The Audio Computing Engine [ACE] Bundle from Dome Music Technologies

The Audio Computing Engine (hereafter referred to as "ACE") is a set of modules for Cherry Audio Voltage Modular, based on the core functionality of an analogue computer:

ACE Integrator – As in the analogue computer world, the Integrator is the core workhorse of **ACE**. At its most basic, it outputs a rising or falling slope signal based on the polarity and level of the input voltage. However, by combining several Integrators and adding the other **ACE** modules, you can configure VM patches to perform a wide variety of useful and creative CV and audio functions.

ACE Constants & Multipliers – A combination of precision DC voltage sources or constant-value multipliers. Designed to set the coefficients of **ACE** calculations, they can also be used as attenuverters or gain stages (up to +/- 10x).

ACE Comparator – Detects when the X input is greater than or less than the Y input (or a constant value). A logic-level Overflow output and analogue clipped X output are provided.

ACE Four Quadrant Multipliers – Three independent precision linear multipliers which allow X and Y inputs to be positive or negative values. There is a mathematically precise output, which multiplies X and Y voltage inputs on a volt-for-volt basis. There is also an 'Audio' output, which divides the resultant signal by 5. This allows you to use it as a ring modulator or VCA within the VM 'standard' voltage range of +/- 5V.

ACE Powers – Raises the values 2, *e* or 10 to the power of the input voltage. 2 to the power X is very handy for changing linear-response (volts per Hz) oscillators into exponential-response (volts per octave).

ACE Min-Max – The Min-Max module takes up to four inputs and provides two outputs which represent the minimum and maximum values out of the inputs. This is the analogue equivalent of a Boolean logic OR (maximum) or AND (minimum) function.

ACE SR-Latch – Combines two edge-detectors (both switchable to rising or falling edge) and a Set-Reset Latch (flip-flop) circuit. This is very useful for detecting events and controlling the initialisation, starting and halting of an **ACE** program.

ACE Delay Line – Allows the programmer to set up delay compensation as values and results propagate from one operation to the next. It also allows precise control over event sequencing such as setting initial conditions and starting a program running on every key press.

ACE Notes (10HP) – A simple editable blank panel on which to keep notes about what your patch is doing.

So far, the **ACE** modules have been combined to build the following useful modular subsystems:

Slew Limiter (Linear, Exp or Log), Half-Wave and Full-Wave Rectifiers, Relaxation Oscillator / Function Generator (Sawtooth, Pulse, Triangle, PWM, Exponential Curve), Harmonic Oscillator (Sine Quadrature), Damped Oscillator (for 'Pinging'), True Thru-Zero Oscillator ('Zeroscillator'), Deadband.

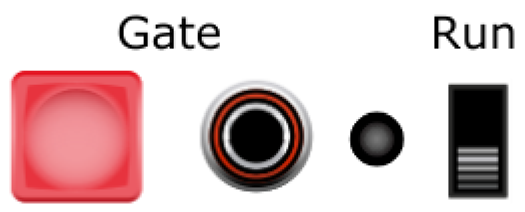
In addition, a lot of fascinating classic analogue computer programs can be implemented, such as a Bouncing Ball Simulation, Chaotic Attractor and Ballistic Trajectories.

However, perhaps the most fun to be had from the **ACE** is to create your own custom CV and audio modules without having to learn a single line of Java code or learn DSP techniques.

ACE Integrator

As mentioned in the introduction, the Integrator is the real heart of any analogue computer. This is true of the **ACE**, too. The function of an Integrator can be broadly understood within minutes. However, the full power and flexibility of the module can only be realised by taking time to learn the detailed interaction between all the switches, knobs and connections.

Run Controls



The Integrator will only process its inputs and internal state when in Run mode. When Run is off, the output will remain static. From left to right, the controls are:

Gate Push Button – Will run when pressed and freeze when released.

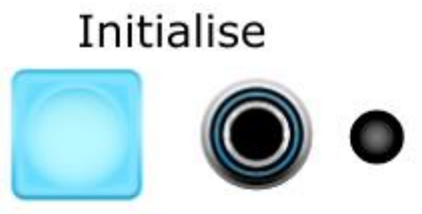
Gate Input – Will run when input voltage exceeds 2.5V, and freeze when below.

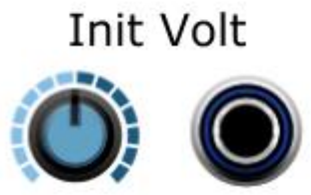
Run LED – Illuminates red when running, blank when frozen.

