

The logo for Expert Sleepers, featuring three stylized 'z' characters stacked vertically on the left side of the text.

# Expert Sleepers

## **disting mk4**

## **Firmware v4.21**

## **User Manual**

Copyright © 2021 Expert Sleepers Ltd. All rights reserved.

This manual, as well as the hardware and software described in it, is furnished under licence and may be used or copied only in accordance with the terms of such licence. The content of this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Expert Sleepers Ltd. Expert Sleepers Ltd assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

# Table of Contents

Introduction.....	<a href="#">5</a>	B-7 VCO with linear FM.....	<a href="#">29</a>
Installation.....	<a href="#">5</a>	B-8 VCO with waveshaping.....	<a href="#">30</a>
Power requirements.....	<a href="#">5</a>	C-1 Precision Adder (fractional offsets).....	<a href="#">32</a>
Inputs, Outputs and Controls.....	<a href="#">5</a>	C-2 Voltage Controlled Delay Line.....	<a href="#">32</a>
Inserting a MicroSD card.....	<a href="#">5</a>	C-3 Clockable Ping Pong Delay (Z feedback)	<a href="#">33</a>
Startup.....	<a href="#">6</a>	.....	<a href="#">33</a>
The encoder & menu system.....	<a href="#">6</a>	C-4 Clockable Ping Pong Delay (Z input pan)	<a href="#">33</a>
Selecting an algorithm.....	<a href="#">6</a>	.....	<a href="#">33</a>
Settings.....	<a href="#">7</a>	C-5 Resonator.....	<a href="#">34</a>
Help.....	<a href="#">7</a>	C-6 Vocoder.....	<a href="#">35</a>
Parameters.....	<a href="#">8</a>	C-7 Phaser.....	<a href="#">36</a>
Knob Recorder.....	<a href="#">8</a>	C-8 Bit Crusher.....	<a href="#">36</a>
Tap Tempo.....	<a href="#">9</a>	D-1 DJ Filter.....	<a href="#">38</a>
Favourites.....	<a href="#">9</a>	D-2 Tape Delay.....	<a href="#">38</a>
Presets.....	<a href="#">10</a>	D-3 Waveform Animator.....	<a href="#">39</a>
Preset 0.....	<a href="#">10</a>	D-4 State Variable Filter.....	<a href="#">40</a>
Select Bus.....	<a href="#">11</a>	D-5 LP/HP Filter.....	<a href="#">40</a>
SD Card Playback.....	<a href="#">12</a>	D-6 LP/BP Filter.....	<a href="#">41</a>
Supported MicroSD cards.....	<a href="#">12</a>	D-7 BP/HP Filter.....	<a href="#">41</a>
Supported audio files.....	<a href="#">12</a>	D-8 BP/Notch Filter.....	<a href="#">41</a>
Audio file naming & Playlists.....	<a href="#">12</a>	E-1 AR Envelope.....	<a href="#">42</a>
Loop markers in audio files.....	<a href="#">13</a>	E-2 AR Envelope & VCA.....	<a href="#">43</a>
Supported MIDI files.....	<a href="#">13</a>	E-3 Dual AR Envelope.....	<a href="#">44</a>
MIDI file naming & Playlists.....	<a href="#">13</a>	E-4 Stereo Compressor.....	<a href="#">45</a>
Audio Playlist Format.....	<a href="#">14</a>	E-5 Side-chain Compressor.....	<a href="#">46</a>
MIDI Playlist Format.....	<a href="#">16</a>	E-6 Mono Compressor.....	<a href="#">46</a>
Example Files.....	<a href="#">17</a>	E-7 Euro to Buchla Converter.....	<a href="#">47</a>
MIDI/CV conversion.....	<a href="#">17</a>	E-8 Buchla to Euro Converter.....	<a href="#">48</a>
Scala Support.....	<a href="#">17</a>	F-1 Clockable AD Envelope (with mute).....	<a href="#">48</a>
Organising the files on the card.....	<a href="#">17</a>	F-2 Clockable AD Envelope (with gate).....	<a href="#">49</a>
Example Files.....	<a href="#">18</a>	F-3 Clockable AD Envelope (with trigger).....	<a href="#">50</a>
Scale Playlist Format.....	<a href="#">18</a>	F-4 Clockable AD Envelope & VCA.....	<a href="#">51</a>
Possible Sources of Error.....	<a href="#">18</a>	F-5 Shift Register Random CVs.....	<a href="#">51</a>
Algorithm Overview.....	<a href="#">19</a>	F-6 Shift Register Random Quantized CVs.....	<a href="#">53</a>
A-1 Precision Adder.....	<a href="#">20</a>	F-7 Shift Register Random Triggers.....	<a href="#">54</a>
A-2 Four Quadrant Multiplier.....	<a href="#">20</a>	F-8 Shift Register Random Dual Triggers.....	<a href="#">55</a>
A-3 Full-wave Rectifier.....	<a href="#">21</a>	G-1 ES-1 Emulation.....	<a href="#">56</a>
A-4 Minimum/maximum.....	<a href="#">21</a>	G-2 ES-2 Emulation.....	<a href="#">56</a>
A-5 Linear/Exponential Converter.....	<a href="#">22</a>	G-3 Pitch Reference.....	<a href="#">57</a>
A-6 Quantizer.....	<a href="#">22</a>	G-4 Frequency Reference.....	<a href="#">57</a>
A-7 Comparator.....	<a href="#">24</a>	G-5 Tuner.....	<a href="#">58</a>
A-8 Dual Waveshaper.....	<a href="#">24</a>	G-6 Clock.....	<a href="#">58</a>
B-1 Sample and Hold.....	<a href="#">25</a>	G-7 MIDI/CV.....	<a href="#">61</a>
B-2 Slew Rate Limiter.....	<a href="#">26</a>	G-8 CV/MIDI.....	<a href="#">62</a>
B-3 Pitch and Envelope Tracker.....	<a href="#">26</a>	H-1 Crossfade/Pan.....	<a href="#">62</a>
B-4 Clockable Delay/Echo.....	<a href="#">27</a>	FX Loop usage.....	<a href="#">63</a>
B-5 LFO.....	<a href="#">28</a>	H-2 Dual Sample and Hold.....	<a href="#">63</a>
B-6 Clockable LFO.....	<a href="#">29</a>	H-3 Dual Quantizer (Z scale).....	<a href="#">64</a>

H-4 Dual Quantizer.....	<a href="#">65</a>	L-3 Dual Reverb.....	<a href="#">96</a>
H-5 Dual Euclidean Patterns.....	<a href="#">67</a>	L-4 Dual Vowel Filter.....	<a href="#">97</a>
H-6 Dual Delayed Pulse Generator.....	<a href="#">68</a>	L-5 Stereo Chorus.....	<a href="#">98</a>
H-7 Noise.....	<a href="#">69</a>	L-6 Mono Chorus.....	<a href="#">99</a>
H-8 Quantizer 2.....	<a href="#">69</a>	L-7 Mixer.....	<a href="#">100</a>
I-1 Audio Playback.....	<a href="#">71</a>	L-8 Gate.....	<a href="#">100</a>
I-2 Clocked Audio Playback.....	<a href="#">72</a>	M-1 Delayed LFO.....	<a href="#">101</a>
I-3 Audio Playback with V/Oct.....	<a href="#">73</a>	M-2 Scaled LFO.....	<a href="#">102</a>
I-4 Audio Playback with Z Speed.....	<a href="#">74</a>	M-3 Logic.....	<a href="#">103</a>
I-5 Audio Playback with Reverse.....	<a href="#">75</a>	M-4 Half-wave Rectifier.....	<a href="#">104</a>
I-6 Audio Playback with Scrub.....	<a href="#">75</a>	M-5 Stereo Filter.....	<a href="#">104</a>
I-7 Dual Audio Playback.....	<a href="#">76</a>	M-6 Stereo Tape Delay.....	<a href="#">105</a>
I-8 Dual Audio Playback with Z Speed.....	<a href="#">77</a>	M-7 Granular Pitch Shifter.....	<a href="#">106</a>
J-1 MIDI File Playback (Clocked).....	<a href="#">78</a>	M-8 Chaos.....	<a href="#">107</a>
J-2 Multisample 2 Audio Playback.....	<a href="#">78</a>	N-1 Switch.....	<a href="#">108</a>
J-3 MIDI File Playback (Free Running).....	<a href="#">79</a>	N-2 Rotary.....	<a href="#">109</a>
J-4 Audio Playback with End CV.....	<a href="#">80</a>	N-3 Attenuverter/Offset.....	<a href="#">110</a>
J-5 Audio Recorder.....	<a href="#">81</a>	N-4 Low Pass Gate.....	<a href="#">110</a>
J-6 Multisample Audio Playback.....	<a href="#">81</a>	N-5 Pulsar VCO.....	<a href="#">111</a>
Multisample playlists.....	<a href="#">82</a>	N-8 Clockable SD Ping Pong.....	<a href="#">112</a>
J-7 Mono Audio Recorder.....	<a href="#">84</a>	MIDI I/O.....	<a href="#">113</a>
J-8 Audio Playback with Crossfade.....	<a href="#">84</a>	Introduction.....	<a href="#">113</a>
K-1 Wavetable VCO.....	<a href="#">86</a>	Connections.....	<a href="#">113</a>
Loading wavetables.....	<a href="#">87</a>	Breakout module.....	<a href="#">113</a>
K-2 Clockable Wavetable LFO.....	<a href="#">88</a>	MIDI "Low-Voltage Signaling".....	<a href="#">113</a>
K-3 Wavetable Waveshaper.....	<a href="#">89</a>	Controlling the disting mk4 via MIDI.....	<a href="#">114</a>
K-4 Clockable Wavetable Envelope.....	<a href="#">89</a>	MIDI Thru.....	<a href="#">114</a>
K-5 Programmable Quantizer.....	<a href="#">90</a>	Calibration.....	<a href="#">115</a>
K-6 Clockable SD Delay.....	<a href="#">92</a>	Calibration Procedure.....	<a href="#">115</a>
K-7 Stereo Clockable SD Delay.....	<a href="#">93</a>	Calibration Data.....	<a href="#">115</a>
K-8 Stereo Clockable SD Delay (Z clock)..	<a href="#">93</a>	Firmware Updates.....	<a href="#">117</a>
L-1 Stereo Reverb.....	<a href="#">94</a>	Acknowledgments.....	<a href="#">118</a>
L-2 Mono-to-Stereo Reverb.....	<a href="#">95</a>		

## Introduction

Congratulations on your purchase of an Expert Sleepers disting mk4. Please read this user manual before operating your new module.

## Installation

House the module in a Eurorack case of your choosing. The power connector is [Doepfer standard](#). If using the power cable supplied with the module, the red edge of the cable is nearest the bottom of the PCB, and carries -12V. ("-12V" is marked on the PCB itself next to this end of the connector.) Be sure to connect the other end of the power cable correctly, again so -12V corresponds to the red stripe on the cable.

## Power requirements

The disting mk4 draws 51mA on the +12V rail, and 19mA on the -12V rail.

It does not use the 5V rail.

## Inputs, Outputs and Controls



From top to bottom, the disting has

- A rotary encoder/push button, named 'S'.
- A dot matrix display.
- A MicroSD card slot.
- A knob, referred to as the 'Z' control, which controls some aspect of the algorithm. This knob also has a push button function.
- Three input sockets.
  - The top input is the 'Z' CV input, which controls the same thing as the 'Z' knob. The two are added together.
  - The other two inputs are 'X' and 'Y', and their function depends on the current algorithm.
- Two output sockets, named 'A' and 'B'.

The sockets are illuminated to reflect the voltage at the socket (or in the case of 'Z', the combined voltage of the input and the knob). Red indicates a positive voltage; blue indicates a negative voltage.

## Inserting a MicroSD card

The disting's MicroSD card slot is to the left of the display.

The exposed contacts of the card should face towards the display; the angled side of the card itself points up.

The slot is of the "push-push" type - to remove the card, push it in slightly and it will spring out.

## Startup

### [Video](#)

When the module powers up it first runs through some patterns on its display. The sequence is as follows:

- Each of the seven columns in turn, from left to right.
- Each of the five rows in turn, from bottom to top.

It then shows the bootloader version (e.g. "b1") and the current firmware version (e.g. "v4.0").

If after the above the display shows the message "Uncalibrated", then the calibration data in flash is missing or corrupt. A default calibration will be used, but for accurate results, please recalibrate the module (see below).

## The encoder & menu system

### [Video](#)

Pressing the encoder ('S') enters the display's menu system, which is indicated by the display blinking.

At this time the menus are particularly simple, there being only six menu items to choose from, only one of which you will normally need. In general, the menu system works like this:

- Press the encoder to enter menu mode.
- Turn the encoder to select the menu item, and press to select.
- At this point the behaviour depends on the chosen menu item.

Pressing 'Z' at any time cancels and exits the menus.

The menu items are:

- Algorithm – select algorithm
- Save – save a preset
- Load – load a preset
- Help – access help
- Settings – change settings
- Calibrate – begin calibration
  - If you accidentally enter this mode, press the Z knob to cancel

## Selecting an algorithm

### [Video](#)

There is a single list of algorithms, named A1 up to P8 (i.e. A1, A2, ... up to A8, then B1, B2 etc.). See below for details.

The algorithm can be chosen via menu 1 "Algorithm". Press the encoder twice (to enter the menu, and to select "Algorithm"). Then turn the encoder to select the algorithm. Finally, press the encoder again to switch to the chosen algorithm.

# Settings

[Video](#) 

The disting has a number of settings, which are stored in flash memory. They are accessed by menu 4 "Settings". Once you've entered the Settings menu, turn the encoder to select the setting to change.

The settings are:

- **Brightness** – set the display brightness.
- **Recall enable** – enable/disable recall from the Select Bus (see below).
- **In MIDI ch** – the input MIDI channel.
- **Out MIDI ch** – the output MIDI channel.
- **Thru MIDI** – enables a soft MIDI thru.
- **Pgm Chng Alg** – controls whether MIDI program change messages select the current algorithm (1) or load a preset (0).
- **Cont filenames** – if set, WAV and MIDI filenames scroll continuously in the display; if unset, filenames scroll once when a new file is chosen, and then the display reverts to the default algorithm name view.
- **Scroll speed** – the speed at which text scrolls across the display.
- **WAV sample rate** – sample rate to use for the Audio Playback algorithms. The default (0) uses the disting's usual sample rate of 75kHz. The alternative (1) uses a reduced rate of 46875Hz. This can help reduce the load on the SD card, useful if e.g. you have a slower card.
- **SD delay max** – the maximum delay time, in minutes, for the various Clockable SD Delay algorithms. If the time between two taps is longer than this, the second will be ignored. This also affects how much of the delay file is wiped when the algorithm is chosen, which therefore affects how long the algorithm takes to start.
- **Switch param** – if set, when a MIDI CC message is received to change a parameter value, that parameter becomes the current parameter (and so its value is displayed).

Once you've chosen the setting to change, press the encoder to select it, at which point the setting's current value will be shown. Turn the encoder to change the value. Then press the encoder to accept and store the value into flash, or press Z to leave the menu without storing the value.

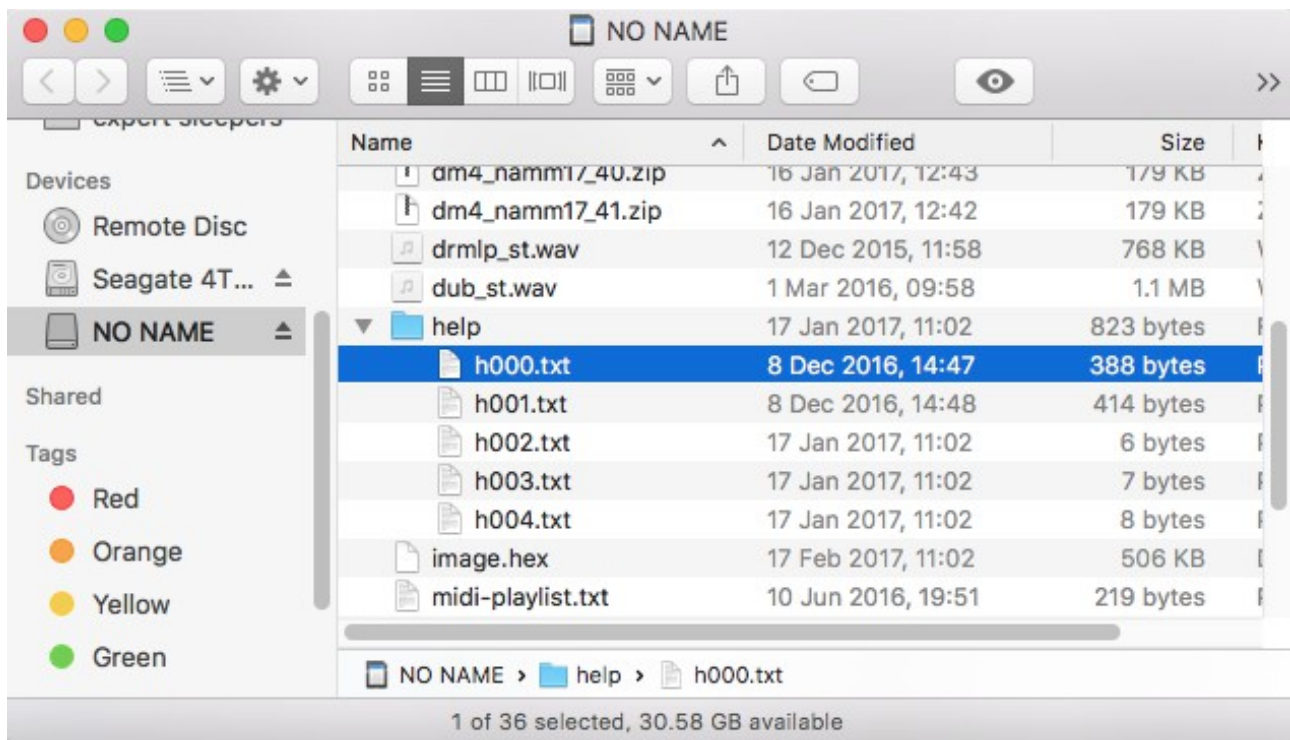
# Help

[Video](#) 

Accessing 'Help' from the menu displays algorithm-specific information. The help is loaded from the MicroSD card, if present (else the message "No card" is displayed), and scrolled across the display. To exit the help function, press Z.

The help text is a simple plain text file on the card, so it can be freely edited (e.g. to put your own choice of information first, or to translate it into a different language).

The default set of help files can be downloaded from the firmware update page [here](#). Simply unzip the file onto your MicroSD card. The file structure should look something like this:



i.e. the 'help' folder is at the root level of the card, and inside is a file per algorithm, named 'hXXX.txt', where XXX is the algorithm number, starting from zero for algorithm A1.

## Parameters

[Video](#) 

In addition to the control provided on the Z knob, each algorithm may have up to eight 'parameters', offering further control over the algorithm's operation.

Unlike the Z control, which is sampled at the same high rate as the X & Y inputs, parameters should be considered coarse, non-real-time adjustments to the algorithm. They may cause glitching in the outputs when changed.

Parameters are adjusted via the S knob. When changed, the parameter value is displayed for a couple of seconds.

If an algorithm has more than one parameter, pressing the Z knob cycles through them (unless the algorithm uses Z for Tap Tempo or some other purpose). The current parameter number is displayed briefly (prefixed with 'P') followed by its name.

The current parameter can also be selected by pushing in the S knob and turning it while holding it in. The parameter number (prefixed with 'P') is shown when you turn the knob; if you pause briefly without turning the knob, the parameter name is shown.

## Knob Recorder

[Video](#) 

In many algorithms which do not use a Z knob press for another purpose, pressing Z offers a 'knob



recorder' function.

When this is available, holding Z down while turning it causes the Z value changes to be recorded. As soon as the knob is released, the recording is played back in a loop. To end playback, turn the Z knob slightly to regain manual control.

The maximum duration of knob recording is a little under 14 seconds.

## Tap Tempo

[Video](#) 

Some algorithms (notably the Clockable LFO and Clockable Delay/Echo) use a press of the Z knob as a 'tap tempo' function. In these modes, pressing Z has the same effect as sending a clock pulse into the clock input. For example, pressing Z twice will set the delay time/LFO period to the time between the two button presses.

Note that if the algorithm is dividing down the clocked frequency, pressing Z twice will not set the output to the division of the time between the two presses; rather, it will be the time between a corresponding number of button presses. E.g. if the Clockable LFO is set to 1/4 (divide by 4), the LFO time will be set by pressing the Z button 5 times, and it will only change on the 5th press.

## Favourites

You can define your own list of 16 'favourite' algorithms, via a text file on the MicroSD card.

The favourites are accessed as algorithms O1-P8. If no favourites are defined, or if there are empty slots, the disting runs algorithm A1, the Precision Adder.

The favourites are read from the SD card, if present, at power on. They are then stored in the disting's flash memory, and so persist even after the card is removed.

The favourites file is a simple text file which must be named 'favourites.txt' and placed in the root of the SD card (not inside any folder). The contents look something like this:

```
disting favourites v2
B8 VCO
a1
C5 resonator
I4 SD z speed
b5 LFO
e6 dual AR w/ push
```

The first line must be "disting favourites v2". Subsequent lines specify an algorithm to add to the list of favourites. After 16 algorithms, any further lines are ignored.

Each algorithm is specified as its two character name (a letter followed by a number). Any further text after the algorithm number is ignored, so you can add a descriptive string (as in the example above) to help you remember what's what.

## Presets

[Video](#) 

The disting can store its current state to flash memory as a preset. The contents of a preset are

- the chosen algorithm.
- the parameter values.
- the current parameter number.

Anything controlled by the Z knob/CV is not stored – its value is always defined by the knob position and CV input.

There are 64 preset slots. Presets are saved and recalled using the menu. To save a preset:

1. Press the encoder to enter menu mode.
2. Turn the encoder once (to show 'Save'), and press to confirm.
3. Turn the encoder to choose the preset slot.
4. Press the encoder to complete the save.

As usual, pressing Z at any point exits the menu without saving.

To load a preset:

1. Press the encoder to enter menu mode.
2. Turn the encoder twice (to show 'Load'), and press to confirm.
3. Turn the encoder to choose the preset slot.
4. Press the encoder to complete the load.

If the current algorithm was selected via the Favourites, the actual algorithm chosen is saved in the preset, not the favourites slot.<sup>1</sup>

### **Preset 0**

Preset 0, the first slot, is special:

- The contents of preset 0 are loaded at power-up.
- When switching algorithm, the new algorithm state is saved to preset 0.<sup>2</sup>

Therefore if you want to save the disting's state so it powers up as it is currently, you can manually save to preset 0, which is “click, turn, click, click” on the encoder.

Preset 0 is also an exception to the above comment regarding favourites. The actual favourites slot is stored.

---

<sup>1</sup> So if you change the favourites, the presets are unaffected.

<sup>2</sup> The thinking here is that if you forget to save your preset before turning off the power, at least the disting will come back up in the same algorithm next time you start.

## Select Bus

[Video](#) 

If enabled in the settings, the disting mk4 will respond to save/recall messages on the Select Bus.

The Select Bus is a means of inter-module communication currently supported by a handful of modules from various manufacturers, including the [Malekko Varigate 8±](#), [Macro Machines Storage Strip](#) and the [Make Noise Tempî & René 2](#).

Note that the disting only responds to save/recall messages. It does not initiate them.

Any MIDI that arrives on the Select Bus that is not explicitly part of the Select Bus protocol will be used as if it arrived on the disting's regular MIDI input – see below.

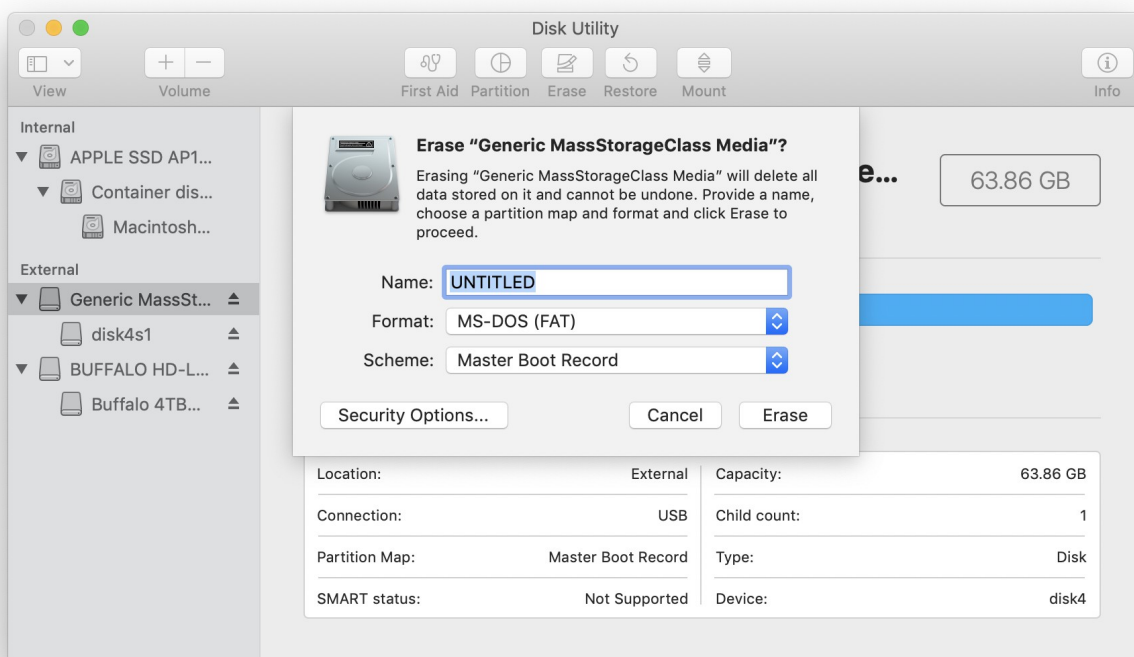
# SD Card Playback

## Supported MicroSD cards

In theory any MicroSD card will work, but in practice there is a huge variety of cards on the market and we cannot possibly test them all. We recommend 32GB SDHC cards, and in particular have found "SanDisk Extreme 32GB microSDHC UHS-I U3" cards work well.

The MicroSD card must be formatted in FAT32 format, which is the as-sold state for many cards. If not, cards can easily be reformatted to FAT32 in Windows or macOS.

The card also needs to be partitioned using the 'Master Boot Record' (MBR) scheme. In macOS's Disk Utility, you may need to select View->Show All Devices to inspect the partitioning scheme. The Erase settings should look like this:



## Supported audio files

As of now, the only supported audio file format is 16 bit WAV, mono or stereo. Any sample rate can be used, however.

Since higher sample rates impose greater bandwidth demands on the SD card streaming, we suggest you use the lowest sample rate that gives acceptable audio quality. All our testing generally uses 44.1kHz files.

Note that because the disting's outputs are DC-coupled, the 'audio' files can actually contain recorded or generated CVs, so all the sample playback modes can also be used as complex modulation or sequencing sources.

## Audio file naming & Playlists

Audio files must have the extension '.wav' (or any other capitalisation of that e.g. '.WAV').

The files should be placed in folders on the SD card. Unless noted otherwise, audio playback algorithms all support up to 100 folders of 100 sample files each. The folders themselves should be at the top level of the card i.e. not inside any higher level folder.

The disting also supports 'playlist' files, which are simple text files that instruct the disting which files/folders to look at and which ones to ignore, and also allow setting of options on a per-sample file basis. Unless noted otherwise these files are optional.

The playlist filename is 'playlist.txt'. If a playlist is found in the root of the card, it will be taken as a list of folders in which to look for WAV files. If a playlist is found inside a folder of WAV files, it will be used to determine which files to use, the order in which to use them, and to set the per-file options.

Note that it is explicitly allowed to have a playlist file which only contains global settings, but lists no actual WAV files. In this case, the settings in the playlist will be used, and the folder scanned for WAV files automatically.

If playlists are not used, folder and file names are sorted alphabetically.

