

Expert Sleepers

disting mk4 **Firmware v4.4** **User Manual**

Copyright © 2017 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.....	6
Installation.....	6
Power requirements.....	6
Inputs, Outputs and Controls.....	6
Inserting a MicroSD card.....	6
Startup.....	7
The encoder & menu system.....	7
Selecting an algorithm.....	7
Settings.....	8
Help.....	8
Parameters.....	9
Knob Recorder.....	9
Tap Tempo.....	10
Favourites.....	10
Presets.....	10
Preset 0.....	11
Select Bus.....	11
SD Card Playback.....	11
Supported MicroSD cards.....	11
Supported audio files.....	12
Supported MIDI files.....	12
File naming & Playlists.....	12
Audio Playlist Format.....	12
MIDI Playlist Format.....	14
Example Files.....	15
MIDI/CV conversion.....	15
Scala Support.....	15
Organising the files on the card.....	15
Example Files.....	15
Scale Playlist Format.....	16
Possible Sources of Error.....	16
Algorithm Overview.....	17
A-1 Precision Adder.....	18
A-2 Four Quadrant Multiplier.....	18
A-3 Full-wave Rectifier.....	19
A-4 Minimum/maximum.....	19
A-5 Linear/Exponential Converter.....	19
A-6 Quantizer.....	20
A-7 Comparator.....	21
A-8 Dual Waveshaper.....	21
B-1 Sample and Hold.....	22
B-2 Slew Rate Limiter.....	22
B-3 Pitch and Envelope Tracker.....	22
B-4 Clockable Delay/Echo.....	23
B-5 LFO.....	24
B-6 Clockable LFO.....	25
B-7 VCO with linear FM.....	25
B-8 VCO with waveshaping.....	26

C-1 Precision Adder (fractional offsets).....	26
C-2 Voltage Controlled Delay Line.....	27
C-3 Clockable Ping Pong Delay (Z feedback).....	27
C-4 Clockable Ping Pong Delay (Z input pan).....	28
C-5 Resonator.....	28
C-6 Vocoder.....	29
C-7 Phaser.....	30
C-8 Bit Crusher.....	30
D-2 Tape Delay.....	32
D-3 Waveform Animator.....	32
D-4 State Variable Filter.....	33
D-5 LP/HP Filter.....	34
D-6 LP/BP Filter.....	34
D-7 BP/HP Filter.....	34
D-8 BP/Notch Filter.....	35
E-1 AR Envelope.....	35
E-2 AR Envelope (with push).....	36
E-3 AR Envelope & VCA.....	36
E-4 AR Envelope & VCA (with push).....	37
E-5 Dual AR Envelope.....	38
E-6 Dual AR Envelope (with push).....	38
E-7 Euro to Buchla Converter.....	39
E-8 Buchla to Euro Converter.....	39
F-1 Clockable AD Envelope (with mute).....	40
F-2 Clockable AD Envelope (with gate).....	40
F-3 Clockable AD Envelope (with trigger).....	41
F-4 Clockable AD Envelope & VCA.....	41
F-5 Shift Register Random CVs.....	42
F-6 Shift Register Random Quantized CVs.....	43
F-7 Shift Register Random Triggers.....	44
F-8 Shift Register Random Dual Triggers.....	44
G-1 ES-1 Emulation.....	45
G-2 ES-2 Emulation.....	45
G-3 Pitch Reference.....	46
G-4 Frequency Reference.....	46
G-5 Tuner.....	47
G-6 MIDI Clock.....	47
G-7 MIDI/CV.....	49
G-8 CV/MIDI.....	49
H-1 Crossfade/Pan.....	50
H-2 Dual Sample and Hold.....	50
H-3 Dual Quantizer (Z scale).....	51
H-4 Dual Quantizer.....	52
H-5 Dual Euclidean Patterns.....	53
H-6 Dual Delayed Pulse Generator.....	53
H-7 Noise.....	54
I-1 Audio Playback.....	55
I-2 Clocked Audio Playback.....	55
I-3 Audio Playback with V/Oct.....	56
I-4 Audio Playback with Z Speed.....	56