Run Switch – Runs continuously when in the Run position, frozen when off.

Note that the three Run inputs (Button, Gate Input and Switch) are logically OR'ed together, so if any one of them is in the Run state, the Integrator will be running. It will only freeze when all three are off.

Initialisation Controls





The Initialisation section allows you to inject the initial voltage level from which the module will start to integrate when it starts running.

The Init Volt knob allows you to specify a static voltage from -10V to +10V.

The Init Volt CV Input allows you to inject a dynamic initial voltage level from an external CV source. The value of the Init Volt Knob is added to the Init Volt CV Input. Set the knob to 0.0V if you want the initial voltage to be determined completely by the external CV Input.

The Initialise Push Button will inject the Init Volt level for as long as it is depressed. When released, integration will take off from that level (if in the Run state).

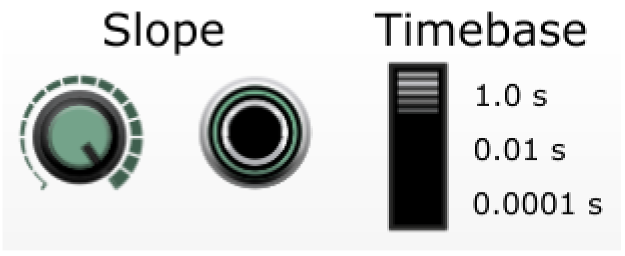
The Initialise CV Input will inject the Init Volt level for as long as it is above 2.5V. When it falls below this value, integration will take off from the Init Volt level (if in the Run state). If you want to inject an initial voltage only for one sample period, then an Edge Detector from the SR-Latch module fits the bill nicely.

The Initialise Push Button and CV Input are logically OR'ed together. The Initialise LED will illuminate blue for as long as the CV Input is above 2.5V, or for as long as the button is depressed.

Vin Socket

The Vin Socket is where to enter the signal to be integrated over time. In practical, modular synthesis terms, the value at Vin dictates the *rate* at which the output voltage rises or falls. Positive values at Vin will cause Vout to *rise* at a steady rate. The rate of rise increases as Vin increases in a positive direction. Similarly, negative values at Vin will cause Vout to *fall* at a steady rate. The rate of fall increases as Vin increases in a negative direction.

Timebase and Slope Controls



The Timebase Switch controls the speed of integration:

In the **1.0 s** position, the output voltage will increase by **1V per second** for every volt at the Vin socket. This is most suitable for slow CV processes, such as LFO, envelopes or portamento.

In the **0.01 s** position, the output voltage will increase by **1V per 0.01 seconds** (100V per second) for every volt at the Vin socket. (i.e., the output increases 100 times faster than in the **1.0 s** position). This is most suitable for fast LFOs or even the lower end of audio rate processes.

In the **0.0001 s** position, the output voltage will increase by **1V per 0.0001** seconds (10,000V per second) for every volt at the Vin socket. (i.e., the output increases 10,000 times faster than in the **1.0 s** position). This is most suitable for audio rate processing.

The Slope Knob allows you to fine-tune the timebase by multiplying the rate by a factor of 0.0 (hard-anticlockwise) to 1.0 (hard-clockwise). In practical terms, it allows you to run your program at normal speed (1.0), or progressively slow it down until it comes to a complete halt at 0.0. For example: If the Timebase Switch is set to **1.0 s**, and voltage of 3V is present at Vin, and the Slope Knob is set to 0.25, then the output will rise at a rate of (3V x 0.25 = 0.75V) per second.

The Slope CV Input Socket allows further modification of the Slope value by an external CV source. If the socket is disconnected, then the setting of the Slope Knob overrides everything. However, if an external source is connected, then the Slope Knob acts as an attenuator on the CV input. For example, if you had a unipolar 5V square wave LFO connected to the Slope CV Input, and you set the Slope Knob to 0.2, and the Timebase Switch was set to 1.0 s, then during the 'high' phase of the LFO, the output would rise at (5V x 0.2 = 1V per second for every volt at Vin). During the 'low' phase, the LFO outputs 0V (as it is unipolar). This would make the output rise at (0V x 0.2 = 0V per second for every volt at Vin. In other words, during the 'high' phase of the LFO, the integrator runs at full speed, and during the 'low' phase, it comes to a complete halt.

