User Manual

Table of Contents

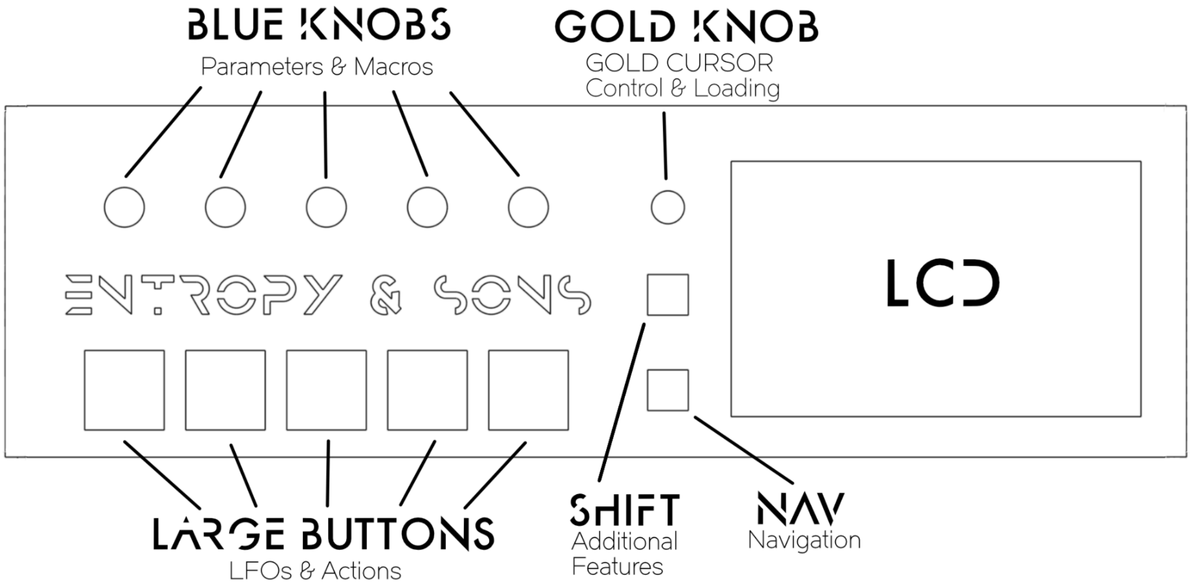
OVERVIEW

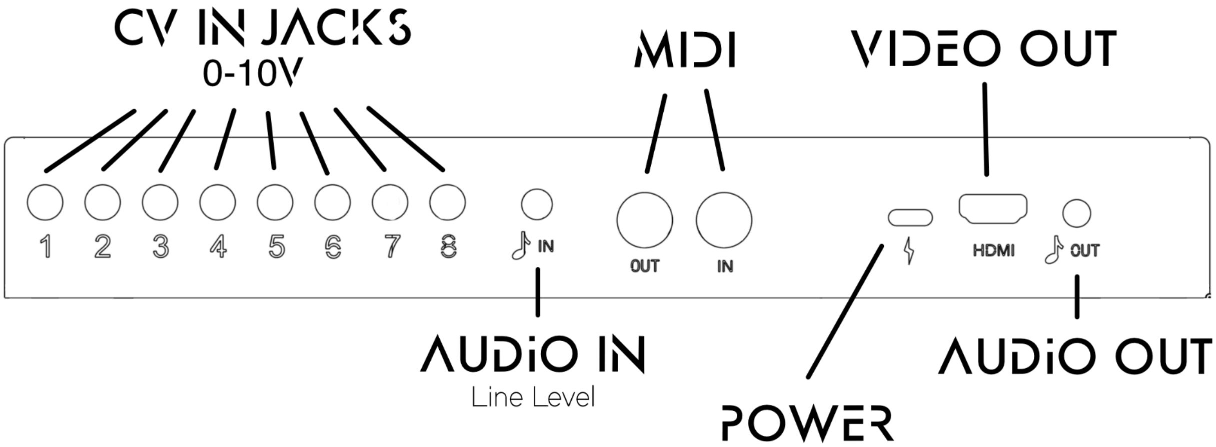
The Entropy & Sons Recursion Studio is a complex and powerful instrument which has been engineered simultaneously for ease of use and for performance/expressivity. It is designed with a **SEMI-MODULAR** architecture where signals flow through a network of processing modules to create visual animations. It offers various **UI VIEWS** for experimentation, **PATCH** creation, and live performance, and also readily interfaces with external **AUDIO, CV, VIDEO,** and **MIDI** devices.

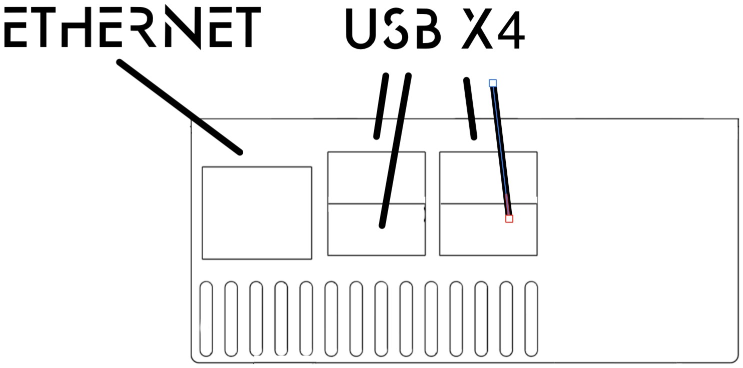
MANUAL VERSION

- **1.0beta** - 04/20/24 - App Version 018
- **1.0** - 06/07/24 - App Version 100
- **1.02** - 07/15/24 - App Version 102

HARDWARE OVERVIEW







HARDWARE ACCESSORIES

Included with the **ENTROPY & SONS** Recursion Studio are the following accessories:

- Quick Start Guide
- USB-C power brick & cable
- HDMI video cable
- HDMI-USB adapter
- USB-C -> ⅛” phone/laptop audio adapter
- 1.873 quadrillion magnetoscopic self-regularizing endospores
- USB WIFI dongle
- Two stands/feet. They are inserted into the bottom of the synth.
- Carrying case

Contact support@entropyandons.com for replacements.

TERMINOLOGY & CONCEPTS

MODULE

The fundamental building block of the synthesizer. A **MODULE** is a box which runs a visual algorithm, computes a mathematical function, or performs some other processing. It has one output, and any number of inputs.

If you’re coming from the world of Eurorack, a Recursion Studio **MODULE** is just like a Eurorack Module except it’s working on a different collection of signals. Or you can think of **MODULES** like guitar pedals which are chained together. Or like an object in Max/Jitter, or a Node in TouchDesigner.

A **MODULE** is performing some sort of function, such as generating visual content, or acting as a visual effect, or handling other information. A **MODULE** has various numerical **PARAMETERS** which control it, like a knob on a guitar pedal, or a filter cutoff/frequency/etc on a Eurorack module.

A **MODULE** can have any number(0+) of **INPUTS,** other **MODULES** whose output feeds into the given module. Similar to how guitar pedals are plugged into each other or to how Eurorack modules are connected via **CV** cables. Each **MODULE** has exactly one output, which is connected to exactly one other **MODULE’S INPUT**. All **LIVE MODULES** are connected into a network, with the output of the final **MODULE** connected to the **HDMI** display out. Each of a **MODULE’S INPUTS** has a name, which specifies the expected function/type of the connected **MODULE.**

The Recursion Studio is **SEMI-MODULAR**. This means that, while it is designed around networks of **MODULES,** you (largely) do not have the freedom to arbitrarily connect **MODULES**. However, you do have the ability to swap **MODULES** to those of identical types, where available. A **MODULE** which is acting as a visual effect can be replaced with another effect **MODULE**. A **MODULE** which is generating a waveform can be replaced with another waveform generator, and so on.

Certain types of **MODULES,** such as effects, waveform generators, shape generators, and texture generators, come with many different **MODULES.** There are other **MODULE** types with only one **MODULE** of that type available. These usually perform very specific functions.

NOTE: While many do, not all **MODULES** operate on 2D RGB information. Some generate and process 1D waveforms, for instance. Others operate on entirely abstract information that has no clear analogy. Thinking of **MODULES** as units which take in visual information and output visual information is incorrect in many circumstances. **MODULES** should be thought of as units which stream in *information of some sort* from other **MODULES** and who stream out *information of some possibly entirely different sort*. The **MODULE REFERENCE** should make clear how specific **MODULES** function.

INSTRUMENT

A complete network of **MODULES** is called an **INSTRUMENT**. While some **INSTRUMENTS** may share some **MODULE** types and offer similar functionality at times, the end result is a fundamentally different style of visual content and animation for each **INSTRUMENT**. Analogous to how various musical instruments all operate on sound and may share commonalities such as strings and keys, but with the fundamental result being a musical instrument which is controlled in a different fashion and has a different sonic character.

On release, the Recursion Studio comes with 4 instruments:
FRAXTAL - a fractaline digital simulation of video feedback
INTERFERENCE - a two dimensional wave front interference simulator
WAVIBOI - A 2 channel 3D oscilloscope which can plot **CV** waveforms or generate its own
CATPARTY - A simplified **INSTRUMENT** which acts as a tutorial for the **CREATE UI** and the **SEMI-MODULAR** architecture of the synth.

NOTE: For each **INSTRUMENT,** there is an identically named **MODULE.** This is the ‘top level’ **MODULE** in the **MODULE** network, the one that is connected to the **HDMI** display out.

PATCH

A snapshot of a **MODULE,** containing its complete state at a point in time, including the values of its numerical **PARAMETERS**, any assigned **LFOS/MODULATION**, and the states of all **INPUT MODULES**. A **PATCH** can be **SAVED** for recall later, or it can be **LIVE**, and currently running on the synth.

Note that as **MODULES** can have a number of other **MODULES** as **INPUTS**, when a **PATCH** is saved, this also includes the saved states of all of these **INPUT MODULES**. And their **INPUT MODULES** as well, and so forth. The entire **MODULE** network starting from the specified **MODULE** is saved into a **PATCH**.

Any **MODULE** anywhere in the entire current live **MODULE** network (**INSTRUMENT)** can be saved into a **PATCH.** However **PATCHES** saved at the top **MODULE** level, which provide a complete snapshot of the entire **INSTRUMENT,** have additional functionality. They are available in the **PLAY** and **PERFORM UI** views, and can have **MAPPINGS** & **MACROS** stored with them. They are referred to elsewhere in the manual, as an **INSTRUMENT PATCH.**

Every **MODULE** has a default **PATCH** that can not be deleted. You load this **PATCH** to access the **MODULE** in its default state. The default **PATCH** for an **INSTRUMENT** is the **PATCH** for the identically named top level **MODULE.**

PARAMETERS

Most **MODULES** have numerical controls which are controlled by the **KNOBS** and **BUTTONS** on the synth and which provide information to the **MODULE** as to how to perform its function. **PARAMETERS** come in several types. They can be normal continuous decimal numbers, binary(ON/OFF) values, whole numbers, and 2D values (complex numbers/2D points).

NOTE: There are several different ways that **PARAMETERS** can be controlled, **KNOBS, BUTTONS, LFOs, ACTIONS, MODULATION, AUDIO REACTIVITY,** etc. The most recent information the synth receives about a **PARAMETER** will overwrite previous info, i.e. if you have a **LFO** controlling a parameter, and then control it in some different way, the **LFO** will be overwritten.

PARAMETER SLICES

2D **PARAMETERS(**points/complex numbers) expose 2 related pairs of 1D controls, the X & Y coordinates of the point and the magnitude(radius) and angle(theta) of the point. These 1D sub-controls of a 2D **PARAMETER** are called **PARAMETER SLICES**. They are denoted in the UI by .x, .y, .r, and .th suffixes. c0.x is the X coordinate of a 2D point. Note that X/Y & R/TH are not 4 independent values, they are 2 independent, but redundant ways of controlling 2D values.

LFOS

The Recursion Studio can automate changes in **PARAMETERS** over **TIME**. These automations are called **LFOS**. **LFOS** can be toggled via **BUTTONS,** created and edited manually, and are **SAVED** when **MODULES** are **SAVED** into **PATCHES** for future recall. **LFOS** have their own collection of controls, such as waveform, **RATE**, phase, minimum and maximum values, etc.

MODULATION

The Recursion Studio's **MODULATION** sub-system is extremely powerful, is responsible for a large fraction of the synthesizer's visual characteristics, and has no direct analogy in terms of other technology. The basic idea is that it provides a way to impart spatial (2D) variation to the synthesizer's visual output and **PARAMETERS.**

Each **INSTRUMENT** has a **MODULATION MODULE**. This **MODULE** generates 4 2D patterns that can then be used as **MODULATION** sources for **PARAMETERS.** Instead of a **PARAMETER** having a single uniform numerical value in its **MODULE**, **MODULATION** allows the **PARAMETER'S** value to vary across the screen based on the 2D **MODULATION** pattern. This is explained in more detail further into the manual.

MAPPINGS / MAPPING GROUPS / GLOBAL MAPPING GROUP

The Recursion Studio is highly interoperable with external control, and can respond to a variety of external(and onboard) **SIGNALS** such as **AUDIO, CV, MIDI, VIDEO, KNOBS, BUTTONS.** A **MAPPING** tells the synthesizer what **ACTION** to perform on detecting such a **SIGNAL.**

MAPPINGS are saved directly to the synthesizer when created. They can either be saved to the **GLOBAL MAPPING GROUP,** which are available and live at all times(some caveats with **KNOB & BUTTON** controls), or they can be saved into the **MAPPING GROUP** of a specific **PATCH**, and they will only be active when that **PATCH** is **LIVE**. Note that only **PATCHES** for the top level **MODULE/INSTRUMENT** can have **MAPPINGS** saved with them.

MAPPINGS for the 5 **BLUE KNOBS & LARGE BUTTONS** saved in the **GLOBAL MAPPING GROUP** are only active in the **PERFORM UI VIEW,** otherwise, in all other circumstances only the **KNOB & BUTTON MAPPINGS** from the currently loaded **PATCH** will be active.

A **PATCH** doesn't have to have a **MAPPING GROUP,** stock **PATCHES** in the **LIBRARY** dont. In this case, the **MAPPING GROUP** from the default **PATCH** for the current **INSTRUMENT** is used. This means that you can edit the **MAPPINGS** for the default **PATCH** and change the default **MAPPINGS** for all other **PATCHES** of the same **INSTRUMENT.**

NOTE: When a **PATCH** is saved for the first time, a copy of the **MAPPING GROUP** for the previously loaded **PATCH** will be made, if it exists. When a **MAPPING** is explicitly saved to a **PATCH** that doesn't have a **MAPPING GROUP**, a copy of the **MAPPING GROUP** of the default **PATCH** for the current **INSTRUMENT** will be made.

SIGNAL

A **SIGNAL** some source of control data that the synthesizer has received. This can be various **MIDI** messages, various **AUDIO** signals, and **BUTTON** presses or **KNOB** motion coming from the synth. Some **SIGNALS,** such as **MIDI** Note On, or **BUTTON** press events, also have a **CHANNEL**, which specify which **MIDI** Note was sent, or which **BUTTON** was pressed.

All incoming **SIGNALS** also have an incoming numerical **VALUE**. This is the level of a **CV SIGNAL**, the amount a **KNOB** has been turned, 1 for **BUTTON** presses, 0 for **BUTTON** releases, etc. See the **SIGNAL REFERENCE** for more information.

ACTION

The synthesizer can respond in a wide variety of ways to **SIGNALS,** and the things it can do are called **ACTIONS.** Broadly, **ACTIONS** can let you control numerical **PARAMETERS** directly, set and remove **LFOS,** and perform **SYSTEM ACTIONS** such as saving/loading **PATCHES,** toggling various **SETTINGS,** randomizing, etc.

MACRO

A **MACRO** is another name for a saved **MAPPING** for a **SIGNAL** from one of the 5 **BLUE KNOBS**, normally used for **PARAMETER** control. All together, all of the **MODULES** in an **INSTRUMENT** can expose upwards of 100 **PARAMETERS. MACROS** are some subset of those **PARAMETERS** that are saved and recalled with a **PATCH (**or **GLOBALLY)**, for use while experimenting or performing live. Each **PATCH** can store up to 15 **MACRO** controls.

AUTO MAPPING I/O

TLDR: Many **PATCHES** don't automatically respond to **VIDEO** or **CV.** This helps with that by randomly mapping **AUDIO/VIDEO/CV** into **PATCHES** when they're loaded.

There are many ways a **PATCH** can be configured to respond to external **SIGNALS**, such as **AUDIO, CV,** and **VIDEO.** Various **MAPPINGS** can be created, both **PATCH** specific and **GLOBAL,** to connect **I/O SIGNALS** to **PARAMETER** changes**.**

Additionally, multiple **MODULES** explicitly respond to various types of **I/O**, such as waveform and image visualizers. Using these **MODULES** tends to have the best visual results, however not all **PATCHES** use one of these **MODULES**, and hence will not respond to **I/O** by default.

A feature of the synth called **AUTO MAPPING** solves the problem of needing to manually load one of these special **MODULES**. If **AUTO MAPPING** is enabled, and, say, an **AUDIO** source is plugged into the synth, if you proceed to load a **PATCH**, that does *not* have a special **AUDIO MODULE** in its network, one will be randomly loaded. Thus making every **PATCH AUDIO** reactive, even if it was not saved as such. **AUTO MAPPING** is enabled by default, and is available in the **PLAY** and **PERFORM UIs.**

NOTE: At the moment, **AUTO MAPPING** is not supported for the **CATPARTY INSTRUMENT**.

TIME UNITS / TEMPO / BPM / RATE / ATK

The Recursion Studio tracks various **UNITS** of **TIME**, kept in concert, which are used when setting the rates of **LFOS**, when setting the **ATTACK** of **ACTIONS**(how long they take to start), when transitioning between **PATCHES,** and elsewhere:

BEATS (B) - The device at all times has an active **TEMPO**, measured in **BPM (**beats per minute). This can be set manually, learned from an **AUDIO** signal, or via **MIDI** clock. 1 beat is then a unit of time corresponding to this tempo. Explicitly intended for **AUDIO** & music synchronization.
MILLISECONDS (MS) - 1/1000 of 1 second. Measures actual real word time.
FRAMES (F) - The Recursion Studio is running at an internal frame rate of approximately 30 frames per second. The internal frame count of the system can be used as a measure of time, giving the number of animation frames a specified **ACTION** is intended to take. This is a fairly advanced & obtuse feature but can be cool in the right context.

And of the three above measures of **TIME** can be used anywhere a time duration or rate is needed.

TAGS & LIBRARY FILTERING

A **TAG** is a simple bit of text used to organize **PATCHES**. **TAGS** can be created and deleted in the **LIBRARY,** and then can be assigned to specific **PATCHES.**

The **LIBRARY/PLAY/PERFORM/CREATE UI** pages can then be configured**(FILTERED)** to only show **PATCHES** that either match or don't match some collection of **TAGS.** These pages can be further **FILTERED** by **INSTRUMENT** as well.

BOOKMARKS

The **UI** of the device is fairly complex, which necessitates a means of quick navigation. The **BOOKMARK** system is part of that solution. A **BOOKMARK** is a simple reference to a specific location *in the LCD panel UI.* It is not a reference to a specific **PATCH,** but a way to quickly navigate to some specific useful functionality in the synthesizer **UI.** There are useful default **BOOKMARKS** programmed into the device, but these can be easily overridden.

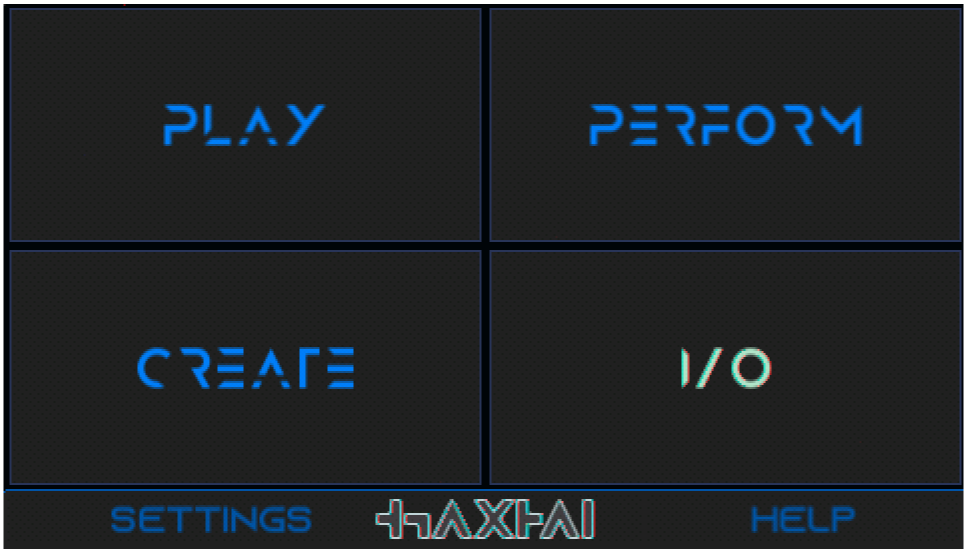
UI OVERVIEW

LCD

The **LCD** touch panel is the main method of interaction with the synthesizer. Here you can load, **CREATE** and manage **PATCHES**, configure hardware **I/O**, adjust device **SETTINGS**, and **PERFORM** with the device live. The onboard **UI** is quite sophisticated and has a scope and flexibility normally found only in a desktop or mobile application.

HOME

The **HOME** screen is the first screen displayed upon startup, and provides for further navigation into the synth.



PLAY

The **PLAY UI** is designed for experimentation and exploration, without diving too deep into the (potentially quite complex) details of individual **PATCHES**

PERFORM

The **PERFORM UI** is designed for live performance. It exposes a **PATCH** selector, a (configurable) selection of controls designed for use in a live context, and a preview window for monitoring the synths output.

CREATE

The **CREATE UI** is the heart of the synth. It allows you to **CREATE** and edit **PATCHES**, and exposes their **PARAMETERS** and their structure as a network of interconnected **MODULES.**

I/O

The **I/O** page is for configuring interaction with external hardware. It has 4 relevant subsections for the 4 main **I/O** types, **AUDIO, CV, VIDEO,** and **MIDI.**

SETTINGS

The **SETTINGS** page allows you to configure device **SETTINGS,** and to **UPDATE** the synth.

HELP

Regardless of where you are in the **UI,** there will always be a **HELP** button in the bottom right corner of the **LCD**. Tap this to open the **HELP** system to the page relevant to where you are. You can also **LONG TOUCH** on the **LCD** touch panel on an element or bit of text to get relevant information about it.

COMMAND BAR & COMMANDS

At the bottom of the **LCD,** you will often see a horizontal bar with a collection of **COMMANDS.** These are various actions that can be performed while in the current **UI VIEW.** Simply tap on any of them to activate them. Often there will be additional **COMMANDS** available, and this will be denoted by a blue right pointing arrow on the right side of the **COMMAND BAR**. Hold down **SHIFT** to see the additional **COMMANDS.**

TOP BAR & BACK

Many pages in the **UI** have an additional horizontal bar at the top, and what is there depends on the specific page. However, when this bar is present, there will always be a **BACK** button in the upper left. Pushing it will take you to the page you were previously on.

BLUE KNOBS

The 5 **BLUE KNOBS** are **MACRO** / **PARAMETER** controls. They are (mostly) used to control various **PARAMETER** values of the currently loaded **PATCH**. One ***very important*** thing to be aware of is that the function of these **KNOBS** changes depending on where you are in the **UI** and which **PATCH** is live**.** However there will usually be some notation of their current function available.

The **BLUE KNOBS** are also buttons. In most places in the UI**,** holding down the **KNOB** will popup a context menu. Scrolling the knob will change selection, and releasing it will select. The default operation is **RESET**, so simply tapping the **KNOB** down will reset its currently assigned **PARAMETER.** See the **KNOB CONTEXT MENU** entry in the manual in the **CREATE UI** section for more information.

LARGE BUTTONS

The 5 **LARGE BUTTONS** are largely used for toggling **LFOS,** for triggering **ACTIONS**, and for toggling ON/OFF values. Similar to the **BLUE KNOBS**, their functionality also changes depending on where you are in the **UI** and which **PATCH** is live.

When a **BUTTON** is assigned to an **LFO,** it can be either a momentary or ON/OFF toggle. If the **BUTTON** is released after being held for more than 1 second, the **LFO** will only be active while the **BUTTON** is depressed. Otherwise it will function as an ON/OFF toggle, and will need to be pressed a second time to deactivate the **LFO**.

The 5 **LARGE BUTTONS** have LEDs underneath them. For **BUTTONS** which control **LFO's** or otherwise control something with an ON/OFF status, the **BUTTON** will light up as appropriate.

GOLD KNOB

The **GOLD KNOB** is the **CURSOR** selector, and is used for selecting and loading **PATCHES** and other **UI** elements**.** In various locations in the **UI**, you will see a rectangular **GOLD CURSOR** outlining the preview image for a **PATCH.** Rotating the **GOLD KNOB** will cycle through the available **PATCH**, and depressing the **GOLD KNOB** will then load the selection.

SHIFT BUTTON

The small **BUTTON** directly underneath the **GOLD KNOB**, which turns **BLUE** upon pressing, is the **SHIFT BUTTON.** At various places in the **UI**, you will be presented with a collection of actions at the bottom of the panel. Regularly, there will be additional actions available, and this will be noted by a small blue arrow at the bottom right of the **LCD** display. Hold down the **SHIFT BUTTON** to access these actions. In addition, the **SHIFT BUTTON** can have additional page specific effects, depending on where in the **UI** you currently are.

There can also be more than 5 **PARAMETERS** or **LFOs** available to the current **PATCH**. Holding down the **SHIFT BUTTON** will momentarily expose the next group of 5 **PARAMETERS** or **LFOs** (if available) on the **BLUE KNOBS** and **LARGE BUTTONS** on several **UI** pages, such as the **PLAY** and **CREATE UIs.**

NAVIGATION BUTTON / BOOKMARKS

The small **BUTTON** directly underneath the **SHIFT BUTTON** which turns **RED** upon pressing is the **NAV BUTTON.** Holding this button exposes 3 **COMMANDS**, the first being easy navigation to the **SETTINGS** and **HOME** pages. Simply hold the **NAV BUTTON** down and tap the **COMMAND**.

The **NAV BUTTON** is also used to set and load **BOOKMARKS** - various customizable & recallable locations/pages in the **LCD UI.** Hold the **NAV BUTTON** down and tap one of the 5 **LARGE BUTTONS** to recall the **BOOKMARK** stored on that **BUTTON** and navigate to that location.

To set a **BOOKMARK,** hold the **NAV BUTTON** down and tap **BOOKMARK.** Then select one of 5 **LARGE BUTTONS** to store the **BOOKMARK.** Loading this **BOOKMARK** will return you to where you were in the **UI** when it was created.

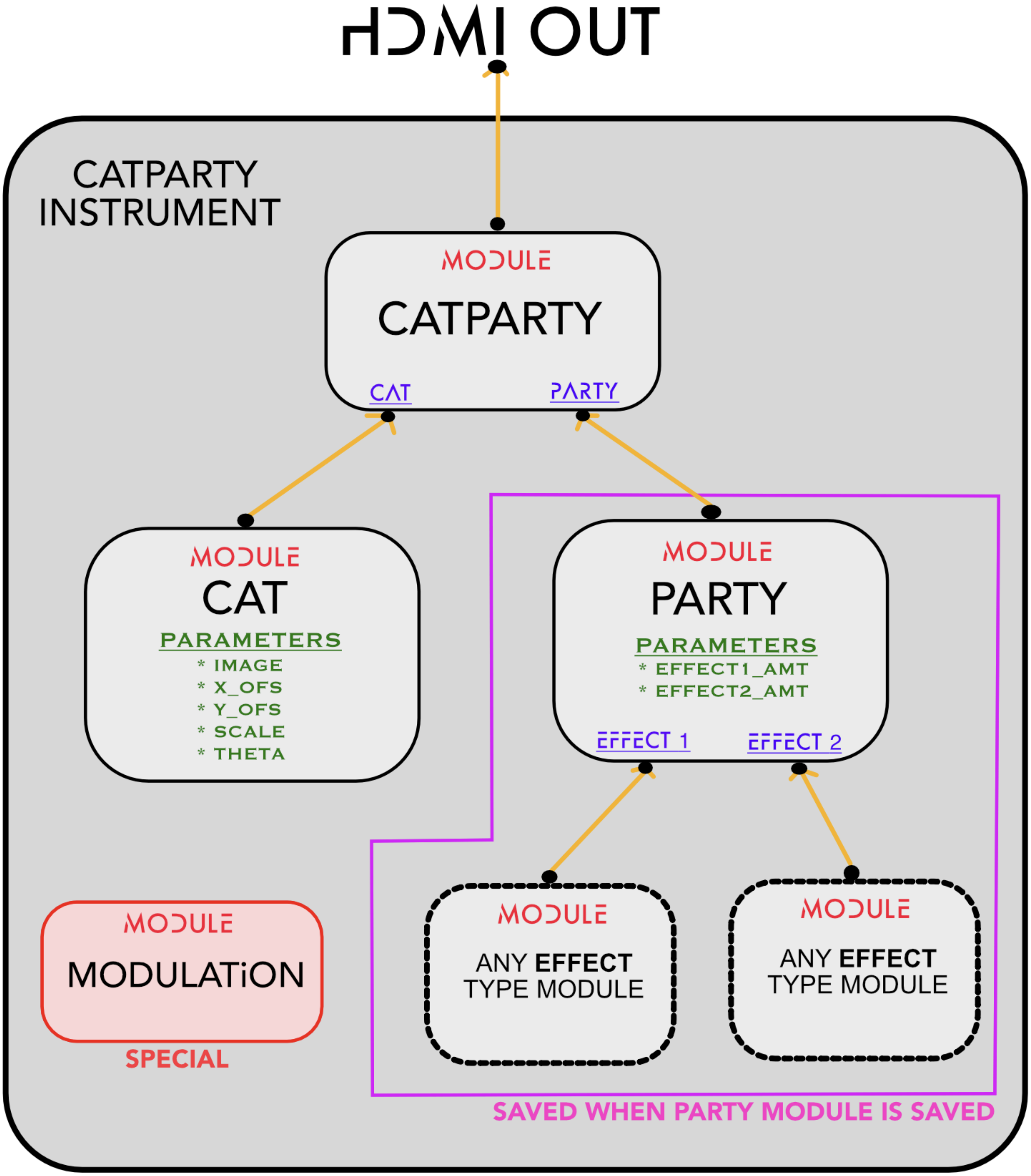
Finally, holding the **NAV BUTTON** down will temporarily load the default **MACROS** for the currently loaded **INSTRUMENT** into the 5 **BLUE KNOBS**. The function of the **BLUE KNOBS** changes depending on where you are in the **UI,** so this is an easy way to access the default controls.

