Harmonizer®
Modules Manual

Part No: 141035a Manual Release1.3 12 June, 2000

This page intentionally left blank.

HARMONIZER® MODULES - CONTENTS

GROUPS

MODULES

HARMONIZER® MODULES - CONTENTS

HARMONIZER® MODULES - CONTENTS

HARMONIZER® MODULES - CONTENTS

HARMONIZER® MODULES - CONTENTS

This page intentionally left blank.

HARMONIZER® MODULES - GROUPS

This manual covers ***Orville*** as well as the **4000** and **7000** families of Eventide's Harmonizer Brand Effects Processors. Some modules are not available on all machines, especially earlier 4000s.

The reader should study the charts below to determine whether a given module is available on his system.

Some systems may contain modules that are not covered in this manual - these are usually present for system or debug purposes and should not be used.

Where the **7000** is referred to, this usually also includes **7500**. Similarly, **4000** refers to the **DSP4000** variants, as well as the **4500**.

BRIDGE MODULES

These modules in most cases convert an audio signal to a control signal or vice versa. When converting from audio to control, the audio signal is sampled to generate the control signal, which may result in values which do not transit smoothly. The filter variants of bridges may be used to smooth the values. These modules do not have a major impact on either control process or signal processing resources, although they make use of both.

			4000	*7000*	*7500*	*Orville*
a_fltr_c	Audio to control, filtered	**afc**	✓	✓	✓	✓
a_to_c	Audio to control	**a_c**	✓	✓	✓	✓
c_bridge	Machine to Machine	**cbr**				✓
c_fltr_a	Control to audio, filtered	**cfa**	✓	✓	✓	✓
c_to_a	Control to audio	**c_a**	✓	✓	✓	✓
c_tweak_a	Control to audio, fine	**c_a**	✓	✓	✓	✓

CONTROL MATH MODULES

These modules perform math on control signals. The processing takes place using the control processor and does not result in any signal processing resources being used. Except for those modules that allow for large numbers of inputs, these modules do not have a major impact on processing or memory resources.

			4000	*7000*	*7500*	*Orville*
c_adder	Control signal adder	**add**	✓	✓	✓	✓
c_and	Logical AND control signals	**and**	✓	✓	✓	✓
c_bound	Control signal bounder	**bnd**	✓	✓	✓	✓
c_cmp2	Advanced compare of control signals	**cmp**	✓	✓	✓	✓
c_comparator	Compare control signals	**cmp**	✓	✓	✓	✓
c_constant	Constant control signal	**con**	✓	✓	✓	✓
c_curve	Control signal map	**crv**	v2.3	✓	✓	✓
c_display	Graphical control input array editor	**dis**		✓	✓	✓
c_divide	Divide two control signals	**div**	✓	✓	✓	✓
c_ftop	Frequency to pitch converter	**ftp**	✓	✓	✓	✓
c_graph	Graphical control input array editor	**gph**	✓	✓	✓	✓
c_lintodb	Linear to dB conversion	**ldb**	v2.3	✓	✓	✓
c_many	Multiple output scaler	**mny**	v2.3	✓	✓	✓
c_lintodb	Linear to dB conversion	**ldb**	v2.3	✓	✓	✓
c_master	Master control math scaling module	**mst**	✓	✓	✓	✓
c_minmax	Min and max of several inputs	**max**	✓	✓	✓	✓
c_multiply	Multiply two control signals	**mul**	✓	✓	✓	✓
c_not	Logical NOT of control signal	**not**	✓	✓	✓	✓
c_or	Logical OR control signals	**or**	✓	✓	✓	✓
c_ptof	Pitch to frequency converter	**ptf**	✓	✓	✓	✓
c_quantize	Control signal quantizer	**qnt**	✓	✓	✓	✓
c_random	Random number generator	**rnd**	✓	✓	✓	✓
c_sincos	Control Sine and Cosine	**csc**		✓	✓	✓
c_sqrt	Control Signal Square Root	**csq**		✓	✓	✓
c_subtract	Subtract two control signals	**sub**	✓	✓	✓	✓
c_xor	Logical exclusive OR control signals	**xor**	✓	✓	✓	✓

HARMONIZER® MODULES

CONTROL PROCESS MODULES

These modules perform switching, selecting, and signal manipulation of control signals. The processing takes place using the control processor and does not result in any signal processing resources being used. Except where large numbers of inputs or entries (c_switch & c_table) are supported, these modules do not have a major impact on processing or memory resources in the control processor.

			4000	*7000*	*7500*	*Orville*
c_1shot	Control signal one shot	**1sh**	✓	✓	✓	✓
c_adsr	Control signal envelope generator	**cnv**	✓	✓	✓	✓
c_counter	Control signal counter	**cnt**	✓	✓	✓	✓
c_flop	Control signal flip flop	**flp**	✓	✓	✓	✓
c_impulse	Impulse train	**cim**	✓	✓	✓	✓
c_merge	Merge control signals	**mrg**	✓	✓	✓	✓
c_relay	Rear panel control access	**rly**	v2.3	✓	✓	✓
c_samp	Control signal sample and hold	**smp**	✓	✓	✓	✓
c_smooth	Control signal smoother	**smu**	✓	✓	✓	✓
c_switch	Select one of N control inputs	**swt**	✓	✓	✓	✓
c_table	Control signal table look-up	**tbl**	✓	✓	✓	✓
c_timer	Real time clock/timer	**tim**	v2.2	✓	✓	✓

DELAY MODULES

These modules create and use audio delay lines to produce delay and filter effects. Delay processing is done by the signal processor and uses DSP memory resources. The 4000 has a limited amount of delay memory equivalent to nearly 10.5 seconds of delay-line (monophonic, at 48000 samples/second). Orville and the 7000 family have about eight times as much.These modules use this delay memory and also a certain amount of signal processing and control processing resources. The *sampler* and *longdelay* modules are optional items on the 4000 series.

			4000	*7000*	*7500*	*Orville*
allpass	Allpass filter	**aps**	✓	✓	✓	✓
comb	Comb filter	**com**	✓	✓	✓	✓
delay	Audio delay	**dly**	✓	✓	✓	✓
dlysamp	Small delay-based sampler	**spl**			✓	✓
easytaps	Easy multitap delay line	**etp**	✓	✓	✓	✓
lattice	Lattice filter	**lat**	✓	✓	✓	✓
longdelay	Long Audio Delay	**dly**	option		✓	✓
microdelay	Modulatable micro-delay	**udl**	✓	✓	✓	✓
moddelay	Modulatable delay	**mdl**	✓	✓	✓	✓
multitap	Multi-tap delay line	**mtp**	✓	✓	✓	✓
picodelay	Fine grain delay	**pdl**	v2.300	✓	✓	✓
rampdelay	Rampable delay	**rdl**		✓	✓	✓
revdly	Reverse Audio Delay	**rdl**		✓	✓	✓
sampler	Audio Recorder	**smp**	option		✓	✓
stereotaps	Multi-tap with tap controls	**stp**	✓	✓	✓	✓

DETECTOR MODULES

These modules analyze an audio input and generate control signals to be used elsewhere in the patch. The modules are intensive in signal processing and DSP memory.

			4000	*7000*	*7500*	*Orville*
peak	Improved Peak Detector	**pkd**	v2.158	✓	✓	✓
peakdetect	Peak detector	**pkd**	✓	✓	✓	✓
pitchdetect	Pitch detector	**pdt**	✓	✓	✓	✓

HARMONIZER® MODULES - GROUPS

DYNAMIC MODULES

These modules control the dynamics of a signal, i.e. the range between minimum and maximum signal levels.

			4000	7000	7500	Orville
compressor	Soft knee compressor	**cpr**	v2.158	✓	✓	✓
ducker	Ducker	**duk**	✓	✓	✓	✓
gate	Audio noise gate	**gat**	✓	✓	✓	✓
gate2	Audio noise gate with sidechain i/p	**gat**	v2.158	✓	✓	✓

EXTERNAL MODULES

These modules allow external equipment to control the parameters of a preset.

			4000	7000	7500	Orville
extcontrol	External control	**ext**	✓	✓	✓	✓
exttrig	External trigger control	**ext**	✓	✓	✓	✓
midiclock	MIDI realtime control	**mck**	v2.300	✓	✓	✓
midicout	MIDI output	**out**	✓	✓	✓	✓
midinote	MIDI note interface	**mnt**	✓	✓	✓	✓
midinout	MIDI note output	**mno**	v2.200	✓	✓	✓
midinout	MIDI note output	**mno**	v2.200	✓	✓	✓

FILTER MODULES

			4000	7000	7500	Orville
de_emphasis	De-emphasis filter	**dee**	✓	✓	✓	✓
eq	Equalizer	**eq**	✓	✓	✓	✓
filter	Audio frequency filter	**flt**	✓	✓	✓	✓
fir	FIR filter/convolution	**fir**		✓	✓	✓
Harmonix	Harmonics Generator	**hrm**		✓	✓	✓
highcut	Highcut filter	**hct**	✓	✓	✓	✓
iir	Audio frequency filter	**iir**	✓	✓	✓	✓
lms	LMS adaptive filter	**lms**	✓	✓	✓	✓
mod_slew	Slew rate limit for mod signals	**msl**	✓	✓	✓	✓
modfilter	Modulatable filter	**mfr**	✓	✓	✓	✓
phaseshift	Phase shift	**pha**	✓	✓	✓	✓
pre_emphasis	Pre-emphasis	**pre**	✓	✓	✓	✓
slew	Slew rate limit	**slw**	✓	✓	✓	✓
tone	Audio tone control	**ton**	✓	✓	✓	✓

INTERFACE MODULES

These modules create the PARAMETER menu displays. These modules do not have any impact on signal processing resources. The control processor needs memory and time resources to process each of these modules.

			4000	7000	7500	Orville
gang	Gang of user interface objects	**gng**	✓	✓	✓	✓
head	Start of program	**hed**	✓	✓	✓	✓
headm	Start of program (multichannel)	**hed**		✓	✓	✓
hfader	Horizontal fader knob	**fdr**	✓	✓	✓	✓
hmenupage	Advanced menupage	**hmp**		✓	✓	✓
hmonitor	Horizontal control signal monitor	**hmn**	✓	✓	✓	✓
knob	Manual adjust of a control signal	**knb**	✓	✓	✓	✓
menupage	Menu page and soft keys	**mnu**	✓	✓	✓	✓
meter	Control signal meter	**mtr**	✓	✓	✓	✓
monitor	Control signal monitor	**mon**	✓	✓	✓	✓

HARMONIZER® MODULES

multiknob	Multiple value knob	**mkb**	v2.300	✓	✓	✓
percentknob	Percent knob	**pkb**	✓	✓	✓	✓
rfader	Round knob	**fdr**	✓	✓	✓	✓
sknob	Control signal adjuster with limits	**skb**	v2.112	✓	✓	✓
tapknob	Tapered knob	**knb**	✓	✓	✓	✓
textblock	Multi line text display	**txt**	✓	✓	✓	✓
textknob	Text knob	**tkb**	✓	✓	✓	✓
textline	Single line text display	**txt**	✓	✓	✓	✓
texttrigger	Control trigger with variable name	**ttg**	v2.300	✓	✓	✓
tmenupage	Menupage with variable name	**tmn**	v2.300	✓	✓	✓
tmonitor	Text monitor	**tmn**	✓	✓	✓	✓
trigger	Manual control signal trigger	**trg**	✓	✓	✓	✓
vfader	Vertical fader knob	**fdr**	✓	✓	✓	✓
vmonitor	Vertical monitor display	**vmn**	✓	✓	✓	✓

MATH MODULES

These modules perform mathematics on audio or mod signals.

			4000	***7000***	***7500***	***Orville***
abs	Absolute value	**abs**	✓	✓	✓	✓
add	Add two audio signals	**add**	✓	✓	✓	✓
adder	Add multiple audio signals	**add**	✓	✓	✓	✓
ampmod	Amplitude modulator	**mod**	✓	✓	✓	✓
bound	Signal bounder	**bnd**	✓	✓	✓	✓
comparator	Audio signal comparator	**cmp**	✓	✓	✓	✓
constant	Constant	**con**	✓	✓	✓	✓
cosine	Cosine Function	**cos**	✓	✓	✓	✓
curve	Mapping or waveshaping function	**crv**	✓	✓	✓	✓
differentiator	Differential function	**du**	✓	✓	✓	✓
dither	Dithering/Requantization	**dit**	v2.300	✓	✓	✓
exp	Exponentiator function	**exp**	✓	✓	✓	✓
exp_mod	Exponentiator for mod signals	**exp**	✓	✓	✓	✓
gain	Audio gain adjust	**gan**	✓	✓	✓	✓
integrator	Integration function	**int**	✓	✓	✓	✓
log	Logarithm	**log**	✓	✓	✓	✓
log_mod	Logarithm, for mod signals	**log**	✓	✓	✓	✓
m_curve	Mapping for mod signals	**crv**	✓	✓	✓	✓
m_ucurve	Unipolar mapping for mod signals	**crv**	✓	✓	✓	✓
multiply	Multiply two audio signals	**mul**		✓	✓	✓
noiseshape	First order noise shaper	**nsh**	✓	✓	✓	✓
plex	Reverberation tool	**plx**	✓	✓	✓	✓
quadrature	Quadrature transformer	**qad**	✓	✓	✓	✓
quantize	Audio bit quantizer	**qnt**	✓	✓	✓	✓
scale	Audio signal scaler (attenuator)	**scl**	✓	✓	✓	✓
sinus	Sine Function	**sin**		✓	✓	✓
sqrt	Square root function	**sqt**		✓	✓	✓
subtract	Subtract two audio signals	**sub**	✓	✓	✓	✓

HARMONIZER® MODULES - GROUPS

MISCELLANEOUS MODULES

When inserted into a patch in the Patch editor, those modules in this set that have control inputs will automatically be added to the head module. For each module inserted, a new softkey will be created and an associated menu page will appear.

			4000	*7000*	*7500*	*Orville*
flipflop	Audio octave divider	**ffp**	✓	✓	✓	✓
oneshot	One shot generator	**sht**	✓	✓	✓	✓
samphold	Audio sample and hold	**smp**	✓	✓	✓	✓
scales	Ultrashifter component	**scl**	v2.3	✓	✓	✓
scope	Single Trace Oscilloscope	**scp**		✓	✓	✓
sequencer	Mod signal sequencer	**seq**	✓	✓	✓	✓
Sourceanalyzer	Ultrashifter component	**src**	v2.3	✓	✓	✓
spectrum	Spectrum Analyzer	**spc**		✓	✓	✓

MIXER MODULES

When inserted into a patch in the Patch editor, those modules in this set that have control inputs will automatically be added to the head module. For each module inserted, a new softkey will be created and an associated menu page will appear.

			4000	*7000*	*7500*	*Orville*
crossin	Input crossfader	**mix**	✓	✓	✓	✓
crossout	Output crossfader	**crs**	✓	✓	✓	✓
crossoutq	Quad Cross-out	**xoq**				✓
iswitch	Click-less input audio signal switch	**isw**	v2.300	✓	✓	✓
mix	Two-input audio mixer	**mix**	✓	✓	✓	✓
mixer	Multi-Input Audio Mixer	**mix**	✓	✓	✓	✓
oswitch	Click-less output audio signal switch	**isw**	v2.300	✓	✓	✓
quadmixer	Quadrophonic mixing and panning	**qmx**				✓
stereomixer	Multi-input stereo audio mixer	**smx**	✓	✓	✓	✓
switch	Audio signal switch	**swi**	✓	✓	✓	✓

NODES

These are pseudo modules, found in Vsigfile only, that act as 'binding posts' They are typically used where multiple inputs need to be fed from a common source. The process of downloading from Vsigfile causes them to be removed, thus they will not be visible in the patch editor or any sigfiles exported from the system,

			4000	*7000*	*7500*	*Orville*
a_to_a	Audio Node	see text				
c_to_c	Control Node					

OSCILLATOR MODULES

When inserted into a patch in the Patch editor, those modules in this set that have control inputs will automatically be added to the head module. For each module inserted, a new softkey will be created and an associated menu page will appear.

			4000	*7000*	*7500*	*Orville*
envelope	Envelope generator	**env**	✓	✓	✓	✓
impulse	Impulse generator	**imp**	✓	✓	✓	✓
lfo	Low-frequency oscillator	**lfo**	✓	✓	✓	✓
lfo2	Retriggerable LFO	**lf2**		✓	✓	✓
noise	Noise Generator	**noi**	✓	✓	✓	✓
oscillator	Audio oscillator	**osc**	✓	✓	✓	✓
waveform	Programable audio waveform	**wfm**	✓	✓	✓	✓

HARMONIZER® MODULES

PITCHSHIFT MODULES

When inserted into a patch in the Patch editor, those modules in this set that have control inputs will automatically be added to the head module. For each module inserted, a new softkey will be created and an associated menu page will appear. These modules use a considerable amount of delay memory and signal processing resources

			4000	*7000*	*7500*	*Orville*
detune	Audio signal detuner	**tun**	✓	✓	✓	✓
diatonicshift	Diatonic pitch shifter	**dsh**	✓	✓	✓	✓
freqshift	Audio frequency shifter	**fsh**	✓	✓	✓	✓
multishift	Multi-output pitch shifter	**msh**	✓	✓	✓	✓
pitchshift	Pitch shifter	**psh**	✓	✓	✓	✓
reverse	Reverse shift	**rev**	✓	✓	✓	✓
stereoshift	Stereo pitch shifter	**ssh**	✓	✓	✓	✓
ultrashifter	Formant-corrected pitch shifter	**ush**	v2.300	✓	✓	✓

REVERB MODULES

When inserted into a patch in the Patch editor, those modules in this set that have control inputs will automatically be added to the head module. For each module inserted, a new softkey will be created and an associated menu page will appear. These modules use a considerable amount of delay memory and signal processing resources

			4000	*7000*	*7500*	*Orville*
diffchorus	Diffusor with chorus	**dfc**	✓	✓	✓	✓
diffusor	Diffusor	**dfr**	✓	✓	✓	✓
reverb_a	Reverberator (12 delays)	**rva**	✓	✓	✓	✓
reverb_b	Reverberator (8 delays)	**rvb**	✓	✓	✓	✓
reverb_c	Reverberator (6 delays)	**rvc**	✓	✓	✓	✓
reverb_d	Reverberator (4-32 delays)	**rvd**		✓	✓	✓

HARMONIZER® MODULES

A_FLTR_C — GROUP: BRIDGE

Audio to Control, Filtered **afc**

This module converts an audio signal into a control signal. Before doing the conversion, the audio signal is lowpass filtered. This reduces errors caused by the control signal being updated at a much slower rate than the audio signal. The actual update rate depends on program complexity, MIDI operation, and front panel usage.

This module is useful in allowing an audio input signal or internal oscillator module to control signal processing parameters that are only accessible via control signals.

Specifiers:

time_constant

This is the time constant for the filter that is used on the audio signal before it is converted to a control signal. It controls the amount of smoothing. Range: 0 to 100 seconds.

Audio inputs:

in

audio or mod signal

Control outputs:

out

0 if no audio, swings from -1 to 1 if full audio

Order:

A_FLTR_C modulename time_constant in

A_TO_A — GROUP: NODE

Audio Node **a_to_a**

This is a pseudo-module, found in Vsigfile only. Its purpose is to act as a 'binding post' in supermodules, allowing a single audio input to the module to drive multiple inputs within the module. It may also have use in tidying up connections within a graphic display.

Note that this module will disappear when saved as a .sig file, or when downloaded to the system, and the signal on its input will then be directly connected to those inputs connected to its output.

	signal	min	max	description
Control inputs				
	in			input value
Control outputs				
	out			output value (same as input).

Order

A_TO_A modulename in

Resource Usage

none

A_TO_C — GROUP: BRIDGE

Audio to Control **a_c**

Audio to control converter. Takes an audio signal and converts it to a control signal. This process is done by sampling the audio signal at intervals and outputting a control signal proportional to the level of the audio signal at the time of the sample. If the audio signal is changing too fast, the output will become unstable. Use a_fltr_c instead. This module has the feature of being very efficient with resources.

This module is useful in allowing an audio input signal or internal oscillator module to control signal processing parameters that are only accessible via control signals.

Audio inputs:

in

audio or mod signal

Control outputs:

out

0 if no audio, swings from -1 to 1 if full audio

Order:

A_TO_C modulename in

HARMONIZER® MODULES

ABS — GROUP: MATH

Absolute Value — abs

This module takes the arithmetic absolute value of an audio input signal. This is equivalent to full-wave rectification. This can be used as a crude frequency doubler, as distortion, or in level detection applications.

Audio inputs:

in

audio input

Audio outputs:

out

Absolute value of audio input.

Order:

ABS modulename in

ADD — GROUP: MATH

Add Two Audio Signals — add

This module adds the two audio signals 'in1' and 'in2'. It is the simplest way of mixing two signals together. It is often used for creating feedback loops around delay lines.

Audio inputs:

in1 in2

audio inputs

Audio outputs:

out

audio output in1 + in2

Order:

ADD modulename in1 in2

ADDER — GROUP: MATH

Add Multiple Audio Signals — add

Mix (without gain control) a specified number of input audio signals. This module adds two or more audio signals together. The number of signals to be added is specified by the *ninputs* specifier.

Specifiers:

ninputs

number of inputs. Range: 2 to 50.

Audio inputs:

in1 in2 ... inN

audio inputs. There will be multiple audio inputs as specified by *ninputs*.

Audio outputs:

out

sum of audio inputs.

Order:

ADDER modulename ninputs in1 in2 ... inN

ALLPASS — GROUP: DELAY

Allpass Filter — aps

This module implements an allpass filter of the type described in Manfred Schroeder's seminal paper on digital reverb simulation. As such, this module is intended to be used as a building block in reverb and room simulations.

In effect, this module is less like a filter and more like a repeating delay line. It is called an allpass filter because it has the unique characteristic of having a FLAT frequency response. This enables a user to cascade several allpass filters in series without generating excess coloration of the sound. This technique is typically used in reverberators to generate diffusion, a dense grouping of echoes.

Specifiers:

maxdelay

maximum delay. Range: 1 to 660 milliseconds

Audio inputs:

in

audio input

Audio outputs:

out

audio output

Control inputs:

delayamt

This controls the amount of delay in the feedback loop. Range: 0 to maxdelay milliseconds.

g

Controls the feedback gain. 0 is no feedback and 1 is 100 per cent feedback. Negative numbers invert the phase of the feedback. Range: -1.0 to 1.0.

Userobjects:

obj

Menupage of controls. (collection)

Order:

ALLPASS modulename maxdelay in delayamt g

AMPMOD GROUP: MATH

Amplitude Modulator mod

The ampmod module uses one audio input (mod) to control the amplitude of another (in). This is equivalent to one signal being multiplied by the other.

The ampmod module is useful for creating tremolo effects, autopanning, envelope control, and many other applications.

Both inputs of this module are full bandwidth audio. You can use this module to multiply two signals together.

Audio inputs:

in

base audio

mod

amplitude control.

Audio outputs:

out

The scaled output. It has the value: out = in * (offset + mod * modamt)

Control inputs:

modamt

This control signal scales the mod input before it is multiplied with the input signal. In combination with the offset control, this can be used to control the depth of amplitude modulation. Range: -1.0 to 1.0.

offset

The offset control signal determines the amplitude of the output signal in the absence of any modulation signal. Its value is added to the mod input scaled by the modamt. Range:-1.0 to 1.0.

Userobjects:

obj

Menupage of controls. (collection)

Order:

AMPMOD modulename in mod modamt offset

BOUND GROUP: MATH

Signal bounder bnd

This module will make a audio signal stay within the bounds of a maximum and a minimum. If the minimum setting is more than the maximum, the output will be at the minimum setting.

The obvious use of this module is as a hard limiter. It's also good for simulating transistor distortion. If the minimum is zero and the maximum is one, you have a rectifier.

Audio inputs:

in

The input to be bounded.

Audio outputs:

out

The bounded signal.

Control inputs:

minimum

The lower bound of the signal. Range: -1.0 to 1.0.

maximum

The upper bound of the signal. Range: -1.0 to 1.0.

Order:

CONSTANT modulename in minimum maximum

C_1SHOT — GROUP: CONTROL PROCESS

Control Signal One shot — 1sh

This module will produce a control signal trigger whenever the input goes from less than 1 to greater or equal to 1. A control signal trigger is normally zero. When the trigger occurs, the output goes to 1 and then back to zero.

Control inputs:

in

The input control which we are converting to a one-shot control signal.

Control outputs:

out

The one shot result.

Userobjects:

obj

The control input if it is not connected to a control signal.

Order:

C_1SHOT modulename in

C_ADDER — GROUP: CONTROL MATH

Control Signal Adder — add

This module adds together a specified number of control signals. This is needed in creating patches where more than one source can affect a single parameter.

Specifiers:

ninputs

Specifies how many inputs are to be added together. Range: 2 to 32 inputs.

Control inputs:

in1 in2 ... inN

The input control signals that are to be added together. Range -32768 to 32767.

Control outputs:

out

The sum of all of the input control signals If the resultant value exceeds -32768 or +32767, it will be limited at those values.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

C_ADDER modulename ninputs in_1 in_2 ... in_N

C_ADSR — GROUP: CONTROL PROCESS

Control Signal Envelope Generator — cnv

This module implements an ADSR-type envelope generator for control signals. It has 3 states:
Attack: Rising until it reaches max level (1).
Decay/sustain: falling until it reaches the sustain level (and waiting for the gate_off).
Release: falling back to the min level (0) until another gate_on.

Control inputs:

Attack

This input controls the attack rate. Range 0 to 1.

Decay

This input controls the decay rate. Range 0 to 1.

Sustain

This input controls the sustain level. Range 0 to 1.

Release

This input controls the attack rate. Range 0 to 1.

Gate

A value less than 1 is considered to be "0". Values greater or equal to 1 are considered to be "1". A rising edge triggers attack mode, while a falling edge triggers release mode.

Control outputs:

out

The current level.

state
The current state of the ADSR:
0 - release
1 - attack
2 - decay/sustain

Userobjects:
obj
Menupage of control inputs not connected to control signals. (collection)

Order:
C_ADDER modulename attack decay sustain release gate

C_AND GROUP: CONTROL MATH

Logical AND Control Signals **and**

This module execute a logical AND of two control signals. An input signal of greater than or equal to 1 (values 1 through 32767) is considered to be a logical true. An input signal of less than 1 (values -32768 through .99999) is considered to be a logical false. If both inputs have a value of 1.0 or greater, the output is set to 1.0, otherwise it is set to 0.0.

Control inputs:
in1 in2
Input signal to be ANDed range: -32768 to 32767

Control outputs:
out
The logical AND of the input control signals

Userobjects:
obj
Menupage of control inputs not connected to control signals. (collection)

Order:
C_AND modulename in1 in2

C_BRIDGE GROUP: BRIDGE

Inter-machine control link **cbr** **(O*rville* only)**

This module allows control signals to be passed between the A and the B machines on Orville. If it is included in a preset on both the A and B machine, a control signal connected to an input on one *c_bridge* module will appear at the corresponding output on the other. Note that for one-way communication, the HEADM's global1-4 in one machine will receive from a *c_bridge* in the other.

	signal	min	max	description
Control inputs				
	in1..4	-32768	32767	value to be sent to other machine
Control outputs				
	out1..4	-32768	32767	value received from other machine

Order
C_BRIDGE, modulename, in1, in2, in3, in4

Resource Usage
low

C_BOUND GROUP: CONTROL MATH

Control Signal bounder **bnd**

This module will make a control signal stay within the bounds of a maximum and a minimum. If the minimum setting is more than the maximum, the output will be at the minimum setting.

All modules already bound their control signal inputs. But when you have complex manipulation and bridges into the audio domain, things can get hairy. Use this module to tame things.