### ***Loop markers in audio files***

The disting supports reading loop information embedded in the WAV file. If this information is not present, or is disabled with the playlist option (see below), any mode that loops the sample simply loops the whole file.

Loops are inferred either from markers (cue points in WAV format parlance) or regions, as follows:

1 marker point in file	Marker is assumed to be loop start; loop is from the marker to the end of the sample.
2 marker points in file	Markers are used as loop start and end.
3 or more marker points in file	First marker is ignored (assumed to be playback start point); second and third markers used as loop points. Remaining markers ignored.
1 or more regions in file	First region is used as the loop; other regions and markers ignored.

This video illustrates the use of loop markers:



### ***Supported MIDI files***

Currently we support Format 0 files (single track). The division field of the header chunk must be in "ticks per quarter note" format.

Happily this is the format that Ableton Live spits out if you do "Export MIDI Clip".

### ***MIDI file naming & Playlists***

All MIDI files need to be in a folder named 'MIDI'<sup>3</sup>. They can use any naming convention but the extension '.mid' is usual.

A playlist file (see below) specifies what MIDI files the algorithm will use, how they will be ordered, and what playback settings they will use. It is a simple text file, as described below, and is

<sup>3</sup> Note that this changed in firmware 4.19 – previously the MIDI files went in the card root.

optional. If no playlist is provided, the MIDI playback algorithms search for any '.mid' files in the 'MIDI' folder. Of course, if you do not provide a playlist, you cannot customise the settings for the MIDI files.

By default, MIDI algorithms look for a file named 'midi-playlist.txt'. However, each MIDI playback algorithm first looks for an algorithm-specific playlist file, the name of which is specified in the algorithm descriptions below.

The playlist files must be plain text. Note that TextEdit on macOS defaults to rich text. From the 'Format' menu choose, 'Make Plain Text'.

## **Audio Playlist Format**

An audio playlist file might look something like this:

```
disting playlist v1
-loop=0
-gap=0
-fadeIn=0
-fadeOut=0
TR66_BD_aOrig_b_R1.wav
-fixedPitch=0.9
TR66_Snare_aOrig_b_R1.wav
sample01.wav
-loop=1
-ramp=1
funky_lp.wav
-loop=1
-triggers=4
586447main_JFKwechoosemoonspeech.wav
-loop=1
```

This breaks down as:

- The header: "disting playlist v1"
- Default settings (optional)
- List of audio files
- Per-file settings after each file (optional)

Files may appear more than once in the playlist, with the same or different settings.

Settings are in the format "-" <setting name> "=" <setting value>

Settings that appear before any audio file apply globally to all the files in the playlist, unless overridden by the per-file settings.

The available settings are as follows.

<b>Setting</b>	<b>Default</b>	<b>Description</b>
loop	1	Whether the sample loops (value 1) or is a one-shot (value 0).
fadeOut	3	Fade out to apply when a sample is retriggered, in milliseconds.
fadeIn	3	Fade in to apply when a sample is retriggered, in milliseconds.
gap	3	Gap (silence) between fade out and fade in, in milliseconds.
retriggerOnSampleChange	1	See below.

Setting	Default	Description
fixedPitch	0	Play the sample at a fixed pitch, not controlled by the pitch CV (if any).
ramp	0	Switch Output B to emit a ramp CV.
triggers	0	Switch Output B to emit trigger pulses.
clocks	4	Set the number of clocks per loop for the Clocked Audio Playback algorithm.
wavelength	600	Set the number of sample frames per waveform for the Wavetable VCO algorithm.
natural	0	The file's natural pitch as a MIDI note number.
switch	0	The file's switch point as a MIDI note number.
playToCompletion	0	Prevents a one-shot sample being retriggered while playing.
useStartOnSampleChange	0	Whether switching to a new sample continues from an equivalent playback position or restarts.
startQuantize	0	Sets the quantisation of the start position.
useLoopMarkers	1	Use loop markers from the WAV file, if available.

The fades are useful when retriggering samples, especially when starting them at arbitrary points within the sample, to avoid clicks. However, for maximum responsiveness when triggering, say, drum samples, set the fades and gap to zero.

The default behaviour (with `retriggerOnSampleChange=1`) when selecting a new sample (via S or Z depending on algorithm) is to immediately begin playing the new sample. If a sample has `retriggerOnSampleChange=0` (or, if the default is set to 0) then the next sample does not begin playback until triggered by the input. A typical use case for this would be when you have a set of one-shot (say, drum) samples and you want to trigger each sample and have it play until the next trigger, not changing in the middle if the sample select CV changes. At an extreme, you could in this case feed white noise into the sample select CV, to choose a random sample on each trigger.

The `fixedPitch` setting is a floating point number, where 1.0 corresponds to playing the sample at its natural speed, 0.5 means play it at half speed etc. `fixedPitch=0` (the default) means that the sample respects the CV-controlled pitch.

`ramp=1` outputs a rising ramp from 0V to +8V corresponding to the current position in the sample. `ramp=-1` outputs a falling ramp (+8V to 0V).

`triggers` outputs a number of triggers during the sample e.g. `triggers=4` will output 4 trigger pulses, equally spaced over the sample length. The triggers are +5V pulses lasting about 10ms. The maximum number of triggers per sample is 32767. This video demonstrates the triggers feature:

[Video](#) 

`ramp` and `triggers` are mutually exclusive. The last setting in the playlist file for a given sample is the one that takes effect.

`natural` and `switch` are used only by the Multisample Audio Playback algorithm (see below).

`playToCompletion` applies to one-shot (non-looping) samples only. If set, the sample can not be retriggered, either by a trigger CV or by the current sample changing, until it has played all the way through.

`useStartOnSampleChange` controls the behaviour when the current sample is changed, without an explicit trigger (e.g. when turning the Z knob in the I-1 Audio Playback algorithm).

When this is 0 (the default), the new sample starts playing at the same point relative to the start as the previous sample. When set to 1, the new sample starts at the point defined by the start position CV (if the algorithm has one) or at the beginning.

`startQuantize` applies to any algorithm where playback can be retriggered and which provides a CV for the start position. The length of the file will be equally divided into the number of steps set by `startQuantize`, and retriggering will restart playback at the start of one of these steps. For example if you have a one bar loop (in 4/4 time) and specify `startQuantize=4` then playback will always be triggered at the beginning of a beat within the sample.

The maximum number of sample files per playlist is 100. Files beyond that limit are ignored.

## **MIDI Playlist Format**

A MIDI playlist file might look something like this:

```
disting playlist v1
-zeroVNote=60
bach_2ptinv.mid
CDE_bend.mid
-bendRange=12
CDE_bend.mid
-loop=0
cc1_2.mid
-cc1offset=-1.5
-cc1scale=3
-cc2offset=2
-cc2scale=6
```

This breaks down as:

- The header: "disting playlist v1"
- Default settings (optional)
- List of MIDI files
- Per-file settings after each file (optional)

Files may appear more than once in the playlist, with the same or different settings.

Settings are in the format "-" <setting name> "=" <setting value>

Settings that appear before any MIDI file apply globally to all the files in the playlist, unless overridden by the per-file settings.

The available settings are as follows.

Setting	Default	Description
loop	1	Whether the MIDI file loops (value 1) or is a one-shot (value 0).
zeroVNote	48	MIDI note number corresponding to zero Volts output pitch CV.
bendRange	2	Pitch bend range, in semitones.
cc1offset	0	Offset for CC#1, in Volts.
cc1scale	5	Range for CC#1, in Volts.
cc2offset	0	Offset for CC#2, in Volts.
cc2scale	5	Range for CC#2, in Volts.



retriggerOnSampleChange	1	See below.
-------------------------	---	------------

The default behaviour (with `retriggerOnSampleChange=1`) when selecting a new file is to immediately begin playing the new file. If a file has `retriggerOnSampleChange=0` (or, if the default is set to 0) then the next file not begin playback until triggered by the input.

The maximum number of MIDI files per playlist is 32. Files beyond that limit are ignored.

## Example Files

A simple example playlist and some audio files which are known to be formatted correctly can be downloaded [here](#)<sup>4</sup>.

## MIDI/CV conversion

There are three primary modes of MIDI/CV conversion, which can be mixed within a MIDI file:

1. Note on/off messages, and pitch bend messages, are used to generate a pitch CV from output A and a gate CV from output B.
2. The two highest MIDI notes (126 & 127, aka F#9 & G9) are converted to gates on outputs A & B respectively. The amplitude of the gates is 5V scaled by the note velocity.
3. CCs #1 & #2 are used to directly control voltages on outputs A & B respectively. The 7 bit CC value (0-127) is converted to a voltage according to the scale and offset settings (by default, 0-5V).

## Scala Support

Some disting algorithms support microtonal scales, using files generated by the program Scala<sup>5</sup>. The files are loaded from the MicroSD card. See 'Supported MicroSD cards' above.

## Organising the files on the card

To use Scala with the disting, you will need a MicroSD card set up as follows.

- **An 'scl' folder.** A folder named 'scl' in the root folder of the card, which contains all the scale (.scl) files that you wish to use. Note: you are advised not to simply unzip the archive of over 4000 scales available from the Scala website onto the card – you will find the disting slow to load scales if you do.
- **A 'kbm' folder.** A folder named 'kbm' in the root folder of the card, which contains all the keyboard mapping (.kbm) files that you wish to use.
- **The log tables.** The disting uses precomputed mathematical tables when processing the Scala files. Download the file 'logTables\_16\_20.bin' from [here](#)<sup>6</sup> and place it in the root of your SD card.
- **The playlist.** A file named 'playlist-scales.txt' which tells the disting which scales to use.

4 [http://expert-sleepers.co.uk/downloads/content/disting\\_sd\\_content\\_v4\\_8.zip](http://expert-sleepers.co.uk/downloads/content/disting_sd_content_v4_8.zip)

5 <http://www.huygens-fokker.org/scala/>

6 [http://expert-sleepers.co.uk/downloads/content/logTables\\_16\\_20.bin](http://expert-sleepers.co.uk/downloads/content/logTables_16_20.bin)

## Example Files

A zip file containing some example files, which are known to work, is [here](#)<sup>7</sup>.

## Scale Playlist Format

A scale playlist file might look something like this:

```
disting playlist v1
-kbm=example.kbm
equal.scl
pyth_7a.scl
pyth_12.scl
johnson_7.scl
-kbm=7.kbm
equal.scl
-kbm=root+fifth.kbm
```

This breaks down as:

- The header: "disting playlist v1"
- Default settings (optional)
- List of scale files
- Per-file settings after each file (optional)

Files may appear more than once in the playlist, with the same or different settings.

Settings are in the format "-" <setting name> "=" <setting value>

Settings that appear before any scale file apply globally to all the files in the playlist, unless overridden by the per-file settings.

Currently there is only one setting, which is the name of the keyboard mapping file to use with the scale file.

## Possible Sources of Error

If there is a problem reading or interpreting the scale files, the disting will not be able to use them.

Possible things to check include:

- The scale and keyboard mapping files must obviously be valid Scala format files, that the Scala application itself will load successfully.
- The scale and keyboard mapping must match – for example, the keyboard mapping must not refer to scale degrees that the scale does not have.
- For the keyboard mapping files, the reference frequency must be given for a note that is actually mapped.

---

<sup>7</sup> [http://expert-sleepers.co.uk/downloads/content/disting\\_example\\_scales.zip](http://expert-sleepers.co.uk/downloads/content/disting_example_scales.zip)

## Algorithm Overview

	1	2	3	4	5	6	7	8
A	Precision Adder	Four Quadrant Multiplier	Full-wave Rectifier	Minimum/maximum	Linear/Exponential Converter	Quantizer	Comparator	Dual Waveshaper
B	Sample and Hold	Slew Rate Limiter	Pitch and Envelope Tracker	Clockable Delay/Echo	LFO	Clockable LFO	VCO with linear FM	VCO with waveshaping
C	Precision Adder (fractional offsets)	Voltage Controlled Delay Line	Clockable Ping Pong (Z feedback)	Clockable Ping Pong (Z input pan)	Resonator	Vocoder	Phaser	Bit Crusher
D	DJ Filter	Tape Delay	Waveform Animator	State Variable Filter	LP/HP Filter	LP/BP Filter	BP/HP Filter	BP/Notch Filter
E	AR Envelope	AR Envelope & VCA	Dual AR Envelope	Stereo Compressor	Side-chain Compressor	Mono Compressor	Euro to Buchla Converter	Buchla to Euro Converter
F	Clockable AD Envelope (with mute)	Clockable AD Envelope (with gate)	Clockable AD Envelope (with trigger)	Clockable AD Envelope & VCA	Shift Register Random CVs	Shift Register Random Quantized CVs	Shift Register Random Triggers	Shift Register Random Dual Triggers
G	ES-1 Emulation	ES-2 Emulation	Pitch Reference	Frequency Reference	Tuner	Clock	MIDI/CV	CV/MIDI
H	Crossfade/Pan	Dual Sample and Hold	Dual Quantizer (Z scale)	Dual Quantizer	Dual Euclidean Patterns	Dual Delayed Pulse Generator	Noise	Quantizer 2
I	Audio Playback	Clocked Audio Playback	Audio Playback with V/Oct	Audio Playback with Z Speed	Audio Playback with Reverse	Audio Playback with Scrub	Dual Audio Playback	Dual Audio Playback with Z Speed
J	MIDI File Playback (Clocked)	Multi-sample 2 Audio Playback	MIDI File Playback (Free Running)	Audio Playback with End CV	Audio Recorder	Multi-sample Audio Playback	Mono Audio Recorder	Audio Playback with Crossfade
K	Wavetable VCO	Clockable Wavetable LFO	Wavetable Waveshaper	Clockable Wavetable Envelope	Programmable Quantizer	Clockable SD Delay	Stereo Clockable SD Delay	Stereo Clockable SD Delay (Z clock)
L	Stereo Reverb	Mono-to-Stereo Reverb	Dual Reverb	Dual Vowel Filter	Stereo Chorus	Mono Chorus	Mixer	Gate
M	Delayed LFO	Scaled LFO	Logic	Half-wave Rectifier	Stereo Filter	Stereo Tape Delay	Granular Pitch Shifter	Chaos
N	Switch	Rotary	Attenuverter /Offset	Low Pass Gate	Pulsar VCO			Clockable SD Ping Pong

## A-1 Precision Adder

[Video](#) 

$$A = X + Y + \text{offset, or } X + \text{offset}$$

$$B = X - Y - \text{offset, or } Y \pm \text{offset}$$

$$\text{offset} = \pm 10V \text{ in } 1V \text{ steps derived from } Z$$

**Knob recorder enabled**

Parameter	Min	Max	Default	Description
0	0	1	0	Z Mode.
1	0	2	0	Sum Mode.

Outputs A & B are simple sums of inputs X & Y and an offset, according to parameter 1:

Parameter value	Output A	Output B	Notes
0	Sum of inputs X & Y, plus the offset.	Difference between inputs X & Y, minus the offset.	With nothing plugged into input X, B is therefore simply an inverted copy of Y.
1	X plus the offset.	Y minus the offset.	This is particularly useful for shifting between bipolar (e.g. $\pm 5V$ ) and unipolar (e.g. 0-10V) CVs and back again.
2	X plus the offset.	Y plus the offset.	

The Z knob/CV sets an offset which is applied to both A and B. The offset is a whole number of Volts. If X/Y are 1V/Octave pitch CVs, Z is therefore an octave shift control. The maximum shift is 10V, positive or negative.

When Z changes, the offset is shown on the display. After a short while the display reverts to showing the current algorithm.

When parameter 0 is 1, the offset is not restricted to 1V steps, and changes smoothly with Z.

## A-2 Four Quadrant Multiplier

[Video](#) 

$$A = X * Y * \text{scale}$$

$$B = -X * Y * \text{scale}$$

$$\text{scale} = 1/10 \text{ to } 10x \text{ in steps derived from } Z$$

**Knob recorder enabled**

Parameter	Min	Max	Default	Description
0	0	1	0	Z Mode.
1	-10	10	0	Y Offset.

Output A is the result of multiplying inputs X & Y. Output B is the inverse of output A.

If for example X is a signal and Y is an envelope, then this algorithm is a VCA. If both inputs are signals, then this is a ring modulator.

The Z knob/CV sets a scale factor which is applied to both outputs. The scale is an integer (whole number) which either multiplies or divides the result, and ranges in value from 1-10.

When Z changes, the scale is shown on the display. If the value shown is negative, it is a divisor rather than a multiplier. Note that there is no value of "-1" since dividing by 1 and multiplying by 1 are the same. After a short while the display reverts to showing the current algorithm.

When parameter 0 is 1, the scale is not restricted to integer steps, and changes smoothly with Z.

Parameter 1 applies an offset to the Y input, in whole Volts. This is convenient if you just want to multiply the X input by a constant, which you can then dial in with the Y Offset and Z.

### **A-3 Full-wave Rectifier**

[Video](#) 

**A = abs( X + Y ) or abs( X )**  
**B = abs( X - Y ) or abs( Y )**  
**Z selects mode**

This algorithm provides a full-wave rectifier or absolute value function. The Z knob/CV select between one of two modes. In 'independent' mode, A and B are the absolute values of X and Y, respectively. In 'combined' mode, A is the absolute value of the sum of X & Y; B is the absolute value of the difference of X & Y.

When Z changes, the mode is displayed. 'Independent' mode is shown as "-0"; 'combined' mode as "0". After a short while the display reverts to showing the current algorithm.

### **A-4 Minimum/maximum**

[Video](#) 

**A = min( X, Y )**  
**B = max( X, Y )**  
**Z is gate**

Output A is the minimum of inputs X & Y; output B is the maximum of the two inputs. If one input is zero (or disconnected), this is a half-wave rectifier.

The Z knob/CV provides a gate function. When Z goes higher than approximately 2.5V, the gate goes high and the outputs follow the inputs according to the min/max relationship. When Z goes below approximately -1.5V, the gate goes low and the outputs are frozen.

When the gate changes state, "LO" or "HI" is displayed to indicate the gate status. After a short while the display reverts to showing the current algorithm.

## A-5 Linear/Exponential Converter

[Video](#) 

$A = (2^X) * \text{scale}$   
 $B = \log_2(Y / \text{scale})$   
Z is Hz/V scale, centred on 1kHz

Parameter	Min	Max	Default	Description
0	0	1	0	Type A.
1	0	1	0	Type B.

This algorithm provides a linear-to-exponential converter and an exponential-to-linear converter. You might use this to interface 1V/octave modules (Eurorack standard) with Hz/V synths (e.g. old Korg or Yamaha synths), but it could also be useful within Eurorack e.g. to convert an LFO (commonly with Hz/V pitch control) to a V/octave oscillator, or to convert an exponential FM input on a VCO into a linear FM input.

Input X is the exponential input; its corresponding linear output is A. Y is the linear input, whose exponential output is B.

Z sets the scale factor which is common to both conversions. It sets the number of Hz per Volt, with arrange from near zero to about 2kHz. The Yamaha CS-15, for example, uses about 1100Hz/V, which is about half way on the Z knob here.

The zero Volt point on the exponential scale used is C3 (approximately 130.81Hz).

Parameters 0 & 1 switch their respective outputs to function as gate inverters. (Some old synths e.g. the Korg MS-20 have a Hz/V pitch input and an inverted gate, compared to the Eurorack norm.) The output is then 5V/0V for an input gate in excess of 1V.

## A-6 Quantizer

[Video](#) 

$A = \text{quantized}(X)$   
 $B = \text{trigger on note change}$   
Z chooses scale & function of Y  
Y = transpose (Z positive) or trigger (Z negative)

### Outputs & Receives MIDI

Parameter	Min	Max	Default	Description
0	0	32	32	Input X attenuation.
1	0	1	0	Transpose mode.
2	-31	31	0	Key.
3	0	31	0	Offset.
4	0	1	0	MIDI Gate.
5	-48	48	0	Transpose.

Output A is a quantized version of input X; the closest whole-semitone value to the unquantized V/octave pitch CV X. Output B is a trigger signal which fires whenever output A changes - a 5V pulse approximately 10ms long.

As well as providing a chromatic scale, this algorithm can also constrain the quantized values to a

musical scale or chord. This is controlled by the Z knob/CV.

When Z changes, the scale is displayed on the LEDs. The scale's number in the following table is shown, and after a short pause, the scale's name.

Number	Scale	Displayed name
0	chromatic	Chromatic
1	major scale	Major
2	minor scale	Minor
3	major triad	Triad
4	minor triad	3b+5
5	root+5th	Fifth
6	major triad+6th	Triad+6
7	minor triad+6th	3b+5+6
8	major triad+7th	Triad+7
9	minor triad+7th	3b+5+7
10	root+5th+6th	5+6
11	root+5th+7th	5+7
12	pentatonic major	Pent
13	pentatonic minor	Minor Pent
14	natural minor scale	Nat Minor
15	harmonic minor scale	Harm Minor

A further option is controlled by Z being positive or negative. When Z is negative, the scale number is shown as a negative value.

**When Z is positive** input Y is a transpose control. The CV on input Y is quantized (to a chromatic scale) and added to the quantized result in one of two ways, controlled by parameter 1. If parameter 1 is zero, the transposition is applied to output A, after input X has been quantized to the chosen scale. If parameter 1 is one, the transposition is applied to input X before it is forced into the chosen scale. So, in mode zero the overall key of the quantization is transposed, whereas in mode one the transposition moves the notes around within the same key.

**When Z is negative** input Y is a trigger. In this mode, input X is only sampled and converted to a new quantized value when input Y rises over approximately 1V. (In non-triggered mode, X is constantly sampled and a new note is output as soon as X moves into the next semitone range.)

Parameter 2 sets the root **key** of the chosen scale. At zero, the first note of the scale (e.g. C in the key of C) corresponds to 0V. If the parameter is set for example to 2, the first note of the scale is at  $2/12 = 0.1667V$  - or to look at it another way, if your VCO is tuned so that 0V gives you a C, the quantizer is now working in the key of D (D major, minor, triad etc. depending on the scale setting).

Parameter 3 sets an offset between the timing of the X & Y inputs (in sample frames). When using Y as a trigger, it is often a good idea to delay it slightly relative to the pitch, to ensure that the pitch has changed and settled before the trigger samples a new note.

Parameter 5 applies transposition, exactly as per the description of using Y for transposition above. Note that transposition with parameter 5 is always available, even if Y is being used as a trigger.

**MIDI Output:** each new note (i.e. when the Output B trigger fires) is output as a MIDI note message on the MIDI channel selected in the settings.

Parameter 4 controls when note off messages are sent. If parameter 4 is zero (the default), the note off is sent just before the note on – so the note lasts from one change to the next, which is conceptually the same as the CV output. If parameter 4 is one, and Z is negative (so Y is a trigger), then the trigger is treated as a gate, and the note off message is sent when the gate goes low.

**MIDI Input:** MIDI note messages received on MIDI channel 1 set parameter 2 (root key).

## A-7 Comparator

[Video](#) 

**A = gate from X > Y**  
**B = inverted gate**  
**Z is hysteresis**

Parameter	Min	Max	Default	Description
0	-40	40	0	X Offset.
1	-40	40	0	Y Offset.

Output A is a gate signal (zero or +5V), high when input X has a higher voltage than input Y. Output B is an inverted copy of A (i.e. +5V when A is 0V and vice versa.)

The Z knob/CV input sets the hysteresis (for an explanation of hysteresis see [here](#)). Inputs of approximately 0-4V correspond to hysteresis values of approximately 0-10V. Negative values are clamped at zero.

Parameters 0 & 1 apply offsets to inputs X & Y respectively (scaled such that a parameter value of 40 corresponds to 10V).

## A-8 Dual Waveshaper

[Video](#) 

**A = shaped X**  
**B = shaped Y**  
**Z is gain**

**Knob recorder enabled**

Parameter	Min	Max	Default	Description
0	0	1	0	Type A.
1	0	1	1	Type B.

This algorithm provides two independent waveshaping functions. The Z knob/CV is a gain control, with a range of approximately 30x. Negative values of Z invert the signal.

Output A provides a waveshaped version of input X; output B provides a waveshaped version of input Y.

Parameters 0 & 1 choose what type of waveshaping is applied.

Type 0 is what is usually termed a wavefolder. This increases the harmonic content of the sound in interesting ways, especially as the gain changes.

Type 1 is a triangle-to-sine waveshaper. Used on most audio this is a relatively gentle form of overdrive/saturation. However, when fed with the right level of triangle wave, the output is exactly a sine wave, which is useful when you have a triangle wave VCO handy but really want a pure sine



wave instead.

## B-1 Sample and Hold

[Video](#) 

**A = X when Y exceeds 1V**  
**B = noise  $\pm 8V$**   
**Z is slew rate**

Parameter	Min	Max	Default	Description
0	0	1	0	Mode.
1	0	31	0	Offset.
2	0	3	1	Noise colour.
3	0	64	0	Noise X.
4	0	32	32	Atten B.
5	-1	31	-1	Up slew.
6	-1	31	-1	Down slew.

### If parameter 0 is 0 (the default):

Output A is a sample of input X, taken when the trigger input Y goes over 1V. (Sample and Hold)

### If parameter 0 is 1:

Output A follows input X while input Y is over 1V. When Y falls below 0.5V, output A is held until Y goes over 1V again. (Track and Hold)

Output B is a noise signal. Noise is commonly fed into the input of a sample and hold device to generate clocked random voltages. The amplitude of output B is set by parameter 4; the maximum is  $\pm 8V$ .

Parameter 2 controls the noise colour:

Parameter value	Noise colour
0	Violet
1	White
2	Pink
3	Red

The Z knob/CV controls the slew rate of output A. At the minimum value of Z, changes in A are instantaneous. As Z increases, changes in A take place more slowly.

Parameters 5 & 6 allow you to specify different slew rates when the signal rises or falls. If the parameter is -1, the slew time is set from Z; otherwise the slew is set from the parameter value.

Pressing the Z knob triggers a sample manually.

Parameter 1 sets an offset between the timing of the X & Y inputs (in sample frames). If X & Y are being driven together, for example by a sequencer's pitch and gate, it is often a good idea to delay the trigger slightly to allow time for the sampled CV to settle before the sample is taken.

Parameter 3 adds the noise output into the input signal, equivalent to patching output B into input X via an attenuator.

## B-2 Slew Rate Limiter

[Video](#) 

**A = slew rate limited ( X + Y ) or X**  
**B = slew rate limited ( X + Y ) or Y**  
**Z is slew rate**

Parameter	Min	Max	Default	Description
0	-1	32	-1	Up slew.
1	-1	32	-1	Down slew.
2	0	1	0	Type A.
3	0	1	1	Type B.
4	0	1	0	Mode.

Outputs A & B are slew rate limited versions of X & Y. If parameter 4 is 0, they are both based on the sum of X + Y; if parameter 4 is 1, A is based on X, and B is based on Y.

Parameters 2 & 3 choose the type of slew rate limiting applied to A & B respectively. A parameter value of 0 gives linear slew rate limiting; a step change in the input will typically result in a ramp output, until the output reaches its desired value, at which point it will be constant. A parameter value of 1 gives logarithmic slew rate limiting; a step change in input results in a smooth curve that gradually approaches the desired value.

The Z knob/CV controls the slew rate for both outputs. At the minimum value of Z, changes are very rapid. As Z increases, changes take place more slowly.

Parameters 0 & 1 allow for independent control of the slew times when the signal is rising or falling. If each parameter is -1, the corresponding slew rate is controlled by Z. Otherwise, the slew rate is set by the parameter value.

## B-3 Pitch and Envelope Tracker

[Video](#) 

**A = V/octave pitch derived from X, plus Y**  
**B = envelope derved from X**  
**Z is slew rate for envelope**

This algorithm provides pitch and envelope tracking of an incoming audio signal. It will track frequencies down to about 27Hz, which is just below the lowest note on a standard 88 key piano.

Output A is a 1V/octave pitch CV reflecting the pitch of the signal on input X. The 0V point is C3 (approximately 130.81Hz). Input Y is simply added to the pitch CV, providing a means of applying e.g. vibrato, or transposition.

Output B tracks the envelope of the signal on input X. It goes to zero when the algorithm fails to track a pitch.

