D.CAM synth squad

Operation Manual

fxpansion

Contents

Credits

DSP Programming – Andrew Simper

Additional Programming – Angus Hewlett, Steve Baker, Paul Chana, SKoT McDonald, Caleb Reach

Project Co-ordination – Steve Baker, Rhiannon Bankston-Thomas

Interface Design – Alex Akers, Steve Baker, Angus Hewlett

Artwork/Packaging Design – Rus Brockman

Web Programming – Andreas Schnetzler

Manual/Copywriting – Mayur Maha

Video Production – Rory Dow

QA – Rob Bantin, Andrew Vernon

Preset design and QA – Peter 'Mod' Slotwinski

Support – Ryan Sellers, Rory Dow, Alex Volmer, Andrew Vernon

Artist Relations – Clare O'Brien

FXpansion USA – Terry Hardin, Daniel Wilkinson

Preset design

Patchen Preston
Kickflip
DJ Subject
Dave Blakely
Bobule
Michal Garstecki
Janis Chaffin
Francesco Silvestri
Daniel Stawczyk
Rory Dow
Angus Hewlett
Peter 'Mod' Slotwinski
Tasmodia tasmodia.com
Raphael S. electric-himalaya.com
Brian Lee xenossoundworks.com
Derik White dustedwilliamsounds.com
SoundsDivine soundsdivine.com

Beta testing

Mark Williams, Sean Ahern, Andrew Capon, Robert Sfeir, Ashley Smith, Mikael Adle, Murray McDowall, Jordan Chillcot, Christopher Reis, Andy Reaburn, Anthony Webster, Paul Sureno, Andrew Shantz, Joost Gransjean, Scott McGrath, Kris Giampa, Roy Queenan, Andreas Beisler, Alice Bernier

Special thanks to Zimon Drake at French Kiss Films: frenchkissfilms.com

fxpansion
www.fxpansion.com

About this manual

It is beyond the scope of this manual to fully explain every concept of synthesis in detail.

A lot of background information is given in order for you to fully understand DCAM: Synth Squad's features, but some prior knowledge of synthesis and audio processing/mixing functions is assumed.

Suggested further reading

The Sound On Sound magazine articles database includes an excellent resource of synthesis information, particularly the comprehensive 'Synth Secrets' series. There are also articles on mixing, effects processing and other relevant topics when using DCAM: Synth Squad:

http://www.soundonsound.com/articles/Technique.php

There is also an excellent book about synthesis available for free viewing on the web (or as a downloadable PDF file), called 'Advanced Programming Techniques for Modular Synthesizers' by James J. Clark. It is written for the Nord Modular DSP synthesis environment, but its information is relevant for all kinds of synthesis, no matter what equipment you're using:

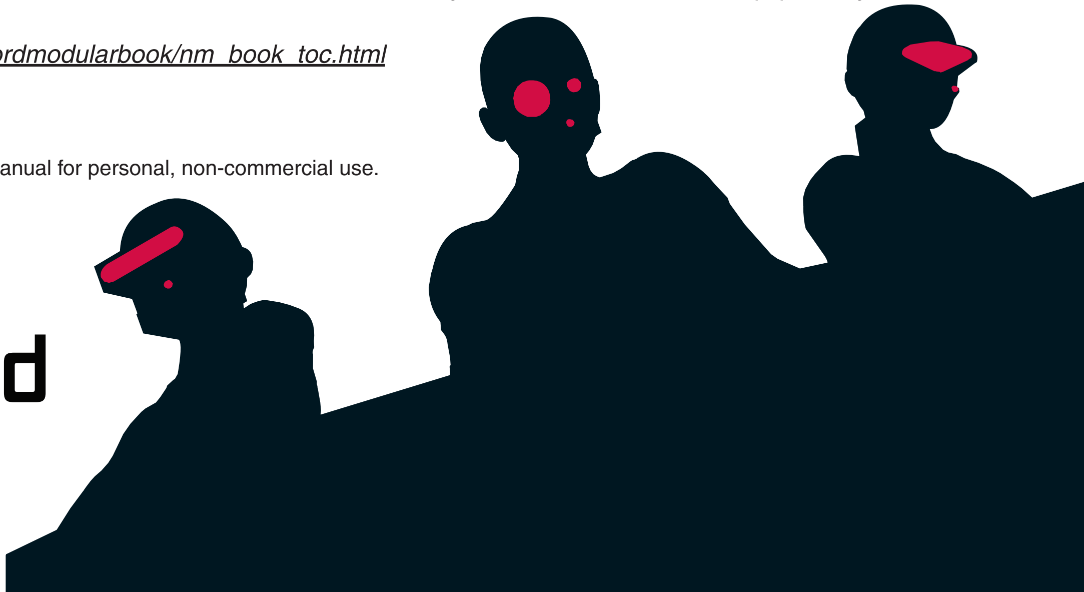
http://www.cim.mcgill.ca/~clark/nordmodularbook/nm_book_toc.html

Printing this manual

Full permission is granted to print this manual for personal, non-commercial use.

D.CAM
synth squad

Manual revision 1.01



1: Introduction

1:1 Welcome to DCAM: Synth Squad

DCAM: Synth Squad at a glance

DCAM: Synth Squad contains 4 individual devices:

Strobe

- souped-up performance synth
- designed for easy programming of classic subtractive analogue synth sounds
- centred around a simple 1-oscillator architecture
- includes a built-in arpeggiator

Strobe is installed as an instrument and as an effect for processing audio through its circuit.

Cypher

- 3-oscillator, complex synth
- FM and other audio-rate modulation
- several other specialized oscillator functions
- a dual-filter/waveshaper architecture
- a built-in arpeggiator

Cypher is a more challenging environment for programming – if you require a traditional analogue-style sound, you'll achieve quicker results using Strobe.

Cypher is installed as an instrument and as an effect for processing audio through its circuit.

Amber

- classic string synthesizer model with divide-down oscillator structure
- constantly generates the notes for a 96-note keyboard range within a single voice
- 3 vintage chorus models and a formant filter

Amber is installed as an instrument and as an effect for processing audio through its circuit.

Fusor

Fusor is an environment for:

- layering synths
- applying FX processing
- step-sequencing
- performing modulation between the various loaded elements

Fusor is installed only as an instrument plugin – it cannot be used as an effect for processing external audio.

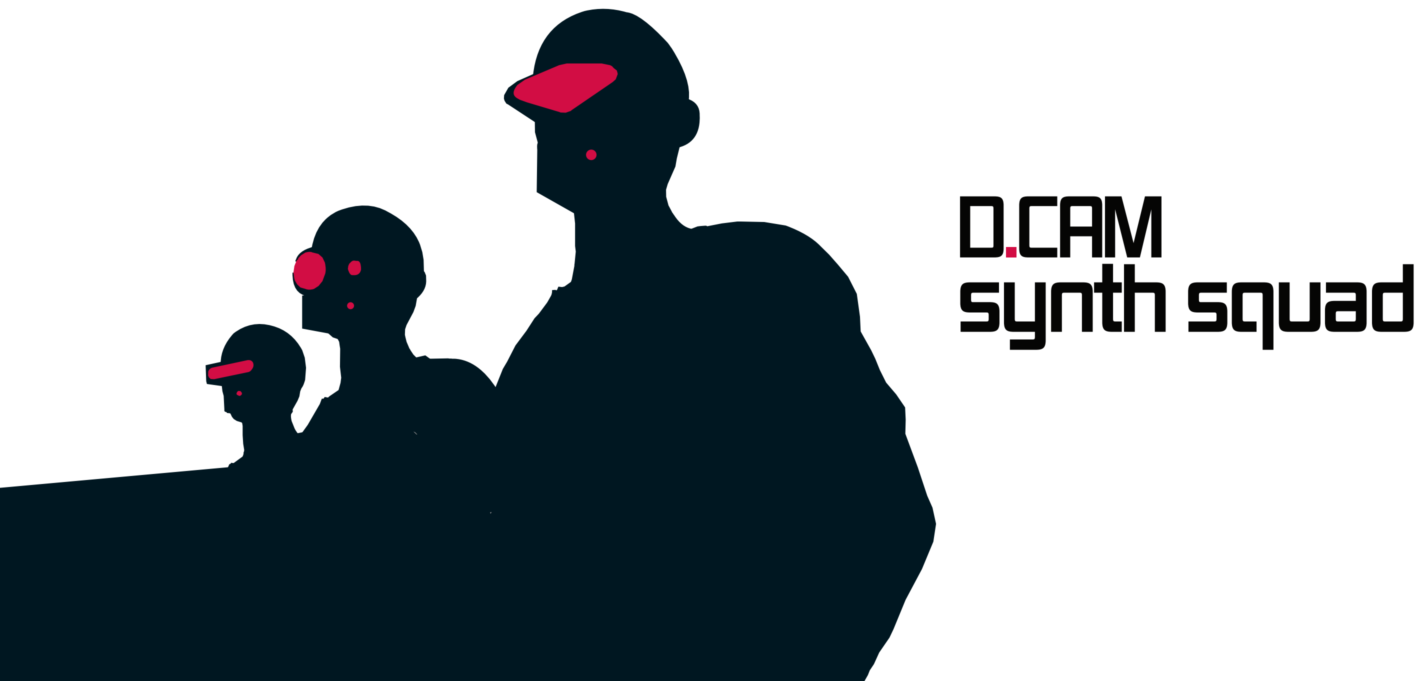
A note on effects

None of the individual synths feature built-in audio effects, with the exception of Amber's chorus, which is essential for a string synthesizer. You may wonder why we took this decision. There are a couple of main reasons:

1. Classic analogue synthesizers have no problem sounding great without any effects

2. We wanted the synths in DCAM: Synth Squad to sound equally great without relying on any effects

However, we realise that effected super-patches are valid in their own right. This is where Fusor comes in – it includes an extremely versatile suite of built-in effects, as well as a comprehensive way of modulating them. If effects are vital to you, we encourage you to use the synths within Fusor to see (and hear) what's possible.



1:2 Shared aspects of all instruments

DCAM: Discrete Component Analogue Modelling

DCAM: Synth Squad has been built by accurately analysing and modelling real-world components and circuits found in vintage analogue synths.

As a result, don't expect a standard 'clean VA' sound from these synths – you would never get perfect waveshapes on a real VCO-based analogue synthesizer.

What you can expect is a set of meticulously crafted synthesizer instruments that sound truly alive in a way that is very rare in the digital world.

Voices, Unison voices and polyphony

Each of the synths within DCAM: Synth Squad features an identical approach to synth voices and unison.

These settings result in 4-note polyphony, with each note being 2 stacked voices.

Strobe and Cypher feature a classic polyphonic architecture – to achieve 2 note polyphony, for example, 2 whole monosynths are effectively played together at the relevant pitches.

You can consider each of these 'monosynths' as a 'voice'.

Additionally, you can set up 'unison voices' in order to stack more than 1 voice for each note of polyphony. This is a classic technique used in analogue synthesis to achieve huge, fat sounds.

Unison voices are taken from the total available voices. The total polyphony is:

voices ÷ unison voices

Amber's architecture, like any classic string machine, is different to a classic polysynth. It provides 96 simultaneous notes within a single voice. However, multiple voices can be used for programming more unconventional sounds.

TransMod modulation

Common between all the DCAM: Synth Squad instruments is the modulation architecture, called the TransMod system. While there are some hard-wired modulation routings in the synths, you will need to use the TransMod modulation system to perform more sophisticated synth programming functions.

By default, no TransMod slot is selected, and 'Main' is highlighted. In this state, only the initial values of parameters can be adjusted.

Main view selected – only the initial value of all parameters is shown.

First TransMod slot selected – controls can now set and display modulation depths away from initial values.

Clicking on one of the 8 TransMod slots changes the controls so that you can set a modulation depth on each of them.

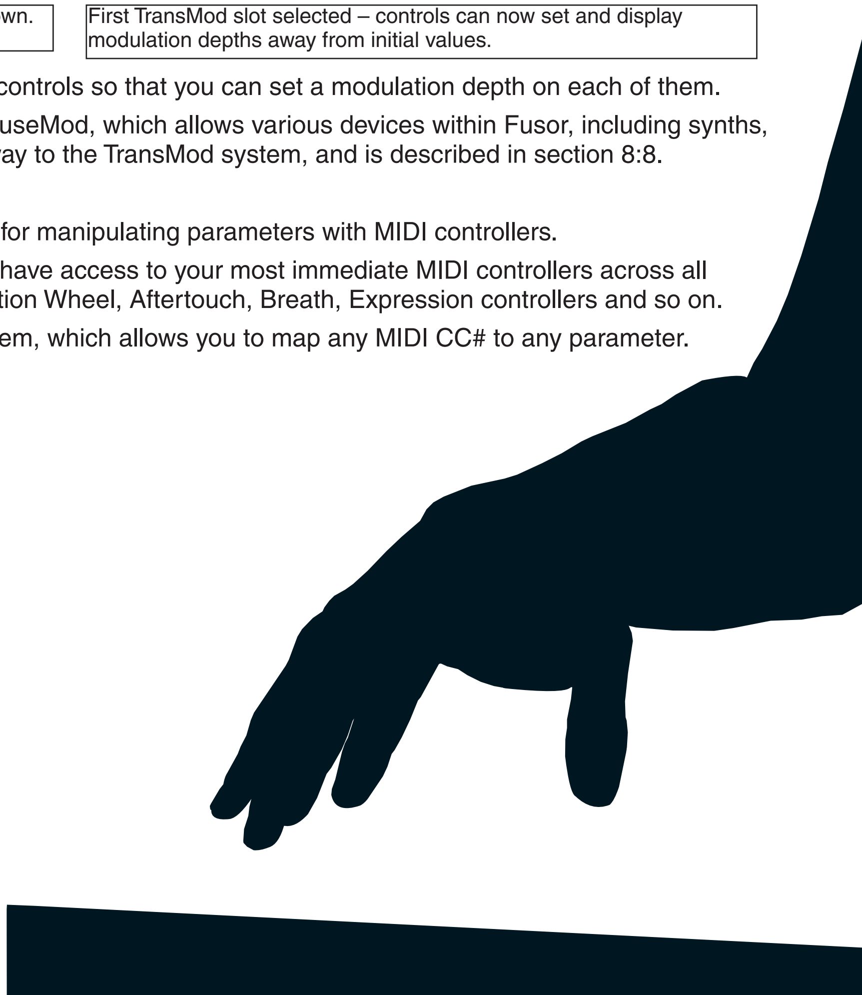
Fusor includes a slight variant on the system called FuseMod, which allows various devices within Fusor, including synths, to modulate each other. It operates in a very similar way to the TransMod system, and is described in section 8:8.

Performance controllers and MIDI Learn

DCAM: Synth Squad features a standardized system for manipulating parameters with MIDI controllers.

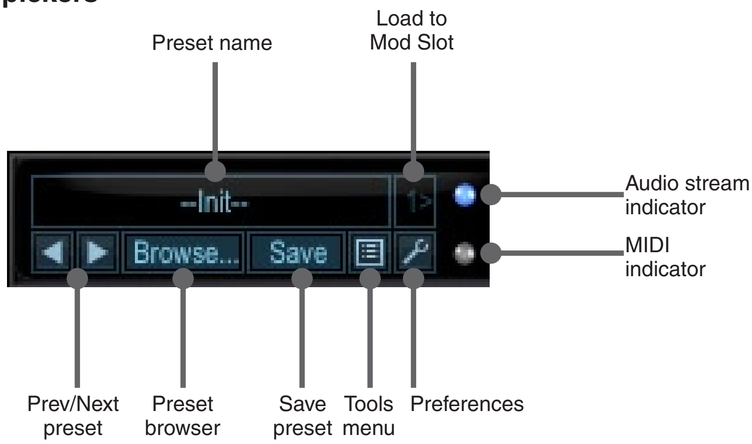
Performance controllers are designed to allow you to have access to your most immediate MIDI controllers across all presets. Typical performance controls include Modulation Wheel, Aftertouch, Breath, Expression controllers and so on.

DCAM: Synth Squad also provides a MIDI Learn system, which allows you to map any MIDI CC# to any parameter.



1:3 Interface basics

Loading presets: preset pickers



The instruments in DCAM: Synth Squad have a standardized preset selection system. There are 3 ways of browsing through presets:

- **Drop-down preset menu**

  Click on the preset name to display a simple drop-down list of available presets, arranged by styles.

  In order to access all available presets, hover the mouse over the red up/down arrows at the top and bottom of each sub-menu. The contents of the sub-menu scroll up/down to reveal more presets if they are present.

- **Prev/Next preset buttons**

  These buttons provide a quick way of flicking through presets sequentially.

- **Preset browser**

  Click on the **Browse...** button in the preset picker to launch the preset browser, which lets you browse through presets 'in-context' – see below for more details.

Clicking on the preset name displays a drop-down menu of all available presets. Click on any preset in the menu to load it.

Other Preset picker controls

Save preset

This button launches a system Save As... dialog, directed at the user preset location, It is recommended to save your presets into this folder so that they are easily accessible within the preset-loading functions!

Saving a preset with the relevant 2-letter prefix (for example, BA for basses) results in it appearing in the respective category sub-menu.

The user preset location is:

<user location>/FXpansion/<synth name>/Presets

Tools menu

The Tools menu contains items for launching the preset browser and Save As... dialog. It also features **Cut**, **Copy** and **Paste** functions, which can be useful for transferring the current preset to other instances of the synth that are currently open in your host.

Preferences

Click this button to open the Preferences panel, which is discussed in section 6:5.

Load to Mod Slot

The synth preset pickers contain the **Load to Mod Slot** button, which offers an advanced function for use with the TransMod system. See section 7:3 for more details.

Audio stream indicator

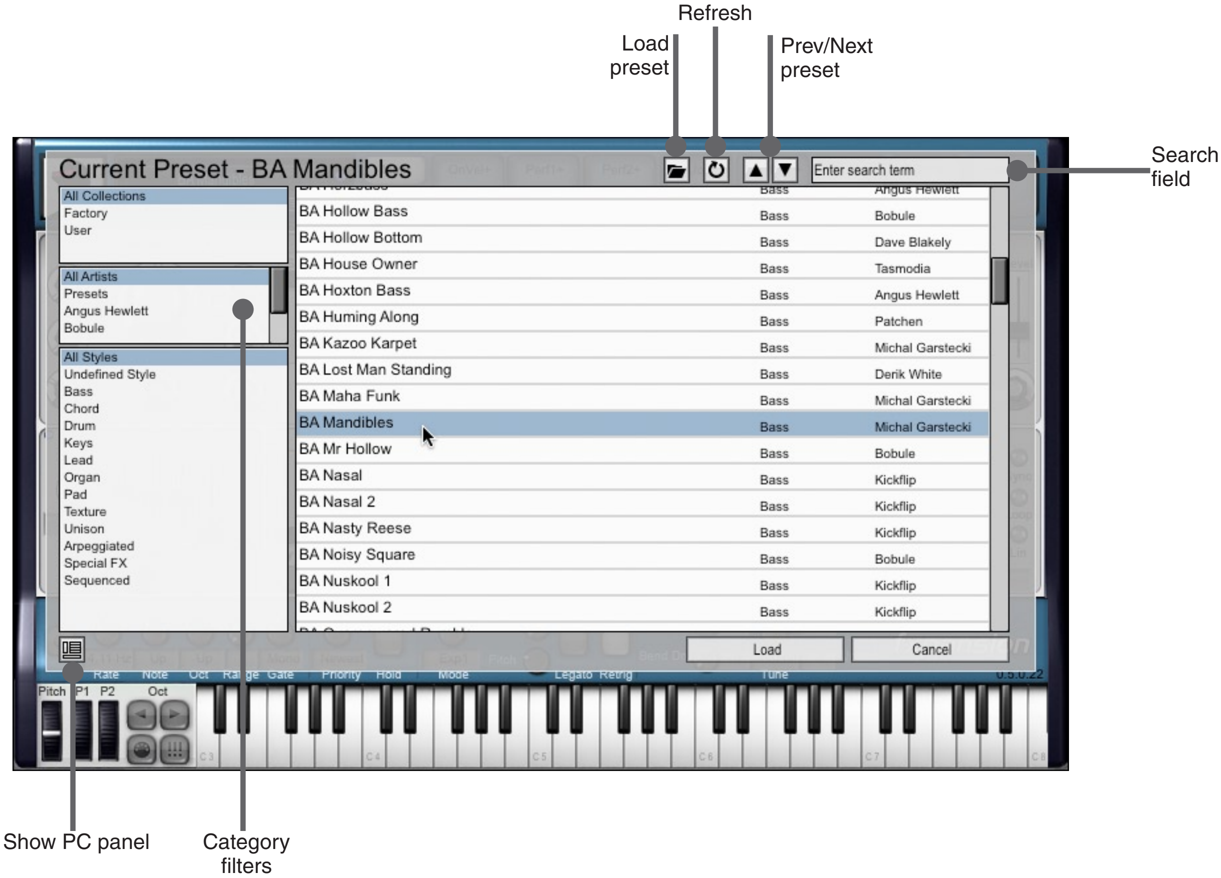
This indicator lights up if the synth receives an audio stream. The indicator should be lit if the synth is operating properly in your host's audio engine, or within the standalone application if a valid audio interface is selected.

MIDI indicator

This indicator lights up when any MIDI event input is received.

Preset browser

The preset browser allows you to browse through presets ‘in-context’.



1. Click on the **Browse...** button in the preset picker in order to display the preset browser.
2. Click on a preset in the list.
3. Play some MIDI notes into the synth – you’ll hear the selected preset.
4. If you click the **Load** button, the preset will be loaded into the synth.
5. If you click the **Cancel** button, the synth reverts back to its state before you opened the preset browser.

Presets can be selected either by clicking them in the listing, or by using the **Prev**/**Next** buttons.

The **Load Preset** button allows you to load a preset from any location on your system (rather than merely within the synth plugins’ own preset folders), while the **Refresh** button re-reads the contents of the preset folder.

Searching and filtering

The preset browser contains additional functions to filter and search the available presets.

Click in the **Search field** at the top-right of the preset browser, type a search term and press ENTER in order to search the name, author and style fields and display matching results.

The 3 **category filters** allow you to narrow down the displayed presets by the following criteria:

Collections	Filters by preset collection. By default, Factory and User collections exist. You can also compile new collections using the PC panel (see below).
Preset author	Filters by the name of the preset designer.
Preset style	Filters by preset style (Bass, Chord, Drum, Keys, Lead, Organ, Pad, Texture, Unison, Arpeggiated, Special FX or Sequenced).

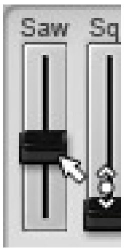
All 3 filters and the search facility can be used at the same time.

The preset browser contains the program change (PC) panel, which is shown by clicking the **Show PC Panel** button. This function is described in section 6:9.

Controlling parameters: initial values (Main view)

When the ‘Main’ view is selected, only the initial value of a control can be adjusted.

Main view

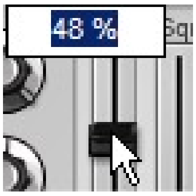


Main view:
Sliders

The initial value of a control is adjusted by clicking it and dragging up/down.

Main view:
Rotary pots

The initial value of a control is adjusted by clicking it and dragging up/down.



You can also double-click the control, type a value and press ENTER or RETURN.

Fine control over parameters

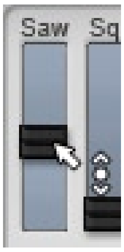
Hold down the SHIFT key while adjusting a control for finer resolution.

Controlling parameters: TransMod modulation depths

Each synth features 8 TransMod slots, which allow modulation depths to be set on almost all synth parameters.

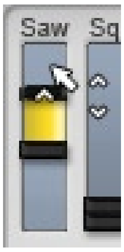
When a TransMod slot is selected (see below), the synth’s controls can be adjusted in the following ways:

Sliders



TransMod slot selected:
Setting initial value

Click the slider cap and drag up/down, or double-click to type a value and press ENTER or RETURN.



TransMod slot selected:
Setting modulation depth

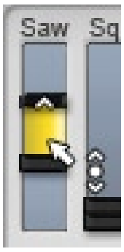
Click outside the slider cap and drag up/down.

The slider cap divides and the modulation depth from the initial value is set.

You can also move the part of the slider cap with the arrow to adjust modulation depth.

You can also double-click outside the slider cap, type a value and press ENTER or RETURN.

- When modulation already exists, 2 additional ways of manipulation are possible:

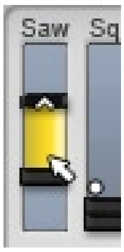


TransMod slot selected:
Setting initial value + depth

Drag the initial part of the slider cap (without the arrow):

The initial value and the modulation depth are adjusted together.

You can also double-click to type a value.



TransMod slot selected:
Setting initial value only

Hold down ALT and click/drag the initial slider cap (without the arrow):

Only the initial value is set – the modulation depth is unaffected.

Rotary pots

TransMod slot selected:
Setting initial value

Click the centre of the rotary pot and drag up/down.

TransMod slot selected:
Setting modulation depth

Click and drag the outer ring around the rotary pot to set the modulation depth from the initial value.

You can also double-click in the outer ring, type a value and press ENTER or RETURN.

- When modulation already exists, 2 additional ways of manipulation are possible:

TransMod slot selected:
Setting initial value + depth

Click and drag the centre of the rotary pot to adjust the initial value and modulation depth together.

You can also double-click to type a value.

TransMod slot selected:
Setting initial value only

Hold down ALT and click/drag the centre of the pot:

Only the initial value is set – the modulation depth is unaffected.

TransMod slots

When you first launch a DCAM synth, it displays its *Main view*.

The Main view only edits the 'basic state' of the synth – the initial values of all controls before any TransMod modulation.

If you click on any of the 8 TransMod slots' selection area (where it says 'via'), or use the **Prev/Next slot** buttons, the synth's interface changes so that most of its controls are able to set and display modulation depths away from the initial value.

In this example, Strobe's first TransMod slot (assigned to key-on velocity) contains modulation for the osc **Saw** waveform level, filter **Cutoff** and amp **Level** parameters.

See chapter 7 for full details of the TransMod system.

Rotary selectors and drop-down menus

• Rotary selector click & drag (left):

Click the rotary selector and drag up/down.

• Rotary selector drop-down menu (below):

Click on the display that shows the setting and select the relevant setting from the drop-down menu that appears.

• Fusor contains some rotary selectors that are essentially pots (such as **Steps** in the screenshot to the right), some which have associated drop-down menus (such as **Step Duration**) and some which are simply drop-down menus (such as the **Multiplier** controls).

Drop-down menus can also be found in the synths – for example, TransMod slot source/scaler menus, performance controller selectors in the preferences panel and the preset menu (click on the preset name).

Numerical text-boxes

There are two ways to adjust these controls:

• Click & drag:

Click the value and drag it up/down.

• Double-click & type:

Double-click the value, type a new one and press ENTER/RETURN.

Buttons

Buttons are generally 'toggle' type buttons – click to activate, click again to deactivate. Buttons light up when activated. Some 'radio' button-style controls exist, such as the sub-oscillator **Octave** buttons in Strobe.

Indicator LEDs

DCAM: Synth Squad instruments contain indicator LEDs which light up to represent LFO rates, audio-rate modulation and so on. Do not confuse these with buttons, which are larger.

Parameter context menu

Right-click (or CTRL-click) on any control to show its parameter context menu, which offers a number of functions and options relating to the control.

Reset Param

Resets the control's initial value to its default setting.

Clear Param Mod

Clears any modulation depths that exist for the control in the *current* TransMod slot.

Clear Param All Mod

Clears any modulation depths that exist for the control in *all* TransMod slots.

Lock Scope
Unlock Scope

You can lock the visualizer scope to the section of the synth that contains the current parameter (see section 6:6).

Clear Learn

Clears any MIDI Learn assignment that exists for the parameter (see section 6:8).

Snapping/unit options for tuning and filter cutoff controls

Any controls related to audio pitch or frequency (usually oscillator pitch and filter cutoff controls) offer 3 distinct modes of operation, which are accessed via the parameter context menu.

The **Just** and **Harmonic** modes are based on perfect pitch ratios, rather than absolute frequency settings. This is particularly useful for programming musically harmonic frequencies when using Cypher's audio-rate modulation functions. Using these modes means you you don't have to calculate such frequencies manually.

These modes persist independently – all tuning controls within each synth possess their own mode, and all filter controls possess their own mode.

Just

Uses perfect pitch ratios rather than imperfect, equal-tempered pitch.

In this mode, pitch or filter cutoff controls are set in harmonics (Hm).

There is no snapping to whole harmonics.

Harmonic

Similar to **Just**, except that the control snaps to whole harmonics.

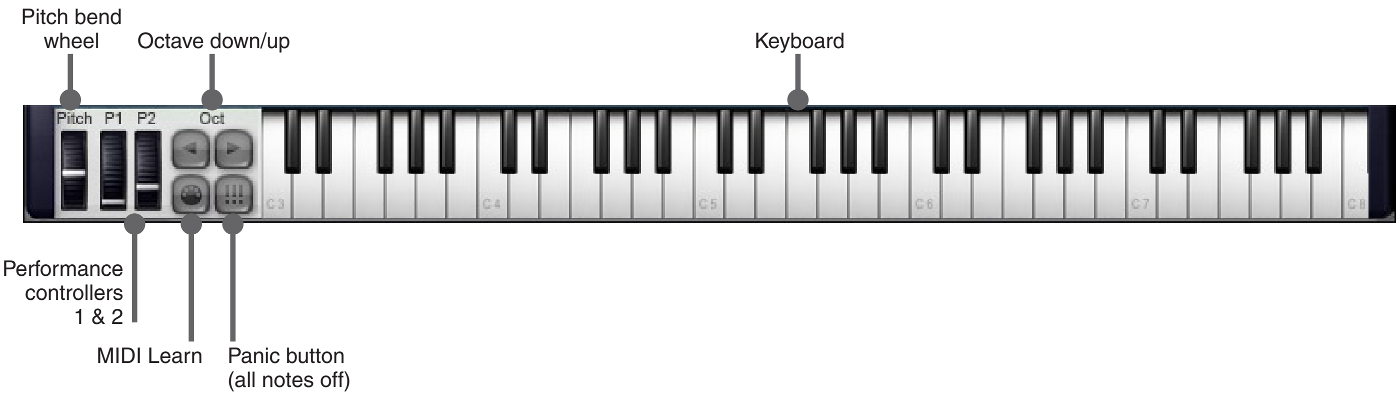
Equal

Represents equal-tempered tuning in semitones, snapping to whole semitones.

Off

Turns off all snapping/unit options: the control is set in semitones but does not snap to whole semitones.

On-screen keyboard and performance controllers



This area of the interface mainly contains controls that are representations of hardware input – the keyboard, pitch-bend wheel and performance controllers 1 and 2. While these are intended to be used with MIDI hardware, the interface elements can nevertheless be controlled with the mouse. Use the **Octave down/up buttons** to access the full range of MIDI notes on the keyboard.

There are a couple of other MIDI-related functions in this area:

MIDI Learn button

Engaging this button enters MIDI Learn mode. This function is discussed in section 6:8.

Panic button

Clicking this button results in a MIDI reset – all notes are turned off. Use this to stop any notes that are hung or have a very long amp envelope release time.

2: Strobe

2:1 Overview

Strobe is a performance synthesizer, designed to be easy to program so that you can get on with playing great-sounding music! A single-oscillator synth with osc-stacking, sub-oscs and osc sync, it also features a versatile multimode filter, LFO, ramp generator, two envelopes and an arpeggiator.

While being inspired by relatively simple monosynths such as the Roland SH series (especially the SH-09 and SH-101), Oberheim OB-1 and SCI Pro-One, Strobe is not an exact model of any particular synth. It has been designed to take the performance synthesizer into a new dimension.

Strobe highlights

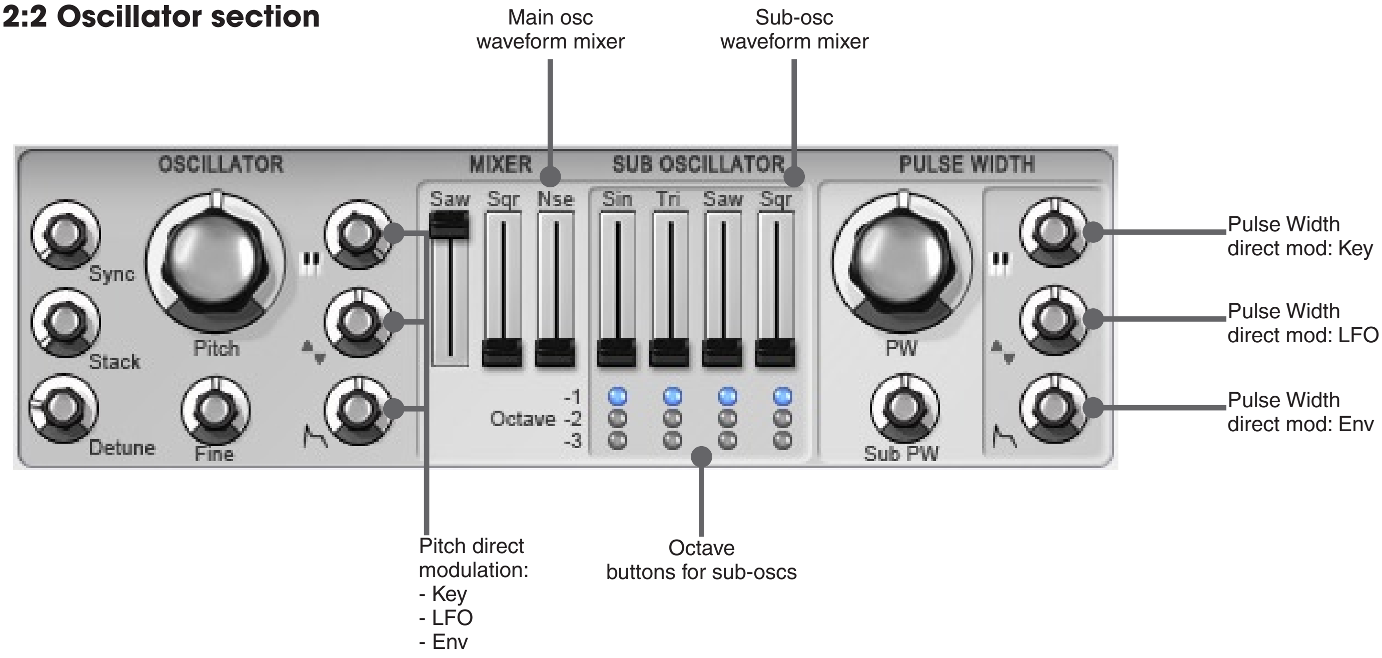
Unlike the monosynths which inspired its design, Strobe is capable of polyphony - you can set up as many voices as your CPU can handle, and define a number of unison voices. For example, with 8 voices and 2 unison voices, the result is 4-note polyphony with each note comprising 2 unison voices.

Strobe is designed for powerful analogue-style bass, lead and pad sounds. The filter section features a large variety of modes, leading to wide range of potential timbres. The detailed filter model includes realistic overdrive of the circuit, further increasing the tonal range. The osc-stacking allows many possibilities that are normally only possible when using multiple oscillators or unison voices.

While the synth's architecture is quite simple, with certain 'obvious' hard-wired modulation routings already available via dedicated depth controls, the TransMod modulation system provides a huge selection of monophonic and polyphonic modulation sources, which can be routed to virtually any synthesis parameter. Strobe's power truly comes alive when using imaginitive modulation, which greatly expands its sound and performance potential.

Try also stacking 2 or 3 instances of Strobe in Fusor, which allows you to inter-connect modulation sources between instances.

2:2 Oscillator section



Oscillator controls

Tuning: Pitch & Fine

These are the main tuning controls for the oscillator. The **Pitch** can be modulated from the keyboard, LFO and Mod Envelope using the dedicated controls.

Sync

The oscillator features a 'hard sync' function: you can increase the frequency of the oscillator with the **Sync** control, but it is re-synced on each cycle of the lowest sub-oscillator.

The resulting waveform has the same overall pitch as the original osc but with added harmonics that create more complex timbres.

Stack & Detune

These controls comprise the oscillator-stack function, which provides classic multi-osc sounds using a single oscillator, without needing to use additional unison voices.

The **Stack** control sets the number of stacked oscillators, while the **Detune** control detunes them up to an octave apart from the main oscillator pitch. For a 'detuned supersaw' sound, set the Detune control to around 5%, and then increase the Stack control to 2 or 3 (or any number up to 5!).

Waveform mixing controls

Main osc and sub-osc mixer

The main osc mixer contains level sliders for **Saw**, **Square (Sqr)** and **Noise (Nse)** (gaussian white) waveforms.

The sub-osc mixer features level siders for **Sine (Sin)**, **Triangle (Tri)**, **Saw** and **Square (Sqr)** sub-oscillator waveforms. Each of these can be set 1, 2 or 3 octaves below the main osc pitch using the **Octave** buttons.

The level sliders allow you to freely mix the levels of all these parallel waveforms.

The Square waveforms, for both the main osc and sub-osc, are in fact pulse waveforms. Their pulse width can be manipulated with the controls in the Pulse Width section of the oscillator.

When using the StrobeFX plugin (MIDI-controlled audio effect plugin version), the external audio input appears in place of the **Nse** source.

Pulse Width controls

PW & Sub PW

The Square waveforms in Strobe are actually variable-width Pulse waveforms.

The Pulse Width section allows you to control the pulse width of the Square main oscillator and sub-oscillator shapes using the **PW** and **Sub PW** controls respectively.

The main osc pulse width (PW) can be modulated from the keyboard, LFO and Mod Envelope using the dedicated controls.

Direct modulation

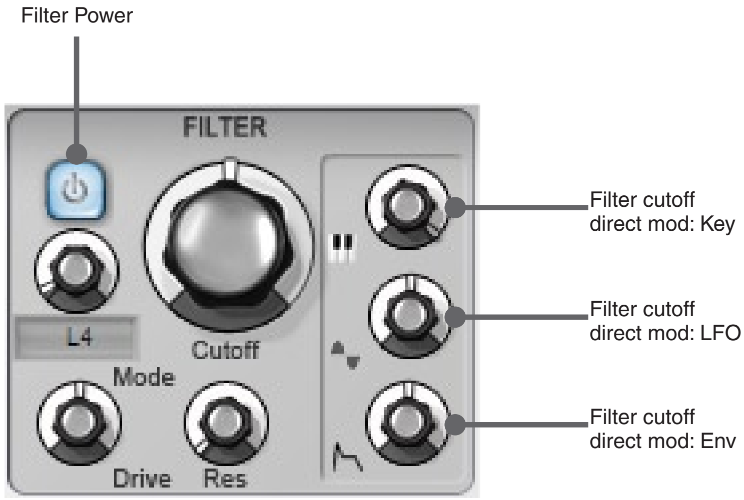
The oscillator section contains dedicated controls for setting the modulation depth of the osc's **Pitch** and the **Pulse Width** of the main Square waveform (the sub-osc pulse width must be modulated via the TransMod system) from 3 modulation sources:

- Keytracking
- LFO
- Mod Envelope

2:3 Filter section

Strobe's filter is its main source of tonal variation, especially when modulated. While there are direct modulation routings to the cutoff from the keyboard pitch, LFO and Mod Envelope, the filter's controls can be modulated with the TransMod system for really creative and varied effects.

Strobe's filter is based on an OTA (operation transconductance amplifier) cascaded core, with a diode clipper in the feedback section. The diodes are slightly mismatched, leading to the characteristic growl of a real analog filter.



Filter controls

Power

The **Power** button switches the filter on or off. When the button is disabled, the audio from the oscillator passes through the filter unaffected.

Cutoff & Res (Resonance)

The **Cutoff** control determines the cutoff frequency of Strobe's filter.

The **Res** control adds emphasis around the cutoff frequency, giving a warm, resonant sound to the osc tone. At extreme Res settings, the filter self-oscillates.

The cutoff can be modulated from the keyboard, LFO and Mod Envelope using the dedicated controls.

The control is adjusted in semitones, allowing you to tune the self-oscillating filter just like an oscillator. Setting the keytracking depth control to maximum allows you to 'play' the filter musically from the keyboard.

Drive

The **Drive** control increases the gain of the signal going into the filter. This overloads the filter's components, drastically changing its sound and character.

Note that the effective resonance is reduced as you turn up the Drive control.

Mode

A large variety of filter modes are available by adjusting the **Mode** control.

Note that filters denoted as 'T' feature 2 or more filters with an octave between the frequency of each. The first filter in the name is at the cutoff frequency, with subsequent filters each an octave higher than the last. The only exceptions are the last 2 filter modes - see the table below. Graphical plots of the filter responses can be found in Appendix 2.

In other combination filters, all filters act at the cutoff frequency.

L2, L4, B2, B4, H2, H4	2-pole (12 dB/oct.) & 4-pole (24 dB/oct.) Low-pass, Band-pass & High-pass filters
P2	2-pole Peak filter
NP2	1-pole Notch and 2-pole Peak filters
N2P2 T	2-pole Notch and 2-pole Peak filters, an octave apart
N2L2 T	2-pole Notch and 2-pole Low-pass filters, an octave apart
N2H1L1 T	2-pole Notch, 1-pole High-pass and 1-pole Low-pass filters, each an octave apart
P2N2 T	2-pole Peak and 2-pole Notch filters, an octave apart
H2N2 T	2-pole High-pass and 2-pole Notch filters, an octave apart
H1L1N2 T	1-pole High-pass, 1-pole Low-pass and 2-pole Notch filters, each an octave apart
N2P2N2 T	2-pole Notch, 2-pole Peak and 2-pole Notch filters, each an octave apart
N2	2-pole notch filter
N4	4-pole notch filter
N2L2	2-pole Notch and 2-pole Low-pass filters
H2N2	2-pole High-pass and 2-pole Notch filters
H1N2L1	1-pole High-pass, 2-pole Notch and 1-pole Low-pass, each an octave apart
N2N2A T	2-pole Notch with 2-pole Notch an octave higher than the cutoff frequency
N2N2B T	2-pole Notch with 2-pole Notch an octave lower than the cutoff frequency

Direct modulation

The filter section contains dedicated controls for setting modulation depth of the filter **Cutoff** by 3 modulation sources:

- Keytracking
- LFO
- Mod Envelope

2:4 Amp section

Strobe's amp section can be overloaded like that of a real analogue synth VCA (voltage-controlled amplifier). There is a waveshaping stage between the **Amp** and final **Level** controls, which is capable of distortion when its components are driven hard.