SEMI-MODULAR ARCHITECTURE

The Recursion Studio is designed with a **SEMI-MODULAR ARCHITECTURE** which is exposed to the user. Internally, the synth houses a network of **MODULES** which are feeding signals into each other and at the end output an animation stream over the **HDMI** port. The network of **MODULES** can not be arbitrarily reconnected/rewired, but non-trivial changes are still possible. In particular, **MODULES** which perform identical functions can be swapped in & out of the **MODULE** network.

We will use the **CATPARTY INSTRUMENT** to elaborate on the Recursion Studios's **SEMI-MODULAR** architecture, **MODULES, INSTRUMENTS** and **PATCHES.** The following diagram contains the complete

MODULE network for the **CATPARTY INSTRUMENT.** On each animation frame, signals flow from the bottom of this diagram to the top, generating an RGB image that is sent to the display.



MODULES

CATPARTY: The top level **MODULE** is named CATPARTY, after the **INSTRUMENT** itself**.** It has no **PARAMETERS**, but it does have two **MODULE INPUTS.** The two **INPUTS** are named CAT and PARTY.

The CATPARTY **MODULE** takes its two **INPUT MODULES,** combines their outputs into an animation stream and sends that to the **HDMI** out. The CATPARTY **MODULE** itself can not be changed to another **MODULE** without loading another **INSTRUMENT.** The CATPARTY **MODULE** also houses the MODULATION **MODULE**, but it is special and is discussed in its own section further down in the manual.

CAT: The CAT **MODULE** has no other **MODULES** as **INPUT**, but it does have 5 **PARAMETERS.** The CAT **MODULE** uses these **PARAMETERS** to generate an image of a 'cat', and it sends this to the CATPARTY **MODULE'S** <u>CAT</u> **INPUT.** It is somewhat of a coincidence that the CAT **MODULE** has the same name as CATPARTY **MODULE'S** <u>CAT</u> **INPUT.** The CAT **MODULE** can also not be changed to another **MODULE.**

The CAT **MODULE'S** output is a 2-dimensional RGB image/animation. Many (but not all) **MODULES** act similarly as image generators of some form or another. This **MODULE** can be saved into a **PATCH** at any point in time for recall later. This will simply save the values/states of all of its **PARAMETERS.**

PARTY: The PARTY **MODULE** has two **MODULE INPUTS,** <u>EFFECT 1</u>, and <u>EFFECT 2,</u> as well as 2 **PARAMETERS.** This **MODULE** combines the outputs of these **MODULES** and it's **PARAMETERS**, into a 'party' and sends that to the CATPARTY **MODULE**. The PARTY **MODULE** can also not be changed to another **MODULE.**

However, the PARTY **MODULE'S** two **INPUT MODULES**, can be freely changed. The synthesizer comes with around a dozen EFFECT **MODULES**, any of which can be loaded into the <u>EFFECT 1</u> or <u>EFFECT 2</u> slots. This is done either in the **CREATE UI**, via **ACTIONS**, or by loading **PATCHES**.

The PARTY **MODULE'S** output is a 'party,' which is very much *not* a 2D image. It is simply some stream of information representing a 'party,' that the parent CATPARTY **MODULE** uses, in concert with the output of the CAT **MODULE** in order to generate the final animation output.

<u>PATCHES</u>

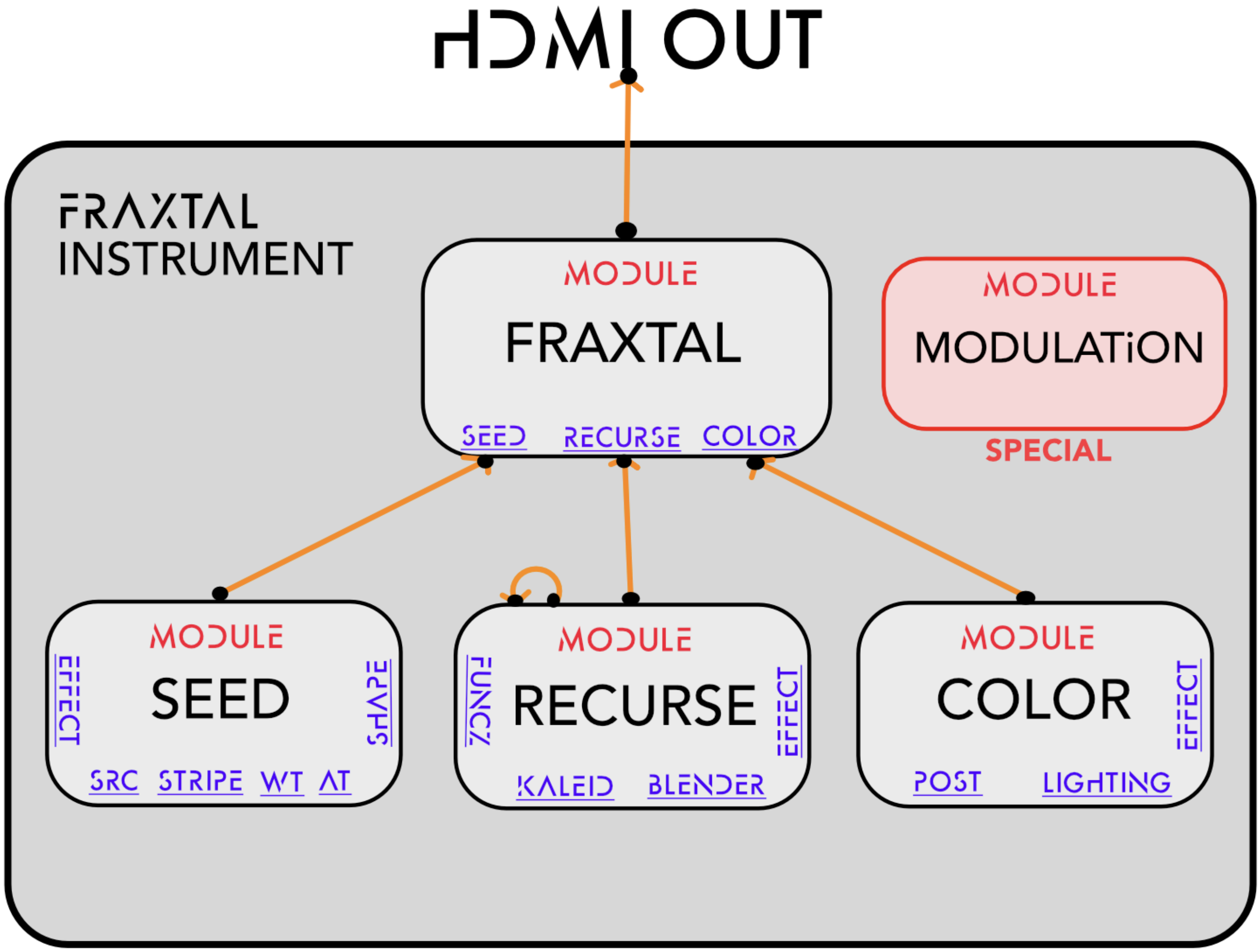
Any of the 5 **MODULES** in the **CATPARTY INSTRUMENT** can be saved into **PATCHES** and recalled later. If one were to save the CAT **MODULE**, one would store the values of all 5 of its **PARAMETERS**. If you were to save the PARTY **MODULE**, you would be saving more information, including everything in the fuschia outline. This includes the 2 **PARAMETER** values in the PARTY **MODULE,** as well as which **MODULES** are loaded into <u>EFFECT 1</u>, and <u>EFFECT 2</u>, as well as any **PARAMETER** values they might have.

If you save the CATPARTY **MODULE** into a **PATCH**, you are saving the complete state of the entire **INSTRUMENT**. This includes *all* **PARAMETERS**, as well as **MODULE** selections. Elsewhere in the manual, this is referred to as an **INSTRUMENT PATCH,** and these are the **PATCHES** that can be loaded in the **PLAY** and **PERFORM UIs.** This **PATCH** has its own **MAPPING GROUP** as well, which is where **MAPPINGS** are stored that are only active when the **PATCH** is **LIVE.**

FRAXTAL

The **FRAXTAL INSTRUMENT** is the main **INSTRUMENT** and namesake of the Recursion Studio. It is a digital simulation of video feedback which incorporates some of the mathematics from classical fractals such as the Mandlebrot and Julia sets as well as kaleidoscopic symmetry transformations and other additional artistic effects. It's pretty great.

The following is a diagram of the top level **MODULE** network for the **FRAXTAL INSTRUMENT.**

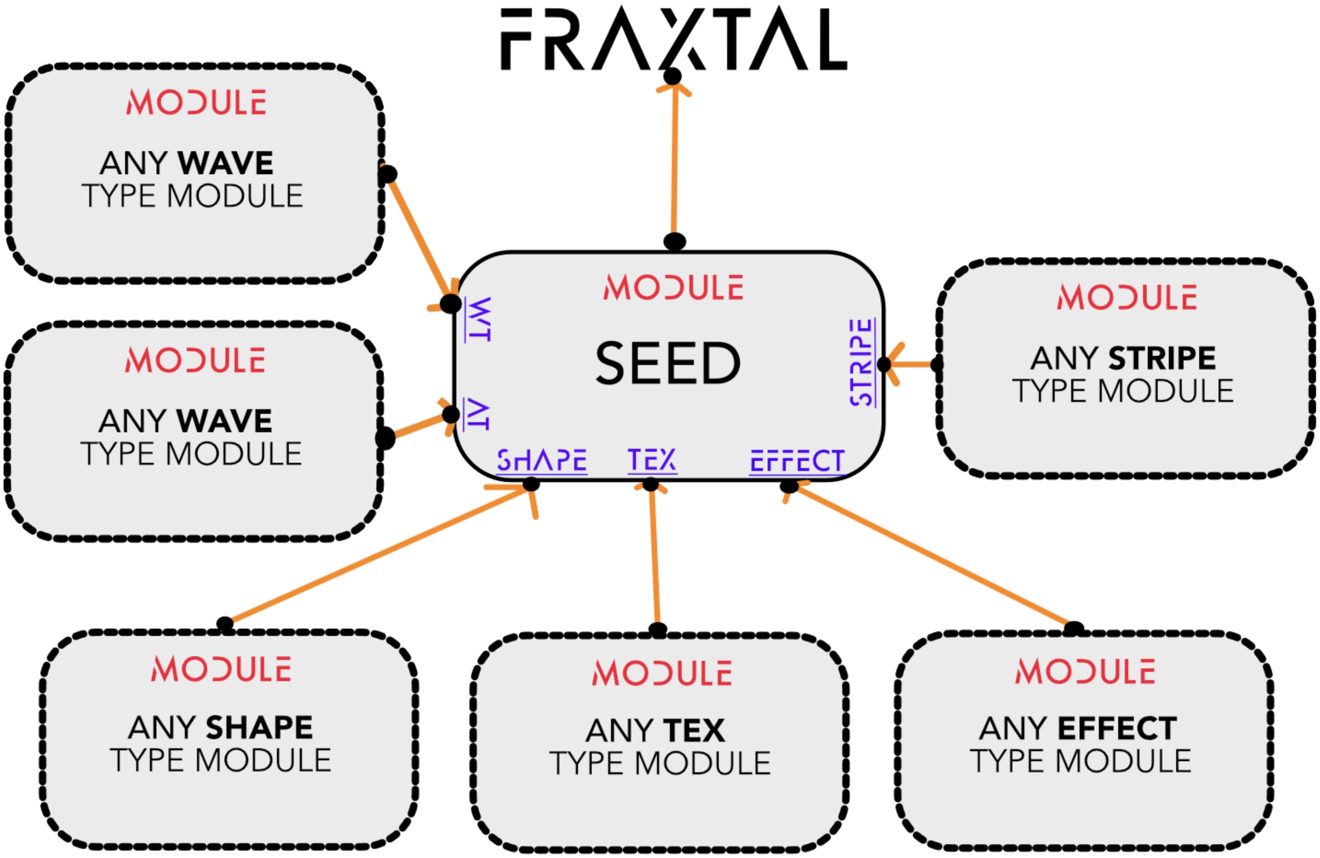


To generate an animation frame, the FRAXTAL **MODULE** first uses the SEED **MODULE** to generate an RGBA image. It then uses the RECURSE **MODULE** to blend the output of the SEED **MODULE** with the previous output(the output one animation frame in the past) of the RECURSE **MODULE,** generating the next animation frame of the RECURSE **MODULE.** This is then put through the COLOR **MODULE** to apply final coloring and lighting effects, and the final result is fed to the FRAXTAL **MODULE'S** output(**HDMI** out).

The COLOR **MODULE** is available in most **INSTRUMENTS** and it is discussed slightly later in the manual**,** along with the MODULATION **MODULE.**

The **MODULE REFERENCE** has additional details about individual **MODULES**.

SEED **MODULE**



The SEED **MODULE** is a procedural RGBA texture generator. Its output is the ‘seed’ content used in the FRAXTAL feedback loop. It has numerous **PARAMETERS** and 6 variable **INPUT MODULES.**

First, the **MODULE** connected to the SHAPE **INPUT** is used to create a 2D shape, defining regions of transparency and opacity. This also generates XY texture coordinates for use by the TEX **MODULE** and a height value used for 3D lighting calculations. Many basic shapes are available.

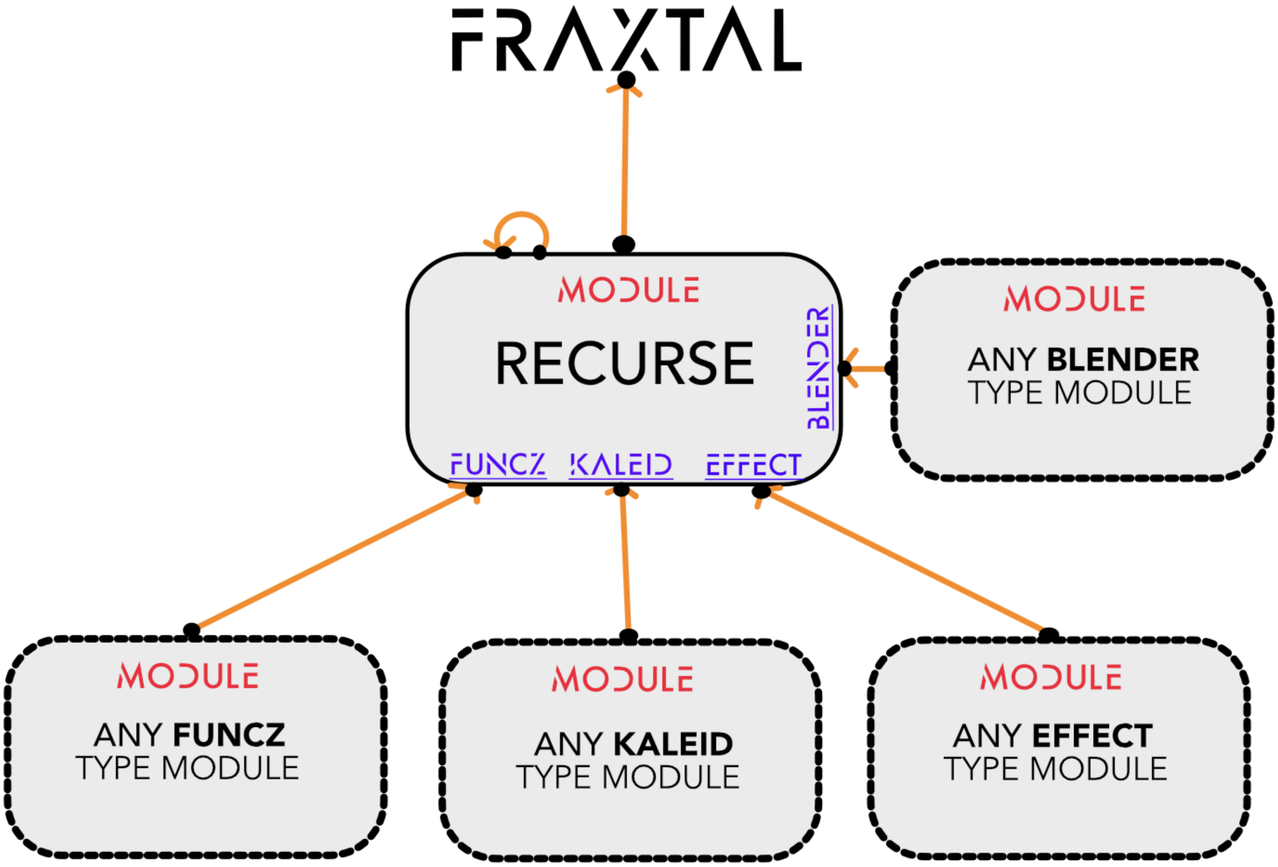
The region generated by SHAPE can be transformed by the WT & AT **MODULES**. Each of these can house a 1D waveform. WT is used to transform the 3D height & shading based effect, from the inside of the shape to its edge. AT is used to add transparency to the opaque regions, from the inside of the shape to its edge

Next, the **MODULE** at the TEX **INPUT** uses these XY texture coordinates to texturize the opaque regions of SHAPE. A large variety of **MODULES** are available, from image & video sources, to procedural algorithms, to wave renderers, to various textures used elsewhere in the **INSTRUMENT.**

This image is then fed into the STRIPE & EFFECT **MODULES.** The STRIPE is a fixed function effect which creates a ‘stripe’ in the SHAPE, a rectangular region of differing color or opacity. The EFFECT is an RGBA->RGBA transformation which can transform the image colors & transparency.

This final image is provided to the **MODULE** output.

RECURSE **MODULE**



The RECURSE **MODULE** implements a digital feedback loop. It exposes a number of mathematical constant **PARAMETERS** for use in the simulation, and has 4 variable **MODULE INPUTS**.

In the first step, the **MODULE** connected to the BLENDER **INPUT** is used to combine the previous RECURSE **MODULE** output with the output of the SEED **MODULE**, which has been provided by the parent FRAXTAL **MODULE**.

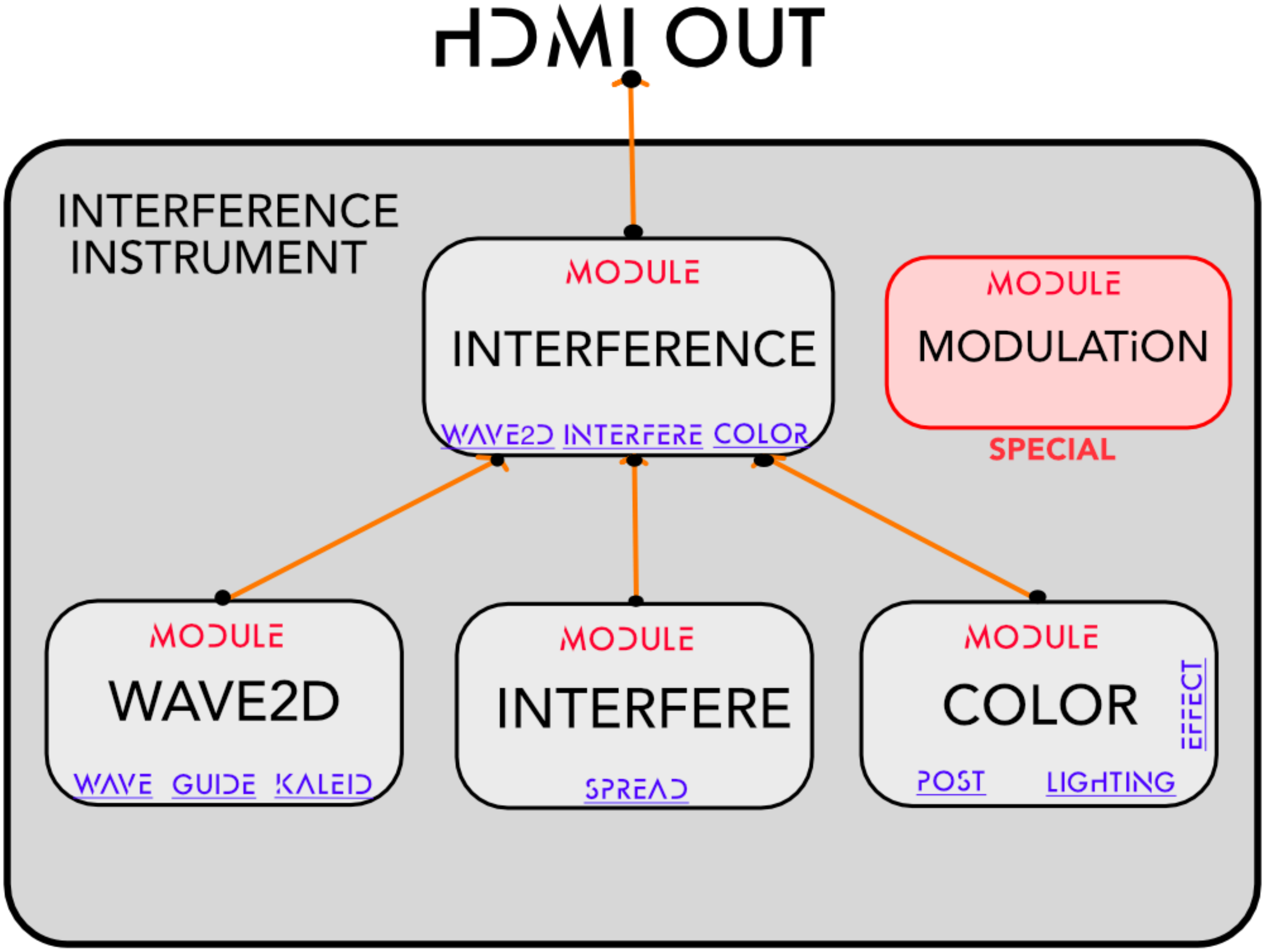
The result is then distorted spatially by the KALEID & FUNCZ **MODULES.** The various KALEID **MODULES** implement a collection of symmetry mappings, while the FUNCZ **MODULES** include a wide variety of mathematical functions. These functions are similar in nature to those used to create the classic Mandelbrot and Julia set fractals.

The final blended and distorted result is then put through the EFFECT **MODULE** (an RGB-RGB effect). It is then passed to the output and is also stored for use in the next animation frame.

INTERFERENCE

The **INTERFERENCE INSTRUMENT** is a 2D plane wave interference simulator. Multiple 2D wavefronts are propagated through space, and peaks and valleys constructively and destructively interfere with each other. The final result is colorized and output to the display.

The following is a diagram of the top level **MODULE** network for the **INTERFERENCE INSTRUMENT**



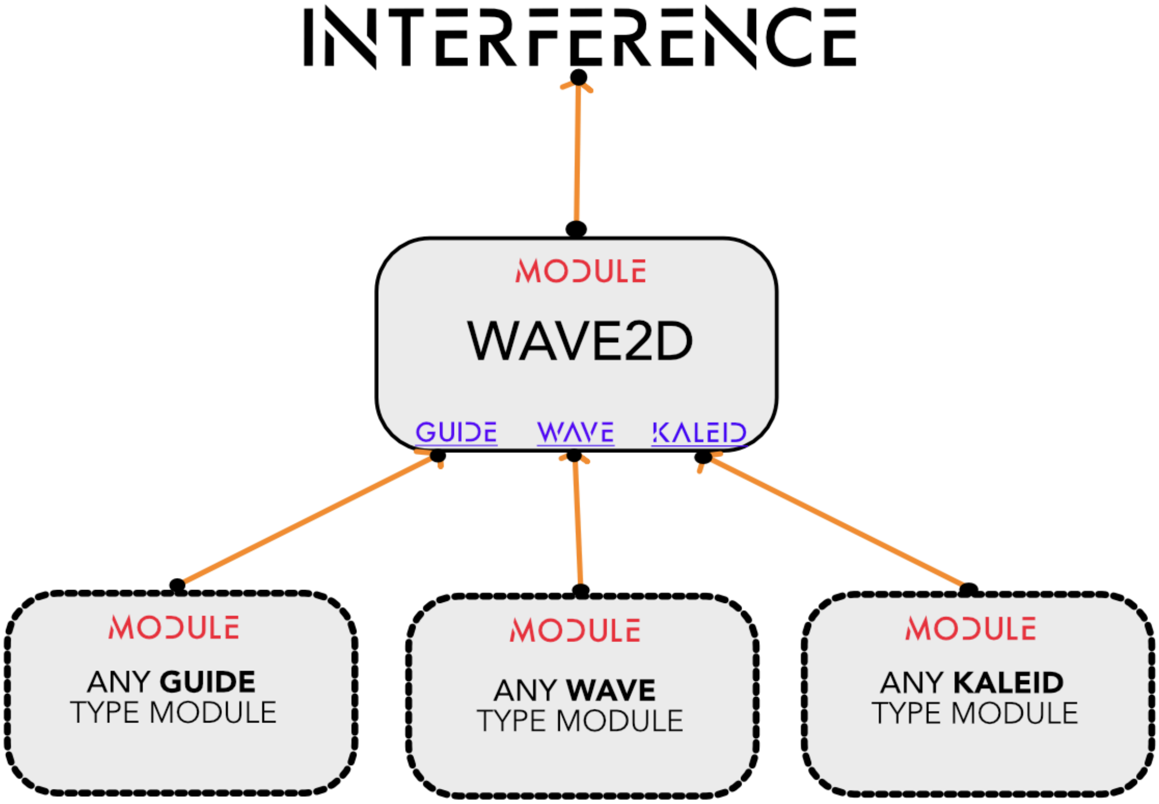
To generate an animation frame, the INTERFERENCE **MODULE** first uses the WAVE2D **MODULE** to generate a 2D monochrome(black/white) wave pattern. This wave pattern is then passed to the INTERFERE **MODULE** which makes several copies of the wave pattern and arranges them in space. The multiple copies of the wave pattern are then summed by the INTERFERE **MODULE**, and put through a basic RGB pallet.

This is then put through the COLOR **MODULE** to apply final coloring and lighting effects, and the final result is fed to the INTERFERENCE **MODULE'S** output(**HDMI** out).

The COLOR **MODULE** is shared among most **INSTRUMENTS**, and it is discussed slightly later in the manual**,** along with the MODULATION **MODULE.**

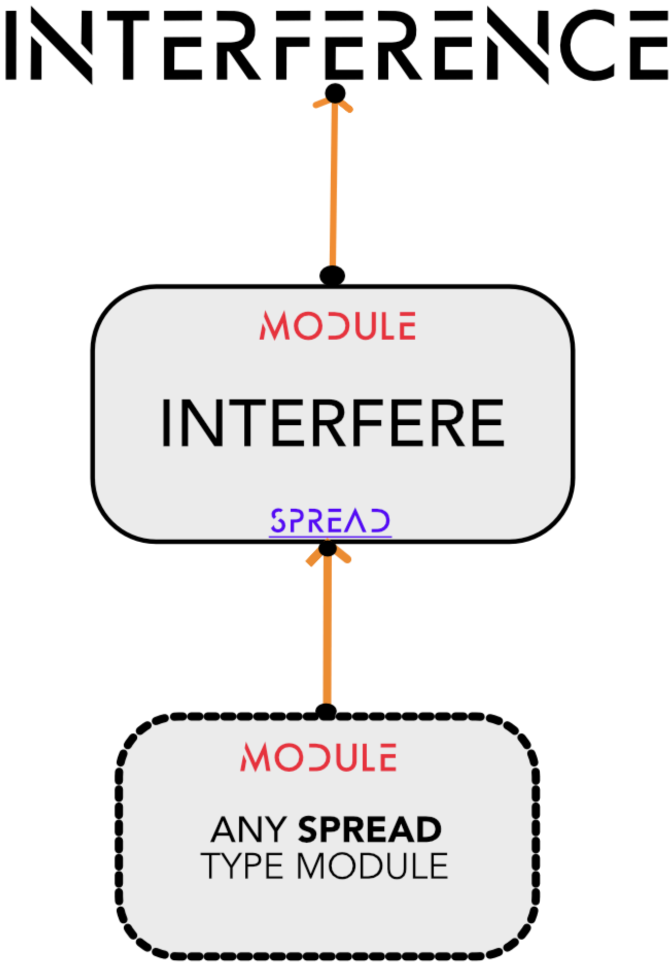
The **MODULE REFERENCE** has additional details about individual **MODULES**.