Knob/CV Z sets the slew rate of the envelope, controlling how quickly it tracks changes in level. At the minimum value of Z, changes can be very rapid, which may produce undesirable effects, especially if pitch tracking is not working well. As Z increases, changes take place more slowly.

## B-4 Clockable Delay/Echo

[Video](#) 

**X is signal**  
**Y is clock input**  
**Z is feedback**  
**A = output according to mode**  
**B = output according to mode**

**Tap tempo enabled**

Parameter	Min	Max	Default	Description
0	-15	8	0	Delay time multiplier.
1	0	2	0	Output mode.

This algorithm is a delay/echo effect, primarily intended for processing audio signals, where the delay time is set from a clock pulse. It operates at a quarter of the standard sample rate (i.e. at about 19kHz) and offers a maximum delay time of about 1700ms.

Input X is the signal input. Any audio signal can be fed in here.

Input Y is the clock input. Any clock pulse in excess of 1V can be used. The time between rising trigger edges is used to set the delay time. If the time between triggers is greater than the maximum delay time, the time is divided by two until it is small enough. This way, you always end up with a rhythmically useful delay time.

The Z knob/CV controls the feedback, from zero to slightly more than 100%.

Outputs A & B carry signals according to parameter 1:

Parameter value	Output mode
0	A is mix; B is delay only.
1	A & B are mix.
2	A & B are delay only.

'Mix' is a mix of the dry (undelayed) signal and the delay effect. The amount of delay in the mix rises in direct proportion to the amount of feedback.

'Delay only' is the delayed signal only. Use this and the input signal, plus an external mixer, if you need more flexibility in the dry/wet balance than is offered by the mixed mode.

The first parameter applies a multiplier to the delay time, according to the following table:

Parameter value	Multiplier	Parameter value	Multiplier	Parameter value	Multiplier
-15	1/64	-7	3/16	1	x1.5
-14	1/48	-6	1/4	2	x2
-13	1/32	-5	5/16	3	x3
-12	1/24	-4	1/3	4	x4
-11	1/16	-3	3/8	5	x5
-10	1/12	-2	1/2	6	x6
-9	1/8	-1	3/4	7	x8
-8	1/6	0	x1	8	x16

## B-5 LFO

[Video](#) 

**X is Hz/V frequency**

**Y is waveshape**

**Z is tune**

**A is saw -> sine -> triangle**

**B is pulse -> square -> pulse**

Parameter	Min	Max	Default	Description
0	0	40	32	Sets an attenuation for output A.
1	0	40	32	Sets an attenuation for output B.
2	-40	40	0	Sets an offset for output A.
3	-40	40	0	Sets an offset for output B.
4	-32	32	0	Y offset.

Outputs A & B are LFOs (low frequency oscillators), with CV control of frequency and waveshape. The output signals are  $\pm 8V$  (16V peak-to-peak) by default, but can be attenuated via the parameters.

The outputs can also have a DC offset applied. The offset in Volts is (parameter value)/4 i.e. the range is  $\pm 10V$ , in steps of 0.25V.

Input X is a Hz/V frequency control, scaled at 1Hz/V. Note that the input is allowed to go negative, resulting in a phase-reversed output.

Knob/CV Z is a tuning control, with a range of approximately  $\pm 10Hz$ . This is simply added to the setting from input X (so with input X disconnected, the knob can be used to manually set an LFO rate).

Input Y controls the waveshape of the output signals. Signals in the range  $\pm 5V$  give the full range of possible waveshapes:

Input Y	-5V	0V	+5V
Output A	saw	sine	triangle
Output B	0% duty cycle pulse	50% duty cycle pulse (square)	100% duty cycle pulse

Parameter 4 offsets the Y input, providing a way to set the waveshape manually.

## B-6 Clockable LFO

[Video](#) 

**X is clock input**  
**Y is waveshape**  
**Z is integer multiplier/divider**  
**A is saw -> sine -> triangle**  
**B is pulse -> square -> pulse**

**Tap tempo enabled**

Parameter	Min	Max	Default	Description
0	-32	32	32	Output A attenuverter.
1	-32	32	32	Output B attenuverter.
2	-32	32	0	Y Offset.

Outputs A & B are LFOs (low frequency oscillators), with CV control of waveshape, and with the LFO cycle time set from a clock input. The output signals are  $\pm 8V$  (16V peak-to-peak) by default, but can be attenuated and inverted via parameters 0 & 1.

Input X is the clock input. Any clock pulse in excess of 1V can be used. The time between rising trigger edges is used to set the cycle time.

Input Y controls the waveshape of the output signals. Signals in the range  $\pm 5V$  give the full range of possible waveshapes:

Input Y	-5V	0V	+5V
Output A	saw	sine	triangle
Output B	0% duty cycle pulse	50% duty cycle pulse (square)	100% duty cycle pulse

Parameter 2 is an offset added to the Y CV, allowing manual selection of the waveshape.

The Z knob/CV sets a scale factor which is applied to the cycle time. The scale is an integer (whole number) which either multiplies or divides the frequency, and ranges in value from 1-16.

When Z changes, the scale is shown on the display. If the value shown is negative, it is a divisor rather than a multiplier. Note that there is no value of "-1" since dividing by 1 and multiplying by 1 are the same.

## B-7 VCO with linear FM

## Video

**X is V/Oct pitch input**  
**Y is linear FM input**  
**Z is tune  $\pm 0.5$  octaves**  
**A & B are selectable output waveforms**

### Receives MIDI

Parameter	Min	Max	Default	Description
0	-16	8	0	Octave shift.
1	0	32	32	Sets an attenuation for output A.
2	0	32	32	Sets an attenuation for output B.
3	0	5	1	Type A.
4	0	5	3	Type B.

This algorithm is a VCO with a 1V/octave pitch CV input (X), and a linear FM input (Y), scaled at 100Hz/V. Note that, if the FM input goes sufficiently negative, it will take the frequency through and below zero, resulting in a phase inversion ("thru-zero FM").

The 0V point for the pitch input is C3 (approximately 130.81Hz).

The Z knob/CV provides a tuning control, with a range of approximately  $\pm 0.5$  octaves.

The A and B outputs provide waveforms according to the following table:

Parameter value	Waveform
0	Triangle.
1	Sine.
2	Square.
3	Saw.
4	Sub-octave square.
5	MIDI gate.

Parameters 3 & 4 control the waveform of outputs A & B respectively. The output signals are  $\pm 8V$  (16V peak-to-peak) by default, but can be attenuated via parameters 1 & 2.

**MIDI Input:** MIDI note messages received on the MIDI channel set in the Settings are converted to a CV which is added to that on input X. The 0V note is 48 (C3). MIDI notes also drive the 'MIDI gate' output option.

### ***B-8 VCO with waveshaping***

**Video** 

**X is V/Oct pitch input**  
**Y is waveshape/PWM**  
**Z is tune  $\pm 0.5$  octaves or sync**  
**A & B are selectable output waveforms**

**Receives MIDI**

Parameter	Min	Max	Default	Description
0	-16	8	0	Octave shift.
1	0	32	32	Sets an attenuation for output A.
2	0	32	32	Sets an attenuation for output B.
3	-32	32	0	Offset for input Y (manual shape control).
4	0	4	0	Selects output waveforms.
5	-12	12	0	Transpose.
6	-51	50	-51	Fine tune/enable sync.

This algorithm is a VCO with a 1V/octave pitch CV input (X), and waveshape/PWM input (Y).

The 0V point for the pitch input is C3 (approximately 130.81Hz).

The Z knob/CV provides a tuning control, with a range of approximately  $\pm 0.5$  octaves (but see also parameter 6, below).

Parameter 4 selects the output waveforms according to the following table:

Parameter value	WaveformA	WaveformB
0	Triangle/saw	Square/pulse
1	Triangle/saw	Sub-octave square
2	Square/pulse	Sub-octave square
3	Triangle/saw	MIDI gate.
4	Square/pulse	MIDI gate.

Input Y controls the waveshape of the output signals. Signals in the range  $\pm 5V$  give the full range of possible waveshapes:

Input Y	-5V	0V	+5V
<b>Triangle/saw</b>	saw (falling)	triangle	saw (rising)
<b>Square/pulse</b>	0% duty cycle pulse	50% duty cycle pulse (square)	100% duty cycle pulse

The output signals are  $\pm 8V$  (16V peak-to-peak) by default, but can be attenuated via parameters 1 and 2.

Parameter 3 offsets the Y input, effectively providing a manual shape control if required.

Parameter 5 transposes the VCO pitch, in semitone increments.

Parameter 6 controls the function of Z. At the default value of -51, Z is the tune control as already mentioned. When the parameter is set to a value from -50 to 50, Z no longer affects the tuning, and

the parameter value is used as a fine tune control, in cents. Z then becomes a sync input: when Z goes over approximately 1V, the VCO phase is reset to zero, which can produce 'oscillator sync' sounds<sup>8</sup>.

**MIDI Input:** MIDI note messages received on the MIDI channel set in the Settings are converted to a CV which is added to that on input X. The 0V note is 48 (C3). MIDI notes also drive the 'MIDI gate' output option.

### C-1 Precision Adder (fractional offsets)

**A = X + Y + offset**  
**B = X - Y - offset**  
**offset in steps derived from Z**

**Knob recorder enabled**

Parameter	Min	Max	Default	Description
0	1	12	12	Sets the offset divisor.
1	0	2	0	Sum mode.

This algorithm is basically the same as A-1 Precision Adder above, except that where the other algorithm has offsets in steps of 1V, this algorithm allows you to set a divisor for the offset. This defaults to 12, so when used to offset a 1V/octave pitch CV the offsets correspond to semitones.

### C-2 Voltage Controlled Delay Line

[Video](#) 

**X is audio input**  
**Y is delay time**  
**Z is feedback (bipolar)**  
**A is delay output**  
**B is delay output plus input signal**

Parameter	Min	Max	Default	Description
0	-32	32	0	Applies an offset to the Y input.
1	-32	32	32	Attenuator for the Y input.
2	0	1	0	Enable saturation.

This algorithm is a voltage controlled delay line, with a maximum delay time of just over 200ms. As well as simple echoes, this can produce a wide variety of effects such as vibrato, chorus and flange.

A control signal of 0-8V on the Y input sets the delay time, with a linear voltage/time relationship. Parameter 0 can be used to set this to a 'centre' value, which makes it easier to patch in an LFO to the Y input without having to add a DC offset to the LFO.

Parameter 1 is an attenuverter for Y, applied before the offset.

Z is a feedback control. It is zero at 0V input, and provides negative feedback for negative voltages.

A outputs the delay signal only. Use this for vibrato effects, or if you want a controllable mix of dry and delayed signal.

<sup>8</sup> [https://en.wikipedia.org/wiki/Oscillator\\_sync](https://en.wikipedia.org/wiki/Oscillator_sync)



B outputs the delay and dry signals mixed in equal amounts. In combination with an LFO on the delay time, this is the quickest route to chorus/flange type effects.

Parameter 2 enables a saturation stage on the input to the delay line. This can prevent unpleasant distortion when the feedback becomes high, at the expense of some colouration of the sound.

### **C-3 Clockable Ping Pong Delay (Z feedback)**

[Video](#) 

**X is audio input**  
**Y is clock**  
**Z is feedback**  
**A is left output**  
**B is right output**

**Tap tempo enabled**

Parameter	Min	Max	Default	Description
0	0	1	0	Output mode.
1	-15	8	0	Delay time multiplier.
2	-32	32	-32	Input pan.

This algorithm is a stereo ping-pong delay/echo effect, primarily intended for processing audio signals, where the delay time is set from a clock pulse. It operates at a quarter of the standard sample rate (i.e. at about 19kHz) and offers a maximum delay time of about 900ms.

Input X is the signal input. Any audio signal can be fed in here.

Input Y is the clock input. Any clock pulse in excess of 1V can be used. The time between rising trigger edges is used to set the delay time. If the time between triggers is greater than the maximum delay time, the time is divided by two until it is small enough. This way, you always end up with a rhythmically useful delay time.

Parameter 1 applies a multiplier to the delay time, according to the table above.

The Z knob/CV controls the feedback, from zero to slightly more than 100%.

Outputs A & B are the left and right outputs respectively. If the output mode parameter is 0 (the default), they are a mix of the dry (undelayed) signal and the delay effect. The amount of delay in the mix rises in direct proportion to the amount of feedback. If the output mode parameter is 1, the outputs are the delayed signals only. Use this and the input signal, plus an external mixer, if you need more flexibility in the dry/wet balance than is offered by output mode 0.

Parameter 2 controls the left/right pan position of the input signal.

### **C-4 Clockable Ping Pong Delay (Z input pan)**

## [Video](#)

**X is audio input**  
**Y is clock**  
**Z is input pan**  
**A is left output**  
**B is right output**

### Tap tempo enabled

Parameter	Min	Max	Default	Description
0	0	31	24	Feedback.
1	-15	8	0	Delay time multiplier.
2	0	1	0	Output mode.

This algorithm is a stereo ping-pong delay/echo effect, primarily intended for processing audio signals, where the delay time is set from a clock pulse. It operates at a quarter of the standard sample rate (i.e. at about 19kHz) and offers a maximum delay time of about 900ms.

Input X is the signal input. Any audio signal can be fed in here.

Input Y is the clock input. Any clock pulse in excess of 1V can be used. The time between rising trigger edges is used to set the delay time. If the time between triggers is greater than the maximum delay time, the time is divided by two until it is small enough. This way, you always end up with a rhythmically useful delay time.

Parameter 1 applies a multiplier to the delay time, according to the table above.

The Z knob/CV controls the left/right pan position of the input signal.

Outputs A & B are the left and right outputs respectively. If the output mode parameter is 0 (the default), they are a mix of the dry (undelayed) signal and the delay effect. The amount of delay in the mix rises in direct proportion to the amount of feedback. If the output mode parameter is 1, the outputs are the delayed signals only. Use this and the input signal, plus an external mixer, if you need more flexibility in the dry/wet balance than is offered by output mode 0.

The delay feedback is set via parameter 0.

## **C-5 Resonator**

### [Video](#)

**X is audio or trigger input**  
**Y is centre frequency (pitch)**  
**Z is gain**  
**A is audio output**  
**B is envelope of audio output**  
**Z press is 'strike'**

Parameter	Min	Max	Default	Description
0	-32	32	0	Applies an offset to the Y input.
1	0	1	0	Mode.

This algorithm implements a resonator, which is a kind of filter, but which is also often found as the basis for analogue drum synthesis.

X is the input for the signal to filter. If used as a drum synth, this is the trigger input. The amplitude and nature of the trigger signal will affect the resulting sound. If parameter 1 is set to '1', the input is treated as an actual trigger, not an audio signal, triggering a pulse when the input goes over 1V.

Y sets the pitch of the resonator, with a 1V/octave response. The 0V point for the pitch input is C3 (approximately 130.81Hz). Parameter 0 provides a pitch offset in semitones.

Z sets the gain. In terms of drum synthesis, more gain means a longer decay time.

A is the audio output.

B outputs the result of envelope tracking the audio output A.

Pressing Z simulates hitting X with a 1ms 5V pulse, and so works as a manual trigger of the 'drum'.

## C-6 Vocoder

[Video](#) 

**X is modulator input**  
**Y is carrier input**  
**Z sets the decay time**  
**A is audio output**  
**B is envelope output**

Parameter	Min	Max	Default	Description
0	0	1	0	Selects the filter bank.
1	0	64	16	Output A attenuation/gain.
2	0	64	16	Output B attenuation/gain.

This algorithm implements a vocoder. The spectral characteristics of the modulator input are applied to the carrier input. In classic usage, the modulator might be a human voice, and the carrier might be a synth sound, or simply noise.

X is the modulator input, and Y is the carrier input.

A is the audio output. B outputs a CV related to the envelope of the modulator signal.

The Z control sets the decay time of the internal envelope trackers, which track each band of the modulator signal. Use low values (negative Z) for most intelligible speech.

Parameter 0 selects between alternative filter banks:

Bank	Description
<b>0</b>	Half octave spacing, based on 100Hz.
<b>1</b>	Third octave spacing, based on 250Hz.

Parameters 1 & 2 provide attenuation or gain of outputs A & B respectively.

## C-7 Phaser

[Video](#) 

**X is audio input**  
**Y is sweep**  
**Z is feedback (bipolar)**  
**A is phase-shifted output plus input signal**  
**B is phase-shifted output**

Parameter	Min	Max	Default	Description
0	-31	31	0	Applies an offset to the Y input.
1	1	10	10	Sets the number of phaser stages.

This algorithm implements a phase shifter or 'phaser' effect.

X is the audio input.

Y is the sweep input. 0V to 8V covers the whole range; negative voltages are treated as 0V. You will almost always want to patch an LFO in here. Parameter 0 can be used to set this to a 'centre' value, which makes it easier to patch in an LFO to the Y input without having to add a DC offset to the LFO.

Z controls feedback. More feedback results in more extreme phasing effects. When Z is negative, the feedback is inverted, which gives a different-sounding phasing effect.

A outputs the combination of the phase-shifted signal and the original signal, which is usually what you need for a classic phaser effect, since it's the interaction of the original signal and the phase-shifted version which produces the 'comb filtering' effect.

Output B provides just the phase shifted signal, if you need more control over how this and the original signal are mixed.

Parameter 1 sets the number of phase shifting stages. The more stages, the more notches there are in the comb filter, which results in a more pronounced effect.

## C-8 Bit Crusher

[Video](#) 

**X is signal input**  
**Y is sample rate input**  
**Z sets bit reduction**  
**A is signal output**  
**B is comparator output**

Parameter	Min	Max	Default	Description
0	-31	31	0	Applies an offset to the Y input.
1	0	7	0	Selects the bit reduction mode.
2	0	7	0	Selects the bit mangling mode.

This algorithm is a 'bit crusher' - it applies sample rate and sample depth reduction to deliberately introduce quantisation and aliasing artefacts. It also optionally applies bitwise modification of the samples for non-linear distortion effects.

X is the signal input.

Y is the sample rate reduction CV input. It has a 1V/octave response i.e. the sample rate is reduced by a factor of 2 for every 1V rise in CV. This CV is offset by parameter 0, so you can set a rate reduction without a CV input.

A is the signal output.

B is the output of a comparator. Output A is compared against zero; output B is +5V if A is above zero, or 0V if A is below zero.

The Z control sets the bit depth reduction. There are two types of bit reduction available:

- **Type I** - the signal is quantised to a 16 bit word, and the low bits thrown away. The resulting signal uses a power of 2 bits. Changing between bit depths is therefore discontinuous.
- **Type II** - quantisation is achieved via the limited precision of integer maths when dividing the signal by a factor. Since the factor can be continuously varied, this offers a smooth variation between 'bit depths'.

Furthermore, positive and negative values of Z have different effects.

- Positive Z treats the whole signal range as one number to be quantised.
- Negative Z treats positive and negative sections of the input signal differently. Negative sections are flipped positive, quantized, and flipped back.

Parameter 1 selects the type of bit reduction. The positive and negative sections of the input signal can have different types of reduction applied.

Parameter 1 value	Positive signal	Negative signal
0	Type I	Type I
1	Type II	Type II
2	Type I	Type II
3	Type II	Type I
4	Type I	None
5	Type II	None
6	None	Type I
7	None	Type II

Parameter 2 selects the bit mangling mode.

Parameter 2 value	Bit mangling
0	None
1	Bit swap variant 1
2	Bit swap variant 2
3	Bit swap variant 3
4	Bit rotation
5	Previous sample XOR variant 1
6	Previous sample XOR variant 2
7	Previous sample XOR variant 3

## D-1 DJ Filter

[Video](#) 

**X & Y are stereo audio input**  
**A & B are stereo audio output**  
**Z is filter sweep**

Parameter	Min	Max	Default	Description
0	0	31	0	Filter resonance.

This is stereo filter which sweeps from lowpass to highpass on a single knob.

The Z knob/CV sets the filter cutoff. At the central, zero position no filter effect is applied. If the knob is turned anticlockwise, a lowpass filter is applied, with a cutoff frequency that sweeps down the more you turn. If the knob is turned clockwise from centre, a highpass filter is applied, with a cutoff frequency that sweeps up the more you turn.

## D-2 Tape Delay

[Video](#) 

**X is audio input**  
**Y is tape speed**  
**Z is feedback**  
**A = output according to mode**  
**B = output according to mode**

Parameter	Min	Max	Default	Description
0	0	43	43	Tape length.
1	-9	9	0	Fine length control.
2	-8	16	0	Tape speed.
3	0	2	0	Output mode.

This is a delay/echo effect which simulates a variable speed tape loop echo device. The delay time at '1x speed', with the tape length parameter at maximum, is just over 400ms.

X is the audio input.

Parameters 0 & 1 together set the 'tape length', effectively setting the delay time. Parameter 0 is scaled in 10ms increments, while parameter 1 is in 1ms increments (both assuming 1x 'tape speed').

Y controls the speed of the 'tape', which also affects the delay time. The voltage/speed relationship is 8V/octave. If you consider 0V as '1x speed', then +8V gives '2x speed' (the maximum) and -4V gives 'half speed' (the minimum). Parameter 2 can be used to set the speed manually; its value is added to that of the Y input.

The Z knob/CV controls the feedback, from zero to slightly more than 100%.

Outputs A & B carry signals according to parameter 3:

Parameter value	Output mode
0	A is mix; B is delay only.
1	A & B are mix.
2	A & B are delay only.

'Mix' is a mix of the dry (undelayed) signal and the delay effect. The amount of delay in the mix rises in direct proportion to the amount of feedback.

'Delay only' is the delayed signal only. Use this and the input signal, plus an external mixer, if you need more flexibility in the dry/wet balance than is offered by the mixed mode.

### D-3 Waveform Animator

[Video](#) 

**X is audio input**  
**Y is threshold**  
**Z is separation**  
**A = animated output**  
**B = square waves output**

Parameter	Min	Max	Default	Description
0	-1	31	11	LFO depth.
1	-31	31	0	Y offset.
2	0	31	23	LFO rate.
3	-1	31	-1	Scale.

This algorithm recreates a popular analogue circuit variously known as a waveform animator or wave multiplier. Such a circuit, when given (typically) a sawtooth waveform as input, uses a comparator to generate a square wave of suitable phase and pulsewidth such that when the square and saw waves are added together the result is a phase shifted saw waveform. This is usually done a number of times and the results added with the overall effect of turning the original, rather plain, waveform into a much richer one which to all intents and purposes is the same as if you'd started with a number of VCOs rather than just one, and so sounds "fatter". In combination with LFOs varying the comparator thresholds, very rich textures can be generated.

Here, four comparators and four LFOs are used.

X is the audio input.

The thresholds of the comparators are set by a combination of Y and Z. Y sets the 'centre' threshold; Z sets a 'spread' of the individual thresholds around the centre. Parameter 1 also sets the centre.

Parameter 0 sets the LFO depth. Parameter 2 scales the LFO rates; the four LFOs have preset individual speeds, which the parameter multiplies.

Parameter 3 sets the 'scale'. The recreation of phase shifted sawtooths works best when the square wave amplitude is matched to that of the saw. The default setting of -1 uses an envelope tracker to automatically set the square amplitude from the incoming audio. Other values directly set the amplitude (31 corresponds to  $\pm 8V$ ). Fixing the amplitude of the square waves and varying the input signal's amplitude is a useful effect in its own right.

Output A is the animated output (sum of the original signal and the comparator outputs). Output B is just the square waves.

## D-4 State Variable Filter

[Video](#) 

**X is audio input**  
**Y is filter frequency**  
**Z is filter type**  
**A is filter output LP->BP->HP**  
**B is filter output HP->BP->LP**

Parameter	Min	Max	Default	Description
0	-80	80	0	Y Offset.
1	0	31	0	Filter resonance.

This is a 2nd-order filter, with a voltage controllable response type. By varying Z, the output can be low pass, band pass or high pass, or blends in between.

X is the audio input.

Y sets the filter frequency, with a 1V/octave response. The 0V point is C3 (approximately 130.81Hz). Parameter 0 is added to the Y CV; its range of  $\pm 80$  corresponds to  $\pm 5V$ .

Z sets the filter type output at A & B.

Output A blends smoothly between low pass (at minimum Z), through band pass (at zero Z), to high pass (at maximum Z).

Output B blends smoothly between high pass (at minimum Z), through band pass (at zero Z), to low pass (at maximum Z).

Parameter 1 sets the filter resonance.

## D-5 LP/HP Filter

[Video](#) 

**X is audio input**  
**Y is filter frequency**  
**Z is filter resonance**  
**A is low pass filter output**  
**B is high pass filter output**

Parameter	Min	Max	Default	Description
0	-80	80	0	Y Offset.

This is a filter with simultaneous low pass and high pass outputs.

X is the audio input.

Y sets the filter frequency, with a 1V/octave response. The 0V point is C3 (approximately 130.81Hz). Parameter 0 is added to the Y CV; its range of  $\pm 80$  corresponds to  $\pm 5V$ .

Z controls the filter resonance.



## D-6 LP/BP Filter

[Video](#) 

**X is audio input**  
**Y is filter frequency**  
**Z is filter resonance**  
**A is low pass filter output**  
**B is band pass filter output**

Parameter	Min	Max	Default	Description
0	-80	80	0	Y Offset.

This is a filter with simultaneous low pass and band pass outputs.

X is the audio input.

Y sets the filter frequency, with a 1V/octave response. The 0V point is C3 (approximately 130.81Hz). Parameter 0 is added to the Y CV; its range of  $\pm 80$  corresponds to  $\pm 5V$ .

Z controls the filter resonance.

## D-7 BP/HP Filter

[Video](#) 

**X is audio input**  
**Y is filter frequency**  
**Z is filter resonance**  
**A is band pass filter output**  
**B is high pass filter output**

Parameter	Min	Max	Default	Description
0	-80	80	0	Y Offset.

This is a filter with simultaneous band pass and high pass outputs.

X is the audio input.

Y sets the filter frequency, with a 1V/octave response. The 0V point is C3 (approximately 130.81Hz). Parameter 0 is added to the Y CV; its range of  $\pm 80$  corresponds to  $\pm 5V$ .

Z controls the filter resonance.

## D-8 BP/Notch Filter

[Video](#) 

**X is audio input**  
**Y is filter frequency**  
**Z is filter resonance**  
**A is band pass filter output**  
**B is notch filter output**

Parameter	Min	Max	Default	Description
0	-80	80	0	Y Offset.

This is a filter with simultaneous band pass and notch outputs.

X is the audio input.

Y sets the filter frequency, with a 1V/octave response. The 0V point is C3 (approximately 130.81Hz). Parameter 0 is added to the Y CV; its range of  $\pm 80$  corresponds to  $\pm 5V$ .

Z controls the filter resonance.

## E-1 AR Envelope

[Video](#) 

**X is trigger input**  
**Y is trigger input**  
**Z sets the envelope times**  
**A is envelope output**  
**B is envelope output**

**Press Z to trigger**

**Receives MIDI**

Parameter	Min	Max	Default	Description
0	0	2	0	Trigger Mode.
1	0	1	0	Z Mode.
2	-40	41	32	Output A Attenuverter.
3	-40	40	32	Output B Attenuverter.
4	-40	40	0	Output A Offset.
5	-40	40	0	Output B Offset.
6	0	32	26	Attack shape.
7	0	32	10	Release shape.

This algorithm is a two-stage (attack/release or attack/decay) envelope generator.

Inputs X & Y are trigger inputs. A signal in excess of 1V on either input will trigger the envelope, according to the mode set by parameter 0. In trigger mode 0, the envelope will rise to full level and stay there as long as the input is high (AR mode). In trigger mode 1, the envelope will execute one full attack/decay cycle in response to a trigger input (AD mode). In trigger mode 2, the envelope will continually execute attack/decay cycles as long as the trigger is high (looped AD mode).

Z sets the envelope times, according to the mode set by parameter 1. In Z mode 0, the full range of Z values sweeps from short A & D, through short A & long D, through long A & D, through long A & short D, and finally back to short A & D. In Z mode 1, Z sets the A & D times to the same value, from very short times (about 10ms) to very long times (about 8s).