If you want a cleaner sound, keep the Amp control at low settings and increase the Level.

If you want to overload the amp by increasing the Amp parameter, remember to turn down the Level control. Otherwise, the output of Strobe may clip.

Amp

The **Amp** parameter is directly modulated by the Amp Envelope. Its value represents the amplitude at the maximum value of the AmpEnv (at the end of its attack stage).

In order to achieve velocity-sensitivity for amplitude, modulate the Amp parameter with the OnVel+ source (found in TransMod slot 1 in all DCAM: Synth Squad presets).

Pan

The **Pan** parameter sets the position of the voice in the stereo field.

Try modulating this control with the TransMod system, using a Voice modulation source – this leads to a rich stereo spread of the voices when playing chords.

Level

This parameter sets the final **Level** of the voice before it is summed with all other active voices.

Analogue

The **Analogue** parameter simulates the effect of noise and mains hum in certain parts of the audio and control signal paths, something that always occurs in real analogue synth circuits. At lower settings, it leads to a subtly gritty and slurring character, while higher settings create a more unstable and noisy sound.

2:5 Other functions

Built-in Arpeggiator

The arpeggiator is common to both Strobe and Cypher. See section 6:1 for details on using the arpeggiator.

Settings, Glide and Keying controls

These parameters are common to all synths, and are covered in sections 6:2, 6:3 and 6:4.

Visualizer Scope

The visualizer scope provides context-sensitive graphical feedback for each section of Strobe. See section 6:6 for more details on this feature.

2:6 Modulation

Strobe contains several modulators that feature direct routings and depth controls for certain destinations. These sources can also be used in the TransMod system (see chapter 7) to modulate other parameters and perform more complex modulation. There are many other TransMod sources available beyond the modulators shown on the interface.

Gateable modulators

Gateable modulators (which can be gated with note-on/-off or triggered with note-on) feature a number of different gating/ triggering modes, including the ability to gate or trigger each other. For more details, see section 7:5.

LFO

Direct routings: Osc Pitch, Pulse Width, Filter Cutoff

Gateable by: Poly, PolyOn, Mono, Ramp, ModEnv, Song

Ramp

Direct routings: none

Gateable by: Poly, PolyOn, Mono, ModEnv, LFO, Song

Mod Envelope

Direct routings: Osc Pitch, Pulse Width, Filter Cutoff

Gateable by: Poly, PolyOn, Mono, Ramp, LFO, Song

Amp Envelope

Direct routings: VCA Amp

Gateable by: Poly, PolyOn, ModEnv, Ramp, LFO, Song

Glide

Strobe is capable of smoothing both velocity and pitch control signals.

For further details on the **Pitch Glide** and **Vel Glide** functions, see sections 6:3 and 7:6.

Keytracking (modulation from keyboard pitch)

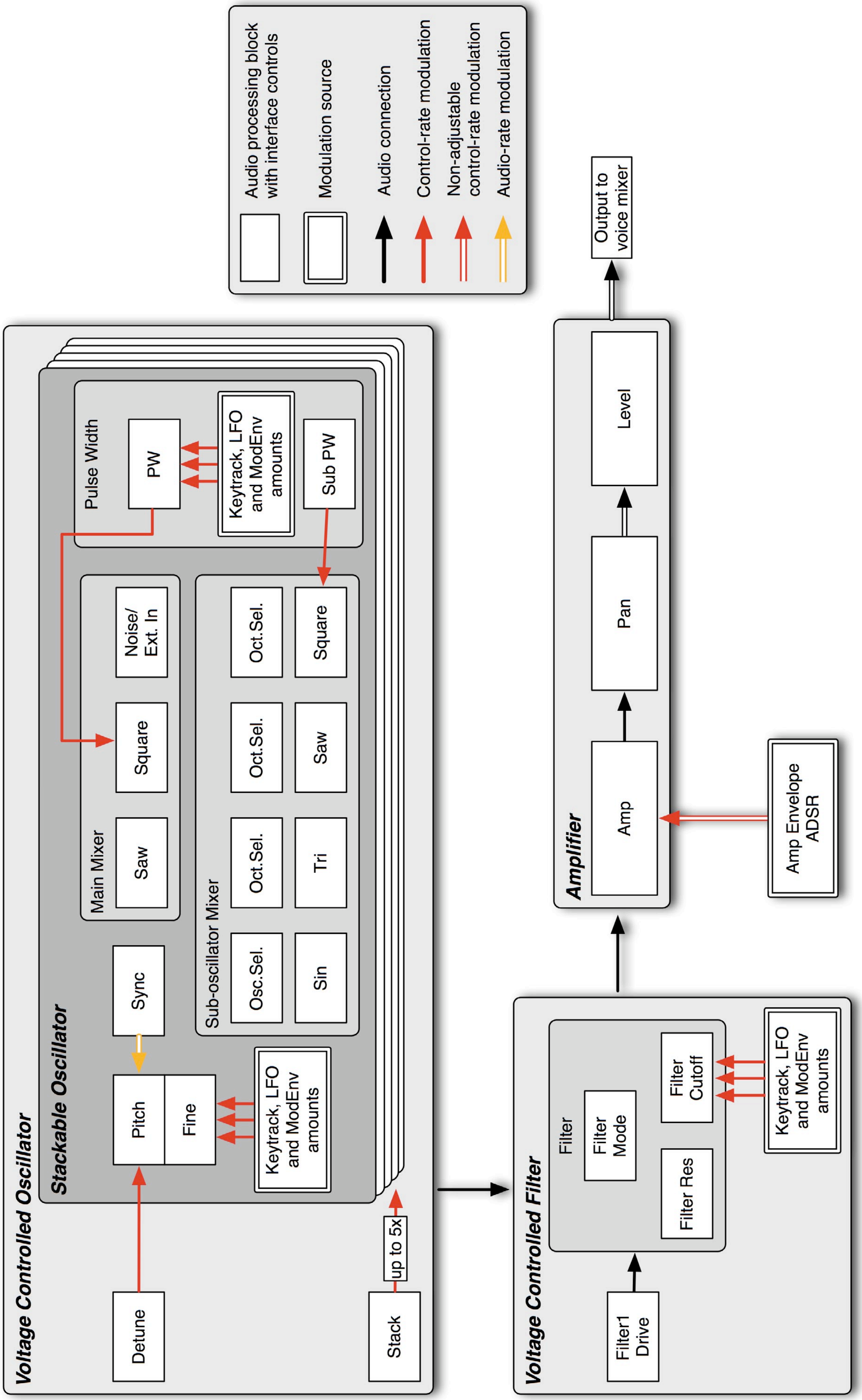
Direct routings: Osc Pitch, Pulse Width, Filter Cutoff

Keytracking is modulation from the pitch of the keyboard. The pitch modulation source value increases by 1 over each octave. As well as the direct routings and depth controls specified above, keytracking can also be achieved for any other parameter by using the Pitch source in the TransMod system.

See section 7:7 for more details about keytracking.

2:7 Strobe signal flow

The following signal flow diagram represents a single voice of Strobe. Only direct modulation is shown – there are many further possibilities when using the TransMod system.



3: Cypher

3:1 Overview

Cypher features a dual-filter/waveshaper architecture and a highly complex, versatile set of 3 modelled oscillators. While it can be used as a 3-osc analogue-style subtractive synth, it has been designed to be used primarily for FM, audio-rate modulation and other advanced oscillator functions. Using Strobe is recommended for most 'conventional' subtractive sounds, as it is optimized for quick and easy programming.

Also included in Cypher is an arpeggiator identical to that found in Strobe, and other common DCAM: Synth Squad functions such as the TransMod system for advanced modulation.

Background

The ethos behind Cypher is to provide a truly accurate model of an analogue synth capable of analogue-style FM and other audio-rate modulation (as opposed to multi-operator FM such as that found in Yamaha DX-series digital synths). It features thru-zero FM which allows musical digital FM-style sounds with an analogue-style circuit.

Some examples of analogue synths that feature FM are the SCI Prophet 5 and Roland Jupiter series, which allowed one osc to modulate another. Most modular synths also allow you to patch any audio-rate signal to vary the osc frequency.

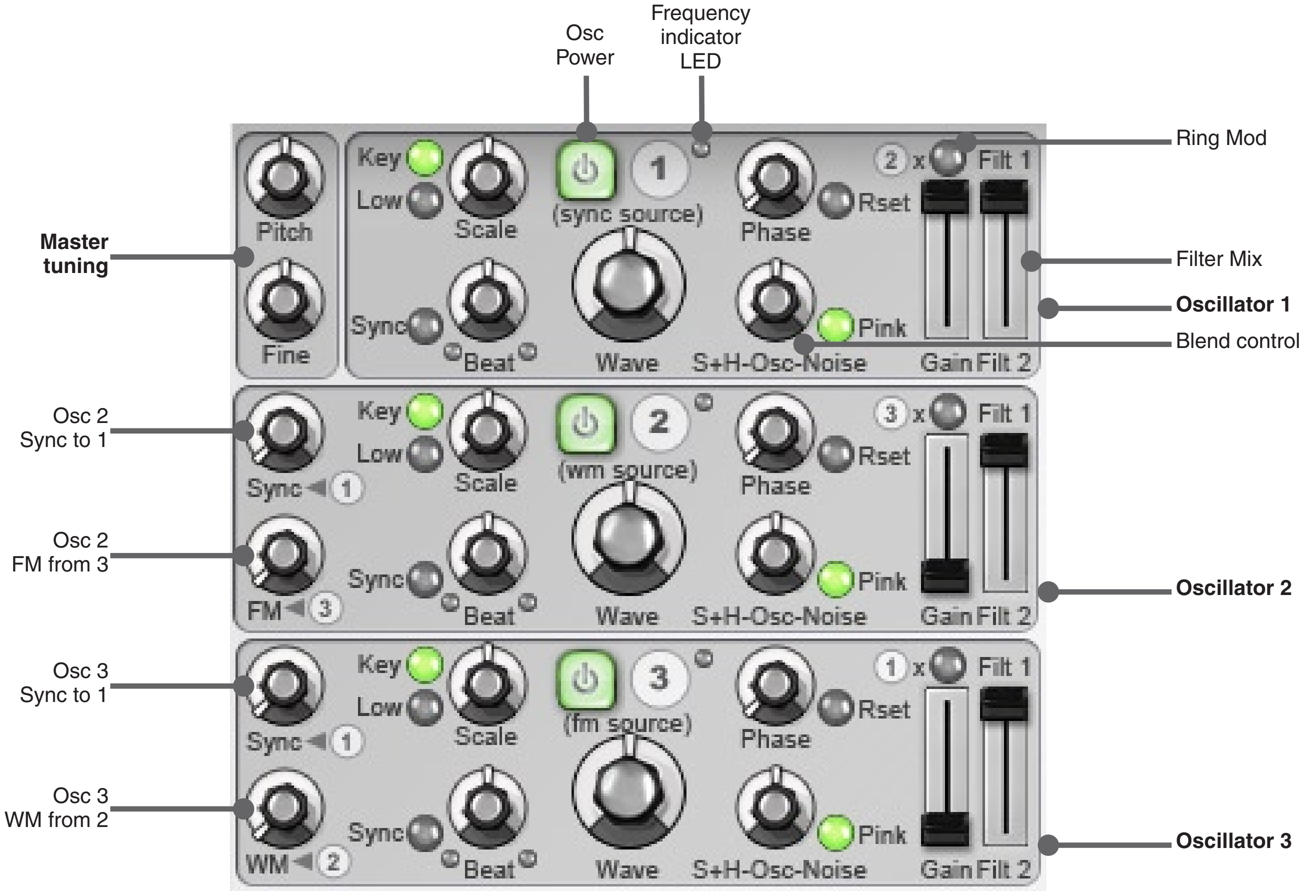
Programming with Cypher

Cypher requires a slightly different approach to a more conventional analogue-style synth. Chapter 4 features a detailed discussion of the specialized functions of Cypher's oscillators, which are the main focus of timbral variety in the synth.

Each osc is tuned using a **Scale** control in relation to a variable master pitch. Since any pitch-related control in DCAM: Synth Squad features several snapping modes including 'harmonic', you can tune each osc with harmonic ratio multipliers instead of absolute frequencies. This means you don't have to calculate harmonic frequencies yourself, making musically useful FM sounds much easier to achieve than with traditional analogue oscillators. See section 1:3 for more details of the snapping functions.

3:2 Oscillator section

This section is a brief overview of Cypher's oscillator functions. Please read chapter 4 to gain a full understanding of the oscs' special capabilities.



Pitch, Fine & Scale

The pitch of each of the 3 oscillators is scaled against the master tuning (set using the **Pitch** and **Fine** controls), using their respective **Scale** parameters.

By setting the **Harmonic** mode on a pitch control's parameter context menu, each osc's Scale control can operate in harmonic pitch ratios instead of semitones, useful when dealing with FM and other audio-rate modulation processes.

Power

Each osc features a **Power** button to enable it. With the button disabled, the osc is deactivated.

Wave

The shape of each oscillator can be varied continuously between Triangle, Saw, Square and Pulse waveforms, using the **Wave** control.

Due to their continuously variable nature, Cypher's oscs do not feature a dedicated pulse width control – in order to vary pulse width, you must modulate the Wave parameter in the square/pulse region of its travel.

The waveform of Osc3 can be varied at audio rate by Osc2 using the **WM from 2** control.

FM from 3 & WM from 2

Audio-rate modulation refers to modulating a parameter at the very fast speeds of audio waveforms, as opposed to the slower speeds of LFOs and envelopes (known as control-rate modulation). The audio-rate modulation in Cypher is the result of extremely detailed modelling, and strives to reproduce the complex behaviour that results when modulating analogue components at audio-rate. Here's a summary of what Cypher can do in this respect:

- Audio-rate Frequency modulation of Osc2 by Osc3 using the **FM from 3** control on Osc2 (see sections 4:1 and 4:2)
- Audio-rate wave modulation of Osc3 by Osc2 using the **WM from 2** control on Osc3 (see section 4:3)

Sync to 1

The **Sync to 1** controls in the Osc2 and Osc3 sections allow you to sync these oscs to Osc1's frequency. Rather than a simple on/off button for this function, Cypher features a continuous control that provides full 'hard' sync at 100% and variable degrees of sync between 0 and 100%. This feature is covered in detail in section 4:5.

Beat

The **Beat** control allows you to detune one oscillator against another while keeping the rate of the resulting 'beating' constant across the keyboard. This opens up new rhythmic ways of playing sounds with classic detuned-beating characteristics. See section 4:6 for a detailed discussion of how the Beat control works.

This control also sets the osc's frequency (or rate) while the Osc is in LFO mode (when the **Low** button is enabled).

Phase

Each osc's phase can be set with the **Phase** parameter.

With the **Rset** (Reset) button enabled, the phase is reset to the value defined by the Phase parameter on each envelope retrigger event (which is dependent on the Retrig button).

Disabling the Reset button means that the phase is effectively free-running.

The Phase control can be continuously modulated in real time for phase modulation effects (at control-rate rather than audio-rate).

Low

The **Low** button on each osc switches it to act as an LFO, with the **Beat** parameter controlling its rate in Hz. Each osc is available as a control-rate TransMod source for use with this function (see section 7:8).

Key

The **Key** button turns keytracking (modulation of pitch by the keyboard) on or off for each osc.

Blend (S+H - Osc - Noise) & Pink

The **Blend** control crossfades the final audio output from each oscillator between the oscillator itself (centre position), white/pink noise (fully right), and another osc sampled and held at audio-rate by the osc's frequency (fully left).

Section 4:4 offers a detailed analysis of the audio-rate sample and hold feature.

When the **Pink** button is engaged, the noise source outputs pink noise. When the button is deactivated, the source is set to white noise.

Note that when using the CypherFX plugin (MIDI-controlled audio effect plugin version), external audio appears in place of the *white* noise source in each osc – the Pink noise source is still available for each osc. Therefore, to hear the external audio source, you must disable the Pink button.

Ring mod

Each oscillator features a ring modulation function, allowing it to be multiplied by one of the other oscillators. Enabling the **Ring Mod** button for an osc results in the sum and difference of the oscs' frequencies being used as the output of the osc. If the oscs' frequencies are harmonically related, the result is a musical sound; if not, the output may be dissonant. The ring modulation process occurs after the **Blend** control in the signal path.

Osc1: Osc2 x Osc1
Osc2: Osc3 x Osc2
Osc3: Osc1 x Osc3

Gain

The **Gain** control sets the final output level of each oscillator. At a setting of zero, the osc is not heard in the final output. However, its effect is still heard if it is modulating one of the other oscs.

Filter Mix

Each Osc's **Filter Mix** control determines the amount of signal from the osc that is routed to each of the two filter/shaper paths. When at the centre position, the osc's signal is sent equally to both filter/shaper paths.

3:3 Shaper section

Each of Cypher's two filter paths features a polyphonic waveshaping block that can be placed before or after the filter.

A waveshaper applies a mathematical function to the incoming waveform in order to alter its harmonic content and introduce distortion. It is useful for overdriving the signal to add abrasive grit and character.

Power

To enable each of the waveshaper blocks, engage its **Power** button.

Post

With its **Post** button turned off, the waveshaper block is placed before the corresponding filter block (for example, shaper 1 → filter 1). By engaging the Post button, the shaper is moved after the filter (filter 1 → shaper 1).

Mode

This control selects from 4 distinct waveshaper models, each providing its own distortion flavour. 'Diode', 'OTA' (operational transconductance amplifier), 'OpAmp' (operational amplifier) and 'HalfRect' (half-rectifier) shapers are available. Graphical plots of these shaper modes are provided in Appendix 3.

Drive

The **Drive** control increases the gain of the signal going into the waveshaper – higher values drive the waveshaper harder, leading to more grit and distortion.

LPF

The **LPF** control applies a 1-pole Low-pass filter after the waveshaper for dialling out high frequencies and harmonics as required.

3:4 Filter section: shared controls

Cypher's dual-filter section features a number of controls for each filter, while others are shared between them. The following controls are shared between both filters:

Cutoff

The **Cutoff** control adjusts the cutoff frequency of both filters relative to each filter's **Scale** control setting. You can consider this control as the 'master' cutoff frequency that adjusts the cutoff of both filters simultaneously.

Env1 & Env2

Mod Envelopes 1 and 2 are directly routed to modulate the **Cutoff** (the cutoff frequencies of both filters simultaneously). The **Env1** and **Env2** controls adjust the depth of modulation from ModEnv 1 and ModEnv 2 respectively.

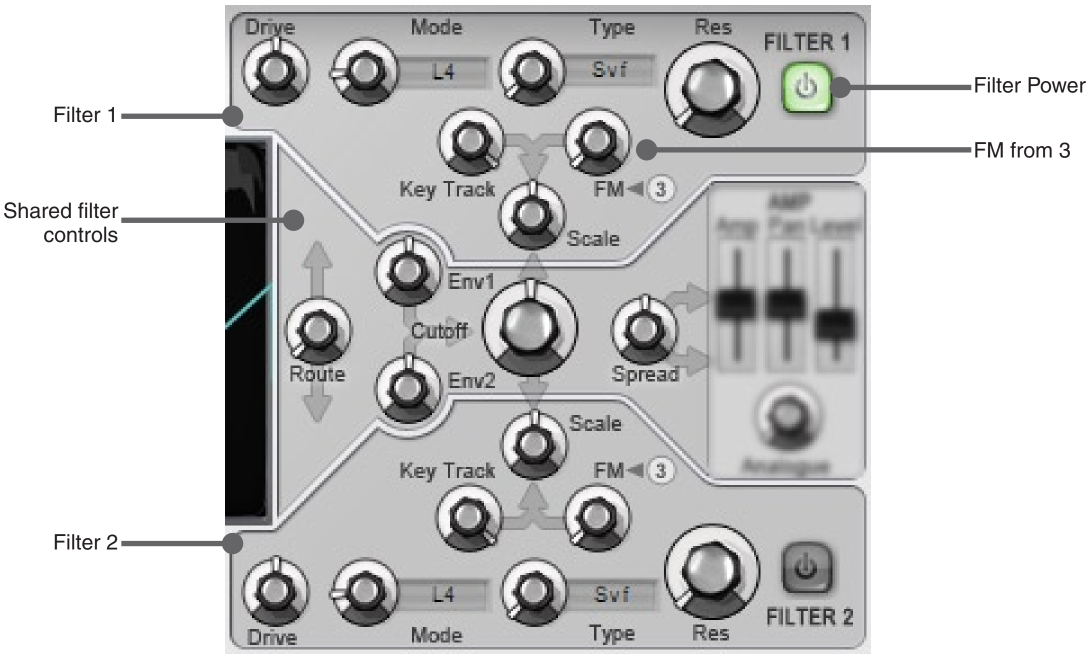
Route

The routing of the filters can be changed between 2 serial and 2 parallel configurations using the **Route** parameter.

1 > 2	Serial, filter 1 before filter 2
2 > 1	Serial, filter 2 before filter 1
1 + 2	Parallel, filter 1 + filter 2
1 - 2	Parallel, inverted filter 1 + filter 2

Spread

The **Spread** control sets the amount of stereo spread between the filters when using parallel routings.



3:5 Filter section: individual controls for filters 1 & 2

In addition to the shared controls that affect the filters in both of Cypher's filter paths, each filter features the following individual controls:

Power

The **Power** button enables or disables the filter. When disabled, incoming audio passes through the filter stage unaffected.

Type

Select from two classic filter designs for either of the two filter paths using the **Type** parameter: state-variable (Svf) and transistor-ladder (Mgf) types are available.

These are both distinct circuit models with very different sonic characteristics.

Mode

The **Mode** control allows you to select from the following filter modes:

- Low-Pass 2-pole or 4-pole
- High-Pass 2-pole or 4-pole
- Band-Pass 2-pole or 4-pole
- Peak 4-pole
- Notch 4-pole

Drive

The **Drive** control increases the gain of the signal entering the filter. This leads to the filter circuit being overloaded, drastically altering its tone and character.

Scale

Each filter's **Scale** control offsets its cutoff frequency relative to the master cutoff by positive or negative amounts. At the centre position, the filter's cutoff is the same as that of the master.

Res (Resonance)

The **Res** control adjusts the resonance of each filter. Higher settings result in self-oscillation.

Keytrack

The **KeyTrack** parameter controls the amount of direct modulation from keyboard pitch, which is applied relative to the filter's **Scale** setting.

FM from 3

The cutoff frequency of each filter can be modulated at audio-rate by the frequency of oscillator 3. The **FM from 3** controls adjust the amount of frequency modulation for each filter.

See section 4:2 for more details of filter FM, and FM in general.

3:6 Amp section

Cypher's amp section can be overloaded like that of a real analogue synth VCA (voltage-controlled amplifier). There is a waveshaping stage between the **Amp** and final **Level** controls, which is capable of distortion when its components are driven hard.

If you want a cleaner sound, keep the Amp control at low settings and increase the Level.

If you want to overload the amp by turning up the Amp paramater, remember to turn down the Level control. Otherwise, the output of Cypher may clip.

Amp

The **Amp** parameter is directly modulated by the Amp Envelope. Its value represents the amplitude at the maximum value of the AmpEnv (at the end of its attack stage).

In order to achieve velocity-sensitivity for amplitude, modulate this parameter with the OnVel+ source (found in TransMod slot 1 in all DCAM: Synth Squad presets).

Pan

The **Pan** parameter sets the position of the voice in the stereo field.

Try modulating this control with the TransMod system, using a Voice modulation source – this leads to a rich stereo spread of the voices when playing chords.

Level

This parameter sets the final **Level** of each voice before it is summed with all other active voices at the final output of the synth.

Analogue

The **Analogue** parameter simulates the effect of noise and mains hum in certain parts of the audio and control signal paths, something that always occurs in real analogue synth circuits. At lower settings, it leads to a subtly gritty and slurring character, while higher settings create a more unstable and noisy sound.

3:7 Other functions

Built-in Arpeggiator

The arpeggiator is common to both Cypher and Strobe. See section 6:1 for details on using the arpeggiator.

Settings, Glide and Keying controls

These parameters are common to all synths, and are covered in sections 6:2, 6:3 and 6:4.

Visualizer scope

The visualizer scope provides context-sensitive graphical feedback for each section of Cypher. See section 6:6 for more details on this feature.

3:8 Modulation in Cypher

This section covers parameter modulation in Cypher (control-rate modulation). For a detailed guide to Cypher's audio-rate modulation capabilities, see chapter 4.

Some of Cypher's modulators feature dedicated routings and depth controls for certain parameters, while others must be utilized within the TransMod system (see chapter 7)

Gateable modulators

Gateable modulators (which can be gated with note-on/-off or triggered with note-on) feature a number of different gating/triggering modes, including the ability to gate or trigger each other. For more details, see section 7:5.

LFO1 & LFO2

Direct routings: none

LFO1 gateable by: Poly, PolyOn, Mono, Ramp, ModEnv1, Song
LFO2 gateable by: Poly, PolyOn, Mono, Ramp, ModEnv2, Song

Ramp

Direct routings: none

Gateable by: Poly, PolyOn, Mono, Ramp, LFO1, Song

ModEnv1 & ModEnv2

Direct routings: Filter Cutoff

LFO1 gateable by: Poly, PolyOn, Mono, Ramp, LFO1, Song
LFO2 gateable by: Poly, PolyOn, Mono, Ramp, LFO2, Song

Amp Envelope

Direct routings: VCA Amp

Gateable by: Poly, PolyOn, Mono, Ramp, LFO1, Song

Glide

Cypher is capable of smoothing both velocity and pitch control signals.

For further details on the **Pitch Glide** and **Vel Glide** functions, see sections 6:3 and 7:6.

Keytracking (modulation from keyboard pitch)

Direct routings: Osc Pitch (on/off), Filter 1 & Filter 2 (variable)

Keytracking is modulation from the pitch of the keyboard. The pitch modulation source value increases by 1 over each octave. As well as the direct routings and depth controls specified above, keytracking can also be achieved by using the Pitch source in the TransMod system.

See section 7:7 for more details about keytracking.

Oscillators as LFOs

Each osc functions as an LFO if its **Low** switch is enabled. Each osc is available as a TransMod source (Osc1±, Osc2± and Osc3±) for modulating any parameter. See section 7:8 for more details.

Beating rates as LFOs

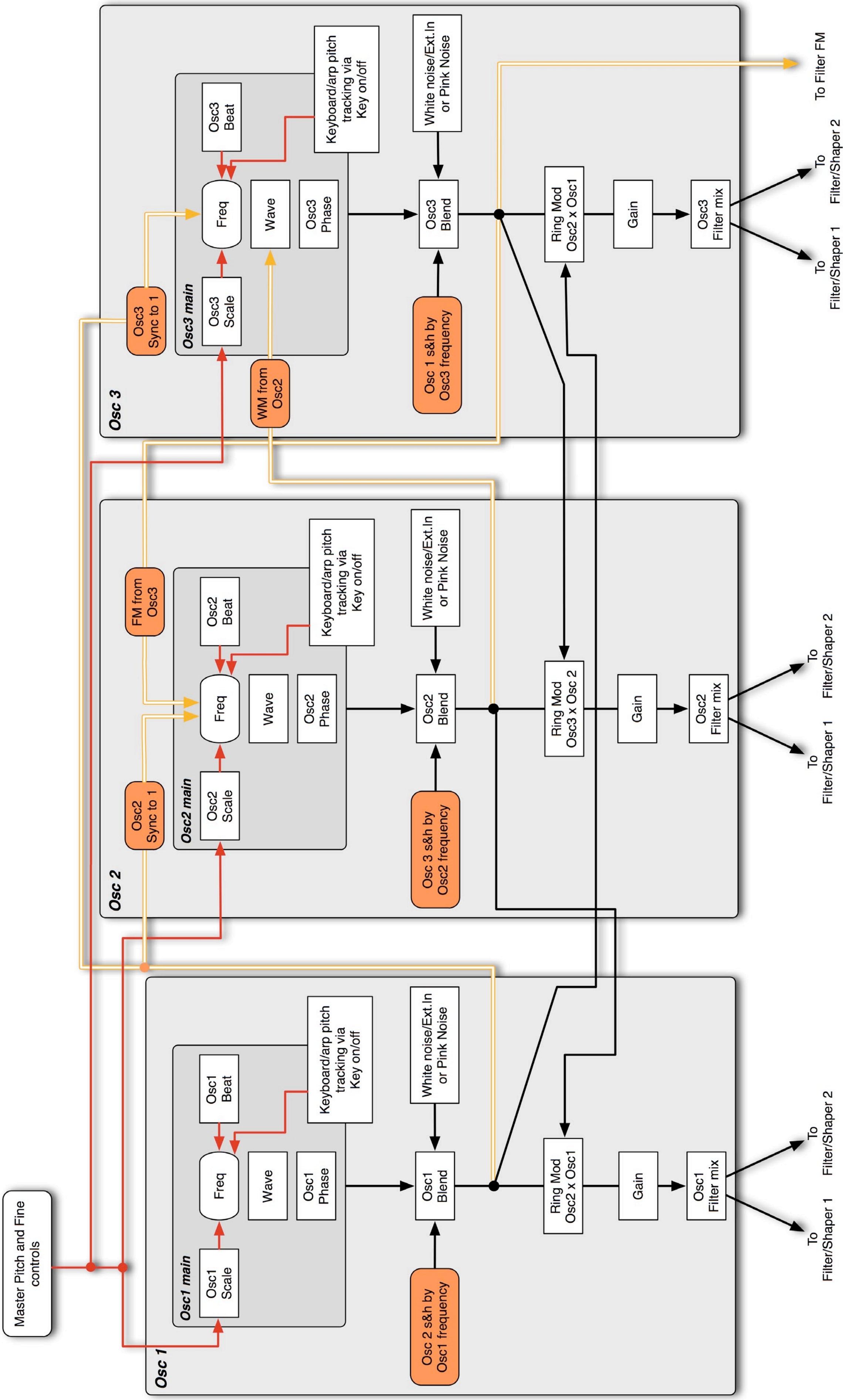
The beating rates between oscillators are available as TransMod sources, which can be used as additional LFOs. These sources operate regardless of the **Beat** control. All beating rates can be utilized in the TransMod system, whether constant beating via the Beat control, or conventional irregular beating when manually detuning oscs against each other.

Section 7:8 contains a summary of the beating rate TransMod sources.

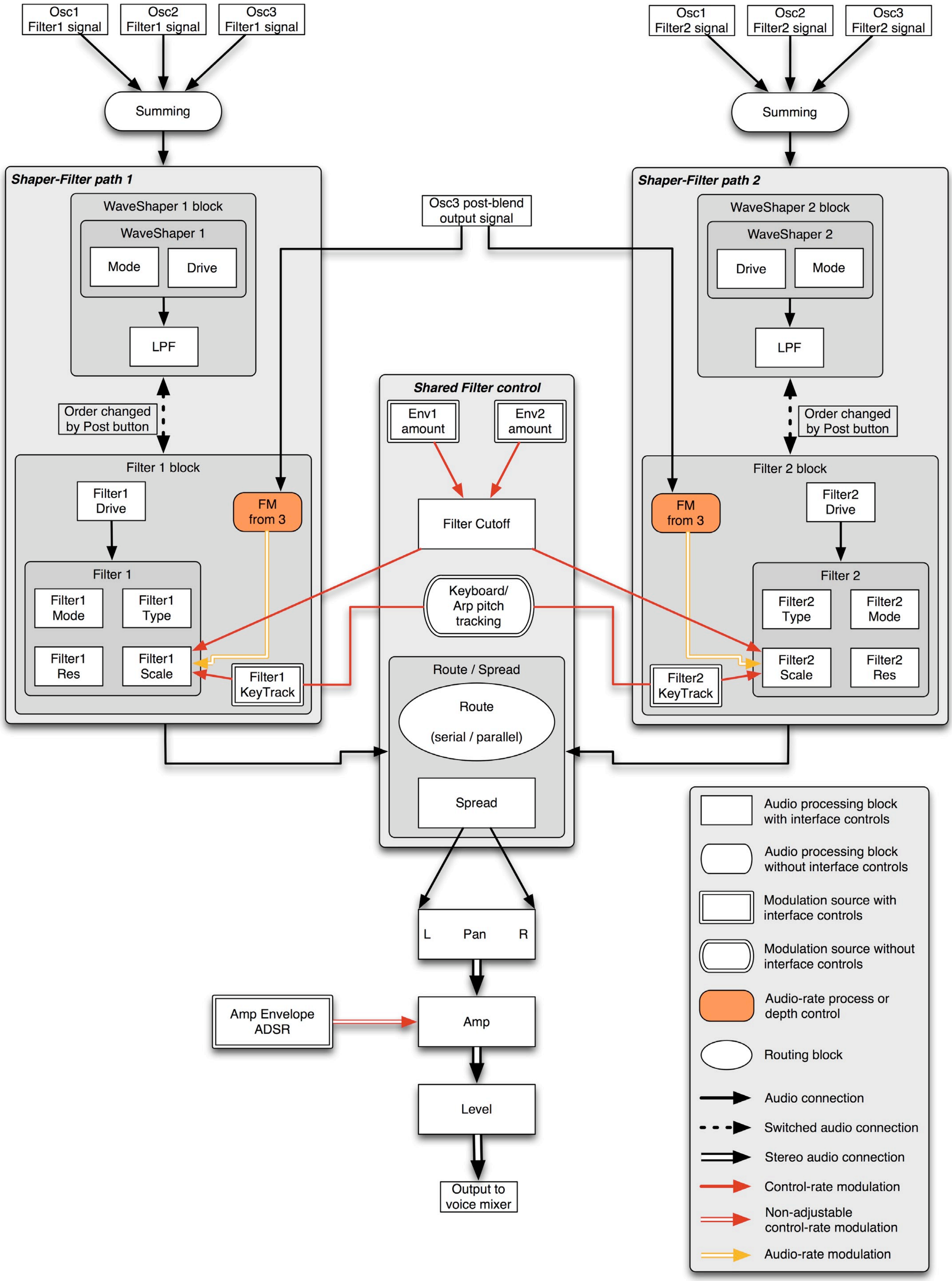
3:9 Cypher signal flow

The following signal flow diagrams represent the Osc section and the Shaper-Filter/Amp sections of a single Cypher voice. Only direct modulation is shown – there are many further possibilities when using the TransMod system.

Osc section



Shaper-Filter & Amp sections



4: Cypher's oscillators in detail

4:1 Introduction to audio-rate modulation

Basics of audio-rate modulation

While subtractive synthesis – using a filter to subtract portions of an oscillator's frequency range – is relatively easy to understand, FM and other audio-rate modulation tends to perplex most musicians.

Conventional subtractive synthesis is great for many types of mostly abstract sounds but it isn't usually sophisticated enough to produce more complex, 'realistic' timbres such as those produced by acoustic instruments. This is because most real-world timbres involve very complex harmonic variation – harmonics change at very high speeds and in more complex ways than those made possible by subtractive filtering.

Most modulation in subtractive synthesizers takes the form of control-rate functions such as LFOs and envelopes, which do not produce fast enough modulation for these complex timbres.

Audio-rate modulation – modulation at frequencies of audible sounds – is capable of producing more complex tones. FM is one of the most common types of audio-rate modulation, and can be very effective at emulating many types of acoustic sounds, particularly when using envelopes and other control-rate modulation to dictate the *amount* of FM applied. For example, sounds reminiscent of acoustic plucked and hammered instruments are much more convincing when using FM rather than conventional subtractive synthesis techniques.

Of course, it is not necessary to restrict the uses of audio-rate modulation to producing real-world timbres. It can also be used for producing new types of abstract sounds that are not possible with conventional subtractive synthesis.

Audio-rate modulation implementations

FM (frequency modulation) is possibly the most well-known type of audio-rate modulation. While it is common to associate FM with digital synths such as the Yamaha DX and TX series, FM capabilities can actually be found on many analogue synthesizers. Many classic poly- and monosynths feature FM capabilities, with one osc typically modulating another osc or the filter. Some examples include the Roland Jupiter series, SCI Prophet 5, Minimoog and ARP Odyssey. Most patchable modular and semi-modular synthesizers also allow this kind of functionality.

Digital synthesizers have a huge advantage over analogue synths when it comes to FM: it is easy to input exact values and frequency ratios, and the tuning is stable and reliable. Despite these drawbacks, the sound of analogue FM is rich and complex. With its meticulously modelled oscillators, Cypher attempts to combine the sound of analogue with the convenience and stability of a digital polyphonic software instrument.

4:2 Frequency modulation

When you modulate the frequency of one oscillator with that of another, the modulator oscillator is essentially acting as a very fast vibrato on the modulated osc's pitch.

An LFO is an oscillator running at low speeds, which produces a vibrato effect when it modulates pitch.

Modulating pitch with an oscillator that runs at audio-rates results in additional harmonics, called 'sidebands', being produced. This leads to a change in tone rather than in pitch.

The nature of these tonal changes depends on the frequency of the modulator oscillator – more accurately, how the frequency of the modulator oscillator relates to the frequency of the modulated oscillator (known as the 'carrier').

Simple, integer-based ratios between the frequencies of the carrier and modulator, such as 2:1, 1:2, 1:3, 1:4, 1:5 and so on, produce 'musical-sounding', or 'harmonic' results. The additional harmonics are musically related to the base oscillator frequency. On the other hand, arbitrary ratios such as 1:2.57, produce more dissonant results.

Cypher allows you to modulate the frequency of Osc2 with the output of Osc3. The overall shape of the Osc3 waveform is used as the modulator – therefore there are a lot of parameters that contribute to its output. However, the most important factor is Osc3's frequency, or pitch.

Osc3 used as FM source for Osc2.

The **FM from 3** control sets the depth of FM.

Oscillator FM in Cypher

Harmonic tuning

Cypher's pitch control unit/snapping options (see section 1:3) allow you to specify each osc's frequency as either of the following:

- an absolute frequency multiplier of the master pitch, expressed in semitones.
- a perfect pitch-ratio of the master pitch, expressed in harmonics (using the **Just** and **Harmonic** modes).

Using the **Harmonic** mode, you can easily set up harmonic pitch relationships between oscs.

FM depth

Before Osc3 has any effect on Osc2, it is necessary to set the depth of modulation with Osc2's **FM from 3** control. As the control is increased, more sidebands are introduced – additional tonal partials appear in the frequency spectrum.

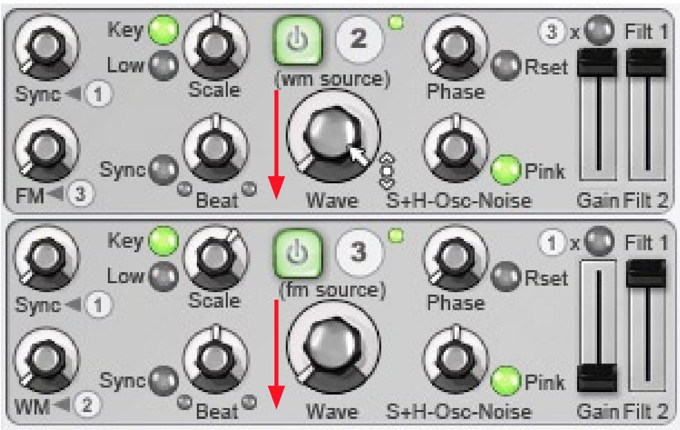
Demonstrating osc FM in Cypher

Here's a practical example to demonstrate FM, starting with an initialized Cypher state:

1. Set the Osc tuning controls to **Harmonic** mode by right-clicking on an Osc **Scale** control and selecting Harmonic from the context menu.

2. With the master **Pitch** set to 0 semitones, set the Osc3 **Scale** control to 3. Leave the Osc2 **Scale** control unchanged (1).

3. Set the **Gain** controls of Osc1 and Osc3 to 0, and fully turn up Osc2's **Gain**.



4. Set the **Wave** parameter of both Osc2 and Osc3 to a Triangle shape (move the control fully to the left).

5. Gradually increase the **FM from 3** control on Osc2 from 0 to 1 while playing a key.

The pitch ratio of 1:3 (carrier:modulator) is a harmonic ratio – the resulting sounds are musical rather than dissonant. Inharmonic ratios can be useful, however, just like detuned pitching of multiple analogue oscillators. Try setting the Osc3 **Scale** to 3.09. The result still has an overall harmonic tone, but with an interesting dissonance. More extreme inharmonic scaling of the modulator can create bell-type sounds, one of the staples of FM-based synthesizers.

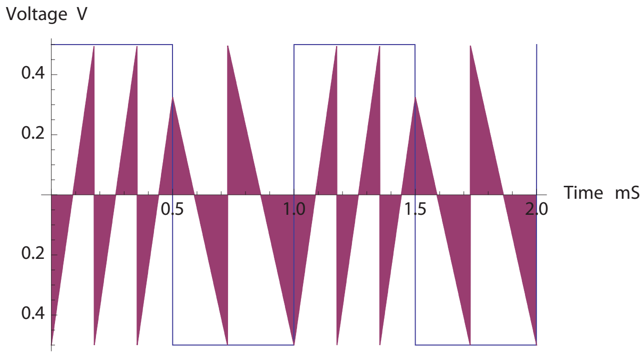
You may have noticed that as you sweep the **FM from 3** control, the change in timbre is reminiscent of a filter sweep. When using FM, it is possible to achieve a very broad range of tonal characteristics using the oscillators alone. More conventional subtractive synthesizers are heavily reliant on the filter stage for most of their tonal variety.

Thru-zero oscillator FM

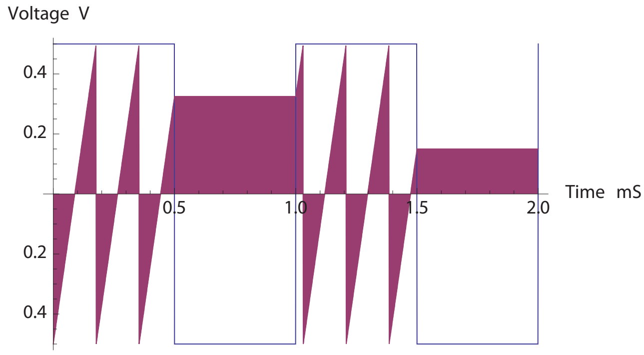
The term 'thru-zero FM' refers to the ability to modulate the frequency of an oscillator beyond zero into negative frequency values. Negative frequencies are vitally important for a stable pitch response with any possible amount of modulation, and is a prominent characteristic of the Yamaha DX-series' implementation of FM. Cypher's implementation gives you digital FM-style sounds rendered by a realistic analogue circuit model, with the convenience and polyphony of a software instrument.