J-1 MIDI File Playback (Clocked).....	57
J-3 MIDI File Playback (Free Running).....	57
J-4 Audio Playback with End CV.....	58
J-5 Audio Recorder.....	58
K-1 Wavetable VCO.....	59
Loading wavetables.....	59
K-5 Programmable Quantizer.....	61
L-1 Stereo Reverb.....	62
L-2 Mono-to-Stereo Reverb.....	62
L-3 Dual Reverb.....	63
L-5 Stereo Chorus.....	64
L-6 Mono Chorus.....	64
MIDI I/O.....	66
Introduction.....	66
Connections.....	66
Breakout module.....	66
MIDI "Low-Voltage Signaling".....	66
Controlling the disting mk4 via MIDI.....	67
MIDI Thru.....	67
Calibration.....	68
Calibration Procedure.....	68
Calibration Data.....	68
Firmware Updates.....	70
Acknowledgments.....	71

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 four 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
- 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.

Alternatively, you can switch between algorithms by pushing and holding the S knob while turning it, though this does visit every algorithm you pass through, rather than jumping directly to the

chosen one, as is the case when using the menu.

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)

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.

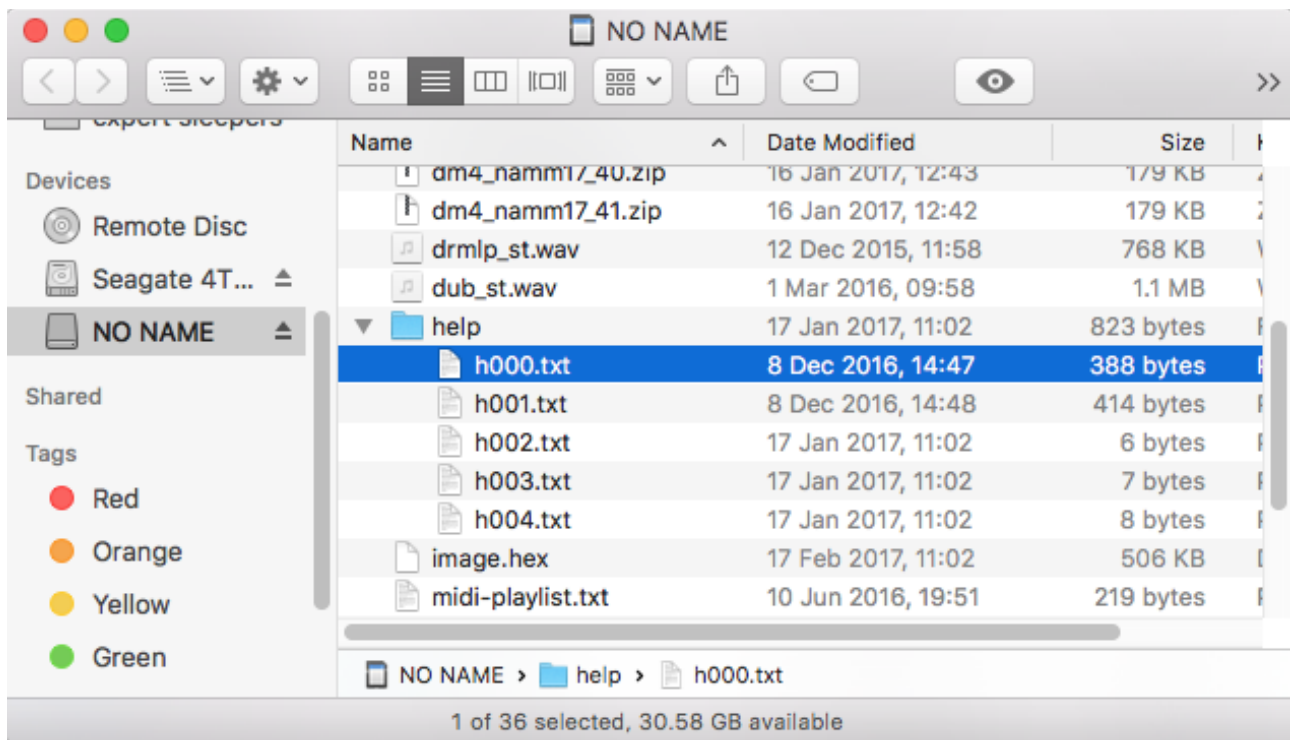
Help

[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 six '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. The current parameter number is displayed briefly followed by its name.

