

THE PARTY VAN

by Rodrigo Constanzo

<http://www.rodrigoconstanzo.com>

v1.1

The screenshot displays the 'THE PARTY VAN' software interface, version v1.1. The interface is organized into several sections:

- Top Row:** A title bar with a volume slider, 'setup' button, and 'THE PARTY VAN' logo by Rodrigo Constanzo.
- Input Section:** Includes 'input' (adc, open, stop), 'thru', 'sample', 'instru', and 'record' buttons.
- Presets Section:** Shows 'master' mode with 'morp' value 0.0, and a 'default settings' dropdown.
- Stutter and Shuffler:** 'stutter' and 'shuffler' modules with various parameters like 'pitch', 'rate', and 'kill'.
- Pitch Section:** 'pitch' module with 'pitch' value 0.4, '2 oct up', 'shift', 'blend', and 'mode' controls.
- Chopper Section:** 'chopper' module with 'chopper' and 'blend' values, and playback mode controls.
- Lofi Section:** 'lofi' module with 'lo-fi', 'sample' value 0.75, and 'bit' controls.
- Dirt Section:** 'dirt' module with 'dirt' and 'mode' controls.
- Filter Section:** 'filter' module with 'filter', 'lowpass 12', and 'blend', 'decay', 'damping' controls.
- Reverb Section:** 'reverb' module with 'reverb', 'stone room', and 'blend', 'decay', 'damping' controls.
- Middle Section:** Contains 'cocolase', 'cd', 'combine', 'pattern', 'brain', and 'presets' modules, each with its own set of parameters and visual waveforms.
- Bottom Section:** A large 'mlr' section featuring four parallel processing chains (1, 2, 3, 4) with 'record', 'replace', 'stop', and 'reverse' buttons, and a 'modifiers' dropdown.

INSTRUCTION MANUAL

DESCRIPTION AND OVERVIEW

The Party Van is a live sampling and performance instrument built using Max. It was programmed around the [Monome 64](#) and [Arc 2](#) controllers but includes mapping for an iPad running [TouchOSC](#) (custom layout included) as well as the [Keith McMillen Softstep](#) foot controller. The patch will work just fine with larger Monomes and Arcs. Additionally, the patch has MIDI learn functionality built in, so every parameter can be mapped to any MIDI controller. Finally, control output of a [Shbobo Shnht](#) can be mapped to The Party Van as well.

THE PARTY VAN INCLUDES SOME OF THE FOLLOWING:

- A variety of samplers/loopers oriented towards live performance
- Granular, concatenative, and convolution based synthesis on recorded buffers
- Buffer-based and real-time audio analysis used to dynamically generate intelligent presets
- Input/output effects
- Input stage convolution and amp simulation
- 8-bit sampler/looper based on the [ciat-lonbarde Cocolase](#) instrument
- Virtual CD skipping module based on “[The Chocolate Grinder](#)”, another one of my patches
- Concatenative synthesis module based on “[C-C-Combine](#)”, another one of my patches
- Attack-based sampling and triggering of effects
- Several synthesis-based playback instruments

SYSTEM REQUIREMENTS

Requires **Max6** and the externals listed below.

EXTERNALS USED

(all are included in the download)

Alex Harker externals:

<http://www.alexanderjharker.co.uk/>

Sigmund~ for pitch tracking in “Pitch” module:

http://www.maxobjects.com/?v=objects&id_objet=4713

Individual externals:

framerank~ for mp3ification

morphfilter~ for Reverb module

multiconvolve~ used in various modules

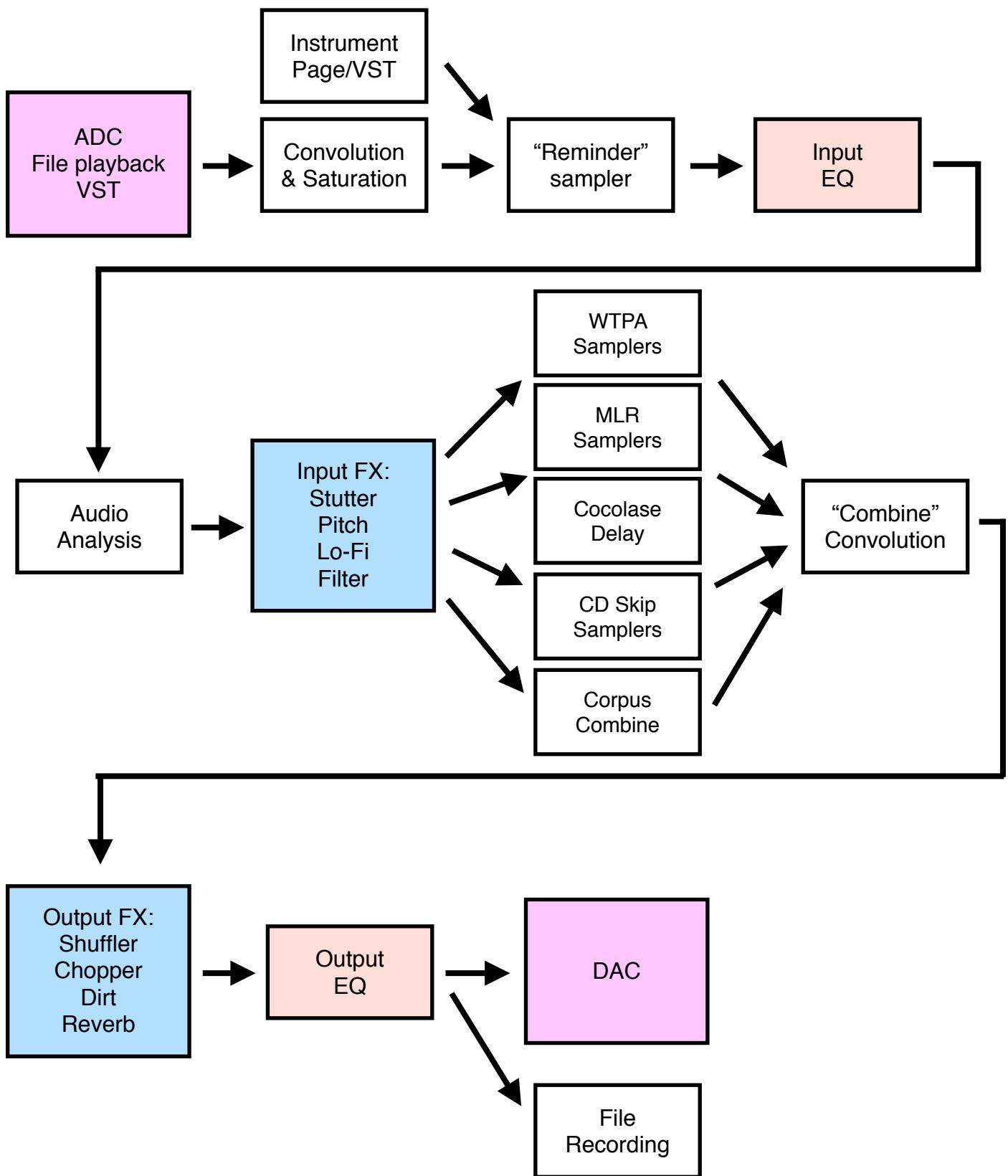
NEW IN THIS VERSION (v1.1)

- bug fixes
- under the hood improvements to codebase
- latest version of serialosc.maxpat to allow patch switching
- released app version as well as max patch version

TABLE OF CONTENTS

Signal Flowchart	5
Input/Output/Info/Record Section	7
Setup Window	8
Input FX	9
Output FX	13
WTPA Modules	17
MLR Modules	25
Monolase	26
The Chocolate Grinder	28
Preset Morphing	30
MIDI Mapping	31
Softstep Mapping	32
Monome Mapping	33
Page Descriptions	34
Arc Mapping	50
Troubleshooting	51

SIGNAL FLOWCHART



MODULE & GUI BREAKDOWN

The GUI was built to be very minimal and utilitarian. It is not intended to be used while performing, as you don't need to interface with things on screen while you play.



INPUT/OUTPUT/INFO/RECORD SECTION



This section has three main parts. The bottom left is your input section. The number boxes, and corresponding meters are the first four inputs of your soundcard. If you only have two inputs, only the first two will appear active (as in the screenshot).

The drop-down menu (currently showing “adc”) is where you select whether you want to use live input, or file playback (or test tone/vst). Open/play are for the file playback, although when in vst mode, open opens the current vst instead.

Clicking on the word ‘input’ opens a window that lets you add convolution (equipment emulation) to the different inputs as well as an overall input/output EQ. There is also a chromatic tuner in this section

On the bottom right we have the output section. Here you set the thru level, the level of all sampler modules (including wtpa, mlr, cocolase, cd, etc...), and the instrument level.

It should be noted that the instrument output goes down the same signal path as everything else, so ‘thru’ needs to be turned up as well.

On the top left you find the master output level/meter and the revamped “Setup” section (explained in detail in the next chapter).

The record button lets you record the summed output of The Party Van to a file. When you press it you are presented with a file dialog window. Name your file and hit ‘OK’ and you will see the meter going, indicating that you are recording audio.

The ‘info’ button gives a general overview of the patch including the required externals and OSC prefix.

SETUP WINDOW



Here you have pop-up window for all the DSP, IO, and monome settings.

The top section of the window mirrors the settings found in the Max/MSP Audio Setup window. You can find more information about the settings in the Max/MSP documentation. Note: It is important to have “Scheduler in Overdrive” turned ON for The Party Van to function properly.

The OSC section lets you send your IP address and receive/send ports. The MIDI learn section lets you map parameters by hovering over them. The Grid/Arc section deals with monome-related devices/connections.

The Shnth section enables mapping of a Shbobo Shnth (www.shbobo.net) to control parameters of TPV.

INPUT FX



Here are the four input effects. Audio routing goes top down with each effect being in series with the previous one.

STUTTER



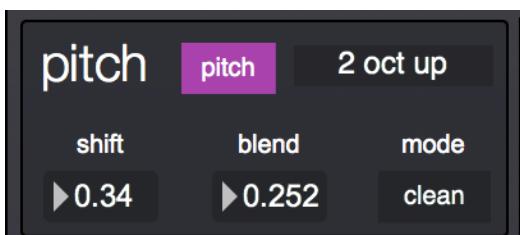
“Stutter” is a stutter/glitch module with two modes. The ‘dd-6’ mode, which does time domain stuttering and is modeled after the [Boss DD-6](#) guitar pedal in ‘hold’ mode. The other one is an FFT mode that does spectral freezing. Both modes are always running so you can switch between them even after creating a stutter/glitch. To change modes press the button in the bottom left.

(Stutter cont..)

To activate the Stutter module you must first record a loop. The ‘record’ button records audio for the duration that it is depressed, so it is best to map it to a momentary button. Then the ‘stutter’ button turns on the audio. On the Monome everything is mapped to one button already by default so pressing and releasing the button records and plays back a glitch.