WAVE2D **MODULE**



The WAVE2D **MODULE** creates a 2D planar waveform. It does this first by using the **MODULE** connected to the GUIDE **INPUT** to create a simple 2D/XY pattern. This pattern is then fed into the WAVE **MODULE**, which allows you to give further variance to the pattern. The WAVE2D **MODULE** provides various **PARAMETERS** that are used while generating the 2D wave pattern, controlling frequency, phase, origin, etc. This wave pattern is then fed through the KALEID **MODULE** to provide an optional additional degree of symmetry.

INTERFERE **MODULE**



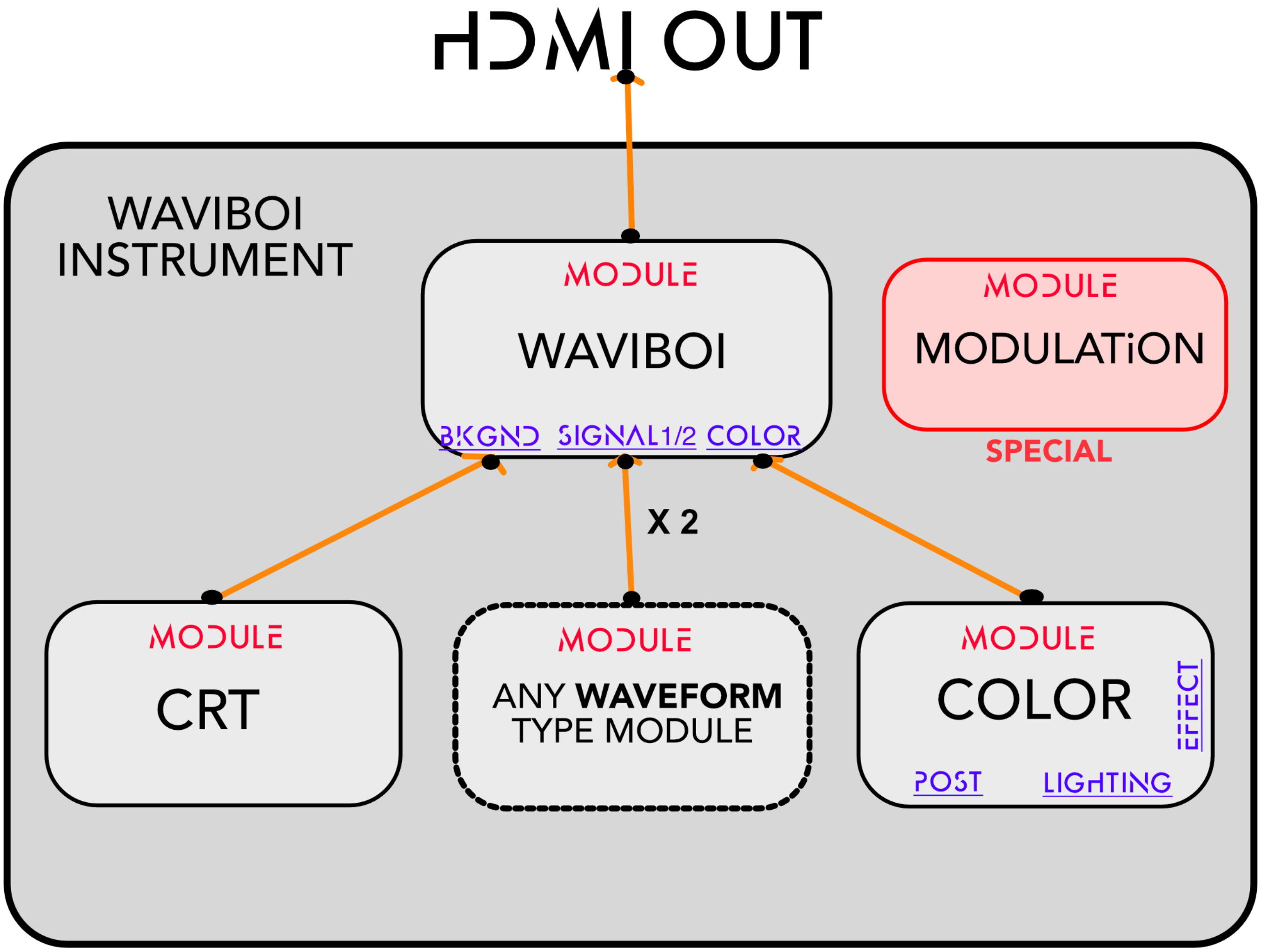
The INTERFERE **MODULE** constructively and destructively interferes multiple copies of the output of the WAVE2D **MODULE.** Anywhere from 1-10 copies of the 2D wave pattern are made, and arranged in 2D space. The **MODULE** connected to the SPREAD **INPUT** specifies how the copies are arranged and move through space.

The INTERFERE **MODULE** also applies a color pallet to the result before passing it to the output, giving the first colorization of the result. Several **PARAMETERS** are control the colorization.

WAVIBOI

The **WAVIBOI INSTRUMENT** is a 2-channel 3D oscilloscope. Each of the two channels can render either 1D, 2D, or 3D curves. The results are both graphed on top of a dynamic/parameterized CRT simulation. As of v1.0 of the Recursion Studio, this **INSTRUMENT** is considered experimental. It's cool enough, but the waveform rendering needs improvement among other things. The **INSTRUMENT** is expected to be improved via a **SYSTEM UPDATE** by Q2/Q3 2024.

The following is a diagram of the top level **MODULE** network for the **WAVIBOI INSTRUMENT**



To generate an animation frame, the WAVIBOI **MODULE** first generates a background image with the CRT **MODULE.** Then the two **MODULES** connected to the SIGNAL1 & SIGNAL 2 **MODULE INPUTS** each generate their waveforms and render them onto the background. This is then put through the COLOR **MODULE** to apply final coloring and lighting effects, and the final result is fed to the WAVIBOI **MODULE'S** output(**HDMI** out).

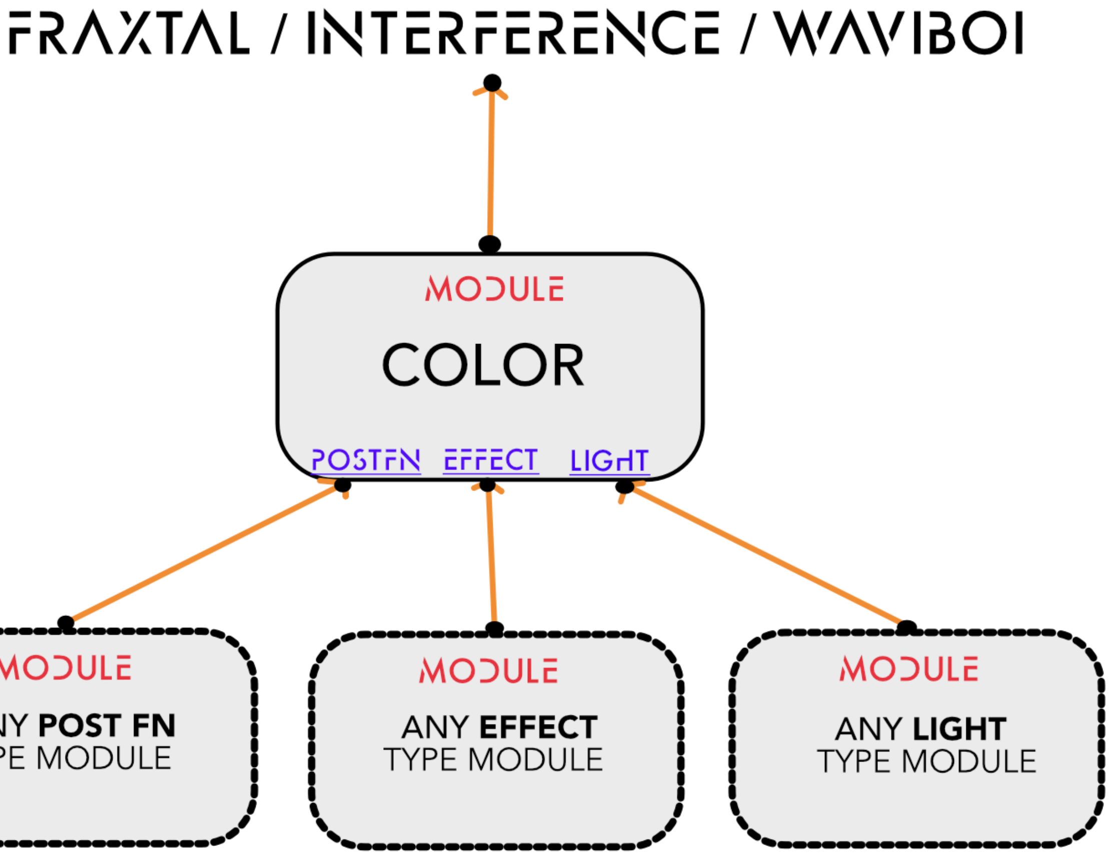
The COLOR **MODULE** is shared among most **INSTRUMENTS**, and it is discussed slightly later in the manual**,** along with the MODULATION **MODULE.**

There are 3 **MODULES** that can be attached to the SIGNAL1/2 **INPUTS**, one for each of 1D, 2D, and 3D waveform renderers, each of which have their own WAVE type **MODULE INPUTS**, for the individual coordinates of the wave. For instance, the WAVEFORM_2D **MODULE** has two **MODULE INPUTS** WAVE_X & WAVE_Y, which specify the x & y coordinates of the waveform to render.

Each of these WAVE coordinates can be an internally generated waveform, or can be controlled via **CV** input, via use of the various **CV** WAVE **MODULES.**

The **MODULE REFERENCE** has additional details about individual **MODULES**.

COLORING & LIGHTING



All **INSTRUMENTS** except for the CATPARTY **INSTRUMENT** use the COLOR **MODULE** as their final stage, to colorize the output and provide simulated lighting. This **MODULE** effectively composes each of its submodules, as well as provides controls for final RGB levels, etc.

First, the input RGB image is put through a POSTFN **MODULE**. These **MODULES** are effectively RGB->RGB effects, except that they each have a guaranteed 'color rotation' **PARAMETER,** and take as an input a 'depth value' provided by the parent **MODULE**. Basically they are fancy effects that are a little more complicated than normal.

Next the result is put through the **MODULE** connected to the EFFECT **INPUT** and then into the LIGHTING **MODULE**. The EFFECT **MODULES** are RGB->RGB effects. There are currently 2 LIGHT **MODULES,** either LIGHT or NO LIGHT. The LIGHT **MODULE** implements a basic ambient/diffuse/specular lighting model, while the NO LIGHT **MODULE** does nothing.

The output of the COLOR **MODULE** is the final processing stage before frames are sent out of the synth.

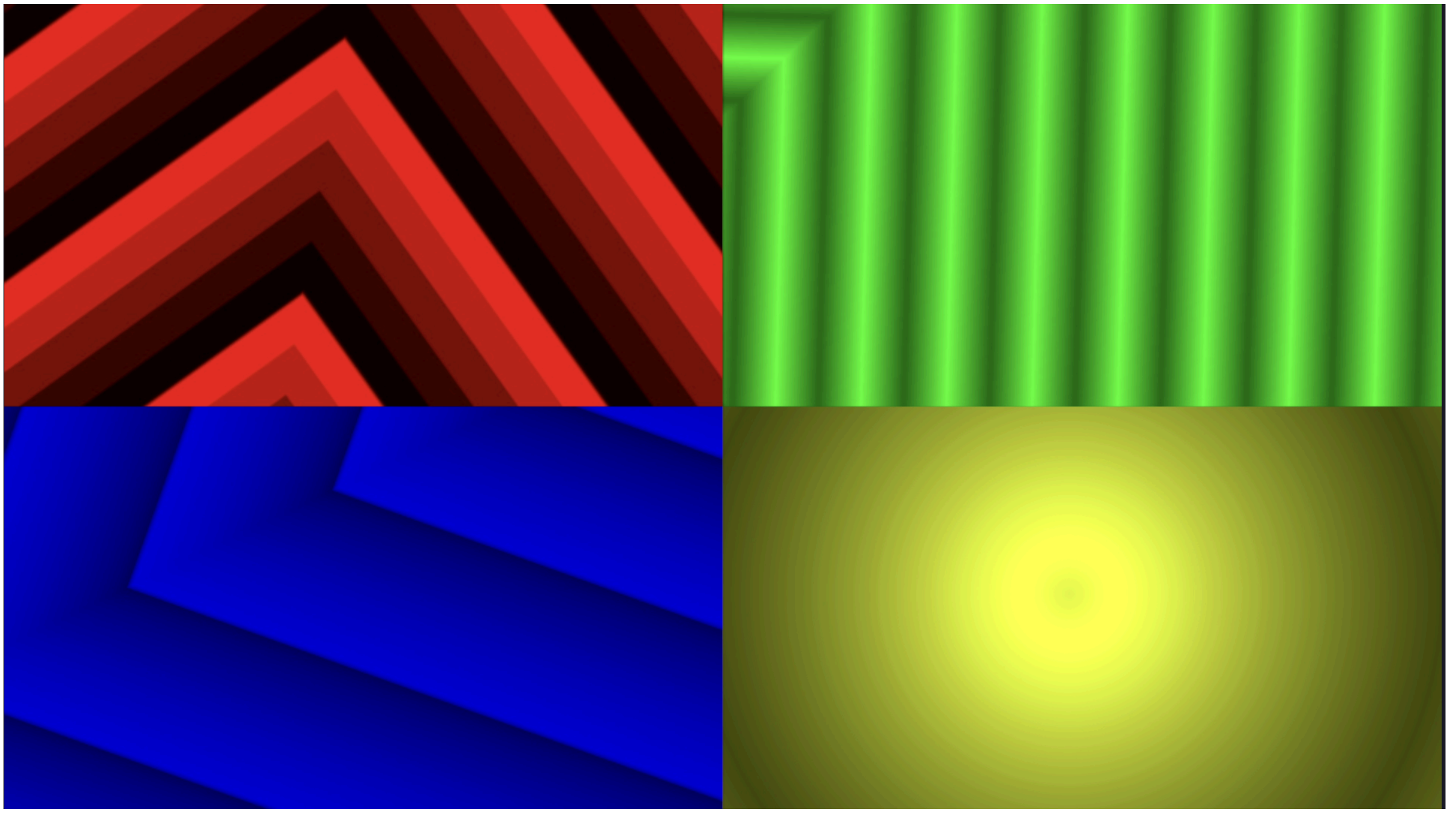
MODULATION

OVERVIEW

The **MODULATION** system in the Recursion Studio is one of its most powerful features. Each **INSTRUMENT** has a MODULATION **MODULE**. It is interacted with via the **CREATE UI**, similar to other **MODULES,** but behaves in a fundamentally different way.

The MODULATION **MODULE** generates 4 1D (black&white) visual patterns, each of which comprise a 'channel' of **MODULATION**. These **MODULATION** patterns are not used directly in the synthesis of the output animation as other **MODULES** are, they are optionally attached to **PARAMETERS**.

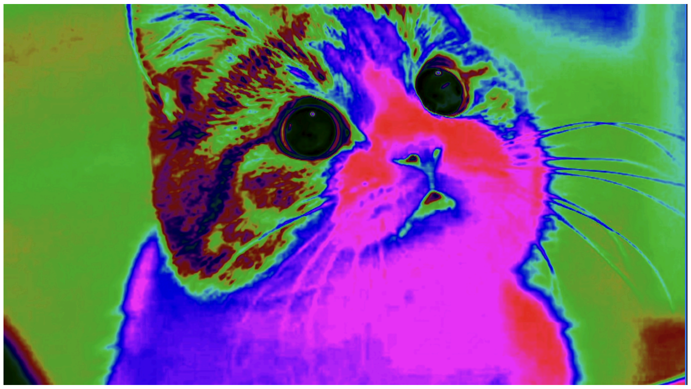
Below is a visual representation of the output of the MODULATION **MODULE,** with each quadrant being one of the 4 channels/patterns. They are colored here and throughout the synth simply for ease of visual identification. The coloring has no actual significance, the signals are inherently 1 dimensional values from 0 to 1.

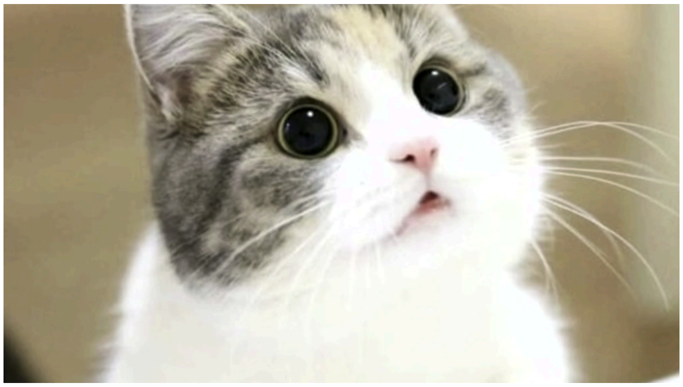


Many **MODULES** have basic 1D continuous **PARAMETERS**. These **PARAMETERS** have numerical values that are set by **KNOBS** and other **MAPPINGS.** The **MODULATION** system allows you to let the value of a **PARAMETER** *vary across the screen.* Instead of a **PARAMETER** having a single numerical value, its value can change depending on the screen location. The **MODULATION** patterns are this variation.

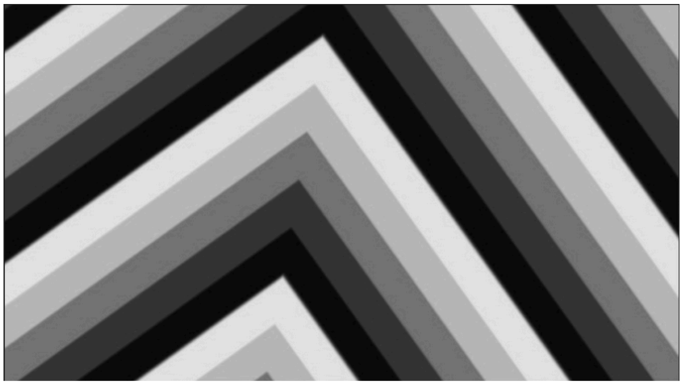
We will use the default **PATCH** for the **CATPARTY INSTRUMENT** as an example. This assumes a basic level of familiarity with the **CATPARTY INSTRUMENT** and the **CREATE UI**.

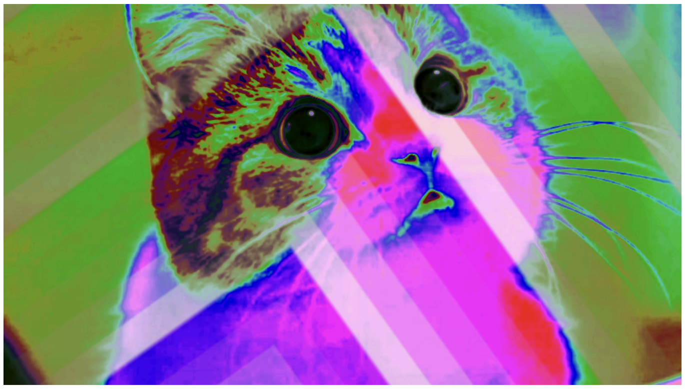
The default output of the **CATPATY INSTRUMENT** is shown below on the left. The output of the CAT **MODULE** is processed via the PARTY **MODULE**. By default, the PARTY **MODULE** has its EFFECT 1 **INPUT** set to the RAINBOW **EFFECT,** with the "effect1 amt" **PARAMETER** set to 1.0. If we turn the "effect1 amt" **PARAMETER** down to 0.0, the **EFFECT** is practically disabled, and the **INSTRUMENT'S** output will be the unprocessed output of the CAT **MODULE** as shown below on the right.





Setting "effect1 amt" to some value between 0 and 1 will activate RAINBOW **EFFECT** to that degree. We now use **MODULATION** pattern 1, shown below on the left to modulate the "effect1 amt" **PARAMETER**. The result is shown below on the right





The value in **MODULATION** pattern 1 now determines how the "effect1 amt" **PARAMETER'S** value varies across the screen. In the regions where **MODULATION** pattern 1 is black the value of the "effect1 amt" **PARAMETER** is set to 0, and we see that no effect is applied. As the **MODULATION** pattern moves toward white, we see that the "effect1 amt" **PARAMETER'S** value increases and we see progressively more of the RAINBOW **EFFECT** applied.

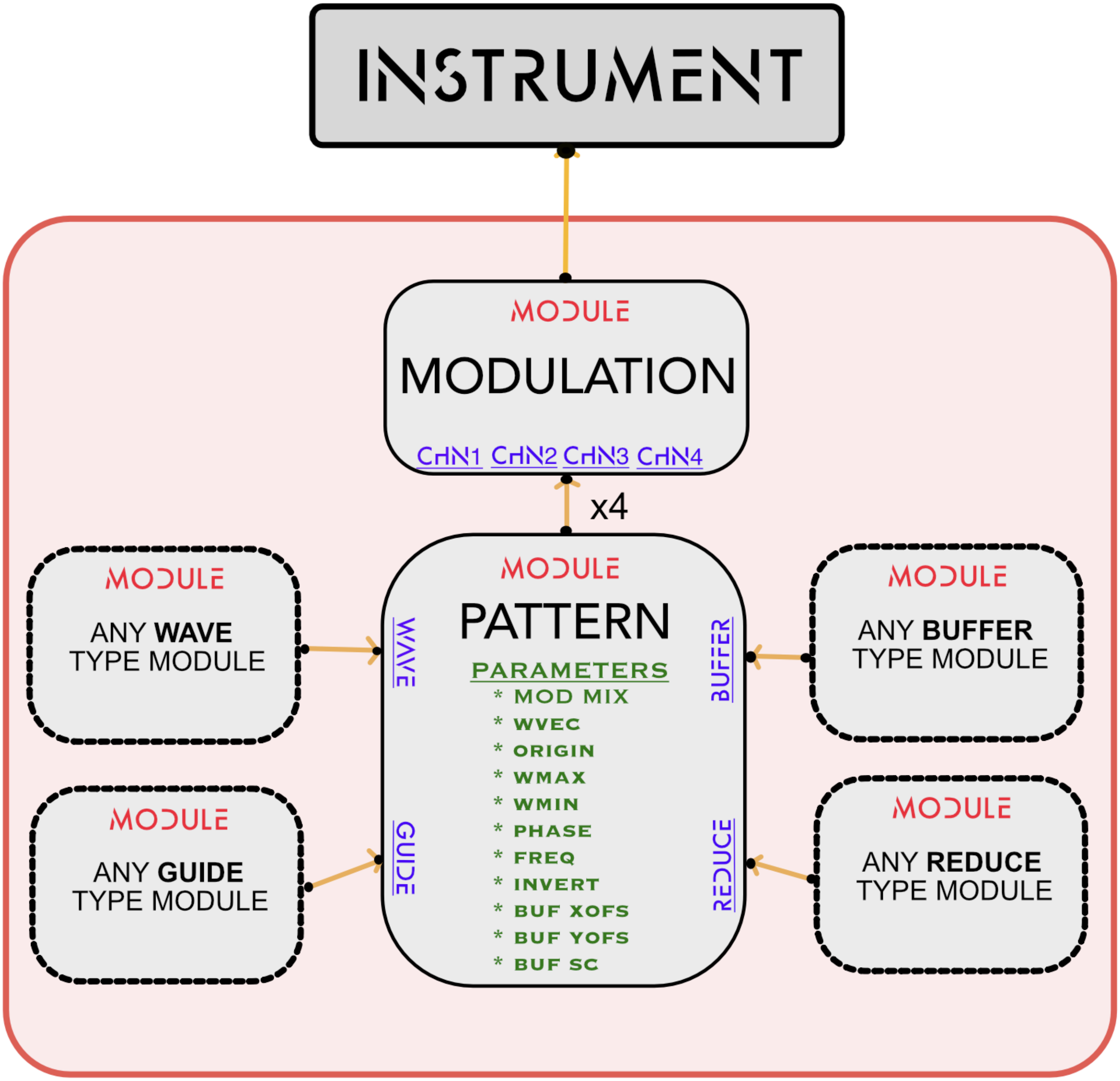
USING MODULATION

A **PARAMETER** can be **MODULATED** via its corresponding **PARAMETER INFO** page, via any of 3 **MODULATION METHODS**. The **METHOD** controls how the **PARAMETER'S** value is combined with the **MODULATION** pattern, and each is useful in different circumstances. The available **MODULATION METHODS** are:

```
SCALE: Final value = (PAR VALUE) * (MODULATION PATTERN)
OFS:   Final value = (1.0 - PAR VALUE) * (MODULATION PATTERN) + (PAR VALUE)
ADD:   Final value = (PAR VALUE) + (MODULATION PATTERN)
```

MODULE NETWORK

The following is a diagram of the **MODULE** network for the MODULATION **MODULE**



The main MODULATION **MODULE** is simply a housing for four PATTERN **MODULES**, each of which generates one of the 4 **MODULATION** patterns. The PATTERN **MODULES** each have numerous **PARAMETERS** and 4 **MODULE INPUTS**. The general idea is that the PATTERN is created as a combination of generative waveforms and RGB image buffers. Each PATTERN has both sources, and the MOD MIX **PARAMETER** blends between them.

The WAVE and GUIDE **MODULES** outputs are composed together to create a 2D wave based pattern. The GUIDE **MODULE** creates a basic 2D pattern. This is then fed into the WAVE **MODULE** to provide additional variance. The BUFFER **MODULE** streams RGB images generated by some other **MODULE** elsewhere in the synth. The REDUCE **MODULE** then converts the RGB BUFFER **MODULE** to monochrome. The result is combined and the end result is a **MODULATION** pattern.

Further details are available in the **MODULE REFERENCE.**

MODULATION PATCHES

PATCHES that are created by saving the **MODULATION MODULE** are slightly different from regular patches. Not only do they store the **MODULE** network's state as a normal **PATCH** does, the **PATCHES** also store any **MODULATION** assignments that have been made elsewhere in the **INSTRUMENT.**

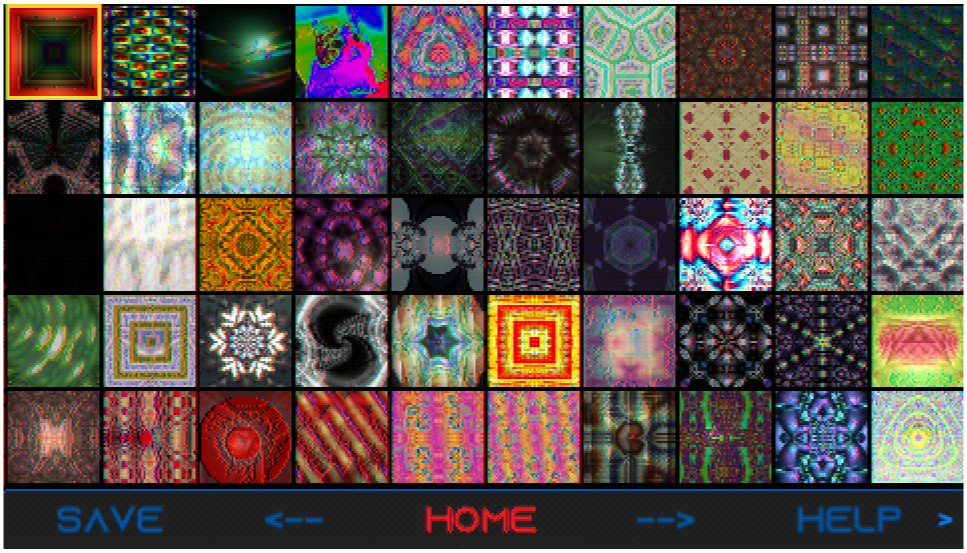
Related, when a **MODULATION PATCH** is loaded, all existing **MODULATION** is first removed from the **INSTRUMENT.** This means that only the **MODULATION** that was stored in the loaded **PATCH** will be active after loading.

Saving & loading **MODULATION PATCHES** entirely controls what **MODULATION** is active.

PLAY UI

OVERVIEW

The **PLAY UI** is designed for experimentation and exploration without needing any deep understanding of the synth's internals. The **PLAY UI** is arranged as a **PATCH** selection grid with a **COMMAND BAR** underneath. Hold down the **SHIFT** button to expose additional **COMMANDS.**



CONTROLS

Every image on the screen represents a **PATCH** for an **INSTRUMENT** that can be loaded. **PATCHES** can be loaded by simply tapping on them, or with the **GOLD KNOB & CURSOR.** Turn the **GOLD KNOB** to move the **GOLD CURSOR**. Press the **GOLD KNOB** down to load the currently selected **PATCH.**

The **PATCH** selection grid can possibly have additional pages. Use the arrow **COMMANDS** to switch pages, or scroll the **GOLD CURSOR** off of the current page in either direction. Additional, swiping touch gestures are supported.