One interesting aspect of the Slope CV Input is that *negative* control voltages are just as valid as positive voltages. This allows programs to run 'backwards in time', which is essential for applications such as a Thru-Zero Oscillator ('Zeroscillator').

It takes a little time to wrap your head around the relationships between the Timebase Switch, Vin, the Slope Knob and the Slope CV Input. The rise/fall rate at the output is determined by this formula:

Rise Rate (Volts per second) = (Vin x Slope Knob x Slope CV In) / Timebase

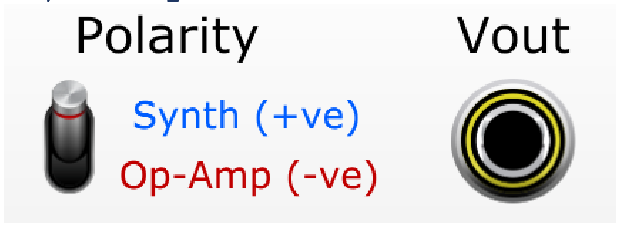
If the Slope CV input is disconnected, the formula simplifies to:

Rise Rate (Volts per second) = (Vin x Slope Knob) / Timebase

As always with modular synthesis, there is no substitute for trying it out for yourself.

Important: If Vin is disconnected, or has zero volts present, then the output voltage will not change at all. Similarly, if the Slope Knob is set to 0.0, the output voltage will not change, either.

Output Polarity Mode Switch and Vout

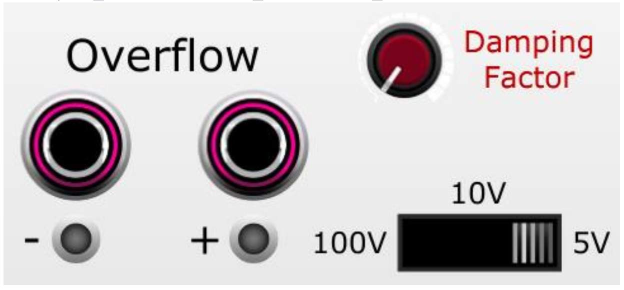


The output of the Integrator is presented at the Vout socket.

The Polarity switch can be swapped between positive and negative. Most of the time, it is most useful to set it to the "Synth (+ve)" position, so that the output will rise at a steady rate if Vin is positive. However, there are some situations where the opposite response is more useful. One example is when two Integrators are chained in a harmonic oscillator feedback loop.

The negative polarity output position is called "Op-Amp" for historical reasons; electronic analogue computers depended heavily on Operational Amplifiers to process their programs. When configured as an integrator, an Op-Amp will invert the output, because it uses negative feedback to perform that operation. As a result of this, analogue computer programmers would have to rearrange the terms in their equations to account for this inversion. If you attempt to implement a 'classic' analogue computer algorithm from a book or website, it makes sense to switch the output of the ACE Integrator to its "Op-Amp" position. This will be simpler than trying to turn the equations 'the right way up'!

Damping Factor, Range-Limiting and Overflow Detection



The Integrator can operate within three output voltage ranges: +/- 5V, +/- 10V and +/- 100V. Choosing the +/- 5V range ensures that the output can be used as an audio signal within the Voltage Modular environment. The 10V and 100V ranges are provided to help avoid ‘runaway’ positive-feedback situations.

Note that the output range limiting switch setting does not affect the range of allowable *input* voltages on any of the Integrator’s sockets.

If the calculated output voltage exceeds the range values (positive or negative), Vout will be clipped to that value. In addition, the appropriate Overflow Output will be set to +5V, and the corresponding LED will be illuminated. This can be very helpful to ‘debug’ a program or patch when it’s not operating as expected.

The Damping Factor knob is not something that you would be likely to find on a ‘classic’ analogue computer. However, it has been included in the **ACE**, as it has proved itself to be incredibly useful for creative synthesis purposes (‘Pinged’ resonators, for example) and also to tame the module when it operates close to the edge! The Damping Factor works by feeding a fraction of Vout back to Vin, with its polarity inverted. “Negative Feedback” is used throughout the world of electronics to make active circuits more stable, and the technique applies just as well to the virtual circuitry of **ACE**!