Control inputs:
in
The input to be bounded.
minimum
The lower bound of the signal. Range: -32000.0 to 32000.0.
maximum
The upper bound of the signal. Range: -32000.0 to 32000.0.

Control Outputs:
out
The bounded signal.

Order:

CONSTANT modulename in minimum maximum

C_CMP2 GROUP: CONTROL MATH

Advanced Compare of Control Signals cmp

This module compares the value of two input control signals. There are six outputs for different conditions for the compare.

Control inputs:

in1 in2

The input control signals to be compared Range:-32768.0 to 32767.0.

Control outputs:

eq

1 if (in1 = in2), else 0

ne

1 if (in1 <> in2), else 0

gt

1 if (in1 > in2), else 0

lt

1 if (in1 < in2), else 0

ge

1 if (in1 >= in2), else 0

le

1 if (in1 <= in2), else 0

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

C_COMPARATOR modulename in1 in2

C_COMPARATOR GROUP: CONTROL MATH

Compare Control Signals cmp

This module compares the value of two input control signals. If the first is greater than that of the second, a logical true (value 1.0) is output. Otherwise, a value of 0.0 (FALSE) is output.

Control inputs:

in1 in2

The input control signals to be compared Range:-32768 to 32767

Control outputs:

out

1 if in1 > in2, else 0

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

C_COMPARATOR modulename in1 in2

C_CONSTANT GROUP: CONTROL MATH

Constant Control Signal con

This module is used to create a control signal of a fixed value which is often necessary to generate a bias value in various control schemes. This module is now essentially obsolete, as control values can be directly entered into unconnected control inputs in either Vsigfile or the patch editor.

Specifiers:

value

The value to be output. Range: -32768.0 to 32767.0

Control outputs:

out

The output value.

Order:

C_CONSTANT modulename value

C_COUNTER GROUP: CONTROL PROCESS

Control Signal Counter cnt

This module implements a control signal counter. It will count up to a specified value and then stop. The counting mechanism is controlled by a "clock" control signal input. Whenever the clock value goes from 0 to 1, the counter will increment by a preset amount.
The counter has an output corresponding to its current counting value, and an output that indicates if the maximum count has been reached.
This module is used in complex control schemes that require delayed reactions to user inputs. For example, a patch can be created that causes one sweep to be triggered a second after you press a button on the front panel. Note that the count parameters may be changed at any time.

Control inputs:

clock
This input controls the counting of the clock mechanism. Each time this input transitions from below 0.5 to above 0.5, the increment value is added to the current count. Range: -32768.0 to 32767.0.

reset
When this input is greater than 0.0 and a clock occurs, the counter is reset to 0. If you want to reset the counter asynchronously (i.e. without a clock), use a C_OR as in the sigfile below. Range: -32768.0 to 32767.0.

incr
This controls how much is added to the count value for each transition of the clock input. Range: -32768.0 to 32767.0.

maxcount
This determines the maximum allowed value of the counter. The counter will stop once it has reached this value. Range: -32768.0 to 32767.0.

Control outputs:

out
The current value of the counter.

timeout
Set to 1.0 if the maxcount has been reached, otherwise set to zero.

Userobjects:

obj
Menupage of control inputs not connected to control signals. (collection)

Order:

C_COUNTER modulename clock reset incr maxcount

Example sigfile:

```
HEADM "adc" 2 2 adc-null adc-null "Empty" "Empty" 3 count-obj reset-obj monitor-obj
TRIGGER "reset" "reset" "reset"
TRIGGER "count" "count" "count"
C_OR "c_or" count-out reset-out
C_COUNTER "c_counter" c_or-out reset-out 1 32000
MONITOR "monitor" c_counter-out "count: %4.0f" "Operate"
TAIL "tail"
```

C_CURVE GROUP: CONTROL MATH

Control Signal Map crv

An arbitrary relationship between an input value and an output value. See the CURVE module for a good explanation of curves. This map works on control signals that are between -1 and 1. Use this module for special tapers on knobs or external controls. This module can be edited under Vsigfile using the Waveform editor.

Specifiers:

npoints
Specifies how many data points there are. Range: 1 to 32.

point1 point2 ... pointN
The points describing the output values along the curve. Range: -1.0 to 1.0.

Control inputs:

in
Value to be mapped. This should be between -1.0 and 1.0. This is not accessible as a userobject.

Control Outputs:

out
The output. Also between -1 and 1.

Userobject:

obj
The curve. (collection)

Order:

C_CURVE modulename in npoints point1 point2 ... pointN

HARMONIZER® MODULES

C_DISPLAY GROUP: CONTROL MATH

This module allows you to edit an arbitrary number of points (up to 32) on a graphical display control. It also provides an master offset control input to add or subtract a value from all the values. The screen width of the control is also variable. Displayed label (x-values) can be specified on control inputs for each point.
The module has the same functionality as C_GRAPH with more display options.

	signal	min	max	description
Specifiers				
	screen_width	2	4	Screen width of control in quarters. (2=half screen, 4=full)
	display		0	2
	8 char name	{string}		Control name; appears in first line of graph display
	arrow text	{string}		Text to be displayed between point x and y. The ^ character will cause a down-arrow to be displayed in its place.
	format labels	{string}		Formatting strings for the x and y values displayed. The standard %.0f style formatting applies.
	format points	{string}		
	number points	1	32	Number of control inputs (and outputs) = points on the graph.
	point min	-32768	32767	Minimimum values for each point.
	point max	-32768	32767	Maximum values for each point.
	point res	0	1	Point resolution. (in .001 increments)
	point 1..N	point_min	point_max	The default y-values for the points
Control inputs				
	offset	point_min	point_max	Value added to points before being displayed & outputted. The internal value of the point does not change.
	label1..N	-32768	32767	Label (x-value) to be displayed by the format labels string when that point is being edited.
Control outputs				
	out1..N	point_min	point_max	internal y-value N + offset

Userobjects
obj — Actual displayed graph control

Order
C_DISPLAY modulename screen_width display 8_char_name arrow_text format_labels format_points number_points point_min point_max point_res point1…pointN offset label1…labelN

Resource Usage
low

C_DIVIDE GROUP: CONTROL MATH

Divide Two Control Signals **div**

This module divides one control signal by another. If the dividend (in1) is 0, then out is zero, regardless of what the divisor (in2) is set to. Otherwise, if the divisor is zero, then out is set to positive infinity (+32767) if the dividend is positive, or minus infinity (-32768) if the dividend is negative. This is useful for creating various user interactions with the DSP parameters.

Control inputs:

in1
dividend Range: -32768.0 to 32767.0.

in2
divisor Range: -32768.0 to 32767.0.

Control outputs:

out
The result of in1/in2. If the resultant value exceeds -32768 or +32767, it will be limited at those values.

Userobjects:

obj
Menupage of control inputs not connected to control signals. (collection)

Order:
C_DIVIDE modulename in1 in2

C_FLOP GROUP: CONTROL PROCESS

Control signal flip flop **flp**

A flip-flop is a basic building block for on-off control systems.
There are three inputs: set, reset, toggle.
When SET>=1, the output will go to 1.
When RESET>=1 (provided SET<1), the output will go to 0.
When TOGGLE>=1 (provided SET and RESET are both <1), the output will toggle from 0 to 1, or from 1 to 0.

Control inputs:

reset

The input control that sets the output to 1.

set

The input control that sets the output to 0.

toggle

The input control that changes the output.

Control Outputs:

out

The current state of the flip flop.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

C_FLOP modulename reset set toggle

C_FLTR_A GROUP: BRIDGE

Control to Audio, Filtered cfa

This module converts a control signal into an audio signal. After doing the conversion, the audio signal is lowpass filtered. This reduces high frequency components caused by the digital conversion process.

This module is useful in taking a user input, like a button press or pedal input, and controlling the modulation of a particular module. The lowpass filter built into the module acts to smooth out the roughness associated with control signals.

Specifiers:

time_constant

The time constant of the filter that is used on the control signal after it is converted to an audio signal, cotrolling the degree of smoothing. Range: 0 to 100 seconds.

Audio outputs:

out

The resultant audio output

Control inputs:

in

The control signal to be converted Range: -1.0 to 1.0

Userobjects:

obj

The control input if it is not connected to a control signal.

Order:

C_FLTR_A modulename time_constant in

C_FTOP GROUP: CONTROL MATH

Frequency to pitch converter ftp

This module converts its input signal from frequency (Hz) to pitch (cents). An input of 440.0 (Hz) produces an output of 5700.0 (cents), because A-440 is 7 semitones (700 cents) above C5 in the octave below.

Control inputs

in	16	32767	Numeric frequency input to be converted.

Control outputs

out			The corresponding pitch cent value of the numeric frequency input.

Order

C_FTOP, modulename, in

C_GRAPH GROUP: CONTROL MATH

Graphical control input array editor gph v2.2

This module allows you to edit an arbitrary number of points (up to 32) on a display graph control. It also provides an offset control input to add/subtract a value from all numbers before results are output. The screen width of the control is also variable. Displayed label (x-values) can be specified on control inputs created for each point.

signal	min	max	description

Specifiers

screen width	2	4	Screen width of control in quarters. (2=half screen, 4=full)
8 char name	{string}		Control name; appears in first line of graph display

arrow text	{string}		Text to be displayed between point x and y. The '^' character will cause a down-arrow to be displayed in its place.
format labels	{string}		These are formatting strings for the x and y values displayed. The standard
format points	{string}		"%.0f" style formatting applies.
number points	1	32	Number of control inputs (and outputs) = points on the graph.
point min	-32768	32767	Min & max values for editing & outputting points.
point max	-32768	32767	
point res	0	1	Point resolution. (in .001 increments)
point 1..N	point min	point max	The default y-values for the points.

Control Inputs

offset			Value added to points before being displayed & outputted. The internal value of the point does not change.
label1..N	-32768	32768	Label (x-value) to be displayed by the format labels string when that point is being edited.

Control Outputs

out1..N			(internal y-value N + offset) bound by point min & max.

User Object

obj			Actual displayed graph control

Order

C_GRAPH, modulename, screen width, 8 char name, arrow text, format labels, format points, number points, point min, point max, point res, point1…point N, offset, label 1…label N

C_LIN2DB — GROUP: CONTROL MATH

Linear to dB conversion — ldb — v2.3

This module converts a linear valued control signal to its corresponding dB value. It is intended to replace the resource-intensive log module for low bandwidth applications, such as on-screen display. An input of 1.0 gives 0 (dB) out.

signal	min	max	description
Control inputs			
linvalue	0.0001	32768	Linear input value.
Control outputs			
dbvalue	-90	90	dB output value.

Order

C_LIN2DB, modulename, linvalue

Demonstration Sigfile:

```
HEAD "adc" adc-null adc-null "C_LIN2DB test" "Empty" 1 menupage-obj
KNOB "knob" "in: %4.2f" "in" 0.0001 32767 10 1
C_LIN2DB "c_lin2db" knob-out
MONITOR "monitor" c_lin2db-dbvalue "out %4.2f" "out"
MENUPAGE "menupage" "display" "display" 2 knob-obj monitor-obj
TAIL "njr"
```

C_IMPULSE — GROUP: CONTROL PROCESS

An Impulse Train — cim

Like *impulse*, *c_impulse* creates a pulse train with a variable frequency. Each impulse is a control signal trigger where the output is high for one update cycle.

Control inputs:

freq

How many pulses per second. Range 0 to 20000.

Control Outputs:

out

The trigger output.

Userobjects:

obj

The control input if it is not connected to a control signal.

Order:

C_IMPULSE modulename freq

C_MANY GROUP: CONTROL MATH

This module takes one control input and produced a number of outputs, each being a scaled representation of the input. The relationship between the input and each output is:

outputn = inputn * multn + offsetn

This module may be used in place of *c_master*. Either of these modules is useful when a single knob is used to control a number of differing parameters.

	signal	min	max	description
Specifiers:				
	noutputs	1	32	the number of control outputs
	inmin	-32767	+32767	the minimum value for the input
	inmax	-32767	+32767	the maximum value for the input
Control Inputs:				
	in	-32767	+32767	master control input
	mult 1..n	-32767	+32767	multiplier for output 1..n
	offset 1..n	-32767	+32767	offset for output 1..n
Control Outputs:				
	output 1..n	-32767	+32767	slave control outputs
Userobjects				
	obj			A userobject to display the in and out values, suitable for placing on a menupage.

Order

C_MANY modulename noutputs inmin inmax in mult1..multn offset1..offsetn

Resource Usage

low, unless very many outputs.

C_MASTER GROUP: CONTROL MATH

Master control math scaling module mst

This module allows a single knob to generate a number of linked outputs, each one of which has a different relationship to the input. The aim of this module is to allow a single knob or input to control many different parameters in a controllable way.

The variable number of outputs are scaled numbers, between outStartN and outStopN (inclusive) based on the input's position between inStart and inStop (inclusive). OffsetN is added to the result, which is then forced between the boundaries of outStartN and outStopN for outputN. Thus as the input traverses its full range (between inStart and inStop) each output will traverse its full range (between outStartN and outStopN) with an optional OffsetN.

As an example, if the input is at instart, the value of output3 will be outstart3 + offset3. If this value is lower than outstart3 it will be set to outstart3, similarily if is is higher than outstop3 it will be set to outstop3.

	signal	min	max	description
Specifier				
	•noutputs	1	32	Specifies how many units are to be created.
	•instart	-32768	32767	start of input value range.
	•instop	-32768	32767	end of input value range.
Control inputs				
	•input	-32768	32767	master input to be scaled and fed to outputs.
	•outstart1..N	-32768	32767	start of output value range.
	•outstop1..N	-32768	32767	end of output value range.
	•offset1..N	-32768	32767	Offset values to be added after scaling but before bounds checking.
Control outputs				
	•output1..N	-32768	32767	scaled outputs.

Order

C_MASTER, modulename, number outputs, instart, instop, in, outstart1..outstartN, outstop1..outstopN, offset1..offset1N.

The following is a sigfile which demonstrates the functioning of this module:

```
HEAD "adc" adc-null adc-null "C_MASTER demo" " " 1 menupage-obj
KNOB "input" "in: %3.0f" "in" -100 100 1 0
KNOB "outstart" "start: %3.0f" "instart" -100 100 1 0
KNOB "outstop" "stop:  %3.0f" "outstop" -100 100 1 50
KNOB "offset" "off: %3.0f" "offset" -100 100 1 0
C_MASTER "c_master" 1 0 50 input-out outstart-out outstop-out offset-out
MONITOR "output" c_master-output1 "out: %3.1f" "out"
MENUPAGE "menupage" "" "" 5 input-obj outstart-obj outstop-obj offset-obj output-obj
TAIL "njr"
```

C_MERGE GROUP: CONTROL PROCESS

Merge Control Signals **mrg**

This module merges together a specified number of control signals. The output is set to the value of the last input that has changed, or, in the case of a tie, the last input on the list.

Specifier:

ninputs

Specifies how many inputs are to be merged together. Range: 2 to 32.

Control inputs:

in1 in2 ... inN

The input control signals to be merged.

Control Outputs:

out

The new value of the last input to have changed.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

C_MERGE modulename ninputs in1 in2 ... inN

C_MINMAX GROUP: CONTROL MATH

Minimum and Maximum of a Group of Control Signals

max

This module has many inputs and two outputs. The max output is the highest input. The min output is the lowest input..

Specifiers:

ninputs

How many control inputs. Range: 2 to 32.

Control inputs:

in1, in2, ... inN

the control inputs

Control Outputs:

max

The highest value of all the inputs

min

The lowest value of all the inputs

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

C_MINMAX modulename ninputs in1 in2 ... inN

C_MULTIPLY GROUP: CONTROL MATH

Multiply Two Control Signals **mul**

This module multiplies two control signals. If the resulting value is greater than +/- 32767, the output value is limited.

Control inputs:

in1 in2

Input signals that are going to be multiplied together. Range: -32768.0 to 32767.0.

Control outputs:

out

The result of multiplying the two signals. If the resultant value exceeds -32768 or +32767, it will be limited at those values.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

C_MULTIPLY modulename in1 in2

C_NOT GROUP: CONTROL MATH

Logical NOT of Control Signal **not**

This module execute a logical NOT of the input control signal. An input signal of greater than or equal to 1 (values 1 through 32767) is considered to be a logical true. An input signal of less than 1 (values -32768 through .99999) is considered to be a logical false. An input value 1.0 or greater yields an output value of zero, otherwise the output is 1.0.

Control inputs:

in

The input control signal to be logically inverted. Range: -32768.0 to 32767.0.

Control outputs:

out

The logical NOT of the input control signal

Userobjects:

obj

The control input if it is not connected to a control signal.

Order:

C_NOT modulename in

C_OR GROUP: CONTROL MATH

Logical OR Control Signals **or**

This module executes a logical OR of two control signals. An input signal of greater than or equal to 1 (values 1 through 32767) is considered to be a logical true. An input signal of less than 1 (values -32768 through .99999) is considered to be a logical false. If either input has a value of 1.0 or greater, the output is set to 1.0, otherwise it is set to 0.0.

Control inputs:

in1 in2

The input control signals that are to be logically ORed together Range: -32768.0 to 32767.0.

Control outputs:

out

The logical OR of the input control signals

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

C_OR modulename in1 in2

C_PTOF GROUP: CONTROL MATH

Pitch to frequency converter **ptf**

This module converts its input signal from pitch (cents) to frequency (Hz). An input of 5700.0 (cents) produces an output of 440.0 (Hz), because A-440 is 7 semitones (700 cents) above C5 in the octave below. This module is the converse of c_ftop.

	signal	min	max	description
Control inputs				
	in	0	13162	Incoming numeric pitch.
Control outputs				
	out			The corresponding frequency Hz value of the incoming numeric pitch

Order

C_PTOF, modulename, in

C_QUANTIZE GROUP: CONTROL MATH

Control Signal quantizer **qnt**

This module will force a control signal onto discrete steps, where you control the step size. You also must provide a value to give the starting point for the steps.

One use of this module is when, with a C_SWITCH, you don't want to slide between switch settings. Use this module with *stepsize* of 1.0 and origin of 0.0 on the input of a C_SWITCH to get a discrete switch.

Control inputs:

in

The input to be quantized.

stepsize
How big of a step. Range: .0001 to 32000.0.
origin
One of the desired output values. This helps find out where all the other output values are. Range: -32000.0 to 32000.0.

Control Outputs:
out
The quantized control signal.

Order:
CONSTANT modulename in stepsize origin

C_RANDOM — GROUP: CONTROL MATH

Random number generator — rnd

This module produces a specified number of (pseudo) random numbers. The numbers on each output are unique and range from 0 to the maximum number specified. A reset input is provided to allow the sequence to be restarted from the beginning.

	signal	min	max	description
Specifier				
	noutputs	1	32	Specifies how many outputs are to be created.
	min number	-32768	32767	Lowest random number generated.
	max number	-32768	32767	Highest random number generated.
Control inputs				
	reset	-32768	32767	When this number changes from 0 to 1, the random sequence is restarted from the beginning.
	delay	0	32767	Control cycles between random numbers generated. 0 means no delay, 32767 is the maximum delay, giving a really long time between the generated numbers.
Control outputs				
	rand1..N			A random number between *min number* and *max number*, inclusive.

Order
C_RANDOM, modulename, number outputs, min number, max number, reset, delay, rand1, rand2 … randN

C_RELAY — GROUP: CONTROL PROCESS

Rear panel control access — rly

This module will allow direct control of the rear panel relays, as well as direct output of the status of the rear panel SW (jack) input.

	signal	min	max	description
Control Inputs:				
	relay1	0	1	will make the tip relay when 1.0
	relay2	0	1	will make the ring relay when 1.0
Control Outputs:				
	tip	0	1	the status of the tip SW input.
	ring	0	1	the status of the ring SW input.
Userobjects				
	obj			A userobject to display the in and out values, suitable for placing on a menupage.

Order
C_RELAY, modulename, relay1,relay2

Resource Usage
low

C_SAMP — GROUP: CONTROL PROCESS

Control signal sample and hold — smp

Just like SAMPLEHOLD, except this works on control signals. When *newsamp* is greater or equal to 1, *in* is passed to *out*. Otherwise *out* remains at the last value.

	signal	min	max	description
Control inputs				
	in	-32768	32767	The input to be quantized.
	newsamp	0	2	If <1 then hold last value. If > 1 then pass in to out.
Control outputs				
	Out			The output of this module

Userobjects

Obj	This module may be treated as a menupage. If this module is pointed to by head or by a menupage then if any of this module's control inputs are unconnected (left as *autoknob) they will be shown as knobs on a menu created under PARAMETERs. That menu will be titled "modulename parms".

Order

C_SAMP, modulename, in, newsamp

C_SINCOS — GROUP: CONTROL MATH

Control Sine and Cosine — csc — v2.4

This module will provide outputs giving the sine and cosine of its input signal. These will be found useful for left/right front/back panning, allowing rotation from a single control.

signal	min	max	description
Control Inputs:			
in	-1	1	
Control Outputs:			
sine	-1	1	the sine of the input signal
cosine	-1	1	the cosine of the input signal
Userobjects			
obj			A userobject to display the in and out values, suitable for placing on a menupage.

Order

C_SINCOS modulename in

Resource Usage

Low

in	-1.00	-0.75	-0.50	-0.25	0.00	0.25	0.50	0.75	1.00
sine	0.00	1.00	0.00	-1.00	0.00	1.00	0.00	-1.00	0.00
cosine	1.00	0.00	-1.00	0.00	1.00	0.00	-1.00	0.00	1.00

C_SMOOTH — GROUP: CONTROL PROCESS

Control Signal Smoother — smu

This module will make a control signal smoother, by only letting it change slowly. It will interpolate between the old and the new value.
The output is calculated by:

newout = in*speed + oldout*(1-speed)

This calculation is performed every control signal update cycle.

Control inputs:

in

The new value the signal is trying to go to.

speed

How fast the signal is allowed to change. Range 0 to 1.

Control Outputs:

out

The smoothed output control signal.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

C_SMOOTH modulename in speed

C_SQRT — GROUP: CONTROL MATH

Control Signal Square Root — csq — v2.4

This module will provide an output giving the square root of its input signal.

signal	min	max	description
Control Inputs:			
in	0	32767	

Control Outputs:

out	0	181	the square root of the value at *in*.

Userobjects

obj	A userobject to display the in value, suitable for placing on a menupage.

Order

C_SQRT, modulename, in

Resource Usage

low

C_SUBTRACT — GROUP: CONTROL MATH

Subtract Two Control Signals — sub

This module computes the difference of two control signals. The output is equal to in1 - in2. The resultant value cannot exceed -32768 or +32767.

Control inputs:

in1 in2

The inputs to be calculated. Range: -32768.0 to 32767.0.

Control outputs:

out

The result of in1 minus in2. If the resultant value exceeds -32768 or +32767 it will be limited at those values.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

C_SUBTRACT modulename in1 in2

C_SWITCH — GROUP: CONTROL PROCESS

Select One of N Control Inputs — swt

This module switches between one of N control inputs, depending on the value of the "select" control signal. A select value of 0,1,2 passes the value of in1,2,3, etc., to the output. Select values other than integers will cause the output to interpolate between two of the inputs. For example, if the select value is set to 1.5, the output will be a 50 % mix of in2 and in3.

Specifiers:

ninputs

number of inputs. Range: 1 to 32.

Control inputs:

select

Selects which input is to be passed on to the output. Range: 0 to ninputs-1.

in1 in2 ... inN

The input control signals that are to be switched. Range: -32768.0 to 32767.0.

Control outputs:

out

The selected output.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

C_SWITCH modulename ninputs select in1 in2 ... inN

C_TABLE — GROUP: CONTROL PROCESS

Control Signal Table Look-up — tbl

This module implements a table look-up for control signals. A variable number of table entries are stored with this module. The output will assume the value of one of the table entries, depending on the state of the select signal. If the select signal is 0,1,2,etc, the output will have the value of table entry 0,1,2 etc. If the Select signal has a fractional value, the output will interpolate between two table entries.

Specifiers:

nentries

Specifies the number of table entries. From 1 to 32.

entry1 entry2 ... entry N

The entries for the table. Range: -32768.0 to 32767.0.

Control inputs:

select

Selects which table entry is to be passed on to the output. Range: 0 to nentries-1.

Control outputs:

out

The selected table entry.

Userobjects:

obj

Menupage of the select input and the entries. (collection)

Order:

C_TABLE modulename nentries select entry1 entry2 ... entryN

C_TIMER — GROUP: CONTROL PROCESS

Real time clock/timer — tim

This module will produce an output in seconds showing how long the RUN input was 1.0. If RESET goes from below 1.0 to 1.0 the output will be set to zero. It will be found useful for timing external events with reasonable accuracy, such as tap-tempo controls.

	signal	min	max	description
Control Inputs:				
	run	0	1	count when equal to or above 1.0
	reset	0	1	set count to zero on +ve edge
Control Outputs:				
	Out	0	32767	The count in seconds

Order

C_TIMER modulename run reset.

Resource Usage

Low

Example Sigfile:

```
HEAD "adc" adc-null adc-null "Stopwatch" "" 3 texttrigger-obj reset-obj monitor-obj
TEXTTRIGGER "texttrigger" 2 c_flop-out "run" "stop"
C_FLOP "c_flop" 0 0 texttrigger-out
TRIGGER "reset" "reset" "reset"
C_TIMER "c_timer" c_flop-out reset-out
MONITOR "monitor" c_timer-out "Time: %4.2f secs" "time"
TAIL "njr"
```

C_TO_A — GROUP: BRIDGE

Control to Audio — c_a

This module converts a control signal into an audio signal. It is useful in taking a user input, like a button press or pedal input, and controlling the modulation of a particular module.

Audio outputs:

out

The resultant audio output.

Control inputs:

in

The control signal to be converted. Range: -1.0 to 1.0

Userobjects:

obj

The control input if it is not connected to a control signal.

Order:

C_TO_A modulename in

HARMONIZER® MODULES

C_TO_C — GROUP: NODE

Control Node

This is a pseudo-module, found in Vsigfile only. Its purpose is to act as a 'binding post' in *supermodules*, allowing a single control input to the module to drive multiple inputs within the module. It may also have use in tidying up connections within a graphic display. Note that this module will disappear when save as a .sig file, or when downloaded to the system, and the signal or numeric on its input will be directly connected to those inputs connected to its output.

	signal	min	max	description
Control inputs				
	in	-32768	32767	input value
Control outputs				
	out	-32768	32767	output value (same as input).

Order

C_TO_C modulename in

Resource Usage

none at all

C_TWEAK_A — GROUP: BRIDGE

Control to Audio, fine. c_a

This module converts a control signal into an audio signal. Before conversion, the signal is divided by 1024. This feature gives you finer control of the audio signal.

Audio outputs:

out

The resultant audio output.

Control inputs:

in

The control signal to be converted Range: -1024.0 to 1024.0

Userobjects:

obj

The control input if it is not connected to a control signal.

Order:

C_TO_A modulename in

C_XOR — GROUP: CONTROL MATH

Logical Exclusive OR Control Signals xor

This module executes a logical Exclusive OR of two control signals. An input signal of greater than or equal to 1 (values 1 through 32767) is considered to be a logical true. An input signal of less than 1 (values -32768 through .99999) is considered to be a logical false. If one of the inputs is TRUE and the other is FALSE, the output is set to TRUE, otherwise it is set to FALSE.

Control inputs:

in1 in2

The inputs to this function. Range: -32768.0 to 32767.0.

Control outputs:

out

The logical XOR for the inputs.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

C_XOR modulename in1 in2

COMB — GROUP: DELAY

Comb Filter com

Like the allpass filter, the comb filter module is a building block to be used in creating reverb simulations. Also like the allpass filter, the comb filter is a delay line with feedback. The difference is, the comb filter does NOT have a flat frequency response. In fact, the frequency response is periodic, resembling the teeth of a comb.

The comb module is useful as a simple repeating delay. To build a reverb, several comb filters are typically connected in parallel. This parallel combination is then typically connected to several allpass filters in series.