While in the **PLAY UI**, the 5 **BLUE KNOBS** will be bound to the first 5 **MACRO MAPPINGS** for the **LIVE INSTRUMENT PATCH.** If the **LIVE PATCH** does not have any **MAPPINGS** stored with it, the **MAPPINGS** for the **DEFAULT PATCH** for the currently live **INSTRUMENT** will be used instead.

Pushing one of the 5 **BLUE KNOBS** down will reset the **PARAMETER** that the **KNOB** is actively controlling to its default value.

The 5 **LARGE BUTTONS** similarly are bound to the first 5 **BUTTON MAPPINGS** for the **LIVE PATCH.**

While the **SHIFT BUTTON** is held, the 5 **BLUE KNOBS,** and **LARGE BUTTONS**, will temporarily be bound to the second page of their respective **MAPPINGS**, if available. Holding the **SHIFT BUTTON** will also make the button press from the **GOLD KNOB** load the default **PATCH** for the currently live **INSTRUMENT.**

While the **NAV BUTTON** is held the 5 **BLUE KNOBS** and **LARGE BUTTONS** will temporarily be bound to the respective **MAPPINGS** for the default **PATCH** for the currently live **INSTRUMENT.** Holding the **NAV BUTTON** will also make the button press from the **GOLD KNOB** reload the currently **LIVE PATCH** from disk.

COMMANDS

SAVE: Allows you to save a **PATCH** for recall later. A new **PATCH** will be created, creating a complete snapshot of what is currently live. It will not affect either the current live **PATCH,** or what is selected with the **GOLD CURSOR**, a fundamentally new **PATCH** will be created.
HOME: Navigate to the **HOME** page.
HELP: Show help.
INFO: Navigate to the **INFO** page for the **PATCH** that is currently selected by the **GOLD CURSOR.**
LIBRARY: Navigate to the **LIBRARY** page.
RANDOM: Randomly generate a new **PATCH**.
SCREENSAVER: Start or stop the **SCREENSAVER.** This **COMMAND** will be highlighted if it is currently active.

SETTINGS

Various **SETTINGS** are available for the **PLAY UI**, available from the **PLAY UI** section of the **SETTINGS** page. The **SETTINGS** page can be accessed via the **COMMAND BAR** of the **HOME** page, or via the **NAV BUTTON.**

Auto Map I/O: Enables or disables **AUTO MAPPING** while in the **PLAY UI.**
Auto Lock: Completely disables all navigation upon entering the **PLAY UI.** You will then need to power cycle the synth in order to exit the **PLAY UI**. This feature is useful for leaving the synth unattended, such that the users are unable to modify settings or make other permanent changes. However new **PATCHES** can still be created.
Screensaver Autostart: When this value is non-zero and the synth has been inactive for the specified period of time, the **SCREENSAVER** will start.

SCREENSAVER / AUTOMATION

OVERVIEW

When enabled, the **SCREENSAVER** periodically changes the live **PATCH.** It will regularly either randomly load an existing **PATCH,** or generate a new one.

NOTES:

- The **SCREENSAVER** will not switch **INSTRUMENTS.** This is a known limitation and will be rectified in a future **SYSTEM UPDATE.**
- When loading existing **PATCHES**, the **SCREENSAVER** will respect any **TAG** filters that are currently enabled in the **LIBRARY**.
- The **SCREENSAVER** does a pretty decent job of making new **PATCHES**, but sometimes it fails miserably. It can spaz out pretty seriously.

TIPS: The screensaver works really best in conjunction with a small degree of human interaction. It's constantly creating new content, but sometimes it doesn't look as great as it could, as it is random. However, it's normally not that hard to tweak something that doesn't look great into something that does. The first few **MACRO** controls(5 **BLUE KNOBS)** are normally worth experimenting with. And you can always just 'skip' by loading a new **PATCH** or hitting **RANDOM,** if it's just not working well. However, in order for you not to turn the **SCREENSAVER** off by interacting with the synth, make sure that "Halt on Interrupt" is off in the **SETTINGS.** It's off by default.

SETTINGS

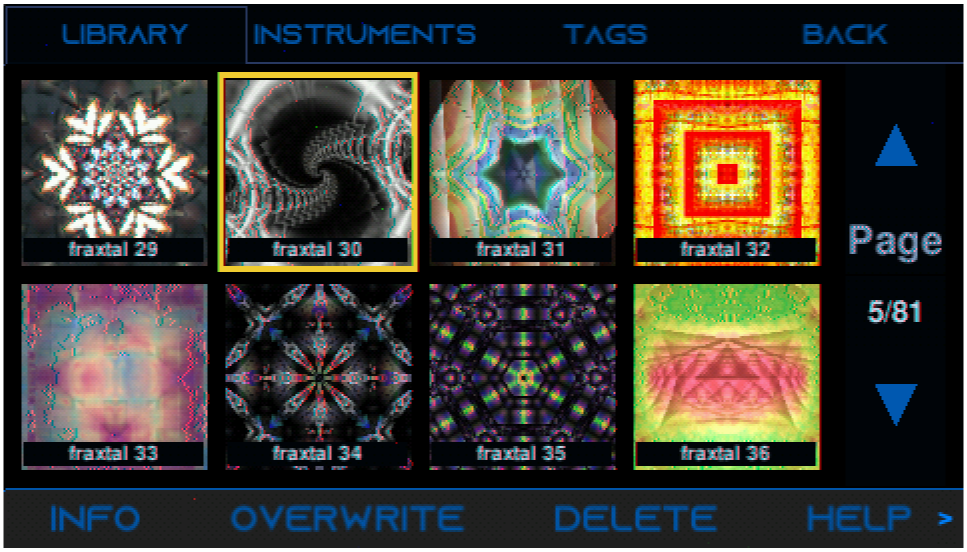
Various **SETTINGS** are available for the **SCREENSAVER** available from the **SCREENSAVER** section of the **SETTINGS** page. The **SETTINGS** page can be accessed via the **COMMAND BAR** of the **HOME** page, or via the **NAV BUTTON.**

Enabled: Whether the **SCREENSAVER** is currently active or inactive.
Interval: How often the **SCREENSAVER** loads or creates a new **PATCH.** This value isn't exactly directly used, the actual interval is randomized around the provided value.
Transition Spd: The amount of time it takes to transition from the current **PATCH** to the next **PATCH**.
Randomness: When a new **PATCH** is created, it uses the current **LIVE PATCH** as a starting point**.** This control specifies how far from the current **PATCH** the **SCREENSAVER** will deviate.
Library Load Probability: How likely it is that the **SCREENSAVER** will load an existing **PATCH** vs create a new **PATCH.**
Auto Map I/O: Whether **AUTO MAPPING** is enabled while the **SCREENSAVER** is running.
Halt on Interrupt: If enabled, the **SCREENSAVER** will stop as soon as somebody interacts with the synth by modifying the current live **PATCH.**
Disable in Create: If enabled, the **SCREENSAVER** will automatically turn off when entering the **CREATE UI**.

LIBRARY

OVERVIEW

The **LIBRARY** page is where you manage **PATCHES.** It is organized as a grid of images, each representing a **PATCH.** To the right are pagination controls. At the bottom of the page is the **COMMAND BAR**, which presents available **COMMANDS.** Hold down the **SHIFT** button to expose additional **COMMANDS.** In addition to the **COMMAND BAR** at the bottom of the **LCD,** there are tabs at the top for selecting the additional**, INSTRUMENTS,** and **TAGS** pages, which are used for **FILTERING PATCHES.**



CONTROLS

Every image on the screen represents a **PATCH** for an **INSTRUMENT** that can be managed. The **GOLD CURSOR** highlights the **PATCH** that is currently selected. The selected **PATCH** can be changed by tapping on the **LCD,** or via the **GOLD KNOB**. On the **LIBRARY** page, selecting a **PATCH** *does not* load it, it only marks it for editing and management. However the currently selected **PATCH** can be loaded by clicking the **GOLD KNOB.**

In addition to the **COMMANDS** available in the **COMMAND BAR**, additional details and **COMMANDS** for the selected **PATCH** can be found by navigating to the **PATCH INFO** page, via the **INFO COMMAND.**

The **PATCH** selection grid can possibly have additional pages. Use the arrows to switch pages, or scroll the **GOLD CURSOR** off of the current page in either direction. Additional, swiping touch gestures are supported.

The **LIBRARY** can be accessed via **COMMANDS** in the **PLAY** and **PERFORM UIs** and by tapping on the upper left “CURRENT PATCH” widget in the **CREATE UI.**

While in the **LIBRARY,** the 5 **BLUE KNOBS** and **LARGE BUTTONS** will behave as they were prior to you entering the **LIBRARY.**

COMMANDS

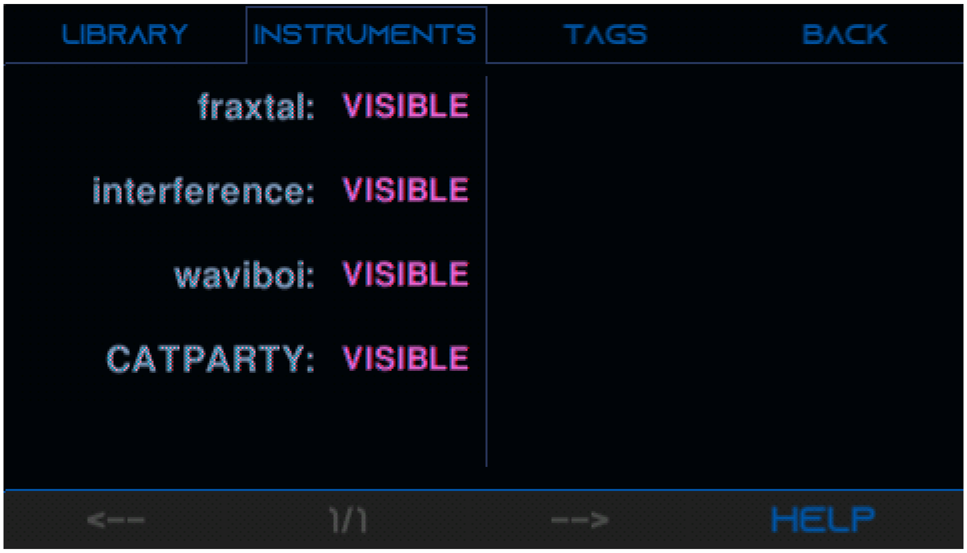
INFO: Navigate to the **INFO** page for the **PATCH** that is currently selected by the **GOLD CURSOR.**
OVERWRITE: Overwrites the **PATCH** that is currently selected by the **GOLD CURSOR** with what is currently **LIVE**.
DELETE: Deletes the **PATCH** that is currently selected by the **GOLD CURSOR**.
HELP: Show help.
SORT ORDER: Cycles through the various available ordering options of **PATCHES**. Affects the **PERFORM, PLAY**, and **CREATE UIs** as well.

LIBRARY FILTERING

The **LIBRARY** page is also where you set the current active **LIBRARY FILTERS**, which let you limit the **PATCHES** that are shown on the **LIBRARY/PLAY/PERFORM/CREATE UI** pages to some subset of all that are available.

INSTRUMENTS

The **INSTRUMENTS** tab of the **LIBRARY** lets you **FILTER PATCHES** created with certain **INSTRUMENTS**. It simply presents a list of each available **INSTRUMENT**, along with its visibility status. **INSTRUMENTS** can be toggled as visible or not simply by tapping them.



fraxtal: VISIBLE

interference: VISIBLE

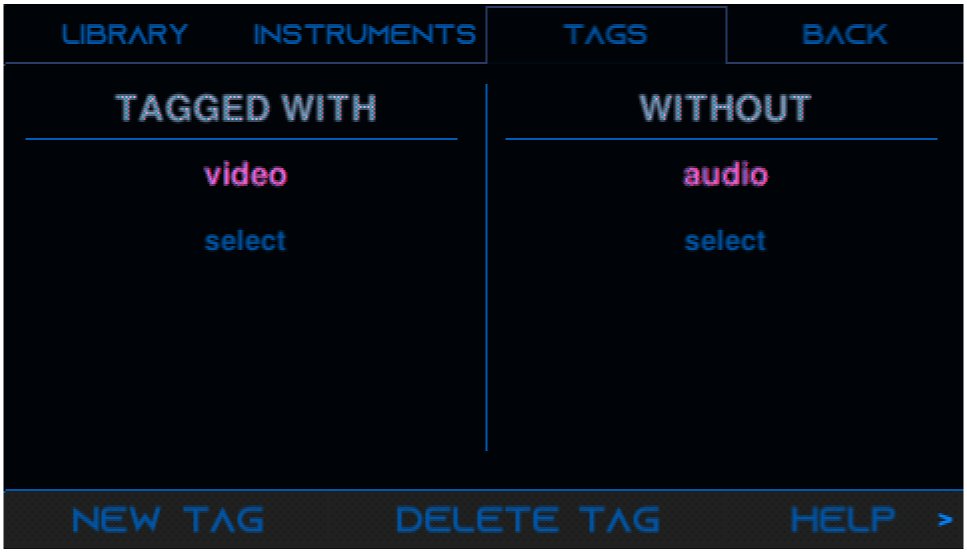
waviboi: VISIBLE

CATPARTY: VISIBLE

If you are only interested in seeing **PATCHES** for one specific **INSTRUMENT,** for instance, simply toggle all other **INSTRUMENTS** to 'HIDDEN,' and then in the **LIBRARY/PLAY/PERFORM/CREATE UI** you will only see **PATCHES** for this specific instrument.

TAGS

The **TAGS** tab of the **LIBRARY** lets you **FILTER** out **PATCHES** that have either been **TAGGED** with certain **TAGS,** or have not been **TAGGED** with certain **TAGS.** You can also create and delete **TAGS** here. The **TAGS** selected here control which **PATCHES** are visible on the **LIBRARY/PLAY/PERFORM/CREATE UI** pages.



TAGGED WITH	WITHOUT
video	audio
select	select

The page is divided into 2 columns, the 'TAGGED WITH' column, and the 'WITHOUT' column. A **TAG** can be added to either column by tapping 'select**.'** A **TAG** can then be removed from the column by simply tapping on it.

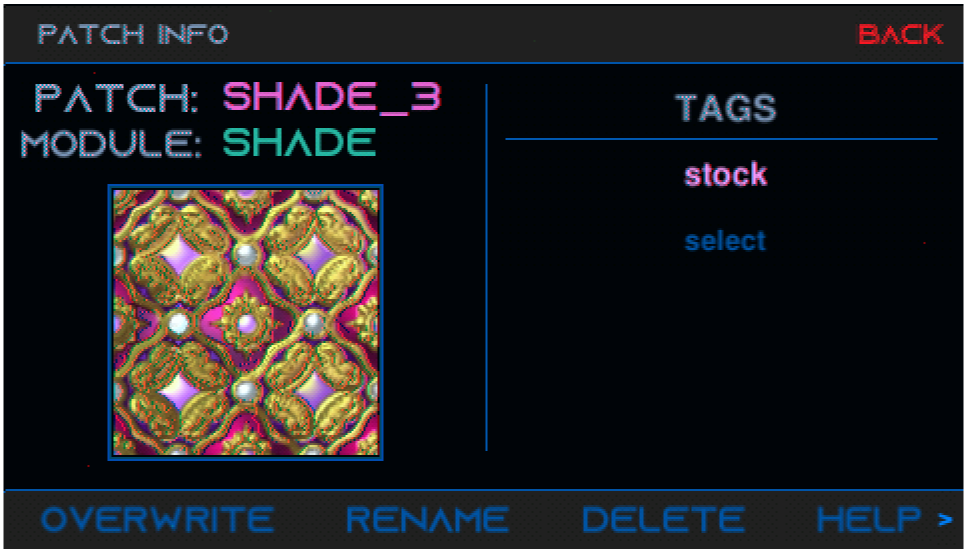
If a **TAG** is added to the 'TAGGED WITH' column, only **PATCHES** that have the specified **TAG** will be visible on the **LIBRARY/PLAY/PERFORM/CREATE UI** pages. If a **TAG** is added to the 'WITHOUT' column, only **PATCHES** that *do not* have the specified **TAG** will be visible.

COMMANDS

<u>**NEW TAG**</u>**:** Allows for the creation of new **TAGS.**
<u>**DELETE TAG:**</u> Allows for the deletion of existing **TAGS.**
<u>**FILTER ALL PATCHES:**</u> By default, **TAG FILTERS** only apply to **PATCHES** that have been saved for **INSTRUMENTS.** However in the **CREATE UI**, you may want to **FILTER PATCHES** that are created for other **MODULES** in the network. This **COMMAND** allows you to do that.

PATCH INFO

The **PATCH INFO** page allows you to manage a specific **PATCH** and to control the **TAGS** that it has assigned. On the left of the **LCD** is the **PATCH's** name, the **MODULE** or **INSTRUMENT** the **PATCH** has been saved from, and a small preview image that has been generated. On the right of the **LCD** is a widget that shows the **TAGS** associated with the **PATCH**, and allows them to be modified.



PATCH INFO

PATCH: SHADE_3
MODULE: SHADE

TAGS
stock
select

CONTROLS

To add a **TAG**, tap on 'select' and select from the available **TAGS**. To remove a **TAG** simply tap on it.

NOTE: Some **TAGS** can't be removed, such as 'stock', 'default', 'video', etc, as they are generated internally by the synth. These will have a slightly lighter color.

MORE NOTE: Some **PATCHES** simply can not be modified in any way. These patches are necessary for the proper functioning of the synth, and all contain the 'default' **TAG.**

COMMANDS

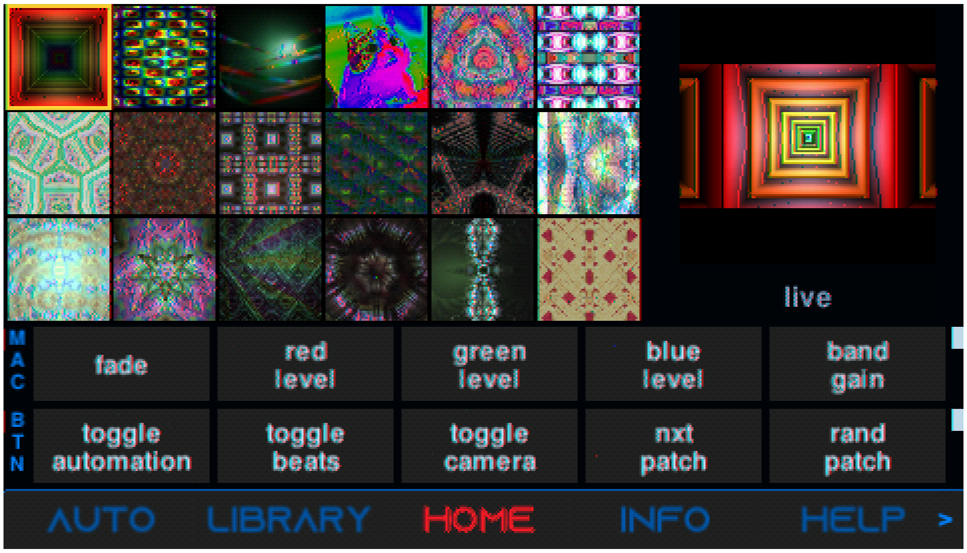
<u>**OVERWRITE:**</u> Overwrites the **PATCH** that is being viewed with what is currently **LIVE**.
<u>**RENAME**</u>**:** Change the name of the **PATCH** that is being viewed
<u>**DELETE:**</u> Deletes the **PATCH** that is being viewed**.**

HELP: Show help.
NEW TAG: Allows for the creation of new **TAGS.**
LOAD IN CREATE: Loads the **PATCH** into the **CREATE UI.**

PERFORM UI

OVERVIEW

The **PERFORM UI** is intended for use live during a concert, event, or other performance, and has various controls, and **COMMANDS** to aid in this. The **PERFORM UI** is organized into a **PATCH** selection grid in the upper left, a preview window in the upper right, two horizontal bars displaying the current **MACRO** and **BUTTON MAPPINGS**, and finally a **COMMAND BAR** at the bottom.



CONTROLS

The **PATCH** selector is a grid of images representing the available **PATCHES**. They can be selected by moving the **GOLD CURSOR** with the **GOLD KNOB** or by touching an image. Once selected, the preview image on the right changes to one of the selected **PATCH.** The currently selected **PATCH** can be loaded by pressing the **GOLD KNOB.** In the **PERFORM UI**, touch loading is disabled in order to prevent mistakes while performing.

The preview on the right side of the screen will either be a live preview of the synth's current output or of the selected **PATCH**. If the preview is showing the selected **PATCH**, tap on it to get a live preview. If the preview is a live view of the output, tap it to get a full screen view. Tap again to revert.

The horizontal bar with the **MAC** label on the left shows the 5 **MACRO** controls that are currently bound to the 5 **BLUE KNOBS.** By default in the **PERFORM UI**, the bound **MACROS** come from the **GLOBAL MAPPING GROUP.**

There can be up to 3 pages(for 15 total) **MACRO** controls in any **MAPPING GROUP,** and the position of the small white rectangle to the right of the horizontal **MACRO** bar denotes which page is currently active. The page can be changed by swiping left or right on the **MACRO** bar.

The horizontal bar with the **BTN** label on the left shows the 5 **BUTTON MAPPINGS** that are currently bound to the 5 **LARGE BUTTONS.** Similarly, these come from the **GLOBAL MAPPING GROUP,** and can have up to 3 pages.

While in the **PERFORM UI,** holding down the **SHIFT BUTTON** will swap out the **MAPPINGS** of the 5 **BLUE KNOBS** and 5 **LARGE BUTTONS** from the **GLOBAL MAPPING GROUP** to the **MAPPINGS** saved with the current **LIVE PATCH.** If the current **PATCH** has no explicit **MAPPINGS** saved, the **MAPPINGS** from the default **PATCH** for the current **INSTRUMENT** will be used. While holding **SHIFT**, the **MACRO** and **BUTTON** bars can also be swiped to access additional pages.

COMMANDS

SAVE: Save a new **PATCH** for recall later. Does not affect the currently selected **PATCH.**
AUTO: Access settings for the **SCREENSAVER** for completely autonomous operation.
HOME: Return to the **HOME** page.
LIBRARY: Access the **LIBRARY**, mainly for **FILTERING** the available **PATCHES**.
HELP: Show help.
INFO: Navigate to the **INFO** page for the **PATCH** that is currently selected by the **GOLD CURSOR.**
I/O: Navigate to the **MAPPINGS LIST** page for the **GLOBAL MAPPING GROUP.**
CLEAR: Sometimes your shit might be spazzing. This clears everything briefly and might help.
SETTINGS: Navigates to the **SETTINGS** page for the **PERFORM UI**

SETTINGS

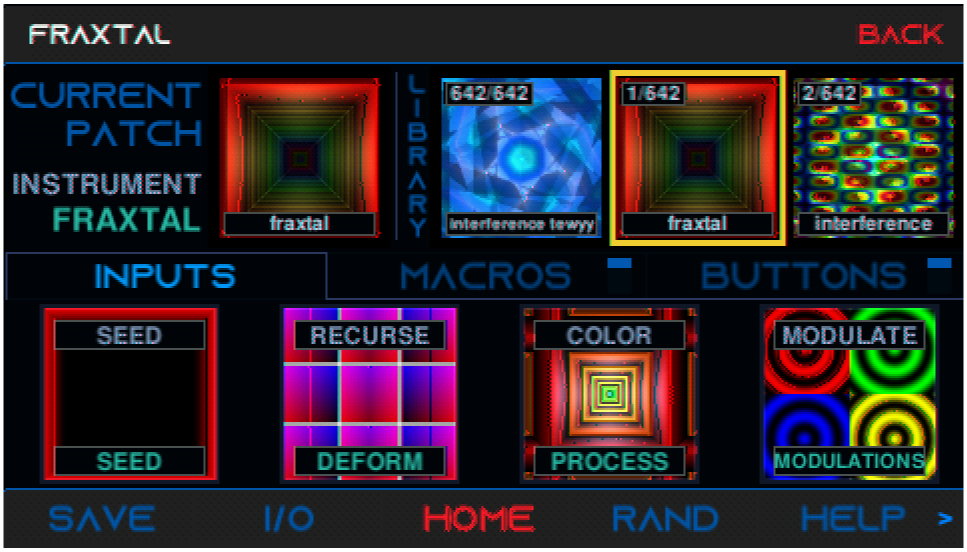
Various **SETTINGS** are available for the **PERFORM UI**, available from the **PERFORM UI** section of the **SETTINGS** page. The **SETTINGS** page can be accessed via the **COMMAND BAR** of the **HOME** page, or via the **NAV BUTTON,** or directly via the **COMMAND** in the **PERFORM UI.**

Auto Map I/O: Enables or disables **AUTO MAPPING** while in the **PERFORM UI.**
Macro Default: Determines which **MAPPING GROUP** is active in the **PERFORM UI** by default. If this is set to 'Global', then the **GLOBAL MAPPING GROUP** will be active by default on the **BLUE KNOBS** and **LARGE BUTTONS**, and the **SHIFT BUTTON** will load the **MAPPINGS** for the live **PATCH**. If this setting is 'Patch,' then this will be reversed.
Macro Sensitivity: An additional sensitivity factor for all **MACROS** that is active while in the **PERFORM** page. This helps avoid jittery controls while live. Set this value to 1.0 and the **MACRO** sensitivity will be identical to elsewhere in the **UI**. If this factor is 0.5, the **KNOB** will have to be moved 2x as much on the **PERFORM UI** for the same **PARAMETER** change**.**
Macro Atk Scale: An additional smoothing factor for all **MACROS** that is active while in the **PERFORM** page. This also helps avoid jittery controls while live. When a **KNOB** is turned, it takes some amount of time to move the associated **PARAMETER** to its new value. This Atk Scale increases the amount of time the **PARAMETER** transition takes.

CREATE UI

OVERVIEW

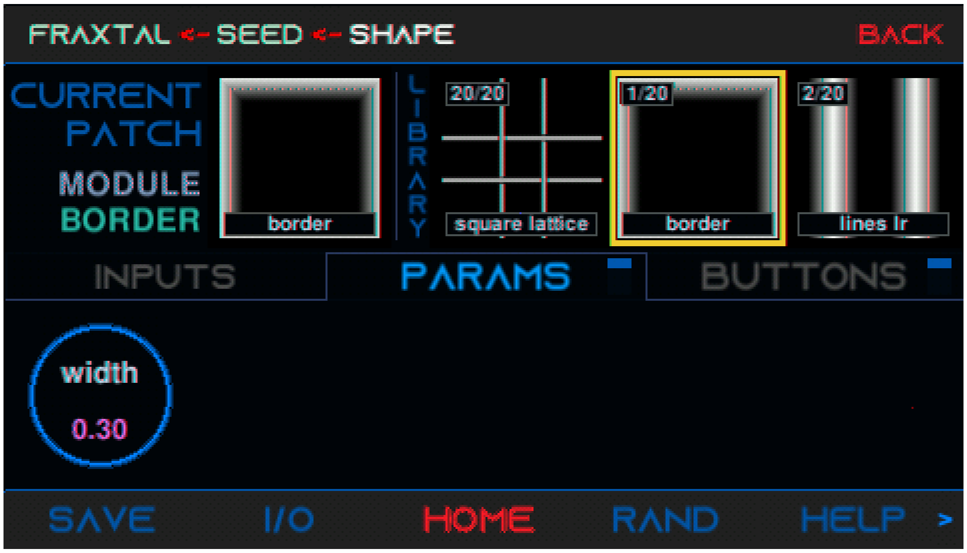
The **CREATE UI** is where new **PATCHES** are created, and the **LIVE PATCH** is modified. The **CREATE UI** allows you to edit the **LIVE** state of the synth, and then to save **PATCHES** that store what is currently **LIVE.** The **CREATE UI** does not operate on **PATCHES** that are not **LIVE** or edit existing **PATCHES** in any way**.**



LOCATION

At any point in time, the **CREATE UI** is operating on a certain **LOCATION** in the **MODULE** network which is denoted by a chain of names in the **LOCATION BAR** at the top of the **LCD. MODULES** have named **INPUTS**, and a **LOCATION** is a sequence of **INPUT** names, starting with the name of the **INSTRUMENT.** At this **LOCATION** in the **LIVE** network is an individual **MODULE,** whose state comprises a **PATCH**. The **CREATE UI** is all about manipulating this **LIVE PATCH,** and storing it for recall later.

In the image below, the **CREATE UI** is currently interacting with the **MODULE** that is connected to the SHAPE **INPUT** of the **MODULE** that is connected to the SEED **INPUT** of the **FRAXTAL INSTRUMENT**. In this instance, the **MODULE** connected in this **LOCATION** is BORDER, and it has exactly one **PARAMETER**, and 20 available **PATCHES,** created from various **MODULES** that can be loaded into the SHAPE **INPUT** of the SEED **MODULE**.



LAYOUT

LOCATION BAR: The horizontal bar at the top shows the **LOCATION** in the **MODULE** network the **CREATE UI** is currently showing. Tapping on an individual item in this bar will navigate the **CREATE UI** to that **LOCATION**.