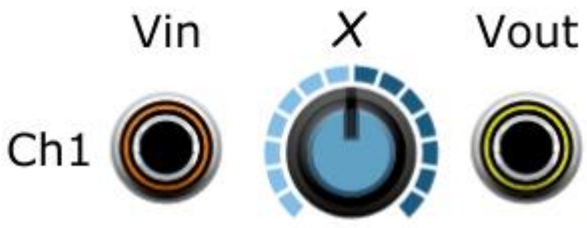
When set to 0.0 (fully counter-clockwise), no negative feedback is applied. This should be the default setting when creating a new patch, and it must **always** be set to zero when running a genuine analogue computer algorithm. When set to 1.0 (fully clockwise), -100% of Vout is fed back into Vin, leading to the greatest amount of damping.

Sometimes, an Integrator-based patch can become unstable, particularly at high audio-rate frequencies. Introducing a little bit of damping can help to bring it back under control, but be careful not to lose the magic of your crazy creation in the process!

ACE Constants & Multipliers

The Constants & Multipliers module allows you to set up the coefficients of formulas which you want to run on the **ACE**.

Or, in modular synth terms, it offers six channels of DC sources, attenuverters or gain stages!



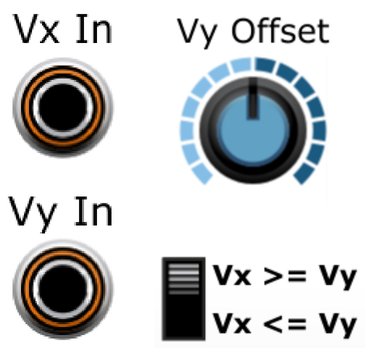
If the Vin socket is disconnected, the channel will operate as a static DC source, with values from -10V to +10V, as set by the *X* knob.

If the Vin socket is connected, the channel will operate as a static voltage multiplier, with values from times -10.0 to times +10.0, as set by the *X* knob. Note that this is quite a wide range, and bipolar in nature; in many cases the standard Cherry Audio attenuverters are more practical and ergonomic for signal scaling within patches.

If you require DC voltages or multiplication factors greater than +/- 10, you can cascade channels by connecting one channel's output to another channel's input. By cascading all six channels in this manner, you can get DC sources of +/-1,000,000V or multiplication factors of times 1,000,000 (roughly 120dB of gain)!

ACE Comparator

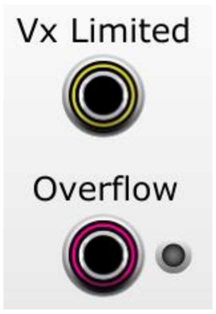
The Comparator allows you to compare two input voltages (Vx and Vy) and output a logic signal based on whether one is greater than or less than the other. In addition, there is an output which provides a copy of the Vx input, limited to Vy whenever the Vy threshold is exceeded.



Vx In and Vy In provide the two values which are compared against one another.

The Vy Offset Knob adds a DC offset of up to +/-10.0V to the Vy input. If Vy is disconnected, then Vx is compared against the absolute value of Vy Offset. If Vy is connected, then Vx is compared against the total sum of (Vy + Vy Offset).

The Comparison Operation Switch selects between Vx >= (Vy + Vy Offset), and Vx <= (Vy + Vy Offset).

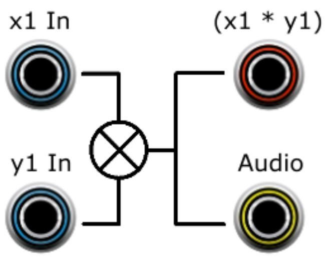


The Vx Limited output socket provides a copy of the Vx input when the comparison condition is not met, and is clipped to the value (Vy + Vy Offset) when the condition is met.

The Overflow output is set to 0V (logic low) when the comparison condition is not met, and 5V (logic high) when the condition is met. The Overflow LED is also illuminated when the comparison condition is met.

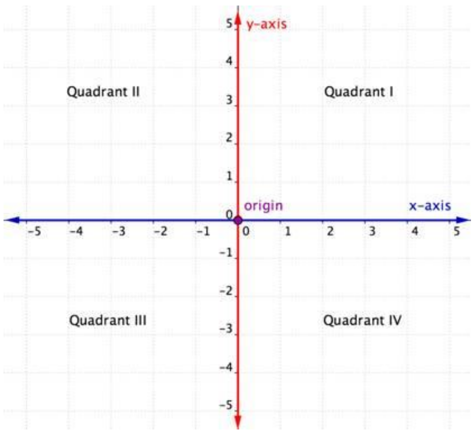
ACE Four-Quadrant Multipliers

The Four-Quadrant Multipliers module features three independent Four-Quadrant Multipliers. Each takes two input voltages and outputs the algebraic product of them. An 'Audio' output is also available which divides the product by 5, in order to remain within the standard limits of the Voltage Modular environment.



