expert sleepers disting NT

User manual

Version 1.4

And you may ask yourself, "How do I work this?"

- Talking Heads, "Once in a Lifetime"

Table of Contents

Introduction

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

A note on navigating this manual

When one part of the manual refers to another, it may say something like "see Settings, below". In such cases the word 'below' (or 'above') is a hyperlink, and can be clicked on. Try it.

It seems very technical

This is intended as a reference manual, not a tutorial. Please visit our YouTube channel for some friendlier demos, or just ask for help.

Where to get help

Email, forum, and social media links can be found at the bottom of every page on our website[1]. We also have a Discord server[2].

We prefer support requests to be made through the forum, if possible - failing that, by email.

If you're lucky enough to have a local bricks and mortar modular store, please visit them, try things out, and get help there. And most of all, buy your modules from them too.

Preamble

The disting NT is the latest in the line of modules from Expert Sleepers that started with the original disting in 2014, which performed one of 16 very different functions at the turn of a knob. This concept has evolved through the disting mk3 (2015) - which added the possibility of firmware updates, and so a continually improving selection of functions - and the disting mk4 (2017) - which added a display, and brought the MicroSD card slot onto the front of the module, emphasising its increased importance for functions such as sample playback - to the disting NT's predecessor, the disting EX (2020). The EX (finally) added an OLED display, and a much more powerful CPU platform, taking the functions it could perform to the next level.

Ever since the mk3 was released, the modules' firmwares have been continuously updated with new features, making them even more useful and better value.[3]

We aim to continue this legacy with the disting NT.

Whereas the disting EX built directly on the selection of functions in the mk4, and then added some of

---

1 https://www.expert-sleepers.co.uk/

2 https://modwiggler.com/forum/viewtopic.php?t=287914

3 Some stats - the mk3 had 10 major updates in 2 years; the mk4 has (so far) had 28 major updates in 7 years; the EX has (so far) had 25 major updates in 4 years.

its own, with the disting NT we've taken everything back to square one. We anticipate (eventually) adding every function ever offered by the previous distings, but every algorithm has been reconsidered and reappraised in the light of the possibilities offered by the new hardware platform. Often things have been streamlined and simplified - for example, a disting EX algorithm that offered a polysynth, a chorus, and a delay would be implemented on the NT as three separate functions, which you could combine as you choose - each function therefore being smaller and easier to deal with.

While the exact future of this module is yet to be written, one thing is certain - it will be driven by user feedback. We have an active user community on the [ModWiggler forum](https://www.modwiggler.com/forum/viewforum.php?f=35)[4], and suggestions (and we'll admit, bug reports) left there have often made it into the firmware. So we strongly encourage you to get in touch! Let us know what the disting NT can be for you.

What does it do?

The module takes audio and/or CV input, runs a bunch of software processing, and generates audio and/or CV output. It does this continuously.

The chunks of software processing are called *algorithms*. You can think of them much as you would think of plug-ins in your DAW. The module has a number of algorithms to choose from, and you can add the ones you want in a big list and the module will run them all.

A collection of algorithms, their various parameters, and associated data such as MIDI mapping, together make up a *preset*.

Like plug-ins in a DAW, we use 'algorithm' to mean both the abstract *type* of processing you want to run, and a particular *instance* of that algorithm in a preset. For example, you might say "I'll load a reverb plug-in on this track" meaning the *type* of plug-in you want to load, and also "I changed the decay time on the reverb plug-in" to refer to the particular *instance* of the reverb plug-in.

What is it not?

The disting NT is not a "virtual modular". There are no virtual cables flying around. It simply collects together a variety of functions that are useful in the context of a modular synthesizer and makes it possible to use a number of them at once.

It is not a "patch player". It is not the intention that you do all the complex setup on a computer and then copy it over to the module. The module is designed to be used *in a modular*, with a UI that makes it fluid to do everything in the module itself.

It is not only useful via MIDI. The large number of inputs (and outputs) makes it possible to use it entirely via CV, interfacing with the rest of your modules in a natural way. (That said, it does also make a really good MIDI polysynth.)

---

4   https://www.modwiggler.com/forum/viewforum.php?f=35

Getting started

First, carefully read the installation instructions below.

Then, at least skim the 'Navigation' section.

This will enable you to load some of the example presets from the MicroSD card.

You might like to watch [this quickstart video](#)[5] before going much further, which was made to help get new users up and running.

Please also check if you have the latest firmware, which can be found on our website [here](#)[6].

Example presets

Some example presets are included on the MicroSD card supplied with the module, but by the time you read this there will almost certainly be more available on our [GitHub](#)[7]. Please look there for the latest examples, and for documentation of how the presets are put together, which can be quite informative.

Installation

House the module in a Eurorack case of your choosing. The power connector is 16-pin [Doepfer standard](#)[8]. If using the power cable supplied with the module, the red edge of the cable is furthest from the top edge 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.

Please observe ESD (electro-static discharge) precautions when handling the module (and indeed, any module).

---

5   https://www.youtube.com/watch?v=EfkrNHkTKF0

6   https://expert-sleepers.co.uk/distingNTfirmwareupdates.html

7   https://github.com/expertsleepersltd/distingNT/tree/main/presets

8   http://www.doepfer.de/a100_man/a100t_e.htm

Please do not operate the module without it being securely fastened in a case.

Physical dimensions

The disting NT is 22HP wide and 25mm deep (including the depth added by the included power cable).

Power requirements

The disting NT draws 211mA on the +12V rail, and 96mA on the -12V rail, when idling. The actual power consumption is heavily dependent on how many input and output socket LEDs are lit. Each fully illuminated socket draws about 4mA (from the +12V rail or -12V rail depending on the polarity), so you could in theory pull another 72mA, but that is highly unlikely to occur in actual use - you'd have to somehow feed a steady 10V into every input socket and be running algorithms that were outputting a steady 10V from every output. When budgeting for case power it's probably safe to split that 72mA between the two rails and say it draws 247mA on the +12V rail and 132mA on the -12V rail. But really, if that 36mA is enough to put your PSU over the edge, you need a bigger PSU anyway.

The disting NT does not use the 5V rail.

Connecting expansion modules

Turn off the power before connecting or disconnecting expansion modules.

ES-5

Connect an ES-5[9] module via the header on the back of the disting NT marked J11 ES-5, using the 10-way cable provided with the ES-5. The red stripe should be oriented down on both modules, as shown in the photo below, and in the ES-5 user manual[10]. At the ES-5 end, connect the cable to the header marked "GT7/To ES-3".

The disting NT can use an attached ES-5 to output a considerable number of extra CVs and gates, where supported by a particular algorithm. Please note however that in the current firmware no algorithm yet makes use of the ES-5.

---

9   https://www.expert-sleepers.co.uk/es5.html

10 https://expert-sleepers.co.uk/es5usermanual.html

MIDI

Connect a MIDI breakout to the header on the back of the disting NT marked J10 MIDI.

MIDI can be used to remotely control the algorithm parameters, to share clocks, and to play notes in the synthesizer algorithms etc. The disting NT can also send MIDI to other devices, for example to sequence them, or simply to provide parameter feedback to a MIDI controller.

The image below shows how the header is wired, including pin numbers for 5-pin DIN connections, and tip/ring indications for a Type A TRS MIDI jack.

See below for more details on hooking up MIDI using the included breakout module.

I2C

A device or module that communicates via I2C can be connected via the header marked J8. The SDA and SCL pins are marked; the third pin is ground.

Like MIDI, I2C can be used for remotely controlling algorithm parameters and playing notes. While less common than MIDI, it is the control mechanism of choice for the whole [Teletype](https://monome.org/docs/teletype/)[11] module ecosystem.

CVM-8

If you feel the need for more CV inputs on the disting NT, please take a look at the [CVM-8](#)[12], which

---

11 https://monome.org/docs/teletype/

has eight inputs and can connect via MIDI, I2C, or the Select Bus.

Using the included breakout module

The disting NT is supplied with a "Tiny MIDI Breakout" (aka TMB), a pre-existing Expert Sleepers product, which is simply six TRS jack sockets on a 2HP panel. The product page for the TMB is here[13]. In the box are also some jump cables (aka "DuPont cables" or "Arduino cables") which you can use to connect the two.

If you need more, or longer, cables they are easily found online.

Each of the six TRS sockets is wired to a 3-pin header on the back of the PCB, like so:
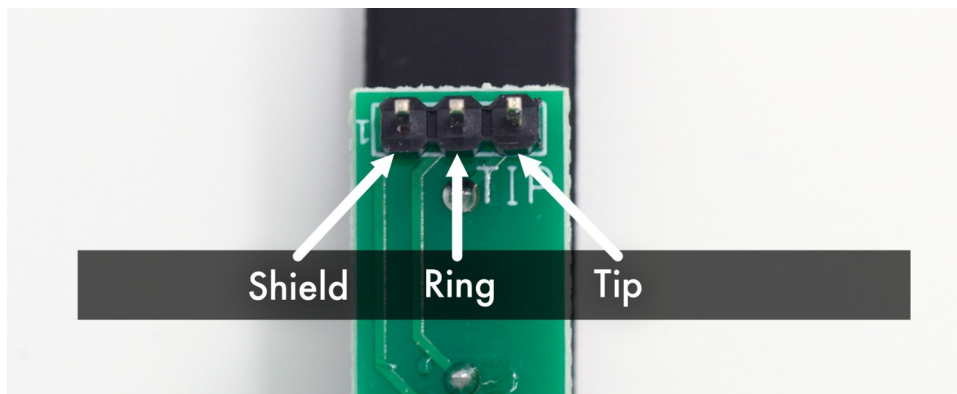
Use the jump cables to connect the relevant pins of the TRS sockets to the pins of the breakout headers on the disting NT that you would like to use.

For example, to use one of the sockets as a TRS MIDI input, use two cables to connect the Tip and Ring of the socket to the MIDI In Tip & Ring as shown above.

There is a video showing how to connect the breakout here[14].

The diagrams below shows typical connections when using the breakout for MIDI and audio. The wire colours match those shown in the video.

---

12 https://expert-sleepers.co.uk/cvm8.html

13 https://expert-sleepers.co.uk/tinymidibreakout.html

14 https://www.youtube.com/watch?v=NxiFBvlkGgo

MIDI In

MIDI Out

Out 7

Out 8
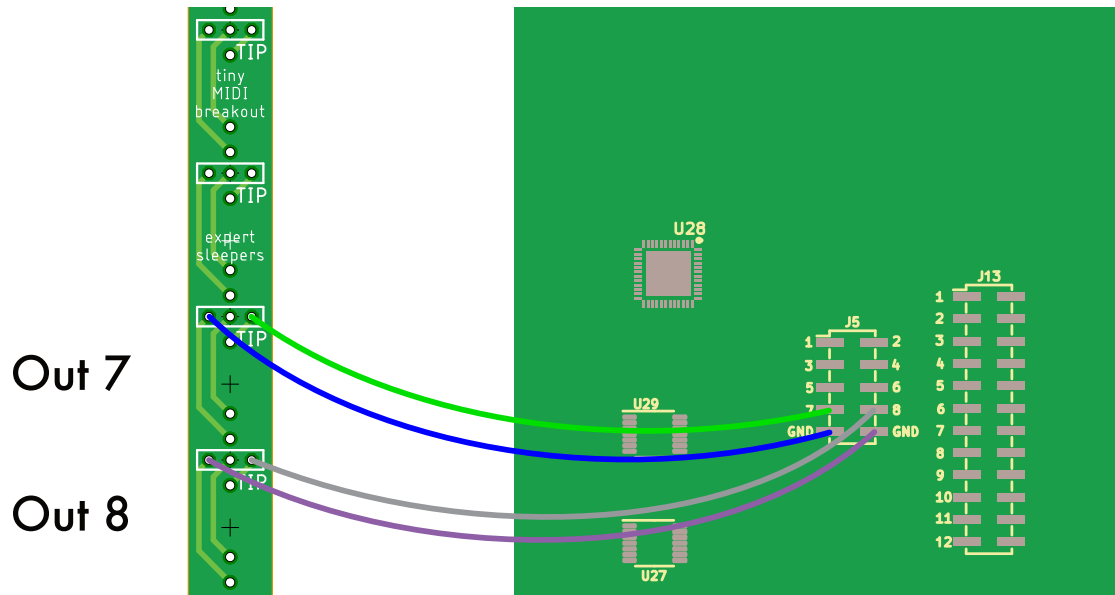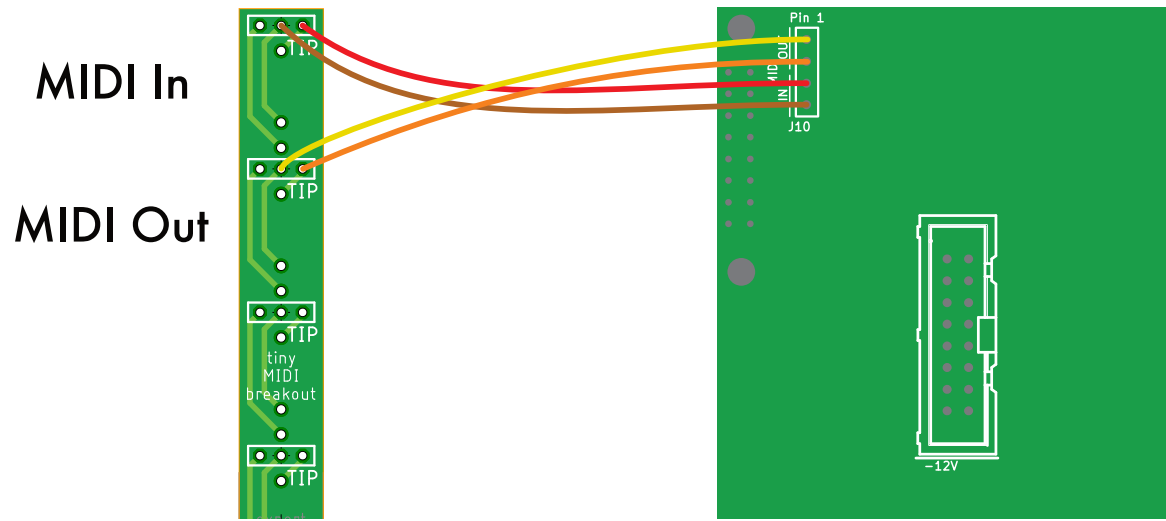
Jumpers/switches

The jumper on the board labelled J12 SELECT BUS connects the module's Select Bus circuitry to the power bus's CV line. Fit the jumper if you want to use the Select Bus; remove it if you use the CV bus for its originally intended purpose.
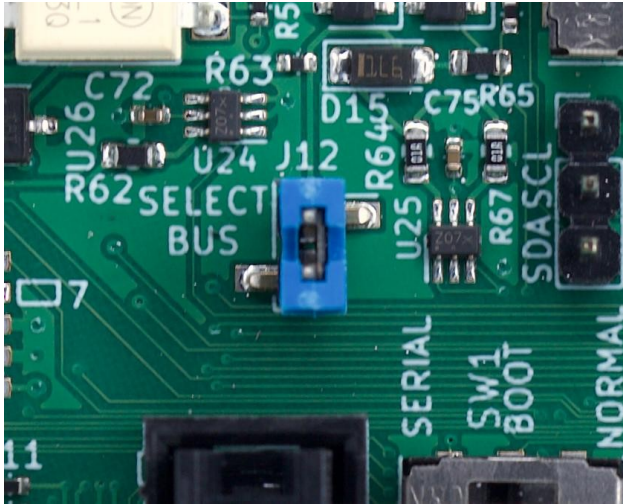
Use the switch marked SW8 I2C PULL UP to enable or disable the I2C pull-up resistors (image above). The pull-ups are enabled when the switch is in the 'up' position (nearest the top of the PCB). At least one device on the I2C bus needs to have pull-up resistors enabled. It may work fine with more than one set of pull-ups enabled, or it may not.

The switch marked SW1 BOOT can be used to force the module into serial bootloader mode. Normally you would leave the switch in the "Normal" position (to the right) and use the module's menu to enter bootloader mode. If however this is not possible, put the switch into the "Serial" position (to the left) to force it into bootloader mode. Note though that you will then have to return it to "Normal" to boot normally after installing new firmware. See below for a description of the usual firmware update procedure.
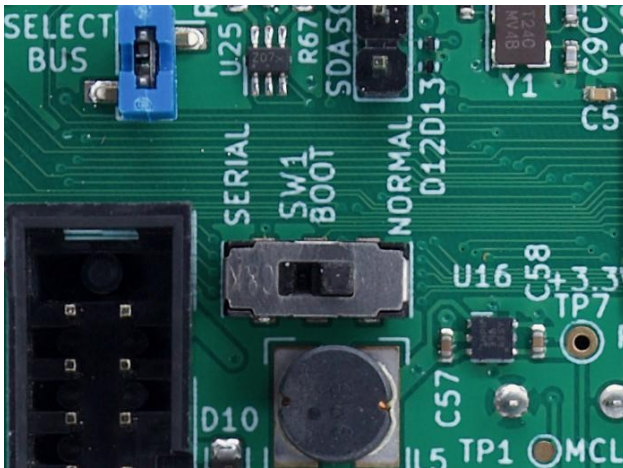
Controls

The disting NT has three pots, two encoders, and four pushbuttons. The pots and encoders also have a pushbutton action.

The controls are unlabelled, on the basis that their function can be and is completely redefined in software. Popup help can be displayed on the display to help you keep track.

In this manual, we will simply refer to "the left pot", "the centre pot", and "the right pot"; similarly, "the left encoder" and "the right encoder". The pushbuttons are referred to as buttons 1-4, ordered from left to right.
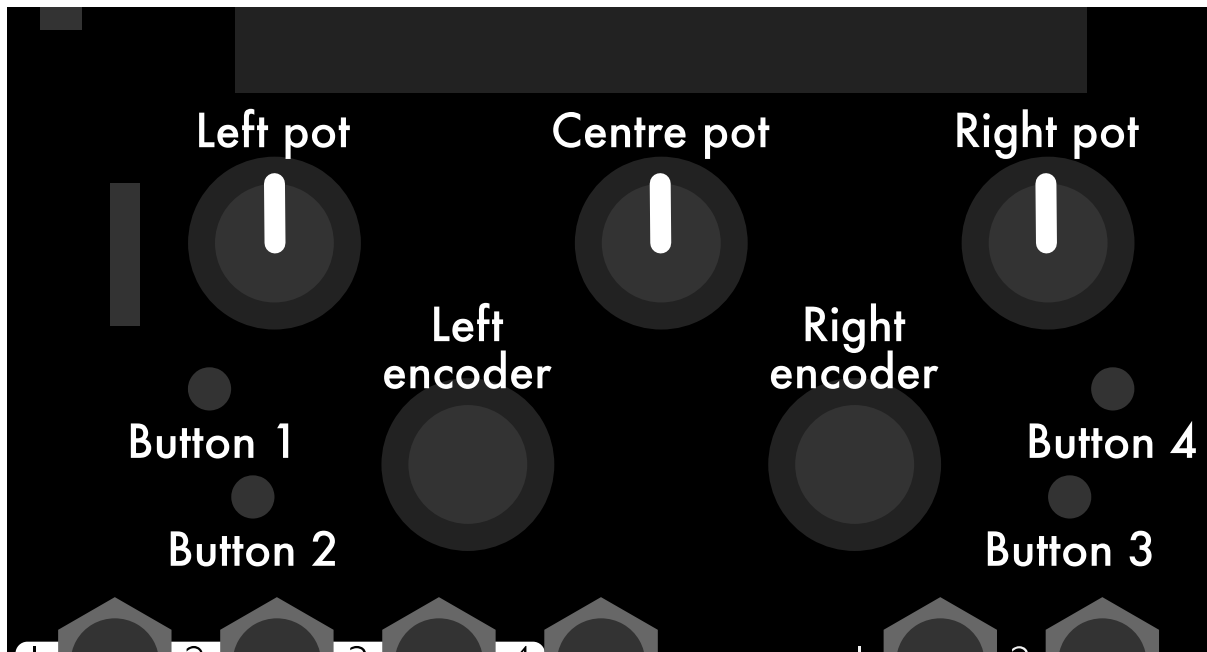
There is a setting you can change to order the buttons in the opposite direction i.e. from right to left. See below.

Navigation

There are two states that the module UI will be in most of the time: the algorithm view, in which you navigate the various active algorithms and interact with their parameters; or the menu system.

Pop-up help is available, indicating the function of each control in a small area at the bottom of the screen. If you've disabled it in the Settings, you can still view the help at any time by holding down button 3.
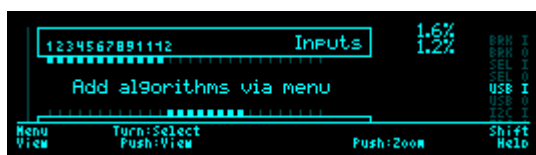
In the pop-up help, words at the extreme left and right sides refer to the pushbuttons. Other help is, as far as space will allow, lined up above the encoder or pot it refers to. In the image above, 'Menu', 'View', 'Help', and 'Shift' refer to pushbuttons 1-4 respectively. 'Turn:Select' and 'Push:View' both refer to the left encoder.

Menus

You can easily tell when you're in the menu system because there will be a dotted line border around the screen.

Button 1 enters the menu system; it also immediately exits the menu system if pressed while you're anywhere in the menus.

The left encoder navigates the menu: turn it to select a submenu or menu item, and press it to descend one level. Button 2 is 'back' i.e. it goes up one menu level. If you keep pressing button 2 you will eventually leave the menu entirely.

Where a menu allows you to change a value, or select from a list of items, you can in general do this either by turning the right encoder, or by pressing the left encoder to descend one more menu level and then turning the left encoder.

There is a setting to swap the function of the two encoders, if you find it preferable to primarily drive the menu with the right encoder. See below.

File browser

Some menus, for example the menu for loading presets, require you to browse for a file on the MicroSD card.

In the file browser, use the left encoder to move up and down the list of files, to select items, and to enter folders. Press the right encoder to leave a folder and go up one level of the folder hierarchy.

Editing text

Some menus, for example the menu to set the preset name, allow you to edit text.

Turn the left encoder to move the cursor, and the right encoder to change the character at the cursor position.

Pressing the right encoder advances to the next character in the list ' ', '0', 'A', 'a'.

Press the left encoder to accept the text and return to the menu above.

Algorithms view

This is where you'll spend most of your time interacting with the module. There are two primary display modes - the algorithms overview, where you can see all the running algorithms and the signals flowing between them, and the single algorithm view, where you edit the parameters of an algorithm.

Algorithms overview

Here you can see all the algorithms running, in a list starting with the module inputs at the top, and flowing down to the module outputs at the bottom.
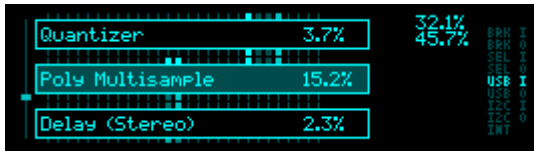
Turning the left encoder or the left pot scrolls through the algorithm list.

The 'current algorithm' is the one highlighted, in the centre of the screen. This is the algorithm that will be the target of any menu commands e.g. Move up/down. It is also the algorithm that will be live for editing when you switch to the single algorithm view.

There is a 'scroll bar' graphic on the left side of the screen showing where you are in the list.

Holding button 4 and turning the left encoder moves the current algorithm up and down in the list, exactly as if you'd used the 'Move algorithm up/down' menu.

Press the right encoder to switch to a different view showing the input and output signals of the current algorithm. Turning the encoder or pots still scrolls through the list as before.

Press the left encoder or left pot to switch to the single algorithm view.

Single algorithm view

This is where you edit the parameters of an algorithm. There are two modes for this view: one which shows only parameters, arranged in pages, and one which shows a single parameter and devotes the rest of the screen to a display specific to the algorithm in question.

In either mode, navigation is exactly the same.

Turn the left pot to select the page of parameters to view.

Turn the centre pot or the left encoder to select a parameter within the current page.

Turn the right pot or the right encoder to change the parameter value.

The pot uses a form of 'soft takeover' rather than jumping to the value that corresponds directly to the knob position. When a new parameter is selected for editing, turning the pot clockwise will always increase the value, and turning it anticlockwise will always decrease the value, wherever the pot happens to be.

Pressing the right pot causes the parameter value to jump directly to the position of the knob,

bypassing the soft takeover. This can be useful, say, if you want to set a number of parameters to a similar value. For example, if you wanted to set a number of mixer channel gains to the same value, you could turn the right pot to set the first value, then step through the other channels with the centre pot and press the right pot to jump the values to the right area, without having to turn the pot a considerable distance for each parameter.

Pressing the right encoder sets the parameter to its default value, with some exceptions:

- For parameters with a 'Confirm' function, for example the wavetable selection of the Poly Wavetable algorithm, pressing the right encoder confirms the new value.
- For string parameters, pressing the right encoder switches to a view where you can edit the string.
- For parameters specified in dB, pressing the right encoder sets the parameter to its default, or if the parameter is already at minimum (usually -∞dB), to 0dB.
- For parameters specified in %, if the parameter is at 0%, pressing the right encoder sets it to 100%, and if it's at 100%, to 0%.

Pressing button 4 is a shortcut to the Mappings menu. If you prefer, you can change this in the Settings to go directly to one of the CV, MIDI, or I2C Mappings submenus instead.

Press the left encoder or the left pot to switch to the algorithm overview.

Sometimes you may see parameters "greyed out". This indicates that the parameter is currently unused, most likely because of some other parameter selection. For example, the tempo parameter of the Clock algorithm is greyed out if the algorithm's internal clock is not being used.
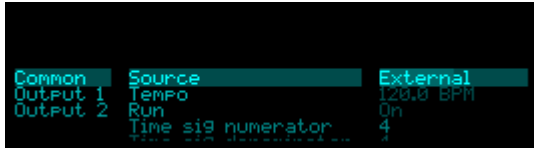
Inputs and outputs

The disting NT's input and output jack sockets are illuminated, lighting red for positive voltage and blue for negative voltage. (Audio-rate signals appear purple, since you see a rapid alternation of positive and negative.)

The sockets are 3.5mm TS jacks. Do not use TRS cables.

The inputs and outputs are all DC coupled.

All of the inputs and outputs are also available on headers on the rear of the module. The outputs are on J5 and the inputs on J13:
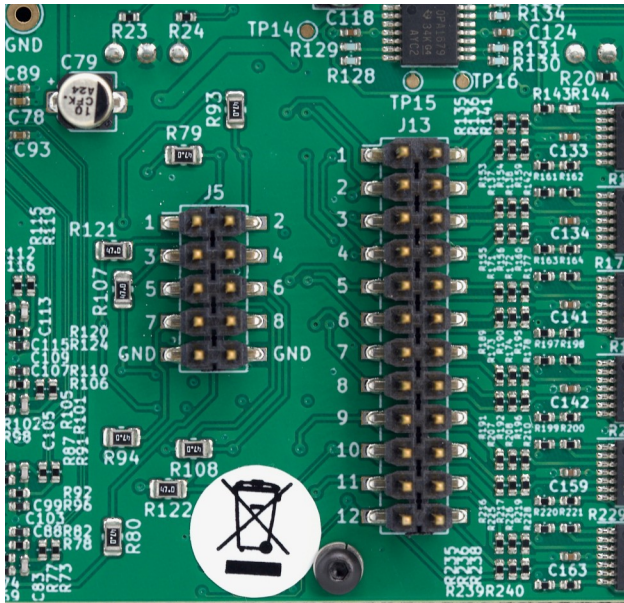
Inputs

The twelve sockets on the left of the module (numbered in black on white) are the inputs.

The inputs can operate at either modular or line level, selected by jumpers. See the image above. The left pins on the J13 header correspond to the front panel sockets, and operate at modular level. The right pins are line level. If you fit a jumper across a pair of pins, this shorts the input socket to the line level input i.e. it makes the front panel socket a line level input.

Mode	Maximum voltage before clipping	Input impedance
Modular	±12V	101kΩ
Line	±3.5V (+10dBu)	29kΩ

Outputs

The six sockets on the right of the module (numbered in white on black) are the outputs.

The outputs have a maximum range of ±11V.

There are an additional two outputs available via the J5 header on the rear of the module. See the image above. The pins of the header are numbered according to the output numbers; pins 1-6 correspond directly to the front panel output sockets, while pins 7-8 are the two additional outputs. Ground ('GND') connections are also exposed on this header.

One possibility is to connect the additional two outputs to two sockets on the included breakout module - or even to connect them to a single socket, as a stereo TRS output.

USB

The module has a type C USB socket to the left of the display. The disting NT is a High Speed USB 2.0 device. It is not a USB host.

The module does not draw power from the USB connection.

If you are planning on using the USB socket more than occasionally, we recommend buying a cable with a right-angled plug, to keep the cable out of the way of the display.

The disting NT is a MIDI Class and Audio 2.0 Class compliant device. As such it needs no drivers when used with macOS, iOS, and Linux. An audio driver for Windows is available via website[15].

Currently the module's USB audio operation is fixed at 48kHz. This may change - let us know if support for other sample rates is a priority for you.

The USB socket can also be used to mount the module's MicroSD card as a removable drive on your computer - see below.

MIDI connections

The disting NT can send and receive MIDI via a breakout header, via the Select Bus, and via USB.

MIDI breakout

The disting NT's MIDI breakout header is a four pin connection, exactly the same as on all of our other modules with MIDI - at the time of writing, these are the disting mk4 and EX, the FH-2, the ES-9, and the CVM-8. The header can be directly connected to DIN or TRS sockets - for example, our own MIDI breakout modules, or the sockets built into certain Eurorack cases.

The wiring is as follows:

Breakout pin	MIDI DIN pin
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 specification[16].

As well as connecting the module to a breakout, you can also use the header to connect directly to another of our modules. In this case, connect pin 1 on one module to pin 4 on the other, pin 2 to pin 3, and so on, so that the outputs connect to inputs and inputs connect to outputs.

---

15 https://expert-sleepers.co.uk/downloads.html

16 https://midi.org/5-pin-din-electrical-specs

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 Tempi & René. It is essentially a MIDI connection piggybacked on what was originally conceived as the CV bus, which is one of the lines on the power bus that connects all modules in the system.

The 'Select Bus protocol' specifies certain messages to be used for preset saving and recall, which are combinations of MIDI program change and CC messages. The disting NT currently does not implement this protocol at all - it simply uses the Select Bus as a MIDI connection. It is particularly convenient since no additional cables are required - the modules are connected simply by virtue of being on the same power bus.

USB MIDI

The disting NT is a MIDI Class compliant device, as noted above.

MicroSD card

The card slot is just to the left of the left pot. Insert cards with the exposed contacts facing right. It is safe to insert a card after the module is powered up.

Do not remove the card while the module is writing to it, or data corruption could result.

If you do remove the card and re-insert it while the module is powered on, you will need to manually re-mount it via the menu. See the Misc menu, below.

Card format

The disting NT requires the card to be formatted as FAT32 or exFAT. If you need to format a card, we recommend that you use the formatting tool provided by the SD Association, which you can download [here](17).

Card content

Nothing on the card is actually required for the module to operate; you can insert a completely blank card (or not insert a card at all) and the module will function.

Note though that, in the current firmware, a card is required for storing presets, so you will probably want to keep a card installed at all times.

The card provided with the module is preloaded with some content created by ourselves and our partners, including

- [Spitfire Audio](18) - multisamples, including the iconic Soft Piano.

---

17 https://www.sdcard.org/downloads/formatter/

- [Goldbaby](#)[19] - samples, mostly drums.
- [Adventure Kid](#)[20] - wavetables.

Card layout

Data on the card is organised into a number of top level folders to keep things tidy. While some functions (e.g. preset loading) allow you to select files from anywhere on the card, many of the core algorithm functions require data to be in a specific place - e.g. the algorithms that play samples require all the samples to be in the 'samples' folder.

The standard folders, as found on the card supplied with the module, are as follows:

FMSYX	FM banks (SysEx format, usually .syx)
kbm	Scala keyboard map files (.kbm)
MIDI	Subfolders, each containing MIDI files (.mid)
MTS	MIDI Tuning Standard SysEx dumps (.syx)
presets	The default folder into which presets are saved. Can be changed in the Settings menu.
recordings	The default folder for audio recordings made by the Audio Recorder algorithm. Can be changed in the Settings menu.
samples	Subfolders, each containing audio files (.wav)
scl	Scala scale files (.scl)
ui_scripts	Scripts for custom UIs (.lua)
wavetables	Wavetables, either as single file wavetables, or subfolders of single cycle waves (.wav, either way)

FM banks

The 'FMSYX' folder on the card contains Yamaha DX7 format voice banks in SysEx format. Such files usually have the extension '.syx' and will be exactly 4104 bytes in size. Literally hundreds, if not thousands, of voice banks can readily be found online, or you can make your own with various software tools designed as DX7 patch editors.

In the current firmware, the FM banks are primarily used by the Poly FM algorithm, and up to 100 banks can be used at once.

---

18 https://www.spitfireaudio.com

19 https://www.goldbaby.co.nz

20 https://www.adventurekid.se

Scala files

Many algorithms on the disting NT support microtuning via the common [Scala](https://huygens-fokker.org/scala/)[21] file formats. Scale files (.scl format) go in the 'scl' folder and keyboard map files (.kbm) format go in the 'kbm' folder.

Keyboard map files are not required - the algorithms can apply microtuning from a .scl file only - but are supported for more detailed control over tone mapping if desired.

Scala files can be generated by the Scala application, by another tool that writes Scala-format files (for example, [Scale Workshop](https://sevish.com/scaleworkshop)[22]), or even written by hand in a text editor.

MTS (MIDI Tuning Standard)

MTS can be used live, over a MIDI connection, or via SysEx dump files. The disting NT supports reading MTS bulk SysEx dumps from the card. Such files will be exactly 408 bytes in size, and should be put in the 'MTS' folder.

MIDI song files

Some algorithms, notably the MIDI Player, support MIDI song files. Such files typically have the extension '.mid'.

Song files should be put in subfolders of the top level 'MIDI' folder (i.e. not directly within the 'MIDI' folder itself).

A good technical description of the MIDI file format is [here](http://www.music.mcgill.ca/~ich/classes/mumt306/StandardMIDIfileformat.html)[23]. Most DAWs export MIDI files in some form – for example, Ableton Live exports MIDI clips as single track (format 0) MIDI files, while Logic Pro will export multi-track (format 1) files. And of course, the internet is full of sites where you can download complete songs or looping patterns in MIDI format.

Those so inclined might find it useful to create MIDI files programmatically, which can be easily done, for example, in Python using libraries such as [mido](https://mido.readthedocs.io/en/stable/)[24].

The current firmware supports up to 1000 folders of 1000 files each.

Presets

The disting NT uses [JSON](https://en.wikipedia.org/wiki/JSON)[25] format for presets. As such they are human readable, and if necessary can be manipulated using a simple text editor, though this is not something we expect most users to do.

Presets can be loaded from anywhere on the card, so you can if you wish organise them into folders. However they are always saved to a location that is chosen in the Settings, which defaults to a top level folder named 'presets'.

---

21 https://huygens-fokker.org/scala/

22 https://sevish.com/scaleworkshop

23 http://www.music.mcgill.ca/~ich/classes/mumt306/StandardMIDIfileformat.html

24 https://mido.readthedocs.io/en/stable/

25 https://en.wikipedia.org/wiki/JSON

Samples

Algorithms that play samples look for them in the 'samples' folder, which should contain subfolders, each of which contains audio files in standard WAV format.

Mono or stereo files are supported, at any sample rate. Supported bit depths are 8 bit, 16 bit, 24 bit, and 32 bit float.

The current firmware supports up to 1000 sample folders and a total of 10,000 samples, with a maximum of 1000 files per folder.

Q: Why is there a maximum at all? Can't I just load a file from the card and play it?

A: Scanning the card and pre-preparing lists of samples means that sample selection can be mapped to a CV (or MIDI CC etc.). One of the design goals of the disting series has always been to integrate well into a modular synth, not simply to be physically a Eurorack module.

The card is scanned for samples at power on, and most of the salient data cached back onto the card, so a slow scan of each folder is only done when it's added or changed.

Avoid turning off the module while the scan is in progress, since it's potentially writing to the MicroSD card.