“Kill” mode mutes incoming audio when there is a glitch playing. Finally ‘random’ turns on/off the record function at a rate and probability set by the rate and density controls.

PITCH



The “Pitch” module uses a delay-based pitch shifting technique to produce somewhat natural, somewhat glitchy sounding pitch shifting. I am aiming for a ‘whammy pedal’ type glitchiness, but it is difficult to emulate.

The preset window (showing “2 down/2 up”) lets you select ranges for the ‘shift’ parameter. These are also inspired by the Whammy pedal. ‘Shift’ controls the pitch shifting in relation to the range of the selected pitch preset. ‘Blend’ controls the dry/wet levels. The ‘dirty/clean’ button affects the pitch tracking. When set to ‘clean’ the window size changes only on new attacks. When set to ‘dirty’ the window changes all the time, so you get lots more artifacts.

LOFI



The “Lofi” module is a combination of signal degradation modules. ‘Sample’ controls traditional sample rate reduction, while ‘bit’ & ‘bit 2’ control bit rate reduction using different techniques that allow for fractional bit rate reduction. Both techniques sound slightly different, but allow for a wide range of sounds when combined. ‘Bitwise’ does bit flip/replacement processing and sounds very grungy. Even though this is controlled by a single float value, it is eight discrete steps.

The new ‘mp3ify’ control does some FFT processing on the signal to allow you to dynamically control the psycho-acoustic compression very similarly to how mp3 compression works. When combined with the other signal degradation parameters, this adds a very distinctive and unique twist to the lofi sound world. Note that this parameter behaves ‘inverted’ compared to the other ones in this module. For other parameters lowering the value ‘reduces’ the fidelity; for mp3ify increasing the value turns up the mp3 compression.

FILTER



The “Filter” module gives you a couple of different filter modes (lowpass, highpass, and resonant) that are modeled after a moog-style ladder filter. Frequency and Q work as expected on a traditional filter. The display is only a rough estimation of the filter shape.

The ‘dirtiness’ parameter adds some noise to the filter coefficients resulting in a dirty, bubbly sound. Use this sparingly as it is possible to ‘blow up’ the filter if pushed too hard, and my patch has no means of resetting the filter other than reopening the patch. The inspiration for this addition was Pierre Alexandre Tremblay’s butt~ filter.

OUTPUT FX



Here are the four output effects. Audio routing goes top down with each effect being in series with the previous.

SHUFFLER



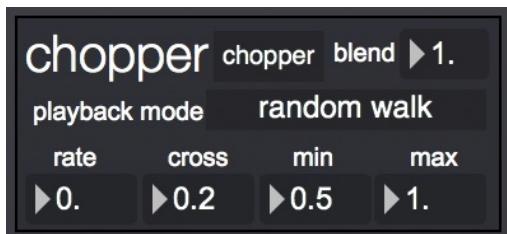
“Shuffler” is a dual mode buffer shuffler. In ‘shuffle’ mode it randomly plays back modulated audio from two prerecorded buffers with the ‘pitch’ controlling how severe the playback speed modulation is.

In ‘spindown’ mode, it takes the last audio fragment being played and repeats it while lowering playback speed, sounding similar to a record being stopped.

(Shuffler cont..)

“Kill” mode mutes incoming audio when there is a glitch playing. Finally, ‘random’ turns on/off the record function at a rate and probability set by the rate and density controls.

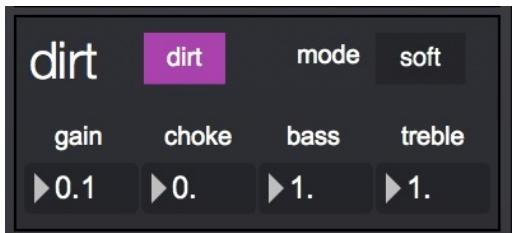
CHOPPER



The “Chopper” module is based on the example patch of the same name included in the Max6 examples folder. It segments audio based on zero crossings and has options to choose how these are played back using the different playback modes. These define how the tiny fragments are played back with each having a characteristic sound to it. The ‘rate’ controls the rate of audio playback. ‘cross’ controls how many positive zero-crossings are stored per segment and ‘min’/‘max’ are the minimum and maximum length of each segment.

Blend controls the wet/dry level of the effect.

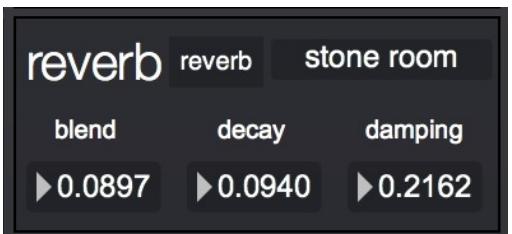
DIRT



The “dirt” module is a distortion module with two modes. The ‘soft’ mode is based on ‘faust’ code by Julius O. Smith and uses cubic nonlinear distortion. It has a milder ‘overdrive’ type sound. The ‘hard’ mode is a variable-hardness clipping function by Laurent de Soras. It is, as the name implies, a harsher distortion.

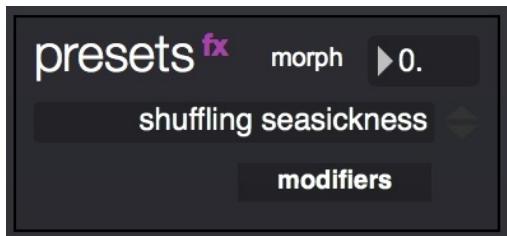
The ‘gain’ control is the overall level of distortion applied to the signal. ‘choke’ effects the harmonic content and gating of the distortion and varies on the mode (soft/hard) being used. Bass and treble are both cut filters so when set at maximum, the signal remains unaffected.

REVERB



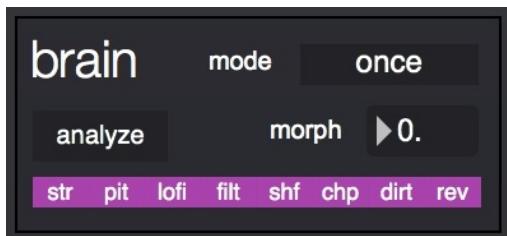
The ‘reverb’ module is a stereo convolution reverb with 24 built in impulse responses ranging from rooms to halls to noise. ‘blend’ controls the wet/dry level, ‘decay’ processes the impulse response to make the reverb shorter (there’s a small pause in the reverb audio while this is calculated) and ‘damping’ removes some of the high frequencies simulating a more muted acoustic.

PRESETS



The ‘presets’ section controls the entire input/output effects chain. You can select some preconfigured presets here as well as save & load your own. Under ‘modifiers’ you can assign a variety of LFOs as well as audio analysis parameters to effects chain settings. All the modifier settings are stored with the preset. The ‘morph’ control morphs between the current and previous preset.

BRAIN

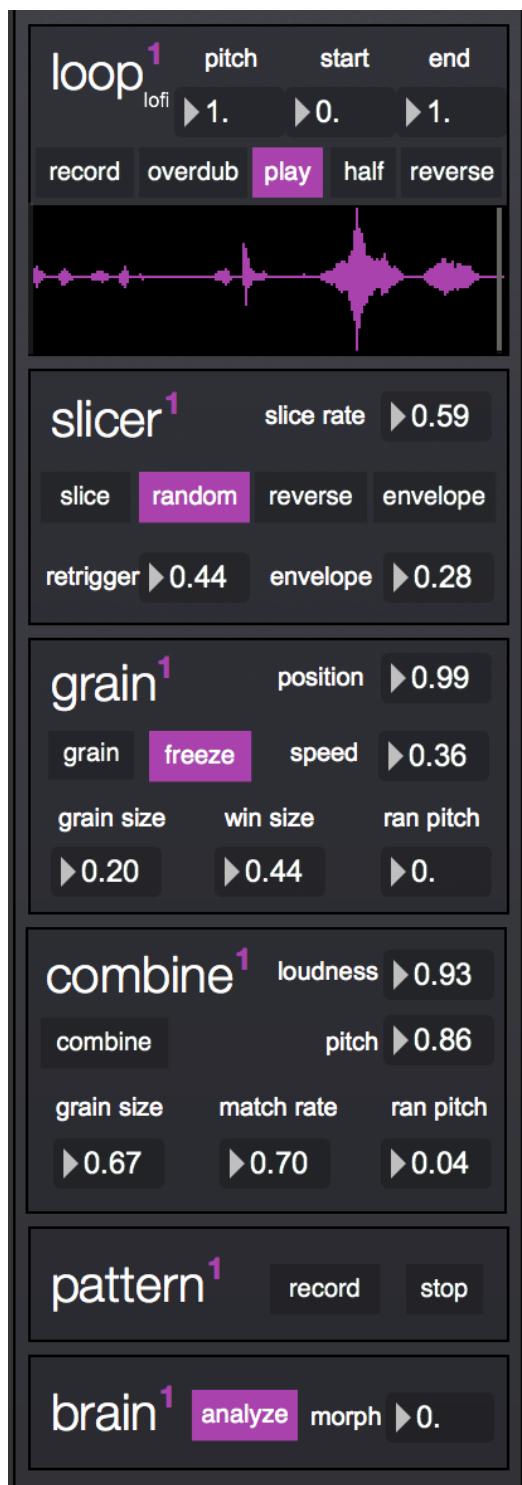


The ‘brain’ module controls both the input and output effects. It works by analyzing incoming audio and creating dynamic presets for each of the effect parameters.

There are multiple modes for the ‘brain’ module. ‘once’ means the preset will be generated only once, until the patch is restarted. ‘every’ changes the preset every time the effect is turned on. The ‘live’ modes control the parameters in realtime by controlling the individual parameters, the effect on/off state, or both.

The ‘morph’ control morphs between the current and previous preset.

THE WTPA MODULES



There are two identical ‘WTPA’ modules which are looper/samplers with different buffer playback/processing options.

LOOP



One of the most time consuming modules to build was the “Loop” module. Partially because I was still coming to grips with using Max/MSP with live sampling, but mainly due to the functionality of it, and its integration with other ‘WTPA’ modules.

The “loop” module is a dynamically sized (as in you start end the loop whenever you want) varispeed looper at its core. It is modeled after the “[Where's The Party At](#)” 8-bit sampler/looper, which is somewhat modeled on the [Line 6 DL-4](#) looper.

Because of how it is coded, recording at speeds below 1 adds a sample rate reduction sound because it is actually recording at a lower sample rate. The pitch control works almost as a ‘clock’ control in that manner. Any changes in pitch made while recording are heard when playing back. While recording you can also engage half-speed and reverse, including reversing ‘past zero’ which just makes the loop jump to the highest point you’ve written in the buffer up to that point.

The ‘start’/‘end’ controls control the start/end points of the loop. Using a monome grid, these can be controlled dynamically in an MLR-type fashion.

Lastly, there is a ‘lofi’ mode that can be engaged by pressing the word ‘lofi’. This puts the looper into a more 8-bit sounding mode with added grunge/saturation all tied into the pitch control knob.