Thru-zero FM is not usually possible with analogue oscillators – instead of the frequency travelling beyond zero when modulated, it simply stops at zero until the modulation causes it to rise into positive values again. The result is that, at harmonic ratios, the overall pitch that is heard is irregular at lower frequencies. Thru-zero FM allows for the timbre to remain harmonic at all frequencies even with very large modulation.

The following waveform plots show thru-zero and normal analog FM using a square wave as the modulator, and a sawtooth wave as the signal being modulated. While the square wave is high, the pitch of the saw is increased, and when the square wave is low the pitch is decreased.



Thru-zero FM:
The decrease in pitch results in a negative frequency, and the saw-up waveform becomes a saw-down waveform. A triangle waveform corner is created at the point of transition.



Normal analogue FM:
The osc is unable to produce a negative frequency when the pitch decreases. It becomes clamped at zero and stays constant until the modulator increases the pitch. The result is that the resulting pitch is not the same as the original saw wave.

There are currently some thru-zero analogue oscillators available for modular systems, such as the Cyndustries Zeroscillator. However, these tend to be rather expensive (especially if you need a polyphonic system) and there are still inherent problems such as the inability to achieve exact tuning ratios.

Thru-zero FM is very easy to implement on digital synths that use wavetable-lookup techniques, such as the Yamaha DX series. Cypher's oscillator FM is capable of thru-zero FM using true virtual oscillators generated in real time, without using wavetable-lookup techniques (which cannot offer the same sound quality as a modelled analogue-style oscillator).

Filter FM

Another classic use of analogue FM is to modulate the cutoff frequency of a filter at audio-rates such as that of an oscillator. Cypher allows you to modulate both of its two filters at audio-rate, using Osc3 as the modulator.

Again, this technique is used in order to obtain a change in timbre by modulating the filter cutoff at very fast speeds. Turning up the FM3 control on the filter reduces the action of the filter while introducing additional harmonics, resulting in a buzzy and aggressive sound.

The effect tends to be more pronounced at higher resonance settings, especially as the filter begins to self-oscillate – in such situations, filter FM is effectively like performing FM on a sine wave.

As with oscillator FM, the frequency (pitch) of the modulating oscillator is an important factor in determining the character of the resulting signal.

Thru-zero FM on a filter is inherently impossible in both analogue and digital domains. Therefore, if the amount of frequency modulation away from the initial filter cutoff frequency would take the frequency value beyond zero into negative values, the frequency stays at 0 Hz until the modulation again raises it higher again. This means that in such situations, the resulting timbre changes with the pitch of the signal going through the filter, meaning that Cypher's filter FM behaves exactly like that of a real analogue filter.

Envelopes and LFOs

FM depths become much more interesting when modulated by envelopes or LFOs. You must use the TransMod system in order to achieve this – select an Env or LFO as the source of a TransMod slot and set a modulation depth on the **FM from 3** control on Osc2 or the filter section.

Programming hints

Extreme amounts of audio-rate modulation can be great when you want wild and experimental sounds, but for most useful, 'musical' sounds, try to use smaller amounts. The same is true of modulating FM depth controls with the TransMod system – for useful rather than crazy sounds, try to find sweet spots within small depth ranges.

Try also looking at the factory Cypher presets for ideas and inspiration. Figuring out how sounds have been created using synthesis parameters is more immediate and practical than studying text-books about mathematical FM theory.

Make sure to look at all used TransMod slots to see exactly what is going on in the preset!

4:3 Wave modulation

Wave Modulation uses audio-rate modulation of an oscillator in order to vary its timbre. However, unlike oscillator FM, it modulates the *waveform* of an oscillator in order to produce these timbral changes, rather than its frequency.

Continuous waveshape of Cypher's oscs

The waveshape of Cypher's oscillators can be continuously varied between the following:

• Triangle
• Saw
• Square
• Pulse

If you move the **Wave** control while watching Cypher's visualizer scope, you can see exactly how the waveform morphs between the available shapes. The control can be modulated by any TransMod source at control-rate.

Audio-rate waveform modulation

Osc3 used as WM source for Osc2.

The **WM from 2** control sets the depth of WM.

A specialized function of Osc3 allows you to modulate its waveform at audio-rate using the output of Osc2 as the modulator. Increase the **WM from 2** control in order to set the amount of waveform modulation. The movement of the oscillator through the waveshapes at these fast rates causes timbral changes due to the varying harmonics of the waveshapes.

Similarly to FM, the frequency of Osc2 (adjusted using the Osc2 **Scale** control) has a major influence on the sound of Osc3 when using audio-rate wave modulation.

The initial position of the Osc3 **Wave** control also has a large influence on the character of the timbral changes, due to the harmonic differences between the waveforms. For example, a square wave contains only odd harmonics, but as the pulse width is changed (as the waveform changes towards a pulse), more even harmonics are introduced.

Changes in harmonics occur throughout the range of the Wave parameter. These changes offer a huge variety of osc sounds, whether modulated at control-rate via the TransMod system, or at audio-rate using the Osc3's **WM from 2** control.

Wave Mod and FM

Follow the oscillator FM example in the previous section and then turn up the **WM from 2** control on Osc3. You will notice the signal getting more complex and aggressive, with more buzzy harmonics. Try routing an LFO to this parameter so that it changes over time while you adjust the **FM from 3** control on Osc2.

Please note that, due to the implementation of the oscillator algorithm, there is a very short delay on the output of each oscillator (4 samples, for high-quality band-limiting of signals between oscs).

If you use large amounts of the **FM from 3** and **WM from 2** controls, the oscs display very chaotic sonic behaviour.

Effects of wave shape

While an osc's frequency (pitch) is the most important factor when performing FM on another osc, its wave shape also has a large influence on the nature and character of the sidebands generated. This is also the case on the modulated osc. When using the Square/Pulse range of the **Wave** control, the resulting sidebands sound very aggressive and extreme.

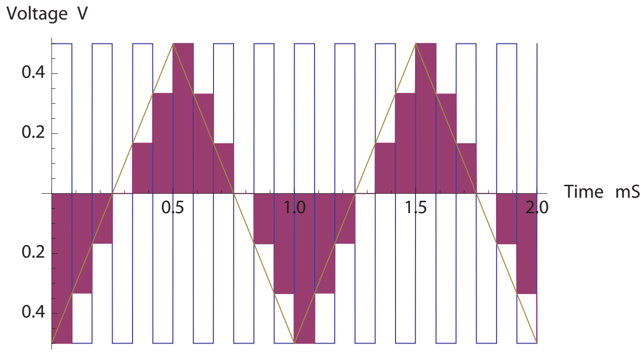
Wave shape, like other elements of the oscillator, is relevant to other processes, not just FM. It also affects the ring mod and sync functions, for example.

4:4 Audio-rate sample and hold

Cypher features the ability to sample and hold (S+H) an oscillator using the frequency of another, at audio-rate. A sample and hold operation involves the amplitude of one signal being sampled and held at a regular clock pulse. On each clock pulse, the signal's amplitude value is snapshotted ('sampled') and kept constant ('held') at the same value until the next clock pulse.

The sample and hold technique is used for a variety of purposes in analogue synthesis, such as generating a random LFO by performing S+H on a noise signal.

In Cypher, it is used to provide another audio-rate modulation technique to offer increased timbral variation from each osc. The clock pulse is provided by the modulating oscillator's frequency. The result is a quantizing effect on the modulated osc's waveshape, illustrated by the following diagram:



Audio-rate sample and hold on a Triangle waveform.

The sonic characteristics are gritty and buzzy, reminiscent of low-resolution FM and sampling.

The S+H is achieved by using the **Blend** control on each Osc, labelled 'S+H-Osc-Noise'. This control allows a blend of the S&H signal and the actual oscillator signal.

The S+H position of the Blend control is not comparable to the **FM from 2** and **WM from 3** amount controls elsewhere in Osc2, Osc3 and the filter section. The S+H process is always active, with the Blend control allowing you to alter the oscillator mix to include the S&H signal.

Here is a summary of what is heard when each osc's Blend control is turned fully to the left:

Osc1	Osc2 sampled & held by Osc1's frequency. Osc1 is the clock trigger acting on Osc2
Osc2	Osc3 sampled & held by Osc2's frequency. Osc2 is the clock trigger acting on Osc3
Osc3	Osc1 sampled & held by Osc3's frequency. Osc3 is the clock trigger acting on Osc1

Using audio-rate sample & hold

1. Fully turn down the **S+H-Osc-Noise** (**Blend**) control on Osc1. The sound of Osc2, sampled and held by Osc1's frequency, is now the output of the Osc1 block.

2. At first you won't hear anything as the pitch of both oscs is the same.

Turn up the Osc1 **Scale** control while playing a key.

You'll now hear the sound of Osc1's frequency applying a sample & hold process to Osc2. Lower Osc1 frequencies result in a more quantized Osc2 signal. Many parameters can influence the sound. Try the following to investigate the affect of these parameters on the timbre (you can adjust the parameters manually or by applying modulation to the controls):

- Change the waveforms of both oscs using the **Wave** control
- The pitch (**Scale**) of Osc2
- Increase Osc3's **Scale** and then increase Osc2's **FM from 3** control

4:5 Variable-depth oscillator sync

Compared to FM, oscillator synchronization is far more common on classic analogue synthesizers. This is because it makes complex harmonic timbres possible without suffering from the inherent tuning drawbacks of analogue frequency modulation.

Most conventional implementations of osc sync are known as ‘hard sync’. This is when the cycle of an oscillator (called the ‘sync source’ or ‘sync master’) is used to reset the phase of a second oscillator (the ‘sync slave’).

In order to create a useful effect, the sync slave osc’s frequency should be set higher than that of the sync source osc.

The end result is a more complex waveform from the slave osc that contains additional harmonics, but maintains the same fundamental frequency as that of the master osc.

Main uses of osc sync

A practical advantage of osc sync is that it allows you to combine oscillators without the inherent beating that occurs by mixing them normally (due to the fact that it is virtually impossible to get two analogue oscs perfectly in tune with each other). While the sound of oscs beating against each other is often desirable, there are situations when it can be a problem.

Perhaps the most well-known uses of osc sync is to create extreme, aggressive lead sounds. Another common technique is to perform ‘sync sweeps’ – where the slave osc’s frequency is swept manually or with an LFO to create harmonically rich movement.

Variable-depth osc sync in Cypher

In Cypher, Osc1 is the sync master, and Osc2 and Osc3 are the sync slaves. Both Osc2 and Osc3 can be synced to Osc1’s frequency – their phase can be reset on each cycle of Osc1.

You will notice that rather than simply having ‘sync on/off’ buttons for the slave oscs, Cypher has variable controls, called **Sync to 1**, for these two oscs.

Minimum (0) and maximum (1) settings

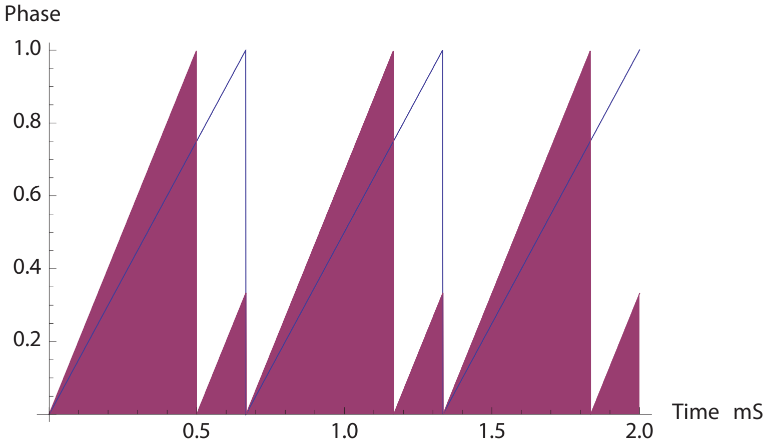
When **Sync to 1** is turned up to its maximum setting (1) on Osc2 or Osc3, the phase of these oscs is reset fully to their initial state upon each cycle of Osc1. This technique is commonly known as ‘hard sync’. When the control is at the minimum setting (0), no osc sync occurs.

Settings between 0 and 1

At **Sync to 1** settings between 0 and 1, Osc2 or Osc3 reset their phase on each Osc1 cycle *only* if the following condition is met:

$$\text{Osc2 [}\textit{or Osc3}\text{] phase} > \left(1- \frac{\text{(Osc2 [}\textit{or Osc3}\text{] }\textbf{Sync to 1}\text{ amount)}}{100}\right)$$

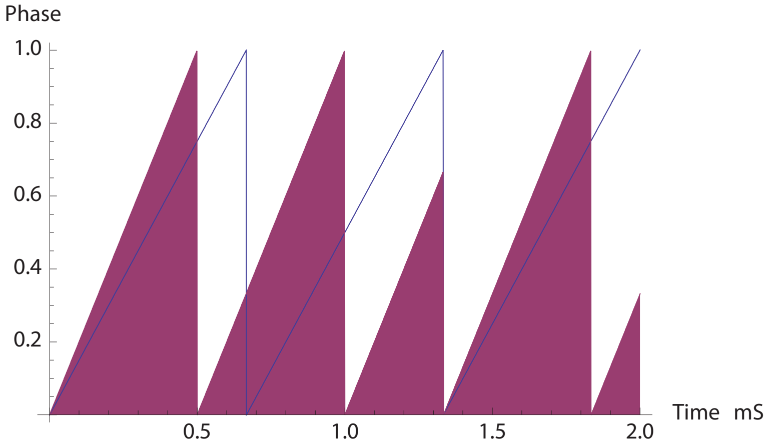
This resembles some analogue implementations of ‘soft sync’, and can create a range of complex sounds that are not possible with full, ‘hard’ osc-sync.



Sync to 1 at 100%

This example shows a saw-up waveform (filled) on Osc2, synced to a lower-frequency saw-up waveform on Osc1. The **Sync to 1** control is set to 100%.

Note the phase of the Osc2 waveform is reset to 0 on every completed cycle of Osc1.



Sync to 1 at 50%

In this example, the Osc2 waveform does not re-sync on the first cycle of Osc1, because Osc2’s phase is approx. 0.35, which is not greater than 0.5 (or 1-($^{50}/_{100}$)).

On the second cycle of Osc1, Osc2’s phase is approx. 0.67, which is greater than 0.5 – therefore Osc2’s phase is reset to 0 and it syncs to Osc1.

Using variable-depth osc sync in Cypher

1. Change the osc section's unit/ snapping options to **Equal**.

2. Fully turn up Osc2's **Gain** control.

3. Play a key – you'll hear a simple saw-wave tone.
Now increase the Osc2 **Scale** control to +15 and play a key. You should hear a 2-note minor chord.

4. Increase the Osc2 **Sync to 1** control.
As the setting goes over about 30%, the chord is no longer heard – it is now a complex waveform.

5. Now try a 'sync sweep': move the Osc2 **Scale** control up and down.

4:6 Beat detune

When two detuned oscillators play together, their differing frequencies cause an audible 'beating' effect against one another. The rhythmic rate of this beating varies depending on the notes played, since beating is directly affected by the frequencies of the notes themselves - different notes generate different frequencies. While it is common to use such sounds in a rhythmic way, any performance is always influenced by each note creating a different beating rate.

Each of Cypher's oscillators includes a useful feature for providing this detuned beating effect with the same rate across the whole keyboard range, resulting in new ways of performing throbbing dubstep basses and rhythmic pads and chords.

Using the Beat control

The **Beat** control sets the frequency rate of the beating effect in Hz. To understand how this control works, follow this simple example:

Normal detuning

Start with an initialized Cypher state. Osc1 and Osc2 are both set to the same frequency in relation to the master **Pitch**. When played together normally, with the **Beat** control set to zero, there is no beating – the oscs are perfectly in tune with each other.

Now, set the Osc2 **Scale** control to 1.04 (in other words, +4 cents or +0.04 semitones). This creates beating at a rate of approximately 1 Hz when playing the A4 key on the keyboard.

This is because this pitch, when combined with keyboard tracking, causes the A4 key to produce a frequency of approximately 441 Hz from Osc2. When this osc frequency plays at the same time as Osc1 at exactly 440 Hz, the difference between each osc's frequencies is the rate of beating.

Manual detuning:

Initialize Cypher, turn up Osc2's **Gain** fully and set Osc2's **Scale** control to 1.04 (double-click the control and type the value).

However, if you were to play the A5 key, the beating rate would be approximately 2 Hz. This is because, with keyboard scaling, each increasing octave doubles the oscillator frequency: the Osc2 frequency is 882 Hz, while Osc1 is at 880 Hz. The difference between them is 2 Hz, which is the rate of beating.

Beat detuning

Now set the Osc2 **Scale** control back to 1 and set the Osc2 **Beat** control to 1 Hz. This results in a beating rate of 1 Hz across the whole keyboard, because the 1 Hz is added *after* the keyboard scaling is applied:

Osc2 at A4: 440 Hz + 1 Hz = 441 Hz
Osc2 at A5: 880 Hz + 1 Hz = 881 Hz

Because the difference between the frequencies of the two oscs is the same, the beat rate is also the same.

With the **Sync** button enabled, the **Beat** control is set in BPM-based values synced to the host tempo, allowing tempo-synced beating effects.

A dedicated beat rate control is very rarely found on hardware analogue synths, with a notable exception being the Moog Taurus bass synthesizer.

Beat detuning:

Initialize Cypher, turn up Osc2's **Gain** fully and set Osc2's **Beat** control to 1 Hz.

Beat rates as TransMod modulators

The beating rates of the various oscillator combinations are available as TransMod control-rate sources. This means you can modulate other parameters at the beat rate, allowing for unprecedented rhythmic modulation effects synced to osc beating.

These sources are discussed at the end of section 7:8.

Usage with other oscillator functions

The **Beat** feature offers interesting possibilities with other oscillator functions, as it provides a rhythmic frequency detuning effect. Try using it alongside FM, WM, ring mod and sync functions.

5: Amber

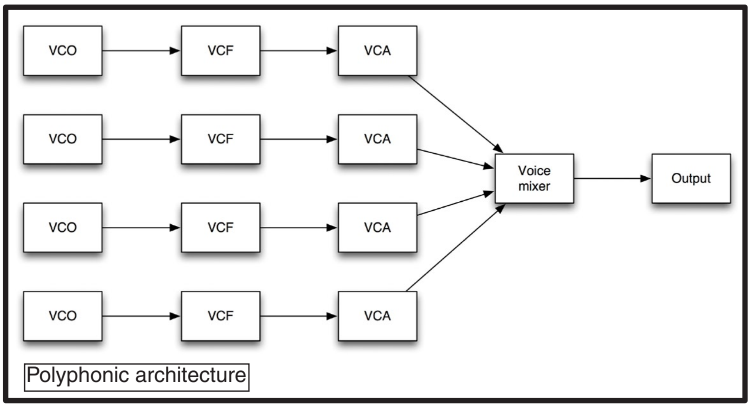
5:1 Overview

Amber represents one of the first attempts to realistically model classic string-ensemble synthesizers from the 1970s. These instruments featured a special oscillator technique for full ‘polyphony’ using a single synth voice. The word ‘polyphony’ is in inverted commas because these instruments were technically not true polyphonic synths.

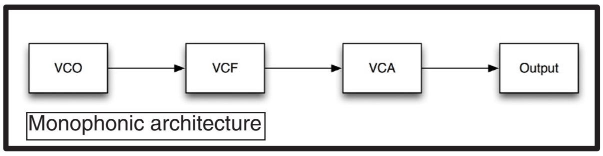
Divide-down synthesis explained

To be fully polyphonic, a synth must be able to produce several entire voices simultaneously. This means that a true polyphonic synth is effectively a stack of full monosynths with some extra processing functions for assigning keyboard input to the available voices.

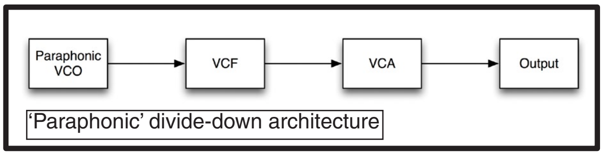
A divide-down synthesis architecture is very different. Generally, the frequencies of a set of 12 master oscillators (for the highest octave) are divided down to lower frequencies multiple times to produce the full range of notes on the built-in keyboard (typically 4 or 5 octaves). The principle is similar to Strobe’s sub-oscillators, except that keyboard input provides access to these sub-octaves instead of a fader.



Polyphonic architecture



Monophonic architecture



‘Paraphonic’ divide-down architecture

This approach became known as ‘paraphonic’ – a term introduced by Roland’s RS-505 Paraphonic Strings instrument. While the oscillators produce multiple notes simultaneously, all notes typically shared a single amp envelope and filter.

Implementations of divide-down synths

Divide-down synths were not intended to produce ‘big’ sounds such as those from a conventional ‘fat’ analogue mono- or polysynth. In fact, most implementations of this technology involved using a final chorus stage to thicken the output and create the illusion of many string instruments playing at once to create an ensemble sound. Turning off the chorus resulted in a rather sterile and characterless timbre – all notes in the keyboard range are perfectly in phase, meaning there is no beating.

Sometimes a monophonic bass synth section was provided, such as that in the ARP Omni series, which typically bypassed the chorus circuit and featured a resonant filter, allowing the bass register to cut through more powerfully.

Instruments based on divide-down technology became primarily used for simple string, piano and organ sounds – actually not very ‘realistic’, but very passable and pleasant representations, with the crucial ability of being able to play all notes on the keyboard simultaneously.

String synths in popular use

In spite of their limitations, string synths became very widely used. It was common in the 1970s for keyboard players to use a string ensemble synth alongside other synths and electromechanical keyboards. Perhaps the most well-known examples were the ARP Omni series, Roland RS-505 and Eminent Solina.

A few string synths such as the Logan String Melody series and the Korg Lambda featured a dedicated AR amp envelope for *each note* on the keyboard. This approach involved increased cost, weight and maintenance issues, and the less sophisticated single-VCA instruments like those by ARP and Eminent were generally more widely used.

Even more sophisticated implementations of divide-down technology were involved in the Moog Polymoog and the Korg 3x00 series, with the latter particularly notable for being a paraphonic semi-modular synth.

Amber: the next step in divide-down synthesis

Amber's design builds on the string synthesizer legacy, while retaining crucial elements that made them so special.

Both single and multiple paraphonic envelope behaviour is possible, along with velocity-sensitivity. A formant filter and three different chorus models are provided for a wide range of sounds.

Because Amber shares the same overall voice, unison and modulation architecture as the other synths in DCAM: Synth Squad, it has the ability to stack and modulate entire paraphonic voices.

Synth and Ensemble sections

Amber features two sophisticated paraphonic sections within a single synth voice. Each of these sections contains a paraphonic note-generating stage, followed by polyphonic processing blocks, which are monophonic for each voice. Both sections are capable of playing all notes on the keyboard simultaneously within a single synth voice – up until a certain point in the signal flow.

Synth section	The paraphonic stage (which features a noise generator) is summed to a single signal before the polyphonic resonant multimode filter.
Ensemble section	The Ensemble section is summed to a single signal before the polyphonic formant filter and chorus.

Both sections are very similar (the Synth section's noise generator is the main difference), with their respective monophonic processing stages defining their character. Optionally, the Synth section can be routed through the Ensemble section's formant filter and/or chorus.

Contents of a single Amber voice

A single voice of Amber processes the following audio generation and processing blocks within both paraphonic sections:

- 12 oscillators for each note in the top octave, sub-divided into 8 lower octaves
- 192 envelopes and VCAs, velocity-sensitive
- 384 1-pole tone filters (2 per note for both the Ensemble and Synth sections)

Additionally, it includes all the polyphonic audio blocks – multimode resonant filter, formant filter, chorus and final mixer/amp.

Therefore, it is very CPU-intensive when running a single voice. Please do not expect to be able to run the same number of Amber voices on your system as those for Cypher and Strobe!

5:2 Master tuning controls

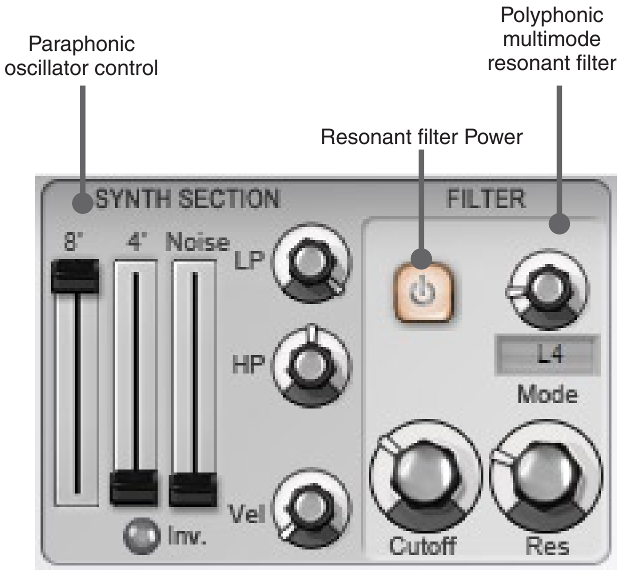
The **Pitch** and **Fine** controls allow you to tune the master oscillator bank.

Due to Amber's paraphonic architecture, these controls adjust the pitch of all paraphonic notes within a single voice simultaneously.

You will mainly need to use these controls when using multiple Amber voices – try setting some modulation on the **Fine** control from a Voice or Unison source in order to detune each voice.

You could also use the Drift source to modulate the pitch when using a single voice, in order to simulate a synth with unstable, drifting oscillators.

5:3 Synth section



Paraphonic oscillator control

Level controls: 8', 4' & Nse

The Synth section features level controls for the **8'** and **4'** octaves, and for a gaussian white noise generator (**Nse**).

When using the AmberFX plugin (MIDI-controlled audio effect plugin version), the external audio input appears in place of the **Nse** source.

Inv (invert)

The **Inv** button inverts the phase of the Synth section in relation to the Ensemble section. This can result in interesting phase-cancellation effects when both sections are in use.

Paraphonic ADSR envelopes and VCAs

Each note in the Synth section possesses its own dedicated ADSR envelope to control its amplitude over time (the envelopes are internally connected to VCAs for each note). The shape of all these envelopes is dictated by the Synth Envelope.

Vel (Velocity)

The **Vel** control sets the velocity sensitivity for the Synth section's VCAs. At the minimum setting, there is no modulation of amplitude by velocity.

LP & HP (paraphonic 1-pole tone filters)

The tone of each paraphonic note is adjusted by a 1-pole filter stage featuring high-pass and low-pass filters. Use the **LP** and **HP** controls to adjust the cutoff frequencies of these filters.

These filters are keytracked (directly modulated by keyboard pitch) relative to the cutoff frequencies, with maximum depth. This direct modulation cannot be adjusted.

Filter stage (polyphonic multimode resonant filter)

All the individual paraphonic Synth section notes are summed before being passed through the multimode resonant filter stage.

This filter stage is polyphonic – *monophonic for each voice*.

Power

The **Power** button enables or disables the filter. When disabled, incoming audio passes through the filter stage unaffected.

Cutoff & Res (Resonance)

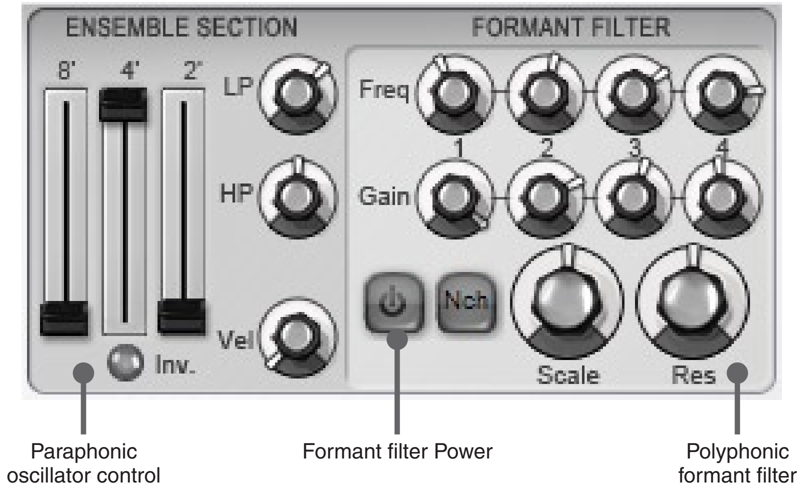
The **Cutoff** and **Res** parameters control the cutoff frequency and resonance of the filter stage,

Mode

The resonant filter can be used in any of the following modes:

- Low-Pass 2-pole or 4-pole
- High-Pass 2-pole or 4-pole
- Band-Pass 2-pole or 4-pole
- Peak 4-pole
- Notch 4-pole

5:4 Ensemble section



Paraphonic oscillator control

Level controls: 8', 4' & 2'

The Ensemble section features level controls for **8'**, **4'** and **2'** octaves.

Inv (Invert)

The **Inv** button inverts the phase of the Ensemble section in relation to the Synth section. This can result in interesting phase-cancellation effects when both sections are in use.

Paraphonic AR envelopes and VCAs

Each note in the Ensemble section possesses its own dedicated AR envelope to control its amplitude over time (the envelopes are internally connected to VCAs for each note). The shape of all these envelopes is dictated by the Ensemble Envelope's parameters.

Vel (Velocity)

The **Vel** control sets the velocity sensitivity for the Ensemble section's VCAs. At the minimum setting, there is no modulation of amplitude by velocity.

LP & HP (paraphonic 1-pole tone filters)

The tone of each paraphonic note is adjusted by a 1-pole filter stage featuring high-pass and low-pass filters. Use the **LP** and **HP** controls to adjust the cutoff frequencies of these filters.

These filters are keytracked (directly modulated by keyboard pitch) relative to the cutoff frequencies, with maximum depth. This direct modulation cannot be adjusted.

Formant filter stage

Amber's formant filter stage is provided primarily to impart more realistic string-like characteristics on the Ensemble section (and optionally the Synth section by adjusting the **Syn Route** control). This allows for a versatile range of string timbres, as well as more experimental sounds beyond the scope of most string synths.

It is unusual to have a formant filter on a string-synth, although some synths, such as the ARP Omni, featured fixed filtering on each part of the string section (violin, viola, cello, bass) to give it its own character. Other synths featured additional filters within the chorus circuit in order to produce these types of effects.

The formant filter is polyphonic – *monophonic for each voice*. All paraphonic notes in the Ensemble section are summed before reaching this point in the circuit.

Formant filter controls: Power, Freq, Gain, Scale & Notch

To enable the formant filter, engage its **Power** button.

The formant filter comprises 4 band-pass filters, each with their own **Freq** (frequency) and **Gain** controls. The resonance of all bands is controlled by a single **Res** parameter, while the **Scale** control adjusts the frequency of all bands simultaneously.

The **Nch** (Notch) button switches all bands to notch filter mode – in this mode, the Gain controls are inactive.

The formant filter is capable of self-oscillation – use the final mixer/amplifier controls to tame the sound so that it does not distort at higher resonance settings.

5:5 Ensemble section chorus stage

The final chorus processing is a very important part of a string ensemble synth, as it provides the lush, 'blended' ensemble sound of a string orchestra playing together. Amber's chorus is modelled on classic bucket-brigade delay (BBD) chorus circuits found in many old synthesizers and stomp-boxes. 3 different models are available using the **Mode** control, with each providing their own unique sonic character:

- 1975
- 1981
- 1984

The chorus stage is part of the Ensemble section, but it is positioned after the **Ens. Amp** in the signal flow. The Synth section is not routed through the chorus by default, although this can be achieved using the **Syn Route** control in the mixer/amplifier section.

Enable the Chorus stage by engaging its **Power** button. The chorus features controls for speed of pitch modulation (**Rate**) and stereo **Spread**.

Extra filters were often integrated into string synth chorus circuits in order to shape the tone to be more string-like. Amber has a dedicated formant filter for this purpose, so you may need to enable 'Bright' mode (using the **Brt** button) to deactivate the extra filtering within its circuit.

5:6 Mixer / Amplifier stage

Synth & Ens. (Synth Amp and Ensemble Amp)

The mixer/amplifier stage features separate Amp controls for the **Synth** and **Ens.** (ensemble) sections of the voice. Since the paraphonic notes have already been summed to single signals by the time they reach this stage, the **Synth** and **Ens.** controls effectively mix the levels of the two sections within the voice. Note that the actual mixing stage occurs after the chorus – the **Ens.** control sits between the formant filter and chorus in the circuit.

There is a waveshaping stage between the **Synth**/**Ens**. and **Level** controls, which is capable of distortion when its components are driven hard.

If you want a cleaner sound, keep the **Synth** and **Ens.** controls at low settings and increase the **Level**.

If you want to overload the amp by increasing the **Synth** and **Ens.** controls, remember to turn down the **Level** control. Otherwise, the output of Amber may clip.

Syn Route

The **Syn Route** control allows the Synth section to be routed through the Ensemble section's formant filter or chorus if desired. This routing occurs *after* the Synth Amp stage.

Pan

The **Pan** parameter exists *after* the Synth and Ensemble sections are summed – therefore if you need to pan the 2 sections separately you must use unison voices and appropriate TransMod modulation.

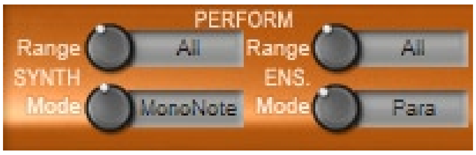
Level

This parameter sets the final **Level** of each voice before it is summed with all other active voices at the final output of the synth.

Analogue

The **Analogue** parameter simulates the effect of noise and mains hum in certain parts of the audio and control signal paths, something that always occurs in real analogue synth circuits. At lower settings, it leads to a subtly gritty and slurring character, while higher settings create a more unstable and noisy sound.

5:7 Perform controls



These controls allow you to change the playing behaviour of the Synth and Ensemble sections.

Range

The **Range** control represents a method of splitting the keyboard range of the Synth and Ensemble sections. Either section can be set to 'Low', 'High' or 'All'.

All: C2 to B9
Low: C2 to G4
High: G#4 to B9

Mode

The Performance Mode settings allow both the Synth and Ensemble sections to follow one of several behaviours – paraphonic, monophonic notes, monophonic envelope attack or monophonic envelope release.

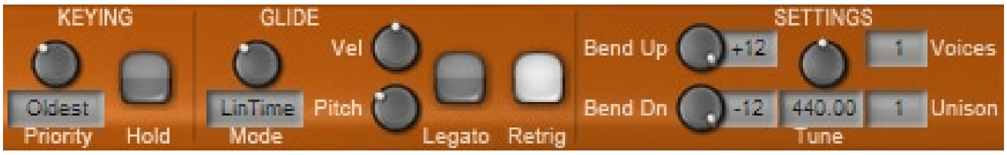
'Paraphonic' emulates the playing style of string machines such as those by Logan and Korg, featuring an individual VCA for each note on the keyboard.

The 'MonoAtk' and 'MonoRel' modes each involve a different style of single-VCA paraphonic playing response. Single-VCA synths, such as the Solina and Omni series, possessed their own feel and sound due to their limitations heavily influencing the way they had to be played.

Setting both sections to 'MonoNote' results in Amber responding like a conventional polyphonic synth: each voice can generate only 1 note for both sections.

Paraphonic	Every key triggers each note's individual amp envelope
MonoNote	Only the highest priority note is heard. The priority is set using the Priority control in the Keying controls. If Priority is set to 'Lowest' and this mode is selected for the Synth section you can play a monophonic bassline to accompany your string chords.
MonoAtk	All notes share a single attack amp envelope, allowing you to play a part with the volume slowly swelling over all the notes.
MonoRel	All notes share a single release envelope, allowing chords to slowly release until new notes are played, whereupon the old notes are muted.

5:8 Other functions



Settings, Glide and Keying controls

These parameters are included in all DCAM: Synth Squad synths, and are described in sections 6:2, 6:3 and 6:4.

Visualizer scope

The visualizer scope provides context-sensitive graphical feedback for each section of Amber. See section 6:6 for more details on this feature.

5:9 Modulation

The TransMod system must be used for all programmed modulation in Amber.

With the exception of the Synth and Ensemble Envelopes (directly routed internally to the individual paraphonic Synth and Ensemble notes' VCAs) and the non-adjustable keytracking for the paraphonic 1-pole tone filters, none of Amber's modulators have direct routings to any parameters. There are no controls for setting the depth of internal direct modulation.

Paraphonic architecture and polyphonic modulation

Amber's divide-down synthesis architecture is very different from the conventional polyphonic architecture of Strobe and Cypher, meaning that modulation must also be approached differently. In polyphonic (or monophonic) architectures, a modulator such as an LFO acts *per voice* – each voice is shaped by the modulator polyphonically. In other words, each voice has its own LFO operating independently of the other voices' LFOs.

Amber's divide-down 'paraphonic' architecture means that one voice actually generates all notes on the keyboard within a single voice. This paraphonic portion of the synth actually contains 192 envelopes and VCAs – one for each note it can produce in the Synth and Ensemble sections.

However, the voice also contains elements that feature a single instance for every voice – LFO, Ramp and ModEnv modulators, and the multi-mode resonant filter, formant filter, chorus and Mixer/Amplifier stages in the signal path. These elements all act polyphonically (monophonically for each voice) rather than paraphonically. Similarly, the TransMod system also acts polyphonically.

It is therefore necessary to disassociate the concepts of 'more than one note at a time' and 'polyphony' – when using Amber in conventional 1-voice mode, the polyphonic elements act monophonically as only a single voice is active.

In order to achieve different modulation values on different notes, you *must* use multiple voices and suitable TransMod modulation.

The Synth and Ensemble envelopes are special cases (see below for further details).

Gateable modulators

Gateable modulators (which actually can be gated with note-on/note-off or triggered with note-on) feature a number of different gating/triggering modes, including the ability to gate or trigger each other. For more details, see section 7:5.

LFO

Gateable by: Poly, PolyOn, Mono, Ramp, ModEnv, Song

Ramp

Gateable by: Poly, PolyOn, Mono, LFO, ModEnv, Song

Mod Envelope

Gateable by: Poly, PolyOn, Mono, LFO, Ramp, Song

Synth and Ensemble Envelopes

These envelopes are not gateable modulators like other envelopes in DCAM Synth Squad – they can only be gated by keyboard input. They have two distinct functions:

1. They are used to determine the envelope shape of the 96 amp envelopes for each paraphonic note in the Synth and Ensemble sections.
2. While the paraphonic envelopes are not available in the TransMod system because it only operates at the polyphonic voice level, *polyphonic copies* of the Synth and Ensemble envelope shapes are available as sources in the TransMod system to provide extra modulation envelopes. In conventional single-voice paraphonic usage, these polyphonic envelopes are effectively monophonic like the ModEnv, and are gated polyphonically.

Synth envelope

The Synth envelope features **Atk** (**Attack**), **Dcy** (**Decay**), **Sus** (**Sustain**) and **Rel** (**Release**) parameters, which operate in the same way as those in the Mod Envelope (see section 7:5 for details of the ADSR stages).

It does not possess the Sync, Loop and Linear parameters of Mod and Amp envelopes.

Ensemble envelope

The Ensemble envelope operates quite differently to the ADSR envelopes found elsewhere in DCAM: Synth Squad. It is an Attack-Release (AR) envelope with optional sustain.

- When a key is gated, the envelope output rises from 0 to 1 over the defined **Atk** (**Attack**) time.
- If the **Sus** (**Sustain**) button is enabled, the output stays at 1 while the key is held down.
- If it is disabled, or when the key is released during **Sustain** mode, the envelope's level falls to 0 over the time defined by the **Rel** (**Release**) parameter.

Glide

Conventional pitch glide is not possible in Amber, because the the divide-down oscillator system constantly generates each key at its exact root pitch.

However, pitch glide signals can still be used with other parameters via the TransMod system, by means of the Pitch modulation source. This modulation can only occur polyphonically (monophonic for each voice).

Similarly, Velocity Glide also works on the polyphonic level.

For more details of the **Pitch Glide** and **Vel Glide** functions, see sections 6:3 and 7:6.

