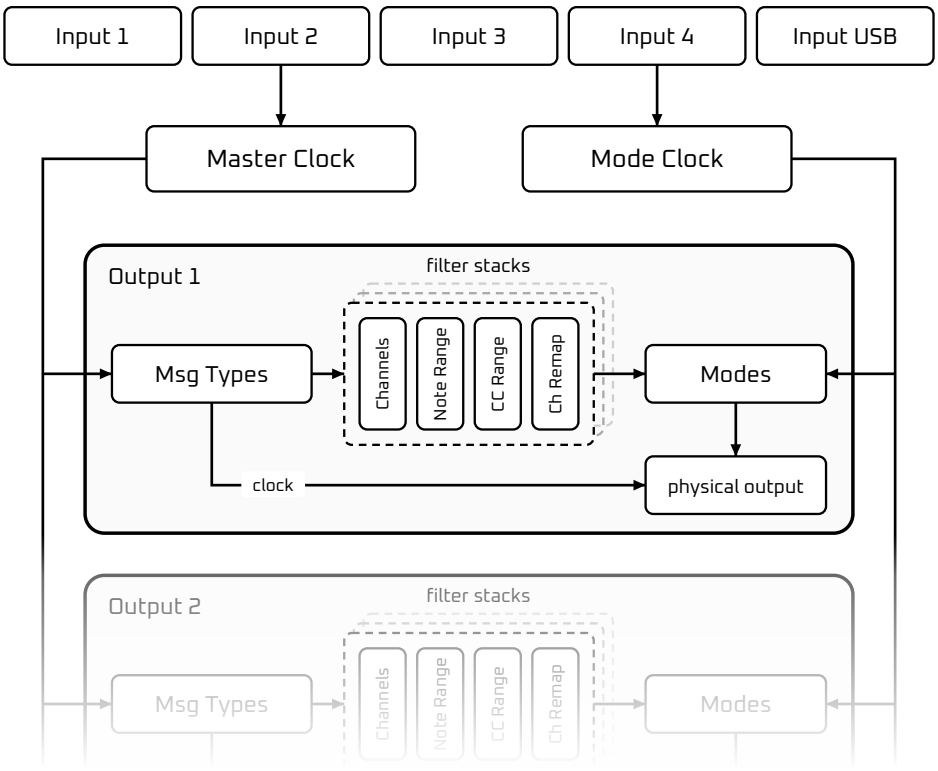

Figure 6.1: Clock routing diagram, showing where the Master Clock and Mode Clock streams go

6.2 Master Clock

The Master Clock setting is located in the main menu. To update its value, just **Change** it.

When an input is set as Master, the clock arriving there is distributed to the outputs. In each output, the clock stream is checked against Msg Types and dropped if the Clock message type is unchecked. If the check passes, the stream goes directly to the physical output. An output receives that clock whatever inputs its own routing configuration names.

If Master Clock is set to None, each output takes its clock stream independently. An output holding a single input uses that input's clock, and an output holding more than one uses the input its Clock Source setting names.

The Master Clock value is stored with the current routing configuration, so it is persisted and recalled with presets.

On the home screen, every input row carries a clock marker, and the marker on the Master Clock input row is drawn whole while the others are drawn broken. Each output column carries one too, drawn whole while a clock stream is leaving that output.

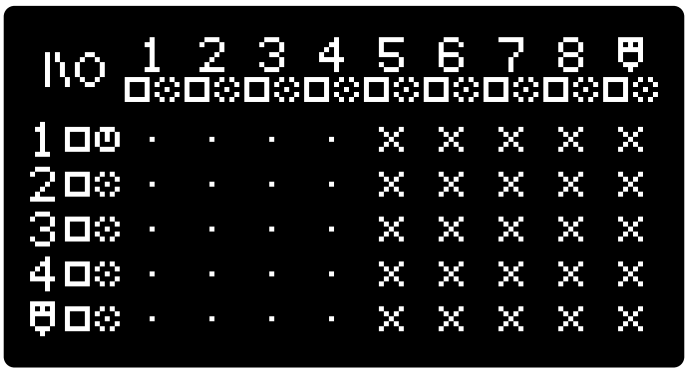

Figure 6.2: Home screen showing Input 1 as Master Clock and no clock stream being emitted

6.3 Mode Clock

Mode Clock is the main menu item directly below Master Clock. Like Master Clock, it is updated with **Change**.

The input selected as Mode Clock drives the pulse every loaded mode follows. This way the physical outputs stay independent of the modes that need a clock to work.

When Mode Clock is set to **Mode**, modes can become clock generators. Some modes are capable of generating clock and will do so as Mode Clock is set to **Mode**. When loaded, a clock-generating mode can provide clock messages that the outputs emit and will set the pace other modes follow. This also suppresses any clock generated by the input set in Master Clock.

Mode Clock is written into the device settings, making it device wide. Preset changes do not alter its value.

> **Start, Stop and Continue**
>
> Transport messages are different from clock messages, both by definition and in the way the device treats them. No clock stream stops because a transport message arrived. Some of the bundled modes do react to them, which is covered in Chapter 10.

6.4 Configuration screen reference

Master Clock and Mode Clock sit together in the main menu, below Presets. Clock Source is shown in each output's configuration screen.

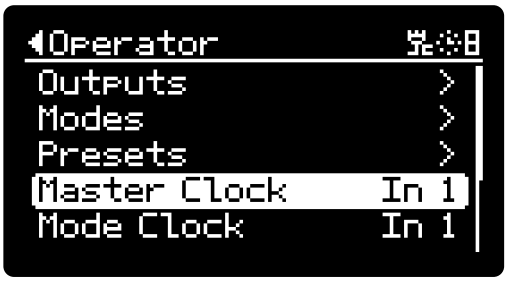

Figure 6.3: Master Clock and Mode Clock on the main menu

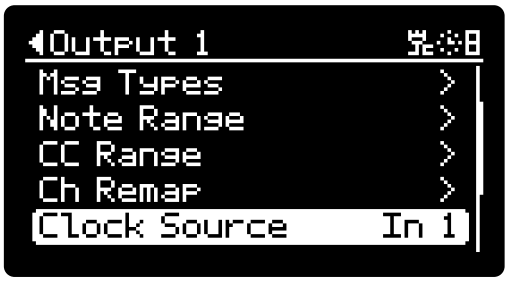

Figure 6.4: Clock Source setting in Output 1 configuration page

7. Presets and settings persistence

Operator saves its state in three separate ways:

1. Global device settings, written as you navigate the device.
2. The loaded modes list, written whenever a mode is loaded or unloaded.
3. Routing configuration, kept in presets and recalled on demand. Current routing state is kept separately from presets.

Presets are the only thing on the device with a save action. Everything else is saved during normal operation.

7.1 Device settings

Mode Clock, Acceleration, Input Map, Status Bar, Dim Timeout, Developer Mode and Mode Timing are the device settings that are saved. They are written when you leave the screen you changed them on. Mode Clock is covered in Chapter 6. The rest are covered in Chapter 12.

The current routing configuration is saved in the same way as these, allowing you to keep your device state even if you power your Operator off.

7.2 Modes

The list of currently loaded modes, both global and output ones, is written whenever it changes. Mode parameter values are also saved, written when you leave a mode's parameter page. All of these are restored when Operator is turned on. Mode chains are covered in Chapter 9.

A preset recall does not touch any of these. They remain independent of any mode configuration set in place.

7.3 Presets

Presets are stored copies of the routing configuration. They allow you to quickly recall and persist all of your routing settings. There are 16 preset slots. Presets store their name (up to 16 characters), the Master Clock setting, and the complete routing configuration for all nine outputs.

Preset slots are written only by the explicit Save Preset action. In the meantime, although the current routing state is persisted, recalling any preset overrides it.