Specifiers:

maxdelay

Specifies a maximum delay. This is the most delay someone might want to use. Range: 1 to 660 milliseconds.

Audio inputs:

in

audio input

Audio outputs:

out

audio output

Control inputs:

delayamt

This controls the actual amount of delay in the feedback loop. Range: 0 to maxdelay milliseconds.

feedback

This controls the amount and polarity of the feedback. Range: -1.0 to 1.0

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

COMB modulename maxdelay in delayamt feedback

COMPARATOR GROUP: MATH

Audio Signal Comparator cmp

This module compares the value of one audio signal to that of another. If the value of the first signal is greater, the output is set to full scale. If the value is smaller, the output is set to minus full scale.

The comparator has a hysteresis control that makes the "turn-on" value greater than the "turn-off" value. This prevents undue oscillation of the output.

Audio inputs:

in1

The "positive" input. If this is greater than in2, the output goes positive

in2

The "negative" input.

Audio outputs:

out

If in1 > in2 then out = 1, else out = -1

Control inputs:

hysteresis

This controls the amount of hysteresis. A setting of zero will make the turn-on and turn-off thresholds identical. A setting of 1.0 will cause the thresholds to be +/- full scale. Range: 0.0 to 1.0.

Userobjects:

obj

The control input if it is not connected to a control signal.

Order:

COMPARATOR modulename in1 in2 hysteresis

COMPRESSOR GROUP: DYNAMIC

Soft Knee Compressor cpr

This a dynamic range compressor with separate inputs for the signal whose gain is to be processed and for the the detection (sidechain) input. It features a ‘soft knee’, giving a smooth translation or gain around the threshold point.

	signal	min	max	description
Specifiers:				
	none			
Audio Inputs:				
	In	-1	1	The audio input to be compressed.
	Sidechain	-1	1	The audio input whose level is measured and is used to alter the dynamics of the "in" audio input.
Audio Outputs:				
	out	-1	1	The output of the compressor.
Mod outputs:				
	lingain	-1	1	The gain applied to the signal. Includes additional gain set by control.
	loggain	-1	1	The gain in logarithmic terms. Does include the additional gain.

Control Inputs:

threshcntl	-100	0 dB	The threshold at gain reduction begins taking place. For sidechain below this threshold, the gain of the input is not affected.
kneecntl	0	24 dB	The width (in dB.) of the soft knee. The soft knee is a region, above the threshold, over which the ratio transitions from 1:1 to the selected ratio.
ratiocntl	1	100	The amount of gain reduction that occurs the sidechain input has gone above the threshold. The value entered selects how many dB of gain reduction occur for every dB the sidechain input is above the threshold.
gaincntl	0	24 dB	This control allows gain to be added to the output signal to make up for the gain lost by gain reduction.
attackcntl	0	10.0 secs.	This control determines how fast the compressor will respond to increasing level at the sidechain input.
decaycntl	0	10.0 secs	This control determines how fast the will respond to decreasing level at the sidechain input.

Order:

COMPARATOR modulename in sidechain threshcntl kneecntl ratiocntl gaincntl attackcntl decaycntl

CONSTANT — GROUP: MATH

Constant — con

Constant audio signal source. This provides a fixed signal value, not a waveform.

Specifier:

value

The value the constant is to produce. Range: -1.0 to 1.0.

Mod outputs:

out

The constant output.

Order:

CONSTANT modulename value

COSINE — GROUP: MATH

Cosine Function — cos

This multichannel module returns either of two cosine functions depending on its func specifier. If func is 0 the output is cos(pt), while if func is 1 the output becomes cos(pt/2). Alternatively one can consider it to be a cos(t) function where the input t is an audio or mod input from –1 to 1 that corresponds either to -180.. 180 (func: 0) or -90 .. 90 (func: 1). It will be useful in mathematics based algorithms, but may find other applications as well.

	signal	min	max	description
Specifier				
	nterms	2	4	Indicates how many polynomial terms should be used to approximate the sine function. Higher values give a more accurate result but use more resources.
	nios	1	32	Indicates how many input/output channels are provided.
	func	0	1	Indicates the type of cosine function. 0 : input swing from -1..1 gives output corresponding to -180..180 1 : input swing from -1..1 gives output corresponding to -90..90
Audio inputs				
	in1 .. N			N audio inputs for the N cosine functions.
Audio outputs				
	out1 .. N			N Audio output from the N cosine functions.

Order

COSINE modulename nterms nios func in1 in2 .. inN out1 out2 .. outN.

Resource Usage

low — nios: 4 — 6% (Orville)

CROSSIN — GROUP: MIXER

Input Crossfader — crs

Select input (mod) will cross fade from one audio input to another. Zero at select connects in1 to the output. One selects in2. A value of half mixes half of each input.

Use this module for smoothly fading from one sound to another. Since the select input is mod type, you can do very fast changes without any "zipper" effect.

Audio inputs:

in1,in2

The two inputs to be crossfaded.

Audio outputs:

out

The result of the crossfade.

Mod Inputs:

select

Select which input is at the output. Zero gets in1, One gets in2, Negative one puts the inverse of in2 onto the output.

Order:

CONSTANT modulename in1 in2 mod

CROSSOUT — GROUP: MIXER

Output Crossfader **crs**

Select input (mod) will route one audio input from one output to another. Zero at select connects in to output1. One selects output2. A value of half puts attenuates the input by half and places it onto both outputs. This is a linear pan.
Use this module for smoothly fading a sound into two different effects. Its nice if the effects (like reverbs) extend the sound after the input is removed.

Audio inputs:

in

The input to be routed.

Audio outputs:

out1,out2

The audio outputs.

Mod Inputs:

select

Select which ouput has the input. Zero puts it in output1, One gets output2, Negative one puts the inverse of in into output2.

Order:

CONSTANT modulename in1 mod

CROSSOUTQ — GROUP: MIXER

Quad Cross-out **xoq** **Orville only**

This module is a single input quad panner. The *frontrear* and *leftright* mod inputs will route the audio input to the four audio outputs, whose linear levels will be as expressed below.

frontleft	(1-leftright)*frontrear
frontright	leftright*frontrear
rearleft	(1-leftright)*(1-frontrear)
rearright	leftright*(1-frontrear)

	Signal	Min	max	description
Audio inputs				
	in			Audio stream to be routed.
Audio outputs				
	frontleft,frontright,rearleft,rearright			The audio outputs.
Mod inputs				
	frontrear			Pans the input between the front and rear outputs.
	leftright			Pans the input between the left and right outputs.

Order

CROSSOUTQ modulename in frontrear leftright.

Resource Usage

low

CURVE — GROUP: MATH

Mapping or waveshaping function. **crv**

An arbitrary relationship between an input value and an output value.
Let's say you have a sliding fader. You want to be silly and have the output of the fader be:

All the way up:	output = 0.
Three quarters:	output = .25
Half up:	output = .75
One quarter:	output = 1.
All the way down:	output = 0.

You have now set up an arbitrary relationship between the fader position and an output. What happens if the fader is one eight up? The output will be .5. Since one eight is halfway between one quarter and down, the output is half the value of one quarter (1) and half the value of down (0).

With a CURVE module, you specify how many points you want to specify. The points will be equally spaced across the range of the input. You then tell what should be the output value for each point. The more points you use, the finer control over the shape of the curve and the more resources the module will use.

A CURVE module will generate a userobject that you can include on a MENUPAGE. You can see and adjust the shape of your curve from the front panel. Going back to our example, the fader is horizontal along the bottom of the graph. The height of the line above a fader position tells you the output.

This CURVE module is full bandwidth. Use it to make interesting distortion effects.

Specifiers:

npoints

the number of points in the mapping function. Range: 1 to 32.

point1 point2 ... pointN

The points describing the curve Range: -1.0 to 1.0.

Audio inputs:

in

The input to be mapped or shaped.

Audio outputs:

out

The mapped output.

Userobjects:

obj

the map of the function

Order:

CURVE modulename in npoints point1 point2 ... pointN

This module can be edited under Vsigfile using the Waveform editor.

DE_EMPHASIS GROUP: FILTER

De-Emphasis filter dee

Provides, to within 0.5 dB, the standard 50 and 15 microsecond de-emphasis. De-emphasis may also performed by the analog output stage.

Pre-emphasis is provided by the analog input section on early DSP4000s, and may be present on some digital input signals. You can use this module to remove such pre-emphasis. Note that on early 4000s. the signal must then be re-emphasized before being fed to an analog output.

When the system is using digital inputs that are not pre-emphasised, this module does not affect the signal.

Audio inputs:

in

The input to the de-emphasis module.

Audio outputs:

out

The de-emphasized output.

Order:

DE_EMPHASIS modulename in

DELAY GROUP: DELAY

Audio Delay dly

This module implements a simple audio delay line. Any audio signal applied to the input appears at the output a specified amount of time later. The amount of delay is controllable via the *delayamt* control signal input. Note: changing the delay value while audio is present may cause clicks in the audio. If it is desired to change the delay time smoothly, use the *moddelay* module.

Specifiers:

maxdelay

Specifies the maximum delay this module will use. Range: 1 to 660 milliseconds.

Audio inputs:

in

audio input

Audio outputs:

out

audio output

Control inputs:

delayamt

Controls how much the audio will be delayed. Range: 0 to maxdelay milliseconds.

Userobjects:

obj

The control input if it is not connected to a control signal.

Order:

DELAY modulename maxdelay in delayamt

HARMONIZER® MODULES

DLYSMP — GROUP: DELAY

Small delay-based sampler **spl** ***Orville*** **and 7500 only**

This is a small small mono sampler that uses delay memory rather than sampler memory. This means that it can be used in either (or both) machine A or machine B. It also limits the maximum recording time to a total of 40 seconds per preset (at 48kHz). Use the normal sampler in machine A if you need longer recording times, or higher quality varispeed, as well as pitch change. From 4 to 8 of these samplers can be used in a single preset, so they can be paralleled for stereo or quad. This module is only available on Orville.

	signal	min	max	description
Specifiers:				
	maxdelay	5000	32760000	Specifies the maximum length audio to be recorded in mS.
Audio Inputs:				
	in			The audio input to be recorded.
Audio Outputs:				
	out			Recorded input signal, output whenever Play is hit.
Control Inputs:				
	beginpt	0	reclength	Playback start point (top).
	endpt	0	reclength	Playback end point (tail).
	fadetime	0	100	Length of fade at start, end in mS.
	playspeed	10	100	Playback speed in %.
	loopmode	0	1	If not zero, playback will loop.
Control Outputs:				
	statusout			A number representing the current operation being performed.
	position	0	reclength	The current playback position, in mS.
	reclength	0	maxdelay	The current playback position, in mS.
	beginpt	0	reclength	Playback start point (top).
	endpt	0	reclength	Playback end point (tail).
	fadetime	0	100	Length of fade at start, end in mS.
	playspeed	10	100	Playback speed in %.
Mod Inputs				
	rectrigger			Starts recording when it goes high.
	stoptrigger			Stops recording or playback when it goes high.
	playtrigger			Starts playback when it goes high.
Mod Outputs				
	posout			The current playback position.
	speedout			The current playback speed.
Included Userobjects				
	begin			a userobject for beginpt
	end			a userobject for endpt
	speed			a userobject for playspeed
	fade			a userobject fadetime
	obj			a userobject for all unconnected control inputs

Order

DLYSMP modulename maxdelay in rectrigger stoptrigger playtrigger beginpt endpt fadetime playspeed loopmode

Resource Usage

moderate/high

DETUNE — GROUP: PITCHSHIFT

Audio Signal Detuner **tun**

The detune module is used to add small amounts of pitch shift to an audio signal. This module is intended as an efficient (i.e. it doesn't use a lot of processing time) method of detuning a signal. In order to accomplish this, this module has traded off deglitch quality (glitches are a common artifact of pitch shifters) for processing efficiency. If totally glitch-free audio is needed, the pitch-shift module should be used instead of the detune module.

Specifiers:

maxdelay

Specifies the maximum delay this module will use. Range: 1 to 660 milliseconds.

Audio inputs:

in

The audio input to be detuned

Audio outputs:

out

A detuned version of the input signal.

Control inputs:

delay_ctl

Controls how much the audio will be delayed. Range: 0 to maxdelay milliseconds.

length_ctl

Controls the splice length for the detuning algorithm. Longer settings of this parameter will provide fewer "glitches" but will add more delay to the signal. Smaller settings may cause more glitches but will give a tighter sound (i.e. smaller delay). Extreme small settings may introduce modulation effects into the audio. Note also that the delay introduced as a result of this control will be variable, that is, the audio delay will be continually changing from 0 through the amount set by the length control. Range: 1 to maxdelay milliseconds.

pitch_ctl

Controls the amount of pitch detuning to be applied to the audio input. The adjustment is in "cents". A cent is one one-hundredth of a semitone. Positive value will shift the pitch upward and negative values will shift it downward. Range:-100 to 100 cents.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

DETUNE modulename maxdelay in delay_ctl length_ctl pitch_ctl

DIATONICSHIFT GROUP: PITCHSHIFT

Diatonic Pitch Shifter dsh

The diatonic pitch shift module shifts the musical pitch of an audio signal while maintaining the proper harmonic relationship to a diatonic scale. To accomplish this, the user specifies the desired key and the desired musical interval. The pitch shifter takes care of finding out what note is being played and automatically adjusts the amount of pitch shift so that the resultant note is in key.

This pitch shifter can have from 1 to 4 independent pitch shift outputs. This can be useful for creating anything from two to five-part harmonies.

The pitch shifter also has a built-in pitch detector whose results are made available through various control outputs

Specifiers:

nvoices

This specifies how many independent outputs or "voices" this module will have. Range: 1 to 4 voices

Audio inputs:

in

The audio input to be pitch shifted.

Audio outputs:

out1 out2 ... outN

This is the output of the pitch shifter. There is one output per pitch shift voice.

Mod inputs:

mod1 mod2 ... modN

modulation input for first voice. This audio input will modulate the amount of pitch shift for a particular voice. This is useful to create vibrato effects. There is one mod input for each pitch shift voice.

Control inputs:

scale

This control input selects the type of scale, or mode, the user will be playing in. The scales are as follows:

0 - Ionian (Root Major)
1 - Dorian
2 - Phrygian
3 - Lydian
4 - Mixolydian
5 - Aeolian (Relative Minor)
6 - Locrian

key

This specifies the key the user will be playing in. The values are as follows:

0 - C
1 - C#
2 - D
3 - D#
4 - E
5 - F
6 - F#
7 - G
8 - G#
9 - A
10 - A#
11 - B

quantization

Controls whether the output pitch is quantized to remain exactly within key or whether it simply tracks the input pitch. A value of zero corresponds to no quantizing. A value of 1 corresponds to full quantizing. Range -2.0 to 2.0.

glide
Controls the "glide" rate of the pitch adjustments. The adjustment is in seconds and controls the time constant that is used to smooth out changes in the amount of pitch shift that may come from changes in the interval. Range: 0 to 100.

tune
This control allows the diatonic shift to be tuned to a pitch reference other than A440 Hertz. The tuning is adjusted in cents referenced to A440. Range: -1200 to 1200 cents.

minpitch
The minpitch control is used to optimize the pitch shifting algorithm. It sets the minimum pitch that the pitch shifter is likely to hear. The values are as follows:
0 - C0
1 - C#0
2 - D0
...
46 - A#3
47 - B3

gatelevel
This control affects only the pitch detection output of this module. The gatelevel control determines at what level the pitch detector will output pitch readings. If the input signal level falls below the level set here, the pitch detect outputs will latch on to the old values. Range: -100 to 0 dB.

xfadetime
This control signal is used to optimize the sound of the pitch shifters. Larger settings may result in smoother overall sound but may add a "flanged" sound to the audio. Smaller settings will result in a crisper sound but may allow more audible pitch shifting artifacts. Range: 0 to 100 milliseconds.

shift1 shift2 ... shiftN
These control signals adjust the pitch shift interval for each voice. The values are as follows:
-21 -3 octaves
-20 -21st
...
-7 octave down
-6 seventh down
-5 sixth down
-4 fifth down
-3 fourth down
-2 third down
-1 second down
0 unison
1 second up
2 third
3 fourth
4 fifth
5 sixth
6 seventh
7 octave up
8 ninth
...
21 3 octaves up

modamt1 modamt2 ... modamtN
These control the amount of modulation to be applied to each pitch shift voice. Adjustment is in cents and it represents the amount of pitch shift that would be added to each voice if the mod input was fully on. Range: -2400 to 2400 cents.

delayamt1 delayamt2 ... delayamtN
These control the amount of delay for each pitch shift voice. Range: 0 to 600 milliseconds.

Control Outputs

pitch
The output of the pitch detector given in cents relative to middle C.

period
The output of the pitch detector given as a period. The value is in milliseconds.

freq
The output of the pitch detector given as a frequency in Hertz.

amp
The R.M.S. amplitude relative to full scale (amp equal to 1 would be a square wave 'hitting the rail').

tonality
A value representing how periodic the input signal is. A value of 1.0 is given for signals which are purely periodic. Lower values represent signals that are less periodic. The smallest value would be given for very noise-like signals.

timbre
Is a measurement of the brightness of the tone independent of its pitch. A sine wave has a timbre equal to 1, other wave shapes result in a higher timbre.

Userobjects:

obj
Menupage of control inputs not connected to control signals. (collection)

Order:

DIATONICSHIFT modulename nvoices in scale key quantization glide tune minpitch gatelevel xfadetime mod1 mod2 ... modN shift1 shift2 ... shiftN modamt1 modamt2 ... modamtN delayamt1 delayamt2 ... delayamtN

DIFFCHORUS GROUP: REVERB

Diffusor with Chorus dfc

The diffusor module creates a dense field of delay repeats that is typically used to create reverberator structures. This particular flavor of diffusor has built-in chorusing; that is, delays are randomly swept so as to prevent a build-up of undesirable resonances in the reverb.
Like the standard diffusor, the chorused diffusor is essentially a chain of series-connected allpass filters.

Specifiers:

nsections
The number of chorus sections. Range: 2 to 32.

Audio inputs:

in
The signal to be diffused.

Audio outputs:

out
The diffused signal.

Control inputs:

diffusion
The amount of diffusion. A value of 1.0 yields maximum diffusion or recirculation in the allpass filter. A value of zero yields no diffusion. Range: 0.0 to 1.0.

size
The size control scales the delays of all of the allpass filters. A value of 1.0 gives the largest size and a value of zero gives the smallest. Range: 0.0 to 1.0.

moddepth
Adjusts the range of delay modulation for the allpass filters. Range: 0 to 100 milliseconds.

modrate
Determines how fast the delays will be modulated. The adjustment is in milliseconds per second. High settings will result in noticeable pitch shift of the audio. Range: 0 to 100 milliseconds per second.

modratespan
This control determines the difference in sweep rate for all of the internal delays. With a setting of zero, all delays will be swept at the same rate. At a setting of 1.0, the delay sweep rates will vary from 0 to the modrate setting. Range: 0.0 to 1.0.

gliderate
This controls how fast the delays will respond to changes in their setting. Range: 0.0 to 1.0.

delayamt1 delayamt2 ... delayamtN
One per section. These inputs control the amount of delay used for each of the allpass sections. Range: 0.0 to 600.0 milliseconds.

g1 g2 ... gN
One per section. These control the amount of feedback for each of the individual allpass sections. Range: -1.0 to 1.0.

Userobjects:

obj
A Menupage of control inputs not connected to control signals. (collection)

Order:

DIFFCHORUS modulename nsections in diffusion size moddepth modrate modratespan gliderate delayamt1 delayamt2 ... delayamtN g1 g2 ... gN

DIFFERENTIATOR GROUP: MATH

differential function du

This module outputs how the input is changing. The output at any one point is how fast the input is changing at that point. If you put a triangle into this function, you will get a square wave out. The upward ramp of the triangle exhibits a constant rate of change. The output will be a flat positive value. When the triangle changes to downward ramp, the output will change to a flat negative value. The filtering effect is to make the lower the frequency, the softer.
The compliment of a differentiator is an INTEGRATOR. You can also use HIGHCUT with a very low frequency and lots of gain to perform integration.

Audio inputs:

in
The signal to process

Audio outputs:

out
The differential of the input

Order:

DIFFERENTIATOR modulename in

DITHER GROUP: MATH

Dithering/Requantization **dit** **v2.3**

This module is intended to reduce the word length of an audio signal, with minimum reduction in quality.
The requested dither signal is added to each of the stereo input channels, and the samples are then re-quantized to the desired resolution (word length).

	signal	min	max	description
Audio inputs				
	leftin			Left input Audio stream to be dithered.
	rightin			Right input Audio stream to be dithered.
Audio outputs				
	leftout			Left dithered Audio stream.
	rightout			Right dithered Audio stream.
Control inputs				
	wrdlen	0	3	The new resolution to quantize the audio to. 0 – 16 bits 1 – 18 bits 2 – 20 bits 3 - 24 bits (Off)

Choosing 24 bits doesn't affect the input audio stream and therefore it appears as if the module is turned off, even though the processing still takes place, using the same amount of DSP resources.

	signal	min	max	description
	dither	0	1	Controls the distribution of the dither noise. The user can choose between rectangular (uniform) or triangular distribution. Triangular distribution being more common, it is set as default. Rectangular noise distribution can be used for audio streams that have already been processed with a rectangular dither noise. 0 – Rectangular dither 1 – Triangular dither
Userobjects				
	obj			All of the parameters arranged in a menupage.

Order

DITHER, modulename, leftin, rightin, wrdlen, dither.

Demonstration Sigfile:

```
HEAD "adc" dither-leftout dither-rightout "Dither Test" "" 1 dither-obj
DITHER "dither" adc-left adc-right 0 0
TAIL "njr"
```

DIFFUSOR GROUP: REVERB

Diffusor **dfr**

The diffusor module creates a dense field of delay repeats that is typically used to create reverberator structures.
This diffusor is essentially a chain of series-connected allpass filters.

Specifiers:

nsections

This specifies how many allpass filters are to be used to construct this diffusor. Range: 2 to 32.

Audio inputs:

in

The signal to be diffused.

Audio outputs:

out

The diffused signal.

Control inputs:

diffusion

The amount of diffusion. A value of 1.0 yields maximum diffusion or recirculation in the allpass filter. A value of zero yields no diffusion. Range: 0.0 to 1.0.

size

The size control scales the delays of all of the allpass filters. A value of 1.0 gives the largest size and a value of zero gives the smallest. Range: 0.0 to 1.0.

delayamt1 delayamt2 ... delayamtN

These inputs control the amount of delay used for each of the allpass sections. Range: 0 to 660 milliseconds.

g1 g2 ... gN

These control the amount of feedback for each of the individual allpass sections Range: -1.0 to 1.0.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

DIFFUSOR modulename nsections in diffusion size delayamt1 delayamt2 ... delayamtN g1 g2 ... gN

DUCKER GROUP: DYNAMIC

Ducker duk

The ducker module is the basic building block for most dynamics control patches. It is essentially a dynamic range compressor with separate inputs for the signal whose gain is to be processed and for the detection (sidechain) input.
By connecting the input and the sidechain to the same source, a basic compressor is built. By connecting a dry signal to the sidechain and a processed signal to the input, the processed signal can be ducked (have its gain reduced) during louder passages of audio.

Audio inputs:

in

The signal to be processed.

Mod inputs:

sidechain

A mod input whose level is measured and is used to alter the dynamics of the "in" audio input

Audio outputs:

out

The processed signal.

Mod outputs:

lingain

The gain applied to the signal. Includes additional gain set by control. Use this with AMPMOD to control the gain of other signals.

loggain

The gain in logarithmic terms. Does not include the additional gain. This can be used to monitor how much compression is occurring.

Control inputs:

threshcntl

This control adjusts the threshold at which gain reduction begins taking place. For sidechain signals below this threshold, the gain of the input is not affected. Range: -100 to 0 dB.

ratiocntl

This controls the amount of gain reduction that occurs once the sidechain input has gone above the threshold. The value entered selects how many dB of gain reduction occur for every dB the sidechain input is above the threshold. Range: 1 to 100 dB.

gaincntl

This control allows gain to be added to the output signal to make up for the gain lost by gain reduction. Range: 0 to 24 dB.

attackcntl

This control determines how fast the ducker will respond to increasing level at the sidechain input. Range: 0.0 to 10.0 seconds.

decaycntl

This control determines how fast the ducker will respond to decreasing level at the sidechain input. Range: 0.1 to 10 seconds.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

DUCKER modulename in sidechain ratio threshcntl gaincntl attackcntl decaycntl

EASYTAPS GROUP: DELAY

Easy Multitap Delay Line etp

The easytaps module is a multitap delay line with a simplified user interface. It allows you to control the delay, amplitude, and pans by specifying "shapes" rather than specifying parameters for each individual tap.

Specifiers:

taps

Specifies the maximum number of delay taps to be used. This number has a major effect on the processing time of this module. Bigger numbers require more processing time. Range 1 to 64.

Audio inputs:

in

The signal to be delayed.

Audio outputs:

left right

The mix of the taps.

Control inputs:

numbertaps

Controls how many delay taps are to be heard. Range: 0 to the specified number of taps.

length
Controls the delay length. Range: 0 to 650 milliseconds.

randomizing
This control will cause the delay values for the individual taps to be slightly randomized so as to prevent the taps from having an undesirable resonance. Range: 0.0 to 1.0.

delayalpha
This controls the shape of the exponential delay curve (Only effective when delayshape is set to exponential). A setting of zero will result in constant spacing. Larger values of delayalpha will gradually exaggerate the exponential effect on delay spacing. Range: 0.0 to 1.0.

width
This control the width of the stereo image. A setting of zero will result in all taps panned center. A setting of 1.0 will give full stereo image while -1.0 will reverse the stereo image. Range: -1.0 to 1.0.

ampalpha
This controls the shape of the amplitude curve when it is set to exponential. Range: 0.0 to 1.0.

ampshape
Controls the shape of the amplitude settings of the delay taps. The values are as follows:

0 - Constant Amplitude	1 - Linearly Increasing Amplitude
2 - Linearly Decreasing Amplitude	3 - Exponentially Increasing Amplitude
4 - Exponentially Decreasing Amplitude	

delayshape
Controls the shape of the delay settings of the delay taps. The values are as follows:

0 - Constant Delay Spacing	1 - Linearly Increasing Delay Spacing
2 - Linearly Decreasing Delay Spacing	3 - Exponentially Increasing Delay Spacing
4 - Exponentially Decreasing Delay Spacing	

panshape
Controls the shape of the pan settings of the delay taps. The values are as follows:

0 - Panned Left	1 - Panned Right
2 - Panned Center	3 - Pan Left to Right
4 - Pan Right to Left.	5 - Spread from Center
6 - Merge to Center	7 - Alternating Pans

Userobjects:

obj
Menupage of control inputs not connected to control signals. (collection)

Order:

EASYTAPS modulename taps in numbertaps length randomizing delayalpha width ampalpha ampshape delayshape panshape

ENVELOPE — GROUP: OSCILLATOR

Envelope Generator — env

The envelope generator module will produce a single cycle of a specified waveshape after being triggered by an audio input signal. This is useful in creating various triggered effects from delay sweeps to sound synthesizers. Typically the output of the envelope generator will be connected to the modulation input of a filter, delay, amplitude modulator, etc.

Mod inputs:

trigger
Triggers the production of the envelope output.

mod
Modulates the frequency (length) of the envelope.

Audio outputs:

out
The envelope generator output.