Keytracking (modulation from keyboard pitch)

Direct routings: Paraphonic 1-pole tone filters in Synth and Ensemble sections (fixed at 100% depth)

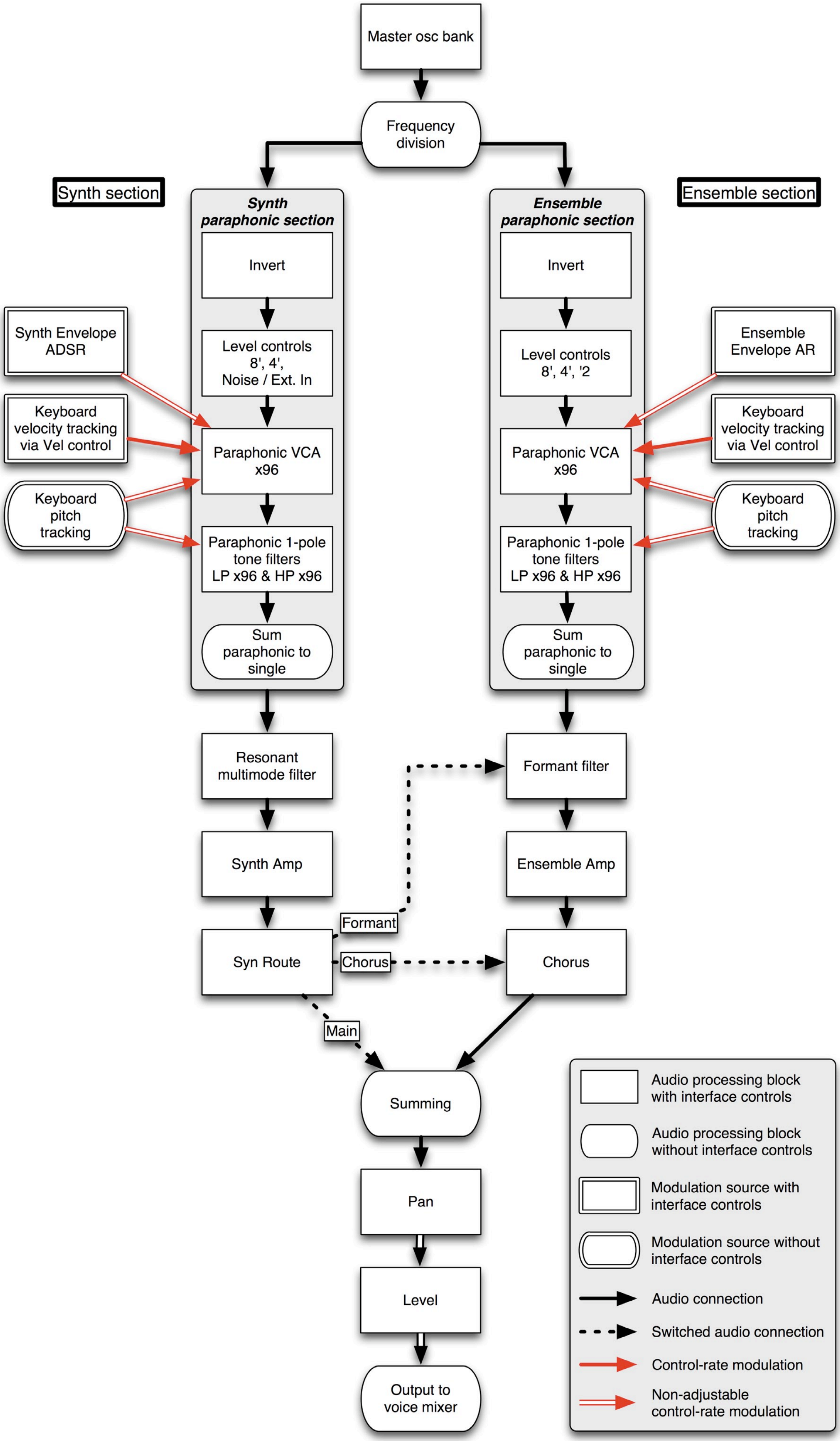
Keytracking is modulation from the pitch of the keyboard. The value of the pitch modulation source increases by 1 over each octave.

The cutoff frequencies of the paraphonic 1-pole HP and LP tone filters in the Synth and Ensemble sections are directly keytracked. The Pitch modulation source can be used in the TransMod system to keytrack any other parameter. However, this will occur polyphonically (monophonically per voice) rather than the paraphonic direct keytracking of the 1-pole filters.

See section 7:7 for more details about keytracking.

5:10 Amber signal flow

The following signal flow diagram represents a single voice of Amber. Only direct modulation is shown – there are many further possibilities using the TransMod system.



6: Common synth functions

6:1 Arpeggiator (Strobe and Cypher only)

Arpeggiators turn a number of keys held down simultaneously into a repeated arpeggio sequence. They're great for interesting melodic textures, or to suggest a chord while only using 1 synth voice (especially at faster rates). This was a very useful feature on monosynths incapable of any polyphony.

Cypher and Strobe both feature a built-in arpeggiator. Amber, on the other hand, is not a typical synth to use with an arpeggiator, and therefore does not include one built into the synth. If you want to arpeggiate Amber, you can use Fusor's Animator device, which allows you to arpeggiate any combination of DCAM synths. It also offers many functions not available in the more simple arpeggiator built into Strobe and Cypher.

Arpeggiator and LFO

As in many classic analogue synth designs, the arpeggiator in Strobe and Cypher is linked to the LFO (LFO1 in Cypher's case).

While the arpeggiator has its own independent **Rate** control, whether this rate is set as Hz or tempo-based values depends on the state of the LFO's **Sync** button.

Arpeggios are also linked to the LFO's **Swing** and **PWM** controls. Changing the LFO PWM changes the length of each note generated by the arpeggiator, while the LFO Swing amount can be used to create swung arpeggios.

The LFO's controls are described in section 7:5.

First LFO's Swing, PW and Sync controls are used for the arpeggiator

Arpeggiator Power

Arpeggiator and Keying controls

The behaviour of the arpeggiator is heavily dependent on the setting of the **Priority** parameter in the Keying controls. Held notes are sequenced in the order dictated by the Priority parameter: the sequence is played *towards* the note dictated by the parameter.

For example, if you want the sequence to follow the order in which you play the notes, set the Priority control to 'Newest'. This way, the first note you held down will always be the first in the arpeggiator sequence, the second note you play will always play second, and so on. If you want the sequence to go from low to high, set the priority to 'Highest'.

The **Hold** button is very useful for indefinite arpeggio duration without having to keep keys held down.

It also allows you to use the arpeggiator as a rudimentary step-input performance sequencer by setting Priority to 'Newest' and playing one note after another. Each note is added into the note queue in the order in which it was played, a sequence which is played back and repeated indefinitely. Please note that held notes are not saved with your preset!

See section 6:2 for further details of the Keying controls.

Arpeggiator controls

Power

Engaging the **Power** button turns on the arpeggiator. When disabled, the synth responds to keyboard input normally.

Rate

This control adjusts the speed of the arpeggiator sequence. While the **Rate** operates independently from the LFO rate, the LFO's **Sync** button must be enabled in order for the arpeggiator Rate parameter to be set in tempo-based values.

When the LFO Sync button is disabled, the arpeggiator Rate is set in hertz.

Note

The **Note** control dictates the order to play notes in the arpeggiator note queue, which in turn depends on the state of the **Priority** control in the Keying section. There are 4 settings:

Up	Plays sequentially upwards through the note priority queue, from low priority to high priority.
Dn (Down)	Plays downwards through the note priority queue from high priority to low priority
UpDn (Up and down)	Plays up through to the end of the note priority queue (high priority) and then backwards to the start (low priority).
Rand (Random)	Plays random notes from the note queue. In this mode, the **Priority** control setting is irrelevant.

Range

This control dictates the number of octaves over which the arpeggio sequence is played.

Octave

This setting is dependent on the number of octaves specified with the Range parameter, and dictates how the octave shifts occur. Multiple octaves, from low to high, are put in a 'queue', and the settings for this control specify how the arpeggiator moves through the queue.

Up	Plays sequentially upwards through the octave queue, from lowest to highest octave.
Dn (Down)	Plays backwards through the octave queue, from highest to lowest octave.
UpDn (Up and down)	Plays upwards through the octave queue from lowest to highest and then backwards to the lowest.
Rand1	Sets a new random octave after all notes in each octave have been played.
Rand2	Sets a new random octave for each note.
Link	The **Octave** mode is the same as the **Note** mode. This is useful for certain sequences that are not possible with the other modes. For example, setting **Note** mode to 'UpDn' and **Octave** mode to 'Link' results in a full upward and downward glissando of all generated notes – this would not be the case if both modes were set to 'UpDn'.

Gate

The arpeggiator can be gated by 2 gate sources:

Mono	This is the default mode of operation. The arpeggio pattern plays from the start, immediately, when keys are played.
Song	In this mode, the start of the arpeggio pattern is quantized to the nearest beat.

6:2 Keying controls

Priority

This control can be set to 'Newest', 'Oldest', 'Highest', 'Lowest', 'Hardest' or 'Softest'. It has 2 functions:

1. It is used to dictate the voice stealing priority if more notes are played than the number of voices available.

   The default setting is 'Newest', which means that new notes get priority. If you have a single voice active, playing a note while another note is already playing results in the new note 'stealing' the voice as it has a higher priority. When this new note is released, the original note is played again.

2. It determines the order in which the currently held notes will be played by the arpeggiator.

   The arpeggiator sequences the held notes in the order dictated by the **Priority** parameter: the sequence is played *towards* the note dictated by the parameter.

   For example, if you want the order to follow the order in which you play the notes, set the Priority control to Newest. This way, the first note you held down will always be the first in the arpeggiator sequence, the second note you play will always play second, and so on.

Hold

Engaging the **Hold** button results in incoming MIDI note off messages being ignored until the button is deactivated. Therefore, if the Hold button is enabled, any keys already playing or subsequently played are sustained indefinitely, even after the keys are released.

This button allows the arpeggiator to play indefinitely without having to keep keys held down.

It also allows you to use the arpeggiator as a rudimentary step-input performance sequencer by setting the **Priority** control to 'Newest' and playing one note after another. Each note is added into the note queue in the order in which it was played, a sequence which is played back and repeated indefinitely.

Please note that held notes are not saved with your preset!

6:3 Glide controls

This section contains a number of controls that relate to glide functions and gateable modulator retriggering.

It also contains the only modulateable controls in the common synth functions – the **Vel Glide** and **Pitch Glide** parameters.

Section 7:6 covers DCAM: Synth Squad's glide functions in detail, while section 7:5 contains a detailed discussion of gateable modulators and their behaviour.

Legato
Retrig

These buttons allow flexible control of fingered portamento and modulator gating.

The **Legato** button relates to glide: enabling it results in a 'fingered' glide – meaning that glides only occur when 2 notes 'overlap'. This occurs for **Vel Glide** and **Pitch Glide**.

The **Retrig** button dictates whether gateable modulators (Ramp/LFO/ModEnv/AmpEnv) retrigger when notes overlap, if they are set to the Poly gating mode.

The fact that each of these functions features a dedicated button means that you can use glides while retriggering the filter envelope, for example.

Glide Mode

This control changes the glide time response between 'LinearTime', 'LinearRate', and 2 exponential settings. They all give the glide 'curve' a different shape and playing feel.

Pitch Glide
Vel Glide (Velocity Glide)

These parameters control glide times, and can be modulated via the TransMod system.

The **Pitch** control sets the glide time towards the pitch of new note events.

The **Vel** control sets the glide time towards new OnVel modulation depths.

6:4 Settings

This section contains controls for voice-related settings of the current preset.

Voices
Unison

These numerical text-boxes set the number of active voices for the synth, as well as the number of unison voices.

If you play more simultaneous notes than the current voice and unison settings allow, voice-stealing functions are applied, the behaviour of which is specified by the **Priority** control in the Keying section. See section 6:2 for details of this control.

Active voices and unison voices generate TransMod modulation values, accessible using the polyphonic Voice and Unison sources. These sources are discussed in sections 7:6 and 7:8.

Voices

A synth voice can be expressed as an entire synth that plays a single note at a time – a monophonic synth features 1 voice, while a polyphonic synth is effectively a stack of monosynths with a simple logic circuit for distributing keyboard input amongst them.

When using Strobe or Cypher, you need more than 1 **Voice** to play more than 1 note at a time (although the number of notes you can play simultaneously also depends on the **Unison** setting – see below).

Amber's paraphonic architecture means that it generates the entire keyboard range simultaneously within a single voice. Please read the respective chapters on each synth for a full understanding of their voice architectures.

Unison

The number of **Unison** voices is a sub-set of the maximum number of voices. It determines how many voices to stack for each note. For example, specifying 8 voices and 2 unison voices results in 4-note polyphony, with each note comprising 2 stacked voices.

In traditional analogue synths, unison voices could usually only be detuned against each other. Using the polyphonic unison TransMod sources in DCAM: Synth Squad, you can affect any parameters of unison voices, leading to unprecedented possibilities.

Bend Up
Bend Dn (Down)

These controls are for setting pitch bend sensitivity – they allow you to separately adjust the depth of upward and downward pitchbend input, to a maximum of 12 semitones in each direction.

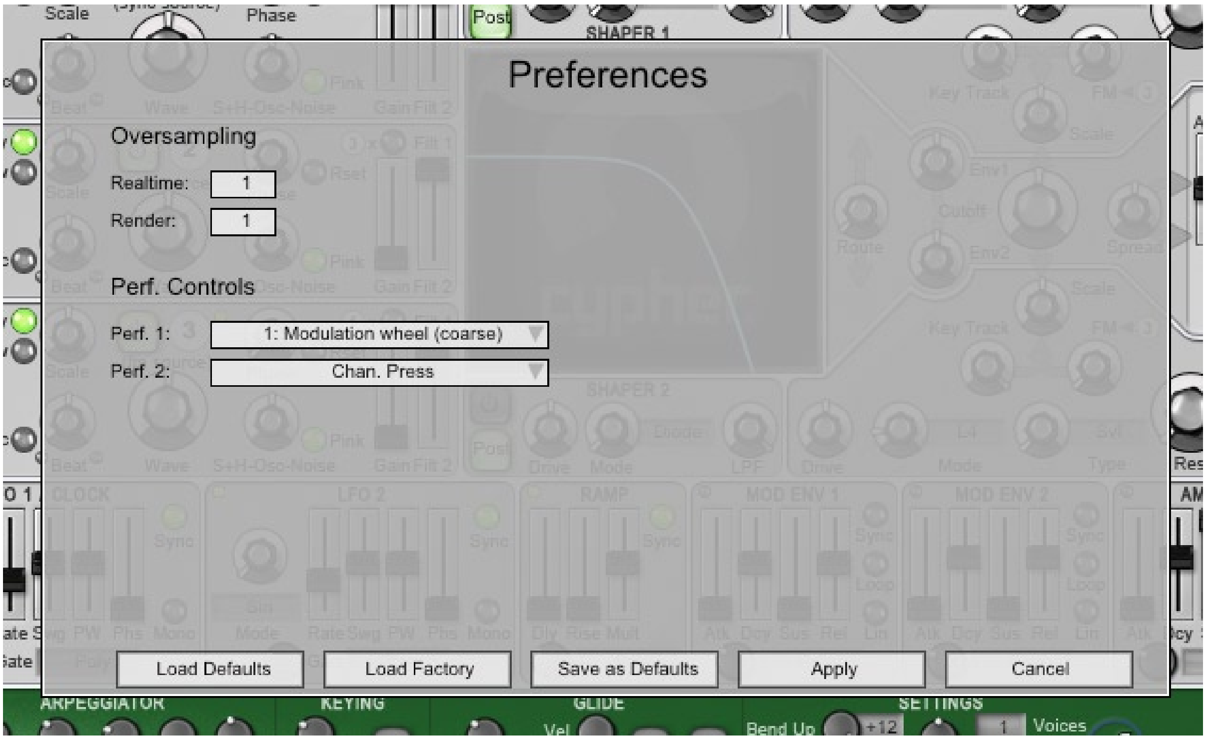
Tune

This control sets the master tuning. It specifies the frequency for the A note above middle C. The default is 440 Hz, and is adjustable between 420 and 460 Hz.

6:5 Preferences panel

The preferences panel stores global settings that are not stored within presets. They are unique to each plugin in DCAM: Synth Squad. Strobe, Cypher, Amber and Fusor each have their own preferences settings, as do the StrobeFX, CypherFX and AmberFX effect plugins. When the synths are used in Fusor, their preferences settings are ignored: Fusor's own performance controls and oversampling settings are used instead.

Click the **Preferences** button in a synth's preset picker in order to launch the preferences panel.



Oversampling

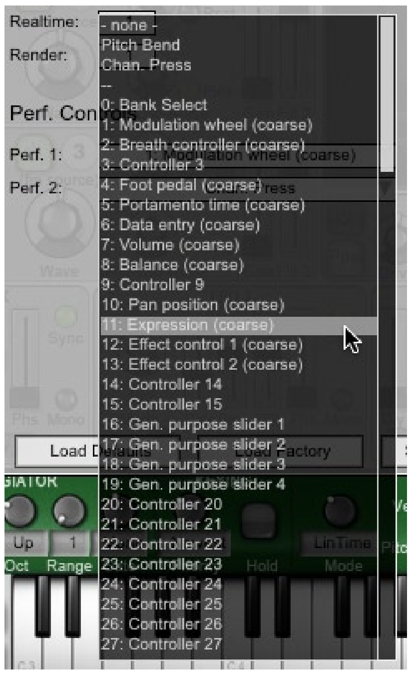
These settings relate to the oversampling in the synth engine. Higher oversampling sounds better but uses more CPU!.

Realtime

This setting relates to the oversampling used for realtime operation. Set this as high as your CPU (and the number of voices) allows.

Render

This setting is for offline rendering – non-realtime mixdowns and 'freeze' operations in your host. You can set this to very high amounts no matter what CPU you use, for stunning sound quality. However, higher settings result in longer render times!



Setting the performance controller 1 assignment via the **Perf.1** drop-down menu in the preferences panel

Perf. Controls (Performance controllers)

The **Perf.1** and **Perf.2** drop-down menus allow you to select the MIDI controller sources to use for performance controllers 1 and 2 (see section 7:4).

Managing settings

Save as Defaults
Load Defaults

Until you save defaults, the preferences panel settings only persist for the current session and return to factory defaults upon relaunching each synth. While they are saved with host sessions containing the plugins, they are *not* saved with presets!

In order for the current settings to persist in future sessions, save them as your preferred default settings using the **Save as Defaults** button.

Clicking this button also saves the current MIDI Learn assignments.

If you've changed the preferences settings and want to revert to your default settings, click the **Load Defaults** button.

Load Factory

Click this button to return to factory default preferences settings.

Apply & Cancel

Changes to Preferences settings do not take affect until they are applied with the **Apply** button. To exit the panel without making any changes, click the **Cancel** button.

6:6 Visualizer scope

Each synth in DCAM: Synth Squad features a real-time context-sensitive visualizer scope. This should not be regarded as an oscilloscope, but rather as an aid to visualising each part of the synths.

For example, when the mouse is positioned over a control in the osc section of a synth, the waveshape is displayed in the scope. The display is updated in real time as any relevant controls are adjusted.

In this example, note the waveform shape is shown in the scope as the Osc1 Wave control is adjusted.

The numerical value of the control is shown in the parameter value display.

Parameter value display

Filter sections feature a display of the filter response curve, even while filter controls are modulated. Cypher's waveshapers show the shaping curve.

Amp sections show the amp envelope shape, while individual modulators such as LFOs and envelopes show their respective shapes.

Locking and unlocking the Scope

The Scope can be locked by right-clicking (or CTRL-clicking) on a control in the relevant section and selecting the **Lock Scope** function from the parameter context menu that appears. The focus of the scope stays on the parameter and its parent section. Use the **Unlock Scope** function in the same menu to return to normal context-sensitive operation.

Parameter value display

The scope area also includes the context-sensitive parameter value display. This shows parameter values as you move the mouse over their controls, and is updated in real time as controls are adjusted.

Note that the parameter value display is unaffected by the **Lock Scope** function.

6:7 FX versions of synth plugins

All 3 synths are also provided as midi-controlled audio effect versions, in order to process external audio through the synth's circuits. Fusor is not provided as an effect version, due to the presence of licensed 3rd-party effects in its built-in FX suite.

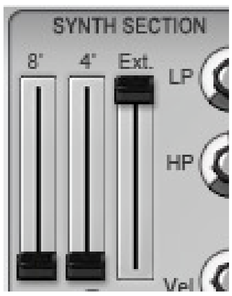
External audio is routed into the synths in place of any *white noise* sources.

After you have loaded and configured a synth FX plugin as required in your host, you must engage the synth by *playing a key!*

If you do not require the sound of the synth's oscillators alongside the external audio input, set the synth's osc controls so that only the noise source is audible. This can be achieved by turning down every osc mixer fader except the Nse fader in Amber or Strobe. In Cypher, turn the Blend control of each osc fully to the right.

Strobe's **Ext.** external audio input appears in place of the main osc **Nse** (Noise) source.

Cypher's **Ext.** external audio input appears in place of the *white* **Noise** source in each osc. The **Pink** button must be deactivated.



Amber's **Ext.** external audio input appears in place of the Synth section **Nse** (Noise) source.

6:8 MIDI Learn and host automation

MIDI Learn

DCAM: Synth Squad features an easy-to-use MIDI Learn system that gives absolute control of a parameter's initial value using standard MIDI CC (continuous controller) messages. Such messages generally have a relatively low resolution of 128 steps. While it is possible to use the pitch-bend wheel (which has a resolution of 16,384 steps) this controller is not suitable for most applications due to the fact that it is generally sprung, and is usually needed for actual pitch-bend duties.

A MIDI CC can only be assigned to a single on-screen parameter.

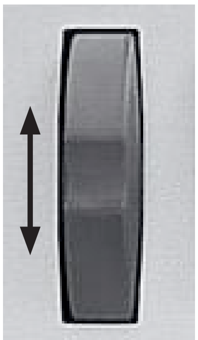
Using the MIDI Learn system

Click the **MIDI Learn** button to the left of the on-screen keyboard to enter MIDI Learn mode:

- the interface is 'locked' – controls cannot be manipulated
- all controls that can be MIDI-controlled are highlighted

Assigning a MIDI CC to a parameter control:

1. Click the parameter control.



2. Move the relevant controller, for example the mod wheel.

3. The MIDI CC is now assigned to the control. Any pre-existing assignments for the parameter or CC are removed.

Note that the CC# can only be assigned to a *single control*. When you assign the modulation wheel to the **Cutoff**, the original mod wheel assignment (to **Perf. 1**) is removed.

You can now click the MIDI Learn button again in order to exit MIDI Learn mode, or continue to make assignments.

Viewing existing assignments

All existing assignments are shown on each control in MIDI Learn mode.

Removing assignments

- Use the Clear Learn command in the control context menu (right-click or CTRL-click the control)

Saving MIDI Learn assignments for future sessions

When you make MIDI Learn assignments, they are saved with the current session but are not saved with the preset.

If you want assignments to persist for new sessions in the future, you must use the **Save as Defaults** button in the preferences panel. See section 6:5 for details of this panel.

Reverting to defaults

If you change MIDI Learn assignments but want to revert to your default assignments or the factory default assignments, use the **Load Defaults** or **Load Factory** buttons in the preferences panel.

The only assignments present in the factory defaults are the Perf1 and Perf2 controllers. See section 7:4 for more details on these controllers.

Perf 1	Mod wheel (MIDI CC #1)
Perf 2	Mono pressure (channel aftertouch)

Clearing a parameter's MIDI Learn assigment

Host automation

All synth parameters can be automated via host automation when running the DCAM: Synth Squad plugins in a host.

Host automation generally features higher resolution than MIDI CCs. It can usually be 'drawn' into tracks in the host, making it easy to quickly create parameter sweeps directly without having to assign a MIDI CC first. Dedicated hardware automation controllers for many hosts exist, but they tend to be more expensive than controllers that only send out MIDI CC messages.

See Appendix 5 for a full reference list of host automation parameters within the DCAM: Synth Squad plugins.

Note that the VST plugin specification does not allow very long parameter names (a maximum of 7 characters). Therefore, in the VST plugin versions of Strobe, Cypher and Amber, parameter names are shortened. In the VST version of Fusor, parameters are simply numbered – Par01, Par02 etc. You will therefore need to consult the table in Appendix 5 in order to use Fusor VST host automation parameters.

Full-length descriptive parameter names are available only in AU and RTAS plugin formats.

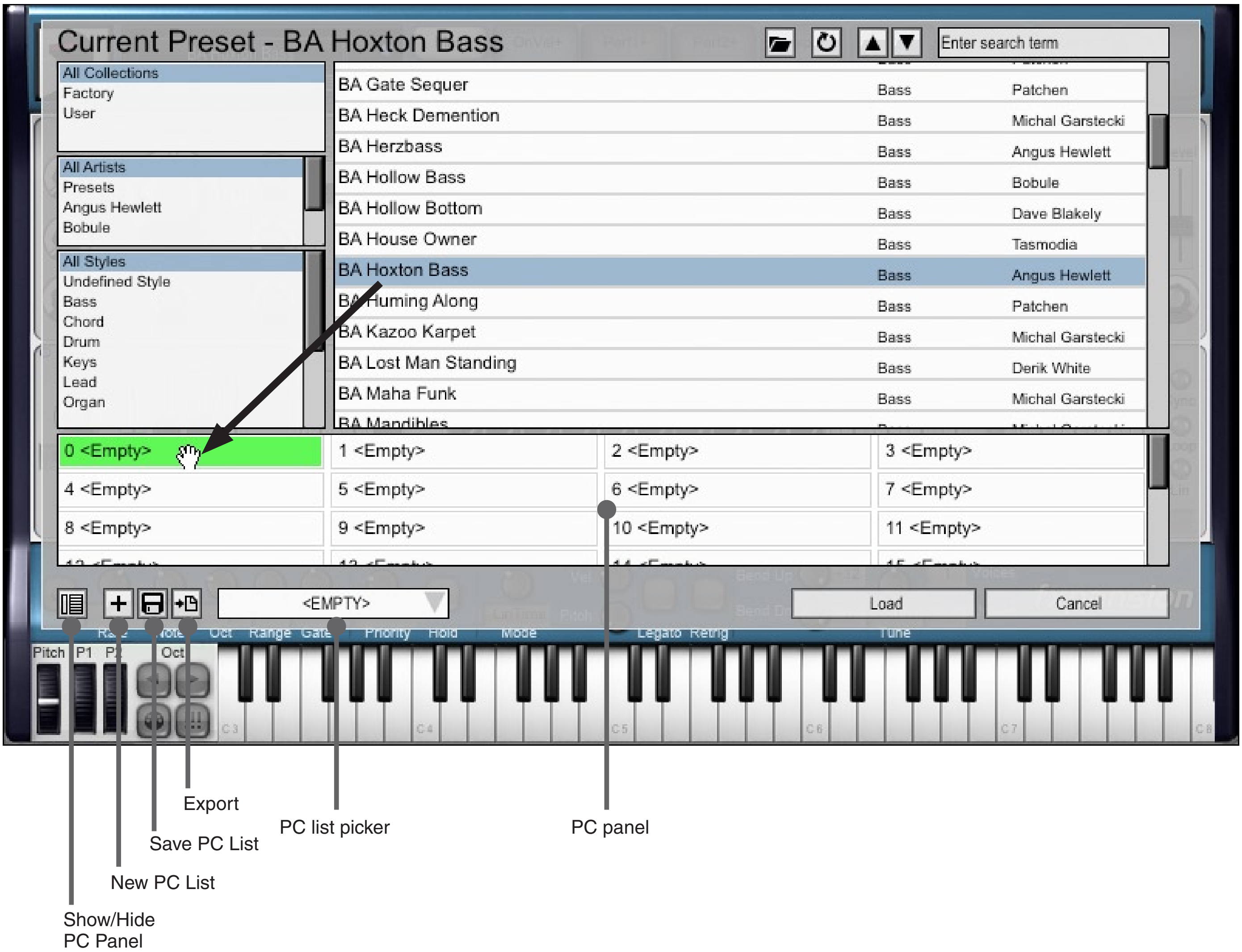
6:9 Program change panel

The program change panel, or PC panel, is opened within the preset browser, using the **Show PC Panel** button. Clicking this button introduces a scrollable list of 128 slots, numbered 0-127.

The preset browser is described in section 1:3.

Using the PC panel

Each slot corresponds to a MIDI program change number, and presets are assigned by dragging and dropping them from the main listing to a slot.



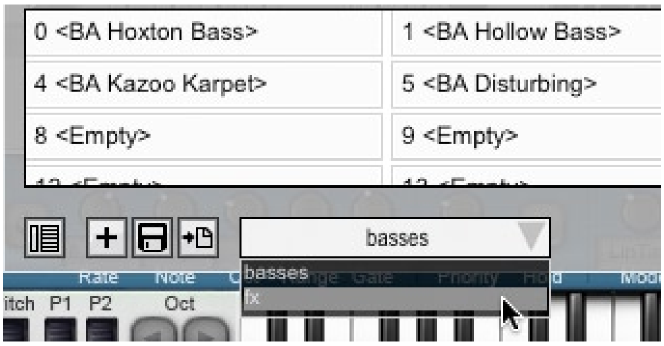
Once you have filled the slots with presets as required, you can exit the preset browser by clicking the **Cancel** button. Sending a relevant MIDI program change message results in the plugin switching to the assigned preset.

Saving program change lists

Program change lists persist for the current session, but can be saved for future use using the **Save PC List** function in the PC panel.

Loading a program change list

All previously saved PC lists can be loaded using the PC list picker. This is simply a drop-down menu of all saved PC lists. Click any PC list in the menu in order to load it.



Loading a program change list using the PC List picker

Exporting the program change list as a preset collection

The current PC list can be exported as a preset collection – this creates a new collection of presets in your user location, with the individual preset files for each preset in the PC list. The set of presets then becomes available as a collection in the preset browser's Collection filter, along with the default collections ('Factory' and 'User').

To export the PC list, click the **Export** button.

Clearing the program change list

Click the **New PC List** button in order to clear the current list, and return all program change slots to an empty state.

7: Modulation

7:1 Introduction to modulation in DCAM: Synth Squad

There are two ways of modulating parameters within Amber, Cypher and Strobe:

- **Direct modulation routings**

Some modulation sources are routed directly to certain commonly-used parameters for convenience. Some examples include LFO, Mod Envelope and keytracking depth controls for the Osc Pitch, Pulse Width and Filter cutoff in Strobe.

- **TransMod system**

The TransMod system is used to map a single modulation source to multiple parameter destinations. It also allows you to scale a source with another. There are a variety of monophonic and polyphonic mod sources available in the TransMod system – many more than those available as direct routings.

MIDI Learn and automation

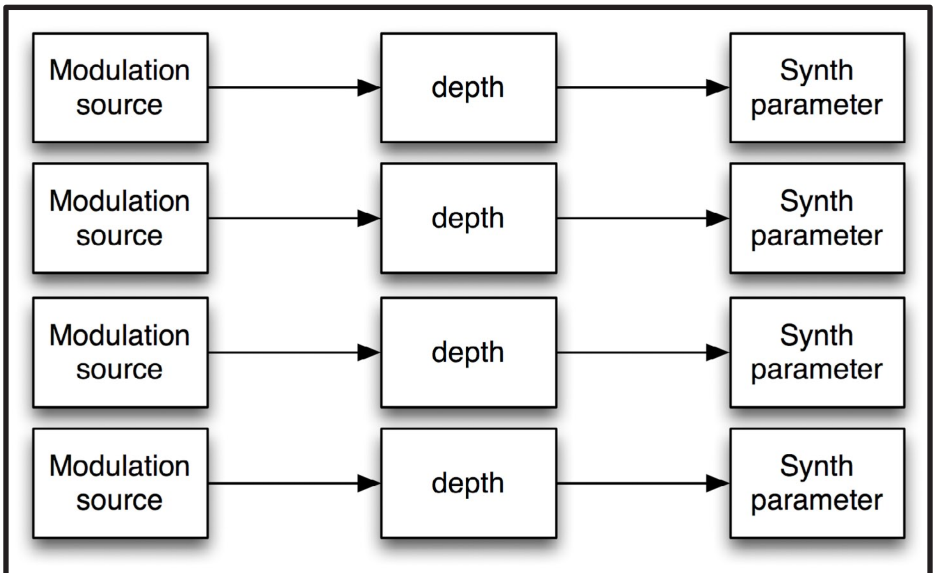
There two further ways of affecting parameters – direct MIDI Learn and parameter automation – which are discussed in section 6:8.

These are not to be considered as synth modulation, but rather as a way of automating parameters from the host or from a hardware controller.

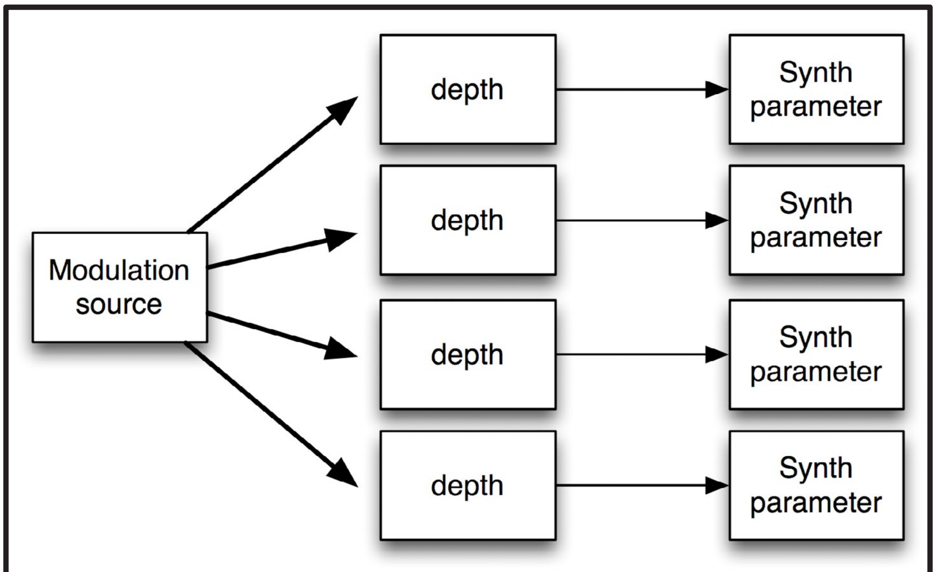
The TransMod system explained

Essentially, TransMod is similar to the well-known 'mod-matrix' found in many synthesizers. However, instead of simply connecting a modulation source to a single synthesis parameter and setting the depth or amount of modulation, TransMod allows you to route a single modulation source to multiple synth parameters, each with its own definable depth.

The following diagram illustrates how a single modulation source, such as an LFO, is routed to destination parameters in a traditional mod-matrix and in DCAM: Synth Squad's TransMod system:



Traditional modulation matrix



DCAM: Synth Squad TransMod system

7:2 TransMod system overview

Each DCAM synth features 9 different ‘views’:

Main view:	controls can set initial values of parameters
8 additional ‘TransMod slot’ views	modulateable controls that set both of the following: • initial values of parameters • modulation depths from the current slot’s mod source

Main view

The *Main view* only edits the ‘initial state’ of all the controls before any TransMod modulation. There are a few types of modulation available in this state – Strobe’s direct modulation was mentioned in the last section; Cypher features some keytracking and ModEnv routings; Amber features keytracking for the paraphonic tone filters. The Amp Envelope is always routed to the Amp parameter in Cypher and Strobe, while Amber’s Synth and Ensemble envelopes dictate the amplitude of the paraphonic notes it generates.

While it is possible to use each synth using only this main view (especially when using MIDI or host automation), the TransMod system opens up infinitely more possibilities. Cypher and Amber in particular feature very few direct modulation functions – the TransMod system is needed for most types of modulation.

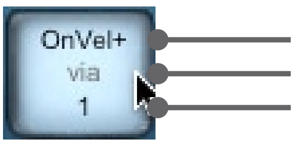
TransMod Slot views

In this example, Strobe’s first TransMod slot (assigned to key-on velocity) contains modulation for the osc **Saw** waveform level, filter **Cutoff** and amp **Level** parameters.

Clicking on a TransMod slot, or using the **Prev/Next slot** buttons to move to a slot, shows a different view of the synth. Most of the synth’s controls feature a different appearance, as they are now capable of setting and displaying modulation depths away from the initial value seen in the Main view.

Each TransMod slot can be assigned to a modulation source, which can be chosen from an extensive list of monophonic and polyphonic sources.

A TransMod slot is comprised of 3 parts:



Upper: Shows *modulation source*. Click to select and show source menu.
Middle: Click to select slot for editing (‘via’ refers to the source being scaled *via* the scaler).
Lower: Shows *modulation scaler*. Click to select and show scaler menu.

Basic TransMod system operations

Selecting a TransMod slot

Click in the middle portion of the slot (the part labelled 'via').

Clicking anywhere else in the slot selects it while also showing either the source or scaler menu.

When viewing a TransMod slot in this way, any modulated controls display their modulation depths, which can be edited on the controls themselves.

Choosing a source for a TransMod slot. This example shows the Monophonic source sub-menu (see 7:8 for more details)

Selecting a modulation source

Click on the upper segment of the slot. The source menu is shown, allowing you to choose from a range of monophonic and polyphonic sources.

See section 7:8 for full details of all available modulation sources.

Creating modulation depths

With a TransMod slot and the desired source selected, you are ready to create modulation depths for parameters. This is achieved by setting depths on the controls themselves.

Section 7:3 explains how to adjust modulation on controls.

The effect of TransMod modulation

When the modulation source varies, the values of any modulated parameters change proportionally within the defined range according to the values they receive.

In effect, this type of modulation can be considered as 'morphing' – setting minimum and maxium values, with the modulation source interpolating the parameter between these values.

Compared to a traditional mod matrix, the TransMod system offers some advantages. It is easier to keep track of large numbers of modulated parameters. There is also no abstracted, table-based relationship between depths and parameters – all depths are shown directly on the parameters themselves using the modulateable controls.

Virtually all synth parameters can be modulated using the TransMod system.

The only exceptions are the following:

- buttons
- gate source controls for Gateable Modulators (LFO, Envs etc)
- all parameters in the Arpeggiator, Keying controls, Glide controls and Settings sections, except the **Vel Glide** and **Pitch Glide** controls

Scaling

Each modulation slot in the TransMod system features an additional multiplying scaler.

By default, this is set to 1 (a numeric constant), meaning that the modulation source is mapped 1:1 with the destinations – in other words, it is unchanged by the scaling function. Destination modulation depths can be multiplied by any available monophonic or polyphonic modulation source, which is an easy way to create more complex modulation behaviours.

Selecting a modulation scaler

Click on the lower segment of the slot. The scaler menu is shown. By default, the scaler is set to '1', which means the source is unaffected.

See section 7:8 for full details of all modulation sources, which can also be used as scalers.

Choosing a scaler for a TransMod slot. This example shows the polyphonic Synth sources (see 7:8 for more details)

Scaling examples

Note-on velocity (OnVel+) is very useful when used as a scaler.

For example, you could set up a number of parameter modulation depths for a mod slot with an MRand source (individual randomization for all destination parameter modulation depths). Scaling this mod slot with note-on velocity means that higher velocity notes result in higher random modulations across the defined depths, while lower velocity notes result in smaller depths.

You can use a performance controller (assigned to aftertouch, the mod wheel or any MIDI CC) to manually control the depth of a mod source over time.

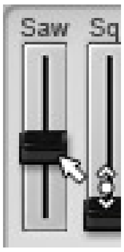
Scaling is also very useful for delaying and gradually ramping in LFO depth by multiplying an LFO source with the Ramp.

7:3 Using the TransMod system

Setting initial value of controls (Main view)

When the 'Main' view is selected, only the initial value of a control can be adjusted.

Main view

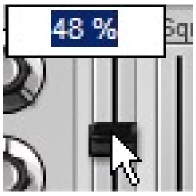


Main view:
Sliders

The initial value of a control is adjusted by clicking it and dragging up/down.

Main view:
Rotary pots

The initial value of a control is adjusted by clicking it and dragging up/down.



You can also double-click the control, type a value and press ENTER or RETURN.

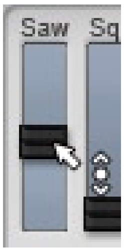
Fine control over parameters

Hold down the SHIFT key while adjusting a control for finer resolution.

Setting TransMod modulation depths

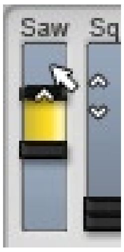
When a TransMod slot is selected, the synth's controls can be adjusted in the following ways:

Sliders



TransMod slot selected:
Setting initial value

Click the slider cap and drag up/down, or double-click to type a value and press ENTER or RETURN.



TransMod slot selected:
Setting modulation depth

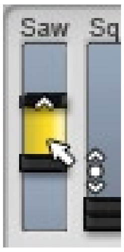
Click outside the slider cap and drag up/down.

The slider cap divides and the modulation depth from the initial value is set.

You can also move the part of the slider cap with the arrow to adjust modulation depth.

You can also double-click outside the slider cap, type a value and press ENTER or RETURN.

• When modulation already exists, 2 additional ways of manipulation are possible:

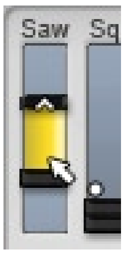


TransMod slot selected:
Setting initial value + depth

Drag the initial part of the slider cap (without the arrow):

The initial value and the modulation depth are adjusted together.

You can also double-click to type a value.



TransMod slot selected:
Setting initial value only

Hold down ALT and click/drag the initial slider cap (without the arrow):

Only the initial value is set – the modulation depth is unaffected.

Rotary pots

TransMod slot selected:
Setting initial value

Click the centre of the rotary pot and drag up/down.

TransMod slot selected:
Setting modulation depth

Click and drag the outer ring around the rotary pot to set the modulation depth from the initial value.

You can also double-click in the outer ring, type a value and press ENTER or RETURN.

• When modulation already exists, 2 additional ways of manipulation are possible:

TransMod slot selected:
Setting initial value + depth

Click and drag the centre of the rotary pot to adjust the initial value and modulation depth together.

You can also double-click to type a value.

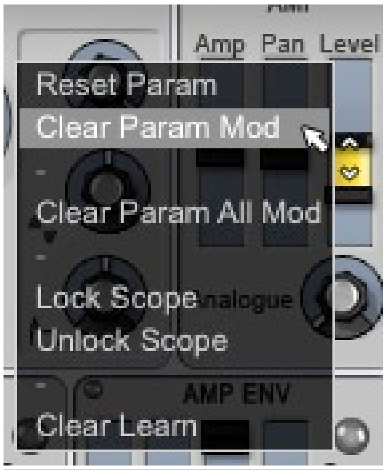
TransMod slot selected:
Setting initial value only

Hold down ALT and click/drag the centre of the pot:

Only the initial value is set – the modulation depth is unaffected.

Managing modulation with the parameter context-menu

Right-clicking or CTRL-clicking on a control brings up the parameter context menu, which contains the following functions relating to the TransMod system (see section 1:3 for details of the rest of the menu):



Parameter context menu

Reset Param

This function resets the control's initial value to its default setting.

Clear Param Mod

This function clears any modulation depths that exist for the control, in the *current* TransMod slot.

Clear Param All Mod

Using this function clears any modulation depths that exist for the control, in *all* TransMod slots.

Managing TransMod slots with the slot context menu

Beneath the items in the source/scaler menus are the slot context menu items. These can also be displayed by right-clicking in the central portion of the slot. These menu items offer a number of useful features for managing TransMod slots.

Slot context menu

Copy/Paste, Swap With...

These functions allow you to manage and interchange the contents of the TransMod Slots.

The **Copy** and **Paste** functions operate on the currently selected slot.

The **Swap with** function opens a sub-menu allowing you to interchange the contents of the current slot with any of the other 8 slots, or with the current initial values using the Main view.