7.3.1 Saving presets

The preset list is located in the main menu. **Entering** it shows the list of saved presets and the Save Preset action as the first row. To save the current routing configuration into a preset:

1. **Enter** over Save Preset. It changes to Save to..., showing that it is waiting for a target slot.
2. **Scroll** to a slot and **Enter** it.
3. If the slot is empty, the name editor opens to save it directly.
4. An occupied slot shows an overwrite alert. If confirmed, the save process continues by showing the name editor.
5. The name editor starts from Preset NN for an empty slot, or the existing name for an already-saved slot. Leaving the editor writes the name. The editor itself is covered in Chapter 4.

Enter Save Preset a second time to stop the saving action without writing anything.

7.3.2 Recalling presets

A preset recall replaces the current routing configuration completely. Whatever the outputs are set to at that moment is overwritten, and it is not kept anywhere unless it was saved into a slot first. If needed, save the current setup before recalling a preset, or assume that the current setup will go away.

With that understood, a recall is done in two steps. **Scroll** to an occupied slot and **Enter** it, then confirm the alert. Routing changes as soon as the dialog closes.

7.3.3 Preset naming and deletion

Names are set only at the moment of saving, which means renaming a preset is just saving it to the same slot. There is no separate rename action.

There is also no delete action, and no way to take a slot back to Empty from the device. A slot that has been written stays written until it is written over. Reinitialize All in Settings restores the live routing to its factory state and leaves every stored preset untouched.

Drive mode is the way out of that. It shows each stored preset on the drive that appears on your computer, where the copies can be kept, replaced or removed. Drive mode is covered in Chapter 11.

7.3.4 Preset screen reference

The screen shows one action row above sixteen slot rows. A slot shows its number and Empty until something is written to it.

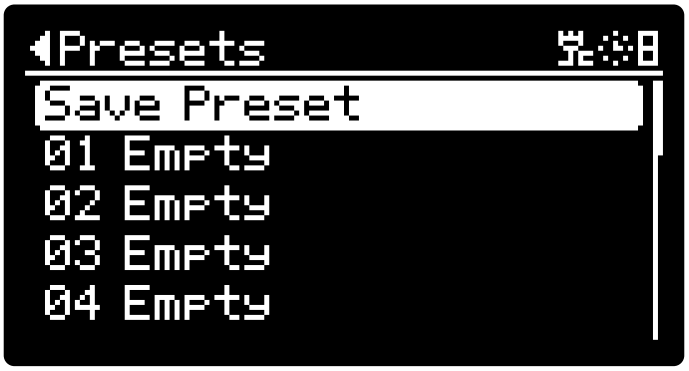

Figure 7.1: Presets screen with no stored presets

Once a slot is written, its row shows the name instead of Empty, and entering it displays the recall alert.

8. MIDI Monitor

The MIDI Monitor lets you inspect the MIDI streams passing through the device in real time. It is useful to check the source, destination and type of messages coming in and out of the Operator in an explicit manner.

The monitor is read only. Opening it, filtering it and pausing it change nothing about how MIDI is routed, filtered or merged, so it is safe to leave open while a set is running.

MIDI Monitor can be accessed from the main menu. **Scroll** to it and **Enter** it. The monitor opens empty. Messages will start appearing as they arrive or leave the Operator from that point onwards. By default, the monitor watches all five inputs and none of the outputs. The set of watched ports is chosen on the Ports screen, described later in this chapter.

8.1 Reading a line

Every line has the same short format:

```
>I1 NOn Ch1 C5 v100
>I1 NOf Ch1 C5 v0
>I1 CC  Ch1 #74 v64
>I1 PgC Ch1 #5
>I1 PBn Ch1 +2048
```

Reading one of those left to right:

- **Direction.** Shows `>` for a message arriving at an input, or `<` for a message leaving an output.
- **Port.** A two-character abbreviation for the port the message was seen on. `I1` is Input 1 and `O3` is Output 3. USB messages have the letter `U` rather than a number.
- **Message type.** A three-character abbreviation for the message type.
- **Channel.** `Ch1` to `Ch16`, printed for channel messages only.
- **Content.** What the message actually says, in the form that suits its type.

8.2 Clock and transport

Due to its nature (24 messages per beat), clock message streams do not appear in the monitor. Start, Continue and Stop messages do appear as `Srt`, `Cnt` and `Stp`, so a transport that is running can be confirmed here even though its tempo cannot. The clock settings themselves are covered in Chapter 6. The status bar carries a clock indicator, described in Chapter 4.

8.3 Freezing and scrolling

The list holds up to 32 messages. Once it is full, every new message drops the oldest line in the list. **Scroll** moves through the list.

To stop new messages appearing on the list, perform the **Parameters** gesture, which freezes the monitor and shows a pause icon on the top right of the screen while it is frozen. The same gesture releases it, making the monitor display new messages again. Independent of the pause, MIDI never stops flowing in or out. The gesture itself is named in Chapter 4.

8.4 Monitoring specific ports

Enter, from the monitor, opens the Ports screen. It picks which input and output ports the list shows messages from.

To watch one output and nothing else:

1. **Enter** on the monitor to open Ports.
2. **Scroll** to the All/None row at the top. **Enter** it once to check every port, then again to clear them all.
3. **Scroll** down to the output to watch and **Enter** it to check it.
4. Go **Back** to the monitor.

8.5 Screen reference

8.5.1 MIDI Monitor



Figure 8.1: MIDI Monitor screen with messages incoming

8.5.2 Port abbreviations

The two characters after the direction marker name the port. Inputs are prefixed with `I` and outputs with `O`.

8.5.3 Message types and values

Three characters name the message. Channel messages print a channel and a payload after the abbreviation.

Abbreviation	Message	Value
`NOf`	Note off	Note name and velocity. Note names run from `C-1` at note 0 to `G9` at 127, so middle C at note 60 reads `C4`
`NOn`	Note on	Note name and velocity. A note on with a velocity of zero is reported as `NOf`
`PAt`	Polyphonic aftertouch	Note name and the pressure on that one note
`CC`	Control change	Controller number and its value
`PgC`	Program change	Program number
`CAt`	Channel aftertouch	Pressure value for the channel
`PBn`	Pitch bend	A signed offset from center, `+0` at rest and `-8192` to `+8191` at the extremes
`SEx`	System exclusive	`F0` and the first two payload bytes, with the rest left off
`SPP`	Song position pointer	Nothing

Abbreviation	Message	Value
`SSl`	Song select	Nothing
`TRq`	Tune request	Nothing
`Srt`	Start	Nothing
`Cnt`	Continue	Nothing
`Stp`	Stop	Nothing
`ASn`	Active sensing	Nothing
`SRs`	System reset	Nothing

Two abbreviations are fallbacks rather than message types. `???` is a channel message whose status the monitor does not recognize, and `Sys` is a system message that has no abbreviation of its own, such as a MIDI time code quarter frame. Neither should appear on a healthy port, and a stream of either is worth following up on the cable and the sending device.

8.5.4 Ports

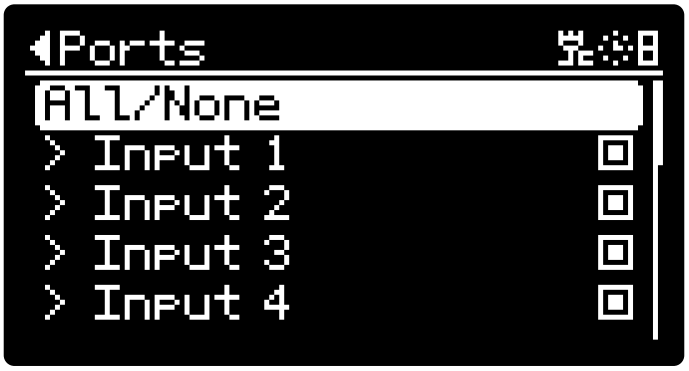

Figure 8.2: Monitor's port filter screen

The direction marker in each row is the same `>` and `<` the monitor prints in front of every line.

9. Modes

A mode is a small program the device loads from its storage and runs inside the MIDI path. Modes can modify the MIDI that flows through the device and even generate their own, allowing the Operator to perform infinite MIDI related things: clocks, arpeggiators, retuners, etc.