Control inputs:

freq
Controls the rate of the envelope generator. Since the envelope generator output is essentially a single cycle of the selected waveform, this directly controls the length of the envelope. A setting of 1000 Hertz produces an envelope that is 1 millisecond long. A setting of 1 Hertz produces an envelope that is 1 second long. Range: 0 to 1000 Hertz.

modamt
This controls how much the mod input affects the rate of the envelope generator. Range: -1000 to 1000 Hertz.

wave
Selects the waveshape to be used. The values are as follows:
0 - Sine shape
1 - Triangle
2 - Square
3 - Peak
4 - Half sine
5 - Warp sin
6 - Full sin
7 - Full triangle
8 - Full square
9 - Full peak

dutycycle
Controls the duty cycle of the generated envelope. This control does not affect the sine wave. A setting of 0.5 will produce a 50 percent duty cycle, i.e. the waveform will be symmetrical. Duty cycle does not affect peak. Range: 0.0 to 1.0.
togglemode
Controls the mode of operation of the envelope generator. The settings are as follows:
0 - One-Shot Mode. Resets waveform to beginning at each trigger.
1 - Toggle Mode. On alternate triggers scans through wave forward then backward.
thresh
Set the threshold at which the envelope will be triggered. Range: -100 to 0 dB.
hysteresis
Controls how much the input must drop below the trigger level before a new trigger will be allowed. This is used to prevent spurious triggering of the envelope Range: 0 to 20 dB.
speed
Controls the trigger sensitivity. The setting is an averaging time for the peak detector that feeds the trigger mechanism. Range: 0.0001 to 10 seconds.

Userobjects:
obj
Menupage of control inputs not connected to control signals. (collection)

Order:
ENVELOPE modulename trigger mod freq modamt wave dutycycle togglemode thresh hysteresis speed

---

EQ GROUP: FILTER

Equalizer eq
This module implements a standard boost/cut type audio equalizer. The equalizer type can be selected to be low shelving (boost/cut below specified freq), peaking (boost/cut a particular freq band), or high shelving (boost/cut above specified freq).

Audio inputs:
in
audio input

Audio outputs:
out
audio output

Control inputs:
freq
Adjust the frequency range over which the equalizer acts. The specific effect of this parameter depends upon the type control setting. Range: 20 to 20000 Hertz.
qfactor
Controls the shape of the equalizer frequency response. For a peaking equalizer this directly controls the width of the band to be boosted/cut. The bandwidth is equal to the freq setting divided by q. This means that higher q settings result in narrower band equalization. For low and high shelving eq, the q setting affects the steepness of the eq at the specified frequency. Settings above 1.0 will also begin to have a resonant peak at the selected frequency. For high q settings, these eq types will sound very "bandpassish". Range: 0.5 to 1000.
boost
Controls how much the specified band will be boosted or cut. Positive values mean the band will be boosted (increased in level) and negative values mean the band will be cut. Range: -18.0 to 18.0 dB.
type
This controls the type of eq this module will perform. The values are as follows:
0 - Low Shelving EQ
1 - Peaking EQ (Presence)
2 - High Shelving EQ

Userobjects:
obj
Menupage of control inputs not connected to control signals. (collection)

Order:
EQ modulename in freq qfactor boost type

EXP GROUP: MATH

Exponentiator Function exp
The exp module passes an audio signal through an exponentially shaped function. This function is the complement of the log function, meaning that a signal passed through the log function and then through the exp function will be restored to its original state.
The actual function that is used is:
out = 2 ^ [(in-1)*16] for in >= 0
out = -2 ^ [(-in-1)*16] for in < 0

Audio inputs:
in
The signal to be processed.

Audio outputs:

out

The processed signal.

Order:

EXP modulename in

EXP_MOD GROUP: MATH

Exponentiator for mod signals **exp**

The exp module passes a mod-type audio signal through an exponentially shaped function. This function is the complement of the log function, meaning that a signal passed through the log function and then through the exp function will be restored to its original state.

The only difference between the exp_mod module and the exp module is that the exp_mod is only calculated at the "mod" sample rate which is 1/4 the audio sample rate.

The actual function that is used is:

out = 2 ^ [(in-1)*16] for in >= 0

out = -2 ^ [(-in-1)*16] for in < 0

Mod inputs:

in

Mod outputs:

out

Order:

EXP_MOD modulename in

EXTCONTROL GROUP: EXTERNAL

External Control **ext**

This module allows an external modulation signal to be piped into an effects program as a control signal.

While inserting this program you will be prompted for a menu to place the module in. That menu will have an entire page dedicated to an external control panel (see Chapter 2).

While inserting you will also be asked for four numbers called spec1, spec2, spec3, and spec4. Set these all to 0. That will leave the extcontrol disabled but once you have inserted the module you may adjust its settings before saving the program. The menu for the extcontrol will give you useful adjustment feedback for setting the spec values.

To use this module in place of a knob module you will probably need a c_multiply and/or a c_subtract to scale extcontrol's out to the desired range

Specifiers:

description

Text used to title the ext-mod control page.

spec1

Pre-defined values used to configure the ext-ctl interface. 0 is off.

spec2

Don't set this to anything except 0.

spec3

Don't set this to anything except 0.

spec4

Don't set this to anything except 0.

Control Output:

out

output proportional to the incoming modulation signal. Range: 0 to 1.

Order:

EXTCONTROL modulename description spec1 spec2 spec3 spec4

EXTTRIG GROUP: EXTERNAL

External Trigger Control **ext**

This module allows an external trigger signal to be piped into an effects program as a control signal.

While inserting this program you will be prompted for a menupage to place the module in. That menupage will have an entire page dedicated to an external mod control panel (see Chapter 2).

While inserting you will also be asked for four numbers called spec1, spec2, spec3, and spec4. Set these all to 0. That will leave the exttrig disabled but once you have inserted the module you may adjust its settings before saving the program. The menu for the exttrig will give you useful adjustment feedback for setting the spec values.

Specifiers:

description

Text used to title the ext-mod control page.

spec1

pre-defined values used to select the external trigger interface. 0 - off.

spec2
Don't set this to anything except 0.
spec3
Don't set this to anything except 0.
spec4
Don't set this to anything except 0.

Control Output:
out
ext trigger output

Order:
EXTTRIG modulename description spec1 spec2 spec3 spec4

FILTER — GROUP: FILTER

Audio Frequency Filter — flt

Provides a configurable and adjustable second order filter. This module can be configured as a low-pass, high-pass, band-pass, or band-reject (notch) filter.

Audio inputs:
in
audio input

Audio outputs:
out
audio output

Control inputs:
freq
Adjust the frequency range over which the filter acts. The specific effect of this parameter depends upon the type control setting. Range: 20 to 20000 Hertz.
qfactor
Controls the shape of the filter frequency response. For a bandpass filter this directly controls the width of the band to be passed. The bandwidth is equal to the freq setting divided by qfactor. This means that higher qfactor settings result in a narrower band filter. For lowpass and highpass filters, the qfactor setting affects the steepness of the filter at the specified frequency. Settings above 1.0 will also begin to have a resonant peak at the selected frequency. For high qfactor settings, these filter types will sound very "bandpassish", and may even oscillate. Range: 0.5 to 1000.
type
This controls the type of eq this module will perform. The values are as follows:
0 - Lowpass (Passes frequencies below freq setting)
1 - Bandpass (Passes freqs in a band center at freq setting)
2 - Highpass (Passes freqs above freq setting)
3 - Notch (Rejects frequencies at and around freq setting)

Userobjects:
obj
Menupage of control inputs not connected to control signals. (collection)

Order:
FILTER modulename in freq qfactor type

FIR — GROUP: FILTER

fir filter/convolution — v2.4

This convolution module allows linear-phase filtering of an audio stream. The module allows the user to select among various windows to account for the finite length of the ideal impulse response. The operation of the module works by adjusting a low and a high cutoff frequency as if they were high and low-pass filters respectively. The main output will consist of the overlapping part of those filters. Additionally, there is a second output that consists of the spectral complement of the main output. In this way, the user can create multiple bands by connecting multiple instances of fir modules in series.

	signal	min	max	description
Specifiers				
	Maxtaps	63	63	Maximum number of taps. Fixed at this time.
Audio inputs				
	in			Input audio stream to be filtered.
Audio outputs				
	outmn			Main filtered output.
	outcp			Spectral complement of main output.
Control inputs				
	lowcut	0	20000	Controls the lower cutoff frequency of the high-pass filtering part. Units are in Herz.
	highcut	0	20000	Controls the upper cutoff frequency of the low-pass filtering part. Units are in Herz.
	Numtaps	3	maxtaps	Controls the number of taps actually used for filtering. The more taps used, the better the stop band attenuation and the shorter the transition band.

Windowtype 0		20000	Controls the desired type of window. 0 – Rectangular 1 – Triangular 2 – Von Hann (Hanning) 3 – Hamming 4 – Blackmann 5 – Kaiser 6 – Dolph-Chebychev
beta	0.01	20	Additional control for the Kaiser window.

Control outputs

status	0	3	Indicates the host processing status.

0 – Filtering
1 – Sending Coefficients
2 – Calculating window
3 – Calculating impulse response

Userobjects

obj — All of the parameters arranged in a menupage.

Order

FIR modulename maxdelay in lowcut highcut numtaps windowtype beta

---

FLIPFLOP GROUP: MISCELLANEOUS

Audio Octave Divider ffp

The flipflop module generates a relatively crude method of producing a tone an octave below that of its input. If the input is a reasonably simple periodic tone, the output of this module will be a square-wave one octave lower. This module operates by changes the state of its output every time the input transitions through zero. For this reason, this module works best when provided with a simple input signal such as a sine or square wave. To get reasonable performance from any other signals, it is usually a good idea to put a lowpass filter before this module.

Audio inputs:

in

The flipflop input.

Audio outputs:

out

The flipflop output.

Control inputs:

hysteresis

Controls the amount of hysteresis, or the immunity to spurious triggering. A value of zero results in no hysteresis, while a value of 1.0 will never change state. Range: 0.0 to 1.0.

Userobjects:

obj

The control input if it is not connected to a control signal.

Order:

FLIPFLOP modulename in hysteresis

FREQSHIFT GROUP: PITCHSHIFT

Audio Frequency Shifter fsh

Provides a frequency shift of all frequencies up to 12 kHz from in to out. freq is the amount of shift in Hz and can be positive or negative.
This module uses Single Sideband Modulation to accomplish its frequency shift. Note that frequency shifting is not the same as pitch shifting. In pitch shifting, audio frequencies are multiplied by a constant factor. In frequency shifting, a constant factor is added to all frequencies. For this reason, frequency shifting audio makes it sound out of tune. However, small amount of frequency shifting (up to about 10 Hertz) can produce a very nice chorus effect.

Audio inputs:

in

The signal to be shifted.

Audio outputs:

out

The shifted signal.

Control inputs:

freq

Controls how much the input will be frequency shifted. Range: -12 kHz to +12 kHz.

Userobjects:

obj

The control input if it is not connected to a control signal.

Order:

FREQSHIFT modulename in freq

GAIN GROUP: MATH

Audio Gain Adjust gan

The gain module allows the gain of an audio signal to be increased or reduced over a wide range. The gain adjustment is in decibels (dB).

Audio inputs:

in

The signal to be amplified.

Audio outputs:

out

The amplified signal.

Control inputs:

gain

How much to amplify the signal. Range: -96 to 48 dB

Userobjects:

obj

The control input if it is not connected to a control signal.

Order:

GAIN modulename in gain

GANG GROUP: INTERFACE

Gang of User Interface Objects gng

Brings a group of objects together into a gang. This works much like a MENUPAGE module in which you insert knobs into a menu. You take knobs and insert them into the gang, and then you insert the gang into a menu as one object. It will appear as though your menu has those knobs. But as you move around the knobs, There will be a point where all knobs in the gang are selected at once. If you turn the wheel, all the knobs will change. You then can move to an individual knob and adjust it separately. In addition to knobs, you can use the different faders in a gang and you can mix different types. You can also insert monitors and textlines but they can't be adjusted.

Some limitations:

All of the objects in a gang must fit on one page. What is left over can't be accessed.

You can't put menupages, curves, or other gangs into a gang. They will be ignored

Specifier:

nobjects

How many objects to include in this gang

Userobject:

obj

The object to include on other menupages. (collection)

Included Userobjects:

object1 object2 ... objectN

User objects included in the gang.

Order:

GANG modulename nobjects object1 object2 ... objectN

GATE GROUP: DYNAMIC

Audio Noise Gate gat

This module implements a noise gate function If the input is below a specified threshold it will silence (or gate) the output.

This noise gate has adjustable attack and release times that control how fast the gate will turn on or off. With proper settings, this makes the gating function much less audible.

Audio inputs:

in

The signal to be gated.

Audio outputs:

out

The gated signal.

Mod outputs:

gain

The gain envelope that is used to gate the audio. This can be used to gate one signal with another.

Control inputs:

attack

Controls how fast the gate transits from the "off" state to the "on" state. Range: 0.0 to 10.0 seconds.

decay

Controls how fast the gate transits from the "on" state to the "off" state. Range: 0.0 to 10.0 seconds.

thresh
Controls the threshold at which the gating takes place. When the input is above the threshold, the gate is turned on, allowing audio to pass. When the input is below the threshold, the gate is turned off. Range: -100 to 0 dB.

hysteresis
Controls how much the input must drop below the trigger level before the gate can be turned on. This is used to prevent spurious triggering of the gate function. Range: 0 to 20 dB.

speed
Controls the trigger sensitivity. The setting is an averaging time for the peak detector that feeds the gate trigger mechanism. Setting this to large values will be similar to a gate "hold" function. Range: 0.001 to 10 seconds.

Userobjects:

obj
Menupage of control inputs not connected to control signals. (collection)

Order:

GATE modulename in attack decay thresh hysteresis speed

GATE2 — GROUP: DYNAMIC

Audio Noise Gate with sidechain input. gat

This module implements a noise gate function with a separate sidechain input, allowing the dynamics of one signal to control another. This feature will typically be used to remove background noise resulting from processing, with the sidechain input connected before the process, and the signal input being the processed signal.
If the sidechain input is below a specified threshold it will silence (gate) the output. Adjustable attack and release times control how fast the gate will turn on or off. Suitable use of these makes the gating function much less audible.

	signal	min	max	description
Audio Inputs:				
	In			The audio input to be gated.
	Sidechain			The audio input whose level is measured and is used to alter the dynamics of the "in" audio input.
Audio Outputs:				
	Out			The noise gated output.
	Gain			The gain envelope used to gate the audio. This can be used to control other noise gates.
Control Inputs:				
	Thresh	-100	0 dB	Controls the threshold at which the gating takes place. When the input is above the threshold, the gate is turned on, allowing audio to pass. When the input is below the threshold, the gate is turned off.
	Decay	0.0	10.0 secs	Controls how fast the gate transitions from the “on" state to the "off" state.
	Attack	0.0	10.0 secs	Controls how fast the gate transitions from the “off" state to the "on" state.
	Hysteresis	0	20 dB	Controls how much the input must drop below the trigger level before the gate can be turned on. This is used to prevent spurious triggering of the gate function.
	Speed	0.001	10.0 secs	Controls the trigger sensitivity. This sets the decay rate of a peak detector used in the gate. Setting it to large values will be similar to a gate "hold" function.
Userobjects:				
	obj			Menupage of control inputs not connected to control signals.

Order:

GATE2 modulename in sidechain attack decay thresh hysteresis speed

---

HARMONIX — GROUP: FILTER

Harmonics Generator hrm v2.4

This module will generate a variable number of harmonics of the input tone signal into separate outputs. A broad band input signal will generate more complicated outputs consisting of the sums and differences of all the frequency pairs present in the input signal. As a result, it might be advisable to band-pass filter and compress the signal before sending it to this module.

	signal	min	max	description
Specifiers				
	nharmonics 4	32	Number of output harmonics.	
	avgtime	1	5000	Time in mS over which the signal is averaged for DC removal
Audio inputs				
	in			Audio input tone.
Audio outputs				
	out[nharmonics]		The output harmonics.	

Order

HARMONIX modulename nharmonics avgtime in

HARMONIZER® MODULES

Example Sigfile

HEADM "adc" 4 4 oscillator-out iswitch-out oscillator-out iswitch-out "Empty" "Empty" 2 menupage-obj oscillator-obj
OSCILLATOR "oscillator" adc-null knob-out 0 0 0.5
KNOB "knob" "osc_freq %6.1f Hz" "" 0 20000 0.1 100
HARMONIX "harmonix" 32 1000 oscillator-out
ISWITCH "iswitch" 33 32 oscillator-out harmonix-out1 harmonix-out2 harmonix-out3 harmonix-out4 harmonix-out5 harmonix-out6 harmonix-out7 harmonix-out8 harmonix-out9 harmonix-out10 harmonix-out11 harmonix-out12 harmonix-out13 harmonix-out14 harmonix-out15 harmonix-out16 harmonix-out17 harmonix-out18 harmonix-out19 harmonix-out20 harmonix-out21 harmonix-out22 harmonix-out23 harmonix-out24 harmonix-out25 harmonix-out26 harmonix-out27 harmonix-out28 harmonix-out29 harmonix-out30 harmonix-out31 harmonix-out32
PITCHDETECT "pitchdetect" iswitch-out 0 0 0
MONITOR "pitch" pitchdetect-pitch "pitch: %8.4f" ""
MONITOR "period" pitchdetect-period "period: %8.4f" ""
MONITOR "freq" pitchdetect-freq "freq: %8.4f" ""
CONSTANT "One" 1
CONSTANT "Alpha" 0.001
CONSTANT "Minus One" -1
MULTIPLY "-Alpha" Alpha-out "Minus One-out"
ADD "1-Alpha" -Alpha-out One-out
MULTIPLY "multiply" iswitch-out Alpha-out
ADD "add" multiply-out multiply1-out
A_TO_C "a_to_c" add-out
MONITOR "monitor" a_to_c-out "DC: %8.4f" "DC: %5.3f"
MENUPAGE "menupage" "" "" 6 knob-obj iswitch-obj monitor-obj pitch-obj period-obj freq-obj
MULTIPLY "multiply1" 1-Alpha-out add-out
TAIL "tail"

HEADM

Multichannel head — hed — Orville only

This module is a multichannel version of the DSP4000's HEAD module. It provides the audio inputs to and from the preset - every preset must have either a HEAD or a HEADM. Note that whilst Orville will accept either a HEAD or a HEADM, a DSP4000 will only accept HEAD.

	signal	min	max	description
Specifiers				
	ninputs	2	4	The number of inputs used by the preset
	noutputs	2	4	The number of outputs used by the preset
	description	19 chars/text	This is the name of the preset.	
	8charname	8 chars/text	Currently unused.	
	# entries	0	100	The number of top-level menupages. For each of these there will be a soft key on the PARAMETER menu.
Audio inputs				
	in1..in4			The audio inputs to the preset. Their sources will be determined by the routing configuration.
Audio outputs				
	out1..out4			The audio outputs from the preset. Their destinations will be determined by the routing configuration.
	null			An invisible audio output that is connected to all unused audio and mod inputs to make them silent.
Control outputs				
	global1..4			These are normally zero, unless there is a c_bridge module in the other preset, when a signal connected to an input on the c_bridge will appear at the corresponding global output on this preset,
	zero			Another invisible output, this time for control signals. Connecting a control signal to this output is the same as giving it a numeric (autoknob) value of 0.
Included Userobjects				
	obj1..objn			These objects will show up as softkeys under the PARAMETER menu.
Userobjects				
	nullobj			Another invisible output, this time for unused userobject inputs.

Order

HEADM modulename ninputs noutputs out1..outn description 8charname #entries obj1..objn

HARMONIZER® MODULES

HFADER — GROUP: INTERFACE

Horizontal Fader Knob **fdr**

This module creates a horizontal fader in a PARAMETER menu. Selecting the fader in the appropriate PARAMETER menu allows the control output of the fader knob to be varied. The display for an hfader takes one half of a PARAMETER menu line (the same size as a knob module).

Specifiers:

menutext
This text not used in the current version DSP4000.

short name
Text displayed when hfader appears on DSP4000's LCD display in a PARAMETER menu page. Size: 6 characters.

min
Value for the bottom of the fader. Range: -32768.0 to 0.0.

max
Value for the top of the fader. Range: min to 32767.0

resolution
Step rate. Range: 0 to 32767.

default
Value that the hfader will be set to when it is first used. Range: from min to max.

Control Output

out
hfader output

Order:

HFADER modulename menutext shortname min max resolution default

HIGHCUT — GROUP: FILTER

Highcut Filter **hct**

Provides a simple, first order, adjustable low-pass filter. The gain at DC is always 0 dB. This filter is very useful for gently rolling off the high end to produce a warm, analog sound.

Audio inputs:

in
Input to the Filter.

Audio outputs:

out
Filtered audio output

Control inputs:

cut
Controls the cutoff frequency of the highcut. The cutoff is defined as the point at which the frequency response drops 3 dB. Range: 0 to 20000 Hertz.

Userobjects:

obj
The control input if it is not connected to a control signal.

Order:

HIGHCUT modulename in freq

HMENUPAGE — GROUP: INTERFACE

Versatile menupage **hmn**

This module is similar to menupage, creating an on-screen menu page, with two differences:

- the name on its softkey may be selected by means of a control input.
- depending on the setting of SETUP/MISC/expert_mode this menupage can be hidden - this allows the creation of expert menus that only appear when required, or similarily allows novice pages which need not trouble the expert user.

	signal	min	max	description
Specifier				
	nstrings	1	10	the number of different names the trigger may have.
	text1	8 chars/text		name when textnum=0
	text2	8 chars/text		name when textnum=1
	....			
	textn	8 chars/text		name when textnum=nstrings-1
	description	19 chars/text		This is the text that will show up in the upper right of the PARAMETER screen when this menupage is selected
	8 char name	8 chars/text	not currently used	
	entries	0	32	number of included userobjects

Control inputs

textnum	0	nstrings-1	sets the name of the menupage, by selecting from the textn strings.
explevel	-20	20	if explevel is positive and greater than the current expert mode, the menu will not be displayed. If explevel is negative and its unsigned value is less than the current expert mode, the menu will also not be displayed. This means that the menu can be either explicitly displayed or hidden above or below a given expert level. If this value is 0, the menupage will always be displayed. Note that the display status only changes when a major screen update occurs - an instantaneous change at this input will not cause the screen to change.

Userobject inputs

object1	may be a knob, other menupage, trigger, etc.
....	
objectn	

Userobjects

obj	The userobject for the trigger to be placed on a menupage or the head userobj inputs.

Order

HMENUPAGE modulename nstrings explevel textnum description 8_char_name entries object1 object2...objectn text1 text2..textn

Resource Usage

low

HMONITOR GROUP: INTERFACE

Horizontal Monitor of A Control Signal hmn

This module creates a graphical display much like a bargraph that shows the value of its control input. You need to provide the upper and lower bounds so the movement of the display is meaningful. The display will indicate if the input has gone beyond those bounds.

Specifiers:

longname

Text Statement, including %f format, which describes how the monitor signal will be displayed.

shortname

8 characters or less of text which describes the monitored signal. This text is not displayed by the DSP4000 at any time but may be used in future products

min

Minimum value

max

Maximum value

Control inputs:

in

Signal to be displayed

Userobjects:

obj

The userobject for this monitor.

Order:

HMONITOR modulename in min max longname shortname

IIR GROUP: FILTER

Audio Frequency Filter iir

Provides a configurable and adjustable high order filter. This module can be configured as a low-pass, high-pass, band-pass, or band-reject (notch) filter.

Specifiers:

i_sections

How many filter sections are used. Each section increases the order of the filter by two. For example: four sections is an eighth order filter. Note that resource usage is essentially proportional to this value. Range: 1 to 32.

Audio inputs:

in

The filter input.

Audio outputs:

out

The filter output.

Control inputs:

freq

Adjusts the frequency range over which the filter acts. The specific effect of this parameter depends upon the type control setting. Range: 20 to 20000 Hertz.

qfactor
Controls the shape of the filter frequency response. For a *bandpass* filter this directly controls the width of the band to be passed. The bandwidth is equal to the *freq* setting divided by *qfactor*, meaning that higher *qfactor* settings result in a narrower band filter. *qfactor* has no effect for high and low pass filters. Range: 0.9 to 1000.

type
This controls the type of equalization this module will perform. The values are as follows:

0 - Lowpass (Passes frequencies below *freq* setting)
1 - Highpass (Passes frequencies above *freq* setting)
2 - Bandpass (Passes frequencies in a band centered at *freq* setting)
3 - Notch (Rejects frequencies at and around *freq* setting)

Userobjects:

obj
A *menupage* with any unconnected control inputs.

Order:

IIR modulename in i_sections freq qfactor type

IMPULSE GROUP: OSCILLATOR

Impulse Generator imp

The impulse generator module creates a pulse train with a variable frequency. The width of the pulses is a single audio sample.
This module is useful in testing reverb patches. By patching the output of the impulse generator into the reverb input, the character of the reverb can be more easily assessed.
Another use is to supply a newsamp input to the SAMPHOLD module.

Mod inputs:

mod
A modulation input that affects the frequency of the impulse generator.

Audio outputs:

out
The pulse out.

Control inputs:

freq
Controls the frequency of the pulse train. For example, a setting of 10 Hertz will produce 10 pulses per second. Range: 0 to 20000 Hertz.

freqmod
Controls how much the mod input will affect the frequency of the impulse generator. Range: 0 to 20000 Hertz.

Userobjects:

obj
Menupage of control inputs not connected to control signals. (collection)

Order:

IMPULSE modulename mod freq freqmod

INTEGRATOR GROUP: MATH

Integration function int

The output of this module is the sum of every input sample in the past. If you input a positive constant the output will ramp until it clips. A square wave will produce a triangle wave since the input will be alternately adding and subtracting from the sum. There are two other inputs to this module. A one on the reset input will force the sum to it's initial value. The initial value is a control input.

Audio inputs:

in
What to add to the sum.

reset
If above .5, then the output is the initial condition.

Audio outputs:

out
The integrated output

Control inputs:

init
The initial value when reset

Order:

INTEGRATOR modulename in reset init

HARMONIZER® MODULES

ISWITCH — GROUP: MIXER

Click-less input audio signal switch — isw — v2.3

This module selects one of N audio inputs to be passed to the output.

	signal	min	max	description
Specifiers				
	ninputs	2	1024	Specifies how many audio signals the switch will select from.
Audio inputs				
	in1			
	in2			
	…			
	inN			One input for each of the audio signals.
Audio outputs				
	out			The audio signal from one and only one audio input as specified by select.
Control inputs				
	select	0	ninputs-1	Controls which one of the audio inputs will be passed along to the output.
Userobjects				
	obj			All of the parameters arranged in a menupage.

Order

ISWITCH, modulename, in1, in2 .. inN, select.

KNOB — GROUP: INTERFACE

Manual Adjust Of A Control Signal — knb

This module is associated with a line of text that appears on a PARAMETER menu. Selecting that line of text in the appropriate PARAMETER menu allows the control output of the knob module to be varied. The knob has one output and no inputs (aside from the menus in which the knob appears).

Specifiers:

menutext

Text description for PARAMETER menu. Use %f to place the value into the description.

shortname

A short name for the knob. (for future use)

min

minimum value. Range: -32768 to 0.0.

max

maximum value. Range: min to 32767.0.

resolution

step rate. Range: 0 to 32767.0.

default

value which the knob will be set to when first used. Range: min to max.

Control outputs:

out

knob output

Userobjects:

obj

The userobject for this knob.

Order:

KNOB modulename menutext shortname min max resolution default

LATTICE — GROUP: DELAY

Lattice Filter — lat

This module implements a lattice structure that can be used to build reverberators.
By connecting several of these structures in series the classic "nested allpass" filter can be created. The last lattice in the series should have its p_out connected to its m_in.

Specifiers:

maxdelay

The maximum amount of delay to be used by this module, in milliseconds. Since delay memory is a limited resource, this should be set as low as possible. Range: 1 to 660 milliseconds.

Audio inputs:

p_in

The positive input to the lattice filter.

m_in
The negative input to the lattice filter.

Audio outputs:

p_out
The positive output of the lattice. This will either be connected to the next lattice in the chain, or to the m_in if this is the last in the chain.

m_out
The negative output of the lattice. This will typically be the final output in a chain of lattices.