Sample naming

The sample naming convention is the same as for our [disting EX](https://expert-sleepers.co.uk/distingEX.html)[26] module, so if you have libraries prepared for that module they can be easily shared by the disting NT. Look at the samples on the card provided with the module for some examples. The filenames are examined for various patterns, all starting with an underscore ('_'), to set various properties of the files. These are as follows:

Key	Example	Description
_<note name>	piano_A2.wav piano_C#5.wav	Sets the natural note pitch, that is, the pitch of the audio in the sample as recorded. Use naturals and sharps ('#') only, not flats.
_SW<note number>	piano_A2_SW40.wav	Sets the sample's switch point when used in multisample algorithms. The 'note number' is the MIDI note number, in decimal. The switch point specifies the lowest pitch that will use the file.
_RR<number>	snare_RR1.wav snare_RR2.wav	Links files as round-robin variants of the same sample.
_V<number>	snare_V1.wav snare_V2.wav	Sets up a velocity switched sample. The lowest-numbered _V is used for the lowest velocity.

---

26 https://expert-sleepers.co.uk/distingEX.html

If round-robins and velocity switches are combined, the velocity switch comes first e.g.

    snare_V1_RR1.wav, snare_V1_RR2.wav
    snare_V2_RR1.wav, snare_V2_RR2.wav

In other words, each velocity layer can have round-robins.

Automatic 'natural' values

If sample files are not assigned a natural pitch (for example, a folder of drum samples), they are assigned values from 48 upwards, so playing a keyboard chromatically from C3 will play a new sample on each semitone.

Automatic 'switch' calculation

If the switch point for multisample files is not explicitly specified, the disting NT calculates sensible defaults, as follows.

If the gap between neighbouring samples is at most 3 semitones, the switch is set so that the higher sample is pitched down within the gap.

For larger gaps, the lower sample is pitched up over half the remaining range.

For example:

Gap between samples (semitones)	Behaviour
1	Higher sample stretched down over 1
2	Higher sample stretched down over 2
3	Higher sample stretched down over 3
4	Higher sample stretched down over 3; lower sample stretched up over 1
5	Higher sample stretched down over 4; lower sample stretched up over 1
6	Higher sample stretched down over 4; lower sample stretched up over 2
	Etc.

Converting from other sample library formats

There are tools in our Github repository[27] for converting the output of Logic Pro's autosampler and of DiscoDSP's Bliss to a format readable by the disting EX (and by extension the disting NT).

We also have a tool for converting SoundFont®s which you can find here[28].

---

27 https://github.com/expertsleepersltd/distingEX_tools

28 https://github.com/expertsleepersltd/sf2_to_dex

Our format is supported by the very capable [ConvertWithMoss](#)[29], which supports many other formats.

Creating your own samples

Creating new sample sets is simply a matter of collecting WAV files in a folder on the MicroSD card and naming them appropriately, if required (see above). Non-pitched samples e.g. drums need no special naming.

There are tools available to automate the sampling of existing instruments, for example DiscoDSP's Bliss as already mentioned.

The disting NT's own Auto-sampler algorithm will automatically sample any instrument it can trigger via CV/gate or MIDI. The files it records to the MicroSD card are named correctly for use by the multisample playback algorithms.

Loop markers in WAV files

The disting NT supports reading loop information embedded in the WAV file. If this information is not present, any playback that loops the sample simply loops the whole file.

The disting NT looks for 'cue ' chunks and 'smpl' chunks. Which of these your files contain will depend on the authoring software.

When interpreting a 'cue ' chunk, loops are inferred either from markers or regions, as follows:

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

When interpreting a 'smpl' chunk, the first loop in the chunk is used, and any others are ignored.

There are some example multisamples with loop points in our GitHub [here](#)[30].

Round robin mode

Algorithms that have a 'Round robin mode' parameter allow you to choose how round-robin variants of the same sample are chosen. The options are as follows:

Sequential	The round-robins are played in numerical order.
Random	The round-robins are played in a random order.

---

29 https://mossgrabers.de/Software/ConvertWithMoss/ConvertWithMoss.html

30 https://github.com/expertsleepersltd/distingNT/tree/main/samples

Random2	The round-robins are played in a random order, except that the same round-robin cannot be played twice in a row.
Random3	The round-robins are played in a random permutation, then another random permutation, and so on.

Wavetables

A number of algorithms (e.g. Dream Machine and Poly Wavetable) include wavetable-based oscillators. These look for wavetables inside the 'wavetables' folder.

Within the 'wavetables' folder, wavetables can take one of two forms: a single WAV file containing all the waveforms concatenated, or a folder of WAV files, one per waveform. All of the examples included on the card use the latter format.

Waveform WAVs can use any supported bit depth. The sample rate is unimportant, since the file is assumed to contain exactly one cycle and so can be pitched arbitrarily.

When using a folder of single waveform WAV files, each file should be the same length. For example, in the 'AKWF_0001' wavetable on the MicroSD card each WAV file is 600 samples long.

When using a single concatenated WAV file, the disting needs to know how many frames in the file make up one waveform. It does this as follows:

- If the filename ends in a dash followed by a number, that number is taken to be the number of frames in one waveform. For example, if the filename is "awesome-256.wav", then the disting will interpret the file as having 256 frames per waveform.
- If the number of frames in the file is a multiple of 2048, then the disting assumes 2048 frames per waveform. This is a convenience to support the popular 'Serum' format of wavetable, since those always have 2048 frames per waveform. This means you can simply download one of thousands of Serum wavetables from the internet and drop them straight onto the disting NT's MicroSD card.
- If neither of the above apply, the disting assumes 600 frames per waveform.

If you have a wavetable that you'd like to use but you're not sure of the correct number of frames per waveform, open the file in a sample editor, or use the waveform view in your DAW. It should be fairly clear where the waveforms lie just from looking at it. Measure the number of frames in one waveform, or count the number of waves in the file and divide that into the total length of the file.

The current firmware supports up to 1000 wavetables on the MicroSD card. The maximum size of each individual wavetable depends on the specific algorithm using it.

Signal flow

The disting NT implements a very straightforward scheme for passing signals around the system. There are no virtual patch cables or hidden paths between algorithms.

All algorithms process data from, and to, a selection of 'busses'[31], of which there are 28. The algorithms get to work on this data one at a time, in sequence, from top to bottom as seen in the algorithm overview.

At the top of the chain, the first 12 busses are driven from the module's 12 inputs. The remaining 16 busses are empty/silent.

The next 8 busses feed the module's 8 outputs, at the end of the chain. The remaining 8 busses are 'aux' busses, provided for more routing flexibility between algorithms.

In the algorithm overview, each algorithm's block has a graphical representation of which busses it is using for input and output:
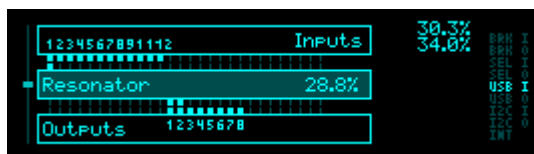
For example, in the image above, the Resonator is taking input from the "Input 1" bus and producing output on the "Output 1" & "Output 2" busses.

Algorithms can be independent

The module is not restricted to running a single signal chain, from inputs to outputs. Implicit in the above bus scheme is the fact that algorithms can work on any bus, no matter where they are in the stack.

To take a simple example, the first algorithm could take inputs from physical inputs 1-2 and output to physical outputs 1-2, while the second algorithm could take inputs from physical inputs 3-4 and output to physical outputs 3-4 - thus operating as two completely independent processes; two separate virtual modules, if you like.
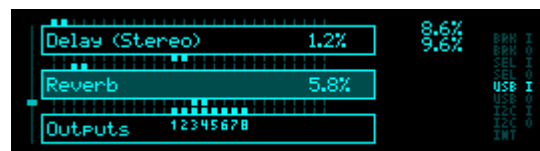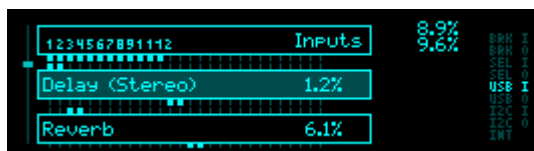
Output mode

Many algorithms have a parameter (or parameters) named 'Output mode', which can either be 'Add' or 'Replace'. This refers to whether the algorithm output in question will be summed onto the output bus, or will replace the bus contents entirely. Both are useful in different situations.

For example, algorithms which are instruments will typically add their outputs to the bus, so if you have multiple instruments feeding the same bus, their outputs will all be audible together.

On the other hand, an EQ will typically want to replace the signal on the bus with the EQ'd version.

---

31 We're using the term 'bus' as commonly used when describing audio mixers: see https://en.wikipedia.org/wiki/Audio_bus

For other algorithms, the two modes are both useful in different usages. For example, a delay might want to add its output to a master mix bus when used as a send effect, but might want to replace the bus signal when used as an insert (then perhaps using its own wet/dry mix to control the signal balance).

In the algorithm overview, outputs which are using 'Add' mode will also show as active inputs to the block:
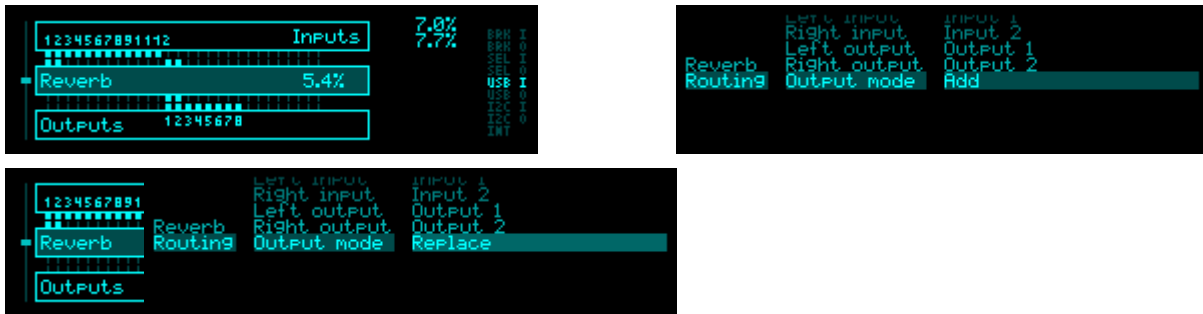
Algorithm specifications

Algorithms have the concept of 'specifications'. These are in general either

- Things that affect the resource (chiefly memory) usage of the algorithm, for example, the maximum delay time of an echo effect.
- Things that affect the number of parameters an algorithm has, for example, the number of audio channels of a mixer.

Some algorithms have no specifications; currently, the highest number of specifications that any algorithm has is two.

Specifications are set when initially adding an algorithm. If necessary, they can be changed later via the menu.

Changing specifications interrupts audio, as it's essentially causing the entire preset to be rebuilt.

CPU usage/overload

The CPU usage of each algorithm is shown in its box in the algorithm overview. The total CPU usage for all algorithms is shown top right. You should aim to keep this below about 90% for reliable operation.

Below this figure is another percentage, which is an estimate of the CPU usage of the entire module, which takes into account MicroSD card access and other background processes. In some cases, you may find this goes significantly higher than the algorithm CPU usage.

If you add too many complex algorithms, or if you adjust algorithm parameters in certain ways, you may end up in a situation where the module's CPU can no longer keep up. This will likely result in crackly audio and the UI may run slowly or even freeze.

If it detects this situation, the module will suspend all algorithm processing and mute the outputs. The

algorithm overview will show the message "CPU OVERLOAD !MUTED!".

The module will stay muted until you do something which is likely to modify the CPU usage, which includes:

- Removing an algorithm.
- Respecifying an algorithm.
- Loading a preset or simply doing 'New preset'.

Preset editor/routing analyser

There is a web-based tool in our GitHub repository ([here](#)[32]) for editing presets. This is very much a work-in-progress, but useful nonetheless. As well as allowing you to view/edit algorithm parameters, it will show you an analysis of the routing within the preset.

---

32 https://github.com/expertsleepersltd/distingNT

Menu reference

The pages that follow describe the disting NT's menus.

Algorithms menu

The Algorithms menu is where you interact with the list of algorithms that makes up a preset - adding/removing algorithms, reordering them etc.

In general the operations here apply to the 'current' algorithm, that is to say, the one highlighted in the algorithm overview.

Add algorithm

Allows you to select an algorithm, which will be added to the end of the list.

You can turn the right encoder to select the algorithm directly, and press the right encoder to add it. You can also press the left encoder to go down to a more detailed view, which shows a description of each algorithm as well as its name.

If the algorithm has specifications, you can adjust these here before adding it.

To the left of the display you'll see three memory usage gauges, showing the impact that the new algorithm will have on the three memory domains. The memory usage of the algorithm may, or may not, be affected by its specifications. If the algorithm exceeds the available memory, the message 'INSUFFICIENT MEMORY' will be shown, and you won't be able to add it.

You can add a total of 32 algorithms to a preset. This limit is completely arbitrary. If you feel it should be higher, please let us know - and we want to see that massive preset you want to make!

Respecify algorithm

Allows you to change an algorithm's specifications, while maintaining its parameters and position in the list.

As in the 'Add algorithm' menu, a memory gauge is shown. You cannot respecify an algorithm if doing so will cause its memory usage to exceed that available.

Respecifying an algorithm may add or remove parameters (for example, respecifying the mixer will add or remove parameters to reflect the change in the number of mixer channels). Any added parameters will assume their default values.

Remove algorithm

Removes the current algorithm from the list.

Move algorithm up

Moves the current algorithm up one position in the list.

Move algorithm down

Moves the current algorithm down one position in the list.

Customise name

Allows you to change the algorithm's name. By default, each algorithm takes the name of the algorithm of which it is an instance e.g. if you add a 'Reverb' the algorithm in the list will be called 'Reverb', but you can change this to anything you like. This can be especially useful if you have more than one instance of the same algorithm, so you can quickly tell them apart.

Duplicate algorithm

Duplicates the current algorithm, if available memory permits. The duplicate algorithm is positioned immediately after the original in the list.

Presets menu

The Presets menu is where you load and save presets, and perform related functions.

A preset contains the full state of the module - the algorithms, their parameters, and any mappings. Presets are stored on the MicroSD card (in this firmware version, at least - future versions may offer the possibility of storing presets in flash memory).

Load preset

Lets you choose a preset file from the MicroSD card to load.

Append preset

Like "Load preset", but the chosen preset file is added to the current state, instead of replacing it - that is, the algorithms in the preset file are appended to the list of running algorithms (if possible, that is, if memory permits).

Name preset

Lets you set the preset name. This will be used as the filename when the preset is saved.

Generate random preset name

Sets the preset name to a randomly generated phrase.

Save preset

Saves the current preset in memory to the MicroSD card, using the name set in 'Name preset' and to the folder chosen in 'Set save location'.

New preset

Removes all algorithms and sets the preset name back to the default 'Init'.

Set save location

Lets you choose the folder on the MicroSD card where presets will be saved. Note that presets can be loaded from anywhere - this only affects the save operation. You're free to take the MicroSD card over to a computer to rearrange, rename, or archive the saved preset files.

Save single algorithm preset

Saves the current algorithm only to a preset file. Useful in combination with 'Append preset' to copy a single algorithm from one preset to another, or simply to save a library of presets for a particular algorithm, independently of whatever else is going on in the preset.

The preset file is named using the algorithm's default name and its custom name as set via the 'Customise name' menu. For example, if you have an instance of the 'Reverb' algorithm and you've customised its name to 'Hall', the preset will be saved as 'Reverb - Hall.json'.

Load disting EX preset

Lets you choose a disting EX preset file and attempt to load it as a disting NT preset. At the time of writing, this will only succeed for disting EX 'Poly Wavetable' presets. If you have disting EX presets for other algorithms that you particularly would like to load into the disting NT, please get in touch!

Author name

Lets you set a name that will be saved in presets as the "author".

Mappings menu

[Video][33]

The Mappings menu is where you set up remote control of algorithm parameters by CVs, MIDI, and I2C. Any parameter can be controlled by any combination of these. Also, any control source can control as many algorithm parameters as you like.

The mappings shown refer to the current algorithm, highlighted in the algorithm overview.

When you enter the menu, the various mapping submenus will reflect the current parameter that was being edited in the single algorithm view. Likewise, if you select a different parameter while setting up the mappings, that parameter will be the current one when you leave the menu.

Mappings are part of the preset, and are saved and loaded with it.

---

33 https://www.youtube.com/watch?v=utpF6fIP_zM

CV Mappings

This is where you set up a mapping from a CV source. This will commonly be an external CV applied to one of the module's inputs, but can also be any signal on any bus within the system - so you can for example run an LFO algorithm and map the CV that it generates to control a parameter on another algorithm. (This is where aux busses can be useful.)

To reiterate - the CV mapping sources are the busses in their state at the algorithm input.

Unless the mapping is defined as a 'Gate', the mapping *adds* to the base parameter value set in the preset - it does not *replace* it. To put it another way, a CV of 0V, or a disconnected cable, will always give you the parameter value as if it wasn't mapped.

The items that can be set up per mapping are as follows:

Parameter	Chooses the parameter to map.
Source	Chooses the CV source bus.
Unipolar?	If set, the CV is clamped to 0V (i.e. negative voltages are treated as 0V).
Gate?	If set, the CV can only toggle the parameter between its minimum and maximum values.
Volts, Delta	These two together determine the relationship between CV voltage and parameter value. The value added to the parameter is equal to (CV voltage) x (Delta) ÷ (Volts) To put it another way, a CV of (Volts) is required to change the parameter value by (Delta).

MIDI Mappings

This is where you set up mappings from MIDI CCs (Continuous Controllers). Channel Pressure (aka Aftertouch) is also supported.

Unlike CV mappings, MIDI mappings directly set the algorithm parameters - they do not add to them.

While in the MIDI Mappings menu, pressing the right encoder enters "Learn" mode. While this is active, the module listens for MIDI CCs and the first one received will be set as the CC for the current mapping. Press the right encoder again to leave Learn mode without altering the mapping. Note that there is a setting for whether setting up a mapping via Learn also disables any other mappings that were already using the CC.

The items that can be set up per mapping are as follows:

Parameter	Chooses the parameter to map.
Channel	Sets the MIDI channel for the mapping controller.
CC	Sets the MIDI CC number (0-127) or selects Channel Pressure.

Enabled?	Sets whether the mapping is enabled or disabled.
Symmetric?	Allows you to specify a mapping as symmetric. For a normal CC, the range 0-127 is mapped across the range set by the min & max settings below. For a symmetric CC, the value 64 is mapped to the midpoint of the range, and values above and below that are scaled by half the total range. A symmetric mapping is appropriate for a parameter which has a bipolar range around zero (for example, a pan position), where you want to be sure that a MIDI value of 64 gives you exactly zero in the middle.
Min	Sets the minimum parameter value that the mapping will set.
Max	Sets the maximum parameter value that the mapping will set.

I2C Mappings

This is where you set up mappings from I2C controllers.

Like MIDI mappings, I2C mappings directly set the algorithm parameters.

The items that can be set up per mapping are as follows:

Parameter	Chooses the parameter to map.
Controller	Sets the I2C controller number.
Enabled?	Sets whether the mapping is enabled or disabled.
Symmetric?	Allows you to specify a mapping as symmetric. For a normal mapping, the range 0-16383 is mapped across the range set by the min & max settings below. For a symmetric mapping, the value 8192 is mapped to the midpoint of the range, and values above and below that are scaled by half the total range. A symmetric mapping is appropriate for a parameter which has a bipolar range around zero (for example, a pan position), where you want to be sure that a controller value of 8192 gives you exactly zero in the middle.
Min	Sets the minimum parameter value that the mapping will set.
Max	Sets the maximum parameter value that the mapping will set.

Settings menu

The Settings menu lets you change various global options. Settings are stored internally, in flash memory, not on the MicroSD card.

UI (User Interface)

This submenu contains options for how you interact with the module.

Setting	Description
Show button help	Sets the number of seconds to show the pop-up help. If set to 1, the help is permanently visible. If set to 0, the help is never shown automatically (but can still be shown via button 3).
Display contrast	Sets the display contrast (brightness).
Display flip	Allows you to flip the display and so use the module upside down.
Screensaver (minutes)	Sets the number of minutes after which the screensaver will kick in.
Button 4 function	Chooses the menu to which button 4 is a shortcut when in the single algorithm view.
Button ordering	Chooses whether buttons 1-4 are numbered from left to right or right to left.
Menu encoder	Chooses which encoder is used to navigate the menus.
Pot threshold	Sets a threshold for movement of the pots.
CPU load display	Sets the update rate of the CPU load figures on the overview page. The options are "Responsive" and "Calm".

Globals

Setting	Description
Tuning A=	Sets a global tuning standard, the default being A=440Hz. The value set here corresponds to MIDI note 69 (A4).

Startup

Options for things that happen when the module powers up.

Setting	Description
Load last preset	Sets whether the module loads the most recently used (that is, saved or loaded) preset at startup. The "Reset preset" menu also has the effect of having the module forget the most recently used preset so it won't be loaded at startup. Please note that the module does not "auto-save". You always have to save manually, if you want to save any changes to the loaded preset.

Select Bus

Setting	Description
Speed multiplier	An experimental feature that allows you to run the Select Bus at higher than usual speeds. Leave this at '1' (normal speed) unless you know you're working with other Select Bus devices that also allow you to change the speed.

MIDI

Setting	Description
CC changes view	If set to 'Parameter', mapped MIDI CCs that change parameter values will also change the currently visible parameter (if the algorithm involved is the currently visible one). If set to 'Parameter and algorithm', the current algorithm will also be changed.
SysEx changes view	If enabled, MIDI SysEx messages that change parameter values will also change the currently visible parameter (if the algorithm involved is the currently visible one).
Soft takeover	If enabled, the disting NT supports the familiar 'soft takeover' paradigm for MIDI CCs, where after loading a preset the controlling MIDI CC has to pass through the value that would set the current parameter value before assuming control. This avoids parameter jumps if the controller's physical position doesn't match that of the preset.
Send CCs	When to send MIDI CCs when parameters change. The options are 'Off', 'On preset load', 'On parameter change', or 'Both'.
Learn is exclusive	If enabled, when a MIDI CC mapping is set via 'Learn', any conflicting mapping of the same CC is unset.
SysEx Device ID	The SysEx device ID to respond to.
Program Change channel	The MIDI channel on which to respond to Program Change messages (see below).
Program Change uses Bank	Sets whether Bank Select (CC #0) is also used when responding to Program Change messages.
Forwarding	Sets whether incoming MIDI on the various ports (USB, Breakout, Select Bus) is passed on to the other ports. The option for "Breakout to Breakout" means that MIDI arriving on the breakout MIDI in will be passed to the breakout MIDI out.

MIDI Program Change

If enabled, the module responds to MIDI Program Change messages by loading a preset. The following rules are followed to determine which preset is loaded:

- If a preset filename contains the program number preceded by a hash ('#'), that preset is

loaded. For example, if a Program Change is sent with a program number of 42, that will load a preset with "#42" anywhere in its name.

- Otherwise, the filenames are simply sorted alphabetically and the program number is used to index into the list.

I2C

Setting	Description
Address (as follower)	The module's I2C address.

Real-time clock

This submenu allows you to set the real-time clock, which is used for timestamping files when writing to the MicroSD card. Set the year, month, day, hour, and minute as desired and then select "> Set <" to actually set the time.

The time will be accurately updated while the module is powered, but will be lost when the module is powered off, since it lacks a battery. However, the MicroSD card is scanned at startup and the clock set to a time after the latest file found, so the times on any newly saved files will at least be later than those that came before.

Calibration

This menu contains various functions related to the calibration of the module. The module requires accurate calibration so that it can receive and generate accurate voltages and translate them to/from software.

The module is calibrated at the factory, so you shouldn't need to worry about doing it yourself. If you do decide to calibrate the module yourself, you will need an accurate source of a +3V reference voltage. Other modules which are designed to deliver accurate pitch CVs are usually a good choice for this, for example, MIDI/CV converters.

Any changes you make to the calibration are not saved to flash unless you specifically do so with the "Save to calibration to flash" menu.

Zero input offsets

This function is not really calibration at all, but is a quick way to remove any DC offsets on the module inputs. Disconnect any patch cables from the module first.

Calibrate an input

This menu allows you to calibrate a single input. Choose an input, select "> Calibrate <", and follow the displayed instructions.

Calibrate an output

This menu allows you to calibrate a single output. Note that it relies on Input 1 already being well calibrated. Choose an output, select "> Calibrate <", and follow the displayed instructions.

Save to calibration to flash

Saves any changes you've made to the calibration to flash memory.

Reset calibration

Resets the calibration to a vaguely sensible, but uncalibrated, state. Note that this function does not save anything to flash.

Read factory calibration

Loads the calibration as performed by the factory. Note that this function does not save anything to flash. Use "Save to calibration to flash" if you want to save the factory calibration as your new calibration.

View calibration

Allows you to see the various values that have been measured during calibration.

View input voltages

Allows you to see the voltages that the module thinks are on its various inputs. Ideally these would all be very close to zero when no patch cables are connected. The readings here will only be accurate if the module is well calibrated.

Audio recordings folder

This setting allows you to choose the folder for new recordings made by the Audio recorder algorithm.

Factory reset

This menu restores all settings to their default values. Note that this does not affect the calibration.

Misc(ellaneous) menu

The 'Misc' menu contains a variety of functions that don't fit anywhere else.

About

Displays the current firmware version information and the module serial number.

MicroSD card

This submenu contains some utilities related to the MicroSD card.

Remount

Lets the module know that you've removed the MicroSD card and reinserted it. Unlike a full-size SD card, a MicroSD card has no physical mechanism for letting the module know that the card was removed - as far as the module is concerned, the card simply stopped responding properly. Selecting this menu allows the module to reinitialise communication with the card.

Test speed

Performs a speed test on the card, and displays some statistics about the results.

Browse

Allows you to browse the contents of the card, display information about files, and in some cases view the contents of the files.

Files with the extensions ".txt", ".json", and ".lua" can be viewed as text. Use the left pot or left encoder to scroll through the file.

Files with the extension ".wav" can be viewed as a waveform, with the file's sample rate, channel count, bit depth, and length (in seconds) displayed.

Enter USB disk mode

Video[34]

This reboots the module into a mode where the MicroSD card appears as a USB disk. All other functionality is suspended while in this mode.

If you connect the disting NT to a computer while in this mode it will appear as a removable drive,

---

34 https://www.youtube.com/watch?v=nHvTzokMLYA

just as if you'd plugged in a USB thumb drive or portable harddrive. You can use this to manage files on the card without having to remove the card from the module and plug it into a MicroSD adaptor on your computer.

Remember to eject the drive on your computer before powering off the module or rebooting into normal mode.

Press both encoders together to reboot the module and resume normal operation.

Reboot

Reboots the module, as if you'd turned the power off and on again.

Enter bootloader mode

Puts the module into a mode in which it can communicate with the firmware update tool. Note that there is no way back from this mode other than to physically power cycle it.

MIDI monitor

Runs a MIDI monitor, showing the MIDI messages that the module is receiving as text. Select the "> GO <" item to start the monitor.

The 'Show Clocks' option selects whether to include MIDI clock messages (F8h) in the display, which tend to flood out any useful information if shown (unless, of course, you're actually checking whether you're receiving clocks or not).

I2C monitor

Runs an I2C monitor, showing the I2C messages that the module is receiving as text.

MIDI

This menu contains a couple of MIDI utilities.

- **All notes off** sends a MIDI "all notes off" message (CC 123, value 0) on all 16 MIDI channels, to all destinations (breakout, Select Bus, USB, and all algorithms).
- **All sound off** is similar but sends a MIDI "all sound off" message (CC 120, value 0).

UI Scripts menu

This menu allows you to run a UI script. See below.

Algorithm-specific menu

The last menu, if present, offers functions specific to the current algorithm (the one highlighted in the algorithm overview). For example, the Granulator menu allows you to load and save samples. See each algorithm for details of its menu, if any.

Common algorithm features

A number of the disting NT algorithms share some features, which are described below.

Microtuning (Scala/MTS)

Microtuning is supported by both Scala files and MTS (MIDI Tuning Standard). Please see the section above for how files defining these tunings are arranged on the MicroSD card.

There are a number of parameters which together define how microtuning will be applied in the algorithm. The most important is 'Microtuning', which can be 'None', 'Scala', or 'MTS'. This should be fairly self explanatory.

Scala

If 'Microtuning' is set to 'Scala', the tuning is defined by a Scala .scl file and optionally a .kbm file as well.

It is possible to send the files over MIDI SysEx, but note that sending Scala over MIDI is not a standard operation – to our knowledge it is currently only supported by our own tool, which you'll find in our [GitHub](https://github.com/expertsleepersltd/distingNT/tree/main/tools)[35]. Choose 'From SysEx' for the 'Scala .scl' or 'Scala .kbm' parameter to use the file sent in this manner.

It is much more common to use files directly, which in our case means from the MicroSD card. Simply choose the files you want to use via the 'Scala .scl' and 'Scala .kbm' parameters.

The 'Scala .kbm' parameter has another option, which is 'Automatic'. In this case, a .kbm file is not required, and the algorithm constructs the equivalent information from two further parameters, 'Auto kbm root' and 'Auto kbm Hz'. Refer to the official .kbm format documentation [here](https://www.huygens-fokker.org/scala/help.htm#mappings)[36]. The automatic keyboard map generated is equivalent to the following .kbm file:

```
! Template for a keyboard mapping
!
! Size of map. The pattern repeats every so many keys:
<number of pitches in the .scl file>
! First MIDI note number to retune:
0
! Last MIDI note number to retune:
127
! Middle note where the first entry of the mapping is mapped to:
'Auto kbm root' value
! Reference note for which frequency is given:
'Auto kbm root' value
! Frequency to tune the above note to (floating point e.g. 440.0):
'Auto kbm Hz' value
! Scale degree to consider as formal octave (determines difference in pitch
! between adjacent mapping patterns):
<number of pitches in the .scl file>
! Mapping.
! The numbers represent scale degrees mapped to keys. The first entry is for
```

---

35 https://github.com/expertsleepersltd/distingNT/tree/main/tools

36 https://www.huygens-fokker.org/scala/help.htm#mappings

```
! the given middle note, the next for subsequent higher keys.
! For an unmapped key, put in an "x". At the end, unmapped keys may be left out.
0
1
2
<etc. up to the number of pitches in the .scl file minus 1>
```

MTS

If 'Microtuning' is set to 'MTS', the tuning can come either from MIDI SysEx messages, or loaded from the MicroSD card. This is selected by the 'MTS .syx' parameters. If this is set to 'From SysEx', the algorithm will use the tuning sent to it over MIDI; otherwise, it will load a SysEx dump from the card and use that.

The supported format is the bulk tuning dump as described [here](#)[37].

Chord/quantizer scales

The available scales are as follows.

Name	Notes	Example (on C)
Chromatic	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	
Major	1, 3, 5, 6, 8, 10, 12	C D E F G A B
Natural Minor	1, 3, 4, 6, 8, 9, 11	C D E♭ F G A♭ B♭
Harmonic Minor	1, 3, 4, 6, 8, 9, 12	C D E♭ F G A♭ B
Dominant	1, 3, 5, 6, 8, 10, 11	C D E F G A B♭
Fully Diminished	1, 3, 4, 6, 7, 9, 10, 12	C D E♭ F F♯ A♭ A B
Dominant Dim	1, 2, 4, 5, 7, 8, 10, 11	C D♭ E♭ E F♯ G A B♭
Augmented	1, 4, 5, 8, 9, 12	C E♭ E G A♭ B
Whole Tone	1, 3, 5, 7, 9, 11	C D E F♯ G♯ A♯

Common polysynth features

A number of the disting NT algorithms are "polysynths", and share some features, which are described below.

CV/gate setup

Polysynths can use up to 6 input busses as gates. Each gate can use up to 11 input busses as CVs. The CV busses follow immediately after the gate bus e.g. if you select input 1 as the gate, and choose to use two CVs for that gate, the CVs will be inputs 2 and 3.

---

37 https://midi.org/midi-tuning-updated-specification

While it might be convenient to have free assignment of each CV bus, this rapidly becomes unwieldy when you have a lot of busses to assign, so the present system is offered as a compromise between usability and flexibility.

Q: I thought CV/gates came in pairs? What does it mean to have one gate and multiple CVs?

A: It means that a gate will trigger a chord - one note per CV.

Gate velocity

Polysynth gates are "velocity sensitive" - the voltage of the gate is used as a velocity value, just like the velocity that you get from a MIDI keyboard. Typically this is used to scale note volume, but various synths might map it to filter cutoff, wavetable position, etc.

A gate voltage of 5V is "maximum velocity".

Microtuning (Scala/MTS)

All polysynths support microtuning, as described above.

MPE

Polysynths support MPE (MIDI Polyphonic Expression). Currently this is limited to correctly handling note allocation when receiving MPE, and to supporting per-note pitch bend. Per-note aftertouch and the "Y dimension" are not currently supported.

MIDI channels for MPE are set using the 'MIDI channel' and 'MPE channels' parameters. The 'MIDI channel' sets the MPE common channel (typically channel 1). The 'MPE channels' parameter sets the range of channels that will be used for notes. If the common channel is 1, then the note channels will be from 2 up to a maximum. A special case is common channel as 16, in which case the note channels will go from 15 down to a minimum (this is an official MPE usage called the "upper zone").

Chord generator/arpeggiator

The polysynths contain chord generators, which can add harmony notes to the notes you play, and an arpeggiator, to play those harmony notes as patterns instead of all at the same time.

The arpeggiator pattern is advanced by the gate, if playing via CV/gate, or by the MIDI or I2C note on message.

Harmony modes

There are two different ways in which the chord notes can be chosen.

The first, 'Shape', builds up a chord using the original note as the root note, according to the shape and inversion parameters.

The second, 'SATB', builds a chord according to common practice[38] four part harmony. (SATB stands for soprano/alto/tenor/bass.) The original note is assigned to one of the four voices according to the 'cantus firmus' parameter, and the remaining voices fill out the chord according to the root degree and position parameters. The root can also optionally be set from a CV input.

Note that the root is not the same as the key. In the key of C major for example, a root degree of II gives the chord D-F-A. In the key of D major, a root degree of I gives D-F♯-A.

Chords, arpeggios, and multiple CV inputs per gate

How these features interact warrants some clarification.

If there are multiple pitch CVs per gate, the chord generation is applied to each pitch CV. For example, if you supply the notes C and D, and set the chord generation to 'triad' in C major, you'll get the notes C, E, G, D, F, A.

If you enable arpeggiation, but not chords, then the arpeggiation is over the input pitch CVs. So for example if you have three CVs and one gate, then the arpeggiator will run over the three notes supplied to the CV inputs. Note that this is useful even if you have a single pitch CV per gate, since you can still use the arpeggio range setting to get octaves.

If chords and arpeggiation are both enabled, then the arpeggio is over all the notes generated by the chords. In most cases, the notes are sorted and then the Up, Down, etc. direction imposed. The exception is 'As Played' mode – in this mode the notes are ordered according to the CV input that generated them. In the above example, 'As Played' mode would give you C, E, G, D, F, A, in that order, whereas 'Up' would give you C, D, E, F, G, A.

Chord/arp parameters

Name	Min	Max	Default	Unit	Description
Enable	0	2	0		Enables the chord generator and chooses which variant to use. The options are 'Off', 'Shape', and 'SATB'.
Key	-12	12	0		Sets the key of the chord (C, D ♭ , D, etc.).
Mode	1	12	1		Sets the mode, that is, the rotation of the notes within the scale. If the selected scale is 'Major', the mode is displayed by one of the familiar names Ionian, Dorian, Phrygian, Lydian, Mixolydian, Aeolian, or Locrian.
Scale	0	8	1		Sets the scale to use to build the chord. See above.
Shape	0	13	0		In 'Shape' mode: sets the chord shape. See below.