Clear Mod

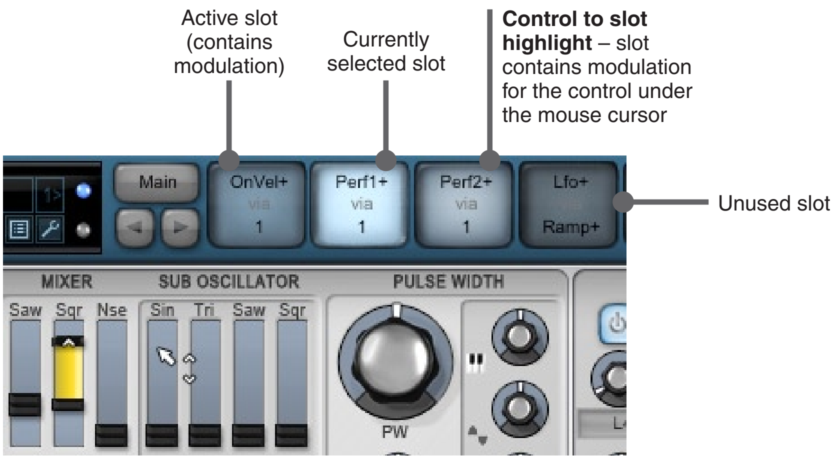
The **Clear Mod** function clears the modulation depths in the *current* TransMod Slot.

Clear All Mod

This function clears all current modulations depths in *all* TransMod Slots, so that only the initial values in the Main View affect the sound of the synth.

Mouseover indicators and slot highlighting

Whichever TransMod slot you are currently viewing, there are several ways of knowing when any controls are being modulated within other slots.



Control to slot highlight

When any control possesses any modulation depths in any TransMod slots, hovering the mouse over the control highlights the relevant slots.

Active slot highlight

When any of the 8 mod slots feature modulation depths, it is highlighted on the interface. This allows you to easily see which slots are modulating parameters.

Current slot highlight

The currently selected slot is highlighted more brightly than any others.

Slot to control highlight

Whichever TransMod slot you are currently viewing, you can also hover the mouse over any other slot to temporarily display its modulation depths.

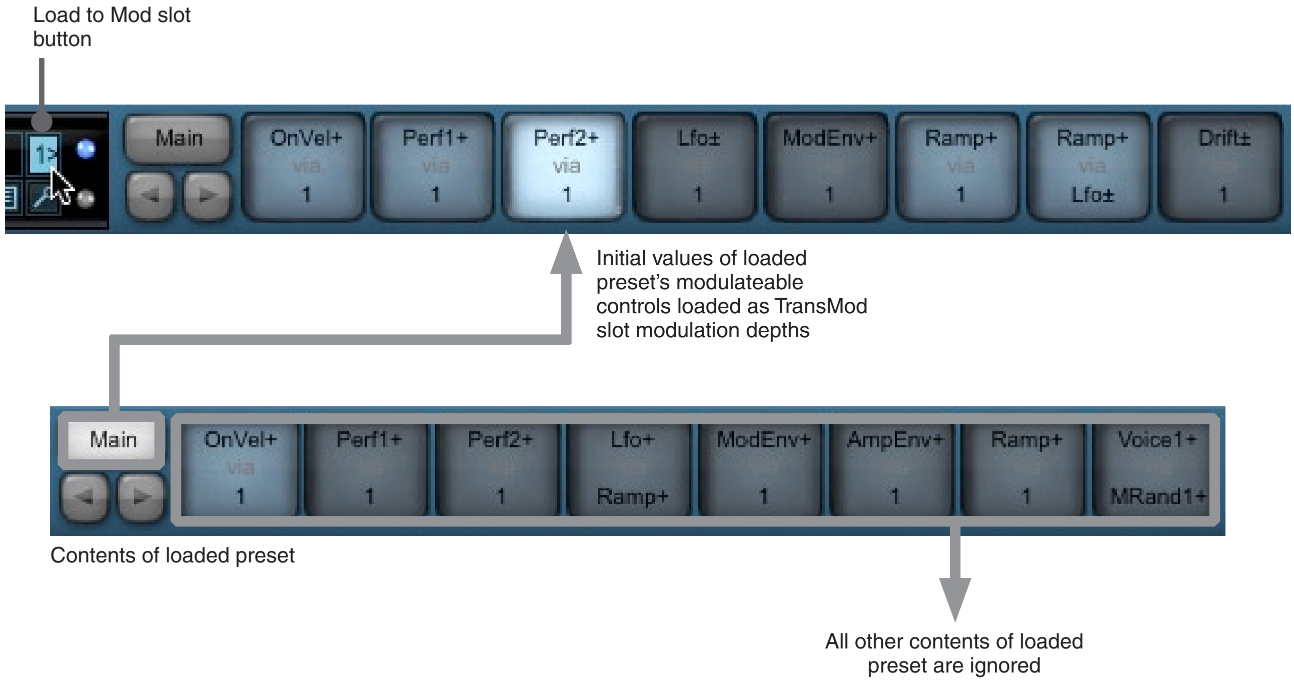
Load to Mod Slot

The TransMod system includes a feature that allows you to load the initial parameter values from a separate preset as the current TransMod slot's modulation depths.

To use this function, firstly, load a preset or create your own. Then, select a TransMod slot and enable the **Load to Mod Slot** button (located in the synth's preset picker).

Then, load a preset in any of the usual ways (described in section 1:3).

The initial values of all controls (those seen in the Main view) are loaded as the new modulation depths for the selected TransMod slot. All other data in the loaded preset (its own TransMod modulation depths and all initial values of non-modulated controls) are ignored.



Patch-morphing effects

Any non-modulateable parameter settings are not translated into modulation depths, so this feature does not represent full patch morphing. However, it allows most aspects of patch morphing while avoiding non-useful morphing parameters and clicks through button modulation. It is still possible to experience clicking during TransMod modulation – the main possible cause of this is modulation of the following parameters:

- Filter mode
- Changing LFO shapes
- Sub-osc octave parameters (Strobe)

There are other possible ways that clicks can occur in sounds – with stepped/square LFO shapes, fast envelope attack/release settings and so on – but these are not due to the TransMod system.

Usage hints

The **Load to Mod Slot** function puts incredibly dynamic sounds within easy reach, allowing wild fluctuations in parameters when altering just one modulator. You can load up to 8 presets into the available TransMod slots and modulate them all in real time, resulting in a staggering variety of complex new sounds.

Some useful modulation sources for use with this function are Voice, Unison and MRand, which can produce richly varied and chaotic performances featuring different parameter settings for each voice. Also, try using a performance modulator like aftertouch, mod wheel or velocity as the modulation source.

Be warned that it in many situations, it is unlikely that the same sound will ever be heard more than once when using the synths in this way. Therefore, you may want to constantly record an instrument's output to capture the performance.

7:4 Velocity and Performance controllers

Velocity and performance controllers are special cases in DCAM: Synth Squad. They are the default sources for the first 3 TransMod slots in each supplied preset, based on the philosophy that most people possess 3 main performance controllers.

The first of these is MIDI key velocity, with the others typically being two of the following:

- modulation wheel
- aftertouch
- expression pedal
- breath controller

Velocity and performance controllers are arranged as follows in all presets:

TransMod slot 1	OnVel+: Key-on Velocity
TransMod slot 2	Perf1+: Performance controller 1
TransMod slot 3	Perf2+: Performance controller 2

Velocity

This is probably the most immediate control available to anyone with a MIDI keyboard. It requires no extra features or hardware, and is available with ‘digital piano’ keyboards, which usually lack even a mod wheel.

It is also polyphonic! The TransMod system allows you to modulate any parameter with velocity, meaning that incredibly dynamic sounds are possible. If you scale it through a random source, perhaps using the Load to Mod slot function, you can create some incredibly chaotic effects.

Velocity Glide

This is a special type of glide function that smooths transitions to the modulation depths defined in slots driven by the OnVel+ source (this source is assigned to slot 1 by default). Velocity glide is an additional way of creating glide effects using a very accessible polyphonic performance modulator – keyboard velocity. See sections 6:3 and 7:6 for more details.

Performance controllers

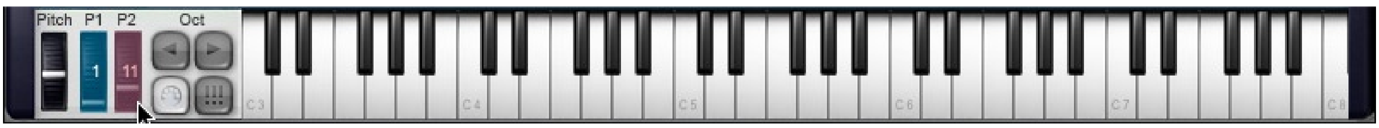
Performance controllers are monophonic – they are applied to all voices. They are represented on the interface to the left of the on-screen keyboard, but they are intended to be assigned to hardware controllers.

By default, the performance controllers are assigned as follows:

P1 (Perf 1)	Mod wheel (MIDI CC #1)
P2 (Perf 2)	MnPress (mono pressure, or channel aftertouch)

You can change these assignments to any other source such as pitch bend, breath controller, expression pedal or any MIDI CC. In the preferences panel, select from a drop-down menu of possible sources for each of the 2 performance controllers. See section 6:5 for more details of the preferences panel.

You can also assign performance controllers using the built-in MIDI Learn system. See section 6:8 for details of how to use MIDI Learn. The on-screen controls representing the Perf1 and Perf2 controls are shown to the left of the keyboard and to the right of the pitch-bend wheel.



Changing the **P2** (**Perf 2**) assignment using MIDI Learn.

If you change these assignments (using either method), it is recommended that you save the preferences settings as defaults (using the **Save as Defaults** button in the preferences panel) so that they remain active for future sessions. The preferences are unique for each synth, so you can have different assignments for different synths if you have a suitable array of hardware controllers. Fusor’s own assignments override those of the synths when hosting them.

7:5 Gateable modulators

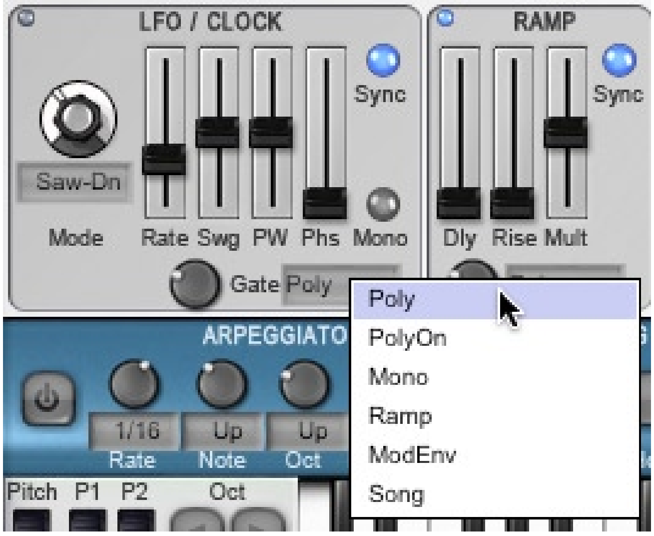
The term 'gateable modulators' in DCAM: Synth Squad refers to LFOs, Ramps and Envelopes. LFOs and Ramps are actually 'triggered' rather than gated for a variable duration – they are called gateable modulators here in order to simplify the interfaces of the synths.

Gateable modulators can be gated or triggered with a number of selectable gate sources, and even with each other to create very complex modulation effects.

While some gateable modulators feature direct routings and depth controls for commonly used parameters in some of the synths, they are all available as TransMod sources to create more complex sounds. Inverted envelopes and ramps are also available.

Types of gating behaviour (Gate control settings)

Note that the **Gate** control is not labelled on the Ramp modulator.



Setting the Gate source for the LFO in Strobe.

Poly

This setting represents normal polyphonic gating mode.

The gating behaviour depends on the state of the **Retrig** button in each synth's Glide controls section.

PolyOn

This gating mode disregards the **Retrig** button setting entirely.

When using this mode, a gateable modulator is always gated or triggered by key-on events for each voice.

Mono

In this mode, a gateable modulator is gated/triggered by the first key-on when all notes are off – subsequent keys that are played without releasing all keys do not re-gate the modulator.

This mode is especially useful when using the arpeggiator, to use an envelope or LFO over the sequence of notes rather than on each note. It can also be used when you need an LFO to modulate parameters of all voices together, rather than modulating each voice independently of the others.

Song

The main function of the Song gate type is for providing a 'free-running' LFO which is not triggered by key-on events or other modulators.

When playback is started in your host, the LFO is triggered and runs freely afterwards. This allows free-running behaviour but *with repeatability* – each time you play through your song it sounds identical.

When playback is stopped in your host, the LFO simply runs freely, continuously. In such a situation, it behaves just like a standalone synth that features a free-running LFO.

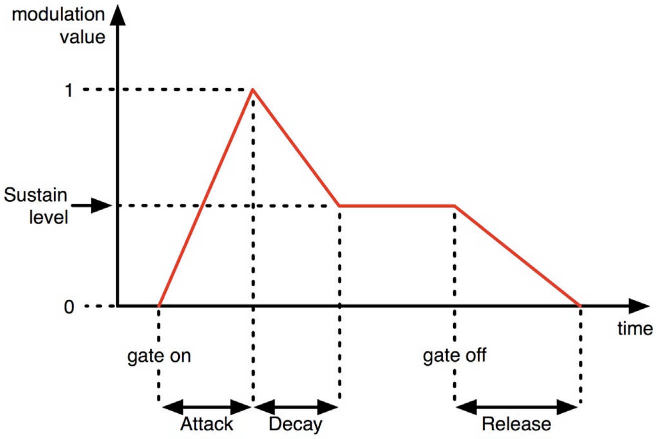
You can use this gate type as a gate source for the Ramp, ModEnv and AmpEnv modulators. This is to achieve a number of purposes – for example, when using looped envelopes as substitute LFOs, you may want the phase to be dictated by song playback. It can also be useful for creating very long envelopes for drone sounds over the length of a song.

Other gateable modulators

Each of the gateable modulators can be gated or triggered by each other – the gating or triggering occurs when the gate source reaches a value of 1.

For example, if you set the ModEnv to be gated by the LFO, the ModEnv is retriggered every time the LFO reaches its highest point (1).

Envelope



An envelope allows you to shape a sound over time, from its beginning to its end. It is sometimes called a 'contour'.

The Amp Envelope is directly routed internally to the **Amp** parameter in Strobe and Cypher. This is to avoid needing to make a dedicated modulation connection between them purely for engaging the amp so that notes can be heard.

Amber's paraphonic architecture means that it doesn't have a conventional amp envelope, but dedicated Synth and Ensemble envelopes that dictate the amplitude shape of each paraphonic note. These envelopes are special cases, and are explained in section 5:9.

Mod Envelopes, meanwhile, can be used for a variety of purposes. They are similar to the second envelope typically found on many monophonic and polyphonic synths, which were usually routed to the filter cutoff or oscillator pitch. The TransMod system allows you to route Mod Envelopes to any synth parameter.

ADSR controls

Atk (**Attack**), **Dcy** (**Decay**) and **Rel** (**Release**) are time periods, while **Sus** (**Sustain**) is a level, expressed in %.

When the envelope is gated, the following processes occur:

- The envelope level rises from 0 to 1 over the defined **Attack** time
- After this has been reached, it decays towards the level defined by the **Sustain** control, over a time period defined by the **Decay** parameter (if Sustain is at 100%, there is effectively no Decay stage).
- This occurs as long as the envelope is gated (while the gate signal is 'on'). Whenever the gate is released (when the gate signal returns to an 'off' state), no matter which stage has been reached, the envelope's level falls to 0 over the time defined by the **Release** parameter.

All of these parameters can be modulated with the TransMod system.

Sync

With the **Sync** button enabled, the time-based envelope controls (**Attack**, **Decay**, **Release**) are set in BPM units. With the Sync button disabled, these controls operate in seconds.

Loop

Enabling the **Loop** button causes the envelope to repeat after it has completed the **Attack** and **Decay** phase. After the Decay time is complete, the envelope starts again - this continues while the envelope is gated. If the **Sustain** control is higher than 0, the envelope travels towards the sustain level and remains at this level for the duration of the Decay time.

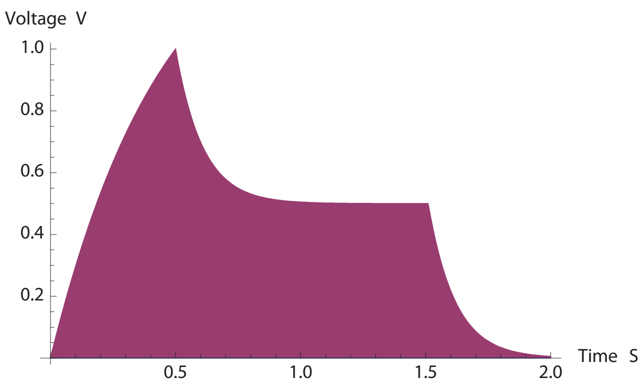
When the gate is released, the level of the looping Attack and Decay stages is scaled down to 0 over the **Release** time.

The level of the envelope returns to 0 when it is gated, and at the end of each 'loop' (after each decay stage).

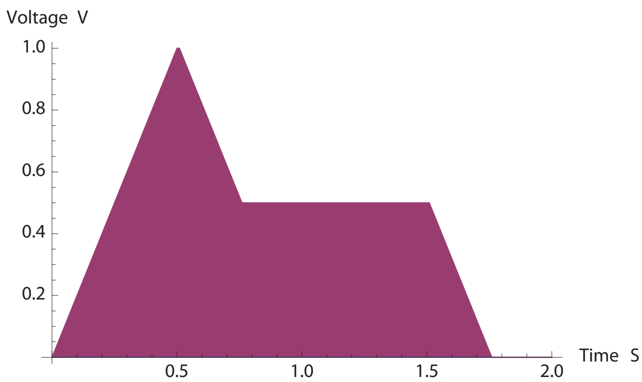
The Loop function is great for simulating echo/delay effects on gate release or for providing an additional LFO capable of alternative shapes.

Lin (Linear)

By default, the **Attack**, **Decay** and **Release** phases of envelopes react exponentially. When the **Linear** button is enabled, their behaviour changes to a linear response. The difference between the two is illustrated by the following diagram:



Exponential envelope (**Linear** button off)



Linear envelope (**Linear** button on)

LFO

Rate & Sync

The **Rate** parameter controls the rate or speed of the LFO's cycle. If the **Sync** button is enabled, BPM-based units are used. When the Sync button is disabled, the Rate control is adjusted in Hz.

The state of the Sync button also dictates how Cypher and Strobe's built-in arpeggiator Rate is set.

Phs (Phase)

The **Phase** parameter allows you to adjust the phase of the LFO between 0 and 360 degrees. Phase can be modulated in real time, and always resets when the LFO is gated.

Mode

The **Mode** control selects the LFO shape and can be modulated via the TransMod system. Note that modulating this control is liable to result in audible clicks.

The following 'standard' LFO shape modes are available:

- Sine
- Cosine
- Tri-S
- Tri-C
- Arc-S
- Arc-C
- Saw-Up
- Saw-Down
- Square

A number of other LFO shapes are available:

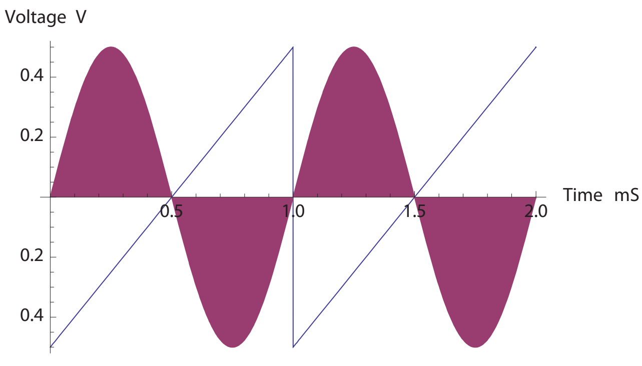
• Stab-White • Stab-Pink • Stab-Brown	The various noise types provide random points to which the LFO 'jumps'. After it reaches the next point, it drops down vertically to the position of the previous point. This process then repeats. The end result is a series of sharp 'stab' shapes.
• SnH-White • SnH-Pink • SnH-Brown	These modes are sample+hold values taken from noise sources. They create 'steppy' sounding random LFOs, because each value stays constant until the next value, at which point the LFO immediately travels vertically to the new value.
• Ramp-White • Ramp-Pink • Ramp-Brown	In these modes, the LFO ramps up or down to the next value – the end result is somewhat like a complex triangle shape but with random amplitudes.
• White noise • Pink noise • Brown noise	These are straightforward noise sources for use as an LFO, that output noise values continuously at control-rate.

Mono

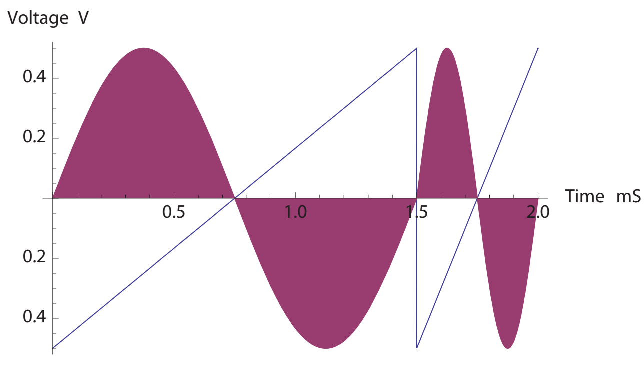
Enabling this button forces the LFO into monophonic mode. In monophonic mode, there is only 1 global LFO instead of an LFO for each voice.

Swg (Swing)

This is an unusual control for an LFO to possess. It creates a 'swung' feel by changing the length of each part of 2 LFO cycles so that each cycle has a different length, but their combined length remains the same.



Sine LFO, Swing 0

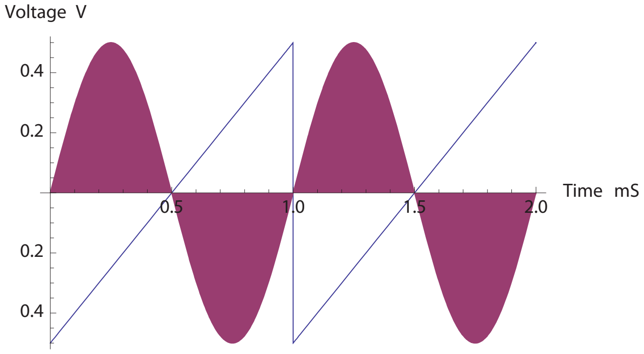


Sine LFO, Swing 50%

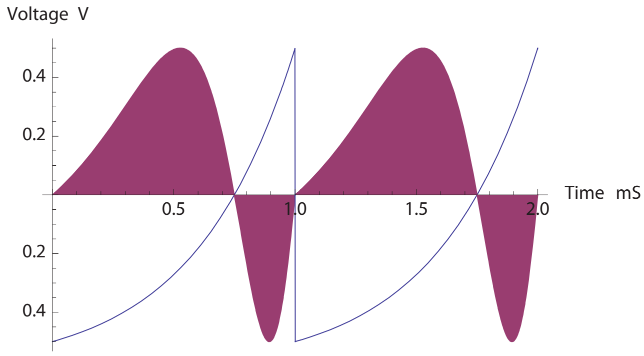
PW (Pulse Width)

Again, you probably won't have seen a pulse width control on an LFO before. This function changes the curve of each LFO cycle in a similar way to changing the pulse width on a pulse oscillator. It works on all shape modes.

The following diagram illustrates how it works on a sine shape:

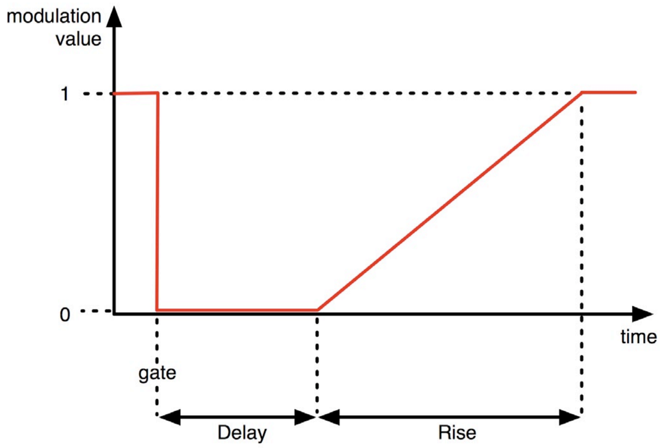


Sine LFO, Pulse Width 0



Sine LFO, Pulse Width 75%

Ramp



The Ramp generator is a versatile gateable modulator that is intended for use with the TransMod system or as a gate source for other gateable modulators.

When triggered, its value drops immediately from 1 to 0. Then, after a period of time defined by the **Delay** parameter (during which its value remains at 0), it rises to 1 over a period defined by the **Rise** parameter.

It can be seen as a simple inverted envelope with an instant attack and a variable decay, although it is triggered like an LFO rather than gated like an envelope, and has a definite period, rather than an indefinite period of sustain.

These are the main intended uses for the Ramp:

- A delay before the LFO is triggered – set the LFO **Gate** to Ramp, and use the Ramp's **Rise** time to set a delay time
- A simple, fixed decay envelope using the inverted Ramp as a TransMod slot source (the **Rise** control sets the decay)
- A ramp-up scaling function – use it as a scaler in a TransMod slot to scale an LFO over time (Rise) from 0 to 1

  You can also use this method to set a delay on an LFO, using the Ramp's **Delay** parameter.

Note that that the **Gate** control is not labelled on the Ramp.

Dly (Delay)

The **Delay** parameter sets the delay after the ramp value drops from 1 to 0, and before it begins rising to 1. It is scaled by the value of the **Mult** control.

Rise

The **Rise** control adjusts the time taken for the ramp to rise from 0 to 1. It is scaled by the **Mult** control.

Mult (Multiplier)

This control is a multiplier for the Ramp's **Delay** and **Rise** times. By default, it is set at 100%, meaning that the Delay and Rise times occur exactly as set on the respective controls. The **Mult** control allows you to multiply these times between 0% and 200%.

Sync

With the **Sync** button enabled, the **Delay** and **Rise** times are set in BPM units. With the Sync button disabled, these parameters are set in seconds.

7:6 Glide, voices and unison

Glide

All the synths in DCAM: Synth Squad feature 2 types of Glide.

Pitch (Pitch Glide)

The **Pitch Glide** control sets the glide time towards the pitch of new note events.

This is the Glide control you will have seen on virtually all classic analogue monosynths. It smooths pitch control signals from the keyboard, meaning that the pitch that you hear 'glides' up or down to the next note.

Vel (Velocity Glide)

The **Vel Glide** control sets the glide time towards new OnVel modulation depths on each key-on event.

It utilizes the fact that key-on velocity is one of the most immediate performance controllers on modern controller keyboards. Velocity Glide offers a means of injecting life into a synth part without having to manipulate any additional performance controller.

For example, if you set up the OnVel+ TransMod source to increase the filter cutoff, playing a harder note after a soft note would cause the cutoff frequency to glide to the higher value over the **Vel Glide** time period. If the Vel Glide control is set to zero, the cutoff frequency is fully modulated as soon as the new harder note is played.

Modulation of Glide times

Pitch Glide and **Vel Glide** times can be modulated using the TransMod system – they are the only controls in this area of the synths' interfaces that can be modulated.

Other parameters that affect Glide

As well as the **Pitch Glide** and **Vel Glide** time controls, several other controls in the Glide controls section also have an effect on Glide behaviour (see sections 6:3 and 7:6 for full details of the Glide controls area of the interface).

• Legato

If enabled, glides occur only when 2 notes overlap (often known as 'fingered portamento'). If it is disabled, glides always occur regardless of how the input notes are positioned.

• Retrig

The **Retrig** button allows you to re-gate envelopes and other modulators during legato pitch glides.

• Mode

The Glide **Mode** control changes the glide response between 'LinearTime', 'LinearRate', and 2 exponential settings.

Using Glide in Amber

Amber's paraphonic architecture means that voice-based pitch glide is not possible during conventional 1-voice paraphonic usage. This is because all notes in the keyboard range are constantly generated at exact pitch.

However, you can still use glide in the TransMod system for modulating other parameters polyphonically (monophonic for each voice). Use the Pitch TransMod source for utilizing keyboard pitch glides as a control signal.

Voices and Unison

Voice sources allow you to set up polyphonic modulation effects that vary according to the number of active voices. Active voice counts are converted into a set of values from -1 to 1 (bi-polar) and 0 to 1(uni-polar) . Each source is provided in 2 different varieties of value distribution between the minimum and maximum.

The Unison sources allow any parameter to be varied across unison voices – not just the pitch detuning found in classic synths. The Alt sources output alternating values for each voice used in sequence.

Voice card manipulation in classic polysynths

Certain Oberheim polysynths such as the OB-8 featured trim pots for the pan position of each voice, and many polysynths with individual voice cards had trim pots on each card that could be manipulated to detune each voice, for example. Unison and voice modulation in the TransMod system allows you to go way beyond these techniques to create some startling effects, full of movement and variation. You can even mimic the kind of voice variation found on old Oberheim SEM-based polysynths, which allowed the programming of a totally different patch for each voice.

Examples for using voice and unison TransMod sources

- Modulate the **Pan** control for a stereo spread of voices during chords or unison notes.
- Subtle voice and unison modulation on envelope, filter, osc pitch and other controls can result in more variation during performances, and emulate the slightly uneven, unpredictable feel of a vintage synthesizer with slightly differing voice cards.
- Use the **Load to Mod Slot** function in a slot with a voice or unison source for wildly varied, dynamic sounds.
- Modulate **Pitch Glide** and **Vel Glide** times with a voice source to give each voice its own glide time.
- Use voice, unison and alternating sources as scalers on other modulation sources to inject more variation into them.

7:7 Tracking the keyboard

Pitch keytracking

It is usually vital to modulate parameters according to the pitch of keyboard input. This type of modulation is used on oscillator pitch to create a musically useful effect according to the keys played – in other words, to scale notes up the keyboard in semitones. It can also be used to 'play' the frequency cutoff of a filter, which is especially useful when using a self-oscillating filter.

Keytrack controls in Strobe

Keytrack depth is provided as a dedicated control for several parts of Strobe – for the osc **Pitch**, **Pulse Width** and filter **Cutoff** – in order to facilitate quick and easy programming.

Keytracking controls in Cypher

Cypher features Keytrack on/off buttons for each of its 3 oscs, with each of its 2 filters also providing a dedicated variable **KeyTrack** depth control.

Keytracking in Amber

Due to its paraphonic nature, Amber does not feature any dedicated keytrack controls. It does, however, include hard-wired, non-adjustable keytracking of its 1-pole paraphonic tone filters.

Amber's paraphonic note generation reacts to the keyboard in a different way to a conventional mono/poly synths. The pitch of all notes on the keyboard is effectively fixed and generated constantly (although their pitch can be changed together using the master tuning controls). Notes are heard when the keyboard gates each note's individual envelopes and VCAs.

If you're using more than 1 voice, keytracking can be used polyphonically (monophonic for each voice) by using the Pitch source in the TransMod system.

Using keytracking in the TransMod system

Keytracking can be used in the TransMod system via the Pitch modulation source (found in the Poly source sub-menu).

The Pitch modulator is simply a value that increases by 1 over each octave.

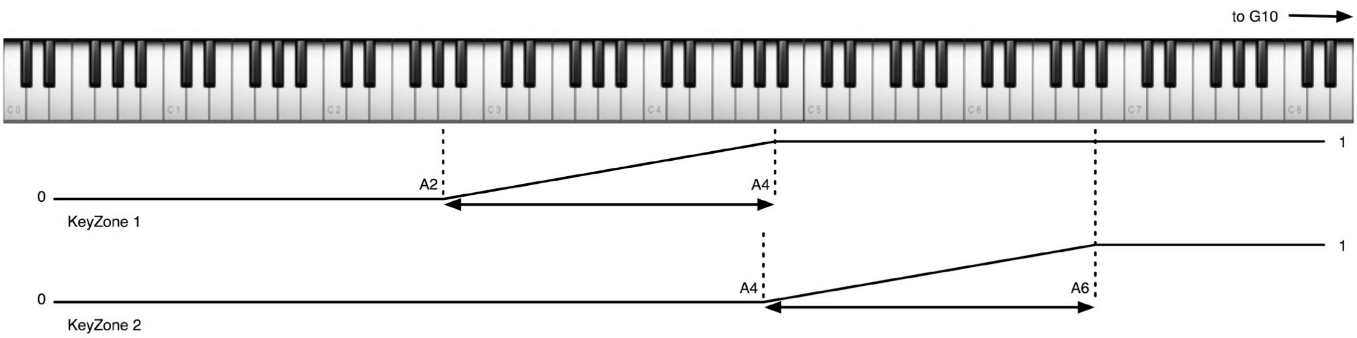
Therefore, keyboard input can be used as a TransMod source and scaler for affecting any synth parameter polyphonically.

Keyboard zones

Keyboard zones (keyzones) are intended to be used alongside the TransMod system's **Load to Mod Slot** function (see section 7:3), in order to create 2 keyboard zones for 2 different patches. There is also a 2-octave range between these zones where 1 patch can 'crossfade' into another.

There are 2 keyzones available: Keyzone1 and Keyzone2. These modulators output a value of 0 below a certain key, a value between 0 and 1 within a 2 octave keyrange above this key, and a value of 1 for all keys above this 2 octave keyrange.

The following diagram illustrates the two available Keyboard zones:



Here is an example for setting up Keyzone1:

1. Load a preset and set Keyzone 1 as the source for a TransMod slot.
2. Enable the **Load to Mod Slot** button.
3. Load another preset.
4. Play the keyboard range from A2 to A4, as well as keys above and below this range. You will hear the sound change between the 2 presets within the A2 to A4 range.

7:8 Summary of TransMod sources

DCAM: Synth Squad features a wide variety of interesting modulation sources to use with TransMod slots (either as sources or scalers).

In the list below, sources are denoted as follows:

+	**Uni-polar** / uni-directional (only positive values: 0 to 1)
±	**Bi-polar** / bi-directional (positive & negative values: -1 to 1)
-	**Inverted** (this is only for envelopes and ramps)

Monophonic sources

Monophonic TransMod sources operate globally on all active voices – they do not operate individually for each voice.

Most of them are continuous MIDI messages such as MIDI CCs, mono pressure (channel aftertouch) or mod wheel. There are also a number of numeric constants, a set of mono gates and song-phase ramps.

• Numeric constants 0, 0.1,0.5, 1, 2, 10, -1	Numeric constants are very useful scalers. 1 is the default scaler for all TransMod slots, meaning that each source is unchanged before modulating the destination parameters.

Mono source sub-menu

Common MIDI controllers

These are provided to facilitate direct control of a TransMod slot from common MIDI performance controllers.

DCAM: Synth Squad also provides the dedicated Perf1 and Perf2 controllers, which are intended to be your 2 main performance controls. They are defined using the preferences panel in each DCAM plugin (including Fusor). Each plugin features its own set of defaults. Performance controllers are covered later in this section.

• Mod+	Modulation wheel – MIDI CC #1
• Breath+	Breath controller – MIDI CC #2
• Expr+	Expression – MIDI CC #11
• MonoBend±	Pitch bend
• MnPress+	Mono pressure (channel aftertouch)

Monophonic gate/trigger sources

These sources simply output a value of 1 or 0 depending on several conditions.

Normally, gate and trigger sources are used for various types of gateable modulator behaviour. However, they are also provided as TransMod sources for complex modulation operations, or as a way of achieving quick pulses for filter stabs and so on without having to use an envelope.

• MnGate+	Outputs 1 if one or more keys are held down, otherwise 0
• MnOnGate+	Outputs 1 if one or more keys are held down, but is retriggered on every key-on (output returns to 0 and immediately to 1)
• MnTrig+	Outputs 1 then immediately 0 on key-on when a note is played after all notes are off
• SongGate+	Outputs 0 if the host song is not playing, and 1 if it is playing – this gates on the nearest beat

Song phase

These sources provide a saw-up ramp synchronized to the host tempo. Normally, song phase is used for mono free-running gateable modulators with repeatable behaviour, but they are also provided in the TransMod system for extra mono LFOs synced to 1 or 4 beats of the host tempo.

• Phase1±	Song phase: a saw-up ramp from 0 to 1 per beat, based on song position
• Phase1+	As above, but uni-polar
• Phase2±	Song phase: a saw-up ramp from 0 to 1 over 4 beats, based on song position
• Phase2+	As above, but uni-polar

Perf source sub-menu

This sub-menu contains the two performance controllers, shown to the left of the pitch-bend wheel on each synth's on-screen keyboard area. Each of these have an assigned monophonic hardware MIDI controller source, which is defined in each synth's preferences panel.

The hardware assignments for these controls can be altered via MIDI Learn or the preferences panel. See section 7:4 for details.

All presets supplied with DCAM: Synth Squad use these performance controllers in TransMod slots 2 and 3 (alongside key-on velocity in slot 1). It is highly recommended to keep to this standard, especially if you intend to share your own presets with others.

Perf1+	Performance controller 1: default source for TransMod slot 2 Factory default assignment: modulation wheel (MIDI CC #1)
Perf2+	Performance controller 2: default source for TransMod slot 3 Factory default assignment: MnPress (mono pressure or channel aftertouch)

Polyphonic sources

Polyphonic modulation occurs for each voice. For example, all the main modulators on Strobe's interface, such as the LFO, Ramp and envelopes, are polyphonic – there is a separate, distinct LFO, Ramp and 2 envelopes for each voice that is playing.

TransMod sources are not limited to LFOs, envelopes and ramps: random and voice/unison sources are available, amongst many others.

Poly source sub-menu

Polyphonic keyboard input

• OnVel+	Key-on velocity: default source for TransMod slot 1 Modulate the **Amp** parameter with this source in order to vary amplitude with velocity. The **Vel Glide** control creates glide times to new notes' OnVel modulation values, allowing a further expressive element to your playing.
• OffVel+	Key-off velocity
• Pitch±	Derived from keys played on the keyboard – increases by 1 for each octave going up the keyboard It can be used to harness polyphonic keytrack and pitch glide modulation for any parameter.
• Press±	Poly pressure (polyphonic aftertouch) Polyphonic aftertouch keyboards are very rare, but the source is provided for use in the TransMod system if you do possess one.
• KeyZone1+	Keyboard zone 1: 0 at A2 and lower, 0.5 at A3, 1 at A4 and higher
• KeyZone2+	Keyboard zone 2: 0 at A4 and lower, 0.5 at A5, 1 at A6 and higher

See section 7:7 for details of the Keyzone sources.

Polyphonic gate/trigger sources

These are very similar to the monophonic gate/trigger sources, except that they operate polyphonically (they can occur multiple times depending on the number of available voices).

They simply output a value or 0 or 1 depending on various conditions.

Normally, gate and trigger sources are used for various types of gateable modulator behaviour. However, they are also provided as TransMod sources for complex modulation operations (try using them as scalers), or as a way of achieving quick pulses for filter stabs and so on without having to use an envelope.

• PlGate+	Outputs 1 if one or more keys are held down, otherwise 0 Retriggering occurs according to the state of the **Retrig** button (returns to 0 and immediately to 1).
• PlOnGate+	1 if one or more keys are held down, but retriggers on every key-on (returns to 0 and immediately to 1)
• PlTrig+	This is the same as PlGate+, except that it outputs a 1 and then returns to 0 immediately when keys are played.

Alt (alternating voice) sources

These output a different value for each alternating voice played via the same note on the keyboard. This allows you to enliven parts containing sequential notes of the same pitch by varying parameters for each.

• Alt1±	Alternating source per key-on: -1, 1, -1, 1...
• Alt1+	As above but positive only: 0, 1, 0, 1...
• Alt2±	Alternating source per key-on: -1, 0, 1, -1, 0, 1...
• Alt2+	As above but positive only: 0, 0.5, 1, 0, 0.5, 1...

Voice and unison sources

Voice and unison sources allow you to set up polyphonic modulation effects that vary according to the number of active voices. Active voice counts are converted into a set of values from -1 to 1 (bi-polar) and 0 to 1(uni-polar) which can be used to modulate parameters. Each source is provided in 2 different varieties of value distribution between the minimum and maximum. The Unison sources allow any parameter to be varied across unison voices – not just the pitch detuning found in classic synths.

See section 7:6 for more details of these sources.

Unison sources

These sources are used in conjunction with polyphonic Unison voices. Unison modulation sources proportionally distribute values within a range for each operational unison voice.

Modulating the Pan control with this source produces a rich stereo spread of unison voices.

• Unison1±	Each active unison voice outputs a value proportionally distributed between -1 and 1 This source is especially suited to panning voices across the stereo field.
• Unison1+	As above, but uni-polar
• Unison2±	As Unison1 but distributed in a different order This distribution can be more suitable for unison detuning with osc pitch modulation.
• Unison2+	As above, but uni-polar

Voice sources

These sources spread values proportionally within a range for each active voice, when using multiple voices with or without unison.

• Voice1±	Each active voice generates a value proportionally distributed between -1 and 1
• Voice1+	As above but uni-polar
• Voice2±	As Voice1 but distributed in a different order
• Voice2+	As above but uni-polar

Random source sub-menu

The TransMod system contains a range of polyphonic random modulation sources of various types, from simple noise sources to more specific ways of generating random values.

• Noise1±	Fast-varying white noise random value
• Noise1+	As above, but uni-polar
• Noise2±	Fast-varying pink noise random value
• Noise2+	As above, but uni-polar
• Rand1±	A single random value for all destination parameters, generated by key-on sample-and-hold on white noise
• Rand1+	As above but uni-polar
• Rand2±	A single random value for all destination parameters, generated by key-on sample-and-hold on pink noise
• Rand2+	As above but uni-polar
• MRand1±	Individual random values for all destination parameters (multiple random values), generated by key-on sample-and-hold on white noise This source is useful for modulating each note differently, especially when using the ‘Load to Mod slot’ function for loading presets as modulation depths.
• MRand1+	As above but uni-polar

• MRand2±	Identical to MRand1± but using pink noise
• MRand2+	As above but uni-polar
• VMRand1±	A random number per voice, per parameter, that is generated at synth load time, generated using white noise This source is useful for having a set of random values that stays constant throughout the current session.
• VMRand1+	As above but uni-polar
• VMRand2±	Identical to VMRand1± but using pink noise
• VMRand2+	As above but uni-polar
• Drift±	Slowly changing random LFO This can be used to simulate subtle parameter drift in hardware analogue synths – for example. small amounts of modulation on osc pitch can simulate drifting VCOs.