Control inputs:

dly_amt
This controls the amount of delay in the lattice. Range: 0.0 to maxdelay milliseconds.

g
This controls the amount of filter gain, and typically controls the amount of recirculation, or reverberation time. Range: 0.0 to 1.0.

Userobjects:

obj
Menupage of control inputs not connected to control signals. (collection)

Order:

LATTICE modulename maxdelay p_in m_in dly_amt g

LFO GROUP: OSCILLATOR

Low-Frequency Oscillator lfo

The lfo module will produce a waveform of variable shape and frequency. The waveform is produced at the mod rate, 1/4 the audio sample rate. See also oscillator.

Typically the output of the lfo will be connected to the modulation input of a filter, delay, amplitude modulator, etc. It is useful in creating flangers, chorus effects, autopanning, etc.

Mod inputs:

mod
Modulation

Mod outputs:

out
The output of the LFO.

Control inputs:

freq
Controls the rate of the LFO. Range: 0 to 1000 Hertz.

modamt
This controls how much the mod input affects the rate of the LFO. Range: -1000 to 1000 Hertz.

wave
Selects the waveshape to be used. The values are as follows:
0 - Sine shape
1 - Triangle
2 - Square
3 - Peak
4 - Warp sine
5 - Warp tri
6 - Half sine
7 - Half peak

dutycycle
Controls the duty cycle of the LFO. This control does not affect the sine wave. A setting of 0.5 will produce a 50 percent duty cycle, i.e. the waveform will be symmetrical Range: 0 to 1.0.

polarity
Controls whether the output is unipolar (only produces positive values) or bipolar (produces both positive and negative values).
1 - Bipolar
0 - Unipolar

Userobjects:

obj
Menupage of control inputs not connected to control signals. (collection)

Order:

LFO modulename mod freq modamt wave dutycycle polarity

LFO2

Low-Frequency Oscillator **lf2**

The lfo module will produce a waveform of variable shape and frequency. The waveform is produced at the mod rate, 1/4 the audio sample rate. The output can be reset to any point in its range by an edge-triggered input. See also LFO, OSCILLATOR.

Typically the output of the lfo will be connected to the modulation input of a filter, delay, amplitude modulator, etc. It is useful in creating flangers, chorus effects, autopanning, etc.

Mod inputs:

mod	Modulation
trig	When this input passes from below 0.5 to above 0.5, the oscillator output will be reset according to trigval.
trigval	The position to which the oscillator output is reset. The range from -1 to +1 corresponds to -180 to +180 degrees.

Mod outputs:

out	The output of the LFO.

Control inputs:

freq	Controls the rate of the LFO. Range: 0 to 1000 Hertz.
modamt	Controls how much the mod input affects the rate of the LFO. Range: -1000 to 1000 Hertz.
wave	Selects the waveshape to be used. The values are as follows: 0 - Sine shape 1 - Triangle 2 - Square 3 - Peak 4 - Warp sine 5 - Warp tri 6 - Half sine 7 - Half peak
dutycycle	Controls the duty cycle of the LFO. This control does not affect the sine wave. A setting of 0.5 will produce a 50 percent duty cycle, i.e. the waveform will be symmetrical Range: 0 to 1.0.
polarity	Controls whether the output is unipolar (only produces positive values) or bipolar (produces both positive and negative values). 1 - Bipolar 0 - Unipolar

Userobjects:

obj	Menupage of control inputs not connected to control signals. (collection)

Order:

LFO2 modulename mod trigval trig freq modamt wave dutycycle polarity

LMS GROUP: FILTER

LMS Adaptive Filter **lms**

The LMS module implements an adaptive filter, that is, a filter whose characteristics are dependent upon the applied input signal. The adaptive filter implemented here adapts itself so as to cancel out noise or other interference from the desired audio signal. The noisy signal is applied to the input of the filter and the noise is applied to the noise_in input. The output of the filter will be the original signal with substantially attenuated interference.

In the recording studio, this can have useful applications. One is to remove timecode leakage into an audio track. To do this, apply the audio track to the input and the timecode track to the noise_in.

If you need to remove a periodic interference but don't have a source for the interference, apply a delayed version of the audio track to the noise_in input. If the interference is periodic enough, this filter should be able to handle cancelling out the noise.

Specifiers:

num_taps
Specifies how many pairs of taps (zeros) are in the adaptive filter. Range: 1 to 50 tap pairs.

Audio inputs:

in
The audio source input (plus noise).

noise_in
The noise input (to be cancelled).

Audio outputs:

out
The noise reduced audio.

noise_out
The noise that was removed.

Control inputs:

adaptation_gain

Controls how fast the filter adapts to changing noise inputs. Bigger values adapt faster. Range: 0.0 to 1.0.

Userobjects:

obj

The control input if it is not connected to a control signal.

Order:

LMS modulename num_taps in noise_in adaptation_gain

LOG — GROUP: MATH

Logarithm — log

The log module passes an audio signal through a logarithmically shaped function. This function is the complement of the exp module, meaning that a signal passed through the log module and then through the exp module will be restored to its original state.
The actual function that is used is:
out = [logbase2(in)/16] + 1 for in >= 0
out = -[logbase2(-in)/16] - 1 for in < 0
Basicly, the output will go up .0625 each time the input doubles. Some examples:
in out <LIST>0.0: 0.0 <LIST>0.0001: 0.16952 <LIST>0.001: 0.37714 <LIST>0.01: 0.58476 <LIST>0.1: 0.79236 <LIST>0.5: 0.9375 <LIST>1.0 1.0

Audio inputs:

in

The input of the log function.

Audio outputs:

out

The log output.

Order:

LOG modulename in

LOG_MOD — GROUP: MATH

Logarithm, for mod signals — log

The log_mod module passes an audio signal through a logarithmically shaped function. This function is the complement of the exp_mod module, meaning that a signal passed through the log_mod module and then through the exp function will be restored to its original state.
The only difference between the log_mod module and the log module is that the log_mod is only calculated at the "mod" sample rate which is 1/4 the audio sample rate. The module uses less resources.

Mod inputs:

in

The input to the log function.

Mod outputs:

out

The log output.

Order:

LOG_MOD modulename in

LONGDELAY — GROUP: DELAY

Long Audio Delay — dly

This module implements an audio delay line that one uses the sampler board and thus can be very long. Any audio signal applied to the input appears at the output a specified amount of time later. The amount of delay is controllable via the delayamt control signal input. Note: changing the delay value while audio is present may cause clicks in the audio.
Maxdelay specifies how much delay you want. This can be different from how much you get. If the total desired memory of the entire patch is larger than what is on the board, then the memory is divided up proportionally. For example: you have two longdelays where one wants 10 seconds, the other wants 30. But you only have a 20 second sampler card. The first would get 5 seconds and the second would get 15 seconds. The delaymax output tells you how much you got.
NOTE: presets using this module can only run on DSP A.

Specifiers:

maxdelay	1	720	seconds. Specifies the maximum delay desired for this module.

Audio inputs

in	audio input

Audio outputs:

out	audio output

Control inputs:

delayamt	0	maxdelay	(seconds). Controls how much the audio will be delayed.

Control outputs

delaymax	How much delay was allocated.

Userobjects

obj	The control input if it is not connected to a control signal.

Order

LONGDELAY modulename maxdelay in delayamt

M_CURVE GROUP: MATH

Mapping for mod signals crv

An arbitrary relationship between an input value and an output value. See the CURVE module for a detailed explaination.
<Body Text>This module is for mod type signals that go from -1 to 1.
Use this module to warp the outputs of envelopes and LFOs

Specifiers:

npoints

the number of points in the mapping function. Range: 1 to 32.

point1 point2 ... pointN

The points describing the curve. Range: -1.0 to 1.0.

Mod Inputs:

in

The input to be mapped. Can be from -1 to 1.

Mod Outputs:

out

The mapped output.

Userobjects:

obj

The map of the function

Order:

M_CURVE modulename in npoints point1 point2 ... pointN
This module can be edited under Vsigfile using the Waveform editor.

M_UCURVE GROUP: MATH

Unipolar Mapping for mod signals crv

An arbitrary relationship between an input value and an output value. See the CURVE module for a detailed explaination.
The module generates a userobject that you can include on a menu. The userobject allows you to adjust the curve from the front panel
This module is for mod type signals that go from 0 to 1. If input is less than one, the output is the value of point zero.
This module is good to warp the outputs of envelopes and LFOs

Specifiers:

npoints

the number of points in the mapping function. Range: 1 to 32.

point1 point2 ... pointN

The points describing the curve. Range: -1.0 to 1.0.

Mod Inputs:

in

The input to the mapping module. Range: 0 to 1.

Mod Outputs:

out

The mapped output.

Userobjects:

obj

The map of the function

Order:

M_UCURVE modulename in npoints point1 point2 ... pointN
This module can be edited under Vsigfile using the Waveform editor.

MENUPAGE — GROUP: INTERFACE

Menu Page and Softkeys — mnu

This creates a menu page and menu listing. The menu page is part of the PARAMETER menu area system and either adds a new softkey or a new page under an existing softkey. See Chapter 3 for more on menupages. When you are planning a new menupage or softkey using the PATCH editor you must insert the menupage before inserting the knobs or monitors you intend to use. You cannot move a knob or monitor (or menupage) after it is placed.

Specifiers:

description
This is the text that will show in the upper right of the PARAMETER screen when this menu page is selected. Size: 19 characters.

8_char_name
8 characters or less of text which is used for the soft key in first level menupages. If this is a menupage that is inserted into an existing menupage this text will not be used. Creating this text is recommended because this name might be used in later generation Ultra-Harmonizers.

#entries
How many userobjects in this menupage. (Patch editor takes care of this for you.)

Userobjects:

obj
The userobject for this menupage.

Included Userobjects:

object1 object2 ... objectN
The userobjects that are in this menu page. (Patch editor takes care of this for you.)

Order:

MENUPAGE modulename description 8_char_name #entries object1 object2 ... objectN

METER — GROUP: INTERFACE

Meter that monitors of A Control Signal — mtr

This module creates a graphical display much like an analog meter that shows the value of its control input. You need to provide the upper and lower bounds so the movement of the display is meaningful. The display will indicate if the input has gone beyond those bounds.

Specifiers:

longname
Text Statement. You may include a %f format, which show the numeric value of the control signal.

shortname
8 characters or less of text which describes the monitored signal.

min
minimum value

max
maximum value

Control inputs:

in
Signal to be displayed Range: -32768.0 to 32767.0.

Userobjects:

obj
The userobject for this meter.

Order:

METER modulename in min max longname shortname

MICRODELAY — GROUP: DELAY

Modulatable Micro-Delay — udl

The microdelay module provides a precisely adjustable delay amount. The delay is adjustable in increments of 1/128 of an audio sample. This is done with a high-order interpolation filter. This module also provides high-quality delay modulation, maintaining full-bandwidth and minimizing aliasing artifacts. If this high quality is essential, this module should be used instead of the moddelay module.

Specifiers:

maxdelay
Maximum total delay for this module in milliseconds.
Range:
DSP4000 - 1 to 660 milliseconds.
Orville - 1 to 1360 milliseconds at 48kHz

Audio inputs:

in
audio input

Mod inputs:

mod

For delay modulation.

Audio outputs:

out

The delayed audio output.

Control inputs:

delayamt

Provides adjustment of delay time. The accuracy is better than 200 nanoseconds for 48 kHz audio. Range: 0 to maxdelay milliseconds.

modamt

Controls how much the mod input will affect the delay amount. Range: 0 to maxdelay milliseconds.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

MICRODELAY modulename maxdelay in mod delayamt modamt

MIDICLOCK — GROUP: EXTERNAL

MIDI realtime control — mck — v2.3

This module allows access to the following MIDI realtime functions: MIDI clock, MIDI start and MIDI stop. These allow a process to be controlled by, and synchronized to, an external MIDI sequencer or other controller.

	signal	min	max	description
Control inputs				
	time_in	0	10	A local time or delay setting. Will be modified according to the received MIDI clock value and sent to time_out.
	start_in	0	1	a local start trigger input
	stop_in	0	1	a local stop trigger input
	clock_in	0	10	the value of the generated MIDI clock value (where implemented). An input of 1.0 corresponds to 120 BPM.
	remote_mode	0	2	Enables/disables either the local or MIDI controls: 0: both local and MIDI effective 1: local only 2: MIDI only
Control outputs				
	time_out	0	10	The value of time_in, corrected according to MIDI clocks received. At 120 BPM, time_out will equal time_in.
	start_out	0	1	a start trigger output, being the combination of start_in and MIDI start.
	stop_out	0	1	a stop trigger output, being the combination of stop_in and MIDI stop.
	bpm_out	30	240	The BPM value currently being recieved.
Userobjects				
	obj			The remote_mode parameter in a form that can be plced on a menupage..

Order

MIDICLOCK, modulename, time_in, start_in, stop_in, clock_in, remote_mode

Example sigfile

```
HEAD "adc" adc-null adc-null "Midiclock test" "" 2 menupage-obj info-obj
KNOB "knob" "time:  %2.2f" "time" 0 10 0.1 1
KNOB "knob1" "start: %1.0f" "start" 0 1 1 0
KNOB "knob2" "stop:  %1.0f" "stop" 0 1 1 0
MIDICLOCK "midiclock" knob-out knob1-out knob2-out 1 0
MONITOR "monitor1" midiclock-bpm_out "BPM:   %3.0f" "BPM"
HMONITOR "hmonitor" midiclock-start_out 0 1 "start" "start"
HMONITOR "hmonitor1" midiclock-stop_out 0 1 "stop" "stop"
MONITOR "monitor" midiclock-time_out "time:  %2.2f" "time"
MENUPAGE "menupage" "Operate" "Operate" 8 knob-obj knob1-obj knob2-obj midiclock-obj monitor-obj hmonitor-obj hmonitor1-obj monitor1-obj
TEXTBLOCK "info" 3 "This is a simple program to illustrate " "the use of the MIDICLOCK module." "Nothing in, nothing out"
TAIL "njr"
```

MIDICOUT GROUP: EXTERNAL

Send Control Signal Over Midi **out**

Takes a control input and puts it out the MIDI Port.

Control inputs:

channel

What channel to send message over.
0 - This units base channel.
1 thru 16 - channel number

mode

What type of message.
0 - single controller message.
1 - double controller message. (only for controler numbers 0 thru 31) two controller messages, one MSB the other LSB

number

Which controller number to use. Range: 0 to 121.

input

The signal to output over MIDI. Range: 0.0 to 1.0.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

MIDIOUT modulename channel mode number input

MIDINOTE GROUP: EXTERNAL

midi note interface **mnt**

Turns MIDI note messages into control signals. This gives you the ability make a patch that you can play from a midi keyboard.

Specifiers:

nvoices

Number of voice outputs. Range: 1 to 16.

Control inputs:

channel

What channel to respond to.
0 - global setting
1 thru 16 - channel number
17 - omni

mode

What midi mode to use.
0 - global setting
1 - mono (many channels, 1 note per channel)
2 - poly (1 channel, many notes, round robin)

polymode

When in poly mode, how to assign voices.
0 - normal. Voices are assigned in round robin fashion.
1 - ordered. First note played is always Voice1. Then Voice2 and so on.
2 - spread1. All notes play if any keys are pressed. Voice1 is the highest, Voice2 is next highest and so on. The note assignment happens every time you press a key. If you just press one note, all voices play the same note.
3 - spread2. Just like spread1 except that note assignment also happens when you lift up a key. When you lift up from a chord, you're left with all voices playing one note.

pressure

What type of pressure should we use.
0 - global setting
1 - channel pressure
2 - key pressure

pitchbend

How much pitch bend to use.
0 - global setting
1 thru 25 - 0 thru 24 notes

Control Outputs:

gate1 gate2 ... gateN

1 if voice is on (key down), 0 if off

pitch1 pitch2 ... pitchN

the pitch in cents relative to lowest MIDI note 0 to 12800 cents

vel1 vel2 ... velN

the velocity of the note (both on and off). Range: 0 to 1.

pres1 pres2 ... presN
the pressure on the note. Range: 0 to 1.

Userobjects:

obj
Menupage of control inputs not connected to control signals. (collection)

Order:

MIDINOTE modulename nvoices channel mode polymode pressure pitchbend

MIDINOUT — GROUP: EXTERNAL

MIDI note output mno v2.2

This module takes a number of note/pressure inputs and sends corresponding MIDI messages to control an external MIDI device.

	signal	min	max	description
Specifier				
	ninputs	1	32	number of notes to send
Control inputs				
	channel base			global setting
		1	16	channel number
	gate1..N	0	1	Change from 0 to 1 sends note on. Change from 1 to 0 sends note off.
	pitch1..N	0	12700	MIDI note number (times 100)
	velocity1..N	0	127	MIDI velocity
	pressure1..N	0	127	MIDI key pressure

Order

MIDINOUT, number inputs, channel, gate1, pitch1, velocity1, pressure1, gate2…

MIX — GROUP: MIXER

Two-Input Audio Mixer mix

The mix module provides a way to add (mix) two audio signals, with adjustable attenuation on each of the two inputs.
If you need to mix more than two inputs, use "mixer".

Audio inputs:

in1 in2
The inputs to be mixed.

Audio outputs:

out
The result of amp1 * in1 + amp2 * in2.

Control inputs:

amp1
Controls the amount of attenuation for in 1. The gain adjustment is a linear value (not dB), with 1.0 being no attenuation. Negative numbers will invert the phase of the signal. Range: -1.0 to 1.0.

amp2
Attenuation adjust for in2. Range: -1.0 to 1.0.

Userobjects:

obj
Menupage of control inputs not connected to control signals. (collection)

Order:

MIX modulename in1 in2 amp1 amp2

MIXER — GROUP: MIXER

Multi-Input Audio Mixer mix

Mixes a specified number of audio signals to a single output. Each audio input signal has its own attenuation control.

Specifiers:

ninputs
number of inputs. Range: 1 to 50.

Audio inputs:

in1 int2 ... inN
There are multiple audio inputs as specified by ninputs.

Audio outputs:

out
The mix of the signals.

Control inputs:

gain1 gain2 ... gainN

There are multiple attenuation controls as specified by ninputs. gainN affects the level of inN. Range: -100 to 0 dB.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

MIXER modulename ninputs in1 in2 ... inN gain1 gain2 ... gainN

MOD_SLEW GROUP: FILTER

Slew rate limit for mod_type signals msl

This module will limit the slew rate of a signal. The slew rate is how fast the signal changes. This module works on mod signals. Use this module to smooth out abrupt changes that go to modules like modfilter, ampmod, and the like.

Mod inputs:

in

The mod signal to be processed

Mod Outputs:

out

the mod signal output

Control inputs:

pslew

Positive slew limit. Used when input is higher than output. The amount of time it takes to go from 0 to full scale. 0 disables function. Range: 0 to 60 seconds.

nslew

Negitive slew limit. Used when input is lower than output. The amount of time it takes to go from 0 to full scale. Range: 0 to 60 seconds.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

MOD_SLEW modulename in pslew nslew

MODDELAY GROUP: DELAY

Modulatable Delay mdl

The *moddelay* module is an audio delay line that can have its delay amount modulated by a mod rate signal. This is useful for creating flanging and chorus effects. This module implements its delay modulation with linear interpolation. This can result in some attenuation of high frequencies as well as potential for aliasing. In most applications, this is not a problem. However, if utmost audio quality is important, use the *microdelay* module.

Specifiers:

maxdelay

Maximum delay for this module Range varies on different systems:

DSP4000:	0 to 660 milliseconds.
Orville:	10 seconds at 48kHz

Audio inputs:

in

The signal to be delayed.

Mod inputs:

mod

For delay modulation.

Audio outputs:

out

The delayed signal.

Control inputs:

delayamt

Controls the amount of delay. Range: 0 to maxdelay milliseconds.

modamt

Controls how much the delay time will be modulated by the mod input. Range: 0 to maxdelay milliseconds.

glidespeed

Adjusts how fast changes to the delay parameter will be "glided" to their actual value. Range: 0 to 1000 milliseconds per second.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

MODDELAY modulename maxdelay in mod delayamt modamt glidespeed

MODFILTER GROUP: FILTER

Modulatable Filter mfr

This module implements a classic state-variable audio filter. It provides simultaneous lowpass, bandpass, highpass, and notch outputs. It has variable Q (1/bandwidth) and frequency and has mod rate frequency and q factor modulation inputs.
This is the module to use to create any type of swept filter effects

Audio inputs:

in
The delay input.

Mod inputs:

fmod
Modulation input for cutoff frequency.

qmod
Modulation input for q factor.

Audio outputs:

low
The lowpass filter output.

band
The bandpass filter output.

high
The highpass filter output.

notch
The notch filter output.

Control inputs:

freq
Controls the cent frequency (or cutoff) of all filter outputs. Range: 0 to 20000 Hertz.

freqmodamt
Adjusts how much the freqmod input will modulate the filter center frequency. Range: -20000 to 20000 Hertz.

q
Adjusts the sharpness of the filter. The bandwidth of the filter is freq/q. Range: 0.5 to 1000.

qmodamt
Controls how much the qmod input modulate the filter q. Range: -1000 to 1000.

Userobjects:

obj
Menupage of control inputs not connected to control signals. (collection)

Order:

MODFILTER modulename in fmod qmod freq freqmodamt q qmodamt

MONITOR GROUP: INTERFACE

Monitor A Control Signal mon

This module creates a one line display on a selected parameter menu to allow a numerical output which will describe the value of its control input. When the module is installed in a program using the patch editor, the editor will prompt for the parameter menupage to be used and for the text used in the menu.

Specifiers:

menutext
Statement, including %f format, which describes how the monitor signal will be displayed. Size: 19 characters.

shortname
8 characters or less of text which describes the monitored signal. This text is not displayed by the DSP4000 at any time but may be used in future products

Control inputs:

in
Signal to be displayed Range: -32768.0 to 32767.0.

Order:

MONITOR modulename in menutext shortname

MULTIKNOB — GROUP: INTERFACE

Multiple value knob **mkb** **v2.3** **Interface**

This module is a knob storing a variable number of preset values, the active one being chosen by a control input. It may be useful as:
(1) a user changeable "lookup table"
(2) to allow a preset to offer a number of built-in "tweaks", typically driven by a textknob, giving the name of the tweak.
The appearance of the knob is determined by a specifier.

	signal	min	max	description
Specifiers:				
	numtweaks	1	50	the number of stored values
	min	-32767	+32767	the minimum value for the output
	max	-32767	+32767	the maximum value for the output
	resolution	-32767	+32767	the change for each click of the wheel
	type	0	3	the appearance of the knob 0 : normal 1 : vfader 2 : hfader 3 : round
	val1..n	minimum	maximum	the stored values
Control Inputs:				
	tweaknum	0	numtweaks-1	the desired stored value (tweak).
Control Outputs:				
	out	-32767	+32767	
Userobjects				
	obj			The remote_mode parameter in a form that can be plced on a menupage

Order

MULTIKNOB operator_name menu_statement 8_char_name minimum maximum resolution type num_tweaks tweaknum val1..valn

Resource Usage

low, unless many tweaks.

MULTIPLY — GROUP: MATH

Multiply two audio signals **mul** **v2.4**

This module multiplies two audio signals 'in1' and 'in2'. It is the simplest way of modulating an audio signal with another one.

	signal	min	max	description
Audio inputs				
	in1,2			Audio input 1,2
Audio outputs				
	out			Audio output: in1*in2.

Order

MULTIPLY, in1, in2, out.

Resource Usage

low

MULTISHIFT — GROUP: PITCHSHIFT

Multi-output Pitch Shifter **msh**

This module shifts the pitch of an input over a range of +/- four octaves.
This pitch shifter can have from 1 to 4 independent pitch shift outputs. This can be useful for creating anything from two to five-part harmonies or very dense detuned chorus effects.
The pitch shifter also has a built-in pitch detector whose results are made available through various control outputs.

Specifiers:

nvoices

This specifies how many independent outputs or "voices" this module will have. Range: 1 to 4.

Audio inputs:

in

The audio input to be pitch shifted

Audio outputs

out1 out2 ... outN

This is the output of the pitch shifter. There is one output per pitch shift voice.

Mod inputs:

mod1 mod2 ... modN

This audio input will modulate the amount of pitch shift for a particular voice. This is useful to create vibrato effects. There is one mod input for each pitch shift voice.

Control inputs:

minpitch

The minpitch control is used to optimize the pitch shifting algorithm. It sets the minimum pitch that the pitch shifter is likely to hear. The values are as follows:

0 - C0	1 - C#0	2 - D0
...		
46 - A#3	47 - B3	

gatelevel

This control affects only the pitch detection output of this module. The gatelevel control determines at what level the pitch detector will output pitch readings. If the input signal level falls below the level set here, the pitch detect outputs will latch on to the old values. Range: -100 to 0 dB.

xfadetime

This control signal is used to optimize the sound of the pitch shifters. Larger settings may result in smoother overall sound but may add a "flanged" sound to the audio. Smaller settings will result in a crisper sound but may allow more audible pitch shifting artifacts. Range: 0 to 100 milliseconds.

shift1 shift2 ... shiftN

Controls the amount of pitch shift to be applied to the audio input. The adjustment is in "cents". A cent is one one-hundredth of a semitone. Positive value will shift the pitch upward and negative values will shift it downward. Range: -4800 to 4800 cents.

modamt1 modamt2 ... modamtN

These control the amount of modulation to be applied to each pitch shift voice. Adjustment is in cents and it represents the amount of pitch shift that would be added to each voice if the mod input was fully on. Range: -2400 to 2400 cents.

delayamt1 delayamt2 ... delayamtN

These control the amount of delay for each pitch shift voice. Range: 0 to 600 milliseconds.

Control Outputs

pitch

The output of the pitch detector given in cents relative to middle C.

period

The output of the pitch detector given as a period. The value is in milliseconds.

freq

The output of the pitch detector given as a frequency in Hertz.

amp

'amp' is the r.m.s. amplitude relative to full scale ('amp' equal to 1 would be a square wave 'hitting the rail').

tonality

A value representing how periodic the input signal is. A value of 1.0 is given for signals which are purely periodic. Lower values represent signals that are less periodic. The smallest value would be given for very noise-like signals.

timbre

is a measurement of the brightness of the tone independent of its pitch. A sine wave has a timbre equal to 1, other wave shapes result in a higher timbre.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

MULTISHIFT modulename nvoices in minpitch gatelevel xfadetime mod1 mod2 ... modN shift1 shift2 ... shiftN modamt1 modamt2 ... modamtN delayamt1 delayamt2 ... delayamtN

MULTITAP, MULTITAP2 GROUP: DELAY

Multi-Tap Delay Line mtp

This module implements a multi-tap delay line, with a selectable number of delay taps. Each tap has adjustable level, pan and delay. This can be used to create early reflections for room simulation, strange reverse reverb effect, and much more.

This module has its own built-in menupage that creates a graphical display on the LCD screen. It allows adjustment of level, pan, and delay for each tap. During the insert of this module you will be prompted for what menu to place this in.

In addition, multitap2 has a greater delay range. It also has a fixed delay output, which may be used to feed another multitap.

Specifiers:

taps

number of taps. From 2 to 50 (100 for Orville).

maxdelay (multitap2 only)

the maximum delay that may be set for any tap. Range 0 to 32500 mS.

delay1 delay2 ... delayN

How much delay for a tap. (This can be adjusted via PARAMETER)

Range:

0 to 660 mS	(DSP4000)
0 to 2000mS	(Orville).
0 to maxdelay	(multitap2)

gain1 gain2 ... gainN

How much gain for a tap. (This can be adjusted via PARAMETER) Range: -40 to 0 Db.

pan1 pan2 ... panN
Where to pan a tap. (This can be adjusted via PARAMETER) Range: -1.0 (left) to +1.0 (right).

Audio inputs:
in
The signal to be delayed.