SLICER

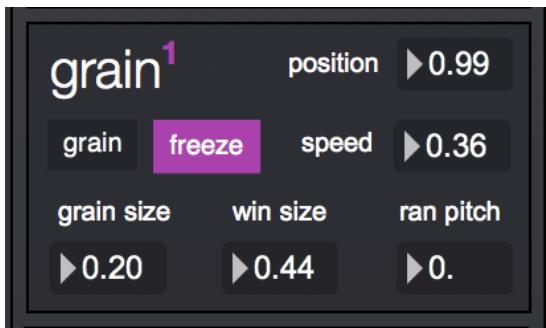


Like all the ‘WTPA’ modules, the “Slicer” module is linked in with other modules. You can start creating a loop with the ‘slicer’ engaged and it will jump around the buffer writing tiny slices into it, creating a pointillistic/glitchy loop.

The basic functionality of the “Slicer” is to jump around the buffer (either written, or while recording) at regular or random intervals (controlled by the ‘random’ button). ‘Rate’ controls the overall speed of this process. The ‘reverse’ toggle allows for slices to be played back in reverse as well as forward (this is independent of the global ‘reverse’ controlled in the “Loop” module). The ‘envelope’ control adds a linear fade out to each slice playback which adds an almost percussive sound to each slice. The ‘envelope’ parameter controls how long the duration of that envelope is. Finally, ‘retrigger’ controls how likely the slicer is going to play back the same slice again. This can add a ‘retriggery’ kind of sound to the slicer module.

There is a small amount of amplitude enveloping when jumping to new slice points so the “Slicer” has some clicks/pops when jumping. This is intentional, but can be a bit overwhelming with very bassy sounds.

GRAIN



Like its name would imply, “Grain” is a granular synthesis module. It is based on [sugarSynth](#) by Nobuyasa Sakonda.

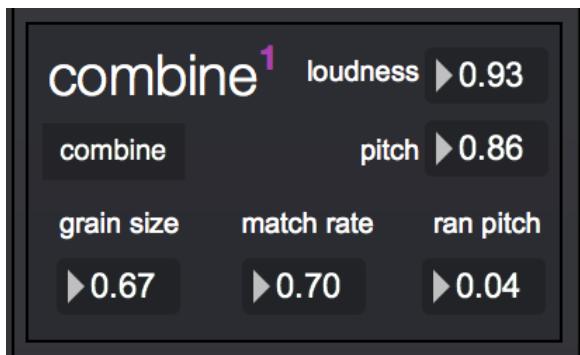
The ‘position’ parameter controls the position of the grain playback engine in the sample buffer. This is tied to the playhead of the “Loop” module in such a way that you can turn on the “Grain” module at any point and it will start granulating the sound present at playhead position.

‘Freeze’ & ‘speed’ control how “Grain” moves through a buffer. If ‘freeze’ mode is engaged “Grain” will just stay put and continue to spit out grains. If ‘freeze’ is turned off, the granular synth will move through the buffer as controlled by the ‘speed’ control. A ‘speed’ setting of 1 would be 100% playback speed.

‘Grain size’, ‘window size’, and ‘random pitch’ control the granular synth engine and do exactly what their name implies. ‘Grain size’ controls the size of each grain of audio. ‘Window size’ controls the size of the buffer window that grains are played back from and ‘random pitch’ randomizes the playback pitch of each grain.

“Grain” and “Slice” can be used in combination to jump around the buffer randomly while doing granular synthesis on that portion of the buffer.

COMBINE



The “Combine” module is similar to the “Grain” module in that it is based on granular synthesis, but it is driven by an audio-analysis engine to handle grain replacement (concatenative synthesis). Effectively, as you are recording a loop in the “Loop” module, the “Combine” module is analyzing each grain of audio (20ms slices) and storing the position, loudness, and pitch of each grain in a database. When the “Combine” module is engaged it then analyzes incoming real-time audio and matches/replaces each grain producing a very ‘granular synthesis’ sound, but controlled very differently.

There are two modes to the “Combine” module and they are selected contextually. If you engage the “Combine” module while the loop module is NOT playing, then you have concatenative synthesis as described. If you engage it while playing back audio, then it does FFT convolution on your incoming audio and buffer playback. There are no controls over the convolution, and none of the displayed controls effect convolution. The convolution is effected by all other playback modes in the sense that the audio is being convolved with whatever the playback audio is (be it sliced, low pitch, granulized etc...).

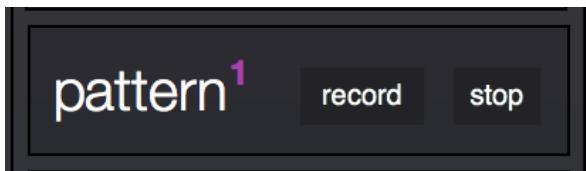
(Combine cont..)

The parameters ‘loudness’ and ‘pitch’ control the weight that is given to each in terms of playback analysis. If you have a high ‘pitch’ setting, it will only match grains that match in pitch very closely, same goes for loudness. If you record an audio loop into the “Loop” module and play back that audio loop with loudness/pitch set to high settings you will hear, more or less, exactly the same audio. It will be a little glitchy sounding, but it will match and replace each grain with its real-time counterpart.

‘Match rate’ controls how often a grain is selected and played back. This only effects the start point of the grain, that is to say it won’t just play that ‘grain’ of audio. It jumps to the start point and will play as far as is set by the ‘grain size’ parameter.

Finally ‘random pitch’ will, like in the “Grain” module, randomize the playback pitch of each grain.

PATTERN

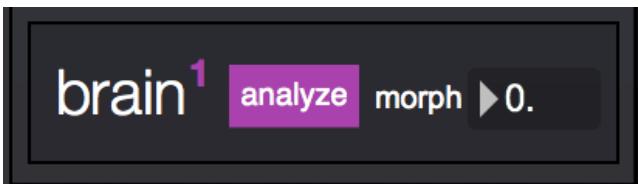


The “Pattern” module lets you record, and loop, any parameter changes in the ‘WTPA’ section. This effects button presses, parameter controls, and Monome MLR-style presses. It is coded in a way that allows you to make a pattern out of absolutely anything in this section, including creating loops.

The “Pattern” mode will allow you to start/stop a pattern for as long as you want but once you press ‘record’ it creates a new one. Note that patterns are not saved with presets, so when you load the patch there is no pattern loaded.

The Monome mapping for this control creates a new pattern when engaged. It is not possible to stop, and restart the same pattern from the Monome. This is intentional, although sometimes not ideal.

BRAIN



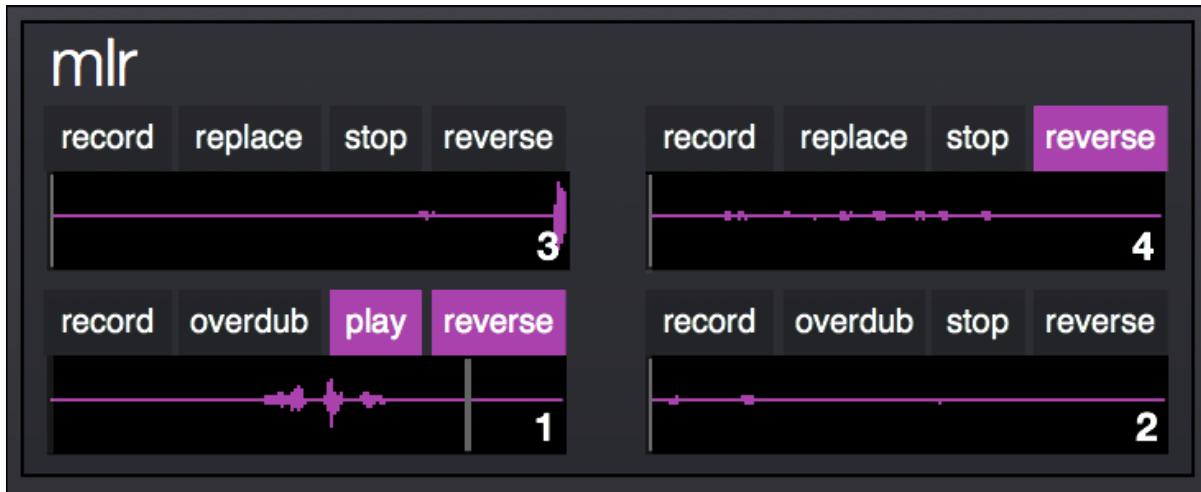
The “Brain” module is the ‘backseat driver’ of the whole ‘WTPA’ section of the patch. It uses complex* buffer analysis to change all the parameters of the “Slice”, “Grain”, & “Combine” modules. When the ‘analyze’ button is engaged, every time you create a new loop, the “Brain” module will change the above mentioned parameters to something musically/sonically relevant. You can still manually control things, but this is meant as an intelligent preset generator so that when I turn on “Grain” it doesn’t always sound the same, but it isn’t random or unrelated to the contents of the buffer.

I spent a tremendous amount of time coding, tweaking, testing, and updating the analysis algorithms to something I find musically useful.

The ‘morph’ control lets you morph between the current analysis preset and the previous one.

*Fifteen different analysis parameters are used and weighed against each other different for each control parameter. Each parameter is also weighted against what it was set to previously.

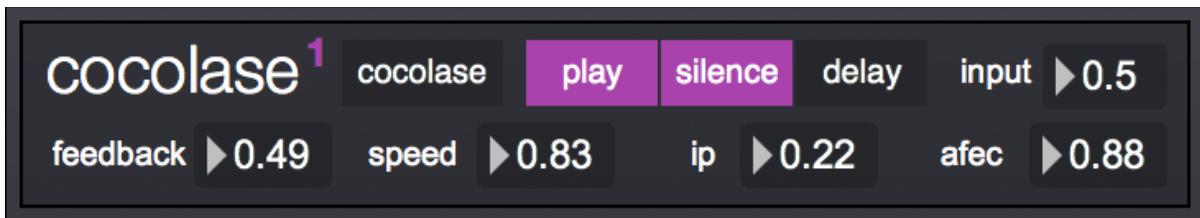
MLR MODULES



The “MLR” modules are built using the same core sampler as the “Loop” modules in the ‘WTPA’ section of the patch. They are mapped onto a separate page on the Monome GUI and are simply here to allow for more complex/layered MLR-style playing*.

*MLR is a very popular and powerful core Monome patch that lets you trigger and loop sections of the buffer using the Monome buttons, with LED indication of playhead.

COCOLASE



Cocolase essentially mimics the “Monolase” modules from my program of the same name, which are in turn modeled on the [ciat-lonbarde Cocolase](#), an 8-bit sampler and modulated delay. Most of the routing/modulation control of this module is only accessible via a Monome, although what is in the GUI makes for a great sounding lofi delay/sampler already. The controls mirror that of the Cocolase in terms of control and names, so they can be a bit confusing for the uninitiated.