Modes, when loaded into a slot, can be configured from their parameters page or in custom interfaces for those that support them.

The Operator ships with 13 modes, each described in Chapter 10. Modes can also be added or updated at any time without flashing the device software, which is covered in Chapter 11.

9.1 Global and output modes

Modes have two categories: global and output. The difference is where in the MIDI path they sit.

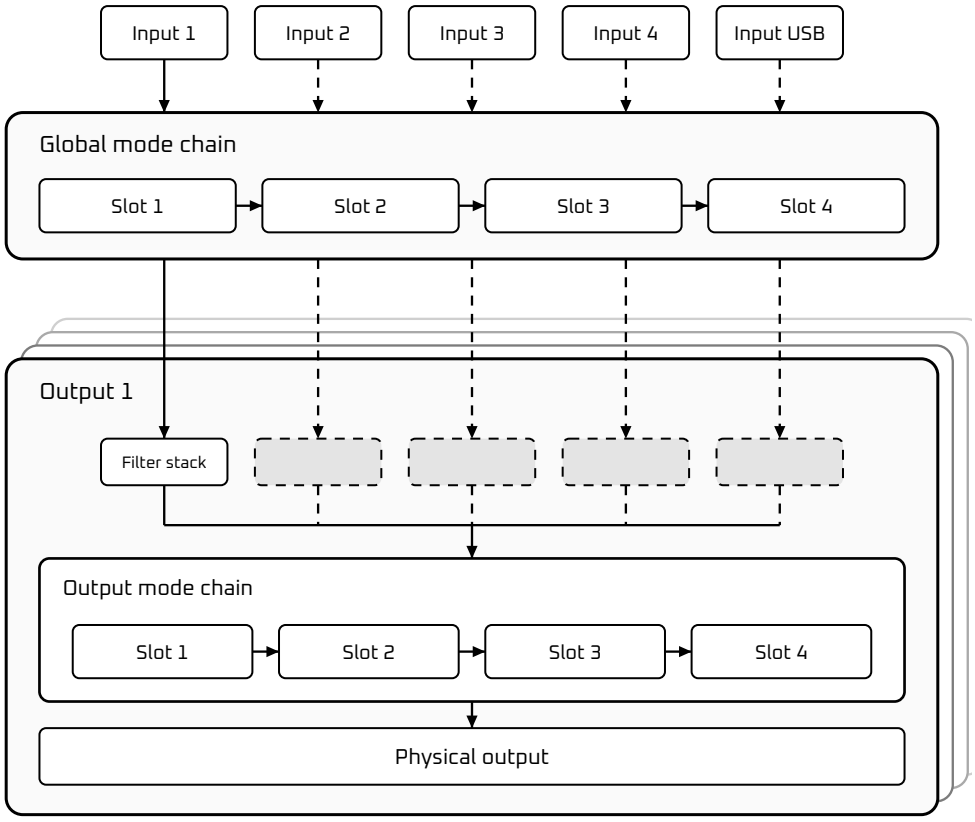

Figure 9.1: Global and output mode chain message path

A **global mode** runs before routing, on everything that arrived from all inputs at once. It sees the whole incoming MIDI stream and it can add messages to it, which is why clocks, sequencers and other generators are global modes. There is one global chain for the whole device, and it is reached by **Entering Modes** in the main menu.

An **output mode** runs after routing, and sees just the messages that passed the routing filters. Transposers, quantizers and velocity curves are typical output modes. Every output has its own chain, so the same mode can be loaded on multiple outputs and set up differently in each.

An output's chain is reached from the **Modes** menu inside an output's configuration screen, covered in Chapter 5.

Because global and output modes sit in different places, they are stored separately. The mode browser only shows mode files for the kind of chain it was opened from, so it is not possible to load the wrong kind into a slot by accident.

9.2 Mode slots and pool size

Each mode chain holds four slots. Modes run in the order given by their position in the chain, and each one hands its result to the next.

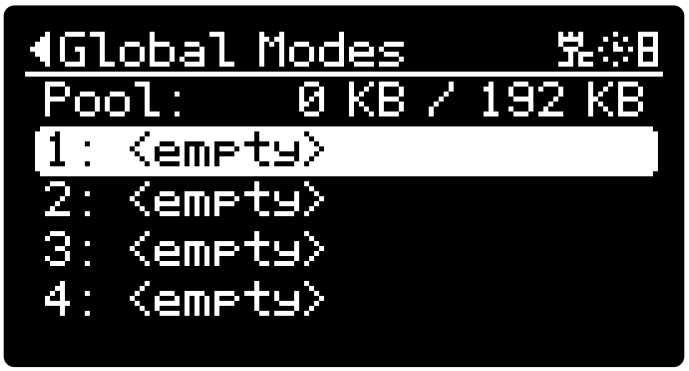

Figure 9.2: Global mode chain screen

Modes take a certain amount of RAM when they load, shown on the **Pool** indicator at the top of each mode chain. This pool is shared across the device, so modes can be loaded as long as there is space left in it. Modes usually take very little RAM, so the pool rarely runs out during normal operation. The device refuses to load a mode that takes more RAM than the pool has left.

9.3 Loading a mode

To load a mode into the global chain:

1. Open the main menu and **Enter Modes**. The four global slots appear.
2. **Scroll** to an empty slot and **Enter** it. The browser opens, listing the global modes installed on the device.
3. **Scroll** to the one to load and **Enter** it.

The browser closes and the chain screen comes back with the mode's name in the slot. Loading an output mode is the same process, starting from the **Modes** menu inside an output configuration screen.

If no modes are found on the device, the browser shows an empty list with a `(no modes)` message. Putting mode files onto the device is done over USB, covered in Chapter 11.

9.4 Editing parameters

Entering a loaded slot from the chain screen opens the parameters page directly, unless the mode has a custom interface, in which case that is what opens instead and the parameters page is reached with the **Parameters** gesture.

Every row is one parameter. **Scroll** moves between rows and **Change** edits the row the cursor is on. The parameters page uses the same patterns described in Section 4.5.

Values are saved on the way out. Leaving the parameters page writes the settings of that mode. These settings come back when it is loaded again, and some settings are saved independently per output.

9.4.1 File parameters

Some modes are used to read a file, a player reading a MIDI file being the obvious case. Those parameters show the chosen file name, or `(none)` when nothing is chosen. **Enter** opens a picker for the kind of file it needs, such as **Pick .mid**.

The picker shows the contents of a folder only the current mode has access to. Modes cannot access any other folder on the device. Choosing a file closes the picker and the mode loads it. An empty mode folder shows a single `[No files]` message. The location of the folders a mode can read is covered in Chapter 11.

9.5 Modes with custom interfaces

Some modes draw their own screen instead of a list of parameters: a sequencer showing its steps, a scale editor showing a keyboard, etc. Entering such a mode from its slot opens that interface.

In a custom interface, the mode owns the screen and both knobs, so what they do is up to the mode. The bundled modes have their interfaces described in Chapter 10. Leaving a mode's custom interface does not stop the mode.

Inside custom interfaces, the Operator recognizes two gestures to navigate in and out of the mode:

- **Exit** leaves the mode's custom interface and returns to the chain.
- **Parameters** opens the mode's parameters page. **Back**, from that page, returns to the custom interface.

Both are long presses, and which knob corresponds to each gesture is set in the Input Map setting described in Chapter 4.

9.6 Unloading a mode

Unload is the last item of the parameters page, below a divider that separates it from the parameters above. **Enter** on it to stop the mode, freeing its space in the pool and returning to the chain, where the slot reads `<empty>` again. The mode's saved settings are kept, so loading it again brings its parameters back as they were.

9.7 High latency protection

Modes run inside the real-time MIDI path and are given a time budget for their execution. A mode that overruns that budget repeatedly is switched off automatically and its slot is marked with `[!]` on the chain screen.