Bipolar DC voltages and AC voltages are allowed as inputs, and the output sign follows the inputs over all four quadrants:

Quadrant Number	X Input Sign	Y Input Sign	Output Sign
I	+	+	+
II	-	+	-
III	-	-	+
IV	+	-	-

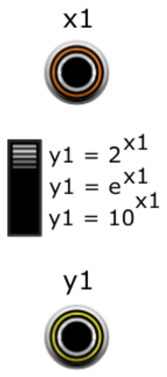


In terms of analogue computing, the Four-Quadrant Multipliers allow you to multiply two variables together to get their product. By feeding the same value into both X and Y inputs, you will get X^2 at the output. To get higher powers of X, you can cascade several Four-Quadrant Multipliers in series.

In synthesizer terms, the Four-Quadrant Multiplier is normally used as a Ring Modulator. However, it will also operate as a standard VCA if only positive values are used on one of the inputs. Hardware Ring Modulators are often AC-coupled on the inputs, but the ACE Four-Quadrant Multipliers will work equally as well with steady DC voltages.

ACE Powers

International Module of Mystery.



ACE Powers lets you raise a constant value to the power of the X input. There are three selectable constants which are useful within an analogue computer and modular synth environment:

Y = 2^X is very useful for applying Volts/Octave control over linear parameters.

Y = e^X is useful for creating exponential curves from linear inputs.

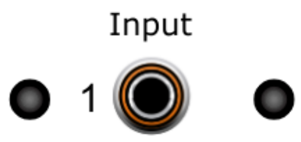
Y = 10^X is most useful for representing a number in exponential form, or scientific notation. For example, if the input was 6V, then the output would be 1,000,000V. Similarly, an input of -3V would produce an output of 0.001V. If this output is then fed to the input of a Constants and Multipliers channel, the channel's multiplier knob can be used to set the mantissa/significand of very large or very small floating-point numbers.

There are two channels in the ACE Powers module, which can be set independently of each other.

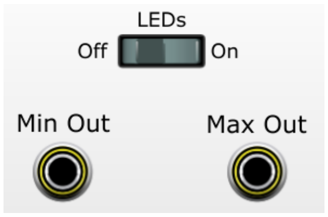
Note that the calculations within this module are performed with full 64-bit floating-point precision. This makes them fairly expensive in CPU time. However, if the X input value does not change from one sample to the next, it will not be recalculated. For example, if the 2^X function was used to translate the keyboard's 1V/Octave pitch to a linear multiplier value, then it would only recalculate the output value whenever a new note was played. However, if a continually-varying source is used as input (e.g., a VCO or LFO), it will recalculate the output on every sample period. Just something to be aware of – if you need to do it every sample, go for it!

ACE Min-Max

The **ACE** Min-Max module takes up to four inputs and outputs the overall minimum and maximum values. This is the analogue voltage equivalent of the Boolean logical operations OR (Maximum) and AND (Minimum).



Each input channel has an Input socket and two indicator LEDs. The left-hand LED is red, and will illuminate if this channel currently has the minimum value. The right-hand LED is green, and will illuminate if this channel currently has the maximum value.



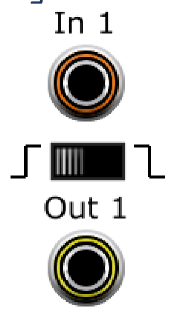
The LEDs Off/On switch allows you to switch off the LED indicators. This is useful if the flashing is distracting, or if the inputs are changing too quickly to provide any meaningful information.

The Min Out and Max Out sockets will output the minimum and maximum values of all input channels respectively. If no inputs are connected, the outputs will be 0V. If only one input is connected, then its value will appear on both Min Out and Max Out.

ACE SR-Latch

The **ACE** SR-Latch features two independent Edge Detectors and a Set-Reset Latch with complementary outputs. It is very useful for detecting events and controlling the initialisation, starting and halting of an **ACE** program. In modular synth terms, it can be very useful for controlling the sequential operation of envelope generators or cyclic LFOs.