---

38 https://en.wikipedia.org/wiki/Common_practice_period

Inversion	0	3	0		In 'Shape' mode: sets the chord inversion. For example the first inversion takes the first note of the chord and moves it an octave up, so the lowest note in the chord is now the second (e.g. C E G becomes E G C). See e.g. here[39] for a fuller explanation of inversions.
Root degree	0	7	1		In 'SATB' mode: the root degree. Specified manually (I-VII) or as 'From CV'.
Root CV	0	28	0		The input bus to use as the root degree pitch if 'Root degree' is 'From CV'.
Cantus firmus	0	3	3		In 'SATB' mode: which voice (soprano/alto/tenor/bass) corresponds to the original note.
Position	0	5	0		In SATB mode: chooses Close or Open position, or one of the T-voice options. See below.
Arpeggio	0	14	0		Enables the arpeggiator and chooses the arpeggiation mode. See below.
Range	1	3	1		When set to 1, the arpeggio is simply the notes formed by the chord. When set to 2 or 3, a copy of the chord is appended to the pattern, one or two octaves higher, creating a longer pattern that spans multiple octaves.
Arp reset	0	28	0		The input bus to use as the arpeggiator reset. A trigger pulse into this input will reset the arpeggiator back to step 1.
Break time	0	1000	0	ms	The 'break' time for chords. See below.
Break dir	0	2	0		The 'break' direction. See below.

Chord shapes

The available shapes are as follows.

Name	Notes (within scale)	Example (in C major)
None	1	C
Octave	1-1(8ve)	C C(8ve)
Two Octaves	1-1(8ve)-1(15ma)	C C(8ve) C(15ma)
Root/Fifth	1-5	C G

---

39 https://en.wikipedia.org/wiki/Inversion_(music)#Inversions

Root/Fifth + 8ve	1-5-1(8ve)	C G C(8ve)
Triad	1-3-5	C E G
Triad + 8ve	1-3-5-1(8ve)	C E G C(8ve)
Sus4	1-4-5	C F G
Sus4 + 8ve	1-4-5-1(8ve)	C F G C(8ve)
Sixth	1-3-5-6	C E G A
Sixth + 8ve	1-3-5-6-1(8ve)	C E G A C(8ve)
Seventh	1-3-5-7	C E G B
Seventh + 8ve	1-3-5-7-1(8ve)	C E G B C(8ve)
Ninth	1-3-5-7-2(8ve)	C E G B D

SATB mode – positions

In SATB mode, the 'Position' parameter controls the spacing between the various voices.

When set to 'Close', the voices adopt the closest triad notes possible to the cantus firmus. For example, if the cantus firmus is the soprano, the key is 'C', and the soprano voice is C4, then the alto, tenor, and bass notes will be G3, E3, and C3 respectively.

When set to 'Open', as compared to 'Close', the alto takes the tenor note, and the tenor takes the alto note dropped by an octave. The bass also drops an octave if required to keep it below the tenor. So in our example above, the alto, tenor, and bass notes will be E3, G2, and C2 respectively.

The remaining options for position are based on the 'Tintinnabuli'[40] style developed by the composer Arvo Pärt. They make most sense when the cantus firmus is the alto part, which we will assume here – if not, the voices are calculated as explained and then simply permuted so what we call the alto here is the desired voice.

The bass voice is simply chosen as one octave below the alto. The tenor and soprano are chosen as follows:

T-vce +1 -1	Soprano is first position superior (the closest triad note above the alto). Tenor is first position inferior (the closest triad note below the alto).
T-vce +2 -1	Soprano is second position superior (the triad note one up from first position). Tenor is first position inferior.
T-vce +1 -2	Soprano is first position superior. Tenor is second position inferior (the triad note one down from first position).
T-vce +2 -2	Soprano is second position superior.

---

40 See e.g. https://en.wikipedia.org/wiki/Tintinnabuli

	Tenor is second position inferior.

Arpeggio modes

The options are as follows:

Name	Behaviour	Example (on C major triad)
Up	Notes are played from lowest to highest.	C E G C E G ...
Down	Notes are played from highest to lowest.	G E C G E C ...
Alt	Notes are played alternately up and down.	C E G E C E G ...
Alt2	Notes are played alternately up and down, repeating the top & bottom notes.	C E G G E C C E G ...
Up -8ve	See below.	
Down -8ve	See below.	
Alt -8ve	See below.	
Alt2 -8ve	See below.	
Step Up	The lowest note alternates with the other notes in the chord, in rising order.	C E C G C E C G ...
Step Down	The highest note alternates with the other notes in the chord, in descending order.	G E G C G E G C ...
Random	Notes are played in a random order.	
Random2	Notes are played in a random order, except that the same note cannot be played twice in a row.	
Random3	The notes in the chord are played in a random permutation, then another random permutation, and so on.	
As Played	Notes are played according to the order of their CV inputs.	

The "-8ve" modes differ from the basic modes in how they treat the Range parameter (above), for shapes which end in "+8ve". As an example, consider the Triad+8ve shape in C major, which contains the notes:

C E G C(8ve)

If Range is set to 2, this pattern is repeated an octave higher, so modes Up/Down/Alt/Alt2 will arpeggiate the notes:

C E G C(8ve) C(8ve) E(8ve) G(8ve) C(15ma)

Note how C(8ve) is repeated. The "-8ve" modes skip this repeated note, so for example the Up-8ve mode will play:

C E G C(8ve) E(8ve) G(8ve) C(15ma) C E G …

Broken chords

The 'Break time' and 'Break direction' parameters allow you to automatically 'break' chords i.e. the notes are played one at a time, instead of simultaneously. With a small time, this gives a 'strumming' effect, though you can specify much longer times if you wish so each note is quite distinctly separated.

Note that this applies to multiple notes triggered when using a single gate input with multiple CV inputs, as well as when chord mode is activated.

The 'Break direction' allows you to specify whether the chord is played from bottom to top ('Up'), from top to bottom ('Down'), or alternately up and down ('Alternating').

The arpeggio reset input, if used, also serves to reset the alternating break direction to up.

Sustain mode

The polysynth 'Sustain mode' parameter sets the behaviour of the sustain function, when triggered either by MIDI or by the 'Sustain' parameter. The options are "Synth" (sustained notes cannot be retriggered) and "Piano" (sustained notes can be retriggered).

Algorithm reference

The pages that follow detail the various algorithms available.

Accent sweep

"AD envelope with variable peak"

File format guid: 'acsw'

Specifications: None

Description

This algorithm recreates a very specific aspect of the Roland TB-303's circuitry. There is a lengthy write-up on it on the Devil Fish page [here](#)[41], but essentially it creates an AD (attack/decay) envelope which will reach a higher maximum level if repeatedly triggered.

GUI

The display shows the input (dark) and output (light) signals.
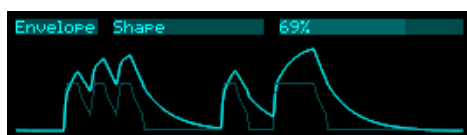
Envelope parameters

Name	Min	Max	Default	Unit	Description
Input mode	0	1	0		Sets how the input is treated: "Direct" feeds the input signal directly into the AD shaper. "Trigger" uses the input to trigger an internal pulse, which is fed into the shaper.
Trigger length	1	200	100	ms	If the input mode is "Trigger", the length of the pulse.
Trigger voltage	0.1	10.0	10.0	V	If the input mode is "Trigger", the voltage of the pulse.
Shape	0	100	100	%	This is the position of the pot (potentiometer), if you studied the Devil Fish write-up. At 0% the output is quite close to the input; at 100% it is heavily affected by the AD shaper.
Gain	0	24	0	dB	Applies a gain to the output shape.
Charge	1	470	47	kΩ	The value of the charge resistor.

---

[41] https://www.firstpr.com.au/rwi/dfish/303-unique.html

Discharge	1	470	100	kΩ	The value of the discharge resistor.
Pot	1	470	100	kΩ	The value of the potentiometer.
Capacitance	0.1	10.0	1.0	µF	The value of the capacitor.

Routing parameters

Name	Min	Max	Default	Unit	Description
Input	1	28	1		The bus to use as input.
Output	1	28	15		The bus to use as output.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.

Attenuverter

"Attenuates and offsets signals"

File format guid: 'attn'

Specifications:

- Channels, 1-12: The number of bus channels to process.

Description

This simple algorithm scales and offsets signals. It is probably most useful for CVs, but can be used for audio as well.

The signal is scaled and then offset i.e.

$$output = offset + (input \times scale)$$

Various offset parameters are provided, which might be useful in different scenarios - they are all simply added together.

Note that it is valid to use this algorithm with no input bus selected, in which case the output is simply the offset voltage.

Parameters

Name	Min	Max	Default	Unit	Description
Bypass	0	1	0		If set, effectively disables all processing.

Per-channel parameters

Name	Min	Max	Default	Unit	Description
Input	0	28	1		The input bus to process.
Enable	0	1	0		Enables the channel. Disabled channels have no effect on the bus.
Output	0	28	0		The output bus. If set to 'None', the input bus is used as output, and the mode is always 'Replace'.
Output mode	0	1	1		The standard Add/Replace mode selector as described above.
Scale	-200.0	200.0	100.0	%	Sets the channel scale.

Offset	-10.0	10.0	0.0	V	Sets the channel offset (Volts).
Fine	-1000	1000	0	mV	Sets the channel offset (millivolts).
Octaves	-10	10	0	V	Sets the channel offset (whole Volts).
Semitones	-60	60	0	ST	Sets the channel offset (semitones i.e. 1/12th of a Volt).

Audio recorder

"Records audio to the MicroSD card"

File format guid: 'wavr'

Specifications:

- Max files, 1-10: The maximum number of simultaneous files to record.

Description

This algorithm records WAV files to the MicroSD card. A typical use for this would be to record the module's inputs, but any bus can be the source for a recording, so you could also record the output of other algorithms - for example, you could run a mixer and record the stereo mix-down.

The algorithm can also play back its recordings. It does so to the same bus channels as are chosen for recording.

Since the module is fully DC-coupled, it can also record and play CVs.

Up to 10 files can be recorded simultaneously, each mono or stereo. It goes without saying that the more files you record, and the larger the chosen bit depth, the more demands are placed on the MicroSD card. You would be advised to check your card performance before relying on the algorithm for any crucial recordings.

GUI

The display shows a level meter for each file. This will be grayed out if the file is disabled. The meters show the input levels when idle or recording, and the playback levels when playing.

While recording, the display shows the name of the current recording, and the elapsed time. It also shows a gauge of the pressure on the MicroSD card.
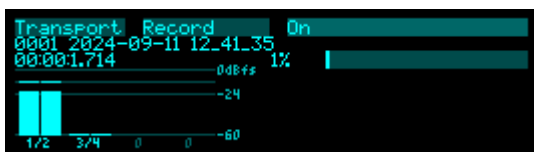
When idle, the display shows the name of the last recording completed or played.
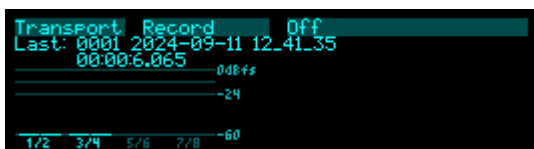
During playback, the display shows the name of the recording being played, and the elapsed time.
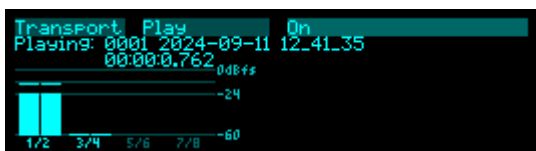
Parameters

Name	Min	Max	Default	Unit	Description
Record	0	1	0		Starts/stops recording.
Play	0	1	0		Starts/stops playback.
Record lock	0	1	0		If on, changes to the Record parameter are ignored.
Play lock	0	1	0		If on, changes to the Play parameter are ignored.
Bit depth	0	2	0		Chooses the bit depth for recording. The options are 16, 24, or 32 bit.
Normalisation	0	1	0		Sets the voltage that corresponds to full-scale in the recorded files. The options are 10V or 12V. Applies to both recording and playback.
Which recording					Selects the recording to play.
Gain	-40	24	0	dB	Sets the playback level.

Per-file parameters

Name	Min	Max	Default	Unit	Description
Enable	0	1	0		Enables the file for recording and playback.
Left/mono input	1	28	1		Sets the left or mono channel to record.
Right input	0	28	2		Sets the right channel to record.
Playback	0	1	1		Enables the file for playback. (Both 'Enable' and 'Playback' must be on for the file to be included in playback.)

Augustus Loop

"Tape-like delay"

File format guid: 'augu'

Specifications:

- Max delay time, 1-44 seconds: The maximum delay time.

Description

Augustus Loop is a disting NT implementation of one of Expert Sleepers' oldest products, the VST plug-in of the same name ([here](#)[42]). Essentially, it's a tape-inspired stereo delay.

The delay time can be dialled in manually, or set by tap tempo or a clock input. The longest delay possible is around 44 seconds.

Being a tape delay, you can change the tape speed via CV. Patching an LFO into here is your route to all manner of subtle detuning or extreme mangling effects. You can also stop and reverse the tape.

There is an option to run the delay signal through an 'effects loop', allowing you to insert other effects or processing into the guts of the delay.

Note that the 'Pitch CV' input changes the tape speed. It is labelled pitch rather than speed to emphasise the fact that it is scaled 1V/octave.

This diagram is reproduced from the VST plug-in user manual, and explains the signal flow graphically:

---

42 https://expert-sleepers.co.uk/augustusloop.html
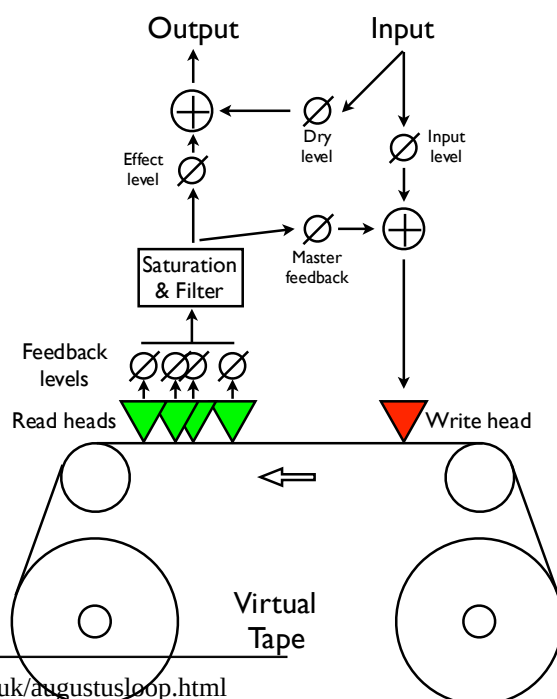
There are four 'tape read heads' with independent delay times and stereo positions, allowing for straight stereo delays, ping pong delays, or hybrid multi-tap style effects.

GUI

The display shows the feedback amount bottom left and the delay time bottom right. The centre area shows an animation indicating the tape direction and speed.

If using a clock input, the word 'Clock' flashes up every time the algorithm receives a clock pulse.

The word 'Tap' is displayed if the tap tempo function will set the delay time on the next tap i.e. it shows whether tap tempo is 'live'.

Delay parameters

Name	Min	Max	Default	Unit	Description
Time (coarse)	0.0	43.7	1.0	s	Sets the delay time. The coarse and fine delay times are added to produce the actual delay time.
Time (fine)	-100	100	0	ms	Sets an adjustment to the delay time, in milliseconds. The coarse and fine delay times are added to produce the actual delay time.
Delay multiplier	0	23	15		A multiplier to apply to the delay time set by the parameters, the tap tempo, or the clock. See below for the available values.
Feedback	0	100	50	%	The overall delay feedback amount.
L-L Time	0	100	100	%	Scales the delay time of the left-to-left feedback path, as a percentage of the overall delay time.
L-R Time	0	100	50	%	Scales the delay time of the left-to-right feedback path, as a percentage of the overall delay time.
R-L Time	0	100	50	%	Scales the delay time of the right-to-left feedback path, as a percentage of the overall delay time.
R-R Time	0	100	100	%	Scales the delay time of the right-to-right feedback path, as a percentage of the overall delay time.
L-L Level	0	100	100	%	Scales the amount of the delayed left signal mixed into the left feedback path.
L-R Level	0	100	0	%	Scales the amount of the delayed left signal mixed into the right feedback path.

R-L Level	0	100	0	%	Scales the amount of the delayed right signal mixed into the left feedback path.
R-R Level	0	100	100	%	Scales the amount of the delayed right signal mixed into the right feedback path.
Mono-ize	0	100	0	%	Reduces the stereo width of the incoming signal. At zero the signal is reduced to mono, at a pan position set by the 'Initial pan' parameter.
Initial pan	-100	100	-100	%	Sets the pan position of the mono-ized signal. -100 is fully left; 100 is fully right.

Mix parameters

Name	Min	Max	Default	Unit	Description
Dry gain	-40	6	-40	dB	The amount of the dry signal to mix into the outputs. At "-40" there is no dry signal at all i.e. it's actually –∞dB.
Effect gain	-40	6	-3	dB	The amount of the effect (delay) signal to mix into the outputs. At "-40" there is no effect signal at all i.e. it's actually –∞dB.
Input level	0	100	100	%	Attenuates the input signal fed to the tape write head.

Tape parameters

Name	Min	Max	Default	Unit	Description
Pitch inertia	0	125	64		Sets the amount of 'inertia' or slew on the pitch CV input. At zero, the tape speed follows the pitch input closely; at the maximum value, pitch changes are quite gradual.
Stop tape	0	1	0		When on, the tape speed is set to zero. Note that the Pitch inertia affects how quickly the tape stops and starts.
Reverse tape	0	1	0		When on, the tape is reversed. Note that the Pitch inertia affects how quickly the tape reverses.
Bit depth	0	1	1		Controls the bit depth used in the delay memory (note, not the bit depth used in any other processing). Setting this to '0' (16 bit) doubles the maximum delay time.
Inertia free	0	1	0		Enables 'Inertia free' mode. See below.

Inertia fade time	1	1000	100	ms	The fade time to use when in Inertia free mode.
Pitch CV input	0	28	0		The CV input to use for pitch.
Pitch LFO speed	0	127	96		Sets the speed of the pitch modulation LFO.
Pitch LFO depth	0	100	0	%	Sets the depth of pitch modulation by the LFO.
Clear loop	0	1	0		When on, instigates a rapid clear of the delay buffer (while maintaining passthrough of the dry signal).

Filter/Sat parameters

Name	Min	Max	Default	Unit	Description
Filter type	0	400	0		Sets the filter type. See below.
Filter freq	0	127	64		Sets the filter frequency.
Filter Q	0	100	20		Sets the filter resonance.
Saturation enable	0	1	1		Enables the saturation processing (on the tape output, before the filter).
Saturation	0	110	0		Sets the depth of the saturation effect, by applying gain before the saturation waveshaper.
Saturation shape	0	100	100		Controls the shape of the saturation. At '100' the effect is that of soft saturation and clipping. At '0' the effect is of hard digital clipping. Note that at shape settings other than '0', some alteration is applied to the signal even if the Saturation level is zero.

Tempo parameters

Name	Min	Max	Default	Unit	Description
Clock input	0	28	0		The CV input to use as the clock. The delay time is set as the time between two rising clock edges.
Tap tempo	0	1	0		When this parameter transitions from 0 to 1, the algorithm acts on a tap tempo event. See tap tempo, below.
Clocks	1	10	1		Sets the number of consistent clocks required to

required					change the delay time. See below.

Routing parameters

Name	Min	Max	Default	Unit	Description
Left input	1	28	1		Sets the bus for the left input.
Right input	1	28	1		Sets the bus for the right input.
Left output	1	28	13		Sets the bus for the left output.
Right output	1	28	14		Sets the bus for the right output.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.
FX Loop position	0	2	0		Enables the effects loop and sets in position the signal flow. See below.
FX Loop output L	0	28	0		Sets the left output to the effect loop.
FX Loop output R	0	28	0		Sets the right output to the effect loop.
FX Loop input L	0	28	0		Sets the left input from the effects loop.
FX Loop input R	0	28	0		Sets the right input from the effects loop.

Delay time multipliers

The available delay time multipliers are as follows:

1/64	1/48	1/32	1/24	1/16	1/12	1/8	1/6
3/16	1/4	5/16	1/3	3/8	1/2	3/4	x1
x1.5	x2	x3	x4	x5	x6	x8	x16

Tap tempo

The 'Tap tempo' parameter allows for a tap tempo function. You might like to control this from a button push in a UI script, or from an external MIDI controller.

Two taps are required to set the delay time. Taps more than 11 seconds apart are ignored.

Clocks required

When using the clock input, the algorithm's default behaviour is to follow every clock pulse and immediately change the delay time. This is appropriate if you're using a clock with variable timing (perhaps the gate output from a sequencer rather than a clock per se).

However, sometimes you're actually wanting a steady clock, but occasionally the time between clocks changes anyway – for example, if the clock is coming from your DAW or sequencer, the clock will stop when the transport stops, and then the first clock when the transport starts will be interpreted as a really long clock (the time between stopping and starting the transport).

The 'Clocks required' parameter is a solution to this problem. By raising the value above '1', you're telling the algorithm to only change the delay time when it receives a number of clocks of the same time in succession ('same' here means within 10%) - so it will ignore the rogue clock you get from stopping and starting the transport.

Inertia free mode

"Inertia free" mode relates to the algorithm's behaviour when the delay time is changed, either by changing the master delay time, the multiplier, or the four L-L, L-R, etc. times.[43]

When inertia free mode is not activated, the effect is as if the physical tape heads on a tape machine were slid along the tape to adjust the write/read head gap. This results in a characteristic and fairly drastic pitch slew sound.

When inertia free mode is activated, the algorithm crossfades between the old and new delay times, which is a much more subtle effect. The length of the crossfade can be set with the 'Inertia fade time' parameter.

Filter

A second order state variable filter is available within the delay feedback path. The 'Filter type' parameter lets you smoothly fade between the various responses – 'thru' (i.e. no filtering), low pass, band pass, high pass, and back to thru.

Effects Loop

Enabling the 'FX Loop position' (i.e. setting it to something other than Off) breaks the internal delay feedback path and sends it out of and back into the module, allowing you to insert other effects or processing into the delay. Note that this is different to simply inserting another effect after the delay output – using the effects loop, each delay repeat is progressively more processed by the external effects.

———————————————————————

43 Granted, the name of this parameter isn't particularly well chosen, but this is what it's called in the VST version, and we're sticking with it for the sake of consistency.

Simple ideas include putting external VCAs in the loop to control the amount of feedback. Or you could put other delays, reverbs, or pitch effects (e.g. chorus) into the loop. Putting a pitch shifter into the loop gives you the classic "pitch spiralling off to infinity" sound.

The two options for the effects loop position are 'Pre-Filter' and 'Post-Filter', which as you might expect places the external loop either before or after the filter, giving you the option of filtering before you send the audio to the external effects, or filtering the result that comes back in.

Auto-calibrator

"Calibrates pitch CV outputs"

File format guid: 'cali'

Specifications: None

Description

This algorithm offers oscillator calibration, similar to that in our Silent Way Voice Controller VST plug-in, in the FH-2 module, and in various algorithms on the disting EX.

Pass the pitch CV from another algorithm (for example, the Quantizer) or another module through this algorithm, and connect the algorithm output to your VCO's CV input.

To perform the calibration, you will also need to connect an output from the VCO to an input on this algorithm. This is only required during calibration, and can be disconnected afterwards.

To start the calibration, set the 'Start' parameter to 'On'. The module will output a series of voltages, ranging from -4 Volts up to +6 Volts, and analyse the pitch of the resulting signal from the VCO. As it does so, it will draw a graph of the results, which will ideally look like a nice straight line.
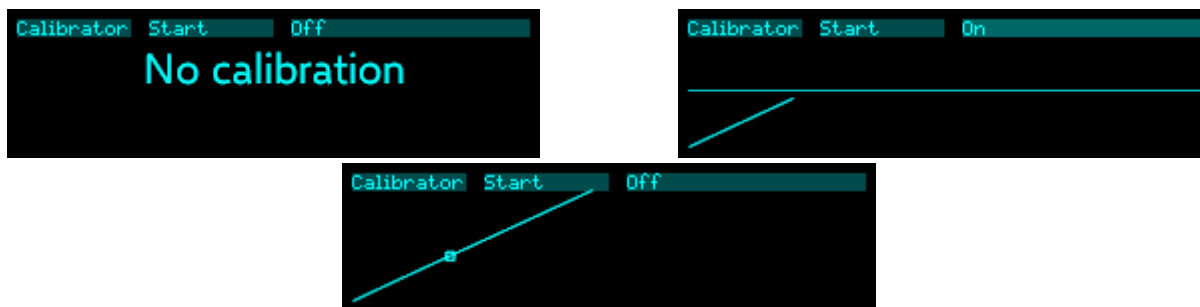
Once the calibration is complete, the algorithm will continuously modify the pitch CV it receives to achieve the expected frequency from the VCO.

When discussing this process we may say "calibrate an output" or "calibrate a VCO" but what is really being calibrated is the *combination* of the disting NT output *and* the VCO. Both may in fact be perfectly well calibrated, in terms of tracking, but the absolute pitch of a VCO is usually determined by a physical tuning knob (not to mention temperature and other factors) and so is hard to know precisely. This process saves you having to tune by hand.

Parameters

Name	Min	Max	Default	Unit	Description
Enable	0	1	1		Enables the algorithm. If disabled, the algorithm passes through CVs unmodified.
Start	0	1	0		Used to start the calibration process.

CV input	1	28	16		The pitch CV input bus.
CV output	1	28	16		The pitch CV output bus. Always uses "Replace" output mode.
Audio input	1	28	1		The audio input to use during calibration.

Auto-sampler

"Creates multisamples"

File format guid: 'auto'

Specifications: None

Description

This algorithm allows you to automatically create multi-sampled instruments in a format that can be used by the Poly Multisample algorithm, by triggering an external synth (by MIDI or CV/gate) and recording the resulting audio.

You can set the range of notes to be sampled, and the step size (e.g. every note, every fourth note etc.). You can also choose to sample multiple velocity levels per note, and multiple round-robins of each note.

The samples are stored on the MicroSD card, in a folder within the root 'samples' folder.

GUI

While sampling, the display shows a level meter, the filename of the sample currently being recorded, and the elapsed time. It also shows a gauge of the pressure on the MicroSD card.

Recording parameters

Name	Min	Max	Default	Unit	Description
Folder name					The name of the folder to create when starting a recording. If the folder already exists, a number will be appended to make the new name unique.
Bit depth	0	2	0		Chooses the bit depth for recording. The options are 16, 24, or 32 bit.
Normalisation	0	1	0		Sets the input voltage that corresponds to full-scale in the recorded files. The options are 10V or 12V.
Left/mono input	1	28	1		The left channel input to record.
Right input	0	28	2		The right channel input, if recording in stereo.

Outputs parameters

Name	Min	Max	Default	Unit	Description
CV output	0	28	14		The pitch CV output bus.
Gate output	0	28	15		The gate output bus.
MIDI output	0	4	0		The MIDI output port (None, Breakout, Select Bus, USB, Internal).
MIDI channel	1	16	1		The output MIDI channel.

Setup parameters

Name	Min	Max	Default	Unit	Description
Start Note	0	127	21		The first note to sample.
End Note	0	127	108		The last note to sample.
Note Step	1	127	1		The number of notes to increment by after each sample (e.g. '1' samples every note, '12' samples every octave etc.).
Vel switches	1	9	1		The number of velocity switch layers to sample.
Min velocity	1	127	25		The minimum velocity value to use when sampling with velocity switch layers.
Max velocity	1	127	127		The maximum velocity value to use when sampling with velocity switch layers, or the fixed value to use if not using velocity switches.
Round robins	1	9	1		The number of round-robins to sample.
Length	0.1	60.0	1.0	s	The note length (i.e. how long the gate is held high, or between MIDI note on and note off).
Gap	0.1	60.0	0.1	s	The time to keep recording after the note is released.
Latency	0	2048	0		Adjusts for latency. See below.
Preview note	0	127	48		Sets the MIDI note number to use when previewing.

Activate parameters

Name	Min	Max	Default	Unit	Description
Activate	0	1	0		Activates auto-sampling when on.
Test	0	1	0		When on, triggers a note so you can test the timing parameters (Length and Gap) and the latency.

Latency

When you activate the Test note, the module listens for incoming audio and measures the delay (latency) between generating the note (sending MIDI or outputting a gate) and receiving audio. This latency figure is shown in the display.
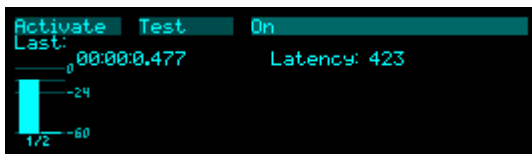
Set the Latency parameter to something close to the shown value to avoid a silence at the start of every recorded sample. Some experimentation may be required to find the optimum value.

Chaos

"CVs from the Lorenz equations"

File format guid: 'xaoc'

Specifications: None

Description

This algorithm generates chaotic CVs according to the Lorenz equations[44] or Rössler equations[45]. Each generates three CVs, named X, Y, and Z, which are available on separate outputs.

You can modify the parameters of the Lorenz equation – 'Rho mod' affects 'r' (aka 'ρ'), and 'Beta mod' affects 'b' (aka 'β'). With the values at 0V, the parameters are the classic values as studied by Lorenz (28 and 8/3 respectively).

GUI

The display shows a graphical representation of the outputs, projected onto the X/Y, X/Z, and Y/Z planes.

Chaos parameters

Name	Min	Max	Default	Unit	Description
Attractor	0	1	0		Chooses which equations to use: Lorenz or Rössler.
Speed range	-64	32	0		Sets the range of the Speed control, in quarter octaves (i.e. a change of 4 doubles or halves the speed).
Speed	-500	500	0		Controls the speed of the simulation.
Rho mod	-10.00	10.00	0.00	V	Modulates the Lorenz 'r' parameter.
Beta mod	-10.00	10.00	0.00	V	Modulates the Lorenz 'b' parameter.

---

44 https://en.wikipedia.org/wiki/Lorenz_system

45 https://en.wikipedia.org/wiki/Rössler_attractor

Reset	0	1	0		When this parameter is set to 1, the system is reset to initial conditions.

Scale/offset parameters

Name	Min	Max	Default	Unit	Description
X scale	-20.0	20.0	10.0	V	The scale of the X output.
X offset	-20.0	20.0	0.0	V	An offset added to the X output.
Y scale	-20.0	20.0	10.0	V	The scale of the Y output.
Y offset	-20.0	20.0	0.0	V	An offset added to the Y output.
Z scale	-20.0	20.0	10.0	V	The scale of the Z output.
Z offset	-20.0	20.0	-8.0	V	An offset added to the Z output.

Routing parameters

Name	Min	Max	Default	Unit	Description
X output	0	28	15		The bus to use for the X output.
X output mode	0	1	0		The standard Add/Replace mode selector as described above.
Y output	0	28	16		The bus to use for the Y output.
Y output mode	0	1	0		The standard Add/Replace mode selector as described above.
Z output	0	28	17		The bus to use for the Z output.
Z output mode	0	1	0		The standard Add/Replace mode selector as described above.

Chorus (Vintage)

"Chorus effect based on an 80s polysynth"

File format guid: 'junc'

Specifications: None

Description

This algorithm is a stereo chorus effect, modelled on that of the classic Juno-6 polysynth, based on measurements of the author's own unit.

Those with keen ears may recognise it as the chorus that is built into the Poly Wavetable algorithm on the disting EX.

Routing parameters

Name	Min	Max	Default	Unit	Description
Left/mono input	1	28	1		The left or mono input bus.
Right input	0	28	0		The right input bus, if stereo.
Left output	1	28	13		The left output bus.
Right output	1	28	14		The right output bus.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.

Chorus parameters

Name	Min	Max	Default	Unit	Description
Mode	0	2	2		The chorus effect to apply. The options are "None", "Mode 1", and "Mode 2".

Clock

"Generates clocks"

File format guid: 'clck'

Specifications:

- Outputs, 1-8: The number of clock outputs to generate.

Description

This algorithm produces and/or receives analogue clock pulses and/or MIDI clock. It is designed to synchronise the module with others, or with other devices that it communicates with via MIDI, or simply to be a source of clock pulses for other algorithms within the module.

It can generate its own BPM-based clock, or sync to external clock pulses, or sync to MIDI.

It can output clock pulses, or send MIDI clock, or both.

If syncing to external clock, the algorithm expects a 24ppqn DINsync style clock. If you only have a slower clock available, use the Clock multiplier algorithm to get the clock up to the right speed before this algorithm sees it.

GUI

The display shows the current tempo, and the transport location in bars, beats, and sixteenths. It also shows the clock source, and whether the transport is running or stopped.
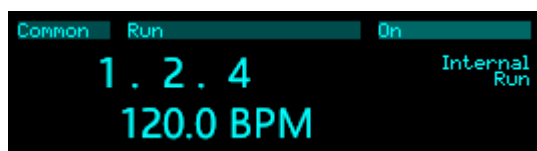
Parameters

Name	Min	Max	Default	Unit	Description
Source	0	2	0		The clock source. The options are "Internal", "External", and "MIDI".
Tempo	30.0	240.0	120.0	BPM	The internal clock tempo.
Run	0	1	0		Starts and stops the internal clock.
Time sig numerator	1	99	4		The time signature numerator.

Time sig denominator	0	4	2		The time signature denominator: one of 1, 2, 4, 8, or 16.
Clock input	1	28	1		The external clock input.
Run/stop input	0	28	2		The run/stop input for the external clock.
Output to breakout	0	1	0		If on, send MIDI clock to the breakout.
Output to Select Bus	0	1	0		If on, send MIDI clock to the Select Bus.
Output to USB	0	1	0		If on, send MIDI clock to USB.

Per-output parameters

Name	Min	Max	Default	Unit	Description
Enable	0	1	0		Enables the output.
Output	0	28	0		The output bus.
Output mode	0	1	1		The standard Add/Replace mode selector as described above.
Type	0	2	0		The output type, one of "Clock", "Run/stop", "Reset", or "Trigger".
Divisor	0	19	9		The clock division to generate. See below.
Low voltage	-10.0	10.0	0.0	V	The output voltage when the clock is low/inactive.
High voltage	-10.0	10.0	5.0	V	The output voltage when the clock is high/active.
Ratchet mode	0	2	1		The ratchet mode, one of "Off", "Twos", and "Twos and threes".
Ratchet	0	7	0		The ratchet division for the clock. If mode is "Twos", selects from 1, 2, 4, 8, 16. If mode is "Twos and threes", selects from 1, 2, 3, 4, 6, 8, 12, 16.
Trigger length	1	100	10	ms	The length of the trigger pulse, if the type is "Reset" or "Trigger".

Clock divisors

The following options are available for the 'Divisor' parameter:

1/64T, 1/32T, 1/32, 1/16T, 1/16, 1/8T, 1/8, 1/4T, 3/16, 1/4, 1/2T, 3/8, 1/2, 1/1T, 3/4, 1/1, 3/2, 2/1, 3/1, 4/1

Clock divider

"Divides clocks"

File format guid: 'clkd'

Specifications:

- Channels, 1-8: The number of clock channels to process.

Description

This algorithm is a simple clock divider, outputting slower clocks from a faster one. The channels can use completely independent clocks, or share clocks.

There is a shared reset input and a per-channel reset input. If either is active, the channel is reset.

Common parameters

Name	Min	Max	Default	Unit	Description
Reset input	0	28	0		The shared reset input bus.

Per-channel parameters

Name	Min	Max	Default	Unit	Description
Type	0	2	0		The divisor type. The options are "Free", "Metrical (2)", and "Metrical (2,3)".
Divisor	1	32	2		The divisor, if the type is "Free". The divisor value is simply the parameter value.
Divisor	0	5	1		The divisor, if the type is "Metrical (2)". Chooses between 1, 2, 4, 8, 16, & 32.
Divisor	0	9	1		The divisor, if the type is "Metrical (2,3)". Chooses between 1, 2, 3, 4, 6, 8, 12, 16, 24, & 32.
Enable	0	1	0		Enables the channel. A disabled channel outputs nothing.
Input	1	28	1		The clock input bus to divide.
Reset input	0	28	0		The reset input bus for this channel.
Output	1	28	15		The clock output bus.
Output	0	1	1		The standard Add/Replace mode selector as

mode					described above.

Clock multiplier

"Multiplies clocks"

File format guid: 'clkm'

Specifications: None

Description

This algorithm is a simple clock multiplier, generating a faster clock from a slower one.