Audio outputs:
left right
The sums of the taps.
out (multitap2 only)
A separate delay output

Control inputs:
delayamt (multitap2 only)
The delay value for out.

Userobjects:
obj
The menupage that allows you to adjust the taps.

Order:
MULTITAP modulename taps in delay1 gain1 pan1 delay2 gain2 pan2 ... delayN gainN panN
MULTITAP2 modulename maxdelay taps in delayamt delay1 gain1 pan1 delay2 gain2 pan2 ... delayN gainN panN
This module can be edited under Vsigfile using the Waveform editor.

NOISE — GROUP: OSCILLATOR

Noise Generator — **noi**
This code generator produces an audio output signal that is white noise.

Audio outputs:
noise
White noise out.

Order:
NOISE modulename

NOISESHAPE — GROUP: MATH

First Order Noise Shaper — **nsh**
This module quantizes the digital audio data and uses first order noise shaping to render the resulting quantization distortion less audible. This is especially useful in mastering operations where it is necessary to quantize 20-bit audio data to 16 bits for storage on DAT or CD.

Audio inputs:
in
The signal to noise shape.

Audio outputs:
out
The output of the noise shaper.

Control inputs:
bits
Output resolution. Range: 1 to 24 bits.

Userobjects:
obj
The control input if it is not connected to a control signal.

Order:
NOISESHAPE modulename in bits

ONESHOT — GROUP: MISCELLANEOUS

One shot generator — **sht**
When a low to high transition occurs on the input, the output goes high for a select period of time. This is retriggerable in that the output will stay high for the time period from the last low to high transition

Audio inputs:
in
signal to look for a trigger.

Audio outputs:
out
The signal output.

Control inputs:

time

How long to stay high. Range: 0 to 30000 ms.

threshold

Trigger threshold. When input crosses this value, the trigger is generated. Range: -1.0 to 1.0.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

ONESHOT modulename in time threshold

OSCILLATOR — GROUP: OSCILLATOR

Audio Oscillator — osc

The oscillator module produces a waveform of variable shape and frequency. The waveform is produced at the audio sample rate. If you are using this to slowly modulate a parameter, the more efficient LFO may work just as well.
Typically the oscillator is used to generate an audio range waveform. It is useful in creating synthesis effects and for audio range modulations.

Mod inputs:

mod

Modulates the frequency of the oscillator.

Audio outputs:

out

The oscillator output.

Control inputs:

freq

Controls the rate of the oscillator. Range: 0 to 20000 Hertz.

modamt

This controls how much the mod input affects the rate of the oscillator Range: -20000 to 20000 Hertz.

wave

Selects the waveshape to be used. The values are as follows:
0 - Sine Shape
1 - Triangle
2 - Square

dutycycle

Controls the duty cycle of the oscillator. This control does not affect the sine wave. A setting of 0.5 will produce a 50 percent duty cycle, i.e.the waveform will be symmetrical. Range: 0.0 to 1.0.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

OSCILLATOR modulename mod freq modamt wave dutycycle

OSWITCH — GROUP: MIXER

Click-less output audio signal switch — isw — v2.3

This module sends a single input to any one of N outputs.

	signal	min	max	description
Specifiers				
	noutputs	2	1024	Specifies how many audio signals to select as output.
Audio inputs				
	in			The audio input signal to be switched.
Audio outputs				
	out1,2 .. outN			The audio signals the input can be switched to.
Control inputs				
	select	0	noutputs-1	Controls which output the input audio will be switched to.
Userobjects				
	obj			All of the parameters arranged in a menupage.

Order

OSWITCH, modulename, in, select

PEAK — GROUP: DETECTOR

Peak Detector **pkd**

The peak detect module is an adjustable rectifier of audio data. It is typically used to get an indication of the level of an audio signal. This can then be used as a modulation source to create effects that vary with input level.
This module offers improved performance over the existing peakdetect module, and is thus recommended for new designs.

	signal	min	max	description
Specifiers:				
	none			
Audio Inputs:				
	in	-1	1	The input to the peak detector.
Audio Outputs:				
	out	0	1	The output of the peak detector.
Control Inputs:				
	attackcntl	0	100 secs	This controls the speed at which the output of the peak detector responds to increasing signal level. This is usually set to a very small value.
	decaycntl	0	100 secs	This adjust the speed at which the peak detector responds to decreases in signal level. This is usually set to a larger value, depending on the application.
Control Outputs:				
	none			
User Objects:				
	obj			All of the parameters arranged in a menupage.

Order

PEAK modulename in attackcntl decaycntl

PEAKDETECT — GROUP: DETECTOR

Peak Detector **pkd**

The peak detect module is an adjustable rectifier of audio data. It is typically used to get an indication of the level of an audio signal which can then be used as a modulation source to create effects that vary with input level.
The newer peak module offers improved performance over this module, and is thus recommended for new designs.

Audio inputs:

in

The input to be analyzed.

Audio outputs:

out

Output that can be used for modulation.

Control inputs:

attackcntl

This controls the speed at which the output of the peak detector responds to increasing signal level. The time specified is the time it takes for the output to reach 67% of the value at the input. This is usually set to a very small value. Range: 0 to 100 seconds.

decaycntl

This adjust the speed at which the peak detector responds to decreases in signal level. This is usually set to a larger value, depending on the application. Range: 0 to 100 seconds.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

PEAKDETECT modulename in attackcntl decaycntl

PERCENTKNOB — GROUP: INTERFACE

Percent Knob **pkb**

This is a modification of the standard knob. In this knob the output is divided by 100. So, if you insert this knob and specify a min of 0 and a max of 100, then the display will go from 0 to 100 but the output will go from 0 to 1. If you specify a resolution of .1 then the output resolution will be .001 although the display will show steps of .1.

Specifiers:

menutext

Text description for PARAMETER menu, use %?.?f format. Note that in order to make your text actually print a % on the screen you will have to use %% in the text. So, to make a display that would show Volume 45.6% you will need Volume %3.1f%%

shortname
8 character description, for future use
min
minimum value. Range: -32768 to 0.0
max
maximum value. Range: min to 32767.0
resolution
step rate. Range: 0 to 32767.0
default
Value which the knob will be set to when first used. Range: min to max.

Control outputs:
out
knob output = display/100

Order:
PERCENTKNOB modulename menutext shortname min max resolution default

PHASESHIFT GROUP: FILTER

Phase Shift pha

This module is used to build the classic phase-shift effect. The effect is created by connecting several allpass filter stages in series, and sweeping the frequency parameter. In order to hear the phase shift effect (moving notches), the output of the phaseshift module must be summed with the input through the use of an adder module.

Specifiers:
poles
Controls how many poles the phase shifter will have. More poles will cause greater phase shift effect. A setting of 6 usually works nicely. Range: 1 to 50

Audio inputs:
in
The signal to phase shift.

Mod inputs:
mod
Modulates the frequency of the notch

Audio outputs:
out
The phase shifted signal.

Control inputs:
depth
Adjusts the depth of the notch. Range: 0.0 to 1.0.

Userobjects:
obj
The control input if it is not connected to a control signal.

Order:
PHASESHIFT modulename poles in mod depth

PICODELAY GROUP: DELAY

Fine grain delay pdl v2.3

This delay module allows sample accurate small delay adjustment. The main purpose is to resynchronize two audio streams that have different group delays due to different path lengths or filtering.

	signal	min	max	description
Specifiers				
	maxdelay	0	2048	Maximum delay in samples.
Audio inputs				
	in			Input audio stream to be delayed.
Audio outputs				
	out			Delayed audio stream.
Control inputs				
	delayamt	0	maxdelay	Controls the amount the audio should be delayed. Units are in samples.
Userobjects				
	obj			All of the parameters arranged in a menupage.

Order
PICODELAY modulename in delayamt

PITCHDETECT GROUP: DETECTOR

Pitch Detector **pdt**

This module measures the pitch and other qualities of an audio signal. The resultant control and mod signals can be used to create effects that vary with changes in the musical pitch of the input. Also, the pitch detector can be used to control oscillators so as to create a pitch-triggered musical synthesizer.

Audio inputs:

in

The signal to analyze.

Mod outputs:

freqout

A fast (mod rate) output of the detected period. A full scale output represent 4096 Hertz. Anything below that is proportional.

Control inputs:

minpitch

The minpitch control is used to optimize the pitch-shifting algorithm. It sets the minimum pitch that pitch shifter is likely to hear. The values are as follows:

0 - C0
1 - C#0
2 - D0
...
46 - A#3
47 - B3

maxpitch

Used to optimize the pitch shifting algorithm. It sets the maximum pitch that pitch shifter is likely to hear. The values are as follows:

0 - C4
1 - C#4
2 - D4
...
46 - A#7
47 - B7

gatelevel

The gatelevel control determines at what level the pitch detector will output pitch readings. If the input signal level falls below the level set here, the pitch detect outputs will latch on to the old values. Range: -100 to 0 dB

Control outputs:

pitch

Pitch of the input in cents relative to middle C.

period

The period in milliseconds.

freq

Frequency in Hertz.

amp

the r.m.s. amplitude relative to full scale. amp = 1 would be a square wave `hitting the rails'

tonality

A value representing how periodic the input signal is. A value a of 1.0 is given for signals which are purely periodic. Lower values represent signals that are less periodic. The smallest value is given for very noise-like signals.

timbre

A measurement of the brightness of the tone independent of its pitch. A sine wave has a `timbre' equal to 1, other wave shapes result in a higher `timbre'.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

PITCHDETECT modulename in minpitch maxpitch gatelevel

HARMONIZER® MODULES

PITCHSHIFT GROUP: PITCHSHIFT

Pitch Shifter psh

This module shifts pitch of an input over a range of +/- three octaves. If more than one output needs pitch shifting, use multishift.
The pitch shifter also has a built-in pitch detector whose results are made available through various control outputs:.

Audio inputs:

in

The signal to pitch shift.

Audio outputs:

out

The pitch shifted signal.

Mod inputs:

mod

This audio input modules the amount of pitch shift. This is useful in creating vibrato effects. The amount of modulation is dependent on the *modamt* control input. If *modamt* is set to 900 and mod has a value of .1 then the actual modulation amount is 90 cents.

Control inputs:

minpitch

The minpitch control is used to optimize the pitch-shifting algorithm. It sets the minimum pitch that the pitch shifter is likely to hear. The values are as follows:
0 - C0
1 - C#0
2 - D0
...
46 - A#3
47 - B3

gatelevel

This control affects only the pitch detection output of this module. The *gatelevel* control determines at what level the pitch detector will output pitch readings. If the input signal level falls below the level set here, the pitch detect outputs will latch on to the old values. Range: -100 to 0 dB

xfadetime

This control signal is used to optimize the sound of the pitch shifters. Larger settings may result in smoother overall sound but may add a "flanged" sound to the audio. Smaller settings will result in a crisper sound but may allow more audible pitch-shifting artifacts. Range: 0 to 100 milliseconds.

shift

Controls the amount of pitch shift to be applied to the audio input. The adjustment is in cents. A cent is one one-hundredth of a semitone. Positive values shift the pitch upward and negative values shift it downward. Range: -4800 to 4800 cents.

modamt

These control the amount of modulation to be applied to the pitch shifter. The adjustment is in cents and it represents the amount of pitch shift that would be added if the mod input were fully on. Range: -2400 to 2400 cents.

delayamt

This controls the amount of delay for the pitch shifter. Range: 0 to 600 milliseconds.

Control Outputs

pitch

The output of the pitch detector given in cents relative to middle C.

period

The output of the pitch detector given as a period. The value is in milliseconds.

freq

The output of the pitch detector given as a frequency in Hertz.

amp

The R.M.S. amplitude relative to full scale (amp equal to 1 is a square wave 'hitting the rail').

tonality

A value representing how periodic the input signal is. The smallest value is given for very noise-like signals.

timbre

A measurement of the brightness of the tone independent of its pitch. A sine wave has a timbre equal to 1, other wave shapes result in a higher timbre.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

PITCHSHIFT modulename in minpitch gatelevel xfadetime mod shift modamt delayamt

PLEX GROUP: MATH

Reverberation Tool plx

The plex module provides a simple way of creating high quality reverberators. To create a reverberator, the outputs several delay lines are fed into the plex module and the outputs of the plex module are fed back to the inputs of the delay line. The plex module combines the delay outputs to produce an exponentially increasing density of echoes, hence a dense reverb. The signal that is to be used as the main input to the reverberator is connected to the "in" of the plex module. You can make things even more interesting by experimenting with other modules besides delay lines in the feedback structure.

Specifiers:
size
number of plex paths (fb -> out) From 2 to 24.

Audio inputs:
in
This is where your source audio goes.
fb1 fb2 ... fbN
This is where your feedback signal goes. The number of fb inputs is specified above.

Audio outputs:
out1 out2 ... outN
This is an output from a plexer. Run this back to the fb input and, for some outs, run this off to OUT or through additional modules.

Control inputs:
feedback
Controls the overall feedback amount. Range: -1.0 1.0.

Userobjects:
obj
The control input if it is not connected to a control signal.

Order:
PLEX modulename size in feedback fb1 fb2 ... fbN

QUADMIXER — GROUP: MIXER

Quadrophonic mixing and panning — qmx — v2.4 — Mixer

The quadmixer module provides a way to add (mix) two or more signals with adjustable attenuation and pan on each input. The main difference with the stereomixer module is that this one provides an additional front/rear pan for each of the inputs, thus allowing to accurately position an audio source within a quadrophonic environment while mixing multiple audio sources together.

	signal	min	max	description
Specifiers				
	ninputs	1	50	Specifies how many audio signals are to be mixed.
Audio inputs				
	in1			
	in2			
	…			
	inN			One input for each of the audio signals.
Audio outputs				
	frontleft			front left mix output.
	frontright			front right mix output.
	rearleft			rear left mix output.
	rearright			rear right mix output.
Control inputs				
	gain1			
	.....			
	gainN	-100	0	Controls the amount of attenuation for each of the inputs. The gain adjustment is a dB value. 0 dB = no attenuation, full level.
	lrpan1			
	…			
	lrpanN	-1.0	1.0	Controls the left/right pan position for each of the inputs. Pan values of –1.0, 0.0 and 1.0 correspond to pan positions of left, center and right respectively.
	frpan1			
	…			
	frpanN	-1.0	1.0	Controls the front/rear pan position for each of the inputs. Pan values of –1.0, 0.0 and 1.0 correspond to pan positions of rear, center and front respectively.
Userobjects				
	obj			All of the parameters arranged in a menupage.

Order
QUADMIXER modulename, in1, in2 .. inN, gain1, gain2 .. gainN, lrpan1, lrpan2 .. lrpanN, frpan1, frpan2 .. frpanN

Resource Usage
low, unless many inputs.

HARMONIZER® MODULES

PRE_EMPHASIS GROUP: FILTER

Pre-Emphasis **pre**

Provides, to within 0.5 dB, the standard 50 and 15 microsecond pre-emphasis. De-emphasis can be performed by the DAC oversampling filter. Pre-emphasis is also provided by the analog input section on older 4000's. This module serves to provide pre-emphasis for internally digitally generated signals.

Audio inputs:

in

The signal to be processed.

Audio outputs:

out

The pre-emphasized output

Order:

PRE_EMPHASIS modulename in

QUADRATURE GROUP: MATH

Quadrature Transformer **qad**

Hilbert quadrature transformer. Provides a -90 degree phase shift of all frequency components up to 1/4 Nyquist (1/8th of the sampling frequency). All frequency components in quad are equal in amplitude, but shifted by -90 degrees, to the frequency components in norm.

Audio inputs:

in

audio input

Audio outputs:

norm

in, delayed 120 sample times.

quad

quadrature output, relative to norm.

Order:

QUADRATURE modulename in

QUANTIZE GROUP: MATH

Audio Bit Quantizer **qnt**

The quantize module truncates a digital audio signal to the specified number of bits. If truncation distortion is not desired, use the noiseshape module. This module is useful in simulating low-resolution digitization or for creating stepped waveforms from an LFO.

Audio inputs:

in

audio input

Audio outputs:

out

quantized output

Control inputs:

bits

Controls how many bits the signal will be quantized to. The remaining low-order bits will be truncated (set to 0). Range: 1 24 bits.

Userobjects:

obj

The control input if it is not connected to a control signal.

Order:

QUANTIZE modulename in bits

REMOTER — GROUP: SYSTEM

Remote control **rem** **Orville and 7000 only**

This module is created by the system to add remote control to a preset parameter. It has no inputs or outputs and should not be modified. If an instance of this module is deleted, the remote control connection will be broken, but no other ill effects will occur. There is no obvious reason why a user would want to or be able to add this module manually, so no further documentation will be provided, except to suggest that the *modulename* gives a pointer to the parameter that it controls.

REVDLY — GROUP: DELAY

Reverse Audio Delay **rdl** **Orville and 7000 only**

This module implements an audio delay line which continuously plays out sections of audio in reverse. The duration of the section played out in reverse is selectable as is the cross fade which occurs between sections. A reset control is provided which allows the reverse effect to be synchronzied to external events.

Specifiers:

maxdelay

Specifies the maximum delay this module will use.

Audio Inputs:

in

The audio input signal to be reversed.

Audio Outputs:

out

A reversed version of the input signal.

Control Inputs:

reverseamt

Controls the duration of thereversed section of audio signal. Range: 5 to maxdelay milliseconds.

xfadetime

This controls the duration of cross-fade that occurs between the reversed sections of audio signal. Range: 1 to 100 milliseconds.

reset

This control causes the reverse delay to start playing out a new reversed section of audio immediately. It is useful for synchronizing the reverse effect.

Order:

REVDLY modulename maxdelay in reverseamt xfadetime reset

HARMONIZER® MODULES

REVERB_A GROUP: REVERB

Reverberator (12 delays) **rva**

This is the first in a family of three reverb modules, reverb_a, b and c. These modules are of high, medium, and low densities respectively.
These modules each make fairly respectable reverberators. When they are combined in a patch with delays, diffusers, EQs, etc., they can be part of very high quality reverbs.
These modules are all stereo in/stereo out and have control over decay (RT60), roomsize, predelay, and equalization. They also have built-in delay randomization that helps to reduce flutter and resonances. You also gain access to the internal delays that make up the reverberator module so that many varieties of rooms may be created.
In order to create a high quality room simulation, it is usually desirable to connect a pair of *diffusor* modules before the input to the reverberator. This provides a smooth attack to the reverb and provides a more natural build-up of reverberant energy.

Audio inputs:

leftin rightin
The signals to apply reverb.

Audio outputs:

left right
The reverberations.

Control inputs:

decay
Controls the overall RT60 of the reverb. Range: 0 to 1000 seconds.

roomsize
Adjusts the relative roomsize of the reverb. A value of 1.0 would be the largest. The actual roomsize depends on both this parameter and the individual delay settings. Range 0.0 to 1.0

predelay
Provides an overall delay before the reverberant effect. Range: 0 to 50 milliseconds.

low_freq
Controls the frequency at which the low frequency attenuation works. Range: 20 to 1000 Hertz.

high_freq
Controls the frequency at which the high frequency attenuation works. Range: 20 to 1000 Hertz.

low_atten
Controls how much frequencies below low_atten will be attenuated. This is used to quiet an overly rumbly reverb. Range: -20 to 0 dB.

high_atten
Controls how much frequencies above high_atten will be attenuated. This is used to diminish the high sizzle of the reverb and to produce a warmer sound. Range: -20 to 0 dB.

moddepth
Adjusts the amount of delay randomization. Too large values may produce an overly chorused effect. Range: 0 to 10 milliseconds.

modrate
Controls the speed at which the delays will change while being randomized. Again, too high values may result in noticeable pitch shift. Range: 0.0 to 1.0.

modratespan
Adjusts the degree to which the different internal delays will be swept at different rates. A setting of zero will result in all delays being swept at the same rate. Higher settings will spread out the sweep rate, reducing the possibility of a build-up of a noticeable pitch shift. Range: 0.0 to 1.0.

gliderate
Controls the rate at which changes to the delay values will be "glided". Affects roomsize settings as well. Range: 0.0 to 1.0.

delay1 delay2 ... delay12
The 12 internal delays of this reverb. Adjust these to create different room characteristics. Range: 0 to 50 milliseconds.

Userobjects:

obj
Menupage of control inputs not connected to control signals. (collection)

Order:

REVERB_A modulename leftin rightin decay roomsize predelay low_freq high_freq low_atten high_atten moddepth modrate modratespan gliderate delay1 delay2 delay3 delay4 delay5 delay6 delay7 delay8 delay9 delay10 delay11 delay12

REVERB_B GROUP: REVERB

Reverberator (8 delays) rvb

This is the second in a family of three reverb modules, reverb_a, b and c. These modules are of high, medium, and low densities respectively.
These modules each make fairly respectable reverberators. When they are combined in a patch with delays, diffusers, EQs, etc., they can be part of very high quality reverbs.
These modules are all stereo in/stereo out and have control over decay (RT60), roomsize, predelay, and equalization. They also have built-in delay randomization that helps to reduce flutter and resonances. You also gain access to the internal delays that make up the reverberator module so that many varieties of rooms may be created.
In order to create a high quality room simulation, it is usually desirable to connect a pair of diffusor modules before the input to the reverberator. This provides a smooth attack to the reverb and provides a more natural build-up of reverberant energy.

Audio inputs:

leftin rightin
The signals to reverberate.

Audio outputs:

left right
The reverberations.

Control inputs:

decay
Controls the overall RT60 of the reverb. Range: 0 to 1000 seconds.

roomsize
Adjusts the relative roomsize of the reverb. A value of 1.0 would be the largest. The actual roomsize depends on both this parameter and the individual delay settings. Range: 0.0 to 1.0.

predelay
Provides an overall delay before the reverberant effect. Range: 0 to 80 milliseconds.

low_freq
Controls the frequency at which the low frequency attenuation works. Range: 0 to 1000 Hertz.

high_freq
Controls the frequency at which the high frequency attenuation works. Range: 20 to 1000 Hertz.

low_atten
Controls how much frequencies below low_atten will be attenuated. This is used to quiet an overly rumbly reverb. Range: -20 to 0 dB.

high_atten
Controls how much frequencies above high_atten will be attenuated. This is used to diminish the high sizzle of the reverb and to produce a warmer sound. Range: -20 to 0 dB.

moddepth
Adjusts the amount of delay randomization. Too large values may produce an overly chorused effect. Range: 0 to 10 milliseconds.

modrate
Controls the speed at which the delays will change while being randomized. Again, too high values may result in noticeable pitch shift. Range: 0.0 to 1.0.

modratespan
Adjusts the degree to which the different internal delays will be swept at different rates. A setting of zero will result in all delays being swept at the same rate. Higher settings will spread out the sweep rate, reducing the possibility of a build-up of a noticeable pitch shift. Range: 0.0 to 1.0.

gliderate
Controls the rate at which changes to the delay values will be "glided". Affects roomsize settings as well. Range: 0.0 to 1.0.

delay1 delay2 ... delay8
The 8 internal delays of this reverb. Adjust these to create different room characteristics. Range: 0 to 80 milliseconds.

Userobjects:

obj
Menupage of control inputs not connected to control signals. (collection)

Order:

REVERB_B modulename leftin rightin decay roomsize predelay low_freq high_freq low_atten high_atten moddepth modrate modratespan gliderate delay1 delay2 delay3 delay4 delay5 delay6 delay7 delay8

HARMONIZER® MODULES

REVERB_C GROUP: REVERB

Reverberator (6 delays) rvc

This is the third in a family of three reverb modules, reverb_a, b and c. These modules are of high, medium, and low densities respectively.
These modules each make fairly respectable reverberators. When they are combined in a patch with delays, diffusers, EQs, etc., they can be part of very high quality reverbs.
These modules are all stereo in/stereo out and have control over decay (RT60), roomsize, predelay, and equalization. They also have built-in delay randomization that helps to reduce flutter and resonances. You also gain access to the internal delays that make up the reverberator module so that many varieties of rooms may be created.
In order to create a high quality room simulation, it is usually desirable to connect a pair of diffusor modules before the input to the reverberator. This provides a smooth attack to the reverb and provides a more natural build-up of reverberant energy.

Audio inputs:

leftin rightin
The signals to reverberate.

Audio outputs:

left right
The reverberations.

Control inputs:

decay
Controls the overall RT60 of the reverb. Range: 0 to 1000 seconds.

roomsize
Adjusts the relative roomsize of the reverb. A value of 1.0 would be the largest. The actual roomsize depends on both this parameter and the individual delay settings. Range: 0.0 to 1.0.

predelay
Provides an overall delay before the reverberant effect. Range: 0 to 80 milliseconds.

low_freq
Controls the frequency at which the low frequency attenuation works. Range: 0 to 1000 Hertz.

high_freq
Controls the frequency at which the high frequency attenuation works. Range: 20 to 1000 Hertz.

low_atten
Controls how much frequencies below low_atten will be attenuated. This is used to quiet an overly rumbly reverb. Range: -20 to 0 dB.

high_atten
Controls how much frequencies above high_atten will be attenuated. This is used to diminish the high sizzle of the reverb and to produce a warmer, more natural sound. Range: -20 to 0 dB.

moddepth
Adjusts the amount of delay randomization. Too large values may produce an overly chorused effect. Range: 0 to 10 milliseconds.

modrate
Controls the speed at which the delays will change while being randomized. Again, too high values may result in noticeable pitch shift. Range: 0.0 to 1.0.

modratespan
Adjusts the degree to which the different internal delays will be swept at different rates. A setting of zero will result in all delays being swept at the same rate. Higher settings will spread out the sweep rate, reducing the possibility of a build-up of a noticeable pitch shift. Range: 0.0 to 1.0.

gliderate
Controls the rate at which changes to the delay values will be "glided". Affects roomsize settings as well. Range: 0.0 to 1.0.

delay1 delay2 ... delay6
The 6 internal delays of this reverb. Adjust these to create different room characteristics. Range: 0 to 100 milliseconds.

Userobjects:

obj
Menupage of control inputs not connected to control signals. (collection)

Order:

REVERB_C modulename leftin rightin decay roomsize predelay low_freq high_freq low_atten high_atten moddepth modrate modratespan gliderate delay1 delay2 delay3 delay4 delay5 delay6

REVERB_D GROUP: REVERB

Variable Reverberator (4-32 delays) ***Orville and 7000 only***

rvd

This module is similar to *reverb_a*, with the difference that its density and total delay time can be varied.
This modules is stereo in/stereo out and has control over decay (RT60), roomsize, predelay, and equalization. It also have built-in delay randomization that helps to reduce flutter and resonances. You also gain access to the internal delays that make up the reverberator module so that many varieties of rooms may be created.

Audio inputs:

leftin rightin

The signals to apply reverb.

Audio outputs:

left right

The reverberations.

Specifiers:

maxdelay

Controls the total delay of the reverb, i.e. the sum of the maximums of the $delay_n$ values. Range: 100 to 32500 (22000 at 96k) milliseconds.

numdelays

Controls the number of individual delay sections in the relay. This determines the density of the reverb's sound. Range: 4, 8, 12, 16, 20, 24, 28, 32.