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, and
- the parameter values.

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.

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 0.¹

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.

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 Tempj](#).

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

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.

¹ 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.

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.

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".

File naming & Playlists

All files need to be in the root of the drive i.e. not inside any folder.

For audio files, we still support the legacy naming convention from the early disting mk3 implementation, which required that files be named 'sampleXX.wav' where XX is a decimal number from 00 to 31 e.g. sample01.wav, sample23.wav. However the playlist file is the preferred method going forward.

The playlist file specifies what audio/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.

By default, audio algorithms look for a playlist file called 'playlist.txt' and MIDI algorithms look for 'midi-playlist.txt'. However, each 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.

Setting	Default	Description
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.
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.

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.

[Video](#) 

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

The maximum number of sample files per playlist is 64. 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.

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](#)².

MIDI/CV conversion

There are two 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. 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 Scala³. 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 'kbn' folder.** A folder named 'kbn' in the root folder of the card, which contains all the keyboard mapping (.kbn) 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](#)⁴ 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.

Example Files

A zip file containing some example files, which are known to work, is [here](#)⁵.

2 http://expert-sleepers.co.uk/downloads/content/disting_sd_content.zip

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

4 http://expert-sleepers.co.uk/downloads/content/logTables_16_20.bin

5 http://expert-sleepers.co.uk/downloads/content/disting_example_scales.zip

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.

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		Tape Delay	Waveform Animator	State Variable Filter	LP/HP Filter	LP/BP Filter	BP/HP Filter	BP/Notch Filter
E	AR Envelope	AR Envelope (with push)	AR Envelope & VCA	AR Envelope & VCA (with push)	Dual AR Envelope	Dual AR Envelope (with push)	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	MIDI 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	
I	Audio Playback	Clocked Audio Playback	Audio Playback with V/Oct	Audio Playback with Z Speed				
J	MIDI File Playback (Clocked)		MIDI File Playback (Free Running)	Audio Playback with End CV	Audio Recorder			
K	Wavetable VCO				Programmable Quantizer			
L	Stereo Reverb	Mono-to-Stereo Reverb	Dual Reverb		Stereo Chorus	Mono Chorus		

A-1 Precision Adder

[Video](#) 

$$\begin{aligned}A &= X + Y + \text{offset} \\B &= X - Y - \text{offset} \\ \text{offset} &= \pm 10\text{V in } 1\text{V steps derived from } Z\end{aligned}$$

Knob recorder enabled

Parameter	Min	Max	Default	Description
0	0	1	0	When 1, the offset is not restricted to 1V steps, and changes smoothly with Z.

Output A is the sum of inputs X & Y; output B is the difference between inputs X & Y. With nothing plugged into input X, B is therefore simply an inverted copy of Y.

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.

A-2 Four Quadrant Multiplier

[Video](#) 

$$\begin{aligned}A &= X * Y * \text{scale} \\B &= -X * Y * \text{scale} \\ \text{scale} &= 1/10 \text{ to } 10\text{x in steps derived from } Z\end{aligned}$$

Knob recorder enabled

Parameter	Min	Max	Default	Description
0	0	1	0	When 1, the scale is not restricted to integer steps, and changes smoothly with Z.

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.