CURRENT PATCH DISPLAY: In the top-left of the **UI** is a small widget which displays information about the **PATCH** and **MODULE** that are currently **LIVE** at the **CREATE UI**'s current **LOCATION**. If this **LIVE PATCH** has been saved directly to the synth, a small preview of the **PATCH** displays along with the **PATCH** name. If not, the preview window will say "Creating New Patch." Tap anywhere on this widget to open up the **LIBRARY** page.

LIBRARY SELECTOR: To the right of the current **PATCH** display is a **LIBRARY** selector, which allows you to switch the **LIVE PATCH** at the **CREATE UI**'s current **LOCATION.** The names at the bottom of the **PATCH** previews are their names. Push the **SHIFT** button to see which **MODULE** the **PATCH** has been saved for. Tap on a **PATCH** preview to load it, or interact with it via the **GOLD CURSOR** and **GOLD KNOB**.

INPUTS/MACROS/BUTTONS: This next section is organized by tabs. Tapping on either of 'INPUTS', 'MACROS', and 'BUTTONS' changes the active tab. If the current **MODULE** has no **INPUTS,** or relevant **PARAMETERS,** the respective tab will be grayed out.

INPUTS: This section allows you to control which **MODULES** and **PATCHES** are connected to the current **LIVE MODULE'S INPUTS**, and allows you to navigate the **CREATE UI** further into the **MODULE** network.

While this section is active, you will see a horizontal collection of preview images of the individual **INPUT MODULES**, optionally with some pagination controls. The text at the top of the preview image is the name of the **MODULE INPUT**. At the bottom of the preview, you will see the name of the **MODULE** that is currently attached to the individual **INPUT**.

When the **GOLD CURSOR** has selected a **MODULE INPUT**, scrolling the **GOLD KNOB** will cycle through **PATCHES** that can be loaded into this **INPUT**, and the name at the bottom of the preview will then be the **PATCH NAME**. Pressing the **GOLD KNOB** will load the selected **PATCH.**

MACROS: This section shows you what is currently active on the 5 **BLUE KNOBS**. It is arranged horizontally with up to 5 elements, and optional paging controls. There can be up to 3 pages of **MACRO** controls for a total of 15 controls, and the current page is denoted by the position of the horizontal blue bar to the right of the 'MACROS' text.

If the corresponding **BLUE KNOB** is controlling a single **PARAMETER**, its name and current value will be displayed. Tap on this element to navigate to the **PARAMETER INFO** page for the given **PARAMETER.** If the **PARAMETER** is currently **MODULATED**, its name will appear in a red color. If the **PARAMETER** is currently controlled via an **LFO**, its name will appear blue.

If the corresponding **BLUE KNOB** is assigned to some other function, it will display a bit of descriptive text, and tapping on it will navigate to the **MAPPING LIST** page, showing the **MAPPINGS** that are currently assigned to that particular **MACRO** control.

BUTTONS: This section shows you what is currently active on the 5 **LARGE BUTTONS**. It is arranged horizontally with up to 5 elements, and optional paging controls. There can be up to 3 pages of **BUTTON** controls for a total of 15 controls, and the current page is denoted by the position of the horizontal blue bar to the right of the 'BUTTONS' text.

If the corresponding **LARGE BUTTON** is controlling a single **PARAMETER,** likely via a **LFO**, its name will be displayed along with additional descriptive information. Tap on this element to navigate to the **PARAMETER INFO** page for the given **PARAMETER.**

If the corresponding **LARGE BUTTON** is assigned to some other function, it will display a bit of descriptive text, and tapping on it will navigate to the **MAPPING LIST** page, showing the **MAPPINGS** that are currently assigned to that particular **BUTTON** control.

<u>**MACROS & BUTTONS NOTE:**</u> When the **CREATE UI's** at the top-level **MODULE** for the current **INSTRUMENT**, then what is bound to the 5 **BLUE KNOBS**, and 5 **LARGE BUTTONS**, are the **MAPPINGS** from the top-level/**INSTRUMENT PATCH'S MAPPING GROUP.**

When the **CREATE UI** is at a different **LOCATION** in the **MODULE** network, then what is bound to the 5 **BLUE KNOBS**, and 5 **LARGE BUTTONS** are the **PARAMETER** controls for the given **LIVE MODULE**, and the title of the **'**MACROS' tab will change to 'PARAMS.' Continuous **PARAMETERS** will be bound to the **BLUE KNOBS**, while discrete **PARAMETERS** will be bound to the **LARGE BUTTONS**, and neither can be reassigned or re-configured. Otherwise these sections will behave identically.

<u>**COMMAND BAR**</u>: Identical to elsewhere in the synth, it shows available **COMMANDS**. Press **SHIFT** to see additional **COMMANDS.**

CONTROLS

<u>**GOLD CURSOR**</u>

The **GOLD CURSOR** highlights the active element **-** the element which is currently controlled by the **GOLD KNOB**, and allows the **GOLD KNOB** to cycle through available **PATCHES**. The cursor can be on the **LIBRARY** widget at the top of the page, in which case pushing the **GOLD KNOB** down will load the currently highlighted **PATCH** into the current **MODULE LOCATION**.

Or the **CURSOR** can be on one of the **MODULE'S INPUTS**, in which case the **GOLD KNOB** cycles through available **PATCHES** for that particular **INPUT**. Pushing down the **GOLD KNOB** loads the highlighted **PATCH** into that **MODULE INPUT**.

Tapping on a **MODULE INPUT** moves the **GOLD CURSOR** to that particular **INPUT**. Tapping again on it returns the **CURSOR** to the **LIBRARY** selector at the top.

<u>**KNOB CONTEXT MENU**</u>

When a **BLUE KNOB** is controlling an individual **PARAMETER**, pressing down the **KNOB** will replace the **COMMAND BAR** with a context menu. Scrolling the **BLUE KNOB** will highlight a **COMMAND**, and releasing it will execute it. There are 4 commands:

<u>**RESET**</u> - Resets the **PARAMETER** to its default value.
<u>**INFO**</u> - Navigate to the relevant **PARAMETER INFO** page.
<u>**MACRO**</u> - Open up the **MAPPING EDITOR** with a template to create a **MACRO MAPPING** for the given **PARAMETER**.
<u>**LFO**</u> **-** Open up **LFO EDITOR** with information about the currently assigned **LFO**, or a blank template to create a new one.

<u>**SHIFT & NAV BUTTON**</u>

While in the **CREATE UI**, the **SHIFT BUTTON** has additional functionality. While held down, the names in the upper right **LIBRARY** selector will temporarily change from the name of the **PATCH** to the name of the **MODULE** that the **PATCH** is saved from.

While the **SHIFT BUTTON** is held, the 5 **BLUE KNOBS,** and **LARGE BUTTONS**, will temporarily be bound to the second page of their respective **MAPPINGS**, if available. Holding the **SHIFT BUTTON** will also make the button press from the **GOLD KNOB** load the default **PATCH** for whichever **MODULE** is currently selected by the **GOLD CURSOR**.

While the **NAV BUTTON** is held the 5 **BLUE KNOBS,** and **LARGE BUTTONS**, will temporarily be bound to the respective **MAPPINGS** for the **DEFAULT PATCH** for the currently live **INSTRUMENT.** Holding the **NAV BUTTON** will also make the button press from the **GOLD KNOB** reload the currently **LIVE PATCH** for whichever **MODULE** is currently selected by the **GOLD CURSOR**.

COMMANDS

SAVE: Allows you to save a **PATCH** for recall later. A new **PATCH** will be created, creating a complete snapshot of what is currently **LIVE**. It will not affect either the current **LIVE PATCH,** or what is selected with the **GOLD CURSOR**, a fundamentally new **PATCH** will be created.
I/O: Opens up the **MAPPING LIST** page for the currently **LIVE PATCH**.
HOME: Navigate to the **HOME** page.
EXPAND: Navigate into the **INPUT MODULE** that is selected by the **GOLD CURSOR,** loading it into the **CREATE UI,** and updating the **LOCATION**.
RAND: Randomly generate a new **PATCH** for the current **LOCATION**
HELP: Show help.
INFO: Navigate to the **INFO** page for the **PATCH** that is currently selected by the **GOLD CURSOR.**
MOD LIST: Navigate to **MODULATION LIST** for the **PATCH** currently loaded in the **CREATE UI.**
LFO LIST: Navigate to the **LFO LIST** for the **PATCH** currently loaded in the **CREATE U**I.
PREVIEW: Show a **PREVIEW** of the **PATCH** currently loaded in the **CREATE UI** directly on the device's output screen. This can be exceedingly useful for designing and understanding **PATCHES** for individual **MODULES.** This **COMMAND** will change to **STOP** if a **PREVIEW** is currently being displayed.

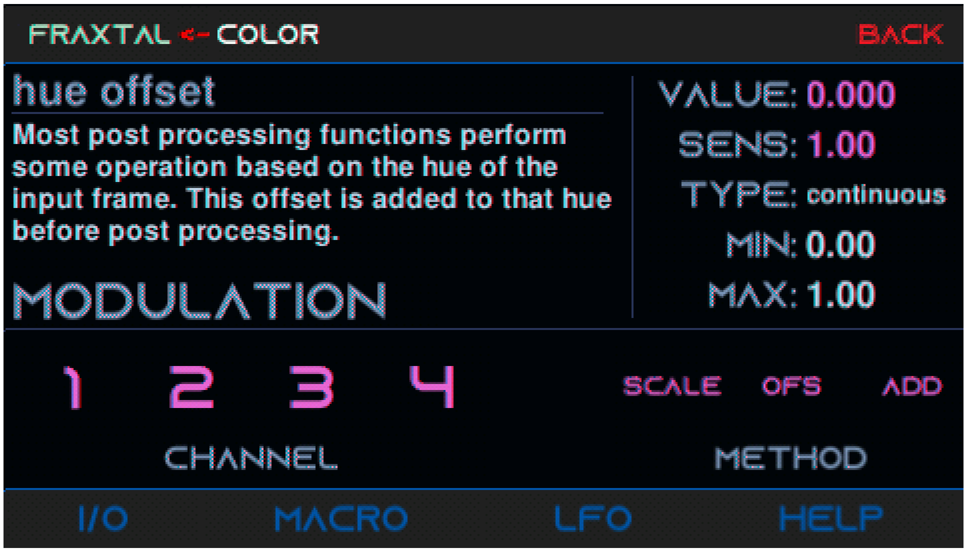
SETTINGS

Transition Spd: The amount of time it takes to transition from the current **PATCH** to the next **PATCH**, exclusively in the **CREATE UI.**

PARAMETER INFO

OVERVIEW

The **PARAMETER INFO** page provides detailed information about individual **PARAMETERS**, and also controls **MODULATION** assignment. The page can be accessed via the **CREATE UI**, either via the **INFO COMMAND** in the **KNOB** context menu, or via tapping on the **PARAMETER** name in either the **MACROS/PARAMS** or **BUTTONS** tabs.



hue offset

Most post processing functions perform some operation based on the hue of the input frame. This offset is added to that hue before post processing.

VALUE: 0.000
SENS: 1.00
TYPE: continuous
MIN: 0.00
MAX: 1.00

At the top of the **PARAMETER INFO** page, the **LOCATION** of the **MODULE** containing the **PARAMETER** is displayed, identically to in the **CREATE UI.** Below this are the three main sections of the page, and at the bottom is a **COMMAND BAR** as usual.

The top left main section of the page displays the **PARAMETER'S** name as well as a bit of descriptive text. The top right section has detailed information about the **PARAMETER:**

<u>**VALUE:**</u> The **PARAMETER'S** current value. Tap on it to set the value manually. This value will be **blue** if the **PARAMETER** is controlled by an **LFO**, and red if the **PARAMETER** is **MODULATED**.
<u>**SENS:**</u> The sensitivity of all **"INCREMENT VALUE**" **ACTIONS.** Practically, this controls how sensitive **MACRO** controls are. Increasing this value to 2, for instance will mean that a **KNOB** will need to be turned half as much for the same effect while controlling this **PARAMETER**.
<u>**TYPE:**</u> The type of the **PARAMETER**, either '**ON/OFF**', '**continuous**,' or **'discrete'** or **'2D'**
<u>**MIN:**</u> The absolute minimum possible value for the **PARAMETER.** This can be restricted further via settings in the relevant **MAPPINGS.**
<u>**MAX:**</u> The absolute maximum possible value for the **PARAMETER.** This can be restricted further via settings in the relevant **MAPPINGS.**

The bottom main section controls **MODULATION** settings. Tap on one of the 4 **MODULATION CHANNELS** to modulate the **PARAMETER** with that specific **CHANNEL**, at which point, a border will appear around both the selected channel, and the **MODULATION METHOD**. Tap another **CHANNEL** or **METHOD** to change the setting, tap the existing **CHANNEL** to disable **MODULATION**. See the **MODULATION** section of the manual for further information.

CONTROLS

The controls that are live on the 5 **BLUE KNOBS**, and 5 **LARGE BUTTONS** vary depending on the type of the **PARAMETER** that is loaded into the **PARAMETER INFO** page.

<u>**Continuous**</u>: The first **BLUE KNOB** will control the value of the **PARAMETER**, the last **BLUE KNOB** will control the sensitivity. The first 4 **LARGE BUTTONS** can be used to toggle **MODULATION CHANNELS.**
<u>**2D**</u>:

- **BLUE KNOB 1:** Controls the radius of the **PARAMETER.**
- **BLUE KNOB 2:** Controls the angle(theta) of the **PARAMETER.**

- **BLUE KNOB 3:** Controls the x-coordinate of the **PARAMETER.**
- **BLUE KNOB 4:** Controls the y-coordinate of the **PARAMETER.**
- **BLUE KNOB 5:** Controls the sensitivity of the **PARAMETER.**

ON/OFF: The first **LARGE BUTTON** will toggle the **PARAMETER** on or off. The **5 BLUE KNOBS** have no function.

Discrete: The first **LARGE BUTTON** will decrement the **PARAMETER** by 1, the second **LARGE BUTTON** will increment the **PARAMETER** by 1. The **5 BLUE KNOBS** have no function.

COMMANDS

I/O: Opens up the **MAPPING LIST** page for the current **PARAMETER**.

MACRO - Open up the **MAPPING EDITOR** page with a template to create a **MACRO MAPPING** for the current **PARAMETER**.

LFO - Open up **LFO EDITOR** with information about the currently assigned **LFO**, or a blank template to create a new one. This **COMMAND** will be highlighted if an **LFO** currently exists.

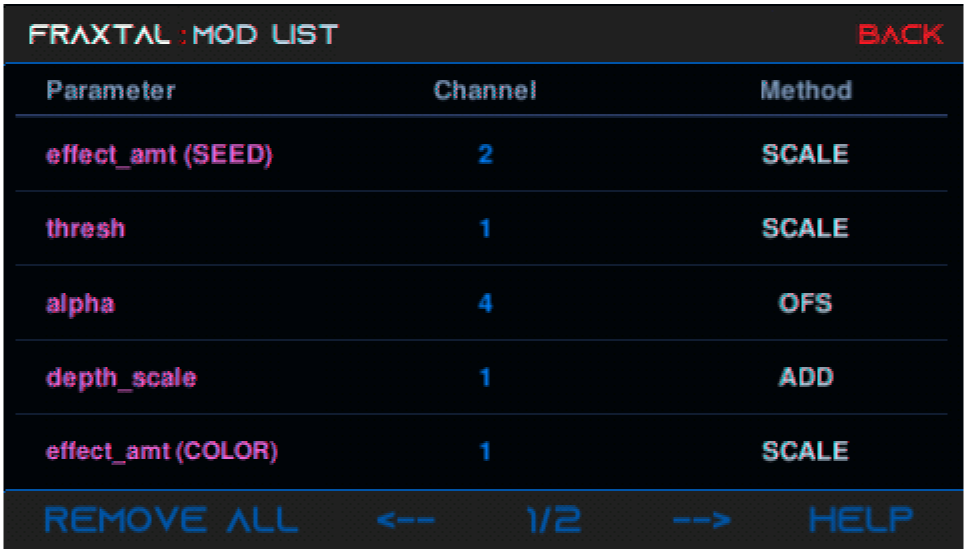
HELP: Show help.

MODULATION LIST

OVERVIEW

The **MODULATION LIST** page lets you see all **MODULATION** that is currently active for the **LIVE PATCH.** The top bar will display the **LOCATION** of this **PATCH**. The page is organized into a list of elements, each of them identifying a **PARAMETER** that is being **MODULATED**, the **MODULATION** channel, and the **MODULATION METHOD.** As usual, a **COMMAND BAR** is at the bottom of the screen.

This page can be accessed via a **COMMAND** in the **CREATE UI.**



Parameter	Channel	Method
effect_amt (SEED)	2	SCALE
thresh	1	SCALE
alpha	4	OFS
depth_scale	1	ADD
effect_amt (COLOR)	1	SCALE

The **MODULATION LIST'S** page can be changed via the pagination controls at the bottom of the page, or via swiping gestures. You can select an element by tapping on it, at which point it becomes highlighted, and the available **COMMANDS** in the **COMMAND BAR** change. Tap again to deselect.

COMMANDS

While no element is selected:

REMOVE ALL: Removes all active **MODULATION** for the current **LIVE PATCH.**
HELP: Show help.

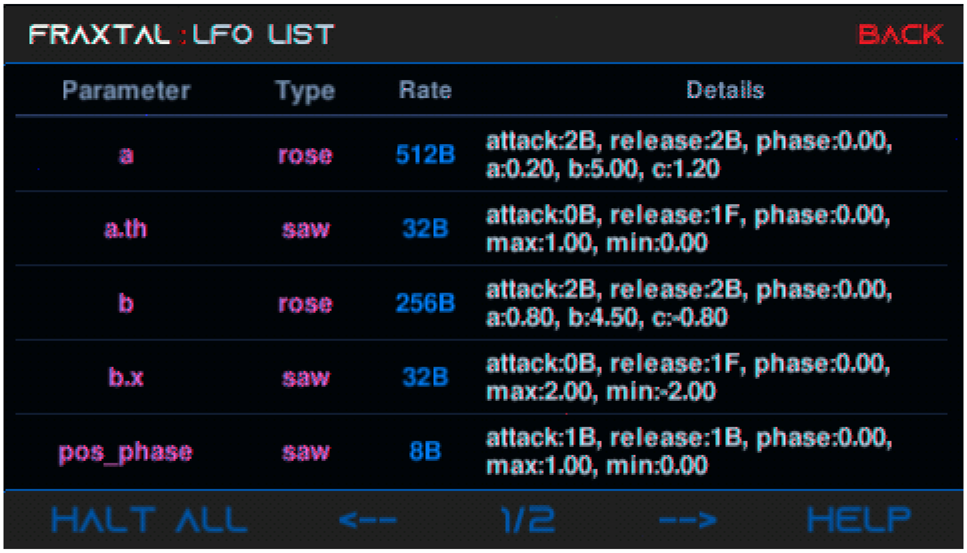
While an element is selected:
VIEW PARAM: Navigate to **PARAMETER INFO** page for the selected **PARAMETER.**
VIEW MODULATION: Navigate to the selected **MODULATION** channel in the **CREATE UI**

LFO LIST

OVERVIEW

The **LFO LIST** page lets you see all **LFOs** that are currently active for the **LIVE PATCH.** The top bar will display the **LOCATION** of this **PATCH**. The page is organized into a list of elements, each of them identifying a **PARAMETER** that is currently being animated via an **LFO**, the **LFO** type and **RATE**, as well as additional details. As usual, a **COMMAND BAR** is at the bottom of the screen.

This page can be accessed via a **COMMAND** in the **CREATE UI.**



Parameter	Type	Rate	Details
a	rose	512B	attack:2B, release:2B, phase:0.00, a:0.20, b:5.00, c:1.20
a.th	saw	32B	attack:0B, release:1F, phase:0.00, max:1.00, min:0.00
b	rose	256B	attack:2B, release:2B, phase:0.00, a:0.80, b:4.50, c:-0.80
b.x	saw	32B	attack:0B, release:1F, phase:0.00, max:2.00, min:-2.00
pos_phase	saw	8B	attack:1B, release:1B, phase:0.00, max:1.00, min:0.00

The **LFO LIST'S** page can be changed via the pagination controls at the bottom of the page, or via swiping gestures. You can select an element by tapping on it, at which point it becomes highlighted, and the available **COMMANDS** in the **COMMAND BAR** change. Tap again to deselect.

COMMANDS

While no element is selected:
HALT ALL: Removes all active **LFOs** for the current **LIVE PATCH.**
HELP: Show help.

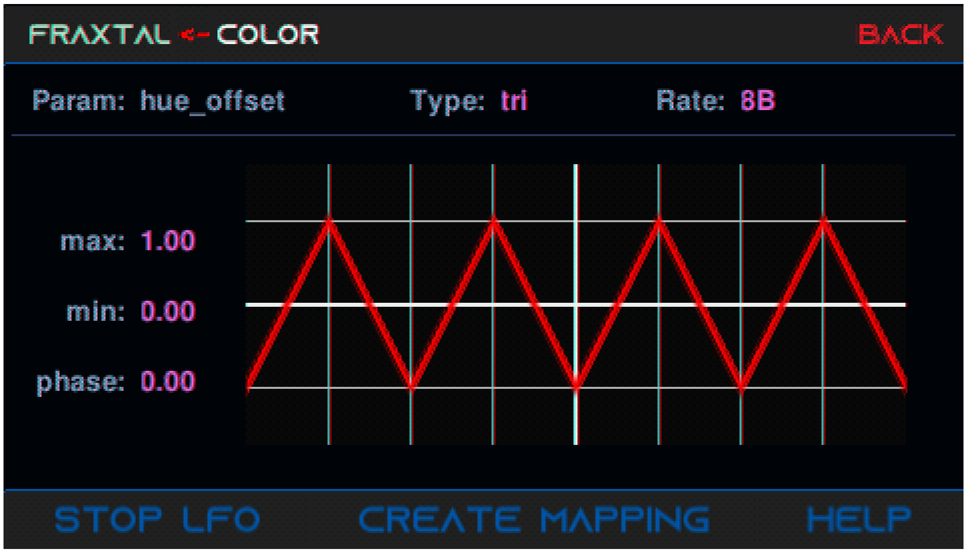
While an element is selected:
VIEW: Navigate to **LFO EDITOR** for the selected **LFO.**
HALT: Halts the selected **LFO.**

LFO EDITOR

OVERVIEW

The **LFO EDITOR** allows you to edit existing **LFOS** and to create new ones for the specified **PARAMETER**. It can be accessed via the **PARAMETER INFO** page, the **LFO LIST** page, or the popup menu on the **CREATE UI.**

The top bar will display the **LOCATION** of the **PARAMETER** and a **COMMAND BAR** is located at the bottom of the screen. **LFOs** are created and edited via the controls in the middle section.



Param: hue_offset Type: tri Rate: 8B

max: 1.00
min: 0.00
phase: 0.00

If no **LFO** is currently active for the specified **PARAMETER**, this page will let you create one. Simply modify the relevant settings and tap the **START LFO COMMAND**. If there is an existing **LFO**, the page will display its current settings. Modifying any of these settings will immediately update the existing **LFO.**

The first row of the main section has the name of the parameter in question, along with information about the currently selected **LFO** type, and its **RATE**. Tap on either to change them.

Along the left side of the middle section are additional, **LFO** type specific settings, that can also be changed by tapping on them. And to the right is a simple render of the **LFO**.

COMMANDS

STOP/START LFO: If an **LFO** is currently live for the specified **PARAMETER**, this stops it, and resets the **PARAMETER** to its default value. Otherwise, an **LFO** is started, given the current settings.
CREATE MAPPING: Opens up the **MAPPING EDITOR** with the current **LFO** settings as a template.
HELP: Show help.

MAPPING LIST

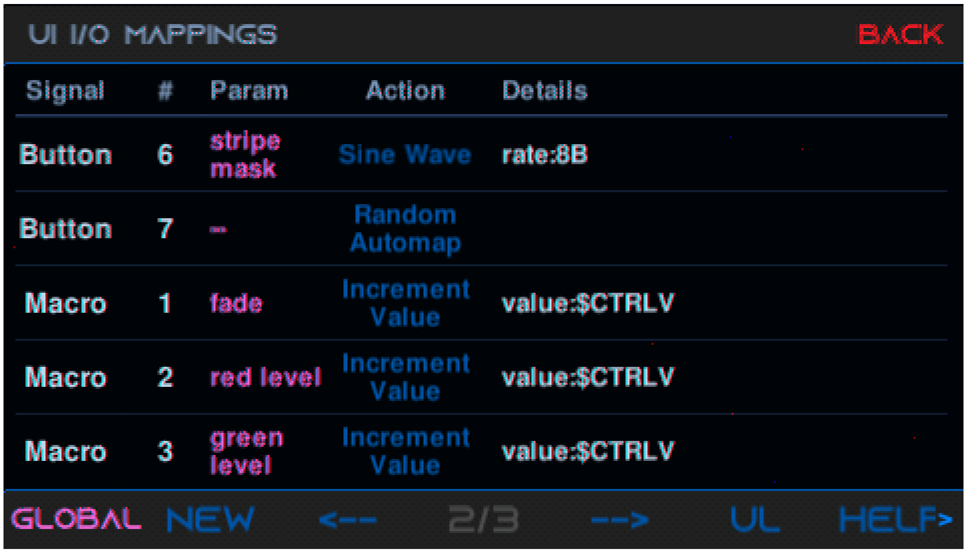
OVERVIEW

The **MAPPING LIST** page shows the list of **I/O MAPPINGS** for a specific **MAPPING GROUP**. This **MAPPING GROUP** can either be the **GLOBAL MAPPING GROUP**, or the **MAPPING GROUP** for the **LIVE INSTRUMENT PATCH.**

If the current **LIVE PATCH** has no **MAPPINGS** explicitly saved, the **MAPPINGS** for the default **PATCH** for the current **LIVE INSTRUMENT** will be used, as described in **MAPPINGS & MAPPINGS GROUP** section of the manual.

The **MAPPING LIST** can either show all **MAPPINGS** for the **MAPPING GROUP** or some subset of them, depending on the **SIGNAL** type.

The **MAPPING LIST** can be viewed from the **CREATE** & **PERFORM UIS** by tapping on the **I/O COMMAND**, or via the various **I/O CONFIG** pages.



Signal	#	Param	Action	Details
Button	6	stripe mask	Sine Wave	rate:8B
Button	7	--	Random Automap	
Macro	1	fade	Increment Value	value:$CTRLV
Macro	2	red level	Increment Value	value:$CTRLV
Macro	3	green level	Increment Value	value:$CTRLV

The **MAPPING LIST** is organized as a list of **MAPPINGS** and a **COMMAND BAR** at the bottom. Push the **SHIFT BUTTON** to view additional **COMMANDS.** The additional **COMMANDS** include controls for selecting which **MAPPINGS** are shown.

The horizontal bar for each **MAPPING** has 5 sections
Signal: This is the **SIGNAL** for the corresponding **MAPPING.**
#(channel): This is the **CHANNEL** for the corresponding **SIGNAL**. Not all **SIGNALS** have a **CHANNEL.**
Param: This is the **PARAMETER** that the **ACTION** acts on. Not all **ACTIONS** have a **PARAMETER**.
Action: This is the **ACTION** for the corresponding **MAPPING,** which will be triggered when the synth receives the corresponding **SIGNAL.**
Details: These are the details of the **ACTION.**

CONTROLS

The list of **MAPPINGS** can be scrolled with swipe gestures or via the pagination **COMMANDS** in the **COMMAND BAR**. Tapping on a **MAPPING** selects and highlights it, changing the available **COMMANDS**. Tap again to deselect.

COMMANDS

GLOBAL / PATCH NAME: The first command will either show the term 'GLOBAL' or it will show the name of the **LIVE INSTRUMENT PATCH.** Tap on it to cycle. Whichever is displayed will determine which **MAPPING GROUP** is shown in the **MAPPING LIST.**

NEW: Navigate to the **MAPPING EDITOR** to create a new **MAPPING**

UL: Short for **UN-LEARN.** The synth will wait to receive a **SIGNAL**. You will then be able to delete all **MAPPINGS** corresponding to the received **SIGNAL.**

HELP: Show help.

ALL: Show all **MAPPINGS** for the current **MAPPING GROUP.**

MIDI: Show only **MAPPINGS** for **MIDI SIGNALS** for the current **MAPPING GROUP.**

CV: Show only **MAPPINGS** for **CV SIGNALS** for the current **MAPPING GROUP.**

AUDIO: Show only **MAPPINGS** for **AUDIO SIGNALS** for the current **MAPPING GROUP.**

UI: Show only **MAPPINGS** for **KNOB & BUTTON SIGNALS** for the current **MAPPING GROUP.**

CLEAR: This will *delete all **MAPPINGS*** that are currently in the **MAPPING LIST**, including those on other pages.

When a **MAPPING** is selected, the **COMMANDS** available change.

EDIT: Navigate to the **MAPPING EDITOR** to edit the currently selected **MAPPING**

DELETE: Delete the currently selected **MAPPING**

DISABLE: Disable the currently selected **MAPPING.** If the **MAPPING'S SIGNAL** is received by the synth, the corresponding **ACTION** will *not* be performed.