The output clock rate is updated on every clock received, except that the clock duration is limited to at most double each time. This is mainly to prevent the output clock rate changing when the input clock is paused and restarted, which would otherwise be interpreted as a really long clock pulse.

Parameters

Name	Min	Max	Default	Unit	Description
Clock input	1	28	1		The clock input bus.
Clock output	1	28	1		The clock output bus.
Output mode	0	1	1		The standard Add/Replace mode selector as described above.
Multiplier	1	24	2		The clock multiplier.
Low voltage	-10.0	10.0	0.0	V	The output voltage when the clock is low/inactive.
High voltage	-10.0	10.0	5.0	V	The output voltage when the clock is high/active.

Convolver

"Computes audio convolutions"

File format guid: 'conv'

Specifications:

- Max impulse, 1-10 seconds: The maximum length of the impulse sample.

Description

This algorithm performs real-time [convolution](46) of the input signal and another signal (often called an 'impulse response'), which is loaded from the MicroSD card.

This is often used for realistic reverb effects, since the required impulse responses can be sampled from a physical location, but the technique is much more general and can be a very creative way to mangle and combine sounds.

Convolution is notoriously CPU-intensive, and the disting NT is a resource-constrained embedded system – you should not expect to achieve the same results as you might in the latest new fangled convolution reverb plug-in in your DAW, with orders of magnitude more CPU power and RAM. However, this algorithm remains a useful tool within the context of a modular synth.

The algorithm is based on the disting EX algorithm of the same name. There is an in-depth video for that algorithm [here](47).

Impulse length and the factors that affect it

When choosing the parameters for this algorithm there are various trade-offs to be made, all of which affect the maximum length of impulse response that you can use. (A longer impulse response is usually desirable, especially if you're using this for reverb.) These factors are:

Allocated memory: perhaps most obviously, the more memory you allocate to the algorithm via its specifications, the longer the impulse it can handle.

Mono vs stereo: if the output is stereo, the algorithm has to do almost twice the work, so the impulse time is more or less halved. Using the routing parameters, you can select mono, mono-to-stereo, or stereo operation. The last two count as stereo in terms of processing time. Note that the stereo option passes the dry signal in stereo but sums the two input channels to mono before convolution; it does not do true stereo convolution, which would double the CPU cost again.

Latency: the algorithm works on blocks of audio at a time. It is more efficient to work on larger blocks, but conversely using larger blocks means a longer wait until you get the results i.e. higher latency. The Latency parameter offers you the choice between lower latency with smaller impulse

---

46 https://en.wikipedia.org/wiki/Convolution

47 https://www.youtube.com/watch?v=tDFZA9b2A0Y

time, or higher latency with longer impulse time. Note that in either case, the dry signal is still synced up with the convolution result. So for example you could use this as a master bus insert effect to put a nice reverb on your whole mix and probably get away with a long latency.

Sample rate: the disting NT's algorithms all run at 48kHz. However, by downsampling to a lower sample rate internally, a given time's worth of audio is cheaper to process and a longer impulse time can be achieved, at the expense of the bandwidth of the processed audio. The 'Sample rate' parameter offers several choices, from a high-quality 48kHz to a decidedly murky 6kHz. Note that the dry signal is always 48kHz.

Impulse responses on the MicroSD card

Any sample file on the card can be loaded as an impulse response. For convenience, when the algorithm is loaded it defaults to a folder called 'impulses', if such a folder exists. Any kind of WAV file is supported, but stereo 32 bit float is preferred. For fastest loading, choose a sample rate for the file equal to the sample rate at which you want to run the algorithm, but other sample rates will be converted.

There are plenty of free examples online, for example these[48] by EchoThief, or these[49] rather lovely Bricasti M7 impulses. You may find that you have a plug-in in your DAW which contains a library of impulse responses (e.g. Live's Hybrid Reverb) that you could copy to the MicroSD card.

GUI

The display shows a graphical representation of the impulse sample on the left of the screen. On the right side, it shows the mono/stereo processing mode, the maximum impulse time (as a result of the current parameter values), the actual length of the sample file, and the current overall CPU usage.
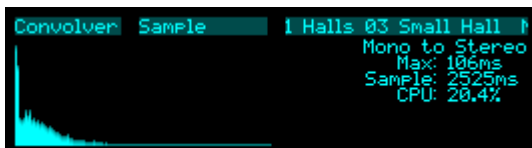
Convolver parameters

Name	Min	Max	Default	Unit	Description
Folder	1				Chooses the folder from which to load the impulse response sample.
Sample	1				Chooses the impulse response sample within the folder.
Latency	0	3	2		Sets the processing latency (5ms, 11ms, 22ms, or 43ms).

---

48 https://www.echothief.com/downloads/

49 https://web.archive.org/web/20190201211631/http://www.samplicity.com/bricasti-m7-impulse-responses/

Sample rate	0	3	0		Sets the processing sample rate (48kHz, 24kHz, 12kHz, or 6kHz).
Partitions	1				Allows you to fine tune the maximum impulse length, and consequently the CPU usage.

Mix parameters

Name	Min	Max	Default	Unit	Description
Dry gain	-40	12	0	dB	The output level of the dry signal.
Convolution gain	-40	12	0	dB	The output level of the convolution result.

Routing parameters

Name	Min	Max	Default	Unit	Description
Left/mono input	1	28	1		The left or mono audio input bus.
Right input	0	28	0		The right audio input bus.
Left output	1	28	13		The left audio output bus.
Right output	0	28	14		The right audio output bus.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.

Crossfader

"Crossfades signals"

File format guid: 'xfad'

Specifications: None

Description

This algorithm crossfades between two sets of signals (from mono up to two sets of eight channels).

Three different crossfade curves are available:

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

Inputs A & B, and the output, use a contiguous range of busses, starting with the one set via the parameters, and with the channel count set by the 'Width' parameter. For example, to crossfade stereo signals, set the Width to 2.

The crossfade can be controlled by both a parameter and a CV, which are simply summed if both are in use.

GUI

The display shows a graphical representation of the crossfade curves, and the amount of the two signals that are combined in the output mix.

Crossfader parameters

Name	Min	Max	Default	Unit	Description
Crossfader	0.0	100.0	50.0	%	The crossfade position, from 0% (input A only) to 100% (input B only).
Curve	0	2	1		The crossfade curve, as above.

Routing parameters

Name	Min	Max	Default	Unit	Description
Input A	1	28	1		The first bus for input A.
Input B	1	28	2		The first bus for input B.
Output	1	28	13		The first bus for the output.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.
Width	1	8	1		The number of busses to process.
Crossfade input	0	28	0		The CV input to drive the crossfade, scaled so that 5V covers the range 0-100%.

Debouncer

"A switch debouncer"

File format guid: 'debo'

Specifications:

- Channels, 1-8: The number of bus channels to process.

Description

This algorithm implements a simple switch debouncer (see e.g. [this Wikipedia page](#)[50]).

It was added primarily so that simple passive guitar footswitches can be attached to the module's inputs (with a suitable voltage source) and used with, for example, the Looper algorithm.

Globals parameters

Name	Min	Max	Default	Unit	Description
Time	1	100	10	ms	The hold-off time. Once the output has changed, it is not allowed to change again for the this duration.

Per-channel parameters

Name	Min	Max	Default	Unit	Description
Input	0	28	13		The input (and output) bus.
Enable	0	1	0		Enables debouncing on this channel.

---

50 https://en.wikipedia.org/wiki/Switch#Contact_bounce

Delay (Mono)

"A simple mono delay effect"

File format guid: 'delm'

Specifications:

- Max delay time, 1-30 seconds: The maximum delay time.

Description

This algorithm is a simple delay effect.

The delay time can be set manually or via a clock pulse.

It can also be modulated via its 'V/oct input', which halves the delay time for every 1V rise in CV (conversely, doubles it for every 1V fall in CV).

The 'Fractional' parameter allows for delay times that are not integer multiples of the sample rate. This slightly increases CPU load but will sound significantly better if the delay time is being modulated, for example for chorus or flange type sounds. Conversely, turning it off makes for an extremely clean digital delay, with no colouration in the repeats.

Delay parameters

Name	Min	Max	Default	Unit	Description
Mix	0	100	100	%	The wet/dry mix.
Level	-40	0	0	dB	The gain applied to the delay/wet signal.
Time	1	32767	250	ms	The delay time.
Feedback	0	100	50	%	The delay feedback.
Delay multiplier	0	23	15		Sets a multiplier to apply to the delay time. Affects both the 'Time' parameter and the delay time set by the clock.
Time change	0	1	0		Sets how changes of delay time will be handled. The options are 'Slew' (the delay time is smoothly interpolated, sounding a bit like a tape being played slower or faster) and 'Crossfade' (the effect rapidly crossfades from one time setting to the other).
Fractional	0	1	0		Enables fractional delay times. See above.

Routing parameters

Name	Min	Max	Default	Unit	Description
Input	1	28	1		The audio input bus.
Output	1	28	13		The audio output bus.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.
Clock input	0	28	0		The input bus to use for the clock.
V/oct input	0	28	0		The input bus to use for 1V/octave modulation of the delay time.

Delay (Stereo)

"A simple stereo delay effect"

File format guid: 'dels'

Specifications:

- Max delay time, 1-30 seconds: The maximum delay time.

Description

This algorithm is a simple stereo in/stereo out delay effect.

The delay time can be set manually or via a clock pulse.

It can also be modulated via its 'V/oct input', which halves the delay time for every 1V rise in CV (conversely, doubles it for every 1V fall in CV).

The 'Fractional' parameter allows for delay times that are not integer multiples of the sample rate. This slightly increases CPU load but will sound significantly better if the delay time is being modulated, for example for chorus or flange type sounds. Conversely, turning it off makes for an extremely clean digital delay, with no colouration in the repeats.

Delay parameters

Name	Min	Max	Default	Unit	Description
Mode	0	2	0		The delay mode. The options are "Stereo", "Stereo ping-pong" (each echo swaps left and right), and "Ping-pong" (the input to the delay is summed to mono, panned, and then swaps left and right on each echo).
Mix	0	100	100	%	The wet/dry mix.
Level	-40	0	0	dB	The gain applied to the delay/wet signal.
Time	1	32767	250	ms	The delay time.
Feedback	0	100	50	%	The delay feedback.
Initial pan	-100	100	-100	%	If the mode is "Ping-pong", sets the pan position of the first delay.
Delay multiplier	0	23	15		Sets a multiplier to apply to the delay time. Affects both the 'Time' parameter and the delay time set by the clock.

Time change	0	1	0		Sets how changes of delay time will be handled. The options are 'Slew' (the delay time is smoothly interpolated, sounding a bit like a tape being played slower or faster) and 'Crossfade' (the effect rapidly crossfades from one time setting to the other).
Fractional	0	1	0		Enables fractional delay times. See above.

Routing parameters

Name	Min	Max	Default	Unit	Description
Left input	1	28	1		The left input bus.
Right input	1	28	2		The right input bus.
Left output	1	28	13		The left output bus.
Right output	1	28	14		The right output bus.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.
Clock input	0	28	0		The input bus to use for the clock.
V/oct input	0	28	0		The input bus to use for 1V/octave modulation of the delay time.

Delay (Tape)

"A tape delay effect"

File format guid: 'delt'

Specifications:

- Max tape length, 1-30 seconds: The maximum length of the 'tape'.
- Stereo: Whether the algorithm is mono or stereo.

Description

This algorithm is based on the disting mk4 algorithm "D-2 Tape Delay", which is itself a simplified version of the "Augustus Loop" effect. It is a delay/echo effect which simulates a variable speed tape loop echo device.

The 'Tape length' parameter sets the range of delay times available. Note though that changing this may introduce clicks and pops. The primary means of changing the delay time is the "Tape speed" parameter and/or the Speed CV input.

The tape speed can be set in the range from half speed (0.5x) to double speed (2x). If using the CV input, the scaling is 8V/octave i.e. +8V gives double speed and -4V gives half speed.

Three options are available for how the wet/dry mix is controlled:

- **With feedback** - the amount of delay in the mix rises in direct proportion to the amount of feedback.
- **Crossfade** - the 'Mix' parameter is a crossfader.
- **Add delay** - the dry level is constant, and the 'Mix' parameter adds in the delay signal.

Delay parameters

Name	Min	Max	Default	Unit	Description
Mix mode	0	2	0		Sets how the delay and dry signals are mixed. See above.
Mix	0	100	100	%	Controls the wet/dry mix, unless the 'Mix mode' is 'With feedback'.
Feedback	0	110	50	%	The delay feedback.
Tape length	1		250	ms	Sets the length of the tape i.e. the delay time at 1x speed.
Tape speed	-500	1000	0		Sets the tape speed (in conjunction with the CV input).

Routing parameters

Name	Min	Max	Default	Unit	Description
Left/mono input	1	28	1		The left input bus.
Right input	1	28	2		The right input bus.
Left/mono output	1	28	13		The left output bus.
Right output	1	28	14		The right output bus.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.
Speed input	0	28	0		The tape speed CV input bus.

Delayed Function

"Generates functions after a delay"

File format guid: 'delf'

Specifications:

- Channels, 1-8: The number of functions to generate.

Description

This algorithm generates a variety of one-shot functions after a delay, when triggered. The logic of each channel is as follows:

- When triggered, it immediately jumps to its idle voltage, if not already there.

- It stays at that voltage for the duration set by the Delay parameter.

- It then executes the function, for the duration set by the Time parameter, if relevant (for example, the Step function has no duration).

- Depending on the function, it then remains at the active or idle voltage until triggered again.

The available functions are:

Pulse	A simple rectangular pulse of the specified duration.
Step	A step – the output goes from idle to active after the delay, and remains at the active level until retriggered.
Ramp	A rising ramp of the specified duration. Stays at the active level after the ramp is complete.
Sawtooth	A rising ramp of the specified duration, which jumps back to the idle level when complete.

GUI

The display shows, for each channel, from top to bottom:

- The remaining time of the delay, once triggered.
- The remaining time of the function.
- The output function as a bar graph.
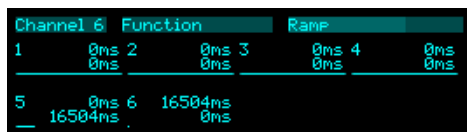
Common parameters

Name	Min	Max	Default	Unit	Description
Disable all	0	1	0		If set, disables all channels.
Detection	0	1	0		Sets the method of trigger detection, either 'Efficient' or 'Accurate'.

Per-channel parameters

Name	Min	Max	Default	Unit	Description
Enable	0	1	0		Enables the channel.
Function	0	3	0		Chooses the function from the table above.
Delay	0	1000	0		Sets the delay time.
Delay multiplier	0	6	0		Sets a multiplier for the delay time, one of x1, x2, x5, x10, x20, x50, or x100.
Time	1	1000	100		Sets the function duration.
Time multiplier	0	6	0		Sets a multiplier for the function duration, one of x1, x2, x5, x10, x20, x50, or x100.
Active voltage	-10.00	10.00	5.00	V	Sets the channel's active voltage.
Idle voltage	-10.00	10.00	0.00	V	Sets the channel's idle voltage.

Per-channel routing parameters

Name	Min	Max	Default	Unit	Description
Trigger input	0	28	1		The bus to use for the trigger input.
Output	0	28	15		The bus to use for the channel output.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.
MIDI channel	0	16	0		The MIDI channel on which notes can trigger the function generator.
I2C channel	0	255	0		The I2C channel on which notes can trigger the function generator.

Note number	-1	127	-1		The MIDI or I2C note number that will trigger the function generator, or '-1' for 'Any'.

Dream Machine

"Just intonation drone synth"

File format guid: 'drea'

Specifications: None

Description

This algorithm is an implementation of the original Dream Machine algorithm on the disting EX.

It is designed to generate drones, allowing the user to explore non-traditional harmonies based on prime ratios. It was inspired by the theories of composer [La Monte Young](#)[51]. An interesting read is the pdf "Notes on The Theatre of Eternal Music" available in a number of places online e.g. [here](#)[52].

The output is a combination of five sounds – the fundamental and four harmonies. The prime ratios that define the frequency relationships are controlled by parameters.

The algorithm uses wavetable synthesis to generate the tones. Additionally the fundamental may instead be a pure sine, triangle, or square wave.

Each tone has a simple attack/release envelope, controlled by its own gate parameter.

By default, little is mapped to the CV inputs, and it is perfectly possible to drive the algorithm entirely by hand. You may however like to map inputs as FM inputs, or to control the gates.

Setting the fundamental

The fundamental is the fixed tone (usually the bass note) that everything else revolves around. You may like to set it to a concert pitch (it defaults to concert B♭) or some other frequency (La Monte Young sometimes chose the mains frequency – 60Hz in the USA – or you could tune it to the resonant frequency of whatever environment you find yourself in).

Note that the Hz value shown in the display also takes into account the octave parameter.

Setting the primes

Four parameters let you choose the set of prime numbers that will make up the allowable values for the frequency ratios. La Monte Young famously chose the primes 2, 3, 7, & 31, and moreover specifically avoided the prime 5, thereby excluding major thirds from his tunings.

If you want less than four primes, set the unwanted parameters to '1'.

---

51 https://en.wikipedia.org/wiki/La_Monte_Young

52 https://www.melafoundation.org/theatre.pdf

Setting the frequency ratios

To set the ratios of tones 1-4 relative to the fundamental, set the parameters for the denominator and the four numerators.

When setting the numerators, the pitch of the tone is shown, as well as the ratio reduced to its lowest form. For example 48/32 reduces to 3/2, the familiar form of the perfect fifth in just intonation.

GUI

The display shows the current waveform within the wavetable on the right of the screen.

The rest of the screen shows the frequency, ratio, and gate level of the five tones (from left to right).
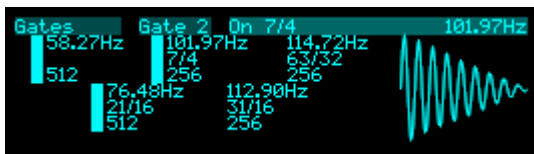
Wavetable parameters

Name	Min	Max	Default	Unit	Description
Wavetable					Chooses the wavetable for the oscillators.
Wave offset	-100.0	100.0	0.0	%	Sets the position within the wavetable.
Wave input	0	28	0		Sets an input bus to modulate the wave offset.
Waveform 0	0	3	0		Chooses the waveform for the fundamental. The options are "Wavetable", "Sine", "Triangle", and "Square".

Pitches parameters

Name	Min	Max	Default	Unit	Description
Prime 1	2	32749	2		Sets one of the four primes that may be multiplied to create the denominator and numerators of the frequency ratios.
Prime 2	1	32749	3		Sets the second prime.
Prime 3	1	32749	7		Sets the third prime.
Prime 4	1	32749	11		Sets the fourth prime.
Fundamental	0.001	32.767	29.135	Hz	Sets the fundamental frequency.

Octave	-8	8	1		Sets an octave shift for the fundamental.
Transpose	-60	60	0	ST	Adjusts the tuning of all frequencies in (12-TET) semitone steps.
Denominator	1	256	32		Sets the denominator of the frequency ratios.
Numerator 1	1	1024	42		Sets the numerator of the frequency ratio of tone 1.
Numerator 2	1	1024	56		Sets the numerator of the frequency ratio of tone 2.
Numerator 3	1	1024	62		Sets the numerator of the frequency ratio of tone 3.
Numerator 4	1	1024	63		Sets the numerator of the frequency ratio of tone 4.

Gates parameters

Name	Min	Max	Default	Unit	Description
Gate 0	0	1	0		Gate for tone 0.
Gate 1	0	1	0		Gate for tone 1.
Gate 2	0	1	0		Gate for tone 2.
Gate 3	0	1	0		Gate for tone 3.
Gate 4	0	1	0		Gate for tone 4.

Mix parameters

Name	Min	Max	Default	Unit	Description
Gain 0	-40	6	0	dB	Gain for the fundamental. "-40" is treated as –∞dB.
Gain 1	-40	6	0	dB	Gain for tone 1.
Gain 2	-40	6	0	dB	Gain for tone 2.
Gain 3	-40	6	0	dB	Gain for tone 3.
Gain 4	-40	6	0	dB	Gain for tone 4.
Pan 1	-100	100	0	%	Stereo pan position for tone 1.
Pan 2	-100	100	0	%	Stereo pan position for tone 2.

Pan 3	-100	100	0	%	Stereo pan position for tone 3.
Pan 4	-100	100	0	%	Stereo pan position for tone 4.

Other parameters

Name	Min	Max	Default	Unit	Description
Attack time	0	127	0		Attack time for the envelopes.
Decay time	0	127	0		Decay time for the envelopes.
FM input 1	0	28	0		Which input bus to use to frequency modulate (FM) tone 1. The input is scaled according to the FM Range parameter.
FM input 2	0	28	0		As above for tone 2.
FM input 3	0	28	0		As above for tone 3.
FM input 4	0	28	0		As above for tone 4.
FM Range	0	3	0		Sets the scaling for the FM inputs. The options are 1Hz/V, 10Hz/V, 100Hz/V or 1kHz/V.
Frequency slew	0	127	0		Sets a slew rate for any change in voice frequency.
Crossfade time	0	127	0		Sets a crossfade time for any change in voice frequency.

Routing parameters

Name	Min	Max	Default	Unit	Description
Left output	1	28	13		The left output bus.
Right output	1	28	14		The right output bus.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.

EQ Parametric

"Multi-band parametric EQ"

File format guid: 'eqpa'

Specifications:

- Channels, 1-12: The number of bus channels to process.
- Bands: 1-4: The number of EQ bands per channel.

Description

This algorithm is a multi-channel, multi-band, parametric EQ.

Each channel can process a number of busses - you might typically have a channel processing one or two busses for a mono or stereo signal, but you could, say, process all 12 of the inputs with the same EQ settings by using a single channel with a width of 12.

GUI

The display shows a graphical representation of the EQ curve for the current channel. The frequency and gain of the current band within the channel is indicated with a small box.
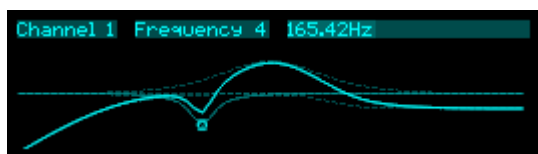
Common parameters

Name	Min	Max	Default	Unit	Description
Bypass	0	1	0		If on, the entire algorithm is bypassed.

Per-channel parameters

Name	Min	Max	Default	Unit	Description
Input	0	28	1		The first input bus to process.
Width	1	12	1		The number of busses to process, starting from the Input. For example, for a stereo signal, set this to 2.
Output	0	28	0		The first output bus. If set to 'None', the input bus is used as output, and the mode is always 'Replace'.

Output mode	0	1	1		The standard Add/Replace mode selector as described above.

Per-channel per-band parameters

Name	Min	Max	Default	Unit	Description
Enable	0	1	1		Enables the EQ band.
Type	0	6	6		Chooses the type of the EQ band. The options are "Low pass (1st)", "High pass (1st)", "Low pass (2nd)", "High pass (2nd)", "Low shelf", "High shelf", and "Peak".
Frequency	0	16383	8192		Sets the EQ band centre frequency.
Q	0	127	48		Sets the EQ band 'Q' or resonance. Note that some EQ types do not use this parameter.
Gain	-12.0	12.0	0.0	dB	Sets the EQ band gain. Note that some EQ types do not use this parameter.

Envelope (AR/AD)

"A simple attack/release envelope"

File format guid: 'env2'

Specifications:

- Channels, 1-8: The number of envelopes to generate.

Description

This algorithm generates simple attack/release or attack/decay envelopes. It is based on the disting mk4 algorithm "E-1 AR Envelope".

In the disting mk4 version the envelope times were always set together from a single knob; this version adds the option of setting them independently.

The envelope times and shapes are set globally; each output channel can however be independently scaled and offset.

In "Attack/release" mode, the envelope will rise to full level and stay there as long as the input is high. In "Attack/decay" mode, the envelope will execute one full attack/decay cycle in response to a trigger input. In "Looped AD" mode, the envelope will continually execute attack/decay cycles as long as the trigger input is high.

In "Macro (asymmetric)" mode, the "Joint time" parameter 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 "Macro (symmetric)" mode, the "Joint time" parameter sets the A & D times to the same value, from very short times (about 10ms) to very long times (about 8s). Note that the actual times are heavily dependent on the shape parameters.

GUI

The display shows the envelope shape. If the visible parameter page is one of the Channel pages, it also shows the current envelope level.
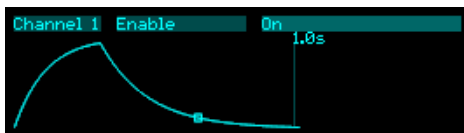
Parameters

Name	Min	Max	Default	Unit	Description
Trigger mode	0	2	0		The trigger mode. The options are: • Attack/release • Attack/decay • Looped AD
Time mode	0	2	0		The envelope time mode. The options are: • Macro (asymmetric) • Macro (symmetric) • Independent
Joint time	0	1023	0		Sets the attack and release times if the 'Time mode' is one of the two 'Macro' options.
Attack time	0	1023	64		Sets the attack time if the 'Time mode' is 'Independent'.
Release time	0	1023	64		Sets the release time if the 'Time mode' is 'Independent'.
Attack shape	0	128	104		Sets the attack shape, from an exaggerated exponential curve at 0 to an almost linear shape at 128.
Release shape	0	128	40		Sets the release shape, from an exaggerated exponential curve at 0 to an almost linear shape at 128.

Per-channel parameters

Name	Min	Max	Default	Unit	Description
Enable	0	1	0		Enables the channel.
Trigger input	0	28	1		Sets the trigger input bus.
Output	0	28	15		Sets the envelope output bus.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.
Amplitude	-10.00	10.00	8.00	V	Sets the amplitude of the envelope. Note that negative values give an inverted envelope.
Offset	-10.00	10.00	0.00	V	Sets a constant voltage offset for the envelope.
MIDI channel	0	16	0		The MIDI channel on which notes can trigger the envelope.

I2C channel	0	255	0		The I2C channel on which notes can trigger the envelope.
Note number	-1	127	-1		The MIDI or I2C note number that will trigger the envelope, or '-1' for 'Any'.

Euclidean patterns

"Generates Euclidean rhythm patterns"

File format guid: 'eucp'

Specifications:

- Channels, 1-8: The number of simultaneous patterns to generate.

Description

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[53] or here[54].

A pattern is described by the total number of steps and the number of pulses (i.e. the number steps on which a pulse is output).

The patterns can also be rotated. 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 later by a number of steps.

GUI

The display shows a graphical representation of the patterns (channel 1 at the top).
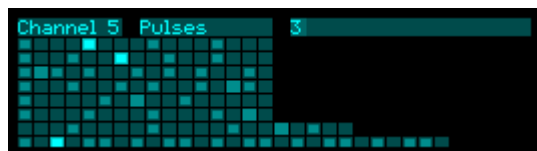
Globals parameters

Name	Min	Max	Default	Unit	Description
Clock input	1	28	1		The clock input bus. A rising edge advances the patterns by one step.
Reset input	0	28	0		The reset input bus. A high level resets the patterns to step 1 (and holds them there).

53 https://en.wikipedia.org/wiki/Euclidean_rhythm

54 https://www.hisschemoller.com/blog/2011/euclidean-rhythms/

Per-channel parameters

Name	Min	Max	Default	Unit	Description
Enable	0	1	0		Enables the channel.
Steps	1	32	16		The number of steps in the pattern.
Pulses	1	32	4		The number of pulses in the pattern.
Rotation	0	32	0		The rotation of the pattern.
Repeat	0	128	0		The overall repeat count i.e. the number of clocks until the pattern repeats. If this is zero, the number of steps is used as the repeat. If the repeat count is greater than the number of steps, the pattern is extended with silence.
Output type	0	1	0		Sets whether the output is a fixed length trigger, or a clock pulse related to the period of the input clocks.
Length	1	100	10	ms	Sets the trigger length, if the 'Output type' is 'Trigger'.
Length	1	99	50	%	Sets the output clock pulse width, if the 'Output type' is '% of clock'.

Per-channel routing parameters

Name	Min	Max	Default	Unit	Description
Output	1	28	15		The output bus for the channel.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.

Filter bank

"A fixed filter/resonator bank"

File format guid: 'fbnk'

Specifications:

- Filters, 1-12: The number of filters in the bank.

Description

This algorithm is based on the disting EX algorithm of the same name. You may like to review the in-depth video for that algorithm [here](#)[55].

The disting EX manual states "this algorithm provides a bank of eight parallel stereo bandpass filters or resonators. The filters' levels can be controlled manually, via CV, or with envelopes driven from gates, and their pitch can also be set manually or via CV or MIDI. The resonators in particular are good for being played as chords over MIDI, in the manner of the Alesis Quadraverb Plus (see e.g. [here](#)[56])."

This implementation goes further in that the number of filters can be increased to 12, and the filters can be mono, stereo, or up to 8 channels wide.

Please refer to the section "Common polysynth features" above.

Filter modes

The filters can be set to one of three modes: Resonator, Bandpass or Multiband.

Resonators are peaking all-pass filters that greatly emphasise the narrow band of frequencies around their centre frequencies.

Bandpass filters are a basic filter type that attenuate frequencies away from their centre frequencies.

In both of the modes above, all of the filters are completely independent. In Multiband mode however, the filter frequencies set a series of crossover points, such as you might find in a multiband compressor, for example. Therefore the frequencies passed by each band are bounded by its own frequency on the one side, and the frequency of the next band on the other.

---

55 https://www.youtube.com/watch?v=rn04rBEhIPM

56 https://www.youtube.com/watch?v=ZObDS3Hvfuo&t=90

GUI

The display shows the notes to which each filter is tuned, overlaid on a bar to represent the envelope level of the voice.
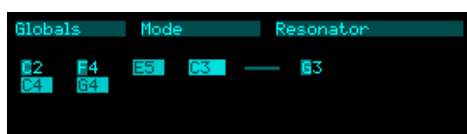
Globals parameters

Name	Min	Max	Default	Unit	Description
Audio input	1	28	1		The first audio input bus.
Width	1	8	1		The number of audio input busses (e.g. set to 2 for stereo).
Mode	0	2	0		The filter mode – see above.
Resonance/ Q	1	100	50	%	Sets the filter gain (for resonators) or resonance (for bandpass/multiband).
Attack time	0	1023	64		The envelope attack time. The scale is exponential, from 1ms up to 4s.
Release time	0	1023	64		The envelope release time. The scale is exponential, from 1ms up to 4s.
Gain	-40	24	0	dB	An overall gain control.

Per-channel parameters

Name	Min	Max	Default	Unit	Description
Pitch	0	127	0		The filter frequency/pitch, as a MIDI note number.
Gate	0	1	0		The filter gate.
Output	1	28	13		The filter's (first) output bus.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.
Gain	-40.0	24.0	0.0	dB	The filter's gain.

Microtuning parameters

The algorithm uses the standard polysynth microtuning parameters, as described above.

MIDI/I2C parameters

Name	Min	Max	Default	Unit	Description
MIDI channel	0	16	1		The MIDI channel to listen on.
MPE channels	1	16	1		Controls how the algorithm will respond to MPE. See above.
I2C channel	0	255	0		Sets the I2C channel.
Controlled voices	1	12	12		Sets the number of filters that will be controlled by MIDI or CV/gate. By reducing this, you can elect to control some filters by MIDI while the rest remain under manual control.
Transpose	-60	60	0	ST	Coarse tuning control.
Fine tune	-100	100	0	cents	Fine tuning control.
Sustain	0	1	0		Directly controls the sustain (like a MIDI sustain pedal).
Bend range	0	48	2		The MIDI pitch bend range.
Unison	1	8	1		The number of voices to play simultaneously for each note triggered.
Unison detune	0	100	10	cents	The detune amount when Unison is active.
Sustain mode	0	1	0		The standard polysynth sustain mode parameter. See above.

CV/gate parameters

The algorithm uses the standard polysynth CV/gate parameters, as described above.

Chord/arp parameters

The algorithm uses the standard polysynth chord and arpeggiator parameters, as described above.

Granulator

"A granulator"

File format guid: 'gran'

Specifications:

- Max buffer size, 1-32 seconds: The maximum size of the recording buffer.

Description

This algorithm is an implementation of the original Granulator algorithm on the disting EX.

It implements a granular synthesis[57] engine, taking as its source material either live audio input or audio loaded from the MicroSD card. Live audio can be recorded into a buffer or streamed continuously.

Granular synthesis works by playing many short snippets of sound, or 'grains', typically of the order of 100ms in length. Often various properties of the grains (e.g. their timing, length, pitch, stereo panning etc.) are randomised to some extent.

In this algorithm, the creation ('spawning') of grains is controlled by 'notes'. Notes control when grain clouds begin and end, and affect other features e.g. the grain pitch.

Notes can be played via CV/gate pairs, or MIDI, or I2C. The algorithm also offers three 'drone' voices, which can simply be enabled via the usual parameter interface. When using the algorithm as an audio processing effect you're likely to just enable one or more of these drones and leave them on while manipulating other grain parameters.

The algorithm does nothing until you've recorded some audio into it or loaded some audio from the SD card.

Two suggested ways of getting started:

1) Connect an audio input, enable Record and enable Drone 1.

2) Connect CV/gate or MIDI, load a sample from the card, and play.

You may like to review the in-depth videos about the disting EX algorithm - the YouTube playlist is [here](#)[58].

Effects and routing

Those familiar with the disting EX version of this algorithm may wonder where the delay and reverb

---

57 Essential reading on granular synthesis includes Microsound by Curtis Roads, https://mitpress.mit.edu/9780262681544/microsound/

58 https://www.youtube.com/playlist?list=PLlY5j4QDwxWtIlNfDq_dCdryxR2hXTqvg

effects have gone. You can of course simply add them as extra algorithms on the disting NT, and do so with more routing flexibility.

The "Main feedback" (which echoes the granulator output to the buffer input) and "Input feedback" (which effectively applies an echo on the input signal) paths are still available.

GUI
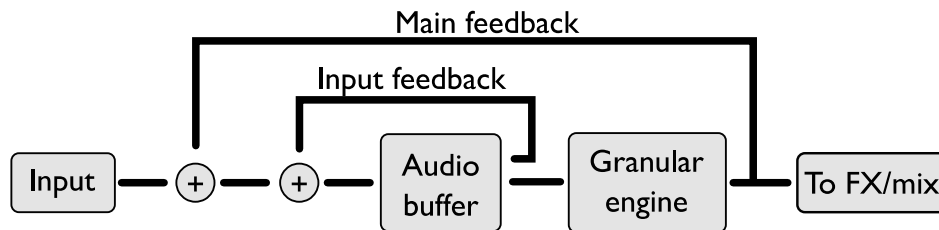
The display shows a waveform representation of the audio buffer, on which are superimposed dots indicating the currently playing grains. The two dotted lines represent the "Size mean" (by their separation) and the "Delay mean" (by their position within the buffer).

On the right of the screen you can see the current buffer size, the current delay, and the number of grains currently playing.
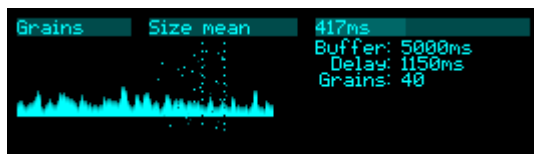
Loading/saving audio

Functions for loading and saving audio from/to the MicroSD card are available through the 'Granulator' menu.

Once a sample has been loaded via the menu, the algorithm remembers it, and displays its name in the GUI:
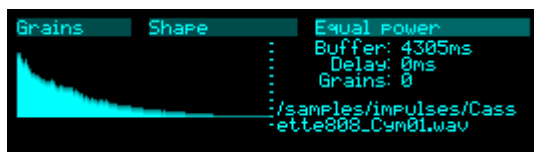
If the preset is then saved and reloaded, the sample will also be reloaded into the buffer.

Load sample

This submenu offers four versions of loading a WAV file from the MicroSD card into the Granulator's audio buffer.

- **Into buffer/As buffer** - The 'into buffer' options keep the buffer size as-is, and load the sample into it. If the sample is shorter than the buffer, the remaining space is cleared. The 'as

Page 109

buffer' options change the buffer length to match the sample length (up to the maximum buffer size as per the specifications).
- **& Normalize** - The options with '& normalize' post-process the sample to normalize its level so the peak signal level is the nominal maximum. Can be useful when loading quiet samples.