A-3 Full-wave Rectifier

[Video](#) 

$$\begin{aligned} \mathbf{A} &= \mathbf{abs(X + Y) \text{ or } abs(X)} \\ \mathbf{B} &= \mathbf{abs(X - Y) \text{ or } abs(Y)} \\ \mathbf{Z} &\mathbf{ selects mode} \end{aligned}$$

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](#) 

$$\begin{aligned} \mathbf{A} &= \mathbf{min(X, Y)} \\ \mathbf{B} &= \mathbf{max(X, Y)} \\ \mathbf{Z} &\mathbf{ is gate} \end{aligned}$$

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](#) 

$$\begin{aligned} \mathbf{A} &= \mathbf{(2 ^ X) * scale} \\ \mathbf{B} &= \mathbf{log2(Y / scale)} \\ \mathbf{Z} &\mathbf{ is Hz/V scale, centred on 1kHz} \end{aligned}$$

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).

A-6 Quantizer

[Video](#) 

A = quantized(X)
B = 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	-1	31	31	Input X attenuation.
1	0	1	0	Transpose mode.
2	-31	31	0	Key.
3	0	31	0	Offset.

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. After the name has scrolled across, the display reverts to showing the current algorithm.

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.

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: 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

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.

A-8 Dual Waveshaper

[Video](#) 

A = folded X
B = triangle-to-sine Y
Z is gain

Knob recorder enabled

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.

Input X/output A provide what is usually termed a wavefolder. This increases the harmonic content of the sound in interesting ways, especially as the gain changes.

Input Y/output B provide 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.

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 white noise signal, with range $\pm 8V$. A noise signal is commonly fed into the input of a sample and hold device to generate clocked random voltages.

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.

Pressing the Z knob triggers a sample manually.

B-2 Slew Rate Limiter

[Video](#) 

A = linear slew rate limited (X + Y)
B = log slew rate limited (X + Y)
Z is slew rate

Outputs A & B are both slew rate limited copies of the sum of X & Y. Output A uses 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. Output B uses 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.

B-3 Pitch and Envelope Tracker

[Video](#) 

A = V/octave pitch derived from X, plus Y
B = envelope derived 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 = dry + delay in ratio according to feedback
B = delay signal only

Tap tempo enabled

Parameter	Min	Max	Default	Description
0	-15	8	0	Delay time multiplier.

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%.

Output A 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.

Output B 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 output A.

The 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	-1	31	31	Sets an attenuation for output A.
1	-1	31	31	Sets an attenuation for output B.
2	-31	31	0	Sets an offset for output A.
3	-31	31	0	Sets an offset for output B.

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 scale factor is $(1 + \text{parameter value})/32$, so at the minimum parameter value of -1 there is no output at all.

The outputs can also have a DC offset applied. The offset in Volts is $(\text{parameter value})/4$ i.e. the range is ± 7.75 , 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 $\pm 10\text{Hz}$. 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

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	-1	31	31	Sets an attenuation for both LFO outputs.

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 via the parameter. The scale factor is $(1 + \text{parameter value})/32$, so at the minimum parameter value of -1 there is no output at all.

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

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. After a short while the display reverts to showing the current algorithm.

B-7 VCO with linear FM

[Video](#) 

X is V/Oct pitch input
Y is linear FM input
Z is tune ± 0.5 octaves
A is sine
B is saw

Parameter	Min	Max	Default	Description
0	-16	8	0	Octave shift.
1	-1	31	31	Sets an attenuation for output A.
2	-1	31	31	Sets an attenuation for output 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 ±0.5 octaves.

The A and B outputs provide sine and saw waves respectively. The output signals are ±8V (16V peak-to-peak) by default, but can be attenuated via the parameters. The scale factor is (1+parameter value)/32, so at the minimum parameter value of -1 there is no output at all.

B-8 VCO with waveshaping

[Video](#) 

X is V/Oct pitch input
Y is waveshape/PWM
Z is tune ±0.5 octaves
A is saw -> tri -> saw
B is pulse -> square -> pulse

Parameter	Min	Max	Default	Description
0	-16	8	0	Octave shift.
1	-1	31	31	Sets an attenuation for output A.
2	-1	31	31	Sets an attenuation for output B.
3	-31	31	0	Offset for input Y (manual shape control).

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 ±0.5 octaves.

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

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

The output signals are ±8V (16V peak-to-peak) by default, but can be attenuated via the parameters. The scale factor is (1+parameter value)/32, so at the minimum parameter value of -1 there is no output at all.