The Cocolase modules are always recording into a very small circular buffer. When you engage the ‘cocolase’ buttons, we can hear the output of the delay. ‘Play’ starts/stops the playhead. The ‘sound’ button mutes/erases the buffer as you go over that section. Useful for creating silences in the loop. The last button ‘delay/sample’ selects between the two modes. ‘Delay’ is a traditional circular delay whereas ‘sample’ freezes the buffer in its current state with no new audio being written.

The GUI was mostly removed for v1.0 but the functionality is identical as before.

‘Input’ controls the input level to that particular module. ‘Feedback’ works as it would on a traditional delay module. ‘Speed’ also works as in a traditional delay module, but since the buffer is of a fixed length (1 second) ‘speed’ also controls loop length. The longer your loop the lower the sample rate.

(Monolase cont..)

There are several LFOs working behind the scenes that are randomized and continuously repatched to create complex control signals for the three playback parameters.

The parameters that can be controlled are:

Afec = Pitch/speed modulation

Skip = Jumping position in delay

Flip = Change direction (forward/reverse)

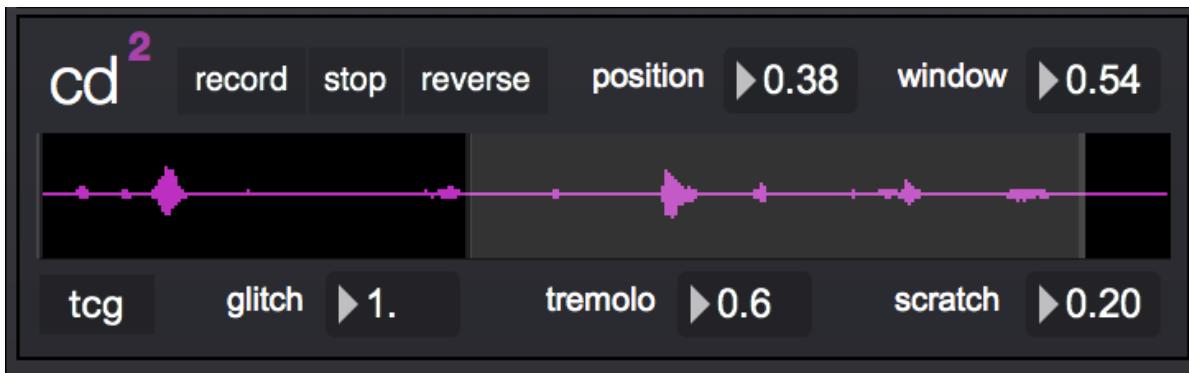
The 'ip' works like a threshold for each trigger.

The Cocolase is a complex instrument and system in itself.

For more information you can read the manual for the original hardware unit here:

<http://www.ciat-lonbarde.net/cocolase/man/index.html>

THE CHOCOLATE GRINDER



“The Chocolate Grinder” comes from another patch of the same name that I wrote. It is a virtual skipping CD player, with authentic glitch samples and complex/weighted randomization for glitch playback. In addition to the ‘glitch’ sounds of a skipping CD player, this module also emulates the fast forward/rewind sound of digital CD players.

The ‘level’, ‘record’ and ‘stop/play’ controls are all self explanatory. One thing worth mentioning is that there is no overdubbing or pitch/speed control. To make it true to a CD player, those features are not available. The ‘glitch’ control turns on the actual glitch sounds, which are made up of actual samples as well as some synthesized glitches. The ‘level’ control next to ‘glitch’ controls the level of the glitches.

As of version 1.0, there are three modes of “The Chocolate Grinder”. ‘tcg’ is classic chocolate grinder functionality. ‘tml’ is an adaptation of Tehn’s ‘tml’ app, and ‘tbl’ is a turntable-like mode. Each mode has a corresponding control that only works when in that mode (glitch, tremolo, scratch).

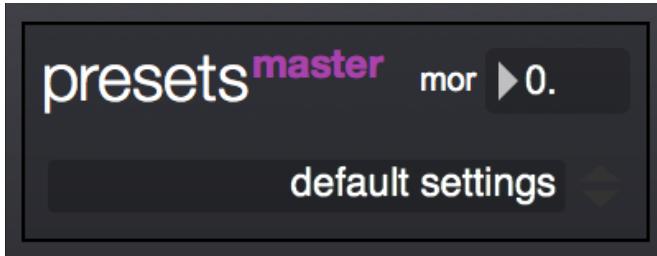
(The Chocolate Grinder cont..)

Along the bottom row we have ‘position’, which controls the position of the START of the playback window. Then ‘window’ controls the SIZE of that window. Those two parameters determine when a glitch/skip happens. When the playhead reaches the end of the window, it jumps to the start of the window. This happens with sample accuracy, so that you can ‘play’ this module by moving the window around the playhead. It will only glitch/skip when you let it. The ‘jitter’ control adds some noise/randomization to the ‘position’ and ‘window’ controls, giving it a more realistic glitch rather than a predictable rhythmic one.

An important control, that is not exposed on the user interface, is the ‘lockout’ control. This sets the sound/speed of the fast forward/rewind. Essentially, the ‘lockout’ only lets a glitch/skip happen every 100ms or so (the ‘lockout’ is randomized on each patch load, to simulate different brand/era CD players).

In addition to the ‘lockout’ randomization, the glitch randomization weighting and levels are also randomized on patch loading. I did this so as to allow for different ‘feeling’ CD players each time you load it. The changes aren’t radical, but it won’t always sound the same (like a real skipping CD player).

PRESET MORPHING/HANDLING



There is complex preset handling including a hierarchical preset control and behind the scenes preset generation. The hierarchical presets are traditional store/recall presets and the behind the scene presets are generated automatically based on both buffer-based samples as well as incoming audio analysis.

There are presets for the WTPA sections, for the input/output fx, and then global presets which store/recall the wtpa/fx presets.

There are three ‘behind the scenes’ preset sections:

WTPA1 (CC97): This generates an intelligent preset for all WTPA1 modules (slice1, grain1, combine1) based on buffer analysis of loop1.

WTPA2 (CC98): Same as above but for WTPA2 modules.

EFFECT (CC99): This generates an intelligent preset for all input/output effects based on incoming audio analysis.

Whenever a new preset is made, it is saved to a preset slot. There are two more presets which are not editable. One has relatively low values and the other has relatively high values. When adjusting the appropriate interpolation parameter (WTPA1, WTPA2, effect) each value is individually interpolated between where it currently is, and either a higher or lower value. Additionally, hand-drawn lookup tables were created to add non-linearities to the interpolation.

This setup creates a very powerful, macro-like “more” and “less” type control, allowing you to control every parameter but without having to micromanage.

MIDI CC MAPPING

In addition to the MIDI-learn functionality of the patch, there are some parameters which are assigned to CC numbers by default.

CC mapping

- 050-057 - wtpa1 slice triggers
- 090 - corners gravity
- 091 - corners friction
- 092 - straw param 1
- 093 - straw param 2
- 097 - wtpa1 preset
- 098 - wtpa2 preset
- 099 - effect preset
- 100 - mlr1 pitch
- 101 - mlr2 pitch
- 102 - mlr3 pitch
- 103 - mlr4 pitch
- 104 - mlr1 volume
- 105 - mlr2 volume
- 106 - mlr3 volume
- 107 - mlr4 volume
- 108 - wtpa1 volume
- 109 - wtpa2 volume
- 110 - wtpa preset up
- 111 - wtpa preset down
- 112 - fx preset up
- 113 - fx preset down
- 114 - global fx up
- 115 - global fx down
- 116 - wtpa1 pattern
- 117 - wtpa2 pattern
- 118 - cocolase randomize matrix

SOFTSTEP MAPPING

The [Keith McMillen Softstep](#) has mappings by default using sysex commands. The communication happens directly with the Softstep using objects created by [Jeff Kaiser](#). There are three ‘pages’ to the softstep mapping. These can be accessed with the Nav button and the text display updates to show which page you are on.

In order to use the Softstep mapping you must first press the “g” key on your computer keyboard.

WTPA

- 1 - record/overdub
- 2 - play/stop
- 3 - halfspeed/reverse (tap/doubletap)
- 4 - slice
- 5 - concat
- 6 - stutter
- 7 - pitch
- 8 - lofi
- 9 - chopper
- 0 - dirt

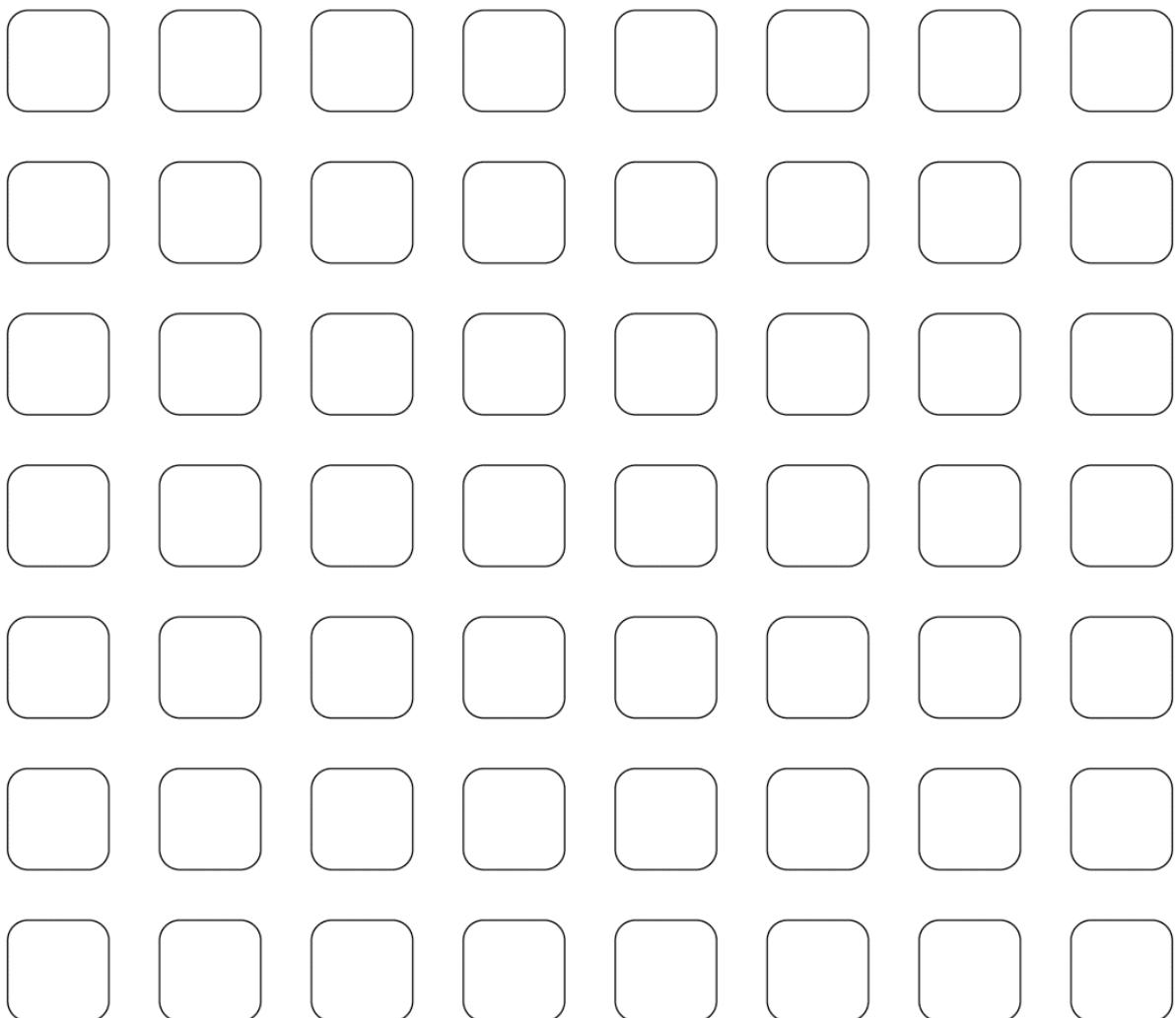
MLR1 (and MLR2)