Control inputs:

decay

Controls the overall decay characteristics of the reverb. Range: 0 to 1000 seconds.

roomsize

Adjusts the relative room size of the reverb. A value of 1.0 would be the largest. The actual room size depends on both this parameter and the individual delay settings, as it is in effect a master control for the individual delay settings. Range: 0.0 to 1.0

predelay

Provides an overall delay before the reverberant effect. Range: 0 to *maxdelay/numdelays* milliseconds.

low_freq

Controls the frequency at which the low frequency attenuation works. Range: 20 to 1000 Hertz.

high_freq

Controls the frequency at which the high frequency attenuation works. Range: 20 to 1000 Hertz.

low_atten

Controls how much frequencies below *low_freq* will be attenuated. This is used to quiet an overly rumbly reverb. Range: -20 to 0 dB.

high_atten

Controls how much frequencies above *high_freq* will be attenuated. This is used to diminish the high sizzle of the reverb and to produce a warmer sound. Range: -20 to 0 dB.

moddepth

Adjusts the amount of delay randomization. Too large values may produce an overly chorused effect. Range: 0 to 10 milliseconds.

modrate

Controls the speed at which the delays will change while being randomized. Again, too high values may result in noticeable pitch shift. Range: 0.0 to 1.0.

modratespan

Adjusts the degree to which the different internal delays will be swept at different rates. A setting of zero will result in all delays being swept at the same rate. Higher settings will spread out the sweep rate, reducing the possibility of a build-up of a noticeable pitch shift. Range: 0.0 to 1.0.

gliderate

Controls the rate at which changes to the delay values will be "glided". Affects *roomsize* settings as well. Range: 0.0 (slow) to 1.0 (fast).

delay1 delay2 ... delayn

The internal delays of this reverb. Adjust these to create different room characteristics. Range: 0 to *maxdelay/numdelays* milliseconds.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

REVERB_D modulename maxdelay numdelays leftin rightin decay roomsize predelay low_freq high_freq low_atten high_atten moddepth modrate modratespan gliderate delay1 delay2 .. delayn

REVERSE GROUP: PITCHSHIFT

Reverse Shift rev

This module implements a version of reverse pitch-shifting, a fixture on Eventide Harmonizer . This module takes small segments of audio and plays them in reverse. At the same time, it can change the playback pitch. This module can also operate as a standard pitch shifter, albeit without deglitching, i.e., it sounds like an H910 (Original Eventide Harmonizer).

Specifiers:

maxdelay

Specifies the maximum delay this module will use. Range: 1 to 660 milliseconds.

Audio inputs:

in

The signal to process.

Audio outputs:

out

The reverse shifted output.

Control inputs:

delay_ctl

Controls how much the audio will be delayed. Range: 0 to maxdelay milliseconds.

length_ctl

Controls the splice length for the reverse algorithm. This controls the length of the audio segment that is played backwards. Range: 1 to maxdelay milliseconds.

pitch_ctl

Controls the amount of pitch shift to be applied to the reverse shifted audio. The adjustment is in "cents". A cent is one one-hundredth of a semitone. Positive value will shift the pitch upward and negative values will shift it downward. Range: -4800 to 4800 cents.

direction

Controls whether the shifter is operating as a reverse pitch shifter or as a standard pitch shifter (without de-glitching). The values are:
0 - reverse
1 - normal pitch shift

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

REVERSE modulename in delay_ctl length_ctl pitch_ctl direction

RFADER GROUP: INTERFACE

Round Knob fdr

Rotating the knob causes the line to rotate. Up to four vfaders may be pictured on a single screen. two rfaders may share a display page with four knobs.

Specifiers;

longname

description for future use.

shortname

6 character description for PARAMETER menu

min

Minimum value. Range: -32768.0 to 0.0.

max

Maximum value. Range: min to 32767.0.

resolution

Step rate. Range: 0 to 32767.0.

default

value which the knob will be set to when first used. Range: min to max.

Control outputs:

out

knob output

Userobjects:

obj

The userobject for the fader.

Order:

RFADER modulename longname shortname min max resolution default

SAMPHOLD — GROUP: MISCELLANEOUS

Sample and Hold a Signal — smp

This module will sample the IN input signal, as long as the NEWSAMP output remains high (>0). When NEWSAMP is low (=<0), the output will remain unchanged.

Audio inputs:

in

The input control to be sampled.

newsamp

Tells when to take a new sample of IN.

Audio outputs:

out

The currently-held sample.

Order:

SAMPHOLD modulename in newsamp

SAMPLER — GROUP: DELAY

Audio Recorder — rec

The module is capable of recording and playing mono or stereo samples with varying pitch and time scale. Up to 16 sound samples may be stored in memory so that various drum hits or vocal takes can be easily accessed. You can loop a sample, step through multiple samples on successive triggers, and trigger the sample from audio.

Presets using this module may only run on DSP A on Orville.

Audio recording:

To begin recording, trigger the record input. Recording will begin immediately and will continue until stop is triggered or all memory is used up. Another way to begin record is to trigger the triggered record. Recording will start when audio exceeds record threshold. There is a pre-trigger capture, causing a small section of audio preceding the trigger event to be recorded. For each new recording, the audio is stored into a new sample location. This can be done until all the available memory is used up.

Audio Playback:

A trigger to play causes the current sample to be played until the stop or pause inputs are triggered or the stop time is reached. You can have an audio signal trigger the playback by triggering triggered play.

The particular sample to be played back is controlled by the play select control. After recording, this control will be automatically set to the new sample. If a new play trigger is received while the current sample is playing, the new sample will begin playing immediately. The old sample will continue playing for the length of time set in the overlap control. By changing the *nextplaymode* setting, the playback can step through consecutive samples. The samples to be stepped through are controlled by the *playmin* and *playmax* controls. If looping is set to enabled, the sample will automatically repeat, and the end of the sample will be crossfaded to the beginning.

Editing:

Change start and stop to single out what part of the sample you want. If start is after stop the sample will be played in reverse. When adjusting, you have a choice of either a tape recorder scrub emulation, or *pitchscrub* mode where the sound at the point will be played looped. Fadein and fadeout allow you to adjust how the sound is turned off and on. Use *pitchamt* to adjust the pitch of the sample and *timeamt* to adjust how fast the sample is played back.

Saving Samples:

To save the recorded data, simply set save audio to yes and then save the current program as a preset from the PROGRAM menus.. This means that your preset can be quite large and may exceed the available PROGRAM memory space. To exclude the sound data from being saved, press clear before saving the preset, deleting the audio, or set save audio to no (default). To recall a saved sample, load the sampler preset with which the sample was saved.

The sound data will remain in memory even after you go to other presets unless a preset uses the memory. If you load a sampler preset that was saved without sound data, that preset will use the sound data (if any) that is found in memory.

Special parameters:

A number of control signals can be altered by the sampler itself. This can cause confusion if you connect a KNOB to the control signal input. You can turn the knob, but the value the sampler uses is something else. These control signals have associated userobjects which can be attached to your menupages. There are also associated control signal outputs to let you know what the real value is. The inputs should only be used for special effects where you must change the value from the outside.

Certain parameters are stored with each sample. When you change a sample, these parameters will change. In addition, when you load an empty sampler preset, these parameters are updated from the sample in memory. The information stored with a sample is: start time, stop time, fade in time, fade out time, overlap, loop xfade time, record mode, sample rate, pitch amount, pitch mod amount, time amount, time mod amount, and delay range.

	signal	min	max	description
Specifiers:				
	maxtime	1	720	seconds. Specifies the maximum delay desired for this module. This may not be what you get, depending on how big a sampler card is present. The output totaltime can tell you how much time you actually got.
Audio inputs:				
	recinl		left record input	
	recinr		right recorder inputs	

HARMONIZER® MODULES

Audio outputs:

recoutl	left channel monitor of what is being recorded. This output is enabled only during recording
recoutr	right channel monitor of what is being recorded. This output is enabled only during recording
playoutl	The left channel output of sample being played. These outputs are only enabled during playback.
playoutr	The right channel output of sample being played. These outputs are only enabled during playback.
playmonl	A mix of the recording monitor and the playback, left channel.
playmonr	A mix of the recording monitor and the playback, right channel

Mod inputs:

rectrigger	The presence of audio on this input can cause the sampler to begin to record. You must arm this function by triggering trigrec. Triggering stop will cancel.
playtrigger	The presence of audio on this input can cause the sampler to begin play. To arm this operation, you must trigger trigplay. Triggering stop will bring you back to normal.
pitchmod	Input for dynamically changing the playback pitch. The control signal pitchmodamt controls how much pitchmod affects the pitch.
timemod	Input for dynamically changing the playback time. The control signal timemodamt controls how much timemod affects the pitch.

Mod outputs:

playouttime	Sweeps from 0 to 1 as the sample is played. This is guaranteed to be zero for at least 1 mod sample before playback even when playing samples next to each other. You can use this as a mark of the beginning of a playback.
samp	Which sample is being played at this moment. The value is determined by taking the sample number, subtracting one, and dividing by 100. For example, sample number 3 is .03. Sample number 1 is the first. This output can be used to redirect the output of the sampler so different sounds can be modified by different effects. A value of 0 means that no samples are stored in memory.

Control inputs:

signal	min	max	description
play	0	1	A zero to one transition causes the sample to be played out. This is a trigger type control signal input.
trigplay	0	1	A control signal trigger causes the playtrigger input to be active.
stop	0	1	A control signal trigger causes the sampler to stop whatever it is doing. This includes recording and playing.
pause	0	1	A control signal trigger causes the sampler to pause wherever it is during the playback. A trigger one play causes the sample to be played from where it's left off.
record	0	1	A control signal trigger causes the sampler to begin recording.
trigrec	0	1	A control signal trigger causes the rectrigger input to be active.
clear	0	1	A control signal trigger causes the sampler to remove the currently selected sample.
clearall	0	1	A control signal trigger causes the sampler to clear the entire memory. This will remove all samples
recordmode	0	1	Selects whether the next sample recorded will be in stereo or mono. 0 for mono, 1 for stereo.
loopmode	0	1	Selects whether the sample loop around when done playing. The sample will start playing after being triggered and will loop until stop is pressed. 0 for disabled, 1 for enabled.
editmode	0	2	Selects how the sampler helps the user when the start and stop points are being adjusted. 0 - none. No help. 1 - pitchscrub. Plays the sound at the point. 2 - tapescrub. Mimics tape across heads as you jog the reels.
recthresh	-100	0	dB. The threshold at which the signal level at trigrec should be for the sampler to start recording.
pretrigtime	0	5	How much should be recorded before the command to record is triggered.
playthresh	-100	0	dB. The threshold at which the signal level at trigplay should be for the sampler to start playing.
playhyst	0	20	After the signal at trigplay has started playing back, the sampler needs to know when to trigger again. The signal at trigplay has to reach playhyst lower than playthresh before the sampler is armed for another play trigger.
playselect	0	num of	samples. Selects which stored sample will be played. This control input will change an internal value which can also be changed by the sampler. A control output of the internal value is available. For simplicity, use the select userobject.
nextplay			After a sample is played, does the sampler stay at the same sample (simple) or go to another sample (rotate)? Each sample has a number. Rotate will go to the next highest number unless the next number is larger than playmax. In that case, the next sample will be at playmin.
playmin	0	249	The sample at the bottom of the list of rotated samples.
playmax	0	249	The sample at the top of the list of rotated samples. This number can be larger than the number of samples in which case the real playmax is the top of the list.
starttime	0	maxtime	seconds. Where in the sample we start playing. This is in seconds and zero is at pretrigtime before recording started. The internal value can be changed. It is set to zero each time a new sample is recorded. This value is saved with the sample so if you change the current

			sample, this value will revert to the saved value. A userobject and a current value output are available.
stoptime	0	maxtime	seconds. Where in the sample we stop playing. This is in seconds and the internal value can be changed. If you adjust above the actual sample length, the value is brought back. It is set to the end of the sample each time a new sample is recorded. This value is saved with the sample so if you change the current sample, this value will revert to the saved value. A userobject and a current value output are available.
pitchamt	-4800	2400	cents. The amount the sample is pitch shifted. This is in cents and the internal value can be changed. This value is saved with the sample so if you change the current sample, this value will revert to the saved value. If the sample was recorded at a sample rate that is different from the current sample rate, the range will be limited. A userobject and a current value output are available.
pitchmodamt	-4800	2400	cents. The amount the sample is pitch shifted when pitchmod is at 1 is added to pitchamt. This value is saved with the sample so if you change the current sample, this value will revert to the saved value. A userobject and a current value output are available.
timeamt	0	4	The amount the sample is time stretched. A 1 will play the sample at normal rate. A 2 will be twice as fast. A .5 will be twice as slow. Zero will stop the playing. The autoknob for this actually shows a percentage. This value is saved with the sample so if you change the current sample, this value will revert to the saved value. If the sample was recorded at a sample rate that is different from the current sample rate, the range will be limited. A userobject and a current value output are available.
pitchmodamt	-1	1	The amount the sample is time stretched when timemod is at 1 added to timeamt. The autoknob will show a percentage of time stretching. This value is saved with the sample so if you change the current sample, this value will revert to the saved value. A userobject and a current value output are available.
fadeintime	0	1000	milliseconds. When the sample is started, the output is turned on slowly. This parameter governs how long it takes to fade in the sample. This value is saved with the sample so if you change the current sample, this value will revert to the saved value. A userobject and a current value output is available.
fadeouttime	0	1000	Just before the sample stops, the output is turned off slowly. This parameter governs how long it takes to fade out the sample. This value is saved with the sample so if you change the current sample, this value will revert to the saved value. A userobject and a current value output is available.
loopxfade	0	1000	milliseconds. In loop mode, the end of the sample is faded out while the beginning of the sample is faded in. This parameter governs how long it takes to perform the crossfade. This value is saved with the sample a userobject and a current value output is available.
overlaptime	0	1000	When a sample is to be played while another is still playing, There is a point where both samples are playing at the same time. This parameter governs how much the sample will overlap. This value is saved with the sample and a userobject and a current value output is available.
delayrange	10	71	milliseconds. This is usually an expert parameter where it helps to know what type of material you want to shift or stretch. A long delayrange is good for low notes, chords, and program material but can be choppy. Decreasing delayrange will smooth the playback if you have a single note source but can get glitchy if it's too small. This value is saved with the sample and a userobject and a current value output is available.
saveMode	0	1	Determines whether recorded audio is saved when a preset is saved. Set to 1 to save, 0 to not save (default).

Control outputs

Currentmode — What mode the sampler is current in. This is an integer number representing:

0 - stop: The sampler is stopping the sample.
1 - pause: Paused in the middle of playing a sample.
2 - fill: The sampler is busy.
3 - ready: Ready and waiting to do something
4 - play: Playing a sample.
5 - decay: The end of the sample is being ramped down.
6 - trigready: Waiting for an audio trigger.
7 - trigplay: Playing a sample that was triggered.
8 - trigdecay: Ending a sample that was triggered.
9 - record: Recording a sample.
10 - recordinit: Getting ready to record.
11 - trigrecord: Recording a sample that was triggered.
12 - trigrecordinit: Getting ready to record triggered.
13 - trigfill: The sampler is busy.
14 - pscrubinit: Getting ready to pitch scrub.
15 - tscrubinit: Getting ready to tape scrub.
16 - pitchscrub: Finding endpoint in pitchscrub mode.
17 - tapescrub: Finding endpoint in tapescrub mode.
18 - recorddone: Finished recording, tying up loose ends.
19 - file I/O 1: Internal sampler mode.
20 - file I/O 2: Internal sampler mode.

21 - file I/O 3: Internal sampler mode.
22 - file I/O 4: Internal sampler mode.
23 - initialize1: Internal sampler mode.
24 - initialize2: Internal sampler mode.
25 - initialize3: Internal sampler mode.
26 - initialize4: Internal sampler mode.
27 - null: The sampler is doing nothing at all.

Currenttime	The current position in the sample in seconds. The sample beginning is at 0.0 and is the first recorded moment. This is at the pretrigtime.
Currentsample	The current sample being played. If no samples are in memory, this is zero.
Recordrate	What the sample rate was when the current sample was recorded. This is in kilohertz.
playmode	How was the sample recorded: 0 mono, 1 stereo
numbersamples	Current number of samples in memory.
processstatus	This is an indicator of where the analyzer is at in the sample. Before a sample can be played back, the sample needs to be analyzed. This happens while recording and is usually done when recording is finished. However, there may be some modules that may slow down the analysis process. This output is provided to inform the user this is happening. This value is in seconds.
endtime	How large the current sample is in seconds.
totaltime	The total time available to the sampler. This will depend on the size of the big memory card. Maxtime will ask for an amount of time, where totaltime tells you how much you actually have.
playselect_out	Current value of playselect
starttime_out	Current value of starttime
stoptime_out	Current value of stoptime
pitchamt_out	Current value of pitchamt
pitchmodamt_out	Current value of pitchmodamt
timeamt_out	Current value of timeamt
timemodamt_out	Current value of timemodamt
fadein_out	Current value of fadein
fadeout_out	Current value of fadeout
loopxfade_out	Current value of loopxfadeout
overlap_out	Current value of overlapout

Userobjects

obj	All of the parameters arranged in menupages.
start	The starttime knob.
rec	The record trigger
ply	The play trigger
stp	The stop trigger
stop	The stoptime knob.
select	The playselect knob.
pitch	The pitchamt knob.
pitchmod	The pitchmodamt knob.
time	The timeamt knob.
timemod	The timemodamt knob.
fadein	The fadeinttime knob.
fadeout	The fadeouttime knob.
xfade	The loopxfade knob.
overlap	The overlaptime knob.

Order

SAMPLER, modulename, maxtime, recinl, recinr, rectrigger, playtrigger, pitchmod, timemod, play, trigplay, stop, pause, record, trigrec, clear, clearall, recordmode, loopmode, editmode, recthresh, pretrigtime, playthresh, playhyst, playselect, nextplay, playmin, playmax, saveMode, starttime, stoptime, pitchamt, pitchmodamt, timeamt, timemodamt, fadeintime, fadeouttime, loopxfade, overlaptime, delayrange

SCALE — GROUP: MATH

Audio Signal Scaler (Attenuator) scl

Scales an audio waveform. This is useful for reducing an audio signal's amplitude.

Audio inputs:

in

The signal to be scaled.

Audio outputs:

out

The scaled signal.

Control inputs:

amp

Controls the amount of attenuation for the input. The gain adjustment is a linear value (not dB), with 1.0 being no attenuation. Negative numbers will invert the phase of the signal. Range: -1.0 to 1.0

Userobjects:

obj

The control input if it is not connected to a control signal.

Order:

SCALE modulename in amp

SCALES — GROUP: MISCELLANEOUS

scl

This module implements an advanced version of diatonic pitch shifting and pitch correction. It will determine the pitch shift required to stay in key given a desired interval and will also determine the shift needed to correct an out of tune note. It does this by using the given pitch to determine what note of the current scale is being played. It will then use this information along with the selected scale, interval tuning, etc., to figure out how much pitch shift to apply.

This module is part of the diatonic processing for the UltraShifter(tm). It performs processing which is used by the *ultrashifter* module and as such is unlikely to be useful by itself.

	signal	min	max	description
Specifiers:				
	steps	12	12	The number of notes in the scale. It should be 12.
Control Inputs:				
	scale	0	18	This control selects the scale, or mode, the user will be in. The scales are as follows:

0 - User	6 - Whole-tone	12 - Lydian
1 - Major	7 - Pentatonic Major	13 - Mixolydian
2 - Minor	8 - Pentatonic Minor	14 - Aeolian
3 - Harmonic Minor	9 - Ionian	16 - Enigmatic
4 - Melodic Minor	10 - Dorian	17 - Neapolitan
5 - Chromatic	11 - Phrygian	18 - Hungarian

	signal	min	max	description
	key	0	16	This specifies the key the user will be playing in. The values are as follows:

0 - C	6 - E	12 - A♭
1 - C♯	7 - F	13 - A
2 - D♭	8 - F♯	14 - A♯
3 - D	9 - G♭	15 - B♭
4 - D♯	10 - G	16 - B
5 - E♭	11 - G♯	

interval	-14	14	Controls the interval of the desired pitch shift. If the interval specified does not exist in the current scale then the largest existing interval in the scale will be used in its place The values are as follows:

14 - 2 octaves up	2 - 3rd	-4 - 5th down
7 - octave up	1 - 2nd up	-5 - 6th down
6 - 7th	0 - unison	-6 - 7th down
5 - 6th	-1 - 2nd down	-7 - octave down .....
4 - 5th	-2 - 3rd down	-14 - 2 octaves down
3 - 4th	-3 - 4th down	

hystersis	0	50 cents	Controls how much the pitch may move outside the range of the current note before the current note is changed to reflect that pitch. *Hystersis* can prevent the oscillating between adjacent notes that can occur when the pitch is in between two notes. Notes are typically 100 to 200 cents apart so 50 cents of *hystersis* would allow the pitch to wander a lot before a higher or lower note was recognized.
correctrate	0	2000 mS	Controls how rapidly the pitch is corrected to bring it into tune. The numeric value gives the time for the pitch to change 100 cents. Actual pitch correction values will be much less than 100 cents and hence will be performed more rapidly.
mincorrect	0	50 cents	Controls the smallest error in pitch that will be corrected - if the pitch deviates from a true note by less than this amount, no correction will be made. This value may be used to let vibrato through or to enable correction only on notes that are badly out of tune.
intervalglide	1	2000 mS	Controls how rapidly the pitch is changed to bring it into key. The numeric value is the time taken for the pitch to change 100 cents.
tuning	0	4.	Selects the tuning system used. The values are as follows: 0 - Equal Temperament 1 - Just Major 2 - Just Minor 3 - Pythagorean 4 - Meantone
tune	392	494 Hz	Allows the pitch reference to be set to a value other than the usual 440 Hz.
quantize$_{1..n}$	On	Off	Controls whether or not an individual note is quantized (corrected.) Each note of the scale may be set to on or off, allowing pitch correction to operate only on desired notes.
userscale $_{1..n}$			This control selects the notes of a custom scale. The notes must be in ascending order and can only cover an octave in range from the lowest note to the highest note with no gaps. The scale must have a least five notes. If the scale does not follow these rules it will be ignored and a major scale used instead. If the scale has seven or less notes then the intervals will be derived from the scale. If it has more than seven notes a default major scale interval pattern will be used.

Control Outputs:

freq	The current frequency in Hertz.
period	The current period in milliseconds.
pitch	The current pitch given in cents relative to middle C.
pitcherror	The instantaneous pitch error in cents. This is the amount the input pitch is currently out of tune.
correction	The amount of pitch correction. This value will generally not be the same as the negative of *pitcherror*. This is because the correction is never made instantaneously, as that would crush all the inflection, resulting in a robotic sound.
intervalshift	The amount in cents of the current interval shift.
totalshift	The total pitch shift i.e. correction and interval.
spentry	A special output used in by the *ultrashifter* module to support diatonic pitch shifting.

User Objects:

obj	The main menupage.
quantize	The quantize menupage.
userscales	The userscales menupage.

Order:

SCALES modulename steps scale key interval hysteresis correctrate mincorrect intervalglide tuning tune quantize$_1$..quantize$_n$ userscale$_1$..userscale$_n$

SCOPE GROUP: MISCELLANEOUS

Single Trace Oscilloscope scp (*Orville and 7000 only*)

signal	min	max	description
Specifiers			
width	10	240	display width in pixels. Full width : 238, half width : 119.
height	10	64	display height in pixels. Full height: 41, half height: 19.
Audio inputs			
in	-1	1	the signal to be displayed.
trigin	-1	1	the trigger input when trig_mode is set to “external”.

Audio outputs			
trigout	-1	1	1 when the scope has triggered, -1 when not triggered. Can be connected to the trigin of another scope module to ensure sample-synchronized displays.
Control inputs			
trig_level	-1	-1	The level at which triggering occurs. This will happen when the input signal goes from below this level to above it.
trig_mode	0	2	How the scope triggers. 0: Single - freezes the current display 1: Normal - triggers according to the signal level on in. 2: External - triggers according to the signal level on trigin.
xgain	1	100	the horizontal scale, expressed as samples per pixel. A value of 1 will cause a single cycle of a 250Hz sinewave to fill the screen.
ygain	0	10000.0	the vertical scale. When ygain is 1.0, a maximum level signal will fill the screen, other values pro-rata.
run	0	1	0 to freeze, 1 to re-trigger.
Control outputs			
tmode	0	4	the current trigger status: 0: invalid 1: idle 2: wait_lo 3: wait_hi 4: acquire
Userobjects			
display_obj			a screen to show the display, sized according to width and height above.
menu_obj			A number of the parameters arranged in a menupage.

Order

SCOPE modulename in trigin width height trig_level trig_mode xgain ygain run

Resource Usage

moderate.

SEQUENCER — GROUP: MISCELLANEOUS

Mod Signal Sequencer — seq

The sequencer module is used to create a repetitive sequence of "mod" signal output values. This is very useful in creating a pitch arpeggiator with a pitch shifter.

The sequencer functions like a table lookup. A specified number of values are stored in memory, and depending on the value of the audio input signal, the corresponding value is sent to the output. To create a sequencer, a ramp waveform from an LFO drives the input of the sequencer, producing a consistent pattern of output values. (On the LFO, a ramp is a triangle with 100% duty cycle)

Specifiers:

n_steps

Controls how many entries there are in the sequencer table. Range 2 to 50.

Mod inputs:

in

This input controls which sequencer table entry is sent to the output. A value of 0 will cause the first entry to be output and a value of full scale will cause the last to be output. The remaining values will be output at evenly spaced thresholds between zero and full scale.

Mod outputs:

out

The sequencer output. This is at the "mod" rate of 1/4 the audio sample rate.

Control inputs:

fullscale

This control is used to scale the table entries so that more meaningful numbers may be used. The value entered here will control what table entry value will produce a fullscale output. For example, if this is set to 100, a table entry of 100 will produce a fullscale value at the sequencer output. A table entry of 50 would produce an output of 1/2 fullscale, etc.

Typically this number would be set to a value that makes sense for the modules that are being modulated by the sequencer output. For example, if you were modulating a pitch shifter, this parameter would be set to 1200 (for 1200 cents), and the pitch shifters modamount would be set to 1200 cents. With these settings, entries in the sequencer table correspond exactly to pitch values in "cents".

Range: -32768.0 to 32767.9.

glide

This control sets a time constant for the output of the sequencer. The sequencer output glides from value to value in the amount of time specified here. This is useful to smooth out transitions between values, or to create special effects, like portamento on a pitch shifter.

Range: 0 to 100 seconds.

step_value1 step_value2 ... step_valueN

These values are the sequencer entries that will be output. There is control input for each entry (as given by n_steps). Range: -fullscale to fullscale.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

SEQUENCER modulename n_steps in fullscale glide step_value1 step_value2 ... step_valueN

SINUS — GROUP: MATH

Sine Function — sin

This multichannel module returns either of two sine functions depending on its func specifier. If func is 0 the output is sin(pt), while if func is 1 the output becomes sin(pt/2). Alternatively one can consider it to be a sin(t) function where the input t is an audio or mod input from –1 to 1 that corresponds either to -180.. 180 (func: 0) or -90 .. 90 (func: 1). It will be useful in mathematics based algorithms, but with suitable input scaling will produce a smooth tube-type distortion as well.

	signal	min	max	description
Specifier				
	nterms	2	4	Indicates how many polynomial terms should be used to approximate the sine function. Higher values give a more accurate result but use more resources.
	nios	1	32	Indicates how many input/output channels are provided.
	func	0	1	Indicates the type of sine function. 0 : input swing from -1..1 gives output corresponding to -180..180 1 : input swing from -1..1 gives output corresponding to -90..90
Audio inputs				
	in1 .. N			N audio inputs for the N sine functions.
Audio outputs				
	out1 .. N			N Audio output from the N sine functions.

Order

SINUS modulename nterms nios func in1 in2 .. inN out1 out2 .. outN.