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.

This algorithm is basically the same as 1-a Precision Adder, 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	-31	31	0	Applies an offset to the Y input.

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. The parameter 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 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.

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.

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.

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.

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.

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.

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.

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

Outputs A & B are the left and right outputs respectively. 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.

The delay feedback is set via the parameter.

C-5 Resonator

[Video](#) 

X is audio 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	-31	31	0	Applies an offset to the Y input.

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.

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.

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.

The parameter selects between alternative filter banks.

Bank	Description
0	Half octave spacing, based on 100Hz.
1	Third octave spacing, based on 250Hz.

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-2 Tape Delay

[Video](#) 

X is audio input
Y is tape speed
Z is feedback
A = dry + delay in ratio according to feedback
B = delay signal only

Parameter	Min	Max	Default	Description
0	0	31	31	Tape length.
1	-8	16	0	Tape speed.

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.

Y controls the speed of the 'tape', and thus 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 1 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%.

Output A 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.

Output B 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 output A.

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	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).

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).

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

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).

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

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).

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

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).

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

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).

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

Parameter	Min	Max	Default	Description
0	0	2	0	Trigger Mode.
1	0	1	0	Z Mode.
2	-31	31	31	Output A Attenuverter.
3	-31	31	31	Output B Attenuverter.

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 8V.

E-2 AR Envelope (with push)

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

Parameter	Min	Max	Default	Description
0	0	2	0	Trigger Mode.

See previous algorithm for video.

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. 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.

A & B both output the envelope CV. The maximum envelope level is 8V.

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

E-3 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

Parameter	Min	Max	Default	Description
0	0	2	0	Trigger Mode.
1	0	1	0	Z Mode.
2	-31	31	31	Output A Attenuverter.
3	-31	31	31	Output B Attenuverter.

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).

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 8V, which corresponds to unity gain for the VCA.

E-4 AR Envelope & VCA (with push)

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

Parameter	Min	Max	Default	Description
0	0	2	0	Trigger Mode.

See previous algorithm for video.

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).

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. 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.

Output A is the envelope CV, and output B is the VCA output as mentioned above. The maximum envelope level is 8V, which corresponds to unity gain for the VCA.

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

E-5 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

Parameter	Min	Max	Default	Description
0	0	2	0	Trigger Mode.
1	0	1	0	Z Mode.
2	-31	31	31	Output A Attenuverter.
3	-31	31	31	Output B Attenuverter.

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 8V.

E-6 Dual AR Envelope (with push)

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

Parameter	Min	Max	Default	Description
0	0	2	0	Trigger Mode.

See previous algorithm for video.

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. 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.

A & B are the envelope CV outputs. The maximum envelope level is 8V.

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

E-7 Euro to Buchla Converter

X is 1V/octave input
Y is gate input
Z is tune ± 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.

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

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

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

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 ± 0.5 octaves.

E-8 Buchla to Euro Converter

X is 1.2V/octave input
Y is gate/trigger input
Z is tune ± 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 ± 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	-31	31	31	Output Attenuverter.

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.

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.

A & B both output the envelope CV. Parameter 0 is an attenuverter setting, which can attenuate and/or invert the signal. The maximum envelope level is 8V.

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	-31	31	31	Output Attenuverter.

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.

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.

A & B both output the envelope CV. Parameter 0 is an attenuverter setting, which can attenuate and/or invert the signal. The maximum envelope level is 8V.

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	-31	31	31	Output Attenuverter.

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.

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.

A & B both output the envelope CV. Parameter 0 is an attenuverter setting, which can attenuate and/or invert the signal. The maximum envelope level is 8V.

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	-31	31	31	Output Attenuverter.

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.

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.

Output A is the envelope CV, and output B is the VCA output as mentioned above. Parameter 0 is an attenuverter setting, which can attenuate and/or invert the signal. This applies to both outputs. The maximum envelope level is 8V, which corresponds to unity gain for the VCA.

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

Parameter	Min	Max	Default	Description
0	0	1	0	Direction.
1	1	16	8	Length.
2	0	31	0	Slew rate.
3	-1	31	31	Output attenuator.