- 1 - record/overdub
- 6 - play/stop
- 2 - mlr step 1
- 3 - mlr step 2
- 4 - mlr step 3
- 5 - mlr step 4
- 7 - mlr step 5
- 8 - mlr step 6
- 9 - mlr step 7
- 0 - mlr step 8

MONOME MAPPING

THE TOP ROW REPRESENTS DIFFERENT “PAGES”. EACH ONE BEING IT’S OWN APPLICATION/SETUP

1 2 3 4 5 6 7 8



PAGE DESCRIPTIONS

PAGE1 - WTPA MODULE/GREATEST HITS

Two sampler/loopers with mlr-style display/control, granular synthesis, grain replacement, sample slicing, pattern recording, input/output fx, audio input based triggering of effects, and circular buffer “reminder”.

PAGE2 - MLR ADAPTATION

Four sampler/loopers with mlr-style display/control along with four pattern recorders.

PAGE3 - MONOLASE

An 8-bit modulated delay/sampler based on the Cocolase by ciat-lonbarde.

PAGE4 - THE CHOCOLATE GRINDER

A virtual skipping CD, tml emulation, and turntable emulation.

PAGE5 - C-C-COMBINE

An audio-mosaicing module based on concatenative synthesis.

PAGE6 - PRESET MORPHING

Select between global presets. (*Will eventually contain a preset sequencer system.*)

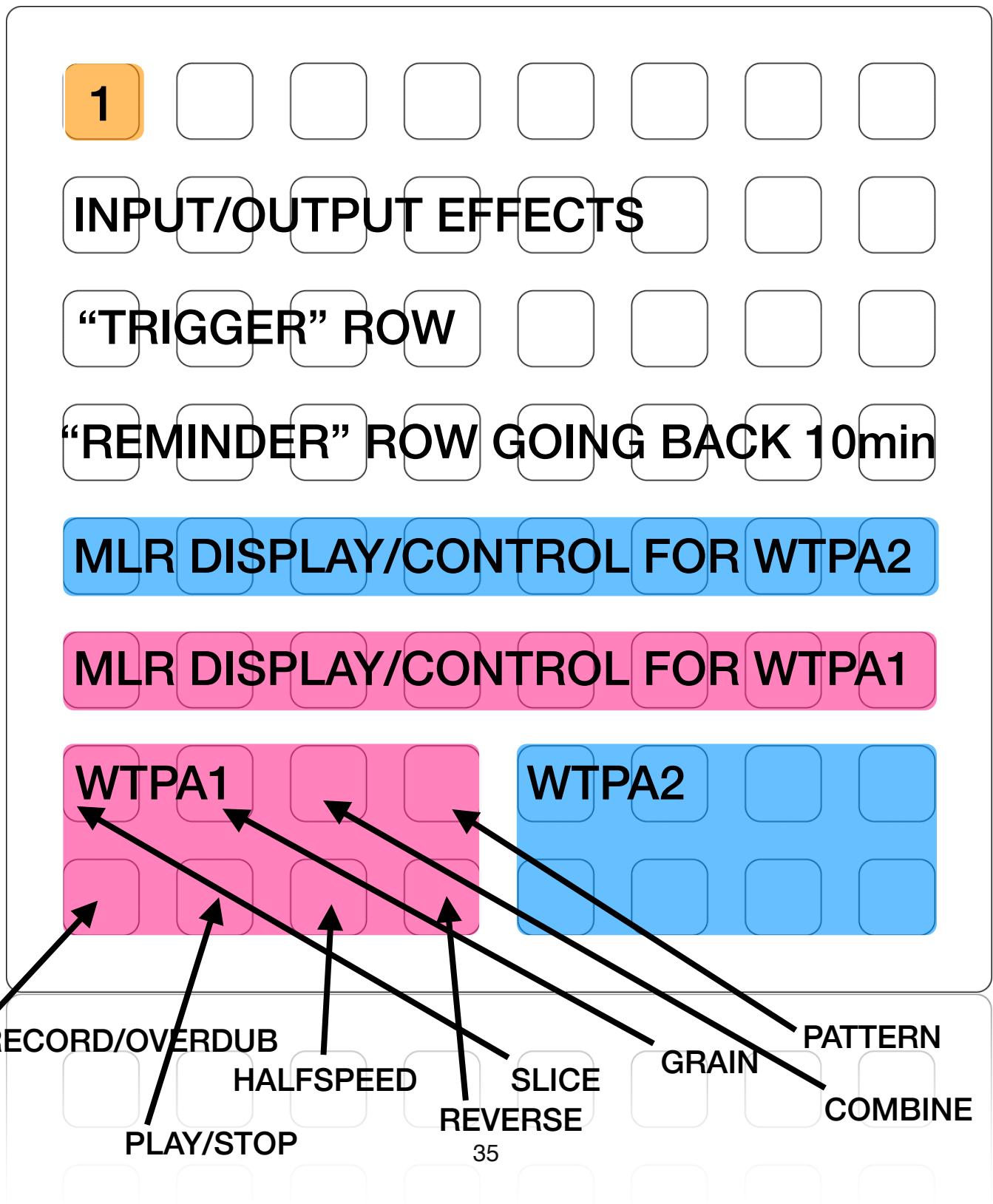
PAGE7 - INSTRUMENT

A collection of different instruments on subpages including piano/rhodes and adaptations of the Fourths, Straw, & Corners monome apps.

PAGE8 - LEVELS

“Fader” style control over four input levels, overall sample levels, overall instrument levels, thru level, and master level.

PAGE1 IS A “GREATEST HITS” TYPE PAGE SO IT CONTAINS A LOT OF DIFFERENT COMPONENTS



ALL MLR ROWS (PAGE1/PAGE2) ARE USING MY NEW “SMLR” (SUPER-MLR) MAPPING

1 2

The ‘smlr’ mapping applies to WTPA1/WTPA2 on Page1 as well as MLR1/MLR2/MLR3/MLR4 on Page2.

Pressing single buttons jumps to that position in the buffer.
Pressing two buttons sets the min/max points for a loop.
Pressing three buttons changes the loop direction.
Pressing four buttons starts recording a new loop.

You can very dynamically/expressively “play” the ‘mlr’ row in a piano-like fashion.

You can also go right into any of those button press combinations from a loop being recorded. So if you begin recording a loop, you can then press two mlr buttons to beginning looping that section immediately (rather than hitting play, letting the loop start from the beginning, and then setting the loop points).

BREAKDOWN OF TOP THREE ROWS

1

INPUT/OUTPUT EFFECTS

“TRIGGER” ROW

“REMINDER” ROW GOING BACK 10min

EFFECTS:

- 1 - STUTTER (PRE)
- 2 - PITCH SHIFT (PRE)
- 3 - LOFI (PRE)
- 4 - FILTER (PRE)
- 5 - SHUFFLER (POST)
- 6 - CHOPPER (POST)
- 7 - DIRT (POST)
- 8 - REVERB (POST)

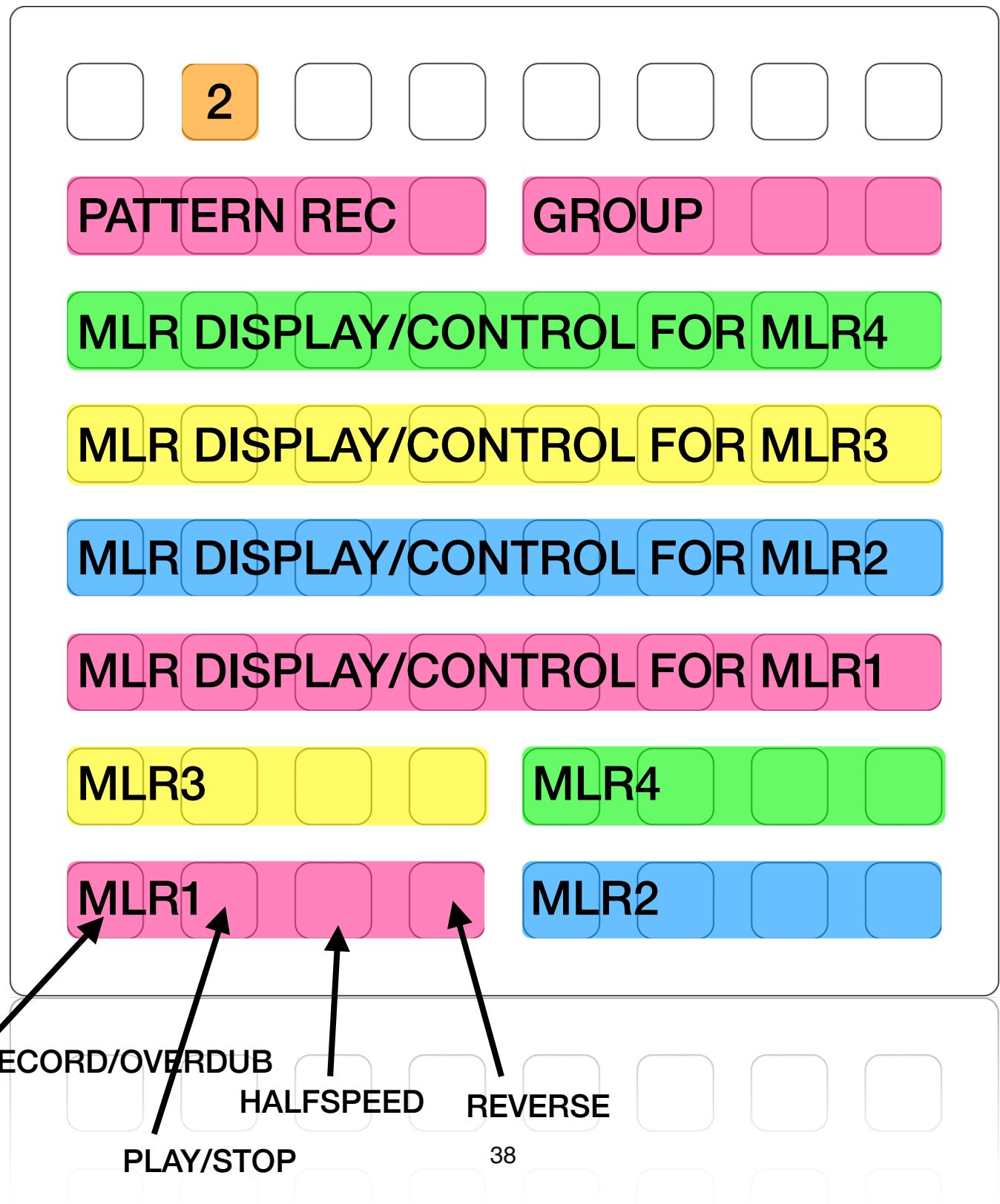
TRIGGERS:

- 1 - RECORD/OVERDUB
- 2 - STOP/PLAY
- 3 - HALFSPEED
- 4 - REVERSE
- 5 - SLICER
- 6 - GRAIN
- 7 - COMBINE
- 8 - POSITION JUMP

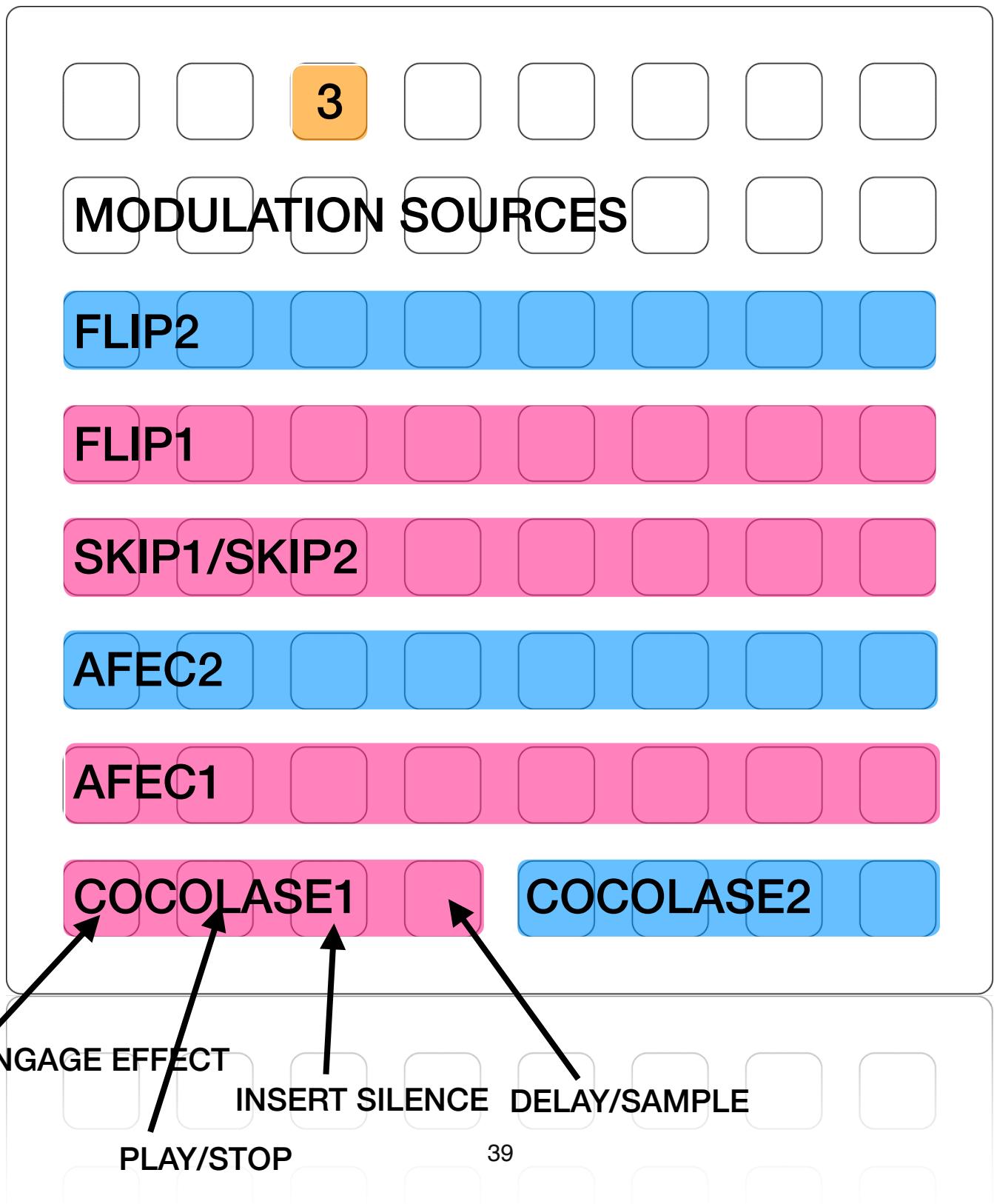
Trigger row works by analyzing incoming audio for attack onsets. When an attack is detected it triggers the designated function. There are two modes. The first mode toggles the function on/off alternately with each attack. The second mode turns the function on, then off automatically based on audio analysis.

The Reminder row goes back in time with the leftmost button jumping 5 seconds into the past, and the rightmost button jumping 10 minutes into the past. When pressed, each button plays back from the loudest moment within each time frame. The 10 minute button does concatenative synthesis on the last 10 minutes of audio (using the same analysis/playback engine as the “Combine” module).

PAGE2 IS AN MLR-ADAPTATION WITH FOUR MLR LOOPERS AND PATTERN RECORDERS



PAGE3 IS AN 8-BIT MODULATED DELAY/SAMPLER BASED ON THE COCOLASE



DETAILED EXPLANATION OF COCOLASE PAGE

3

The modulation sources can be altered by pressing any button on that row. Each source has a rate (random at startup) and can be either square or triangle, and LFO or audio rate. Pressing the buttons toggles between square/triangle and lfo/audio rate.

The parameters that can be controlled are:

Afec = Pitch/speed modulation

Skip = Jumping position in delay

Flip = Change direction (forward/reverse)

Any one of the parameters can be controlled by multiple sources, and combining modulation sources will create complex waveforms.

The bottom row of buttons work as follows.

Cocolase = Engages playback of effect. The delay is constantly running/capturing audio

Play = Stops/start the audio right where it left off

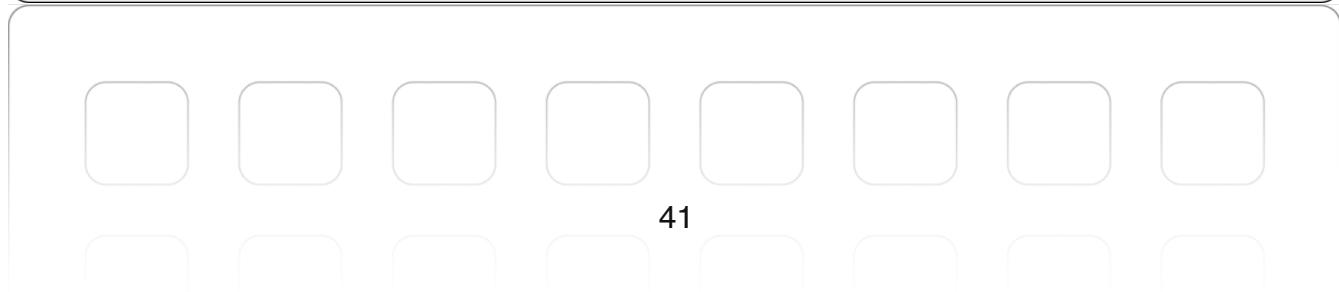
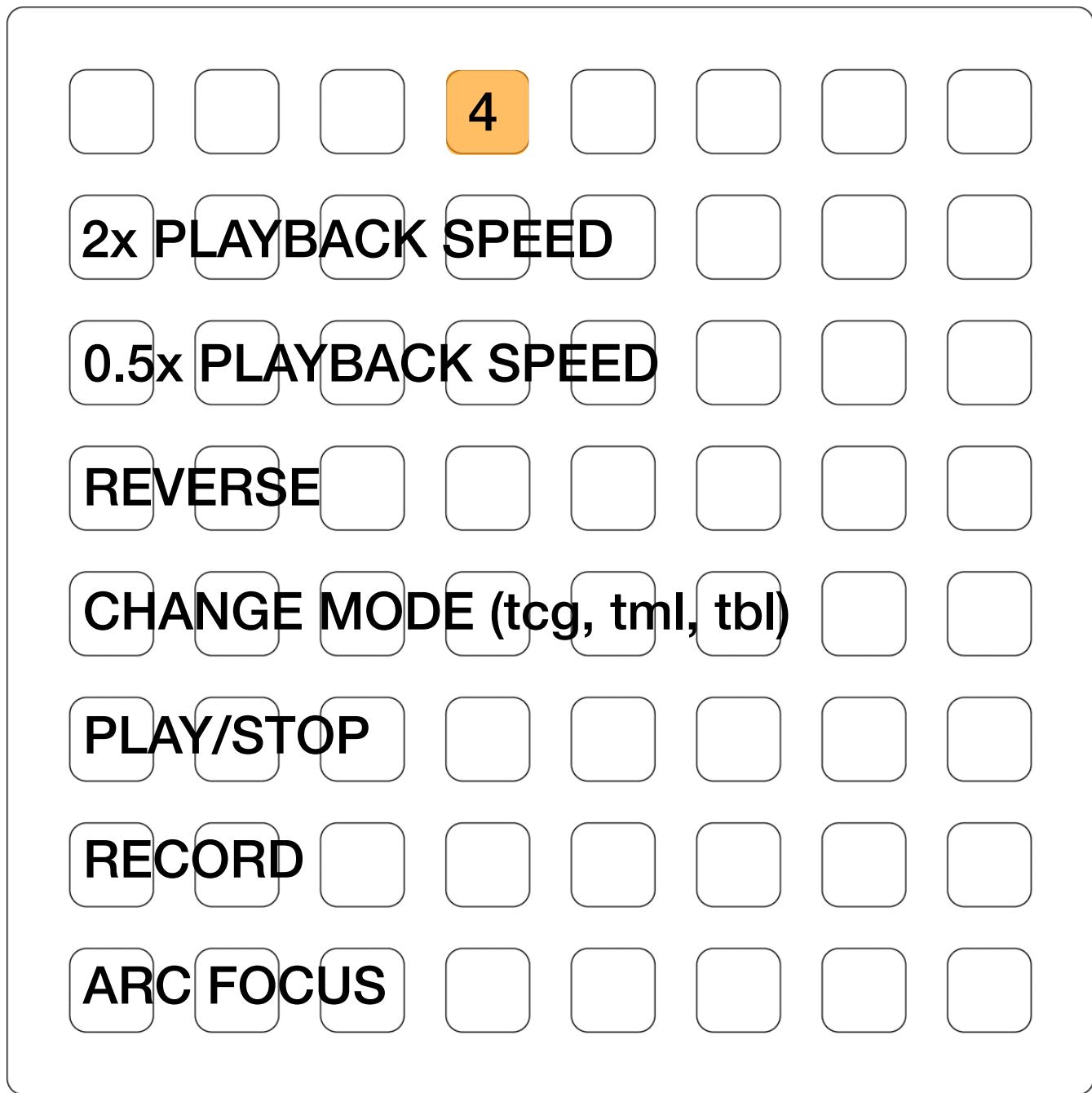
Silence = Inserts silence into the loop.