Resource Usage

low — nios: 4 — 6% (Orville)

SKNOB,SKNOB2,SKNOB3 — GROUP: INTERFACE

Versatile Knobs — skb

An sknob is a combination of a knob and a monitor, in that it produces a user-controllable output and displays the value of an input. If the input is connected to the output, it will act much like a normal knob. However, its reason to exist is the fact that the output can be bounded or otherwise processed before being fed to the input. In these cases, the relationship between then input, output and displayed values can be complex.
The maximum and minimum values can be set by control inputs, and the appearance (numeric, bar graph, round, etc) can be set by a specifier.
sknob2 and sknob3 also have a merge input - they act as a combination of an sknob and a c_merge.
sknob3 also has a default specifier, so that its initial value is saved.

	signal	min	max	description
Specifiers				
	knob type	0	3	determines the appearance of the knob 0: numeric 1: hfader 2: vfader 3: round
	default	-32768	32767	(sknob3 only) The initial value for the knob.
Control Inputs:				
	ctrl in	-32768	32767	The value to be displayed
	min in	-32768	32767	The minimum value to be output
	max in	-32768	32767	The maximum value to be output
	merge in	-32768	32767	(sknob2,3 only) a value that is sent to the output if it changes.
Control Outputs:				
	ctrl out	min in	max in	The output value of the knob
Userobjects				
	obj			A display object for the knob

Example Sigfile

```
HEADM "adc" 2 2 adc-null adc-null "Sknob Demo" "Empty" 1 menupage-obj ;=50,0,100,0
KNOB "min_knob" "min:  %2.1f" "min" -32767.0002 32766 1.0000 -10.0000 ;=100,25,100,0
KNOB "in_knob" "in:  %2.1f" "in" -32767.0002 32766 1.0000 0.0000 ;=250,25,100,0
KNOB "max_knob" "max:  %2.1f" "" -32767.0002 32766 1.0000 10.0000 ;=100,125,100,0
KNOB "merge_knob" "merge: %2.1f" "merge" -10 10 1 0 ;=100,225,100,0
```

SKNOB3 "sknob3" "sknob: %2.1f" "sknob" 0 1 5 in_knob-out min_knob-out max_knob-out merge_knob-out ;=400,100,100,0
MONITOR "monitor" sknob3-out "out: %2.1f" "out" ;=400,250,100,0
MENUPAGE "menupage" "Operations" "operate" 6 in_knob-obj min_knob-obj max_knob-obj merge_knob-obj sknob3-obj monitor-obj ;=550,125,100,0
TAIL "njr"

SLEW — GROUP: FILTER

Slew Rate Limit — slw

This module will limit the slew rate of a signal. The slew rate is how fast the signal changes. If you put a square wave into this function, you will get a ramping on the output. It can be an unusual type of low pass filter where only the loud high frequencies are attenuated.

Audio inputs:

in
Signal to be processed

Audio outputs:

out
The slew rate limited output

Control inputs:

pslew
positive slew limit. This is used when input is higher than output. The amount, in milliseconds, it takes to go from 0 to full scale. 0 disables function. Range: 0.0 to 30000.0 milliseconds.

nslew
negative slew limit This is used when input is lower than output. The amount, in milliseconds, it takes to go from 0 to full scale. Range: 0.0 to 30000.0 milliseconds.

Userobjects:

obj
Menupage of control inputs not connected to control signals. (collection)

Order:

SLEW modulename in pslew nslew

SOURCEANALYZER — GROUP: MISCELLANEOUS

Source Analyzer — src

This module is the "front-end" for the Ultrashifter(tm). It performs processing which is used by the *ultrashifter* module and as such is unlikely to be useful by itself. A built-in pitch detector's results are made available through various control outputs.

This module is very complex and is intended for experts only. The less experienced user may be well advised to tweak the existing presets, rather than trying to build new ones. Note that the Orville and 4000 versions are NOT compatible - a 4000 Ultrashifter(tm) program will NOT run on an Orville.

	signal	min	max	description
Audio Inputs:				
	in			The audio input signal to be analyzed.
Audio Outputs: (4000 only)				
	prtsA, prtsB, envA, envB, sout			Special signals for the ultra module.
Audio Outputs: (Orville only)				
	prts			Special signals for the ultra module.
Control Inputs:				
	minpitch	50	150 Hz	Controls the lowest pitch that the pitch detector will accept as valid. This control can be used to improve pitch detection on troublesome source material.
	maxpitch	150	1000 Hz	Controls the highest pitch that the pitch detector will accept as valid. This control can be used to improve pitch detection on troublesome source material.
	gatelevel	-100	0 dB	Controls the level below which no pitch detection will occur. If this is set too high the input pitch will not be tracked and problems will occur.
	speed	0	300	Controls how rapidly changes in the source material are reacted to.
	poles	10	36	Controls how closely the spectral properties of the signal are followed. Set to 20.
	maxp,spentry,special,adjust1,adjust2			Set to 0
Control Inputs: (4000 only)				
	uprc			Special signal from the ultra module
	pdamode	0	10	Selects the pitch detection mode. 0 - Normal. 10 - Unused.
Control Outputs:				
	pitch			The output of the pitch detector given in cents relative to middle C.
	period			The output of the pitch detector given as a period. The value is in milliseconds.
	freq			The output of the pitch detector given as a frequency in Hertz.

User Objects:

obj	This module may be treated as a menupage. If this module pointed to by head or by a menupage then if any of this module's control inputs are unconnected (left as *autoknob) they will be shown as knobs on a menu under PARAMETER. That menu will be titled "modulename parms".

Order (4000 only)

SOURCEANALYZER modulename in pdamode minpitch maxpitch gatelevel speed uprc maxp poles spentry special adjust1 adjust2

Order (Orville only)

SOURCEANALYZER modulename in minpitch maxpitch gatelevel speed maxp poles spentry special adjust1 adjust2

SPECTRUM — GROUP: MISCELLANEOUS

Spectrum Analyzer — spc — Orville and 7000 only

This module provides a 512 band FFT based spectrum analyzer. Each band is the same width, around 50Hz at 48k sampling. A userobject is provided that can give a realtime linear or log display.

	signal	min	max	description
Specifiers				
	width	10	240	display width in pixels. Full screen : 238, half screen : 119.
	height	10	64	display height in pixels. Suggestion: 41
Audio inputs				
	in	-1	1	The signal to be displayed
Control inputs				
	xgain	1	16	number of bands per pixel on the display. Suggestion: 2 for full screen, 4 for half screen to give 20kHz bandwidth at 48kHz.
	ygain	0	5	scaling factor for display 0 : auto - display scaled to fill screen 1 : 0dB 2 : +10dB 3 : +20dB 4 : +30dB 5 : +40dB
	disptype	0	3	type of display 0 : linear 1 : 1dB/pixel 40dB full screen 2 : 2dB/pixel 80dB full screen 3 : 3dB/pixel 120dB full screen
	mode	0	1	Allows the display to be frozen. 0 : stop 1 : run
	windowtype	0	4	A control for the way the data is analyzed. 0 : (Rectangular) Sharper peaks, at the expense of dynamic range. 4 : (Blackman) Best dynamic range, with some spreading of peaks.
	outputbin, windowsize			for experts only. Set to defaults: 1 1024
Control outputs				
	magnitude, array		for experts only	
Userobjects				
	display_obj			a screen to show the spectrum display, sized according to width and height above.
	scale_obj			a selector for disptype.
	gain_obj			a selector for ygain.
	menu_obj			A number of the parameters arranged in a menupage.

Order

SPECTRUM modulename in width height xgain ygain disptype run outputbin windowtype windowsize

SQRT — GROUP: MATH

Audio Signal Square Root — sqt

This module produces a signal which is the square root of its input signal. Its main value will be in magnitude calculations in mathematics-based algorithms.

signal	min	max	description

Audio Inputs:

in	-1	+1	Input signal.

Audio Outputs:

out	0	+1	Equal to the square root of the input signal for positive inputs, and zero for negative inputs.

Order

SQRT modulename in

STEREOMIXER — GROUP: MIXER

Multi-Input Stereo Audio Mixer — smx

The mix module provides a way to add (mix) two or more audio signals, with adjustable attenuation and pan on each input. The mixing is done with a dB control.

Specifiers:

ninputs

Specifies how many audio signals are to be mixed. Range: 2 to 50 inputs.

Audio inputs:

in1 in2 ... inN

One input for each of the audio signals.

Audio outputs:

left

left mix output

right

right mix output

Control inputs:

gain1 gain2 ... gainN

Controls the amount of attenuation for each input. The gain adjustment is a dB value. 0 dB = no attenuation, full level. Range: -100 to 0 dB

pan1 pan2 ... panN

Controls the left/right pan position for each of the inputs. Pan values of -1.0, 0.0, and 1.0 correspond to pan positions of left, center and right, respectively. Range: -1.0 to 1.0.

Userobjects:

obj

Menupage of control inputs not connected to control signals. (collection)

Order:

STEREOMIXER modulename ninputs in1 in2 ... inN gain1 gain2 ... gainN pan1 pan2 ... panN

STEREOSHIFT — GROUP: PITCHSHIFT

Stereo Pitch Shifter — ssh

This module shifts the pitch of a stereo input over a range of +/- four octaves. The pitch shifter also has a built-in pitch detector whose results are made available through various control outputs.

Audio inputs:

leftin rightin

The audio inputs to be pitch shifted.

Audio outputs:

left right

The pitch shifted outputs of the pitch shifter.

Mod Inputs:

mod

This audio input will modulate the amount of pitch shift. This is useful to create vibrato effects.

Control inputs:

minpitch

The minpitch control is used to optimize the pitch shifting algorithm. It sets the minimum pitch that pitch shifter is likely to hear. The values are as follows:

0 - C0
1 - C0#
2 - D0
...
46 - A3#
47 - B3

gatelevel

This control affects only the pitch detection output of this module. The gatelevel control determines at what level the pitch detector will output pitch readings. If the input signal level falls below the level set here, the pitch detect outputs will latch on to the old values. Range: -100 to 0 dB.

xfadetime
This control signal is used to optimize the sound of the pitch shifters. Larger settings may result in smoother overall sound but may add a "flanged" sound to the audio. Smaller settings will result in a crisper sound but may allow more audible pitch shifting artifacts. Range: 0 to 100 milliseconds.

shift
Controls the amount of pitch shift to be applied to the audio input. The adjustment is in "cents". A cent is one one-hundredth of a semitone. Positive value will shift the pitch upward and negative values will shift it downward. Range: -4800 to 4800 cents.

modamt
These control the amount of modulation to be applied to the pitch shifter. The adjustment is in cents and it represents the amount of pitch shift that would be added if the mod input was fully on. Range: -2400 to 2400 cents.

delayamt
This controls the amount of delay for the pitch shifter. Range: 0 to 600 milliseconds.

pan
This is used to optimize the pitch shifting algorithm. It sets which channel is more important. A 1 is right channel, a -1 is the left channel. Range: -1.0 to 1.0.

Control Outputs:

pitch
The output of the pitch detector given in cents relative to middle C.

period
The output of the pitch detector given as a period. The value is in milliseconds.

freq
The output of the pitch detector given as a frequency in Hertz.

amp
'amp' is the r.m.s. amplitude relative to full scale ('amp' equal to 1 would be a square wave 'hitting the rail').

tonality
A value representing how periodic the input signal is. A value a 1.0 is given for signals which are purely periodic. Lower values represent signals that are less periodic. The smallest value would be given for very noise-like signals.

Userobjects:

obj
Menupage of control inputs not connected to control signals. (collection)

Order:

STEREOSHIFT modulename leftin rightin minpitch gatelevel xfadetime mod shift modamt delayamt pan

---

STEREOTAPS — GROUP: DELAY

Multitap Delay with Individual Controls — stp

This module is a multi-tap delay (like the multitap module) that provides individual control signal inputs for adjusting the delay, amplitude, and panning of each delay tap.

Specifiers:

taps
Number of Taps. Range: 2 to 50

Audio inputs:

in
There is one audio input

Audio outputs:

left
Left output receives signals from the different taps depending on the settings of that tap's pan control.

right
Left output receives signals from the different taps depending on the settings of that tap's pan control.

Control inputs:

delayamt1 delayamt2 ... delayamtN
Controls the amount of t delay for each tap. Range: 0 to 660 milliseconds.

amp1 amp2 ... ampN
Controls the level (amplitude) for each tap. The level adjustment is a linear value (not dB), with 1.0 being no attenuation. Negative numbers will invert the phase of the signal. Range: 0.0 to 1.0.

pan1 pan2 ... panN
Controls the left/right pan position for each of the taps. Pan values of -1.0, 0.0, and 1.0 correspond to pan positions of left, center and right, respectively. Range: -1.0 to 1.0.

Userobjects:

obj
Menupage of control inputs not connected to control signals. (collection)

Order:

STEREOTAPS modulename taps in delayamt1 delayamt2 ... delayamtN amp1 amp2 ... ampN pan1 pan2 ... panN

SUBTRACT — GROUP: MATH

Subtract Two Audio Signals **sub**

This module subtracts one audio signal from another.

Audio inputs:

in1

signal to have something subtracted from.

in2

signal to be subtracted

Audio outputs:

out

The output (in1 - in2)

Order:

SUBTRACT modulename in1 in2

SWITCH — GROUP: MIXER

Audio Signal Switch **swi**

This module selects one of N audio inputs to be passed to the output.

Specifiers:

ninputs

Specifies how many inputs the switch will select from. Range: 2 to 1024.

Audio inputs:

in1 in2 ... inN

There are ninput audio inputs.

Audio outputs:

out

The audio signal from one and only one audio input as specified by select.

Control inputs:

select

This controls which of the audio inputs will be passed along to the switch output. A value of zero will select the first audio input, a value of 1 selects the second, etc. Range: 0 to ninputs-1.

Userobjects:

obj

The control input if it is not connected to a control signal.

Order:

SWITCH modulename ninputs select in1 in2 ... inN

TAPKNOB — GROUP: INTERFACE

Tapered Knob **knb**

tapknob gives you a knob with a number of different tapers. You supply a minimimum and maximum and the number of steps in between. The steps are not uniform and the non-uniformity is controlled by taper. A 0 taper is linear. tapknob allows you to select the resolution of the knob by setting the number of steps. The actual value of the knob at each step is automatically generated by the tapknob module. You only need to select the taper, min->max and number of steps.

Specifiers:

description

26 character description for PARAMETER menu. Use %.?s format for the number where ? is the number of displayed digits. This should fit into 19 characters

short name

8 character description, for future use.

min

Minimum value. This is actually a misnomer for this particular knob. This value is the value you will get if the knob is rotated fully counter-clockwise. This value can actually be higher than the maximum value. Range: -32768.0 to 0.

max

Value that the knob generates if the knob is rotated fully clockwise. See min. Range: min to 32767.0.

steps

This is the number of different values that the knob can generate. This controls the rate of knob change versus knob rotation speed. Range: 1 to 32767 number of steps.

taper

What kind of taper to use on the knob.

0 - linear taper

1 - square taper

2 - cube taper
3 - inverse square taper
4 - inverse cube taper
5 - S taper

default
value which the knob will be set to when first used. Range: from min to max.

Control outputs:

out
knob output

Userobjects:

obj
The userobject for this knob.

Order:

TAPKNOB modulename menutext shortname min max steps taper default

---

TEXTBLOCK — GROUP: INTERFACE

A block of text to be displayed on the screen — txt

Use this module to place a block of text on a display page. You control how high and wide. There is a number you specify that tells the system how many lines of text. If there are more than 4 lines, the system will allow you to scroll thru the text. The width of the text block is determined by reading all the lines and picking the longest. In your patch, place this module's userobject on a menupage.
The maximum characters per line is 40. However, If you have more than 4 lines, You will only see 39. The system places an arrow on the left to indicate that you can scroll. Each line uses 42 bytes of storage. Even if there is 1 character.

Specifiers:

nlines
how many lines of text, 1 to 100

text1 text2 ... textN
lines of text from 0 to 40 characters

Userobjects:

obj
The userobject for this block of text.(collection)

Order:

KNOB modulename nlines text1 text2 ... textN

TEXTKNOB — GROUP: INTERFACE

Text Knob — tkb

You can insert a knob into your program that allows a user to select from multiple text items. The knob outputs a number which indicates which text string was selected. You'll have to use multiply or c_table to make the number into the value you'll probably need. One interesting application for this module is to allow a user to select a note, as in "MODULES1C, 4 middle C" or "A, 4 A-440".

Specifiers:

menutext
description for PARAMETER menu, use %s format.

shortname
8 character description, for future use.

nvalues
number of selectable values. Range: 0 to 100.

default
value knob will be set to when first used. Range: 0 to 100.

value1 value2 ... valueN
Text to be displayed in place of %s, for value #N.

Control outputs:

out
knob output

Userobjects:

obj
The userobject for this textknob.

Order:

TEXTKNOB modulename menutext shortname nvalues default value1 value2 ... valueN

TEXTLINE — GROUP: INTERFACE

A line of text to be displayed on the screen — txt

Simply a line of text the size of a knob. You can use this to place a short message alongside knobs and such.

Specifiers:

text

The text you want to show. Size: 20 characters.

Userobjects:

obj

The userobject for this line of text.

Order:

KNOB modulename text

TEXTTRIGGER — GROUP: INTERFACE

Control trigger with variable name — ttg — v2.3

This module is similar to trigger, putting a button on the screen or the softkeys, with the difference that its name may be selected by means of a control input.

	signal	min	max	description
Specifier				
	nstrings	1	10	the number of different names the trigger may have.
	text1	8 chars/text		name when textnum = 0
	text2	8 chars/text		name when textnum = 1
	....			
	textn	8 chars/text		name when textnum = *nstrings - 1*
Control inputs				
	textnum	0	*nstrings-1*	sets the name of the trigger, by selecting from the textn strings.
Userobjects				
	obj			The userobject for the trigger to be placed on a menupage or the head userobj inputs.

Order

TEXTTRIGGER, modulename, nstrings, textnum, text1, text2,..textn

TMENUPAGE — GROUP: INTERFACE

Menupage with variable name — tmn — v2.3

This module is similar to menupage, creating an on-screen menu page, with the difference that the name on its soft key may be selected by means of a control input. It is suggested that the *hmenupage* module be used in place of this one due to its greater versatility.

	signal	min	max	description
Specifier				
	nstrings	1	10	the number of different names the menupage may have.
	text1	8 chars/text		name when textnum = 0
	text2	8 chars/text		name when textnum = 1
	....			
	textn	8 chars/text		name when textnum = *nstrings-1*
	description	19 chars/text		This is the text that will show up in the upper right of the PARAMETER screen when this menupage is selected
	8charname	8 chars/text		not currently used
	entries	0	32	number of userobject inputs
Control inputs				
	textnum	0	nstrings-1	sets the name of the menupage, by selecting from the text$_n$ strings.
Userobject Inputs				
	object1			may be a knob, other menupage or trigger, etc.
	....			
	objectn			
Userobjects				
	obj			The userobject for the menupage to be placed on another menupage or the *head* userobj inputs.

Order

TMENUPAGE, modulename, nstrings, textnum, description,8charname entries, object1, object2,….objectn, text1, text2,..textn

Example sigfile (also covers TEXTTRIGGER)
HEAD "adc" adc-null adc-null "TEXTTRIG example" "" 3 tmenupage-obj texttrigger-obj info-obj
TEXTTRIGGER "texttrigger" 3 knob-out "trig1" "trig2" "trig3"
TMENUPAGE "tmenupage" 3 knob1-out "" "" 3 knob-obj knob1-obj texttrigger-obj "menu1" "menu2" "menu3"
TEXTBLOCK "info" 3 "A simple program to demonstrate the use" " of the TEXTTRIG and TMENUPAGE modules" "Nothing in, nothing out."
KNOB "knob" "trig: %2.0f" "trig" 1 10 1 2
KNOB "knob1" "menu: %2.0f" "menu" 1 10 1 2
TAIL "njr"

TMONITOR — GROUP: INTERFACE

Text monitor — tmn

Just like *textknob* except that it monitors a controls signal. If the control signal is .5 or less, it shows the first text value. If the control signal is between .5 and 1.5 it will show the next text value, etc.

Specifiers:

longname
Text description for PARAMETER menu, use %s format

shortname
8 character description, for future use.

nvalues
number of selectable values. Range: 0 to 100.

value1 value2 ... valueN
text to be displayed in place of %s, for value #N.

Control inputs:

in
input to monitor

Userobjects:

obj
The userobject for this monitor.

Order:

TMONITOR modulename in longname shortname nvalues value1 value2 .. valueN

TONE — GROUP: FILTER

Audio Tone Control — ton

The tone control module provides a simple tone control equalizer. It has a gentle low and high shelving control with adjustable frequencies.

Audio inputs:

in
The signal to be processed.

Audio outputs:

out
The processed signal.

Control inputs:

low_freq
Adjusts the frequency at which the low shelving filter begins affecting the audio. Range: 0 to 20000 Hertz.

low_level
Control how much the low frequencies are boosted or cut. This is like a "bass" control. Range: -20 to 20 dB.

high_freq
Adjusts the frequency at which the high shelving filter begins affecting the audio. Range: 20 to 20000 Hertz.

high_level
Control how much the high frequencies are boosted or cut. This is like a "treble" control. Range: -20 to 20 dB.

Userobjects:

obj
Menupage of control inputs not connected to control signals. (collection)

Order:

TONE modulename in low_freq low_level high_freq high_level

HARMONIZER® MODULES

TRIGGER — GROUP: INTERFACE

Manual generation of A Control Signal trigger trg

Puts a button on the screen that when you highlight and press select will produce a control signal impulse. Normally the output of this module is zero, but when the button is hit, a short high is produced. This can be used to start events and such. If this module is connected directly to the HEAD's userobject inputs, a soft key will be created.

Specifiers:

name
description for PARAMETER menu

tag
description, for soft key name

Control outputs:

out
trigger output

Userobjects:

obj
The userobject for this trigger.

Order:

TRIGGER modulename name tag

ULTRASHIFTER — GROUP: PITCHSHIFT

Formant Correct Pitch Shifter ush

This module can pitch shift a vocal two octaves up or one octave down while maintaining a natural vocal quality. It can also alter the overall formant structure of a vocal signal independently of any pitch shift. Ultrashifter is optimized for vocal signals although it may be suitable for other monophonic source material.
Due to the extensive processing performed by this module, the input signal will be delayed a total of 50 milliseconds (a delayed dry signal is available for mixing). By comparison, Eventide's other pitch shifters typically delay the signal 20-25mS.

This module must be connected to the *sourceanalyzer* module in order to function. The combination of these two modules will use ALL the available DSP power on a 4000 - no other signal handling modules can be added. However, a simple mixer is included for wet/dry mixing and panning.

The Orville version omits the mixer but has a number of other enhancements and thus the combination uses about 50% of an Orville DSP.

This module is very complex and is intended for experts only. The less experienced user may be well advised to tweak the existing presets, rather than trying to build new ones. Note that the Orville and 4000 versions are NOT compatible - a 4000 Ultrashifter(tm) program will NOT run on an Orville.

	signal	min	max	description
Audio Inputs: (4000 only)				
	prtsA,prtsB,envA,envB,sin			Special signals from the *sourceanalyzer* module.
	dryin			The dry audio signal, used by built in mixer only.
Audio Inputs: (Orville only)				
	prts			Special signals from the *sourceanalyzer* module.
Audio Outputs:(4000 only)				
	left			The left output of the built in mixer.
	right			The right output of the built in mixer.
Audio Outputs:(Orville only)				
	out			The output signal.
Control Inputs:				
	pitchshift	-2400	2400 cents	Controls the amount of pitch shift. This adjustment is in "cents"; a cent is a hundredth of a semitone. Positive values shift the pitch up, negative values shift it down.
	formantshift	-2400	2400 cents	Controls the amount the overall formant structure is shifted (in *formantmode* 3 only, see below). This adjustment is in "cents"; a cent is a hundredth of a semitone. Positive values shift the formant up, negative values shift it down.
	formantscale	-90	200 percent.	Controls the scaling of the overall formant structure (in *formantmode* 1 & 2 only, see below).
	shiftmode	0	1	Selects how pitch shifting is to be performed. 0 - Regular. Less likely to glitch badly but average quality is lower. 1- High. Glitches are more noticeable, not good for polyphonic input.

formantmode	0	3	Selects how the formant structure is to be modified. 0 - Unmodified. No modifications made, operates more like a regular pitch shifter. Less artifacts generated. 1 - Linked. Modified according to the current pitch shift value to preserve naturalness, may be modified by *formantscale* control only. 2 - Unlinked1. Modified according to the *formantscale* control only. 3 - Unlinked2. Modified according to the *formantshift* control only, which is in cents.
spentry			This control is a special input used in conjunction with the scales module to support diatonic pitch shifting.

Control Inputs: (Orville only)

delayamt	50	70 mS	Controls the amount of delay. Note that this value includes the processing delay and thus can't be less than 50.

Control Inputs: (4000 only)

wetgain	-100	6 dB.	This controls the gain applied to the wet signal.
wetamp	0	1	Controls the attenuation applied to the wet signal.
wetpan	-1	1	Controls the left/right output balance of the wet signal.
wetdelayamt	50	70 mS	Controls the amount of wet delay. Note that this value includes the processing delay and thus can't be less than 50.
dryamp	0	1	Controls the attenuation applied to the dry signal.
drypan	-1	1	Controls the left/right output balance of the dry signal.
drydelayamt	50	100 mS	Controls the amount of dry delay.
gatelevel	-100	0	Not currently used.
special			Used to control special features.

User Objects:

obj	This module may be treated as a menupage. If this module pointed to by head or by a menupage then if any of this module's control inputs are unconnected (left as *autoknob) they will be shown as knobs on a menu under PARAMETER. That menu will be titled "<modulename> parms".

Order (4000 only)

ULTRASHIFTER prtsA prtsB envA envB sin dryin pitchshift formantshift formantscale shiftmode formantmode wetgain wetamp wetpan wetdelayamt dryamp drypan drydelayamt gatelevel spentry special adjust1 adjust2

Order (Orville only)

ULTRASHIFTER prts pitchshift formantshift formantscale shiftmode formantmode delayamt spentry special adjust1 adjust2

VFADER GROUP: INTERFACE

Vertical Fader Knob fdr

knob with graphical representation. Rotating the knob causes the 'low 1' to slide up or down. Up to six *vfaders* may be pictured on a single screen. Three vfaders may share a display page with four knobs.

Specifiers:

longname
description for future use.

shortname
6 character description for PARAMETER menu

min
minimum value. Range: -32768.0 to 0.0.

max
maximum value. Range: *min* to 32767.0.

resolution
step rate. Range: 0.0 to 32767.0

default
value which the knob will be set to when first used. Range: *min* to *max*.

Control outputs:

out
knob output.

Userobjects:

obj
The userobject for this fader.

Order:

VFADER modulename longname shortname min max resolution default

VMONITOR GROUP: INTERFACE

Vertical Monitor of a Control Signal vmn

This module creates a graphical display much like a VFADER that shows the value of its control input. You need to provide the upper and lower bounds so the movement of the display is meaningful. The display will indicate if the input has gone beyond those bounds.

Specifiers:

longname

Text Statement, including %f format, which describes how the monitor signal will be displayed.

shortname

8 characters or less of text which describes the monitored signal. This text is not displayed by the DSP4000 at any time but may be used in future products

min

minimum value

max

maximum value

Control inputs:

in

Signal to be displayed

Userobjects:

obj

The userobject for this monitor.

Order:

VMONITOR modulename in min max longname shortname

WAVEFORM GROUP: OSCILLATOR

Programable Waveform Audio Oscillator wfm

The waveform module produces a waveform of variable shape and frequency. It is much like the oscillator module except you define the waveform by adjusting 32 points along the waveform. The waveform is then derived from drawing straight lines between the points (linear interpolation). This works much like a CURVE module.

Typically the oscillator will be used to generate an audio range waveform. It is useful in creating synthesis effects and for very fast modulations.

Specifiers:

point1 point2 ... point32

32 values that define the waveform.

Mod Inputs:

mod

Modulates the frequency of the oscillator.

Audio outputs:

out

The oscillator output. level is at +20dBm

Control inputs:

freq

Controls the rate of the oscillator. Range: 0 to 20000 Hertz.

modamt

This controls how much the mod input affects the rate of the oscillator. Range: -20000 to 20000 Hertz.

Userobjects:

obj

A collection of the control inputs and the waveform

Order:

WAVEFORM modulename mod freq modamt point1 point2 ... point32

This module can be edited under Vsigfile using the Waveform editor.