Forget sample

This tells the algorithm to forget the sample location, so it will not be reloaded when the preset is loaded. The audio remains in the buffer, however.

Save recording

This saves the current buffer contents to the MicroSD card as a WAV file.

The file is saved into a folder called 'granulator', which will be created if it doesn't already exist.

The filename is 'saved <date and time>.wav' where 'date and time' come from the real-time clock.

Mix parameters

Name	Min	Max	Default	Unit	Description
Input gain	-34	12	0	dB	Gain applied to the audio being recorded (does not affect the dry signal).
Dry gain	-40	6	0	dB	Level of the input signal in the output mix.
Granulator gain	-40	6	0	dB	Level of the granulator signal in the output mix.
Input feedback	0	100	0	%	The amount of feedback to apply around the audio buffer itself when recording (resulting in an echo effect on the input material, with a delay time equal to the buffer size).
Main feedback	0	100	0	%	The amount of the granulator output to feed back into the audio buffer when recording.
Normalize	0	1	1		If enabled, the overall volume of the grain cloud is lowered according to how many grains are active.

Record parameters

Name	Min	Max	Default	Unit	Description
Record	0	1	0		Enables recording into the buffer.
Buffer size	100			ms	The audio buffer size in milliseconds.
Record	0	1000	5	ms	The duration of the fade applied when starting and

					stopping recording, to avoid clicks.
fade					

Grains parameters

Name	Min	Max	Default	Unit	Description
Shape	0	5	0		The grain envelope/window shape. See below.
Spawn mode	0	4	0		How grains are spawned. See below.
Rate mean	1	1000	5	ms	The average time between new grains being spawned.
Rate spread	0	200	10	%	The amount of variation in the spawn rate, expressed as a percentage of 'Rate mean'.
Size mean	0	1000	100	ms	The average grain size. '0' has a special meaning – see below.
Size spread	0	200	10	%	The amount of variation in grain size, expressed as a percentage of 'Size mean'.
Pitch mean	-24	24	0	ST	The average grain pitch shift (in semitones). This is added to the pitch shift determined by the note's pitch CV.
Pitch spread	0	1200	0	cents	The amount of variation in grain pitch shift.
Pan mean	-100	100	0	%	The average grain pan position.
Pan spread	0	100	10	%	The amount of variation in grain pan.
Delay mean	0	100	50	%	The average grain delay (equivalently, the position in the audio buffer), expressed as a percentage of the buffer size.
Delay spread	0	100	5	%	The amount of variation in grain delay, expressed as a percentage of the buffer size.
Reverse	0	100	0	%	Sets the probability that a grain will be played backwards.
Grain limit	1	40	40		Imposes an arbitrary limit on the number of simultaneous grains.
Natural pitch	0	127	48	ST	Sets the natural pitch of the audio i.e. the MIDI note number that will play back the audio at the same pitch at which it was recorded.
Opacity	0	100	100	%	The 'opacity' of a note, which is the percentage of

					grains that would normally make up the note that actually sound.

Modulation parameters

Name	Min	Max	Default	Unit	Description
LFO depth	-100	100	0	%	The depth of the LFO that affects the grain delay, expressed as a percentage of the buffer size.
LFO speed	0	255	196		The speed of the grain delay LFO. This is scaled relative to the buffer size – at the default value of 196 the LFO will cause the 'play head' (to use a tape metaphor) to advance at 1x speed.
LFO shape	0	2	0		Sets the LFO shape. The options are "Triangle", "Ramp up", and "Ramp down".
Attack time	0	127	64		The envelope attack time, from 100ms to 30s with an exponential scaling.
Release time	0	127	64		The envelope release time, from 100ms to 30s with an exponential scaling.
Env -> opacity	0	100	0	%	The amount by which the note envelope affects the note opacity.
Env -> level	0	100	100	%	The amount by which the note envelope affects the note level (volume).
Veloc -> level	0	100	100	%	The amount by which the note velocity affects the note level (volume).
Veloc -> delay	0	100	0	%	The amount by which the note velocity affects the grain delay.
Pitch -> pitch	0	100	100	%	The amount by which the note pitch affects the grain pitch. Commonly this will either be 100% (normal pitch tracking) or 0% (the incoming pitch doesn't affect the grain pitch at all, but may still affect e.g. the grain delay).
Pitch -> delay	0	100	0	%	The amount by which the note pitch affects the grain delay.

Drone 1-3 parameters

Name	Min	Max	Default	Unit	Description
Drone 1-3	0	127	48/36/60	ST	The MIDI note number for the drone. Remember

pitch					that 'Natural pitch' sets how these are interpreted. For the default Natural pitch of 48, done pitch 48 is the same pitch, drone pitch 36 is an octave down, and drone pitch 60 is an octave up.
Drone 1-3 enable	0	1	0		Enables (gates) the drone.
Drone 1-3 opacity	0	100	100	%	The opacity of the drone.
Drone 1-3 level	-40	6	0	dB	The level (volume) of the drone.

Setup parameters

Name	Min	Max	Default	Unit	Description
MIDI channel	0	16	0		The MIDI channel to listen on.
MPE channels	1	16	1		Controls how the algorithm will respond to MPE. See above.
I2C channel	0	255	1		Sets the I2C channel.
Bend range	0	48	2	ST	The MIDI pitch bend range.
Delay unit	0	1	0		Determines whether the Delay mean and spread parameters work in terms of % or ms.

Routing parameters

Name	Min	Max	Default	Unit	Description
Left input	1	28	1		The left audio input bus.
Right input	1	28	1		The right audio input bus.
Left output	1	28	13		The left audio output bus.
Right output	1	28	14		The right audio output bus.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.
Delay mean input	0	28	0		The input bus to use to control the grain delay. A CV of 5V corresponds to 100% of the buffer size.

Delay mean sampled	0	1	0		If enabled, the Delay mean input is sampled once at the start of a note. Otherwise, it is sampled afresh as every grain spawns.

CV/gate parameters

The algorithm uses the standard polysynth CV/gate parameters, as described above.

Spawn mode

The spawn mode parameter controls the algorithm by which new grains are spawned. The options are:

- **Stochastic** (the default): grains are spawned randomly according to the 'Rate mean' and 'Rate spread' parameters.
- **Mid-grain**: a new grain is spawned at the middle point of the current grain. This is intended to be used with 'Size spread' at zero and with 'Shape' as 'Equal power', resulting in smooth crossfade looping, but you are of course free to use it creatively as you wish.
- **Single**: exactly one grain is spawned when a note is triggered. Essentially this gives you manual control over when grains are spawned.

The Stochastic and Mid-grain options also have 'fixed' variants ('Stochastic (fixed)' and 'Mid-grain (fixed)'). These behave identically unless recording is active. The difference relates to how the grain positions are calculated, using the delay mean and spread parameters.

In the non-fixed modes, the grain positions are relative to the current audio buffer write position – effectively, relative to 'now'.

In the fixed modes, the grains' positions are relative to the buffer write position at the time the note started. Conceptually, the grains are fixed relative to the audio, and indeed if you watch the grains spawn in the visual display you'll see them travelling across the screen as the audio scrolls.
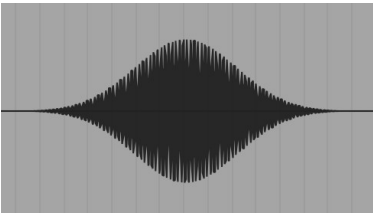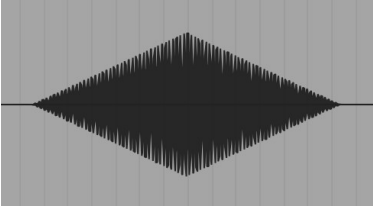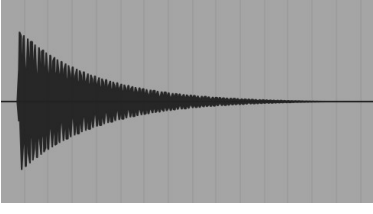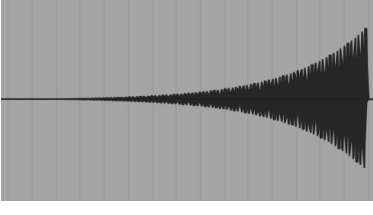
In practice this means that in the 'fixed' modes, notes tend to have a fixed timbre for their duration, whereas in the non-'fixed' modes the timbre will vary as the audio scrolls past under the note position.
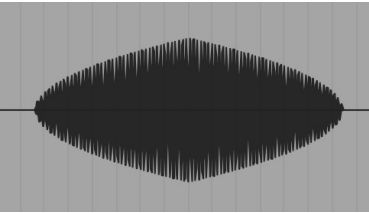
Size mean '0'

If the 'Size mean' parameter is set to zero, the grain size is calculated from the note pitch, and set to be the duration of one cycle of an oscillation that corresponds to the given pitch. This is particularly effective in conjunction with the 'Mid-grain (fixed)' spawn mode.

Shape

The 'Shape' parameter sets the volume envelope (also called 'window' in some of the literature) of the grains. The options are as follows.

Value	Name	Description	Image
0	Gaussian	A gaussian bell curve.	
1	Tukey	A rectangle convolved with a raised cosine.	
2	Triangle	A simple triangle shape.	
3	Expodec	A decaying exponential curve.	
4	Rexpodec	A rising exponential curve.	

5	Equal power	Back-to-back square root curves.	

Kirbinator

"Stochastic audio processor"

File format guid: 'krby'

Specifications:

- Buffer size, 1-44 seconds: The maximum size of the audio buffer.

Description

This algorithm continuously records audio into a buffer and plays slices of the buffer under the influence of various probabilities. The slices can be pitched up and down, played forwards and backwards, and panned in stereo.

Key to the operation of the algorithm are two trigger inputs, the Mark input and the Trigger input. These can be the same signal, but more interesting results often arise when they're not. Typically both signals would be regular clocks, but this is by no means a requirement.

The Mark input defines the slices in the audio buffer. Say you have a rhythmic pattern going into the audio input, and the Mark signal is a (synchronised) clock. If that clock is a quarter note clock, the audio slices will be a quarter note long; if the clock is faster, say an eighth note, then the slices will be shorter. If your audio pattern is playing eighth notes and the clock is an eighth note, then one slice will be one note.

The Trigger input tells the Kirbinator when to choose a new slice to play and to change its playback parameters. In our example before, if the Trigger were a quarter note clock, the playback would jump and potentially change pitch and direction on every beat.

Note that playback continues with the new parameters until another Trigger is received; it doesn't, say, play one slice and then stop.

The audio buffer is stereo, but the algorithm can be used as mono, stereo, or mono-in stereo-out, according to the routing parameters.

GUI

The display shows a waveform representation of the audio buffer. The shorter, darker line represents the current recording position. The longer line or lines show the current playback position (if any). The small checkmarks along the top of the waveform indicate the 'marks' defined by the Mark input.
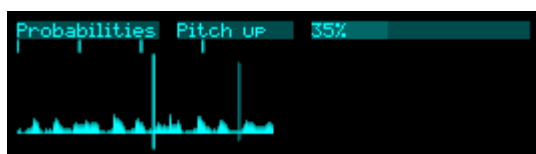
Probabilities parameters

Name	Min	Max	Default	Unit	Description
Pitch up	0	100	0	%	The probability that playback will be pitched up.
Pitch down	0	100	0	%	The probability that playback will be pitched down.
Fifths	0	100	0	%	The probability that, if playback is pitched up or down, it will be by a perfect fifth; otherwise, it will be by an octave.
Reverse	0	100	0	%	The probability that playback will be reversed.
Glide	0	100	0	%	The probability that a glide will be applied to any pitch change.
Stutter	0	100	0	%	The probability that playback of the new slice will be stuttered, that is, that the initial part of the playback will be repeated.
Triplet	0	100	0	%	If 'Metrical' stutter has been selected, the probability that a stutter will be a triplet division, rather than a division by a power of 2.
Play	0	100	100	%	The probability that playback will actually happen; otherwise, playback stops.

Jumps parameters

Name	Min	Max	Default	Unit	Description
Glide	0	1000	100	ms	The time of the glide time to apply, if a glide is activated by the probability.
Through zero	0	1	1		Whether glides are allowed when the playback direction reverses. Such glides can produce a very strong "tape stop" effect.
Min stutter	1	16	2		If a stutter is activated, the minimum number of restarts.
Max stutter	1	16	3		If a stutter is activated, the maximum number of restarts.
Min jump	0	16	0		The minimum number of slices away from the current record position to jump.
Max jump	0	16	0		The maximum number of slices away from the current record position to jump.
Stutter	0	1	1		Sets whether stutter is 'Free' (unconstrained apart

					from by the minimum and maximum parameters), or 'Metrical' (restricted to powers of 2 and 3, according to the Triplet probability).

Tweaks parameters

Name	Min	Max	Default	Unit	Description
Pitch offset	-24	24	0	ST	Applies a constant pitch change to playback, added to any probabilistic pitch change.
Glide offset	-1000	1000	0	ms	Applies a constant glide time, added to any probabilistic glide.
Fade	0	1000	10	ms	Sets the fade in/out time for slice playback.

Mix parameters

Name	Min	Max	Default	Unit	Description
Dry gain	-40	6	-40	dB	The level of the input signal passed through to the output.
Effect gain	-40	6	0	dB	The output level of the Kirbinator effect.
Pan mean	-100	100	0	%	The average pan position of slice playback.
Pan deviation	0	100	0	%	The amount of random deviation away from the average pan position, calculated for each trigger.

Routing parameters

Name	Min	Max	Default	Unit	Description
Left/mono input	1	28	1		The left or mono audio input.
Right input	0	28	0		The right audio input.
Left/mono output	1	28	13		The left or mono audio output.
Right output	0	28	0		The right audio output.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.
Mark input	0	28	3		The bus to use as the Mark signal.

Trigger input	0	28	3		The bus to use as the Trigger signal.

LFO

"A low frequency oscillator"

File format guid: 'lfo '

Specifications:

- Channels, 1-4: The number of LFO outputs.

Description

This algorithm offers a number of flexible low frequency oscillators.

The global 'Quality' parameter offers two options:

- 'Stepped' quality is extremely low CPU usage, and is appropriate for automating parameters of other algorithms (via mapping).
- 'Linear' quality should be selected if you plan to use the LFOs to output CVs to other modules.

The LFOs can run freely, or be synced to external clocks or to MIDI clock.

Phase-locking

When there is more than one LFO channel, channels 2-4 can be locked to the phase of LFO 1. This allows you to, for example, create a quadrature LFO.

Use the 'Phase' parameter of LFOs 2-4 to set the phase relationship in degrees to LFO 1. Note that in this case, the 'Speed' and 'Multiplier' parameters of the phase-locked LFO have no effect.

GUI

The display shows the shape of the current channel's output, or if the 'Globals' page is active, the shape of all the channels.
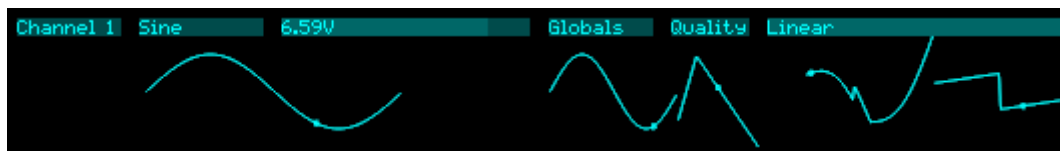
Globals parameters

Name	Min	Max	Default	Unit	Description
Quality	0	1	1		Sets the LFO output quality. The options are "Stepped" and "Linear".

Clock input	0	28	0		The bus to use as the clock input.

Per-channel parameters

Name	Min	Max	Default	Unit	Description
Enable	0	1	0		Enables the channel.
Speed	0	16383	8605		Sets the LFO speed, from 0.05Hz to 15Hz with an exponential scaling.
Multiplier	0	3	1		Sets a multiplier for the LFO speed. The options are "x0.1", "x1", "x10", & "x100".
Sine	-10.00	10.00	0.00	V	Sets the amplitude of a sine wave to add into the LFO output.
Triangle	-10.00	10.00	0.00	V	Sets the amplitude of a triangle wave to add into the LFO output.
Saw	-10.00	10.00	0.00	V	Sets the amplitude of a saw wave to add into the LFO output.
Square	-10.00	10.00	0.00	V	Sets the amplitude of a square/pulse wave to add into the LFO output.
Pulse width	0.0	100.0	50.0	%	Sets the pulse width of the square/pulse wave.
Offset	-10.00	10.00	0.00	V	Adds a constant voltage to the LFO output.
Asymmetry	-100.0	100.0	0.0	%	Sets the asymmetry. This is somewhat like applying a pulse width variation to a square wave, but applies equally to all the waveforms.
Random	-10.00	10.00	0.00	V	Adds a stepwise random voltage to the LFO output.
Phase	0	360	0		Phase-locks LFOs. See above.

Per-channel routing parameters

Name	Min	Max	Default	Unit	Description
Output	0	28	15		The LFO output bus.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.

Per-channel sync parameters

Name	Min	Max	Default	Unit	Description
Sync	0	4	0		Selects how the LFO will sync to external clocks or MIDI. See below.
Clock multiplier	0	23	15		Sets the clock multiplier, if syncing to external clocks.
MIDI divisor	0	19	9		Sets the MIDI clock divisor, if syncing to MIDI clock.

LFO sync modes

The Sync parameter allows LFOs to synchronise themselves to external clocks or to MIDI clock. The options are:

Clock	The LFO syncs its speed to a clock on the bus specified by the 'Clock input' parameter.
Clock with reset	As 'Clock', but the clock also causes the LFO to reset its phase.
MIDI tempo	The LFO syncs its speed to incoming MIDI clock.
MIDI transport	The LFO syncs its speed and phase to incoming MIDI clock.

Linear/Exponential

"Converts e.g. V/Oct <-> Hz/V"

File format guid: 'lexp'

Specifications: None

Description

This algorithm implements a linear-to-exponential converter and an exponential-to-linear converter, typically used when interfacing between Volt/octave and Hz/Volt CV standards.

The 0V point for the V/octave signals is taken as C3 (130.8Hz approximately).

GUI

The display shows the input and output voltages, and the internal calculated frequency, for each converter.
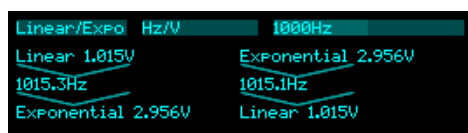
Linear/Expo parameters

Name	Min	Max	Default	Unit	Description
Hz/V	1	2000	1000	Hz	The Hz/Volt scaling.
Offset	-1000	1000	0	mV	A voltage offset subtracted from the input.
Input	0	28	0		The input (linear, Hz/V) bus.
Output	0	28	0		The output (exponential, V/octave) bus.
Output mode	0	1	1		The standard Add/Replace mode selector as described above.

Expo/Linear parameters

Name	Min	Max	Default	Unit	Description
Hz/V	1	2000	1000	Hz	The Hz/Volt scaling.
Offset	-1000	1000	0	mV	A voltage offset added to the output.

Input	0	28	0		The input (exponential, V/octave) bus.
Output	0	28	0		The output (linear, Hz/V) bus.
Output mode	0	1	1		The standard Add/Replace mode selector as described above.

Looper

"A looper"

File format guid: 'loop'

Specifications:

- Loops, 1-4: The number of simultaneous loops.
- Max time, 1-90: The maximum total loop time

Description

This algorithm is a looper, based on the disting EX algorithm of the same name, but significantly extended. You may like to review the video about the EX version, which is [here](#)[59].

Loops can have crossfades, they can fade in and out, and overdubbing can fade in and out, all of which is designed to make it easy to achieve smooth, ambient looping.

All looping operations can also be synchronised to a clock input, if tight rhythmic looping is more your thing.

Loops may also be reversed, and played at half speed (an octave down).

A 'layers' system is also implemented, which is like the 'undo/redo' found on some loopers but more powerful.

The core control setup is a two button record/play/overdub paradigm in the manner of many stomp-box loopers, though many more controls are available. These can be mapped to CV, MIDI, or I2C as usual; a dedicated MIDI-note based interface is also provided for ease of use with MIDI pedalboards which send notes when footswitches are pressed.

You may like to review the Debouncer algorithm (above) which is designed to make it easy to hook up guitar footswitches to the looper controls.

A UI script (see below for information about UI scripts) is available which puts the looper's main record/play buttons on the module's two encoders. This is available from [our GitHub](#)[60] if it's not already on your MicroSD card.

Available loop time

The total time set in the specifications is shared equally between the loops, so if you choose to have one loop with a 'max time' of 90 seconds, that loop can be up to 90 seconds, but if you choose to have four loops with the same 'max time', each loop can only be 22.5 seconds long.

The actual maximum time for each loop also depends on its bit depth and whether it's stereo or mono. The quoted time in the specification is for mono 32 bit. Making a loop stereo halves its maximum

---

59 https://www.youtube.com/watch?v=ZkL7v5CCzgc

60 https://github.com/expertsleepersltd/distingNT

length; making it 16 bit doubles the maximum length.

Be aware that clipping can occur when using 16 or 24 bit modes, so watch your input levels. No clipping can occur when using 32 bit float.

Changing parameters that affect a loop's maximum memory cause an instant clear of the loop, reverting it to the 'empty' state. These parameters are 'Channels', 'Bit depth', and 'Minimum layers'.

Loop targets and commands

Fundamental to the operation of the Looper is the concept of the 'target loop'. This is how we expose the control of up to four independent loopers through a two button interface. The record/overdub/play/etc. operations are considered as 'commands' that apply to whichever loop or loops is the 'target'.

The target loop is set by the 'Target loop' parameter. The commands are given by the remaining parameters on the 'Controls' page. The command is given when the parameter changes from 'Released' to 'Held'.

Each command parameter has an equivalent MIDI note parameter on the 'MIDI' page.

All commands are simple toggles, with the exception of the 'Target 1/2/3/4' commands. While at least one of these is held, holding another adds to the targetted loops. When all are released, the next target held replaces the target selection. Thought of in terms of MIDI notes, this means you can hold 'chords' to select multiple target loops at once.

GUI

The display shows a considerable amount of information about the loop(s). A special large version of the GUI is shown if there is only a single loop.

The target loops are indicated with a box drawn around their section of the display.

Each loop shows an icon to indicate its state, as follows:

- Square: Empty
- Circle: Record (or Record x-fade if the circle is pulsing)
- Triangle: Play (reversed if triangle points left; octave down if a small hole in the triangle)
- Triangle and circle: Overdub
- Two vertical lines: Stop
- Two vertical lines with downward triangle: Fade to stop, or Fade to clear
- 'M' instead of two vertical lines indicates 'mute' rather than 'pause'

The length of the loop, and the position within the loop, are shown top right (both in seconds).

The loop's envelope level is indicated by the vertical bar at the right side.

Basic looping

The procedure is as follows:

- Connect an input signal.

- Choose the target loop.

  The display will initially show a square, which is the 'stopped' symbol. The target loops are shown with an outline.

- Press 'Record' to begin recording.

  The symbol for the target loop will change to a circle, for 'recording'. The recording time will start counting up in the upper right of the display.

  We say 'Press Record' etc. here – obviously there's not a dedicated record button. By this we mean activate record however you've chosen to do it – with a MIDI note, a CC or CV mapped to the parameter, via the UI script, or simply by manually changing the parameter.

- Press 'Record' again to end recording and immediately enter playback.

  The symbol for the loop will change to a triangle, for 'play'. The upper right area shows the current time in the loop, and the loop length. The moving line on the waveform display also shows the loop playback position.

- If the recording approaches the maximum loop time, the upper left of the display will show 'remaining' and the remaining loop time.

  Recording will automatically stop if it reaches the maximum time.

- Recording can also be ended by pressing 'Play'.

- If 'Record' is held down for longer than the time set by the 'Record hold time' parameter, releasing 'Record' is then treated as if you'd pressed it again. To reiterate – recording can be done by either tapping 'Record' twice (once to start and once to stop), or by holding it once.

Recording a blank loop

Sometimes it is useful to set the loop length without actually recording anything into it. To do this, press 'Play' instead of 'Record' to begin recording.

The display will show the record icon (circle) with an 'M' for mute.

Pausing/muting/retriggering the loop

Once a loop is playing, pressing 'Play' pauses or mutes the loop, depending on the 'Pause/mute' parameter.

If the loop is paused, the symbol changes to the 'double vertical line' icon.

If the loop is muted, the symbol changes to an 'M'.

Pressing 'Play' again returns the loop to play mode.

A third option, 'Retrigger', is like Mute except that when the loop starts playing again it does so from the beginning of the loop.

Loops may also be retriggered with the 'Retrigger' control.

Overdubbing

While a loop is playing, pressing 'Record' enters overdub mode. Incoming material is recorded on top of the previous loop. The loop's symbol shows both the play (triangle) and record (circle) icons.

Press 'Record' again to stop overdubbing.

The 'Overdub fade' parameter controls how previous material fades out during overdubbing, if at all. If you're building up a rhythmic part piece by piece, you would probably leave this at 0dB i.e. no fade. If you're layering up some swooshy ambience, you probably want this down at maybe -3 to -6dB.

The 'Record hold time' parameter also applies to overdubbing, that is, you can enter and leave overdub by holding down 'Record' instead of tapping it twice.

Clearing the loop

When a loop is paused or muted, you can clear the loop, which means erasing it completely and returning to 'stopped'.

To do so, hold down 'Record'.

While you hold the button, the word 'CLEAR' will appear with a graphical countdown. When the countdown reaches zero, the loop will be cleared.

Crossfades

Each loop can have a crossfade time set, which helps to make smooth loops. Note that raising the crossfade time effectively reduces the loop time, since two passes of the loop have to be overlaid and mixed.

If the crossfade time is set before recording the loop, the recording will run on after the end-of-record command, in order to record extra material for the crossfade. The overall loop time will still be correct i.e. the time between the start- and end-of-record commands.

Envelopes

Two attack-decay envelopes are provided; one for when a loop is started and stopped, and one for

when overdubbing starts and ends.

While the loop is in the decay stage, fading out towards pause/mute, a downwards-pointing triangle is shown next to the icon in its display.

While the end of overdub decay stage is active, the record indicator (the circle) pulses.

Clocked operation

The algorithm has a 'Clock source' parameter, which can be 'None', 'Clock input', or 'MIDI clock'. If a clock source is chosen, any looping command is deferred until the next clock that arrives. This makes it possible to synchronise the loop length, and the starting and stopping of playback etc., with a clock.

When in clocked mode, the 'Lock range' parameter activates automatic re-triggering of the loops to keep them locked to the clock. The parameter specifies a time window around the clock – if a loop would naturally loop during this window, it is retriggered so that it begins exactly on the clock pulse.

The 'Bypass clock' parameter is provided to allow the clocks to be temporarily ignored (for example if you want to implement an "instant stop" even if clocked operation is active).

Stop all

When 'Stop all' is pressed, all currently playing loops are selected as targets, and stopped. When released, all stopped loops (including loops that were already stopped before the note was sent) are selected as targets (which can then all be conveniently cleared if desired).

Layers and Undo/redo

If layers are enabled, each time an overdub is started the contents of the loop are saved as a new layer, so you can go back to it at a later time. This is like the undo/redo system offered by some stomp-box loopers, but rather than there being a single undo, you can go up and down a whole stack of layers.

You can move between layers with the 'Undo' and 'Redo' commands, or by directly manipulating the 'Current layer' parameter. If you 'Undo' on the first layer this is treated as a 'Redo' – this is a convenience for situations where you only have a single pedal/button set up.

The 'Layer fade' parameter sets the duration of a crossfade that is applied when moving between layers.

A new layer is started only when overdub is activated, not every time around the loop while you remain in overdub.

The number of available layers depends on the available memory. If, say, you have 60 seconds of loop memory, and record a 10 second loop, then you can use up to 6 layers – but if you record a loop longer than 30 seconds, only one layer will be possible.

If you're already on the last available layer when you activate overdub, your overdub will be recorded into the existing layer. You can still undo to the previous layer, but redoing to the last layer will bring

back the recording after the second overdub.

The 'Minimum layers' parameter tells the looper how many layers you want to reserve space for. For example, if you have 60 seconds of loop memory and request a minimum of 4 layers, then the longest loop you can record will be 15 seconds.

If you set 'Minimum layers' to 'Layers disabled', layers won't be used at all. Starting an overdub will not create a new layer, even if there's enough memory to do so.

Controls parameters

Name	Min	Max	Default	Unit	Description
Target loop	1		1		Sets the target loop(s).
Record	0	1	0		Activates record or overdub.
Play	0	1	0		Activates play or stop.
Reverse	0	1	0		Activates 'reverse' (toggles the target loops between playing forwards and backwards).
Octave down	0	1	0		Activates 'octave down' (toggles the target loops between playing at normal speed and half speed).
Retrigger	0	1	0		Activates retrigger.
Stop all	0	1	0		Activates 'stop all'. See above.
Bypass clock	0	1	0		While this parameter is held, commands such as Record, Play, etc. behave as if the 'Clock source' parameters were 'None'.
Undo	0	1	0		Activates 'Undo'. See Layers, above.
Redo	0	1	0		Activates 'Redo'. See Layers, above.
Target 1	0	1	0		Activates loop 1 as a target loop.
Target 2	0	1	0		Activates loop 2 as a target loop.
Target 3	0	1	0		Activates loop 3 as a target loop.
Target 4	0	1	0		Activates loop 4 as a target loop.
Target next	0	1	0		Activates the loop after the current target as the new target.
Fade to clear	0	1	0		Activates 'fade to clear' – the same as pressing Play to stop the loops, except that the loops will be automatically cleared when the fade out has completed.

Globals parameters

Name	Min	Max	Default	Unit	Description
Pause/mute	0	2	0		Controls what happens when a loop is stopped. See above.
Attack time	0	1000	0		The loop envelope attack time. The scale is exponential, from 1ms to 30s.
Decay time	0	1000	0		The decay envelope attack time. The scale is exponential, from 1ms to 30s.
Attack shape	0	100	0	%	The loop envelope attack stage shape.
Decay shape	0	100	0	%	The loop envelope decay stage shape.
Overdub attack time	0	1000	0		The overdub envelope attack time. The scale is exponential, from 1ms to 30s.
Overdub decay time	0	1000	0		The overdub envelope attack time. The scale is exponential, from 1ms to 30s.
Overdub fade	-24.0	0.0	0.0	dB	Sets how much the previous loop content will fade out on each pass while overdubbing.
Retrigger fade	0.5	100.0	2.0	ms	Sets the duration of the crossfade applied when retriggering.
Layer fade	1	1000	10	ms	Sets the duration of the crossfade when changing layers.
Clear hold time	0.5	5.0	1.0	s	Sets how long Record needs to be held before a Clear happens.
Record hold time	0.5	5.0	1.0	s	Sets how long Record needs to be held before releasing it is treated as pressing it again.
Controls wake UI	0	1	0		If on, activating any looper control wakes the UI and dismisses the screensaver, as if you'd moved an encoder/pot/button on the module itself.

Sync parameters

Name	Min	Max	Default	Unit	Description
Clock source	0	2	0		The clock source for synchronisation – one of 'None', 'Clock input', or 'MIDI clock'.

Clock input	1	28	3		The input bus to use as clock if the 'Clock source' is 'Clock input'.
MIDI divisor	0	19	9		The MIDI clock divisor to sync to, if the 'Clock source' is 'MIDI clock'.
MIDI numerator	1	16	1		A multiplier for the MIDI clock. For example, if your song is in 5/4 time, and you want to sync the loops to whole bars, you could set the divisor to 1/4 and the numerator to 5.
Lock range	0	1000	0	ms	See 'Clocked operation', above.

Per-loop parameters

Name	Min	Max	Default	Unit	Description
Channels	0	1	0		Selects stereo or mono operation.
Bit depth	0	2	0		Selects '32 bit (float)', '24 bit', or '16 bit'.
Crossfade time	0	1000	70		Sets the loop crossfade time. The scale is exponential, from 0.5ms to 5s.
Minimum layers	0	8	0		The minimum number of layers the loop should have, or '0' for 'Layers disabled'.
Current layer	1	999	1		The current layer for the loop.

Per-loop routing parameters

Name	Min	Max	Default	Unit	Description
Input left/mono	1	28	1		The input bus for the left (if stereo) or mono signal.
Input right	1	28	2		The input bus for the right signal, if stereo.
Output left/mono	1	28	13		The output bus for the left (if stereo) or mono signal.
Output right	1	28	14		The output bus for the right signal, if stereo.
Output mode	0	1	0		The standard Add/Replace mode selector as described above, for the loop outputs.
Trigger output	0	28	0		The output bus for an end-of-loop trigger pulse.

Trigger mode	0	1	0		The standard Add/Replace mode selector as described above, for the trigger output.

MIDI parameters

Name	Min	Max	Default	Unit	Description
MIDI channel	0	16	0		The MIDI channel on which to receive notes.
Target 1	0	127	48		The MIDI note to automate 'Target 1'.
Target 2	0	127	50		The MIDI note to automate 'Target 2'.
Target 3	0	127	52		The MIDI note to automate 'Target 3'.
Target 4	0	127	53		The MIDI note to automate 'Target 4'.
Target next	0	127	54		The MIDI note to automate 'Target next'.
Record	0	127	55		The MIDI note to automate 'Record'.
Play	0	127	57		The MIDI note to automate 'Play'.
Reverse	0	127	59		The MIDI note to automate 'Reverse'.
Octave down	0	127	60		The MIDI note to automate 'Octave down'.
Retrigger	0	127	58		The MIDI note to automate 'Retrigger'.
Stop all	0	127	62		The MIDI note to automate 'Stop all'.
Bypass clock	0	127	71		The MIDI note to automate 'Bypass clock'.
Undo	0	127	72		The MIDI note to automate 'Undo'.
Redo	0	127	74		The MIDI note to automate 'Redo'.
Fade to clear	0	127	76		The MIDI note to automate 'Fade to clear'.

Saving/loading loops

Functions for saving and loading audio to/from the MicroSD card are available through the 'Looper' menu.

Save loops

This menu saves the contents of any non-empty loops to the MicroSD card as WAV files.

A new folder with the name 'saved <date> <time>' is created within the top level 'looper' folder (which is created if it doesn't already exist).

The WAV files are named 'loop1.wav', 'loop2.wav' etc. if the loop has no layers, or 'loop1_layer01.wav' etc. if there are layers.

Load loops

This menu loads WAV files from the MicroSD card. It expects to find them using a naming convention as used by the 'Save loops' menu.

The mono/stereo channel count and bit depth of the loops will be changed to match the loaded WAV files, but the algorithm will not be respecified if the number of loops doesn't match the number of saved loop files on the card – it will simply load as many as it can.

Macro Oscillator 2

"It's Plaits!"

File format guid: 'mac2'

Specifications: None

Description

This algorithm is an implementation of the open source [Plaits](61) module by Émilie Gillet.

This implementation should behave and sound identical to the original. Any demos or tutorials you may find online should apply just as well to this version. The user manual for the original module is [here](62). The documentation below assumes some familiarity with this.

This version of the disting NT firmware is based on v1.2 of the Plaits code.

Input normalization

The original Plaits module made extensive use of detecting which inputs had jacks plugged in and adjusting its behaviour accordingly. For example, if nothing is plugged into the Trigger input, the envelope is not used and the module constantly emits sound.

The disting NT has no way of knowing whether its sockets are connected or not, so this algorithm relies on the various 'input' settings being enabled or not. To take the same example, if the 'Trigger input' setting is set to 'None', the algorithm will generate constant sound; if the setting is something else, the algorithm assumes you're going to supply triggers on the input you choose.

Outputs

As on the original Plaits, there are two outputs, 'Main' and 'Aux'. If you select the same bus for both outputs they are mixed.

Plaits parameters

Name	Min	Max	Default	Unit	Description
Model	0	23	0		Chooses the synthesis model.
Coarse tune	-60	60	0	ST	Coarse tuning control.
Fine tune	-100	100	0	cents	Fine tuning control.