Delay/Sample = Freezes the buffer

More information can be found here:

<http://www.ciat-lonbarde.net/cocolase/man/index.html>

PAGE4 IS THE CHOCOLATE GRINDER, A VIRTUAL SKIPPING CD PLAYER.



DETAILED EXPLANATION OF THE CHOCOLATE GRINDER PAGE

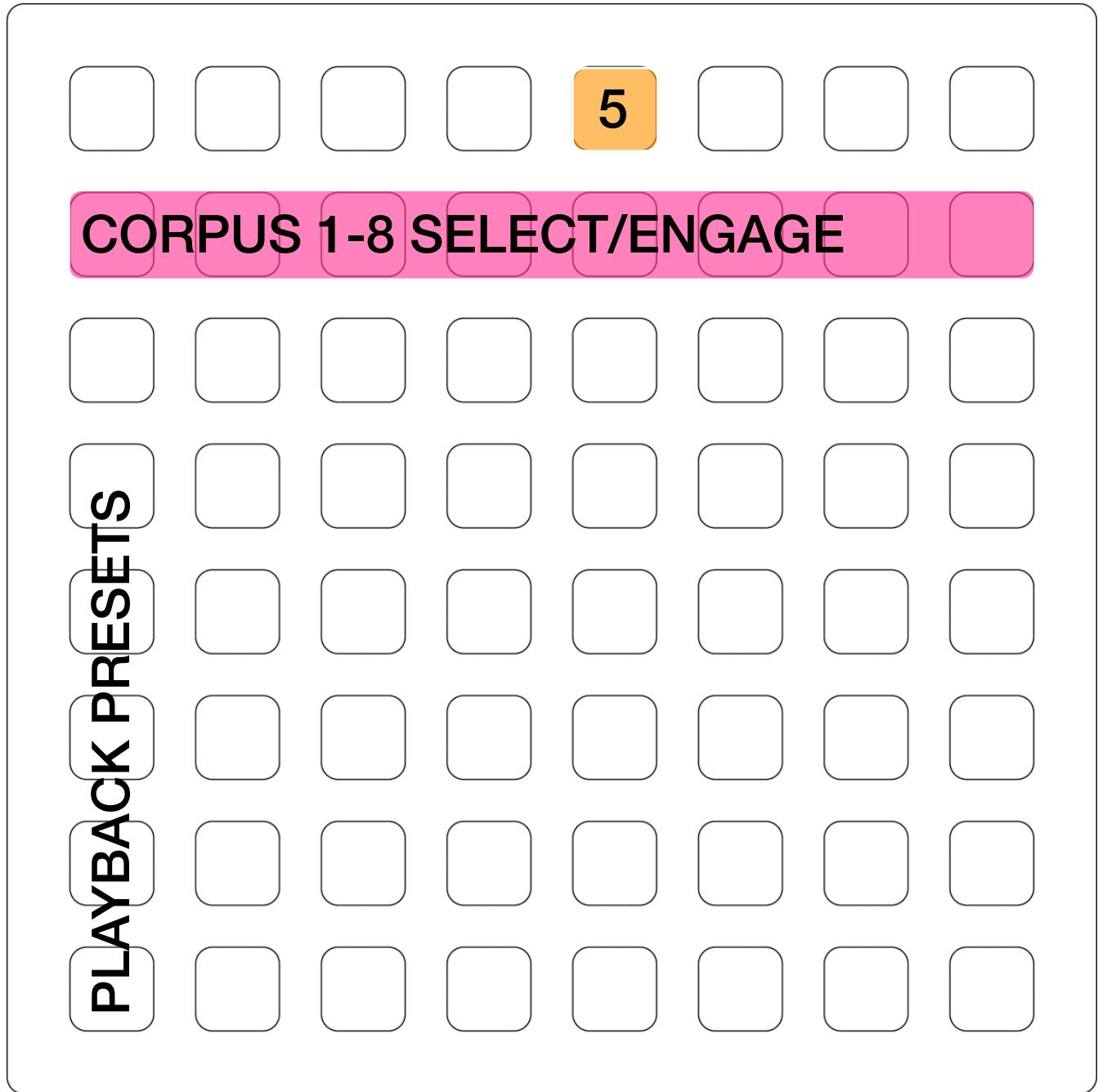
4

The Chocolate Grinder is a multi-mode buffer playback page. There are 8 separate buffers (one per column) and each can have a different playback mode. The playback modes are tcg, tml, and tbl. tcg is the classic ‘skipping CD player’ mode. tml is an adaptation of the monome app tml, and tbl is a turntable playback mode. Each mode has a characteristic sound/functionality, and the arc mappings change dynamically based on what mode you are in.

The record button functions different depending on the mode you are in. While in tcg/tbl the record button toggles on and off, whereas in tml mode, the record button is momentary (it only records while your finger is pressed).

The 2x/0.5x speed modifiers are aggregate meaning that pressing 2x twice will have the audio playing back 4x. Pressing both 2x and 0.5x at the same time resets the playback speed to 1x.

PAGE5 IS A CONCATENATIVE SYNTHESIS, AUDIO MOSAICING PLAYBACK ENGINE



DETAILED EXPLANATION OF C-C-COMBINE PAGE

5

C-C-Combine is a corpus-based audio mosaicing module, built based on concatenative synthesis. Concatenative synthesis is a technique for synthesizing sounds by concatenating short samples of recorded sound. It is essentially a type of granular synthesis that, instead of playing back grains based on position/windows, plays back grains based on analysis of incoming audio. In order to do this you must have a corpus (the pre-analyzed body of sounds) and an input sound source (either incoming audio or file playback).

To create a corpus follow the instructions in the included “create corpus.maxpat” subpatch. Once you have created a corpus you can load it by dragging the audio file onto the corpus waveform display (and then loading the analysis file when prompted). You can have up to 8 corpuses loaded and playing back at the same time. The first one is loaded via the GUI. For corpuses 2-8 you must rename the audio file and analysis file “corpus2” (or corpus3 etc..) and have them in the same folder as the main patch. The first time you load a corpus (including 2-8) it takes a few seconds to load so if you plan on using it in a performance, pre-load the corpuses so they are ready to use when you need them.

DETAILED EXPLANATION OF C-C-COMBINE PAGE CONTINUED...

5

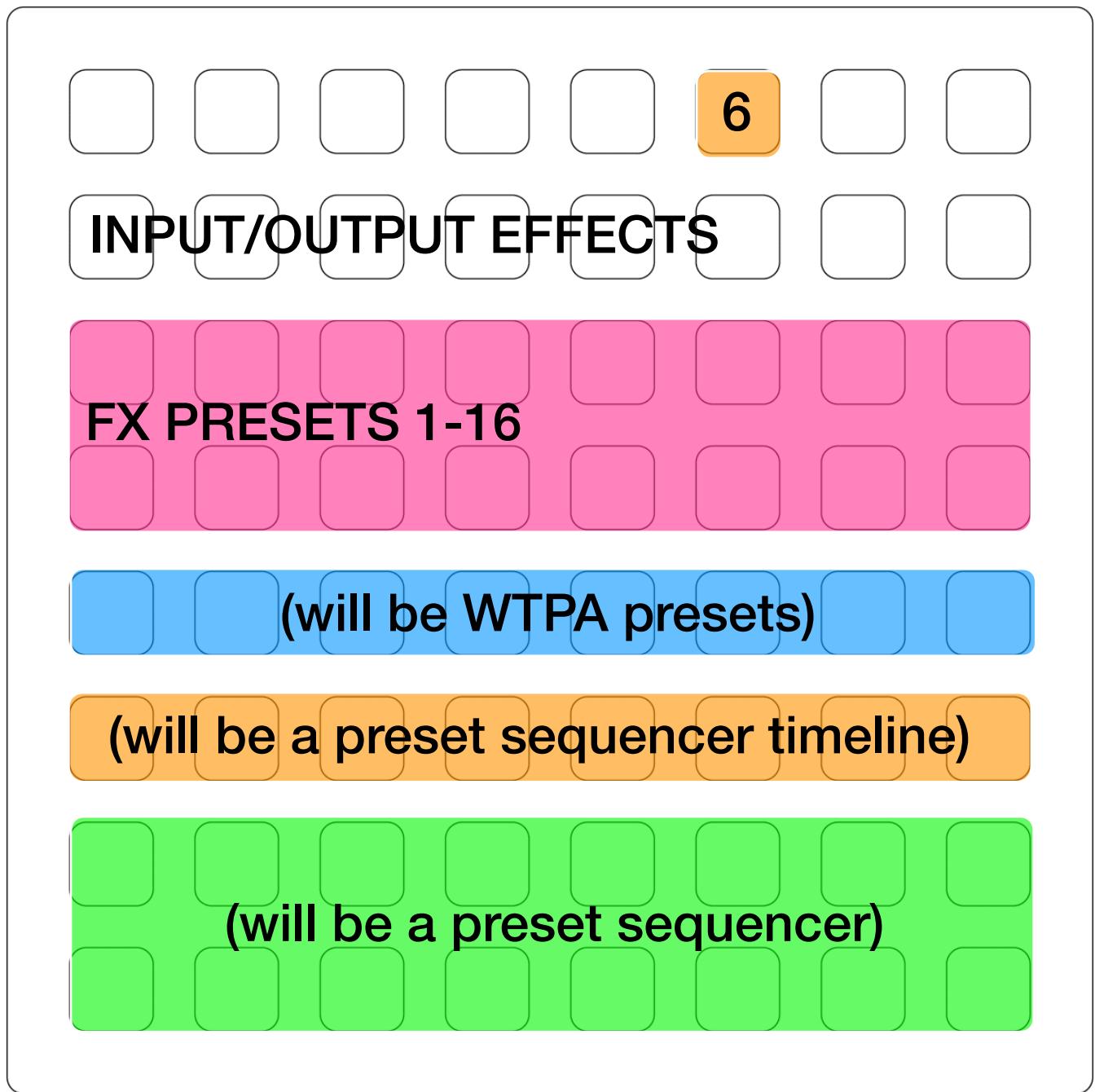
Each of the 8 corpuses can be engaged by turning on the appropriate button on the top row and used simultaneously meaning that each will individually try to match/replace grains with incoming audio. This can make for a 'hybrid' corpus of sorts as it is unlikely all corpuses will match the same grains. This page can get very CPU intensive so be wary of your usage.