A & B both output the envelope CV. Each has its own attenuverter parameter, which can attenuate and/or invert the signal. The maximum envelope level is 10V, with the default parameter value of 32 giving an 8V envelope. Each output also has an offset parameter, which can move the whole envelope up or down by up to 10V.

If parameter 2 is set to the special value 41, output A becomes an end-of-cycle trigger output, generating a 10ms 5V pulse at the end of the envelope's release stage.

Parameters 6 & 7 set the attack and release shapes of the envelope respectively, from an exaggerated exponential curve at 0 to an almost linear shape at 32.

Pushing the Z knob has the same effect as triggering the envelope via the X or Y inputs.

**MIDI Input:** Any MIDI note messages received on the MIDI channel set in the Settings trigger the envelope.

## E-2 AR Envelope & VCA

[Video](#) 

**X is trigger input**  
**Y is VCA input**  
**Z sets the envelope times**  
**A is envelope output**  
**B is VCA output**

**Press Z to trigger**  
**Receives MIDI**

Parameter	Min	Max	Default	Description
0	0	8	0	Trigger Mode.
1	0	1	0	Z Mode.
2	-40	41	32	Output A Attenuverter.
3	-40	40	32	Output B Attenuverter.
4	-40	40	0	Output A Offset.
5	-40	40	0	Output B Offset.
6	0	32	26	Attack shape.
7	0	32	10	Release shape.

This algorithm is a combined two-stage (attack/release or attack/decay) envelope generator & VCA (voltage controlled amplifier).

Input X is the trigger input. A signal in excess of 1V will trigger the envelope, according to the mode set by parameter 0. In trigger mode 0, the envelope will rise to full level and stay there as long as the input is high (AR mode). In trigger mode 1, the envelope will execute one full attack/decay cycle in response to a trigger input (AD mode). In trigger mode 2, the envelope will continually execute attack/decay cycles as long as the trigger is high (looped AD mode).

Trigger modes 3-8 work like modes 1-2 except that the source of the trigger is the tracked envelope of one of the inputs. Modes 3-5 use input X and modes 6-8 use input Y. So for example mode 6 triggers the envelope generator from the VCA input signal, i.e. the algorithm is behaving as an 'envelope modifier' effect – put a guitar in here for automatic slow swells.

Input Y is the VCA input. The signal here will be multiplied by the envelope and output on output B.

Z sets the envelope times, according to the mode set by parameter 1. In Z mode 0, the full range of Z values sweeps from short A & D, through short A & long D, through long A & D, through long A & short D, and finally back to short A & D. In Z mode 1, Z sets the A & D times to the same value, from very short times (about 10ms) to very long times (about 8s).

Output A is the envelope CV, and output B is the VCA output as mentioned above. Each has its own attenuverter parameter, which can attenuate and/or invert the signal. The maximum envelope level is 10V, with the default parameter value of 32 giving an 8V envelope, which corresponds to unity gain for the VCA. Each output also has an offset parameter, which can move the whole envelope up or down by up to 10V.

If parameter 2 is set to the special value 41, output A becomes an end-of-cycle trigger output, generating a 10ms 5V pulse at the end of the envelope's release stage.

Parameters 6 & 7 set the attack and release shapes of the envelope respectively, from an exaggerated exponential curve at 0 to an almost linear shape at 32.

Pushing the Z knob has the same effect as triggering the envelope via the X input.

**MIDI Input:** Any MIDI note messages received on the MIDI channel set in the Settings trigger the envelope.

### E-3 Dual AR Envelope

[Video](#) 

**X is trigger input A**  
**Y is trigger input B**  
**Z sets the envelope times**  
**A is envelope output A**  
**B is envelope output B**

**Press Z to trigger**  
**Receives MIDI**

Parameter	Min	Max	Default	Description
0	0	2	0	Trigger Mode.
1	0	1	0	Z Mode.
2	-40	41	32	Output A Attenuverter.
3	-40	40	32	Output B Attenuverter.
4	-40	40	0	Output A Offset.
5	-40	40	0	Output B Offset.
6	0	32	26	Attack shape.
7	0	32	10	Release shape.

This algorithm offers dual two-stage (attack/release or attack/decay) envelope generators, with shared time settings.

Inputs X & Y are trigger inputs. A signal in excess of 1V on either input will trigger its respective envelope, according to the mode set by parameter 0. In trigger mode 0, the envelope will rise to full level and stay there as long as the input is high (AR mode). In trigger mode 1, the envelope will execute one full attack/decay cycle in response to a trigger input (AD mode). In trigger mode 2, the envelope will continually execute attack/decay cycles as long as the trigger is high (looped AD mode).

Z sets the envelope times, according to the mode set by parameter 1. In Z mode 0, the full range of Z values sweeps from short A & D, through short A & long D, through long A & D, through long A & short D, and finally back to short A & D. In Z mode 1, Z sets the A & D times to the same value, from very short times (about 10ms) to very long times (about 8s).

A & B are the envelope CV outputs. Each has its own attenuverter parameter, which can attenuate and/or invert the signal. The maximum envelope level is 10V, with the default parameter value of 32 giving an 8V envelope. Each output also has an offset parameter, which can move the whole envelope up or down by up to 10V.

Parameters 6 & 7 set the attack and release shapes of the envelope respectively, from an exaggerated exponential curve at 0 to an almost linear shape at 32.

Pushing the Z knob has the same effect as triggering both envelopes via the X & Y inputs.

**MIDI Input:** Any MIDI note messages received on the MIDI channel set in the Settings trigger envelope A; any MIDI note messages on the next channel up trigger envelope B.

## E-4 Stereo Compressor

[Video](#) 

**X is left input**  
**Y is right input**  
**A is left output**  
**B is right output**  
**Z is compression ratio**

Parameter	Min	Max	Default	Description
0	-36	0	-12	Threshold.
1	0	99	20	Attack time.
2	0	99	20	Release time.
3	-1	36	0	Make-up gain.
4	0	99	0	Lookahead.

This algorithm implements a fairly traditional stereo compressor.

X & Y are the stereo audio inputs; A & B are the stereo audio outputs.

The Z knob/CV sets the compression ratio, from 1 (no compression) at fully counterclockwise, to infinity (effectively limiting rather than compression) at fully clockwise.

Parameter 0 sets the threshold level above which compression applies (in dB).

Parameters 1 & 2 set the attack and release times. The attack time range is approximately 0.1ms-1s; the release time range is approximately 1ms-3s.

Parameter 3 sets the make-up gain (in dB). If set to -1, the make-up gain is calculated automatically from the threshold and compression ratio.

Parameter 4 sets a lookahead time, effectively delaying the audio slightly relative to the envelope detection. This can be used to ensure that sharp transients are not lost during the attack time of the envelope. The maximum lookahead is about 10ms.

## E-5 Side-chain Compressor

[Video](#) 

**X is left input**  
**Y is right input**  
**A is left output**  
**B is right output**  
**Z is side-chain input**

Parameter	Min	Max	Default	Description
0	-36	0	-12	Threshold.
1	0	99	20	Attack time.
2	0	99	20	Release time.
3	-1	36	0	Make-up gain.
4	0	99	0	Lookahead.
5	0	99	50	Compression ratio.

This algorithm is much the same as E-4 Stereo Compressor above, except that the gain reduction is driven not by the audio signal itself, but by a side-chain input. Refer to the description of E-4 for explanation of the common parameters not detailed below.

The Z knob/CV is the side-chain input. If you're using an audio signal here, you'll want to set the knob to its central (un-illuminated) position first. However, you can also use Z with any CV to control the gain reduction directly.

Parameter 5 sets the compression ratio, with a range from 1 (no compression) to infinity (effectively limiting rather than compression).

## E-6 Mono Compressor

[Video](#) 

**X is audio input**  
**Y is side-chain input**  
**A is audio output**  
**B is gain reduction output**  
**Z is compression ratio**

Parameter	Min	Max	Default	Description
0	-36	0	-12	Threshold.
1	0	99	20	Attack time.
2	0	99	20	Release time.
3	-1	36	0	Make-up gain.
4	0	99	0	Lookahead.
5	-80	12	-80	Side-chain mix.

This algorithm implements a mono compressor with side-chain input.

X is the audio input; A is the audio output.

Y is the side-chain input, which drives the compression amount applied to the audio signal on X.

The B output provides a CV reflecting the amount of gain reduction being applied.

The Z knob/CV sets the compression ratio, from 1 (no compression) at fully counterclockwise, to infinity (effectively limiting rather than compression) at fully clockwise.

Parameter 0 sets the threshold level above which compression applies (in dB).

Parameters 1 & 2 set the attack and release times. The attack time range is approximately 0.1ms-1s; the release time range is approximately 1ms-3s.

Parameter 3 sets the make-up gain (in dB). If set to -1, the make-up gain is calculated automatically from the threshold and compression ratio.

Parameter 4 sets a lookahead time, effectively delaying the audio slightly relative to the envelope detection. This can be used to ensure that sharp transients are not lost during the attack time of the envelope. The maximum lookahead is about 10ms.

Parameter 5 allows the side-chain input to be mixed into the audio output. The parameter sets the gain in multiples of 0.5dB (i.e. the maximum is +6dB and the minimum is -40dB, which is treated as  $-\infty$ dB).

## E-7 Euro to Buchla Converter

[Video](#) 

**X is 1V/octave input**  
**Y is gate input**  
**Z is tune  $\pm 0.5$  octaves**  
**A is 1.2V/octave output**  
**B is gate/trigger output**

Parameter	Min	Max	Default	Description
0	-8	8	0	Octave shift.
1	-99	99	99	Coarse.
2	-99	99	10	Fine.
3	-99	99	0	Offset.
4	0	99	49	Gate level.

This algorithm was designed for interfacing Eurorack (or more generally, any analogue synths using the 1V/octave standard) with Buchla synths, though it has options to cater for other, non-standard formats.

Input X/Output A convert a pitch CV from the 1V/octave standard to the 1.2V/octave standard by default. The exact scaling is set by parameters 1 & 2:

$$\text{scale} = 1.0 + 0.002 * (\text{parameter } 1) + 0.0002 * (\text{parameters } 2)$$

Input Y is a gate input, triggering when the level exceeds 1V. From this, a Buchla-format combined gate & trigger is generated from output B. This is a 4ms pulse at 10V, followed by a sustained gate at 5V.

The Z knob/CV provides a tuning control, with a range of approximately  $\pm 0.5$  octaves.

Parameter 0 shifts the X input by whole octaves, equivalent to adding a whole number of Volts to the X CV.

Parameter 3 applies an offset in mV to the A output.

Parameter 4 sets the gate level as  $(1 + \text{the parameter value})\%$  of the 10V trigger pulse. The default

value of 49 therefore gives a 50% gate level i.e. 5V.

## E-8 Buchla to Euro Converter

**X is 1.2V/octave input**  
**Y is gate/trigger input**  
**Z is tune  $\pm 0.5$  octaves**  
**A is 1V/octave output**  
**B is trigger output**

Parameter	Min	Max	Default	Description
0	-8	8	0	Octave shift.

No video yet - we don't own a Buchla! Volunteers please get in touch.

This algorithm is for interfacing Buchla synths with Eurorack (or more generally, any analogue synths using the 1V/octave standard).

Input X/Output A convert a pitch CV from the 1.2V/octave standard to the 1V/octave standard.

Input Y is intended to receive a Buchla-format combined trigger & gate signal (typically a 10V trigger pulse followed by a sustained 5V gate). From this, output B generates just the trigger pulse (at 5V). If you need the gate signal, you can directly use the Buchla output.

The Z knob/CV provides a tuning control, with a range of approximately  $\pm 0.5$  octaves.

## F-1 Clockable AD Envelope (with mute)

[Video](#) 

**X is clock input**  
**Y is mute input**  
**Z sets the envelope shape**  
**A is envelope output**  
**B is envelope output**

**Tap tempo enabled**

Parameter	Min	Max	Default	Description
0	-15	8	0	Delay time multiplier.
1	-40	41	32	Output A Attenuverter.
2	-40	40	32	Output B Attenuverter.
3	0	32	26	Attack shape.
4	0	32	10	Decay shape.

This algorithm is a two-stage (attack/decay) envelope generator, whose overall time is set from an input clock pulse. The envelope is continuously output (starting on each new clock pulse) unless muted.

Input X is the clock input. Any clock pulse in excess of 1V can be used. The time between rising trigger edges is used to set the envelope time.

Parameter 0 applies a multiplier to the envelope time, according to the table above.

Input Y is a mute input. While this input is over 1V, the output is forced to 0V.

Z sets the envelope shape, from short attack & long decay, to long attack and short decay.



Parameters 3 & 4 set the attack and decay shapes of the envelope respectively, from an exaggerated exponential curve at 0 to an almost linear shape at 32.

A & B both output the envelope CV. Parameters 1 & 2 are attenuverters, which can attenuate and/or invert the signal for each output independently. The unattenuated envelope level is 8V.

If parameter 1 is set to the special value 41, output A becomes an end-of-cycle trigger output, generating a 10ms 5V pulse at the end of the envelope's release stage.

## **F-2 Clockable AD Envelope (with gate)**

[Video](#) 

**X is clock input**  
**Y is gate input**  
**Z sets the envelope shape**  
**A is envelope output**  
**B is envelope output**

**Tap tempo enabled**

Parameter	Min	Max	Default	Description
0	-15	8	0	Delay time multiplier.
1	-40	41	32	Output A Attenuverter.
2	-40	40	32	Output B Attenuverter.
3	0	32	26	Attack shape.
4	0	32	10	Decay shape.

This algorithm is a two-stage (attack/decay) envelope generator, whose overall time is set from an input clock pulse. The envelope is continuously output (starting on each new clock pulse) while enabled by the gate.

Input X is the clock input. Any clock pulse in excess of 1V can be used. The time between rising trigger edges is used to set the envelope time.

Parameter 0 applies a multiplier to the envelope time, according to the table above.

Input Y is a gate input. While this input is below 1V, the output is forced to 0V.

Z sets the envelope shape, from short attack & long decay, to long attack and short decay.

Parameters 3 & 4 set the attack and decay shapes of the envelope respectively, from an exaggerated exponential curve at 0 to an almost linear shape at 32.

A & B both output the envelope CV. Parameters 1 & 2 are attenuverters, which can attenuate and/or invert the signal for each output independently. The unattenuated envelope level is 8V.

If parameter 1 is set to the special value 41, output A becomes an end-of-cycle trigger output, generating a 10ms 5V pulse at the end of the envelope's release stage.

### F-3 Clockable AD Envelope (with trigger)

[Video](#) 

**X is clock input**  
**Y is trigger input**  
**Z sets the envelope shape**  
**A is envelope output**  
**B is envelope output**

**Tap tempo enabled**

Parameter	Min	Max	Default	Description
0	-15	8	0	Delay time multiplier.
1	-40	41	32	Output A Attenuverter.
2	-40	40	32	Output B Attenuverter.
3	0	32	26	Attack shape.
4	0	32	10	Decay shape.

This algorithm is a two-stage (attack/decay) envelope generator, whose overall time is set from an input clock pulse. The envelope is triggered by a separate trigger pulse.

Input X is the clock input. Any clock pulse in excess of 1V can be used. The time between rising trigger edges is used to set the envelope time.

Parameter 0 applies a multiplier to the envelope time, according to the table above.

Input Y is a trigger input. Any clock pulse in excess of 1V can be used. A rising clock pulse triggers the envelope.

Z sets the envelope shape, from short attack & long decay, to long attack and short decay.

Parameters 3 & 4 set the attack and decay shapes of the envelope respectively, from an exaggerated exponential curve at 0 to an almost linear shape at 32.

A & B both output the envelope CV. Parameters 1 & 2 are attenuverters, which can attenuate and/or invert the signal for each output independently. The unattenuated envelope level is 8V.

If parameter 1 is set to the special value 41, output A becomes an end-of-cycle trigger output, generating a 10ms 5V pulse at the end of the envelope's release stage.

## F-4 Clockable AD Envelope & VCA

[Video](#) 

**X is clock input**  
**Y is VCA input**  
**Z sets the envelope shape**  
**A is envelope output**  
**B is VCA output**

**Tap tempo enabled**

Parameter	Min	Max	Default	Description
0	-15	8	0	Delay time multiplier.
1	-40	41	32	Output A Attenuverter.
2	-40	40	32	Output B Attenuverter.
3	0	32	26	Attack shape.
4	0	32	10	Decay shape.

This algorithm is a two-stage (attack/decay) envelope generator, whose overall time is set from an input clock pulse. The envelope is continuously output (starting on each new clock pulse). This algorithm also offers a VCA (voltage controlled amplifier) function.

Input X is the clock input. Any clock pulse in excess of 1V can be used. The time between rising trigger edges is used to set the envelope time.

Parameter 0 applies a multiplier to the envelope time, according to the table above.

Input Y is the VCA input. The signal here will be multiplied by the envelope and output on output B.

Z sets the envelope shape, from short attack & long decay, to long attack and short decay.

Parameters 3 & 4 set the attack and decay shapes of the envelope respectively, from an exaggerated exponential curve at 0 to an almost linear shape at 32.

Output A is the envelope CV, and output B is the VCA output as mentioned above. Parameters 1 & 2 are attenuverters, which can attenuate and/or invert the signal for each output independently. The unattenuated envelope level is 8V; output B is unity gain with respect to input Y if unattenuated.

If parameter 1 is set to the special value 41, output A becomes an end-of-cycle trigger output, generating a 10ms 5V pulse at the end of the envelope's release stage.

## F-5 Shift Register Random CVs

## Video

**X is clock input**  
**Y is modify input**  
**Z sets the randomness**  
**A is unipolar output**  
**B is bipolar output**

**Receives MIDI**

Parameter	Min	Max	Default	Description
0	0	1	0	Direction.
1	1	32	8	Length.
2	0	31	0	Slew rate.
3	-40	40	40	Output attenuverter.
4	-40	40	0	Offset.
5	0	15	0	MIDI clock divisor.

This algorithm generates random CVs via the popular rotating shift register method. The joy of this method is that it generates a loop of CVs, with a controllable likelihood of change, including the possibility to lock the loop so it does not change.

X is the clock input. Any clock pulse in excess of 1V can be used. On each rising edge the shift register rotates and a new CV is output. On each rotation, there is the possibility that one bit of the shift register will be flipped, changing the pattern. The likelihood of a flip is set by Z. When Z is zero there is a 50% chance that the bit will flip, which is the most random setting. As Z rises, the chance of a flip reduces, until at around 2V the chance of a flip hits zero and the pattern is effectively locked. Conversely, as Z goes negative, the chance of a flip goes up, reaching 100% at around -2V. This also effectively locks the pattern, but at twice the length (since it is alternating between the locked pattern and its inverse). When Z crosses the  $\pm 2V$  boundaries, in either direction, the display shows "\*\*\*" for a short while to let you know that the pattern has been locked or unlocked.

Input Y allows for modification of the sequence, even when the loop is locked. If input Y is above 1V, the bit will always be flipped on a clock pulse, regardless of the setting of Z.

Output A is the random pattern interpreted as a unipolar CV i.e. it is always a positive voltage. Output B is the random pattern interpreted as a bipolar CV i.e. it can swing both positive and negative.

Parameter 0 sets the direction of rotation. The two directions have a different sound to the patterns they tend to generate.

Parameter 1 sets the length of the shift register, and so the length of the repeating CV pattern in terms of clocks.

Parameter 2 sets the output slew rate. This has the same effect as the Slew Rate Limiter algorithm being applied to the outputs.

Parameter 3 is an attenuverter for both outputs.

Parameter 4 is an offset for both outputs.

**MIDI Input:** if parameter 5 is non-zero, incoming MIDI clock advances the shift register, according to the following table:

Parameter value	Clock divisor	Parameter value	Clock divisor
1	1/32	9	1/4
2	1/16 triplet	10	1/2 triplet
3	1/16	11	3/8
4	1/8 triplet	12	1/2
5	3/32	13	1/1 triplet
6	1/8	14	3/4
7	1/4 triplet	15	1/1
8	3/16		

## F-6 Shift Register Random Quantized CVs

[Video](#) 

**X is clock input**  
**Y is modify input**  
**Z sets the randomness**  
**A is quantized CV output**  
**B is trigger output**

### Outputs & Receives MIDI

Parameter	Min	Max	Default	Description
0	0	1	0	Direction.
1	1	32	8	Length.
2	0	15	0	Scale.
3	-40	40	32	Output attenuverter.
4	0	1	0	MIDI Gate.
5	-48	48	0	Transpose.
6	0	31	0	Slew.
7	0	15	0	MIDI clock divisor.

This algorithm generates random CVs via the popular rotating shift register method. The joy of this method is that it generates a loop of CVs, with a controllable likelihood of change, including the possibility to lock the loop so it does not change. The CVs are quantized to semitones or to a chosen musical scale.

X is the clock input. Any clock pulse in excess of 1V can be used. On each rising edge the shift register rotates and a new CV is output. On each rotation, there is the possibility that one bit of the shift register will be flipped, changing the pattern. The likelihood of a flip is set by Z. When Z is zero there is a 50% chance that the bit will flip, which is the most random setting. As Z rises, the chance of a flip reduces, until at around 2V the chance of a flip hits zero and the pattern is effectively locked. Conversely, as Z goes negative, the chance of a flip goes up, reaching 100% at around -2V. This also effectively locks the pattern, but at twice the length (since it is alternating between the locked pattern and its inverse). When Z crosses the  $\pm 2V$  boundaries, in either direction, the display shows "\*\*\*" for a short while to let you know that the pattern has been locked or unlocked.

Input Y allows for modification of the sequence, even when the loop is locked. If input Y is above

1V, the bit will always be flipped on a clock pulse, regardless of the setting of Z.

Output A is the random pattern of CVs, quantized to the scale chosen via parameter 2. The list of scales is the same as that for the Quantizer algorithm, above.

Output B is a trigger output.

Parameter 0 sets the direction of rotation. The two directions have a different sound to the patterns they tend to generate.

Parameter 1 sets the length of the shift register, and so the length of the repeating CV pattern in terms of clocks.

Parameter 3 is an attenuverter for the random CV, applied before quantization.

Parameter 5 is a transpose amount, in semitones, applied after quantization.

Parameter 6 sets the output slew rate. This has the same effect as the Slew Rate Limiter algorithm being applied to output A.

**MIDI Output:** each new note (i.e. when the Output B trigger fires) is output as a MIDI note message on the MIDI channel selected in the settings.

**MIDI Input:** if parameter 7 is non-zero, incoming MIDI clock advances the shift register, according to the table above.

The MIDI Gate parameter works in the same way as for the Quantizer, above.

## F-7 Shift Register Random Triggers

[Video](#) 

**X is clock input**  
**Y is modify input**  
**Z sets the randomness**  
**A is trigger output**  
**B is inverse trigger output**

**Press Z to modify sequence**  
**Receives MIDI**

Parameter	Min	Max	Default	Description
0	1	32	8	Length.
1	0	31	0	Pulse A.
2	0	31	0	Pulse B.
3	0	15	0	MIDI clock divisor.

This algorithm generates random triggers via the popular rotating shift register method. The joy of this method is that it generates a loop of triggers, with a controllable likelihood of change, including the possibility to lock the loop so it does not change.

X is the clock input. Any clock pulse in excess of 1V can be used. On each rising edge the shift register rotates and a new CV is output. On each rotation, there is the possibility that one bit of the shift register will be flipped, changing the pattern. The likelihood of a flip is set by Z. When Z is zero there is a 50% chance that the bit will flip, which is the most random setting. As Z rises, the chance of a flip reduces, until at around 2V the chance of a flip hits zero and the pattern is effectively locked. Conversely, as Z goes negative, the chance of a flip goes up, reaching 100% at

around -2V. This also effectively locks the pattern, but at twice the length (since it is alternating between the locked pattern and its inverse). When Z crosses the  $\pm 2V$  boundaries, in either direction, the display shows "\*\*\*" for a short while to let you know that the pattern has been locked or unlocked.

Input Y allows for modification of the sequence, even when the loop is locked. If input Y is above 1V, the bit will always be flipped on a clock pulse, regardless of the setting of Z.

Output A is the random pattern of triggers. A trigger is emitted for every bit set in the shift register.

Output B is the inverse of A - a trigger is generated for every bit not set in the shift register.

Parameter 0 sets the length of the shift register, and so the length of the repeating trigger pattern in terms of clocks.

Parameters 1 & 2 set the length of the output pulses. At zero, the pulse is a fixed length of 10ms. Values of 1-31 set the pulse length to a fraction of the clock time.

**MIDI Input:** if parameter 3 is non-zero, incoming MIDI clock advances the shift register, according to the table above.

## F-8 Shift Register Random Dual Triggers

[Video](#) 

**X is clock input**  
**Y is modify input**  
**Z sets the randomness**  
**A is trigger output A**  
**B is trigger output B**

**Press Z to modify sequence**

**Receives MIDI**

Parameter	Min	Max	Default	Description
0	1	32	8	Length A.
1	1	32	8	Length B.
2	0	31	0	Pulse A.
3	0	31	0	Pulse B.
4	0	15	0	MIDI clock divisor.

This algorithm generates random triggers via the popular rotating shift register method. The joy of this method is that it generates a loop of triggers, with a controllable likelihood of change, including the possibility to lock the loop so it does not change. In this algorithm there are two shift registers for the two outputs, which share a common clock.

X is the clock input. Any clock pulse in excess of 1V can be used. On each rising edge the shift register rotates and a new CV is output. On each rotation, there is the possibility that one bit of the shift register will be flipped, changing the pattern. The likelihood of a flip is set by Z. When Z is zero there is a 50% chance that the bit will flip, which is the most random setting. As Z rises, the chance of a flip reduces, until at around 2V the chance of a flip hits zero and the pattern is effectively locked. Conversely, as Z goes negative, the chance of a flip goes up, reaching 100% at around -2V. This also effectively locks the pattern, but at twice the length (since it is alternating between the locked pattern and its inverse). When Z crosses the  $\pm 2V$  boundaries, in either direction, the display shows "\*\*\*" for a short while to let you know that the pattern has been locked or

unlocked.

Input Y allows for modification of the sequence, even when the loop is locked. If input Y is above 1V, the bit will always be flipped on a clock pulse, regardless of the setting of Z.

Output A & B are the random patterns of triggers. A trigger is emitted on each output for every bit set in that output's shift register.

Parameters 0 & 1 set the length of the shift registers, and so the length of the repeating trigger patterns in terms of clocks.

Parameters 2 & 3 set the length of the output pulses. At zero, the pulse is a fixed length of 10ms. Values of 1-31 set the pulse length to a fraction of the clock time.

**MIDI Input:** if parameter 4 is non-zero, incoming MIDI clock advances the shift register, according to the table above.

## **G-1 ES-1 Emulation**

[Video](#) 

**X is input 1**  
**Y is input 2**  
**Z is trim**  
**A is output 1**  
**B is output 2**

This algorithm provides a software implementation of the Expert Sleepers [ES-1](#) module. This in conjunction with the [Silent Way AC Encoder](#) plug-in this allows you to pass CVs from your DAW to your modular via an AC coupled audio interface.

X & Y are the two inputs - connect these to outputs from your audio interface.

A & B are the corresponding outputs - connect these to CV inputs in your modular.

Z provides a trim control. Adjust this so that a zero CV entering the AC Encoder plug-in gives you a zero CV out of the disting (either with a voltmeter, or by eye, just looking at the colour of the disting's output jacks).

## **G-2 ES-2 Emulation**

[Video](#) 

**X is input 1**  
**Y is input 2**  
**Z is trim**  
**A is output 1**  
**B is output 2**

This algorithm provides a software implementation of the Expert Sleepers [ES-2](#) module. This in conjunction with the [Silent Way CV Inout](#) plug-in this allows you to pass CVs from your modular into your DAW via a regular audio interface.

X & Y are the two inputs - connect these to CV outputs from your modular.

A & B are the corresponding outputs - connect these to inputs on your audio interface.