Synth source sub-menu

LFOs

Each available LFO is presented as 3 sources in the TransMod system:

• LFO±	Bi-polar LFO
• LFO+	Uni-polar LFO
• LFOGate+	Gate output of lfo (square wave)

Envelopes & Ramps

Each available Envelope and Ramp is presented as 2 sources in the TransMod system. This includes all Mod Envelopes (Env), Amp envelopes (AmpEnv) and the Synth/Ensemble envelopes in Amber.

• Ramp+	Ramp (positive)
• Ramp-	Inverted ramp This is very useful as a triggered envelope shape (it is not gated like an envelope).
• Env+	Envelope (positive)
• Env-	Inverted envelope – emulates the invert switch available on many synth envelopes

Additional Synth sources in Cypher

Cypher features some extra sources derived from its specialised oscillator functions.

Note: While the oscillators feature certain hard-wired audio-rate modulation functions, all of the TransMod sources derived from the oscillators are *quantized to control-rate*.

Oscillators as LFOs

The main purpose of these sources is to use an oscillator as an additional LFO, by engaging the **Low** button to set the osc to LFO mode (the LFO rate is set using the **Beat** control). The LFO shape is derived from the oscillator's waveform, which can be modulated using the **Wave** control.

• Osc1±	The frequency of Osc1 (shaped with Osc1 waveform)
• Osc2±	The frequency of Osc2 (shaped with Osc2 waveform)
• Osc3±	The frequency of Osc3 (shaped with Osc3 waveform)

Oscillator beating rates as LFOs

The following sources provide the beating rates of two oscillators against each other (effectively the difference in phase between them), applied to the shape of one of the oscs. Whether you are using the **Beat** control for constant beating rates, or simply using the **Scale** controls to detune normally with variable beating rates across the keyboard, osc beating rates are always calculated and provided as TransMod sources.

• Beat12±	Osc1 phase - Osc2 phase (shaped with Osc1 waveform)
• Beat21±	Osc2 phase - Osc1 phase (shaped with Osc2 waveform)
• Beat23±	Osc2 phase - Osc3 phase (shaped with Osc2 waveform)
• Beat32±	Osc3 phase - Osc2 phase (shaped with Osc3 waveform)
• Beat31±	Osc3 phase - Osc1 phase (shaped with Osc3 waveform)
• Beat13±	Osc1 phase - Osc3 phase (shaped with Osc1 waveform)

8: Fusor

8:1 What is Fusor?

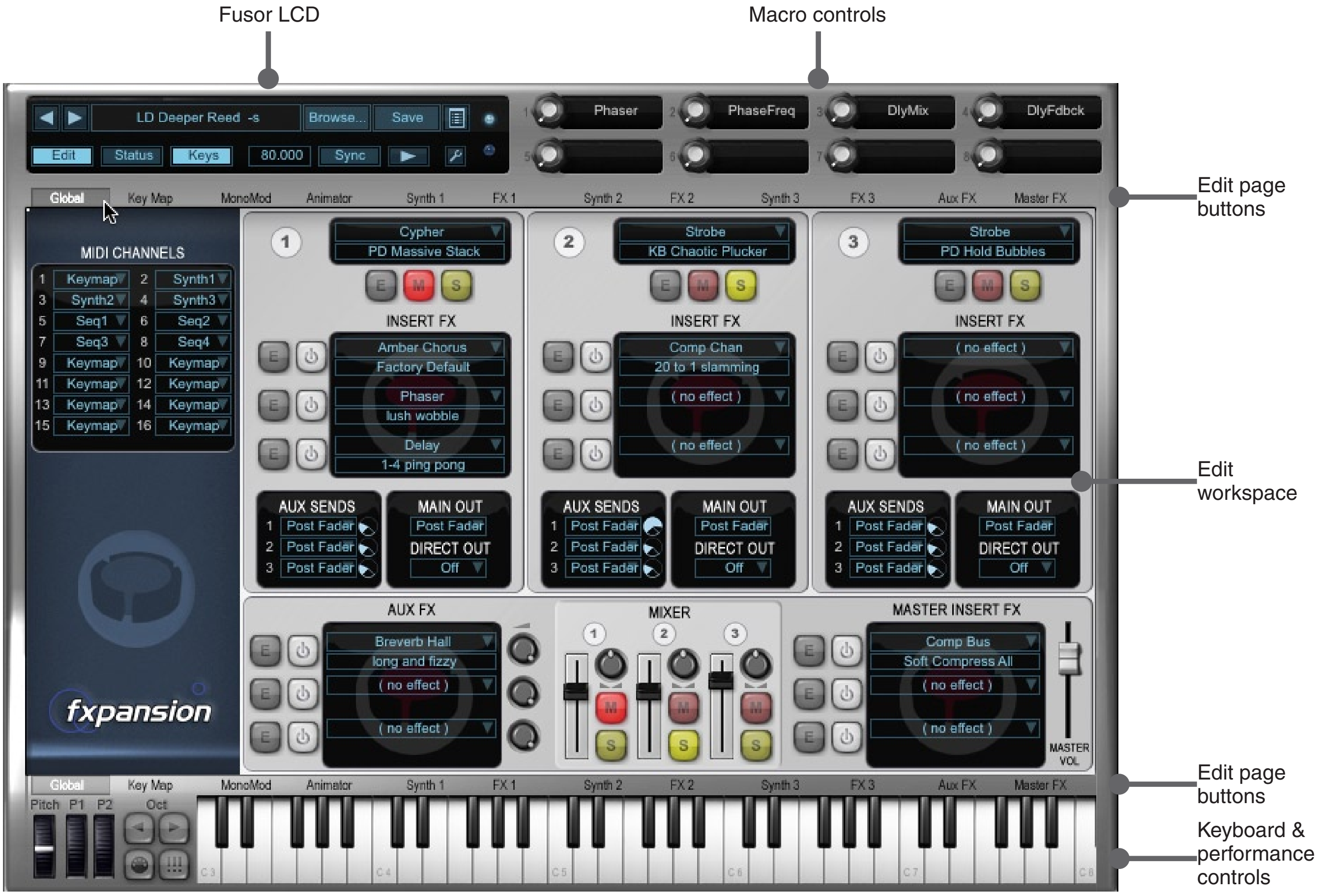
Fusor is a layering and performance environment for any combination of 3 DCAM synth instances. Although it provides a full effects suite, mixer and step-sequencer, it should be considered as an super-instrument or 'synth rig' rather than a complete host. It is intended for layered 'super-patches', not for full songs or as a conventional multi-timbral sound module.

Fusor hosts 3 synths, features 4 additional LFOs, 4 envelope followers, an effects suite and a complex four-engined step sequencer/arpeggiator called Animator. It is inspired by multi-section instruments like the Korg Trident and ARP Quadra – some of the most versatile instruments of their time. It is also influenced by the large, wraparound synth rigs employed by jazz fusion and progressive rock artists like George Duke, Joe Zawinul and Keith Emerson. These would typically comprise several analogue polys, monos and maybe even an analogue sequencer or a modular like an ARP 2600, alongside other keyboards and various FX units.

FuseMod

FuseMod is Fusor's modulation system, which is essential for using many Fusor devices. It operates in a very similar way to the synths' TransMod system, and effectively gives most parts of Fusor (including the hosted synths) a wide array of control signal inputs and outputs, allowing you to modulate one device with control signals from another. See section 8:8 for a full guide to the FuseMod system.

8:2 Interface Overview



Fusor LCD

The Fusor LCD contains the main preset picker (which loads and saves presets for the entire state of Fusor) and a number of additional controls. See section 1:3 for a guide to the standard preset picker controls (**Preset name**, **Prev/Next preset**, **Browse...**, **Tools**, **Preferences** and audio/MIDI indicators).

Edit	Toggles visibility of the Edit workspace and page buttons
Status	Toggles visibility of the Status display
Keys	Toggles visibility of the keyboard and performance controls
Tempo	Click and drag the tempo up/down to change it, or double-click it to type the required tempo manually. This control can only be adjusted when running the Fusor standalone application. When running Fusor as a plugin in a host, it is synchronized to the host tempo.
Sync	When this button is engaged, Fusor is synchronized to the host's tempo and transport when running as a plugin.
Start/Stop	Starts and stops Fusor's internal transport
Preferences	Displays Fusor's preferences panel. Note that Fusor's own preferences settings override those of the individual synths loaded into synth channels. The synths' own Preferences panels are disabled.

Macro controls

Fusor has 8 macro controls, which can be used to control any parameter that can be modulated within Fusor. They are designed to be used in a similar way to the performance controllers within the synths and Fusor (see below).

As such, they are intended to be mapped to a set of pots/faders that send out MIDI CCs (using MIDI Learn or the preferences panel, where they are assigned in the same way as performance controllers), but they can also be controlled via host automation. See section 6:8 for more details about host automation and mapping MIDI CCs using MIDI Learn.

Macro control MIDI assignments are stored in the Fusor preferences. Section 6:5 features more details about the preferences panel, including saving preferences settings for use in future sessions using the **Save as Defaults** button.

Like performance controllers, the destinations for macro controls are assigned via the FuseMod system. You can modulate most parameters within Fusor and its hosted synths. Like any DCAM modulation source, Macro control FuseMod sources can control multiple destinations.

Edit page buttons

The Edit page buttons provide access to each part of Fusor – click a button to display the associated page in the Edit workspace. The Global page also provides **Edit** shortcut buttons to the various pages.

Note that you can right-click (or CTRL-click) on any Synth page button to select the synth for the slot from a drop-down menu.

Edit workspace

This workspace contains all editing functions in Fusor. The Global page provides overall control and mixing functions, while the other pages allow you to adjust the various Fusor devices.

The Edit workspace can be hidden entirely by de-activating the **Edit** button in the Main LCD.

Keyboard and performance controls

Like each synth plugin, Fusor features a keyboard and set of performance controls – pitch bend, **P1** (**Perf 1**) and **P2** (**Perf 2**) controls – at the bottom of the interface. This area of the interface also contains the **MIDI Learn** and **Panic** buttons.

This keyboard is used for MIDI channel 1 in the Fusor MIDI channel router (see section 8:3). When synths are hosted in Fusor, their individual keyboards are not accessible.

The **Perf 1** and **Perf 2** controls function exactly as they do in the individual synths – see section 7:4 for more details on these controls, which are available as FuseMod sources. They are saved with Fusor's preferences, and override the synths' individual assignments when they are hosted within Fusor.

The keyboard and performance controls can be hidden from view by de-activating the **Keys** button in Fusor's main LCD.

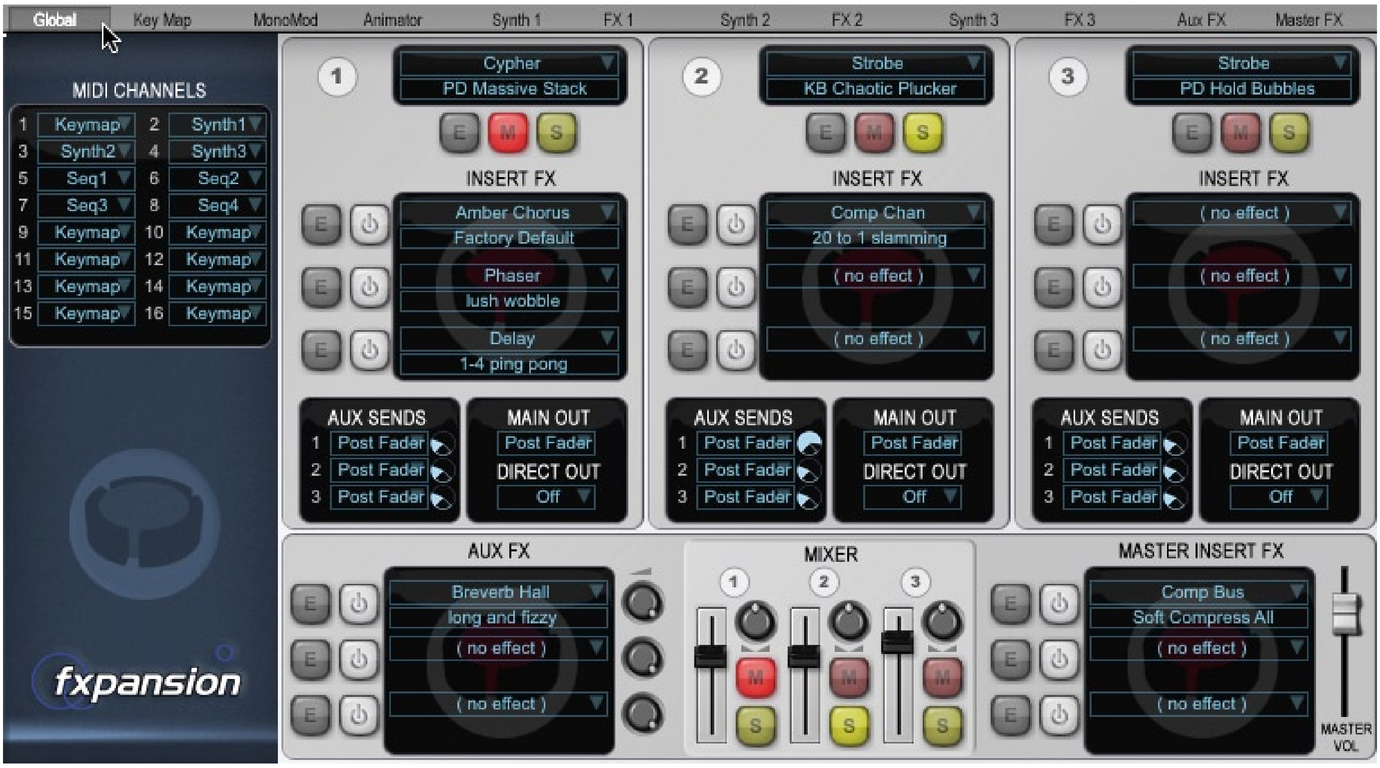
Status display

The Status display area provides access to some of the most important Fusor controls from any Fusor Edit page.

This display is disabled by default to save screen space. It can be shown by activating the **Status** button in the Fusor LCD. The following controls are available:

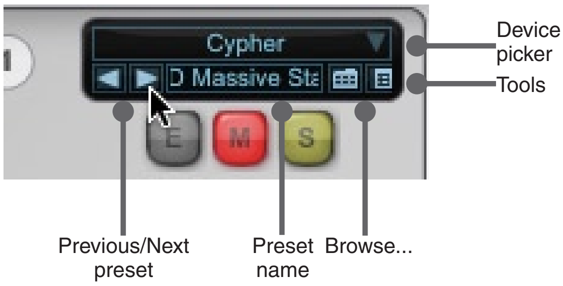
Synth channels	• Synth and synth preset picker • **Volume** and **Pan** • Aux **Send Level** controls and **Mute**/**Solo** buttons • Insert FX device and preset pickers
Aux FX	• Aux FX device and preset pickers
Master FX	• Master FX device and preset pickers

8:3 Global page



The Global page performs several functions:

- MIDI input channel routing to various parts of Fusor
- Mixing and routing of channels, aux sends and returns
- Device selectors, preset pickers and Edit shortcuts to device pages



Device and preset pickers

Device and preset pickers are available for all of the following:

- Synth and insert FX slots on each synth channel
- Aux FX slots
- Master insert FX slots

Note that each slot's preset picker appears after a device has been loaded into it.

The preset picker is used in the same way as the synth preset pickers that are described in section 1:3.

Use the **Previous/Next** buttons to cycle through the available presets, or click on the preset name to show a drop-down menu of presets sorted by style. The **Browse...** button launches the device's preset browser, while the **Tools** button displays the Tools menu.

Edit and Power buttons

Edit shortcut buttons are available on the interface to allow quick access to Edit pages for all elements that feature a device selector and preset picker.

These can be used as alternatives to the Edit page buttons.

All device slots also feature a **Power** button for the devices they contain.

MIDI channels

The MIDI channels router is used to distribute MIDI events to various parts of Fusor. You can use up to 16 MIDI input channels, each of which is represented by a drop-down menu. The following destinations are available for each MIDI channel:

Keymap	This can be used for layering and setting up keyboard split zones for playing multiple synths via a single MIDI channel
Synths 1, 2, 3	These are direct destinations to the 3 synth channels
Seq 1, 2, 3, 4	These destinations are for input to the 4 engines of the Animator step-sequencer/arpeggiator

The MIDI channels router is used only for note events – MIDI CC assignment is performed using MIDI Learn and the preferences panel.

Synth channels

Synth and insert FX controls

Each of the 3 synth channels feature the following:

- Synth device & preset pickers
- 3 insert FX (3 device selectors, preset pickers and **Power** buttons)
- **Edit** shortcut buttons for synth and 3 insert FX
- **Mute** and **Solo** buttons for the channel

Aux Sends

The 3 synth channels feature an aux **Send Level** control for each of the 3 aux FX slots. Sends are very useful for applying the same effect, such as a reverb, to more than one synth.

Aux 1-3 routing

Each aux send can be routed from several points in the signal path.

Main out routing

The Channel's Main output is fed into Fusor's Master stereo mix, where it is processed by the master insert FX before being fed to the master plugin output (output 1-2).

Direct out routing

Each synth channel features a Direct output from the Fusor plugin.

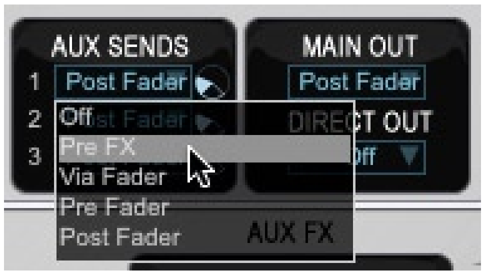
Note that the Fusor standalone application is restricted to the Master stereo output only – for multiple outputs, you must use Fusor as a plugin in a host.

Routing options

Each of the above can be routed from any of the following positions (note that the channel level faders are in the Mixer area):

OFF	No signal is sent to the destination.
Pre FX	The signal is sent to the destination before the channel's insert FX and level fader are applied.
Via Fader	As 'Pre FX', but the signal is affected by the channel's level fader.
Pre Fader	The signal is sent to the destination after the channel's insert FX are applied, but without being affected by the channel's level fader.
Post Fader	As 'Pre Fader', but the signal is affected by the channel's **Level** fader.

Aux Send Levels



Setting the Aux 1 send routing

Aux FX, Mixer & Master insert FX

Aux FX Return Levels

Aux FX & Master insert FX device and preset pickers

The aux FX and master insert FX slots all feature device and preset pickers, as well as **Edit** shortcut and **Power** buttons.

Aux FX return levels

Each of the 3 aux FX slots feature a **Return Level** control for setting the final output level from the aux FX.

Mixer

The Mixer area features controls for setting each synth channel's **Level** and **Pan**.

Mute and **Solo** buttons are to the right of each fader (they are also provided under the synth channels' device picker).

Master Vol (Master Volume)

The **Master Vol** fader controls the main Fusor output level. Direct output levels are unaffected.

8:4 Input device pages

Key Map page

In this example, the lower part of the keyboard is mapped to Synth 1, and the upper part is mapped to Synths 2 & 3.

Note the crossfaded area between the 2 keyboard zones. 2 key maps are used for Synths 2 and 3 to facilitate different velocity and **Transpose** settings.

The Key Map page allows you to create up to 4 synth or Animator keyboard zones, with optional keyboard split/break points and velocity adjustments. It is perfect for layering and arranging synths across the keyboard for performance uses or for huge, creative sounds.

Power

To enable a Key Mapper, engage its **Power** button.

Key Low

This is the lowest key in the key map's defined range. Anything below this key is not routed to the destination(s).

Break Low

Setting up break points allows velocity crossfades between two key maps directed at different synths. Incoming MIDI velocities are scaled between 0 at the **Low** key and maximum at the **Low Break** key.

Break High
Key High

These settings work identically to the above, but they define the upper limit of the key map's range, not the lower limit.

Transpose

Each key map features the ability to transpose incoming notes up or down by up to 36 semitones with this control.

Vel Scale

This control enables you to scale the velocity response of the key map.

Velocity Curve
Low & High velocity

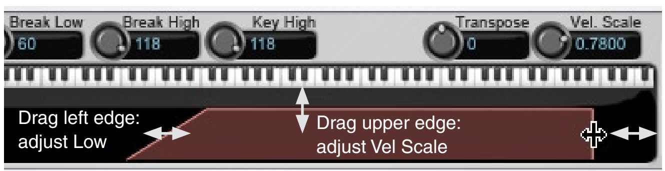
The **Velocity Curve** control adjusts the velocity response of the key map's range. Click/drag the curve to adjust its shape.

The controls on either side of the curve define the lowest and highest velocity.

Key map display

The display representing each key map can be directly edited by clicking and dragging the active keyboard range.

Click/drag the right edge of the range to adjust the **Key Low** and **Break Low** settings simultaneously, and click/drag the left edge to adjust the **Key High** and **Break High** settings. Click/drag the upper edge to adjust the **Vel Scale**.



Editing the key map display directly

Drag right edge: adjust High

Destinations

Each of the 4 key maps can be routed to *any number* of the following:

- Synth channels 1, 2 and 3
- Animator engines 1-4

To simply layer synths with a full keyboard range for each, you only need to use one of the 4 key maps with a full keyboard range defined (**Key Low** = 0, **Break Low** = 0, **Break High** = 127, **Key High** = 127). Then, enable all the required synths in the key map's **Destinations** section.

MonoMod page

This page contains Fusor's additional monophonic modulation sources:

- 4 LFOs
- 4 envelope followers with hard-wired audio sources

These modulators are useful for extra modulation sources for synths or FX devices within Fusor. The FuseMod system must be used on a destination device to modulate its parameters with MonoMod control signals.

The LFOs' outputs are available as the LFO1-4 and LFO1-4 [+] FuseMod sources.

The LFO's parameters can themselves be modulated – each LFO features 2 FuseMod slots.

See section 8:8 for more details on the FuseMod system.

LFO

The MonoMod LFO controls are almost identical to those of the standard DCAM LFO in Strobe, Amber and Cypher, but with the following differences:

- They are always limited to monophonic operation
- They cannot be looped
- They do not feature any gate source options – they are always gated by Song Phase. In other words, they are free-running LFOs which repeat identically each time you play through your host project.

Please see section 7:5 for details of the synths' LFOs.

Envelope followers

The 4 MonoMod envelope followers track audio from various parts of Fusor in order to generate envelope control signals that can be used in the FuseMod system.

As with LFOs, suitable FuseMod modulation must be used on the destination device to modulate its parameters with the envelope follower control signals.

The envelope follower outputs are available as the EnvFl1-4 FuseMod sources.

Attack, **Release** and **Gain** adjust the output envelope shape in relation to the input. These controls cannot be modulated.

Env Follower	Source
1	Synth 1 output
2	Synth 2 output
3	Synth 3 output
4	Master output

Audio inputs for MonoMod envelope followers. These inputs are hard-wired and cannot be adjusted.

Visualizer scope

The context-sensitive visualizer scope shows the output shape of the MonoMod device under the mouse.

It operates in a very similar way to the synths' scopes, which are described in section 6:6.

Note that the **Lock Scope** function is not possible.

Animator page

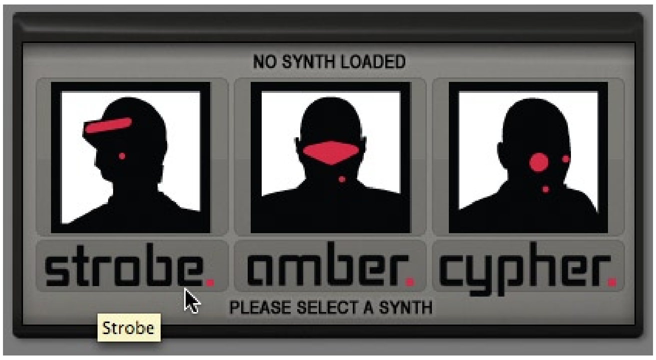
Animator is an advanced 4-engine step-sequencer with arpeggiator functions.

See section 8:9 for details of how to use Animator.

8:5 Synth channel pages

Each synth channel features 2 pages:

Synth	A single instance of Strobe, Cypher or Amber can be launched in each Synth page
FX	Each Synth channel can contain 3 insert FX



Synth selector on an empty synth channel

Synth1, Synth2 and Synth3 pages

When the channel is empty, the synth page displays the synth selector – click the desired synth to load it into the synth channel.

You can also load synths on the Global page, using the device selectors for the synth channels, or by right-clicking on a synth channel's Edit page button.

Routing MIDI events to a synth

To play a synth, you must make sure that events are routed to it from the Global page, the Key Map page or Animator.

When a synth is loaded into a channel, its on-screen keyboard and performance controls are discarded – to use Fusor's on-screen keyboard with a synth, MIDI channel 1 must be routed to the synth in the Global page, either directly or through a key map or Animator engine.

Note that the synth's own preferences panel is disabled when it is running in Fusor. This is because Fusor's performance controls and oversampling settings are used instead of those in the individual synths.

Modulation

Each synth retains its 8 TransMod slots within Fusor. In addition to the regular range of modulation sources normally available within the synth, FuseMod sources can also be used – from Animator engines, MonoMod devices and even other synths.

Meanwhile, certain modulation sources from the loaded synth are available to modulate other Fusor devices (these sources are denoted as 'Syn' sources within the FuseMod source menu). You can therefore modulate parameters of effects and other synths using envelopes, LFOs and other modulators from each synth.

See section 8:8 for full details of modulation within Fusor.

FX1, FX2 and FX3 pages

Each synth channel features 3 insert FX slots.

4 FuseMod slots are available for each FX slot, allowing any FuseMod sources to modulate FX parameters.

Hovering the mouse over any unused slot displays the FX selector – click on the desired FX device to load it into the slot. You can also use each slot's device picker.

Once loaded, the FX slot's preset picker becomes available.

See section 8:7 for an overview of the available FX devices.

8:6 Aux FX and Master FX pages

The aux FX and master FX pages are identical in appearance to the synth channel FX pages (see above).

Aux FX page

Fusor features 3 aux FX slots, to which audio can be sent from any of the 3 synth channels.

Sends to the aux FX are configured either from the Global page or in the Status display area, while their return levels are controlled on the Global page.

Master FX page

Fusor's master output features 3 insert FX slots. These operate on the Main output busonly, which comprises audio sent to the Main destination from synth channels (make sure their Main output routing is not set to 'OFF') and from the aux returns.

The master insert FX do not affect the synth channels' individual Direct outputs.

8:7 Fusor FX suite

Overview

FX can be used in the following ways in Fusor:

- 3 insert FX in each of the 3 synth channels
- 3 aux FX with sends from each synth channel
- 3 insert FX on the master output

Wet Mix control

With the exception of the Gain and EQ, all effect devices feature a **WetMix** control that allows you to blend between dry and processed signals, for quick parallel processing.

Effect types

Gain

This is a very simple gain and pan device, but its parameters can be modulated via the FuseMod system for enveloping, tremelo and auto-panning. The **Amp** control provides exponential gain from -inf. to +36dB, while the **Pan** control sets the position in the stereo field. **AmpLin** provides a linear amplitude function, which is useful if you want to modulate the amount of gain with an exponential source.

Gate (Noise Gate)

The Gate attenuates the signal until its amplitude reaches the **Threshold** level. The **Attack** and **Release** controls set the speed at which the gate opens and closes after the signal goes above or below the threshold.

The **HPFreq** and **LPFreq** controls allow you to high-pass and low-pass filter the signal used for the amplitude-detection circuit, while the **Preview** control allows you to listen to this signal.

Increasing the **Hysteresis** control smooths the gate's response, at the expense of sensitivity to small changes around the threshold level.

EQ

The 4-band EQ is great for refining the final output of a synth, especially when layering synths together.

Each Band is selected using buttons **1-4**. All 4 parameters in each band can be modulated via FuseMod. The EQ also features the Rel/Abs/Dif/Rnd edit modes of the X4 quad effects (see below for details).

Bands 1 and 4 feature a control for switching between shelf and bell modes (**LF-Bell** and **HF-Bell**), which is replaced by a **Q** control for the parametric bands 2 (**LMF-Q** - low mid) and 3 (**HMF-Q** - high mid). All bands feature power (**In**), frequency (**Freq**) and **Gain** controls.

EQ-Filter

The EQ-Filter provides simple 1-pole high-pass and low-pass filters. Each filter features power (**LPin** and **HPin**) and frequency (**LPFreq** and **HPFreq**) controls.

Comp Chan

This is a DCAM circuit-modelled channel compressor, based on a classic 'limiting amplifier' design. Increase the **Input** control to make the sound more compressed, and adjust the **Output** level as required. Use the **Ratio**, **Attack** and **Release** controls to affect the compression characteristics.

Comp Bus

The Comp Bus is DCAM circuit model of a classic British console bus compressor design, and features **Threshold**, **Ratio**, **Makeup**, **Attack** and **Release** controls.

The **Sidechain** control allows you to highpass-filter the signal used for the amplitude-detection circuit, while the **Analogue Limit** control applies non-linearities to the detection circuit, resulting in a compression characteristic that is more transparent on attacks.

Delay

This is a classic stereo delay line. As well as **Time** and **Feedback** controls, it features a **Swing** control for achieving a variety of delay feels, and adjustable **LowCut** and **HighCut** filters in the feedback chain.

The **Sync** control switches between time-based (Sec) and tempo-synced (BPM) delay times. The **SumInput** control sums the left and right inputs to a single mono signal and feeds one delay line instead of two.

Phaser
Flanger
Chorus

These are classic time and pitch modulation effects to add richness, movement and psychedelic character to the signal.

RingMod

This effect is used for radical timbral shifts and experimental effects. It multiplies the input signal with its internal oscillator, the shape and pitch of which are set with the **Mode** and **Pitch** controls. The **Drive** control sets the amount of distortion on the input signal.

TinCanVerb

This effect is a recreation of a low-end room reverb unit, perfect for emulating 'cheap and nasty' onboard synth FX. Use the included Overloud Breverb devices for high-quality reverb FX.

As well as the ubiquitous **Size**, **Decay Time** and **Damp** parameters, TinCanVerb features **Pinch** and **Squeeze** controls for manipulating the room shape, while the **Freeze** control loops the current reverb buffer indefinitely until it is turned down again – useful for dubby special effects.

Filter Mod

This is a DCAM circuit-modelled multimode resonant filter with modulation and drive. It is based on a simplified version of Strobe's filter. As well as the **Pitch** (cutoff frequency) and **Res** (resonance) controls, the filter features a **Mode** control to switch between low-pass, band-pass, high-pass, peak and notch modes. There is even an audio-rate **FMDepth** control which sets the amount of cutoff modulation modulated by the input signal's waveform. The Drive function features controls for input (**Drive**) and output (**OutDrive**) drive stages.

The effect contains a built-in envelope follower for modulating the cutoff frequency with the amplitude of the input signal. It features controls for **Attack**, **Release** and **EnvDepth**.

FreqShift (Frequency Shifter)

This is a frequency shifter effect which changes pitch without preserving the harmonic information, resulting in very alien and abstract timbres.

The **Pitch** control sets the amount of frequency shifting above or below the original signal, while the **Amp** control adjusts the level of the output signal.

BitCrusher

This effect emulates the digital distortion that occurs when lowering the bit-depth or sample-rate of an audio signal. Adjust the bit-depth with **Bits** control and the sample-rate with the **Freq** control.

The **Dirty** control sets the amount of the distorted signal that is heard at the output, while the **Clean** control sets the amount of the signal that was filtered out before the bitcrushing stage by the high-pass and low-pass filters. The **Tone** control is a simple 1-pole low-pass filter to remove unwanted high-frequencies from the Dirty signal.

Drive

Drive is a versatile, DCAM-modelled overdrive/distortion effect. The **HPFreq** and **LPFreq** controls allow you to filter the signal before the distortion stage, while the **Mode** control switches between 4 different distortion models – 'Diode', 'OTA', 'OpAmp' and 'HalfRect'.

Like the BitCrusher effect, it features pre-distortion high-pass and low-pass filters before the distortion stage, as well as similar **Dirty**, **Clean** and **Tone** controls.

Freezer

This is a granular buffer-looping/freezing effect that is useful for glitchy and stuttery sounds.

Once the **Gate** control is turned to the 'On' position, the Freezer effect starts recording audio from the input into a buffer, whose length is dictated by the **LoopLength** control (1-16 beats). The loop buffer is filled until the end of the LoopLength, after which subsequent incoming audio is ignored, until the buffer is re-gated (Gate control set to 'Off' and 'On' again).

The audio in the loop buffer is divided up into slices, the size of which is dictated by the **GrainSize** parameter. This can be set in seconds (to a maximum of 2 seconds) or BPM units, depending on the setting of the **Sync** parameter.

Once the buffer has been filled, the Freezer loops the first grain – it 'freezes' the grain – while the **Speed**, **JumpRand** and **JumpManual** parameters are set to 0.

Increasing the Speed control plays through the grains sequentially starting with a grain between 0 and 16, set using the JumpManual parameter, as long as JumpRand = 0. A setting of 100% is normal speed.

Increasing the JumpRand parameter results in the Freezer jumping to random grains instead of playing through them sequentially.

The **Scratch** parameter scales the pitch of the loop just like a record on a turntable, so you can play the loop forwards and backwards and everywhere in between.

Breverb Hall
Breverb Room
Breverb Plate
Breverb Inverse

The 4 Breverb algorithms, licensed from Overloud Technologies, provide high-end digital reverberation effects.

Each algorithm is provided as an individual device, each containing specialised controls. See Appendix 4 for a detailed description of the parameters.

Amber Chorus
Amber Formants

These effects feature Amber's circuit-modelled chorus and formant filter stages. The controls are identical to those in Amber – see sections 5:4 and 5:5 for details.

X4 Quad FX

These FX devices provide 4 bands of the same effect running in parallel. The buttons numbered 1-4 switch between each of these 4 bands for editing.

Four additional editing modes are also available for adjusting all 4 bands together. The controls in these modes cannot be modulated (apart from Wet Mix which is a global control for all bands) – they are only for editing all 4 bands' initial values together. The controls in each of the 4 bands have a suffix of 1-4, depending on the band.

Abs	Sets an absolute value across all bands
Rel	Adjusts parameters across all bands relative to their original positions where possible within their range
Dif	Adjusts the overall difference of parameters' values between the 4 bands
Rnd	Randomizes parameters across all bands

Common parameters

As well as the **WetMix** control, which operates on all bands simultaneously, each band in all the Quad FX features the following controls:

Pwr	Turns the band on or off – in the 'Off' position, audio passes through the band unaffected.
Pan	Sets the position of the band in the stereo field.
Amp	Sets the final level at the effect output.

X4-Filter

Each band features 2 filters, selectable from the following combinations using the **Mode** control:

L2L2	2x 2-pole low-pass filters
P2L2	2-pole peak and 2-pole low-pass filters
N2L2	2-pole notch and 2-pole low-pass filters
L2H2	2-pole low-pass and 2-pole high-pass filters
H2H2	2x 2-pole high-pass filters
H2P2	2-pole high-pass and 2-pole peak filters
H2N2	2-pole high-pass and 2-pole notch filters
N2N2	2x 2-pole notch filters
P2P2	2x 2-pole peak filters

Each band features an optional pre-delay which is turned on/off using the **PrePwr** control and set using the **Pre** control (whether this parameter is set in seconds or tempo-based values depends on the state of the **Sync** control).

The **Pitch** control sets the filter cutoff of both filters together, while **Res** sets the resonance of both. The **Sep** control adjusts the separation between the 2 filters.

There is a drive stage in each band, which is adjusted using the **Drive** control.

X4-Drive

Each band can be set to one of 4 **Modes** – 'Diode', 'OTA', 'OpAmp' and 'HalfRect' – with each offering a different distortion flavour. There is a band-pass filter before the drive stage, which can be adjusted using the **Pitch** and **Width** controls. The **Drive** control sets the amount of distortion, while the **Lp** control adjusts a 1-pole low-pass filter after the drive stage.

X4-Comb

The **Pitch** and **Feed** controls set the frequency and feedback amount of the comb filter.

The **Mode** control sets the comb polarity to positive or negative. Each band features an optional pre-delay which is turned on/off using the **PrePwr** control and set using the **Pre** parameter (whether this parameter is set in seconds or tempo-based values depends on the state of the **Sync** control).

The **Hp** and **Lp** controls adjust 1-pole high-pass and low-pass filters in the comb signal path.

X4-Delay

The **Time** and **Feed** controls adjust the delay time and feedback respectively. The Time control is set in seconds or tempo-based values depending on the state of the **Sync** control.

Each delay band also features a pre-delay, which is adjusted using the **TimePre** control. Whether this is set in seconds or tempo-based values depends on the state of the **SyncPre** control.

The **Hp** and **Lp** controls adjust 1-pole low-pass and high-pass filters within the feedback path.

8:8 Modulation in Fusor

FuseMod

Fusor's modulation system, FuseMod, is very similar in concept to the TransMod system in Strobe, Cypher and Amber. It allows most parts of Fusor to modulate each other with control-rate signals, providing many possibilities that can only be found in semi-modular synths.

For example, synths can modulate each other's parameters, or those of FX, while being modulated by the macro controls, the Animator step-sequencer/arpeggiator and the extra MonoMod modulators (LFOs and envelope followers). The possibilities with FuseMod are limited only by your imagination!

You can think about it as a system that exposes 'CV inputs and outputs' for most parameters.

It is important to note that all of the FuseMod modulation capabilities are for control signals only – LFOs, envelopes, ramps, sequencer modulation and so on – and *not* for audio signals.

Make sure to carefully read chapter 7, which describes the TransMod system. Most of the same functionality exists in the FuseMod system.

Devices that can be modulated

The following Fusor devices feature FuseMod modulation slots:

Device	FuseMod slots on each
LFOs (4)	2
Animator Graph modulation (Advanced engine type)	4
Animator External Clock Source	1
Synth channel insert FX (3 FX on each of 3 channels)	4
Aux FX (3)	4
Main FX (3)	4

Using the FuseMod system

The FuseMod system is functionally very similar to the TransMod system found in the individual DCAM synths (see chapter 7 for a full guide to the TransMod system).

FuseMod slots work identically to TransMod slots. Therefore, FuseMod slots, sources and scalers are selected in the same way, with the same highlighting and mouseover indicators. This functionality is described in sections 7:2 and 7:3.

FuseMod modulation sources

The following modulation sources are available in all FuseMod modulation slots.

• Perf1 [+] • Perf2 [+]	Performance controller MIDI CC assignments, like macro controls, are stored in the Fusor preferences and are designed to be accessible across all Fusor presets which feature these modulation sources. These are uni-polar sources.
• Macro 1-8 [+]	The 8 macro controls are conceptually similar to the performance controllers described above – they are designed to allow quick access to important parameters in Fusor presets. These controls are always visible at the top-right of the Fusor interface, regardless of which page is currently in view. They are uni-polar sources.
• Step1-4	The modulation value output from each of Animator's 4 engines, bi-polar
• Step1-4 [+]	The modulation value output from each of Animator's 4 engines, uni-polar
• LFO1-4	The output of the 4 MonoMod LFOs, bi-polar
• LFO1-4 [+]	The output of the 4 MonoMod LFOs, uni-polar
• EnvFl1-4 [+]	The output of the 4 MonoMod Envelope followers, uni-polar
• Syn	A selection of control-rate modulation sources from each synth These sources can be uni-polar or bi-polar. See below for details.
• ModWheel [+] • Breath [+] • Expr [+] • MonoBend • MonoPress [+]	These are monophonic keyboard input sources which function in the same way as the identically named TransMod sources described in section 7:8. They are all uni-polar except MonoBend (pitch-bend wheel), which is bi-polar.
• KeyRand	Provides a new random number between -1 and 1 on each key-on, bi-polar

All of these modulation sources operate monophonically since the sources and destination devices throughout Fusor are monophonic in nature.

However, it is possible to use modulation between synths polyphonically using the synths' own TransMod slots (see below).

'Syn' FuseMod sources

When a synth is loaded into one of Fusor's 3 channels, 8 TransMod sources from the synth become available as 'Syn' FuseMod sources for modulating Animator, MonoMod LFOs and FX parameters anywhere in Fusor (monophonically). The following sources are available from each synth:

Strobe	Cypher	Amber
Pitch±	Pitch±	Pitch±
OnVel+	OnVel+	OnVel+
Ramp+	LFO1±	Ramp+
LFO±	LFO2±	LFO±
Drift±	Noise1±	Rand1±
LFO+	Env1+	ModEnv+
ModEnv+	Env2+	SynEnv+
AmpEnv+	AmpEnv+	EnsEnv+

See section 7:8 for details of these sources.

Using FuseMod modulation within synths' TransMod slots

FuseMod sources can be found in the Macro, Macro S+H and Synth *n* sub-menus of TransMod slot sources.

As well as FuseMod sources (such as those from Animator and MonoMod devices), the entire TransMod menus of other loaded synths are available. These sources can act polyphonically if each synths' voice/unison settings allow it.

This function allows you to use an instance of Cypher as an extension to Strobe, for example – they can share each other's envelopes, LFOs and so on, and of course you can combine their audio outputs in interesting ways using Fusor's mixing and effects capabilities. It's like having 3 analogue synths with far more CV i/o than costs would usually allow, alongside a versatile effects and step-sequencing rig with its own array of CV i/o.

Macro source sub-menu

The following sources are available in this sub-menu, in monophonic form:

- Macro controls
- Animator engine outputs
- MonoMod device outputs

Macro sources S+H sub-menu

This sub-menu has the same sources as those in the Macro sub-menu, except that the sources are 'key-on sample-and-hold' polyphonic versions. Each time you play a new note, the FuseMod source values are sampled and held. This held value is used for the played note's modulation.

A good use of these sources is to assign a MonoMod LFO or envelope follower to a synth's pan position, so that each note is placed at a different position in the stereo image.

Synth sources sub-menus

When a synth is loaded into a synth channel in Fusor, its entire TransMod menu is accessible by other loaded synths, in the Synth *n* sub-menu, where n is the synth channel number.