Edge Detectors

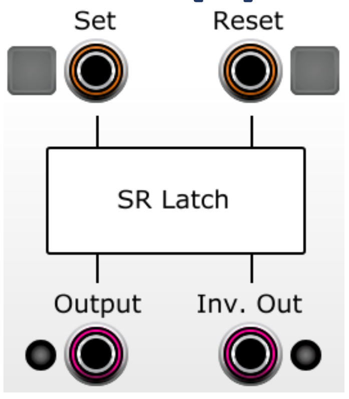


The Input socket accepts any analogue input signal. The threshold for detecting edge transitions is +2.5V.

The Edge Mode Switch selects between Rising Edge (Input voltage has gone from below 2.5V to above) on the left and Falling Edge (Input voltage has gone from above 2.5V to below) on the right.

The Output socket will output +5V for the single sample period when the edge was detected and 0V at all other times. This is particularly important for triggering an Initialisation event on the Integrator, as it will only force the injection of the Init Volt value for one sample, then let it run in normal mode from that point onwards.

The Set-Reset [SR] Latch



The operation of the SR-Latch is very straightforward. If the Set Input is high (>2.5V) and the Reset Input is low (<2.5V), the Output will be set high (5.0V). Conversely, if the Reset Input is high (>2.5V) and the Set Input is low (<2.5V), the Output will be set low (0.0V). If both Set and Reset are high or low at the same time, the Output state will not change. The Inverted Output will always be in the opposite state to the main Output (low when the main out is high, and high when the main output is low). A green LED to the left of the main Output is illuminated when the main out is high. A red LED to the right of the Inverted Output is illuminated when the Inv. Out is high.

The Set and Reset buttons can be used to manually set the SR-Latch to either state. These buttons are OR'ed with the Set and Reset CV input states.

Edge Detector Isolate/Connect Switches

Edge 1
to Set
Isolate
Connect
Edge 2
to Reset

The Edge 1 to Set switch will pass through Edge Detector 1’s output to the SR-Latch’s Set Input, when it is set to the “Connect” position. In the “Isolate” position, Edge Detector 1 will operate completely independently of the SR-Latch.

Similarly, the Edge 2 to Reset switch will pass through Edge Detector 2’s output to the SR-Latch’s Reset Input, when it is set to the “Connect” position. In the “Isolate” position, Edge Detector 2 will operate completely independently of the SR-Latch.

ACE Delay Line

The **ACE** Delay Line has 8 cascaded analogue shift registers which delay the input signal for 1 to 8 sample periods.

The Delay Line has two main functions within the **ACE** environment:

1. Each module in Voltage Modular requires at least one sample period before a signal at the input gets processed and then presented to the output. The Delay Line allows the programmer to ensure that values propagate through different processing paths with the same time delay, so that they can be operated on simultaneously when they 'join up' again.
2. It allows the programmer to have a series of events take place in a specific order, such as injecting initial conditions into Integrators, placing them into the Run state, and triggering state changes on SR-Latches.

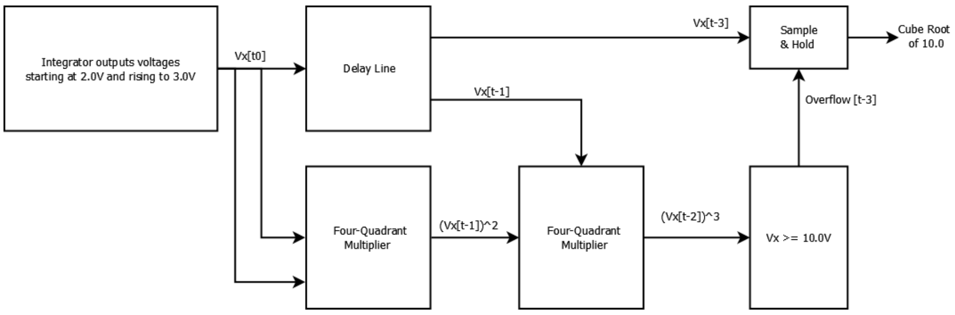
Input Socket [t0]