Z provides a trim control, allowing adjustment of the ES-2's operating frequency. Start with this at zero (LEDs off) and adjust to minimise noise in the CV recovered from the plug-in.



### G-3 Pitch Reference

[Video](#) 

**X & Y are not used**  
**A is sine output**  
**B is square output**  
**Z sets the amplitude of A & B**

Parameter	Min	Max	Default	Description
0	0	11	0	Semitone.
1	-1	9	4	Octave.

Generates sine and square waves at a fixed pitch, specified by the parameters.

Z sets the amplitude of the outputs.

When the pitch is changed, the note name is scrolled across the display, first as a normal name (e.g. 'C#4') and then as a MIDI note number (e.g. 61).

### G-4 Frequency Reference

[Video](#) 

**X & Y are not used**  
**A is sine output**  
**B is square output**  
**Z sets the amplitude of A & B**

Generates sine and square waves at a fixed frequency, specified in Hz.

Z sets the amplitude of the outputs.

When the pitch is changed, the frequency is scrolled across the display (e.g. '440Hz').

This algorithm uses an interface unlike any other, in order to be able to set the digits of the frequency individually. It is somewhat like the usual parameters interface, in that pressing Z advances through the digits, and turning S changes the values.

When S is turned, a pair of digits is displayed. The first is the decimal place that is being edited in the frequency value e.g. '0' is ones, '1' is tens, '2' is hundreds and so on. The second digit is the value of that digit within the frequency value.

Pressing Z advances the decimal place, and displays a pair of digits as above representing that new part of the frequency value, which can then be edited with S.

## G-5 Tuner

[Video](#) 

**X is input**  
**Y is not used**  
**A is a copy of X**  
**B is a sine wave at the tracked pitch**  
**Z sets the amplitude of B**

Parameter	Min	Max	Default	Description
0	-99	99	0	Reference frequency.

This algorithm provides a chromatic tuner. The pitch of the input signal is tracked and displayed as a note name, with indication of whether the note is sharp or flat.

Natural notes are shown in capitals i.e. 'C', 'D', 'E' etc. Lower case letters indicate sharps e.g. 'c' indicates C sharp, 'd', indicates D sharp.

The fifth column of LEDs in the display shows the octave.

The sixth and seventh columns show how far away from the correct pitch the note is. When the note is tuned, the sixth column will be blank and the seventh will have the centre LED lit.

Output A is a copy of X, so the tuner can be left inline without needing a mult.

Output B is a sine wave at the pitch of the detected note. The amplitude of the sine wave is set by the Z knob/CV.

Parameter 0 sets the tuning reference frequency, relative to A = 440Hz. E.g. set the parameter to -8 for A = 432Hz.

## G-6 Clock

[Video](#) 

**X is clock input**  
**Y is run/stop input**  
**A & B are clock outputs**  
**Z is ratchet**  
**Z press is start/stop or tap**

**Outputs & Receives MIDI**

Parameter	Min	Max	Default	Description
0	0	5	0	Mode.
1	-2	14	6	Clock divisor for A.
2	-2	14	-2	Clock divisor for B.
3	0	11	8	Clock divisor for X.
4	0	3	0	Y mode.
5	-120	120	0	Tempo.
6	0	1	0	Z mode.
7	0	6	0	Ratchet mode.

This algorithm converts clocks between MIDI clocks, analogue clocks and an internal timebase, in

any combination. The mode of operation is determined by parameter 0.

Value	A/B driven by	MIDI driven by	Typical application
0	MIDI	X	Legacy disting behaviour.
1	X	X	Another module is the clock leader; disting converts to MIDI clock and generates clock divisions.
2	Internal	X	
3	MIDI	Internal	Disting converts MIDI clock to analogue clock.
4	X	Internal	
5	Internal	Internal	Disting is the clock leader, generating analogue clock and MIDI clock.

The internal clock is started & stopped by pushing the Z knob. There are two choices, set by parameter 6: a tempo-based clock or a tap tempo clock.

Value	Mode	Description
0	Start/Stop	Push Z to start the internal clock; push it again to stop the clock. Parameter 5 sets the tempo, as 120 plus the parameter value (i.e. the range is 0-240 bpm).
1	Tap Tempo	Push Z five times to start the clock (i.e. a one bar count-in in 4 time). Hold Z for 2 seconds to stop the clock. While the clock is running, you may continue to tap Z to correct or change the tempo. Parameter 5 offsets the tempo set by tapping (up or down in BPM steps).

While the internal clock is running, the top left pixel of the display flashes in time.

Whatever the clock source, outputs A & B generate 5V clock pulses according to parameters 1 & 2. The parameter/divisor relationship is as per the following table:

Parameter value	Output A/B	Parameter value	Output A/B
-2	Run/stop		
-1	Reset pulse		
0	1/96	8	1/4
1	1/48	9	1/3
2	1/32	10	1/2
3	1/24	11	1/1
4	1/16	12	2/1
5	1/12	13	3/1
6	1/8	14	4/1
7	1/6		

In modes that use it, input X is a clock input, with a threshold of 1V. Parameter 3 sets the interpretation of X according to the following table:

Parameter value	Input X	Parameter value	Input X
0	1/96	6	1/8
1	1/48	7	1/6
2	1/32	8	1/4
3	1/24	9	1/3
4	1/16	10	1/2
5	1/12	11	1/1

Parameter 4 controls whether input Y is used as a run/stop input:

Value	Y mode	Description
0	Not used.	The X clock is taken to start as soon as pulses appear on the X input, and is assumed to have stopped if no pulses appear for two seconds.
1	Run/stop for X.	The X clock is ignored until Y goes high (over 1V) and stops when Y goes low.
2	Run/stop for internal.	The internal clock starts when Y goes high and stops when Y goes low.
3	Run/stop for both.	Both the X clock and the internal clock are controlled by Y as above.

Parameter 7 enables "ratcheting" - a CV controllable clock multiplier. This can be enabled for outputs A, B, or both, and can multiply by powers of 2 only (1, 2, 4, 8, 16), or by multiples of 2 and 3 (1, 2, 3, 4, 6, 8, 12, 16).

Parameter value	Ratchet mode A	Ratchet mode B
0	Off	Off
1	2's	Off
2	Off	2's
3	2's	2's
4	2's/3's	Off
5	Off	2's/3's
6	2's/3's	2's/3's

When enabled, the Z input/pot controls the multiplier. When Z is at 0V, the clock output is suppressed; higher voltages give higher multipliers. If you prefer a 0V CV input to correspond to 1x multiplier instead of stopping the clock, simply turn Z slightly clockwise to bias the input positive.

## G-7 MIDI/CV

[Video](#) 

**X & Y are not used**  
**A is pitch CV output**  
**B is gate output**  
**Z is not used**

**Receives MIDI**

Parameter	Min	Max	Default	Description
0	-48	48	0	Transpose.
1	0	24	2	Pitch bend depth.
2	-1	99	-1	Scale.
3	0	32	0	Portamento.
4	0	100	0	Interrupt.
5	0	2	0	Priority.
6	0	1	0	Mode.

Performs a MIDI to CV conversion on MIDI notes arriving on the disting's MIDI port.

The received MIDI channel is selected in the settings.

Parameter 6 sets the mode of operation.

Parameter value	Mode description
0	Output A is the pitch CV output; output B is the gate output.
1	Output A is the pitch CV output; output B is another pitch CV output using the next MIDI channel up.
2	Output A is the pitch CV output; output B is a velocity gate output.

Outputs A & B can also be controlled directly via MIDI CCs according to the following table:

CC numbers (A/B)	Output voltage range
33/34	-5V to +5V
35/36	0V to 10V
37/38	0V to 5V
39/40	0V to 8V

Parameter 2 selects a scale to use, if any are provided on the SD card. See “Scala Support” above. The default value of -1 disables this function.

Parameter 3 enables a 'fingered portamento' effect, that is, a glide from one note to the next if they are played legato. Higher parameter values give a slower glide.

Parameter 4 enables an interruption of the gate output when notes are played legato. The parameter value sets the time in milliseconds to send the gate low when a new note is played.

Parameter 5 sets the note priority mode. The options are Last, Low, and High.

## G-8 CV/MIDI

[Video](#) 

**X is pitch CV input**  
**Y is gate input**  
**A & B pass through X & Y**  
**Z is mod wheel or velocity CV input**

### Outputs MIDI

Parameter	Min	Max	Default	Description
0	0	99	16	Channel offset.
1	0	127	0	Z mode.
2	0	99	0	X CC.
3	0	99	0	Y CC.

Generates MIDI notes (on the MIDI channel selected in the settings) from the pitch and gate CVs on the X & Y inputs.

Parameter 0 sets an offset between the timing of the X & Y inputs (in sample frames). It is generally a good idea to delay the pitch slightly relative to the gate, to ensure that the pitch has changed and settled before the gate triggers a new note.

Parameter 1 chooses the function of the Z knob/CV. If 0, Z sets the velocity of the generated MIDI notes. Otherwise, Z generates CC messages using the parameter value as the CC number, and the note velocity is fixed at 64.

Parameters 2 & 3, when non-zero, allow you to generate CC messages from the X & Y inputs (using the parameter value as the CC number). If Y is to be converted to a CC, then notes are no longer generated.

## H-1 Crossfade/Pan

[Video](#) 

**A = Mix of X & Y according to Z**  
**B = Inverted mix of X & Y according to Z, or LFO**

### Knob recorder enabled

Parameter	Min	Max	Default	Description
0	0	3	0	Crossfade/pan law.
1	0	2	0	Clip mode.
2	-1	99	0	Output B option/LFO speed.
3	0	32	32	Output B attenuation.

Viewed as a stereo output, A & B provide a panned version of X, plus an oppositely panned version of Y.

Viewed as mono outputs, A provides a crossfaded mix of X & Y, while B provides a mix with an inverted mix control.

In both cases Z provides the pan position or crossfade amount. When Z is turned, the crossfade position is displayed as a percentage (so 50% is an equal mix of the two sources/centre pan).

Parameter 0 sets the pan/crossfade law:

<b>0</b> Equal gain	Appropriate for crossfading phase-coherent material.
<b>1</b> Equal power	Appropriate for crossfading non-phase-coherent material.
<b>2</b> Transition	DJ-style crossfade where both sources are at full gain at the 50% position.
<b>3</b> FX Loop	Output A is unattenuated input X plus a faded amount of input Y.

Parameter 1 sets the clipping/saturation mode, for when the mix of the inputs exceeds the dynamic range of the output:

<b>0</b> Legacy	The behaviour pre-firmware 4.8, which could cause digital overflows and fairly drastic wavefolding-type sounds.
<b>1</b> Hard clip	A hard digital clip is applied.
<b>2</b> Soft clip	A soft clip/saturation is applied.

Parameter 2 controls the function of output B:

<b>-1</b> Copy	Output B follows input X.
<b>0</b> Normal	Normal crossfade/pan function.
<b>1-99</b> LFO	Output B is a $\pm 5V$ triangle wave LFO, the speed of which is set from the parameter.

Parameter 3 attenuates output B (in all modes).

## **FX Loop usage**

A combination of parameters 0 and 2 can be used to set up the disting as an effect loop controller, to integrate external effect units (e.g. pedals, rack processors) into the modular.

Use input X as the main signal input. Setting parameter 2 to -1 makes output B a copy of the input, attenuated by parameter 3, which can be sent to the effects unit.

Return the effected signal to input Y. There are then two options for how you want to mix in the effect. With parameter 0 at 0, 1, or 2, the Z knob/CV gives you a wet/dry mix control. With parameter 0 at 3, the amount of input X in the mix is constant, so Z gives you a simple fader on the effect return.

## ***H-2 Dual Sample and Hold***

## Video

**A = X when Z exceeds 1V**

**B = Y when Z exceeds 1V**

Parameter	Min	Max	Default	Description
0	0	1	0	Mode.
1	0	31	0	Offset.
2	0	64	0	Noise X.
3	0	64	0	Noise Y.
4	0	32	0	Slew A.
5	0	32	0	Slew B.

### If parameter 0 is 0 (the default):

Output A is a sample of input X, taken when the trigger input Z goes over 1V. (Sample and Hold)

### If parameter 0 is 1:

Output A follows input X while input Z is over 1V. When Z falls below 0.5V, output A is held until Z goes over 1V again. (Track and Hold)

Similarly for output B/input Y.

Pressing the Z knob triggers a sample manually.

Parameter 1 sets an offset between the timing of the X/Y & Z inputs (in sample frames). If X/Y & Z are being driven together, for example by a sequencer's pitch and gate, it is often a good idea to delay the trigger slightly to allow time for the sampled CV to settle before the sample is taken.

Parameters 2 & 3 add white noise into the signal from inputs X & Y respectively, allowing you to use the algorithm as a source of random voltages without having to patch anything into the inputs.

Parameters 4 & 5 control the amount of slew applied to outputs A & B respectively.

## H-3 Dual Quantizer (Z scale)

### Video

**A = quantized( X )**

**B = quantized( Y )**

**Z chooses scale**

### Outputs MIDI

Parameter	Min	Max	Default	Description
0	0	32	32	Input X attenuation.
1	0	32	32	Input Y attenuation.
2	-31	31	0	X transpose.
3	-31	31	0	Y transpose.

Output A is a quantized version of input X; the closest whole-semitone value to the unquantized V/octave pitch CV X. Similarly output B is a quantized version of input Y.

As well as providing a chromatic scale, this algorithm can also constrain the quantized values to a musical scale or chord. This is controlled by the Z knob/CV.



When Z changes, the scale is displayed on the LEDs. The scale's number in the following table is shown, and after a short pause, the scale's name.

Number	Scale	Displayed name
0	chromatic	Chromatic
1	major scale	Major
2	minor scale	Minor
3	major triad	Triad
4	minor triad	3b+5
5	root+5th	Fifth
6	major triad+6th	Triad+6
7	minor triad+6th	3b+5+6
8	major triad+7th	Triad+7
9	minor triad+7th	3b+5+7
10	root+5th+6th	5+6
11	root+5th+7th	5+7
12	pentatonic major	Pent
13	pentatonic minor	Minor Pent
14	natural minor scale	Nat Minor
15	harmonic minor scale	Harm Minor

For each quantizer, a parameter provides a transpose control. The transposition is applied to the input before it is forced into the chosen scale i.e. the transposition moves the notes around within the same key.

The quantized notes names are shown on the display (A left, B right). Natural notes are shown in capitals i.e. 'C', 'D', 'E' etc. Lower case letters indicate sharps e.g. 'c' indicates C sharp, 'd', indicates D sharp.

**MIDI Output:** each new note is output as a MIDI note message on the MIDI channel selected in the settings (input X/output A) or on the next highest channel (input Y/output B). E.g. if the settings specify MIDI channel 4, the notes are output on channels 4 (X/A) and 5 (Y/B).

### ***H-4 Dual Quantizer***

## Video

**A = quantized( X )**  
**B = quantized( Y )**  
**Z is trigger**

### Outputs & Receives MIDI

Parameter	Min	Max	Default	Description
0	0	32	32	Input X attenuation.
1	0	32	32	Input Y attenuation.
2	-18	18	0	X scale/trigger mode.
3	-18	18	0	Y scale/trigger mode.
4	0	1	0	MIDI Gate.

Output A is a quantized version of input X; the closest whole-semitone value to the unquantized V/octave pitch CV X. Similarly output B is a quantized version of input Y.

As well as providing a chromatic scale, this algorithm can also constrain the quantized values to a musical scale or chord. This is controlled by parameters 2 & 3.

Parameter value	Scale
0, ±1	chromatic
±2	major scale
±3	minor scale
±4	major triad
±5	minor triad
±6	root +5th
±7	major triad +6th
±8	minor triad +6th
±9	major triad +7th
±10	minor triad +7th
±11	root +5th +6th
±12	root +5th +7th
±13	pentatonic major
±14	pentatonic minor
±15	natural minor
±16	harmonic minor
±17	MIDI 1
±18	MIDI 2

Additionally, the parameter being positive or negative controls whether the quantizer works in triggered mode or not. The triggers are provided by input Z (approximately 1V is required to trigger it). If the parameter is negative, the quantizer is triggered - the input is only sampled and converted to a new quantized value when triggered by input Z. If the parameter is positive, the input is constantly sampled and a new note is output as soon as the input moves into the next semitone range.

If the scale selected is 'MIDI 1' or 'MIDI 2' (values 17 & 18) the candidate notes for quantization are set via MIDI. The algorithm listens for MIDI note on messages and uses the received notes for the

quantization pattern. While any notes are held, playing new notes adds to the pattern. When all notes are released and a new note played, the pattern is reset and formed from the new note.

In 'MIDI 1' mode the played MIDI notes set the note names, but quantization can happen in any octave (i.e. if you play a 'D' the output will always be 'D' but the octave will depend on the octave of the incoming CV).

In 'MIDI 2' mode the played MIDI notes are the only ones that can be output (so say if you play middle C and the D above, the output will always be one of those two notes, depending on whether the input CV is below or above middle C).

The quantized notes names are shown on the display (A left, B right). Natural notes are shown in capitals i.e. 'C', 'D', 'E' etc. Lower case letters indicate sharps e.g. 'c' indicates C sharp, 'd', indicates D sharp.

**MIDI Output:** each new note is output as a MIDI note message on the MIDI channel selected in the settings (input X/output A) or on the next highest channel (input Y/output B). E.g. if the settings specify MIDI channel 4, the notes are output on channels 4 (X/A) and 5 (Y/B).

The MIDI Gate parameter works in the same way as for the Quantizer, above.

## H-5 Dual Euclidean Patterns

[Video](#) 

**A is pattern 1 out**  
**B is pattern 2 out**  
**X is clock input**  
**Y is reset input**  
**Z sets the 'pulses' for pattern 2**

**Receives MIDI**

Parameter	Min	Max	Default	Description
0	1	16	16	Steps.
1	1	16	4	Pulses (pattern 1).
2	0	96	0	A Rotation.
3	0	31	0	Pulse length.
4	-1	96	-1	B Rotation.
5	0	96	0	Repeat.
6	-15	15	0	MIDI clock divisor.

This algorithm generates rhythmic patterns of output pulses known as Euclidean patterns. For a detailed description of these patterns and how they are commonly found in music around the world see e.g. [here](#)<sup>9</sup> or [here](#)<sup>10</sup>.

A pattern is described by the total number of steps (controlled by parameter 0) and the number of pulses (i.e. the number steps on which a pulse is output) (controlled by parameter 1 for output A and by Z for output B).

Parameter 2 sets a 'rotation' of the pattern. At zero rotation, the first step in the pattern will always be a pulse, and the remaining pulses distributed according to the algorithm. The rotation setting moves the first pulse by a number of steps i.e. moves the down beat.

<sup>9</sup> [https://en.wikipedia.org/wiki/Euclidean\\_rhythm](https://en.wikipedia.org/wiki/Euclidean_rhythm)

<sup>10</sup> <http://www.hisschemoller.com/blog/2011/euclidean-rhythms/>

The two patterns can use the same rotation value, or independent values. If parameter 4 is -1, pattern 2 shares the rotation value set by parameter 2. Otherwise, it uses the rotation set by parameter 4.

Parameter 3 sets the length of the output pulse. At zero, the pulse is a fixed length of 10ms. Values of 1-31 set the pulse length to a fraction of the clock time.

Parameter 5 sets the overall repeat count i.e. the number of clocks until the pattern repeats. If this is zero, the number of steps (parameter 0) is used as the repeat. If the repeat count is greater than the number of steps, the pattern is extended with silence.

Input X is the clock input, advancing the pattern by one step each time the input exceeds 1V. Input Y is a reset input, resetting the pattern to step 1.

**MIDI Input:** if parameter 6 is non-zero, the patterns are controlled by incoming MIDI clock. If the parameter is positive, the patterns simply advance according to the clock divisor table above. If the parameter is negative, the patterns are also locked in to the MIDI transport: the pattern position depends absolutely on the number of clocks since the last MIDI start message (and so is predictable and repeatable).

## H-6 Dual Delayed Pulse Generator

[Video](#) 

**A is pulse triggered by X  
B is pulse triggered by Y  
Z function depends on parameter setting**

Parameter	Min	Max	Default	Description
0	0	6	0	Z mode.
1	0	12	6	Range.
2	-1	31	-1	Delay.
3	0	31	7	Length.

Outputs A & B are pulse generators (0-5V) triggered by inputs X & Y respectively (1V required to trigger). The delay between the trigger and the output pulse, and the length of the pulse, can both be controlled.

Unless in a mode where they are set by the Z input/CV, the delay and length are set from parameters 2 & 3. Parameter 1 provides a range control, which scales both the delay and length times. When the range parameter is 0, the maximum time is 10ms. Higher settings for range progressively double the maximum time; at the maximum range setting of 12, the maximum time is therefore 40.96s.

Parameter 0 controls the function of Z:

Parameter 0 value	Z function
0	Z controls delay
1	Z controls length
2	Output override (high). Z over 1V forces both outputs high.
3	Output override (low). Z over 1V forces both outputs low.
4	Input enable. Z below 1V disables input triggers.
5	Input disable. Z above 1V disables input triggers.

## H-7 Noise

[Video](#) 

**A is noise, optionally scaled by X**  
**B is noise, optionally scaled by Y**  
**Z is blend**

Parameter	Min	Max	Default	Description
0	-1	3	-1	Type A.
1	-1	3	-1	Type B.
2	-1	31	31	Attenuation A.
3	-1	31	31	Attenuation B.

Dual output variable-colour noise generator, with two optional VCAs.

Outputs A and B are noise, with colour according to the parameter values:

Parameter value	Noise colour
-1	Blended
0	Violet
1	White
2	Pink
3	Red

If 'blended' is chosen, the noise colour can be smoothly swept from violet to red with the Z knob/CV.

Parameters 2 & 3 set the attenuation of the outputs (31 corresponds to  $\pm 8V$  output for white noise). If the parameter is set to -1, the corresponding X/Y input is used to set the output amplitude (equivalent to following the noise output with a VCA driven by X/Y). The X/Y inputs are clamped at 0V i.e. negative input voltages yield silence.

## H-8 Quantizer 2



**X is pitch CV input**  
**Y is trigger input or pitch CV input**  
**Z sets the interval between A & B**  
**A is quantized CV out**  
**B is quantized CV out**

**Receives MIDI**

Parameter	Min	Max	Default	Description
0	N/A			Pattern.
1	0	2	0	Scale.
2	-12	12	0	Key.
3	0	32	32	Input X attenuation.
4	-32	32	-32	Trigger mode / offset / Y attenuation.
5	0	32	0	Slew rate.

The imaginatively-named Quantizer 2 algorithm offers pitch quantization to musical scales, with real-time user control over the notes in the scale to quantize to, and with two CV outputs that can be offset by a musical interval (e.g. a third, a fifth, an octave etc.).

X is the pitch CV input. Parameter 3 allows you to attenuate the signal before it is quantized.

Y is either a second pitch CV input, simply added to X, or a trigger input. When it is a pitch input, it is added to X after X has been attenuated – the attenuation parameter does not affect Y.

Parameter 4 controls the function of Y. When the parameter is in the range -32 to 0, Y is a CV input, attenuated by the parameter (so -32 means no attenuation, -16 means 50%, and 0 means complete attenuation). When the parameter is 1-32 Y is a trigger input, and the parameter defines an offset (in sample frames) between the timing of the X & Y inputs. It is often a good idea to delay the trigger slightly relative to the pitch, to ensure that the pitch has changed and settled before the trigger samples a new note.

A & B are the quantized pitch CV outputs. The Z knob/CV controls an interval between the two CVs, applied to output B. The interval is specified in scale degrees, so for example '3' is a third, and '8' is an octave. 'U' is displayed for unison. Note that this is not simply a constant voltage offset between the outputs – the voltage difference changes according to the position in the scale. For example, if the interval is a third, in C major, and output A is C (0V), then output B will be E (0.333V, a difference of 0.333V); if output A is D (0.167V), output B will be F (0.417V, a difference of 0.25V).

Parameter 5 sets a slew rate for both outputs. At a value of 0 there is no slew; larger values apply increasingly more slew.

Parameter 1 chooses the scale. The available options are:

Value	Name	Notes	Example (on C)
0	Major	1, 3, 5, 6, 8, 10, 12	C D E F G A B
1	Natural Minor	1, 3, 4, 6, 8, 9, 11	C D E $\flat$ F G A $\flat$ B $\flat$
2	Harmonic Minor	1, 3, 4, 6, 8, 9, 12	C D E $\flat$ F G A $\flat$ B

Parameter 2 sets the key to quantize into. The displayed key names assume that 0V pitch CV corresponds to the note C. The parameter value does not represent semitones, but degrees around

the circle of fifths. Equivalently, the parameter indicates the number of sharps or flats:

Parameter value	Key	Parameter value	Key
-5	D $\flat$	0	C
-4	A $\flat$	1	G
-3	E $\flat$	2	D
-2	B $\flat$	3	A
-1	F	4	E
		5	B
		6	F $\sharp$

Parameter 0 defines the quantizer pattern – the selection of notes from the scale that are available for the CV to adopt. The user interface to editing the pattern is unique to this algorithm. When parameter 0 is the current parameter, turning the S knob scrolls through the degrees of the scale e.g. from 1 to 7 for a major scale. (The actual notes that these scale degrees refer to depend on the selected key and scale.) Next to the number of the scale degree is either a '+', which indicates the degree is included in the quantizer pattern, or '.', which indicates that it is not. Pressing the S knob toggles between these two states.

When the display is not showing a parameter value or name, it shows the note name of the pitch CV on output A (assuming, as mentioned above, that 0V corresponds to a C).

**MIDI Input:** MIDI note messages received on MIDI channel 1 or 2 set the quantizer pattern. The notes of the C major scale (i.e. the white keys) turn on & off the corresponding degrees of the pattern. Note that C major is always used for the MIDI input, regardless of which key the quantizer is actually working in. The octave of the MIDI notes makes no difference. How the notes control the pattern depends on the MIDI channel:

- MIDI channel 1: A note on message toggles the state of the degree in the pattern on or off.
- MIDI channel 2: The pattern is set from the notes currently being held down.

## I-1 Audio Playback

[Video](#) 

**X is retrigger CV**  
**Y is start position CV**  
**Z selects the sample**  
**A is left audio output**  
**B is right audio output**

Parameter	Min	Max	Default	Description
0	0	99	0	Folder.
1	0	99	0	Envelope time.

This algorithm continuously plays audio files, in a manner very much inspired by Tom Whitwell's [Radio Music](#) module.

The current audio file is selected with the Z knob/CV. It plays, forever, at its natural speed (see the algorithms below if you want to control the speed), looping when it gets to the end; or it plays until

the end of the file and stops, depending on the 'loop' setting.

X is a retrigger input. A trigger pulse in excess of 1V will reset the playback position. The initial playback position is set by the Y input. The voltage range 0-8V corresponds to the length of the file so e.g. 0V input (or no input) sets the initial position to the start of the file, 4V input sets the initial position to halfway through the file, etc.

When switching between files, the playback position is maintained, modulo the length of the new file. For example, if you're 5 seconds into a 10 second file:

- if you switch to a 20 second file, the position will be 5 seconds into the new file.
- if you switch to a 2 second file, the position will be 1 second into the new file.

(Though see the 'useStartOnSampleChange' playlist option above to override this behaviour.)

A & B are the left & right audio outputs respectively.

Parameter 0 chooses the sample folder. These are sorted alphabetically, or in the order specified in the playlist, if one is provided.

Parameter 1 sets the trigger mode and envelope time. If the parameter is 0, the trigger input is simply that – only a positive-going edge is used, to retrigger the sample. If the parameter is 1 or more, then the trigger input is treated like a gate: a positive-going edge retriggers the sample, but also when the trigger CV goes low, the sample playback stops. The value of the parameter specifies the length of a decay envelope that is applied to the sample, so it can stop suddenly or gradually fade out.

**MIDI Input:** MIDI note on & off messages are handled equivalently to a trigger or gate on the X input – that is, a note on retriggers the sample, and if the envelope time is non-zero, a note off will stop the playback.