Any monophonic or polyphonic TransMod source can be used. As long as the current synth has suitable voice, unison and note priority settings, polyphonic TransMod signals from a source synth will act polyphonically on the destination synth.

See section 7:8 for a summary of synth TransMod sources.

This example shows ModEnv2 from Cypher (synth channel 2) being selected as the source for a TransMod slot in Strobe (synth channel 1).

Note the Macro and Macro S+H sub-menus also within the TransMod source menu.

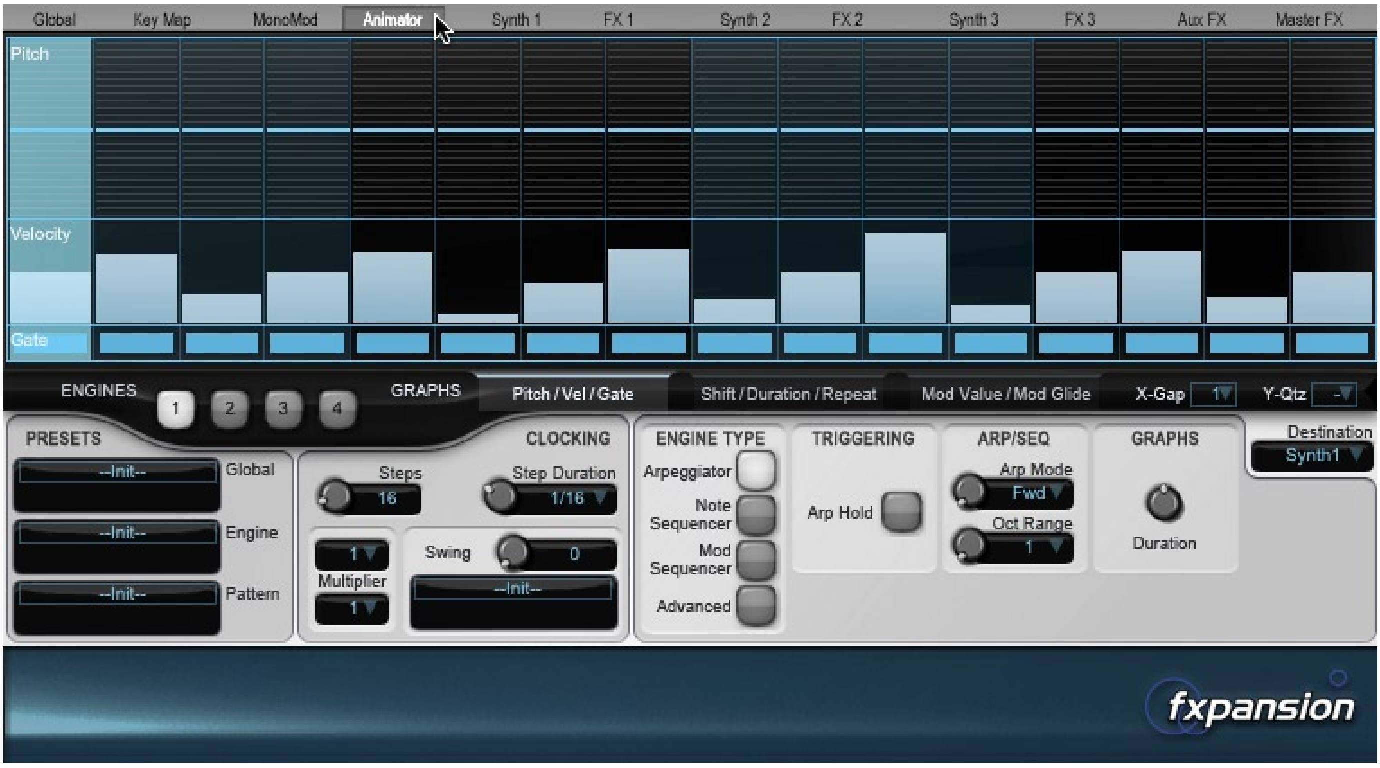
8:9 Animator: global controls

Animator is Fusor's built-in step-sequencer and arpeggiator. Many classic analogue synths feature a performance sequencer or arpeggiator, while many synthesists employ dedicated analogue step-sequencers to trigger a variety of synths and modulars through CV connections.

Step-sequencers and arpeggiators remain extremely popular to this day. Many would suggest that they have been crucial to the existence and development of many modern forms of electronic and dance music. Their imprint can be heard in the early experiments of pioneers like Raymond Scott and the futuristic space-jazz of Herbie Hancock's 'Sextant' album, and more recently in 303-inspired acid and all flavours of modern techno, house and related genres.

Architecture overview

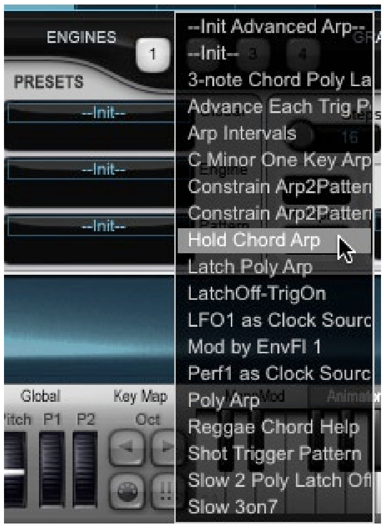
Animator has 4 engines, each essentially being an independent step-sequencer/arpeggiator in its own right. Each engine features patterns which contain step events arranged on a number of Graph lanes.



Preset pickers

Preset pickers are provided for the following areas of Animator:

Global	The entire state of Animator (all 4 engines)
Engine	The current engine
Pattern	The current pattern
Swing	The current pattern's swing template Note that this picker does not feature a Save As... function or a preset browser.



Using the engine preset picker

The preset pickers feature the usual preset picker controls. Use the **Prev**/**Next** buttons, preset browser (click the **Browse...** button) or preset menu (click on the preset name) to load presets for each area of Animator.

Save presets with the **Save** button, or use the **Tools** menu to Cut, Copy and Paste presets (especially useful for engines and patterns).

Engine selector

Animator's 4 engines can be selected using the engine selector buttons.

Engine type

Each of the engines can be set to any one of four engine types.

The controls available in the engine are dependent on the current engine type setting. Note that some controls are common between several engine types, and their *settings persist when switching between engine types*.

1. Arpeggiator (section 8:10)

This engine type allows easy use of Animator as an arpeggiator. Pattern Graph values are applied to the arpeggiator pattern, allowing much more interesting sequences than are possible with conventional arpeggiators.

2. Note Sequencer (section 8:11)

The Note Sequencer engine type is used to generate note events. It operates in a similar way to classic 'groovebox' step-sequencers such as the TB-303.

3. Modulation Sequencer (section 8:12)

This engine type is used for parameter modulation sequencing. Note input is fed through to the destination synth, while the engine must be routed to parameter modulation via the destination synth's TransMod slots, or to any other modulateable parameter within Fusor.

4. Advanced (section 8:13)

The Advanced engine type is a complex combination of the other types, with many extra functions and possibilities.

There are 12 pattern memories and the base values of Graphs can be modulated via the FuseMod system.

Clocking

This area of Animator controls each pattern's length and timing settings. Each pattern in each of the 4 engines can be clocked independently with different settings. This is great for polyrhythmic music and avoiding overly repetitive sequencing. Try combining modulation sequences and note sequences of different lengths and/or speeds for interesting polyrhythmic effects.

Steps

This control sets the length of the pattern in steps, from 1 to 128.

The default setting is 16 steps.

Step Duration

This control adjusts the length of each step, in musical note length values.

The default setting is 1/16th notes.

Multiplier

The 2 drop-down menus for this function allow you to multiply the clock rate by a fraction or by a multiple.

A setting of 1/2 means that the clock rate is halved (the effective step duration is doubled), while a setting of 2/1 results in doubling the clock rate (halving the effective step duration).

The default setting is 1/1, meaning that the effective step duration is unmodified.

Swing

Each pattern can have a different amount of swing applied (for syncopation effects), using any available swing template.

Use the swing preset picker to choose a swing template, and adjust the **Swing** control to specify the amount of swing applied.

Important note about clocking and triggering

For most triggering modes, Fusor's clock *must* be running for Animator to function.

Therefore, the Fusor transport should be started when using these modes.

The 'Advance', 'NoteScale', 'NoteWrap' and 'Ext. Mod' modes (available when using the Advanced engine type) do not require that the clock be started, although it must be running for full functionality. See section 8:13 for more details.

Driving devices with Animator

When using an engine for sequenced notes or arpeggiator patterns, you can route its output to any single synth channel, or to the Key Map page if you want to use it for more than 1 synth at a time.

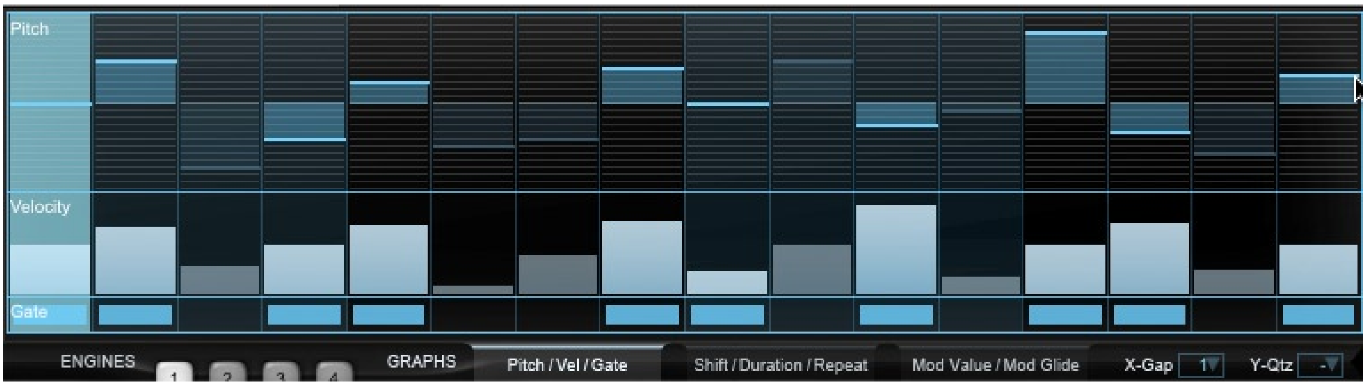
If you are using Animator as a modulation sequencer (to sequence parameter values), you must route this modulation using the FuseMod system, via the Step1-4 sources.

Animator Graph editor

The main 'LCD' display in Animator is the Graph editor, which allows you to draw pattern events to use with Animator's various modes. There are 3 pages of Graph lanes in the Animator LCD:

- **Pitch / Velocity / Gate**
- **Shift / Duration / Repeat**
- **Modulation / Mod. Glide**

Graph lanes are always scaled to the current pattern length (set with the **Steps** parameter).



Using the Graph editor

- Click in a step to set its value.
- Click and drag left/right in order to 'paint' values across multiple steps.
- Hold down ALT and click and drag up/down after a painting across multiple steps in order to ramp the value up or down.
- Hold down CTRL when drawing or painting events in order to revert to the steps' default values.
- Hold down SHIFT to edit multiple values at once: click and drag across multiple steps with SHIFT held down, then drag the mouse up/down.

X-Gap

This function allows you to skip steps (leaving gaps) when painting multiple events by clicking and dragging left/right. It is especially useful for restricting edits to on- or off-beats.

At its default setting of 1, no steps are skipped. However, setting it to 2, 3 or 4 results in only affecting every 2nd, 3rd or 4th step from that which was originally clicked.

Y-Qtz

The Y-Quantize function restricts the resolution of values, in the Modulation Graph *only*, to 1, 2, 6 or 12 levels.

Note that the number represents the resolution allowed both above and below the center zero-line. Therefore, a setting of 6 gives you 13 values: six above the centre line, six below, and the centre line itself. A setting of 1 gives you 3 values: 1, 0 and -1.

Graph lanes

Pitch (Arp, Note Seq and Advanced modes)

This Graph represents pitch offsets from the input.

When using the Note Sequencer engine type, the central value defaults to middle C (C5). If a MIDI note is routed to it, the input note becomes the central value. This mode also features a **Pitch** control to offset the Graph by positive or negative amounts.

In Arpeggiator mode, Pitch Graph values offset the pitch of each arpeggiator step. This is especially useful for shifting the octave of individual steps in more advanced ways than the regular **Oct. Range** and **Oct. Mode** functions allow.

Velocity (Arp, Note Seq and Advanced modes)

The Velocity Graph sets the absolute velocity of each step when using Note Sequencer/Arpeggiator/Advanced engine types.

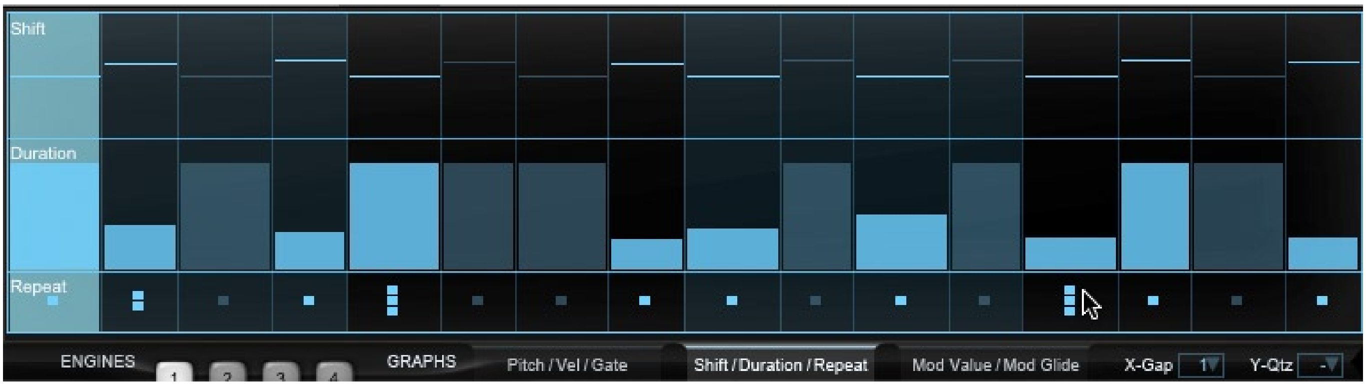
If you are using the Advanced engine type, the **Use Vel** control allows you to offset these values with the velocity of input events for each step of the arpeggiator pattern (see section 8:13).

The Note Sequencer engine type features a **Velocity** control which offsets the Graph by a positive or negative amount.

Gate (All modes)

This Graph switches each step on or off.

When using the Mod Sequencer engine type, if a step is off, the Mod. Value Graph remains at the value of the previous step.



Shift (All modes)

Shift values represent offsets from grid divisions – you can use them to make arpeggios and sequences feel less robotic and mechanical. Positive values shift the start of the step later in time, and negative values shift it earlier.

Using positive Shift values on offbeat steps achieves similar results to classic swing/shuffle functions. This technique can be used as an alternative to the engine-wide **Swing** function.

Duration (Arp, Note Seq and Advanced modes)

The Duration Graph sets the length of notes within each step – it is expressed as a % of the full step duration.

If the Duration is set at 99% for a step, the note duration is the entire length of the step. Setting it higher results in it being 'Tied' – if the next step is the same pitch, their durations are combined. If the next step is a different pitch, the notes overlap, something which is very useful for fingered glides in conjunction with a synth's **Pitch** glide with **Legato** enabled.

The Arpeggiator and Note Sequencer engine types feature a **Duration** control, which offsets the Graph by a positive or negative amount.

Repeat (Arp, Note Seq and Advanced modes)

This Graph allows you to repeat a note within a step up to 4 times, which can create interesting rhythmic and glitch effects.

Mod. Value (All modes)

The Mod. Value (modulation value) Graph sets parameter modulation values for each step.

Unlike note sequencer and arpeggiator events, Mod. Value events are not directly routed to any parameters. You must use the Engine's output as a source in a FuseMod slot or a TransMod slot within a loaded synth, and define modulation depths to parameters.

They can operate alongside note sequencer or arpeggiator events.

The Mod. Value Graph has an additional purpose when using the 'Sequenced' setting for the **Arp. Mode** or **Octave Mode** controls. See the descriptions of these functions in sections 8:10 and 8:11 for more details.

Mod. Glide (All modes)

This Graph adjusts the glide time from the previous step's Mod. Value Graph events.

Note that this Graph only affects the modulation Graph output into the FuseMod system. If you want to create conventional pitch glide effects with note sequencer or arpeggiator events, you must use suitable Duration Graph values and set the destination synth's internal **Pitch** glide control accordingly.

Advanced engine type Graph modulation

When using the Advanced engine type, each Graph features a control to offset its 'base value'. These controls can be modulated via the FuseMod system. See section 8:13 for more details.

The other engine types contain non-modulateable controls for offsetting certain Graph base values.

8:10 Animator: Arpeggiator engine type

The Arpeggiator engine type features minimal controls for simple operation. In addition to the controls listed below, the Graphs can be used to inject more variety than is possible using the standard arpeggiators built into Strobe and Cypher. For example, you can use the Gate Graph to turn steps on and off, or use the Pitch Graph to set specific steps to different octaves or entirely different notes.

All Graphs can be used simultaneously – the Modulation Graph can be used concurrently to modulate a synth's parameters via a TransMod slot.

Destination

Events from the engine can be routed to any of the 3 synth channels, or into the Key Map page if you wish to use the engine with more than one synth.

Arp/Seq section

Arp. Mode

This control sets the way in which the arpeggiator plays through the note priority queue.

By default, the note queue is dictated by the order in which the notes were played. However, this can be changed using the **Priority** control in the Advanced engine type. You can switch to Advanced, make the setting and switch back to the Arpeggiator engine type without losing your settings.

For an 'up-the-scale' arpeggio, Priority needs to be set to 'Lowest' so that the lowest note is always the first in the arpeggio.

Here is a summary of the arpeggiator behaviour for each possible **Arp. Mode** setting:

Fwd	Plays sequentially forward through the note priority queue.
Rev	Plays backwards through the note priority queue.
FwdRev	Plays forward through to the end of the note priority queue and then backwards to the start.
RevFwd	Plays backwards through the note priority queue from the end and then forwards from the start.
Random	Plays random notes from the note queue.
Sequenced	Plays notes from the queue based on the value of the Mod. Value Graph. Low values for the Graph select notes from the front of the priority queue, while high Graph values select notes from the back of the priority queue. This function allows you to play more complex arpeggiator patterns than those possible using other modes.

Oct. Range

This control sets the number of octaves over which the arpeggiator sequence is played. At a setting of 1, this control has no effect.

Graphs section

Duration

This control offsets Duration Graph values by positive or negative amounts, making the effective length of notes longer or shorter.

Triggering section

Arp Hold

Activating this button while notes are held down results in the current arpeggio continuing indefinitely, without needing to keep the keys held down on the keyboard. Playing further notes adds them to the arpeggio.

Deactivate the **Arp Hold** button to release the held notes, or click the **Panic** button.

Using the Advanced engine type as an arpeggiator

The Advanced engine type allows more varied and complex arpeggiator behaviour.

- You can set the **Trigger Mode** to 'Trig' in order to reset to the start of the pattern on new arpeggios, instead of the pattern synchronizing to the transport (as the default setting is 'Sync')
- Set the **Latch / Hold** parameter to 'Off' in order to use the Advanced engine type as a conventional arpeggiator. The 'Latch' and 'AutoHold' settings represent two 'held arpeggiator' behaviours.
- **Oct. Mode** allows you to refine the way in which the arpeggio moves between octaves, if the **Oct. Range** setting is higher than 1.
- **Priority** allows you to specify how MIDI input notes are ordered in the note priority queue.

There are a number of additional settings that allow a variety of advanced arpeggiator behaviours. See section 8:13 for details.

Switching between engine types

It is possible to switch to the Advanced engine type to adjust certain settings, such as the **Trigger Mode**, and then switch back to the Arpeggiator engine type, without losing any settings. You can even switch to new patterns in this way.

8:11 Animator: Note Sequencer engine type

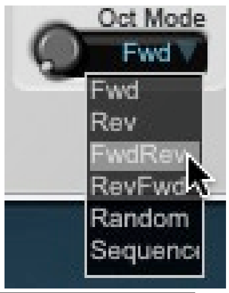
The Note Sequencer engine type offers a 'classic' analogue-style pitch sequencer. By default, it is used as a simple TB303-style start/stop sequencer using the Play transport in the Fusor LCD.

You can switch to the Advanced engine type in order to change this behaviour by adjusting the Trigger Mode control, and then switch back to the Note Sequencer engine type. 'Sync' is the default Trigger Mode – you can change this to keyboard-gated/-triggered behaviour with settings such as 'Gate' or 'Trig'.

Destination

Events from the engine can be routed to any of the 3 synth channels, or into the Key Map page if you wish to use the engine with more than one synth.

Arp/Seq section



Oct Range

If this control is set to 2 or higher, the pitch of Note Sequencer events is shifted to 1 or more higher octaves every 2 steps.

Oct. Mode

This control dictates the sequence of shifts within the octave queue when using the **Oct. Range** control. The octave queue is always ordered from *lowest to highest*.

If Oct. Range is set to 1, the **Oct. Mode** control has no effect.

Fwd	Plays sequentially forward through the octave queue.
Rev	Plays backwards through the octave queue.
FwdRev	Plays forward through to the end of the octave queue and then backwards to the start.
RevFwd	Plays backwards through the octave queue from the end and then forwards to the start.
Random	Uses random octaves from the octave queue.
Sequenced	Uses octaves from the queue based on the value of the Mod. Value Graph. Low values for the Graph select low octaves from the front of the priority queue, while high Graph values select high octaves.

Graphs section

Pitch, Velocity & Duration

These controls offset the Pitch, Velocity and Duration Graphs by positive or negative amounts.

8:12 Animator: Modulation Sequencer engine type

The Modulation Sequencer engine type allows you to send sequenced parameter modulation values (set via the Mod. Value and Mod. Glide Graphs) to a synth while also routing notes through to it.

For Mod. Value events to have any effect, the relevant engine *must* be used as the source in a TransMod or FuseMod slot with suitable depths specified on 1 or more destination parameters. The Step1-4 FuseMod sources correspond to the modulation output from each Animator engine.

The Pitch, Velocity, Duration and Repeat Graphs do not have any effect when using the Modulation Sequencer engine type.

Destination

While the Mod. Value events do not have any effect on any parameters unless routed into a synth via its TransMod slots, or into any Fusor device using its FuseMod slots, the Mod Sequencer engine type feeds note input through to any of the 3 synth channels or into the Key Map page.

This allows you to play a synth while modulating parameters with Mod. Graph values.

If you are using the sequencer only for Fusor devices you do not need to route MIDI note input to the engine, unless you specifically want to use keyboard input for the sequencer (if you have specified keyboard-dependent **Trigger Mode** settings using the Advanced engine type, for example).

Graphs section

Mod Value

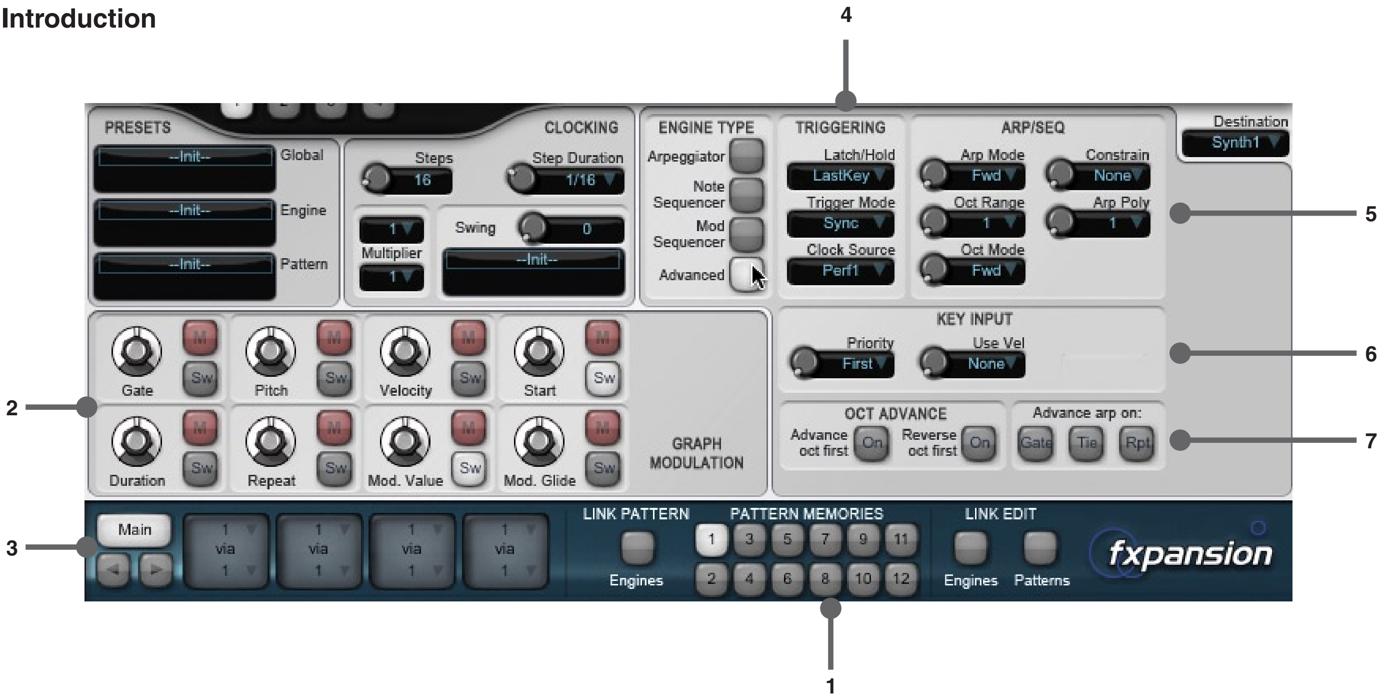
This control offsets the Mod. Value Graph for each step by a positive or negative amount.

Mod Glide

This control offsets the Mod. Glide Graph value for each step by a positive or negative amount.

8:13 Animator: Advanced engine type

Introduction



The Advanced engine type offers many features that are not possible in the other engine types. It may look a little complicated at first, but when you learn it you will have a number of very powerful creative functions at your disposal.

1. Pattern memories

12 pattern memories are available, each with their own clocking and engine settings. Several functions are provided for linking edits between engines and patterns.

2. Graph offset controls

All Graph lanes feature offset controls, which can be modulated.

3. FuseMod slots

4 FuseMod slots are provided in order to modulate the Graph offset controls with any FuseMod source.

4. Advanced Triggering settings

A number additional Triggering settings allow the engine to be used in many creative ways. You can even clock the engine from a FuseMod source, or repurpose the pattern as a programmable keyboard scale for Modulation Graph values.

5. Complex arpeggiator/sequencer functions

Very deep and complex behaviours are possible in the Advanced engine type, such as polyphonic arpeggios and many other permutations not possible in the Arpeggiator engine type or the built-in synth arpeggiators.

6. Control over arpeggiator note priority queue and velocity input

Comprehensive control over the arpeggiator note priority queue is possible, as is the ability to harness keyboard input velocity.

7. Options for step advancing and octave advancing/reversing

These controls allow you to advance steps on certain events, and extra control of the octave and note queue under certain circumstances.

Using the Advanced engine type alongside other engine types

It is possible to switch to the Advanced engine type to adjust certain settings, such as the **Trigger Mode**, and then switch back to the Arpeggiator engine type without losing any settings. You can even switch to new patterns in this way.

Destination

Events from the engine can be routed to any of the 3 synth channels, or into the Key Map page if you wish to use the engine with more than one synth.

Pattern memories

Each engine features 12 pattern memories, each of which is a distinct set of Graph lanes, with independent engine and clocking settings.

The following related functions are also available:

Link Pattern – All Engines

With this button enabled, selecting a pattern in one engine also selects the same pattern in all other engines.

Link Edits – All Engines

Engaging this button results in any changes to the Clocking settings being applied to the same pattern in all engines.

Link Edits – All Patterns

When this button is enabled, any changes to the Clocking settings are applied to all patterns in the current engine.

Graph offset controls and modulation

The Graph offset controls allow you to offset the overall base values of any Graph lanes by positive or negative amounts. These controls can also be modulated by any FuseMod source. For example, you can adjust the overall duration of all notes using the **Duration** control with an LFO or the mod wheel on your keyboard.

Note that these controls and their modulation operate on all patterns in the current engine.

There are only 4 FuseMod slots – the same 4 slots (and their selected sources/scalers) are used for all patterns in all engines. However, each engine (*not* each pattern in each engine) features independent modulation depths from these slots.

The modulation works in the same way as any other FuseMod slot. With a FuseMod slot selected, it is possible to set modulation depths on the Graph offset controls.

Each Graph also features the following controls that operate on the respective Graph in all patterns in the current engine:

Mute

Engaging the **Mute** button causes the graph output for the lane to be muted or disabled, with only the overall base value (specified by its control) being used.

Sw (Swing)

This button affects whether or not the Graph is affected by the current swing template.

Triggering section

Latch / Hold

This is a very important setting, which allows you to switch between several playing styles for arpeggios and step-sequencing within the Advanced engine type.

You should use 'LastKey' for note-sequencing applications, and the other settings for arpeggiator usage or modulation-sequencing.

Off	Non-latching arpeggiator behaviour When one or more notes are held down, the arp/seq outputs events based on the input. When the notes are released, the event output stops.
Latch	A simple latching hold mechanism All received notes are held indefinitely, until the **Latch/Hold** setting is changed or until the **Panic** button is clicked.
AutoHold	This setting holds notes that are played (and released) within a short duration of each other. The notes are held until new notes are played. When this happens, all existing notes are released and the new notes are held instead. This is great for starting arpeggios by playing chords and releasing the notes quickly to do other things.
LastKey	This mode should be used when you want to use the Advanced engine type as a note sequencer. It offsets all pitch events from the last key played (C3 is used if no MIDI notes have been received).

Trigger Mode

This drop-down menu specifies how the sequencer or arpeggiator pattern is triggered and played.

All modes *require the Fusor transport to be started*, with the exception of the 'Advance', 'NoteScale', 'NoteWrap' and 'Ext. Mod.' modes. However, the transport still affects their behaviour, depending on whether you are using the engine for note sequencing or modulation sequencing. See the descriptions of these modes below for more details.

Sync (default)	Plays in sync with Fusor's transport.
Gate	Plays in sync with Fusor's transport, only when one or more notes are held down. If you are sequencing modulation values, output values are only generated while notes are held.
Trig	Plays continually, retriggering whenever new notes are played after all notes are off.
Shot	Plays only while a note is held down, retriggering whenever notes are played after all notes are off. If you are sequencing modulation values, output values are only generated while notes are held.
Trig-P	Plays continually, retriggering whenever any note-on is received.
Shot-P	Plays only when a note is held down, retriggering whenever any note-on is received. If you are sequencing modulation values, output values are only generated while notes are held.
Advance	Advances one step each time a note-on is received. If the transport is stopped, steps are advanced with the notes received, but only modulation values are generated – no sequencer notes are generated and the played notes pass through to the specified **Destination** unaffected. If the transport is started, each received note not only advances a step, but also results in the step's pitch events to be generated and sent to the specified Destination, rather than the played notes passing through.
NoteScale NoteWrap	Keyboard input plays steps. See 'NoteScale and NoteWrap' section below.
Ext. Mod	The sequencer is clocked from an FuseMod source, chosen by the **Clock Source** control. The transport must be running for sequencer note events to be generated – if it is stopped, only modulation values are generated.

NoteScale and NoteWrap

These **Trigger Mode** settings turn Animator's sequencer into a programmable keyboard scale for Modulation Graph values. Notes on the keyboard select steps in the sequencer, instead of the steps cycling at the clock rate.

Steps are triggered with the notes received, and modulation values are generated. Input MIDI notes are sent to the engine's **Destination,** offset by the step's Pitch Graph value.

NoteScale

The keyboard is split according to the number of steps in the pattern, with the lowest note (MIDI note 0) playing the first step in the sequence, and the highest playing the last step. Each time a new note is played, the corresponding step is played.

If **Steps** is set to 128, you can define the Modulation Graph value for each key on the keyboard. Setting it to a lower number divides up the keyboard range proportionally – so for a Steps setting of 32, the first step is triggered by the lowest 4 keys, while the next 4 keys play the next step, and so on.

NoteWrap

The keyboard is split into a scale with a length dictated by the pattern length, with the scale wrapping around up the keyboard. The lowest note (note 0) plays the first step in the sequence, note 1 the next step, and so on.

If you set up a 12-step pattern, each note in the first octave plays steps 1-12. The next octave plays steps 1-12 an octave higher, and so on.

Clock Source

If the **Trigger Mode** is set to 'Ext. Mod', the source selected in the **Clock Source** drop-down menu is used to clock the pattern. This is intended for more experimental uses of Animator.

Any available FuseMod source can be used as the external clock source, meaning that you can clock it with a MonoMod or synth LFO, for example.

Arp/Seq section

Arp. Mode & Oct. Range

These controls are identical to those in the Arpeggiator and Note Sequencer engine types. See section 8:10 for a summary of the various **Arp. Mode** settings, while **Oct. Range** is discussed in section 8:11.

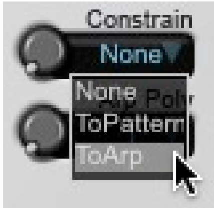
Oct. Mode

This control sets the order in the octave queue when using more than 1 octave. If **Oct. Range** is set to 1, the **Oct. Mode** control has no effect.

See section 8:11 for a summary of the possible Oct. Mode settings.

Constrain

This control determines the relationship between the arpeggiator sequence length and the engine pattern length. It is useful if you want a 7-note arpeggio to 'fit' with a 16-step pattern, for example.



Off	There is no link between the two lengths - the arpeggiator cycles freely through the note queue.
ToPattern	The arpeggiator is reset to the start of the note queue each time the pattern retriggers.
ToArp	The pattern is reset to its start each time the arpeggiator reaches the beginning of its note queue.

Arp. Poly

Animator allows polyphonic arpeggios – this control determines how many notes the arpeggiator generates at each step from the available notes in the note queue.

1	Traditional monophonic arpeggiator behaviour
2	Plays 2 notes from the note queue together
3	Plays 3 notes from the note queue together
Full	Plays all notes in the note queue on every step This is equivalent to the 'chord mode' found on some arpeggiators.

Key Input section

Priority

This control sets the order of the notes in the arpeggiator queue (notes that are held down). Held notes are sequenced in the order dictated by the **Priority** parameter: the sequence is played *starting with* the note indicated by the parameter (and by the **Arp. Mode** setting).

6 settings are available: 'Newest', 'Oldest', 'Highest', 'Lowest', 'Hardest' and 'Softest'.

For example, if you want the arpeggio to follow the order in which you play the notes, set the Priority control to 'Oldest'. This way, the first note you held down will always be the first in the arpeggiator sequence, the second note you play will always play second, and so on (as long as the Arp. Mode control is set to 'Fwd').

Use Vel

This control allows you to specify whether the velocity of received MIDI notes is used in arpeggiator or note-sequencer events.

In the 'Off' position, the velocity of incoming MIDI notes is ignored.

In any of the following modes, input velocities are added to the velocity Graph as follows:

First	The velocity of the first note is used for all arp/seq events
All	The velocity of all input notes is used – whenever the note is triggered by Graph events, the original MIDI note's velocity is added to the relevant velocity value for the step.

Oct Advance section

Advance Arp On

These buttons specify whether or not the arpeggiator advances through its sequence on the following step events:

Muted	Steps with the Gate Graph turned off
Tied	Steps with a long enough duration to be combined or 'tied' with the next step
Rpt'd	Repeat Graph events within a step

Advance Oct. First
Reverse Oct. First

These buttons control how the arpeggiator organises its note queue when multiple octaves are in use (when **Oct. Range** is set to 2 or higher).

When **Advance Oct. First** is enabled, the arpeggiator plays through the available octaves in the note queue first, before advancing through the available notes.

For example, with an **Oct. Range** of 3, and the notes C1, E1 and G1 held down:

• **Advance Oct. First** off:

G3
E3
C3
G2
E2
C2
G1
E1
C1

• **Advance Oct. First** on:

G3
E3
C3
G2
E2
C2
G1
E1
C1

Reverse Oct. First takes effect only when both **Arp. Mode** and **Oct. Mode** are set to ‘FwdRev’ or ‘RevFwd’, and dictates whether the octave queue is played forwards or backwards first.

With an **Oct. Range** of 3, **Arp. Mode** and **Oct. Mode** both set to ‘FwdRev and the notes C1, E1 and G1 held down:

• **Reverse Oct. First** off:

G3
E3 E3
C3
G2 G2
E2 E2 E2 E2
C2 C2
G1
E1 E1
C1

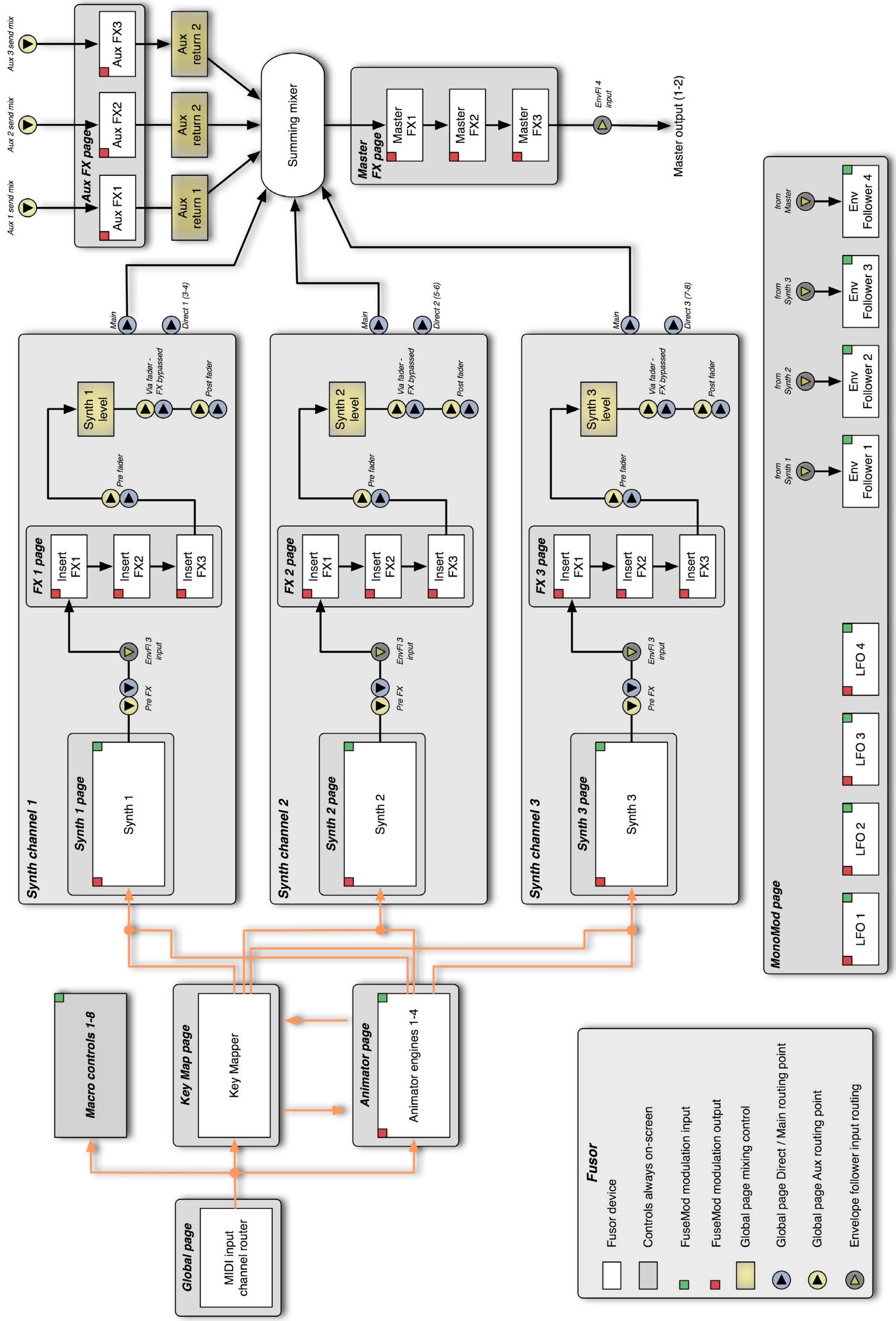
• **Reverse Oct. First** on:

This setting creates a ‘glissando’ of all possible notes.

G3
E3 E3
C3 C3
G2 G2
E2 E2
C2 C2
G1 G1
E1 E1
C1

8:14 Fusor signal flow

The following signal flow diagram represents Fusor's signal flow. FuseMod modulation inputs/outputs are shown where appropriate.



Appendix 1: Standalone application and plugin usage

Using the synth standalone applications

The DCAM: Synth Squad synths and Fusor are provided as standalone applications, which can be useful for live performance. The FX versions of the synths (MIDI-controlled audio effects) are not available as standalone applications.

They are also very useful as a means of authorizing DCAM: Synth Squad, as the plugin-initializing mechanisms of many hosts can disrupt proper authorization.

Please note that if you require advanced functionality such as multiple outputs or synchronization (except slaving to MTC), you should use a suitable host, such as Plogue Bidule (www.plogue.com).

Windows operation

You'll find the Strobe, Cypher, Amber and Fusor standalone applications in these locations:

Start • Programs • FXpansion • DCAM Synth Squad (if you chose to install a start menu shortcut)

C:\Program Files\FXpansion\<instrument> (for example, C:\Program Files\FXpansion\Strobe)

The Windows standalone application is a basic host program that runs the Synth Squad VST plugins. In order to function, it requires the various .dll files to be in the same folder as the standalone application .exe files.

Mac operation

Run the Strobe, Cypher, Amber or Fusor applications in your Applications folder.

The Mac standalone application is a basic host program that runs the Synth Squad AU plugins. In order to function, it requires that the AU plugins exist in the Library/Audio/Plug-ins/Components folder.

Tempo and time signature

When running the standalone applications, you cannot set the tempo and time signature.

If you want to use synced LFOs, envelopes, and other parameters, you must use the DCAM: Synth Squad plugins in a host.