A slot in that state does nothing until it is unloaded and loaded again. If the same mode is switched off repeatedly, it is most likely faulty and must be reported to wherever it came from.

9.8 Authoring modes

Modes can be created by anyone, to share with friends or even to sell. They are built with the Operator SDK, which is an open source toolkit for programmers to author their own modes. Its documentation carries everything needed to write and distribute modes.

Read the Operator SDK documentation at https://xpktro.github.io/operator-sdk/

9.9 Screen reference

9.9.1 Mode chain



Figure 9.3: Mode chain screen with a clock generator in slot 1

9.9.2 Mode browser

The browser shows up to 32 modes. Beyond that a final `(truncated)` row appears for informative purposes only.

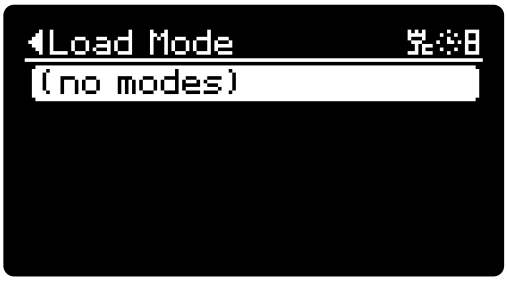

Figure 9.4: Mode browser screen without available modes

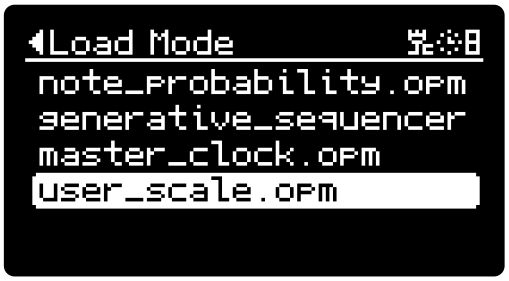

Figure 9.5: Mode browser screen with default global modes

9.9.3 Parameters page

A mode that has no parameters at all shows **Unload** only.

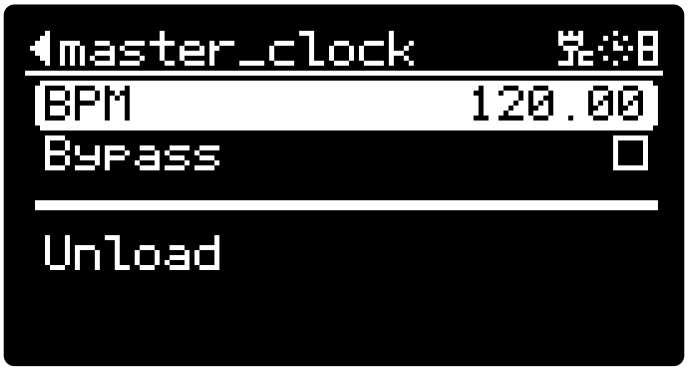

Figure 9.6: Master clock mode parameters page

9.9.4 File picker

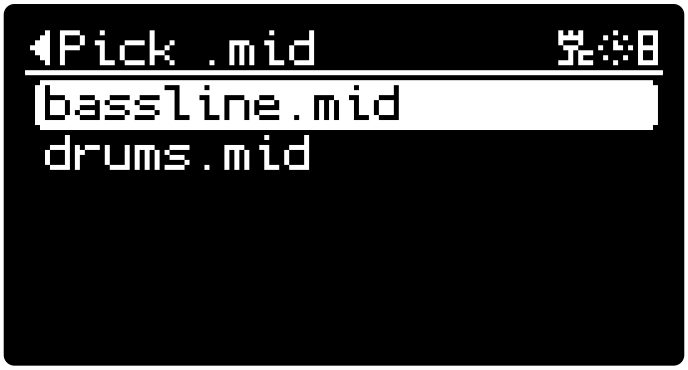

Figure 9.7: File picker screen for the MIDI player mode

10. Bundled Modes Reference

The Operator includes 13 modes out of the box. Of these, 4 are global modes and 9 are output modes. This chapter describes each one of them for reference. To know about how to load and manage modes refer to Chapter 9.

Each mode corresponds to one file named after the mode itself, so the arpeggiator is `arpeggiator.opm` and the quantizer is `quantizer.opm`. Six modes have a custom interface, the rest do not and present just their parameter pages instead.

10.1 Global modes reference

10.1.1 Generative Sequencer

The generative sequencer plays a stream of random notes in time with a clock. On every sixteenth-note step it rolls against **Probability**, and when the roll passes it picks a note at random from **Note Range**, holds it for a random length drawn from **Length Range**, and sends it to the chosen output.

Notes are released once their length has run out. They go out at a fixed velocity on channel 1, with up to 16 sounding at the same time.

The sequencer follows the device clock, so it stays in time with whatever is driving the tempo, and it stops for as long as that clock is stopped. **Internal Clock** makes the mode run on a clock of its own, at the tempo **Internal BPM** sets.

Output decides where the notes go. When set to **All**, they go round the outputs in turn and pass over any that are inactive. Choosing a single output sends every note there.

Note Range covers 0 to 127 and starts at 36 to 127, **Length Range** runs from 10 to 2000 ms and starts at 100 to 2000 ms, and **Probability** is a percentage that starts at 50.

10.1.2 Master Clock

Master Clock turns the Operator into a clock generator. It sends MIDI Timing Clock at the MIDI standard rate to all outputs and it can also drive the clock other modes follow. **BPM** covers 20.00 to 300.00 and starts at 120.00.

Bypass stops the clock. Turning bypass back off picks the clock up. Unloading the mode disarms the clock.

10.1.3 Note Probability

Note Probability drops notes at random as they pass through. **Probability** is the chance that any one note plays. At 100, its starting value, every note plays, and the lower it goes the more of them fall away.

Only the notes themselves are dropped. Note-offs pass through, and so does every other type of MIDI message.

10.1.4 User Scale

User Scale snaps incoming notes to a scale. A note already in the scale passes through, and a note outside it moves up to the next note in the scale.

This mode includes 23 scale presets and a custom one that can be freely edited. The mode's custom interface shows a keyboard, along with **Transpose** and **Root** controls. **Root** is the note the scale starts on. **Transpose** shifts the retuned note by up to three octaves either way, from −36 to +36 semitones.

In the mode's interface, a dot marks every note in the scale and a ring marks the key under the cursor. Editing a key while a preset is showing creates a custom scale with the current set of notes. The keyboard follows **Root**, so the lit keys are the notes that actually sound. Pressing **Back** flips it over to the shape of the scale itself, drawn from C.

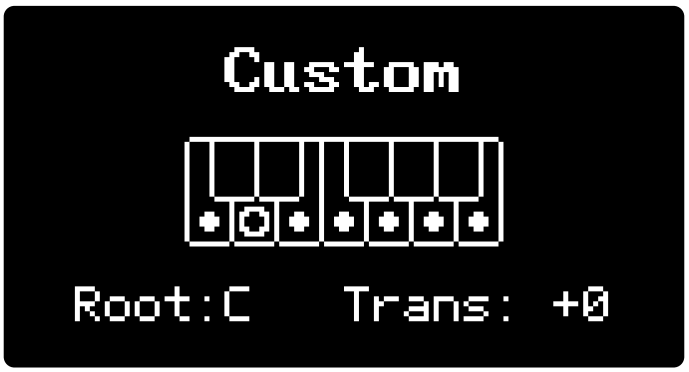

Figure 10.1: User Scale custom interface showing a custom scale

10.2 Output modes reference

10.2.1 Arpeggiator

The arpeggiator plays a held chord one note at a time, in a selected pattern, locked to the clock. Every other MIDI message passes untouched. Notes added or taken away while it runs fold into the pattern on the next step, and the arpeggio stops once the last note is released.

Style sets the order the held notes are played. **Rate** is how often a step fires, as a note division of the beat, and **Triplet** gives that division a triplet feel of three steps in the space of two. **Octave** shifts every played note by up to three octaves either way. **Fill** is how long each note sounds as a percentage of the step. **Chord**, when it is on, plays the whole chord at every step instead of one note, moved so the pattern note becomes the lowest voice while the shape of the chord is kept.