---

61 https://github.com/pichenettes/eurorack/tree/master/plaits

62 https://pichenettes.github.io/mutable-instruments-documentation/modules/plaits/manual/

Harmonics	0	127	64		Sets the harmonics control.
Timbre	0	127	64		Sets the timbre control.
Morph	0	127	64		Sets the morph control.
FM	-100	100	0	%	Sets the FM depth.
Timbre mod	-100	100	0	%	Sets the timbre modulation depth.
Morph mod	-100	100	0	%	Sets the morph modulation depth.
Low-pass gate	0	127	127		Sets the LPG colour (VCFA-VCA blend).
Time/decay	0	127	64		Sets the envelope decay time.

Inputs parameters

Name	Min	Max	Default	Unit	Description
Trigger input	0	28	0		Selects which bus is used as the trigger input.
Level input	0	28	0		Selects which bus is used as the level input.
CV input	0	28	0		Selects which bus is used as the pitch CV input.
FM input	0	28	0		Selects which bus is used as the FM input.
Harmonics input	0	28	0		Selects which bus is used as the harmonics input.
Timbre input	0	28	0		Selects which bus is used as the timbre input.
Morph input	0	28	0		Selects which bus is used as the morph input.

Outputs parameters

Name	Min	Max	Default	Unit	Description
Main output	0	28	13		The bus for the 'Main' output.
Aux output	0	28	0		The bus for the 'Aux' output.
Output	0	1	0		The standard Add/Replace mode selector as

mode					described above.
Main gain	-40	6	0	dB	The level of the 'Main' output signal.
Aux gain	-40	6	0	dB	The level of the 'Aux' output signal.

Control parameters

Name	Min	Max	Default	Unit	Description
MIDI mode	0	3	0		Controls the algorithm's response to MIDI. See below.
I2C mode	0	3	0		Controls the algorithm's response to I2C. See below.
MIDI channel	0	16	1		The MIDI channel to receive on.
I2C channel	0	255	1		The I2C channel to receive on.

MIDI and I2C modes

The MIDI mode and I2C mode parameters allow you to choose how MIDI and I2C will be used. Both have the same options:

Off	Notes will not be used.
Pitch	Note on events will set the pitch.
Trigger	Note on events will cause a trigger.
Pitch & trigger	Note on events will set the pitch and cause a trigger.

Note that in terms of the input normalization logic described above, activating MIDI or I2C for pitch and/or trigger is taken to be equivalent to connecting a CV input for that parameter.

MIDI Player

"Plays MIDI files from the MicroSD card"

File format guid: 'midp'

Specifications: None

Description

This algorithm plays standard MIDI files, outputting the MIDI messages, and converting the MIDI to CVs and gates. It will use an internal clock (using the file's tempo, if specified), or sync to analogue clocks or incoming MIDI clock.

See the MicroSD card section above for information on supported MIDI file formats, and how to arrange the files on the card.

Optionally, this algorithm will also process live MIDI.

MIDI file limitations

This algorithm will use the first 24 tracks of each MIDI file. Tracks beyond that will be ignored.

GUI

On the right side, the display shows the current transport position (in bars, beats, and sixteenths) and the current position within the file (which would be different if, say, the file had looped, or you select a new file to play without stopping the transport). Below that is the file's tempo (if it defines one) and the file's length in bars. At the bottom right is a simple progress bar, indicating the playback position within the file.

There are two lines of activity indicators. The upper one represents the tracks in the file; the lower one represents MIDI channels.

If the current parameter is one of the 'Track mode' parameters, further information about that track is shown: the track name (if defined in the file) and the most recent MIDI message played on that track.
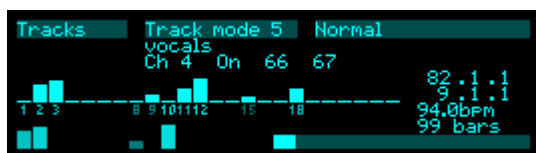
Transport parameters

Name	Min	Max	Default	Unit	Description

Folder					Chooses the folder of MIDI files on the card.
File					Chooses the MIDI file within the folder.
Play	0	1	0		Starts/stops playback using the internal timebase.
Start from bar	1	256	1		Sets the bar number within the file from which to start playback.
File change	0	2	0		Sets the behaviour when a new file is chosen. The options are: ● Wait until the end of the bar and then change to the new file. ● Wait until the end of the file and then change to the new file. ● Change immediately.
File restart	0	1	0		Sets how the file position changes when switching to a new file. The options are: ● The new file starts from the beginning. ● The new file starts from the current position in the old file (modulo the length of the new file).

Tracks parameters

Name	Min	Max	Default	Unit	Description
Track mode 1-24	0	1	0		Set the playback mode for the tracks. The options are: ● Normal playback. ● Track is muted.

MIDI/CV parameters

Name	Min	Max	Default	Unit	Description
1-4 MIDI channel	1	16	1-4		MIDI channel for the MIDI/CV converter.
1-4 Pitch output	0	28	0		Bus for the MIDI/CV converter pitch output.
1-4 Gate output	0	28	0		Bus for the MIDI/CV converter gate output.
Drum first note	0	127	36		The first (lowest) note for the drum converter.
Drum	0	16	0		MIDI channel for the drum converter.

Name	Min	Max	Default	Unit	Description
channel					
Trigger length	1	100	10	ms	Sets the length of output trigger pulses.
Convert live MIDI	0	1	0		If enabled, MIDI sent to the algorithm is converted by its MIDI/CV converters as if it originated in the MIDI file.

Clock parameters

Name	Min	Max	Default	Unit	Description
Clock multiplier	0	4	0		Sets the multiplier for incoming clock pulses. The options are: ● clock is interpreted as 24ppqn. ● clock is interpreted as 1/32 notes. ● clock is interpreted as 1/16 notes. ● clock is interpreted as 1/8 notes. ● clock is interpreted as 1/4 notes.
Clock input	0	28	0		The bus to use as the clock input.
Reset input	0	28	0		The bus to use as the reset input.
Reset mode	0	1	0		Sets the mode for the reset input. The options are: ● the input is a reset trigger. ● the input is a run/stop signal.
Follow MIDI clock	0	1	0		Sets whether the algorithm follows incoming MIDI clock.
Output MIDI clock	0	1	0		Set whether the algorithm outputs MIDI clock.

Outputs parameters

Name	Min	Max	Default	Unit	Description
Output to breakout	0	1	0		Enables MIDI output to the breakout.
Output to Select Bus	0	1	0		Enables MIDI output to the Select Bus.
Output to USB	0	1	0		Enables MIDI output to USB.

Output to internal	0	1	0		Enables internal MIDI output - that is, MIDI is sent to the other algorithms.

Mixer Mono

"A mono mixer"

File format guid: 'mix1'

Specifications:

- Channels, 1-12: The number of mixer channels.
- Sends, 0-4: The number of aux sends per channel.

Description

This algorithm is a mono mixer - it mixes mono inputs to a mono output. Up to four aux send busses are available, which can be switched to pre- or post-fade.

GUI

The display shows a level meter for each channel. The thin line at the top indicates 0dBFS; if a channel goes over this, a clip indicator will show (a small rectangle over the 0dBFS line).

The chevrons to the right of the level meters indicate the mixer gains.

Mute and Solo are indicated with a small 'M' and 'S' respectively. Muted level meters are grayed out.

Common parameters

Name	Min	Max	Default	Unit	Description
Output	1	28	13		The mixer output bus.
Duplicate output	0	28	0		An optional duplicate output. Convenient if you want to drive a stereo bus from a mono mix.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.
Output gain	-70.0	6.0	0.0	dB	The mixer output gain.

Per-send parameters

Name	Min	Max	Default	Unit	Description
Destination	1	28	13		The output bus for the send. Note that the send always uses 'Add' output mode.
Pre/post	0	1	1		Whether the send is Pre-fade or Post-fade.

Per-channel parameters

Name	Min	Max	Default	Unit	Description
Input	0	28	1		The channel input bus.
Gain	-70.0	6.0	-70.0	dB	The channel gain.
Mute	0	1	0		Mutes the channel.
Solo	0	1	0		Solos the channel.

Per-channel per-send parameters

Name	Min	Max	Default	Unit	Description
Send gain	-70.0	6.0	-70.0	dB	The send gain.

Mixer Stereo

"A stereo mixer"

File format guid: 'mix2'

Specifications:

- Channels, 1-12: The number of mixer channels.
- Sends, 0-4: The number of aux sends per channel.

Description

This algorithm is a stereo mixer - it mixes mono or stereo inputs to a stereo output. Up to four (mono) aux send busses are available, which can be switched to pre- or post-fade.

GUI

The display shows a level meter for each channel. The thin line at the top indicates 0dBFS; if a channel goes over this, a clip indicator will show (a small rectangle over the 0dBFS line).

The chevrons to the right of the level meters indicate the mixer gains.

Mute and Solo are indicated with a small 'M' and 'S' respectively. Muted level meters are grayed out.

Common parameters

Name	Min	Max	Default	Unit	Description
Left output	1	28	13		The left output bus.
Right output	1	28	14		The right output bus.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.
Output gain	-70.0	6.0	0.0	dB	The mixer output gain.

Per-send parameters

Name	Min	Max	Default	Unit	Description
Destination	1	28	13		The output bus for the send. Note that the send always uses 'Add' output mode.
Pre/post	0	1	1		Whether the send is Pre-fade or Post-fade.

Per-channel parameters

Name	Min	Max	Default	Unit	Description
Input left/mono	0	28	1		The channel left input bus.
Input right	0	28	1		The channel right input bus (if stereo).
Gain	-70.0	6.0	-70.0	dB	The channel gain.
Pan	-100	100	0	%	The channel stereo pan.
Mute	0	1	0		Mutes the channel.
Solo	0	1	0		Solos the channel.

Per-channel per-send parameters

Name	Min	Max	Default	Unit	Description
Send gain	-70.0	6.0	-70.0	dB	The send gain.

Noise gate

"A simple noise gate"

File format guid: 'nsgt'

Specifications:

- Channels, 1-12: The number of bus channels to process.

Description

This algorithm is a multi-channel noise gate. Each channel is fully independent - this being a multi-channel algorithm is simply a convenience so that if you, say, want a noise gate on all 12 module inputs, you don't need to add 12 copies of the algorithm to do so.

A sidechain input is available, to gate one signal with another.

The algorithm always works as an insert effect i.e. the output replaces the input.

GUI

The display shows a graph of the signal level (travelling from right to left). Superimposed on this are the threshold and hysteresis levels. To the right of the graph, the bar represents the gain reduction currently being applied.
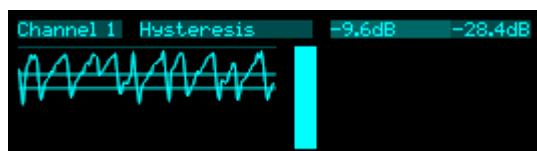
Common parameters

Name	Min	Max	Default	Unit	Description
Bypass	0	1	0		If on, the entire algorithm is bypassed.

Per-channel parameters

Name	Min	Max	Default	Unit	Description
Enable	0	1	0		Enables the channel.
Left/mono input	1	28	1		The left or mono input bus.
Right input	0	28	0		The right input bus, if stereo.

Sidechain input	0	28	0		The sidechain input bus.
Threshold	-70.0	0.0	-24.0	dB	The threshold level required to open the gate.
Hysteresis	-24.0	0.0	-3.0	dB	The level, relative to the threshold, that the input must drop below before the gate closes again.
Attack	0	1023	100		The attack time for the gate opening, from 0.2ms to 200ms with an exponential scale.
Hold	0	1023	100		The minimum time that the gate remains open, from 1ms to 1000ms with an exponential scale.
Release	0	1023	100		The release time for the gate closing, from 2ms to 2000ms with an exponential scale.
Lookahead	0.0	10.0	1.0	ms	The lookahead time. Increase this to prevent the gate from missing sharp attack transients. Note that the audio is delayed by the amount of the lookahead time i.e. this adds latency.
Gain reduction	-80	0	-80	dB	Sets the amount by which the signal is attenuated when the gate is closed. '-80' is actually treated as '-∞dB' i.e. fully attenuated.

Noise generator

"Generates various colours of noise"

File format guid: 'nois'

Specifications:

- Channels, 1-8: The number of output channels.

Description

This algorithm is a simple noise generator. Various standard "colours" of noise can be generated.

Globals parameters

Name	Min	Max	Default	Unit	Description
Gain	-40	6	0	dB	An overall gain to apply, in addition to the per-channel gain.

Per-channel parameters

Name	Min	Max	Default	Unit	Description
Output	0	28	13		The output bus.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.
Amplitude	0.00	10.00	10.00	V	The amplitude of the noise signal (before the gain is applied).
Gain	-40	6	0	dB	The output level.
Colour	0	4	0		The noise colour. The options are "Blended", "Violet", "White", "Pink", and "Red".
Blend	0	300	0		If "Blended" is selected, sets the blend between noise colours.

Notes

"A place to store some text"

File format guid: 'note'

Specifications: None
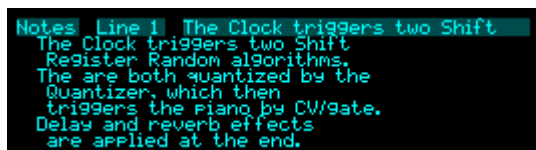
Description

This algorithm is simply a place to enter some text, perhaps an explanation of how the preset works, or your set list, or a reminder to buy milk.

It has absolutely no effect on the busses and consumes no CPU.

GUI

The display simply shows the lines of text all on one screen.

Oscilloscope

"Oscilloscope for viewing waveforms"

File format guid: 'oscs'

Specifications: None

Description

This algorithm implements a simple but useful 2-channel oscilloscope.

GUI

The display shows one or both waveforms according to the display mode.
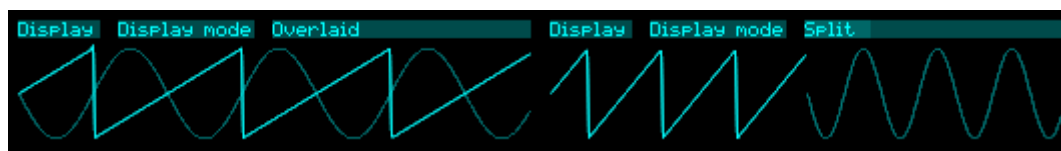
Inputs parameters

Name	Min	Max	Default	Unit	Description
Input 1	1	28	1		The bus for input 1.
Input 2	1	28	2		The bus for input 2.

Trigger parameters

Name	Min	Max	Default	Unit	Description
Trigger type	0	4	0		The type of trigger, to synchronise the display to the waveform. The options are: ● Free running. ● "1 rising" - trigger on the rising edge of input 1. ● "1 falling" - trigger on the falling edge of input 1. ● "2 rising" - trigger on the rising edge of input 2. ● "2 falling" - trigger on the falling edge of input 2.
Trigger voltage	-11.0	11.0	0.0	V	Set the trigger voltage - the voltage the input signal has to cross to trigger the oscilloscope.

Time range	0	13	7		Sets the time range corresponding to the full width of the display. The special value "Auto" sets the time range to the time between consecutive triggers.

Display parameters

Name	Min	Max	Default	Unit	Description
Display mode	0	4	2		Sets which waveforms are displayed and how. The options are "Overlaid", "Split", "Channel 1", "Channel 2", and "XY". In "XY" mode, inputs 1 & 2 are used as the coordinates to draw, instead of using time as the abscissa.
Draw mode	0	2	0		Sets how the waveform signals are drawn. The options are "Lines (antialiased)", "Lines", and "Points".
Vertical scale 1	0	3	0		Sets the vertical scale for channel 1. The options are 10V, 5V, 2V, and 1V.
Vertical scale 2	0	3	0		Sets the vertical scale for channel 2.

Pitch reference

"Generates a pitch reference tone"

File format guid: 'ptch'

Specifications: None

Description

This algorithm simply generates a sine wave tone at a particular pitch or frequency.

Note that if using a note to specify the pitch, it respects the global tuning setting.

GUI

The display shows the chosen pitch as a MIDI note name plus cents, as a fractional MIDI note number, and as a frequency in Hz.

Pitch parameters

Name	Min	Max	Default	Unit	Description
Pitch mode	0	1	0		Sets the mode, either "Pitch", or "Frequency".
Note	0	127	69		Sets the note pitch.
Frequency	27.0	880.0	440.0	Hz	Sets the frequency.

Output parameters

Name	Min	Max	Default	Unit	Description
Output	1	28	13		The output bus.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.
Amplitude	0.00	10.00	10.00	V	The amplitude of the signal (before the gain is applied).

Gain	-40	6	-40	dB	The level of the output signal.

Poly CV

"A polyphonic MIDI/CV converter"

File format guid: 'pycv'

Specifications:

● Voices: 1-14: The number of simultaneous voices.

Description

This algorithm takes the polysynth control logic from the other polysynths and uses it not to produce sound, but to output CVs and gates.

This makes it most obviously useful as a MIDI/CV converter, but it can also be used as a chord/arpeggio engine controlled by CV/gate inputs.

Please refer to the section "Common polysynth features" above.

Voice output allocation

The 'First output' parameter sets the lowest-numbered bus to use for outputs. For each voice, outputs are assigned in the order gate, pitch CV, velocity.

For example, if 'First output' is 'Output 1', 'Voices' is set to 2, gate and pitch outputs are enabled, and velocity outputs are disabled, then the two voices will use busses as follows:

• Voice 1: gate on Output 1, pitch CV on Output 2.

• Voice 2: gate on Output 3, pitch CV on Output 4.

GUI

The display shows the notes being played by the various voices.
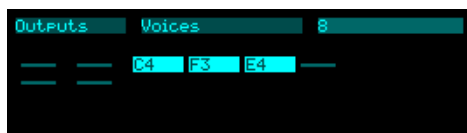
When adjusting the microtuning parameters, more information about the chosen tuning is displayed.
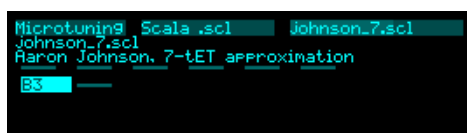
MIDI/I2C parameters

Name	Min	Max	Default	Unit	Description
MIDI channel	0	16	1		The MIDI channel to listen on.
MPE channels	1	16	1		Controls how the algorithm will respond to MPE. See above.
I2C channel	0	255	1		Sets the I2C channel.
Sustain mode	0	1	0		The standard polysynth sustain mode parameter. See above.
Sustain	0	1	0		Directly controls the sustain (like a MIDI sustain pedal).
Transpose	-60	60	0	ST	Coarse tuning control.
Fine tune	-100	100	0	cents	Fine tuning control.
Bend range	0	48	2		The MIDI pitch bend range.

Outputs parameters

Name	Min	Max	Default	Unit	Description
Voices	1	14	1		The number of voices to use (up to the maximum set by the specification).
First output	1	28	15		The first output bus to use.
Gate outputs	0	1	1		Whether to include gate outputs.
Pitch outputs	0	1	1		Whether to include pitch CV outputs.
Velocity outputs	0	1	0		Whether to include velocity outputs.
Gate mode	0	1	1		The standard Add/Replace mode selector as described above for gate outputs.
Pitch mode	0	1	1		The standard Add/Replace mode selector as described above for pitch CV outputs.
Velocity mode	0	1	1		The standard Add/Replace mode selector as described above for velocity outputs.

Microtuning parameters

The algorithm uses the standard polysynth microtuning parameters, as described above.

Chord/arp parameters

The algorithm uses the standard polysynth chord and arpeggiator parameters, as described above.

CV/gate parameters

The algorithm uses the standard polysynth CV/gate parameters, as described above.

Poly FM

"A polyphonic FM synthesizer"

File format guid: 'pyfm'

Specifications:

- Timbres, 1-4: The number of timbres.
- Voices: 1-24: The number of simultaneous voices.

Description

This algorithm is inspired by the original Poly FM algorithm on the disting EX.

It is a polyphonic, multitimbral, FM synthesizer.

It uses the six-operator FM engine from the open source Plaits[63] module by Émilie Gillet (exactly as used in the Macro Oscillator 2 algorithm, above). It loads Yamaha DX7 voice bank SysEx files from the MicroSD card.

Please refer to the section "Common polysynth features" above.

Timbres

The algorithm supports up to four different 'timbres', which here mainly means a different bank/voice, plus some other parameters that make up a particular sound.

The various timbres share the pool of (up to 24) simultaneous voices.

Voice vs Voice

Yamaha's original documentation made the unfortunate decision to name the choice of sound within a bank a 'voice', which conflicts with the more common term for one of the sound-generating elements of the hardware (as in the phrase "eight voice polyphonic" meaning eight notes at once can be sounded). We've tried to be consistent with Yamaha's usage here (as in "a bank contains 32 voices").

GUI

The display shows the notes being played by the various voices, each superimposed on a bar to indicate the note envelope.

---

63 https://github.com/pichenettes/eurorack/tree/master/plaits

When adjusting the microtuning parameters, more information about the chosen tuning is displayed.

Globals parameters

Name	Min	Max	Default	Unit	Description
Global gain	-40	6	0	dB	A global gain adjustment applied to all timbres (in addition to their individual gains).
Sustain mode	0	1	0		The standard polysynth sustain mode parameter. See above.

Microtuning parameters

The algorithm uses the standard polysynth microtuning parameters, as described above.

Per-timbre parameters

Name	Min	Max	Default	Unit	Description
Bank					Chooses the voice bank.
Voice	1	32	1		Chooses the voice within the bank.
Transpose	-60	60	0	ST	Coarse tuning control.
Fine tune	-100	100	0	cents	Fine tuning control.
Brightness	-100	100	0	%	An overall scaling on the modulator depths. Corresponds to the Plaits 'Timbre'.
Envelope scale	-100	100	0	%	An overall scaling on the envelope times. Corresponds to the Plaits 'Morph'.
Gain	-40	6	0	dB	Sets the output level.
Sustain	0	1	0		Directly controls the sustain (like a MIDI sustain pedal).

Chord/arp parameters

The algorithm uses the standard polysynth chord and arpeggiator parameters, as described above.

Per-timbre setup parameters

Name	Min	Max	Default	Unit	Description
Left/mono output	1	28	13		The left or mono output bus.
Right output	0	28	0		The right output bus.
MIDI channel	0	16	1		The MIDI channel to listen on.
MPE channels					Controls how the algorithm will respond to MPE. See above.
I2C channel					Sets the I2C channel.
Bend range					The MIDI pitch bend range.

CV/gate parameters

The algorithm uses the standard polysynth CV/gate parameters, as described above.

Poly Multisample

"A polyphonic sample player"

File format guid: 'pyms'

Specifications:

- Timbres, 1-4: The number of timbres.
- Voices: 1-16: The number of simultaneous voices.

Description

This algorithm is inspired by the original "SD Multisample" algorithm on the disting EX.

It is a polyphonic, multitimbral, sample playback instrument, playing WAV files from the MicroSD card. It can be played via CV/gate, MIDI, or I2C. It supports both velocity switches and round robins per sample.

Please refer to the section "Common polysynth features" above, and to the section on samples on the MicroSD card, also above.

Timbres

The algorithm supports up to four different 'timbres', which here mainly means a different sample folder, plus some other parameters that make up a particular sound.

The various timbres share the pool of (up to 16) simultaneous voices.

GUI

The display shows the notes being played by the various voices, each superimposed on a bar to indicate the note envelope. The folder and filename of the last sample played are also shown.

When adjusting the microtuning parameters, more information about the chosen tuning is displayed.

Globals parameters

Name	Min	Max	Default	Unit	Description
Global gain	-40	6	0	dB	A global gain adjustment applied to all timbres (in addition to their individual gains).
Sustain mode	0	1	0		The standard polysynth sustain mode parameter. See above.
Gate offset	0.0	10.0	0.2	ms	Offsets (delays) the gate inputs relative to the pitch inputs. This is useful to allow pitch CVs to settle before they are sampled on the rising gate, and also to cope with modules which output both a pitch and gate but change their gate first.
Round robin mode	0	3	0		The round-robin mode. See above.

Microtuning parameters

The algorithm uses the standard polysynth microtuning parameters, as described above.

Per-timbre parameters

Name	Min	Max	Default	Unit	Description
Folder	1		1		Sets the sample folder to use.
Gain	-40	24	0	dB	Sets the output level.
Pan	-100	100	0	%	Sets the stereo pan position.
Transpose	-60	60	0	ST	Adjusts the tuning in semitones.
Fine tune	-100	100	0	cents	Adjusts the tuning in cents.
Envelope	0	1	0		Enables an ADSR volume envelope. If the envelope is disabled, the sample simply plays once until the end.
Attack	0	127	0		Sets the envelope attack time (from 1ms to 15s, exponential response).
Decay	0	127	60		Sets the envelope decay time (from 20ms to 15s, exponential response).
Sustain	0	100	100	%	Sets the envelope sustain level.
Release	0	127	77		Sets the envelope release time (from 10ms to 30s, exponential response).

Velocity	0	100	100	%	Sets the amount by which the note velocity affects the note volume.
Sustain	0	1	0		Directly controls the sustain (like a MIDI sustain pedal).

Chord/arp parameters

The algorithm uses the standard polysynth chord and arpeggiator parameters, as described above.

Per-timbre setup parameters

Name	Min	Max	Default	Unit	Description
Left/mono output	1	28	13		The left or mono output bus.
Right output	0	28	0		The right output bus.
MIDI channel	0	16	1		The MIDI channel to listen on.
MPE channels	1	16	1		Controls how the algorithm will respond to MPE. See above.
I2C channel	0	255	1		Sets the I2C channel.
Bend range	0	48	2	ST	The MIDI pitch bend range.

CV/gate parameters

The algorithm uses the standard polysynth CV/gate parameters, as described above.

Poly Wavetable

"A polyphonic wavetable synthesizer"

File format guid: 'pywt'

Specifications:

- Voices: 1-24: The number of simultaneous voices.

Description

This algorithm is inspired by the original Poly Wavetable algorithm on the disting EX.

It is a complete 8 voice polyphonic synthesizer, using wavetable oscillators. Each voice has two envelopes, a filter and an LFO.

Please refer to the section "Common polysynth features" above, and to the section on wavetables on the MicroSD card, also above.

There are a number of in-depth videos on the disting EX version of this algorithm, which mostly apply equally well to this version. The YouTube playlist is [here](https://www.youtube.com/watch?v=6ytLfM9xyRY&list=PLlY5j4QDwxWunH7I7wvOGYGNsRvCMZ2Hx)[64].

GUI

The display shows the notes being played by the various voices, each superimposed on a bar to indicate the note envelope. The waveform within the wavetable of the most recently played note is also shown.

Wavetable parameters

Name	Min	Max	Default	Unit	Description
Wavetable	0		0		Chooses the wavetable from those installed on the MicroSD card.
Wave offset	-100	100	0		An offset for the wavetable position, added to that set from the wave input.
Wave spread	-100	100	0		An amount by which to spread out the per-voice wavetable positions.
Wave input	0	28	0		Which input bus to use to control the position in the

---

64 https://www.youtube.com/watch?v=6ytLfM9xyRY&list=PLlY5j4QDwxWunH7I7wvOGYGNsRvCMZ2Hx

					wavetable.

Envelope 1-2 parameters

Name	Min	Max	Default	Unit	Description
Attack	0	127	20		Envelope attack time. Range 1ms-15s.
Decay	0	127	60		Envelope decay time. Range 20ms-15s.
Sustain	0	127	80		Envelope sustain level.
Release	0	127	60		Envelope release time. Range 10ms-30s.
Attack shape	0	127	64		Envelope attack shape. '0' is highly exponential; '127' is almost linear.
Decay shape	0	127	64		Envelope decay & release shape.

Filter parameters

Name	Min	Max	Default	Unit	Description
Filter type	0	3	0		The filter type; 'Off', 'Lowpass', 'Bandpass' or 'Highpass'.
Filter freq	0	127	64		The filter frequency, specified as a MIDI note number.
Filter Q	0	100	50		The filter resonance.

LFO parameters

Name	Min	Max	Default	Unit	Description
LFO speed	-100	100	90		The LFO speed. Range 0.01Hz-10Hz.
LFO retrigger	0	2	0		Sets whether the LFOs are retriggered at note on. The options are 'Poly' (each voice's LFO triggers independently), 'Mono' (all LFOs are retriggered when the first note is played), or 'Off' (LFOs are free-running).
LFO spread	0	90	0	Degrees	Sets the phase to which LFOs are retriggered. The value, in degrees (360° per LFO cycle), is multiplied by the voice number to give the initial LFO phase. When retrigger is off, this sets the phase relationship between the free-running LFOs.

Modulation parameters

Name	Min	Max	Default	Unit	Description
Veloc -> volume	0	100	100	%	The amount by which the note velocity affects the note volume.
Veloc -> wave	-100	100	0	%	The amount by which the note velocity affects the wavetable position.
Veloc -> filter	-127	127	0		The amount by which the note velocity affects the filter frequency.
Pitch -> wave	-100	100	0	%	The amount by which the note pitch affects the wavetable position.
Pitch -> filter	-100	100	0	%	The amount by which the note pitch affects the filter frequency.
Env-1 -> wave	-100	100	0	%	The amount by which envelope 1 affects the wavetable position.
Env-1 -> filter	-127	127	0		The amount by which envelope 1 affects the filter frequency.
Env-2 -> wave	-100	100	0	%	The amount by which envelope 2 affects the wavetable position.
Env-2 -> filter	-127	127	0		The amount by which envelope 2 affects the filter frequency.
Env-2 -> pitch	-12.0	12.0	0.0	ST	The amount by which envelope 2 affects the note pitch.
LFO -> wave	-100	100	0	%	The amount by which the LFO affects the wavetable position.
LFO -> filter	-127	127	0		The amount by which the LFO affects the filter frequency.
LFO -> pitch	-12.0	12.0	0.0	ST	The amount by which the LFO affects the note pitch.

Girth parameters

Name	Min	Max	Default	Unit	Description
Unison	1	8	1		The number of voices to play simultaneously for each note triggered.
Unison detune	0	100	10	cents	The detune amount when Unison is active.

Output spread	-100	100	0	%	The amount of output spread.
Spread mode	0	2	0		The output spread mode. See below.

Microtuning parameters

The algorithm uses the standard polysynth microtuning parameters, as described above.

Chord/arp parameters

The algorithm uses the standard polysynth chord and arpeggiator parameters, as described above.

Setup parameters

Name	Min	Max	Default	Unit	Description
Gain	-40	24	0	dB	Applies an overall output gain.
Left output	1	28	13		The left output bus.
Right output	0	28	14		The right output bus.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.
MIDI channel	0	16	1		The MIDI channel to listen on.
MPE channels	1	16	1		Controls how the algorithm will respond to MPE. See above.
I2C channel	0	255	1		Sets the I2C channel.
Pitchbend input	0	28	0		The MIDI pitch bend range.
Max voices	1	24	24		Sets the maximum number of simultaneous voices.
Sustain mode	0	1	0		The standard polysynth sustain mode parameter. See above.
Sustain	0	1	0		Directly controls the sustain (like a MIDI sustain pedal).
Bend range	0	48	2		The MIDI pitch bend range.

CV/gate parameters

The algorithm uses the standard polysynth CV/gate parameters, as described above.

Spread modes

The available values for the 'Spread mode' parameter are as follows:

Spread by voice	Voices are spread across the stereo field from left to right.
Spread by voice 2	Voices are spread across the stereo field in an alternating left/right manner, by a small amount for low numbered voices, increasing for the remaining voices.
Spread by pitch	Voices are spread across the stereo field according to their pitch, with note 48 at the centre.

Quantizer

"A CV quantizer"

File format guid: 'quan'

Specifications:

- Channels, 1-12: The number of bus channels to process.

Description

This algorithm is a multi-channel CV quantizer, loosely based on the Quad Quantizer algorithm on the disting EX. It supports microtuning (see above), but this is not a requirement.

The quantization parameters are common to all channels; the individual channels' parameters are solely related to routing.

Quantization signal flow

There are a number of steps between the input CV and the output.

1. The input CV is sampled. If the quantizer has a gate input selected, the CV is sampled when the gate goes high. Otherwise, it is sampled when it has changed sufficiently to select a new note.
2. The input transpose is applied. This simply adjusts the CV in multiples of 12-TET semitones i.e. one twelfth of a Volt.
3. The CV is quantized into the 'lookup' scale, yielding a note number.
4. The shift parameter is applied.
5. The note number selects a CV in the 'output' scale.
6. The output transpose is applied.
7. The output calibration is applied, if in use.

Steps 3 & 5 warrant further explanation, and depend on the selected quantization mode.

In "Nearest" and "Warped" modes, the lookup and output scales are the same. The CV is simply quantized to the nearest note in the scale.

In "Mapped" mode, the lookup scale is 12-TET. The input CV is effectively selecting a standard MIDI note number, with 12 equally spaced semitones per octave. This note number is then used to choose an output CV in the chosen scale, according to its keyboard map.

By way of an example, consider the case where you load a Scala file defining 31-EDO i.e. there are 31 notes per octave. Say you apply a CV from an LFO, ramping up over a couple of Volts.

- In Nearest and Warped modes, you'll get all 31 notes per octave, and a 1V change in the input will give you a one octave rise in pitch.
- In Mapped mode, the output will depend on the chosen keyboard map (kbm) file. You might

have a keyboard map which chooses one of the 31-EDO pitches per semitone, but still only defines 12 pitches per octave. In this case you'll get your 12 microtonally tuned pitches per octave, and a 1V CV change will give you a one octave pitch change. Or, you might have a keyboard map which lays out all 31 pitches over 31 'semitones', as you might if you wanted to play in 31-EDO from a standard MIDI keyboard and be able to access all 31 pitches. In this case, the LFO in our example would have to rise over 2.5V to cover an octave, passing though all 31 pitches on the way.

Warped mode differs from Nearest mode in that the voltages in the lookup scale are redistributed to be evenly spaced. This is often useful in scenarios where you want to create note patterns from LFOs or similar. Consider the case where you're quantizing to a major scale, and the input is a simple ramp LFO. In Nearest mode (which is how most quantizers work), the differing gaps between the notes (e.g. a tone from D to E but a semitone from E to F) will result in an uneven rhythm as the notes of the scale are passed by the LFO voltage. Warped mode neatly solves this, and gives you a regular rhythm even though the notes are unevenly spaced. (Try it – it's much easier to hear this than it is to imagine it.)

GUI

The display shows detailed information about the current quantizer channel.

The top line shows the current input, as a voltage, and as a standard 12-TET MIDI note – a fractional note number, and again as a note name and cents. The second line shows the sampled voltage. The third and fourth lines show the quantized note numbers, first the lookup and then the output. The fifth line shows the output, again as a voltage and as a note number/name/cents. If a Scala file is in use, the tuning of the current note is shown, either in cents or as a ratio. In the examples above, a scale is used which starts on C and which uses A=440 as a reference. The one on the left uses an equal temperament scale, so the quantized note A is 900 cents from the root. The one on the right uses a just intonation, where A, the sixth, uses a ratio of 5/3. Note that in both cases, the output pitch is exactly A5+0, since the scale uses A as the tuning reference. Contrast these with the below:

Now the quantized note is G. The equal temperament scale on the left shows G as 700 cents, and gives G5+0. The just scale on the right shows G as 3/2, and a pitch of G5+17.

Quantizer parameters

Name	Min	Max	Default	Unit	Description

Quantize mode	0	2	0		Sets the quantization mode: 'Nearest', 'Mapped', or 'Warped'.
Input transpose	-48	48	0	ST	Sets the input transposition, in 12-TET semitones.
Shift	-48	48	0		Sets the in-scale shift.
Key	-12	12	0		Sets the key (C, D♭, E, etc.).
Mode	1	12	1		Sets the mode, that is, the rotation of the notes within the scale. If the selected scale is 'Major', the mode is displayed by one of the familiar names Ionian, Dorian, Phrygian, Lydian, Mixolydian, Aeolian, or Locrian.
Scale	0	10	2		The scale to quantize into. Defaults to Chromatic. See below.
Output transpose	-48	48	0	ST	Sets the output transposition, in 12-TET semitones.
Output gate mode	0	2	0		Sets the behaviour of the gate/trigger outputs. The options are: 'Triggers', 'Inv Triggers' (inverted triggers, always high except for a low pulse when the trigger is fired, useful for driving an envelope generator which is generally in sustain but which needs to be retriggered when the chord changes), and 'Gates' (in which case the output is gated by the corresponding gate input).
Gate offset	0.0	10.0	2.0	ms	Offsets (delays) the gate inputs relative to the pitch inputs. This is useful to allow pitch CVs to settle before they are sampled on the rising gate, and also to cope with modules which output both a pitch and gate but change their gate first.
MIDI channel	0	16	1		The MIDI channel to receive on.