## I-2 Clocked Audio Playback

[Video](#) 

**X is retrigger CV**  
**Y is clock**  
**Z selects the sample, or start position**  
**A is left audio output**  
**B is right audio output**

Parameter	Min	Max	Default	Description
0	0	99	0	Folder.
1	0	99	0	Envelope time.
2	-1	99	-1	Selects the sample and function of Z.

This algorithm plays audio files, the speed of which is set by incoming clock pulses.

The current audio file is selected with the Z knob/CV, if parameter 2 is -1, or by the parameter itself, if it is in the range 0-99.

X is a retrigger input. A trigger pulse in excess of 1V will reset the playback position to the start of the sample, if Z is being used to select the sample, or to the position set by Z (the full range of Z, approximately -5V to 5V, corresponds to the length of the sample).

Y is the clock input. The speed of the sample playback is adjusted so that the whole sample fits into the time taken for a number of clocks, where that number is set (per sample) in the playlist.



When switching between files, the playback position is maintained, as for the I-1 Audio Playback algorithm.

A & B are the left & right audio outputs respectively.

Parameters 0 & 1 are the standard sample folder and envelope time parameters, described above.

**MIDI Input:** MIDI note on & off messages are handled equivalently to a trigger or gate on the X input – that is, a note on retriggers the sample, and if the envelope time is non-zero, a note off will stop the playback.

### I-3 Audio Playback with V/Oct

[Video](#) 

**X is retrigger CV**  
**Y is V/octave pitch CV**  
**Z selects the sample**  
**A is left audio output**  
**B is right audio output**

Parameter	Min	Max	Default	Description
0	0	99	0	Folder.
1	-8	8	0	Octave shift.
2	0	99	0	Envelope time.
3	-1	4	-1	MIDI mode.
4	0	24	2	Pitch bend depth.

This algorithm is much the same as 'I-1 Audio Playback', except that it forgoes the sample start position input and instead has a pitch input.

Y is the pitch input, using the 1V/octave standard. 0V corresponds to the sample playing at its natural pitch. Positive voltages speed up playback; negative voltages slow it down.

Parameter 1 shifts the pitch up or down in octave increments.

Note that there is a limit to how fast data can be streamed from the SD card. If you speed playback up by a large factor, you may get dropouts.

Parameters 0 & 2 are the standard sample folder and envelope time parameters, described above.

**MIDI Input:** MIDI note on & off messages are handled equivalently to a trigger or gate on the X input – that is, a note on retriggers the sample, and if the envelope time is non-zero, a note off will stop the playback. Additionally, the pitch of the note on message is used instead of the Y input CV, so you can effectively play the disting like a sampler from a MIDI keyboard. MIDI note 48 (C3) corresponds to a 0V input i.e. playing the sample at its natural pitch. MIDI pitch bend is supported; parameter 4 sets the bend depth in semitones.

Parameter 3 'MIDI mode' allows for polyphonic operation, using multiple disting mk4 modules. If the parameter is -1 or 0, the module responds to MIDI like a monophonic synthesizer<sup>11</sup>. Otherwise, the parameter is a 'voice ID' in a polyphonic setup. By giving different voice IDs to multiple distings, and feeding them the same MIDI, you can create a polyphonic sample player where each disting takes a different note of a held chord.

<sup>11</sup> It is the intention that in future MIDI modes -1 and 0 might differ, for example in reusing the CV inputs that are likely not used if you're controlling the module via MIDI, but currently they are the same.

(Recall that the Select Bus provides a convenient way to send the same MIDI to multiple distings without extra cabling - see above.)

By default the polyphonic operation uses a 'lowest voice' allocation scheme – each new note is allocated to the voice with the lowest ID that is not currently playing a note. It is also possible to select a 'round robin' allocation scheme, which often works better with samples with a decay tail. This is done by sending MIDI CC #21 with a value of the number of polyphonic voices available. Sending this CC also resets the round robin to voice 1 (you may like to do this at the start of a song playback, for example, to ensure consistent results on each play). Sending CC #21 with value 0 reverts to lowest voice allocation.

## ***I-4 Audio Playback with Z Speed***

**Video** 

**X is retrigger CV**  
**Y is start position CV**  
**Z sets the playback speed**  
**A is left audio output**  
**B is right audio output**

Parameter	Min	Max	Default	Description
0	0	99	0	Folder.
1	0	99	0	Selects the sample.
2	0	99	0	Envelope time.

This algorithm is much the same as the 'I-1 Audio Playback' algorithm, except that it forgoes voltage control of the current sample and instead offers a speed control.

Z is the linear speed control. The central position corresponds to the sample playing at its natural pitch. Positive voltages speed up playback; negative voltages slow it down. Because the scale is linear it is possible to slow it down all the way to being stopped.

The current sample is selected via parameter 1.

Parameters 0 & 2 are the standard sample folder and envelope time parameters, described above.

**MIDI Input:** MIDI note on & off messages are handled equivalently to a trigger or gate on the X input – that is, a note on retriggers the sample, and if the envelope time is non-zero, a note off will stop the playback.

## I-5 Audio Playback with Reverse

[Video](#) 

**X is retrigger CV**  
**Y selects the sample**  
**Z sets the playback speed**  
**A is left audio output**  
**B is right audio output**

Parameter	Min	Max	Default	Description
0	0	99	0	Folder.
1	-49	49	0	Y offset.
2	0	99	0	Envelope time.

Similar to “I-4 Audio Playback with Z Speed” except that the playback speed is allowed to go negative i.e. the sample can play backwards. Playback speed is linear with Z, with a maximum of around 2x speed, and with 0V/center Z corresponding to a speed of zero.

Input Y selects the sample. Parameter 1 is added to input Y, so you can effectively use either or a combination of both to select the sample. Note that in this algorithm a maximum of 50 samples are accessible, not the usual 100.

The algorithm caches a reversed copy of each sample to the MicroSD card. It does this once only for each sample, when it finds a new sample or one that has changed. This may take a few seconds, during which time the message “Processing...” scrolls across the display. Do not remove the MicroSD card or turn off the disting during this time, in case the MicroSD card becomes corrupted.

Parameters 0 & 2 are the standard sample folder and envelope time parameters, described above.

**MIDI Input:** MIDI note on & off messages are handled equivalently to a trigger or gate on the X input – that is, a note on retriggers the sample, and if the envelope time is non-zero, a note off will stop the playback.

## I-6 Audio Playback with Scrub

[Video](#) 

**X is not used**  
**Y is playback position**  
**Z selects the sample**  
**A is left audio output**  
**B is right audio output**

Parameter	Min	Max	Default	Description
0	0	99	0	Folder.
1	1	99	16	Speed limit.
2	0	32	0	Slew.
3	-32	32	0	Y offset.

In this algorithm the sample playback position is directly driven from the Y input. Imagine a piece of audio tape passing over a tape head – in other algorithms, that tape is constantly moving at some speed or other as if driven by a motor, but in this algorithm you're basically dragging the tape back and forwards over the playhead yourself.

Parameter 1 sets a limit on the speed at which the audio can play. With long samples or rapidly changing Y inputs it is easy to demand very high playback speeds, and this parameter enables you to put a cap on that.

Parameter 2 sets a slew limit, effectively putting a low-pass filter on the Y input. This is useful particularly when using a manually-generated CV to remove any “twitchiness”.

Parameter 3 sets an offset on the Y input, essentially adding or subtracting a constant voltage.

The algorithm caches a reversed copy of each sample to the MicroSD card. It does this once only for each sample, when it finds a new sample or one that has changed. This may take a few seconds, during which time the message “Processing...” scrolls across the display. Do not remove the MicroSD card or turn off the disting during this time, in case the MicroSD card becomes corrupted.

Parameter 0 is the standard sample folder parameter, described above.

## I-7 Dual Audio Playback

[Video](#) 

**X is trigger A**  
**Y is trigger B**  
**Z selects the samples**  
**A is left audio output**  
**B is right audio output**

Parameter	Min	Max	Default	Description
0	0	99	0	Folder.
1	0	1	0	Selects the panning option.
2	0	99	0	Envelope time.

This algorithm simultaneously plays two audio files, each with its own trigger input. It was developed particularly with drum triggering in mind, but all the usual playback options are supported. For drum applications, the preferred playlist options are:

```
-loop=0  
-gap=0  
-fadeIn=0  
-fadeOut=0  
-retriggerOnSampleChange=0
```

The Z knob/input selects the samples, in pairs – the sample triggered by the X input is always an even-numbered sample, and the Y input triggers the next sample up.

Parameter 1 selects between two options for how the outputs are used:

- If the parameter is 0, then the sample triggered by X plays out of output A, and the sample triggered by Y plays out of output B. Stereo samples are summed to mono.
- If the parameter is 1, then mono samples are panned centrally, and stereo samples are played in stereo.

Parameters 0 & 2 are the standard sample folder and envelope time parameters, described above.

The triggers are 'velocity sensitive' – the volume of the sample depends on the voltage of the trigger pulse.

**MIDI Input:** MIDI note on & off messages are handled equivalently to a trigger or gate on the X & Y inputs – that is, a note on retriggers the sample, and if the envelope time is non-zero, a note off

will stop the playback. MIDI note 48 (C3) triggers sample A; note 49 triggers sample B.

## I-8 Dual Audio Playback with Z Speed

[Video](#) 

**X is trigger A**  
**Y is trigger B**  
**Z sets the playback speed**  
**A is left audio output**  
**B is right audio output**

Parameter	Min	Max	Default	Description
0	0	99	0	Folder.
1	0	99	0	Selects sample A.
2	0	99	1	Selects sample B.
3	0	1	0	Selects the panning option.
4	0	3	0	Select which sample(s) are affected by Z.
5	0	99	0	Envelope time.

Similar to “I-7 Dual Audio Playback” except that the sample playback speed is controlled by the Z knob/CV input.

The two samples are selected by parameters 1 & 2.

Parameter 3 selects the panning option – refer to the description in the algorithm above.

Parameter 4 selects which samples are affected by the speed control, as follows:

Parameter value	Description
0	Both samples' speeds controlled by Z.
1	Sample A speed fixed; B variable.
2	Sample B speed fixed; A variable.
3	Both samples' speeds are fixed.

Parameters 0 & 5 are the standard sample folder and envelope time parameters, described above.

The triggers are 'velocity sensitive' – the volume of the sample depends on the voltage of the trigger pulse.

**MIDI Input:** MIDI note on & off messages are handled equivalently to a trigger or gate on the X & Y inputs – that is, a note on retriggers the sample, and if the envelope time is non-zero, a note off will stop the playback. MIDI note 48 (C3) triggers sample A; note 49 triggers sample B.

## J-1 MIDI File Playback (Clocked)

[Video](#) 

**X is clock**  
**Y is retrigger CV**  
**Z selects the MIDI file**  
**A is pitch CV output**  
**B is gate output**

**Outputs MIDI**

Parameter	Min	Max	Default	Description
0	-1	31	-1	Selects the MIDI file.

Algorithm-specific playlist filename: midi-playlist-clocked.txt

This algorithm plays MIDI files, converting the MIDI into CV outputs on A & B (as detailed above).

The current MIDI file is selected with the Z knob/CV, if parameter 0 is -1, or by the parameter itself if it is in the range 0-31, in which case Z is not used.

The X input is a clock, which controls the speed of the MIDI file playback. A clock pulse should exceed 1V. The clocks correspond to quarter notes in the MIDI file.

Y is a retrigger input. A trigger pulse in excess of 1V will reset the file to the beginning.

**MIDI Output:** the MIDI file being played is transmitted unaltered on the disting's MIDI output.

## J-2 Multisample 2 Audio Playback

[Video](#) 

**X is retrigger CV**  
**Y is V/octave pitch CV**  
**Z is tune  $\pm 0.5$  octaves**  
**A is left audio output**  
**B is right audio output**

Parameter	Min	Max	Default	Description
0	0	99	0	Folder.
1	-8	8	0	Octave shift.
2	0	99	0	Envelope time.
3	-1	4	-1	MIDI mode.
4	0	24	2	Pitch bend depth.
5	0	1	1	Selects the panning option.
6	0	60	0	Gate offset.

This algorithm plays audio files, using a 1V/octave pitch input (Y) to both select the sample and control its pitch. This would typically be used for playing a multisampled instrument across a range of pitches, or for mapping individual sounds (e.g. drums) to different notes.

This is a dual version of the Multisample Audio Playback algorithm below. Two voices play at once; they can play continuously, but a more typical use would be to use the envelope time parameter to have notes decay when the gate closes – then, this being a dual algorithm, one note can

continue to decay while the next one starts.

The other main difference between this and the monophonic Multisample Audio Playback algorithm is in the behaviour of the pitch CV input. In the monophonic algorithm the pitch CV is used to choose the sample when triggered, but also continues to be 'live' during the playback, allowing bends, vibrato etc. In this dual algorithm, the pitch CV is sampled when the playback is triggered; subsequent CV change has no effect until it moves by more than a semitone, in which case a new note is triggered.

X is a retrigger input. A trigger pulse in excess of 1V will reset the playback position to the start of the sample. The trigger is 'velocity sensitive' – the volume of the sample depends on the voltage of the trigger pulse.

The Z knob/CV provides a tuning control, with a range of approximately  $\pm 0.5$  octaves. Note that this is applied after the sample has been selected using the Y CV; changing Z will not affect the sample selection.

A & B are the left & right audio outputs respectively.

Parameter 1 shifts the pitch up or down in octave increments. It is equivalent to offsetting the pitch CV input – it also affects the sample selection.

Parameters 0 & 2 are the standard sample folder and envelope time parameters, described above.

Parameter 6 offsets (delays) the retrigger/gate input relative to the pitch input. This is useful to allow pitch CVs to settle before they are sampled on the rising gate, and also to cope with modules which output both a pitch and gate but change their pitch first. In both cases, an offset allows the algorithm to choose the correct sample, rather than choosing the wrong sample and then re-pitching it to match the CV.

**MIDI Input:** MIDI note on & off messages are handled equivalently to a trigger or gate on the X input – that is, a note on retriggers the sample, and if the envelope time is non-zero, a note off will stop the playback. Additionally, the pitch of the note on message is used instead of the Y input CV, so you can effectively play the disting like a sampler from a MIDI keyboard. MIDI note 48 (C3) corresponds to a 0V input. MIDI pitch bend is supported; parameter 4 sets the bend depth in semitones.

Parameter 3 'MIDI mode' works in the same way as for algorithm I-3 – see above.

Parameter 5 sets the sample panning as in the Dual Audio Playback algorithm above. This algorithm defaults the setting to 1 so that the two voices are summed, but setting it to zero can give you some nice stereo effects where alternate notes are panned oppositely.

Please see below for information regarding Multisample playlists.

### **J-3 MIDI File Playback (Free Running)**

**X is V/octave speed CV**  
**Y is retrigger CV**  
**Z sets the playback speed**  
**A is pitch CV output**  
**B is gate output**

#### **Outputs MIDI**

<b>Parameter</b>	<b>Min</b>	<b>Max</b>	<b>Default</b>	<b>Description</b>
0	0	31	0	Selects the MIDI file.

Please see video for Clocked mode above.

Algorithm-specific playlist filename: midi-playlist-free.txt

This algorithm plays MIDI files, converting the MIDI into CV outputs on A & B (as detailed above).

The current MIDI file is selected via the parameter i.e. by turning the S knob.

The playback speed is set by a combination of the Z knob/CV and the X CV. With both at zero, the MIDI file plays at 120bpm. Z offers a linear speed control - positive voltages speed up playback; negative voltages slow it down. Because the scale is linear it is possible to slow it down all the way to being stopped. X offers an exponential (1V/octave) speed control - the speed doubles for every 1V above zero, and halves for every 1V below zero.

Y is a retrigger input. A trigger pulse in excess of 1V will reset the file to the beginning.

**MIDI Output:** the MIDI file being played is transmitted unaltered on the disting's MIDI output.

### ***J-4 Audio Playback with End CV***

[Video](#) 

**X is retrigger or end position CV**  
**Y is start position CV**  
**Z is end position or retrigger CV**  
**A is left audio output**  
**B is right audio output**

Parameter	Min	Max	Default	Description
0	0	99	0	Folder.
1	0	99	0	Selects the sample.
2	0	99	0	Envelope time.

This algorithm is much the same as the base I-1 Audio Playback algorithm, except that it forgoes voltage control of the current sample and instead offers control over the sample end point.

X and Z are the retrigger and end position controls.

- If Z is negative, X is the retrigger input, and Z controls the sample end point. So, if you require manual control over the end point, turn the Z knob counterclockwise to shorten the sample.
- If Z is positive, Z is the retrigger input, and X controls the sample end point. In this case X is scaled in the same way as input Y, so varying Y and feeding X the same CV plus a small offset will give you a short segment of the sample that moves around with the input. The trigger fed into Z should be a positive pulse over about 1V.

When Z is adjusted through the switching point between these two modes, the display will show "Xt" or "Zt" (for X trigger or Z trigger) for a short while.

The current sample is selected via parameter 1.

Parameters 0 & 2 are the standard sample folder and envelope time parameters, described above.

**MIDI Input:** MIDI note on & off messages are handled equivalently to a trigger or gate on the X input – that is, a note on retriggers the sample, and if the envelope time is non-zero, a note off will stop the playback.



## J-5 Audio Recorder

**Video** 

**X is left audio input**  
**Y is right audio input**  
**Z controls recording**  
**A is left audio output**  
**B is right audio output**

This algorithm records the incoming audio to a WAV file on the MicroSD card.

X & Y are the audio inputs. The incoming audio is echoed directly to the A & B outputs.

The Z knob/CV controls recording. When Z exceeds approximately 1V, recording starts. When it falls below 0.5V, recording stops. If Z already exceeds 1V when the algorithm is started, it must fall below 0.5V before recording can be started for the first time.

Do not remove the MicroSD card or power off the disting while recording is active, as this could corrupt the card.

The recorded WAV files are stereo, 16 bit, 46.875kHz. The files are named sequentially starting with REC00000.WAV (then REC00001.WAV, REC00002.WAV etc.). If there are already recordings on the card, the next available filename will be used; the old files will not be overwritten.

The files are placed in a folder named '!RECORD'<sup>12</sup>. If this folder does not exist, it will be created.

When recording is complete, a small text file will also be written to the card, with the same filename but with the extension '.TXT'. This will detail any dropouts that occurred while recording the file (or ideally simply 'No dropouts').

The disting's screen shows different information than the usual algorithm name or parameter values in this mode. The left two columns of LEDs display level meters for the two audio inputs. The right section of the screen either shows a rectangle (for recording stopped) or a recording animation (when recording).

## J-6 Multisample Audio Playback

---

<sup>12</sup> The thinking behind this slightly odd name is that it places it alphabetically before any folder beginning with a letter, and so it is likely to be the first folder in the list when switching to a playback algorithm – so if you want to rapidly switch from record to playback, you don't have to scroll through a list of folders.

## Video

**X is retrigger CV**  
**Y is V/octave pitch CV**  
**Z is tune  $\pm 0.5$  octaves**  
**A is left audio output**  
**B is right audio output**

Parameter	Min	Max	Default	Description
0	0	99	0	Folder.
1	-8	8	0	Octave shift.
2	0	99	0	Envelope time.
3	-1	4	-1	MIDI mode.
4	0	24	2	Pitch bend depth.
5	0	63	0	Gate offset.

This algorithm plays audio files, using a 1V/octave pitch input (Y) to both select the sample and control its pitch. This would typically be used for playing a multisampled instrument across a range of pitches, or for mapping individual sounds (e.g. drums) to different notes.

X is a retrigger input. A trigger pulse in excess of 1V will reset the playback position to the start of the sample.

The Z knob/CV provides a tuning control, with a range of approximately  $\pm 0.5$  octaves. Note that this is applied after the sample has been selected using the Y CV; changing Z will not affect the sample selection.

A & B are the left & right audio outputs respectively.

Parameter 1 shifts the pitch up or down in octave increments. It is equivalent to offsetting the pitch CV input – it also affects the sample selection.

Parameters 0 & 2 are the standard sample folder and envelope time parameters, described above.

Parameter 5 offsets (delays) the retrigger/gate input relative to the pitch input. This is useful to allow pitch CVs to settle before they are sampled on the rising gate, and also to cope with modules which output both a pitch and gate but change their pitch first. In both cases, an offset allows the algorithm to choose the correct sample, rather than choosing the wrong sample and then re-pitching it to match the CV.

**MIDI Input:** MIDI note on & off messages are handled equivalently to a trigger or gate on the X input – that is, a note on retriggers the sample, and if the envelope time is non-zero, a note off will stop the playback. Additionally, the pitch of the note on message is used instead of the Y input CV, so you can effectively play the disting like a sampler from a MIDI keyboard. MIDI note 48 (C3) corresponds to a 0V input. MIDI pitch bend is supported; parameter 4 sets the bend depth in semitones.

Parameter 3 'MIDI mode' works in the same way as for algorithm I-3 – see above.

## Multisample playlists

As with other algorithms, playlist files are optional, but you will almost certainly want to use playlists within the folders to specify the “natural” and “switch” settings that determine the mapping of samples onto pitch.

Within a folder, if the files looked like this:

Name	Kind
playlist.txt	Plain Text
thumbc3.wav	Waveform audio
thumbe4.wav	Waveform audio
thumbf2.wav	Waveform audio
thumbg3.wav	Waveform audio

then "playlist.txt" might contain

```
disting playlist v1
-loop=0
-retriggerOnSampleChange=0
thumbf2.wav
-natural=41
thumbc3.wav
-switch=44
-natural=48
thumbg3.wav
-switch=52
-natural=55
thumbe4.wav
-switch=60
-natural=64
```

In this case we've turned off looping (the files are quite percussive thumb piano samples) and turned off retriggerOnSampleChange, so a new sample is chosen only when a trigger is received.

Each sample file uses the “-natural” setting to let the algorithm know its 'natural pitch', that is, the pitch of the audio in the sample as recorded. The setting uses MIDI note numbers, where 48 is C3 (see for example [here](#)<sup>13</sup>). For example the file 'thumbf2.wav' is a recording of an F at 87.3Hz, so its natural pitch is 41.

The “-switch” setting tells the algorithm when to choose one sample over another. Again, MIDI note numbers are used. The files are arranged in order of ascending pitch, and the “-switch” setting specifies the lowest pitch that will use that file. In the example above, thumbf2.wav will be used up to MIDI note 43, then thumbc3.wav up to note 51, thumbg3.wav from 52-59, and finally thumbe4.wav for notes 60 and up.

With this in mind, the pitch CV is used as follows. The incoming CV is converted to a MIDI note number – say 49, C#3. First the sample is looked up based on the switch settings – here 49 will choose thumbc3.wav. This file has natural pitch of 48, so it will be pitched up 1 semitone to play at the desired pitch.

The default for “-switch” is one up from the previous file in the list. This makes it convenient to simply list files which are to be mapped onto neighbouring keys e.g.

```
disting playlist v1
-loop=0
-retriggerOnSampleChange=0
-fixedPitch=1.0
GM KICK BASS.wav
-switch=48
GM MED SNARE.wav
GM CLOSED HH.wav
GM OPEN HH .wav
GM PEDAL HH .wav
```

13 <https://newt.phys.unsw.edu.au/jw/notes.html>

Here five drum samples are mapped onto the five notes from 48 up. Note we've also here specified “-fixedPitch=1.0” so that the drums are always played at their natural speed.

## J-7 Mono Audio Recorder

This algorithm is identical to 'J-5 Audio Recorder' except that the files recorded are in mono, using the left (X) input, and the files are placed in a folder named '!RECORD1'.

This can be useful when intending to use the recorded audio in the disting's own playback modes, since mono samples place less stress on the SD card throughput, and can be re-pitched higher.

## J-8 Audio Playback with Crossfade

[Video](#) 

**X is gate/clock CV**  
**Y is V/octave pitch CV**  
**Z selects the sample**  
**A is left audio output**  
**B is right audio output**

Parameter	Min	Max	Default	Description
0	0	99	0	Folder.
1	-8	8	0	Octave shift.
2	0	99	50	Crossfade time.
3	0	2	1	Gate mode.

The *raison d'être* of this algorithm is creating smooth textural loops out of any source material<sup>14</sup>. It is particularly useful for material captured by the disting's own Audio Recorder modes, which won't have carefully edited loop points. It can also take a clock and adjust the loop time to fit.

In this algorithm, the playlist settings 'loop', 'fadeIn', 'fadeOut', and 'gap' are ignored.

X is a gate or clock input, depending on the parameter 3, 'Gate mode':

Parameter value	Gate mode
0	X is a gate CV, starting and stopping sample playback.
1	The sample plays continuously.
2	The sample plays continuously. X is a clock CV. The loop length is adjusted to be related to the time between clocks by a power of two e.g. 2, 4, 8 but also ½, ¼, etc. if the clocks are slower than the longest possible loop.

Note that with nothing plugged in to X, switching between modes 0 & 1 gives you a means to start and stop playback manually.

Input Y is a pitch CV input following the 1V/octave standard. Note that pitching up a crossfaded sample places great demands on the SD card throughput, and there is a limit to how high you can raise the pitch. In our tests, stereo files can only be raised by about a fourth. Mono files can be

<sup>14</sup> Expert Sleepers have long had a love affair with crossfade sample looping. Our plug-in Crossfade Loop Synth dates back to 2004 ([http://expert-sleepers.co.uk/xfadelooper\\_1\\_3.html](http://expert-sleepers.co.uk/xfadelooper_1_3.html)) and is itself inspired by the Roland S-330 hardware sampler (<http://www.vintagesynth.com/roland/s330.php>), with which we made a lot of music in the early '90s.

played at considerably higher pitches.

The Z knob/CV selects the sample. The transition from one sample to the next is a smooth crossfade. Note that if the sample has the 'retriggerOnSampleChange' option set to 0, changing Z will not change the sample – it will only change when playback is stopped and restarted (by the X CV or manually via parameter 3).

A & B are the left & right audio outputs respectively.

Parameter 0 is the standard sample folder parameter, described above.

Parameter 1 shifts the pitch up & down by octaves. (Note the comments above on how high the pitch may be raised.)

Parameter 2 sets the crossfade time, in percent of the sample time.

## K-1 Wavetable VCO

[Video](#) 

**X is V/Oct pitch input**  
**Y is wavetable input**  
**Z is tune  $\pm 0.5$  octaves**  
**A is wavetable output**  
**B is sub-octave square output or detuned output**

**Receives MIDI**

Parameter	Min	Max	Default	Description
0	0	99	0	Chooses the wavetable.
1	-16	8	0	Octave shift.
2	-32	32	0	Y offset.
3	-4	99	0	Output B option/detune.
4	-1	127	-1	Phase.

This algorithm is a wavetable VCO. See for example [here](#) for a brief overview of wavetable synthesis.

X is the pitch input for the VCO, with a 1V/octave response. The 0V point for the pitch input is C3 (approximately 130.81Hz).

The Z knob/CV provides a tuning control, with a range of approximately  $\pm 0.5$  octaves.

The Y input controls the lookup point in the wavetable, with a range of  $\pm 5V$  covering the whole table. Parameter 2 offsets the Y CV, effectively providing a means of manual wave selection.

Output A is the main output of the wavetable VCO.

Output B generates a signal according to parameters 3 and 4. If parameter 4 is -1 (the default), the output is as follows:

Parameter 3 value	Output B
-4	MIDI gate.
-3	A $\pm 8V$ square wave one octave below output A
-2	Wavetable VCO one octave below output A
-1	Wavetable VCO one octave above output A
0-99	Wavetable VCO detuned relative to output A; output A is also detuned in the opposite direction.

If parameter 4 is 0-99, output B is a copy of output A with a fixed phase offset as defined by the parameter, scaled so 0-128 corresponds to a phase of 0-360°. (So for example if you want quadrature outputs, set the phase to 32.)

**MIDI Input:** MIDI note messages received on the MIDI channel set in the Settings are converted to a CV which is added to that on input X. The 0V note is 48 (C3). MIDI notes also drive the 'MIDI gate' output option.

## Loading wavetables

All wavetable files need to be in a top-level folder on the SD card called 'wavetables'.