10.2.2 CC LFO

CC LFO is four independent low-frequency oscillators that emit control change messages.

Each one starts Off and turns on when its **Source** is set to **Free**, a fixed speed in Hz, or to **Sync**, locked to the incoming clock. The **CC** number to send is set by the parameter of the same name. **Shape** defines the pattern in which the CC values are sent. **Low** and **High** bound the values the pattern sweeps between.

When **Source** is set to **Free**, **Rate Hz** appears. It covers 0.01 to 20.00 Hz. When set to **Sync**, **Rate Beat** defines the oscillator speed relative to the current beat.

10.2.3 Custom Scale Tuner

Custom Scale Tuner retunes incoming notes to a microtonal scale supplied as a Scala file. Every note is mapped to the nearest degree of the scale and nudged to its exact pitch with a pitch-bend message, so this mode can play any non-standard tuning system on any synth that responds to bend.

Scale File takes a valid Scala file placed in `/extras/custom_scale_tuner/` on the device's USB drive. If no file is loaded or if it is unreadable, MIDI passes through unchanged.

> **Scala files**
>
> Read more about Scala files at https://www.huygens-fokker.org/scala/scl_format.html

Root Key is the key that anchors the base degree of the scale, and **Root Freq Hz** is the frequency that key is tuned to. **Bend Range** has to match the pitch-bend range the synth downstream is set to. Most synths have this set to 2 which is the default value for this parameter.

MPE, when it is on, spreads concurrent notes across channels 2 to 16, one note per channel, so each note can bend independently. This allows polyphonic retuning in synths that support the MPE standard.

10.2.4 Delay

Delay trails each note with a decaying echo. It repeats the note a few times, each repeat quieter than the last and optionally shifted in pitch.

Notes play as usual and then trail with up to **Repeats** echoes, spaced by the delay time and fading out by the **Feedback** amount. The echoes start on release.

Sync decides how the spacing is set. Set to **On**, it makes the echoes follow the clock at the division **Rate** names, with **Rhythm** applying a specific groove feel. When **Off**, the echoes run free at the **Time** in milliseconds.

Repeats sets the number of echoes after the played note. **Feedback** is how much quieter each echo is than the one before. **Transpose** adds a semitone shift to each successive echo. **Channel** picks which MIDI channel the delay works on.

10.2.5 Euclidean Sequencer

The Euclidean sequencer is a four-track note sequencer that builds its patterns from Euclidean rhythms. Each track divides the bar into a number of steps and spreads a number of hits as evenly as it can around them.

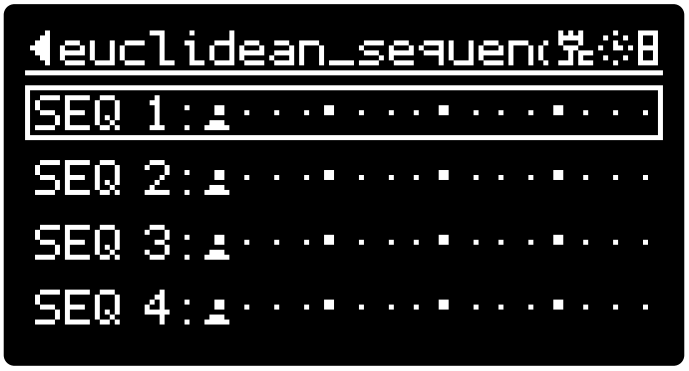

Figure 10.2: Euclidean sequencer main screen

A track spreads its **Events** hits across its **Steps** positions, so 3 hits over 8 steps gives the familiar `x . . x . . x .`, and **Offset** rotates that pattern around the bar without changing its shape. The four tracks are independent, so different step counts layer into polyrhythms.

It plays one note per track, the notes being 36, 38, 40 and 42, which are the bass drum, snare, electric snare and closed hi-hat of the General MIDI drum map.

The mode has a two-screen custom interface. The main screen shows all four tracks at once, with a playhead sweeping across as the bar plays and the selected row outlined. The detail

screen edits a single track and shows its trigger row alongside its **Steps**, **Events** and **Offset** values.

10.2.6 MIDI Player

MIDI Player plays Standard MIDI Files from the device. It passes incoming MIDI along untouched.



Figure 10.3: MIDI Player interface

MIDI File lets you pick a song placed in `/extras/midi_player/`. Single-track and multi-track files are both supported. **Clock Source** picks what tempo the song follows. On **File** it plays at its own, and on **Ext.** it follows the device clock.

10.2.7 Quantizer

Quantizer tightens up timing by snapping notes to a grid as they are played. A note that lands a little late is held back and let go on the next grid point.



Figure 10.4: Quantizer interface

Only the start of a note is moved. A note played just after a grid point is played at once, and a note played in the second half of the window waits for the grid point ahead of it.

Division sets how far apart the grid points are. **Triplet** moves the grid points closer together, three where two would go.

10.2.8 User Scale

Covered above in Section 10.1.4. The only difference is that this one works on a single output rather than on everything the device receives.

10.2.9 Velocity Remap

Velocity Remap reshapes note velocities. Every note's velocity is bent through a curve that is highly customizable using different parameters.

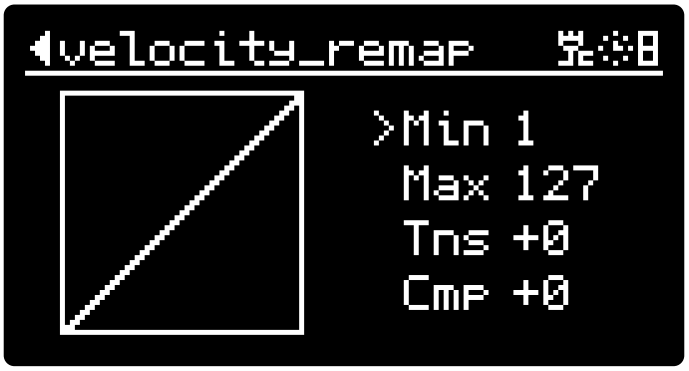

Figure 10.5: Velocity Remap interface

Minimum and **Maximum** are the softest and loudest the output can be. **Tension** bends the curve towards minimum or maximum. **Compand** opens and closes the value spread of the curve. **Constant** makes the curve totally horizontal, so every note has the same velocity.

11. USB Modes

The USB-C port can be used for more than powering the device. You can use the Operator as a MIDI interface and as a drive to store files that modes will use.

There are 2 USB modes: Client and Drive. In Client mode, the Operator becomes a USB MIDI class-compliant device so it can be connected to a computer, phone or tablet to act as a MIDI interface or processor. Drive mode lets you load modes or back up the current device configuration to your computer.

11.1 Change USB mode

To see and change the current USB mode, go to **Settings** in the main menu and pick **USB Mode**. To change the current mode just **Enter** it and a confirmation will prompt you to restart. Answering YES restarts the device straight away in the new USB mode.

11.2 Client

In Client mode Operator presents itself as a USB MIDI device named Operator, and it appears under that name in the MIDI port list of whatever it is plugged into. This mode carries MIDI in both directions and shows up on the Operator itself as two more ports:

- **Input USB** is MIDI arriving from outside.
- **Output USB** is MIDI leaving to the outside.

Two things on the home screen change when Client mode is running. The home screen grows a fifth input row and a ninth output column for the two USB ports, and the status bar carries a small USB indicator. Both disappear when no USB mode is selected.

The two USB ports are ordinary routing destinations and sources, and they start out unconnected. Output USB is disabled by default, so it needs to be enabled and routed to start working. Routing is covered in Chapter 5.

11.3 Drive

Drive mode exposes Operator's internal storage to the computer as a USB drive. It is how modes are installed and removed, how a mode's data files are loaded, and how the device state is backed up, edited and restored.

While Drive mode is running the display shows a lock screen. To go back into normal operation, either eject the drive from your computer or turn the Operator off and on again.