Microtuning parameters

The algorithm uses the standard microtuning parameters, as described above.

Per-channel parameters

Name	Min	Max	Default	Unit	Description
CV input	0	28	1		The pitch CV input bus.
Gate input	0	28	0		The (optional) gate input bus.

CV output	0	28	15		The pitch CV output bus. Always uses 'Replace' output mode.
Gate output	0	28	0		The gate output bus, according to the 'Output gate mode' parameter. Always uses 'Replace' output mode.
Change output	0	28	0		The 'change trigger' output bus. Fires a trigger pulse when the quantized note changes. Always uses 'Replace' output mode.

Key and scale

The Key and Scale parameters can be used to constrain the quantized notes to a particular scale.

This works internally by removing the unwanted notes from the keyboard map, so if you're already using a .kbm file with unmapped notes you may get unexpected results.

The available scales are the common selection (see above), plus two 'scales' that use live MIDI input to define the notes to quantize to. When these scales are used, playing MIDI notes adds notes to the selection, so long as any key is held. When all keys are released, the next note played clears the selection and starts a new one.

'MIDI (any octave)' allows the quantizer to choose notes as held, but in any octave. For example, if you hold any C and any G on your keyboard, the quantizer is free to choose any C or G.

'MIDI (exact)' allows the quantizer to choose only the actual notes played. For example, if you hold C4 and G5, the quantizer may only choose C4 or G5, not any other C or G.

When one of the MIDI scales is used, the Key parameter becomes a simple transpose. For example, if you hold C and G, and set the Key to 'D' (parameter value 2), the quantizer may choose D or A.

Resonator

"It's Rings!"

File format guid: 'resn'

Specifications: None

Description

This algorithm is an implementation of the open source [Rings](#)[65] module by Émilie Gillet.

It can be played via MIDI or I2C, or simply via CV/gate as the original.

Any demos or tutorials you may find online should apply just as well to this implementation. The user manual for the original module is [here](#)[66]. The documentation below assumes some familiarity with this.

Input normalization

The original Rings module made extensive use of detecting which inputs had jacks plugged in and adjusting its behaviour accordingly. For example, if nothing is plugged into the Strum input, the module generates strums internally from pitch changes or audio transients.

The disting NT has no way of knowing whether its sockets are connected or not, so this algorithm relies on the various 'input' parameters being enabled or not. To take the same example, if the 'Strum input' parameter is set to 'None', the algorithm will generate its own strums; if the parameter is something else, the algorithm assumes you're going to supply strums on the input you choose.

Rings parameters

Name	Min	Max	Default	Unit	Description
Mode	0	6	0		Selects the resonator mode.
Synth effect	0	5	0		If the mode is 'Synth', selects the audio effect applied to the synth output.
Polyphony	1	4	1		Sets the number of simultaneous voices.
Coarse tune	-36	24	0	ST	Provides a coarse tuning control.
Fine tune	-100	100	0	cents	Provides a fine tuning control.
Structure	0	127	64		Controls the resonator 'structure'.

---

65 https://github.com/pichenettes/eurorack/tree/master/rings

66 https://pichenettes.github.io/mutable-instruments-documentation/modules/rings/manual/

Brightness	0	127	64		Controls the resonator 'brightness'.
Damping	0	127	64		Controls the resonator 'damping'.
Position	0	127	64		Controls the resonator 'position'.
Chord	0	10	0		Chooses the chord to use for resonator modes that use one. In the original, set from the Structure knob.
Noise gate	0	1	1		Enables a noise gate on the audio input. Always enabled in the original Rings.
Input gain	-40	12	0	dB	A gain applied to the input audio, before it hits the resonator.

Inputs parameters

Name	Min	Max	Default	Unit	Description
Strum input	0	28	0		Chooses which input bus to use for 'strum'.
V/Oct input	0	28	0		Chooses which input bus to use for the V/octave pitch CV.
Audio input	0	28	0		Chooses which input bus to use for the audio input.
Brightness input	0	28	0		Chooses which input bus to use for 'brightness'.
Frequency input	0	28	0		Chooses which input bus to use for 'frequency'.
Damping input	0	28	0		Chooses which input bus to use for 'damping'.
Structure input	0	28	0		Chooses which input bus to use for 'structure'.
Position input	0	28	0		Chooses which input bus to use for 'position'.

Outputs parameters

Name	Min	Max	Default	Unit	Description
Odd output	0	28	13		The bus to use for the Odd output.
Odd output mode	0	1	0		The Add/Replace mode for the Odd output.

Even output	0	28	13		The bus to use for the Even output.
Even output mode	0	1	0		The Add/Replace mode for the Even output.
Output gain	-40	12	0	dB	The output level (of both the Odd and Even outputs).
Dry gain	-40	12	-40	dB	The level of the input audio mixed into the output(s).

Control parameters

Name	Min	Max	Default	Unit	Description
MIDI mode	0	3	0		Controls the algorithm's response to MIDI. See below.
I2C mode	0	3	0		Controls the algorithm's response to I2C. See below.
MIDI channel	0	16	1		The MIDI channel to receive on.
I2C channel	0	255	1		The I2C channel to receive on.

MIDI and I2C modes

The MIDI mode and I2C mode parameters allow you to choose how MIDI and I2C will be used. Both have the same options:

Off	Notes will not be used.
Pitch	Note on events will set the pitch.
Strum	Note on events will cause a strum.
Pitch & strum	Note on events will set the pitch and cause a strum.

Note that in terms of the input normalization logic described above, activating MIDI or I2C for pitch and/or strum is taken to be equivalent to connecting a CV input for that parameter. E.g. setting the MIDI mode to 'Strum' will stop the algorithm generating its own strums internally.

Reverb

"A general purpose reverb effect"

File format guid: 'revb'

Specifications: None

Description

This algorithm offers a classic algorithmic reverb effect. It does not seek to emulate any particular hardware, or for that matter, any particular physical reverberant space.

Reverb parameters

Name	Min	Max	Default	Unit	Description
Mix	0	100	100	%	Sets the wet/dry mix.
Time	400	30000	1700	ms	Sets the reverb time.
Model	0	3	1		Chooses the reverb model.
Size	1	100	100	%	Sets the size of the reverb space – mainly affects the times of the early reflections.
High damp	0	100	60		Sets the amount of high frequency damping in the reverb tail.
Modulation speed	0.01	5.00	2.50	Hz	Sets the speed of reverb modulation.
Modulation depth	0	100	25		Sets the depth of reverb modulation.
Early gain	-40	6	-12	dB	Sets the output level of the early reflections.
Diffuse gain	-40	6	-6	dB	Sets the output level of the diffuse reflections.

Routing parameters

Name	Min	Max	Default	Unit	Description
Left input	1	28	1		The left audio input bus.
Right input	1	28	2		The right audio input bus.

Left output	1	28	13		The left audio output bus.
Right output	1	28	14		The right audio output bus.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.

Reverb (Clouds)

"The reverb from Clouds"

File format guid: 'revc'

Specifications: None

Description

This algorithm implements the reverb and diffuser parts of the [Clouds](https://github.com/pichenettes/eurorack/tree/master/clouds)[67] module by Émilie Gillet.

The diffuser is a simple network of all-pass filters which 'smears' the sound slightly. It is effective at turning a mono source into something that sounds stereo.

Effects parameters

Name	Min	Max	Default	Unit	Description
Diffuser mix	0	100	0	%	The diffuser wet/dry mix.
Reverb mix	0	100	100	%	The reverb wet/dry mix.
Reverb time	0	100	0	%	The reverb time.
Reverb diffusion	0	100	70	%	The reverb diffusion. (In Clouds this is fixed at 70%.)
Reverb low pass	0	100	50	%	The cut-off frequency of a low-pass filter within the reverb.
Reverb input gain	-40	0	-14	dB	The input gain to the reveberator.
Sample rate	0	1	0		Allows you to run the algorithm at the disting NT's own sample rate ('Native') or the sample rate of a real Clouds module ('Authentic (32kHz)').

Routing parameters

Name	Min	Max	Default	Unit	Description
Left input	1	28	1		The left audio input bus.
Right input	1	28	2		The right audio input bus.

---

67 https://github.com/pichenettes/eurorack/tree/master/clouds

Left output	1	28	13		The left audio output bus.
Right output	1	28	14		The right audio output bus.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.

Sample and Hold

"Simple sample (or track) and hold"

File format guid: 'saho'

Specifications:

- Channels, 1-8: The number of bus channels to process.

Description

This algorithm is a simple sample/track and hold utility. A common gate/trigger input controls the sampling/tracking of a number of signal channels.

There are two modes of operation:

- 'Sample and hold' - the output is a sample of the input, taken when the gate/trigger input goes high.
- 'Track and hold' - the output follows the input while the gate is high, and freezes when the gate goes low.

Common parameters

Name	Min	Max	Default	Unit	Description
Gate input	1	28	1		The input bus to use for the gate/trigger.
Gate offset	0.0	10.0	2.0	ms	Offsets (delays) the gate input relative to the signal inputs. This is useful to allow signal CVs to settle before they are sampled on the rising gate, and also to cope with modules which output both a pitch and gate but change their gate first.

Per-channel parameters

Name	Min	Max	Default	Unit	Description
Input	1	28			The signal input bus to sample/track.
Output	0	28	0		The output bus.
Output mode	0	1	1		The standard Add/Replace mode selector as described above.
Mode	0	1	0		Chooses between 'Sample and hold' and 'Track and hold'.

Sample player

"A simple sample player"

File format guid: 'samp'

Specifications:

- Triggers, 1-8: The number of individual sample triggers.

Description

This algorithm plays samples from the MicroSD card. It is loosely based on the "SD 6 Triggers" algorithm on the disting EX.

Whereas the "Poly Multisample" algorithm is aimed more at playing pitched instruments, this algorithm is mainly designed for playing simple one-shot samples e.g. drums.

It supports both velocity switches and round robins per sample.

The trigger inputs are velocity sensitive – the voltage of the gate signal is used like the velocity of a MIDI note. 5V corresponds to maximum velocity.

GUI

The display shows the MIDI note names of the samples being played by the various triggers, superimposed on a bar indicating the trigger's envelope. The folder and filename of the most recently triggered sample is also shown.

Globals parameters

Name	Min	Max	Default	Unit	Description
Gain	-40	24	0	dB	An overall gain to apply in addition to the per-trigger gain.
Round robin mode	0	3	0		The round-robin mode. See above.

Per-trigger parameters

Name	Min	Max	Default	Unit	Description

Folder	1				Sets the folder from which to choose a sample.
Sample	1				Sets the sample within the folder.
Transpose	-60	60	0	ST	Sets the sample tuning in semitones.
Fine tune	-100	100	0	cents	Sets the sample fine tuning.
Gain	-40	24	0	dB	The output level.
Pan	-100	100	0	%	The stereo pan position.
Vel(ocity) depth	0	100	100	%	Sets the amount by which the velocity affects the playback level.
Choke group	0	8	0		The voice's "choke group". When a voice in a choke group is triggered, it ends the playback of any other voices in the same choke group.
Play	0	1	0		Manually triggers sample playback.
Velocity	1	127	127		The velocity to use when 'Play' triggers playback.
Loop	0	2	0		Sets whether the sample will loop. 'From WAV file' will loop if the sample has loop points defined in the file - see above - and not otherwise. 'Off' and 'On' force the sample to loop or not.

Per-trigger setup parameters

Name	Min	Max	Default	Unit	Description
Left output	1	28	13		The left audio output bus.
Right output	0	28	14		The right audio output bus.
MIDI channel	0	16	1		The MIDI channel to respond to.
MIDI note	-1	127	48		The MIDI note to respond to, or "Any".
Trigger input	0	28	0		The input bus to use as the gate/trigger.

Per-trigger envelope parameters

Name	Min	Max	Default	Unit	Description
Envelope	0	1	0		Enables an ADSR volume envelope. If the envelope is disabled, the sample simply plays once

					until the end.
Attack	0	127	0		Sets the envelope attack time.
Decay	0	127	60		Sets the envelope decay time.
Sustain	0	100	100	%	Sets the envelope sustain level.
Release	0	127	77		Sets the envelope release time.

Saturation

"Soft-clipping saturation"

File format guid: 'satu'

Specifications:

- Channels, 1-8: The number of bus channels to process.

Description

This algorithm implements a simple soft-saturation effect. It can be used simply for creative tone-shaping, but it is particularly useful for avoiding harsh digital clipping when signals get loud. (It is the same processing that is built into many of the disting EX algorithms as a final "output bus" limiter.)

Globals parameters

Name	Min	Max	Default	Unit	Description
Bypass	0	1	0		If enabled, bypasses the entire algorithm.
Pre gain	-40	24	0	dB	A gain to apply before the saturation.
Post gain	-40	6	0	dB	A gain to apply after the saturation.

Per-channel parameters

Name	Min	Max	Default	Unit	Description
Input	0	28	13		The input (and output) bus.
Enable	0	1	0		Enables saturation on this channel.

Shift Register Random

"Generates random CVs"

File format guid: 'srra'

Specifications: None

Description

This algorithm generates random CVs via the popular rotating shift register method, often known simply as a "Turing Machine" after [this module](https://www.musicthing.co.uk/Turing-Machine/)[68].

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.

On each clock, 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.

This algorithm can output either CVs, or triggers, or both. You may like to feed the CVs into the Quantizer algorithm to generate random melodies.

GUI

The display shows a graphical representation of the shift register. If the unrandomness is ±100%, it also shows "<LOCKED>".

Register parameters

Name	Min	Max	Default	Unit	Description
Unrandomness	-100	100	0	%	Sets the likelihood of the pattern not changing on each clock. At 0%, there is a 50:50 chance of a bit flip, which is the most random setting. At 100%, there will never be a fit flip, so the pattern is locked. At -100% there will always be a bit flip, which has the effect of locking the pattern with a repeat count of twice the length.
Direction	0	1	0		Sets whether the pattern moves forwards or in reverse when clocked.

---

68 https://www.musicthing.co.uk/Turing-Machine/

Length	1	32	8		Sets the length of the shift register, and so sets the number of steps in the sequence.

CV parameters

Name	Min	Max	Default	Unit	Description
Scale	-20.0	20.0	10.0	V	Scales the output CV.
Offset	-10.0	10.0	0.0	V	Offsets the output CV.

Trigger parameters

Name	Min	Max	Default	Unit	Description
Cause	0	4	0		Chooses what will cause a trigger to be generated: ● The new bit value is high. ● The new bit value is low. ● The bit changes. ● The bit goes from low to high. ● The bit goes from high to low.
Type	0	1	0		Chooses whether the output trigger will be of a fixed length, or calculated as a percentage of the clock length.
Length (ms)	1	100	10	ms	The trigger length, if fixed length is chosen.
Length (%)	1	100	50	%	The trigger length, if percentage of clock is chosen.

Routing parameters

Name	Min	Max	Default	Unit	Description
Clock input	0	28	1		Chooses the clock input bus.
Modify input	0	28	0		Chooses the "modify" input bus. If this gate input is high, the pattern will always be modified on a clock, even if the pattern is locked.
CV output	0	28	15		The output bus for the CV.
Trigger output	0	28	0		The output bus for the trigger.
CV output	0	1	0		The Add/Replace mode for the CV bus.

mode					
Trigger output mode	0	1	0		The Add/Replace mode for the trigger bus.

Slew rate limiter

"Smooths CVs and creates glissandos"

File format guid: 'slew'

Specifications:

- Channels, 1-8: The number of bus channels to process.

Description

This algorithm is a simple slew rate limiter, offering both logarithmic and linear slew.

You can choose to use a single slew rate for both rising and falling signals, or specify them separately. You can also choose to use the same rates for all the busses, or specify them individually.

Common parameters

Name	Min	Max	Default	Unit	Description
Bypass	0	1	0		If enabled, bypasses the entire algorithm.
Up/shared slew	0	1000	0		The slew time for both, or only rising, signals.
Down slew	0	1000	0		The slew time for falling signals.

Per-channel parameters

Name	Min	Max	Default	Unit	Description
Type	0	1	0		Chooses Logarithmic or Linear slew rate limiting.
Control	0	3	0		Chooses which controls will set the slew times. • 'Single' - use 'Up/shared slew' for both slew times. • 'Dual' - use separate slew times for rising and falling signals. • 'Common single' and 'Common dual' - as above but using the shared times from the Common parameters.
Up/shared slew	0	1000	0		The slew time for both, or only rising, signals.
Down slew	0	1000	0		The slew time for falling signals.
Input	1	28	1		The input bus.

Output	0	28	0		The output bus. If set to 'None', the input bus is used as output, and the mode is always 'Replace'.
Output mode	0	1	1		The standard Add/Replace mode selector as described above.

Spectral Freeze

"An audio 'freeze' effect"

File format guid: 'spfz'

Specifications:

- Voices, 1-8: The number of simultaneous voices.
- History, 4-2048: The length of the history stored for each voice.
- FFT size, 8-12: The size of the FFT used internally.

Description

This algorithm is an audio 'freeze' effect – it captures sound at a moment in time and sustains it indefinitely. The 'spectral' part of the name indicates that it works in the frequency domain – unlike simple loopers or granular effects, which work in the time domain, this algorithm captures the spectrum of a sound (using an FFT) and uses that to resynthesize audio[69].

As well as completely freezing sound, the algorithm can actually play it back at any desired speed (i.e. it does time stretching) and pitch.

There are a number (depending on the specification) of different 'voices' i.e. you can freeze different moments of audio and play them simultaneously.

The algorithm is based on the disting EX algorithm of the same name. There is an in-depth video for that algorithm [here](#)[70].

GUI

The display shows information about each of the voices.
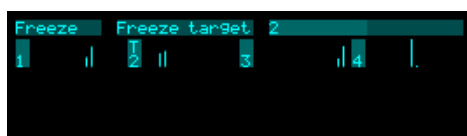
The vertical bar on the left shows the envelope level. The small vertical lines visualise the segment of the audio that is being used. The 'T' above to the voice number indicates the 'Freeze target'; if there is no 'T', the target is set to 'Auto'.

Movement

The simplest spectral freeze would take just one FFT 'frame' and use that to continuously synthesize audio. This generates a completely static sound, which, while sometimes a useful effect, doesn't give you the sort of textural sustain that is often wanted. For this reason, this algorithm offers means of

---

69 For the technically minded, it uses the phase vocoder technique.

70 https://www.youtube.com/watch?v=8HFm0zzLHRM

moving the synthesis source around in a small history window previous to the moment of freezing.

The 'Movement', 'Rate', and 'Depth' parameters allow you to apply an amount of motion automatically.

Depth sets the amount of time over which motion can occur. It is specified in terms of FFT frames, though an equivalent time in milliseconds is also shown. If this is set to 1, motion is effectively disabled.

Rate sets the speed of motion. If set to 0, motion is stopped. If set to 1000, the audio is played at its natural speed. In between those two extremes, playback is slowed down by a factor which is shown in the display.

Movement sets the direction of motion. 'Forwards' plays the audio forwards, in a loop. Note though that since this is a resynthesis algorithm, not a looper, the jump from the end of the loop to the start is spectrally interpolated, not simply crossfaded. 'Backwards' is similar, except that it plays the audio in reverse. 'Alternating' plays the audio alternately forwards and backwards. 'Random walk' moves randomly forwards and backwards – the playback position is continuous, but randomly changes direction. 'Random skip' randomly jumps around within the audio.

Finally, the 'Offset' parameter gives you direct control over the playback position within the buffer. Note that the motion set by Movement, Rate, and Depth is added to the Offset position – if you only want to have Offset control the playback, set Depth to 1 and/or Rate to 0.

Pitch shifting

Each voice can have a pitch shift specified. It should be noted that pitch shifting is not the primary focus of this algorithm, and the results are not the last word in quality pitch shifting. Nevertheless, it's provided as a potentially useful option.

There are coarse and fine shift controls which apply globally to all of the voices, and individual controls for each voice. The global and per-voice shifts are simply added.

Etherization

The 'Etherization' parameter progressively drops out components of the sound which are judged to be 'transient'. The result tends towards a small number of pure tones which represent the strongest harmonics of the material captured.

Freeze parameters

Name	Min	Max	Default	Unit	Description
Freeze toggle	0	1	0		Activates a new freeze when the parameter changes from '0' to '1'.

Freeze gate	0	1	0		Activates a new freeze when the parameter changes from '0' to '1', and releases the freeze when the parameter changes from '1' to '0'.
Freeze target	0		2		Chooses the voice for the next freeze, or '0' for 'Auto'.
Unfreeze all	0	1	0		When this parameter changes to '1', all active freezes are released.

Movement parameters

Name	Min	Max	Default	Unit	Description
Movement	0	4	2		Chooses the type of movement. The options are 'Forwards', 'Backwards', 'Alternating', 'Random walk', and 'Random skip'.
Rate	0	1000	340		Sets the rate of movement. '0' is stopped; '1000' is 'normal speed' (as recorded). In between, the scale is highly non-linear to give more control at slower speeds.
Depth	1		2		Sets the maximum amount of movement.
Offset	0		0		Offsets the freeze position back in time.

Changes parameters

Name	Min	Max	Default	Unit	Description
Etherization	0.0	100.0	0.0	%	Drops out transient elements of the sound.
Live process	0	1	0		When this parameter changes to '1', a new voice is started which uses live audio input instead of actually freezing. This is provided mainly as a means to use the etherization effect on live audio.
Attack	0	127	10		Sets the envelope attack time.
Release	0	127	10		Sets the envelope release time.
Coarse shift	-48	24	0	ST	Sets the coarse pitch shift amount for all voices. Added to the individual voice shifts.
Fine shift	-100	100	0	cents	Sets the fine pitch shift amount for all voices. Added to the individual voice shifts.

Per-voice parameters

Name	Min	Max	Default	Unit	Description
Freeze	0	1	0		Activates a new freeze on the corresponding voice when the parameter changes from '0' to '1', and releases the freeze when the parameter changes from '1' to '0'.
Coarse shift	-48	24	0	ST	Sets the coarse pitch shift amount.
Fine shift	-100	100	0	cents	Sets the fine pitch shift amount.
Output	1	28	13		Sets the output bus for the voice.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.

Routing parameters

Name	Min	Max	Default	Unit	Description
Input	1	28	1		Sets the input audio bus.

Step sequencer

"A simple note sequencer"

File format guid: 'spsq'

Specifications: None

Description

This algorithm is a 16-step note sequencer, based on the sequencers in the Expert Sleepers [FH-2](https://expert-sleepers.co.uk/fh2.html)[71] module. It can output CVs and/or MIDI.

Each step has pitch, velocity, and a general-purpose "modulation" value. It also has a "Division", which can either be a repeat count (the step is repeated a number of times on subsequent clocks) or a ratchet count (the step fires a number of gates within the duration of one clock). Each substep (either repeat or ratchet) can be on or off – this is set by the "Pattern" parameter. Furthermore each step or substep can be a tie – this is set by the "Ties" parameter.

A tie means a note which continues until just after the beginning of the next (sub)step. When outputting MIDI, this means the note off will follow the next note on i.e. the notes are played legato. When outputting CV, a tie means that the gate will not go low between the notes, and that the pitch CV will glide (if the glide time is non-zero).

The algorithm has a function for randomising the pitches, rhythm, or both.

Remember that every pitch etc. is a parameter and therefore mappable. Something as simple as an LFO modulating a sequence step's pitch can produce fascinating evolving patterns. If you need to constrain the sequencer to a scale/key, follow it with a Quantizer.

GUI

The display shows an overview of the 16 steps.

The step pitches are displayed across the top of the screen. The current step being played is indicated by a highlight on the pitch value. The start and end steps are indicated by solid bars between the pitches.

The step whose parameter is being edited is shown highlighted. The steps' divisions, patterns, and ties are shown graphically. If the current parameter is a pitch, velocity, or modulation, these values are shown via the thin lines to the left of the steps.

---

71 https://expert-sleepers.co.uk/fh2.html

Divisions (repeats or ratchets) are shown as stacks of blocks:

Ties are indicated by longer bars:

Sequencer parameters

Name	Min	Max	Default	Unit	Description
Start	1	16	1		The first step to play. A reset will jump to this step.
End	1	16	16		The last step to play.
Direction	0	6	0		The sequencer direction. See below.
Permutation	0	3	0		The sequencer permutation. See below.
Gate type	0	1	0		Sets the gate type: "% of clock" or "Trigger".
Gate length	1	99	50	%	Sets the length of the gate output if the type is "% of clock".
Trigger length	1	100	10	ms	Sets the length of the gate output if the type is "Trigger".
Glide	0	1000	100	ms	Sets the glide time for tied notes.

Step parameters

Name	Min	Max	Default	Unit	Description
Pitch	0	127	48		Sets the step pitch, as a MIDI note number.
Division	0	14	7		Sets the step repeat or ratchet count.
Pattern	0	255	0		Sets the step pattern – which substeps are on or off.
Ties	0	255	0		Sets the step tie pattern – which substeps are ties.
Velocity	1	127	64		Sets the step velocity.
Mod	-10.0	10.0	0.0	V	Sets the step modulation value.

Mute	0	100	0	%	Sets the probability that the step will be muted.
Skip	0	1	0		If on, the step is skipped.
Reset	0	1	0		If on, the step causes a reset.

Randomise parameters

Name	Min	Max	Default	Unit	Description
Randomise	0	1	0		When this parameter changes from 0 to 1, the sequence steps between the current start and end steps will be randomised according to the parameters that follow.
Randomise what	0	3	3		What will be randomised – "Nothing", "Pitches", "Rhythm", or "Both".
Note distribution	0	1	0		Chooses the probability distribution for the note pitches – "Uniform" or "Normal".
Min note	0	127	36		The minimum note pitch, if the distribution is "Uniform".
Max note	0	127	60		The maximum note pitch, if the distribution is "Uniform".
Mean note	0	127	48		The mean note pitch, if the distribution is "Normal".
Note deviation	0	127	12		The note pitch deviation, if the distribution is "Normal".
Min repeat	2	8	2		The minimum repeat count, if a step is a repeat.
Max repeat	2	8	8		The maximum repeat count, if a step is a repeat.
Min ratchet	2	8	2		The minimum ratchet count, if a step is a ratchet.
Max ratchet	2	8	8		The maximum ratchet count, if a step is a ratchet.
Note probability	0	100	50	%	The probability that a note will be on.
Tie probability	0	100	0	%	The probability that a step will be a tie.
Accent probability	0	100	0	%	The probability that a step will be accented.
Repeat probability	0	100	0	%	The probability that a step will be a repeat.
Ratchet	0	100	0	%	The probability that a step will be a ratchet.

probability					
Unaccented velocity	1	127	1		The velocity value to use for unaccented steps. Accented steps use velocity 127.

Routing parameters

Name	Min	Max	Default	Unit	Description
Clock input	0	28	0		The bus to use as the clock input.
Reset input	0	28	0		The bus to use as the reset input.
Reset mode	0	1	0		Sets the mode for the reset input. The options are: • the input is a reset trigger. • the input is a run/stop signal.
Pitch output	0	28	15		The bus to use for the pitch CV output.
Pitch output mode	0	1	1		The standard Add/Replace mode selector as described above, for the pitch CV output.
Gate output	0	28	16		The bus to use for the gate output.
Gate output mode	0	1	1		The standard Add/Replace mode selector as described above, for the gate output.
Velocity output	0	28	0		The bus to use for the velocity output.
Velocity output mode	0	1	1		The standard Add/Replace mode selector as described above, for the velocity output.
Mod output	0	28	0		The bus to use for the modulation CV output.
Mod output mode	0	1	1		The standard Add/Replace mode selector as described above, for the modulation CV output.

MIDI parameters

Name	Min	Max	Default	Unit	Description
Follow MIDI clock	0	1	0		Sets whether the sequencer follows MIDI clock.
Divisor	0	19	4		Sets the clock divisor when following MIDI clock.
MIDI channel	1	16	1		Sets the output MIDI channel.

Output to breakout	0	1	0		Enables MIDI output to the breakout.
Output to Select Bus	0	1	0		Enables MIDI output to the Select Bus.
Output to USB	0	1	0		Enables MIDI output to USB.
Output to internal	0	1	0		Enables internal MIDI output - that is, MIDI is sent to the other algorithms.
Mod CC	0	127	1		The MIDI CC to generate from the modulation lane.

Sequencer directions

Forwards	The steps play in ascending order e.g. 1, 2, 3, 4, 1, 2, 3, 4 …
Reverse	The steps play in descending order e.g. 4, 3, 2, 1, 4, 3, 2, 1 …
Alt	The steps play alternately ascending and descending e.g. 1, 2, 3, 4, 3, 2, 1, 2 …
Alt2	The steps play alternately ascending and descending, repeating the first and last steps e.g. 1, 2, 3, 4, 4, 3, 2, 1, 1, 2 …
Random	The steps play in a random order.
Random2	Steps are played in a random order, except that the same step cannot be played twice in a row.
Random3	All the steps are played once in a random order, then again in another random order, and so on.

Sequencer permutations

Note that if the direction is set to one of the random options, the permutation has no effect.

Straight	The steps are played in their numerical order e.g. 1, 2, 3, 4, 5, 6, 7, 8 …
Odd/Even	Odd numbered steps play, followed by the even numbered steps e.g. 1, 3, 5, 7, 2, 4, 6, 8, 1 …
Halves	The second half of the steps are interleaved with the first half e.g. 1, 5, 2, 6, 3, 7, 4, 8, 1 …
Inwards	The first and last steps play, then the second and penultimate, and so on e.g.

	1, 8, 2, 7, 3, 6, 4, 5, 1 …

Stopwatch

"Real-time clock for timing things"

File format guid: 'stpw'

Specifications: None

Description

This algorithm is simply a clock, for timing how long things (for example, your performance) have been going on, or for showing a countdown (for example, until you should stop your performance and get off the stage).

It has no effect on any input/output bus and consumes almost no CPU.

GUI

The display shows the stopwatch time.

While the stopwatch is visible and running the screensaver is prevented from activating.

Setup parameters

Name	Min	Max	Default	Unit	Description
Mode	0	1	0		Chooses whether to display a timer or a countdown.
Start/stop mode	0	1	0		Sets whether the start/stop control is a gate (start when high, stop when low) or a trigger (trigger to start, another trigger to stop).

Countdown parameters

Name	Min	Max	Default	Unit	Description
Hours	0	24	0		Sets the number of hours for the countdown.
Minutes	0	59	0		Sets the number of minutes for the countdown.
Seconds	0	59	0		Sets the number of seconds for the countdown.

Controls parameters

Name	Min	Max	Default	Unit	Description
Start/stop	0	1	0		Starts and stops the timer/countdown.
Reset	0	1	0		Sets the timer to zero, or resets the countdown to that specified by the countdown parameters.

Tuner (fancy)

"A sophisticated tuner"

File format guid: 'tunf'

Specifications: None

Description

This algorithm provides four simultaneous tuners.

It uses an autocorrelation pitch detection method which is fairly CPU heavy, but which provides reliable pitch detection for most signals. For simpler tones, you may prefer the "Tuner (simple)" algorithm.

GUI

The display shows the detected pitch of the active channels, as a MIDI note number and cents, and as a value in Hz. It also shows a bar which grows to the left or right depending on how flat or sharp the note is relative to the 'true' pitch of the detected note.

Parameters

Name	Min	Max	Default	Unit	Description
Channel 1	0	28	1		The input bus for tuner number 1.
Channel 2	0	28	0		The input bus for tuner number 2.
Channel 3	0	28	0		The input bus for tuner number 3.
Channel 4	0	28	0		The input bus for tuner number 4.
Range	1	20	9		Sets the range of pitches which can be tracked. Set this appropriately for the lowest note that you need to tune.
Track bias	0	100	10	%	An internal parameter of the pitch tracking algorithm, which can help it avoid errors of octave in some circumstances. The default value is recommended in most cases.

Env threshold	-80	0	-40	dB	Sets a signal threshold below which no attempt is made to track pitch.

Tuner (simple)

"A basic tuner for simple tones"

File format guid: 'tuns'

Specifications: None

Description

This algorithm provides four simultaneous tuners.

It uses a simple pitch detection method which will only work reliably for simple tones, but happily "simple tones" covers most of those that you will get from an analogue VCO, which in a Eurorack environment you might find yourself tuning often.

Being simple, it consumes little CPU.

For a more sophisticated tuner, please see the "Tuner (fancy)" algorithm.

GUI

The display shows the detected pitch of the active channels, as a MIDI note number and cents, and as a value in Hz. It also shows a bar which grows to the left or right depending on how flat or sharp the note is relative to the 'true' pitch of the detected note.
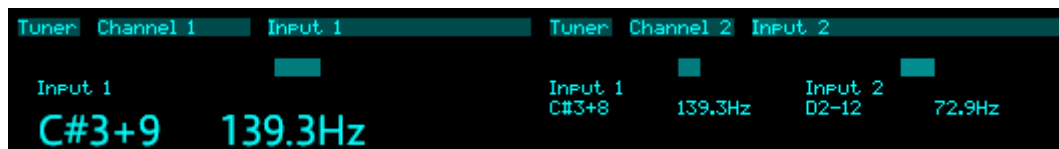
Parameters

Name	Min	Max	Default	Unit	Description
Channel 1	0	28	1		The input bus for tuner number 1.
Channel 2	0	28	0		The input bus for tuner number 2.
Channel 3	0	28	0		The input bus for tuner number 3.
Channel 4	0	28	0		The input bus for tuner number 4.

USB audio (from host)

"Receives audio from USB host"

File format guid: 'usbf'

Specifications: None

Description

This algorithm exposes the USB audio from the host, if one is connected.

The host sees 8 output channels. Each one of these can be output to any of the disting NT's busses.

Note that the signals from the USB host don't just have to emerge directly from the module's outputs; they could just as well be the inputs to other algorithms, allowing you to process audio from the host through the disting.

Note also that "USB audio" doesn't mean that the signals have to be audio; they could also be CVs, if you'd like to exchange signals with, say, VCV Rack or similar.

Parameters

Name	Min	Max	Default	Unit	Description
to	0	28	13		The bus on which to output USB channel 1.
mode	0	1	1		The standard Add/Replace mode selector as described above, for channel 1.
Etc. up to channel 8					

USB audio (to host)

"Sends audio to USB host"

File format guid: 'usbt'

Specifications: None

Description

This algorithm routes signals to the USB audio host, if one is connected.

The host sees 12 input channels. Each one of these can be set to receive any of the disting NT's busses.

Note that the signals seen by the algorithm, and so by the host, depend where in the list of algorithms this one appears. If it's the first algorithm, it can send to the host the module's own inputs, unmodified; if it's the last algorithm, it can send to the host the outputs of all the other algorithms.

Note also that "USB audio" doesn't mean that the signals have to be audio; they could also be CVs, if you'd like to exchange signals with, say, VCV Rack or similar.

Common parameters

Name	Min	Max	Default	Unit	Description
USB channel 1 from	1	28	1		The bus to send to USB channel 1.
USB channel 2 from	1	28	2		The same for USB channel 2.
Etc. up to channel 12					

VCA/Multiplier

"Four quadrant multiplier"

File format guid: 'vcam'

Specifications:

- Channels, 1-8: The number of bus channels to process.

Description

This algorithm is a voltage multiplier, which can be used as a VCA. The CV on the common channel is used to multiply the voltages on the other channels.

Common parameters

Name	Min	Max	Default	Unit	Description
Input	1	28	1		The common input bus.
Clamp to 0V	0	1	1		Whether to clamp the common input to 0V (typical VCA usage) or not (four quadrant multiplier usage).
Divider	1	12	8	V	Sets the scaling of the input CV that corresponds to "unity gain".

Per-channel parameters

Name	Min	Max	Default	Unit	Description
Input	1	28	1		The channel input bus.
Output	0	28	0		The output bus. If set to 'None', the input bus is used as output, and the mode is always 'Replace'.
Output mode	0	1	1		The standard Add/Replace mode selector as described above.