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. 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 attenuator for both outputs.

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 MIDI

Parameter	Min	Max	Default	Description
0	0	1	0	Direction.
1	1	16	8	Length.
2	0	15	0	Scale.
3	-1	31	31	Output attenuator.

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. 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.

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 attenuator for the random CV, applied before quantization.

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.

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

Parameter	Min	Max	Default	Description
0	1	16	8	Length.

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. 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.

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

Parameter	Min	Max	Default	Description
0	1	16	8	Length A.
1	1	16	8	Length B.

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. 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.

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. The 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. The in conjunction with the [Silent Way CV Input](#) 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

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.

Following the note name a '-' indicates that the note is below the true pitch (flat) and '+' indicates that the note is above the true pitch (sharp). If the note is within 1 cent of the true pitch, neither '-' nor '+' is shown.

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.

G-6 MIDI Clock

[Video](#) 

X is clock input
Y is run/stop input
A & B are clock outputs
Z is not used

Outputs & Receives MIDI

Parameter	Min	Max	Default	Description
0	-2	12	6	Clock divisor for A.
1	-2	12	-2	Clock divisor for B.
2	0	11	8	Clock divisor for MIDI clock output.
3	0	1	0	Y mode.

Generates clock outputs on A & B from an incoming MIDI clock; generates a MIDI clock from clocks on input X.

An incoming MIDI clock on the disting's MIDI input is used to generate 5V clock pulses on outputs A & B. Each has an independent clock divisor controlled by parameters 0 & 1. The parameter/divisor relationship is as in the following table:

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

Clocks fed into input X (1V threshold) generate MIDI clock on the disting's MIDI output. Parameter 2 sets the divisor relationship between the two clocks.

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

Parameter 3 controls whether input Y is used as a run/stop input:

When parameter 3 is 0 (the default), input Y is not used. The MIDI clock starts as soon as clocks appear on X; the MIDI clock stops if no clocks appear on X for two seconds.

When parameter 3 is 1, input Y is a run/stop input. The MIDI clock starts when Y goes high (over 1V), and stops when Y goes low (below 0.5V).

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	-31	31	0	Transpose.
1	0	24	2	Pitch bend depth.
2	-1	31	-1	Scale.

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

Output A is the pitch CV output; output B is the gate output.

Outputs A & B can also be controlled directly via MIDI CCs 33 & 34 respectively. The CC range of 0-127 is mapped to -5V to +5V.

The received MIDI channel is selected in the settings.

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.

G-8 CV/MIDI

[Video](#) 

X is pitch CV input
Y is gate input
A & B are not used
Z is mod wheel or velocity CV input

Outputs MIDI

Parameter	Min	Max	Default	Description
0	0	31	16	Channel offset.
1	0	1	0	Z mode.

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. If 1, Z generates CC 1 (mod wheel) messages, and the note velocity is fixed at 64.

H-1 Crossfade/Pan

[Video](#) 

A = Mix of X & Y according to Z
B = Inverted mix of X & Y according to Z

Knob recorder enabled

Parameter	Min	Max	Default	Description
0	0	2	0	Crossfade/pan law.

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).

The parameter sets the pan/crossfade law.

- 0 Equal gain Appropriate for crossfading phase-coherent material.
- 1 Equal power Appropriate for crossfading non-phase-coherent material.
- 2 Transition DJ-style crossfade where both sources are at full gain at the 50% position.

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.

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.

H-3 Dual Quantizer (Z scale)

[Video](#) 

A = quantized(X)

B = quantized(Y)

Z chooses scale

Outputs MIDI

Parameter	Min	Max	Default	Description
0	-1	31	31	Input X attenuation.
1	-1	31	31	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. After the name has scrolled across, the display reverts to showing the current algorithm.

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.

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 MIDI

Parameter	Min	Max	Default	Description
0	-1	31	31	Input X attenuation.
1	-1	31	31	Input Y attenuation.
2	-16	16	0	X scale/trigger mode.
3	-16	16	0	Y scale/trigger mode.