The volume that appears on the computer is named OPERATOR and carries a `README.txt` at its root, plus four folders:

Folder	What belongs in it
`/global/`	Global mode files, one `.opm` file per mode
`/output/`	Output mode files, one `.opm` file per mode
`/extras/`	Data files that modes read and write, kept in a folder named after the mode that owns them
`/device/`	The device state, saved presets and loaded modes, written by Operator and read back by it

`/global/` and `/output/` keep `.opm` files, which are the format of mode executables, and `/device/` keeps the three device file types: `.opd` for the device state, `.opl` for the loaded modes and `.opp` for a saved preset. Anything else in those folders, and any loose file at the root of the volume,

is removed the next time the device starts. `/extras/` is the exception and accepts any file type, because what a mode keeps there is the mode's business.

When entering Drive mode, Operator writes the following files to `/device/`:

- `state.opd` holds the current routing and device settings.
- `modes.opl` contains the current loaded modes list.
- `preset_01.opp` to `preset_16.opp` hold the saved presets, numbered the way the Presets screen numbers them. An empty slot has no file.

The folder works in both directions. When leaving Drive mode, Operator reads `/device/` and updates itself depending on the folder contents:

- An edited or previously archived file updates what it stands for.
- A deleted preset file empties that preset slot.
- A deleted `modes.opl` unloads every mode.
- A deleted `state.opd` returns the routing and the settings to their factory defaults.

A file Operator cannot read properly counts as deleted, so a damaged or half-copied file has the same effect as removing it.

Copying `/device/` somewhere safe is how a configuration is archived, and copying an archived set back before leaving Drive mode is how it is restored.

11.4 Screen reference

11.4.1 The USB Mode screen

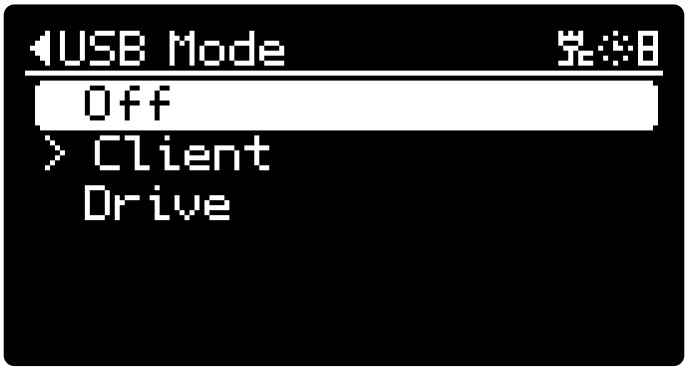

Figure 11.1: USB Mode screen. A marker appears next to the current mode

11.4.2 Drive Mode screen



Figure 11.2: The Drive lock screen

12. Settings

Settings cover the USB mode, how the knobs behave, where the status bar sits, when the display powers down, the diagnostic tools, and a reinitialization action.

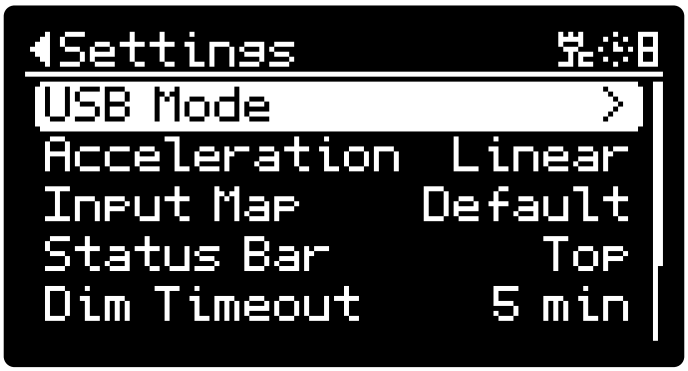

Figure 12.1: Settings screen

Every change in the settings screen takes effect immediately. There are no save or confirmation steps, except on the setting that destroys work. The new configuration is persisted when the screen is left.

12.1 USB Mode

USB Mode chooses whether the USB-C port carries MIDI to a computer, acts as a storage device, or stays dormant. Read more in Chapter 11.

12.2 Acceleration

Acceleration decides how fast anything changes when turning knobs.

Value	Behavior
Linear	One detent is always one step, however fast the knob is turned
Medium	A quick turn multiplies each detent, a slow one does not
Fast	A quick turn multiplies each detent further independent of speed

12.3 Input Map

Input Map decides which gesture is associated with each knob. Read more in Section 4.3.1.

12.4 Status Bar

The status bar is the strip that carries the back arrow, the screen title and the status icons. This setting puts it at the Top of the display or at the Bottom.

12.5 Dim Timeout

After a certain time without knob movement, the display powers off without interrupting the device. The next movement of either knob powers it back on. This setting defines the number of minutes for the display to power off.

12.6 Developer Mode

Developer Mode enables the advanced diagnostic tools. Turning it On reveals two things that are otherwise absent:

- **Mode Timing**, the next setting down on this screen.
- **Log Monitor**, a submenu on the main menu directly below MIDI Monitor.

Read more in Chapter 13.

12.7 Reinitialize All

This restores the current routing to a factory-fresh state. It takes effect immediately after confirmation and there is no undo. Specifically, this action resets:

- Every output goes back to its factory assignment. Output 1 to Output 4 each carry their matching physical inputs. Output 5 to Output 8 and Output USB are turned off with no inputs assigned.
- Every filter on every output goes back to passing everything, and channel remap goes back to Input 1.
- The master clock source goes back to Input 1.

13. Developer Mode

Developer Mode enables two diagnostic tools: a Log Monitor that shows what the loaded modes are reporting, and a Mode Timing readout that shows how long each loaded mode takes to do its work.

> **Note**
>
> The timing measurement is always taken, whether Developer Mode is on or off, so switching it on costs nothing in MIDI processing time.

13.1 Enabling Developer Mode

Enter Settings from the main menu and **Scroll** down to the Developer Mode row, then **Change** it to On. Two things happen as a result:

- **Mode Timing** appears in Settings, directly below Developer Mode.
- **Log Monitor** appears in the main menu, directly below MIDI Monitor.

The setting is stored with the rest of the device settings when the Settings screen is left. A device that was left in Developer Mode comes back in Developer Mode.

13.2 Log Monitor

Modes can write messages into the Log Monitor for debugging or informative purposes. None of the bundled modes writes to the log, which means a stock device shows an empty Log Monitor no matter how long it is left running.



Figure 13.1: Log Monitor with nothing logged

13.2.1 Reading a line

Each entry starts with a single letter giving its severity, and the rest of the line is the text the mode logged:

Letter	Severity	Meaning
D	Debug	Fine-grained details
I	Info	Ordinary progress reporting
W	Warn	Something handled but needs attention
E	Error	Something extraordinary and unhandled

A message too long for one row wraps onto further rows. The screen keeps the last 32 entries. Once it is full, each new entry pushes the oldest one out.

13.2.2 Scrolling and freezing

Scroll moves through the entries. The **Parameters** gesture freezes the screen and a pause icon appears in the top right corner. The same gesture releases it.

13.2.3 Log levels filtering

Enter opens Log Levels to decide which severities are shown. It lists all severities checked by default.

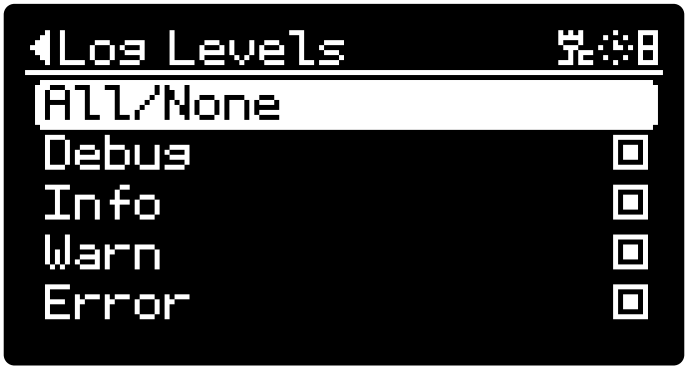

Figure 13.2: Log Levels filter screen

Going back to the Log Monitor applies the filter at once. Clearing a level hides the entries of that level without discarding them, so checking it again brings them straight back, including the ones that arrived while it was hidden.