VCF (State Variable)

"Second order LP/BP/HP filter"

File format guid: 'fsvf'

Specifications: None

Description

This algorithm is a voltage controlled filter using the common 'State Variable' topology, which yields simultaneous low-, band-, and highpass filter responses.

All three filter outputs are available individually, plus an output which can be blended from lowpass, through bandpass, to highpass.

The input and the various outputs use a contiguous range of busses, starting with the one set via the parameters, and with the channel count set by the 'Width' parameter. For example, to filter stereo signals, set the Width to 2.

GUI

The display simply shows the filter centre frequency, in Hz and as a MIDI note number.

Filter parameters

Name	Min	Max	Default	Unit	Description
Blend	0	200	0		Sets the blend for the blended output.
Sweep	-36.00	84.00	0.00	ST	A manual frequency sweep control; offsets the frequency CV.
Resonance	0	100	20		Sets the filter resonance.
Saturate	0	1	1		Enables an internal saturation stage, which can keep high resonances from running out of control.

Gain parameters

Name	Min	Max	Default	Unit	Description

Blended gain	-40	6	0	dB	Level control for the blended output.
Lowpass gain	-40	6	0	dB	Level control for the lowpass output.
Bandpass gain	-40	6	0	dB	Level control for the bandpass output.
Highpass gain	-40	6	0	dB	Level control for the highpass output.

Routing parameters

Name	Min	Max	Default	Unit	Description
Audio input	1	28	1		The audio input bus.
Width	1	8	1		The number of busses to process.
Frequency input	0	28	0		The frequency CV input bus (1V/octave).
Resonance input	0	28	0		The resonance CV input bus. Scaled such that 5V corresponds to the full range.
Blended output	0	28	13		The output bus for the blended signal.
Blended mode	0	1	0		Add/replace mode for the blended signal.
Lowpass output	0	28	0		The output bus for the lowpass signal.
Lowpass mode	0	1	0		Add/replace mode for the lowpass signal.
Bandpass output	0	28	0		The output bus for the bandpass signal.
Bandpass mode	0	1	0		Add/replace mode for the bandpass signal.
Highpass output	0	28	0		The output bus for the highpass signal.
Highpass mode	0	1	0		Add/replace mode for the highpass signal.

VCO with waveshaping

"Simple VCO with adjustable outputs"

File format guid: 'vcow'

Specifications: None

Description

This algorithm is based on the OG disting algorithms "B-8 VCO with waveshaping" and "B-7 VCO with linear FM".

There are four oscillator outputs:

- Triangle/saw
- Square/pulse
- Sub-octave square
- Sine

A shared waveshape parameter/CV controls both the shape of the triangle/saw and sine waves and the pulse width of the square/pulse wave.

Linear through-zero FM is available via the 'FM input' parameter.

If the 'Sync' input is used, a rising edge on the input sets the VCO phase to zero, which can produce 'oscillator sync' sounds[72].

Note that it is perfectly valid to set the output bus for the four signals to the same bus, in which case they can be blended to taste with their gain parameters.

GUI

The display shows a single cycle of the triangle/saw and square/pulse waveshapes, and the VCO's pitch as a MIDI note name plus cents and as a frequency in Hz.
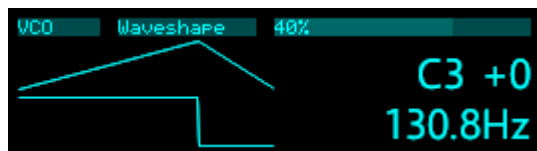
VCO parameters

Name	Min	Max	Default	Unit	Description
Waveshape	-100	100	0	%	Sets the wave shape/pulsewidth.

---

72 https://en.wikipedia.org/wiki/Oscillator_sync

Octave	-16	8	0		Adjusts the VCO tuning in octaves.
Transpose	-60	60	0	ST	Adjusts the VCO tuning in semitones.
Fine tune	-100	100	0	cents	Adjusts the VCO tuning in cents.
Oversampling	0	2	0		Enables oversampling, to reduce aliasing noise at higher frequencies. The options are "None", "2x", and "4x".
FM scale	1	1000	100	Hz	Sets the Hz/V sensitivity of the linear FM input.

Gain parameters

Name	Min	Max	Default	Unit	Description
Triangle/ saw amplitude	0.00	10.00	10.00	V	Sets the amplitude of the triangle/saw output (before it's affected by the gain).
Square/ pulse amplitude	0.00	10.00	10.00	V	Sets the amplitude of the square/pulse output (before it's affected by the gain).
Sub amplitude	0.00	10.00	10.00	V	Sets the amplitude of the sub-octave output (before it's affected by the gain).
Sine amplitude	0.00	10.00	10.00	V	Sets the amplitude of the sine output (before it's affected by the gain).
Triangle/ saw gain	-40	6	0	dB	Sets the level of the triangle/saw output.
Square/ pulse gain	-40	6	0	dB	Sets the level of the square/pulse output.
Sub gain	-40	6	0	dB	Sets the level of the sub-octave output.
Sine gain	-40	6	0	dB	Sets the level of the sine output.

Routing parameters

Name	Min	Max	Default	Unit	Description
Pitch input	0	28	1		The pitch CV input (1V/octave).
Shape input	0	28	0		The waveshape CV input (scaled so that ±5V corresponds to the ±100% parameter range).
Sync input	0	28	0		The oscillator sync input.

FM input	0	28	0		The linear FM input.
Triangle/ saw output	0	28	13		The output bus for the triangle/saw signal.
Triangle/ saw mode	0	1	0		The add/replace mode for the triangle/saw signal.
Square/ pulse output	0	28	0		The output bus for the square/pulse signal.
Square/ pulse mode	0	1	0		The add/replace mode for the square/pulse signal.
Sub output	0	28	0		The output bus for the sub-octave signal.
Sub output mode	0	1	0		The add/replace mode for the sub-octave signal.
Sine output	0	28	0		The output bus for the sine signal.
Sine output mode	0	1	0		The add/replace mode for the sine signal.

VCO - wavetable

"A wavetable VCO"

File format guid: 'vcot'

Specifications: None

Description

This algorithm is a simple VCO which uses wavetables for its wave shapes. Please see the section above on how wavetables are formatted and arranged on the MicroSD card.

GUI

The display shows a single cycle of the current wavetable waveform, and the VCO's pitch as a MIDI note name plus cents and as a frequency in Hz.
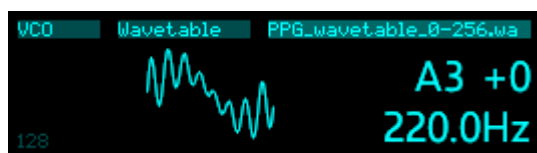
VCO parameters

Name	Min	Max	Default	Unit	Description
Wavetable	0				Chooses the wavetable.
Wave offset	-100.0	100.0	0.0	%	Sets the position within the wavetable.
Octave	-16	8	0		Adjusts the VCO tuning in octaves.
Transpose	-60	60	0	ST	Adjusts the VCO tuning in semitones.
Fine tune	-100	100	0	cents	Adjusts the VCO tuning in cents.

Gain parameters

Name	Min	Max	Default	Unit	Description
Amplitude	0.00	10.00	10.00	V	Sets the amplitude of the output (before it's affected by the gain).
Gain	-40	6	0	dB	Sets the level of the output.

Routing parameters

Name	Min	Max	Default	Unit	Description
Pitch input	0	28	1		The pitch CV input (1V/octave).
Wave input	0	28	0		Sets an input bus to modulate the wave offset.
Output	0	28	13		The output bus.
Output mode	0	1	0		The standard Add/Replace mode selector as described above.

UI Scripts

The disting NT offers the ability to completely redefine the module's UI using the popular scripting language [Lua](https://www.lua.org)[73].

It is anticipated that this will be particularly useful in a live performance scenario, where you might want to reduce the available controls to a few key items, and not want to see (or risk changing) the bulk of the preset.

The version of Lua implemented in the disting NT is currently 5.4.6.

Scripts can be loaded from anywhere on the MicroSD card, but the examples provided are in the 'ui_scripts' folder.

> This feature can probably best be considered a "technology preview" - it works, but there's so much more that could be added. We're very interested in your feedback on whether this is a useful feature to develop further.

Running a script

From the 'UI Scripts' menu, choose 'Run UI script', which will let you browse the MicroSD card for your chosen script.

The script completely takes over the module UI until it is exited. The combination of pressing all four pushbuttons at once is reserved as a means of exiting a UI script. You are also free to define your own choice of means of leaving the scripted UI via the script itself.

If the script fails for any reason the module will do its best to return any error information generated by the Lua interpreter.

Anatomy of a script

There are three main sections that a script needs to implement:

- Initialisation. The module calls this section once when the script is loaded to allow it to perform any required setup. This will typically include initialising any local state, and identifying the algorithms and parameters that the script will control.
- Responding to UI events. The script can choose to respond to any or all button pushes, encoders turns, pot turns etc.
- Drawing the UI. This can be as simple as showing the value of the parameter being controlled, or completely freeform vector graphics and text.

---

73 https://www.lua.org

With that in mind, here is a simple example:

```
local augustus
local p_multiplier

return
{
      name = 'Example UI script'
,     author = 'Expert Sleepers Ltd'
,     description = 'controls one parameter of Augustus Loop'

,     init = function()
            augustus = findAlgorithm( "Augustus Loop" )
            if augustus == nil then
                  return "Could not find 'Augustus Loop'"
            end
            p_multiplier = findParameter( augustus, "Delay multiplier" )
            if p_multiplier == nil then
                  return "Could not find 'Delay multiplier'"
            end
            return true
      end

,     pot3Turn = function( value )
            setParameterNormalized( augustus, p_multiplier, value )
      end

,     button2Push = function()
            exit()
      end

,     draw = function()
            drawStandardParameterLine()
            drawText( 30, 40, "Hello!" )
      end
}
```

The return value from the script is a table, most of the elements of which are functions to handle various events. You can define some script-local variables before returning the table - here 'augustus' and 'p_multiplier' are such variables.

The 'name', 'author', and 'description' elements are optional but encouraged.

The 'init' function is called once when the script is loaded. In this case, the script takes the opportunity to search for and cache the indices of the algorithm and parameter that it would like to control. This is purely in the interests of efficiency - it could search every time it wanted to change the parameter. Or indeed, the script could just hard code the indices, but that would make it very brittle and likely to break if any changes were made to the preset that the script is designed to work with. The 'init' function returns true if everything is OK; if it doesn't, the module will abandon the script and revert to normal operation.

This example script watches for two UI events - turning pot 3, and pressing button 2. The latter causes the script to exit and return to the normal module UI; the former sets a parameter on the algorithm that was previously identified.

The final function 'draw' is where the script gets to actually display something. 'drawStandardParameterLine()' draws the most recently changed parameter across the top of the screen, as in the default algorithm view.

Functions that a script can define

init

Called once when the script is loaded.

Takes no arguments.

Return Boolean true on success, else a string indicating the cause of failure.

pot1Turn/pot2Turn/pot3Turn

Called when the relevant pot is turned.

One argument: a number in the range [0.0,1.0].

Returns nothing.

encoder1Turn/encoder2Turn

Called when the relevant encoder is turned.

One argument: a number which is +1 for clockwise movement or -1 or anticlockwise movement.

Returns nothing.

Button push/release functions

All take no arguments and return nothing. The following are defined:

button1Push, button2Push, button3Push, button4Push

button1Release, button2Release, button3Release, button4Release

pot1Push, pot2Push, pot3Push

pot1Release, pot2Release, pot3Release

encoder1Push, encoder2Push

encoder1Release, encoder2Release

draw

Called (continuously) to allow the UI to draw on the display.

Takes no arguments; returns nothing.

Drawing

Functions that perform drawing use pixels as their coordinate unit. The display is 256x64 pixels, with the origin (0,0) at the top left.

The display supports 16 shades; functions that take a colour argument use values from 0 (pixel off) to 15 (pixel fully lit).

Functions that a script may call

In addition to the base Lua language features, the following functions are implemented on the disting NT.

exit

Returns control to the normal module UI.

No arguments; returns nothing.

findAlgorithm

Allows the script to look up an algorithm within the preset.

Takes one argument, a string. Currently this is matched against the algorithms' (customised) names.

Returns as many results as matches found. Each is a 1-based index into the list of algorithms.

findParameter

Allows the script to look up a parameter within an algorithm.

Takes two arguments: a number, the algorithm index; and a string, which is matched against the parameter names.

Returns as many results as matches found. Each is a 1-based index into the list of parameters.

For algorithms with variable numbers of parameters (according to their specification), the string is matched against the base parameter name and the prefixed parameter name as seen in, for example, the preset editor tool. Say you have an LFO algorithm with two channels - the string 'Speed' will match against the parameter for all channels, so this function will return two results, but the string '1:Speed' or '2:Speed' will give you only the parameter for the specific channel.

setParameter

Sets an algorithm parameter's value.

Takes three arguments: the algorithm index, the parameter index, and the parameter value.

Returns nothing.

setParameterNormalized

The same as setParameter, except the third argument is a number in the range [0.0,1.0], which is mapped onto the full range of the parameter being changed.

getParameter

Gets an algorithm parameter's value.

Takes two arguments: the algorithm index, and the parameter index.

Returns the value.

focusParameter

Sets the current algorithm and parameter (the parameter that drawStandardParameterLine() will show).

Takes two arguments: the algorithm index, and the parameter index.

Returns the nothing.

standardPot1Turn/standardPot2Turn/standardPot3Turn

Performs the standard function of the pot when in the algorithm view. Typically used as e.g.

```
,    pot1Turn = function( value )
        standardPot1Turn( value )
    end
```

drawAlgorithmUI

Draws the specified algorithm's custom GUI.

Takes one argument: the algorithm index.

Returns the nothing.

drawBox

Draws a box (an unfilled rectangle).

Takes five arguments: top left x/y, bottom right x/y, and colour. Coordinates are converted to integer values before drawing.

Returns nothing.

drawLine

Draws a line.

Takes five arguments: top left x/y, bottom right x/y, and colour. Coordinates are converted to integer values before drawing.

Returns nothing.

drawParameterLine

Draws a line of information similar to that drawn by drawStandardParameterLine, but for a specific algorithm and parameter.

Takes three arguments: the algorithm index, the parameter index, and a y coordinate offset (from the default position at the top of the screen).

Returns nothing.

drawRectangle

Draws a filled rectangle.

Takes five arguments: top left x/y, bottom right x/y, and colour. Coordinates are converted to integer values before drawing.

Returns nothing.

drawSmoothLine

Draws an antialiased line.

Takes five arguments: top left x/y, bottom right x/y, and colour, all of which can meaningfully be floating point values.

Returns nothing.

drawStandardParameterLine

Draws the standard algorithm parameter line at the top of the screen, as in the default algorithm view, showing the most recently modified parameter.

No arguments; returns nothing.

drawText

Draws a string on the display in the module's standard font.

Takes three arguments: the x and y coordinates, and the string to draw. The y coordinate specifies the text baseline.

Returns nothing.

MIDI SysEx reference

The disting NT supports a variety of functions via MIDI System Exclusive (SysEx) messages.

The browser-based tools [here](#)[74] can be considered as informal documentation of these messages, as in most cases the messages only exist to support these tools.

SysEx Header

All SysEx messages are prefixed with a manufacturer's ID, which is a unique series of hex bytes assigned by the MIDI Manufacturers Association. The Expert Sleepers ID is 00H 21H 27H, so all SysEx messages relating to Expert Sleepers hardware will begin

F0 00 21 27

Messages for the disting NT follow this with 6DH:

F0 00 21 27 6D

followed by the module's SysEx ID (see the Settings, above)

F0 00 21 27 6D <SysEx ID>

and then with a byte to identify the specific type of message e.g.

F0 00 21 27 6D <SysEx ID> 01

16 bit values

Where '16 bit value' is indicated below, this is a sequence of 3 bytes:

<most significant 2 bits> <middle 7 bits> <least significant 7 bits>

Received SysEx messages

01H – Take screenshot

F0 00 21 27 6D <SysEx ID> 01 F7

This causes the disting NT to respond with a SysEx message containing a screenshot of what is currently on the module's display, using the '33H – Screenshot' format, below.

04H – Set real-time clock

F0 00 21 27 6D <SysEx ID> 04 <time MSB> <time> <time> <time> <time LSB> F7

This sets the module's real-time clock to the time represented as a 32 bit number of seconds since the start of the epoch (January 1st, 1970).

---

74 https://github.com/expertsleepersltd/distingNT/tree/main/tools

11H – .scl file

F0 00 21 27 6D <SysEx ID> 11 <ignored byte> <file contents> F7

Sends a Scala .scl file for the module to use.

12H – .kbm file

F0 00 21 27 6D <SysEx ID> 11 <ignored byte> <file contents> F7

Sends a Scala .kbm file for the module to use.

22H – Request version string

F0 00 21 27 6D <SysEx ID> 22 F7

This causes the disting NT to respond with a SysEx message containing the module's version string as text, using the '32H – Message' format, below.

30H – Request number of algorithms

F0 00 21 27 6D <SysEx ID> 30 F7

Queries the number of algorithms - that is, the number of different algorithms that the module can run, not the number of algorithms instantiated in the current preset. The response uses the '30H – Number of algorithms' format, below.

31H – Request algorithm info

F0 00 21 27 6D <SysEx ID> 31 <16 bit index> F7

Queries for details of the indexed algorithm (in the list returned by the 30H message). The response uses the '31H – Algorithm info' format, below.

32H – Add algorithm

F0 00 21 27 6D <SysEx ID> 32 <guid> <16 bit specification value> <16 bit specification value> F7

Adds an algorithm to the current preset.

33H – Remove algorithm

F0 00 21 27 6D <SysEx ID> 33 <algorithm index> F7

Removes an algorithm from the current preset.

34H – Load preset

F0 00 21 27 6D <SysEx ID> 34 <append> <NULL terminated ASCII string> F7

Loads or appends the preset specified by the filename string.

35H – New preset

F0 00 21 27 6D <SysEx ID> 35 F7

Starts a new preset (exactly as the "New Preset" menu).

36H – Save preset

F0 00 21 27 6D <SysEx ID> 36 <option> F7

Saves the preset. The option value can be 0 (prompt for file overwrite), 1 (never overwrite), or 2 (always overwrite).

37H – Move algorithm

F0 00 21 27 6D <SysEx ID> 37 <algorithm index> <new index> F7

Requests that an algorithm be moved to a new position in the list.

40H – Request algorithm guid

F0 00 21 27 6D <SysEx ID> 40 <algorithm index> F7

Queries the indexed algorithm in the current preset. The response uses the '40H – Algorithm guid' format, below.

41H – Request preset name

F0 00 21 27 6D <SysEx ID> 41 F7

Requests the current preset name. Responds with '41H – Preset name', as below.

42H – Request number of parameters

F0 00 21 27 6D <SysEx ID> 42 <algorithm index> F7

Requests the number of parameters in the indexed algorithm. Responds with '42H – Number of parameters', as below.

43H – Request parameter info

F0 00 21 27 6D <SysEx ID> 43 <algorithm index> <16 bit parameter number> F7

Requests information for the given parameter in the indexed algorithm. Responds with '43H – Parameter info', as below.

44H – Request all parameter values

F0 00 21 27 6D <SysEx ID> 44 <algorithm index> F7

Requests the current values of all parameters in the indexed algorithm. Responds with '44H – All parameter values', as below.

45H – Request parameter value

F0 00 21 27 6D <SysEx ID> 45 <algorithm index> <16 bit parameter number> F7

Requests the value of the given parameter in the indexed algorithm. Responds with '45H – Parameter value', as below.

46H – Set parameter value

F0 00 21 27 6D <SysEx ID> 46 <algorithm index> <16 bit parameter number> <16 bit value> F7

Sets the value of the given parameter in the indexed algorithm.

47H – Set preset name

F0 00 21 27 6D <SysEx ID> 47 <NULL terminated ASCII string> F7

Sets the preset name.

48H – Request unit strings

F0 00 21 27 6D <SysEx ID> 48 F7

Requests string descriptions of the possible parameter units (Hz, ms etc.). Responds with '48H – Unit strings', as below.

49H – Request enum strings

F0 00 21 27 6D <SysEx ID> 49 <algorithm index> <16 bit parameter number> F7

Requests the value strings for an 'enum' type parameter. Responds with '49H – Enum strings', as below.

4AH – Set focus

F0 00 21 27 6D <SysEx ID> 4A <algorithm index> <16 bit parameter number> F7

Sets the given parameter to be the currently active one in the module's display.

4BH – Request mappings

F0 00 21 27 6D <SysEx ID> 4B <algorithm index> <16 bit parameter number> F7

Requests the mappings for a parameter. Responds with '4BH – Mapping', as below.

4DH – Set mapping

F0 00 21 27 6D <SysEx ID> 4D <algorithm index> <16 bit parameter number> <packed mapping data> F7

Sets the CV mapping for the given parameter.

4EH – Set MIDI mapping

F0 00 21 27 6D <SysEx ID> 4E <algorithm index> <16 bit parameter number> <packed mapping

data> F7

Sets the MIDI mapping for the given parameter.

4FH – Set I2C mapping

F0 00 21 27 6D <SysEx ID> 4F <algorithm index> <16 bit parameter number> <packed mapping data> F7

Sets the I2C mapping for the given parameter.

50H – Request parameter value string

F0 00 21 27 6D <SysEx ID> 50 <algorithm index> <16 bit parameter number> F7

Requests the value string for a parameter. Responds with '50H – Parameter value string', as below.

60H – Request number of algorithms

F0 00 21 27 6D <SysEx ID> 60 F7

Requests the number of algorithms in the current preset. Responds with '60H – Number of algorithms', as below.

61H – Request routing

F0 00 21 27 6D <SysEx ID> 61 <algorithm index> F7

Requests routing information for the given algorithm. Responds with '61H – Routing information', as below.

Sent SysEx messages

30H – Number of algorithms

F0 00 21 27 6D <SysEx ID> 30 <16 bit value> F7

This message is transmitted in response to '30H – Request number of algorithms'.

31H – Algorithm info

F0 00 21 27 6D <SysEx ID> 31 <16 bit index> <4 byte guid> <number of specifications> [<specification min> <max> <default> <type> …] <NULL terminated algorithm name> [<NULL terminated specification name> …] F7

This message is transmitted in response to '31H – Request algorithm info'.

32H – Message

F0 00 21 27 6D <SysEx ID> 32 <NULL terminated ASCII string> F7

This message is transmitted in response to any request for a string e.g the version string.

33H – Screenshot

F0 00 21 27 6D <SysEx ID> 33 00 <screenshot data> F7

This message is transmitted in response to a '01H – Take screenshot' message.

40H – Algorithm guid

F0 00 21 27 6D <SysEx ID> 40 <index> <4 byte guid> F7

This message is transmitted in response to a '40H – Request algorithm guid' message.

41H – Preset name

F0 00 21 27 6D <SysEx ID> 41 <NULL terminated ASCII string> F7

Contains the current preset name.

42H – Number of parameters

F0 00 21 27 6D <SysEx ID> 42 <algorithm index> <16 bit number of parameters> F7

Contains the number of parameters in the indexed algorithm.

43H – Parameter info

F0 00 21 27 6D <SysEx ID> 43 <algorithm index> <16 bit parameter number> <16 bit minimum> <16 bit maximum> <16 bit default> <unit> <ASCII string> F7

Contains information for the given parameter in the indexed algorithm.

44H – All parameter values

F0 00 21 27 6D <SysEx ID> 44 <algorithm index> [<16 bit value>] F7

Contains the current values of all parameters in the indexed algorithm.

45H – Parameter value

F0 00 21 27 6D <SysEx ID> 45 <algorithm index> <16 bit parameter number> <16 bit value> F7

Contains the value of the given parameter in the indexed algorithm.

48H – Unit strings

F0 00 21 27 6D <SysEx ID> 48 <number of strings> [<ASCII string>] F7

Contains an array of string descriptions of the possible parameter units (Hz, ms etc.).

49H – Enum strings

F0 00 21 27 6D <SysEx ID> 49 <algorithm index> <16 bit parameter number> <number of strings> [<ASCII string>] F7

Contains the value strings for an 'enum' type parameter.

4BH – Mapping

F0 00 21 27 6D <SysEx ID> 4B <algorithm index> <16 bit parameter number> <version number> <mapping data> F7

Contains the mapping information for one parameter.

50H – Parameter value string

F0 00 21 27 6D <SysEx ID> 50 <algorithm index> <16 bit parameter number> <ASCII string> F7

Contains a value string for a parameter.

60H – Number of algorithms

F0 00 21 27 6D <SysEx ID> 60 <count> F7

Sent in response to '60H – Request number of algorithms', as above.

61H – Routing information

F0 00 21 27 6D <SysEx ID> 61 <algorithm index> <routing data> F7

Sent in response to '61H – Request routing', as above.

I2C reference

In general the disting NT acts as a follower on the I2C bus, not as a leader. It receives messages in the following format:

<address> <command> <optional bytes according to command>

A table of supported commands is below.

Some commands are "get" commands. The disting expects the get command to be followed immediately by a read of the requested data.

Devices on an I2C bus have an address, which a sending device uses to identify the intended recipient. The disting NT's address is set in the Settings (see above).

Value ranges

Preset, algorithm & parameter numbers are 1-based.

Voltages (and related quantities e.g. pitch) are signed and scaled as 16384 ↔ 10V.

Velocities are 0-16384.

I2C Channels

The note on/off messages have the concept of a 'channel', which is entirely analogous to a MIDI channel. Algorithms which receive notes will have a parameter to set which I2C channel to receive on, and will only respond to notes on that channel. Messages which don't explicitly contain a channel use the last channel set via the 0x6B message.

Messages

Controllers

Set i2c controller X to value Y

<address> 0x11 <controller number> <value MSB> <value LSB>

These messages are used via the Mappings (see above) to control algorithm parameters.

Notes

Set pitch for note id.

<address> 0x54 <note id> <pitch MSB> <pitch LSB>

Note on for specified note id.

<address> 0x55 <note id> <velocity MSB> <velocity LSB>

Note off for specified note id.

<address> 0x56 <note id>

All notes off.

<address> 0x57

Set pitch for note id, with channel.

<address> 0x68 <channel> <note id> <pitch MSB> <pitch LSB>

Note on for specified note id, with channel.

<address> 0x69 <channel> <note id> <velocity MSB> <velocity LSB>

Note off for specified note id, with channel.

<address> 0x6A <channel> <note id>

Set channel for subsequent note based commands.

<address> 0x6B <channel>

Updating the firmware

The disting NT's firmware is updated over its USB connection. You will need a computer and the appropriate USB cable to connect the module to it.

You can jump directly to any firmware version - you don't need to apply them incrementally. You can upgrade or downgrade at will, but note that older firmware versions may not be able to read presets saved with newer versions; similarly, if you downgrade you may find that your settings reset to defaults.

We support two methods of flashing new firmware - one using a GUI tool, and one using a script. The latter will probably be quicker and simpler if you're comfortable with using the terminal/command prompt on your computer.

In both cases, you first need to:

- Download the firmware from our website [here](#)[75].

- Put the disting NT into bootloader mode, which you do via the 'Misc' menu. Select "Enter bootloader mode", click past the "Are you sure?", and the module will show you the message "Entering serial downloader" and appear to hang.

If you decide you don't actually want to flash new firmware at this point, simply turn the module off and on again.

Method 1: GUI tool

When downloading the firmware, do not let the browser automatically unzip the file (in Safari the option is "Open safe files after downloading" - that needs to be off). It is most unlikely to work if you attempt to make a zip from the unarchived files.

Download and install the "MCUXpresso Secure Provisioning Tool" for your platform of choice from [this page](#)[76]. It is available for macOS, Linux, and Windows.

Note that various versions of the tool may be available. At the time of writing, the current version is v10. You need to use the correct version that matches the disting NT firmware package that you want to install. The tool version for each firmware version is noted on the firmware download page.


75 https://expert-sleepers.co.uk/distingNTfirmwareupdates.html

76 https://www.nxp.com/design/design-center/software/development-software/mcuxpresso-software-and-tools-/mcuxpresso-secure-provisioning-tool:MCUXPRESSO-SECURE-PROVISIONING

When you first run the tool, you should be presented with the "New Workspace" dialog:
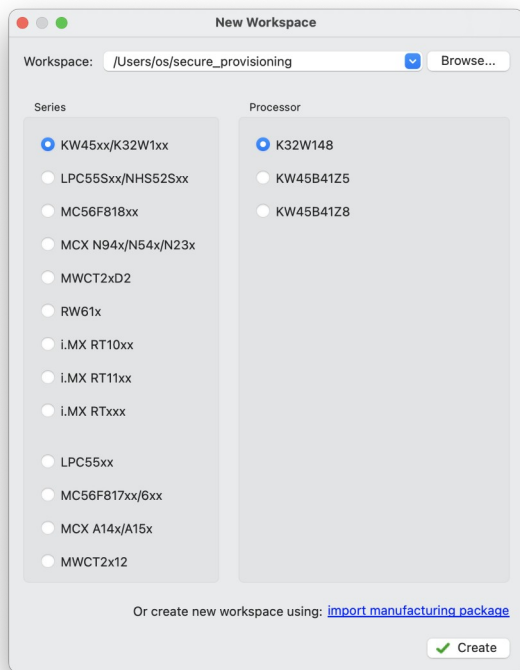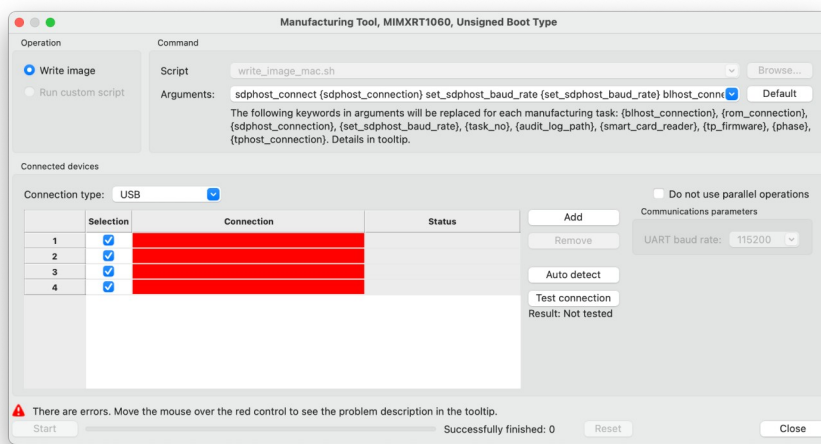

Click the text at the bottom that says "Import manufacturing package". Browse for the firmware zip file that you downloaded.
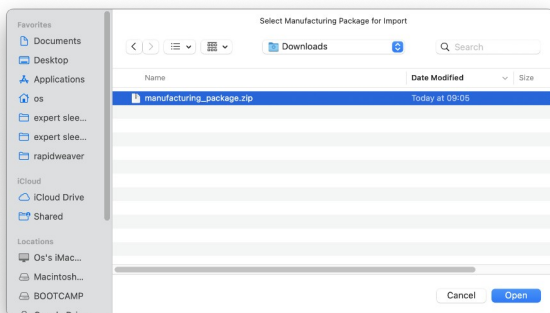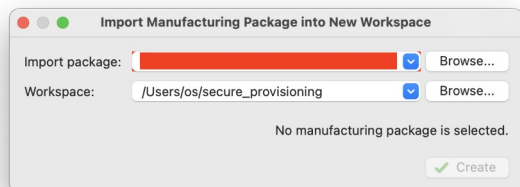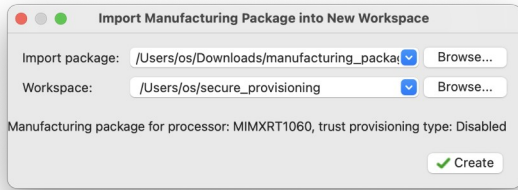
You can also choose where the tool creates its "workspace" - a folder containing a variety of temporary files that you can delete later, or just leave for next time.

Click the "Create" button.

You should now see a window titled "Manufacturing Tool". The connection section may be all red; if so, click "Auto detect". If a disting NT in bootloader mode is connected via USB, the tool will now find it.
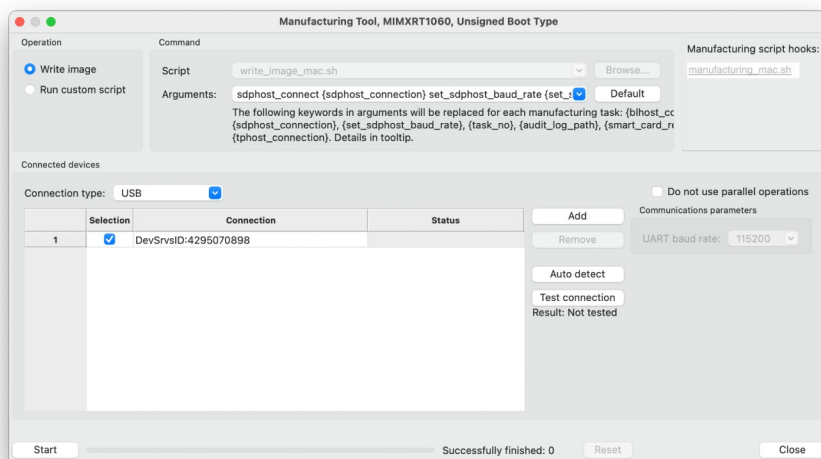
You can then click on the "Start" button (bottom left) to actually perform the firmware update. This should take maybe 15 seconds, and show "Success" when done. Turn the module off and back on to resume normal operation. You may like to check that the "About" screen shows the expected firmware version.

Next time you run the tool, you may see this dialog:
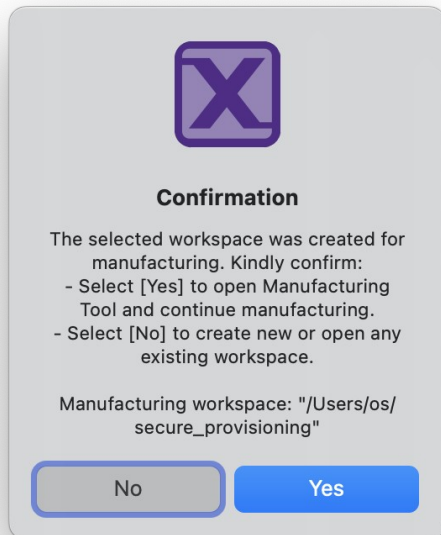
Click "Yes" if you want to reflash the same firmware as before (perhaps you're the lucky owner of two disting NTs). Otherwise, click "No", which will take you back to the "New Workspace" dialog as above.

Method 2: Script

As mentioned above, this method is provided for users who prefer a command line approach over using a GUI application. In preparation for this:

1. Install SPSDK. Instructions are [here](#)[77]. Note that you may also need to install development tools; for example on macOS you may need to install Xcode.
2. Download the scripts and bootloader package from the firmware download page. Unzip these files into a folder on your computer.

To flash the firmware:

1. Delete any existing 'disting_NT.hex' files.
2. Download the 'hex' version of the firmware and unzip it into the same folder that contains the scripts you downloaded previously.
3. Put the disting NT into bootloader mode as described above.

---

77 https://spsdk.readthedocs.io/en/latest/

4. Run the script 'flash.sh' (macOS[78]) or 'flash.bat' (Windows).

---

78 This may work for Linux as well - we've not been able to test it. Let us know if you do!

Acknowledgments

The Kirbinator algorithm was designed in collaboration with [Simon Kirby](79)[79].

Many thanks to all the beta testers.

Fonts

The small pixel font used throughout the UI is 'PixelMix' by Andrew Tyler, for which Expert Sleepers Ltd has a commercial license.

The large font is Microsoft Selawik (Copyright 2015, Microsoft Corporation), licensed under the SIL Open Font License version 1.1.

The tiny font used in parts of the UI is an adapted version of 'Tom Thumb' by Robey Pointer (MIT license).

Freetype

Portions of this software are copyright © 2023 The FreeType Project (www.freetype.org). All rights reserved.

Lua

Copyright © 1994–2024 Lua.org, PUC-Rio.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

---

[79] https://www.simonkirby.net