TRIGGER: Manually trigger the **ACTION** of the currently selected **MAPPING.**

UNMAP SIGNAL: This will *delete all **MAPPINGS*** that are currently in the **MAPPING LIST** that have the same **SIGNAL** and **CHANNEL** as the selected **MAPPING.**

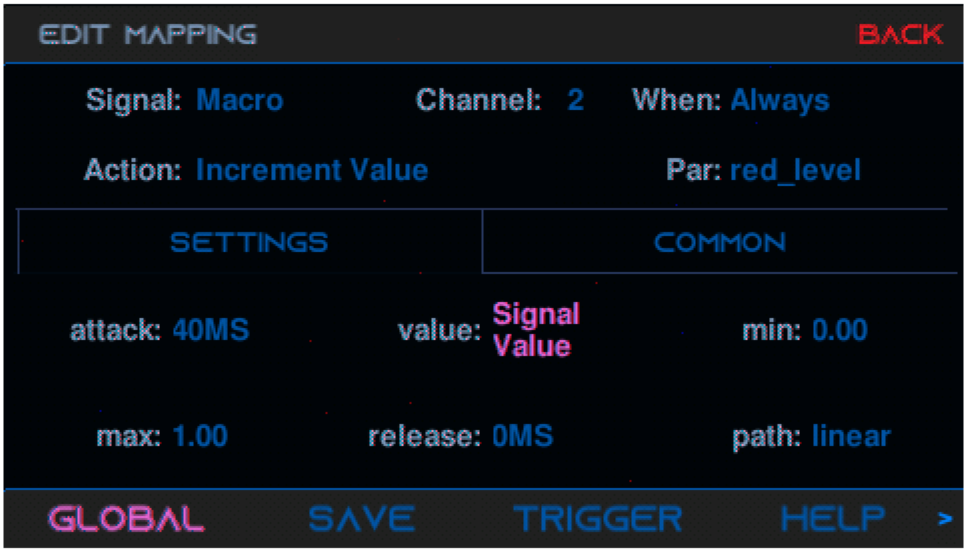
UNMAP PARAM: This will *delete all **MAPPINGS*** that are currently in the **MAPPING LIST** that have the same **PARAMETER** as the selected **MAPPING**.

MAPPING EDITOR

OVERVIEW

The **MAPPING EDITOR** is where new **MAPPINGS** are created and existing **MAPPINGS** are modified**.** In the top section the **SIGNAL** and **ACTION** for the **MAPPING** are specified**.** At the bottom is a **COMMAND BAR.** Push the **SHIFT BUTTON** to view additional **COMMANDS.**

In the middle of the page is a (possibly empty) dynamic selector. When changing the **SIGNAL** of the **MAPPING**, the selector will show available **SIGNALS.** When changing the **ACTION** of the **MAPPING**, the selector will show available **ACTIONS**. When both a **SIGNAL** and **ACTION** have been selected, the middle of the page will present configuration settings for the specific **ACTION.**



Signal: Macro Channel: 2 When: Always

Action: Increment Value Par: red_level

attack: 40MS value: Signal Value min: 0.00

max: 1.00 release: 0MS path: linear

CREATING MAPPINGS - SIGNALS

The first step in creating a **MAPPING** is selecting the **SIGNAL** which is done by tapping on the text to the right of **SIGNAL**. The center section of the page will then change to a selector organized by tabs, which sort the supported **SIGNALS** by type. See the **SIGNAL REFERENCE** for details on specific **SIGNALS**.

After a **SIGNAL** has been selected, most **SIGNALS** further require a **CHANNEL/NOTE/CC** to be specified**.** This, for instance**,** is used to identify which **CV** jack a **SIGNAL** is coming from, or which particular **MIDI NOTE** is pressed, or which **KNOB** is turned, etc. Tap on this value to set it.

The **MAPPING'S ACTION** can be conditioned to trigger only under certain circumstances placed upon the incoming **SIGNAL'S VALUE**, by tapping on the text to the right of **WHEN**. The center section of the page will then change to a selector organized by tabs, which sort the available options by type. The available options are as follows:

Always: The **ACTION** will trigger regardless of the **SIGNAL'S VALUE.**
Changed: The **ACTION** will trigger only if the **SIGNAL'S VALUE** has changed.
Value >, Value <, Value >=, Value <=: The **ACTION** will trigger only when the **SIGNAL'S VALUE** is > (<, >=, <= respectively) the provided **THRESHOLD** value.
Rises >: The **ACTION** will trigger only as the **SIGNAL'S VALUE** rises past the provided **THRESHOLD** value.
Falls <: The **ACTION** will trigger only as the **SIGNAL'S VALUE** falls below the provided **THRESHOLD** value.

CREATING MAPPINGS - ACTIONS

Once a **SIGNAL** has been specified for the **MAPPING**, an **ACTION** can then be selected by tapping the text to the right of 'Action'. The center section of the page will then change to a selector organized by tabs, which sort the supported **ACTIONS** by type. See the **ACTION REFERENCE** for details on specific **ACTIONS**.

Some **ACTIONS** operate on a specific **MODULE LOCATION.** When a **MODULE LOCATION** selection is needed, a bit of text to the right of the **ACTION** selection appears, titled 'Module.' Tap the bit of text to the right of that to open the **MODULE LOCATION** selector. This will show all **MODULE LOCATIONS** available for the current **INSTRUMENT**. Select one which the selected **ACTION** will act upon.

Many **ACTIONS** operate on a specific **PARAMETER.** When a **PARAMETER** selection is needed, a bit of text to the right of the **ACTION** selection appears, titled 'Par.' Tap the bit of text to the right of that to open the **PARAMETER** selector. This will show all **PARAMETERS** available for the current **INSTRUMENT**. Select one which the selected **ACTION** will act upon.

NOTE: Some **PARAMETERS** exist in more than one **MODULE**. These will be differentiated by the **LOCATION** of the containing **MODULE** in parentheses by the **PARAMETER** name. When you select one of these **PARAMETERS** in the selector, you will have the option for your **MAPPING** to target all instances of the selected **PARAMETER** in all **MODULES** or just the individual one.

2D PARAMETERS & SLICES

Some **PARAMETERS** are 2-dimensional. When this is the case, you will be further asked to select a **SLICE** of the **PARAMETER** for the **ACTION** to act on. You can either choose the entire 2D value, in which case, **ACTIONS** will be limited to those compatible with 2D values. In particular, the only **LFOs** available will be 2D **LFOs**.

If you choose a 1D **SLICE** you will have access to the normal collection of **ACTIONS**, but the action will only control a '**SLICE**' of the 2D **PARAMETER** value. You can have multiple **MAPPINGS** for the various **SLICES** of 2D **PARAMETERS.**

X - the X coordinate of the 2D **PARAMETER.**
Y - the Y coordinate of the 2D **PARAMETER.**
R - the radius of the 2D **PARAMETER.**
TH - the angle of the 2D **PARAMETER.**
2D - the entire 2D **PARAMETER**.

CREATING MAPPINGS - ACTION DETAILS

After an **ACTION** has been selected, the center section of the page will then change to a selector organized by tabs for setting the specific details of an **ACTION**. The left tab is for settings particular to the specified **ACTION**, the right section is for settings that all **ACTIONS** have. Please see the **ACTION REFERENCE** for details on specific settings for specific **ACTIONS**.

All **ACTIONS** shard the following settings:
prob: This is the probability(from 0 to 1) that the **ACTION** will be triggered when the synth receives the corresponding **SIGNAL**. By default it is 1.0, so the **ACTION** will always be triggered.
delay: This is the delay from when the synth receives the **MAPPING'S SIGNAL** to when it triggers the **MAPPING'S ACTION.** By default it is 0MS, so the **ACTION** will be triggered immediately. You can use this delay to create sequencer-like effects.
label: When the **UI** is displaying an **ACTION** in the **CREATE** or **PERFORM UI's,** this bit of text will be shown. If it is set to '**AUTO'** a name will be automatically generated. Additionally some **MAPPING ACTIONS** refer to other **MAPPINGS**. The provided label is used as the identifier for this purpose.

CREATING MAPPINGS - TIME & SPECIAL VALUES

Most **ACTIONS** have settings that require a **TIME** value. When editing one of these values, additional options will show up on the right side of the **KEYPAD** with for selecting from the 3 supported **TIME UNITS.** Simply tap one to change the **TIME UNIT.**

When editing an **ACTION'S** other numerical settings, in addition to simply specifying the numerical value explicitly, certain **SPECIAL VALUES** are available as alternatives on the right side of the **KEYPAD:**

Signal Value: Each time the **ACTION** is triggered, this setting will be provided the **VALUE** of the **SIGNAL** which triggered the **ACTION.**
Random (0,1): Each time the **ACTION** is triggered, this setting will be provided a random value from 0 to 1
Random Integer: Each time the **ACTION** is triggered, this setting will be provided a random whole number from 0 to 1000

Additionally, if the specified **ACTION** has an associated **PARAMETER,** the following **SPECIAL VALUES** will be available:
Current Value: The current numerical value of the specified **PARAMETER.**
!Current Value: 1.0 - the current numerical value of the specified **PARAMETER.**
Default Value: The default value of the specified **PARAMETER.**

COMMANDS

GLOBAL / PATCH NAME: The first command will either show the term 'GLOBAL' or it will show the name of the **LIVE INSTRUMENT PATCH.** Tap on it to cycle. Whichever is displayed will determine which **MAPPING GROUP** the **MAPPING** will be saved to.
SAVE: Saves the **MAPPING.** If this page was arrived at via editing an existing **MAPPING**, saving will overwrite this **MAPPING**. Otherwise it will create a new **MAPPING.**
TRIGGER: Manually trigger the **ACTION** specified on the page.
HELP: Show help.
SAVE NEW: Creates a new **MAPPING**. Does not overwrite the previous **MAPPING**, if it exists.
DELETE: Delete the current **MAPPING**, if it exists.
DISABLE: Disable the current **MAPPING**, if it exists. If the **MAPPING'S SIGNAL** is received by the synth, the corresponding **ACTION** will *not* be performed.

Available when selecting a **SIGNAL:**

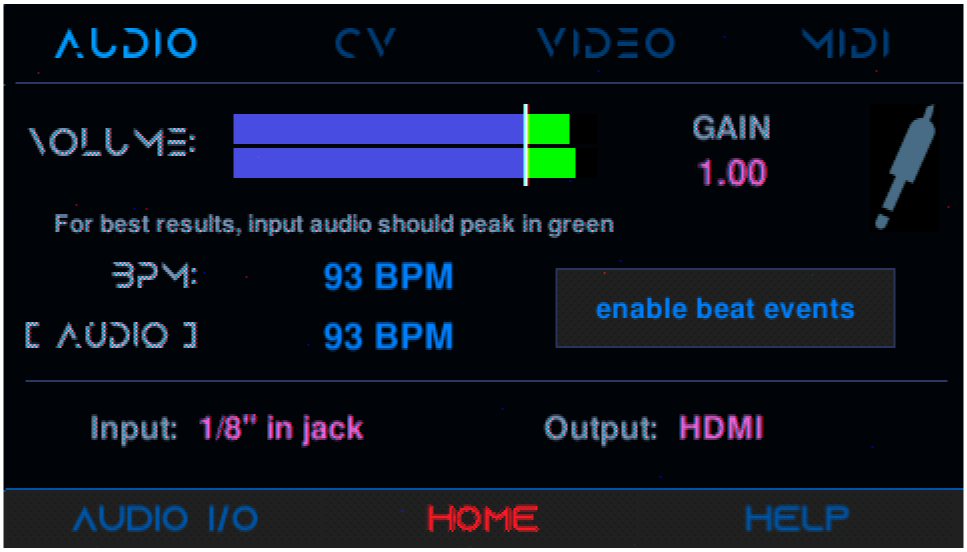
LEARN: When this is pressed, the synth will wait to receive an external **SIGNAL.** The first **SIGNAL** received by the synth will be set as this **MAPPING'S SIGNAL**, along with the provided **CHANNEL**, if it exists.

I/O CONFIGURATION

The **I/O** page is for configuring interaction with external hardware. It has 4 relevant subsections for the 4 main **I/O** types, **AUDIO, CV, VIDEO,** and **MIDI.**

AUDIO CONFIG

The **AUDIO I/O** page is used for configuration of **AUDIO** input & output, and for managing some aspects of how the synth responds to **AUDIO SIGNALS.**



GAIN 1.00

For best results, input audio should peak in green

BPM: 93 BPM

[AUDIO] 93 BPM

Input: 1/8" in jack

Output: HDMI

The volume meter in the top left shows the level of the incoming **AUDIO** signal. Currently, line level ⅛” **AUDIO** through the back panel, and **USB AUDIO** input via external **USB** sound cards are supported. If there is no activity in this meter, the synth isn’t receiving **AUDIO**. Tapping on the value to the right of “Input” will cycle the available **AUDIO** inputs. The synth can also pass through any incoming **AUDIO**. Tap the value to the right of “Output” to cycle through the available outputs.

For best results, the incoming **AUDIO** should peak in the green region at its loudest. If whatever is sending **AUDIO** into the synth is at or close to maximum volume, you should be fine. In cases where the signal still isn’t loud enough, the gain control to the right of the volume meter can be used to amplify the signal internally. Although if this value is too large, it will introduce noise. You should never need to increase this value beyond 1.5 or so.

The icon to the right of the gain control will light up if an ⅛” audio cable is plugged into the back of the synth.

The synth always tracks a current **TEMPO,** in beats per minute. This **TEMPO** can come from several sources, one of which being the **AUDIO** input. The first **BPM** value tells you what the current **TEMPO** of the synth is. The second **BPM** value tells you what the **TEMPO** of the incoming audio signal is. These very well may be different if the synth is not using the **AUDIO TEMPO** as its current **TEMPO** source. Tap on either of these to open the **TEMPO MANAGER.**

The **AUDIO** analysis engine is constantly attempting to determine the **TEMPO** of the incoming **AUDIO** signal, if present. If the analysis engine is currently unable to determine the **TEMPO**, the **AUDIO TEMPO** display will be colored red on this page.

There are 3 **SETTINGS** which affect the system’s ability to detect a **TEMPO**. **Min** and **Max Audio Tempo** set the bounds for tempo detection. The narrower this range is, the more likely the **TEMPO** will be successfully detected. The **‘Tempo Conf Threshold’** also determines how confident the system needs to be to detect a **TEMPO**. The lower this value is, the more likely the system is to detect a **TEMPO**, but the more likely it also is that the detected **TEMPO** will be incorrect. The proper value for this parameter largely seems to be dependent on the musical genre of the incoming **AUDIO**.

AUDIO & BEAT REACTIVITY

The synthesizer can react to **AUDIO** in 3 major ways. Some **MODULES** explicitly respond to **AUDIO**. The synth also analyzes the incoming **AUDIO** stream and generates several **AUDIO SIGNALS** that can be used in **MAPPINGS.**

The synth can also detect when a beat hits in the incoming **AUDIO** stream and react in real time. This feature is designed explicitly for reacting to more high energy danceable **AUDIO** input. It can be toggled on & off by tapping the “enable beat signals” button, or via an **ACTION**. It is off by default, as it can take some degree of fineness to make it look good and not be overtly chaotic.

When a beat is detected by the synth, two different **I/O MAPPINGS** with the **BEAT SIGNAL** are randomly chosen and triggered. These **MAPPINGS** can be created and edited like any other **MAPPINGS,** giving control over how the synth will respond. In particular, it is the **CHANNEL** of the **I/O MAPPINGS** that is randomly selected. All **BEAT MAPPINGS** of a randomly selected **CHANNEL** will be triggered. When a beat is detected by the synth, the **‘**enable beat signals**’** button also will temporarily flash white.

The synth will not respond to beat events if it doesn’t have sufficient confidence. This feature was designed for use with music with repetitive, consistent and not overtly complex rhythms. It will not work well(if at all) for individual instruments, with music with an inconsistent **TEMPO,** or with music that is overly complex. It was largely optimized to work with fairly loud and straightforward electronic music.

This can be controlled with the ‘**Beat Conf Threshold’ SETTING**. The lower this value is, the more likely the system is to detect a **BEAT**, but the more likely it also is that the detected **BEAT** will be incorrect

The synth will also not respond to beat events if the volume of the **AUDIO** over the last few seconds is too quiet, in essence turning off the feature during track breakdowns and transitions. This can be controlled with the ‘**Beat Loud Threshold’ SETTING.**

This feature is considered somewhat experimental, and while it works well in many circumstances, its functionality will likely change over time.

NOTE: The **FRAXTAL INSTRUMENT** comes with default **BEAT MAPPINGS** for **BEAT REACTIVITY.** The other **INSTRUMENTS** do not, and must be created manually.

COMMANDS

<u>AUDIO I/O</u>: Navigate to the **MAPPING LIST** page for **AUDIO I/O MAPPINGS**
<u>HOME:</u> Return to the **HOME** page.
<u>SETTINGS:</u> Navigate to the **SETTINGS** page for the **AUDIO** system.
<u>HELP</u>: Show help.

SETTINGS

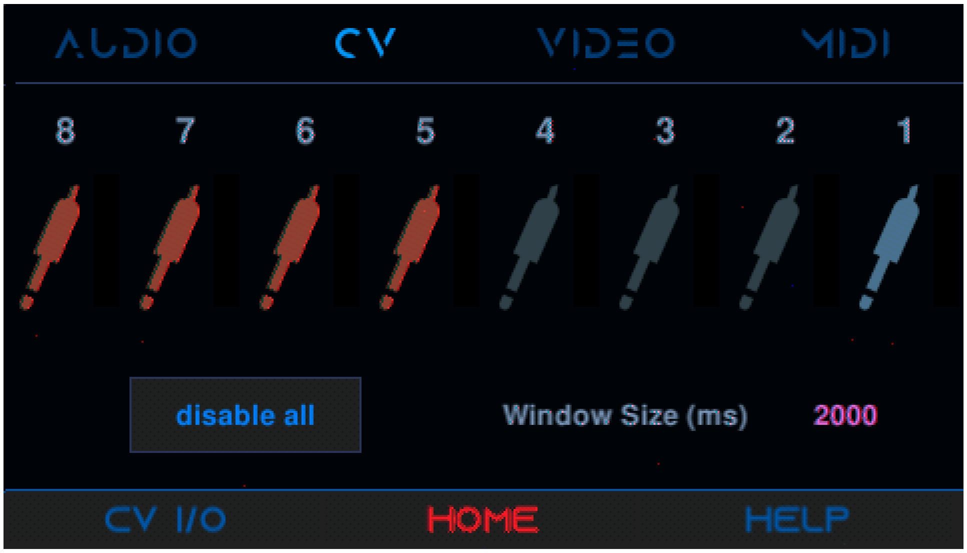
Various **SETTINGS** are available for the **AUDIO** system available from the **AUDIO** section of the **SETTINGS** page.

<u>Max Audio Tempo:</u> The maximum possible **TEMPO** returned by the **AUDIO** analysis system. A wider range means a higher likelihood of the **TEMPO** detection algorithm failing.
<u>Min Audio Tempo</u> The minimum possible **TEMPO** returned by the **AUDIO** analysis system. A wider range means a higher likelihood of the **TEMPO** detection algorithm failing.

Tempo Conf Thresh: The confidence threshold a **TEMPO** detection must reach. The lower this value is, the more likely the system is to detect a **TEMPO**, but the more likely it also is that the detected **TEMPO** will be incorrect.
Beat Signals: When this is enabled, and a **BEAT** is detected, a **BEAT SIGNAL** will be sent.
Beat Conf Thresh: The confidence threshold a **BEAT** detection must reach. The lower this value is, the more likely the system is to detect a **BEAT**, but the more likely it also is that the detected **BEAT** will be incorrect.
Beat Loud Thresh: A threshold for the **AVG VOLUME SIGNAL** the **AUDIO** must exceed for a **BEAT SIGNAL** to be sent. In essence turning off the feature during track breakdowns and transitions.

CV CONFIG

The **CV I/O** page is used for configuration of **CV** input and for managing how the synth responds to **CV SIGNALS.**



Window Size (ms) 2000

There are 8 non-polar 0-10v **CV** input jacks on the back panel of the synth, visibly labeled, all of which can be used simultaneously. When a **CV** cable is plugged into the synth the corresponding icon on the **CV I/O** page will illuminate.

CV inputs can be disabled by tapping on the corresponding icon, highlighting it in red. No **CV SIGNALS** will then be generated on that specific **CHANNEL.** It can similarly then be enabled. Tapping “disable all” will deactivate all **CV SIGNALS**, which can similarly be re-enabled.

A rolling window of the recently incoming **CV** data is kept for use in some specialized **MODULES.** The window size is configurable, up to 8 seconds, and can be changed simply by tapping on the current value and entering a new one.

COMMANDS

CV I/O: Navigate to the **MAPPING LIST** page for **CV I/O MAPPINGS**
HOME: Return to the **HOME** page.
HELP: Show help.

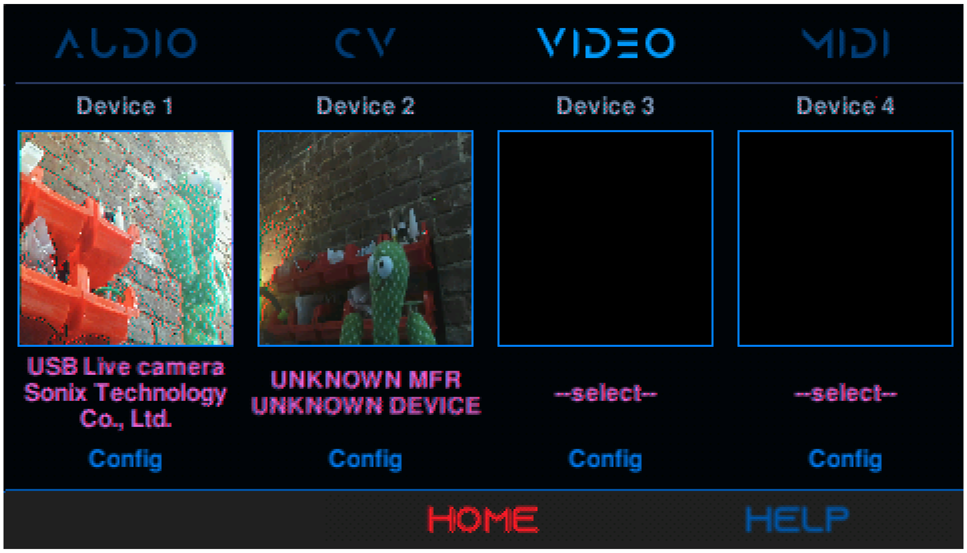
VIDEO CONFIG

The **VIDEO I/O** page is used for configuration of **VIDEO** input and devices. The synth supports up to 4 **VIDEO** streams, however, care must be taken to not overload the synth's processing capabilities. The input resolution of **VIDEO** streams can be configured, however in highly processing intensive **INSTRUMENTS** such as **FRAXTAL**, the resolutions should be kept low, particularly if multiple **VIDEO** sources are live, or else the synth can become overloaded and may slow down.

VIDEO input into the Recursion Studio is done solely via the 4 **USB** ports on the left panel of the synth. Most(but not all) **USB** video devices are supported, including web cameras. **VIDEO** input from laptops and other **HDMI** sources is handled via the **HDMI-USB** adapter(included).

The **VIDEO I/O** page is arranged as a list of 4 'device slots' which show a live preview of the **VIDEO** device that is currently assigned to this slot. If the preview is not active, your **VIDEO** device is either not active or not supported.

NOTE: ***Do not unplug USB VIDEO devices while they are LIVE. Doing so will almost certainly cause the synth to crash. This is a known issue and will be fixed in a future SYSTEM UPDATE. However, this can be avoided by disabling the VIDEO device before unplugging, as explained below.***



There are 4 available **VIDEO** device slots, which get assigned directly to various **PATCHES/MODULES** in the synth. In particular, the **VIDEO** device slot, 'Device 1' will be assigned to two **PATCHES** named 'video_tex1', and 'video_buf1'. Similarly for the other **VIDEO** device slots.

All **VIDEO** devices plugged into the synth can be freely mapped to individual device slots, although they are automatically mapped until you start manually assigning them. Devices can be mapped by tapping on the name of the device under the live preview image, or the '--select—' tag if no device is currently assigned. Then tap on the name of your **VIDEO** device.   You can also tap 'None' to unmap a device from the current slot, at which point it will be safe to unplug.   Note also that a **VIDEO** device can only be used in one device slot at a time.

If your device is not listed, it is likely that it is not supported at this time. If this is the case, feel free to contact support@entropyandsons.com with the name of your **VIDEO** device, to lodge a request for support, and we will look into it.

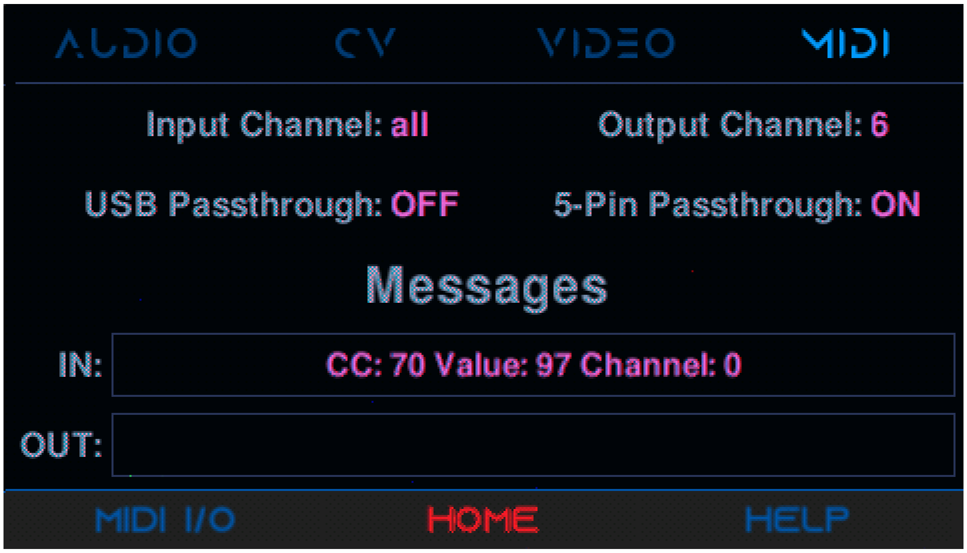
The resolution of the **VIDEO** device can also be configured, by tapping the 'Config' label on the relevant device slot. Note however that increasing the resolution requires additional processing capabilities and can slow down the synth. For heavy processing **INSTRUMENTS** such as **FRAXTAL** and **INTERFERENCE**, a resolution of 640x480 is recommended, and almost always sufficient as it is likely going to be distorted and highly processed.

COMMANDS

HOME: Return to the **HOME** page.
HELP: Show help.

MIDI CONFIG

The **MIDI I/O** page is used for configuration of **MIDI** input and for managing how the synth responds to **MIDI SIGNALS.** The synth can respond to **MIDI** input over **USB,** and it can handle both **MIDI** input and output over the 5-Pin DIN connectors.



Input Channel: all

Output Channel: 6

USB Passthrough: OFF

5-Pin Passthrough: ON

Messages

IN: CC: 70 Value: 97 Channel: 0

OUT:

There is a configurator for **MIDI I/O** settings at the top of the page, and a monitor widget for input & output messages at the bottom.

By default the synth listens on all **MIDI** channels. However this can be restricted to a single channel by tapping the 'Input Channel' setting. The output **MIDI** channel can also be specified.

Output is limited to 5-Pin, and currently is only passthrough. Passthrough can be toggled for USB and 5-Pin, passing through either **MIDI** input to 5-pin output. **MIDI** out **ACTIONS** will be added in a future **SYSTEM UPDATE**.

Incoming and outgoing **MIDI** messages are monitored in the relevant 'MESSAGES' sections.

COMMANDS

MIDI I/O: Navigate to the **MAPPING LIST** page for **MIDI I/O MAPPINGS**
HOME: Return to the **HOME** page.
HELP: Show help.

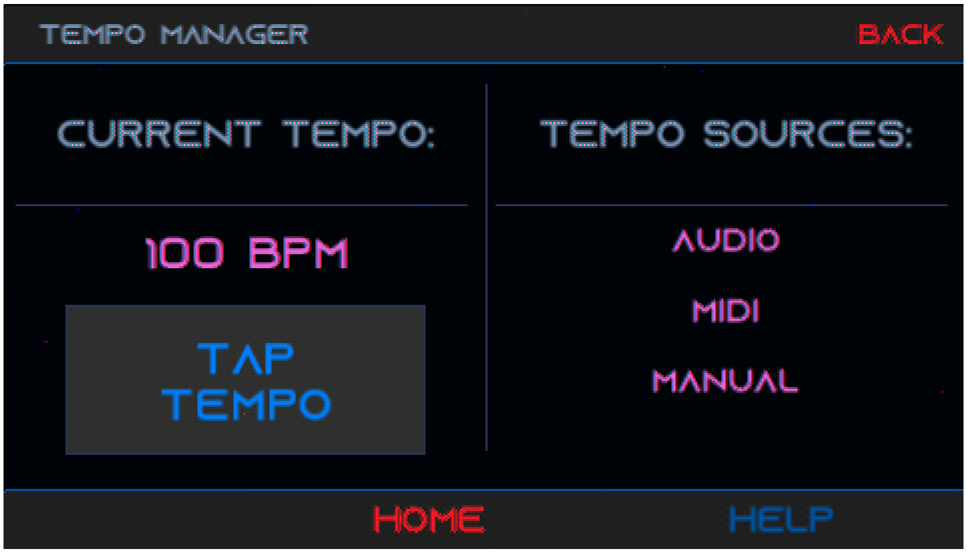
TEMPO MANAGER

OVERVIEW

The device at all times tracks an active **TEMPO**, measured in **BPM (**beats per minute). This can be set manually, learned from an **AUDIO** signal, or via **MIDI** clock, and is managed via the **TEMPO MANAGER**. The device's active **TEMPO** is used when setting **RATES, ATTACKS,** and **TRANSITION SPEEDS** in terms of **BEATS**. The **TEMPO MANAGER** can be accessed via the **SETTINGS** page by tapping on the **TEMPO** on the **AUDIO IO** config page.