Each engine has a preset row that lets you select a playback engine setting. These are as follows (from bottom to top):

1. Slow playback speed, short grains
2. Slow playback speed, long grains
3. Medium playback speed, short grains
4. Medium playback speed, long grains
5. Fast playback speed, short grains
6. Fast playback speed, long grains

Setting 6 consumes the most CPU as it produces the most simultaneous grains. Similarly, setting 1 consumes the least CPU. For using multiple corpuses at the same time it is recommended that you use lower settings. It will also sound better as there will be less overall clutter.

PAGE6 IS A PRESET SELECTION, MORPHING PAGE.



DETAILED EXPLANATION OF PRESET PAGE

6

This page is not complete yet, but it will eventually be a ‘high level’ performance page where you can create ‘song structures’ with sequences of presets that advance based on certain criteria (ie after 15 loud events, or 45seconds, or once a certain note is held for 2 seconds).

You will be able to create these kinds of sequences and then dynamically move through them in an mlr-like manner, creating loops.

At the moment the top row mirrors the FX row on Page1 and then next two rows are the first 16 FX presets.

PAGE7 IS AN INSTRUMENT-BASED PAGE WITH EACH INSTRUMENT BEING A SUBPAGE

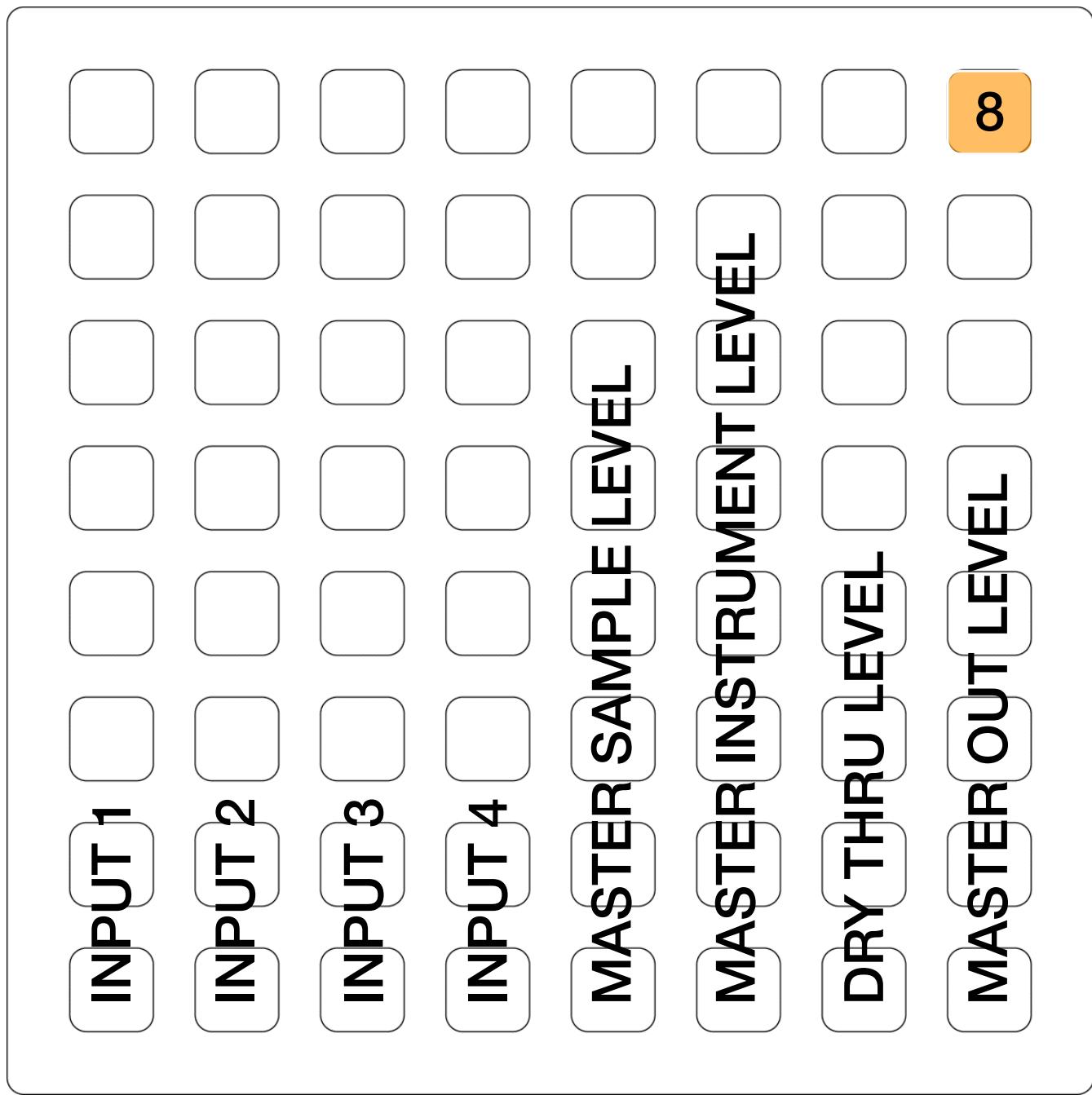
The image shows a grid of 8 numbered buttons (1-8) on a page labeled '7'. The buttons are arranged in two rows: the top row contains buttons 1 through 6, and the bottom row contains buttons 7 and 8. The button for '7' is highlighted with a yellow background, while the other buttons are white with black outlines. The page number '7' is also highlighted with a yellow background.

SUBPAGE DESCRIPTIONS:

- 1 - Adaptation of “Fourths” using a Synth
- 2 - Adaptation of “Fourths” using a Low/Hi Synth
- 3-4 Adaptations of “Corners”
- 5-8 Adaptations of “Straw”

Straw gets its pitch choices by analyzing incoming audio. The list of pitches is updated every time you revisit Page7. While on the instrument page, the pitches remain the same.

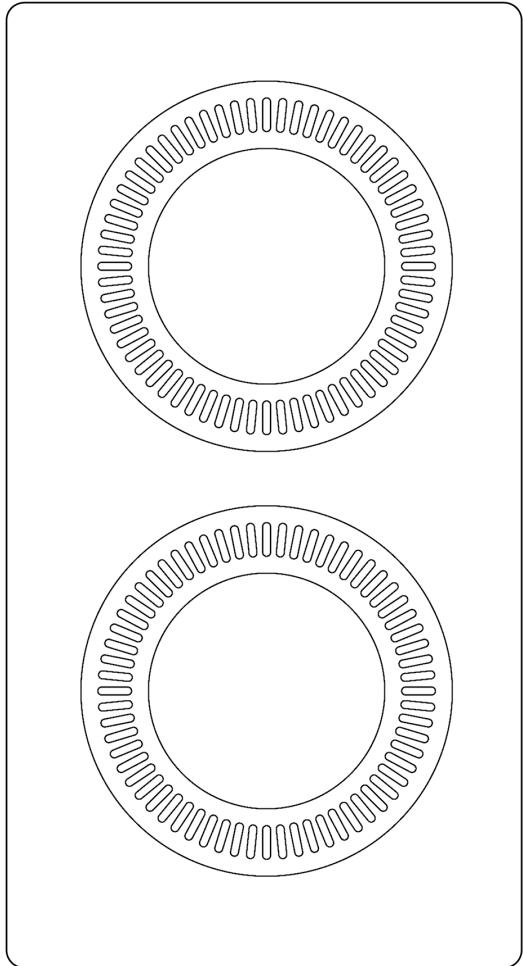
PAGE8 DEALS WITH INPUT AND OUTPUT LEVELS



ARC MAPPING

EACH MONOME ‘PAGE’ HAS IT’S OWN ARC MAPPING

(for newer arcs you can use key-presses (ZXCV) to simulate arc presses)



Page1 : Each encoder cycles through the following controls : loop1 pitch, loop2 pitch, wtpa1 fx control, wtpa2 fx control, loop1 start/end, loop2 start/end, in/out effects

Page2 : The top encoder is mapped to mlr1/mlr2 pitch and the bottom encoder is mlr3/mlr4 pitch.

Page3 : The top encoder is cocolase1 pitch, and the bottom one is cocolase2 pitch. Pressing the top one randomizes the ‘sidrassi brain’ modulation ‘petals’ and pressing the bottom one randomly repatches the modulation points.

Page4 : The top encoder controls position/window (for tcg/tml) and start/end (for tbl). Turning controls position/start and turn+press controls window/end. The bottom encoder controls ‘glitch’ when in tcg mode, ‘lfo’ when in tml mode and ‘scratch’ when in tbl mode. Pressing the bottom encoder creates a new position/window when in tcg mode, resets the lfo in tml mode and jumps to a new position in tbl mode.

Page5 : The top encoder controls overall “Combine” play-back level. The bottom encoder controls overall ‘selectiveness’ of each corpus. A higher setting means it be more particular about the grains that get matched. A lower setting with match more grains (at the expense of accuracy). Both encoders effect all corpuses.

Page6 : The top encoder controls the FX Brain ‘morph’ parameter. The bottom encoder controls the complex preset morphing described in the preset morphing section.

Page7 : Each subpage has it’s own mapping. On Subpages 1/2 the bottom encoder controls min velocity and the top encoder controls max velocity of each note.. On Subpages 3/4 the encoders control gravity (top) and friction (bottom). On Subpages5-8 each encoder controls a parameter for the “Straw” adaptation.

Page8 : The top encoder controls master volume. Pressing the top encoder mutes the patch. The bottom encoder cycles through the following controls : in1, in2, in3, in4, sample level, thru level.

TROUBLESHOOTING GUIDE

“I’ve opened the patch but nothing is happening”

Make sure the DSP engine is turned ON and that you have your soundcard and I/O selected properly in the “SETUP” window.

Also make sure you have installed the externals in the appropriate folder (Max6/Cycling ’74/msp-externals/).

“I can sample and playback but I can’t hear my dry audio”

Turn up the “thru” volume in the top right of the main window, but be careful if you are using a microphone as this can cause a feedback loop.

“My monome LEDs aren’t working right”

Make sure you have “OVERDRIVE” turned ON and that you have the right compatibility setting for your specific monome device in the “SETUP” window.

“My CPU usage is through the roof and causing dropouts”

Every module is coded to be as efficient as possible and most use no CPU at all when not in use. This means that you can really push the CPU by having many things going at once. It is probably possible to crash Max by engaging everything at once (though it hasn’t happened to me). The most CPU intensive sections are all the granular synthesis modules (“Grain”, and both “Combine” modules), convolution (inside the “Combine” module). The “Reverb” and “Dirt” modules also consume a fair amount of CPU.

STUFF

For any questions, comments, bugs, or feature requests contact me at rodrigo.constanzo@gmail.com

Special thanks to Alex Harker, Pierre Alexandre Tremblay, Dominic Thibault, Benjamin Van Esser, Thorsten Vieth, Morgan Lebihan and countless people on the cycling74 forum.

Patch can be downloaded at my webpage:
<http://www.rodrigoconstanzo.com>