13.3 Mode Timing

Turning this setting on enables a small timing figure that appears next to every loaded mode in every mode chain. The figure marks a peak, in microseconds, of the longest single run of that mode's MIDI processing since it was loaded.



Figure 13.3: Global mode list with Mode Timing on

The measurement covers the mode's MIDI processing only, even if it includes a custom interface.

13.3.1 Reading the figure

The Operator has two limits for how long a mode will process MIDI. If a mode surpasses a limit, the device does the following:

Limit	Value	What happens
Soft	500 µs	The mode is considered over budget
Hard	5000 µs	Three of these in a row disables the mode

A mode that goes past the hard limit on three consecutive runs is switched off by the device. Its slot keeps the mode and its row gains a `[!]` marking after its name. Opening the slot offers Unload, which is the way to clear it out and start again.

13.4 Turning Developer Mode off

Set the Developer Mode setting back to Off. The Log Monitor row disappears from the main menu and the Mode Timing row leaves Settings.

> **⚠ Warning**
>
> Turn Mode Timing off before turning Developer Mode off. The two settings are independent, so a Mode Timing left on keeps adding its figures to the mode chain rows, and the row that turns it off is no longer in Settings. Turning Developer Mode back on brings that row back with its value as it was left.

14. Updating Firmware

Whenever a new version of the Operator software is available, the device needs to be updated. Updating it means replacing the current software with a different one, which is a single file dragged onto the device from a computer. No extra tools are required.

The current version is drawn in the bottom right corner of the startup screen, as a number formatted like `v1.0.0`. It shows for a moment every time the device is powered on.

14.1 Update files

A firmware update is published as two files:

File	What it does
`operator_update_firmware_only.uf2`	Replaces the current firmware without affecting settings or data
`operator_full_factory_reset.uf2`	Replaces the firmware and returns all data and settings to their factory defaults

The firmware-only file is what you would use regularly. The factory reset file is to be used in misbehaving devices or if you want to start from scratch again.

> **⚠ Warning**
>
> The factory reset only restores the bundled mode files, but erases the routing configuration, every saved preset, the device settings, and any modes installed along with the files they keep.
>
> Before performing a factory reset, make sure you have a backup of your current settings and mode data. Read more in Chapter 11.

14.2 Flashing mode

Flashing mode is the state in which firmware is updated. To enable it:

1. Unplug the USB cable.
2. Press and hold the B button using a straightened paper clip or similar tool. It is located in the hole marked B at the top right of the panel.
3. Plug the USB cable back in with the button still held, then release it.

The display will not turn on. A new drive appears on the computer instead, which signals that the device is waiting for a file.

14.3 Flashing the file

Drag one of the two files onto that drive.

The file will transfer to the device and it will restart itself the moment the transfer is complete, which is completely normal. The whole write takes a few seconds.

Do not unplug the cable while the file is being transferred. If the cable is pulled during transfer, the device comes back into flashing mode on the next start, and the file can be dragged again.

After restarting, the device will carry the new firmware. The startup screen will show the new version in its bottom right corner.

14.4 If something goes wrong

Flashing mode lives in the chip itself rather than in the firmware, so it cannot be erased by a failed update. Holding B while plugging the cable in always reaches it, whatever state the firmware is in, and the file can be dragged again from there.

A. MIDI Implementation Chart

Operator is a MIDI router by design, so what reaches an output is whatever the routing and the filters let through. There is no direct response from the device to any incoming message other than routing it to some destination. Read about routing configuration in Chapter 5.

This page describes two specific things: the messages the Operator sends by itself, and the limits of what it can carry at once.

A.1 Messages sent by the Operator itself

Message	When it is sent
Timing Clock	Only while a clock generating mode is loaded and Mode Clock is set to Mode
Control Change 120, 64, 123 and 121, all value 0	MIDI Panic, on all sixteen channels of every physical output, and of Output USB while the device is in Client mode
A System Exclusive terminator	To close a message whose route was cut part way through
Whatever a loaded mode emits	Under that mode's own rules, described within each mode in Chapter 10.

Every message leaves with its own status byte. Running status is resolved on the way in and never used on the way out, which is what makes merging safe: several inputs can feed one output.

A.2 MIDI processing limits

Messages arriving or leaving the Operator are buffered. In everyday operation this is completely unnoticeable, since they are processed at speeds higher than the MIDI protocol allows. Here are the reference numbers of those buffers, in case anyone is curious:

Limit	Value
Messages queued per input	256
System Exclusive chunks queued per input	64
Messages queued per output	512
Messages taken from one input per pass	32
System Exclusive chunks taken from one input per pass	8

When a buffer is full the newest message is dropped and what is already queued is kept. The per input caps are independent, so one flooded jack cannot take the space the other four need.

A System Exclusive message has no length limit. It travels in 32 byte chunks and is never buffered whole. An output carries one such message at a time, locking it for the duration of its emission or until a full second passes with no further bytes of that message arriving. In the extremely rare case that more than one input is receiving SysEx, any other input's SysEx message is refused and everything else waits, with System Real-Time the exception that passes through.

One message type does not pass at all. MIDI Time Code quarter frame is parsed, and shows up in the MIDI Monitor, but no message type filter covers it, so it reaches no output. A status byte the MIDI standard leaves undefined is treated the same way.

B. Specifications

B.1 Ports

Port	Number	Connector
Input 1 to Input 4	4	3.5mm TRS jack, each with its own A/B switch
Output 1 to Output 8	8	3.5mm TRS jack, each with its own A/B switch
Input and Output USB	1	USB-C

The twelve physical ports carry standard MIDI serial at 31250 baud, with eight data bits, no parity and one stop bit.

A two-position switch sits beside each of the twelve jacks and selects its TRS MIDI type. What the types are, and how to find out which one another device expects, is explained in Chapter 2.

Input USB and Output USB carry MIDI only while the USB port is in Client mode. What the USB port does in each of its modes is covered in Chapter 11.

B.2 Latency

Operator works in real time. A common, three byte MIDI message spends about 960 microseconds arriving over a MIDI cable, because each byte is ten bits at 31250 baud. The device adds about 33 microseconds to that, measured from the last bit reaching an input to the first bit leaving an output. That is a tenth of the time a single byte spends on the wire, so a message leaves at the first slot MIDI gives it and the delay is unnoticeable.

B.3 Power

The device draws well under 100 mA from the USB-C port, which is less than any USB charger, computer port or powered hub gives, so anything with a USB socket runs it. There is no power switch, and the device is ready about a second after power arrives, once the splash clears. Where the power comes from and what else that port does is covered in Chapter 2.

B.4 Display

The screen is a monochrome OLED with a resolution of 128 x 64 pixels. Every screenshot in this manual is a pixel perfect reproduction of what shows in the device.

B.5 Storage

The device has an internal flash storage of 16 MB. It is divided into three regions, and only one of those regions appears on a computer when the Operator is in USB Drive mode.

Region	Size	Holds
Firmware	1 MB	Operator firmware
Exchange	13 MB	USB Drive contents
Internal	2044 KB	internal Operator settings

Nothing outside the Exchange region is visible over USB. The drive's layout is set out in Chapter 11.

B.6 Firmware specifications

Item	Figure
Preset slots	16
Modes per chain	4
Maximum number of modes listed	32
Maximum mode size	256 KB
Mode RAM pool	192 KB
Maximum parameters per mode	36

The buffer sizes on the MIDI path are listed in Appendix A.

C. Troubleshooting

Here are some quick solutions to the most common problems found during normal operation.

C.1 Use the MIDI Monitor

Most routing problems are invisible because MIDI is invisible. The MIDI monitor makes the traffic visible and is a great tool to check exactly what is reaching or leaving the device.