The **TEMPO MANAGER** is divided into 2 sections, the left displaying and offering direct control over the current **TEMPO**, and the right being used to enable/disable the available **TEMPO SOURCES.**

If more than one **TEMPO SOURCE** is active, whichever source has sent the most recent **TEMPO** information will determine the devices current **TEMPO.**



CONTROLS

On the left of the **TEMPO MANAGER** you can tap on the current active **TEMPO** to enter a manual numerical value for the **TEMPO**. You can tap the 'TAP TEMPO' button with some degree of regularity in order to learn a **TEMPO.**

The right side of the **TEMPO MANAGER** determines which **TEMPO** sources are currently active. They can be tapped to enable/disable them.

<u>Three **TEMPO** sources are currently available:</u>
AUDIO - The BPM analyzed from incoming **AUDIO**.
MIDI - The **TEMPO** sent by an incoming **MIDI CLOCK** signal.
MANUAL - The value entered manually in the **TEMPO MANAGER** of via tapping 'TAP TEMPO'

The most recent **TEMPO** signal coming from an active **TEMPO** source will determine the device's **TEMPO,** overwriting previous information.

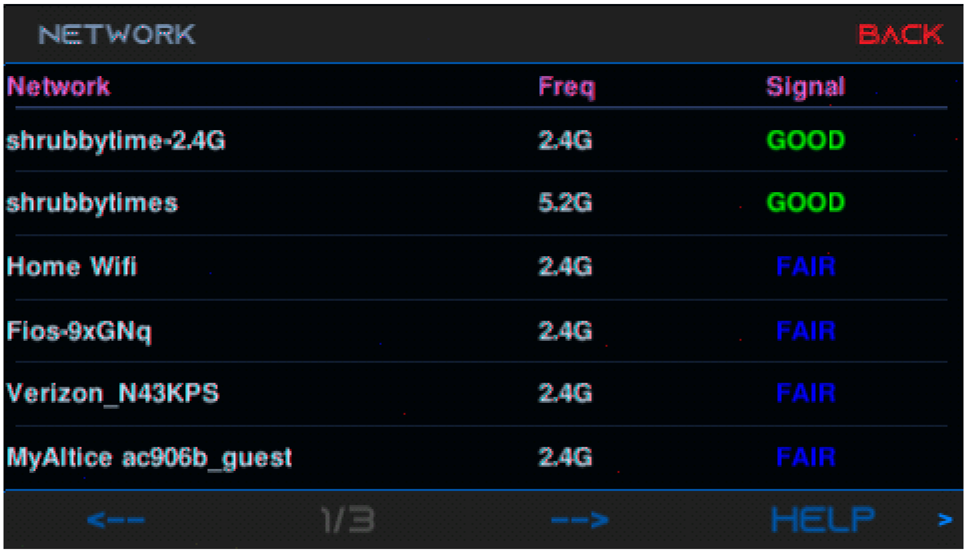
NETWORK MANAGER & INTERNET

OVERVIEW

The **NETWORK MANAGER** is used for connecting to the internet in order to **UPDATE** the software running on the synthesizer. The synth can be connected to the internet either via ethernet or WIFI.

To connect via ethernet simply plug an ethernet cable into the jack on the left panel of the device, and proceed to connect the synth to your router or modem. If you are successfully connected to the internet via ethernet, the **NETWORK MANAGER** will display a notification and WIFI networks will not be displayed.

To connect via WIFI, first the external WIFI dongle (included) must be plugged into a **USB** port on the left panel of the device. Then you navigate to the **NETWORK MANAGER** via the **SETTINGS** page.



Network	Freq	Signal
shrubbytime-2.4G	2.4G	GOOD
shrubbytimes	5.2G	GOOD
Home Wifi	2.4G	FAIR
Fios-9xGNq	2.4G	FAIR
Verizon_N43KPS	2.4G	FAIR
MyAltice ac906b_guest	2.4G	FAIR

The **NETWORK MANAGER** shows a list of detected WIFI networks, as well as their frequency bands and signal strengths. If you are currently connected to the internet via WIFI, the network you are connected to will be highlighted. When the synth first starts, if the WIFI dongle is inserted, it will attempt to connect to the last WIFI network that was successfully connected. Use the pagination controls in the **COMMAND BAR** to select additional pages or WIFI networks.

To attempt to connect to a WIFI network, simply tap on it. If the synth has successfully connected to this network in the past, it will use the previously entered password. If it has not, you will be prompted for a password. Hold the **SHIFT BUTTON** for capital letters, and the **NAV BUTTON** for symbols.

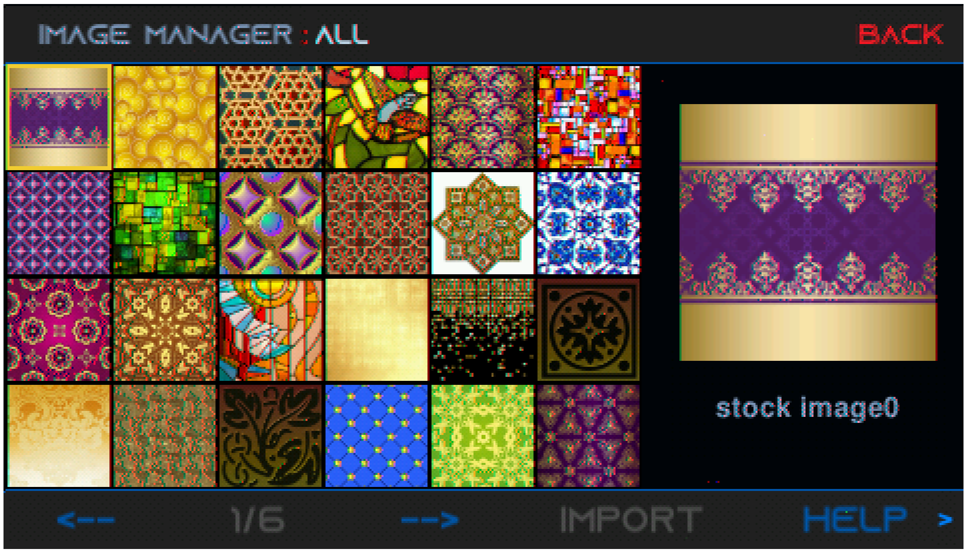
IMAGE MANAGER & IMAGES

OVERVIEW

Several **MODULES** in the synth have access to a collection of **IMAGES** stored on the device. The **IMAGE MANAGER** allows you to manage this collection. The **IMAGE MANAGER** is accessed via the **SETTINGS** page.

IMAGES automatically get assigned to **PATCHES/MODULES** in the synth. **IMAGES** are available by loading the 'image tex' and 'image buffer' **MODULES**, which wrap all **IMAGES**, and then selecting the specific image within those **MODULES.**

IMAGES in the device are organized into 'banks,' and at the top of the **IMAGE MANAGER**, you can see which bank is currently selected, the default being 'All'. All banks of **IMAGES** are active and available at all points in time throughout the synth, banks are used only for organization while managing **IMAGES**.



The **IMAGE MANAGER** shows the currently live bank of **IMAGES** on the left. If there are additional pages, pagination controls will be available in the **COMMAND BAR.** The **GOLD CURSOR** is visible in this window, and whichever **IMAGE** is highlighted will be shown in a preview on the right side of the page.

NOTE: The maximum number of images that can be loaded onto the device is 256. The device comes with 128, and these can be deleted if you need additional space. This is a known limitation of the device, and will be improved upon in a future **SYSTEM UPDATE**.

IMPORTING IMAGES

IMAGES are imported via **USB.** First, transfer your **IMAGES** onto a **USB** drive, then insert the drive into one of the **USB** slots on the left panel of the synth. Most **USB** drive file system formats are supported, with the exception of NTFS. If in doubt, use fat32 or exfat.

NOTE: Your **USB** drive *MUST* be plugged into the synth ***BEFORE IT IS POWERED ON**.* Otherwise the drive will *not* be recognized.

Your images must be 'png' or 'jpg' format, and end in the required file extension. All image files on your drive will be found, and imported as a group. All resolutions are supported, the images will be centered and either magnified and cropped as needed.

To proceed, navigate to the **IMAGE MANAGER**. If your drive has been recognized, and images are available for import, the **IMPORT COMMAND** will be active in the **COMMAND BAR.** If your drive is not recognized, or no images are found, the **IMPORT COMMAND** will be grayed out.

Tap **IMPORT** to continue. You will then be prompted to enter a bank name. This is simply for organization. Proceed to import the images. This may take a few minutes, please do not unplug or otherwise interrupt the process. Afterwards, the synth will reboot.

Input resolution for the images doesn't matter. Imported images are cropped and scaled automatically, maintaining aspect ratio. The largest square region is taken from within the image, and then this region is scaled up or down to 1024x1024 which is the resolution used internally.

COMMANDS

IMPORT: Import images via **USB**
HELP: Show help.
ALL: Tap this to open a selector for the current **IMAGE** bank. When all **IMAGES** are currently being shown, this **COMMAND** will display 'ALL,' when a specific bank is selected, the text will display the bank name.
DELETE: Deletes the **IMAGE** that is currently highlighted by the **GOLD CURSOR.**
DELETE ALL: Deletes all **IMAGES** that are a member of the currently selected bank.

SETTINGS

App Version: Opens up a screen with further details about your synth, including its serial number
Network: Opens the **NETWORK MANAGER**
Image Manager: Opens the **IMAGE MANAGER**
Screensaver: Settings for the **SCREENSAVER**
Tempo: Opens the **TEMPO MANAGER**
Transition Spd: The amount of time it takes to transition from the current **PATCH** to the next **PATCH**.
Auto Mapping: Opens up the setting for **AUTO MAPPING**, allowing you to enable and disable it for specific **UI VIEWS**, and for specific **I/O** types.
Play UI: The settings for the **PLAY UI.** See the **PLAY** section of the manual for details.
Perform UI: The settings for the **PERFORM UI.** See the **PERFORM** section of the manual for details.
Create UI: The settings for the **CREATE UI.** See the **CREATE** section of the manual for details.
Start Page: The initial initial page the synth starts on
Global Macro Sensitivity: Adjusting this setting will change the sensitivity of all **MACRO** controls. 1.0 is the default value. If you increase the sensitivity to 2.0, the **KNOBS** on the device will be twice as sensitive, and will need to be turned half as much for the same effect.
Calibrate Screen: A calibration widget for the touchpanel. Be careful when using this. If you ever entirely break calibration, insert a PC keyboard in the a **USB** port, and press 'c'
Max Fan Spd: Sometimes the fan sounds bad. If it's bothering you, turn this down. Your synth will probably be safe, even with it entirely off.
Force 720p Out: Attempts to open the output display at 720p. This is not a guarantee. This will ***not*** make the synth faster, as it only affects output resolution, not internal resolution of the synth.
Force 30fps Out: Attempts to open the output display a fixed 30pfs refresh rate. This is not a guarantee.
Soft Reset: Resets all **SETTINGS** back to factory defaults, as well as all **MAPPINGS** for the **DEFAULT PATCHES**. Your custom **PATCHES** will not be affected in any way.

MODULE REFERENCE

CATPARTY

CATPARTY: This is the root **MODULE** of the CATPARTY **INSTRUMENT**. It's role is to house the CAT and PARTY **INPUT MODULES,**and compose their outputs into the final output of the **INSTRUMENT**. The CAT **MODULE** creates an image of a cat, and the PARTY **MODULE** creates a 'party', which is basically just an effect. The two are then combined. CATPARTY**!**

CAT: The CAT **MODULE** outputs an RGB image of a cat, and provides some basic spatial transformation options.

- **cat img:** Allows the choice of any of a number of cats.
- **x ofs:** An x coordinate offset for the image. Allows for horizontal translation.
- **y ofs:** A y coordinate offset for the image. Allows for vertical translation.
- **theta:** An angle by which to rotate the image.
- **scale:** A scale/zoom factor.

PARTY: The PARTY **MODULE** provides a 'party' which is a composition of 2 effects. These effects are provided by the <u>EFFECT1</u> & <u>EFFECT2</u> **INPUT MODULES.** The amount of each effect can be individually controlled.

- **effect amt1:** The amount of the first effect to apply.
- **effect amt2:** The amount of the second effect to apply.

FRAXTAL

FRAXTAL: This is the root **MODULE** of the FRAXTAL **INSTRUMENT.** It's function is to house the <u>SEED</u>, <u>RECURSE</u>, <u>COLOR</u>, and <u>MODULATE</u> **MODULES**, and compose their outputs into the final output of the **INSTRUMENT.** FRAXTAL is a digital simulation of video feedback. The <u>SEED</u> **MODULE** procedurally creates an RGBA image which is then fed into the <u>RECURSE</u> **MODULE** which is a parameterized feedback loop. This result is then passed to the <u>COLOR</u> **MODULE** which colorizes the output into the final result. The colorization step is not incorporated back into the feedback (by default). The <u>MODULATE</u> **MODULE** is special and creates 4 patterns used to provide spatial variance to parameters.

SEED: The SEED **MODULE** creates a procedural RGBA image, which is the 'seed' content for the recursive simulation. It has numerous **MODULE INPUTS** and **PARAMETERS**.

INPUTS

- <u>SHAPE:</u> Creates a 2D shape, defining regions of transparency and opacity. Also generates XY texture coordinates and a height value used for 3D lighting calculations.
- <u>TEX:</u> Texturizes the opaque regions generated by <u>SHAPE</u> using the provided XY texture coordinates.
- <u>STRIPE:</u> A fixed function effect which creates a 'stripe' in the <u>SHAPE,</u> a rectangular region of differing color or opacity.
- <u>EFFECT:</u> An effect which is applied to the final image.
- <u>WT:</u> A waveform which is used to transform the 3D height & shading based effect, from the inside of the shape to its edge.
- <u>AT:</u> A waveform which is used to add transparency to the opaque regions, from the inside of the shape to its edge.

PARAMETERS

- **a**: A 2D parameter/complex number in the linear equation a * z + b. This allows you to scale and rotate the entire **MODULE** output.
- **b**: A 2D parameter/complex number in the linear equation a * z + b. This allows you to translate/offset the entire **MODULE** output.

- **tx_a**: A 2D parameter/complex number in the linear equation tx_a * z + tx_b. This allows you to scale and rotate the texture coordinates for the TEX **MODULE INPUT.**
- **tx_b**: A 2D parameter/complex number in the linear equation tx_a * z + tx_b. This allows you to translate/offset the texture coordinates for the TEX **MODULE INPUT.**
- **height**: A scale factor which is multiplied by the height output of the SHAPE **MODULE INPUT.** Only useful if lighting is enabled.
- **effect_amt**: The amount of the EFFECT to apply

SHAPE - These **MODULES** generate a variety of 2D shapes for use in the SEED **MODULE.**

BORDER: A square border.

- **width:** The width of the shape.
- **inner width:** The inner width of the shape, creating an annulus when it's non-zero

LINES LR: A pair of lines with a controlable distance between them.

- **width:** The width of the shape.
- **offset:** Central offset for various shape functions.

LINES INNER: A pair of perpindicular lines.

- **width:** The width of the shape.

LINE: A line. All alone.

- **width:** The width of the shape.

CIRCLE: A circle with controllable inner and outer width.

- **width:** The width of the shape.
- **inner width:** The inner width of the shape, creating an annulus when it's non-zero
- **N:** number of divisions the texture will take around the circle.

TRIANGLE: A triangle shape.

- **width:** The width of the shape.
- **inner width:** The inner width of the shape, creating an annulus when it's non-zero

SQUARE: A square shape.

- **width:** The width of the shape.
- **inner width:** The inner width of the shape, creating an annulus when it's non-zero

SQUARE2: A square shape with different texture coordinates.

- **width:** The width of the shape.

HEXAGON: A hexagon shape

- **width:** The width of the shape.
- **inner width:** The inner width of the shape, creating an annulus when it's non-zero

TRI LACE: A laced triangle shape.

- **width:** The width of the shape.
- **offset:** Central offset for various shape functions.

BORDER LACE: A laced border.

- **width:** The width of the shape.
- **offset:** Central offset for various shape functions.

BORDER LACE MASK: A different laced border.

- **width:** The width of the shape.
- **offset:** Central offset for various shape functions.

SOLID: A solid fill.

FADE: A gradient from left to right.

TEX - These **MODULES** provide a variety of texture sources for use in the SEED **MODULE.**

GEN TEX: Generates a texture based on one of a number of simple algorithms.

GTX_FN

GTX FN0-6: Simple procedural textures

GEN_GUIDE: A more advanced procedural texture which uses waveforms to generate a texture.

IMAGE TEX: Loads a texture from the library.

- **image:** Image selector

RECURSION TEX: The output of the RECURSE **MODULE**.

SEED TEX: The output of the SEED **MODULE**.

OUTPUT TEX: The final output of the **INSTRUMENT**.

VIDEO TEX: Streams a texture from a **VIDEO** source.

MOD TEX: One of the 4 **MODULATION** PATTERNS.

CAT TEX: CATS!!

- **cat img:** Image selector

FFT TEX BW: A black/white texture created from the input **AUDIO**.

- **band**: Which frequency analysis 'band' to use. Not a numerical frequency selector, more a selector of which audio analysis method is used.

FFT TEX RBW: A rainbow texture created from the input **AUDIO**.

- **band**: Which frequency analysis 'band' to use. Not a numerical frequency selector, more a selector of which audio analysis method is used.

CV TEX BW: A black/white texture created from a **CV** channel.

- **cv chn:** CV channel selector

CV TEX RBW: A rainbow texture created from a **CV** channel.

- **cv chn:** CV channel selector

OSC TEX (1D/2D): Generates a texture using the oscilloscope. Can be either 1D or 2D

STRIPE - These **MODULES** generate simple rectangular regions which override the color/texture or transparency generated by the SEED **MODULE.**

XY STRIPE: Generates a rectangular colored 2D pattern and applies it to a sub-region of the generated SEED image.

- **stripe w:** The width of the STRIPE.
- **stripe dir:** The direction of the STRIPE, 0 is horizontal, 1 is vertical.
- **pos_phase:** Phase offset for STRIPE position.
- **rgb_phase:** Phase offset for STRIPE rgb animation.
- **stripe mod:** 0 = STRIPE overwrites SEED, 1 = STRIPE is multiplied to SEED
- **stripe mask:** 1 = use the SEED's pseudo 3D shading

- STRIPE RGB - These **MODULES** generate simple animated rectangular RGB patterns

  XY0: Solid white and opaque.

  XY1: Solid black and transparent.

  XY2: Horizontal rainbow.

  XY3: Vertical rainbow.

  XY4: Red & Green

  XY5: RGB gradient.

  XY6: Quantized BW gradient.

  XY7: Quantized BW gradient with phasic animation.

  XY8: Rainbow with phasic animation.

  XY9: B&W with phasic animation.
- STRIPE A - A waveform that determines the transparency of the STRIPE, along the length of the STRIPE
- STRIPE MX - A waveform that determines the mixture of the STRIPE vs TEX, along the length of the STRIPE

ALPHA STRIPE: Generates a simple 2D pattern and uses it to override the opacity channel of the generated SEED image.

- **stripe dir:** The direction of the STRIPE, 0 is horizontal, 1 is vertical.

RECURSE: The RECURSE **MODULE** implements a digital feedback loop in a highly parameterized fashion. First, the output of the SEED **MODULE** is blended with the previous output of the RECURSE **MODULE.** This result is then transformed/deformed spatially. This result is put through an EFFECT and is then stored for use in the next animation frame.

INPUTS

- EFFECT: An effect which is applied to the final image.
- BLENDER: Blends the SEED RGBA image with the previous output of the RECURSE **MODULE** in any of a number of ways.
- FUNCZ: A 2D complex(z->z) equation, used to spatially transform the RECURSE **MODULE's** output in a mathematical fashion.
- KALEID: A 2D symmetry transformation, used to add symmetry to the RECURSE **MODULE's** output.

PARAMETERS

- **c0-c5**: 2D **PARAMETERS/**complex numbers that are used as coefficients in the FUNCZ and KALEID mathematical equations.
- **alpha**: Used in the BLENDER **MODULE**. This value is multiplied by the alpha channel of the SEED RGBA output to determine the mix of SEED <-> PREVIOUS FRAME
- **effect amt**: The amount of the EFFECT to apply

BLENDER

ALPHA BLEND: Basic alpha compositing of the seed image with the recursion frame.

SAT BEND: Basic alpha compositing of the seed image with the recursion frame, with saturation boost.

DARKEN: Darken blending mode.

MULTIPLY: Multiply blending mode.

COLOR BURN: Color burn blending mode.

LINEAR BURN: Linear burn blending mode.

LIGHTEN: Lighten blending mode.

SCREEN: Screen blending mode.

COLOR DODGE: Color dodge blending mode.

DIFFERENCE: Difference blending mode.

EXCLUSION: Exclusion blending mode.

BLEND MISC0: Experimental blending mode.

REDUCE BLEND: Alpha compositing of the seed image with the recursion frame, using a function to generate alpha from the current frame.

ONLY SEED: No blending, only use the seed image.

FUNCZ - These **MODULES** all are 2D spatial transformation functions.

FN0-34: Miscellaneous mathematical equations

FNX: A mathematical equation that blends a second equation with the identity.

- **a:** how much blending to do

KALEID - These **MODULES** all are 2D spatial transformation functions, but with high degrees of symmetry

FOLD 1X: Fold along the central X axis.

FOLD 1Y: Fold along the central Y axis.

FOLD 2-8: N-way radial folds

FOLD LINESY: Fold along two offset horizontal lines.
FOLD LINESX: Fold along two offset vertical lines.
FOLD LINESXY: Fold along two pairs of offset horizontal & vertical lines.
FOLD SQ: Fold along two offset horizontal & vertical lines.
NOFOLD: no folding

INTERFERENCE

INTERFERENCE: This is the root **MODULE** of the INTERFERENCE **INSTRUMENT.** It's function is to house the WAVE2D, INTERFERE, COLOR, and MODULATE **MODULES**, and compose their outputs into the final output of the **INSTRUMENT.** A 2D wavefront is generated by the WAVE2D **MODULE** and peaks and valleys of multiple copies of it constructively and destructively interfere with each other in the INTERFERE **MODULE**. This result is then passed to the COLOR **MODULE** which colorizes the output into the final result. The MODULATE **MODULE** is special and creates 4 patterns used to provide spatial variance to **PARAMETERS**.

WAVE2D: Generates an animated 2D wave pattern. The functionality of this **MODULE** is fairly complex and is implementing a non-trivial mathematical equation. The general overview is that, first a spatial symmetry is added via the KALEID **MODULE.** Then the GUIDE **MODULE** creates a simple 1D spatial pattern. This pattern is then fed into the WAVE **MODULE** for additional variance. Several **PARAMETERS** are used during this process as mathematical constants, giving additional control of the output.

- **w0:** A constant used in the equation w0 * z + w1. This can be used to scale and rotate the entire output.
- **w1:** A constant used in the equation w0 * z + w1. This can be used to translate the entire output.
- **wK:** Used to set the center of the KALEID **MODULEs** operation. Sort of.
- **wvec:** Determines the direction of propagation of the 2D wave output, as well as its frequency.
- **origin:** Determines the origin of the 2D wave output.
- **phase:** A phase factor added to the output of GUIDE, before being fed into WAVE
- **wave height:** The 'height' of the wave, only used for 3D lighting calculations.

INTERFERE: This **MODULE** generates multiple copies of the WAVE2D waveform, arranges them in space, animates them, and adds them together to create the interference pattern. The result is then palletized to give color variation.

- **pallet:** A numerical pallet selector
- **pallet amt:** The amount of palletization to apply
- **lz sc:** The amount of lz to sc
- **wave anim:** Used to animate the wave front in space. Intended to be used with an infinite scroll LFO in order to get a wave front that is constantly propagating through space at a regular pace.
- **anim ofs:** An offset that is added to **wave anim** in order to get the parameter which is used to animate the wave front.

SPREAD
The WAVE2D **MODULE** creates a 2D waveform, and the **INTERFERE** module combines some copies of them. The various SPREAD **MODULES** determine how the copies of the waveforms are arranged.

FAN OUT: This **MODULE** fans the waves out in an outwardly fanning fashion.
ROT OUT: This **MODULE** rotates the waves out in an outwardly rotating fashion.
XOSC: This **MODULE** oscillates the waves out in an x-ward direction.
ZMOT: This **MODULE** moves the waves in some fashion idk find out.
ZLIN: This **MODULE** moves the waves in a linear fashion.

ZLINROT: This **MODULE** moves the waves in a linearly rotating fashion.
JIBBERZ: This **MODULE** jibbers the waves all the way to the bank.
WANGDX: This **MODULE** wangs your grandma.
XYSKRIGGLEZ: This **MODULE** is full of hugs and skrigglez.
ZBLERB: This **MODULE** blerbs the waves out blerbingly.

WAVIBOI

WAVIBOI: This is the root **MODULE** of the WAVIBOI **INSTRUMENT.** It’s function is to house the SIGNAL1, SIGNAL2, BKGND**,** COLOR, and MODULATE **MODULES**, and compose their outputs into the final output of the **INSTRUMENT.** WAVIBOI is a 2-channel 3D oscilloscope. SIGNAL1 and SIGNAL2 generate 2 independent waveforms which are rendered onto the output of BKGND**.** This result is then passed to the COLOR **MODULE** which colorizes the output into the final result. The MODULATE **MODULE** is special and creates 4 patterns used to provide spatial variance to **PARAMETERS**.

NOTE: As of Q1 2024, WAVIBOI is still considered experimental. The INSTRUMENT will be fleshed out into its finished state around Q2-3 2024.

CRT: Generates a simulated CRT tube screen

- **Grid amt** - Amount of background grid to show
- **atten** - Attenuate scope background or not
- **HUE** - The hue of an HSV color used for the scope background.
- **SAT** - The saturation of an HSV color used for the scope background.
- **VALUE** - The value of an HSV color used for the scope background.

WAVEFORM: Generates a 1,2, or 3D waveform. The available **PARAMETERS** change depending on the number of dimensions.

- **signal width** - The rendered width of the waveform.
- **head spd** - The rate at which the scope head moves. This doesn't do anything if **follow cv** is enabled
- **follow cv** - Whether the oscilloscope head follows the most recent information read from the CV jacks. Ignores **head spd** when enabled.
- **pallet** - A numerical pallet selector
- **X freq** - Frequency of the X wave for the oscilloscope
- **X scale** - Magnitude of X wave for the oscilloscope
- **Y freq** - Frequency of the Y wave for the oscilloscope
- **Y scale** - Magnitude of Y wave for the oscilloscope
- **Z freq** - Frequency of the Z wave for the oscilloscope
- **Z scale** - Magnitude of Z wave for the oscilloscope
- **wave theta** - An amount of 3D rotation for the wveform.
- **wave phi** - An angle in spherical coordinates for the rotation axis of the waveform.
- **wave psi** - An angle in spherical coordinates for the rotation axis of the waveform.

EFFECTS

NO EFFECT: No effect
PERMUTE RBG: Returns color.rbg
PERMUTE BRG: Returns color.bgr
RAINBOW: Entropy & Sons is a proud ally of the LGBTQZX~ := ~ community.
INVERT: color = 1 - color

SCROB: I scrobbed your mother last night.
SPAZ: spaz
SPHR: A complex effect which does a rotation of a 'spherised' version of HSL space by an angle theta around an axis specified by phi & psi. Has several **PARAMETERS**

- **theta** - the angle of rotation about the axis
- **phi** - one of the angles of the rotation axis, in spherical coordinates
- **psi** - one of the angles of the rotation axis, in spherical coordinates

BLACKWHITE: Does what you think it does.
SATURATE: Saturates the color.
PASTEL: Pastelizes the color.
QUANTIZE: Quantizes the rgb color to the specified number of levels.

- **levels** - Num levels for color quantization.

RED: Makes things redier.
MAGENTA: Makes things magentaier.
RGB KEY: Sets opacity to the similarity of the input color to the specified RGB value. Only useful in the SEED module.

- **red** - The red value of an RGB color.
- **green** - The green value of an RGB color.
- **blue** - The blue value of an RGB color.
- **thresh** - How similar the input color needs to be to the RGB triplet to become transparent. 1 is exact similarity.
- **softness** - How abruptly the opacity transitions around the threshold. 0 is an immediate transition

MIXER: Allows you to mix the current color value with the output of another **MODULE** somewhere in the **LIVE** network.