I/O Settings

The I/O Settings menu in the standalone application is located in the standard OS menu. It alllows you to make various settings for audio and MIDI input and output.

Audio Devices

Using this sub-menu, select the desired audio interface device with which you want to use the standalone application.

On Windows, ASIO and MME devices are shown. For best results, a good ASIO device is recommended. If your audio interface lacks an ASIO driver, it is often possible to achieve very good results using the ASIO4ALL free universal ASIO driver, which can be downloaded from www.asio4all.com

On Mac OSX, all installed CoreAudio devices are shown in the menu.

Audio Channels

This part of the sub-menu shows the available stereo output pairs for the currently selected Audio Device.

Currently, only 1 stereo channel can be selected – the standalone application does not support multiple output channels at this time. If you need to set up more than a stereo output, please use the plugins in a suitable host that supports this feature.

Audio Buffer Size

The standalone application attempts to detect the current buffer size setting for your audio device. In some cases this may not be possible, due to the device's driver not properly responding to the standalone application's request. In such cases, set the Audio Buffer Size to the same value as that defined in your audio device's control panel.

Sample Rate

Again, the standalone application attempts to detect the current sample rate setting for your audio device. In some cases this may not be possible, due to the device's driver not properly responding to the standalone application's request. In such cases, set the Sample Rate to the same value as that defined in your audio device's control panel.

MIDI Ports

This sub-menu shows all detected MIDI ports in your system. Click a MIDI port to enable MIDI input from the port to the standalone application.

You can select more than one MIDI port in order to combine MIDI input devices if required.

Sync to MTC

Enabling the Sync to MTC function results in the standalone application responding to MIDI timecode in order to synchronize with another device. You must make sure that the currently enabled MIDI ports include the port on which the MTC is being transmitted.

The standalone application can only sync to an external MTC source. It cannot send out its own MTC or other sync methods such as MIDI clock. If you need such functionality, please use the plugins in a suitable host that supports these features.

ASIO Control Panel... (Windows only)

This function brings up the ASIO control panel for your audio device, in which you can adjust its settings if required.

This function is not available on Mac – in order to adjust your audio card's settings on Mac, you must directly launch its particular companion software/control panel.

Using DCAM: Synth Squad as RTAS plugins in Digidesign Pro Tools 7.x

DCAM: Synth Squad requires Pro Tools 7 or later. For best performance, version 7.1 or later is highly recommended. Pro Tools 6.x, or any earlier versions are not compatible.

Launching a plugin

1. Insert the Strobe, Cypher, Amber or Fusor RTAS plugin onto a stereo Instrument track.

Setting up additional output monitoring for Fusor

1. Create a stereo Aux track.
2. Set the input of the Aux track as the relevant stereo output from Fusor (after Fusor is inserted into the project, its outputs become available as track input sources).
3. Repeat this process for as many additional Fusor outputs that you need to monitor.

If you encounter any problems performing the steps above, please consult your host's documentation or contact the manufacturer's tech support channels – the DCAM: Synth Squad plugins operate in the same way as any other instrument plugins.

Recording the output as audio

To do this you must use Pro Tools' routing and recording functions (see the Pro Tools documentation).

Using DCAM: Synth Squad as AU plugins in Apple Logic 7.2 or later

Launching a plugin

1. Insert the Strobe, Cypher, Amber or Fusor AU plugin onto an audio instrument channel.
2. Create a track in Logic's Arrange window, corresponding to the audio instrument channel on which you inserted the plugin. You can now play it from this track.

Setting up additional output monitoring for Fusor

The instrument channel's output plays the output of all Fusor channels. To assign the other channels, follow this procedure:

1. Create a stereo Aux channel.
2. Set the input of the Aux channel as the relevant stereo output from Fusor (after it is inserted into the project, its outputs become available as track input sources).
3. Repeat this process for as many additional outputs that you need to monitor.

If you encounter any problems performing the steps above, please consult your host's documentation or contact the manufacturer's tech support channels – the DCAM: Synth Squad plugins operate in the same way as any other instrument plugins.

Recording the output as audio

To do this you must use Logic's bounce functions (see the Logic documentation).

Using DCAM: Synth Squad as AU plugins in Apple Logic 8

Launching a plugin

1. Create a new Instrument track and insert the Strobe, Cypher, Amber or Fusor AU plugin onto it. You can now play the plugin from this track.

Setting up additional output monitoring for Fusor

The instrument channel's output plays the output of all Fusor channels. To assign the other channels, follow this procedure:

1. Create a stereo Aux channel.
2. Set the input of the Aux channel as the stereo output from Fusor (after it is inserted into the project, its outputs become available as track input sources).
3. Repeat this process for as many additional outputs that you need to monitor.

If you encounter any problems performing the steps above, please consult your host's documentation or contact the manufacturer's tech support channels – the DCAM: Synth Squad plugins operate in the same way as any other instrument plugins.

Recording the output as audio

To do this you must use Logic's bounce functions (see the Logic documentation).

Using DCAM: Synth Squad as VST or AU plugins in Ableton Live

On Mac, Ableton Live supports both VST and AU plugins. There is no real difference between using either format, although you may want to use the VST plugins if you intend to share projects with Windows users (Live only supports VST plugins on Windows).

Launching a plugin

1. Drag and drop the Strobe, Cypher, Amber or Fusor VST or AU plugin (Mac only) from the Plug-In Devices list into the Live workspace.

Setting up additional output monitoring for Fusor

1. Create an audio track.
2. Click the 'I-O' button on the right hand side of the Live interface to reveal each track's input/output settings.
3. Set the 'Audio From' selector to the Fusor track.
4. Set the selector immediately underneath this to the required output.

If you encounter any problems performing the steps above, please consult your host's documentation or contact the manufacturer's tech support channels – the DCAM: Synth Squad plugins operate in the same way as any other instrument plugins.

Recording the output as audio

To do this you must use Live's export/merge/freeze functions.

Using DCAM: Synth Squad as AU plugins in Digital Performer

Launching a plugin

1. In a new project, add Strobe, Cypher, Amber or Fusor as an instrument track, using the Project/Add Track/ Instrument Track menu option. For Strobe, for instance, select Strobe (stereo).
2. In the Sequence page, create a MIDI track and assign its output to the plugin – for example Strobe-1-1.

Setting up additional output monitoring

1. Add a new aux track.

3. Assign the aux track's input to one of the Fusor Bundles in the New Stereo Bundle menu, which shows Fusor's stereo outputs.

If you encounter any problems performing the steps above, please consult your host's documentation or contact the manufacturer's tech support channels – the DCAM: Synth Squad plugins operate in the same way as any other instrument plugins.

Recording the output as audio

To do this you must use DP's export functions.

Using DCAM: Synth Squad as VST plugins in Cubase 4

Launching a plugin

1. With a project open, bring up the VST Instruments panel.
2. Click on an instrument slot and select the Strobe, Cypher, Amber or Fusor plugin. If Cubase asks you whether it should create a MIDI track routed to the plugin, click 'Yes'.
3. Cubase 4 does not automatically create additional Fusor outputs in its mixer. To enable the additional outputs, click the output button in the VST Instruments panel for Fusor (it is to the right of the 'e' button that displays the plugin interface). In the pop-up list that appears, you can enable individual or all outputs. This button is described on page 63 of the Cubase 4.1 plug-in reference PDF manual.
4. Assign a MIDI track in the Project Window to the plugin if you did not click 'Yes' in step 2.

If you encounter any problems performing the steps above, please consult your host's documentation or contact the manufacturer's tech support channels – the DCAM: Synth Squad plugins operate in the same way as any other instrument plugins.

Recording the output as audio

To do this you must use Cubase's export functions.

Using DCAM: Synth Squad as VST plugins in Sonar

Launching a plugin

1. With a project open, use the Insert menu to insert Strobe, Cypher, Amber or Fusor as a soft synth:
   Insert • Soft Synths • [VST plugins folder name] • *plugin*
2. Choose the appropriate output configuration – the synths are stereo, while Fusor has additional stereo outputs.
3. If the ‘MIDI Source’ checkbox was not left checked in the dialog box, you need to create a MIDI track and route its output to the plugin.

If you encounter any problems performing the steps above, please consult your host’s documentation or contact the manufacturer’s tech support channels – the DCAM: Synth Squad plugins operate in the same way as any other instrument plugins.

Recording the output as audio

To do this you must Sonar’s mixdown functions.

Using DCAM: Synth Squad as VST plugins in FL Studio

Launching a plugin

1. The DCAM: Synth Squad plugins may not be displayed in the plugin list by default. To make them part of the list select ‘More…’ to popup a list of all available plugins. From the bottom-right of this window click ‘Refresh’ then ‘Fast Scan (recommended)’.
2. Enable the checkboxes next to the Strobe, Cypher, Amber and Fusor plugins which are shown in red (meaning that it is a newly found plugin).
3. Now you can add the required plugin to the project by selecting ‘Channels • Add one…’ and then selecting the plugin.
4. Assign the plugin to an FX track, using the ‘Channel Settings’ window. This FX track plays its output (main output for Fusor).

Setting up additional output monitoring for Fusor

1. Enable the additional outputs using the down-arrow menu, just underneath the red plugin icon in the top-left corner of the plugin window.
2. The additional outputs occupy the susequent FX tracks from the FX track you originally specified for Fusor’s main output (1-2).

If you encounter any problems performing the steps above, please consult your host’s documentation or contact the manufacturer’s tech support channels – the DCAM: Synth Squad plugins operate in the same way as any other instrument plugins.

Recording the output as audio

To do this you must use FL Studio’s export/mixdown functions.

Using DCAM: Synth Squad as VST plugins in Reaper

Launching a plugin

1. Create a new track and bring up the FX Browser.
2. Add the Strobe, Cypher, Amber or Fusor plugin (located in the VSTi section of the FX Browser).

Setting up additional output monitoring for Fusor

1. Bring up the FX Chain for the track on which you inserted Fusor.
2. Right-click on the Fusor VSTi in the FX Chain and click on the ‘Build multichannel routing for output of selected FX’ function in the menu that appears.

Recording the output as audio

To do this you must use Reaper’s audio mixdown features.

Potential problems

Missing plugin

Windows

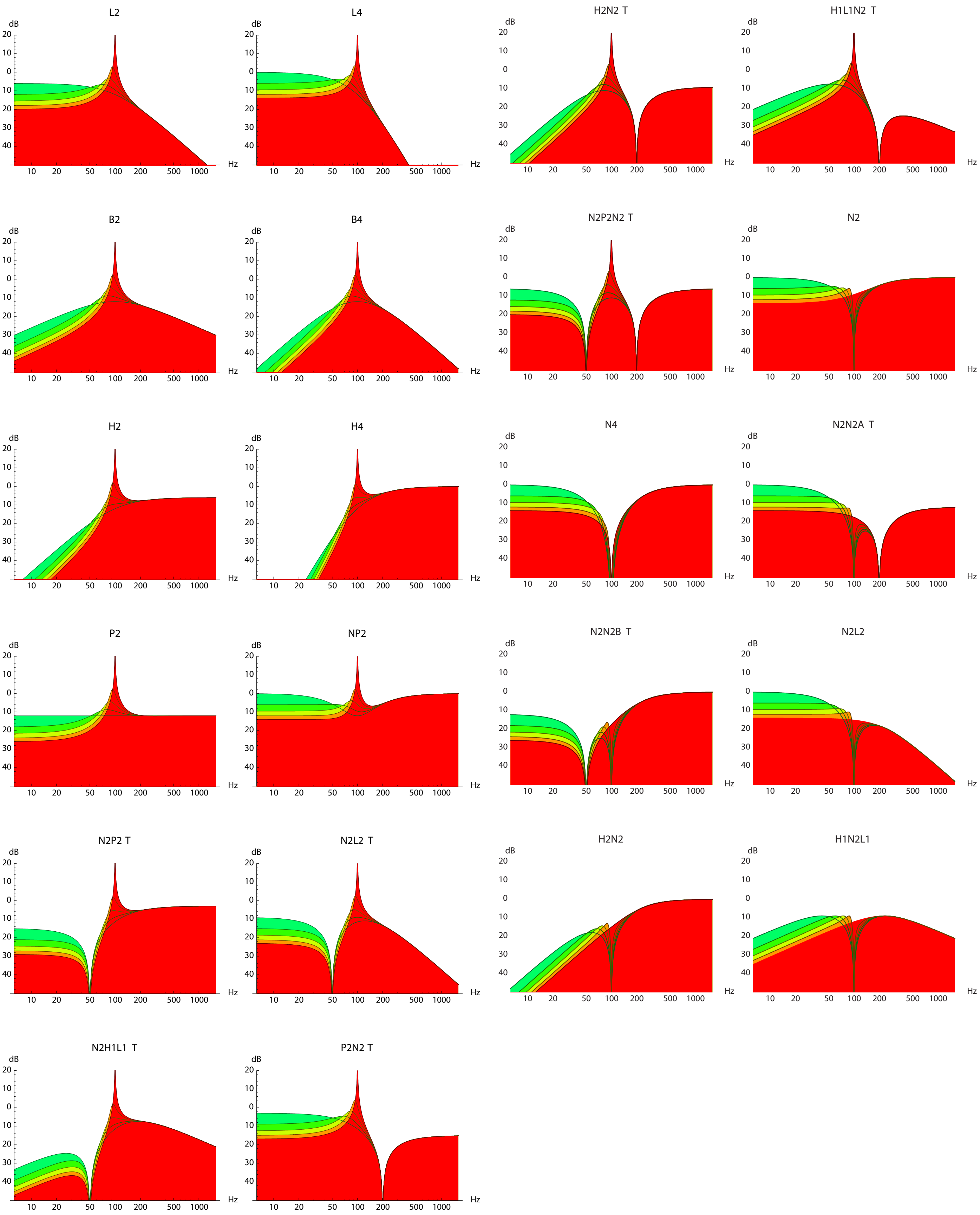
- You must make sure that your host application is set to use the VST plugins folder to which you installed DCAM: Synth Squad's plugins. See your host's documentation or contact its manufacturer's tech support if you do not know how to do this.
- By default, the installer suggests the VST folder defined in the HKEY_LOCAL_MACHINE • SOFTWARE • VST registry key. Unless you changed this location during the installer, this is where the plugins are installed.
- If you cannot find the VST plugins, you can copy them into the VST plugins folder of your choice from the following folders:

  C:\Program Files\FXpansion\<instrument> (for example, C:\Program Files\FXpansion\Strobe)

Mac

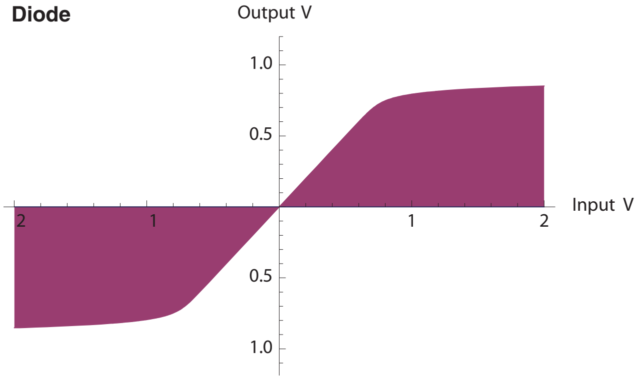
- The VST plugins are installed to Library • Audio • Plug-Ins • VST
- The AU plugins are installed to Library • Audio • Plug-Ins • Components
- There are not many situations when your host will not find the DCAM: Synth Squad plugins. However, if you use Logic, you need to make sure the AU plugins are properly validated in the Logic AU Manager, located in the Logic 8 Preferences menu. Please see the Logic documentation if you are unsure how to do this.

Appendix 2: Strobe filter modes

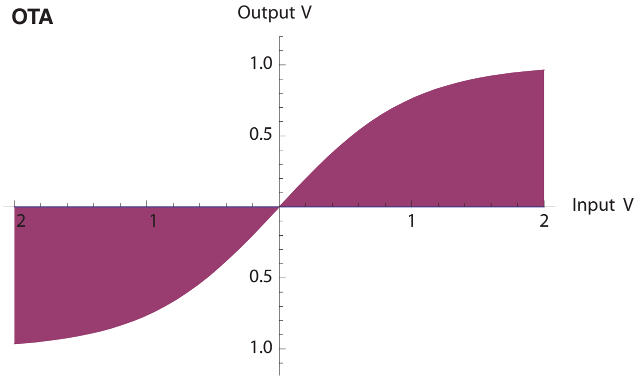


Appendix 3: Cypher waveshaper modes

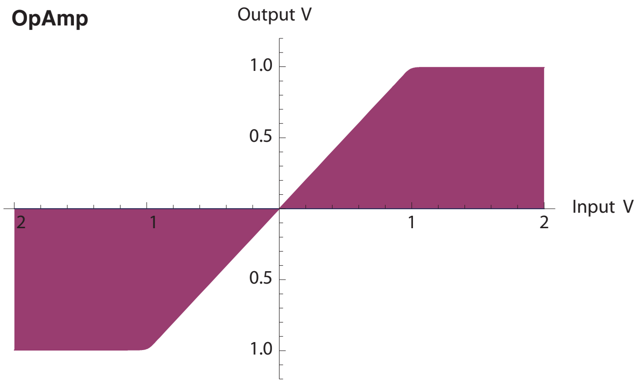
Diode



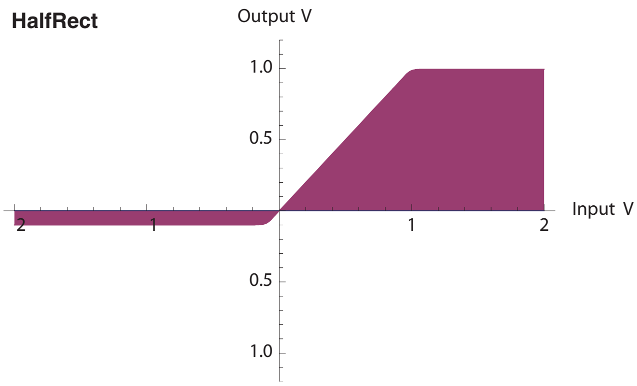
OTA



OpAmp



HalfRect



Appendix 4: Fusor – Overloud Breverb algorithms

This adapted version of Overloud's Breverb provides high-end artificial reverberation effects within Fusor.

The controls are cut down from those available in the full Breverb product, but all 4 algorithms are present with their most important controls.

Breverb Hall

Time (Sec)

Sets the duration of the reverberation tail. It is also influenced by the **Size** parameter.

Size

Sets the rate of build-up diffusion after the initial period, which is controlled by the **Diffusion** parameter. It also acts as a master control for **Time** and **Spread**. Despite its name, the apparent size of the space created is actually a combination of the settings of the **Size**, **Shape** and **Spread** controls.

Diffusion

Controls the degree to which the initial echo density increases over time.

Shape

Works together with the **Spread** parameter to control the overall ambience of the reverberation created by the Breverb Room algorithm. It specifically determines the contour of the reverberation envelope. With the **Shape** control all the way down, reverberation builds explosively and decays very quickly. As the control is is increased, reverberation builds up more slowly and sustains for the time set by the **Spread** parameter.

Spread

Controls the duration of the initial contour of the reverberation envelope. Low **Spread** settings result in a rapid onset of reverberation at the beginning of the envelope, with little or no sustain, while higher settings spread out both the build-up and sustain.

Predelay (Sec)

Sets the amount of time that elapses between the input signal and the onset of reverberation. It can be used to create a sense of distance and volume within an acoustic space.

Low
High

The **Low** and **High** parameters can be used to tweak the frequency response of the reverb.

• Low (kHz)

Sets the frequency under which the reverberation is attenuated.

• High (kHz)

Sets the frequency over which the reverberation is attenuated.

Breverb Inverse

Time (Sec)

Sets the duration of the reverberation. This time, added to the **Predelay** time, is the time that elapses from the direct sound to the end of the reverberation process.

Diffusion

Controls the degree to which the initial echo density increases over time.

Predelay (Sec)

Sets the amount of time that elapses between the input signal and the onset of reverberation. It can be used to create a sense of distance and volume within an acoustic space.

Low
High

The **Low** and **High** parameters can be used to tweak the frequency response of the reverb.

• Low (kHz)

Sets the frequency under which the reverberation is attenuated.

• High (kHz)

Sets the frequency over which the reverberation is attenuated.

Breverb Room

Time (Sec)

Sets the duration of the reverberation tail. It is also influenced by the **Size** parameter.

Size

Sets the apparent size of the acoustic space being emulated by the algorithm. Values from minimum to half way up are typical of the ambience of a recording studio.

Diffusion

Controls the degree to which the initial echo density increases over time.

Decay

Balances between the late reverberation and the early reflections. When the **Decay** control is turned down fully, only the early reflections are present. When it is increased, late reverberations are gradually added.

Predelay (Sec)

Sets the amount of time that elapses between the input signal and the onset of reverberation. It can be used to create a sense of distance and volume within an acoustic space.

Low
High

The **Low** and **High** parameters can be used to tweak the frequency response of the reverb.

• Low (kHz)

Sets the frequency under which the reverberation is attenuated.

• High (kHz)

Sets the frequency over which the reverberation is attenuated.

Breverb Plate

Time (Sec)

Sets the duration of the reverberation tail. It is also influenced by the **Size** parameter.

Size

Sets the apparent size of the plate emulated by the algorithm.

Diffusion

Controls the degree to which the initial echo density increases over time.

Predelay (Sec)

Sets the amount of time that elapses between the input signal and the onset of reverberation. It can be used to create a sense of distance and volume within an acoustic space.

Shape

Determines the contour of the reverberation envelope. With the **Shape** control turned all the way down, reverberation builds explosively and decays very quickly. As the control is raised, reverberation builds up more gradually and sustains longer.

Low
High

The **Low** and **High** parameters can be used to tweak the frequency response of the reverb.

• Low (kHz)

Sets the frequency under which the reverberation is attenuated.

• High (kHz)

Sets the frequency over which the reverberation is attenuated.

Appendix 5:
Host automation parameters

Note that the VST plugin specification does not allow very long parameter names (a maximum of 7 characters).
In the VST plugin versions of Strobe, Cypher and Amber, parameter names are shortened.

Full-length names (shown in the 2nd column in the tables below) are available only in AU and RTAS plugin formats.

Strobe

0	Perf1	Performance controller 1
1	Perf2	Performance controller 2
2	PerfX	(reserved for future use)
3	PerfY	(reserved for future use)
4	Cutoff	Filter Cutoff
5	Res	Filter Resonance
6	FiltDrive	Filter Drive
7	FiltMode	Filter Mode
8	CutoffEnv	Direct Cutoff modulation from Mod Envelope
9	CutoffLfo	Direct Cutoff modulation from LFO
10	CutoffKey	Direct Cutoff modulation from KeyTrack
11	FiltPwr	Filter Power
12	Amp	VCA Amp
13	Pan	VCA Pan
14	Level	VCA Level
15	AnalogNoise	Analogue
16	Pitch	Osc Pitch
17	Fine	Osc Fine Pitch
18	PitchEnv	Direct Pitch modulation from Mod Envelope
19	PitchLfo	Direct Pitch modulation from LFO
20	PitchKey	Direct Pitch modulation from KeyTrack
21	SyncPitch	Osc Sync
22	OscStack	Osc Stack
23	OscDetune	Osc Detune
24	Saw	Saw waveform Level
25	Pls	Pulse waveform Level
26	Noise	Noise Level
27	SubSin	Sub-osc Sine Level
28	SubSinOct	Sub-osc Sin Octave
29	SubTri	Sub-osc Triangle Level
30	SubTriOct	Sub-osc Triangle Octave
31	SubSaw	Sub-osc Saw Level
32	SubSawOct	Sub-osc Saw Octave
33	SubPls	Sub-osc Pulse Level
34	SubPlsOct	Sub-osc Pulse Octave
35	Pw	Pulse Width (main pulse)
36	PwEnv	Direct Pulse Width modulation from Mod Envelope
37	PwLfo	Direct Pulse Width modulation from LFO
38	PwKey	Direct Pulse Width modulation from KeyTrack
39	SubPw	Sub-osc Pulse Width
40	LfoMode	LFO Mode
41	LfoRate	LFO Rate
42	LfoSwing	LFO Swing
43	LfoPw	LFO Pulse Width
44	LfoPhase	LFO Phase
45	LfoSync	LFO Sync
46	LfoMono	LFO Mono
47	LfoGate	LFO Gate
48	RampDelay	Ramp Delay
49	RampRise	Ramp Rise
50	RampMultiplier	Ramp Mult
51	RampSync	Ramp Sync
52	RampGate	Ramp Gate
53	ModA	Mod Envelope Attack
54	ModD	Mod Envelope Decay
55	ModS	Mod Envelope Sustain
56	ModR	Mod Envelope Release
57	ModLoop	Mod Envelope Loop
58	ModLinear	Mod Envelope Linear
59	ModSync	Mod Envelope Sync
60	ModGate	Mod Envelope Gate
61	AmpA	Amp Envelope Attack
62	AmpD	Amp Envelope Decay
63	AmpS	Amp Envelope Sustain
64	AmpR	Amp Envelope Release
65	AmpLoop	Amp Envelope Loop
66	AmpLinear	Amp Envelope Linear
67	AmpSync	Amp Envelope Sync
68	AmpGate	Amp Envelope Gate
69	ArpRate	Arpeggiator Rate
70	ArpRange	Arpeggiator Range
71	ArpNoteMode	Arpeggiator Note
72	ArpOctMode	Arpeggiator Oct
73	ArpGate	Arpeggiator Gate
74	ArpOn	Arpeggiator Power
75	Tune	Settings Tune
76	Hold	Keying Hold
77	BendUp	Settings Bend Up
78	BendDn	Settings Bend Dn
79	Priority	Keying Priority
80	VelGlide	Vel Glide
81	PitchGlide	Pitch Glide
82	GlideMode	Glide Mode
83	Legato	Glide Legato
84	Retrigger	Glide Retrig

Cypher

0	Perf1	Performance controller 1
1	Perf2	Performance controller 2
2	PerfX	(reserved for future use)
3	PerfY	(reserved for future use)
4	FiltCutoff	Filter Cutoff (adjusts Filter1 and Filter2 cutoff)
5	Filt1Res	Filter 1 Resonance
6	Filt2Res	Filter 2 Resonance
7	FiltEnv1	Filter Cutoff direct modulation from Mod Envelope 1
8	FiltEnv2	Filter Cutoff direct modulation from Mod Envelope 2
9	FiltRouting	Filter Route
10	FiltSpread	Filter Spread
11	Filt1Scale	Filter 1 Scale
12	Filt1Drive	Filter 1 Drive
13	Filt1Mode	Filter 1 Mode
14	Filt1Type	Filter 1 Type
15	Filt1Key	Filter 1 Scale direct modulation from KeyTrack
16	Filt1Fm3	Filter 1 FM from 3
17	Filt1Pwr	Filter 1 Power
18	Filt2Scale	Filter 2 Scale
19	Filt2Drive	Filter 2 Drive
20	Filt2Mode	Filter 2 Mode
21	Filt2Type	Filter 2 Type
22	Filt2Key	Filter 2 Scale direct modulation from KeyTrack
23	Filt2Fm3	Filter 2 FM from 3
24	Filt2Pwr	Filter 2 Power
25	Osc2Sync1	Osc2 Sync to 1
26	Osc2Fm3	Osc2 FM from 3
27	Osc3Sync1	Osc3 Sync to 1
28	Osc3Wm2	Osc3 WM from 2
29	Amp	VCA Amp
30	Pan	VCA Pan
31	Level	VCA Level
32	AnalogNoise	Analogue
33	OscPitch	Master Pitch (Coarse)
34	OscFine	Master Pitch (Fine)
35	Shpr1Drive	Shaper 1 Drive
36	Shpr1Lp	Shaper 1 LPF
37	Shpr1Type	Shaper 1 Type
38	Shpr1Post	Shaper 1 Post
39	Shpr1Pwr	Shaper 1 Power
40	Shpr2Drive	Shaper 2 Drive
41	Shpr2Lp	Shaper 2 LPF
42	Shpr2Type	Shaper 2 Type
43	Shpr2Post	Shaper 2 Post
44	Shpr2Pwr	Shaper 2 Power
45	Osc1Wave	Osc1 Wave
46	Osc1Scale	Osc1 Scale
47	Osc1Low	Osc1 Low
48	Osc1Beat	Osc1 Beat
49	Osc1BeatSync	Osc1 Beat Sync
50	Osc1Key	Osc1 Key
51	Osc1Phase	Osc1 Phase
52	Osc1Reset	Osc1 Rset (Reset)
53	Osc1Pink	Osc1 Pink
54	Osc1Blend	Osc1 Blend (Osc2 S+H by Osc1 / Osc1 / Noise)
55	Osc1*2	Ring Mod Osc1*2
56	Osc1Level	Osc1 Level
57	Osc1Mix	Osc1 Filter Mix
58	Osc1Pwr	Osc1 Power
59	Osc2Wave	Osc2 Wave
60	Osc2Scale	Osc2 Scale
61	Osc2Low	Osc2 Low
62	Osc2Beat	Osc2 Beat
63	Osc2BeatSync	Osc2 Beat Sync
64	Osc2Key	Osc2 Key
65	Osc2Phase	Osc2 Phase
66	Osc2Reset	Osc2 Rset (Reset)
67	Osc2Pink	Osc2 Pink
68	Osc2Blend	Osc2 Blend (Osc3 S+H by Osc2 / Osc2 / Noise)
69	Osc2*3	Ring Mod Osc2*3
70	Osc2Level	Osc2 Level
71	Osc2Mix	Osc2 Filter Mix
72	Osc2Pwr	Osc2 Power
73	Osc3Wave	Osc3 Wave
74	Osc3Scale	Osc3 Scale
75	Osc3Low	Osc3 Low
76	Osc3Beat	Osc3 Beat
77	Osc3BeatSync	Osc3 Beat Sync
78	Osc3Key	Osc3 Key
79	Osc3Phase	Osc3 Phase
80	Osc3Reset	Osc3 Rset (Reset)
81	Osc3Pink	Osc3 Pink
82	Osc3Blend	Osc3 Blend (Osc1 S+H by Osc3 / Osc3 / Noise)
83	Osc3*1	Ring Mod Osc3*1
84	Osc3Level	Osc3 Level
85	Osc3Mix	Osc3 Filter Mix
86	Osc3Pwr	Osc3 Power
87	Lfo1Mode	LFO 1 Mode
88	Lfo1Rate	LFO 1 Rate
89	Lfo1Swing	LFO 1 Swing
90	Lfo1Pw	LFO 1 Pulse Width
91	Lfo1Phase	LFO1 Phase
92	Lfo1Sync	LFO 1 Sync
93	Lfo1Mono	LFO 1 Mono
94	Lfo1Gate	LFO 1 Gate
95	Lfo2Mode	LFO 2 Mode
96	Lfo2Rate	LFO 2 Rate
97	Lfo2Swing	LFO 2 Swing
98	Lfo2Pw	LFO 2 Pulse Width
99	Lfo2Phase	LFO 2 Phase

100	Lfo2Sync	LFO 2 Sync
101	Lfo2Mono	LFO 2 Mono
102	Lfo2Gate	LFO 2 Gate
103	RampDelay	Ramp Delay
104	RampRise	Ramp Rise
105	RampMultiplier	Ramp Mult
106	RampSync	Ramp Sync
107	RampGate	Ramp Gate
108	Env1A	Mod Envelope 1 Attack
109	Env1D	Mod Envelope 1 Decay
110	Env1S	Mod Envelope 1 Sustain
111	Env1R	Mod Envelope 1 Release
112	Env1Loop	Mod Envelope 1 Loop
113	Env1Linear	Mod Envelope 1 Linear
114	Env1Sync	Mod Envelope 1 Sync
115	Env1Gate	Mod Envelope 1 Gate
116	Env2A	Mod Envelope 2 Attack
117	Env2D	Mod Envelope 2 Decay
118	Env2S	Mod Envelope 2 Sustain
119	Env2R	Mod Envelope 2 Release
120	Env2Loop	Mod Envelope 2 Loop
121	Env2Linear	Mod Envelope 2 Linear
122	Env2Sync	Mod Envelope 2 Sync
123	Env2Gate	Mod Envelope 2 Gate
124	AmpA	Amp Envelope Attack
125	AmpD	Amp Envelope Decay
126	AmpS	Amp Envelope Sustain
127	AmpR	Amp Envelope Release
128	AmpLoop	Amp Envelope Loop
129	AmpLinear	Amp Envelope Linear
130	AmpSync	Amp Envelope Sync
131	AmpGate	Amp Envelope Gate
132	ArpRate	Arpeggiator Rate
133	ArpRange	Arpeggiator Range
134	ArpNoteMode	Arpeggiator Note
135	ArpOctMode	Arpeggiator Oct
136	ArpGate	Arpeggiator Gate
137	ArpOn	Arpeggiator Power
138	Tune	Settings Tune
139	Hold	Keying Hold
140	BendUp	Settings Bend Up
141	BendDn	Settings Bend Dn
142	Priority	Keying Priority
143	VelGlide	Velocity Glide
144	PitchGlide	Pitch Glide
145	GlideMode	Glide Mode
146	Legato	Glide Legato
147	Retrigger	Glide Retrig

Amber

0	Perf1	Performance controller 1
1	Perf2	Performance controller 2
2	PerfX	(reserved for future use)
3	PerfY	(reserved for future use)
4	FltCut	Synth section filter Cutoff
5	FltRes	Synth section filter Resonance
6	FltMode	Synth section filter Mode
7	FltPwr	Synth section filter Power
8	FmtScale	Ens. formant filter Scale
9	FmtRes	Ens. formant filter Resonance
10	FmtFreq1	Ens. formant filter Freq 1
11	FmtFreq2	Ens. formant filter Freq 2
12	FmtFreq3	Ens. formant filter Freq 3
13	FmtFreq4	Ens. formant filter Freq 4
14	FmtGain1	Ens. formant filter Gain 1
15	FmtGain2	Ens. formant filter Gain 2
16	FmtGain3	Ens. formant filter Gain 3
17	FmtGain4	Ens. formant filter Gain 4
18	FmtNotch	Ens. formant filter Notch mode
19	FmtPwr	Ens. formant filter Power
20	ChrSpeed	Chorus Speed
21	ChrSpread	Chorus Spread
22	ChrMode	Chorus Mode
23	ChrBright	Chorus Bright Mode (Brt)
24	ChrPwr	Chorus Power
25	SynAmp	Synth Amp
26	EnsAmp	Ens. Amp
27	Pan	Pan
28	Level	Level
29	AnalogNoise	Analogue
30	Pitch	Master Pitch
31	Fine	Master Fine
32	Syn8’	Synth section 8’ level
33	Syn4’	Synth section 4’ level
34	SynNoise	Synth section Noise/Ext input level
35	SynInv	Synth section Inv. (Invert)
36	SynHp	Synth section paraphonic HP filter
37	SynLp	Synth section paraphonic LP filter
38	SynVel	Synth section velocity sensitivity
39	SynRoute	Syn Route (route Synth section to main/formant filter/chorus)
40	Ens8’	Ens. section 8’ level
41	Ens4’	Ens. section 4’ level
42	Ens2’	Ens. section 2’ level
43	EnsInv	Ens. section Inv. (Invert)
44	EnsHp	Ens. section paraphonic HP filter
45	EnsLp	Ens. section paraphonic LP filter
46	EnsVel	Ens. section velocity sensitivity
47	LfoMode	LFO Mode
48	LfoRate	LFO Rate
49	LfoSwing	LFO Swing
50	LfoPw	LFO Pulse Width
51	LfoPhase	LFO Phase
52	LfoSync	LFO Sync
53	LfoMono	LFO Mono
54	LfoGate	LFO Gate
55	RampDelay	Ramp Delay
56	RampRise	Ramp Rise
57	RampMultiplier	Ramp Mult
58	RampSync	Ramp Sync
59	RampGate	Ramp Gate
60	ModA	Mod Envelope Attack
61	ModD	Mod Envelope Decay
62	ModS	Mod Envelope Sustain
63	ModR	Mod Envelope Release
64	ModLoop	Mod Envelope Loop
65	ModLinear	Mod Envelope Linear
66	ModSync	Mod Envelope Sync
67	ModGate	Mod Envelope Gate
68	SynA	Synth Envelope Attack
69	SynD	Synth Envelope Decay
70	SynS	Synth Envelope Sustain
71	SynR	Synth Envelope Release
72	SynRange	Synth section Perform Range
73	SynMode	Synth section Perform Mode
74	EnsA	Ens. Envelope Attack
75	EnsR	Ens. Envelope Release
76	EnsSus	Ens. Envelope Sustain
77	EnsRange	Ens. section Perform Range
78	EnsMode	Ens. section Perform Mode
79	Tune	Settings Tune
80	Hold	Keying Hold
81	BendUp	Settings Bend Up
82	BendDn	Settings Bend Dn
83	Priority	Keying Priority
84	VelGlide	Vel Glide
85	PitchGlide	Pitch Glide
86	GlideMode	Glide Mode
87	Legato	Glide Legato
88	Retrigger	Glide Retrig

Fusor

Note that the VST plugin specification does not allow very long parameter names (a maximum of 7 characters). Therefore, in the VST plugin version of Fusor, parameter names are numbered – Par0, Par1 etc.

Descriptive parameter names are available only in AU and RTAS plugin formats.

0-7	Macro1-Macro8	Macro controls 1-8
8-15	Synth 1 P0-P7	Synth channel 1, params 0-7
16-23	Synth2 P0-P7	Synth channel 2, params 0-7
24-31	Synth3 P0-P7	Synth channel 3, params 0-7
32	LFO1 (reserved)	(reserved for future use)
33	LFO1 (reserved)	(reserved for future use)
34	LFO1 (reserved)	(reserved for future use)
35	LFO1 MODE	MonoMod LFO1 Mode
36	LFO1 FREQ	MonoMod LFO1 Rate
37	LFO1 SYNC	MonoMod LFO1 Sync
38	LFO1 SWING	MonoMod LFO1 Swing
39	LFO1 ASYM	MonoMod LFO1 PW
40	LFO1 PHASE	MonoMod LFO1 Phase
41	LFO1 (reserved)	(reserved for future use)
42	LFO2 (reserved)	(reserved for future use)
43	LFO2 (reserved)	(reserved for future use)
44	LFO2 (reserved)	(reserved for future use)
45	LFO2 MODE	MonoMod LFO2 Mode
46	LFO2 FREQ	MonoMod LFO2 Rate
47	LFO2 SYNC	MonoMod LFO2 Sync
48	LFO2 SWING	MonoMod LFO2 Swing
49	LFO2 ASYM	MonoMod LFO2 PW
50	LFO2 PHASE	MonoMod LFO1 Phase
51	LFO2 (reserved)	(reserved for future use)
52	LFO3 (reserved)	(reserved for future use)
53	LFO3 (reserved)	(reserved for future use)
54	LFO3 (reserved)	(reserved for future use)
55	LFO3 MODE	MonoMod LFO3 Mode
56	LFO3 FREQ	MonoMod LFO3 Rate
57	LFO3 SYNC	MonoMod LFO3 Sync
58	LFO3 SWING	MonoMod LFO3 Swing
59	LFO3 ASYM	MonoMod LFO3 PW
60	LFO3 PHASE	MonoMod LFO3 Phase
61	LFO3 (reserved)	(reserved for future use)
62	LFO4 (reserved)	(reserved for future use)
63	LFO4 (reserved)	(reserved for future use)
64	LFO4 (reserved)	(reserved for future use)
65	LFO4 MODE	MonoMod LFO4 Mode
66	LFO4 FREQ	MonoMod LFO4 Rate
67	LFO4 SYNC	MonoMod LFO4 Sync
68	LFO4 SWING	MonoMod LFO4 Swing
69	LFO4 ASYM	MonoMod LFO4 PW
70	LFO4 PHASE	MonoMod LFO4 Phase
71	LFO4 (reserved)	(reserved for future use)
72	EnvFol1 Attack	MonoMod Env. Follower1 Attack
73	EnvFol1 Release	MonoMod Env. Follower1 Release
74	EnvFol1 Gain	MonoMod Env. Follower1 Gain
75	EnvFol2 Attack	MonoMod Env. Follower2 Attack
76	EnvFol2 Release	MonoMod Env. Follower2 Release
77	EnvFol2 Gain	MonoMod Env. Follower2 Gain
78	EnvFol3 Attack	MonoMod Env. Follower3 Attack
79	EnvFol3 Release	MonoMod Env. Follower3 Release
80	EnvFol3 Gain	MonoMod Env. Follower3 Gain
81	EnvFol4 Attack	MonoMod Env. Follower4 Attack
82	EnvFol4 Release	MonoMod Env. Follower4 Release
83	EnvFol4 Gain	MonoMod Env. Follower4 Gain
84-107	Master FX 1 P1-P24	Master insert FX 1, params 1-24
108-131	Master FX 2 P1-P24	Master insert FX 2, params 1-24
132-155	Master FX 3 P1-P24	Master insert FX 3, params 1-24
156-179	Aux FX 1 P1-P24	Aux FX 1, params 1-24
180-203	Aux FX 2 P1-P24	Aux FX 2, params 1-24
204-227	Aux FX 3 P1-P24	Aux FX 3, params 1-24
228-255	(reserved)	(reserved for future use)
256-383	Synth 1 P0-P127	Synth channel 1, params 0-127
384-407	Ch1 FX1 P1-P24	Synth 1 FX slot 1, params 1-24
408-431	Ch1 FX2 P1-P24	Synth 1 FX slot 2, params 1-24
432-455	Ch1 FX3 P1-P24	Synth 1 FX slot 3, params 1-24
456-583	Synth 2 P0-P127	Synth channel 2, params 0-127
584-607	Ch2 FX1 P1-24	Synth 2 FX slot 1, params 1-24
608-631	Ch2 FX2 P1-P24	Synth 2 FX slot 2, params 1-24
632-655	Ch2 FX3 P1-P24	Synth 2 FX slot 3, params 1-24
656-783	Synth 3 P0-P127	Synth channel 3, params 0-127
784-807	Ch3 FX1 P1-P24	Synth 3 FX slot 1, params 1-24
808-831	Ch3 FX2 P1-P24	Synth 3 FX slot 2, params 1-24
832-855	Ch3 FX3 P1-P24	Synth 3 FX slot 3, params 1-24