Within the 'wavetables' folder, wavetables can take of one two forms: a single WAV file containing all the waveforms concatenated, or a folder of WAV files, one per waveform.

By default the disting will consider all WAV files and folders of WAV files in the 'wavetables' folder as wavetables. Optionally (say if you want to reorder the tables) you can specify a playlist: the wavetables are then listed in the standard playlist format as above, but with the dedicated name "playlist-wavetable.txt". Each entry is either a WAV file (as in the other playlist formats), in which case the file is assumed to contain all the waveforms concatenated, or a folder name.

If a folder is specified in "playlist-wavetable.txt", the folder contains the individual WAV files that represent one waveform each. The algorithm does not use any 'playlist.txt' file within the folder – any files found files are used in alphabetical order.

Waveform WAVs must be in 16 bit mono format. The sample rate is unimportant, since the file is assumed to contain exactly one cycle and so can pitched arbitrarily.

When using a single concatenated WAV file, the disting needs to be know how many frames in the file make up one waveform. This is specified in the playlist via the -wavelength setting (default 600). There are no settings that apply to wavetable folders, nor to the individual files inside the folders.

To reiterate – the use of playlists is entirely optional if your wavetables are in the form of folders of individual waveform files.<sup>15</sup>

A typical SD card might look like this:

Name	^	Date Modified	Size	Kind
▶ folder stuff		Today at 09:19	123.4 MB	Folder
▼ folder wavetables		Today at 11:07	271 KB	Folder
▶ folder AKWF_0001		Today at 11:07	136 KB	Folder
▼ folder AKWF_0002		Today at 11:08	136 KB	Folder
📄 AdventureKidWaveforms.txt		28 Sep 2010 at 15:37	1 KB	Plain Text
🎵 AKWF_0101.wav		8 Sep 2003 at 21:34	1 KB	Wave...audio
🎵 AKWF_0102.wav		8 Sep 2003 at 21:34	1 KB	Wave...audio
🎵 AKWF_0103.wav		8 Sep 2003 at 21:34	1 KB	Wave...audio
🎵 AKWF_0104.wav		8 Sep 2003 at 21:34	1 KB	Wave...audio
📁 UNTITLED > 📁 wavetables > 📁 AKWF_0002 > 📄 AKWF_0101.wav				
1 of 105 selected, 63.71 GB available				

<sup>15</sup> This applies from firmware version 4.11 onwards.

An example using a concatenated wavetable file would look like this:

Name	Date Modified	Size	Kind
▶ stuff	Today at 09:19	123.4 MB	Folder
▼ wavetables	Today at 11:14	304 KB	Folder
▶ AKWF_0001	Today at 11:07	136 KB	Folder
▶ AKWF_0002	Today at 11:08	136 KB	Folder
▶ playlist-wavetable.txt	Today at 11:14	79 bytes	Plain Text
▶ PPG_wavetable_01.wav	27 Feb 2018 at 04:30	33 KB	Wave...audio

UNTITLED > wavetables > PPG\_wavetable\_01.wav

1 of 6 selected, 63.71 GB available

where "playlist-wavetable.txt" contains

```
disting playlist v1
PPG_wavetable_01.wav
-wavelength=256
AKWF_0001
AKWF_0002
```

The first entry 'PPG\_wavetable\_01.wav' is used as a single WAV file containing concatenated wave cycles, each of length 256 frames. The remaining entries are taken as folders of individual wave samples.

## K-2 Clockable Wavetable LFO

[Video](#) 

**X is clock input**  
**Y is wavetable input**  
**Z is integer multiplier/divider**  
**A & B are wavetable output**

Parameter	Min	Max	Default	Description
0	0	99	0	Chooses the wavetable.
1	-32	32	0	Y Offset.
2	-32	32	32	Output A attenuverter.
3	-32	32	-32	Output B attenuverter.
4	-15	8	0	LFO time multiplier.

This algorithm offers a wavetable oscillator intended for LFO use, though with an audio rate clock it can also be used at audio frequencies. Consider it a combination of the Wavetable VCO and the Clockable LFO.

See 'Loading wavetables' above for a description of how to load wavetables onto the SD card.

Input X is the clock input. Any clock pulse in excess of 1V can be used. The time between rising trigger edges is used to set the cycle time.

The Y input controls the lookup point in the wavetable, with a range of  $\pm 5V$  covering the whole



table. Parameter 1 offsets the Y CV, effectively providing a means of manual wave selection.

The Z knob/CV sets a scale factor which is applied to the cycle time. The scale is an integer (whole number) which either multiplies or divides the frequency, and ranges in value from 1-16.

When Z changes, the scale is shown on the display. If the value shown is negative, it is a divisor rather than a multiplier. Note that there is no value of "-1" since dividing by 1 and multiplying by 1 are the same.

Outputs A & B are the result of the wavetable lookup. Each output has its own attenuverter on parameters 2 & 3 respectively.

Parameter 4 applies a multiplier to the LFO time, according to the table in the Clockable Delay/Echo section, above.

### **K-3 Wavetable Waveshaper**

[Video](#) 

**X is signal input**  
**Y is wavetable input**  
**Z is gain**  
**A is wavetable output**  
**B is complementary wavetable output**

Parameter	Min	Max	Default	Description
0	0	99	0	Chooses the wavetable.
1	-32	32	0	Y Offset.

Applied to audio, this algorithm is a wavetable-based waveshaper effect. More generally, considering the input as a CV lookup, this algorithm provides a wavetable-based transfer function.

See 'Loading wavetables' above for a description of how to load wavetables onto the SD card.

X is the input audio or CV. The range  $\pm 5V$  spans the full waveform.

Z applies a gain to X, with a range from 1x to approximately 16x.

The Y input controls the lookup point in the wavetable, with a range of  $\pm 5V$  covering the whole table. Parameter 1 offsets the Y CV, effectively providing a means of manual wave selection.

Output A is the main output of the wavetable lookup. Output B is the same but using an inverted Y i.e. when A is using the first wave in the table, B will be using the last and *vice versa*.

### **K-4 Clockable Wavetable Envelope**

### [Video](#)

**X is clock input**  
**Y is wavetable input**  
**Z is trigger**  
**A & B are wavetable output**

Parameter	Min	Max	Default	Description
0	0	99	0	Chooses the wavetable.
1	-32	32	0	Y Offset.
2	-32	32	32	Output A attenuverter.
3	-32	32	-32	Output B attenuverter.
4	-15	8	0	Envelope time multiplier.

This algorithm is essentially a version of the Clockable Wavetable LFO, but with a trigger input instead of being free-running. As such it can be thought of as a kind of envelope generator, where the envelope shape is defined by an audio file in the wavetable.

See 'Loading wavetables' above for a description of how to load wavetables onto the SD card.

Input X is the clock input. Any clock pulse in excess of 1V can be used. The time between rising trigger edges is used to set the envelope time.

The Y input controls the lookup point in the wavetable, with a range of  $\pm 5V$  covering the whole table. Parameter 1 offsets the Y CV, effectively providing a means of manual wave selection.

The Z input is the trigger for the envelope to start.

Outputs A & B are the result of the wavetable lookup. Each output has its own attenuverter on parameters 2 & 3 respectively.

Parameter 4 applies a multiplier to the envelope time, according to the table in the Clockable Delay/Echo section, above.

## **K-5 Programmable Quantizer**

### [Video](#)

**A = quantized( X+Y )**  
**B = trigger on note change**  
**Z is slew**  
**Y = pitch or trigger**

#### **Outputs MIDI**

Parameter	Min	Max	Default	Description
0	0	99	0	Scale.
1	0	32	32	Input X attenuation.
2	-1	32	32	Input Y attenuation.
3	-31	31	0	Transpose.
4	0	31	0	Offset.
5	0	1	0	Mode.

Output A is a quantized version of the sum of inputs X & Y, according to the scale chosen with parameter 0.

The scales are defined by files on the MicroSD card – see “Scala Support”, above. The keyboard mapping file is used to define the subset of notes that the quantizer may use. For example, in the provided example files, the file 'root+fifth.kbm' can be used with a 12 note scale to make the quantizer only output the root or fifth in each octave.

Parameter 5 chooses between two fundamentally different modes of operation. When parameter 5 is 0 (the default), an incoming CV is quantised to the closest pitch value defined by the chosen scale. When the parameter is 1, the incoming CV is assumed linear 1V/octave and is used to look up a note in the scale according to the keyboard map, based on the nearest semitone value of the input.

To understand the difference between these two modes, consider a scale with 19 degrees per octave, and an input CV coming from a ramp LFO. In mode 0, the LFO will step through all 19 tones, arriving at an octave when the CV is 1V higher than it started. In mode 1, when the LFO rises by 1V, the output will have only risen by 12 degrees of the scale; it will need to rise by another half Volt or so to reach the octave. Or consider the same situation but with the input from a keyboard, starting at 0V and rising 1/12<sup>th</sup> of a Volt per semitone. In mode 0, each semitone will be quantized to the closest degree of the 19 tone scale, but you'll still only get 12 tones per octave. In mode 1, playing the keyboard up from, say, C to C will give you 12 tones of the 19 tone scale; you would need to continue up to the next G to reach the octave.

Output B is a trigger signal which fires whenever output A changes - a 5V pulse approximately 10ms long.

Parameters 1 & 2 are attenuators for inputs X & Y respectively. At 0 the input is completely attenuated; at 32 the input is unattenuated.

If parameter 2 is set to -1, input Y becomes a trigger. In this mode, input X is only sampled and converted to a new quantized value when input Y rises over approximately 1V. (In non-triggered mode, X is constantly sampled and a new note is output as soon as X moves to the next scale degree.)

Parameter 3 is a transpose control. The transposition occurs after the quantization, and is in terms of the mapped notes of the scale. For example, if you're using a seven tone scale, with all tones mapped, then a transposition of 7 raises or lowers the pitch by an octave.

Parameter 4 sets an offset between the timing of the X & Y inputs (in sample frames). When using Y as a trigger, it is often a good idea to delay it slightly relative to the pitch, to ensure that the pitch has changed and settled before the trigger samples a new note.

The Z knob/CV controls the slew rate of output A. At the minimum value of Z, changes in A are instantaneous. As Z increases, changes in A take place more slowly.

**MIDI Output:** each new note (i.e. when the Output B trigger fires) is output as a MIDI note message on the MIDI channel selected in the settings. The note value is that of the position in the keyboard mapping of the current output.

## K-6 Clockable SD Delay

[Video](#) 

**X is signal**  
**Y is clock input**  
**Z is feedback**  
**A = output according to mode**  
**B = output according to mode**

**Tap tempo enabled**

Parameter	Min	Max	Default	Description
0	-15	8	0	Delay time multiplier.
1	0	8	8	Maximum feedback.
2	0	2	0	Output mode.

This algorithm is a delay/echo effect, where the delay time is set from a clock pulse. It is basically the same as the Clockable Delay/Echo algorithm, except that whereas that algorithm uses the module's RAM for storage, this one uses the SD card, meaning that the delay time can be much, much longer – about 95 minutes. It also runs at the full sample rate, so the audio fidelity is higher.

**SD card requirements**

This algorithm is continually reading from and writing to the SD card. You will want a good quality, high-speed card. A microSDHC card like that recommended above will work well.

The algorithm creates a file on the card called 'DELAY.BIN'. This is created in a very specific way using low-level card access. Do not mess with the file, or attempt to copy it onto another SD card for another disting to use. The disting must create the file itself. You can however copy the file to your computer to process its contents, should you so wish.

It may be that the disting cannot create the file if the card is too fragmented, even if it says it has enough free space. In this case, the easiest solution is to reformat the card.

Input X is the signal input. Any audio signal can be fed in here.

Input Y is the clock input. Any clock pulse in excess of 1V can be used. The time between rising trigger edges is used to set the delay time. The minimum delay time that can be set is about 90ms.

The Z knob/CV controls the feedback, from zero to a maximum determined by parameter 1. At a parameter value of 0, the maximum feedback is exactly 100%. At higher values, the maximum feedback goes over 100%, up to 125% at a parameter value of 8.

Outputs A & B carry signals according to parameter 2:

Parameter value	Output mode
0	A is mix; B is delay only.
1	A & B are mix.
2	A & B are delay only.

'Mix' is a mix of the dry (undelayed) signal and the delay effect. The amount of delay in the mix rises in direct proportion to the amount of feedback.

'Delay only' is the delayed signal only. Use this and the input signal, plus an external mixer, if you need more flexibility in the dry/wet balance than is offered by the mixed mode.

Parameter 0 applies a multiplier to the delay time, according to the table in the Clockable Delay/Echo section, above.

This algorithm is affected by the 'SD delay max' setting – see above.

### ***K-7 Stereo Clockable SD Delay***

[Video](#) 

**X & Y are stereo audio inputs  
A & B are stereo audio outputs  
Z is feedback**

**Tap tempo enabled**

<b>Parameter</b>	<b>Min</b>	<b>Max</b>	<b>Default</b>	<b>Description</b>
0	-15	8	0	Delay time multiplier.
1	0	8	8	Maximum feedback.

This is a stereo version of the Clockable SD Delay algorithm as described above.

Being stereo, the maximum delay time is halved, to a mere 48 minutes.

Please be sure to read the 'SD card requirements' box.

This algorithm is affected by the 'SD delay max' setting – see above.

### ***K-8 Stereo Clockable SD Delay (Z clock)***

[Video](#) 

**X & Y are stereo audio inputs  
A & B are stereo audio outputs  
Z is clock input**

**Tap tempo enabled**

<b>Parameter</b>	<b>Min</b>	<b>Max</b>	<b>Default</b>	<b>Description</b>
0	-15	8	0	Delay time multiplier.
1	0	40	16	Feedback.

This is a stereo version of the Clockable SD Delay algorithm as described above.

Being stereo, the maximum delay time is halved, to a mere 48 minutes.

Please be sure to read the 'SD card requirements' box.

Since this algorithm uses Z for the clock, the delay feedback is set by parameter 1. A value of 32 corresponds to 100% feedback.

This algorithm is affected by the 'SD delay max' setting – see above.

## L-1 Stereo Reverb

[Video](#) 

**X is left input**  
**Y is right input**  
**A is left output**  
**B is right output**  
**Z is wet/dry or freeze**  
**Z press is freeze**

Parameter	Min	Max	Default	Description
0	1	32	16	Size.
1	1	32	24	Feedback.
2	0	5	1	Character.
3	-1	31	8	Low pass filter.
4	0	12	0	Gain.
5	-33	32	-33	Wet/dry.
6	0	99	0	High pass filter.

A stereo reverberation effect.

X and Y are the stereo audio inputs. An equal mix of these is used to feed the reverberator.

A and B are the stereo audio outputs. These are a mix of the inputs and the reverb output.

Z is a wet/dry control. At zero, the output is 100% dry i.e. the input signals with no reverb added. For positive Z, reverb is added while holding the dry level constant. For negative Z, the dry and reverb signals are progressively crossfaded, until eventually the output is 100% wet i.e. just the reverberation signal.

Pressing the Z knob 'freezes' the reverb – the reverb does not decay, and the input signal is not added into the reverb. Press Z again to unfreeze it.

Parameters 0 & 1 together control the reverb time.

Parameter 2 changes the reverb 'character' - it chooses between a number of options for the reverb algorithm's internal parameters leading to different sounding reverbs (some quite natural, others deliberately unnatural).

Parameter 3 applies a low pass filter to the reverb input. It does not affect the dry portion of the signal.

Parameter 4 is a gain control for the reverb output, in dB. It does not affect the dry signal.

Parameter 5 allows Z to become a CV input for the freeze feature. When the parameter is from -32 to 32, this value is used to control the wet/dry balance, and the Z input becomes a freeze trigger. The reverb freezes when Z goes above approximately 1V and unfreezes when Z falls below approximately 0.5V.

Parameter 6 applies a high pass filter to the reverb input. The range of cutoff frequencies is approximately 3Hz to 1.6kHz.

## L-2 Mono-to-Stereo Reverb

[Video](#) 

**X is audio input**  
**Y is feedback CV**  
**A is left output**  
**B is right output**  
**Z is wet/dry or freeze**  
**Z press is freeze**

Parameter	Min	Max	Default	Description
0	1	32	16	Size.
1	1	32	24	Feedback.
2	0	5	1	Character.
3	-1	31	8	Low pass filter.
4	0	12	0	Gain.
5	-33	32	-33	Wet/dry.
6	0	99	0	High pass filter.

A mono-to-stereo reverberation effect.

X is the audio input.

A and B are the stereo audio outputs. These are a mix of the input and the reverb output.

Z is a wet/dry control. At zero, the output is 100% dry i.e. the input signal with no reverb added. For positive Z, reverb is added while holding the dry level constant. For negative Z, the dry and reverb signals are progressively crossfaded, until eventually the output is 100% wet i.e. just the reverberation signal.

Pressing the Z knob 'freezes' the reverb – the reverb does not decay, and the input signal is not added into the reverb. Press Z again to unfreeze it.

Parameters 0 & 1 together control the reverb time. Input Y also affects the feedback.

Parameter 2 changes the reverb 'character' - it chooses between a number of options for the reverb algorithm's internal parameters leading to different sounding reverbs (some quite natural, others deliberately unnatural).

Parameter 3 applies a low pass filter to the reverb input. It does not affect the dry portion of the signal.

Parameter 4 is a gain control for the reverb output, in dB. It does not affect the dry signal.

Parameter 5 allows Z to become a CV input for the freeze feature. When the parameter is from -32 to 32, this value is used to control the wet/dry balance, and the Z input becomes a freeze trigger. The reverb freezes when Z goes above approximately 1V and unfreezes when Z falls below approximately 0.5V.

Parameter 6 applies a high pass filter to the reverb input. The range of cutoff frequencies is approximately 3Hz to 1.6kHz.

## L-3 Dual Reverb

[Video](#) 

**A is X plus reverb**  
**B is Y plus reverb**  
**Z is wet/dry or freeze**  
**Z press is freeze**

Parameter	Min	Max	Default	Description
0	1	32	16	Size.
1	1	32	24	Feedback.
2	0	5	1	Character.
3	-1	31	8	Low pass filter.
4	0	12	0	Gain.
5	-33	32	-33	Wet/dry.
6	0	99	0	High pass filter.

A dual mono reverberation effect.

X and Y are the audio inputs.

A and B are the audio outputs. Each is a mix of its input and corresponding reverb output.

Z is a wet/dry control. At zero, the outputs are 100% dry i.e. the input signal with no reverb added. For positive Z, reverb is added while holding the dry level constant. For negative Z, the dry and reverb signals are progressively crossfaded, until eventually the outputs are 100% wet i.e. just the reverberation signals.

Pressing the Z knob 'freezes' the reverbs – the reverbs do not decay, and the input signals are not added into the reverbs. Press Z again to unfreeze them.

Parameters 0 & 1 together control the reverb time.

Parameter 2 changes the reverb 'character' - it chooses between a number of options for the reverb algorithm's internal parameters leading to different sounding reverbs (some quite natural, others deliberately unnatural).

Parameter 3 applies a low pass filter to the reverb inputs. It does not affect the dry portion of the signals.

Parameter 4 is a gain control for the reverb output, in dB. It does not affect the dry signal.

Parameter 5 allows Z to become a CV input for the freeze feature. When the parameter is from -32 to 32, this value is used to control the wet/dry balance, and the Z input becomes a freeze trigger. The reverbs freeze when Z goes above approximately 1V and unfreeze when Z falls below approximately 0.5V.

Parameter 6 applies a high pass filter to the reverb input. The range of cutoff frequencies is approximately 3Hz to 1.6kHz.



## L-4 Dual Vowel Filter

[Video](#) 

**X & Y are inputs**  
**A & B are outputs**  
**Z is vowel selection**

Parameter	Min	Max	Default	Description
0	-1	8	0	Vowel A.
1	-1	8	-1	Vowel B.
2	-20	0	0	BPF gain 2.
3	-20	0	-6	BPF gain 3.

A dual vowel filter effect. A set of three bandpass filters (BPFs) is used to mimic the response of the human vocal tract, resulting in vowel-like sounds when provided suitable source material.

The algorithm can be used as a dual-mono effect, or as stereo (in which case you'd want to select the same vowels for each channel).

Parameters 0 & 1 select the vowel sound for X/A and Y/B respectively. The available sounds are as follows:

Parameter value	Vowel	Example word
-1	Chosen from the range below by the Z knob/CV.	
0	/ow/	bought
1	/oo/	boot
2	/a/	hot <sup>16</sup>
3	/uh/	but
4	/er/	bird
5	/ae/	bat
6	/e/	bet
7	/i/	bit
8	/iy/	beet

For a good vowel effect, the input audio should be rich in harmonics. A simple sawtooth works reasonably well. However, you can of course put any audio you want through the filters.

Parameters 2 & 3 allow fine tuning of the gains of the second and third BPFs, which can be used to tweak the vowel sounds if they're not quite right for your usage.

---

<sup>16</sup> American pronunciation!

## L-5 Stereo Chorus

[Video](#) 

**X is audio input**  
**Y is LFO rate**  
**A is left output**  
**B is right output**  
**Z is wet/dry**

Parameter	Min	Max	Default	Description
0	0	99	8	LFO depth.
1	-64	32	0	Y offset.
2	-63	63	0	Feedback.
3	-1	31	-1	Lowpass filter.
4	1	64	16	Delay time.
5	-32	32	0	Fine delay time.
6	1	6	6	Stages.
7	0	1	0	Saturation.

A stereo chorus/flange effect.

X is the audio input.

A and B are the stereo audio outputs. These are a mix of the input and the chorus output.

Z is a wet/dry control. At zero, the outputs are 100% dry i.e. the input signal with no chorus added. For positive Z, chorus is added while holding the dry level constant. For negative Z, the dry and chorus signals are progressively crossfaded, until eventually the outputs are 100% wet i.e. just the chorus signals.

The chorus effect is generated via a number of LFOs. The LFO depth is set with parameter 0. The LFO speeds are set with a combination of parameter 1 and input Y, which is a 1V/octave input centred on 4Hz for 0V.

Parameter 2 controls a feedback loop around the effect. Parameter 7 engages a saturation stage in the feedback loop, which can prevent clipping, while adding some colouration to the sound.

Parameter 3 applies a low pass filter to the chorus input. It does not affect the dry portion of the signals.

Parameters 4 & 5 together set the delay time, with a range of approximately 200ms. Very small values give flanging; medium values (25-50ms) are more typical for chorus. Larger values give recognisable echoes.

Parameter 6 sets the number of stages i.e. the number of simultaneous delayed/modulated signals that combine to produce the effect. Higher values will give a smoother, lush chorus; small values are more appropriate to give a sharper, more defined modulation, especially for flanging.

## L-6 Mono Chorus

[Video](#) 

**X is audio input**  
**Y is LFO rate**  
**A is blended output**  
**B is wet output**  
**Z is wet/dry**

Parameter	Min	Max	Default	Description
0	0	99	8	LFO depth.
1	-64	32	0	Y offset.
2	-63	63	0	Feedback.
3	-1	31	-1	Lowpass filter.
4	1	64	16	Delay time.
5	-32	32	0	Fine delay time.
6	1	6	6	Stages.
7	0	1	0	Saturation.

A mono chorus/flange effect.

X is the audio input.

A and B are audio outputs. Output A is a mix of the input and the chorus output; output B is just the chorus output.

Z is a wet/dry control. At zero, the output A is 100% dry i.e. the input signal with no chorus added. For positive Z, chorus is added while holding the dry level constant. For negative Z, the dry and chorus signals are progressively crossfaded, until eventually the output is 100% wet i.e. just the chorus signal.

The chorus effect is generated via a number of LFOs. The LFO depth is set with parameter 0. The LFO speeds are set with a combination of parameter 1 and input Y, which is a 1V/octave input centred on 4Hz for 0V.

Parameter 2 controls a feedback loop around the effect. Parameter 7 engages a saturation stage in the feedback loop, which can prevent clipping, while adding some colouration to the sound.

Parameter 3 applies a low pass filter to the chorus input. It does not affect the dry portion of the signal.

Parameters 4 & 5 together set the delay time, with a range of approximately 200ms. Very small values give flanging; medium values (25-50ms) are more typical for chorus. Larger values give recognisable echoes.

Parameter 6 sets the number of stages i.e. the number of simultaneous delayed/modulated signals that combine to produce the effect. Higher values will give a smoother, lusher chorus; small values are more appropriate to give a sharper, more defined modulation, especially for flanging.

## L-7 Mixer

[Video](#) 

**X & Y are inputs**  
**A is left output**  
**B is right output**  
**Z is pan for input X**

**Knob recorder enabled**

Parameter	Min	Max	Default	Description
0	-40	6	-6	Input X gain.
1	-40	6	-6	Input Y gain.
2	-32	32	0	Y pan.

A simple 2 channel mono-to-stereo mixer.

Parameters 0 & 1 set the gain for inputs X & Y respectively (in dB). At -40 the signal is completely attenuated (i.e. it's actually  $-\infty$ dB).

Z sets the pan for input X. The pan for input Y is set by parameter 2.

## L-8 Gate

[Video](#) 

**X is left input**  
**Y is right input**  
**A is left output**  
**B is right output**  
**Z is threshold**

Parameter	Min	Max	Default	Description
0	0	99	0	Attack time.
1	0	99	0	Hold time.
2	0	99	0	Release time.
3	0	99	0	Lookahead.

A stereo (noise) gate. The outputs are muted unless the input signal exceeds a threshold level.

The Z knob/CV sets the threshold, with a range of approximately 0-8V.

The outputs are muted by an envelope, triggered by the input signal rising above the threshold level. Parameters 0-2 set the attack, hold and release times of the envelope. The hold time is the minimum amount of time the envelope is considered to be triggered. The approximate time ranges for each parameter are: attack 10ms-1.8s, hold 0-600ms, release 45ms-8s.

Parameter 3 sets a lookahead time, effectively delaying the audio slightly relative to the trigger detection. This can be used to ensure that sharp transients are not lost during the attack time of the envelope. The maximum lookahead is about 10ms.

## M-1 Delayed LFO

[Video](#) 

**A and B are LFOs, ramps, or ramped LFOs**  
**X is trigger input**  
**Y is ramp time**  
**Z is LFO speed**

Parameter	Min	Max	Default	Description
0	0	6	1	Output type A.
1	0	6	0	Output type B.
2	-32	32	0	LFO speed range.
3	-32	32	0	Ramp time range.
4	-32	32	32	Attenuverter A.
5	-32	32	32	Attenuverter B.

This algorithm combines LFOs and a ramp generator, a combination often used for vibrato which starts gradually from the beginning of each note. The LFO is in fact not 'delayed' as such, but the naming is a nod to the equivalent feature on such classic synths as the Juno-6.

Input X triggers the ramp, which starts from zero and ramps linearly up to a maximum over a time set from Y, which is an exponential, 1V/octave, input. The LFO may be multiplied by the ramp, so it also starts from zero and ramps up to full amplitude.

The trigger also resets the LFO phase to zero. This could be used even without the ramp to give you a resettable LFO.

The Z knob/input controls the LFO speed, which is also an exponential control.

Parameters 0 and 1 select the output waveforms on outputs A & B respectively. The choices are:

0	the ramp itself
1	triangle affected by ramp
2	sine affected by ramp
3	square affected by ramp
4	triangle
5	sine
6	square

Parameters 2 & 3 set the range of the LFO and ramp times, both in quarter octaves (i.e. raising the value by 4 doubles the speed).

Parameters 4 & 5 are attenuverters for outputs A & B, which apply whatever the selected output waveforms. Negative values mean you can have a descending ramp, or LFOs with inverted phase. The unattenuated levels are 0-8V for the ramp and  $\pm 8V$  for the LFOs.

## M-2 Scaled LFO

[Video](#) 

**A and B are LFOs**  
**X and Y are min/max or offset/scale**  
**Z is LFO speed**

Parameter	Min	Max	Default	Description
0	0	6	1	Output type A.
1	0	6	0	Output type B.
2	-32	32	0	LFO speed range.
3	0	1	0	Mode.
4	-48	48	0	X Offset.
5	-48	48	0	Y Offset.