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

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.

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-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

Parameter	Min	Max	Default	Description
0	1	16	16	Steps.
1	1	16	4	Pulses (pattern 1).
2	0	16	0	Rotation.
3	0	31	0	Pulse length.

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](#) or [here](#).

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.

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.

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.

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.
6	Z is an additional trigger input which triggers both outputs.

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.

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

Algorithm-specific playlist filename: playlist-rm.txt

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.

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

I-2 Clocked Audio Playback

[Video](#) 

X is retrigger CV
Y is clock
Z selects the sample
A is left audio output
B is right audio output

Algorithm-specific playlist filename: playlist-clocked.txt

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. It plays, forever, 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 to the start 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.

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	-8	8	0	Octave shift.

Algorithm-specific playlist filename: playlist-voct.txt

This algorithm is much the same as the preceding one, 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.

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.

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	31	0	Selects the sample.

Algorithm-specific playlist filename: playlist-zspeed.txt

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 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 the parameter i.e. by turning the S knob.

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

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.

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-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

Parameter	Min	Max	Default	Description
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	31	0	Selects the sample.

Algorithm-specific playlist filename: playlist-endcv.txt

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 the parameter i.e. by turning the S knob.

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.

The recorded WAV files are stereo, 16 bit, 46.875kHz. The files are named sequentially starting with rec0.wav (then rec1.wav, rec2.wav etc.). If there are already recordings on the card, the next available filename will be used; the old files will not be overwritten.

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').

Do not remove the MicroSD card or power off the disting while recording is active, as this could corrupt the card.

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).

K-1 Wavetable VCO

[Video](#) 

X is V/Oct pitch input
Y is wavetable input
Z is tune ± 0.5 octaves
A is wavetable output
B is sub-octave square output

Parameter	Min	Max	Default	Description
0	-31	31	0	Chooses the wavetable.
1	-16	8	0	Octave shift.
2	-31	31	0	Y offset.

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 ± 0.5 octaves.

The Y input controls the lookup point in the wavetable, with a range of $\pm 5V$ covering the whole table.

Output A is the main output of the wavetable VCO. Output B generates a $\pm 8V$ square wave one octave below output A.

Loading wavetables

Wavetables can be loaded in one of two ways: as a single WAV file containing all the waveforms concatenated, or as a folder of WAV files, one per waveform.

The wavetables are 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.

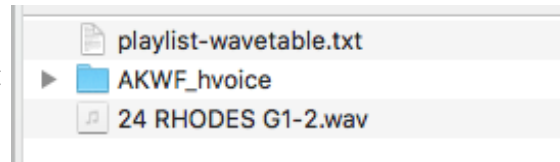
If a folder is specified in "playlist-wavetable.txt", the folder contains the individual WAV files that represent one waveform each. The folder must also contain a playlist (named "playlist.txt"), listing the waveform WAV files in order.

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 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.

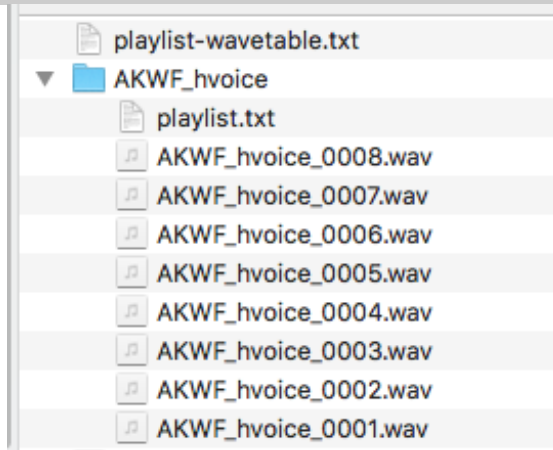
As an example, consider these SD card contents:



"playlist-wavetable.txt" might contain

```
disting playlist v1  
24 RHODES G1-2.wav  
-wavelength=700  
AKWF_hvoice
```

If the folder contents were:



then the folder's "playlist.txt" might contain

```
disting playlist v1  
AKWF_hvoice_0001.wav  
AKWF_hvoice_0002.wav  
AKWF_hvoice_0003.wav  
AKWF_hvoice_0004.wav  
AKWF_hvoice_0005.wav  
AKWF_hvoice_0006.wav  
AKWF_hvoice_0007.wav  
AKWF_hvoice_0008.wav
```

etc.