At the MIDI Monitor, its display can narrow most problems to:

- Nothing appears at all when the sending device plays. The problem is upstream of the routing, so work through the physical entries below.
- Messages appear on an input but never on an output. The problem is in the routing or the filters, so work through the routing entries below.

The monitor, its port filter and the abbreviations it uses are covered in Chapter 8.

C.2 Physical connections

C.2.1 One port shows no activity at all

A port that shows no activity at all, using a cable **that works on another device**, is almost always an **A/B switch in the wrong position**. A switch set to the type the other device does not use passes no data at all.

Check the switch beside that jack against the type the other device uses. The switches, the two types and how to find out which one another device expects are covered in Chapter 2.

C.3 Routing

C.3.1 Nothing comes out of an output

Perform the following checks in order for the output in question:

1. **Disable.** A disabled output emits nothing and shows the disabled option only in its Output settings screen.
2. **Inputs.** An output with no input assigned has nothing to pass. At least one input must be selected for the output to work.
3. **Msg Types.** An output only passes the message types checked here.

C.3.2 An output emits some messages but not others

Routing filters are set per message type and per channel. Below are the most common symptoms of missing messages. Read more in Chapter 5.

What is missing	Setting to check
A whole class of message: notes, CC, etc...	Msg Types
Everything on one MIDI channel	Channels
Notes above or below a point on the keyboard	Note Range
One controller number, or some of them	CC Range

When more than one input is assigned to an output, these settings open a submenu with one entry per assigned input, so a filter can be right for one input and wrong for another. Check the input the traffic is actually arriving on.

C.3.3 Notes arrive on the wrong MIDI channel

The **Ch Remap** setting rewrites the channel of everything the output passes from that input. Read more in Chapter 5.

C.3.4 Input USB or Output USB is missing from the menus

The two USB ports exist only **while the USB Mode is set to Client**. If it is Off or Drive, the ports are absent from the Inputs list, from the Outputs list and from the clock menus. Read more in Chapter 11.

C.4 Clock and timing

C.4.1 An output does not emit clock

Clock is a message type like any other. The output passes it only **when Clock is checked in that output's Msg Types**. That filter is covered in Chapter 5. Clock distribution across the outputs is covered in Chapter 6.

C.4.2 Everything is following the wrong device

Master Clock, on the main menu, names the input whose clock is distributed to every output that accepts clock. If it is set to None, no input has that role and each output falls back to its own Clock Source. Read more in Chapter 6.

C.4.3 Clock Source setting is absent from an output screen

Clock Source appears only when there is no global clock master and the output carries more than one input with Clock enabled. With a single input assigned, that input is the output's clock by definition and the Clock source setting disappears by design. **Set Master Clock to None first**. Read more in Chapter 6.

C.4.4 A loaded mode generates a clock but the outputs follow an external one

Mode Clock, on the main menu, decides which clock source drives modes. **When Mode Clock is set to Mode and a clock generating mode is loaded**, that mode becomes the only source of clock and the device stops passing external clock sources. With Mode Clock set to an input, loaded modes follow that input instead. Read more in Chapter 6.

C.5 Stuck notes

C.5.1 A note keeps sounding non-stop

Panic silences everything. It is the last row of the main menu, and can also be quickly activated by **pressing both knobs simultaneously in any screen for a while**.

A note that comes back immediately after a panic is still being sent from the outside, so the next thing to look at is the sending device.

C.6 Modes

C.6.1 A mode does not appear in the browser

The browser lists the modes for the kind of chain they would load into, and shows `(no modes)` when there are none. Global modes live in the `/global/` folder and output modes in `/output/`, on the Operator internal drive.

Loading is covered in Chapter 9. Drive mode is covered in Chapter 11.

C.6.2 A loaded mode does nothing

Modes act only on the chain they were loaded into. The global mode chain sees the whole device, and output mode chains see only that output. Loading an output mode into one output has no effect on any other output. Read more in Chapter 9.

Check the output itself too. A mode on an output whose filters drop the traffic has nothing to work on, so the routing entries above still apply.

C.6.3 A mode is misbehaving

Turn on Developer Mode in Settings. It enables the Log Monitor on the main menu, and the Mode Timing setting, which puts each active slot's measured time beside its name on the chain screen. What both of them show is covered in Chapter 13.

C.7 USB

C.7.1 My computer does not see the Operator

The USB port does one thing at a time. It carries MIDI only in Client mode, and shows as an external drive in Drive Mode. Changing the USB Mode always asks to restart, because the port changes its behavior on reboot. Read more in Chapter 11.

C.8 My settings did not save

A screen writes its changes only when it is left, not on every turn of a knob. Backing out to the home screen after an edit is what persists it. Powering the device off while still inside the screen that was being edited will not save any changes. Read more in Chapter 12.

D. Glossary

A/B switch

The two-position switch beside each jack, selecting whether that jack is wired as TRS MIDI Type A or Type B. Chapter 2.

B button

The small button behind the hole labeled B at the top right of the panel. Held with a straightened clip or a pin while the device powers on, it puts the device into flashing mode. Chapter 14.

Chain

The ordered set of slots that modes are loaded into. The device has one global chain and one chain per output. Chapter 9.

Channel remap

The output setting that rewrites the MIDI channel of everything passing to that output from a given input. Chapter 5.

Client mode

The USB mode in which the device acts as a MIDI interface for a computer. It is the mode that makes Input USB and Output USB available. Chapter 11.

Clock Source

The per-output setting naming which input supplies that output's clock. The row appears only when Master Clock is None and the output carries more than one input with clock enabled. Chapter 6.

DATA

The right knob, named for what it does. Turning it changes the value the cursor is on. Chapter 4.

DIN

The five-pin round connector that older MIDI equipment carries. Operator's own jacks are 3.5mm TRS, so a device with a five-pin socket needs an adapter. Chapter 2.

Drive mode

The USB mode in which the device presents its mode storage to a computer as a drive. Chapter 11.

Global mode and output mode

The two kinds of modes. A global mode sits in the device's own chain and works on all the traffic. An output mode sits in one output's chain and works only on what reaches that output. Both kinds are covered in Chapter 9. The modes the device ships with are listed in Chapter 10.

Input 1 to Input 4

The four physical MIDI inputs. Appendix B.

Input USB

The MIDI input the device offers over its USB-C port while it is in Client mode. Chapter 11.

Master Clock

The main menu setting naming the one input whose clock reaches every output that accepts clock. Chapter 6.

Merge

What an output does when more than one input is assigned to it. The streams are combined onto that one output, and each input keeps its own filters. Chapter 5.

Mode

A small program, built separately from the firmware, that the device loads from storage to do something the routing alone cannot. Chapter 9.

Mode Clock

The main menu setting that names the clock which modes will follow. Chapter 6.

Output 1 to Output 8

The eight physical MIDI outputs. Appendix B.

Output USB

The MIDI output the device offers over its USB-C port while it is in Client mode. Chapter 11.

Panic

The command that silences everything downstream at once, by putting the silencing controllers on all sixteen MIDI channels out of every output.

Preset

A saved copy of the whole routing setup, filed under a name in one of sixteen slots and recalled from the same screen. Chapter 7.

Running status

A MIDI standard way of transmitting messages without sending the status byte, implying every following message has the same status.

SELECT

The left knob, named for what it does. Turning it moves the cursor. Chapter 4.

Slot

One position in a mode chain. A slot is empty or holds exactly one mode. Chapter 9.

Status bar

The strip along one edge of the screen carrying the current screen's title and a small set of state icons, described in Chapter 4. Which edge it sits on is a setting in Chapter 12.

TRS MIDI Type A

The TRS MIDI wiring the 2018 MMA standard settled on, and the position every A/B switch is set to when the device leaves the factory. Chapter 2.

TRS MIDI Type B

The other TRS MIDI wiring, chosen by moving a jack's A/B switch to its right position. Chapter 2.