This algorithm generates LFOs, the amplitude and DC offset of which is controlled by the input CVs, according to parameter 3 ('Mode').

If parameter 3 is zero, the X & Y inputs set the minimum and maximum values that the LFO will reach. For example, if X is 3V and Y is 5V, the LFOs will sweep between 3-5V, or to look at it another way, the LFOs will be centred at 4V, with a peak-to-peak amplitude of 2V.

If parameter 3 is one, the X input sets the centre of the LFO, while the Y input sets its amplitude. For example, if X is 3V and Y is 1V, the LFOs' range will be 2V-4V. Note that in this mode if Y is zero, then A & B follow X. If X is a pitch CV, then you effectively have a vibrato control on Y.

The Z knob/input controls the LFO speed, which is an exponential control.

Parameters 0 and 1 select the output waveforms on outputs A & B respectively. The choices are:

0	triangle
1	sine
2	square
3	rising ramp
4	falling ramp

Parameter 2 sets the range of the LFO speed, in quarter octaves (i.e. raising the value by 4 doubles the speed).

Parameters 4 & 5 are offsets applied to inputs X & Y respectively. They are scaled in twelfths of a Volt, so if interpreted as pitch CVs, in semitones.

## M-3 Logic

[Video](#) 

**A and B are logic outputs**  
**X and Y are inputs**  
**Z selects logical operation for output A**

Parameter	Min	Max	Default	Description
0	-2	7	0	Output B logical operation.
1	-100	100	20	Input X threshold.
2	-100	100	20	Input Y threshold.
3	-100	100	10	Input X hysteresis.
4	-100	100	10	Input Y hysteresis.

This algorithm performs logical operations on the X & Y inputs. The inputs first pass through adjustable comparators, so you can feed X & Y either with straight gate/trigger signals, or with more general CVs.

X & Y are the inputs. A & B are the outputs, which are 0/5V logical (on/off) signals.

The Z knob/CV sets the logical operation performed for output A; parameter 0 sets the logical operation performed for output B. The name of the operation selected for output A is shown in the display as Z changes; the options available are the same as for output B, excluding the -1 and -2 values below:

Parameter 0 value	Operation
-2	Output B follows output A
-1	Output B is inverse of output A
0	AND
1	OR
2	XOR
3	NAND
4	NOR
5	XNOR
6	SR flip-flop
7	D flip-flop

The SR flip-flop is set high by a rising edge on input X, and cleared low by a rising edge on input Y.

The D flip-flop takes the level of input X on a rising edge on input Y.

Parameters 1-4 set the input comparator thresholds and hysteresis (for an explanation of hysteresis see [here](http://en.wikipedia.org/wiki/Hysteresis#Electronic_circuits)<sup>17</sup>). All are scaled such that 100 corresponds to 5V. The input is taken as logical '1' if the input voltage exceeds the threshold voltage.

<sup>17</sup> [http://en.wikipedia.org/wiki/Hysteresis#Electronic\\_circuits](http://en.wikipedia.org/wiki/Hysteresis#Electronic_circuits)

## M-4 Half-wave Rectifier

[Video](#) 

**A and B are outputs**  
**X and Y are inputs**  
**Z is threshold**

**Knob recorder enabled**

Parameter	Min	Max	Default	Description
0	0	3	0	Mode.

This algorithm processes the inputs with half-wave rectification<sup>18</sup>, in one of two different ways. Equally useful for audio and CV manipulation, this allows you to split a bipolar signal into its positive and negative sections, and to combine two different waveforms, using the positive part of one and the negative part of the other.

For ease of explanation, assume first that the Z knob/CV is at zero.

If the mode (parameter 0) is 0: output A is the positive part of X, plus the negative part of Y. Output B is the negative part of X, plus the positive part of Y.

If the mode is 1 or 2: the inputs are summed (mode 1) or subtracted (mode 2). Then output A is the positive part of  $X \pm Y$ , and output B is the negative part of  $X \pm Y$ .

If the mode is 3: output A is the positive part of input X; output B is the positive part of input Y.

Z is a threshold control, allowing the various comparisons above to be against a non-zero value. So for 'positive part' above read 'part where the value is more than Z' etc.

## M-5 Stereo Filter

[Video](#) 

**X is left input**  
**Y is right input**  
**A is left output**  
**B is right output**  
**Z is filter frequency**

**Knob recorder enabled**

Parameter	Min	Max	Default	Description
0	0	2	0	Mode.
1	0	31	0	Resonance (Q).

A stereo filter, switchable between low-, band-, and high-pass (via parameter 0).

Parameter 1 sets the filter resonance.

<sup>18</sup> [https://en.wikipedia.org/wiki/Rectifier#Half-wave\\_rectification](https://en.wikipedia.org/wiki/Rectifier#Half-wave_rectification)



## M-6 Stereo Tape Delay



**X is left input**  
**Y is right input**  
**A is left output**  
**B is right output**  
**Z is tape speed or feedback**

Parameter	Min	Max	Default	Description
0	0	26	26	Tape length.
1	-9	9	0	Fine length control.
2	-1	36	24	Feedback.
3	0	1	0	Output mode.
4	-8	16	0	Tape speed.

This is a stereo delay/echo effect which simulates a variable speed tape loop echo device. The delay time at '1x speed', with the tape length parameter at maximum, is just over 250ms.

X & Y are the audio inputs.

Parameters 0 & 1 together set the 'tape length', effectively setting the delay time. Parameter 0 is scaled in 10ms increments, while parameter 1 is in 1ms increments (both assuming 1x 'tape speed').

If parameter 2 is 0 or more:

Z controls the speed of the 'tape', which also affects the delay time. The speed range is from half speed to double speed. Parameter 4 is added to the Z value as an offset.

Parameter 2 controls the feedback. A value of 32 gives 100% feedback.

If parameter 2 is -1:

Z controls the feedback.

Parameter 4 controls the speed of the 'tape', which also affects the delay time. The speed range is from half speed to double speed.

Outputs A & B carry signals according to parameter 3:

Parameter value	Output mode
0	A & B are mix.
1	A & B are delay only.

'Mix' is a mix of the dry (undelayed) signal and the delay effect. The amount of delay in the mix rises in direct proportion to the amount of feedback.

'Delay only' is the delayed signal only. Use this and the input signals, plus an external mixer, if you need more flexibility in the dry/wet balance than is offered by the mixed mode.

## M-7 Granular Pitch Shifter

[Video](#) 

**X is audio input**  
**Y is pitch 1V/octave**  
**A is mix output**  
**B is pitch shifted output**  
**Z is mix or freeze**  
**Z press is freeze**

Parameter	Min	Max	Default	Description
0	0	99	50	Grain length.
1	0	3	0	Window type.
2	0	32	16	Delay.
3	0	64	0	Feedback.
4	-48	48	0	Y Offset (semitones).
5	-99	99	0	Fine (cents).
6	-65	64	-65	Mix.

This is a pitch shifting audio effect, which can also be used as a delay. The pitch shifting is done using a granular technique, that is, the output is composed of many short segments or 'grains' taken from the input.

Input X is the audio input.

Input Y is the pitch CV input, following the 1V/octave standard. Positive input voltages shift the pitch up; negative voltages shift the pitch down.

Output A is a mix of the original audio and the pitch shifted version (according to Z); output B is the pitch shifted version only.

Z is a wet/dry control. At zero, output A is 100% dry i.e. the input signal only. For positive Z, the pitch shifted version is added while holding the dry level constant. For negative Z, the dry and pitch shifted signals are progressively crossfaded, until eventually the output is 100% wet i.e. just the pitch shifted version.

Parameter 0 sets the grain length. Parameter 1 sets the shape of the grain 'window', or if you prefer, its envelope. Best results will depend on the source material – experiment.

The effect uses an internal buffer to store the source material so it can be replayed as grains – therefore there is some delay (or latency) inherent in the algorithm. The delay can be controlled by parameter 2. If you are only ever pitching down, this can be set very low. Pitching up requires a higher latency. If you are viewing this as a delay effect, then set the delay to whatever you like. The maximum delay time is about 400ms.

Parameter 3 sets the delay feedback.

Parameters 4 & 5 provide a means of manually setting a pitch shift without feeding a signal into input Y. Parameter 4 shifts in semitones, while parameter 5 shifts in cents.

Pressing the Z knob 'freezes' the delay buffer – the input signal is no longer written into the buffer and the effect will continuously output the frozen material. Press Z again to unfreeze it.

Parameter 6 allows you to set the mix and free up the Z CV input to control Freeze. When the parameter is at -65 (the default) Z controls the mix. Other values of the parameter set the mix

directly (from 100% wet to dry to an equal mix, just like when controlling it via Z), and the Z input becomes a gate control for the Freeze function, freezing the buffer when the CV goes over approximately 1V.

## M-8 Chaos

[Video](#) 

**X & Y influence the chaos**  
**A & B are chaotic outputs**  
**Z is speed**

Parameter	Min	Max	Default	Description
0	-64	32	0	Range.
1	0	11	0	Outputs.
2	-64	64	32	Atten A.
3	-64	64	32	Atten B.
4	-32	32	0	Offset A.
5	-32	32	0	Offset B.
6	0	1	0	Clamp.

This algorithm generates chaotic CVs and/or gates according to the [Lorenz equations](#)<sup>19</sup>.

The X & Y inputs set parameters of the equations – X affects 'r' (aka 'ρ'), and Y affects 'b' (aka 'β'). With the CVs at 0V, the parameters are the classic values as studied by Lorenz (28 and 8/3 respectively).

The A & B outputs generate the x, y or z values of the Lorenz system, or gates based on these values, according to parameter 1 as follows:

Parameter 1	Output A	Output B
0	x	z
1	y	z
2	x	y
3	x-based gate	z
4	y-based gate	z
5	x-based gate	y
6	x	z-based gate
7	y	z-based gate
8	x	y-based gate
9	x-based gate	z-based gate
10	y-based gate	z-based gate
11	x-based gate	y-based gate

When a gate output is chosen, the output is 5V when the x/y/z value is above zero, and 0V when the value is below zero. This comparison happens after the scale/offset from parameters 2-5, so the

<sup>19</sup> [https://en.wikipedia.org/wiki/Lorenz\\_system](https://en.wikipedia.org/wiki/Lorenz_system)

precise gates obtainable are affected by these parameters.

The Z knob/CV sets the speed of the simulation i.e. the rate at which the outputs change.

Parameter 0 sets the range of speed values attainable, in quarter octaves (i.e. raising the value by 4 doubles the speed).

Parameters 2 & 3 are attenuverters for outputs A & B.

Parameters 4 & 5 are offsets for outputs A & B (with a range of  $\pm 8V$ ).

Parameter 6 enables a clamp, in the event that the simulation becomes numerically unstable, in which case it is reset to initial conditions. With the clamp disabled, the simulation can generate extreme values, which have nothing to do with the Lorenz equations but are potentially useful in their own right.

Pushing Z resets the simulation to initial conditions.

## N-1 Switch

[Video](#) 

**X and Y are inputs  
A and B are outputs  
Z is control CV or trigger**

**Knob recorder enabled**

Parameter	Min	Max	Default	Description
0	0	1	0	Mode.
1	0	32	3	Hysteresis.
2	0	99	0	Fade.
3	-1	99	99	Chance.

This algorithm is a voltage controlled or triggered crossover switch. That is, either

$X \rightarrow A$  and  $Y \rightarrow B$

or

$X \rightarrow B$  and  $Y \rightarrow A$

Note that by using only one input, the algorithm effectively directs an input to one of two outputs (e.g. X to either A or B); likewise, using only one output, the algorithm chooses between two inputs (e.g. A is either X or Y).

Being fully DC-coupled, the algorithm can be used for switching either audio or CV.

Parameter 0 chooses the mode – whether the Z knob/CV is a direct switch or a trigger.

In mode 0, Z is a switch. The switch occurs at 0V. Hysteresis can be applied to the switching voltage with parameter 1.

In mode 1, Z is a trigger. A positive-going input edge toggles the switch.

When using an external CV as a switch or trigger, you may like to turn the Z knob slightly counterclockwise.

A crossfade can be applied when switching to avoid clicks. The length of the fade (in milliseconds) is controlled by parameter 2.

Parameter 3 sets the probability that a switch transition will actually occur when triggered. Add one to the parameter value to get the chance in %. The default value of 99 means a 100% chance of the switch transitioning; a value of -1 means a 0% chance i.e. the switch will never change.

## N-2 Rotary

[Video](#) 

**X is audio input**  
**Y is mix**  
**A and B are stereo outputs**  
**Z is speed**

Parameter	Min	Max	Default	Description
0	0	32	32	Mix.
1	0	63	24	Depth.
2	0	99	10	Slew.
3	0	32	15	Crossover.
4	0	64	15	Bass modulation.

This algorithm provides an emulation of a rotary speaker effect.

X is the audio input; A & B are the stereo audio outputs.

The dry/wet mix is controlled by the sum of the Y input and parameter 0.

The Z knob/CV controls the rotation speed. The 0-5V range corresponds to speeds from 0.8Hz to 6.7Hz. Below about -1V the speed decreases and eventually stops altogether.

Parameter 1 sets the depth of the pitch modulation.

Parameter 2 sets the slew rate for speed changes. This plus a 5V gate signal into Z are your route to the classic slow/fast rotary speed transitions.

Parameter 3 controls the frequency of a (first order) crossover filter. The part of the signal above the filter frequency is fed into the rotary pitch modulation effect. The bass signal (below the crossover frequency) is not pitch modulated, but is amplitude modulated at a slightly slower rate than the treble modulation. The depth of the bass modulation is set by parameter 4.

If parameter 3 is zero, the crossover is disabled and the separate bass modulation is not applied.

If the crossover is used, but parameter 4 is zero, this essentially removes the effect entirely from the bass portion of the input, keeping it centre stereo and at unmodulated pitch.

### N-3 Attenuverter/Offset

[Video](#) 

**A = X \* scale + offset**

**B = Y \* scale + offset**

**Z is not used**

Parameter	Min	Max	Default	Description
0	-100	100	50	Attenuversion A.
1	-100	100	0	Offset A.
2	-100	100	50	Attenuversion B.
3	-100	100	0	Offset B.

This algorithm provides two independent channels of attenuversion (that is, attenuation and possible inversion) and offset.

Parameters 0 & 2 are the attenuversion parameters, scaled such that a parameter value of 50 gives 100% i.e. the signal level is unchanged.

Parameters 1 & 3 are the offset parameters, scaled such that a parameter value of 100 gives 10V.

### N-4 Low Pass Gate

[Video](#) 

**X and Y are stereo inputs**

**A and B are stereo outputs**

**Z is frequency/'strike'**

Parameter	Min	Max	Default	Description
0	0	31	0	Resonance.
1	1	99	50	Up slew.
2	1	99	50	Down slew.

This algorithm is a stereo low pass gate, that is, a combination of low pass filter and VCA. It makes no claim to emulate any particular bit of hardware, classic or modern, but the response of the filter to a 'strike' on the frequency input has been designed to be somewhat like that of a vactrol-based low pass gate.

X & Y are the audio inputs; A & B are the audio outputs.

Z controls the filter frequency/VCA CV. This has slew applied to it, which is controlled by parameters 1 & 2.

Parameter 0 sets the filter resonance.

## N-5 Pulsar VCO

[Video](#) 

**X is V/Oct pitch input (fundamental)**  
**Y is V/Oct pitch input (formant)**  
**Z is tune or masking**  
**A & B are pulsar outputs**

**Receives MIDI**

Parameter	Min	Max	Default	Description
0	0	99	0	Chooses the wavetable.
1	0	99	0	Chooses the wave.
2	0	3	0	Window.
3	0	1	0	Y Mode.
4	0	99	0	Masking.
5	-16	8	0	Octave.

This algorithm is an implementation of pulsar synthesis, inspired by the description in Curtis Roads's book *Microsound* (MIT Press)<sup>20</sup> pp137-157.

In pulsar synthesis, a small grain of sound (known as a *pulsaret*, typically a pulse or single cycle waveform) is repeated periodically, followed by a small section of silence. The length of the sound within the repeating period (equivalently, the speed at which it is played) offers a means of control the timbre without affecting the perceived pitch of the sound. By the same token, changing the fundamental pitch without changing the speed at which the grain of sound is played offers pitch change without simply shifting the entire spectrum up and down, an effect reminiscent of 'vocal' or 'formant' synthesis.

This algorithm uses wavetables from the SD card as the source of the pulsarets. See 'Loading wavetables' above for a description of how to load wavetables onto the SD card.

Input X sets the fundamental frequency – the rate at which the pulsaret train is generated.

Input Y sets the 'formant' frequency – the rate at which the pulsaret is played. By default this is independent of the fundamental frequency. If parameter 3 is set to 1, the X input is added to the Y input, so changing the fundamental frequency also changes the formant frequency.

Parameters 0 & 1 choose the wavetable and the wave within the table, respectively.

Parameter 2 chooses the window or envelope to apply to the pulsaret. The options are:

Parameter 2 value	Window shape
0	Rectangular
1	Linear attack
2	Linear decay
3	Gaussian

The Z knob/CV depends on parameter 4. When parameter 4 is zero (the default), Z is a tuning control, with a range of approximately  $\pm 0.5$  octaves. Otherwise, masking is applied – that is, pulsarets are omitted from the train. In this case, outputs A & B use inverse masks – output B will output a pulsaret when output A is suppressed, and *vice versa*. If parameter 4 is 1, stochastic

<sup>20</sup> Some of the text of this book is available on [Google Books](#) but I would urge you to pick up a physical copy.

masking is applied i.e. pulsarets are randomly masked. The likelihood of a pulsaret being masked is set by Z. If parameter 4 is 2 or more, burst masking is applied – the parameter sets a number of pulsarets, and Z controls how many pulsarets within that number will be masked.

Parameter 5 offsets the X input in octave steps.

**MIDI Input:** MIDI note messages received on the MIDI channel set in the Settings are converted to a CV which is added to that on input X. The 0V note is 48 (C3).

## N-8 Clockable SD Ping Pong

[Video](#) 

**X is signal**  
**Y is clock input**  
**A & B are stereo audio outputs**  
**Z is feedback**

---

**Tap tempo enabled**

Parameter	Min	Max	Default	Description
0	-15	8	0	Delay time multiplier.
1	0	8	8	Maximum feedback.
2	0	1	0	Output mode.
3	-32	32	-32	Input pan.

This is a ping-pong version of the Clockable SD Delay algorithm as described above.

Being stereo, the maximum delay time is halved, to a mere 48 minutes.

Please be sure to read the 'SD card requirements' box.

Outputs A & B are the left and right outputs respectively. If the output mode parameter is 0 (the default), they are a mix of the dry (undelayed) signal and the delay effect. The amount of delay in the mix rises in direct proportion to the amount of feedback. If the output mode parameter is 1, the outputs are the delayed signals only. Use this and the input signal, plus an external mixer, if you need more flexibility in the dry/wet balance than is offered by output mode 0.

Parameter 2 controls the left/right pan position of the input signal.



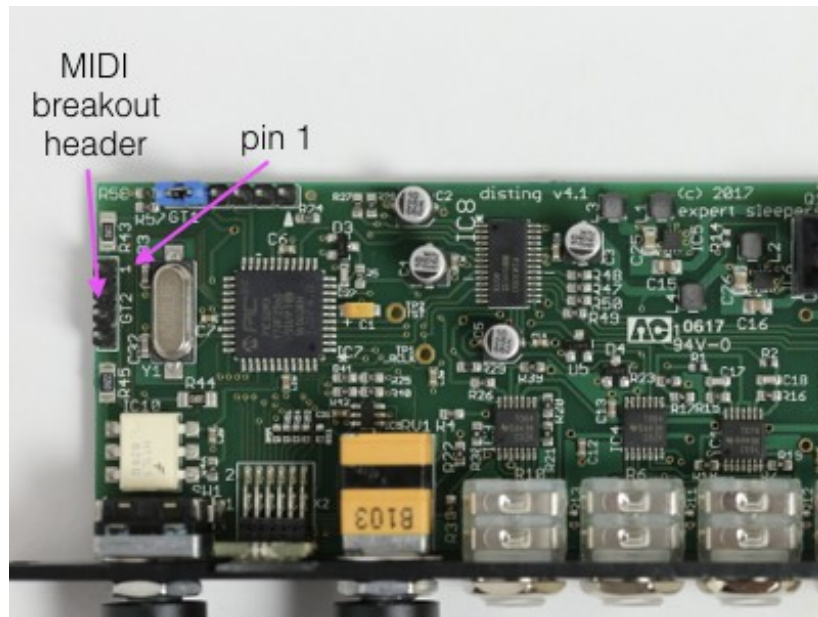
# MIDI I/O

## Introduction

The disting mk4 has a pair of industry standard MIDI ports - one input, one output - which can be used to connect to a huge variety of external equipment.

## Connections

The MIDI ports are exposed on a 4-pin expansion header (labelled GT2) at the top of the PCB:



All of the necessary electronics are on the PCB; all that needs to be added is the actual sockets.

The header pin-out is as follows:

Pin	Function
1	OUT pin 4
2	OUT pin 5
3	IN pin 5
4	IN pin 4

Pin numbers refer to the 5-pin DIN socket as in the [MIDI standard](#).

The header is a standard 0.1" pitch header compatible with e.g. typical IDC cable sockets.

## Breakout module

A passive breakout module which exposes the MIDI ports as 5-pin DIN sockets is available (also in kit form). See [this page](#).

## MIDI "Low-Voltage Signaling"

When MIDI was first conceived, standard logic circuits ran at 5V, and the initial hardware reference

designs worked on this assumption. However, there is nothing in the MIDI standard that actually requires a specific voltage, since it operates as a current loop via an opto-isolator.

Nowadays it is much more common for logic to run at 3.3V or lower. The MIDI Manufacturers Association has now released a reference hardware design based on 3.3V, which they call the "Low-Voltage Signaling" standard, and it is this that the disting mk4 uses.

The disting is therefore fully compatible with any device that adheres strictly to the MIDI specification, but may not be with devices that play fast and loose with the spec and assume 5V.

Download the full MIDI electrical specification [here](#).

## **Controlling the disting mk4 via MIDI**

In addition to any algorithm-specific MIDI implementation, there are common MIDI controls shared by all algorithms. These are as follows. '<in>' for the MIDI channel indicates the MIDI input channel as selected in the settings.

<b>MIDI channel</b>	<b>Message</b>	<b>Function</b>	<b>Note</b>
<in>	CC 1	Set algorithm parameter 0	
<in>	CC 2	Set algorithm parameter 1	
<in>	CC 3	Set algorithm parameter 2	
<in>	CC 4	Set algorithm parameter 3	
<in>	CC 5	Set algorithm parameter 4	
<in>	CC 6	Set algorithm parameter 5	
<in>	CC 17	Set Z	Sending this CC takes control of the algorithm's Z function and overrides the Z knob and CV input. The Z function remains overridden until the algorithm is changed.
<in>	CC 18	Select algorithm	
<in>	CC 19	Free Z	If Z has been overridden by CC 17, sending this CC with a value of 64 or more cancels the override and restores Z control to the knob/CV input.
<in>	Program change	Select algorithm / Load preset	Either selects an algorithm or loads a preset, as set in the settings.

## **MIDI Thru**

If enabled in the settings, the MIDI out port echoes any incoming MIDI arriving at the in port. Note that no attempt is made to process or merge this MIDI stream. If the disting is itself generating MIDI output, using the Thru function at the same time is likely to result in garbage MIDI output.

## Calibration

Here you will find details of

- How a disting is calibrated.
- How to use the calibration data in your own code.

The disting is calibrated before it leaves the factory. It is not expected that you would need to do this yourself in normal circumstances.

### **Calibration Procedure**

Calibration requires a 3V voltage reference. Any stable and accurate voltage source can be used. It's a good idea to let the whole system warm up for 5 minutes before calibrating to stabilise any temperature-dependent factors.

The procedure is as follows:

1. With the disting powered up and with nothing connected to it, use the menu to select option 5, 'Calibrate'. The message 'GO' will be displayed.
2. Wait a few seconds for the display to show '1' and for the left-most column to flash.
3. Using patch cables (as short as possible), connect output A to input X, and output B to input Y.
4. Press the encoder.
5. After a short delay, all four jack sockets should light up red. After another short delay, the display will show 'X' and the left-most column will flash.
6. Remove the patch cables. Connect the 3V voltage reference to input X.
7. Verify the reference voltage, and press the encoder.
8. Wait a few seconds for the display to show 'Y' and the left-most column to flash.
9. Connect the 3V voltage reference to input Y.
10. Verify the reference voltage, and press the encoder.
11. After a short pause, the disting will reboot, at which point the reference can be removed.

If after step 10 the disting displays the message "Uncalibrated", then the calibration was not performed correctly, and the data will not be used. Start back at step 1.

### **Calibration Data**

The disting's calibration data is stored in eight 32 bit words from address 0xBD01FE00 (disting mk1) or 0xBD03FE00 (disting mk2/mk3). The first four refer to the left channel (input X/output A); the second four refer to the right channel (input Y/output B).

Each value is a sample of the relevant input under certain conditions. These conditions are

1. Input disconnected (and therefore grounded)
2. Input connected to output, output outputting a nominal zero level
3. Input connected to output, output outputting a nominal half full range level
4. Input connected to known 3V reference

From these we can derive constants A, Br, D, Er so that conversion between codec codes and actual voltages can be handled simply in the code as follows:

```
int vL = ( ( inL - A[0] ) * Br[0] ) >> 24;
int vR = ( ( inR - A[1] ) * Br[1] ) >> 24;

int cL = ( ( vOutL - D[0] ) * Er[0] ) >> 24;
int cR = ( ( vOutR - D[1] ) * Er[1] ) >> 24;
```

where inL/inR are the values read directly from the audio codec, cL/cR are the codes written directly to the audio codec, and vL/vR/vOutL/vOutR are .19 format voltages.

The derivation of A, Br, D, Er is as follows:

```
Let the four input samples be zeroIn, zeroOut, halfOut, threeVolt.
A = zeroIn
B = ( threeVolt - zeroIn )/3
Br = 0x800000000000LL / B
D = ( zeroOut - zeroIn )/B
E = ( halfOut - zeroOut )/( B * 0x400000 )
Er = 0x800000000000LL / E
```

The above is pseudo-code - a real C implementation is left as an exercise for the reader.

# Firmware Updates

## [Video](#)

The disting mk4's firmware can be updated using a Micro SD card. The process is as follows:

- Download the firmware from the Expert Sleepers [website](#)<sup>21</sup>.
- Unzip the download.
- Copy the file 'disting4.bin' (firmware versions from 4.15 for bootloader version b2) or 'image.hex' (firmware versions up to 4.14.2 for bootloader version b1) from the unzipped download to the root folder of a Micro SD card.
- Turn off the disting and insert the Micro SD card into the socket.
  - The exposed contacts of the card should face towards the display; the angled side of the card itself points up.
  - **You are advised to observe basic ESD (electrostatic discharge) precautions** - see for example [here](#). Don't be wearing a nylon sweater in an air conditioned room.
- Turn on the disting with the S knob pressed. It will go through the first stage of its startup sequence, showing the bootloader version number.
- Keep the S knob pressed until the message "Confirm..." is shown.
  - If at this point the disting shows the message "No firmware found on card", no firmware file could be found on the card. Turn off the power, remove the card and check its contents.
- Release the knob to begin the update process. (If you change your mind and don't want to proceed, simply turn the power off.)
- When the update starts, the display will flash rapidly to indicate progress. Do not remove the card or turn off the power during the update.
- When the update is complete, the disting will show the message "Success" if successful, or "Error:" plus an error code if there was a problem.
- Turn off the disting and turn it back on to resume normal operation.

Note that the Micro SD card needs to be formatted as FAT32 with MBR partitioning (see the section on supported cards, above).

---

<sup>21</sup> <https://expert-sleepers.co.uk/distingfirmwareupdates.html>

## Acknowledgments

The font used in the disting mk4 is an adapted version of 'Tom Thumb' by Robey Pointer. More on the font [here](#).