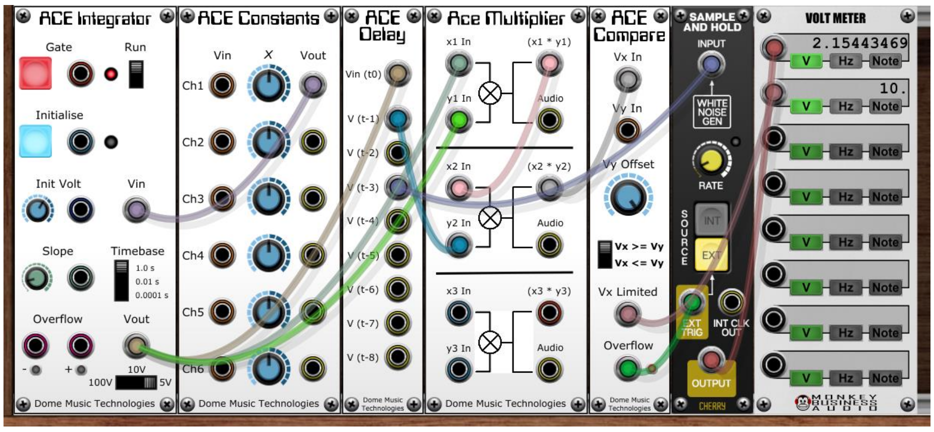
The signal which is to be delayed is connected to the Vin (t0) socket.

Output Sockets [t-1] to [t-8]

The V (t-*n*) socket outputs the value of Vin after *n* sample periods (1/48,000 of a second).

Example of program to find the cube root of 10:





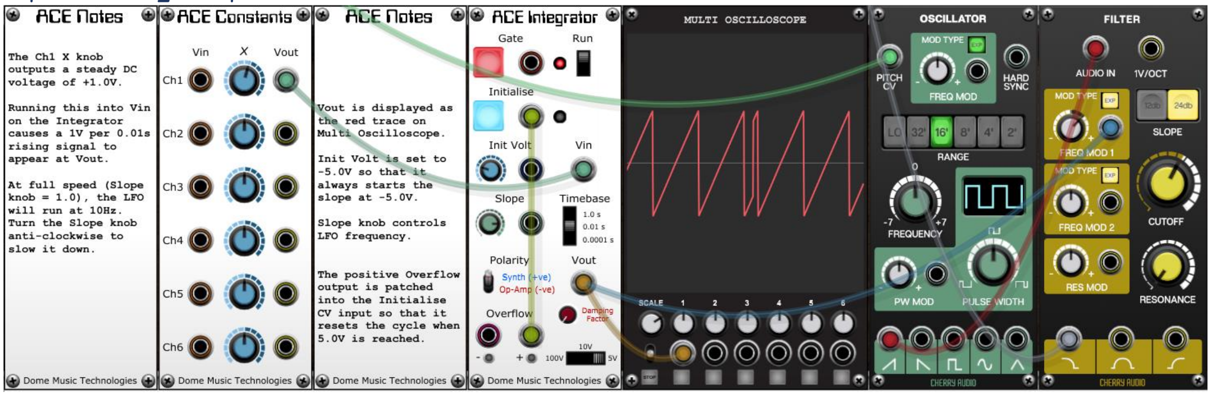
ACE Notes [10HP]

I think I'll just leave this one to explain itself…

ACE Notes
Double-click in this area to enter text.
Dome Music Technologies

Example Patches

Bipolar Rising Ramp LFO


The Ch1 X knob outputs a steady DC voltage of +1.0V.

Running this into Vin on the Integrator causes a 1V per 0.01s rising signal to appear at Vout.

At full speed (Slope knob = 1.0), the LFO will run at 10Hz. Turn the Slope knob anti-clockwise to slow it down.

Vout is displayed as the red trace on Multi Oscilloscope.

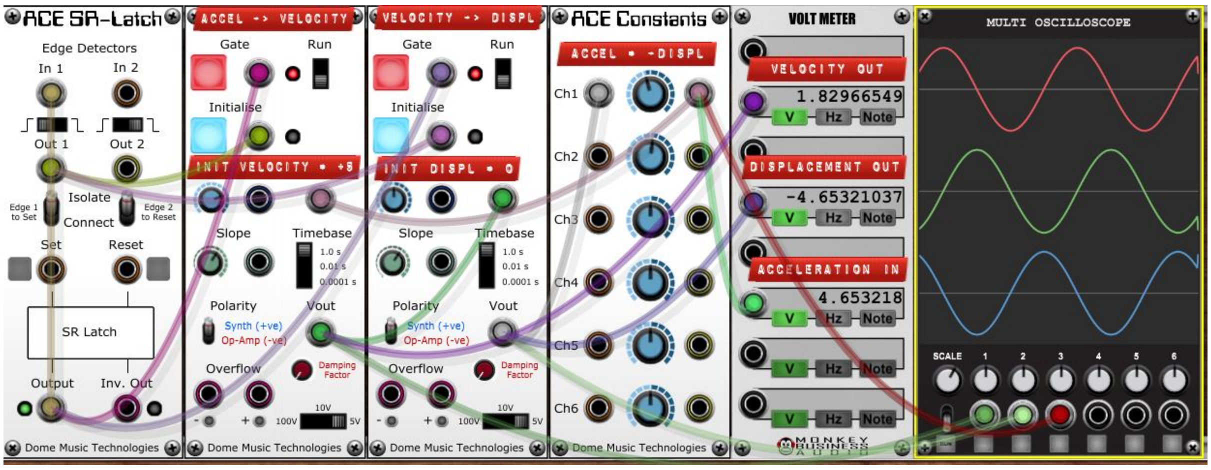
Init Volt is set to -5.0V so that it always starts the slope at -5.0V.