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	31	0	Scale.
1	-1	31	31	Input X attenuation.
2	-2	31	31	Input Y attenuation.
3	-31	31	0	Transpose.
4	0	31	0	Offset.

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.

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 -1 the input is completely attenuated; at 31 the input is unattenuated.

If parameter 2 is set to -2, 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.

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

Parameter	Min	Max	Default	Description
0	0	31	15	Size.
1	0	31	23	Feedback.
2	0	5	1	Character.
3	-1	31	8	Lowpass 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.

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.

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

Parameter	Min	Max	Default	Description
0	0	31	15	Size.
1	0	31	23	Feedback.
2	0	5	1	Character.
3	-1	31	8	Lowpass 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.

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.

L-3 Dual Reverb

[Video](#) 

A is X plus reverb

B is Y plus reverb

Z is wet/dry

Parameter	Min	Max	Default	Description
0	0	31	15	Size.
1	0	31	23	Feedback.
2	0	5	1	Character.
3	-1	31	8	Lowpass 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.

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.

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	31	8	LFO depth.
1	-31	31	0	Y offset.
2	-31	31	0	Feedback.
3	-1	31	-1	Lowpass filter.

A stereo chorus 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.

Parameter 2 controls a feedback loop around the effect.

Parameter 3 applies a low pass filter to the chorus input. It does not affect the dry portion of the signals.

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	31	8	LFO depth.
1	-31	31	0	Y offset.
2	-31	31	0	Feedback.
3	-1	31	-1	Lowpass filter.

A mono chorus 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.

Parameter 2 controls a feedback loop around the effect.

Parameter 3 applies a low pass filter to the chorus input. It does not affect the dry portion of the signal.

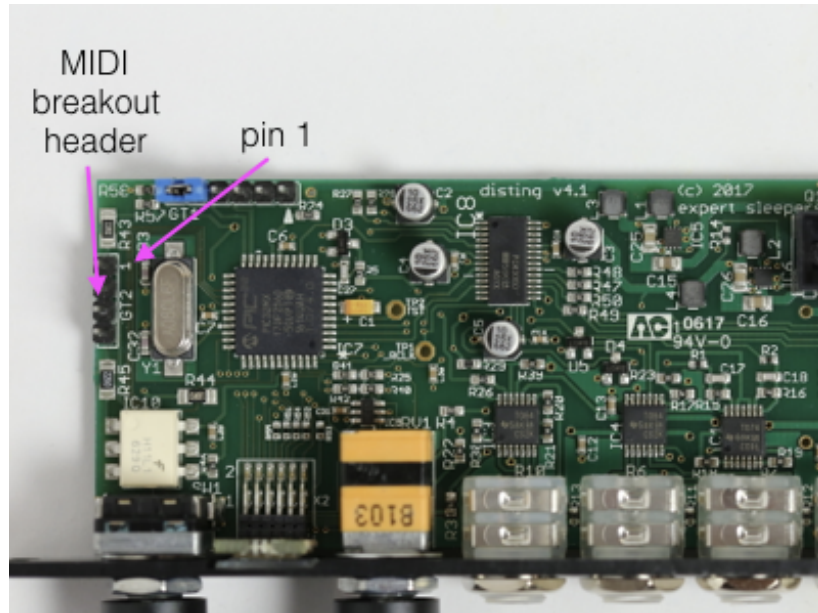
MIDI I/O

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.

MIDI channel	Message	Function	Note
<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>	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.
- Unzip the download.
- Copy the file 'image.hex' 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 (which is how many of them ship, so chances are you don't need to do anything about this).

Acknowledgments

The font used in the disting mk4 is an adapted version of 'Tom Thumb' by Robey Pointer. More on the font [here](#).