- BUFFER - The **MODULE** to mix in with the current color
- BLENDER - The method by which to blend the current color with the auxiliary **MODULE**
- **mix** - Mix control for color <-> buffer.
- **buf xofs:** Buffer lookup x offset
- **buf yofs:** Buffer lookup y offset
- **buf sc:** Buffer lookup scale factor

ADJUST: Exposes an array of color adjustment functions.

- **brightness** - Brightness adjustment factor.
- **contrast** - Contrast adjustment factor.
- **saturate** - Saturation adjustment factor.
- **exposure** - Exposure adjustment factor.

MONOCHROME: REDUCES the input color to 1 dimension, and then maps this to a single color.

- REDUCE - Downsamples an RGB source into a 1D output in some fashion.
- **HUE** - The hue component of the output color.
- **SAT** - The saturation value of the output color.

WAVEOFS: Takes the input RGB color, adds a phase to each channel, and puts the result through a waveform.

- WAVE - The waveform used for processing
- **phase** - Phase of waveform.

COLORING & LIGHTING

COLOR: The COLOR **MODULE** colorizes the final **INSTRUMENT** output in several stages before it is sent out of the synth.

INPUTS

- POST FN: This is effectively an EFFECT with an additional 'depth' value which is provided to the COLOR **MODULE.**
- EFFECT: An effect which is applied to the final image.
- LIGHTING: One of two **MODULES**, either the NO LIGHT **MODULE**, which does nothing, or the **LIGHT MODULE**, which applies a pseudo 3D lighting effect to the input of the COLOR **MODULE.**

PARAMETERS

- **depth scale**: A scale factor for the 'depth' value that is provided to the COLOR **MODULE.**
- **hue offset**: A hue rotation amount that is provided to the POST FN
- **post amt**: The amount of the POST FN to apply
- **effect amt**: The amount of the EFFECT to apply
- **orig value**: The POST FN changes the value(darkness) of the input frame. This variable controls how much of the original value is maintained.
- **fade**: A global fade out control
- **red level**: A global red level control
- **green level**: A global green level control
- **blue level**: A global blue level control
- **mod zx**: Final x-coordinate modulation target. Translates the final image horizontally.
- **mod zy**: Final y-coordinate modulation target. Translates the final image horizontally.
- **mod zr**: Final radial modulation target. Scales the final image.

POST FN - These **MODULES** colorize the final output of the system. Very similar to EFFECT's, except they take as an additional input a 'depth' value provided to the COLOR **MODULE.**

POSTFN ID: Returns the original color.

BASIC HUE: Performs a hue rotation on the input color proportional to the depth. A more performant but less feature rich version of hsv.

HSV: Performs a hue rotation of the input color proportional to the depth, as well an optional windowing of the output hue, and scaling of the saturation & values.

- **saturation:** Saturation scale factor.
- **value:** Value scale factor
- **hue window:** Limits the hue to a finite range. A value of 0.0 means no windowing is in effect

COLOR SPHERE: Converts the input color to a spherical color space & performs a rotation proportional to the depth.

- **phi** - one of the angles of the rotation axis, in spherical coordinates
- **psi** - one of the angles of the rotation axis, in spherical coordinates

METAL: Metalizes the input image as a function of the schiz matrix.

- **phi** - one of the angles of the rotation axis, in spherical coordinates
- **schiz -** rectabular schiz coefficient
- **infinity hue -** Hue at infinity

XYZ PAL: Generates a pallet from three specified RGB colors - X, Y, & Z

- **red X:** The value of the red color channel for color X.
- **green X:** The value of the green color channel for color X.
- **blue X:** The value of the blue color channel for color X.
- **red Y:** The value of the red color channel for color Y.

- **green Y:** The value of the green color channel for color Y.
- **blue Y:** The value of the blue color channel for color Y.
- **red Z:** The value of the red color channel for color Z.
- **green Z:** The value of the green color channel for color Z.
- **blue Z:** The value of the blue color channel for color Z.

TURBO: Uses the hue of the input color as an index into a color pallet.
PALLET 0-30: Uses the hue of the input color as an index into a color pallet.

LIGHTING
NO LIGHT: Does nothing
LIGHT: Implements the Blinn-Phong (ambient/diffuse/specular) lighting algorithm to provide a pseudo 3D effect depending on the position of a virtual light source.

- **ambient:** Amount of ambient lighting.
- **diffuse:** Amount of diffuse lighting.
- **specular:** Amount of specular lighting.
- **shiny:** The shininess of the 'material'
- **sp hue:** The hue of specular highlights
- **sp mix:** The mix between the specular hue and light color
- **light red:** The red component of the light.
- **light green:** The green component of the light.
- **light blue:** The blue component of the light.
- **light x:** The x coordinate of the light.
- **light y:** The y coordinate of the light.
- **light z:** The z coordinate of the light.
- **atten:** The amount of attenuation as the distance from the light increases.

MODULATION

MODULATE: Generates 4 1D spatial modulation patterns, via 4 copies of the PATTERN **MODULE**. These patterns are used to provide spatial variation to numerical **PARAMETERS** throughout the synth.

PATTERN: Generates a **MODULATION** PATTERN, via mixing a 2D waveform with an image BUFFER**.** The 2D waveform is generated via a mathematical algorithm which begins by using the GUIDE **MODULE** to create a simple 1D spatial pattern. This pattern is then fed into the WAVE **MODULE** for additional variance. Several **PARAMETERS** are used during this process as mathematical constants, giving additional control of the output.

Additionally, an image BUFFER can be mixed in. This BUFFER can has several options, including image & video sources, buffers used elsewhere in the synth, and even the output of other **MODULATION** PATTERN. Since the output of a **MODULATION** PATTERN is 1D, and images are 3 (RGB), the BUFFER must be downcast to 1D, which is done but the REDUCE **MOULE.**

- **mod mix:** Mix control for waveform <-> texture buffer.
- **invert:** Inverts the result
- **buf xofs:** Buffer lookup x offset
- **buf yofs:** Buffer lookup y offset
- **buf sc:** Buffer lookup scale factor
- **wvec:** Determines the direction of propagation of the 2D wave output as well as its frequency.
- **origin:** The origin point of a 2D waveform.

- **wmax:** Maximum of a waveform.
- **wmin:** Minimum of a waveform.
- **phase:** Phase of a waveform.
- **freq:** Spatial frequency of a waveform.

BUFFERS - Various RGB image textures available in the synth

NO BUFFER: No buffer.

RECURSION BUFFER: The output of the RECURSE **MODULE.**

SEED BUFFER: The output of the SEED **MODULE.**

INTERFERE BUFFER: The output of the INTERFERE **MODULE.**

WAVE2D BUFFER: The output of the WAVE2D **MODULE.**

OUTPUT BUFFER: The final output of the **INSTRUMENT.**

MOD BUFFER: One of the 4 MODULATION PATTERNS.

IMAGE BUFFER: A buffer containing an **IMAGE.**

VIDEO BUFFER: A buffer containing one of the 4 **VIDEO** device inputs.

REDUCE - Downsamples an RGB source into a 1D output in some fashion.

VALUE: Returns the value(HSV) channel of the input buffer.

LUM: Computes a perceptual(weighted) luminance channel from the input buffer.

INV LUM: Returns 1.0 - the luminance channel of the input buffer.

FUNC0: A 1D mathematical function of the input buffer.

FUNC1: A 1D mathematical function of the input buffer.

FUNC2: A 1D mathematical function of the input buffer.

HUE: Returns the hue of the input buffer.

RGB SIM: Returns the similarity of the input buffer to the specified RGB value.

- **red** - The red value of an RGB color.
- **green** - The green value of an RGB color.
- **blue** - The blue value of an RGB color.
- **thresh** - How similar the input color needs to be to the RGB triplet. 1 is exact similarity.
- **softness** - How abruptly the output transitions around the threshold. 0 is an immediate transition

CONST: Returns a constant value

- **const:** the constant to return

RX RED: Returns the red color channel of the input buffer.

RX GREEN: Returns the green color channel of the input buffer.

RX BLUE: Returns the blue color channel of the input buffer

1D & 2D WAVES

GUIDE - Creates a basic 1D pattern from a 2D(X/Y) input.

RADIAL: The distance from the origin.

WSQUARE: A square pattern.

LINEAR: A linear gradient.

- **theta:** the angle of the gradient

LINEAR MIR: A mirrored linear gradient.

- **theta:** the angle of the gradient

MOD: Idk but it looks cool

- **N:** the number of discrete steps

SPIRAL: A spiral.

THETA: The angle of the incoming coordinate.

- **N:** the number of times to wrap around the origin

THETA MIR: The angle of the incoming coordinate, mirrored

- **N:** the number of times to wrap around the origin

RADIAL COSZ: The magnitude of the complex cosine function.

LOGRAD: The logarithm of the distance from the origin.

CONST2D: A constant value.

- **const:** the constant to return

WAVE - Simple 1 dimensional waveforms.

COS WAVE: A cosine wave.

SIN WAVE: A sine wave.

TRI WAVE: A triangle wave.

SAW WAVE: A saw wave.

SQ WAVE: A square wave with controllable pulse width.

- **pulse_width:** Square wave pulse width, from 0 to 1.

SQRD WAVE: A waveform in the shape of the square (x * x) function.

SQRT WAVE: A waveform in the shape of the sqrt function.

STEP WAVE: A step wave with a controllable number of steps.

- **N:** the number of discrete steps

BAND WAVE: A waveform generated by the frequency(FFT) bands of the incoming audio waveform. An audio analysis band is selected from those available. Its value is multiplied by **band gain**, and possibly inverted. There is an optional threshold. If its value is non-zero, the output of this module will be 1 if the selected frequency band is greater than the threshold value, otherwise zero.

- **band**: Which frequency analysis 'band' to use. Not a numerical frequency selector, more a selector of which audio analysis method is used.
- **thresh**: The threshold. When this is 0 this is turned off.
- **inv**: A flag to possibly invert the results.
- **band gain**: Controls amplification of audio band waveforms.

STAT WAVE: A wave derived from statistics of the incoming audio waveform.

- **stat band:** Which statistic band to use.

CV WAVE: A waveform generated by the last N seconds of CV data.

- **cv chn:** CV channel selector

POS WAVE: A constant waveform with value = 1.0.

ZERO WAVE: A constant waveform with value = 0.0.

NEG WAVE: A constant waveform with value = 0.0.

CONST WAVE: A constant waveform with a controllable value.

- **const:** the constant to return

COS WAVE INV: An inverted cosine wave.

SIN WAVE INV: An inverted sine wave.

TRI WAVE INV: An inverted triangle wave.

SAW WAVE INV: An inverted saw wave.

SQ WAVE INV: An inverted square wave with controllable pulse width.

- **pulse_width:** Square wave pulse width, from 0 to 1.

SQRD WAVE INV: An inverted waveform in the shape of the square (x * x) function.

SQRT WAVE INV: An inverted waveform in the shape of the sqrt function.

STEP WAVE INV: An inverted step wave with a controllable number of steps.

- **N:** the number of discrete steps

COS WAVE HF: The first half period of a cosine wave.

SIN WAVE HF: The first half period of a sine wave.
TRI WAVE HF: The first half period of a tri wave.
SAW WAVE HF: The first half period of a saw wave.
COS WAVE INV HF: The inverted first half period of a cosine wave.
SIN WAVE INV HF: The inverted first half period of a sine wave.
TRI WAVE INV HF: The inverted first half period of a tri wave.
SAW WAVE INF HF: The inverted first half period of a saw wave.
STAT WAVE INV: An inverted wave derived from statistics of the incoming audio waveform.

- **stat band:** Which statistic band to use.

CV WAVE INV: An inverted waveform generated by the last N seconds of CV data.

- **cv chn:** CV channel selector

SIGNALS REFERENCE

MIDI

Note On: A **MIDI** Note event has been received. The **SIGNAL'S** value is the **MIDI** velocity, for Note On events. The value is 0 for Note Off events. The **CHANNEL** is the the note, from 0 to 127
MIDI CC: A **MIDI** CC event has been received. The **SIGNAL'S** value will be the CC value / 127.0, from 0 to 1. The **CHANNEL** is the **MIDI** CC number.
Program Change: A **MIDI** Program Change message has been received. The **SIGNAL'S** value is always 1.0, and the **CHANNEL** is the **MIDI** Program number.
Pitch Bend: A **MIDI** Pitch Bend message has been received. The **SIGNAL'S** value is 2.0 * PITCH BEND AMT - 1.0, giving -1.0 for maximum pitch decrease, and 1.0 for maximum pitch increase. There is no **CHANNEL**
Key Track: Similar to Note On. A **MIDI** Note event has been received. The **SIGNAL'S** value is the **MIDI** note / 127.0. There is no **CHANNEL.**
Key Poly: Similar to Note On. A **MIDI** Note event has been received. The **SIGNAL'S** value is (# of MIDI keys that are currently pressed) / 10.0 There is no **CHANNEL.**

CV

CV Signal: A **CV** signal has been received. The **SIGNAL'S** value is the **CV** voltage / 10, a value from 0 to 1. The **CHANNEL** is the number of the **CV** jack. If a **CV** cable is plugged in, the synth will continuously receive this signal on the appropriate **CHANNEL**

UI

Macro: One of the 5 **BLUE KNOBS** has been turned. The **SIGNAL'S** value is the 'velocity' with which the **KNOB** has been turned, a value from (-1..1) which determines the direction and rate of turning. The **CHANNEL** is the **KNOB** number, a value from 1 to 15. Values 1-5 are the first 5 pages of **KNOBS,** etc.
Button: One of the 5 **LARGE BUTTONS** has been pressed. The **SIGNAL'S** value is 1 for **BUTTON** presses, and 0 for release. The **CHANNEL** is the **BUTTON** number, a value from 1 to 15. Values 1-5 are the first 5 pages of **BUTTONS,** etc.
PC Keyboard: A **PC KEYBOARD** signal has been received. The **SIGNAL'S** value is 1 for key presses, and 0 for release. I forget what the **CHANNEL** is, you should use the **LEARN** functionality for this.
Extern Detect: An external **VIDEO** source has been plugged into the synth, or removed. The **SIGNAL'S** value is 1 for detected, and 0 for removed. The **CHANNEL** is the **USB** port, from 1-4 that was plugged into.

AUDIO

Beat: A Beat has been detected by the synth in the incoming **AUDIO** stream. This is used by the **BEAT REACTIVITY** system. The value of the signal is always 1, and the **CHANNEL** is used to randomly select **ACTIONS** to trigger.

Beat Wave: An internally generated waveform that spikes every time a **BEAT** is detected and rapidly decays. Generated continuously by the synth while **AUDIO** input is available.

Fundamental Freq: An internally generated waveform whose value represents the fundamental frequency of the incoming **AUDIO** source. Generated continuously by the synth while **AUDIO** input is available.

Avg Volume: An internally generated waveform whose value represents the volume of the incoming **AUDIO** source, averaged over the last few seconds. Generated continuously by the synth while **AUDIO** input is available.

Volume: An internally generated waveform whose value represents the volume of the incoming **AUDIO** source, averaged over the past few milliseconds. Generated continuously by the synth while **AUDIO** input is available.

Max FFT Band: An internally generated waveform whose value represents the frequency band of the incoming **AUDIO** source with the most energy. Generated continuously by the synth while **AUDIO** input is available.

LF Energy: An internally generated waveform whose value represents the low band energy of incoming **AUDIO** source, averaged over the past few milliseconds. Generated continuously by the synth while **AUDIO** input is available.

MF Energy: An internally generated waveform whose value represents the mid band energy of incoming **AUDIO** source, averaged over the past few milliseconds. Generated continuously by the synth while **AUDIO** input is available.

HF Energy: An internally generated waveform whose value represents the high band energy of incoming **AUDIO** source, averaged over the past few milliseconds. Generated continuously by the synth while **AUDIO** input is available.

Noisiness: An internally generated waveform whose value represents the ‘noisiness’ of the incoming **AUDIO** source. Closer to 1 the closer the incoming **AUDIO** is to white noise. Generated continuously by the synth while **AUDIO** input is available.

Hit Rate: An internally generated waveform whose value represents the rate of ‘hits,’ the onset of any new sound in the **AUDIO** stream. This includes percussive elements as well as notes, chords, etc. Generally measures the complexity of incoming **AUDIO**, as a value from 0 to 1. Generated continuously by the synth while **AUDIO** input is available.

Hit Wave: An internally generated waveform which spikes every time a ‘hit’ is detected in the incoming **AUDIO** source, and then rapidly decays. Generated continuously by the synth while **AUDIO** input is available.

No **AUDIO SIGNALS (**aside from **BEAT)** provide a **CHANNEL.**

ACTIONS REFERENCE

VALUE

All **VALUE ACTIONS** transition the value of a **PARAMETER** or **PARAMETER SLICE** from its current value to a ‘target value,’ and then optionally back to its original value.

ALL **VALUE ACTIONS** SHARE THE FOLLOWING SETTINGS:

attack: The amount of time it takes for the **PARAMETER’S** value to change from where it started to the ‘target value’

release: If this is 0, the **PARAMETER** will remain at its 'target-value' once reached. Otherwise it will return to its original value over this amount of time.
path: Determines the path the **PARAMETER'S** value takes to the 'target value'. Either linear or smooth.

VALUE ACTIONS:
Set Value: Used to set the value of a **PARAMETER.** The **value** setting determines the 'target value' in the range (**min, max).** When **value=0**, the 'target value' is **min**, when **value=1**, the 'target value' is **max.**
Increment Value: Used to set the value of a **PARAMETER**, *relative to its current value.* The 'target value' is the current value of the **PARAMETER** + the **ACTION'S change** setting. This is further clamped between the provided **min** and **max** values.
AR Envelope: A simple attack-release style envelope. This is a *relative* **ACTION**, where the 'target value' is the current value of the **PARAMETER +** the **ACTION'S change** setting.
Toggle Value: Used to simply set the value of a **PARAMETER** to 1.0 - its current value. Useful for ON/OFF type **PARAMETERS.**
Reset Value: Used to set the value of a **PARAMETER** to its default value.
Random Value: Used to set the value of a **PARAMETER** to a random value in the **PARAMETERS** range.

LFO

All **LFO ACTIONS** create an **LFO** and assign it to a **PARAMETER** or **PARAMETER SLICE. LFOS** generate a regular waveform with variable **RATE** and range (**min, max)** and use this to control a **PARAMETER's** value. Only 1 **LFO** can be active per **PARAMETER(/SLICE)** at a time.

Retriggering an **ACTION** which creates an **LFO** will deactivate the **LFO**, and return the **PARAMETER** to its previous value.

ALL **LFOs** SHARE THE FOLLOWING SETTINGS:
rate: A time value that specifies how long it takes for the **LFO** to complete one full cycle.
phase: A value from 0 to 1 that specifies where the **LFO** begins in its complete **RATE** cycle.
attack: The amount of time it takes for the **PARAMETER'S** value to transition from its current value to the starting value of the **LFO** when the **LFO** is activated.
release: If non-zero, the amount of time it takes for the **PARAMETER'S** value to return from its current value to its value prior to the **LFO** being started when the **LFO** is deactivated.

ALL 1D **LFOs** SHARE THE FOLLOWING SETTINGS:
max: A value that specifies the maximum value the **LFO** is to take. If this value exceeds the **PARAMETERS** maximum value, the **LFO** will be clipped.
min: A value that specifies the minimum value the **LFO** is to take. If this value exceeds the **PARAMETERS** maximum value, the **LFO** will be clipped.

1D LFOS:
Sine Wave: Creates a sine wave LFO. Begins at (**max** + **min**) / 2, and proceeds to **max.**
Triangle Wave: Creates a triangle wave LFO. Begins at **min**, and proceeds to **max.**
Saw Wave: Creates a saw wave LFO. Begins at **min**, and proceeds to **max.**
Square Wave: Creates a square wave LFO. Begins at **max**, then switches to **min** at t=**rate/**2.
Constant: Outputs **max.** Ignores **min**
Step Wave: Starts at **min,** and increases stepwise over equally divided steps to **max,** controlled by the **steps** setting.

Infinite Scroll: Outputs 0 when started, and linearly increases by 1 for every **rate** unit of time. Ignores **max** and **min.**
Random: Outputs a constant(between **min** and **max)** value which changes randomly.

2D LFOS:
Rose 2D: A 2D rose curve. th = time, r = **a** * cos(**b** * th) + **c,** where **a, b, c** are user specified values.

SYSTEM

Load Patch: Loads a **PATCH** into the specified **LOCATION.** The **attack** setting specifies the transition time.
Next Patch: Loads the next **PATCH** in the **PATCH LIBRARY** into the specified **LOCATION.** This uses the active **LIBRARY TAG FILTERS** to determine which **PATCH** is next. The **del** setting specifies how many **PATCHES** to move forward(or backward) in the **LIBRARY.** The **attack** setting specifies the transition time.
Reset Patch: Loads the default **PATCH** for current **INSTRUMENT.** The **attack** setting specifies the transition time.
Random Patch: Loads a random from the **PATCH LIBRARY** into the specified **LOCATION.** This uses the active **LIBRARY TAG FILTERS** to determine which **PATCHES** are available. The **attack** setting specifies the transition time.
Random New: Generates a new **PATCH** for the current **INSTRUMENT**. This **ACTION** generates a new **PATCH**, using the current one as a starting point. This **ACTION** has extensive configuration options:

- **randomness:** How far to deviate from the existing **PATCH**. 0 = no deviation, 1 = full deviation.
- **attack:** The transition time from the current **PATCH** to the new one.
- **modulation:** Whether to add **MODULATION** while randomizing.
- **structure:** Whether to randomly swap **MODULES** while randomizing.
- **params:** Whether to add randomized **PARAMETERS** while randomizing.
- **lfos:** Whether to add random **LFOs** while randomizing.
- **automap:** Whether to use **AUTO MAPPING** while randomizing.

Save Next: Saves the **MODULE** at the specified **LOCATION** into a new **PATCH**, using the next available auto-generated name. An optional **TAG** can be provided that will be added to the new **PATCH.**
Toggle Modulation: Toggle **MODULATION** for the specified **PARAMETER** ON/OFF. The **channel** setting specifies the **MODULATION CHANNEL** to use, and the **method** setting controls the **MODULATION METHOD.**
Remove Modulation: Removes all **MODULATION** for the **MODULE** at the specified **LOCATION**, and reverts **MODULATED PARAMETERS** to their default values.
Halt LFOs: Removes all **LFOs** for the **MODULE** at the specified **LOCATION**, and reverts animated **PARAMETERS** to their default values. The **attack** setting determines how long this takes.
Toggle Auto: Toggle automation (the **SCREENSAVER)** ON/OFF.
Toggle Beat Events: Toggle **BEAT REACTIVITY** ON/OFF.
Toggle Camera: Toggles the video device in the specified **device slot** ON/OFF by loading/unloading the video source into the **BUFFER MODULE** of all 4 **MODULATION CHANNELS.** Simultaneously toggles **AUTOMAPPING** for **VIDEO I/O**. The **attack** setting determines how long this takes.
Cycle Cameras: Any **MODULE** in the **LIVE MODULE** network that is streaming from a **VIDEO** device slot will be switched to the next available **VIDEO** device slot. The **attack** setting determines how long this takes.
Enable Mapping: Toggles a separate **MAPPING** ON/OFF**.** Use the **mapping label** setting to specify the **label** of the **MAPPING** in question.
Random Automap: Loads a random **AUTOMAP** for the specified **signal** type. The **attack** setting determines how long this takes
Toggle Automap: Turns **AUTOMAPPING** ON/OFF for the specified **signal** type.
Clear Buffers: Might fix your shit if its spazzin

TROUBLESHOOTING

HELP, IT WONT START!

The first thing to do is make sure you've thoroughly read the Quick Start guide included with the synth and linked above. An important thing to note is that the synth takes about 30 seconds to fully start, and you might not see any obvious signs of life on the LCD for about half of that.

If the synth still isn't starting, or the internal lights don't come on at all, it's more likely than not a problem with one of the accessories/cables. Try swapping out the power supply or power cable with one from a cell phone, most should be fine. Or try a different HDMI cable.

If you're still having problems, either swing by our discord server, you can probably get relatively quick support, or email support@entropyandsons.com

HELP, THE SYNTH CRASHED!

Most synths won't see any crashing or instability, but there is some chance that you will run into a bug that causes it to crash. If this happens, please contact support@entropyandsons.com with any relevant info, or just plug go online with the synth so we can get a bug report. The synth is still under active development, so we'll get a fix out asap

After many years of development, the synth is incredibly stable, however a small (<1%) of devices have a hardware defect that makes them crash consistently. If your synth crashes with any regular frequency, it might need to be swapped out with a new one. Again, contact support@entropyandsons.com

HELP, VIDEO INPUT IS CRASHING THE SYNTH!

The **VIDEO** input system in the synth admittedly still needs some work. We are aware of issues with it, and are working towards getting them straightened out. The most important thing to know is not to unplug a **USB VIDEO** device while the synth is plugged in, as it will almost certainly cause the synth to crash. It is recommended that you plug your devices in before starting up the synth, and simply leave them in.

HELP, THE LCD ISN'T CALIBRATED AND I CAN'T RE-CALIBRATE IT!

You have two options in this case, you can either **FACTORY RESET** the synth, to restore default calibration, or you can insert a PC KEYBOARD into a usb port and press the 'c' key. If neither of these work, or if you continue to have issues, contact us at support@entropyandsons.com

HELP, THE LCD PANEL HAS SOME WEIRD BANDING ARTIFACTS!

Sometimes, after the LCD Panel has been running for a few hours, it develops some artifacts that don't go away. If this happens to your synth, contact us at support@entropyandsons.com and we'll replace it for you.

HELP, THE LITTLE FEET BROKE OR I LOST SOMETHING!

Contact us at support@entropyandsons.com and we'll send out replacements.

HELP, THE SCREENSAVER KEEPS GOING BLACK!

This is probably because your audio source isn't loud enough. See the Quick Start guide for info about audio levels.

HELP, THE SYNTH IS STUCK UPDATING!

In most circumstances, the synth should finish updating in under 3 minutes. If your internet connection is slow, or if you haven't updated the synth in a very long time, it may take upwards of 10 minutes. If the synth says it's updating for longer than 10 minutes, there may be some error. You are likely safe to unplug it and try again. If you run into any further issues, please contact us at support@entropyandsons.com

FACTORY RESET

Your device can be **FACTORY RESET,** via pressing and holding one of the two following button combinations while the synth is booting up. If successful, when the logo appears, it will also say “Performing Factory Reset” on the synth.

```
X means pressed.

(BLUE KNOBS x5 | GOLD KNOB)        X X X _ _ | _                  _ _ _ _ _ | X
(              | SHIFT BTN)                  | _      - OR -                | _
(LARGE BTNS x5 | NAV   BTN)        _ _ _ _ _ | X                  X _ X _ X | _
```

This will entirely wipe your device, restoring the software, saved patch library, and settings to factory defaults, however, before doing this, the synth will make a backup of your data. You can then attempt to update your synth again, and when successful, you will have the option to attempt to recover your data.

WARRANTY INFORMATION

If your Recursion Studio dies or runs into any serious issues within 2 years of purchase, and it’s our fault, we will replace/repair it and cover all costs, including shipping. If your synth is out of warranty, and it’s not, like, the year 2063, we will probably still take care of it for you. Just contact us at **support@entropyandsons.com**

If your Recursion Studio dies or runs into any serious issues, and it’s because you threw it at your enemies or dropped it into a vat of acid, we will still fix it for you, charging for materials & labor. It probably won’t be much. Again, just contact us at **support@entropyandsons.com**

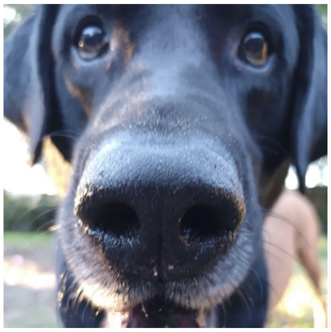
APPENDIX 1:

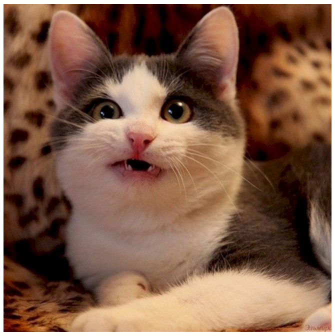
List CV Offset & Scaling Eurorack Modules

The device accepts 0 <-> 10v CV signals, so you need to scale/offset 0 <-> 5v & -5v <-> 5v signals appropriately. Courtesy of @scuto, a list of relevant modules:

- Noise Engineering Sinc Pravus
- Nerding MIA
- Erogenous Tones Levit8
- Intellijel Triplatt/Triatt/Quadratt
- Mutable Instruments' Shades & Blinds (or clones like After Later's Monocle)
- bubblesound Lvl+RM
- Frap Tools 321 (though trim pots)
- Befaco A*B+C
- WMD TBVCA
- Doepfer 138-c
- Xaoc Samara
- Vermona quadroPol
- Olivella Signos
- Toppobrillo Cluster
- Mazzatron Gain Invert Offset...
- Mutable Instruments' Frames

APPENDIX 2:





UNITED STATES POSTAL SERVICE
PRIORITY MAIL
MEDIUM FLAT RATE BOX
TRACKED • INSURED