Slope knob controls LFO frequency.

The positive Overflow output is patched into the Initialise CV input so that it resets the cycle when 5.0V is reached.

In this patch, the Integrator is outputting a steadily rising voltage. The Init Volt knob is set to -5.0V, so that is where the output voltage starts. The input voltage to the Integrator is 1.0V and the Timebase switch is set to 0.01s. This means that it will take 0.1 seconds for the output to ramp up from -5.0V to +5.0V (a 10V range). The output range switch is set to 5V, which means that the +ve Overflow output will be set to 5V when Vout reaches +5V. By connecting the +ve Overflow output to the Initialise input, the rising cycle will immediately start over from -5.0V. The Slope knob can be used to change the LFO's frequency from 10Hz (fully clockwise) down to 0Hz (fully anti-clockwise). If you want to have a higher maximum frequency, just increase the value of the Ch1 knob on the ACE Constants panel; +3V will give you a max frequency of 30Hz and +10V will give you a max frequency of 100Hz. Setting the Ch1 knob to a negative value will kill the oscillations. The reasons for this are left as an exercise for the reader!

Sine-Wave Harmonic Oscillator


VELOCITY OUT
1.82966549
DISPLACEMENT OUT
-4.65321037
ACCELERATION IN
4.653218

A harmonic oscillator uses two Integrators linked in a feedback loop to generate a very pure sine wave output. In fact, it generates two complementary sine waves, separated by 90 degrees. The easiest way to understand its operation is to imagine that the left-hand Integrator takes an Acceleration value at Vin and outputs a Velocity value at Vout. The right-hand Integrator takes that Velocity value as its input, and outputs a Displacement value (bipolar distance) at its Vout. The clever bit is when you take the Displacement value, invert it, then feed it back into the left-hand Integrator as the new Acceleration value. Once you've got your head around how a harmonic oscillator works, you can reduce the number of modules required by removing the ACE Constants module, feeding the right-hand Vout to the left-hand Vin, then flicking the right-hand Polarity switch to "Op-Amp".

Further Information

Wikipedia is a great place to start learning about analogue computers and integrators:

https://en.wikipedia.org/wiki/Analog_computer#Electronic_analog_computers

https://en.wikipedia.org/wiki/Integrator

There are some fantastic videos on YouTube which explain the fundamentals of calculus in simple terms. I particularly like this series, created by 3Blue1Brown:

https://www.youtube.com/playlist?list=PL0-GT3co4r2wlh6UHTUeQsrf3mlS2lk6x

There are even videos which demonstrate how analogue computers can be integrated with modular synths, like:

"Sounds of Chaos" by Tim Thompson

Pt 1 - https://www.youtube.com/watch?v=ViDYAYrW9zI

Pt 2 - https://www.youtube.com/watch?v=MgAgP0OU6Ho

"Analog Computers for music" livestream by Hainbach, with Hans Kulk & Professor Bernd Ulmann

https://www.youtube.com/watch?v=bgyzeyatS-0

Limitations

Because the **ACE** is implemented in a digital, discrete-time environment, you can never find the "limit as delta-t tends to 0". This means that calculations can become unstable as the timebase starts to get close to the sampling rate of Voltage Modular (48kHz, or roughly 0.00002s per sample). This instability can become cumulative if you have a feedback loop in your patch; when a harmonic oscillator gets into the kHz range, you can experience inaccuracies in tuning and distortion in the output waveshape. Sometimes, introducing a degree of negative feedback via the "Damping Factor" knob can help to stabilise things.

Acknowledgements

I'd like to thank Andy Bloyce for agreeing to be a beta-tester for **ACE**, and for being a constant sounding-board during its development.